aws

使用者指南

Amazon Elastic Compute Cloud



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Elastic Compute Cloud: 使用者指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

什麼是 Amazon EC2？ 1
功能 1
相關服務 2
存取 EC2 3
定價 4
估算、計費和成本最佳化 5
Resources 6
開始教學課程 7
步驟 1：啟動執行個體 8
步驟 2：連線到您的執行個體 10
步驟 3：清理您的執行個體 12
後續步驟 13
最佳實務 14
Amazon Machine Image 16
AMI 特性 17
啟動許可 17
根磁碟區類型 18
確定 AMI 根磁碟區類型 19
虛擬化類型 21
尋找 AMI 22
Systems Manager 參數 25
Systems Manager 公有參數 28
中的付費 AMIs AWS Marketplace 32
在 中銷售您的 AMI AWS Marketplace 33
尋找已支付 AMI 33
購買已支付 AMI 35
擷取產品代碼 35
使用已支付支援 36
已支付及支援的 AMI 的帳單 37
管理您的 訂閱 37
AMI 生命週期 39
建立 AMI 40
建立 Amazon S3 支援的 AMI 48
使用 Windows Sysprep 建立 AMI 85

複製 AMI 100
存放和還原 AMI 115
AMI 原始伺服器 124
AMI 浮水印 129
AMI 允許的執行個體類型 135
AMI 用量 140
棄用 AMI 163
停用 AMI 168
取消註冊 AMI 174
開機模式 179
UEFI 開機模式需求 181
AMI 開機模式參數 182
執行個體類型開機模式 184
執行個體開機模式 187
作業系統開機模式 188
設定 AMI 開機模式 190
UEFI 變數 196
UEFI 安全開機 196
AMI 加密 212
執行個體啟動案例 212
映像複製案例 215
共用 AMI 217
已驗證的供應商 217
尋找共享 AMI 218
準備使用適用於 Linux 的共用 AMI 221
允許 AMI 222
使您的 AMI 公有化 242
封鎖 AMI 的公有存取權 245
與組織或組織單位共用 AMI 255
與特定 AWS 帳戶共用 AMI 265
取消與您的帳戶共用 AMI 269
建立共用 Linux AMI 的建議 270
監控 AMI 事件 275
事件詳細資訊 276
available 事件 277
failed 事件 278

deregistered 事件 ... 278
disabled 事件 ... 279
了解 AMI 帳單 ... 279
AMI 帳單欄位 ... 280
尋找 AMI 帳單資訊 ... 282
驗證帳單上的 AMI 費用 ... 284
AMI 配額 ... 285
請求增加 AMI 的配額 ... 286
執行個體 ... 287
執行個體類型 ... 288
可用的執行個體類型 ... 289
硬體規格 ... 289
Hypervisor 類型 ... 289
AMI 虛擬化類型 ... 291
處理器 ... 291
尋找執行個體類型 ... 293
EC2 執行個體類型搜尋工具 ... 300
Compute Optimizer 建議 ... 302
執行個體類型變更 ... 305
爆量效能執行個體 ... 312
GPU 執行個體 ... 359
Mac 執行個體 ... 372
EBS 最佳化 ... 423
CPU 選項 ... 553
AMD SEV-SNP ... 862
處理器狀態控制 ... 876
受管執行個體 ... 880
受管執行個體的帳單 ... 881
識別受管執行個體 ... 881
受管資源可見性設定 ... 883
開始使用受管執行個體 ... 887
巢狀虛擬化 ... 887
運作方式 ... 888
考量事項 ... 888
在啟用巢狀虛擬化的情況下啟動新的執行個體 ... 888
設定現有執行個體以使用巢狀虛擬化 ... 889

尋找支援巢狀虛擬化的執行個體類型 891
帳單與購買選項 892
隨需執行個體 893
預留執行個體 895
Spot 執行個體 955
專用執行個體 1044
專用執行個體 1099
Capacity Reservations 1106
啟動範本 1241
限制 1242
許可 1243
控制啟動執行個體 1249
建立 1251
修改 (管理版本) 1265
刪除 1269
啟動執行個體 1271
教學 1274
執行個體參數參考 1294
使用啟動執行個體精靈進行啟動 1305
使用啟動範本啟動 1308
從現有執行個體啟動 1315
從 AWS Marketplace AMI 啟動 1316
連線到您的執行個體 1318
一般連線先決條件 1320
使用 SSH 連線至您的 Linux 執行個體 1325
使用 RDP 連線至您的 Windows 執行個體 1340
使用工作階段管理員進行連接 1349
使用公有 IP 與 EC2 Instance Connect 建立連線 1350
使用私有 IP 與 EC2 Instance Connect 端點建立連線 1383
執行個體狀態變更 1413
依執行個體狀態計費 1413
待定執行個體 1414
停止的執行個體 1415
休眠的執行個體 1415
重新啟動執行個體 1416
終止的執行個體 1416

執行個體狀態之間的差異 1416
停止和啟動 1418
休眠 1436
重新啟動 1464
終止 1466
淘汰 1481
自動執行個體復原功能 1485
自動執行個體復原的核心概念 1486
簡化自動復原與基於 CloudWatch 動作的復原之間的差異 1488
建置彈性系統 1489
驗證是否發生自動復原 1489
簡化的自動復原 1491
基於 CloudWatch 動作的復原 1495
執行個體中繼資料 1499
執行個體中繼資料分類 1500
動態資料分類 1511
存取執行個體中繼資料 1511
設定 IMDS 選項 1549
啟動時執行命令 1583
範例：AMI 啟動索引值 1608
偵測主機是否為 EC2 執行個體 1612
檢查 執行個體身分文件 1613
檢查系統 UUID 1613
檢查系統虛擬機器產生識別符 1614
執行個體身分文件 1619
擷取執行個體身分文件 1620
驗證執行個體身分文件 1622
公用憑證 1632
STIG 合規 1695
STIG 設定 1696
下載 STIG 指令碼 1715
使用 Systems Manager 來套用 STIG 設定 1736
時鐘同步 1739
硬體封包時間戳記 1739
閏秒 1740
使用本機 Amazon Time Sync Service 1741

使用公有 Amazon Time Sync Service 1756
比較 Linux 執行個體的時間戳記 1758
變更執行個體的時區 1759
EC2 容量管理器 1762
主要功能 1762
啟用容量管理器 1763
整理資料 1774
在主控台導覽容量管理器 1789
匯出資料 1791
管理裝置驅動程式 1795
網路驅動程式 1795
圖形驅動程式 1796
儲存裝置驅動程式 1796
其他 Windows 裝置驅動程式 1796
AMD 驅動程式 1797
NVIDIA 驅動程式 1802
在 Windows 上安裝 ENA 驅動程式 1847
Windows PV 驅動程式 1863
NVMe 驅動程式 1893
其他 Windows 裝置驅動程式 1903
設定 Windows 執行個體 1904
Windows 特定的系統設定 1905
AWS Windows 執行個體的裝置驅動程式 1905
Windows 啟動代理程式 1907
EC2 Fast Launch for Windows 2047
變更 Windows 管理員密碼 2068
新增 Windows 系統元件 2070
在 Windows 安裝 WSL 2074
Windows 公用程式 2076
升級 Windows 執行個體 2078
執行就地升級 2079
執行自動升級 2082
遷移至 Nitro 型執行個體類型 2092
故障診斷升級 2099
教學課程：將 EC2 執行個體連線至 RDS 資料庫 2100
教學課程目標 2100

Context 2101
Architecture 2101
考量事項 2102
教學課程所需時間 2103
成本 2103
選項 1：使用 EC2 主控台自動連線 2103
選項 2：使用 RDS 主控台自動連線 2114
選項 3：手動連線 2124
機群 2134
功能和優勢 2134
要使用哪種機群方法？ 2135
組態選項 2136
請求類型 2137
花費限制 2165
屬性型執行個體類型選項 2167
執行個體權重 2200
分配策略 2203
容量重新平衡 2209
Capacity Reservations 2213
使用 EC2 機群 2214
EC2 Fleet 請求狀態 2215
EC2 Fleet 先決條件 2215
建立 EC2 Fleet 2220
標記 EC2 Fleet 2229
說明 EC2 機群 2231
修改 EC2 Fleet 2235
刪除 EC2 Fleet 2237
使用 Spot 機群 2242
Spot Fleet 請求狀態 2243
Spot Fleet 許可 2244
建立 Spot 機群 2254
標記 Spot Fleet 2262
說明 Spot 機群 2270
修改 Spot Fleet 請求 2272
取消 (刪除) Spot 機群請求 2274
Spot Fleet 的自動擴展 2277

監控您的機群 2285
使用 CloudWatch 監控您的機群 2286
使用 EventBridge 監控您的機群 2289
教學 2306
教學課程：使用執行個體權重設定 EC2 機群 2308
教學課程：將 EC2 機群設定為使用隨需執行個體做為主要容量 2311
教學課程：使用目標容量保留設定 EC2 機群啟動隨需執行個體 2312
教學課程：設定您的 EC2 機群以啟動執行個體至容量區塊 2318
教學課程：設定您的 EC2 機群，在可中斷的容量保留中啟動執行個體 2320
教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體 2325
EC2 機群的範例 CLI 組態 2330
範例 1：啟動 Spot 執行個體做為預設購買選項 2331
範例 2：啟動 隨需執行個體 做為預設購買選項 2331
範例 3：啟動 隨需執行個體 做為主要容量 2332
範例 4：使用多個容量保留來啟動隨需執行個體 2332
範例 5：當總目標容量超過未使用的容量保留數目時，使用容量保留來啟動隨需執行個體 2336
範例 6：使用目標容量保留啟動隨需執行個體 2339
範例 7：設定容量重新平衡，以啟動取代 Spot 執行個體 2342
範例 8：在容量最佳化機群中啟動 Spot 執行個體 2344
範例 9：使用優先順序在容量最佳化機群中啟動 Spot 執行個體 2345
範例 10：在 price-capacity-optimized 機群中啟動 Spot 執行個體 2346
範例 11：設定屬性型執行個體類型選取 2348
範例 12：使用容量保留資源群組在多個容量保留類型中啟動執行個體 2349
範例 CLI 組態 Spot 機群 2350
範例 1：價格最低的可用區域 2351
範例 2：清單中價格最低的可用區域 2352
範例 3：清單中價格最低的執行個體類型 2354
範例 4。覆寫請求的價格 2355
範例 5：使用多樣化配置策略 2357
範例 6：使用執行個體權重 2360
範例 7：使用隨需容量 2361
範例 8：設定容量重新平衡 2362
範例 9：使用容量最佳化機群 2364
範例 10：使用具有優先順序的容量最佳化機群 2365
範例 11：使用 priceCapacityOptimized 機群 2366

範例 12：設定屬性型執行個體類型選取 2367
機群配額 2368
請求增加目標容量的配額 2369
聯網 2371
區域 (Region) 和區域 (Zone) 2372
大區 (Regions) 2372
可用區域 2373
本機區域 2374
Wavelength 區域 2375
AWS Outposts 2377
執行個體 IP 定址 2378
私有 IPv4 地址 2379
公有 IPv4 地址 2380
公有 IPv4 地址最佳化 2381
IPv6 地址 2382
多個 IP 地址 2383
EC2 執行個體主機名稱 2384
Link-local 地址 2385
IPv4 地址 2385
IPv6 地址 2391
次要 IP 位址 2396
Windows 上的 IPv4 位址 2400
EC2 執行個體主機名稱與網域 2407
了解 EC2 執行個體主機名稱與網域 2407
主機名稱類型 2408
使用自有 IP 地址 2421
BYOIP 定義 2422
需求與配額 2423
區域可用性 2424
Local Zone 可用性 2424
先決條件 2426
加入您的位址範圍 2434
使用您的位址範圍 2442
彈性 IP 位址 2443
彈性 IP 位址定價 2444
彈性 IP 位址基本概念 2444

彈性 IP 位址配額 2445
建立彈性 IP 位址的關聯 2445
轉移彈性 IP 位址 2450
釋出彈性 IP 位址 2455
針對電子郵件應用程式使用反向 DNS 2456
網路介面 2460
網路介面概念 2462
網路卡 2463
每個執行個體類型每網路介面的 IP 位址 2467
建立網路介面 2470
網路介面連接 2475
管理 IP 位址 2478
修改網路介面屬性 2481
多個網路介面 2485
申請者受管的網路介面 2488
字首委派 2490
刪除網路介面 2500
網路頻寬 2501
可用執行個體頻寬 2502
頻寬加權 2504
監控執行個體頻寬 2542
增強型網路 2543
彈性網路介面卡 (ENA) 2544
ENA Express 2620
Intel 82599 VF 2659
監控網路效能 2670
改善 Linux 的網路延遲 2680
Nitro 效能考量因素 2683
在 Windows 上最佳化網路效能 2690
Elastic Fabric Adapter 2691
EFA 基本概念 2692
支援的介面和程式庫 2694
支援的執行個體類型 2695
支援的作業系統 2711
EFA 限制 2712
EFA 定價 2713

開始使用 EFA 和 MPI 2713
開始使用 EFA 和 NCCL 2730
開始使用 EFA 和 NIXL 2756
最大化網路頻寬 2776
在容器中安裝 EFA 2790
建立和連接 EFA 2792
分離和刪除 EFA 2796
監控 EFA 2797
驗證 EFA 安裝程式 2804
版本備註 2818
EC2 拓撲 2840
運作方式 2841
先決條件 2847
範例 2849
置放群組 2864
置放策略 2865
建立置放群組 2870
變更執行個體的放置位置 2873
刪除置放群組 2874
共用放置群組 2875
上的置放群組 AWS Outposts 2877
網路 MTU 2878
巨型訊框 (9001 MTU) 2879
路徑 MTU 探索 2880
為您的執行個體設定 MTU 2880
疑難排解 2886
Virtual Private Cloud (VPC) 2886
您的預設 VPC 2887
非預設 VPC 2887
網際網路存取 2888
共用子網路 2888
僅限 IPv6 的子網路 2889
次要網路 2889
什麼是次要網路？ 2890
重要概念 2890
Architecture 2890

其他考量 2890
開始使用 2891
管理次要網路資源 2893
網路設計最佳實務 2894
疑難排解 2894
配額和限制 2895
安全 2896
資料保護 2897
Amazon EBS 資料安全 2898
靜態加密 2898
傳輸中加密 2899
基礎設施安全 2901
網路隔離 2901
實體主機上的隔離 2901
控制網路流量 2902
恢復能力 2903
法規遵循驗證 2904
Identity and Access Management 2905
身分型政策 2905
API 的政策範例 2916
主控台的範例政策 2958
AWS 受管政策 2970
IAM 角色 2976
更新管理 2987
Windows 執行個體的最佳實務 2987
高階安全最佳做法 2988
更新管理 2989
組態管理 2990
變更管理 2991
Amazon EC2 Windows 執行個體的稽核和責任 2992
金鑰對 2992
建立金鑰對 2994
描述金鑰對 3000
刪除您的金鑰對 3005
在執行個體上新增或取代公有金鑰 3006
驗證指紋 3007

Security groups (安全群組) 3010
概要 3010
建立安全群組 3011
變更執行個體的安全群組 3013
刪除安全群組 3019
連線追蹤 3020
不同使用案例的安全群組規則 3025
NitroTPM 3031
要求 3031
啟用 NitroTPM 的 Linux AMI 3033
驗證已為 NitroTPM 啟用 AMI 3034
使用 NitroTPM 來啟用或停止 3036
驗證已為 NitroTPM 啟用執行個體 3037
擷取公有背書金鑰 3038
EC2 執行個體驗證 3040
與 整合 AWS KMS 3041
驗證隔離的運算環境 3042
AWS 共同責任模型 3042
可驗證的 AMI 3042
AWS KMS 準備證明 3050
獲取 NitroTPM 驗證文件 3051
與 整合 AWS KMS 3058
將資料與您自己的運算子隔離 3059
Windows 執行個體的 Credential Guard 3060
先決條件 3060
啟動支援的執行個體 3061
停用記憶體完整性 3062
開啟 Credential Guard 3063
驗證 Credential Guard 正在執行 3065
AWS PrivateLink 3065
建立介面 VPC 端點 3066
建立端點政策 3066
儲存 3068
AWS 儲存體定價 3069
Amazon EBS 3069
EBS 磁碟區限制 3070

EBS 卡 3073
Amazon EC2 執行個體儲存體 3075
資料持久性 3077
執行個體儲存體磁碟區限制 3078
SSD 執行個體存放區磁碟區 3083
新增執行個體存放區磁碟區 3086
啟用 M1 和 C1 執行個體的交換磁碟區 3093
初始化執行個體儲存體磁碟區 3097
詳細的效能統計資料執行個體儲存體磁碟區 3098
根磁碟區 3102
具有 Amazon EBS 根磁碟區的執行個體 3102
具有執行個體儲存體根磁碟區的執行個體 （僅限 Linux 執行個體） 3104
在執行個體終止後保留根磁碟區 3105
取代根磁碟區 3109
磁碟區的裝置名稱 3119
可用裝置名稱 3120
裝置名稱考量 3122
區塊型儲存設備對應 3123
區塊型儲存設備映射概念 3124
將區塊型儲存設備映射新增至 AMI 3127
新增區塊型儲存設備映射至執行個體 3131
如何為 Windows 執行個體連接和映射磁碟區 3139
將 NVME 磁碟映射至磁碟區 3140
將非 NVME 磁碟映射至磁碟區 3145
預防未完成的寫入 3155
受支援的區塊大小 3156
要求 3156
檢查執行個體支援 3157
設定工作負載 3158
Windows VSS EBS 快照 3160
什麼是 IVS？ 3160
基於 VSS 的 Amazon EBS 快照解決方案運作方式 3161
VSS 的先決條件 3162
建立 VSS 快照 3173
疑難排解 VSS 快照 3182
AWS VSS 解決方案的還原選項 3186

版本歷史記錄 3187
物件儲存、檔案儲存和檔案快取 3191
Amazon S3 3192
Amazon EFS 3196
Amazon FSx 3200
Amazon File Cache 3205
管理資源 3206
為您的資源選擇區域 3206
標記您的資源 3208
主控台步驟 3208
命令列範例 3216
全域檢視 (跨區域) 3219
AWS 全域檢視 3219
標記您的 資源 3223
標籤基本概念 3224
標記您的 資源 3225
標籤限制 3225
標籤與存取管理 3226
標記您的資源以便計費 3226
標籤資源許可 3227
新增或移除標籤。 3230
依標籤篩選資源 3234
使用執行個體中繼資料檢視標籤 3236
Service Quotas 3241
檢視您目前的配額 3242
請求增加 3242
使用通訊埠 25 傳送的電子郵件限制 3243
監控 資源 3244
監控您的執行個體狀態 3245
狀態檢查 3245
應用程式狀態檢查 3254
狀態變更事件 3272
排程事件 3274
使用 CloudWatch 監控您的執行個體 3310
執行個體警示 3311
管理詳細監控 3312

CloudWatch 指標 3315
安裝及設定 CloudWatch 代理程式 3329
指標的統計資料 3333
檢視監控圖表 3342
建立警示 3342
建立警示以停止、終止、重新啟動或復原執行個體 3343
使用 EventBridge 進行自動化 3355
Amazon EC2 事件類型 3355
使用 CloudTrail 記錄 API 呼叫 3356
CloudTrail 中的 Amazon EC2 API 管理事件 3357
Amazon EC2 API 事件範例 3357
稽核使用 EC2 Instance Connect 的連線 3359
監控 .NET 和 SQL Server 應用程式 3360
追蹤您的免費方案用量 3361
2025 年 7 月 15 日之前和之後的免費方案優惠 3361
追蹤 2025 年 7 月 15 日之前建立之帳戶的免費方案使用情況 3363
帳單與用量報告 3365
執行個體 3366
裸機執行個體 3366
專用執行個體 3367
專用執行個體 3367
EBS 最佳化 3367
Capacity Reservations 3368
清查 EC2 執行個體 3368
地理位置 3369
硬體規格 3370
映像 3371
IP 位址與 DNS 主機名稱 3373
安全群組規則 3374
使用者資料 3376
相關資源 3377
疑難排解 3378
執行個體啟動問題 3378
無效裝置名稱 3379
超過執行個體限制 3380
執行個體容量不足 3380

目前並不支援要求的組態。請查看文件以了解支援的組態。 3381
立刻終止執行個體 3381
許可不足 3382
Windows 啟動後立即出現高 CPU 用量 (僅限 Windows 執行個體) 3383
啟動IMDSv1-enabled的執行個體失敗 3384
執行個體停止問題 3385
強制停止執行個體 3386
(選用) 建立替代執行個體 3387
執行個體終止問題 3390
立刻終止執行個體 3390
延遲的執行個體終止 3390
仍顯示已終止的執行個體 3392
錯誤：執行個體可能無法終止。修改其 'disableApiTermination' 執行個體屬性 3392
執行個體自動啟動或終止 3393
無法連線的執行個體 3393
重新啟動執行個體 3393
執行個體主控台輸出 3394
擷取無法連線執行個體的螢幕擷取畫面 3395
Windows 執行個體的常見螢幕擷取畫面 3396
當主機電腦故障時的執行個體復原 3405
執行個體離線且意外重新啟動 3405
Linux 執行個體 SSH 問題 3406
連線問題的常見原因 3406
連線到執行個體時發生錯誤：連線時間逾時 3409
錯誤：無法載入金鑰 ... 預期為：任一私有金鑰 3411
錯誤：伺服器無法辨識使用者金鑰 3412
錯誤：許可遭拒或 [執行個體] 連接埠 22 已關閉連線 3414
錯誤：未受保護的私有金鑰檔案 3416
錯誤：私有金鑰的開頭必須為「-----BEGIN RSA PRIVATE KEY-----」，結尾為「-----END RSA PRIVATE KEY-----」 3417
錯誤：主機金鑰驗證失敗 3418
錯誤：伺服器拒絕了我們的金鑰，或是沒有可用的支援驗證方法 3418
無法對執行個體執行 Ping 動作 3419
錯誤：伺服器意外關閉網路連線 3419
錯誤：EC2 Instance Connect 的主機金鑰驗證失敗 3420
無法使用 EC2 Instance Connect 連線至 Ubuntu 執行個體 3421

我的私有金鑰遺失。如何連線到我的執行個體？ 3422
Linux 執行個體狀態檢查失敗 3428
檢閱狀態檢查資訊 3429
擷取系統日誌 3430
針對 Linux 執行個體，進行系統日誌錯誤的疑難排解 3430
記憶體不足：終止程序 3431
錯誤：mmu_update 失敗 (記憶體管理更新作業失敗) 3432
I/O 錯誤 (區塊型儲存設備故障) 3433
I/O 錯誤：非本機也非遠端磁碟 (故障的分散式區塊型儲存設備) 3435
request_module：失控迴圈 modprobe (在舊版 Linux 上建立舊核心 modprobe 的迴圈) 3436
「嚴重：核心版本太舊」和「fsck：嘗試開啟 /dev 時，找不到此等檔案或目錄」(核心與 AMI 不符) 3437
「嚴重：無法載入 /lib/modules」或「BusyBox」(缺少核心模組) 3438
錯誤：無效的核心 (與 EC2 不相容的核心) 3440
fsck：嘗試開啟時找不到此等檔案或目錄… (找不到檔案系統) 3441
掛載檔案作業系統的一般錯誤 (掛載失敗) 3443
VFS：無法在未知的區塊上掛載根 fs (根檔案系統不符) 3445
錯誤：無法判定根設備的主/次編號… (根檔案系統/裝置不相符) 3446
XENBUS：裝置缺少驅動程式… 3448
…未進行檢查的天數，強制進行檢查 (需要進行檔案系統檢查) 3449
fsck 凍結於結束狀態… (缺少裝置) 3449
GRUB 提示 (grubdom>) 3451
帶出界面 eth0：裝置 eth0 的 MAC 地址和預期的不同，略過。(將 MAC 地址直接寫入程式) 3453
無法載入 SELinux 政策。機器正處於強制執行模式。現在停止中。(SELinux 組態設定不當) 3454
XENBUS：連線到裝置的作業逾時 (Xenbus 逾時) 3456
從錯誤磁碟區啟動 Linux 執行個體 3457
Windows 執行個體 RDP 問題 3458
遠端桌面無法連線到遠端電腦 3459
使用 macOS RDP 用戶端時發生錯誤 3460
RDP 顯示黑色螢幕而非桌面 3460
無法使用非管理員的使用者從遠端登入至執行個體 3461
使用 對遠端桌面問題進行故障診斷 AWS Systems Manager 3461
使用遠端登錄在 EC2 執行個體上啟用遠端桌面 3465
我的私有金鑰遺失。如何連接到我的 Windows 執行個體？ 3466

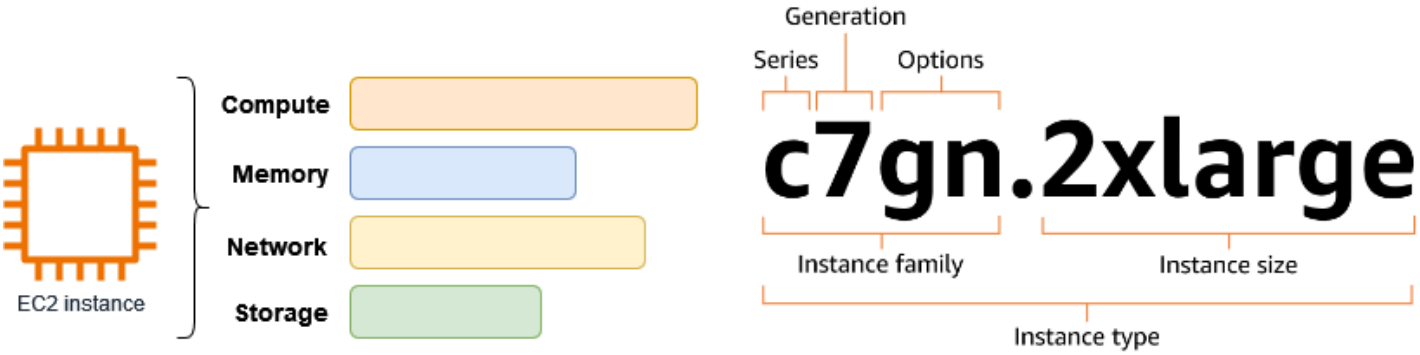
Windows 執行個體啟動問題 3467
「未提供密碼」 3467
「尚未提供密碼」 3468
「無法擷取 Windows 密碼」 3468
等待中繼資料服務 3468
無法啟動 Windows 3472
Windows 不是正版 (0x80070005) 3474
無終端機伺服器的授權伺服器可提供授權 3474
「部分設定是由您的組織所管理」 3475
Windows 執行個體問題 3475
無法將 AWS Systems Manager Sessions Manager 連線至 Windows Server 2025 執行個體 3476
EBS 磁碟區不是在 Windows Server 2016 和 2019 和更新版本上初始化 3476
在目錄服務還原模式 (DSRM) 中開機 EC2 Windows 執行個體 3477
執行個體遺失網路連線能力或排程任務未如預期執行 3480
無法取得主控台輸出 3480
網路不提供 Windows Server 2012 R2 3481
磁碟簽章衝突 3481
透過網路進行 Windows 執行個體核心偵錯 3482
先決條件 3483
步驟 1：在偵錯主機上安裝 Windows 偵錯工具 3484
步驟 2：設定偵錯目標 3484
步驟 3：在偵錯主機上啟動偵錯工作階段 3486
步驟 4：重新啟動偵錯目標 3487
偵錯後清除偵錯設定 3487
限制 3488
其他備註 3488
重設 Windows 管理員密碼 3488
使用 EC2Launch v2 重設密碼 3489
使用 EC2Launch 重設密碼 3494
使用 EC2Config 重設密碼 3499
疑難排解 Sysprep 問題 3504
適用於 Linux 執行個體的 EC2Rescue 3505
安裝 EC2Rescue 3506
執行 EC2Rescue 命令 3510
開發 EC2Rescue 模組 3512

適用於 Windows 執行個體的 EC2Rescue 3518
使用 EC2Rescue GUI 來疑難排解 3519
使用 EC2Rescue CLI 來疑難排解 3525
使用 EC2Rescue 和 Systems Manager 來進行疑難排解 3532
EC2 序列主控台 3536
先決條件 3536
設定對 EC2 序列主控台的存取 3544
連線至 EC2 序列主控台 3554
終止與 EC2 序列主控台的連線 3563
使用 EC2 序列主控台進行執行個體疑難排解 3564
傳送診斷中斷 3572
支援的執行個體類型 3573
先決條件 3573
傳送診斷中斷 3575
文件歷史紀錄 3577
2018 年及以前的歷程記錄 3608
..... mmmdcxxvii

什麼是 Amazon EC2？

Amazon Elastic Compute Cloud (Amazon EC2) 在 Amazon Web Services (AWS) Cloud 中提供可擴展的隨需運算容量。使用 Amazon EC2 可降低硬體成本，讓您以更快的速度開發和部署應用程式。您可使用 Amazon EC2 按需要啟動任意數量的虛擬伺服器，設定安全性和聯網功能以及管理儲存。您可以新增容量 (擴充) 來處理需要大量運算的工作，例如每月或每年的例行程序，或是網站流量暴增。當使用量降低時，您可以再次減少容量 (縮減)。

EC2 執行個體是 AWS 雲端中的虛擬伺服器。啟動 EC2 執行個體時，您指定的執行個體類型將決定執行個體適用的硬體。每個執行個體類型皆會提供不同平衡的運算、記憶體、網路和儲存體資源。如需詳細資訊，請參閱 Amazon EC2 執行個體類型指南。



Amazon EC2 的功能

Amazon EC2 提供以下進階功能：

執行個體

虛擬伺服器。

Amazon Machine Images (AMI)

供執行個體使用的預先設定範本，在其中封裝伺服器所需的元件 (包括作業系統和其他軟體)。

執行個體類型

適用執行個體的各種 CPU、記憶體、儲存、連網容量和圖形硬體組態。

Amazon EBS 磁碟區

資料的持久性儲存磁碟區請使用 Amazon Elastic Block Store (Amazon EBS)。

執行個體儲存體磁碟區

儲存磁碟區可供您停止、休眠或終止執行個體時所刪除的暫時性資料使用。

金鑰對

執行個體的安全登入資訊。 會 AWS 存放公有金鑰，而您將私有金鑰存放在安全的地方。

Security groups (安全群組)

虛擬防火牆可讓您指定存取執行個體的協定、連接埠和來源 IP 範圍，以及執行個體可連線的目的地 IP 範圍。

Amazon EC2 支援處理、儲存、傳輸商家或服務供應商的信用卡資料，並且已驗證符合支付卡產業 (PCI) 資料安全標準 (DSS)。如需 PCI DSS 的詳細資訊，包括如何請求 AWS PCI 合規套件的副本，請參閱 PCI DSS 第 1 級。

相關服務

搭配 Amazon EC2 一起使用的服務

您可以使用其他 AWS 服務 搭配您使用 Amazon EC2 部署的執行個體。

Amazon EC2 Auto Scaling

協助確保您有正確的 Amazon EC2 執行個體數量可應付應用程式的負載。

AWS Backup

自動備份 Amazon EC2 執行個體和連接至該執行個體的 Amazon EBS 磁碟區。

Amazon CloudWatch

監控執行個體和 Amazon EBS 磁碟區。

Elastic Load Balancing

自動將傳入應用程式流量分配至多個執行個體。

Amazon GuardDuty

偵測 EC2 執行個體可能未授權或惡意的使用情況。

EC2 Image Builder

自動建立、管理及部署自訂、安全且最新的伺服器映像。

AWS Launch Wizard

為第三方應用程式調整資源的大小、設定和部署 AWS 資源，而不必手動識別和佈建個別 AWS 資源。

AWS Systems Manager

使用此安全端對端管理解決方案，在 EC2 執行個體上大規模地執行操作。

其他運算服務

您可以使用其他 AWS 運算服務啟動執行個體，而不是使用 Amazon EC2。

Amazon Lightsail

使用 Amazon Lightsail 建置網站或 web 應用程式，其為雲端平台，會以低廉且可預測的每月價格來提供快速部署專案所需之資源。若要比較 Amazon EC2 和 Lightsail，請參閱 Amazon Lightsail 或 Amazon EC2。

Amazon Elastic Container Service (Amazon ECS)

在 EC2 執行個體的叢集上部署、管理和擴展容器化應用程式。如需詳細資訊，請參閱選擇 AWS 容器服務。

Amazon Elastic Kubernetes Service (Amazon EKS)

在 AWS上執行您的 Kubernetes 應用程式。如需詳細資訊，請參閱選擇 AWS 容器服務。

存取 Amazon EC2

您可以使用下列界面來建立和管理 Amazon EC2 執行個體：

Amazon EC2 主控台

用於建立和管理 Amazon EC2 執行個體和資源的簡易 Web 界面。如果您已註冊 AWS 帳戶，您可以登入 AWS 管理主控台 並從主控台首頁選取 EC2 來存取 Amazon EC2 主控台。 EC2

AWS Command Line Interface

可讓您使用命令列 shell 中的命令與 AWS 服務互動。Windows、Mac 和 Linux 系統皆提供支援。如需有關 AWS CLI 的詳細資訊，請參閱 「AWS Command Line Interface 使用者指南」。您可以在 AWS CLI 命令參考 中找到 Amazon EC2 命令。

CloudFormation

Amazon EC2 支援使用 CloudFormation建立資源。您可以建立 JSON 或 YAML 格式的範本，描述您的 AWS 資源，並為您 CloudFormation 佈建和設定這些資源。您可以重複使用 CloudFormation 範本來多次佈建相同的資源，無論是在相同的區域和帳戶，還是在多個區域和帳戶中。如需 Amazon EC2 適用的支援資源類型和屬性的詳細資訊，請參閱 「AWS CloudFormation 使用者指南」 中的 EC2 資源類型參考。

AWS SDKs

如果您偏好使用特定語言 APIs 來建置應用程式，而不是透過 HTTP 或 HTTPS 提交請求， AWS 會提供程式庫、範本程式碼、教學課程和其他資源給軟體開發人員。這些程式庫提供的基本功能可自動化任務，例如密碼編譯簽署要求、重試要求，以及處理錯誤回應，讓您更容易入門。如需詳細資訊，請參閱 在 AWS上建置的工具。

AWS Tools for PowerShell

一組建立在 適用於 .NET 的 SDK 公開的功能之上的 PowerShell 模組。您可以透過 Tools for PowerShell 從 PowerShell 命令列撰寫 AWS 資源的操作指令碼。若要開始使用，請參閱「 使用者指南AWS Tools for PowerShell」。您可以在 AWS Tools for PowerShell Cmdlet 參考 中找到適用於 Amazon EC2 的 cmdlet。

查詢 API

Amazon EC2 提供查詢 API。這些 HTTP 或 HTTPS 請求，需要使用 HTTP 動詞 GET 或 POST，以及名為 `Action` 的查詢參數。如需適用於 Amazon EC2 的 API 動作詳細資訊，請參閱 Amazon EC2 API Reference中的動作。

Amazon EC2定價

Amazon EC2 提供下列定價選項：

免費方案

您可以免費開始使用 Amazon EC2。若要探索免費方案選項，請參閱 AWS 免費方案。

隨需執行個體

您只需要按秒支付使用執行個體的費用 (最少 60 秒)，無需長期購買或預付款。

Savings Plans

您可以透過對一致的使用量設定綁約來降低 Amazon EC2 成本，每小時以美金計價，為期一年或三年。

Reserved Instances

您可以透過設定綁約至特定的執行個體設定來降低您 Amazon EC2 的成本，包含執行個體類型和區域，為期一年或三年。

Spot Instances

請求未使用的 EC2 執行個體，可大幅降低您的 Amazon EC2 成本。

專用執行個體

使用完全供您使用的實體 EC2 伺服器 (隨需或 Savings Plan 的一部分) 來降低成本。您可以使用現有繫結伺服器的軟體授權，並取得符合合規要求的協助。

隨需容量保留

在特定可用區域中為 EC2 執行個體預留任何期限的運算容量。

以秒計費

從帳單中移除未使用分鐘和秒數的成本。

如需 Amazon EC2 的完整收費與定價清單，以及有關購買模型的詳細資訊，請參閱 Amazon EC2 定價。

估算、計費和成本最佳化

若要為您的 AWS 使用案例建立預估值，請使用 AWS 定價計算工具。

若要預估將 Microsoft 工作負載轉換為使用部署在 上之開放原始碼和雲端原生服務的現代架構的成本 AWS，請使用 AWS Microsoft 工作負載的現代化計算器。

若要查看您的帳單，請前往 AWS 帳單與成本管理 主控台中的帳單與成本管理儀表板。您的帳單內含用量報告的連結，可提供帳單的詳細資訊。若要進一步了解 AWS 帳戶帳單，請參閱AWS 帳單與成本管理使用者指南。

如果您對 AWS 帳單、帳戶和事件有任何疑問，請聯絡 AWS Support。

若要計算範本佈建環境的成本，請參閱雲端成本中心。在計算佈建環境的成本時，請記住要包括附帶成本，如 EBS 磁碟區的快照儲存。

您可以使用 最佳化 AWS 環境的成本、安全性和效能AWS Trusted Advisor。

您可以使用 AWS Cost Explorer 來分析 EC2 執行個體的成本和用量。您可以檢視多達前13個月的資料，以及預測未來12月的可能花費。如需詳細資訊，請參閱《AWS Cost Management 使用者指南》中的使用 AWS Cost Explorer分析成本和使用情況。

Resources

- Amazon EC2 功能
- AWS re：Post
- AWS 技能建置器
- AWS 支援
- 實際操作教學課程
- Web 託管
- 上的 Windows AWS

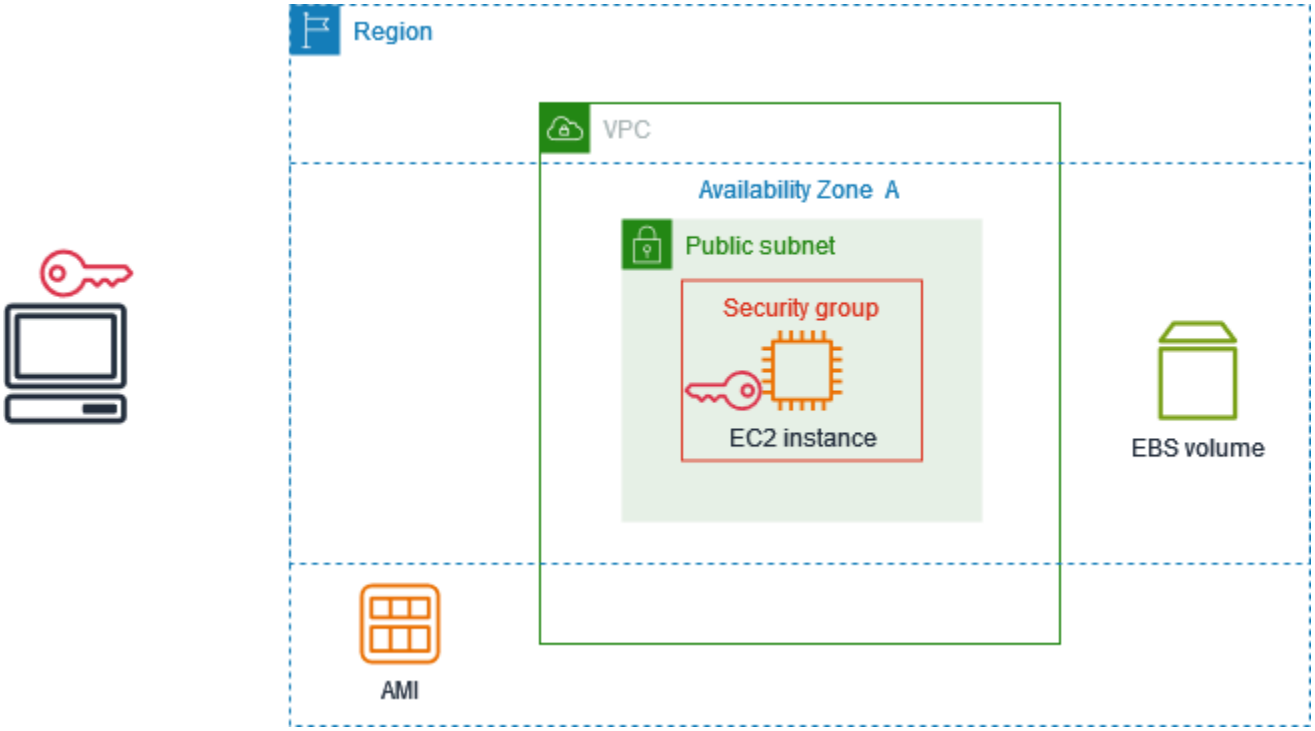
開始使用 Amazon EC2

使用此教學課程開始使用 Amazon Elastic Compute Cloud (Amazon EC2) 。您將了解如何啟動及連線到 EC2 執行個體。執行個體是 AWS 雲端中的虛擬伺服器。使用 Amazon EC2，依您的執行個體安裝和設定作業系統與應用程式。

概觀

下圖顯示您將在本教學課程中使用的金鑰元件：

- 映像 – 包含在執行個體上所執行軟體的範本，例如作業系統。
- 金鑰對 – 一組安全性登入資料，您可以在連線至執行個體時用來證明身分。公有金鑰會在執行個體上，而私有金鑰則會在電腦上。
- 網路 – 虛擬私有雲端 (VPC) 是您的 AWS 帳戶所專用的虛擬網路。為了協助您快速開始使用，您的帳戶在每個 中隨附預設 VPC AWS 區域，而每個預設 VPC 在每個可用區域中都有預設子網路。
- 安全群組 – 做為虛擬防火牆以控制傳入及傳出流量。
- EBS 磁碟區 – 映像需要根磁碟區。您可以選擇性地新增資料磁碟區。



此教學課程的成本

建立 時 AWS 帳戶，您可以使用 免費開始使用 Amazon EC2AWS 免費方案。

如果您在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，但它還不到 12 個月，而且您尚未超過 Amazon EC2 的免費方案優惠，則完成本教學課程無需支付任何費用，因為本教學課程可協助您選取免費方案優惠範圍內的選項。否則，從您啟動執行執行個體 (即使執行個體為閒置狀態) 到終止執行個體為止，會產生基本的 Amazon EC2 使用費用。

如果您在 2025 年 7 月 15 日當天 AWS 帳戶 或之後建立 ，但它還不到 6 個月，而且您尚未用完所有抵用金，則完成本教學課程無需支付任何費用，因為本教學課程可協助您選取 免費方案優惠範圍內的選項。

如需如何判斷您是否符合免費方案資格的資訊，請參閱 the section called “追蹤您的免費方案用量”。

任務

- 步驟 1：啟動執行個體
- 步驟 2：連線到您的執行個體
- 步驟 3：清理您的執行個體
- 後續步驟

步驟 1：啟動執行個體

您可以使用 啟動 EC2 執行個體 AWS 管理主控台 ，如下列程序所述。本教學旨在協助您快速啟動您的第一個執行個體，所以不涵蓋全部的可能選項。

啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面頂端的導覽列中， AWS 區域 會顯示目前的 - 例如俄亥俄。您可以使用選取的區域，或選擇性地選取距離您最近的區域。
3. 在 EC2 主控台儀表板的啟動執行個體窗格中，選擇啟動執行個體。
4. 在 Name and tags (名稱和標籤) 下的 Name (名稱) 欄位中，輸入執行個體的描述性名稱。
5. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，執行下列動作：

    a. 選擇快速入門，然後選擇執行個體的作業系統 (OS)。針對您的第一個 Linux 執行個體，建議您選擇 Amazon Linux。

    b. 從 Amazon Machine Image (AMI) 中，選取標記為符合免費方案資格的 AMI。

6. 在執行個體類型下，針對執行個體類型，選取標記為符合免費方案資格的執行個體類型。
7. 在金鑰對 (登入) 下：針對金鑰對名稱，選擇現有的金鑰對，或選擇建立新的金鑰對以建立新的金鑰對。

> **Warning**
>
> 如果您選擇不使用金鑰對繼續 (不建議)，則無法使用本教學中所述的方法連線到執行個體。

8. 在網路設定下，請注意，您的預設 VPC 已選取、在為您選擇的可用區域中使用預設子網路的選項，以及具有規則的安全群組，允許從任何位置連線至您的執行個體 (`0.0.0.0/0`)。

> **Warning**
>
> 若您指定 `0.0.0.0/0`，您便會啟用來自世界任何 IP 地址的流量。針對 SSH 和 RDP 通訊協定，您可能會將在測試環境中短暫進行此作業視為沒有問題，但用在生產環境則不安全。在生產環境中，請務必僅授權來自適當個別 IP 地址或地址範圍的存取。

針對您的第一個執行個體，建議您使用預設設定。否則，您可以更新網路設定，如下所示：

- (選用) 若要使用特定的預設子網路，請選擇編輯，然後選擇子網路。
- (選用) 若要使用不同的 VPC，請選擇編輯，然後選擇現有的 VPC。如果 VPC 未設定為公有網際網路存取，您將無法連線至執行個體。
- (選用) 若要限制特定網路的傳入連線流量，請選擇自訂而非任何位置，然後輸入網路的 CIDR 區塊。
- (選用) 若要使用不同的安全群組，請選擇選取現有安全群組，然後選擇現有的安全群組。如果安全群組沒有任何規則允許來自您網路的連線流量，您將無法連線至執行個體。對於 Linux 執行個體，您必須允許 SSH 流量。對於 Windows 執行個體，您必須允許 RDP 流量。

9. 在設定儲存體下，請注意已設定根磁碟區，但沒有資料磁碟區。這足以用於測試目的。
10. 檢閱 Summary (摘要) 面板中執行個體組態的摘要，並在準備就緒時選擇 Launch instance (啟動執行個體)。
11. 如果啟動成功，請從成功通知中選擇執行個體的 ID，以開啟執行個體頁面，並監控啟動的狀態。
12. 選取執行個體的核取方塊。初始執行個體狀態為 `pending`。在執行個體啟動後，它的生命週期狀態會變為 `running`。選擇狀態和警示索引標籤。執行個體通過其狀態檢查後，隨即就緒可接收連線請求。

步驟 2：連線到您的執行個體

您使用的程序會依執行個體的作業系統而定。如果您無法連線至您的執行個體，請參閱對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解取得協助。

Linux 執行個體

您可以使用任何 SSH 用戶端連線至 Linux 執行個體。如果您的電腦正執行 Windows，請開啟終端機並執行 `ssh` 命令，以確認您已安裝 SSH 用戶端。如果找不到命令，請安裝 OpenSSH for Windows。

使用 SSH 連線到您的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 Connect (連線)。
4. 在連線至執行個體頁面，選擇 SSH 用戶端索引標籤。
5. (選用) 如果您在啟動執行個體時建立金鑰對，並將私有金鑰 (.pem 檔案) 下載至執行 Linux 或 macOS 的電腦，請執行範例 chmod 命令來設定私有金鑰的許可。
6. 複製範例 SSH 命令。在以下範例中，*key-pair-name*.pem 是私有金鑰檔案的名稱，*ec2-user* 是與映像相關聯的使用者名稱，而 @ 符號後面的字串是執行個體的公用 DNS 名稱。

   ```
   ssh -i key-pair-name.pem ec2-user@ec2-198-51-100-1.us-east-2.compute.amazonaws.com
   ```

7. 在電腦的終端機視窗中，執行您在上一個步驟中儲存的 ssh 命令。如果私有金鑰檔案不在目前的目錄中，則您必須在此命令中指定金鑰檔案的完整路徑。

   以下是回應範例：

   ```
   The authenticity of host 'ec2-198-51-100-1.us-east-2.compute.amazonaws.com
    (198-51-100-1)' can't be established.
   ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
   Are you sure you want to continue connecting (yes/no)?
   ```

8. (選用) 當您第一次啟動執行個體時，請確認安全提醒中的指紋符合主控台輸出中所包含的執行個體指紋。若要取得主控台輸出，請選擇動作、監控和疑難排解、取得系統日誌。如果這些指紋不符，可能有人嘗試發動中間人攻擊。如果它們符合，請繼續下一個步驟。
9. 輸入 **yes**。

   以下是回應範例：

```
Warning: Permanently added 'ec2-198-51-100-1.us-
east-2.compute.amazonaws.com' (ECDSA) to the list of known hosts.
```

Windows 執行個體

若要使用 RDP 連線到 Windows 執行個體，您必須擷取初始管理員密碼，然後在連線到執行個體時輸入此密碼。在執行個體啟動後，需要幾分鐘的時間處理才能提供密碼。您的帳戶必須擁有呼叫 GetPasswordData 動作的權限。如需詳細資訊，請參閱控制 Amazon EC2 API 存取的政策範例。

管理員帳戶的預設使用者名稱取決於 AMI 包含的作業系統 (OS) 的語言。若要確定正確的使用者名稱，請識別作業系統的語言，然後選擇對應的使用者名稱。例如，對於英文作業系統，使用者名稱為 Administrator，對於法文作業系統，使用者名稱為 Administrateur，對於葡萄牙文作業系統，使用者名稱為 Administrador。如果作業系統的語言版本沒有相同語言的使用者名稱，請選擇使用者名稱 Administrator (Other)。如需詳細資訊，請參閱 Microsoft 網站中的管理員帳戶在 Windows 中的當地語系化名稱。

擷取初始管理員密碼

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 Connect (連線)。
4. 在連線至執行個體頁面，選擇 RDP 用戶端索引標籤。
5. 對於使用者名稱，選擇管理員帳戶的預設使用者名稱。您選擇的使用者名稱必須與您用於啟動執行個體的 AMI 所包含的作業系統 (OS) 的語言相符。若無與作業系統相同語言的使用者名稱，請選擇管理員 (其他)。
6. 選擇取得密碼。
7. 在取得 Windows 密碼頁面，執行下列動作：

    a. 選擇上傳私有金鑰檔案，然後導覽至您在啟動執行個體時指定的私有金鑰 (.pem) 檔案。選取檔案並選取 Open (開啟)，將檔案的完整內容複製至此視窗。

    b. 選擇解密密碼。取得 Windows 密碼頁面隨即關閉，執行個體的預設管理員密碼會顯示在密碼下，取代先前顯示的取得密碼連結。

    c. 複製密碼並將其保存在安全處。必須使用此密碼來連接至執行個體。

以下程序使用適用於 Windows 的遠端桌面連線用戶端 (MSTSC)。如果您使用其他 RDP 用戶端，請下載 RDP 檔案，然後查看 RDP 用戶端的文件，了解建立 RDP 連線的步驟。

使用 RDP 用戶端連線至 Windows 執行個體。

1. 在連線至執行個體頁面，選擇下載遠端桌面檔案。檔案下載完成後，請選擇取消以返回執行個體頁面。RDP 檔案已下載至您的 Downloads 資料夾。
2. 執行 `mstsc.exe` 以開啟 RDP 用戶端。
3. 展開顯示選項，選擇開啟，然後從 Downloads 資料夾中選取 .rdp 檔案。
4. 依預設，電腦是執行個體的公有 IPv4 DNS 名稱，使用者名稱是管理員帳戶。若要改用 IPv6 連線至執行個體，請將執行個體的公有 IPv4 DNS 名稱替換為其 IPv6 地址。檢視預設設定並視需要進行變更。
5. 選擇連線。如果您收到遠端連線發布者不明的警告，請選擇連線以繼續。
6. 輸入先前儲存的密碼，然後選擇確定。
7. 由於自我簽署憑證的性質，您可能會收到安全憑證無法驗證的警告。執行以下任意一項：
   - 如果您信任憑證，請選擇是，以連線至您的執行個體。
   - [Windows] 在繼續之前，請將憑證指紋與系統日誌的值進行比較，以確認遠端電腦身分。選擇檢視憑證，然後從詳細資料索引標籤選擇指紋。將此值與動作、監控和疑難排解、取得系統日誌中的 `RDPCERTIFICATE-THUMBPRINT` 值進行比較。
   - [Mac OS X] 在繼續之前，請將憑證指紋與系統日誌的值進行比較，以確認遠端電腦身分。選擇顯示憑證、展開詳細資訊，然後選擇 SHA1 指紋。將此值與動作、監控和疑難排解、取得系統日誌中的 `RDPCERTIFICATE-THUMBPRINT` 值進行比較。
8. 如果 RDP 連線成功，RDP 用戶端會顯示 Windows 登入畫面，然後顯示 Windows 桌面。如果您收到錯誤訊息，請參閱 the section called “遠端桌面無法連線到遠端電腦”。完成 RDP 連線後，即可關閉 RDP 用戶端。

步驟 3：清理您的執行個體

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。

在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

在您完成為本教學建立的執行個體後，您應該終止此執行個體完成清理。如果想在清理前使用此執行個體執行更多作業，請參閱後續步驟。

一旦執行個體狀態變更為 `shutting down` 或 `terminated`，您將停止產生該執行個體的費用，或計入您免費方案限制的使用量。若要保留執行個體以待後用，但不想產生費用或計入您免費方案限制的使用量，您可立即停止執行個體，稍後再啟動執行個體。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。

終止您的執行個體

1. 在導覽窗格中，選擇 Instances (執行個體)。在執行個體清單中，選取執行個體。
2. 選擇執行個體狀態、終止 (刪除) 執行個體。
3. 出現確認提示時，請選擇終止 (刪除)。

   Amazon EC2 關閉並終止您的執行個體。在您的執行個體終止後，它仍會短暫顯示於主控台，然後項目才會自動刪除。您無法自行從主控台顯示畫面中移除已終止的執行個體。

後續步驟

在您啟動執行個體後，建議您探索下列後續步驟：

- 使用入門教學課程探索 Amazon EC2 核心概念。如需詳細資訊，請參閱啟動 EC2 執行個體的教學課程。
- 了解如何使用主控台追蹤 Amazon EC2 免費方案用量。如需詳細資訊，請參閱the section called “追蹤您的免費方案用量”。
- 設定 CloudWatch 警示當使用超出免費方案時通知您 (適用於 2025 年 7 月 15 日之前建立的帳戶)。如需詳細資訊，請參閱「AWS Billing 使用者指南」中的追蹤 AWS 免費方案 使用量。
- 新增 EBS 磁碟區。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的建立 Amazon EBS 磁碟區。
- 了解如何使用 Run 命令從遠端管理您的 EC2 執行個體。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 AWS Systems Manager 執行命令。
- 了解執行個體購買選項。如需詳細資訊，請參閱Amazon EC2 帳單與購買選項。
- 取得有關執行個體類型的建議。如需詳細資訊，請參閱從 EC2 執行個體類型搜尋工具取得建議。

Amazon EC2 的最佳實務

為了確保 Amazon EC2 獲得最大利益，我們建議您執行以下最佳實務。

安全

- 盡可能使用身分提供者和 IAM 角色的聯合身分來管理對 AWS 資源和 API 的存取。 APIs 如需詳細資訊，請參閱《IAM 使用者指南》中的身分提供者和聯合。
- 為您的安全群組實作最嚴苛的規則。
- 定期修補、更新和保護您執行個體上的作業系統與應用程式。如需詳細資訊，請參閱更新管理。有關 Windows 作業系統的指南，請參閱 Windows 執行個體的安全性最佳做法。
- 使用 Amazon Inspector 自動探索和掃描 Amazon EC2 執行個體的軟體漏洞和意外網路暴露。如需更多資訊，請參閱 Amazon Inspector 使用者指南。
- 使用 AWS Security Hub CSPM 控制項，根據安全最佳實務和安全標準來監控 Amazon EC2 資源。如需使用 Security Hub CSPM 的詳細資訊，請參閱AWS Security Hub CSPM 《 使用者指南》中的 Amazon Elastic Compute Cloud 控制項。

儲存

- 了解根磁碟區類型對資料持久性、備份和復原的意涵。如需詳細資訊，請參閱根磁碟區類型。
- 為作業系統和資料使用不同的 Amazon EBS 磁碟區。確保執行個體終止後能持續儲存您資料的磁碟區。如需詳細資訊，請參閱 在執行個體終止時保留資料。
- 使用您執行個體可用的執行個體儲存體存放暫存資料。請記住，當您讓您的執行個體停止、休眠或終止時，會刪除存放在執行個體儲存體的資料。如果您將執行個體儲存體用為資料庫儲存體，請確保您的叢集有能確保容錯能力的複寫因素。
- 加密 EBS 磁碟區和快照。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 加密。

資源管理

- 使用執行個體中繼資料和自訂資源標籤追踪並找出您的 AWS 資源。如需詳細資訊，請參閱使用執行個體中繼資料管理您的 EC2 執行個體和標記您的 Amazon EC2 資源。
- 檢視您目前的 Amazon EC2 限制。計劃在您需要的時候，先行請求提高限制。如需詳細資訊，請參閱Amazon EC2 服務配額。

- 使用 AWS Trusted Advisor 檢查您的 AWS 環境，然後在有機會節省成本、改善系統可用性和效能，或協助彌補安全漏洞時提出建議。如需詳細資訊，請參閱「AWS 支援 使用者指南」中的 AWS Trusted Advisor。

備份與復原

- 使用 Amazon EBS 快照定期備份您的 EBS 磁碟區，並從您的執行個體建立 Amazon Machine Image (AMI) 將組態儲存為範本，以啟動未來的執行個體。如需協助達成此使用案例之 AWS 服務的詳細資訊，請參閱 AWS Backup和 Amazon Data Lifecycle Manager。
- 跨多個可用區域部署您應用程式的關鍵元件，並適當複寫您的資料。
- 設計您的應用程式在執行個體重新啟動時處理動態 IP 定址。如需詳細資訊，請參閱 Amazon EC2 執行個體 IP 定址。
- 監控與回應事件。如需詳細資訊，請參閱 監控 Amazon EC2 資源。
- 請確定您已準備好可處理容錯移轉。如需基本解決方案，您可將網路界面或彈性 IP 地址手動連接到替代執行個體。如需詳細資訊，請參閱 彈性網路介面。如需自動化解決方案，您可使用 Amazon EC2 Auto Scaling。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南。
- 定期測試執行個體和 Amazon EBS 磁碟區的復原程序，以確保資料和各項服務成功還原。

聯網

- 針對 IPv4 和 IPv6，將應用程式的存留時間 (TTL) 值設定為 255。如果您使用較小的值，則存在 TTL 會在應用程式流量傳輸期間過期的風險，從而導致執行個體的可存取性問題。

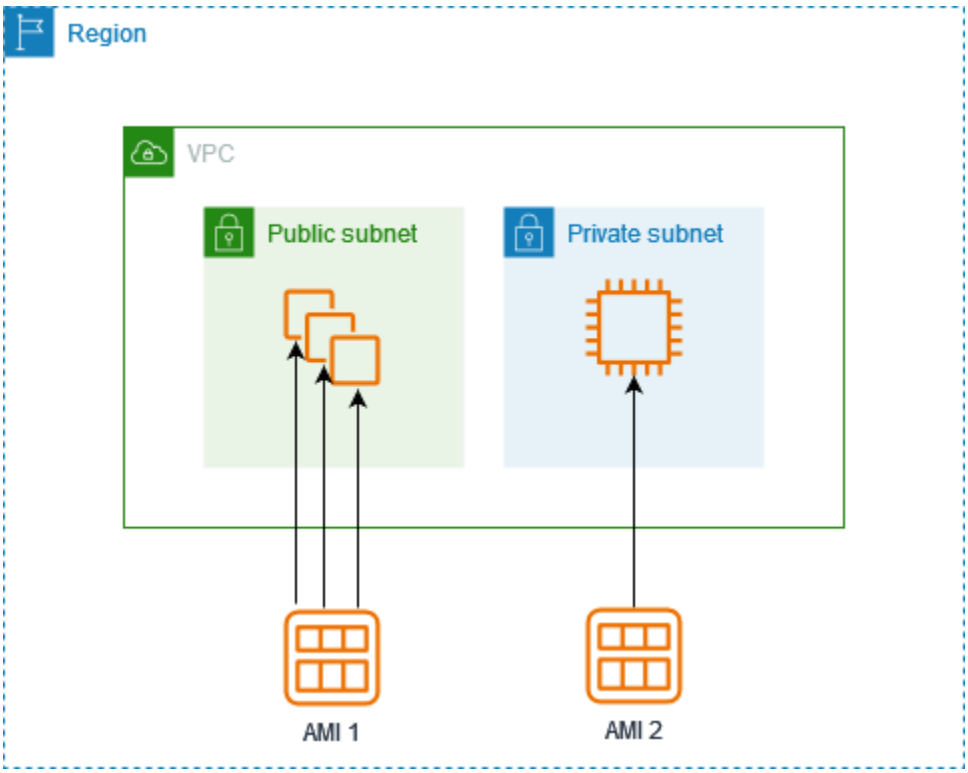
Amazon EC2 的 Amazon Machine Images

Amazon Machine Image (AMI) 是一種映像，提供設定並啟動 Amazon EC2 執行個體所需的軟體。每個 AMI 也包含區塊型儲存設備映射，用於指定要附加到您啟動的執行個體的區塊型儲存設備。在您啟動執行個體時，必須指定 AMI。AMI 必須與您為執行個體選擇的執行個體類型相容。您可以使用 提供的 AMI AWS、公有 AMI、其他人與您共用的 AMI，或是您從 購買的 AMI AWS Marketplace。

AMI 視以下內容而有所不同：

- 區域
- 作業系統
- 處理器架構
- 根磁碟區類型
- 虛擬化類型

當您需要多個具有相同組態的執行個體時，可以從單一 AMI 啟動多個執行個體。當您需要具有不同組態的執行個體時，可以使用不同的 AMI 啟動執行個體，如下圖所示。



您可從 Amazon EC2 執行個體建立 AMI，然後使用它來啟動具有相同配置的執行個體。您可以將 AMI 複製到另一個 AWS 區域，然後使用它在該區域中啟動執行個體。您也可與其他帳戶共用您建立的 AMI，以便其可使用相同配置來啟動執行個體。您可以使用 來銷售您的 AMI AWS Marketplace。

目錄

- Amazon EC2 的 AMI 類型和特性
- 尋找符合 EC2 執行個體需求的 AMI
- AWS Marketplace 適用於 Amazon EC2 執行個體的 中的付費 AMIs
- Amazon EC2 AMI 生命週期
- 使用 Amazon EC2 開機模式執行個體啟動行為
- 對 EBS 後端 AMI 使用加密
- 了解 Amazon EC2 的共用 AMI 使用情況
- 使用 Amazon EventBridge 監控 AMI 事件
- 了解 AMI 帳單資訊
- Amazon EC2 的 AMI 配額

Amazon EC2 的 AMI 類型和特性

當您啟動執行個體時，您選擇的 AMI 必須與您選擇的執行個體類型相容。您可以根據下列特性來選取要使用的 AMI：

- 區域
- 作業系統
- 處理器架構
- 啟動許可
- 根磁碟區類型
- 虛擬化類型

啟動許可

啟動許可確定誰可使用 AMI 來啟動執行個體。您可將啟動許可視為共用 AMI – 若您授予啟動許可，則將與其他使用者共用 AMI。僅 AMI 的擁有者可藉由指定啟動許可來確定其可用性。啟動許可分為以下類別。

啟動許可	描述
public	擁有者會將啟動許可授予所有 AWS 帳戶。
explicit	擁有者會將啟動許可授予特定 AWS 帳戶、組織或組織單位 (OUs)。
implicit	擁有者具有 AMI 的絕對啟動許可。

Amazon 和 Amazon EC2 社群可大量選取公用 AMI。如需詳細資訊，請參閱 了解 Amazon EC2 的共用 AMI 使用情況。開發人員可以收取其 AMI 的費用。如需詳細資訊，請參閱AWS Marketplace 適用於 Amazon EC2 執行個體的 中的付費 AMIs。

根磁碟區類型

所有 AMI 都會分類為 Amazon EBS 支援或 Amazon S3 支援。

- Amazon EBS 支援的 AMI – 從 AMI 啟動之執行個體的根磁碟區是從 Amazon EBS 快照建立的 Amazon Elastic Block Store (Amazon EBS) 磁碟區。同時支援 Linux 和 Windows AMI。
- Amazon S3 支援的 AMI –從 AMI 所啟動之執行個體的根磁碟區是從 Amazon S3 中存放的範本建立的執行個體儲存體磁碟區。僅支援 Linux AMI。Windows AMI 不支援根磁碟區的執行個體儲存體。

如需詳細資訊，請參閱Amazon EC2 執行個體的根磁碟區。

> **Note**
>
> Amazon S3 支援的 AMI 被認為生命週期結束，且不建議用於新用途。其僅在以下舊版執行個體類型上受支援：C1、C3、D2、I2、M1、M2、M3、R3 與 X1。

下表摘要說明使用兩種類型之 AMI 時的重要差異。

特性	Amazon EBS 後端 AMI	Amazon S3 支援的 AMI
根磁碟區	EBS 磁碟區	執行個體儲存體磁碟區
執行個體的開機時間	通常少於 1 分鐘	通常少於 5 分鐘

特性	Amazon EBS 後端 AMI	Amazon S3 支援的 AMI
資料持久性	根據預設，該執行個體終止時會刪除根磁碟區。* 任何其他 EBS 磁碟區上的資料在執行個體終止之後，預設為會持續留存。	只有在執行個體的生命週期，才會持續保存執行個體儲存體磁碟區上的資料。
停止狀態	可以處於停止狀態。即使執行個體已停止且未執行，根磁碟區仍會保留在 Amazon EBS。	無法進入停止狀態；執行個體正在執行或已終止。
修改	在執行個體停止時，可以變更執行個體類型、核心、RAM 磁碟和使用者資料。	在執行個體的生命週期，執行個體屬性是固定的。
費用	會向您收取執行個體使用費、EBS 磁碟區使用費，以及將 AMI 存放為 EBS 快照的費用。	會向您收取執行個體使用費，以及在 Amazon S3 中存放 AMI 的費用。
AMI 建立/綁定	使用單一命令/呼叫	需要安裝和使用 AMI 工具

* 根據預設，EBS 根磁碟區會將 `DeleteOnTermination` 標記設定為 `true`。如需如何變更此標記，以便在終止之後持續保存磁碟區的資訊，請參閱 在 Amazon EC2 執行個體終止後保留 Amazon EBS 根磁碟區。

** 僅支援 `io2` EBS Block Express。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的佈建 IOPS SSD Block Express 磁碟區。

依據 AMI 確定的根磁碟區類型

您用來啟動 EC2 執行個體的 AMI 決定了根磁碟區的類型。EC2 執行個體的根磁碟區為 EBS 磁碟區或執行個體儲存體磁碟區。

基於 Nitro 的執行個體僅支援 EBS 根磁碟區。僅下面的上一代執行個體類型是支援執行個體儲存體根磁碟區的執行個體類型：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

Console

確定依據 AMI 決定的根磁碟區類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 AMIs，然後選取 AMI。
3. 在 Details (詳細資訊) 索引標籤中，檢查 Root device type (根設備類型) 的值，如下所示：
   - `ebs` – 透過此 AMI 啟動的執行個體會獲得 EBS 根磁碟區
   - `instance store` – 透過此 AMI 啟動的執行個體會獲得執行個體儲存體根磁碟區。

AWS CLI

確定依據 AMI 決定的根磁碟區類型

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].RootDeviceType
```

以下為範例輸出。

```
ebs
```

PowerShell

確定依據 AMI 決定的根磁碟區類型

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -ImageId ami-0abcdef1234567890).RootDeviceType.Value
```

以下為範例輸出。

```
ebs
```

虛擬化類型

Amazon Machine Image 會使用兩種虛擬化類型的其中之一：全虛擬化 (PV) 或硬體虛擬機器 (HVM)。PV 和 HVM AMI 之間的主要區別在於開機的方式以及是否可以利用特殊的硬體延伸 (CPU、網路和儲存) 來獲得更好的效能。Windows AMI 為 HVM AMI。

下表比較 HVM 和 PV AMI。

特性	HVM	PV
描述	透過執行映像之根區塊型儲存設備的主開機記錄，HVM AMI 會以完全虛擬化的一組硬體呈現並開機。這種虛擬化類型提供在沒有任何修改的情況下直接在虛擬機器上執行作業系統之能力，就像執行在裸機硬體上一樣。Amazon EC2 主機系統模擬提供給訪客部分或全部底層硬體。	PV AMIs 使用稱為 PV-GRUB 的特殊開機載入器開機，該開機載入器會啟動開機週期，然後將 `menu.lst` 檔案中指定的核心載入映像中。全虛擬化訪客可以在不明確支援虛擬化的主機硬體上執行。如需 Amazon EC2 中 PV-GRUB 及其用途的詳細資訊，請參閱使用者提供的核心。
支援的執行個體類型	所有目前世代的執行個體類型都支援 HVM AMI。	下列的前代執行個體類型支援 PV AMI：C1、C3、M1、M3、M2 和 T1。最新一代執行個體類型不支援 PV AMI。
硬體延伸支援	HVM 訪客可以利用硬體延伸來快速存取主機系統上的底層硬體。其需使用增強型網路和 GPU 處理。為了將指令傳遞給專用網路和 GPU 裝置，作業系統必須能夠存取本地硬體平台；HVM 虛擬化提供此存取。如需詳細資訊，請參閱增強 Amazon EC2 執行個體的聯網能力。	不支援，它們無法利用特殊的硬體延伸，例如增強聯網或 GPU 處理。

特性	HVM	PV
如何尋找	使用主控台或 describe-images 命令來確認 AMI 的虛擬化類型已設定為 hvm。	使用主控台或 describe-images 命令來確認 AMI 的虛擬化類型已設定為 paravirtual 。

HVM 上的 PV

全虛擬化訪客傳統上在儲存和網路操作效能上較 HVM 訪客更佳，因為他們可以利用特殊的 I/O 驅動程式避免模擬網路和硬碟硬體的成本，而 HVM 訪客必須將這些指令轉換為模擬硬體。現在，PV 驅動程式可用於 HVM 訪客，因此無法移植到全虛擬化環境中執行的作業系統，仍然可以藉此在儲存體和網路 I/O 中看到效能優勢。使用 HVM 驅動程式上的這些 PV，HVM 訪客可以取得與全虛擬化訪客相同或更佳的效能。

尋找符合 EC2 執行個體需求的 AMI

AMI 包含啟動執行個體所需的元件和應用程式，例如作業系統和根磁碟區類型。若要啟動執行個體，您必須找到符合您需求的 AMI。

選取 AMI 時，請針對要啟動的執行個體，考慮下列可能會有的要求：

- 做為 AMI IDs 的 AMI AWS 區域對每個區域都是唯一的。
- 作業系統 (例如 Linux 或 Windows)。
- 架構 (例如，32 位元、64 位元或 64 位元 ARM)。
- 根磁碟區類型 (例如 Amazon EBS 或執行個體儲存體)。
- 供應商 (例如，Amazon Web Services)。
- 其他軟體 (例如，SQL Server)。

Console

當您使用啟動執行個體精靈來啟動執行個體時，可以從 AMI 清單中選取，也可以使用映像頁面搜尋所有可用的 AMI。

如需使用啟動執行個體精靈來尋找 Quick Start AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 從導覽列選取要在其中啟動執行個體的區域。無論您的位置為何，皆可選取任何可用的區域。AMI IDs 對每個 AWS 區域都是唯一的。
3. 從主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. 在應用程式和作業系統映像 (Amazon Machine Image) 下，選取快速入門，為您的執行個體選取作業系統 (OS)，然後在 Amazon Machine Image (AMI) 中，從清單中選取其中一個常用的 AMI。若未看到您需要的 AMI，您可選擇 Browser more AMIs (瀏覽更多 AMI) 以瀏覽完整的 AMI 目錄。如需詳細資訊，請參閱應用程式和作業系統映像 (Amazon Machine Image)。

使用 AMI 頁面來尋找 AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽列選取要在其中啟動執行個體的區域。無論您的位置為何，皆可選取任何可用的區域。AMI IDs 對每個 AWS 區域都是唯一的。
3. 在導覽窗格中，選擇 AMI。
4. (選用) 利用 Filter (篩選) 和 Search (搜尋) 選項來限定所顯示 AMI 的清單範圍，只列出符合標準的 AMI。

   例如，若要列出 提供的所有 AMIs AWS，請選擇公有映像。然後使用 Search (搜尋) 選項進一步確定顯示 AMI 清單的範圍。選擇 Search (搜尋) 列，然後從選單中依序選擇 Owner alias(擁有者別名)、= 運算子和值 amazon。若要尋找符合特定平台 (例如 Linux 或 Windows) 的 AMI，請再次選擇搜尋列，並依序選擇平台、= 運算子，然後從提供的清單選擇作業系統。
5. (選用) 選擇喜好設定圖示，以選取要顯示的映像屬性，例如根磁碟區類型。或者，您可以從清單中選取 AMI，然後在 Details (詳細資訊) 索引標籤上檢視其屬性。
6. 在選取 AMI 之前，重要的是檢查 AMI 是否為執行個體儲存體支援的執行個體或 Amazon EBS 支援的執行個體，而且您知悉這種差異造成的影響。如需詳細資訊，請參閱根磁碟區類型。
7. 若要從這個 AMI 啟動執行個體，請先選取該 AMI，然後再選擇從映像啟動執行個體。如需有關使用主控台啟動執行個體的詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。如果您目前尚未準備好要啟動執行個體，請記下該 AMI ID，以待日後使用。

AWS CLI

請使用 describe-images 命令，來尋找滿足您要求的 AMI。此命令預設會傳回全部公有、您擁有，以及與您共用的 AMI。

如需尋找 Amazon 擁有的 AMI

使用 `--owners` 選項並搭配 describe-images 命令。

```
aws ec2 describe-images --owners amazon
```

如需尋找 Windows AMI

新增下列篩選條件只會顯示 Windows AMI。

```
--filters "Name=platform,Values=windows"
```

如需尋找 EBS 支援的 AMI

新增下列篩選條件只會顯示 Amazon EBS 支援的 AMI：

```
--filters "Name=root-device-type,Values=ebs"
```

PowerShell

請使用 Get-EC2Image cmdlet，來尋找滿足您要求的 AMI。此 cmdlet 預設會傳回全部公有、您擁有，或者與您共用的 AMI。

如需尋找 Amazon 擁有的 AMI

使用 `-Owner` 參數搭配 Get-EC2Image 命令。

```
Get-EC2Image -Owner amazon
```

如需尋找 Windows AMI

新增下列篩選條件只會顯示 Windows AMI。

```
-Filter @{Name="platform"; Values="windows"}
```

如需額外範例，請參閱 AWS Tools for PowerShell 使用者指南中的使用 Windows PowerShell 尋找 Amazon 系統映像。

相關資源

若要了解特定作業系統 AMI 的相關詳細資訊，請參閱以下各項：

- Amazon Linux 2023 – Amazon Linux 2023 使用者指南中的 AL2023 on Amazon EC2
- Ubuntu – Canonical Ubuntu 網站上的 Amazon EC2 AMI 定位器
- RHEL – Red Hat 網站的 Amazon Web Services (AWS) 提供的 Red Hat Enterprise Linux Images (AMI)
- Windows Server – AWS Windows AMI 參考

如需您可以在 上訂閱AMIs 相關資訊， AWS Marketplace 請參閱 AWS Marketplace 適用於 Amazon EC2 執行個體的 中的付費 AMIs。

若要了解有關如何使用 Systems Manager，協助使用者尋找在啟動執行個體時應使用的最新 AMI 的詳細資訊，請參閱以下各項：

- 使用 Systems Manager 來參數參考 AMI
- 使用 Systems Manager 公有參數來參考最新的 AMI

使用 Systems Manager 來參數參考 AMI

當您在 Amazon EC2 主控台中使用 EC2 啟動執行個體精靈啟動執行個體時，您可以從清單中選擇 AMI，也可以選取指向 AMI ID 的 AWS Systems Manager 參數 （本節中說明）。 Amazon EC2 如果您使用自動化程式碼來啟動執行個體，則可以指定 Systems Manager 參數而非 AMI ID。

Systems Manager 參數是您可以在 Systems Manager 參數存放區中建立的客戶定義索引鍵/值組。參數存放區提供一個中央存放區，可將您的應用程式組態值外部化。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 AWS Systems Manager 參數存放區。

當您建立指向 AMI ID 的參數時，請務必將資料類型指定為 `aws:ec2:image`。指定此資料類型可確保在建立或修改參數時，參數值會驗證為 AMI ID。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 Amazon Machine Image ID 的原生參數支援。

內容

- 使用案例
- 許可
- 限制
- 使用 Systems Manager 參數啟動執行個體

使用案例

使用 Systems Manager 參數指向 AMI ID 時，能讓使用者在啟動執行個體時更輕鬆選取正確的 AMI。Systems Manager 參數還可以簡化自動化程式碼的維護。

Easier for users (對使用者更簡單)

如果您要求使用特定 AMI 啟動執行個體，且該 AMI 定期更新，我們建議您要求使用者選取 Systems Manager 參數來尋找 AMI。透過要求使用者選取 Systems Manager 參數，您可以確保使用最新的 AMI 來啟動執行個體。

例如，您可以在組織中每月建立新版本 AMI，其中包含最新的作業系統和應用程式修補程式。您也要求使用者使用最新版本的 AMI 來啟動執行個體。為了確保您的使用者使用最新版本，您可以建立指向正確 AMI ID 的 Systems Manager 參數 (例如 `golden-ami`)。每次建立新版 AMI 時，您都會更新參數中的 AMI ID 值，使其永遠指向最新的 AMI。您的使用者不需要知道 AMI 的定期更新，因為他們每次都會繼續選取相同的 Systems Manager 參數。使用 AMI 的 Systems Manager 參數可在啟動執行個體時更輕鬆地選取正確的 AMI。

Simplify automation code maintenance (簡化自動化程式碼維護)

如果您使用自動化程式碼來啟動執行個體，則可以指定 Systems Manager 參數而非 AMI ID。如果新版本的 AMI 已建立，您可以變更參數中的 AMI ID 值，使其指向最新的 AMI。每次建立新版本的 AMI 時，都不需要修改參考參數的自動化程式碼。如此可簡化自動化維護作業，並有助於降低部署成本。

> **Note**
>
> 當您變更 Systems Manager 參數指向的 AMI ID 時，執行中的執行個體不會受到影響。

許可

如果您在啟動執行個體精靈中使用指向 AMI ID 的 Systems Manager 參數，則必須新增下列許可至 IAM 政策：

- `ssm:DescribeParameters` - 授予許可，以檢視及選取 Systems Manager 參數。
- `ssm:GetParameters` - 授予許可，以擷取 Systems Manager 參數值。

您也可以限制對特定 Systems Manager 參數的存取。如需更多資訊與範例 IAM 政策，請參閱 範例：使用 EC2 啟動執行個體精靈。

限制

AMI 和 Systems Manager 參數是區域特定的。若要跨區域使用相同的 Systems Manager 參數名稱，請在每個區域中建立具有相同名稱的 Systems Manager 參數 (例如 `golden-ami`)。在每個區域中，將 Systems Manager 參數指向該區域中的 AMI。

參數名稱區分大小寫。參數名稱的反斜線只有在參數是階層的一部分時才需要，例如 `/amis/production/golden-ami`。如果參數不是階層架構的一部分，則可以省略反斜線。

使用 Systems Manager 參數啟動執行個體

啟動執行個體時，您可以指定指向 AMI ID 的 Systems Manager 參數，而非指定 AMI ID。

若要以程式設定方式指定參數，請使用下列語法，其中 `resolve:ssm` 是標準字首，`parameter-name` 是不重複的參數名稱。

```
resolve:ssm:parameter-name
```

Systems Manager 參數擁有版本支援。參數的每個反覆項目都會指定一個唯一的版本號碼。您可以參考參數的版本，如下所示，其中 `version` 是不重複的版本號碼。根據預設，在未指定任何版本時，會使用最新版本的參數。

```
resolve:ssm:parameter-name:version
```

若要使用 提供的公有參數啟動執行個體 AWS，請參閱 使用 Systems Manager 公有參數來參考最新的 AMI。

Console

如需使用 Systems Manager 參數來尋找 AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽列選取要在其中啟動執行個體的區域。無論您的位置為何，皆可選取任何可用的區域。
3. 從主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，選擇 Browse more AMIs (瀏覽更多 AMI)。
5. 選擇搜尋列右側的箭頭按鈕，然後選擇 Search by Systems Manager parameter (依 Systems Manager 參數搜尋)。

6. 對於 Systems Manager parameter (Systems Manager 參數)，請選取一個參數。對應的 AMI ID 會出現在 Currently resolves to (目前解析為) 下方。
7. 選擇 Search (搜尋)。與 AMI ID 相符的 AMI 會顯示在清單中。
8. 從清單中選取 AMI，然後選取 Select (選取)。

如需有關使用啟動執行個體精靈啟動執行個體的詳細資訊，請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

如需使用 Systems Manager 參數來啟動執行個體

請使用 `--image-id` 選項並搭配 run-instances 命令。此範例使用名稱為 **golden-ami** 的 Systems Manager 參數，這會指定 AMI ID。

```
--image-id resolve:ssm:/golden-ami
```

您可建立 Systems Manager 參數版本。下列範例會指定 **golden-ami** 參數的版本 2。

```
--image-id resolve:ssm:/golden-ami:2
```

PowerShell

如需使用 Systems Manager 參數來啟動執行個體

使用 New-EC2Instance cmdlet 搭配 `-ImageId` 參數。此範例使用名稱為 **golden-ami** 的 Systems Manager 參數，這會指定 AMI ID。

```
-ImageId "resolve:ssm:/golden-ami"
```

您可建立 Systems Manager 參數版本。下列範例會指定 **golden-ami** 參數的版本 2。

```
-ImageId "resolve:ssm:/golden-ami:2"
```

使用 Systems Manager 公有參數來參考最新的 AMI

AWS Systems Manager 為 維護的公有 AMIs 提供公有參數 AWS。您可在啟動執行個體時使用公有參數來確保您使用的是最新 AMI。例如，公有參數 `/aws/service/ami-amazon-linux-latest/`

`al2023-ami-kernel-default-arm64` 可在所有區域中使用，並且一律指向指定區域中適用於 arm64 架構的 Amazon Linux 2023 AMI 最新版本。

公有參數可從以下路徑取得：

- Linux – `/aws/service/ami-amazon-linux-latest`
- Windows – `/aws/service/ami-windows-latest`

如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的使用公有參數。

尋找公有 AMI 的 SSM 參數

當您描述公有 AMI 時，回應會在 `PublicSsmParameterName` 欄位中包含 associated AWS Systems Manager (SSM) 參數。使用此參數來識別一律指向該 AMI 最新版本的 SSM 參數。

Console

尋找公有 AMI 的 SSM 參數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取公有 AMI。
4. 在詳細資訊索引標籤上，尋找公有 SSM 參數名稱。

AWS CLI

尋找公有 AMI 的 SSM 參數

使用 describe-images 命令。回應包含具有相關聯 SSM 參數的公AMIs `PublicSsmParameterName` 欄位。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890
```

下列範例輸出顯示 `PublicSsmParameterName` 欄位。為了簡潔起見，省略某些欄位。

```
{
    "Images": [
        {
            "ImageId": "ami-0abcdef1234567890",
```

```
            "Name": "al2023-ami-2023.7.20260601.0-kernel-6.1-x86_64",
            ...
            "PublicSsmParameterName": "aws/service/ami-amazon-linux-latest/al2023-
ami-kernel-default-x86_64"
        }
    ]
}
```

PowerShell

尋找公有 AMI 的 SSM 參數

使用 Get-EC2Image cmdlet。回應包含具有相關聯 SSM 參數的公AMIs `PublicSsmParameterName` 欄位。

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

下列範例輸出顯示 `PublicSsmParameterName` 欄位。為了簡潔起見，省略某些欄位。

```
ImageId                : ami-0abcdef1234567890
Name                   : al2023-ami-2023.7.20260601.0-kernel-6.1-x86_64
...
PublicSsmParameterName : aws/service/ami-amazon-linux-latest/al2023-ami-kernel-
default-x86_64
```

列示 Amazon Linux AMI

AWS CLI

列出目前區域中AMIs AWS

請使用下面的 get-parameters-by-path 命令。`--path` 參數的值為 Linux AMI 特定。

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-amazon-linux-latest \
    --query "Parameters[].Name"
```

PowerShell

列出目前區域中AMIs AWS

使用 Get-SSMParametersByPath cmdlet。

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-amazon-linux-latest" | `
    Sort-Object Name | Format-Table Name
```

列示 Windows AMI

AWS CLI

列出目前 AWS 區域中AMIs

請使用下面的 get-parameters-by-path 命令。`--path` 參數的值為 Windows AMI 特定。

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-windows-latest \
    --query "Parameters[].Name"
```

PowerShell

列出目前 AWS 區域中AMIs

使用 Get-SSMParametersByPath cmdlet。

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-windows-latest" | `
    Sort-Object Name | Format-Table Name
```

使用公有參數啟動執行個體

如需在啟動執行個體時指定公有參數，請使用下列語法：`resolve:ssm:`*`public-parameter`*，其中 `resolve:ssm` 是標準字首，而 *`public-parameter`* 是公有參數的路徑和名稱。

AWS CLI

使用公有參數啟動執行個體

請使用 `--image-id` 選項並搭配 run-instances 命令。此範例指定映像 ID 的 Systems Manager 公有參數，以使用最新 Amazon Linux 2023 AMI 啟動執行個體

```
--image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64
```

PowerShell

使用公有參數啟動執行個體

使用 New-EC2Instance cmdlet 搭配 `-ImageId` 參數。此範例指定映像 ID 的 Systems Manager 公有參數，以使用最新 AMI for Windows Server 2022 啟動執行個體

```
-ImageId "resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base"
```

如需使用 Systems Manager 參數的更多範例，請參閱使用 AWS Systems Manager 參數存放區查詢最新的 Amazon Linux AMI IDs 和使用 AWS Systems Manager 參數存放區查詢最新的 Windows AMI。

AWS Marketplace 適用於 Amazon EC2 執行個體的 中的付費 AMIs

付費 AMI 是在 AWS Marketplace列出待銷售的 AMI。 AWS Marketplace 是一個線上商店，您可以在其中購買 上執行的軟體 AWS，包括可用來啟動 EC2 執行個體AMIs。 AWS Marketplace AMIs 會組織成各種類別，例如開發人員工具，讓您能夠找到符合您需求的產品。如需 的詳細資訊 AWS Marketplace，請參閱 AWS Marketplace網站。

您可以從 AWS Marketplace 第三方購買 中的 AMIs，包括來自 Red Hat 等組織的服務合約隨附的 AMIs。您也可以建立 AMI，並在 中將它銷售 AWS Marketplace 給其他 Amazon EC2 使用者。只要遵循一些簡單準則，您即可經由簡單明瞭的程序建立供公眾取用的安全可用 AMI。如需如何建立和使用共用 AMI 的資訊，請參閱 了解 Amazon EC2 的共用 AMI 使用情況。

從已支付 AMI 啟動執行個體與從任何其他的 AMI 啟動執行個體的方式相同。不需額外的參數。執行個體會根據 AMI 擁有者設定的費率收費，其相關 Web 服務的標準使用費 (例如在 Amazon EC2 中執行 m5.small 執行個體類型的每小時費率) 也相同。可能也需支付額外稅金。已支付 AMI 的擁有者可確認特定執行個體是否使用該已支付 AMI 啟動。

> **Important**
>
> Amazon DevPay 不再接受新的賣方或產品。 現在 AWS Marketplace 是用於透過 銷售軟體和服務的唯一、統一的電子商務平台 AWS。如需有關如何從中部署和銷售軟體的資訊 AWS Marketplace，請參閱在 Marketplace 中 AWS 銷售。 AWS Marketplace 支援 Amazon EBS 支援的 AMIs。

目錄

- 在 中銷售您的 AMI AWS Marketplace
- 尋找已支付 AMI
- 在 中購買付費 AMI AWS Marketplace
- 從執行個體擷取 AWS Marketplace 產品程式碼
- 針對支援 AWS Marketplace 的產品使用付費支援
- 已支付及支援的 AMI 的帳單
- 管理您的 AWS Marketplace 訂閱

在 中銷售您的 AMI AWS Marketplace

您可以使用 銷售您的 AMI AWS Marketplace。 AWS Marketplace 提供有條不紊的購物體驗。此外，AWS Marketplace 也支援 Amazon EBS 支援的 AMIs、預留執行個體和 Spot 執行個體等 AWS 功能。

如需有關如何在 上銷售 AMI 的資訊 AWS Marketplace，請參閱在 Marketplace 中 AWS 銷售。

尋找已支付 AMI

付費 AMI 是可供購買的 Amazon Machine Image (AMI)。付費 AMI 亦有產品代碼。您可在 AWS Marketplace中尋找可供購買的 AMI。

Console

如需尋找付費 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMIs (AMI)。
3. 中，選擇 Public images (公有映像) 作為第一個篩選條件。
4. 執行以下任意一項：
   - 如果您知道產品代碼，請選擇 Product code (產品代碼) 和 =，然後輸入產品代碼。
   - 若不知道產品代碼，可在搜尋列指定以下篩選條件：擁有者別名=aws-marketplace。視需指定額外的篩選條件。
5. 儲存 AMI 的 ID。

AWS CLI

如需尋找付費 AMI

使用下面的 describe-images 命令。

```
aws ec2 describe-images --owners aws-marketplace
```

輸出包括大量映像。您可指定篩選條件，協助您確定需要哪個 AMI。找到 AMI 之後，在下面的命令中指定其 ID，以獲取產品代碼。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].ProductCodes[].ProductCodeId
```

以下為範例輸出。

```
[
    "cdef1234abc567def8EXAMPLE"
]
```

如果您知道產品代碼，您可以依產品代碼篩選結果。此範例會傳回最新 AMI 與指定的產品代碼。

```
aws ec2 describe-images \
    --filters "Name=product-code,Values=cdef1234abc567def8EXAMPLE" \
    --query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

PowerShell

如需尋找付費 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image -Owner aws-marketplace
```

輸出包括大量映像。您可指定篩選條件，協助您確定需要哪個 AMI。找到 AMI 之後，在下面的命令中指定其 ID，以獲取產品代碼。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ProductCodes
```

以下為範例輸出。

```
ProductCodeId             ProductCodeType
```

```
-------------              ---------------
cdef1234abc567def8EXAMPLE marketplace
```

如果您知道產品代碼，您可以依產品代碼篩選結果。此範例會傳回最新 AMI 與指定的產品代碼。

```
(Get-EC2Image -Owner aws-marketplace -Filter @{"Name"="product-
code";"Value"="cdef1234abc567def8EXAMPLE"} | sort CreationDate -Descending | Select-
Object -First 1).ImageId
```

在 中購買付費 AMI AWS Marketplace

您必須先為付費 AMI 註冊 (購買)，然後才能使用 AMI 啟動 Amazon EC2 執行個體。

通常已支付 AMI 的賣方會提供您 AMI 的相關資訊，包含價格和可購買該 AMI 的連結。當您選擇連結時，系統會先要求您登入 AWS，然後您可以購買 AMI。

使用主控台購買已支付 AMI

您可以使用 Amazon EC2 Launch Wizard 購買已支付 AMI。如需詳細資訊，請參閱從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體。

使用 訂閱產品 AWS Marketplace

若要使用 AWS Marketplace，您必須擁有 AWS 帳戶。若要從 AWS Marketplace 產品啟動執行個體，您必須註冊才能使用 Amazon EC2 服務，而且您必須訂閱要從中啟動執行個體的產品。您可使用下列其中一種方式，在 AWS Marketplace訂閱產品：

- AWS Marketplace 網站：您可以使用 1-Click 部署功能快速啟動預先設定的軟體。如需詳細資訊，請參閱 中的 AMI 型產品 AWS Marketplace。
- Amazon EC2 啟動精靈：您可以搜尋 AMI，然後直接從精靈啟動執行個體。如需詳細資訊，請參閱從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體。

從執行個體擷取 AWS Marketplace 產品程式碼

您可以使用執行個體中繼資料擷取執行個體 AWS Marketplace 的產品碼。若執行個體具有產品代碼，Amazon EC2 便會傳回它。如需擷取中繼資料的詳細資訊，請參閱 存取 EC2 執行個體的執行個體中繼資料。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/product-codes
```

Windows

透過 Windows 執行個體執行下面的 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/product-codes
```

Windows

透過 Windows 執行個體中執行以下命令。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/product-codes
```

針對支援 AWS Marketplace 的產品使用付費支援

Amazon EC2 同時也可讓開發人員提供軟體 (或衍生 AMI) 的支援。開發人員能建立可讓您註冊使用的支援產品。在註冊支援產品時，開發人員會給予您產品代碼，您必須將該產品代碼與您擁有的 AMI 建

立關聯。這可讓開發人員確認您的執行個體符合取得支援的資格。它也能確認在您執行產品執行個體的時候，您確實有根據開發人員指定的產品規定支付費用。

限制

- 在您設定產品代碼屬性後，您便無法變更或移除它。
- 您無法搭配預留執行個體使用支援產品。您一律必須支付支援產品賣方指定的價格。

AWS CLI

如需關聯產品代碼與 AMI

請使用 modify-image-attribute 命令。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --product-codes "cdef1234abc567def8EXAMPLE"
```

PowerShell

如需關聯產品代碼與 AMI

使用 Edit-EC2ImageAttribute cmdlet。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ProductCode "cdef1234abc567def8EXAMPLE"
```

已支付及支援的 AMI 的帳單

在每個月結束的時候，您會收到一封電子郵件，其中包含在該月份中使用任何已支付或支援的 AMI，而向您信用卡收取的金額。此帳單與您的一般 Amazon EC2 帳單各自獨立。如需詳細資訊，請參閱「AWS Marketplace 買方指南」中的產品付款。

管理您的 AWS Marketplace 訂閱

在 AWS Marketplace 網站上，您可以檢查訂閱詳細資訊、檢視廠商的使用說明、管理您的訂閱等等。

檢查 訂閱詳細資料

檢查您的訂閱詳細資訊

1. 登入 AWS Marketplace。
2. 選擇 Your Marketplace Account (您的 Marketplace 帳戶)。
3. 選擇 Manage your software subscriptions (管理您的軟體訂閱)。
4. 將會列出所有您目前的訂閱。選擇使用說明檢視使用產品的特定說明，例如：用來連線到您執行中執行個體的使用者名稱。

取消訂閱

> **Note**
>
> - 取消訂閱不會終止透過該 AMI 啟動的執行個體。我們將繼續向您收取執行中執行個體的費用，直至終止使用執行個體。您必須終止使用透過 AMI 啟動的全部執行個體，方可停止對訂閱計費。
> - 在您取消您的訂閱之後，您便無法從該 AMI 啟動任何執行個體。若要再次使用該 AMI，您需要在 AWS Marketplace 網站上或透過 Amazon EC2 主控台中的啟動精靈重新訂閱該 AMI。

若要取消 AWS Marketplace 訂閱

1. 如需停止對訂閱計費，請確認您已終止任何從該訂閱執行的執行個體。

   > **Warning**
   >
   > 終止執行個體的操作為永久性且無法復原。
   > 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
   > 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

   a. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
   b. 在導覽窗格中，選擇執行個體。

c. 選取執行個體，然後選取執行個體狀態、終止 (刪除) 執行個體。

d. 出現確認提示時，請選擇終止 (刪除)。

2. 登入 AWS Marketplace，然後依序選擇 Your Marketplace Account (您的 Marketplace 帳戶) 及 Manage your software subscriptions (管理您的軟體訂閱)。

3. 選擇 Cancel subscription (取消訂閱)。系統會提示您確認取消。

Amazon EC2 AMI 生命週期

Amazon Machine Image (AMI) 是一種映像，包含設定並啟動執行個體所需的軟體組態。在您啟動執行個體時，必須指定 AMI。您可使用 Amazon 提供的 AMI，亦可建立自己的 AMI。AMI 必須位於您要啟動執行個體 AWS 區域 的 中。

AMI 的生命週期包括建立、複製、棄用、停用，以及刪除 (取消註冊) AMI。

建立 AMI。Amazon 提供 AMI，您可用來啟動執行個體，也可根據您的需求量身建立自訂 AMI。如需建立自訂 AMI，可透過現有的 AMI 來啟動執行個體、自訂執行個體 (例如，安裝軟體及設定作業系統設定)，然後透過執行個體建立 AMI。所有執行個體自訂內容都會儲存至新 AMI，因此透過新 AMI 啟動的執行個體會包括這些自訂內容。

可驗證的 AMI。如需建立支援 EC2 執行個體驗證的 AMI，請參閱 可驗證的 AMI。

複製 AMI。您只能在 AMI 所在的 中，使用 AWS 區域 AMI 啟動執行個體。若您需要在多個區域啟動包含相同組態的執行個體，則複製 AMI 到其他區域。

棄用 AMI。如需將 AMI 標記為已取代或過時，可設定立即或未來的棄用日期。AMI 清單會隱藏棄用的 AMI，但若使用者及服務知道 AMI ID，可繼續使用棄用的 AMI。

停用 AMI。如需暫時阻止 AMI 使用執行個體，可停用 AMI。停用 AMI 後，無法將其用於啟動新的執行個體。然而，若重新啟用 AMI，可再次將其用於啟動執行個體。請注意，停用 AMI 不影響已經從中啟動的現有執行個體。

取消註冊 (刪除) AMI。若不再需要 AMI，可取消註冊，以免將其用於啟動新的執行個體。若 AMI 與保留規則相符，則會將其移至資源回收筒，其保留期到期之前可將其還原，在此之後會將其永久刪除。若與保留規則不相符，則會立即將其永久刪除。請注意，當您取消註冊 AMI 時，不會影響到任何透過該 AMI 啟動的現有執行個體。

自動化 AMI 生命週期。您可以使用 Amazon Data Lifecycle Manager 來自動建立、保留、複製、取代和取消註冊 Amazon EBS 後端 AMI 及其備份快照。您亦可使用 EC2 Image Builder 來自動建立、管

理、部署自訂 AMI。若要了解詳細資訊，請參閱 Amazon EBS 使用者指南和 EC2 Image Builder 使用者指南中的藉助 Amazon Data Lifecycle Manager 來自動化備份

目錄

- 建立 Amazon EBS 支援的 AMI
- 建立 Amazon S3 支援的 AMI
- 使用 Windows Sysprep 建立 Amazon EC2 AMI
- 複製 Amazon EC2 AMI
- 使用 S3 存放和還原 AMI
- 使用 AMI 原始伺服器追蹤 AMI 的原始伺服器
- 使用 AMI 浮水印來追蹤和識別 AMIs
- AMI 允許的執行個體類型
- 管理及監控 AMI 用量
- 棄用 Amazon EC2 AMI
- 停用 Amazon EC2 AMI
- 取消註冊 Amazon EC2 AMI

建立 Amazon EBS 支援的 AMI

您可從 Amazon EC2 執行個體或從 Amazon EC2 執行個體的根磁碟區快照建立自己的 Amazon EBS 支援的 AMI。

若要從執行個體建立 Amazon EBS 支援的 AMI，請從使用現有 Amazon EBS 支援的 AMI 啟動執行個體開始。此 AMI 可以是您從 取得 AWS Marketplace、使用 VM Import/Export 建立，或您可以存取的任何其他 AMI。在自訂執行個體以滿足您的特定需求後，請建立並註冊新 AMI。然後，您可使用新 AMI 並運用您的自訂內容來啟動全新執行個體。

> **Note**
>
> 如需建立支援 EC2 執行個體驗證的 AMI，請參閱 可驗證的 AMI。

下列說明的程序同時適用於加密 Amazon Elastic Block Store (Amazon EBS) 磁碟區 (包含根磁碟區) 及未加密磁碟區後端 Amazon EC2 執行個體。

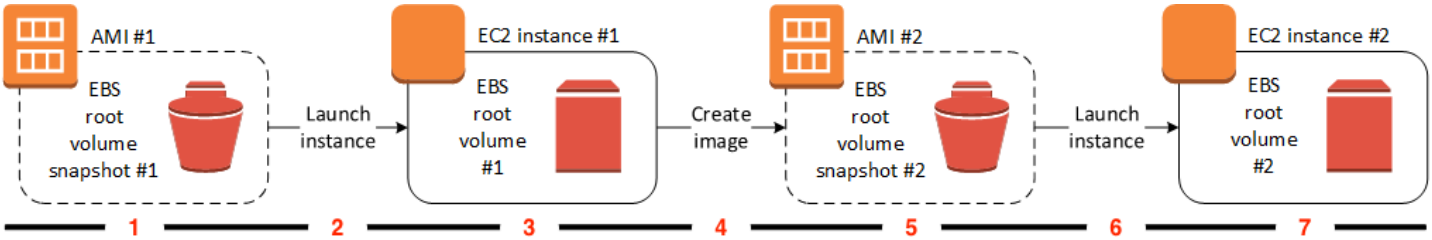
AMI 建立程序與 Amazon S3 支援的 AMI 不同。如需詳細資訊，請參閱建立 Amazon S3 支援的 AMI。

目錄

- 從執行個體建立 AMI 概覽
- 從執行個體建立 AMI
- 從快照建立 AMI

從執行個體建立 AMI 概覽

下列圖表總結了從正在執行的 EC2 執行個體建立 Amazon EBS 後端 AMI 的過程：從現有 AMI 開始，啟動執行個體，自訂執行個體，根據其建立新的 AMI，以及最終啟動您的新 AMI 執行個體。該圖表中的數字與隨後描述中的數字相符。



1 – AMI #1：從現有 AMI 開始

查找與要建立的 AMI 相似的現有 AMI。這可以是您從 取得的 AMI AWS Marketplace、您使用 VM Import/Export 建立的 AMI，或是您可以存取的任何其他 AMI。您將根據需要自訂此 AMI。

在圖表中,EBS 根磁碟區快照 #1 指示 AMI 是 Amazon EBS 後端 AMI，並且有關根磁碟區的資訊存放在此快照中。

2 – 從現有 AMI 啟動執行個體

設定 AMI 的方法是從您希望作為新 AMI 基礎的 AMI 啟動執行個體，然後自訂執行個體 (由圖表中的 3 指示)。然後，您將建立新的 AMI，其中包含自訂執行個體 (由圖表中的 4 指示)。

3 – EC2 執行個體 #1：自訂執行個體

連線到您的執行個體並根據需要自訂。您的新 AMI 將包含這些自訂執行個體。

您可以在執行個體上執行下列任一個動作以自訂：

- 安裝軟體和應用程式
- 複製資料
- 刪除暫存檔案並重組您的硬碟，來減少啟動時間。

- 連接額外的 EBS 磁碟區

4 – 建立映像

從執行個體建立 AMI 時，Amazon EC2 會在建立 AMI 前關閉執行個體，確認執行個體上的一切在建立程序中都維持在停止且一致的狀態。若您相信您的執行個體處於一致且可用來建立 AMI 的狀態，您可以告知 Amazon EC2 不要關閉及重新開機執行個體。有些檔案系統 (例如 XFS) 可能會凍結及取消凍結活動，使其足夠安全，可用來建立映像，而無須重新開機執行個體。

在 AMI 建立程序期間，Amazon EC2 會建立您執行個體根磁碟區及任何連接到您執行個體之其他 EBS 磁碟區的快照。您必須為快照支付費用，直到您取消註冊 AMI 及刪除快照為止。若任何連接到執行個體的磁碟區經過加密，則新的 AMI 只能在支援 Amazon EBS 加密的執行個體上成功啟動。

AMI 建立程序可能需要數分鐘的時間完成 (有些可能會花上 24 小時)，具體取決於磁碟區的大小。您可能會發現在建立您的 AMI 前先建立您磁碟區的快照會更有效率。透過此方式，建立 AMI 時便只需要建立小型的增量快照，且程序將能更快完成 (建立快照的總時間維持不變)。

5 – AMI #2：新 AMI

在程序完成後，您便會擁有新的 AMI 及從執行個體根磁碟區建立的快照 (快照 #2) 。若除了根磁碟區之外，您還為您的執行個體新增執行個體儲存體磁碟區或 EBS 磁碟區，則新 AMI 的區塊型儲存設備映射便會包含這些磁碟區的資訊。

Amazon EC2 會自動為您註冊 AMI。

6 –從新的 AMI 啟動執行個體

您可以使用新 AMI 啟動執行個體。

7 – EC2 執行個體 #2：新執行個體

當您使用新的 AMI 啟動執行個體時，Amazon EC2 會使用快照，建立此執行個體的根磁碟區的新 EBS 磁碟區。如果您在自訂執行個體時為新增執行個體儲存體磁碟區或 EBS 磁碟區，則新 AMI 的區塊型儲存設備映射便會包含這些磁碟區的資訊，並且您從新的 AMI 啟動之執行個體的區塊型儲存設備映射也會自動包含這些磁碟區的資訊。新執行個體之區塊型儲存設備映射中指定的執行個體儲存體磁碟區都是全新的，不會包含您用來建立 AMI 之執行個體的執行個體儲存體磁碟區中的任何資料。EBS 磁碟區上的資料會持續。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射。

當您從 EBS 後端 AMI 建立新的執行個體時，建議您在將其用於生產環境前初始化其根磁碟區和任何額外的 EBS 儲存體。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的初始化 Amazon EBS 磁碟區。

從執行個體建立 AMI

如果您有現有執行個體，您可從此執行個體建立 AMI。

Console

若要建立 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取要從中建立 AMI 的執行個體，然後依序選擇 Actions (動作)、Image and templates (映像和範本)，Create image (建立映像)。

> **Tip**
> 若此選項已停用，表示您的執行個體並非 Amazon EBS 後端執行個體。

4. 在 Create image (建立映像) 頁面上，指定下列資訊：
   a. 對於 Image name (映像名稱)，輸入映像的唯一名稱，最多 127 個字元。
   b. 對於 Image description (映像描述)，輸入映像的選擇性描述，最多 255 個字元。
   c. 對於重新啟動執行個體，請保持選取核取方塊 (預設值)，或加以清除。
      - 如果選取重新啟動執行個體，當 Amazon EC2 建立新的 AMI 時，它會重新啟動執行個體，以便在資料處於靜態時擷取連接磁碟區的快照，以確保狀態一致。
      - 若清除重新啟動執行個體，Amazon EC2 建立新 AMI 時，不會關閉及重新啟動執行個體。

> **Warning**
> 若您清除重新啟動執行個體，我們無法保證建立映像的檔案系統完整性。

   d. Instance volumes (執行個體磁碟區) – 您可以修改根磁碟區，以及新增額外的 Amazon EBS 及執行個體儲存體磁碟區，如下所示：
      i. 根磁碟區在第一列中定義。
         - 若要變更根磁碟區的大小，請在 Size (大小) 中輸入所需的值。

- 若您選取 Delete on termination (終止時刪除)，則當您終止從此 AMI 建立的執行個體時，便會刪除 EBS 磁碟區。若您清除 Delete on termination (終止時刪除)，則當您終止執行個體時，便不會刪除 EBS 磁碟區。如需詳細資訊，請參閱在執行個體終止時保留資料。

ii. 若要新增 EBS 磁碟區，請選擇 Add volume (新增磁碟區) (這會新增一個新的資料列)。針對儲存體類型，選擇 EBS，然後填入資料列中的欄位。當您從您新的 AMI 啟動執行個體時，額外的磁碟區會自動連接到執行個體。空白磁碟區必須進行格式化及掛載。以快照為基礎的磁碟區必須掛載。

iii. 若要新增執行個體儲存體磁碟區，請參閱將執行個體儲存體磁碟區新增至 Amazon EC2 AMI。當您從您新的 AMI 啟動執行個體時，額外的磁碟區會自動初始化並掛載。這些磁碟區不會包含做為您 AMI 基礎之執行中執行個體的執行個體儲存體磁碟區中的任何資料。

e. 快照目的地 – 若執行個體磁碟區位於支援 EBS 本機快照的本機區域，則選擇建立 AMI 快照的位置：

- AWS 區域：在磁碟區的本機區域父區域建立快照。
- AWS Local Zone：在與磁碟區相同的 Local Zone 中建立快照。

> **Note**
>
> 此選項僅在支援 EBS 本機快照的本機區域顯示，且僅當執行個體在本機區域建立時才會顯示。若磁碟區位於區域，則不會顯示此選項，且自動在與磁碟區相同的區域建立快照。若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的本機區域中的本機快照。

> **Important**
>
> 執行個體磁碟區的全部快照必須位於同一位置。請確認現有快照的位置。若任何現有的快照與選取的目的地位於不同位置，AMI 建立會失敗。

f. Tags (標籤) - 當您建立 AMI 時，可以使用相同的標籤來標記 AMI 和快照，或者可以使用不同的標籤來進行標記。

- 若要使用相同的 標籤來標記 AMI 和快照，請選擇 Tag image and snapshots together (共同標記映像和快照)。相同的標籤即套用至 AMI 和每個建立的快照。

- 若要使用不同的 標籤來標記 AMI 和快照，請選擇 Tag image and snapshots separately (個別標記映像和快照)。不同的標籤即套用至 AMI 和建立的快照。但是所有快照都會取得相同的標籤；您無法使用不同的標籤標記每個快照。

若要新增標籤，請選擇 Add tag (新增標籤)，然後輸入標籤的鍵和值。針對每個標籤，重複執行上述動作。

g. 當您準備好建立 AMI 時，請選擇 Create image (建立映像)。

5. 若要在建立您 AMI 的過程中檢視狀態：

a. 在導覽窗格中，選擇 AMI。

b. 將篩選條件設為 Owned by me (由我擁有)，然後在清單中尋找您的 AMI。

一開始，狀態為 `pending`，但在幾分鐘後應該會變更為 `available`。

6. (選用) 若要檢視為新 AMI 建立的快照：

a. 記下您在先前步驟中找到的 AMI ID。

b. 在導覽窗格中，選擇 Snapshots (快照)。

c. 將篩選條件設定為 Owned by me (由我擁有)，然後在 Description (說明) 欄中尋找含有新 AMI ID 的快照。

當您從此 AMI 啟動執行個體時，Amazon EC2 會使用此快照建立執行個體的根磁碟區。

AWS CLI

若要建立 AMI

請使用 create-image 命令。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-web-server" \
    --description "My web server image" \
    --no-reboot
```

PowerShell

若要建立 AMI

使用 New-EC2Image cmdlet。

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-web-server" `
    -Description "My web server image" `
    -NoReboot $true
```

從快照建立 AMI

若您有執行個體根磁碟區的快照，您可以從此快照建立 AMI。

> **Note**
>
> 在多數情況，適用於 Windows、Red Hat、SUSE 和 SQL Server 的 AMI 需要 AMI 具正確授權資訊。如需詳細資訊，請參閱了解 AMI 帳單資訊。當從快照建立 AMI 時，RegisterImage 操作會從快照的中繼資料衍生出正確帳單資訊，但這需要有適當的中繼資料。若要驗證是否已套用正確帳單資訊，請參閱新 AMI 的平台詳細資料欄位。如果欄位為空或與預期的作業系統程式碼不相符 (例如， Windows、Red Hat、SUSE 或 SQL)，則 AMI 建立失敗，您應該捨棄 AMI 並遵循 從執行個體建立 AMI 的說明。

Console

從快照建立 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選取要從中建立 AMI 的快照，然後選取 Actions (動作)、Create image from snapshot (從快照建立映像)。
4. 在從快照建立映像頁面上，指定下列資訊：
   a. 對於 Image name (映像名稱)，輸入映像的描述性名稱。
   b. 對於 Description (描述)，輸入映像的簡短描述。
   c. 對於 Architecture (架構)，選擇映像架構。選擇 i386 (若為 32 位元)、x86_64 (若為 64 位元)、arm64 (若為 64 位元 ARM)，或 x86_64 (若為 64 位元 macOS)。
   d. 針對根裝置名稱，輸入要用於根磁碟區的設備名稱。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的裝置名稱。

e. 對於 Virtualization type (虛擬化類型)，選擇從這個 AMI 啟動之執行個體要使用的虛擬化類型。如需詳細資訊，請參閱 虛擬化類型。

f. (僅適用於半虛擬化) 對於 Kernel ID (核心 ID)，選取映像的作業系統核心。如果您是使用執行個體的根磁碟區快照，請選取與原始執行個體相同的核心 ID。如果不確定，請使用預設核心。

g. (僅適用於半虛擬化) 對於 RAM disk ID (RAM 磁碟 ID)，選取映像的 RAM 磁碟。如果選取特定核心，則可能需要選取具有可支援之驅動程式的特定 RAM 磁碟。

h. 對於開機模式，請選擇映像的開機模式，或選擇使用預設值，以便在使用此 AMI 啟動執行個體時，以執行個體類型支援的開機模式開機。如需詳細資訊，請參閱設定 Amazon EC2 AMI 的開機模式。

i. (選用) 在區塊型儲存設備映射，自訂根磁碟區並新增其他資料磁碟區。

   針對每個磁碟區，您可以指定大小、類型、效能特性、終止時的刪除行為，以及加密狀態。若為根磁碟區，大小不可小於快照的大小。磁碟區類型的預設選項是「一般用途 SSD gp3」。

j. (選用) 在標籤下，您可以新增一個或多個標籤至新 AMI。若要新增標籤，請選擇 Add tag (新增標籤)，然後輸入標籤的鍵和值。針對每個標籤，重複執行上述動作。

k. 當您準備好建立 AMI 時，請選擇 Create image (建立映像)。

5. (僅限 Windows、Red Hat、SUSE 和 SQL Server) 若要驗證是否已套用正確帳單資訊，請檢查新 AMI 的平台詳細資料欄位。如果欄位為空或與預期的作業系統程式碼不相符 (例如，Windows 或 Red Hat)，則 AMI 建立失敗，您應該捨棄 AMI 並遵循 從執行個體建立 AMI 的說明。

AWS CLI

使用 從快照建立 AMI AWS CLI

使用 register-image 命令。

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f}
```

PowerShell

如需使用 PowerShell 透過快照建立 AMI

使用 Register-EC2Image cmdlet。

```
$block = @{SnapshotId=snap-0db2cf683925d191f}
Register-EC2Image `
    -Name my-image `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block}
```

建立 Amazon S3 支援的 AMI

您啟動執行個體時指定的 AMI，會判定根磁碟區的類型。

若要建立 Amazon S3 支援的 Linux AMI，請從您已從現有 Amazon S3 支援的 Linux AMI 啟動的執行個體開始。在您將執行個體自訂到符合您的需求後，建立磁碟區並註冊新的 AMI。您可以搭配這些自訂項目，使用這個新的 AMI 啟動新的執行個體。

您無法建立 Amazon S3 支援的 Windows AMI，因為 Windows AMI 不支援根磁碟區的執行個體儲存體。

> **Important**
>
> 只有下列執行個體類型支援執行個體儲存體磁碟區做為根磁碟區，且需要 Amazon S3 支援的 AMI：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

AMI 建立程序與 Amazon EBS 後端 AMI 不同。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

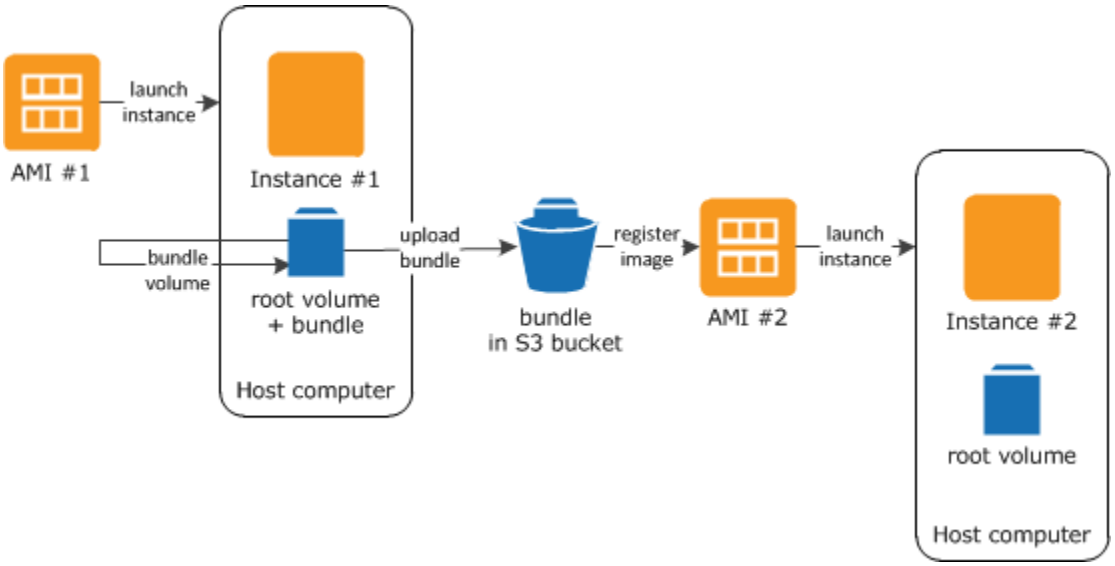
目錄

- AMI 建立概觀
- 先決條件
- 從 Amazon Linux 執行個體建立 AMI
- 設定 Amazon EC2 AMI 工具

- Amazon EC2 AMI 工具參考
- 將 Amazon S3 支援的 AMI 轉換為 EBS 支援的 AMI

AMI 建立概觀

以下圖表顯示透過具有執行個體儲存體根磁碟區的執行個體建立 AMI 之程序的摘要。



首先，請先從與您要建立的 AMI 相似的 AMI 啟動執行個體。您可以連線到您的執行個體並自訂。設定好所要的執行個體後，您可以組合執行個體。完成綁定程序需要幾分鐘。程序完成後，您將擁有 bundle，其中包含映像工作資訊清單檔案 (`image.manifest.xml`) 和含有根磁碟區範本的檔案 (`image.part.xx`)。接著您上傳 bundle 至 Amazon S3 儲存貯體，然後登錄您的 AMI。

> **Note**
>
> 若要為您的 Amazon S3 支援的 Linux AMI 將物件上傳到 S3 儲存貯體，必須為儲存貯體啟用 ACL。否則，Amazon EC2 將無法在要上傳的物件上設定 ACL。如果您的目的地儲存貯體將儲存貯體擁有者強制設定用於 S3 物件擁有權，這將無法運作，因為 ACL 已停用。如需詳細資訊，請參閱控制物件的擁有權並停用儲存貯體的 ACL。

當您使用新的 AMI 啟動執行個體時，我們會使用上傳至 Amazon S3 的 bundle 建立執行個體的根磁碟區。Amazon S3 中 bundle 所用的儲存空間會使您的帳戶產生費用，直到刪除為止。如需詳細資訊，請參閱取消註冊 Amazon EC2 AMI。

若除了根磁碟區之外，您還為您的執行個體新增執行個體儲存體磁碟區，則新 AMI 的區塊型儲存設備映射便會包含這些磁碟區的資訊，並且您從新的 AMI 啟動之執行個體的區塊型儲存設備映射也會自動

包含這些磁碟區的資訊。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射。

先決條件

在您可以開始建立 AMI 之前，務必完成以下工作：

- 安裝 AMI 工具。如需詳細資訊，請參閱設定 Amazon EC2 AMI 工具。
- 安裝 AWS CLI。如需詳細資訊，請參閱 AWS CLI入門。
- 請確定您具有套件組合的 S3 儲存貯體，並確定您的儲存貯體已啟用 ACL。如需有關設定 ACL 的詳細資訊，請參閱設定 ACL。
  - 若要使用 建立 S3 儲存貯體 AWS 管理主控台，請在 https：//https://console.aws.amazon.com/s3/ 開啟 Amazon S3 主控台，然後選擇建立儲存貯體。
  - 若要使用 建立 S3 儲存貯體 AWS CLI，您可以使用 mb 命令。如果安裝的 AMI 工具版本為 1.5.18 或更新版本，也可以使用 `ec2-upload-bundle` 命令建立 S3 儲存貯體。如需詳細資訊，請參閱ec2-upload-bundle。
- 確保綁定的檔案未在 S3 儲存貯體加密。如果您需要對 AMI 進行加密，則可以改用 EBS 支援的 AMI。如需詳細資訊，請參閱對 EBS 後端 AMI 使用加密。
- 請確定您擁有 AWS 帳戶 ID。如需詳細資訊，請參閱《AWS 帳戶管理參考指南》中的檢視 AWS 帳戶 識別符。
- 請確定您擁有可使用 AWS CLI的憑證。如需詳細資訊，請參閱 AWS Command Line Interface 使用者指南中的 AWS CLI的身分驗證與存取憑證。
- 請確定您擁有 X.509 憑證和對應的私有金鑰。
  - 如果需要建立 X.509 憑證，請參閱管理簽署憑證。X.509 憑證和私有金鑰用於加密和解密 AMI。
  - [中國 (北京)] 使用 `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` 憑證。
  - 【AWS GovCloud （美國西部）】 使用`$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem`憑證。
- 連線到您的執行個體並加以自訂。例如，您可以安裝軟體和應用程式、複製資料、刪除暫存檔，以及修改 Linux 組態。

從 Amazon Linux 執行個體建立 AMI

以下程序介紹如何從執行 Amazon Linux 1 的具有執行個體儲存體根磁碟區的執行個體建立 AMI。它們可能不適用於執行其他 Linux 發行版的執行個體。

準備使用 AMI 工具 (僅適用於 HVM 執行個體)

1. AMI 工具需要 GRUB 舊版才能正常開機。請使用下列命令安裝 GRUB：

```
[ec2-user ~]$ sudo yum install -y grub
```

2. 使用下列命令安裝分割區管理套件：

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

如需從具有執行個體儲存體根磁碟區的 Amazon Linux 執行個體建立 AMI

此程序假設您已滿足 先決條件 中的先決條件。

在下列命令中，將每個*########*替換為您自己的資訊。

1. 將您的登入資料上傳到執行個體。這些登入資料是用來確定只有您與 Amazon EC2 可存取您的 AMI。

   a. 在執行個體上為登入資料建立如下的暫時目錄：

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   如此您就能將登入資料從建立的映像中排除。

   b. 使用 scp 等安全複製工具從電腦上將 X.509 憑證和對應的私有金鑰複製到執行個體上的 /tmp/cert 目錄內。以下 -i *my-private-key*.pem 命令中的 scp 選項為用來透過 SSH 連接執行個體的私有金鑰，而非 X.509 私有金鑰。例如：

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
   path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  685     0.7KB/s   00:00
   ```

   或者，由於這些檔案為純文字檔案，因此您可用文字編輯器開啟憑證和金鑰，然後將內容複製到 /tmp/cert 中的新檔案內。

2. 從執行個體內執行 ec2-bundle-vol 命令，準備要上傳至 Amazon S3 的 bundle。務必指定 -e 選項，以排除憑證存放位置的目錄。根據預設，bundle 程序不包含可能帶有敏感資訊的檔案。這些檔案包含 *.sw、*.swo、*.swp、*.pem、*.priv、*id_rsa*、*id_dsa* *.gpg、*.jks、*/.ssh/authorized_keys 和 */.bash_history。若要包含所有這些檔案，請使用 --no-filter 選項。若要包含這些檔案中的一部分，請使用 --include 選項。

   > **Important**
   >
   > 根據預設，AMI 綁定程序會在代表根磁碟區的 /tmp 目錄內建立經過壓縮加密的檔案集合。如果 /tmp 內沒有足夠的可用磁碟空間可存放 bundle，您需要用 -d */path/to/bundle/storage* 選項指定其他的 bundle 存放位置。部分執行個體會有暫時性儲存區掛載於 /mnt 或 /media/ephemeral0 以供您使用，或者您也可以建立、連接及掛載新的 Amazon EBS 磁碟區，以存放 bundle。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的建立 Amazon EBS 磁碟區。

   a. 您必須以 root 執行 ec2-bundle-vol 命令。針對多數命令，您皆能用 sudo 來取得更高級的許可，但在此情況下您應執行 sudo -E su 以保持環境變數。

      ```
      [ec2-user ~]$ sudo -E su
      ```

      請注意，bash 提示現在將您識別為根使用者，且貨幣符號已換成雜湊標籤，表示您正在根 shell 下：

      ```
      [root ec2-user]#
      ```

   b. 若要建立 AMI bundle，請照下列所示執行 ec2-bundle-vol 命令：

      ```
      [root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-
      HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-
      HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --
      partition gpt
      ```

      > **Note**
      >
      > 對於中國（北京）和 AWS GovCloud（美國西部）區域，請使用 --ec2cert 參數，並根據先決條件指定憑證。

建立映像可能需要幾分鐘。完成此命令後，您的 /tmp (或非預設的) 目錄將包含 bundle (image.manifest.xml，加上多個 image.part.*xx* 檔案)。

c. 退出根 shell。

```
[root ec2-user]# exit
```

3. (選用) 若要新增更多執行個體儲存體磁碟區，請在 AMI 的 image.manifest.xml 檔案中編輯區塊型裝置映射。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射。

a. 為您的 image.manifest.xml 檔案建立備份。

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

b. 重新格式化 image.manifest.xml 檔案，以方便讀取和編輯。

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/
image.manifest.xml
```

c. 使用文字編輯器編輯 image.manifest.xml 中的區塊型裝置映射。下列範例顯示 ephemeral1 執行個體存放磁碟區的新項目。

> **Note**
>
> 如需排除檔案清單，請參閱 ec2-bundle-vol。

```
    <block_device_mapping>
      <mapping>
        <virtual>ami</virtual>
        <device>sda</device>
      </mapping>
      <mapping>
        <virtual>ephemeral0</virtual>
        <device>sdb</device>
      </mapping>
      <mapping>
        <virtual>ephemeral1</virtual>
        <device>sdc</device>
      </mapping>
```

```
      <mapping>
        <virtual>root</virtual>
        <device>/dev/sda1</device>
      </mapping>
    </block_device_mapping>
```

d. 儲存 `image.manifest.xml` 檔案並結束文字編輯器。

4. 若要上傳 bundle 至 Amazon S3，請照下列所示執行 ec2-upload-bundle 命令。

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name -
m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

> **Important**
>
> 若要在US East (N. Virginia) 以外的區域登錄 AMI，您必須使用 `--region` 選項指定兩個目標區域，和目標區域中已存在的儲存貯體路徑，或可在目標區域中建立的唯一儲存貯體路徑。

5. (選用) bundle 上傳至 Amazon S3 之後，您可使用下列 `/tmp` 命令將 bundle 從執行個體的 rm 目錄中移除：

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

> **Important**
>
> 如果用 `-d` */path/to/bundle/storage* 選項指定 Step 2 中的路徑，請使用該路徑，而非 `/tmp`。

6. 若要登錄 AMI，請照下列所示執行 register-image 命令。

```
[ec2-user ~]$ aws ec2 register-image --image-location amzn-s3-demo-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --
virtualization-type hvm
```

> **Important**
>
> 如果您先前已指定 ec2-upload-bundle 命令的區域，請在此命令中再次指定該區域。

設定 Amazon EC2 AMI 工具

您可使用 AMI 工具來建立及管理 Amazon S3 支援的 Linux AMI。若要使用工具，您必須在 Linux 執行個體上安裝工具。AMI 工具在不支援 RPM 的 Linux 發行版本上會同時以 RPM 和 .zip 檔案的方式提供。

使用 RPM 設定 AMI 工具

1. 使用 Linux 發行版本的套件管理工具 (例如 yum) 來安裝 Ruby。例如：

```
[ec2-user ~]$ sudo yum install -y ruby
```

2. 使用 wget 或 curl 等工具下載 RPM 檔案。例如：

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. 使用下列命令驗證 RPM 檔案的簽章：

```
[ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
```

上述命令應該會顯示檔案的 SHA1 和 MD5 雜湊為 OK。如果命令顯示雜湊為 NOT OK，請用下列命令檢視檔案的 Header SHA1 和 MD5 雜湊：

```
[ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
```

接著比較檔案的 Header SHA1 和 MD5 雜湊與下列已驗證的 AMI 工具雜湊，以確認檔案的真偽：

- Header SHA1：a1f662d6f25f69871104e6a62187fa4df508f880
- MD5：9faff05258064e2f7909b66142de6782

如果檔案的 Header SHA1 和 MD5 雜湊符合已驗證的 AMI 工具雜湊，請繼續下一步。

4. 使用以下命令安裝 RPM：

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. 使用 ec2-ami-tools-version 命令確認 AMI 工具的安裝。

```
[ec2-user ~]$ ec2-ami-tools-version
```

> **Note**
>
> 如果出現「cannot load such file -- ec2/amitools/version (LoadError)」之類的載入錯誤，請完成下一個步驟，將 AMI 工具的安裝位置加入到 RUBYLIB 路徑。

6. (選用) 如果您在上一步驟中發生錯誤，請將 AMI 工具的安裝位置加入到 RUBYLIB 路徑。

   a. 執行下列命令來判斷要加入的路徑。

   ```
   [ec2-user ~]$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
   /usr/lib/ruby/site_ruby/ec2/amitools/version.rb
   /usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
   ```

   在上例中，先前載入錯誤所缺少的檔案位在 /usr/lib/ruby/site_ruby 和 /usr/lib64/ruby/site_ruby。

   b. 在先前的步驟中將位置加入到 RUBYLIB 路徑。

   ```
   [ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
   ```

   c. 使用 ec2-ami-tools-version 命令確認 AMI 工具的安裝。

   ```
   [ec2-user ~]$ ec2-ami-tools-version
   ```

使用 .zip 檔案設定 AMI 工具

1. 使用 Linux 發行版本的套件管理工具 (例如 apt-get) 來安裝 Ruby 及解壓縮。例如：

   ```
   [ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
   ```

2. 使用 wget 或 curl 等工具下載 .zip 檔案。例如：

   ```
   [ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
   ```

3. 將檔案解壓縮到適當的安裝目錄，例如 /usr/local/ec2。

   ```
   [ec2-user ~]$ sudo mkdir -p /usr/local/ec2
   $ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
   ```

請注意，.zip 檔案包含資料夾 ec2-ami-tools-*x.x.x*，其中 *x.x.x* 是工具的版本號碼 (例如，ec2-ami-tools-1.5.7)。

4. 將 EC2_AMITOOL_HOME 環境變數設定為工具的安裝目錄。例如：

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. 將工具加入到 PATH 環境變數。例如：

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. 使用 ec2-ami-tools-version 命令確認 AMI 工具的安裝。

```
[ec2-user ~]$ ec2-ami-tools-version
```

管理簽署憑證

AMI 工具中的部分命令需要簽署憑證 (亦稱為 X.509 憑證)。您必須建立憑證，然後將其上傳至 AWS。例如，您可用 OpenSSL 等第三方工具來建立憑證。

建立簽署憑證

1. 安裝及設定 OpenSSL。
2. 使用 openssl genrsa 命令建立私有金鑰，並將輸出儲存至 .pem 檔案。建議新建一個 2048 或 4096 位元 RSA 金鑰。

```
openssl genrsa 2048 > private-key.pem
```

3. 使用 openssl req 命令產生憑證。

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -
out certificate.pem
```

若要將憑證上傳至其中 AWS，請使用 upload-signing-certificate 命令。

```
aws iam upload-signing-certificate --user-name user-name --certificate-body
 file://path/to/certificate.pem
```

若要列出使用者的憑證，請用 list-signing-certificates 命令：

```
aws iam list-signing-certificates --user-name user-name
```

若要停用或重新啟用簽署憑證，請用 update-signing-certificate 命令。下列命令可停用憑證：

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --
status Inactive --user-name user-name
```

若要刪除憑證，請用 delete-signing-certificate 命令：

```
aws iam delete-signing-certificate --user-name user-name --certificate-
id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

Amazon EC2 AMI 工具參考

您可使用 AMI 工具命令來建立及管理 Amazon S3 支援的 Linux AMI。欲設定這些工具，請參閱設定 Amazon EC2 AMI 工具。

如需有關存取金鑰的詳細資訊，請參閱「IAM 使用者指南」中的管理 IAM 使用者的存取金鑰。

命令

- ec2-ami-tools-version
- ec2-bundle-image
- ec2-bundle-vol
- ec2-delete-bundle
- ec2-download-bundle
- ec2-migrate-manifest
- ec2-unbundle
- ec2-upload-bundle
- AMI 工具常用選項

ec2-ami-tools-version

說明

描述 AMI 工具的版本。

語法

ec2-ami-tools-version

Output

版本資訊。

範例

此範例命令會顯示您正使用之 AMI 工具的版本資訊。

```
[ec2-user ~]$ ec2-ami-tools-version
1.5.2 20071010
```

ec2-bundle-image

說明

透過迴路檔案中建立的作業系統映像，來建立 Amazon S3 支援的 Linux AMI。

語法

ec2-bundle-image -c *path* -k *path* -u *account* -i *path* [-d *path*] [--ec2cert *path*] [-r *architecture*] [--productcodes *code1*,*code2*,...] [-B *mapping*] [-p *prefix*]

選項

`-c, --cert` path

使用者以 PEM 編碼的 RSA 公有金鑰憑證檔案。

必要：是

`-k, --privatekey` path

指向以 PEM 編碼的 RSA 金鑰檔案路徑。您必須指定此金鑰來解開此 bundle，因此請將其置於安全位置。請注意，金鑰不需要註冊到您的 AWS 帳戶。

必要：是

`-u, --user` account

使用者 AWS 的帳戶 ID，不含破折號。

必要：是

`-i, --image` path

欲綁定的映像路徑。

必要：是

`-d, --destination` path

欲建立 bundle 的目錄。

預設：`/tmp`

必要：否

`--ec2cert` path

用於加密映像資訊清單的 Amazon EC2 X.509 公有金鑰憑證路徑。

`us-gov-west-1` 和 `cn-north-1` 區域使用非預設的公有金鑰憑證，而該憑證的路徑必須以此選項加以指定。憑證的路徑視 AMI 工具的安裝方法而異。以 Amazon Linux 而言，此憑證位於 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。若您從 RPM 或 ZIP 檔案將 AMI 工具安裝進 設定 Amazon EC2 AMI 工具，則憑證位於 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必要：僅適用於 `us-gov-west-1` 和 `cn-north-1` 區域。

`-r, --arch` architecture

映像架構。若您未於命令列提供架構，則會在綁定開始時提示您輸入架構。

有效值：`i386 | x86_64`

必要：否

`--productcodes` code1,code2,...

欲在註冊時連接至映像的產品代碼，以逗號分隔。

必要：否

`-B, --block-device-mapping` mapping

定義區塊型儲存設備向此 AMI 的執行個體公開的方式 (若其執行個體類型支援所指定的設備)。

指定以逗號分隔的索引鍵/值組清單，其中索引鍵為虛擬名稱，而值為相對應的設備名稱。虛擬名稱包括下列：

- ami - 執行個體所看見的根檔案系統設備
- root - 核心所看見的根檔案系統設備
- swap - 執行個體所看見的置換裝置
- ephemeralN - 第 N 個執行個體存放磁碟區

必要：否

-p, --prefix prefix

綁定的 AMI 檔案之檔案名稱字首。

預設：映像檔案名稱。舉例來說，若映像路徑為 /var/spool/my-image/version-2/debian.img，則預設字首為 debian.img。

必要：否

--kernel kernel_id

已廢除. 使用 register-image 來設定核心。

必要：否

--ramdisk ramdisk_id

已廢除. 使用 register-image 來視需要設定 RAM 磁碟。

必要：否

Output

描述綁定程序階段與狀態的狀態訊息。

範例

此範例會從迴路檔案中建立的作業系統映像，建立綁定 AMI。

```
[ec2-user ~]$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
```

```
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

說明

針對執行個體的根磁碟區，透過壓縮、加密並簽署其複本，建立 Amazon S3 支援的 Linux AMI。

Amazon EC2 會嘗試從執行個體繼承產品代碼、核心設定、RAM 磁碟設定和區塊型儲存設備映射。

根據預設，bundle 程序不包含可能帶有敏感資訊的檔案。這些檔案包含 `*.sw`、`*.swo`、`*.swp`、`*.pem`、`*.priv`、`*id_rsa*`、`*id_dsa*` `*.gpg`、`*.jks`、`*/.ssh/authorized_keys` 和 `*/.bash_history`。若要包含所有這些檔案，請使用 `--no-filter` 選項。若要包含這些檔案中的一部分，請使用 `--include` 選項。

如需詳細資訊，請參閱 建立 Amazon S3 支援的 AMI。

語法

ec2-bundle-vol -c *path* -k *path* -u *account* [-d *path*] [--ec2cert *path*] [-r *architecture*] [--productcodes *code1*,*code2*,...] [-B *mapping*] [--all] [-e *directory1*,*directory2*,...] [-i *file1*,*file2*,...] [--no-filter] [-p *prefix*] [-s *size*] [--[no-]inherit] [-v *volume*] [-P *type*] [-S *script*] [--fstab *path*] [--generate-fstab] [--grub-config *path*]

選項

`-c, --cert` path

使用者以 PEM 編碼的 RSA 公有金鑰憑證檔案。

必要：是

`-k, --privatekey` path

使用者以 PEM 編碼的 RSA 金鑰檔案路徑。

必要：是

`-u, --user` account

使用者 AWS 的帳戶 ID，不含破折號。

必要：是

`-d, --destination` destination

欲建立 bundle 的目錄。

預設：`/tmp`

必要：否

`--ec2cert` path

用於加密映像資訊清單的 Amazon EC2 X.509 公有金鑰憑證路徑。

`us-gov-west-1` 和 `cn-north-1` 區域使用非預設的公有金鑰憑證，而該憑證的路徑必須以此選項加以指定。憑證的路徑視 AMI 工具的安裝方法而異。以 Amazon Linux 而言，此憑證位於 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。若您從 RPM 或 ZIP 檔案將 AMI 工具安裝進 設定 Amazon EC2 AMI 工具，則憑證位於 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必要：僅適用於 `us-gov-west-1` 和 `cn-north-1` 區域。

`-r, --arch` architecture

映像架構。若您未於命令列提供架構，則會在綁定開始時提示您提供架構。

有效值：`i386` | `x86_64`

必要：否

--productcodes code1,code2,...

欲在註冊時連接至映像的產品代碼，以逗號分隔。

必要：否

-B, --block-device-mapping mapping

定義區塊型儲存設備向此 AMI 的執行個體公開的方式 (若其執行個體類型支援所指定的設備)。

指定以逗號分隔的索引鍵/值組清單，其中索引鍵為虛擬名稱，而值為相對應的設備名稱。虛擬名稱包括下列：

- ami - 執行個體所看見的根檔案系統設備
- root - 核心所看見的根檔案系統設備
- swap - 執行個體所看見的置換裝置
- ephemeralN - 第 N 個執行個體存放磁碟區

必要：否

-a, --all

綁定所有目錄，包括遠端掛載之檔案系統上的目錄。

必要：否

-e, --exclude directory1,directory2,...

欲從 bundle 操作排除的絕對目錄路徑和檔案清單。此參數會覆寫 --all 選項。指定排除項目時，隨參數列出的目錄和子目錄將不會與磁碟區一同綁定。

必要：否

-i, --include file1,file2,...

欲納入 bundle 操作的檔案清單。因為指定的檔案可能具備敏感資訊，若未指定則會從 AMI 排除。

必要：否

--no-filter

若已指定，我們不會因其可能具備敏感資訊而將其從 AMI 排除。

必要：否

`-p, --prefix` prefix

綁定的 AMI 檔案之檔案名稱字首。

預設：`image`

必要：否

`-s, --size` size

欲建立的映像檔案大小，以 MB (1024 * 1024 位元) 表示。大小上限為 10240 MB。

預設：10240

必要：否

`--[no-]inherit`

指示映像是否繼承執行個體的中繼資料 (預設為繼承)。若啟用 `--inherit` 但無法存取執行個體中繼資料，則綁定會失敗。

必要：否

`-v, --volume` volume

欲從中建立 bundle 的掛載磁碟區絕對路徑。

預設：根目錄 (/)

必要：否

`-P, --partition` type

指示磁碟映像是否使用分割區表格。若未指定分割區表格類型，預設為用於磁碟區父區塊型儲存設備的類型 (若適用)，否則預設為 `gpt`。

有效值：`mbr` | `gpt` | `none`

必要：否

`-S, --script` script

欲在綁定前執行的自訂指令碼。此指令碼必須具備單一引數，亦即磁碟區的掛載點。

必要：否

`--fstab` path

欲綁定至映像的 fstab 路徑。若未指定，Amazon EC2 會綁定 /etc/fstab。

必要：否

`--generate-fstab`

使用 Amazon EC2 提供的 fstab 綁定磁碟區。

必要：否

`--grub-config`

欲綁定至映像的替代 grub 組態檔案路徑。根據預設，`ec2-bundle-vol` 預期所複製映像上應存在 `/boot/grub/menu.lst` 或 `/boot/grub/grub.conf`。此選項可讓您指定替代 grub 組態檔案路徑，並將其複製以覆蓋預設值 (如有)。

必要：否

`--kernel` kernel_id

已廢除. 使用 register-image 來設定核心。

必要：否

`--ramdisk`ramdisk_id

已廢除. 使用 register-image 來視需要設定 RAM 磁碟。

必要：否

Output

描述綁定階段與狀態的狀態訊息。

範例

此範例會壓縮、加密並簽署本機機器根檔案系統的快照，藉此建立綁定的 AMI。

```
[ec2-user ~]$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c
 cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
  Copying / into the image file /mnt/image...
  Excluding:
       sys
       dev/shm
       proc
       dev/pts
       proc/sys/fs/binfmt_misc
```

```
      dev
      media
      mnt
      proc
      sys
      tmp/image
      mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

ec2-delete-bundle

說明

從 Amazon S3 儲存空間刪除指定 bundle。刪除 bundle 之後，即無法從相對應 AMI 啟動執行個體。

語法

ec2-delete-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] [--url *url*] [--region *region*] [--sigv *version*] [-m *path*] [-p *prefix*] [--clear] [--retry] [-y]

選項

`-b, --bucket` bucket

內含綁定 AMI 的 Amazon S3 儲存貯體名稱，其後為選用「/」以分隔路徑字首

必要：是

`-a, --access-key` access_key_id

AWS 存取金鑰 ID。

必要：是

`-s, --secret-key` secret_access_key

AWS 私密存取金鑰。

必要：是

`-t, --delegation-token` token

要傳遞至 AWS 請求的委派字符。如需詳細資訊，請參閱 IAM 使用者指南中的臨時安全憑證。

必要：僅適用您使用暫時安全登入資料時。

預設：`AWS_DELEGATION_TOKEN` 環境變數的值 (如已設定)。

`--region`region

於請求簽章內使用的區域。

預設：`us-east-1`

必要：若使用簽章第 4 版則為必要

`--sigvversion`

簽署請求時使用的簽章版本。

有效值：2 | 4

預設：4

必要：否

`-m, --manifest`path

指向資訊清單檔案的路徑。

必要：您必須指定 `--prefix` 或 `--manifest`。

`-p, --prefix` prefix

綁定的 AMI 檔案名稱字首。請提供完整字首。舉例來說，若字首為 image.img，則使用 `-p image.img` 而非 `-p image`。

必要：您必須指定 --prefix 或 --manifest。

--clear

若 Amazon S3 儲存貯體在刪除指定 bundle 後為空白，則加以刪除。

必要：否

--retry

在所有 Amazon S3 錯誤後自動重試，每次操作至多五次。

必要：否

-y, --yes

自動假設所有提示的答案為 Yes (是)。

必要：否

Output

Amazon EC2 顯示的狀態訊息，會指出刪除程序的階段與狀態。

範例

此範例會自 Amazon S3 刪除一個 bundle。

```
[ec2-user ~]$ ec2-delete-bundle -b amzn-s3-demo-bucket -a your_access_key_id -
s your_secret_access_key
Deleting files:
amzn-s3-demo-bucket/image.manifest.xml
amzn-s3-demo-bucket/image.part.00
amzn-s3-demo-bucket/image.part.01
amzn-s3-demo-bucket/image.part.02
amzn-s3-demo-bucket/image.part.03
amzn-s3-demo-bucket/image.part.04
amzn-s3-demo-bucket/image.part.05
amzn-s3-demo-bucket/image.part.06
Continue? [y/n]
y
Deleted amzn-s3-demo-bucket/image.manifest.xml
Deleted amzn-s3-demo-bucket/image.part.00
Deleted amzn-s3-demo-bucket/image.part.01
```

```
Deleted amzn-s3-demo-bucket/image.part.02
Deleted amzn-s3-demo-bucket/image.part.03
Deleted amzn-s3-demo-bucket/image.part.04
Deleted amzn-s3-demo-bucket/image.part.05
Deleted amzn-s3-demo-bucket/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

說明

從 Amazon S3 儲存體下載指定的 Amazon S3 支援的 Linux AMI。

語法

ec2-download-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* -k *path* [--url *url*] [--region *region*] [--sigv *version*] [-m *file*] [-p *prefix*] [-d *directory*] [--retry]

選項

`-b, --bucket` bucket

bundle 所在的 Amazon S3 儲存貯體名稱，其後為選用「/」以分隔路徑字首。

必要：是

`-a, --access-key` access_key_id

AWS 存取金鑰 ID。

必要：是

`-s, --secret-key` secret_access_key

AWS 私密存取金鑰。

必要：是

`-k, --privatekey` path

用於解密資訊清單的私有金鑰。

必要：是

`--url` url

Amazon S3 服務的 URL。

預設：`https://s3.amazonaws.com/`

必要：否

`--region` region

於請求簽章內使用的區域。

預設：`us-east-1`

必要：若使用簽章第 4 版則為必要

`--sigv` version

簽署請求時使用的簽章版本。

有效值：2 | 4

預設：4

必要：否

`-m, --manifest` file

不含路徑的資訊清單檔案名稱。建議您指定資訊清單 (`-m`) 或字首 (`-p`)。

必要：否

`-p, --prefix` prefix

綁定的 AMI 檔案之檔案名稱字首。

預設：`image`

必要：否

`-d, --directory` directory

所下載 bundle 的儲存目錄。此目錄必須存在。

預設：目前使用中的目錄。

必要：否

`--retry`

在所有 Amazon S3 錯誤後自動重試，每次操作至多五次。

必要：否

Output

會顯示指出下載程序各個階段的狀態訊息。

範例

此範例會建立 `bundled` 目錄 (使用 Linux mkdir 命令) 並從 `amzn-s3-demo-bucket` Amazon S3 儲存貯體下載 bundle。

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundles/bundle_name
 -m image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from amzn-s3-demo-bucket to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.00 ...
Downloaded image.part.00 from amzn-s3-demo-bucket
Downloading part image.part.01 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.01 ...
Downloaded image.part.01 from amzn-s3-demo-bucket
Downloading part image.part.02 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.02 ...
Downloaded image.part.02 from amzn-s3-demo-bucket
Downloading part image.part.03 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.03 ...
Downloaded image.part.03 from amzn-s3-demo-bucket
Downloading part image.part.04 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.04 ...
Downloaded image.part.04 from amzn-s3-demo-bucket
Downloading part image.part.05 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.05 ...
Downloaded image.part.05 from amzn-s3-demo-bucket
Downloading part image.part.06 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.06 ...
Downloaded image.part.06 from amzn-s3-demo-bucket
```

ec2-migrate-manifest

說明

修改 Amazon S3 支援的 Linux AMI (如其憑證、核心和 RAM 磁碟)，以支援不同區域。

語法

```
ec2-migrate-manifest -c path -k path -m path {(-a access_key_id -
s secret_access_key --region region) | (--no-mapping)} [--ec2cert
ec2_cert_path] [--kernel kernel-id] [--ramdisk ramdisk_id]
```

選項

`-c, --cert` path

使用者以 PEM 編碼的 RSA 公有金鑰憑證檔案。

必要：是

`-k, --privatekey` path

使用者以 PEM 編碼的 RSA 金鑰檔案路徑。

必要：是

`--manifest` path

指向資訊清單檔案的路徑。

必要：是

`-a, --access-key` access_key_id

AWS 存取金鑰 ID。

必要：若使用自動映射則為必要。

`-s, --secret-key` secret_access_key

AWS 私密存取金鑰。

必要：若使用自動映射則為必要。

`--region` region

欲在映射檔案內查詢的區域。

必要：若使用自動映射則為必要。

`--no-mapping`

停用核心和 RAM 磁碟的自動映射。

在遷移過程中，Amazon EC2 會將資訊清單檔案內的核心和 RAM 磁碟，取代為專為目的地區域設計的核心與 RAM 磁碟。除非提供 `--no-mapping` 參數，否則 `ec2-migrate-bundle` 可能會使用 `DescribeRegions` 和 `DescribeImages` 操作來執行自動映射。

必要：若您未提供用於自動映射的 `-a`、`-s` 和 `--region` 選項則為必要。

`--ec2cert` path

用於加密映像資訊清單的 Amazon EC2 X.509 公有金鑰憑證路徑。

`us-gov-west-1` 和 `cn-north-1` 區域使用非預設的公有金鑰憑證，而該憑證的路徑必須以此選項加以指定。憑證的路徑視 AMI 工具的安裝方法而異。以 Amazon Linux 而言，此憑證位於 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。若您從 ZIP 檔案將 AMI 工具安裝進 設定 Amazon EC2 AMI 工具，則憑證位於 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必要：僅適用於 `us-gov-west-1` 和 `cn-north-1` 區域。

`--kernel` kernel_id

欲選取的核心 ID。

> **Important**
>
> 建議您使用 PV-GRUB，而不要使用核心與 RAM 磁碟。如需詳細資訊，請參閱「Amazon Linux 2 使用者指南」中的使用者提供的核心。

必要：否

`--ramdisk` ramdisk_id

欲選取的 RAM 磁碟 ID。

> **Important**
>
> 建議您使用 PV-GRUB，而不要使用核心與 RAM 磁碟。如需詳細資訊，請參閱「Amazon Linux 2 使用者指南」中的使用者提供的核心。

必要：否

Output

描述綁定程序階段與狀態的狀態訊息。

範例

此範例會將 my-ami.manifest.xml 資訊清單中指定的 AMI，從美國複製至歐洲。

```
[ec2-user ~]$ ec2-migrate-manifest --manifest my-ami.manifest.xml
 --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

ec2-unbundle

說明

透過 Amazon S3 支援的 Linux AMI 重新建立 bundle。

語法

ec2-unbundle -k *path* -m *path* [-s *source_directory*] [-d *destination_directory*]

選項

-k, --privatekey path

指向您以 PEM 編碼的 RSA 金鑰檔案路徑。

必要：是

-m, --manifest path

指向資訊清單檔案的路徑。

必要：是

-s, --source source_directory

內含 bundle 的目錄。

預設：目前的目錄。

必要：否

`-d, --destination` destination_directory

欲從中解開 AMI 的目錄。此目的地目錄必須存在。

預設：目前的目錄。

必要：否

範例

此 Linux 和 UNIX 範例會解開 `image.manifest.xml` 檔案內指定的 AMI。

```
[ec2-user ~]$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

Output

會顯示指出解開程序各個階段的狀態訊息。

ec2-upload-bundle

說明

將 Amazon S3 支援的 Linux AMI 的套件組合上傳至 Amazon S3，並在所上傳的物件上設定適當的存取控制清單 (ACL)。如需詳細資訊，請參閱建立 Amazon S3 支援的 AMI。

> **Note**
>
> 若要為您的 Amazon S3 支援的 Linux AMI 將物件上傳到 S3 儲存貯體，必須為儲存貯體啟用 ACL。否則，Amazon EC2 將無法在要上傳的物件上設定 ACL。如果您的目的地儲存貯體將儲存貯體擁有者強制設定用於 S3 物件擁有權，這將無法運作，因為 ACL 已停用。如需詳細資訊，請參閱控制物件的擁有權並停用儲存貯體的 ACL。

語法

ec2-upload-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] -m *path* [--url *url*] [--region *region*] [--sigv *version*] [--acl *acl*] [-d *directory*] [--part *part*] [--retry] [--skipmanifest]

選項

`-b, --bucket` bucket

存放 bundle 的 Amazon S3 儲存貯體名稱，其後為選用「/」以分隔路徑字首。若該儲存貯體不存在，則會加以建立 (如該儲存貯體名稱為可用)。此外，如果儲存貯體不存在且 AMI 工具版本為 1.5.18 或更新版本，則此命令會設定儲存貯體的 ACL。

必要：是

`-a, --access-key` access_key_id

您的 AWS 存取金鑰 ID。

必要：是

`-s, --secret-key` secret_access_key

您的 AWS 私密存取金鑰。

必要：是

`-t, --delegation-token` token

要傳遞至 AWS 請求的委派字符。如需詳細資訊，請參閱 IAM 使用者指南中的臨時安全憑證。

必要：僅適用您使用暫時安全登入資料時。

預設：`AWS_DELEGATION_TOKEN` 環境變數的值 (如已設定)。

`-m, --manifest` path

指向資訊清單檔案的路徑。資訊清單檔案會於綁定程序期間建立，位於內含 bundle 的目錄。

必要：是

`--url` url

已廢除. 改用 `--region` 選項，除非您的儲存貯體被限制在 EU 位置 (而非 `eu-west-1`)。`--location` 旗標是確定該特定位置限制的唯一方法。

Amazon S3 端點服務 URL。

預設：https://s3.amazonaws.com/

必要：否

--region region

要在請求簽章內針對目的地 S3 儲存貯體使用的區域。

- 若該儲存貯體不存在且您未指定區域，此工具會建立不具位置限制的儲存貯體 (位於us-east-1)。
- 若該儲存貯體不存在且您已指定區域，此工具會在該區域建立儲存貯體。
- 若該儲存貯體存在但您未指定區域，此工具會使用該儲存貯體的位置。
- 若該儲存貯體存在且您指定 us-east-1 為區域，此工具會使用該儲存貯體的實際位置，不會出現錯誤訊息，而且會覆寫所有現有的相符檔案。
- 若該儲存貯體存在且您指定的區域 (us-east-1 之外) 與儲存貯體的實際位置不相符，此工具會退出並顯示錯誤。

若您的儲存貯體被限制在 EU 位置 (而非 eu-west-1)，請改用 --location 旗標。--location 旗標是確定該特定位置限制的唯一方法。

預設：us-east-1

必要：若使用簽章第 4 版則為必要

--sigv version

簽署請求時使用的簽章版本。

有效值：2 | 4

預設：4

必要：否

--acl acl

綁定映像的存取控制清單政策。

有效值：public-read | aws-exec-read

預設：aws-exec-read

必要：否

-d, --directory directory

內含綁定 AMI 組件的目錄。

預設：內含資訊清單檔案的目錄 (請參閱 -m 選項)。

必要：否

--part part

開始上傳指定組件及所有後續組件。例如 --part 04。

必要：否

--retry

在所有 Amazon S3 錯誤後自動重試，每次操作至多五次。

必要：否

--skipmanifest

不會上傳資訊清單。

必要：否

--location location

已廢除. 改用 --region 選項，除非您的儲存貯體被限制在 EU 位置 (而非 eu-west-1)。--location 旗標是確定該特定位置限制的唯一方法。

目的地 Amazon S3儲存貯體的位置限制。若該儲存貯體存在且您指定的位置與儲存貯體的實際位置不相符，此工具會退出並顯示錯誤。若該儲存貯體存在但您未指定位置，此工具會使用該儲存貯體的位置。若該儲存貯體不存在且您已指定位置，此工具會在該位置建立儲存貯體。若該儲存貯體不存在且您未指定位置，此工具會建立不具位置限制的儲存貯體 (位於 us-east-1)。

預設：若已指定 --region，則位置會設定為該指定區域。若未指定 --region，則位置預設為 us-east-1。

必要：否

Output

Amazon EC2 顯示的狀態訊息，會指出上傳程序的階段與狀態。

範例

此範例會上傳 `image.manifest.xml` 資訊清單指定的 bundle。

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundles/bundle_name -m
 image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket amzn-s3-demo-bucket ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

AMI 工具常用選項

多數 AMI 工具接受下列選用參數。

`--help, -h`

顯示協助訊息。

`--version`

顯示版本及著作權聲明。

`--manual`

顯示手動輸入。

`--batch`

以批次模式執行，不使用互動式提示。

`--debug`

顯示可用於故障排除問題的資訊。

將 Amazon S3 支援的 AMI 轉換為 EBS 支援的 AMI

您可將擁有的 Amazon S3 支援的 Linux AMI 轉換為 Amazon EBS 支援的 Linux AMI。

> **Important**
>
> 您無法轉換不屬於您的 AMI。

如需將 Amazon S3 支援的 AMI 轉換為 Amazon EBS 支援的 AMI

1. 從 Amazon EBS 後端 AMI 啟動 Amazon Linux 執行個體。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。Amazon Linux 執行個體已預先安裝 AWS CLI 和 AMI 工具。
2. 請將您用來組合 Amazon S3 支援的 AMI 的 X.509 私有金鑰上傳至您的執行個體。此金鑰是用來確定只有您與 Amazon EC2 可存取您的 AMI。

   a. 在執行個體上為 X.509 私有金鑰建立如下的暫時目錄：

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   b. 使用 scp 等安全複製工具從電腦上將 X.509 私有金鑰複製到執行個體上的 `/tmp/cert` 目錄內。以下命令中的 *`my-private-key`* 參數為用來透過 SSH 連接執行個體的私有金鑰。例如：

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   ```

3. 設定環境變數以使用 AWS CLI。如需詳細資訊，請參閱環境變數。

   a. （建議）為您的 AWS 存取金鑰、私密金鑰和工作階段字符設定環境變數。

   ```
   [ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
   [ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
   ```

```
[ec2-user ~]$ export AWS_SESSION_TOKEN=your_session_token
```

b. 為您的 AWS 存取金鑰和私密金鑰設定環境變數。

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

4. 為新的 AMI 準備 Amazon Elastic Block Store (Amazon EBS) 磁碟區。

a. 使用 create-volume 命令在與您執行個體相同的可用區域中建立空的 EBS 磁碟區。記下命令輸出中的磁碟區 ID。

> **Important**
>
> 此 EBS 磁碟區的大小必須等於或大於原本的執行個體儲存體根磁碟區。

```
aws ec2 create-volume \
    --size 10 \
    --region us-west-2 \
    --availability-zone us-west-2b
```

b. 使用 attach-volume 命令將磁碟區連接到您的 Amazon EBS 後端執行個體。

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --region us-west-2
```

5. 建立您的 bundle 資料夾。

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. 使用 `/tmp/bundle` 命令將執行個體後端 AMI 的 bundle 下載至 ec2-download-bundle。

```
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name
 -m image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --
privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. 使用 ec2-unbundle 命令從 bundle 重新建構映像檔。

a. 將目錄變更為 bundle 資料夾。

```
[ec2-user ~]$ cd /tmp/bundle/
```

b. 執行 ec2-unbundle 命令。

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. 從解開映像將檔案複製到新的 EBS 磁碟區。

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. 探測磁碟區中是否有任何解開的新分割區。

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

10. 列出區塊型儲存設備，以找出要掛載的設備名稱。

```
[ec2-user bundle]$ lsblk
NAME         MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/sda    202:0    0   8G  0 disk
##/dev/sda1 202:1    0   8G  0 part /
/dev/sdb    202:80   0  10G  0 disk
##/dev/sdb1 202:81   0  10G  0 part
```

在此範例中，要掛載的分割區為 /dev/sdb1，但您的設備名稱可能不同。如果磁碟區尚未分割，要掛載的設備會類似 /dev/sdb (沒有設備分割區尾碼)。

11. 為新的 EBS 磁碟區建立掛載點，然後掛載磁碟區。

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. 用您偏好的文字編輯器 (例如 /etc/fstab 或 vim) 在 EBS 磁碟區上開啟 nano 檔案，並移除執行個體 (暫時性) 存放磁碟區的任何項目。EBS 磁碟區掛載於 /mnt/ebs，因此 fstab 檔案位於 /mnt/ebs/etc/fstab。

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/     /           ext4    defaults,noatime  1   1
```

```
tmpfs       /dev/shm    tmpfs   defaults        0   0
devpts      /dev/pts    devpts  gid=5,mode=620  0   0
sysfs       /sys        sysfs   defaults        0   0
proc        /proc       proc    defaults        0   0
/dev/sdb        /media/ephemeral0       auto    defaults,comment=cloudconfig    0
     2
```

在此範例中，應該移除最後一行。

13. 卸載磁碟區，並從執行個體分離。

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id vol-01234567890abcdef --
region us-west-2
```

14. 從新的 EBS 磁碟區建立 AMI，如下所示。

    a. 建立新 EBS 磁碟區的快照。

    ```
    [ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
     "your_snapshot_description" --volume-id vol-01234567890abcdef
    ```

    b. 檢查快照是否完整。

    ```
    [ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
    id snap-0abcdef1234567890
    ```

    c. 使用 describe-images 命令找出原始 AMI 的處理器架構、虛擬化類型與核心映像 (aki)。在此步驟中，您需要擁有原始 Amazon S3 支援的 AMI 的 AMI ID。

    ```
    [ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-
    id ami-0abcdef1234567890 --output text
    IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon
     available public machine aki-fc8f11cc instance-store paravirtual xen
    ```

    在此範例中，其架構為 x86_64，核心映像 ID 為 aki-fc8f11cc。請將這些數值使用在下列步驟。如果上述命令的輸出也列出 ari ID，請將其記下。

    d. 使用新 EBS 磁碟區的快照 ID 和先前步驟中的數值登錄您的新 AMI。如果先前的命令輸出列出 ari ID，請用 --ramdisk-id *ari_id* 將其加入到下列命令。

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --
name your_new_ami_name --block-device-mappings DeviceName=device-
name,Ebs={SnapshotId=snap-0abcdef1234567890} --virtualization-type paravirtual
 --architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

15. (選用) 確認可以從新 AMI 啟動執行個體後，您便能刪除為此程序建立的 EBS 磁碟區。

```
aws ec2 delete-volume --volume-id vol-01234567890abcdef
```

使用 Windows Sysprep 建立 Amazon EC2 AMI

Microsoft 系統準備 (Windows Sysprep) 工具建立作業系統的通用版本，並在擷取新影像之前刪除特定於執行個體的系統組態。

我們建議您使用 EC2 Image Builder 來自動建立、管理和部署自訂、安全且最新的「最佳」伺服器映像，這些映像會預先安裝並預先設定為軟體和設定。

您也可以使用 Windows Sysprep 透過 Windows 啟動代理程式來建立標準化 AMI：EC2Launch v2、EC2Launch 和 EC2Config。

> **Important**
>
> 請不要使用 Windows Sysprep 來建立執行個體備份。Windows Sysprep 會移除系統特定資訊；移除此資訊可能會在執行個體備份時發生意料之外的結果。

若要疑難排解 Windows Sysprep，請參閱 針對 Amazon EC2 Windows 執行個體的 Sysprep 問題進行疑難排解。

目錄

- Windows Sysprep 階段
- 開始之前
- 使用 Windows Sysprep 搭配 EC2Launch v2 來建立 AMI
- 使用 Windows Sysprep 搭配 EC2Launch 建立 AMI
- 使用 Windows Sysprep 搭配 EC2Config 建立 AMI

Windows Sysprep 階段

Windows Sysprep 會執行下列階段：

- 一般化：此 Sysprep 工具會移除映像特定資訊和組態。例如，Windows Sysprep 會移除安全識別符 (SID)、電腦名稱、事件日誌和特定驅動程式等。完成本階段之後，作業系統 (OS) 即準備好建立 AMI。

  > **Note**
  >
  > 當您使用 Windows 啟動代理程式執行 Windows Sysprep 時，系統會防止驅動程式被移除，因為 `PersistAllDeviceInstalls` 設定預設為 true。

- 特製化：隨插即用會掃描電腦，並為任何偵測到的裝置安裝驅動程式。此 Sysprep 工具會產生電腦名稱和 SID 等 OS 要求。或者，您可以在此階段執行命令。
- 立即可用體驗 (OOBE)：系統執行縮簡版的 Windows 安裝程式，並要求您輸入系統語言、時區和已註冊的組織這類資訊。當您使用 Windows 啟動代理程式執行 Windows Sysprep 時，回應檔案會自動執行此階段。

開始之前

- 執行 Windows Sysprep 之前，除了用於執行 Windows Sysprep 的單一管理員帳戶外，建議您移除其他所有本機使用者帳戶和所有帳戶描述檔。如果您以其他帳戶和描述檔執行 Windows Sysprep，可能會發生非預期的行為，包括遺失描述檔資料或無法完成 Windows Sysprep。
- 了解有關 Sysprep 概觀的更多資訊。
- 了解 Sysprep 支援的伺服器角色。

使用 Windows Sysprep 搭配 EC2Launch v2 來建立 AMI

若透過已安裝 EC2Launch v2 代理程式的執行個體建立映像，EC2Launch v2 會在映像就緒時執行特定任務。這包括搭配使用 Windows Sysprep。如需詳細資訊，請參閱the section called “Windows Sysprep 階段”。

目錄

- Windows Sysprep 操作
- Sysprep 後續階段

- 使用 EC2Launch v2 執行 Windows Sysprep

Windows Sysprep 操作

Windows Sysprep 和 EC2Launch v2 會在準備映像時執行下列動作。

1. 當您在 EC2Launch settings (EC2Launch 設定) 對話方塊中選擇 Shutdown with Sysprep (使用 Sysprep 關機) 時，系統會執行 `ec2launch sysprep` 命令。
2. EC2Launch v2 透過讀取位於 `unattend.xml` 的登錄值來編輯 `HKEY_USERS\.DEFAULT\Control Panel\International\LocaleName` 檔案的內容。檔案位於以下目錄：`C:\ProgramData\Amazon\EC2Launch\sysprep`。
3. 系統會執行 `BeforeSysprep.cmd`。此命令會建立一個登錄機碼，如下所示：

   ```
   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f
   ```

   除非重新啟用 RDP 連接，否則此登錄機碼會停用連線。停用 RDP 連線是必要的安全措施，因為在執行 Windows Sysprep 之後的第一個開機工作階段期間，RDP 會有一小段時間允許連接，而且管理員密碼會空白。
4. EC2Launch v2 服務會執行下列命令來呼叫 Windows Sysprep：

   ```
   sysprep.exe /oobe /generalize /shutdown /unattend: "C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml"
   ```

一般化階段

- EC2Launch v2 會移除映像特定資訊和組態 (例如電腦名稱和 SID)。如果執行個體是網域成員，即會從網域中移除。`unattend.xml` 回應檔案包括下列可影響此階段的設定：
  - PersistAllDeviceInstalls：此設定可防止 Windows 安裝程式移除和重新設定裝置，以加速映像準備程序，因為 Amazon AMI 需要執行特定驅動程式，而且重新偵測這些驅動程式需要一些時間。
  - DoNotCleanUpNonPresentDevices：此設定會保留目前不存在之裝置的隨插即用資訊。
- Windows Sysprep 會在準備建立 AMI 時關閉 OS。系統會啟動新的執行個體，或啟動原始執行個體。

特製化階段

系統會產生電腦名稱和 SID 這類 OS 特定要求。系統也會根據您在 unattend.xml 回應檔案中指定的組態，來執行下列動作。

- CopyProfile：Windows Sysprep 可以設定為刪除所有使用者描述檔 (包括內建管理員描述檔)。此設定會保留內建管理員帳戶，因此您對該帳戶進行的任何自訂都會帶到新的映像。預設值為 True。

  CopyProfile 會以現有的本機管理員描述檔取代預設描述檔。執行 Windows Sysprep 之後登入的所有帳戶，將於第一次登入時收到該描述檔的複本及其內容。

  如果您沒有特定的使用者描述檔自訂要帶到新映像，請將此設定變更為 False。Windows Sysprep 將移除所有使用者描述檔 (這樣可節省時間和磁碟空間)。
- TimeZone：時區會預設為國際標準時間 (UTC)。
- 順序為 1 的同步命令：系統會執行下列命令，以啟用管理員帳戶並指定密碼要求。

  ```
  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /
  PASSWORDREQ:YES
  ```
- 順序為 2 的同步命令：系統會將管理員密碼變碼。此安全措施的用途在於您未設定 setAdminAccount 任務時，可防止在 Windows Sysprep 完成之後存取執行個體。

  系統從本機啟動代理程式目錄執行以下命令 (C:\Program Files\Amazon\EC2Launch\)。

  ```
  EC2Launch.exe internal randomize-password --username Administrator
  ```
- 若要啟用遠端桌面連接，系統將終端伺服器 fDenyTSConnections 登錄機碼設定為 false。

OOBE 階段

1. 系統使用 EC2Launch v2 回應檔案指定下列組態：
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
   - <ProtectYourPC>3</ProtectYourPC>

- `<BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>`
- `<TimeZone>UTC</TimeZone>`
- `<RegisteredOrganization>Amazon.com</RegisteredOrganization>`
- `<RegisteredOwner>EC2</RegisteredOwner>`

> **Note**
>
> 在一般化和特製化階段期間，EC2Launch v2 會監控 OS 的狀態。如果 EC2Launch v2 偵測到 OS 處於 Sysprep 階段，即會將下列訊息發佈至系統日誌：
> Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. 系統會執行 EC2Launch v2。

Sysprep 後續階段

Windows Sysprep 完成之後，EC2Launch v2 會將下列訊息傳送至主控台輸出：

```
Windows sysprep configuration complete.
```

EC2Launch v2 接著會執行下列動作：

1. 讀取 `agent-config.yml` 檔案的內容並執行已設定的任務。
2. 執行 `preReady` 階段中的所有任務。
3. 完成後，會將 `Windows is ready` 訊息傳送至執行個體系統日誌。
4. 執行 `PostReady` 階段中的所有任務。

如需 EC2Launch v2 的詳細資訊，請參閱在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v2 代理程式執行任務。

使用 EC2Launch v2 執行 Windows Sysprep

請使用下列程序，使用 Windows Sysprep 搭配 EC2Launch v2 來建立標準化 AMI。

1. 在 Amazon EC2 主控台中，找出您要複製的 AMI。
2. 啟動和連線至 Windows 執行個體。
3. 自訂設定

a. 在 Windows 的開始功能表中，搜尋並選擇 Amazon EC2Launch 設定。如需有關 Amazon EC2Launch 設定對話方塊中選項和設定的詳細資訊，請參閱設定 Windows 執行個體的 EC2Launch v2 設定。

b. 若已做出變更，選擇儲存後再關閉。

4. 選擇透過 Sysprep 關閉或不透過 Sysprep 關閉。

當系統要求您確認是否要執行 Windows Sysprep 並關閉執行個體時，請選擇是。EC2Launch v2 執行 Windows Sysprep。接著，系統會將您登出執行個體，然後執行個體會關機。如果您檢查 Amazon EC2 主控台中的 Instances (執行個體) 頁面，執行個體狀態會從 `Running` 變更為 `Stopping`，再變更為 `Stopped`。這時便能放心地從此執行個體建立 AMI。

您可以從命令列使用下列命令手動調用 Windows Sysprep 工具：

```
"%programfiles%\amazon\ec2launch\ec2launch.exe" sysprep --shutdown=true
```

使用 Windows Sysprep 搭配 EC2Launch 建立 AMI

若透過已安裝 EC2Launch 代理程式的執行個體建立映像，EC2Launch 會在映像就緒時執行特定任務。這包括搭配使用 Windows Sysprep。如需詳細資訊，請參閱the section called “Windows Sysprep 階段”。

EC2Launch 會為 Windows Sysprep 提供預設回應檔案和批次檔案，這可自動化和保護您 AMI 上的映像準備程序。您可選擇性修改這些檔案。這些檔案預設位於下列目錄：`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep`。

> **Important**
>
> 請不要使用 Windows Sysprep 來建立執行個體備份。Windows Sysprep 會移除系統專屬資訊。如果您移除此資訊，執行個體備份可能產生意外結果。

目錄

- Windows Sysprep 的 EC2Launch 回應和批次檔案
- 使用 EC2Launch 執行 Windows Sysprep
- 啟動自訂 AMI 時更新 Server 2016 和更新版本的中繼資料/KMS 路由

Windows Sysprep 的 EC2Launch 回應和批次檔案

Windows Sysprep 的 EC2Launch 回應檔案和批次檔案包含下列內容：

`Unattend.xml`

這是預設回應檔案。如果您執行 `SysprepInstance.ps1`，或在使用者介面中選擇 ShutdownWithSysprep，系統會從這個檔案讀取設定。

`BeforeSysprep.cmd`

自訂此批次檔案可在 EC2Launch 執行 Windows Sysprep 前執行命令。

`SysprepSpecialize.cmd`

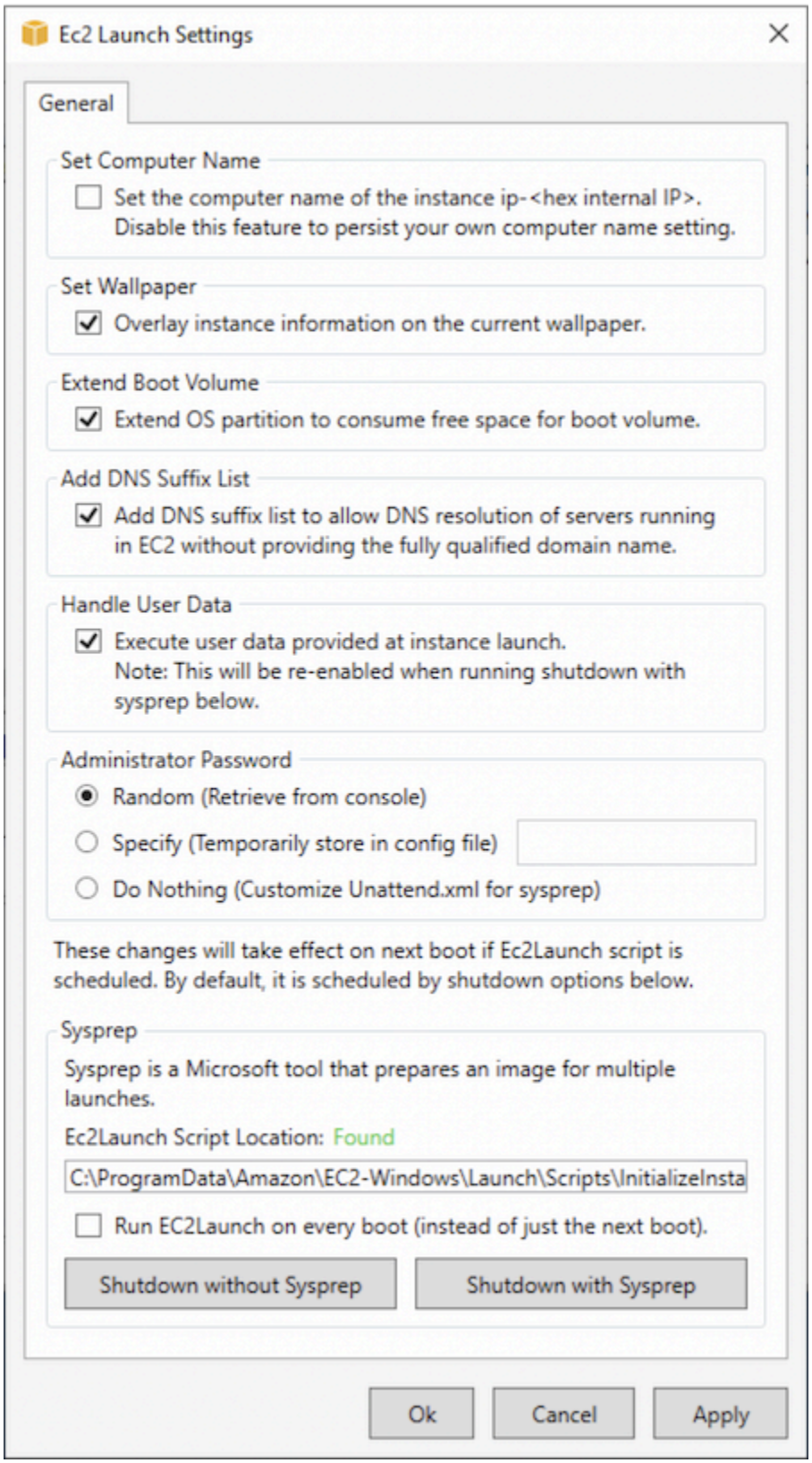
自訂此批次檔案可在 Windows Sysprep 特製化階段期間執行命令。

使用 EC2Launch 執行 Windows Sysprep

完整安裝 Windows Server 2016 和更新版本 (包含桌面體驗) 後，您可以使用 EC2Launch 手動執行 Windows Sysprep，或使用 EC2 Launch 設定應用程式執行 Sysprep。

使用 EC2Launch 設定應用程式執行 Windows Sysprep

1. 在 Amazon EC2 主控台中，尋找或建立 Windows Server 2016 或更新版本 AMI。
2. 從 AMI 啟動 Windows 執行個體。
3. 連線到您的 Windows 執行個體並加以自訂。
4. 搜尋並執行 EC2LaunchSettings 應用程式。根據預設，其位於下列目錄：`C:\ProgramData\Amazon\EC2-Windows\Launch\Settings`。



5. 視需要選取或清除選項。這些設定存放在 `LaunchConfig.json` 檔案中。
6. 針對 Administrator Password (管理員密碼)，請執行下列其中一項作業：

- 選擇 Random (隨機)。EC2Launch 會產生密碼並以使用者金鑰將其加密。系統會在執行個體啟動後停用此設定，以便執行個體重新開機或停止再啟動時，能持續使用這個密碼。
- 選擇 Specify (指定) 並輸入符合系統要求的密碼。系統會將該密碼以純文字存放於 `LaunchConfig.json`，並在 Windows Sysprep 設定管理員密碼後予以刪除。如果您立即關機，即會立即設定密碼。EC2Launch 會以使用者金鑰加密該密碼。
- 選擇 DoNothing (不執行動作)，並在 `unattend.xml` 檔案中指定密碼。如果您未在 `unattend.xml` 中指定密碼，會停用管理員帳戶。

7. 選擇 Shutdown with Sysprep (使用 Sysprep 關機)。

使用 EC2Launch 手動執行 Windows Sysprep

1. 在 Amazon EC2 主控台中尋找或建立您想要複製的 Windows Server 2016 或更新版本 Datacenter 版 AMI。
2. 啟動和連線至 Windows 執行個體。
3. 自訂執行個體。
4. 在 `LaunchConfig.json` 檔案中指定設定。根據預設，此檔案位於 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。

   針對 `adminPasswordType`，請指定下列其中一個值：

   `Random`

   EC2Launch 會產生密碼並以使用者金鑰將其加密。系統會在執行個體啟動後停用此設定，以便執行個體重新開機或停止再啟動時，能持續使用這個密碼。

   `Specify`

   EC2Launch 會使用您在 `adminPassword` 中指定的密碼。如果該密碼不符合系統要求，EC2Launch 會改為產生隨機密碼。系統會將該密碼以純文字存放於 `LaunchConfig.json`，並在 Windows Sysprep 設定管理員密碼後予以刪除。EC2Launch 會以使用者金鑰加密該密碼。

   `DoNothing`

   EC2Launch 會使用您在 `unattend.xml` 檔案中指定的密碼。如果您未在 `unattend.xml` 中指定密碼，會停用管理員帳戶。

5. (選用) 請在 unattend.xml 和其他組態檔案中指定設定。如果您計畫安裝，就不需要變更這些檔案。根據預設，檔案位於下列目錄：C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep。
6. 在 Windows PowerShell 中，執行 ./InitializeInstance.ps1 -Schedule。根據預設，指令碼位於下列目錄：C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts。此指令碼會排程執行個體在下一次開機期間初始化。您必須在執行下一步的 SysprepInstance.ps1 指令碼前，執行此指令碼。
7. 在 Windows PowerShell 中，執行 ./SysprepInstance.ps1。根據預設，指令碼位於下列目錄：C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts。

系統會將您登出執行個體，然後執行個體會關機。如果您檢查 Amazon EC2 主控台中的 Instances (執行個體) 頁面，執行個體狀態會從 Running 變更為 Stopping，然後變更為 Stopped。這時便能放心地從這個執行個體建立 AMI。

啟動自訂 AMI 時更新 Server 2016 和更新版本的中繼資料/KMS 路由

若要在啟動自訂 AMI 時更新 Server 2016 和更新版本的中繼資料/KMS 路由，請執行下列其中一項動作：

- 執行 EC2LaunchSettings GUI (C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe)，然後選取選項以搭配 Windows Sysprep 來關機。
- 在建立 AMI 之前，請執行 EC2LaunchSettings，且不搭配 Windows Sysprep 來關機。這樣會將 EC2 啟動初始化任務設為在下一次開機時執行，如此將會根據執行個體的子網來設定路由。
- 在從 PowerShell 建立 AMI 之前，手動重新排程 EC2 啟動初始化任務。

> **Important**
>
> 在重新排定任務之前，請記下預設密碼重設行為。

- 若要在發生 Windows 啟用失敗或與執行個體中繼資料通訊失敗的執行中的執行個體上更新路由，請參閱「無法啟用 Windows」。

使用 Windows Sysprep 搭配 EC2Config 建立 AMI

若透過已安裝 EC2Config 服務的執行個體建立映像，EC2Config 會在映像就緒時執行特定任務。這包括搭配使用 Windows Sysprep。如需詳細資訊，請參閱the section called “Windows Sysprep 階段”。

目錄

- Windows Sysprep 操作
- Sysprep 後續階段
- 搭配 EC2Config 服務執行 Windows Sysprep

Windows Sysprep 操作

Windows Sysprep 和 EC2Config 服務會在準備映像時執行下列動作。

1. 當您在 EC2 服務屬性對話方塊中選擇使用 Sysprep 關機時，系統會執行 ec2config.exe -sysprep 命令。
2. EC2Config 服務會讀取 `BundleConfig.xml` 檔案的內容。此檔案預設位在下列目錄：`C:\Program Files\Amazon\Ec2ConfigService\Settings`。

   `BundleConfig.xml` 檔案包括下列設定。您可以變更這些設定：

   - AutoSysprep：指出是否自動使用 Windows Sysprep。如果您要從 EC2 服務屬性對話方塊中執行 Windows Sysprep，則不需要變更此值。預設值為 No。
   - SetRDPCertificate：設定遠端桌面伺服器的自簽憑證。這可讓您安全地使用遠端桌面通訊協定 (RDP) 連線至執行個體。如果新的執行個體應該使用憑證，請將值變更為 Yes。此設定不適用於 Windows Server 2012 執行個體，因為這些作業系統可以產生自己的憑證。預設值為 No。
   - SetPasswordAfterSysprep：在新啟動的執行個體上設定隨機密碼，並以使用者啟動金鑰加密，然後將加密的密碼輸出到主控台。如果新執行個體不應該設定為隨機加密密碼，請將值變更為 No。預設值為 Yes。
   - PreSysprepRunCmd：要執行之命令的位置。根據預設，命令位於下列目錄：`C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd`。
3. 系統會執行 `BeforeSysprep.cmd`。此命令會建立一個登錄機碼，如下所示：

   ```
   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
    fDenyTSConnections /t REG_DWORD /d 1 /f
   ```

   除非重新啟用 RDP 連接，否則此登錄機碼會停用連線。停用 RDP 連線是必要的安全措施，因為在執行 Windows Sysprep 之後的第一個開機工作階段期間，RDP 會有一小段時間允許連接，而且管理員密碼會空白。
4. EC2Config 服務會執行下列命令來呼叫 Windows Sysprep：

```
sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /
oobe /generalize /shutdown
```

一般化階段

- 此工具會移除映像特定資訊和組態 (例如電腦名稱和 SID)。如果執行個體是網域成員，即會從網域中移除。`sysprep2008.xml` 回應檔案包括下列可影響此階段的設定：
  - PersistAllDeviceInstalls：此設定可防止 Windows 安裝程式移除和重新設定裝置，以加速映像準備程序，因為 Amazon AMI 需要執行特定驅動程式，而且重新偵測這些驅動程式需要一些時間。
  - DoNotCleanUpNonPresentDevices：此設定會保留目前不存在之裝置的隨插即用資訊。
- Windows Sysprep 會在準備建立 AMI 時關閉 OS。系統會啟動新的執行個體，或啟動原始執行個體。

特製化階段

系統會產生電腦名稱和 SID 這類 OS 特定要求。系統也會根據您在 sysprep2008.xml 回應檔案中指定的組態，來執行下列動作。

- CopyProfile：Windows Sysprep 可以設定為刪除所有使用者描述檔 (包括內建管理員描述檔)。此設定會保留內建管理員帳戶，因此您對該帳戶進行的任何自訂都會帶到新的映像。預設值為 True。

  CopyProfile 會以現有的本機管理員描述檔取代預設描述檔。執行 Windows Sysprep 之後登入的所有帳戶，將於第一次登入時收到該描述檔的複本及其內容。

  如果您沒有特定的使用者描述檔自訂要帶到新映像，請將此設定變更為 False。Windows Sysprep 將移除所有使用者描述檔；這樣可節省時間和磁碟空間。
- TimeZone：時區會預設為國際標準時間 (UTC)。
- 順序為 1 的同步命令：系統會執行下列命令，以啟用管理員帳戶並指定密碼要求。

  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /
  PASSWORDREQ:YES
- 順序為 2 的同步命令：系統會將管理員密碼變碼。此安全措施的用途在於您未啟用 ec2setpassword 設定時，可防止在 Windows Sysprep 完成之後存取執行個體。

  C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
- 順序為 3 的同步命令：系統會執行下列命令：

C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd

此命令會新增下列登錄機碼，以重新啟用 RDP：

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f

OOBE 階段

1. 使用 EC2Config 服務回應檔案，系統可以指定下列組態：
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
   - <NetworkLocation>Other</NetworkLocation>
   - <ProtectYourPC>3</ProtectYourPC>
   - <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
   - <TimeZone>UTC</TimeZone>
   - <RegisteredOrganization>Amazon.com</RegisteredOrganization>
   - <RegisteredOwner>Amazon</RegisteredOwner>

   > **Note**
   >
   > 在一般化和特製化階段期間，EC2Config 服務會監控 OS 狀態。如果 EC2Config 偵測到 OS 處於 Sysprep 階段，即會將下列訊息發佈至系統日誌：
   > EC2ConfigMonitorState: 0 Windows is being configured.
   > SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. OOBE 階段完成後，系統會從下列位置執行 `SetupComplete.cmd`：`C:\Windows\Setup\Scripts\SetupComplete.cmd`。在 2015 年 4 月之前的 Amazon 公有 AMI 中，此檔案為空白，而且不會對映像執行任何操作。在 2015 年 4 月之後的公有 AMI 中，此檔案包含以下數值：call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd"。

3. 系統會執行 PostSysprep.cmd，執行下列操作：
   - 將本機管理員密碼設定為不過期。如果密碼過期，管理員可能會無法登入。
   - 設定 MSSQLServer 機器名稱 (如果已安裝)，讓名稱與 AMI 同步。

Sysprep 後續階段

Windows Sysprep 完成之後，EC2Config 服務會將下列訊息傳送至主控台輸出：

```
Windows sysprep configuration complete.
   Message: Sysprep Start
   Message: Sysprep End
```

EC2Config 接著會執行下列動作：

1. 讀取 config.xml 檔案的內容，並列出所有已啟用的外掛程式。
2. 同時執行所有「Windows 就緒之前」(“Before Windows is ready”) 外掛程式。
   - Ec2SetPassword
   - Ec2SetComputerName
   - Ec2InitializeDrives
   - Ec2EventLog
   - Ec2ConfigureRDP
   - Ec2OutputRDPCert
   - Ec2SetDriveLetter
   - Ec2WindowsActivate
   - Ec2DynamicBootVolumeSize
3. 完成之後，會將「Windows 就緒」(“Windows is ready”) 訊息傳送至執行個體系統日誌。
4. 同時執行所有「Windows 就緒之後」(“After Windows is ready”) 外掛程式。
   - Amazon CloudWatch Logs
   - UserData
   - AWS Systems Manager (Systems Manager)

如需 Windows 外掛程式的詳細資訊，請參閱在 EC2 舊版 Windows 作業系統執行個體啟動期間，使用 EC2Config 服務執行任務。

搭配 EC2Config 服務執行 Windows Sysprep

使用下列程序，使用 Windows Sysprep 和 EC2Config 服務來建立標準化 AMI。

1. 在 Amazon EC2 主控台中，找出或建立您要複製的 AMI。
2. 啟動和連線至 Windows 執行個體。
3. 自訂執行個體。
4. 在 EC2Config 服務回應檔案中指定組態設定：

   ```
   C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml
   ```

5. 從 Windows 開始選單中，選擇所有程式，然後選擇 EC2ConfigService Settings (EC2ConfigService 設定)。
6. 選擇 Ec2 Service Properties (Ec2 服務屬性) 對話方塊中的 Image (映像) 標籤。如需 Ec2 Service Properties (Ec2 服務屬性) 對話方塊中之選項和設定的詳細資訊，請參閱 Ec2 服務屬性。
7. 選取管理員密碼的選項，然後選取 Shutdown with Sysprep (使用 Sysprep 關機) 或 Shutdown without Sysprep (不使用 Sysprep 關機)。EC2Config 會根據您選取的密碼選項來編輯設定檔案。
   - Random (隨機)：EC2Config 會產生密碼，並以使用者金鑰加密，然後將已加密的密碼輸出到主控台。我們會在第一次啟動後停用此設定，以便執行個體重新開機或是停止後再啟動時，能持續使用這個密碼。
   - 指定：密碼會以未加密形式 (純文字) 存放於 Windows Sysprep 回應檔案。Windows Sysprep 接著執行時，會設定管理員密碼。如果您立即關機，即會立即設定密碼。在服務重新啟動時，會移除管理員密碼。請務必記住此密碼，之後即無法再擷取此密碼。
   - 保留現有項目：執行 Windows Sysprep 或重新啟動 EC2Config 時，管理員帳戶的現有密碼不會變更。請務必記住此密碼，之後即無法再擷取此密碼。
8. 選擇 OK (確定)。

當系統要求您確認是否要執行 Windows Sysprep 並關閉執行個體時，請選擇是。您會注意到 EC2Config 執行 Windows Sysprep。接著，系統會將您登出執行個體，然後執行個體會關機。如果您查看 Amazon EC2 主控台的 Instances (執行個體) 頁面，執行個體狀態會從 `Running` 變更為 `Stopping`，最後再變更為 `Stopped`。這時便能放心地從此執行個體建立 AMI。

您可以從命令列使用下列命令手動調用 Windows Sysprep 工具：

```
"%programfiles%\amazon\ec2configservice\"ec2config.exe -sysprep""
```

> **Note**
>
> 如果 CMD shell 已在 C:\Program Files\Amazon\EC2ConfigService\ 目錄中，則不需要命令中的雙引號標記。

不過，您必須審慎確認 `Ec2ConfigService\Settings` 資料夾中指定的 XML 檔案選項正確；否則，您可能無法連線至該執行個體。如需設定檔案的詳細資訊，請參閱 EC2Config 設定檔。如需從命令列設定 Windows Sysprep 並執行的範例，請參閱 `Ec2ConfigService\Scripts\InstallUpdates.ps1`。

複製 Amazon EC2 AMI

若需要跨多個區域採用一致的 Amazon EC2 執行個體組態，您可將單一 Amazon Machine Image (AMI) 用做範本來啟動所有執行個體。不過，AMIs是區域特定的資源，若要在特定 中啟動執行個體 AWS 區域，AMI 必須位於該區域。因此，如需在多個區域採用同一個 AMI，必須從來源區域複製該 AMI 到每個目標區域。

您用於複製 AMI 的方法，視乎您是要跨同一分區內的區域，還是跨不同分區進行複製而定：

- 跨區域複製 – 跨同一分區內的區域複製 AMI，舉例來說，跨商業分區內的區域。本主題介紹了此複製方法。
- 跨分割區複製 – 將 AMIs 從一個分割區複製到另一個分割區，例如從商業分割區複製到 AWS GovCloud (US) 分割區。若要了解此方法的相關詳細資訊，請參閱 存放和還原 AMI。
- 跨帳戶複製 – 建立另一個 AWS 帳戶 與您的 共用 AWS 帳戶的 AMI 複本。本主題介紹了此複製方法。

跨區域與跨帳戶 AMI 複製的複製操作完成所花時間為盡力而為。若需要控制完成時間，您可指定從 15 分鐘至 48 小時的完成時段，並確保在所需時間範圍內複製 AMI。基於時間的 AMI 複製操作適用額外費用。若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的基於時間的複製。

目錄

- 考量事項
- 成本
- 授予 Amazon EC2 AMI 複製許可
- 複製 AMI

- 停用待定的 AMI 複製作業
- Amazon EC2 AMI 複製的運作方式

考量事項

- 複製 AMI 的許可 - 您可以使用 IAM 政策來授予或拒絕使用者複製 AMI 的許可。自 2024 年 10 月 28 日起，您可在來源 AMI 指定 `CopyImage` 動作的資源層級許可。新 AMI 的資源層級許可與以前一樣可用。
- 啟動許可和 Amazon S3 儲存貯體許可 – AWS 不會將啟動許可或 Amazon S3 儲存貯體許可從來源 AMI 複製到新的 AMI。複製操作完成後，您可將啟動許可和 Amazon S3 儲存貯體許可套用至新的 AMI。
- 標籤 - 您只能複製連接至來源 AMI 的使用者定義之 AMI 標籤。將不會複製連接至其他 AWS 帳戶 的系統標籤 (字首為 `aws:`) 和使用者定義的標籤。當複製 AMI 時，您可以連接新標籤至新 AMI 及其備份快照。
- 基於時間的 AMI 複製配額 – 在達到累積快照複製輸送量配額之後，後續基於時間的 AMI 複製請求會失敗。若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的基於時間的複製配額。
- 受支援的來源目的地複製 – 來源 AMI 的位置會決定您是否可以複製，及新的 AMI 允許的目的地：
  - 若來源 AMI 位於一個區域，您可在該區域內對其進行複製、複製到另一個區域、複製到與該區域關聯的 Outpost，或者複製到該區域中的本機區域。
  - 如果來源 AMI 位於本機區域，您可以在該本機區域內進行複製、將其複製到該本機區域的父區域，或複製到具有相同父區域的某些其他本機區域。
  - 若來源 AMI 位於 Outpost，則您無法將其複製。
- 來源和目的地的 CLI 參數 – 若使用 CLI，會支援以下參數，以便指定待複製 AMI 的來源位置及新 AMI 的目的地。請注意，複製操作必須在目的地區域中啟動；如果您省略 `--region` 參數，則目的地會假設 AWS CLI 設定中設定的預設區域。

來源至目的地	來源參數	目的地參數
區域至區域	`--source-region`	`--region`
區域至 Outpost	`--source-region`	`--destination-outpost-arn` (Outpost 的 ARN)
區域至本機區域	`--source-region`	`--destination-availability-zone` (本機

來源至目的地	來源參數	目的地參數
	必須為本機區域的父區域。	區域名稱) 或 `--destination-availability-zone-id` (本機區域 ID)
本機區域至區域	`--source-region` 必須為本機區域的父區域。 來源本機區域係由指定之來源 AMI ID 的位置推定。	`--region` 必須為本機區域的父區域。
本機區域至本機區域	`--source-region` 必須為本機區域的父區域。 來源本機區域係由指定之來源 AMI ID 的位置推定。	`--destination-availability-zone` (本機區域名稱) 或 `--destination-availability-zone-id` (本機區域 ID)

成本

若未指定完成時間，複製 AMI 無須費用。然而，額外費用適用於基於時間的 AMI 複製操作。若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的基於時間的複製。

標準儲存與資料傳輸費用適用。如果您複製 EBS 後端 AMI，任何其他 EBS 快照的儲存則都會產生費用。

授予 Amazon EC2 AMI 複製許可

若要複製 EBS 支援的 AMI 或 Amazon S3 支援的 AMI，您需要下列 IAM 許可：

- `ec2:CopyImage` - 複製 AMI。對於 EBS 支援的 AMI，其還針對 AMI 備份快照授予複製許可。
- `ec2:CreateTags` - 標記目標 AMI。對於 EBS 支援的 AMI，其還針對目標 AMI 備份快照授予標記許可。

如果您要複製執行個體儲存體支援的 AMI，則需要下列額外 IAM 許可：

- `s3:CreateBucket` - 在新 AMI 的目標區域建立 S3 儲存貯體

- `s3:PutBucketOwnershipControls` – 為新建立的 S3 儲存貯體啟用 ACLs，以便使用`aws-exec-read`標準 ACL 寫入物件
- `s3:GetBucketAcl` – 讀取來源儲存貯體ACLs
- `s3:ListAllMyBuckets` - 為目標區域的 AMI 尋找現有 S3 儲存貯體
- `s3:GetObject` - 讀取來源儲存貯體物件
- `s3:PutObject` - 將物件寫入目標儲存貯體
- `s3:PutObjectAcl` - 將新物件的許可寫入目標儲存貯體

> **Note**
>
> 自 2024 年 10 月 28 日起，您可在來源 AMI 指定 `CopyImage` 動作的資源層級許可。目標 AMI 的資源層級許可與以前一樣可用。如需詳細資訊，請參閱「服務授權參考」中 Amazon EC2 定義的操作下表中的 CopyImage。

範例 IAM 政策，用於複製 EBS 支援的 AMI 並標記目標 AMI 和快照

以下範例政策授予您許可，以複製任何 EBS 支援的 AMI 並標記目標 AMI 及其備份快照。

> **Note**
>
> 自 2024 年 10 月 28 日起，您可在 `Resource` 元素中指定快照。如需詳細資訊，請參閱「服務授權參考」中 Amazon EC2 定義的操作下表中的 CopyImage。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "PermissionToCopyAllImages",
        "Effect": "Allow",
        "Action": [
            "ec2:CopyImage",
            "ec2:CreateTags"
        ],
        "Resource": [
```

```
            "arn:aws:ec2:*::image/*",
            "arn:aws:ec2:*::snapshot/*"
        ]
    }]
}
```

範例 IAM 政策，用於複製 EBS 支援的 AMI，但禁止標記新快照

當您獲得 ec2:CopySnapshot 許可時，系統會自動授予 ec2:CopyImage 許可。可明確拒絕授予標記新備份快照的許可，以覆蓋 ec2:CreateTags 動作的 Allow 效果。

以下範例政策授予您許可，以複製任何 EBS 支援的 AMI，但禁止您標記目標 AMI 的新備份快照。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:::snapshot/*"
        }
    ]
}
```

範例 IAM 政策，用於複製 Amazon S3 支援的 AMI 並標記目標 AMI

以下範例政策授予您許可，可將指定來源儲存貯體的任何 Amazon S3 支援的 AMI 複製到指定區域，並標記目標 AMI。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "PermissionToCopyAllImages",
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amis-for-111122223333-in-us-east-2-hash"
            ]
        }
    ]
}
```

若要尋找 AMI 來源儲存貯體的 Amazon Resource Name (ARN)，請透過 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台，在導覽窗格中選擇 AMI，然後在來源欄中找到該儲存貯體名稱。

> **Note**
>
> 只有在您第一次將 Amazon S3 支援的 AMI 複製到個別區域時，才需要 `s3:CreateBucket` 許可。之後，已在區域中建立的 Amazon S3 儲存貯體用來儲存您複製到該區域的所有未來 AMIs。

複製 AMI

您可複製您擁有的 AMI，或者由另一個帳戶共享給您的 AMI。若要了解受支援的來源與目的地組合，請參閱 考量事項。

Console

複製 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從主控台導覽列選取內含該 AMI 的區域。
3. 在導覽窗格中選擇 AMI，以顯示您區域可用的 AMI 清單。
4. 如果您未看到要複製的 AMI，請選擇其他篩選條件。您可按以下方式篩選 AMI：由我擁有、私有映像、公有映像與已停用映像。
5. 選取欲複製的 AMI，然後選擇動作、複製 AMI。
6. 在複製 Amazon Machine Image (AMI) 頁面，指定以下資訊：
   a. AMI copy name (AMI 複本名稱)：新 AMI 的名稱。名稱中可納入作業系統資訊，因為 Amazon EC2 在顯示有關 AMI 的詳細資訊時不提供此資訊。
   b. AMI copy description (AMI 複本描述)：根據預設，描述包括來源 AMI 的資訊，讓您能夠辨識原始和複本內容。您可視需要變更此描述。
   c. Destination region (目的地區域)：AMI 複製到的區域。如需詳細資訊，請參閱跨區域複製及跨帳戶複製。
   d. 複製標籤：選取此核取方塊可在複製 AMI 時包含使用者定義的 AMI 標籤。將不會複製連接至其他 AWS 帳戶 的系統標籤 (字首為 `aws:`) 和使用者定義的標籤。
   e. 基於時間的複製：您可如下所示指定複製操作是在特定時間範圍內完成，還是盡力完成：

- 如需在特定時間範圍內完成複製：
  - 選取啟用基於時間的複製。
  - 對於完成持續時間，請輸入複製操作允許的分鐘數 (增量為 15 分鐘)。完成持續時間適用於與 AMI 關聯的全部快照。

    若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的基於時間的複製。
- 盡力完成複製：
  - 將啟用基於時間的複製保持不選。

f. (EBS 支援的 AMI) 加密 AMI 複本的 EBS 快照：選取此核取方塊來加密目標快照或使用不同金鑰重新加密它們。如果您已啟用預設加密，加密 AMI 複本的 EBS 快照核取方塊已選取且無法清除。如需詳細資訊，請參閱加密和複製。

g. (EBS 支援的 AMI) KMS 金鑰：用於加密目標快照的 KMS 金鑰。

h. 標籤：您可以使用相同標籤來標記新 AMI 和新快照，或者可以使用不同標籤來進行標記。

- 若要使用相同標籤來標記新 AMI 和新快照，請選擇共同標記映像和快照。相同標籤即套用至新 AMI 和每個建立的快照。
- 若要使用不同標籤來標記新 AMI 和新快照，請選擇個別標記映像和快照。不同標籤即套用至新 AMI 和建立的快照。但是，請注意，建立的所有新快照都會取得相同標籤；您無法使用不同的標籤標記每個新快照。

若要新增標籤，請選擇 Add tag (新增標籤)，然後輸入標籤的鍵和值。針對每個標籤，重複執行上述動作。

i. 當您準備好建立 AMI 時，請選擇複製 AMI。

新 AMI 的初始狀態為 `Pending`。當狀態為 `Available` 時，表示 AMI 複製作業完成。

AWS CLI

如需將 AMI 從一個區域複製到另一個區域

使用 copy-image 指令。您必須同時指定來源和目的地區域。請使用 `--source-region` 參數指定來源區域。您可使用 `--region` 參數來指定目的地區域 (或省略此參數，假設為您的 AWS CLI 設定中設定的預設區域)。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
```

```
    --source-region us-west-2 \
    --name my-ami \
    --region us-east-1
```

在 AMI 複製期間加密目標快照時，務必指定其他這些參數：--encrypted 和 --kms-key-id。

如需從一個區域複製 AMI 到本機區域

使用 copy-image 指令。您必須同時指定來源與目的地。請使用 --source-region 參數指定來源區域。您可使用 --destination-availability-zone 參數來指定目的地本機區域 (您可改為使用 --destination-availability-zone-id)。請注意，您僅可從一個區域複製 AMI 到同一區域內的本機區域。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

如需從本機區域複製 AMI 到一個區域

使用 copy-image 指令。您必須同時指定來源與目的地。請使用 --source-region 參數指定來源區域。您可使用 --region 參數來指定目的地區域 (或者省略此參數，假設為您的 AWS CLI 設定中設定的預設區域)。來源本機區域係由指定之來源 AMI ID 的位置推定。請注意，您僅可從本機區域複製 AMI 到其父區域。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --name my-ami \
    --region cn-north-1
```

如需從一個本機區域複製 AMI 到另一個本機區域

使用 copy-image 指令。您必須同時指定來源與目的地。請使用 --source-region 參數來指定本機區域的來源區域。您可使用 --destination-availability-zone 參數來指定目的地本機區域 (您可改為使用 --destination-availability-zone-id)。來源本機區域係由指定之來源 AMI ID 的位置推定。您可以使用 --region 參數指定目的地 Local Zone 的父區域 （ 或省略此參數，以假設 AWS CLI 設定中設定的預設區域 ）。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

PowerShell

如需將 AMI 從一個區域複製到另一個區域

使用 Copy-EC2Image cmdlet。您必須同時指定來源和目的地區域。請使用 -SourceRegion 參數指定來源區域。您可使用 -Region 參數或者 Set-AWSDefaultRegion cmdlet，來指定目的地區域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion us-west-2 `
    -Name my-ami `
    -Region us-east-1
```

在 AMI 複製期間加密目標快照時，務必指定其他這些參數：-Encrypted 和 -KmsKeyId。

如需從一個區域複製 AMI 到本機區域

使用 Copy-EC2Image cmdlet。您必須同時指定來源與目的地。請使用 -SourceRegion 參數指定來源區域。您可使用 -DestinationAvailabilityZone 參數來指定目的地本機區域 (您可改為使用 -DestinationAvailabilityZoneId)。請注意，您僅可從一個區域複製 AMI 到同一區域內的本機區域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
    -Name my-ami `
    -Region cn-north-1
```

如需從本機區域複製 AMI 到一個區域

使用 Copy-EC2Image cmdlet。您必須同時指定來源與目的地。請使用 -SourceRegion 參數指定來源區域。您可使用 -Region 參數或者 Set-AWSDefaultRegion cmdlet，來指定目的地區域。

來源本機區域係由指定之來源 AMI ID 的位置推定。請注意，您僅可從本機區域複製 AMI 到其父區域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -Name my-ami `
    -Region cn-north-1
```

如需從一個本機區域複製 AMI 到另一個本機區域

使用 Copy-EC2Image cmdlet。您必須同時指定來源與目的地。請使用 -SourceRegion 參數來指定本機區域的來源區域。您可使用 -DestinationAvailabilityZone 參數來指定目的地本機區域 (您可改為使用 -DestinationAvailabilityZoneId)。來源本機區域係由指定之來源 AMI ID 的位置推定。您可使用 -Region 參數或者 Set-AWSDefaultRegion cmdlet，來指定目的地本機區域的父區域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
    -Name my-ami `
    -Region cn-north-1
```

停用待定的 AMI 複製作業

您可使用下列程序來停止待定的 AMI 複製。

Console

如需停止 AMI 複製操作

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽列中的區域選取器，選取目的地區域。
3. 在導覽窗格中，選擇 AMIs (AMI)。
4. 選取要停止複製的 AMI，然後選擇動作、取消註冊 AMI。
5. 要求確認時，請選擇 Deregister AMI (取消註冊 AMI)。

AWS CLI

如需停止 AMI 複製操作

請使用 deregister-image 命令。

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

PowerShell

如需停止 AMI 複製操作

使用 Unregister-EC2Image cmdlet。

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

Amazon EC2 AMI 複製的運作方式

複製來源 AMI 會產生相同但獨立的新 AMI，我們也稱之為目標 AMI。目標 AMI 有其唯一 AMI ID。您可變更來源 AMI 或取消註冊，不會影響目標 AMI。反之亦然。

藉由 EBS 支援的 AMI，每個備份快照會複製成相同但完全獨立的目標快照。如果您複製 AMI 到新區域，則快照為完整 (非增量) 複本。如果您加密未加密的備份快照，或將其加密為新的 KMS 金鑰，則快照為完整 (非增量) 複本。AMI 的後續複製作業會產生備份快照的增量複本。

目錄

- 跨區域複製
- 跨帳戶複製
- 基於時間的 AMI 複製操作
- 加密和複製

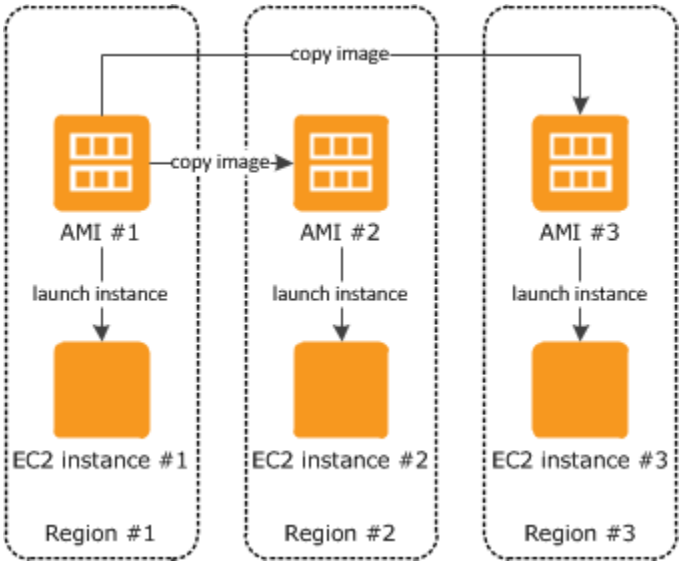
跨區域複製

跨越地理區域複製 AMI，具有下列優點：

- 一致的全球部署：將 AMI 從一個區域複製到另一個區域，可讓您以相同 AMI 在不同區域啟動一致的執行個體。

- 擴展能力：無論使用者的位置，設計並建置滿足全球使用者需求的應用程式將變得更輕鬆。
- 效能：您可分散應用程式，並將應用程式的重要元件放置於靠近使用者的位置，藉此增加效能。您亦可運用區域特定的功能，例如執行個體類型或其他 AWS 服務。
- 高可用性：您可跨 AWS 區域設計並部署應用程式，增加可用性。

下圖顯示來源 AMI 和兩個位於不同區域之複製 AMI 之間的關係，以及從每個 AMI 啟動的 EC2 執行個體。從 AMI 啟動執行個體時，其所處位置與該 AMI 的區域相同。若您變更來源 AMI 並希望在目標區域的 AMIs 反映這些變更，您必須將來源 AMI 重新複製到目標區域。



首次將 Amazon S3 支援的 AMI 複製到某個區域時，我們會為複製到該區域的 Amazon S3 建立 AMIs 儲存貯體。所有複製到該區域的 Amazon S3 支援的 AMI，都會存放於此儲存貯體。儲存貯體名稱為下列格式：amis-for-*account*-in-*region*-*hash*。例如：`amis-for-123456789012-in-us-east-2-yhjmxvp6`。

先決條件

複製 AMI 前，您必須確認來源 AMI 的內容已更新為可支援在不同區域運作。例如，您應更新資料庫連線字串或類似的應用程式組態資料，以指向適當的資源。否則，目的地區域內新 AMI 所啟動的執行個體，仍可能會使用來源區域的資源，影響效能和成本。

限制

- 目的地區域限制為 300 個並行 AMI 複製操作。這亦適用於基於時間的 AMI 複製操作。
- 您無法將全虛擬化 (PV) AMI 複製到不支援 PV AMI 的區域。如需詳細資訊，請參閱虛擬化類型。

跨帳戶複製

如果來自另一個 的 AMI AWS 帳戶 與您的 共用 AWS 帳戶，您可以複製共用的 AMI。這稱為跨帳戶複製。與您共用的 AMI 是來源 AMI。當您複製來源 AMI 時，您將建立新 AMI。新 AMI 通常稱為目標 AMI。

AMI 費用

- 對於共用 AMI，會向共用 AMI 的帳戶收取該區域的儲存費用。
- 若您複製與您的帳戶共享的 AMI，則您是您帳戶內目標 AMI 的擁有者。
  - 來源 AMI 的擁有者需支付標準 Amazon EBS 或 Amazon S3 傳輸費用。
  - 我們會向您收取目的地區域的目標 AMI 儲存費用。

資源許可

若要從另一個帳戶複製已與您共享的 AMI，來源 AMI 的擁有者必須授予您支援 AMI 的儲存體的讀取許可，而非 AMI 本身。該儲存體為相關聯的 EBS 快照 (Amazon EBS 支援的 AMI) 或相關聯的 S3 儲存貯體 (Amazon S3 支援的 AMI)。如果共用 AMI 具有加密的快照，則擁有者必須與您共用金鑰。如需有關授予資源許可的詳細資訊，對於 EBS 快照，請參閱 Amazon EBS 使用者指南中的與其他 AWS 帳戶共享 Amazon EBS 快照，對於 S3 儲存貯體，請參閱 Amazon S3 使用者指南中的 Amazon S3 的身分與存取管理。

> **Note**
>
> 連接到來源 AMI 的標籤不會跨帳戶複製到目標 AMI。

基於時間的 AMI 複製操作

若針對包含單一關聯快照的 EBS 支援的 AMI 來起始基於時間的 AMI 複製操作，則其行為方式與個別基於時間的快照複製操作相同，且相同的輸送量限制適用。

若針對包含多個關聯快照的 EBS 支援的 AMI 來起始基於時間的 AMI 複製操作，則其行為方式與並行的基於時間的快照複製操作相同，且相同的輸送量限制適用。每個關聯快照皆會產生單獨的快照複製請求，每個請求都會計入您的累積快照複製輸送量配額。您指定的完成持續時間會套用於每個關聯快照。

若要了解詳細資訊，請參閱 Amazon EBS 使用者指南中的基於時間的複製。

加密和複製

下表說明各種 AMI 複製案例支援的加密。雖然您可將未加密快照複製為加密快照，但是無法將加密快照複製為未加密快照。

案例	描述	支援
1	未加密到未加密	是
2	加密到加密	是
3	未加密到加密	是
4	加密到未加密	否

> **Note**
>
> 在 CopyImage 動作期間的加密僅適用於 Amazon EBS 後端 AMIs。因為 Amazon S3 支援的 AMI 不使用快照，所以您無法透過複製來變更其加密狀態。

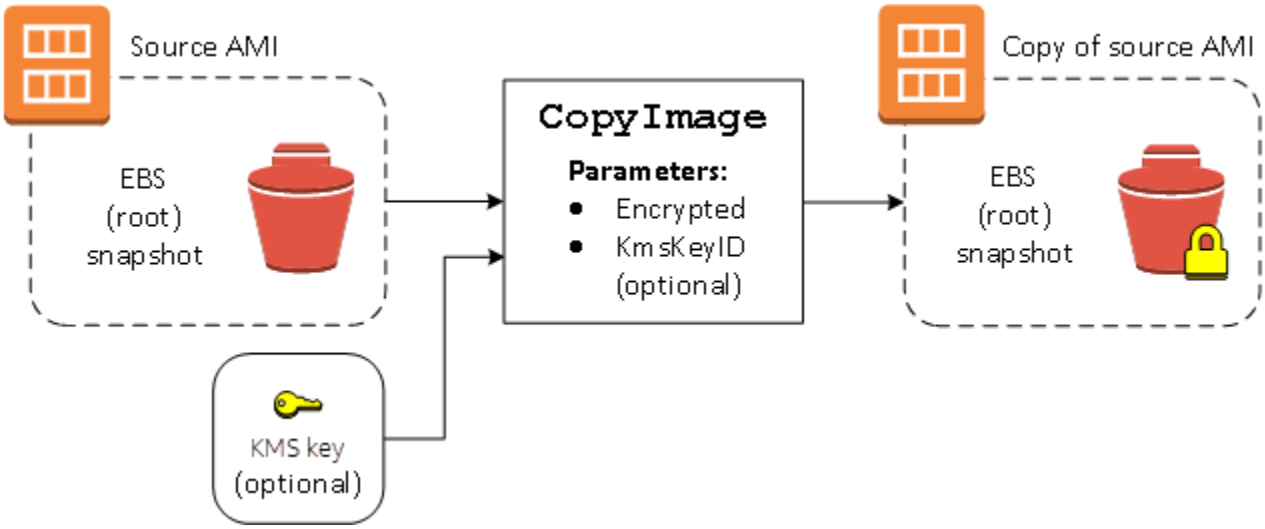
當您在不指定加密參數的情況下複製 AMI 時，預設會以其原始加密狀態複製後備快照。因此，如果來源 AMI 由未加密的快照支援，則產生的目標快照也將未加密。同樣地，如果來源 AMI 的快照已加密，則產生的目標快照也會由相同的 AWS KMS 金鑰加密。對於由多個快照支援的 AMI，每個目標快照都會保留其對應來源快照的加密狀態。

若要在 AMI 複製期間變更目標備份快照的加密狀態，您可指定加密參數。下列範例顯示非預設情況，其中指定了加密參數並執行 CopyImage 動作以變更目標 AMI 的加密狀態。

將未加密來源 AMI 複製到加密目標 AMI

在此案例中，未加密根快照所支援的 AMI 將複製到具有加密根快照的 AMI。透過兩個加密參數叫用 CopyImage 動作，包括客戶受管金鑰。因此，根快照的加密狀態會變更，目標 AMI 會受到包含與來源快照相同資料的根快照支援，但使用指定的金鑰進行加密。您需要為兩個 AMI 中的快照支付儲存成本，以及從任一 AMI 啟動的任何執行個體的費用。

> **Note**
>
> 啟用預設加密，這與在 AMI 中針對所有快照將 Encrypted 參數設定為 true 具有相同效果。



設定 `Encrypted` 參數會加密此執行個體的單一快照。若您未指定 `KmsKeyId` 參數，則會使用預設的客戶受管金鑰來加密快照複本。

如需複製具有加密快照的 AMIs 的詳細資訊，請參閱對 EBS 後端 AMI 使用加密。

使用 S3 存放和還原 AMI

您可以將 Amazon Machine Image (AMI) 存放在 Amazon S3 儲存貯體中，將 AMI 複製到另一個 S3 儲存貯體，然後從 S3 儲存貯體中還原。透過使用 S3 儲存貯體存放和還原 AMI，您可以將 AMIs 從一個 AWS 分割區複製到另一個分割區，例如從主要商業分割區複製到 AWS GovCloud (US) 分割區。您也可以透過將其存放在 S3 儲存貯體中，來建立 AMI 的封存複本。

支援使用 S3 存放和還原 AMI 的 API 包括 `CreateStoreImageTask`、`DescribeStoreImageTasks` 和 `CreateRestoreImageTask`。

`CopyImage` 是建議用於複製 AWS 分割區內 AMIs API。但是，`CopyImage` 無法將 AMI 複製到另一個分割區。

如需 AWS 分割區的相關資訊，請參閱《IAM 使用者指南》中的 Amazon Resource Names (ARNs) 頁面上的###。

> **Warning**
>
> 在 AWS 分割區或 AWS 區域之間移動資料時，請務必遵循所有適用的法律和業務需求，包括但不限於任何適用的政府法規和資料落地要求。

目錄

- 使用案例
- 限制
- 成本
- AMI 存放和還原如何運作
- 建立存放映像任務

使用案例

使用存放和還原 API 來執行下列動作：

- 在 AWS 分割區之間複製 AMI
- 製作 AMI 的封存複本

在 AWS 分割區之間複製 AMI

透過使用 S3 儲存貯體存放和還原 AMI，您可以將 AMI 從一個 AWS 分割區複製到另一個分割區，或從一個 AWS 區域複製到另一個區域。在下列範例中，您將 AMI 從主要商業分割區複製到 AWS GovCloud (US) 分割區，特別是從 `us-east-2`區域複製到 `us-gov-east-1`區域。

如果要將 AMI 從一個分割區複製到另一個分割區，請依照下列步驟執行：

- 透過使用 `CreateStoreImageTask` 將 AMI 存放在當前區域的 S3 儲存貯體中。在此範例中，S3 儲存貯體位於 `us-east-2`。
- 使用 `DescribeStoreImageTasks` 監控存放任務的進度。當任務完成時，該物件在 S3 儲存貯體中變得可見。
- 使用您選擇的程序，將存放的 AMI 物件複製到目標分割區中的 S3 儲存貯體。在此範例中，S3 儲存貯體位於 `us-gov-east-1`。

> **Note**
>
> 由於每個分割區都需要不同的 AWS 登入資料，因此您無法將 S3 物件直接從一個分割區複製到另一個分割區。跨分割區複製 S3 物件的程序超出本文件的範圍。我們提供下列複製程序作為範例，但您必須使用符合安全性需求的複製程序。
>
> - 若要跨分割區複製一個 AMI，複製程序非常簡單明瞭，具體如下：從來源儲存貯體下載物件至中繼主機 (例如 EC2 執行個體或筆記型電腦)，然後從中繼主機上傳物件至目標儲存貯體。對於程序的每個階段，請使用分割區的 AWS 登入資料。

- 為了更能夠持續使用，請考慮開發一個能管理複製的應用程式，最好可使用 S3 分段下載和上傳。

- 使用 `CreateRestoreImageTask` 從目標分割區中的 S3 儲存貯體還原 AMI。在此範例中，S3 儲存貯體位於 `us-gov-east-1`。
- 透過描述 AMI 來監控還原任務的進度，以查看其狀態何時變為可用。您也可以透過描述快照來監控組成還原 AMI 的快照進度百分比。

製作 AMI 的封存複本

您可以透過將其存放在 S3 儲存貯體中，來建立 AMI 的封存複本。AMI 被打包為 S3 中的單一物件，並且所有 AMI 中繼資料 (不包括共享資訊) 作為存放之 AMI 的一部分保留。AMI 資料被壓縮為儲存程序的一部分。包含可輕鬆壓縮之資料的 AMI 會在 S3 中產生較小的物件。為了降低成本，您可以使用較低成本的 S3 儲存層。如需詳細資訊，請參閱 Amazon S3 儲存類別和 Amazon S3 定價

限制

- 若要存放 AMI，您的 AWS 帳戶 必須擁有 AMI 及其快照，或者 AMI 及其快照必須直接與您的帳戶共用。如果 AMI 僅公開共用，則無法儲存 AMI。
- 使用這些 API 僅能存放 EBS 後端 AMI。
- 不支援半虛擬 (PV) AMI。
- 可存放的 AMI (壓縮之前) 大小限制為 5,000 GB。
- 儲存映像請求配額：1，200 GB 的儲存工作 （ 快照資料 ） 正在進行中。
- 還原映像請求的配額：進行中 600 GB 的還原工作 （ 快照資料 ）。
- 在存放任務期間，不得刪除快照，且執行存放的 IAM 主體必須具有快照的存取權，否則存放程序將會失敗。
- 您無法在同一個 S3 儲存貯體中建立 AMI 的多個複本。
- 存放在 S3 儲存貯體中的 AMI 無法使用其原始 AMI ID 還原。您可以透過使用 AMI 別名來減輕這種情況。
- 目前，只有使用 AWS 、 AWS Command Line Interface SDKs APIs 和 Amazon EC2 API 才能支援存放區和還原 API。您無法使用 Amazon EC2 主控台存放和還原 AMI。

成本

當您使用 S3 存放和還原 AMI 時，需支付存放和還原 API 所使用之服務的費用，以及資料傳輸的費用。這些 API 使用 S3 和 EBS 直接 API (由這些 API 內部使用以存取快照資料)。如需詳細資訊，請參閱 Amazon S3 定價和 Amazon EBS 定價。

AMI 存放和還原如何運作

要使用 S3 存放和還原 AMI，請使用以下 API：

- `CreateStoreImageTask` – 將 AMI 存放在 S3 儲存貯體中
- `DescribeStoreImageTasks` – 提供 AMI 存放任務的進度
- `CreateRestoreImageTask` – 從 S3 儲存貯體還原 AMI

API 的運作方式

- CreateStoreImageTask
- DescribeStoreImageTasks
- CreateRestoreImageTask
- 檔案路徑

CreateStoreImageTask

`CreateStoreImageTask` API 將 AMI 做為單一物件存放在 S3 儲存貯體。

API 建立一個從 AMI 及其快照中讀取所有資料的任務，然後使用 S3 分段上傳將資料存放在 S3 物件中。API 採用 AMI 的所有元件，包括大多數非特定區域的 AMI 中繼資料以及 AMI 中包含的所有 EBS 快照，並將其打包為 S3 中的單一物件。資料會在上傳過程中進行壓縮，以減少 S3 中使用的空間量，因此 S3 中的物件可能會小於 AMI 中快照大小的總和。

如果呼叫此 API 的帳戶可見 AMI 和快照標籤，則會將其保留。

S3 中的物件與 AMI 具有相同的 ID，但有 `.bin` 副檔名。下列資料也會作為 S3 中繼資料標籤存放於 S3 物件：AMI 名稱、AMI 描述、AMI 註冊日期、AMI 擁有者帳戶，以及存放操作的時間戳記。

完成任務所需的時間取決於 AMI 的大小。這也取決於有多少其他任務正在進行中，因為任務已排入佇列。您可以呼叫 `DescribeStoreImageTasks` API 追蹤任務進度。

所有進行中 AMIs 的大小總和限制為每個帳戶 1，200 GB 的 EBS 快照資料。系統會拒絕更多的任務建立，除非進行中的任務數量小於限制數量。例如，如果目前存放了一個具有 200 GB 快照資料的 AMI，

以及另一個具有 400 GB 快照資料的 AMI，則會接受另一個請求，因為進行中的總量為小於限制的 600 GB。但是，如果目前正在存放具有 1，200 GB 快照資料的單一 AMI，則進一步的任務會被拒絕，直到任務完成為止。

DescribeStoreImageTasks

`DescribeStoreImageTasks` API 描述了 AMI 存放任務的進度。您可以描述指定 AMI 的任務。如果未指定 AMI，您會取得過去 31 天內處理的所有存放映像任務的分頁清單。

對於每個 AMI 任務，回應會指示任務為 `InProgress`、`Completed` 或 `Failed`。對於任務 `InProgress`，回應會以百分比顯示預估進度。

任務會以相反的時間順序列出。

目前，只能檢視上個月的任務。

CreateRestoreImageTask

`CreateRestoreImageTask` API 啟動一個任務，從先前使用 `CreateStoreImageTask` 請求建立的 S3 物件還原為 AMI。

還原任務可以在執行存放任務的相同或不同區域中執行。

將從中還原 AMI 物件的 S3 儲存貯體必須位於請求還原任務的相同區域。AMI 將在此區域還原。

AMI 會還原其中繼資料，例如名稱、描述和對應於存放 AMI 之值的區塊型儲存設備映射。此帳戶的區域中，AMI 的名稱必須是唯一的。如果您沒有提供名稱，新的 AMI 將取得與原始 AMI 相同的名稱。AMI 取得在還原過程中產生的新 AMI ID。

完成 AMI 還原任務所需的時間取決於 AMI 的大小。這也取決於有多少其他任務正在進行中，因為任務已排入佇列。您可以透過描述 AMI (describe-images) 或其 EBS 快照 (describe-snapshots) 來查看任務的進度。如果任務失敗，AMI 和快照集會轉為失敗狀態。

所有進行中 AMIs 的大小總和限制為每個帳戶 EBS 快照資料的 600 GB （根據還原後的大小）。系統會拒絕更多的任務建立，除非進行中的任務數量小於限制數量。

檔案路徑

您可以透過下列方式在存放和還原 AMI 時使用檔案路徑：

- 在 S3 中存放 AMI 時，可以將檔案路徑新增到儲存貯體名稱。在內部，系統會將路徑與儲存貯體名稱分隔開來，然後將路徑新增至產生用來存放 AMI 的物件金鑰。完整的物件路徑會顯示在 API 呼叫的回應中。

- 還原 AMI 時，由於物件索引鍵參數可用，因此可以將路徑新增至物件索引鍵值的開頭。

範例：使用附加檔案路徑的儲存貯體名稱

儲存 AMI 時，請在儲存貯體名稱後指定檔案路徑。

```
amzn-s3-demo-bucket/path1/path2
```

下列為產生的物件金鑰。

```
path1/path2/ami-0abcdef1234567890.bin
```

若要還原 AMI，您可指定儲存貯體名稱和物件金鑰。如需範例，請參閱 the section called “建立存放映像任務”。

建立存放映像任務

當您將 AMI 存放在 S3 儲存貯體時，會建立存放映像任務。您可以使用存放映像任務來監控流程的進度與結果。

目錄

- 保護您的 AMI
- 使用 S3 存放和還原 AMI 的許可
- 建立存放映像任務
- 建立還原映像任務

保護您的 AMI

請務必確保 S3 儲存貯體具有足夠的安全性來保護 AMI 的內容，而且只要 AMI 物件保留在儲存貯體中，安全性就必須要維護。如果無法完成，則不建議使用這些 API。確保不允許公開存取 S3 儲存貯體。我們建議針對存放 AMI 的 S3 儲存貯體啟用伺服器端加密，儘管這不是必要的。

如需如何為 S3 儲存貯體設置適當安全性設定的相關資訊，請檢閱下列安全性主題：

- 封鎖對 Amazon S3 儲存體的公開存取權
- 對 Amazon S3 儲存貯體設定預設伺服器端加密行為
- 我可以使用哪些 S3 儲存貯體政策來符合 AWS Config 規則 s3-bucket-ssl-requests-only？
- 啟用 Amazon S3 伺服器存取記錄日誌

將 AMI 快照複製到 S3 物件時，資料會透過 TLS 連線複製。您可以將 AMI 與加密的快照一起存放，但快照會在存放程序中解密。

使用 S3 存放和還原 AMI 的許可

若您的 IAM 主體將使用 Amazon S3 存放或還原 AMI，則您需要授予其所需的許可。

下列範例政策包含允許 IAM 主體執行存放和還原任務所需的所有動作。

您還可建立僅授予委託人存取特定資源的 IAM 政策。如需更多範例政策，請參閱《IAM 使用者指南》中的存取 AWS 資源的管理。

> **Note**
>
> 若組成 AMI 的快照已進行加密，或者您的帳戶依預設啟用了加密功能，則您的 IAM 主體必須具有使用 KMS 金鑰的許可權限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectTagging",
                "s3:AbortMultipartUpload",
                "ebs:CompleteSnapshot",
                "ebs:GetSnapshotBlock",
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
                "ebs:PutSnapshotBlock",
                "ebs:StartSnapshot",
                "ec2:CreateStoreImageTask",
                "ec2:DescribeStoreImageTasks",
                "ec2:CreateRestoreImageTask",
                "ec2:GetEbsEncryptionByDefault",
```

```
                "ec2:DescribeTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

建立存放映像任務

若要在 S3 儲存貯體存放 AMI，請先建立存放映像任務。完成任務所需的時間取決於 AMI 的大小。您可追蹤任務進度，直到任務成功或失敗為止。

AWS CLI

若要建立存放映像任務

使用 create-store-image-task 命令。

```
aws ec2 create-store-image-task \
    --image-id ami-0abcdef1234567890 \
    --bucket amzn-s3-demo-bucket
```

以下為範例輸出。

```
{
  "ObjectKey": "ami-0abcdef1234567890.bin"
}
```

若要描述存放任務的進度

使用 describe-store-image-tasks 命令。

```
aws ec2 describe-store-image-tasks \
    --image-ids ami-0abcdef1234567890 \
    --query StoreImageTaskResults[].StoreTaskState \
    --output text
```

以下為範例輸出。

```
InProgress
```

PowerShell

若要建立存放映像任務

使用 New-EC2StoreImageTask cmdlet。

```
New-EC2StoreImageTask `
    -ImageId ami-0abcdef1234567890 `
    -Bucket amzn-s3-demo-bucket
```

以下為範例輸出。

```
ObjectKey         : ami-0abcdef1234567890.bin
```

若要描述存放任務的進度

使用 Get-EC2StoreImageTask cmdlet。

```
(Get-EC2StoreImageTask -ImageId ami-0abcdef1234567890).StoreTaskState
```

以下為範例輸出。

```
InProgress
```

建立還原映像任務

您必須指定已還原 AMI 的名稱。此帳戶的區域中，AMI 的名稱必須是唯一的。還原的 AMI 會取得一個新的 AMI ID。

AWS CLI

若要建立還原映像任務

使用 create-restore-image-task 命令。

```
aws ec2 create-restore-image-task \
    --object-key ami-0abcdef1234567890.bin \
    --bucket amzn-s3-demo-bucket \
    --name "my-restored-ami"
```

以下為範例輸出。

```
{
   "ImageId": "ami-1234567890abcdef0"
}
```

PowerShell

若要建立還原映像任務

使用 New-EC2RestoreImageTask cmdlet。

```
New-EC2RestoreImageTask `
    -ObjectKey ami-0abcdef1234567890.bin `
    -Bucket amzn-s3-demo-bucket `
    -Name "my-restored-ami"
```

以下為範例輸出。

```
ImageId         : ami-1234567890abcdef0
```

使用 AMI 原始伺服器追蹤 AMI 的原始伺服器

AMI 祖系透過傳回其所有祖系 AMI IDs 和區域，協助您追蹤 AMIs 的原始伺服器。當您建立或複製 AMI 時，新的 AMI 會保留其來源（父系）AMI 的 ID 和區域。這可讓您將 AMIs 鏈追蹤回根 AMI。

主要優點

使用 AMI 原始伺服器可協助您：

- 追蹤 AMI 衍生產品以確保符合內部政策。
- 在舊版 AMIs 中發現安全問題時，識別潛在易受攻擊的 AMI。
- 維持跨多個區域的 AMI 原始伺服器可見性。

主題

- AMI 原始伺服器的運作方式
- 考量事項
- 檢視 AMI 原始伺服器

- 識別來源 AMI

AMI 原始伺服器的運作方式

AMI 原始伺服器會識別用來建立指定 AMI 的父系 AMI、父系的父系，以此類推，直到根 AMI。以下是其運作方式：

- 每個 AMI 會顯示其來源（父系）AMI 的 ID 和區域。
- 從您選取的 AMI 開始，原始項目清單會依序顯示每個父系 AMI。
- 原始項目清單會追蹤回 ，直到到達根 AMI。根 AMI 是下列其中一項：
  - 來自已驗證供應商的公有 AMI（由其擁有者別名識別，即 `amazon`或 `aws-marketplace`)。
  - 沒有記錄祖先的 AMI。例如，使用 RegisterImage 直接從一組快照建立 AMI 時，沒有要追蹤的來源 AMI，與從執行個體建立 AMI 時不同。
  - 來源 AMI 來自不同分割區的 AMI。
  - 清單中的第 50 個 AMI。原始清單中AMIs 數目上限為 50。

考量事項

- 來源 AMI 的 ID 和區域僅適用於使用 CreateImage、CopyImage 或 CreateRestoreImageTask 建立 AMIs。
- 對於使用 CreateImage 建立AMIs（從執行個體建立 AMI)，來源 AMI ID 是用來啟動執行個體的 AMI ID。
- 來源 AMI 資訊不適用於：
  - 使用 RegisterImage 建立AMIs，因為它們是從快照建立的。
  - 對於某些較舊AMIs。
- 來源 AMI 資訊會在下列情況下保留：
  - AMIs會跨區域複製。
  - 已取消註冊（刪除）來源 AMIs。
  - 您無法存取來源 AMIs。
- 每個舊版清單限制為 50 AMIs。

檢視 AMI 原始伺服器

您可以使用下列方法檢視 AMI 的祖系。

Console

檢視 AMI 的祖先

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI，然後選擇 AMI 原始標籤。
4. AMI 祖系項目表列出祖系清單中的所有 AMIs。

   - AMI ID – 原始清單中每個 AMI 的識別符。資料表中的第一個項目是選取的 AMI，後面接著其祖先。
   - 來源 AMI ID – 建立 AMI ID 欄中 AMI 的 AMI ID。破折號 (-) 表示 AMI 原始清單的結尾。
   - 來源 AMI 區域 – AWS 區域 來源 AMI 所在的 。
   - 上階層級 – 上階清單中的位置，其中：
     - 0 （輸入 AMI) 表示您想知道其前代的所選 AMI。
     - 增加數字會顯示較舊的祖先。
     - ***n*** （原始 AMI) 表示根 AMI，數字表示原始清單往前多遠。
   - 建立日期 – 建立 AMI 時，採用 UTC 格式。
   - 擁有者別名 – AMI 擁有者的別名 （例如 amazon)。破折號 (-) 表示 AMI 沒有擁有者別名。

AWS CLI

檢視 AMI 的祖先

使用 get-image-ancestry 命令並指定 AMI ID。

```
aws ec2 get-image-ancestry \
    --image-id ami-1111111111EXAMPLE \
    --region us-east-1
```

以下為範例輸出。輸出會依祖先順序列出 AMIs：第一個項目是指定的 （輸入） AMI，後面接著其父系、父系的父系等，並以根 AMI 結尾。

```
{
    "ImageAncestryEntries": [
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
```

```
            "ImageId": "ami-1111111111EXAMPLE", // Input AMI
            "SourceImageId": "ami-2222222222EXAMPLE",
            "SourceImageRegion": "us-east-1"

        },
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
            "ImageId": "ami-2222222222EXAMPLE", // Parent AMI
            "SourceImageId": "ami-3333333333EXAMPLE",
            "SourceImageRegion": "us-east-1"
        },
        ...
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
            "ImageId": "ami-8888888888EXAMPLE", // Root AMI
            "ImageOwnerAlias": "aws-marketplace",
            "SourceImageId": "ami-9999999999EXAMPLE",
            "SourceImageRegion": "us-east-2"
        }
    ]
}
```

PowerShell

檢視 AMI 的祖先

使用 Get-EC2ImageAncestry cmdlet。

```
Get-EC2ImageAncestry -ImageId ami-1111111111EXAMPLE
```

以下為範例輸出。輸出會依祖先順序列出 AMIs：第一個項目是指定的（輸入）AMI，後面接著其父系、父系的父系等，並以根 AMI 結尾。

```
ImageAncestryEntries : {
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
        ImageId = "ami-1111111111EXAMPLE"    # Input AMI
        SourceImageId = "ami-2222222222EXAMPLE"
        SourceImageRegion = "us-east-1"
    },
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
```

```
        ImageId = "ami-2222222222EXAMPLE"    # Parent AMI
        SourceImageId = "ami-3333333333EXAMPLE"
        SourceImageRegion = "us-east-1"
    },
    ...
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
        ImageId = "ami-8888888888EXAMPLE"    # Root AMI
        ImageOwnerAlias = "aws-marketplace"
        SourceImageId = "ami-9999999999EXAMPLE"
        SourceImageRegion = "us-east-2"
    }
}
```

識別來源 AMI

如果您只需要識別用來建立 AMI 的直接父系（來源）AMI，您可以使用下列方法。

Console

識別用來建立所選 AMI 的來源 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI 以檢視詳細資訊。

   來源 AMI 資訊於下列欄位顯示：來源 AMI ID 和來源 AMI 區域

AWS CLI

識別用於建立指定 AMI 的來源 AMI

使用 describe-images 命令。

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].{ID:SourceImageId,Region:SourceImageRegion}"
```

以下為範例輸出。

```
[
    {
        "ID": "ami-0abcdef1234567890",
        "Region": "us-west-2"
    }
}
```

PowerShell

識別用於建立指定 AMI 的來源 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image -ImageId ami-0abcdef1234567890 | Select SourceImageId,
 SourceImageRegion
```

以下為範例輸出。

```
SourceImageId           SourceImageRegion
-------------           -----------------
ami-0abcdef1234567890 us-west-2
```

使用 AMI 浮水印來追蹤和識別 AMIs

AMI 浮水印是您連接到私有 AMIs識別符，用於追蹤來源並強制執行控管政策。浮水印會在整個 AMI 生命週期中持續存在：

- 如果您從浮水印 AMI 啟動的執行中執行個體建立新的 AMI，則新的 AMI 會繼承浮水印。
- 如果您複製浮水印 AMI，則複製會攜帶浮水印。
- 如果您與其他帳戶共用浮水印 AMI，收件人仍然可以看到浮水印。

主要優點

- 追蹤帳戶和區域的來源 - 識別哪些 AMIs 衍生自您核准的基礎映像。
- 篩選和尋找您帳戶之間的相關 AMIs。
- 協助 AMI 消費者探索和識別與專案或組織相關聯的信任 AMIs。
- 結合浮水印與允許的 AMIs 來強制執行控管，將執行個體啟動限制為僅攜帶已核准浮水印AMIs。

主題

- AMI 浮水印的運作方式
- 所需的許可
- 將浮水印連接至 AMI
- 從 AMI 分離浮水印
- 檢視 AMI 浮水印
- 依浮水印篩選 AMIs

AMI 浮水印的運作方式

AMI 浮水印是您連接到 AMIs結構化識別符。以下說明浮水印的主要特性：

- 持久性 – 當您將浮水印附加至 AMI 時，它會轉送至所有衍生 AMIs。
- 僅限擁有者 – 只有 AMI 擁有者可以將浮水印連接至 AMI。
- 所有人都可看見 – 任何有權存取 AMI 的人都可以檢視其浮水印。
- 限制為 5 – AMI 最多可有 5 個浮水印。
- 不適用於公有 AMIs – 您無法將浮水印連接至公有 AMIs，或如果 AMIs具有浮水印，則無法將其設為公有。
- 可篩選 – 使用 時，您可以依浮水印篩選 AMIs`describe-images`。

Watermark 格式

浮水印是具有下列欄位的結構化物件：

- `WatermarkKey` – 浮水印的唯一識別符，由 組成*`account-id`*:*`watermark-name`*。帳戶 ID 部分是 AMI 擁有者的 12 位數 AWS 帳戶 ID。浮水印名稱部分是客戶指定的名稱。
- `SourceImageRegion` – 您最初連接浮水印的 AMI 區域。
- `SourceImageId` – 您最初連接浮水印的 AMI。
- `SourceImageCreationDate` – 您最初連接浮水印的 AMI 建立日期。
- `WatermarkCreationTime` – 您套用浮水印時的時間戳記。

浮水印名稱必須為 3-128 個字元，可包含英數字元、括號 (())、方括號 (【】)、空格、句點 (.)、斜線 (/)、破折號 (-)、單引號 (')、符號 (@) 或底線 (_)。

所需的許可

若要使用 AMI 浮水印，您需要下列 IAM 許可：

- `ec2:AttachImageWatermark` – 將浮水印連接至 AMI。
- `ec2:DetachImageWatermark` – 從 AMI 分離浮水印。
- `ec2:DescribeImages` – 檢視 AMIs 上的浮水印。

將浮水印連接至 AMI

您可以使用 主控台 AWS CLI、 或 PowerShell 將浮水印連接至 AMI。

Console

將浮水印連接至 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤的浮水印區段中，選擇管理浮水印。
5. 輸入浮水印名稱，然後選擇連接。

AWS CLI

將浮水印連接至 AMI

使用 attach-image-watermark 命令。

```
aws ec2 attach-image-watermark \
    --image-id ami-1111111111EXAMPLE \
    --image-watermark-name "prod-baseline"
```

以下為範例輸出。

```
{
    "WatermarkKey": "123456789012:prod-baseline"
}
```

PowerShell

將浮水印連接至 AMI

使用 Add-EC2ImageWatermark cmdlet。

```
Add-EC2ImageWatermark `
    -ImageId ami-1111111111EXAMPLE `
    -ImageWatermarkName "prod-baseline"
```

您最多可以將 5 個浮水印連接至單一 AMI。

從 AMI 分離浮水印

您可以使用 主控台 AWS CLI、 或 PowerShell，從 AMI 分離浮水印。

Console

從 AMI 分離浮水印

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤的浮水印區段中，選擇管理浮水印。
5. 選取要移除的浮水印，然後選擇移除。

AWS CLI

從 AMI 分離浮水印

使用 detach-image-watermark 命令。

```
aws ec2 detach-image-watermark \
    --image-id ami-1111111111EXAMPLE \
    --image-watermark-key "111122223333:prod-baseline"
```

PowerShell

從 AMI 分離浮水印

使用 Remove-EC2ImageWatermark cmdlet。

```
Remove-EC2ImageWatermark `
    -ImageId ami-1111111111EXAMPLE `
    -ImageWatermarkKey "111122223333:prod-baseline"
```

> **Note**
>
> 從 AMI 分離浮水印並不會將其從已攜帶浮水印的衍生 AMIs中移除。為了確保浮水印保持持久性，請僅將 ec2:DetachImageWatermark 許可授予需要管理浮水印的信任管理員。

檢視 AMI 浮水印

您可以使用 主控台 AWS CLI、 或 PowerShell 來檢視 AMI 的浮水印。

Console

檢視 AMI 的浮水印

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤的浮水印區段中檢視浮水印。

AWS CLI

檢視 AMI 的浮水印

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-046863d776a820ccd \
    --region us-east-1
```

回應包含每個 AMI 的ImageWatermarks陣列。

```
{
    "Images": [
```

```
        {
            "ImageId": "ami-046863d776a820ccd",
            "Public": false,
            "OwnerId": "123456789012",
            ...
            "ImageWatermarks": [
                {
                    "WatermarkKey": "111122223333:prod-baseline",
                    "Region": "us-east-1",
                    "SourceImageId": "ami-0b752bf1df193a6c4",
                    "SourceImageCreationDate": "2024-07-10T08:15:00",
                    "CreationDate": "2024-07-12T14:30:00"
                },
                {
                    "WatermarkKey": "222222222222:security-approved",
                    "Region": "eu-north-1",
                    "SourceImageId": "ami-12345678",
                    "SourceImageCreationDate": "2024-06-01T10:00:00",
                    "CreationDate": "2024-06-05T09:45:00"
                }
            ]
        }
    ]
}
```

PowerShell

檢視 AMI 的浮水印

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-046863d776a820ccd).ImageWatermarks
```

依浮水印篩選 AMIs

您可以使用 主控台 AWS CLI、 或 PowerShell，依浮水印篩選 AMIs。

Console

依浮水印篩選 AMIs

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 AMI。
3. 在搜尋列中，選擇浮水印索引鍵篩選條件，然後輸入浮水印索引鍵值。

AWS CLI

依浮水印篩選 AMIs

使用具有 `image-watermark-key` 篩選條件的 describe-images 命令。

```
aws ec2 describe-images \
    --filters "Name=image-watermark-key,Values=111122223333:prod-baseline"
```

這會傳回您有權存取且帶有指定浮水印的所有 AMIs，包括透過複製操作繼承該浮水印的衍生 AMIs。

PowerShell

依浮水印篩選 AMIs

使用 Get-EC2Image cmdlet 搭配 `-Filter` 參數。

```
Get-EC2Image `
    -Filter @{Name="image-watermark-key"; Values="111122223333:prod-baseline"}
```

這會傳回您有權存取且帶有指定浮水印的所有 AMIs，包括透過複製操作繼承該浮水印的衍生 AMIs。

AMI 允許的執行個體類型

身為 AMI 擁有者，您可以指定 AMI 支援或不支援的執行個體類型。這可防止在不相容的執行個體類型上啟動。Amazon EC2 會在啟動時強制執行允許的執行個體類型。

例如，如果您的 AMI 需要 GPU 硬體，您只能將其限制為 GPU 執行個體類型。如果您的 AMI 與特定執行個體系列不相容，您可以從允許的執行個體類型中排除該系列。

目錄

- 允許執行個體類型的運作方式
- 設定 AMI 的允許執行個體類型

- 檢視 AMI 允許的執行個體類型
- 使用允許的執行個體類型啟動行為
- 考量事項

允許執行個體類型的運作方式

您可以透過 AMI 上的 InstanceTypeSpecification 屬性控制允許的執行個體類型。此屬性包含兩個清單：

- SupportedInstanceTypes – AMI 支援的執行個體類型。只有這些執行個體類型可以使用 AMI 啟動。
- UnsupportedInstanceTypes – AMI 不支援的執行個體類型。所有其他執行個體類型都可以使用 AMI 啟動。

Amazon EC2 使用以下邏輯評估規格：

- 如果InstanceTypeSpecification未設定 ，Amazon EC2 會允許所有執行個體類型。這是預設行為。
- 如果只設定 SupportedInstanceTypes ，則只允許指定的執行個體類型。Amazon EC2 會封鎖所有其他執行個體類型。
- 如果只設定 UnsupportedInstanceTypes ，Amazon EC2 會允許指定執行個體類型以外的所有執行個體類型。
- 如果已設定兩個清單，則執行個體類型必須位於 中SupportedInstanceTypes，且不得位於 中UnsupportedInstanceTypes。

萬用字元支援

這兩個清單都支援使用 * 字元的萬用字元模式。您可以使用萬用字元來比對多個執行個體類型，而無需個別列出每個執行個體類型。

下表顯示萬用字元模式的範例。

模式	相符
t3.*	t3 執行個體系列中的所有大小（例如，t3.micro、t3.small和 t3.large)

模式	相符
p4d.*	p4d 執行個體系列中的所有大小
g5.*	g5 執行個體系列中的所有大小
*xlarge	所有執行個體系列中 xlarge 或更大的任何執行個體類型
*.12xlarge	所有執行個體系列的任何 12xlarge 執行個體類型

設定 AMI 的允許執行個體類型

您可以使用 Amazon EC2 主控台或 AWS Command Line Interface () 來設定或變更 AMI 的允許執行個體類型AWS CLI。您必須是 AMI 擁有者才能執行此程序。

Console

設定 AMI 的允許執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI，然後選擇動作、管理執行個體類型規格。
4. 在支援的執行個體類型或不支援的執行個體類型下，選擇新增。
5. 選取 執行個體類型。
6. 選擇儲存。

AWS CLI

設定 AMI 支援的執行個體類型

使用 replace-image-instance-type-specification 命令。下列範例允許 以外的所有 t3和 a2執行個體類型t3.micro。

```
aws ec2 replace-image-instance-type-specification \
    --image-id ami-1234567890abcdef0 \
    --instance-type-specification '{"SupportedInstanceTypes": ["t3.*", "a2.*"],
 "UnsupportedInstanceTypes": ["t3.micro"]}'
```

僅設定 AMI 不支援的執行個體類型

使用以下命令來封鎖特定執行個體類型，同時允許所有其他執行個體類型。

```
aws ec2 replace-image-instance-type-specification \
    --image-id ami-1234567890abcdef0 \
    --instance-type-specification '{"UnsupportedInstanceTypes": ["t3.micro",
 "t3.nano"]}'
```

從 AMI 移除執行個體類型規格

使用以下命令而不指定 `--instance-type-specification`來移除限制，並允許所有執行個體類型。

```
aws ec2 replace-image-instance-type-specification \
    --image-id ami-1234567890abcdef0
```

檢視 AMI 允許的執行個體類型

您可以使用 Amazon EC2 主控台或 檢視 AMI 的執行個體類型規格 AWS CLI。

Console

檢視 AMI 允許的執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤上，檢視支援和不支援的執行個體類型。

AWS CLI

檢視 AMI 的執行個體類型規格

使用 describe-images 命令。如果 AMI 具有執行個體類型規格，回應會包含 `InstanceTypeSpecification` 欄位。

```
aws ec2 describe-images \
```

```
    --image-ids ami-1234567890abcdef0
```

以下是具有執行個體類型規格之 AMI 的範例輸出。

```
{
    "Images": [
        {
            "ImageId": "ami-1234567890abcdef0",
            ...
            "InstanceTypeSpecification": {
                "SupportedInstanceTypes": [
                    {
                        "InstanceType": "t3.*"
                    },
                    {
                        "InstanceType": "a2.*"
                    }
                ],
                "UnsupportedInstanceTypes": [
                    {
                        "InstanceType": "t3.micro"
                    }
                ]
            }
        }
    ]
}
```

使用允許的執行個體類型啟動行為

當您啟動執行個體時，Amazon EC2 會檢查指定的執行個體類型是否與 AMI 的執行個體類型規格相容。如果不允許執行個體類型，Amazon EC2 會封鎖啟動並傳回錯誤。

範例：封鎖啟動

下列命令會嘗試使用不支援 的 AMI 啟動`t3.micro`執行個體`t3.micro`。

```
aws ec2 run-instances \
    --image-id ami-1234567890abcdef0 \
    --instance-type t3.micro
```

Amazon EC2 傳回下列錯誤：

```
An error occurred (InvalidParameterCombination) when calling the RunInstances
 operation: This AMI does not support the specified instance type. Check DescribeImages
 for InstanceTypeSpecification, and try again.
```

若要解決此錯誤，請選擇 AMI 支援的執行個體類型。使用 `describe-images`命令來檢視 AMI 的執行個體類型規格。

考量事項

當您使用允許的執行個體類型時，請記住下列資訊。

- 根據預設，AMI 沒有執行個體類型規格。Amazon EC2 允許所有執行個體類型，直到您明確設定規格為止。
- 只有 AMI 擁有者可以設定或變更執行個體類型規格。
- 當您使用 複製 AMI 時`CopyImage`，Amazon EC2 會在新的 AMI 中保留執行個體類型規格。
- Amazon EC2 會以硬式區塊的形式強制執行執行個體類型規格。如果不允許執行個體類型，則啟動會失敗並顯示`InvalidParameterCombination`錯誤。
- 規格不會影響現有的執行個體。它僅適用於新的啟動。
- 如果不允許設定執行個體類型，則使用執行個體類型規格參考 AMI 的啟動範本和 Auto Scaling 群組可能會失敗。我們建議在共用 AMIs 上設定規格之前驗證相容性。
- `ReplaceImageInstanceTypeSpecification` 動作會取代整個規格。若要新增或移除個別執行個體類型，您必須在請求中包含完整的更新規格。

管理及監控 AMI 用量

AWS 提供多種功能，可協助您有效管理和監控 AMI 用量。您可以追蹤哪些帳戶正在使用您的共用 AMIs、識別您的 AMIs 上次使用的時間，以及探索您 中哪些資源 AWS 帳戶 參考特定 AMIs。

下表提供了管理及監控 AMI 用量的功能概觀：

功能	使用案例	主要優點
AMI 用量報告	了解哪些 AWS 帳戶 正在使用您的 AMIs，以及每個 AMI 的使用量。	• 識別參考 AMIs AWS 帳戶 和資源類型，以便您可以安全地取消註冊或停用 AMIs。

功能	使用案例	主要優點
		• 確定未使用的 AMI 以取消註冊，進而降低儲存費用。 • 確定您最常用的 AMI。
上次使用追蹤	檢查上次使用您的 AMI 的時間	• 確定未使用的 AMI，以便您安全地取消註冊 AMI。 • 確定未使用的 AMI 以取消註冊，進而降低儲存費用。
AMI 參考檢查	確保您的 AWS 資源使用最新的合規 AMIs。	• 稽核帳戶中 AMI 的使用。 • 檢查正在參考的特定 AMI 的位置。 • 藉由更新資源來參考最新 AMI，以確保合規。

主題

- 檢視您的 AMI 用量
- 檢查上次使用 Amazon EC2 AMI 的時間
- 確定參考指定 AMI 的資源

檢視您的 AMI 用量

如果您與其他 共用 Amazon Machine Image AMIs) AWS 帳戶，無論是與特定 AWS 帳戶、組織、組織單位 (OUs) 或公開 ，您都可以透過建立 AMIs用量報告來查看這些 AMI 的使用方式。報告提供對以下項目的深入解析：

- 在 EC2 執行個體或啟動範本中 AWS 帳戶 使用您的 AMIs

- 參考每個 AMI 的 EC2 執行個體或啟動範本的數目

AMI 用量報告藉由提供以下協助可助您更高效地管理 AMI：

- 識別參考 AMIs AWS 帳戶 和資源類型，以便您可以安全地取消註冊或停用 AMIs。
- 確定未使用的 AMI 以取消註冊，進而降低儲存費用。
- 確定您最常用的 AMI。

目錄

- AMI 用量報告運作方式
- 建立 AMI 用量報告
- 檢視 AMI 用量報告
- 刪除 AMI 用量報告
- 匯入配額

AMI 用量報告運作方式

若建立 AMI 用量報告，您可指定：

- 待報告的 AMI。
- AWS 帳戶 要檢查的 （特定帳戶或所有帳戶）。
- 待檢查的資源類型 (EC2 執行個體、啟動範本，或者兩者）。
- 若是啟動範本，則為待檢查的版本數目 (預設為 20 個最新版本）。

Amazon EC2 可針對每個 AMI 建立一個單獨的請求。每份報告皆提供：

- AWS 帳戶 使用 AMI 的 清單。
- 根據每個帳戶的資源類型參考 AMI 的資源計數。請注意，針對啟動範本，若在啟動範本的多個版本中參考 AMI，則計數僅為 1。

> **Important**
>
> 若產生 AMI 用量報告，可能不會包含最新活動。過去 24 小時小時的執行個體活動，以及過去幾天的啟動範本活動，可能皆不會在報告中顯示。

Amazon EC2 在建立報告後 30 天會自動刪除報告。您可透過 EC2 主控台下載報告，以便在本機保留。

建立 AMI 用量報告

如需檢視如何使用 AMI，必須先建立 AMI 用量報告，指定要報告的帳戶與資源類型。建立報告之後，您可以檢視報告的內容。您可透過 EC2 主控台下載報告。

Console

如需建立 AMI 用量報告

1. 請前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI，然後選擇動作、AMI 用量、檢視我的 AMI 用量。
4. 在建立我的 AMI 用量報告頁面，執行以下操作：
   a. 針對資源類型，選取要報告的一個或多個資源類型。
   b. 針對帳戶，執行下列其中一項操作：
      - 選擇指定帳戶 ID，然後針對要報告的每個帳戶，選擇新增帳戶 ID。
      - 選擇包括所有帳戶，以報告全部帳戶。
   c. 選擇建立我的 AMI 用量報告。
5. 在 AMI 頁面，選擇我的 AMI 用量索引標籤。
6. 選擇報告 ID 來檢視其詳細資訊。

AWS CLI

如需針對帳戶清單建立 AMI 用量報告

請使用以下必要參數並搭配 create-image-usage-report 命令：

- `--image-id` – 待報告 AMI 的 ID。
- `--resource-types` – 待檢查資源的類型。在以下範例中，待檢查資源類型為 EC2 執行個體及啟動範本。另外還會指定待檢查啟動範本版本的數目 (`version-depth=100`)。

如需報告特定帳戶，使用 `--account-ids` 參數來指定待報告的每個帳戶 ID。

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --account-ids 111122223333 444455556666 123456789012 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

如需建立所有帳戶的 AMI 用量報告

如需使用指定 AMI 來報告所有帳戶，可使用同一命令但省略 `--account-ids` 參數。

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

以下為範例輸出。

```
{
    "ReportId": "amiur-00b877d192f6b02d0"
}
```

如需監控報告建立狀態

請使用 describe-image-usage-reports 命令，然後指定報告 ID。

```
aws ec2 describe-image-usage-reports --report-ids amiur-00b877d192f6b02d0
```

以下為範例輸出。`State` 欄位的初始值為 `pending`。為了能夠檢視報告項目，狀態必須為 `available`。

```
{
```

```
    "ImageUsageReports": [
        {
            "ImageId": "ami-0e9ae3dc21c2b3a64",
            "ReportId": "amiur-abcae3dc21c2b3999",
            "ResourceTypes": [
                {"ResourceType": "ec2:Instance"}
            ],
            "State": "pending",
            "CreationTime": "2025-09-29T13:27:12.322000+00:00",
            "ExpirationTime": "2025-10-28T13:27:12.322000+00:00"
        }
    ]
}
```

PowerShell

如需針對帳戶清單建立 AMI 用量報告

請使用以下必要參數並搭配 New-EC2ImageUsageReport cmdlet：

- `-ImageId` – 待報告 AMI 的 ID。
- `-ResourceType` – 待檢查資源的類型。在以下範例中，待檢查資源類型為 EC2 執行個體及啟動範本。另外還會指定待檢查啟動範本版本的數目 (`'version-depth' = 100`)。

如需報告特定帳戶，使用 `-AccountId` 參數來指定待報告的每個帳戶 ID。

```
New-EC2ImageUsageReport `
    -ImageId ami-0abcdef1234567890 `
    -AccountId 111122223333 444455556666 123456789012 `
    -ResourceType @(
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

如需建立所有帳戶的 AMI 用量報告

如需使用指定 AMI 來報告所有帳戶，可使用同一命令但省略 `-AccountId` 參數。

```
New-EC2ImageUsageReport `
    -ImageId ami-0abcdef1234567890 `
    -ResourceType @(
```

```
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

以下為範例輸出。

```
ReportId
--------
amiur-00b877d192f6b02d0
```

如需監控報告建立狀態

請使用 Get-EC2ImageUsageReport 命令，然後指定報告 ID。

```
Get-EC2ImageUsageReport -ReportId amiur-00b877d192f6b02d0
```

以下為範例輸出。State 欄位的初始值為 pending。為了能夠檢視報告項目，狀態必須為 available。

```
ImageUsageReports
-----------------
{@{ImageId=ami-0e9ae3dc21c2b3a64; ReportId=amiur-abcae3dc21c2b3999;
 ResourceTypes=System.Object[]; State=pending; CreationTime=2025-09-29;
 ExpirationTime=2025-10-28}}
```

檢視 AMI 用量報告

您可檢視在過去 30 天針對 AMI 建立的全部用量報告。Amazon EC2 在建立報告後 30 天會自動刪除報告。

針對每個報告，您可以看到使用 AMI AWS 帳戶 的 ，以及針對每個帳戶，依資源類型參考 AMI 的資源計數。您還可查看報告起始建立時間。只有在報告處於完成 （ 主控台 ） 或 available(AWS CLI) 狀態時，才能使用此資訊。

> **Important**
>
> 若產生 AMI 用量報告，可能不會包含最新活動。過去 24 小時小時的執行個體活動，以及過去幾天的啟動範本活動，可能皆不會在報告中顯示。

Console

檢視 AMI 用量報告

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI。
4. 選擇我的用量報告索引標籤。

   報告清單會顯示：

   - 過去 30 天內為所選 AMI 產生的所有報告。
   - 對於每個報告，報告起始時間欄會顯示報告建立的日期。
5. 選擇報告的 ID，以便檢視其內容。
6. 如需返回 AMI 詳細資訊頁面的我的用量報告索引標籤，選擇檢視此 AMI 的所有報告。

AWS CLI

如需列出指定 AMI 的全部 AMI 用量報告

請使用 describe-image-usage-reports 命令，然後指定 AMI 的 ID，以便獲取其報告清單。

```
aws ec2 describe-image-usage-reports --image-ids ami-0abcdef1234567890
```

以下為範例輸出。每個報告 ID 會與掃描的資源類型，以及報告建立日期與到期日期一同列示。您可透過此資訊來確定您想要檢視其項目的報告。

```
{
  "ImageUsageReports": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-1111111111111111",
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        }
      ],
      "State": "available",
      "CreationTime": "2025-09-29T13:27:12.322000+00:00",
```

```
      "ExpirationTime": "2025-10-28T13:27:12.322000+00:00",
      "Tags": []
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-2222222222222222",
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        },
        {
          "ResourceType": "ec2:LaunchTemplate"
        }
      ],
      "State": "available",
      "CreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ExpirationTime": "2025-10-30T13:27:12.322000+00:00",
      "Tags": []
    }
  ],
  "NextToken": "opaque"
}
```

如需檢視指定 AMI 的 AMI 用量報告內容

請使用 describe-image-usage-report-entries 命令，然後指定 AMI 的 ID。回應會傳回指定 AMI 的所有報告，並顯示已使用該 AMI 的帳戶及其資源計數。

```
aws ec2 describe-image-usage-report-entries --image-ids ami-0abcdef1234567890
```

以下為範例輸出。

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-1111111111111111"
    },
```

```
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 2,
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "001100110011",
      "UsageCount": 39,
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    }
  ],
  "NextToken": "opaque"
}
```

如需檢視指定報告的 AMI 用量報告內容

請使用 describe-image-usage-report-entries 命令，然後指定報告的 ID。針對指定報告，回應可傳回全部項目，並顯示已使用 AMI 的帳戶以及資源計數。

```
aws ec2 describe-image-usage-report-entries --report-ids amiur-11111111111111111
```

以下為範例輸出。

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
```

```
      "AccountId": "123412341234",
      "UsageCount": 4,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-1111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
      "AccountId": "001100110011",
      "UsageCount": 2,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-1111111111111111"
    }
  ],
  "NextToken": "opaque"
}
```

PowerShell

如需列出指定 AMI 的全部 AMI 用量報告

請使用 Get-EC2ImageUsageReport cmdlet，然後指定 AMI 的 ID，以便獲取其報告清單。

```
Get-EC2ImageUsageReport -ImageId ami-0abcdef1234567890
```

以下為範例輸出。每個報告 ID 會與掃描的資源類型，以及報告建立日期與到期日期一同列示。您可透過此資訊來確定您想要檢視其項目的報告。

```
@{
    ImageUsageReports = @(
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-1111111111111111"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-09-29T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-28T13:27:12.322000+00:00"
        },
        @{
```

```
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-2222222222222222"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-09-30T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-29T13:27:12.322000+00:00"
        },
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-33333333333333333"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-10-01T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-30T13:27:12.322000+00:00"
        }
    )
    NextToken = "opaque"
}
```

如需檢視指定 AMI 的 AMI 用量報告內容

請使用 Get-EC2ImageUsageReportEntry cmdlet，然後指定 AMI 的 ID。回應會傳回指定 AMI 的所有報告，並顯示已使用該 AMI 的帳戶及其資源計數。

```
Get-EC2ImageUsageReportEntry -ImageId ami-0abcdef1234567890
```

以下為範例輸出。

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
```

```
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 7
    ReportCreationTime = "2025-09-30T13:27:12.322000+00:00"
    ReportId = "amiur-2222222222222222"
    }...}
NextToken : opaque
```

如需檢視指定報告的 AMI 用量報告內容

請使用 Get-EC2ImageUsageReportEntry cmdlet，然後指定報告的 ID。針對指定報告，回應可傳回全部項目，並顯示已使用 AMI 的帳戶以及資源計數。

```
Get-EC2ImageUsageReportEntry -ReportId amiur-1111111111111111
```

以下為範例輸出。

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "123412341234"
    UsageCount = 4
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "************"
    UsageCount = 2
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }}
NextToken : opaque
```

刪除 AMI 用量報告

Amazon EC2 會在報告建立之後 30 天內自動將其刪除。您可在該時間之前手動將其刪除。

Console

如需刪除 AMI 用量報告

1. 請前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取 AMI。
4. 選擇我的 AMI 用量索引標籤。
5. 選擇待刪除報告旁邊的選項按鈕，然後選擇刪除。

AWS CLI

如需刪除 AMI 用量報告

請使用 delete-image-usage-report 命令，然後指定報告的 ID。

```
aws ec2 delete-image-usage-report --report-id amiur-0123456789abcdefg
```

PowerShell

如需刪除 AMI 用量報告

請使用 Remove-EC2ImageUsageReport cmdlet，並指定報告的 ID。

```
Remove-EC2ImageUsageReport -ReportId amiur-0123456789abcdefg
```

匯入配額

在建立 AMI 用量報告時，以下配額適用。這些配額適用於每個 AWS 區域。

說明	配額
每個 的進行中 (pending) AMI 用量報告 AWS 帳戶	2,000

說明	配額
每個 AMI 的進行中 (pending) AMI 用量報告	1

檢查上次使用 Amazon EC2 AMI 的時間

Amazon EC2 會自動追蹤上次使用 AMI 啟動執行個體的日期和時間。如果您有長時間未用於啟動執行個體的 AMI，請考慮該 AMI 是否適合取消註冊或棄用。

考量事項

- 使用 AMI 啟動執行個體時，在報告該使用情況之前會有 24 小時延遲。
- 您必須是 AMI 的擁有者才能取得上次啟動時間。
- AMI 用量資料自 2017 年 4 月開始提供。

Console

檢視 AMI 的上次啟動時間

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 AMI。
3. 從篩選條件列中，選擇 Owned by me (由我擁有)。
4. 選取 AMI 的核取方塊。
5. 在詳細資訊索引標籤上，找到上次啟動的時間。

AWS CLI

如需藉由描述 AMI 來檢視上次啟動的時間

使用 describe-images 命令。若 `LastLaunchedTime` 在輸出中不存在，請確認您擁有該 AMI。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].LastLaunchedTime \
    --output text
```

以下為範例輸出。

```
2025-02-17T20:22:19Z
```

如需檢視 AMI 的上次啟動的時間屬性

請使用 describe-image-attribute 命令。您必須為指定 AMI 的擁有者。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute lastLaunchedTime \
    --query LastLaunchedTime.Value \
    --output text
```

以下為範例輸出。

```
2025-02-17T20:22:19Z
```

PowerShell

如需藉由描述 AMI 來檢視上次啟動的時間

使用 Get-EC2Image cmdlet。若 `LastLaunchedTime` 在輸出中不存在，請確認您擁有該 AMI。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).LastLaunchedTime
```

以下為範例輸出。

```
2025-02-17T20:22:19Z
```

如需檢視 AMI 的上次啟動的時間屬性

使用 Get-EC2ImageAttribute cmdlet。您必須為指定 AMI 的擁有者。

```
(Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LastLaunchedTime).LastLaunchedTime
```

以下為範例輸出。

```
2025-02-17T20:22:19Z
```

確定參考指定 AMI 的資源

您可以識別參考指定 Amazon Machine Image (AMIs) AWS 的資源，無論 AMIs 是公有還是私有，還是擁有者。此可見度有助您確認資源使用最新的合規 AMI。

主要優點

檢查 AMI 參考有助您：

- 稽核帳戶中 AMI 的使用。
- 檢查正在參考的特定 AMI 的位置。
- 藉由更新資源來參考最新 AMI，以確保合規。

主題

- 支援的資源
- AMI 參考檢查運作方式
- 所需的 IAM 許可
- 檢查 AMI 參考的步驟

支援的資源

可簽入的 AMI 參考：

- EC2 執行個體
- 啟動範本
- SSM 參數
- Image Builder 映像配方
- Image Builder 容器配方

AMI 參考檢查運作方式

基本操作

您在執行 AMI 參考檢查時可：

- 指定待檢查的 AMI。
- 選擇掃描哪些資源類型。
- 接收可參考指定 AMI 的資源清單。

資源類型選取

在主控台中，您可選取待掃描的資源類型。

在 CLI 中，您可使用以下一個或兩個 CLI 參數，來指定待掃描的資源類型：

- `IncludeAllResourceTypes`：掃描全部受支援的資源類型。
- `ResourceTypes`：掃描您指定的資源類型。

回應範圍

您可使用 `ResourceTypes` 參數來自訂 `ResourceTypeOptions` 值，藉此限制 EC2 執行個體與啟動範本的回應範圍。主控台與 `IncludeAllResourceTypes` 參數皆使用預設選項值。若 `ResourceTypes` 與 `IncludeAllResourceTypes` 一起使用，`ResourceTypes` 選項值會優先於預設值。

下面為預設值：

Resource Type (資源類型)	範圍選項 (**`OptionName`**)	用途	**`OptionValue`** 與主控台的預設值

EC2 執行個體	`state-name`	依據執行個體狀態篩選	`pending`、`running`、`shutti`
			`down` 、`terminate`
			`d` 、`stopping`、`stopped`
			(所有狀態)

啟動範本	`version-depth`	指定待檢查的啟動範本版本數目 (自最新版本開始)	`10` (最新版本)

所需的 IAM 許可

如需使用 DescribeImageReferences API 來確定參考指定 AMI 的資源，需要下面的 IAM 許可來描述資源：

- ec2:DescribeInstances
- ec2:DescribeLaunchTemplates
- ec2:DescribeLaunchTemplateVersions
- ssm:DescribeParameters
- ssm:GetParameters
- imagebuilder:ListImageRecipes
- imagebuilder:ListContainerRecipes
- imagebuilder:GetContainerRecipe

使用 DescribeImageReferences API 的 IAM 政策範例

下面的範例政策授予您使用 DescribeImageReferences API 的許可，其中包括描述 EC2 執行個體、啟動範本、Systems Manager 參數、Image Builder 映像配方，以及 Image Builder 容器配方的許可。

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ec2:DescribeImageReferences",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions",
    "ssm:DescribeParameters",
    "ssm:GetParameters",
    "imagebuilder:ListImageRecipes",
    "imagebuilder:ListContainerRecipes",
    "imagebuilder:GetContainerRecipe"
   ],
   "Resource": "*",
   "Condition": {
    "ForAnyValue:StringEquals": {
```

```
      "aws:CalledVia": [
       "ec2-images.amazonaws.com"
      ]
     }
    }
   }
  ]
}
```

Important

我們強烈建議使用 AWS 受管政策，AmazonEC2ImageReferencesAccessPolicy而不是自行建立政策。建立僅提供必要許可的自訂 IAM 政策，需要時間與專業知識，並且在新的資源類型可用時，還需要更新。
AmazonEC2ImageReferencesAccessPolicy 受管政策：

- 授予使用 DescribeImageReferences API 所需的全部許可 (這些許可包括描述 EC2 執行個體、啟動範本、Systems Manager 參數，以及 Image Builder 容器配方與映像配方)。
- 在新資源類型變得可用時自動提供支援 (使用 IncludeAllResourceTypes 參數時尤其重要)。

您可將 AmazonEC2ImageReferencesAccessPolicy 政策連接至您的 IAM 身分 (使用者、群組或角色)。
若要檢視此政策中包括的許可，請參閱 AWS 受管政策參考中的 AmazonEC2ImageReferencesAccessPolicy。

檢查 AMI 參考的步驟

使用下列程序來識別哪些 AWS 資源正在參考指定的 AMIs。

Console

確定參考指定 AMI 的資源

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 選取一個或多個要檢查參考的 AMI。

4. 請選擇動作、AMI 用量、檢視參考的資源。
5. 在檢視參考所選 AMI 的資源頁面：
   a. 對於資源類型，請選取一個或多個資源類型。
   b. 請選擇檢視資源。
6. 參考所選 AMI 的資源區段即會顯示。清單顯示參考指定 AMI 的資源。每列可提供以下資訊：
   - AMI ID – 參考 AMI 的 ID。
   - 資源類型 – 參考 AMI 的資源的資源類型。
   - 資源 ID – 參考 AMI 的資源的 ID。

AWS CLI

如需檢查特定資源類型的 AMI 參考

請使用 `--resource-types` 參數並搭配 describe-image-references 命令。下面的範例將檢查 EC2 執行個體 (依執行個體狀態確定範圍)、啟動範本 (範圍確定為 20 個最近的啟動範本版本)，以及其他特定資源類型。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --resource-types \
        'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]' \
        'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=[20]}]' \
        'ResourceType=ssm:Parameter' \
        'ResourceType=imagebuilder:ImageRecipe' \
        'ResourceType=imagebuilder:ContainerRecipe'
```

以下為範例輸出。

```
{
    "ImageReferences": [
        {
            "ImageId": "ami-0abcdef1234567890",
            "ResourceType": "ec2:Instance",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
        },
        {
```

```
            "ImageId": "ami-1234567890abcdef0",
            "ResourceType": "ec2:LaunchTemplate",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:launch-template/
lt-1234567890abcdef0"
        }
    ]
}
```

如需檢查所有受支援資源類型的 AMI 參考

請使用 `--include-all-resource-types` 參數並搭配 describe-image-references 命令。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types
```

檢查所有受支援資源類型與特定選項的 AMI 參考

請使用 `--include-all-resource-types` 和 `--resource-types` 參數並搭配 describe-image-references 命令。此範例將檢查所有資源類型，同時將 EC2 執行個體的回應範圍限定為執行中或擱置中執行個體。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types \
    --resource-types
 'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]'
```

PowerShell

如需檢查特定資源類型的 AMI 參考

請使用 `-ResourceType` 參數並搭配 New-EC2Instance cmdlet。下面的範例將檢查 EC2 執行個體 (依執行個體狀態確定範圍)、啟動範本 (範圍確定為 20 個最近的啟動範本版本)，以及其他特定資源類型。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -ResourceType @(
        @{
            ResourceType = 'ec2:Instance'
```

```
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
                }
            )
        },
        @{
            ResourceType = 'ec2:LaunchTemplate'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'version-depth'
                    OptionValues = @('20')
                }
            )
        },
        @{
            ResourceType = 'ssm:Parameter'
        },
        @{
            ResourceType = 'imagebuilder:ImageRecipe'
        },
        @{
            ResourceType = 'imagebuilder:ContainerRecipe'
        }
    )
```

如需檢查所有受支援資源類型的 AMI 參考

請使用 `-IncludeAllResourceTypes` 參數並搭配 New-EC2Instance cmdlet。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -IncludeAllResourceTypes
```

檢查所有受支援資源類型與特定選項的 AMI 參考

請使用 `-IncludeAllResourceTypes` 與 `-ResourceType` 參數並搭配 Get-EC2ImageReference cmdlet。此範例將檢查所有資源類型，同時將 EC2 執行個體的回應範圍限定為執行中或擱置中執行個體。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
```

```
    -IncludeAllResourceTypes `
    -ResourceType @(
        @{
            ResourceType = 'ec2:Instance'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
                }
            )
        }
    )
```

棄用 Amazon EC2 AMI

您可以取代 AMI 來指出它已過期且不應該使用。您也可以為 AMI 指定未來的取代日期，指出 AMI 何時過期。例如，您可能會取代不再主動維護的 AMI，或者您可能會取代已被較新版本替代的 AMI。依預設，已取代的 AMI 不會出現在 AMI 清單中，以防止新使用者使用過期的 AMI。但是，現有使用者和啟動服務 (例如啟動範本和 Auto Scaling 群組) 可以透過指定其 ID 來繼續使用已取代的 AMI。若要刪除 AMI，以便使用者和服務無法使用它，您必須取消註冊它。

AMI 被取代後：

- 對於 AMI 使用者，已取代的 AMI 不會出現在 DescribeImages API 呼叫中，除非您指定其 ID 或指定已取代的 AMI 必須出現。AMI 擁有者繼續在 DescribeImages API 呼叫中查看已取代的 AMI。
- 對於 AMI 使用者，已取代的 AMI 無法在 EC2 主控台中選取。例如，已取代的 AMI 不會出現在啟動執行個體精靈的 AMI 目錄中。AMI 擁有者繼續在 EC2 主控台中查看已取代的 AMI。
- 對於 AMI 使用者，如果您知道已取代的 AMI 的 ID，則可以使用 API、CLI 或 SDK 透過已取代的 AMI 來啟動執行個體。
- 啟動服務 (例如啟動範本和 Auto Scaling 群組) 可以繼續參考已取代的 AMI。
- 使用隨後被取代的 AMI 來啟動的 EC2 執行個體不會受到影響，而且可以停止、啟動和重新啟動。

您可以同時取代私有和公有 AMI。

目錄

- 成本
- 考量事項

- 棄用 AMI
- 描述已取代的 AMI
- 取消 AMI 棄用

成本

當您取代 AMI 時，不會刪除 AMI。AMI 擁有者繼續為 AMI 的快照付費。要停止支付快照的費用，AMI 擁有者必須透過取消註冊 AMI 來刪除它。

考量事項

- 若要取代 AMI，您必須是 AMI 的擁有者。
- 最近未用於啟動執行個體的 AMI 可能就是待移除或取消註冊的項目。如需詳細資訊，請參閱the section called “檢查上次使用 AMI 的時間”。
- 您可建立 Amazon Data Lifecycle Manager EBS 支援的 AMI 政策，以自動化 EBS 後端 AMI 的取代。若要了解詳細資訊，請參閱建立 AMI 生命週期政策。
- 在預設情況下，所有公用 AMI 的棄用日期都設定為自 AMI 建立日期起兩年。您可將棄用日期設定為早於兩年。若要取消棄用日期或將棄用日期延後，您必須將與指定的 AWS 帳戶共享 AMI，使該 AMI 成為私有。

棄用 AMI

您可以在特定的日期和時間取代 AMI。您必須為 AMI 的擁有者。

棄用日期的上限是從現在起 10 年，但公用 AMI 除外，其上限是自建立日期起 2 年。您無法指定過去的日期。

Console

在特定日期取代 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇 AMI。
3. 從篩選條件列中，選擇 Owned by me (由我擁有)。
4. 選取 AMI，然後依序選取 Actions (動作)、Manage AMI Deprecation (管理 AMI 取代)。您可以選取多個 AMI，一次設定多個 AMI 的相同取代日期。

5. 選取啟用核取方塊，然後輸入取代日期和時間。
6. 選擇儲存。

AWS CLI

在特定日期取代 AMI

使用 enable-image-deprecation 命令。如果您指定秒數值，Amazon EC2 會將秒數四捨五入到最接近的分鐘。

```
aws ec2 enable-image-deprecation \
    --image-id ami-0abcdef1234567890 \
    --deprecate-at "2025-04-15T13:17:12.000Z"
```

PowerShell

在特定日期取代 AMI

使用 Enable-EC2ImageDeprecation cmdlet。如果您指定秒數值，Amazon EC2 會將秒數四捨五入到最接近的分鐘。

```
Enable-EC2ImageDeprecation `
    -ImageId ami-0abcdef1234567890 `
    -DeprecateAt 2025-04-15T13:17:12.000Z
```

描述已取代的 AMI

您可以檢視 AMI 的取代日期和時間，並依取代日期篩選 AMI。

Console

檢視 AMI 的取代日期

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽器中，選取 AMI，然後選取該 AMI。
3. 檢查取代時間欄位 (若您選取 AMI 旁的核取方塊，其位於詳細資料索引標籤)。該欄位顯示 AMI 的取代日期和時間。若欄位為空的，則 AMI 不會遭到取代。

依取代日期篩選 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇 AMI。
3. 從篩選條件列中，選擇 Owned by me (由我擁有) 或 Private images (私有映像) (私有映像包括與您共享以及您擁有的 AMI)。
4. 在搜尋列中，輸入 **`Deprecation time`** (當您輸入字母時，Deprecation time (取代時間) 篩選條件隨即顯示)，隨後選擇一個運算子及一個日期和時間。

AWS CLI

若描述全部 AMI，結果取決於您是 AMI 使用者還是 AMI 擁有者。

- AMI 使用者 – 依預設，若描述全部 AMI，將會排除與您共享但並非您擁有的棄用 AMI。如需要在結果中包括已棄用的 AMI，您必須指定 `--include-deprecated` 參數。
- AMI 擁有者 – 若描述全部 AMI，您擁有的全部 AMI 將會包括在內，包括棄用的 AMI。使用 `--no-include-deprecated` 選項不能排除您擁有的棄用 AMI。

如需在描述所有 AMI 時包括所有已棄用的 AMI

使用下面的 describe-images 命令。

```
aws ec2 describe-images
    --owners 123456789012 \
    --include-deprecated
```

描述帳戶中已棄用的 AMI

使用下面的 describe-images 命令。

```
aws ec2 describe-images \
    --owners self \
    --query "Images[?DeprecationTime!=null].ImageId" \
    --output text
```

以下為範例輸出。

```
ami-0abcdef1234567890
```

描述 AMI 的取代日期

使用下面的 describe-images 命令。若輸出中不存在 DeprecationTime，不會棄用 AMI，或者將其設定為在未來日期棄用。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].DeprecationTime \
    --output text
```

以下為範例輸出。

```
2025-05-01T00:00:00.000Z
```

PowerShell

如需列出帳戶的棄用 AMI

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -Owner self | Where-Object {$_.DeprecationTime -ne $null}).ImageId
```

以下為範例輸出。

```
ami-0abcdef1234567890
```

描述 AMI 的取代日期

使用 Get-EC2Image cmdlet。若輸出中不存在 DeprecationTime，不會棄用 AMI，或者將其設定為在未來日期棄用。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).DeprecationTime
```

以下為範例輸出。

```
2025-05-01T00:00:00.000Z
```

取消 AMI 棄用

您可取消棄用 AMI，此操作會移除棄用日期與時間。您必須是 AMI 擁有者才能執行此程序。

Console

取消取代 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇 AMI。
3. 從篩選條件列中，選擇 Owned by me (由我擁有)。
4. 選取 AMI，然後依序選取 Actions (動作)、Manage AMI Deprecation (管理 AMI 取代)。您可以選取多個 AMI，一次取消多個 AMI 的取代。
5. 清除啟用核取方塊，然後選擇儲存。

AWS CLI

取消棄用 AMI

請使用下面的 disable-image-deprecation 命令。

```
aws ec2 disable-image-deprecation --image-id ami-0abcdef1234567890
```

PowerShell

取消棄用 AMI

使用 Disable-EC2ImageDeprecation cmdlet。

```
Disable-EC2ImageDeprecation -ImageId ami-0abcdef1234567890
```

停用 Amazon EC2 AMI

可以停用 AMI 以防止其用於執行個體啟動。無法從已停用的 AMI 中啟動新執行個體。可以重新啟用已停用的 AMI，以便在執行個體啟動時再次使用它。

可以同時停用私有和公有 AMI。

如需減少很少使用但需長期保留的停用 EBS 支援的 AMI 的儲存費用，您可封存其關聯的快照。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的檔案 Amazon EBS 快照。

目錄

- AMI 停用運作方式

- 成本
- 先決條件
- 所需的 IAM 許可
- 停用 AMI
- 描述已停用的 AMI
- 重新啟用已停用的 AMI

AMI 停用運作方式

> **Warning**
>
> 停用 AMI 會移除其所有啟動權限。

停用 AMI 時：

- AMI 的狀態會變更為 `disabled`。
- 無法共用已停用的 AMI。如果 AMI 為公有或先前被共用，則會將其設為私有。如果 AMI 與 AWS 帳戶、組織或組織單位共用，則會失去對已停用 AMI 的存取權。
- 依預設，已停用的 AMI 不會出現在 DescribeImages API 呼叫中。
- 已停用的 AMI 不會出現在我所擁有主控台篩選條件下。若要尋找已停用的 AMI，請使用已停用的映像主控台篩選條件。
- 對於 EC2 主控台中的執行個體啟動，不可選擇已停用的 AMI。例如，在啟動執行個體精靈中或建立啟動範本時，已停用的 AMI 不會出現在 AMI 目錄中。
- 啟動服務 (例如啟動範本和 Auto Scaling 群組) 可以繼續參考已停用的 AMI。從已停用的 AMI 中進行後續執行個體啟動將會失敗，因此建議您更新啟動範本和 Auto Scaling 群組以僅參考可用的 AMI。
- 先前使用隨後被停用的 AMI 來啟動的 EC2 執行個體不會受到影響，而且可以停止、啟動和重新啟動。
- 您無法刪除與已停用的 AMI 相關聯的快照。嘗試刪除關聯的快照會導致`snapshot is currently in use`錯誤。

重新啟用 AMI 時：

- AMI 的狀態會變更為 `available`，而且可用來啟動執行個體。

- 可以共用 AMI。
- 由於停用 AMI 而無法對其進行存取的AWS 帳戶、組織和組織單位不會自動重新獲得存取權，但可以再次與他們共用 AMI。

成本

當您停用 AMI 時，不會刪除 AMI。如果 AMI 是 EBS 後端 AMI，您會繼續支付 AMI 的 EBS 快照。如果想要保留 AMI，可以透過封存快照來降低儲存成本。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的檔案 Amazon EBS 快照。如果不想保留 AMI 及其快照，必須取消註冊 AMI 並刪除快照。如需詳細資訊，請參閱取消註冊 AMI。

先決條件

若要停用或重新啟用 AMI，您必須是 AMI 的擁有者。

所需的 IAM 許可

若要停用和重新啟用 AMI，您必須具有下列 IAM 許可：

- `ec2:DisableImage`
- `ec2:EnableImage`

停用 AMI

您可以使用 EC2 主控台或 AWS Command Line Interface () 停用 AMI AWS CLI。您必須是 AMI 擁有者才能執行此程序。

Console

停用 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 AMI。
3. 從篩選條件列中，選擇 Owned by me (由我擁有)。
4. 選取 AMI，然後選擇動作、停用 AMI。可以選取多個 AMI，以便一次停用。
5. 在停用 AMI 視窗中，選擇停用 AMI。

AWS CLI

停用 AMI

使用下列 disable-image 命令。

```
aws ec2 disable-image --image-id ami-0abcdef1234567890
```

PowerShell

停用 AMI

使用 Disable-EC2Image cmdlet。

```
Disable-EC2Image -ImageId ami-0abcdef1234567890
```

描述已停用的 AMI

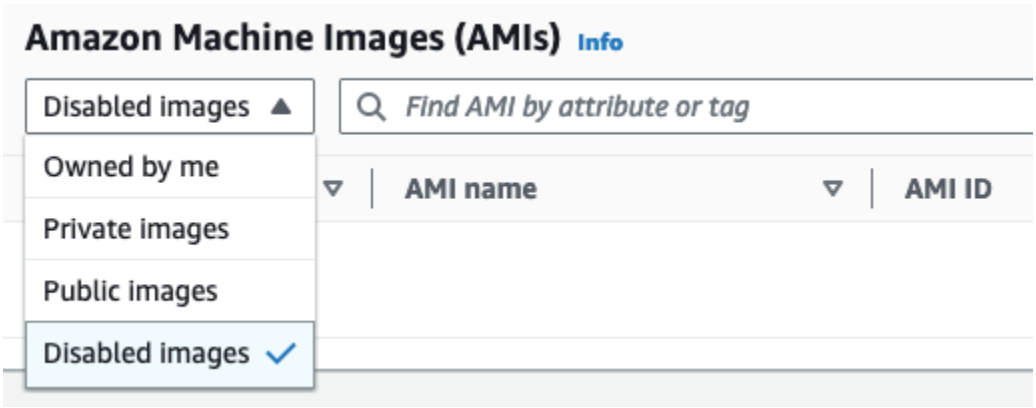
可以使用 EC2 主控台或 AWS CLI來檢視已停用的 AMI。

您必須是 AMI 擁有者，才能檢視已停用的 AMI。因為已停用的 AMI 會設為私有，如果您不是擁有者，就無法檢視已停用的 AMI。

Console

若要檢視已停用的 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 AMI。
3. 從篩選列中，選擇已停用的映像。



AWS CLI

依預設，當您描述所有 AMI 時，停用的 AMI 不會包括在結果中。如需在結果中包括停用的 AMI，請指定 `--include-disabled` 選項。若已停用 AMI，AMI 的 `State` 欄位為 `disabled`。

如需在為帳戶描述所有 AMI 時包括已停用的 AMI

使用下列 describe-images 命令。

```
aws ec2 describe-images \
    --owners 123456789012 \
    --include-disabled
```

如需列出帳戶的停用 AMI

使用下列 describe-images 命令。

```
aws ec2 describe-images \
    --owners self \
    --include-disabled \
    --filters Name=state,Values=disabled \
    --query Images[].ImageId \
    --output text
```

以下為範例輸出。

```
ami-0abcdef1234567890
```

如需描述 AMI 的狀態

使用下列 describe-images 命令。若輸出中不存在 `DeprecationTime`，不會棄用 AMI，或者將其設定為在未來日期棄用。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].State \
    --output text
```

以下為範例輸出。

```
disabled
```

PowerShell

依預設，當您描述所有 AMI 時，停用的 AMI 不會包括在結果中。若要在結果中包含已停用的 AMI，請指定 `-IncludeDisabled` 參數。若已停用 AMI，AMI 的 `State` 欄位為 `disabled`。

如需列出帳戶的停用 AMI

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -Owner self `
    -IncludeDisabled $true | Where-Object {$_.State -eq "disabled"}).ImageId
```

以下為範例輸出。

```
ami-0abcdef1234567890
```

如需描述 AMI 的狀態

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).State.Value
```

以下為範例輸出。

```
disabled
```

重新啟用已停用的 AMI

可以重新啟用已停用的 AMI。您必須是 AMI 擁有者才能執行此程序。

Console

重新啟用已停用的 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 AMI。
3. 從篩選列中，選擇已停用的映像。

4. 選取 AMI，然後選擇動作、啟用 AMI。可以選取多個 AMI，一次重新啟用多個 AMI。
5. 在啟用 AMI 視窗中，選擇啟用。

AWS CLI

重新啟用已停用的 AMI

使用下列 enable-image 命令。

```
aws ec2 enable-image --image-id ami-0abcdef1234567890
```

PowerShell

重新啟用已停用的 AMI

使用 Enable-EC2Image cmdlet。

```
Enable-EC2Image -ImageId ami-0abcdef1234567890
```

取消註冊 Amazon EC2 AMI

當您取消註冊 AMI 時，Amazon EC2 會將其永久刪除。在您取消註冊 AMI 之後，您便無法用它來啟動新的執行個體。您可考慮在完成使用之後取消註冊 AMI。

為了防止意外或惡意取消註冊 AMI，您可以開啟取消註冊保護。如果您意外登出 EBS 支援的 AMI，則只有在永久刪除之前的允許時間內復原，才可以使用資源回收筒來復原。

若取消註冊 AMI，您可選擇同時刪除其關聯的快照。然而，若快照與多個 AMI 關聯，儘管指定要將其刪除，亦不會刪除，但仍會取消註冊 AMI。任何未刪除快照皆會持續產生儲存費用。

取消註冊 AMI 不會影響從 AMI 啟動的任何執行個體。您可以繼續使用這些執行個體。依預設，取消註冊 AMI 亦不會影響 AMI 建立過程之中建立的任何快照。您將需要為這些執行個體繼續支付使用成本，以及快照的儲存成本。因此，為了避免產生不必要的成本，我們建議您終止所有執行個體並刪除不需要的快照。在取消註冊期間，您可自動刪除快照；或在取消註冊之後，手動刪除快照。如需詳細資訊，請參閱避免因未使用的資源而產生成本。

對於從後續取消註冊的 AMI 啟動的執行個體，您仍然可以使用 `describe-instance-image-metadata` AWS CLI 命令檢視一些有關 AMI 的高階資訊。若要了解詳細資訊，請參閱 describe-instance-image-metadata。

目錄

- 考量事項
- 取消註冊 AMI
- 避免因未使用的資源而產生成本
- 保護 Amazon EC2 AMI 免遭註銷

考量事項

- 您無法取消註冊不是您帳戶擁有的 AMI。
- 您無法使用 Amazon EC2 取消註冊由 AWS Backup 服務管理的 AMI。反之，請使用 AWS Backup 刪除備份文件庫中對應的復原點。若要了解詳細資訊，請參閱 AWS Backup 開發人員指南中的刪除備份。

取消註冊 AMI

您可取消註冊 EBS 支援的 AMI 及 Amazon S3 支援的 AMI。若是 EBS 支援的 AMI，您可選擇同時刪除關聯的快照。然後，若快照與其他 AMI 關聯，儘管指定要將其刪除，亦不會刪除。

Console

取消註冊 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 在篩選列中，選擇我擁有以列出可用的 AMI，或選擇已停用的影像以列出已停用的 AMI。
4. 選取要取消註冊的 AMI。
5. 選擇 Actions (動作)、Deregister AMI (取消註冊 AMI)。
6. (選用) 如需在取消註冊期間刪除關聯的快照，勾選刪除關聯快照核取方塊。

   > **Note**
   >
   > 若快照與其他 AMI 關聯，儘管勾選核取方塊，亦不會將其刪除。

7. 選擇取消註冊 AMI。

   主控台從清單移除 AMI 可能需要幾分鐘的時間。選擇 Refresh (重新整理) 以重新整理狀態。

AWS CLI

取消註冊 AMI

請使用下面的 deregister-image 命令。

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

取消註冊 AMI 及刪除其關聯的快照

請使用下面的 deregister-image 命令及指定 `--delete-associated-snapshots` 參數。請注意，若快照與其他 AMI 關聯，儘管指定了此參數，亦不會將其刪除。

```
aws ec2 deregister-image \
    --image-id ami-0abcdef1234567890 \
    --delete-associated-snapshots
```

PowerShell

取消註冊 AMI

使用 Unregister-EC2Image cmdlet。

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

取消註冊 AMI 及刪除其關聯的快照

請使用 Unregister-EC2Image cmdlet 並指定 `-DeleteAssociatedSnapshots` 參數。請注意，若快照與其他 AMI 關聯，儘管指定了此參數，亦不會將其刪除。

```
Unregister-EC2Image `
    -ImageId ami-0abcdef1234567890 `
    -DeleteAssociatedSnapshots
```

避免因未使用的資源而產生成本

依預設，取消註冊 AMI 不會刪除與 AMI 關聯的全部資源。這些資源包括 EBS 支援的 AMI 快照以及 Amazon S3 中 Amazon S3 支援的檔案。當您取消註冊 AMI 時，您也不會終止或停止從 AMI 啟動的任何執行個體。

您將繼續產生儲存快照與檔案的成本，並且將產生任何執行個體的成本。

為了避免產生此類不必要的費用，我們建議刪除任何您不需要的資源。

EBS 支援的 AMI

- 取消註冊 AMI 的同時刪除關聯的快照。如需詳細資訊，請參閱取消註冊 AMI。
- 若取消註冊 AMI 但不刪除其關聯的快照，您可手動刪除快照。在 AMI 建立期間，建立的執行個體根磁碟區快照具有以下說明格式：

```
Created by CreateImage(i-1234567890abcdef0) for ami-0abcdef1234567890
```

- 若您不再需要透過 AMI 啟動的執行個體，可停止或終止這些執行個體。如需列出執行個體，依 AMI 的 ID 進行篩選。

Amazon S3 支援的 AMI

- 使用 ec2-delete-bundle (AMI 工具) 命令刪除 Amazon S3 中的套件。
- 若刪除套件之後 Amazon S3 儲存貯體為空白，且您不再進一步使用該儲存貯體，可刪除儲存貯體。
- 若您不再需要透過 AMI 啟動的執行個體，可終止這些執行個體。如需列出執行個體，依 AMI 的 ID 進行篩選。

保護 Amazon EC2 AMI 免遭註銷

您可以在 AMI 上開啟註銷保護以防止意外或惡意刪除。當您開啟註銷保護時，任何使用者都無法註銷 AMI，無論其 IAM 權限為何。如果您想要註銷 AMI，您必須先關閉其註銷保護。

當您在 AMI 上開啟註銷保護時，您可以選擇包含 24 小時冷卻時間。此冷卻時間是您關閉註銷保護之後，註銷保護仍然有效的時間。在此冷卻時間，無法註銷 AMI。當冷卻時間結束，便可以註銷 AMI。

預設情況下，所有現有與新 AMI 的註銷保護均處於關閉狀態。

開啟註銷保護

使用下列程序來開啟註銷保護。

Console

如需開啟取消註冊保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。

3. 在篩選列中，選擇我擁有以列出可用的 AMI，或選擇已停用的影像以列出已停用的 AMI。
4. 選擇要開啟註銷保護的 AMI，然後選擇執行，管理 AMI 註銷保護。
5. 在管理 AMI 註銷保護對話方塊中，您可以開啟有或沒有冷卻時間的註銷保護。請選擇下列其中一個選項：
   - 啟用 24 小時冷卻時間 — 具有冷卻時間，當註銷保護關閉時，AMI 在 24 小時內無法註銷。
   - 無需冷卻即可啟用 — 無冷卻時間，當註銷保護關閉，AMI 可以立即註銷。
6. 選擇儲存。

AWS CLI

如需開啟取消註冊保護

請使用 enable-image-deregistration-protection 命令。如需啟用包括 `--with-cooldown` 選項在內的選用冷卻時間。

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0abcdef1234567890 \
    --with-cooldown
```

PowerShell

如需開啟取消註冊保護

使用 Enable-EC2ImageDeregistrationProtection cmdlet。如需啟用選用冷卻時間，請設定 `-WithCooldown` 參數為 `true`。

```
Enable-EC2ImageDeregistrationProtection `
    -ImageId ami-0abcdef1234567890 `
    -WithCooldown $true
```

關閉註銷保護

使用下列程序來關閉註銷保護。

如果您在開啟 AMI 註銷保護時選擇包含 24 小時冷卻時間，那麼當您關閉註銷保護時，您將無法立即註銷 AMI。冷卻時間是 24 小時的時間段，在此期間註銷保護即使在關閉之後仍然有效。在此冷卻時間，無法註銷 AMI。當冷卻時間結束，便可註銷 AMI。

Console

如需關閉取消註冊保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 在篩選列中，選擇我擁有以列出可用的 AMI，或選擇已停用的影像以列出已停用的 AMI。
4. 選擇要關閉註銷保護的 AMI，然後選擇執行，管理 AMI 註銷保護。
5. 在管理 AMI 註銷保護對話方塊中，選擇停用。
6. 選擇儲存。

AWS CLI

如需關閉取消註冊保護

請使用 disable-image-deregistration-protection 命令。

```
aws ec2 disable-image-deregistration-protection --image-id ami-0abcdef1234567890
```

PowerShell

如需關閉取消註冊保護

使用 Disable-EC2ImageDeregistrationProtection cmdlet。

```
Disable-EC2ImageDeregistrationProtection -ImageId ami-0abcdef1234567890
```

使用 Amazon EC2 開機模式執行個體啟動行為

電腦開機時，第一個執行的軟體會負責初始化平台，並提供作業系統執行平台特定作業的介面。

在 Amazon EC2 中，支援兩種開機模式軟體：整合可延伸韌體介面 (UEFI) 和 Legacy BIOS。

AMI 上可能的開機模式參數

AMI 可以有下列其中一個開機模式參數值：`uefi`、`legacy-bios` 或 `uefi-preferred`。AMI 開機模式參數為選用。針對沒有開機模式參數的 AMI，從這些 AMI 啟動的執行個體會使用執行個體類型的預設開機模式值。

AMI 開機模式參數的用途

AMI 開機模式參數會向 Amazon EC2 發出訊號，這是在啟動執行個體時要使用的開機模式。當開機模式參數設定為 `uefi` 時，EC2 會嘗試在 UEFI 上啟動執行個體。如果作業系統未設定為支援 UEFI，則執行個體啟動將會失敗。

UEFI 偏好的開機模式參數

您可以使用 `uefi-preferred` 開機模式參數來建立同時支援 UEFI 和 Legacy BIOS 的 AMI。當開機模式參數設定為 `uefi-preferred`，並且如果執行個體類型支援 UEFI，則會在 UEFI 上啟動執行個體。如果執行個體類型不支援 UEFI，則會在 Legacy BIOS 上啟動執行個體。

> **Warning**
>
> 某些功能 (例如 UEFI 安全開機) 僅適用於透過 UEFI 開機的執行個體。當您將 `uefi-preferred` AMI 開機模式參數與不支援 UEFI 的執行個體類型搭配使用時，執行個體會以 Legacy BIOS 的形式啟動，而且會停用與 UEFI 相關的功能。如果您依賴 UEFI 相關功能的可用性，則請將 AMI 開機模式參數設定為 `uefi`。

執行個體類型的預設開機模式

- Graviton 執行個體類型：UEFI
- Intel 和 AMD 執行個體類型：Legacy BIOS

區域支援

在 Wavelength 區域或不支援 UEFI 啟動。

開機模式主題

- 在 UEFI 開機模式啟動 EC2 執行個體的需求
- 確定 Amazon EC2 AMI 的開機模式參數
- 確定 EC2 執行個體類型的支援開機模式
- 確定 EC2 執行個體的開機模式
- 確定 EC2 執行個體作業系統的開機模式
- 設定 Amazon EC2 AMI 的開機模式
- Amazon EC2 執行個體的 UEFI 變數

- Amazon EC2 執行個體的 UEFI 安全開機

在 UEFI 開機模式啟動 EC2 執行個體的需求

執行個體的開機模式取決於 AMI 的組態、其中包含的作業系統以及執行個體類型。若要在 UEFI 開機模式啟動執行個體，您必須符合下列需求。

AMI

必須為 UEFI 設定 AMI，如下所示：

- 作業系統 - AMI 中包含的作業系統必須設定為使用 UEFI；否則，執行個體啟動將會失敗。如需詳細資訊，請參閱確定 EC2 執行個體作業系統的開機模式。
- AMI 開機模式參數 – AMI 的開機模式參數必須設定為 `uefi` 或 `uefi-preferred`。如需詳細資訊，請參閱確定 Amazon EC2 AMI 的開機模式參數。

Linux - 下列 Linux AMI 支援 UEFI：

- Amazon Linux 2023
- Amazon Linux 2 (僅限 Graviton 執行個體類型)

對於其他 Linux AMI，您必須設定 AMI，透過 VM Import/Export 匯入 AMI，或透過 CloudEndure 匯入 AMI。

Windows – 下列 Windows AMI 支援 UEFI：

- Windows_Server-2025-* (以 `BIOS-` 為名稱字首的 AMI 除外)
- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base
- TPM-Windows_Server-2016-English-Core-Base

執行個體類型

在 AWS Nitro 系統上建置的所有執行個體都支援 UEFI 和舊版 BIOS，但以下除外：裸機執行個體、DL1、G4ad、P4、u-3tb1、u-6tb1、u-9tb1、u-12tb1、u-18tb1、u-24tb1 和 VT1。如需詳細資訊，請參閱the section called “執行個體類型開機模式”。

下表顯示執行個體的開機模式 (由產生的執行個體開機模式資料欄表示) 由 AMI 的開機模式參數 (資料欄 1)、AMI 所包含之作業系統的開機模式組態 (資料欄 2) 以及執行個體類型的開機模式支援 (資料欄 3) 的組合決定。

AMI 開機模式參數	作業系統開機模式組態	執行個體類型開機模式支援	產生的執行個體開機模式
UEFI	UEFI	UEFI	UEFI
Legacy BIOS	Legacy BIOS	Legacy BIOS	Legacy BIOS
UEFI Preferred	UEFI	UEFI	UEFI
UEFI Preferred	UEFI	UEFI 與 Legacy BIOS	UEFI
UEFI Preferred	Legacy BIOS	Legacy BIOS	Legacy BIOS
UEFI Preferred	Legacy BIOS	UEFI 與 Legacy BIOS	Legacy BIOS
未指定開機模式 - ARM	UEFI	UEFI	UEFI
未指定開機模式 - x86	Legacy BIOS	UEFI 與 Legacy BIOS	Legacy BIOS

確定 Amazon EC2 AMI 的開機模式參數

AMI 開機模式參數為選用。AMI 可以有下列其中一個開機模式參數值：`uefi`、`legacy-bios` 或 `uefi-preferred`。

有些 AMI 沒有開機模式參數。當 AMI 沒有開機模式參數時，從 AMI 啟動的執行個體會使用執行個體類型的預設值，即 Graviton 上的 `uefi` 以及 Intel 和 AMD 執行個體類型上的 `legacy-bios`。

Console

如需確定 AMI 的開機模式參數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 AMI，然後選取該 AMI。
3. 檢查開機模式欄位。

- 顯示 uefi 值表示 AMI 支援 UEFI。
- 顯示 uefi-preferred 值表示 AMI 支援 UEFI 和 Legacy BIOS。
- 如果未顯示任何值，表示透過 AMI 啟動的執行個體會使用其執行個體類型的預設值。

如需啟動執行個體時確定 AMI 的開機模式參數

使用啟動執行個體精靈啟動執行個體時，在選取 AMI 的步驟中檢查 Boot mode (開機模式) 欄位。如需詳細資訊，請參閱應用程式和作業系統映像 (Amazon Machine Image)。

AWS CLI

如需確定 AMI 的開機模式參數

使用 describe-images 命令確定 AMI 的開機模式。

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-id ami-0abcdef1234567890 \
    --query Images[].BootMode \
    --output text
```

以下為範例輸出。

```
uefi
```

在輸出中，uefi 的值表示 AMI 支援 UEFI。uefi-preferred 的值表示 AMI 支援 UEFI 和 Legacy BIOS。如果未顯示任何值，表示透過 AMI 啟動的執行個體會使用其執行個體類型的預設值。

PowerShell

如需確定 AMI 的開機模式參數

請使用 Get-EC2Image cmdlet 來確定 AMI 的啟動模式。

```
Get-EC2Image -Region us-east-1 `
    -ImageId ami-0abcdef1234567890 | Format-List Name, BootMode, TpmSupport
```

以下為範例輸出。

```
Name       : TPM-Windows_Server-2016-English-Full-Base-2023.05.10
```

```
BootMode   : uefi
TpmSupport : v2.0
```

在輸出中，BootMode 的值會指示 AMI 的啟動模式。uefi 的值表示 AMI 支援 UEFI。uefi-preferred 的值表示 AMI 支援 UEFI 和 Legacy BIOS。如果未顯示任何值，表示透過 AMI 啟動的執行個體會使用其執行個體類型的預設值。

確定 EC2 執行個體類型的支援開機模式

您可確定執行個體類型的支援啟動模式。

Amazon EC2 主控台不顯示執行個體類型的支援啟動模式。

AWS CLI

使用 describe-instance-types 命令確定執行個體類型的支援開機模式。--query 參數會篩選輸出，只傳回支援的開機模式。

下列範例顯示了可同時支援 UEFI 和舊式 BIOS 啟動模式的指定執行個體類型。

```
aws ec2 describe-instance-types \
    --instance-types m5.2xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

以下為範例輸出。

```
[
    [
        "legacy-bios",
        "uefi"
    ]
]
```

下列範例顯示 t2.xlarge 僅支援 Legacy BIOS。

```
aws ec2 describe-instance-types \
    --instance-types t2.xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

以下為範例輸出。

```
[
    [
        "legacy-bios"
    ]
]
```

PowerShell

使用 Get-EC2InstanceType cmdlet 來確定執行個體類型的支援啟動模式。

下列範例顯示 m5.2xlarge 同時支援 UEFI 和 Legacy BIOS 開機模式。

```
Get-EC2InstanceType -InstanceType m5.2xlarge | Format-List InstanceType,
 SupportedBootModes
```

以下為範例輸出。

```
InstanceType       : m5.2xlarge
SupportedBootModes : {legacy-bios, uefi}
```

下列範例顯示 t2.xlarge 僅支援 Legacy BIOS。

```
Get-EC2InstanceType -InstanceType t2.xlarge | Format-List InstanceType,
 SupportedBootModes
```

以下為範例輸出。

```
InstanceType       : t2.xlarge
SupportedBootModes : {legacy-bios}
```

確定支援 UEFI 的執行個體類型

您可確定支援 UEFI 的執行個體類型 Amazon EC2 主控台不顯示執行個體類型的 UEFI 支援。

AWS CLI

可用的執行個體類型因 AWS 區域而異。如需查看區域中支援 UEFI 的可用執行個體類型，請使用 describe-instance-types 命令。包含 --filters 參數以將結果範圍限定為支援 UEFI 的執行個體類型，以及包含 --query 參數以將輸出範圍限定為 InstanceType 的值。

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

可用的執行個體類型因 AWS 區域而異。如需查看區域中支援 UEFI 的可用執行個體類型，請使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType -Filter @{Name="supported-boot-mode"; Values="uefi"} |
 Sort-Object @{E={$_.ProcessorInfo.SupportedArchitectures}}, InstanceType |
 Format-Table InstanceType, SupportedBootModes, `
      @{Name="SupportedArchitectures"; E={$_.ProcessorInfo.SupportedArchitectures}}
```

若要確定支援 UEFI 安全開機並保留非揮發性變數的執行個體類型

裸機執行個體不支援 UEFI 安全開機和非揮發性變數，因此這些範例將其排除在輸出之外。如需有關 UEFI 安全開機的資訊，請參閱 Amazon EC2 執行個體的 UEFI 安全開機。

AWS CLI

請使用 describe-instance-types 命令，然後從輸出中排除裸機執行個體。

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi Name=bare-metal,Values=false \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

請使用 Get-EC2InstanceType cmdlet，然後從輸出中排除裸機執行個體。

```
Get-EC2InstanceType -Filter `
      @{Name="supported-boot-mode"; Values="uefi"}, `
      @{Name="bare-metal"; Values="false"} |
 Sort-Object @{E={$_.ProcessorInfo.SupportedArchitectures}}, InstanceType |
 Format-Table InstanceType, SupportedBootModes, `
      @{Name="SupportedArchitectures"; E={$_.ProcessorInfo.SupportedArchitectures}}
```

確定 EC2 執行個體的開機模式

執行個體的開機模式會顯示在 Amazon EC2 主控台的開機模式欄位中，並透過 AWS CLI中的 `currentInstanceBootMode` 參數顯示。

啟動執行個體時，其啟動模式參數的值取決於用於將其啟動的 AMI 開機模式參數的值，如下所示：

- 具有 `uefi` 開機模式參數的 AMI 會建立 `currentInstanceBootMode` 參數為 `uefi` 的執行個體。
- 具有 `legacy-bios` 開機模式參數的 AMI 會建立 `currentInstanceBootMode` 參數為 `legacy-bios` 的執行個體。
- 如果執行個體類型支援 UEFI，則開機模式參數為 `uefi-preferred` 的 AMI 會建立 `currentInstanceBootMode` 參數為 `uefi` 的執行個體；否則，其會建立 `currentInstanceBootMode` 參數為 `legacy-bios` 的執行個體。
- 沒有開機模式參數值的 AMI 會建立具有 `currentInstanceBootMode` 參數值的執行個體，該參數值取決於 AMI 架構是 ARM 還是 x86，以及支援的執行個體類型開機模式。Graviton 執行個體類型上的預設開機模式為 `uefi`，Intel 和 AMD 執行個體類型上的預設開機模式為 `legacy-bios`。

Console

如需確定執行個體的啟動模式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 在 Details (詳細資訊) 標籤上，檢查 Boot mode (開機模式) 欄位。

AWS CLI

如需確定執行個體的啟動模式

使用 describe-instances 命令確定執行個體的開機模式。您也可以判斷用來建立執行個體之 AMI 的開機模式。

```
aws ec2 describe-instances \
    --region us-east-1 \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BootMode \
    --output text
```

以下為範例輸出。

```
uefi
```

PowerShell

如需確定執行個體的啟動模式

請使用 Get-EC2Image cmdlet 來確定執行個體的開機模式。您也可以判斷用來建立執行個體之 AMI 的開機模式。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | Format-List BootMode,
 CurrentInstanceBootMode, InstanceType, ImageId
```

以下為範例輸出。

```
BootMode                : uefi
CurrentInstanceBootMode : uefi
InstanceType            : c5a.large
ImageId                 : ami-0abcdef1234567890
```

確定 EC2 執行個體作業系統的開機模式

AMI 的開機模式會引導 Amazon EC2 使用哪種開機模式來啟動執行個體。若要檢視執行個體的作業系統是否已針對 UEFI 進行設定，您必須使用 SSH (Linux 執行個體) 或 RDP (Windows 執行個體) 連線至您的執行個體。

請遵循於執行個體作業系統的指示。

Linux

確定執行個體作業系統的開機模式

1. 使用 SSH 連線至您的 Linux 執行個體。
2. 若要檢視作業系統的開機模式，請嘗試下列其中一個動作：

   - 執行下列命令。

     ```
     [ec2-user ~]$ sudo /usr/sbin/efibootmgr
     ```

在 UEFI 開機模式下啟動的執行個體預期輸出

```
BootCurrent: 0001
Timeout: 0 seconds
BootOrder: 0000,0001
Boot0000* UiApp
Boot0001* UEFI Amazon Elastic Block Store vol-xyz
```

- 執行下列命令，以驗證 /sys/firmware/efi 目錄是否存在。只有在執行個體使用 UEFI 開機時，才會存在此目錄。如果此目錄不存在，命令會傳回 Legacy BIOS Boot Detected。

```
[ec2-user ~]$ [ -d /sys/firmware/efi ] && echo "UEFI Boot Detected" || echo
 "Legacy BIOS Boot Detected"
```

在 UEFI 開機模式下啟動的執行個體預期輸出

```
UEFI Boot Detected
```

在 Legacy BIOS 開機模式下啟動的執行個體預期輸出

```
Legacy BIOS Boot Detected
```

- 執行下列命令，以驗證 EFI 是否出現在 dmesg 輸出中。

```
[ec2-user ~]$ dmesg | grep -i "EFI"
```

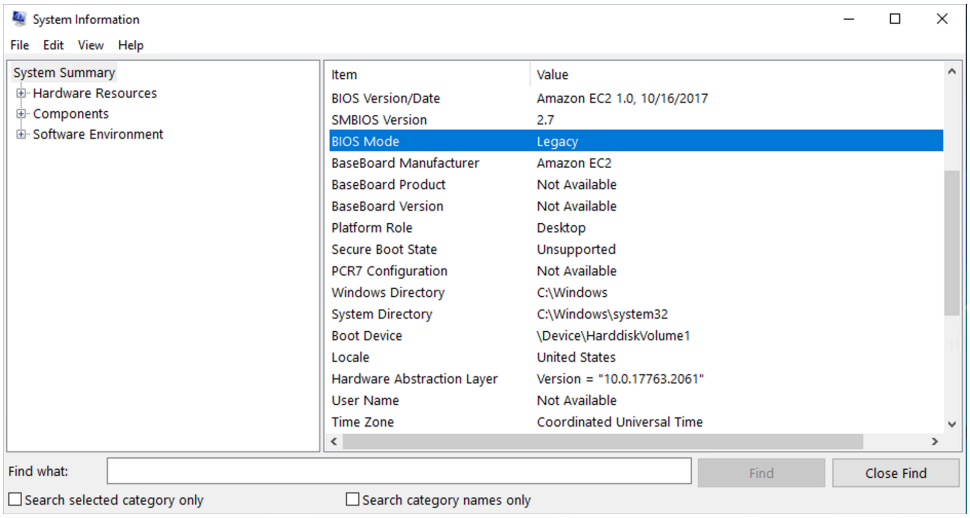
在 UEFI 開機模式下啟動的執行個體預期輸出

```
[    0.000000] efi: Getting EFI parameters from FDT:
[    0.000000] efi: EFI v2.70 by EDK II
```

Windows

確定執行個體作業系統的開機模式

1. 使用 RDP 連線至您的 Windows 執行個體。
2. 請移至 System Information (系統資訊)，然後檢查 BIOS Mode (BIOS 模式) 列。



設定 Amazon EC2 AMI 的開機模式

AMI 預設會繼承用於建立 AMI 的 EC2 執行個體的啟動模式。例如，若透過舊式 BIOS 上執行的 EC2 執行個體來建立 AMI，新 AMI 的啟動模式為 `legacy-bios`。若透過使用啟動模式 `uefi-preferred` 的 EC2 執行個體來建立 AMI，新 AMI 的啟動模式為 `uefi-preferred`。

當您註冊 AMI 時，可將 AMI 的啟動模式設定為 `uefi`、`legacy-bios` 或 `uefi-preferred`。

當 AMI 開機模式設為 `uefi-preferred` 時，執行個體會以下列方式開機：

- 對於同時支援 UEFI 和 Legacy BIOS (例如 `m5.large`) 的執行個體類型，執行個體會使用 UEFI 開機。
- 對於僅支援 Legacy BIOS (例如 `m4.large`) 的執行個體類型，執行個體會使用 Legacy BIOS 開機。

如果您將 AMI 開機模式設定為 `uefi-preferred`，則作業系統必須支援 UEFI 和 Legacy BIOS 開機功能。

若要將現有的 Legacy BIOS 型執行個體轉換為 UEFI，或將現有的 UEFI 型執行個體轉換為 Legacy BIOS，您必須先修改執行個體的磁碟區和作業系統，以支援選取的開機模式。然後，建立磁碟區的快照。最後，透過快照建立 AMI

考量事項

- 設定 AMI 開機模式參數並不會自動針對指定的開機模式設定作業系統。您必須先對執行個體的磁碟區和作業系統進行適當的修改，以支援使用選取的啟動模式來啟動。否則，產生的 AMI 將不可使用。例如，如果您要將 Legacy BIOS Windows 執行個體轉換為 UEFI，可以使用 Microsoft 的

MBR2GPT 工具，將系統磁碟從 MBR 轉換為 GPT。所需的修改視作業系統而異。如需詳細資訊，請參閱作業系統的手冊。

- 您不能使用 register-image 命令或 Register-EC2Image cmdlet 來建立同時支援 NitroTPM 和 UEFI Preferred 的 AMI。
- 某些功能 (例如 UEFI 安全開機) 僅適用於透過 UEFI 開機的執行個體。當您將 `uefi-preferred` AMI 開機模式參數與不支援 UEFI 的執行個體類型搭配使用時，執行個體會以 Legacy BIOS 的形式啟動，而且會停用與 UEFI 相關的功能。如果您依賴 UEFI 相關功能的可用性，則請將 AMI 開機模式參數設定為 `uefi`。

AWS CLI

如需設定 AMI 的啟動模式

1. 對執行個體的磁碟區和作業系統進行適當的修改，以使用選取的開機模式支援開機。所需的修改視作業系統而異。如需詳細資訊，請參閱作業系統的手冊。

   > **Warning**
   >
   > 如果您不執行此步驟，則 AMI 將無法使用。

2. 若要尋找執行個體的磁碟區 ID，請使用 describe-instances 命令。您將在下一個步驟中建立此磁碟區的快照。

   ```
   aws ec2 describe-instances \
       --instance-ids i-1234567890abcdef0 \
       --query Reservations[].Instances[].BlockDeviceMappings
   ```

   以下為範例輸出。

   ```
   [
       [
           {
               "DeviceName": "/dev/xvda",
               "Ebs": {
                   "AttachTime": "2024-07-11T01:05:51+00:00",
                   "DeleteOnTermination": true,
                   "Status": "attached",
                   "VolumeId": "vol-1234567890abcdef0"
               }
   ```

```
        }
    ]
]
```

3. 若要建立磁碟區的快照，請使用 create-snapshot 命令。使用上一個步驟的磁碟區 ID。

```
aws ec2 create-snapshot \
    --volume-id vol-01234567890abcdef \
    --description "my snapshot"
```

以下為範例輸出。

```
{
    "Description": "my snapshot",
    "Encrypted": false,
    "OwnerId": "123456789012",
    "Progress": "",
    "SnapshotId": "snap-0abcdef1234567890",
    "StartTime": "",
    "State": "pending",
    "VolumeId": "vol-01234567890abcdef",
    "VolumeSize": 30,
    "Tags": []
}
```

4. 等到快照的狀態為 completed 後，再進行下一個步驟。如需獲取快照狀態，請使用上一個步驟的快照 ID 並搭配 describe-snapshots 命令。

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-0abcdef1234567890 \
    --query Snapshots[].State \
    --output text
```

以下為範例輸出。

```
completed
```

5. 若要建立新 AMI，請使用 register-image 命令。使用輸出 CreateSnapshot 中的值 SnapshotId。

   - 如需將啟動模式設定為 UEFI，請新增包含值 uefi 的 --boot-mode 參數。

```
aws ec2 register-image \
   --description "my image" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
   --ena-support \
   --boot-mode uefi
```

- 如需將啟動模式設定為 uefi-preferred，請將 --boot-mode 的值設定為 uefi-preferred

```
aws ec2 register-image \
   --description "my description" \
   --name "my-image" \
   --block-device-mappings "DeviceName=/dev/
sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
   --root-device-name /dev/sda1 \
   --virtualization-type hvm \
   --ena-support \
   --boot-mode uefi-preferred
```

6. (選用) 如需驗證新建立的 AMI 是否具有您指定的啟動模式，請使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-1234567890abcdef0 \
    --query Images[].BootMode \
    --output text
```

以下為範例輸出。

```
uefi
```

PowerShell

如需設定 AMI 的啟動模式

1. 對執行個體的磁碟區和作業系統進行適當的修改，以使用選取的開機模式支援開機。所需的修改視作業系統而異。如需詳細資訊，請參閱作業系統的手冊。

> **Warning**
>
> 如果您不執行此步驟，則 AMI 將無法使用。

2. 如需尋找執行個體的磁碟區 ID，請使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

以下為範例輸出。

```
AssociatedResource  :
AttachTime          : 7/11/2024 1:05:51 AM
DeleteOnTermination : True
Operator            :
Status              : attached
VolumeId            : vol-01234567890abcdef
```

3. 如需建立磁碟區的快照，請使用 New-EC2Snapshot cmdlet。使用上一個步驟的磁碟區 ID。

```
New-EC2Snapshot `
    -VolumeId vol-01234567890abcdef `
    -Description "my snapshot"
```

以下為範例輸出。

```
AvailabilityZone          :
Description               : my snapshot
Encrypted                 : False
FullSnapshotSizeInBytes   : 0
KmsKeyId                  :
OwnerId                   : 123456789012
RestoreExpiryTime         :
SnapshotId                : snap-0abcdef1234567890
SseType                   :
StartTime                 : 4/25/2025 6:08:59 PM
State                     : pending
StateMessage              :
VolumeId                  : vol-01234567890abcdef
VolumeSize                : 30
```

4. 等到快照的狀態為 completed 後，再進行下一個步驟。如需獲取快照狀態，請使用上一個步驟的快照 ID 並搭配 Get-EC2Snapshot cmdlet。

```
(Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890).State.Value
```

以下為範例輸出。

```
completed
```

5. 如需使用 Register-EC2Image cmdlet 來建立新的 AMI。使用輸出 New-EC2Snapshot 中的值 SnapshotId。

- 如需將啟動模式設定為 UEFI，請新增包含值 uefi 的 -BootMode 參數。

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
   -Description "my image" `
   -Name "my-image" `
   -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
   -RootDeviceName /dev/xvda `
   -EnaSupport $true `
   -BootMode uefi
```

- 如需將啟動模式設定為 uefi-preferred，請將 -BootMode 的值設定為 uefi-preferred

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
   -Description "my image" `
   -Name "my-image" `
   -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
   -RootDeviceName /dev/xvda `
   -EnaSupport $true `
   -BootMode uefi-preferred
```

6. (選用) 如需驗證新建立的 AMI 是否具有您指定的啟動模式，請使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -ImageId ami-1234567890abcdef0).BootMode.Value
```

以下為範例輸出。

```
uefi
```

Amazon EC2 執行個體的 UEFI 變數

啟動其中啟動模式設定為 UEFI 的執行個體時，就會建立變數的機碼值存放區。UEFI 和執行個體作業系統可以使用存放區來儲存 UEFI 變數。

開機載入器和作業系統會使用 UEFI 變數來設定早期系統啟動。它們允許作業系統管理開機程序的特定設定，例如開機順序，或管理 UEFI 安全開機的金鑰。

> **Warning**
>
> 任何可以連線至執行個體 (以及可能會在執行個體上運行的任何軟體)，或任何具有在執行個體上使用 GetInstanceUefiData API 許可的人員都可以讀取變數。您決不應該將敏感資料 (例如密碼或個人識別資訊) 存放在 UEFI 變數存放區中。

UEFI 變數持久性

- 對於在 2022 年 5 月 10 日或之前啟動的執行個體，UEFI 變數將在重新開機或停止時被抹除。
- 對於在 2022 年 5 月 11 日或之後啟動的執行個體，標示為非揮發性的 UEFI 變數會在重新開機和停止/啟動時持續存在。
- 裸機執行個體不會在執行個體停止/啟動操作中保留 UEFI 非揮發性變數。

Amazon EC2 執行個體的 UEFI 安全開機

UEFI 安全開機依托 Amazon EC2 的長期安全開機程序，提供額外的深度防禦，幫助客戶保護軟體免受重新開機期間持續存在的威脅。它確保執行個體僅對使用密碼編譯金鑰簽署的軟體進行開機。這些金鑰存放在 UEFI 非揮發性變數存放區的金鑰資料庫中。UEFI 安全開機可避免對執行個體開機流程進行未經授權的修改。

目錄

- UEFI 安全開機如何與 Amazon EC2 執行個體搭配使用
- Amazon EC2 的 UEFI 安全啟動要求

- 驗證 Amazon EC2 執行個體是否已啟用 UEFI 安全開機
- 使用自訂 UEFI 安全開機金鑰建立 Linux AMI
- 建立 AWS UEFI 安全開機的二進位 Blob

UEFI 安全開機如何與 Amazon EC2 執行個體搭配使用

UEFI 安全開機是 UEFI 中指定的一項功能，它提供有關開機鏈狀態的驗證。該功能旨在確保韌體自我初始化後，只會執行以密碼編譯驗證的 UEFI 二進位檔案。這些二進位檔案包括 UEFI 驅動程式和主要開機載入器，以及鏈式載入元件。

UEFI 安全開機指定在信任鏈中使用的四個金鑰資料庫。資料庫存放在 UEFI 變數存放區中。

信任鏈如下：

平台金鑰 (PK) 資料庫

PK 資料庫是信任根。它包含在信任鏈中用於更新金鑰交換金鑰 (KEK) 資料庫的單一公有 PK 金鑰。

若要變更 PK 資料庫，您必須具有私有 PK 金鑰才能簽署更新要求。這包括透過寫入空 PK 金鑰來刪除 PK 資料庫。

金鑰交換金鑰 (KEK) 資料庫

KEK 資料庫是一份公有 KEK 金鑰清單，這些金鑰用於在信任鏈中更新簽章 (db) 和拒絕清單 (dbx) 資料庫。

若要變更公有 KEK 資料庫，您必須具有私有 PK 金鑰才能簽署更新要求。

簽章 (db) 資料庫

db 資料庫是一份公有金鑰和雜湊清單，這些金鑰和雜湊用於在信任鏈中驗證所有 UEFI 開機二進位檔案。

若要變更 db 資料庫，您必須具有私有 PK 金鑰或任何一項私有 KEK 金鑰才能簽署更新要求。

簽章拒絕清單 (dbx) 資料庫

dbx 資料庫是一份不受信任的公有金鑰和二進位檔案雜湊清單，在信任鏈中作為撤銷檔案使用。

dbx 資料庫始終優先於所有其他金鑰資料庫。

若要變更 dbx 資料庫，您必須具有私有 PK 金鑰或任何一項私有 KEK 金鑰才能簽署更新要求。

UEFI 論壇維護有適用於許多已知不良二進位檔案和憑證的公開可用的 dbx，網址為：https://uefi.org/revocationlistfile。

> **Important**
>
> UEFI 安全開機強制對任何 UEFI 二進位檔案執行簽章驗證。若要允許在 UEFI 安全開機中執行 UEFI 二進位檔案，請使用上述任何私有 db 金鑰予以簽署。

UEFI 安全開機預設為停用，且系統處於 SetupMode 模式。當系統處於 SetupMode 模式時，所有金鑰變數都可以在沒有密碼編譯簽章的情況下更新。設定 PK 後，UEFI 安全開機將啟用，並結束設定模式。

Amazon EC2 的 UEFI 安全啟動要求

當您啟動 Amazon EC2 執行個體，且其具支援 AMI 和支援執行個體類型時，該執行個體會自動根據其 UEFI 安全開機資料庫驗證 UEFI 開機二進位檔案。不需任何其他設定。您還可以在啟動後在執行個體上設定 UEFI 安全開機。

> **Note**
>
> UEFI 安全開機可保護您的執行個體及其作業系統免受開機流程修改的影響。如果您從啟用 UEFI 安全開機的來源 AMI 建立新 AMI，並在複製過程修改部分參數 (例如變更 AMI 內的 UefiData)，您可停用 UEFI 安全開機。

目錄

- 支援的 AMI
- 支援的執行個體類型

支援的 AMI

Linux AMI

若要啟動 Linux 執行個體，Linux AMI 必須啟用 UEFI 安全開機。

Amazon Linux 從 AL2023 2023.1 版起開始支援 UEFI 安全開機。不過，AMI 在預設情況下並不會啟用 UEFI 安全開機。如需詳細資訊，請參閱「AL2023 使用者指南」中的 UEFI 安全開機。舊版 Amazon

Linux AMI 並未啟用 UEFI 安全開機。若要使用支援的 AMI，必須在自有的 Linux AMI 上執行多個設定步驟。如需詳細資訊，請參閱使用自訂 UEFI 安全開機金鑰建立 Linux AMI。

Windows AMI

若要啟動 Windows 執行個體，Windows AMI 必須啟用 UEFI 安全開機。若要尋找已針對使用 Microsoft 金鑰的 UEFI 安全開機預先設定的 AWS Windows AMI，請參閱 Windows AMIs參考中的尋找使用 NitroTPM 和 UEFI 安全開機設定的 Windows Server AMI。 AWS AMIs

目前，我們不支援使用 import-image 命令來匯入具有 UEFI 安全開機的 Windows。

支援的執行個體類型

所有支援 UEFI 的虛擬化執行個體類型也支援 UEFI 安全開機。如需有關支援 UEFI 安全開機的執行個體類型詳細資訊，請參閱UEFI 開機模式需求。

> **Note**
>
> 裸機執行個體類型不支援 UEFI 安全開機。

驗證 Amazon EC2 執行個體是否已啟用 UEFI 安全開機

您可使用下列程序來確定 Amazon EC2 是否已啟用 UEFI 安全開機。

Linux 執行個體

可使用 `mokutil` 公用程式來驗證 Linux 執行個體是否已啟用 UEFI 安全開機。如果執行個體上沒有安裝 `mokutil`，則必須安裝。如需 Amazon Linux 2 安裝說明，請參閱在 Amazon Linux 2 執行個體尋找並安裝軟體套件。若是其他 Linux 發行版，請參閱其特定文件。

驗證 Linux 執行個體是否已啟用 UEFI 安全開機

連線您的執行個體，並在終端機視窗以 `root` 身分執行以下命令。

```
mokutil --sb-state
```

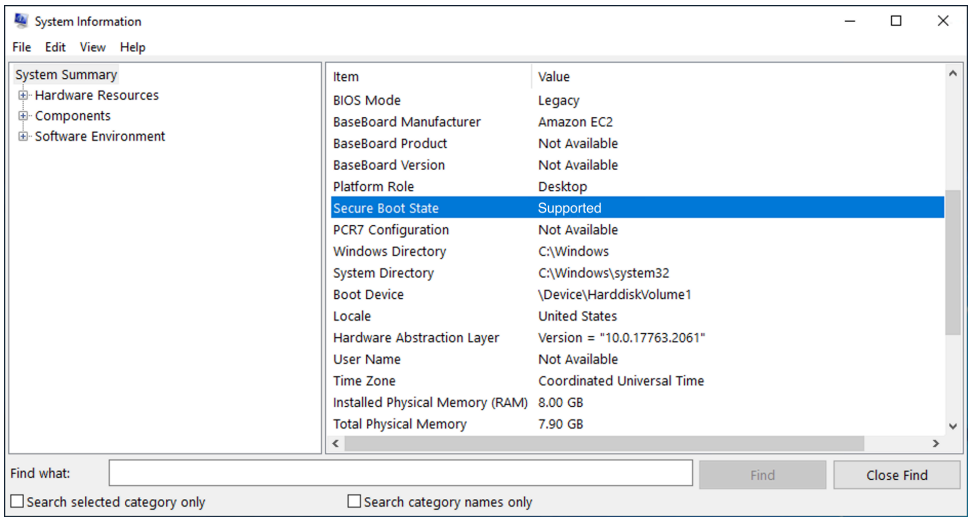
以下為範例輸出。

- 如果啟用了 UEFI 安全開機，則輸出會包含 `SecureBoot enabled`。
- 如果未啟用 UEFI 安全開機，則輸出會包含 `SecureBoot disabled` 或 `Failed to read SecureBoot`。

Windows 執行個體

驗證 Windows 執行個體是否已啟用 UEFI 安全開機

1. 連線到您的執行個體。
2. 開啟 msinfo32 工具。
3. 檢查 Secure Boot State (安全開機狀態) 欄位。如果啟用了 UEFI 安全開機，則值為支援，如下圖所示。



您也可以使用 Windows PowerShell Cmdlet `Confirm-SecureBootUEFI` 檢查安全開機狀態。如需有關 cmdlet 的詳細資訊，請參閱 Microsoft 文件的 Confirm-SecureBootUEFI。

使用自訂 UEFI 安全開機金鑰建立 Linux AMI

這些說明向您展示了如何使用 UEFI 安全開機和自訂私有金鑰來建立 Linux AMI。Amazon Linux 從 AL2023 2023.1 版起開始支援 UEFI 安全開機。如需詳細資訊，請參閱 Amazon Linux 2023 使用者指南中的 AL2023 的 UEFI 安全啟動。

> **Important**
>
> 以下程序僅適用於進階使用者。您必須具備充分的 SSL 和 Linux 發行版本開機流程的知識才能使用這些程序。

先決條件

- 將會使用下列工具：

  - OpenSSL - https://www.openssl.org/
  - efivar - https://github.com/rhboot/efivar
  - efitools - https://git.kernel.org/pub/scm/linux/kernel/git/jejb/efitools.git/
  - get-instance-uefi-data 命令
- 您的 Linux 執行個體必須使用支援 UEFI 開機模式的 Linux AMI 啟動，並且有非揮發性資料存在。

沒有 UEFI 安全開機金鑰的新建立執行個體是在 SetupMode 中建立，允許您註冊自己的金鑰。某些 AMI 預先設定為具有 UEFI 安全開機功能，您無法變更現有金鑰。如果您想要變更金鑰，您必須根據原始 AMI 建立新 AMI。

您有兩種在變數存放區傳播金鑰的方法，這些方法在接下來的選項 A 和選項 B 中說明。選項 A 說明如何在執行個體中執行此操作，並模仿真實硬體的流程。選項 B 說明如何建立二進位檔案 Blob，其隨後會在建立 AMI 時作為 base64 編碼的檔案傳遞。對於這兩個選項，您必須先建立用於信任鏈的三對金鑰對。

若要建立支援 UEFI 安全開機的 Linux AMI，請先建立三個金鑰對，然後完成選項 A 或選項 B，但不能同時完成兩者：

- 任務 1：建立金鑰對
- 任務 2 – 選項 A：從執行個體中將金鑰新增至變數存放區
- 任務 2 – 選項 B：建立一個包含預先填寫的變數存放區的二進位 blob

任務 1：建立金鑰對

UEFI 安全開機基於下列用於信任鏈的三個金鑰資料庫：平台金鑰 (PK)、金鑰交換金鑰 (KEK) 和簽章資料庫 (db)。[1]

您需在執行個體上建立每個金鑰。若要以對 UEFI 安全開機標準有效的格式準備公有金鑰，請為每個金鑰建立一個憑證。DER 定義 SSL 格式 (格式的二進位編碼)。然後，您可以將每個憑證轉換為 UEFI 簽章清單，該清單為 UEFI 安全開機中所理解的二進位格式。最後，您需使用相關金鑰簽署每個憑證。

任務

- 準備建立金鑰對
- 金鑰對 1：建立平台金鑰 (PK)
- 金鑰對 2：建立金鑰交換金鑰 (KEK)
- 金鑰對 3：建立簽章資料庫 (db)

- 使用私有金鑰簽署開機映像 (核心)

準備建立金鑰對

建立金鑰對之前，請建立一個用於金鑰產生的全域唯一識別符 (GUID)。

1. 連線到執行個體。
2. 在 shell 提示中執行下列命令。

```
uuidgen --random > GUID.txt
```

金鑰對 1：建立平台金鑰 (PK)

PK 是 UEFI 安全開機執行個體的信任根。私有 PK 用於更新 KEK，其又可接續用於將授權的金鑰新增至簽章資料庫 (db)。

X.509 標準用於建立金鑰對。如需有關標準的資訊，請參閱維基百科上的 X.509。

建立 PK

1. 建立金鑰。您必須將變數命名為 PK。

```
openssl req -newkey rsa:4096 -nodes -keyout PK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Platform key/" -out PK.crt
```

   指定了下列參數：

   - `-keyout PK.key` - 私有金鑰檔案。
   - `-days 3650` - 憑證有效的天數。
   - `-out PK.crt` - 用於建立 UEFI 變數的憑證。
   - `CN=Platform key` - 金鑰的通用名稱 (CN)。您可以輸入自己的組織名稱，而不是####。
2. 建立憑證。

```
openssl x509 -outform DER -in PK.crt -out PK.cer
```

3. 將憑證轉換至 UEFI 簽章清單。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" PK.crt PK.esl
```

4. 使用私有 PK (自我簽署) 簽署 UEFI 簽章清單。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt PK PK.esl PK.auth
```

金鑰對 2：建立金鑰交換金鑰 (KEK)

私有 KEK 用於將金鑰新增至 db，這是要在系統上開機使用的授權簽章清單。

建立 KEK

1. 建立金鑰。

```
openssl req -newkey rsa:4096 -nodes -keyout KEK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Key Exchange Key/" -out KEK.crt
```

2. 建立憑證。

```
openssl x509 -outform DER -in KEK.crt -out KEK.cer
```

3. 將憑證轉換至 UEFI 簽章清單。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" KEK.crt KEK.esl
```

4. 使用私有 PK 簽署簽章清單。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt KEK KEK.esl KEK.auth
```

金鑰對 3：建立簽章資料庫 (db)

db 清單包含授權在系統上開機的授權金鑰。必須使用私有 KEK 才能修改清單。將使用在此步驟中建立的私有金鑰簽署開機映像。

建立 db

1. 建立金鑰。

```
openssl req -newkey rsa:4096 -nodes -keyout db.key -new -x509 -sha256 -days 3650 -
subj "/CN=Signature Database key/" -out db.crt
```

2. 建立憑證。

```
openssl x509 -outform DER -in db.crt -out db.cer
```

3. 將憑證轉換至 UEFI 簽章清單。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" db.crt db.esl
```

4. 使用私有 KEK 簽署簽章清單。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k KEK.key -c KEK.crt db db.esl db.auth
```

使用私有金鑰簽署開機映像 (核心)

若為 Ubuntu 22.04，下列映像需要簽章。

```
/boot/efi/EFI/ubuntu/shimx64.efi
/boot/efi/EFI/ubuntu/mmx64.efi
/boot/efi/EFI/ubuntu/grubx64.efi
/boot/vmlinuz
```

簽署映像

使用下列語法簽署映像。

```
sbsign --key db.key --cert db.crt --output /boot/vmlinuz /boot/vmlinuz
```

> **Note**
>
> 您必須簽署所有新核心。*/boot/vmlinuz* 通常會以符號連結到上次安裝的核心。

請參閱您的發行版本說明文件，了解您的開機鏈和所需映像。

[1] 感謝 ArchWiki 社群所做的所有工作。用於建立 PK、建立 KEK、建立 DB 和簽署映像的命令來自 Creating keys (建立金鑰)，該文章由 ArchWiki 維護團隊和/或 ArchWiki 貢獻者撰寫。

任務 2 – 選項 A：從執行個體中將金鑰新增至變數存放區

建立三個金鑰對後，您即可以連接到執行個體，並透過完成以下步驟從執行個體中將金鑰新增至變數存放區。或者，完成 the section called “任務 2 – 選項 B：建立一個包含預先填寫的變數存放區的二進位 blob” 中的步驟。

選項 A 步驟：

- 步驟 1：啟動將支援 UEFI 安全開機的執行個體
- 步驟 2：設定執行個體以支援 UEFI 安全開機
- 步驟 3：從執行個體建立 AMI

步驟 1：啟動將支援 UEFI 安全開機的執行個體

當您啟動執行個體 (具有下列先決條件) 時，執行個體將準備好設定為支援 UEFI 安全開機。您只能在啟動時在執行個體上啟用對 UEFI 安全開機的支援；以後無法啟用它。

先決條件

- AMI - Linux AMI 必須支援 UEFI 開機模式。若要驗證 AMI 是否支援 UEFI 開機模式，AMI 開機模式參數必須為 uefi。如需詳細資訊，請參閱確定 Amazon EC2 AMI 的開機模式參數。

  請注意， AWS 僅提供設定為支援 Graviton 型執行個體類型的 UEFI 的 Linux AMIs。 AWS 目前不提供支援 UEFI 開機模式的 x86_64 Linux AMIs。您可以設定自有的 AMI，以支援所有架構的 UEFI 開機模式。若要設定支援 UEFI 開機模式的 AMI，必須在自有的 AMI 上執行多個設定步驟。如需詳細資訊，請參閱設定 Amazon EC2 AMI 的開機模式。
- 執行個體類型 - 所有支援 UEFI 的虛擬化執行個體類型也支援 UEFI 安全開機。裸機執行個體類型不支援 UEFI 安全開機。如需有關支援 UEFI 安全開機的執行個體類型詳細資訊，請參閱UEFI 開機模式需求。
- 在發佈 UEFI 安全開機後啟動您的執行個體。只有在 2022 年 5 月 10 日 (發佈 UEFI 安全開機時) 後啟動的執行個體才能支援 UEFI 安全開機。

啟動執行個體後，您可以透過檢查 UEFI 資料是否存在，驗證執行個體是否已準備好設定為支援 UEFI 安全開機 (換句話說，您可以繼續執行步驟 2)。UEFI 資料的存在表示非揮發性資料是持續的。

驗證執行個體是否準備好執行步驟 2

使用 get-instance-uefi-data 命令並指定執行個體 ID。

```
aws ec2 get-instance-uefi-data --instance-id i-1234567890abcdef0
```

如果輸出中存在 UEFI 資料，則執行個體已準備好執行步驟 2。如果輸出為空白，則無法將執行個體設定為支援 UEFI 安全開機。如果您的執行個體在 UEFI 安全開機支援可用之前啟動，則會發生這種情況。啟動新的執行個體，然後再試一次。

步驟 2：設定執行個體以支援 UEFI 安全開機

在執行個體上的 UEFI 變數存放區中註冊金鑰對

> **Warning**
>
> 您必須在註冊金鑰之後簽署開機映像，否則您將無法啟動執行個體。

建立已簽署的 UEFI 簽章清單 (PK、KEK 以及db) 後，必須將他們註冊至 UEFI 韌體。

只有在以下情況下才能寫入 PK 變數：

- 如果 SetupMode 變數為 1，表示尚未註冊 PK。透過以下命令來檢查這一點：輸出為 1 或 0。

  ```
  efivar -d -n 8be4df61-93ca-11d2-aa0d-00e098032b8c-SetupMode
  ```

- 新 PK 由現有 PK 的私有金鑰簽署。

在 UEFI 變數存放區中註冊金鑰

必須在執行個體上執行以下命令。

如果啟用了設定模式 (值為 1)，則可以在執行個體上執行下列命令來註冊金鑰：

```
[ec2-user ~]$ efi-updatevar -f db.auth db
```

```
[ec2-user ~]$ efi-updatevar -f KEK.auth KEK
```

```
[ec2-user ~]$ efi-updatevar -f PK.auth PK
```

驗證 UEFI 安全開機是否已啟用

若要驗證 UEFI 安全開機是否已啟用，請依照 驗證 Amazon EC2 執行個體是否已啟用 UEFI 安全開機 中的步驟執行動作。

現在，您可以使用 get-instance-uefi-data CLI 命令匯出 UEFI 變數存放區，或者繼續執行下一個步驟，簽署開機映像以重新開機到已啟用 UEFI 安全開機的執行個體。

步驟 3：從執行個體建立 AMI

若要從執行個體建立 AMI，您可以使用主控台或 `CreateImage` API、CLI 或 SDK。如需主控台的說明，請參閱 建立 Amazon EBS 支援的 AMI。如需 API 的說明，請參閱 CreateImage。

> **Note**
>
> `CreateImage` API 會自動將執行個體的 UEFI 變數存放區複製到 AMI。控制台使用 `CreateImage` API。使用此 AMI 啟動執行個體後，執行個體將具有相同的 UEFI 變數存放區。

任務 2 – 選項 B：建立一個包含預先填寫的變數存放區的二進位 blob

建立三個金鑰對之後，您可以建立一個包含預先填寫的變數存放區 (包含 UEFI 安全開機金鑰) 的二進位 blob。或者，完成 the section called “任務 2 – 選項 A：從執行個體中將金鑰新增至變數存放區” 中的步驟。

> **Warning**
>
> 您必須在註冊金鑰之前簽署開機映像，否則您將無法啟動執行個體。

選項 B 步驟：

- 步驟 1：建立新的變數存放區或更新現有變數存放區
- 步驟 2：在建立 AMI 時上傳二進位 blob

步驟 1：建立新的變數存放區或更新現有變數存放區

您可以使用 python-uefivars 工具，在沒有執行個體執行的情況下離線建立變數存放區。該工具可以從您的金鑰建立一個新的變數存放區。指令碼目前支援 EDK2 格式、 AWS 格式和 JSON 表示法，以便使用高階工具進行編輯。

在沒有執行個體執行的情況下離線建立變數存放區

1. 在下列連結下載該工具。

```
https://github.com/awslabs/python-uefivars
```

2. 通過執行下列命令從您的金鑰建立新的變數存放區。這將在 *your_binary_blob*.bin 中建立 base64 編碼二進位 blob。此工具也支援使用 -I 參數更新二進位 Blob。

```
./uefivars.py -i none -o aws -O your_binary_blob.bin -P PK.esl -K KEK.esl --db
 db.esl --dbx dbx.esl
```

步驟 2：在建立 AMI 時上傳二進位 blob

使用 register-image 傳遞您的 UEFI 變數存放區資料。對於 --uefi-data 參數，請指定您的二進位 blob，而對於 --boot-mode 參數，則指定 uefi。

```
aws ec2 register-image \
    --name uefi_sb_tpm_register_image_test \
    --uefi-data $(cat your_binary_blob.bin) \
    --block-device-mappings "DeviceName=/dev/sda1,Ebs=
 {SnapshotId=snap-0123456789example,DeleteOnTermination=true}" \
    --architecture x86_64 \
    --root-device-name /dev/sda1 \
    --virtualization-type hvm \
    --ena-support \
    --boot-mode uefi
```

建立 AWS UEFI 安全開機的二進位 Blob

您可以使用以下步驟在 AMI 建立過程中自定義 UEFI 安全開機變數。在這些步驟中使用的 KEK 截至 2021 年 9 月都是最新的。如果 Microsoft 更新 KEK，則您必須使用最新的 KEK。

建立 AWS 二進位 Blob

1. 建立一個空白的 PK 簽章清單。

```
touch empty_key.crt
cert-to-efi-sig-list empty_key.crt PK.esl
```

2. 下載 KEK 憑證。

```
https://go.microsoft.com/fwlink/?LinkId=321185
```

3. 將 KEK 憑證包裝在 UEFI 簽章清單中 (`siglist`)。

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_KEK.esl MicCorKEKCA2011_2011-06-24.crt
```

4. 下載 Microsoft 的 db 憑證。

```
https://www.microsoft.com/pkiops/certs/MicWinProPCA2011_2011-10-19.crt
https://www.microsoft.com/pkiops/certs/MicCorUEFCA2011_2011-06-27.crt
```

5. 產生 db 簽章清單。

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_db.esl MicWinProPCA2011_2011-10-19.crt
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_UEFI_db.esl MicCorUEFCA2011_2011-06-27.crt
cat MS_Win_db.esl MS_UEFI_db.esl > MS_db.esl
```

6. Unified Extensible Firmware Interface 論壇不再提供 DBX 檔案。這些檔案目前由 Microsoft 在 GitHub 上提供。請前往 https://github.com/microsoft/secureboot_objects，透過 Microsoft 安全啟動更新儲存庫下載最新 DBX 更新。

7. 解壓縮簽署的 update-binary。

   透過下面的指令碼內容來建立 `SplitDbxContent.ps1`。或者，您可使用 `Install-Script -Name SplitDbxContent`，透過 PowerShell Gallery 來安裝指令碼。

```
<#PSScriptInfo

.VERSION 1.0

.GUID ec45a3fc-5e87-4d90-b55e-bdea083f732d

.AUTHOR Microsoft Secure Boot Team

.COMPANYNAME Microsoft

.COPYRIGHT Microsoft

.TAGS Windows Security
```

```
.LICENSEURI

.PROJECTURI

.ICONURI

.EXTERNALMODULEDEPENDENCIES

.REQUIREDSCRIPTS

.EXTERNALSCRIPTDEPENDENCIES

.RELEASENOTES
Version 1.0: Original published version.

#>

<#
.DESCRIPTION
 Splits a DBX update package into the new DBX variable contents and the signature
 authorizing the change.
 To apply an update using the output files of this script, try:
 Set-SecureBootUefi -Name dbx -ContentFilePath .\content.bin -SignedFilePath .
\signature.p7 -Time 2010-03-06T19:17:21Z -AppendWrite'
.EXAMPLE
.\SplitDbxAuthInfo.ps1 DbxUpdate_x64.bin
#>


# Get file from script input
$file  = Get-Content -Encoding Byte $args[0]

# Identify file signature
$chop = $file[40..($file.Length - 1)]
if (($chop[0] -ne 0x30) -or ($chop[1] -ne 0x82 )) {
    Write-Error "Cannot find signature"
    exit 1
}

# Signature is known to be ASN size plus header of 4 bytes
$sig_length = ($chop[2] * 256) + $chop[3] + 4
$sig = $chop[0..($sig_length - 1)]
```

```
if ($sig_length -gt ($file.Length + 40)) {
    Write-Error "Signature longer than file size!"
    exit 1
}

# Content is everything else
$content = $file[0..39] + $chop[$sig_length..($chop.Length - 1)]

# Write signature and content to files
Set-Content -Encoding Byte signature.p7 $sig
Set-Content -Encoding Byte content.bin $content
```

使用指令碼來解壓縮簽署的 DBX 檔案。

```
PS C:\Windows\system32> SplitDbxContent.ps1 .\dbx.bin
```

這會產生兩個檔案 - signature.p7 與 content.bin。在下一個步驟使用 content.bin。

8. 使用 uefivars.py 指令碼建置 UEFI 變數存放區。

```
./uefivars.py -i none -o aws -O uefiblob-microsoft-keys-empty-pk.bin -P ~/PK.esl -K
 ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx ~/content.bin
```

9. 檢查二進位 blob 和 UEFI 變數存放區。

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o json | less
```

10. 您可以透過再次將 blob 傳遞給同一工具來更新 Blob。

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o aws -O uefiblob-
microsoft-keys-empty-pk.bin -P ~/PK.esl -K ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx
 ~/content.bin
```

預期的輸出結果

```
Replacing PK
Replacing KEK
Replacing db
Replacing dbx
```

對 EBS 後端 AMI 使用加密

以 Amazon EBS 快照為後端的 AMI 可以利用 Amazon EBS 加密。資料和根磁碟區的快照可以經過加密再連接至 AMI。您可以啟動執行個體並複製映像，而且包括完整 EBS 加密支援在內。 AWS KMS 可使用 的所有區域都支援這些操作的加密參數。

具有 EBS 磁碟區的 EC2 執行個體從 AMIs 啟動的方式，如同其他執行個體一樣。此外，從以未加密 EBS 快照為後端的 AMI 啟動執行個體時，您可以在啟動期間加密部分或所有磁碟區。

如同 EBS 磁碟區，AMIs 中的快照可由您的預設金鑰 AWS KMS key或您指定的客戶受管金鑰加密。無論是哪種情況，您都必須擁有許可才能使用所選取的 KMS 金鑰。

具有加密快照的 AMIs 可以跨 AWS 帳戶共用。如需詳細資訊，請參閱了解 Amazon EC2 的共用 AMI 使用情況。

對 EBS 後端 AMI 加密主題

- 執行個體啟動案例
- 映像複製案例

執行個體啟動案例

透過 或直接使用 Amazon EC2 API AWS 管理主控台 或 CLI，透過區塊型設備映射提供的參數，使用 `RunInstances`動作從 AMIs 啟動 Amazon EC2 執行個體。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射。如需從 控制區塊型設備映射的範例 AWS CLI，請參閱啟動、列出和終止 EC2 執行個體。

根據預設，如果沒有明確的加密參數，從 AMI 的來源快照還原 EBS 磁碟區時，`RunInstances` 動作會維護快照的現有加密狀態。如果預設啟用加密，則從 AMI (無論是加密快照還是未加密快照) 建立的所有磁碟區都加密。如果預設加密未啟用，則執行個體會維護 AMI 的加密狀態。

您也可以啟動執行個體並提供加密參數，以同時將新的加密狀態套用到產生的磁碟區。因此，可看到有下列行為：

在無加密參數的情況下啟動

- 未加密快照會還原到未加密的磁碟區，除非預設加密已啟用，在此情況下，所有新建立的磁碟區都加密。

- 您擁有的加密快照會還原到以相同 KMS 金鑰 所加密的磁碟區。
- 您未擁有的加密快照 （例如，與您共用 AMI) 會還原至由您 AWS 帳戶預設 KMS 金鑰加密的磁碟區。

您可以提供加密參數來覆寫預設行為。可用的參數是 Encrypted 和 KmsKeyId。只設定 Encrypted 參數會產生下列行為：

已設定 **Encrypted** 但未指定 **KmsKeyId** 的執行個體啟動行為

- 未加密快照會還原到以您的 AWS 帳戶的預設 KMS 金鑰所加密的 EBS 磁碟區。
- 您擁有的加密快照會還原到以相同 KMS 金鑰 所加密的 EBS 磁碟區。(換言之，Encrypted 參數沒有效果。)
- 您未擁有的加密快照 （即與您共用的 AMI) 會還原至由您 AWS 帳戶預設 KMS 金鑰加密的磁碟區。(換言之，Encrypted 參數沒有效果。)

同時設定 Encrypted 和 KmsKeyId 參數可讓您為加密操作指定非預設 KMS 金鑰。產生下列行為：

Encrypted 和 **KmsKeyId** 都設定的執行個體

- 未加密快照會還原到以指定的 KMS 金鑰 所加密的 EBS 磁碟區。
- 加密快照會還原到不是以原始 KMS 金鑰 所加密的 EBS 磁碟區，而是以指定的 KMS 金鑰 所加密的 EBS 磁碟區。

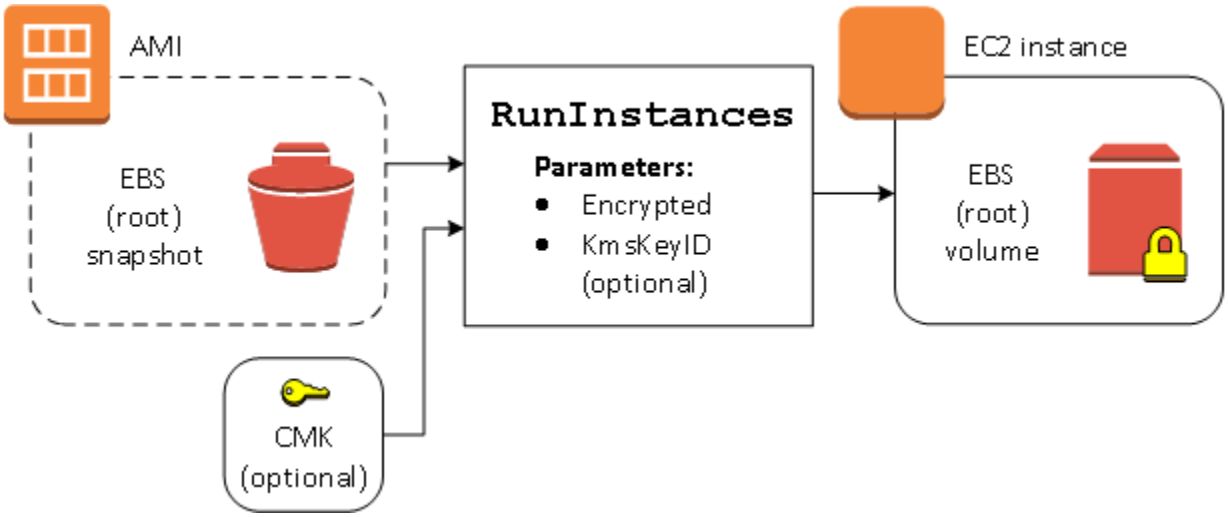
提交 KmsKeyId 但未同時設定 Encrypted 參數會導致錯誤。

下列各節提供從 AMI 使用非預設加密參數來啟動執行個體的範例。在每個案例中，提供給 RunInstances 動作的參數會導致從快照還原磁碟區期間變更加密狀態。

如需使用主控台從 AMI 啟動執行個體的資訊，請參閱 啟動 Amazon EC2 執行個體。

啟動期間加密磁碟區

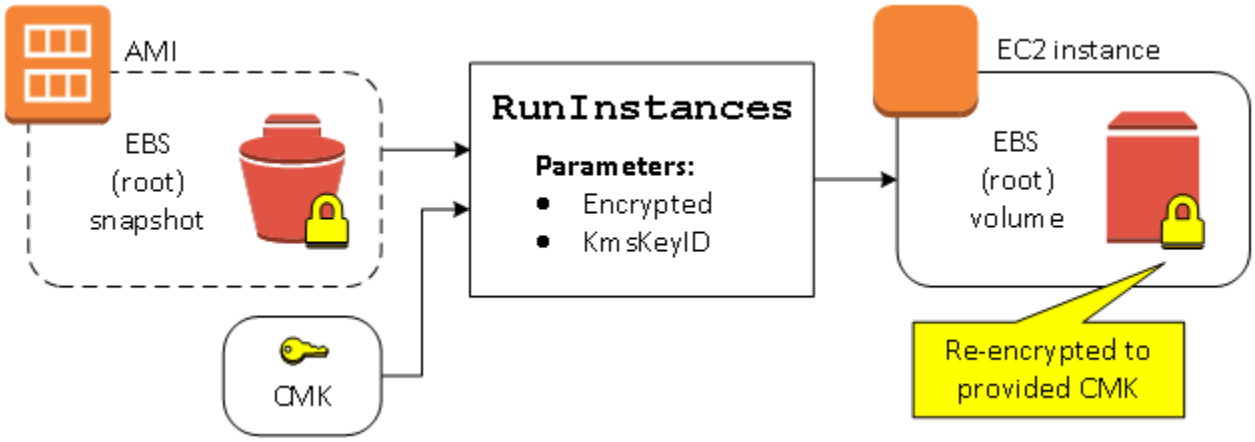
在此範例中，以未加密快照為後端的 AMI 用於啟動具有加密 EBS 磁碟區的 EC2 執行個體。



單獨 Encrypted 參數會導致加密此執行個體的磁碟區。提供 KmsKeyId 參數為選擇性。如果未指定 KMS 金鑰 ID，則會使用 AWS 帳戶的預設 KMS 金鑰來加密磁碟區。若要以您擁有的不同 KMS 金鑰來加密磁碟區，請提供 KmsKeyId 參數。

啟動期間重新加密磁碟區

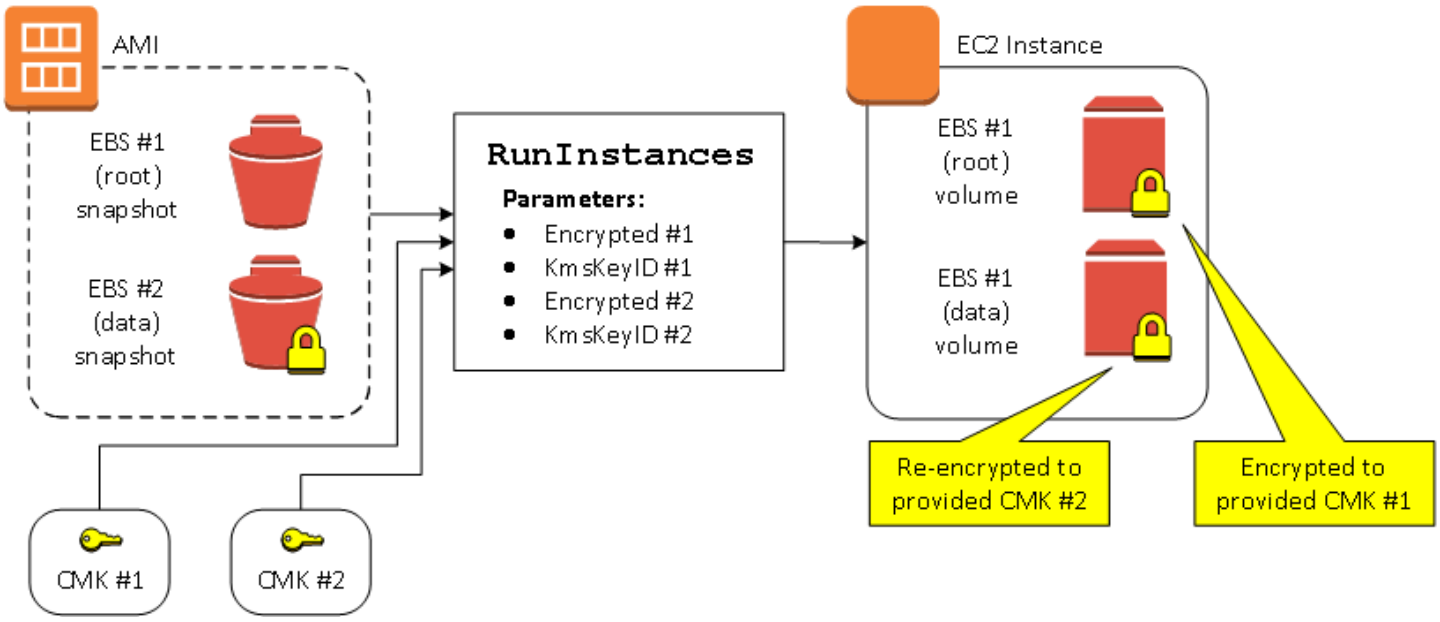
在此範例中，受加密快照支援的 AMI 用於啟動 EC2 執行個體，而此執行個體具有以新的 KMS 金鑰 所加密的 EBS 磁碟區。



如果您擁有 AMI 但未提供加密參數，則產生的執行個體具有使用與快照相同的 KMS 金鑰 所加密的磁碟區。如果 AMI 是共享，而非您所擁有，而且您未提供加密參數，則會以您的預設 KMS 金鑰 來加密磁碟區。按上圖所示提供加密參數，磁碟區會以指定的 KMS 金鑰 加密。

啟動期間變更多個磁碟區的加密狀態

在這個較複雜的範例中，以多個快照 (各有其自己的加密狀態) 為後端的 AMI 用於啟動 EC2 執行個體，而此執行個體具新加密的磁碟區和重新加密的磁碟區。



在此案例中，提供每個來源快照的加密參數給 RunInstances 動作。指定所有可能的加密參數時，無論您是否擁有 AMI，產生的執行個體都相同。

映像複製案例

透過 AWS 管理主控台 或直接使用 Amazon EC2 API 或 CLI，使用 CopyImage 動作來複製 Amazon EC2 AMI。

根據預設，如果沒有明確的加密參數，在複製期間，CopyImage 動作會維護 AMI 的來源快照的現有加密狀態。您也可以啟動 AMI 並提供加密參數，以同時將新的加密狀態套用到其相關聯的 EBS 快照。因此，可看到有下列行為：

在無加密參數的情況下複製

- 未加密快照會複製到另一個未加密的快照，除非預設加密已啟用，在此情況下，所有新建立的快照都加密。
- 您擁有的加密快照會複製到以相同 KMS 金鑰 加密的快照。
- 您未擁有的加密快照 （ 也就是與您共用的 AMI) 會複製到您 AWS 帳戶預設 KMS 金鑰所加密的快照。

您可以提供加密參數來覆寫所有這些預設行為。可用的參數是 Encrypted 和 KmsKeyId。只設定 Encrypted 參數會產生下列行為：

已設定 **Encrypted** 但未指定 **KmsKeyId** 的複製映像行為

- 未加密快照會複製到以 AWS 帳戶的預設 KMS 金鑰所加密的快照。
- 加密快照會複製到以相同 KMS 金鑰 所加密的快照。(換言之，Encrypted 參數沒有效果。)
- 您未擁有的加密快照 （ 也就是與您共用的 AMI) 會複製到您 AWS 帳戶預設 KMS 金鑰所加密的磁碟區。(換言之，Encrypted 參數沒有效果。)

同時設定 Encrypted 和 KmsKeyId 參數可讓您為加密操作指定客戶受管 KMS 金鑰。產生下列行為：

Encrypted 和 KmsKeyId 都設定的複製映像行為

- 未加密快照會複製到以指定的 KMS 金鑰 所加密的快照。
- 加密快照會複製到不是以原始 KMS 金鑰 所加密的快照，而是以指定的 KMS 金鑰 加密的快照。

提交 KmsKeyId 但未同時設定 Encrypted 參數會導致錯誤。

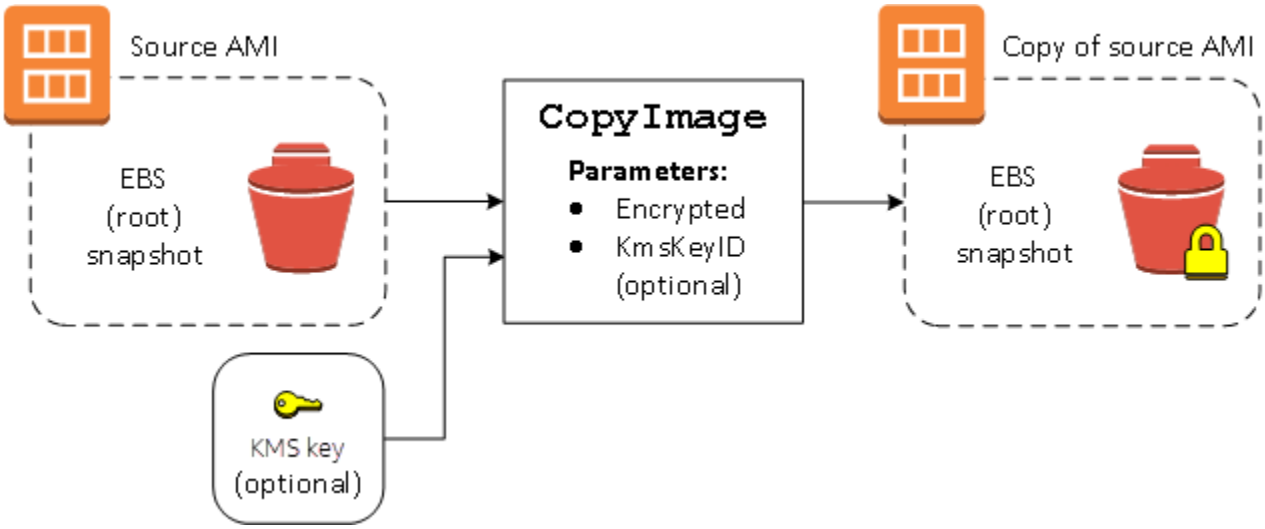
下一節提供使用非預設加密參數來複製 AMI 的範例，結果導致變更加密狀態。

如需使用主控台的詳細說明，請參閱 複製 Amazon EC2 AMI。

複製期間加密未加密的映像

在此案例中，未加密根快照所支援的 AMI 將複製到具有加密根快照的 AMI。透過兩個加密參數叫用 CopyImage 動作，包括客戶受管金鑰。因此，根快照的加密狀態會變更，目標 AMI 會受到包含與來源快照相同資料的根快照支援，但使用指定的金鑰進行加密。您需要為兩個 AMI 中的快照支付儲存成本，以及從任一 AMI 啟動的任何執行個體的費用。

> **Note**
>
> 啟用預設加密，這與在 AMI 中針對所有快照將 Encrypted 參數設定為 true 具有相同效果。



設定 Encrypted 參數會加密此執行個體的單一快照。若您未指定 KmsKeyId 參數，則會使用預設的客戶受管金鑰來加密快照複本。

> **Note**
>
> 您也可以複製具有多個快照的映像，並個別設定各快照的加密狀態。

了解 Amazon EC2 的共用 AMI 使用情況

共享 AMI 是由開發人員建置並可讓其他人員使用的 AMI。開始使用 Amazon EC2 最簡單的方式之一，就是使用具備您所需元件的共享 AMI，然後再新增自訂的內容。您也可以建立自己的 AMI 並與其他人共享。

但使用共享 AMI 所面臨的風險由您自行承擔。對於其他 Amazon EC2 使用者共享的 AMI，Amazon 不保證其完整性或安全性。因此，您對待共享 AMI 的方式應與您在考慮將任何外部程式碼部署到自己的資料中心時相同，且應善盡相關責任。建議您從信任的來源取得 AMI，例如經過驗證的供應商。

已驗證的供應商

在 Amazon EC2 主控台中，Amazon 或已驗證的 Amazon 合作夥伴擁有的公用 AMI 標記為 Verified provider (已驗證的供應商)。

您也可以使用 describe-images AWS CLI 命令來識別來自已驗證提供者的公AMIs。Amazon 或已驗證的合作夥伴擁有的公有映像有一個採用別名的擁有者，即 amazon、aws-backup-vault 或 aws-marketplace。在 CLI 輸出中，這些值顯示為 ImageOwnerAlias。其他使用者無法為其 AMI 設定別名。這可讓您輕鬆地尋找來自 Amazon 或已驗證的合作夥伴的 AMI。

若要成為已驗證的供應商，您必須在 AWS Marketplace上註冊為賣方。註冊後，您可以將您的 AMI 在 AWS Marketplace上列出。如需詳細資訊，請參閱「AWS Marketplace 賣方指南」中的賣方入門和以 AMI 為基礎的產品。

共享 AMI 主題

- 尋找用於 Amazon EC2 執行個體的共用 AMI
- 準備使用適用於 Linux 的共用 AMI
- 透過允許 AMI 在 Amazon EC2 控制 AMI 的探索及使用
- 在 Amazon EC2 中公開提供您的 AMI
- 了解封鎖 AMI 的公有存取權
- 與組織或組織單位共用 AMI
- 與特定 AWS 帳戶共用 AMI
- 取消將 AMI 與您的 共用 AWS 帳戶
- 建立共用 Linux AMI 的建議

如果您正在尋找有關其他主題的資訊

- 如需有關建立 AMI 的詳細資訊，請參閱 the section called “建立 Amazon S3 支援的 AMI” 或 the section called “建立 AMI”.。
- 如需有關在 AWS Marketplace上建置、交付和維護應用程式的資訊，請參閱 AWS Marketplace 文件。

尋找用於 Amazon EC2 執行個體的共用 AMI

您可使用 Amazon EC2 主控台或命令列來尋找要與 Amazon EC2 執行個體搭配使用的公用或私有共用 AMI。

AMI 是一種區域性的資源。因此，您必須在 AMI 進行共享的區域中搜尋，才能搜尋到共享 AMI (公有或私有)。若要讓 AMI 在其他區域可用，請將 AMI 複製到該區域，然後共享。如需詳細資訊，請參閱複製 Amazon EC2 AMI。

Console

主控台提供了一個 AMI 篩選欄位。您亦可用搜尋欄位中提供的篩選條件，來確定搜尋範圍。

如需尋找共享項或 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 在第一個篩選條件中，選擇下列其中一個選項：
   - 私有映像 – 列示與您共享的全部 AMI。
   - 公有映像 – 列示全部公有 AMI。
4. (選用) 如需僅顯示 Amazon 的公有映像，選擇搜尋欄位，然後從功能表選項中依序選擇擁有者別名、= 和 amazon。
5. (選用) 新增篩選條件，以便將搜尋範圍限定在符合您要求的 AMI 上。

如需透過經驗證的服務提供者來共享公有 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI Catalog (AMI 型錄)。
3. 選擇 Community AMI (社群 AMI)。
4. 在優化結果窗格中，選取經驗證的服務提供者。經驗證的服務提供者標籤指示來自 Amazon 或已驗證的合作夥伴的 AMI。

AWS CLI

請使用 describe-images 命令來列出 AMI。您可縮小範圍，僅列出感興趣的 AMI 類型，如下列範例所示。

如需列出全部公有 AMI

下列命令列出所有公用 AMI，包括您所擁有的任何公用 AMI。

```
aws ec2 describe-images --executable-users all
```

如需列出具有明確啟動許可的 AMI

下列命令列出您擁有明確啟動許可的 AMI。此清單不含您所擁有的任何 AMI。

```
aws ec2 describe-images --executable-users self
```

如需列出經驗證的服務提供者擁有的 AMI

下列命令會列示經驗證的服務提供者擁有的 AMI。已驗證的供應商 (Amazon 或已驗證的合作夥伴) 擁有的公用 AMI 有一個採用別名的擁有者，其在帳戶欄位顯示為 amazon、aws-backup-vault 或 aws-marketplace。這可協助您輕鬆尋找來自已驗證的供應商的 AMI。其他使用者無法為其 AMI 設定別名。

```
aws ec2 describe-images \
    --owners amazon aws-marketplace \
    --query 'Images[*].[ImageId]' \
    --output text
```

如需列出某一帳戶擁有的 AMI

下列命令會列出特定 AWS 帳戶擁有的 AMI。

```
aws ec2 describe-images --owners 123456789012
```

如需使用篩選條件確定 AMI 範圍

若要減少顯示的 AMI 數量，請使用篩選條件，只列出感興趣的 AMI 類型。例如，下列篩選條件只會顯示 EBS 後端 AMI。

```
--filters "Name=root-device-type,Values=ebs"
```

PowerShell

請使用 Get-EC2Image cmdlet (Tools for Windows PowerShell) 來列出 AMI。您可縮小範圍，僅列出感興趣的 AMI 類型，如下列範例所示。

如需列出全部公有 AMI

下列命令列出所有公用 AMI，包括您所擁有的任何公用 AMI。

```
Get-EC2Image -ExecutableUser all
```

如需列出具有明確啟動許可的 AMI

下列命令列出您擁有明確啟動許可的 AMI。此清單不含您所擁有的任何 AMI。

```
Get-EC2Image -ExecutableUser self
```

如需列出經驗證的服務提供者擁有的 AMI

下列命令會列示經驗證的服務提供者擁有的 AMI。已驗證的供應商 (Amazon 或已驗證的合作夥伴) 擁有的公用 AMI 有一個採用別名的擁有者，其在帳戶欄位顯示為 amazon、aws-backup-vault 或 aws-marketplace。這可協助您輕鬆尋找來自已驗證的供應商的 AMI。其他使用者無法為其 AMI 設定別名。

```
Get-EC2Image -Owner amazon aws-marketplace
```

如需列出某一帳戶擁有的 AMI

下列命令會列出特定 AWS 帳戶擁有的 AMI。

```
Get-EC2Image -Owner 123456789012
```

如需使用篩選條件確定 AMI 範圍

若要減少顯示的 AMI 數量，請使用篩選條件，只列出感興趣的 AMI 類型。例如，下列篩選條件只會顯示 EBS 後端 AMI。

```
-Filter @{Name="root-device-type"; Values="ebs"}
```

準備使用適用於 Linux 的共用 AMI

在使用適用於 Linux 的共用 AMI 之前，請先採取下列步驟，確定沒有預先安裝的登入資料可能允許第三方對您的執行個體執行不必要的存取，並確定沒有預先設定的遠端登入可能將機密資料傳送至第三方。請參閱 AMI 所使用 Linux 發行版本的說明文件，以取得改善系統安全性的詳細資訊。

為確保您不會意外失去執行個體的存取權，建立您初始化兩個 SSH 工作階段，並保持第二個工作階段為開啟狀態，直到您移除無法識別的登入資料並確認您仍可用 SSH 登入您的執行個體為止。

1. 識別並停用任何未經授權的公有 SSH 金鑰。檔案中唯一能存在的金鑰應該是您用來啟動 AMI 的金鑰。以下命令可找出 authorized_keys 檔案：

   ```
   [ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
   ```

2. 停用根使用者的密碼型態身分驗證。開啟 sshd_config 檔案並編輯 PermitRootLogin 行，如下所示：

   ```
   PermitRootLogin without-password
   ```

或者，您可停用公用程式，以根使用者身分登入執行個體：

```
PermitRootLogin No
```

重新啟動 sshd 服務。

3. 檢查是否有任何其他使用者能夠登入您的執行個體。具有超級使用者權限的使用者特別危險。移除或鎖定任何未知帳戶的密碼。
4. 檢查未使用且正在執行網路服務偵聽是否有傳入連線的開放中連接埠。
5. 若要避免預先設定的遠端登入，您應刪除現有的組態檔案，然後重新啟動 `rsyslog` 服務。例如：

```
[ec2-user ~]$ sudo rm /etc/rsyslog.conf
[ec2-user ~]$ sudo service rsyslog restart
```

6. 確認所有的 cron 任務均為合法。

如果發現可能存有安全風險的公用 AMI，請聯絡 AWS 安全團隊。如需詳細資訊，請參閱 AWS 安全中心。

透過允許 AMI 在 Amazon EC2 控制 AMI 的探索及使用

若要控制使用者在 AWS 帳戶對於 Amazon Machine Images (AMI) 的探索及使用，您可以使用允許 AMI 功能。您可指定 AMI 必須符合的條件，才能在您的帳戶中可見且可供使用。在啟用條件之後，啟動執行個體的使用者只會看到並存取符合指定條件的 AMI。例如，您可指定受信任的 AMI 供應商清單作為條件，只有來自這些供應商的 AMI 才可見且可供使用。

在啟用允許 AMI 設定之前，您可啟用稽核模式，以預覽 AMI 是否可見且可供使用。這可讓您根據需要調整條件，確保只有預期的 AMI 可見且可供您帳戶的使用者使用。此外，使用 describe-instance-image-metadata 命令，來尋找使用不符合指定條件的 AMI 啟動的執行個體。此資訊可指導您決定是否更新啟動組態以使用符合規範的 AMI (例如，在啟動範本指定不同 AMI)，或調整條件以允許這些 AMI。

您可直接在帳戶或使用宣告式政策，在帳戶層級指定允許 AMI 設定。這些設定必須在您想要控制 AMI 用量的每個 AWS 區域 位置設定。您可使用宣告式政策同時在多個區域及多個帳戶套用設定。使用宣告式政策時，您無法直接在帳戶中修改設定。本主題說明如何直接在帳戶內配置設定。如需使用宣告式政策的相關資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

> **Note**
>
> 允許 AMI 功能僅可針對公共 AMI 或與您的帳戶共享的 AMI 控制其探索及使用。其不限制您帳戶擁有的 AMI。無論您設定什麼條件，您帳戶建立的 AMI 一律可供您帳戶的使用者探索及使用。

允許 AMI 的主要優點

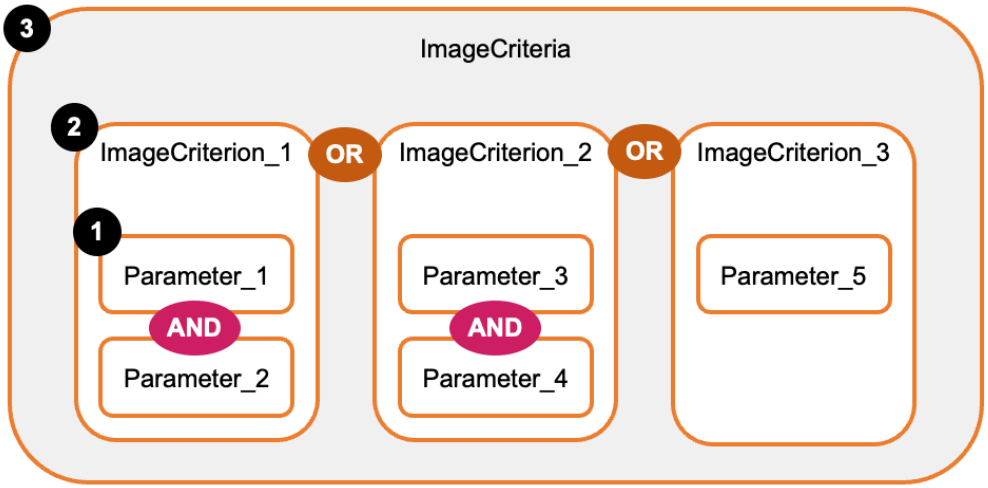
- 合規性和安全性：使用者只能探索及使用符合指定條件的 AMI，進而降低以不合規方式使用 AMI 的風險。
- 高效管理：透過減少允許 AMI 數量可讓管理剩餘的 AMI 變得更簡單、更有效率。
- 集中帳戶層級實作：直接在帳戶內或使用宣告式政策，在帳戶層級配置允許 AMI 設定。這可提供集中且有效的方式來控制整個帳戶的 AMI 使用情況。

目錄

- 允許 AMI 運作方式
- 實作允許 AMI 的最佳實務
- 所需的 IAM 許可
- 管理允許的 AMI 的設定

允許 AMI 運作方式

如需控制在帳戶中可探索及使用哪些 AMI，您可定義評估 AMI 的條件集。這些條件由如下圖所示的一個或多個 `ImageCriterion` 組成。說明性文字遵循圖表。



組態具有三個層級：

- 1 – 參數值
  - 多值參數：
    - `ImageProviders`
    - `ImageNames`
    - `MarketplaceProductCodes`

      AMI 可比對參數內允許的任何值。

      範例：`ImageProviders` = amazon OR 帳戶 111122223333 OR 帳戶 444455556666 (圖表中並未顯示參數值的評估邏輯。)
    - `ImageWatermarks`
  - 單一值參數：
    - `CreationDateCondition`
    - `DeprecationTimeCondition`
- 2 – `ImageCriterion`
  - 透過 AND 邏輯來分組多個參數。
  - AMI 必須比對 `ImageCriterion` 允許的所有參數。
  - 範例：`ImageProviders` = amazon AND `CreationDateCondition` = 300 天或以下
- 3 – `ImageCriteria`
  - 使用 OR 邏輯來分組多個 `ImageCriterion`。
  - AMI 可比對允許的任何 `ImageCriterion`。

- 構成用於評估 AMI 的完整組態。

主題

- 允許的 AMI 參數
- 允許的 AMI 組態
- 條件評估方式
- 限制
- 允許 AMI 操作

允許的 AMI 參數

還可設定以下參數來建立 `ImageCriterion`：

`ImageProviders`

允許其 AMI 的 AMI 服務提供者。

有效值是由 AWS和 AWS 帳戶 IDs 定義的別名，如下所示：

- `amazon` - 此別名可識別由 Amazon 的經驗證的服務提供者建立的 AMI
- `aws-marketplace` – 識別 中已驗證提供者所建立 AMIs 的別名 AWS Marketplace
- `aws-backup-vault` - 此別名可識別位於邏輯氣隙 AWS 備份保存庫帳戶的備份 AMI。如果您使用 AWS Backup 邏輯氣隙隔離保存庫功能，請確定此別名包含為 AMI 供應商。
- AWS 帳戶 IDs – 一或多個 12 位數 AWS 帳戶 IDs
- `none` - 表示只能探索及使用由您帳戶建立的 AMI。無法探索及使用公用或共用 AMI。指定後不可指定其他條件。

`ImageNames`

允許的 AMI 名稱，使用完全相符或萬用字元 (? 或 *)。

`MarketplaceProductCodes`

允許 AMIs AWS Marketplace 的產品代碼。

`CreationDateCondition`

允許的 AMI 的最長存留期。

`DeprecationTimeCondition`

自允許的 AMI 棄用以來的最長期間。

`ImageWatermarks`

浮水印篩選條件清單。如果 AMI 上的任何浮水印符合任何篩選條件項目 (OR 邏輯），則篩選條件允許 AMI。每個篩選條件可以包含下列欄位：

- `WatermarkKey` – 要比對的浮水印索引鍵 (`account-id:watermark-name`)
- `SourceImageRegion` – 您最初連接浮水印的 AMI 區域
- `MaximumDaysSinceSourceImageCreated` – 套用浮水印時來源 AMI 的最長存留期
- `MaximumDaysSinceWatermarkCreated` – 自您套用浮水印以來的天數上限

若要了解每個條件的有效值與約束條件，請參閱 Amazon EC2 API 參考中的 ImageCriterionRequest。

允許的 AMI 組態

允許的 AMI 核心組態為 `ImageCriteria` 組態，其定義允許 AMI 的條件。下面的 JSON 結構顯示可指定的參數：

```
{
    "State": "enabled" | "disabled" | "audit-mode",
    "ImageCriteria" : [
        {
            "ImageProviders": ["string",...],
            "MarketplaceProductCodes": ["string",...],
            "ImageNames":["string",...],
            "CreationDateCondition" : {
                "MaximumDaysSinceCreated": integer
            },
            "DeprecationTimeCondition" : {
                "MaximumDaysSinceDeprecated": integer
            },
            "ImageWatermarks": [
                {
                    "WatermarkKey": "string",
                    "SourceImageRegion": "string",
                    "MaximumDaysSinceSourceImageCreated": integer,
                    "MaximumDaysSinceWatermarkCreated": integer
                },...
```

```
            ]
         },
         ...
}
```

ImageCriteria 範例

下列ImageCriteria範例會設定五個 ImageCriterion。若 AMI 與這些 ImageCriterion 中的任何一個相符，則允許該 AMI。若要了解有關如何評估條件的詳細資訊，請參閱 條件評估方式。

```
{
    "ImageCriteria": [
        // ImageCriterion 1: Allow AWS Marketplace AMIs with product code
 "abcdefg1234567890"
        {
            "MarketplaceProductCodes": [
                "abcdefg1234567890"
            ]
        },
        // ImageCriterion 2: Allow AMIs from providers whose accounts are
        // "123456789012" OR "123456789013" AND AMI age is less than 300 days
        {
            "ImageProviders": [
                "123456789012",
                "123456789013"
            ],
            "CreationDateCondition": {
                "MaximumDaysSinceCreated": 300
            }
        },
        // ImageCriterion 3: Allow AMIs from provider whose account is "123456789014"
        // AND with names following the pattern "golden-ami-*"
        {
            "ImageProviders": [
                "123456789014"
            ],
            "ImageNames": [
                "golden-ami-*"
            ]
        },
        // ImageCriterion 4: Allow AMIs from Amazon or verified providers
        // AND which aren't deprecated
        {
```

```
            "ImageProviders": [
                "amazon"
            ],
            "DeprecationTimeCondition": {
                "MaximumDaysSinceDeprecated": 0
            }
        },
        // ImageCriterion 5: Allow AMIs that carry an approved watermark
        // from account "111122223333" AND the watermark was applied less than 90 days
 ago
        {
            "ImageWatermarks": [
                {
                    "WatermarkKey": "111122223333:prod-baseline",
                    "MaximumDaysSinceWatermarkCreated": 90
                }
            ]
        }
    ]
}
```

條件評估方式

下表說明了確定是否允許 AMI 的評估規則，並顯示了 AND 或 OR 運算子如何在每個層級套用：

評估層級	運算子	需要是允許的 AMI
ImageProviders 、ImageNames 、MarketplaceProductCodes 和 的參數值 ImageWatermarks	OR	AMI 必須與每個參數清單中至少一個值相符
ImageCriterion	AND	AMI 必須與每個 ImageCriterion 的全部參數相符
ImageCriteria	OR	AMI 必須與 ImageCriterion 任意一個參數相符

我們使用上述評估規則，來了解如何套用規則至 ImageCriteria 範例：

- ImageCriterion 1：允許具有 AWS Marketplace 產品代碼的 AMIs abcdefg1234567890

  OR

- ImageCriterion 2：允許滿足以下兩項條件的 AMI：
  - 由任一帳戶 123456789012 OR 123456789013 擁有
    - AND
  - 在過去 300 天內建立

  OR

- ImageCriterion 3：允許滿足以下兩項條件的 AMI：
  - 由帳戶 123456789014 擁有
    - AND
  - 以模式 golden-ami-* 命名

  OR

- ImageCriterion 4：允許滿足以下兩項條件的 AMI：
  - 由 Amazon 或經驗證的服務提供者發布 (由 amazon 別名指定)
    - AND
  - 未棄用 (自棄用以來的最長天數為 0)

  OR

- ImageCriterion 5：允許帶有浮水印比對的 AMIs：
  - Watermark 金鑰 111122223333:prod-baseline
    - AND
  - 您在 90 天內套用浮水印

限制

ImageCriteria 可包括最多：

- 10 ImageCriterion

每個 ImageCriterion 可包括最多：

- ImageProviders 的 200 個值

- ImageNames 的 50 個值
- MarketplaceProductCodes 的 50 個值
- ImageWatermarks 的 50 個值

限制範例

使用上述 ImageCriteria 範例：

- 有 5 個 ImageCriterion。最多可以再新增 5 個請求，以達到 10 的限制。
- 在第一個 ImageCriterion 中， 有 1 個 MarketplaceProductCodes 的值。最多可新增 49 個至此 ImageCriterion，達到上限 50 個。
- 在第二個 ImageCriterion 中， 有 2 個 ImageProviders 的值。最多可新增 198 個至此 ImageCriterion，達到上限 200 個。
- 在第三個 ImageCriterion 中， 有 1 個 ImageNames 的值。最多可新增 49 個至此 ImageCriterion，達到上限 50 個。

允許 AMI 操作

允許的 AMI 功能有三種操作狀態來管理映像條件：已啟用、已停用和稽核模式。這些模式可讓您啟用或停用映像條件，或根據需要來檢閱這些條件。

已啟用

當啟用允許 AMI 時：

- 會套用 ImageCriteria。
- 僅可在 EC2 主控台以及透過使用映像的 API (例如，描述、複製、儲存或執行使用映像的其他動作) 來探索允許 AMI。
- 只能使用允許 AMI 啟動執行個體。

Disabled

當停用允許 AMI 時：

- 不會套用 ImageCriteria。
- AMI 的可探索性或使用方式不受任何限制。

稽核模式

在稽核模式中：

- 會套用 ImageCriteria，但 AMI 的可探索性或使用方式不受任何限制。
- 在 EC2 主控台中，對於每個 AMI，允許映像欄位會顯示是或否，以指示啟用允許 AMI 時，帳戶使用者可否探索及使用該 AMI。
- 在命令列，describe-image 操作的回應包括 "ImageAllowed": true 或 "ImageAllowed": false，以指示當允許 AMI 啟用時，帳戶使用者可否探索及使用 AMI。
- 在 EC2 主控台，當啟用允許 AMI 時，AMI 目錄會在帳戶使用者無法探索或使用的 AMI 旁邊顯示不允許。

實作允許 AMI 的最佳實務

實作允許 AMIs 時，請考慮這些最佳實務，以確保順利轉換，並將對您 AWS 環境的潛在干擾降至最低。

1. 啟用稽核模式

   從在稽核模式啟用允許 AMI 開始。此狀態可讓您無需實際限制存取，即可查看哪些 AMI 會受到您的條件影響，提供無風險的評估期。

2. 設定允許 AMI 條件

   仔細確定哪些 AMI 供應商符合您組織的安全性策略、合規性要求和營運需求。

   > **Note**
   >
   > 使用 AWS 受管服務時，例如 Amazon ECS、Amazon EKS 或 AWS Lambda 受管執行個體，我們建議指定amazon別名以允許 建立AMIs AWS。這些服務取決於 Amazon 發布的 AMI 來啟動執行個體。
   >
   > 設定任何 AMI CreationDateCondition 約束條件時要謹慎。設定過度限制的日期條件（例如，AMIs 必須少於 5 天）可能會導致執行個體啟動失敗，如果 AMIs 無論是來自 AWS 或其他供應商，都未在指定的時間範圍內更新。
   >
   > 建議將 ImageNames 與 ImageProviders 配對，以便實現更好的控制與特徵。單獨使用 ImageNames 可能無法唯一識別 AMI。

3. 檢查對預期業務流程的影響

您可使用主控台或者 CLI，來確定透過不滿足指定條件的 AMI 來啟動的任何執行個體。此資訊可指導您決定是否更新啟動組態以使用符合規範的 AMI (例如，在啟動範本指定不同 AMI)，或調整條件以允許這些 AMI。

主控台：使用 ec2-instance-launched-with-allowed-ami AWS Config 規則，檢查執行中或已停止的執行個體是否使用AMIs 啟動。 AMIs 若 AMI 不滿足允許的 AMI 條件，則規則為 NON_COMPLIANT；若滿足條件，則規則為 COMPLIANT。僅當允許的 AMI 設定被設為已啟用或稽核模式時，規則才能運作。

CLI：執行 describe-instance-image-metadata 命令並篩選回應，確定使用不符合指定條件的 AMI 啟動的任何執行個體。

若要了解主控台與 CLI 的說明，請參閱 尋找透過不允許的 AMI 啟動的執行個體。

4. 啟用允許 AMI

   在您確認條件不會對預期的業務流程造成負面影響之後，請啟用允許 AMIs。

5. 監控執行個體啟動

   繼續監控您應用程式和所用 AWS 受管服務的 AMIs 執行個體啟動，例如 Amazon EMR、Amazon ECR、Amazon EKS 和 AWS Elastic Beanstalk。檢查是否有任何意外問題，並對允許 AMI 條件進行必要調整。

6. 試驗新的 AMI

   若要測試不符合目前允許 AMIs 設定的第三方 AMIs， AWS 建議下列方法：

   - 使用單獨的 AWS 帳戶：建立無法存取您關鍵業務資源的帳戶。確認在此帳戶中，未啟用允許的 AMI 設定；或者明確允許您想要測試的 AMI，以便測試這些 AMI。
   - 在另一個測試 AWS 區域：使用可使用第三方 AMIs的區域，但尚未啟用允許 AMIs設定的區域。

   在測試新的 AMI 時，這些方法可協助確保您的業務關鍵性資源保持安全。

所需的 IAM 許可

若要使用允許 AMI 功能，您需要下列 IAM 許可：

- `GetAllowedImagesSettings`
- `EnableAllowedImagesSettings`
- `DisableAllowedImagesSettings`

- `ReplaceImageCriteriaInAllowedImagesSettings`

管理允許的 AMI 的設定

您可管理允許的 AMI 的設定。這些設定依據每個帳戶每個區域設定。

任務

- 啟用允許 AMI
- 設定允許的 AMI 條件
- 停用允許 AMI
- 獲取允許的 AMI 條件
- 尋找允許的 AMI
- 尋找透過不允許的 AMI 啟動的執行個體

啟用允許 AMI

您可啟用允許的 AMI 並指定允許的 AMI 條件 建議以稽核模式開始，該模式可顯示哪些 AMI 會受條件影響，而不會實際限制存取權。

Console

若要啟用允許 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡的設定下，選擇允許的 AMIs。
4. 在允許的 AMI 索引標籤上，選擇管理。
5. 針對允許的 AMI 設定，選擇稽核模式或已啟用。建議以稽核模式開始，測試條件，然後返回此步驟，以便啟用允許的 AMI。
6. (選用) 針對 AMI 條件，輸入 JSON 格式的條件。
7. 選擇更新。

AWS CLI

若要啟用允許 AMI

請使用 enable-allowed-images-settings 命令。

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state enabled
```

如需轉為啟用稽核模式，指定 audit-mode 而不是 enabled。

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state audit-mode
```

PowerShell

若要啟用允許 AMI

使用 Enable-EC2AllowedImagesSetting cmdlet。

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState enabled
```

如需轉為啟用稽核模式，指定 audit-mode 而不是 enabled。

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState audit-mode
```

設定允許的 AMI 條件

在啟用允許的 AMI 之後，您可設定或取代允許的 AMI 條件。

如需了解正確的組態與有效值，請參閱 允許的 AMI 組態 和 允許的 AMI 參數。

Console

如需設定允許的 AMI 條件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡的設定下，選擇允許的 AMIs。
4. 在允許的 AMI 索引標籤上，選擇管理。
5. 針對 AMI 條件，輸入 JSON 格式的條件。
6. 選擇更新。

AWS CLI

如需設定允許的 AMI 條件

請使用 replace-image-criteria-in-allowed-images-settings 命令，然後指定包含允許的 AMI 條件的 JSON 檔案。

```
aws ec2 replace-image-criteria-in-allowed-images-settings --cli-input-json
 file://file_name.json
```

PowerShell

如需設定允許的 AMI 條件

請使用 Set-EC2ImageCriteriaInAllowedImagesSetting cmdlet，然後指定包含允許的 AMI 條件的 JSON 檔案。

```
$imageCriteria = Get-Content -Path .\file_name.json | ConvertFrom-Json
Set-EC2ImageCriteriaInAllowedImagesSetting -ImageCriterion $imageCriteria
```

停用允許 AMI

您可如下所示停用允許的 AMI。

Console

若要停用允許 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡的設定下，選擇允許的 AMIs。
4. 在允許的 AMI 索引標籤上，選擇管理。
5. 對於允許 AMI 設定，選擇已停用。
6. 選擇更新。

AWS CLI

若要停用允許 AMI

請使用 disable-allowed-images-settings 命令。

```
aws ec2 disable-allowed-images-settings
```

PowerShell

若要停用允許 AMI

使用 Disable-EC2AllowedImagesSetting cmdlet。

```
Disable-EC2AllowedImagesSetting
```

獲取允許的 AMI 條件

您可獲取允許的 AMI 設定的目前狀態及允許的 AMI 條件。

Console

如需獲取允許的 AMI 狀態及條件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡的設定下，選擇允許的 AMIs。
4. 在允許的 AMI 索引標籤上，將允許的 AMI 設定設定為已啟用、已停用或稽核模式。
5. 若允許的 AMI 狀態為已啟用或稽核模式、AMI 條件，會顯示 JSON 格式的 AMI 條件。

AWS CLI

如需獲取允許的 AMI 狀態及條件

請使用 get-allowed-images-settings 命令。

```
aws ec2 get-allowed-images-settings
```

在下面的範例輸出中，狀態為 `audit-mode`，且映像條件已在帳戶中設定。

```
{
    "State": "audit-mode",
```

```
    "ImageCriteria": [
        {
            "MarketplaceProductCodes": [
                "abcdefg1234567890"
            ]
        },
        {
            "ImageProviders": [
                "123456789012",
                "123456789013"
            ],
            "CreationDateCondition": {
                "MaximumDaysSinceCreated": 300
            }
        },
        {
            "ImageProviders": [
                "123456789014"
            ],
            "ImageNames": [
                "golden-ami-*"
            ]
        },
        {
            "ImageProviders": [
                "amazon"
            ],
            "DeprecationTimeCondition": {
                "MaximumDaysSinceDeprecated": 0
            }
        }
    ],
    "ManagedBy": "account"
}
```

PowerShell

如需獲取允許的 AMI 狀態及條件

使用 Get-EC2AllowedImagesSetting cmdlet。

```
Get-EC2AllowedImagesSetting | Select-Object `
    State, `
    ManagedBy, `
```

```
    @{Name='ImageProviders'; Expression={($_.ImageCriteria.ImageProviders)}}, `
    @{Name='MarketplaceProductCodes';
Expression={($_.ImageCriteria.MarketplaceProductCodes)}}, `
    @{Name='ImageNames'; Expression={($_.ImageCriteria.ImageNames)}}, `
    @{Name='MaximumDaysSinceCreated';
Expression={($_.ImageCriteria.CreationDateCondition.MaximumDaysSinceCreated)}}, `
    @{Name='MaximumDaysSinceDeprecated';
Expression={($_.ImageCriteria.DeprecationTimeCondition.MaximumDaysSinceDeprecated)}}
```

在下面的範例輸出中，狀態為 audit-mode，且映像條件已在帳戶中設定。

```
State      : audit-mode
ManagedBy  : account
ImageProviders            : {123456789012, 123456789013, 123456789014, amazon}
MarketplaceProductCodes   : {abcdefg1234567890}
ImageNames                : {golden-ami-*}
MaximumDaysSinceCreated   : 300
MaximumDaysSinceDeprecated: 0
```

尋找允許的 AMI

您可尋找依據目前允許的 AMI 條件而被允許或不被允許的 AMI。

> **Note**
>
> 允許的 AMI 必須為稽核模式。

Console

如需檢查 AMI 是否滿足允許的 AMI 條件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資料索引標籤 (如選取核取方塊) 或在摘要區域 (如選取 AMI ID)，找到允許映像欄位。
   - 是 – AMI 滿足允許的 AMI 條件。在啟用允許的 AMI 之後，您帳戶中的使用者將可使用此 AMI。

- 否 – AMI 不滿足允許的 AMI 條件。

5. 在導覽窗格中，選擇 AMI Catalog (AMI 型錄)。

   當 AMI 標記為不允許時，表示其不符合允許 AMI 條件。當啟用允許 AMI 時，您帳戶的使用者將無法看到或使用此 AMI。

AWS CLI

如需檢查 AMI 是否滿足允許的 AMI 條件

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].ImageAllowed \
    --output text
```

以下為範例輸出。

```
True
```

如需尋找滿足允許的 AMI 條件的 AMI

使用 describe-images 命令。

```
aws ec2 describe-images \
    --filters "Name=image-allowed,Values=true" \
    --max-items 10 \
    --query Images[].ImageId
```

以下為範例輸出。

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
```

```
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

PowerShell

如需檢查 AMI 是否滿足允許的 AMI 條件

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ImageAllowed
```

以下為範例輸出。

```
True
```

如需尋找滿足允許的 AMI 條件的 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -Filter @{Name="image-allows";Values="true"} `
    -MaxResult 10 | `
    Select ImageId
```

以下為範例輸出。

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

尋找透過不允許的 AMI 啟動的執行個體

您可確定使用不滿足允許的 AMI 條件的 AMI 啟動的執行個體。

Console

如需檢查識別執行個體是否使用不允許的 AMI 啟動

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 在詳細資訊索引標籤的執行個體詳細資訊下，找到允許的映像。
   - 是 – AMI 滿足允許的 AMI 條件。
   - 否 – AMI 不滿足允許的 AMI 條件。

AWS CLI

如需尋找使用不允許的 AMI 啟動的執行個體

請使用 `image-allowed` 篩選條件並搭配 describe-instance-image-metadata 命令。

```
aws ec2 describe-instance-image-metadata \
    --filters "Name=image-allowed,Values=false" \
    --query "InstanceImageMetadata[*].[InstanceId,ImageMetadata.ImageId]" \
    --output table
```

以下為範例輸出。

```
-----------------------------------------------------
|          DescribeInstanceImageMetadata          |
+----------------------+--------------------------+
|  i-08fd74f3f1595fdbd |  ami-09245d5773578a1d6   |
|  i-0b1bf24fd4f297ab9 |  ami-07cccf2bd80ed467f   |
|  i-026a2eb590b4f7234 |  ami-0c0ec0a3a3a4c34c0   |
|  i-006a6a4e8870c828f |  ami-0a70b9d193ae8a799   |
|  i-0781e91cfeca3179d |  ami-00c257e12d6828491   |
|  i-02b631e2a6ae7c2d9 |  ami-0bfddf4206f1fa7b9   |
+----------------------+--------------------------+
```

PowerShell

如需尋找使用不允許的 AMI 啟動的執行個體

使用 Get-EC2InstanceImageMetadata cmdlet。

```
Get-EC2InstanceImageMetadata `
    -Filter @{Name="image-allowed";Values="false"} | `
    Select InstanceId, @{Name='ImageId'; Expression={($_.ImageMetadata.ImageId)}}
```

以下為範例輸出。

```
InstanceId          ImageId
----------          -------
i-08fd74f3f1595fdbd ami-09245d5773578a1d6
i-0b1bf24fd4f297ab9 ami-07cccf2bd80ed467f
i-026a2eb590b4f7234 ami-0c0ec0a3a3a4c34c0
i-006a6a4e8870c828f ami-0a70b9d193ae8a799
i-0781e91cfeca3179d ami-00c257e12d6828491
i-02b631e2a6ae7c2d9 ami-0bfddf4206f1fa7b9
```

AWS Config

您可以新增 ec2-instance-launched-with-allowed-ami AWS Config 規則、根據您的需求進行設定，然後使用它來評估您的執行個體。

如需詳細資訊，請參閱《 AWS Config 開發人員指南》中的新增 AWS Config 規則和 ec2-instance-launched-with-allowed-ami。

在 Amazon EC2 中公開提供您的 AMI

您可以與所有 共用 AMI，使其可公開使用 AWS 帳戶。

如果您希望阻止公開共用您的 AMI，則可以啟用封鎖 AMI 的公有存取權。這樣可以阻止任何公開 AMI 的嘗試，有助於防止未經授權的存取和 AMI 資料的潛在濫用。請注意，啟用封鎖公開存取並不會影響您已公開提供的 AMI；它們仍然可以公開使用。如需詳細資訊，請參閱了解封鎖 AMI 的公有存取權。

若要只允許特定帳戶使用 AMI 來啟動執行個體，請參閱 與特定 AWS 帳戶共用 AMI。

目錄

- 考量事項
- 與所有 AWS 帳戶共用 AMI （公開共用）

考量事項

在將 AMI 設定為公有之前，請考慮以下項目。

- 擁有權 – 若要公開 AMI，您的 AWS 帳戶 必須擁有 AMI。
- 區域 - AMI 是一種區域性的資源。共享 AMI 後，只能從共享 AMI 的區域中使用 AMI。若要讓 AMI 在其他區域可用，請將 AMI 複製到該區域，然後共用。如需詳細資訊，請參閱複製 Amazon EC2 AMI。
- 封鎖公開存取 – 若要公開共用 AMI，必須在將公開共用 AMI 的每個區域中停用封鎖 AMI 的公有存取權。公開共用 AMI 後，您可以重新啟用「封鎖 AMI 的公有存取權」，以防止進一步公開共用您的 AMI。
- 部分 AMI 無法設定為公有 - 如果您的 AMI 有下列任一情況，則無法將其設定為公有 (但您可以與指定的 AWS 帳戶共用 AMI)：
  - 加密磁碟區
  - 加密磁碟區的快照
  - 產品代碼
- 避免敏感資料外洩 - 若要避免共享 AMI 時洩漏敏感資料，請詳讀 建立共用 Linux AMI 的建議 中的安全性考量，並遵照推薦的動作進行。
- 使用 - 共享 AMI 後，使用者只能從 AMI 啟動執行個體。他們無法刪除、共用或修改它。不過，他們在使用您的 AMI 啟動了執行個體之後，就可以從其啟動的執行個體建立 AMI。
- 自動棄用 - 在預設情況下，所有公用 AMI 的棄用日期都設定為自 AMI 建立日期起兩年。您可將棄用日期設定為早於兩年。若要取消棄用日期，或將棄用移至稍後日期，您必須僅與特定 共用 AMI，使其 AWS 帳戶成為私有。
- 移除過時的 AMIs – 在公有 AMI 到達其棄用日期之後，如果六個月或更長時間沒有從 AMI 啟動新的執行個體， AWS 則 會移除公有共用屬性，讓過時的 AMIs 不會出現在公有 AMI 清單中。
- 帳單 – 當其他 使用 AMI AWS 帳戶 啟動執行個體時，您不需要付費。使用 AMI 啟動執行個體的帳戶會根據它們啟動的執行個體計費。

與所有 AWS 帳戶共用 AMI （公開共用）

將 AMI 公有化之後，可以在主控台的社群 AMI 中使用，您可以從 EC2 主控台左側導覽器中的 AMI 目錄存取，也可在使用主控台啟動執行個體時進行存取。請注意，AMI 公有化後要經過一段時間才會出現在 Community AMI (社群 AMI) 內。

Console

使 AMI 公有化

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 AMI。
3. 從清單中選取您的 AMI，然後選取 Actions (動作)、Edit AMI permissions (編輯 AMI 許可)。
4. 在 AMI 可用性下，選擇公有。
5. 選擇儲存變更。

AWS CLI

每個 AMI 都有一個launchPermission屬性，可控制除了擁有者的 AWS 帳戶之外，還允許使用該 AMI 來啟動執行個體。透過修改 AMI 的 launchPermission 屬性，您可以將 AMI 設為公有（授予所有 的啟動許可 AWS 帳戶)，或僅與您指定的 AWS 帳戶 共用。

您可在具有 AMI 啟動許可的帳戶清單中新增或移除帳戶 ID。若要將 AMI 設定為公有，請指定 all 群組。您可同時指定公有和明確啟動許可。

使 AMI 公有化

1. 如下所示，使用 modify-image-attribute 命令，新增 all 群組至指定 AMI 的 launchPermission 清單。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Add=[{Group=all}]"
```

2. 若要確認 AMI 的啟動許可，請使用 describe-image-attribute 命令。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

3. (選用) 若要重新將 AMI 設定為私有，請從啟動許可中移除 all 群組。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{Group=all}]"
```

PowerShell

每個 AMI 都有一個launchPermission屬性，可控制除了擁有者的 AWS 帳戶之外，還允許使用該 AMI 來啟動執行個體。透過修改 AMI 的 launchPermission 屬性，您可以將 AMI 設為公有（授予所有 的啟動許可 AWS 帳戶），或僅與您指定的 AWS 帳戶 共用。

您可在具有 AMI 啟動許可的帳戶清單中新增或移除帳戶 ID。若要將 AMI 設定為公有，請指定 all 群組。您可同時指定公有和明確啟動許可。

使 AMI 公有化

1. 如下所示，使用 Edit-EC2ImageAttribute 命令，新增 all 群組至指定 AMI 的 launchPermission 清單。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserGroup all
```

2. 若要確認 AMI 的啟動許可，請使用下列 Get-EC2ImageAttribute 命令。

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

3. (選用) 若要重新將 AMI 設定為私有，請從啟動許可中移除 all 群組。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -UserGroup all
```

了解封鎖 AMI 的公有存取權

若要阻止公開共用您的 AMI，您可以在帳戶層級啟用封鎖 AMI 的公有存取權。

啟用「封鎖公有存取權」時，將會自動封鎖任何將 AMI 公開的嘗試。但是，若您已有公有 AMI，則其將保持公開可用。

若要公開共用 AMI，則需要停用封鎖公有存取權。完成共用後，最佳實務是重新啟用「封鎖公有存取權」，以防止任何非預期性公開共用您的 AMI。

> **Note**
>
> 此設定是在帳戶層級配置，可直接在帳戶中設定，或使用宣告式政策設定。它必須在您想要防止公開共用 AMIs的每個 AWS 區域 中設定。您可使用宣告式政策同時在多個區域及多個帳戶套用設定。使用宣告式政策時，您無法直接在帳戶中修改設定。本主題說明如何直接在帳戶內配置設定。如需使用宣告式政策的相關資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

您可以將 IAM 許可限制為管理員使用者，這樣便只有他們才能啟用或停用「封鎖 AMI 的公有存取權」。

主題

- 預設設定
- 管理封鎖 AMI 的公有存取權設定

預設設定

根據您的帳戶是新帳戶還是現有帳戶，以及您是否擁有公有 AMI，封鎖 AMI 的公有存取權設定預設為啟用或停用。下表列出了預設的設定：

AWS 帳戶	封鎖 AMI 預設設定的公有存取權
新帳戶	已啟用
沒有公有 AMI 的現有帳戶 [1]	已啟用
具有一個或多個公有 AMI 的現有帳戶	Disabled

[1] 如果您的帳戶在 2023 年 7 月 15 日或之後擁有一個或多個公有 AMI，即使您隨後將所有 AMI 設為私有，依預設會停用您帳戶的封鎖 AMI 的公有存取權。

管理封鎖 AMI 的公有存取權設定

您可管理封鎖您 AMI 的公有存取權設定，以控制可否公開分享 AMI。您可使用 Amazon EC2 主控台或 AWS CLI來啟用、停用或檢視 AMI 目前的封鎖公有存取權狀態。

檢視封鎖 AMI 的公有存取權的狀態

若要查看您帳戶中 AMI 的公開共用是否已封鎖，您可以檢視封鎖 AMI 之公有存取權的狀態。您必須檢視您希望查看您的 AMI 的公開共用是否已封鎖之每個 AWS 區域 的狀態。

所需的許可

若要取得目前封鎖 AMI 的公有存取權設定，您必須擁有 `GetImageBlockPublicAccessState` IAM 許可。

Console

在指定區域中檢視封鎖 AMI 的公有存取權的狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽列 (位於畫面頂端) 中，選取要檢視封鎖 AMI 的公有存取權的狀態的區域。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。
5. 在封鎖 AMI 的公有存取權下，請勾選公有存取權欄位。值可能是已封鎖新的公開共用，也可能是已允許新的公開共用。

AWS CLI

若要取得封鎖 AMI 的公有存取權狀態

使用 get-image-block-public-access-state 命令。此值為 `block-new-sharing` 或 `unblocked`。

範例：針對特定區域

```
aws ec2 get-image-block-public-access-state --region us-east-1
```

`ManagedBy` 欄位指示配置設定的實體。在此範例中，`account` 表示設定是直接在帳戶內配置。值為 `declarative-policy` 表示設定是透過宣告式政策來配置。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

```
{
    "ImageBlockPublicAccessState": "block-new-sharing",
    "ManagedBy": "account"
}
```

範例：針對帳戶中的所有區域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-image-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下為範例輸出。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       unblocked
eu-west-3        block-new-sharing
...
```

PowerShell

若要取得封鎖 AMI 的公有存取權狀態

使用 Get-EC2ImageBlockPublicAccessState cmdlet。此值為 `block-new-sharing` 或 `unblocked`。

範例：針對特定區域

```
Get-EC2ImageBlockPublicAccessState -Region us-east-1
```

以下為範例輸出。

```
block-new-sharing
```

範例：針對帳戶中的所有區域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
        [PSCustomObject]@{
            Region   = $_
            PublicAccessState = (Get-EC2ImageBlockPublicAccessState -Region $_)
        }
} | `
Format-Table -AutoSize
```

以下為範例輸出。

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

啟用「封鎖 AMI 的公有存取權」

若要阻止公開共用您的 AMI，請在帳戶層級啟用「封鎖 AMI 的公有存取權」。您必須在您希望阻止公開共用您的 AMI 的每個 AWS 區域 中啟用「封鎖 AMI 的公有存取權」。若您已有公有 AMI，則它們將保持公開可用。

所需的許可

若要啟用封鎖 AMI 的公有存取權設定，您必須擁有 EnableImageBlockPublicAccess IAM 許可。

考量事項

- 可能最多需要 10 分鐘的時間來設定此設定。在此期間，若描述公有存取權狀態，回應為 unblocked。完成組態設定後，回應為 block-new-sharing。

Console

在指定區域中啟用 AMI 的封鎖公開存取

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽列 (位於畫面頂端) 中，選取要啟用「封鎖 AMI 的公有存取權」的區域。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。
5. 在封鎖 AMI 的公有存取權下，選擇管理。
6. 選取封鎖新的公開共用核取方塊，然後選擇更新。

AWS CLI

啟用封鎖 AMI 的公有存取權

使用 enable-image-block-public-access 命令。

範例：針對特定區域

```
aws ec2 enable-image-block-public-access \
--region us-east-1 \
--image-block-public-access-state block-new-sharing
```

以下為範例輸出。

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

範例：針對帳戶中的所有區域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
```

```
        aws ec2 enable-image-block-public-access \
            --region $region \
            --image-block-public-access-state block-new-sharing \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下為範例輸出。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       block-new-sharing
eu-west-3        block-new-sharing
...
```

PowerShell

啟用封鎖 AMI 的公有存取權

使用 Enable-EC2ImageBlockPublicAccess 命令。

範例：針對特定區域

```
Enable-EC2ImageBlockPublicAccess `
    -Region us-east-1 `
    -ImageBlockPublicAccessState block-new-sharing
```

以下為範例輸出。

```
Value
-----
block-new-sharing
```

範例：針對帳戶中的所有區域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (
```

```
        Enable-EC2ImageBlockPublicAccess `
         -Region $_ `
         -ImageBlockPublicAccessState block-new-sharing)
    }
} | `
Format-Table -AutoSize
```

以下為範例輸出。

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

停用「封鎖 AMI 的公有存取權」

若要允許您帳戶中的使用者公開共用您的 AMI，請在帳戶層級停用「封鎖 AMI 的公有存取權」。在您要允許公開共用 AMIs 的每個 AWS 區域 中，您必須停用封鎖 AMIs的公開存取。

所需的許可

若要停用封鎖 AMI 的公有存取權設定，您必須擁有 `DisableImageBlockPublicAccess` IAM 許可。

考量事項

- 可能最多需要 10 分鐘的時間來設定此設定。在此期間，若描述公有存取權狀態，回應為 `block-new-sharing`。完成組態設定後，回應為 `unblocked`。

Console

在指定區域中停用 AMI 的封鎖公開存取

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽列 (位於畫面頂端) 中，選取要停用「封鎖 AMI 的公有存取權」的區域。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。

5. 在封鎖 AMI 的公有存取權下，選擇管理。
6. 清除封鎖新的公開共用核取方塊，然後選擇更新。
7. 出現確認提示時，請輸入 **confirm**，然後選擇允許公開共用。

AWS CLI

停用封鎖 AMI 的公有存取權

使用 disable-image-block-public-access 命令。

範例：針對特定區域

```
aws ec2 disable-image-block-public-access --region us-east-1
```

以下為範例輸出。

```
{
   "ImageBlockPublicAccessState": "unblocked"
}
```

範例：針對帳戶中的所有區域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-image-block-public-access \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下為範例輸出。

```
Region           Public Access State
```

```
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
...
```

PowerShell

停用封鎖 AMI 的公有存取權

使用 Disable-EC2ImageBlockPublicAccess cmdlet。

範例：針對特定區域

```
Disable-EC2ImageBlockPublicAccess -Region us-east-1
```

以下為範例輸出。

```
Value
-----
unblocked
```

範例：針對帳戶中的所有區域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2ImageBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

以下為範例輸出。

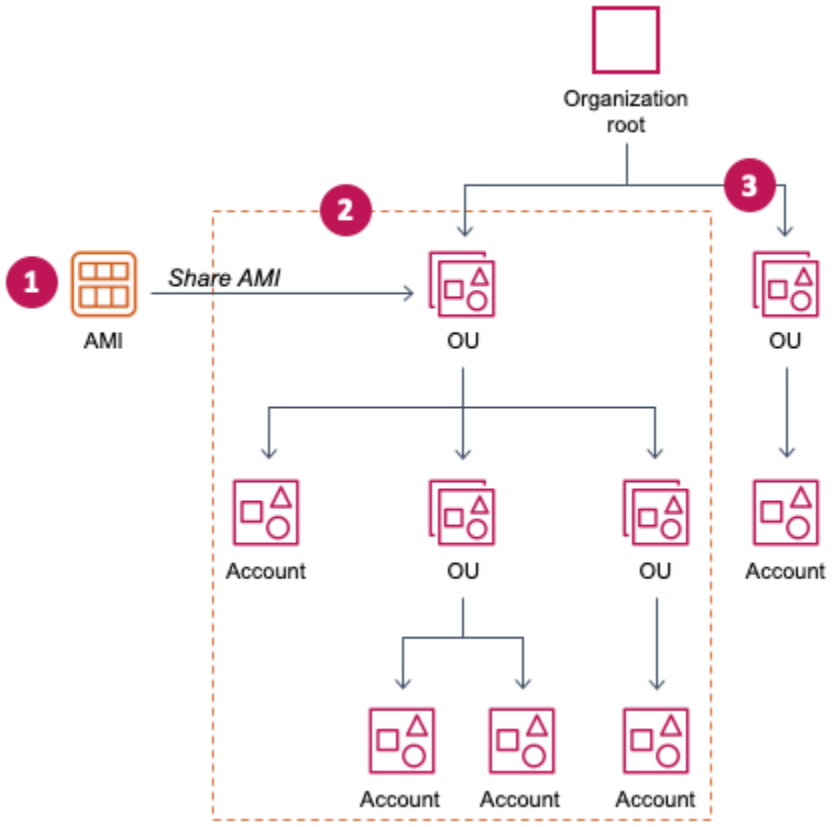
```
Region         PublicAccessState
------         -----------------
ap-south-1     unblocked
eu-north-1     unblocked
eu-west-3      unblocked
...
```

與組織或組織單位共用 AMI

AWS Organizations 是一種帳戶管理服務，可讓您將多個 合併 AWS 帳戶 到您建立並集中管理的組織。除了與指定的帳戶共享 AMI 之外，您還可以與組織或您建立的組織單位 (OU) 共享 AMI。

組織是您建立來合併並集中管理 AWS 帳戶的實體。您可以將帳戶組織為階層式、類似樹狀目錄的結構，其中的根位於頂層，而組織單位在組織根下形成巢狀。每個帳戶都可以直接新增至根中，或放在階層中的其中一個 OU 中。如需帳戶類型的詳細資訊，請參閱「AWS Organizations 使用者指南」中的 AWS Organizations 術語與概念。

當您與組織或 OU 共用 AMI 時，所有子帳戶都可以存取 AMI。例如，在下圖中，與頂層 OU 共用 AMI (由編號 1 的箭頭表示)。在該頂層 OU 之下形成巢狀的所有 OU 和帳戶 (由編號 2 的虛線表示) 也可以存取 AMI。組織中的帳戶和虛線外的 OU (由編號 3 表示) 無法存取 AMI，因為它們不是與其共用 AMI 之 OU 的子項。



主題

- 考量事項
- 取得組織或組織單位的 ARN
- 允許組織和 OU 使用 KMS 金鑰

- 管理與組織或 OU 的 AMI 共用

考量事項

與特定組織或組織單位共享 AMI 時，請考慮以下項目。

- 擁有權 – 若要共用 AMI，您的 AWS 帳戶 必須擁有 AMI。
- 共用限制 – AMI 擁有者可以與任何組織或 OU 共用 AMI，包括他們不是其成員的組織和 OU。

  如需可在區域內共用 AMI 的最大實體數量，請參閱 Amazon EC2 服務配額。
- 標籤 – 您無法共用使用者定義的標籤 (連接至 AMI 的標籤)。當您共用 AMI 時，使用者定義的標籤不適用於與 AMI 共用的組織或 OU AWS 帳戶 中的任何 。
- ARN 格式 - 在命令中指定組織或 OU 時，請務必使用正確的 ARN 格式。如果只指定 ID，則您會收到錯誤，例如，如果只指定 `o-123example` 或 `ou-1234-5example`。

  正確的 ARN 格式：

  - 組織 ARN：`arn:aws:organizations::`*`111122223333`*`:organization/`*`organization-id`*
  - OU ARN：`arn:aws:organizations::`*`111122223333`*`:ou/`*`organization-id`*`/`*`ou-id`*

  其中：

  - *`111122223333`* 為管理帳戶 12 位數帳戶 ID 範例。如果不知道管理帳號，您可以描述組織或組織單位以取得 ARN，其中包括管理帳號。如需詳細資訊，請參閱 取得組織或組織單位的 ARN。
  - *`organization-id`* 是組織 ID，例如，`o-123example`。
  - *`ou-id`* 是組織單位 ID，例如，`ou-1234-5example`。

  如需 ARN 格式的詳細資料，請參閱「IAM 使用者指南」中的 Amazon Resource Name (ARN)。
- 加密和金鑰 – 您可以共用未加密和加密快照所支援的 AMI。
  - 加密快照必須透過客戶受管金鑰進行加密。您無法共用由使用預設 AWS 受管金鑰加密的快照所支援的 AMIs。
  - 如果您共用加密快照所支援的 AMI，則必須允許組織或 OU 使用客戶受管金鑰，而這些金鑰用來加密快照。如需詳細資訊，請參閱允許組織和 OU 使用 KMS 金鑰。
- 區域 - AMI 是一種區域性的資源。共享 AMI 後，只能從共享 AMI 的區域中使用 AMI。若要讓 AMI 在其他區域可用，請將 AMI 複製到該區域，然後共用。如需詳細資訊，請參閱複製 Amazon EC2 AMI。

- 使用 - 共享 AMI 後，使用者只能從 AMI 啟動執行個體。他們無法刪除、共用或修改它。不過，他們在使用您的 AMI 啟動了執行個體之後，就可以從其啟動的執行個體建立 AMI。
- 帳單 – 當其他 使用 AMI AWS 帳戶 啟動執行個體時，您不需要付費。使用 AMI 啟動執行個體的帳戶會根據它們啟動的執行個體計費。

取得組織或組織單位的 ARN

組織和組織單位 ARN 包含 12 位數的管理帳號。如果不知道管理帳號，您可以描述組織或組織單位以取得其中每一個的 ARN。在下面的範例中，123456789012 為管理帳號。

所需的許可

在取得 ARN 之前，您必須具有描述組織和組織單位的許可。下列政策提供必要的許可。

JSON

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

AWS CLI

取得組織的 ARN

使用 describe-organization 命令。新增 `--query` 選項，以便僅傳回組織 ARN。

```
aws organizations describe-organization --query 'Organization.Arn'
```

以下為範例輸出。

```
"arn:aws:organizations::123456789012:organization/o-1234567abc"
```

取得組織單位的 ARN

使用 describe-organizational-unit 命令。使用 `--query` 參數，以便僅傳回組織單位 ARN。

```
aws organizations describe-organizational-unit \
    --organizational-unit-id ou-a123-b4567890 \
    --query 'OrganizationalUnit.Arn'
```

以下為範例輸出。

```
"arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890"
```

PowerShell

取得組織的 ARN

使用 Get-ORGOrganization cmdlet。

```
(Get-ORGOrganization).Arn
```

以下為範例輸出。

```
arn:aws:organizations::123456789012:organization/o-1234567abc
```

取得組織單位的 ARN

使用 Get-ORGOrganizationalUnit cmdlet。

```
(Get-ORGOrganizationalUnit -OrganizationalUnitId "ou-a123-b4567890").Arn
```

以下為範例輸出。

```
arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890
```

允許組織和 OU 使用 KMS 金鑰

如果您共用加密快照支援的 AMI，您還必須允許組織或組織單位 (OU) 使用用於加密快照的 KMS 金鑰。

> **Note**
>
> 加密快照必須透過客戶受管金鑰進行加密。您無法共用由使用預設 AWS 受管金鑰加密的快照所支援的 AMIs。

若要控制 KMS 金鑰的存取權，您可在金鑰政策使用 aws:PrincipalOrgID 和 aws:PrincipalOrgPaths 條件金鑰，僅允許特定主體對指定動作的權限。委託人可以是使用者、IAM 角色、聯合身分使用者或 AWS 帳戶 根使用者。

條件金鑰使用方式如下：

- aws:PrincipalOrgID - 允許屬於指定 ID 所代表組織的任何主體。
- aws:PrincipalOrgPaths - 允許屬於指定路徑所代表 OU 的任何主體。

若要授予組織 (包括所屬的 OU 和帳戶) 使用 KMS 金鑰的權限，請將以下陳述式新增至金鑰政策。

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

若要授予特定 OU (及其所屬帳戶) 使用 KMS 金鑰的權限，您可以使用與下列範例類似的政策。

```
{
        "Sid": "Allow access for specific OUs and their descendants",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "kms:Describe*",
            "kms:List*",
            "kms:Get*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:CreateGrant"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": "o-123example"
            },
            "ForAnyValue:StringLike": {
                "aws:PrincipalOrgPaths": [
                    "o-123example/r-ab12/ou-ab12-33333333/*",
                    "o-123example/r-ab12/ou-ab12-22222222/*"
                ]
            }
        }
}
```

如需更多條件陳述式範例，請參閱「IAM 使用者指南」中的 aws:PrincipalOrgID 和 aws:PrincipalOrgPaths。

如需跨帳户存取權的更多相關資訊，請參閱「AWS Key Management Service 開發人員指南」中的允許其他帳戶中的使用者使用 KMS 金鑰。

管理與組織或 OU 的 AMI 共用

您可以管理與組織和組織單位 (OU) 共用的 AMI，以控制其可否啟動 Amazon EC2 執行個體。

查看與之共用 AMI 的組織和 OU

您可尋找您與之共享 AMI 的組織和 OU

Console

檢查與之共用 AMI 的組織和 OU

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 從清單中選取 AMI，選取 Permissions (許可) 索引標籤，然後向下滾動至 Shared organizations/OUs (共用的組織/OU)。

   若要尋找與您共用的 AMI，請參閱尋找用於 Amazon EC2 執行個體的共用 AMI。

AWS CLI

檢查與之共用 AMI 的組織和 OU

請使用 `launchPermission` 屬性並搭配 describe-image-attribute 命令。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

以下是回應範例。

```
{
    "ImageId": "ami-0abcdef1234567890",
    "LaunchPermissions": [
        {
            "OrganizationalUnitArn": "arn:aws:organizations::111122223333:ou/
o-123example/ou-1234-5example"
        }
    ]
}
```

PowerShell

檢查與之共用 AMI 的組織和 OU

使用 Get-EC2ImageAttribute cmdlet。

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
```

```
    -Attribute launchPermission
```

與組織或 OU 共用 AMI

您可與組織或 OU 共享 AMI。

> **Note**
>
> 您不需要共用 AMI 參考的 Amazon EBS 快照即可共用 AMI。需要共享的只有 AMI 本身，而且系統可自動讓執行個體存取參考的 EBS 快照，以進行啟動。不過，您的確需要共用 KMS 金鑰，而這些金鑰用來加密 AMI 參考的快照。如需詳細資訊，請參閱允許組織和 OU 使用 KMS 金鑰。

Console

與組織或 OU 共用 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 在清單中選取您的 AMI，然後選取 Actions (動作)、Edit AMI permissions (編輯 AMI 許可)。
4. 在 AMI availability (AMI 可用性) 下，選擇 Private (私有)。
5. 在 Shared organizations/OUs (共用組織/OU) 旁邊，選擇 Add organization/OU ARN (新增組織/OU ARN)。
6. 對於 Organization/OU ARN (組織/OU ARN)，輸入要與之共用 AMI 的組織 ARN 或 OU ARN，然後選擇 Share AMI (共用 AMI)。請注意，您必須指定完整的 ARN，而不只是 ID。

   若要與多個組織或 OU 共用此 AMI，請重覆此步驟，直到新增所有需要的組織或 OU。
7. 完成後，請選擇 Save changes (儲存變更)。
8. (選用) 若要查看與其共用 AMI 的組織或 OU，請從清單中選取 AMI，然後選取 Permissions (許可) 索引標籤，向下滾動至 Shared organizations/OUs (共用的組織/OU)。若要尋找與您共用的 AMI，請參閱尋找用於 Amazon EC2 執行個體的共用 AMI。

AWS CLI

與組織共用 AMI

使用 modify-image-attribute 命令，可將所指定 AMI 的啟動許可授予給指定組織。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

與 OU 共用 AMI

modify-image-attribute 命令可將所指定 AMI 的啟動許可授予指定的 OU。請注意，您必須指定完整的 ARN，而不只是 ID。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationalUnitArn=arn:aws:organizations::123456789012:ou/o-123example/
ou-1234-5example}]"
```

PowerShell

使用 Edit-EC2ImageAttribute 命令 (Tools for Windows PowerShell) 來共用 AMI，如下列範例所示。

與組織或 OU 共用 AMI

下列命令可將所指定 AMI 的啟動許可授予指定的組織。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止與組織或 OU 共用 AMI

下列命令可從指定的組織移除所指定 AMI 的啟動許可。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
```

```
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止與所有組織、OUs 和 共用 AMI AWS 帳戶

下列命令可從指定的 AMI 移除所有的公有和明確啟動許可。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

停止與組織或 OU 共用 AMI

您可停止與組織或 OU 共享 AMI。

> **Note**
>
> 如果特定帳戶位於與其共用 AMI 的組織或 OU 中，則無法停止與該帳戶共用 AMI。如果您嘗試透過移除帳戶的啟動許可來停止共用 AMI，Amazon EC2 會傳回成功訊息。不過，會繼續與該帳戶共用 AMI。

Console

停止與組織或 OU 共用 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMIs。
3. 在清單中選取您的 AMI，然後選取 Actions (動作)、Edit AMI permissions (編輯 AMI 許可)。
4. 在 Shared organizations/OUs (共用的組織/OU)，選取要停止與其共用 AMI 的組織或 OU，然後選取 Remove Selected (移除所選項)。
5. 完成後，請選擇 Save changes (儲存變更)。
6. (選用) 若要確認您已停止與組織或 OU 共用 AMI，請從清單中選取此 AMI，然後選取Permissions (許可) 索引標籤，向下滾動至 Shared organizations/OUs (共用的組織/OU)。

AWS CLI

停止與組織或 OU 共用 AMI

請使用 modify-image-attribute 命令。此範例可從指定的組織中移除所指定 AMI 的啟動許可。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Remove=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

停止與所有組織、OUs 和 共用 AMI AWS 帳戶

使用 reset-image-attribute 命令。此範例可從指定的 AMI 中移除所有的公有和明確啟動許可。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

停止與組織或 OU 共用 AMI

使用 Edit-EC2ImageAttribute cmdlet。此範例可從指定的組織中移除所指定 AMI 的啟動許可。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止與所有組織、OUs 和 共用 AMI AWS 帳戶

使用 Reset-EC2ImageAttribute cmdlet。此範例可從指定的 AMI 中移除所有的公有和明確啟動許可。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LaunchPermission
```

與特定 AWS 帳戶共用 AMI

您可以與特定 共用 AMI， AWS 帳戶 而無需公開 AMI。您只需要 AWS 帳戶 IDs

AWS 帳戶 ID 是可唯一識別 的 12 位數號碼012345678901，例如 AWS 帳戶。如需詳細資訊，請參閱 AWS 帳戶管理 參考指南中的檢視 AWS 帳戶 識別符。

考量事項

與特定 共用 AMIs時，請考慮下列事項 AWS 帳戶。

- 擁有權 – 若要共用 AMI，您的 AWS 帳戶 必須擁有 AMI。
- 共用限制 - 如需可在區域內共用 AMI 的最大實體數量，請參閱 Amazon EC2 服務配額。
- 標籤 – 您無法共用使用者定義的標籤 (連接至 AMI 的標籤)。當您共用 AMI 時，使用者定義的標籤不適用於與 AWS 帳戶 AMI 共用的任何 。
- 快照 – 共享 AMI 時，不需要共享 AMI 所參考的 Amazon EBS 快照。您可僅共享 AMI 本身；系統會為執行個體提供所參考 EBS 快照的存取權，供其啟動。不過，您必須共享任何用來加密快照的 KMS 金鑰，而 AMI 會參考這些快照。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的共用 Amazon EBS 快照。
- 加密和金鑰 – 您可以共用未加密和加密快照所支援的 AMI。
  - 加密快照必須透過 KMS 金鑰進行加密。您無法共用透過預設 AWS 受管金鑰加密之快照支援的 AMI。
  - 如果您共用由加密快照支援的 AMI，您必須允許 AWS 帳戶 使用用來加密快照的 KMS 金鑰。如需詳細資訊，請參閱允許組織和 OU 使用 KMS 金鑰。若要設定當您使用客戶受管金鑰進行加密時啟動 Auto Scaling 執行個體所需的金鑰政策，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的搭配加密磁碟區使用的必要 AWS KMS key 政策。
- 區域 - AMI 是一種區域性的資源。當您共用 AMI 時，其只能在該區域中使用。若要讓 AMI 在其他區域可用，請將 AMI 複製到該區域，然後共用。如需詳細資訊，請參閱複製 Amazon EC2 AMI。
- 使用 - 共享 AMI 後，使用者只能從 AMI 啟動執行個體。他們無法刪除、共用或修改它。不過，在使用您的 AMI 啟動了執行個體之後，他們可以從其執行個體建立 AMI。
- 複製共享 AMI – 如果其他帳戶中的使用者想要複製共享 AMI，則必須向他們授予用於支援 AMI 的儲存的讀取許可。如需詳細資訊，請參閱跨帳戶複製。
- 帳單 – 當其他人使用您的 AMI AWS 帳戶 啟動執行個體時，您不需要付費。使用 AMI 啟動執行個體的帳戶會根據它們啟動的執行個體計費。

Console

授予明確啟動許可

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 在清單中選取您的 AMI，然後選取 Actions (動作)、Edit AMI permissions (編輯 AMI 許可)。
4. 選擇 Private (私有)。
5. 在 Shared accounts (共享帳戶) 中，選擇 Add account ID (新增帳戶 ID)。
6. 針對 AWS 帳戶 ID，輸入您要共用 AMI 的 AWS 帳戶 ID，然後選擇共用 AMI。

   若要與多個帳戶共用此 AMI，請重覆步驟 5 和 6，直到您新增了所有必要的帳戶 ID。
7. 完成後，請選擇 Save changes (儲存變更)。
8. （選用）若要檢視您已共用 AMI 的 AWS 帳戶 IDs，請在清單中選取 AMI，然後選擇許可索引標籤。若要尋找與您共用的 AMI，請參閱尋找用於 Amazon EC2 執行個體的共用 AMI。

AWS CLI

請使用 modify-image-attribute 命令來共享 AMI，如下列範例所示。

授予明確啟動許可

下列範例可向指定的 AWS 帳戶授予指定 AMI 的啟動許可。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Add=[{UserId=123456789012}]"
```

移除帳戶的啟動許可

下列範例可從指定的 AWS 帳戶移除指定 AMI 的啟動許可。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

移除所有的啟動許可

下列範例可從指定的 AMI 移除所有的公有和明確啟動許可。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

使用 Edit-EC2ImageAttribute 命令 (Tools for Windows PowerShell) 來共用 AMI，如下列範例所示。

授予明確啟動許可

下列範例可向指定的 AWS 帳戶授予指定 AMI 的啟動許可。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserId "123456789012"
```

移除帳戶的啟動許可

下列範例可從指定的 AWS 帳戶移除指定 AMI 的啟動許可。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission -OperationType remove `
    -UserId "123456789012"
```

移除所有的啟動許可

下列範例可從指定的 AMI 移除所有的公有和明確啟動許可。請注意，AMI 擁有者始終具備啟動許可，因此不受此命令影響。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

取消將 AMI 與您的 共用 AWS 帳戶

透過將帳戶新增到 AMI 的啟動許可，可以與指定的 AWS 帳戶共用 Amazon Machine Image (AMI)。如果 AMI 已與 共用， AWS 帳戶 而且您不希望再與您的帳戶共用，您可以從 AMI 的啟動許可中移除您的帳戶。您可以執行 `cancel-image-launch-permission` AWS CLI 命令來執行此操作。執行此命令時，您的 AWS 帳戶 會從指定 AMI 的啟動許可中移除。若要尋找與 共用AMIs AWS 帳戶，請參閱尋找用於 Amazon EC2 執行個體的共用 AMI。

例如，您可能會取消與您的帳戶共用 AMI，以降低啟動與您共用的未使用或已棄用 AMI 的執行個體的可能性。當您取消與您的帳戶共用 AMI 時，它將不再出現在 EC2 主控台的任何 AMI 清單或 describe-images 的輸出中。

主題

- 限制
- 取消與您的帳戶共用 AMI

限制

- 您可以從 AWS 帳戶 僅與 共用的 AMI 啟動許可中移除您的帳戶。您無法使用 `cancel-image-launch-permission` 從與組織或組織單位 (OU) 共用的 AMI 的啟動許可中移除帳戶，也無法移除公用 AMI 的存取權。
- 您無法從 AMI 的啟動許可中永久移除您的帳戶。AMI 擁有者可再次與您的帳戶共用 AMI。
- AMI 是一種區域性的資源。執行 `cancel-image-launch-permission` 時，必須指定 AMI 所在的區域。在命令中指定區域，或使用 AWS_DEFAULT_REGION 環境變數。
- 只有 AWS CLI 和 SDKs支援從 AMI 的啟動許可中移除您的帳戶。EC2 主控台目前不支援此動作。

取消與您的帳戶共用 AMI

> **Note**
>
> 取消與您的帳戶共用 AMI 後，將無法復原。若要重新獲得對 AMI 的存取權，AMI 擁有者必須與您的帳戶共用它。

AWS CLI

如需取消與您的帳戶共享 AMI

使用 cancel-image-launch-permission 命令。

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0abcdef1234567890 \
    --region us-east-1
```

PowerShell

如需取消與您的帳戶共享 AMI

使用 Stop-EC2ImageLaunchPermission cmdlet。

```
Stop-EC2ImageLaunchPermission `
    -ImageId ami-0abcdef1234567890 `
    -Region us-east-1
```

建立共用 Linux AMI 的建議

請使用下列準則來減少可能遭受攻擊的範圍，以及改善所建立 AMI 的可靠性。

> **Important**
>
> 沒有任何一份清單可詳盡列出所有的安全性準則。請小心建置您的共享 AMI，並花些時間思考那些位置可能暴露機密資料。

目錄

- 停用根使用者的密碼型態遠端登入
- 停用本機根存取
- 移除 SSH 主機金鑰對
- 安裝公有金鑰登入資料
- 停用 sshd DNS 檢查 (選用)
- 移除敏感性資料

如果您要為 建置 AMIs AWS Marketplace，請參閱《 AWS Marketplace 賣方指南》中的建置 AMIs 的最佳實務，以取得指導方針、政策和最佳實務。

停用根使用者的密碼型態遠端登入

公用 AMI 使用固定的根密碼會帶來安全風險，且資訊會迅速向外散佈。即便仰賴使用者於第一次登入時變更密碼，都會留下一些可能遭到濫用的空窗。

若要解決此問題，請停用根使用者的密碼型態遠端登入。

停用根使用者的密碼型態遠端登入

1. 使用文字編輯器開啟 /etc/ssh/sshd_config 檔案，並找出下行：

   ```
   #PermitRootLogin yes
   ```

2. 將行變為：

   ```
   PermitRootLogin without-password
   ```

   此組態檔案的位置可能與您的發行版本不同，或您不是執行 OpenSSH。若是這種情況，請參閱相關文件。

停用本機根存取

使用共享 AMI 時，最佳實務之一就是停用直接根登入。若要停用，請登入運作中的執行個體，然後發出下列命令：

```
[ec2-user ~]$ sudo passwd -l root
```

> **Note**
>
> 此命令不會影響 sudo 的使用。

移除 SSH 主機金鑰對

如果您打算共享來自公用 AMI 的 AMI，請移除 /etc/ssh 中現有的 SSH 主機金鑰對。有其他人使用您的 AMI 啟動執行個體時，會強迫 SSH 產生新的唯一 SSH 金鑰對，藉此以改善安全性和減少發生「中間人」攻擊的可能性。

移除位在您系統中所有下列的金鑰檔案。

- ssh_host_dsa_key
- ssh_host_dsa_key.pub
- ssh_host_key
- ssh_host_key.pub
- ssh_host_rsa_key
- ssh_host_rsa_key.pub
- ssh_host_ecdsa_key
- ssh_host_ecdsa_key.pub
- ssh_host_ed25519_key
- ssh_host_ed25519_key.pub

您可以使用下列命令安全地移除所有這些檔案：

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

Warning

shred 等安全移除公用程式可能不會從儲存媒體中移除檔案的所有複本。檔案的隱藏複本可能由日誌檔案系統 (包含 Amazon Linux 預設 ext4)、快照、備份、RAID 和暫時性快取建立。若要了解詳細資訊，請參閱銷毀文件。

Important

如果您忘記從公用 AMI 中移除現有的 SSH 主機金鑰對，我們的定期稽核流程會通知您和執行您可能有安全風險之 AMI 執行個體的所有客戶。過一段短時間的寬限期後，我們會將 AMI 標示為私有。

安裝公有金鑰登入資料

設定 AMI 避免使用密碼登入後，您必須確定使用者可用其他機制登入。

Amazon EC2 允許使用者在啟動執行個體時指定公私金鑰對名稱。為 RunInstances API 呼叫 (或透過命令列 API 工具) 提供有效的金鑰對名稱時，公有金鑰 (呼叫 CreateKeyPair 或 ImportKeyPair

後金鑰對中 Amazon EC2 保留在伺服器上的部分) 會透過執行個體中繼資料的 HTTP 查詢提供給執行個體。

若要透過 SSH 登入，您的 AMI 必須在開機時擷取金鑰值，並將其附加至 `/root/.ssh/authorized_keys` (或 AMI 上任何其他使用者帳戶的對等檔案)。使用者可用金鑰對啟動您 AMI 的執行個體並登入，無需根密碼。

包括 Amazon Linux 和 Ubuntu 等許多發行版本皆使用 `cloud-init` 套件來注入所設定使用者的公有金鑰登入資料。如果您的發行版本不支援 `cloud-init`，您可新增下列程式碼至系統啟動指令碼 (例如 `/etc/rc.local`)，以納入您在啟動時為根使用者指定的公有金鑰。

> **Note**
>
> 在下列範例中，IP 地址 http://169.254.169.254/ 是 link-local 地址且僅在執行個體中有效。

IMDSv2

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

IMDSv1

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
```

```
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

這會套用到任何使用者；無需將其限制到 root 使用者。

> **Note**
>
> 重新綁定以此 AMI 為基礎的執行個體，會加入其啟動金鑰。若要避免加入金鑰，您必須清除 (或刪除) authorized_keys 檔案或將此檔案排除於重新綁定之外。

停用 sshd DNS 檢查 (選用)

停用 sshd DNS 檢查會稍微降低 sshd 的安全性。但假如 DNS 解析失敗，SSH 登入仍能如預期運作。如果您不停用 sshd 檢查，DNS 解析失敗將封鎖所有登入。

停用 sshd DNS 檢查

1. 使用文字編輯器開啟 /etc/ssh/sshd_config 檔案，並找出下行：

```
#UseDNS yes
```

2. 將行變為：

```
UseDNS no
```

> **Note**
>
> 此組態檔案的位置可能與您的發行版本不同，或您不是執行 OpenSSH。若是這種情況，請參閱相關文件。

移除敏感性資料

我們建議您不要將機密資料或軟體存放在共享的任何 AMI 上。啟動共享 AMI 的使用者可能會重新綁定，並將其註冊為自己的 AMI。請按照以下準則幫助您避免一些容易遭到忽視的安全風險：

- 建議使用 `--exclude` *`directory`* 的 `ec2-bundle-vol` 選項跳過含有您不想加入到套件內之機密資訊的任何目錄和子目錄。尤其請在綁定映像時排除所有使用者擁有的 SSH 公有/私有金鑰對和 SSH `authorized_keys` 檔案。Amazon 公有 AMI 會將這些資訊存放在根使用者的 `/root/.ssh`，和一般使用者的 `/home/`*`user_name`*`/.ssh/` 內。如需詳細資訊，請參閱ec2-bundle-vol。
- 綁定前務必刪除 shell 歷程記錄。如果您在同一 AMI 嘗試一次以上的套件上傳，shell 歷程記錄會包含您的存取金鑰。以下範例應該是您在從執行個體內綁定之前執行的最後一項命令。

```
[ec2-user ~]$ shred -u ~/.*history
```

> **Warning**
>
> 上面警告中提及的 shred 限制同樣適用於此處。
> 請注意，bash 會在退出時將目前工作階段的歷程記錄寫入到磁碟。如果您在刪除 `~/.bash_history` 後登出執行個體，然後再重新登入，您會發現 `~/.bash_history` 已重新建立，並包含在先前工作階段中執行的所有命令。
> bash 以外的其他程式也會將歷程記錄寫入到磁碟。請小心使用，並移除或排除不必要以點開頭的檔案或以點開頭的目錄。

- 綁定執行中的執行個體需要您的私有金鑰和 X.509 憑證。因此請將這些憑證和其他登入資料放在不會加入綁定的位置 (例如執行個體儲存體)。

使用 Amazon EventBridge 監控 AMI 事件

當 Amazon Machine Image (AMI) 的狀態變更時，Amazon EC2 會產生傳送至 Amazon EventBridge (之前稱為 Amazon CloudWatch Events) 的事件。事件會以 JSON 格式傳送至預設 EventBridge 事件匯流排。您可以使用 Amazon EventBridge 來偵測並對這些事件做出反應。為此，您可以在 EventBridge 中建立觸發動作以回應事件的規則。例如，您可以建立 EventBridge 規則，以偵測 AMI 建立程序完成的時間，然後呼叫 Amazon SNS 主題來傳送電子郵件通知給您。

當 AMI 進入下列任何狀態時，Amazon EC2 會產生 `EC2 AMI State Change` 事件：

- `available`
- `failed`
- `deregistered`
- `disabled`

盡可能產生事件。

下表列出 AMI 操作和 AMI 可以進入的狀態。在該表中，是表示當對應操作執行時 AMI 可進入的狀態。

AMI 操作	available	failed	deregistered	disabled
CopyImage	是	是		
CreateImage	是	是		
CreateRestoreImageTask	是	是		
DeregisterImage			是	
DisableImage				是
EnableImage	是			
RegisterImage	是	是		

EC2 AMI State Change 事件

- 事件詳細資訊
- available 事件
- failed 事件
- deregistered 事件
- disabled 事件

事件詳細資訊

您可以使用事件中的下列欄位來建立觸發動作的規則：

`"source": "aws.ec2"`

識別該事件是來自 Amazon EC2。

`"detail-type": "EC2 AMI State Change"`

識別事件名稱。

`"detail": { "ImageId": "ami-0abcdef1234567890", "State": "available", }`

提供 AMI ID 和 AMI 狀態 (`available`、`failed`、`deregistered`、或 `disabled`)。

如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的以下內容：

- Amazon EventBridge 事件
- Amazon EventBridge 事件模式
- Amazon EventBridge 規則

如需如何建立 Lambda 函數和執行 Lambda 函數的 EventBridge 規則的教學課程，請參閱「AWS Lambda 開發人員指南」中的教學課程：使用 EventBridge 記錄 Amazon EC2 執行個體的狀態。

available 事件

以下是 `CreateImage`、`CopyImage`、`RegisterImage`、`CreateRestoreImageTask` 或 `EnableImage` 操作成功之後，AMI 進入 `available` 狀態時，Amazon EC2 產生的事件範例。

`"State": "available"` 表示操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "available",
        "ErrorMessage": ""
    }
}
```

failed 事件

以下是 CreateImage、CopyImage、RegisterImage 或 CreateRestoreImageTask 操作失敗之後，AMI 進入 failed 狀態時，Amazon EC2 產生的事件範例。

下列欄位會提供相關資訊：

- "State": "failed" - 表示操作失敗。
- "ErrorMessage": "" - 提供操作失敗的原因。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "failed",
        "ErrorMessage": "Description of failure"
    }
}
```

deregistered 事件

以下是在 DeregisterImage 操作成功之後，AMI 進入 deregistered 狀態時，Amazon EC2 產生的事件範例。如果操作失敗，則不會產生任何事件。因為 DeregisterImage 是一個同步操作，所以會立刻發現所有失敗。

"State": "deregistered" 表示 DeregisterImage 操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
```

```
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "deregistered",
        "ErrorMessage": ""
    }
}
```

disabled 事件

以下是在 DisableImage 操作成功之後，AMI 進入 disabled 狀態時，Amazon EC2 產生的事件範例。如果操作失敗，則不會產生任何事件。因為 DisableImage 是一個同步操作，所以會立刻發現所有失敗。

"State": "disabled" 表示 DisableImage 操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "disabled",
        "ErrorMessage": ""
    }
}
```

了解 AMI 帳單資訊

啟動執行個體時，有多種 Amazon Machine Image (AMI) 可供選擇，而且它們支援各種作業系統平台和功能。若要了解您在啟動執行個體時選擇的 AMI 如何影響 AWS 帳單的底線，您可以研究相關聯的

作業系統平台和帳單資訊。在啟動任何隨需或 競價型執行個體，或購買 預留執行個體 之前執行此動作。

以下兩個例子說明事先研究 AMI 如何幫助您選擇最符合需求的 AMI：

- 針對 競價型執行個體，您可以使用 AMI 平台詳細資訊來確認 競價型執行個體 是否支援 AMI。
- 購買 預留執行個體 時，您可確保對應 AMI 平台詳細資訊選取操作系統平台 (平台)。

如需執行個體定價的詳細資訊，請參閱 Amazon EC2 定價。

內容

- AMI 帳單資訊欄位
- 尋找 AMI 帳單和用量詳細資訊
- 驗證帳單上的 AMI 費用

AMI 帳單資訊欄位

下列欄位提供與 AMI 相關的帳單資訊：

平台詳細資料

與 AMI 帳單代碼相關的平台詳細資訊。例如：`Red Hat Enterprise Linux`。

使用操作

Amazon EC2 執行個體的操作以及與 AMI 相關聯的帳單代碼。例如：`RunInstances:0010`。用量操作會對應至 AWS 成本和用量報告 (CUR) 和AWS 價格清單 API 中的 lineitem/Operation 欄。

您可以在 Amazon EC2 主控台的執行個體或 AMI 頁面上，或是在 describe-image 或 Get-EC2Image 命令傳回的回應中，檢視這些欄位。

範例資料：使用操作 (依平台)

下表列出可在 Amazon EC2 主控台的執行個體或 AMI 頁面上，或是在 describe-image 或 Get-EC2Image 命令傳回的回應中顯示的部分平台詳細資訊和使用操作值。

平台詳細資料	使用操作 [2]

平台詳細資料	使用操作 [2]
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0 [3]
Red Hat Enterprise Linux	RunInstances:0010
Red Hat Enterprise Linux with HA	RunInstances:1010
Red Hat Enterprise Linux with SQL Server Standard and HA	RunInstances:1014
Red Hat Enterprise Linux with SQL Server Enterprise and HA	RunInstances:1110
Red Hat Enterprise Linux with SQL Server Standard	RunInstances:0014
Red Hat Enterprise Linux with SQL Server Web	RunInstances:0210
Red Hat Enterprise Linux with SQL Server Enterprise	RunInstances:0110
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Ubuntu Pro	RunInstances:0g00
Windows	RunInstances:0002

平台詳細資料	使用操作 [2]
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise [1]	RunInstances:0102
Windows with SQL Server Standard [1]	RunInstances:0006
Windows with SQL Server Web [1]	RunInstances:0202

[1] 如果有兩個軟體授權與 AMI 相關聯，平台詳細資料欄位會顯示兩者。

[2] 如果您執行 Spot 執行個體， AWS 成本和用量報告lineitem/Operation上的 可能與此處列出的用量操作值不同。例如，如果 `lineitem/Operation` 顯示 `RunInstances:0010:SV006`，表示 Amazon EC2 在 Zone 6 的美國東部 (維吉尼亞北部) 正在執行 Red Hat Enterprise Linux Spot 執行個體小時。

[3] 這會在您的用量報告顯示為 RunInstances (Linux/UNIX)。

尋找 AMI 帳單和用量詳細資訊

下列屬性可協助您驗證帳單上的 AMI 費用：

- 平台詳細資料
- 使用操作
- AMI ID

Console

如需針對 AMI 尋找 AMI 帳單資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤上，尋找平台詳細資訊與使用操作。

如需針對執行個體尋找 AMI 帳單資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 在詳細資訊索引標籤上，展開執行個體詳細資訊，然後尋找平台詳細資訊與使用操作。

AWS CLI

如需針對 AMI 尋找 AMI 帳單資訊

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

下列為 Linux AMI 的範例輸出。

```
[
    {
        "PlatformDetails": "Linux/UNIX",
        "UsageOperation": "RunInstances"
    }
]
```

如需針對執行個體尋找 AMI 帳單資訊

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

下列為 Windows 執行個體的範例輸出。

```
[
    {
```

```
        "PlatformDetails": "Windows",
        "UsageOperation": "RunInstances:0002"
    }
]
```

PowerShell

如需針對 AMI 尋找 AMI 帳單資訊

使用 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Format-List PlatformDetails, UsageOperation
```

下列為 Linux AMI 的範例輸出。

```
PlatformDetails : Linux/UNIX
UsageOperation  : RunInstances
```

如需針對執行個體尋找 AMI 帳單資訊

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | `
    Format-List PlatformDetails, UsageOperation
```

下列為 Windows 執行個體的範例輸出。

```
PlatformDetails : Windows
UsageOperation  : RunInstances:0002
```

驗證帳單上的 AMI 費用

為了確保不會產生意外成本，您可以驗證 AWS 成本和用量報告 (CUR) 中執行個體的帳單資訊是否與您用來啟動執行個體的 AMI 相關聯的帳單資訊相符。

若要確認帳單資訊，請在 CUR 中尋找執行個體 ID，然後檢查 `lineitem/Operation` 欄中對應的值。該值應該符合 AMI 相關聯使用操作 的值。

例如，AMI ami-0123456789EXAMPLE 具有下列帳單資訊：

- 平台詳細資訊 = Red Hat Enterprise Linux
- 使用操作 = RunInstances:0010

如果使用此 AMI 啟動執行個體，可在 CUR 中找到該執行個體 ID，然後檢查 lineitem/Operation 欄中的相應值。在此範例中，此值應為 RunInstances:0010。

Amazon EC2 的 AMI 配額

下列配額適用於建立及共用 AMI。這些配額適用於每個 AWS 區域。

配額名稱	Description	每個區域的預設配額
AMI	每個區域允許的公有和私有 AMI 的最大數量。其中包括可用、待定和已停用的 AMI，以及資源回收筒中的 AMI。	50,000
公有 AMI	每個區域允許的公有 AMI 數目上限，包括資源回收筒中的公有 AMI。	5
AMI 共用	可在區域中與 AMI 共用的實體 (組織、組織單位 (OU) 和帳戶) 的最大數目。請注意，如果您與組織或 OU 共用 AMI，則組織或 OU 中的帳戶數目不會計入配額中。	1,000

如果超出配額且想要建立或共用更多 AMI，則您可以執行以下操作：

- 如果超過總 AMI 配額或公有 AMI 配額，則請考慮取消註冊未使用的映像。
- 如果超過公有 AMI 配額，則請考慮將一個或多個公有 AMI 設為私有。
- 如果超過 AMI 共用配額，則請考慮與組織或 OU 共用您的 AMI，而不是與個別帳戶共用。
- 請求增加 AMI 的配額。

請求增加 AMI 的配額

如果您需要超過 AMI 的預設配額，則可以請求增加配額。

若要請求增加 AMI 的配額

1. 開啟 Service Quotas 主控台，網址為 https://console.aws.amazon.com/servicequotas/。
2. 在導覽窗格中，選擇 AWS services (AWS 服務)。
3. 從清單中選擇 Amazon Elastic Compute Cloud (Amazon EC2)，或在搜尋方塊中輸入服務名稱。
4. 選擇 AMI 配額以請求增加。您可以選取的 AMI 配額為：
   - AMI
   - 公有 AMI
   - AMI 共用
5. 選擇 Request quota increase (請求增加配額)。
6. 對於 Change quota value (變更配額值)，輸入新的配額值，然後選擇 Request (請求)。

若要檢視任何擱置或最近解決的請求，請從瀏覽窗格中選擇 Dashboard (儀表板)。對於擱置的請求，請選擇請求狀態以開啟請求回條。請求的初始狀態為 Pending (待定)。狀態變更為 Quota requested (已請求配額) 後，您會在 Support Center case number (支援中心案例編號) 中看到案例編號。選擇案例編號，為請求開啟票證。

解決請求之後，配額的 Applied quota value (套用的配額值) 會設定為新值。

如需詳細資訊，請參閱「Service Quotas 使用者指南」。

Amazon EC2 執行個體

Amazon EC2 執行個體是 AWS 雲端環境中的虛擬伺服器。從您第一次啟動執行個體 (稱為啟動執行個體) 到您刪除執行個體 (稱為終止執行個體) 為止，您可以完全控制執行個體。啟動執行個體時，您可以從各種作業系統中進行選擇。您可以連線到您的執行個體並進行自訂，以符合您的需求。例如，您可以設定作業系統、安裝作業系統更新，以及在執行個體上安裝應用程式。

Amazon EC2 提供廣泛的執行個體類型。您可以選擇執行個體類型，其提供執行應用程式所需的運算資源、記憶體、儲存體和網路效能。

使用 Amazon EC2 時，只需按實際用量付費。在您啟動執行個體時會開始向執行個體計費，且其會轉換為執行中狀態。當您停止執行個體時即停止計費，並在您啟動執行個體時繼續計費。當您終止執行個體時，會在轉換為關閉狀態時停止計費。

Amazon EC2 提供各項功能，可讓您用來最佳化執行個體效能和成本。例如，您可以使用 Amazon EC2 Fleet 或 Amazon EC2 Auto Scaling，隨著執行個體使用率變更而擴展或縮減容量。您可以使用 Spot 執行個體或 Savings Plans 來降低執行個體的成本。

受管執行個體由服務供應商管理，例如 Amazon EKS Auto Mode。您無法直接修改受管執行個體的設定。受管執行個體由受管欄位中的 true 值識別。如需詳細資訊，請參閱Amazon EC2 受管執行個體。

功能和任務

- Amazon EC2 執行個體類型
- Amazon EC2 受管執行個體
- 使用巢狀虛擬化在 Amazon EC2 執行個體中執行 Hypervisor
- Amazon EC2 帳單與購買選項
- 將執行個體啟動參數存放在 Amazon EC2 啟動範本中
- 啟動 Amazon EC2 執行個體
- 連線至 EC2 執行個體
- Amazon EC2 執行個體狀態變更
- 自動執行個體復原功能
- 使用執行個體中繼資料管理您的 EC2 執行個體
- 偵測主機是否為 EC2 執行個體
- Amazon EC2 執行個體的執行個體身分文件
- EC2 執行個體的 STIG 合規

- EC2 執行個體上的精確時鐘和時間同步
- EC2 容量管理器
- 管理 EC2 執行個體的裝置驅動程式
- 設定您的 Amazon EC2 Windows 執行個體
- 將 EC2 Windows 執行個體升級至較新版本的 Windows Server
- 教學課程：將 Amazon EC2 執行個體連線至 Amazon RDS 資料庫

Amazon EC2 執行個體類型

啟動執行個體時，您指定的執行個體類型將決定執行個體所用的主機電腦硬體。每個執行個體類型均提供不同的運算、記憶體和儲存功能，且會依照這些功能分組為不同的執行個體系列。請根據您計劃在執行個體上執行之應用程式或軟體的需求選取執行個體類型。如需功能和使用案例的詳細資訊，請參閱 Amazon EC2 執行個體類型。

Amazon EC2 會將主機電腦的部分資源 (例如 CPU、記憶體和執行個體儲存體) 專用於特定執行個體。Amazon EC2 在執行個體間共用主機電腦的其他資源，例如網路和磁碟子系統。如果主機電腦上的每個執行個體嘗試對其中一項共用資源作最大的利用，該資源會平均分配給每個執行個體。但是，當某一資源使用率較低時，執行個體便能使用該資源較高的比例 (需有資源可用)。

每個執行個體類型從共同資源獲得的最低效能可能更高或更低。例如，具有高 I/O 效能的執行個體類型會分配到較多的共用資源。分配較多的共用資源，也可減少 I/O 效能的變動。對多數的應用程式來說，中等的 I/O 效能即堪使用。但是，對於需要更高或更一致 I/O 效能的應用程式，請考慮使用具有較高 I/O 效能的執行個體類型。

內容

- 可用的執行個體類型
- 硬體規格
- Hypervisor 類型
- AMI 虛擬化類型
- 處理器
- 尋找 Amazon EC2 執行個體類型
- 從 EC2 執行個體類型搜尋工具取得建議
- 從 Compute Optimizer 取得 EC2 執行個體建議
- Amazon EC2 執行個體類型變更

- 爆量效能執行個體
- 使用 GPU 執行個體加速效能
- Amazon EC2 Mac 執行個體
- Amazon EBS 最佳化的執行個體類型
- Amazon EC2 執行個體的 CPU 選項
- Amazon EC2 執行個體的 AMD SEV-SNP
- Amazon EC2 Linux 執行個體的處理器狀態控制

可用的執行個體類型

Amazon EC2 提供各式各樣的最佳化執行個體類型，以滿足不同的使用案例。執行個體類型包含 CPU、記憶體、儲存體和聯網功能的各種組合，供您靈活選擇適用於應用程式的適當資源組合。每個執行個體類型都包含一個或多個執行個體大小，可讓您根據目標工作負載的需求擴展資源。

執行個體類型命名慣例

名稱是根據執行個體系列、世代、處理器系列、功能和大小來命名。如需詳細資訊，請參閱「Amazon EC2 執行個體類型指南」中的命名慣例。

尋找執行個體類型

若要判斷哪些執行個體類型符合您的需求，例如支援的區域、運算資源或儲存資源，請參閱「Amazon EC2 執行個體類型指南」中的 尋找 Amazon EC2 執行個體類型 和 Amazon EC2 執行個體類型規格。

硬體規格

如需執行個體類型的詳細規格，請參閱「Amazon EC2 執行個體類型指南」中的規格。如需定價的詳細資訊，請參閱 Amazon EC2 隨需定價。

為了決定最符合您需求的執行個體類型，我們建議您啟動執行個體並使用您自己的基準參考應用程式。由於您必須為執行個體按秒數付費，因此在作決定前先測試多個執行個體類型是比較方便且省錢的方式。假如您在作決定後需要變更，也可稍後再變更執行個體類型。如需詳細資訊，請參閱Amazon EC2 執行個體類型變更。

Hypervisor 類型

Amazon EC2 支援下列 Hypervisor：Xen 和 Nitro。

Nitro 型執行個體

- 一般用途：M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8a | M8azn | M8g | M8gb | M8gd | M8gn | M8i | M8id | M8i-flex | M8in | M8idn | M8ine | M8ib | M8idb | M9g | M9gd | T3 | T3a | T4g
- 運算最佳化：C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8a | C8g | C8gb | C8gd | C8gn | C8i | C8id | C8i-flex | C8in | C8ine | C8ib | C9g | C9gd
- 記憶體最佳化：R5 | R5a | R5ad | R5b | R5d | R5dn R5n | R6a | R6g | R6gd | R6i | R6id | R6idn | R6in | R7a | R7g | R7gd | R7i | R7iz | R8a | R8g | R8gb | R8gd | R8gn | R8i | R8id | R8i-flex | R8in | R8idn | R8ib | R8idb | R9g | R9gd | U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1tb1 | U-24tb1 U7i-6tb U7i-8tb U7i-12tb U7in-16tb U7in-24tb U7in-32tb U7inh-32tb X2gd X2idn X2iedn X2iezn X8g X8aedz X8i
- 儲存最佳化：D3 | D3en | I3en | I4g | I4i | I7i | I7ie | I8g | I8ge | Im4gn | Is4gen
- 加速運算：DL1 | DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | G6f | Gr6 | Gr6f | G7 | G7e | Inf1 | Inf2 | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6-B300 | P6e-GB200 | Trn1 | Trn1n | Trn2 | Trn2 | Trn2u | VT1
- 高效能運算：Hpc6a | Hpc6id | Hpc7a | Hpc7g | Hpc8a
- 上一代：A1 | P3dn

如需 Nitro Hypervisor 支援版本的詳細資訊，請參閱「Amazon EC2 執行個體類型指南」中的網路功能支援。

Xen 型執行個體

- 一般用途：M1 | M2 | M3 | M4 | T1 | T2
- 運算最佳化：C1 | C3 | C4
- 記憶體最佳化：R3 | R4 | X1 | X1e
- 儲存最佳化：D2 | H1 | I2 | I3
- 加速運算：G3 | P3

AMI 虛擬化類型

執行個體的虛擬化類型由用來啟動執行個體的 AMI 決定。最新一代的執行個體類型僅支援硬體虛擬機器 (HVM)。有些上一代執行個體類型支援全虛擬化 (PV)，有些 AWS 區域則支援 PV 執行個體。如需詳細資訊，請參閱虛擬化類型。

為獲得最高效能，我們建議您使用 HVM AMI。此外，需要 HVM AMI 才能充分利用增強型聯網。HVM 虛擬化使用 AWS 平台提供的硬體協助技術。透過 HVM 虛擬化，訪客 VM 便能如同在原生硬體平台上一般執行，只是仍是使用全虛擬化 (PV) 網路和儲存驅動程式來提升效能。

處理器

EC2 執行個體支援各種處理器。

處理器

- Intel 處理器
- AMD 處理器
- AWS Graviton 處理器
- AWS Trainium
- AWS Inferentia

Intel 處理器

在 Intel 處理器上執行的 Amazon EC2 執行個體可能包含下列處理器功能。並非所有在 Intel 處理器上執行的執行個體都支援這些處理器功能。如需有關每種執行個體類型可用哪些功能的詳細資訊，請參閱 Amazon EC2 執行個體類型。

- Intel AES New Instructions (AES-NI) – Intel AES-NI 加密指令集改進了原先 Advanced Encryption Standard (AES) 的演算法，可以提供更快的資料保護和更佳的安全性。所有最新一代的 EC2 執行個體都支援此處理器功能。
- Intel Advanced Vector Extensions (Intel AVX、Intel AVX2 和 Intel AVX-512) – Intel AVX 和 Intel AVX2 是 256 位元，而 Intel AVX-512 是 512 位元指令集延伸，專為浮點 (FP) 密集型應用程式所設計。Intel AVX instructions 可提升如圖片和音效/影片處理、科學模擬、財務分析、3D 模組化與分析等應用程式的效能。只有以 HVM AMI 啟動的執行個體可使用這些功能。
- Intel Turbo Boost Technology (Intel 渦輪加速技術) – Intel Turbo Boost Technology (Intel 渦輪加速技術) 處理器會自動執行核心，比基礎操作頻率更快。

- Intel Deep Learning Boost (Intel DL Boost) – 可加速 AI 深度學習使用案例。第二代 Intel Xeon 可擴充處理器透過新的向量神經網路指令 (VNNI/INT8) 擴展 Intel AVX-512，相較於前一代 Intel Xeon 可擴充處理器 (搭配 FP32)，大幅提升了影像辨識 / 切割、物件偵測、語音辨識、語言翻譯、推薦系統、強化學習等的深度學習推斷效能。VNNI 可能無法與所有 Linux 發行版相容。

  下列執行個體支援 VNNI：`M5n`、`R5n`、`M5dn`、`M5zn`、`R5b`、`R5dn`、`D3`、`D3en` 和 `C6i`。`C5` 和 `C5d` 執行個體僅針對 `12xlarge`、`24xlarge` 以及 `metal` 執行個體支援 VNNI。

業界對 64 位元 CPU 的命名慣例會導致混淆。自晶片製造商 Advanced Micro Devices (AMD) 成功推出首款以 Intel x86 指令集為基礎的商用 64 位元架構後，該架構自此便廣泛統稱為 AMD64，無論產品來自哪一間晶片製造商。Windows 和多個 Linux 發行版本也遵照此作法。這說明了為何執行 Ubuntu 或 Windows 之執行個體上的內部系統資訊會顯示 CPU 架構為 AMD64，即使該執行個體是在 Intel 硬體上執行。

AMD 處理器

在 AMD EPYC 處理器上執行的 Amazon EC2 執行個體可協助您最佳化工作負載的成本和效能。這些執行個體可能支援下列處理器功能。並非所有在 AMD 處理器上執行的執行個體都支援所有這些處理器功能。如需有關每種執行個體類型可用哪些功能的詳細資訊，請參閱 Amazon EC2 執行個體類型。

- AMD Secure Memory Encryption (SME)
- AMD Transparent Single Key Memory Encryption (TSME)
- AMD Advanced Vector Extensions (AVX)
- AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP)。
- 向量神經網路指令 (VNNI)
- BFloat16

AWS Graviton 處理器

AWS Graviton 是一系列處理器，旨在為在 Amazon EC2 執行個體上執行的工作負載提供最佳價格效能。

如需詳細資訊，請參閱開始使用 Graviton。

AWS Trainium

由 AWS Trainium 支援的執行個體專為提供高效能、符合成本效益的深度學習訓練而建置。您可以使用這些執行個體來訓練自然語言處理、電腦視覺以及廣泛應用程式所使用的推薦模型，例如語音辨識、建

議、詐騙偵測以及影像和影片分類。在常見的機器學習架構 (例如 PyTorch 和 TensorFlow) 中使用您現有的工作流程。

AWS Inferentia

採用 AWS Inferentia 技術的執行個體專為加速機器學習而設計。其提供高效能和低延遲的機器學習推斷。這些執行個體已進行最佳化，適用於部署不同應用的深度學習 (DL) 模型，例如自然語言處理、物件偵測與分類、內容個人化與篩選，以及語音辨識等應用。

您可以透過多種方式開始使用：

- 使用 SageMaker AI，這是一個完整受管理的服務，是開始使用機器學習模型的最簡單方式。如需詳細資訊，請參閱「Amazon SageMaker AI 開發人員指南」中的開始使用 SageMaker AI。
- 使用深度學習 AMI 啟動 Inf1 或 Inf2 執行個體。如需詳細資訊，請參閱 AWS 深度學習 AMIs 開發人員指南中的具有 DLAMI 的AWS Inferentia。
- 使用您自己的 AMI 啟動 Inf1 或 Inf2 執行個體，並安裝 AWS Neuron SDK，以便能夠編譯、執行和分析 AWS Inferentia 的深度學習模型。
- 使用 Inf1 或 Inf2 執行個體和最佳化 Amazon ECS 的 AMI 啟動容器執行個體。若要取得更多資訊，請參閱 Amazon Elastic Container Service Developer Guide中的 Amazon Linux 2 (Inferentia) AMI。
- 建立具有執行 Inf1 執行個體之節點的 Amazon EKS 叢集。如需詳細資訊，請參閱 Amazon EKS 使用者指南中的 Inferentia 支援。

尋找 Amazon EC2 執行個體類型

您必須先選取要使用的執行個體類型，才能啟動執行個體。您選擇的執行個體類型可能取決於您的工作負載所需的資源，如運算、記憶體或儲存資源。識別數種可能適合您的工作負載的執行個體類型，並評估其在測試環境中的效能，可能是有利的做法。沒有任何替代方法可以測量您的應用程式在負載條件下的效能。

您可以使用 EC2 執行個體類型搜尋工具取得 EC2 執行個體類型的建議和指導。如需詳細資訊，請參閱the section called “EC2 執行個體類型搜尋工具”。

如果您已經有執行中的 EC2 執行個體，您可以使用 AWS Compute Optimizer 取得有關應該用來改善效能、節省成本或兩者之執行個體類型的建議。如需詳細資訊，請參閱the section called “Compute Optimizer 建議”。

任務

- 使用主控台尋找執行個體類型
- 使用 描述執行個體類型 AWS CLI
- 使用 尋找執行個體類型 AWS CLI
- 使用 Tools for PowerShell 尋找執行個體類型

使用主控台尋找執行個體類型

您可以使用 Amazon EC2 主控台來尋找符合您需求的執行個體類型。

使用主控台尋找執行個體類型

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽列選取要在其中啟動執行個體的區域。無論您的位置為何，皆可選取任何可用的區域。
3. 在導覽窗格中，選擇 Instance Types (執行個體類型)。
4. (選用) 選取偏好設定 (齒輪) 圖示來選取要顯示的執行個體類型屬性，例如隨需 Linux 定價，然後選取 Confirm (確認)。或者，選取執行個體類型的名稱，以開啟其詳細資訊頁面，並透過主控台檢視可用的所有屬性。主控台不會顯示透過 API 或命令列可用的所有屬性。
5. 使用執行個體類型屬性，將顯示的執行個體類型清單篩選為僅符合您需求的執行個體類型。例如，您可以篩選下列屬性：

   - Availability zones (可用區域) – 可用區域、本機區域或 Wavelength 區域的名稱。如需詳細資訊，請參閱 the section called “區域 (Region) 和區域 (Zone)”。
   - vCPUs 或 Cores (核心) – vCPU 或核心的數量。
   - Memory (GiB) (記憶體 (GiB)) – 記憶體大小 (以 GiB 為單位)。
   - Network performance (網路效能) – 網路效能 (以 GB 為單位)。
   - Local instance storage (本機執行個體儲存體) – 表示執行個體類型是否具有本機執行個體儲存體 (`true` | `false`)。
6. (選用) 若要查看並排比較結果，請選取多個執行個體類型的核取方塊。比較結果顯示在螢幕畫面的底部。
7. (選用) 若要將執行個體類型清單儲存為逗號分隔值 (.csv) 檔案以供進一步檢閱，請選擇 Actions (動作)、Download list CSV (下載清單 CSV)。檔案包括符合您所設定篩選的所有執行個體類型。
8. (選用) 若要使用滿足您需求的執行個體類型來啟動執行個體，請選取執行個體類型的核取方塊，然後選取 Actions (動作)、Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

使用 描述執行個體類型 AWS CLI

您可以使用 describe-instance-types 命令來說明特定執行個體類型。

完整說明執行個體類型

下列命令顯示指定執行個體類型的所有可用詳細資訊。輸出很冗長，因此在此處省略。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2
```

說明執行個體類型並篩選輸出

下列命令會顯示指定執行個體類型的網路詳細資訊。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].NetworkInfo"
```

以下為範例輸出。

```
[
    {
        "NetworkPerformance": "Low to Moderate",
        "MaximumNetworkInterfaces": 2,
        "MaximumNetworkCards": 1,
        "DefaultNetworkCardIndex": 0,
        "NetworkCards": [
            {
                "NetworkCardIndex": 0,
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "BaselineBandwidthInGbps": 0.064,
                "PeakBandwidthInGbps": 1.024
            }
        ],
        "Ipv4AddressesPerInterface": 2,
        "Ipv6AddressesPerInterface": 2,
        "Ipv6Supported": true,
        "EnaSupport": "unsupported",
```

```
        "EfaSupported": false,
        "EncryptionInTransitSupported": false,
        "EnaSrdSupported": false
    }
]
```

下列命令會顯示指定執行個體類型的可用記憶體。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].MemoryInfo"
```

以下為範例輸出。

```
[
    {
        "SizeInMiB": 1024
    }
]
```

使用 尋找執行個體類型 AWS CLI

您可以使用 describe-instance-types 和 describe-instance-type-offerings 命令來尋找符合您需求的執行個體類型。

範例

- 範例 1：依可用區域尋找執行個體類型
- 範例 2：依可用的記憶體大小尋找執行個體類型
- 範例 3：依可用的執行個體儲存體尋找執行個體類型
- 範例 4：尋找支援休眠的執行個體類型

範例 1：依可用區域尋找執行個體類型

下列範例僅顯示指定的可用區域中所提供的執行個體類型。

```
aws ec2 describe-instance-type-offerings \
    --location-type "availability-zone" \
```

```
    --filters "Name=location,Values=us-east-2a" \
    --region us-east-2 \
    --query "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

輸出是執行個體類型的清單，依字母順序排序。以下僅為輸出的開始：

```
a1.2xlarge
a1.4xlarge
a1.large
a1.medium
a1.metal
a1.xlarge
c4.2xlarge
   ...
```

範例 2：依可用的記憶體大小尋找執行個體類型

下列範例僅顯示包含 64GiB (65536MiB) 記憶體的目前世代執行個體類型。

```
aws ec2 describe-instance-types \
    --filters "Name=current-generation,Values=true" "Name=memory-info.size-in-
mib,Values=65536" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

輸出是執行個體類型的清單，依字母順序排序。以下僅為輸出的開始：

```
c5a.8xlarge
c5ad.8xlarge
c6a.8xlarge
c6g.8xlarge
c6gd.8xlarge
c6gn.8xlarge
c6i.8xlarge
c6id.8xlarge
c6in.8xlarge
   ...
```

範例 3：依可用的執行個體儲存體尋找執行個體類型

下列範例顯示具有執行個體儲存體磁碟區所有 R7 執行個體的執行個體儲存體總大小。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r7*" "Name=instance-storage-
supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
    --output table
```

以下為範例輸出。

```
----------------------------
|  DescribeInstanceTypes   |
+----------------+---------+
|  r7gd.xlarge   |  237    |
|  r7gd.8xlarge  |  1900   |
|  r7gd.16xlarge |  3800   |
|  r7gd.medium   |  59     |
|  r7gd.4xlarge  |  950    |
|  r7gd.2xlarge  |  474    |
|  r7gd.metal    |  3800   |
|  r7gd.large    |  118    |
|  r7gd.12xlarge |  2850   |
+----------------+---------+
```

範例 4：尋找支援休眠的執行個體類型

下列範例顯示支援休眠的執行個體類型。

```
aws ec2 describe-instance-types \
    --filters "Name=hibernation-supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

輸出是執行個體類型的清單，依字母順序排序。以下僅為輸出的開始：

```
c4.2xlarge
c4.4xlarge
c4.8xlarge
c4.large
c4.xlarge
c5.12xlarge
```

```
c5.18xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
...
```

使用 Tools for PowerShell 尋找執行個體類型

您可使用 Get-EC2InstanceType 與 Get-EC2InstanceTypeOffering cmdlet 來尋找符合您需求的執行個體類型。

範例

- 依可用區域尋找執行個體類型
- 依可用的記憶體大小尋找執行個體類型
- 依可用的執行個體儲存體尋找執行個體類型
- 尋找支援休眠的執行個體類型

依可用區域尋找執行個體類型

下列範例僅顯示指定的可用區域中所提供的執行個體類型。

```
(Get-EC2InstanceTypeOffering `
    -LocationType "availability-zone" `
    -Region us-east-2 `
    -Filter @{Name="location"; Values="us-east-2a"}).InstanceType | Sort-Object `
```

依可用的記憶體大小尋找執行個體類型

下列範例僅顯示包含 64GiB (65536MiB) 記憶體的目前世代執行個體類型。

```
(Get-EC2InstanceType `
    -Filter @{Name="current-generation"; Values="true"},
            @{Name="memory-info.size-in-mib"; Values="65536"}).InstanceType | Sort-
Object
```

依可用的執行個體儲存體尋找執行個體類型

下列範例顯示具有執行個體儲存體磁碟區所有 R7 執行個體的執行個體儲存體總大小。

```
Get-EC2InstanceType `
    -Filter @{Name="instance-type"; Values="r7*"},
            @{Name="instance-storage-supported"; Values="true"} | `
     Select InstanceType, @{Name="TotalSizeInGB";
 Expression={($_.InstanceStorageInfo.TotalSizeInGB)}}
```

以下為範例輸出。

```
InstanceType  TotalSizeInGB
------------  -------------
r7gd.8xlarge           1900
r7gd.16xlarge          3800
r7gd.xlarge             237
r7gd.4xlarge            950
r7gd.medium              59
r7gd.2xlarge            474
r7gd.large              118
r7gd.metal             3800
r7gd.12xlarge          2850
```

尋找支援休眠的執行個體類型

下列範例顯示支援休眠的執行個體類型。

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-Object
```

從 EC2 執行個體類型搜尋工具取得建議

EC2 執行個體類型搜尋工具會考量您的使用案例、工作負載類型、CPU 製造商偏好設定、排定價格和效能優先順序的方式，以及您可以指定的其他參數。然後，它會使用這些資料，針對您的新工作負載，提供最適合的 Amazon EC2 執行個體類型建議和指引。

由於有太多可用的執行個體類型，為您的工作負載尋找適當的執行個體類型可能既耗時又複雜。透過使用 EC2 執行個體類型搜尋工具，您可以保持最新的執行個體類型，並為工作負載達到最佳的價格效能。

您可以使用 Amazon EC2 主控台取得 EC2 執行個體類型的建議和指引。您也可以直接前往 Amazon Q 尋求執行個體類型建議。如需詳細資訊，請參閱 Amazon Q Developer 使用者指南。

如果您正在尋找適用於現有工作負載的執行個體類型建議，請使用 AWS Compute Optimizer. 如需詳細資訊，請參閱從 Compute Optimizer 取得 EC2 執行個體建議。

使用 EC2 執行個體類型搜尋工具

在 Amazon EC2 主控台中，您可以在啟動執行個體精靈中、建立啟動範本時或在執行個體類型頁面上，從 EC2 執行個體類型搜尋工具取得執行個體類型建議。

依照下列說明，在 Amazon EC2 主控台中使用 EC2 執行個體類型搜尋工具取得 EC2 執行個體類型的建議和指引。若要檢視步驟動畫，請參閱 檢視動畫：使用 EC2 執行個體類型搜尋工具，取得執行個體類型的建議。

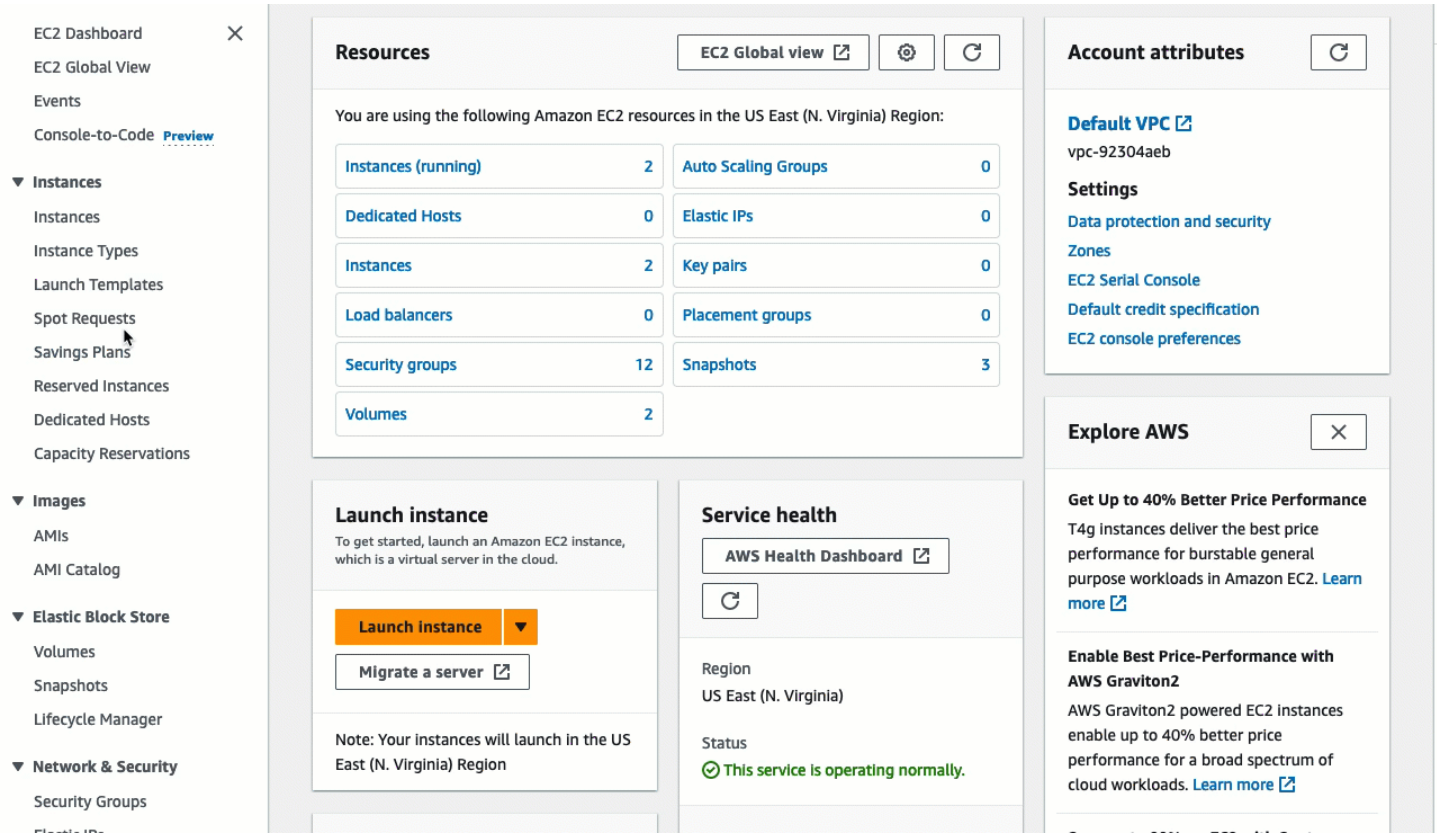
使用 EC2 執行個體類型搜尋工具取得執行個體類型建議

1. 使用下列任何一項開始您的流程：
   - 請遵循該程序來啟動執行個體。在執行個體類型旁，選擇取得建議連結。
   - 請依照程序建立啟動範本。在執行個體類型旁，選擇取得建議連結。
   - 在導覽窗格中，依序選擇執行個體類型和執行個體類型搜尋工具按鈕。
2. 在取得執行個體類型選擇建議畫面中，執行下列動作：
   a. 使用工作負載類型、使用案例、優先順序和 CPU 製造商等選項，指定您的執行個體類型。
   b. (選用) 若要為您的工作負載指定更詳細的需求，請執行下列動作：
      i. 展開進階參數。
      ii. 若要新增參數，請選取參數並選擇新增，然後指定該參數的值。針對每個您要新增的其他參數重複此步驟。若要指示沒有最小值或最大值，請將欄位留空。
      iii. 若要移除剛新增的參數，請選擇參數旁的 X。
   c. 選擇取得執行個體類型建議。

      Amazon EC2 為您提供符合指定需求的執行個體系列建議。
3. 若要檢視建議執行個體系列中每個執行個體類型的詳細資訊，請選擇檢視建議的執行個體系列詳細資訊。
4. 選取符合您需求的執行個體類型，然後選擇動作、啟動執行個體或動作、建立啟動範本。

   或者，如果您開始了啟動執行個體精靈或啟動範本頁面中的流程，且偏好返回原始流程，請記下要使用的執行個體類型。然後，在啟動執行個體精靈或啟動範本中，針對執行個體類型選擇執行個體類型，並完成啟動執行個體或建立啟動範本的程序。

檢視動畫：使用 EC2 執行個體類型搜尋工具，取得執行個體類型的建議



從 Compute Optimizer 取得 EC2 執行個體建議

AWS Compute Optimizer 提供 Amazon EC2 建議，協助您改善效能、節省成本或兩者。您可以根據這些建議來決定是否變更為新的執行個體類型。

若要提出建議，Compute Optimizer 會分析您現有的執行個體規格和使用率指標。然後，編譯的資料會用來建議哪些 Amazon EC2 執行個體類型最適合處理現有的工作負載。建議會連同每小時執行個體定價一起傳回。如需詳細資訊，請參閱「AWS Compute Optimizer 使用者指南」中的 Amazon EC2 執行個體指標。

目錄

- 要求
- 尋找分類
- 檢視建議
- 評估建議的考量事項

要求

若要從 Compute Optimizer 取得建議，您必須先選擇加入 Compute Optimizer。如需詳細資訊，請參閱「 AWS Compute Optimizer使用者指南」中的 AWS Compute Optimizer 入門。

Compute Optimizer 會為某些執行個體類型產生建議，但並非所有執行個體類型。如果您使用的是不受支援的執行個體類型，則 Compute Optimizer 不會產生建議。如需支援執行個體類型清單，請參閱「AWS Compute Optimizer 使用者指南」中的 Amazon EC2 執行個體需求。

尋找分類

Compute Optimizer 會將其 EC2 執行個體的發現結果分類如下：

- Under-provisioned (佈建不足)–當您的執行個體至少有一個規格 (例如 CPU、記憶體或網路) 不符合工作負載的效能需求時，EC2 執行個體即視為佈建不足。佈建不足的 EC2 執行個體可能會導致應用程式效能不佳。
- Over-provisioned (過度佈建)–當執行個體至少一個規格可以縮小 (例如 CPU、記憶體或網路)，同時仍然符合工作負載的效能需求時，以及當沒有規格佈建不足時，EC2 執行個體即視為過度佈建。過度佈建的 EC2 執行個體可能會導致不必要的基礎設施成本。
- Optimized (最佳化)–當執行個體的所有規格 (例如 CPU、記憶體和網路) 都符合工作負載的效能需求，且執行個體未過度佈建時，EC2 執行個體即視為最佳化。最佳化的 EC2 執行個體會以最佳效能和基礎設施成本執行您的工作負載。對於最佳化的執行個體，Compute Optimizer 有時可能會建議新一代執行個體類型。
- None (無)–沒有此執行個體的建議。如果您已在 12 小時內選擇加入 Compute Optimizer，或是執行個體執行時間不到 30 小時，或 Compute Optimizer 不支援執行個體類型，就可能會發生這種情況。

檢視建議

選擇加入 Compute Optimizer 之後，您可以在 Amazon EC2 主控台中檢視 Compute Optimizer 為 EC2 執行個體所產生的調查結果。然後，您可以存取 Compute Optimizer 主控台來檢視建議。如果您最近選擇加入，發現結果可能需要長達 12 小時才會反映在 EC2 主控台中。

透過 Amazon EC2 主控台檢視執行個體的建議

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選擇執行個體 ID 以開啟執行個體詳細資訊頁面。

4. 在執行個體詳細資訊頁面的最上方摘要區段，尋找 AWS Compute Optimizer 調查結果。如果有調查結果，系統會顯示調查結果分類及檢視詳細資訊的連結。否則，系統會顯示沒有適用於此執行個體的建議。
5. 如果有調查結果，請選擇檢視詳細資訊。這會在 Compute Optimizer 主控台開啟 EC2 執行個體的建議頁面。目前的執行個體類型會標記為目前。還有最多三種執行個體類型建議，標示為選項 1、選項 2 和選項 3。此頁面也會顯示執行個體的最新 CloudWatch 指標資料。

檢視所有區域所有執行個體的建議

您可以使用 Compute Optimizer 主控台檢視所有區域所有 Amazon EC2 執行個體的建議。如需詳細資訊，請參閱「AWS Compute Optimizer 使用者指南」中的檢視 EC2 執行個體建議已及檢視 EC2 執行個體詳細資訊。

評估建議的考量事項

當收到建議時，您必須決定是否對其採取行動。變更執行個體類型之前，請考慮下列事項：

- 建議不會預測您的使用量。建議是根據您最近 14 天期間的歷史使用情況而定。請務必選擇符合未來資源需求的執行個體類型。
- 著重於圖形指標，以判斷實際使用量是否低於執行個體容量。您也可以檢視CloudWatch 中的指標資料 (平均、峰值、百分比)，以進一步評估您的 EC2 執行個體建議。例如，注意 CPU 百分比指標在白天的變化，以及是否有需要應付的尖峰時期。如需詳細資訊，請參閱「Amazon CloudWatch 使用者指南」中的 檢視可用指標。
- Compute Optimizer 可能會提供高載效能執行個體的建議，亦即 T3、T3a 和 T2 執行個體。如果您定期爆增超過基準線，請確定您可以根據新執行個體類型的 vCPU 繼續執行此作業。如需詳細資訊，請參閱爆量效能執行個體的重要概念。
- 如果您購買的是預留執行個體，隨需執行個體可能會以預留執行個體方式計費。變更目前的執行個體類型之前，請先評估對預留執行個體使用率和涵蓋範圍的影響。
- 盡可能考慮轉換為較新一代的執行個體。
- 移轉至不同的執行個體系列時，請確定目前的執行個體類型和新的執行個體類型是相容的，例如在虛擬化、架構或網路類型方面。如需詳細資訊，請參閱變更執行個體類型的相容性。
- 最後，請考慮針對每個建議所提供的效能風險評分。效能風險表示您可能需要花費多少精力來驗證建議的執行個體類型是否符合工作負載的效能需求。我們也建議在進行任何變更之前和之後進行嚴格的負載和效能測試。

Amazon EC2 執行個體類型變更

隨著您的需求變更，您可能會發現 Amazon EC2 執行個體對其工作負載而言太大或太小。若出現此情況，您可以透過變更執行個體類型來調整執行個體的大小。例如，如果您的 `t2.micro` 執行個體對於其工作負載來說太小，則您可以透過將其變更為較大的 T2 執行個體類型 (例如 `t2.large`) 來增加其大小。或者，您可以將其變更為另一個執行個體類型，例如 `m5.large`。您可能也會希望將先前世代的執行個體類型變更為目前世代的執行個體類型，以利用某些功能，例如 IPv6 的支援。

如果想要獲得有關最能處理現有工作負載的執行個體類型建議，可以使用 AWS Compute Optimizer。如需詳細資訊，請參閱從 Compute Optimizer 取得 EC2 執行個體建議。

若變更執行個體類型，您將開始支付新執行個體類型的費率。如需所有執行個體類型的隨需費率，請參閱 Amazon EC2 隨需定價。

若要在不變更執行個體類型的情況下為執行個體新增額外儲存空間，請將 EBS 磁碟區新增至執行個體。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的將 Amazon EBS 磁碟區連接到執行個體。

要遵循哪些指示？

變更執行個體類型有不同的指示。要使用的指示取決於執行個體的根磁碟區，以及執行個體類型是否與執行個體的目前組態相容。如需如何判斷相容性的相關資訊，請參閱 變更執行個體類型的相容性。

使用下表判斷要遵循哪些指示。

根磁碟區	相容性	使用這些指示
EBS	相容	變更執行個體類型
EBS	不相容	遷移至新的執行個體類型
執行個體存放區	不適用	遷移至新的執行個體類型

變更執行個體類型的相容性

只在執行個體的目前組態與您想要的執行個體類型相容時，您才能變更執行個體類型。如果您想要的執行個體類型與執行個體的目前狀態不相容，您必須搭配與執行個體類型相容的組態，啟動新的執行個體，然後將您的應用程式遷移到新的執行個體。

下列方式可用於判斷相容性：

虛擬化類型

Linux AMI 會使用兩種虛擬化類型的其中之一：全虛擬化 (PV) 或硬體虛擬機器 (HVM)。如果已從 PV AMI 啟動執行個體，則您無法將其變更為僅限 HVM 的執行個體類型。如需詳細資訊，請參閱虛擬化類型。若要檢查您執行個體的虛擬化類型，請檢查 Amazon EC2 主控台中 Instances (執行個體) 畫面詳細資訊窗格上的 Virtualization (虛擬化) 值。

Architecture

AMI 是處理器特有架構，所以您選取的執行個體類型，必須和目前的執行個體類型有相同的處理器架構。例如：

- 如果目前執行個體類型具有 Arm 架構處理器，您會受限於支援 Arm 架構處理器的執行個體類型，例如 C6g 和 M6g。
- 下列執行個體類型是唯一支援 32 位元 AMIs 的執行個體類型：`t2.nano`、`t2.micro`、`t2.small`、`t2.medium`、`c3.large`、`t1.micro`、`m1.small`、`m1.me` 和 `c1.medium`。如果您要變更 32 位元執行個體的執行個體類型，則會受限於這些執行個體類型。

網路介面卡

如果您從某個網路介面卡的驅動程式切換到另一個網路介面卡，該網路介面卡設定會在作業系統建立新的介面卡時重設。若要重新進行設定，您可能需要存取具有系統管理員許可的本機帳戶。以下是從某個網路介面卡移至另一個網路介面卡的範例：

- AWS PV (T2 執行個體 ） 至 Intel 82599 VF (M4 執行個體 ）
- Intel 82599 VF (大多數 M4 執行個體) 至 ENA (M5 執行個體)
- ENA (M5 執行個體) 至高頻寬 ENA (M5n 執行個體)

增強型網路

支援增強型聯網的執行個體類型需要安裝必要的驅動程式。例如，以 Nitro 為基礎的執行個體需要 EBS 後端 AMI，並且安裝彈性網路介面卡 (ENA) 驅動程式。若要將不支援增強型聯網的執行個體類型變更為支援增強型聯網的執行個體類型，您必須視需要在執行個體上安裝 ENA 驅動程式或 ixgbevf 驅動程式。

> **Note**
>
> 當您在啟用 ENA Express 的情況下調整執行個體大小時，新的執行個體類型也必須支援 ENA Express。如需支援 ENA Express 的執行個體類型清單，請參閱 ENA Express 支援的執行個體類型。

若要將支援 ENA Express 的執行個體類型變更為不支援其的執行個體類型，請確定目前在調整執行個體大小之前，ENA Express 未啟用。

NVMe

EBS 磁碟區會公開為 Nitro 型執行個體上的 NVMe 區塊型儲存設備。如果將不支援 NVMe 的執行個體類型變更為支援 NVMe 的執行個體類型，您必須先在執行個體上安裝 NVMe 驅動程式。此外，您在區塊型儲存設備映射中指定之裝置的裝置名稱也會使用 NVMe 裝置名稱 (`/dev/nvme[0-26]n1`) 重新命名。

[Linux 執行個體] 因此，若要在開機時使用 `/etc/fstab` 來掛載檔案系統，您必須使用 UUID/標籤，而非裝置名稱。

磁碟區限制

可連接到執行個體的 Amazon EBS 磁碟區數目上限，取決於執行個體類型和執行個體大小。如需詳細資訊，請參閱Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。

只有執行個體類型或執行個體大小支援的磁碟區數量，等於或大於目前連接至執行個體的磁碟區數量，您才能變更為這樣的執行個體類型或執行個體大小。如果您變更為不支援目前連接磁碟區數量的執行個體類型或執行個體大小，則該請求會失敗。例如，如果您從具有 32 個連接磁碟區的 `m7i.4xlarge` 執行個體變更為最多支援 27 個磁碟區的 `m6i.4xlarge`，則該請求會失敗。

NitroTPM

如果您使用已啟用 NitroTPM 的 AMI，以及支援 NitroTPM 的執行個體類型來啟動執行個體，即會啟動執行個體並啟用 NitroTPM。您只能變更為同樣支援 NitroTPM 的執行個體類型。

變更 Amazon EC2 執行個體的執行個體類型

如果您需要的執行個體類型與執行個體的目前組態相容，請使用下列指示來變更 Amazon EBS 後端執行個體的執行個體類型。如需詳細資訊，請參閱the section called “相容性”。

考量事項

- 您必須先停止執行個體，才能變更其執行個體類型。請確認您已為執行個體停止時的停機時間做好規劃。停止執行個體並變更其執行個體類型可能會花費數分鐘，重新啟動您的執行個體所需要花費的時間則會根據您應用程式的啟動指令碼而有所不同。如需詳細資訊，請參閱 停止和啟動 Amazon EC2 執行個體。

- 當您停止並啟動執行個體時，我們會將執行個體移至新硬體。如果您的執行個體具有公有 IPv4 地址且不是彈性 IP，我們會釋出位址並向您的執行個體提供新的公有 IPv4 地址。如需有關整個執行個體生命週期中 IP 位址行為的詳細資訊，請參閱 執行個體狀態之間的差異。
- 您無法變更 Spot 執行個體的執行個體類型。
- 【Windows 執行個體】 建議您在變更執行個體類型之前更新 AWS PV 驅動程式套件。如需詳細資訊，請參閱the section called “升級 PV 驅動程式”。
- 如果您的執行個體位於 Auto Scaling 群組中，Amazon EC2 Auto Scaling 服務會將停止的執行個體標記為狀況不良，且可能將其終止並啟動替代執行個體。若要避免這種情況，您可以在變更執行個體類型時，暫停群組的擴展程序。如需詳細資訊，請參閱Amazon EC2 Auto Scaling 使用者指南中的暫停及繼續擴展程序。
- 當您變更具有 NVMe執行個體存放區磁碟區之執行個體的執行個體類型時，更新後的執行個體可能具有其他執行個體存放區磁碟區，因為即使未在 AMI 或執行個體區塊型設備映射中加以指定，所有 NVMe 執行個體存放區磁碟區都可用。否則，更新後的執行個體通常會有與您在啟動原始執行個體時指定之執行個體儲存體磁碟區相同數目的執行個體儲存體磁碟區。
- 可連接到執行個體的 Amazon EBS 磁碟區數目上限，取決於執行個體類型和執行個體大小。如果執行個體類型或執行個體大小不支援已連接至執行個體的磁碟區數量，您就無法變更為這樣的執行個體類型或執行個體大小。如需詳細資訊，請參閱Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。
- [Linux 執行個體] 您可以使用 `AWSSupport-MigrateXenToNitroLinux` 執行手冊，將相容的 Linux 執行個體從 Xen 執行個體類型遷移至 Nitro 執行個體類型。如需詳細資訊，請參閱《AWS Systems Manager Automation 執行手冊參考》中的 AWSSupport-MigrateXenToNitroLinux runbook。
- [Windows 執行個體] 如需將相容 Windows 執行個體從 Xen 執行個體類型遷移至 Nitro 執行個體類型的其他指引，請參閱遷移至最新一代執行個體類型。

變更 Amazon EBS 為後端之執行個體的執行個體類型

1. (選用) 若新執行個體類型需要的驅動程式並未安裝於現有的執行個體，您必須先連接您的執行個體並安裝驅動程式。如需詳細資訊，請參閱變更執行個體類型的相容性。
2. [Windows 執行個體] 如果您將 Windows 執行個體設定為使用靜態 IP 定址，且要將不支援增強型網路的執行個體類型變更為支援增強型網路的執行個體類型，重新設定靜態 IP 位址時可能會收到出現 IP 位址衝突的警告。為避免出現警告，請在執行個體的網路介面上啟用 DHCP，然後再變更執行個體類型。在執行個體中開啟 Network and Sharing Center (網路和共用中心)，開啟網路介面的 Internet Protocol Version 4 (TCP/IPv4) Properties (網際網路通訊協定第 4 版 (TCP/IPv4) 內容)，然後選擇 Obtain an IP address automatically (自動取得 IP 地址)。在網路介面上變更執行個體類型，然後重新設定靜態 IP 地址。

3. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
4. 在導覽窗格中，選擇 Instances (執行個體)。
5. 選取執行個體，並選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。出現確認提示時，請選擇 Stop (停止)。停止執行個體可能需要幾分鐘。
6. 在仍然選取執行個體的情況下，選取 Actions (動作)、Instance settings (執行個體設定)、Change instance type (變更執行個體類型)。如果執行個體狀態不是 `stopped`，此選項會呈現灰色。
7. 在 Change instance type (變更執行個體類型) 頁面上，執行下列動作：

   a. 針對 Instance type (執行個體類型)，選取您想要的執行個體類型。

      如果執行個體類型不在清單中，表示其與您執行個體的組態不相容。請改用下列指示：透過啟動新的 EC2 執行個體，遷移至新的執行個體類型。

   b. (選用) 如果您選取的執行個體類型支援 EBS 最佳化，請選取 EBS-optimized (EBS 最佳化) 以啟用 EBS 最佳化，或取消選取 EBS-optimized (EBS 最佳化) 以停用 EBS 最佳化。

      如果您選取的執行個體類型預設為 EBS 最佳化，則 EBS-optimized (EBS 最佳化) 便會處於選取狀態，而且您無法取消選取。

   c. (選用) 在新的執行個體類型上設定 vCPU 選項。

      當您變更現有執行個體的執行個體類型時，Amazon EC2 會盡可能將現有執行個體的 CPU 選項設定套用至新執行個體。若新的執行個體類型不支援這些設定，則 CPU 選項會重設為無。此選項使用預設 vCPU 數量的新執行個體類型。

      如果您選取的執行個體類型支援 vCPU 組態，請選取進階詳細資訊面板中的指定 CPU 選項，將 vCPU 設定為新的執行個體類型。

   d. 選擇變更以接受新的設定。
8. 若要啟動執行個體，請選取執行個體，然後選取 Instance state (執行個體狀態)、Start instance (啟動執行個體)。執行個體進入 `running` 狀態可能需要數分鐘。若您的執行個體未啟動，請參閱對變更執行個體類型進行疑難排解。
9. [Windows 執行個體] 如果您的執行個體執行配備 EC2Launch v1 的 Windows Server 2016 或 Windows Server 2019，請連線到您的 Windows 執行個體，然後執行下列 EC2Launch PowerShell 指令碼，在變更執行個體類型之後設定執行個體。

> **Important**
>
> 當您啟用初始化執行個體 EC2 Launch 指令碼時，管理員密碼將重設。您可以修改組態檔來停用管理員密碼重設，方法為在初始化任務的設定中指定它。如需如何停用密碼重設的步驟，請參閱設定初始化任務 (EC2Launch) 或變更設定 (EC2Launch v2)。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

透過啟動新的 EC2 執行個體，遷移至新的執行個體類型

只有在 EBS 後端執行個體的組態與您想要的新執行個體類型相容時，您才能變更 EC2 執行個體的執行個體類型。否則，如果組態或您的執行個體與新的執行個體類型不相容，或者該執行個體是以執行個體儲存體為基礎的執行個體，您必須啟動與您想要之執行個體類型相容的替代執行個體。如需有關如何判斷相容性的詳細資訊，請參閱 變更執行個體類型的相容性。

遷移程序概觀

- 備份原始執行個體上的資料。
- 搭配與您想要之新執行個體類型相容的組態，啟動新的執行個體，並連接任何已連接至原始執行個體的 EBS 磁碟區。
- 在新的執行個體上安裝您的應用程式。
- 還原任何資料。
- 如果原始執行個體具有彈性 IP 位址，您必須將其與新執行個體建立關聯，以確保您的使用者可以繼續使用應用程式而不會中斷。

將執行個體遷移至新的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 備份任何仍然需要的資料，如下所示：
   - 連線到您的執行個體，並將執行個體儲存體磁碟區上的資料複製到持久性儲存。
   - 為您的 EBS 磁碟區建立快照，以便建立具有相同資料的新磁碟區，或從原始執行個體分離磁碟區，以便將它們連接到新的執行個體。

3. 在導覽窗格中，選擇執行個體。
4. 選擇啟動執行個體。設定執行個體時，請執行下列動作：
   a. 選取支援您想要之執行個體類型的 AMI。例如，您必須選取支援新執行個體類型之處理器類型的 AMI。此外，目前世代的執行個體類型也需要 HVM AMI。
   b. 選取您想要的新執行個體類型。如果您想要的執行個體類型無法使用，表示該類型與您選取的 AMI 組態不相容。
   c. 如果您想允許相同流量觸及新的執行個體，請選取與原始執行個體搭配使用的安全群組。
   d. 完成設定新的執行個體後，請完成步驟來選取金鑰對並啟動您的執行個體。執行個體進入 `running` 狀態可能需要數分鐘。
5. 如果您將資料備份至 EBS 快照，請從快照建立磁碟區，再將磁碟區連接至新的執行個體。

   若要將原始執行個體的 EBS 磁碟區移至新的執行個體，請將磁碟區與原始執行個體分離，再將磁碟區連接至新的執行個體。
6. 在新的執行個體上安裝您的應用程式及任何必要的軟體。
7. 還原您從原始執行個體的執行個體儲存體磁碟區備份的任何資料。
8. 如果原始執行個體有彈性 IP 位址，將其指派給新的執行個體，如下所示：
   a. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
   b. 選取與原始執行個體相關聯的彈性 IP 地址，選取 Actions (動作)，然後選取 Disassociate Elastic IP address (與彈性 IP 地址取消關聯)。出現確認提示時，請選擇 Disassociate (取消關聯)。
   c. 在彈性 IP 地址仍在選取狀態的情況下，選取 Actions (動作)，然後選取 Associate Elastic IP address (與彈性 IP 地址建立關聯)。
   d. 對於 Resource type (資源類型)，選擇 Instance (執行個體)。
   e. 在執行個體中，選擇新的執行個體。
   f. (選用) 對於 Private IP address (私人 IP 位址)，指定要與彈性 IP 位址建立關聯的私人 IP 位址。
   g. 選擇 Associate (關聯)。
9. (選用) 若您不再需要原始執行個體，則可將其終止。選取執行個體，確認您要終止的是原始執行個體，而非新的執行個體 (例如，檢查名稱或啟動時間)，然後選取 Instance state (執行個體狀態)、Terminate instance (終止執行個體)。

對變更執行個體類型進行疑難排解

請使用以下資訊來協助您診斷並修正在變更執行個體類型時可能遇到的問題。

在變更執行個體類型之後執行個體不會啟動

可能原因：未符合新執行個體類型的需求

如果您的執行個體未啟動，則原因可能是未符合新執行個體類型的其中一個需求。如需詳細資訊，請參閱為何我的 Linux 執行個體在我變更其類型之後未啟動？

可能原因：AMI 不支援執行個體類型

如果您使用 EC2 主控台變更執行個體類型，則只能使用所選 AMI 支援的執行個體類型。不過，如果您使用 AWS CLI 啟動執行個體，您可以指定不相容的 AMI 和執行個體類型。如果 AMI 與執行個體類型不相容，則執行個體無法啟動。如需詳細資訊，請參閱 變更執行個體類型的相容性。

可能原因：執行個體位於集群放置群組中

如果您的執行個體位於集群放置群組中，並在變更執行個體類型之後，執行個體無法啟動，請嘗試下列動作：

1. 停止集群放置群組中的所有執行個體。
2. 變更受影響執行個體的執行個體類型。
3. 啟動集群放置群組中的所有執行個體。

在變更執行個體類型之後，無法從網際網路連線到應用程式或網站

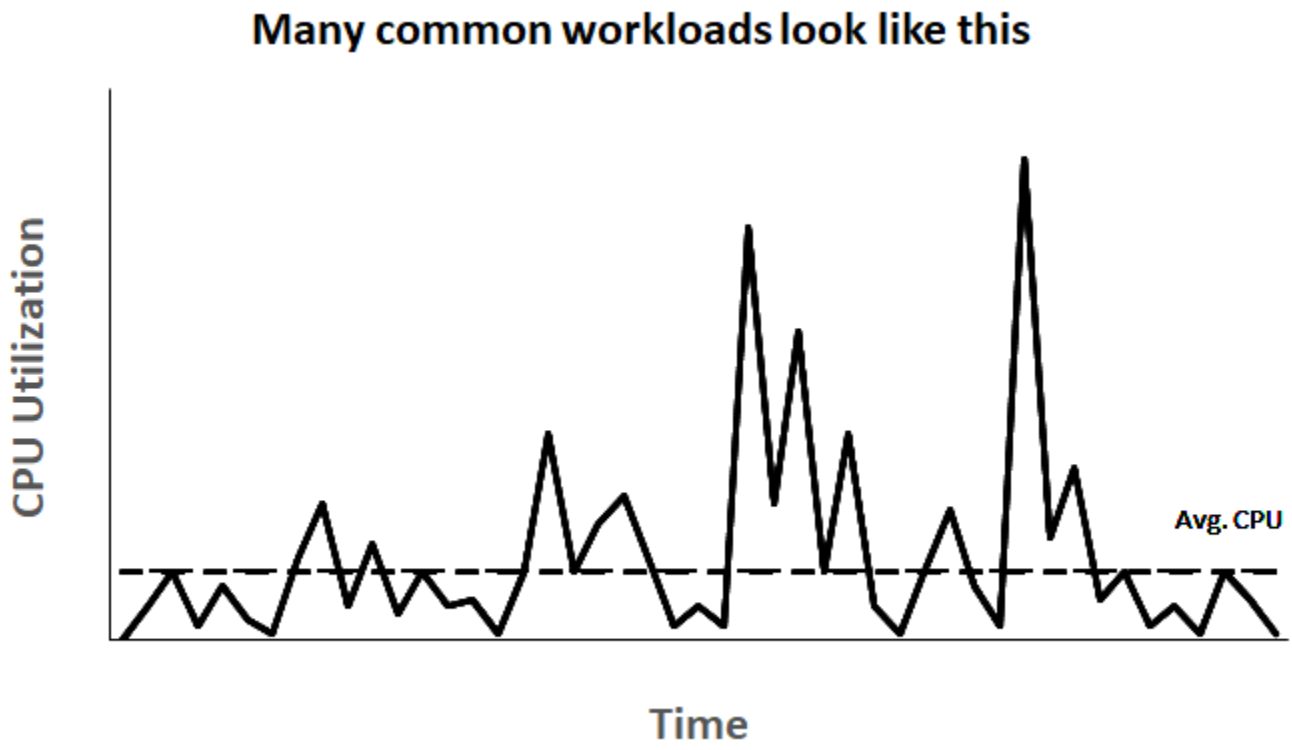
可能原因：公有 IPv4 地址已釋出

變更執行個體類型時，您必須先停止執行個體。當您停止執行個體時，我們會釋出公有 IPv4 地址，並給予您的執行個體新的公有 IPv4 地址。

若要在執行個體停止和啟動之間保留公有 IPv4 地址，建議您使用彈性 IP 地址。如需詳細資訊，請參閱彈性 IP 位址。

爆量效能執行個體

許多一般用途工作負載平均起來並不忙碌，而且不需要高級別的持續 CPU 效能。下圖說明客戶目前在 AWS 雲端中執行的許多常見工作負載的 CPU 使用率。



這些低至中等 CPU 使用率工作負載會導致 CPU 週期的浪費，因此您付出的費用超過使用量。為了克服此問題，您可以利用低成本爆量一般用途執行個體，也就是 T 執行個體。

T 執行個體系列提供基準 CPU 效能，能夠隨時長時間超越基準高效能執行。基準 CPU 的定義是為了滿足大多數一般用途工作負載的需求，包括大規模微服務、Web 伺服器、中小型資料庫、資料記錄、程式碼儲存庫、虛擬桌面、開發和測試環境，以及業務關鍵應用程式。T 執行個體在運算、記憶體和網路資源之間取得平衡，並為您提供最符合成本效益的方式，來執行各種具有低至中等 CPU 使用率的一般用途應用程式。與 M 執行個體相比，這些執行個體最多可節省 15% 的成本，並且透過更小、更經濟的執行個體大小可節省更多成本，提供最低 2 個 vCPU 和 0.5 GiB 的記憶體。較小的 T 執行個體大小 (例如超微型、微型、小型和中型) 非常適合需要少量記憶體且不需要高 CPU 使用率的工作負載。

> **Note**
>
> 本主題說明爆量 CPU。如需有關爆量網路效能的詳細資訊，請參閱 Amazon EC2 執行個體網路頻寬。

EC2 爆量執行個體類型

EC2 爆量執行個體包含 T4g、T3a 和 T3 執行個體類型以及前一代 T2 執行個體類型。

T4g 執行個體類型是最新一代的爆量執行個體。它們提供最佳的效能價格，並為您提供所有 EC2 執行個體類型中最低的成本。T4g 執行個體類型採用 Arm 型 AWS Graviton2 處理器，具有作業系統廠商、獨立軟體廠商以及熱門 AWS 服務和應用程式的廣泛生態系統支援。

下表摘要列出爆量執行個體類型之間的主要差異。

類型	描述	處理器系列
最新一代		
T4g	成本最低的 EC2 執行個體類型，與 T3 相比價格/效能高 40%，成本低 20%	具有 Arm Neoverse N1 核心的 AWS Graviton2 處理器
T3a	成本最低的 x86 型執行個體，成本比 T3 執行個體低 10%	AMD 第一代 EPYC 處理器
T3	適用於 x86 工作負載的最佳峰值價格/效能，比上一代 T2 執行個體的價格/效能低 30%	Intel Xeon 可擴充處理器 (Skylake、Cascade Lake 處理器)
上一代		
T2	上一代爆量執行個體	Intel Xeon 處理器

如需執行個體定價和其他規格的相關資訊，請參閱 Amazon EC2 定價和 Amazon EC2 執行個體類型。如需有關爆量網路效能的詳細資訊，請參閱 Amazon EC2 執行個體網路頻寬。

如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的 ，且不到 12 個月，您可以在特定用量限制內免費使用t2.micro執行個體 （ 或t2.micro無法使用 區域中的t3.micro執行個體 ）。如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t4g.micro、、t3.smallt4g.small執行個體類型 6 個月，或直到您的點數用完為止。如需詳細資訊，請參閱AWS 免費方案。

T 執行個體支援的購買選項

- On-Demand Instances
- Reserved Instances
- 專用執行個體 (僅限 T3)

- 專用執行個體 (僅限 T3，僅限 `standard` 模式)
- Spot 執行個體

如需詳細資訊，請參閱Amazon EC2 帳單與購買選項。

內容

- 最佳實務
- 爆量效能執行個體的重要概念
- 爆量效能執行個體的無限制模式
- 爆量效能執行個體的標準模式
- 設定爆量效能執行個體
- 監控爆量執行個體的 CPU 額度

最佳實務

請遵循這些最佳實務，讓高載效能執行個體發揮最大的效益。

- 請確定您所選擇的執行個體容量，高於作業系統和應用程式的最低記憶體容量需求。使用圖形化使用者介面的作業系統 (例如 Windows) 可能會耗用大量記憶體和 CPU 資源，因此在許多使用案例中可能需要 `t3.micro` 或更大的執行個體大小。隨著工作負載對記憶體和 CPU 的需求隨時間增加，使用 T 執行個體可獲得一定的靈活性，您可以擴展到相同執行個體類型的較大執行個體大小，或選取其他執行個體類型。
- 為您的帳戶啟用 AWS Compute Optimizer，並檢閱工作負載的 Compute Optimizer 建議。Compute Optimizer 可協助評估執行個體是否應該擴大規模以提高效能，還是應縮小規模以節省成本。Compute Optimizer 也可能會根據您的情況建議不同的執行個體類型。如需詳細資訊，請參閱「AWS Compute Optimizer 使用者指南」中的檢視 EC2 執行個體建議。

爆量效能執行個體的重要概念

傳統的 Amazon EC2 執行個體類型提供固定的 CPU 資源；爆量效能執行個體則提供 CPU 基準使用率，並可大幅提升 CPU 使用率至超越基準水準。這可確保您只需支付基準 CPU 外加任何額外的爆量 CPU 使用的費用，進而降低運算成本。基準使用率與大幅提升效能的能力，取決於 CPU 的額度。高載效能執行個體是唯一透過額度來使用 CPU 的執行個體類型。

當每個爆量效能執行個體維持在 CPU 基準以下時，會持續獲得額度，並在爆量超出基準時持續消耗額度。獲得或消耗的額度取決於執行個體的 CPU 使用率：

- 如果 CPU 使用率低於基準，則獲得的額度大於消耗的額度。
- 如果 CPU 使用率等於基準，則獲得的額度等於消耗的額度。
- 如果 CPU 使用率高於基準，則消耗的額度大於獲得的額度。

當獲得的額度大於消耗的額度時，差額稱為累積額度，它稍後可用於超出基準 CPU 使用率。同樣地，如果消耗的額度大於獲得的額度，則執行個體行為取決於額度組態模式 – 標準模式或無限制模式。

在標準模式中，當消耗的額度超過獲得的額度時，執行個體會使用累積額度以超出基準 CPU 使用率。如果沒有剩餘的累積額度，則執行個體會逐漸降至基準 CPU 使用率，並且無法超出基準線，直到累積更多額度為止。

在無限制模式中，如果執行個體超出基準 CPU 使用率，則執行個體會先使用累積額度進行超出。如果沒有剩餘的累積額度，則執行個體會消耗剩餘額度進行超出。當 CPU 使用率掉到基準以下，該執行個體會使用自己先前所獲得的 CPU 額度，來償還先前用掉的剩餘額度。獲得 CPU 額度來償還剩餘額度的能力，讓 Amazon EC2 能夠在 24 小時的期間，實現平均的執行個體 CPU 使用率。如果在 24 小時的期間，CPU 平均使用量超過基準，則會以每 vCPU 小時的其他固定費率，向執行個體收取其他使用量的費用。

目錄

- 重要概念和定義
- 獲得 CPU 額度
- CPU 額度獲得率
- CPU 額度累積限額
- 累積的 CPU 額度壽命
- 基準使用率

重要概念和定義

下列重要概念和定義適用於爆量效能執行個體。

CPU 使用率

CPU 使用率是執行個體上目前正在使用的已分配的 EC2 運算單位的百分比。此指標會測量執行個體上使用的已配置 CPU 週期百分比。CPU 使用率 CloudWatch 指標顯示每個執行個體的 CPU 使用率，而非每個核心的 CPU 使用率。執行個體的基準 CPU 規格也以每個執行個體的 CPU 使用率為基礎。若要使用 AWS 管理主控台 或 測量 CPU 使用率 AWS CLI，請參閱 取得特定執行個體的統計資料。

CPU 額度

vCPU 時間單位。

範例:

1 個 CPU 額度 = 1 個 vCPU * 100% 使用率 * 1 分鐘。

1 個 CPU 額度 = 1 個 vCPU * 50% 使用率 * 2 分鐘。

1 個 CPU 額度 = 2 個 vCPU * 25% 使用率 * 2 分鐘。

基準使用率

基準使用率是當獲得的 CPU 點數符合使用的 CPU 點數時，CPU 可用於淨點數餘額為零的層級。基準使用率也稱為基準。基準使用率表示為 vCPU 使用率百分比，其計算方式如下：基準使用率百分比 = (獲得的額度/vCPU 數)/60 分鐘。

如需每個爆量效能執行個體類型的基準使用率，請參閱點數表。

獲得的額度

執行個體在執行時持續獲得的額度。

每小時獲得的額度 = 基準使用率百分比 * vCPU 數量 * 60 分鐘

範例：

具有 2 個 vCPU 和 5% 基準使用率的 t3.nano 每小時可獲得 6 個額度，其計算方式如下：

2 個 vCPU * 5% 基準使用率 * 60 分鐘 = 每小時 6 個額度

消耗或使用的額度

執行個體在執行時持續使用的額度。

每分鐘消耗的 CPU 額度 = vCPU 數量 * CPU 使用率 * 1 分鐘

累積額度

當執行個體使用的額度少於基準使用率所需的額度時，未消耗的 CPU 額度。換句話說，累積額度 = (獲得的額度 - 使用的額度) 低於基準。

範例：

如果在 2% CPU 使用率的情況下執行 t3.nano (低於一小時 5% 的基準)，則累積額度的計算方式如下：

累積的 CPU 額度 = (每小時獲得的額度 - 每小時使用的額度) = 6 - 2 個 vCPU * 2% CPU 使用率 * 60 分鐘 = 6 - 2.4 = 每小時 3.6 個累積額度

額度累積限額

它取決於執行個體大小，但通常等於 24 小時內獲得的最大額度。

範例：

對於 t3.nano，額度累積限額 = 24 * 6 = 144 個額度

啟動額度

僅適用於為標準模式設定的 T2 執行個體。啟動額度是分配給新 T2 執行個體的有限 CPU 額度，因此在標準模式下啟動時，它可以超出基準線。

剩餘額度

執行個體耗盡其累積額度餘額之後所消耗的額度。剩餘額度是專為爆量執行個體所設計，以便長時間維持高效能，並且僅用於無限制模式。剩餘額度餘額用於決定執行個體在無限制模式下爆量使用了多少額度。

標準模式

額度組態模式，透過消耗掉額度餘額中所累積的額度，此模式可讓執行個體超出基準。

無限制模式

額度組態模式，透過在任何期間皆隨需提供高 CPU 使用率，此模式可讓執行個體超出基準。如果在累計 24 小時的期間或執行個體的生命週期期間 (以較短者為準)，執行個體的 CPU 平均使用率等於或低於基準，則執行個體的每小時定價將自動涵蓋所有 CPU 使用量峰值。如果執行個體需要長時間以更高的 CPU 使用率執行，也可使用每 vCPU 小時的其他固定費率來執行。

下表摘要列出爆量執行個體類型之間的主要額度差異。

類型	支援的 CPU 額度類型	額度組態模式	執行個體啟動和停止之間累積的 CPU 額度的使用壽命
最新一代			
T4g	獲得的額度、累積的額度、消耗的額度、	標準、無限制 (預設)	7 天 (額度會在執行個體停止後保留 7 天)

類型	支援的 CPU 額度類型	額度組態模式	執行個體啟動和停止之間累積的 CPU 額度的使用壽命
	剩餘額度 (僅限無限制模式)		
T3a	獲得的額度、累積的額度、消耗的額度、剩餘額度 (僅限無限制模式)	標準、無限制 (預設)	7 天 (額度會在執行個體停止後保留 7 天)
T3	獲得的額度、累積的額度、消耗的額度、剩餘額度 (僅限無限制模式)	標準、無限制 (預設)	7 天 (額度會在執行個體停止後保留 7 天)
上一代			
T2	獲得的額度、累積的額度、消耗的額度、啟動額度 (僅限標準模式)、剩餘額度 (僅限無限制模式)	標準 (預設)、無限制	0 天 (當執行個體停止時，額度會消失)

> **Note**
>
> 專用執行個體上啟動的 T3 執行個體不支援無限制模式。

獲得 CPU 額度

視執行個體的大小而定，每個爆量效能執行個體每小時會持續獲得 (以毫秒級的解析度) 固定比率的 CPU 額度。管理額度累積或使用的會計程序，也會以毫秒級的解析度進行，因此您無需擔心 CPU 額度超支；CPU 的短時間突發只會使用一小部分的 CPU 額度。

如果爆量效能執行個體所使用的 CPU 資源比基準使用率所需的還少 (例如在閒置時)，則未使用的 CPU 額度會累計到 CPU 額度餘額。如果爆量效能執行個體的效能需要突增到超過基準使用率，將會用掉累

積的額度。爆量效能執行個體所累積的額度愈多，在需要更多的 CPU 使用率時，能夠突增超過基準使用率的時間也愈長。

下表列出了爆量效能執行個體類型、每小時獲得的 CPU 額度、執行個體可累計獲得的 CPU 額度上限、每個執行個體的 vCPU 數量，以及使用單一 vCPU 時的基準使用率 (以完整核心的百分比表示)。

執行個體類型	每小時獲得的 CPU 額度	可累積的額度上限*	vCPU***	每個 vCPU 的基準使用率
T2				
t2.nano	3	72	1	5%
t2.micro	6	144	1	10%
t2.small	12	288	1	20%
t2.medium	24	576	2	20%**
t2.large	36	864	2	30%**
t2.xlarge	54	1296	4	22.5%**
t2.2xlarge	81.6	1958.4	8	17%**
T3				
t3.nano	6	144	2	5%**
t3.micro	12	288	2	10%**
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**
T3a				

執行個體類型	每小時獲得的 CPU 額度	可累積的額度上限*	vCPU***	每個 vCPU 的基準使用率
t3a.nano	6	144	2	5%**
t3a.micro	12	288	2	10%**
t3a.small	24	576	2	20%**
t3a.medium	24	576	2	20%**
t3a.large	36	864	2	30%**
t3a.xlarge	96	2304	4	40%**
t3a.2xlarge	192	4608	8	40%**
T4g				
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**
t4g.small	24	576	2	20%**
t4g.medium	24	576	2	20%**
t4g.large	36	864	2	30%**
t4g.xlarge	96	2304	4	40%**
t4g.2xlarge	192	4608	8	40%**

備註

* 可累計的額度等於在 24 小時期間可獲得的額度。

備註

** 表格中所顯示的是每個 vCPU 的百分比基準使用率。在 CloudWatch 中，會顯示每 vCPU 的 CPU 使用率。例如，t3.large 執行個體以基準水準運作時的 CPU 使用率，在 CloudWatch CPU 指標中顯示為 30%。如需如何計算基準使用率的資訊，請參閱基準使用率。

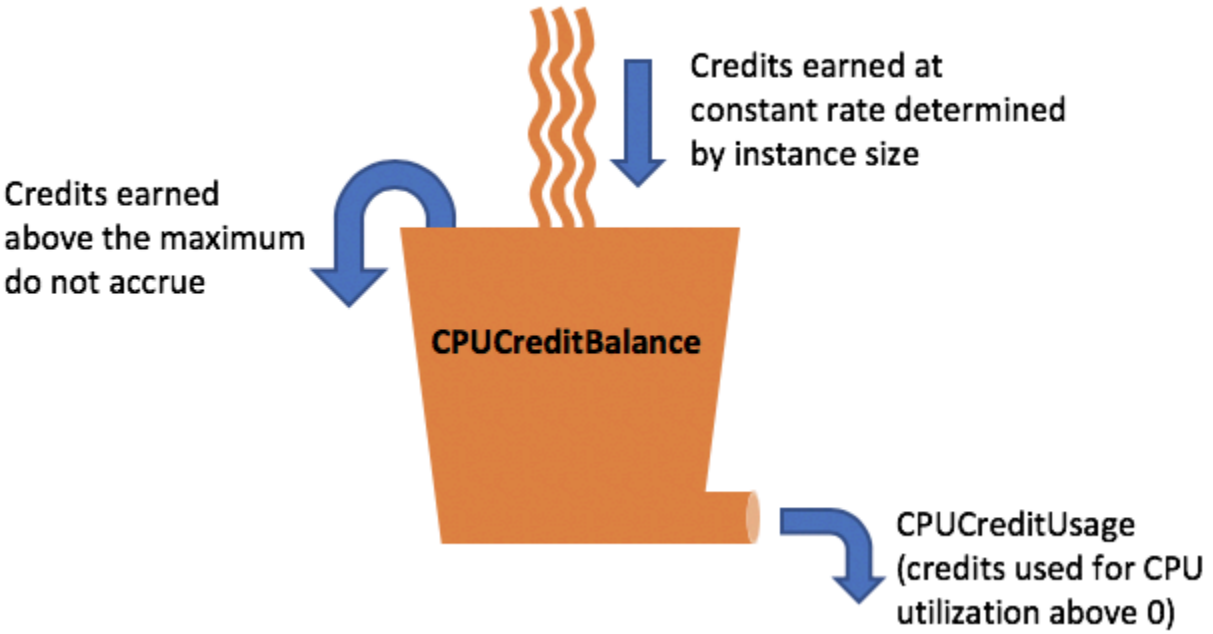
*** 每個 vCPU 都是 Intel Xeon 核心或 AMD EPYC 核心的一個執行緒，但 T2 和 T4g 執行個體除外。

CPU 額度獲得率

每小時所獲得的 CPU 額度，是由執行個體的大小所決定。例如，t3.nano 每小時可獲得 6 點，而 t3.small 每小時可獲得 24 點。上表列出所有執行個體的額度獲得率。

CPU 額度累積限額

執行中的執行個體所獲得的額度，雖然永遠不會過期，但是執行個體可累積獲得的額度有其上限。此額度上限取決於 CPU 額度餘額的限額。在達到上限之後，任何新獲得的額度都會遭到捨棄，如下圖所示。全滿的水桶代表 CPU 額度餘額的限額，溢出的水代表新獲得的額度已超過限額。



每種 執行個體容量大小各有不同的 CPU 額度餘額限制。例如，t3.micro 執行個體可在 CPU 額度餘額中，累積最多 288 點獲得的 CPU 額度。上面的表格列出了每個 執行個體的可累積獲得的額度上限。

T2 標準執行個體也會獲得啟動額度。啟動額度不會列入 CPU 額度餘額限額的計算。如果 T2 執行個體並未用掉其啟動額度，而且在 24 小時的期間維持閒置，同時累計獲得的額度，則其 CPU 額度餘額會顯示為超過限額。如需詳細資訊，請參閱啟動額度。

T4g、T3a 和 T3 執行個體不會獲得啟動抵用金。這些執行個體會根據預設啟動為 unlimited，因此啟動時可立即大幅提升效能，而不需任何啟動額度。依預設，在專用執行個體啟動上啟動的 T3 執行個體為 standard；專用執行個體上的 T3 執行個體不支援 unlimited 模式。

累積的 CPU 額度壽命

執行中的執行個體，其 CPU 額度不會過期。

若為 T2，CPU 額度餘額不會在執行個體停止與啟動之間持續保存。如果您停止 T2 執行個體，則執行個體會失去其所有累積的額度。

對於 T4g、T3a 和 T3，CPU 抵用金餘額會在執行個體停止之後持續保存七天，之後抵用金便會消失。如果您在七天內啟動執行個體，則不會失去任何額度。

如需詳細資訊，請參閱 CloudWatch 指標表格中的 CPUCreditBalance。

基準使用率

基準使用率是當獲得的 CPU 點數符合使用的 CPU 點數時，CPU 可用於淨點數餘額為零的層級。基準使用率也稱為基準。

基準使用率是以 vCPU 使用率百分比表示，其計算方式如下：

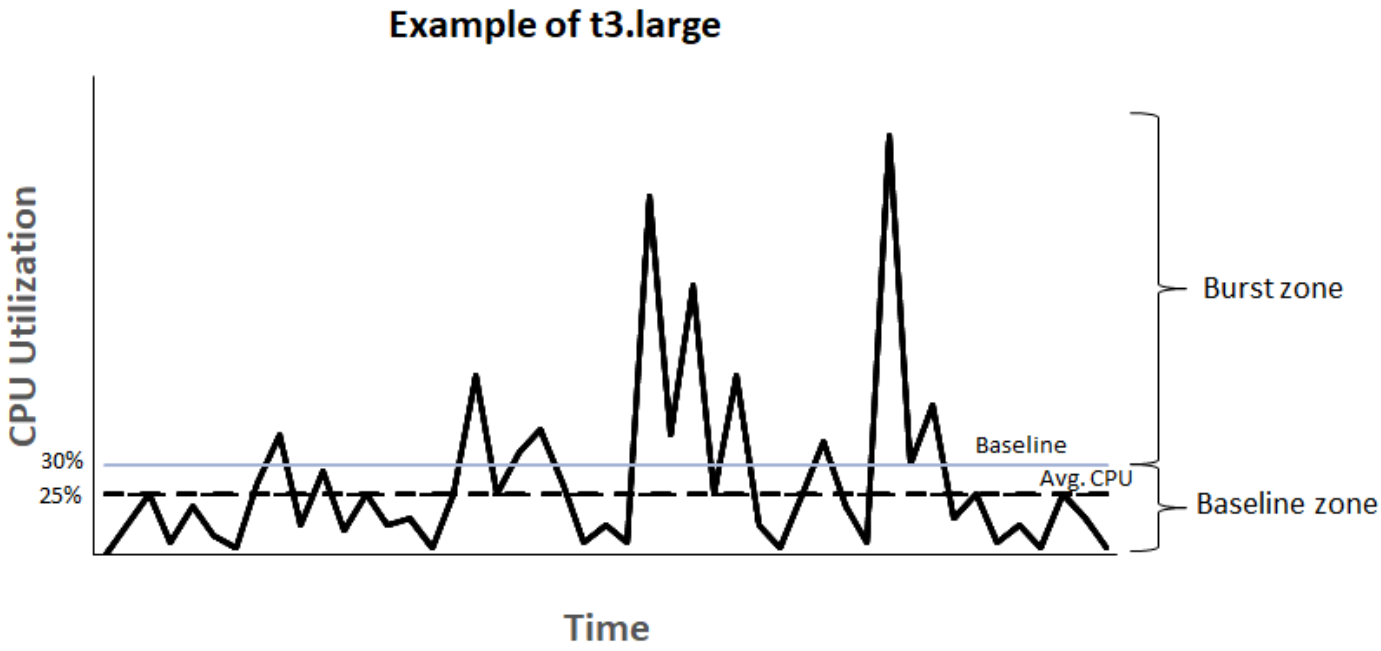
```
(number of credits earned/number of vCPUs)/60 minutes = % baseline
utilization
```

例如，具有 2 個 vCPU 的 t3.nano 執行個體每小時可獲得 6 點，因此會產生 5% 的基準使用率，其計算方式如下：

```
(6 credits earned/2 vCPUs)/60 minutes = 5% baseline utilization
```

具有 2 個 vCPU 的 t3.large 執行個體每小時可獲得 36 點，因此會產生 30% 的基準使用率 ((36/2)/60)。

下圖提供平均 CPU 使用率低於基準的 t3.large 範例。



爆量效能執行個體的無限制模式

設定為 unlimited 的高載效能執行個體，在任何期間皆可隨需提供高 CPU 使用率。如果在累計 24 小時的期間或執行個體的生命週期期間 (以較短者為準)，執行個體的 CPU 平均使用率等於或低於基準，則執行個體的每小時定價將自動涵蓋所有 CPU 使用量峰值。

針對絕大多數的一般用途工作負載，設定為 unlimited 的執行個體可提供足夠效能，而不需支付任何額外費用。如果執行個體需要長時間以更高的 CPU 使用率執行，也可使用每 vCPU 小時的其他固定費率來執行。如需有關定價的資訊，請參閱 Amazon EC2 定價與 T2/T3/T4 無限制模式定價。

如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，且您在AWS 免費方案優惠下使用 t2.micro或 t3.micro執行個體，並在 unlimited 模式下使用，則如果您在滾動 24 小時期間的平均使用率超過執行個體的基準使用率，則可能需要付費。

依預設，T4g、T3a 和 T3 執行個體會以 unlimited 狀態啟動 (除非您變更預設值)。如果 24 小時期間內的平均 CPU 使用量超過基準，則剩餘額度會產生費用。如果您以 unlimited 形式啟動 Spot 執行個體，並計劃立即持續使用這些執行個體一段短暫的時間，且沒有產生 CPU 抵用金的閒置時間，則剩餘抵用金可能會產生費用。建議您在標準模式下啟動 Spot 執行個體，以免需要支付更高的費用。如需詳細資訊，請參閱剩餘額度可能會產生費用及啟動爆量效能執行個體。

> **Note**
>
> 依預設，在專用執行個體啟動上啟動的 T3 執行個體為 `standard`；專用執行個體上的 T3 執行個體不支援 `unlimited` 模式。

目錄

- 爆量執行個體的無限制模式概念
  - 無限制爆量效能執行個體如何運作
  - 使用無限制模式與固定 CPU 的時機
  - 剩餘額度可能會產生費用
  - 無限高載效能的成本是多少？
  - T2 無限制執行個體沒有啟動額度
  - 啟用無限制模式
  - 在無限制與標準之間切換時，額度會有什麼變化
  - 監控額度用量
- 爆量執行個體的無限制模式範例
  - 範例 1：說明 T3 Unlimited 的額度使用
  - 範例 2：說明 T2 Unlimited 的額度使用

爆量執行個體的無限制模式概念

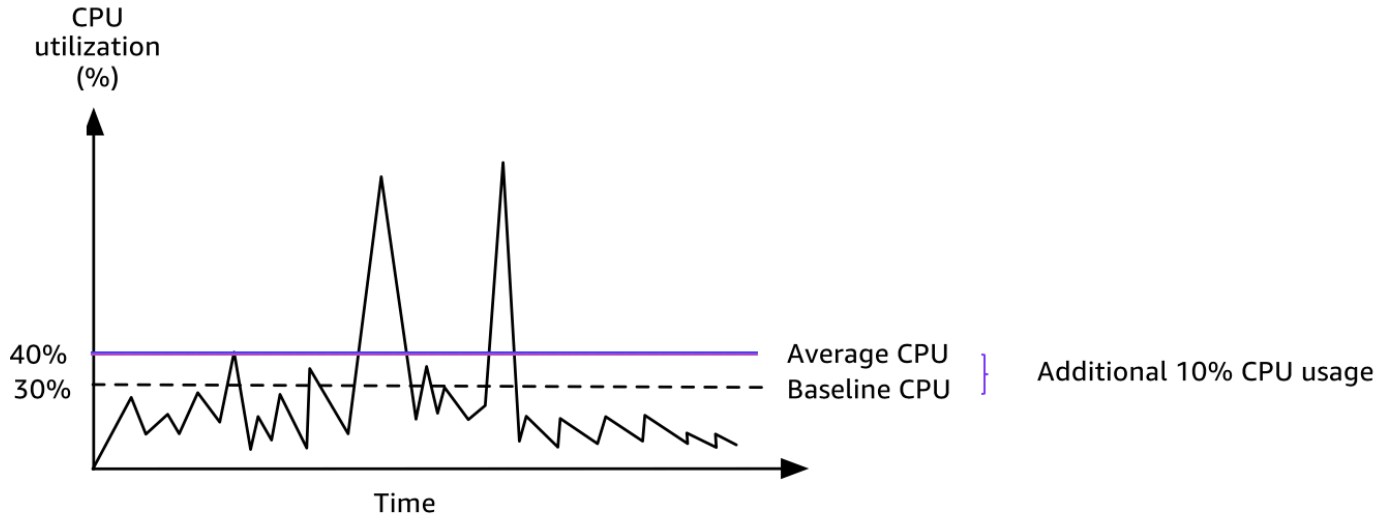
`unlimited` 模式是爆量效能執行個體的額度組態選項。對於執行中或已停止的執行個體，隨時都可以啟用或停用它。您可以在每個可爆量效能執行個體系列的每個 AWS 區域的帳戶層級將 設定為預設點數選項，以便使用`unlimited`預設點數選項啟動帳戶中的所有新可爆量效能執行個體。

無限制爆量效能執行個體如何運作

如果設定為 `unlimited` 的爆量效能執行個體將其 CPU 額度餘額用完了，則可以使用多餘的額度來大幅提升效能以超過基準。當 CPU 使用率掉到基準以下，該執行個體會使用自己先前所獲得的 CPU 額度，來償還先前用掉的剩餘額度。獲得 CPU 額度來償還剩餘額度的能力，讓 Amazon EC2 能夠在 24 小時的期間，實現平均的執行個體 CPU 使用率。如果在 24 小時的期間，CPU 平均使用量超過基準，則會以每 vCPU 小時的其他固定費率，向執行個體收取其他使用量的費用。

下圖顯示 `t3.large` 的 CPU 使用量。`t3.large` 的基準 CPU 使用率為 30%。如果在 24 小時的期間，執行個體以平均 30% 或更少 CPU 使用率執行，則沒有額外費用，因為執行個體每小時定價已涵

蓋成本。不過，如果在 24 小時的期間，執行個體以平均 40% CPU 使用率執行 (如圖所示)，則會以每 vCPU 小時的其他固定費率，向執行個體收取額外 10% CPU 使用率的費用。



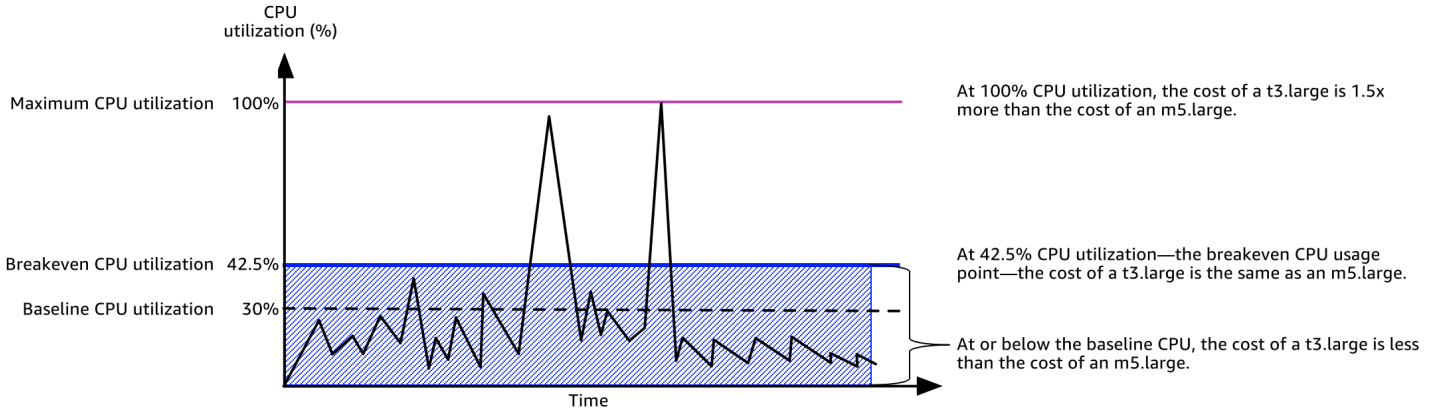
如需每一個執行個體類型的每 vCPU 基準使用率的詳細資訊，以及每一個執行個體類型可以賺取多少額度，請參閱額度表。

使用無限制模式與固定 CPU 的時機

判斷您應該在 unlimited 模式中使用爆量效能執行個體 (例如 T3)，還是固定效能執行個體 (例如 M5) 時，您需要判定損益兩平的 CPU 使用量。爆量效能執行個體的損益兩平 CPU 使用量，即是爆量效能執行個體成本與固定效能執行個體相同的那個點。損益兩平的 CPU 使用量可協助您判斷下列事項：

- 如果在 24 小時的期間，平均 CPU 使用量為等於或低於損益兩平的 CPU 使用量，請在 unlimited 模式下使用爆量效能執行個體，讓您能夠得益於爆量效能執行個體的更低價格，同時又能得到與固定效能執行個體相同的效能。
- 如果在 24 小時的期間，平均 CPU 使用量為超過損益兩平的 CPU 使用量，則爆量效能執行個體的成本將超過同等大小的固定效能執行個體。如果 T3 執行個體持續以 100% CPU 爆量，則您最終付出的成本將是同等大小 M5 執行個體的 1.5 倍。

下圖顯示損益兩平的 CPU 使用量點，其中 t3.large 的成本與 m5.large 相同。t3.large 的損益兩平 CPU 使用量點為 42.5%。如果平均 CPU 使用量為 42.5%，則執行 t3.large 的成本與 m5.large 相同，但若平均 CPU 使用量超過 42.5%，則更為昂貴。如果工作負載需要的平均 CPU 使用量小於 42.5%，則您可以得益於 t3.large 的更低價格，同時又可得到與 m5.large 相同的效能。



下表顯示如何計算損益兩平的 CPU 使用量閾值，讓您可以在 `unlimited` 模式下使用爆量效能執行個體，或使用固定效能執行個體時，判斷何者更便宜。資料表中的資料欄標示為 A 到 K。

執行個體類型	vCPU	T3 價格*/小時	M5 價格*/小時	價格差異	每個 vCPU 的 T3 基準使用率 (%)	剩餘額度的每 vCPU 小時費用	每 vCPU 分鐘費用	每 vCPU 可用的額外爆量分鐘	其他可用的 CPU %	損益兩平 CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I / 60) / B	K = F + J
t3.large	2	0.0835 USD	0.096 USD	0.0125 USD	30%	0.05 USD	0.000833 USD	15	12.5%	42.5%

備註

* 價格是以 us-east-1 和 Linux OS 為基礎。

資料表提供下列資訊：

- 資料欄 A 顯示執行個體類型，即 `t3.large`。

- 資料欄 B 顯示 t3.large 的 vCPU 數目。
- 資料欄 C 顯示 t3.large 每小時的價格。
- 資料欄 D 顯示 m5.large 每小時的價格。
- 資料欄 E 顯示t3.large與m5.large的價格差異。
- 資料欄 F 顯示 t3.large 的每 vCPU 基準使用率，即 30%。根據基準，執行個體的每小時成本涵蓋 CPU 使用量的成本。
- 資料欄 G 顯示每 vCPU 小時的其他固定費率，若執行個體在耗盡其賺取的額度之後爆量為 100% CPU，即會以此費率向執行個體收費。
- 資料欄 H 顯示每 vCPU 分鐘的其他固定費率，若執行個體在耗盡其賺取的額度之後爆量為 100% CPU，即會以此費率向執行個體收費。
- 資料欄 I 顯示 t3.large 每小時可以額外爆量為 100% CPU 幾分鐘，同時又可支付與 m5.large 相同的每小時價格。
- 資料欄 J 顯示超過基準的額外 CPU 使用量 (以 % 表示)，而執行個體可以爆量為這個使用量，同時又可支付與 m5.large 相同的每小時價格。
- 資料欄 K 顯示損益兩平的 CPU 使用量 (以 % 表示)，t3.large 可以爆量為這個使用量，而支付的費用不會超過 m5.large。除此之外，t3.large 的成本超過 m5.large。

下表顯示相較於類似大小的 M5 執行個體類型，T3 執行個體類型的損益兩平 CPU 使用量 (以 % 表示)。

T3 執行個體類型	相較於 M5，T3 的損益兩平 CPU 使用量 (以 % 表示)
t3.large	42.5%
t3.xlarge	52.5%
t3.2xlarge	52.5%

剩餘額度可能會產生費用

如果執行個體的 CPU 平均使用率等於或低於基準，執行個體不會產生額外的費用。由於執行個體在 24 小時的期間獲得了最高額度 (例如，t3.micro 執行個體可以在 24 小時的期間獲得最多 288 點)，因此可以用掉等於最高額度的剩餘額度，而不會產生費用。

不過，如果 CPU 使用率保持在基準以上，則執行個體就無法獲得足夠的額度，來償還先前用掉的剩餘額度。未償還的剩餘額度，會以每 vCPU 小時的其他固定費率來收費。如需有關費率的資訊，請參閱 T2/T3/T4g 無限制模式定價。

發生下列任一情況時，將會針對先前用掉的剩餘額度來收取費用。

- 支出剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量。在小時結束時，將收取超過最大值的支出剩餘額度的費用。
- 執行個體已停止或終止。
- 執行個體從 `unlimited` 切換至 `standard`。

支出剩餘額度可透過 CloudWatch 指標 `CPUSurplusCreditBalance` 來追蹤。收費的剩餘額度可透過 CloudWatch 指標 `CPUSurplusCreditsCharged` 來追蹤。如需詳細資訊，請參閱高載效能執行個體的其他 CloudWatch 指標。

無限高載效能的成本是多少？

若您使用剩餘抵用金，且未透過獲得的抵用金來支付這些抵用金 (請參閱 剩餘額度可能會產生費用)，您為每個剩餘抵用金的 vCPU 小時支付固定額外費率。費率列在 Amazon EC2 隨需定價頁面上的 T2/T3/T4g 無限制模式定價區段中。

T2 無限制執行個體沒有啟動額度

T2 標準執行個體會收到啟動額度，但 T2 無限制執行個體則不會收到。只要在累計 24 小時的期間或是其生命週期期間 (以較短者為準)，T2 無限制執行個體的 CPU 平均使用率等於或低於基準，此等執行個體即可隨時讓效能爆量超過基準水準，而不會產生額外的費用。因此，T2 Unlimited 執行個體不需要使用啟動額度，以在啟動之後立即達成高效能。

如果 T2 執行個體從 `standard` 切換為 `unlimited`，則在將剩下的 `CPUCreditBalance` 結轉過去之前，會先從 `CPUCreditBalance` 移除所有累積的啟動額度。

T4g、T3a 和 T3 執行個體不會收到啟動抵用金，因為這些執行個體預設會以無限制模式啟動，因此會在啟動時立即爆量。無限制模式抵用金組態可讓 T4g、T3a 和 T3 執行個體根據需要使用盡可能多的 CPU，以盡可能長時間地超出基準。

啟用無限制模式

在執行中或已停止的執行個體上，可以隨時從 `unlimited` 切換為 `standardUnlimited`，也可從 `standard` 切換為 `unlimited`。如需詳細資訊，請參閱在啟動時設定抵用金規格及管理爆量效能執行個體的抵用金規格。

您可以在每個可爆量效能執行個體系列的每個 AWS 區域的帳戶層級將 設定為unlimited預設點數選項，以便使用預設點數選項啟動帳戶中的所有新可爆量效能執行個體。如需詳細資訊，請參閱管理帳戶的預設抵用金規格。

您可以使用 Amazon EC2 主控台或 AWS CLI，檢查爆量效能執行個體是否已設定為 unlimited 或 standard。如需詳細資訊，請參閱設定爆量效能執行個體。

在無限制與標準之間切換時，額度會有什麼變化

CPUCreditBalance 是 CloudWatch 指標，會追蹤執行個體已累積的額度。CPUSurplusCreditBalance 是 CloudWatch 指標，會追蹤執行個體花費的剩餘額度。

當您將設定為 unlimited 的執行個體變更為 standard，會發生下列情況：

- CPUCreditBalance 值會維持不變結轉過去。
- CPUSurplusCreditBalance 值會立即收取費用。

當 standard 執行個體切換為 unlimited 時，會發生下列情況：

- 將包含累積獲得之額度的 CPUCreditBalance 值結轉過去。
- 若為 T2 Standard 執行個體，從 CPUCreditBalance 值移除所有啟動額度，然後將剩下的 CPUCreditBalance 值 (包含累積獲得的額度) 結轉過去。

監控額度用量

若要了解執行個體花費的額度是否超過基準提供的額度，您可以使用 CloudWatch 指標來追蹤使用量，並設定每小時警示，來收到額度使用狀況的通知。如需詳細資訊，請參閱監控爆量執行個體的 CPU 額度。

爆量執行個體的無限制模式範例

下列範例說明設定為 unlimited 之執行個體的額度使用。

範例

- 範例 1：說明 T3 Unlimited 的額度使用
- 範例 2：說明 T2 Unlimited 的額度使用

範例 1：說明 T3 Unlimited 的額度使用

在此範例中，您會看到啟動為 t3.nano 之 unlimited 執行個體的 CPU 使用率，以及該執行個體如何用掉獲得的和多餘的額度，來維持 CPU 的使用率。

t3.nano 執行個體會在累計 24 小時的期間獲得 144 點的 CPU 額度，這些額度可用來兌換 144 分鐘的 vCPU 使用。當該執行個體用完其 CPU 額度餘額 (以 CloudWatch 指標 CPUCreditBalance 表示) 時，可以用掉多餘的 CPU 額度—尚未獲得的額度—以在所需的期間爆量。由於 t3.nano 執行個體在 24 小時期間已獲得最高的 144 點，因此可以用掉等於此最高額度的剩餘額度，而不會立即產生費用。如果該執行個體用掉超過 144 點的 CPU 額度，則在該小時結束時，將會收取額度差額的費用。

此範例的目的 (如下列圖表所示)，是要說明即使用完 CPUCreditBalance，執行個體如何可利用剩餘額度來大幅提升效能。下列的工作流程參考了圖表上的編號點：

P1 – 在圖表上的第 0 小時，執行個體會啟動為 unlimited，並立即開始獲得額度。該執行個體從啟動後即維持閒置，其 CPU 使用率為 0%，而且沒有用掉額度。所有為使用的額度都會累計到額度餘額。前 24 小時，CPUCreditUsage 為 0，而 CPUCreditBalance 值達到其最大值 144。

P2 – 接下來 12 小時，CPU 使用率為 2.5%，低於 5% 基準。執行個體獲得的額度超出其用掉的額度，但 CPUCreditBalance 值不得超出其最大值，即 144 點。

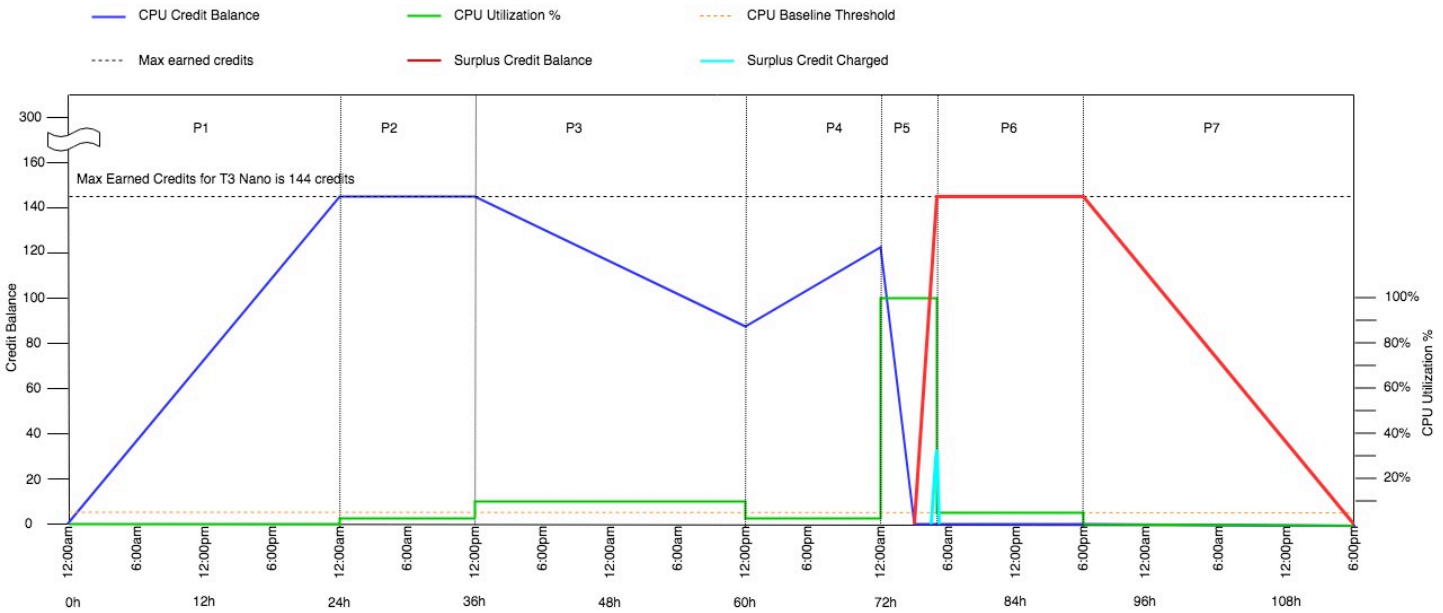
P3 – 接下來 24 小時，CPU 使用率為 7% (高於基準)，其需要用掉 57.6 點。執行個體用掉的額度超過獲得的額度，而且 CPUCreditBalance 值減少至 86.4 點。

P4 – 接下來 12 小時，CPU 使用率下降至 2.5% (低於基準)，其需要用掉 36 點。同時執行個體獲得 72 點。執行個體獲得的額度超過用掉的額度，而且 CPUCreditBalance 值增加至 122 點。

P5 – 接下來 5 小時，執行個體的 CPU 使用率爆量至 100%，並總共用掉 570 點來維持此爆量。在此期間大約一小時，執行個體耗盡其整個 CPUCreditBalance (即 122 點)，並開始用掉剩餘額度以維持高 CPU 使用率，在此期間總計剩餘 448 點 (570-122=448)。當 CPUSurplusCreditBalance 值達到 144 CPU 額度 (t3.nano 執行個體可在 24 小時期間獲得的最大額度) 時，獲得的額度無法抵銷任何用掉的剩餘額度。之後用掉的剩餘額度合計達 304 點 (448-144=304)，因此在該小時結束時，會對 304 點產生少許的額外費用。

P6 – 接下來 13 小時，CPU 使用率為 5% (基準)。執行個體獲得與其用掉一樣多的額度，不需額外支付 CPUSurplusCreditBalance。CPUSurplusCreditBalance 值仍為 144 點。

P7 – 在此範例的最後 24 小時，執行個體閒置且 CPU 使用率為 0%。在此期間，執行個體獲得 144 點，用來支付 CPUSurplusCreditBalance。



範例 2：說明 T2 Unlimited 的額度使用

在此範例中，您會看到啟動為 t2.nano 之 unlimited 執行個體的 CPU 使用率，以及該執行個體如何用掉獲得的和多餘的額度，來維持 CPU 的使用率。

t2.nano 執行個體會在累計 24 小時的期間獲得 72 點的 CPU 額度，這些額度可用來兌換 72 分鐘的 vCPU 使用。當該執行個體用完其 CPU 額度餘額 (以 CloudWatch 指標 CPUCreditBalance 表示) 時，可以用掉多餘的 CPU 額度—尚未獲得的額度—以在所需的期間爆量。由於 t2.nano 執行個體在 24 小時期間已獲得最高的 72 點，因此可以用掉等於此最高額度的剩餘額度，而不會立即產生費用。如果該執行個體用掉超過 72 點的 CPU 額度，則在該小時結束時，將會收取額度差額的費用。

此範例的目的 (如下列圖表所示)，是要說明即使用完 CPUCreditBalance，執行個體如何可利用剩餘額度來大幅提升效能。您可以假設，在圖表中時間線的開頭，執行個體已有累積的額度餘額，此餘額等於該執行個體在 24 小時期間內可以獲得的最高額度。下列的工作流程參考了圖表上的編號點：

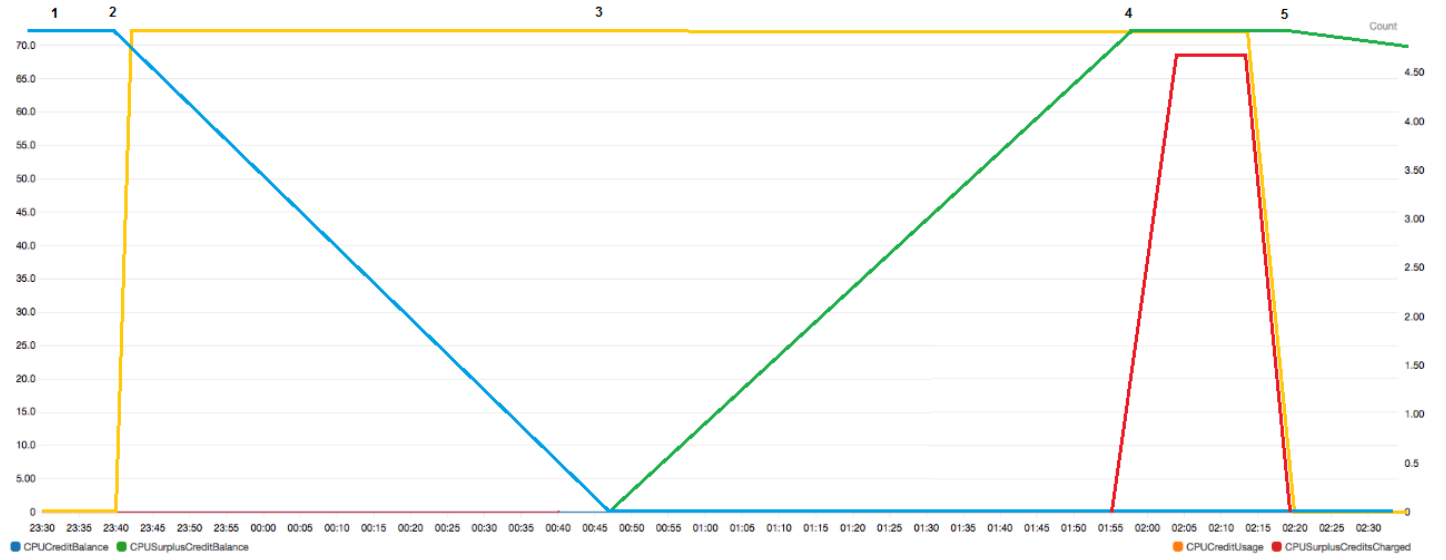
1 – 在前 10 分鐘期間，CPUCreditUsage 為 0，而 CPUCreditBalance 維持在最大值 72 點。

2 – 在 23:40，隨著 CPU 的使用率增加，執行個體用掉了 CPU 額度，CPUCreditBalance 值減少。

3 – 在大約 00:47 時，執行個體用完整個 CPUCreditBalance，並且開始使用剩餘額度來維持 CPU 的高使用率。

4 – 剩餘額度一直用到 01:55，此時 CPUSurplusCreditBalance 到達 72 點的 CPU 額度。這等於 t2.nano 執行個體在 24 小時期間內可獲得的最高點數。之後，任何用掉的剩餘額度，都不能用 24 小時期間內所獲得的額度抵銷，因此在該小時結束時，會產生少許的額外費用。

5 – 執行個體繼續使用剩餘額度，直到大約 02:20。此時，CPU 的使用率掉到基準以下，執行個體開始以每小時 3 點的速度獲得額度 (或每 5 分鐘 0.25 點)，並使用這些額度來償還 CPUSurplusCreditBalance。在 CPUSurplusCreditBalance 值減少到 0 之後，執行個體開始以每 5 分鐘 0.25 點的速度，累積其 CPUCreditBalance 中的獲得額度。



All metrics | Graphed metrics (4) | Graph options

Label	Details	Statistic	Period	Y Axis	Actions
CPUCreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUCreditBalance	Maximum	5 Minutes		
CPUCreditUsage	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUCreditUsage	Maximum	5 Minutes		
CPUSurplusCreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditBalance	Maximum	5 Minutes		
CPUSurplusCreditsCharged	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditsCharged	Maximum	5 Minutes		

計算帳單 (Linux 執行個體)

剩餘額度的成本為每 vCPU 小時 0.05USD。該執行個體在 01:55 到 02:20 之間，用掉了約 25 點剩餘額度，這等於 0.42 vCPU 小時。此執行個體其他的費用包括 0.42 vCPU 小時 x 0.05USD/vCPU 小時 = 0.021USD，四捨五入至 0.02USD。下列是這個 T2 Unlimited 執行個體月底的帳單：

Amazon Elastic Compute Cloud running Linux/UNIX		
$0.0058 per On Demand Linux t2.nano Instance Hour	720.000 Hrs	$4.18

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.05 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.02

計算帳單 (Windows 執行個體)

剩餘額度的成本為每 vCPU 小時 0.096USD。該執行個體在 01:55 到 02:20 之間，用掉了約 25 點剩餘額度，這等於 0.42 vCPU 小時。此執行個體其他的費用包括 0.42 vCPU 小時 x 0.096USD/vCPU 小時 = 0.04032USD，四捨五入至 0.04USD。下列是這個 T2 Unlimited 執行個體月底的帳單：

Amazon Elastic Compute Cloud running Windows		
$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	$5.83

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.04

您可以設定帳單提醒，每小時通知任何累計的費用，並根據需要採取行動。

爆量效能執行個體的標準模式

設定為 `standard` 的高載效能標準執行個體，適用於平均 CPU 使用率持續低於執行個體之基準 CPU 使用率的工作負載。若要讓效能突增至超過基準，執行個體會用掉在 CPU 額度餘額中所累積的額度。如果執行個體累積的額度過低，CPU 使用率會逐漸降到基準水準，如此執行個體就不會在其累積的 CPU 額度餘額用完時，經歷效能急遽下降的狀況。如需詳細資訊，請參閱爆量效能執行個體的重要概念。

目錄

- 爆量執行個體的標準模式概念
  - 標準爆量效能執行個體如何運作
  - 啟動額度
  - 啟動額度限制
  - 啟動額度與獲得額度之間的差異
- 爆量執行個體的標準模式範例
  - 範例 1：說明 T3 Standard 的額度使用
  - 範例 2：說明 T2 Standard 的額度使用
    - 期間 1：1 – 24 小時
    - 期間 2：25 – 36 小時
    - 期間 3：37 – 61 小時
    - 期間 4：62 – 72 小時
    - 期間 5：73 – 75 小時
    - 期間 6：76 – 90 小時
    - 期間 7：91 – 96 小時

爆量執行個體的標準模式概念

standard 模式是爆量效能執行個體的組態選項。對於執行中或已停止的執行個體，隨時都可以啟用或停用它。您可以在每個可爆量效能執行個體系列的每個 AWS 區域的帳戶層級將 設定為standard預設點數選項，以便使用預設點數選項啟動帳戶中的所有新可爆量效能執行個體。

標準爆量效能執行個體如何運作

當設定為 standard 的爆量效能執行個體處於執行中狀態時，每小時會持續獲得 (以毫秒級的解析度) 固定比率的額度。若為 T2 Standard，當執行個體停止時，會失去所有已累積的額度，而且其額度餘額會重設為零。當此等執行個體重新啟動時，會獲得一組新的啟動額度，並且開始累積獲得的額度。對於 T4g、T3a 和 T3 標準執行個體，CPU 抵用金餘額會在執行個體停止之後持續保存七天，之後抵用金便會消失。如果您在七天內啟動執行個體，則不會失去任何額度。

T2 標準執行個體會獲得兩種 CPU 額度：獲得的額度和啟動額度。當 T2 Standard 執行個體處於執行狀態時，每小時會持續獲得 (以毫秒級的解析度) 固定比率的額度。一開始，執行個體尚未獲得額度，可能無法順暢啟動；因此，為了提供流暢啟動的體驗，執行個體在開始時會獲得啟動額度，可以在累積獲得額度的同時先用掉這些額度。

T4g、T3a 和 T3 執行個體不會收到啟動抵用金，因為這些執行個體支援無限制模式。無限制模式抵用金組態可讓 T4g、T3a 和 T3 執行個體根據需要使用盡可能多的 CPU，以盡可能長時間地超出基準。

啟動額度

在啟動或開始時，T2 Standard 執行個體會獲得每 vCPU 30 點的啟動額度，T1 Standard 執行個體可獲得 15 點的啟動額度。例如，t2.micro 執行個體擁有一個 vCPU，會獲得 30 點的啟動額度，而 t2.xlarge 執行個體擁有四個 vCPU，會獲得 120 點的啟動額度。啟動額度的設計旨在提供流暢的啟動體驗，讓執行個體能夠在尚未獲得累計的額度之前，先在啟動之後立即大幅提升效能。

啟動額度會比獲得的額度先用掉。未用掉的啟動額度會累計到 CPU 額度餘額，但不會列入 CPU 額度餘額限額的計算。例如，t2.micro 執行個體具有最多 144 點獲得額度的 CPU 額度餘額上限。如果該執行個體在啟動後保持閒置 24 小時，其 CPU 額度餘額會達到 174 點 (30 點的啟動額度 + 144 點的獲得額度)，超過了上限。不過，在該執行個體用掉 30 點的啟動額度之後，額度餘額不得超過 144 點。如需每種執行個體容量大小的 CPU 額度餘額限額詳細資訊，請參閱額度表。

下表列出了在啟動或開始時所獲得的初始 CPU 額度配額，以及 vCPU 的數量。

執行個體類型	啟動額度	vCPU
t1.micro	15	1

執行個體類型	啟動額度	vCPU
t2.nano	30	1
t2.micro	30	1
t2.small	30	1
t2.medium	60	2
t2.large	60	2
t2.xlarge	120	4
t2.2xlarge	240	8

啟動額度限制

T2 Standard 執行個體可以獲得啟動額度的次數是有限的。預設的限制是所有 T2 Standard 執行個體共 100 次啟動或開始 (每個帳戶、每個區域、每個累計 24 小時期間合計)。例如，當 1 個執行個體在 24 小時期間內停止和啟動 100 次、當 100 個執行個體在 24 小時期間內啟動，或是當等於 100 次啟動的其他組合動作發生時，就會到達限制的次數。新帳戶的限制值可能較低，而根據您的使用量，此限制會隨時間提高。

> **Tip**
>
> 為確保工作負載隨時都能獲得所需的效能，請切換為 爆量效能執行個體的無限制模式，或考慮使用較大的執行個體大小。

啟動額度與獲得額度之間的差異

下表列出了啟動額度與獲得額度之間的差異。

	啟動額度	獲得的額度
額度獲得率	在啟動或開始時，T2 Standard 執行個體會獲得每 vCPU 30 點的啟動額度。	視執行個體的大小而定，每個 T2 執行個體每小時會持續獲得 (以毫秒級的解析度) 固定比率的 CPU 額度。如需每種

	啟動額度	獲得的額度
	如果 T2 執行個體從 `unlimited` 切換為 `standard`，在切換時不會獲得啟動額度。	執行個體容量大小所獲得的 CPU 額度詳細資訊，請參閱額度表。
額度獲得限制	獲得啟動額度的限制，是所有 T2 Standard 執行個體共 100 次啟動或開始 (每個帳戶、每個區域、每個累計 24 小時期間合計)。新帳戶的限制值可能較低，而根據您的使用量，此限制會隨時間提高。	T2 執行個體所累積的額度不能超過 CPU 額度餘額限額。如果 CPU 額度餘額到達其上限，則在到達限額之後所獲得的任何額度都將被捨棄。啟動額度不會列入此限額的計算。如需每種 T2 執行個體容量大小的 CPU 額度餘額限額詳細資訊，請參閱額度表。
額度使用	啟動額度會比獲得的額度先用掉。	獲得的額度只會在所有啟動額度用掉之後才使用。
額度過期	當 T2 Standard 執行個體正在執行時，啟動額度不會過期。如果 T2 Standard 執行個體停止，或是切換為 T2 Unlimited，所有的啟動額度都會消失。	當 T2 執行個體執行時，已經累積獲得的額度不會過期。當 T2 執行個體停止時，所有累積獲得的額度都會消失。

累積的啟動額度和獲得額度，會透過 CloudWatch 指標 `CPUCreditBalance` 來追蹤其數量。如需詳細資訊，請參閱 CloudWatch 指標表格中的 `CPUCreditBalance`。

爆量執行個體的標準模式範例

下列範例說明當執行個體設定為 `standard` 時的額度使用。

範例

- 範例 1：說明 T3 Standard 的額度使用
- 範例 2：說明 T2 Standard 的額度使用

範例 1：說明 T3 Standard 的額度使用

在此範例中，您會看到啟動為 `t3.nano` 的 `standard` 執行個體如何獲得、累積和用掉獲得的額度。您會看到額度餘額如何反映累積獲得的額度。

執行中的 t3.nano 執行個體每 24 小時會獲得 144 點。其額度餘額限制為獲得的 144 點。到達限額之後，獲得的新額度都將被捨棄。如需可獲得和累積的額度詳細資訊，請參閱額度表。

您可能會啟動 T3 Standard 執行個體，並立即使用它。或是您可能會啟動 T3 Standard 執行個體，然後讓該執行個體閒置幾天之後，才在其上執行應用程式。執行個體是否在使用中或處於閒置狀態，將決定額度是否會用掉或累積。如果執行個體在啟動後 24 小時的期間仍維持閒置，則額度餘額會達到其限制，即獲得的額度在累積後的上限。

此範例說明在啟動後 24 小時期間仍維持閒置的執行個體，並逐步說明 96 小時期間的 7 個階段、顯示獲得、累積、用掉和捨棄的比率，以及每個期間結束時的額度餘額值。

下列的工作流程參考了圖表上的編號點：

P1 – 在圖表上的第 0 小時，執行個體會啟動為 standard，並立即開始獲得額度。該執行個體從啟動後即維持閒置，其 CPU 使用率為 0%，而且沒有用掉額度。所有為使用的額度都會累計到額度餘額。前 24 小時，CPUCreditUsage 為 0，而 CPUCreditBalance 值達到其最大值 144。

P2 – 接下來 12 小時，CPU 使用率為 2.5%，低於 5% 基準。執行個體獲得的額度超出其用掉的額度，但 CPUCreditBalance 值不得超出其最大值，即 144 點。任何獲得的額度若超出限制，即會遭到捨棄。

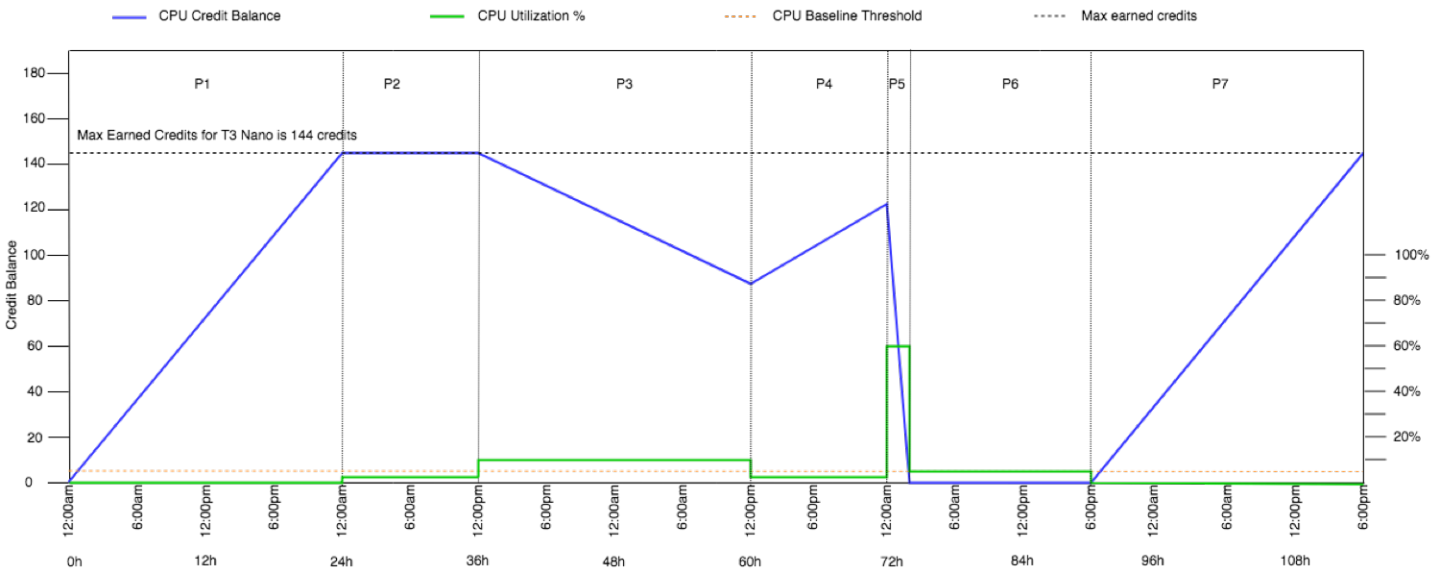
P3 – 接下來 24 小時，CPU 使用率為 7% (高於基準)，其需要用掉 57.6 點。執行個體用掉的額度超過獲得的額度，而且 CPUCreditBalance 值減少至 86.4 點。

P4 – 接下來 12 小時，CPU 使用率下降至 2.5% (低於基準)，其需要用掉 36 點。同時執行個體獲得 72 點。執行個體獲得的額度超過用掉的額度，而且 CPUCreditBalance 值增加至 122 點。

P5 – 接下來 2 小時，執行個體的 CPU 使用率爆量至 60%，並耗盡整個 CPUCreditBalance 值，即 122 點。在此期間結束時，由於 CPUCreditBalance 為零，CPU 使用率被迫下降至基準使用率，即 5%。處於基準時，執行個體獲得與其用掉一樣多的額度。

P6 – 接下來 14 小時，CPU 使用率為 5% (基準)。執行個體獲得與其用掉一樣多的額度。CPUCreditBalance 值仍為 0。

P7 – 在此範例的最後 24 小時，執行個體閒置且 CPU 使用率為 0%。在此期間，執行個體獲得 144 點，其會累積在 CPUCreditBalance。



範例 2：說明 T2 Standard 的額度使用

在此範例中，會示範做為 t2.nano 啟動的 standard 執行個體，如何獲得、累積和用掉啟動與獲得額度。範例中將會顯示額度餘額不只反映累積獲得的額度，也會反映累積的啟動額度。

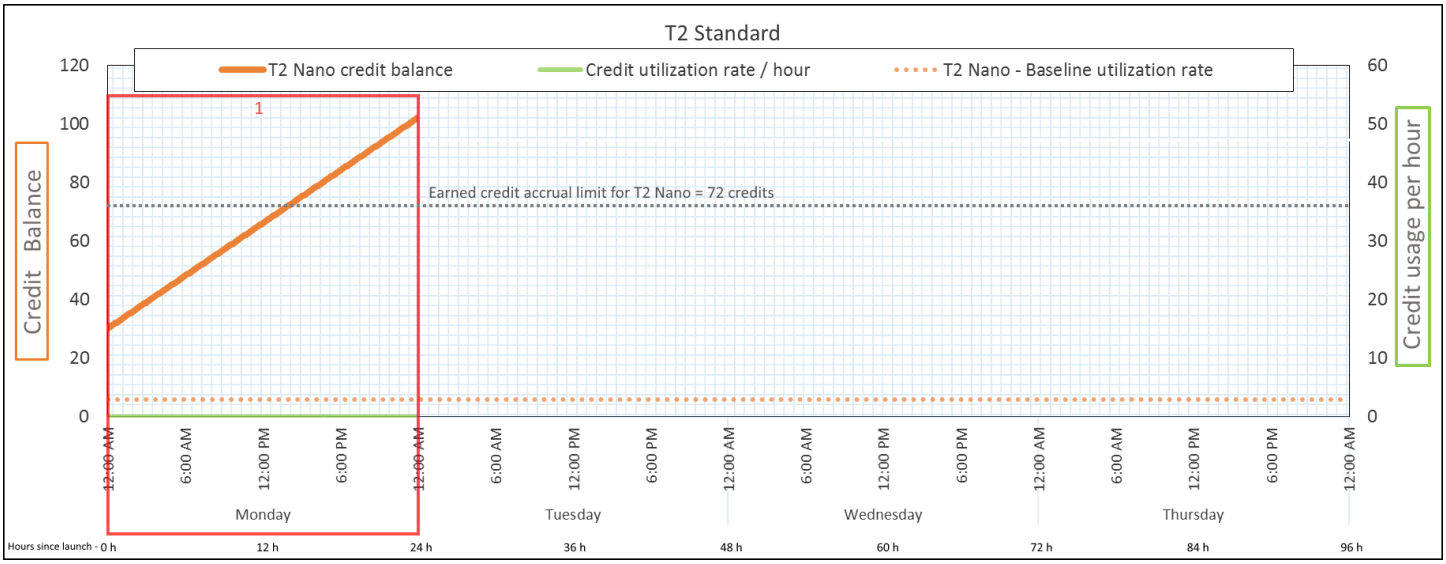
t2.nano 執行個體會在啟動時獲得 30 點的啟動額度，而且每 24 小時可獲得 72 點。其額度餘額上限為 72 點獲得的額度；啟動額度不會列入此限額的計算。到達限額之後，獲得的新額度都將被捨棄。如需可獲得和累積的額度詳細資訊，請參閱額度表。如需限制的詳細資訊，請參閱 啟動額度限制。

您可能會啟動 T2 Standard 執行個體，並立即使用它。或是您可能會啟動 T2 Standard 執行個體，然後讓該執行個體閒置幾天之後，才在其上執行應用程式。執行個體是否在使用中或處於閒置狀態，將決定額度是否會用掉或累積。如果執行個體在啟動後 24 小時的期間仍維持閒置，額度餘額會顯示為超過其限額，因為該餘額同時反映了累積獲得的額度和累積的啟動額度。不過，在使用 CPU 之後，會先用掉啟動額度。之後，此限額一律反映可累積獲得額度的上限。

此範例說明在啟動後 24 小時期間仍維持閒置的執行個體，並逐步說明 96 小時期間的 7 個階段、顯示獲得、累積、用掉和捨棄的比率，以及每個期間結束時的額度餘額值。

期間 1：1 – 24 小時

在圖表上的第 0 小時，T2 執行個體會啟動為 standard，並立即獲得 30 點的啟動額度。該執行個體會在執行狀態中獲得額度。該執行個體從啟動後即維持閒置，其 CPU 使用率為 0%，而且沒有用掉額度。所有為使用的額度都會累計到額度餘額。在啟動後大約 14 小時，其額度餘額為 72 點 (30 點的啟動額度 + 42 點獲得的額度)，等於該執行個體可以在 24 小時中所獲得的額度。在啟動 24 小時後，額度餘額超過 72 點，因為有未使用的啟動額度累計到額度餘額—額度餘額為 102 點 (30 點的啟動額度 + 72 點獲得的額度)。



指標	Value
CPU 額度使用率	每 24 小時 0 點 (0% CPU 使用率)
額度獲得率	每 24 小時 72 點
額度捨棄率	每 24 小時 0 點
額度餘額	102 點 (30 點的啟動額度 + 72 點獲得的額度)

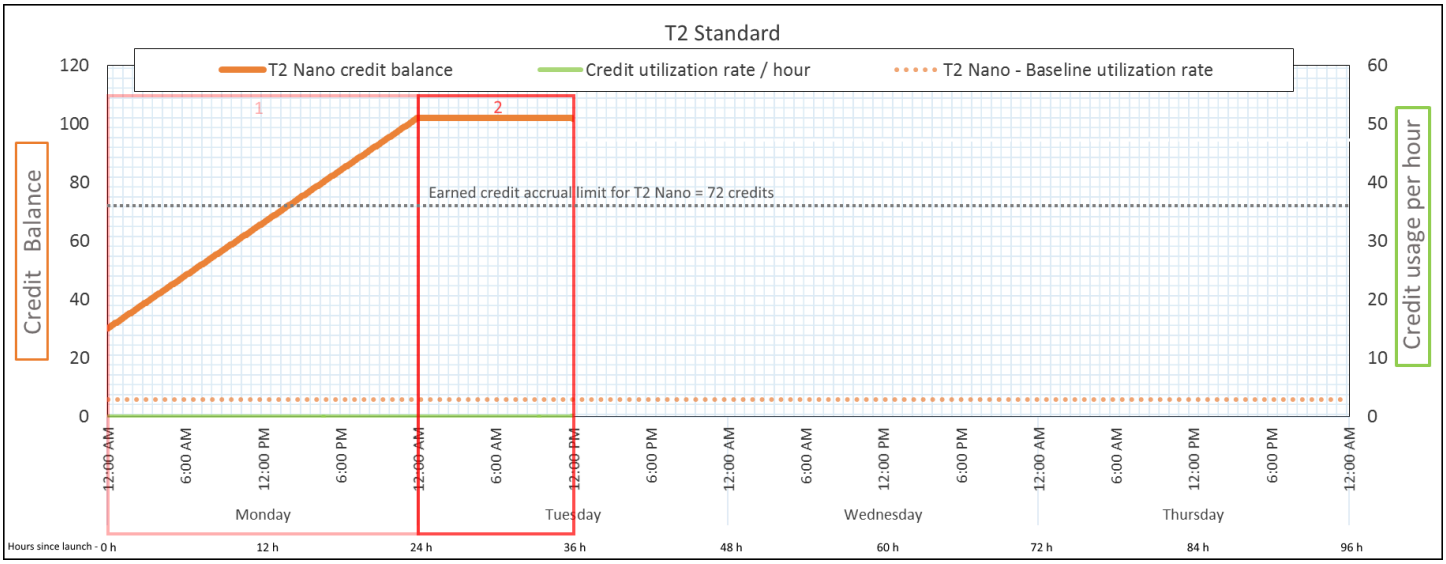
結論

如果啟動之後未使用 CPU，執行個體累積的額度，會比 24 小時中可獲得的更多 (30 點的啟動額度 + 72 點獲得的額度 = 102 點)。

在實際的情境中，EC2 執行個體會在啟動和執行時用掉少數的額度，這可防止餘額到達本範例中的理論值上限。

期間 2：25 – 36 小時

在接下來的 12 小時內，該執行個體會繼續保持閒置和獲得額度，但額度餘額不會增加。餘額會穩定地保持在 102 點 (30 點的啟動額度 + 72 點獲得的額度)。額度餘額已到達其上限 (72 點獲得的額度)，因此新獲得的額度將被捨棄。



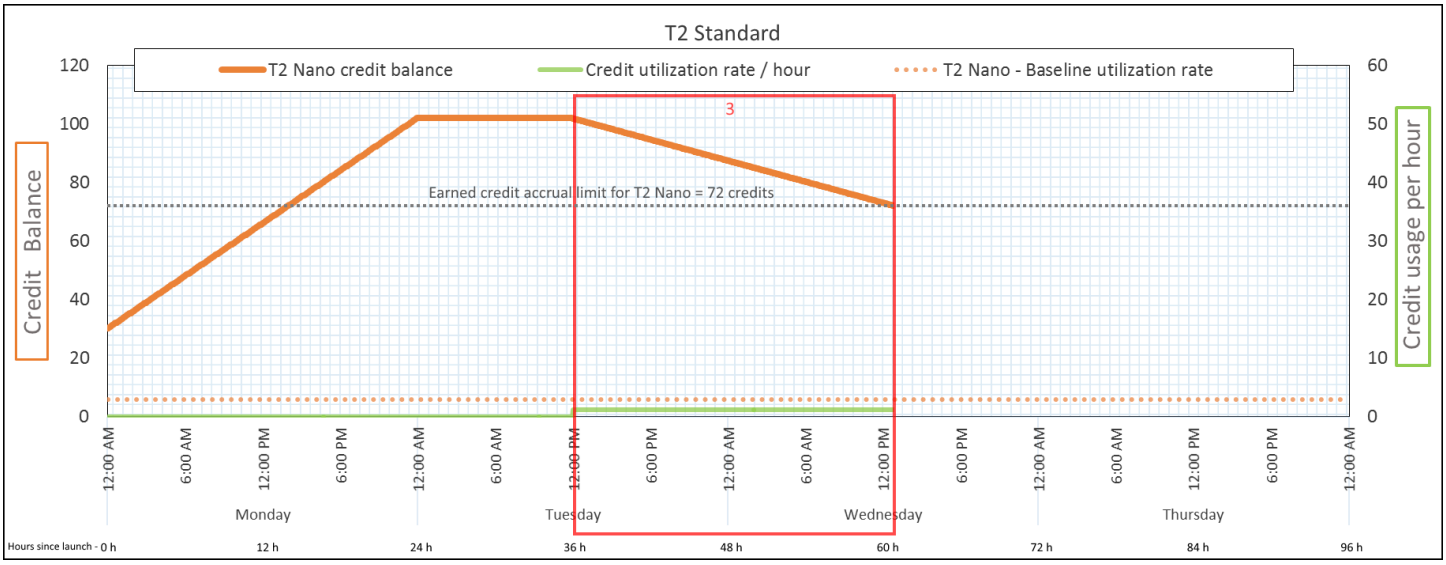
指標	Value
CPU 額度使用率	每 24 小時 0 點 (0% CPU 使用率)
額度獲得率	每 24 小時 72 點 (每小時 3 點)
額度捨棄率	每 24 小時 72 點 (100% 的額度獲得率)
額度餘額	102 點 (30 點的啟動額度 + 72 點獲得的額度) – 餘額未變

結論

執行個體會持續地獲得額度，但如果額度餘額已到達其上限，就不能再累積更多獲得的額度。到達限額之後，獲得的新額度都將被捨棄。啟動額度不會列入額度餘額限額的計算。如果餘額中包含累計的啟動額度，則餘額看起來會像是超過限額。

期間 3：37 – 61 小時

在接下來的 25 個小時期間，執行個體使用 2% 的 CPU，這需要 30 點。在同一個期間，執行個體獲得 75 點，但額度餘額減少了。餘額減少是由於累積的啟動額度會先用掉，而額度餘額已到達 72 點獲得額度的限制，因此新獲得的額度被捨棄。



指標	Value
CPU 額度使用率	每 24 小時 28.8 點 (每小時 1.2 點、2% CPU 使用率、40% 額度獲得率) – 25 小時 30 點
額度獲得率	每 24 小時 72 點
額度捨棄率	每 24 小時 72 點 (100% 的額度獲得率)
額度餘額	72 點 (30 點的啟動額度已用掉；72 點獲得的額度尚未用掉)

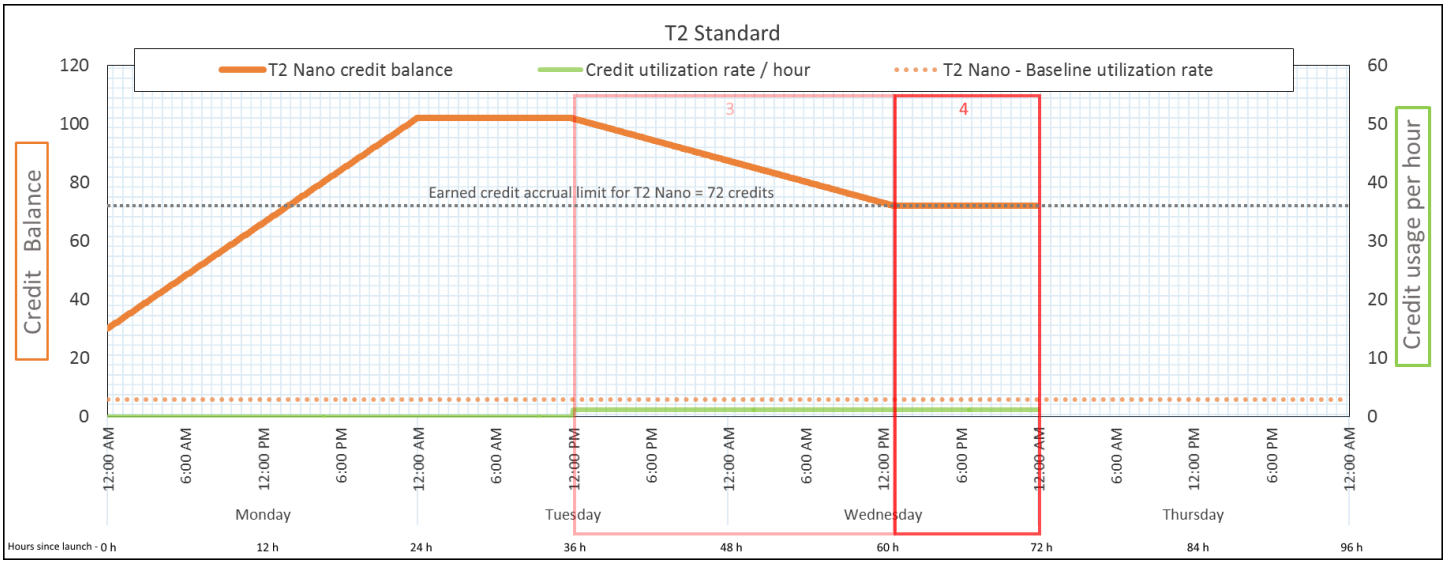
結論

執行個體會先用掉啟動額度，然後才會使用獲得的額度。啟動額度不會列入額度限額的計算。在啟動額度用掉之後，餘額就永遠不會高於 24 小時內可獲得的額度。此外，當執行個體執行時，就無法再獲得更多啟動額度。

期間 4：62 – 72 小時

在接下來的 11 個小時期間，執行個體使用 2% 的 CPU，這需要 13.2 點。這和前一個期間中的 CPU 使用率相同，但是餘額沒有減少，而是維持在 72 點。

餘額沒有減少是因為額度獲得率高於額度使用率。在此期間執行個體用掉 13.2 點，但也獲得了 33 點。但是，餘額的上限為 72 點，因此任何獲得的額度只要超過此限額，就會被捨棄。餘額會穩定地保持在 72 點，這跟期間 2 維持在 102 點不同，因為沒有累積的啟動額度。



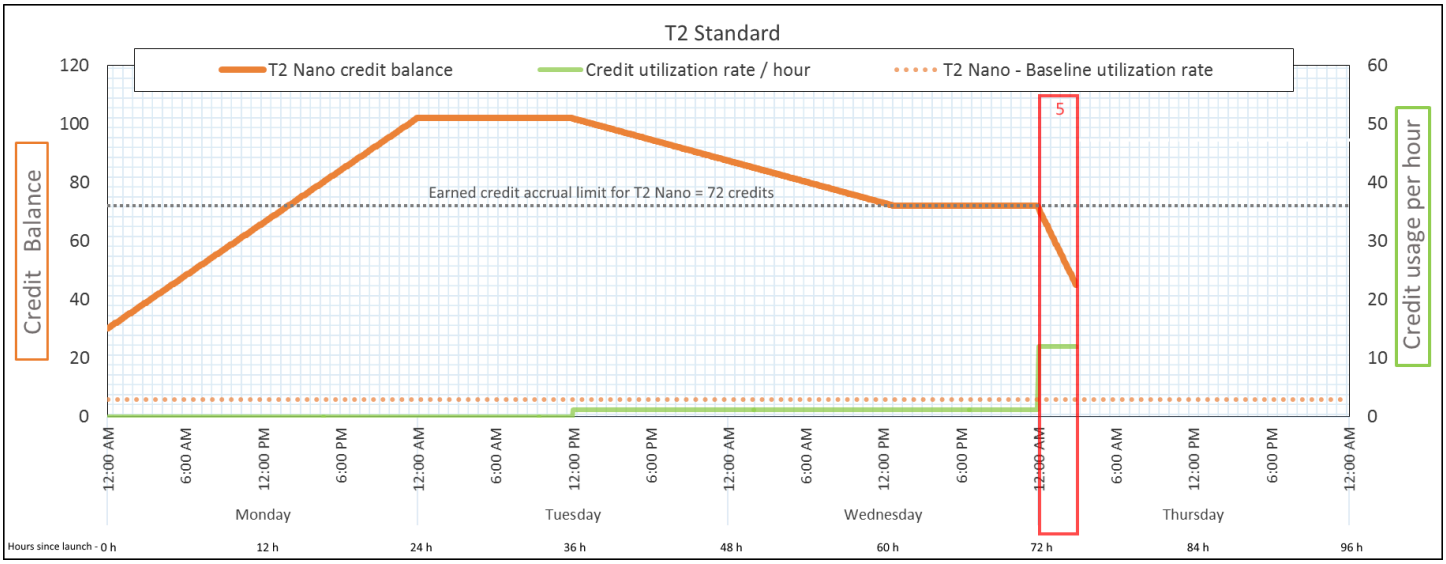
指標	Value
CPU 額度使用率	每 24 小時 28.8 點 (每小時 1.2 點、2% CPU 使用率、40% 額度獲得率) – 11 小時 13.2 點
額度獲得率	每 24 小時 72 點
額度捨棄率	每 24 小時 43.2 點 (60% 的額度獲得率)
額度餘額	72 點 (0 點的啟動額度 + 72 點獲得的額度) – 餘額到達上限

結論

在啟動額度用掉之後，額度餘額的上限，會取決於執行個體在 24 小時內可獲得的額度。如果執行個體獲得的額度比用掉的更多，新獲得的額度只要超過限額，就會被捨棄。

期間 5：73 – 75 小時

在接下來的 3 個小時期間，執行個體運作的需求會突增為 20% 的 CPU 使用率，這需要 36 點。在同樣的 3 小時期間，執行個體會獲得 9 點，這會使得額度淨餘額減少 27 點。在 3 小時期間結束時，額度餘額為 45 點累積獲得的額度。



指標	Value
CPU 額度使用率	每 24 小時 288 點 (每小時 12 點、20% CPU 使用率、400% 額度獲得率) – 3 小時 36 點
額度獲得率	每 24 小時 72 點 (3 個小時 9 點)
額度捨棄率	每 24 小時 0 點
額度餘額	45 點 (先前的餘額 (72) - 用掉的額度 (36) + 獲得的額度 (9)) – 在每 24 小時獲得 216 點的額度獲得率之下，但額度餘額減少了，額度餘額減少 (使用率 288/24 + 獲得率 72/24 = 餘額減少率 216/24)

結論

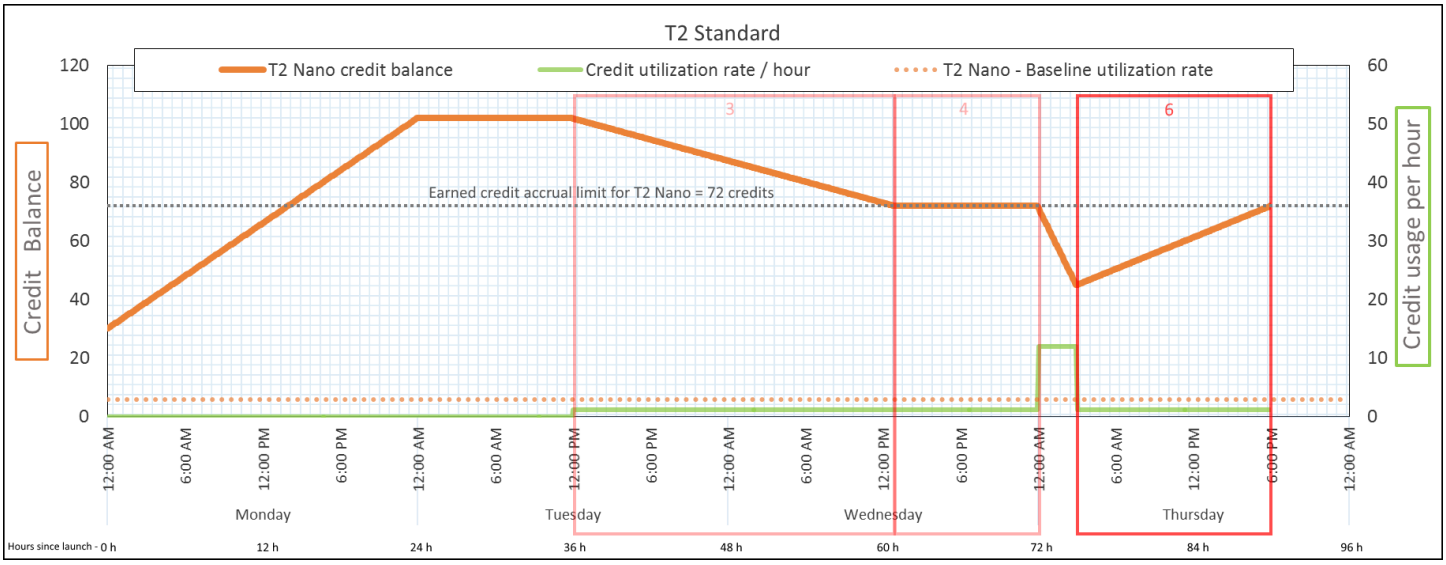
如果執行個體用掉的額度比獲得的多，其額度餘額會減少。

期間 6：76 – 90 小時

在接下來的 15 個小時期間，執行個體使用 2% 的 CPU，這需要 18 點。此期間的 CPU 使用率，和期間 3 與期間 4 的相同。不過，餘額在此期間是增加的，而期間 3 和期間 4 的餘額則分別是減少和持平。

在期間 3 中用掉了累積的啟動額度，並捨棄超過額度限額的所有獲得額度，因此使額度餘額減少。在期間 4 中，執行個體用掉的額度比獲得的少。捨棄超過額度限額的所有獲得額度，因此額度餘額持平，維持在 72 點的最高值。

在此期間中沒有累積的啟動額度，而餘額中累積獲得的額度低於限額。沒有任何獲得的額度被捨棄。此外，執行個體獲得的額度比用掉的多，因此使額度餘額增加。



指標	Value
CPU 額度使用率	每 24 小時 28.8 點 (每小時 1.2 點、2% CPU 使用率、40% 額度獲得率) – 15 小時 18 點
額度獲得率	每 24 小時 72 點 (15 個小時 45 點)
額度捨棄率	每 24 小時 0 點
額度餘額	72 點 (餘額以每小時 43.2 點的速度增加 – 變動率 = 使用率 28.8/24 + 獲得率 72/24)

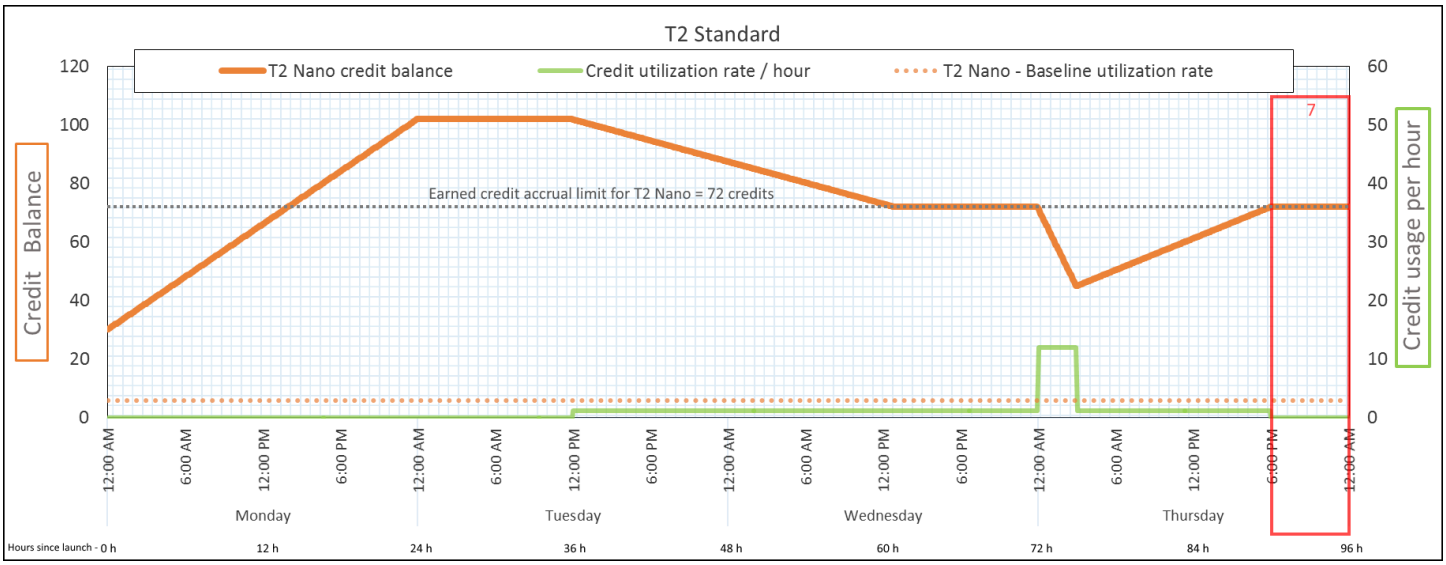
結論

如果執行個體用掉的額度比獲得的少，其額度餘額會增加。

期間 7：91 – 96 小時

接下來的 6 小時，執行個體會維持閒置—其 CPU 使用率為 0%—而且沒有用掉額度。此 CPU 使用率與期間 2 的相同，但餘額並未持平維持於 102 點—而是持平保持在 72 點，這是執行個體的額度餘額上限。

在期間 2 中，額度餘額包含 30 點累積的啟動額度。在期間 3 中用掉了啟動額度。執行中的執行個體無法再獲得更多的啟動額度。在到達其額度餘額上限後，任何獲得的額度只要超過此限額，就會被捨棄。



指標	Value
CPU 額度使用率	每 24 小時 0 點 (0% CPU 使用率)
額度獲得率	每 24 小時 72 點
額度捨棄率	每 24 小時 72 點 (100% 的額度獲得率)
額度餘額	72 點 (0 點的啟動額度、72 點獲得的額度)

結論

執行個體會持續地獲得額度，但如果額度餘額已到達其上限，就不能再累積更多獲得的額度。到達限額之後，獲得的新額度都將被捨棄。額度餘額的上限，會取決於執行個體在 24 小時內可獲得的額度。如需額度餘額上限的詳細資訊，請參閱額度表。

設定爆量效能執行個體

啟動、監控及修改爆量效能執行個體 (T 執行個體) 的步驟類似。主要差異為它們啟動時的預設額度規格。

每個 T 執行個體系列都有下列預設額度規格：

- T4g、T3a 和 T3 執行個體以 `unlimited` 模式啟動
- 專用執行個體中的 T3 執行個體只能以 `standard` 狀態啟動
- T2 執行個體啟動為 `standard`

可變更帳戶的預設額度規格。

任務

- 在啟動時設定抵用金規格
- 設定 Auto Scaling 群組，將抵用金規格設定為 unlimited
- 管理爆量效能執行個體的抵用金規格
- 管理帳戶的預設抵用金規格

在啟動時設定抵用金規格

您可使用 `unlimited` 或 `standard` 的抵用金規格來啟動 T 執行個體。

下列程序說明如何使用 EC2 主控台或 AWS CLI。如需有關使用 Auto Scaling 群組的資訊，請參閱 設定 Auto Scaling 群組，將抵用金規格設定為 unlimited。

Console

在啟動時設定執行個體的抵用金規格

1. 請遵循該程序來啟動執行個體。
2. 在 Instance type (執行個體類型) 中，選取 T 執行個體類型。
3. 展開 Advanced Details (進階詳細資訊)。對於抵用金規格，選取抵用金規格。
4. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。

AWS CLI

在啟動時設定執行個體的抵用金規格

請使用 `--credit-specification` 選項並搭配 run-instances 命令。

```
--credit-specification CpuCredits=unlimited
```

PowerShell

在啟動時設定執行個體的抵用金規格

使用 New-EC2Instance cmdlet 搭配 `-CreditSpecification_CpuCredit` 參數。

```
-CreditSpecification_CpuCredit unlimited
```

設定 Auto Scaling 群組，將抵用金規格設定為 unlimited

當 T 執行個體啟動或開始時，需要 CPU 的額度來提供流暢的引導體驗。如果您使用 Auto Scaling 群組來啟動您的執行個體，我們建議將您的執行個體設定為 `unlimited`。如此在由 Auto Scaling 群組自動啟動或重新啟動時，這些執行個體即可使用剩餘額度。請使用剩餘額度來防止效能受限。

建立啟動範本

您必須使用「啟動範本」，在 Auto Scaling 群組中將執行個體啟動為 `unlimited`。啟動組態不支援將執行個體啟動為 `unlimited`。

Console

建立設定抵用金規格的啟動範本

1. 請遵循「Amazon EC2 Auto Scaling 使用者指南」中的使用進階設定建立啟動範本。
2. 在啟動範本內容中，針對執行個體類型，選擇執行個體大小。
3. 若要在 Auto Scaling 群組中以 `unlimited` 的形式啟動執行個體，請在進階詳細資料下，針對額度規格選擇無限制。
4. 在您完成定義啟動範本參數時，選擇 Create launch template (建立啟動範本)。

AWS CLI

建立設定抵用金規格的啟動範本

使用 create-launch-template 命令。

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --version-description FirstVersion \
    --launch-template-data CreditSpecification={CpuCredits=unlimited}
```

PowerShell

建立設定抵用金規格的啟動範本

使用 New-EC2LaunchTemplate cmdlet。定義啟動範本資料的抵用金規格，如下所示。

```
$creditSpec = New-Object Amazon.EC2.Model.CreditSpecificationRequest
$creditSpec.CpuCredits = "unlimited"
$launchTemplateData = New-Object Amazon.EC2.Model.RequestLaunchTemplateData
$launchTemplateData.CreditSpecification = $creditSpec
```

使用 Auto Scaling 群組與啟動範本建立關聯

若要建立啟動範本與 Auto Scaling 群組的關聯，請使用啟動範本來建立 Auto Scaling 群組，或是將啟動範本加入現有的 Auto Scaling 群組。

Console

使用啟動範本建立 Auto Scaling 群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面上方的導覽列中，選取與您用來建立啟動範本相同的區域。
3. 在導覽窗格中，選擇 Auto Scaling Groups (AS 群組) 與 Create Auto Scaling group (建立 AS 群組)。
4. 選取 Launch Template (啟動範本)、選取您的啟動範本，然後選取 Next Step (下一步)。
5. 填寫 Auto Scaling 群組的欄位。檢閱完您在 Review page (檢閱頁面) 上的組態設定後，請選擇 Create Auto Scaling group (建立 Auto Scaling 群組)。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南 中的使用啟動範本建立 Auto Scaling 群組。

將啟動範本新增至現有的 Auto Scaling 群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面上方的導覽列中，選取與您用來建立啟動範本相同的區域。
3. 在導覽窗格中，選擇 Auto Scaling Groups (AS 安全群組)。
4. 從 Auto Scaling 群組清單中，選取 Auto Scaling 群組，然後選取 Actions (動作)、Edit (編輯)。
5. 在 Details (詳細資訊) 標籤上，對於 Launch Template (啟動範本)，選擇一個啟動範本，然後選擇 Save (儲存)。

AWS CLI

使用啟動範本建立 Auto Scaling 群組

使用 create-auto-scaling-group 命令，並指定 `--launch-template` 參數。

將啟動範本新增至現有的 Auto Scaling 群組

使用 update-auto-scaling-group 命令，並指定 `--launch-template` 參數。

PowerShell

使用啟動範本建立 Auto Scaling 群組

使用 New-ASAutoScalingGroup cmdlet 並指定 `-LaunchTemplate_LaunchTemplateId` 或 `-LaunchTemplate_LaunchTemplateName` 參數。

將啟動範本新增至現有的 Auto Scaling 群組

使用 Update-ASAutoScalingGroup cmdlet 並指定 `-LaunchTemplate_LaunchTemplateId` 或 `-LaunchTemplate_LaunchTemplateName` 參數。

管理爆量效能執行個體的抵用金規格

您可以在 `unlimited` 與 `standard` 之間，隨時將執行中或已停止 T 執行個體的額度規格進行切換。

請注意，在 `unlimited` 模式中，執行個體可能會使用剩餘抵用金，這可能會產生額外費用。如需詳細資訊，請參閱剩餘額度可能會產生費用。

Console

管理 T 執行個體的抵用金規格

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. (選用) 選取執行個體。在詳細資訊索引標籤上，尋找抵用金規格。此值為 `unlimited` 或 `standard`。
4. (選用) 若要同時修改多個執行個體的抵用金規格，請選取所有執行個體。
5. 選擇動作、執行個體設定、變更額度規格。只有在您已選取 T 執行個體的情況下，才會啟用此選項。
6. 對於無限制模式，選取或清除每個執行個體 ID 旁的核取方塊。

AWS CLI

取得 T 執行個體的抵用金規格

使用 describe-instance-credit-specifications 命令。若您未指定執行個體 ID，則會傳回具有 `unlimited` 抵用金規格的所有執行個體。傳出也會包含先前使用 `unlimited` 抵用金規格設定的執行個體。例如，如果您將 T3 執行個體的大小重新調整為 M4 執行個體，並且其設定為 `unlimited` 時，Amazon EC2 會傳回 M4 執行個體。

```
aws ec2 describe-instance-credit-specifications \
    --instance-id i-1234567890abcdef0 \
    --query InstanceCreditSpecifications[].CpuCredits \
    --output text
```

以下為範例輸出。

```
unlimited
```

設定 T 執行個體的抵用金規格

使用 modify-instance-credit-specification 命令。

```
aws ec2 modify-instance-credit-specification \
    --region us-east-1 \
    --instance-credit-specification
 "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

PowerShell

取得 T 執行個體的抵用金規格

使用 Get-EC2CreditSpecification cmdlet。

```
(Get-EC2CreditSpecification `
    -InstanceId i-1234567890abcdef0).CpuCredits
```

以下為範例輸出。

```
unlimited
```

設定 T 執行個體的抵用金規格

使用 Edit-EC2InstanceCreditSpecification cmdlet。

```
Edit-EC2InstanceCreditSpecification `
    -Region us-east-1 `
    -InstanceCreditSpecification @({InstanceId="i-1234567890abcdef0"
 CpuCredits="unlimited"})
```

管理帳戶的預設抵用金規格

每個 T 執行個體系列都有預設額度規格。您可以在每個 AWS 區域的帳戶層級變更每個 T 執行個體系列的預設點數規格。預設抵用金規格的有效值為 `unlimited` 和 `standard`。

如果您使用 EC2 主控台中的啟動執行個體精靈來啟動執行個體，您針對額度規格選取的值則會覆寫帳戶層級的預設額度規格。如果您使用 AWS CLI 啟動執行個體，帳戶中的所有新 T 執行個體都會使用預設點數規格啟動。現有執行中或已停止執行個體的額度規格不受影響。

考量事項

執行個體系列的預設額度規格只能在連續 5 分鐘的期間內修改一次，而在連續 24 小時內最多可修改四次。

Console

管理預設抵用金規格

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 若要變更 AWS 區域，請使用頁面右上角的區域選擇器。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇預設點數規格。
5. 選擇 Manage (管理)。
6. 針對每個執行個體系列選擇 Unlimited (無限制) 或 Standard (標準)，然後選擇 Update (更新)。

AWS CLI

取得預設抵用金規格

使用 get-default-credit-specification 命令。

```
aws ec2 get-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --query InstanceFamilyCreditSpecifications[].CpuCredits \
    --output text
```

以下為範例輸出。

```
standard
```

設定預設抵用金規格

使用 modify-default-credit-specification 命令。下列範例將值設定為 `unlimited`。

```
aws ec2 modify-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --cpu-credits unlimited
```

PowerShell

取得預設抵用金規格

使用 Get-EC2DefaultCreditSpecification cmdlet。

```
(Get-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2).CpuCredits
```

以下為範例輸出。

```
standard
```

設定預設抵用金規格

使用 Edit-EC2DefaultCreditSpecification cmdlet。下列範例將值設定為 unlimited。

```
Edit-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2 `
    -CpuCredit unlimited
```

監控爆量執行個體的 CPU 額度

EC2 會將指標傳送至 Amazon CloudWatch。您可以在 CloudWatch 主控台的 Amazon EC2 每個執行個體指標中查看 CPU 點數指標，或使用 AWS CLI 列出每個執行個體的指標。如需詳細資訊，請參閱您的執行個體可用的 CloudWatch 指標。

目錄

- 高載效能執行個體的其他 CloudWatch 指標
- 計算 CPU 額度用量

高載效能執行個體的其他 CloudWatch 指標

爆量效能執行個體具有這些其他的 CloudWatch 指標，每五分鐘會更新一次：

- CPUCreditUsage – 在衡量期間用掉的 CPU 額度。
- CPUCreditBalance – 執行個體已累積的 CPU 額度。當 CPU 效能瞬間突增，以及 CPU 額度用掉的速度比獲得的快時，此額度餘額會用完。
- CPUSurplusCreditBalance – 當 CPUCreditBalance 值為 0 時，為維持 CPU 使用率所用掉的多餘 CPU 額度。
- CPUSurplusCreditsCharged – 超過 24 小時期間可獲得最高 CPU 額度的多餘 CPU 額度，因此會產生額外的費用。

最後兩種指標只適用於設定為 unlimited 的執行個體。

下表說明高載效能執行個體的 CloudWatch 指標。如需詳細資訊，請參閱您的執行個體可用的 CloudWatch 指標。

指標	描述
CPUCreditUsage	執行個體為 CPU 使用率花費的 CPU 額度數量。一個 CPU 額度等於一個 vCPU 以 100% 使用率執行 1 分鐘，或同等的 vCPU、使用率與時間的組合 (例如，一個 vCPU 以 50% 使用率執行 2 分鐘，或兩個 vCPU 以 25% 使用率執行 2 分鐘)。 CPU 額度指標僅提供 5 分鐘頻率。如果您要指定大於 5 分鐘的期間，請使用 Sum 統計資訊代替 Average 統計資訊。 單位：額度 (vCPU-分鐘)
CPUCreditBalance	自執行個體啟動或開始後，累積獲得的 CPU 額度數量。如果是 T2 Standard，CPUCreditBalance 也包含已產生的啟動額度。 獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。額度餘額有最大值限制，它取決於執行個體大小。到達限制之後，任何獲得的新額度都會遭到捨棄。如果是 T2 Standard，啟動額度不會計入此限制。 CPUCreditBalance 中的額度可供執行個體支付以大幅提升並超越基準 CPU 使用率。 當執行個體執行時，CPUCreditBalance 中的額度不會過期。當 T4g、T3a 或 T3 執行個體停止時，CPUCreditBalance 值會持續保存七天。因此，所有累積的額度都會消失。當 T2 執行個體停止時，CPUCreditBalance 值不會持續保存，而且所有累積的額度都將消失。 CPU 額度指標僅提供 5 分鐘頻率。 單位：額度 (vCPU-分鐘)
CPUSurplusCreditBalance	當 unlimited 執行個體的 CPUCreditBalance 值為 0 時，該執行個體已支出的剩餘額度數量。

指標	描述
	CPUSurplusCreditBalance 值由獲得的 CPU 額度支付。如果剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量，超過最大值的支出剩餘額度將必須負擔額外的費用。 單位：額度 (vCPU-分鐘)
CPUSurplusCreditsCharged	若支出剩餘額度數量未由獲得的 CPU 額度付清，會產生額外的費用。 發生以下任何情況時，將收取支出剩餘額度的費用。 • 支出剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量。在小時結束時，將收取超過最大值的支出剩餘額度的費用。 • 執行個體已停止或終止。 • 執行個體從 unlimited 切換至 standard。 單位：額度 (vCPU-分鐘)

計算 CPU 額度用量

執行個體的 CPU 額度用量，會透過上表中所說明的執行個體 CloudWatch 指標計算。

Amazon EC2 每五分鐘會將指標傳送給 CloudWatch。在任何時點參考指標的 prior 值，即表示該指標先前的值 (五分鐘前傳送的值)。

計算標準執行個體的 CPU 額度使用

- 當用掉的額度，少於先前五分鐘間隔中所獲得的額度時，如果 CPU 使用率低於基準，則 CPU 額度餘額會增加。
- 當用掉的額度，多於先前五分鐘間隔中所獲得的額度時，如果 CPU 使用率高於基準，則 CPU 額度餘額會減少。

在數學上，這可以透過下列方程式計算：

Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) -
 CPUCreditUsage]
```

執行個體的大小，決定了執行個體每小時可獲得的額度，以及執行個體額度餘額可累積獲得額度的上限。如需每小時所獲得的額度、每種執行個體大小的額度餘額限額資訊，請參閱額度表。

範例

此範例使用 t3.nano 執行個體。若要計算執行個體的 CPUCreditBalance 值，請使用前述的方程式，如下所示：

- CPUCreditBalance – 要計算的最新額度餘額。
- prior CPUCreditBalance – 5 分鐘前的額度餘額。在此範例中，執行個體累積了 2 種額度。
- Credits earned per hour – 一個 t3.nano 執行個體每小時會獲得 6 點。
- 5/60 – 代表 CloudWatch 指標發佈之間的 5 分鐘間隔。將每小時獲得的額度乘上 5/60 (5 分鐘)，以算出該執行個體在過去 5 分鐘內獲得的額度。t3.nano 執行個體每 5 分鐘會獲得 0.5 點。
- CPUCreditUsage – 在過去 5 分鐘內執行個體所用掉的額度。在此範例中，執行個體過去 5 分鐘內用掉了 1 點。

使用這些值，可以計算出 CPUCreditBalance 值：

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

計算無限制執行個體的 CPU 額度使用

當爆量效能執行個體的效能需要爆發超過基準時，一律會先用掉累積的額度，然後再使用剩餘額度。當該執行個體累積的 CPU 額度餘額用完時，可以使用剩餘額度，讓 CPU 在所需的期間大幅提升。如果 CPU 的使用率掉到基準以下，執行個體一律會先償還剩餘額度，然後才能累積獲得的額度。

在下列的方程式中，我們使用 Adjusted balance 一詞，來代表這個 5 分鐘間隔內所發生的動作。我們會使用此值來算出 CPUCreditBalance 和 CPUSurplusCreditBalance CloudWatch 指標的值。

Example

```
Adjusted balance = [prior CPUCreditBalance - prior CPUSurplusCreditBalance] + [Credits
 earned per hour * (5/60) - CPUCreditUsage]
```

0 的 Adjusted balance 值代表該執行個體為了突增的效能，已經用完所有獲得的額度，而且未使用剩餘額度。因此，CPUCreditBalance 和 CPUSurplusCreditBalance 都會設定為 0。

正的 Adjusted balance 值表示該執行個體已累積獲得的額度，而且已償還先前的剩餘額度 (如果有的話)。因此，Adjusted balance 值會指派給 CPUCreditBalance，而 CPUSurplusCreditBalance 會設定為 0。執行個體的大小，會決定執行個體所能累積的額度上限。

Example

```
CPUCreditBalance = min [max earned credit balance, Adjusted balance]
CPUSurplusCreditBalance = 0
```

負的 Adjusted balance 值代表執行個體已經用完所有累積獲得的額度，而且也因為突增的效能用掉了剩餘額度。因此，Adjusted balance 值會指派給 CPUSurplusCreditBalance，而 CPUCreditBalance 會設為 0。再說一次，執行個體的大小，會決定執行個體所能累積的額度上限。

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]
CPUCreditBalance = 0
```

如果用掉的剩餘額度超過執行個體可累積的最高額度，則剩餘額度餘額會設定為最高額度，如先前的方程式中所示。剩下的剩餘額度會收取費用 (以 CPUSurplusCreditsCharged 指標表示)。

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

最後，當執行個體終止時，透過 CPUSurplusCreditBalance 追蹤的所有剩餘額度都會計費。如果執行個體從 unlimited 切換為 standard，也會針對所有剩下的 CPUSurplusCreditBalance 收費。

使用 GPU 執行個體加速效能

GPU 執行個體可讓您存取含數千個運算核心的 NVIDIA GPU。您可以使用這些執行個體，藉由運用 CUDA 或開放運算語言 (OpenCL) 平行運算架構來加速科學、工程和算圖應用程式。您也可以將它們用於圖形應用程式，包括遊戲串流、3D 應用程式串流和其他圖形工作負載。

您必須先安裝適當的驅動程式，才能啟用或最佳化 GPU 執行個體，如下所示：

- 若要在已連接 NVIDIA GPU 的執行個體 (例如 P3 或 G4dn 執行個體) 安裝 NVIDIA 驅動程式，請參閱 NVIDIA 驅動程式。
- 若要在已連接 AMD GPU 的執行個體 (例如 G4ad 執行個體) 安裝 AMD 驅動程式，請參閱 AMD 驅動程式。

目錄

- 在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式
- 在 Amazon EC2 執行個體最佳化 GPU 設定
- 在 G4ad Linux 執行個體設定雙 4K 顯示器
- 開始使用 GPU 加速執行個體

在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式

若要在具有 NVIDIA GPU 的基於 GPU 的執行個體啟動 GRID 虛擬應用程式 (預設啟用 NVIDIA GRID 虛擬工作站)，您必須定義驅動程式的產品類型。您使用的程序取決於執行個體的作業系統。

Linux 執行個體

在 Linux 執行個體上啟用 GRID 虛擬應用程式

1. 從提供的範本檔案建立 `/etc/nvidia/gridd.conf` 檔案。

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. 在您偏好的文字編輯器中開啟 `/etc/nvidia/gridd.conf` 檔案。
3. 找到 `FeatureType` 行，並將其設定為等於 `0`。然後新增一行 `IgnoreSP=TRUE`。

```
FeatureType=0 IgnoreSP=TRUE
```

4. 儲存檔案並結束。
5. 重新開機執行個體，套用新的組態。

```
[ec2-user ~]$ sudo reboot
```

Windows 執行個體

在 Windows 執行個體上啟用 GRID 虛擬應用程式

1. 開啟 regedit.exe 來開啟登錄編輯程式。
2. 導覽至 HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing。
3. 開啟右窗格中的內容選單 (按一下滑鼠右鍵)，然後選擇 New (新增)、DWORD。
4. 針對 Name (名稱)，輸入 FeatureType 然後輸入 Enter。
5. 開啟 FeatureType 上的內容選單 (按一下滑鼠右鍵)，然後選擇 Modify (修改)。
6. 針對 Value data (值資料)，請為 NVIDIA GRID 虛擬應用程式輸入 0，並選擇 OK (確定)。
7. 開啟右窗格中的內容選單 (按一下滑鼠右鍵)，然後選擇 New (新增)、DWORD。
8. 針對 Name (名稱)，輸入 IgnoreSP 然後輸入 Enter。
9. 開啟 IgnoreSP 上的內容選單 (按一下滑鼠右鍵)，然後選擇 Modify (修改)。
10. 針對 Value data (值資料)，輸入 1 並選擇 OK (確定)。
11. 關閉登錄編輯器。

在 Amazon EC2 執行個體最佳化 GPU 設定

您可以執行數種 GPU 設定最佳化，讓 NVIDIA GPU執行個體發揮最佳效能。透過其中一些執行個體類型，NVIDIA 驅動程式會使用 Autoboost 功能，而這會改變 GPU 時脈速度。您可以停用 Autoboost 並將 GPU 時脈速度設為頻率上限，即可持續發揮 GPU 執行個體的最大效能。

在 Linux 最佳化 GPU 設定

1. 設定 GPU 以保有持續性。此命令可能需要幾分鐘的執行時間。

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. 【僅限 G3 執行個體】 停用執行個體上所有 GPUs 的自動啟動功能。

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```

3. 將所有 GPU 時脈速度設為頻率上限。使用下列命令中指定的記憶體和圖形時脈速度。

   某些版本的 NVIDIA 驅動程式不支援設定應用程式時脈速度，會顯示錯誤 "Setting applications clocks is not supported for GPU..."，您可以忽略此錯誤。

   - G3 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
     ```

   - G4dn 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
     ```

   - G5 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 6250,1710
     ```

   - G6、G6f、Gr6 與 Gr6f 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 6251,2040
     ```

   - G6e 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 9001,2520
     ```

   - G7e 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 12481,2430
     ```

   - G7 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 12501,2415
     ```

   - P3 和 P3dn 執行個體：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 877,1530
     ```

   - P4d 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 1215,1410
```

- P4de 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 1593,1410
```

- P5 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 2619,1980
```

- P5e 與 P5en 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3201,1980
```

- P6-B200 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3996,1965
```

- P6-B300 執行個體：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3996,2032
```

在 Windows 最佳化 GPU 設定

1. 開啟 PowerShell 視窗，然後瀏覽至 NVIDIA 安裝資料夾。

```
PS C:\> cd "C:\Windows\System32\DriverStore\FileRepository\nvgridsw_aws.inf_*\"
```

2. 【僅限 G3 執行個體】 停用執行個體上所有 GPUs 的自動啟動功能。

```
PS C:\> .\nvidia-smi --auto-boost-default=0
```

3. 將所有 GPU 時脈速度設為頻率上限。使用下列命令中指定的記憶體和圖形時脈速度。

   某些版本的 NVIDIA 驅動程式不支援設定應用程式時脈速度，會顯示錯誤 "Setting applications clocks is not supported for GPU..."，您可以忽略此錯誤。

   - G3 執行個體：

```
PS C:\> .\nvidia-smi -ac "2505,1177"
```

- G4dn 執行個體：

```
PS C:\> .\nvidia-smi -ac "5001,1590"
```

- G5 執行個體：

```
PS C:\> .\nvidia-smi -ac "6250,1710"
```

- G6、G6f、Gr6 與 Gr6f 執行個體：

```
PS C:\> .\nvidia-smi -ac "6251,2040"
```

- G6e 執行個體：

```
PS C:\> .\nvidia-smi -ac "9001,2520"
```

- P3 和 P3dn 執行個體：

```
PS C:\> .\nvidia-smi -ac "877,1530"
```

在 G4ad Linux 執行個體設定雙 4K 顯示器

啟動 G4ad 執行個體後，您可以設定雙 4K 顯示器。

安裝 AMD 驅動程式並設定雙螢幕

1. 連線到 Linux 執行個體以獲取您希望用於雙 4K (2x4k) 的目標 GPU PCI 匯流排地址：

```
lspci -vv | grep -i amd
```

您將會得到類似下列的輸出：

```
00:1e.0 Display controller: Advanced Micro Devices, Inc. [*AMD*/ATI] Device 7362 (rev
 c3)
Subsystem: Advanced Micro Devices, Inc. [AMD/ATI] Device 0a34
```

2. 注意上述輸出中 PCI 匯流排地址為 00:1e.0。建立名為 /etc/modprobe.d/amdgpu.conf 的檔案並新增：

```
options amdgpu virtual_display=0000:00:1e.0,2
```

3. 若要在 Linux 安裝 AMD 驅動程式，請參閱 適用於 EC2 執行個體的 AMD 驅動程式。如果您已經安裝了 AMD GPU 驅動程式，則需要透過 DKMS 重建 amdgpu 核心模組。

4. 使用下面的 xorg.conf 檔案定義雙 (2x4k) 畫面拓撲，並將檔案儲存在 /etc/X11/xorg.conf：

```
~$ cat /etc/X11/xorg.conf
Section "ServerLayout"
    Identifier     "Layout0"
    Screen          0 "Screen0"
    Screen        1 "Screen1"
    InputDevice     "Keyboard0" "CoreKeyboard"
    InputDevice     "Mouse0" "CorePointer"
    Option          "Xinerama" "1"
EndSection
Section "Files"
    ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
    ModulePath "/opt/amdgpu/lib/xorg/modules"
    ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
    ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
    ModulePath "/usr/lib64/xorg/modules"
    ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Mouse0"
    Driver         "mouse"
    Option         "Protocol" "auto"
    Option         "Device" "/dev/psaux"
    Option         "Emulate3Buttons" "no"
    Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Keyboard0"
    Driver         "kbd"
EndSection

Section "Monitor"
```

```
    Identifier     "Virtual"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "Primary" "true"
EndSection

Section "Monitor"
    Identifier     "Virtual-1"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "RightOf" "Virtual"
EndSection

Section "Device"
    Identifier     "Device0"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Device"
    Identifier     "Device1"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Extensions"
    Option         "DPMS" "Disable"
EndSection

Section "Screen"
    Identifier     "Screen0"
    Device         "Device0"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection
```

```
Section "Screen"
    Identifier     "Screen1"
    Device         "Device1"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection
```

5. 請依照「設定互動式桌面」中的指示來設定 DCV。
6. DCV 設定完成後，請重新啟動。
7. 確認驅動程式可正常運作：

```
dmesg | grep amdgpu
```

回應應如下所示：

```
Initialized amdgpu
```

8. 您應該會在 DISPLAY=:0 xrandr -q 輸出中看到您已連接 2 個虛擬顯示器：

```
~$ DISPLAY=:0 xrandr -q
Screen 0: minimum 320 x 200, current 3840 x 1080, maximum 16384 x 16384
Virtual connected primary 1920x1080+0+0 (normal left inverted right x axis y axis)
 0mm x 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
```

```
   1280x800 59.96
   1280x720 59.97
   1152x768 59.95
   1024x768 60.00 59.95
   800x600  60.32 59.96 56.25
   848x480  60.00 59.94
   720x480  59.94
   640x480  59.94 59.94
Virtual-1 connected 1920x1080+1920+0 (normal left inverted right x axis y axis) 0mm x
   0mm
   4096x3112  60.00
   3656x2664  59.99
   4096x2160  60.00
   3840x2160  60.00
   1920x1200  59.95
   1920x1080  60.00
   1600x1200  59.95
   1680x1050  60.00
   1400x1050  60.00
   1280x1024  59.95
   1440x900 59.99
   1280x960 59.99
   1280x854 59.95
   1280x800 59.96
   1280x720 59.97
   1152x768 59.95
   1024x768 60.00 59.95
   800x600  60.32 59.96 56.25
   848x480  60.00 59.94
   720x480  59.94
  640x480  59.94 59.94
```

9. 當您連接到 DCV 時，將解析度變更為 2x4k，確認 DCV 已註冊雙顯示器支援。



設定適用於 Linux 的互動式桌面

在您確認 Linux 執行個體已安裝 AMD GPU 驅動程式且 amdgpu 正在使用中之後，您就可以安裝互動式桌面管理員。為了獲得最佳的相容性和效能，我們建議您使用 MATE 桌面環境。

先決條件

開啟文字編輯器，並將以下內容儲存為名為 xorg.conf 的檔案。你的執行個體需要這個檔案。

```
Section "ServerLayout"
Identifier     "Layout0"
Screen          0 "Screen0"
InputDevice     "Keyboard0" "CoreKeyboard"
InputDevice     "Mouse0" "CorePointer"
EndSection
Section "Files"
ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
ModulePath "/opt/amdgpu/lib/xorg/modules"
ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
ModulePath "/usr/lib64/xorg/modules"
ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Mouse0"
Driver         "mouse"
Option         "Protocol" "auto"
Option         "Device" "/dev/psaux"
Option         "Emulate3Buttons" "no"
Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Keyboard0"
Driver         "kbd"
EndSection
Section "Monitor"
Identifier     "Monitor0"
VendorName     "Unknown"
ModelName      "Unknown"
EndSection
Section "Device"
Identifier     "Device0"
Driver         "amdgpu"
VendorName     "AMD"
BoardName      "Radeon MxGPU V520"
BusID          "PCI:0:30:0"
EndSection
Section "Extensions"
Option         "DPMS" "Disable"
```

```
EndSection
Section "Screen"
Identifier     "Screen0"
Device         "Device0"
Monitor        "Monitor0"
DefaultDepth   24
Option         "AllowEmptyInitialConfiguration" "True"
SubSection "Display"
    Virtual    3840 2160
    Depth      32
EndSubSection
EndSection
```

若要在 Amazon Linux 2 上設定互動式桌面

1. 安裝 EPEL 儲存庫。

```
[ec2-user ~]$ sudo amazon-linux-extras install epel -y
```

2. 安裝 MATE 桌面。

```
[ec2-user ~]$ sudo amazon-linux-extras install mate-desktop1.x -y
[ec2-user ~]$ sudo yum groupinstall "MATE Desktop" -y
[ec2-user ~]$ sudo systemctl disable firewalld
```

3. 將 `xorg.conf` 檔案複製至 `/etc/X11/xorg.conf`。
4. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

5. (選用) 安裝 Amazon DCV 伺服器，使用 Amazon DCV 做為高效能顯示通訊協定，然後使用偏好的用戶端連線至 Amazon DCV 工作階段。

在 Ubuntu 上設定互動式桌面

1. 安裝 MATE 桌面。

```
$ sudo apt install xorg-dev ubuntu-mate-desktop -y
$ sudo apt purge ifupdown -y
```

2. 將 `xorg.conf` 檔案複製至 `/etc/X11/xorg.conf`。

3. 重新啟動執行個體。

```
$ sudo reboot
```

4. 為適當版本的 Ubuntu 安裝 AMF 編碼器。

```
$ sudo apt install ./amdgpu-pro-20.20-*/amf-amdgpu-pro_20.20-*_amd64.deb
```

5. (選用) 安裝 Amazon DCV 伺服器，使用 Amazon DCV 做為高效能顯示通訊協定，然後使用偏好的用戶端連線至 Amazon DCV 工作階段。
6. 安裝 DCV 之後給予 DCV 使用者視訊許可：

```
$ sudo usermod -aG video dcv
```

在 CentOS 上設定互動式桌面

1. 安裝 EPEL 儲存庫。

```
$ sudo yum update -y
$ sudo yum install epel-release -y
```

2. 安裝 MATE 桌面。

```
$ sudo yum groupinstall "MATE Desktop" -y
$ sudo systemctl disable firewalld
```

3. 將 `xorg.conf` 檔案複製至 `/etc/X11/xorg.conf`。
4. 重新啟動執行個體。

```
$ sudo reboot
```

5. (選用) 安裝 Amazon DCV 伺服器，使用 Amazon DCV 做為高效能顯示通訊協定，然後使用偏好的用戶端連線至 Amazon DCV 工作階段。

開始使用 GPU 加速執行個體

最新一代 GPU 加速執行個體類型，例如下列清單所示，可為深度學習及高效能運算 (HPC) 應用程式提供最高效能功能。選取執行個體類型連結，進一步了解其功能。

- P6 系列
- P6 系列
- P5 系列

如需加速執行個體類型之執行個體類型規格的完整清單，請參閱「Amazon EC2 執行個體類型」參考中的加速運算。

軟體組態

開始使用最新一代 GPU 加速執行個體類型的最簡單方法是從預先設定所有必要軟體的 AWS 深度學習 AMI 啟動執行個體。如需搭配 GPU 加速執行個體類型使用的最新 AWS 深度學習 AMIs 相關資訊，請參閱《 AWS 深度學習 AMIs 開發人員指南》中的 P6 支援的 DLAMI。

如果您需要建置自訂 AMI 以啟動託管深度學習或 HPC 應用程式的執行個體，建議您在基礎映像上方安裝下列最低軟體版本。

執行個體類型	NVIDIA 驅動程式	CUDA	NVIDIA GDRCopy	EFA 安裝程式	NCCL	EFA K8s [1]
G7	595	13.2	2.5.1	1.48.0	2.29.7-1	0.5.10
G7e	575	12.9	2.5	1.45.0	2.28.3	0.5.10
P5	530	12.1	2.3	1.24.1	2.18.3	0.4.4
P5.4xlarge	530	12.1	2.3	1.43.1 [2]	2.18.3	0.4.4
P5e	550	12.1	2.3	1.24.1	2.18.3	0.5.5
P5en	550	12.1	2.3	1.24.1	2.18.3	0.5.6
P6-B200	570	12.8	2.5	1.41.0	2.26.2-1	0.5.10
P6e-GB200	570	12.8	2.5	1.41.0	2.26.2-1	0.5.10
P6-B300	580	13.0	2.5	1.44.0	2.28.3	0.5.10

[1] EFA K8s 欄包含適用於 `aws-efa-k8s-device-plugin` 的最低建議版本。

[2] 當 GPU 至 GPU 通訊使用 Elastic Fabric Adapter (EFA) 與 NVIDIA Collective Communications Library (NCCL) 時，存在影響 P5.4xlarge 執行個體的相容性問題。如需緩解此問題，設定環境變數 FI_HMEM_DISABLE_P2P 為 1，並確保安裝版本 EFA 1.43.1 或更新版本。

> **Note**
>
> 若使用 EFA 安裝程式版本 1.41.0，aws-ofi-nccl plugin 會隨附該版本。若是較舊 EFA 安裝程式版本，則使用 aws-ofi-nccl plugin 版本 1.7.2-aws 或更新版本。

我們也建議您將執行個體設定為不使用深層的 C-states。如需詳細資訊，請參閱「Amazon Linux 2 使用者指南」中的限制深層的 C-state 達到高效能與低延遲。最新的 AWS 深度學習基礎 GPU AMIs 已預先設定為不使用更深層的 C 狀態。

如需網路和 Elastic Fabric Adapter (EFA) 組態，請參閱 使用多個網路卡最大化 Amazon EC2 執行個體的網路頻寬。

Amazon EC2 Mac 執行個體

EC2 Mac 執行個體適用於開發、建置、測試和簽署 Apple 平台的應用程式，這些平台如 iPhone、iPad、iPod、Mac、Vision Pro、Apple Watch、Apple TV 和 Safari。您可以使用 SSH 或 Apple 遠端桌面 (ARD) 連線至 Mac 執行個體。

> **Note**
>
> 以專用執行個體為計費單位。在該主機上執行的執行個體並無額外費用。

Amazon EC2 Mac 執行個體原生支援 macOS 作業系統。

- EC2 x86 Mac 執行個體 (mac1.metal) 基於 2018 Mac mini 硬體建置，並搭載 3.2 GHz Intel 第八代 (Coffee Lake) Core i7 處理器，6 個實體核心與 12 個邏輯核心，以及 32 GiB 記憶體。
- EC2 M1 Mac 執行個體 (mac2.metal) 基於 2020 Mac mini 硬體建置，並搭載 Apple 晶片 M1 處理器、8 個 CPU 核心、8 個 GPU 核心、16 GiB 記憶體和 16 核心 Apple Neural 引擎。
- EC2 M1 Ultra Mac 執行個體 (mac2-m1ultra.metal) 基於 2022 Mac Studio 建置，並搭載 2022 Apple 晶片 M1 Ultra 處理器、20 個 CPU 核心、64 個 GPU 核心、128 GiB 記憶體和 32 核心 Apple Neural 引擎。

- EC2 M2 Mac 執行個體 (mac2-m2.metal) 基於 2023 Mac mini 硬體建置，並搭載 Apple 晶片 M2 處理器、8 個 CPU 核心、10 個 GPU 核心、24 GiB 記憶體和 16 核心 Apple Neural 引擎。
- EC2 M2 Pro Mac 執行個體 (mac2-m2pro.metal) 基於 2023 Mac mini 硬體建置，並搭載 Apple 晶片 M2 Pro 處理器、12 個 CPU 核心、19 個 GPU 核心、32 GiB 記憶體和 16 核心 Apple Neural 引擎。
- EC2 M4 Mac 執行個體 (mac-m4.metal) 基於 2024 Mac mini 硬體建置，並搭載 Apple 晶片 M4 處理器、10 個 CPU 核心、10 個 GPU 核心、24 GiB 記憶體和 16 核心 Apple Neural 引擎。
- EC2 M4 Pro Mac Mac 執行個體 (mac-m4pro.metal) 基於 2024 Mac mini 硬體建置，並搭載 Apple 晶片 M4 Pro 處理器、14 個 CPU 核心、20 個 GPU 核心、48 GiB 記憶體和 16 核心 Apple Neural 引擎。

Amazon EC2 Mac 專用主機支援專用主機自動復原與重新啟動型主機維護。

目錄

- 考量事項
- 執行個體整備
- EC2 macOS AMI
- EC2 macOS Init
- 適用於 macOS 的 Amazon EC2 系統監控
- 相關資源
- 使用 AWS 管理主控台 或 啟動 Mac 執行個體 AWS CLI
- 使用 SSH 或 GUI 連線到您的 MAC 執行個體
- 更新 Amazon EC2 Mac 執行個體上的作業系統和軟體
- 增加 Mac 執行個體上 EBS 磁碟區的大小
- 停止或終止 Amazon EC2 Mac 執行個體
- 設定 Amazon EC2 Mac 執行個體的系統完整性保護
- 尋找支援 Amazon EC2 Mac 專用主機的 macOS 版本
- 訂閱 macOS AMI 通知
- 使用 AWS Systems Manager 參數存放區 API 擷取 macOS AMI IDs
- Amazon EC2 macOS AMI 的版本備註

考量事項

Mac 執行個體適用下列考量：

- Mac 執行個體只能做為專用執行個體上的裸機執行個體使用，最小配置期間為 24 小時，這段期間過後您才可以發佈專用執行個體。每個 專用執行個體 可以啟動一個 Mac 執行個體。您可以與 AWS 組織內 AWS 的帳戶或組織單位或整個 AWS 組織共用專用主機。
- Mac 執行個體可在不同的 中使用 AWS 區域。如需每個 中的 Mac 執行個體可用性清單 AWS 區域，請參閱依區域分類的 Amazon EC2 執行個體類型。
- Mac 執行個體只能做為 隨需執行個體 使用，不可做為 競價型執行個體 或 預留執行個體 使用。您可以購買 Savings Plan，節省 Mac 執行個體的費用。
- 不同 Mac 執行個體類型與特定 macOS Amazon Machine Image (AMI) 的相容性有所不同。如需詳細資訊，請參閱Amazon EC2 macOS AMI 的版本備註。
- 支援 EBS 熱插拔。
- AWS 不會管理或支援 Apple 硬體上的內部 SSD。強烈建議您改用 Amazon EBS 磁碟區。EBS 磁碟區在 Mac 執行個體上提供與任何其他 EC2 執行個體相同的彈性、可用性和耐用性優勢。
- 建議您將具有 10,000 IOPS 和 400 MiB/s 輸送量的 Amazon EBS 磁碟區搭配 Mac 執行個體使用，以達到最佳效能。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 磁碟區類型。
- Mac 執行個體支援 Amazon EC2 Auto Scaling。
- 在 x86 Mac 執行個體上，自動軟體更新會停用。我們建議您在將執行個體投入實際執行環境前套用更新，並在執行個體上測試更新。如需詳細資訊，請參閱更新 Amazon EC2 Mac 執行個體上的作業系統和軟體。
- 當您停止或終止 Mac 執行個體時，系統會在 專用主機 上執行清理工作流程。如需詳細資訊，請參閱停止或終止 Amazon EC2 Mac 執行個體。
- > **Important**
  >
  > 從外部磁碟區啟動 Mac 硬體時，無法使用 Apple Intelligence 功能。EC2 Mac 執行個體預設從外部 EBS 磁碟區啟動，因此不支援 Apple Intelligence 功能。
- > **Warning**
  >
  > 請勿使用 FileVault。若您啟用 FileVault，主機無法開機，因為分區已鎖定。如果需要資料加密，請使用 Amazon EBS 加密，以避免開機問題和效能影響。使用 Amazon EBS 加密，加密操作會在託管執行個體的伺服器上進行，確保執行個體和與其連接之 EBS 儲存體間

靜態資料和傳輸中資料的安全。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 加密。

執行個體整備

啟動 Mac 執行個體之後，您將需要等待執行個體準備就緒後，才能夠與其連接。對於具有 x86 Mac 執行個體或 Apple 晶片 Mac 執行個體的 AWS 付費 AMI，啟動時間約為 6 分鐘至 20 分鐘。根據選擇的 Amazon EBS 磁碟區大小、使用者資料中包含的額外指令、或自訂 macOS AMI 上的其他載入軟體，啟動時間可能會增加。

您可以使用如下所示的小型 shell 指令碼來輪詢 describe-instance-status API，以瞭解執行個體何時可以連接。將以下命令中的範例執行個體 ID 取代為您自己的 ID：

```
for i in $(seq 1 200); do aws ec2 describe-instance-status --instance-
ids=i-1234567890abcdef0 \
    --query='InstanceStatuses[0].InstanceStatus.Status'; sleep 5; done;
```

EC2 macOS AMI

Amazon EC2 macOS 設計能為在 Amazon EC2 Mac 執行個體上執行的開發人員工作負載提供穩定、安全且高效能的環境。EC2 macOS AMIs 包含可讓 輕鬆整合的套件 AWS，例如啟動組態工具和熱門 AWS 程式庫和工具。

如需有關 EC2 macOS AMI 的詳細資訊，請參閱 Amazon EC2 macOS AMI 的版本備註。

AWS 定期提供更新的 EC2 macOS AMIs，其中包含 擁有的套件更新， AWS 以及最新的完整測試 macOS 版本。此外， AWS 也會在更新的 AMI 經過完整測試和驗證後，提供具有最新次要版本更新或主要版本更新的 AMI。如果您無須保留 Mac 執行個體的資料或自訂，您可以使用目前的 AMI 啟動新的執行個體，然後終止上一個執行個體，以取得最新的更新。否則您可以選擇哪些更新要套用到您的 Mac 執行個體。

如需如何訂閱 macOS AMI 通知的詳細資訊，請參閱 訂閱 macOS AMI 通知。

EC2 macOS Init

EC2 Init 用於在啟動時初始化 EC2 Mac 執行個體。它會使用優先順序群組，來同時執行任務的邏輯群組。

launchd plist 檔案是 `/Library/LaunchDaemons/com.amazon.ec2.macos-init.plist`。EC2 macOS Init 的檔案位於 `/usr/local/aws/ec2-macos-init`。

如需詳細資訊，請參閱 https://github.com/aws/ec2-macos-init。

適用於 macOS 的 Amazon EC2 系統監控

適用於 macOS 的 Amazon EC2 系統監控會提供 Amazon CloudWatch 的 CPU 使用率指標。它會在 1 分鐘的期間內透過自訂序列裝置傳送這些指標至 CloudWatch。您可以啟用或停用此代理程式，如下所示。依預設會啟用此功能。

```
sudo setup-ec2monitoring [enable | disable]
```

> **Note**
>
> Apple 晶片 Mac 執行個體上目前不支援適用於 macOS 的 Amazon EC2 系統監控。

相關資源

如需關於定價的資訊，請參閱定價。

如需 Mac 執行個體的詳細資訊，請參閱 Amazon EC2 Mac 執行個體。

如需 Mac 執行個體的硬體規格和網路效能的詳細資訊，請參閱一般用途執行個體。

使用 AWS 管理主控台 或 啟動 Mac 執行個體 AWS CLI

EC2 Mac 執行個體需要專用執行個體。您首先需要將主機配置到您的帳戶，然後在主機上啟動執行個體。

您可以使用 AWS 管理主控台 或 啟動 Mac 執行個體 AWS CLI。

使用主控台啟動 Mac 執行個體

在 專用主機 上啟動 Mac 實體啟動

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 配置專用執行個體，如下所示：
    a. 在導覽窗格中，選擇 專用主機。
    b. 選擇 Allocate 專用執行個體 (配置 專用執行個體)，然後執行下列動作：
        i. 對於執行個體系列，選擇 Mac 執行個體系列。如果 Instance family (執行個體系列) 未在清單中出現，則表示目前所選的區域不支援。

ii. 針對執行個體類型，根據所選的執行個體系列，選擇執行個體類型。

iii. 在 Availability Zone (可用區域) 中，選擇 專用執行個體 的可用區域。

iv. 在 Quantity (數量) 中，保留 1。

v. 選擇 Allocate (配置)。

3. 在主機上啟動執行個體，如下所示：

a. 選取您建立的 專用執行個體，然後執行下列動作：

i. 選擇 Actions (動作)、Launch instance(s) onto host (在主機上啟動執行個體)。

ii. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，選取 macOS AMI。

iii. 在執行個體類型中，選取 Mac 執行個體類型。

iv. 在進階詳細資訊下，確認已根據您建立的專用主機預先設定租用、租用執行個體者執行個體者) 和 Tenancy host ID (租用執行個體 ID)。視需要更新 Tenancy affinity (租用親和性)。

v. 視需要完成精靈、指定 EBS 磁碟區、安全群組和金鑰對。

vi. 在 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

b. 會有確認頁面讓您知道您的執行個體正在啟動。選擇 View all instances (檢視所有執行個體)，以關閉確認頁面並返回主控台。執行個體的初始狀態為 `pending`。當其狀態變更為 `running` 並通過狀態檢查時，則表示該執行狀態就緒。

使用 啟動 Mac 執行個體 AWS CLI

配置專用執行個體

使用下列 allocate-hosts 指令為您的 Mac 執行個體配置專用主機，將 `instance-type` 取代為有效 mac 執行個體類型，並將 `region` 和 `availability-zone` 取代為適合您環境的項目。

```
aws ec2 allocate-hosts --region us-east-1 --instance-type mac1.metal --availability-
zone us-east-1b --auto-placement "on" --quantity 1
```

在主機上啟動執行個體

使用下列 run-instances 指令來啟動 Mac 執行個體，再次將其 `instance-type` 取代為 有效 mac 執行個體，並將 `region` 和 `availability-zone` 取代為之前使用的執行個體。

```
aws ec2 run-instances --region us-east-1 --instance-type mac1.metal --placement
 Tenancy=host --image-id ami_id --key-name my-key-pair
```

執行個體的初始狀態為 pending。當其狀態變更為 running 並通過狀態檢查時，則表示該執行狀態就緒。使用以下 describe-instance-status 命令顯示您執行個體的狀態資訊。

```
aws ec2 describe-instance-status --instance-ids i-017f8354e2dc69c4f
```

以下是執行中並通過狀態檢查之執行個體的範例輸出。

```
{
    "InstanceStatuses": [
        {
            "AvailabilityZone": "us-east-1b",
            "InstanceId": "i-017f8354e2dc69c4f",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "InstanceStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            },
            "SystemStatus": {
                "Details": [
                    {
                        "Name": "reachability",
                        "Status": "passed"
                    }
                ],
                "Status": "ok"
            }
        }
    ]
}
```

使用 SSH 或 GUI 連線到您的 MAC 執行個體

您可以使用 SSH 或圖形化使用者介面 (GUI) 連線到 Mac 執行個體。

多個使用者可以同時存取作業系統。由於連接埠 5900 上提供內建螢幕共享服務，所以通常有一個 1:1 user:GUI 工作階段。在 macOS 中使用 SSH 支援多個工作階段，最多可達 `sshd_config` 檔案中的「工作階段上限」限制。

使用 SSH 連線到您的執行個體

Amazon EC2 Mac 執行個體預設不允許遠端根 SSH。ec2-user 帳戶設為使用 SSH 進行遠端登入。ec2-user 帳戶也擁有 sudo 權限。連線到您的執行個體後，您可以新增其他使用者。

若要支援使用 SSH 連線到您的執行個體，請使用允許 SSH 存取的金鑰對和安全群組啟動執行個體，並確保執行該個體具有網際網路連線能力。連線到執行個體時，您可以提供金鑰對的 `.pem` 檔案。

請使用下列程序使用 SSH 用戶端連線至您的 Mac 執行個體。如果您在嘗試連線至執行個體時遇到錯誤，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

使用 SSH 連線到您的執行個體

1. 在命令列輸入 ssh，以確認您的本機電腦已安裝 SSH 用戶端。如果您的電腦無法辨識命令，請搜尋適用於作業系統的 SSH 用戶端並安裝該用戶端。
2. 取得執行個體的公有 DNS 名稱。您可以使用 Amazon EC2 主控台尋找 Details (詳細資訊) 和 Networking (網路) 標籤上的公有 DNS 名稱。您可以使用 describe-instances 命令 AWS CLI來尋找公有 DNS 名稱。
3. 找到您在啟動執行個體時所指定金鑰對的 `.pem` 檔案。
4. 使用以下 ssh 命令、指定執行個體的公有 DNS 名稱和 `.pem` 檔案，以連線到您的執行個體。

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

密碼驗證也將停用，以避免密碼遭到暴力破解攻擊。變更 SSH 組態之前，請開啟 `/usr/local/aws/ec2-macos-init/init.toml` 並將 `secureSSHDConfig` 設為 `false`。

連線至您的執行個體的圖形使用者介面 (GUI)

請使用下列程序，藉助 VNC、Apple 遠端桌面 (ARD) 或 Apple 螢幕共享應用程式 (隨附於 macOS) 連線至您的執行個體 GUI。

> **Note**
>
> macOS 10.14 及以上版本只允許控制是否透過 System Preferences (系統偏好設定) 啟用 Screen Sharing (螢幕共用)。

使用 ARD 用戶端或 VNC 用戶端連線到您的執行個體

1. 確認您的本機電腦已安裝 ARD 用戶端，或支援已安裝 ARD 的 VNC 用戶端。在 macOS 上，您可以利用內建的「螢幕共用」應用程式。否則請搜尋適用於您作業系統的 ARD，然後安裝該用戶端。
2. 從您的本機電腦，使用 SSH 連線到您的執行個體。
3. 如下所示使用 passwd 命令設定 ec2-user 帳戶的密碼。

```
[ec2-user ~]$ sudo passwd ec2-user
```

4. 使用下列命令，安裝並啟動 macOS Screen Sharing。

```
[ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/com.apple.screensharing.plist
```

5. 輸入 exit 並按 Enter，斷開與執行個體的連線。
6. 從您的電腦，使用以下 ssh 命令連線到執行個體。除了上一節所述的選項之外，使用 -L 選項也可啟用連接埠轉送，並將本機連接埠 5900 的所有流量轉送至執行個體上的 ARD 伺服器。

```
ssh -L 5900:localhost:5900 -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

7. 從您的本機電腦，使用支援 ARD 的 ARD 用戶端或 VNC 用戶端連線至 `localhost:5900`。例如，使用 macOS 上的「螢幕共用」應用程式，如下所示：
   a. 開啟 Finder，並選取開始。
   b. 選取連線至伺服器。
   c. 在伺服器地址欄位中，輸入 `vnc://localhost:5900`。
   d. 將 **`ec2-user`** 當作您為 ec2-user 帳戶建立的使用者名稱和密碼，依提示登入。

修改 Mac 執行個體上的 macOS 螢幕解析度

使用 ARD 或支援 ARD 的 VNC 用戶端連線至 EC2 Mac 執行個體後，您就可以使用任何公開可用的 macOS 工具或公用程式，例如 displayplacer 來修改 macOS 環境的螢幕解析度。

若要使用 displayplacer 修改螢幕解析度

1. 安裝 displayplacer。

```
[ec2-user ~]$ brew tap jakehilborn/jakehilborn && brew install displayplacer
```

2. 顯示目前的螢幕資訊和可能的螢幕解析度。

```
[ec2-user ~]$ displayplacer list
```

3. 套用所需的螢幕解析度。

```
[ec2-user ~]$ displayplacer "id:<screenID> res:<width>x<height> origin:(0,0)
 degree:0"
```

例如：

```
RES="2560x1600"
displayplacer "id:69784AF1-CD7D-B79B-E5D4-60D937407F68 res:${RES} scaling:off
 origin:(0,0) degree:0"
```

更新 Amazon EC2 Mac 執行個體上的作業系統和軟體

下列主題說明如何更新 Apple 晶片 Mac 執行個體 (Mac2、Mac2-m1ultra, Mac2-m2, Mac2-m2pro, Mac-m4 和 Mac-m4pro) 和 x86 Mac 執行個體 (Mac1) 上的作業系統與軟體。

> **Warning**
>
> 只有 Apple 晶片 Mac 執行個體才能安裝 Beta 版或預覽 macOS 版本。Amazon EC2 不符合 Beta 版或預覽版 macOS 版本的資格，且無法確保執行個體在更新至生產前 macOS 版本後仍能正常運作。

嘗試在 Amazon EC2 x86 Mac 執行個體上安裝 Beta 版或預覽版 macOS 版本，會導致停止或終止執行個體時，EC2 Mac 專用執行個體降級，並且會阻止您在該專用執行個體上開始或啟動新的執行個體。

Note

若您在 AWS 發行官方 AMI 之前執行就地 macOS 更新，則更新僅適用於選取的主機。若您有其他主機，或啟動新的主機，也必須在這些主機上執行相同的更新程序。每個 macOS 版本在基礎 Apple Mac 硬體上都需要最低韌體版本。就地更新僅會更新所選主機上的韌體，而不會傳輸至其他現有或新的主機。要檢查哪些 macOS 版本與您的 Amazon EC2 Mac 專用主機相容，請參閱為您的 Amazon EC2 Mac 專用主機尋找支援的 macOS 版本。

在 Apple 晶片 Mac 執行個體上更新軟體

先決條件

由於網路驅動程式組態中的更新，ENA 驅動程式 1.0.2 版與 macOS 13.3 或更新版本不相容。如果您要安裝 Beta 版、預覽版或生產 macOS 13.3 版或更新版本，且尚未安裝最新的 ENA 驅動程式，請使用下列程序安裝新的驅動程式版本。

安裝 ENA 驅動程式的新版本

1. 在「終端機」視窗中，使用 SSH 連線至 Apple 晶片 Mac 執行個體。
2. 使用下列命令更新 Homebrew 並將 ENA 應用程式下載到 `Applications` 檔案中。

```
[ec2-user ~]$ brew update
```

```
[ec2-user ~]$ brew install amazon-ena-ethernet-dext
```

3. 輸入 exit 並按 return，斷開與執行個體的連線。
4. 使用 VNC 用戶端啟動 ENA 應用程式。
   a. 使用 連線至您的執行個體的圖形使用者介面 (GUI) 設定 VNC 用戶端。
   b. 使用螢幕共用應用程式連線至執行個體後，請前往應用程式資料夾並開啟 ENA 應用程式。
   c. 選擇啟用

d. 若要確認驅動程式已正確啟動，請在終端視窗中執行下列命令。命令輸出顯示舊驅動程式處於終止狀態，並且新驅動程式處於啟動狀態。

```
systemextensionsctl list;
```

e. 重新啟動執行個體後，只會出現新的驅動程式。

執行軟體更新

在 Apple 晶片 Mac 執行個體上，您必須完成數個步驟才能執行就地作業系統更新。其中包括將 Amazon EBS 根磁碟區的所有權委派給 EBS 根磁碟區管理使用者。您可選擇使用 Amazon EC2 API 自動執行此操作，也可以在執行個體上執行命令來手動執行此操作。

Automated volume ownership delegation (Recommended)

考量事項

- 完成磁碟區的所有權委派任務可能需要 30 到 90 分鐘。在此期間，該執行個體將處於無法存取的狀態。
- 以下為可支援的 macOS 版本：
  - Mac2 | Mac2-m1ultra – macOS Ventura (13.0 版或更新版本)
  - Mac2-m2 | Mac2-m2pro – macOS Ventura (13.2 版或更新版本)
  - Mac-m4 | Mac-m4pro – macOS Sequoia (15.6 版或更新版本)
- 執行個體只能有一個可啟動磁碟區，而且每個連接的磁碟區只能有一個額外的管理員使用者。

步驟 1：設定密碼，並啟用 EBS 根磁碟區管理使用者的安全字符

您必須設定密碼，並為 Amazon EBS 根磁碟區管理使用者 (`ec2-user`) 啟用安全權杖。

> **Note**
>
> 密碼和安全字符在您第一次使用 GUI 連接到 Apple 晶片 Mac 執行個體時設定。若您之前使用 GUI 連接到執行個體，則不需要執行這些步驟。

1. 使用 SSH 連線至執行個體。
2. 設定 `ec2-user` 使用者的密碼。

```
$ sudo /usr/bin/dscl . -passwd /Users/ec2-user
```

3. 為 ec2-user 使用者啟用安全字符。對於 -oldPassword，指定與上一個步驟相同的密碼。對於 -newPassword，請指定不同的密碼。下列命令假設您已將舊密碼與新密碼儲存在 .txt 檔案中。

```
$ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
 new_password.txt`
```

4. 確認安全字符已啟用。

```
$ sysadminctl -secureTokenStatus ec2-user
```

步驟 2：將 Amazon EBS 根磁碟區的所有權委派給 EBS 根磁碟區的管理使用者

若要委派所有權，您必須建立磁碟區所有權委派任務。

1. 使用 create-delegate-mac-volume-ownership-task 命令建立任務。對於 --instance-id，指定執行個體 ID。對於 --mac-credentials，請指定下列憑證：

   - 內部磁碟的管理使用者
     - 使用者名稱 – 僅支援預設的管理使用者 (aws-managed-user)，並且預設會使用。您無法指定其他管理員使用者。
     - 密碼 – 若您未變更 aws-managed-user 的預設密碼，請指定預設密碼，即空白。否則，請指定您自訂的密碼。
   - Amazon EBS 根磁碟區管理使用者
     - 使用者名稱 – 若您未變更預設管理使用者，請指定 ec2-user。否則，為您的管理使用者指定使用者名稱。
     - 密碼 – 指定您在上述步驟 1 中為根磁碟區管理員使用者設定的密碼。

```
aws ec2 create-delegate-mac-volume-ownership-task \
--instance-id i-1234567890abcdef0 \
--mac-credentials file://mac-credentials.json
```

之前範例中所參考 mac-credentials.json 檔案的內容如以下所示。

```
{
  "internalDiskPassword":"internal-disk-admin_password",
  "rootVolumeUsername":"root-volume-admin_username",
  "rootVolumePassword":"root-volume-admin_password"
}
```

2. 等待磁碟區所有權委派任務完成，並等候執行個體恢復至良好運作狀態。使用 describe-mac-modification-tasks 命令。針對 `--mac-modification-task-id`，指定上一個步驟的磁碟區所有權委派任務 ID。

```
aws ec2 describe-mac-modification-tasks \
--mac-modification-task-id task-id
```

3. 磁碟區所有權委派任務完成後，繼續執行步驟 3。

步驟 3：更新軟體

在您委派 Amazon EBS 根磁碟區的所有權之後，請依照 在 x86 Mac 執行個體上更新軟體 (下方) 中所述的步驟更新軟體。

Manual volume ownership delegation

當您執行此程序時，會建立兩個密碼。一個密碼用於 Amazon EBS 根磁碟區管理使用者 (`ec2-user`)，另一個密碼用於內部磁碟管理使用者 (`aws-managed-user`)。請記住這些密碼，因為將在執行該程序時使用這些密碼。

> **Note**
>
> 在 macOS Big Sur 上使用此程序，只能執行次要更新，如從 macOS Big Sur 11.7.3 更新為 macOS Big Sur 11.7.4。對於 macOS Monterey 或更高版本，您可以執行主要軟體更新。

存取內部磁碟

1. 從本機電腦的「終端機」中，透過下列命令使用 SSH 連線至 Apple 晶片 Mac 執行個體。如需詳細資訊，請參閱使用 SSH 連線到您的執行個體。

```
ssh -i /path/key-pair-name.pem ec2-user@instance-public-dns-name
```

2. 使用下列命令，安裝並啟動 macOS Screen Sharing。

```
[ec2-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

3. 使用以下命令設定 ec2-user 的密碼。記住密碼，供以後使用。

```
[ec2-user ~]$ sudo /usr/bin/dscl . -passwd /Users/ec2-user
```

4. 輸入 exit 並按回車鍵，斷開與執行個體的連線。
5. 從本機電腦的「終端」中，使用下列命令透過 VNC 連接埠的 SSH 通道重新連線至執行個體。

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 ec2-user@instance-public-
dns-name
```

> **Note**
>
> 在完成下列 VNC 連線和 GUI 步驟之前，請勿結束此 SSH 工作階段。重新啟動執行個體時，連線會自動關閉。

6. 從您的本機電腦，使用下列步驟連線到 localhost:5900：
   a. 開啟 Finder，並選取開始。
   b. 選取連線至伺服器。
   c. 在伺服器地址欄位中，輸入 vnc://localhost:5900。
7. 在 macOS 視窗中，使用您在步驟 3 中建立的密碼以 ec2-user 身分連線至 Apple 晶片 Mac 執行個體的遠端工作階段。
8. 使用下列其中一個選項存取名為 InternalDisk 的內部磁碟。
   a. 對於 macOS Ventura 或更高版本：開啟系統設定，在左窗格中選取一般，然後選取窗格右下角的啟動磁碟。
   b. 對於 macOS Monterey 或以下版本：開啟系統偏好設定，選取啟動磁碟，然後選擇視窗左下角的鎖定圖像來解鎖窗格。

> **疑難排解秘訣**
>
> 如果您需要掛載內部磁碟，則請在終端中執行下列命令。

```
APFSVolumeName="InternalDisk" ; SSDContainer=$(diskutil list | grep
 "Physical Store disk0" -B 3 | grep "/dev/disk" | awk {'print $1'} ) ;
 diskutil apfs addVolume $SSDContainer APFS $APFSVolumeName
```

9. 選擇名為 InternalDisk 的內部磁碟，然後選取重新啟動。出現提示時再次選取重新啟動。

> **Important**
>
> 如果內部磁碟名為 Macintosh HD 而非 InternalDisk，則您的執行個體必須停止並重新啟動，才能更新專用執行個體。如需詳細資訊，請參閱停止或終止 Amazon EC2 Mac 執行個體。

使用下列程序將擁有權委派給管理使用者。當透過 SSH 重新連線至執行個體時，可以使用特殊管理使用者 (aws-managed-user) 從內部磁碟開機。aws-managed-user 的初始密碼為空白，因此您需要在第一次連線時覆寫它。然後，由於開機磁碟區已變更，因此您需要重複以下步驟來安裝和啟動 macOS Screen Sharing。

將擁有權委派給 Amazon EBS 磁碟區上的管理員

1. 從本機電腦的「終端機」中，使用下列命令連線至 Apple 晶片 Mac 執行個體。

```
ssh -i /path/key-pair-name.pem aws-managed-user@instance-public-dns-name
```

2. 當您收到警告 WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! 時，請使用下列其中一個命令來解決此問題。

   a. 使用以下命令清除已知主機。然後，重複上一個步驟。

   ```
   rm ~/.ssh/known_hosts
   ```

   b. 將下列命令新增到上一個步驟的 SSH 命令。

   ```
   -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no
   ```

3. 使用下列命令設定 aws-managed-user 的密碼。aws-managed-user 初始密碼為空白，因此您需要在第一次連線時覆寫它。

a.
```
[aws-managed-user ~]$ sudo /usr/bin/dscl . -passwd /Users/aws-managed-user password
```

b. 當您收到 `Permission denied. Please enter user's old password:` 提示時，請按 Enter 鍵。

> **疑難排解秘訣**
>
> 如果發生 `passwd: DS error: eDSAuthFailed` 錯誤，請使用下列命令。
>
> ```
> [aws-managed-user ~]$ sudo passwd aws-managed-user
> ```

4. 使用下列命令，安裝並啟動 macOS Screen Sharing。

```
[aws-managed-user ~]$ sudo launchctl enable system/com.apple.screensharing
sudo launchctl load -w /System/Library/LaunchDaemons/
com.apple.screensharing.plist
```

5. 輸入 exit 並按回車鍵，斷開與執行個體的連線。
6. 從本機電腦的「終端」中，使用下列命令透過 VNC 連接埠的 SSH 通道重新連線至執行個體。

```
ssh -i /path/key-pair-name.pem -L 5900:localhost:5900 aws-managed-user@instance-public-dns-name
```

7. 從您的本機電腦，使用下列步驟連線到 `localhost:5900`：

   a. 開啟 Finder，並選取開始。

   b. 選取連線至伺服器。

   c. 在伺服器地址欄位中，輸入 `vnc://localhost:5900`。

8. 在 macOS 視窗中，使用您在步驟 3 中建立的密碼以 `aws-managed-user` 身分連線至 Apple 晶片 Mac 執行個體的遠端工作階段。

> **Note**
>
> 當系統提示您使用 Apple ID 登入時，請選取稍後設定。

9. 使用下列其中一個選項，存取 Amazon EBS 磁碟區。

a. 對於 macOS Ventura 或更新版本：開啟系統設定，在左窗格中選取一般，然後選取窗格右下角的啟動磁碟。

b. 對於 macOS Monterey 或較早的版本：開啟系統偏好設定，選取啟動磁碟，然後使用視窗左下角的鎖定圖像來解鎖窗格。

> **Note**
>
> 在重新開機之前，當系統提示輸入管理員密碼時，請使用您上面設定的 aws-managed-user 密碼。此密碼可能與您為 ec2-user 或執行個體上的預設管理員帳戶設定的密碼不同。下列說明指定何時使用執行個體的管理員密碼。

10. 選取 Amazon EBS 磁碟區 (啟動磁碟視窗中未命名為 InternalDisk 的磁碟區)，然後選擇重新啟動。

> **Note**
>
> 如果您將多個可開機的 Amazon EBS 磁碟區連接至 Apple 晶片 Mac 執行個體，則請務必為每個磁碟區使用唯一的名稱。

11. 確認重新啟動，然後在出現提示時選擇授權使用者。
12. 於在此磁碟區授權使用者窗格中，確認已選取管理使用者 (預設為 ec2-user)，然後選取授權。
13. 輸入您在上一個程序的步驟 3 中建立的 ec2-user 密碼，然後選取繼續。
14. 出現提示時，請輸入特殊管理使用者的密碼 (aws-managed-user)。
15. 從本機電腦的「終端」中，使用 SSH 與使用者名稱 ec2-user 重新連線到執行個體。

> **疑難排解秘訣**
>
> 如果收到警告 WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!，則請執行下列命令，然後使用 SSH 重新連線至執行個體。
>
> ```
> rm ~/.ssh/known_hosts
> ```

16. 若要執行軟體更新，請使用 在 x86 Mac 執行個體上更新軟體 下的命令。

在 x86 Mac 執行個體上更新軟體

在 x86 Mac 執行個體上，您可以使用 `softwareupdate` 指令從 Apple 安裝作業系統更新。

若要在 x86 Mac 執行個體上從 Apple 安裝作業系統更新

1. 使用以下命令列出具有可用更新的套件。

   ```
   [ec2-user ~]$ softwareupdate --list
   ```

2. 安裝所有更新或僅特定更新。若要安裝特定更新，請使用下列命令。

   ```
   [ec2-user ~]$ sudo softwareupdate --install label
   ```

   若要安裝所有更新，請使用下列命令。

   ```
   [ec2-user ~]$ sudo softwareupdate --install --all --restart
   ```

系統管理員可以使用 AWS Systems Manager 在 x86 Mac 執行個體上推出預先核准的作業系統更新。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」。

您可以使用 Homebrew 將套件更新安裝於 EC2 macOS AMI，因此您在執行個體上可以擁有這些套件的最新版本。您也可以使用 Homebrew 在 Amazon EC2 macOS 上安裝和執行常用的 macOS 應用程式。如需詳細資訊，請參閱 Homebrew 文件。

使用 Homebrew 安裝更新

1. 使用以下命令更新 Homebrew。

   ```
   [ec2-user ~]$ brew update
   ```

2. 使用以下命令列出具有可用更新的套件。

   ```
   [ec2-user ~]$ brew outdated
   ```

3. 安裝所有更新或僅特定更新。若要安裝特定更新，請使用下列命令。

   ```
   [ec2-user ~]$ brew upgrade package name
   ```

   若要安裝所有更新，請使用下列命令。

```
[ec2-user ~]$ brew upgrade
```

增加 Mac 執行個體上 EBS 磁碟區的大小

您可以增加 Mac 執行個體上 Amazon EBS 磁碟區的大小。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 彈性磁碟區。

增加磁碟區的大小後，您必須依照下列方式增加 APFS 容器的大小。

使增加的磁碟空間可供使用

1. 判斷是否需要重新啟動。如果您已調整正在執行的 Mac 執行個體上的現有 EBS 磁碟區的大小，則須重新啟動執行個體，以使得新的大小可用。如果在啟動期間修改磁碟空間，則不需要重新開機。

   檢視磁碟大小的目前狀態：

   ```
   [ec2-user ~]$  diskutil list external physical
   /dev/disk0 (external, physical):
      #:                       TYPE NAME                    SIZE       IDENTIFIER
      0:                 GUID_partition_scheme            *322.1 GB     disk0
      1:                 EFI EFI                           209.7 MB     disk0s1
      2:                 Apple_APFS Container disk2        321.9 GB     disk0s2
   ```

2. 複製並貼上下列命令。

   ```
   [ec2-user ~]$ PDISK=$(diskutil list physical external | head -n1 | cut -d" " -f1)
   APFSCONT=$(diskutil list physical external | grep "Apple_APFS" | tr -s " " | cut -
   d" " -f8)
   yes | sudo diskutil repairDisk $PDISK
   ```

3. 複製並貼上下列命令。

   ```
   [ec2-user ~]$ sudo diskutil apfs resizeContainer $APFSCONT 0
   ```

停止或終止 Amazon EC2 Mac 執行個體

當您停止 Mac 執行個體時，該執行個體會在進入 `stopped` 狀態之前維持約 15 分鐘的 `stopping` 狀態。

當您停止或終止 Mac 執行個體時，Amazon EC2 會在底層的專用執行個體上執行清理工作流程以清除內部 SSD、清除持續性 NVRAM 變數，並更新至最新的裝置韌體。這可確保 Mac 執行個體提供與其他 EC2 Nitro 執行個體相同的安全性和資料隱私權。這也可讓您執行最新的 macOS AMI。在清理工作流程期間，專用執行個體會暫時進入 pending (待定) 狀態。在 x86 Mac 執行個體上，清理工作流程最多可能需要 50 分鐘才能完成。如果 Amazon EC2 需要更新裝置韌體，工作流程最多可能需要 3 小時才能完成。在 Apple Silicon Mac 執行個體上，清理工作流程最多可能需要 4.5 小時才能完成。

直到清理工作流程完成，也即 專用執行個體 進入 `available` 狀態，您才能啟動已停止的 Mac 執行個體或啟動新的 Mac 執行個體。

專用執行個體進入 `pending` 狀態時，計量和計費會暫停。清理工作流程期間，您不需支付費用。

為您的 Mac 執行個體釋出 專用主機

當您完成 Mac 執行個體時，您可以釋出 專用執行個體。您必須先停止或終止 Mac 執行個體，才能發佈專用執行個體。至少在配置期間經過 24 小時後才能發佈主機。

發佈專用執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體並選取 Instance state (執行個體狀態)，然後選取 Stop instance (停止執行個體) 或 Terminate instance (終止執行個體)。
4. 在導覽窗格中，選擇 專用執行個體。
5. 選取 專用執行個體 並選取 Actions (動作)，Release host (釋出主機)。
6. 出現確認提示時，請選擇 Release (釋出)。

設定 Amazon EC2 Mac 執行個體的系統完整性保護

您可為 x86 Mac 執行個體和 Apple 晶片 Mac 執行個體設定系統完整性保護 (SIP) 設定。SIP 是 macOS 重要的安全功能，可協助防止未經授權的程式碼執行與系統層級修改。如需詳細資訊，請參閱關於系統完整性保護。

您可以完全啟用或停用 SIP，也可選擇性地啟用或停用特定 SIP 設定。建議您僅暫時停用 SIP 來執行必要的任務，接著盡快重新啟用。保持 SIP 停用狀態可能會讓您的執行個體易受惡意程式碼攻擊。

支援 Amazon EC2 Mac 執行個體的所有 AWS 區域都支援 SIP 組態。

主題

- 考量事項
- 預設 SIP 組態
- 檢查您的 SIP 組態
- Apple 晶片 Mac 執行個體的先決條件
- 設定 SIP 設定
- 檢查 SIP 組態任務狀態

考量事項

- 支援下列 Amazon EC2 Mac 執行個體類型與 macOS 版本：
  - Mac1 | Mac2 | Mac2-m1ultra – macOS Ventura (13.0 版或更新版本)
  - Mac2-m2 | Mac2-m2pro – macOS Ventura (13.2 版或更新版本)
  - Mac-m4 | Mac-m4pro – macOS Sequoia (15.6 版或更新版本)

  > **Note**
  >
  > 不支援 macOS 的 Beta 版與預覽版。

- 您可指定自訂 SIP 組態，以選擇性地啟用或停用個別 SIP 設定。若您實作自訂組態，請連線至執行個體並驗證設定，確保您的需求已正確實施且運作符合預期。

  SIP 組態可能會隨 macOS 更新而變更。建議您在每次升級 macOS 版本後，檢閱自訂 SIP 設定，確保安全配置的持續相容性與功能持續正常。

- 對於 x86 Mac 執行個體，SIP 設定會於執行個體層級套用。任何連接到執行個體的根磁碟區都會自動繼承已設定的 SIP 設定。

  對於 Apple 晶片 Mac 執行個體，SIP 設定會在磁碟區層級套用。連接到執行個體的根磁碟區不會繼承 SIP 設定。若您連接到另一個根磁碟區，則必須將 SIP 設定重新設定為所需的狀態。

- SIP 組態任務最長可能需要 90 分鐘才能完成。SIP 組態任務執行期間，執行個體將處於無法存取的狀態。
- SIP 組態不會轉移至您後續從該執行個體建立的快照或 AMI。
- Apple 晶片 Mac 執行個體只能有一個可啟動磁碟區，而且每個連接的磁碟區只能有一個額外的管理員使用者。

預設 SIP 組態

下列表格列出 x86 Mac 執行個體與 Apple 晶片 Mac 執行個體的預設 SIP 配置。

	Apple 晶片 Mac 執行個體	x86 Mac 執行個體
Apple 內部	已啟用	Disabled
檔案系統保護	已啟用	Disabled
基礎系統	已啟用	已啟用
偵錯限制	已啟用	已啟用
Dtrace 限制	已啟用	已啟用
Kext 簽署	已啟用	已啟用
Nvram 保護	已啟用	已啟用

檢查您的 SIP 組態

我們建議您在進行變更前後檢查 SIP 組態，確保其設定符合預期。

檢查 Amazon EC2 Mac 執行個體的 SIP 組態

使用 SSH 連線至執行個體，然後在命令列執行下列命令。

```
$ csrutil status
```

以下為範例輸出。

```
System Integrity Protection status: enabled.

Configuration:
    Apple Internal: enabled
    Kext Signing: disabled
    Filesystem Protections: enabled
    Debugging Restrictions: enabled
    DTrace Restrictions: enabled
```

```
    NVRAM Protections: enabled
    BaseSystem Verification: disabled
```

Apple 晶片 Mac 執行個體的先決條件

設定 Apple 晶片 Mac 執行個體的 SIP 設定前，您必須為 Amazon EBS 根磁碟區管理使用者 (ec2-user) 設定密碼並啟用安全字符。

> **Note**
>
> 密碼和安全字符在您第一次使用 GUI 連接到 Apple 晶片 Mac 執行個體時設定。若您先前已透過 GUI 連線至執行個體，或使用的是 x86 Mac 執行個體，則無需執行這些步驟。

> **Note**
>
> 用於 macOS 身分驗證的所有 macOS 使用者名稱和密碼都必須介於 4 到 16 個字元之間，才能與 SIP 設定 API 呼叫搭配使用。

為 EBS 根磁碟區管理使用者設定密碼並啟用安全字符

1. 使用 SSH 連線至執行個體。
2. 設定 ec2-user 使用者的密碼。

   ```
   $ sudo /usr/bin/dscl . -passwd /Users/ec2-user
   ```

3. 為 ec2-user 使用者啟用安全字符。對於 -oldPassword，指定與上一個步驟相同的密碼。對於 -newPassword，請指定不同的密碼。下列命令假設您已將舊密碼與新密碼儲存在 .txt 檔案中。

   ```
   $ sysadminctl -oldPassword `cat old_password.txt` -newPassword `cat
    new_password.txt`
   ```

4. 確認安全字符已啟用。

   ```
   $ sysadminctl -secureTokenStatus ec2-user
   ```

設定 SIP 設定

當您為執行個體設定 SIP 設定時，可以啟用或停用所有 SIP 設定，也可指定選擇性啟用或停用特定 SIP 設定的自訂組態。

> **Note**
>
> 若您實作自訂組態，請連線至執行個體並驗證設定，確保您的需求已正確實施且運作符合預期。
>
> SIP 組態可能會隨 macOS 更新而變更。建議您在每次升級 macOS 版本後，檢閱自訂 SIP 設定，確保安全配置的持續相容性與功能持續正常。

要設定執行個體的 SIP 設定，您必須建立 SIP 組態任務。SIP 組態任務會指定執行個體的 SIP 設定。

為 Apple 晶片 Mac 執行個體建立 SIP 組態時，必須指定下列憑證：

- 內部磁碟的管理使用者
  - 使用者名稱 – 僅支援預設的管理使用者 (`aws-managed-user`)，並且預設會使用。您無法指定其他管理員使用者。
  - 密碼 – 若您未變更 `aws-managed-user` 的預設密碼，請指定預設密碼，即空白。否則，請指定您自訂的密碼。
- Amazon EBS 根磁碟區管理使用者
  - 使用者名稱 – 若您未變更預設管理使用者，請指定 `ec2-user`。否則，為您的管理使用者指定使用者名稱。
  - 密碼 – 您必須一律指定密碼。

使用下列方法建立 SIP 組態任務。

Console

使用主控台建立 SIP 組態任務

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取執行個體，然後選取 Amazon EC2 Mac 執行個體。
3. 在安全性索引標籤中，選擇修改 Mac、修改系統完整性保護。
4. 要啟用所有 SIP 設定，請選取啟用 SIP。要停用所有 SIP 設定，請清除啟用 SIP。

5. 若要指定選擇性啟用或停用特定 SIP 設定的自訂組態，請選取指定自訂 SIP 組態，隨後選取要啟用的 SIP 設定，或清除要停用的 SIP 設定。
6. 指定根磁碟區使用者和內部磁碟擁有者的憑證。
7. 選擇建立 SIP 修改任務。

AWS CLI

使用 建立 SIP 組態任務 AWS CLI

使用 create-mac-system-integrity-protection-modification-task 命令。

啟用或停用所有 SIP 設定

若要完全啟用或停用所有 SIP 設定，僅使用 `--mac-system-integrity-protection-status` 參數。

下列範例命令會啟用全部 SIP 設定。

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
--mac-credentials file://mac-credentials.json
```

指定自訂 SIP 組態

要指定選擇性啟用或停用特定 SIP 設定的自訂 SIP 組態，請指定 `--mac-system-integrity-protection-status` 和 `--mac-system-integrity-protection-configuration` 參數。在此情況下，使用 `mac-system-integrity-protection-status` 指定整體 SIP 狀態，並使用 `mac-system-integrity-protection-configuration` 選擇性啟用或停用個別 SIP 設定。

下列範例命令會建立 SIP 組態任務，以啟用 `NvramProtections` 和 `FilesystemProtections` 以外的所有 SIP 設定。

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status enabled \
--mac-system-integrity-protection-configuration
 "NvramProtections=disabled, FilesystemProtections=disabled" \
--mac-credentials file://mac-credentials.json
```

下列範例命令會建立 SIP 組態任務，以停用除 DtraceRestrictions 以外的所有 SIP 設定。

```
aws ec2 create-mac-system-integrity-protection-modification-task \
--instance-id i-0abcdef9876543210 \
--mac-system-integrity-protection-status disabled \
--mac-system-integrity-protection-configuration "DtraceRestrictions=enabled" \
--mac-credentials file://mac-credentials.json
```

mac-credentials.json 檔案的內容

之前範例中所參考 mac-credentials.json 檔案的內容如以下所示。

```
{
  "internalDiskPassword":"internal-disk-admin_password",
  "rootVolumeUsername":"root-volume-admin_username",
  "rootVolumePassword":"root-volume-admin_password"
}
```

檢查 SIP 組態任務狀態

使用下列其中一種方法來檢查 SIP 組態任務的狀態。

Console

使用主控台來檢視 SIP 組態設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取執行個體，然後選取 Amazon EC2 Mac 執行個體。
3. 在安全性索引標籤中，向下捲動至 Mac 修改任務區段。

AWS CLI

使用 檢查 SIP 組態任務的狀態 AWS CLI

使用 describe-mac-modification-tasks 命令。

尋找支援 Amazon EC2 Mac 專用主機的 macOS 版本

您可以檢視 Amazon EC2 Mac 專用主機支援的最新 macOS 版本。有了這項功能，您就能驗證專用主機是否可以支援使用偏好的 macOS 版本來啟動執行個體。

每個 macOS 版本在基礎 Apple Mac 上都需要最低韌體版本，才能成功開機。如果已配置的 Mac 專用主機長時間仍處於閒置狀態，或者其上有長時間執行的執行個體，則 Apple Mac 韌體版本可能會過期。

為了確保最新 macOS 版本的能力，您可以停止或終止已配置 Mac 專用主機上的執行個體。這會觸發主機的清理工作流程，並更新基礎 Apple Mac 上的韌體，以支援最新的 macOS 版本。當您停止或終止執行中的執行個體時，具有長時間執行之執行個體的專用主機將自動更新。

如需有關清理工作流程的詳細資訊，請參閱 停止或終止 Amazon EC2 Mac 執行個體。

如需有關啟動 Mac 執行個體的詳細資訊，請參閱 使用 AWS 管理主控台 或 啟動 Mac 執行個體 AWS CLI。

您可以使用 Amazon EC2 主控台或 AWS CLI，檢視有關已配置專用主機所支援最新 macOS 版本的資訊。

Console

使用主控台檢視專用主機的韌體資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用主機。
3. 在專用主機詳細資訊頁面的最新支援的 macOS 版本下方，您可以查看主機支援的最新 macOS 版本。

AWS CLI

使用 檢視專用主機韌體資訊 AWS CLI

使用 `describe-mac-hosts` 命令，以適當的 AWS 區域取代 `region`。

```
$ aws ec2 describe-mac-hosts --region us-east-1
  {
      "MacHosts": [
          {
              "HostId": "h-07879acf49EXAMPLE",
              "MacOSLatestSupportedVersions": [
                  "14.3",
                  "13.6.4",
                  "12.7.3"
              ]
```

```
            }
        ]
    }
```

訂閱 macOS AMI 通知

若要在發行新的 AMI 或 BridgeOS 更新後收到通知，請使用 Amazon SNS 訂閱通知。

如需有關 EC2 和 macOS AMI 的詳細資訊，請參閱 Amazon EC2 macOS AMI 的版本備註。

訂閱 macOS AMI 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須使用此區域，因為您要訂閱的 SNS 通知已在本區域中建立完成。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選擇 Create subscription (建立訂閱)。
5. 針對 Create subscription (建立訂閱) 對話方塊，執行下列作業：
   a. 針對 Topic ARN (主題 ARN)，複製並貼上下列 Amazon Resource Name (ARN) 之一：
      - **arn:aws:sns:us-east-1:898855652048:amazon-ec2-macos-ami-updates**
      - **arn:aws:sns:us-east-1:898855652048:amazon-ec2-bridgeos-updates**
   b. 在專案中，選擇下列其中一個項目：
      - 電子郵件：

        針對 Endpoint (端點)，輸入可用於接收通知的電子郵件地址。建立訂閱後，您將收到主旨行為 `AWS Notification - Subscription Confirmation` 的確認訊息。開啟電子郵件並選擇 Confirm subscription (確認訂閱) 完成訂閱。
      - 簡訊：

        對於 Endpoint (端點)，輸入可用於接收通知的電子郵件地址。
      - AWS Lambda、Amazon SQS、Amazon Data Firehose (將以 JSON 格式傳送通知)：

        在 Endpoint (端點) 中，請輸入 Lambda 函數、SQS 佇列或 Firehose 串流的 ARN，以供您接收通知。
   c. 選擇 Create subscription (建立訂閱)。

只要有 macOS AMI 發行，我們就會向 amazon-ec2-macos-ami-updates 主題的訂閱者傳送通知。只要 bridgeOS 有所變更，我們就會向 amazon-ec2-bridgeos-updates 主題的訂閱者傳送通知。如果您不想再接收這些通知，請使用下列程序來取消訂閱。

取消訂閱 macOS AMI 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須使用此區域，因為 SNS 通知已在本區域中建立完成。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選取訂閱，接著選取 Actions (動作)、Delete subscriptions (刪除訂閱)，提示確認時請選取 Delete (刪除)。

使用 AWS Systems Manager 參數存放區 API 擷取 macOS AMI IDs

在您啟動執行個體時，必須指定 AMI。AMI 專屬於 AWS 區域、作業系統和處理器架構。您可以在 中檢視所有 macOS AMIs， AWS 區域 並透過查詢 AWS Systems Manager 參數存放區 API 來擷取最新的 macOS AMI。使用這些公有參數時，您不需要手動查詢 macOS AMI ID。公有參數同時適用於 x86 和 ARM64 macOS AMIs，並可與您現有的 AWS CloudFormation 範本整合。

所需的許可

若要執行此動作，IAM 主體必須具有呼叫 ssm:GetParameter API 動作的權限。

AWS 區域 使用 檢視目前 中所有 macOS AMIs的清單 AWS CLI

使用下列 get-parameters-by-path 命令來檢視目前區域中所有 macOS AMI 的清單。

```
aws ssm get-parameters-by-path --path /aws/service/ec2-macos --recursive --query
 "Parameters[].Name"
```

使用 擷取最新主要 macOS AMI 的 AMI ID AWS CLI

將下列 get-parameter 命令搭配子參數 image_id 使用。在下列範例中，sonoma將 取代為 macOS 支援的主要版本、將 x86_64_mac取代為處理器，並將 region-code取代為您想要取得最新 macOS AMI ID AWS 區域 的支援。

```
aws ssm get-parameter --name /aws/service/ec2-macos/sonoma/x86_64_mac/latest/image_id
 --region region-code
```

如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的呼叫 macOS 的 AMI 公有參數。

Amazon EC2 macOS AMI 的版本備註

下列資訊提供有關 EC2 macOS AMI 預設包含套件的詳細資訊，並總結每個 EC2 macOS AMI 版本的變更。

如需如何訂閱 macOS AMI 通知的詳細資訊，請參閱 訂閱 macOS AMI 通知。

Mac 執行個體可以執行下列其中一個作業系統：

- macOS Mojave (版本 10.14) (僅限 x86 Mac 執行個體)
- macOS Catalina (版本 10.15) (僅限 x86 Mac 執行個體)
- macOS Big Sur (版本 11) (x86 和 M1 Mac 執行個體)
- macOS Monterey (版本 12) (x86 和 M1 Mac 執行個體)
- macOS Ventura (13 版) (所有 Mac 執行個體、M2 和 M2 Pro Mac 執行個體支援 macOS Ventura 13.2 版或更新版本)
- macOS Sonoma (版本 14) (所有 Mac 執行個體)
- macOS Sequoia (版本 15) (所有 Mac 執行個體)

> **Note**
>
> M4 與 M4 Pro Mac 執行個體支援 macOS Sequoia 15.6 版或更新版本。

核准 macOS Sequoia 的本機網路隱私權政策

macOS Sequoia (版本 15) 具有全新本機網路隱私權功能，將影響本機 IP 型服務使用者，包括 Amazon EC2 執行個體中繼資料服務 (IMDS)。

> **Important**
>
> 若要確保您能夠不間斷存取本機 IP 型服務，請使用下列步驟來核准本機網路隱私權政策。

核准本機網路隱私權政策

1. 連線至您的執行個體的圖形使用者介面 (GUI).

2. 依照畫面上的提示來核准本機網路隱私權政策。
3. 請在核准政策後建立 EC2 Mac 執行個體的 AMI。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

任何從新建立 AMI 啟動的 EC2 Mac 執行個體都會保留本機網路隱私權權限。

Amazon EC2 macOS AMI 中包含的預設套件

下表說明 EC2 macOS AMI 中預設包含的套件。

套件	版本備註
EC2 macOS Init	https://github.com/aws/ec2-macos-init/tags
EC2 macOS Utils	https://github.com/aws/ec2-macos-utils/tags
Amazon SSM 代理程式	https://github.com/aws/amazon-ssm-agent/releases
AWS Command Line Interface (AWS CLI) 第 2 版	https://raw.githubusercontent.com/aws/aws-cli/v2/CHANGELOG.rst
Xcode 的命令列工具	https://developer.apple.com/documentation/xcode-release-notes
Homebrew	https://github.com/Homebrew/brew/releases
EC2 執行個體連線	https://github.com/aws/aws-ec2-instance-connect-config/releases
Safari	https://developer.apple.com/documentation/safari-release-notes

Amazon EC2 macOS AMI 更新

下表說明 EC2 macOS AMI 版本中包含的變更。請注意，有些變更適用於所有的 EC2 macOS AMI，但有些僅適用於這些 AMI 的一部分。

EC2 macOS AMI 更新

發行版本	改變
2026.09.02	所有 AMI • 更新`awscli`至 2.36.36 • 已將 Homebrew 更新至 6.0.20 macOS Sonoma 14.8.9 (Apple 晶片) 的版本 • macOS Sonoma 14.8.9 的安全內容 • 將 Safari 更新至 26.6.1 • Safari 26.6.1 的安全內容 macOS Tahoe 26.6.2 的版本 • macOS Tahoe 26.6.2 的安全內容

2026.08.26	所有 AMI
	• 更新`awscli`至 2.36.27
	• 已將 Homebrew 更新至 6.0.18
	
	macOS Sonoma 14.8.9 的版本 (**`mac1.metal`** 僅限 Intel)
	• macOS Sonoma 14.8.9 的安全內容
	• 將 Safari 更新至 26.6.1
	  • Safari 26.6.1 的安全內容
	
	macOS Sequoia 15.7.9 的版本
	• macOS Sequoia 15.7.9 的安全內容
	• 將 Safari 更新至 26.6.1
	  • Safari 26.6.1 的安全內容

發行版本	改變
	macOS Tahoe 26.6.1 的版本 • macOS Tahoe 26.6.1 的安全內容

2026.08.13	所有 AMI
	• 更新`awscli`至 2.36.12
	• 已將 Homebrew 更新至 6.0.13
	
	macOS Sonoma 14.8.8 的版本 (**`mac1.metal`** 僅限 Intel)
	• macOS Sonoma 14.8.8 的安全內容
	• 將 Safari 更新至 26.6
	  • Safari 26.6 的安全內容
	
	macOS Sequoia 15.7.8 的版本
	• macOS Sequoia 15.7.8 的安全內容
	• 將 Safari 更新至 26.6
	  • Safari 26.6 的安全內容
	
	macOS Tahoe 26.6 的版本
	• macOS Tahoe 26.6 的安全內容
	• 命令列工具已更新至 26.6

2026.07.29	macOS Tahoe 26.5.2 的版本 • macOS Tahoe 26.5.2 的安全內容 • 更新`awscli`至 2.36.1 • 已將 Homebrew 更新至 6.0.11

發行版本	改變

2026.05.19	所有 AMI
	• 更新`awscli`至 2.34.46
	• Homebrew 已更新至 5.1.11
	
	macOS Sonoma 14.8.7 的版本
	• macOS Sonoma 14.8.7 的安全內容
	• 將 Safari 更新至 26.5
	  • Safari 26.5 的安全內容
	
	macOS Sequoia 15.7.7 的版本
	• macOS Sequoia 15.7.7 的安全內容
	• 將 Safari 更新至 26.5
	  • Safari 26.5 的安全內容
	
	macOS Tahoe 26.5 的版本
	• macOS Tahoe 26.5 的安全內容
	• 命令列工具已更新至 26.5

2026.04.20	macOS Tahoe 26.4.1 的版本 • macOS Tahoe 26.4.1 的安全內容 • 命令列工具已更新至 26.4.1 • 更新`awscli`至 2.34.32 • 已將 Homebrew 更新至 5.1.7

發行版本	改變

2026.04.16	所有 AMI
	• 更新`awscli`至 2.34.27
	• 更新`ec2-macos-init` 至 1.5.15
	• 已將 Homebrew 更新至 5.1.5
	
	macOS Sonoma 14.8.5 的版本
	• macOS Sonoma 14.8.5 的安全內容
	• 將 Safari 更新至 26.4
	  • Safari 26.4 的安全內容
	
	macOS Sequoia 15.7.5 的版本
	• macOS Sequoia 15.7.5 的安全內容
	• 將 Safari 更新至 26.4
	  • Safari 26.4 的安全內容
	
	macOS Tahoe 26.4 的版本
	• macOS Tahoe 26.4 的安全內容
	• 命令列工具已更新至 26.4

2026.03.17	macOS Tahoe 26.3.1 的版本 • macOS Tahoe 26 更新中的新功能 • 命令列工具已更新至 26.3 • 更新`awscli`至 2.34.10 • 已將 Homebrew 更新至 5.1.0

發行版本	改變

2026.03.03	所有 AMI
	• 更新`awscli`至 2.33.31
	• 已將 Homebrew 更新至 5.0.15
	
	macOS Sonoma 14.8.4 的版本
	• macOS Sonoma 14.8.4 的安全內容
	• 將 Safari 更新至 26.3
	  • Safari 26.3 的安全內容
	
	macOS Sequoia 15.7.4 的版本
	• macOS Sequoia 15.7.4 的安全內容
	• 將 Safari 更新至 26.3
	  • Safari 26.3 的安全內容
	
	macOS Tahoe 26.3 的版本
	• macOS Tahoe 26.3 的安全內容

發行版本	改變

2025.12.26	所有 AMI
	• 更新`awscli`至 2.32.19
	• 更新`amazon-ssm-agent` 至 3.3.3270.0
	• 已將 Homebrew 更新至 5.0.6
	
	macOS Sonoma 14.8.3 的版本
	• macOS Sonoma 14.8.3 的安全內容
	• 將 Safari 更新至 26.2
	  • Safari 26.2 的安全內容
	
	macOS Sequoia 15.7.3 的版本
	• macOS Sequoia 15.7.3 的安全內容
	• 將 Safari 更新至 26.2
	  • Safari 26.2 的安全內容
	• 命令列工具已更新至 26.2
	
	macOS Tahoe 26.2 的版本
	• macOS Tahoe 26.2 的安全內容
	• 命令列工具已更新至 26.2

發行版本	改變

2025.12.17	所有 AMI
	• 更新`awscli`至 2.32.16
	• 已將 Homebrew 更新至 5.0.5
	
	macOS Sonoma 14.8.2 的版本
	• macOS Sonoma 14.8.2 的安全內容
	• 將 Safari 更新至 26.1
	  • Safari 26.1 的安全內容
	
	macOS Sequoia 15.7.2 的版本
	• macOS Sequoia 15.7.2 的安全內容
	• 將 Safari 更新至 26.1
	  • Safari 26.1 的安全內容
	• 命令列工具已更新至 26.1
	
	macOS Tahoe 26.1 的版本
	• macOS Tahoe 26.1 的安全內容
	• 命令列工具已更新至 26.1

發行版本	改變

2025.11.18	所有 AMI
	• 更新`awscli`至 2.31.35
	• 更新`ec2-macos-init` 至 1.5.13
	• 更新`ec2-macos-utils` 至 1.0.7
	• 已將 Homebrew 更新至 5.0.1
	
	macOS Sonoma 14.8.1 的版本
	• macOS Sonoma 14.8.1 的安全內容
	• 將 Safari 更新至 26.0.1
	  • Safari 26.0.1 的安全內容
	
	macOS Sequoia 15.7.1 的版本
	• macOS Sequoia 15.7.1 的安全內容
	• 將 Safari 更新至 26.0.1
	  • Safari 18.6 的安全內容
	• 命令列工具已更新至 26.0
	
	macOS Tahoe 26.0.1 的版本
	• macOS Tahoe 26.0.1 的安全內容
	• 命令列工具已更新至 26.0

發行版本	改變
2025.09.04	所有 AMI • 更新`awscli`至 2.28.19 • 更新`ec2-instance-connect` 至 2.0.0-5 • 已將 Homebrew 更新至 4.6.7 macOS Ventura 13.7.8 的版本 • macOS Ventura 13.7.8 的安全內容 macOS Sonoma 14.7.8 的版本 • macOS Sonoma 14.7.8 的安全內容 macOS Sequoia 15.6.1 的版本 • macOS Sequoia 15.6.1 的安全內容

發行版本	改變

2025.08.05	所有 AMI
	• 更新`awscli`至 2.28.0
	• Homebrew 已更新至 4.5.13
	
	macOS Ventura 13.7.7 的版本
	• macOS Ventura 13.7.7 的安全內容
	• 將 Safari 更新至 18.6
	  • Safari 18.6 的安全內容
	
	macOS Sonoma 14.7.7 的版本
	• macOS Sonoma 14.7.7 的安全內容
	• 將 Safari 更新至 18.6
	  • Safari 18.6 的安全內容
	
	macOS Sequoia 15.6 的版本
	• macOS Sequoia 15.6 的安全內容
	• 已將 AWS ENA 乙太網路更新至 2.0.0

發行版本	改變

2025.06.27	所有 AMI
	• 更新`awscli`至 2.27.40
	• 已將 Homebrew 更新至 4.5.7
	• 將 x86_64 映像遷移至 AWS ENA Ethernet dext
	
	macOS Ventura 13.7.6 的版本
	• macOS Ventura 13.7.6 的安全內容
	• 將 Safari 更新至 18.5
	  • Safari 18.5 的安全內容
	
	macOS Sonoma 14.7.6 的版本
	• macOS Sonoma 14.7.6 的安全內容
	• 將 Safari 更新至 18.5
	  • Safari 18.5 的安全內容
	
	macOS Sequoia 15.5 的版本
	• macOS Sequoia 15.5 的安全內容
	• 命令列工具已更新至 16.4
	• 已將 AWS ENA 乙太網路更新至 1.0.9

發行版本	改變

2025.05.21	所有 AMI
	• 更新awscli至 2.27.7
	• 更新ec2-macos-init 至 1.5.11
	• 更新ec2-macos-utils 至 1.0.5
	• 已將 Homebrew 更新至 4.5.1
	
	macOS Ventura 13.7.5 的版本
	• macOS Ventura 13.7.5 的安全內容
	• 將 Safari 更新至 18.4
	  • Safari 18.4 的安全內容
	
	macOS Sonoma 14.7.5 的版本
	• macOS Sonoma 14.7.5 的安全內容
	• 將 Safari 更新至 18.4
	  • Safari 18.4 的安全內容
	
	macOS Sequoia 15.4.1 的版本
	• macOS Sequoia 15.4.1 的安全內容
	• 命令列工具已更新至 16.3

發行版本	改變

2025.05.05	所有 AMI
	• 更新`awscli`至 2.27.1
	• 更新`ec2-macos-init` 至 1.5.11
	• 更新`ec2-macos-system-monitor` 至 1.3.1
	• 更新`ec2-macos-utils` 至 1.0.5
	• Homebrew 已更新至 4.4.32
	
	macOS Ventura 13.7.5 的版本
	• macOS Ventura 13.7.5 的安全內容
	• 將 Safari 更新至 18.4
	  • Safari 18.4 的安全內容
	
	macOS Sonoma 14.7.5 的版本
	• macOS Sonoma 14.7.5 的安全內容
	• 將 Safari 更新至 18.4
	  • Safari 18.4 的安全內容
	
	macOS Sequoia 15.4.1 的版本
	• macOS Sequoia 15.4.1 的安全內容

發行版本	改變
2025.03.18	所有 AMI • 更新`awscli`至 2.24.2 • Homebrew 已更新至 4.4.20 macOS Sequoia 15.3.1 的版本 • macOS Sequoia 15.3.1 的安全內容 macOS Sonoma 14.7.4 的版本 • macOS Sonoma 14.7.4 的安全內容 • 將 Safari 更新至 18.3 macOS Ventura 13.7.4 的版本 • macOS Ventura 13.7.4 的安全內容 • 將 Safari 更新至 18.3

發行版本	改變
2025.01.24	所有 AMI • 更新`awscli`至 2.22.33 • Homebrew 已更新至 4.4.15 macOS Sequoia 15.2 的版本 • macOS Sequoia 15.2 的安全內容 • 命令列工具已更新至 16.2 macOS Sonoma 14.7.2 的版本 • macOS Sonoma 14.7.2 的安全內容 • 將 Safari 更新至 18.2 • 命令列工具已更新至 16.2 macOS Ventura 13.7.2 的版本 • macOS Ventura 13.7.2 的安全內容 • 將 Safari 更新至 18.2

發行版本	改變
2024.12.20	所有 AMI • 已將 Homebrew 更新至 4.4.8 • 更新`aws-cli`至 2.22.5 • `amazon-ssm-agent` 已更新到 3.3.987.0 macOS Sequoia 15.1.1 的版本 • macOS Sequoia 15.1.1 的安全內容 macOS Sonoma 14.7.1 的版本 • macOS Sonoma 14.7.1 的安全內容 • 將 Safari 更新至 18.1.1 macOS Ventura 13.7.1 的版本 • macOS Ventura 13.7.1 的安全內容 • 將 Safari 更新至 18.1.1

發行版本	改變

2024.10.28	所有 AMI
	• Homebrew 已更新到 4.4.2
	• `aws-cli` 已更新到 2.18.13
	• `amazon-ssm-agent` 已更新到 3.3.987.0
	• `ec2-macos-init` 已更新到 1.5.10
	• `ec2-macos-utils` 已更新到 1.0.4
	
	macOS Sequoia 15.0 的版本
	• macOS Sequoia 15 的安全內容
	
	macOS Sonoma 14.7 的版本
	• macOS Sonoma 14.7 的安全內容。
	• 命令列工具已更新到 16.0
	• Safari 已更新到 18.0.1
	  • Safari 18 的安全內容
	
	macOS Ventura 13.7 的版本
	• macOS Ventura 13.7 的安全內容
	• Safari 已更新到 18.0.1
	  • Safari 18 的安全內容

發行版本	改變

2024.08.20	所有 AMI
	• Homebrew 已更新到 4.3.14
	• `aws-cli` 已更新到 2.17.29
	
	macOS Sonoma 14.6.1 的版本
	• 沒有已發佈的 CVE 項目。
	
	macOS Ventura 13.6.9 的版本
	• 沒有已發佈的 CVE 項目。
	• Safari 已更新到 17.6
	  • Safari 17.6 的安全內容
	
	macOS Monterey 12.7.6 的版本
	• macOS Monterey 12.7.6 的安全內容
	• Safari 已更新到 17.6
	  • Safari 17.6 的安全內容

發行版本	改變

2024.06.07	所有 AMI
	• Homebrew 已更新到 4.3.1-1
	• `aws-cli` 已更新到 2.15.56
	• `amazon-ssm-agent` 已更新到 3.3.380.0-1
	
	macOS Sonoma 14.5 的版本
	• macOS Sonoma 14.5 的安全內容
	
	macOS Ventura 13.6.7 的版本
	• macOS Ventura 13.6.7 的安全內容
	• Safari 已更新到 17.5
	  • Safari 17.5 的安全內容
	
	macOS Monterey 12.7.5 的版本
	• macOS Monterey 12.7.5 的安全內容
	• Safari 已更新到 17.5
	  • Safari 17.5 的安全內容

發行版本	改變

2024.04.12	所有 AMI
	• Homebrew 已更新到 4.2.16-1
	• `aws-cli` 已更新到 2.15.36
	
	macOS Sonoma 14.4.1 的版本
	• macOS Sonoma 14.4.1 的安全內容
	
	macOS Ventura 13.6.6 的版本
	• macOS Ventura 13.6.6 的安全內容
	• Safari 已更新到 17.4.1
	  • Safari 17.4.1 的安全內容
	
	適用於 macOS Monterey
	• Safari 已更新到 17.4.1
	  • Safari 17.4.1 的安全內容

Amazon EBS 最佳化的執行個體類型

Amazon EBS 最佳化執行個體使用最佳化的組態堆疊，並為 Amazon EBS I/O 提供額外的專用頻寬。此最佳化透過盡可能減少 Amazon EBS I/O 與執行個體的其他流量之間的爭用情況，為您的 EBS 磁碟區提供最佳效能。

當連接至 EBS 最佳化執行個體時，一般用途 SSD (gp2 和 gp3) 磁碟區設計為能在給定年份 99% 的時間裡提供至少 90% 的佈建 IOPS 效能，佈建 IOPS SSD (io1 和 io2) 磁碟區則設計為能在給定年份 99.9% 的時間裡提供至少 90% 的佈建 IOPS 效能。輸送量最佳化 HDD (st1)和冷 HDD (sc1) 在給定年份 99% 的時間裡提供至少 90% 的預期輸送量效能。不相容的期間約為統一分佈，目標為每小時 99% 的預期總輸送量。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 磁碟區類型。

某些執行個體類型預設為 EBS 最佳化，無需啟用，且嘗試停用時也不會產生任何影響。其他執行個體類型可選擇支援 EBS 最佳化，您可以在啟動期間或之後啟用，需支付額外小時費用。部分執行個體規格不支援 EBS 最佳化。

如需詳細的執行個體類型規格與功能，請參閱《Amazon EC2 執行個體類型指南》。

考量事項

- 執行個體的 EBS 效能受限於執行個體類型效能上限，或其連接磁碟區的彙總效能 (以較小者為準)。若要達到最大 EBS 效能，執行個體必須具有連接磁碟區，這些磁碟區提供的組合效能必須等於或大於最大執行個體效能。例如，`r6i.16xlarge` 若要達到 `80,000` IOPS，每個執行個體至少須佈建 5 gp2 個 `16,000` IOPS 的磁碟區 (5 個磁碟區 x `16,000` IOPS = `80,000` IOPS)，或者也可配置 1 gp3 個磁碟區並提供 `80,000` IOPS。我們建議您選擇提供的專屬 Amazon EBS 輸送量高於應用程式所需的執行個體類型；否則，Amazon EBS 和 Amazon EC2 之間的連線將成為效能瓶頸。
- 可連接到執行個體的 Amazon EBS 磁碟區數目上限，取決於執行個體類型和執行個體大小。如需詳細資訊，請參閱Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。
- IOPS 上限和輸送量限制相互依存。根據 I/O 大小，您可能會在達到另一個限制之前先達到其中一個，進而影響整體效能。若要達到最佳效果，規劃工作負載時請同時考量這兩項限制。

預設為 EBS 最佳化

下列執行個體類型預設為 EBS 最佳化。不需要啟用 EBS 最佳化，且停用 EBS 最佳化不具任何效果。

執行個體

- 一般用途
- 運算最佳化
- 記憶體最佳化
- 儲存最佳化
- 加速運算
- 高效能運算

一般用途

> **Note**
>
> M8a、M8g、M8gd、M8i、M8id、M8i-flex、M9g、M9gd 執行個體類型支援可設定的頻寬加權。透過這些執行個體類型，您可針對聯網效能或 Amazon EBS 效能最佳化執行個體的頻寬。

下列表格列出這些執行個體類型的預設 Amazon EBS 頻寬效能。如需詳細資訊，請參閱EC2 執行個體頻寬加權組態。

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
a1.medium [1]	300	3500	37.50	437.50	2500	20000
a1.large [1]	525	3500	65.62	437.50	4000	20000
a1.xlarge [1]	800	3500	100.00	437.50	6000	20000
a1.2xlarge [1]	1750	3500	218.75	437.50	10000	20000
a1.4xlarge [2]	3500		437.5		20000	
a1.metal [2]	3500		437.5		20000	
m4.large [2]	450		56.25		3600	
m4.xlarge [2]	750		93.75		6000	
m4.2xlarge [2]	1000		125.0		8000	
m4.4xlarge [2]	2000		250.0		16000	
m4.10xlarge [2]	4000		500.0		32000	
m4.16xlarge [2]	10000		1250.0		65000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5.large [1]	650	4750	81.25	593.75	3600	18750
m5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5.4xlarge [2]	4750		593.75		18750	
m5.8xlarge [2]	6800		850.0		30000	
m5.12xlarge [2]	9500		1187.5		40000	
m5.16xlarge [2]	13600		1700.0		60000	
m5.24xlarge [2]	19000		2375.0		80000	
m5.metal [2]	19000		2375.0		80000	
m5a.large [1]	650	2880	81.25	360.00	3600	16000
m5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5a.4xlarge [2]	2880		360.0		16000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5a.8xlarge [2]	4750		593.75		20000	
m5a.12xlarge [2]	6780		847.5		30000	
m5a.16xlarge [2]	9500		1187.5		40000	
m5a.24xlarge [2]	13750		1718.75		60000	
m5ad.large [1]	650	2880	81.25	360.00	3600	16000
m5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5ad.4xlarge [2]	2880		360.0		16000	
m5ad.8xlarge [2]	4750		593.75		20000	
m5ad.12xlarge [2]	6780		847.5		30000	
m5ad.16xlarge [2]	9500		1187.5		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5ad.24xlarge [2]	13750		1718.75		60000	
m5d.large [1]	650	4750	81.25	593.75	3600	18750
m5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5d.4xlarge [2]	4750		593.75		18750	
m5d.8xlarge [2]	6800		850.0		30000	
m5d.12xlarge [2]	9500		1187.5		40000	
m5d.16xlarge [2]	13600		1700.0		60000	
m5d.24xlarge [2]	19000		2375.0		80000	
m5d.metal [2]	19000		2375.0		80000	
m5dn.large [1]	650	4750	81.25	593.75	3600	18750

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5dn.4xlarge [2]	4750		593.75		18750	
m5dn.8xlarge [2]	6800		850.0		30000	
m5dn.12xlarge [2]	9500		1187.5		40000	
m5dn.16xlarge [2]	13600		1700.0		60000	
m5dn.24xlarge [2]	19000		2375.0		80000	
m5dn.metal [2]	19000		2375.0		80000	
m5n.large [1]	650	4750	81.25	593.75	3600	18750
m5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5n.4xlarge [2]	4750		593.75		18750	
m5n.8xlarge [2]	6800		850.0		30000	
m5n.12xlarge [2]	9500		1187.5		40000	
m5n.16xlarge [2]	13600		1700.0		60000	
m5n.24xlarge [2]	19000		2375.0		80000	
m5n.metal [2]	19000		2375.0		80000	
m5zn.large [1]	800	3170	100.00	396.25	3333	13333
m5zn.xlarge [1]	1564	3170	195.50	396.25	6667	13333
m5zn.2xlarge [2]	3170		396.25		13333	
m5zn.3xlarge [2]	4750		593.75		20000	
m5zn.6xlarge [2]	9500		1187.5		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m5zn.12xlarge [2]	19000		2375.0		80000	
m5zn.metal [2]	19000		2375.0		80000	
m6a.large [1]	650	10000	81.25	1250.00	3600	40000
m6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6a.8xlarge [2]	10000		1250.0		40000	
m6a.12xlarge [2]	15000		1875.0		60000	
m6a.16xlarge [2]	20000		2500.0		80000	
m6a.24xlarge [2]	30000		3750.0		120000	
m6a.32xlarge [2]	40000		5000.0		160000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6a.48xlarge [2]	40000		5000.0		240000	
m6a.metal [2]	40000		5000.0		240000	
m6g.medium [1]	315	4750	39.38	593.75	2500	20000
m6g.large [1]	630	4750	78.75	593.75	3600	20000
m6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6g.4xlarge [2]	4750		593.75		20000	
m6g.8xlarge [2]	9500		1187.5		40000	
m6g.12xlarge [2]	14250		1781.25		50000	
m6g.16xlarge [2]	19000		2375.0		80000	
m6g.metal [2]	19000		2375.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6gd.medium [1]	315	4750	39.38	593.75	2500	20000
m6gd.large [1]	630	4750	78.75	593.75	3600	20000
m6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6gd.4xlarge [2]	4750		593.75		20000	
m6gd.8xlarge [2]	9500		1187.5		40000	
m6gd.12xlarge [2]	14250		1781.25		50000	
m6gd.16xlarge [2]	19000		2375.0		80000	
m6gd.metal [2]	19000		2375.0		80000	
m6i.large [1]	650	10000	81.25	1250.00	3600	40000
m6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6i.8xlarge [2]	10000		1250.0		40000	
m6i.12xlarge [2]	15000		1875.0		60000	
m6i.16xlarge [2]	20000		2500.0		80000	
m6i.24xlarge [2]	30000		3750.0		120000	
m6i.32xlarge [2]	40000		5000.0		160000	
m6i.metal [2]	40000		5000.0		160000	
m6id.large [1]	650	10000	81.25	1250.00	3600	40000
m6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6id.8xlarge [2]	10000		1250.0		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6id.12xlarge [2]	15000		1875.0		60000	
m6id.16xlarge [2]	20000		2500.0		80000	
m6id.24xlarge [2]	30000		3750.0		120000	
m6id.32xlarge [2]	40000		5000.0		160000	
m6id.metal [2]	40000		5000.0		160000	
m6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
m6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6idn.8xlarge [2]	25000		3125.0		100000	
m6idn.12xlarge [2]	37500		4687.5		150000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6idn.16xlarge [2]	50000		6250.0		200000	
m6idn.24xlarge [2]	75000		9375.0		300000	
m6idn.32xlarge [2]	100000		12500.0		400000	
m6idn.metal [2]	100000		12500.0		400000	
m6in.large [1]	1562	25000	195.31	3125.00	6250	100000
m6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6in.8xlarge [2]	25000		3125.0		100000	
m6in.12xlarge [2]	37500		4687.5		150000	
m6in.16xlarge [2]	50000		6250.0		200000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m6in.24xlarge [2]	75000		9375.0		300000	
m6in.32xlarge [2]	100000		12500.0		400000	
m6in.metal [2]	100000		12500.0		400000	
m7a.medium [1]	325	10000	40.62	1250.00	2500	40000
m7a.large [1]	650	10000	81.25	1250.00	3600	40000
m7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7a.8xlarge [2]	10000		1250.0		40000	
m7a.12xlarge [2]	15000		1875.0		60000	
m7a.16xlarge [2]	20000		2500.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m7a.24xlarge [2]	30000		3750.0		120000	
m7a.32xlarge [2]	40000		5000.0		160000	
m7a.48xlarge [2]	40000		5000.0		240000	
m7a.metal-48xl [2]	40000		5000.0		240000	
m7g.medium [1]	315	10000	39.38	1250.00	2500	40000
m7g.large [1]	630	10000	78.75	1250.00	3600	40000
m7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7g.8xlarge [2]	10000		1250.0		40000	
m7g.12xlarge [2]	15000		1875.0		60000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m7g.16xlarge [2]	20000		2500.0		80000	
m7g.metal [2]	20000		2500.0		80000	
m7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m7gd.large [1]	630	10000	78.75	1250.00	3600	40000
m7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7gd.8xlarge [2]	10000		1250.0		40000	
m7gd.12xlarge [2]	15000		1875.0		60000	
m7gd.16xlarge [2]	20000		2500.0		80000	
m7gd.metal [2]	20000		2500.0		80000	
m7i.large [1]	650	10000	81.25	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i.8xlarge [2]	10000		1250.0		40000	
m7i.12xlarge [2]	15000		1875.0		60000	
m7i.16xlarge [2]	20000		2500.0		80000	
m7i.24xlarge [2]	30000		3750.0		120000	
m7i.48xlarge [2]	40000		5000.0		240000	
m7i.metal-24xl [2]	30000		3750.0		120000	
m7i.metal-48xl [2]	40000		5000.0		240000	
m7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
m7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8a.medium [1]	325	10000	40.62	1250.00	2500	40000
m8a.large [1]	650	10000	81.25	1250.00	3600	40000
m8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8a.8xlarge [2]	10000		1250.0		40000	
m8a.12xlarge [2]	15000		1875.0		60000	
m8a.16xlarge [2]	20000		2500.0		80000	
m8a.24xlarge [2]	30000		3750.0		120000	
m8a.48xlarge [2]	60000		7500.0		240000	
m8a.metal-24xl [2]	30000		3750.0		120000	
m8a.metal-48xl [2]	60000		7500.0		240000	
m8azn.medium [1]	625	15000	78.12	1875.00	2500	60000
m8azn.large [1]	1250	15000	156.25	1875.00	5000	60000
m8azn.xlarge [1]	2500	15000	312.50	1875.00	10000	60000
m8azn.3xlarge [1]	7500	15000	937.50	1875.00	30000	60000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8azn.6xlarge [2]	15000		1875.0		60000	
m8azn.12xlarge [2]	30000		3750.0		120000	
m8azn.24xlarge [2]	60000		7500.0		240000	
m8azn.metal-12xl [2]	30000		3750.0		120000	
m8azn.metal-24xl [2]	60000		7500.0		240000	
m8g.medium [1]	315	10000	39.38	1250.00	2500	40000
m8g.large [1]	630	10000	78.75	1250.00	3600	40000
m8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8g.8xlarge [2]	10000		1250.0		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8g.12xlarge [2]	15000		1875.0		60000	
m8g.16xlarge [2]	20000		2500.0		80000	
m8g.24xlarge [2]	30000		3750.0		120000	
m8g.48xlarge [2]	40000		5000.0		240000	
m8g.metal-24xl [2]	30000		3750.0		120000	
m8g.metal-48xl [2]	40000		5000.0		240000	
m8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
m8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
m8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000
m8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
m8gb.4xlarge [2]	25000		3125.0		120000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8gb.8xlarge [2]	50000		6250.0		240000	
m8gb.12xlarge [2]	75000		9375.0		360000	
m8gb.16xlarge [2]	100000		12500.0		480000	
m8gb.24xlarge [2]	150000		18750.0		720000	
m8gb.48xlarge [2]	300000		37500.0		1440000	
m8gb.metal-24xl [2]	150000		18750.0		720000	
m8gb.metal-48xl [2]	300000		37500.0		1440000	
m8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m8gd.large [1]	630	10000	78.75	1250.00	3600	40000
m8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8gd.8xlarge [2]	10000		1250.0		40000	
m8gd.12xlarge [2]	15000		1875.0		60000	
m8gd.16xlarge [2]	20000		2500.0		80000	
m8gd.24xlarge [2]	30000		3750.0		120000	
m8gd.48xlarge [2]	40000		5000.0		240000	
m8gd.metal-24xl [2]	30000		3750.0		120000	
m8gd.metal-48xl [2]	40000		5000.0		240000	
m8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
m8gn.large [1]	1250	10000	156.25	1250.00	5000	40000
m8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8gn.4xlarge [2]	10000		1250.0		40000	
m8gn.8xlarge [2]	20000		2500.0		80000	
m8gn.12xlarge [2]	30000		3750.0		120000	
m8gn.16xlarge [2]	40000		5000.0		160000	
m8gn.24xlarge [2]	60000		7500.0		240000	
m8gn.48xlarge [2]	120000		15000.0		480000	
m8gn.metal-24xl [2]	60000		7500.0		240000	
m8gn.metal-48xl [2]	120000		15000.0		480000	
m8i.large [1]	650	10000	81.25	1250.00	3600	40000
m8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8i.8xlarge [2]	10000		1250.0		40000	
m8i.12xlarge [2]	15000		1875.0		60000	
m8i.16xlarge [2]	20000		2500.0		80000	
m8i.24xlarge [2]	30000		3750.0		120000	
m8i.32xlarge [2]	40000		5000.0		160000	
m8i.48xlarge [2]	60000		7500.0		240000	
m8i.96xlarge [2]	80000		10000.0		480000	
m8i.metal-48xl [2]	60000		7500.0		240000	
m8i.metal-96xl [2]	80000		10000.0		480000	
m8id.large [1]	650	10000	81.25	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8id.8xlarge [2]	10000		1250.0		40000	
m8id.12xlarge [2]	15000		1875.0		60000	
m8id.16xlarge [2]	20000		2500.0		80000	
m8id.24xlarge [2]	30000		3750.0		120000	
m8id.32xlarge [2]	40000		5000.0		160000	
m8id.48xlarge [2]	60000		7500.0		240000	
m8id.96xlarge [2]	80000		10000.0		480000	
m8id.metal-48xl [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8id.metal-96xl [2]	80000		10000.0		480000	
m8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
m8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
m8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8in.large [1]	650	10000	81.25	1250.00	3600	40000
m8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8in.8xlarge [2]	10000		1250.0		40000	
m8in.12xlarge [2]	15000		1875.0		60000	
m8in.16xlarge [2]	20000		2500.0		80000	
m8in.24xlarge [2]	30000		3750.0		120000	
m8in.32xlarge [2]	40000		5000.0		160000	
m8in.48xlarge [2]	60000		7500.0		240000	
m8in.96xlarge [2]	120000		15000.0		480000	
m8in.metal-48xl [2]	60000		7500.0		240000	
m8in.metal-96xl [2]	120000		15000.0		480000	
m8idn.large [1]	650	10000	81.25	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8idn.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8idn.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8idn.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8idn.8xlarge [2]	10000		1250.0		40000	
m8idn.12xlarge [2]	15000		1875.0		60000	
m8idn.16xlarge [2]	20000		2500.0		80000	
m8idn.24xlarge [2]	30000		3750.0		120000	
m8idn.32xlarge [2]	40000		5000.0		160000	
m8idn.48xlarge [2]	60000		7500.0		240000	
m8idn.96xlarge [2]	120000		15000.0		480000	
m8idn.metal-48xl [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8idn.metal-96xl [2]	120000		15000.0		480000	
m8ine.large [1]	650	10000	81.25	1250.00	3600	40000
m8ine.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8ine.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8ine.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8ine.8xlarge [2]	10000		1250.0		40000	
m8ine.12xlarge [2]	15000		1875.0		60000	
m8ib.large [1]	1563	25000	195.38	3125.00	7500	120000
m8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
m8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
m8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8ib.8xlarge [2]	25000		3125.0		120000	
m8ib.12xlarge [2]	37500		4687.5		180000	
m8ib.16xlarge [2]	50000		6250.0		240000	
m8ib.24xlarge [2]	75000		9375.0		360000	
m8ib.32xlarge [2]	100000		12500.0		480000	
m8ib.48xlarge [2]	150000		18750.0		720000	
m8ib.96xlarge [2]	300000		37500.0		1440000	
m8ib.metal-48xl [2]	150000		18750.0		720000	
m8ib.metal-96xl [2]	300000		37500.0		1440000	
m8idb.large [1]	1563	25000	195.38	3125.00	7500	120000
m8idb.xlarge [1]	3125	25000	390.62	3125.00	15000	120000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m8idb.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
m8idb.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
m8idb.8xlarge [2]	25000		3125.0		120000	
m8idb.12xlarge [2]	37500		4687.5		180000	
m8idb.16xlarge [2]	50000		6250.0		240000	
m8idb.24xlarge [2]	75000		9375.0		360000	
m8idb.32xlarge [2]	100000		12500.0		480000	
m8idb.48xlarge [2]	150000		18750.0		720000	
m8idb.96xlarge [2]	300000		37500.0		1440000	
m8idb.metal-48xl [2]	150000		18750.0		720000	
m8idb.metal-96xl [2]	300000		37500.0		1440000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m9g.medium [1]	380	12000	47.50	1500.00	2500	48000
m9g.large [1]	760	12000	95.00	1500.00	3600	48000
m9g.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
m9g.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
m9g.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
m9g.8xlarge [2]	12000		1500.0		48000	
m9g.12xlarge [2]	18000		2250.0		72000	
m9g.16xlarge [2]	24000		3000.0		96000	
m9g.24xlarge [2]	36000		4500.0		144000	
m9g.48xlarge [2]	72000		9000.0		288000	
m9g.metal-48xl [2]	72000		9000.0		288000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
m9gd.medium [1]	380	12000	47.50	1500.00	2500	48000
m9gd.large [1]	760	12000	95.00	1500.00	3600	48000
m9gd.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
m9gd.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
m9gd.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
m9gd.8xlarge [2]	12000		1500.0		48000	
m9gd.12xlarge [2]	18000		2250.0		72000	
m9gd.16xlarge [2]	24000		3000.0		96000	
m9gd.24xlarge [2]	36000		4500.0		144000	
m9gd.48xlarge [2]	72000		9000.0		288000	
m9gd.metal-48xl [2]	72000		9000.0		288000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
mac1.metal [2]	14000		1750.0		80000	
mac2.metal [2]	10000		1250.0		55000	
mac2-m1ultra.metal [2]	10000		1250.0		55000	
mac2-m2.metal [2]	8000		1000.0		55000	
mac2-m2pro.metal [2]	8000		1000.0		55000	
mac-m3ultra.metal [2]	10000		1250.0		55000	
mac-m4.metal [2]	8000		1000.0		55000	
mac-m4pro.metal [2]	8000		1000.0		55000	
mac-m4max.metal [2]	10000		1250.0		55000	
t3.nano [1]	43	2085	5.38	260.62	250	11800
t3.micro [1]	87	2085	10.88	260.62	500	11800
t3.small [1]	174	2085	21.75	260.62	1000	11800

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
t3.medium [1]	347	2085	43.38	260.62	2000	11800
t3.large [1]	695	2780	86.88	347.50	4000	15700
t3.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.nano [1]	45	2085	5.62	260.62	250	11800
t3a.micro [1]	90	2085	11.25	260.62	500	11800
t3a.small [1]	175	2085	21.88	260.62	1000	11800
t3a.medium [1]	350	2085	43.75	260.62	2000	11800
t3a.large [1]	695	2780	86.88	347.50	4000	15700
t3a.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.nano [1]	43	2085	5.38	260.62	250	11800
t4g.micro [1]	87	2085	10.88	260.62	500	11800
t4g.small [1]	174	2085	21.75	260.62	1000	11800
t4g.medium [1]	347	2085	43.38	260.62	2000	11800
t4g.large [1]	695	2780	86.88	347.50	4000	15700

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
t4g.xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.2xlarge [1]	695	2780	86.88	347.50	4000	15700

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

運算最佳化

> **Note**
>
> C8a、C8g、C8gd、C8i、C8id、C8i-flex、C9g、C9gd 執行個體類型支援可設定的頻寬加權。透過這些執行個體類型，您可針對聯網效能或 Amazon EBS 效能最佳化執行個體的頻寬。下列表格列出這些執行個體類型的預設 Amazon EBS 頻寬效能。如需詳細資訊，請參閱EC2 執行個體頻寬加權組態。

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c4.large [2]	500		62.5		4000	
c4.xlarge [2]	750		93.75		6000	
c4.2xlarge [2]	1000		125.0		8000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c4.4xlarge [2]	2000		250.0		16000	
c4.8xlarge [2]	4000		500.0		32000	
c5.large [1]	650	4750	81.25	593.75	4000	20000
c5.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5.4xlarge [2]	4750		593.75		20000	
c5.9xlarge [2]	9500		1187.5		40000	
c5.12xlarge [2]	9500		1187.5		40000	
c5.18xlarge [2]	19000		2375.0		80000	
c5.24xlarge [2]	19000		2375.0		80000	
c5.metal [2]	19000		2375.0		80000	
c5a.large [1]	200	3170	25.00	396.25	800	13300
c5a.xlarge [1]	400	3170	50.00	396.25	1600	13300

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c5a.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5a.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5a.8xlarge [2]	3170		396.25		13300	
c5a.12xlarge [2]	4750		593.75		20000	
c5a.16xlarge [2]	6300		787.5		26700	
c5a.24xlarge [2]	9500		1187.5		40000	
c5ad.large [1]	200	3170	25.00	396.25	800	13300
c5ad.xlarge [1]	400	3170	50.00	396.25	1600	13300
c5ad.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5ad.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5ad.8xlarge [2]	3170		396.25		13300	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c5ad.12xlarge [2]	4750		593.75		20000	
c5ad.16xlarge [2]	6300		787.5		26700	
c5ad.24xlarge [2]	9500		1187.5		40000	
c5d.large [1]	650	4750	81.25	593.75	4000	20000
c5d.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5d.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5d.4xlarge [2]	4750		593.75		20000	
c5d.9xlarge [2]	9500		1187.5		40000	
c5d.12xlarge [2]	9500		1187.5		40000	
c5d.18xlarge [2]	19000		2375.0		80000	
c5d.24xlarge [2]	19000		2375.0		80000	
c5d.metal [2]	19000		2375.0		80000	
c5n.large [1]	650	4750	81.25	593.75	4000	20000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c5n.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5n.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5n.4xlarge [2]	4750		593.75		20000	
c5n.9xlarge [2]	9500		1187.5		40000	
c5n.18xlarge [2]	19000		2375.0		80000	
c5n.metal [2]	19000		2375.0		80000	
c6a.large [1]	650	10000	81.25	1250.00	3600	40000
c6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6a.8xlarge [2]	10000		1250.0		40000	
c6a.12xlarge [2]	15000		1875.0		60000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6a.16xlarge [2]	20000		2500.0		80000	
c6a.24xlarge [2]	30000		3750.0		120000	
c6a.32xlarge [2]	40000		5000.0		160000	
c6a.48xlarge [2]	40000		5000.0		240000	
c6a.metal [2]	40000		5000.0		240000	
c6g.medium [1]	315	4750	39.38	593.75	2500	20000
c6g.large [1]	630	4750	78.75	593.75	3600	20000
c6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6g.4xlarge [2]	4750		593.75		20000	
c6g.8xlarge [2]	9500		1187.5		40000	
c6g.12xlarge [2]	14250		1781.25		50000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6g.16xlarge [2]	19000		2375.0		80000	
c6g.metal [2]	19000		2375.0		80000	
c6gd.medium [1]	315	4750	39.38	593.75	2500	20000
c6gd.large [1]	630	4750	78.75	593.75	3600	20000
c6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6gd.4xlarge [2]	4750		593.75		20000	
c6gd.8xlarge [2]	9500		1187.5		40000	
c6gd.12xlarge [2]	14250		1781.25		50000	
c6gd.16xlarge [2]	19000		2375.0		80000	
c6gd.metal [2]	19000		2375.0		80000	
c6gn.medium [1]	760	9500	95.00	1187.50	2500	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6gn.large [1]	1235	9500	154.38	1187.50	5000	40000
c6gn.xlarge [1]	2375	9500	296.88	1187.50	10000	40000
c6gn.2xlarge [1]	4750	9500	593.75	1187.50	20000	40000
c6gn.4xlarge [2]	9500		1187.5		40000	
c6gn.8xlarge [2]	19000		2375.0		80000	
c6gn.12xlarge [2]	28500		3562.5		120000	
c6gn.16xlarge [2]	38000		4750.0		160000	
c6i.large [1]	650	10000	81.25	1250.00	3600	40000
c6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6i.8xlarge [2]	10000		1250.0		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6i.12xlarge [2]	15000		1875.0		60000	
c6i.16xlarge [2]	20000		2500.0		80000	
c6i.24xlarge [2]	30000		3750.0		120000	
c6i.32xlarge [2]	40000		5000.0		160000	
c6i.metal [2]	40000		5000.0		160000	
c6id.large [1]	650	10000	81.25	1250.00	3600	40000
c6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6id.8xlarge [2]	10000		1250.0		40000	
c6id.12xlarge [2]	15000		1875.0		60000	
c6id.16xlarge [2]	20000		2500.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6id.24xlarge [2]	30000		3750.0		120000	
c6id.32xlarge [2]	40000		5000.0		160000	
c6id.metal [2]	40000		5000.0		160000	
c6in.large [1]	1562	25000	195.31	3125.00	6250	100000
c6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
c6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
c6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
c6in.8xlarge [2]	25000		3125.0		100000	
c6in.12xlarge [2]	37500		4687.5		150000	
c6in.16xlarge [2]	50000		6250.0		200000	
c6in.24xlarge [2]	75000		9375.0		300000	
c6in.32xlarge [2]	100000		12500.0		400000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c6in.metal [2]	100000		12500.0		400000	
c7a.medium [1]	325	10000	40.62	1250.00	2500	40000
c7a.large [1]	650	10000	81.25	1250.00	3600	40000
c7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7a.8xlarge [2]	10000		1250.0		40000	
c7a.12xlarge [2]	15000		1875.0		60000	
c7a.16xlarge [2]	20000		2500.0		80000	
c7a.24xlarge [2]	30000		3750.0		120000	
c7a.32xlarge [2]	40000		5000.0		160000	
c7a.48xlarge [2]	40000		5000.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c7a.metal-48xl [2]	40000		5000.0		240000	
c7g.medium [1]	315	10000	39.38	1250.00	2500	40000
c7g.large [1]	630	10000	78.75	1250.00	3600	40000
c7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7g.8xlarge [2]	10000		1250.0		40000	
c7g.12xlarge [2]	15000		1875.0		60000	
c7g.16xlarge [2]	20000		2500.0		80000	
c7g.metal [2]	20000		2500.0		80000	
c7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c7gd.large [1]	630	10000	78.75	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7gd.8xlarge [2]	10000		1250.0		40000	
c7gd.12xlarge [2]	15000		1875.0		60000	
c7gd.16xlarge [2]	20000		2500.0		80000	
c7gd.metal [2]	20000		2500.0		80000	
c7gn.medium [1]	521	10000	65.12	1250.00	2083	40000
c7gn.large [1]	1042	10000	130.25	1250.00	4167	40000
c7gn.xlarge [1]	2083	10000	260.38	1250.00	8333	40000
c7gn.2xlarge [1]	4167	10000	520.88	1250.00	16667	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c7gn.4xlarge [1]	8333	10000	1041.62	1250.00	33333	40000
c7gn.8xlarge [1]	16667	20000	2083.38	2500.00	66667	80000
c7gn.12xlarge [1]	25000	30000	3125.00	3750.00	100000	120000
c7gn.16xlarge [1]	33333	40000	4166.62	5000.00	133333	160000
c7gn.metal [1]	33333	40000	4166.62	5000.00	133333	160000
c7i.large [1]	650	10000	81.25	1250.00	3600	40000
c7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i.8xlarge [2]	10000		1250.0		40000	
c7i.12xlarge [2]	15000		1875.0		60000	
c7i.16xlarge [2]	20000		2500.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c7i.24xlarge [2]	30000		3750.0		120000	
c7i.48xlarge [2]	40000		5000.0		240000	
c7i.metal-24xl [2]	30000		3750.0		120000	
c7i.metal-48xl [2]	40000		5000.0		240000	
c7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
c7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
c7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
c7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8a.medium [1]	325	10000	40.62	1250.00	2500	40000
c8a.large [1]	650	10000	81.25	1250.00	3600	40000
c8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8a.8xlarge [2]	10000		1250.0		40000	
c8a.12xlarge [2]	15000		1875.0		60000	
c8a.16xlarge [2]	20000		2500.0		80000	
c8a.24xlarge [2]	30000		3750.0		120000	
c8a.48xlarge [2]	60000		7500.0		240000	
c8a.metal-24xl [2]	30000		3750.0		120000	
c8a.metal-48xl [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8g.medium [1]	315	10000	39.38	1250.00	2500	40000
c8g.large [1]	630	10000	78.75	1250.00	3600	40000
c8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8g.8xlarge [2]	10000		1250.0		40000	
c8g.12xlarge [2]	15000		1875.0		60000	
c8g.16xlarge [2]	20000		2500.0		80000	
c8g.24xlarge [2]	30000		3750.0		120000	
c8g.48xlarge [2]	40000		5000.0		240000	
c8g.metal-24xl [2]	30000		3750.0		120000	
c8g.metal-48xl [2]	40000		5000.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
c8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
c8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000
c8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
c8gb.4xlarge [2]	25000		3125.0		120000	
c8gb.8xlarge [2]	50000		6250.0		240000	
c8gb.12xlarge [2]	75000		9375.0		360000	
c8gb.16xlarge [2]	100000		12500.0		480000	
c8gb.24xlarge [2]	150000		18750.0		720000	
c8gb.48xlarge [2]	300000		37500.0		1440000	
c8gb.metal-24xl [2]	150000		18750.0		720000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8gb.metal-48xl [2]	300000		37500.0		1440000	
c8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c8gd.large [1]	630	10000	78.75	1250.00	3600	40000
c8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gd.8xlarge [2]	10000		1250.0		40000	
c8gd.12xlarge [2]	15000		1875.0		60000	
c8gd.16xlarge [2]	20000		2500.0		80000	
c8gd.24xlarge [2]	30000		3750.0		120000	
c8gd.48xlarge [2]	40000		5000.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8gd.metal-24xl [2]	30000		3750.0		120000	
c8gd.metal-48xl [2]	40000		5000.0		240000	
c8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
c8gn.large [1]	1250	10000	156.25	1250.00	5000	40000
c8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
c8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gn.4xlarge [2]	10000		1250.0		40000	
c8gn.8xlarge [2]	20000		2500.0		80000	
c8gn.12xlarge [2]	30000		3750.0		120000	
c8gn.16xlarge [2]	40000		5000.0		160000	
c8gn.24xlarge [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8gn.48xlarge [2]	120000		15000.0		480000	
c8gn.metal-24xl [2]	60000		7500.0		240000	
c8gn.metal-48xl [2]	120000		15000.0		480000	
c8i.large [1]	650	10000	81.25	1250.00	3600	40000
c8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8i.8xlarge [2]	10000		1250.0		40000	
c8i.12xlarge [2]	15000		1875.0		60000	
c8i.16xlarge [2]	20000		2500.0		80000	
c8i.24xlarge [2]	30000		3750.0		120000	
c8i.32xlarge [2]	40000		5000.0		160000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8i.48xlarge [2]	60000		7500.0		240000	
c8i.96xlarge [2]	80000		10000.0		480000	
c8i.metal-48xl [2]	60000		7500.0		240000	
c8i.metal-96xl [2]	80000		10000.0		480000	
c8id.large [1]	650	10000	81.25	1250.00	3600	40000
c8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8id.8xlarge [2]	10000		1250.0		40000	
c8id.12xlarge [2]	15000		1875.0		60000	
c8id.16xlarge [2]	20000		2500.0		80000	
c8id.24xlarge [2]	30000		3750.0		120000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8id.32xlarge [2]	40000		5000.0		160000	
c8id.48xlarge [2]	60000		7500.0		240000	
c8id.96xlarge [2]	80000		10000.0		480000	
c8id.metal-48xl [2]	60000		7500.0		240000	
c8id.metal-96xl [2]	80000		10000.0		480000	
c8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
c8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
c8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
c8in.large [1]	650	10000	81.25	1250.00	3600	40000
c8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8in.8xlarge [2]	10000		1250.0		40000	
c8in.12xlarge [2]	15000		1875.0		60000	
c8in.16xlarge [2]	20000		2500.0		80000	
c8in.24xlarge [2]	30000		3750.0		120000	
c8in.32xlarge [2]	40000		5000.0		160000	
c8in.48xlarge [2]	60000		7500.0		240000	
c8in.96xlarge [2]	120000		15000.0		480000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8in.metal-48xl [2]	60000		7500.0		240000	
c8in.metal-96xl [2]	120000		15000.0		480000	
c8ine.large [1]	650	10000	81.25	1250.00	3600	40000
c8ine.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8ine.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8ine.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8ine.8xlarge [2]	10000		1250.0		40000	
c8ine.12xlarge [2]	15000		1875.0		60000	
c8ib.large [1]	1563	25000	195.38	3125.00	7500	120000
c8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
c8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
c8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c8ib.8xlarge [2]	25000		3125.0		120000	
c8ib.12xlarge [2]	37500		4687.5		180000	
c8ib.16xlarge [2]	50000		6250.0		240000	
c8ib.24xlarge [2]	75000		9375.0		360000	
c8ib.32xlarge [2]	100000		12500.0		480000	
c8ib.48xlarge [2]	150000		18750.0		720000	
c8ib.96xlarge [2]	300000		37500.0		1440000	
c8ib.metal-48xl [2]	150000		18750.0		720000	
c8ib.metal-96xl [2]	300000		37500.0		1440000	
c9g.medium [1]	380	12000	47.50	1500.00	2500	48000
c9g.large [1]	760	12000	95.00	1500.00	3600	48000
c9g.xlarge [1]	1500	12000	187.50	1500.00	6000	48000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c9g.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
c9g.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
c9g.8xlarge [2]	12000		1500.0		48000	
c9g.12xlarge [2]	18000		2250.0		72000	
c9g.16xlarge [2]	24000		3000.0		96000	
c9g.24xlarge [2]	36000		4500.0		144000	
c9g.48xlarge [2]	72000		9000.0		288000	
c9g.metal-48xl [2]	72000		9000.0		288000	
c9gd.medium [1]	380	12000	47.50	1500.00	2500	48000
c9gd.large [1]	760	12000	95.00	1500.00	3600	48000
c9gd.xlarge [1]	1500	12000	187.50	1500.00	6000	48000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
c9gd.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
c9gd.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
c9gd.8xlarge [2]	12000		1500.0		48000	
c9gd.12xlarge [2]	18000		2250.0		72000	
c9gd.16xlarge [2]	24000		3000.0		96000	
c9gd.24xlarge [2]	36000		4500.0		144000	
c9gd.48xlarge [2]	72000		9000.0		288000	
c9gd.metal-48xl [2]	72000		9000.0		288000	

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

記憶體最佳化

> **Note**
>
> - R8a、R8g、R8gd、R8i、R8id、R8i-flex、R9g、R9gd、X8g、X8aedz、X8i 執行個體類型支援可設定的頻寬加權。透過這些執行個體類型，您可針對聯網效能或 Amazon EBS 效能最佳化執行個體的頻寬。下列表格列出這些執行個體類型的預設 Amazon EBS 頻寬效能。如需詳細資訊，請參閱EC2 執行個體頻寬加權組態。
> - 要發揮 U7i 執行個體的最大 IOPS 效能，建議您使用 io2 BlockExpress 磁碟區。

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r4.large [2]	425		53.125		3000	
r4.xlarge [2]	850		106.25		6000	
r4.2xlarge [2]	1700		212.5		12000	
r4.4xlarge [2]	3500		437.5		18750	
r4.8xlarge [2]	7000		875.0		37500	
r4.16xlarge [2]	14000		1750.0		75000	
r5.large [1]	650	4750	81.25	593.75	3600	18750
r5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r5.4xlarge [2]	4750		593.75		18750	
r5.8xlarge [2]	6800		850.0		30000	
r5.12xlarge [2]	9500		1187.5		40000	
r5.16xlarge [2]	13600		1700.0		60000	
r5.24xlarge [2]	19000		2375.0		80000	
r5.metal [2]	19000		2375.0		80000	
r5a.large [1]	650	2880	81.25	360.00	3600	16000
r5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5a.4xlarge [2]	2880		360.0		16000	
r5a.8xlarge [2]	4750		593.75		20000	
r5a.12xlarge [2]	6780		847.5		30000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r5a.16xlarge [2]	9500		1187.5		40000	
r5a.24xlarge [2]	13570		1696.25		60000	
r5ad.large [1]	650	2880	81.25	360.00	3600	16000
r5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5ad.4xlarge [2]	2880		360.0		16000	
r5ad.8xlarge [2]	4750		593.75		20000	
r5ad.12xlarge [2]	6780		847.5		30000	
r5ad.16xlarge [2]	9500		1187.5		40000	
r5ad.24xlarge [2]	13570		1696.25		60000	
r5b.large [1]	1250	10000	156.25	1250.00	5417	43333
r5b.xlarge [1]	2500	10000	312.50	1250.00	10833	43333

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r5b.2xlarge [1]	5000	10000	625.00	1250.00	21667	43333
r5b.4xlarge [2]	10000		1250.0		43333	
r5b.8xlarge [2]	20000		2500.0		86667	
r5b.12xlarge [2]	30000		3750.0		130000	
r5b.16xlarge [2]	40000		5000.0		173333	
r5b.24xlarge [2]	60000		7500.0		260000	
r5b.metal [2]	60000		7500.0		260000	
r5d.large [1]	650	4750	81.25	593.75	3600	18750
r5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5d.4xlarge [2]	4750		593.75		18750	
r5d.8xlarge [2]	6800		850.0		30000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r5d.12xlarge [2]	9500		1187.5		40000	
r5d.16xlarge [2]	13600		1700.0		60000	
r5d.24xlarge [2]	19000		2375.0		80000	
r5d.metal [2]	19000		2375.0		80000	
r5dn.large [1]	650	4750	81.25	593.75	3600	18750
r5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5dn.4xlarge [2]	4750		593.75		18750	
r5dn.8xlarge [2]	6800		850.0		30000	
r5dn.12xlarge [2]	9500		1187.5		40000	
r5dn.16xlarge [2]	13600		1700.0		60000	
r5dn.24xlarge [2]	19000		2375.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r5dn.metal [2]	19000		2375.0		80000	
r5n.large [1]	650	4750	81.25	593.75	3600	18750
r5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge [2]	4750		593.75		18750	
r5n.8xlarge [2]	6800		850.0		30000	
r5n.12xlarge [2]	9500		1187.5		40000	
r5n.16xlarge [2]	13600		1700.0		60000	
r5n.24xlarge [2]	19000		2375.0		80000	
r5n.metal [2]	19000		2375.0		80000	
r6a.large [1]	650	10000	81.25	1250.00	3600	40000
r6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6a.8xlarge [2]	10000		1250.0		40000	
r6a.12xlarge [2]	15000		1875.0		60000	
r6a.16xlarge [2]	20000		2500.0		80000	
r6a.24xlarge [2]	30000		3750.0		120000	
r6a.32xlarge [2]	40000		5000.0		160000	
r6a.48xlarge [2]	40000		5000.0		240000	
r6a.metal [2]	40000		5000.0		240000	
r6g.medium [1]	315	4750	39.38	593.75	2500	20000
r6g.large [1]	630	4750	78.75	593.75	3600	20000
r6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6g.4xlarge [2]	4750		593.75		20000	
r6g.8xlarge [2]	9500		1187.5		40000	
r6g.12xlarge [2]	14250		1781.25		50000	
r6g.16xlarge [2]	19000		2375.0		80000	
r6g.metal [2]	19000		2375.0		80000	
r6gd.medium [1]	315	4750	39.38	593.75	2500	20000
r6gd.large [1]	630	4750	78.75	593.75	3600	20000
r6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6gd.4xlarge [2]	4750		593.75		20000	
r6gd.8xlarge [2]	9500		1187.5		40000	
r6gd.12xlarge [2]	14250		1781.25		50000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6gd.16xlarge [2]	19000		2375.0		80000	
r6gd.metal [2]	19000		2375.0		80000	
r6i.large [1]	650	10000	81.25	1250.00	3600	40000
r6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6i.8xlarge [2]	10000		1250.0		40000	
r6i.12xlarge [2]	15000		1875.0		60000	
r6i.16xlarge [2]	20000		2500.0		80000	
r6i.24xlarge [2]	30000		3750.0		120000	
r6i.32xlarge [2]	40000		5000.0		160000	
r6i.metal [2]	40000		5000.0		160000	
r6id.large [1]	650	10000	81.25	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge [2]	10000		1250.0		40000	
r6id.12xlarge [2]	15000		1875.0		60000	
r6id.16xlarge [2]	20000		2500.0		80000	
r6id.24xlarge [2]	30000		3750.0		120000	
r6id.32xlarge [2]	40000		5000.0		160000	
r6id.metal [2]	40000		5000.0		160000	
r6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
r6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6idn.8xlarge [2]	25000		3125.0		100000	
r6idn.12xlarge [2]	37500		4687.5		150000	
r6idn.16xlarge [2]	50000		6250.0		200000	
r6idn.24xlarge [2]	75000		9375.0		300000	
r6idn.32xlarge [2]	100000		12500.0		400000	
r6idn.metal [2]	100000		12500.0		400000	
r6in.large [1]	1562	25000	195.31	3125.00	6250	100000
r6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6in.8xlarge [2]	25000		3125.0		100000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r6in.12xlarge [2]	37500		4687.5		150000	
r6in.16xlarge [2]	50000		6250.0		200000	
r6in.24xlarge [2]	75000		9375.0		300000	
r6in.32xlarge [2]	100000		12500.0		400000	
r6in.metal [2]	100000		12500.0		400000	
r7a.medium [1]	325	10000	40.62	1250.00	2500	40000
r7a.large [1]	650	10000	81.25	1250.00	3600	40000
r7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7a.8xlarge [2]	10000		1250.0		40000	
r7a.12xlarge [2]	15000		1875.0		60000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r7a.16xlarge [2]	20000		2500.0		80000	
r7a.24xlarge [2]	30000		3750.0		120000	
r7a.32xlarge [2]	40000		5000.0		160000	
r7a.48xlarge [2]	40000		5000.0		240000	
r7a.metal-48xl [2]	40000		5000.0		240000	
r7g.medium [1]	315	10000	39.38	1250.00	2500	40000
r7g.large [1]	630	10000	78.75	1250.00	3600	40000
r7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7g.8xlarge [2]	10000		1250.0		40000	
r7g.12xlarge [2]	15000		1875.0		60000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r7g.16xlarge [2]	20000		2500.0		80000	
r7g.metal [2]	20000		2500.0		80000	
r7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r7gd.large [1]	630	10000	78.75	1250.00	3600	40000
r7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7gd.8xlarge [2]	10000		1250.0		40000	
r7gd.12xlarge [2]	15000		1875.0		60000	
r7gd.16xlarge [2]	20000		2500.0		80000	
r7gd.metal [2]	20000		2500.0		80000	
r7i.large [1]	650	10000	81.25	1250.00	3600	40000
r7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7i.8xlarge [2]	10000		1250.0		40000	
r7i.12xlarge [2]	15000		1875.0		60000	
r7i.16xlarge [2]	20000		2500.0		80000	
r7i.24xlarge [2]	30000		3750.0		120000	
r7i.48xlarge [2]	40000		5000.0		240000	
r7i.metal-24xl [2]	30000		3750.0		120000	
r7i.metal-48xl [2]	40000		5000.0		240000	
r7iz.large [1]	792	10000	99.00	1250.00	3600	40000
r7iz.xlarge [1]	1584	10000	198.00	1250.00	6667	40000
r7iz.2xlarge [1]	3168	10000	396.00	1250.00	13333	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r7iz.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7iz.8xlarge [2]	10000		1250.0		40000	
r7iz.12xlarge [2]	19000		2375.0		76000	
r7iz.16xlarge [2]	20000		2500.0		80000	
r7iz.32xlarge [2]	40000		5000.0		160000	
r7iz.metal-16xl [2]	20000		2500.0		80000	
r7iz.metal-32xl [2]	40000		5000.0		160000	
r8a.medium [1]	325	10000	40.62	1250.00	2500	40000
r8a.large [1]	650	10000	81.25	1250.00	3600	40000
r8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8a.8xlarge [2]	10000		1250.0		40000	
r8a.12xlarge [2]	15000		1875.0		60000	
r8a.16xlarge [2]	20000		2500.0		80000	
r8a.24xlarge [2]	30000		3750.0		120000	
r8a.48xlarge [2]	60000		7500.0		240000	
r8a.metal-24xl [2]	30000		3750.0		120000	
r8a.metal-48xl [2]	60000		7500.0		240000	
r8g.medium [1]	315	10000	39.38	1250.00	2500	40000
r8g.large [1]	630	10000	78.75	1250.00	3600	40000
r8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8g.8xlarge [2]	10000		1250.0		40000	
r8g.12xlarge [2]	15000		1875.0		60000	
r8g.16xlarge [2]	20000		2500.0		80000	
r8g.24xlarge [2]	30000		3750.0		120000	
r8g.48xlarge [2]	40000		5000.0		240000	
r8g.metal-24xl [2]	30000		3750.0		120000	
r8g.metal-48xl [2]	40000		5000.0		240000	
r8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
r8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
r8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8gb.4xlarge [2]	25000		3125.0		120000	
r8gb.8xlarge [2]	50000		6250.0		240000	
r8gb.12xlarge [2]	75000		9375.0		360000	
r8gb.16xlarge [2]	100000		12500.0		480000	
r8gb.24xlarge [2]	150000		18750.0		720000	
r8gb.48xlarge [2]	300000		37500.0		1440000	
r8gb.metal-24xl [2]	150000		18750.0		720000	
r8gb.metal-48xl [2]	300000		37500.0		1440000	
r8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r8gd.large [1]	630	10000	78.75	1250.00	3600	40000
r8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8gd.8xlarge [2]	10000		1250.0		40000	
r8gd.12xlarge [2]	15000		1875.0		60000	
r8gd.16xlarge [2]	20000		2500.0		80000	
r8gd.24xlarge [2]	30000		3750.0		120000	
r8gd.48xlarge [2]	40000		5000.0		240000	
r8gd.metal-24xl [2]	30000		3750.0		120000	
r8gd.metal-48xl [2]	40000		5000.0		240000	
r8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
r8gn.large [1]	1250	10000	156.25	1250.00	5000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
r8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8gn.4xlarge [2]	10000		1250.0		40000	
r8gn.8xlarge [2]	20000		2500.0		80000	
r8gn.12xlarge [2]	30000		3750.0		120000	
r8gn.16xlarge [2]	40000		5000.0		160000	
r8gn.24xlarge [2]	60000		7500.0		240000	
r8gn.48xlarge [2]	120000		15000.0		480000	
r8gn.metal-24xl [2]	60000		7500.0		240000	
r8gn.metal-48xl [2]	120000		15000.0		480000	
r8i.large [1]	650	10000	81.25	1250.00	3600	40000
r8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8i.8xlarge [2]	10000		1250.0		40000	
r8i.12xlarge [2]	15000		1875.0		60000	
r8i.16xlarge [2]	20000		2500.0		80000	
r8i.24xlarge [2]	30000		3750.0		120000	
r8i.32xlarge [2]	40000		5000.0		160000	
r8i.48xlarge [2]	60000		7500.0		240000	
r8i.96xlarge [2]	80000		10000.0		480000	
r8i.metal-48xl [2]	60000		7500.0		240000	
r8i.metal-96xl [2]	80000		10000.0		480000	
r8id.large [1]	650	10000	81.25	1250.00	3600	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8id.8xlarge [2]	10000		1250.0		40000	
r8id.12xlarge [2]	15000		1875.0		60000	
r8id.16xlarge [2]	20000		2500.0		80000	
r8id.24xlarge [2]	30000		3750.0		120000	
r8id.32xlarge [2]	40000		5000.0		160000	
r8id.48xlarge [2]	60000		7500.0		240000	
r8id.96xlarge [2]	80000		10000.0		480000	
r8id.metal-48xl [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8id.metal-96xl [2]	80000		10000.0		480000	
r8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
r8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
r8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
r8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
r8in.large [1]	650	10000	81.25	1250.00	3600	40000
r8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8in.8xlarge [2]	10000		1250.0		40000	
r8in.12xlarge [2]	15000		1875.0		60000	
r8in.16xlarge [2]	20000		2500.0		80000	
r8in.24xlarge [2]	30000		3750.0		120000	
r8in.32xlarge [2]	40000		5000.0		160000	
r8in.48xlarge [2]	60000		7500.0		240000	
r8in.96xlarge [2]	120000		15000.0		480000	
r8in.metal-48xl [2]	60000		7500.0		240000	
r8in.metal-96xl [2]	120000		15000.0		480000	
r8idn.large [1]	650	10000	81.25	1250.00	3600	40000
r8idn.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8idn.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8idn.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8idn.8xlarge [2]	10000		1250.0		40000	
r8idn.12xlarge [2]	15000		1875.0		60000	
r8idn.16xlarge [2]	20000		2500.0		80000	
r8idn.24xlarge [2]	30000		3750.0		120000	
r8idn.32xlarge [2]	40000		5000.0		160000	
r8idn.48xlarge [2]	60000		7500.0		240000	
r8idn.96xlarge [2]	120000		15000.0		480000	
r8idn.metal-48xl [2]	60000		7500.0		240000	
r8idn.metal-96xl [2]	120000		15000.0		480000	
r8ib.large [1]	1563	25000	195.38	3125.00	7500	120000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
r8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
r8ib.8xlarge [2]	25000		3125.0		120000	
r8ib.12xlarge [2]	37500		4687.5		180000	
r8ib.16xlarge [2]	50000		6250.0		240000	
r8ib.24xlarge [2]	75000		9375.0		360000	
r8ib.32xlarge [2]	100000		12500.0		480000	
r8ib.48xlarge [2]	150000		18750.0		720000	
r8ib.96xlarge [2]	300000		37500.0		1440000	
r8ib.metal-48xl [2]	150000		18750.0		720000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8ib.metal-96xl [2]	300000		37500.0		1440000	
r8idb.large [1]	1563	25000	195.38	3125.00	7500	120000
r8idb.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
r8idb.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8idb.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
r8idb.8xlarge [2]	25000		3125.0		120000	
r8idb.12xlarge [2]	37500		4687.5		180000	
r8idb.16xlarge [2]	50000		6250.0		240000	
r8idb.24xlarge [2]	75000		9375.0		360000	
r8idb.32xlarge [2]	100000		12500.0		480000	
r8idb.48xlarge [2]	150000		18750.0		720000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r8idb.96xlarge [2]	300000		37500.0		1440000	
r8idb.metal-48xl [2]	150000		18750.0		720000	
r8idb.metal-96xl [2]	300000		37500.0		1440000	
r9g.medium [1]	380	12000	47.50	1500.00	2500	48000
r9g.large [1]	760	12000	95.00	1500.00	3600	48000
r9g.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
r9g.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
r9g.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
r9g.8xlarge [2]	12000		1500.0		48000	
r9g.12xlarge [2]	18000		2250.0		72000	
r9g.16xlarge [2]	24000		3000.0		96000	
r9g.24xlarge [2]	36000		4500.0		144000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r9g.48xlarge [2]	72000		9000.0		288000	
r9g.metal-48xl [2]	72000		9000.0		288000	
r9gd.medium [1]	380	12000	47.50	1500.00	2500	48000
r9gd.large [1]	760	12000	95.00	1500.00	3600	48000
r9gd.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
r9gd.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
r9gd.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
r9gd.8xlarge [2]	12000		1500.0		48000	
r9gd.12xlarge [2]	18000		2250.0		72000	
r9gd.16xlarge [2]	24000		3000.0		96000	
r9gd.24xlarge [2]	36000		4500.0		144000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
r9gd.48xlarge [2]	72000		9000.0		288000	
r9gd.metal-48xl [2]	72000		9000.0		288000	
u-3tb1.56xlarge [2]	19000		2375.0		80000	
u-6tb1.56xlarge [2]	38000		4750.0		160000	
u-6tb1.112xlarge [2]	38000		4750.0		160000	
u-6tb1.metal [2]	38000		4750.0		160000	
u-9tb1.112xlarge [2]	38000		4750.0		160000	
u-9tb1.metal [2]	38000		4750.0		160000	
u-12tb1.112xlarge [2]	38000		4750.0		160000	
u-12tb1.metal [2]	38000		4750.0		160000	
u-18tb1.112xlarge [2]	38000		4750.0		160000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
u-18tb1.metal [2]	38000		4750.0		160000	
u-24tb1.112xlarge [2]	38000		4750.0		160000	
u-24tb1.metal [2]	38000		4750.0		160000	
u7i-6tb.112xlarge [2]	100000		12500.0		560000	
u7i-8tb.112xlarge [2]	100000		12500.0		560000	
u7i-12tb.224xlarge [2]	100000		12500.0		560000	
u7in-16tb.224xlarge [2]	100000		12500.0		560000	
u7in-24tb.224xlarge [2]	100000		12500.0		560000	
u7in-32tb.224xlarge [2]	100000		12500.0		560000	
u7inh-32tb.480xlarge [2]	160000		20000.0		840000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x1.16xlarge [2]	7000		875.0		40000	
x1.32xlarge [2]	14000		1750.0		80000	
x1e.xlarge [2]	500		62.5		3700	
x1e.2xlarge [2]	1000		125.0		7400	
x1e.4xlarge [2]	1750		218.75		10000	
x1e.8xlarge [2]	3500		437.5		20000	
x1e.16xlarge [2]	7000		875.0		40000	
x1e.32xlarge [2]	14000		1750.0		80000	
x2gd.medium [1]	315	4750	39.38	593.75	2500	20000
x2gd.large [1]	630	4750	78.75	593.75	3600	20000
x2gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x2gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
x2gd.4xlarge [2]	4750		593.75		20000	
x2gd.8xlarge [2]	9500		1187.5		40000	
x2gd.12xlarge [2]	14250		1781.25		60000	
x2gd.16xlarge [2]	19000		2375.0		80000	
x2gd.metal [2]	19000		2375.0		80000	
x2idn.16xlarge [2]	40000		5000.0		173333	
x2idn.24xlarge [2]	60000		7500.0		260000	
x2idn.32xlarge [2]	80000		10000.0		260000	
x2idn.metal [2]	80000		10000.0		260000	
x2iedn.xlarge [1]	2500	20000	312.50	2500.00	8125	65000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x2iedn.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
x2iedn.4xlarge [1]	10000	20000	1250.00	2500.00	32500	65000
x2iedn.8xlarge [2]	20000		2500.0		65000	
x2iedn.16xlarge [2]	40000		5000.0		130000	
x2iedn.24xlarge [2]	60000		7500.0		195000	
x2iedn.32xlarge [2]	80000		10000.0		260000	
x2iedn.metal [2]	80000		10000.0		260000	
x2iezn.2xlarge [2]	3170		396.25		13333	
x2iezn.4xlarge [2]	4750		593.75		20000	
x2iezn.6xlarge [2]	9500		1187.5		40000	
x2iezn.8xlarge [2]	12000		1500.0		55000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x2iezn.12xlarge [2]	19000		2375.0		80000	
x2iezn.metal [2]	19000		2375.0		80000	
x8g.medium [1]	315	10000	39.38	1250.00	2500	40000
x8g.large [1]	630	10000	78.75	1250.00	3600	40000
x8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
x8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8g.8xlarge [2]	10000		1250.0		40000	
x8g.12xlarge [2]	15000		1875.0		60000	
x8g.16xlarge [2]	20000		2500.0		80000	
x8g.24xlarge [2]	30000		3750.0		120000	
x8g.48xlarge [2]	40000		5000.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x8g.metal-24xl [2]	30000		3750.0		120000	
x8g.metal-48xl [2]	40000		5000.0		240000	
x8aedz.large [1]	1250	15000	156.25	1875.00	5000	60000
x8aedz.xlarge [1]	2500	15000	312.50	1875.00	10000	60000
x8aedz.3xlarge [1]	7500	15000	937.50	1875.00	30000	60000
x8aedz.6xlarge [2]	15000		1875.0		60000	
x8aedz.12xlarge [2]	30000		3750.0		120000	
x8aedz.24xlarge [2]	60000		7500.0		240000	
x8aedz.metal-12xl [2]	30000		3750.0		120000	
x8aedz.metal-24xl [2]	60000		7500.0		240000	
x8i.large [1]	650	10000	81.25	1250.00	3600	40000
x8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8i.8xlarge [2]	10000		1250.0		40000	
x8i.12xlarge [2]	15000		1875.0		60000	
x8i.16xlarge [2]	20000		2500.0		80000	
x8i.24xlarge [2]	30000		3750.0		120000	
x8i.32xlarge [2]	40000		5000.0		160000	
x8i.48xlarge [2]	60000		7500.0		240000	
x8i.64xlarge [2]	70000		8750.0		320000	
x8i.96xlarge [2]	80000		10000.0		480000	
x8i.metal-48xl [2]	60000		7500.0		240000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
x8i.metal-96xl [2]	80000		10000.0		480000	
z1d.large [1]	800	3170	100.00	396.25	3333	13333
z1d.xlarge [1]	1580	3170	197.50	396.25	6667	13333
z1d.2xlarge [2]	3170		396.25		13333	
z1d.3xlarge [2]	4750		593.75		20000	
z1d.6xlarge [2]	9500		1187.5		40000	
z1d.12xlarge [2]	19000		2375.0		80000	
z1d.metal [2]	19000		2375.0		80000	

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

儲存最佳化

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
d2.xlarge [2]	750		93.75		6000	
d2.2xlarge [2]	1000		125.0		8000	
d2.4xlarge [2]	2000		250.0		16000	
d2.8xlarge [2]	4000		500.0		32000	
d3.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3.4xlarge [2]	2800		350.0		15000	
d3.8xlarge [2]	5000		625.0		30000	
d3en.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3en.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3en.4xlarge [2]	2800		350.0		15000	
d3en.6xlarge [2]	4000		500.0		25000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
d3en.8xlarge [2]	5000		625.0		30000	
d3en.12xlarge [2]	7000		875.0		40000	
h1.2xlarge [2]	1750		218.75		12000	
h1.4xlarge [2]	3500		437.5		20000	
h1.8xlarge [2]	7000		875.0		40000	
h1.16xlarge [2]	14000		1750.0		80000	
i3.large [2]	425		53.125		3000	
i3.xlarge [2]	850		106.25		6000	
i3.2xlarge [2]	1700		212.5		12000	
i3.4xlarge [2]	3500		437.5		16000	
i3.8xlarge [2]	7000		875.0		32500	
i3.16xlarge [2]	14000		1750.0		65000	
i3.metal [2]	19000		2375.0		80000	
i3en.large [1]	576	4750	72.10	593.75	3000	20000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i3en.xlarge [1]	1153	4750	144.20	593.75	6000	20000
i3en.2xlarge [1]	2307	4750	288.39	593.75	12000	20000
i3en.3xlarge [1]	3800	4750	475.00	593.75	15000	20000
i3en.6xlarge [2]	4750		593.75		20000	
i3en.12xlarge [2]	9500		1187.5		40000	
i3en.24xlarge [2]	19000		2375.0		80000	
i3en.metal [2]	19000		2375.0		80000	
i4g.large [1]	625	10000	78.12	1250.00	2500	40000
i4g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4g.8xlarge [2]	10000		1250.0		40000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i4g.16xlarge [2]	20000		2500.0		80000	
i4i.large [1]	625	10000	78.12	1250.00	2500	40000
i4i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4i.8xlarge [2]	10000		1250.0		40000	
i4i.12xlarge [2]	15000		1875.0		60000	
i4i.16xlarge [2]	20000		2500.0		80000	
i4i.24xlarge [2]	30000		3750.0		120000	
i4i.32xlarge [2]	40000		5000.0		160000	
i4i.metal [2]	40000		5000.0		160000	
i7i.large [1]	625	10000	78.12	1250.00	2500	40000
i7i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i7i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i7i.8xlarge [2]	10000		1250.0		40000	
i7i.12xlarge [2]	15000		1875.0		60000	
i7i.16xlarge [2]	20000		2500.0		80000	
i7i.24xlarge [2]	30000		3750.0		120000	
i7i.48xlarge [2]	60000		7500.0		240000	
i7i.metal-24xl [2]	30000		3750.0		120000	
i7i.metal-48xl [2]	60000		7500.0		240000	
i7ie.large [1]	625	10000	78.12	1250.00	2500	40000
i7ie.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i7ie.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i7ie.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i7ie.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000
i7ie.12xlarge [2]	15000		1875.0		60000	
i7ie.18xlarge [2]	22500		2812.5		90000	
i7ie.24xlarge [2]	30000		3750.0		120000	
i7ie.48xlarge [2]	60000		7500.0		240000	
i7ie.metal-24xl [2]	30000		3750.0		120000	
i7ie.metal-48xl [2]	60000		7500.0		240000	
i8g.large [1]	625	10000	78.12	1250.00	2500	40000
i8g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i8g.8xlarge [2]	10000		1250.0		40000	
i8g.12xlarge [2]	15000		1875.0		60000	
i8g.16xlarge [2]	20000		2500.0		80000	
i8g.24xlarge [2]	30000		3750.0		120000	
i8g.48xlarge [2]	60000		7500.0		240000	
i8g.metal-24xl [2]	30000		3750.0		120000	
i8g.metal-48xl [2]	60000		7500.0		240000	
i8ge.large [1]	625	10000	78.12	1250.00	2500	40000
i8ge.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8ge.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8ge.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i8ge.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
i8ge.12xlarge [2]	15000		1875.0		60000	
i8ge.18xlarge [2]	22500		2812.5		90000	
i8ge.24xlarge [2]	30000		3750.0		120000	
i8ge.48xlarge [2]	60000		7500.0		240000	
i8ge.metal-24xl [2]	30000		3750.0		120000	
i8ge.metal-48xl [2]	60000		7500.0		240000	
im4gn.large [1]	1250	10000	156.25	1250.00	5000	40000
im4gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
im4gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
im4gn.4xlarge [2]	10000		1250.0		40000	
im4gn.8xlarge [2]	20000		2500.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
im4gn.16xlarge [2]	40000		5000.0		160000	
is4gen.medium [1]	625	10000	78.12	1250.00	2500	40000
is4gen.large [1]	1250	10000	156.25	1250.00	5000	40000
is4gen.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
is4gen.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
is4gen.4xlarge [2]	10000		1250.0		40000	
is4gen.8xlarge [2]	20000		2500.0		80000	

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

加速運算

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
dl1.24xlarge [2]	19000		2375.0		80000	
dl2q.24xlarge [2]	19000		2375.0		80000	
f2.6xlarge [2]	7500		937.5		30000	
f2.12xlarge [2]	15000		1875.0		60000	
f2.48xlarge [2]	60000		7500.0		240000	
g3.4xlarge [2]	3500		437.5		20000	
g3.8xlarge [2]	7000		875.0		40000	
g3.16xlarge [2]	14000		1750.0		80000	
g4ad.xlarge [1]	400	3170	50.00	396.25	1700	13333
g4ad.2xlarge [1]	800	3170	100.00	396.25	3400	13333
g4ad.4xlarge [1]	1580	3170	197.50	396.25	6700	13333

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
g4ad.8xlarge [2]	3170		396.25		13333	
g4ad.16xlarge [2]	6300		787.5		26667	
g4dn.xlarge [1]	950	3500	118.75	437.50	3000	20000
g4dn.2xlarge [1]	1150	3500	143.75	437.50	6000	20000
g4dn.4xlarge [2]	4750		593.75		20000	
g4dn.8xlarge [2]	9500		1187.5		40000	
g4dn.12xlarge [2]	9500		1187.5		40000	
g4dn.16xlarge [2]	9500		1187.5		40000	
g4dn.metal [2]	19000		2375.0		80000	
g5.xlarge [1]	700	3500	87.50	437.50	3000	15000
g5.2xlarge [1]	850	3500	106.25	437.50	3500	15000
g5.4xlarge [2]	4750		593.75		20000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
g5.8xlarge [2]	16000		2000.0		65000	
g5.12xlarge [2]	16000		2000.0		65000	
g5.16xlarge [2]	16000		2000.0		65000	
g5.24xlarge [2]	19000		2375.0		80000	
g5.48xlarge [2]	19000		2375.0		80000	
g5g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
g5g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
g5g.4xlarge [2]	4750		593.75		20000	
g5g.8xlarge [2]	9500		1187.5		40000	
g5g.16xlarge [2]	19000		2375.0		80000	
g5g.metal [2]	19000		2375.0		80000	
g6.xlarge [1]	1000	5000	125.00	625.00	4000	20000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
g6.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6.4xlarge [2]	8000		1000.0		32000	
g6.8xlarge [2]	16000		2000.0		64000	
g6.12xlarge [2]	20000		2500.0		80000	
g6.16xlarge [2]	20000		2500.0		80000	
g6.24xlarge [2]	30000		3750.0		120000	
g6.48xlarge [2]	60000		7500.0		240000	
g6e.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6e.4xlarge [2]	8000		1000.0		32000	
g6e.8xlarge [2]	16000		2000.0		64000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
g6e.12xlarge [2]	20000		2500.0		80000	
g6e.16xlarge [2]	20000		2500.0		80000	
g6e.24xlarge [2]	30000		3750.0		120000	
g6e.48xlarge [2]	60000		7500.0		240000	
g6f.large [1]	936	5000	117.00	625.00	3750	20000
g6f.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6f.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6f.4xlarge [2]	6000		750.0		24000	
gr6.4xlarge [2]	8000		1000.0		32000	
gr6.8xlarge [2]	16000		2000.0		64000	
gr6f.4xlarge [2]	8000		1000.0		32000	
g7.2xlarge [1]	4000	8000	500.00	1000.00	16000	32000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
g7.4xlarge [2]	8000		1000.0		32000	
g7.8xlarge [2]	16000		2000.0		64000	
g7.12xlarge [2]	20000		2500.0		80000	
g7.24xlarge [2]	40000		5000.0		160000	
g7.48xlarge [2]	80000		10000.0		320000	
g7e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g7e.4xlarge [2]	8000		1000.0		32000	
g7e.8xlarge [2]	16000		2000.0		64000	
g7e.12xlarge [2]	25000		3125.0		100000	
g7e.24xlarge [2]	50000		6250.0		200000	
g7e.48xlarge [2]	100000		12500.0		400000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
inf1.xlarge [1]	1190	4750	148.75	593.75	4000	20000
inf1.2xlarge [1]	1190	4750	148.75	593.75	6000	20000
inf1.6xlarge [2]	4750		593.75		20000	
inf1.24xlarge [2]	19000		2375.0		80000	
inf2.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
inf2.8xlarge [2]	10000		1250.0		40000	
inf2.24xlarge [2]	30000		3750.0		120000	
inf2.48xlarge [2]	60000		7500.0		240000	
p3.2xlarge [2]	1750		218.75		10000	
p3.8xlarge [2]	7000		875.0		40000	
p3.16xlarge [2]	14000		1750.0		80000	

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
p3dn.24xlarge [2]	19000		2375.0		80000	
p4d.24xlarge [2]	19000		2375.0		80000	
p4de.24xlarge [2]	19000		2375.0		80000	
p5.4xlarge [2]	10000		1250.0		32500	
p5.48xlarge [2]	80000		10000.0		260000	
p5e.48xlarge [2]	80000		10000.0		260000	
p5en.48xlarge [2]	100000		12500.0		400000	
p6-b200.48xlarge [2]	100000		12500.0		400000	
p6-b300.48xlarge [2]	100000		12500.0		400000	
p6e-gb200.36xlarge [2]	60000		7500.0		240000	
trn1.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
trn1.32xlarge [2]	80000		10000.0		260000	
trn1n.32xlarge [2]	80000		10000.0		260000	
trn2.3xlarge [2]	5000		625.0		16250	
trn2.48xlarge [2]	80000		10000.0		260000	
trn2u.48xlarge [2]	80000		10000.0		260000	
vt1.3xlarge [1]	2375	4750	296.88	593.75	10000	20000
vt1.6xlarge [2]	4750		593.75		20000	
vt1.24xlarge [2]	19000		2375.0		80000	

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

高效能運算

執行個體大小	基準頻寬 (Mbps)	最高頻寬 (Mbps)	基準輸送量 (MB/s, 128 KB I/O)	最大傳輸量 (MB/s、128 KiB I/O)	基準 IOPS (16 KiB I/O)	最大 IOPS (16 KiB I/O)
hpc6a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc6id.32xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.12xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.24xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.96xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.4xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.8xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.16xlarge [1]	87	2085	10.88	260.62	500	11000
hpc8a.96xlarge [1]	87	2085	10.88	260.62	500	11000

[1] 這些執行個體至少每 24 小時一次維持最佳效能 30 分鐘，然後就會回復至其基準效能。

[2] 這些執行個體可以無限期維持其上述效能。如果您的工作負載需要維持超過 30 分鐘的最佳效能，請選取其中一個執行個體。

支援 EBS 最佳化

以下執行個體類型支援 EBS 最佳化，但預設不啟用 EBS 最佳化。您必須在啟動期間或之後啟用 EBS 最佳化 (需支付額外小時費用)，以達到所述的 EBS 效能水準。

執行個體大小	最高頻寬 (Mbps)	最大傳輸量 (MB/s、128 KiB I/O)	最大 IOPS (16 KiB I/O)
c1.xlarge	1000	125.0	8000
c3.xlarge	500	62.5	4000
c3.2xlarge	1000	125.0	8000
c3.4xlarge	2000	250.0	16000
i2.xlarge	500	62.5	4000
i2.2xlarge	1000	125.0	8000
i2.4xlarge	2000	250.0	16000
m1.large	500	62.5	4000
m1.xlarge	1000	125.0	8000
m2.2xlarge	500	62.5	4000
m2.4xlarge	1000	125.0	8000
m3.xlarge	500	62.5	4000
m3.2xlarge	1000	125.0	8000
r3.xlarge	500	62.5	4000
r3.2xlarge	1000	125.0	8000
r3.4xlarge	2000	250.0	16000

> **Note**
>
> i2.8xlarge、c3.8xlarge 和 r3.8xlarge 執行個體不具備專屬 EBS 頻寬，因此不提供 EBS 最佳化。在這些執行個體上，網路流量和 Amazon EBS 流量將共用同一個 10 GB 網路介面。

獲得最大的 Amazon EBS 最佳化效能

執行個體的 EBS 效能受限於執行個體類型效能上限，或其連接磁碟區的彙總效能 (以較小者為準)。若要達到最大 EBS 效能，執行個體必須具有連接磁碟區，這些磁碟區提供的組合效能必須等於或大於最大執行個體效能。例如，r6i.16xlarge 若要達到 80,000 IOPS，每個執行個體至少須佈建 5 gp2 個 16,000 IOPS 的磁碟區 (5 個磁碟區 x 16,000 IOPS = 80,000 IOPS)，或者也可配置 1 gp3 個磁碟區並提供 80,000 IOPS。我們建議您選擇提供的專屬 Amazon EBS 輸送量高於應用程式所需的執行個體類型；否則，Amazon EBS 和 Amazon EC2 之間的連線將成為效能瓶頸。

> **Important**
>
> 使用可設定的頻寬加權時，您執行個體的 EBS 頻寬限制可能會變更。對於具有增加聯網頻寬的 VPC-1 加權組態的執行個體，由於在達到 IOPS 限制前先觸及 EBS 頻寬限制，您可能會遇到 EBS 磁碟區的 IOPS 低於預期的情況。這在較大的 I/O 大小中尤其明顯。請務必測試您的特定工作負載，確保在使用選定的頻寬加權時符合效能需求。如需詳細資訊，請參閱EC2 執行個體頻寬加權組態。

您可以使用 EBSIOBalance% 和 EBSByteBalance% 指標來協助您判斷執行個體的大小是否正確。您可以在 CloudWatch 主控台檢視這些指標，並依據您指定的閾值設定會觸發的警示。這些指標以百分比表示。維持低平衡百分比的執行個體視為有待擴展規模。平衡百分比從未低於 100% 的執行個體視為有待縮減規模。如需詳細資訊，請參閱 使用 CloudWatch 監控您的執行個體。

此記憶體密集執行個體專為在雲端執行大型記憶體內資料庫而設計，包括 SAP HANA 記憶體內資料庫的生產部署。若要將 EBS 效能最大化，請使用具有偶數數量 io1 或 io2 磁碟區且有完全相同佈建效能的高記憶體執行個體。例如，對於 IOPS 繁重的工作負載，使用具有 40,000 個佈建 IOPS 的四個 io1 或 io2 磁碟區，以獲得最大 160,000 個執行個體 IOPS。同樣地，對於輸送量繁重的工作負載，使用具有具有 48,000 佈建 IOPS 的六個 io1 或 io2 磁碟區，以獲得最大 4,750 MB/s 輸送量。如需其他建議，請參閱 SAP HANA 的儲存組態。

考量事項

- 2020 年 2 月 26 日之後推出的 G4dn、I3en、Inf1、M5a、M5ad、R5a、R5ad、T3、T3a 和 Z1d 執行個體可提供最大的 EBS 最佳化效能。若要從 2020 年 2 月 26 日之前啟動的執行個體獲得最大效能，請停止並啟動該執行個體。
- 2019 年 12 月 3 日之後推出的 C5、C5d、C5n、M5、M5d、M5n、M5dn、R5、R5d、R5n、R5dn 和 P3dn 執行個體可提供最大的 EBS 最佳化效能。若要從 2019 年 12 月 3 日之前啟動的執行個體獲得最大效能，請停止並啟動該執行個體。
- `u-6tb1.metal`、`u-9tb1.metal`、`u-12tb1.metal` 以及 2020 年 3 月 12 日之後推出的執行個體可提供最大的 EBS 最佳化效能。2020 年 3 月 12 日之前推出的這些類型執行個體可能會提供較低的效能。若要從 2020 年 3 月 12 日之前啟動的執行個體獲得最大效能，請聯絡您的客戶團隊以免費升級該執行個體。

尋找 EBS 最佳化 EC2 執行個體類型

您可以檢視每個區域中支援 EBS 最佳化的執行個體類型。

Console

尋找預設為 EBS 最佳化的執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instance Types (執行個體類型)。
3. 新增篩選條件 EBS 最佳化支援 = 預設。
4. （選用）選擇偏好設定圖示，然後開啟相關欄，例如 EBS 最大 IOPS 和 EBS 基準 IOPS。
5. (選用) 新增篩選條件以進一步限定到感興趣的特定執行個體類型。

AWS CLI

尋找預設為 EBS 最佳化的執行個體類型

使用下列 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
--filters Name=ebs-info.ebs-optimized-support,Values=default  \
--query 'InstanceTypes[].{InstanceType:InstanceType,
 "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,
```

```
 MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, "MaxThroughput(MB/
s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
--output=table
```

尋找可選支援 EBS 最佳化的執行個體類型

使用下列 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
--filters Name=ebs-info.ebs-optimized-support,Values=supported \
--query 'InstanceTypes[].{InstanceType:InstanceType,
 "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,
 MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, "MaxThroughput(MB/
s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
--output=table
```

下列為 eu-west-1 的範例輸出。

```
-------------------------------------------------------------------------
|                          DescribeInstanceTypes                        |
+--------------+----------------------+----------+-----------------------+
| InstanceType | MaxBandwidth(Mb/s)   | MaxIOPS  |  MaxThroughput(MB/s)  |
+--------------+----------------------+----------+-----------------------+
|  i2.2xlarge  |  1000                |  8000    |  125.0                |
|  m2.4xlarge  |  1000                |  8000    |  125.0                |
|  m2.2xlarge  |  500                 |  4000    |  62.5                 |
|  c1.xlarge   |  1000                |  8000    |  125.0                |
|  i2.xlarge   |  500                 |  4000    |  62.5                 |
|  m3.xlarge   |  500                 |  4000    |  62.5                 |
|  m1.xlarge   |  1000                |  8000    |  125.0                |
|  r3.4xlarge  |  2000                |  16000   |  250.0                |
|  r3.2xlarge  |  1000                |  8000    |  125.0                |
|  c3.xlarge   |  500                 |  4000    |  62.5                 |
|  m3.2xlarge  |  1000                |  8000    |  125.0                |
|  r3.xlarge   |  500                 |  4000    |  62.5                 |
|  i2.4xlarge  |  2000                |  16000   |  250.0                |
|  c3.4xlarge  |  2000                |  16000   |  250.0                |
|  c3.2xlarge  |  1000                |  8000    |  125.0                |
|  m1.large    |  500                 |  4000    |  62.5                 |
+--------------+----------------------+----------+-----------------------+
```

PowerShell

尋找預設為 EBS 最佳化的執行個體類型

使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="default"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

尋找可選支援 EBS 最佳化的執行個體類型

使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="supported"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

下列為 `eu-west-1` 的範例輸出。

```
InstanceType MaxBandwidth(Mb/s) MaxIOPS MaxThroughput (MB/s)
------------ ------------------ ------- --------------------
m2.4xlarge                 1000    8000              125.000
i2.2xlarge                 1000    8000              125.000
c1.xlarge                  1000    8000              125.000
m2.2xlarge                  500    4000               62.500
r3.2xlarge                 1000    8000              125.000
m3.xlarge                   500    4000               62.500
r3.4xlarge                 2000   16000              250.000
m1.xlarge                  1000    8000              125.000
i2.xlarge                   500    4000               62.500
c3.xlarge                   500    4000               62.500
```

```
c3.4xlarge                2000   16000              250.000
c3.2xlarge                1000    8000              125.000
i2.4xlarge                2000   16000              250.000
r3.xlarge                  500    4000               62.500
m3.2xlarge                1000    8000              125.000
m1.large                   500    4000               62.500
```

針對 Amazon EC2 執行個體啟用 EBS 最佳化

您僅能為上一代可選擇支援 EBS 最佳化的執行個體類型手動啟用該功能。如果您為這些執行個體類型啟用 EBS 最佳化，需支付額外的每小時費用

先決條件

- 確認該執行個體類型是否需要您啟用 EBS 最佳化。如需詳細資訊，請參閱支援 EBS 最佳化。
- 若要在啟動後啟用 EBS 最佳化，您必須停止執行個體。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

Console

若要在啟動期間啟用 Amazon EBS 最佳化

在啟動執行個體精靈中，選取所需的執行個體類型。展開進階詳細資訊區段，然後對於 EBS 最佳化執行個體，選取啟用。

如果所選執行個體類型不支援 Amazon EBS 最佳化，則會停用下拉式選單。如果執行個體類型預設為 Amazon EBS 最佳化，則已選擇啟用。

若要在啟動後啟用 Amazon EBS 最佳化

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取該執行個體。
3. 停止執行個體。依序選擇 Actions (動作)、Instance state (執行個體狀態)、Stop instance (停止執行個體)。

4. 在仍然選取執行個體的情況下，選取 Actions (動作)、Instance settings (執行個體設定)、Change instance type (變更執行個體類型)。
5. 選取 EBS最佳化，然後選擇套用。

   如果執行個體類型預設為 Amazon EBS 最佳化，或不支援 Amazon EBS 最佳化，則會停用核取方塊。
6. 重新啟動執行個體。選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。

AWS CLI

若要在啟動期間啟用 Amazon EBS 最佳化

請使用 `--ebs-optimized` 選項並搭配 run-instances 命令。

若要在啟動後啟用 Amazon EBS 最佳化

1. 如果執行個體正在執行中，使用 stop-instance 命令來停止它。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

2. 使用 modify-instance-attribute 命令並搭配 `--ebs-optimized` 選項來啟用 EBS 最佳化。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --ebs-optimized
```

PowerShell

若要在啟動期間啟用 Amazon EBS 最佳化

使用 New-EC2Instance cmdlet 搭配 `-EbsOptimized` 選項。

若要在啟動後啟用 Amazon EBS 最佳化

1. 如果執行個體正在執行中，請使用 Stop-EC2Instance cmdlet 來停止它。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

2. 使用 Edit-EC2InstanceAttribute cmdlet 搭配 `-EbsOptimized` 選項來啟用 EBS 最佳化。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -EbsOptimized $true
```

Amazon EC2 執行個體的 CPU 選項

許多 Amazon EC2 執行個體支援同時多執行緒 (SMT)，此種技術可在單一 CPU 核心上同時執行多個執行緒。每個執行緒皆代表執行個體上的一個虛擬 CPU (vCPU)。執行個體有預設數量的 CPU 核心，該數量因執行個體類型而異。例如，m5.xlarge 執行個體類型有兩個 CPU 核心，每個核心有兩個執行緒—因此預設共有四個 vCPU。

多數情況下都會有 Amazon EC2 執行個體類型，其有符合您工作負載的記憶體和 vCPU 數量組合。但是，您還是可以在執行個體啟動期間和之後指定下列 CPU 選項，以針對特定的工作負載或商業需求將您的執行個體最佳化：

- Number of CPU cores (CPU 核心數量)：您可自訂執行個體的 CPU 核心數量。這種做法可針對有足夠 RAM 數量可執行記憶體密集型工作負載但 CPU 核心較少的執行個體，將軟體的授權成本最佳化。
- 每核心的執行緒數量：您可以為每個 CPU 核心指定單一執行緒，來停用多執行緒。這種做法適用於特定的工作負載，例如高效能運算 (HPC) 的工作負載。

考量事項

- 您無法修改 T2、C7a、M7a、R7a 和 Apple 矽 Mac 執行個體的每個核心執行緒數目，以及以 AWS Graviton 處理器為基礎的執行個體。
- 您可執行的執行個體數量視所使用執行個體類型的預設 vCPU 而定。變更執行個體的 CPU 選項並不會影響我們計算該執行個體耗用 vCPU 的方式。

定價

指定 CPU 選項不會增加費用。對於從含許可的 Windows 及 SQL Server AMI 啟動的 EC2 執行個體，您可自訂 CPU 選項，以運用 EC2「最佳化 CPU」功能，依執行個體設定的 vCPU 數量支付許可費。對於其他 EC2 執行個體，您要支付的費用就跟使用預設 CPU 選項啟動的執行個體一樣。

目錄

- 指定 Amazon EC2 執行個體 CPU 選項的規則

- 支援 Amazon EC2 執行個體類型的 CPU 選項
- 指定 Amazon EC2 執行個體的 CPU 選項
- 檢視 Amazon EC2 執行個體的 CPU 執行緒與核心
- 針對含許可的執行個體最佳化 CPU

指定 Amazon EC2 執行個體 CPU 選項的規則

若要為執行個體指定 CPU 選項，請注意下列規則：

- 您無法指定裸機執行個體的 CPU 選項。
- 您可以在執行個體啟動期間和之後指定 CPU 選項。
- 若要設定 CPU 選項時，您必須在請求中同時指定 CPU 核心數量和每個核心的執行緒數量。如需請求範例，請參閱指定 Amazon EC2 執行個體的 CPU 選項。
- 執行個體的 vCPU 數目是 CPU 核心數量乘以每核心的執行緒數量。若要指定自訂數量的 vCPU，您必須指定對該執行個體類型有效的 CPU 核心和每核心執行緒數量。不得超過執行個體預設的 vCPU 數量。如需詳細資訊，請參閱支援 Amazon EC2 執行個體類型的 CPU 選項。
- 若要停用也稱為超執行緒的同時多執行緒 (SMT)，請為每個核心指定一個執行緒。
- 在主控台中，當您變更現有執行個體的執行個體類型時，Amazon EC2 會盡可能將現有執行個體的 CPU 選項設定套用至新執行個體。若新的執行個體類型不支援這些設定，則 CPU 選項會重設為使用預設 CPU 選項。此選項使用預設 vCPU 數量的新執行個體類型。

  若要更新新執行個體的設定，請在變更執行個體類型檢視的進階詳細資訊下方，選取指定 CPU 選項。
- 您指定的 CPU 選項在執行個體停止、啟動或重新啟動後仍會保留。
- 若您使用預留執行個體，則在同一付款帳戶中，為從含許可的 Windows AMI 啟動的執行個體設定「最佳化 CPU」時，可能無法套用折扣。我們建議您使用 Savings Plans 來降低基於 vCPU 的授權成本，並為運算成本提供相當的節省效果。
- 若要節省從 Windows 及 SQL Server 含許可的 AMI 所啟動執行個體的許可成本，您必須設定至少四個 vCPU。若您設定的 vCPU 少於四個，將適用預設帳單方式。
- T3 執行個體類型不支援為含許可的執行個體最佳化 CPU。
- 專用執行個體不支援針對包含授權的執行個體最佳化 CPUs。

支援 Amazon EC2 執行個體類型的 CPU 選項

下表列出支援指定 CPU 選項的執行個體類型。

目錄

- 一般用途執行個體
- 運算最佳化執行個體
- 記憶體最佳化執行個體
- 儲存最佳化執行個體
- 加速運算執行個體
- 高效能運算執行個體

一般用途執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m2.xlarge	2	2	1	1、2	1
m2.2xlarge	4	4	1	1、2、3、4	1
m2.4xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m3.large	2	1	2	1	1、2
m3.xlarge	4	2	2	1、2	1、2
m3.2xlarge	8	4	2	1、2、3、4	1、2
m4.large	2	1	2	1	1、2
m4.xlarge	4	2	2	1、2	1、2
m4.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m4.10xlarge	40	20	2	2、4、6、8、10、12、14、16、18、20	1、2
m4.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5.large	2	1	2	1	1、2
m5.xlarge	4	2	2	2	1、2
m5.2xlarge	8	4	2	2、4	1、2
m5.4xlarge	16	8	2	2、4、6、8	1、2
m5.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m5a.large	2	1	2	1	1、2
m5a.xlarge	4	2	2	2	1、2
m5a.2xlarge	8	4	2	2、4	1、2
m5a.4xlarge	16	8	2	2、4、6、8	1、2
m5a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
m5a.12xlarge	48	24	2	6、12、18、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5a.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5a.24xlarge	96	48	2	12、18、24、36、48	1、2
m5ad.large	2	1	2	1	1、2
m5ad.xlarge	4	2	2	2	1、2
m5ad.2xlarge	8	4	2	2、4	1、2
m5ad.4xlarge	16	8	2	2、4、6、8	1、2
m5ad.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
m5ad.12xlarge	48	24	2	6、12、18、24	1、2
m5ad.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5ad.24xlarge	96	48	2	12、18、24、36、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5d.large	2	1	2	1	1、2
m5d.xlarge	4	2	2	2	1、2
m5d.2xlarge	8	4	2	2、4	1、2
m5d.4xlarge	16	8	2	2、4、6、8	1、2
m5d.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5d.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5dn.large	2	1	2	1	1、2
m5dn.xlarge	4	2	2	1、2	1、2
m5dn.2xlarge	8	4	2	2、4	1、2
m5dn.4xlarge	16	8	2	2、4、6、8	1、2
m5dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5dn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5n.large	2	1	2	1	1、2
m5n.xlarge	4	2	2	1、2	1、2
m5n.2xlarge	8	4	2	2、4	1、2
m5n.4xlarge	16	8	2	2、4、6、8	1、2
m5n.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5n.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5n.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5n.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m5zn.large	2	1	2	1	1、2
m5zn.xlarge	4	2	2	1、2	1、2
m5zn.2xlarge	8	4	2	2、4	1、2
m5zn.3xlarge	12	6	2	2、4、6	1、2
m5zn.6xlarge	24	12	2	2、4、6、8、10、12	1、2
m5zn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m6a.large	2	1	2	1	1、2
m6a.xlarge	4	2	2	1、2	1、2
m6a.2xlarge	8	4	2	1、2、3、4	1、2
m6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
m6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
m6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
m6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
m6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
m6g.large	2	2	1	1、2	1
m6g.xlarge	4	4	1	1、2、3、4	1
m6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m6gd.large	2	2	1	1、2	1
m6gd.xlarge	4	4	1	1、2、3、4	1
m6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m6i.large	2	1	2	1	1、2
m6i.xlarge	4	2	2	1、2	1、2
m6i.2xlarge	8	4	2	2、4	1、2
m6i.4xlarge	16	8	2	2、4、6、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6id.large	2	1	2	1	1、2
m6id.xlarge	4	2	2	1、2	1、2
m6id.2xlarge	8	4	2	2、4	1、2
m6id.4xlarge	16	8	2	2、4、6、8	1、2
m6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6idn.large	2	1	2	1	1、2
m6idn.xlarge	4	2	2	1、2	1、2
m6idn.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m6idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m6idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m6idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m6idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6in.large	2	1	2	1	1、2
m6in.xlarge	4	2	2	1、2	1、2
m6in.2xlarge	8	4	2	1、2、3、4	1、2
m6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m7a.large	2	2	1	1、2	1
m7a.xlarge	4	4	1	1、2、3、4	1
m7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
m7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
m7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
m7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
m7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
m7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
m7g.large	2	2	1	1、2	1
m7g.xlarge	4	4	1	1、2、3、4	1
m7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m7gd.large	2	2	1	1、2	1
m7gd.xlarge	4	4	1	1、2、3、4	1
m7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m7i.large	2	1	2	1	1、2
m7i.xlarge	4	2	2	1、2	1、2
m7i.2xlarge	8	4	2	1、2、3、4	1、2
m7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
m7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7i-flex.large	2	1	2	1	1、2
m7i-flex.xlarge	4	2	2	1、2	1、2
m7i-flex.2xlarge	8	4	2	1、2、3、4	1、2
m7i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m7i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m7i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m7i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8a.medium	1	1	1	1	1
m8a.large	2	2	1	1、2	1
m8a.xlarge	4	4	1	1、2、3、4	1
m8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
m8a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
m8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
m8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
m8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
m8azn.medium	1	1	1	1	1
m8azn.large	2	2	1	1、2	1
m8azn.xlarge	4	4	1	1、2、3、4	1
m8azn.3xlarge	12	12	1	1、2、4、6、8、10、12	1
m8azn.6xlarge	24	24	1	1, 2, 3, 4, 8, 12, 16, 20, 24	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8azn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、16、24、32、40、48	1
m8azn.24xlarge	96	96	1	4、6、8、10、12、14、16、32、48、64、80、96	1
m8g.large	2	2	1	1、2	1
m8g.xlarge	4	4	1	1、2、3、4	1
m8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
m8gb.large	2	2	1	1、2	1
m8gb.xlarge	4	4	1	1、2、3、4	1
m8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
m8gd.large	2	2	1	1、2	1
m8gd.xlarge	4	4	1	1、2、3、4	1
m8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
m8gn.large	2	2	1	1、2	1
m8gn.xlarge	4	4	1	1、2、3、4	1
m8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
m8i.large	2	1	2	1	1、2
m8i.xlarge	4	2	2	1、2	1、2
m8i.2xlarge	8	4	2	1、2、3、4	1、2
m8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m8id.large	2	1	2	1	1、2
m8id.xlarge	4	2	2	1、2	1、2
m8id.2xlarge	8	4	2	1、2、3、4	1、2
m8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8i-flex.large	2	1	2	1	1、2
m8i-flex.xlarge	4	2	2	1、2	1、2
m8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
m8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8in.large	2	1	2	1	1、2
m8in.xlarge	4	2	2	1、2	1、2
m8in.2xlarge	8	4	2	1、2、3、4	1、2
m8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idn.large	2	1	2	1	1、2
m8idn.xlarge	4	2	2	1、2	1、2
m8idn.2xlarge	8	4	2	1、2、3、4	1、2
m8idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idn.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8idn.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m8ine.large	2	1	2	1	1、2
m8ine.xlarge	4	2	2	1、2	1、2
m8ine.2xlarge	8	4	2	1、2、3、4	1、2
m8ine.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8ine.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8ine.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8ib.large	2	1	2	1	1、2
m8ib.xlarge	4	2	2	1、2	1、2
m8ib.2xlarge	8	4	2	1、2、3、4	1、2
m8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idb.large	2	1	2	1	1、2
m8idb.xlarge	4	2	2	1、2	1、2
m8idb.2xlarge	8	4	2	1、2、3、4	1、2
m8idb.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8idb.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8idb.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idb.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8idb.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8idb.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m8idb.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8idb.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m9g.large	2	2	1	1、2	1
m9g.xlarge	4	4	1	1、2、3、4	1
m9g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m9g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m9g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9g.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
m9gd.large	2	2	1	1、2	1
m9gd.xlarge	4	4	1	1、2、3、4	1
m9gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m9gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m9gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1
m9gd.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
m9gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
t3.nano	2	1	2	1	1、2
t3.micro	2	1	2	1	1、2
t3.small	2	1	2	1	1、2
t3.medium	2	1	2	1	1、2
t3.large	2	1	2	1	1、2
t3.xlarge	4	2	2	2	1、2
t3.2xlarge	8	4	2	2、4	1、2
t3a.nano	2	1	2	1	1、2
t3a.micro	2	1	2	1	1、2
t3a.small	2	1	2	1	1、2
t3a.medium	2	1	2	1	1、2
t3a.large	2	1	2	1	1、2
t3a.xlarge	4	2	2	2	1、2
t3a.2xlarge	8	4	2	2、4	1、2
t4g.nano	2	2	1	1、2	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
t4g.micro	2	2	1	1、2	1
t4g.small	2	2	1	1、2	1
t4g.medium	2	2	1	1、2	1
t4g.large	2	2	1	1、2	1
t4g.xlarge	4	4	1	1、2、3、4	1
t4g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

運算最佳化執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c3.large	2	1	2	1	1、2
c3.xlarge	4	2	2	1、2	1、2
c3.2xlarge	8	4	2	1、2、3、4	1、2
c3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c4.large	2	1	2	1	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c4.xlarge	4	2	2	1、2	1、2
c4.2xlarge	8	4	2	1、2、3、4	1、2
c4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c4.8xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5.large	2	1	2	1	1、2
c5.xlarge	4	2	2	2	1、2
c5.2xlarge	8	4	2	2、4	1、2
c5.4xlarge	16	8	2	2、4、6、8	1、2
c5.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c5.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c5.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c5a.large	2	1	2	1	1、2
c5a.xlarge	4	2	2	1、2	1、2
c5a.2xlarge	8	4	2	1、2、3、4	1、2
c5a.4xlarge	16	8	2	1, 2, 3, 4, 8	1、2
c5a.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1、2
c5a.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c5a.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1、2
c5a.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1、2
c5ad.large	2	1	2	1	1、2
c5ad.xlarge	4	2	2	1、2	1、2
c5ad.2xlarge	8	4	2	1、2、3、4	1、2
c5ad.4xlarge	16	8	2	1, 2, 3, 4, 8	1、2
c5ad.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1、2
c5ad.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1、2
c5ad.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1、2
c5ad.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c5d.large	2	1	2	1	1、2
c5d.xlarge	4	2	2	2	1、2
c5d.2xlarge	8	4	2	2、4	1、2
c5d.4xlarge	16	8	2	2、4、6、8	1、2
c5d.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
c5d.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c5d.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c5n.large	2	1	2	1	1、2
c5n.xlarge	4	2	2	2	1、2
c5n.2xlarge	8	4	2	2、4	1、2
c5n.4xlarge	16	8	2	2、4、6、8	1、2
c5n.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5n.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c6a.large	2	1	2	1	1、2
c6a.xlarge	4	2	2	1、2	1、2
c6a.2xlarge	8	4	2	1、2、3、4	1、2
c6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
c6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
c6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
c6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
c6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
c6g.large	2	2	1	1、2	1
c6g.xlarge	4	4	1	1、2、3、4	1
c6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6gd.large	2	2	1	1、2	1
c6gd.xlarge	4	4	1	1、2、3、4	1
c6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6gn.medium	1	1	1	1	1
c6gn.large	2	2	1	1、2	1
c6gn.xlarge	4	4	1	1、2、3、4	1
c6gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c6gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6i.large	2	1	2	1	1、2
c6i.xlarge	4	2	2	1、2	1、2
c6i.2xlarge	8	4	2	2、4	1、2
c6i.4xlarge	16	8	2	2、4、6、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
c6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
c6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c6id.large	2	1	2	1	1、2
c6id.xlarge	4	2	2	1、2	1、2
c6id.2xlarge	8	4	2	2、4	1、2
c6id.4xlarge	16	8	2	2、4、6、8	1、2
c6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
c6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c6in.large	2	1	2	1	1、2
c6in.xlarge	4	2	2	1、2	1、2
c6in.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c7a.large	2	2	1	1、2	1
c7a.xlarge	4	4	1	1、2、3、4	1
c7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
c7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1
c7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
c7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
c7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
c7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
c7g.large	2	2	1	1、2	1
c7g.xlarge	4	4	1	1、2、3、4	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7gd.large	2	2	1	1、2	1
c7gd.xlarge	4	4	1	1、2、3、4	1
c7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7gn.large	2	2	1	1、2	1
c7gn.xlarge	4	4	1	1、2、3、4	1
c7gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c7gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7i.large	2	1	2	1	1、2
c7i.xlarge	4	2	2	1、2	1、2
c7i.2xlarge	8	4	2	1、2、3、4	1、2
c7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
c7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7i-flex.large	2	1	2	1	1、2
c7i-flex.xlarge	4	2	2	1、2	1、2
c7i-flex.2xlarge	8	4	2	1、2、3、4	1、2
c7i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c7i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c7i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c7i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8a.medium	1	1	1	1	1
c8a.large	2	2	1	1、2	1
c8a.xlarge	4	4	1	1、2、3、4	1
c8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
c8a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
c8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
c8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
c8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
c8g.large	2	2	1	1、2	1
c8g.xlarge	4	4	1	1、2、3、4	1
c8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
c8gb.large	2	2	1	1、2	1
c8gb.xlarge	4	4	1	1、2、3、4	1
c8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
c8gd.large	2	2	1	1、2	1
c8gd.xlarge	4	4	1	1、2、3、4	1
c8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
c8gn.large	2	2	1	1、2	1
c8gn.xlarge	4	4	1	1、2、3、4	1
c8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
c8i.large	2	1	2	1	1、2
c8i.xlarge	4	2	2	1、2	1、2
c8i.2xlarge	8	4	2	1、2、3、4	1、2
c8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
c8id.large	2	1	2	1	1、2
c8id.xlarge	4	2	2	1、2	1、2
c8id.2xlarge	8	4	2	1、2、3、4	1、2
c8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8i-flex.large	2	1	2	1	1、2
c8i-flex.xlarge	4	2	2	1、2	1、2
c8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
c8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8in.large	2	1	2	1	1、2
c8in.xlarge	4	2	2	1、2	1、2
c8in.2xlarge	8	4	2	1、2、3、4	1、2
c8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8ine.large	2	1	2	1	1、2
c8ine.xlarge	4	2	2	1、2	1、2
c8ine.2xlarge	8	4	2	1、2、3、4	1、2
c8ine.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8ine.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8ine.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8ib.large	2	1	2	1	1、2
c8ib.xlarge	4	2	2	1、2	1、2
c8ib.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
c9g.large	2	2	1	1、2	1
c9g.xlarge	4	4	1	1、2、3、4	1
c9g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c9g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c9g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9g.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
c9gd.large	2	2	1	1、2	1
c9gd.xlarge	4	4	1	1、2、3、4	1
c9gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c9gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c9gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1
c9gd.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
c9gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	

記憶體最佳化執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r3.large	2	1	2	1	1、2
r3.xlarge	4	2	2	1、2	1、2
r3.2xlarge	8	4	2	1、2、3、4	1、2
r3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r4.large	2	1	2	1	1、2
r4.xlarge	4	2	2	1、2	1、2
r4.2xlarge	8	4	2	1、2、3、4	1、2
r4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r4.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				10、11、12、13、14、15、16	
r4.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5.large	2	1	2	1	1、2
r5.xlarge	4	2	2	2	1、2
r5.2xlarge	8	4	2	2、4	1、2
r5.4xlarge	16	8	2	2、4、6、8	1、2
r5.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5a.large	2	1	2	1	1、2
r5a.xlarge	4	2	2	2	1、2
r5a.2xlarge	8	4	2	2、4	1、2
r5a.4xlarge	16	8	2	2、4、6、8	1、2
r5a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
r5a.12xlarge	48	24	2	6、12、18、24	1、2
r5a.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5a.24xlarge	96	48	2	12、18、24、36、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5ad.large	2	1	2	1	1、2
r5ad.xlarge	4	2	2	2	1、2
r5ad.2xlarge	8	4	2	2、4	1、2
r5ad.4xlarge	16	8	2	2、4、6、8	1、2
r5ad.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
r5ad.12xlarge	48	24	2	6、12、18、24	1、2
r5ad.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5ad.24xlarge	96	48	2	12、18、24、36、48	1、2
r5b.large	2	1	2	1	1、2
r5b.xlarge	4	2	2	1、2	1、2
r5b.2xlarge	8	4	2	2、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5b.4xlarge	16	8	2	2、4、6、8	1、2
r5b.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5b.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5b.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5b.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5d.large	2	1	2	1	1、2
r5d.xlarge	4	2	2	2	1、2
r5d.2xlarge	8	4	2	2、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5d.4xlarge	16	8	2	2、4、6、8	1、2
r5d.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5d.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5dn.large	2	1	2	1	1、2
r5dn.xlarge	4	2	2	1、2	1、2
r5dn.2xlarge	8	4	2	2、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5dn.4xlarge	16	8	2	2、4、6、8	1、2
r5dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5dn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5n.large	2	1	2	1	1、2
r5n.xlarge	4	2	2	1、2	1、2
r5n.2xlarge	8	4	2	2、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r5n.4xlarge	16	8	2	2、4、6、8	1、2
r5n.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5n.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5n.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5n.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6a.large	2	1	2	1	1、2
r6a.xlarge	4	2	2	1、2	1、2
r6a.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
r6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
r6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
r6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
r6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
r6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
r6g.large	2	2	1	1、2	1
r6g.xlarge	4	4	1	1、2、3、4	1
r6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r6gd.large	2	2	1	1、2	1
r6gd.xlarge	4	4	1	1、2、3、4	1
r6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r6i.large	2	1	2	1	1、2
r6i.xlarge	4	2	2	1、2	1、2
r6i.2xlarge	8	4	2	2、4	1、2
r6i.4xlarge	16	8	2	2、4、6、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6id.large	2	1	2	1	1、2
r6id.xlarge	4	2	2	1、2	1、2
r6id.2xlarge	8	4	2	2、4	1、2
r6id.4xlarge	16	8	2	2、4、6、8	1、2
r6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6idn.large	2	1	2	1	1、2
r6idn.xlarge	4	2	2	1、2	1、2
r6idn.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r6idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r6idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r6idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6in.large	2	1	2	1	1、2
r6in.xlarge	4	2	2	1、2	1、2
r6in.2xlarge	8	4	2	1、2、3、4	1、2
r6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r7a.large	2	2	1	1、2	1
r7a.xlarge	4	4	1	1、2、3、4	1
r7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
r7a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
r7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
r7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
r7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
r7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
r7g.large	2	2	1	1、2	1
r7g.xlarge	4	4	1	1、2、3、4	1
r7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r7gd.large	2	2	1	1、2	1
r7gd.xlarge	4	4	1	1、2、3、4	1
r7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r7i.large	2	1	2	1	1、2
r7i.xlarge	4	2	2	1、2	1、2
r7i.2xlarge	8	4	2	1、2、3、4	1、2
r7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
r7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7iz.large	2	1	2	1	1、2
r7iz.xlarge	4	2	2	1、2	1、2
r7iz.2xlarge	8	4	2	1、2、3、4	1、2
r7iz.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r7iz.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r7iz.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r7iz.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r7iz.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8a.medium	1	1	1	1	1
r8a.large	2	2	1	1、2	1
r8a.xlarge	4	4	1	1、2、3、4	1
r8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
r8a.8xlarge	32	32	1	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1
r8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1
r8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
r8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
r8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
r8g.large	2	2	1	1、2	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8g.xlarge	4	4	1	1、2、3、4	1
r8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
r8gb.large	2	2	1	1、2	1
r8gb.xlarge	4	4	1	1、2、3、4	1
r8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
r8gd.large	2	2	1	1、2	1
r8gd.xlarge	4	4	1	1、2、3、4	1
r8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
r8gn.large	2	2	1	1、2	1
r8gn.xlarge	4	4	1	1、2、3、4	1
r8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
r8i.large	2	1	2	1	1、2
r8i.xlarge	4	2	2	1、2	1、2
r8i.2xlarge	8	4	2	1、2、3、4	1、2
r8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
r8id.large	2	1	2	1	1、2
r8id.xlarge	4	2	2	1、2	1、2
r8id.2xlarge	8	4	2	1、2、3、4	1、2
r8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8i-flex.large	2	1	2	1	1、2
r8i-flex.xlarge	4	2	2	1、2	1、2
r8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
r8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8in.large	2	1	2	1	1、2
r8in.xlarge	4	2	2	1、2	1、2
r8in.2xlarge	8	4	2	1、2、3、4	1、2
r8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idn.large	2	1	2	1	1、2
r8idn.xlarge	4	2	2	1、2	1、2
r8idn.2xlarge	8	4	2	1、2、3、4	1、2
r8idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idn.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8idn.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
r8ib.large	2	1	2	1	1、2
r8ib.xlarge	4	2	2	1、2	1、2
r8ib.2xlarge	8	4	2	1、2、3、4	1、2
r8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idb.large	2	1	2	1	1、2
r8idb.xlarge	4	2	2	1、2	1、2
r8idb.2xlarge	8	4	2	1、2、3、4	1、2
r8idb.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8idb.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8idb.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idb.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8idb.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8idb.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r8idb.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8idb.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
r9g.large	2	2	1	1、2	1
r9g.xlarge	4	4	1	1、2、3、4	1
r9g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r9g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r9g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9g.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
r9gd.large	2	2	1	1、2	1
r9gd.xlarge	4	4	1	1、2、3、4	1
r9gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r9gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r9gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1
r9gd.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
r9gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
u-3tb1.56xlarge	224	112	2	8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96, 100, 104, 108, 112	1、2
u-6tb1.56xlarge	224	224	1	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u-6tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1、2
u-9tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u-12tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1、2
u-18tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u-24tb1.112xlarge	448	224	2	16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7i-6tb.112xlarge	448	224	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、116、120、124、128、132、136、140、144、148、152、156、160、164、168、172、176、180、184、188、192、196、200、204、208、212、216、220、224	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7i-8tb.112xlarge	448	224	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、116、120、124、128、132、136、140、144、148、152、156、160、164、168、172、176、180、184、188、192、196、200、204、208、212、216、220、224	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7i-12tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7in-16tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7in-24tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7in-32tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
u7inh-32tb.480xlarge	1920	960	2	32、48、64、80、96、112、128、144、160、176、192、208、224、240、256、272、288、304、320、336、352、368、384、400、416、432、448、464、480、496、512、528、544、560、576、592、608、624、640、656、672、688、704、720、736、752、768、784、800、816、832、848、864、880、896、912、928、944、960	1、2
x1.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x1.32xlarge	128	64	2	4、8、12、16、20、24、28、32、36、40、44、48、52、56、60、64	1、2
x1e.xlarge	4	2	2	1、2	1、2
x1e.2xlarge	8	4	2	1、2、3、4	1、2
x1e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
x1e.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
x1e.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
x1e.32xlarge	128	64	2	4、8、12、16、20、24、28、32、36、40、44、48、52、56、60、64	1、2
x2gd.large	2	2	1	1、2	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2gd.xlarge	4	4	1	1、2、3、4	1
x2gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
x2gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
x2gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
x2idn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2idn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x2idn.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x2iedn.xlarge	4	2	2	1、2	1、2
x2iedn.2xlarge	8	4	2	2、4	1、2
x2iedn.4xlarge	16	8	2	2、4、6、8	1、2
x2iedn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2iedn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
x2iedn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x2iedn.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x2iezn.2xlarge	8	4	2	2、4	1、2
x2iezn.4xlarge	16	8	2	2、4、6、8	1、2
x2iezn.6xlarge	24	12	2	2、4、6、8、10、12	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x2iezn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
x2iezn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
x8g.large	2	2	1	1、2	1
x8g.xlarge	4	4	1	1、2、3、4	1
x8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
x8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
x8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
x8aedz.large	2	2	1	1、2	1
x8aedz.xlarge	4	4	1	1、2、3、4	1
x8aedz.3xlarge	12	12	1	1、2、4、6、8、10、12	1
x8aedz.6xlarge	24	24	1	1, 2, 3, 4, 8, 12, 16, 20, 24	1
x8aedz.12xlarge	48	48	1	1、2、3、4、5、6、7、8、16、24、32、40、48	1
x8aedz.24xlarge	96	96	1	4、6、8、10、12、14、16、32、48、64、80、96	1
x8i.large	2	1	2	1	1、2
x8i.xlarge	4	2	2	1、2	1、2
x8i.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
x8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
x8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
x8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
x8i.64xlarge	256	128	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、16、120、124、128	1、2
x8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
z1d.large	2	1	2	1	1、2
z1d.xlarge	4	2	2	1、2	1、2
z1d.2xlarge	8	4	2	2、4	1、2
z1d.3xlarge	12	6	2	2、4、6	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
z1d.6xlarge	24	12	2	2、4、6、8、10、12	1、2
z1d.12xlarge	48	24	2	4、6、8、10、12、14、16、18、20、22、24	1、2

儲存最佳化執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
d2.xlarge	4	2	2	1、2	1、2
d2.2xlarge	8	4	2	1、2、3、4	1、2
d2.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
d2.8xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
d3.xlarge	4	2	2	1、2	1、2
d3.2xlarge	8	4	2	2、4	1、2
d3.4xlarge	16	8	2	2、4、6、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
d3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
d3en.xlarge	4	2	2	1、2	1、2
d3en.2xlarge	8	4	2	2、4	1、2
d3en.4xlarge	16	8	2	2、4、6、8	1、2
d3en.6xlarge	24	12	2	2、4、6、8、10、12	1、2
d3en.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
d3en.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
h1.2xlarge	8	4	2	1、2、3、4	1、2
h1.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
h1.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
h1.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i2.xlarge	4	2	2	1、2	1、2
i2.2xlarge	8	4	2	1、2、3、4	1、2
i2.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i2.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
i3.large	2	1	2	1	1、2
i3.xlarge	4	2	2	1、2	1、2
i3.2xlarge	8	4	2	1、2、3、4	1、2
i3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i3en.large	2	1	2	1	1、2
i3en.xlarge	4	2	2	1、2	1、2
i3en.2xlarge	8	4	2	2、4	1、2
i3en.3xlarge	12	6	2	2、4、6	1、2
i3en.6xlarge	24	12	2	2、4、6、8、10、12	1、2
i3en.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3en.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i4g.large	2	2	1	1、2	1
i4g.xlarge	4	4	1	1、2、3、4	1
i4g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i4g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
i4g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i4g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
i4i.large	2	1	2	1	1、2
i4i.xlarge	4	2	2	1、2	1、2
i4i.2xlarge	8	4	2	1、2、3、4	1、2
i4i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i4i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
i4i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i4i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i4i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i4i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
i7i.large	2	1	2	1	1、2
i7i.xlarge	4	2	2	1、2	1、2
i7i.2xlarge	8	4	2	1、2、3、4	1、2
i7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
i7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
i7ie.large	2	1	2	1	1、2
i7ie.xlarge	4	2	2	1、2	1、2
i7ie.2xlarge	8	4	2	1、2、3、4	1、2
i7ie.3xlarge	12	6	2	1、2、3、4、5、6	1、2
i7ie.6xlarge	24	12	2	1、2、3、4、5、6、7、8、9、10、11、12	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i7ie.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i7ie.18xlarge	72	36	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i7ie.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
i7ie.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
i8g.large	2	2	1	1、2	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8g.xlarge	4	4	1	1、2、3、4	1
i8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
i8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
i8ge.large	2	2	1	1、2	1
i8ge.xlarge	4	4	1	1、2、3、4	1
i8ge.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i8ge.3xlarge	12	12	1	1、2、3、4、5、6、7、8、9、10、11、12	1
i8ge.6xlarge	24	24	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8ge.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8ge.18xlarge	72	72	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8ge.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
i8ge.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
im4gn.large	2	2	1	1、2	1
im4gn.xlarge	4	4	1	1、2、3、4	1
im4gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
im4gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
im4gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
im4gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
is4gen.medium	1	1	1	1	1
is4gen.large	2	2	1	1、2	1
is4gen.xlarge	4	4	1	1、2、3、4	1
is4gen.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
is4gen.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
is4gen.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

加速運算執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
dl1.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
dl2q.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、2	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				2、24、26、28、30、32、34、36、38、40、42、44、46、48	
f2.6xlarge	24	12	2	1、2、3、6、9、12	1、2
f2.12xlarge	48	24	2	1、2、3、6、9、12、15、18、21、24	1、2
f2.48xlarge	192	96	2	4、6、8、10、12、24、36、48、60、72、84、96	1、2
g3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
g3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
g4ad.xlarge	4	2	2	2	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g4ad.2xlarge	8	4	2	2、4	1、2
g4ad.4xlarge	16	8	2	2、4、8	1、2
g4ad.8xlarge	32	16	2	2、4、8、16	1、2
g4ad.16xlarge	64	32	2	2、4、8、16、32	1、2
g4dn.xlarge	4	2	2	2	1、2
g4dn.2xlarge	8	4	2	2、4	1、2
g4dn.4xlarge	16	8	2	2、4、6、8	1、2
g4dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
g4dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
g4dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g5g.xlarge	4	4	1	1、2、3、4	1
g5g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
g5g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
g5g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g5g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
g6.xlarge	4	2	2	1、2	1、2
g6.2xlarge	8	4	2	1、2、3、4	1、2
g6.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g6.8xlarge	32	16	2	1、2、4、6、8、10、12、14、16	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g6.12xlarge	48	24	2	1、2、3、6、9、12、15、18、21、24	1、2
g6.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1、2
g6.24xlarge	96	48	2	1、2、3、4、5、6、12、18、24、30、36、42、48	1、2
g6.48xlarge	192	96	2	4、6、8、10、12、24、36、48、60、72、84、96	1、2
g6e.xlarge	4	2	2	1、2	1、2
g6e.2xlarge	8	4	2	1、2、3、4	1、2
g6e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g6e.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
g6e.12xlarge	48	24	2	3、6、9、12、15、18、21、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g6e.16xlarge	64	32	2	4、8、12、16、20、24、28、32	1、2
g6e.24xlarge	96	48	2	6、12、18、24、30、36、42、48	1、2
g6e.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2
g6f.large	2	1	2	1	1、2
g6f.xlarge	4	2	2	1、2	1、2
g6f.2xlarge	8	4	2	1、2、3、4	1、2
g6f.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
gr6.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
gr6.8xlarge	32	16	2	1、2、4、6、8、10、12、14、16	1、2
gr6f.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g7.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g7.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
g7.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
g7.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g7.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g7.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				178、180、182、184、186、188、190、192	
g7e.2xlarge	8	4	2	1、2、3、4	1、2
g7e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g7e.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
g7e.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
g7e.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
g7e.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
inf1.xlarge	4	2	2	2	1、2
inf1.2xlarge	8	4	2	2、4	1、2
inf1.6xlarge	24	12	2	2、4、6、8、10、12	1、2
inf1.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
inf2.xlarge	4	2	2	1、2	1、2
inf2.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
inf2.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
inf2.48xlarge	192	96	2	4、8、12、16、20、24、28、32、64、96	1、2
p3.2xlarge	8	4	2	1、2、3、4	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
p3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
p3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
p3dn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
p4d.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
p4de.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
p5.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
p5.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2
p5e.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
p5en.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
p6-b200.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
p6-b300.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
p6e-gb200.36xlarge	144	144	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96、97、98、99、100、101、1	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				02、103、104、105、106、107、108、109、110、111、112、113、114、115、116、117、118、119、120、121、122、123、124、125、126、127、128、129、130、131、132、133、134、135、136、137、138、139、140、141、142、143、144	
trn1.2xlarge	8	4	2	2、4	1、2
trn1.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
trn1n.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
trn2.3xlarge	12	6	2	1、2、3、4、5、6	1、2
trn2.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
trn2u.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
vt1.3xlarge	12	6	2	6	1、2
vt1.6xlarge	24	12	2	6、12	1、2
vt1.24xlarge	96	48	2	6、12、48	1、2

高效能運算執行個體

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
hpc6id.32xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、	1

執行個體類型	預設 vCPU	預設 CPU 核心	預設每核心執行緒	有效的 CPU 核心	每個核心的有效執行緒
				24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	
hpc8a.96xlarge	192	192	1	24、48、72、96、120、144、168、192	1

指定 Amazon EC2 執行個體的 CPU 選項

您可以在執行個體啟動期間或之後指定 CPU 選項。

任務

- 停用同時多執行緒
- 啟動時指定 vCPU 的自訂數量
- 在啟動範本中指定 vCPU 的自訂數量
- 變更 EC2 執行個體的 CPU 選項

停用同時多執行緒

若要停用也稱為超執行緒的同時多執行緒 (SMT)，請為每個核心指定 1 個執行緒。

Console

在執行個體啟動期間停用 SMT

1. 遵照 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體 程序並視需要設定執行個體。
2. 展開進階詳細資訊，然後選取指定 CPU 選項核取方塊。
3. 針對 Core count (核心計數)，選擇所需的 CPU 核心數量。在本範例中，若要指定 r5.4xlarge 執行個體的預設 CPU 核心計數，請選擇 8。

4. 若要停用 SMT，請在每核心執行緒中選擇 1。
5. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

在執行個體啟動期間停用 SMT

使用 run-instances AWS CLI 命令，並為 `--cpu-options` 參數指定 1 `ThreadsPerCore` 的值。在 `CoreCount` 指定 CPU 核心數目。在本範例中，若要指定 `r7i.4xlarge` 執行個體的預設 CPU 核心計數，請指定值 8。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=8,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

在執行個體啟動期間停用 SMT

使用 New-EC2Instance 命令，為 `-CpuOptions` 參數的 `ThreadsPerCore` 指定值 1。在 `CoreCount` 指定 CPU 核心數目。在本範例中，若要指定 `r7i.4xlarge` 執行個體的預設 CPU 核心計數，請指定值 8。

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=8; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

> **Note**
>
> 若要停用現有執行個體的 SMT，請依照 變更 EC2 執行個體的 CPU 選項 中所示的流程操作，並將每個核心執行的執行緒數量變更為 1。

啟動時指定 vCPU 的自訂數量

當您從 EC2 主控台或 AWS CLI啟動執行個體時，您可以自訂 CPU 核心數量和每個核心的執行緒數量。本節中的範例使用具有下列預設設定的`r5.4xlarge` 執行個體類型：

- CPU 核心：8
- 每核心執行緒：2

執行個體啟動時，預設會有執行個體類型可用的 vCPUs 數量上限。對於此執行個體類型，總共 16 個 vCPU (8 個核心，每個執行 2 個執行緒)。如需有關此執行個體類型的詳細資訊，請參閱 記憶體最佳化執行個體。

以下範例啟動有 4 個 vCPU 的 `r5.4xlarge` 執行個體。

Console

在執行個體啟動期間指定 vCPU 的自訂數量

1. 遵照 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體 程序並視需要設定執行個體。
2. 展開進階詳細資訊，然後選取指定 CPU 選項核取方塊。
3. 指定 2 個 CPU 核心和每核心 2 個執行緒，即可得到 4 個 vCPU，如下所示：
   - 在核心計數中，選擇 2。
   - 針對 Threads per core (每核心執行緒)，選擇 2。
4. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

在執行個體啟動期間指定 vCPU 的自訂數量

使用 run-instances AWS CLI 命令，並在 `--cpu-options` 參數中指定 CPU 核心數量和執行緒數量。您可以指定 2 個 CPU 核心和每核心 2 個執行緒，如此便能得到 4 個 vCPU。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=2,ThreadsPerCore=2" \
    --key-name my-key-pair
```

或者，指定 4 個 CPU 核心和每核心 1 個執行緒 (停用 SMT)，也可以產生 4 個 vCPU：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=4,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

在執行個體啟動期間指定 vCPU 的自訂數量

使用 New-EC2Instance 命令，在 `-CpuOptions` 參數中指定 CPU 核心數量和執行緒數量。您可以指定 2 個 CPU 核心和每核心 2 個執行緒，如此便能得到 4 個 vCPU。

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=2; ThreadsPerCore=2} `
    -KeyName 'my-key-pair'
```

或者，指定 4 個 CPU 核心和每核心 1 個執行緒 (停用 SMT)，也可以產生 4 個 vCPU：

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=4; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

在啟動範本中指定 vCPU 的自訂數量

可在啟動範本中自訂執行個體的 CPU 核心數量和每個核心的執行緒數量。本節中的範例使用具有下列預設設定的`r5.4xlarge` 執行個體類型：

- CPU 核心：8
- 每核心執行緒：2

執行個體啟動時，預設會有執行個體類型可用的 vCPUs 數量上限。對於此執行個體類型，總共 16 個 vCPU (8 個核心，每個執行 2 個執行緒)。如需有關此執行個體類型的詳細資訊，請參閱 記憶體最佳化執行個體。

以下範例建立的啟動範本會指定具有 4 個 vCPU 的 `r5.4xlarge` 執行個體的組態。

Console

在啟動範本中指定 vCPU 的自訂數量

1. 遵照 指定參數來建立啟動範本 程序並視需要設定啟動範本。
2. 展開進階詳細資訊，然後選取指定 CPU 選項核取方塊。
3. 指定 2 個 CPU 核心和每核心 2 個執行緒，即可得到 4 個 vCPU，如下所示：
   - 在核心計數中，選擇 2。
   - 針對 Threads per core (每核心執行緒)，選擇 2。
4. 在摘要面板中，檢閱執行個體組態，然後選擇建立啟動範本。如需詳細資訊，請參閱將執行個體啟動參數存放在 Amazon EC2 啟動範本中。

AWS CLI

在啟動範本中指定 vCPU 的自訂數量

使用 create-launch-template AWS CLI 命令，並在 `CpuOptions` 參數中指定 CPU 核心數量和執行緒數量。您可以指定 2 個 CPU 核心和每核心 2 個執行緒，如此便能得到 4 個 vCPU。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForCPUOptions \
    --version-description CPUOptionsVersion1 \
    --launch-template-data file://template-data.json
```

以下是包含啟動範本資料的 JSON 檔案範例，其中包括此範例的執行個體組態的 CPU 選項。

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
```

```
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":2,
        "ThreadsPerCore":2
    }
}
```

或者，指定 4 個 CPU 核心和每核心 1 個執行緒 (停用 SMT)，也可以產生 4 個 vCPU：

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":1
    }
}
```

PowerShell

在啟動範本中指定 vCPU 的自訂數量

使用 New-EC2LaunchTemplate。

```
New-EC2LaunchTemplate `
    -LaunchTemplateName 'TemplateForCPUOptions' `
```

```
    -VersionDescription 'CPUOptionsVersion1' `
    -LaunchTemplateData (Get-Content -Path 'template-data.json' | ConvertFrom-Json)
```

變更 EC2 執行個體的 CPU 選項

由於您的需求會隨著時間發生變化，您可能想要變更現有執行個體的 CPU 選項組態。每個在執行個體上執行的執行緒都稱為虛擬 CPU (vCPU)。您可以在 Amazon EC2 主控台、 AWS CLI API 或 SDKs 中變更為現有執行個體執行的 vCPUs 數量。執行個體狀態必須是 `Stopped`，才能進行此變更。

若要檢視主控台或命令列步驟，請選取符合您環境的標籤。如需 API 請求和回應資訊，請參閱「Amazon EC2 API 參考」中的 ModifyInstanceCpuOptions。

Console

請依照此程序操作，從 AWS 管理主控台變更執行個體的作用中 vCPU 數量。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。這會開啟為目前 AWS 區域定義的執行個體清單。
3. 從執行個體清單中，選取執行個體。或者，您可以選取執行個體連結以開啟執行個體詳細資訊頁面。
4. 如果執行個體正在執行，您必須先將其停止才能繼續。從執行個體狀態功能表中，選擇停止執行個體。
5. 若要變更 vCPU 組態，請從動作功能表的執行個體設定中，選擇變更 CPU 選項。這會開啟變更 CPU 選項頁面。
6. 選擇下列其中一個 CPU 選項，以變更執行個體的組態。

   使用預設 CPU 選項

   此選項會將執行個體重設為執行個體類型的預設 vCPU 數量。預設為執行所有 CPU 核心的所有執行緒。

   指定 CPU 選項

   此選項會啟用在執行個體上執行的 vCPUs 數量組態。
7. 若您選擇指定 CPU 選項，則會顯示作用中 vCPU 欄位。
   - 使用第一個選擇工具設定為每個 CPU 核心執行的執行緒數量。若要停用同步多執行緒，請選擇 1。

- 使用第二個選擇工具會設定為執行個體執行的 CPU 數量。

當您對 CPU 選項選擇工具進行變更時，以下欄位會動態更新。

- 作用中 vCPU：根據您所進行的選擇，CPU 核心的數量乘以每核心執行緒。例如，如果您選取 2 個執行緒和 4 個核心，則等於 8 vCPU。
- vCPUs 總數：執行個體類型的 vCPU 數量上限。例如，對於 `m6i.4xlarge` 執行個體類型，這是 16 個 vCPU (8 個核心，每個執行 2 個執行緒)。

8. 若要套用您的更新，請選擇變更。

AWS CLI

請依照此程序操作，從 AWS CLI變更執行個體的作用中 vCPU 數量。

使用 modify-instance-cpu-options 命令，並指定 `--core-count` 參數中執行的 CPU 核心數量，以及 `--threads-per-core` 參數中每個核心執行的執行緒數量。

下列範例顯示 `m6i.4xlarge` 執行個體類型上兩個可能的組態，以在指定的執行個體上執行 8 個 vCPU。此執行個體類型的預設值為 16 個 vCPU (8 個核心，每個核心執行 2 個執行緒)。

範例 1：執行 4 個 CPU 核心，每個核心有 2 個執行緒，總共 8 個 vCPU。

```
aws ec2 modify-instance-cpu-options \
    --instance-id i-1234567890abcdef0 \

    --core-count=4 \
    --threads-per-core=2
```

範例 2：將每個核心執行的執行緒數量變更為 1，停用同時多執行緒。產生的組態也會執行總共 8 vCPU (8 個 CPU 核心，每個核心 1 個執行緒)。

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=8 \
    --threads-per-core=1
```

PowerShell

變更執行個體的作用中 vCPU 數量

使用 Edit-EC2InstanceCpuOption cmdlet 命令，並指定 `-CoreCount` 參數中執行的 CPU 核心數量，以及 `ThreadsPerCore` 參數中每個核心執行的執行緒數量。

範例 1：執行 4 個 CPU 核心，每個核心有 2 個執行緒，總共 8 個 vCPU。

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 4 `
    -ThreadsPerCore 2
```

範例 2：將每個核心執行的執行緒數量變更為 1，停用同時多執行緒。產生的組態也會執行總共 8 vCPU (8 個 CPU 核心，每個核心 1 個執行緒)。

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 8 `
    -ThreadsPerCore 1
```

檢視 Amazon EC2 執行個體的 CPU 執行緒與核心

您可透過描述執行個體，檢視現有執行個體的 CPU 選項。

Console

檢視執行個體的 CPU 選項

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側的導覽窗格中，選取 Instances (執行個體)，然後選取執行個體。
3. 在 Details (詳細資料) 標籤的 Host and placement group (主機與置放群組) 下，尋找 Number of vCPUs (vCPU 的數目)。

AWS CLI

檢視執行個體的 CPU 選項

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
```

```
    --query Reservations[].Instances[].CpuOptions
```

以下為範例輸出。CoreCount 欄位會指出執行個體的核心數量。ThreadsPerCore 欄位顯示每核心的執行緒數量。

```
[
    {
        "CoreCount": 24,
        "ThreadsPerCore": 2
    },
]
```

PowerShell

檢視執行個體的 CPU 選項

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId 'i-1234567890abcdef0').Instances.CpuOptions
```

以下為範例輸出。

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
          24        2
```

或者，若要檢視 CPU 資訊，您可以連線到執行個體並使用下列其中一個系統工具：

- Windows 執行個體上的 Windows Task Manager
- Linux 執行個體上的 lscpu 命令

您可以使用 AWS Config 記錄、評估、稽核和評估執行個體的組態變更，包括終止的執行個體。如需詳細資訊，請參閱《AWS Config 開發人員指南》中的開始使用 AWS Config。

針對含許可的執行個體最佳化 CPU

Microsoft SQL Server 等工作負載通常需要高層級的記憶體和 IOPS，但 vCPU 計數較低。 AWS 提供廣泛的執行個體類型，可涵蓋您大部分的基礎設施需求。不過，若要降低 Windows 和 Microsoft SQL

Server 的 vCPU 型授權成本，您可以自訂 EC2 執行個體上執行vCPUs 數量，同時維護相同的記憶體、儲存體和網路規格。此方法可以節省包含授權和 Bring-Your-Own-License (BYOL) 工作負載的 vCPU 型授權成本。您可以使用 主控台或 Amazon EC2 API，在啟動或修改執行個體時指定 CPU 選項。如需說明，請參閱指定 Amazon EC2 執行個體的 CPU 選項。

如需詳細資訊，請參閱這篇部落格文章，了解為 SQL Server 工作負載最佳化 CPU 的最佳實務。

支援的許可類型

「最佳化 CPU」支援對從含許可 AMI 啟動之執行個體的下列許可組態類型，依作用中 CPU 數量帳單。如需有關許可類型的詳細資訊，請參閱 AMI 帳單資訊欄位。

含許可 AMI 執行個體帳單

已含許可	使用操作	每個 vCPU 小時的價格
Windows Server	RunInstances:0002	0.046 美元
Windows Server 搭配 SQL Server Enterprise	RunInstances:0102	0.421 美元
Windows Server 搭配 SQL Server Standard	RunInstances:0006	0.166 美元
Windows Server 搭配 SQL Server Web	RunInstances:0202	0.063 美元

支援的購買選項

針對含許可的執行個體，「最佳化 CPU」支援下列購買選項：

- On-Demand
- Savings Plans

> **Warning**
>
> 若您使用預留執行個體，在同一付款帳戶中為含許可執行個體設定「最佳化 CPU」時，可能無法套用折扣。我們建議您使用 Savings Plans 來降低基於 vCPU 的授權成本，並為運算成本提供相當的節省效果。
>
> 在 2025 年 10 月 15 日之前，對相同執行個體類型的 Windows 及 SQL Server 同時使用「最佳化 CPU」和「預留執行個體」的帳戶，已新增至退出清單以維持其目前的帳單體驗。若要利用「最佳化 CPU」許可節省，請聯絡 AWS 支援 中心，將您從退出清單中移除。

「最佳化 CPU」如何節省授權費用

下列範例協助說明設定 CPU 用量時可能實現的成本節省。

範例 1：預設帳單此範例顯示從含許可的 Windows 及 SQL Server Enterprise AMI 啟動的 r7i.8xlarge 執行個體，該執行個體以執行個體類型的預設 CPU 組態 (32 個 vCPU) 執行了 100 小時 (3200 個 vCPU 小時)。

帳單會顯示一個明細項目，其中包含了用量和授權費用的合併費率。

For illustration purposes only **Amazon Web Services, Inc. charges by service** **Total services: 1** **Description**	**Usage Quantity**	**Total charges** **$1,558.88** **Amount in USD**
Elastic Compute Cloud		$1,558.88
|- US East (N. Virginia)		$1,558.88
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$1,558.88
| |- $15.5888 per On Demand Windows with SQL Server Enterprise r7i.8xlarge Instance Hour	100 Hrs	$1,558.88

範例 2：最佳化 CPU 帳單此範例顯示從含許可的 Windows 及 SQL Server Enterprise AMI 啟動的 r7i.8xlarge 執行個體。為節省許可費用，作用中 CPU 數量已減少為 16 vCPU。執行個體之後會使用新組態執行 100 小時。

帳單顯示下列兩個明細項目。

帳單描述：Elastic Compute Cloud

第一個明細項目顯示執行 100 小時 (211.68 USD) 的 Windows 及 SQL Server 執行個體的基準成本。

帳單描述：Amazon EC2 最佳化 CPU 含許可第三方費用

第二個明細項目涵蓋以帳單期間作用中 vCPU 數量為基礎的授權費用 (673.60 USD)。

For illustration purposes only Amazon Web Services, Inc. charges by service Total services: 2 Description	Usage Quantity	Total charges $885.28 Amount in USD
Elastic Compute Cloud		$211.68
|- US East (N. Virginia)		$211.68
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$211.68
| |- $2.1168 per On Demand Windows and SQL Server Enterprise without licenses r7i.8xlarge Instance Hr	100 Hrs	$211.68
Amazon EC2 Optimize CPU License Included Third Party Fees		$673.60
|- US East (N. Virginia)		$673.60
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$673.60
| |- $0.421 per vCPU-Hour of On Demand Windows with SQL Server licenses on r7i Instance	1,600 vCPU-Hrs	$673.60

範例 3：使用 Savings Plans 最佳化 CPU 帳單此範例顯示從含許可的 Windows 及 SQL Server Enterprise AMI 啟動的 r7i.8xlarge 執行個體。為節省許可費用，作用中 CPU 數量已減少為 16 vCPU。執行個體之後會使用新組態執行 100 小時。

一項為期一年、每小時 1.60 USD (四捨五入) 承諾額的無預付款 Compute Savings Plan，可進一步節省成本，降低 Windows 及 SQL Server 執行個體的基準成本。Savings Plans 的承諾額以每小時 1.53362 USD 的 Savings Plans 費率，涵蓋 r7i.8xlarge 執行個體整整 100 小時的使用時間。

帳單會顯示下列明細項目。

帳單描述：適用於運算用量的 Savings Plans

第一個明細項目顯示完整 100 小時用量的 Savings Plan 承諾額 (160.00 USD)。

帳單描述：Elastic Compute Cloud

第二個明細項目包括兩個項目。第一個項目顯示未使用 Savings Plan 時，執行了 100 小時的 Windows 及 SQL Server 執行個體之基準成本 (211.68 USD)。第二個明細項目顯示 Compute Savings Plan (-211.68 USD) 涵蓋了全部基準成本，因此產生此明細項目的淨成本為 0。

帳單描述：Amazon EC2 最佳化 CPU 含許可第三方費用

第三個明細項目涵蓋以帳單期間作用中 vCPU 數量為基礎的授權費用 (673.60 USD)。

For illustration purposes only Amazon Web Services, Inc. charges by service Total services: 3 Description	Usage Quantity	Total charges $833.60 Amount in USD
Savings Plans for Compute usage		$160.00
|- Any		$160.00
|- Compute Savings Plans		$160.00
| |- 1 year No Upfront Compute Savings Plan	100 Hrs	$160.00
Elastic Compute Cloud		$0.00
|- US East (N. Virginia)		$0.00
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$0.00
| |- $2.1168 per On Demand Windows and SQL Server Enterprise without licenses r7i.8xlarge Instance Hr	100 Hrs	$211.68
| |- r7i.8xlarge Windows with SQL Server Enterprise without licenses instance usage covered by CSP	100 Hrs	($211.68)
Amazon EC2 Optimize CPU License Included Third Party Fees		$673.60
|- US East (N. Virginia)		$673.60
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$673.60
| |- $0.421 per vCPU-Hour of On Demand Windows with SQL Server licenses on r7i Instance	1,600 vCPU-Hrs	$673.60

Amazon EC2 執行個體的 AMD SEV-SNP

AMD Secure Encrypted Virtualization-Secure Nested Paging (AMD SEV-SNP) 是一項 CPU 功能，可提供下列屬性：

- 證明 – AMD SEV-SNP 可讓您擷取已簽署的證明報告，其中包含可用於驗證執行個體狀態和身分的密碼編譯措施，且其正在真正的 AMD 硬體上執行。如需詳細資訊，請參閱使用 AMD SEV-SNP 來認證 Amazon EC2 執行個體。
- 記憶體加密 – 從 AMD EPYC （米蘭）、 AWS Graviton2 和 Intel Xeon 可擴充 (Ice Lake) 處理器開始，執行個體記憶體一律會加密。

您可以搭配專用主機或共用租用使用 AMD SEV-SNP。

目錄

- 概念和術語
- 要求
- 考量事項
- 定價
- 尋找支援 AMD SEV-SNP 的 EC2 執行個體類型
- 為 EC2 執行個體啟用 AMD SEV-SNP
- 使用 AMD SEV-SNP 來認證 Amazon EC2 執行個體

概念和術語

在開始使用 AMD SEV-SNP 之前，請確保您已熟悉下列概念和術語。

AMD SEV-SNP 認證報告

AMD SEV-SNP 認證報告是執行個體可向 CPU 請求的文件。AMD SEV-SNP 認證報告可用於驗證執行個體的狀態和身分，以及確認執行個體是否在准許的 AMD 環境中執行。該報告包含啟動測量，這是執行個體初始啟動狀態的密碼編譯雜湊，包含其初始執行個體記憶體內容和 vCPU 的初始狀態。AMD SEV-SNP 認證報告會使用 VCEK 簽章（在專用主機上）或 VLEK 簽章（在共用租用上）簽署，該簽章會鏈結回信任的 AMD 根目錄。

VCEK

版本控制晶片背書金鑰 (VCEK) 是每個晶片簽署金鑰，經過 AMD 認證，並由 AMD CPU 用來簽署專用主機的 AMD SEV-SNP 認證報告。VCEK 簽章可以使用 AMD 提供的憑證進行驗證。

VLEK

Versioned Loaded Endorsement Key (VLEK) 是一種版本控制的簽署金鑰，經過 AMD 認證，並由 AMD CPU 用來簽署共用租用的 AMD SEV-SNP 認證報告。可以使用 AMD 提供的憑證來驗證 VLEK 簽章。

OVMF 二進位

Open Virtual Machine Firmware (OVMF) 是用來為執行個體提供 UEFI 環境的早期開機程式碼。早期開機程式碼在 AMI 中的程式碼啟動之前執行。OVMF 也會尋找並執行 AMI 中提供的開機載入器。更多詳細資訊，請參閱 OVMF 儲存庫。

要求

AMD SEV-SNP 的需求取決於您使用的是專用主機或共用租用。

專用主機的需求

若要搭配專用主機使用 AMD SEV-SNP，您必須符合下列要求：

- 使用下列其中一個支援的執行個體類型：
  - 一般用途：`m6a.large | m6a.xlarge | m6a.2xlarge | m6a.4xlarge | m6a.8xlarge`
  - 運算優化：`c6a.large | c6a.xlarge | c6a.2xlarge | c6a.4xlarge | c6a.8xlarge | c6a.12xlarge | c6a.16xlarge`

  - 記憶體優化：r6a.large | r6a.xlarge | r6a.2xlarge | r6a.4xlarge
- 配置已啟用 AMD SEV-SNP 的專用主機。如需詳細資訊，請參閱使用 AMD SEV-SNP 配置專用主機。
- 在任何商業環境中啟動執行個體 AWS 區域。
- 使用具有 uefi 或 uefi-preferred 開機模式的 AMI，以及支援 AMD SEV-SNP 的作業系統。針對 AWS，在 AL2023、RHEL 9.3、SLES 15 SP4 和 Ubuntu 23.04 與更新版本上支援 AMD SEV-SNP。

共用租用的需求

若要搭配共用租用使用 AMD SEV-SNP，您必須符合下列要求：

- 使用下列其中一個支援的執行個體類型：
  - 一般用途：m6a.large | m6a.xlarge | m6a.2xlarge | m6a.4xlarge | m6a.8xlarge
  - 運算優化：c6a.large | c6a.xlarge | c6a.2xlarge | c6a.4xlarge | c6a.8xlarge | c6a.12xlarge | c6a.16xlarge
  - 記憶體優化：r6a.large | r6a.xlarge | r6a.2xlarge | r6a.4xlarge
- 在支援的 中啟動執行個體 AWS 區域：美國東部（俄亥俄）或歐洲（愛爾蘭）。
- 使用具有 uefi 或 uefi-preferred 開機模式的 AMI，以及支援 AMD SEV-SNP 的作業系統。針對 AWS，在 AL2023、RHEL 9.3、SLES 15 SP4 和 Ubuntu 23.04 與更新版本上支援 AMD SEV-SNP。

考量事項

您僅可在啟動執行個體時啟用 AMD SEV-SNP。在為執行個體啟動啟用 AMD SEV-SNP 時，將適用以下規則。

- 啟用後，便無法停用 AMD SEV-SNP。其在整個執行個體生命週期中皆會保持啟用狀態。
- 您僅可變更執行個體類型為其他支援 AMD SEV-SNP 的執行個體類型。
- 不支援 Hibernation 和 Nitro Enclaves。

專用主機的其他考量事項

當您搭配專用主機使用 AMD SEV-SNP 時，適用下列其他考量事項：

- 當您配置主機時，會套用韌體更新。若要取得最新的韌體，請釋出現有的主機並配置新的主機。如需詳細資訊，請參閱更新 AMD SEV-SNP 專用主機上的韌體。
- 您可以在支援 AMD SEV-SNP 的專用主機上啟動啟用或停用 AMD SEV-SNP 的執行個體。
- 支援主機資源群組和專用主機預留。

共用租用的其他考量事項

當您搭配共用租用使用 AMD SEV-SNP 時，適用下列其他考量事項：

- 如果執行個體的基礎主機已排定進行維護，則您將在事件發生前 14天收到排定的事件通知。您必須手動停止或重新啟動執行個體，才能將其移至新主機。

定價

專用執行個體

在專用主機上使用 AMD SEV-SNP 無需額外費用。您只需支付標準專用主機定價。如需詳細資訊，請參閱專用主機定價。

共用租用

當您在共用租用上啟用 AMD SEV-SNP 的情況下啟動 Amazon EC2 執行個體時，需支付額外的每小時使用費，相當於所選執行個體類型的隨需每小時費率的 10%。此 AMD SEV-SNP 用量費是針對 Amazon EC2 執行個體之用量而單獨收取的費用。預留執行個體、Savings Plans 和作業系統用量不會影響此費用。

如果您將 Spot 執行個體設定為在共享租用上啟用 AMD SEV-SNP 時啟動，則需支付額外的每小時使用費，相當於所選執行個體類型隨需每小時費率的 10%。如果配置策略使用價格作為輸入，則 Spot 機群不包含此額外費用；僅使用 Spot 價格。

尋找支援 AMD SEV-SNP 的 EC2 執行個體類型

可以尋找支援 AMD SEV-SNP 的執行個體類型。Amazon EC2 主控台不會顯示執行個體類型的此資訊。

AWS CLI

若要尋找支援 AMD SEV-SNP 的執行個體類型

使用下列 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
    --filters Name=processor-info.supported-features,Values=amd-sev-snp \
    --query 'InstanceTypes[*].[InstanceType]' \
    --output text | sort
```

以下為範例輸出。

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

PowerShell

若要尋找支援 AMD SEV-SNP 的執行個體類型

使用 Get-EC2InstanceType cmdlet。

```
(Get-EC2InstanceType `
    -Filter @{Name="processor-info.supported-features"; Values="amd-sev-
snp"}).InstanceType.Value | Sort-Object
```

以下為範例輸出。

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
```

```
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

為 EC2 執行個體啟用 AMD SEV-SNP

可以在啟用 AMD SEV-SNP 的情況下啟動執行個體。無法在啟動後啟用 AMD SEV-SNP。

目錄

- 使用 AMD SEV-SNP 配置專用主機
- 在 AMD SEV-SNP 專用主機上啟動執行個體
- 更新 AMD SEV-SNP 專用主機上的韌體
- 在共用租用上使用 AMD SEV-SNP 啟動執行個體
- 檢查是否已為 AMD SEV-SNP 啟用 EC2 執行個體

使用 AMD SEV-SNP 配置專用主機

在專用主機上使用 AMD SEV-SNP 啟動執行個體之前，您必須先配置已啟用 AMD SEV-SNP 支援的專用主機。您可以依執行個體系列 （例如 m6a、r6a 或 c6a) 配置專用主機，然後您可以從主機上的該系列執行任何支援的執行個體類型。您可以在任何商業 中使用 AMD SEV-SNP 配置專用主機 AWS 區域。

Console

配置已啟用 AMD SEV-SNP 的專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用主機。
3. 選擇配置專用主機。
4. 針對 AMD SEV-SNP，選擇啟用。

5. 視需要設定剩餘的主機設定，然後選擇配置。

AWS CLI

配置已啟用 AMD SEV-SNP 的專用主機

使用 allocate-hosts 命令與 `--instance-type` 和 `--cpu-options` 參數。

```
aws ec2 allocate-hosts \
    --instance-type "m6a.large" \
    --availability-zone "us-east-1a" \
    --auto-placement "off" \
    --cpu-options AmdSevSnp=enabled \
    --quantity 1
```

以下為範例輸出。

```
{
    "HostIds": [
        "h-0123456789abcdef0"
    ]
}
```

> **Note**
>
> 當您在啟用 AMD SEV-SNP 的情況下配置專用主機時，主機會進入 `configuring` 狀態，同時 AWS 套用最新的韌體。這可能需要幾分鐘到 2 小時。在 `configuring` 狀態期間會向您收取費用。

在 AMD SEV-SNP 專用主機上啟動執行個體

您可以在設定專用主機後，在啟用 AMD SEV-SNP 的情況下啟動執行個體。

> **Note**
>
> 您也可以在配置 AMD SEV-SNP 的專用主機上啟動未啟用 AMD SEV-SNP 的執行個體。AMD SEV-SNP 和非 AMD SEV-SNP 執行個體都可以在相同的主機上一起執行。

AWS CLI

在專用主機上使用 AMD SEV-SNP 啟動執行個體

使用 run-instances命令搭配 `--cpu-options`和 `--placement`選項。將 *h-0123456789abcdef0* 取代為您配置的專用主機 ID。

```
aws ec2 run-instances \
    --instance-type m6a.xlarge \
    --image-id ami-0123456789example \
    --cpu-options AmdSevSnp=enabled \
    --placement HostId=h-0123456789abcdef0
```

PowerShell

在專用主機上使用 AMD SEV-SNP 啟動執行個體

使用 New-EC2Instance cmdlet 搭配 `-CpuOption`和 `-Placement_HostId` 參數。

```
New-EC2Instance `
    -InstanceType m6a.xlarge `
    -ImageId ami-0123456789example `
    -CpuOption @{AmdSevSnp="enabled"} `
    -Placement_HostId h-0123456789abcdef0
```

更新 AMD SEV-SNP 專用主機上的韌體

當您在啟用 AMD SEV-SNP 的情況下配置專用主機時，會將最新的可用韌體 AWS 套用至主機。韌體更新可以包含 AMD 的安全修補程式和功能改進。

若要套用韌體更新，請配置新主機（接收最新韌體）、將執行個體移至新主機，然後釋出舊主機。

1. 配置啟用 AMD SEV-SNP 的新專用主機。新主機會收到最新的可用韌體。如需詳細資訊，請參閱使用 AMD SEV-SNP 配置專用主機。
2. 在舊的專用主機上停止執行中的執行個體。

   ```
   aws ec2 stop-instances \
       --instance-ids i-0123456789example
   ```

3. 修改執行個體置放以鎖定新的專用主機。使用 modify-instance-placement 命令。

```
aws ec2 modify-instance-placement \
    --instance-id i-0123456789example \
    --host-id h-09876543210abcdef
```

4. 在新的專用主機上啟動執行個體。

```
aws ec2 start-instances \
    --instance-ids i-0123456789example
```

5. 釋出舊的專用主機。使用 release-hosts 命令。

```
aws ec2 release-hosts \
    --host-ids h-0123456789abcdef0
```

> **Note**
>
> 您會收到專用主機的維護通知，就像收到其他專用主機維護事件的維護通知一樣。當您收到通知時，請依照上述步驟移至具有更新韌體的新主機。

在共用租用上使用 AMD SEV-SNP 啟動執行個體

您可以在美國東部（俄亥俄）和歐洲（愛爾蘭）的共用租用上使用 AMD SEV-SNP 啟動執行個體。

AWS CLI

在共用租用上使用 AMD SEV-SNP 啟動執行個體

搭配使用 run-instances 命令與 `--cpu-options` 選項。

```
aws ec2 run-instances \
    --instance-type m6a.xlarge \
    --image-id ami-0123456789example \
    --cpu-options AmdSevSnp=enabled
```

PowerShell

在共用租用上使用 AMD SEV-SNP 啟動執行個體

使用 New-EC2Instance cmdlet 搭配 `-CpuOption` 參數。

```
New-EC2Instance `
    -InstanceType m6a.xlarge `
    -ImageId ami-0123456789example `
    -CpuOption @{AmdSevSnp="enabled"}
```

檢查是否已為 AMD SEV-SNP 啟用 EC2 執行個體

可以尋找已為 AMD SEV-SNP 啟用的執行個體。Amazon EC2 主控台不會顯示此資訊。

AWS CLI

若要檢查是否已為執行個體啟用 AMD SEV-SNP

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].CpuOptions
```

以下為範例輸出。如果 CpuOptions 中不存在 AmdSevSnp，則會停用 AMD SEV-SNP。

```
[
    {
        "AmdSevSnp": "enabled",
        "CoreCount": 1,
        "ThreadsPerCore": 2
    }
]
```

PowerShell

若要檢查是否已為執行個體啟用 AMD SEV-SNP

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.CpuOptions
```

以下為範例輸出。如果 AmdSevSnp 的值不存在，則會停用 AMD SEV-SNP。

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
enabled   1         2
```

AWS CloudTrail

在執行個體啟動請求 AWS CloudTrail 的情況下，下列屬性表示執行個體已啟用 AMD SEV-SNP。

```
"cpuOptions": {"AmdSevSnp": "enabled"}
```

使用 AMD SEV-SNP 來認證 Amazon EC2 執行個體

認證是一種程序，可讓您的執行個體證明其狀態和身分。當您為執行個體啟用 AMD SEV-SNP 後，可以向基礎處理器請求 AMD SEV-SNP 認證報告。AMD SEV-SNP 認證報告包含初始訪客記憶體內容的密碼編譯雜湊 (稱為啟動測量) 和初始 vCPU 狀態。認證報告會使用 VCEK 簽章（在專用主機上）或 VLEK 簽章（在共用租用上）簽署，該簽章會鏈結回信任的 AMD 根目錄。您可以使用認證報告中包含的啟動測量來驗證執行個體是否在正版 AMD 環境中執行，並驗證用於啟動執行個體的初始啟動程式碼。

先決條件

啟動已為 AMD SEV-SNP 啟用的執行個體。如需詳細資訊，請參閱為 EC2 執行個體啟用 AMD SEV-SNP。

目錄

- 專用主機上執行個體的證明
- 共用租用上執行個體的證明

專用主機上執行個體的證明

專用主機使用版本化晶片背書金鑰 (VCEK)，這是一種唯一的每個晶片簽署金鑰，用於簽署證明報告。您必須驗證從 VCEK 憑證到 AMD 信任根的信任鏈。

> **Note**
>
> 目前支援的處理器模型為 `milan`。

先決條件

在已啟用 AMD SEV-SNP 且已安裝 Git、Cargo 和 Perl 的專用主機上執行的執行個體。

步驟 1：建置 snpguest 公用程式

在此步驟中，您會安裝並建置snpguest公用程式，用來產生證明報告、擷取所需的憑證，以及驗證證明報告。

1. 連線到您的執行個體。
2. 驗證是否已安裝先決條件。

```
$ perl --version; cargo --version; git --version
```

3. 執行下列命令以從 snpguest repository 建置 snpguest 公用程式。

```
$ git clone https://github.com/virtee/snpguest.git
$ cd snpguest
$ cargo build -r
$ cd target/release
```

步驟 2：產生證明文件報告

產生要求認證報告的請求。snpguest 公用程式會透過主機向 AMD 安全處理器請求認證報告，並將其寫入二進位檔案。下列範例會建立隨機請求 nonce，並將報告存放在 中report.bin。

```
$ ./snpguest report report.bin request-file.txt --random
```

步驟 3：擷取並驗證 VCEK 憑證鏈

認證報告由 VCEK 簽署，該 VCEK 是您專用主機上 AMD 晶片獨有的。您必須驗證從 VCEK 到 AMD 信任根的信任鏈。

1. 從 AMD Key Distribution Service (KDS) 擷取 VCEK 憑證。憑證由來自證明文件報告的晶片 ID 和 TCB 版本識別。

```
$ ./snpguest fetch vcek pem ./ ./report.bin --processor-model milan
```

2. 擷取構成信任鏈的 AMD 根憑證 (AMD 根金鑰 (ARK) 和 AMD SEV 金鑰 (ASK))。

```
$ ./snpguest fetch ca PEM ./ milan --endorser vcek
```

3. 驗證憑證鏈。

```
$ ./snpguest verify certs ./
```

以下為範例輸出。

```
The AMD ARK was self-signed!
The AMD ASK was signed by the AMD ARK!
The VCEK was signed by the AMD ASK!
```

選用：為了提高透明度，您可以直接從 AMD 下載憑證並使用 來獨立驗證鏈openssl。

```
$ curl --proto '=https' --tlsv1.2 -sSf https://kdsintf.amd.com/vcek/v1/Milan/cert_chain
 -o ./cert_chain.pem
$ openssl verify --CAfile ./cert_chain.pem vcek.pem
```

以下為範例輸出。

```
vcek.pem: OK
```

步驟 4：驗證證明文件報告簽章

確認認證報告已由 VCEK 憑證簽署。這可確認報告是在正版 AMD 硬體上產生，且未遭到竄改。

```
$ ./snpguest verify attestation ./ report.bin
```

以下為範例輸出。

```
Reported TCB Boot Loader from certificate matches the attestation report.
Reported TCB TEE from certificate matches the attestation report.
Reported TCB SNP from certificate matches the attestation report.
Reported TCB Microcode from certificate matches the attestation report.
VEK signed the Attestation Report!
```

TCB（可信運算基礎）版本欄位會確認憑證中的韌體版本符合認證報告中報告的韌體版本。最後一行確認 VCEK 已簽署報告。

共用租用上執行個體的證明

共用租用執行個體使用版本載入背書金鑰 (VLEK)，由 AMD 為其發行 AWS。您必須驗證從 VLEK 憑證到 AMD 信任根的信任鏈。

步驟 1：建置 snpguest 公用程式

在此步驟中，您會安裝並建置snpguest公用程式，用來產生證明報告、擷取所需的憑證，以及驗證證明報告。

1. 連線到您的執行個體。
2. 執行下列命令以從 snpguest repository 建置 snpguest 公用程式。

```
$ git clone https://github.com/virtee/snpguest.git
$ cd snpguest
$ cargo build -r
$ cd target/release
```

步驟 2：產生證明文件報告

產生要求認證報告的請求。snpguest 公用程式會向主機請求認證報告，並將其寫入二進位檔案。下列範例會建立隨機請求 nonce，並將報告存放在 中report.bin。

```
$ ./snpguest report report.bin request-file.txt --random
```

從主機記憶體請求憑證，並將其儲存為 PEM 檔案。

```
$ ./snpguest certificates PEM ./
```

步驟 3：擷取並驗證 VLEK 憑證鏈

證明文件報告由 VLEK 簽署，該 VLEK 由 AMD 發出 AWS。您必須驗證從 VLEK 到 AMD 信任根的信任鏈。

1. 從官方 AMD Key Distribution Service 下載信任憑證的 VLEK 根目錄到目前的目錄。

```
$ sudo curl --proto '=https' --tlsv1.2 -sSf https://kdsintf.amd.com/vlek/v1/Milan/
cert_chain -o ./cert_chain.pem
```

2. 使用 openssl 來驗證 VLEK 憑證是否由 AMD 信任根憑證簽署。

```
$ sudo openssl verify --CAfile ./cert_chain.pem vlek.pem
```

以下為範例輸出。

```
vlek.pem: OK
```

步驟 4：驗證證明文件報告簽章

驗證證明報告已由 VLEK 憑證簽署。這可確認報告是在正版 AMD 硬體上產生，且未遭到竄改。

```
$ ./snpguest verify attestation ./ report.bin
```

以下為範例輸出。

```
Reported TCB Boot Loader from certificate matches the attestation report.
Reported TCB TEE from certificate matches the attestation report.
Reported TCB SNP from certificate matches the attestation report.
Reported TCB Microcode from certificate matches the attestation report.
VEK signed the Attestation Report!
```

Amazon EC2 Linux 執行個體的處理器狀態控制

C-states 可控制核心在閒置狀態要進入的休眠等級。C-state 從 C0 (最淺閒置狀態，此時核心處於喚醒狀態並執行指令) 開始編號，最高可到 C6 (最深閒置狀態，此時核心會關閉)。

P-states 則可控制所需的核心效能 (CPU 頻率)。P-state 從 P0 (最高效能設定，此時核心可視需要使用 Intel Turbo Boost 技術來提高頻率) 開始編號，之後則從 P1 (請求最大基準頻率的 P-state) 到 P15 (最低適用頻率)。

> **Note**
>
> AWS Graviton 處理器具有內建省電模式，並以固定頻率運作。因此，它們不能為作業系統提供控制 C-state 和 P-states 的能力。

C-state 和 P-state

下列執行個體類型可讓作業系統控制 C-state 和 P-state：

- 一般用途： m4.10xlarge | m4.16xlarge
- 運算優化： c4.8xlarge
- 記憶體優化： r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge
- 儲存最佳化：d2.8xlarge | i3.8xlarge | i3.16xlarge | h1.8xlarge | h1.16xlarge
- 加速運算：g3.16xlarge | p3.16xlarge
- 裸機：所有使用 Intel 和 AMD 處理器的裸機執行個體

僅 C-state

下列執行個體類型可讓作業系統控制 C-states：

- 一般用途： m5.12xlarge | m5.24xlarge | m5d.12xlarge | m5d.24xlarge | m5n.12xlarge m5n.24xlarge | m5dn.12xlarge || m5dn.24xlarge m5zn.6xlarge ||||| m5zn.12xlarge m6a.24xlarge m6a.48xlarge m6i.16xlarge m6i.32xlarge m6id.16xlarge m6id.32xlarge m6idn.16xlarge m6in.16xlarge m6in.32xlarge | m7a.medium || m7a.large |||||||| m7a.xlarge m7a.2xlarge m7a.4xlarge m7a.8xlarge m7a.12xlarge m7a.16xlarge m7a.24xlarge m7a.32xlarge m7a.48xlarge m7i.large m7i.xlarge m7i.2xlarge |||||||||||||||||||||||||||||||||||||||||||||||||| ||||||| m7i.4xlarge m7i.8xlarge m7i.12xlarge m7i.16xlarge m7i.24xlarge m7i.48xlarge m8a.medium m8a.large m8a.xlarge m8a.2xlarge m8a.4xlarge m8a.8xlarge m8a.12xlarge m8a.16xlarge m8a.24xlarge m8a.48xlarge m8azn.medium m8azn.large m8azn.xlarge m8azn.3xlarge m8azn.6xlarge m8azn.12xlarge m8azn.24xlarge m8i.large m8i.xlarge m8i.2xlarge m8i.4xlarge m8i.8xlarge m8i.12xlarge m8i.16xlarge m8i.24xlarge m8i.32xlarge m8i.48xlarge m8i.96xlarge m8id.large m8id.xlarge m8id.2xlarge m8id.4xlarge m8id.8xlarge m8id.12xlarge m8id.16xlarge m8id.24xlarge m8id.32xlarge m8id.48xlarge m8id.96xlarge m8in.large m8in.xlarge m8in.2xlarge m8in.4xlarge m8in.8xlarge m8in.12xlarge m8in.16xlarge m8in.24xlarge m8in.32xlarge m8in.48xlarge m8in.96xlarge m8idn.large m8idn.xlarge m8idn.2xlarge m8idn.4xlarge m8idn.8xlarge m8idn.12xlarge m8idn.16xlarge m8idn.24xlarge m8idn.32xlarge m8idn.48xlarge m8idn.96xlarge m8ib.large m8ib.xlarge m8ib.2xlarge m8ib.4xlarge m8ib.8xlarge m8ib.12xlarge m8ib.16xlarge m8ib.24xlarge m8ib.32xlarge m8ib.48xlarge m8ib.96xlarge m8idb.large m8idb.xlarge m8idb.2xlarge m8idb.4xlarge m8idb.8xlarge m8idb.12xlarge m8idb.16xlarge

m8idb.24xlarge m8idb.32xlarge m8idb.48xlarge m8idb.96xlarge m8ine.large m8ine.xlarge m8ine.2xlarge m8ine.4xlarge m8ine.8xlarge m8ine.12xlarge

- 運算最佳化： c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5a.24xlarge | c5ad.24xlarge c5d.9xlarge || c5d.12xlarge c5d.18xlarge ||||| c5d.24xlarge c5n.9xlarge c5n.18xlarge c6a.24xlarge c6a.32xlarge c6a.48xlarge c6i.16xlarge c6i.32xlarge c6id.24xlarge c6id.32xlarge || c6in.32xlarge || c7a.medium |||| ||||| c7a.large c7a.xlarge c7a.2xlarge c7a.4xlarge c7a.8xlarge c7a.12xlarge c7a.16xlarge c7a.24xlarge c7a.32xlarge c7a.48xlarge c7i.large c7i.xlarge ||c8in.xlarge c8in.8xlarge c8in.96xlarge c8ib.xlarge c8ib.24xlarge c8ib.96xlarge c8in.xlarge c8in.8xlarge |||||||||||||||||||||||||||| |||||||||||||||||||||||||||||| c7i.2xlarge c7i.4xlarge c7i.8xlarge c7i.12xlarge c7i.16xlarge c7i.24xlarge c7i.48xlarge c8a.medium c8a.large c8a.xlarge c8a.2xlarge c8a.4xlarge c8a.8xlarge c8a.12xlarge c8a.16xlarge c8a.24xlarge c8a.48xlarge c8i.large c8i.xlarge c8i.2xlarge c8i.4xlarge c8i.8xlarge c8i.12xlarge c8i.16xlarge c8i.24xlarge c8i.32xlarge c8i.48xlarge c8i.96xlarge c8id.large c8id.xlarge c8id.2xlarge c8id.4xlarge c8id.8xlarge c8id.12xlarge c8id.16xlarge c8id.24xlarge c8id.32xlarge c8id.48xlarge c8id.96xlarge c8in.large c8in.2xlarge c8in.4xlarge c8in.12xlarge c8in.16xlarge c8in.24xlarge c8in.32xlarge c8in.48xlarge c8ib.large c8ib.2xlarge c8ib.4xlarge c8ib.8xlarge c8ib.12xlarge c8ib.16xlarge c8ib.32xlarge c8ib.48xlarge c8in.large c8in.2xlarge c8in.4xlarge c8in.12xlarge x8i.large x8i.xlarge x8i.2xlarge x8i.4xlarge x8i.8xlarge x8i.12xlarge x8i.16xlarge x8i.24xlarge x8i.32xlarge x8i.48xlarge x8i.48xlarge x8i.96xlarge
- 記憶體最佳化： r5.12xlarge | r5.24xlarge | r5b.12xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge |||||||||||||| |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| |||||||||||||||| r5dn.24xlarge r6a.24xlarge r6a.48xlarge r6i.16xlarge r6i.32xlarge r6id.16xlarge r6id.32xlarge r6in.16xlarge r6in.32xlarge r7a.medium r7a.large r7a.xlarge r7a.2xlarge r7a.4xlarge r7a.8xlarge r7a.12xlarge r7a.16xlarge r7a.24xlarge r7a.32xlarge r7a.48xlarge r7i.large r7i.xlarge r7i.2xlarge r7i.4xlarge r7i.8xlarge r7i.12xlarge r7i.16xlarge r7i.24xlarge r7i.48xlarge r7iz.large r7iz.xlarge r7iz.2xlarge r7iz.4xlarge r7iz.8xlarge r7iz.12xlarge r7iz.16xlarge r7iz.32xlarge r8a.medium r8a.large r8a.xlarge r8a.2xlarge r8a.4xlarge r8a.8xlarge r8a.12xlarge r8a.16xlarge r8a.24xlarge r8a.48xlarge r8i.large r8i.xlarge

r8i.2xlarge r8i.4xlarge r8i.8xlarge r8i.12xlarge r8i.16xlarge r8i.24xlarge r8i.32xlarge r8i.48xlarge r8i.96xlarge r8id.large r8id.xlarge r8id.2xlarge r8id.4xlarge r8id.8xlarge r8id.12xlarge r8id.16xlarge r8id.24xlarge r8id.32xlarge r8id.48xlarge r8id.96xlarge r8in.large r8in.xlarge r8in.2xlarge r8in.4xlarge r8in.8xlarge r8in.12xlarge r8in.16xlarge r8in.24xlarge r8in.32xlarge r8in.48xlarge r8in.96xlarge r8idn.large r8idn.xlarge r8idn.2xlarge r8idn.4xlarge r8idn.8xlarge r8idn.12xlarge r8idn.16xlarge r8idn.24xlarge r8idn.32xlarge r8idn.48xlarge r8idn.96xlarge r8ib.large r8ib.xlarge r8ib.2xlarge r8ib.4xlarge r8ib.8xlarge r8ib.12xlarge r8ib.16xlarge r8ib.24xlarge r8ib.32xlarge r8ib.48xlarge r8ib.96xlarge r8idb.large r8idb.xlarge r8idb.2xlarge r8idb.4xlarge r8idb.8xlarge r8idb.12xlarge r8idb.16xlarge r8idb.24xlarge r8idb.32xlarge r8idb.48xlarge r8idb.96xlarge u-3tb1.56xlarge u-6tb1.56xlarge u-6tb1.112xlarge u-9tb1.112xlarge u-12tb1.112xlarge u-18tb1.112xlarge u-24tb1.112xlarge u7i-6tb.112xlarge u7i-8tb.112xlarge u7i-12tb.224xlarge u7in-16tb.224xlarge u7in-24tb.224xlarge u7in-32tb.224xlarge u7inh-32tb.480xlarge x2idn.32xlarge x2iedn.16xlarge x2iedn.32xlarge x2iezn.12xlarge x8aedz.large x8aedz.xlarge x8aedz.3xlarge x8aedz.6xlarge x8aedz.12xlarge x8aedz.24xlarge z1d.6xlarge z1d.12xlarge

- 儲存最佳化：d3en.12xlarge | dl1.24xlarge | i3en.12xlarge | i3en.24xlarge | i4i.16xlarge | i7i.large | i7i.xlarge | i7i.2xlarge | i7i.4xlarge | i7i.8xlarge | i7i.12xlarge | i7i.16xlarge | i7i.24xlarge | i7i.48xlarge | i7ie.large | i7ie.xlarge | i7ie.2xlarge | i7ie.3xlarge | i7ie.6xlarge | i7ie.12xlarge | i7ie.18xlarge | i7ie.24xlarge | i7ie.48xlarge | r5b.12xlarge | r5b.24xlarge
- 加速運算： dl1.24xlarge | f2.6xlarge | f2.12xlarge | f2.48xlarge | g5.24xlarge | g5.48xlarge | g6.24xlarge | g6.48xlarge || g6e.12xlarge g6e.24xlarge | g6e.48xlarge g7e.2xlarge | g7e.4xlarge g7e.8xlarge | g7e.12xlarge | g7e.24xlarge | g7e.48xlarge ||| g7.2xlarge | g7.4xlarge || g7.8xlarge | g7.12xlarge | g7.24xlarge | g7.48xlarge | inf1.24xlarge p3dn.24xlarge ||| p4d.24xlarge p4de.24xlarge p5.48xlarge p5e.48xlarge | p5en.48xlarge | p6-b200.48xlarge || p6-b300.48xlarge | trn1.32xlarge trn2.3xlarge | trn2.48xlarge trn2a.3xlarge trn2a.48xlarge trn2n.3xlarge trn2n.48xlarge trn2p.48xlarge trn2u.48xlarge vt1.24xlarge
- 高效能運算： hpc7a.12xlarge | hpc7a.24xlarge | hpc7a.48xlarge | hpc7a.96xlarge || hpc8a.96xlarge

建議您變更 C-state 或 P-state 設定來提高處理器效能穩定性、減少延遲或針對特定工作負載微調執行個體。預設 C-state 和 P-state 設定提供最大效能，適合多數工作負載使用。然而，若您的應用程式更適合犧牲較高的單核心或雙核心頻率以降低延遲，或在較低頻率更能展現穩定效能 (而不適合使用突增 Turbo Boost 頻率)，請考慮嘗試調整這些執行個體的 C-state 或 P-state 設定。

如需不同處理器組態以及如何監控 Amazon Linux 組態效果的資訊，請參閱「Amazon Linux 2 使用者指南」中的 Amazon EC2 Amazon Linux 執行個體的處理器狀態控制。這些程序專為 Amazon Linux 撰寫並使用，然而，亦可能適用其他搭載 Linux 核心 3.9 或更新版本的 Linux 發行版本。如需其他 Linux 發行版本和處理器狀態控制的詳細資訊，請參閱您的系統特定文件。

Amazon EC2 受管執行個體

Amazon EC2 受管執行個體是由指定服務供應商佈建和管理的 EC2 執行個體。受管執行個體可讓您將執行個體的操作控制權委派給服務提供者，藉此提供輕鬆在 Amazon EC2 上執行運算工作負載的方法。。

委派的控制權是針對受管執行個體引進的唯一變更。技術規格和帳單與非受控 EC2 執行個體維持相同。由於受管執行個體可讓您將控制權委派給服務提供者，因此您可以從服務提供者的營運專業知識和最佳實務中受益。在管理執行個體時，服務提供者會負責佈建執行個體、設定軟體、擴展容量、處理執行個體故障和替換，以及終止執行個體等任務。

您無法直接修改受管執行個體的設定或或終止執行個體。服務和特定操作取決於您與服務提供者之間簽訂的協議。不過，您可以從受管執行個體新增、修改或移除標籤，讓您在 AWS 環境中進行分類。

下列 AWS 服務提供受管執行個體：

- EKS 自動模式
- Amazon ECS 受管執行個體
- WorkSpaces Core 受管執行個體
- Lambda 受管執行個體
- Amazon EC2 快速啟動
- Bedrock AgentCore 執行時間執行個體

目錄

- 受管執行個體的帳單
- 識別受管執行個體

- 受管資源可見性設定
- 開始使用受管執行個體

受管執行個體的帳單

Amazon EC2 受控執行個體會產生與非受控 Amazon EC2 執行個體相同的基本費用，以及服務提供者的個別費用。這筆額外費用由管理執行個體的服務提供者收取，且另外計費。其涵蓋了為操作和維護受控執行個體而提供的服務成本。

所有 Amazon EC2 購買選項均適用於受管執行個體，包括隨需執行個體、預留執行個體、Spot 執行個體和 Savings Plans。透過直接從 EC2 取得計算資源並提供給您的服務提供者，即可從套用至帳戶的現有預留執行個體或 Savings Plans 中受益，確保使用了最具成本效益的可用運算容量。

例如，在使用 Amazon EKS 自動模式時，您需要為基礎執行個體支付標準 EC2 執行個體的費率，以及 Amazon EKS 為您管理執行個體的額外費用。如果您隨後決定註冊 Savings Plans，則 Savings Plans 會減少 EC2 執行個體的費率，而 Amazon EKS 的額外費用保持不變。

識別受管執行個體

受管執行個體由受管欄位中的 true 值識別。服務提供者是在運算子欄位 (位於主控台) 或 `Principal` 欄位 (位於 CLI) 中識別。

使用下列程序來識別受管執行個體。

Console

識別受控執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您要檢查的執行個體。
4. 在詳細資訊索引標籤 (如果選取核取方塊) 或摘要區域中 (如果選取執行個體 ID)，尋找受控欄位。
   - true 值表示受控執行個體。
   - false 值表示非受控執行個體。
5. 如果將受控設為 true，運算子欄位會顯示某個值，可識別負責管理執行個體的服務提供者。例如，eks.amazonaws.com 的值會將 Amazon EKS 識別為服務提供者。

AWS CLI

識別受控執行個體

使用 describe-instances 命令並指定執行個體 ID。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].Operator
```

以下為範例輸出。如果 Managed 為 true，則執行個體是受控執行個體並包含 Principal。主體是管理執行個體的服務提供者。例如，eks.amazonaws.com 的值會將 Amazon EKS 識別為服務提供者。

```
[
    {
        "Managed": true,
        "Principal": "eks.amazonaws.com"
    }
]
```

尋找受管執行個體

使用 describe-instances 命令，並使用 true 的值指定 operator.managed 篩選條件。--query 選項只會顯示受管執行個體 ID。

```
aws ec2 describe-instances \
    --filters "Name=operator.managed,Values=true" \
    --query Reservations[*].Instances[].InstanceId
```

PowerShell

識別受控執行個體

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.Operator
```

以下為範例輸出。

```
Managed Principal
```

```
------- ---------
True    eks.amazonaws.com
```

尋找受管執行個體

使用 Get-EC2Instance cmdlet。此範例只會顯示受管執行個體 ID。

```
(Get-EC2Instance -Filter @{Name="operator.managed";
 Values="true"}).Instances.InstanceId
```

受管資源可見性設定

您可以控制 AWS 服務代表您佈建的資源是否出現在 Amazon EC2 主控台檢視和 API 清單操作中。

什麼是受管資源可見性？

AWS 服務，例如 Amazon EKS、Amazon ECS、WorkSpaces Core、 AWS Lambda 和 Amazon Bedrock AgentCore 佈建，並直接在您的帳戶中操作 Amazon EC2 執行個體。這些服務會承擔擴展、作業系統修補程式、安全性更新和生命週期管理的責任。產生的 Amazon EC2 執行個體、Amazon EC2 啟動範本、Amazon EBS 磁碟區、快照和網路介面 (ENIs) 會顯示在 Amazon EC2 主控台和 APIs 中的客戶受管資源旁。受管資源可見性設定可讓您控制這些受管資源是否出現在資源檢視中。

受影響的資源類型

Resource Type (資源類型)	佈建這些資源的服務	說明
Amazon EC2 執行個體	Amazon EKS 工作者節點、Amazon ECS 容器執行個體、 AWS Lambda 執行環境、Amazon WorkSpaces Core、Amazon Bedrock AgentCore	受可見性設定影響的主要資源類型
Amazon EC2 啟動範本	Amazon EKS、Amazon ECS、Amazon WorkSpaces Core、Amazon Bedrock AgentCore	受管服務建立的啟動範本
Amazon EBS 磁碟區	Amazon EKS、Amazon ECS、Amazon WorkSpaces	連接至受管執行個體的磁碟區

Resource Type (資源類型)	佈建這些資源的服務	說明
	Core、Amazon Bedrock AgentCore 、Amazon EC2 Fast Launch	
Amazon EBS 快照	Amazon EC2 快速啟動	Amazon EC2 Fast Launch 管理的快照
網路界面 ENIs)	Amazon EKS、Amazon ECS、 AWS Lambda、Amazon WorkSpaces Core、Amazon Bedrock AgentCore 、Amazon EC2 應用程式狀態檢查	針對受管工作負載佈建的網路介面

> **Note**
>
> 根據預設，Amazon EC2 在可見性設定變成可用之前，會針對沒有受管資源的帳戶隱藏受管資源。對於已有受管資源的帳戶，Amazon EC2 會將可見性設定設為 可見，以保留現有的工作流程。可見性設定適用於區域中的所有受管資源，無論它們何時建立。您可以隨時變更可見性設定。

為什麼要設定可見性設定

設定可見性設定可讓您自訂受管資源在操作工具中的顯示方式。常用案例包括：

- 將合規儀表板中的資源計數減少為僅客戶管理的資源，以簡化控管。
- 減少可觀測性工具中的雜訊，這些工具會彙總帳戶中所有執行個體的 Amazon EC2 指標。
- 防止雲端安全狀態管理 (CSPM) 掃描器 （ 例如 Qualys) 中將受管資源標記為客戶錯誤組態的誤報。
- 透過受管執行個體， AWS 負責 Amazon EC2 執行個體的組態、修補和運作狀態。透過控制可見性，您可以更好地向最終使用者闡明共同的責任模型。

> **Note**
>
> 可見性設定控制 AWS 主控台檢視和 API 清單操作中的資源顯示。它們不會影響帳單、資源操作或實際存取許可。隱藏的資源仍能完全運作並計費。

設定受管資源可見性

您可以使用 Amazon EC2 主控台或 來設定受管資源的可見性 AWS CLI。

Console

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡片的設定下，選擇受管資源。
4. 選擇 Modify (修改)。
5. 在預設可見性下，選擇下列其中一個選項：
   - 隱藏 （ 預設 ） – 隱藏所有受管資源。
   - 可見 – 顯示所有受管資源。
6. 選擇修改可見性。

AWS CLI

取得目前的可見性設定

使用 get-managed-resource-visibility 命令來擷取目前的可見性組態：

```
aws ec2 get-managed-resource-visibility
```

回應範例：

```
{
    "visibility": {
        "defaultVisibility": "hidden"
    }
}
```

隱藏所有受管資源

使用 modify-managed-resource-visibility 命令來隱藏所有受管資源，無論運算子為何：

```
aws ec2 modify-managed-resource-visibility \
    --default-visibility "hidden"
```

探索隱藏的受管資源

關閉可見性時，您仍然可以存取受管資源。下列方法會隨需顯示它們：

1. 服務特定主控台：導覽至個別 AWS 的服務主控台（例如 Amazon EKS 主控台），以檢視為該服務佈建的執行個體。服務主控台提供有關服務在您帳戶中管理的所有資源的完整詳細資訊。
2. 直接 API 查詢：搭配特定`instance-id`參數使用 `describe-instances` API。無論可見性設定為何，具有已知執行個體 IDs直接查詢都會傳回結果。可見性設定只會影響清單和篩選操作。您也可以使用 `describe-instances`搭配 `include-managed-resources` 參數來探索受管執行個體。

> **Note**
>
> 相同的 direct-query-by-ID 行為適用於所有受影響的資源類型。您可以使用 `describe-volumes`、 `describe-launch-templates`和 `describe-network-interfaces`搭配特定資源 IDs 來存取這些類型的隱藏受管資源。

帳單考量

受管資源可見性設定不會影響帳單。隱藏的受管執行個體會繼續出現在帳單資料中，因為它們是在您帳戶中執行、代表您佈建的資源，無論可見性組態為何，都會保持完全計費。

隱藏的資源仍然可見於：

- AWS 帳單
- AWS 成本和用量報告

> **Important**
>
> 受管執行個體會在您的帳戶中佈建，並使用運算資源。從主控台檢視中隱藏它們並不會降低成本。檢閱服務特定的帳單文件（例如 Amazon EKS 定價、Amazon ECS 定價），以取得受管執行個體費用的詳細資訊。

限制

- 可見性設定適用於整個帳戶，並統一地影響所有 IAM 主體。

- 您無法依資源類型或建立資源的服務選擇性地顯示或隱藏受管資源。例如，您無法選擇在隱藏由 Lambda、Amazon ECS 或 Amazon WorkSpaces 建立的受管執行個體時顯示由 Amazon EKS 建立的受管執行個體。

開始使用受管執行個體

若要開始使用受管執行個體，請參閱您偏好的服務供應商文件：

- 《Amazon EKS 使用者指南》中的使用 EKS Auto Mode 自動化叢集基礎設施
- 《Amazon ECS 開發人員指南》中的 Amazon ECS 受管執行個體
- 《WorkSpaces 核心規劃指南》中的部署受管執行個體
- 《Lambda 開發人員指南》中的 Lambda 受管執行個體
- 本指南中的 Amazon EC2 Fast Launch
- 《Amazon Bedrock AgentCore 開發人員指南》中的 Bedrock AgentCore 執行期執行個體 AgentCore

使用巢狀虛擬化在 Amazon EC2 執行個體中執行 Hypervisor

巢狀虛擬化可讓您在虛擬 Amazon EC2 執行個體內執行 Hyper-V 和 KVM 等 Hypervisor。虛擬 EC2 執行個體是非裸機執行個體。此功能透過將處理器層級虛擬化支援新增至虛擬 EC2 執行個體來擴展虛擬化彈性，讓執行個體中執行的 Hypervisor 能夠建立和管理虛擬機器。

當您在開發工作流程中執行 Docker Desktop、適用於 Linux 2 的 Windows 子系統 (WSL2)、Android Studio 模擬器或 QEMU 等開發工具時，巢狀虛擬化可協助您選擇符合特定效能和價格需求的各種標準虛擬 Amazon EC2 執行個體類型。

使用巢狀虛擬化無需額外費用。

目錄

- 運作方式
- 考量事項
- 在啟用巢狀虛擬化的情況下啟動新的執行個體
- 設定現有執行個體以使用巢狀虛擬化
- 尋找支援巢狀虛擬化的執行個體類型

運作方式

虛擬 EC2 執行個體會在具有 Nitro Hypervisor 的實體主機上執行。為了支援巢狀虛擬化，Nitro 系統會將 Intel VT-x 等處理器延伸模組傳遞至執行個體，以利執行巢狀虛擬機器。巢狀虛擬化架構包含三層：實體 AWS 基礎設施和 Nitro Hypervisor (L0)、執行 Hypervisor (L1) 的 EC2 執行個體，以及在該執行個體 (L2) 內建立的一或多個虛擬機器。

考量事項

開始使用巢狀虛擬化之前，請考慮下列事項：

- 支援的執行個體類型 – 下列執行個體類型目前支援巢狀虛擬化：
  - 一般用途：M7i | M7i-flex | M8i | M8id | M8i-flex
  - 運算最佳化：C7i | C7i-flex | C8i | C8id | C8i-flex
  - 記憶體最佳化：R7i | R7iz | R8i | R8id | R8i-flex | X8i
  - 儲存最佳化：I7i | I7ie
- 支援的 Hypervisor – 目前，KVM 和 Hyper-V 是支援的 L1 Hypervisor。
- Windows 執行個體 – 在 Windows 執行個體上啟用巢狀虛擬化時：
  - Credential Guard – 虛擬安全模式 (VSM) 會自動停用。
  - 休眠 – 不支援執行個體休眠和繼續。
  - CPU 限制 – 不支援超過 192 個 CPUs Windows 執行個體，例如 `m8i.96xl`。
- 安全責任 – 在 EC2 執行個體上使用巢狀虛擬化時， AWS 負責保護基礎基礎設施的「雲端安全性」，並維護 AWS Nitro 系統提供的 EC2 執行個體之間的強隔離界限。客戶負責「雲端安全」，其中包括保護 EC2 執行個體內的作業系統、Hypervisor、巢狀虛擬機器、訪客作業系統、應用程式和資料。
- 效能 – AWS 建議想要執行需要存取硬體虛擬化延伸模組的工作負載，且對效能敏感或具有嚴格延遲要求的客戶評估裸機執行個體。

在啟用巢狀虛擬化的情況下啟動新的執行個體

當您啟動新的執行個體時，您可以開啟巢狀虛擬化，在其上執行 Hypervisor 和虛擬機器。

Console

在執行個體啟動期間啟用巢狀虛擬化

1. 遵照 the section called “使用啟動執行個體精靈進行啟動” 程序並視需要設定執行個體。
2. 確定已選取支援的執行個體類型。
3. 展開進階詳細資訊，針對巢狀虛擬化，選擇啟用。
4. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。

AWS CLI

在啟用巢狀虛擬化的情況下啟動執行個體

使用 run-instances 命令。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r8i.4xlarge \
    --cpu-options "NestedVirtualization=enabled" \
    --key-name my-key-pair
```

PowerShell

在啟用巢狀虛擬化的情況下啟動執行個體

使用 New-EC2Instance 命令。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType r8i.4xlarge `
    -CpuOption @{NestedVirtualization='enabled'} `
    -KeyName my-key-pair
```

設定現有執行個體以使用巢狀虛擬化

您可以在現有的 Amazon EC2 執行個體上開啟巢狀虛擬化。

先決條件

- 執行個體必須處於 stopped 狀態。
- 執行個體類型必須支援巢狀虛擬化。

Console

在現有執行個體上啟用巢狀虛擬化

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 從執行個體資料表中選取您要修改的執行個體。
4. 選擇動作、執行個體設定、變更 CPU 選項。
5. 在變更 CPU 選項頁面上，針對巢狀虛擬化，選擇下列其中一個選項：
   - 啟用 – 開啟執行個體的巢狀虛擬化
   - 停用 – 關閉執行個體的巢狀虛擬化
6. 檢閱您的變更，然後選擇變更以套用新的 CPU 選項。

AWS CLI

在現有執行個體上啟用巢狀虛擬化

首先停止執行個體，然後使用 modify-instance-cpu-options 命令。

```
aws ec2 modify-instance-cpu-options \
    --instance-id i-1234567890abcdef0 \
    --core-count 4 \
    --threads-per-core 2 \
    --nested-virtualization enabled
```

PowerShell

在現有執行個體上啟用巢狀虛擬化

首先停止執行個體，然後使用 Edit-EC2InstanceCpuOption 命令。

```
Edit-EC2InstanceCpuOption `
```

```
    -InstanceId i-1234567890abcdef0 `
    -CoreCount 4 `
    -ThreadsPerCore 2 `
    -NestedVirtualization enabled
```

尋找支援巢狀虛擬化的執行個體類型

您可以找到哪些 Amazon EC2 執行個體類型支援巢狀虛擬化。您可以查詢整個 AWS 區域 或檢查特定執行個體類型。

AWS CLI

列出支援區域中巢狀虛擬化的所有執行個體系列

使用 describe-instance-types 命令，並將processor-info.supported-features篩選條件設定為 的值nested-virtualization：

```
aws ec2 describe-instance-types --region us-east-1 \
  --filters "Name=processor-info.supported-features,Values=nested-virtualization" \
  --query "InstanceTypes[].InstanceType" --output text \
  | tr '\t' '\n' | cut -d. -f1 | sort -u
```

檢查特定執行個體類型是否支援巢狀虛擬化

使用 describe-instance-types 命令並查詢指定執行個體類型的ProcessorInfo.SupportedFeatures欄位：

```
aws ec2 describe-instance-types \
  --instance-types m8i.2xlarge \
  --query "InstanceTypes[].ProcessorInfo.SupportedFeatures"
```

如果執行個體類型支援巢狀虛擬化，則輸出會包含 nested-virtualization。如果沒有，命令會傳回空陣列 ([])。

PowerShell

列出支援區域中巢狀虛擬化的所有執行個體系列

使用 Get-EC2InstanceType 命令，並將processor-info.supported-features篩選條件設定為 的值nested-virtualization：

```
Get-EC2InstanceType -Region us-east-1 -Filter @{ Name='processor-info.supported-
features'; Values='nested-virtualization' } |
  ForEach-Object { ($_.InstanceType -split '\.')[0] } |
  Sort-Object -Unique
```

檢查特定執行個體類型是否支援巢狀虛擬化

使用 Get-EC2InstanceType 命令並查詢指定執行個體類型的 `ProcessorInfo.SupportedFeatures` 屬性：

```
(Get-EC2InstanceType -InstanceType m8i.2xlarge).ProcessorInfo.SupportedFeatures
```

如果執行個體類型支援巢狀虛擬化，則輸出會包含 `nested-virtualization`。如果沒有，命令不會傳回任何輸出。

Amazon EC2 帳單與購買選項

您可以使用下列選項來最佳化 Amazon EC2 的成本：

- 隨需執行個體 – 對於啟動的執行個體，按秒數付費。
- Savings Plans – 對一致的使用量設定綁約來降低 Amazon EC2 成本，每小時以美金計價，為期一年或三年。
- 預留執行個體 – 透過設定綁約至一致的執行個體設定來降低 Amazon EC2 的成本，包含執行個體類型和區域，為期一年或三年。
- Spot 執行個體 – 請求未使用的 EC2 執行個體，可大幅降低您的 Amazon EC2 成本。
- 專用主機 – 付費使用專供您執行執行個體的實體主機，並使用您現有的每個通訊端、每個核心或每個 VM 軟體的許可，以降低成本。
- 專用執行個體 – 依時數為單一租戶硬體上執行的執行個體付費。
- 容量預留：在特定的可用區域中，為 EC2 執行個體預留容量。

如果您無法對特定執行個體組態進行綁約，但可以對使用量進行綁約，請購買 Savings Plans 以降低隨需執行個體成本。如果您需要容量預留，請針對特定可用區域購買預留執行個體或容量預留。容量區塊可用於保留 GPU 執行個體叢集。如能彈性決定應用程式執行的時機以及應用程式是否可以中斷，則 Spot 執行個體會是一項具成本效益的選擇。專用執行個體或專用執行個體可協助您處理合規性要求，並使用您現有的伺服器軟體授權來減少成本。

如需詳細資訊，請參閱 Amazon EC2 定價 和 Amazon EC2 受管執行個體。

購買 Amazon EC2 的隨需執行個體

透過隨需執行個體，您只需要按秒數支付運算容量開銷，無需簽訂長期合約。您可以完整控制執行個體的生命週期 — 您決定何時啟動、停止、休眠、開始、重新開機或終止它。

購買隨需執行個體時，無需簽訂長期合約。您僅需按隨需執行個體處於 `running` 狀態的秒數付費，最少 60 秒。執行中隨需執行個體的每秒小時價格是固定的，列在 Amazon EC2 定價、隨需定價頁面上。

我們建議您將隨需執行個體用於具有短期、不定期但無法中斷之工作負載的應用程式。

若要大幅節省隨需執行個體，請使用 AWS Savings Plans、Spot 執行個體 或 Amazon EC2 的預留執行個體概觀。

內容

- 隨需執行個體配額
  - 監控隨需執行個體配額和用量
  - 請求提高配額
- 查詢隨需執行個體的價格

隨需執行個體配額

AWS 帳戶 每個區域每個 執行中的隨需執行個體數量有配額。隨需執行個體配額是依照執行中隨需執行個體正在使用的虛擬中央處理單元 (vCPU) 的數量來進行管理，與執行個體類型無關。每個配額類型可指定一個或多個執行個體系列的 vCPU 數量上限。

您的帳戶包括下列隨需執行個體的配額。對於處於待處理、停止中、已停止和休眠狀態的執行個體，不會計入您的隨需執行個體配額。容量保留則會計入隨需執行個體的配額，即使未使用也一樣。

名稱	預設	可調整
執行中隨需 DL 執行個體	0	是
執行中隨需 F 執行個體	0	是
執行中隨需 G 和 VT 執行個體	0	是

名稱	預設	可調整
執行中隨需 HPC 執行個體	0	是
執行中隨需記憶體密集型執行個體	0	是
執行中隨需 Inf 執行個體	0	是
執行中隨需 P 執行個體	0	是
執行中隨需的標準 (A、C、D、H、I、M、R、T、Z) 執行個體	5	是
執行中隨需 Trn 執行個體	0	是
執行中隨需 X 執行個體	0	是

如需不同執行個體系列、產生和大小的相關資訊，請參閱 Amazon EC2 執行個體類型指南。

只要 vCPU 的數量不超過您的帳戶配額，就可以啟動滿足不斷變化之應用程式需求的任意執行個體類型組合。例如，在 256 個 vCPU 的標準執行個體配額下，您可以啟動 32 個 `m5.2xlarge` 執行個體 (32 x 8 個 vCPU) 或 16 個 `c5.4xlarge` 執行個體 (16 x 16 個 vCPU)。如需詳細資訊，請參閱EC2 隨需執行個體限制。

任務

- 監控隨需執行個體配額和用量
- 請求提高配額

監控隨需執行個體配額和用量

可以使用下列方法檢視和管理每個區域的隨需執行個體配額。

使用 Service Quotas 主控台檢視目前的配額

1. 在 https：//https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/ 開啟 Service Quotas 主控台。
2. 從導覽列上，選取一個區域。
3. 在篩選條件欄位中，輸入 **`On-Demand`**。

4. 套用的配額值欄會顯示帳戶中每個隨需執行個體配額類型的 vCPU 數目上限。

使用 AWS Trusted Advisor 主控台檢視您目前的配額

在 AWS Trusted Advisor 主控台中開啟服務限制頁面。

設定 CloudWatch 警示

透過 Amazon CloudWatch 指標整合，您可以根據配額來監控 EC2 用量。您也可以設定警示，在接近配額時提出警告。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的 Service Quotas 與 Amazon CloudWatch 警示。

請求提高配額

即使 Amazon EC2 根據使用量自動提高隨需執行個體配額，您也可以在必要時要求提高配額。例如，如果您打算啟動的執行個體超過目前配額允許的數量，則可以使用或 Amazon EC2 服務配額 中所述的 Service Quotas 主控台來請求提高配額。

查詢隨需執行個體的價格

您可以使用 Price List Service API 或 AWS Price List API 來查詢隨需執行個體的價格。如需詳細資訊，請參閱AWS Billing 《 使用者指南》中的使用 AWS 價目表 API。

Amazon EC2 的預留執行個體概觀

> **Important**
>
> 建議選擇 Savings Plans,而非預留執行個體。Savings Plans 是節省 AWS 運算成本最簡單且最靈活的方式，並提供較低的價格（高達隨需定價 72% 的折扣），就像預留執行個體一樣。不過，Savings Plans 與預留執行個體不同。使用預留執行個體時，您可以對特定執行個體組態做出承諾，而使用 Savings Plans 時，您可以靈活地使用最符合您需求的執行個體組態。若要使用 Savings Plans，您可以對一致的使用量設定綁約，每小時以美金計價。如需詳細資訊，請參閱 AWS Savings Plans 使用者指南。

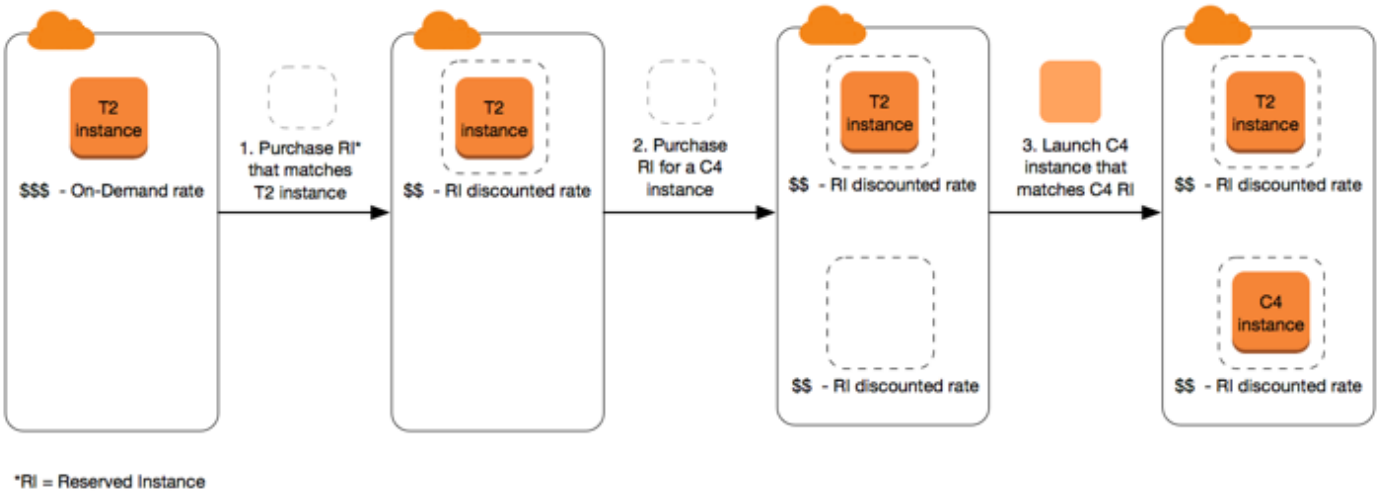
相較於隨需資料庫執行個體的定價，預留執行個體可大幅節省 Amazon EC2 成本。預留執行個體並非實體執行個體，而是一種套用到您帳戶中隨需執行個體用量的計費折扣。這些隨需執行個體必須符合特定屬性，例如執行個體類型和區域，才能受益於帳單折扣。

預留執行個體主題

- 預留執行個體範例案例
- 決定預留執行個體定價的重要變數
- 地區和區域預留執行個體 (範圍)
- 預留執行個體類型 (優惠類別)
- 預留執行個體折扣的套用方式
- 使用您的 預留執行個體
- 預留執行個體的計費方式
- 購買 Amazon EC2 預留執行個體
- 銷售 Amazon EC2 可轉換預留執行個體不能在預留執行個體 Marketplace 中銷售
- 修改 預留執行個體
- 交換 可轉換預留執行個體
- 預留執行個體配額

預留執行個體範例案例

下圖顯示購買及使用預留執行個體的基本案例。



在此案例中，您的帳戶中會有一個執行中且目前以隨需費率付款的隨需執行個體 (T2)。您購買符合您執行中執行個體屬性的預留執行個體，即會立即套用計費利益。接下來，您會為 C4 執行個體購買預留執行個體。您的帳戶中沒有任何符合此預留執行個體屬性的執行中執行個體。在最終步驟中，您會啟動符合 C4 預留執行個體屬性的執行個體，並會立即套用計費利益。

決定預留執行個體定價的重要變數

預留執行個體定價取決於以下重要變數。

執行個體屬性

預留執行個體有四個執行個體屬性可決定其價格。

- 執行個體類型：例如，`m4.large`。這是由執行個體系列 (例如 `m4`) 和執行個體大小 (例如 `large`) 所組成。
- 區域：購買 預留執行個體 的區域。
- 租用：您的執行個體是在共享 (預設) 或是單一租戶 (專用) 硬體上執行。如需詳細資訊，請參閱Amazon EC2 專用執行個體。
- 平台：作業系統，例如 Windows 或 Linux/Unix。如需詳細資訊，請參閱選擇平台。

期限承諾

您可以購買一年或三年承諾的預留執行個體，三年承諾提供更大折扣。

- 一年：一年定義為 31536000 秒 (365 天)。
- 三年：三年定義為 94608000 秒 (1095 天)。

預留執行個體不會自動續訂。當它們到期時，您可以繼續使用 EC2 執行個體而無須中斷，但您將需要支付隨需費率的費用。在以上的範例中，當涵蓋 T2 和 C4 執行個體的預留執行個體到期時，您會重新開始支付隨需費率，直到您終止執行個體或購買符合執行個體屬性的新預留執行個體。

> **Important**
>
> 在您購買預留執行個體之後，您便無法取消購買。但是，若您需要變更，您可以修改、交換或販售您的預留執行個體。

付款選項

預留執行個體有下列付款選項：

- 全預付款：期間開始時便支付完整的款項，並在期間的剩餘部分不會產生其他成本或額外的每小時費用，無論使用多少小時。

- 部分預付款：必須預先支付其中一部分的成本，期間內剩餘的小時數則會以折扣後的每小時費率計費，無論是否有使用預留執行個體。
- 無預付款：您在期間中的每個小時都會以折扣後的每小時費率計費，無論是否有使用預留執行個體。不需要預付款。

> **Note**
>
> 無預付款預留執行個體是以合約義務為基礎，針對保留的完整期間每月付款。基於此原因，在您購買無預付款預留執行個體前，將需要成功的帳單歷史記錄。

一般來說，為預留執行個體支付較高的預付款可以節省更多費用。您也可以在預留執行個體 Marketplace 上尋找由第三方提供、價格較低且有效期間長度較短的預留執行個體。如需詳細資訊，請參閱銷售 Amazon EC2 可轉換預留執行個體不能在預留執行個體 Marketplace 中銷售。

方案類別

若您的運算需求變更，根據優惠類別，您可能可以修改或交換您的 預留執行個體。

- 標準：這些提供更多的折扣，但只能修改。標準版 預留執行個體 無法交換。
- 可轉換：這些提供比標準預留執行個體更少的折扣，但可以與具有不同執行個體屬性的另一個可轉換預留執行個體交換。可轉換預留執行個體亦可修改。

如需詳細資訊，請參閱預留執行個體類型 (優惠類別)。

> **Important**
>
> 在您購買預留執行個體之後，您便無法取消購買。但是，若您需要變更，您可以修改、交換或販售您的預留執行個體。

如需詳細資訊，請參閱 Amazon EC2 預留執行個體定價頁面。

地區和區域預留執行個體 (範圍)

您購買預留執行個體時會決定預留執行個體的範圍。範圍是地區或區域。

- 地區：當您為某一地區購買預留執行個體時，它便稱為地區預留執行個體。

- 區域：當您為某一特定可用區域購買預留執行個體時，它便稱為區域預留執行個體。

範圍不影響價格。您為區域或區域 預留執行個體 支付相同的價格。如需 預留執行個體 定價的詳細資訊，請參閱 決定預留執行個體定價的重要變數 和 Amazon EC2 預留執行個體定價。

如需有關如何指定預留執行個體範圍的詳細資訊，請參閱 RI 屬性，特別是可用區域項目符號。

地區與區域預留執行個體之間的差異

下表特別列出地區預留執行個體與區域預留執行個體之間的一些主要差異：

	地區預留執行個體	區域預留執行個體
能夠預留容量	區域預留執行個體不會預留容量。	指定可用區域中的區域預留執行個體會預留容量。
可用區域彈性	預留執行個體折扣會套用到指定區域內任何可用區域中的執行個體用量。	無可用區域彈性 – 預留執行個體折扣僅套用到指定可用區域中的執行個體用量。
執行個體大小彈性	預留執行個體折扣會套用到執行個體系列中的執行個體用量，而無論大小。 只在使用預設租用的 Amazon Linux/Unix 預留執行個體上才支援。如需詳細資訊，請參閱由正規化因素決定的執行個體大小彈性。	無執行個體大小彈性 - 預留執行個體折扣僅適用於指定執行個體系列和執行個體大小的執行個體用量。
將購買項目放入佇列	您可以將區域預留執行個體的購買項目放入佇列。	您無法將區域預留執行個體的購買項目放入佇列。

如需詳細資訊和範例，請參閱 預留執行個體折扣的套用方式。

預留執行個體類型 (優惠類別)

預留執行個體 的產品類別為標準版或可轉換版。標準版的 預留執行個體 提供的折扣比可轉換版的 預留執行個體 更多，但您無法交換標準版的 預留執行個體。您可以交換可轉換版的 預留執行個體。您可以修改標準版和可轉換版的 預留執行個體。

預留執行個體 的組態會在期間內包含單一執行個體類型、平台、範圍和租用。若您的運算需求變更，您可能可以修改或交換您的 預留執行個體。

標準版與可轉換版 預留執行個體 的差異

以下為標準版和可轉換版 預留執行個體 的不同。

	標準預留執行個體	Convertible Reserved Instance
修改 預留執行個體	可以修改某些屬性。如需詳細資訊，請參閱修改 預留執行個體。	可以修改某些屬性。如需詳細資訊，請參閱 修改 預留執行個體。
交換預留執行個體	無法交換。	可使用新的屬性 (包含執行個體系列、執行個體類型、平台、範圍或租用) 在另一個 可轉換預留執行個體 的期間內交換。如需詳細資訊，請參閱 交換 可轉換預留執行個體。
在預留執行個體 Marketplace 銷售	可在預留執行個體 Marketplace 销售。	不可在預留執行個體 Marketplace 销售。
在預留執行個體 Marketplace 購買	可在預留執行個體 Marketplace 購買。	不可在預留執行個體 Marketplace 購買。

預留執行個體折扣的套用方式

預留執行個體並非實體執行個體，而是一種套用到您帳戶中執行中隨需執行個體的計費折扣。隨需執行個體必須符合預留執行個體的特定規格，才能受益於帳單折扣。

如果您購買預留執行個體並且已有符合預留執行個體規格的執行中預留執行個體，即會立即並自動套用計費折扣。您不需要重新啟動您的執行個體。如果您沒有符合資格的執行中隨需執行個體，請啟動具有與預留執行個體相同規格的隨需執行個體。如需詳細資訊，請參閱 使用您的 預留執行個體。

預留執行個體的方案類別 (標準或可轉換) 不會影響計費折扣的套用方式。

主題

- 區域預留執行個體的套用方式
- 區域預留執行個體的套用方式
- 執行個體大小彈性
- 套用預留執行個體的範例

區域預留執行個體的套用方式

為了在特定可用區域預留容量而購買的預留執行個體稱為區域預留執行個體。

- 預留執行個體折扣會套用到該可用區域中相符的執行個體用量。
- 執行中執行個體的屬性（租用、平台、可用區域、執行個體系列和執行個體大小）必須符合預留執行個體的屬性。

例如，如果您為可用區域 us-east-1a 購買兩個 `c4.xlarge` 預設租用 Linux/Unix 標準預留執行個體，則最多能有兩個在可用區域 us-east-1a 中執行的 `c4.xlarge` 預設租用 Linux/Unix 執行個體獲得預留執行個體折扣帶來的利益。

區域預留執行個體的套用方式

為區域購買的預留執行個體稱為區域預留執行個體，並會提供可用區域和執行個體大小彈性。

- 預留執行個體折扣會套用到該區域內任何可用區域中的執行個體用量。
- 預留執行個體折扣會套用到執行個體系列中的執行個體用量，而無論大小—這稱為執行個體大小彈性。

執行個體大小彈性

由於執行個體大小彈性的因素，執行個體專屬折扣適用於具相同系列的執行個體使用。根據正規化因素，預留執行個體先套用至執行個體系列中最小的執行個體，接著逐步套用至最大的執行個體。有關如何套用預留執行個體折扣的範例，請參閱 案例 2：使用正規化因素的單個帳戶中的預留執行個體。

限制

- 支援：僅區域預留執行個體支援執行個體大小彈性。
- 不支援：下列預留執行個體不支援執行個體大小彈性：
  - 為特定可用區域購買的預留執行個體 (區域預留執行個體)
  - G4ad、G4dn、G5、G5g、G6、G6e、G6f、Gr6、Gr6f、hpc7a、P5、Inf1、Inf2、u7i-6tb 和 u7i-8tb 執行個體的預留執行個體
  - 預留執行個體 適用於 Windows Server、Windows Server with SQL Standard、Windows Server with SQL Server Enterprise、Windows Server with SQL Server Web、RHEL 和 SUSE Linux Enterprise Server
  - 搭配專用租用的預留執行個體

由正規化因素決定的執行個體大小彈性

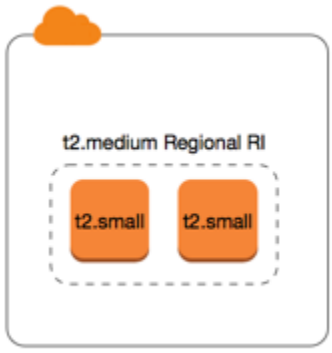
執行個體大小彈性取決於執行個體大小的正規化因素。在區域內的任何可用區域中，折扣可完整或部分套用到相同執行個體系列的執行中執行個體，而這取決於保留的執行個體大小而定。只要執行個體系列、租用和平台三個屬性符合即可。

下表列出執行個體系列中不同的大小，以及其對應的正規化因素。此規模會用於將預留執行個體折扣費率套用到執行個體系列的標準化用量。

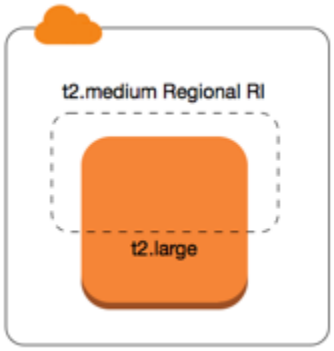
執行個體大小	正規化因素
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24

執行個體大小	正規化因素
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
96xlarge	768
112xlarge	896

舉例而言，`t2.medium` 執行個體的正規化因素為 2。若您在US East (N. Virginia) 中購買 `t2.medium` 預設租用 Amazon Linux/Unix 預留執行個體，且在該區域中您的帳戶內有兩個執行中的 `t2.small` 執行個體，計費利益便會完整套用到兩個執行個體。



或者，若您在US East (N. Virginia) 區域中的帳戶有一個執行中的 `t2.large` 執行個體，計費利益便會套用到執行個體的 50% 用量。



正規化因數也會在修改預留執行個體時套用。如需詳細資訊，請參閱修改 預留執行個體。

裸機執行個體的正規化因素

執行個體大小彈性也套用到執行個體系列中的裸機執行個體。如果您在裸機執行個體上有搭配共用租用的地區 Amazon Linux/Unix 預留執行個體，則可享受在相同執行個體系列中節省預留執行個體。反之亦然：如果您在與裸機執行個體相同的系列中有搭配共用租用的地區 Amazon Linux/Unix 預留執行個體，則可享受在裸機執行個體系列中節省預留執行個體。

`metal` 執行個體大小沒有單一正規化因素。裸機執行個體的正規化因素與相同執行個體系列中的對等虛擬化執行個體大小相同。例如，`i3.metal` 執行個體具有與 `i3.16xlarge` 執行個體相同的正規化因素。

執行個體大小	正規化因素
`a1.metal`	32
`m5zn.metal` | `x2iezn.metal` `z1d.metal`	96

執行個體大小	正規化因素
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

例如，i3.metal 執行個體的正規化因素為 128。如果您在US East (N. Virginia) 購買 i3.metal 預設租用 Amazon Linux/Unix 預留執行個體，則計費利益可套用如下：

- 如果您在該區域中的帳戶有一個執行中的 i3.16xlarge，計費利益會完全套用到 i3.16xlarge 執行個體 (i3.16xlarge 正規化因素 = 128)。
- 或者，如果您在該區域中的帳戶有兩個執行中的 i3.8xlarge 執行個體，計費利益會完全套用到這兩個 i3.8xlarge 執行個體 (i3.8xlarge 正規化因素 = 64)。
- 或者，如果您在該區域中的帳戶有四個執行中的 i3.4xlarge 執行個體，計費利益會完全套用到所有這四個 i3.4xlarge 執行個體 (i3.4xlarge 正規化因素 = 32)。

反之亦然。例如，若您在US East (N. Virginia) 購買兩個 i3.8xlarge 預設租用 Amazon Linux/Unix 預留執行個體，而且您在該區域中有一個執行中的 i3.metal 執行個體，計費利益會完全套用到 i3.metal 執行個體。

套用預留執行個體的範例

下列案例涵蓋套用預留執行個體的方式。

- 案例 1：單一帳戶中的預留執行個體
- 案例 2：使用正規化因素的單個帳戶中的預留執行個體

- 案例 3：連結帳戶中的區域預留執行個體
- 案例 4：連結帳戶中的區域預留執行個體

案例 1：單一帳戶中的預留執行個體

您正在帳戶 A 中執行下列隨需執行個體：

- 4 個 m3.large Linux，位於 us-east-1a 可用區域中的預設租用執行個體
- 2 個 m4.xlarge Amazon Linux，位於 us-east-1b 可用區域中的預設租用執行個體
- 1 個 c4.xlarge Amazon Linux，位於 us-east-1c 可用區域中的預設租用執行個體

您在帳戶 A 中購買下列預留執行個體：

- 4 個 m3.large Linux，位於 us-east-1a 可用區域中的預設租用預留執行個體 (容量已預留)
- 4 個 m4.large Amazon Linux，位於 us-east-1 區域中的預設租用預留執行個體
- 1 個 c4.large Amazon Linux，位於 us-east-1 區域中的預設租用預留執行個體

預留執行個體利益會以下列方式套用：

- 四個 m3.large 執行個體會使用四個 m3.large 區域預留執行個體的折扣和容量保留，因為它們之間的屬性 (執行個體大小、區域、平台、租用) 相符。
- m4.large 區域預留執行個體可提供可用區域和執行個體大小彈性，因為它們是使用預設租用的區域 Amazon Linux 預留執行個體。

  m4.large 相當於 4 個標準化單位/小時。

  您已購買四個 m4.large 區域預留執行個體，總共等於 16 個標準化單位/小時 (4x4)。帳戶 A 擁有兩個執行中的 m4.xlarge 執行個體，相當於 16 個標準化單位/小時 (2x8)。在此案例中，四個 m4.large 區域預留執行個體可為兩個 m4.xlarge 執行個體提供完整用量的計費利益。
- 位於 us-east-1 中的 c4.large 區域預留執行個體可提供可用區域和執行個體大小彈性，因為它是使用預設租用的區域 Amazon Linux 預留執行個體，並會套用到 c4.xlarge 執行個體。c4.large 執行個體相當於 4 個標準化單位/小時，c4.xlarge 則相當於 8 個標準化單位/小時。

  在此案例中，c4.large 區域預留執行個體會為 c4.xlarge 用量提供部分利益。這是因為 c4.large 預留執行個體相當於 4 個標準化單位/小時用量，但 c4.xlarge 執行個體需要 8 個標準

化單位/小時。因此，c4.large 預留執行個體計費折扣會套用到 50% 的 c4.xlarge 用量。剩餘的 c4.xlarge 用量會以隨需費率收費。

案例 2：使用正規化因素的單個帳戶中的預留執行個體

您正在帳戶 A 中執行下列隨需執行個體：

- 2 個 m3.xlarge Amazon Linux，位於可用區域 us-east-1a 中的預設租用執行個體
- 2 個 m3.large Amazon Linux，位於 us-east-1b 可用區域中的預設租用執行個體

您在帳戶 A 中購買下列預留執行個體：

- 1 個 m3.2xlarge Amazon Linux，位於 us-east-1 區域中的預設租用預留執行個體

預留執行個體利益會以下列方式套用：

- 位於 us-east-1 中的 m3.2xlarge 區域預留執行個體可提供可用區域和執行個體大小彈性，因為它是使用預設租用的區域 Amazon Linux 預留執行個體。它首先套用至 m3.large 執行個體，然後套用至 m3.xlarge 執行個體，因為它會根據正規化因素，先套用至執行個體系列中最小的執行個體，接著逐步套用至最大的執行個體。

  m3.large 執行個體相當於 4 個標準化單位/小時。

  m3.xlarge 執行個體相當於 8 個標準化單位/小時。

  m3.2xlarge 執行個體相當於 16 個標準化單位/小時。

  此利益適用於如下情況：

  m3.2xlarge 區域預留執行個體為 2 x m3.large 用量提供全部利益，因為這些執行個體共佔用 8 個標準化單位/小時。這將留下 8 個標準化單位/小時來套用至 m3.xlarge 執行個體。

  藉助剩餘的 8 個標準化單位/小時，m3.2xlarge 區域預留執行個體可為 1 x m3.xlarge 用量提供全部利益，因為每個 m3.xlarge 執行個體相當於 8 個標準化單位/小時。剩餘的 m3.xlarge 用量會以隨需費率收費。

案例 3：連結帳戶中的區域預留執行個體

預留執行個體會先套用到購買帳戶中的用量，之後則是組織內任何其他帳戶內的合格用量。如需詳細資訊，請參閱預留執行個體 與合併帳單。針對提供執行個體大小彈性的區域預留執行個體，利益會先套用至執行個體系列中最小的執行個體，接著逐步套用至最大的執行個體。

您正在帳戶 A (購買帳戶) 中執行下列隨需執行個體：

- 2 個 m4.xlarge Linux，位於 us-east-1a 可用區域中的預設租用執行個體
- 1 個 m4.2xlarge Linux，位於 us-east-1b 可用區域中的預設租用執行個體
- 2 個 c4.xlarge Linux，位於 us-east-1a 可用區域中的預設租用執行個體
- 1 個 c4.2xlarge Linux，位於 us-east-1b 可用區域中的預設租用執行個體

另一個客戶正在帳戶 B – 一個連結帳戶中執行下列隨需執行個體：

- 2 個 m4.xlarge Linux，位於 us-east-1a 可用區域中的預設租用執行個體

您在帳戶 A 中購買下列區域預留執行個體：

- 4 個 m4.xlarge Linux，位於 us-east-1 區域中的預設租用預留執行個體
- 2 個 c4.xlarge Linux，位於 us-east-1 區域中的預設租用預留執行個體

地區預留執行個體利益會以下列方式套用：

- 帳戶 A (購買帳戶) 中的兩個 m4.xlarge 執行個體及單一 m4.2xlarge 執行個體會使用四個 m4.xlarge 預留執行個體的折扣。三個執行個體全部符合屬性 (執行個體系列、區域、平台、租用)。扣折會先套用到購買帳戶 (帳戶 A) 中的執行個體，即使帳戶 B (連結帳戶) 有兩個也符合預留執行個體的 m4.xlarge 也一樣。沒有容量保留，因為預留執行個體是地區預留執行個體。
- 兩個 c4.xlarge 預留執行個體的折扣會套用至兩個 c4.xlarge 執行個體，因為這兩個執行個體的大小小於 c4.2xlarge 執行個體。沒有容量保留，因為預留執行個體是地區預留執行個體。

案例 4：連結帳戶中的區域預留執行個體

一般而言，某帳戶擁有的預留執行個體會先套用到該帳戶中的用量。但是，若組織中其他帳戶內有合格且未使用的特定可用區域預留執行個體 (區域性預留執行個體)，便會在套用該帳戶擁有的區域預留執行個體前套用到帳戶。這是為了確保最高的預留執行個體使用率和較低的帳單。基於計費目的，組織中的所有帳戶都會視為一個帳戶處理。以下範例可協助說明。

您正在帳戶 A (購買帳戶) 中執行下列隨需執行個體：

- 1 個位於 us-east-1a 可用區域中的 m4.xlarge Linux 預設租用執行個體

一名客戶正在連結帳戶 B 中執行下列隨需執行個體：

- 1 個位於 us-east-1b 可用區域中的 m4.xlarge Linux 預設租用執行個體

您在帳戶 A 中購買下列區域預留執行個體：

- 1 個 m4.xlarge Linux，位於 us-east-1 區域中的預設租用預留執行個體

一名客戶也在連結帳戶 C 中購買下列區域預留執行個體：

- 1 個 m4.xlarge Linux，位於 us-east-1a 可用區域中的預設租用預留執行個體

預留執行個體利益會以下列方式套用：

- 帳戶 C 擁有的 m4.xlarge 區域性預留執行個體折扣會套用到帳戶 A 中的 m4.xlarge 用量。
- 帳戶 A 擁有的 m4.xlarge 區域預留執行個體折扣會套用到帳戶 B 中的 m4.xlarge 用量。
- 若帳戶 A 擁有的區域預留執行個體先套用到帳戶 A 的用量，則帳戶 C 擁有的區域性預留執行個體便會處於未使用的狀態，且帳戶 B 的用量會以隨需費率收費。

如需詳細資訊，請參閱 AWS Cost and Usage Report中的了解您的預留執行個體。

> **Note**
>
> 區域預留執行個體預留容量僅適用於擁有的帳戶，無法與其他 AWS 帳戶共用。如果您需要與其他 AWS 帳戶共用容量，請使用 使用 EC2 隨需容量保留來保留運算容量。

使用您的 預留執行個體

預留執行個體會自動套用至執行中的隨需執行個體，只要其規格相符。若您沒有符合您預留執行個體規格的執行中隨需執行個體，預留執行個體會處於未使用的狀態，直到您使用必要的規格啟動執行個體為止。

如果您要啟動隨需執行個體，以利用預留執行個體的計費利益，請確定您在設定隨需執行個體時指定下列資訊：

平台

您必須指定符合您預留執行個體平台 (產品描述) 的 Amazon Machine Image (AMI)。例如，如果針對您的預留執行個體指定了 Linux/UNIX，您可以從 Amazon Linux AMI 或 Ubuntu AMI 啟動執行個體。

執行個體類型

如果購買了區域預留執行個體，您必須指定與預留執行個體相同的執行個體類型，例如 t3.large。如需詳細資訊，請參閱 區域預留執行個體的套用方式。

如果購買了區域預留執行個體，您必須從預留執行個體之執行個體的同一執行個體系列指定執行個體類型。例如，如果針對預留執行個體指定了 t3.xlarge，您必須從 T3 系列啟動執行個體，但可以指定任何大小，例如 t3.medium。如需詳細資訊，請參閱 區域預留執行個體的套用方式。

可用區域

如果針對特定可用區域購買了區域預留執行個體，您必須將執行個體啟動到相同的可用區域。

如果購買了區域預留執行個體，您可以將執行個體啟動到區域中針對預留執行個體指定的任何可用區域。

租用

執行個體的租用 (dedicated 或 shared) 必須符合預留執行個體的租用。如需詳細資訊，請參閱 Amazon EC2 專用執行個體。

如需預留執行個體如何套用到執行中隨需執行個體的範例，請參閱預留執行個體折扣的套用方式。如需詳細資訊，請參閱為什麼我的 Amazon EC2 預留執行個體不如預期套用至我的 AWS 帳單？

您可以使用各種方法，啟動使用預留執行個體折扣的隨需執行個體。如需不同啟動方法的詳細資訊，請參閱 啟動 Amazon EC2 執行個體。您也可以使用 Amazon EC2 Auto Scaling 來啟動執行個體。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南。

預留執行個體的計費方式

所有預留執行個體都可提供您相較於隨需定價的折扣。您可透過預留執行個體，為整個期間支付費用 (無論實際用量為何)。您可以選擇為您的預留執行個體預付款項、部分預付款項或每月付款，取決於為預留執行個體指定的付款選項。

預留執行個體;到期時，將會根據 EC2 執行個體用量的隨需費率向您收費。最早可提前三年將 預留執行個體 放入佇列供購買。這可協助您確保您的涵蓋範圍不會間斷。如需詳細資訊，請參閱將您的購買項目放入佇列。

AWS 免費方案 適用於新的 AWS 帳戶。如果您使用 AWS 免費方案 執行 Amazon EC2 執行個體，並購買預留執行個體，則會向您收取標準定價。如需相關資訊，請參閱AWS 免費方案。

目錄

- 用量帳單
- 檢視您的帳單
- 預留執行個體 與合併帳單
- 預留執行個體折扣定價方案

用量帳單

預留執行個體會在您選取的期間內以每時鐘小時計價，無論執行個體是否正在執行中。每個鐘點小時都會於標準 24 小時制的整點 (每小時的零分和零秒) 開始。例如，1:00:00 至 1:59:59 是一點鐘。如需執行個體狀態的詳細資訊，請參閱Amazon EC2 執行個體狀態變更。

預留執行個體計費利益可以每秒套用到執行中的執行個體。以秒計費適用於使用開放原始碼 Linux 發行版本 (例如 Amazon Linux 和 Ubuntu) 的執行個體。每小時計費用於商業 Linux 發行版本，例如 Red Hat Enterprise Linux 和 SUSE Linux Enterprise Server。

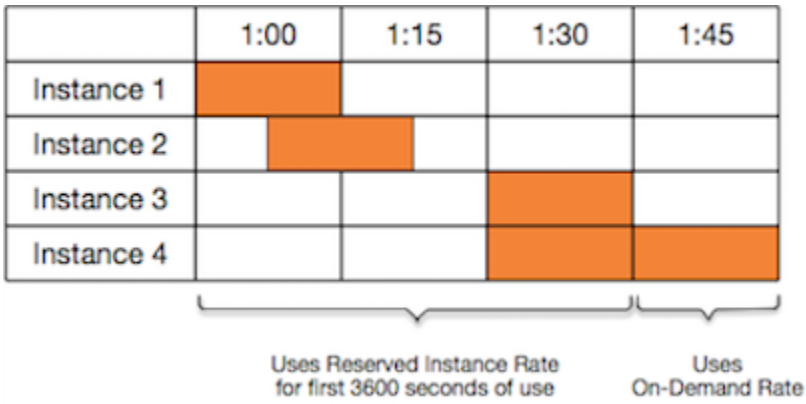
預留執行個體計費利益每個時鐘小時可套用到最多 3600 秒 (一小時) 的執行個體用量。您可以同時執行多個執行個體，但每個時鐘小時最多只能接收到 3600 秒的預留執行個體折扣利益；時鐘小時內超過 3600 秒的執行個體用量會以隨需費率計費。

例如，若您購買一個 `m4.xlarge` 預留執行個體，並同時執行四個 `m4.xlarge` 執行個體一個小時，一個執行個體會以預留執行個體用量收取一個小時的費用，其餘三個執行個體則會以隨需用量收取三個小時的費用。

但是，若您購買一個 `m4.xlarge` 預留執行個體，並在同一個小時內執行四個 `m4.xlarge` 執行個體，每一個執行 15 分鐘 (900 秒)，執行個體的總執行時間便為一個小時，其結果便是一個小時的預留執行個體用量和 0 個小時的隨需用量。



若同時執行多個符合資格的執行個體，預留執行個體計費利益會在一個時鐘小時內同時套用到所有執行個體，最多達 3600 秒；在這之後，便會套用隨需費率。



Billing and Cost Management 主控台上的 Cost Explorer 可讓您根據執行中的隨需執行個體分析節省費用。預留執行個體常見問答集包括清單值計算的範例。

如果您關閉 AWS 帳戶， 資源的隨需計費會停止。但是，若您的帳戶中有任何預留執行個體，您會繼續收到其帳單，直到它們到期為止。

檢視您的帳單

您可以透過檢視 AWS 帳單與成本管理 主控台了解您帳戶所需支付的費用。

- Dashboard (儀表板) 會顯示您帳戶的支出摘要。
- 在 Bills (帳單) 頁面上，於 Details (詳細資訊) 下方，展開 Elastic Compute Cloud 區段及區域，以取得預留執行個體的帳單資訊。

您可在線上檢視費用，或是下載 CSV 檔案。

您也可以使用 AWS 成本和用量報告來追蹤預留執行個體使用率。如需詳細資訊，請參閱了解您的預留。

預留執行個體 與合併帳單

若購買帳戶是在單一合併帳單付款人帳戶下一組帳戶的一部分，則會共用預留執行個體的計費利益。所有成員帳戶的執行個體用量每個月都會彙總到付款人的帳戶中。這通常對有不同功能團隊或群組的公司來說非常有用；然後，會套用標準預留執行個體邏輯計算帳單。如需詳細資訊，請參閱 AWS Organizations的合併帳單。

如果您關閉購買預留執行個體的帳戶，系統會繼續為預留執行個體對付款人帳戶計費，直到預留執行個體到期。在關閉的帳戶於 90 天後永久刪除後，會員帳戶無法再從預留執行個體帳單折扣中受益。

> **Note**
>
> 區域預留執行個體預留容量僅適用於擁有的帳戶，無法與其他 AWS 帳戶共用。如果您需要與其他 AWS 帳戶共用容量，請使用 使用 EC2 隨需容量保留來保留運算容量。

預留執行個體折扣定價方案

若您的帳戶符合折扣定價方案的資格，從您在該方案層級中購買預留執行個體的時間點開始，預付及執行個體使用費便會自動接收到折扣。若要符合折扣的資格，您在區域中的預留執行個體清單價值必須為 500,000 USD (含) 以上。

適用的規定如下：

- 定價方案和相關折扣只會套用到購買的 Amazon EC2 標準預留執行個體。
- 定價方案不會套用到 Windows with SQL Server Standard、SQL Server Web 和 SQL Server Enterprise 的預留執行個體。
- 定價方案不適用於搭載任何類型的 SQL Server 的 Linux 預留執行個體。
- 定價方案折扣僅適用於從 進行的購買 AWS。它們不會套用到購買的第三方預留執行個體。
- 折扣定價方案目前無法套用到可轉換預留執行個體購買。

主題

- 計算 預留執行個體 定價折扣
- 使用折扣方案購買
- 跨定價方案
- 定價方案的合併帳單

計算 預留執行個體 定價折扣

您可以藉由計算您在區域中所有預留執行個體的清單值，來決定帳戶的定價方案。將每個保留每小時的經常性價格，乘上期間內的總小時數，並加上在購買時間點的未折扣預付價格 (也稱為固定價格)。因為清單值是以未折扣 (公有) 定價為基礎，無論您是否符合批發折扣，或是價格是否在您購買預留執行個體之後降低，都不會受到影響。

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

例如，針對 1 年部分預付款的 t2.small 預留執行個體，假設預付價格為 60.00 USD，每小時的費率為 0.007 USD。這會提供清單值 121.32 USD。

```
121.32 = 60.00 + (0.007 * 8760)
```

使用 Amazon EC2 主控台檢視預留執行個體的固定價格值

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 若要顯示預付價格欄，請選擇右上角的設定 ()，開啟預付價格，然後選擇確認。

使用命令列檢視預留執行個體的固定價格值

- describe-reserved-instances (AWS CLI)
- Get-EC2ReservedInstance (AWS Tools for Windows PowerShell)

使用折扣方案購買

當您購買預留執行個體時，Amazon EC2 會自動將任何折扣套用到您購買中位於折扣定價方案內的部分。您無須進行任何不同操作，並且您可以使用任何 Amazon EC2 工具購買預留執行個體。如需詳細資訊，請參閱購買 Amazon EC2 預留執行個體。

當您在區域中的作用中預留執行個體清單值進入折扣定價方案後，日後在該區域購買的任何預留執行個體都會以折扣費率計費。若您在區域中購買的單一預留執行個體超過折扣方案的閾值，則該購買超過價格閾值的部分便會以折扣費率收費。如需在購買程序中建立之暫時預留執行個體 ID 的詳細資訊，請參閱跨定價方案。

若您的清單值低於該折扣定價方案的價格點 (例如：若您的某些預留執行個體到期)，則日後在該區域中購買的預留執行個體將不會有折扣。但是，您原先在折扣定價方案內購買的預留執行個體，仍會繼續套用折扣。

當您購買預留執行個體時，便會發生下列四種可能案例中的其中一種：

- 無折扣 – 您在區域內的購買仍低於折扣閾值。
- 部分折扣 – 您在區域內的購買已超過第一個折扣方案的閾值。不會套用折扣到一或多個保留，剩餘的保留則會套用折扣費率。
- 完整折扣 – 您在區域內的整個購買都已符合單一折扣方案，並已取得適當折扣。
- 雙折扣費率 – 您在區域內的購買已從較低的折扣方案升等至更高的折扣方案。會根據兩種不同費率向您收費：一或多個保留會以較低的折扣費率計費，其餘保留則會以更高的折扣費率計費。

跨定價方案

若您的購買進入折扣定價方案，您會看到多個該購買的項目：其中一個為該購買中以標準價格收費的部分，另外一個則為該購買中以適用折扣費率收費的部分。

預留執行個體服務會產生數個預留執行個體 ID，因為您的購買跨越自未折扣方案，或是從其中一種折扣方案跨越至另一種。方案中每一組保留都會有一個 ID。因此，您購買 CLI 命令或 API 動作傳回的 ID，將會與新預留執行個體的實際 ID 不同。

定價方案的合併帳單

合併帳單帳戶會彙總區域內成員帳戶的清單值。當合併帳單帳戶所有作用中預留執行個體的清單值達到折扣定價方案，合併帳單帳戶的任何成員帳戶在之後購買的預留執行個體都會以折扣費率計費 (只要合併帳戶的清單值持續高於折扣定價方案的閾值)。如需詳細資訊，請參閱預留執行個體 與合併帳單。

購買 Amazon EC2 預留執行個體

若要購買 Amazon EC2 的預留執行個體，您可以使用 Amazon EC2 主控台、命令列工具或 SDK 從 AWS 和第三方賣方搜尋預留執行個體產品，調整搜尋參數，直到找到您正在尋找的完全相符項目為止。

當您在尋找要購買的預留執行個體時，您會收到傳回優惠的成本報價。當您繼續購買時， AWS 會自動對購買價格設定限制價格。您預留執行個體的總成本不會超過向您報價的金額。

若價格因為任何原因產生變更，購買便無法完成。當您從 Amazon EC2 預留執行個體 Marketplace 購買第三方賣方的預留執行個體時，如果有類似您選擇的供應項目，但預付價格較低， AWS 會以較低的預付價格向您銷售供應項目。

在您確認購買前，請檢閱您計劃購買之預留執行個體的詳細資訊，並確認所有的參數都正確。購買預留執行個體後 （無論是從預留執行個體 Marketplace 中的第三方賣方或從中 AWS)，您無法取消購買。您可以排定在未來日期進行購買，並在已排定時間之前取消佇列的購買。

若要購買和修改預留執行個體，請確保使用者具有適當的許可，例如描述可用區域的能力。如需詳細資訊，請參閱 the section called “使用 預留執行個體” (API) 或 the section called “使用 預留執行個體” (主控台)。

任務

- 選擇平台
- 將您的購買項目放入佇列
- 購買標準版 預留執行個體
- 購買 可轉換預留執行個體
- 從預留執行個體 Marketplace 購買
- 取消佇列購買
- 續約 預留執行個體

選擇平台

Amazon EC2 支援下列適用於預留執行個體的平台：

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux
- 帶 HA 的 Red Hat Enterprise Linux
- Windows
- Windows with SQL Server Standard
- Windows with SQL Server Web
- Windows with SQL Server Enterprise

考量事項

- 若您攜帶 Red Hat Enterprise Linux、SUSE Linux 或 Ubuntu Pro 的現有訂閱 (BYOS)，必須為 Linux/Unix 平台選擇方案。
- 執行 macOS 或 Ubuntu Pro 的執行個體不支援預留執行個體 （包含 EC2 訂閱，也就是不支援 BYOS)。若要透過隨需執行個體定價來節省成本，建議您搭配 Savings Plans 使用 macOS 和 Ubuntu Pro (包含 EC2 訂閱) 執行個體。如需詳細資訊，請參閱 Savings Plans 使用者指南。

若要確保執行個體在特定預留執行個體中執行，預留執行個體的平台必須與用於啟動執行個體的 AMI 平台相符 若是 Linux AMI，檢查 AMI 平台是使用一般值 Linux/UNIX，還是 SUSE Linux 等具體的值非常重要。

Console

如需檢查 AMI 平台

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。
3. 請選取 AMI。
4. 在詳細資訊索引標籤上，查看平台詳細資訊的值。

AWS CLI

如需檢查 AMI 平台

使用 describe-images 命令，然後檢查 `PlatformDetails` 的值。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

以下為範例輸出。

```
[
    "Linux/UNIX"
]
```

PowerShell

如需檢查 AMI 平台

使用 Get-EC2Image cmdlet，然後檢查 PlatformDetails 的值。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

以下為範例輸出。

```
PlatformDetails
---------------
Linux/UNIX
```

將您的購買項目放入佇列

根據預設，當您購買 預留執行個體 時，購買會立即進行。或者，您可以將購買放入未來日期和時間的佇列。例如，您可以將購買放入約為現有 預留執行個體 到期時的佇列。這可協助您確保您的涵蓋範圍不會間斷。

您可以將購買放入地區 預留執行個體 的佇列，但不是其他賣方的區域 預留執行個體 或 預留執行個體。最早可提前三年將購買放入佇列。在排程日期和時間，購買會依據預設的付款方式進行。付款成功之後，計費優勢便會套用。

您可以在 Amazon EC2 主控台中為已佇列的購買設定日期，而該購買將排定至該日期的國際標準時間 00:00 分。若要為排入佇列的購買指定不同的時間，請使用 AWS SDK 或命令列工具。

您可以在 Amazon EC2 主控台中查看您的佇列購買。佇列購買的狀態為 queued (佇列)。在排程時間之前，您可以隨時取消佇列購買。如需詳細資訊，請參閱取消佇列購買。

購買標準版 預留執行個體

您可以在特定可用區域中購買標準預留執行個體，取得容量保留。或者，您可以放棄容量保留，購買地區標準預留執行個體。

購買完成之後，若您已有符合預留執行個體規格的執行中執行個體，即會立即套用計費優勢。您不需要重新啟動您的執行個體。若您沒有適合的執行中執行個體，請啟動執行個體，確保符合您在預留執行個體中指定的條件。如需詳細資訊，請參閱使用您的 預留執行個體。

如需預留執行個體套用到您執行中執行個體方式的範例，請參閱預留執行個體折扣的套用方式。

Console

購買預留執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)，然後選擇 Purchase 預留執行個體 (購買 預留執行個體)。
3. 針對 Offering class (優惠類別)，選擇 Standard (標準) 以顯示標準 預留執行個體。
4. 若要購買容量保留，請在購買畫面的右上角開啟 Only show offerings that reserve capacity (只顯示預留容量的優惠)。當您切換此設定時，會出現 Availability Zone (可用區域) 欄位。

   若要購買區域 預留執行個體，請關閉此設定。當您關閉此設定時，Availability Zone (可用區域) 欄位即會消失。
5. 視需要選取其他組態，然後選取 Search (搜尋)。
6. 在您想要購買的每一個 預留執行個體 中，輸入所需數量，然後選擇 Add to cart (新增到購物車)。

   若要從預留執行個體市場購買標準預留執行個體，請在搜尋結果中的 Seller (賣方) 列中尋找 3rd party (第三方)。Term (期間) 資料行會顯示非標準期間。如需詳細資訊，請參閱從預留執行個體 Marketplace 購買。
7. 若要查看您選取之預留執行個體的摘要，請選取 View cart (檢視購物車)。
8. 如果 Order on (訂購日期) 為 Now (立即)，則在您選擇 Order all (全部訂購) 之後會立即完成購買。若要將購買加入佇列，請選取 Now (立即) 並選取日期。您可以為購物車中每項合格產品選取不同的日期。購買項目會放入佇列，直到所選日期的 00:00 UTC。
9. 若要完成訂單，請選擇 Order all (全部訂購)。

   如果在下訂單時，有類似您選擇的優惠，但價格較低， 會以較低的價格 AWS 向您銷售優惠。
10. 選擇 Close (關閉)。

    訂單的狀態會列在 State (狀態) 資料行中。當您的訂單完成時，State (狀態) 的值便會從 `Payment-pending` 變更為 `Active`。當預留執行個體為 `Active` 時，即已準備好可供使用。

    如果狀態變為 `Retired`， AWS 可能尚未收到您的付款。

AWS CLI

購買標準預留執行個體

1. 使用 describe-reserved-instances-offerings 命令搜尋可用的預留執行個體。為 `--offering-class` 選項指定 `standard`，以僅傳回標準預留執行個體。您可以套用其他條件來縮小結果範圍。例如，使用下列命令來購買使用預設租用 `t2.large` 為期僅一年的區域 `Linux/UNIX` 預留執行個體：

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class standard \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=duration,Values=31536000 \
              Name=scope,Values=Region
```

若只尋找預留執行個體 Marketplace 上的預留執行個體，請使用 `marketplace` 篩選條件，並且不要在請求中指定有效期間，因為有效期間可能會短於 1 年或 3 年。

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class standard \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=marketplace,Values=true
```

當您找到符合您需求的預留執行個體時，請記下產品 ID。例如：

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. 使用 purchase-reserved-instances-offering 命令購買您的預留執行個體。您必須指定您在先前步驟中取得的預留執行個體優惠 ID，並指定保留的執行個體數目。

```
aws ec2 purchase-reserved-instances-offering \
    --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
    --instance-count 1
```

依預設，購買會立刻完成。或者，若要將購買加入佇列，則在先前的呼叫中新增下列選項。

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. 使用 describe-reserved-instances 命令取得您預留執行個體的狀態。

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

購買標準預留執行個體

1. 使用 Get-EC2ReservedInstancesOffering cmdlet，尋找可用的預留執行個體。為 `-OfferingClass` 參數指定 `standard`，以僅傳回標準預留執行個體。您可以套用其他條件來縮小結果範圍。例如，使用下列命令來購買使用預設租用 `t2.large` 為期僅一年的區域 `Linux/UNIX` 預留執行個體：

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="duration"; Values="31536000"} `
             @{Name="scope"; Values="Region"
```

若只尋找預留執行個體 Marketplace 上的預留執行個體，請使用 `marketplace` 篩選條件，並且不要在請求中指定有效期間，因為有效期間可能會短於 1 年或 3 年。

```
Get-EC2ReservedInstancesOffering `
    -InstanceType t2.large `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy default `
    -Filters @{Name="marketplace"; Values="true"}
```

當您找到符合您需求的預留執行個體時，請記下產品 ID。例如：

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. 使用 New-EC2ReservedInstance cmdlet，購買預留執行個體。您必須指定您在先前步驟中取得的預留執行個體優惠 ID，並指定保留的執行個體數目。

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

依預設，購買會立刻完成。或者，若要將購買加入佇列，則在先前的呼叫中新增下列參數。

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. 使用 Get-EC2ReservedInstance cmdlet，取得預留執行個體的狀態。

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

購買 可轉換預留執行個體

您可以在特定可用區域中購買可轉換預留執行個體，取得容量保留。或者，您可以放棄容量保留並購買地區可轉換預留執行個體。

若您已有符合預留執行個體規格的執行中執行個體，即會立即套用計費利益。您不需要重新啟動您的執行個體。若您沒有適合的執行中執行個體，請啟動執行個體，確保符合您在預留執行個體中指定的條件。如需詳細資訊，請參閱使用您的 預留執行個體。

如需預留執行個體套用到您執行中執行個體方式的範例，請參閱預留執行個體折扣的套用方式。

Console

購買可轉換預留執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)，然後選擇 Purchase 預留執行個體 (購買 預留執行個體)。
3. 針對 Offering class (優惠類別)，選擇 Convertible (可轉換) 以顯示標準 可轉換預留執行個體。
4. 若要購買容量保留，請在購買畫面的右上角開啟 Only show offerings that reserve capacity (只顯示預留容量的優惠)。當您切換此設定時，會出現 Availability Zone (可用區域) 欄位。

若要購買區域 預留執行個體，請關閉此設定。當您關閉此設定時，Availability Zone (可用區域) 欄位即會消失。

5. 視需要選取其他組態，然後選取 Search (搜尋)。
6. 在您想要購買的每一個 可轉換預留執行個體 中，輸入數量，然後選擇 Add to cart (新增到購物車)。
7. 若要查看您選取項目的摘要，請選取 View cart (檢視購物車)。
8. 如果 Order on (訂購日期) 為 Now (立即)，則在您選擇 Order all (全部訂購) 之後會立即完成購買。若要將購買加入佇列，請選取 Now (立即) 並選取日期。您可以為購物車中每項合格產品選取不同的日期。購買項目會放入佇列，直到所選日期的 00:00 UTC。
9. 若要完成訂單，請選擇 Order all (全部訂購)。

   如果在下訂單時，有類似您選擇的優惠，但價格較低， 會以較低的價格 AWS 向您銷售優惠。
10. 選擇 Close (關閉)。

    訂單的狀態會列在 State (狀態) 資料行中。當您的訂單完成時，State (狀態) 的值便會從 `Payment-pending` 變更為 `Active`。當預留執行個體為 `Active` 時，即已準備好可供使用。

    如果狀態變為 `Retired`， AWS 可能尚未收到您的付款。

AWS CLI

購買可轉換預留執行個體

1. 使用 describe-reserved-instances-offerings 命令搜尋可用的預留執行個體。為 `--offering-class` 選項指定 `convertible`，以僅傳回可轉換預留執行個體。您可以套用其他條件來縮小結果範圍。例如，使用下列命令，購買適用於 `Linux/UNIX` 之預設租用的區域 `t2.large` 預留執行個體。

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class convertible \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=scope,Values=Region
```

   當您找到符合您需求的預留執行個體時，請記下產品 ID。例如：

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. 使用 purchase-reserved-instances-offering 命令購買您的預留執行個體。您必須指定您在先前步驟中取得的預留執行個體優惠 ID，並指定保留的執行個體數目。

```
aws ec2 purchase-reserved-instances-offering \
    --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
    --instance-count 1
```

依預設，購買會立刻完成。或者，若要將購買加入佇列，則在先前的呼叫中新增下列選項。

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. 使用 describe-reserved-instances 命令取得您預留執行個體的狀態。

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

購買可轉換預留執行個體

1. 使用 Get-EC2ReservedInstancesOffering cmdlet，尋找可用的預留執行個體。為 `-OfferingClass` 參數指定 `convertible`，以僅傳回可轉換預留執行個體。您可以套用其他條件來縮小結果範圍。例如，使用下列命令，購買適用於 Linux/UNIX 之預設租用的區域 `t2.large` 預留執行個體。

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "convertible" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="scope"; Values="Region"
```

當您找到符合您需求的預留執行個體時，請記下產品 ID。例如：

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. 使用 New-EC2ReservedInstance cmdlet，購買預留執行個體。您必須指定您在先前步驟中取得的預留執行個體優惠 ID，並指定保留的執行個體數目。

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

依預設，購買會立刻完成。或者，若要將購買加入佇列，則在先前的呼叫中新增下列參數。

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. 使用 Get-EC2ReservedInstance cmdlet，取得預留執行個體的狀態。

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

從預留執行個體 Marketplace 購買

您可以從第三方賣方購買預留執行個體，他們擁有預留執行個體 Marketplace 不再需要的預留執行個體。您可以使用 Amazon EC2 主控台或命令列工具來執行這項作業。程序類似於從 購買預留執行個體 AWS。如需詳細資訊，請參閱購買標準版 預留執行個體。

在預留執行個體 Marketplace 中購買的預留執行個體與直接從中購買的預留執行個體之間存在一些差異 AWS：

- 有效期間 – 您從第三方賣方購買的預留執行個體具有低於完整標準有效期間的剩餘時間。從 AWS 執行一年或三年的完整標準術語。
- 預付價格 – 第三方預留執行個體可使用不同的預付價格出售。使用費或經常性費用仍保持不變，等同於最初從 AWS購買預留執行個體時所設定的費用。
- 預留執行個體類型 – 只能從預留執行個體市場購買 Amazon EC2 標準預留執行個體。可轉換的預留執行個體、Amazon RDS 和 Amazon ElastiCache 預留執行個體無法在預留執行個體 Marketplace 上購買。

您的基本資訊會與賣方共享 (例如您的郵遞區號和地區/國家資訊)。

此資訊可讓賣方計算任何需要匯款給政府的必要交易稅 (例如營業稅或加值稅)，並會做為收款報告提供。在極少數情況下， AWS 可能需要向賣方提供您的電子郵件地址，以便他們可以就與銷售相關的問題與您聯絡 （ 例如稅務問題 ）。

基於類似原因， 會在買方的購買發票上 AWS 共用賣方的法人實體名稱。若您因為稅務或其他相關原因，需要賣方的額外資訊，請聯絡 支援。

取消佇列購買

最早可提前三年將購買放入佇列。在排程時間之前，您可以隨時取消佇列購買。

Console

取消佇列購買

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 選取一或多個 預留執行個體。
4. 選擇 Actions (動作)、Delete queued Reserved Instances (刪除佇列預留執行個體)。
5. 出現確認提示時，依序選擇 Delete (刪除)、Close (關閉)。

AWS CLI

取消佇列購買

使用 delete-queued-reserved-instances 命令。

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327
```

PowerShell

取消佇列購買

使用 Remove-EC2QueuedReservedInstance cmdlet。

```
Remove-EC2QueuedReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327
```

續約 預留執行個體

您可以在預留執行個體排定到期前續約。續約預留執行個體會將具有相同預留執行個體組態的購買排入佇列，直到當前預留執行個體到期為止。

您必須透過 Amazon EC2 主控台續約預留執行個體。

使用已佇列的購買續約預留執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 選取要續約的預留執行個體。
4. 選擇 Actions (動作)、Renew Reserved Instances (續約預留執行個體)。
5. 若要完成訂單，請選擇 Order all (全部訂購)，然後選擇 Close (關閉)。

銷售 Amazon EC2 可轉換預留執行個體不能在預留執行個體 Marketplace 中銷售

Amazon EC2 預留執行個體 Marketplace 是一種平台，可協助 AWS 向客戶和第三方賣方銷售未使用的標準預留執行個體。這些預留執行個體的期限長度和定價選項可能會有所不同。當您不再需要預留執行個體時，例如當您將執行個體移至新的 AWS 區域、變更為不同的執行個體類型、在預留執行個體期限到期之前完成專案、您的業務需求變更，或您有多餘的容量時，您可能會想要銷售預留執行個體。

在預留執行個體 Marketplace 中列出您的預留執行個體後，它們便可供潛在買方尋找。所有預留執行個體都會根據剩餘期間的長度和每小時的價格分組。

為了滿足買方透過預留執行個體 Marketplace 購買第三方賣方預留執行個體的請求， AWS 首先在指定的分組中以最低的預付價格銷售預留執行個體。然後，以下一個最低價格 AWS 銷售預留執行個體，直到買方完成整個訂單為止。 AWS 然後， 會處理交易並將預留執行個體的所有權轉移給買方。

直到賣出之前，您都會擁有您的預留執行個體。在銷售之後，您便已放棄容量保留和折扣後的經常性費用。若您繼續使用您的執行個體， AWS 會從您賣出預留執行個體的時間開始，以隨需價格向您收費。

若您希望在預留執行個體 Marketplace 中銷售您未使用的預留執行個體，您必須符合特定適用性條件。

關於在預留執行個體 Marketplace 上購買預留執行個體的資訊，請參閱 從預留執行個體 Marketplace 購買。

內容

- 法規與限制
- 註冊為賣家

- 用於收款的銀行帳戶
- 稅務資訊
- 為您的 預留執行個體 定價
- 列出您的 預留執行個體
- 預留執行個體清單狀態
- 清單的生命週期
- 在您的預留執行個體售出後
- 收取款項
- 與買方共享的資訊

法規與限制

在您販售未使用的預留前，您必須先在預留執行個體 Marketplace 中註冊為賣方。如需相關資訊，請參閱註冊為賣家。

以下限制會在販售預留執行個體時適用：

- Amazon EC2 標準地區與區域預留執行個體只能在預留執行個體市場中銷售。
- Amazon EC2 可轉換預留執行個體不能在預留執行個體 Marketplace 中銷售。
- Amazon RDS 和 Amazon ElastiCache 等 AWS 其他服務的預留執行個體無法在預留執行個體 Marketplace 中販售。
- 標準預留執行個體的剩餘期間至少要一個月。
- 您無法在預設停用的區域中販售標準預留執行個體。
- 預留執行個體 Marketplace 允許的最低價為 0.00 美元。
- 您可以在預留執行個體 Marketplace 中銷售無預留、部分預付或全部預付等執行個體，只要其在您的帳戶中處於活動狀態至少 30 天。此外，如果預留執行個體上有預付款，則只有在 AWS 收到預付款之後才能販售。
- 如果您使用磁碟區折扣購買預留執行個體，則無法在預留執行個體 Marketplace 中銷售預留執行個體。
- 您無法直接在預留執行個體 Marketplace 中修改您的清單。但是，您可以藉由先取消它，然後再使用新的參數建立另一個清單，來變更您的清單。如需相關資訊，請參閱為您的 預留執行個體 定價。您也可以在將它們列出之前修改您的預留執行個體。如需相關資訊，請參閱修改 預留執行個體。
- AWS 對於您在預留執行個體 Marketplace 中銷售的每個標準預留執行個體，收取總預付價格的 12% 的服務費用。預付價格為賣方向標準預留執行個體收取的價格。

- 當您註冊成為賣方時，您指定的銀行必須具有美國地址。如需詳細資訊，請參閱 AWS Marketplace 賣方指南中的付費產品的其他賣方需求。
- Amazon Web Services India Private Limited (AWS 印度） 客戶無法在 EC2 預留執行個體 Marketplace 上註冊為賣方，也無法在 EC2 預留執行個體 Marketplace 中列出或銷售預留執行個體，即使他們擁有美國銀行帳戶。如需詳細資訊，請參閱 AWS 帳戶 和 AWS 印度帳戶之間的差異有哪些？
- 如果您將記錄賣方變更為 Amazon Web Services India Private Limited (AWS India)，將從 EC2 預留執行個體 Marketplace 取消註冊為賣方，並將移除 EC2 預留執行個體 Marketplace 上所有現有的作用中清單。要還原賣方狀態，您必須將帳戶位置變更為印度以外的國家/地區，並重新完成賣方註冊程序。

註冊為賣家

> **Note**
>
> 只有 AWS 帳戶根使用者 可以將帳戶註冊為賣方。

若要在預留執行個體 Marketplace 中進行銷售，您必須先註冊為賣方。在註冊期間，您會提供下列資訊：

- 銀行資訊 -AWS 必須擁有您的銀行資訊，才能在您銷售保留時支付款項。您指定的銀行必須具有美國地址。如需詳細資訊，請參閱用於收款的銀行帳戶。
- 稅務資訊 – 所有賣方都必須完成稅務資訊面試，才能判斷任何必要的稅務報告義務。如需詳細資訊，請參閱稅務資訊。

在 AWS 收到您完成的賣方註冊後，您會收到一封電子郵件，確認您的註冊並通知您可以在預留執行個體 Marketplace 中開始銷售。

用於收款的銀行帳戶

AWS 必須有您的銀行資訊，才能在您銷售預留執行個體時支付已收取的資金。您指定的銀行必須具有美國地址。如需詳細資訊，請參閱 AWS Marketplace 賣方指南中的付費產品的其他賣方需求。

註冊預設銀行帳戶以用於接收款項

1. 開啟預留執行個體市場賣方註冊頁面，然後使用您的 AWS 登入資料登入。
2. 在 Manage Bank Account (管理銀行帳戶) 頁面上，提供下列關於接收款項銀行的資訊：

- 銀行帳戶持有人名稱
- 路由號碼
- 帳戶號碼
- 銀行帳戶類型

> **Note**
>
> 如果您使用公司銀行帳戶，系統會提示您透過傳真 (1-206-765-3424) 傳送銀行帳戶的相關資訊。

在註冊之後，提供的銀行帳戶便會設為預設，等待銀行進行驗證。驗證新的銀行帳戶最多可能需要兩週的時間，在此期間您無法收到款項。針對已建立的帳戶，支付款項通常需要兩天的時間才能完成。

變更接收款項的預設銀行帳戶

1. 在預留執行個體市場賣方註冊頁面上，使用您用來註冊的帳戶登入。
2. 在 Manage Bank Account (管理銀行帳戶) 頁面上，新增銀行帳戶或視需要修改銀行帳戶。

稅務資訊

您販售預留執行個體的行為必須遵守交易式稅務的規範，例如營業稅或加值稅。建議您和您公司的稅務、法律、財務或會計部門確認，以判斷是否適用於交易式的稅務。您必須負責收集和將交易式的稅務傳送給適當的稅務主管機關。

做為賣方註冊程序的一部分，您必須在賣方註冊入口網站中完成稅務面試。面試會收集您的稅務資訊，並填入 IRS W-9、W-8BEN，或 W-8BEN-E 表單。這份表單會用於判斷是否有任何必要的稅務報告義務。

您做為稅務面試一部分所輸入的稅務資訊可能會有所不同，取決於您是否是以個人或公司運作，以及您或您的公司是否為美國或非美國的個人或實體。當您填寫稅務面試時，請謹記下列幾點：

- 提供的資訊 AWS，包括本主題中的資訊，並不構成稅務、法律或其他專業建議。若要了解 IRS 報告需求影響您企業的方式，或是您有其他問題，請聯絡您的稅務、法律或其他專業顧問。
- 若要盡可能有效率的完成 IRS 報告需求，請回答所有問題並輸入所有面試中請求的資訊。
- 檢查您的答案。避免拼字錯誤或輸入不正確的稅務識別號碼。這可能會導致無效的稅務表單。

根據您的稅務面試回應和 IRS 報告閾值，Amazon 可能會提出 1099-K 表單。Amazon 會將您 1099-K 表單的複本在您的稅務帳戶達到閾值層級年份的下一年，於 1 月 31 日或在 1 月 31 日前寄送給您。例如，若您的帳戶在 2018 年達到閾值，則您的 1099-K 表單會在 2019 年 1 月 31 日或之前寄送給您。

如需 IRS 需求和 1099-K 表單的詳細資訊，請參閱 IRS 網站上的 1099-K 表單常見問題集。

為您的 預留執行個體 定價

設定預留執行個體的價格時，請考慮下列事項：

- 預付價格 - 預付價格是您可以為要販售之預留執行個體指定的唯一價格。預付價格是買方購買每個預留執行個體時所需支付的一次性價格。

  由於預留執行個體的值會隨著時間而降低，預設情況下， AWS 可以設定價格，以逐月減少相等的增量。但是，您可以根據販售您保留的時間，設定不同的預付價格。例如，若您的預留執行個體具有九個月的剩餘期間，您可以指定客戶購買具有九個月剩餘期間的該預留執行個體時，您所能接受的價格。您可以為剩餘期間五個月設定另一個價格，以及為剩餘期間一個月設定另一個價格。

  預留執行個體 Marketplace 允許的最低價為 0.00 美元。
- 限制 - 銷售預留執行個體的以下限制適用於 AWS 帳戶的生命週期。其並非年度限制，並且無法增加。
  - 您可以在預留執行個體賣出最多 50,000 美元。
  - 您最多可以賣 5,000 個預留執行個體
- 無法修改 - 您無法直接修改清單。但是，您可以藉由先取消它，然後再使用新的參數建立另一個清單，來變更您的清單。
- 可以取消 – 只要清單處於 `active` 狀態，即可隨時取消您的清單。若清單已相符或正在為銷售進行處理，您便無法取消清單。若您清單中的某些執行個體已相符且您取消了清單，則只有剩餘未相符的執行個體會從清單中移除。

列出您的 預留執行個體

身為已註冊賣方，您可以選擇銷售一或多個預留執行個體。您可以選擇一次出售全部或部分出售。此外，您可以使用任何執行個體類型、平台和範圍組態，列出預留執行個體。

主控台會決定建議價格。它會檢查符合 預留執行個體 的優惠，並匹配最低價格的產品。否則會根據 預留執行個體 剩餘時間的成本來計算建議價格。如果計算的值低於 1.01 USD，則建議價格為 1.01 USD。

若您取消您的清單，但該清單中的一部分已賣出，則取消對已經賣出的部分不會有任何效果。只有清單中尚未賣出的部分才會無法在預留執行個體 Marketplace 中取得。

Console

列出預留執行個體 Marketplace 中的預留執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 選取要列出的預留執行個體，然後選取 Actions (動作)、Sell 預留執行個體 (販售 預留執行個體)。
4. 在 Configure Your 預留執行個體 Listing (設定您的預留執行個體清單) 頁面上，在相關資料行中設定要販售的執行個體數目，以及剩餘期間的預付價格。選取 Months Remaining (剩餘月數) 資料行，查看您保留的值隨剩餘期間變化的方式。
5. 若您是進階使用者希望自訂定價，您可以為後續月份輸入不同的值。若要返回預設的線性價格下降，請選擇 Reset (重設)。
6. 當您完成設定清單後，請選擇 Continue (繼續)。
7. 在 Confirm Your 預留執行個體 Listing (確認您的預留執行個體清單) 頁面上確認您清單的詳細資訊，若您已感到滿意，請選擇 List Reserved Instance (列出預留執行個體)。

在 Amazon EC2 主控台中檢視您的清單

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 選取您已列出的預留執行個體，並選取頁面底部附近的 My Listings (我的清單) 索引標籤。

AWS CLI

管理預留執行個體 Marketplace 中的預留執行個體

1. 透過使用 describe-reserved-instances 命令來取得您預留執行個體的清單。記下您要列出的預留執行個體 ID。
2. 使用 create-reserved-instances-listing 命令。您必須指定預留執行個體的 ID、執行個體數目，以及定價排程。
3. 若要檢視您的清單，請使用 describe-reserved-instances-listings 命令。
4. 若要取消您的清單，請使用 cancel-reserved-instances-listing 命令。

PowerShell

管理預留執行個體 Marketplace 中的預留執行個體

1. 使用 Get-EC2ReservedInstance cmdlet，取得預留執行個體的清單。記下您要列出的預留執行個體 ID。
2. 使用 New-EC2ReservedInstancesListing cmdlet。您必須指定預留執行個體的 ID、執行個體數目，以及定價排程。
3. 要檢視您的清單，請使用 Get-EC2ReservedInstancesListing cmdlet。
4. 要取消您的清單，請使用 Stop-EC2ReservedInstancesListing cmdlet。

預留執行個體清單狀態

預留執行個體頁面的 My Listings (我的清單) 標籤上的 Listing State (清單狀態) 會顯示您清單目前的狀態：

Listing State (清單狀態) 顯示的資訊與預留執行個體市場中的清單狀態有關。它與 Reserved Instances (預留執行個體) 中 State (狀態) 資料行顯示的狀態資訊不同。此 State (狀態) 資訊與您的保留有關。

- active (作用中) – 清單可供購買。
- canceled (已取消) - 清單已取消，無法在預留執行個體市場中購買。
- closed (已關閉) – 預留執行個體並未列出。預留執行個體可能會因為清單的銷售已完成而為 `closed` 狀態。

清單的生命週期

當您清單中的所有執行個體都已符合並售出，My Listings (我的清單) 標籤便會顯示 Total instance count (總執行個體計數) 符合 Sold (已售出) 下方列出的計數。此外，您的清單中將不會有任何剩餘 Available (可用) 的執行個體，且其 Status (狀態) 為 `closed`。

當您的清單只有一部分售出時， 會 AWS 淘汰清單中的預留執行個體，並建立與計數中剩餘預留執行個體相等的預留執行個體數目。因此，清單 ID 和其代表的清單 (目前具有較少銷售中的保留) 仍會處於作用中。

任何之後此清單內發生的預留執行個體銷售都會使用相同的方式處理。當清單中所有預留執行個體售出時， 會將清單 AWS 標記為 `closed`。

例如，您建立 預留執行個體 listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample，其清單計數為 5。

Reserved Instance (預留執行個體) 主控台頁面中的 My Listings (我的清單) 標籤會以此方式顯示清單：

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 總保留計數 = 5
- 售出 = 0
- 可用 = 5
- 狀態 = 作用中

買方購買兩個保留，使剩餘可供銷售的保留計數為 3。由於此部分銷售， AWS 會建立計數為 3 的新保留，以代表尚待銷售的剩餘保留。

這是在 My Listings (我的清單) 標籤中您清單的外觀：

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 總保留計數 = 5
- 售出 = 2
- 可用 = 3
- 狀態 = 作用中

若您取消您的清單，但該清單中的一部分已賣出，則取消對已經賣出的部分不會有任何效果。只有清單中尚未賣出的部分才會無法在預留執行個體 Marketplace 中取得。

在您的預留執行個體售出後

當您的預留執行個體售出時， 會 AWS 傳送電子郵件通知給您。每天若有任何類型的活動，您都會接收到一封電子郵件通知，擷取一天內所有的活動。當您建立或銷售清單，或 AWS 將資金傳送到您的帳戶時，活動可以包含 。

Console

追蹤預留執行個體清單的狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。

3. 我的清單索引標籤上，尋找清單狀態值。該索引標籤同時也包含期間、清單價格，清單中有多少可用、擱置中、已售出和已取消執行個體解析的相關資訊。

AWS CLI

追蹤預留執行個體清單的狀態

可以搭配適當的篩選條件，使用 describe-reserved-instances-listings 命令來取得您清單的相關資訊。

```
aws ec2 describe-reserved-instances-listings
```

PowerShell

追蹤預留執行個體清單的狀態

使用 Get-EC2ReservedInstancesListing cmdlet。

```
Get-EC2ReservedInstancesListing
```

收取款項

一旦 AWS 收到買方的資金，就會傳送訊息給已售出預留執行個體的已註冊擁有者帳戶電子郵件。

AWS 會將自動清算中心 (ACH) 電匯傳送至您指定的銀行帳戶。通常，這種匯款會在賣出您預留執行個體的一到三天後發生。每天支付一次。您將收到一封電子郵件，含有釋出基金後的收款報告。請記住，在 收到您的銀行驗證之前，您無法 AWS 收到付款。這可能需要兩週的時間。

您賣出的預留執行個體會繼續在您描述您的預留執行個體時出現。

您透過直接電匯至銀行帳戶的方式收到預留執行個體的現金支出。 AWS 會針對您在預留執行個體 Marketplace 中銷售的每個預留執行個體收取總預付價格的 12% 的服務費用。

與買方共享的資訊

當您在預留執行個體 Marketplace 中銷售 時， 會根據美國法規，在買方的陳述式上 AWS 共用您公司的法定名稱。此外，若買方因為需要聯絡您取得發票或其他與稅務相關的原因而呼叫 支援 ， AWS 可能會需要向買方提供您的電子郵件地址，以便買方能夠直接聯絡您。

基於相似的原因，也會在賣方的收款報告中提供買方的郵遞區號和國家資訊。做為賣方，您可能需要此資訊伴隨任何您匯款給政府的必要交易稅 (例如營業稅和加值稅)。

AWS 無法提供稅務建議，但如果您的稅務專家判斷您需要特定的其他資訊，請聯絡 支援。

修改 預留執行個體

當您的需求變更時，您可以修改您的標準或可轉換預留執行個體，並繼續獲得計費利益。您可以修改預留執行個體的屬性，例如可用區域、執行個體大小 (位於相同執行個體系列和層代內) 和範圍。

> **Note**
>
> 您也可以將可轉換預留執行個體與另一個具有不同組態的可轉換預留執行個體交換。如需詳細資訊，請參閱交換 可轉換預留執行個體。

您也可以修改所有預留執行個體或子集。您可以將原始預留執行個體分隔為兩個或多個新的預留執行個體。例如，若您在 us-east-1a 中有 10 個執行個體的保留，並決定將 5 個執行個體移動到 us-east-1b，修改請求結果會導致兩個新的保留：其中一個保留針對位於 us-east-1a 中的 5 個執行個體，另外一個保留則針對位於 us-east-1b 的 5 個執行個體。

您也可以將兩個或多個預留執行個體合併為單一預留執行個體。例如，若您有四個 t2.small 預留執行個體，每一個都具有一個執行個體，您可以合併它們來建立一個 t2.large 預留執行個體。如需詳細資訊，請參閱支援執行個體大小的修改。

在修改後，預留執行個體的利益僅會套用到符合新參數的執行個體。例如，若您變更保留的可用區域，容量保留和定價利益會自動套用到新可用區域中的執行個體用量。不再符合新參數的執行個體會以隨需費率計費，除非您的帳戶有其他適用的保留。

若您的修改請求成功：

- 修改後的保留會立即生效，定價利益會在修改請求的小時內開始套用到新的執行個體。例如，若您在下午 9:15 成功修改您的保留，定價利益會在下午 9:00 傳送到您新的執行個體。您可以藉由使用 describe-reserved-instances 命令來取得修改後預留執行個體的生效日期。
- 原始的保留會淘汰。其結束日期便是新保留的開始日期，新保留的結束日期則與原始預留執行個體的結束日期相同。若您修改具有 16 個月剩餘期間的三年保留，其結果產生的修改後保留便是 16 個月的保留，其結束日期與原先的保留相同。
- 修改後的保留列出的固定價格為 0 USD，而非原始保留的固定價格。

- 修改後保留的固定價格不會影響套用到您帳戶的折扣定價方案計算，因為該計算是以原始保留的固定價格為基礎。

若您的修改請求失敗，您的預留執行個體將會維持其原始的組態，並可立即進行另一次的修改請求。

修改無須支付任何費用，您不會收到任何新的帳單或發票。

您可以無限次數的修改您的保留，但在您提交後，您便無法變更或取消擱置中的修改請求。在修改成功完成後，您可以提交另一個修改請求，轉返任何您進行的變更 (若需要的話)。

內容

- 修改的要求和限制
- 支援執行個體大小的修改
- 提交修改請求
- 故障診斷修改請求

修改的要求和限制

您可以修改這些屬性如下。

可修改屬性	支援平台	限制及考量
變更相同區域內的可用區域	Linux 與 Windows	-
將範圍從「可用區域」變更為「區域」，反之亦然	Linux 與 Windows	區域性預留執行個體的範圍限定為可用區域，並在該可用區域中預留容量。若您將範圍從可用區域變更為區域 (換言之，從區域到地區)，您將會喪失容量保留利益。 地區性預留執行個體的範圍限定為區域。您的預留執行個體折扣可套用到該區域內任何可用區域中正在執行的執行個體。此外，預留執行個體折扣

可修改屬性	支援平台	限制及考量
		適用於選定執行個體系列中所有大小的執行個體用量。若您將範圍從區域變更為可用區域 (換言之，從地區到區域)，您將會喪失可用區域彈性和執行個體大小彈性 (若適用)。 如需詳細資訊，請參閱預留執行個體折扣的套用方式。
在相同執行個體系列和層代內變更執行個體大小	僅限 Linux/UNIX 執行個體大小的彈性不適用於其他平台上的預留執行個體，這些平台包括 Linux with SQL Server Standard、Linux with SQL Server Web、Linux with SQL Server Enterprise、Red Hat Enterprise Linux、SUSE Linux、Windows、Windows with SQL Standard、Windows with SQL Server Enterprise 和 Windows with SQL Server Web。	保留必須使用預設租用。某些執行個體系列不受支援，因為沒有其他可用的大小。如需詳細資訊，請參閱支援執行個體大小的修改

要求

Amazon EC2 會在有足夠容量接受您的新組態 (若適用)，以及滿足下列條件時處理您的修改請求：

- 預留執行個體在您購買之前或當時無法修改
- 預留執行個體必須為作用中
- 不能有擱置的修改請求
- 預留執行個體 Marketplace 中未列出預留執行個體。

- 原始保留的執行個體大小使用量和新組態必須相符。如需詳細資訊，請參閱支援執行個體大小的修改。
- 原始 預留執行個體 全部為標準版 預留執行個體 或全部為 可轉換預留執行個體，而非每個類型各有一些
- 原始 預留執行個體 如果是標準版 預留執行個體，則必須在同一小時內過期
- 如需修改執行個體大小，預留執行個體必須支援執行個體大小彈性。如需不支援執行個體大小彈性的預留執行個體清單，請參閱 執行個體大小彈性。

支援執行個體大小的修改

若符合下列要求，您可以修改預留執行個體的執行個體大小。

要求

- 此平台為 Linux/UNIX。
- 您必須選取相同執行個體系列 (用字母表示，例如 T) 和層代 (用數字表示，例如 2) 中的其它執行個體大小。

  例如，您可以將預留執行個體從 `t2.small` 修改為 `t2.large`，因為它們都位於相同的 T2 系列和層代中。但是您無法將預留執行個體從 T2 修改為 M2，或從 T2 修改為 T3，因為在這兩個範例中，目標執行個體系列和層代與原始預留執行個體不同。
- 只有在支援執行個體大小彈性時，您才能修改預留執行個體的執行個體大小。如需不支援執行個體大小彈性的預留執行個體清單，請參閱 執行個體大小彈性。
- 您無法修改 `t1.micro` 執行個體的預留執行個體大小，因為 `t1.micro` 只有一個大小。
- 原始和新的 預留執行個體 必須具有相同的執行個體大小使用量。

內容

- 執行個體大小使用量
- 裸機執行個體的正規化因素

執行個體大小使用量

每一個預留執行個體都具有執行個體大小使用量，由執行個體大小的正規化因素和保留中的執行個體數目決定。當您修改 預留執行個體 中的執行個體大小時，新組態的使用量必須符合原始組態的使用量，否則不會處理修改請求。

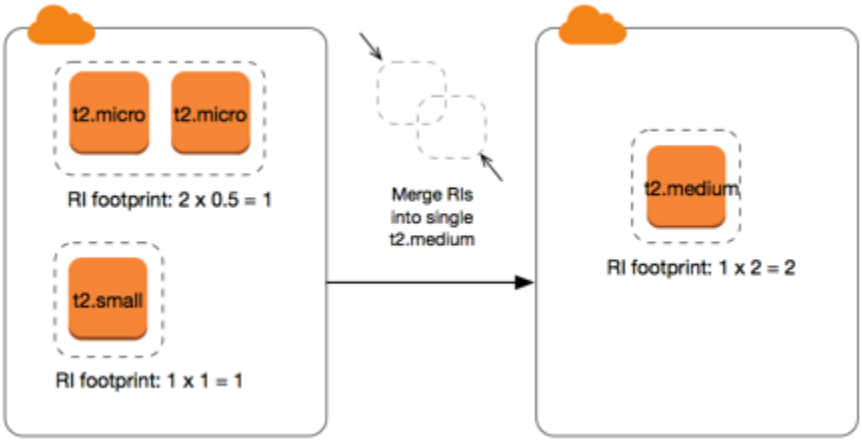
若要計算預留執行個體的執行個體大小使用量，請將執行個體的數目乘上正規化因素。在 Amazon EC2 主控台中，正規化因素以單位進行測量。下表說明執行個體系列中執行個體大小的正規化因素。例如，`t2.medium` 的正規化因素為 2，因此四個 `t2.medium` 執行個體的保留具有 8 個單位的使用量。

執行個體大小	正規化因素
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192

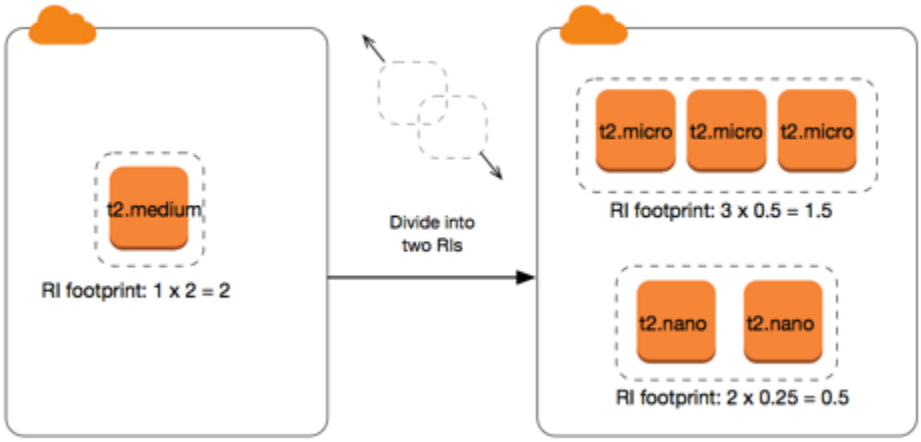
執行個體大小	正規化因素
32xlarge	256
48xlarge	384
56xlarge	448
96xlarge	768
112xlarge	896

您可以將您的保留在相同執行個體系列中配置到不同的執行個體大小，只要您保留的執行個體大小使用量維持不變。例如，您可以將一個 t2.large (1 @ 4 個單位) 執行個體的保留分割為四個 t2.small (4 @ 1 個單位) 執行個體。同樣地，您可以將四個 t2.small 執行個體的保留合併為一個 t2.large 執行個體。不過，您無法將兩個 t2.small 執行個體的保留變更為一個 t2.large 執行個體，因為新保留的使用量 (4 個單位) 大於原始保留的使用量 (2 個單位)。

在下列範例中，您有兩個 t2.micro 執行個體的保留 (1 個單位)，以及一個 t2.small 執行個體的保留 (1 個單位)。如果您將這兩個保留合併為具有一個 t2.medium 執行個體 (2 個單位) 的單一保留，新保留的使用量等於合併保留的使用量。



您也可以修改保留，將其分割為兩個或多個保留。在下列範例中，您有一個具有 t2.medium 執行個體 (2 個單位) 的保留。您可以將保留分割為兩個保留，一個具有兩個 t2.nano 執行個體 (5 個單位)，另一個具有三個 t2.micro 執行個體 (1.5 個單位)。



裸機執行個體的正規化因素

您可以使用相同執行個體系列內的其他大小，修改具有 metal 執行個體的保留。同樣地，您可以使用相同執行個體系列內的 metal 大小，修改具有裸機執行個體之外的執行個體的保留。一般而言，裸機執行個體的大小與相同執行個體系列中的最大可用執行個體大小相同。例如，i3.metal 執行個體的大小與 i3.16xlarge 執行個體相同，因此有相同的正規化因素。

下表說明具有裸機執行個體之執行個體系列中，裸機執行個體大小的正規化因素。與其他執行個體大小不同的是，metal 執行個體的正規化因素取決於執行個體系列。

執行個體大小	正規化因素
a1.metal	32
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256

執行個體大小	正規化因素
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

例如，i3.metal 執行個體的正規化因素為 128。如果您購買 i3.metal 預設租用 Amazon Linux/Unix 預留執行個體，您可以將保留分割如下：

- i3.16xlarge 的大小與 i3.metal 執行個體相同，因此正規化因素為 128 (128/1)。一個 i3.metal 執行個體的保留可以修改為一個 i3.16xlarge 執行個體。
- i3.8xlarge 的大小是 i3.metal 執行個體的一半，因此正規化因素為 64 (128/2)。一個 i3.metal 執行個體的保留可以分割成兩個 i3.8xlarge 執行個體。
- i3.4xlarge 的大小是 i3.metal 執行個體的四分之一，因此正規化因素為 32 (128/4)。一個 i3.metal 執行個體的保留可以分割成四個 i3.4xlarge 執行個體。

提交修改請求

在修改您的預留執行個體前，請確認您已詳閱適用的限制。在修改執行個體大小之前，請先計算您希望修改之原始預留的總執行個體大小使用量，並確保其符合您的新組態的總執行個體大小使用量。

Console

修改您的預留執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在 Reserved Instances (預留執行個體) 頁面上，選取一或多個要修改的預留執行個體，然後選取 Actions (動作)，Modify Reserved Instances (修改預留執行個體)。

   若您的預留執行個體並未處於作用中的狀態或無法修改，Modify 預留執行個體 (修改預留執行個體) 便會停用。
3. 修改表格的第一個項目會顯示選取 預留執行個體 的屬性，以及至少一個位於下方的目標組態。Units (單位) 資料行會顯示總執行個體大小使用量。針對每個要新增的新組態，選擇 Add (新增)。視需要修改每個組態的屬性。

   - Scope (範圍)：選擇組態是否套用到可用區域或是整個區域。
   - Availability Zone (可用區域)：選擇必要的可用區域。不適用於區域預留執行個體。

- 執行個體類型：選取必要的執行個體類型。結合的組態必須等於原始組態的執行個體大小使用量。
- Count (計數)：指定執行個體的數目。若要將 預留執行個體 分割為多個組態，請減少計數，選擇 Add (新增)，然後指定額外組態的計數。例如，如果您有一個計數為 10 的組態，您可以將其計數變更為 6，然後新增計數為 4 的組態。此程序會在新的 預留執行個體 啟動之後重試原始的 預留執行個體。

4. 選擇 Continue (繼續)。
5. 若要在完成指定您的目標組態後確認您的修改選擇，請選擇 Submit modifications (提交修改)。
6. 您可以透過查看預留執行個體畫面中的 State (狀態) 資料行，判斷您修改請求的狀態。以下為可能的狀態。

   - 作用中 (待定修改) — 原始預留執行個體的轉移狀態
   - 已淘汰 (待定修改) – 建立新的預留執行個體時，原始預留執行個體的轉移狀態
   - 淘汰 – 已成功修改並取代預留執行個體
   - 作用中 – 下列其中一項：
     - 已成功從修改請求建立新的預留執行個體
     - 修改請求失敗後的原始預留執行個體

AWS CLI

修改您的預留執行個體

使用 modify-reserved-instances 命令。您可在 JSON 檔案中提供組態詳細資訊。

```
aws ec2 modify-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 \
    --target-configurations file://configuration.json
```

取得修改請求的狀態

使用 describe-reserved-instances-modifications 命令。狀態為 `processing`、`fulfilled` 或 `failed`。

```
aws ec2 describe-reserved-instances-modifications \
    --reserved-instances-modification-ids rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687
 \
```

```
    --query ReservedInstancesModifications[].Status
```

PowerShell

修改您的預留執行個體

使用 Edit-EC2ReservedInstance cmdlet。您可在類型為 Amazon.EC2.Model.ReservedInstancesConfiguration 的物件中提供組態詳細資訊。

```
Edit-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342f5bd7c02 `
    -TargetConfiguration $configuration
```

取得修改請求的狀態

使用 Get-EC2ReservedInstancesModification cmdlet。狀態為 processing、fulfilled 或 failed。

```
Get-EC2ReservedInstancesModification `
    -ReservedInstancesModificationId rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687 | `
    Select Status
```

故障診斷修改請求

若您請求的目標組態設定為獨特，您會收到訊息，告知您請求正在處理中。此時，Amazon EC2 僅判斷您修改請求的參數為有效。您的修改請求仍然可能會在處理過程中因為不可用的容量而失敗。

在某些情況下，您可能會收到訊息，表示修改請求未完成或失敗，而非確認。請使用此類訊息中的資訊，做為重新提交另一個修改請求的出發點。確認您已在提交請求前詳閱適用的限制。

並非所有選取的預留執行個體都可進行修改處理

Amazon EC2 會識別並列出無法修改的預留執行個體。若您收到如這類的訊息，請前往 Amazon EC2 主控台中的 Reserved Instances (預留執行個體) 頁面，檢查預留執行個體的資訊。

處理您的修改請求時發生錯誤

您提交一或多個預留執行個體進行修改，但您的請求全部都無法處理。取決於您要修改的保留數目，您可能會收到不同版本的訊息。

Amazon EC2 會顯示無法處理您請求的原因。例如，您可能為一或多個您要修改的預留執行個體子集指定了相同的目標組態 (可用區域和平台的組合)。請嘗試再次提交修改請求，但請確認保留的執行個體詳細資訊確實符合，且每個要修改子集的目標組態都是唯一的。

交換 可轉換預留執行個體

您可以將一或多個可轉換預留執行個體與另一個具有不同組態的可轉換預留執行個體交換，包括執行個體系列、作業系統和租用。執行交換的次數沒有限制，只要新可轉換預留執行個體的值等於或高於您要交換的可轉換預留執行個體即可。

當您交換您的可轉換預留執行個體時，您目前保留中的執行個體數目會和涵蓋新可轉換預留執行個體組態相同或更高值的執行個體數目交換。Amazon EC2 會計算您在交換之後能接收到的預留執行個體數目。

您無法交換標準版 預留執行個體，但您可以修改它們。如需詳細資訊，請參閱 修改 預留執行個體。

內容

- 交換可轉換預留執行個體的需求
- 計算 可轉換預留執行個體 交換
- 合併 可轉換預留執行個體
- 交換 可轉換預留執行個體 的一部分
- 提交交換請求

交換可轉換預留執行個體的需求

若符合下列條件，Amazon EC2 便會處理您的交換請求。您的可轉換預留執行個體必須是：

- 作用中
- 並未擱置先前的修改請求
- 在到期前至少還剩 24 小時

適用的規定如下：

- 可轉換預留執行個體必須與另一個目前由 AWS提供的可轉換預留執行個體交換。
- 可轉換預留執行個體與特定區域相關聯，該區域在保留的期間內固定不變。您無法和位於不同區域的可轉換預留執行個體交換可轉換預留執行個體。

- 若要交換區域可轉換預留執行個體， AWS 必須為區域中的新執行個體類型有足夠的容量。
- 您只能一次為一個可轉換預留執行個體交換一或多個可轉換預留執行個體。
- 若要交換可轉換預留執行個體的一部分，您可以修改它，將它分成兩個或多個保留，然後為新的可轉換預留執行個體交換一或多個保留。如需詳細資訊，請參閱交換 可轉換預留執行個體 的一部分。如需修改預留執行個體的詳細資訊，請參閱修改 預留執行個體。
- 所有預付可轉換預留執行個體都可和部分預付可轉換預留執行個體交換，反之亦然。

> **Note**
>
> 如果交換所需的總預付款 （ 校正成本 ） 低於 0.00 美元， AWS 則 會自動在可轉換預留執行個體中提供一定數量的執行個體，以確保校正成本為 0.00 美元或以上。

> **Note**
>
> 如果新的可轉換預留執行個體的總值 （ 預付價格 + 每小時價格 * 剩餘時數 ） 小於交換的可轉換預留執行個體的總值， AWS 會自動在可轉換預留執行個體中提供執行個體的數量，以確保總值等於或高於交換的可轉換預留執行個體。

- 若要從更佳的定價獲得利益，您可以將無預付款可轉換預留執行個體交換為全預付款或部分預付款可轉換預留執行個體。
- 您無法以全預付款及部分預付款可轉換預留執行個體交換無預付款可轉換預留執行個體。
- 只有在新可轉換預留執行個體的每小時價格等於或高於交換可轉換預留執行個體的每小時價格時，您才能以無預付款可轉換預留執行個體交換另一個無預付款可轉換預留執行個體。

> **Note**
>
> 若新可轉換預留執行個體的總值 (每小時價格 * 剩餘小時數) 低於交換可轉換預留執行個體的總值， AWS 會自動在可轉換預留執行個體中給予您執行個體的數量，確保總值等於或高於交換可轉換預留執行個體的值。

- 若您交換多個具有不同過期日期的可轉換預留執行個體，新可轉換預留執行個體的過期日期將會是距離現在最遠的日期。
- 若您交換單一可轉換預留執行個體，它必須與新可轉換預留執行個體具有相同的期間 (1 年或 3 年)。若您合併多個具有不同期間長度的可轉換預留執行個體，新的可轉換預留執行個體將具有 3 年的期間。如需詳細資訊，請參閱合併 可轉換預留執行個體。

- Amazon EC2 交換可轉換預留執行個體時，會淘汰相關的預留，並將結束日期轉移到新的預留。交換後，Amazon EC2 會將舊預留的結束日期和新預留的開始日期設定為等於交換日期。例如，若您交換具有 16 個月剩餘期間的三年預留，新預留是 16 個月預留，其結束日期與您交換的可轉換預留執行個體的預留相同。

計算 可轉換預留執行個體 交換

交換可轉換預留執行個體是免費的。但是，您可能需要支付校正成本，即您擁有的原始可轉換預留執行個體與您從交換中獲得新可轉換預留執行個體之間的差異，依比例分配的預付成本。

每個可轉換預留執行個體都具有清單值。此清單值會與您要決定可從交換接收多少執行個體保留的可轉換預留執行個體清單值進行比較。

例如：您有 1 個清單值為 35 USD 的可轉換預留執行個體，希望和清單值為 10 USD 的執行個體類型交換。

```
$35/$10 = 3.5
```

您可以使用三個 10 USD 的可轉換預留執行個體交換您的可轉換預留執行個體。因為您無法購買半個保留，因此您必須購買額外的可轉換預留執行個體才能涵蓋剩餘的部分：

```
3.5 = 3 whole Convertible Reserved Instances + 1 additional Convertible Reserved
 Instance
```

第四個可轉換預留執行個體具有和其他三個可轉換預留執行個體相同的結束日期。若您要交換部分或全預付可轉換預留執行個體，您必須為第四個保留支付校正成本。若您 可轉換預留執行個體 剩餘的預付成本為 500 USD，則依比例分配，新保留通常會需要花費 600 USD。您將需要支付 100 USD 的費用。

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of old
 reservations = $100 difference
```

合併 可轉換預留執行個體

若您合併兩個或多個可轉換預留執行個體，新可轉換預留執行個體的期間必須和舊可轉換預留執行個體相同，或為舊可轉換預留執行個體中最高的值。新可轉換預留執行個體的過期日期將會是距離現在最遠的日期。

例如，您的帳戶中有下列可轉換預留執行個體：

Reserved Instance ID	術語	過期日期
aaaa1111	1 年	2018-12-31
bbbb2222	1 年	2018-07-31
cccc3333	3 年	2018-06-30
dddd4444	3 年	2019-12-31

- 您可以合併 aaaa1111 和 bbbb2222，並以它們交換 1 年的可轉換預留執行個體。您無法以它們交換 3 年的可轉換預留執行個體。新可轉換預留執行個體的過期日期為 2018-12-31。
- 您可以合併 bbbb2222 和 cccc3333，並以它們交換 3 年的可轉換預留執行個體。您無法以它們交換 1 年的可轉換預留執行個體。新可轉換預留執行個體的過期日期為 2018-07-31。
- 您可以合併 cccc3333 和 dddd4444，並以它們交換 3 年的可轉換預留執行個體。您無法以它們交換 1 年的可轉換預留執行個體。新可轉換預留執行個體的過期日期為 2019-12-31。

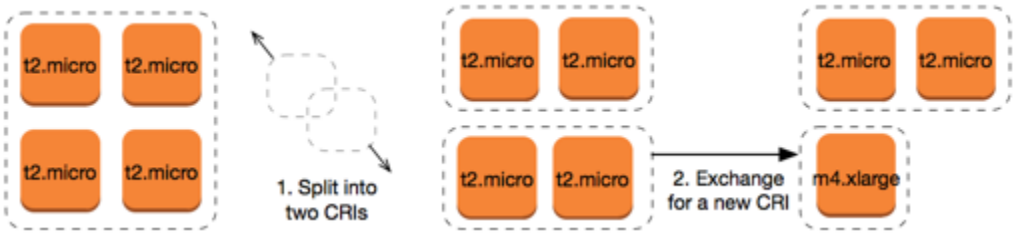
交換 可轉換預留執行個體 的一部分

您可以使用修改程序將您的可轉換預留執行個體分割為較小的保留，然後以一或多個新保留交換新的可轉換預留執行個體。下列範例示範執行此作業的方式。

Example範例：具有多個執行個體的可轉換預留執行個體

在此範例中，您有一個保留中有四個執行個體的 t2.micro 可轉換預留執行個體。若要以兩個 t2.micro 執行個體交換一個 m4.xlarge 執行個體：

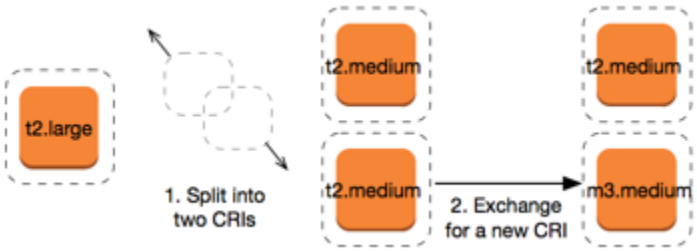
1. 修改 t2.micro 可轉換預留執行個體，方式為將其分割為兩個 t2.micro 可轉換預留執行個體，各自具有兩個執行個體。
2. 以其中一個新的 t2.micro 可轉換預留執行個體交換 m4.xlarge 可轉換預留執行個體。



Example範例：具有單一執行個體的可轉換預留執行個體

在此範例中，您有一個 t2.large 可轉換預留執行個體。若要將其變更為較小的 t2.medium 執行個體和 m3.medium 執行個體：

1. 修改 t2.large 可轉換預留執行個體，方式為將其分割為兩個 t2.medium 可轉換預留執行個體。執行個體大小使用量與兩個 t2.large 執行個體相同的單一 t2.medium 執行個體。
2. 以其中一個新的 t2.medium 可轉換預留執行個體交換 m3.medium 可轉換預留執行個體。



如需詳細資訊，請參閱支援執行個體大小的修改及提交交換請求。

提交交換請求

您可以交換可轉換預留執行個體。交換的預留執行個體已淘汰。

Console

交換可轉換預留執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選取 Reserved Instances (預留執行個體)、選取要交換的可轉換預留執行個體，然後選取 Actions (動作)，Exchange 預留執行個體 (交換預留執行個體)。
3. 選取所需組態的屬性，然後選取 Find offering (尋找優惠)。
4. 選取新的 可轉換預留執行個體。在畫面底部，您可以檢視兌換所收到的 預留執行個體 數量，以及任何額外費用。
5. 當您選取符合您需求的 可轉換預留執行個體 時，請選取 Review (檢閱)。
6. 選擇 Exchange (交換)，然後選擇 Close (關閉)。

AWS CLI

交換可轉換預留執行個體

1. 使用 describe-reserved-instances-offerings 命令，尋找符合需求的新可轉換預留執行個體。
2. 使用 get-reserved-instances-exchange-quote 命令，取得交換的報價。這包含您可以從交換中取得的預留執行個體數目，以及交換的校正成本：
3. 使用 accept-reserved-instances-exchange-quote 命令，以執行交換。

PowerShell

交換可轉換預留執行個體

1. 使用 Get-EC2ReservedInstancesOffering cmdlet，尋找符合您需求的新可轉換預留執行個體。
2. 使用 GetEC2-ReservedInstancesExchangeQuote cmdlet，取得交換的報價。這包含您可以從交換中取得的預留執行個體數目，以及交換的校正成本：
3. 使用 Approve-EC2ReservedInstancesExchangeQuote cmdlet，以執行交換

預留執行個體配額

您可以每月購買新的預留執行個體。您每月可購買的新預留執行個體數量取決於您的每月配額，如下所示：

配額說明	預設配額
新的區域預留執行個體	每月每區域 20 個
新的地區預留執行個體	每月每可用區域 20 個

例如，在具有 3 個可用區域的區域中，預設配額為每月 80 個新的預留執行個體 (計算方式如下)：

- 該區域的 20 個區域預留執行個體
- 加上 60 個地區預留執行個體 (三個可用區域各 20 個)

`running` 狀態中的執行個體會計入您的配額。處於 `pending`、`stopping`、`stopped` 和 `hibernated` 狀態的執行個體不會計入您的配額。

檢視您已購買的預留執行個體數

您購買的預留執行個體數量由 Instance count (執行個體計數) 欄位 (主控台) 或 `InstanceCount` 參數 (AWS CLI) 來指示。當您購買新的預留執行個體時，會根據執行個體總數來測量配額。例如，如果您購買執行個體數量為 10 的單一預留執行個體組態，則該購買所佔配額將為 10，而非 1。

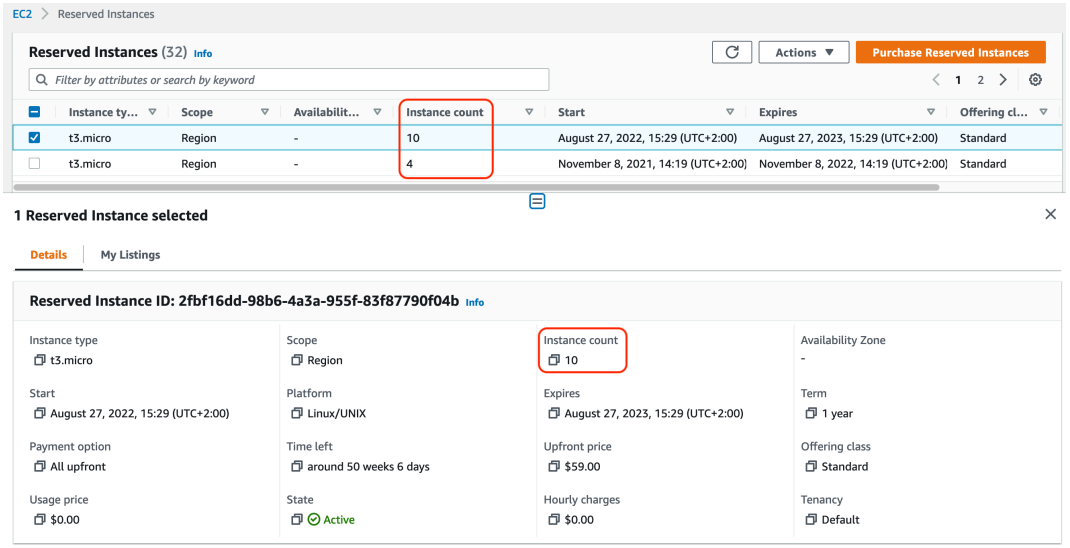
可以使用 Amazon EC2 或 AWS CLI來檢視您已購買了多少個預留執行個體。

Console

若要檢視您已購買的預留執行個體數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留執行個體)。
3. 從資料表中選取預留執行個體組態，然後勾選 Instance count (執行個體計數) 欄位。

   在下列螢幕擷取畫面中，所選行表示 `t3.micro` 執行個體類型的單一預留執行個體組態。資料表檢視中的 Instance count (執行個體計數) 欄和詳細資訊檢視中的 Instance count (執行個體計數) 欄位指示此組態有 10 個預留執行個體。



AWS CLI

若要檢視您已購買的預留執行個體數

使用 describe-reserved-instances 命令並指定預留執行個體組態的 ID。

```
aws ec2 describe-reserved-instances \
```

```
    --reserved-instances-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --output table
```

以下為範例輸出。InstanceCount 欄位指示此組態有 10 個預留執行個體。

```
-------------------------------------------------------------------
|                    DescribeReservedInstances                    |
+-----------------------------------------------------------------+
||                       ReservedInstances                       ||
|+----------------------+----------------------------------------+|
||  CurrencyCode        |  USD                                   ||
||  Duration            |  31536000                              ||
||  End                 |  2023-08-27T13:29:44+00:00             ||
||  FixedPrice          |  59.0                                  ||
||  InstanceCount       |  10                                    ||
||  InstanceTenancy     |  default                               ||
||  InstanceType        |  t3.micro                              ||
||  OfferingClass       |  standard                              ||
||  OfferingType        |  All Upfront                           ||
||  ProductDescription  |  Linux/UNIX                            ||
||  ReservedInstancesId |  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ||
||  Scope               |  Region                                ||
||  Start               |  2022-08-27T13:29:45.938000+00:00      ||
||  State               |  active                                ||
||  UsagePrice          |  0.0                                   ||
|+----------------------+----------------------------------------+|
|||                       RecurringCharges                      |||
||+---------------------------------+--------------------------+||
|||  Amount                         |  0.0                     |||
|||  Frequency                      |  Hourly                  |||
||+---------------------------------+--------------------------+||
```

PowerShell

若要檢視您已購買的預留執行個體數

使用 Get-EC2ReservedInstance Cmdlet 並指定預留執行個體組態的 ID。

```
Get-EC2ReservedInstance -ReservedInstancesId a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

以下為範例輸出。InstanceCount 欄位指示此組態有 10 個預留執行個體。

```
AvailabilityZone    :
```

```
CurrencyCode        : USD
Duration            : 31536000
End                 : 1/12/2017 8:57:08 PM
FixedPrice          : 0
InstanceCount       : 10
InstanceTenancy     : default
InstanceType        : t3.medium
OfferingClass       : standard
OfferingType        : All Upfront
ProductDescription  : Windows
RecurringCharges    : {}
ReservedInstancesId : a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
Scope               : Region
Start               : 10/12/2016 4:00:00 PM
State               : active
Tags                : {}
UsagePrice          : 0
```

考量事項

區域預留執行個體會將折扣套用至執行中隨需執行個體。預設隨需執行個體限制為 20。您無法透過購買區域預留執行個體來超出執行中隨需執行個體限制。例如，如果您已有 20 個執行中隨需執行個體，而且您購買 20 個區域預留執行個體，則 20 個區域預留執行個體是用來將折扣套用至 20 個執行中隨需執行個體。如果您購買更多的區域預留執行個體，將無法啟動更多的執行個體，因為您已達到隨需執行個體限制。

在購買地區預留執行個體之前，請確認您的隨需執行個體限制符合或超過您想要擁有的地區預留執行個體數。如有必要，請務必先請求調高您的隨需執行個體限制，再購買更多地區預留執行個體。

區域性預留執行個體 (針對特定可用區域購買的預留執行個體) 會提供容量保留以及折扣。您可以透過購買區域性預留執行個體來超出執行中隨需執行個體限制。例如，如果您已有 20 個執行中隨需執行個體，而且購買 20 個區域性預留執行個體，則您可以另外啟動 20 個符合區域性預留執行個體之規格的隨需執行個體，如此總共有 40 個執行中執行個體。

檢視預留執行個體配額並請求增加配額

Amazon EC2 主控台提供了配額資訊。您還可以請求增加您的配額。如需詳細資訊，請參閱檢視您目前的配額及請求增加。

Spot 執行個體

Spot 執行個體是使用備用 EC2容量的執行個體，可在低於隨需價格時使用它。競價型執行個體可讓您以相當優惠的折扣，來請求未使用的 EC2 執行個體，進而大幅降低 Amazon EC2 成本。Spot 執行個體的每小時價格稱為 Spot 價格。各可用區域中各執行個體類型的 Spot 價格由 Amazon EC2 決定，並會根據 競價型執行個體的長期供需情況逐漸調整。只要有容量可用，您的 Spot 執行個體便會執行。

如能彈性決定應用程式執行的時機以及應用程式是否可以中斷，則 競價型執行個體會是一項具成本效益的選擇。例如，競價型執行個體非常適合用於資料分析、批次處理任務、背景處理和選擇性的任務。如需詳細資訊，請參閱 Amazon EC2 競價型執行個體。

如需 EC2 執行個體不同購買選項的比較，請參閱 Amazon EC2 帳單與購買選項。

概念

在開始使用Spot 執行個體之前，您應該先熟悉以下概念：

- Spot 容量集區 – 一組具有相同執行個體類型 (例如 `m5.large`) 和可用區域的未使用 EC2 執行個體。
- Spot 價格 – Spot 執行個體目前的每小時價格。
- Spot 執行個體請求 – 請求 Spot 執行個體。當容量可用時，Amazon EC2 會履行您的請求。Spot 執行個體請求為一次性或持久性。與請求相關聯的 Spot 執行個體中斷之後，Amazon EC2 會自動重新提交持續的 Spot 執行個體請求。
- EC2 執行個體重新平衡建議 - Amazon EC2 發出執行個體重新平衡建議訊號，以通知您 Spot 執行個體的中斷風險升高。此訊號讓您有機會在現有或新的 Spot 執行個體上主動重新平衡工作負載，而無需等待兩分鐘的 Spot 執行個體中斷通知。
- Spot 執行個體中斷 – 當 Amazon EC2 需要取回容量，Amazon EC2 會終止、停止 Spot 執行個體或將其休眠。Amazon EC2 會提供 Spot 執行個體中斷通知，在執行個體中斷前會向執行個體發出兩分鐘的警告。

競價型執行個體與隨需執行個體之間的差異

下表列出了 Spot 執行個體與隨需執行個體之間的主要差異。

	Spot Instances	On-Demand Instances
啟動時間		

	Spot Instances	On-Demand Instances
	只有當 Spot 執行個體請求正在作用中，而且有可用的容量時，才能立即啟動。	只有在進行手動啟動請求，而且有可用的容量時，才能立即啟動。
可用容量	如果沒有可用的容量，則 Spot 執行個體請求會繼續自動發出啟動請求，直到有可用的容量出現。	如果在您發出啟動請求時沒有可用的容量，會出現容量不足錯誤 (ICE)。
每小時價格	Spot 執行個體的每小時價格會隨長期的供給和需求而有所不同。	隨需執行個體 的每小時價格為靜態定價。
重新平衡建議	執行個體的中斷風險升高時，Amazon EC2 為執行中的 Spot 執行個體發出的訊號。	您可以決定中斷 (停止、休眠或終止) 隨需執行個體 的時間點。
執行個體中斷	您可以停止和啟動 Amazon EBS 支援的 Spot 執行個體。此外，如果不再有可用的容量，Amazon EC2 還可以中斷個別的 Spot 執行個體。	您可以決定中斷 (停止、休眠或終止) 隨需執行個體 的時間點。

定價與節省

您會為 競價型執行個體支付 Spot 價格，該價格由 Amazon EC2 決定，並會根據 競價型執行個體的長期供需情況逐漸調整。在您終止 Spot 執行個體前、不再有可用的容量前，或是 Amazon EC2 Auto Scaling 群組在縮減時終止 Spot 執行個體前，您的 Spot 執行個體會持續執行。

如果您或 Amazon EC2 中斷執行中的 Spot 執行個體，則會按使用秒數或整整一小時向您收費，也可能免費，根據使用的作業系統和中斷 Spot 執行個體的人而定。如需詳細資訊，請參閱針對已中斷的 競價型執行個體收費。

Savings Plans 不涵蓋 Spot 執行個體。如果您有 Savings Plans，該計劃不會在您使用 Spot 執行個體時所享有的節省上提供額外的折扣。此外，您在 Spot 執行個體上的支出不會計入 Compute Savings Plans 中的承諾用量。

檢視價格

若要檢視每個 AWS 區域 和 執行個體類型的目前 （每五分鐘更新一次） 最低 Spot 價格，請參閱 Amazon EC2 Spot 執行個體定價頁面。

若要檢視過去三個月的 Spot 價格歷史，請使用 Amazon EC2 主控台或 describe-spot-price-history 命令。如需詳細資訊，請參閱檢視 Spot 執行個體定價歷史記錄。

我們獨立將可用區域對應到每個可用區域的代碼 AWS 帳戶。如此，不同帳戶之間的相同可用區域代碼 (例如 `us-west-2a`)，就會得到不同的結果。

檢視節省

您可以檢視針對單一 Spot 機群或所有 Spot 執行個體使用 Spot 執行個體所節省的成本。您可以檢視前一個小時或前三天的節省，也可以檢視每 vCPU 小時和每記憶體 (GiB) 小時的平均成本。節省是估計值，因為不含用量的帳單調整，所以會與實際的節省有所出入。如需檢視節省資訊的詳細資訊，請參閱購買 競價型執行個體省下的成本。

檢視帳單

您的帳單會提供您服務用量的詳細資訊。如需詳細資訊，請參閱 AWS Billing 使用者指南中的檢視您的帳單。

Amazon EC2 Spot 的最佳實務

相較於隨需價格，Amazon EC2 AWS 雲端 可透過 Spot 執行個體存取 中的備用 EC2 運算容量，最多可節省 90%。隨需執行個體與 Spot 執行個體之間唯一的差別在於，若 Amazon EC2 需要取回容量時，Spot 執行個體可以由 Amazon EC2 利用兩分鐘的通知來中斷。為了確保 Spot 執行個體最佳體驗，了解並套用最佳實務以供使用至關重要。

競價型執行個體 建議用於無狀態、容錯、靈活的應用程式。例如，競價型執行個體 適用於大數據、容器化工作負載、CI/CD、無狀態 Web 伺服器、高效能運算 (HPC) 和轉譯工作負載。

執行時，競價型執行個體 與 隨需執行個體 完全相同。但是，Spot 並無法保證您可以保持執行中的執行個體有足夠時間來完成工作負載。Spot 也無法保證您可以立即取得所尋找執行個體的可用性，也無法保證您一律可以取得所請求的彙總容量。此外，Spot 執行個體中斷和容量可能會隨著時間而改變，因為 Spot 執行個體可用性會因供給和需求而有所不同，而過去的效能並不能保證未來的結果。

競價型執行個體 不適合執行個體節點之間不靈活、可設定狀態、錯誤不容忍或緊密結合的工作負載。不建議將 Spot 執行個體用於目標容量偶爾無法完全可用的工作負載。儘管遵循 Spot 最佳實務 (例如對執行個體類型和可用區域保持靈活性) 可提供高可用性的最佳機會，但由於隨需執行個體需求激增可能會中斷 Spot 執行個體的工作負載，因此無法保證可以使用資源容量。

強烈不建議您對這些工作負載使用 Spot 執行個體，或嘗試容錯移轉至隨需執行個體以處理中斷或資源不可用的情況。容錯移轉至隨需執行個體可能會在不經意間導致其他 Spot 執行個體中斷。此外，如果執行個體類型和可用區域組合的 Spot 執行個體中斷，您可能難以取得具有相同組合的隨需執行個體。

無論您是經驗豐富的 Spot 使用者或 Spot 執行個體的新使用者，如果您目前遇到 Spot 執行個體中斷或可用性問題，建議您遵循這些最佳做法，以獲得使用 Spot 服務的最佳體驗。

Spot 最佳實務

- 針對中斷準備個別執行個體
- 對執行個體類型和可用區域具有彈性
- 使用屬性型執行個體類型選取範圍
- 使用 Spot 配置分數來識別最佳區域和可用區域
- 使用 EC2 Auto Scaling 群組或 EC2 Fleet 來管理彙總容量
- 使用價格和容量最佳化分配策略
- 使用整合 AWS 服務來管理您的 Spot 執行個體
- 使用哪種 Spot 請求方法最好？

針對中斷準備個別執行個體

從容處理 Spot 執行個體中斷的最佳方法，就是建立容錯的應用程式架構。若要達到此目的，您可以利用 EC2 執行個體重新平衡建議和 Spot 執行個體中斷通知。

EC2 執行個體重新平衡推薦是一種訊號，可在 Spot 執行個體的中斷風險升高時通知您。該訊號使您有機會在兩分鐘 Spot 執行個體中斷通知之前主動管理 Spot 執行個體。您可以決定將工作負載重新平衡至未處於提高之中斷風險的新的或現有 競價型執行個體。我們已經使用 Auto Scaling 群組和 EC2 Fleet 中的容量重新平衡功能，讓您輕鬆地使用此訊號。

Spot 執行個體中斷通知是在 Amazon EC2 中斷 Spot 執行個體的兩分鐘之前發出的警告。如果您的工作負載具有「時間彈性」，當執行個體中斷時，您可以將它們設定為停止或休眠，而非終止。Amazon EC2 會在中斷時自動停止或休眠 Spot 執行個體，並在我們有可用容量時自動繼續執行個體。

建議您在 Amazon EventBridge 中建立擷取重新平衡建議和中斷通知的規則，然後觸發工作負載進度的檢查點，或從容處理中斷。如需詳細資訊，請參閱監控重新平衡建議訊號。如需如何建立和使用事件規則的詳細範例，請參閱利用 Amazon EC2 Spot 執行個體中斷通知。

如需詳細資訊，請參閱EC2 執行個體重新平衡建議及Spot 執行個體中斷。

對執行個體類型和可用區域具有彈性

Spot 容量集區是一組未使用的 EC2 執行個體，具有相同執行個體類型 (例如 m5.large) 和可用區域 (例如 us-east-1a)。您應該對於請求的執行個體類型，以及可在其中部署工作負載的可用區域具有彈性。這讓 Spot 有更好的機會找到並配置您所需的運算容量。例如，不要只要求 c5.large 是否願意使用來自 c4、m5 和 m4 系列的 large。

根據您的特定需求，您可以評估哪些執行個體類型可以具彈性，以滿足您的運算需求。如果工作負載可以垂直擴展，則應在請求中包含較大的執行個體類型 (更多 vCPU 和記憶體)。如果您只能水平擴展，則應納入舊一代執行個體類型，因為隨需客戶的需求較少。

一個很好的經驗法則是針對每個工作負載的至少 10 個執行個體類型上具有彈性。此外，請確定所有可用區域均已設定為在 VPC 中使用，並針對您的工作負載選取。

使用屬性型執行個體類型選取範圍

透過屬性型執行個體類型選項，您可以為要執行的工作負載指定執行個體屬性，例如 vCPU、記憶體和儲存體。EC2 Auto Scaling 或 EC2 Fleet 會自動識別並啟動符合您指定屬性的執行個體。這樣就省去了手動選取特定執行個體類型所需的工作，該工作需要深入了解每一種執行個體類型提供的功能。

此外，屬性型執行個體類型選項可讓您在新版本執行個體類型推出時自動使用。這可確保順暢存取日益廣泛的 Spot 執行個體容量。

屬性型執行個體類型選項非常適用於其執行的執行個體類型具有彈性的工作負載和架構，例如高效能運算 (HPC) 和大數據工作負載。

如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用屬性型執行個體類型選取範圍來建立 Auto Scaling 群組以及此指南中的 指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型。

使用 Spot 配置分數來識別最佳區域和可用區域

Spot 執行個體是未使用的 EC2 容量，此容量會根據 EC2 的供需而變化。因此，您不一定會在特定時間和特定位置中取得所需的確切 Spot 容量。若要緩解此不可預測性，您可以使用 Spot 配置分數功能。此功能會針對容量較可能足以滿足 Spot 容量需求的區域或可用區域提供建議，而無需先在這些位置啟動 Spot 執行個體。

Spot 配置分數最適用於對執行個體類型，以及它們可使用的區域或可用區域具有彈性的工作負載。您只需要指定所需的 Spot 容量、執行個體類型需求，以及是否想要針對區域或可用區域提供建議。作為交換，您會收到每個區域或可用區域從 1 到 10 的分數，這表示在該位置順利佈建您要求的 Spot 容量的可能性。分數 10 表示您的 Spot 請求很有可能成功。

請注意，Spot 配置分數是時間點建議，因為容量可能會隨著時間而有所不同。其不保證可用容量，也無法預測中斷風險。

您可以在 Amazon EC2 主控台 AWS CLI或 SDK 中使用 Spot 配置分數功能。如需詳細資訊，請參閱Spot 配置分數。

使用 EC2 Auto Scaling 群組或 EC2 Fleet 來管理彙總容量

Spot 可讓您以彙總容量 (包括 vCPU、記憶體、儲存或網路輸送量單位) 的方式來思考，而不是以個別執行個體的方式來思考。Auto Scaling 群組和 EC2 Fleet 可讓您啟動和維護目標容量，並自動請求資源以取代中斷或手動終止的任何資源。設定 Auto Scaling 群組或 EC2 Fleet 時，只需要根據應用程式需求指定執行個體類型和目標容量。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南 中的 Auto Scaling 群組和本使用者指南中的 建立 EC2 Fleet。

使用價格和容量最佳化分配策略

Auto Scaling 群組中的分配策略可協助您佈建目標容量，而無需手動尋找具有備用容量的 Spot 容量集區。建議使用 `price-capacity-optimized` 策略，因為此策略會自動從最可用且價格最低的 Spot 容量集區佈建執行個體。您也可以利用 EC2 Fleet 中的 `price-capacity-optimized` 配置策略。由於您的 Spot 執行個體容量來自具有最佳容量的集區，因此可降低回收 Spot 執行個體的可能性。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南中的多種執行個體類型的配置策略和本使用者指南中的 當工作負載具有較高的中斷成本時。

使用整合 AWS 服務來管理您的 Spot 執行個體

AWS 其他服務與 Spot 整合，可降低整體運算成本，而無需管理個別執行個體或機群。我們建議您考慮適用工作負載的下列解決方案：Amazon EMR、Amazon Elastic Container Service AWS Batch、Amazon Elastic Kubernetes Service、Amazon SageMaker AI AWS Elastic Beanstalk和 Amazon GameLift Servers。若要深入了解搭配這些服務的 Spot 最佳做法，請參閱 Amazon EC2 競價型執行個體 研討會網站。

使用哪種 Spot 請求方法最好？

使用下表確定在請求 Spot 執行個體時使用哪個 API。

API	何時使用？	使用案例	我應該使用此 API 嗎？
CreateAutoScalingGroup	•	建立 Auto Scaling 群組，管理執行個體的生	是

API	何時使用？	使用案例	我應該使用此 API 嗎？
	您需要具有單一組態或混合組態的多個執行個體。 • 您希望透過可設定的 API 自動執行生命週期管理。	命週期，同時維護所需的執行個體數量。支援在指定的最小和最大限制之間水平擴展 (新增更多執行個體)。	
CreateFleet	• 您需要具有單一組態或混合組態的多個執行個體。 • 您希望自行管理您的執行個體生命週期。 • 如果您不需要自動擴展，建議您使用 `instant` 類型機群。	在單一請求中建立隨需執行個體和 Spot 執行個體的機群，並具有隨執行個體類型、AMI、可用區域或子網而異的多個啟動規格。Spot 執行個體分配策略預設為每單位 `lowest-price` ，但您可以將其變更為 `price-capacity-optimized` 、`capacity-optimized` 或 `diversified` 。	是 – 在 `instant` 模式中 (如果您不需要自動擴展)

API	何時使用？	使用案例	我應該使用此 API 嗎？
RunInstances	• 您已經在使用 RunInstances API 來啟動隨需執行個體，而且您只想透過變更單一參數來變更為啟動 Spot 執行個體。 • 您不需要具有不同執行個體類型的多個執行個體。	使用 AMI 和一個執行個體類型，啟動指定數量的執行個體。	否 – 因為 RunInstances 不允許在單一請求中使用混合執行個體類型
RequestSpotFleet	• 我們強烈反對使用 RequestSpotFleet API，因為這是一個沒有計劃投資的舊式 API。 • 如果您想要管理執行個體生命週期，請使用 CreateFleet API。 • 如果您不想要管理執行個體生命週期，請使用 CreateAutoScalingGroup API。	請勿使用。RequestSpotFleet 是沒有計劃投資的舊式 API。	否

API	何時使用？	使用案例	我應該使用此 API 嗎？
RequestSpotInstances	• 我們強烈反對使用 RequestSpotInstances API，因為這是一個沒有計劃投資的舊式 API。	請勿使用。RequestSpotInstances 是沒有計劃投資的舊式 API。	否

競價型執行個體的運作方式

若要啟動 Spot 執行個體，您可以建立 Spot 執行個體請求，或 Amazon EC2 代表您建立 Spot 執行個體請求。Spot 執行個體請求完成後，會啟動 Spot 執行個體。

您可以使用數個不同的服務來啟動 Spot 執行個體。如需詳細資訊，請參閱 Amazon EC2 Spot 執行個體入門。在本使用者指南中，我們描述了使用 EC2 啟動 Spot 執行個體的以下方法：

- 您可以使用 Amazon EC2 主控台中的啟動執行個體精靈或 run-instances 命令來建立 Spot 執行個體請求。如需詳細資訊，請參閱管理您的 Spot 執行個體。
- 您可以建立 EC2 Fleet，在其中指定所需數量的 Spot 執行個體。Amazon EC2 為 EC2 Fleet 中指定的每個 Spot 執行個體建立 Spot 執行個體請求。如需詳細資訊，請參閱建立 EC2 Fleet。
- 您可以建立 Spot Fleet 請求，在其中指定所需數量的 Spot 執行個體。Amazon EC2 為 Spot Fleet 請求中指定的每個 Spot 執行個體建立 Spot 執行個體請求。如需詳細資訊，請參閱建立 Spot 機群。

> **Note**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

如果有可用容量，您的 Spot 執行個體就會啟動。您的 Spot 執行個體會執行直到您停止或終止它，或者直到 Amazon EC2 中斷它 (稱為 Spot 執行個體中斷)。Amazon EC2 可以在 Spot 執行個體中斷時停止、終止或進入休眠。

使用 Spot 執行個體時，您必須對中斷做好準備。當 Spot 執行個體的需求增加，或是Spot 執行個體的供給減少時，Amazon EC2 可以中斷您的 Spot 執行個體。當 Amazon EC2 中斷 Spot 執行個體時，會提供 Spot 執行個體中斷通知，在 Amazon EC2 中斷執行個體前，向該執行個體發出兩分鐘的警告。您無法為 競價型執行個體啟用終止保護。如需詳細資訊，請參閱Spot 執行個體中斷。

目錄

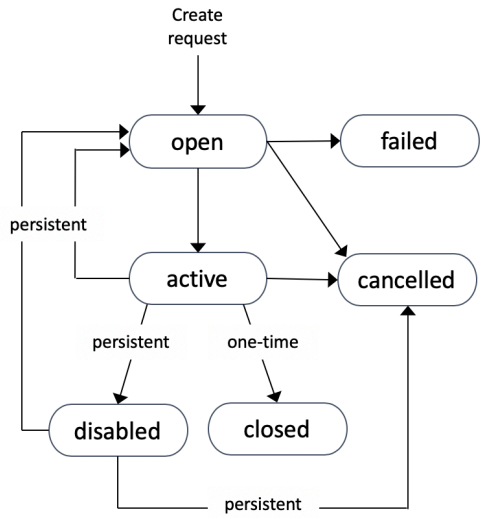
- Spot 執行個體請求狀態
- 在啟動群組中啟動 競價型執行個體
- 在可用區域群組中啟動 競價型執行個體
- 在 VPC 中啟動 競價型執行個體
- 啟動爆量效能執行個體
- 在單一租戶硬體上啟動

Spot 執行個體請求狀態

Spot 執行個體請求可以是下列任一種狀態：

- `open` – 請求正等待完成。
- `active` – 請求已完成，並具有關聯的 Spot 執行個體。
- `failed` – 請求具有一個或多個無效的參數。
- `closed` – Spot 執行個體已中斷或終止。
- `disabled` – 您已停止 Spot 執行個體。
- `cancelled` – 您已取消請求，或請求已過期。

下圖顯示了請求狀態之間的轉換。請注意，轉換取決於請求的類型 (一次性或持久性)。



在 Amazon EC2 啟動 Spot 執行個體、請求過期、或者您取消請求之前，一次性 Spot 執行個體請求會維持有效運作。如果沒有容量可用，則 Spot 執行個體會終止，而且 Spot 執行個體請求也會結束。

即使請求已完成，持久性的 Spot 執行個體請求在過期或是您取消它之前，都會繼續維持有效運作。如果沒有容量可用，則 Spot 執行個體會中斷。在執行個體中斷後，當容量可再次提供使用時，會啟動停止或恢復休眠的 Spot 執行個體。如果有容量可用，您可以停止 Spot 執行個體並將其重新啟動。如果 Spot 執行個體已終止 (不論 Spot 執行個體處於已停止或執行中狀態)，則 Spot 執行個體請求都會再次開啟，而 Amazon EC2 會啟動新的 Spot 執行個體。如需詳細資訊，請參閱 停止 Spot 執行個體、啟動 Spot 執行個體 及 終止 Spot 執行個體。

您可以透過狀態，追蹤 Spot 執行個體請求的狀態和已啟動的 Spot 執行個體的狀態。如需詳細資訊，請參閱取得 Spot 執行個體請求狀態。

在啟動群組中啟動 競價型執行個體

在 Spot 執行個體請求中指定啟動群組，告知 Amazon EC2 啟動一組 Spot 執行個體，但前提它能夠啟動其中的所有執行個體。此外，如果 Spot 服務必須終止啟動群組內的其中一個執行個體，則必須終止群組中所有的執行個體。不過，如果您終止啟動群組內的一個或多個執行個體，Amazon EC2 不會終止該啟動群組中其餘的執行個體。

雖然此選項很實用，但加上此限制可能會降低完成 Spot 執行個體請求的機率，並提高終止 Spot 執行個體的機率。例如，您的啟動群組包含多個可用區域的執行個體。如果其中一個可用區域的容量減少，且不再提供使用，則 Amazon EC2 會終止該啟動群組的所有執行個體。

如果建立另一個成功的 Spot 執行個體請求，它指定與先前的成功請求相同的 (現有的) 啟動群組，則新的執行個體會加入該啟動群組。之後，如果這個啟動群組中的執行個體遭到終止，則該啟動群組中的所有執行個體都會終止，這也包括由第一個和第二個請求啟動的執行個體。

在可用區域群組中啟動 競價型執行個體

在您的 Spot 執行個體請求中指定可用區域群組，以告知 Amazon EC2 在同一個 Availability Zone (可用區域) 中啟動一組 Spot 執行個體。Amazon EC2 不需同時中斷可用區域群組中的所有執行個體。如果 Amazon EC2 必須中斷可用區域群組中的其中一個執行個體，則其他的執行個體仍會繼續執行。

雖然此選項很實用，但加上這項限制可能會降低完成 Spot 執行個體請求的機率。

如果指定了可用區域群組，但未在 Spot 執行個體請求中指定可用區域，則結果取決於您指定的網路。

預設 VPC

Amazon EC2 會使用指定子網適用的可用區域。如未指定子網，它會替您選取一個可用區域及其預設子網，但不一定是最低價的區域。如果您刪除可用區域適用的預設子網，則必須指定不同的子網。

非預設 VPC

Amazon EC2 會使用指定子網適用的可用區域。

在 VPC 中啟動 競價型執行個體

您可以用為 競價型執行個體指定子網的同一種方法，來為 隨需執行個體指定子網。

- [預設 VPC] 如果希望在指定的低價可用區域中啟動 Spot 執行個體，您必須在 Spot 執行個體請求中指定對應的子網。如果未指定子網，Amazon EC2 會替您選取一個，而這個子網的可用區域可能不具有最低的 Spot 價格。
- [非預設 VPC] 您必須指定 Spot 執行個體適用的子網。

啟動爆量效能執行個體

T 執行個體類型是爆量效能執行個體。如果您使用爆量效能執行個體啟動 Spot 執行個體，而且您計劃立即使用爆量效能 Spot 執行個體，且在短時間內沒有閒置時間來累積 CPU 額度，建議您以標準模式將其啟用，以避免支付更高的成本。如果您在無限制模式下啟動高載效能 Spot 執行個體，而且 CPU 立即高載，則您將耗費高載的剩餘額度。如果您短暫使用執行個體，則您的執行個體沒有時間產生 CPU 額度來償還剩餘額度，而且當終止執行個體時，會針對剩餘額度向您收費。

只有在執行個體執行夠長時間，可為爆量產生 CPU 額度時，無限制模式才適合用於高載效能 Spot 執行個體。否則，支付剩餘額度會使爆量效能 Spot 執行個體比使用其他執行個體更為昂貴。如需詳細資訊，請參閱使用無限制模式與固定 CPU 的時機。

T2 執行個體在標準模式中設定時，可取得啟動額度。T2 執行個體是唯一可獲得啟動額度的爆量效能執行個體。啟動額度旨在透過提供足夠的運算資源來設定執行個體，創造 T2 執行個體具有生產力的初始啟動體驗。不允許重複啟動 T2 執行個體來獲得新的啟動額度。如果您需要持續的 CPU，可以賺取額度 (藉由在某些期間閒置)、為 T2 Spot 執行個體使用無限制模式，或將執行個體類型與專屬 CPU 搭配使用。

在單一租戶硬體上啟動

您可以在單一租戶的硬體上執行 Spot 執行個體。專用 Spot 執行個體實際與屬於其他 AWS 帳戶的執行個體隔離。如需詳細資訊，請參閱 Amazon EC2 專用執行個體 和 Amazon EC2 專用執行個體。

若要執行專用 Spot 執行個體，請進行下列其中一項動作：

- 在建立 Spot 執行個體請求時指定 `dedicated` 的租用。如需詳細資訊，請參閱 管理您的 Spot 執行個體。
- 使用 `dedicated` 的執行個體租用，在 VPC 中請求 Spot 執行個體。如需詳細資訊，請參閱在使用預設租用的 VPC 中啟動專用執行個體。如果使用 `default` 的執行個體租用，在 VPC 中請求 Spot 執行個體，就不能使用 `dedicated` 的租用來請求它。

除了 T 執行個體之外，所有執行個體群都支援專用 競價型執行個體。對於每個支援的執行個體系列而言，只有最大型的執行個體大小或硬體大小可支援專用 競價型執行個體。

檢視 Spot 執行個體定價歷史記錄

Spot 執行個體的價格由 Amazon EC2 制定，然後根據 Spot 執行個體容量的長期供需趨勢逐漸調整。

滿足您的 Spot 請求後，您會以目前的 Spot 價格啟動 Spot 執行個體，而不會超過隨需價格。您可以檢視過去 90 天的 Spot 價格歷史，根據執行個體類型、作業系統和可用區域來進行篩選。

如需目前的 Spot 執行個體價格，請參閱 Amazon EC2 Spot 執行個體定價。

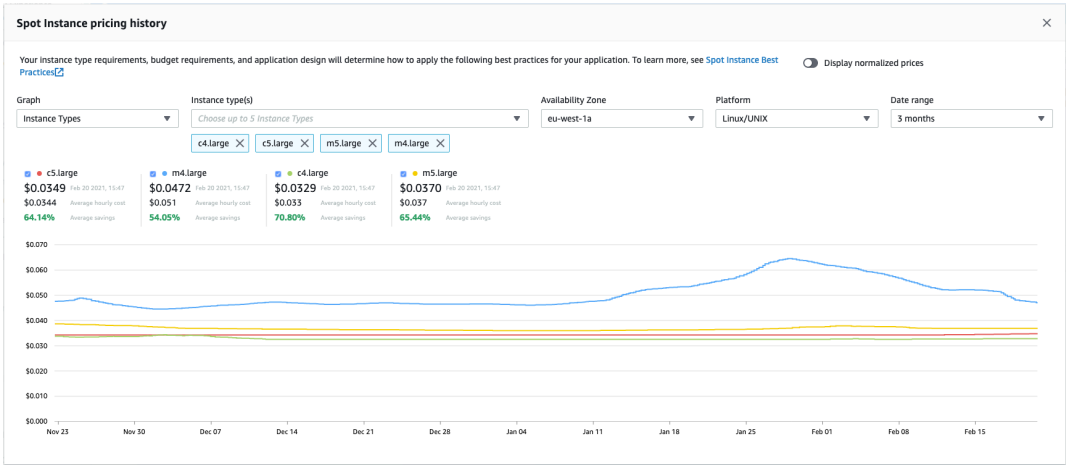
Console

檢視 Spot 價格歷史記錄

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選擇定價歷史。
4. 在 Graph (圖表) 中，選擇依 Availability Zones (可用區域) 或 Instance Types (執行個體類型) 來比較價格歷史記錄。

- 若您選擇可用區域，則選擇執行個體類型、作業系統 (平台)，以及要檢視價格歷史記錄的日期範圍。
- 若您選擇執行個體類型，則選擇最多五種執行個體類型、可用區域、作業系統 (平台)，以及要檢視價格歷史記錄的日期範圍。

下列螢幕擷取畫面顯示不同執行個體類型的價格比較。



5. 將您的指標懸停至圖表上方，可顯示所選取日期範圍中特定時間點的價格。價格顯示在圖表上方的資訊區塊中。顯示於頂列的價格會顯示指定日期的价格。顯示於第二列的價格會顯示所選日期範圍的平均價格。
6. 若要顯示每個 vCPU 的價格，請開啟 Display normalized prices (顯示標準化價格)。若要顯示該執行個體類型的價格，請關閉Display normalized prices (顯示標準化價格)。

AWS CLI

檢視 Spot 價格歷史記錄

使用下列 describe-spot-price-history 命令。

```
aws ec2 describe-spot-price-history \
    --instance-types c6i.xlarge \
    --product-descriptions "Linux/UNIX" \
    --start-time 2025-04-01T00:00:00 \
    --end-time 2025-04-02T00:00:0
```

PowerShell

檢視 Spot 價格歷史記錄

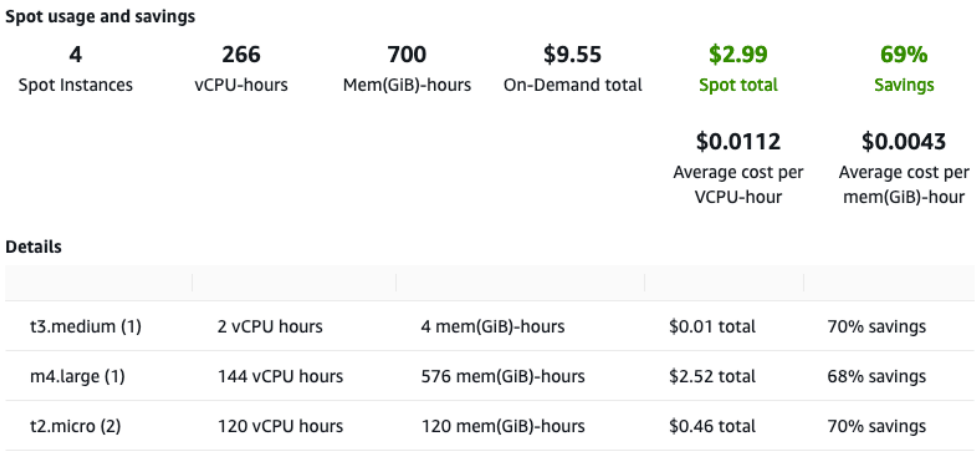
使用 Get-EC2SpotPriceHistory cmdlet。

```
Get-EC2SpotPriceHistory `
    -InstanceType c6i.xlarge `
    -ProductDescription "Linux/UNIX" `
    -UtcStartTime 2025-04-01T00:00:00 `
    -UtcEndTime 2025-04-02T00:00:0
```

購買 競價型執行個體省下的成本

您可檢視每個叢集之 競價型執行個體或所有執行中 競價型執行個體的用量和節省資訊。每個叢集的用量和節省資訊包括叢集啟動及終止的所有執行個體。您可檢視前一小時或前三天的資訊。

Savings (節省) 區段中的下列螢幕擷取畫面顯示 Spot 用量和 Spot Fleet 的節省資訊。



您可以檢視下列用量和節省資訊：

- Spot 執行個體— Spot Fleet 啟動和終止的 Spot 執行個體數量。檢視節省摘要時，數字代表所有正在執行的 競價型執行個體。
- vCPU-hours (vCPU 時數) – 所選時段內所有 競價型執行個體使用的 vCPU 時數。
- Mem(GiB)-hours (記憶體 (GiB) 時數) – 所選時段內所有 競價型執行個體使用的 GiB 時數。
- On-Demand total (隨需總額) – 如果在所選時段內啟動這些執行個體做為隨需執行個體使用，所要支付的總金額。
- Spot total (Spot 總額) – 所選時段要支付的總金額。
- Savings (節省) – 不支付隨需價格的節省百分比。

- Average cost per vCPU-hour (vCPU 每小時的平均成本) – 所有 競價型執行個體在選取時段內使用 vCPU 的每小時平均成本，計算方式如下：vCPU 每小時的平均成本 = Spot 總額 / vCPU 時數。
- Average cost per mem(GiB)-hour (記憶體 (GiB) 每小時的平均成本) – 所有 競價型執行個體在選取時段內使用 GiB 的每小時平均成本，計算方式如下：記憶體 (GiB) 每小時的平均成本 = Spot 總額 / 記憶體 (GiB) 時數。
- Details (詳細資訊) 表 – 構成 Spot Fleet 的不同執行個體類型 (每種執行個體類型的執行個體數量所佔百分比)。檢視節省摘要時，這些項目構成您所有正在執行的 競價型執行個體。

您只能使用 Amazon EC2 主控台檢視節省資訊。

> **Note**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

若要檢視 Spot Fleet 的節省資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上，選擇 Spot Requests (Spot 請求)。
3. 選取 Spot Fleet 請求的 ID 並捲動至 Savings (節省) 區段。

   或者，選取 Spot 機群請求 ID 旁邊的核取方塊，然後選取節省成本索引標籤。
4. 根據預設，頁面會顯示前三天的用量和節省資訊。您可選擇 last hour (前一小時) 或 last three days (前三天)。若為啟動不到一小時的 Spot Fleets，頁面顯示一小時的預估節省。

檢視所有正在執行的 競價型執行個體節省資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上，選擇 Spot Requests (Spot 請求)。
3. 選擇 Savings Summary (節省摘要)。

建立 Spot 執行個體請求

若要使用 Spot 執行個體，您可以建立 Spot 執行個體請求，其中包含所需的執行個體數量、執行個體類型，以及 Availability Zone (可用區域)。如果容量可用，Amazon EC2 會即刻履行您的請求。否則，Amazon EC2 會等到您的請求可以完成，或是您取消請求。

您可以使用 Amazon EC2 主控台中的啟動執行個體精靈或 run-instances 命令以您可以啟動隨需執行個體的相同方式請求 Spot 執行個體。僅出於下列原因，建議使用此方法：

- 您已經在使用啟動執行個體精靈或 run-instances 命令來啟動隨需執行個體，而且您只想透過變更單一參數來變更為啟動 Spot 執行個體。
- 您不需要具有不同執行個體類型的多個執行個體。

通常不建議在啟動 Spot 執行個體時使用此方法，因為您無法指定多個執行個體類型，並且無法在同一請求中同時啟動 Spot 執行個體和隨需執行個體。如需有關啟動 Spot 執行個體的慣用方法，其中包括啟動含有使用多個執行個體類型的 Spot 執行個體和隨需執行個體的機群，請參閱使用哪種 Spot 請求方法最好？

如果您一次請求多個 Spot 執行個體，Amazon EC2 會建立不同的 Spot 執行個體請求，如此您就可以單獨追蹤每個請求的狀態。如需追蹤 Spot 執行個體請求的詳細資訊，請參閱 取得 Spot 執行個體請求狀態。

Console

建立 Spot 執行個體請求

步驟 1–9 與您用於啟動隨需執行個體的步驟相同。在步驟 10 中，您可以設定 Spot 執行個體請求。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在螢幕上方的導覽列上，選取「區域」。
3. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. (選用) 在 Name and tags (名稱和標籤) 下，您可以命名執行個體，並標記 Spot 執行個體請求、執行個體、磁碟區和彈性圖形。如需標籤的相關資訊，請參閱標記您的 Amazon EC2 資源。

   a. 對於 Name (名稱)，輸入執行個體的描述性名稱。

      執行個體名稱是一個標籤，其中鍵是 Name (名稱)，而值是您指定的名稱。如果您未指定名稱，則可以透過其 ID 來標識執行個體，該 ID 將在您啟動執行個體時自動產生。

b. 若要標記 Spot 執行個體請求、執行個體、磁碟區和彈性圖形，請選擇 Add additional tags (新增其他標籤)。選取 Add tag (新增標籤)，然後輸入鍵和值，然後選取要標記的資源類型。為每個要新增的其他標籤重新選擇 Add tag (新增標籤)。

5. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，選取執行個體的作業系統，然後選取 AMI。如需詳細資訊，請參閱應用程式和作業系統映像 (Amazon Machine Image)。

6. 在 Instance type (執行個體類型) 下，選取符合您執行個體硬體組態和大小要求的執行個體類型。如需詳細資訊，請參閱執行個體類型。

7. 在 Key pair (login) (金鑰對 (登入))，選擇現有的金鑰對，或選擇 Create new key pair (建立新的金鑰對) 以建立新的金鑰對。如需詳細資訊，請參閱Amazon EC2 金鑰對和 Amazon EC2 執行個體。

> **Important**
>
> 如果您選擇 Proceed without key pair (Not recommended) (繼續而不使用金鑰對 (不建議)) 選項，您將無法連線到執行個體，除非您選擇已設定為允許使用者透過其他方式登入的 AMI。

8. 在 Network settings (網路設定) 下，使用預設設定，或視需要選擇 Edit (編輯) 以便設定網路設定。

   安全群組構成網路設定的一部分，並定義執行個體的防火牆規則。這些規則會指定應交付至您執行個體的傳入網路流量。

   如需詳細資訊，請參閱網路設定。

9. 您選取的 AMI 包含一或多個儲存體磁碟區，包含根設備磁碟區。在 Configure storage (設定儲存) 下，您可以透過選擇 Add new volume (新增新磁碟區)，指定要連接到執行個體的額外磁碟區。如需詳細資訊，請參閱設定儲存。

10. 在 Advanced details (進階詳細資訊)，請設定 Spot 執行個體請求，如下所示：

    a. 在購買選項下方，選取請求 Spot 執行個體核取方塊。

    b. 您可以保留 Spot 執行個體請求的預設組態，也可以選擇右側的 Customize (自訂) 為您的 Spot 執行個體請求指定自訂設定。

       當您選擇 Customize (自訂) 時，會出現下列欄位。

i. Maximum price (最高價格)：您可以採用 Spot 價格 (以隨需價格為上限) 請求 Spot 執行個體，也可以指定您願意支付的最高金額。

> ⚠ Warning
>
> 如果您指定最高價格，您的執行個體將會比選擇無最高價時更頻繁地遭到中斷。
> 若您指定最高價格，它必須超過 0.001 USD。指定低於 0.001 USD 的值會導致啟動失敗。

- No maximum price (無最高價格)：您的 Spot 執行個體將以目前的 Spot 價格啟動。價格絕不會超過隨需價格。(建議)
- Set your maximum price (per instance/hour) (設定您的最高價格 (每個執行個體/小時))：您可以指定您願意支付的最高金額。
  - 如果您指定的最高價小於目前的 Spot 價格，則不會啟動 Spot 執行個體。
  - 如果您指定的最高價格大於目前的 Spot 價格，Spot 執行個體將會啟動並按目前的 Spot 價格收費。Spot 執行個體執行後，如果 Spot 價格超過您的最高價格，Amazon EC2 將中斷您的 Spot 執行個體。
  - 無論您指定的最高價格如何，都將始終按當前 Spot 價格向您收取費用。

若要回顧 Spot 價格的趨勢，請參閱 檢視 Spot 執行個體定價歷史記錄。

ii. Request type (請求類型)：您選擇的 Spot 執行個體請求類型會決定 Spot 執行個體中斷時會發生什麼狀況。

- One-time (一次性)：Amazon EC2 會為您的 Spot 執行個體發出一次性請求。如果您的 Spot 執行個體中斷，即不會重新提交請求。
- Persistent request (持久性請求)：Amazon EC2 會為您的 Spot 執行個體發出持久性請求。如果您的 Spot 執行個體中斷，系統會重新提交請求以補充中斷的 Spot 執行個體。

若未指定數值，則預設值為一次性請求。

iii. Valid to (有效至)：持久性 Spot 執行個體請求的到期日期。

一次性請求不支援此欄位。一次性請求在請求中的所有執行個體均啟動，或是您取消請求之前，都會繼續維持有效運作。

- No request expiry date (無請求過期日期)：請求會維持在有效狀態，直到您將其取消為止。
- Set your request expiry date (設定您的請求過期日期)：持久性請求在您指定的日期之前會維持在有效狀態，或者直到您將其取消為止。

iv. Interruption behavior (中斷行為)：您選擇的行為會決定 Spot 執行個體中斷時會發生什麼狀況。

- 針對持久性請求，有效值為 Stop (停止) 和 Hibernate (休眠)。當執行個體停止時，將收取 EBS 磁碟區儲存費用。

> **Note**
>
> Spot 執行個體現在使用與隨需執行個體相同的休眠功能。若要啟用休眠功能，您可以在此選擇休眠，也可以從停止 - 休眠行為欄位 (顯示於啟動執行個體精靈下半部) 中選擇啟用。如需休眠必要條件，請參閱EC2 執行個體休眠的先決條件。

- 針對一次性請求，有效值僅有 Terminate (終止)。

若未指定值，則預設值為 Terminate (終止)，這對於持久性 Spot 執行個體請求無效。如果您保留預設值，並嘗試啟動持久性 Spot 執行個體請求，則會出現錯誤。

如需詳細資訊，請參閱Spot 執行個體中斷的行為。

11. 在 Summary (摘要) 面板上，針對 Number of instances (執行個體的數量)，輸入要啟動的執行個體數量。

> **Note**
>
> Amazon EC2 為每個 Spot 執行個體建立一個單獨的請求。

12. 在 Summary (摘要) 面板上，檢閱執行個體的詳細資訊，並進行任何必要的變更。提交 Spot 執行個體請求後，您就無法變更請求的參數。您可以直接導覽至啟動執行個體精靈中的某個區段，方法是在 Summary (摘要) 面板中選擇其連結。如需詳細資訊，請參閱摘要。
13. 當您準備好啟動執行個體時，請選擇 Launch instance (啟動執行個體)。

如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

AWS CLI

若要使用 run-instances 建立 Spot 執行個體請求

使用 run-instances 命令，並在 `--instance-market-options` 參數中指定 Spot 執行個體選項。

```
--instance-market-options file://spot-options.json
```

以下是在 JSON 檔案中指定的資料結構。您也可以指 `ValidUntil` 和 `InstanceInterruptionBehavior`。如果您沒有在資料結構中指定欄位，則會使用預設值。

以下範例會建立 `persistent` 請求。

```
{
  "MarketType": "spot",
  "SpotOptions": {
    "SpotInstanceType": "persistent"
  }
}
```

使用 request-spot-instances 建立 Spot 執行個體請求

> **Note**
>
> 我們強烈反對使用 request-spot-instances 命令來請求 Spot 執行個體，因為這是一個沒有計劃投資的舊式 API。如需詳細資訊，請參閱使用哪種 Spot 請求方法最好？

使用 request-spot-instances 命令來建立一次性請求。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json
```

使用 request-spot-instances 命令來建立一個永久的請求。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "persistent" \
    --launch-specification file://specification.json
```

例如，啟動規格檔案以搭配這些命令使用，請參閱Spot 執行個體請求啟動規格範例。如果您從 Spot 請求主控台下載啟動規格檔案，則必須改用 request-spot-fleet 命令 (Spot 請求主控台使用 Spot 機群指定 Spot 執行個體請求)。

PowerShell

建立 Spot 執行個體請求

使用 New-EC2Instance cmdlet，並使用 `-InstanceMarketOption` 參數來指定 Spot 執行個體選項。

```
-InstanceMarketOptions $marketOptions
```

如下所示，建立 Spot 執行個體選項的資料結構。

```
$spotOptions = New-Object Amazon.EC2.Model.SpotMarketOptions
$spotOptions.SpotInstanceType="persistent"
$marketOptions = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketOptions.MarketType = "spot"
$marketOptions.SpotOptions = $spotOptions
```

Spot 執行個體請求啟動規格範例

下列範例顯示您可搭配 request-spot-instances 命令使用的啟動組態，以建立 Spot 執行個體請求。如需詳細資訊，請參閱管理您的 Spot 執行個體。

> **Important**
>
> 我們強烈反對使用 request-spot-instances 命令來請求 Spot 執行個體，因為這是一個沒有計劃投資的舊式 API。如需詳細資訊，請參閱 使用哪種 Spot 請求方法最好？

範例

- 範例 1：啟動 競價型執行個體
- 範例 2：在指定的可用區域中啟動 競價型執行個體
- 範例 3：在指定的子網中啟動 競價型執行個體
- 範例 4：啟動專用 Spot 執行個體

範例 1：啟動 競價型執行個體

下列範例不包括可用區域或子網。Amazon EC2 為您選取可用區域。Amazon EC2 會在所選可用區域的預設子網中啟動執行個體。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

範例 2：在指定的可用區域中啟動 競價型執行個體

下列範例包括可用區域。Amazon EC2 會在指定可用區域的預設子網中啟動執行個體。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

範例 3：在指定的子網中啟動 競價型執行個體

下列範例包含子網。Amazon EC2 會在指定子網中啟動執行個體。如果 VPC 為非預設 VPC，則執行個體預設不會接收公有 IPv4 地址。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

若要指派公有 IPv4 位址給非預設 VPC 中的執行個體，請依下列範例所示來指定 AssociatePublicIpAddress 欄位。指定網路介面時，您必須填入網路介面的子網 ID 和安全群組 ID，而非使用前例程式碼區塊所示的 SubnetId 和 SecurityGroupIds 欄位。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "InstanceType": "m5.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
      "Groups": [ "sg-1a2b3c4d5e6f7g8h9" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

範例 4：啟動專用 Spot 執行個體

下列範例請求具有dedicated租用的 Spot 執行個體。專用 Spot 執行個體必須在 VPC 中啟動。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "c5.8xlarge",
  "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
  "Placement": {
```

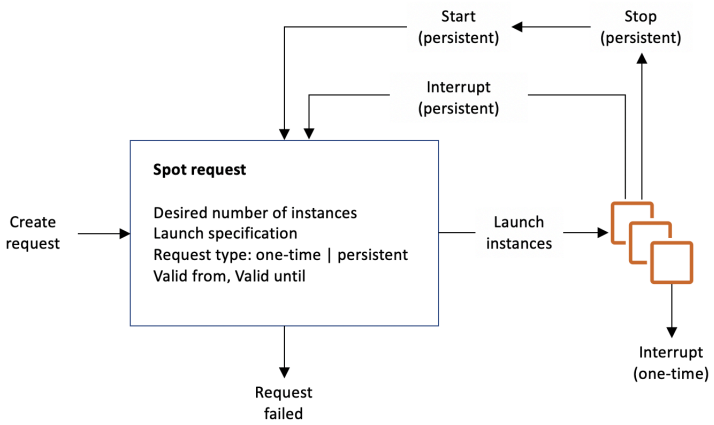
```
    "Tenancy": "dedicated"
  }
}
```

取得 Spot 執行個體請求狀態

為協助您追蹤 Spot 執行個體請求並規劃 Spot 執行個體的使用，請使用 Amazon EC2 提供的請求狀態。例如，請求狀態可提供 Spot 請求尚未完成的理由，或列出阻礙完成 Spot 請求的限制。

在程序的每個步驟中— (此程序也稱為 Spot 請求生命週期)—，特定的事件會決定相繼請求的狀態。

下圖顯示了 Spot 執行個體請求的運作方式。請注意，請求類型 (一次性或持久性) 會決定在 Amazon EC2 中斷 Spot 執行個體時或者如果您停止 Spot 執行個體，是否會再次開啟請求。如果是持久性請求，則會在 Spot 執行個體中斷之後再次開啟。如果是持久性請求，而且您停止 Spot 執行個體，只會在您啟動 Spot 執行個體後開啟請求。



目錄

- 取得請求狀態資訊
- Spot 請求狀態碼
- EC2 Spot 執行個體請求履行事件
- Spot 請求的狀態變更

取得請求狀態資訊

您可取得 Spot 執行個體請求的狀態資訊。

Console

取得請求狀態資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選取 Spot 請求 ，然後選取 Spot 請求。
3. 若要檢查狀態，請在描述索引標籤上檢查狀態欄位。

AWS CLI

取得請求狀態資訊

使用下列 describe-spot-instance-requests 命令。

```
aws ec2 describe-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

取得請求狀態資訊

使用 Get-EC2SpotInstanceRequest cmdlet。

```
Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

Spot 請求狀態碼

Spot 請求狀態資訊包含狀態碼、更新時間和狀態訊息。綜合這些資訊，可協助您決定 Spot 請求的處理方式。

下列是 Spot 請求的狀態碼：

`az-group-constraint`

Amazon EC2 無法在同一個可用區域中，啟動您請求的所有執行個體。

`bad-parameters`

Spot 請求的一個或多個參數無效 (例如，指定的 AMI 不存在)。狀態訊息會顯示哪個參數無效。

`canceled-before-fulfillment`

Spot 請求在完成之前已遭到使用者取消。

`capacity-not-available`

您所請求的執行個體沒有足夠的可用容量。

`constraint-not-fulfillable`

無法完成 Spot 請求，因為有一個或多個限制條件無效 (例如，可用區域不存在)。狀態訊息會顯示哪個限制條件無效。

`fulfilled`

Spot 請求是 `active`，並且 Amazon EC2 正在啟動您的 競價型執行個體。

`instance-stopped-by-price`

您的執行個體已停止，因為 Spot 價格超過您的最高價格。

`instance-stopped-by-user`

您的執行個體已停止，因為使用者已停止該執行個體或從該執行個體執行關機命令。

`instance-stopped-no-capacity`

由於 EC2 容量管理需求，您的執行個體已停止。

`instance-terminated-by-price`

您的執行個體已終止，因為 Spot 價格超過您的最高價格。如果請求為永久性，程序會重新開始執行，因此請求會處於等待評估的狀態。

`instance-terminated-by-schedule`

您的 Spot 執行個體已在排程持續時間結束時終止。

`instance-terminated-by-service`

執行個體會從停止狀態終止。

`instance-terminated-by-user` 或 `spot-instance-terminated-by-user`

您終止了已完成的 Spot 執行個體，因此請求狀態為 `closed` (除非為持久性請求)，而執行個體狀態為 `terminated`。

`instance-terminated-launch-group-constraint`

啟動群組中的一個或多個執行個體已終止，因此不再滿足啟動群組的限制條件。

`instance-terminated-no-capacity`

由於標準容量管理程序，您的執行個體已終止。

`launch-group-constraint`

Amazon EC2 無法同時啟動您請求的所有執行個體。啟動群組中的所有執行個體會同時啟動和終止。

`limit-exceeded`

已超過 EBS 磁碟區數量或磁碟區儲存空間總數量的限制。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon VPC 配額。

`marked-for-stop`

Spot 執行個體已標記停止。

`marked-for-termination`

Spot 執行個體已標記終止。

`not-scheduled-yet`

在尚未達到排程日期之前，將不會評估 Spot 請求。

`pending-evaluation`

在您發出 Spot 執行個體請求後，在系統評估請求的參數時，該請求會進入 `pending-evaluation` 狀態。

`pending-fulfillment`

Amazon EC2 正在嘗試佈建 競價型執行個體。

`placement-group-constraint`

尚無法完成 Spot 請求，因為目前無法將 Spot 執行個體加入置放群組。

`price-too-low`

尚無法完成 Spot 請求，因為最高價格低於 Spot 價格。在這種情況中，不會啟動任何執行個體，而請求的狀態仍為 open。

`request-canceled-and-instance-running`

您在 競價型執行個體仍在執行時取消了 Spot 請求。請求已 `cancelled`，但執行個體仍處於 `running` 狀態。

`schedule-expired`

Spot 請求已過期，因為未在指定日期之前完成。

`system-error`

出現未預期的系統錯誤。如果此問題重複發生，請聯絡 AWS 支援 尋求協助。

EC2 Spot 執行個體請求履行事件

履行 Spot 執行個體請求後，Amazon EC2 會將 EC2 Spot 執行個體請求履行事件傳送至 Amazon EventBridge。您可以建立規則以在發生此事件時採取動作，例如叫用 Lambda 函數或通知 Amazon SNS 主題。

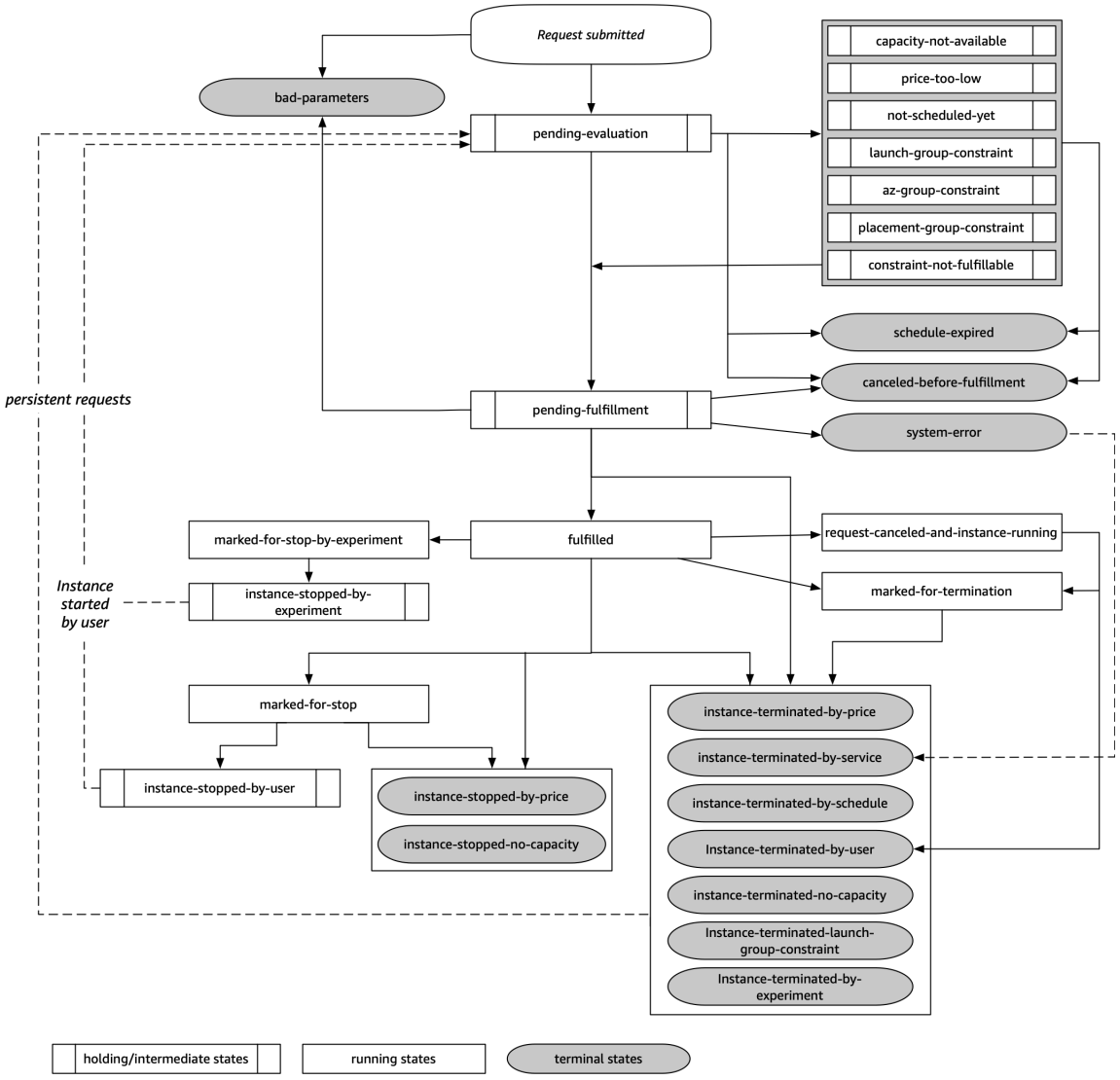
以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "01234567-1234-0123-1234-012345678901",
    "detail-type": "EC2 Spot Instance Request Fulfillment",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "spot-instance-request-id": "sir-0e54a519c9EXAMPLE",
        "instance-id": "i-1234567890abcdef0"
    }
}
```

如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

Spot 請求的狀態變更

下圖顯示 Spot 請求在整個生命週期的期間中可遵循的路徑 (從提交到終止)。每個步驟皆以節點表示，而每個節點的狀態碼則描述了 Spot 請求和 Spot 執行個體的狀態。



等待評估

除非有一個或多個請求參數無效 (pending-evaluation)，否則當您建立 Spot 執行個體請求後，它就會立即進入 bad-parameters 狀態。

狀態碼	請求狀態	執行個體狀態
pending-evaluation	open	不適用
bad-parameters	closed	不適用

保留

如果有一個或多個請求限制為有效但尚未滿足，或是沒有足夠的容量，則請求會進入保留狀態，以等待限制條件滿足。請求選項會影響請求完成的可能性。例如，如果沒有容量，則請求會持續處於保留狀態，直到有可用的容量為止。如果指定了可用區域群組，則請求會持續處於保留狀態，直到可用區域的限制條件滿足。

如果其中一個可用區域中斷，其他可用區域內的 Spot 執行個體請求可用的備用 EC2 容量也可能會受到影響。

狀態碼	請求狀態	執行個體狀態
capacity-not-available	open	不適用
price-too-low	open	不適用
not-scheduled-yet	open	不適用
launch-group-constraint	open	不適用
az-group-constraint	open	不適用
placement-group-constraint	open	不適用
constraint-not-fulfillable	open	不適用

等待評估/實現-終止

如果所建立的請求只在特定期間內有效，而且在請求進入待完成階段之前，此期間就會到期，則此 Spot 執行個體請求可以進入 terminal 狀態。如果您取消請求或系統發生錯誤，也會發生此狀況。

狀態碼	請求狀態	執行個體狀態
schedule-expired	cancelled	

狀態碼	請求狀態	執行個體狀態
		不適用
canceled-before-fulfillment [1]	cancelled	不適用
bad-parameters	failed	不適用
system-error	closed	不適用

[1] 若您取消請求。

等待實現

當您所指定的限制條件 (如果有的話) 已滿足，則 Spot 請求會進入 pending-fulfillment 狀態。

此時，Amazon EC2 會準備佈建您所請求的執行個體。如果程序在此時停止，可能是因為使用者已在啟動 Spot 執行個體前取消程序。也可能是因為發生未預期的系統錯誤。

狀態碼	請求狀態	執行個體狀態
pending-fulfillment	open	不適用

已實現

當滿足 Spot 執行個體的所有規格時，Spot 請求即已完成。Amazon EC2 會啟動 Spot 執行個體，這可能需要幾分鐘。如果 Spot 執行個體已進入休眠或是在中斷時停止，它將會繼續處於此狀態，直到請求可以再次完成或取消。

狀態碼	請求狀態	執行個體狀態
fulfilled	active	pending → running
fulfilled	active	stopped → running

如果您停止 Spot 執行個體，則 Spot 請求會進入 marked-for-stop 或 instance-stopped-by-user 狀態，直到 Spot 執行個體可以再次啟動或取消請求為止。

狀態碼	請求狀態	執行個體狀態
marked-for-stop	active	stopping
instance-stopped-by-user [1]	disabled 或是 cancelled [2]	stopped

[1] 如果您停止執行個體或從執行個體執行 shutdown 指令，Spot 執行個體就會進入 instance-stopped-by-user 狀態。停止執行個體之後，您可以重新啟動它。重新啟動時，Spot 執行個體請求返回 pending-evaluation 狀態，然後在符合限制時 Amazon EC2 會啟動新的 Spot 執行個體。

[2] 如果您停止 Spot 執行個體但不取消請求，Spot 請求狀態則為 disabled。如果 Spot 執行個體已停止且請求到期，請求狀態則為 cancelled。

已實現-終止

只要有該執行個體類型可用的容量，且您未終止執行個體，您的Spot 執行個體就會持續運作。如果 Amazon EC2 必須終止 Spot 執行個體，則 Spot 請求會進入終止狀態。如果取消 Spot 請求或終止 競價型執行個體，請求也會進入終止狀態。

狀態碼	請求狀態	執行個體狀態
request-canceled-and-instance-running	cancelled	running
marked-for-stop	active	running
marked-for-termination	active	running
instance-stopped-by-price	disabled	stopped
instance-stopped-by-user	disabled	stopped

狀態碼	請求狀態	執行個體狀態
instance-stopped-no-capacity	disabled	stopped
instance-terminated-by-price	closed (一次性)，open (持續性)	terminated
instance-terminated-by-schedule	closed	terminated
instance-terminated-by-service	cancelled	terminated
instance-terminated-by-user	closed 或是 cancelled [1]	terminated
instance-terminated-no-capacity	closed (一次性)，open (持續性)	running †
instance-terminated-no-capacity	closed (一次性)，open (持續性)	terminated
instance-terminated-launch-group-constraint	closed (一次性)，open (持續性)	terminated

[1] 如果您終止執行個體但未取消請求，則請求狀態為 closed。如果終止執行個體而且取消請求，則請求狀態為 cancelled。即使您在取消 Spot 執行個體的請求之前就先終止它，在 Amazon EC2 偵測到 Spot 執行個體已終止前，可能會有一段延遲時間。在此種情況中，請求的狀態可以是 cancelled 或 closed。

† 當 Amazon EC2 中斷 Spot 執行個體 (若其需要回收容量)和將執行個體設定為中斷時終止，狀態會立即設定為 instance-terminated-no-capacity (其未設定為 marked-for-termination)。但是，執行個體會保留在 running 狀態 2 分鐘，反映執行個體收到 Spot 執行個體中斷通知的 2 分鐘時段。2 分鐘後，執行個體狀態設定為 terminated。

中斷實驗

您可以使用 AWS Fault Injection Service 啟動 Spot 執行個體中斷，以測試 Spot 執行個體上的應用程式如何回應。如果 AWS FIS 停止 Spot 執行個體，您的 Spot 請求會進入 `marked-for-stop-by-experiment` 狀態，然後進入 `instance-stopped-by-experiment` 狀態。如果 AWS FIS 終止 Spot 執行個體，您的 Spot 請求會進入 `instance-terminated-by-experiment` 狀態。如需詳細資訊，請參閱the section called “啟動中斷”。

狀態碼	請求狀態	執行個體狀態
`marked-for-stop-by-experiment`	`active`	`running`
`instance-stopped-by-experiment`	`disabled`	`stopped`
`instance-terminated-by-experiment`	`closed`	`terminated`

永久性請求

當 Spot 執行個體終止時 (由您或 Amazon EC2 終止)，如果 Spot 請求為持久性請求，則會回到 `pending-evaluation` 狀態，之後 Amazon EC2 就可以在滿足限制條件時啟動新的 Spot 執行個體。

標記 Spot 執行個體請求

為了協助分類和管理您的 Spot 執行個體請求，您可以使用自訂中繼資料來標記它們。您可以在建立 Spot 執行個體請求時或之後將標記指派給請求。您可以使用 Amazon EC2 主控台或命令列工具來指派標籤。

當您標記 Spot 執行個體請求時，系統不會自動標記 Spot 執行個體請求所啟動的執行個體和磁碟區。您需要明確標記 Spot 執行個體請求啟動的執行個體和磁碟區。您可以在啟動期間或之後將標籤指派給 Spot 執行個體和磁碟區。

如需標籤運作方式的詳細資訊，請參閱標記您的 Amazon EC2 資源。

目錄

- 先決條件
- 標記新的 Spot 執行個體請求

- 標記現有的 Spot 執行個體請求
- 檢視 Spot 執行個體請求標籤

先決條件

授予使用者標記資源的許可。如需有關 IAM 政策和政策範例的詳細資訊，請參閱 範例：標籤資源。

您建立的 IAM 政策取決於建立 Spot 執行個體請求時使用的方法。

- 如果您使用啟動執行個體精靈或 `run-instances` 來請求 競價型執行個體，請參閱 To grant a user the permission to tag resources when using the launch instance wizard or run-instances。
- 如果您使用 `request-spot-instances` 命令請求 Spot 執行個體，請參閱 To grant a user the permission to tag resources when using request-spot-instances。

若要授予使用者許可，以便在使用啟動執行個體精靈或 run-instances 時標記資源

建立包含下列項目的 IAM 政策：

- `ec2:RunInstances` 動作。這會授予使用者啟動執行個體的許可。
- 對於 `Resource`，請指定 `spot-instances-request`。這可讓使用者建立 Spot 執行個體請求，它們會請求 Spot 執行個體。
- `ec2:CreateTags` 動作。這會授予使用者建立標籤的許可。
- 對於 `Resource`，請指定 `*`。這可讓使用者標記執行個體啟動期間建立的所有資源。

JSON

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

```
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagSpotInstanceRequests",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

當您使用 RunInstances 動作來建立 Spot 執行個體請求，並在建立時標記 Spot 執行個體請求時，您必須知道 Amazon EC2 如何評估 RunInstances 陳述式中的 `spot-instances-request` 資源，其在 IAM 政策中的評估方式如下：

- 如果您沒有在建立時標記 Spot 執行個體請求，則 Amazon EC2 不會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。
- 如果您在建立時標記 Spot 執行個體請求，Amazon EC2 會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。

因此，對於 `spot-instances-request` 資源，下列規則適用於 IAM 政策：

- 如果您使用 RunInstances 來建立 Spot 執行個體請求，而且您不打算在建立時標記 Spot 執行個體請求，則不需要明確允許 `spot-instances-request` 資源；呼叫將會成功。
- 如果您使用 RunInstances 來建立 Spot 執行個體請求，並打算在建立時標記 Spot 執行個體請求，則必須在 RunInstances 允許陳述式中包含 `spot-instances-request` 資源，否則呼叫將會失敗。
- 如果您使用 RunInstances 建立 Spot 執行個體請求，並打算在建立時標記 Spot 執行個體請求，則必須在 CreateTags 允許陳述式中指定 `spot-instances-request` 資源或包含 * 萬用字元，否則呼叫將會失敗。

例如，IAM 政策 (包括 Spot 執行個體請求不支援的政策)，請參閱 使用 競價型執行個體。

若要授予使用者許可，以便在使用 request-spot-instances 時標記資源

建立包含下列項目的 IAM 政策：

- `ec2:RequestSpotInstances` 動作。這會授予使用者建立 Spot 執行個體請求的許可。
- `ec2:CreateTags` 動作。這會授予使用者建立標籤的許可。
- 對於 `Resource`，請指定 `spot-instances-request`。這允許使用者只標記 Spot 執行個體請求。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotInstanceRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:RequestSpotInstances",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-instances-
request/*"
        }
    ]
}
```

標記新的 Spot 執行個體請求

在 AWS CLI 和 PowerShell 範例中，設定 Spot 執行個體請求，如下所示：

- 對於 `ResourceType`，請指定 `spot-instances-request`。如果您指定另一個值，Spot 執行個體請求將失敗。
- 對於 `Tags`，指定金鑰值對。您可以指定多個金鑰值對。

Console

標記新的 Spot 執行個體請求

1. 遵循管理您的 Spot 執行個體程序。
2. 若要新增標記，請在新增標記頁面上選擇新增標記，然後輸入標記的金鑰和值。為每個額外的標籤選擇新增其他標籤。

   對於每個標籤，您可以用相同的標籤來標記 Spot 執行個體請求、Spot 執行個體以及磁碟區。若要標記所有這三個項目，請確保已選取執行個體、磁碟區和 Spot 執行個體請求。若只要標記一個或兩個，請確定已選取要標記的資源，並清除其他資源。
3. 完成建立 Spot 執行個體請求的必要欄位，然後選擇 Launch (啟動)。如需詳細資訊，請參閱管理您的 Spot 執行個體。

AWS CLI

標記新的 Spot 執行個體請求

使用 `--tag-specification` 選項並搭配 request-spot-instances 命令。

標籤規格會將兩個標籤新增至 Spot 執行個體請求：`Environment=Production` 與 `Cost-Center=123`。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json \
    --tag-specification 'ResourceType=spot-instances-
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

PowerShell

標記新的 Spot 執行個體請求

使用 Request-EC2SpotInstance cmdlet 搭配 `-TagSpecification`。

```
-TagSpecification $tagspec
```

標籤規格的定義如下。它會將兩個標籤新增至 Spot 執行個體請求：`Environment=Production` 與 `Cost-Center=123`。

```
$tag1 = @{Key="Environment"; Value="Production"}
$tag2 = @{Key="Cost-Center"; Value="123"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "spot-instances-request"
$tagspec.Tags = @($tag1,$tag2)
```

標記現有的 Spot 執行個體請求

Console

標記現有的 Spot 執行個體請求

建立 Spot 執行個體請求後，您可以使用主控台為 Spot 執行個體請求新增標籤。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot 執行個體請求。
4. 選擇標籤索引標籤，然後選擇建立標籤。

若要使用主控台標記現有的 Spot 執行個體

在您的 Spot 執行個體請求啟動 Spot 執行個體之後，您可以使用主控台將標籤新增到執行個體。如需詳細資訊，請參閱使用主控台新增標籤。

AWS CLI

標記現有的 Spot 執行個體請求或 Spot 執行個體

使用建立標籤命令來標記現有資源。在下面的例子中，現有的 Spot 執行個體請求和 Spot 執行個體會使用 `purpose=test` 進行標記。

```
aws ec2 create-tags \
    --resources sir-0e54a519c9EXAMPLE i-1234567890abcdef0 \
    --tags Key=purpose,Value=test
```

PowerShell

標記現有的 Spot 執行個體請求或 Spot 執行個體

使用 New-EC2Tag cmdlet。下列範例會將標籤 purpose=test 新增至現有 Spot 執行個體請求和 Spot 執行個體。

```
New-EC2Tag `
    -Resource sir-0e54a519c9EXAMPLE, i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="test"}
```

檢視 Spot 執行個體請求標籤

Console

檢視 Spot 執行個體請求標籤

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot 執行個體請求並選取 Tags (標籤) 標籤。

AWS CLI

若要描述 Spot 執行個體請求標籤

使用 describe-spot-instance-requests 命令來檢視指定 Spot 執行個體請求的組態，其中包括為請求指定的任何標籤。

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].Tags"
```

以下為範例輸出。

```
[
    [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Department",
            "Value": "101"
        }
```

```
    ]
]
```

PowerShell

若要描述 Spot 執行個體請求標籤

使用 Get-EC2SpotInstanceRequest cmdlet。

```
(Get-EC2SpotInstanceRequest `
    -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).Tags
```

以下為範例輸出。

```
Key         Value
---         -----
Environment Production
Department  101
```

若要取消 Spot 執行個體請求

如果您不再需要 Spot 執行個體請求，可以取消它。您只能取消狀態為 `open`、`active` 或 `disabled` 的 Spot 執行個體請求。

- 當請求尚未處理完成，而且尚未啟動任何執行個體時，Spot 執行個體請求的狀態為 `open`。
- 當請求已處理完成，而且因此啟動了 Spot 執行個體時，Spot 執行個體請求的狀態為 `active`。
- 當您停止 Spot 執行個體時，您的 Spot 執行個體請求的狀態為 `disabled`。

如果您的 Spot 執行個體請求的狀態為 `active`，並且擁有相關聯的執行中 Spot 執行個體，則取消請求並不會終止執行個體。如需有關終止 Spot 執行個體的詳細資訊，請參閱終止 Spot 執行個體。

Console

若要取消 Spot 執行個體請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取 Spot 執行個體請求。

4. 選擇動作、取消請求。
5. (選擇性) 如果您已完成與 競價型執行個體 的關聯，您可以終止它們。在取消 Spot 請求對話方塊中，選取終止執行個體，然後選取確認。

AWS CLI

若要取消 Spot 執行個體請求

使用下列 cancel-spot-instance-requests 命令。

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

若要取消 Spot 執行個體請求

使用 Stop-EC2SpotInstanceRequest cmdlet。

```
Stop-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

管理您的 Spot 執行個體

當有容量可用時，Amazon EC2 會啟動 Spot 執行個體。Spot 執行個體會執行，直到它中斷或您自己終止它。

目錄

- 尋找您的 Spot 執行個體
- 尋找由特定請求啟動的執行個體
- 停止 Spot 執行個體
- 啟動 Spot 執行個體
- 終止 Spot 執行個體

尋找您的 Spot 執行個體

Spot 執行個體與隨需執行個體會出現在主控台的執行個體頁面中。使用下列程序，尋找您的 Spot 執行個體。

Console

尋找您的 Spot 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 若要尋找所有 Spot 執行個體，請在搜尋窗格中，選擇執行個體 lifecycle=spot。
4. 若要驗證執行個體是否為 Spot 執行個體，請選取執行個體，選擇詳細資訊索引標籤，然後檢查生命週期的值。Spot 執行個體的值為 `spot`，隨需執行個體的值為 `normal`。

AWS CLI

尋找您的 Spot 執行個體

使用下面的 describe-instances 命令。

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

若要判斷執行個體是否為 Spot 執行個體

使用下面的 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[*].InstanceLifecycle" \
    --output text
```

若該輸出為 `spot`，表示該執行個體為 Spot 執行個體。如果沒有輸出，執行個體即為隨需執行個體。

PowerShell

尋找您的 Spot 執行個體

使用 Get-EC2Instance cmdlet。

```
Get-EC2Instance -Filter @{Name="instance-lifecycle"; Values="spot"}
```

若要判斷執行個體是否為 Spot 執行個體

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.InstanceLifecycle
```

若該輸出為 Spot，表示該執行個體為 Spot 執行個體。如果沒有輸出，執行個體即為隨需執行個體。

尋找由特定請求啟動的執行個體

使用下列程序來尋找從特定 Spot 執行個體或 Spot 機群請求啟動的 Spot 執行個體。

Console

尋找請求的 Spot 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。包含 Spot 執行個體請求和 Spot 機群請求的清單。
3. 如果已完成 Spot 執行個體請求，容量就是 Spot 執行個體的 ID。對於 Spot Fleet ，容量表示已完成所請求容量的程度。若要檢視 Spot Fleet 中的執行個體 ID，請選取展開箭頭，或選取該機群並選取 Instances (執行個體)。
4. 對於 Spot 機群，容量表示已完成所請求容量的程度。若要檢視 Spot 機群中執行個體 ID，請選擇機群 ID 以開啟其詳細資訊頁面，並尋找執行個體窗格。

AWS CLI

尋找請求的 Spot 執行個體

使用下列 describe-spot-instance-requests 命令。

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

下列為範例輸出：

```
[
    {
        "ID": "i-1234567890abcdef0"
    },
```

```
    {
        "ID": "i-0598c7d356eba48d7"
    }
]
```

PowerShell

尋找請求的 Spot 執行個體

使用 Get-EC2SpotInstanceRequest cmdlet。

```
(Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).InstanceId
```

停止 Spot 執行個體

如果您現在不需要您的 競價型執行個體，但想稍後重新啟動它們，而不會遺失 Amazon EBS 磁碟區中保留的資料，您可以將其停止。停止 Spot 執行個體的步驟與停止隨需執行個體的步驟類似。

> **Note**
>
> 停止 Spot 執行個體時，您可以修改其某些執行個體屬性，但無法修改執行個體類型。
> 我們不會對已停止的 Spot 執行個體計費，也不會收取數據傳輸費，但是會收取任何 Amazon EBS 磁碟區的儲存費用。

限制

- 如果從 `persistent` Spot 執行個體請求中啟動 Spot 執行個體，則只能停止 Spot 執行個體。
- 如果已取消關聯的 Spot 執行個體請求，則無法停止 Spot 執行個體。當 Spot 執行個體請求被取消時，您只能終止 Spot 執行個體。
- 如果 Spot 執行個體屬於某個機群或啟動群組或可用區域群組，則無法停止它。

Console

停止 Spot 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。

3. 選取 Spot 執行個體。如果您未儲存 Spot 執行個體的執行個體 ID，請參閱 the section called "尋找您的 Spot 執行個體"。
4. 選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
5. 出現確認提示時，請選擇 Stop (停止)。

AWS CLI

停止 Spot 執行個體

使用 stop-instances 命令手動停止 Spot 執行個體。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

停止 Spot 執行個體

使用 Stop-EC2Instance cmdlet。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

啟動 Spot 執行個體

您可以啟動之前已停止的 Spot 執行個體。

先決條件

只有在下列情況下，才能啟動 Spot 執行個體：

- 您手動停止了 Spot 執行個體。
- Spot 執行個體是 EBS 支援的執行個體。
- Spot 執行個體容量可用。
- Spot 價格低於您的最高價格。

限制

- 如果 Spot 執行個體屬於某個機群或啟動群組或可用區域群組，則無法啟動它。

啟動 Spot 執行個體的步驟與啟動隨需執行個體的步驟類似。

Console

啟動 Spot 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取 Spot 執行個體。如果您未儲存 Spot 執行個體的執行個體 ID，請參閱 the section called "尋找您的 Spot 執行個體"。
4. 選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。

AWS CLI

啟動 Spot 執行個體

使用 start-instances 命令手動啟動 Spot 執行個體。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

啟動 Spot 執行個體

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

終止 Spot 執行個體

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

如果終止持久性 Spot 執行個體請求啟動的執行中或已停止的 Spot 執行個體，Spot 執行個體請求會轉變為 open 狀態，以便啟動新的 Spot 執行個體。若要確保沒有啟動新的 Spot 執行個體，您必須先取消 Spot 執行個體請求。

如果您取消具有執行中 Spot 執行個體的 active Spot 執行個體請求，則執行中的 Spot 執行個體不會自動終止；您必須手動終止 Spot 執行個體。

如果您取消具有已停止的 Spot 執行個體的 disabled Spot 執行個體請求，則 Amazon EC2 Spot 服務會自動終止已停止的 Spot 執行個體。取消 Spot 執行個體請求和 Spot 服務終止 Spot 執行個體之間可能有短暫延遲。

如需詳細資訊，請參閱若要取消 Spot 執行個體請求。

Console

手動終止 Spot 執行個體

1. 終止執行個體前，請檢查您的 Amazon EBS 磁碟區不會在終止時被刪除，並確認已將執行個體存放區磁碟區上的所需資料複製到持久性儲存 (例如 Amazon EBS 或 Amazon S3)，藉此確認您不會遺失任何資料。
2. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇執行個體。
4. 選取 Spot 執行個體。如果您未儲存 Spot 執行個體的執行個體 ID，請參閱 the section called “尋找您的 Spot 執行個體”。
5. 選擇執行個體狀態、終止 (刪除) 執行個體。
6. 出現確認提示時，請選擇終止 (刪除)。

AWS CLI

手動終止 Spot 執行個體

使用 terminate-instances 命令手動終止 Spot 執行個體。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

PowerShell

手動終止 Spot 執行個體

使用 Remove-EC2Instance cmdlet。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

Spot 執行個體中斷

您可以在備用 EC2 容量上啟動 競價型執行個體以獲得大幅折扣，做為在 Amazon EC2 需要取回容量時歸還的交換。當 Amazon EC2 回收 Spot 執行個體時，我們將此事件稱為 Spot 執行個體中斷。

對 競價型執行個體的需求可能會隨時產生極大的變化，而取決於有多少可用的未使用 EC2 執行個體，競價型執行個體的可用性也可能會有顯著的變化。Spot 執行個體隨時都有可能中斷。下列是 Amazon EC2 會中斷 競價型執行個體的可能原因：

容量

Amazon EC2 可以在需要取回時中斷 Spot 執行個體。EC2 取回您的執行個體主要用於重新規劃容量，但也可能基於其他原因，例如主機維護或硬體解除委任。

價格

Spot 價格高於您的最高價格。

您可以在 Spot 請求中指定最高價格。然而，如果您指定了最高價格，執行個體將比未指定最高價時更頻繁受到中斷。

限制條件

如果 Spot 請求包含限制條件，例如啟動群組或可用區域群組，則不再滿足限制條件時，Spot 執行個體會以群組的形式終止。

當 Amazon EC2 中斷 Spot 執行個體時，它會終止、停止或休眠執行個體，具體取決於您在建立 Spot 請求時指定的中斷行為。

目錄

- Spot 執行個體中斷的行為
- 準備 Spot 執行個體中斷
- 啟動 Spot 執行個體中斷
- Spot 執行個體中斷通知

- 尋找中斷的 競價型執行個體
- 確定 Amazon EC2 是否已終止 Spot 執行個體
- 針對已中斷的 競價型執行個體收費

Spot 執行個體中斷的行為

建立 Spot 請求時，您可以指定中斷行為。可能的中斷行為如下：

- 停止
- 休眠
- 終止

預設行為是 Amazon EC2 在 Spot 執行個體中斷時將其終止。

停止中斷的 競價型執行個體

您可以指定 Amazon EC2 在您的 Spot 執行個體中斷時將其停止。Spot 執行個體請求類型必須為 `persistent`。您無法在 Spot 執行個體請求中指定啟動群組。EC2 Fleet 或 Spot 機群請求類型必須為 `maintain`。

考量事項

- 只有 Amazon EC2 才能重新啟動中斷的已停止 Spot 執行個體。
- 針對由 `persistent` Spot 執行個體請求啟動的 Spot 執行個體：當相同可用區域中的容量可用，且執行個體類型與已停止執行個體相同時，Amazon EC2 便會重新啟動已停止的執行個體。
- 停止 Spot 執行個體時，您可以修改其某些執行個體屬性，但無法修改執行個體類型。如果卸載或刪除了 EBS 磁碟區，當 Spot 執行個體啟動時，不會連接 EBS 磁碟區。如果卸載了根磁碟區，而 Amazon EC2 試著啟動 Spot 執行個體，則執行個體會啟動失敗，而且 Amazon EC2 會終止已停止的執行個體。
- 您可以在 Spot 執行個體停止時將其終止。
- 如果取消 Spot 執行個體請求、EC2 Fleet 或 Spot Fleet，Amazon EC2 會終止已停止的任何相關 Spot 執行個體。
- 當終止的 Spot 執行個體停止時，您只需支付保留 EBS 磁碟區的費用。使用 EC2 Fleet 和 Spot Fleet 時，如果有許多已停止的執行個體，可能會超出您帳戶的 EBS 磁碟區數量限制。如需在 Spot 執行個體中斷時計費方式的詳細資訊，請參閱 針對已中斷的 競價型執行個體收費。

- 確定您熟悉停止執行個體的含義。如需停止執行個體時會發生狀況的資訊，請參閱 執行個體狀態之間的差異。

休眠中斷的 競價型執行個體

您可以指定 Amazon EC2 在您的 Spot 執行個體中斷時將其休眠。如需詳細資訊，請參閱讓 Amazon EC2 執行個體進入休眠。

Amazon EC2 現在提供與隨需執行個體目前使用相同的 Spot 執行個體休眠體驗。支援更廣泛，現在針對 Spot 執行個體休眠支援下列內容：

- 更多支援的 AMI
- 更多支援的執行個體系列
- 使用者啟動的休眠

終止已中斷的 Spot 執行個體

當 Amazon EC2 中斷 Spot 執行個體時，預設情況下會終止執行個體，除非您指定了其他中斷行為，例如停止或休眠。如需詳細資訊，請參閱終止 Amazon EC2 執行個體。

準備 Spot 執行個體中斷

對 競價型執行個體的需求可能會隨時產生極大的變化，而取決於有多少可用的未使用 EC2 執行個體，競價型執行個體的可用性也可能會有顯著的變化。Spot 執行個體隨時都有可能中斷。因此，您必須確定應用程式已經為 Spot 執行個體中斷做好準備。

建議您遵循這些最佳實務，以便為 Spot 執行個體中斷做好準備。

- 使用 Auto Scaling 群組建立 Spot 請求。如果您的 Spot 執行個體中斷，Auto Scaling 群組將自動啟動替換執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的具備多個執行個體類型及購買選項的 Auto Scaling 群組。
- 使用包含所需軟體組態的 Amazon Machine Image (AMI)，來確保執行個體在請求完成時能夠立即開始運作。您也可以利用使用者資料，在啟動時執行命令。
- 在停止或終止執行個體時，執行個體儲存體磁碟區上的資料會遺失。將執行個體儲存體磁碟區上的所有重要資料備份到更持久的儲存體，例如 Amazon S3、Amazon EBS 或 Amazon DynamoDB。
- 請將重要的資料定期存放於當 Spot 執行個體終止時不會受到影響的位置。例如，您可以使用 Amazon S3、Amazon EBS 或 DynamoDB。

- 將工作分割為較小的任務 (使用 Grid、Hadoop 或採用佇列的架構) 或使用檢查點，如此您就可以經常儲存工作。
- Amazon EC2 會在執行個體處在提高的中斷風險時對 Spot 執行個體發出重新平衡建議訊號。您可以依賴重新平衡建議，主動管理 Spot 執行個體中斷，而不必等待兩分鐘的 Spot 執行個體中斷通知。如需詳細資訊，請參閱 EC2 執行個體重新平衡建議。
- 使用兩分鐘的 Spot 執行個體中斷通知來監控 Spot 執行個體的狀態。如需詳細資訊，請參閱 Spot 執行個體中斷通知。
- 雖然我們努力地盡快提供這些警示，但您的 Spot 執行個體還是有可能在警示發布之前中斷。測試您的應用程式，確保它能夠從容地處理非預期的執行個體中斷，即使您正在監控重新平衡建議訊號和中斷通知。您可以執行使用隨需執行個體的應用程式，然後自行終止隨需執行個體，從而完成此任務。
- 使用 執行受控故障注入實驗 AWS Fault Injection Service ，以測試您的應用程式在 Spot 執行個體中斷時如何回應。如需詳細資訊，請參閱 AWS Fault Injection Service 使用者指南中教學課程：使用 AWS FIS測試 Spot 執行個體中斷。

啟動 Spot 執行個體中斷

您可以在 Amazon EC2 主控台中選取 Spot 執行個體請求或 Spot 機群請求並啟動 Spot 執行個體中斷，以便您可以測試 Spot 執行個體上的應用程式如何處理中斷。當您啟動 Spot 執行個體中斷時，Amazon EC2 會通知您 Spot 執行個體將在兩分鐘後中斷，接著在兩分鐘後，執行個體會中斷。

執行 Spot 執行個體中斷的基礎服務為 AWS Fault Injection Service (AWS FIS)。如需 的詳細資訊 AWS FIS，請參閱 AWS Fault Injection Service。

> **Note**
>
> 中斷行為是 `terminate`、`stop` 以及 `hibernate`。如果將中斷行為設定為 `hibernate`，當您啟動 Spot 執行個體中斷時，休眠程序將立即開始。

除了亞太區域 （雅加達）、亞太區域 （大阪）、中國 （北京）、中國 （寧夏） 和中東 （阿拉伯聯合大公國） AWS 區域 以外，所有 都支援啟動 Spot 執行個體中斷。

目錄

- 啟動 Spot 執行個體中斷
- 確認 Spot 執行個體中斷
- 配額

啟動 Spot 執行個體中斷

您可以使用 EC2 主控台快速啟動 Spot 執行個體中斷。當您選取 Spot 執行個體請求時，您可以啟動一個 Spot 執行個體的中斷。當您選取 Spot 機群請求時，您可以一次啟動多個 Spot 執行個體的中斷。

如需測試 Spot 執行個體中斷的更進階實驗，您可以使用 AWS FIS 主控台建立自己的實驗。

使用 EC2 主控台啟動 Spot 執行個體請求中的一個 Spot 執行個體的中斷

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot Requests (Spot 請求)。
3. 選取 Spot 執行個體請求，然後選取 Actions (動作)、Initiate interruption (啟動中斷)。您無法選取多個 Spot 執行個體請求來啟動中斷。
4. 在 Initiate Spot Instance interruption (啟動 Spot 執行個體中斷) 對話方塊，於 Service access (服務存取) 下，使用預設角色或選擇現有角色。若要選取現有角色，請選取使用現有的服務角色，然後對於 IAM 角色，選取要使用的角色。
5. 當您準備好要啟動 Spot 執行個體中斷時，請選擇 Initiate interruption (啟動中斷)。

使用 EC2 主控台啟動 Spot 機群請求中的一個或多個 Spot 執行個體的中斷

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot Requests (Spot 請求)。
3. 選取 Spot 機群請求，然後選取動作、啟動中斷。您無法選取多個 Spot 機群請求來啟動中斷。
4. 在指定 Spot 執行個體的數目對話方塊中，對於要中斷之執行個體的數目，輸入要中斷之 Spot 執行個體的數目，然後選擇確認。

   > **Note**
   >
   > 數量不能超過機群中的 Spot 執行個體數量，或每個實驗 AWS FIS 可以中斷的 Spot 執行個體數量配額。

5. 在 Initiate Spot Instance interruption (啟動 Spot 執行個體中斷) 對話方塊，於 Service access (服務存取) 下，使用預設角色或選擇現有角色。若要選取現有角色，請選取使用現有的服務角色，然後對於 IAM 角色，選取要使用的角色。
6. 當您準備好要啟動 Spot 執行個體中斷時，請選擇 Initiate interruption (啟動中斷)。

使用 AWS FIS 主控台建立更進階的實驗來測試 Spot 執行個體中斷

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot Requests (Spot 請求)。
3. 選擇 Actions (動作)，Create advanced experiments (建立進階實驗)。

   AWS FIS 主控台隨即開啟。如需詳細資訊，請參閱「AWS Fault Injection Service 使用者指南」中的教學課程：使用 AWS FIS測試 Spot 執行個體中斷。

確認 Spot 執行個體中斷

在您啟動中斷後，會發生下列情況：

- Spot 執行個體會收到執行個體重新平衡建議。
- Spot 執行個體中斷通知會在 AWS FIS 中斷執行個體的兩分鐘前發出。
- 兩分鐘後，系統會中斷 Spot 執行個體。
- 由 停止的 Spot 執行個體會 AWS FIS 保持停止狀態，直到您重新啟動為止。

確認執行個體在您啟動中斷後是否已中斷

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽窗格中，在單獨的瀏覽器索引標籤或視窗中開啟 Spot Requests (Spot 請求) 和 Instances (執行個體)。
3. 對於 Spot 請求，選取 Spot 執行個體請求或 Spot 機群請求。起始狀態為 `fulfilled`。執行個體中斷後，根據中斷行為，狀態會有如下變更：
   - `terminate` – 狀態會變成 `instance-terminated-by-experiment`。
   - `stop` – 狀態會變成 `marked-for-stop-by-experiment`，然後再變成 `instance-stopped-by-experiment`。
4. 對於 Instances (執行個體)，選取 Spot 執行個體。起始狀態為 `Running`。收到 Spot 執行個體中斷通知後兩分鐘，根據中斷行為，狀態會有如下變更：
   - `stop` – 狀態會變成 `Stopping`，然後再變成 `Stopped`。
   - `terminate` – 狀態會變成 `Shutting-down`，然後再變成 `Terminated`。

配額

對於每個實驗 AWS FIS 可以中斷的 Spot 執行個體數量，您的 AWS 帳戶 具有下列預設配額。

名稱	預設	可調整	說明
aws:ec2:send-spot-instance-interruptions 的目標 Spot 執行個體	每個受支援的區域：5	是	當您每個實驗使用標籤識別目標時，aws:ec2:send-spot-instance-interruptions 可以鎖定之 Spot 執行個體的數目上限。

您可以要求增加配額。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的請求提高配額。

若要檢視 的所有配額 AWS FIS，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務，然後選取 AWS Fault Injection Service。您也可以在 AWS Fault Injection Service 使用者指南中檢視 AWS Fault Injection Service 的所有配額。

Spot 執行個體中斷通知

Spot 執行個體中斷通知是在 Amazon EC2 停止或終止 Spot 執行個體的兩分鐘之前發出的警告。如果將休眠指定為中斷時的行為，您會收到中斷通知，但不會提前兩分鐘收到警告，因為休眠程序會立刻開始執行。

從容處理 Spot 執行個體中斷的最佳方法，就是建立容錯的應用程式架構。若要達成此目的，您可以利用 Spot 執行個體中斷通知。建議您每隔 5 秒檢查這些中斷通知。

中斷通知可做為 EventBridge 事件和 Spot 執行個體上執行個體中繼資料中的項目。中斷通知的原則是盡可能發出。

EC2 Spot Instance Interruption Warning 事件

當 Amazon EC2 要中斷您的 Spot 執行個體時，它會在實際中斷前兩分鐘發出一個事件 (休眠除外，它會收到中斷通知，但不會提前兩分鐘，因為休眠會立即開始)。Amazon EventBridge 可以偵測到此事件。如需 EventBridge 事件的詳細資訊，請參閱「Amazon EventBridge 使用者指南」。如需如何建立和使用事件規則的詳細範例，請參閱利用 Amazon EC2 Spot 執行個體中斷通知。

下列是 Spot 執行個體中斷事件的範例。instance-action 可能的值為 hibernate、stop 或 terminate。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2a:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

> **Note**
>
> Spot 執行個體中斷事件的 ARN 格式是 arn:aws:ec2:*availability-zone*:instance/*instance-id*。此格式與 EC2 資源 ARN 格式不同。

instance-action

instance-action 項目會指定動作和動作發生的約略時間 (UTC)。

如果 Spot 執行個體已標記為由 Amazon EC2 停止或終止，則執行個體中繼資料中會出現 instance-action 項目。否則不會存在。您可以使用執行個體中繼資料服務第 2 版 (IMDSv2) 擷取 instance-action，如下所示。

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/spot/instance-action
```

Windows

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/instance-action
```

下列範例輸出指出此執行個體將會停止的時間點。

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

下列範例輸出指出此執行個體將會終止的時間點。

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

如果 Amazon EC2 未準備停止或終止執行個體，或是您自行終止了執行個體，則執行個體中繼資料中不會出現 `instance-action`，而且您會在嘗試擷取它時收到 HTTP 404 錯誤。

termination-time

`termination-time` 項目會指定執行個體將接收結束訊號的約略時間 (UTC)。

> **Note**
>
> 保留此項目是為了回溯相容性；您應改用 `instance-action`。

如果您的 Spot 執行個體被 Amazon EC2 標記為終止 (由於 Spot 執行個體中斷，中斷行為設定為 `terminate`，或者由於取消持續性 Spot 執行個體請求)，則 `termination-time` 項目會出現在執行個體中繼資料中。否則不會存在。您可以使用 `termination-time` IMDSv2 擷取，如下所示。

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"`
if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-
data/spot/termination-time | grep -q .*T.*Z; then echo termination_scheduled; fi
```

Windows

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/
termination-time
```

以下為範例輸出。

```
2015-01-05T18:02:00Z
```

如果 Amazon EC2 沒有準備終止執行個體 (無論是因為沒有 Spot 執行個體中斷，或是因為中斷行為設定為 `stop` 或 `hibernate`) 或是您自行終止 Spot 執行個體，則 `termination-time` 項目可能不存在於執行個體中繼資料中 (因此您會收到 HTTP 404 錯誤)，或是包含的值並非時間值。

如果 Amazon EC2 終止執行個體的動作失敗，則請求狀態會設定為 `fulfilled`。`termination-time` 值仍會存在於執行個體中繼資料中，包含原來的約略時間 (現在這個時間已經成為過去)。

尋找中斷的 競價型執行個體

當您描述 EC2 執行個體時，結果會包含 Spot 執行個體。Spot 執行個體的執行個體生命週期為 `spot`。Spot 執行個體的執行個體狀態為 `stopped` 或 `terminated`，視您設定的中斷行為而定。若為已休眠的 Spot 執行個體，執行個體狀態為 `stopped`。

如需中斷原因的其他詳細資訊，請檢查 Spot 要求狀態碼。如需詳細資訊，請參閱the section called “取得 Spot 執行個體請求狀態”。

Console

尋找中斷的 Spot 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 套用下列篩選條件：執行個體生命週期=spot。
4. 根據您設定的中斷行為，套用執行個體狀態=已停止或執行個體狀態=已終止篩選條件。
5. 對於每個 Spot 執行個體，在詳細資訊索引標籤的執行個體詳細資訊下，找到狀態轉換訊息 下列程式碼指出 Spot 執行個體已中斷。

   - `Server.SpotInstanceShutdown`
   - `Server.SpotInstanceTermination`

AWS CLI

尋找中斷的 Spot 執行個體

使用 `--filters` 選項並搭配 describe-instances 命令。如需在輸出中列出執行個體 ID，請包含 `--query` 選項。

如果中斷行為是終止 Spot 執行個體，請使用下列範例：

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=terminated \
              Name=state-reason-code,Values=Server.SpotInstanceTermination \
    --query "Reservations[*].Instances[*].InstanceId"
```

如果中斷行為是停止 Spot 執行個體，請使用下列範例：

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=stopped \
              Name=state-reason-code,Values=Server.SpotInstanceShutdown \
    --query "Reservations[*].Instances[*].InstanceId"
```

PowerShell

尋找中斷的 Spot 執行個體

使用 Get-EC2Instance cmdlet。

如果中斷行為是終止 Spot 執行個體，請使用下列範例：

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="terminated"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

如果中斷行為是停止 Spot 執行個體，請使用下列範例：

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="stopped"} `
```

```
            @{Key="state-reason-code";
Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

確定 Amazon EC2 是否已終止 Spot 執行個體

Spot 執行個體會持續執行，直到 Amazon EC2 回應 Spot 執行個體中斷而終止，或直到您自行終止為止。如需詳細資訊，請參閱the section called “中斷行為”。

Spot 執行個體終止後，您可以使用 AWS CloudTrail 來查看 Amazon EC2 是否將其終止。如果 CloudTrail 日誌包含 `BidEvictedEvent`，這表示 Amazon EC2 已終止 Spot 執行個體。如果您看到的是 `TerminateInstances` 事件，則表示使用者終止了該 Spot 執行個體。

或者，如果您想要收到 Amazon EC2 將中斷 Spot 執行個體的通知，請使用 Amazon EventBridge 來回應 EC2 Spot 執行個體中斷警告事件。

在 CloudTrail 中檢視 BidEvictedEvent 事件

1. 透過 https://console.aws.amazon.com/cloudtrail/ 開啟 CloudTrail 主控台。
2. 在導覽窗格中，選擇 Event history (事件歷史記錄)。
3. 從篩選條件清單中，選擇事件名稱，然後在右側的篩選條件欄位中，輸入 **`BidEvictedEvent`**。
4. (選用) 選取時間範圍。
5. 如果清單不是空的，請從產生的項目中選擇 BidEvictedEvent，以開啟其詳細資訊頁面。您可以在事件記錄窗格中找到 Spot 執行個體的相關資訊，包括 Spot 執行個體的 ID。以下為此事件記錄的範例。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "ec2.amazonaws.com"
    },
    "eventTime": "2016-08-16T22:30:00Z",
    "eventSource": "ec2.amazonaws.com",
    "userAgent": "ec2.amazonaws.com",
    "sourceIPAddress": "ec2.amazonaws.com",
    "eventName": "BidEvictedEvent",
    "awsRegion": "us-east-2",
    "eventID": "d27a6096-807b-4bd0-8c20-a33a83375054",
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
```

```
    "RequestParameters": null,
    "ResponseElements": null,
    "serviceEventDetails": {
        "instanceIdSet": [
          "i-1eb2ac8eEXAMPLE"
        ]
    }
}
```

6. 如果您找不到 BidEvictedEvent 事件的項目，請輸入 **TerminateInstances** 作為事件名稱。如需 TerminateInstances 記錄格式的詳細資訊，請參閱 the section called “Amazon EC2 API 事件範例”。

針對已中斷的 競價型執行個體收費

當 Spot 執行個體中斷時，您得支付執行個體和 EBS 磁碟區用量的費用，您也可能需要支付其他費用，收費如下。

執行個體用量

誰中斷 Spot 執行個體	作業系統	在第一個小時中斷	在第一個小時之後的任何小時中斷
如果您停止或終止 Spot 執行個體	Windows 和 Linux (不包括 SUSE)	按使用秒數收費	按使用秒數收費
	SUSE	按整整一小時收費，即使只使用半小時亦同	按使用的完整時數收費，中斷半小時以整整一小時收費
如果 Amazon EC2 中斷 Spot 執行個體	Windows 和 Linux (不包括 SUSE)	免費	按使用秒數收費
	SUSE	免費	按使用的完整時數收費，但中斷半小時免費

EBS 磁碟區用量

當中斷的 Spot 執行個體停止時，您只需支付保留 EBS 磁碟區的費用。

使用 EC2 Fleet 和 Spot Fleet 時，如果有許多已停止的執行個體，可能會超出您帳戶的 EBS 磁碟區數量限制。

EC2 執行個體重新平衡建議

EC2 執行個體重新平衡推薦是一種訊號，可在 Spot 執行個體的中斷風險升高時通知您。該訊號能夠比兩分鐘 Spot 執行個體中斷通知更快到達，讓您有機會主動管理 Spot 執行個體。您可以決定將工作負載重新平衡至未處於提高之中斷風險的新的或現有 競價型執行個體。

Amazon EC2 並不總是能夠在兩分鐘 Spot 執行個體中斷通知之前傳送重新平衡建議訊號。因此，重新平衡建議訊號可以與兩分鐘的中斷通知一起到達。

重新平衡建議可做為 EventBridge 事件和 Spot 執行個體上執行個體中繼資料中的項目。盡可能發出事件。

> **Note**
>
> 僅針對在 2020 年 11 月 5 日 00:00 UTC 之後推出的 競價型執行個體 支援重新平衡建議。

目錄

- 您可以採取的重新平衡動作
- 監控重新平衡建議訊號
- 使用重新平衡建議訊號的服務

您可以採取的重新平衡動作

以下是您可以採取的一些可能的重新平衡動作：

順利關機

當您收到 Spot 執行個體的重新平衡建議訊號時，您可以啟動執行個體關機程序，其中可能包括確保在程序停止之前完成程序。例如，您可以將系統或應用程式日誌上傳到 Amazon Simple Storage Service (Amazon S3)、關閉 Amazon SQS 工作者，或者您可以從網域名稱系統 (DNS) 完成取消註冊。您也可以將工作儲存在外部儲存體中，稍後再繼續執行。

防止排程新工作

當您收到 Spot 執行個體的重新平衡建議訊號時，您可以防止在執行個體上排程新工作，同時繼續使用執行個體，直到排定的工作完成為止。

主動啟動新的替代執行個體

您可以設定 Auto Scaling 群組、EC2 Fleet 或 Spot Fleet ，在發出重新平衡建議訊號時自動啟動替代 Spot 執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用容量重新平衡處理 Amazon EC2 Spot 中斷，以及此指南中的 在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體。

監控重新平衡建議訊號

您可以監視重新平衡建議訊號，以便在發出重新平衡建議訊號時，您可以採取在上一部分中指定的動作。重新平衡建議訊號可作為傳送至 Amazon EventBridge (之前稱為 Amazon CloudWatch Events) 的事件，以及 Spot 執行個體上的執行個體中繼資料。

監控重新平衡建議訊號：

- 使用 Amazon EventBridge
- 使用執行個體中繼資料

使用 Amazon EventBridge

在針對 Spot 執行個體發出重新平衡建議訊號時，訊號的事件會傳送至 Amazon EventBridge。如果 EventBridge 偵測到事件模式符合規則中定義的模式，EventBridge 會叫用規則中指定的目標 (或多個目標)。

以下是重新平衡建議訊號的範例事件。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Instance Rebalance Recommendation",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0"
    }
}
```

下列欄位會形成規則中定義的事件模式：

`"detail-type": "EC2 Instance Rebalance Recommendation"`

識別事件是重新平衡建議事件

`"source": "aws.ec2"`

識別該事件是來自 Amazon EC2

建立 EventBridge 規則

您可以寫入 EventBridge 規則，並自動化在事件模式符合規則時要採取哪些動作。

下列範例會建立 EventBridge 規則，以便在 Amazon EC2 每次發出重新平衡建議訊號時傳送電子郵件、簡訊或行動裝置推播通知。將該信號作為 `EC2 Instance Rebalance Recommendation` 事件發出 ，這觸發了規則定義的動作。

建立 EventBridge 規則之前，您必須針對電子郵件、文字訊息或行動推送通知建立 Amazon SNS 主題。

針對重新平衡建議事件建立 EventBridge 規則

1. 在 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 選擇 Create rule (建立規則)。
3. 針對 Define rule detail (定義規則詳細資訊) 執行下列動作：
   a. 輸入規則的Name (名稱)，或者輸入描述。

      在同一個區域和同一個事件匯流排上，規則不能與另一個規則同名。
   b. 針對 Event bus (事件匯流排) 選擇 default (預設值)。當您帳戶中的 AWS 服務產生事件時，一律會前往您帳戶的預設事件匯流排。
   c. 針對 Rule type (規則類型) 選擇 Rule with an event pattern (具有事件模式的規則)。
   d. 選擇下一步。
4. 針對 Build event pattern (建置事件模式) 執行下列動作：
   a. 在 Event source (事件來源) 欄位中，選擇 AWS events or EventBridge partner events (事件或 EventBridge 合作夥伴事件)。
   b. 針對此範例的 Event pattern (事件模式)，您需指定下列事件模式，以便與 `EC2 Instance Rebalance Recommendation` 事件相符，然後選擇 Save (儲存)。

```
{
```

```
  "source": ["aws.ec2"],
  "detail-type": ["EC2 Instance Rebalance Recommendation"]
}
```

若要新增事件模式，您可以透過選擇 Event pattern form (事件模式表單) 使用範本，或者透過選擇 Custom pattern (JSON editor) (自定義模式 (JSON 編輯器)) 指定自有模式，如下：

i. 若要使用範本建立事件模式，請執行下列動作：

A. 選擇 Event pattern form (事件模式表單)。

B. 在 Event source (事件來源) 欄位中，選擇 AWS services (服務)。

C. 針對 AWS Service (服務)，請選擇 EC2 Spot Fleet (EC2 Spot 機群)。

D. 針對 Event type (事件類型)，選擇 EC2 Instance Rebalance Recommendation (EC2 執行個體重新平衡建議)。

E. 若要自訂範本，請選擇 Edit pattern (編輯模式) 並進行變更以與範例事件模式相符。

ii. (替代) 若要指定自訂事件模式，請執行下列動作：

A. 選擇 Custom pattern (JSON editor) (自訂模式 (JSON 編輯器))。

B. 在 Event pattern (事件模式) 方塊中，為此範例新增事件模式。

c. 選擇下一步。

5. 針對 Select target(s) (選取目標) 執行下列動作：

a. 在 Target types (目標類型) 欄位中，選擇 AWS service (服務)。

b. 針對 Select a target (選取目標) 選取 SNS topic (SNS 主題)，以便在事件發生時傳送電子郵件、簡訊或行動裝置推送通知。

c. 針對 Topic (主題)，請選擇現有的主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

d. (選用) 在 Additional settings (其他設定) 下，您可以選擇性地設定其他設定。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的建立對事件做出反應的 Amazon EventBridge 規則 (步驟 16)。

e. 選擇下一步。

6. (選用) 針對 Tags (標籤)，您可以選擇性地將一或多個標籤指派給您的規則，然後選擇 Next (下一步)。

7. 針對 Review and create (檢閱和建立) 執行下列動作：

a. 檢閱規則的詳細資訊，然後視需求進行修改。
b. 選擇建立規則。

如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 規則和 Amazon EventBridge 事件模式。

使用執行個體中繼資料

執行個體中繼資料類別 `events/recommendations/rebalance` 在為 Spot 執行個體發出重新平衡建議訊號時提供大約的時間 (UTC)。

我們建議您每 5 秒檢查重新平衡建議訊號，這樣您就不會錯過對重新平衡建議採取行動的機會。

如果 Spot 執行個體收到重新平衡建議，則發出訊號的時間會出現在執行個體中繼資料中。您可以擷取如下所示的發出訊號時間。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令。

IMDSV2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

從 Windows 執行個體執行下列 cmdlet

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

透過 Windows 執行個體中執行下列 cmdlet。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/events/
recommendations/rebalance
```

以下是範例輸出，它表示為 Spot 執行個體發出重新平衡建議訊號的時間 (UTC)。

```
{"noticeTime": "2020-10-27T08:22:00Z"}
```

如果尚未針對執行個體發出訊號，則 `events/recommendations/rebalance` 不存在， 並且您會在嘗試擷取它時收到 HTTP 404 錯誤。

使用重新平衡建議訊號的服務

Amazon EC2 Auto Scaling、EC2 Fleet 和 Spot Fleet 使用重新平衡建議信號，讓您在執行中的執行個體收到兩分鐘 Spot 執行個體中斷通知之前，藉由使用新的 Spot 執行個體主動擴增您的機群，讓您輕鬆維持工作負載可用性。您可以讓這些服務主動地監控並回應影響 競價型執行個體 的可用性的變更 如需詳細資訊，請參閱下列內容：

- 「Amazon EC2 Auto Scaling 使用者指南」中的使用容量重新平衡處理 Amazon EC2 Spot 中斷
- 在本使用者指南 EC2 Fleet 和 Spot 機群主題中的 在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體

Spot 配置分數

Spot 配置分數功能可以根據您的 Spot 容量需求建議 AWS 區域、可用區域或本機區域。Spot 容量會波動，而且您無法確定您始終都能取得所需的容量。Spot 配置分數表示 Spot 請求在區域或可用區域中成功的可能性。

> **Note**
>
> Spot 配置分數不會就可用容量或中斷風險提供任何保證。Spot 配置分數僅作為建議提供。

使用案例

您可以針對下列情況使用 Spot 配置分數功能：

- 視需要在不同區域中重新放置並擴展 Spot 運算容量，以回應目前區域中增加的容量需求或降低的可用容量。
- 識別執行單一可用區域工作負載的最佳可用區域。
- 模擬未來的 Spot 容量需求，以便您可以挑選最佳區域來擴充 Spot 型工作負載。
- 尋找執行個體類型的最佳組合以滿足您的 Spot 容量需求。

目錄

- 限制
- 成本
- Spot 配置分數的運作方式
- Spot 配置分數的必要許可
- 計算 Spot 配置分數

限制

- 目標容量限制 – 您的 Spot 配置分數目標容量限制是基於最近的 Spot 用量，同時考量潛在的用量成長。如果沒有最近的 Spot 用量，我們會為您提供符合 Spot 請求限制的低預設限制。
- 請求組態限制 – 如果我們偵測到未與 Spot 配置分數功能之預期用途相關聯的模式，可以在 24 小時期間內限制新請求組態的數目。如果達到限制，您可以重試已使用的請求組態，但無法指定新的請求組態，直到下一個 24 小時期間。
- 執行個體類型數目下限 – 如果指定執行個體類型，則您必須至少指定三種不同的執行個體類型，否則 Amazon EC2 將傳回低 Spot 配置分數。同樣地，如果您指定執行個體屬性，它們必須解析為至少三種不同的執行個體類型。如果執行個體類型具有不同的名稱，則會將其視為不同。例如，m5.8xlarge、m5a.8xlarge 和 m5.12xlarge 都會視為不同。

成本

使用 Spot 配置分數功能無需額外費用。

Spot 配置分數的運作方式

當您使用 Spot 配置分數功能時，首先要針對 Spot 執行個體指定運算需求，然後 Amazon EC2 會傳回 Spot 請求可能成功的前 10 個區域或可用區域。每個區域或可用區域的分數範圍從 1 到 10，其中 10 表示您的 前 Spot 請求很可能成功，1 表示您的 Spot 請求不可能成功。區域結果可以包含可用區域和本機區域。

若要使用 Spot 配置分數功能，請遵循下列步驟：

- 步驟 1：指定您的 Spot 需求
- 步驟 2：篩選 Spot 配置分數回應
- 步驟 3：檢閱建議
- 步驟 4：使用建議

步驟 1：指定您的 Spot 需求

首先，您可以指定所需的目標 Spot 容量和運算需求，如下所示：

1. 指定目標 Spot 容量，並選擇性指定目標容量單位。

   您可以根據執行個體或 vCPU 的數量或 MiB 中的記憶體容量來指定所需的目標 Spot 容量。若要以 vCPU 數目或記憶體數量指定目標容量，您必須將目標容量單位指定為 `vcpu` 或 `memory-mib`。否則，其會預設為執行個體的數量。

   根據 vCPU 數目或記憶體數量來指定目標容量，您可以在計算總容量時使用這些單位。例如，如果想要混合使用不同大小的執行個體，您可以將目標容量指定為 vCPU 總數。Spot 配置分數功能接著會透過其 vCPU 數目來考量請求中的每個執行個體類型，並在加總目標容量時計算 vCPU 總數，而不是執行個體總數。

   例如，假設您指定的目標容量總計為 30 個 vCPU，而執行個體類型清單包含 c5.xlarge (4 個 vCPU)、m5.2xlarge (8 個 vCPU) 和 r5.large (2 個 vCPU)。若要達到總數 30 個的 vCPU，您可以取得 2 c5.xlarge (2*4 個 vCPU)、2 m5.2xlarge (2*8 個 vCPU) 和 3 r5.large (3*2 個 vCPUs) 的混合。
2. 指定執行個體類型或執行個體屬性。

   您可以指定要使用的執行個體類型，也可以指定運算需求所需的執行個體屬性，然後讓 Amazon EC2 識別具有這些屬性的執行個體類型。這就是所謂的屬性型執行個體類型選項。

您無法在相同的 Spot 配置分數請求中同時指定執行個體類型和執行個體屬性。

如果指定執行個體類型，則您必須至少指定三種不同的執行個體類型，否則 Amazon EC2 將傳回低 Spot 配置分數。同樣地，如果您指定執行個體屬性，它們必須解析為至少三種不同的執行個體類型。

如需指定 Spot 需求的不同方式範例，請參閱 範例組態。

步驟 2：篩選 Spot 配置分數回應

Amazon EC2 會計算每個區域或可用區域的 Spot 配置分數，並傳回 Spot 請求可能成功的前 10 個區域或前 10 個可用區域。預設值是傳回得分區域的清單。如果您計劃將所有 Spot 容量啟動至單一可用區域，請求得分可用區域的清單很有用。

您可以指定區域篩選條件來縮小將在回應中傳回的區域。

您可以結合區域篩選條件和得分可用區域的請求。如此一來，得分可用區域僅限於您已篩選的區域。若要找出區域中得分最高的可用區域，請僅指定該區域，然後回應將傳回該區域中所有可用區域的得分清單。

根據預設，Spot 置放分數會忽略 Local Zones。若要包含它們，請將 `IncludeLocalZones`設定為 `true`。對於區域分數，本地區域容量會計入其父區域的分數。對於區域分數，回應可以包含個別的 Local Zone 分數。

步驟 3：檢閱建議

每個區域或可用區域的 Spot 配置分數是根據目標容量、執行個體類型的組成、目前的 Spot 用量趨勢，以及請求的時間來計算。由於 Spot 容量持續波動，因此在不同時間計算時，相同的 Spot 配置分數請求可能會產生不同的分數。

區域和可用區域的分數範圍是從 1 到 10。分數 10 表示您的 Spot 請求很有可能 (但不保證) 成功。分數 1 表示您的 Spot 請求完全不可能成功。對於不同的區域或可用區域，可能會傳回相同的分數。

如果傳回低分數，您可以編輯運算需求並重新計算分數。您也可以在一天中的不同時間，針對相同的運算需求，請求 Spot 配置分數建議。

步驟 4：使用建議

僅在 Spot 請求的組態與 Spot 配置分數組態 (目標容量、目標容量單位，以及執行個體類型或執行個體屬性) 完全相同，且該請求設定為使用 `capacity-optimized` 配置策略時，Spot 配置分數才相關。否則，取得可用 Spot 容量的可能性將不符合分數。

如果您的分數包含 Local Zones，請將後續 Spot 請求設定為使用相同的 Local Zones。否則，取得容量的可能性可能與分數不符。

雖然 Spot 配置分數充當指導方針，而且沒有任何分數保證您的 Spot 請求會完全或部分實現，但您可以使用下列資訊來取得最佳結果：

- 使用相同組態 – 僅在 Auto Scaling 群組、EC2 Fleet 或 Spot Fleet 中的 Spot 請求組態 (目標容量、目標容量單位，以及執行個體類型或執行個體屬性) 與您為了取得 Spot 配置分數而輸入的組態相同時，Spot 配置分數才相關。

  如果您在 Spot 配置分數請求中使用了屬性型執行個體類型選取，則可以使用屬性型執行個體類型選取，來設定 Auto Scaling 群組、EC2 Fleet 或 Spot Fleet。如需詳細資訊，請參閱使用參數型執行個體類型選項建立混合執行個體群組和 指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型。

  > **Note**
  >
  > 如果您根據 vCPU 數目或記憶體數量指定目標容量，並在 Spot 配置分數組態中指定了執行個體類型，請注意，您目前無法在 Auto Scaling 群組、EC2 Fleet 或 Spot Fleet 中建立此組態。反之，您必須使用 `WeightedCapacity` 參數，手動設定執行個體加權。

- 使用 **`capacity-optimized`** 分配策略 – 任何分數都假設您的機群請求將設定為使用所有的可用區域 (用於跨區域請求容量) 或單一可用區域 (如果在某個可用區域請求容量) 和 `capacity-optimized` Spot 分配策略，以便您可以成功請求 Spot 容量。如果您使用其他配置策略 (例如 `lowest-price`)，取得可用 Spot 容量的可能性將不符合分數。
- 立即對分數採取行動 – Spot 配置分數建議會反映請求時的可用 Spot 容量，並且由於 Spot 容量波動，因此在不同時間計算時，相同的組態可能會產生不同的分數。雖然分數 10 表示您的 Spot 容量請求很可能 (但不保證) 成功，但為了取得最佳結果，我們建議您立即對分數採取行動。我們也建議您在每次嘗試容量請求時取得全新的分數。

Spot 配置分數的必要許可

根據預設，IAM 身分 (使用者、角色或群組) 沒有使用 the section called “Spot 配置分數” 的許可。若要允許 IAM 身分使用 Spot 配置分數，您必須建立 IAM 政策，授予使用 `ec2:GetSpotPlacementScores` EC2 API 動作的許可。接著將政策連接至需要此許可的 IAM 身分。

以下為一個範例 IAM 政策，該政策授予使用 `ec2:GetSpotPlacementScores` EC2 API 動作的許可。

JSON

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

如需編輯 IAM 政策的相關資訊，請參閱「IAM 使用者指南」https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-edit.html中的編輯 IAM 政策。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

計算 Spot 配置分數

您可以根據目標容量和運算需求計算 Spot 配置分數。如需詳細資訊，請參閱the section called “Spot 配置分數的運作方式”。

所需的 許可

請確保備妥必要許可。如需詳細資訊，請參閱the section called “所需的許可”。

選項

- 使用執行個體屬性計算
- 使用執行個體類型計算
- 使用 計算 AWS CLI

是否在尋找自動化解決方案？ 您可以建置 Spot 配置分數追蹤器儀表板，以自動擷取分數並將分數儲存在 Amazon CloudWatch 中，而不是遵循本「使用者指南」中的手動步驟。如需詳細資訊，請參閱「 AWS上的建置 Spot 配置分數追蹤器儀表板指南」。

本機區域

若要在使用 Amazon EC2 主控台時包含 Local Zones，您的帳戶必須至少啟用一個 Local Zone。如需詳細資訊，請參閱「AWS Local Zones 使用者指南」中的 AWS 本機區域入門。

使用執行個體屬性計算

執行個體屬性來計算 Spot 配置分數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選擇請求 Spot 執行個體旁的向下箭頭，然後選擇計算 Spot 配置分數。
4. 選擇 Enter requirements (輸入需求)。
5. 對於 Target capacity (目標容量)，根據 instances (執行個體) 或 vCPUs 的數目，或 memory (MiB) (記憶體 (MiB)) 的數量輸入您所需的容量。
6. 對於 Instance type requirements (執行個體類型需求)，若要指定您的運算需求，並讓 Amazon EC2 識別符合這些要求的最佳執行個體類型，請選擇 Specify instance attributes that match your compute requirements (指定符合運算需求的執行個體屬性)。
7. 對於 vCPUs，輸入所需的 vCPU 數量下限和上限。若要指定無限制，請選取 No minimum (無下限)、No maximum (無上限)，或兩者。
8. 對於 Memory (GiB) (記憶體 (GiB))，輸入所需記憶體數量的下限和上限。若要指定無限制，請選取 No minimum (無下限)、No maximum (無上限)，或兩者。
9. 對於 CPU architecture (CPU 架構)，選取所需的執行個體架構。
10. (選用) 對於 Additional instance attributes (其他執行個體屬性)，您可以選擇性地指定一或多個屬性，以更詳細地表達您的運算需求。每個額外屬性都會將進一步的限制新增至您的請求。您可以省略額外屬性；省略時，就會使用預設值。如需每個屬性及其預設值的描述，請參閱 get-spot-placement-scores。

11. (選用) 若要檢視具有所指定屬性的執行個體類型，請展開 Preview matching instance types (預覽相符的執行個體類型)。若要排除執行個體類型，使其不在配置評估中使用，請選取執行個體，然後選取 Exclude selected instance types (排除選取的執行個體類型)。
12. 選擇 Load placement scores (載入配置分數)，然後檢閱結果。
13. (選用) 若要顯示特定區域的 Spot 配置分數，請針對 Regions to evaluate (要評估的區域)，選取要評估的區域，然後選取 Calculate placement scores (計算配置分數)。
14. （選用）在分數選項下，選取包含本機區域。
15. (選用) 若要顯示所顯示區域中可用區域的 Spot 配置分數，請選取提供每個可用區域的配置分數核取方塊。如果您想要將所有 Spot 容量啟動至單一可用區域，得分可用區域的清單很有用。
16. (選用) 若要編輯您的運算需求並取得新的配置分數，請選擇 Edit (編輯)、進行必要的調整，然後選擇 Calculate placement scores (計算配置分數)。

使用執行個體類型計算

指定執行個體類型來計算 Spot 配置分數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選擇請求 Spot 執行個體旁的向下箭頭，然後選擇計算 Spot 配置分數。
4. 選擇 Enter requirements (輸入需求)。
5. 對於 Target capacity (目標容量)，根據 instances (執行個體) 或 vCPUs 的數目，或 memory (MiB) (記憶體 (MiB)) 的數量輸入您所需的容量。
6. 對於 Instance type requirements (執行個體類型需求)，若要指定要使用的執行個體類型，請選取 Manually select instance types (手動選取執行個體類型)。
7. 選取 Select instance types (選取執行個體類型)、選取要使用的執行個體類型，然後選取 Select (選取)。若要快速尋找執行個體類型，您可以使用篩選條件列，依不同屬性篩選執行個體類型。
8. 選擇 Load placement scores (載入配置分數)，然後檢閱結果。
9. (選用) 若要顯示特定區域的 Spot 配置分數，請針對 Regions to evaluate (要評估的區域)，選取要評估的區域，然後選取 Calculate placement scores (計算配置分數)。
10. （選用）在分數選項下，選取包含本機區域。
11. (選用) 若要顯示所顯示區域中可用區域的 Spot 配置分數，請選取提供每個可用區域的配置分數核取方塊。如果您想要將所有 Spot 容量啟動至單一可用區域，得分可用區域的清單很有用。
12. (選用) 若要編輯執行個體類型的清單並取得新的配置分數，請選擇 Edit (編輯)、進行必要的調整，然後選擇 Calculate placement scores (計算配置分數)。

使用 計算 AWS CLI

計算 Spot 配置分數

1. (選用) 若要產生可以針對 Spot 配置分數組態指定的所有可能參數，請使用 get-spot-placement-scores 命令和 `--generate-cli-skeleton` 參數。

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton
```

以下為範例輸出。

```
{
    "InstanceTypes": [
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
```

```
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
```

```
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
    "NextToken": "",
    "IncludeLocalZones": true
}
```

2. 使用上一個步驟的輸出來建立 JSON 組態檔，並將其設定如下：

   a. 對於 `TargetCapacity`，根據執行個體 或 vCPU 的數目，或記憶體 (MiB) 的數量輸入您所需的 Spot 容量。

   b. 對於 `TargetCapacityUnitType`，輸入目標容量的單位。如果省略此參數，則其會預設為 `units`。

      有效值：`units` (其會轉換為執行個體的數量) | `vcpu` | `memory-mib`

   c. 對於 `SingleAvailabilityZone`，指定 `true`，以取得傳回得分可用區域清單的回應。如果您想要將所有 Spot 容量啟動至單一可用區域，得分可用區域的清單很有用。如果省略此參數，則其會預設為 `false`，並且回應會傳回得分區域的清單。

   d. （選用）針對 `IncludeLocalZones`，指定 `true`以包含 Local Zones。對於區域分數，本地區域容量會計入父區域。對於區域分數，回應可以包含個別的 Local Zone 分數。如果省略或設定為 `false`，則會忽略 Local Zones。

   e. (選用) 對於 `RegionNames`，請指定要用作篩選條件的區域。您必須指定區域代碼，例如 `us-east-1`。

      搭配區域篩選條件，回應只會傳回您指定的區域。如果您針對 `SingleAvailabilityZone` 指定了 `true`，則回應只會傳回所指定區域中的可用區域。

   f. 您可以包含 `InstanceTypes` 或 `InstanceRequirements`，但兩者不能同時位於相同的組態中。

      在 JSON 組態中指定下列其中一個項目：

      - 若要指定執行個體類型的清單，請在 `InstanceTypes` 參數中指定執行個體類型。指定至少三種不同的執行個體類型。如果您只指定一個或兩個執行個體類型，Spot 配置分數會傳回低分數。如需執行個體類型的清單，請參閱 Amazon EC2 執行個體類型。

- 若要指定執行個體屬性，以便 Amazon EC2 可以識別符合這些屬性的執行個體類型，請指定位於 `InstanceRequirements` 結構中的屬性。

  您必須提供 `VCpuCount`、`MemoryMiB` 和 `CpuManufacturers` 的值。您可以省略其他屬性；省略時，就會使用預設值。如需每個屬性及其預設值的描述，請參閱 get-spot-placement-scores。

如需組態範例，請參閱 範例組態。

3. 若要取得您在 JSON 檔案中指定之需求的 Spot 配置分數，請使用 get-spot-placement-scores 命令，並使用 `--cli-input-json` 參數指定 JSON 檔案的名稱和路徑。

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

範例輸出：如果 `SingleAvailabilityZone` 設定為 `false` 或將其省略 (如果省略，其會預設為 `false`)，則會傳回區域的得分清單。

```
"SpotPlacementScores": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

如果 `SingleAvailabilityZone` 設定為 ，則輸出範例`true`為 – 可用區域的評分清單，當 `IncludeLocalZones`為 時`true`，會傳回 Local Zones。

```
"SpotPlacementScores": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1",
        "Score": 8
    },
    {
```

```
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3",
        "Score": 6
    },
    ...
```

範例組態

使用 時 AWS CLI，您可以使用下列範例組態。

範例組態

- 範例：指定執行個體類型和目標容量
- 範例：根據記憶體指定執行個體類型和目標容量
- 範例：為屬性型執行個體類型選取指定屬性
- 範例：為屬性型執行個體類型選取指定屬性，並傳回可用區域的得分清單

範例：指定執行個體類型和目標容量

下列範例組態指定三種不同的執行個體類型，以及有 500 個 Spot 執行個體的目標 Spot 容量。

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500
}
```

範例：根據記憶體指定執行個體類型和目標容量

下列範例組態指定了三種不同的執行個體類型，以及有 500,000 MiB 記憶體的目標 Spot 容量，其中要啟動的 Spot 執行個體數目必須提供總計 500,000 MiB 的記憶體。

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
```

```
    ],
    "TargetCapacity": 500000,
    "TargetCapacityUnitType": "memory-mib"
}
```

範例：為屬性型執行個體類型選取指定屬性

下列範例組態是針對屬性型執行個體類型選取設定的，其後是範例組態的文字說明。

```
{
    "TargetCapacity": 5000,
    "TargetCapacityUnitType": "vcpu",
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

InstanceRequirementsWithMetadata

若要使用屬性型執行個體類型選取，您必須在組態中包含 InstanceRequirementsWithMetadata 結構，並為 Spot 執行個體指定所需的屬性。

在上述範例中，指定了下列必要的執行個體屬性：

- ArchitectureTypes – 執行個體類型的架構類型必須是 arm64。
- VirtualizationTypes – 執行個體類型的虛擬化類型必須是 hvm。
- VCpuCount – 執行個體類型必須具有最少 1 個，最多 12 個的 vCPU。
- MemoryMiB – 執行個體類型必須具有最少 512 MiB 的記憶體。省略 Max 參數，表示沒有最大限制。

請注意，有幾個您可以指定的其他選用屬性。如需屬性清單，請參閱 get-spot-placement-scores。

TargetCapacityUnitType

TargetCapacityUnitType 參數會指定目標容量的單位。在此範例中，目標容量為 5000，而目標容量單位類型為 vcpu，它們一起指定了所需目標容量 (5000 個 vCPU)，其中要啟動的 Spot 執行個體數目必須提供總計 5000 個的 vCPU。

範例：為屬性型執行個體類型選取指定屬性，並傳回可用區域的得分清單

下列範例組態是針對屬性型執行個體類型選取而設定的。透過指定 "SingleAvailabilityZone": true，回應將傳回得分可用區域的清單。

```
{
    "TargetCapacity": 1000,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

使用 Spot 執行個體資料摘要追蹤 Spot 執行個體成本

為協助您了解 Spot 執行個體的費用，Amazon EC2 提供了資料摘要來說明您的 Spot 執行個體使用狀況和定價。當您訂閱資料摘要時，此資料摘要會傳送到您所指定的 Amazon S3 儲存貯體。

資料摘要檔案通常會每小時抵達您的儲存貯體一次。如果在某個小時內沒有執行 Spot 執行個體，您不會收到那個小時的資料摘要檔案。

Spot 執行個體的每小時使用量通常涵蓋在單一資料檔案中。這些檔案在傳送到您的儲存貯體之前，會先經過壓縮 (gzip)。如果有特定某小時的使用狀況，產生了大型檔案，則 Amazon EC2 可以寫入多個檔案 (例如，當該小時的檔案內容在壓縮前已超過 50 MB 時)。

> **Note**
>
> 每個 只能建立一個 Spot 執行個體資料饋送 AWS 帳戶。

除中國（北京）、中國（寧夏）、 AWS GovCloud（美國）和預設停用的區域 AWS 以外，所有區域都支援 Spot 執行個體資料饋送。

目錄

- 資料摘要檔案名稱與格式
- Amazon S3 儲存貯體需求
- 訂閱 Spot 執行個體資料摘要
- 檢視資料饋送中的資料
- 刪除 Spot 執行個體資料摘要

資料摘要檔案名稱與格式

Spot 執行個體資料摘要檔案名稱使用下列格式 (日期和小時採用 UTC 時間)：

```
bucket-name.s3.amazonaws.com/optional-prefix/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

例如，如果儲存貯體的名稱為 **amzn-s3-demo-bucket**、前綴字首為 **my-prefix**，則檔案名稱會類似於下列名稱：

```
amzn-s3-demo-bucket.s3.amazonaws.com/my-prefix/111122223333.2023-12-09-07.001.b959dbc6.gz
```

如需儲存貯體名稱的詳細資訊，請參閱「Amazon S3 使用者指南」中的儲存貯體命名規則。

Spot 執行個體資料摘要檔案以製表符 (tab) 分隔。資料檔案中的每一行對應於一個時鐘小時的執行個體用量，並包含下表所列的欄位。

欄位	描述
`Timestamp`	時間戳記，用來判定使用此執行個體的收費價格。

欄位	描述
UsageType	計費的使用類型和執行個體類型。對於 m1.small 競價型執行個體，此欄位設定為 SpotUsage 。如果是其他所有的執行個體類型，此欄位會設定為 SpotUsage: {instance-type}。例如：SpotUsage:c1.medium 。
Operation	收費的產品。若是 Linux 競價型執行個體，此欄位會設定為 RunInstances 。若是 Windows 競價型執行個體，此欄位會設定為 RunInstances:0002 。Spot 的使用會根據可用區域分組。
InstanceID	產生此執行個體用量的 Spot 執行個體的 ID。
MyBidID	產生此執行個體用量的 Spot 執行個體請求的 ID。
MyMaxPrice	針對此 Spot 請求所指定的最高價格。
MarketPrice	Timestamp 欄位中所指定時間點當時的 Spot 價格。
Charge	針對此執行個體用量的計費價格。
Version	資料饋送版本。可用的版本為 1.0。

Amazon S3 儲存貯體需求

訂閱資料摘要時，您必須指定用來存放資料摘要檔案的 Amazon S3 儲存貯體。

在選擇資料摘要的 Amazon S3 儲存貯體之前，請考量下列事項：

- 您必須具有儲存貯體的 FULL_CONTROL 許可。如果您是儲存貯體擁有者，預設會具有此項權限。否則，儲存貯體擁有者必須授予您的 AWS 帳戶 此許可。
- 當您訂閱資料饋送時，這些許可會用來更新儲存貯體 ACL，以提供 AWS 資料饋送帳戶FULL_CONTROL許可。 AWS 資料饋送帳戶會將資料饋送檔案寫入儲存貯體。如果您的帳戶沒有所需的權限，則資料摘要檔案無法寫入儲存貯體。如需詳細資訊，請參閱「Amazon CloudWatch Logs 使用者指南」中的傳送至 Amazon S3 的日誌。

如果您更新 ACL 並移除 AWS 資料饋送帳戶的許可，則無法將資料饋送檔案寫入儲存貯體。您必須重新訂閱資料摘要，以接收資料摘要檔案。

- 每個資料摘要檔案皆擁有自己的 ACL (和儲存貯體的 ACL 不同)。儲存貯體擁有者具有對資料檔案的 FULL_CONTROL 權限。 AWS 資料饋送帳戶具有讀取和寫入許可。
- 如果您刪除資料饋送訂閱，Amazon EC2 不會移除儲存貯體或資料檔案上 AWS 資料饋送帳戶的讀取和寫入許可。您必須自行移除這些權限。
- 如果您使用伺服器端加密搭配存放在 AWS Key Management Service (SSE-KMS) 中的 AWS KMS 金鑰來加密 Amazon S3 儲存貯體，則必須使用客戶受管金鑰。如需詳細資訊，請參閱「Amazon CloudWatch Logs 使用者指南」中的 Simple Storage Service (Amazon S3) 儲存貯體伺服器端加密。

訂閱 Spot 執行個體資料摘要

您可隨時訂閱 Spot 執行個體資料摘要。您無法使用 Amazon EC2 主控台完成此任務。

如果您收到儲存貯體沒有足夠許可的錯誤，請參閱以下文章以取得疑難排解資訊：疑難排解 Spot 執行個體的資料摘要。

AWS CLI

訂閱您的資料摘要

使用 create-spot-datafeed-subscription 命令。

要指定帶字首的儲存貯體，請使用下列範例：

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
    --prefix my-prefix
```

要指定不帶字首的儲存貯體，請使用下列範例：

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket
```

PowerShell

訂閱您的資料摘要

使用 New-EC2SpotDatafeedSubscription cmdlet。

要指定帶字首的儲存貯體，請使用下列範例：

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket `
    -Prefix my-prefix
```

要指定不帶字首的儲存貯體，請使用下列範例：

```
New-EC2SpotDatafeedSubscription `
    -Bucket amzn-s3-demo-bucket
```

檢視資料饋送中的資料

在 中 AWS 管理主控台，開啟 AWS CloudShell。使用下列 s3 sync 命令，從資料摘要的 S3 儲存貯體取得 .gz 檔案，並將這些檔案存放於您指定的資料夾。

```
aws s3 sync s3://amzn-s3-demo-bucket ./data-feed
```

若要顯示 .gz 檔案的內容，請變更至您存放 S3 儲存貯體內容的資料夾。

```
cd data-feed
```

使用 ls 命令檢視檔案的名稱。使用具有檔案名稱的 zcat 命令，顯示壓縮檔案的內容。下列是範例命令。

```
zcat  111122223333.2023-12-09-07.001.b959dbc6.gz
```

以下為範例輸出。

```
#Version: 1.0
#Fields: Timestamp UsageType Operation InstanceID MyBidID MyMaxPrice MarketPrice Charge
 Version
2023-12-09 07:13:47 UTC USE2-SpotUsage:c7a.medium       RunInstances:SV050
 i-0c3e0c0b046e050df     sir-pwq6nmfp    0.0510000000 USD        0.0142000000 USD
  0.0142000000 USD        1
```

刪除 Spot 執行個體資料摘要

當您不再需要 Spot 執行個體資料饋送時，即可將其刪除。

AWS CLI

刪除您的資料摘要

使用 delete-spot-datafeed-subscription 命令。

```
aws ec2 delete-spot-datafeed-subscription
```

PowerShell

刪除您的資料摘要

使用 Remove-EC2SpotDatafeedSubscription cmdlet。

```
Remove-EC2SpotDatafeedSubscription
```

Spot 執行個體請求的服務連結角色

Amazon EC2 使用許可的服務連結角色，它需要代表您呼叫其他 AWS 服務。服務連結角色是直接連結至 的唯一 IAM 角色類型 AWS 服務。服務連結角色提供將許可委派給 的安全方式， AWS 服務 因為只有連結的服務可以擔任服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的服務連結角色。

Amazon EC2 使用名為 AWSServiceRoleForEC2Spot 的服務連結角色來代您管理 競價型執行個體。

AWSServiceRoleForEC2Spot 授予的許可

Amazon EC2 會使用 AWSServiceRoleForEC2Spot 來完成下列動作：

- `ec2:DescribeInstances` - 描述 Spot 執行個體
- `ec2:StopInstances` - 停止 Spot 執行個體
- `ec2:StartInstances` - 啟動 Spot 執行個體

建立服務連結角色

在大部分情況下，您不需要手動建立服務連結角色。您第一次使用主控台建立 Spot 執行個體時，Amazon EC2 會建立 AWSServiceRoleForEC2Spot 服務連結角色。

如果您在 2017 年 10 月之前有作用中的 Spot 執行個體請求，當 Amazon EC2 開始支援此服務連結角色時，Amazon EC2 會在您的帳戶中建立 AWSServiceRoleForEC2Spot 角色 AWS 。如需詳細資訊，請參閱IAM 使用者指南中的 在我的帳戶中出現新的角色。

如果您使用 AWS CLI 或 API 來請求 Spot 執行個體，您必須先確保此角色存在。

使用主控台建立 AWSServiceRoleForEC2Spot

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 選擇 Create Role (建立角色)。
4. 在 Select type of trusted entity (選取信任實體類型) 頁面上，選取 EC2、選取 EC2 - Spot Instances (EC2 - Spot 執行個體)，然後選取 Next: Permissions (下一步：許可)。
5. 在下一頁選擇 Next:Review (下一步：檢閱)。
6. 在檢閱頁面上選擇建立角色。

使用 建立 AWSServiceRoleForEC2Spot AWS CLI

使用如下所示的 create-service-linked-role 命令。

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

如果您不再需要使用 競價型執行個體，建議您刪除 AWSServiceRoleForEC2Spot 角色。從您的帳戶中刪除此角色後，如果您請求 競價型執行個體，Amazon EC2 會再次建立此角色。

授予客戶受管金鑰存取權，以便與加密的 AMI 和 EBS 快照搭配使用

如果您為 Spot 執行個體指定加密的 AMI 或加密的 Amazon EBS 快照，而且使用客戶受管金鑰進行加密，您必須授予 AWSServiceRoleForEC2Spot 角色許可才能使用客戶受管金鑰，使得 Amazon EC2 可以代表您啟動 Spot 執行個體。若要執行此動作，您必須將授予新增至客戶受管金鑰，如下列程序所示。

提供許可時，授與為金鑰政策的替代方案。如需詳細資訊，請參閱 開發人員指南中的使用授予 AWS KMS和在 AWS Key Management Service 中使用金鑰政策。

若要授予 AWSServiceRoleForEC2Spot 角色許可以使用客戶受管金鑰

- 使用 create-grant 命令將授予新增至客戶受管金鑰，並指定獲授予許可能夠執行授予所允許作業的委託人 (AWSServiceRoleForEC2Spot 服務連結角色)。客戶受管金鑰由 `key-id` 參數和客戶受

管金鑰的 ARN 指定。委託人是透過 `grantee-principal` 參數和 AWSServiceRoleForEC2Spot 服務連結角色的 ARN 指定。

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
spot.amazonaws.com/AWSServiceRoleForEC2Spot \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

Spot 執行個體配額

每個區域每個 AWS 帳戶 的執行中 Spot 執行個體請求和待定 Spot 執行個體請求有數量配額。待定 Spot 執行個體請求完成之後，請求就不會再計入配額中，因為執行中的執行個體會計入配額中。

Spot 執行個體配額是根據執行中的 Spot 執行個體正在使用或將使用的虛擬中央處理單元 (vCPU) 的數量來管理的，直到完成開放的 Spot 執行個體請求。如果您終止 Spot 執行個體但未取消 Spot 執行個體請求，請求會計入您的 Spot 執行個體 vCPU 配額，直到 Amazon EC2 偵測到 Spot 執行個體終止並關閉請求。

我們提供以下 Spot 執行個體配額類型。

名稱	預設	可調整
所有 DL Spot 執行個體請求	0	是
所有 F Spot 執行個體請求	0	是
所有 G 和 VT Spot 執行個體請求	0	是
所有 Inf Spot 執行個體請求	0	是
所有 P Spot 執行個體請求	0	是
所有標準 (A, C, D, H, I, M, R, T, Z) Spot 執行個體請求	5	是

名稱	預設	可調整
所有 Trn Spot 執行個體請求	0	是
所有 X Spot 執行個體請求	0	是

即使 Amazon EC2 根據使用量自動調整 Spot 執行個體配額，您也可以在必要時要求提高配額。例如，如果您打算啟動的 Spot 執行個體超過目前配額允許的範圍，您可以請求提高配額。如果您送出 Spot 執行個體請求並收到錯誤訊息 `Max spot instance count exceeded`，您可以要求增加請求的配額。若要請求增加配額，可以使用 Amazon EC2 服務配額 中所述的 Service Quotas 主控台。

您可以啟動任意執行個體類型的組合，這些類型組合可滿足不斷變化的應用程式需求。例如，在 256 個 vCPU 的標準 Spot 執行個體請求配額下，您可以啟動 32 個 `m5.2xlarge` Spot 執行個體 (32 x 8 個 vCPU) 或 16 個 `c5.4xlarge` Spot 執行個體 (16 x 16 個 vCPU)。

透過 Amazon CloudWatch 指標整合，您可以根據配額來監控 EC2 用量。您也可以設定警示，在接近配額時提出警告。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的 Service Quotas 和 Amazon CloudWatch 警示以及。

Amazon EC2 專用執行個體

Amazon EC2 專用主機是完全供您使用的實體伺服器。您可選擇將執行個體容量分享給其他 AWS 帳戶。如需詳細資訊，請參閱 跨帳戶 Amazon EC2 專用主機共用。

專用主機提供執行個體置放的可見性和控制，並支援主機親和性。這表示您可以在特定主機上啟動和執行執行個體，並確保執行個體僅在特定主機上執行。如需詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。

專用主機提供全面的自帶授權 (BYOL) 支援。其允許您根據授權條款，使用現有依通訊端數、依核心數或依虛擬機器數的軟體授權，包括 Windows Server、SQL Server、SUSE Linux Enterprise Server、Red Hat Enterprise Linux 或其他綁定到 VM、通訊端或實體核心的軟體授權。

如果您需要執行個體在專用硬體上執行，但不需要可見性或控制執行個體配置，也不需要使用依通訊端數、依核心數的軟體授權，您可以考慮改用專用執行個體。專用執行個體及專用主機都可用於在專用實體伺服器啟動 Amazon EC2 執行個體。專用執行個體和專用執行個體上的執行個體之間沒有效能、安全或實體差異。不過，兩者之間還是有一些差異。下表特別列出了專用執行個體與專用執行個體之間的主要差異：

	Dedicated Host	Dedicated Instance
專用實體伺服器	具備執行個體容量的實體伺服器，完全專供您使用。	專用於單一客戶帳戶的實體伺服器。
執行個體容量共用	可與其他帳戶共用執行個體容量。	不支援
帳單	按每台主機計費	按每個執行個體計費
通訊端、核心和主機 ID 的可見度	提供通訊端與實體核心數目的可見度	沒有可見度
主機和執行個體親和性	可讓您隨時間一致地將執行個體部署到相同的實體伺服器	不支援
訂定執行個體配置目標	對於執行個體在實體伺服器上的配置方式，提供額外可見性和控制權	不支援
自動執行個體復原功能	支援。如需詳細資訊，請參閱 Amazon EC2 專用主機復原。	支援
使用自有授權 (BYOL)	支援	部分支援 *
Capacity Reservations	不支援	支援

* 透過軟體保證而實現的授權行動性的 Microsoft SQL Server 和 Windows 虛擬桌面存取 (VDA) 授權可以與專用預留執行個體一起使用。

如需專用預留執行個體的詳細資訊，請參閱 Amazon EC2 專用執行個體。

專用執行個體 限制

在您配置專用執行個體之前，請記下以下的法規與限制：

- 若要在專用主機執行 RHEL 及 SUSE Linux，您必須使用自有的 AMI。您無法搭配 AWS Marketplace 專用主機使用由 提供 AWS 或在 上提供的 RHEL 和 SUSE Linux AMIs。如需如何建立自己的 AMI 的詳細資訊，請參閱 將您自己的軟體授權帶到 Amazon EC2 專用主機。

  此限制不適用於配置給記憶體密集型執行個體 (`u-6tb1.metal`、`u-9tb1.metal`、`u-12tb1.metal`、`u-18tb1.metal` 和 `u-24tb1.metal`) 的主機。由 提供 AWS 或在 上提供的 RHEL 和 SUSE Linux AMIs AWS Marketplace 可與這些主機搭配使用。
- 每個區域每個 AWS 帳戶每個執行個體系列的執行中專用執行個體有數量限制。配額僅適用於執行中的執行個體。如果執行個體處於待定、停止中或已停止狀態，則不會計入您的配額。若要檢視帳戶配額，或請求提升配額，可使用 Service Quotas 主控台。
- 僅在使用指定主機資源群組的啟動範本時，才支援 Auto Scaling 群組。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用進階設定建立啟動範本。
- 不支援 Amazon RDS 執行個體。
- AWS 免費用量方案不適用於專用主機。
- 執行個體的配置控制是指針對專用執行個體中所啟動的執行個體進行管理。您無法在置放群組中啟動專用主機。
- 如果您為虛擬執行個體類型分配主機，則在分配主機之後，無法將執行個體類型修改為 `.metal` 執行個體類型。例如，如果您為 `m5.large` 執行個體類型分配主機，則無法將執行個體類型修改為 `m5.metal`。

  同樣，如果您為 `.metal` 執行個體類型分配主機，則在分配主機之後，無法將執行個體類型修改為虛擬執行個體類型。例如，如果您為 `m5.metal` 執行個體類型分配主機，則無法將執行個體類型修改為 `m5.large`。

目錄

- Amazon EC2 專用主機定價和計費
- Amazon EC2 專用主機執行個體容量配置
- Amazon EC2 專用主機上的爆量效能 T3 執行個體
- 將您自己的軟體授權帶到 Amazon EC2 專用主機
- Amazon EC2 專用主機自動配置和主機親和性
- 配置 Amazon EC2 專用主機供您的帳戶使用
- 在 Amazon EC2 專用主機上啟動 Amazon EC2 執行個體
- 將 Amazon EC2 執行個體啟動至主機資源群組

- 修改現有 Amazon EC2 專用主機的自動配置設定
- 修改現有 Amazon EC2 專用主機支援的執行個體類型
- 修改 Amazon EC2 執行個體的 Amazon EC2 專用主機租用和親和性
- 釋出 Amazon EC2 專用主機
- 遷移至 Nitro 型 Amazon EC2 專用主機
- 以專用主機帳單折扣購買專用主機保留
- 跨帳戶 Amazon EC2 專用主機共用
- 上的 Amazon EC2 專用主機 AWS Outposts
- Amazon EC2 專用主機復原
- Amazon EC2 專用主機的主機維護
- 監控您的 Amazon EC2 專用主機的狀態
- 使用 追蹤 Amazon EC2 專用主機組態變更 AWS Config

Amazon EC2 專用主機定價和計費

專用主機 定價會根據付款選項有所差異。

付款選項

- 隨需專用執行個體
- Dedicated Host Reservations
- Savings Plans
- 在專用執行個體的 Windows Server 定價

隨需專用執行個體

當您將專用執行個體配置到帳戶時，會自動啟用隨需計費。

專用執行個體的隨需價格依執行個體系列和區域而異。不論您選擇在專用執行個體中啟動的執行個體數量或大小為何，都會以每秒 (最低 60 秒) 費率計費。如需隨需定價的詳細資訊，請參閱 Amazon EC2 專用執行個體 隨需定價。

您可以隨時釋出隨需專用執行個體以停止累積計費。如需如何釋出專用執行個體的資訊，請參閱釋出 Amazon EC2 專用主機。

Dedicated Host Reservations

專用執行個體預留提供相當於執行隨需專用執行個體的計費折扣。保留有三種付款選項可用：

- 無預付— 無預付保留可提供您一段期間的專用執行個體用量折扣，且不需要預付款。提供一年期和三年期。只有部分執行個體系列支援三年期無預付保留。
- 部分預付— 必須預先支付其中一部分的保留，期間內剩餘的小時數則會以折扣後的費率計費。提供一年期和三年期。
- 全額預付— 提供最低的有效價位。提供一年期和三年期，並預付整期的成本，日後不需另外付費。

您的帳戶中必須具備作用中專用執行個體，才能購買保留。每個保留可以涵蓋單一可用區域中支援相同執行個體系列的一或多部主機。保留會套用至主機上的執行個體系列，而不是執行個體大小。若您有三個具有不同大小執行個體的專用執行個體 (`m4.xlarge`、`m4.medium` 和 `m4.large`)，您可以將單一 `m4` 保留與所有那些專用執行個體建立關聯。保留的執行個體系列和可用區域必須符合您想與其建立關聯之專用執行個體的執行個體系列和可用區域。

當保留與專用執行個體建立關聯時，直到保留的期間結束前都不能釋出專用執行個體。

如需保留定價的詳細資訊，請參閱 Amazon EC2 專用執行個體定價。

Savings Plans

Savings Plans 是個彈性定價模式，提供 隨需執行個體 大幅節省成本。透過 Savings Plans，您可以對一致的使用量設定合約，每小時以美金計價，綁約一年或三年。這可提供您彈性使用 專用執行個體 以達到更佳的需求，且持續的節省金費，而不是與特定的 專用執行個體 綁約。如需詳細資訊，請參閱 AWS Savings Plans 使用者指南。

> **Note**
>
> Savings Plans 不支援 `u-6tb1.metal`、`u-9tb1.metal`、`u-12tb1.metal`、`u-18tb1.metal` 及 `u-24tb1.metal` 專用執行個體。

在專用執行個體的 Windows Server 定價

根據 Microsoft 授權條款，您可以將現有的 Windows Server 和 SQL Server 授權帶到 專用執行個體。如果您選取自帶授權，則軟體使用不收取額外費用。

此外，您也可以使用 Amazon 提供的 Windows Server AMI，在 專用執行個體 上執行最新版本的 Windows Server。對於現有 SQL Server 授權有資格執行 專用執行個體，但需要 Windows Server 來執行 SQL Server 工作負載時，這種情況很常見。由 Amazon 提供的 Windows Server AMI 僅在最新一代執行個體類型上受支援。如需詳細資訊，請參閱 Amazon EC2 專用執行個體定價。

Amazon EC2 專用主機執行個體容量配置

專用執行個體支援不同組態 (實體核心、通訊端和 VCPU)，能讓您執行不同系列和大小的執行個體。

在帳戶中配置專用執行個體時，您可以選擇能支援單一執行個體類型的組態，或相同執行個體系列中多種執行個體類型的組態。您可在主機上執行的執行個體數量將依您選擇的組態而定。

目錄

- 單一執行個體類型支援
- 多種執行個體類型支援

單一執行個體類型支援

您可以配置僅支援一種執行個體類型的專用執行個體。若使用此組態，您在專用執行個體上啟動的每個執行個體，其執行個體類型皆必須與配置主機時指定的執行個體類型相同。

例如，您可以配置僅支援 `m5.4xlarge` 執行個體類型的主機。在此情況下，您僅能執行該主機上的 `m5.4xlarge` 執行個體。

您可在主機上啟動的執行個體數量，將依該主機提供的實體核心數量以及指定執行個體類型使用的核心數目而定。例如，如果您為 `m5.4xlarge` 執行個體配置主機，且該主機提供 48 個實體核心，則每個 `m5.4xlarge` 執行個體會消耗 8 個實體核心。這表示您最多可在該主機上啟動 6 個執行個體 (48 個實體核心 / 每個執行個體 8 個核心 = 6 個執行個體)。

多種執行個體類型支援

您可以配置能支援相同執行個體系列中多種執行個體類型的專用執行個體。此舉可讓您在相同主機上執行不同類型的執行個體，只要這些執行個體位於相同執行個體系列中，且主機具有足夠的執行個體容量即可。

例如，您可以配置能支援 R5 執行個體系列中不同執行個體類型的主機。在此情況下，您可以根據主機的實體核心容量，在該主機上啟動特定 R5 執行個體類型的組合，例如 `r5.large`、`r5.xlarge`、`r5.2xlarge` 以及 `r5.4xlarge`。

下列執行個體系列可支援具有多種執行個體類型支援的專用執行個體：

- 一般用途：A1 | M5 | M5n | M6i | M7i | T3
- 運算優化：C5 | C5n | C6i | C7i
- 記憶體最佳化：R5 | R5b | R5n | R6i | R7i | R7iz | X2iedn

您可在主機上執行的執行個體數量，將依該主機提供的實體核心數量以及在該主機上執行的每種執行個體類型所消耗的核心數目而定。例如，如果配置一個會提供 48 個實體核心的 R5 主機，在執行兩個 r5.2xlarge 執行個體 (4 個核心 x 2 個執行個體) 以及三個 r5.4xlarge 執行個體 (8 個核心 x 3 個執行個體) 時，這些執行個體總共會消耗 32 個核心，而且只要在剩餘的 16 個核心範圍內，您就可執行特定組合的 R5 執行個體。

但是，對於每個執行個體系列，每種執行個體類型都有限制可執行的執行個體數目。例如，R5 專用執行個體最多可支援 2 個 r5.8xlarge 執行個體，使用 32 個實體核心。在此情況下，可使用其他較小的 R5 執行個體類型來將主機填滿至核心容量。如需了解每個執行個體系列支援的執行個體類型數量，請參閱專用主機組態表格。

下表所示為執行個體類型組合範例：

執行個體系列	範例執行個體類型組合
R5	• 範例 1：4 x r5.4xlarge + 4 x r5.2xlarge • 範例 2：1 x r5.12xlarge + 1 x r5.4xlarge + 1 x r5.2xlarge + 5 x r5.xlarge + 2 x r5.large
C5	• 範例 1：1 x c5.9xlarge + 2 x c5.4xlarge + 1 x c5.xlarge • 範例 2：4 x c5.4xlarge + 1 x c5.xlarge + 2 x c5.large
M5	• 範例 1：4 x m5.4xlarge + 4 x m5.2xlarge •

執行個體系列	範例執行個體類型組合	
	範例 2：1 x m5.12xlarge + 1 x m5.4xlarge + 1 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large	

考量事項

在使用支援多種執行個體類型的專用執行個體時，請注意下列各項：

- 同一主機上使用多種執行個體類型，僅限於相同的執行個體系列。
- 混合使用多種執行個體類型時，若要最大化主機使用率，建議先啟動較大型的執行個體類型，再啟動較小型的執行個體類型。
  - 根據專用主機上執行個體類型的組合與啟動順序，實際上可能無法最大化主機使用率。在主機上混合使用多種執行個體類型時，主機可能會有部分可用容量但無法使用。例如，您可能會看到 r5n 主機上有 16 個可用 vCPU，但即使 r5n.4xlarge 需使用 16 個 vCPU 執行，仍可能無法在該主機上啟動 4xlarge 執行個體。

> **Note**
>
> 若您將 A1 專用主機設定為支援多種執行個體類型，僅能在該主機上混合啟動 a1.xlarge 與 a1.2xlarge 執行個體。若您在該主機上啟動 a1.medium 或 a1.large 執行個體，此後僅能在該主機上繼續啟動相同類型的執行個體。單一 a1.4xlarge 執行個體會佔用主機的所有容量。若您需要主機專門用於 a1.medium 或 a1.large 執行個體，建議為這些執行個體類型單獨配置主機。

Amazon EC2 專用主機上的爆量效能 T3 執行個體

專用執行個體支援爆量效能 T3 執行個體。T3 執行個體可讓您在專用硬體上使用合格的 BYOL 授權軟體，可說是符合成本效益的方式。T3 執行個體的 vCPU 佔用空間較小，合併工作負載時所需的主機較少，而且每個核心授權都可以發揮最大的使用率。

T3 專用執行個體最適合執行 BYOL 軟體，而 CPU 使用率為低度至中等。這包括符合資格的依通訊端數、依核心數或依虛擬機器數的軟體授權，例如 Windows Server、Windows 桌面、SQL Server、SUSE Enterprise Linux Server、Red Hat Enterprise Linux 及 Oracle 資料庫。例如，適用於 T3 專用執行個體的工作負載包括中小型資料庫、虛擬桌面、開發及測試環境、程式碼儲存庫和產品原

型等。T3 專用執行個體不建議用於 CPU 持續使用率高的工作負載，或同時遇到相關 CPU 高載問題的工作負載。

專用執行個體上的 T3 執行個體使用的額度模型是與共用租用硬體上的 T3 執行個體相同。但是，其僅支援 `standard` 額度模型；並不支援 `unlimited` 額度模型。在 `standard` 模式中，專用主機上的 T3 執行個體使用與共用租用硬體上爆量效能執行個體相同的方式獲得、花費以及累積抵用金。它們提供基準 CPU 效能，並可大幅超越基準水準。若要讓效能突增至超過基準，執行個體會用掉在 CPU 額度餘額中所累積的額度。當累積的額度耗盡時，CPU 使用率會降低至基準水準。如需 `standard` 模式的詳細資訊，請參閱 標準爆量效能執行個體如何運作。

T3 專用執行個體支援 Amazon EC2 專用執行個體提供的所有功能，包括在單一主機、主機資源群組和 BYOL 上的多種執行個體規模。

支援的 T3 執行個體大小和組態

T3 專用執行個體執行一般用途高載 T3 執行個體，透過提供基準 CPU 效能及在需要時高載執行更高的效能，來分享主機的 CPU 資源。如此一來，擁有 48 個核心的 T3 專用執行個體，每台主機最多可支援 192 個執行個體。為了有效使用主機的資源並提供最佳執行個體效能，Amazon EC2 執行個體置放演算法會自動計算可在主機上啟動的受支援執行個體數量和執行個體大小組合。

T3 專用執行個體支援同一主機上的多個執行個體類型。所有 T3 執行個體大小都受專用執行個體的支援。您可執行各種不同的 T3 執行個體組合，只要不超出主機的 CPU 上限即可。

下表列出支援的執行個體類型、摘要說明每個執行個體類型的效能，並指明可啟動之每種大小的執行個體數目上限。

執行個體類型	vCPUs	記憶體 (GiB)	每個 vCPU 的基準 CPU 使用率	網路高載頻寬 (Gbps)	Amazon EBS 高載頻寬 (Mbps)	每個專用執行個體的執行個體數目上限
t3.na	2	0.5	5%	5	最高 2,085	192
t3.m	2	1	10%	5	最高 2,085	192
t3.sn	2	2	20%	5	最高 2,085	192

執行個體類型	vCPUs	記憶體 (GiB)	每個 vCPU 的基準 CPU 使用率	網路高載頻寬 (Gbps)	Amazon EBS 高載頻寬 (Mbps)	每個專用執行個體的執行個體數目上限
t3.m	2	4	20%	5	最高 2,085	192
t3.la	2	8	30%	5	2,780	96
t3.xl	4	16	40%	5	2,780	48
t3.2x e	8	32	40%	5	2,780	24

監控 T3 專用執行個體的 CPU 使用率

您可使用 `DedicatedHostCPUUtilization` Amazon CloudWatch 指標來監控專用執行個體的 vCPU 使用率。EC2 命名空間和 `Per-Host-Metrics` 維度均可使用此指標。如需詳細資訊，請參閱專用執行個體指標。

將您自己的軟體授權帶到 Amazon EC2 專用主機

專用執行個體可讓您使用現有的每個通訊端、每個核心或每個 VM 軟體授權。當您使用自有授權時，需負責管理自己的授權。不過，Amazon EC2 具有的功能可協助您保有授權合規，例如執行個體親和性和鎖定的放置。

以下是將您自己的磁碟區授權機器映像帶入 Amazon EC2 的一般步驟。

1. 確定授權條款 (控管機器映像的使用辦法) 允許您在虛擬化雲端環境中使用機器映像。如需 Microsoft 授權的詳細資訊，請參閱 Amazon Web Services 和 Microsoft 授權。
2. 確定您可以在 Amazon EC2 中使用機器映像後，請使用 VM Import/Export 將其匯入。如需如何匯入機器映像的資訊，請參閱 VM Import/Export 使用者指南。
3. 匯入機器映像之後，您就可以透過該映像在帳戶中的作用中專用執行個體上啟動執行個體。
4. 依據作業系統而定，當您執行這些執行個體時，您可能必須為自己的 KMS 伺服器 (例如 Windows Server 或 Windows SQL Server) 啟用執行個體。您無法為 Amazon Windows KMS 伺服器啟用匯入的 Windows AMI。

> **Note**
>
> 若要追蹤映像的使用方式 AWS，請在其中啟用主機錄製 AWS Config。您可以使用 AWS Config 記錄專用主機的組態變更，並使用輸出做為授權報告的資料來源。如需詳細資訊，請參閱使用 追蹤 Amazon EC2 專用主機組態變更 AWS Config。

Amazon EC2 專用主機自動配置和主機親和性

您可以在執行個體層級和主機層級上進行 專用執行個體 配置控制。

自動配置

您會在主機層級設定自動配置。這可讓您管理執行個體要在特定主機上啟動，或在任何具有相符組態的可用主機上啟動。

當為專用主機停用自動配置時，其只接受指定其唯一主機 ID 的主機租用執行個體啟動。這是新專用執行個體的預設設定。

當為專用主機啟用自動配置時，它會接受任何與其執行個體類型組態相符的非目標託管租用執行個體啟動。

在啟動執行個體時，您必須設定其租用。若在專用執行個體上啟動執行個體而未提供特定 `HostId`，該執行個體就會在啟用自動配置且符合其執行個體類型的任何專用執行個體上啟動。

主機親和性

您必須在執行個體層級設定主機親和性。它會建立執行個體和專用執行個體之間的啟動關係。

當親和性設為 `Host` 時，先前在特定主機上啟動的執行個體若曾停止，執行個體一律會在相同的主機上重新啟動。這適用於有訂定與未訂定目標的啟動。

當親和性設為 `Default` 且您停止並重新啟動執行個體時，它會在任何可用的主機上重新啟動。不過，它會嘗試在先前執行的專用執行個體上啟動 (以最佳做法為主)。

配置 Amazon EC2 專用主機供您的帳戶使用

若要開始使用專用主機，您必須先在您的帳戶進行分配。配置 專用執行個體 之後，會立即將 專用執行個體 容量提供給您的帳戶，而且您可以開始在 專用執行個體 上啟動執行個體。

在帳戶中配置專用執行個體時，您可以選擇能支援單一執行個體類型的組態，或相同執行個體系列中多種執行個體類型的組態。您可在主機上執行的執行個體數量將依您選擇的組態而定。如需更多資訊，請參閱Amazon EC2 專用主機執行個體容量配置。

Console

配置專用執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dedicated Hosts (專用執行個體)，然後選擇 Allocate 專用執行個體 (配置專用執行個體)。
3. 對於 Instance family (執行個體系列)，選擇適用於 專用執行個體 的執行個體系列。
4. 指定 專用執行個體 是否支援所選執行個體系列內的多種執行個體容量，或是僅支援特定執行個體類型。執行下列操作。
   - 若要將 專用執行個體 設定為支援所選執行個體系列內的多個執行類型，請針對 Support multiple instance types (支援多個執行個體類型) 選取 Enable (啟用)。啟用此功能，您就能在 專用執行個體 上啟動來自相同執行個體系列的不同執行個體容量。例如，如果您選擇 m5 執行個體系列且選擇了此選項，您就能在 專用執行個體 上啟動 m5.xlarge 及 m5.4xlarge。
   - 若要將 專用執行個體 設定為支援所選執行個體系列內的單一執行特定類型，請清除 Support multiple instance types (支援多個執行個體類型)，然後針對 Instance type (執行個體類型) 選取要支援的執行個體類型。這可讓您在 專用執行個體 上啟動單一執行個體類型。例如，如果您選擇此選項並指定 m5.4xlarge 作為支援的執行個體類型，則您僅能在 專用執行個體 上啟動 m5.4xlarge。
5. 對於 Availability Zone (可用區域)，選擇要配置 專用執行個體 的可用區域。
6. 若要允許 專用執行個體 接受符合其執行個體類型的未鎖定執行個體啟動，對於 Instance auto-placement (執行個體自動放置)，請選擇 Enable (啟用)。如需自動配置的詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。
7. 若要啟用 專用執行個體 的主機復原，對於 Host recovery (主機復原)，請選擇 Enable (啟用)。如需詳細資訊，請參閱Amazon EC2 專用主機復原。
8. 對於 Quantity (數量)，輸入要配置的 專用執行個體 數量。
9. (選用) 選擇新增標籤，然後輸入標籤金鑰和標籤值。
10. 選擇 Allocate (配置)。

AWS CLI

配置專用執行個體

使用 allocate-hosts 命令。下列範例會配置一個專用主機，其可支援 us-east-1a 可用區域中來自 m5 執行個體系列的多個執行個體類型。此操作還會啟用主機復原功能，並停用自動置放功能。

```
aws ec2 allocate-hosts \
    --instance-family "m5" \
    --availability-zone "us-east-1a" \
    --auto-placement "off" \
    --host-recovery "on" \
    --quantity 1
```

以下範例會配置一個專用主機，其可支援在指定可用區域中啟動之未訂定目標的執行個體並啟用主機復原，同時啟用主機復原功能與自動置放功能。

```
aws ec2 allocate-hosts \
    --instance-type "m5.large" \
    --availability-zone "eu-west-1a" \
    --auto-placement "on" \
    --host-recovery "on" \
    --quantity 1
```

PowerShell

配置專用執行個體

使用 New-EC2Host cmdlet。下列範例會配置一個專用主機，其可支援 us-east-1a 可用區域中來自 m5 執行個體系列的多個執行個體類型。主機也已啟用主機復原功能，而且已停用自動置放功能。

```
New-EC2Host `
    -InstanceFamily m5 `
    -AvailabilityZone us-east-1a `
    -AutoPlacement Off `
    -HostRecovery On `
    -Quantity 1
```

以下範例會配置一個專用主機，其可支援在指定可用區域中啟動之未訂定目標的執行個體並啟用主機復原。

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement On `
    -HostRecovery On `
    -Quantity 1
```

在 Amazon EC2 專用主機上啟動 Amazon EC2 執行個體

配置好專用執行個體之後，您就可以在其中啟動執行個體。如果您不具備作用中的專用執行個體，且其可用容量不足以啟動您的執行個體類型，您就無法啟動含 host 租用的執行個體。

考量事項

- 由 Amazon EC2 提供的 Windows、SUSE 和 RHEL AMI 無法搭配專用執行個體使用。
- 對於支援多種執行個體規模的專用執行個體，建議您先啟動較大的執行個體，然後視需要以較小的執行個體填滿剩餘的執行個體容量。
- 在您啟動執行個體之前，請記下以下的限制。如需詳細資訊，請參閱專用執行個體 限制。

Console

在專用執行個體頁面於特定專用執行個體啟動執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 在 Dedicated Hosts (專用執行個體) 頁面上，選取一個主機，然後選取 Actions (動作)、Launch Instance(s) onto Host (在主機啟動執行個體)。
4. 在 Application and OS Images (應用程式和 OS 映像) 區段中，從清單中選取 AMI。
5. 在 Instance type (執行個體類型) 區段中，選取要啟動的執行個體類型。

> **Note**
>
> 如果 專用執行個體 僅支援單一執行個體類型，則預設會選取支援的執行個體類型，而且不可變更。

如果專用執行個體支援多個執行個體類型，您必須根據專用執行個體的可用執行個體容量，選取支援執行個體系列內的執行個體類型。建議您先啟動較大的執行個體，然後視需要以較小的執行個體填滿剩餘的執行個體容量。

6. 在 Key pair (金鑰對) 區段中，選取要與執行個體建立關聯的金鑰對。
7. 在進階詳細資訊區段，針對租用親和性，請選擇下列其中一項：

   - 關閉 — 已停用主機親和性。在指定主機啟動的執行個體，不保證停止後會在相同的專用主機重新啟動。
   - 專用主機 ID — 已啟用主機親和性。如果停止，執行個體一律會在此指定主機重新啟動 (如果主機有容量)。如果主機沒有容量，則無法重新啟動執行個體；您必須與其他主機建立親和性。

   如需親和性的詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。

   > **Note**
   >
   > Tenancy (租用) 和 Host (主機) 選項會依據您選取的主機預先設定。

8. 視需要設定其餘的執行個體選項。如需詳細資訊，請參閱Amazon EC2 執行個體組態參數的參考。
9. 選擇啟動執行個體。

使用啟動執行個體精靈，在專用執行個體中啟動執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)、Launch Instance (啟動執行個體)。
3. 在 Application and OS Images (應用程式和 OS 映像) 區段中，從清單中選取 AMI。
4. 在 Instance type (執行個體類型) 區段中，選取要啟動的執行個體類型。
5. 在 Key pair (金鑰對) 區段中，選取要與執行個體建立關聯的金鑰對。
6. 在 Advanced details (進階詳細資料) 區段中，執行下列操作：

   a. 對於 Tenancy (租用)，選取 Dedicated Host (專用執行個體)。
   b. 對於 Target host by (目標主機依據) 中，選取 Host ID (主機 ID)。
   c. 對於 Target host ID (目標主機 ID)，選取要在其上啟動執行個體的主機。

d. 對於租用親和性，請選擇下列其中一項：

- 關閉 — 已停用主機親和性。在指定主機啟動的執行個體，不保證停止後會在相同的專用主機重新啟動。
- 專用主機 ID — 已啟用主機親和性。如果停止，執行個體一律會在此指定主機重新啟動 (如果主機有容量)。如果主機沒有容量，則無法重新啟動執行個體；您必須與其他主機建立親和性。

如需親和性的詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。

7. 視需要設定其餘的執行個體選項。如需詳細資訊，請參閱Amazon EC2 執行個體組態參數的參考。
8. 選擇啟動執行個體。

AWS CLI

將執行個體啟動至專用執行個體

使用 run-instance 命令，並在 `--placement` 選項中指定執行個體的親和性、租用和主機。

若要在具有主機親和性的特定專用主機上啟動 （ 如果停止，執行個體一律會在相同主機上重新啟動 ）：

```
--placement Affinity=host,Tenancy=host,HostId=h-07879acf49EXAMPLE
```

若要在沒有主機親和性的專用主機上啟動 （ 執行個體可以在任何可用的主機上重新啟動 ）：

```
--placement Tenancy=host,HostId=h-07879acf49EXAMPLE
```

若要在已啟用自動置放且符合執行個體類型的任何可用專用主機上啟動：

```
--placement Tenancy=host
```

PowerShell

將執行個體啟動至專用執行個體

使用 New-EC2Instance cmdlet，並在 `-Placement` 參數中指定執行個體的親和性、租用和主機。

若要在具有主機親和性的特定專用主機上啟動（如果停止，執行個體一律會在相同主機上重新啟動）：

```
-Placement_Affinity host `
-Placement_Tenancy host `
-Placement_HostId h-07879acf49EXAMPLE
```

若要在沒有主機親和性的專用主機上啟動（執行個體可以在任何可用的主機上重新啟動）：

```
-Placement_Tenancy host `
-Placement_HostId h-07879acf49EXAMPLE
```

若要在已啟用自動置放且符合執行個體類型的任何可用專用主機上啟動：

```
-Placement_Tenancy host
```

將 Amazon EC2 執行個體啟動至主機資源群組

專用主機也與 整合 AWS License Manager。您可以使用 License Manager 建立主機資源群組，這是做為單一實體管理的專用執行個體集合。建立主機資源群組時，您可以指定專用執行個體的主機管理偏好設定，例如自動配置和自動發佈。這讓您在無需手動配置和管理這些主機的情況下，在專用執行個體中啟動執行個體。如需詳細資訊，請參閱「AWS License Manager 使用者指南」中的主機資源群組。

當您將執行個體啟動至具有可用執行個體容量之專用執行個體的主機資源群組時，Amazon EC2 會將執行個體啟動到該主機上。如果主機資源群組沒有具有可用執行個體容量的主機，則 Amazon EC2 會自動在主機資源群組中配置新主機，然後將執行個體啟動到該主機上。如需詳細資訊，請參閱「AWS License Manager 使用者指南」中的主機資源群組。

需求與限制

- 為主機資源群組建立授權組態（自我管理授權）是選用的：
  - 如果您在建立主機資源群組時需要授權組態，您只能從具有相符核心或通訊端型授權組態的 AMIs 啟動執行個體。
  - 如果您建立的主機資源群組不需要授權組態，則不需要設定自我管理的授權或建立 AMI 的關聯。
- 您無法將 Amazon EC2 提供的 SQL Server、SUSE 或 RHEL AMI 與 專用執行個體 搭配使用。
- 您無法透過選擇主機 ID 鎖定特定主機為目標，而且在將執行個體啟動至主機資源群組時，也無法啟用執行個體親和性。

Console

將執行個體啟動至主機資源群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)、Launch Instance (啟動執行個體)。
3. 在 Application and OS Images (應用程式和 OS 映像) 區段中，從清單中選取 AMI。
4. 在 Instance type (執行個體類型) 區段中，選取要啟動的執行個體類型。
5. 在 Key pair (金鑰對) 區段中，選取要與執行個體建立關聯的金鑰對。
6. 在 Advanced details (進階詳細資料) 區段中，執行下列操作：
   a. 對於 Tenancy (租用)，選取 Dedicated Host (專用執行個體)。
   b. 對於 Target host by (目標主機依據)，選取 Host resource group (主機資源群組)。
   c. 對於 Tenancy host resource group (租用主機資源群組)，選取要在其中啟動執行個體的主機資源群組。
   d. 對於 Tenancy affinity (租用親和性)，執行下列其中一個操作：
      - 選取 Off (關閉) – 在指定主機上啟動的執行個體，不保證停止後會在相同的專用執行個體上重新啟動。
      - 選取專用執行個體 ID – 執行個體在停止後始終會在此特定主機上重新啟動。

   如需親和性的詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。
7. 視需要設定其餘的執行個體選項。如需詳細資訊，請參閱Amazon EC2 執行個體組態參數的參考。
8. 選擇啟動執行個體。

AWS CLI

將執行個體啟動至主機資源群組

使用 run-instances 命令。在 `--placement` 選項中，省略租用，並指定主機資源群組的 ARN。

```
--placement HostResourceGroupArn=arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

PowerShell

將執行個體啟動至主機資源群組

使用 New-EC2Instance cmdlet。在 `-Placement` 參數中，省略租用，並指定主機資源群組的 ARN。

```
-Placement_HostResourceGroupArn arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

修改現有 Amazon EC2 專用主機的自動配置設定

您可以在將專用主機配置到 AWS 您的帳戶之後，修改專用主機的自動置放設定。

Console

修改專用執行個體的自動配置

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 選取主機並選取 Actions (動作)、Modify host (修改主機)。
4. 對於 Instance auto-placement (執行個體自動配置)，選擇 Enable (啟用) 以啟用自動配置，或清除 Enable (啟用) 以停用自動配置。如需詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。
5. 選擇 Save (儲存)。

AWS CLI

修改專用執行個體的自動配置

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --auto-placement on \
    --host-ids h-012a3456b7890cdef
```

PowerShell

修改專用執行個體的自動配置

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -AutoPlacement 1 `
    -HostId h-012a3456b7890cdef
```

修改現有 Amazon EC2 專用主機支援的執行個體類型

您可以修改 專用執行個體 來變更其支援的執行個體類型。如果其目前支援單一執行個體類型，您可以進行修改，以支援該執行個體系統內的多個執行個體類型。同樣地，如果其目前支援多個執行個體類型，您可以進行修改，僅支援特定執行個體類型。

若要修改 專用執行個體 以支援多個執行個體類型，您必須先停止主機上所有執行的執行個體。此修改需要約 10 分鐘的時間完成。進行修改時，專用執行個體 會轉移為 pending 狀態。當執行個體處於 pending 狀態時，您無法開始已停止的執行個體，或是啟動 專用執行個體 上的新執行個體。

若要將支援多個執行個體類型的 專用執行個體 修改為僅支援單一執行個體類型，主機必須沒有執行中的執行個體，或是執行中的執行個體必須是您希望主機支援的執行個體類型。例如，若要將支援 m5 執行個體系列內多個執行個體類型的主機修改為僅支援 m5.large 執行個體，專用執行個體 必須沒有執行中的執行個體，或是其上必須只有 m5.large 個執行個體在執行中。

如果您為虛擬執行個體類型分配主機，則在分配主機之後，無法將執行個體類型修改為 .metal 執行個體類型。例如，如果您為 m5.large 執行個體類型分配主機，則無法將執行個體類型修改為 m5.metal。同樣，如果您為 .metal 執行個體類型分配主機，則在分配主機之後，無法將執行個體類型修改為虛擬執行個體類型。例如，如果您為 m5.metal 執行個體類型分配主機，則無法將執行個體類型修改為 m5.large。

Console

修改專用執行個體支援的執行個體類型

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dedicated Host (專用執行個體)。
3. 選取要修改的專用執行個體，然後選取 Actions (動作)、Modify host (修改主機)。
4. 根據專用執行個體的目前組態，執行下列其中一個動作：
   - 如果專用執行個體目前支援特定執行個體類型，則不會啟用 Support multiple instance types (支援多個執行個體類型)，且 Instance type (執行個體類型) 會列出支援的執行個體類型。

若要將主機修改為支援目前執行個體系列中的多個類型，請針對 Support multiple instance types (支援多個執行個體類型)，選取 Enable (啟用)。

您必須先停止主機上所有執行的執行個體，然後再將之修改為支援多個執行個體類型。

- 如果專用執行個體目前支援執行個體系列內的多個執行個體類型，請針對 Support multiple instance types (支援多個執行個體類型) 選取 Enabled (已啟用)。若要將主機修改為支援特定執行個體類型，請針對 Support multiple instance types (支援多個執行個體類型) 清除 Enable (啟用)，然後針對 Instance type (執行個體類型) 選取要支援的特定執行個體類型。

  您無法變更 專用執行個體 支援的執行個體系列。

5. 選擇 Save (儲存)。

AWS CLI

修改專用執行個體支援的執行個體類型

使用 modify-hosts 命令。

下列範例會修改專用主機以支援 m5 執行個體系列內的多個執行個體類型。

```
aws ec2 modify-hosts \
    --instance-family m5 \
    --host-ids h-012a3456b7890cdef
```

下列範例會修改專用主機，以便僅支援 m5.xlarge 執行個體。

```
aws ec2 modify-hosts \
    --instance-type m5.xlarge \
    --instance-family --host-ids h-012a3456b7890cdef
```

PowerShell

修改專用執行個體支援的執行個體類型

使用 Edit-EC2Host cmdlet。

下列範例會修改專用主機以支援 m5 執行個體系列內的多個執行個體類型。

```
Edit-EC2Host `
    -InstanceFamily m5 `
```

```
    -HostId h-012a3456b7890cdef
```

下列範例會修改專用主機，以便僅支援 m5.xlarge 執行個體。

```
Edit-EC2Host `
    -InstanceType m5.xlarge `
    -HostId h-012a3456b7890cdef
```

修改 Amazon EC2 執行個體的 Amazon EC2 專用主機租用和親和性

啟動執行個體之後，就可以變更執行個體的租用。您也可以修改執行個體的親和性以鎖定特定主機，或允許在帳戶中具有相符屬性的任何可用專用執行個體上啟動。若要修改執行個體租用或親和性，執行個體必須處於 stopped 狀態。

執行個體的作業系統詳細資訊 (以及是否已安裝 SQL Server) 會影響支援的轉換。如需執行個體可用租用轉換路徑的詳細資訊，請參閱「License Manager 使用指南」中的租用轉換。

> **Note**
>
> 若使用 T3 執行個體，您必須在專用執行個體上啟動執行個體，才能使用 host 的租用。使用 T3 執行個體時，您無法將租用從 host 變更為 dedicated 或 default。嘗試進行上述任一不受支援的租用變更時，會顯示 InvalidRequest 錯誤代碼。

Console

修改執行個體租用或親和性

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選取 Instances (執行個體)，再選取要修改的執行個體。
3. 選擇 Instance state (執行個體狀態)、Stop (停止)。
4. 在已選取執行個體的情況下，選擇動作、執行個體設定、修改執行個體的配置。
5. 在修改執行個體配置頁面，設定下列：
   - Tenancy (租用)— 選擇下列其中一項：
     - 執行專用硬體執行個體 – 以專用預留執行個體形式啟動執行個體。如需詳細資訊，請參閱Amazon EC2 專用執行個體。

- 在專用執行個體上啟動執行個體 – 在可設定親和性的專用執行個體上啟動執行個體。
- Affinity (親和性)— 選擇下列其中一項：
  - 此執行個體可在我的任一台主機上執行 – 執行個體會在您帳戶中任何支援其執行個體類型的可用專用執行個體上啟動。
  - 此執行個體僅能在所選的主機上執行 – 執行個體僅能在為 Target Host (目標主機) 選取的專用執行個體上執行。
- Target Host (目標主機)— 選取執行個體必須在其上執行的專用執行個體。如果沒有列出任何目標主機，您的帳戶中可能沒有可用且相容的專用執行個體。

如需詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。

6. 選擇 Save (儲存)。

AWS CLI

修改執行個體租用或親和性

使用 modify-instance-placement 命令。以下範例會將指定執行個體的親和性從 `default` 變更為 `host`，並指定執行個體要與哪個專用執行個體保有親和性。

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
    --affinity host \
    --tenancy host \
    --host-id h-012a3456b7890cdef
```

PowerShell

修改執行個體租用或親和性

使用 Edit-EC2InstancePlacement cmdlet。以下範例會將指定執行個體的親和性從 `default` 變更為 `host`，並指定執行個體要與哪個專用執行個體保有親和性。

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Affinity host `
    -Tenancy host `
    -HostId h-012a3456b7890cdef
```

釋出 Amazon EC2 專用主機

如果您不再需要專用主機，可以停止在主機上執行的執行個體，並指示它們在另一個主機上啟動，接著再釋出主機。

您必須先停止在專用執行個體上執行的任何執行個體，才能釋出主機。您可以將這些執行個體遷移到帳戶中的其他專用執行個體，以便繼續使用。這些步驟僅適用於隨需專用執行個體。

Console

釋出專用執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 在 專用執行個體 (專用執行個體) 頁面中，選取要釋出的專用執行個體。
4. 選擇 Actions (動作)、Release host (釋出主機)。
5. 若要確認，請選擇 Release (釋出)。

AWS CLI

釋出專用執行個體

使用 release-hosts 命令。

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

釋出專用執行個體

使用 Remove-EC2Host cmdlet。

```
Remove-EC2Host -HostId h-012a3456b7890cdef
```

在您釋出專用執行個體之後，即無法再次使用相同的主機或主機 ID，也不會再以隨需計費費率向您收費。專用執行個體的狀態會變更為 `released`，且您將無法在該主機上啟動任何執行個體。

> **Note**
>
> 如果您最近曾釋出專用執行個體，其可能需要一些時間才能停止計入限制。在這段期間，您可能會在嘗試配置新的專用執行個體時遇到 `LimitExceeded` 錯誤。若是這種情況，請嘗試在幾分鐘後再次配置新的主機。

停止的執行個體仍可供使用，並會列在 Instances (執行個體) 頁面中。這些執行個體仍保有其 `host` 租用設定。

遷移至 Nitro 型 Amazon EC2 專用主機

Nitro 系統結合一系列由 AWS 建立的硬體和軟體元件，能保障高效能、高可用性和高安全性。與 Xen 型專用主機相比，Nitro 型專用主機的性價比更高。如果您的帳戶中有任何 Xen 型專用主機，我們建議您將工作負載遷移到 Nitro 型專用主機。如需更多資訊，請參閱 AWS Nitro 系統。

若要從 Xen 型專用主機遷移到 Nitro 型專用主機，您需要將專用主機的 Xen 型執行個體遷移到 Nitro 型執行個體類型，配置新 Nitro 型專用主機，然後將遷移的 Nitro 型執行個體移動到新 Nitro 型專用主機。

本主題提供從 Xen 型專用主機遷移到 Nitro 型專用主機的詳細步驟。

移轉步驟

- 步驟 1：識別您的 Xen 型專用主機
- 步驟 2：將 Xen 型執行個體遷移至 Nitro 型執行個體類型
- 步驟 3：配置 Nitro 型專用主機
- 步驟 4：將已遷移的執行個體移至新 Nitro 型專用主機
- 步驟 5：釋出未用 Xen 型專用主機

步驟 1：識別您的 Xen 型專用主機

以下專用主機為 Xen 型，符合資格遷移到 Nitro 型專用主機。

- 一般用途：M3 | M4
- 運算最佳化：C3 | C4
- 記憶體最佳化：R3 | R4 | X1 | X1e
- 儲存最佳化：D2 | H1 | I2 | I3

- 加速運算：G3 | P3

檢查帳戶中是否有 Xen 型專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格，選擇專用主機。
3. 在搜尋欄位中，使用執行個體系列篩選條件來搜尋上述 Xen 型專用主機。例如，執行個體系列 = m3。

步驟 2：將 Xen 型執行個體遷移至 Nitro 型執行個體類型

在 Xen 型專用主機上執行的執行個體也是 Xen 型。您必須將這些執行個體遷移至 Nitro 型執行個體類型，才能將其移動至 Nitro 型專用主機。

> **Important**
>
> 在開始遷移執行個體之前，建議您備份資料。如需詳細資訊，請參閱從 Amazon EC2 執行個體建立多磁碟區 Amazon EBS 快照。

尋找在 Xen 型專用主機上執行的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格，選擇專用主機。
3. 選取要遷移的 Xen 型主機，然後選取執行中執行個體索引標籤。此索引標籤會列出所選主機執行的所有執行個體。

若要遷移 Linux 執行個體，請參閱 Amazon EC2 執行個體類型變更。

若要遷移 Windows 執行個體，請參閱 將 EC2 Windows 執行個體遷移至 Nitro 型執行個體類型。

> **Note**
>
> 請確保將執行個體遷移至與您要遷移到的 Nitro 型專用主機相符的執行個體類型。例如，如果您要遷移至 M7i 專用主機，請確定將執行個體遷移至 M7i 執行個體類型。

步驟 3：配置 Nitro 型專用主機

尋找支援的 Nitro 型專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格，選取執行個體類型。
3. 套用下列篩選條件：
   - Hypervisor = nitro
   - 專用主機支援 = true

找到合適的 Nitro 型執行個體類型後，請配置新專用主機。

步驟 4：將已遷移的執行個體移至新 Nitro 型專用主機

在配置 Nitro 型專用主機並達到 `available` 狀態後，您可以將先前遷移到 Nitro 型專用主機的執行個體類型移至新專用主機。

若要將執行個體移至新 Nitro 型專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格，執行個體。
3. 選取您遷移的執行個體，選擇動作、執行個體設定、修改執行個體的配置。
4. 對於目標專用主機，選擇新 Nitro 型專用主機，然後選擇儲存。
5. 重新啟動執行個體。選取執行個體，並選取執行個體狀態、停止執行個體。

步驟 5：釋出未用 Xen 型專用主機

將工作負載從 Xen 型專用主機遷移到新 Nitro 型專用主機後，如果不再需要，您可以釋出 Xen 型專用主機。

以專用主機帳單折扣購買專用主機保留

與隨需專用主機定價相比，專用主機保留可為您提供最高 70% 的折扣。您的帳戶必須配置作用中專用主機，然後才能購買專用主機保留。如需詳細資訊，請參閱Dedicated Host Reservations。

Console

購買保留

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 專用執行個體、專用執行個體預留、Purchase 專用執行個體預留 (購買 專用執行個體預留)。
3. 在尋找方案畫面，執行下列動作：
   a. 對於執行個體系列，請選擇要購買專用主機保留的專用主機執行個體系列。
   b. 在支付選項中，選擇並設定偏好支付選項。
4. 選擇下一步。
5. 選擇與專用主機保留相關的專用主機，然後選擇下一步。
6. (選用) 為專用主機保留指派標籤。
7. 檢閱您的訂單，然後選擇購買。

AWS CLI

購買保留

1. 使用 describe-host-reservation-offerings 命令，列出符合您需求的可用方案。下列範例列出的方案可支援 m4 執行個體系列中的執行個體，並具有一年期限。

   此期限以秒為單位指定。一年期限包括 31,536,000 秒，三年期限則為 94,608,000 秒。

   ```
   aws ec2 describe-host-reservation-offerings \
       --filter Name=instance-family,Values=m4 \
       --max-duration 31536000
   ```

   此命令會傳回符合您條件的方案清單。請記下要購買方案的 ID。
2. 使用 purchase-host-reservation 命令來購買方案，並提供上一個步驟中所記下的 `offeringId`。下列範例會購買指定的保留，並將其與 AWS 帳戶中已配置的特定專用主機建立關聯，並套用索引鍵為 `purpose`且值為 的標籤`production`。

   ```
   aws ec2 purchase-host-reservation \
       --offering-id hro-03f707bf363b6b324 \
       --host-id-set h-013abcd2a00cbd123 \
   ```

```
    --tag-specifications 'ResourceType=host-
reservation,Tags={Key=purpose,Value=production}'
```

PowerShell

購買保留

1. 使用 Get-EC2HostReservationOffering cmdlet，列出符合您需求的可用方案。下列範例列出的方案可支援 m5 執行個體系列中的執行個體，並具有一年期限。

   此期限以秒為單位指定。一年期限包括 31,536,000 秒，三年期限則為 94,608,000 秒。

   ```
   $filter = @{Name="instance-family"; Values="m5"}
   Get-EC2HostReservationOffering `
       -Filter $filter `
       -MaxDuration 31536000
   ```

   此命令會傳回符合您條件的方案清單。請記下要購買方案的 ID。

2. 使用 New-EC2HostReservation cmdlet 來購買方案，並提供上一個步驟中所記下的優惠 ID。下列範例會購買指定的保留，並將其與 AWS 帳戶中已配置的特定專用主機建立關聯。

   ```
   New-EC2HostReservation `
       -OfferingId hro-03f707bf363b6b324 `
       -HostIdSet h-013abcd2a00cbd123
   ```

跨帳戶 Amazon EC2 專用主機共用

專用主機共用可讓專用主機擁有者與其他 AWS 帳戶或 AWS 組織內的專用主機共用。這可讓您集中建立和管理專用主機，並在多個 AWS 帳戶或 AWS 組織內共用專用主機。

在此模型中，擁有專用主機 (擁有者) AWS 的帳戶會與其他 AWS 帳戶 (消費者） 共用。消費者可以在與他們共用的專用執行個體中啟動執行個體，就好像他們在自己的帳戶中配置的專用執行個體中啟動執行個體一樣。擁有者負責管理專用執行個體和他們在其中啟動的執行個體。擁有者無法修改消費者在共用的專用執行個體上啟動的執行個體。消費者負責管理在分享給他們的專用執行個體中啟動的執行個體。消費者無法檢視或修改其他消費者或專用執行個體擁有者所擁有的執行個體，也無法修改與他們共用的專用執行個體。

專用執行個體擁有者可以將專用執行個體分享給下列對象：

- AWS 組織內部或外部的特定 AWS 帳戶
- AWS 組織內的組織單位
- 其整個 AWS 組織

目錄

- 共用 專用執行個體 的先決條件
- 共用 專用執行個體 的限制
- 相關服務
- 跨可用區域共用
- 共用的 專用執行個體 許可
- 計費和計量
- 專用執行個體 限制
- 主機復原和 專用執行個體 共用
- 跨 AWS 帳戶共用 Amazon EC2 專用主機
- 取消共用與其他 AWS 帳戶共用的專用主機
- 檢視您 AWS 帳戶中的共用 Amazon EC2 專用主機

共用 專用執行個體 的先決條件

- 若要共用專用主機，您必須在 AWS 帳戶中擁有該主機。您無法將已分享給您的專用執行個體再分享出去。
- 若要與 AWS 組織或 AWS 組織中的組織單位共用專用主機，您必須啟用與 共用 AWS Organizations。如需詳細資訊，請參閱 AWS RAM 使用者指南中的透過 AWS Organizations啟用共用。

共用 專用執行個體 的限制

您無法共用已針對下列執行個體類型配置的 專用執行個體：`u-6tb1.metal`、`u-9tb1.metal`、`u-12tb1.metal`、`u-18tb1.metal` 和 `u-24tb1.metal`。

相關服務

AWS Resource Access Manager

專用主機共用與 AWS Resource Access Manager (AWS RAM) 整合。 AWS RAM 是一項服務，可讓您與任何 AWS 帳戶或透過 共用 AWS 資源 AWS Organizations。透過 AWS RAM，您可以透過建立資源共用來共用您擁有的資源。資源共享指定要共用的資源，以及共用它們的消費者。消費者可以是個別 AWS 帳戶、組織單位或整個組織的來源 AWS Organizations。

如需 的詳細資訊 AWS RAM，請參閱AWS RAM 《 使用者指南》 。

跨可用區域共用

為了確保資源分散到區域的可用區域，可用區域會獨立對應到每個帳戶的名稱。這可能導致帳戶之間的可用區域命名出現差異。例如，`us-east-1a`您 AWS 帳戶的可用區域可能沒有與`us-east-1a`另一個 AWS 帳戶相同的位置。

若要基於您的帳戶來識別專用執行個體的相對位置，您必須使用可用區域 ID (AZ ID)。可用區域 ID 是所有 AWS 帳戶的可用區域唯一且一致的識別符。例如，`use1-az1` 是 `us-east-1` 區域的可用區域 ID，它在每一個 AWS 帳戶的位置都相同。

檢視您帳戶中可用區域的可用區域 ID

1. 在 https：//https://console.aws.amazon.com/ram/home 開啟 AWS RAM 主控台。
2. 畫面右側的 Your AZ ID (您的 AZ ID) 面板中會顯示目前區域的可用區域 ID。

共用的 專用執行個體 許可

擁有者的許可

擁有者負責管理他們共用的專用執行個體以及在其上啟動的執行個體。擁有者可以檢視在共用的專用執行個體上執行的所有執行個體，包括消費者啟動的執行個體。不過，擁有者無法對消費者啟動的執行個體採取任何動作。

消費者的許可

消費者負責管理他們在共用的專用執行個體上啟動的執行個體。消費者完全無法修改共用的專用執行個體，也無法檢視或修改其他消費者或專用執行個體擁有者所擁有的執行個體。

計費和計量

共用專用執行個體無須額外收費。

擁有者會按他們共用的專用執行個體計費。對於消費者在共用的專用執行個體上啟動的執行個體則不計費。

專用執行個體預留會繼續為共用的專用執行個體提供帳單折扣。只有專用執行個體擁有者可以為他們擁有的共用專用執行個體購買專用執行個體預留。

專用執行個體 限制

共用的專用執行個體只會計入擁有者的專用執行個體限制。消費者的專用執行個體限制不受已與他們共用的專用執行個體所影響。同樣地，消費者在共用的專用執行個體上啟動的執行個體不會計入他們的執行個體限制。

主機復原和 專用執行個體 共用

主機復原可復原專用執行個體擁有者和其所共用之消費者啟動的執行個體。替換專用執行個體會配置到擁有者的帳戶。它會新增至與原始專用執行個體相同的資源共享，並與相同的消費者共用。

如需詳細資訊，請參閱Amazon EC2 專用主機復原。

跨 AWS 帳戶共用 Amazon EC2 專用主機

當擁有者分享專用執行個體，它可讓消費者在主機上啟動執行個體。在共用主機上可用容量所允許下，消費者可以在當中啟動任意數量的執行個體。

> **Important**
>
> 請注意，您應負責確保您擁有適當的授權，可共用您的 專用執行個體 上的任何 BYOL 授權。

如果您在啟用自動配置時共用專用執行個體，請記住下列事項，因為這可能會導致意外的專用執行個體使用量：

- 如果消費者使用專用執行個體租用啟動執行個體，但他們的帳戶中擁有的專用執行個體上沒有容量，則該執行個體會自動啟動到共用的專用執行個體。

若要分享專用執行個體，您必須將它新增至資源共享。資源共用是一種 AWS RAM 資源，可讓您跨 AWS 帳戶共用資源。資源共享指定要分享的資源，以及共用它們的消費者。您可以將專用執行個體新增至現有的資源，或將其新增至新的資源共享。

如果您是 中組織的一部分， AWS Organizations 且已啟用組織內的共用，則組織中的消費者會自動獲得共用專用主機的存取權。否則，消費者會收到加入資源共享的邀請，並且在接受邀請後便能存取共用的專用執行個體。

> **Note**
>
> 分享專用執行個體後，消費者可能需要幾分鐘的時間才能存取它。

Console

如何使用 Amazon EC2 主控台共用您擁有的專用執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 選擇專用執行個體來共用，然後選擇 Actions (動作) > Share host (共用主機)。
4. 選取資源共享以新增專用執行個體，然後選取 Share host (共用主機)。

   可能需要經過幾分鐘，消費者才能存取共用的主機。

使用 AWS RAM 主控台共享您擁有的專用主機

請參閱《AWS RAM 使用者指南》中的建立資源共享。

AWS CLI

共用您擁有的專用主機

使用 create-resource-share 命令。

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

PowerShell

共用您擁有的專用主機

使用 New-RAMResourceShare cmdlet。

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

取消共用與其他 AWS 帳戶共用的專用主機

專用執行個體擁有者可以隨時取消共享已共用的專用執行個體。當您取消共享已共用的專用執行個體時，請注意以下規則：

- 與其共用專用執行個體的消費者無法再於其上啟動新的執行個體。
- 取消共用時在專用執行個體上執行中的消費者所擁有的執行個體會繼續執行，但排定進行淘汰。消費者會收到執行個體的淘汰通知，而且他們有兩週的時間可對通知採取行動。但是，如果在淘汰通知期間內與消費者重新共用專用執行個體，則會取消執行個體淘汰。

若要取消共享您擁有的已共用專用執行個體，您必須從資源共享中移除它。

Console

取消共用您擁有的已共用專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 選擇要取消共享的專用執行個體，然後選擇 Sharing (共用) 索引標籤。
4. Sharing (共用) 索引標籤列出已新增專用執行個體的資源共享。選取要從中移除專用執行個體的資源共享，然後選取 Remove host from resource share (從資源共享中移除主機)。

使用 AWS RAM 主控台取消共用您擁有的共享專用主機

請參閱《AWS RAM 使用者指南》中的更新資源共用。

AWS CLI

取消共用您擁有的已共用專用主機

使用 disassociate-resource-share 命令。

```
aws ram disassociate-resource-share \
```

```
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE
```

PowerShell

取消共用您擁有的已共用專用主機

使用 Disconnect-RAMResourceShare cmdlet。

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:dedicated-
host/h-07879acf49EXAMPLE"
```

檢視您 AWS 帳戶中的共用 Amazon EC2 專用主機

您可檢視與其他帳戶共用的專用主機，以及與您共用的專用主機。如果您擁有專用主機，則可以看到主機執行的所有執行個體，包括消費者啟動的執行個體。如果專用主機與您共用，您只能看到您在共用主機上啟用的執行個體，而不能看到其他消費者啟用的執行個體。

擁有者和消費者可以使用下列其中一種方法，來識別共用的專用執行個體。

Console

確定共用的專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。畫面會列出您擁有的專用執行個體和已分享給您的專用執行個體。
3. Owner (擁有者) 資料欄會顯示專用執行個體擁有者的 AWS 帳戶 ID。
4. 若要檢視主機上執行的執行個體，請選取執行個體索引標籤。

AWS CLI

確定共用的專用主機

使用 describe-hosts 命令。此命令會傳回您擁有的專用執行個體且和已分享給您的專用執行個體。0wner 的值是專用主機擁有者的帳戶 ID。此 Instances 清單說明在主機上執行的執行個體。

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

PowerShell

確定共用的專用主機

使用 Get-EC2Host cmdlet。此 cmdlet 會傳回您擁有的專用主機且和已分享給您的專用主機。回應中 0wner 的值是專用主機擁有者的帳戶 ID。此 Instances 清單說明在主機上執行的執行個體。

```
Get-EC2Host -Filter @{Name="state"; Values="available"}
```

上的 Amazon EC2 專用主機 AWS Outposts

AWS Outposts 是一種全受管服務，可將 AWS 基礎設施、服務、APIs 和工具擴展到您的內部部署。透過提供 AWS 受管基礎設施的本機存取權， AWS Outposts 可讓您使用與 AWS 區域相同的程式設計界面在內部部署中建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。

Outpost 是部署在客戶站點的 AWS 運算和儲存容量集區。 會在 AWS 區域中 AWS 操作、監控和管理此容量。

您可以在帳戶中擁有的 Outpost 上配置專用執行個體。這讓您可以更輕鬆地將需要專用實體伺服器的現有軟體授權和工作負載帶至 AWS Outposts。您也可以鎖定 Outpost 上的特定硬體資產，協助將工作負載之間的延遲降到最低。

專用執行個體可讓您在 Amazon EC2 上使用符合條件的軟體授權，進而獲得使用自有授權的靈活性和成本效益。綁定至虛擬機器、通訊端或實體核心的其他軟體授權也可以在專用執行個體上使用，但需遵守其授權條款。雖然 Outpost 一直是符合 BYOL 工作負載條件的單一租用戶環境，但專用執行個體允許您將所需的授權限制為單個主機，而不是整個 Outpost 部署。

此外，在 Outpost 上使用專用執行個體可為您在執行個體類型部署方面提供更大的靈活性，並對執行個體放置進行更精細的控制。您可以針對特定主機進行執行個體啟動，並使用主機親和性來確保執行個體始終在該主機上執行，您也可以使用自動放置，在具有相符組態和可用容量的任何可用主機上啟動執行個體。

內容

- 先決條件
- 支援的功能
- 考量事項
- 在 上配置 Amazon EC2 專用主機 AWS Outposts

先決條件

您的站點必須安裝 Outpost。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立 Outpost 並訂購 Outpost 容量。

支援的功能

- 支援下列執行個體系列：
  - 一般用途：M5 | M5d | M7i | M8i
  - 運算最佳化：C5 | C5d | C7i | C8i
  - 記憶體最佳化：R5 | R5d | R7i | R8i
  - 儲存最佳化：I3en
  - 加速運算：G4dn
- Outpost 上的專用執行個體可設定為支援多種執行個體規模。下列執行個體系列支援多種執行個體大小。
  - 一般用途：M5 | M5d | M7i
  - 運算最佳化：C5 | C5d | C7i
  - 記憶體最佳化：R5 | R5d | R7i

  如需詳細資訊，請參閱Amazon EC2 專用主機執行個體容量配置。
- Outpost 上的專用執行個體支援自動放置和鎖定執行個體啟動。如需詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。
- Outpost 上的專用執行個體支援主機親和性。如需詳細資訊，請參閱Amazon EC2 專用主機自動配置和主機親和性。
- Outposts 上的專用主機支援與 共用 AWS RAM。如需詳細資訊，請參閱跨帳戶 Amazon EC2 專用主機共用。

考量事項

- Outpost 上不支援專用執行個體保留功能。

- Outposts AWS License Manager 不支援主機資源群組 和 。
- Outpost 上的專用執行個體不支援爆量 T3 執行個體。
- Outpost 上的專用執行個體不支援主機復原。
- 對於在 Outposts 上具有專用主機租用的執行個體，不支援簡化自動復原。

在 上配置 Amazon EC2 專用主機 AWS Outposts

您在 Outpost 上配置和使用專用執行個體的方式與在 AWS 區域中使用專用執行個體的方式相同。

先決條件

在 Outpost 上建立子網。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立子網。

若要在 Outpost 中配置專用執行個體，請使用下列其中一種方法：

Console

使用 AWS Outposts 主控台在 Outpost 上配置專用主機

1. 在 https：//https://console.aws.amazon.com/outposts/ 開啟 AWS Outposts 主控台。
2. 在導覽窗格中，選擇 Outposts。選取 Outpost，然後選取 Actions (動作)、Allocate Dedicated Host (配置專用執行個體)。
3. 視需設定專用執行個體。如需詳細資訊，請參閱配置 Amazon EC2 專用主機供您的帳戶使用。

   > **Note**
   >
   > Availability Zone (可用區域) 和 Outpost ARN 應使用選定 Outpost 的可用區域和 ARN 預先填充。

4. 選擇 Allocate (配置)。

使用 Amazon EC2 主控台在 Outpost 上配置專用主機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dedicated Hosts (專用執行個體)，然後選擇 Allocate Dedicated Host (配置專用執行個體)。
3. 對於 Availability Zone (可用區域)，選取與 Outpost 關聯的可用區域。
4. 對於 Outpost ARN，輸入 Outpost 的 ARN。

5. 若要鎖定 Outpost 上的特定硬體資產，請選擇 啟用 鎖定 Outpost 上的特定硬體資產。針對要鎖定的每項硬體資產，請選擇 新增資產 ID，然後輸入硬體資產的 ID。

   > **Note**
   >
   > 您為 數量 指定的值必須等於所指定的資產 ID 的數量。例如，如果您指定 3 個資產 ID，則數量也必須為 3。

6. 視需進行其餘的專用執行個體設定。如需詳細資訊，請參閱配置 Amazon EC2 專用主機供您的帳戶使用。
7. 選擇 Allocate (配置)。

AWS CLI

在 Outpost 上配置專用主機

使用 allocate-hosts 命令。對於 `--availability-zone`，指定與 Outpost 關聯的可用區域。對於 `--outpost-arn`，指定 Outpost 的 ARN。您也可以針對 `--asset-ids` 選擇指定要鎖定的 Outpost 硬體資產的 ID。

```
aws ec2 allocate-hosts \
    --availability-zone "us-east-1a" \
    --outpost-arn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" \
    --asset-ids asset_id \
    --instance-family "m5" \
    --auto-placement "off" \
    --quantity 1
```

PowerShell

在 Outpost 上配置專用主機

使用 New-EC2Host cmdlet。指定與 Outpost 關聯的可用區域。您也可以針對 `-AssetId` 選擇指定要鎖定的 Outpost 硬體資產的 ID。

```
New-EC2Host `
    -AvailabilityZone "us-east-1a" `
    -OutpostArn "arn:aws:outposts:us-east-1a:111122223333:outpost/
op-4fe3dc21baEXAMPLE" `
    -AssetId asset_id `
```

```
    -InstanceFamily "m5" `
    -AutoPlacement "off" `
    -Quantity 1
```

將執行個體啟動至 Outpost 上的專用執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。選取您在上一步驟中配置的專用執行個體，然後選取 Actions (動作)、Launch instance onto host (在主機上啟動執行個體)。
3. 根據需要設定執行個體，然後啟動執行個體。如需詳細資訊，請參閱在 Amazon EC2 專用主機上啟動 Amazon EC2 執行個體。

Amazon EC2 專用主機復原

如果在您的專用執行個體上偵測到特定問題狀況，專用執行個體自動復原會自動將您的執行個體重新啟動到新的替代主機。若有關於系統電源或網路連線事件的非預期專用執行個體失敗，主機復原可減少手動介入的需要，並降低操作負擔。其他專用執行個體問題將需要手動介入才能從中復原。

目錄

- Amazon EC2 專用主機復原的運作方式
- 支援的執行個體類型
- 定價
- 管理 Amazon EC2 專用主機復原
- 檢視 Amazon EC2 專用主機的主機復原設定
- 手動復原 Amazon EC2 專用主機復原不支援的執行個體

Amazon EC2 專用主機復原的運作方式

專用執行個體和主機資源群組復原程序會使用主機層級運作狀態檢查，來存取專用執行個體可用性，以及偵測基礎系統失敗。專用執行個體失敗的類型決定專用執行個體自動復原是否可行。導致主機層級運作狀態檢查失敗的問題範例包括：

- 網路連線中斷
- 系統電力中斷
- 實體主機上的硬體或軟體問題

> **Important**
>
> 主機排定為淘汰時，不會發生專用執行個體自動復原。

專用執行個體自動復原

當您的專用主機偵測到系統電源或網路連線故障時，將啟動專用主機自動復原，並且 Amazon EC2 會自動在與原始專用主機相同的可用區中指派替換專用主機。替代的 專用執行個體 會收到新的主機 ID，但保留與原始 專用執行個體 相同的屬性，包括：

- 可用區域
- 執行個體類型
- 標籤
- 自動配置設定
- 保留

當配置替代的專用執行個體時，會將執行個體復原至替代的專用執行個體。復原的執行個體會保留與原始執行個體相同的屬性，包括：

- 執行個體 ID
- 私有 IP 地址
- 彈性 IP 地址
- EBS 磁碟區連接
- 所有執行個體中繼資料

此外，與 AWS License Manager 的內建整合會自動追蹤和管理授權。

> **Note**
>
> AWS 只有在可使用 License Manager 的區域才支援 AWS License Manager 整合。

如果執行個體與受損的 專用執行個體 具有主機親和性關係，復原的執行個體會與替代的 專用執行個體 建立主機親和性。

當所有執行個體都已復原至替代的 專用執行個體 上時，受損的 專用執行個體 會被釋出，並且替代的 專用執行個體 會變得可供使用。

啟動主機復原時，系統會透過電子郵件和 AWS Health 儀板表 事件來通知 AWS 帳戶擁有者。在主機復原已成功完成時，會傳送第二個通知。

如果您使用 AWS License Manager 來追蹤授權， AWS License Manager 會根據授權組態限制，為替代的專用主機配置新的授權。如果授權組態具有因主機復原而違反的硬性限制，則不允許執行復原程序，而且您會透過 Amazon SNS 通知收到主機復原失敗的通知 （如果已為 AWS License Manager 設定通知設定）。如果授權組態具有將因為主機復原而違反的軟性限制，將會允許繼續復原，並且將透過 Amazon SNS 通知來通知您相關的限制違反。如需詳細資訊，請參閱 AWS License Manager 使用者指南中的使用授權組態和 License Manager 中的設定。

主機復原狀態

偵測到專用執行個體失敗時，受損的專用執行個體會進入 `under-assessment` 狀態，並且所有執行個體會進入 `impaired` 狀態。當受損的 專用執行個體 處於 `under-assessment` 狀態時，您無法對其啟動執行個體。

配置替代的 專用執行個體 之後，它會進入 `pending` 狀態。在主機復原程序完成之前，它會保持在此狀態。當替代的 專用執行個體 處於 `pending` 狀態時，您無法對其啟動執行個體。在復原程序期間，替代的 專用執行個體 上復原的執行個體會維持為 `impaired` 狀態。

在主機復原完成之後，替代的 專用執行個體 會進入 `available` 狀態，而復原的執行個體會回到 `running` 狀態。當替代的 專用執行個體 進入 `available` 狀態時，您可以對其啟動執行個體。原始受損的 專用執行個體 會永久釋出並且進入 `released-permanent-failure` 狀態。

如果受損的專用主機 具有不支援主機復原的執行個體，例如具備執行個體儲存體根磁碟區的執行個體，則不會釋出專用主機。而是會將它標記為供淘汰，並進入 `permanent-failure` 狀態。

沒有專用執行個體自動復原的案例

主機排定為淘汰時，不會發生專用執行個體自動復原。您會在 中收到淘汰通知 AWS Health 儀板表、Amazon CloudWatch 事件，而 AWS 帳戶擁有者電子郵件地址會收到有關專用主機故障的訊息。請在指定的時間段內遵循淘汰通知中所述的補救步驟，以手動復原淘汰主機中的執行個體。

已停止的執行個體不會復原到替代的 專用執行個體 上。如果您嘗試啟動以受損的 專用執行個體 為目標的停止的執行個體，該執行個體啟動會失敗。建議您將已停止的執行個體修改為以不同的 專用執行個體 為目標，或是在具有相符組態並且已啟用自動配置的任何可用的 專用執行個體 上啟動。

具有執行個體儲存體的執行個體不會復原到替代的專用執行個體上。作為補救措施，會將受損的 專用執行個體 標示為供淘汰，並且您會在主機復原完成之後收到淘汰通知。請在指定的時間期間內遵循淘汰通知中所述的補救步驟，以手動復原受損的 專用執行個體 上其餘的執行個體。

支援的執行個體類型

下列執行個體系列支援主機復原：

- 一般用途：A1 | M3 | M4 | M5 | M5n | M5zn | M6a | M6g | M6i | T3 | Mac1 | Mac2 | Mac2-m1ultra | Mac2-m2 | Mac2-m2pro | M6in | M7a | M7g | M7i | Mac-m4 | Mac-m4pro | Mac-m4max | Mac-m3ultra
- 運算最佳化：C3 | C4 | C5 | C5n | C6a | C6g | C6i | C6gn | C6in | C7g | C7gn | C7i
- 記憶體最佳化：R3 | R4 | R5 | R5b | R5n | R6g | R6i | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | X1 | X1e | X2iezn | R6a | R6in | R7g | R7i | R7iz
- 加速運算：Inf1 | G3 | G5g | P3

若要復原不受支援的執行個體，請參閱手動復原 Amazon EC2 專用主機復原不支援的執行個體。

> **Note**
>
> 支援的金屬執行個體類型的專用執行個體自動復原將需要比非金屬執行個體類型更長的時間來偵測和復原。

定價

使用主機復原沒有額外的費用，但適用一般的 專用執行個體 費用。如需詳細資訊，請參閱 Amazon EC2 專用執行個體定價。

一旦初始化主機復原，將不再就受損的 專用執行個體 向您收費。替代專用執行個體的計費方式只會在進入 `available` 狀態之後開始。

如果受損的 專用執行個體 是使用隨需費率計費，則替代的 專用執行個體 也會使用隨需費率計費。如果受損的 專用執行個體 有作用中 專用執行個體預留，則會傳輸到替代的 專用執行個體。

管理 Amazon EC2 專用主機復原

如果在您的專用執行個體上偵測到特定問題狀況，專用執行個體自動復原會自動將您的執行個體重新啟動到新的替代主機。您可以在指派專用主機時或指派後啟用主機復原。

使用以下程序在配置主機時啟用主機復原。

Console

在配置時啟用主機復原

使用 Amazon EC 2主控台配置專用主機時，若要進行主機復原，請選擇啟用。如需詳細資訊，請參閱配置 Amazon EC2 專用主機供您的帳戶使用。

AWS CLI

在配置時啟用主機復原

使用 allocate-hosts 命令。

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

PowerShell

在配置時啟用主機復原

使用 New-EC2Host cmdlet。

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement on `
    -HostRecovery on `
    -Quantity 1
```

使用以下程序來管理專用主機的主機復原。

Console

在配置後管理主機復原

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用主機。

3. 選取專用主機。
4. 選擇動作、修改主機。
5. 若要復原主機，請選取或清除啟用。
6. 選擇儲存。

AWS CLI

在配置之後啟用主機復原

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-recovery on \
    --host-ids h-012a3456b7890cdef
```

在配置之後停用主機復原

使用 modify-hosts 命令，並指定 host-recovery 參數的值為 off。

```
aws ec2 modify-hosts \
    --host-recovery off \
    --host-ids h-012a3456b7890cdef
```

PowerShell

在配置之後啟用主機復原

使用 Edit-host cmdlet。

```
Edit-EC2Host `
    -HostRecovery on `
    -HostId h-012a3456b7890cdef
```

在配置之後停用主機復原

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -HostRecovery off `
    -HostId h-012a3456b7890cdef
```

檢視 Amazon EC2 專用主機的主機復原設定

您可以隨時檢視 專用執行個體 的主機復原組態。

Console

檢視專用主機的主機復原組態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 選取 專用執行個體，並在 Description (描述) 索引標籤中，檢閱 Host Recovery (主機復原) 欄位。

AWS CLI

檢視專用主機的主機復原組態

使用 describe-hosts 命令。

```
aws ec2 describe-hosts \
    --host-ids h-012a3456b7890cdef \
    --query Hosts[].HostRecovery
```

以下為範例輸出。

```
on
```

PowerShell

檢視專用主機的主機復原組態

使用 Get-EC2Host cmdlet。

```
(Get-EC2Host -HostId h-012a3456b7890cdef).Hosts | Select HostRecovery
```

以下為範例輸出。

```
HostRecovery
------------
```

on

手動復原 Amazon EC2 專用主機復原不支援的執行個體

主機復原不支援復原使用執行個體存放區磁碟區的執行個體。遵循以下的指示來手動復原您無法自動復原的任何執行個體。

> **Warning**
>
> 位在執行個體存放區磁碟區的資料都會在執行個體停止、休眠或終止時遺失。這包含連接到擁有 EBS 根磁碟區之執行個體的執行個體儲存體磁碟區。若要保護執行個體存放區磁碟區的資料，請在執行個體停止或終止之前，將資料備份至持久性儲存。

手動復原 EBS 後端執行個體

對於無法自動復原的 EBS 後端執行個體，建議您手動停止並啟動執行個體，以將它們復原至新的 專用執行個體。如需停止您執行個體的詳細資訊，以及在停止時您執行個體組態中發生的變更，請參閱停止和啟動 Amazon EC2 執行個體。

手動復原具有執行個體儲存體根磁碟區的執行個體。

對於無法自動復原、具有執行個體儲存體根磁碟區的執行個體，建議您執行下列：

1. 從您的最近的 AMI 的新 專用執行個體 上啟動替代的執行個體。
2. 將所有必要資料遷移至替代的執行個體。
3. 終止受損的 專用執行個體 上的原始執行個體。

Amazon EC2 專用主機的主機維護

透過主機維護，在專用主機降級的罕見情況下，Amazon EC2 會自動將在其上執行的執行個體遷移至運作狀態良好的替代專用主機。這有助盡量減少工作負載的停機時間，並簡化專用主機的管理。也會針對規劃和例行的 Amazon EC2 維護執行主機維護。

Amazon EC2 支援兩種類型的主機維護：

- 即時移轉主機維護 – 執行個體會在 24 小時內自動移轉到替換主機，無需停止和重新啟動。
- 重新啟動型主機維護 – 會安排執行個體進行執行個體重新啟動排程事件，在此過程中，執行個體將自動停止並在替代主機上重新啟動。

目錄

- 主機維護與主機復原
- 考量事項
- 相關服務
- 定價
- Amazon EC2 專用主機的主機維護運作方式
- 配置 Amazon EC2 專用主機的主機維護設定

主機維護與主機復原

下表顯示主機復原與主機維護之間的主要差異。

	主機復原	主機維護
執行個體連線能力	無法連接	可連接
專用主機狀態	`under-assessment`	`permanent-failure`
主機資源群組	支援	不支援

如需有關主機復原的詳細資訊，請參閱主機復原。

考量事項

- 所有 都提供主機維護 AWS 區域，但中國區域和 除外 AWS GovCloud (US) Regions。
- AWS Outposts、 AWS 本機區域和 AWS Wavelength 區域不支援主機維護。
- 對於已在主機資源群組中的主機，無法開啟或關閉主機維護。新增至主機資源群組的主機會保留其主機維護設定。如需詳細資訊，請參閱主機資源群組。
- 下列執行個體類型不支援主機維護，因為具有執行個體儲存體根磁碟區：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

相關服務

專用主機與 AWS License Manager 整合 - 跨 Amazon EC2 專用主機追蹤授權 （僅在提供 AWS License Manager 的區域中支援）。如需詳細資訊，請參閱 AWS License Manager 使用者指南。

您的 中必須有足夠的授權 AWS 帳戶 ，才能用於新的專用主機。當主機在排定的事件完成後釋出時，會釋出與降級主機相關聯的授權。

定價

使用主機維護沒有額外的費用，但通常收取專用執行個體費用。如需詳細資訊，請參閱 Amazon EC2 專用執行個體定價。

啟動主機維護之後，將不再為降級專用執行個體付費。替代專用執行個體的計費方式只會在進入 `available` 狀態之後開始。

如果降級的專用主機是使用隨需費率計費，則替代的專用主機也會使用隨需費率計費。如果降級的專用執行個體有作用中專用執行個體預留，則會傳輸到新專用執行個體。

Amazon EC2 專用主機的主機維護運作方式

在已啟用主機維護的專用主機上偵測到降級時，系統會自動在您的帳戶中配置替代的專用主機。替代的專用執行個體 會收到新的主機 ID，但保留與原始 專用執行個體 相同的屬性，包括：

- 自動配置設定
- 可用區域
- 專用主機保留關聯
- 主機親和性
- 主機維護設定
- 主機復原設定
- 執行個體類型
- Tags (標籤)

配置替代主機之後，會根據執行個體，使用即時遷移主機維護或重新啟動型主機維護來遷移執行個體。

當已降級的主機不再有執行個體後，其會從您的帳戶中永久釋出。

即時遷移主機維護

需要即時移轉主機維護的執行個體會在 24 小時內自動移轉到替代主機，無需停止和重新啟動。遷移後的執行個體會保留其現有的屬性，包括：

- 執行個體 ID

- 執行個體中繼資料
- Amazon EBS 磁碟區連接
- 彈性 IP 位址及私有 IP 位址
- 記憶體、CPU 與網路狀態

在遷移過程中，某些較大執行個體大小可能會出現輕微效能下降。

執行個體自動遷移至替代主機後，您會收到電子郵件和 AWS Health 儀表板通知。通知包括降級和替換主機 ID、使用即時移轉主機維護自動移轉的執行個體相關資訊，以及有關其餘執行個體的資訊。

重新啟動型主機維護

需要重新開機型主機維護的執行個體會安排於自通知之日起 14 天內進行重新開機排程事件。您可在排程事件前繼續存取已降級專用主機的執行個體。

您可將重新開機事件重新排程至原定事件日期和時間起 7 天內的日期。如需詳細資訊，請參閱重新排程 EC2 執行個體的排程事件。

Amazon EC2 會自動為這些執行個體預留替換主機容量。您無法以此預留容量執行執行個體。

Amazon EC2 主控台會將預留容量顯示為已用容量。執行個體可能似乎同時在降級主機和替代主機上執行。不過，執行個體將僅繼續在已降級的主機上執行，直到停止執行或遷移至替代主機的預留容量為止。

在排程事件的日期與時間，執行個體會自動停止並重新啟動至替代主機的預留容量。遷移後的執行個體會保留其現有的屬性，包括：

- 執行個體 ID
- 執行個體中繼資料
- Amazon EBS 磁碟區連接
- 彈性 IP 位址及私有 IP 位址

然而，由於執行個體在遷移期間會停止並重新啟動，因此不會保留其記憶體、CPU 和網路狀態。

您也可以在排程事件前隨時手動停止並重新啟動這些執行個體，將其遷移到替代主機或其他主機。您可能需要修改執行個體的主機親和性，才能在不同主機上重新啟動。如果您在排程事件之前停止執行個體，則會釋出替代主機的預留容量，且其將可供使用。

主機維護狀態

當主機降級時，就會進入 permanent-failure 狀態。您無法在狀態為 permanent-failure 的專用主機啟動執行個體。

在配置替代主機後，它會維持在 pending 狀態，直到支援即時移轉主機維護的執行個體已自動從已降級的主機移轉，以及直到已為其餘執行個體安排排程事件。完成這些任務後，替換主機會進入 available 狀態。

當替代主機進入 available 狀態後，您可以像使用帳戶中的任何主機一樣加以使用。但是，替換主機上的某些執行個體容量會保留給需要重新開機型主機遷移的執行個體。您無法在此預留容量中啟動新執行個體。

當已降級的主機沒有執行個體時，它就會進入 released, permanent-failure 狀態，並從您的帳戶中永久釋出。請注意，主機及其資源會在主控台中短暫保持可見。

自動移轉

部分執行個體無法自動遷移至替代主機。

具有 EBS 支援的根磁碟區的執行個體

對於這些執行個體，執行個體停止事件排程為從通知日期起 28 天。在排程事件的日期與時間，執行個體會停止。我們建議您在替代主機或其他主機上重新啟動執行個體時手動停止。您可能需要修改執行個體的主機親和性，才能在不同主機上重新啟動。

具有執行個體儲存體根磁碟區的執行個體

對於這些執行個體，執行個體淘汰事件排程為從通知日期起 28 天。在排程事件的日期與時間，執行個體將永久終止。我們建議您在替換主機上手動啟動替換執行個體，然後在排程事件之前將所需資料遷移到替換執行個體。

下列執行個體具有執行個體儲存體根磁碟區：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

您可在排程事件前繼續存取已降級專用主機的執行個體。

配置 Amazon EC2 專用主機的主機維護設定

啟用主機維護以確保在專用主機執行的執行個體可在排定維護事件期間自動復原到新專用主機上。

如果停用主機維護，則會收到電子郵件通知，以移出降級主機，並在 28 天內將執行個體手動遷移到其他主機。如果您有專用執行個體預留，則會配置替代主機。28 天後，在降級主機上執行的執行個體將終止，並自動釋放主機。

Console

啟用專用主機的主機維護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用主機。
3. 詳情「專用執行個體」>「動作」>「修改主機」。
4. 在主機維護欄位中選取開啟。

停用專用主機的主機維護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用主機。
3. 詳情「專用執行個體」>「動作」>「修改主機」。
4. 在主機維護欄位中選取關閉。

AWS CLI

啟用專用主機的主機維護

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-maintenance on \
    --host-ids h-0d123456bbf78910d
```

停用專用主機的主機維護

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-maintenance off \
    --host-ids h-0d123456bbf78910d
```

PowerShell

啟用專用主機的主機維護

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -HostMaintenance on `
    -HostId h-0d123456bbf78910d
```

停用專用主機的主機維護

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -HostMaintenance off `
    -HostId h-0d123456bbf78910d
```

監控您的 Amazon EC2 專用主機的狀態

Amazon EC2 會持續監控您的 專用執行個體 的狀態。更新會在 Amazon EC2 主控台上傳達。您可以使用下列方法，檢視專用執行個體的相關資訊。

Console

檢視專用執行個體的狀態

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 專用執行個體。
3. 在清單中找出專用執行個體，並檢閱 State (狀態) 欄的值。

AWS CLI

檢視專用執行個體的狀態

使用 describe-hosts 命令。

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

檢視專用執行個體的狀態

使用 Get-EC2Host cmdlet。

```
Get-EC2Host -HostId h-012a3456b7890cdef
```

下表說明可能的專用執行個體狀態。

州	Description
available	AWS 尚未偵測到專用主機的問題。沒有排定的維護或修復。執行個體可在此專用執行個體上啟動。
released	專用執行個體已釋出。主機 ID 已不再使用。已釋出的主機即無法重複使用。
under-assessment	AWS 正在探索專用主機的可能問題。如果必須採取動作，系統會透過 AWS 管理主控台 或 電子郵件通知您。執行個體無法在此狀態的專用執行個體上啟動。
pending	專用主機無法用於新的執行個體啟動。已經修改為支援多個執行個體類型，或是主機復原正在進行中。
permanent-failure	已偵測到無法復原的失敗。您會透過執行個體和電子郵件收到移出通知。執行個體可繼續運作。如果您停止或終止具有此狀態的專用主機上的所有執行個體， 會 AWS 淘汰主機。 AWS 不會在此狀態下重新啟動執行個體。執行個體無法在此狀態下於專用執行個體啟動。
released-permanent-failure	AWS 永久發行失敗且不再在其上執行執行個體的專用主機。專用執行個體 ID 將無法再使用。

使用 追蹤 Amazon EC2 專用主機組態變更 AWS Config

您可以使用 AWS Config 記錄專用主機的組態變更，以及啟動、停止或終止的執行個體。接著，您就可以將 AWS Config 擷取的資訊做為授權回報的資料來源。

AWS Config 會個別記錄專用主機和執行個體的組態資訊，並透過關係配對此資訊。有下列三種回報條件：

- AWS Config 記錄狀態 - 開啟時， AWS Config 正在記錄一或多個 AWS 資源類型，其中可包含專用主機和專用執行個體。為了擷取授權回報所需的資訊，請確認主機和執行個體的下列欄位都有記錄到。
- 主機記錄狀態— 為 Enabled (啟用) 時，會記錄專用執行個體的組態資訊。
- 執行個體記錄狀態 – 為 Enabled (啟用) 時，會記錄專用預留執行個體的組態資訊。

如果上述三個條件中有任一個停用，Edit Config Recording (編輯組態記錄) 按鈕的圖示就會是紅色。若要獲得這項工具的完整效益，請確認所有三個記錄方法均為啟用。所有三個項目均啟用時，圖示即為綠色。若要編輯設定，請選擇 Edit Config Recording (編輯組態記錄)。系統會將您導向 AWS Config 主控台中的設定 AWS Config頁面，您可以在其中設定 AWS Config 和開始記錄主機、執行個體和其他支援的資源類型。如需詳細資訊，請參閱《 AWS Config 開發人員指南》中的AWS Config 使用主控台設定。

> **Note**
>
> AWS Config 會在發現您的資源後記錄這些資源，這可能需要幾分鐘的時間。

AWS Config 開始將組態變更記錄到主機和執行個體之後，您可以取得已配置或發行之任何主機的組態歷史記錄，以及已啟動、停止或終止的任何執行個體。例如，您可以在任何時間點於專用執行個體的組態歷史記錄中，查詢該主機上啟動了多少執行個體，以及該主機上的通訊端與核心數目。對於任何這些執行個體，您也可以查詢其 Amazon Machine Image (AMI) 的 ID。您可以使用這些資訊來回報您自有伺服器固定的軟體授權 (依據每個通訊端或每個核心)。

您可以透過以下任何方式來檢視組態歷史記錄：

- 使用 AWS Config 主控台。對於每個記錄的資源，您可以查看時間軸頁面，其中提供組態詳細資訊的歷史記錄。若要查看此頁面，請選擇專用執行個體頁面中的 Config Timeline (組態時間軸) 欄內的灰色圖示。如需詳細資訊，請參閱《 AWS Config 開發人員指南》中的在 AWS Config 主控台中檢視組態詳細資訊。
- 透過執行 AWS CLI 命令。首先，您可以使用 list-discovered-resources 命令，取得所有主機和執行個體的清單。然後，您可以使用 get-resource-config-history 命令，取得在特定時間間隔內之主機和執行個體的組態詳細資訊。
- 在應用程式中使用 AWS Config API。首先，您可以使用 ListDiscoveredResources 動作，取得所有主機和執行個體的清單。然後，您可以使用 GetResourceConfigHistory 動作，取得在特定時間間隔內之主機和執行個體的組態詳細資訊。

例如，若要從中取得所有專用主機的清單 AWS Config，請執行 CLI 命令，如下所示。

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

若要從 取得專用主機的組態歷史記錄 AWS Config，請執行 CLI 命令，如下所示。

```
aws configservice get-resource-config-history \
    --resource-type AWS::EC2::Instance \
    --resource-id i-1234567890abcdef0
```

使用主控台管理 AWS Config 設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在 專用執行個體 頁面上，選擇 Edit Config Recording (編輯組態記錄)。
3. 在 AWS Config 主控台中，遵循提供的步驟來開啟錄製。如需詳細資訊，請參閱AWS Config 使用主控台設定 。

如需詳細資訊，請參閱 AWS Config 主控台中的檢視組態詳細資訊。

AWS Config 使用命令列或 API 啟用

- AWS CLI：《 開發人員指南》中的檢視組態詳細資訊 (AWS CLI)。 AWS Config
- Amazon EC2 API：GetResourceConfigHistory。

Amazon EC2 專用執行個體

依預設，EC2 執行個體在共用的租用硬體上執行。這表示多個 AWS 帳戶可能會共用相同的實體硬體。

專用執行個體是在單一 AWS 帳戶專用硬體上執行的 EC2 執行個體。這表示專用執行個體在主機硬體層級實際與屬於其他的執行個體隔離 AWS 帳戶，即使這些帳戶連結至單一付款人帳戶。不過，專用執行個體可能會與來自非專用執行個體 AWS 帳戶 之相同 的其他執行個體共用硬體。

專用執行個體不會提供執行個體置放的可見性或控制權，且不支援主機親和性。如果您停止並啟動專用執行個體，其可能不會在相同的主機上執行。同樣地，您無法以要啟動或執行執行個體的特定主機為目標。此外，專用執行個體提供自帶授權 (BYOL) 的有限支援。

如果您需要對執行個體置放的可見性或控制權，以及更全面的 BYOL 支援，請考慮改用專用主機。專用執行個體及專用主機都可用於在專用實體伺服器啟動 Amazon EC2 執行個體。專用執行個體和專用

執行個體上的執行個體之間沒有效能、安全或實體差異。不過，兩者之間還是有一些差異。下表特別列出了專用執行個體與專用執行個體之間的主要差異：

	Dedicated Host	Dedicated Instance
專用實體伺服器	具備執行個體容量的實體伺服器，完全專供您使用。	專用於單一客戶帳戶的實體伺服器。
執行個體容量共用	可與其他帳戶共用執行個體容量。	不支援
帳單	按每台主機計費	按每個執行個體計費
通訊端、核心和主機 ID 的可見度	提供通訊端與實體核心數目的可見度	沒有可見度
主機和執行個體親和性	可讓您隨時間一致地將執行個體部署到相同的實體伺服器	不支援
訂定執行個體配置目標	對於執行個體在實體伺服器上的配置方式，提供額外可見性和控制權	不支援
自動執行個體復原功能	支援。如需詳細資訊，請參閱 Amazon EC2 專用主機復原。	支援
使用自有授權 (BYOL)	支援	部分支援 *
Capacity Reservations	不支援	支援

* 透過軟體保證而實現的授權行動性的 Microsoft SQL Server 和 Windows 虛擬桌面存取 (VDA) 授權可以與專用預留執行個體一起使用。

如需詳細資訊，請參閱Amazon EC2 專用執行個體。

目錄

- 專用執行個體 基本概念

- 支援的功能
- 專用執行個體 限制
- 專用執行個體定價
- 在使用預設租用的 VPC 中啟動專用執行個體
- 變更 EC2 執行個體租用
- 變更 VPC 的執行個體租用

專用執行個體 基本概念

VPC 的租用可以是 `default` 或 `dedicated`。依預設，您的 VPC 具有 `default` 租用，並且在 `default` 租用 VPC 中啟動的執行個體具有 `default` 租用。可以執行下列作業來啟動專用執行個體：

- 建立具有 `dedicated` 租用的 VPC，以便 VPC 中的所有執行個體作為專用執行個體執行。如需詳細資訊，請參閱在使用預設租用的 VPC 中啟動專用執行個體。
- 建立具有 `default` 租用的 VPC，並為執行個體手動指定 `dedicated` 的租用以作為專用執行個體執行。如需詳細資訊，請參閱在使用預設租用的 VPC 中啟動專用執行個體。

支援的功能

專用執行個體支援下列功能 AWS 和服務整合：

功能

- 預留執行個體
- 自動擴展規模
- 自動復原
- 專用的 競價型執行個體
- 爆量效能執行個體

預留執行個體

若要為您的專用執行個體預留容量，您可以購買專用預留執行個體或容量保留。如需詳細資訊，請參閱Amazon EC2 的預留執行個體概觀及使用 EC2 隨需容量保留來保留運算容量。

當您購買專用的預留執行個體時，您是以更為低廉的費用購買啟動專用執行個體的容量；唯有以專用租用啟動執行個體時才適用此使用費價格優惠。當您購買搭配預設租用的預留執行個體時，它僅適用於搭配 `default` 租用的運作中執行個體，不適用於搭配 `dedicated` 租用的運作中執行個體。

購買後，您就無法使用修改程序變更預留執行個體的租用。但您可以交換可轉換預留執行個體和具有不同租用的新可轉換預留執行個體。

自動擴展規模

您可以使用 Amazon EC2 Auto Scaling 來啟動專用執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用進階設定建立啟動範本。

自動復原

如果專用執行個體由於基礎硬體故障或需要 AWS 參與修復的問題而受損，您可以為專用執行個體設定自動復原。如需詳細資訊，請參閱自動執行個體復原功能。

專用的 競價型執行個體

您可以在建立 Spot 執行個體請求時，藉由指定 `dedicated` 的租用來執行專用 Spot 執行個體。如需詳細資訊，請參閱 在單一租戶硬體上啟動。

爆量效能執行個體

您可以將在專用租用硬體上執行的好處運用於 the section called “爆量效能執行個體”。T3 專用執行個體預設會在無限模式下啟動，提供基準水準的 CPU 效能，並可在您的工作負載要求較高 CPU 水準時大幅提升 CPU 效能。T3 基準效能與大幅提升效能的能力，取決於 CPU 的額度。由於 T3 執行個體類型的高載本質，我們建議您監視 T3 執行個體如何使用專用硬體的 CPU 資源以獲得最佳效能。T3 專用執行個體適用於具有各種工作負載且顯示隨機 CPU 行為的客戶，但理想情況下，CPU 平均用量達到或低於基準用量。如需詳細資訊，請參閱the section called “重要概念”。

Amazon EC2 備有系統來識別和更正效能變化。不過，如果您啟動多個具有相關 CPU 使用模式的 T3 專用執行個體，仍有可能遇到短期變化。對於這些更高要求或相關的工作負載，我們建議使用 M5 或 M5a 專用執行個體，而不是 T3 專用執行個體。

專用執行個體 限制

使用專用執行個體時，請謹記以下事項：

- 執行個體租用設定為 的 VPC 不支援某些 AWS 服務或其功能`dedicated`。請參閱各自服務的文件以確認是否有任何限制。

- 有些執行個體類型無法在執行個體租用設為 dedicated 的 VPC 中啟用。如需受支援之執行個體類型的詳細資訊，請參閱 Amazon EC2 專用執行個體。
- 當您啟動 Amazon EBS 支援的專用執行個體時，EBS 磁碟區不會在單一租用戶硬體上執行。

專用執行個體定價

專用執行個體定價與隨需執行個體定價不同。如需詳細資訊，請參閱 Amazon EC2 專用執行個體。

在使用預設租用的 VPC 中啟動專用執行個體

當您建立 VPC 時，您可選擇指定其執行個體租用。如果您在具有執行個體 dedicated 租用的 VPC 中啟動執行個體，則它會以硬體上之專用執行個體 (專供您使用) 的形式執行。

如需啟動具有 host 租用之執行個體的詳細資訊，請參閱在 Amazon EC2 專用主機上啟動 Amazon EC2 執行個體。

如需有關 VPC 租用選項的詳細資訊，請參閱「Amazon VPC 使用者指南」中的建立 VPC。

要求

- 選擇支援的執行個體類型。如需詳細資訊，請參閱 Amazon EC2 專用執行個體。

Console

在預設租用 VPC 中啟動專用執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)、Launch Instance (啟動執行個體)。
3. 在 Application and OS Images (應用程式和 OS 映像) 區段中，從清單中選取 AMI。
4. 在 Instance type (執行個體類型) 區段中，選取要啟動的執行個體類型。
5. 在 Key pair (金鑰對) 區段中，選取要與執行個體建立關聯的金鑰對。
6. 在 Advanced details (進階詳細資訊) 區段，對於 Tenancy (租用)，選取 Dedicated (專用)。
7. 視需要設定其餘的執行個體選項。如需詳細資訊，請參閱Amazon EC2 執行個體組態參數的參考。
8. 選擇啟動執行個體。

AWS CLI

在啟動期間設定執行個體的租用選項

使用 run-instances 命令，並包含 Tenancy 與 --placement 選項。

```
--placement Tenancy=dedicated
```

PowerShell

在啟動期間設定執行個體的租用選項

使用 New-EC2Instance cmdlet 搭配 -Placement_Tenancy 參數。

```
-Placement_Tenancy dedicated
```

變更 EC2 執行個體租用

可在啟動後變更已停止的執行個體的租用。您所做的變更會在執行個體下次開始時生效。

或者，您可以變更虛擬私有雲端 (VPC) 的租用。如需詳細資訊，請參閱the section called “變更 VPC 租用”。

限制

- 您無法使用 AWS 管理主控台變更執行個體的租用。
- 執行個體必須處於 stopped 狀態。
- 執行個體的作業系統詳細資訊 (以及是否已安裝 SQL Server) 會影響支援的轉換。如需執行個體可用租用轉換路徑的詳細資訊，請參閱「License Manager 使用指南」中的租用轉換。
- 若使用 T3 執行個體，您必須在專用執行個體上啟動執行個體，才能使用 host 的租用。您無法將租用從 host 變更為 dedicated 或 default。嘗試進行上述任一不受支援的租用變更時，會顯示 InvalidRequest 錯誤代碼。

AWS CLI

修改執行個體的租用值

使用 modify-instance-placement 命令。

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
    --tenancy dedicated
```

PowerShell

修改執行個體的租用值

使用 Edit-EC2InstancePlacement cmdlet。

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Tenancy Dedicated
```

變更 VPC 的執行個體租用

您在建立虛擬私有雲端 (VPC) 的執行個體租用之後，可將其從 dedicated 變更為 default。修改 VPC 的執行個體租用不影響 VPC 中任何現有執行個體的租用。下次在 VPC 中啟動執行個體時，除非您在執行個體啟動期間另行指定，否則其會有 default 租用。

或者，您可變更特定執行個體的租用。如需詳細資訊，請參閱the section called “變更執行個體租用”。

限制

- 您無法在建立之後，將 VPC 的執行個體租用從 default 變更為 dedicated。
- 您無法使用 AWS 管理主控台修改 VPC 的執行個體租用。

AWS CLI

修改 VPC 的執行個體租用屬性

使用 modify-vpc-tenancy 命令。唯一支援的租用值為 default。

```
aws ec2 modify-vpc-tenancy \
    --vpc-id vpc-1234567890abcdef0 \
    --instance-tenancy default
```

PowerShell

修改 VPC 的執行個體租用屬性

使用 Edit-EC2VpcTenancy cmdlet。唯一支援的租用值為 `Default`。

```
Edit-EC2VpcTenancy `
    -VpcId vpc-1234567890abcdef0 `
    -InstanceTenancy Default
```

隨需容量保留和 ML 容量區塊

容量保留可讓您在特定的可用區域中，為 Amazon EC2 執行個體保留運算容量。容量保留有兩種類型，用於不同的使用案例。

容量保留類型

- 隨需容量保留
- ML 的容量區塊

若要將容量保留組織為邏輯集合，請參閱容量保留資源群組。

以下是隨需容量保留的一些常見使用案例：

- 擴展事件 – 在關鍵業務事件之前建立隨需容量保留，以確保您可以在需要時進行擴展。
- 法規要求和災難復原 – 使用隨需容量保留來滿足法規要求，以達到高可用性，並在不同可用區域或區域中保留容量以進行災難復原。
- 共用未使用的容量 – 使用可中斷的容量預留，讓未使用的容量可供您帳戶中的其他工作負載使用，同時保持控制以在需要時回收。

以下是 ML 容量區塊的一些常見使用案例：

- 機器學習 (ML) 模型訓練和微調 – 不間斷存取保留的 GPU 執行個體，以完成 ML 模型訓練和微調。
- ML 實驗和原型 – 進行實驗並建置短時間需要 GPU 執行個體的原型。

使用隨需容量保留的時機

如果您有嚴格的容量要求，且目前或未來的關鍵業務工作負載需要容量保證，請使用隨需容量保留。使用隨需容量保留，您可以確保只要需要時，即可存取 Amazon EC2 容量。

使用 ML 容量區塊的時機

如果您需要確保可在未來的設定時段內不間斷存取 GPU 執行個體，請使用 ML 容量區塊。容量區塊適合訓練和微調 ML 模型、短期實驗執行，以及處理未來暫時提高的推論需求。使用容量區塊，您可以確保在特定日期可存取 GPU 資源以執行 ML 工作負載。

使用 EC2 隨需容量保留來保留運算容量

Amazon EC2 容量保留可讓您在特定可用區域中，為 Amazon EC2 執行個體保留任何期限的運算容量。如果您對目前或未來的業務關鍵工作負載有嚴格的容量需求，需要特定層級的長期或短期容量保證，建議您建立容量保留，以協助確保只要在需要時，即可有 Amazon EC2 容量存取權。

您可以隨時建立容量保留，並選擇開始時間。您可以請求立即使用容量保留，也可以請求未來日期的容量保留。

- 如果您請求立即使用容量保留，容量保留將立即可供使用，且無需任何期限承諾。您可以隨時修改容量保留，也可以隨時取消，以釋出保留容量並停止產生費用。
- 如果您請求未來日期的容量保留，您可以指定何時需要容量，以及承諾保留多久。容量保留會在指定的未來日期可供使用，並開始計費。在承諾期間，您無法將執行個體計數或承諾期間減少到初始承諾以下。您可以取消容量保留，但視取消時間而定，可能會收取取消費用。在承諾期間過後，您可以以任何方式修改容量保留或取消容量保留，無需支付費用。

容量預留 只能用於符合屬性的執行個體。根據預設，容量保留會自動比對新的執行個體和具有相符屬性 (執行個體類型、平台、可用區域與租用) 之執行中執行個體。這代表任何屬性相符的執行個體會在 容量預留 中自動執行。不過，您也可以針對特定工作負載的容量預留。這可讓您明確控制哪些執行個體可以在該預留容量中執行。您也可以指定執行個體只會在容量保留或容量保留資源群組中執行。

> **Important**
>
> 未來日期的容量保留旨在幫助您啟動和涵蓋增量執行個體，而不是涵蓋現有執行中執行個體。如果您需要涵蓋現有執行中執行個體，請改用立即啟動的容量保留。

擁有相符屬性的所有受支援 Amazon EC2 執行個體皆符合在容量保留中執行的資格，即執行個體類型、平台、可用區域，以及租用。Amazon EC2 執行個體可以由您 (非受管執行個體) 啟動，也可以由 AWS 服務 (受管執行個體) 代表您啟動。開放式容量保留尤其如此，可自動比對任何擁有相符屬性的執行中執行個體。舉例來說，以下服務代您啟動的受管執行個體符合在您建立及管理的容量保留中執行的資格。

- Amazon EC2 Auto Scaling

- Amazon ECS
- Amazon EKS
- Amazon EMR
- AWS Batch
- AWS Elastic Beanstalk
- AWS ParallelCluster
- AWS Parallel Computing 服務 (AWS PCS)

目錄

- Amazon EC2 容量保留概念
- 容量預留、預留執行個體和 Savings Plans 之間的差異
- 支援的平台
- 配額
- 限制
- 容量預留定價和帳單
- 建立 容量預留
- 檢視容量保留狀態
- 使用現有的 容量預留 啟動執行個體
- 修改作用中容量保留
- 修改執行個體容量保留設定
- 在容量保留之間移動容量
- 從現有容量保留分割容量
- 取消 容量預留
- 將容量保留與置放群組搭配使用
- Local Zones 中的容量保留
- 在 Wavelength 區域中的容量保留
- 上的容量保留 AWS Outposts
- 共用容量保留
- 容量預留機群
- 使用 CloudWatch 指標監控容量保留用量
- 監控容量保留使用率不足

- 監控未來日期容量保留的狀態變更
- 可中斷的容量保留

Amazon EC2 容量保留概念

以下重要概念適用於容量保留。

主題

- 開始日期和時間
- 結束日期和時間
- 承諾期限
- 未來日期的容量保留評估
- 容量保留屬性
- 執行個體符合條件

開始日期和時間

開始日期和時間定義容量保留何時可供使用。容量保留可以立即開始，也可以在未來日期開始。

- 如果您選擇立即啟動容量保留，預留容量將在您建立後立即可用，且在容量保留進入作用中狀態後會立即開始計費。您無需簽署任何期限承諾。您可以隨時根據需要修改容量保留，以滿足您的需求，也可以隨時取消保留，以釋出容量並停止產生費用。
- 如果您選擇在未來日期開始容量保留，您可以指定何時需要容量，以及承諾保留多久。在指定的未來日期，容量保留可供使用，並在容量保留進入作用中狀態後開始計費。一旦您的帳戶佈建容量保留，承諾持續時間就會開始。在此期間，您無法將執行個體計數減少到低於遞交的執行個體計數，或選擇遞交持續時間之前的結束日期。您可以取消容量保留，但視取消時間而定，可能會收取取消費用。如需詳細資訊，請參閱取消 容量預留。在承諾期間過後，您可以以任何方式修改容量保留，或取消它以釋出預留容量並停止產生費用。

結束日期和時間

結束日期和時間定義了容量保留何時結束，以及預留容量何時從您的帳戶中釋出。您可以將容量保留設定為在特定日期和時間自動結束，也可以將其設定為無限期保持作用中狀態，直到手動取消為止。

如果您將容量保留設定為自動結束，容量保留會在指定時間的一小時內到期。例如，如果您指定 `5/31/2019, 13:30:55`，容量保留保證會在 5/31/2019 的 13:30:55 至 14:30:55 結束。

保留結束後，預留容量將從您的帳戶中釋出，您將無法再指派執行個體至該容量保留。使用預留容量執行的執行個體會持續不間斷執行。如果針對容量保留的執行個體停止，在移除其容量保留針對的偏好設定或設定它們針對不同的容量保留之前，都無法重新啟動它們。如需詳細資訊，請參閱修改執行個體容量保留設定。

承諾期限

承諾期限僅適用於未來日期的容量保留。

承諾期限是您承諾在佈建未來日期的容量保留後，將其以作用中狀態保持在您帳戶的最短持續時間。對於未來日期的容量保留，您可將其保留至超過承諾期限，但不能短於該期限。以下情況適用於承諾期限：

- 您可以在承諾期間取消容量保留，但可能需要支付取消費用。如需詳細資訊，請參閱取消費用。
- 您無法將執行個體數減少到低於承諾的執行個體數，但您可增加數量。
- 您無法將容量保留設定為在承諾期限內的日期或時間自動結束。您可以在承諾期間延長結束日期和時間。

Amazon EC2 會使用您指定的承諾期限來評估是否可支援請求。大多數執行個體類型的最低承諾持續時間為 14 天，U 和 X 執行個體則為 12 週。在評估請求時，Amazon EC2 可能會確定其可支援較短的承諾期限。在這種情況下，Amazon EC2 會排程承諾持續時間較短的未來日期容量保留。這代表您承諾在帳戶中保持容量保留的時間會短於您最初請求的時間。

未來日期的容量保留評估

當您請求未來日期的容量保留時，Amazon EC2 會根據容量可用性和您指定的承諾期限來評估請求，以確定可否支援該請求。評估通常會在 5 天內完成。Amazon EC2 在評估請求時會考慮多個因素，包括：

- 預測容量供應
- 承諾期限
- 您請求容量保留的時間距離開始日期有多早
- 您請求的大小

您可提前 5 至 120 天請求未來日期的容量保留。建議您至少提前 56 天 (8 週) 提出請求，以改善我們支援請求的能力。大多數執行個體類型的最低承諾持續時間為 14 天，U 和 X 執行個體則為 12 週，而最低執行個體計數為 32 個 vCPUs。

在評估請求期間，容量保留仍處於 assessing 狀態。

如果可以支援請求，則容量保留會進入 scheduled 狀態，並安排在請求的日期和時間交付。當容量保留處於 scheduled 狀態時，執行個體總數會保持 0。已排定的容量保留將於請求的日期變為 active 且可供使用。

如果無法支援請求，則容量保留會進入 unsupported 狀態。不提供不受支援的的容量保留。

您可以在 assessing、 scheduled或 active 狀態取消未來日期的容量保留。可能需要支付取消費用。如需詳細資訊，請參閱取消 容量預留。

若要建立未來日期的容量保留，請參閱 建立未來日期的容量保留。

容量保留屬性

在建立容量保留時，您必須指定以下屬性：

- 可用區域
- 執行個體類型
- 平台 (作業系統類型)
- 租用 (default 或 dedicated)

只有符合這些屬性的執行個體才能在容量保留中啟動或執行。

執行個體符合條件

執行個體符合條件或執行個體資格決定了哪些執行個體可在容量保留中啟動和執行。容量保留可以具備下列其中一種符合條件：

- 開放式 - 容量保留會自動比對所有具有相符屬性 (執行個體類型、平台和可用區域) 的執行個體。具備相符屬性的現存與新執行個體會在容量保留中自動執行，而無須任何額外設定。
- 目標式 - 容量保留僅接受有相符屬性 (執行個體類型、平台與可用區域) 的執行個體，並明確將其容量保留設為目標。執行個體必須明確指派容量保留，以便在其預留容量中啟動或執行。這可讓您明確控制哪些執行個體可在預留容量執行，並幫助您避免意外使用預留容量。

當您請求未來日期的容量保留時，您只能指定目標符合條件。這可確保容量保留提供的容量對於您在交付時擁有的任何執行中執行個體或預留容量而言是增量的或附加的。在帳戶的容量保留啟用之後，您可根據需要將執行個體符合條件變更為開放式。但是，請記住，任何符合的執行個體都將自動在容量保留中執行，這可能會導致意外使用容量，並阻止您啟動新執行個體以獲得請求的完整執行個體數。

容量預留、預留執行個體和 Savings Plans 之間的差異

下表反白容量預留、預留執行個體和 Savings Plans 之間的主要差異：

	Capacity Reservations	區域預留執行個體	地區預留執行個體	Savings Plans
期間	立即使用的容量保留無需承諾。可視需要加以建立、修改及取消。 對於未來日期的容量保留，您可以指定承諾期限，並承諾在該期限內將容量保留在您的帳戶中。您可以隨時取消；在承諾期間可能會收取取消費用。	需要綁約一年或三年		
容量優勢	在特定的可用區域中預留的容量。		無預留容量。	
帳單折扣	無帳單折扣。†	提供計費折扣。		
執行個體限制	適用每個區域的 隨需執行個體 限制。	每個可用區域預設為 20。您可以請求提高限制。	每個區域預設為 20。您可以請求提高限制。	沒有限制。

† 您可以將容量預留與 Savings Plans 或區域預留執行個體結合，以獲得折扣。

如需詳細資訊，請參閱下列內容：

- Amazon EC2 的預留執行個體概觀
- Savings Plans 使用者指南

支援的平台

您必須使用正確的平台來建立容量預留，以確保它與您的執行個體正確相符。容量保留支援 `platform` 的下列值：

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux
- RHEL with SQL Server Standard
- RHEL with SQL Server Enterprise
- RHEL with SQL Server Web
- RHEL with HA
- RHEL with HA and SQL Server Standard
- RHEL with HA and SQL Server Enterprise
- Ubuntu Pro
- Windows
- 具有 SQL Server 的 Windows
- Windows with SQL Server Web
- Windows with SQL Server Standard
- Windows with SQL Server Enterprise

如需確保執行個體在特定容量保留執行，容量保留的平台必須與用於啟動執行個體的 AMI 平台相符。若是 Linux AMI，檢查 AMI 平台是使用一般值 Linux/UNIX，還是 SUSE Linux 等具體的值非常重要。

Console

如需檢查 AMI 平台

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMI。

3. 請選取 AMI。
4. 在詳細資訊索引標籤上，查看平台詳細資訊的值。

AWS CLI

如需檢查 AMI 平台

使用 describe-images 命令，然後檢查 PlatformDetails 的值。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

以下為範例輸出。

```
[
    "Linux/UNIX"
]
```

PowerShell

如需檢查 AMI 平台

使用 Get-EC2Image cmdlet，然後檢查 PlatformDetails 的值。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).PlatformDetails
```

以下為範例輸出。

```
Linux/UNIX
```

配額

您可以預留容量的執行個體數目是以您帳戶的隨需執行個體配額為基礎。您可以預留容量的執行個體數目，為該配額所允許的數目減去已在執行的執行個體數目。

assessing、scheduled、pending、active 和 delayed 州的容量保留會計入您的隨需執行個體配額。

限制

在您建立容量預留之前，請記下以下的限制與法規。

- 作用中和未使用的 容量預留 都計入 隨需執行個體 限制。
- 容量保留不能從一個 AWS 帳戶轉移到另一個帳戶。不過，您可以與其他 AWS 帳戶共用容量保留。如需詳細資訊，請參閱共用容量保留。
- 區域 預留執行個體 帳單折扣不適用於 容量預留。
- 叢集置放群組中無法建立容量保留。不支援分散和分區置放群組。
- 容量預留 不能與 專用執行個體 搭配使用。容量預留不能與專用執行個體搭配使用。
- [Windows 執行個體] 容量保留不能與自帶授權 (BYOL) 搭配使用。
- [Red Hat 執行個體] 容量保留可與自帶授權 (BYOL) 搭配使用。
- 容量預留 不確保休眠執行個體可以在嘗試啟動後恢復。
- 您可以為執行個體計數請求未來日期的容量保留，且至少包含 32 vCPUs。例如，如果您請求`m5.xlarge`執行個體的未來日期容量保留，您必須至少請求 8 個執行個體 (8 * m5.xlarge = 32 個 vCPUs)。
- 您可以為下列系列的執行個體類型請求未來日期的容量保留：C、G、I、M、R、T、U 和 X。

容量預留定價和帳單

本節主題概述容量保留的定價和計費方式。

主題

- 定價
- 帳單
- 帳單折扣
- 成本和用量報告 2.0 中的取消費用
- 檢視您的帳單

定價

無論您是否以預留容量執行執行個體，容量保留都會按同等的隨需費率收費，包括專用執行個體的任何適用區域附加費用。如果您不使用保留，此項目在 Amazon EC2 帳單中會顯示為未使用的保留。當您

執行符合保留屬性的執行個體時，您只需要支付執行個體的費用，不必支付保留費用。沒有前期或額外費用。

例如，如果您建立了 20 個 m4.large Linux 執行個體的 容量預留，在同一可用區域中執行了 15 個 m4.large Linux 執行個體，則您需要支付 15 個作用中執行個體和保留之 5 個未使用執行個體的費用。

Savings Plans 和區域預留執行個體的帳單折扣適用於容量預留。如需詳細資訊，請參閱帳單折扣。

如需詳細資訊，請參閱 Amazon EC2 定價。

帳單

一旦帳戶中佈建了容量，容量保留就會立即開始計費，並在帳戶容量保留保持佈建狀態時持續計費。對於未來日期的容量保留，這表示只有在您的帳戶中佈建容量保留之後，才會開始計費。

容量保留會以每秒精細程度計費，最少 60 秒。這表示不按小時計費。例如，如果您帳戶的容量保留作用了 24 小時又 15 分鐘，您需要支付 24.25 小時的保留費用。

以下範例示範容量預留的計費方式。您建立了一個 m4.large Linux 執行個體的 容量預留，其隨需費率為小時 0.10 USD。在此範例中，該帳戶的容量保留佈建了 5 個小時。第一個小時未使用 容量預留，所以未使用的這個小時按 m4.large 執行個體類型的標準隨需費率計費。在第二至第五個小時，m4.large 執行個體佔用了 容量預留。在這段期間，容量預留 不產生任何費用，但帳戶需要支付 m4.large 執行個體佔用它的費用。第六個小時取消了 容量預留，m4.large 執行個體在預留容量外正常執行。這個小時按 m4.large 執行個體類型的隨需費率收費。

Hour	1	2	3	4	5	6	Total cost
Unused Capacity Reservation	$0.10	$0.00	$0.00	$0.00	$0.00	$0.00	**$0.10**
On-demand Instance Usage	$0.00	$0.10	$0.10	$0.10	$0.10	$0.10	**$0.50**
Hourly cost	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.60**

帳單折扣

Savings Plans 和區域預留執行個體的帳單折扣適用於容量預留。 AWS 會自動將這些折扣套用至具有相符屬性的容量預留。當執行個體使用 容量預留，折扣會套用至該執行個體。在涵蓋未使用的 容量預留 之前，折扣會優先套用至執行個體用量。

用於區域 預留執行個體 的帳單折扣不適用於 容量預留。

如需詳細資訊，請參閱下列內容：

- Amazon EC2 的預留執行個體概觀

- Savings Plans 使用者指南
- 帳單和購買選項

成本和用量報告 2.0 中的取消費用

如果您取消未來日期的容量保留並收取取消費用，則費用會顯示在您的 AWS 成本和用量報告 (CUR) 2.0 中。如需 CUR 2.0 中容量保留資料欄的詳細資訊，請參閱容量保留資料欄。下表顯示容量保留明細項目在取消前後的顯示方式。

作用中保留的 CUR 2.0 值

capacity_reservation_capacity_reservation_status	line_item_usage_type	它代表什麼
Reserved	保留	預留容量總計
Used	BoxUsage	啟動的執行個體
Unused	UnusedBox	閒置容量

取消後的 CUR 2.0 值

capacity_reservation_capacity_reservation_status	line_item_usage_type	它代表什麼
Cancelling	保留	已取消的容量
Cancelling	UnusedBox	取消費用

檢視您的帳單

您可以在 AWS 帳單與成本管理 主控台上檢視您帳戶的計費和費用。

- Dashboard (儀表板) 會顯示您帳戶的支出摘要。
- 在 Bills (帳單) 頁面上，於 Details (詳細資訊) 下方，展開 Elastic Compute Cloud 區段及區域，以取得容量預留的帳單資訊。

您可在線上檢視費用，或是下載 CSV 檔案。若要了解詳細資訊，請參閱容量保留明細項目。

建立 容量預留

您可隨時建立容量保留，以確保您在特定可用區域中擁有運算容量。容量保留可以立即開始，也可以在未來日期開始。容量只有在容量保留進入 `active` 狀態後才能使用。

> **Note**
>
> 如果您使用 open 執行個體符合條件來建立容量保留，且在容量保留啟用時具有符合屬性的執行個體，則這些執行個體會自動以預留容量執行。若要避免這種情況，請使用 `targeted` 執行個體符合條件。如需詳細資訊，請參閱執行個體符合條件。

如果符合以下條件之一，則您建立容量預留的請求可能會失敗：

- Amazon EC2 沒有足夠的容量可滿足請求。請稍後再試、嘗試其他可用區域，或是嘗試較小的請求。如果您的應用程式能因應各種執行個體類型和大小，請嘗試不同的執行個體屬性。
- 請求的數量超過所選執行個體系列的隨需執行個體限制。請提高執行個體系列的隨需執行個體限制，然後再試一次。如需詳細資訊，請參閱隨需執行個體配額。

主題

- 建立可供立即使用的容量保留
- 建立未來日期的容量保留

建立可供立即使用的容量保留

建立可供立即使用的容量保留

Console

如需建立容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 容量預留，接著選擇 Create 容量預留 (建立 容量預留)。
3. 在執行個體詳細資訊區段中配置以下設定：
   a. 執行個體類型 - 要預留容量的執行個體類型。
   b. 平台 - 執行個體的作業系統。如需詳細資訊，請參閱支援的平台。

c. 可用區域 - 要預留容量的可用區域。

d. 租用 - 用於預留容量的租用類型。選擇預設為共用硬體預留容量，或選擇專用為專屬於您帳戶的硬體預留容量。

e. (選用) 置放群組 ARN - 要在其中建立容量保留的叢集置放群組的 ARN。如需詳細資訊，請參閱將容量保留與置放群組搭配使用。

f. 執行個體總數 - 要預留容量的執行個體數。如果您指定的數量超過所選執行個體類型的剩餘隨需執行個體配額，請求會失敗。

4. 在 Reservation details (保留詳細資訊) 區段中設定以下設定：

a. 容量保留開始 - 選擇立即。

b. 容量保留結束 - 選擇下列其中一個選項：

- 手動 - 預留容量直到您明確取消為止。
- 指定時間 - 在指定的日期和時間自動取消容量保留。

c. 執行個體資格 - 選擇下列其中一個選項：

- 開放式 - (預設值) 容量保留會比對任何有相符屬性 (執行個體類型、平台、可用區域與租用) 的執行個體。如果啟動具有相符屬性的執行個體，它會自動置放到預留容量。
- 目標式 - 容量保留僅接受有相符屬性 (執行個體類型、平台、可用區域與租用) 的執行個體，並明確指派至該保留。

5. 選擇建立。

AWS CLI

如需建立容量保留

使用 create-capacity-reservation 命令。

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria open|targeted
```

PowerShell

如需建立容量保留

使用 Add-EC2CapacityReservation cmdlet。

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion open|targeted
```

建立未來日期的容量保留

如果您需要預留容量以供未來的日期和時間使用，請求未來日期的容量保留。

在您請求未來日期的容量保留之後，系統會進行評估，確定是否可以支援此請求。如需詳細資訊，請參閱未來日期的容量保留評估。

考量事項

- 您可以為下列系列的執行個體類型請求未來日期的容量保留：C、G、I、M、R、T、U 和 X。
- 您可以為執行個體計數請求未來日期的容量保留，且至少包含 32 vCPUs。例如，如果您為 `m5.xlarge` 執行個體請求未來日期的容量保留，必須請求至少 8 個執行個體 (8 * m5.xlarge = 32 個 vCPU)。
- 您可提前 5 至 120 天請求未來日期的容量保留。我們建議您至少提前 56 天 (8 週） 提出請求，以改善支援能力。
- C、G、I、M、R 和 T 執行個體的最短承諾持續時間為 14 天。對於 U 和 X 執行個體，最短承諾持續時間為 12 週。

Console

如需建立容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 容量預留，接著選擇 Create 容量預留 (建立 容量預留)。
3. 在執行個體詳細資訊區段中配置以下設定：
   a. 執行個體類型 - 要預留容量的執行個體類型。
   b. 平台 - 執行個體的作業系統。如需詳細資訊，請參閱支援的平台。
   c. 可用區域 - 要預留容量的可用區域。

d. 租用 - 用於預留容量的租用類型。選擇預設為共用硬體預留容量，或選擇專用為專屬於您帳戶的硬體預留容量。

e. 執行個體總數 - 要預留容量的執行個體數。如果您指定的數量超過所選執行個體類型的剩餘隨需執行個體配額，請求會失敗。

4. 在 Reservation details (保留詳細資訊) 區段中設定以下設定：

a. 容量保留開始 - 選擇在指定時間。

b. 開始日期 - 指定容量保留必須可供使用的日期與時間。如需詳細資訊，請參閱開始日期和時間。

c. 承諾期限 - 指定在交付容量保留後，您承諾保持該容量保留的最短持續時間。如需詳細資訊，請參閱承諾期限。

d. 容量保留結束 - 選擇下列其中一個選項：

- 當我取消時 - 預留容量直到您明確取消為止。
- 指定時間 - 在指定的日期和時間自動取消容量保留。

5. 選擇建立。

AWS CLI

如需建立容量保留

使用 create-capacity-reservation 命令。

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria targeted \
    --delivery-preference incremental \
    --commitment-duration commitment_in_seconds \
    --start-date YYYY-MMDDThh:mm:ss.sssZ
```

PowerShell

如需建立容量保留

使用 Add-EC2CapacityReservation cmdlet。

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion targeted `
    -DeliveryPreference incremental `
    -CommitmentDuration commitment_in_seconds `
    -StartDate  YYYY-MMDDThh:mm:ss.sssZ
```

檢視容量保留狀態

Amazon EC2 會持續監控您的容量保留狀態。

由於 Amazon EC2 API 遵守最終一致性模式，因此在建立容量保留後，可能需要最多 5 分鐘時間，容量保留狀態才會反映為 `active`。在此期間，容量保留可能仍保持在 `pending` 狀態。但是，容量保留可能已經可供使用，在此情況下，嘗試啟動執行個體為容量保留可能會成功。

Console

如需檢視容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選取容量保留。

AWS CLI

如需描述容量保留

使用 describe-capacity-reservations 命令。

例如，下面的命令描述所有 容量預留。

```
aws ec2 describe-capacity-reservations
```

以下為範例輸出。

```
{
    "CapacityReservations": [
```

```
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-16T09:03:18.000Z",
            "AvailableInstanceCount": 1,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 1,
            "State": "active",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "a1.medium",
            "PlacementGroupArn": "arn:aws:ec2:us-east-1:123456789012:placement-
group/MyPG"
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-07T11:34:19.000Z",
            "AvailableInstanceCount": 3,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 3,
            "State": "cancelled",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "m5.large"
        }
    ]
}
```

PowerShell

如需描述容量保留

請使用 Get-EC2CapacityReservation cmdlet。

```
Get-EC2CapacityReservation `
```

```
    -CapacityReservationId cr-1234abcd56EXAMPLE
```

容量保留狀態

容量預留擁有以下可能的狀態。

State	說明
active	容量可供使用。
expired	容量保留在保留請求指定的日期和時間自動過期。預留容量將不再提供使用。
cancelled	容量保留已取消。預留容量將不再提供使用。
pending	容量保留請求成功，但容量佈建仍待定。
failed	容量保留請求失敗。請求會因為請求參數無效、容量限制或執行個體限制而失敗。失敗的請求可供您檢視 60 分鐘。
scheduled	(僅限未來日期的容量保留) 未來日期的容量保留請求已獲批准，且計劃在請求的開始日期交付容量保留。
assessing	(僅限未來日期的容量保留) Amazon EC2 正在評估您對未來日期容量保留的請求。如需詳細資訊，請參閱未來日期的容量保留評估。
delayed	(僅限未來日期的容量保留) Amazon EC2 在佈建請求的未來日期容量保留時遇到延遲。Amazon EC2 無法在請求的開始日期與時間交付請求的容量。
unsupported	(僅限未來日期的容量保留) 由於容量限制，Amazon EC2 無法支援未來日期的容量保留請求。您可在 30 天內檢視不受支援的請求。容量保留將不會交付。
cancelling	(僅限未來日期的容量保留) 正在取消容量保留。容量已釋出，但費用會持續到減少的承諾期間結束為止。承諾結束時cancelled ，容量保留會轉換為 。如需詳細資訊，請參閱取消費用。

使用現有的 容量預留 啟動執行個體

您只能在容量保留中啟動以下情況的執行個體：

- 具有符合屬性 (執行個體類型、平台、可用區域和租用)
- 有足夠可用容量
- 處於 `active` 狀態

當您啟動執行個體時，您可以指定是否在任何 open 容量預留、特定 容量預留、或容量預留群組中啟動執行個體。

或者，您可以設定執行個體以避免在容量預留中執行，即使您的 open 容量預留 有相符屬性和可用容量。

使用容量預留啟動執行個體可依啟動的執行個體數量降低其可用容量。例如，如果啟動三個執行個體，則容量預留的可用容量會減少三個。

Console

如需使用現有的容量保留啟動執行個體

1. 請按照程序啟動執行個體，但在完成以下步驟以指定置放群組和容量保留的設定之前，請勿啟動執行個體。
2. 展開進階詳細資訊，並執行下列動作：
   a. 針對置放群組，選取要在其中啟動執行個體的叢集置放群組。
   b. 對於 Capacity Reservation (容量保留)，根據容量保留的組態選取下列其中一個選項：
      - 無 - 防止啟動執行個體至容量保留。執行個體以隨需容量執行。
      - 開放式 - 使用具有相符屬性以及針對所選執行個體數量具有足夠容量的容量保留來啟動執行個體。如果沒有容量足夠的相符容量預留，執行個體將使用隨需容量。
      - 指定容量保留 - 在選取的容量保留中啟動執行個體。如果選取的容量預留在您所選之執行個體數量沒有足夠的容量，執行個體啟動會失敗。
      - 指定容量保留資源群組 – 將執行個體啟動至具有相符屬性和所選容量保留群組中可用容量的任何容量保留。如果所選群組沒有具有相符屬性和可用容量的容量保留，則執行個體會啟動至隨需容量。
      - 僅指定容量保留 - 在容量保留中啟動執行個體。如果未指定容量保留 ID，執行個體會啟動至開放式容量保留。如果容量不可用，執行個體將無法啟動。

- 僅限指定容量保留資源群組 – 在容量保留資源群組中的容量保留中啟動執行個體。如果未指定容量保留資源群組 ARN，執行個體會在開啟的容量保留中啟動。如果容量不可用，執行個體將無法啟動。

3. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。

AWS CLI

如需使用現有的容量保留啟動執行個體

請使用 run-instances 命令，並指定 `--capacity-reservation-specification` 選項。

以下範例會使用具有相符屬性和可用容量的任何開放式容量保留，來啟動執行個體：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification CapacityReservationPreference=open
```

以下範例會使用 `targeted` 容量保留來啟動執行個體：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

以下範例會使用指定的容量保留群組來啟動執行個體：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
```

```
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

以下範例僅使用容量保留來啟動執行個體。由於未指定容量保留 ID，執行個體將在具有相符屬性和可用容量的任何開放式容量保留中啟動：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only
```

以下範例僅使用指定的容量保留來啟動執行個體。如果在指定的容量保留中無法使用容量，執行個體將無法啟動。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

PowerShell

如需使用現有的容量保留啟動執行個體

使用 New-EC2Instance cmdlet。

以下範例會使用具有相符屬性和可用容量的任何開放式容量保留，來啟動執行個體：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
```

```
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "open"
```

以下範例會使用 targeted 容量保留來啟動執行個體：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

以下範例會使用指定的容量保留群組來啟動執行個體：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

以下範例僅使用容量保留來啟動執行個體。由於未指定容量保留 ID，執行個體將在具有相符屬性和可用容量的任何開放式容量保留中啟動：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only"
```

以下範例僅使用指定的容量保留來啟動執行個體。如果在指定的容量保留中無法使用容量，執行個體將無法啟動。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
```

```
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

修改作用中容量保留

如果您的現有容量保留不適合需要容量的工作負載，您可修改執行個體數量、執行個體資格 (open 或 targeted) 和結束時間 (At specific time 或 Manually)。如果您指定的新執行個體數量超過所選執行個體類型的剩餘隨需執行個體限制，更新會失敗。

允許的修改取決於容量保留狀態：

- assessing 或 scheduled 狀態 - 您只能修改標籤。
- pending 狀態 - 您無法以任何方式修改容量保留。
- active 狀態，但仍在承諾期限內 - 您無法將執行個體數減少至低於承諾的執行個體數，或將結束日期設定為承諾期限之前。允許所有其他修改。
- active 狀態，無承諾期限或承諾期限已過 - 允許所有修改。
- expired、cancelled、unsupported 或 failed 狀態 - 您無法以任何方式修改容量保留。

考量事項

- 在建立後，無法變更執行個體類型、平台、可用區域或租用。如果您需要修改這些屬性的任何一項，建議您取消保留，然後建立具有所需屬性的新保留。
- 如果您將執行個體資格從 targeted 變更為 open 並修改現有容量保留，則任何與容量保留的屬性相符、將 CapacityReservationPreference 參數設定為 open 且尚未在容量保留中執行的執行個體均將自動使用修改後的容量保留。
- 若要變更執行個體資格，容量保留必須完全閒置 (零使用量)，這是因為執行個體在保留內執行時，Amazon EC2 無法修改執行個體資格。

Console

如需修改容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選取 容量預留 (容量預留)，選取要修改的容量預留，然後選取 Edit (編輯)。
3. 根據需要修改總容量、容量保留結束或執行個體資格選項，然後選擇儲存。

AWS CLI

如需修改容量保留

使用 modify-capacity-reservation 命令。下列範例會修改指定的容量保留，以預留八個執行個體的容量。

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 8
```

PowerShell

如需修改容量保留

使用 Edit-EC2CapacityReservation cmdlet。下列範例會修改指定的容量保留，以預留八個執行個體的容量。

```
Edit-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0 `
    -InstanceCount 8
```

修改執行個體容量保留設定

您可以隨時修改已停止執行個體的 容量預留 設定：

- 從具有相符屬性 (執行個體類型、平台、可用區域與租用) 和可用容量的任何容量保留中開始。
- 在特定容量預留中啟動執行個體。
- 在 Capacity Reservation (容量預留) 群組內具有相符屬性和可用容量的任何 Capacity Reservation (容量預留) 中啟動
- 防止在容量預留中開始執行個體。

Console

如需修改執行個體容量保留設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選取 Instances (執行個體)，並選取要修改的執行個體。停止執行個體 (若還未停止的話)。
3. 選擇動作、執行個體設定、修改容量保留設定。

4. 針對 容量預留，請選擇下列其中一種選項：

   - 開放式 - 使用具有相符屬性以及針對所選執行個體數量具有足夠容量的容量保留來啟動執行個體。如果沒有容量足夠的相符容量預留，執行個體將使用隨需容量。
   - 無 - 防止啟動執行個體至容量保留。執行個體以隨需容量執行。
   - 指定容量保留 - 在選取的容量保留中啟動執行個體。如果選取的容量預留在您所選之執行個體數量沒有足夠的容量，執行個體啟動會失敗。
   - 指定容量保留群組 - 在選取的容量保留群組內具有相符屬性和可用容量的任何容量保留中啟動執行個體。如果所選群組沒有具有相符屬性和可用容量的容量保留，則執行個體會啟動至隨需容量。
   - 僅指定容量保留 - 在容量保留中啟動執行個體。如果未指定容量保留 ID，執行個體會啟動至開放式容量保留。如果容量不可用，執行個體將無法啟動。
   - 僅限指定容量保留資源群組 – 在容量保留資源群組中的容量保留中啟動執行個體。如果未指定容量保留資源群組 ARN，執行個體會在開啟的容量保留中啟動。如果容量不可用，執行個體將無法啟動。

AWS CLI

如需修改執行個體容量保留設定

使用 modify-instance-capacity-reservation-attributes 命令。

下列範例會將容量保留喜好變更為 none。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=none
```

下列範例會以特定容量保留為目標。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
    CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

下列範例會將目標變更為特定容量保留群組。

```
aws ec2 modify-instance-capacity-reservation-attributes \
```

```
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

下列範例會將容量保留喜好變更為 capacity-reservation-only。由於未指定容量保留，執行個體將在具有相符屬性及可用容量的任何開放式容量保留中啟動：

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=capacity-
reservation-only
```

下列範例會變更容量保留喜好設定為 capacity-reservation-only，並且將目標變更為特定容量保留。如果在指定的容量保留中無法使用容量，執行個體將無法啟動。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservation-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

PowerShell

如需修改執行個體容量保留設定

使用 Edit-EC2InstanceCapacityReservationAttribute cmdlet。

下列範例會將容量保留喜好變更為 none。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "none"
```

下列範例會以特定容量保留為目標。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

下列範例會將目標變更為特定容量保留群組。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

下列範例會將容量保留喜好變更為 capacity-reservation-only。由於未指定容量保留，執行個體將在具有相符屬性及可用容量的任何開放式容量保留中啟動：

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only"
```

下列範例會變更容量保留喜好設定為 capacity-reservation-only，並且將目標變更為特定容量保留。如果在指定的容量保留中無法使用容量，執行個體將無法啟動。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

在容量保留之間移動容量

您可將容量從一個容量保留移至另一個，以根據需要重新分配保留的運算資源。例如，如果您在用量不斷增加的保留中需要額外容量，且在另一個保留中擁有容量，則您可在這兩個保留之間重新分配容量。

移動容量的先決條件

作為先決條件，兩個容量保留必須滿足以下條件：

- 兩個保留都必須處於作用中狀態。
- 這兩個保留都必須由 擁有 AWS 帳戶。您無法在由不同 AWS 帳戶擁有的保留之間移動容量。
- 兩個保留必須相同：
  - 執行個體類型
  - 平台

- 可用區域
- 租用
- 置放群組
- 結束時間

目的地容量保留執行個體資格 (open 或 targeted) 和標籤無需符合來源保留。除了來源保留的容量減少和目的地保留的容量增加外，兩個保留的配置保持不變。

當您指定要移動的執行個體數量時，根據預設，會先移動任何可用容量，然後再移動任何符合資格的執行中執行個體 (保留中的已用容量)。例如，如果您從具有 5 個已用執行個體和 3 個可用執行個體的保留移動 4 個執行個體，則將移動 3 個可用執行個體和 1 個已用執行個體。

> **Note**
>
> 當您從保留中移動已用容量，且您所指定的要移動的數量大於可用容量時，僅會移動以容量保留規格為 open 來啟動的執行個體。

考量事項

將容量從一個保留移至另一保留時，應考量以下事項：

- 已用容量只能在具有 open 執行個體資格且共用相同帳戶組的容量保留之間移動。
- 當您移動已用容量時，會隨機選取符合條件的執行個體。您無法指定移動哪些執行個體。若未找到足夠數量的合格執行個體以滿足移動數量，移動操作將失敗。
- 如果您從來源保留移動所有容量，容量保留將自動取消。
- 未來日期的容量保留 - 在承諾期內，您無法移動未來日期容量保留的容量。

> **Note**
>
> 不支援從容量區塊移動容量。

移動容量

您可將容量從來源容量保留移至目的地容量保留。

Console

如需移動容量

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇容量保留。
3. 選擇具有可移動容量的隨需容量保留 ID。
4. 在動作的管理容量中，選擇移動。
5. 在移動容量頁面的目的地容量保留中，從清單中選取保留。
6. 在要移動的數量中，使用滑桿或輸入要從來源容量保留移動到目的地容量保留的執行個體數量。
7. 檢閱摘要，當您準備好時，選擇移動。

AWS CLI

如需移動容量

請使用 move-capacity-reservation-instances 命令。下列範例將 10 個執行個體從指定來源容量保留移至特定目的地容量保留。

```
aws ec2 move-capacity-reservation-instances \
    --source-capacity-reservation-id cr-1234567890abcdef0 \
    --destination-capacity-reservation-id cr-021345abcdef56789 \
    --instance-count 10
```

PowerShell

如需移動容量

使用 Move-EC2CapacityReservationInstance cmdlet。下列範例將 10 個執行個體從指定來源容量保留移至特定目的地容量保留。

```
Move-EC2CapacityReservationInstance `
    -SourceCapacityReservationId cr-1234567890abcdef0 `
    -DestinationCapacityReservationId cr-021345abcdef56789 `
    -InstanceCount 10
```

從現有容量保留分割容量

您可從現有容量保留中分割容量來建立新保留。您可分配原先保留的一部分給特定工作負載，或與其他 AWS 帳戶共用。例如，若要與其他帳戶共用部分容量保留，您可分割部分容量，以建立規模較小的容量保留。然後可以使用 AWS Resource Access Manager 與其他帳戶分享較小規模的容量保留。

當您從現有容量保留分割容量時，會自動建立新容量保留。現有保留將維持不變，但從分割執行個體數量中減少的總容量除外。在現有容量保留中執行的執行個體不受影響。您只能將現有保留分割為一個新容量保留。

除標籤外，新容量保留將具有與現有容量保留相同的配置。在預設情況下，新容量保留無任何標籤。您可在分割操作期間指定新標籤。如有必要，也可在建立新容量保留後進行修改。

當您指定要分割的執行個體數量時，根據預設，會先分割任何可用容量，然後再分割任何符合資格的執行中執行個體 (保留中的已用容量)。例如：如果您從具有 5 個已用執行個體和 3 個可用執行個體的容量保留分割 4 個執行個體，則將分割 3 個可用執行個體和 1 個已用執行個體至新保留。

分割容量的先決條件

作為先決條件，您的容量保留必須滿足以下條件：

- 來源保留必須處於作用中狀態。
- 來源保留必須由您的 擁有 AWS 帳戶。

> **Note**
>
> 當您從保留中分割已用容量且您所指定的要分割的數量大於可用容量時，僅會分割以容量保留規格為 open 來啟動的執行個體。

考量事項

將容量從一個保留分割到新保留時，應考量以下事項：

- 使用過的容量只能分割為具有「開啟」執行個體資格且未與任何帳戶共用的容量保留。
- 當您分割已用容量時，會隨機選擇符合條件的執行個體。您無法指定分割哪些執行個體。若未找到足夠數量的合格執行個體以滿足分割數量，則分割操作將失敗。
- 從現有保留中分割出來的執行個體數量上限為保留量減一。例如，如果您的保留總容量為 5 個執行個體，您最多可以將 4 個執行個體分割為新的保留。

- 未來日期的容量保留 - 在承諾期內，您無法分割未來日期容量保留的容量。
- 容量保留資源群組 – 如果現有的容量保留屬於容量保留資源群組，則不會自動將新的容量保留新增至容量保留資源群組。如有必要，您可以將新的容量保留新增至容量保留資源群組。
- 共用 - 如果現有容量保留與消費者帳戶共用，則新容量保留不會自動與消費者帳戶共用。如有必要，您可在建立新容量保留後共用。
- 叢集置放群組 - 如果現有容量保留是叢集置放群組的一部分，則新容量保留將在相同叢集置放群組中建立。

> **Note**
>
> 不支援從容量區塊分割容量。

使用標籤控制分割容量保留的存取權

您可使用標籤來控制 Amazon EC2 資源的存取權，包括從現有容量保留分割容量來建立新容量保留。如需詳細資訊，請參閱《IAM 使用者指南》中的使用標籤控制對 AWS 資源的存取。

若要使用標籤控制分割容量保留的存取權，請務必在政策陳述式中指定資源與請求標籤，因為 IAM 政策會根據來源容量保留與新建立的容量保留進行評估。以下範例政策包含針對來源容量保留的 `ec2:ResourceTag` 條件金鑰及標籤 `Owner=ExampleDepartment1`，以及針對新建立容量保留的 `ec2:RequestTag` 條件金鑰及標籤 `stack=production`。

```
{
  "Statement": [
    {
      "Sid": "AllowSourceCapacityReservation",
      "Effect": "Allow",
      "Action": "ec2:CreateCapacityReservationBySplitting",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/
cr-1234567890abcdef0",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Owner": "ExampleDepartment1"
        }
      }
    },
    {
      "Sid": "AllowNewlyCreatedCapacityReservation",
```

```
      "Effect": "Allow",
      "Action": ["ec2:CreateCapacityReservationBySplitting", "ec2:CreateTags"],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stack": "production"
        }
      }
    }
  ]
}
```

分割容量

您可從現有容量保留中分割容量來建立新的容量保留。

Console

如需分割容量

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇容量保留。
3. 選擇可分割的隨需容量保留 ID。
4. 在動作下方的管理容量，選擇分割。
5. 在分割容量保留頁面的要分割的數量下方，使用滑桿或輸入要從目前保留中分割的執行個體數。
6. (選用) 新增標籤至新容量保留。
7. 檢閱摘要，當您準備好時，選擇分割。

AWS CLI

如需分割容量

使用 `create-capacity-reservation-by-splitting` 命令。下列範例將建立新的容量保留，方法是從指定的容量保留中分割 10 個執行個體。

```
aws ec2 create-capacity-reservation-by-splitting \
    --source-capacity-reservation-id cr-1234567890abdef0 \
    --instance-count 10
```

PowerShell

如需分割容量

使用 New-EC2CapacityReservationBySplitting cmdlet。下列範例將建立新的容量保留，方法是從指定的容量保留中分割 10 個執行個體。

```
New-EC2CapacityReservationBySplitting `
    -SourceCapacityReservationId cr-1234567890abdef0 `
    -InstanceCount 10
```

取消 容量預留

您可以取消處於以下狀態之一的容量保留：

- assessing
- scheduled
- active
- delayed

> **Note**
>
> 您無法修改或取消容量區塊。如需詳細資訊，請參閱ML 的容量區塊。

如果容量保留active沒有承諾期間或已過承諾期間，則可免費取消。

如果容量保留是 assessing或 delayed，則可免費取消。

如果容量保留為 scheduled或active剩餘承諾期間，則可能需要支付取消費用。如需詳細資訊，請參閱取消費用。

如果您取消具有執行中執行個體的容量保留，執行個體會以標準隨需執行個體費率繼續在容量保留之外正常執行，或者如果您有相符的 Savings Plans 或區域預留執行個體，則會以折扣費率繼續執行。

在您取消容量預留後，即不再啟動以其為目標的執行個體。修改這些執行個體，讓它們針對不同的容量預留、使用具相符屬性及足夠容量的任何開啟中容量預留啟動，或避免使用容量預留啟動。如需詳細資訊，請參閱修改執行個體容量保留設定。

主題

- 取消費用
- 取消引號
- 取消 容量預留

取消費用

如果您在開始日期的 8 週內或在交付後的承諾期間取消未來日期的容量保留，則會向您收取承諾期間的百分比。容量會在您取消後立即釋出，但保留會持續按隨需費率計費，直到減少的承諾期間結束為止。如果您有相符的 Savings Plans 或區域預留執行個體，則折扣費率會自動套用。

若要計算費用，Amazon EC2 會將您的承諾持續時間乘以取消率。相對於開始日期，取消的時間越早，速率越低。下表顯示取消率。

取消率

當您取消	最低取消率
原始開始日期前 8 週或以上	免費
原始開始日期前 2 到 8 週	總承諾的 25%
在原始開始日期的 2 週內	總承諾的 50%
交付後，在承諾期間	剩餘承諾的 50%

此表格提供取消費用的一般指引。取消引號會決定確切的取消費用持續時間。

例如，如果您有 70 個執行個體`c8g.xlarge`的容量保留，其承諾期間為 14 天，而且您在原始開始日期的 2 週內取消 (50% 取消率），則會向您收取承諾的 7 天費用。在這 7 天內，您需按隨需費率支付 70 個執行個體的費用。

向您收取取消費用時：

- 容量保留會進入 `cancelling` 狀態。
- 容量保留無法修改、移動、共用或分割。
- 容量保留在主控台和`DescribeCapacityReservations`回應中仍然可見。
- 承諾結束時`cancelled`，容量保留會轉換為 。

若要尋找取消費用何時結束，請使用 describe-capacity-reservations 命令，並檢查回應中的 CommitmentInfo.CommitmentEndDate 欄位。當承諾結束且容量保留轉換至 時，費用會停止cancelled。如需取消費用如何出現在 AWS 成本和用量報告 2.0 中的詳細資訊，請參閱 成本和用量報告 2.0 中的取消費用。

如果容量保留有剩餘的承諾期間，您無法在執行個體仍在執行時取消它。您必須先停止或終止執行中的執行個體，以避免同時產生取消費用和隨需執行個體費用。

若要取消收取取消費用的容量保留，您必須先產生取消報價並接受條款。如需詳細資訊，請參閱取消引號。

取消引號

當取消需要收費時，您必須先檢閱並接受取消報價，才能繼續。引號提供確切的取消條款，包括下列詳細資訊：

- 容量保留的目前組態
- 容量保留在取消後轉換至 的狀態 (cancelling)
- 費用套用至 的遞交執行個體計數
- 費用持續時間
- 費用結束日期 （ 當承諾結束且容量保留轉換為 時cancelled)

檢閱條款後，您可以透過接受 引號來確認取消。

在 Amazon EC2 主控台中，會在您啟動取消時自動產生引號。您可以檢閱條款並直接確認接受。使用 AWS CLI 或 API，使用 CreateCapacityReservationCancellationQuote產生引號，然後將引號 ID 傳遞給 CancelCapacityReservation。

取消報價有效期為 24 小時。24 小時後，您必須產生新的引號。若要檢視現有的取消引號，請使用 DescribeCapacityReservationCancellationQuotes。

> **Important**
>
> 取消是不可復原的。接受取消引號並取消容量保留後，您就無法復原取消。

取消 容量預留

使用下列程序取消容量保留。

Console

如需取消容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選取容量預留 (容量預留)，然後選取要取消的容量預留。
3. 選擇取消保留。
4. 如果需要取消報價，請檢閱取消條款，包括費用持續時間和費用結束日期。輸入 **confirm** 以接受條款。
5. 選擇取消保留。

AWS CLI

如需取消容量保留

請使用 cancel-capacity-reservation 命令。

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0
```

如果需要取消引號，請先使用 create-capacity-reservation-cancellation-quote 產生引號，然後將引號 ID 傳遞至取消命令。

```
aws ec2 create-capacity-reservation-cancellation-quote \
    --capacity-reservation-id cr-1234567890abcdef0
```

檢閱回應CancellationTerms中的 ，包括 ChargeCommitmentDurationHours和 ChargeEndDate。然後使用引號 ID 取消：

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --quote-id crcq-1a2b3c4d5e6f7g8h9 \
    --apply-cancellation-charges commitment-wind-down
```

PowerShell

如需取消容量保留

使用 Remove-EC2CapacityReservation cmdlet。

```
Remove-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0
```

如果需要取消引號，請先使用 New-EC2CapacityReservationCancellationQuote 產生引號，然後將引號 ID 傳遞給取消 cmdlet。

```
New-EC2CapacityReservationCancellationQuote `
    -CapacityReservationId cr-1234567890abcdef0
```

檢閱回應中的取消條件。然後使用引號 ID 取消：

```
Remove-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0 `
    -QuoteId crcq-1a2b3c4d5e6f7g8h9 `
    -ApplyCancellationCharge "commitment-wind-down"
```

將容量保留與置放群組搭配使用

您可以在置放群組中建立容量保留，為您的工作負載保留 Amazon EC2 運算容量。

在置放群組中建立容量保留，可確保您可以在需要時隨時存取置放群組中的運算容量。這可讓您縮減規模，同時確保容量仍可供您使用，以便在需要時進行縮減。

在置放群組中建立容量保留之後，您可以與其他 AWS 帳戶共用。如需詳細資訊，請參閱在置放群組中共用容量保留。

支援的置放策略

您可以在使用下列策略的置放群組中建立容量保留：

- 叢集
- 精確時間

> **Note**
>
> 分散和分割區置放群組不支援容量保留。容量區塊不支援置放群組。

如需支援的執行個體類型、可用區域和置放策略的相關資訊，請參閱 Amazon EC2 執行個體的置放群組。

主題

- 限制
- 在置放群組中使用容量保留
- 在置放群組中共用容量保留

限制

在置放群組中建立容量保留時，請記住下列事項：

- 若現有容量保留不在置放群組，則無法修改容量保留以在置放群組預留容量。若要預留置放群組中的容量，您必須在置放群組中建立容量保留。
- 在置放群組中建立容量保留後，您不能將其修改為置放群組之外的預留容量。
- 您可以透過修改置放群組中的現有容量保留或在置放群組中建立附加容量保留來增加置放群組中的預留容量。但是，這會增加出現容量不足錯誤的可能性。
- 您只能從您擁有的置放群組共用容量保留。您無法從非您擁有的置放群組共用容量保留。
- 您無法刪除具有`active`容量預留的置放群組。您必須先取消置放群組中的所有容量保留，才能將其刪除。

在置放群組中使用容量保留

若要開始將容量保留與置放群組搭配使用，請執行下列步驟。

> **Note**
>
> 如果您想要在現有的置放群組中建立容量保留，請略過步驟 1。然後，針對步驟 2 和 3，指定現有置放群組的 ARN。

任務

- 步驟 1：(條件式) 建立置放群組以與容量保留搭配使用
- 步驟 2：在置放群組中建立容量保留
- 步驟 3：在置放群組中的容量保留中啟動執行個體

步驟 1：(條件式) 建立置放群組以與容量保留搭配使用

只有在您需要建立新的置放群組時，才執行此步驟。若要使用現有的置放群組，請略過此步驟，然後針對步驟 2 和 3，使用該置放群組的 ARN。

Console

建立置放群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Placement Groups (置放群組)，然後選擇 Create placement group (建立置放群組)。
3. 對於 Name (名稱)，指定置放群組的描述性名稱。
4. 針對置放策略，選擇置放策略。
5. 選擇建立群組。
6. 在置放群組資料表的群組 ARN 欄中，記下您建立之置放群組的 ARN。下一個步驟需要此值。

AWS CLI

建立置放群組

使用 create-placement-group 命令。

```
aws ec2 create-placement-group \
    --group-name MyPG \
    --strategy strategy
```

記下輸出中傳回的置放群組 ARN，因為下一個步驟會需要此 ARN。

PowerShell

建立置放群組

使用 New-EC2PlacementGroup cmdlet。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy strategy
```

記下輸出中傳回的置放群組 ARN，因為下一個步驟會需要此 ARN。

步驟 2：在置放群組中建立容量保留

您在置放群組中建立容量保留的方式與建立任何容量保留的方式相同。不過，您還必須指定要在其中建立容量保留之置放群組的 ARN。

考量事項

- 指定的置放群組必須處於 `available` 狀態。如果置放群組處於 `pending`、 `deleting`或 `deleted` 狀態，則請求會失敗。
- 容量保留和置放群組必須位於相同的可用區域。如果建立容量保留的請求指定與置放群組不同的可用區域，則請求會失敗。
- 您只能針對置放群組支援的執行個體類型建立容量保留。若您指定不支援的執行個體類型，請求就會失敗。
- 如果您在置放群組中建立open容量保留，且現有的執行中執行個體具有相符的屬性（置放群組 ARN、執行個體類型、可用區域、平台和租用），則這些執行個體會自動在容量保留中執行。
- 如果符合以下條件之一，則您建立容量預留的請求可能會失敗：
  - Amazon EC2 沒有足夠的容量可滿足請求。請稍後再試、嘗試其他可用區域，或是嘗試較小的容量。如果您的工作負載因應各種執行個體類型和大小，請嘗試不同的執行個體屬性。
  - 請求的數量超過所選執行個體系列的隨需執行個體限制。請提高執行個體系列的隨需執行個體限制，然後再試一次。如需詳細資訊，請參閱隨需執行個體配額。

Console

如需建立容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 容量預留，接著選擇 Create 容量預留 (建立 容量預留)。
3. 在建立容量保留頁面上，根據需要指定執行個體類型、平台、可用區域、租用、數量和結束日期。
4. 針對置放群組，選取要建立容量保留之置放群組的 ARN。
5. 選擇建立。

如需詳細資訊，請參閱建立 容量預留。

AWS CLI

如需建立容量保留

使用 create-capacity-reservation 命令。針對 `--placement-group-arn`，指定要在其中建立容量保留之置放群組的 ARN。

```
aws ec2 create-capacity-reservation \
    --instance-type instance_type \
    --instance-platform platform \
    --availability-zone-id az_id \
    --instance-count quantity \
    --placement-group-arn "placement_group_arn"
```

PowerShell

如需建立容量保留

使用 Add-EC2CapacityReservation cmdlet。針對 `-PlacementGroupArn`，指定要在其中建立容量保留之置放群組的 ARN。

```
Add-EC2CapacityReservation `
    -InstanceType instance_type `
    -InstancePlatform platform `
    -AvailabilityZoneId az_id `
    -InstanceCount quantity `
    -PlacementGroupArn "placement_group_arn"
```

步驟 3：在置放群組中的容量保留中啟動執行個體

您可以使用下列其中一個選項，在置放群組中的容量保留中啟動執行個體：

- 指定要啟動執行個體之置放群組的 ARN – 當您提供置放群組的 ARN 時，Amazon EC2 會在該置放群組中啟動執行個體。您可以使用下列其中一種方法：
  - 指定 *open* – 不必在執行個體啟動請求中指定容量保留。如果執行個體具備與指定置放群組中的容量保留相符的屬性 (置放群組 ARN、執行個體類型、可用區域、平台和租期)，則執行個體將自動在容量保留中執行。
  - 指定容量保留 – 如果容量保留只接受目標執行個體啟動，除了請求中的置放群組之外，您還必須指定目標容量保留。

  - 指定容量保留群組 – 如需詳細資訊，請參閱在具有容量保留群組的置放群組中使用容量保留。
- 僅指定容量保留群組 – 如需詳細資訊，請參閱在具有容量保留群組的置放群組中使用容量保留。
- 僅指定容量保留 – 您可以在置放群組中的容量保留中啟動執行個體。

> **Note**
>
> 當您透過僅指定容量保留或僅指定容量保留群組來啟動執行個體時，執行個體會啟動到置放群組中建立的容量保留中，但執行個體不會直接連接到置放群組。

Console

如需使用現有的容量保留啟動執行個體

1. 請按照程序啟動執行個體，但在完成以下步驟以指定置放群組和容量保留的設定之前，請勿啟動執行個體。
2. 展開進階詳細資訊，並執行下列動作：
   a. 針對置放群組，選取要在其中啟動執行個體的置放群組。
   b. 對於 Capacity Reservation (容量保留)，根據容量保留的組態選取下列其中一個選項：
      - 開啟 – 將執行個體啟動到置放群組中具有相符屬性和足夠容量的任何open容量保留。
      - 依 ID 分類的目標 - 將執行個體啟動到僅接受目標執行個體啟動的容量保留中。
      - 依群組分類的目標 - 在所選容量保留群組內具有相符屬性和可用容量的容量保留中啟動執行個體。
3. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

如需使用現有的容量保留啟動執行個體

使用 run-instances 命令。如果您需要針對特定容量保留或容量保留群組，請指定 `--capacity-reservation-specification` 參數。對於 `--placement`，指定 `GroupName` 參數，然後指定您在上述步驟建立的置放群組的名稱。

```
aws ec2 run-instances \
```

```
    --image-id ami-0abcdef1234567890 \
    --count quantity \
    --instance-type instance_type \
    --key-name key_pair_name \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_reservation_id} \
    --placement "GroupName=placement_group_name"
```

PowerShell

如需使用現有的容量保留啟動執行個體

使用 New-EC2Instance cmdlet。對於 `-Placement`，指定 `GroupName` 參數，然後指定您在上述步驟建立的置放群組的名稱。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType instance_type `
    -KeyName key_pair_name `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId capacity_reservation_id `
    -Placement_GroupName placement_group_name
```

在置放群組中共用容量保留

您可以透過僅共用容量保留，或同時共用容量保留和建立它們的置放群組，在置放群組中共用容量保留。

您可藉由僅共用容量保留，讓取用者帳戶僅存取該容量保留。消費者帳戶對建立容量保留的置放群組沒有可見性或存取權。這樣可讓您精細控制取用者帳戶存取權。消費者帳戶無法檢視置放群組的任何相關資訊，包括其 ARN。

當您共用置放群組和容量保留時，置放群組可供消費者帳戶查看和存取。他們可啟動執行個體，並在其中建立自己的容量保留。

如需詳細資訊，請參閱下列資源。

- 在置放群組中的容量保留中啟動執行個體
- 共用容量保留
- 共用放置群組

Local Zones 中的容量保留

Local Zone 是地理位置接近您使用者的 AWS 區域的延伸。在 Local Zone 中建立的資源可以為本機使用者提供非常低延遲的通訊服務。如需詳細資訊，請參閱 AWS Local Zones。

您可以在該 Local Zone 中建立新的子網路，將 VPC 從其父 AWS 區域延伸到 Local Zone。當您在 Local Zone 中建立子網時，VPC 也會擴展到該 Local Zone。Local Zone 中的子網與 VPC 中的其他子網的運作方式相同。

透過使用 Local Zones，您可以將容量預留放置在更接近使用者的多個位置。您在 Local Zones 中建立和使用容量預留的方式，與在一般可用區域中建立和使用容量預留的方式相同。適用相同的功能和執行個體比對行為。如需 Local Zones 中支援的定價模型的詳細資訊，請參閱 AWS Local Zones 常見問答集。

考量事項

您無法在 Local Zone 中使用容量預留群組。

若要在 Local Zone 中使用容量預留

1. 啟用本機區域以用於您的帳戶 AWS 。如需詳細資訊，請參閱 AWS Local Zones 使用者指南中的 Local Zones 入門。 AWS
2. 在 Local Zone 中建立容量預留。對於 Availability Zone (可用區域)，請選擇 Local Zone。Local Zone 是以一個 AWS 區域代碼加上一個識別符來表示，用以代表位置，例如 `us-west-2-lax-1a`。如需詳細資訊，請參閱建立 容量預留。
3. 在 Local Zone 中建立子網。對於 Availability Zone (可用區域)，請選擇 Local Zone。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的在您的 VPC 中建立子網路。
4. 啟動執行個體。對於子網，選擇 Local Zone 中的子網 (例如 `subnet-123abc | us-west-2-lax-1a`)，對於容量預留，請選擇您在 Local Zone 中建立容量預留所必需的規格 (open 或透過 ID 將其鎖定為目標)。如需詳細資訊，請參閱使用現有的 容量預留 啟動執行個體。

在 Wavelength 區域中的容量保留

AWS Wavelength 可讓開發人員建立提供極低延遲的應用程式給行動裝置與最終使用者。Wavelength 將標準的 AWS 運算與儲存服務部署至電信業者 5G 網路的邊緣。您可將 Amazon Virtual Private Cloud (VPC) 擴充至一或多個 Wavelength 區域。然後 AWS ，您可以使用 Amazon EC2 執行個體之類的資源來執行需要超低延遲和區域中 AWS 服務連線的應用程式。如需詳細資訊，請參閱AWS Wavelength 區域。

當您建立隨需 容量預留 時，可以選擇 Wavelength 區域，並透過指定與 Wavelength 區域相關聯的子網，在 Wavelength 區域中對 容量預留 啟動執行個體。Wavelength 區域由 AWS 一個區域代碼，後面接著一個識別符來表示位置，例如 `us-east-1-wl1-bos-wlz-1`。

並非每個區域都有 Wavelength 區域。如需支援 Wavelength 區域的區域相關資訊，請參閱 AWS Wavelength 開發人員指南中的可用 Wavelength 區域。

考量事項

您無法使用在 Wavelength 區域的 容量預留 群組。

在 Wavelength 區域中使用 容量預留

1. 啟用 Wavelength 區域以用於您的帳戶 AWS 。如需詳細資訊，請參閱 AWS Wavelength 開發人員指南中的 AWS Wavelength入門。
2. 在 Wavelength 區域中建立 容量預留。對於可用區域，請選擇 Wavelength。Wavelength 是以一個 AWS 區域代碼加上一個識別符來表示，用以代表位置，例如 `us-east-1-wl1-bos-wlz-1`。如需詳細資訊，請參閱建立 容量預留。
3. 在 Wavelength 區域中建立子網。對於可用區域，請選擇 Wavelength 區域。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的在您的 VPC 中建立子網路。
4. 啟動執行個體。在子網中，選擇在 Wavelength 區域中的子網 (例如 `subnet-123abc | us-east-1-wl1-bos-wlz-1`)，對於 容量預留，請選擇規格 (open 或透過 ID 將其鎖定為目標)，這對於您在 Wavelength 中建立的 容量預留 為必要條件。如需詳細資訊，請參閱使用現有的 容量預留 啟動執行個體。

上的容量保留 AWS Outposts

AWS Outposts 是一種全受管服務，可將 AWS 基礎設施、服務、APIs 和工具延伸到客戶內部部署。透過提供 AWS 受管基礎設施的本機存取權， AWS Outposts 可讓客戶使用與 AWS 區域相同的程式設計界面在內部部署中建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。

Outpost 是部署在客戶站點的 AWS 運算和儲存容量集區。 會在 AWS 區域中 AWS 操作、監控和管理此容量。

您可以在帳戶中建立的 Outposts 上建立容量保留。這可讓您在站點的 Outpost 上預留運算容量。您在 Outposts 上建立和使用容量預留的方式，與在一般可用區域中建立和使用容量預留的方式相同。適用相同的功能和執行個體比對行為。

您也可以使用 與組織內的其他 AWS 帳戶共用 Outposts 上的容量預留 AWS Resource Access Manager。如需共用容量保留的資訊，請參閱 共用容量保留。

先決條件

您的站點必須安裝 Outpost。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立 Outpost 並訂購 Outpost 容量。

考量

- 您無法在 Outpost 上使用容量預留群組。

若要在 Outpost 上使用容量預留

1. 在 Outpost 上建立子網。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立子網。
2. 在 Outpost 上建立容量保留。
   a. 在 https：//https://console.aws.amazon.com/outposts/ 開啟 AWS Outposts 主控台。
   b. 在導覽窗格中，選擇 Outposts，然後選擇 Actions (動作)、Create Capacity Reservation (建立容量預留)。
   c. 視需要設定容量預留，然後選擇 Create (建立)。如需詳細資訊，請參閱建立 容量預留。

   > **Note**
   >
   > Instance Type (執行個體類型) 下拉式清單只會列出選取的 Outpost 支援的執行個體類型，而 Availability Zone (可用區域) 下拉式清單只會列出與選取的 Outpost 相關聯的可用區域。

3. 在容量預留中啟動執行個體。對於子網，選取您在步驟 1 中建立的子網，對於容量預留，選取您在步驟 2 中建立的容量預留。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的在 Outpost 上啟動執行個體。

共用容量保留

容量保留共用可讓容量保留擁有者與其他 AWS 帳戶或在 AWS 組織內共用其預留容量。這可讓您集中建立和管理容量預留，並在多個 AWS 帳戶或 AWS 組織內共用預留容量。

在此模型中，擁有容量保留 （擁有者） AWS 的帳戶會與其他 AWS 帳戶 （消費者） 共用。消費者可以在與他們共用的 容量預留 中啟動執行個體，就好像他們在自己的帳戶中擁有的 容量預留 中啟動執

行個體一樣。容量預留擁有者負責管理 容量預留和他們在其中啟動的執行個體。擁有者在他們共用的容量預留中無法修改消費者啟動的執行個體。消費者在分享給他們的 容量預留中負責管理他們啟動的執行個體。消費者無法檢視或修改其他消費者或 容量預留擁有者所擁有的執行個體。

容量預留擁有者可以將 容量預留分享給下列對象：

- AWS 組織內部或外部的特定 AWS 帳戶
- AWS 組織內的組織單位
- 其整個 AWS 組織

共用容量預留的先決條件

- 若要共用容量保留，您必須在 AWS 帳戶中擁有它。您無法將已分享給您的 容量預留 再分享出去。
- 針對租用執行個體您只能共用 容量預留。針對專用租用執行個體您無法共用 容量預留。
- 容量保留共用不適用於新的 AWS 帳戶或帳單歷史記錄有限的 AWS 帳戶。
- 若要與 AWS 組織或 AWS 組織中的組織單位共用容量保留，您必須啟用與 共用 AWS Organizations。如需詳細資訊，請參閱《AWS RAM 使用者指南》中的透過 AWS Organizations啟用共用。
- 您可以在 `active`或 `scheduled` 狀態共用容量保留。您無法在其他狀態共用容量保留，例如 `assessing`或 `unsupported`。

相關服務

容量保留共用與 AWS Resource Access Manager (AWS RAM) 整合。 AWS RAM 是一項服務，可讓您與任何 AWS 帳戶或透過 共用 AWS 資源 AWS Organizations。您可以透過 AWS RAM建立資源共享，以共用您擁有的資源。資源共享指定要共用的資源，以及共用它們的消費者。消費者可以是個別 AWS 帳戶、組織單位或整個組織的來源 AWS Organizations。

如需 的詳細資訊 AWS RAM，請參閱AWS RAM 《 使用者指南》。

跨可用區域共用

為確保資源分配至區域中的所有可用區域，可用區域會獨立對應至各個帳戶的名稱。這可能導致帳戶之間的可用區域命名出現差異。例如，`us-east-1a`您 AWS 帳戶的可用區域可能沒有與`us-east-1a`另一個 AWS 帳戶相同的位置。

若要基於您的帳戶來識別 容量預留的相對位置，您必須使用可用區域 ID (AZ ID)。AZ ID 是所有 AWS 帳戶中可用區域的唯一且一致的識別符。例如， use1-az1 是 us-east-1 區域的 AZ ID，而且在每個 AWS 帳戶中的位置都相同。

當您與其他帳戶共用容量保留時，容量保留會與其 AZ ID 識別的特定位置建立關聯。取用者帳戶只能在對應至其帳戶中相同 AZ ID 的可用區域中使用共用容量保留。例如，如果您在 us-east-1a(AZ ID use1-az1) 中建立容量保留，消費者必須在其use1-az1帳戶中映射至 的可用區域中啟動執行個體。該可用區域可能有不同的名稱，例如 us-east-1b。

檢視您帳戶中可用區域的 AZ ID

1. 在 https：//https://console.aws.amazon.com/ram/home 開啟 AWS RAM 主控台。
2. 畫面右側的 Your AZ ID (您的 AZ ID) 面板中會顯示目前區域的 AZ ID。

共用 容量預留

當您與其他人共用您擁有的容量保留時 AWS 帳戶，您可以讓他們在您的保留容量中啟動執行個體。如果您共用開啟的容量預留，請謹記下列事項，因為有可能導致容量預留用量超出預料：

- 如果消費者的執行中執行個體符合 容量預留的屬性、將 CapacityReservationPreference 參數設為 open，以及尚未執行預留容量，他們會自動使用共用的 容量預留。
- 如果消費者啟動的執行個體符合屬性 (執行個體類型、平台、可用區域與租用)，而且將 CapacityReservationPreference 參數設為 open，他們會自動在共用的容量保留中啟動執行個體。

若要分享 容量預留，您必須將它新增至資源共享。資源共用是一種 AWS RAM 資源，可讓您跨 AWS 帳戶共用資源。資源共享指定要共用的資源，以及共用它們的消費者。當您使用 Amazon EC2 主控台分享 容量預留時，您需要將它新增至現有的資源共享。若要將容量預留新增至新的資源共享，您必須使用 AWS RAM 主控台建立資源共享。

如果您是 中組織的一部分， AWS Organizations 且已啟用組織內的共用，則如果符合共用的先決條件，您組織中的取用者會獲得共用容量保留的存取權。如果容量保留與外部帳戶共享，他們會收到加入資源共享的邀請，並且在接受邀請後便能存取共享的容量保留。

> **Important**
>
> 在與您共用的容量保留中啟動執行個體之前，請在主控台中檢視共用的容量保留，或使用 describe-capacity-reservations AWS CLI 命令加以描述，以確認您能夠存取共用的容量保留。

如果您可以在 主控台中檢視共用的容量保留，或使用 描述它 AWS CLI，它可供您使用，而且您可以在其中啟動執行個體。如果您嘗試將執行個體啟動至容量保留中，但由於共享失敗而無法存取，則執行個體將啟動至隨需容量中。

Console

使用 Amazon EC2 主控台來分享您擁有的 容量預留

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選擇要分享的 容量預留，然後選擇 Actions (動作)、Share reservation (共用保留)。
4. 選取資源共享以新增 容量預留，然後選取 Share 容量預留 (共用 容量預留)。

   可能需要經過幾分鐘，消費者才能存取共用的 容量預留。

使用 AWS RAM 主控台共用您擁有的容量保留

請參閱 AWS RAM 使用者指南中的建立資源共用。

AWS CLI

如需共用您擁有的容量保留

使用 create-resource-share 命令。

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

如需共用您擁有的容量保留

使用 New-RAMResourceShare cmdlet。

```
New-RAMResourceShare `
    -Name my-resource-share `
```

```
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

停止共用容量預留

容量預留擁有者可以隨時停止共用容量預留。適用的規定如下：

- 在共用停止時，在共用容量中執行的取用者擁有的執行個體會繼續在預留容量中執行。當取用者終止執行個體時，容量會還原至容量保留。
- 共用 容量預留的消費者不能再於預留容量中啟動新的執行個體。

若要停止共用您擁有的容量預留，您必須從資源共用中移除它。

Console

如何使用 Amazon EC2 主控台停止共用您擁有的容量預留

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選取容量預留，然後選取共用索引標籤。
4. Sharing (共用) 索引標籤列出已新增 容量預留的資源共享。選取要從中移除 容量預留的資源共享，然後選取 Remove from resource share (從資源共享中移除)。

使用 AWS RAM 主控台停止共用您擁有的容量保留

請參閱《AWS RAM 使用者指南》中的更新資源共享。

AWS CLI

如需停止共用您擁有的容量保留

使用 disassociate-resource-share 命令。

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

如需停止共用您擁有的容量保留

使用 Disconnect-RAMResourceShare cmdlet。

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

共用 Amazon EC2 容量保留的帳單指派

根據預設，當共用容量保留時，向擁有者收取的費用包含其在容量保留中執行的執行個體，以及容量保留中的任何可用容量 (也稱為未用容量)；而僅向消費者收取其在共用容量保留中執行的執行個體費用。

如果需要，容量保留擁有者可以將容量保留中任何可用容量的帳單指派給與其共用容量保留的任一帳戶。帳單指派給另一帳戶後，該帳戶將成為容量保留中任何可用容量的帳單擁有者。從那時起，容量保留中可用容量的任何費用都將由指定帳戶而非擁有者的帳戶支付。容量保留擁有者和與其共用容量保留的帳戶將繼續根據其在容量保留中執行的執行個體收取費用。

> **Important**
>
> 容量保留擁有者仍為資源擁有者，並負責管理容量保留。指派帳單的帳戶無任何額外權限；無法以任何方式取消、修改或共用容量保留。

主題

- 運作方式
- 考量事項
- 將共用 EC2 容量保留的帳單指派給另一帳戶
- 檢視共用 EC2 容量保留的帳單指派請求
- 接受或拒絕共用 EC2 容量保留帳單
- 取消或撤銷共用 EC2 容量保留的帳單指派請求
- 監控共用容量保留的帳單指派請求

運作方式

只有容量保留擁有者可將共用容量保留的帳單指派給另一帳戶。帳單只能指派給與容量保留共用且合併在與容量保留擁有者相同 AWS Organizations 付款人帳戶下的帳戶。

若要將容量保留的可用容量帳單指派給另一帳戶，容量保留擁有者必須向所需帳戶提出請求。指定帳戶會收到請求，且必須在 12 小時內接受或拒絕。

- 如果對方接受，就會成為容量保留中任何可用容量 (也稱為未用容量) 的帳單擁有者。從那時起，容量保留中任何可用容量的費用都會由其帳戶而非擁有者的帳戶支付。一旦接受，只有容量保留擁有者可撤銷指定帳戶的帳單。
- 如果對方拒絕，容量保留擁有者仍是容量保留中可用容量的帳單擁有者。容量保留中任何可用容量的費用將繼續由擁有者的帳戶支付。
- 如果對方未於 12 小時內接受或拒絕請求，該請求將過期，並會繼續向擁有者的帳戶收取容量保留中任何可用容量的費用。

在帳單指派給另一帳戶期間，`Reservation` 和 `UnusedBox` 明細項目會顯示在指派帳戶的成本和用量報告 (CUR) 中，而非擁有者的 CUR。

下表顯示在將帳單指派給另一帳戶之前，容量保留擁有者和消費者帳戶的 CUR 中顯示哪些明細項目。

帳戶	指派帳單前的 CUR 明細項目
容量保留擁有者	• `Reservation` • `BoxUsage *` • `UnusedBox`
與其共用容量保留的消費者帳戶	• `BoxUsage *`

下表顯示在將帳單指派給另一帳戶之後，容量保留擁有者和消費者帳戶的 CUR 中顯示哪些明細項目。

帳戶	指派帳單後的 CUR 明細項目
容量保留擁有者	• BoxUsage *
指派帳單的使用者帳戶	• Reservation • BoxUsage * • UnusedBox
與其共用容量保留的其他消費者帳戶	• BoxUsage *

Note

- * 僅當帳戶在容量保留中擁有執行中的執行個體時，BoxUsage 明細項目才會顯示在帳戶的 CUR。如需 CUR 明細項目的詳細資訊，請參閱監控容量保留。
- 使用 CUR 的容量保留 ARN 來確定誰擁有容量保留。如果 ARN 包含 AWS 您的帳戶 ID，則您是容量保留擁有者。否則，容量保留由其他帳戶擁有，但帳單已指派給您。
- 擁有者指派給容量保留的成本分配標籤將不會顯示在消費者帳戶的 CUR。成本分配標籤只會顯示在容量保留擁有者的 CUR。

考量事項

為共用容量保留指派帳單時，請記住以下幾點：

- 您無法執行部分或分割帳單指派。容量保留的所有可用容量帳單一次僅可指派給一個帳戶。

- 容量保留的可用容量可能會隨著時間而變化。這會影響指派帳戶的帳單。例如，如果容量保留擁有者增加容量保留的大小，或者如果其他消費者帳戶停止或終止其在容量保留中執行的執行個體，則可用容量可能會增加。
- 帳單只能指派給合併在同一 AWS Organizations 付款人帳戶下的消費者帳戶。如果消費者離開組織，或不再與其分享容量保留，則系統會自動從消費者帳戶撤銷帳單。
- 在請求接受後，只有容量保留擁有者可取消待定帳單指派請求及撤銷已指派帳戶的帳單。
- 如果擁有者取消已將帳單指派給另一個帳戶的未來日期容量保留，則指派的帳單擁有者會支付取消費用。

將共用 EC2 容量保留的帳單指派給另一帳戶

若要將共用容量保留的可用容量帳單指派給另一帳戶，容量保留擁有者必須向所需帳戶提出請求。在 Amazon EC2 主控台中，此請求稱為轉移請求。

只有在以下情況，容量保留擁有者才能將容量保留的可用容量帳單指派給帳戶：

- 已與該帳戶共用容量保留。
- 帳戶會合併在與容量保留擁有者相同的 AWS Organizations 付款人帳戶下。

只有在對方接受請求後，系統才會將帳單指派給指定帳戶。

當容量保留擁有者提出請求時，會傳送 Amazon EventBridge 事件至請求的帳戶。如需詳細資訊，請參閱監控共用容量保留的帳單指派請求。

Console

指派共用容量保留的帳單

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取容量保留，然後選擇共用的容量保留。
3. 在可用容量帳單區段，選擇指派帳單。
4. 在指派帳單畫面中，選取要指派帳單的消費者帳戶，然後選擇請求。

AWS CLI

指派共用容量保留的帳單

使用 associate-capacity-reservation-billing-owner 命令。對於 `--capacity-reservation-id`，指定共用容量保留的 ID。針對 `--unused-reservation-billing-owner-id`，指定要指派帳單 AWS 的帳戶 ID。

```
aws ec2 associate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

指派共用容量保留的帳單

使用 Register-EC2CapacityReservationBillingOwner cmdlet。對於 `-CapacityReservationId`，指定共用容量保留的 ID。針對 `-UnusedReservationBillingOwnerId`，指定要指派帳單 AWS 的帳戶 ID。

```
Register-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

檢視共用 EC2 容量保留的帳單指派請求

容量保留擁有者只能檢視其所提出的最新帳單指派請求。消費者帳戶只能檢視向其傳送的最新帳單指派請求。

在請求進入 `cancelled`、`expired` 或 `revoked` 狀態之後，您可以在 24 小時內檢視該請求。24 小時過後，無法再檢視。

Console

(容量保留擁有者) 檢視您啟動的請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取容量保留，然後選擇要檢視請求的共用容量保留。
3. 可用容量帳單區段會顯示最新請求及其目前狀態。

(消費者帳戶) 已傳送給您的請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選取容量保留。
3. 若您有待定請求，畫面頂部會顯示待定帳單指派請求橫幅。如果橫幅未顯示，表示您無待定請求。

   若要檢視請求，請在橫幅中選擇檢閱請求。

AWS CLI

(容量保留擁有者) 檢視您啟動的請求

使用 describe-capacity-reservation-billing-requests 命令。

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role odcr-owner
```

(消費者帳戶) 檢視傳送給您的請求

使用 describe-capacity-reservation-billing-requests 命令。

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role unused-reservation-billing-owner
```

PowerShell

(容量保留擁有者) 檢視您啟動的請求

使用 Get-EC2CapacityReservationBillingRequest cmdlet。

```
Get-EC2CapacityReservationBillingRequest `
    -Role odcr-owner
```

(消費者帳戶) 檢視傳送給您的請求

使用 Get-EC2CapacityReservationBillingRequest cmdlet。

```
Get-EC2CapacityReservationBillingRequest `
    -Role unused-reservation-billing-owner
```

請求可以是下列任一種狀態。

State	說明
pending	請求尚未被接受或拒絕，但尚未過期。
accepted	指定帳戶已接受請求。容量保留的可用容量帳單將指派給消費者帳戶。
rejected	消費者帳戶已拒絕該請求。
cancelled	容量保留擁有者在請求處於 pending 狀態時取消了該請求。
revoked	由於以下原因，已從消費者帳戶撤銷帳單： • 已由容量保留擁有者明確撤銷。 • 已不再與消費者帳戶共用容量保留。 • 消費者帳戶不再是 AWS 組織的一部分。
expired	由於消費者帳戶未於 12 小時內接受或拒絕請求，因此請求已過期。

接受或拒絕共用 EC2 容量保留帳單

如果您收到與您共用的容量保留帳單指派請求，您可接受或拒絕。在接受或拒絕請求前，請求會維持在 pending 狀態。

如果您接受請求，請求就會進入 `accepted` 狀態，從那時起，該容量保留的任何可用或未用容量帳單都會指派至您的帳戶。接受請求後，只有容量保留擁有者可從您的帳戶撤銷帳單。

如果您拒絕請求，請求將進入 `rejected` 狀態，且將容量保留的可用容量帳單仍會指派給容量保留擁有者。

如果請求未於 12 小時內被接受或拒絕，則會過期。如果請求過期，容量保留中任何未用容量的帳單仍會指派給容量保留擁有者。

當請求被接受或拒絕時，會傳送 Amazon EventBridge 事件至容量保留擁有者的帳戶。當請求過期時，會傳送 Amazon EventBridge 事件給容量保留擁有者和消費者帳戶。如需詳細資訊，請參閱監控共用容量保留的帳單指派請求。

Console

若要接受或拒絕請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取容量保留。
3. 若您有待定請求，畫面頂部會顯示待定帳單指派請求橫幅。如果橫幅未顯示，表示您無待定請求。

   若要檢視請求，請在橫幅中選擇檢閱請求。
4. 選取要接受或拒絕的請求，然後選擇接受或拒絕。

AWS CLI

若要接受請求

使用 accept-capacity-reservation-billing-ownership 命令。對於 `--capacity-reservation-id`，請指定要接受請求的容量保留 ID。

```
aws ec2 accept-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

若要拒絕請求

使用 reject-capacity-reservation-billing-ownership 命令。對於 `--capacity-reservation-id`，請指定要拒絕請求的容量保留 ID。

```
aws ec2 reject-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

PowerShell

若要接受請求

使用 Approve-EC2CapacityReservationBillingOwnership cmdlet。

```
Approve-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

若要拒絕請求

使用 Deny-EC2CapacityReservationBillingOwnership cmdlet。

```
Deny-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

取消或撤銷共用 EC2 容量保留的帳單指派請求

只有容量保留擁有者可取消 pending 帳單指派請求。如果待定請求被取消，該請求將進入 cancelled 狀態，並且任何可用或未用容量保留的帳單仍會指派給容量保留擁有者。

請求為 accepted 之後，只有容量保留擁有者可撤銷指定帳戶的帳單。如果帳單被撤銷，請求將進入 revoked 狀態，並將容量保留的任何可用容量帳單重新指派給容量保留擁有者。

當請求被取消或撤銷時，會傳送 Amazon EventBridge 事件到容量保留擁有者和指定消費者帳戶。如需詳細資訊，請參閱監控共用容量保留的帳單指派請求。

Console

若要取消或撤銷請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取容量保留，然後選擇要取消或撤銷請求的容量保留。
3. 在可用容量帳單區段，根據請求的目前狀態，選擇取消轉移或撤銷轉移。

AWS CLI

若要取消或撤銷請求

使用 disassociate-capacity-reservation-billing-owner 命令。針對 `--unused-reservation-billing-owner-id`，指定傳送請求 AWS 的帳戶 ID。

```
aws ec2 disassociate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

若要取消或撤銷請求

使用 Unregister-EC2CapacityReservationBillingOwner cmdlet。針對 `-UnusedReservationBillingOwnerId`，指定傳送請求 AWS 的帳戶 ID。

```
Unregister-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

監控共用容量保留的帳單指派請求

在帳單指派請求狀態變更時，Amazon EC2 會傳送 Amazon EventBridge 事件。

- 當請求進入以下狀態時，將傳送事件給容量保留擁有者：`accepted` | `rejected` | `expired` | `revoked`。
- 當請求進入以下狀態時，將傳送事件到請求的消費者帳戶：`pending` | `expired` | `cancelled` | `revoked`。

如需有關 Amazon EventBridge 的詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

以下是 Amazon EventBridge 事件模式。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
```

```
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request pending|
accepted|rejected|cancelled|revoked|expired",
   "source":"aws.ec2",
   "account":"account_id",
   "time":"state_change_timestamp",
   "region":"region",
   "resources":[
      "arn:aws:ec2:region:cr_owner_account_id:capacity-reservation/cr_id"
   ],
   "detail":{
      "capacity-reservation-id":"cr_id",
      "updateTime":timestamp,
      "ownerAccountId":"cr_owner_account_id",
      "unusedReservationChargesOwnerID":"consumer_account_id",
      "status":"pending|accepted|rejected|cancelled|revoked|expired",
      "statusMessage":"message
   }
}
```

以下範例顯示當消費者帳戶 (111111111111) 接受共用容量保留 (cr-01234567890abcdef) 的帳單指派請求時，傳送給容量保留擁有者 (222222222222) 的事件。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request accepted",
   "source":"aws.ec2",
   "account":"222222222222",
   "time":"2024-09-01Thh:59:59Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:222222222222:capacity-reservation/cr-01234567890abcdef"
   ],
   "detail":{
      "capacity-reservation-id":"cr-01234567890abcdef",
      "updateTime":"2024-08-01Thh:59:59Z",
      "ownerAccountId":"222222222222",
      "unusedReservationChargesOwnerID":"111111111111",
      "status":"accepted",
      "statusMessage":"billing transfer status message"
   }
}
```

共用的容量預留許可

擁有者的許可

擁有者負責管理和取消他們共用的 容量預留。擁有者無法修改共用的 容量預留中由其他帳戶擁有的執行中執行個體。擁有者仍然負責管理他們在共用的 容量預留中啟動的執行個體。

消費者的許可

消費者負責管理他們在共用的 容量預留中執行的執行個體。消費者完全無法修改共用的 容量預留，也無法檢視或修改其他消費者或 容量預留擁有者所擁有的執行個體。消費者只能檢視共用保留中的總容量和可用容量。

計費和計量

共用 容量預留無須額外收費。

根據預設，向容量保留擁有者收取的費用包含其在容量保留中執行的執行個體和未用的預留容量，而向消費者收取的費用則是其在共用容量保留中執行的執行個體。不過，您可將共用容量保留的可用容量帳單指派給特定使用者帳戶。如需詳細資訊，請參閱共用 Amazon EC2 容量保留的帳單指派。

如果容量保留擁有者屬於不同的付款人帳戶，且區域預留執行個體或 Savings Plan 涵蓋容量保留，則會繼續針對區域預留執行個體或 Savings Plan 向容量保留擁有者收取費用。在這些情況下，容量保留擁有者會支付區域預留執行個體或 Savings Plan 的費用，並且會針對在共用容量保留中執行的執行個體向消費者收費。

執行個體限制

所有 容量預留用量都計入 容量預留擁有者的 隨需執行個體 限制。其中包括：

- 未用的預留容量
- 容量預留擁有者所擁有者的執行個體的用量
- 消費者擁有的執行個體的用量

消費者在共享容量中啟動的執行個體計入 容量預留擁有者的 隨需執行個體 限制。消費者的執行個體限制等於他們自己的 隨需執行個體 限制和他們在共用的 容量預留中可存取的可用容量的總和。

容量預留機群

隨需容量預留機群是一組容量預留。

容量預留機群要求包含啟動容量預留機群所需的所有組態資訊。使用單一要求，您可以為多個執行個體類型的工作負載預留大量 Amazon EC2 容量，最多至您指定的目標容量。

建立容量預留機群之後，您可以修改或取消容量預留機群，共同管理機群中的容量預留。

主題

- 容量預留機群的運作方式
- 考量事項
- 定價
- 容量保留機群概念與規劃
- 建立容量預留機群
- 修改容量預留機群
- 取消容量預留機群
- 容量預留機群組態範例
- 使用容量保留機群的服務連結角色

容量預留機群的運作方式

建立容量預留機群時，機群會嘗試建立個別容量預留，符合您在機群要求中指定的總目標容量。

機群預留容量的執行個體數量，取決於您指定的總目標容量與執行個體類型權重。預留容量的執行個體類型，取決於您使用的配置策略和執行個體類型優先順序。

如果建立機群時容量不足，且無法立即滿足總目標容量，則機群會非同步嘗試建立容量預留，直到已預留要求的容量數量為止。

機群達到總目標容量時，會嘗試維持該容量。如果取消機群中的容量預留，機群會根據您的機群組態，自動建立一或多個容量預留，取代失去的容量並維持總目標容量。

無法個別管理機群中的容量預留。必須修改機群來集體管理。修改機群時，機群中的容量預留會自動更新以反映變更。

容量預留機群目前支援 open 執行個體符合條件，而機群啟動的所有容量預留會自動使用此執行個體符合條件。使用此條件，具有相符屬性 (執行個體類型、平台、可用區域與租用) 的新的執行個體和現有執行個體會在機群建立的容量保留中自動執行。容量保留機群不支援目標執行個體符合條件。

考量事項

使用容量預留機群時請記住下列事項：

- 您可以使用 和 AWS API 來建立、修改、檢視 AWS CLI 和取消容量保留機群。
- 無法個別管理機群中的容量預留。必須修改或取消機群來集體管理。
- 容量預留機群無法跨區域。
- 容量預留機群無法跨可用區域。
- 容量預留機群建立的容量預留會自動加上下列 AWS 產生的標籤：
  - 索引鍵 – `aws:ec2-capacity-reservation-fleet`
  - 值 – *`fleet_id`*

  您可以使用此標籤，識別容量預留機群建立的容量預留。

定價

使用容量預留機群無須額外收費。系統會針對您容量預留機群建立的個別容量預留計費。如需容量預留計費方式的詳細資訊，請參閱容量預留定價和帳單。

容量保留機群概念與規劃

以下資訊說明如何規劃容量保留機群，並說明容量保留機群概念，包括總目標容量、分配策略、執行個體類型權重和執行個體類型優先順序。

主題

- 規劃容量保留機群
- 總目標容量
- 分配策略
- 執行個體類型權重
- 執行個體類型優先順序

規劃容量保留機群

當您規劃容量保留機群時，建議您執行下列動作：

1. 決定工作負載所需的運算容量。
2. 決定要使用的執行個體類型和可用區域。
3. 根據您的需求和偏好，為每個執行個體類型指派優先順序。如需詳細資訊，請參閱 執行個體類型優先順序。

4. 建立適合您工作負載的容量加權系統。為每個執行個體類型指派權重，並決定總目標容量。如需詳細資訊，請參閱執行個體類型權重及總目標容量。
5. 決定您是否無限期需要容量保留，或是只需要一段特定時間。

總目標容量

總目標容量定義容量預留機群預留的運算容量總量。建立容量預留機群時，您可以指定總目標容量。建立機群之後，Amazon EC2 會自動建立容量預留，最多預留至總目標容量。

容量預留機群為其預留容量的執行個體數量，取決於總目標容量，以及您為容量預留機群中每個執行個體類型指定的執行個體類型權重 (`total target capacity/instance type weight=number of instances`)。

您可以根據對您工作負載有意義的單位數量，指派總目標容量。舉例來說，如果您的工作負載需要特定數量 vCPU，您可以根據所需的 vCPU 數量，指派總目標容量。如果您的工作負載需要 2048 vCPU，請將總目標容量指派為 2048，然後根據機群中執行個體類型提供的 vCPU 數量，指派執行個體類型權重。如需範例，請參閱「執行個體類型權重」。

分配策略

容量預留機群中的分配策略，決定如何從容量預留機群組態中的執行個體類型規格，履行您的預留容量要求。

目前只支援 `prioritized` 分配策略。使用此策略，容量預留機群會使用您指派給容量預留機群組態中每個執行個體類型規格的優先順序，建立容量預留。優先順序值越低，表示使用的優先順序越高。例如，假設您建立使用下列執行個體類型和優先順序的容量預留機群：

- `m4.16xlarge` – 優先順序 = 1
- `m5.16xlarge` – 優先順序 = 3
- `m5.24xlarge` – 優先順序 = 2

機群會先嘗試為 `m4.16xlarge` 建立容量保留。如果 Amazon EC2 `m4.16xlarge` 容量不足時，機群會嘗試為 `m5.24xlarge` 建立容量保留。如果 Amazon EC2 `m5.24xlarge` 容量不足時，機群會為 `m5.16xlarge` 建立容量保留。

執行個體類型權重

執行個體類型權重是您指派給容量預留機群中每個執行個體類型的權重。權重決定計入機群總目標容量的該特定執行個體類型每個執行個體容量單位量。

您可以根據對您工作負載有意義的單位數量來指派權重。例如，如果您的工作負載需要特定數量 vCPU，您可以根據容量預留機群中每個執行個體類型提供的 vCPU 數量來指派權重。在此情況下，如果您使用 m4.16xlarge 和 m5.24xlarge 執行個體建立容量預留機群，您可以指派對應於每個執行個體 vCPU 數量的權重，如下所示：

- m4.16xlarge — 64 個 vCPU，權重 = 64 個單位
- m5.24xlarge — 96 個 vCPU，權重 = 96 個單位

執行個體類型權重決定容量預留機群為其預留容量的執行個體數量。例如，如果總目標容量為 384 單位的容量預留機群，使用先前範例中的執行個體類型和權重，則機群可以為 6 個 m4.16xlarge 執行個體 (384 總目標容量/64 執行個體類型權重 = 6 個執行個體)，或 4 個 m5.24xlarge 執行個體 (384/96 = 4) 預留容量。

如果您未指派執行個體類型權重，或將執行個體類型權重指派為 1，則總目標容量會只根據執行個體數量。舉例來說，如果總目標容量為 384 個單位的容量預留機群使用先前範例中的執行個體類型，但省略權重或對這兩個執行個體類型指定 1，機群可為 384 個 m4.16xlarge 執行個體或 384 個 m5.24xlarge 執行個體預留容量。

執行個體類型優先順序

執行個體類型優先順序是您指派給機群中執行個體類型的值。優先順序是用來決定應優先使用哪些為機群指定的執行個體類型。

優先順序值越低，表示使用的優先順序越高。

建立容量預留機群

建立容量預留機群時，它會自動為機群請求中指定的執行個體類型建立容量預留，最多達指定的總目標容量。容量預留機群預留容量的執行個體數量，取決於您在請求中指定的總目標容量和執行個體類型權重。如需詳細資訊，請參閱執行個體類型權重及總目標容量。

建立機群時，您必須指定要使用的執行個體類型，以及每個執行個體類型的優先順序。如需詳細資訊，請參閱分配策略及執行個體類型優先順序。

> **Note**
>
> 第一次建立容量預留機群時，會在您的帳戶中自動建立 AWSServiceRoleForEC2CapacityReservationFleet 服務連結角色。如需詳細資訊，請參閱 使用容量保留機群的服務連結角色。

容量預留機群目前支援 open 執行個體符合條件。

AWS CLI

若要建立容量預留機群

請使用 create-capacity-reservation-fleet 命令。

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 24 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-12-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

instanceTypeSpecification.json 的內容如下。

```
[
  {
    "InstanceType": "m5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone":"us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

以下為範例輸出。

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

PowerShell

建立容量保留機群

使用 New-EC2CapacityReservationFleet cmdlet。

```
New-EC2CapacityReservationFleet `
    -TotalTargetCapacity 24 `
    -AllocationStrategy "prioritized" `
    -InstanceMatchCriterion "open" `
    -Tenancy "default" `
    -EndDate 2021-12-31T23:59:59.000Z `
    -InstanceTypeSpecification $specification
```

規格定義如下。

```
$specification = New-Object Amazon.EC2.Model.ReservationFleetInstanceSpecification
$specification.InstanceType = "m5.xlarge"
$specification.InstancePlatform = "Linux/UNIX"
$specification.Weight = 3.0
$specification.AvailabilityZone = "us-east-1a"
$specification.EbsOptimized = $true
$specification.Priority = 1
```

修改容量預留機群

您可以隨時修改容量預留機群的總目標容量和日期。修改容量預留叢集的總目標容量時，機群會自動建立新的容量預留，或者修改或取消機群中的現有容量預留，以滿足新的總目標容量。修改機群的結束日期時，會相應地更新所有個別容量預留的結束日期。

考量事項

- 修改機群後，其狀態會轉換為 `modifying`。您無法在機群處於 `modifying` 狀態時，嘗試對機群進行額外修改。
- 您無法修改容量預留機群使用的租用、可用區域、執行個體類型、執行個體平台、優先順序或權重。如果您必須變更任一參數，可能需要取消現有機群，並以所需參數建立新機群。
- 您無法在相同命令中指定 `--end-date` 和 `--remove-end-date`。

AWS CLI

若要修改容量預留機群

請使用 modify-capacity-reservation-fleet 命令。

範例 1：修改總目標容量

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

範例 2：修改結束日期

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2021-07-04T23:59:59.000Z
```

範例 3：移除結束日期

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --remove-end-date
```

PowerShell

若要修改容量預留機群

使用 Edit-EC2CapacityReservationFleet cmdlet。

範例 1：修改總目標容量

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -TotalTargetCapacity 160
```

範例 2：修改結束日期

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -EndDate 2021-07-04T23:59:59.000Z
```

範例 3：移除結束日期

```
Edit-EC2CapacityReservationFleet `
```

```
    -CapacityReservationFleetId crf-01234567890abcedf `
    -RemoveEndDate
```

取消容量預留機群

不再需要容量預留機群及其預留的容量時，您可以取消該機群。取消機群時，其狀態會變更為 `cancelled`，且無法再建立新的容量預留。此外，將取消機群的所有個別容量保留。先前在預留容量執行的執行個體會繼續以共用容量正常執行。

AWS CLI

取消容量保留機群

請使用 cancel-capacity-reservation-fleets 命令。

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

以下為範例輸出。

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

PowerShell

取消容量保留機群

使用 Stop-EC2CapacityReservationFleet cmdlet。

```
Stop-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-abcdef01234567890
```

容量預留機群組態範例

以下範例會建立容量預留機群，此機群使用兩種執行個體類型：m5.4xlarge 和 m5.12xlarge。

它會根據指定執行個體類型提供的 vCPU 數量使用加權系統。總目標容量為 480 個 vCPU。m5.4xlarge 提供 16 個 vCPU，取得權重 16，而 m5.12xlarge 提供 48 個 vCPU，取得權重 48。此加權系統會設定容量預留機群，為 30 個 m5.4xlarge 執行個體 (480/16=30)，或 10 個 m5.12xlarge 執行個體 (480/48= 10) 預留容量。

機群設定為優先預留 m5.12xlarge 容量，取得優先順序 1，而 m5.4xlarge 取得較低的優先順序 2。這表示機群會嘗試先預留 m5.12xlarge 容量，並只在 Amazon EC2 m5.12xlarge 容量不足時，嘗試預留 m5.4xlarge 容量。

機群會為 Windows 執行個體預留容量，且該預留會在 October 31, 2021 23:59:59 UTC 自動過期。

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 480 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-10-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

instanceTypeSpecification.json 內容如下。

```
[
    {
        "InstanceType": "m5.4xlarge",
        "InstancePlatform":"Windows",
        "Weight": 16,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 2
    },
    {
        "InstanceType": "m5.12xlarge",
        "InstancePlatform":"Windows",
        "Weight": 48,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
```

```
        "Priority" : 1
    }
]
```

使用容量保留機群的服務連結角色

隨需容量保留機群使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至容量預留機群的一種特殊 IAM 角色類型。服務連結角色由容量保留機群預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定容量保留機群，因為您不必手動新增必要的許可。容量預留機群定義其服務連結角色的許可，除非另有定義，否則僅有容量預留機群可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。如此可保護您的容量預留機群資源，避免您不小心移除資源的存取許可。

容量保留機群的服務連結角色許可

容量保留機群使用名稱為 AWSServiceRoleForEC2CapacityReservationFleet 的服務連結角色，代您建立、描述、修改，以及取消容量保留機群中建立的容量保留。

AWSServiceRoleForEC2CapacityReservationFleet 服務連結角色信任下列實體可擔任該角色：

- `capacity-reservation-fleet.amazonaws.com`

角色使用 `AWSEC2CapacityReservationFleetRolePolicy` AWS 受管政策。如需詳細資訊，請參閱AWS 受管政策： AWSEC2CapacityReservationFleetRolePolicy。

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立容量保留機群的服務連結角色

您不需要手動建立服務連結角色，當您使用 `create-capacity-reservation-fleet` AWS CLI 命令或 `CreateCapacityReservationFleet` API 建立容量保留機群時，會自動為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。建立容量預留機群時，容量預留機群會再次為您建立服務連結角色。

編輯容量保留機群的服務連結角色

容量預留機群不允許您編輯 AWSServiceRoleForEC2CapacityReservationFleet 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 IAM 使用者指南中的編輯服務連結角色描述。

刪除容量保留機群的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先刪除資源。

> **Note**
>
> 若容量預留機群服務在您試圖刪除資源時正在使用該角色，刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

若要刪除 AWSServiceRoleForEC2CapacityReservationFleet 服務連結角色

1. 使用 `delete-capacity-reservation-fleet` AWS CLI 命令或 `DeleteCapacityReservationFleet` API 來刪除您帳戶中的容量保留機群。
2. 使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForEC2CapacityReservationFleet 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

容量保留機群服務連結角色的支援區域

容量預留機群支援在所有提供服務的區域中，使用服務連結角色。如需詳細資訊，請參閱 AWS 區域與端點。

使用 CloudWatch 指標監控容量保留用量

透過 CloudWatch 指標，您可以設定 CloudWatch 警示，在達到用量閾值時通知您，以有效率地監控容量預留 和識別未使用的容量。這可協助您維護經常使用的 容量預留 磁碟區，並達到更高層級的使用率。

容量保留每隔五分鐘會將指標資料傳送給 CloudWatch。作用中時間少於五分鐘的 容量預留 不支援指標。

如需有關在 CloudWatch 主控台中檢視指標的詳細資訊，請參閱使用 Amazon CloudWatch 指標。如需有關建立警示的詳細資訊，請參閱建立 Amazon CloudWatch 警示。

內容

- 容量預留 用量指標
- 容量預留 指標維度
- 檢視 容量預留 的 CloudWatch 指標

容量預留 用量指標

AWS/EC2CapacityReservations 命名空間包含下列用量指標，可讓您用來監控和維護您針對預留指定之閾值內的隨需容量。

指標	描述
UsedInstanceCount	目前正在使用的執行個體數量。 單位：計數
AvailableInstanceCount	可供使用的執行個體數量。 單位：計數
TotalInstanceCount	您已預留的執行個體總數。 單位：計數
InstanceUtilization	目前正在使用的預留容量執行個體百分比。 單位：百分比

容量預留 指標維度

您可以使用下列維度，在選定區域和帳戶強化上表中列出的指標。

維度	說明
(無維度)	此維度篩選所有容量保留的指定指標。

維度	說明
CapacityReservationId	此維度會篩選已識別容量保留的指定指標。
InstanceType	此維度會篩選已識別執行個體類型的指定指標。
AvailabilityZone	此維度會篩選已識別可用區域的指定指標。
InstanceMatchCriteria	此維度會篩選已識別執行個體符合條件 (open 或 targeted) 的指定指標。
InstancePlatform	此維度會篩選已識別平台的指定指標資料。
Tenancy	此維度會篩選已識別租用的指定指標。

檢視 容量預留 的 CloudWatch 指標

指標會先依服務命名空間分組，再依支援的維度分組。您可以使用下列程序檢視 容量預留 的指標。

使用 CloudWatch 主控台檢視 容量預留 指標

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 如有必要請變更區域。從導覽列，選取您 容量預留 所在的「區域」。如需詳細資訊，請參閱區域與端點。
3. 在導覽窗格中，選擇 Metrics (指標)。
4. 針對所有指標，選擇 EC2 容量預留。
5. 從前述指標維度中選擇，包括所有容量保留、按容量保留、按執行個體類型、按可用性區域、按平台、按執行個體匹配條件或按租用，指標將分別按以下方式分組：無維度、CapacityReservationId、InstanceType、AvailabilityZone、Platform、InstanceMa
   和 Tenancy。
6. 若要排序指標，請使用欄標題。若要將指標圖形化，請勾選指標旁的核取方塊。

使用 檢視容量保留指標 AWS CLI

使用下列 list-metrics 命令：

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

監控容量保留使用率不足

您可使用以下方式來監控容量保留使用率不足：

主題

- Amazon EventBridge 事件
- 電子郵件和 AWS Health 儀表板通知

Amazon EventBridge 事件

AWS Health 當您帳戶中的容量保留在特定期間內低於 20% 用量時，會將事件傳送至 Amazon EventBridge。您可以使用 EventBridge 建立可觸發程式設計動作的規則來回應這類事件。例如您可以建立一個「容量保留在過去 7 天的使用率低於 20% 時，自動取消該容量保留」的規則。

EventBridge 中的事件以 JSON 物件表示。事件的獨特欄位會包含在 JSON 物件的 "detail" 區段中。"event" 欄位則包含事件名稱。"result" 欄位包含觸發事件之動作的完成狀態。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的「Amazon EventBridge 事件模式」。

如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

不支援此功能 AWS GovCloud (US)。

事件

AWS Health 當容量保留的容量使用量低於 20% 時，會傳送下列事件。

- AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION

  以下例子顯示新建立的容量保留在過去 24 小時的使用量低於 20% 時產生的事件。

```
{
    "version": "0",
    "id": "b3e00086-f271-12a1-a36c-55e8ddaa130a",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
```

```
    "time": "2023-03-10T12:03:38Z",
    "region": "ap-south-1",
    "resources": [
        "cr-01234567890abcdef"
    ],
    "detail": {
        "eventArn": "arn:aws:health:ap-south-1::event/EC2/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_cr-01234567890abcdef-6211-4d50-9286-0c9fbc243f04",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION",
        "eventTypeCategory": "accountNotification",
        "startTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "endTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef"
            }
        ]
    }
    }
```

- AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY

  以下例子顯示一個或多個容量保留在過去 7 天的使用量低於 20% 時產生的事件。

```
{
    "version": "0", "id":"7439d42b-3c7f-ad50-6a88-25e2a70977e2",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-07T06:06:01Z",
    "region": "us-east-1",
    "resources": [
        "cr-01234567890abcdef | us-east-1b | t3.medium | Linux/UNIX | 0.0%",
        "cr-09876543210fedcba | us-east-1a | t3.medium | Linux/UNIX | 0.0%"
    ],
```

```
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
EC2/AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY_726c1732-d6f6-4037-b9b8-
bec3c2d3ba65",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY",
        "eventTypeCategory": "accountNotification",
        "startTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "endTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef | us-east-1b | t3.medium |
 Linux/UNIX | 0.0%"
            },
            {
                "entityValue": "cr-09876543210fedcba | us-east-1a | t3.medium |
 Linux/UNIX | 0.0%"
            }
        ]
    }
}
```

建立 EventBridge 規則

若要在容量保留使用率降至 20% 以下時收到電子郵件通知，請建立 Amazon SNS 主題，然後為 AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION 事件建立 EventBridge 規則。

建立 Amazon SNS 主題

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 在導覽窗格中選擇 Topics (主題)，然後選擇 Create topic (建立主題)。
3. 針對類型，選擇標準。
4. 在名稱中輸入新主題的名稱。

5. 請選擇建立主題。
6. 選擇建立訂閱。
7. 在通訊協定中選擇電子郵件，然後在端點中輸入接收通知的電子郵件地址。
8. 選擇建立訂閱。
9. 您會在上方輸入的電子郵件地址收到主旨列如下的電子郵件訊息：`AWS Notification - Subscription Confirmation`。請依照指示來確認訂閱。

若要建立 EventBridge 規則

1. 在 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 在導覽窗格中，選擇 Rules (規則)，然後選擇 Create rule (建立規則)。
3. 在名稱中輸入新規則的名稱。
4. 針對規則類型，選擇具有事件模式的規則。
5. 選擇下一步。
6. 在事件模式中執行下列動作：
   a. 在事件來源欄位中，選擇 AWS 服務。
   b. 在AWS 服務中選擇 AWS Health。
   c. 在事件類型中選擇 EC2 ODCR 使用率不足通知。
7. 選擇下一步。
8. 對目標 1執行下列動作：
   a. 在目標類型欄位中，選擇 AWS 服務。
   b. 對於 Select a target (選取目標)，選擇 SNS topic (SNS 主題)。
   c. 為主題選擇您剛剛建立的主題。
9. 選擇下一步，然後再選擇一次下一步。
10. 選擇建立規則。

電子郵件和 AWS Health 儀表板通知

AWS Health 當您帳戶中容量保留的容量使用率低於 20% 時， 會傳送下列電子郵件和 Health 儀板表通知。

- 每個新建立的容量保留在過去 24 小時使用率低於 20% 的個別通知。

- 所有容量保留在過去 7 天使用率低於 20% 的摘要通知。

電子郵件通知和 Health 儀板表 通知會傳送到與擁有容量預留 AWS 的帳戶相關聯的電子郵件地址。通知包含以下資訊：

- 容量保留 ID。
- 容量保留的可用區域。
- 容量保留的平均使用率。
- 容量保留的執行個體類型和平台 (作業系統)。

此外，當您的帳戶中容量保留的容量使用率在 24 小時和 7 天內低於 20% 時， 會將事件 AWS Health 傳送至 EventBridge。使用 EventBridge，您可以建立啟用自動動作的規則，例如傳送電子郵件通知或觸發 AWS Lambda 函數，以回應此類事件。如需詳細資訊，請參閱監控容量保留使用率不足。

監控未來日期容量保留的狀態變更

若未來日期容量保留的狀態變更，Amazon EC2 會傳送事件至 Amazon EventBridge。

以下為此事件的範例。在此範例中，未來日期的容量保留已進入 `scheduled` 狀態。注意 `detail-type` 欄位中醒目顯示的狀態。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"EC2 Capacity Reservation Scheduled",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"yyyy-mm-ddThh:mm:ssZ",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdefg"
   ],
   "detail":{
      "capacity-reservation-id":"cr-1234567890abcdefg",
      "state":"scheduled"
   }
}
```

`detail-type` 欄位可能的值包括：

- `Scheduled`
- `Active`
- `Delayed`
- `Unsupported`
- `Failed`
- `Expired`

如需這些狀態的詳細資訊，請參閱 檢視容量保留狀態。

您可建立 Amazon EventBridge 事件來監控這些事件，然後在事件發生時觸發特定動作。若要了解詳細資訊，請參閱在 Amazon EventBridge 中建立回應事件的規則。

如需建立針對所有狀態變更事件的規則，您可使用以下事件模式。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation"
  }]
}
```

如需建立僅針對特定狀態變更的規則，您可使用以下事件模式。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation state"
  }]
}
```

舉例來說，若未來日期的容量保留進入 `active` 狀態，以下事件模式可監控此時傳送的事件。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation Active"
  }]
}
```

可中斷的容量保留

可中斷的容量保留可協助您暫時為帳戶中的其他工作負載提供未使用的容量。這可讓您控制在需要時回收容量。當您回收容量時，中斷保留內執行的任何執行個體都會終止。建立可中斷的保留之後，您可以使用 AWS Resource Access Manager (RAM) 與其他 AWS 帳戶或 AWS 組織共用。

當您在離峰期間、部署之間或工作負載縮減時有未使用的預留容量時，請使用可中斷的容量預留。如果您知道另一個團隊可以使用此容量，您可以透過建立可中斷的容量保留來提供此容量。當您的關鍵工作負載需要取回容量時，您可以回收它。

您可以使用可中斷的容量保留做為下列其中一項：

- 容量擁有者 – 您擁有來源容量保留，並建立可中斷的容量保留，與其他團隊共用未使用的容量，同時保留控制權以在需要時回收。
- 容量取用者 – 您在共用的可中斷保留中啟動執行個體，了解當擁有者回收容量時，您的執行個體可能會終止。

主題

- 運作方式
- 帳單
- 考量事項
- 容量擁有者的可中斷容量保留
- 容量消費者的可中斷容量保留
- 使用 EventBridge 和 CloudTrail 監控可中斷的容量保留

運作方式

若要讓其他團隊使用未使用的容量，請指定您要從來源保留共用的未使用執行個體數量，以建立可中斷的保留。當您建立可中斷保留時，我們會將這些執行個體從您的來源保留轉移到您帳戶中的新可中斷保留。

我們會保留來源保留與中斷容量保留之間的關聯。因此，當您回收容量時，任何執行中的取用者執行個體都會終止，並將容量還原至原始來源保留。

主要特色

- 讓未使用的容量暫時可用，同時維持回收的控制權
- 隨時回收容量。如需詳細資訊，請參閱回收程序和追蹤

- 使用 AWS Resource Access Manager (RAM) 與其他帳戶或您的 AWS 組織共用

帳單

當您建立可中斷的保留時，系統會以獨立新保留的形式向您收費。這會分割您的帳單：

- 來源保留：系統會向您收取總容量減去配置容量的費用
- 可中斷的保留：您需要支付配置容量的費用

如需隨需容量保留帳單的詳細資訊，請參閱容量保留定價和帳單。

考量事項

使用可中斷的容量保留之前，請檢閱適用於容量擁有者和消費者的下列限制和要求。

容量擁有者

- 您無法直接修改或取消中斷容量保留。若要編輯，請更新從來源容量保留配置的容量。
- 您只能檢視、啟動、標記、共用和指派可中斷保留的帳單。
- 每個來源容量保留只能建立一個可中斷的配置。

容量取用者

- 根據預設，可中斷的容量保留是目標容量保留，因此您需要在執行個體啟動時將其設為目標。
- 您無法將可中斷的容量保留新增至容量保留群組。
- 我們建議您只對可中斷的應用程式使用可中斷的容量保留。
- 當擁有者回收容量時，您的執行個體將會終止 - 不會回復到隨需或 Spot。如需詳細資訊，請參閱中斷體驗。

容量擁有者的可中斷容量保留

容量擁有者是擁有來源容量保留的帳戶，並建立可中斷的容量保留，與其他團隊共用未使用的容量，同時保留控制權以在需要時回收。

本節涵蓋您（容量擁有者）如何建立、修改、回收和追蹤可中斷的容量保留。

主題

- 建立可中斷的容量保留

- 檢視您的中斷容量保留
- 修改您的中斷容量保留
- 將中斷容量保留保留保留為零容量
- 回收程序和追蹤
- 共用可中斷的保留

建立可中斷的容量保留

建立可中斷的容量保留，讓來源保留中未使用的容量可供其他工作負載使用，同時保持控制以在需要時回收。

先決條件

在建立可中斷的配置之前，請確定您的來源隨需容量保留符合下列要求：

- 您的容量保留必須處於作用中狀態，未設定結束日期。您無法從待定、過期、已取消或已排定結束日期的保留中建立配置。
- 您的容量保留必須具有可用的容量才能配置。您只能配置可用的執行個體（也稱為未使用的容量）。
- 每個來源容量保留只能建立一個可中斷的配置。如果配置已存在，您必須先修改或取消配置，才能建立新的配置。
- 您一次最多可以將 1000 個執行個體配置給可中斷的容量保留。

您可以使用 主控台或 AWS CLI 來建立可中斷的容量保留。

Console

建立可中斷的容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇容量預留。
3. 選取您的容量保留。
4. 選擇動作、建立可中斷的保留。
5. 針對數量，輸入要配置的執行個體數量。
6. 針對回收所有容量時的配置偏好設定，選擇預設或保留。
7. (選用) 新增標籤。

8. 選擇建立可中斷的保留。

AWS CLI

建立可中斷的容量保留

使用 create-interruptible-capacity-reservation-allocation 命令：

```
aws ec2 create-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 10
```

檢視您的中斷容量保留

建立可中斷的容量保留之後，您可以檢視帳戶中或從特定資源中可中斷的保留。

檢視您帳戶中所有可中斷的容量保留

使用下列程序來檢視您帳戶中的中斷容量保留。

Console

檢視您帳戶中的可中斷容量保留

1. 前往 主控台中的容量保留頁面。
2. 在可中斷欄中，尋找標記為是的保留。
3. 選取可中斷的保留以檢視詳細資訊。

AWS CLI

檢視您帳戶中的可中斷容量保留

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-id cr-interruptible-id \
    --filters Name=interruptible,Values=true
```

從特定來源檢視可中斷的容量保留

使用下列程序來檢視從特定來源容量保留建立的中斷容量保留。

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-id cr-source-id
```

在回應中，您會找到一個interruptibleCapacityAllocation物件，其中包含可中斷的容量保留 ID 和配置詳細資訊。如需回應結構的相關資訊，請參閱《Amazon EC2 API 參考》中的InterruptibleCapacityAllocation。

修改您的中斷容量保留

使用下列程序來編輯或取消您的中斷容量保留。

> **Note**
>
> - 當您減少配置時，我們會先回收可用的執行個體，然後執行執行個體，直到達到請求的計數為止。如果我們可以完全符合可用執行個體的計數，則不會發生終止。所有已配置執行個體計數的修改都是透過來源容量保留完成，而不是直接在可中斷的容量保留上完成。
> - 您一次最多只能修改 1000 個執行個體的可中斷容量保留 （增加或減少）。

編輯您的中斷容量保留

使用下列程序來編輯您的中斷容量保留。

Console

1. 從來源容量保留詳細資訊頁面，選擇編輯可中斷的配置連結。
2. 針對數量，輸入新號碼：
   - 新增更多容量來共用
   - 將容量回收至來源容量保留
3. 選擇更新。

AWS CLI

```
aws ec2 update-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --target-instance-count 80
```

取消您的中斷容量保留

使用下列程序永久移除配置並傳回所有容量。

Console

1. 從來源容量保留詳細資訊頁面，導覽至中斷容量分配詳細資訊。
2. 選擇編輯可中斷的配置。
3. 針對數量，輸入 0。
4. 選擇更新。

AWS CLI

```
aws ec2 update-interruptible-capacity-reservation-allocation \
--capacity-reservation-id cr-1234567890abcdef0 \
--target-instance-count 0
```

> **保留零大小偏好設定**
>
> 如果您retain針對零大小偏好設定設定 ，可中斷的容量保留會在零執行個體保持作用中狀態。若要取消，請將零大小偏好設定更新為 default，然後將配置減少為零。如需詳細資訊，請參閱將中斷容量保留保留保留為零容量。

將中斷容量保留保留保留為零容量

根據預設，當您將 Amazon EC2 的配置減少為零執行個體時，會取消可中斷的保留。回收的容量會傳回至來源容量保留。

如果您預期稍後再次共用容量，您可以將可中斷的保留設定為在零執行個體時保持作用中狀態。使用此選項，您可以稍後將執行個體配置到相同的保留，而無需重新建立。

您可以使用零大小偏好設定來控制此行為。您可以在建立或編輯中斷容量保留配置時加以設定，無論是在 主控台或使用 AWS CLI。

Console

在建立可中斷保留或編輯可中斷配置頁面上，針對回收所有容量時的配置偏好設定，選擇下列其中一項：

- 預設 – 當您將配置減少到零時，Amazon EC2 會取消中斷容量保留。
- 保留 – 當您將其配置減少為零時，可中斷的容量保留會在零容量時保持作用中狀態。然後，您可以稍後再次將執行個體配置給它。

儲存變更後，設定會在可中斷容量保留詳細資訊上顯示為唯讀零大小偏好設定欄位。

AWS CLI

若要將中斷容量保留保持為零容量，請指定您呼叫 create-interruptible-capacity-reservation-allocation 或 update-interruptible-capacity-reservation-allocation `--zero-size-preference retain`的時間。若要在配置達到零時取消保留，請指定 `default`。

```
aws ec2 update-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --zero-size-preference retain
```

回收程序和追蹤

當您回收容量時：

- 執行中的執行個體會透過 EventBridge 事件收到 2 分鐘的中斷警告。
- 在通知期間之後，在回收容量中執行的執行個體會進入關閉狀態並終止。
- 終止時，回收的執行個體可在來源容量保留中使用，以供立即使用。
- 完成時，您的配置狀態會從更新變更為作用中。

根據執行個體類型和關閉時間，完成回收可能需要幾分鐘的時間。如需程序完成時所收到 EventBridge 通知的詳細資訊，請參閱 回收完成。

追蹤回收狀態

透過描述您的來源保留來監控回收進度：

```
aws ec2 describe-capacity-reservations \
--capacity-reservation-id cr-1234567890abcdef0
```

回應會顯示 `interruptibleCapacityAllocation` 物件中的這些欄位：

- `instance-count`：目前配置的執行個體

- `target-instance-count`：回收後請求的數量
- `status`：在回收期間更新，並在完成時啟用

共用可中斷的保留

您只能使用 AWS Resource Access Manager (RAM) 在 AWS 組織內共用可中斷的保留。

考量：

- 如果取用者帳戶離開您的組織，中斷保留會自動從該帳戶取消共用。
- 在未共用保留中執行的任何執行個體最終都會終止。
- 所有其他共用功能的運作方式與標準容量預留相同。

如需完整的共用程序，請參閱共用容量預留。

容量消費者的可中斷容量保留

容量取用者是將執行個體啟動至共用可中斷容量預留的帳戶，了解當擁有者回收容量時，其執行個體可能會終止。

本節涵蓋您（容量取用者）如何在可中斷的容量保留中啟動執行個體，並了解擁有者回收容量時會發生的情況。

主題

- 檢視可中斷的容量保留
- 在可中斷的保留中啟動執行個體
- 中斷體驗

檢視可中斷的容量保留

使用下列程序來檢視可中斷的容量保留。

Console

檢視您帳戶中的可中斷容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 選擇容量預留。
3. 在可中斷欄中，尋找標記為是的保留。
4. 請注意要用於執行個體啟動IDs。

AWS CLI

尋找您帳戶中所有可中斷的容量保留

使用 describe-capacity-reservations 命令：

```
aws ec2 describe-capacity-reservations \
--filters Name=state,Values=active
```

在回應`true`中尋找`Interruptible`設定為 的保留。

專門針對可中斷的保留進行篩選

使用下列命令：

```
aws ec2 describe-capacity-reservations \
--capacity-reservation-ids cr-example123 \
--query 'CapacityReservations[?Interruptible==`true`]'
```

> **Note**
>
> 根據預設，可中斷的容量保留是目標容量保留，因此您需要在執行個體啟動時特別將它們設為目標。與開啟的保留不同，可中斷的保留不會自動涵蓋相符的執行個體。您必須在啟動時明確指定保留 ID。

在可中斷的保留中啟動執行個體

使用下列程序，將 Amazon EC2 執行個體啟動至您帳戶中的中斷容量預留。

> **Note**
>
> 建議您只對可中斷的應用程式使用可中斷的容量預留。

Console

在可中斷的容量保留中啟動執行個體

1. 在 https://console.aws.amazon.com/ec2/ : // 開啟 Amazon EC2 主控台。
2. 在 Amazon EC2 儀表板中，選擇 Launch Instance (啟動執行個體)。
3. 設定執行個體設定。
4. 在容量保留的進階詳細資訊中，選擇在作用中保留中啟動可中斷的執行個體。
5. 選取可中斷的保留 ID 和新的執行個體購買選項。
6. 選擇啟動執行個體。

AWS CLI

```
aws ec2 run-instances \
--instance-type m5.large \
--count 2 \
--image-id ami-12345678 \
--instance-market-options '{
    "MarketType": "interruptible-capacity-reservation"
}' \
--capacity-reservation-specification '{
    "CapacityReservationTarget": {
        "CapacityReservationId": "cr-abcdef1234567890"
    }
}'
```

使用 Auto Scaling 群組啟動執行個體

您也可以使用 Auto Scaling 群組搭配啟動範本，在可中斷的保留中啟動執行個體。使用可中斷的市場類型和保留 ID 設定啟動範本，然後使用該範本建立 Auto Scaling 群組。如需詳細資訊，請參閱 EC2 Auto Scaling 的可中斷容量保留。

中斷體驗

當擁有者回收容量時，您會在執行個體終止前 2 分鐘收到中斷通知。此警告是透過 EventBridge 事件，讓您有時間：

- 儲存您的工作或檢查點您的應用程式

- 關閉程序
- 準備執行個體終止

EventBridge 事件包含將終止哪些執行個體的詳細資訊，以及確切的終止時間。如需詳細資訊，請參閱執行個體中斷警告。

使用 EventBridge 和 CloudTrail 監控可中斷的容量保留

可中斷的容量保留會傳送 EventBridge 通知和 CloudTrail 事件，以協助您監控和回應容量變更。

主題

- EventBridge 通知
- CloudTrail 事件

EventBridge 通知

您會收到兩種類型的 EventBridge 通知。如需如何設定 EventBridge 通知的資訊，請參閱建立 Amazon EventBridge 規則。

執行個體中斷警告

如果您在可中斷的保留中執行執行個體，您會在執行個體終止前 2 分鐘收到此通知：

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Capacity Reservation Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "[instance owner Account ID]",
    "time": "[Current time in yyyy-mm-ddThh:mm:ssZ]",
    "resources": "[instance arn]",
    "region": "[region]",
    "detail": {
        "instance-id": "[instance-id]",
        "instance-action": "terminate",
        "instance-termination-time": "yyyy-mm-ddThh:mm:ssZ",
        "azId": "[availability-zone-id]"
    }
}
```

回收完成

如果您擁有來源保留，當容量回收完成時，您會收到此通知：

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Interruptible Capacity Reservation Allocation Reclamation
 Completed",
    "source": "aws.ec2",
    "account": "[source Capacity Reservation Owner Account ID]",
    "time": "[Current time in yyyy-mm-ddThh:mm:ssZ]",
    "region": "us-east-1",
    "resources": ["source_cr_arn"],
    "detail": {
        "sourceCapacityReservationId": "string",
        "instanceType": "string",
        "availabilityZoneId": "string",
        "TotalInstanceCount": "current total count in the source",
        "ReclaimedInstanceCount": "count of instances added to the source",
        "targetInstanceCount": "number"
    }
}
```

CloudTrail 事件

CloudTrail 會將這些事件記錄為可中斷的容量保留：

- `InterruptibleCapacityReservationCreated` — 當您建立可中斷的配置時
- `InterruptibleCapacityReservationAllocationUpdated` — 當您修改配置時
- `InterruptibleCapacityReservationCancelled` — 當您取消配置時
- `CapacityReservationModified` — 當我們修改配置的來源保留時
- `InterruptibleCapacityReservationInstancesTerminated` — 當我們在回收期間終止執行個體時

ML 的容量區塊

ML 的容量區塊可讓您為日後預留基於 GPU 的加速運算執行個體，以支援短時間的機器學習 (ML) 工作負載。在容量區塊內執行的執行個體會自動放置在 Amazon EC2 UltraClusters 內，用於低延遲、PB 級、非封鎖的網路。

您亦可使用容量區塊，來預留 Amazon EC2 UltraServers 的容量。UltraServer 以低延遲、高頻寬加速器互連，來連線多個 Amazon EC2 執行個體。您可藉助 UltraServers，來處理訓練、微調及推論中運算與記憶體最密集的 AI/ML 工作負載。若要了解詳細資訊，請參閱 Amazon EC2 UltraServer。

使用容量區塊，您可以查看未來何時可使用 GPU 執行個體容量，並且可以排定容量區塊在您最方便的時間啟動。保留容量區塊時，您可以獲得 GPU 執行個體的可預測容量保證，同時只須支付所需時間的費用。如果您需要 GPU 一次支援 ML 工作負載數天或數週，且不想在未使用 GPU 執行個體的情況下支付保留費用，建議您使用容量區塊。

以下是容量區塊的一些常見使用案例：

- ML 模型訓練和微調 – 不間斷存取保留的 GPU 執行個體，以完成 ML 模型訓練和微調。
- ML 實驗和原型 – 進行實驗並建置短時間需要 GPU 執行個體的原型。

容量區塊適用於某些 AWS 區域中的特定執行個體類型。如需詳細資訊，請參閱支援的執行個體類型和區域。

您可以保留最多未來八週內的容量區塊。每個容量區塊可包含最多 64 個執行個體，且跨容量區塊可包含最多 256 個執行個體。

主題

- 支援的執行個體類型和區域
- 支援平台
- 考量事項
- 相關資源
- Amazon EC2 容量區塊的運作方式
- 容量區塊定價和計費
- 尋找和購買容量區塊
- 使用容量區塊來啟動執行個體
- 檢視容量區塊
- 延長容量區塊
- 共用容量區塊
- 建立 UltraServer 容量區塊的容量保留資源群組
- 使用 EventBridge 監控容量區塊
- 使用 記錄容量區塊 API 呼叫 AWS CloudTrail

支援的執行個體類型和區域

執行個體和 UltraServer 容量區塊可與下列執行個體類型和 AWS 區域搭配使用。

> **Note**
>
> 並非所有 AWS 區域的所有執行個體類型都支援 64 個執行個體的容量區塊大小。

執行個體容量區塊

執行個體類型	美國東部(維吉尼亞北部) us-eas	美國東部(俄亥俄) us-eas	美國西部(加利佛尼亞北部) us-wes	美國西部(奧勒岡) us-wes	歐洲(其德哥爾摩) eu-nor 1	歐洲(倫敦) eu-wes	歐洲(西班牙) eu–sou 2	亞太區域(東京) ap-nor ast-	亞太區域(首爾) ap-nor ast-	亞太區域(孟買) ap-sou 1	亞太區域(海德拉巴) ap-sou 2	亞太區域(雪梨) ap-sou ast-	亞太區域(雅加達) ap-sou ast-	亞太區域(墨爾本) ap–sou ast-4	南美洲(聖保羅) sa-eas	AWS GovCloud(美國東部) us-gov eas t-1	AWS GovCloud(美國西部) us-gov wes t-1	美國東部(亞特蘭大) us-eas - atl-2	美國西部(鳳凰城) us-west-2 - phx-2a
p6-b300.8xla	✓			✓					✓		✓		✓			✓		✓	
p6-b200.8xla	✓	✓		✓						✓	✓					✓	✓		
p5.4x e	✓	✓		✓		✓		✓		✓		✓			✓				
p5.48 ge	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓		✓			✓	

執行個體類型	美國東部(維吉尼亞北部) us-eas	美國東部(俄亥俄) us-eas	美國西部(加利佛尼亞北部) us-wes	美國西部(奧勒岡) us-wes	歐洲(其德哥爾摩) eu-nor 1	歐洲(倫敦) eu-wes	歐洲(西班牙) eu–sou 2	亞太區域(東京) ap-nor ast-	亞太區域(首爾) ap-nor ast-	亞太區域(孟買) ap-sou 1	亞太區域(海德拉巴) ap-sou 2	亞太區域(雪梨) ap-sou ast-	亞太區域(雅加達) ap-sou ast-	亞太區域(墨爾本) ap–sou ast-4	南美洲(聖保羅) sa-eas	AWS GovCloud(美國東部) us-gov east-1	AWS GovCloud(美國西部) us-gov west-1	美國東部(亞特蘭大) us-eas -atl-2	美國西部(鳳凰城) us-west-2 -phx-2a
p5e.4 rge	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓		✓				✓
p5en. arge	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓						
p4d.2 rge	✓	✓		✓															
p4de. arge	✓			✓															
trn1. arge	✓	✓		✓	✓					✓		✓		✓					
trn2. rge														✓	✓				
trn2. arge		✓																	

UltraServer 容量區塊

執行個體類型	美國東部 (俄亥俄) us-east-2	美國東部 （達拉斯） us-east-1-dfw-2a
Trn2	✓	
P6e-GB200		✓

支援平台

ML 容量區塊目前支援具有預設租用的執行個體和 UltraServer。當您使用 AWS 管理主控台 購買容量區塊時，預設平台選項為 Linux/UNIX。當您使用 AWS Command Line Interface (AWS CLI) 或 AWS SDK 購買容量區塊時，可使用下列平台選項：

- Linux/UNIX
- Red Hat Enterprise Linux
- RHEL with HA
- SUSE Linux
- Ubuntu Pro

考量事項

使用容量區塊之前，請考慮下列詳細資訊和限制。

- 如果在 UltraServer 容量區塊上偵測到受損，您會收到通知，但通常不會在容量區塊上終止您的執行個體。這樣做是為了最大限度地減少對工作負載的非預期中斷。您可以在收到此通知後繼續使用 UltraServer 容量區塊，或藉由終止容量區塊上的所有執行個體並提交 AWS 支援案例來請求修復。收到支援案例後，系統會在修復完成時通知您，而且您可以將執行個體重新啟動至 UltraServer 容量區塊。
- 若是 P6e-GB200 UltraServer 容量區塊，必須在容量區塊結束時間之前至少 60 分鐘終止您的執行個體。
- 若要在 Local Zones 中購買和使用容量區塊，您必須選擇加入 Local Zone。
- 每個容量區塊可包含最多 64 個執行個體，且跨容量區塊可包含最多 256 個執行個體。
- 您可以描述最快 30 分鐘即可啟動的容量區塊供應項目。
- 容量區塊在國際標準時間 (UTC) 上午 11 點 30 分結束。

- 終止容量區塊中執行的執行個體程序會在保留的最後一天國際標準時間 (UTC) 上午 11 點開始。
- 可以保留最多未來 8 週內的容量區塊。
- 不可取消容量區塊。
- UltraServer 容量區塊無法跨 AWS 帳戶或在您的 AWS 組織內共用。
- 無法移動或分割容量區塊。
- 您可以將 UltraServer 容量區塊和執行個體容量區塊新增至容量保留資源群組。如需詳細資訊，請參閱容量保留資源群組。
- 容量區塊不支援置放群組。您無法在將執行個體啟動至容量區塊時指定置放群組，無論您以容量區塊的保留 ID 或容量保留資源群組為目標。
- 組織中所有帳戶在容量區塊中可以保留的執行個體總數 AWS ，在特定日期不得超過 256 個執行個體。
- 若要使用容量區塊，執行個體必須專門鎖定保留 ID。
- 容量區塊中的執行個體不會計入您的隨需執行個體限制。
- 對於使用自訂 AMI 的 P5 執行個體，請確保您擁有 EFA 所需的軟體和設定。
- 對於 Amazon EKS 受管節點群組，請參閱使用適用於 ML 的 Amazon EC2 容量區塊建立受管節點群組。對於 Amazon EKS 自我管理節點群組，請參閱使用自我管理節點來運用 ML 容量區塊。

相關資源

建立容量區塊後，您可以使用容量區塊執行以下操作：

- 在容量區塊中啟動執行個體。如需詳細資訊，請參閱使用容量區塊來啟動執行個體。
- 建立 Amazon EC2 Auto Scaling 群組。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」的使用容量區塊處理機器學習工作負載。

> **Note**
>
> 如果您使用 Amazon EC2 Auto Scaling 或 Amazon EKS，您可以在容量區塊保留開始時排定擴展。透過排程擴展， AWS 會自動為您處理重試，因此您不需要擔心實作重試邏輯來處理暫時性故障。

- 使用 AWS 平行運算服務增強 ML 工作流程。如需詳細資訊，請參閱AWS 平行運算服務的容量區塊支援。
- 使用 增強 ML 工作流程 AWS ParallelCluster。如需詳細資訊，請參閱使用 增強 ML 工作流程 AWS ParallelCluster 和 ML 的 Amazon EC2 容量區塊。

如需 AWS 平行運算服務的詳細資訊，請參閱什麼是 AWS 平行運算服務。

如需 的詳細資訊 AWS ParallelCluster，請參閱什麼是 AWS ParallelCluster。

Amazon EC2 容量區塊的運作方式

您可依照以下規格保留容量區塊：

- 提前最多 8 週來預留開始時間
- 設定保留期間：1 至 14 天或 7 天的倍數，最長 182 天 (範例：21 天、28 天)
- 每個容量區塊可設定最多 64 個執行個體
- 跨多個容量區塊可設定最多 256 個執行個體

若是 Amazon EC2 UltraServer，每個 UltraServer 對應一個容量區塊。您可透過單一請求，來請求多個 UltraServer。

若要保留容量區塊，請先指定容量需求，包括所需的執行個體類型或 UltraServer 類型、執行個體或 UltraServer 數量、時間長度、最早開始日期，以及最晚結束日期。您就可以看到符合您規格的可用容量區塊方案。容量區塊方案包含詳細資訊，例如開始時間、可用區域和保留價格。容量區塊方案的價格取決於交付方案時的可用供給與需求。保留容量區塊後，價格不會變更。如需詳細資訊，請參閱容量區塊定價和計費。

購買容量區塊方案時，系統會根據您選取的日期和執行個體數量保留。容量區塊保留開始時，您可以在啟動請求中指定保留 ID，鎖定執行個體啟動。

您可使用所有預留的執行個體，直到容量區塊結束時間前 30 分鐘 (適用於執行個體類型) 或 60 分鐘 (適用於 UltraServer 類型) 為止。在容量區塊保留中剩餘 30 分鐘 （ 例如執行個體類型 ） 或 60 分鐘 （ 適用於 UltraServer 類型 ） 的情況下，容量區塊中執行的任何執行個體都會開始終止。此時間用於清理執行個體，然後再將容量區塊交付給下一個客戶。事件會在終止程序開始前 10 分鐘透過 EventBridge 發出。如需詳細資訊，請參閱使用 EventBridge 監控容量區塊。

容量區塊定價和計費

使用適用於 ML 的 Amazon EC2 容量區塊，您只需為保留的容量付費。容量區塊的價格取決於購買時容量區塊的可用供給與需求。您可以在保留前檢視容量區塊方案的價格。我們會在保留時預先收取容量區塊的費用。當您在一系列日期中搜尋容量區塊時，會傳回價格最低的可用容量區塊方案。保留容量區塊之後，價格不會變更。

使用容量區塊時，您會為執行個體執行時使用的作業系統付費。如需作業系統定價的詳細資訊，請參閱適用於 ML 的 Amazon EC2 容量區塊訂價。

帳單

我們會預先收取容量區塊方案的費用。您購買容量區塊後 5 分鐘至 12 小時內，我們會向您的 AWS 帳戶收取費用。處理您的付款時，您的容量區塊保留資源會維持 `payment-pending` 狀態。如果無法在區塊開始時間前至少 5 分鐘處理付款，或在 12 小時內 (以較早者為準) 處理付款，系統會釋出容量區塊，並將保留狀態變更為 `payment-failed`。

成功處理您的付款後，容量區塊資源狀態會從 `payment-pending` 變更為 `scheduled`。您會收到反映一次性預付款的發票。在發票中，您可以將已付金額與容量區塊保留 ID 相關聯。

容量區塊保留開始時，系統只會根據保留中執行個體執行時，您使用的作業系統向您收費。您可以在 AWS Cost and Usage Report的年度帳單中查看使用月份的使用量和相關費用。

> **Note**
>
> Savings Plans 和預留執行個體折扣不適用於容量區塊。

檢視您的帳單

您可以在 AWS 帳單與成本管理 主控台中檢視您的帳單。容量區塊的預付款會顯示在您購買保留的月份。

保留開始後，您的帳單會分別顯示區塊保留已使用和未使用時間的明細。您可以透過這些明細項目，查看保留中使用了多少時間。如果您使用付費作業系統，只會在明細中看到已使用時間的使用費用。如需詳細資訊，請參閱容量區塊定價和計費。未使用時間不會收費。

如需詳細資訊，請參閱 AWS 帳單與成本管理 使用者指南中的檢視您的帳單。

如果您的容量區塊在與購買保留時不同的月份開始，預付費用和保留用量會顯示在不同的帳單月份下。在您的 中 AWS Cost and Usage Report，容量區塊保留 ID 會列在預付費用的預留/ReservationARN 明細項目中，以及週年紀念帳單中的 lineitem/ResourceID 中，以便您可以將用量與對應的預付價格建立關聯。

尋找和購買容量區塊

若要保留容量區塊，您必須先找到符合您需求的可用容量時間區塊。若要尋找可保留的容量區塊，請指定以下內容。

- 您需要的執行個體數量

- 您需要執行個體的時間長度
- 您需要保留的日期範圍

若要搜尋可用的容量區塊方案，請指定保留時間長度和執行個體數量。必須以 1 天為增量最多可達 14 天，以 7 天為增量最多可達 182 天來指定保留期間 每個容量區塊可包含最多 64 個執行個體，且跨容量區塊可包含最多 256 個執行個體。

當您請求符合規格的容量區塊時，最多會傳回 6 個可用區塊的詳細資訊。所有容量區塊將於 UTC 時間上午 11:30 結束，因此同日開始的區塊將具有與您所需持續時間最接近的持續時間。一個區塊的持續時間將略短於您所需的持續時間，而另一區塊的持續時間將略長於您所需的持續時間。

方案詳細資訊包括保留的開始時間、保留的可用區域，以及保留的價格。如需詳細資訊，請參閱容量區塊定價和計費。

您可以購買顯示的容量區塊方案，也可以修改搜尋條件以查看其他可用選項。方案沒有預先設定的到期時間，但僅採先到先用方式。

購買容量區塊方案時，您會立即獲得回應，確認已保留您的容量區塊。確認後，您會在帳戶中看到新的容量保留，保留類型為 `capacity-block`，且 `start-date` 設定為您所購買方案的開始時間。保留您的容量區塊時，其狀態為 `payment-pending`。成功處理預付款後，保留狀態會變更為 `scheduled`。如需詳細資訊，請參閱帳單。

> **Note**
>
> 若要在 Local Zones 中購買和使用容量區塊，您必須選擇加入 Local Zone。

Console

如需尋找及購買容量區塊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面上方的導覽列上，選取 AWS 區域。這個選擇很重要，因為並非所有區域的所有執行個體類型都支援 64 個執行個體的容量區塊大小。
3. 在導覽窗格中，依序選擇容量保留、建立容量區塊。
4. 在容量區塊類型下，選擇執行個體或 UltraServers。
5. 在容量屬性中，您可以設定容量區塊搜尋參數。根據預設，平台為 Linux。如果您要選取不同的作業系統，請使用 AWS CLI。如需詳細資訊，請參閱支援平台。

6. 在總容量 (適用於執行個體) 或 UltraServer 計數 (適用於 UltraServers) 下，指定您想要預留的執行個體或 UltraServers 數目。
7. 在持續時間下，輸入您需要保留的天數或週數。
8. 在搜尋容量區塊的日期範圍下方，輸入您希望開始保留的最早日期。
9. 選擇尋找容量區塊。
10. 如果可用的容量區塊符合您的規格，您會在建議的容量區塊下看到方案。如果有多個符合您規格的方案，則會顯示最早可用的容量區塊方案。若要檢視其他容量區塊方案，請調整您的搜尋輸入內容，並再次選擇尋找容量區塊。
11. 找到要購買的容量區塊方案時，請選擇下一步。
12. (選用) 在新增標籤頁面上，選擇新增標籤。
13. 檢閱和購買頁面會列出開始和結束日期、時間長度、執行個體總數和價格。

> **Note**
>
> 容量區塊在您保留之後就無法取消。

14. 在購買容量區塊快顯視窗中，輸入「confirm」，然後選擇購買。

AWS CLI

如需尋找執行個體容量區塊

請使用 describe-capacity-block-offerings 命令。

下面所示為尋找執行個體容量區塊的範例。

```
aws ec2 describe-capacity-block-offerings \
--instance-type p5.48xlarge \
--instance-count 16 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48 \
--all-availability-zones
```

下面所示為尋找 UltraServer 容量區塊的範例。

```
aws ec2 describe-capacity-block-offerings \
--ultraserver-type u-p6e-gb200x72 \
```

```
--ultraserver-count 1 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48
```

如需購買容量區塊

請使用上一個範例輸出中的容量區塊服務 ID 並搭配 purchase-capacity-block。

```
aws ec2 purchase-capacity-block \
--capacity-block-offering-id cb-0123456789abcdefg \
--instance-platform Linux/UNIX
```

PowerShell

如需尋找容量區塊

使用 Get-EC2CapacityBlockOffering cmdlet。

下面所示為尋找執行個體容量區塊的範例。

```
Get-EC2CapacityBlockOffering `
-InstanceType p5.48xlarge `
-InstanceCount 16 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z `
-AllAvailabilityZones $true
```

下面所示為尋找 UltraServer 容量區塊的範例。

```
Get-EC2CapacityBlockOffering `
-UltraserverType u-p6e-gb200x72 `
-UltraserverCount  1 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z
```

如需購買容量區塊

請使用上一個範例輸出中的容量區塊服務 ID 並搭配 New-EC2EC2CapacityBlock cmdlet。

```
New-EC2EC2CapacityBlock `
-CapacityBlockOfferingId cb-0123456789abcdefg `
-InstancePlatform Linux/UNIX
```

使用容量區塊來啟動執行個體

若要使用容量區塊，您必須在啟動執行個體時指定容量區塊保留 ID。在容量區塊中啟動執行個體，會依啟動的執行個體數量減少可用容量。舉例來說，如果您購買的執行個體容量為八個執行個體，而您啟動四個執行個體，則可用容量會減少四個。

如果您在保留結束之前終止在容量區塊中執行的執行個體，可以在其位置啟動新執行個體。當您停止或終止容量區塊中的執行個體時，需要幾分鐘時間清理執行個體，才能啟動另一個執行個體加以取代。在此期間，您的執行個體將處於 stopping 或 `shutting-down` 狀態。完成此程序後，您的執行個體狀態會變更為 `stopped` 或 `terminated`。接著，容量區塊中的可用容量將會更新，顯示另一個可用的執行個體。

要求

- 您的執行個體無法在與容量區塊所在可用區域不同的可用區域子網路中啟動。
- 執行個體無法透過與容量區塊平台不同的 AMI 平台啟動。
- 若要在 Local Zones 中使用容量區塊，您必須選擇加入 Local Zone。

Console

如需在容量區塊中啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在螢幕上方的導覽列上，為容量區塊保留選取區域。
3. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. 請遵循該程序來啟動執行個體。
5. 展開進階詳細資訊，並在購買選項中選擇容量區塊。然後執行下列其中一項：
   - 如需在特定容量區塊啟動執行個體，對於容量保留，請選擇指定容量保留，然後選取容量區塊。
   - (僅限 UltraServers) 若要在 UltraServer 容量區塊容量保留資源群組中啟動執行個體，請在容量保留中選擇指定容量保留資源群組，然後選取容量保留資源群組。

6. 選擇啟動執行個體。

AWS CLI

如需在容量區塊中啟動執行個體

請使用 `instance-market-options MarketType` 選項並搭配 run-instances 命令。

下面所示為在指定容量保留啟動執行個體的範例。

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p5.48xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_block_id}
```

下列範例會在 UltraServer 容量區塊容量預留資源群組中啟動執行個體。

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p6e-gb200.36xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationResourceGroupArn=resource_group_arn}
```

PowerShell

如需在容量區塊中啟動執行個體

請使用如下所示定義的 `-InstanceMarketOption` 選項並搭配 New-EC2Instance cmdlet。

```
$marketoption = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketoption.MarketType = "capacity-block"
```

下面所示為在指定容量保留啟動執行個體的範例。

```
New-EC2Instance `
-ImageId ami-0abcdef1234567890 `
-InstanceType p5.48xlarge `
-KeyName "my-key-pair" `
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationId capacity_block_id
```

下列範例會在 UltraServer 容量區塊容量預留資源群組中啟動執行個體。

```
New-EC2Instance `
-ImageId ami-0abcdef1234567890 `
-InstanceType p6e-gb200.36xlarge `
-KeyName "my-key-pair" `
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationResourceGroupArn "resource_group_arn"
```

相關資源

- 如需了解針對容量區塊建立啟動範本的資訊，請參閱 the section called “啟動範本”。
- 如需了解使用 EC2 機群在容量區塊中啟動執行個體的資訊，請參閱 教學課程：設定您的 EC2 機群以啟動執行個體至容量區塊。
- 如需了解使用容量區塊設定 EKS 受管節點群組，請參閱 Amazon EKS 使用者指南中的建立具 ML 容量區塊的受管節點群組。
- 若要 AWS ParallelCluster 使用容量區塊設定 ，請參閱 上的 ML AWS ParallelCluster。

檢視容量區塊

保留容量區塊後，您可以在 AWS 帳戶中檢視容量區塊保留。您可以檢視 `start-date` 和 `end-date`，查看保留開始和結束時間。在容量區塊保留開始之前，可用容量會顯示為零。您可以依標籤索引鍵 `aws:ec2capacityreservation:incrementalRequestedQuantity` 的標籤值，查看容量區塊中可用的執行個體數量。

容量區塊保留開始時，保留狀態會從 `scheduled` 變更為 `active`。透過 Amazon EventBridge 發出事件，通知您容量區塊可供使用。如需詳細資訊，請參閱使用 EventBridge 監控容量區塊。

容量區塊具有下列狀態：

- `payment-pending` – 尚未處理預付款。
- `payment-failed` – 無法在 12 小時內處理付款。您的容量區塊已釋出。
- `scheduled` – 已處理付款，容量區塊保留尚未開始。
- `active` – 預留容量可供使用。
- `expired` – 容量區塊保留在保留請求中指定的日期和時間自動過期。預留容量將不再提供使用。

Console

如需檢視容量區塊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 **容量預留**。
3. 在容量保留概觀頁面上，您會看到一個資源表格，其中包含所有容量保留資源的詳細資訊。若要尋找您的容量區塊保留，請從**容量保留 ID**上方的下拉式清單中選取**容量區塊**。在表格中，您可以查看容量區塊的相關資訊，例如開始和結束日期、時間長度和狀態。
4. 如需容量區塊的詳細資訊，請選取要檢視之容量區塊的保留 ID。容量保留詳細資訊頁面會顯示保留項目的所有屬性，以及容量區塊中使用中和可用的執行個體數量。

> **Note**
>
> 在容量區塊保留開始之前，可用容量會顯示為零。您可以為標籤鍵使用下列標籤值，以查看容量區塊保留開始時可用的執行個體數量：`aws:ec2capacityreservation:incrementalRequestedQuantity`。

AWS CLI

如需檢視容量區塊

使用 describe-capacity-blocks 命令來檢視容量區塊保留的詳細資訊。

```
aws ec2 describe-capacity-blocks
```

PowerShell

如需檢視容量區塊

使用 Get-EC2CapacityBlock cmdlet 來檢視容量區塊保留的詳細資訊。

```
Get-EC2CapacityBlock
```

延長容量區塊

透過容量區塊，您可為工作負載保留運算容量，確保可用性和一致性。為了滿足您不斷變化的需求，您可以根據需要延長現有容量區塊的持續時間。

若要延長容量區塊，其狀態必須為 `active` 或 `scheduled`，且不具 `payment-pending` 的擴充功能。您可以請求將容量區塊的持續時間延長至最短 1 小時，或最長 57 天才會過期。您可延長容量區塊，若以 1 天為增量最多可達 14 天，若以 7 天為增量最多可達 182 天 (26 週)。當您延長容量區塊時，其結束日期將會更新，以便執行個體可繼續執行而不會中斷。

- 您可套用至容量區塊的延長次數沒有限制
- 在延長區塊後，您的容量保留 ID 將保持不變
- 唯有在有足夠容量支援容量區塊的情況下，才能延長容量區塊，但無法保證。

帳單

我們會預先收取容量區塊方案的費用。在繳付帳單之前，延期將維持在 `payment-pending`。如果無法在 12 小時內處理付款，或在容量區塊預定結束前至少 35 分鐘處理付款 (以較早者為準)，則您的延期將無法成功，狀態將變更為 `payment-failed`。您的容量區塊保留將保持 `active`，且將在原定結束日期終止。

在成功處理付款後，容量區塊延長狀態會變更為 `payment-succeeded`，且容量區塊保留的結束日期會更新為新結束日期。延期詳細資訊可在主控台的容量區塊延期詳細資訊區段中檢視，亦可使用 describe-capacity-block-extension-history 命令檢視。

延長您的容量區塊

Console

如需延期容量區塊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 **容量預留**。
3. 在容量保留概觀頁面，您會看到一個資源表格，其中包含所有容量保留資源的詳細資訊。選取要延長的容量區塊保留 ID。

4. 從動作下拉式功能表中，選擇延長容量區塊。
5. 在持續時間下，輸入您需要延長保留的天數或週數。
6. 選擇尋找容量區塊。
7. 如果可用的容量區塊符合您的規格，建議的容量區塊下方會顯示方案。若要檢視其他容量區塊方案，請調整您的搜尋輸入內容，並再次選擇尋找容量區塊。
8. 找到要購買的容量區塊方案時，請選擇延長。
9. 在彈出視窗延長容量區塊，輸入確認，然後選擇延長。

AWS CLI

如需尋找容量區塊延期

請使用 describe-capacity-block-extension-offerings 命令。下面所示為針對指定保留搜尋 48 小時容量區塊延期的範例。

```
aws ec2 describe-capacity-block-extension-offerings \
    --capacity-reservation-id cr-1234567890abcdefg \
    --capacity-block-extension-duration-hours 48
```

如需延期容量區塊

請使用 purchase-capacity-block-extension 命令。從上一個範例的輸出中指定延期服務 ID。

```
aws ec2 purchase-capacity-block-extension \
    --capacity-block-extension-offering-id cbe-0123456789abcdefg \
    --capacity-reservation-id cr-1234567890abcdefg
```

PowerShell

如需尋找容量區塊延期

使用 Get-EC2CapacityBlockExtensionOffering cmdlet。下面所示為針對指定保留搜尋 48 小時容量區塊延期的範例。

```
Get-EC2CapacityBlockExtensionOffering `
    -CapacityReservationId cr-1234567890abcdefg `
    -CapacityBlockExtensionDurationHour 48
```

如需延期容量區塊

使用 Invoke-EC2CapacityBlockExtension cmdlet。從上一個範例的輸出中指定延期服務 ID。

```
Invoke-EC2CapacityBlockExtension `
    -CapacityBlockExtensionOfferingId cbe-0123456789abcdefg `
    -CapacityReservationId cr-1234567890abcdefg
```

共用容量區塊

容量區塊共用可讓容量區塊擁有者與 AWS 組織內的其他 AWS 帳戶共用 Amazon EC2 容量區塊。這可讓您將不同團隊和專案的預留 GPU 容量使用率最大化，以有效率地使用容量區塊。

擁有容量區塊 AWS 的帳戶 （擁有者） 可以與其他 AWS 帳戶 （消費者） 共用。擁有者可以與其 AWS 組織內的特定 AWS 帳戶、其組織內的組織單位 AWS 或整個 AWS 組織共用容量區塊。取用者可以在與其共用的容量區塊中啟動執行個體，就像在他們擁有的容量區塊中啟動執行個體一樣。

共用容量區塊的先決條件

您必須先符合下列條件，才能共用容量區塊：

- 您必須擁有容量區塊 - 您無法共用已與您共用的容量區塊。
- 容量區塊狀態必須為作用中或排程 - 處於其他狀態的容量區塊，例如 `expired`或 `payment-pending`無法共用。
- 僅限在 AWS 組織內共用 - 擁有者可以與其 AWS 組織內的特定 AWS 帳戶、其組織內的組織單位 AWS 或整個 AWS 組織共用容量區塊。
- 不支援 UltraServer 容量區塊 - 您無法共用 Amazon EC2 UltraServers 的容量區塊。
- 帳戶資格 - 容量區塊共用不適用於帳單歷史記錄有限的新 AWS 帳戶或 AWS 帳戶。

相關服務

容量區塊共用與 AWS Resource Access Manager (AWS RAM) 整合。 AWS RAM 是一項服務，可讓您與任何 AWS 帳戶或透過 共用 AWS 資源 AWS Organizations。您可以透過 AWS RAM建立資源共享，以共用您擁有的資源。資源共享指定要共用的資源，以及共用它們的消費者。消費者可以是個別 AWS 帳戶、組織單位或整個組織的來源 AWS Organizations。

如需 的詳細資訊 AWS RAM，請參閱AWS RAM 《 使用者指南》 。

共用容量區塊許可

擁有者的許可

容量區塊擁有者仍需負責管理容量區塊（例如延伸、共用），以及其在其中啟動的執行個體。擁有者無法修改消費者在共用的容量區塊中啟動的執行個體。

消費者的許可

消費者可以在共用容量中啟動執行個體，並負責管理這些執行個體。消費者無法檢視或修改其他消費者或容量區塊擁有者擁有的執行個體。消費者也只能檢視共用容量區塊中的總容量和可用容量。

共用容量區塊

若要共用容量區塊，您必須將其新增至資源共用。資源共用是可讓您跨 共用資源 AWS RAM 的資源 AWS 帳戶。

如果您將容量區塊新增至與整個 AWS 組織共用的資源共用，則組織中的取用者會獲得共用容量區塊的存取權。

Console

使用 Amazon EC2 主控台共用您擁有的容量區塊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選取要共用的容量區塊，然後選擇動作、共用保留。
4. 選取要新增容量區塊的資源共用，然後選擇共用容量保留。

   取用者可能需要幾分鐘的時間才能存取共用容量區塊。

將容量區塊新增至新的資源共享

您必須先使用 AWS RAM 主控台建立資源共享。如需詳細資訊，請參閱「AWS RAM 使用者指南」中的建立資源共用。

AWS CLI

共用您擁有的容量區塊

使用 create-resource-share 和 associate-resource-share 命令。

```
aws ram create-resource-share \
```

```
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

```
aws ram associate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

共用您擁有的容量區塊

使用 New-RAMResourceShare 和 Connect-RAMResourceShare cmdlet。

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

```
Connect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

容量區塊會以先到先服務的方式運作於所有帳戶，無論擁有權狀態為何。當您共用容量區塊時，如果取用者在擁有者之前啟動執行個體，這些執行個體會佔用容量，直到取用者終止執行個體或容量區塊過期前 30 分鐘為止。

停止共用容量區塊

您可以隨時停止共用容量區塊，直到區塊過期日期前 30 分鐘為止。

當您停止共用時會發生什麼情況：

- 消費者無法再在未共用的容量區塊中啟動新的執行個體。
- 除非消費者終止，否則任何執行中的執行個體都會繼續執行，直到容量區塊過期日期前 30 分鐘為止。

Console

使用 Amazon EC2 主控台停止共用您擁有的容量區塊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選取容量區塊，然後選擇共用標籤。
4. 共用索引標籤會列出已新增容量區塊的資源共用。選取要從中移除容量區塊的資源共用。
5. 選擇從資源共用中移除。

AWS CLI

停止共用您擁有的容量區塊

使用 disassociate-resource-share 命令。

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE
```

PowerShell

停止共用您擁有的容量區塊

使用 Disconnect-RAMResourceShare cmdlet。

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

監控共用容量區塊用量

容量區塊擁有者可以監控哪些帳戶正在使用其共用容量區塊，並追蹤每個帳戶的執行個體用量。

AWS CLI

監控容量區塊的使用情況

使用 get-capacity-reservation-usage 命令。

```
aws ec2 get-capacity-reservation-usage \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

此 API 可讓擁有者：

- 檢視哪些帳戶目前正在使用容量區塊。
- 查看每個帳戶正在執行的執行個體數量。

執行個體終止通知

在容量區塊中執行執行個體的擁有者和取用者帳戶將在容量區塊保留結束前 40 分鐘收到 EventBridge 事件，表示保留中執行的任何執行個體將在 10 分鐘內開始終止。如需詳細資訊，請參閱使用 EventBridge 監控容量區塊。

容量區塊擴充功能

容量區塊可在共用時擴充。只有擁有者帳戶可以擴展共用容量區塊。

當容量區塊延長時，擁有者或取用者啟動的執行中執行個體會自動繼承新的過期日期，而且取用者可以在新的過期日期之前繼續使用共用的容量，而不會中斷任何執行個體。

定價和計費

擁有者需支付他們共用的容量區塊費用，並在購買容量區塊時預付。擁有者也會為其在容量區塊上執行的執行個體支付作業系統費用。

消費者只會針對他們在共用容量區塊中執行的執行個體支付作業系統費用。消費者不需要支付容量區塊保留本身的費用。

建立 UltraServer 容量區塊的容量保留資源群組

您可以使用 AWS Resource Groups 建立 UltraServer 容量區塊的邏輯集合。建立容量保留資源群組之後，您可以新增帳戶中擁有的 UltraServer 容量區塊。新增 UltraServer 容量區塊後，您可以將執行個體啟動的目標設為容量保留資源群組，而不是個別的容量區塊。以容量保留資源群組為目標的執行個

體，會與群組中具有相符屬性和可用容量的任何 UltraServer 容量區塊相符。如果容量保留資源群組沒有具有相符屬性和可用容量的 UltraServer 容量區塊，執行個體啟動會失敗。

如果 UltraServer 容量區塊在執行執行個體時從容量保留資源群組中移除，這些執行個體會繼續在容量區塊中執行。如果群組中有一個 UltraServer 容量區塊在執行個體執行時結束，則會終止該等執行個體。

若要將執行個體容量區塊新增至容量保留資源群組，請參閱 容量保留資源群組。

若要為 UltraServer 容量區塊建立容量保留資源群組，請使用下列其中一種方法。

AWS CLI

為 UltraServer 容量區塊建立容量保留資源群組

請使用 create-group AWS CLI 命令，並針對 `--configuration`，指定以下各項：

```
{
  "Configuration": [
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
          "Values": [
            "AWS::EC2::CapacityReservation"
          ]
```

```
        }
      ]
    }
  ]
}
```

PowerShell

為 UltraServer 容量區塊建立容量保留資源群組

使用 New-RGGroup cmdlet。針對 `-Configuration`，指定以下各項：

```
{
  "Configuration": [
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
          "Values": [
            "AWS::EC2::CapacityReservation"
          ]
        }
      ]
    }
  ]
```

```
}
```

> **Note**
>
> 若要將各種容量保留類型新增至現有的僅限 UltraServer 群組，您必須先使用 put-group-configuration API 修改群組組態，並移除 `instance-type` 和 `reservation-type`組態區段。完成後，請參閱 容量保留資源群組 以取得更多分組操作。

建立 UltraServer 容量區塊的容量保留資源群組後，請使用下列其中一種方法來新增現有的 UltraServer 容量區塊。

AWS CLI

將 UltraServer 容量區塊新增至容量保留資源群組

請使用 group-resources 命令。對於 ，請`--group`指定您建立的容量保留資源群組名稱。對於 `--resource-arns`，請為要新增的 UltraServer 容量區塊指定 ARN。

```
aws resource-groups group-resources \
--group MyCRGroup \
--resource-arns CapacityReservationArn
```

PowerShell

將 UltraServer 容量區塊新增至容量保留資源群組

使用 Add-RGResource cmdlet。對於 ，請`-Group`指定您建立的容量保留資源群組名稱。對於 `-ResourceArn` ，請為要新增的 UltraServer 容量區塊指定 ARN。

下列範例會新增兩個容量保留到指定群組。

```
Add-RGResource `
-Group MyCRGroup `
-ResourceArn CapacityReservationArn
```

使用 EventBridge 監控容量區塊

容量區塊保留開始時，Amazon EC2 會透過 EventBridge 發出事件，表示您的容量可供使用。在容量區塊保留結束前四十分鐘，您會收到另一個 EventBridge 事件，通知您保留中執行的所有執行個體將在 10 分鐘內開始終止。如需 EventBridge 事件的詳細資訊，請參閱 Amazon EventBridge Events。

以下事件為針對容量區塊所發出事件的結構：

已提供容量區塊

以下範例說明已提供容量區塊的事件。

```
{
  "customer_event_id": "[Capacity Reservation Id]-delivered",
  "detail_type": "Capacity Block Delivered",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

容量區塊到期警告

以下範例說明容量區塊到期警告的事件。

```
{
  "customer_event_id": "[Capacity Reservation Id]-approaching-expiry",
  "detail_type": "Capacity Block Expiration Warning",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
```

```
}
```

容量保留執行個體中斷警告

下列範例顯示 EC2 容量保留執行個體中斷警告的事件。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail_type": "EC2 Capacity Reservation Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "[Customer Account ID]",
    "time": "[Current time]",
    "region": "[Region]",
    "resources": [
        "[Instance ARN]"
    ],
    "detail": {
        "instance-id": "[Instance ID]",
        "instance-action": "terminate",
        "instance-termination-time": "[Current time]",
        "availability-zone-id": "[Availability Zone ID]",
        "instance-lifecycle": "capacity-block"
    }
}
```

使用 記錄容量區塊 API 呼叫 AWS CloudTrail

容量區塊已與 服務整合 AWS CloudTrail，此服務提供使用者、角色或容量區塊中 AWS 服務所採取之動作的記錄。CloudTrail 會擷取容量區塊的 API 呼叫當作事件。這些擷取的呼叫包括來自容量區塊主控台的呼叫，以及針對容量區塊 API 操作的程式碼呼叫。若您建立追蹤，便可將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括容量區塊的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。您可以利用 CloudTrail 所收集的資訊來判斷向容量區塊發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的容量區塊資訊

當您建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當容量區塊中發生活動時，該活動會記錄在 CloudTrail 事件中，其他 AWS 服務事件則記錄於事件歷史記錄中。您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱「使用 CloudTrail 事件歷史記錄檢視事件」。

若要持續記錄 中的事件 AWS 帳戶，包括容量區塊的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

CloudTrail 會記錄所有容量區塊動作，列在 Amazon EC2 API 參考中。例如，對 `CapacityBlockScheduled` 和 `CapacityBlockActive` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解容量區塊日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例說明下列項目的 CloudTrail 日誌項目：

- TerminateCapacityBlocksInstances
- CapacityBlockPaymentFailed
- CapacityBlockScheduled
- CapacityBlockActive
- CapacityBlockFailed

- CapacityBlockExpired

> **Note**
>
> 已修改資料隱私權範例的某些欄位。

TerminateCapacityBlocksInstances


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "TerminateCapacityBlockInstances",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
    }
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      }
```

```
}
```

CapacityBlockPaymentFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockPaymentFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "payment-failed"
      }
}
```

CapacityBlockScheduled

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
```

```
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockScheduled",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "scheduled"
      }
}
```


CapacityBlockActive


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockActive",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
```

```
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "active"
      }
 }
```

CapacityBlockFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
```

```
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "failed"
      }
 }
```

CapacityBlockExpired

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockExpired",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "expired"
      }
 }
```

容量保留資源群組

您可以使用 AWS Resource Groups 建立容量保留的邏輯集合，稱為容量保留資源群組（群組）。容量保留資源群組是包含容量保留的資源群組，可讓您透過指定單一容量保留資源群組 ARN，在多

個容量保留中啟動執行個體。如需容量保留資源群組的詳細資訊，請參閱什麼是資源群組？ AWS Resource Groups 《 使用者指南》中的 。

當您指定要啟動執行個體的群組時，Amazon EC2 會將執行個體與群組中具有相符屬性和可用容量的任何容量保留進行比對。

您可以包含您帳戶中擁有的容量預留，以及單一群組中其他 AWS 帳戶與您共用的容量預留。您也可以在單一群組中包含具有不同屬性（執行個體類型、平台、可用區域、租用和置放群組）的容量預留。

群組支援所有保留類型，包括隨需容量保留、可中斷的容量保留，以及 ML 的容量區塊。您可以將保留類型的任何組合新增至單一群組。

任務

- 建立群組
- 將容量保留新增至群組
- 在群組中的容量保留中啟動執行個體
- 從群組移除容量保留
- 群組中的執行個體生命週期
- 刪除群組

建立群組

您可以建立群組來組織容量保留。在您建立群組之後，可新增容量保留至群組。

Console

建立群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選擇容量保留資源群組。
4. 選擇 Create Group (建立群組)。
5. 輸入群組名稱，並選擇性地新增描述、容量保留和標籤。
6. 選擇建立。

AWS CLI

為容量保留建立群組

使用 create-group AWS CLI 命令搭配下列請求參數。

- AWS::EC2::CapacityReservationPool – 確保容量保留資源群組可以成為執行個體啟動的目標。
- AWS::ResourceGroups::Generic allowed-resource-types設定為 AWS::EC2::CapacityReservation ：確保容量保留資源群組僅接受容量保留。

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

PowerShell

為容量保留建立群組

使用 New-RGGroup cmdlet。

```
New-RGGroup `
    -Name MyCRGroup `
    -Configuration `
        @{"Type"="AWS::EC2::CapacityReservationPool"} `
        @{"Type"="AWS::ResourceGroups::Generic"; "Parameters"=@{"allowed-resource-
types"=@{"Values"="AWS::EC2::CapacityReservation"}}}
```

若要建立僅接受特定執行個體類型 UltraServer 容量區塊的群組，請參閱 建立 UltraServer 容量區塊的容量保留資源群組。

將容量保留新增至群組

您可以將您帳戶中擁有的容量預留，或由其他 AWS 帳戶與您共用的容量預留新增至群組。

考量事項

- 您只能將 `active` 狀態的容量保留新增至群組。
- 如果共用容量保留稍後由擁有者取消共用，Amazon EC2 會自動將其從您的群組中移除。

Console

您可以從容量保留資源群組詳細資訊頁面或容量保留主控台，將容量保留新增至群組。

從容量保留資源群組頁面新增容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選擇容量保留資源群組。
4. 選取群組名稱以開啟其詳細資訊頁面。
5. 在容量保留區段中，選擇新增。
6. 選取您要新增的容量保留，然後選擇新增。

從容量保留主控台新增容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選取您要新增至群組的容量保留。
4. 選擇動作，然後選擇新增至群組。
5. 選取群組，然後選擇新增。

AWS CLI

將容量保留新增至群組

請使用 group-resources 命令。提供您要新增的每個容量保留的 ARN。

下列範例會新增兩個容量保留到指定群組。

```
aws resource-groups group-resources \
    --group MyCRGroup \
```

```
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1
\
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

將容量保留新增至群組

使用 Add-RGResource cmdlet。

下列範例會新增兩個容量保留到指定群組。

```
Add-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-1234567890abcdef1", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

在群組中的容量保留中啟動執行個體

您可以透過指定容量保留資源群組 ARN，在現有的容量保留中啟動執行個體。以群組為目標的執行個體，會與群組中具有相符屬性（執行個體類型、平台、可用區域和租用）和可用容量的任何容量保留相符。

當您使用啟動執行個體精靈或 RunInstances API 啟動執行個體時，您每次啟動只能指定一個購買選項（一個保留類型）。 RunInstances 若要一次啟動群組中的多個保留類型，請使用 Amazon EC2 Fleet 或 Auto Scaling 群組。

如需如何使用容量保留資源群組在多個容量保留類型中啟動執行個體step-by-step演練，請參閱下列教學課程：透過 EC2 機群和透過 Amazon EC2 Auto Scaling。

隨需容量保留

若要在群組中的隨需容量保留中啟動執行個體，請使用 `CapacityReservationTarget` 參數搭配 `CapacityReservationResourceGroupArn` 欄位來指定群組 ARN。

Amazon EC2 會在群組中任何相符的隨需容量保留中啟動執行個體。如果沒有可用的相符容量，執行個體會使用隨需容量啟動。

如果您不希望執行個體在沒有相符容量可用時以隨需形式啟動，請將 CapacityReservationPreference設定為 capacity-reservations-only。使用此設定，如果群組中沒有相符的隨需容量保留具有可用容量，則執行個體無法啟動。

Console

在群組中的隨需容量保留中啟動執行個體

1. 請依照程序啟動執行個體，但在您完成下列步驟之前，請不要啟動執行個體。
2. 展開 Advanced Details (進階詳細資訊)。
3. 對於購買選項，將 保留為無 （預設）。
4. 對於容量保留，選擇指定容量保留資源群組 （預設） 或僅在容量保留資源群組中啟動 （以防止隨需備用），然後選取群組。
5. 選擇啟動執行個體。

AWS CLI

在群組中的隨需容量保留中啟動執行個體

使用 run-instances 命令搭配 --capacity-reservation-specification 參數。

下列範例會在群組中啟動執行個體。如果群組沒有具有相符屬性和可用容量的 ODCR，執行個體會以隨需執行個體的形式啟動。

```
aws ec2 run-instances \
    --instance-type c5.xlarge \
    --image-id ami-0abcdef1234567890 \
    --count 3 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup}
```

下列範例會在群組中啟動執行個體。如果群組沒有具有相符屬性和可用容量的 ODCR，則執行個體無法啟動。

```
aws ec2 run-instances \
    --instance-type c5.xlarge \
    --image-id ami-0abcdef1234567890 \
    --count 3 \
```

```
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-
only,CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:sa-east-1:123456789012:group/MyCRGroup}
```

以特定置放群組為目標

或者，如果您的 群組在置放群組中包含隨需容量保留，您也可以指定置放群組 ID 以及容量保留資源群組 ARN，將所有啟動的執行個體放入該置放群組。

> **Note**
>
> 容量區塊不支援置放群組。當您的請求以容量保留資源群組中的容量區塊為目標時，您無法指定置放群組。

可中斷的容量保留

若要在群組中的可中斷容量保留中啟動執行個體，請指定群組 ARN，然後選取可中斷的購買選項。Amazon EC2 會將執行個體放入群組中具有相符屬性和可用容量的任何作用中中斷容量保留。如果沒有可用的相符容量，執行個體將無法啟動。

Console

在群組中啟動執行個體至可中斷的容量保留

1. 請依照程序啟動執行個體，但在您完成下列步驟之前，請不要啟動執行個體。
2. 展開 Advanced Details (進階詳細資訊)。
3. 針對購買選項，選擇可中斷的容量保留。
4. 針對容量保留，選擇指定容量保留資源群組，然後選取群組。
5. 選擇啟動執行個體。

AWS CLI

在群組中啟動執行個體至可中斷的容量保留

使用 run-instances 命令搭配 `--instance-market-options`和 `--capacity-reservation-specification` 參數。

```
aws ec2 run-instances \
    --instance-type c5.xlarge \
    --image-id ami-0abcdef1234567890 \
    --count 3 \
    --instance-market-options MarketType=interruptible-capacity-reservation \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:sa-east-1:123456789012:group/MyCRGroup}
```

容量區塊

若要在群組中的容量區塊中啟動執行個體，請指定群組 ARN，然後選取容量區塊購買選項。Amazon EC2 會將執行個體放入群組中具有相符屬性和可用容量的任何作用中容量區塊。如果沒有可用的相符容量，執行個體將無法啟動。

> **Note**
>
> 在容量保留資源群組中的容量區塊中啟動時，您無法指定置放群組。

Console

在群組中的容量區塊中啟動執行個體

1. 請依照程序啟動執行個體，但在您完成下列步驟之前，請不要啟動執行個體。
2. 展開 Advanced Details (進階詳細資訊)。
3. 針對購買選項，選擇容量區塊。
4. 針對容量保留，選擇指定容量保留資源群組，然後選取群組。
5. 選擇啟動執行個體。

AWS CLI

在群組中的容量區塊中啟動執行個體

使用 run-instances 命令搭配 `--instance-market-options`和 `--capacity-reservation-specification` 參數。

```
aws ec2 run-instances \
    --instance-type p5.48xlarge \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-market-options MarketType=capacity-block \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:sa-east-1:123456789012:group/MyCRGroup}
```

從群組移除容量保留

您可以隨時從群組中移除容量保留。以群組為目標的執行個體會繼續在其目前的容量保留中執行，且不會受到移除的影響。容量保留只會在您終止執行個體時還原容量。

> **Note**
>
> 如果稍後未共用與您共用的容量保留，Amazon EC2 會自動將其從群組中移除。

Console

從群組中移除容量保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選擇容量保留資源群組。
4. 選取群組名稱以開啟其詳細資訊頁面。
5. 在容量保留區段中，選取要移除的容量保留，然後選擇移除。

AWS CLI

從群組中移除容量保留

請使用 ungroup-resources 命令。

下列範例會從指定群組中移除兩個容量保留。

```
aws resource-groups ungroup-resources \
```

```
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd
\
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

從群組移除容量保留

使用 Remove-RGResource cmdlet。

下列範例會從指定群組中移除兩個容量保留。

```
Remove-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-0e154d26a16094dd", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

群組中的執行個體生命週期

當群組中的容量保留變更狀態時，對執行中執行個體的影響取決於保留類型。

隨需容量保留

當群組中的隨需容量保留取消、過期或容量減少時，目標為群組的執行個體會與群組中具有相符屬性（執行個體類型、平台、可用區域和租用）和可用容量的任何其他隨需容量保留相符。如果群組沒有具有相符屬性和可用容量的隨需容量保留，則執行個體會使用隨需容量執行。如果您稍後將相符的隨需容量保留新增至群組，Amazon EC2 會自動將執行個體移至其預留容量。

可中斷的容量保留

當回收可中斷容量保留的容量時，執行個體會在終止前 2 分鐘收到中斷通知。如需詳細資訊，請參閱中斷體驗。

容量區塊

容量保留資源群組中容量區塊中執行的執行個體會在容量區塊結束之前終止，與您直接鎖定的容量區塊相同。如需詳細資訊，請參閱Amazon EC2 容量區塊的運作方式。

刪除群組

如果您不再需要您的 群組，您可以隨時將其刪除。當您刪除具有執行中執行個體和作用中容量保留的群組時，目標為該群組的執行個體會在群組刪除後繼續在其目前的容量保留中執行。當您終止執行個體時，Amazon EC2 會將容量還原至容量保留。

Console

刪除群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 容量預留。
3. 選擇容量保留資源群組。
4. 選取您要刪除的群組，選擇動作，然後選擇刪除。
5. 出現確認提示時，請選擇刪除。

AWS CLI

刪除群組

請使用 delete-group 命令。

```
aws resource-groups delete-group --group MyCRGroup
```

PowerShell

刪除群組

使用 Remove-RGGroup cmdlet。

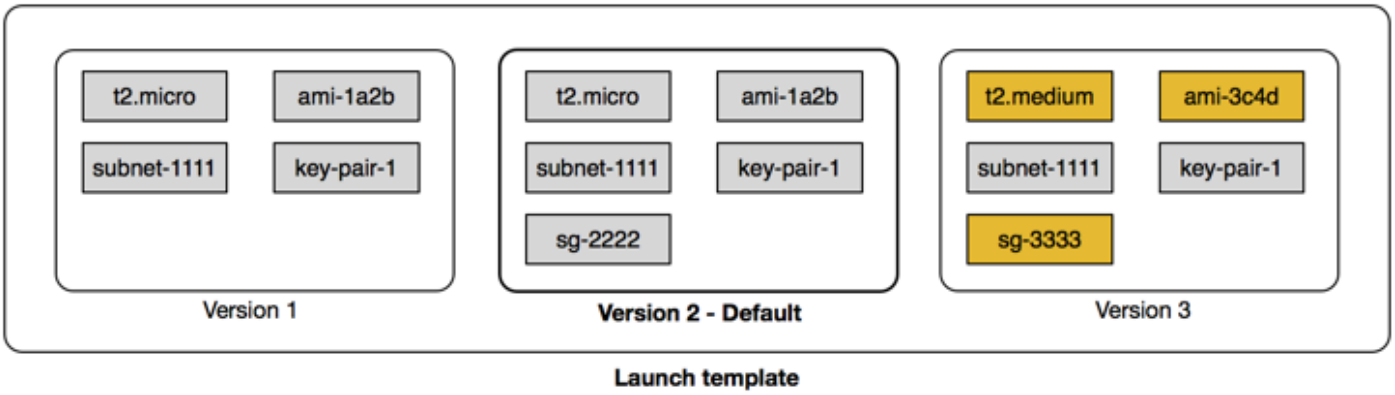
```
Remove-RGGroup -GroupName MyCRGroup
```

將執行個體啟動參數存放在 Amazon EC2 啟動範本中

您可以使用 Amazon EC2 啟動範本來存放執行個體啟動參數，如此您就不需要在每次啟動 Amazon EC2 執行個體時指定參數。例如，您可以建立啟動範本，當中存放您平常用來啟動執行個體的 AMI ID、執行個體類型和網路設定。當您使用 Amazon EC2 主控台、 AWS SDK 或命令列工具啟動執行個體時，您可以指定啟動範本，而不是再次輸入參數。

針對每個啟動範本，您可以建立一或多個已編號的 啟動範本版本。每個版本都可以有不同的啟動參數。當您從啟動範本啟動執行個體時，您可以使用啟動範本的任何版本。如果您未指定版本，則會使用預設版本。您可以設定任意版本的啟動範本做為預設—版本，而該版本即預設為啟動範本的第一個版本。

下圖顯示具有三個版本的啟動範本。第一個版本指定用來啟動執行個體的執行個體類型、AMI ID、子網和金鑰對。第二個版本以第一個版本為基礎，同時指定執行個體的安全群組。第三個版本針對部分參數使用不同的值。第 2 版設定為預設版本。如果您已從此啟動範本啟動執行個體，則會在未指定其他版本時使用第 2 版的啟動參數。



目錄

- Amazon EC2 啟動範本的限制
- Amazon EC2 啟動範本的必要 IAM 許可
- 使用 Amazon EC2 啟動範本，來控制啟動 Amazon EC2 執行個體
- 建立 Amazon EC2 啟動範本
- 修改啟動範本 (管理啟動範本版本)
- 刪除啟動範本或啟動範本版本

Amazon EC2 啟動範本的限制

啟動範本和啟動範本版本的限制如下：

- 配額 – 若要檢視啟動範本和啟動範本版本的配額，請開啟 Service Quotas 主控台或使用 list-service-quotas AWS CLI 命令。每個 AWS 帳戶每個區域最多可有 5，000 個啟動範本，每個啟動範本最多可有 10，000 個版本。您帳戶的配額可能不太一樣，視年齡和使用記錄而定。
- 參數是選用值 – 啟動範本參數是選用的。不過，您必須確定執行個體啟動請求包括所有必要參數。例如，如果您的啟動範本未包括 AMI ID，則必須在使用此啟動範本啟動執行個體時，指定 AMI ID。

- 參數尚未驗證 – 在您建立啟動範本時，啟動範本參數尚未完整驗證。如果您指定了不正確的值，或是使用未支援的參數組合，則執行個體無法使用此啟動範本進行啟動。為了避免問題，請務必指定正確的值，並使用支援的參數組合。例如，若要在置放群組中啟動執行個體，您必須指定支援的執行個體類型。
- 標籤 – 您可以為啟動範本建立標籤，但無法為啟動範本版本建立標籤。
- 固定 – 啟動範本是不可變的。若要修改啟動範本，您必須建立新版本的啟動範本。
- 版本編號 – 啟動範本版本會依其建立順序進行編號。當您建立啟動範本版本時，無法自行指定版本號碼。

Amazon EC2 啟動範本的必要 IAM 許可

您可使用 IAM 許可來控制使用者是否可以列出、檢視、建立或是刪除啟動範本或啟動範本版本。

> **Important**
>
> 使用者建立啟動範本或啟動範本版本時，您無法使用資源層級許可來限制其能夠在啟動範本中指定的資源。因此，請確定只有受信任的管理員才能獲得建立啟動範本和啟動範本版本的許可。

您必須向任何使用啟動範本的使用者授予所需的許可，以建立及存取啟動範本中指定的資源。例如：

- 若要從共用私有 Amazon Machine Image (AMI) 啟動執行個體，啟動範本使用者必須具有 AMI 的啟動許可。
- 若要使用現有快照的標籤建立 EBS 磁碟區，啟動範本使用者必須具有快照的讀取存取權，以及建立和標記磁碟區的許可。

目錄

- ec2:CreateLaunchTemplate
- ec2:DescribeLaunchTemplates
- ec2:DescribeLaunchTemplateVersions
- ec2:DeleteLaunchTemplate
- 控制版本控制許可
- 控制啟動範本上標籤的存取權

ec2:CreateLaunchTemplate

若要在主控台中建立啟動範本或使用 API，主體必須具有 IAM 政策中的 `ec2:CreateLaunchTemplate` 許可。盡可能使用標籤來協助控制帳戶中啟動範本的存取權。

例如，下列 IAM 政策陳述式僅在範本使用指定標籤 (*purpose*=*testing*) 時才授予主體建立啟動範本的許可。

```
{
    "Sid": "IAMPolicyForCreatingTaggedLaunchTemplates",
    "Action": "ec2:CreateLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

建立啟動範本的主體可能需要一些相關許可，例如：

- ec2:CreateTags – 若要在 `CreateLaunchTemplate` 操作期間將標籤新增至啟動範本，`CreateLaunchTemplate` 發起人必須具有 IAM 政策中的 `ec2:CreateTags` 許可。
- ec2:RunInstances – 若要從其建立的啟動範本啟動 EC2 執行個體，主體也必須具有 IAM 政策中的 `ec2:RunInstances` 許可。

針對套用標籤的資源建立動作，使用者必須具有 `ec2:CreateTags` 許可。如下 IAM 政策陳述式使用 `ec2:CreateAction` 條件索引鍵允許使用者僅在 `CreateLaunchTemplate` 的內容中建立標籤。使用者無法為現有啟動範本或任何其他資源套用標籤。如需詳細資訊，請參閱在建立期間授予標籤 Amazon EC2 資源的許可。

```
{
    "Sid": "IAMPolicyForTaggingLaunchTemplatesOnCreation",
    "Action": "ec2:CreateTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateLaunchTemplate"
        }
```

```
    }
}
```

建立啟動範本的 IAM 使用者不會自動擁有使用其所建立啟動範本的許可。與任何其他主體一樣，啟動範本建立者必須透過 IAM 政策取得許可。如果 IAM 使用者想要從啟動範本啟動 EC2 執行個體，他們必須具有 ec2:RunInstances 許可。授予這些許可時，您可以指定使用者只能使用具有特定標籤或特定 ID 的啟動範本。您也可以控制 AMI 和其他資源，讓使用啟動範本的任何人都可以在啟動執行個體時參照和使用，方法是指定 RunInstances 呼叫的資源層級許可。如需範例政策，請參閱 啟動範本。

ec2:DescribeLaunchTemplates

若要列出並檢視帳戶中的啟動範本，主體必須具有 IAM 政策中的 ec2:DescribeLaunchTemplates 許可。因為 Describe 動作不支援資源層級許可，您必須在沒有條件的情況下指定它們，且政策中資源元素的值必須為 "*"。

例如，下列 IAM 政策陳述式會授予主體許可，以列出並檢視帳戶中的所有啟動範本。

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplates",
    "Action": "ec2:DescribeLaunchTemplates",
    "Effect": "Allow",
    "Resource": "*"
}
```

ec2:DescribeLaunchTemplateVersions

列出並檢視啟動範本的主體也應具有擷取組成啟動範本之整組屬性的 ec2:DescribeLaunchTemplateVersions 許可。

若要列出並檢視帳戶中的啟動範本版本，主體必須具有 IAM 政策中的 ec2:DescribeLaunchTemplateVersions 許可。因為 Describe 動作不支援資源層級許可，您必須在沒有條件的情況下指定它們，且政策中資源元素的值必須為 "*"。

例如，下列 IAM 政策陳述式會授予主體許可，以列出並檢視帳戶中的所有啟動範本。

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplateVersions",
    "Effect": "Allow",
    "Action": "ec2:DescribeLaunchTemplateVersions",
    "Resource": "*"
```

```
}
```

ec2:DeleteLaunchTemplate

> **Important**
>
> 在授予主體刪除資源的許可時，您應該小心謹慎。刪除啟動範本可能會導致依賴啟動範本的 AWS 資源失敗。

若要刪除啟動範本，主體必須具有 IAM 政策中的 `ec2:DeleteLaunchTemplate` 許可。盡可能使用以標籤為基礎的條件索引鍵以限制許可。

例如，下列 IAM 政策陳述式僅在範本具有指定標籤 (*purpose*=*testing*) 時，才授予主體刪除啟動範本的許可。

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplates",
    "Action": "ec2:DeleteLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

或者，您可以使用 ARN 來辨識 IAM 政策適用的啟動範本。

啟動範本具有下列 ARN。

```
"Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/lt-09477bcd97b0d310e"
```

您可以透過將其包含在清單中來指定多個 ARN，也可以指定 `"*"` 的 `Resource` 值 (沒有 `Condition` 元素)，以允許主體者刪除帳戶中的任何啟動範本。

控制版本控制許可

對於受信任的管理員，您可以使用類似下列範例的 IAM 政策，授予建立和刪除啟動範本版本的存取權，以及變更啟動範本預設版本的存取權。

Important

授與主體建立啟動範本版本或修改啟動範本的許可時，請務必小心謹慎。

- 當您建立啟動範本版本時，您會影響任何允許 Amazon EC2 以 `Latest` 版本代表您啟動執行個體的 AWS 資源。
- 當您修改啟動範本時，可以變更哪個版本是 `Default`，因此會影響允許 Amazon EC2 使用此修改版本代表您啟動執行個體的任何 AWS 資源。

您也需要謹慎處理與 `Latest` 或 `Default` 啟動範本版本互動 AWS 的資源，例如 EC2 Fleet 和 Spot Fleet。當針對 `Latest` 或 `Default` 使用不同的啟動範本版本時，Amazon EC2 在啟動新執行個體以滿足機群的目標容量時，不會重新檢查完成動作所需的使用者許可，因此沒有使用者與 AWS 資源的互動。透過授與使用者呼叫 `CreateLaunchTemplateVersion` 和 `ModifyLaunchTemplate` API 的許可，如果使用者將機群指向含有執行個體設定檔 (IAM 角色的容器) 的不同啟動範本版本，則該使用者實際上也會獲得 `iam:PassRole` 許可。這表示使用者可能會更新啟動範本，以將 IAM 角色傳遞給執行個體 (即使執行個體沒有 `iam:PassRole` 許可)。在將許可授予可建立和管理啟動範本版本的使用者時保持謹慎，即可妥善管理此風險。

ec2:CreateLaunchTemplateVersion

若要建立啟動範本的新版本，主體必須具有 IAM 政策中啟動範本的 `ec2:CreateLaunchTemplateVersion` 許可。

例如，下列 IAM 政策陳述式僅在版本使用指定標籤 (*environment*=*production*) 時才授予主體建立啟動範本版本的許可。或者，您可以指定一或多個啟動範本 ARN，也可以指定 "*" 的 `Resource` 值 (沒有 `Condition` 元素)，以允許主體在帳戶中建立任何啟動範本的版本。

```
{
    "Sid": "IAMPolicyForCreatingLaunchTemplateVersions",
    "Action": "ec2:CreateLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
```

```
}
```

ec2:DeleteLaunchTemplateVersion

> **Important**
>
> 一如往常，在授予委託人刪除資源的許可時，您應該小心謹慎。刪除啟動範本版本可能會導致依賴啟動範本版本的 AWS 資源失敗。

若要刪除啟動範本版本，主體必須具有 IAM 政策中啟動範本的 `ec2:DeleteLaunchTemplateVersion` 許可。

例如，下列 IAM 政策陳述式僅在版本使用指定標籤 (*environment*=*production*) 時才授予主體刪除啟動範本版本的許可。或者，您可以指定一或多個啟動範本 ARN，也可以指定 "*" 的 Resource 值 (沒有 Condition 元素)，以允許主體在帳戶中刪除任何啟動範本的版本。

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplateVersions",
    "Action": "ec2:DeleteLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:ModifyLaunchTemplate

若要變更與啟動範本相關聯的 Default 版本，主體必須具有IAM 政策中啟動範本的 `ec2:ModifyLaunchTemplate` 許可。

例如，下列 IAM 政策陳述式僅在啟動範本使用指定標籤 (*environment*=*production*) 時才授予主體修改啟動範本的許可。或者，您可以指定一或多個啟動範本 ARN，也可以指定 "*" 的 Resource 值 (沒有 Condition 元素)，以允許主體在帳戶中修改任何啟動範本。

```
{
```

```
    "Sid": "IAMPolicyForModifyingLaunchTemplates",
    "Action": "ec2:ModifyLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

控制啟動範本上標籤的存取權

當資源為啟動範本時，您可以使用條件索引鍵來限制標記許可。例如，下列 IAM 政策只允許從指定帳戶或區域中的啟動範本移除具有 *temporary* 索引鍵的標籤。

```
{
    "Sid": "IAMPolicyForDeletingTagsOnLaunchTemplates",
    "Action": "ec2:DeleteTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "aws:TagKeys": ["temporary"]
        }
    }
}
```

如需可用於控制可套用至 Amazon EC2 資源之標籤索引鍵和值的條件索引鍵的詳細資訊，請參閱 控制對特定標籤的存取。

使用 Amazon EC2 啟動範本，來控制啟動 Amazon EC2 執行個體

您可以指定使用者僅在使用啟動範本時才能啟動執行個體，以及指定使用者只能使用特定的啟動範本，藉此控制 Amazon EC2 執行個體的組態。您也可以控制哪些人員可以建立、修改、描述與刪除啟動範本和啟動範本版本。

使用啟動範本控制啟動參數

啟動範本可以包含在啟動時設定執行個體的所有或部分參數。不過，當您使用啟動範本啟動執行個體時，可以覆寫啟動範本中指定的參數。或者，您可以指定啟動範本中沒有的其他參數。

> **Note**
>
> 您無法在啟動期間移除啟動範本參數 (例如，您無法為參數指定 null 值)。若要移除參數，請在沒有參數的情況下建立新版本的啟動範本，並使用該版本來啟動執行個體。

若要啟動執行個體，使用者必須具備使用 `ec2:RunInstances` 動作的許可。使用者也必須具備許可，才能建立或使用由該執行個體所建立或與其相關聯的資源。您可以使用 `ec2:RunInstances` 動作上的資源層級許可，以控制使用者可指定的啟動參數。或者，您可以將使用啟動範本啟動執行個體的許可授予使用者。這可讓您在啟動範本中 (而非在 IAM 政策中) 管理啟動參數，而且可讓您使用啟動範本做為啟動執行個體的授權媒介。例如，您可以指定使用者只能使用啟動範本來啟動執行個體，而且使用者只能使用特定的啟動範本。您也可以控制使用者能在啟動範本中覆寫的啟動參數。如需範例政策，請參閱 啟動範本。

控制啟動範本的使用

根據預設， 使用者不具使用啟動範本的許可。您可以建立使用者政策，授予使用者建立、修改、描述與刪除啟動範本和啟動範本版本的許可。您也可以將資源層級許可套用至一些啟動範本動作，藉以控制使用者將特定資源用於那些動作的能力。如需詳細資訊，請參閱下列範例政策：範例：使用啟動範本。

將使用 `ec2:CreateLaunchTemplate` 和 `ec2:CreateLaunchTemplateVersion` 動作的許可授予使用者時請小心。您無法使用資源層級許可來控制使用者在啟動範本中能指定的資源。若要限制用來啟動執行個體的資源，請確定您只將建立啟動範本和啟動範本版本的許可授予合適的管理員。

搭配 EC2 機群或 Spot 機群使用啟動範本時的重要安全考量

> **Note**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

若要使用啟動範本，您必須授予使用者許可，以便建立、修改、描述和刪除啟動範本及啟動範本版本。透過控制對 `ec2:CreateLaunchTemplate` 和 `ec2:CreateLaunchTemplateVersion` 動作的存取權，您可以控制誰可以建立啟動範本和啟動範本版本。您也可以透過控制 `ec2:ModifyLaunchTemplate` 動作的存取權來控制可修改啟動範本的對象。

> **Important**
>
> 如果 EC2 機群或 Spot 機群設定為使用「最新」或「預設」啟動範本版本，則機群不會知道「最新」或「預設」稍後是否會變更為不同的啟動範本版本。當「最新」或「預設」使用不同的啟動範本版本時，Amazon EC2 在啟動新執行個體以滿足機群的目標容量時，不會重新檢查完成動作所需的許可。將許可授與可建立和管理啟動範本版本的使用者 (尤其是允許使用者變更「預設」啟動範本版本的 `ec2:ModifyLaunchTemplate` 動作) 時，請務必考量到這點。

透過授與使用者針對啟動範本 API 使用 EC2 動作的許可，如果使用者建立或更新 EC2 機群或 Spot 機群以指向含有執行個體設定檔 (IAM 角色的容器) 的不同啟動範本版本，則該使用者實際上也會獲得 `iam:PassRole` 許可。這表示使用者可能會更新啟動範本，以將 IAM 角色傳遞給執行個體 (即使執行個體沒有 `iam:PassRole` 許可)。如需詳細資訊和範例 IAM 政策，請參閱 「IAM 使用者指南」 中的 利用 IAM 角色來授予許可給在 Amazon EC2 執行個體上執行的應用程式。

如需詳細資訊，請參閱控制啟動範本的使用及範例：使用啟動範本。

建立 Amazon EC2 啟動範本

您可以為執行個體組態參數指定自己的值，或是從現有的啟動範本或 Amazon EC2 執行個體取得值，藉此建立 Amazon EC2 啟動範本。

您不需要為啟動範本中的每個參數指定值；您只需指定一個執行個體組態參數即可建立啟動範本。若要表示您選擇不指定的參數，在使用主控台時，請選取不要包含在啟動範本中。使用命令列工具時，請勿包含參數，以表示您要選取不要將參數包含在啟動範本中。

如果您想要在啟動範本中指定 AMI，可以選取 AMI，或指定 Systems Manager 參數，以在執行個體啟動時指向 AMI。

使用啟動範本啟動執行個體時，會將啟動範本中所指定的值用來設定對應的執行個體參數。如果未在啟動範本中指定值，則會使用對應執行個體參數的預設值。

任務

- 指定參數來建立啟動範本
- 從現有啟動範本建立啟動範本
- 從執行個體建立啟動範本
- 使用 Systems Manager 參數而非 AMI ID

指定參數來建立啟動範本

若要建立啟動範本，您必須指定啟動範本名稱和至少一個執行個體組態參數。

如需每個參數的描述，請參閱 Amazon EC2 執行個體組態參數的參考。

Console

建立啟動範本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本) 接著選擇 Create launch template (建立啟動範本)。
3. 在啟動範本名稱和描述下，執行下列動作：
   a. 針對 Launch template name (啟動範本名稱)，輸入啟動範本的描述性名稱。
   b. 在 Template version description (範本版本描述) 中，提供此啟動範本版本的簡短描述。
   c. 若要在建立時標記啟動範本，請展開範本標籤、選擇新增標籤，然後輸入標籤鍵值組。為每個要新增的其他標籤重新選擇新增新的標籤。

   > **Note**
   > 要標記啟動執行個體時建立的資源，您必須在 Resource tags (資源標籤) 下指定標籤。如需詳細資訊，請參閱程序中的步驟 9。

4. 您可以在應用程式和作業系統映像 (Amazon Machine Image) 下，保持選取不要包含在啟動範本中，或選擇執行個體的作業系統 (OS)，然後選擇 AMI。或者，您可以指定 Systems Manager 參數，而非指定 AMI。如需詳細資訊，請參閱使用 Systems Manager 參數而非 AMI ID。

   AMI 是一種包含啟動執行個體所需作業系統和軟體的範本。
5. 對於執行個體類型，您可以保持選取不要包含在啟動範本中，或是選取執行個體類型，或者指定執行個體屬性，並讓 Amazon EC2 利用這些屬性識別執行個體類型。

   > **Note**
   > 只有在 Auto Scaling 群組、EC2 機群和 Spot 機群使用啟動範本啟動執行個體時，才支援指定執行個體屬性。如需詳細資訊，請參閱使用參數型執行個體類型選項建立混合執行個體群組和 指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型。

如果您打算在啟動執行個體精靈中使用啟動範本，或與 RunInstances API 搭配使用，則您無法指定執行個體類型屬性。

執行個體類型會決定硬體組態 (CPU、記憶體、儲存體和聯網容量)，以及執行個體所使用主機電腦的大小。

如果您不確定要選擇哪個執行個體類型，可以執行下列動作：

- 選擇比較執行個體類型，依下列屬性比較不同的執行個體類型：vCPU 數目、架構、記憶體數量 (GiB)、儲存數量 (GB)、儲存類型，以及網路效能。
- 選擇取得建議，以從 EC2 執行個體類型搜尋工具取得執行個體類型的指導和建議。如需詳細資訊，請參閱從 EC2 執行個體類型搜尋工具取得建議。

> **Note**
>
> 視您建立帳戶的時間而定，您可能有資格使用免費方案下的 Amazon EC2。
> 如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。如果某個執行個體類型可用於免費方案，則標記為符合免費方案條件。
> 如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。
> 如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。

6. 在金鑰對 (登入) 下，針對金鑰對名稱，請保留選取不要包含在啟動範本中，或選擇現有的金鑰對，或者建立新的金鑰對。
7. 在網路設定下，您可以保持選取不要包含在啟動範本中，也可以指定各種網路設定的值。
8. 在設定儲存下，如果您在啟動範本中指定 AMI，則 AMI 會包含一或多個儲存磁碟區，包含根磁碟區 (Volume 1 (AMI Root))。您可以選擇指定要連接至執行個體的其他磁碟區。若要新增磁碟區，請選擇 Add new volume (新增磁碟區)。
9. 在資源標籤下，若要標記啟動執行個體時所建立的資源，請選擇新增標籤，然後輸入標籤鍵值組。針對 Resource types (資源類型)，請指定建立時要標記的資源。您可以為所有資源指定相

同的標籤，或者為不同的資源指定不同的標籤。為每個要新增的其他標籤重新選擇 Add tag (新增標籤)。

您可以針對使用啟動範本時建立的以下資源指定標籤：

- 執行個體
- 磁碟區
- Elastic Graphics
- Spot 執行個體請求
- 網路介面

> **Note**
>
> 若要標記啟動範本本身，您必須在 Template tags (範本標籤) 下指定標籤。如需詳細資訊，請參閱程序中的步驟 3。

10. 針對進階詳細資訊，展開此區段來檢視欄位，並選擇指定執行個體的任何其他參數。
11. 使用摘要面板來檢閱啟動範本組態。您可以選擇區段的連結，然後進行任何必要的變更，以導覽至任何區段。
12. 當您準備好建立啟動範本時，請選擇 Create launch template (建立啟動範本)。

AWS CLI

建立啟動範本

使用 create-launch-template 命令。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --version-description WebVersion1 \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

以下是範例 JSON，會指定執行個體組態的啟動範本資料。將 JSON 儲存到檔案，並將其包含在 `--launch-template-data` 參數中，如範例命令所示。

```
{
```

```
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":2
    }
}
```

以下為範例輸出。

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

PowerShell

建立啟動範本

使用 New-EC2LaunchTemplate cmdlet。

```
$launchTemplateData = [Amazon.EC2.Model.RequestLaunchTemplateData]@{
    ImageId = 'ami-0abcdef1234567890'
    InstanceType = 'r5.4xlarge'
```

```
    NetworkInterfaces = @(

 [Amazon.EC2.Model.LaunchTemplateInstanceNetworkInterfaceSpecificationRequest]@{
            AssociatePublicIpAddress = $true
            DeviceIndex = 0
            Ipv6AddressCount = 1
            SubnetId = 'subnet-0abcdef1234567890'
        }
    )
    TagSpecifications = @(
        [Amazon.EC2.Model.LaunchTemplateTagSpecificationRequest]@{
            ResourceType = 'instance'
            Tags = [Amazon.EC2.Model.Tag]@{
                Key = 'Name'
                Value = 'webserver'
            }
        }
    )
    CpuOptions = [Amazon.EC2.Model.LaunchTemplateCpuOptionsRequest]@{
        CoreCount = 4
        ThreadsPerCore = 2
    }
}
$tagSpecificationData = [Amazon.EC2.Model.TagSpecification]@{
    ResourceType = 'launch-template'
    Tags = [Amazon.EC2.Model.Tag]@{
        Key = 'purpose'
        Value = 'production'
    }
}
New-EC2LaunchTemplate -LaunchTemplateName 'TemplateForWebServer' `
    -VersionDescription 'WebVersion1' `
    -LaunchTemplateData $launchTemplateData `
    -TagSpecification $tagSpecificationData
```

以下為範例輸出。

```
CreatedBy            : arn:aws:iam::123456789012:root
CreateTime           : 9/19/2023 16:57:55
DefaultVersionNumber : 1
LatestVersionNumber  : 1
LaunchTemplateId     : lt-01238c059eEXAMPLE
LaunchTemplateName   : TemplateForWebServer
```

```
Tags                : {purpose}
```

從現有啟動範本建立啟動範本

您可以複製現有啟動範本，然後調整參數以建立新版啟動範本。不過，您只能在使用 Amazon EC2 主控台時執行這項操作。 AWS CLI 不支援複製範本。如需每個參數的描述，請參閱 Amazon EC2 執行個體組態參數的參考。

Console

若要從現有啟動範本建立啟動範本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本) 接著選擇 Create launch template (建立啟動範本)。
3. 針對 Launch template name (啟動範本名稱)，輸入啟動範本的描述性名稱。
4. 在 Template version description (範本版本描述) 中，提供此啟動範本版本的簡短描述。
5. 若要在建立時標記啟動範本，請展開範本標籤、選擇新增標籤，然後輸入標籤鍵值組。
6. 展開來源範本，並針對啟動範本名稱，選擇要以新啟動範本為基礎的啟動範本。
7. 針對 Source template version (來源範本版本)，選擇要做為新啟動範本基礎的啟動範本版本。
8. 視需要調整任何啟動參數，然後選擇 Create launch template (建立啟動範本)。

從執行個體建立啟動範本

您可以複製現有 Amazon EC2 執行個體的參數，然後調整參數來建立啟動範本。如需每個參數的描述，請參閱 Amazon EC2 執行個體組態參數的參考。

Console

若要從執行個體建立啟動範本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選擇動作、映像和範本、從執行個體建立範本。
4. 提供名稱、描述和標籤，並視需要調整啟動參數。

> **Note**
>
> 當您從執行個體建立啟動範本時，範本不包含執行個體的網路介面 ID 和 IP 地址。

5. 選擇 Create launch template (建立啟動範本)。

AWS CLI

您可以使用 從現有執行個體 AWS CLI 建立啟動範本，方法是先從執行個體取得啟動範本資料，然後使用啟動範本資料建立啟動範本。

若要從執行個體啟動範本資料

- 使用 get-launch-template-data 命令，並指定執行個體 ID。您能以輸出做為基礎，用以建立新啟動範本或啟動範本版本。根據預設，輸出包括最上層 `LaunchTemplateData` 物件，而此物件無法於啟動範本資料中指定。使用 `--query` 選項排除此物件。

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData"
```

下列為範例輸出。

```
    {
        "Monitoring": {},
        "ImageId": "ami-8c1be5f6",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "DeleteOnTermination": true
                }
            }
        ],
        "EbsOptimized": false,
        "Placement": {
            "Tenancy": "default",
            "GroupName": "",
            "AvailabilityZone": "us-east-1a"
        },
```

```
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
            {
                "Description": "",
                "NetworkInterfaceId": "eni-35306abc",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateIpAddress": "10.0.0.72"
                    }
                ],
                "SubnetId": "subnet-7b16de0c",
                "Groups": [
                    "sg-7c227019"
                ],
                "Ipv6Addresses": [
                    {
                        "Ipv6Address": "2001:db8:1234:1a00::123"
                    }
                ],
                "PrivateIpAddress": "10.0.0.72"
            }
        ]
    }
```

例如，您可以將輸出直接寫入檔案：

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData" >> instance-data.json
```

使用啟動範本資料建立啟動範本

- 使用 create-launch-template 命令，使用上一個程序的輸出建立啟動範本。如需使用 建立啟動範本的詳細資訊 AWS CLI，請參閱 指定參數來建立啟動範本。

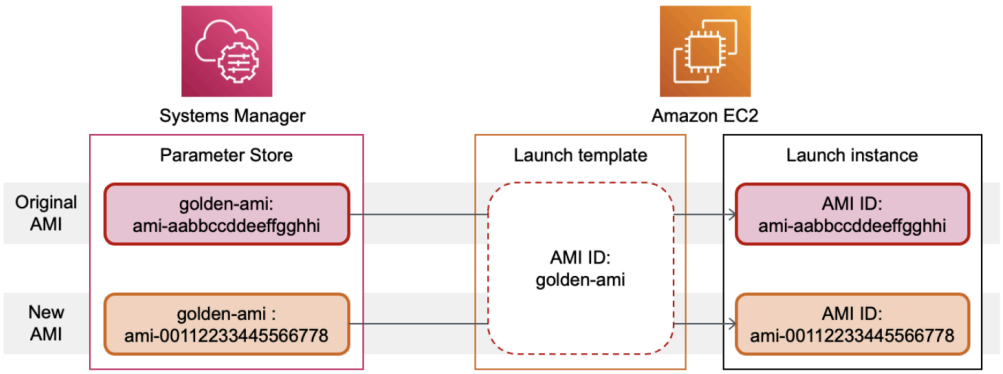
使用 Systems Manager 參數而非 AMI ID

您可以指定 AWS Systems Manager 參數，而非在啟動範本中指定 AMI ID。如果 AMI ID 變更，則您可以透過更新 Systems Manager 參數存放區中的 Systems Manager 參數，在一個位置更新 AMI ID。

參數也可以與其他 共用 AWS 帳戶。您可以將 AMI 參數集中存放在一個帳戶中並進行管理，並與需要參考這些參數的所有其他帳戶共用。透過使用 Systems Manager 參數，您可以在單一動作中更新所有啟動範本。

Systems Manager 參數是使用者定義的鍵值對，您可以在 AWS Systems Manager 參數存放區中建立。參數存放區提供一個中央位置來存放應用程式組態值。

在下圖中，golden-ami 參數首先映射到參數存放區中的原始 AMI ami-aabbccddeeffgghhi。在啟動範本中，AMI ID 的值為 golden-ami。使用此啟動範本啟動執行個體時，AMI ID 會解析為 ami-aabbccddeeffgghhi。稍後，會更新 AMI，從而產生新的 AMI ID。在參數存放區中，golden-ami 參數會映射至新的 ami-00112233445566778。啟動範本保持不變。使用此啟動範本啟動執行個體時，AMI ID 會解析為新的 ami-00112233445566778。



AMI ID 的 Systems Manager 參數格式

啟動範本要求使用者定義的 Systems Manager 參數在用於取代 AMI ID 時必須遵循下列格式：

- 參數類型：String
- 參數資料類型：aws:ec2:image - 這可確保參數存放區驗證您輸入的值是否為格式正確的 AMI ID 。

如需有關為 AMI ID 建立有效參數的詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的建立 Systems Manager 參數。

啟動範本中的 Systems Manager 參數格式

若要使用 Systems Manager 參數取代啟動範本中的 AMI ID，在啟動範本中指定參數時，必須使用下列其中一種格式：

若要參考公用參數：

- resolve:ssm:*public-parameter*

若要參考存放在相同帳戶中的參數：

- resolve:ssm:*parameter-name*
- resolve:ssm:*parameter-name*:*version-number* - 版本編號本身是預設標籤
- resolve:ssm:*parameter-name*:*label*

若要參考從另一個 AWS 帳戶共用的參數：

- resolve:ssm:*parameter-ARN*
- resolve:ssm:*parameter-ARN*:*version-number*
- resolve:ssm:*parameter-ARN*:*label*

參數版本

Systems Manager 參數是版本化資源。更新參數時，會建立新的連續參數版本。Systems Manager 支援參數標籤，您可以將其映射至特定參數版本。

例如，golden-ami 參數可以有三個版本：1、2 和 3。您可以建立映射至版本 2 的參數標籤 beta，以及映射至版本 3 的參數標籤 prod。

在啟動範本中，可以使用下列其中一種格式來指定 golden-ami 參數的版本 3：

- resolve:ssm:golden-ami:3
- resolve:ssm:golden-ami:prod

指定版本或標籤是可選操作。如果未指定版本或標籤，則會使用最新版本的參數。

在啟動範本中指定 Systems Manager 參數

建立啟動範本或新版本的啟動範本時，可以在啟動範本中指定 Systems Manager 參數，而非 AMI ID。

Console

在啟動範本中指定 Systems Manager 參數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Launch Templates (啟動範本) 接著選擇 Create launch template (建立啟動範本)。
3. 針對 Launch template name (啟動範本名稱)，輸入啟動範本的描述性名稱。
4. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，選擇 Browse more AMIs (瀏覽更多 AMI)。
5. 選擇搜尋列右側的箭頭按鈕，然後選擇指定自訂值/Systems Manager 參數。
6. 在指定自訂值或 Systems Manager 參數對話方塊中，執行下列操作：

   a. 對於 AMI ID 或 Systems Manager 參數字串，請使用下列其中一種格式輸入 Systems Manager 參數名稱：

      若要參考公用參數：

      - **resolve:ssm:*public-parameter***

      若要參考存放在相同帳戶中的參數：

      - **resolve:ssm:*parameter-name***
      - **resolve:ssm:*parameter-name*:*version-number***
      - **resolve:ssm:*parameter-name*:*label***

      若要參考從另一個 AWS 帳戶共用的參數：

      - **resolve:ssm:*parameter-ARN***
      - **resolve:ssm:*parameter-ARN*:*version-number***
      - **resolve:ssm:*parameter-ARN*:*label***

   b. 選擇儲存。
7. 視需要指定任何其他啟動範本參數，然後選擇建立啟動範本。

如需詳細資訊，請參閱指定參數來建立啟動範本。

AWS CLI

在啟動範本中指定 Systems Manager 參數

- 使用 create-launch-template 命令，建立啟動範本。若要指定要使用的 AMI，請使用下列其中一種格式輸入 Systems Manager 參數名稱：

若要參考公用參數：

- **resolve:ssm:*public-parameter***

若要參考存放在相同帳戶中的參數：

- **resolve:ssm:*parameter-name***
- **resolve:ssm:*parameter-name*:*version-number***
- **resolve:ssm:*parameter-name*:*label***

若要參考從另一個 AWS 帳戶共用的參數：

- **resolve:ssm:*parameter-ARN***
- **resolve:ssm:*parameter-ARN*:*version-number***
- **resolve:ssm:*parameter-ARN*:*label***

以下範例建立的啟動範本會指定下列項目：

- 啟動範本的名稱 (*TemplateForWebServer*)
- 啟動範本的標籤 (*purpose*=*production*)
- 在 JSON 檔案中指定的執行個體組態的資料：
  - 要使用的 AMI (resolve:ssm:*golden-ami*)
  - 要啟動的執行個體類型 (*m5.4xlarge*)
  - 執行個體的標籤 (*Name*=*webserver*)

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

以下是一個範例 JSON 檔案，其中包含執行個體組態的啟動範本資料。ImageId 的值為 Systems Manager 參數名稱，以所需格式 resolve:ssm:*golden-ami* 輸入。

```
{"LaunchTemplateData": {
    "ImageId": "resolve:ssm:golden-ami",
    "InstanceType": "m5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }]
  }
}
```

確認啟動範本取得正確的 AMI ID

將 Systems Manager 參數解析為實際 AMI ID

使用 describe-launch-template-versions 命令，並包含 `--resolve-alias` 參數。

```
aws ec2 describe-launch-template-versions \
    --launch-template-name my-launch-template \
    --versions $Default \
    --resolve-alias
```

回應包含 `ImageId` 的 AMI ID。在此範例中，使用此啟動範本啟動執行個體時，AMI ID 會解析為 `ami-0ac394d6a3example`。

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t3.micro",
            }
        }
```

```
    ]
}
```

相關資源

如需關於使用 Systems Manager 參數的詳細資訊，請參閱 Systems Manager 文件中的下列參考資料。

- 如需如何查詢 Amazon EC2 所支援 AMI 公用參數的詳細資訊，請參閱呼叫 AMI 公用參數。
- 如需與其他 AWS 帳戶或透過 共用參數的詳細資訊 AWS Organizations，請參閱使用共用參數。
- 如需監控參數是否已成功建立的相關資訊，請參閱「Amazon Machine Image ID 的原生參數支援」https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-ec2-aliases.html。

限制

- 只有 `instant` 類型的 EC2 機群支援使用具有指定 Systems Manager 參數來取代 AMI ID 的啟動範本。
- 目前，`maintain` 和 `request` 類型的 EC2 機群以及 Spot 機群不支援使用具有指定 Systems Manager 參數來取代 AMI ID 的啟動範本。對於 `maintain` 和 `request` 類型的 EC2 機群以及 Spot 機群，如果您在啟動範本中指定 AMI，則必須指定 AMI ID。
- 如果您在 EC2 機群中使用屬性型執行個體選擇，則無法指定 Systems Manager 參數來取代 AMI ID。您在使用屬性型執行個體選擇時，必須指定 AMI ID。
- Amazon EC2 Auto Scaling 會提供其他限制。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的在啟動範本中使用 AWS Systems Manager 參數而非 AMI IDs。

修改啟動範本 (管理啟動範本版本)

啟動範本是不可變的；啟動範本建立後便無法再修改。但是，您可以建立新版本的啟動範本，來包含您需要的任何變更。

您可以建立不同版本的啟動範本、設定預設版本、說明啟動範本版本，以及刪除您不再需要的版本。

任務

- 建立啟動範本版本
- 設定預設啟動範本版本
- 描述啟動範本版本

建立啟動範本版本

當您建立啟動範本版本時，可以指定新的啟動參數，或使用現有版本做為新版本的基礎。如需每個參數的描述，請參閱 Amazon EC2 執行個體組態參數的參考。

Console

若要建立啟動範本版本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本)。
3. 選擇啟動範本，然後選擇動作、修改範本 （ 建立新版本 ）。
4. 在 Template version description (範本版本描述) 中，輸入此啟動範本版本的描述。
5. (選取性) 展開 Source template (來源範本)，並選取啟動範本的版本，以作為新啟動範本版本的基礎。新啟動範本版本會繼承此啟動範本版本的啟動參數。
6. 視需要修改啟動參數。
7. 選擇 Create launch template (建立啟動範本)。

AWS CLI

建立啟動範本版本

使用 create-launch-template-version 命令。您可以指定要做為新版本基礎的來源版本。新版本會繼承此版本的啟動參數，但您可以使用 `--launch-template-data` 來覆寫參數。下列範例會建立以第 1 版啟動範本為基礎的新版本，並指定不同的 AMI ID。

```
aws ec2 create-launch-template-version \
    --launch-template-id lt-0abcd290751193123 \
    --version-description WebVersion2 \
    --source-version 1 \
    --launch-template-data "ImageId=ami-0abcdef1234567890"
```

PowerShell

建立啟動範本版本

使用 New-EC2LaunchTemplateVersion Cmdlet。您可以指定要做為新版本基礎的來源版本。新版本會繼承此版本的啟動參數，但您可以使用 `LaunchTemplateData` 來覆寫參數。下列範例會建立以第 1 版啟動範本為基礎的新版本，並指定不同的 AMI ID。

```
New-EC2LaunchTemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -VersionDescription WebVersion2 `
    -SourceVersion 1 `
    -LaunchTemplateData (
        New-Object `
            -TypeName Amazon.EC2.Model.RequestLaunchTemplateData `
            -Property @{ImageId = 'ami-0abcdef1234567890'}
    )
```

設定預設啟動範本版本

您可以設定啟動範本的預設版本。當您從啟動範本啟動執行個體而且未指定版本時，會使用預設版本的參數來啟動執行個體。

Console

若要設定預設啟動範本版本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本)。
3. 選擇啟動範本，然後選擇動作、設定預設版本。
4. 對於 Template version (範本版本)，請選取要設為預設版本的版本號碼，然後選取 Set as default version (設為預設版本)。

AWS CLI

若要設定預設啟動範本版本

使用 modify-launch-template 命令。

```
aws ec2 modify-launch-template \
    --launch-template-id lt-0abcd290751193123 \
    --default-version 2
```

PowerShell

若要設定預設啟動範本版本

使用 Edit-EC2LaunchTemplate cmdlet。

```
Edit-EC2LaunchTemplate `
    -LaunchTemplateId lt-0abcd290751193123 `
    -DefaultVersion 2
```

描述啟動範本版本

您可以使用主控台，檢視所選啟動範本的所有版本，或取得其最新或預設版本符合特定版本號碼之啟動範本的清單。您可以使用 AWS CLI描述指定啟動範本的所有版本、個別版本或一系列版本。您也可以說明帳戶中所有啟動範本的所有最新版本或所有預設版本。

Console

若要描述啟動範本版本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本)。
3. 您可以檢視特定啟動範本版本，或取得其最新或預設版本符合特定版本號碼之啟動範本的清單。
   - 若要檢視啟動範本的版本：選擇啟動範本。在版本索引標籤的版本中，選擇版本以檢視其詳細資訊。
   - 若要取得其最新版本符合特定版本號碼之所有啟動範本的清單：請從搜尋列中選擇 Latest version (最新版本)，然後選擇版本號碼。
   - 若要取得其預設版本符合特定版本號碼之所有啟動範本的清單：請從搜尋列中選擇 Default version (預設版本)，然後選擇版本號碼。

AWS CLI

若要描述啟動範本版本

使用 describe-launch-template-versions 命令，並指定版本號碼。在下列範例中，指定的是第 *1* 版和第 *3* 版。

```
aws ec2 describe-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1 3
```

描述帳戶中最新和預設啟動範本版本

使用 describe-launch-template-versions 命令，並指定 $Latest、$Default 或兩者。您必須省略呼叫中的啟動範本 ID 和名稱。您無法指定版本號碼。

```
aws ec2 describe-launch-template-versions \
    --versions "$Latest,$Default"
```

PowerShell

若要描述啟動範本版本

使用 Get-EC2TemplateVersion Cmdlet，並指定版本編號。在下列範例中，指定的是第 *1* 版和第 *3* 版。

```
Get-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1,3
```

描述帳戶中最新和預設啟動範本版本

使用 Get-EC2TemplateVersion Cmdlet，並指定 $Latest 和/或 $Default。您必須省略呼叫中的啟動範本 ID 和名稱。您無法指定版本號碼。

```
Get-EC2TemplateVersion `
    -Version '$Latest','$Default'
```

刪除啟動範本或啟動範本版本

如果您不再需要啟動範本，即可將其刪除。刪除啟動範本會刪除它的所有版本。如果您只想要刪除特定版本的啟動範本，可以在保留其他版本啟動範本的情況下執行此操作。

刪除啟動範本或啟動範本版本，並不會影響您從啟動範本啟動的任何執行個體。

刪除啟動範本以及其所有版本

如果您不再需要啟動範本，包括其所有版本，可以刪除啟動範本。刪除啟動範本會刪除它的所有版本。

Console

刪除啟動範本及其所有版本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本)。
3. 選擇啟動範本，然後選擇動作、刪除範本。
4. 輸入 **Delete** 以確認刪除，然後選擇 Delete (刪除)。

AWS CLI

刪除啟動範本及其所有版本

使用 delete-launch-template 命令，並指定啟動範本。

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

PowerShell

刪除啟動範本及其所有版本

使用 Remove-EC2LaunchTemplate (AWS Tools for PowerShell) 命令，並指定啟動範本。如果省略 `-Force`，則 PowerShell 會提示進行確認。

```
Remove-EC2LaunchTemplate -LaunchTemplateId lt-0123456789example -Force
```

刪除啟動範本版本

如果您不再需要啟動範本版本，可以將其刪除。

考量事項

- 版本號碼在刪除之後就無法取代。
- 您無法刪除啟動範本的預設版本；您必須先指派不同的版本作為預設值。如果預設版本是啟動範本的唯一版本，則須刪除整個啟動範本。
- 使用主控台時，您一次可以刪除一個啟動範本版本。使用 時 AWS CLI，您可以在單一請求中刪除最多 200 個啟動範本版本。若要在單個請求中刪除超過 200 個版本，您可以刪除啟動範本，而這也會刪除啟動範本的所有版本。

Console

若要刪除啟動範本版本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Launch Templates (啟動範本)。
3. 選擇啟動範本，然後選擇動作、刪除範本版本。
4. 選擇要刪除的版本，然後選擇刪除。

AWS CLI

刪除啟動範本版本

使用 delete-launch-template-versions 命令，並指定要刪除的版本號碼。您在單個請求中最多可刪除 200 個啟動範本版本。

```
aws ec2 delete-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1
```

PowerShell

刪除啟動範本版本

使用 Remove-EC2TemplateVersion cmdlet，並指定要刪除的版本編號。您在單個請求中最多可刪除 200 個啟動範本版本。

```
Remove-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1
```

啟動 Amazon EC2 執行個體

執行個體是 AWS 雲端中的虛擬伺服器。您從 Amazon Machine Image (AMI) 啟動執行個體。AMI 可為您的執行個體提供作業系統、應用程式伺服器和應用程式。

建立 時 AWS 帳戶，您可以使用 免費開始使用 Amazon EC2AWS 免費方案。您的免費方案優惠取決於您建立 AWS 帳戶的時間。如果您在 2025 年 7 月 15 日前建立 AWS 帳戶 ，時間少於 12 個月，您可以使用免費方案來啟動並免費使用 t2.micro 執行個體 (在無法使用

t2.micro 的區域中，您可以根據免費方案使用 t3.micro 執行個體)。執行個體執行時，即使執行個體保持閒置，也會針對執行個體或計入免費方案限制的使用量產生費用。如需詳細資訊，請參閱 Amazon EC2 定價。若您在 2025 年 7 月 15 日或之後建立 AWS 帳戶 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的抵用金用完為止。如需詳細資訊，請參閱 2025 年 7 月 15 日之前和之後的免費方案優惠。

當您啟動執行個體時，您可以在與下列其中一個資源相關聯的子網中啟動執行個體：

- 可用區域 – 此選項是預設值。
- Local Zone – 若要啟動 Local Zone 中的執行個體，您必須選擇加入 Local Zone，然後在區域中建立子網路。如需詳細資訊，請參閱 Local Zones 入門。
- Wavelength 區域 – 若要啟動 Wavelength 區域中的執行個體，您必須選擇加入 Wavelength 區域，然後在區域中建立子網路。如需有關如何在 Wavelength 區域中啟動執行個體的資訊，請參閱 AWS Wavelength入門。
- Outpost – 若要啟動 Outpost 中的執行個體，您必須建立 Outpost。如需如何建立 Outpost 的資訊，請參閱開始使用 AWS Outposts。

啟動您的執行個體之後，您便可以連線及使用它。開始時，執行個體的狀態為 pending。當執行個體的狀態為 running 時，執行個體便會開始開機。在您連線到執行個體前可能需要短暫的時間。請注意，裸機執行個體類型可能需要更長的時間來啟動。

視您計劃連線到執行個體的方式而定，您可能需要在啟動執行個體時進行特定組態。這些組態可能包括指定特定流量的傳入安全群組規則，或是關聯執行個體設定檔角色。如需關於可用來連線的連線方法及其需求的詳細資訊，請參閱 連線至 EC2 執行個體。

執行個體會獲得公有 DNS 名稱，您可以使用該名稱，從網際網路聯絡執行個體。執行個體也會獲得私有 DNS 名稱，讓其他位於相同 VPC 中的執行個體使用該名稱來聯絡執行個體。

當您完成執行個體時，為了避免產生不必要的成本，請務必將執行個體終止。如需詳細資訊，請參閱終止 Amazon EC2 執行個體。

如果您需要啟動大量執行個體、使用多個執行個體類型，或使用多個購買選項，例如隨需執行個體、預留執行個體和 Spot 執行個體，請考慮使用 EC2 機群。如需詳細資訊，請參閱EC2 機群 和 Spot 機群。

如果您想要自動化執行個體的生命週期，包括自動擴展、運作狀態檢查和取代運作狀態不佳的執行個體，請考慮使用 Amazon EC2 Auto Scaling。

以下是您可以啟動執行個體的一些方法。

Method	工具	文件
使用啟動執行個體精靈指定啟動參數。	Amazon EC2 主控台	使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體
建立啟動範本並從啟動範本啟動執行個體。	Amazon EC2 主控台	使用啟動範本啟動 EC2 執行個體
使用現有執行個體做為基礎。	Amazon EC2 主控台	使用現有執行個體的詳細資訊來啟動 EC2 執行個體
使用您從 AWS Marketplace購買的 AMI。	Amazon EC2 主控台	從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體
使用您指定的 AMI。	AWS CLI	在 中啟動、列出和刪除 Amazon EC2 執行個體 AWS CLI
使用您指定的 AMI。	AWS Tools for Windows PowerShell	使用 Windows PowerShell 啟動 Amazon EC2 執行個體
使用 EC2 機群跨不同的 EC2 執行個體類型和可用區域，及跨隨需執行個體、預留執行個體和 Spot 執行個體購買選項來佈建容量。	AWS CLI	EC2 機群 和 Spot 機群
使用 CloudFormation 範本來指定執行個體。	AWS CloudFormation	AWS CloudFormation 使用者指南中的 AWS::EC2::Instance
使用語言特定的 AWS SDK 來啟動執行個體。	AWS 開發套件	AWS 適用於 .NET 的 SDK AWS 適用於 C++ 的 SDK AWS 適用於 Go 的 SDK AWS 適用於 Java 的開發套件

Method	工具	文件
		AWS 適用於 JavaScript 的 SDK AWS 適用於 PHP V3 的 SDK AWS 適用於 Python 的 SDK AWS 適用於 Ruby V3 的 SDK

啟動 EC2 執行個體的教學課程

啟動和設定 Amazon EC2 執行個體的方法有很多種。您使用的方法和組態取決於您的特定使用案例。

下列教學課程可協助您了解如何啟動 EC2 執行個體。如果您是初次使用 Amazon EC2，我們建議您從第一個教學課程開始。教學課程一開始會向您介紹基本概念，並介紹更多組態選項，協助您產生基本概念。

目標	教學課程連結
啟動我的第一個 EC2 執行個體 了解如何使用 Amazon EC2 啟動執行個體精靈中的預設設定，快速啟動 Amazon EC2 執行個體。也了解如何檢閱執行個體組態欄位並終止執行個體。 持續時間：10 分鐘	教學課程 1：啟動我的第一個 Amazon EC2 執行個體
啟動測試 EC2 執行個體並與其連線 了解如何啟動可用於測試目的的 Amazon EC2 執行個體。此執行個體不會有進階組態，也不會存放敏感資訊。您也將了解基本執行個體組態設定、如何連線至執行個體，以及如何停止它。 持續時間：30 分鐘	教學課程 2：啟動測試 EC2 執行個體並與其連線

在尋找其他教學課程嗎？

- 教學課程：在 AL2023 上安裝 LAMP Web 伺服器
- 教學課程：在 AL2023 上設定 SSL/TLS
- 教學課程：託管 AL2023 上的 WordPress 部落格
- 教學課程：使用 EC2 Instance Connect 完成連線執行個體所需的設定
- 教學課程：將 Amazon EC2 執行個體連線至 Amazon RDS 資料庫

教學課程 1：啟動我的第一個 Amazon EC2 執行個體

屬性	Value
教學課程目標	了解如何使用 Amazon EC2 啟動執行個體精靈中的預設設定，快速啟動 Amazon EC2 執行個體。也了解如何檢閱執行個體組態欄位並終止執行個體。
EC2 體驗	入門
Duration (持續時間)	10 分鐘
成本	符合免費方案資格 建立 時 AWS 帳戶，您可以使用 免費開始使用 Amazon EC2AWS 免費方案。 如果您在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，但它還不到 12 個月，而且您尚未超過 Amazon EC2 的免費方案優惠，則完成本教學課程無需支付任何費用，因為我們會協助您選擇免費方案優惠內的選項。否則，從您啟動執行執行個體 (即使執行個體為閒置狀態) 到終止執行個體為止，會產生基本的 Amazon EC2 使用費用。 如果您在 2025 年 7 月 15 日當天 AWS 帳戶 或之後建立 ，但還不到 6 個月，而且您尚未用完

屬性	Value
	所有抵用金，則完成本教學課程無需支付任何費用，因為我們協助您選取 免費方案優惠內的選項。 如需如何判斷您是否符合免費方案資格的資訊，請參閱 the section called “追蹤您的免費方案用量”。
先決條件	• 您必須擁有 AWS 帳戶、設定具有管理員存取權的使用者，並使用管理員使用者登入 AWS 管理主控台。不確定如何執行此操作嗎？ 嘗試本教學課程：設定 AWS 您的環境 • 您必須熟悉 AWS 主控台。不確定從何處著手嗎？ 嘗試此入門指南： 入門 AWS 管理主控台

教學課程概觀

本教學課程專為先前沒有使用過 Amazon EC2 經驗的初學者而設計。我們會引導您完成使用 EC2 主控台建立第一個 EC2 執行個體的步驟 - 我們稱之為啟動。執行個體基本上是 AWS 雲端中的 Web 伺服器。啟動執行個體後，我們會示範如何在主控台中找到它。最後，為了協助您管理成本，我們會示範如何刪除 - 我們稱之為終止 - 您的執行個體。

本教學分為下列簡單扼要的任務。您必須先完成每個任務，才能移至下一個任務。

- 任務 1：啟動您的執行個體
- 任務 2：尋找您的執行個體
- 任務 3：檢視執行個體組態
- 任務 4：終止您的執行個體

任務 1：啟動您的執行個體

在此任務中，您將只執行必要項目，以最快的速度啟動執行個體。我們將使用 EC2 啟動執行個體精靈，這是一種 Web 型表單，提供設定和啟動執行個體的所有欄位。它會為執行個體組態欄位提供預設值，以簡化程序。

開始之前

請確定您已完成上表中列出的先決條件，包括 AWS 管理主控台 與您的管理員使用者登入 。

請依照下列步驟快速啟動執行個體

1. 開啟 Amazon EC2 主控台：

   前往 https：//https://console.aws.amazon.com/ec2/。
2. 開啟 EC2 啟動執行個體精靈：

   在 EC2 儀表板中，選擇啟動執行個體。

   啟動執行個體 Web 型表單隨即開啟。這是 EC2 啟動執行個體精靈。
3. 命名您的執行個體：

   在名稱和標籤下的名稱欄位中，輸入類似 **`My first EC2 instance`** 的描述性名稱。

   雖然不需要命名執行個體，但命名有助於在稍後識別您的執行個體。
4. 不使用金鑰對繼續：

   在金鑰對 (登入) 中，針對金鑰對名稱，選擇不使用金鑰對繼續 (不建議)。

   金鑰對可用於安全登入。不過，由於我們在本教學課程中不會登入執行個體，因此您目前不需要金鑰對。
5. 啟動您的執行個體：

   在右邊的 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

   Amazon EC2 使用預設設定快速啟動執行個體。成功橫幅代表確認啟動。

恭喜您！您已成功啟動第一個 EC2 執行個體！

任務 2：尋找您的執行個體

在此任務中，您會尋找您剛在 EC2 主控台中啟動的執行個體。

請依照下列步驟，在 EC2 主控台中尋找您的執行個體

1. 開啟執行個體頁面：

如果您仍在成功頁面上，請在畫面頂端的頁面導覽路徑中選擇執行個體。您可能需要先選擇三個點的省略符號才能存取它。

如果您已離開導覽，請從導覽窗格中選擇執行個體。

2. 尋找您的執行個體：

在名稱欄中，依您提供的名稱尋找執行個體。

任務 3：檢視執行個體組態

在此任務中，您將熟悉檢視執行個體的組態詳細資訊。

請依照下列步驟檢視執行個體的組態

1. 尋找執行個體 ID：

在執行個體 ID 欄中，記下執行個體的唯一 ID。ID 以 i- 開頭，後面接著 17 個英數字元，例如 i-01aeed690c9fb5322。

執行個體 ID 會在啟動時自動指派給您的執行個體。

2. 開啟執行個體詳細資訊頁面：

在執行個體 ID 欄中，選擇 ID 連結以開啟執行個體詳細資訊頁面，您可以在其中檢閱其組態。

3. 探索執行個體組態詳細資訊：

請花幾分鐘的時間來探索執行個體的組態詳細資訊。在下一個教學課程中，我們將深入探討組態。現在，請利用這段時間來熟悉執行個體詳細資訊頁面。

秘訣：若要快速尋找欄位，請按鍵盤上的 Ctrl+F 或 command+F。

a. 執行個體類型：您可以找到執行個體類型嗎？ 例如，它可能是 t3.micro。

b. 公有 IPv4 地址：您可以找到已配置給執行個體的公有 IPv4 地址嗎？ 其位於類似下列範例的格式：34.242.148.128。

c. 執行個體擁有者：您可以識別此執行個體的擁有者嗎？ 就是您！您的 AWS 帳戶 號碼會列在擁有者欄位下。

d. 執行個體標籤：您提供給執行個體的名稱實際上是標籤。您可以找到執行個體標籤嗎？ 選擇 Tags (標籤) 索引標籤。金鑰是名稱，而值是您提供的名稱。

e. 啟動時間：您可以找到啟動執行個體的時間嗎？ 選擇詳細資訊索引標籤並尋找啟動時間欄位。

f. 執行個體狀態：您可以驗證執行個體的狀態嗎？ 狀態應該為執行中。

再花幾分鐘的時間來探索其他執行個體組態欄位。準備好繼續時，請進行下一個任務。

任務 4：終止您的執行個體

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

在此任務中，您將刪除執行個體，以保留免費方案權益。在 EC2 中，終止是用於刪除執行個體的術語。

請依照下列步驟來終止執行個體

1. 啟動終止：

   如果您仍在執行個體詳細資訊頁面上，請選擇執行個體狀態功能表 (右上角)，然後選擇終止 (刪除) 執行個體。

   如果您已離開導覽，請從導覽窗格中選擇執行個體。接下來，在執行個體頁面上，選取執行個體名稱旁的核取方塊，然後選擇執行個體狀態功能表 (右上角)，接著選擇終止 (刪除) 執行個體。

2. 確認終止：

   在開啟的終止 (刪除) 執行個體視窗中，選擇終止 (刪除) 按鈕以確認您想要終止執行個體。

3. 監控執行個體狀態：

   在執行個體頁面上，檢查執行個體狀態欄。執行個體的狀態變更為正在關機。如果您沒有看到全文，請嘗試增加資料欄的寬度。

   執行個體關閉後，Amazon EC2 會刪除執行個體，並從執行個體頁面消失。

關鍵要點

在本教學課程中，您的學習範圍已涵蓋下列關鍵概念：

- 執行個體是指 AWS 雲端中的 Amazon EC2 Web 伺服器。
- 啟動是指建立 EC2 執行個體。
- 終止是指刪除 EC2 執行個體。
- EC2 啟動執行個體精靈包含執行個體組態的預設值，可快速輕鬆地啟動執行個體。
- 執行個體 ID 是自動指派給執行個體的唯一識別碼，而執行個體名稱是您可以指派的選用標籤，以便於識別。

後續步驟

若要建立啟動和終止執行個體的信心，請考慮重複本教學課程中的步驟。請務必終止您啟動的任何執行個體，以保留免費方案優惠。

熟悉這些基本概念後，請繼續進行下一個教學課程，這可讓您深入了解關鍵執行個體組態欄位。

教學課程 2：啟動測試 EC2 執行個體並與其連線

屬性	Value
教學課程目標	了解如何啟動可用於測試目的的 Amazon EC2 執行個體。此執行個體不會有進階組態，也不會存放敏感資訊。您也將了解基本執行個體組態設定、如何連線至執行個體，以及如何停止它。
EC2 體驗	入門
Duration (持續時間)	30 分鐘
成本	符合免費方案資格 建立 時 AWS 帳戶，您可以使用 免費開始使用 Amazon EC2AWS 免費方案。 如果您在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，但它還不到 12 個月，而且您尚未超過 Amazon EC2 的免費方案優惠，則完成本教學

屬性	Value
	課程無需支付任何費用，因為我們會協助您選擇免費方案優惠內的選項。否則，從您啟動執行執行個體 (即使執行個體為閒置狀態) 到終止執行個體為止，會產生基本的 Amazon EC2 使用費用。 如果您在 2025 年 7 月 15 日當天 AWS 帳戶 或之後建立 ，但還不到 6 個月，而且您尚未用完所有抵用金，則完成本教學課程無需支付任何費用，因為我們協助您選取 免費方案優惠內的選項。 如需如何判斷您是否符合免費方案資格的資訊，請參閱 the section called “追蹤您的免費方案用量”。
先決條件	完成 教學課程 1：啟動我的第一個 Amazon EC2 執行個體。

教學課程概觀

本教學課程專為想要啟動 EC2 執行個體的初學者而設計，可用於測試用途。

我們會說明金鑰執行個體組態欄位，接著引導您使用 EC2 主控台中的預設值啟動測試執行個體的步驟。啟動執行個體後，我們會示範如何登入 - 我們稱之為連線至 - 您的執行個體。我們也將示範如何建立金鑰對，這是在本教學課程中連線至執行個體時的必要項目。最後，為了協助管理成本，我們會示範停止執行個體，以避免收取使用費。

您將在本教學課程中啟動 Linux 執行個體。雖然本教學中的步驟可用來以其他作業系統啟動執行個體，但連線至執行個體的指示是 Linux 執行個體特有的。

本教學分為下列簡單扼要的任務。您必須先完成每個任務，才能移至下一個任務。

- 任務 1：熟悉啟動執行個體的關鍵元件
- 任務 2：檢閱技術圖表
- 任務 3：建立金鑰對
- 任務 4：啟動您的測試執行個體

- 任務 5：尋找您的執行個體
- 任務 6：檢視執行個體組態
- 任務 7：熟悉連接到執行個體的關鍵元件
- 任務 8：連線到您的執行個體
- 任務 9：停止執行個體

任務 1：熟悉啟動執行個體的關鍵元件

在此任務中，您將探索啟動 EC2 執行個體所需的關鍵元件。這些是 AMI、執行個體類型、金鑰對、安全群組、網路 (VPC 和子網路) 和 Amazon EBS 磁碟區。您也將探索選用元件，即名稱標籤。

為了協助您視覺化這些元件，請把執行個體想像成租房。就像租房一樣，您有一個可以居住的地方，但是不需要擁有和維護房產，EC2 執行個體也提供運算能力，但是不需要擁有和維護底下的基礎架構。

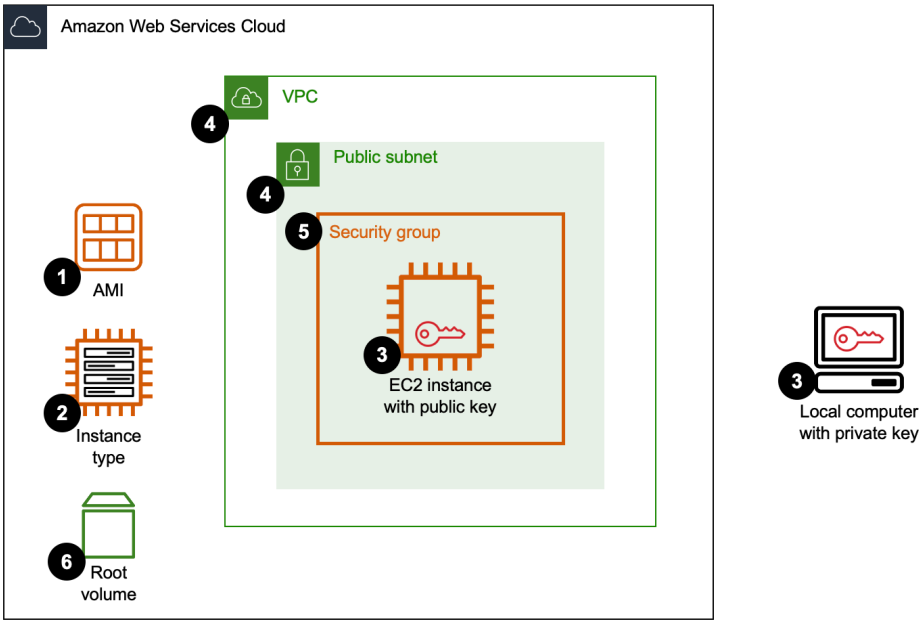
在決定要啟動的執行個體類型時，您會考慮執行個體的組態條件，就像考慮您想要的房屋條件一樣。雖然這種比喻簡化了事物，但這是一種有用的視覺化元件方法，直到您更熟悉它們為止。

- AMI – 建造房屋的材料和設施：Amazon Machine Image (AMI) 會決定執行個體啟動的作業系統和應用程式。這就像選擇您房屋的建材 (例如磚塊、鋼鐵或木頭) 和設施 (如家電和家具) 一樣。基本 AMI 就像是使用基本設備的無家具房屋，而使用預先安裝軟體的自訂 AMI 就像是提供完整家具的房屋。
- 執行個體類型 – 房屋大小和電源：執行個體類型會定義 EC2 執行個體的大小和功能，就像選擇房屋大小、房間數量和能源容量一樣。每個執行個體類型都會決定執行個體的 CPU、記憶體、儲存體和聯網容量。選取的 AMI 可能會限制您可以選擇的執行個體類型。
- 金鑰對 – 前門金鑰：金鑰對就像您房屋前門的鎖和鑰匙。公有金鑰的角色就像執行個體上的鎖，而私有金鑰是您必須在本機電腦上安全保存起來的鑰匙。如果其他人保有您的私有金鑰，他們可以存取您的執行個體，就像擁有您前門金鑰的人可以進入您的房屋一樣。
- 網路 (VPC 和子網路) – 屬性界線、區段區域以及房屋編號：您的虛擬私有雲端 (VPC) 就像您房屋所在地的整個屬性，而子網路是房屋周圍的隔離區段區域。如果您的屬性上有多個房屋 (執行個體)，根據用途，您可能想要將它們分割成不同的區域 (不同的子網路)。有些房屋允許訪客自由漫遊穿越花園 (具有網際網路存取的公有子網路)，有些房屋則有裝設圍欄的花園，限制訪客進入 (沒有網際網路存取的私有子網路)。每個子網路都包含一系列 IP 位址，就像房屋編號一樣，可以指派給子網路內的執行個體。
- 安全群組 – 大門管理員：安全群組的作用如同大門管理員，可控制允許誰造訪您的房屋。它會強制執行一組規則，控制哪些流量允許到達您的執行個體。例如，允許來自特定 IP 位址的 SSH 流量的規則，就像只讓特定人員交付雜貨的大門管理員。同樣地，允許來自任何地方的 HTTPS 流量就像讓公眾來訪並打量您房屋的外觀。

- Amazon EBS 磁碟區 - 儲存體單位：EBS 磁碟區就像儲存體單位一樣，您可以在其中存放個人物品。每個執行個體都有根磁碟區 (存放 AMI 的位置)，而且您可以視需要隨時新增更多磁碟區 (儲存體)。
- 名稱標籤 – 房屋名稱：名稱標籤的功能就像房屋的標誌，可協助您輕鬆識別住在屋內的對象。雖然名稱標籤可讓您更輕鬆地區分執行個體，但在啟動執行個體時則不需要。

任務 2：檢閱技術圖表

在此任務中，您將熟悉我們在 AWS 文件中使用的典型技術圖表。下圖代表您將在本教學課程中啟動之測試執行個體的組態。在上一個任務中，我們使用租房的比喻來介紹這些元件。現在，我們將專注於實際的 EC2 元件本身。編號標籤對應至後面的描述。



1. AMI – AMI 是您在啟動執行個體時選擇的映象。這是包含了要在執行個體上執行的作業系統和軟體的範本。例如，如果您想要啟動 Linux 執行個體，您可以選擇 Amazon Linux 2023 AMI。或者，如果您想要啟動 Windows 執行個體，您可以選擇 Microsoft Windows Server 2022 Base AMI。Amazon EC2 主控台中的 AMI 目錄包含 1000 個可供選擇的映象。
2. 執行個體類型 – 執行個體類型是決定執行個體所用之主機電腦 CPU、記憶體、儲存體和聯網容量的硬體。Amazon EC2 提供超過 600 種執行個體類型供您選擇，每種執行個體的硬體組態和大小各不相同，可讓您選擇最適合您應用程式的需求的類型。
3. 金鑰對 – 金鑰對是一組安全性登入資料，您可以在連線至執行個體時用來證明身分。公有金鑰會在執行個體上，而私有金鑰則會在本機電腦上。

在 EC2 中，連線至執行個體是指從本機電腦登入執行個體。雖然還有其他方式可安全地連線至您的執行個體，但在本教學課程中，我們會使用金鑰對來連線。

4. 網路 – 網路由 VPC 和一個或多個子網路組成。VPC 是 AWS 雲端內的虛擬網路。每個 AWS 客戶都有自己的專用 VPC AWS 帳戶。您會在 VPC 的子網路中啟動執行個體。子網路是 VPC 中的 IP 位址範圍。您的預設子網路是公有子網路，這表示它會指派公有 IP 位址，並提供從 Amazon 網路外部存取執行個體的網際網路。
5. 安全群組 – 安全群組會做為防火牆來控制執行個體的流量。安全群組包含允許特定類型流量進入執行個體的規則。若要從本機電腦透過 SSH 連線至執行個體 (使用金鑰對)，您需要允許本機電腦 SSH 流量的規則。
6. EBS 磁碟區 – Amazon EBS 磁碟區是運作方式與實體硬碟類似的儲存裝置。您的執行個體隨附根磁碟區，這是一個特殊的 EBS 磁碟區，可將 AMI 與啟動執行個體所需的作業系統和軟體一起存放。您可以選擇性地新增資料磁碟區。不過，由於您的測試執行個體不會儲存任何敏感資料，因此您不需要額外的加密資料磁碟區。

恭喜您！您已完成本教學課程中的概念性任務。在以下任務中，您將使用 Amazon EC2 主控台來建立您已學過的元件。

任務 3：建立金鑰對

在此任務中，您將建立金鑰對。金鑰對由兩個部分組成：您要新增至執行個體的公有金鑰，以及相符的私有金鑰，用於安全地連線至執行個體。在下一個任務中，您將在啟動執行個體時選取此金鑰對，這會自動將公有金鑰新增至執行個體。請務必將私有金鑰安全地存放在本機電腦上，因為任何有權存取私有金鑰的人都可以連線到您的執行個體。

如果您偏好在啟動測試執行個體時使用現有的金鑰對，請隨時略過此任務。如不使用現有的金鑰對，請繼續建立新的金鑰對。

開始之前

請確定您已完成上表中列出的先決條件，包括 AWS 管理主控台 與您的管理員使用者登入 。

請執行下列步驟以建立金鑰對。

1. 開啟 Amazon EC2 主控台：

   前往 https：//https://console.aws.amazon.com/ec2/。
2. 導覽至金鑰對主控台頁面：

請在導覽窗格的 Network & Security (網路與安全性) 下方，選擇 Key Pairs (金鑰對)。

- 如果您先前已建立金鑰對，它們會出現在資料表中。
- 如果不存在金鑰對，則資料表為空白。

3. 建立新的金鑰對：

   選擇建立金鑰對按鈕 (右上角) 以開啟建立金鑰對 Web 型表單，並輸入您的金鑰對詳細資訊，如下所示：

   a. 為您的金鑰對命名：針對名稱，輸入可協助您辨識金鑰對的名稱，例如 **`test-instance-key-pair`**。

      名稱長度最多為 255 個 ASCII 字元。它不能包括前置或結尾空格。

   b. 選擇金鑰對類型：針對金鑰對類型，選擇 ED25519。

      Linux 執行個體同時支援 RSA 和 ED25519 金鑰類型，而 Windows 執行個體僅支援 RSA。由於您將在本教學課程中啟動 Linux 執行個體，因此您可以使用 ED25519 金鑰。

   c. 選擇私有金鑰的儲存格式：針對私有金鑰檔案格式，請選擇 .pem。

      這是儲存私有金鑰檔案的格式。

4. 將公有金鑰儲存至 Amazon EC2 並下載私有金鑰：

   選擇建立金鑰對按鈕 (右下方)。

   Amazon EC2 會儲存公有金鑰，而您的瀏覽器會將私有金鑰檔案自動下載至本機電腦。檔案會根據您為金鑰對指定的名稱命名，而副檔名是您選擇的檔案格式。將私有金鑰檔案移至電腦上的安全位置。

   > **Important**
   >
   > 這是您儲存私有金鑰檔案的唯一機會。

5. 設定金鑰的許可 (適用於 macOS 和 Linux 使用者)：

   如果您計劃在 macOS 或 Linux 電腦上使用 SSH 連線到執行個體，您必須為私有金鑰檔案設定正確的許可。開啟終端機視窗並執行下列命令，將 *`test-instance-key-pair`* 取代為您的金鑰對名稱：

```
chmod 400 test-instance-key-pair.pem
```

此命令可確保只有您可以讀取私有金鑰檔案，這是與執行個體建立安全連線所必需的命令。如果沒有這些許可，您將無法使用此金鑰對進行連線。

恭喜您！您已成功建立金鑰對！

任務 4：啟動您的測試執行個體

在此任務中，您將使用 EC2 啟動執行個體精靈快速啟動測試執行個體。您將為 Linux 執行個體設定主要執行個體組態設定，並使用其他欄位的預設值。

為了協助您管理成本，我們建議您選擇符合免費方案資格的元件。

請依照下列步驟來啟動測試執行個體

1. 開啟 Amazon EC2 主控台：

   前往 https：//https://console.aws.amazon.com/ec2/。
2. 開啟 EC2 啟動執行個體精靈：

   在 EC2 儀表板中，選擇啟動執行個體。

   啟動執行個體 Web 型表單隨即開啟。這是 EC2 啟動執行個體精靈。
3. 命名您的執行個體：

   在名稱和標籤下的名稱欄位中，輸入類似 **Test instance** 的描述性名稱。

   執行個體名稱是一個標籤，其中鍵是 Name (名稱)，而值是您指定的名稱。

   秘訣：針對測試執行個體，有一個名稱標籤便足夠了。不過，對於生產執行個體，最佳實務是建立標記政策，以標準化所有資源的標記。
4. 選擇您的作業系統和軟體 - Amazon Machine Image (AMI)：

   在應用程式和作業系統映像 (Amazon Machine Image) 下，Amazon Machine Image (AMI) 的預設選擇是 Amazon Linux 2023 AMI。此 AMI 會標示為符合免費方案。在本教學課程中，您將啟動 Linux 執行個體，因此請將預設設定保持在免費方案限制下。
5. 選擇您的硬體 - 執行個體類型：

在執行個體類型下，對於執行個體類型，保留本教學課程的預設選擇。預設執行個體類型可用於免費方案，且其硬體適用於您的測試執行個體。

6. 準備使用金鑰對進行安全登入：

   在金鑰對 (登入) 下的金鑰對名稱欄位中，選擇您在前一個任務建立的金鑰對。如果在清單中沒有看到您的執行個體，則請選擇重新整理圖示 (在清單右側)。

   當您的執行個體啟動時，它會將公有金鑰放在執行個體上。若要在執行個體啟動後連線到執行個體，您將使用您在先前任務中下載的對應私有金鑰。

7. 設定網路設定以啟用網際網路存取：

   在網路設定下，會依據預設來設定網路 (您的 VPC) 和子網路欄位。保留本教學課程的預設設定，以協助您快速開始使用。如果您尚未修改預設子網路，您的執行個體將具有網際網路存取權。

   秘訣：您的預設子網路是公有子網路，這表示它將指派公有 IP 位址，並提供從 Amazon 網路外部存取執行個體的網際網路。對於測試執行個體，可以使用提供網際網路存取的預設子網路設定。不過，對於生產執行個體，最佳實務是僅指派公有 IP 位址，並在絕對必要時使用具有網際網路存取的子網路。

8. 設定執行個體防火牆 (安全群組)：

   在網路設定下，在防火牆 (安全群組) 下，保持選取允許來自任何位置 (0.0.0.0) 的 SSH 流量核取方塊。這會為您的測試執行個體建立新的安全群組，以允許來自任何 IP 位址的 SSH 流量。

   安全群組會做為防火牆來控制執行個體的流量。若要從本機電腦透過 SSH 連線至執行個體，您需要允許本機電腦 SSH 流量的規則。

   秘訣：如果您的網際網路服務供應商使用動態 IP 指派，則本機電腦的 IP 位址可能會隨著時間而變更。我們假設當您使用執行個體進行測試時，您不會使用執行個體來存放敏感資訊，因此安全措施的限制性較低。對於測試執行個體，通常可以允許來自任何 IP 位址 (`0.0.0.0/0`) 的流量，因此即使您的 IP 位址變更，您仍然可以隨時連線。不過，對於生產執行個體，特別是具有敏感資料的執行個體，最佳實務是僅允許來自已知 IP 位址的流量。

9. 設定執行個體儲存體：

   在設定儲存體下，根磁碟區 (加密) 欄位設定為預設。保持設定不變，以控制在免費方案限制下。

   由於我們的測試執行個體不會儲存任何敏感資料，因此我們不需要額外的加密資料磁碟區。

10. 檢閱執行個體組態：

在右側的摘要面板中，您可以在啟動執行個體之前檢閱高階設定。

11. 啟動您的執行個體：

    當您準備好啟動執行個體時，請在摘要面板中選擇啟動執行個體。

    Amazon EC2 會使用您指定的設定快速啟動執行個體。如果您未指定設定，則會使用預設值。成功橫幅代表確認啟動。

恭喜您！您已成功啟動執行個體！

任務 5：尋找您的執行個體

在此任務中，您會尋找您剛在 EC2 主控台中啟動的執行個體。

請依照下列步驟，在 EC2 主控台中尋找您的執行個體

1. 開啟執行個體頁面：

   如果您仍在成功頁面上，請在成功橫幅中選擇執行個體 ID。

   如果您已離開導覽，請從導覽窗格中選擇執行個體。

2. 尋找您的執行個體：

   在名稱欄中，依您提供的名稱尋找執行個體。

任務 6：檢視執行個體組態

在此任務中，您將熟悉檢視執行個體的組態詳細資訊。

請依照下列步驟檢視執行個體的組態

1. 尋找您的執行個體：

   在名稱欄中，依您提供的名稱尋找執行個體。

2. 開啟執行個體詳細資訊頁面：

   選取執行個體名稱旁的核取方塊，接著選擇動作功能表 (右上角)，然後選擇檢視詳細資訊以開啟執行個體詳細資訊頁面，您可以在其中檢閱其組態。

在先前的教學課程中，您選擇了執行個體的 ID 連結來開啟執行個體詳細資訊頁面。您將發現在 EC2 主控台中有一種以上的可以完成任務的方法。

3. 探索執行個體組態詳細資訊：

   請花幾分鐘的時間來探索執行個體的組態詳細資訊。

   秘訣：若要快速尋找欄位，請按鍵盤上的 Ctrl+F 或 command+F。

   a. AMI：您可以找到用來啟動執行個體的 AMI 嗎？ 您可以在詳細資訊索引標籤的 AMI ID 和 AMI 名稱中找到資訊。
   b. 執行個體類型：您可以找到執行個體類型嗎？ 例如，它可能是 t3.micro。
   c. 金鑰對：您可以找到啟動執行個體時選取的金鑰對嗎？ 它會指定給啟動時指派的金鑰對。請注意，如果您未來變更金鑰對，此處的值不會變更。
   d. VPC：您可以找到 VPC 的 ID 嗎？ 您可以在網路索引標籤上找到所有與網路相關的組態設定。VPC ID 格式類似下列範例：vpc-1a2b3c4d
   e. 子網路：您可以找到啟動執行個體的子網路 ID 嗎？ 其位於類似下列範例的格式：子網路-1a2b3c4d
   f. 公有 IPv4 地址：您可以找到已配置給執行個體的公有 IPv4 地址嗎？ 其位於類似下列範例的格式：34.242.148.128。
   g. 安全群組：您可以找到為了允許來自任何地方 (0.0.0.0/0) 的 SSH 流量而建立的傳入規則嗎？ 您可以在安全性索引標籤中找到所有與安全性相關的組態設定。
   h. 儲存：您可以找到為此執行個體建立的磁碟區嗎？ 您可以在儲存體索引標籤上找到所有與儲存體相關的組態設定。
   i. 執行個體標籤：您提供給執行個體的名稱實際上是標籤。您可以找到執行個體標籤嗎？ 選擇 Tags (標籤) 索引標籤。金鑰是名稱，而值是您提供的名稱。
   j. 執行個體狀態：您可以驗證執行個體的狀態嗎？ 狀態應該為執行中。

再花幾分鐘的時間來探索其他執行個體組態欄位。準備好繼續時，請進行下一個任務。

任務 7：熟悉連接到執行個體的關鍵元件

在此任務中，您將探索連線到 EC2 執行個體所需的關鍵元件。這些是連線通訊協定、公有 DNS、安全群組、金鑰對和執行個體使用者名稱。

為了協助您視覺化這些元件，請把連線到執行個體想像成前往您的家中：

- 連線通訊協定 – 您的傳輸模式：就像選擇返家的方式一樣，您可以選擇將帶您前往執行個體的連線通訊協定。在本教學課程中，我們將使用 SSH (Secure Shell)，這會建立安全的通道，以透過網際網路將您的電腦連線至執行個體。
- 公有 DNS – 房屋位址：就像您的住家具有獨一無二的地址一樣，EC2 執行個體也有自己的公有 DNS 名稱 (例如 `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com`)。此公有 DNS 名稱可讓 SSH 直接連線至您的執行個體。
- 安全群組 – 大門管理員：想像您的房屋有一位大門管理員，控制誰可以進入或離開房屋。同樣地，EC2 執行個體有一個安全群組，其作用如同大門管理員，可控制執行個體內外允許的網路流量類型。僅允許您明確允許的流量 (例如來自電腦 IP 位址的 SSH 流量)。
- 私有金鑰 – 您的前門金鑰：啟動執行個體時，您指定了金鑰對。公有金鑰放在執行個體上，而您保留了電腦上的私有金鑰。私有金鑰的角色就像您的前門金鑰，如果沒有私有金鑰，您就無法進入執行個體。
- 執行個體使用者名稱 – 居民：當您抵達住宅時，您需要表明自己的身分，以證明您是居民。同樣地，在連線至執行個體時，您會提供使用者名稱。取決於其作業系統，不同執行個體會有不同的預設使用者名稱。例如，Amazon Linux 執行個體使用 `ec2-user` 做為預設使用者名稱。

連線命令

若要連線至 EC2 執行個體，請在終端機視窗中使用下列命令：

```
ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
west-1.compute.amazonaws.com
```

以下是命令所執行作業的明細：

- `ssh` – 此命令會指定連線通訊協定，啟動 SSH (Secure Shell) 與您的執行個體的連線。
- `-i "test-instance-key-pair.pem"` – `-i` 標記表示驗證連線所需的私有金鑰檔案。此私有金鑰檔案必須符合您在啟動執行個體時指定的金鑰對。如果您的私有金鑰檔案儲存在特定資料夾中，請指定檔案的完整路徑。
- `ec2-user` – 這是用於登入執行個體的使用者名稱。Amazon Linux 執行個體的預設使用者名稱為 `ec2-user`。其他 AMI 可能會使用不同的預設使用者名稱，例如適用於 Ubuntu 執行個體的 `ubuntu`。
- `@` – 此符號會將使用者名稱與執行個體的位址分開。
- `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com` – 這是執行個體的公有位址 (公有 DNS)，其中包含公有 IPv4 地址和 AWS 區域。它會唯一識別執行個體。

執行命令時會發生什麼情況

在您執行命令之後，SSH 會建立安全通道，並使用私有金鑰進行身分驗證。如果執行個體的安全群組允許流量，您可以存取 EC2 執行個體。您現在可以透過電腦控制執行個體，就像坐在執行個體前面一樣。您可以執行命令、安裝軟體和管理檔案，就像在本機電腦上一樣。

任務 8：連線到您的執行個體

在此任務中，您將使用電腦上的 SSH 用戶端連線到執行個體。在上一個任務中，我們使用前往你家的比喻來介紹連線到執行個體的元件。現在，我們將專注於連線到實際的 EC2 執行個體。

有不同的方法來連線到執行個體。您用來連線的方法取決於執行個體的作業系統。由於您已啟動 Linux 執行個體，因此您將在本機電腦上使用 SSH 用戶端。

首先，檢查您的電腦是否已安裝 SSH 用戶端

大多數電腦都已預先安裝 SSH 用戶端。若要檢查，請開啟電腦上的終端機視窗，然後執行下列命令：

```
ssh
```

如果已辨識命令，您就可以準備連線。

如果無法辨識命令，您必須安裝 SSH 用戶端。安裝 SSH 用戶端的指示超出本教學課程的範圍。如果您需要說明，請參閱本使用者指南中的 SSH 連線先決條件 或線上搜尋，以取得如何在作業系統上安裝 SSH 用戶端的指示。

請依照下列步驟來連結執行個體

1. 啟動連線：

   如果您在 Amazon EC2 主控台的執行個體詳細資訊頁面上，請選擇連線按鈕 (右上角)。

   如果您已離開導覽，請從導覽窗格中選擇執行個體。接下來，在執行個體頁面上，選取執行個體名稱旁的核取方塊，然後選擇連線按鈕 (右上角)。

   這會開啟連線至執行個體頁面。
2. 選擇連線方法：

   在連線至執行個體頁面，選擇 SSH 用戶端索引標籤。

   請花一點時間檢閱此頁面上的文字，因為這些是接下來要遵循的步驟。

3. 檢閱 SSH 命令：

   在範例下，您會看到自動產生並自訂執行個體詳細資訊的命令。私有金鑰名稱衍生自啟動時指定的公有金鑰名稱。

   命令看起來像這樣：

   ```
   ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
   west-1.compute.amazonaws.com
   ```

4. 複製 SSH 命令：

   選擇範例 SSH 命令旁的複製圖示。

5. 開啟終端機視窗：

   在您的本機電腦上開啟終端機視窗。

6. 貼上並執行 SSH 命令：

   將 SSH 命令貼到終端機視窗中。如果您將私有金鑰檔案儲存在特定資料夾中，請編輯命令以包含完整的檔案路徑。

   按下鍵盤上的 Enter。

   您將看到類似以下的回應：

   ```
   The authenticity of host 'ec2-18-201-118-201.eu-west-1.compute.amazonaws.com
    (18-201-118-201)' can't be established.
   ED25519 key fingerprint is SHA256:examplehxj9aOr1MogvKOoMNskVVIRBQBoq0example.This
    key is not known by any other names.
   Are you sure you want to continue connecting (yes/no/[fingerprint])?
   ```

7. 完成連線：

   輸入 **yes** 並按下鍵盤上的 Return。

   驗證指紋超出本教學課程的範圍。如需詳細資訊，請參閱 (選用) 取得執行個體指紋。

   成功連線後，終端提示會變更為顯示執行個體的公有 DNS。

恭喜您！您已成功連線至執行個體！

任務 9：停止執行個體

在此任務中，您將停止執行個體以保留免費方案權益。在您的執行個體停止時，您會停止產生其費用。如果您在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，且符合 免費方案的資格，則您將繼續產生 EBS 儲存體的費用。

請依照下列步驟來停止執行個體

1. 啟動停止：

   如果您仍在連線至執行個體頁面，請從頁面導覽路徑中選擇執行個體。如果您已離開導覽，請從導覽窗格中選擇執行個體。

   接下來，在執行個體頁面上，選取執行個體名稱旁的核取方塊，然後選擇執行個體狀態功能表 (右上角)，接下來選擇停止執行個體。出現提示時，請選擇停止。
2. 監控執行個體狀態：

   在執行個體頁面上，檢查執行個體狀態欄。執行個體的狀態會變更為正在停止，然後變更為已停止。如果您沒有看到全文，請嘗試增加資料欄的寬度。

   如果您認為執行個體狀態已從正在停止變更為已停止，但尚未看到狀態變更為已停止，請選擇重新整理圖示 (資料表上方) 以重新整理執行個體資料表。

關鍵要點

在本教學課程中，您的學習範圍已涵蓋下列關鍵概念：

- AMI 是指 Amazon Machine Image，這是包含啟動執行個體所需的作業系統和軟體的範本。
- 執行個體類型是指您的執行個體使用的主機電腦硬體。它決定執行個體的 CPU、記憶體、儲存體和聯網容量。
- 金鑰對是指一組公有和私有金鑰，您可以用來安全地連線至執行個體。
- 網路是指 VPC （雲端內帳戶專用的虛擬私有雲端 AWS) 和子網路 (VPC 內的 IP 地址範圍）。
- 安全群組是指一組規則，可控制哪些流量可以到達您的執行個體。
- EBS 磁碟區是指執行個體的資料儲存。每個執行個體都有根磁碟區，用於存放 AMI 和一個或多個選用資料磁碟區。
- 標籤是您可以選擇性地指派給執行個體的中繼資料。執行個體名稱是標籤，其金鑰是名稱，而值是您選擇的值。
- 連線是指透過網際網路存取執行個體。

- SSH 是指 Secure Shell 連線通訊協定，可用來連線至執行個體。
- 公有 DNS 是您執行個體的唯一公有位址。
- 執行個體使用者名稱是由執行個體的作業系統決定，並且是連線所需要的項目。
- 正在停止執行個體會停止執行個體的收費，但 EBS 儲存體會繼續收費。

後續步驟

若要建立啟動、連線和停止執行個體的信心，請考慮重複本教學課程中的步驟。請務必終止您啟動的任何執行個體，以保留免費方案優惠。

熟悉這些基本概念後，您可以探索更進階的教學課程。如需更多教學課程，請參閱 在尋找其他教學課程嗎？

如果您在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，請考慮觀看以下 6 分鐘的影片：使用 AWS 免費方案 服務時，如何避免我的帳戶產生費用

如果您在 2025 年 7 月 15 日當天 AWS 帳戶 或之後建立 ，請考慮檢閱下列資訊：AWS Billing 《 使用者指南》中的使用 AWS 免費方案探索 AWS 服務

Amazon EC2 執行個體組態參數的參考

Amazon EC2 主控台中的啟動執行個體精靈和啟動範本會提供設定 Amazon EC2 執行個體的所有參數。

除了金鑰對之外，啟動執行個體精靈還會提供每個參數的預設值。您可以接受任何或全部預設值，或者利用您自己的值來設定執行個體。在建立啟動範本時，參數為選用參數。如果您使用啟動範本來啟動執行個體，則啟動範本中指定的參數會覆寫啟動執行個體精靈中的預設值。啟動範本中未指定的任何參數，預設會是啟動執行個體精靈所提供的值。

這些參數會在啟動執行個體精靈和啟動範本中分組。系統會根據主控台中的參數群組來進行下列說明。

執行個體組態的參數

- 名稱和標籤
- 應用程式和作業系統映像 (Amazon Machine Image)
- 執行個體類型
- 金鑰對 （ 登入 ）
- 網路設定

- 設定儲存
- 進階詳細資訊
- 摘要

名稱和標籤

執行個體名稱是一個標籤，其中鍵是 Name (名稱)，而值是您指定的名稱。您可以標記執行個體、磁碟區以及網路介面。對於 Spot 執行個體，您只能標記 Spot 執行個體請求。如需標籤的相關資訊，請參閱標記您的 Amazon EC2 資源。

指定執行個體名稱和其他標籤是選用的。

- 對於 Name (名稱)，輸入執行個體的描述性名稱。如果您未指定名稱，則可以透過其 ID 來標識執行個體，該 ID 將在您啟動執行個體時自動產生。
- 若要新增其他標籤，請選擇 Add additional tags (新增其他標籤)。選取 Add tag (新增標籤)，然後輸入鍵和值，然後選取要標記的資源類型。為每個要新增的其他標籤重新選擇 Add tag (新增標籤)。

您只能在啟動執行個體時，指定執行個體名稱。您無法在建立啟動範本時命名執行個體，但可以為啟動執行個體時所建立的資源新增標籤。

應用程式和作業系統映像 (Amazon Machine Image)

Amazon Machine Image (AMI) 包含建立執行個體所需的資訊。例如，AMI 可能包含做為 Web 伺服器所必需的軟體，例如 Linux、Apache 和您的網站。

您可以尋找適當的 AMI，如下所示。對於尋找 AMI 的每個選項，可以選擇右上方的 Cancel (取消)，以返回啟動執行個體精靈，而不選擇 AMI。

搜尋列

若要搜尋所有可用的 AMI，請在 AMI 搜尋列中輸入關鍵字，然後按 Enter 鍵。選取 Select (選取) 來選取 AMI。

Recents (最近)

最近使用的 AMI。

選取 Recently launched (最近啟動) 或 Currently in use (目前使用中)，然後從 Amazon Machine Image (AMI) 中，選取 AMI。

My AMIs (我的 AMI)

您擁有的私有 AMI，或是與您共享的私有 AMI。

選取 Owned by me (由我擁有) 或 Shared with me (與我共享)，然後從 Amazon Machine Image (AMI) 中，選取 AMI。

Quick Start

AMI 會依作業系統分組，以協助您快速開始使用。

首先選取您需要的操作系統，然後從 Amazon Machine Image (AMI) 中，選取 AMI。若要選取可在 AWS 免費方案下使用的 AMI，請確定 AMI 已標記為符合免費方案資格。

Browse more AMIs (瀏覽更多 AMI)

選擇 Browse more AMIs (瀏覽更多 AMI) 以瀏覽完整的 AMI 目錄。

- 若要搜尋所有可用的 AMI，請在搜尋列中輸入關鍵字，然後按 Enter 鍵。
- 若要使用 Systems Manager 參數尋找 AMI，請選擇搜尋列右側的箭頭按鈕，然後選擇 Search by Systems Manager parameter (依 Systems Manager 參數搜尋)。如需詳細資訊，請參閱使用 Systems Manager 來參數參考 AMI。
- 若要依類別搜尋，請選擇 Quickstart AMIs (AMI 快速入門)、My AMIs (我的 AMI)、AWS Marketplace AMI, 或 Community AMIs (社群 AMI)。

  AWS Marketplace 是一個線上商店，您可以在其中購買 上執行的軟體 AWS，包括 AMIs。如需從 啟動執行個體的詳細資訊 AWS Marketplace，請參閱 從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體。在 Community AMIs (社群 AMI) 中，您可以尋找 AWS 社群成員已提供給其他人使用的 AMI。來自 Amazon 或已驗證的合作夥伴的 AMI 標記為 Verified provider (已驗證的供應商)。
- 若要篩選 AMI 清單，請在畫面左側的 Refine results (精簡結果) 下選取一或多個核取方塊。篩選條件選項會有所不同，取決於選取的搜尋類別。
- 檢查每個 AMI 列出的 Root device type (根設備類型)。注意哪些 AMI 是您需要的類型：ebs (Amazon EBS 後端) 或 instance-store (執行個體存放區後端))。如需詳細資訊，請參閱 根磁碟區類型。
- 檢查每個 AMI 列出的 Virtualization type (虛擬化類型)。注意哪些 AMI 是您需要的類型：hvm 或 paravirtual。例如，有些執行個體類型需要 HVM。如需關於 Linux 虛擬化類型的詳細資訊，請參閱 虛擬化類型。
- 檢查每個 AMI 列出的開機模式。注意哪些 AMI 使用您需要的開機模式：legacy-bios、uefi 或 uefi-preferred。如需詳細資訊，請參閱使用 Amazon EC2 開機模式執行個體啟動行為。

- 選取符合您需求的 AMI，然後選取 Select (選取)。

變更 AMI 時發出警示

當您啟動執行個體時，如果修改了與選定 AMI 關聯的任何磁碟區或安全群組的組態，然後選取不同的 AMI，則會開啟一個視窗，警示您目前的某些設定將會變更或移除。您可以檢閱對安全群組和磁碟區做出的變更。此外，您可以檢視要新增和刪除的磁碟區，或者僅檢視待新增的磁碟區。建立啟動範本時，不會出現此警告。

執行個體類型

執行個體類型定義執行個體的硬體組態和大小。較大的執行個體類型具有較多的 CPU 和記憶體。如需詳細資訊，請參閱 Amazon EC2 執行個體類型。

- Instance type (執行個體類型)：確定執行個體類型與您指定的 AMI 相容。如需詳細資訊，請參閱Amazon EC2 執行個體類型。

  免費方案 – 您可使用免費方案下標記為符合免費方案資格的執行個體類型。特定執行個體類型取決於您建立 的時間 AWS 帳戶。

  如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立 ，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。

  如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。

  如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。
- Compare instance types (比較執行個體類型)：您可以依下列屬性比較不同的執行個體類型：vCPU 數目、架構、記憶體數量 (GiB)、儲存數量 (GB)、儲存類型，以及網路效能。
- 取得建議：您可以從 EC2 執行個體類型搜尋工具取得執行個體類型的指導和建議。如需詳細資訊，請參閱從 EC2 執行個體類型搜尋工具取得建議。
- (僅限啟動範本) 進階：若要指定執行個體屬性，並讓 Amazon EC2 利用這些屬性識別執行個體類型，請選擇進階，然後選擇指定執行個體類型屬性。
  - Number of vCPUs (vCPU 數量)：輸入運算需求的 vCPU 數量下限和上限。若要表示沒有限制，請輸入下限 **0**，並將上限保留為空白。

- Amount of memory (MiB) (記憶體數量 (MiB))：輸入運算需求的記憶體數量 (MiB) 下限和上限。若要表示沒有限制，請輸入下限 **0**，並將上限保留為空白。
- 展開 Optional instance type attributes (選用執行個體類型屬性)，然後選擇 Add attribute (新增屬性)，以更詳細地表達您的運算需求。如需每個屬性的詳細資訊，請參閱 Amazon EC2 API 參考中的 InstanceRequirementsRequest。
- Resulting instance types (產生的執行個體類型)：您可以預覽符合所指定屬性的執行個體類型。若要排除執行個體類型，請選擇 Add attribute (新增屬性)，然後從 Attribute (屬性) 清單中，選擇 Excluded instance types (排除的執行個體類型)。從 Attribute value (屬性值) 清單中，選取要排除的執行個體類型。

金鑰對 （登入）

針對 Key pair name (金鑰對名稱)，選擇現有的金鑰對，或選擇 Create new key pair (建立新的金鑰對) 以建立新的金鑰對。如需詳細資訊，請參閱 Amazon EC2 金鑰對和 Amazon EC2 執行個體。

> **Important**
>
> 如果您選擇 Proceed without key pair (Not recommended) (繼續而不使用金鑰對 (不建議)) 選項，您將無法連線到執行個體，除非您選擇已設定為允許使用者透過其他方式登入的 AMI。

網路設定

網路設定會定義執行個體的 IP 位址、安全群組及網路介面。您可使用預設的網路設定，或視需要進行設定。

- (僅限啟動執行個體精靈) VPC：為您的執行個體選擇現有的 VPC。預設情況下會選取區域的預設 VPC。或者，您可選擇您建立或與您共用的 VPC。如需詳細資訊，請參閱EC2 執行個體的虛擬私有雲。
- 子網路：為您的執行個體選擇子網路，或選擇建立新子網路，以使用 Amazon VPC 主控台來建立新的子網路。
  - 您可在任何可用區域、本機區域、Wavelength 區域或 Outpost 區域中為選取的 VPC 建立子網路。
  - 若要在僅限 IPv6 的子網路中啟動執行個體，該執行個體必須為 Nitro 型執行個體。

- (僅限啟動執行個體精靈) 自動指派公有 IP：啟用或停用公有 IPv4 位址的自動指派。如果在預設子網路中啟動執行個體，則預設值為啟用。在非預設子網路中啟動執行個體時，預設值是停用。如需詳細資訊，請參閱公有 IPv4 地址。

  若您新增次要網路介面，無法為非預設子網路啟用此選項。如需詳細資訊，請參閱the section called "在啟動時指派公有 IPv4 位址"。
- (僅限啟動執行個體精靈) 自動指派 IPv6 IP：啟用或停用 IPv6 位址的自動指派。如需詳細資訊，請參閱IPv6 地址。
- 防火牆 (安全群組)：選擇現有安全群組，或建立新的安全群組。確保您的安全群組具有規則，允許往返執行個體的流量。所有其他流量都會遭到忽略。

  若您建立新的安全群組，我們會自動建立傳入規則，允許您透過 SSH (Linux 執行個體) 或 RDP (Windows 執行個體) 從所有 IP 位址連線至執行個體。您無法移除或修改這項規則。您可視需要新增規則。如需詳細資訊，請參閱設定安全群組規則。

  > **Warning**
  >
  > 如果您短暫啟動測試執行個體，而且很快就會將其停止或終止，則啟用所有 IP 位址，透過 SSH 或 RDP 存取您執行個體的規則是可接受的。但這些規則在生產環境中並不安全。建議您僅授權特定 IP 位址範圍存取您的執行個體。

  此安全群組會新增至主要網路介面及所有次要網路介面。您可為網路介面選擇額外的安全群組，但無法移除您在此處選擇的安全群組。
- 進階網路組態 – 您可視需要設定主要網路介面。若要新增次要網路介面，請選擇 Add network interface (新增網路介面)。您可新增的網路介面數取決於選取的執行個體類型。請注意，僅有在您選擇子網路時，才能使用本節。
  - 裝置索引：裝置索引。主要網路介面必須指派給索引 0。
  - 網路介面：網路介面。選取新介面讓 Amazon EC2 建立新的介面，或是選取現有的可用網路介面。若您選擇現有網路介面作為主要網路介面，則無法為非預設子網路啟用自動指派公有 IP。
  - 描述：新網路介面的描述。
  - Subnet (子網)：要建立新網路介面的子網。執行個體會在與主要網路介面相同的子網路中啟動。

    您必須從與主要網路介面的子網路相同的可用區域，選擇次要網路介面的子網路。如果您從另一個 VPC 選取子網路，多 VPC 標籤會顯示在網路介面旁。如此一來，您就能在不同 VPC 建立多重主目錄執行個體，分別使用不同的網路和安全組態。

若要啟動 EC2 執行個體進入僅限 IPv6 的子網路中，您必須使用 Nitro 型執行個體。啟動僅限 IPv6 執行個體時，DHCPv6 可能無法立即向執行個體提供 IPv6 DNS 名稱伺服器。在此初始延遲期間，執行個體可能無法解析公有網域。您可以更改組態檔案並重新製作 AMI 的映像，以便在啟動時立即具有 IPv6 DNS 名稱伺服器地址。

- 安全群組：要與網路介面建立關聯的安全群組。您必須選擇與網路介面子網路處於相同 VPC 的安全群組。
- (僅限啟動範本) 自動指派公用 IP：指定您的執行個體是否會收到公用 IPv4 地址。根據預設，預設子網中的執行個體會收到公有 IPv4 地址，非預設子網中的執行個體則不會。您可以選取 Enable (啟用) 或 Disable (停用) 覆寫子網的預設設定。如需詳細資訊，請參閱 公有 IPv4 地址。
- Primary IP (主要 IP)：您子網範圍內的私有 IPv4 地址。保留空白讓 Amazon EC2 為您選擇私有 IPv4 地址。
- 次要 IP：來自子網路範圍的一或多個其他私有 IPv4 地址。選擇手動指派，然後輸入 IPv4 地址。選擇新增 IP 以新增另一個 IPv4 地址。或者，選擇自動指派，然後輸入一個值，以指示 Amazon EC2 為您選擇的 IPv4 地址數目。
- (僅限 IPv6) IPv6 IP：子網路範圍內的 IPv6 地址。選擇手動指派，然後輸入 IPv6 地址。選擇新增 IP 以新增另一個 IPv6 地址。或者，選擇自動指派，然後輸入一個值，以指示 Amazon EC2 為您選擇的 IPv6 地址數目。
- IPv4 Prefixes (IPv4 字首)：網路介面的 IPv4 字首。選擇手動指派，然後輸入 IPv4 字首。或者，選擇自動指派，然後輸入一個值，以指示 Amazon EC2 為您選擇的 IPv4 字首數目。
- IPv6 Prefixes (IPv6 字首)：網路介面的 IPv6 字首。選擇手動指派，然後輸入 IPv6 字首。或者，選擇自動指派，然後輸入一個值，以指示 Amazon EC2 為您選擇的 IPv6 字首數目。
- (雙堆疊和僅 IPv6) 指派主要 IPv6 IP：若您選取雙堆疊或僅 IPv6 子網路，請指派主要 IPv6 位址。這有助於防止中斷執行個體的流量或網路介面。若您倚賴 IPv6 位址而未變更，請啟用此選項。您稍後無法移除主要 IPv6 地址。當您啟用 IPv6 GUA 地址做為主要 IPv6 地址時，第一個 IPv6 GUA 地址會成為主要 IPv6 地址，直到執行個體終止或網路介面中斷連接為止。如果有多個與網路介面相關聯的 IPv6 位址，且您讓 Amazon EC2 指派主要 IPv6 位址，則與網路介面相關聯的第一個 IPv6 GUA 地址會成為主要 IPv6 位址。
- 終止時刪除：指出是否在刪除執行個體時刪除網路介面。
- 介面類型：網路介面類型：
  - ENA：高效能網路介面，旨在處理 TCP/IP 通訊協定的高輸送量和每秒封包速率，同時最大限度減少 CPU 使用。這是預設值。如需關於 ENA 的詳細資訊，請參閱彈性網路介面卡。
  - EFA 與 ENA：支援 ENA 和 EFA 裝置的網路介面，用於傳統 TCP/IP 型傳輸以及 SRD 型傳輸。如需 EFA 的詳細資訊，請參閱 Elastic Fabric Adapter。

  - 僅限 EFA：高效能網路介面，旨在處理 SRD 型傳輸的高輸送量、低延遲節點間通訊，同時繞過作業系統堆疊。僅限 EFA 的網路介面不支援 IP 位址。如需 EFA 的詳細資訊，請參閱 Elastic Fabric Adapter。
- Elastic Fabric Adapter：指出網路介面是否為 Elastic Fabric Adapter。如需詳細資訊，請參閱適用於 Amazon EC2 上的 AI/ML 和 HPC 工作負載的 Elastic Fabric Adapter。
- 網路卡索引：網路卡的索引。主要網路介面必須指派給網路卡索引 0。某些執行個體類型支援多個網路卡。
- ENA Express：ENA Express 採用可擴展可靠資料包 (SRD) AWS 技術。SRD 技術使用封包噴塗機制來分配負載並避免網路擁塞。啟用 ENA Express 便可支援的執行個體在可能的情況下，在一般 TCP 流量之上使用 SRD 進行通訊。除非您從清單中選取啟用或停用，否則啟動執行個體精靈或啟動範本不會包含執行個體的 ENA Express 組態。
- ENA Express UDP：如果您已啟用 ENA Express，您可以選擇性用於 UDP 流量。除非您選取啟用或停用，否則啟動執行個體精靈或啟動範本不會包含執行個體的 ENA Express 組態。

設定儲存

您選取的 AMI 包含一或多個儲存體磁碟區，包括根磁碟區。您可以指定要連接至執行個體的其他磁碟區。

(僅限啟動執行個體精靈) 您可以使用簡易或進階檢視。使用 Simple (簡單) 檢視，您可以指定磁碟區的大小和類型。若要指定所有磁碟區參數，請選擇卡片右上方的 Advanced (進階) 檢視。

使用 Advanced (進階) 檢視，您可以設定每個磁碟區，如下所示：

- Storage type (儲存類型)：選取要與您的執行個體相關聯的 Amazon EBS 磁碟區或執行個體存放區磁碟區。此清單中可用的磁碟區類型取決於您選擇的執行個體類型。如需詳細資訊，請參閱 EC 2 執行個體的執行個體儲存體暫時區塊儲存 和 Amazon EBS 磁碟區。
- Device name (設備名稱)：從磁碟區可用的設備名稱清單中選取。
- Snapshot (快照)：選取要從中還原磁碟區的快照。您可以在 Snapshot (快照) 欄位中輸入文字，搜尋可用的共享和公有快照。
- Size (GiB) (大小 (GiB))：針對 EBS 磁碟區，您可以指定儲存體大小。
- Volume Type (磁碟區類型)：針對 EBS 磁碟區，選取磁碟區類型。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 磁碟區類型。
- IOPS：如果您已選取 io1、io2 或 gp3 磁碟區類型，則可以輸入磁碟區能支援的每秒 I/O 操作數 (IOPS)。對於 io1、io2 和 gp3 磁碟區，這是必要項目。gp2、st1、sc1 或標準磁碟區不支援此項目。如果您針對啟動範本省略此參數，則必須在從啟動範本啟動執行個體時為其指定一個值。

- Delete on termination (終止時刪除)：針對 Amazon EBS 磁碟區，請選擇 Yes (是) 以執行個體終止時刪除磁碟區，或選擇 No (否) 保留磁碟區。如需詳細資訊，請參閱 在執行個體終止時保留資料。
- Encrypted (已加密)：如果執行個體類型支援 EBS 加密，您可以選擇 Yes (是) 以啟用磁碟區的加密。如果您在此區域中預設啟用加密，則會為您啟用預設加密。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 加密。
- KMS Key (KMS 金鑰)：如果您為 Encrypted (已加密) 選取了 Yes (是)，則必須選取要用來加密磁碟區的客戶受管金鑰。如果在此區域中預設啟用加密，則會為您選取預設客戶受管金鑰。您可以選取不同的金鑰，或指定您已建立之任何客戶受管金鑰的 ARN。
- 輸送量：若您已選取 gp3 磁碟區類型，則以 MiB/s 為單位輸入磁碟區可支援的輸送量。
- 磁碟區初始化速率：若您已選取快照，可以選擇指定磁碟區初始化速率 (以 MiB/s 為單位)，快照區塊將以該速率從 Amazon S3 下載至磁碟區。如需詳細資訊，請參閱使用 Amazon EBS 佈建速率進行磁碟區初始化。要使用預設初始化速率或快速快照還原 (若已針對選取的快照啟用)，請勿指定速率。
- File systems (檔案系統)：將 Amazon EFS 或 Amazon FSx 檔案系統掛載到執行個體。如需有關掛載 Amazon EFS 檔案系統的詳細資訊，請參閱 搭配 Amazon EC2 Linux 執行個體一起使用 Amazon EFS。如需有關掛載 Amazon FSx 檔案系統的詳細資訊，請參閱 使用 Amazon FSx 搭配 Amazon EC2 執行個體。

進階詳細資訊

針對 Advanced Details (進階詳細資訊)，展開此區段來檢視欄位，指定執行個體的其他參數。

- （僅限啟動執行個體精靈）網域加入目錄：選取執行個體在啟動後加入的 Directory Service 目錄（網域）。若您選取網域，您必須選取具備必要許可的 IAM 角色。如需網域加入的詳細資訊，請參閱將 Amazon EC2 Linux 執行個體無縫加入您的 AWS Managed Microsoft AD 目錄 (Linux 執行個體），以及將 Amazon EC2 Windows 執行個體無縫加入您的 AWS Managed Microsoft AD 目錄 (Windows 執行個體）。
- IAM 執行個體設定檔：選取要與執行個體建立關聯的 IAM 執行個體設定檔。這是 IAM 角色的容器。如需詳細資訊，請參閱Amazon EC2 的 IAM 的角色。
- Hostname type (主機名稱類型)：選取執行個體的客體作業系統主機名稱是否要包含資源名稱或 IP 名稱。如需詳細資訊，請參閱EC2 執行個體主機名稱與網域。
- DNS Hostname (DNS 主機名稱)：決定資源名稱或 IP 名稱的 DNS 查詢 (根據您在 Hostname type (主機名稱類型) 所選的項目) 是以 IPv4 地址 (A 記錄)、IPv6 地址 (AAAA 記錄) 還是以兩者進行回應。如需詳細資訊，請參閱EC2 執行個體主機名稱與網域。

- 執行個體自動復原：啟用時，如果系統狀態檢查失敗，則會復原您的執行個體。啟動時，針對支援的執行個體類型預設會啟用此設定。如需詳細資訊，請參閱在 Amazon EC2 執行個體上設定簡化的自動復原。
- Shutdown behavior (關機行為)：選取執行個體是否應於關機時停止或終止。如需詳細資訊，請參閱變更執行個體啟動的關機行為。
- Stop - Hibernate behavior (停止 - 休眠行為)：若要啟用休眠，請選取 Enable (啟用)。此欄位只在您的執行個體符合休眠先決條件時才可使用。如需詳細資訊，請參閱 讓 Amazon EC2 執行個體進入休眠。
- Termination protection (終止保護)：若要防止意外性的終止，請選取 Enable (啟用)。如需詳細資訊，請參閱變更執行個體終止保護。
- Stop protection (停止保護)：若要防止意外性的停止，請選擇 Enable (啟用)。如需詳細資訊，請參閱啟用停止保護。
- Detailed CloudWatch monitoring (詳細 CloudWatch 監控)：選擇 Enable (啟用) 即可開啟透過 Amazon CloudWatch 對執行個體進行詳細監控。需支付額外費用。如需詳細資訊，請參閱使用 CloudWatch 監控您的執行個體。
- Credit specification (額度規格)：選擇 Unlimited (無限制)，只要有需要，就讓應用程式爆量超過基線。此欄位僅對 T 執行個體有效。可能會收取額外費用。如需詳細資訊，請參閱爆量效能執行個體。
- 置放群組：指定要在其中啟動執行個體的置放群組。可以選取現有的置放群組或建立新的置放群組。並非所有執行個體類型都支援在置放群組中啟動執行個體。如需詳細資訊，請參閱 Amazon EC2 執行個體的置放群組。
- EBS 最佳化執行個體：針對 Amazon EBS 最佳化的執行個體使用最佳化組態堆疊，並為 Amazon EBS I/O 額外提供專屬的容量。如果執行個體類型支援此功能，請選取 Enable (啟用) 以啟用它。需支付額外費用。如需詳細資訊，請參閱the section called “EBS 最佳化”。
- 執行個體頻寬組態：您可提升聯網頻寬或是 EBS 頻寬。僅適合支援的執行個體類型。如需詳細資訊，請參閱EC2 執行個體頻寬加權組態。
- 購買選項：選擇 Spot 執行個體以 Spot 價格請求 Spot 執行個體，但是不超過隨需價格，並選擇自訂 Spot 執行個體選項變更預設的 Spot 執行個體設定。您可以設定最高價 (不建議)，並變更請求類型、請求持續時間和中斷行為。如不請求 Spot 執行個體，Amazon EC2 預設會啟動隨需執行個體。如需詳細資訊，請參閱管理您的 Spot 執行個體。
- 容量保留：指定是否要在任何開啟的容量保留 (Open (開啟))、特定容量保留 (Target by ID (按 ID 分類的目標)) 或容量保留群組 (Target by group (按群組分類的目標)) 中啟動執行個體。若要指定不應使用容量保留，請選擇 None (無)。如需詳細資訊，請參閱 使用現有的 容量預留 啟動執行個體。

- Tenancy (租用)：選擇在共享硬體 (Shared (共享))、隔離的、專用硬體 (Dedicated (專用)) 或專用執行個體 (Dedicated host (專用執行個體)) 上執行您的執行個體。如果您選擇將執行個體啟動至專用執行個體，則可以指定是否將執行個體啟動至主機資源群組，或是以特定專用執行個體為目標。可能需支付額外費用。如需詳細資訊，請參閱 Amazon EC2 專用執行個體 及 Amazon EC2 專用執行個體。
- RAM 磁碟 ID：(僅適用於半虛擬 (PV) AMIs) 選取執行個體的 RAM 磁碟。若您已選取核心，則可能需要選取具有可支援之驅動程式的特定 RAM 磁碟。
- 核心 ID：(僅適用於全虛擬 (PV) AMIs) 選取執行個體的核心。
- Nitro Enclave：可讓您從 Amazon EC2 執行個體建立稱為隔離區的獨立執行環境。選取啟用以啟用 AWS Nitro Enclaves 的執行個體。如需詳細資訊，請參閱《AWS Nitro Enclaves 使用者指南》中的什麼是 AWS Nitro Enclaves？。
- 授權組態：您可以根據指定的授權配置啟動執行個體，以追蹤您的授權使用情況。如需詳細資訊，請參閱「AWS License Manager 使用者指南」中的建立授權組態。
- 指定 CPU 選項：在啟動執行個體精靈中，只有在選取的執行個體類型支援指定 CPU 選項時，才會顯示此欄位。選擇指定 CPU 選項在啟動期間指定自訂數量的 vCPU。設定 CPU 核心數量及每個核心的執行緒數量。如需詳細資訊，請參閱Amazon EC2 執行個體的 CPU 選項。
- 可存取的中繼資料：您可以啟用或停用對執行個體中繼資料服務 (IMDS) 的存取。如需詳細資訊，請參閱設定新執行個體的執行個體中繼資料選項。
- 中繼資料 IPv6 端點：您可以讓執行個體使用 IMDS IPv6 地址`[fd00:ec2::254]`來擷取執行個體中繼資料。僅在您將 Nitro 型執行個體啟動為 IPv6 支援的子網路 (僅支援雙重堆疊或 IPv6) 時，才能使用此選項。如需有關擷取執行個體中繼資料的詳細資訊，請參閱 存取 EC2 執行個體的執行個體中繼資料。
- 中繼資料版本：如果啟用對 IMDS 的存取，則可以選擇請求執行處理中繼資料時請求使用 執行個體中繼資料服務第 2 版。如需詳細資訊，請參閱設定新執行個體的執行個體中繼資料選項。
- 中繼資料回應躍點限制：如果啟用 IMDS，可以為中繼資料字符設定允許的網路躍點數量。如需詳細資訊，請參閱設定新執行個體的執行個體中繼資料選項。
- Allow tags in metadata (允許中繼資料中的標籤)：如果選取 Enable (啟用)，則執行個體將允許從其中繼資料存取其所有標籤。如果未指定任何值，則預設情況下，不允許存取執行個體中繼資料中的標籤。如需詳細資訊，請參閱在執行個體中繼資料中啟用標籤存取權。
- User data (使用者資料)：您可以指定使用者資料，以在啟動期間設定執行個體或執行組態指令碼。如需有關 Linux 執行個體的詳細資訊，請參閱 當使用使用者資料輸入啟動 EC2 執行個體時執行命令。如需有關 Windows 執行個體使用者資料的詳細資訊，請參閱 Amazon EC2 如何處理 Windows 執行個體的使用者資料。

摘要

使用 Summary (摘要) 面板，以指定要啟動的執行個體數量、檢閱您的執行個體組態，以及啟動您的執行個體。

- Number of instances (執行個體數目)：輸入要啟動的執行個體數目。所有執行個體都會以相同的組態啟動。

> **Tip**
>
> 為了確保執行個體能更快啟動，請將大型請求分成較小的批次。例如，分別建立五個啟動請求，每個請求各啟動 100 個執行個體，而不要以單一請求啟動 500 個執行個體。

- (選用) 若指定了多個執行個體，為了確保足以處理您應用程式需求的正確執行個體數目，您可以選擇 consider EC2 Auto Scaling (考慮 EC2 Auto Scaling) 以建立啟動範本及 Auto Scaling 群組。Auto Scaling 會根據您的規格擴展群組中的執行個體數目。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南。

> **Note**
>
> 如果 Amazon EC2 Auto Scaling 將 Auto Scaling 群組中的執行個體標記為「狀況不良」，將自動排程更換該執行個體，並終止該執行個體，啟動另一個執行個體，而且您會遺失原始執行個體上的資料。如果您停止或重新開機執行個體，或另一個事件將此執行個體標記為「狀況不良」，則執行個體會標記為「狀況不良」。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的針對 Auto Scaling 群組中的執行個體進行運作狀態檢查。

- 檢閱執行個體的詳細資訊，並進行任何必要的變更。您可以直接導覽至某個區段，方法是在 Summary (摘要) 面板中選擇其連結。
- 當您準備好啟動執行個體時，請選擇 Launch instance (啟動執行個體)。

使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體

您可以使用 Amazon EC2 主控台中的啟動執行個體精靈，啟動 Amazon EC2 執行個體。該精靈提供啟動參數的預設值，可接受或修改以滿足您的需求。唯一未指定的參數是金鑰對。如果您選擇接受預設值，則可以透過僅選取金鑰對來快速啟動執行個體。

> **Important**
>
> 執行個體處於 `running` 狀態時，即使仍為閒置狀態，也會產生執行個體費用。但如果您符合免費方案的資格，可能不會產生任何費用。如需詳細資訊，請參閱 追蹤 Amazon EC2 的免費方案用量。

如需啟動執行個體精靈中每個參數的說明，請參閱 Amazon EC2 執行個體組態參數的參考。

主題

- 快速啟動執行個體
- 使用定義的參數啟動執行個體

快速啟動執行個體

若要快速設定執行個體以進行測試，請遵循下列步驟。您將選取作業系統和金鑰對，並接受預設值。除了金鑰對之外，啟動執行個體精靈還會提供所有參數的預設值。您可以接受任何或全部預設值，或者透過為每個參數指定自己的值來設定執行個體。

如需啟動執行個體精靈中每個參數的說明，請參閱 Amazon EC2 執行個體組態參數的參考。

使用啟動執行個體精靈快速啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面頂端的導覽列中，會顯示目前的 AWS 區域 （例如美國東部 （俄亥俄））。如果需要，選取要在其中啟動執行個體的不同區域。
3. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. (選用) 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入執行個體的描述性名稱。
5. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，選擇 Quick Start (快速入門)，然後選擇執行個體的作業系統。
6. 在 Key pair (login) (金鑰對 (登入)) 下，針對 Key pair name (金鑰對名稱)，選擇現有的金鑰對或建立新的金鑰對。
7. 在 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

使用定義的參數啟動執行個體

如果啟動的是要用於生產中的執行個體，則需要設定執行個體以符合您的需求。如需啟動執行個體精靈中每個參數的說明，請參閱 Amazon EC2 執行個體組態參數的參考。

使用啟動執行個體精靈定義所有啟動參數，藉此啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面頂端的導覽列中，會顯示目前的 AWS 區域（例如美國東部（俄亥俄））。如果需要，選取要在其中啟動執行個體的不同區域。
3. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. (選用) 在名稱和標籤下方的名稱中，輸入執行個體的描述性名稱，以便輕鬆進行追蹤。

   執行個體名稱是一個標籤，其中鍵是 Name (名稱)，而值是您指定的名稱。
5. 在應用程式和作業系統映像 (Amazon Machine Image) 下方，依序選擇執行個體的作業系統和 AMI。

   AMI 是一種包含啟動執行個體所需作業系統和軟體的範本。
6. 在 Instance type (執行個體類型)，選擇執行個體類型。

   執行個體類型會決定硬體組態 (CPU、記憶體、儲存體和聯網容量)，以及執行個體所使用主機電腦的大小。

   如果您不確定要選擇哪個執行個體類型，可以執行下列動作：

   - 選擇比較執行個體類型，依下列屬性比較不同的執行個體類型：vCPU 數目、架構、記憶體數量 (GiB)、儲存數量 (GB)、儲存類型，以及網路效能。
   - 選擇取得建議，以從 EC2 執行個體類型搜尋工具取得執行個體類型的指導和建議。如需詳細資訊，請參閱從 EC2 執行個體類型搜尋工具取得建議。

   > **Note**
   >
   > 視您建立帳戶的時間而定，您可在免費方案下免費使用執行個體類型。這些執行個體類型會標記為符合免費方案資格。
   > 如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。

如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。
如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。

7. 在 Key pair (login) (金鑰對 (登入)) 下，針對 Key pair name (金鑰對名稱)，選擇現有的金鑰對或建立新的金鑰對。如果您在連線到執行個體時不需要金鑰對，您可以選擇不使用金鑰對繼續 (不建議)。
8. 在網路設定下方，如果啟動測試執行個體，您可以保留預設值。如果啟動生產執行個體，最佳實務是使用您定義的網路設定和安全群組，以控制傳入和傳出執行個體的流量。
9. 在設定儲存體下方，您可以保留預設值或指定其他儲存體。您選取的 AMI 包含一或多個儲存體磁碟區，包括根磁碟區。您可以指定要連接至執行個體的其他磁碟區。

   您可以使用 Simple (簡單) 或 Advanced (進階) 檢視。使用 Simple (簡單) 檢視，您可以指定磁碟區的大小和類型。若要指定所有磁碟區參數，請選擇卡片右上方的 Advanced (進階) 檢視。
10. 針對進階詳細資訊，展開此區段來檢視欄位並指定執行個體的任何其他參數。
11. 在摘要面板中，您可以執行以下操作：

    a. 指定要啟動的執行個體數目。
    b. 檢閱您的執行個體組態，並選擇其連結直接導覽至區段。
    c. 當您準備好啟動執行個體時，請選擇 Launch instance (啟動執行個體)。

    如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。
12. (選用) 您可以建立執行個體的帳單提醒。在確認畫面上的 Next Steps (後續步驟) 下方，選擇 Create billing alerts (建立帳單提醒)，然後遵循指示進行。您還可以在啟動執行個體後建立帳單提醒。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的建立帳單警示以監控預估 AWS 費用。

使用啟動範本啟動 EC2 執行個體

Amazon EC2 啟動範本會存放執行個體啟動參數，如此您就不必在每次啟動執行個體時指定參數。

有數個執行個體啟動服務可以在啟動執行個體時選擇性地使用啟動範本，而對於如 EC2 機群的其他服務，則無法啟動執行個體，除非使用了啟動範本。此主題說明如何在使用 EC2 啟動執行個體精靈、Amazon EC2 Auto Scaling、EC2 機群和 Spot 機群啟動執行個體時使用啟動範本。

如需啟動範本 (包括如何建立啟動範本) 的詳細資訊，請參閱 將執行個體啟動參數存放在 Amazon EC2 啟動範本中。

主題

- 使用啟動範本啟動 Amazon EC2 執行個體
- 使用啟動範本在 Amazon EC2 Auto Scaling 群組中啟動執行個體
- 使用啟動範本啟動 EC2 機群
- 使用啟動範本啟動 Spot 機群

使用啟動範本啟動 Amazon EC2 執行個體

您可以使用啟動範本中所含的參數來啟動 Amazon EC2 執行個體。在選取啟動範本之後，但在啟動執行個體之前，您可以修改啟動參數。

使用啟動範本所啟動的執行個體會自動獲指派兩個具有索引鍵 `aws:ec2launchtemplate:id` 和 `aws:ec2launchtemplate:version` 的標籤。您無法移除或編輯這些標籤。

Console

使用啟動範本啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 使用以下其中一個選項來選取啟動範本：
   - 從 Amazon EC2 主控台儀表板中，選擇啟動執行個體旁的向下箭頭，選擇從範本啟動執行個體，然後針對來源範本，選取啟動範本。
   - 在導覽窗格中，選擇啟動範本，選取啟動範本，然後選擇動作、從範本啟動執行個體。
3. 對於 Source template version (來源範本版本)，請選取要使用的啟動範本版本。
4. (選用) 您可以修改任何啟動參數的值。如果您不修改值，則會使用啟動範本所定義的值。如果未在啟動範本中指定值，則會使用參數的預設值。
5. 在摘要面板中，針對執行個體的數量，指定要啟動的執行個體數量。
6. 選擇啟動執行個體。

   如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

AWS CLI

從啟動範本啟動執行個體

- 使用 run-instances 命令，並指定 `--launch-template` 參數。選擇指定要使用的啟動範本版本。如果您未指定版本，即使用預設版本。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- 若要覆寫啟動範本參數，請在 run-instances 命令中指定參數。下列範例會覆寫啟動範本中指定的執行個體類型 (如果有的話)。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --instance-type t2.small
```

- 如果您指定屬複雜結構一部分的巢狀參數，則會使用啟動範本中所指定的複雜結構以及您指定的其他巢狀參數，來啟動執行個體。

  在下列範例中，會使用 *Owner*=*TeamA* 標籤以及啟動範本中指定的其他標籤，來啟動執行個體。如果啟動範本的現有標籤包含 *Owner* 索引鍵，則會將值取代為 *TeamA*。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

  在下列範例中，會使用設備名稱為 */dev/xvdb* 的磁碟區以及啟動範本中指定的其他區塊型設備映射，來啟動執行個體。如果啟動範本具有針對 */dev/xvdb* 所定義的現有磁碟區，則會將其值取代為指定的值。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --block-device-mappings "DeviceName=/dev/
xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

PowerShell

使用 從啟動範本啟動執行個體 AWS Tools for PowerShell

- 使用 New-EC2Instance 命令，並指定 -LaunchTemplate 參數。選擇指定要使用的啟動範本版本。如果您未指定版本，即使用預設版本。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- 若要覆寫啟動範本參數，請在 New-EC2Instance 命令中指定參數。下列範例會覆寫啟動範本中指定的執行個體類型 (如果有的話)。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- 如果您指定屬複雜結構一部分的巢狀參數，則會使用啟動範本中所指定的複雜結構以及您指定的其他巢狀參數，來啟動執行個體。

  在下列範例中，會使用 *Owner*=*TeamA* 標籤以及啟動範本中指定的其他標籤，來啟動執行個體。如果啟動範本的現有標籤包含 *Owner* 索引鍵，則會將值取代為 *TeamA*。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
```

```
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
        }
) `
    -TagSpecification (
        New-Object -TypeName Amazon.EC2.Model.TagSpecification -Property @{
            ResourceType = 'instance';
            Tags         = @(
                @{key = "Owner"; value = "TeamA" },
                @{key = "Department"; value = "Operations" }
            )
        }
)
```

在下列範例中，會使用設備名稱為 */dev/xvdb* 的磁碟區以及啟動範本中指定的其他區塊型設備映射，來啟動執行個體。如果啟動範本具有針對 */dev/xvdb* 所定義的現有磁碟區，則會將其值取代為指定的值。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
) `
    -BlockDeviceMapping  (
        New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping -Property @{
            DeviceName = '/dev/xvdb';
            EBS        = (
                New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property @{
                    VolumeSize = 25;
                    VolumeType = 'gp3'
                }
            )
        }
)
```

如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

使用啟動範本在 Amazon EC2 Auto Scaling 群組中啟動執行個體

您可以建立 Auto Scaling 群組，並指定要用於群組的啟動範本。Amazon EC2 Auto Scaling 在 Auto Scaling 群組中啟動執行個體時，會使用相關聯之啟動範本中定義的啟動參數。

在您可以使用啟動範本建立 Auto Scaling 群組之前，您必須先建立啟動範本，其中包含在 Auto Scaling 群組中啟動執行個體所需的參數。有些參數是必要的，例如 AMI 的 ID，而有些參數無法與 Auto Scaling 群組搭配使用。主控台提供指引，協助您建立可搭配 Amazon EC2 Auto Scaling 使用的範本。

使用主控台搭配啟動範本建立 Auto Scaling 群組

- 如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用啟動範本建立 Auto Scaling 群組。

使用 建立或更新具有啟動範本的 Auto Scaling 群組 AWS CLI

- 使用 create-auto-scaling-group 或 update-auto-scaling-group 命令，並指定 `--launch-template` 參數。

如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的下列主題。

- 為 Auto Scaling 群組建立啟動範本
- 使用進階設定建立啟動範本
- 建立及管理 AWS Command Line Interface (AWS CLI) 啟動範本的範例 – 提供顯示如何使用各種參數組合來建立啟動範本的範例。
- 使用啟動範本建立 Auto Scaling 群組
- 更新 Auto Scaling 群組

使用啟動範本啟動 EC2 機群

建立 EC2 機群請求時，啟動範本是必要項目。當 Amazon EC2 滿足 EC2 Fleet 請求時，會使用相關聯之啟動範本中定義的啟動參數。您可以覆寫啟動範本中指定的部分參數。如需詳細資訊，請參閱建立 EC2 Fleet。

使用 建立具有啟動範本的 EC2 機群 AWS CLI

- 使用 create-fleet 命令。使用 `--launch-template-configs` 參數指定啟動範本以及啟動範本的任何覆寫。

使用啟動範本啟動 Spot 機群

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

建立 Spot 機群請求時，啟動範本是選用項目。如果您不使用啟動範本，可以手動指定啟動參數。若您使用啟動範本，則當 Amazon EC2 滿足 Spot 機群請求時，會使用相關聯啟動範本中定義的啟動參數。您可以覆寫啟動範本中指定的部分參數。如需詳細資訊，請參閱建立 Spot 機群。

使用啟動範本來建立 Spot 機群請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選擇 Request Spot Instances (要求 Spot 執行個體)。
4. 在 Launch parameters (啟動參數) 下，選擇 Use a launch template (使用啟動範本)。
5. 針對 Launch template (啟動範本)，選擇啟動範本，然後從右邊的欄位選擇啟動範本版本。
6. 透過在此畫面上選取不同的選項，設定 Spot 機群。如需選項的詳細資訊，請參閱 使用已定義的參數建立 Spot 機群請求。
7. 當您準備好建立 Spot 機群時，選擇 Launch (啟動)。

使用啟動範本來建立 Spot 機群請求

- 使用 request-spot-fleet 命令。使用 `LaunchTemplateConfigs` 參數指定啟動範本以及啟動範本的任何覆寫。

使用現有執行個體的詳細資訊來啟動 EC2 執行個體

Amazon EC2 主控台提供 Launch more like this (啟動更多相似執行個體) 選項，可讓您使用目前的執行個體做為啟動其他執行個體的基礎。此選項會自動使用選取執行個體的特定組態詳細資訊，填入 Amazon EC2 啟動執行個體精靈。

考量事項

- 我們不會複製您的執行個體；我們只會複寫部分的組態詳細資訊。若要建立您執行個體的複本，首先請從它建立 AMI，然後從 AMI 啟動更多執行個體。建立啟動範本，以確保您使用相同的啟動詳細資訊來啟動執行個體。
- 目前的執行個體必須處於 `running` 狀態。

已複製的詳細資訊

下列組態詳細資訊會從選取的執行個體複製到啟動執行個體精靈：

- AMI ID
- 執行個體類型
- 可用區域，或是選取執行個體所在的 VPC 及子網
- 公有 IPv4 地址。若選取的執行個體目前具有公有 IPv4 地址，新的執行個體便會收到公有 IPv4 地址，無論選取執行個體的預設公用 IPv4 地址設定為何。如需公有 IPv4 地址的詳細資訊，請參閱公有 IPv4 地址。
- 置放群組 (如適用)
- 與執行個體相關聯的 IAM 角色 (如適用)
- 關機行為設定 (停止或終止)
- 終止保護設定 (true 或 false)
- CloudWatch 監控 (啟用或停用)
- Amazon EBS 最佳化設定 (true 或 false)
- 租用設定 (若在 VPC 中啟動的話 (共享或專用))
- 核心 ID 和 RAM 磁碟區 ID (如適用)
- 使用者資料 (若指定)
- 與執行個體相關聯的標籤 (如適用)

- 與執行個體相關聯的安全群組
- [Windows 執行個體] 關聯資訊。若選取的執行個體與組態檔相關聯，則相同的檔案便會自動與新的執行個體相關聯。若組態檔包含加入的網域組態，新的執行個體便會加入相同網域。如需加入網域的詳細資訊，請參閱《 AWS Directory Service 管理指南》中的將 Windows EC2 執行個體無縫加入 AWS Managed Microsoft AD Active Directory。

未複製的詳細資訊

下列組態詳細資訊不是從您選取的執行個體複製的。相反地，精靈會套用其預設設定或行為：

- 網路介面數 – 預設為一個網路介面，即主要網路介面 (eth0)。
- 儲存體 – 預設儲存體組態會由 AMI 和執行個體類型判定。

依據現有執行個體的操作方式，啟動更多執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取動作、映像和範本、啟動更多相似執行個體。
4. 啟動執行個體精靈隨即開啟。透過在此畫面上選取不同的選項，對執行個體組態進行任何必要的變更。

   當您準備好啟動執行個體時，請選擇 Launch instance (啟動執行個體)。
5. 如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體

您可以使用 Amazon EC2 主控台或命令列工具來訂閱 AWS Marketplace AMI 並從中啟動執行個體。如需 AWS Marketplace AMIs的詳細資訊，請參閱 AWS Marketplace 適用於 Amazon EC2 執行個體的中的付費 AMIs。

若要在啟動後取消 AMI 訂閱，您必須先終止所有從 AMI 啟動的執行個體。如需詳細資訊，請參閱管理您的 AWS Marketplace 訂閱。

使用 Amazon EC2 主控台從 AWS Marketplace AMI 啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
3. (選用) 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入執行個體的描述性名稱。
4. 在應用程式和作業系統映像 (Amazon Machine Image) 中，選擇瀏覽更多 AMI，然後選擇 AWS Marketplace AMI 索引標籤。透過瀏覽類別或使用搜尋功能，來尋找適合的 AMI。若要選取產品，請選取 Select (選取)。
5. 將打開一個視窗，其中顯示所選產品的概述。您可以檢視定價資訊，以及廠商已提供的任何其他資訊。當您準備好時，請選擇訂閱並啟動。這將立即啟動您的訂閱。訂閱期間，您可以繼續此程序中的步驟來設定執行個體。如果您的信用卡詳細資訊有任何問題，系統會要求您更新帳戶詳細資訊。

> **Note**
>
> 在您使用 AMI 啟動執行個體之前，不會向您收取使用產品的費用。當您選取執行個體類型時，請記下每個受支援執行個體類型的定價。可能也需支付產品的額外稅金。

6. 針對 Instance type (執行個體類型)，選取執行個體的執行個體類型。執行個體類型定義執行個體的硬體組態和大小。
7. 在 Key pair (login) (金鑰對 (登入)) 下，針對 Key pair name (金鑰對名稱)，選擇現有的金鑰對或建立新的金鑰對。
8. 在網路設定、防火牆 (安全群組) 中，記下已根據廠商的產品規格所建立的新安全群組。安全群組可能會包括規則，以允許 Linux 上之 SSH (連接埠 22) 或 Windows 上之 RDP (連接埠 3389) 的所有 IPv4 地址 (`0.0.0.0/0`) 存取。建議您調整這些規則，改為僅允許特定地址或某範圍的地址透過這些連接埠存取您的執行個體。
9. 您可以使用畫面上的其他欄位來設定執行個體、新增儲存以及新增標籤。如需有關您可設定之不同選項的資訊，請參閱 Amazon EC2 執行個體組態參數的參考。
10. 在 Summary (摘要) 面板的 Software Image (AMI) (軟體映像 (AMI)) 中，檢查即將要從中啟動執行個體之 AMI 的詳細資訊。同時檢查您指定的其他組態詳細資訊。當您準備好啟動執行個體時，請選擇 Launch instance (啟動執行個體)。
11. 根據您已訂閱的產品，執行個體可能需要幾分鐘以上的時間啟動。如果您的信用卡詳細資訊有任何問題，系統會要求您更新帳戶詳細資訊。顯示啟動確認頁面時，請選擇 View all instances (檢視所有執行個體) 以前往 Instances (執行個體) 頁面。

> **Note**
>
> 只要您的執行個體處於 `running` 狀態，就會向您收取訂閱費用，即使它處於閒置狀態。如果您的執行個體已停止，仍可能會向您收取儲存體的費用。

12. 當您的執行個體處於 `running` 狀態時，即可連線。若要執行此操作，請選取清單中的執行個體，然後選取 Connect (連線)，並選取連線選項。如需連線至執行個體的詳細資訊，請參閱連線至 EC2 執行個體。

    > **Important**
    >
    > 請仔細檢查廠商使用說明，因為您可能需要使用特定使用者名稱連線到執行個體。如需有關存取訂閱的資訊，請參閱 管理您的 AWS Marketplace 訂閱。

13. 如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

使用命令列工具從 AWS Marketplace AMI 啟動執行個體

若要使用命令列工具從 AWS Marketplace 產品啟動執行個體，請先確定您已訂閱產品。您接著可以使用下列方法以產品的 AMI ID 來啟動執行個體：

方法	文件
AWS CLI	使用 run-instances 命令，或者參閱下列主題，查看詳細資訊：在 AWS Command Line Interface 使用者指南中啟動執行個體。
AWS Tools for Windows PowerShell	使用 New-EC2Instance 命令，或者參閱下列主題，查看詳細資訊：使用 Windows PowerShell 啟動 Amazon EC2 執行個體
查詢 API	使用 RunInstances 請求。

連線至 EC2 執行個體

您的 Amazon EC2 執行個體是 AWS 雲端中的虛擬伺服器。若要登入執行個體，您必須建立執行個體的連線。連線到執行個體的方式取決於執行個體的作業系統，以及您用來連線到執行個體的電腦作業系統。下表詳細說明每種連線方式的要求。

連線選項	執行個體的作業系統	傳入流量規則	IAM 許可	執行個體設定檔角色	執行個體上的軟體	連線系統上的軟體	金鑰對
SSH 用戶端	Linux	是	否	否	否	是	是
EC2 執行個體連線	Linux	是	是	否	是 [1]	否	否
PuTTY	Linux	是	否	否	否	是	是
RDP 用戶端	Windows	是	否	否	否	是	是 [2]
Fleet Manager	Windows	否	是	是	是 [1]	否	是
工作階段管理員	Linux、Windows	否	是	是	是 [1]	否	否
EC2 Instance Connect 端點	Linux、Windows	是	是	否	否	否	否 [3]

[1]僅在特定 AMI 預先安裝所需軟體。您可以在支援的作業系統視需要手動安裝所需軟體。

[2]只有在您使用本機管理員使用者帳戶隨機產生的密碼時，才需要金鑰對。

[3] 若使用 SSH 連線方法，需要金鑰對。

如需詳細資訊，請參閱您要使用的連線選項文件。

連線選項

- 使用 SSH 用戶端連接至 Linux 執行個體
- 使用 PuTTY 連線至您的 Linux 執行個體
- 使用 RDP 用戶端連線至您的 Windows 執行個體。

- 使用 Fleet Manager 連線到 Windows 執行個體
- 使用工作階段管理員進行連接
- 使用公有 IP 與 EC2 Instance Connect 建立連線
- 使用私有 IP 與 EC2 Instance Connect 端點建立連線

一般連線先決條件

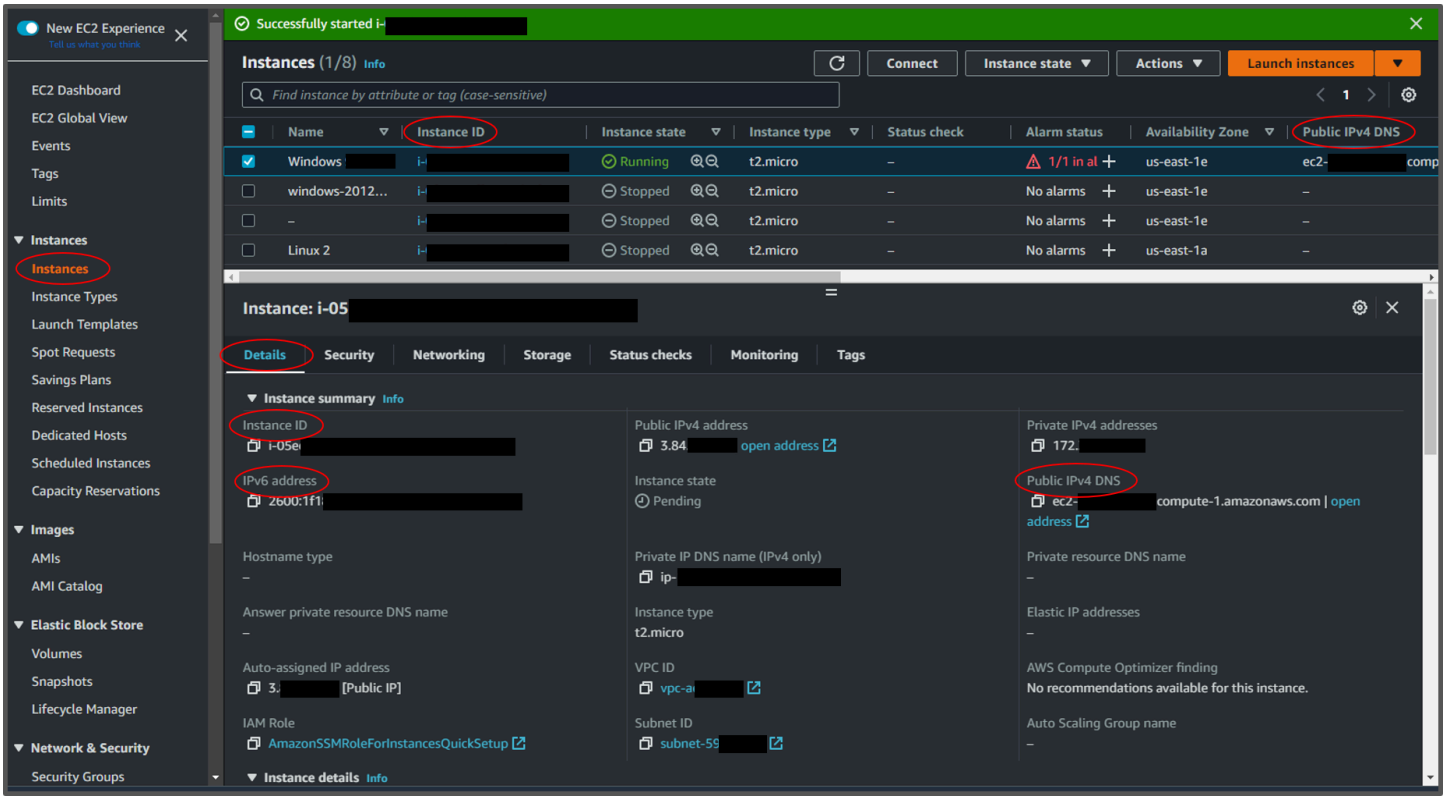
以下是連線至執行個體的一般先決條件。請注意，您選擇的連線選項可能還有其他特定的先決條件。

一般先決條件

- 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
- 取得必要的執行個體詳細資訊.
- 找出私有金鑰並設定許可.
- (選用) 取得執行個體指紋.

取得必要的執行個體詳細資訊

若準備連線至執行個體，請從 Amazon EC2 主控台或使用命令列取得下列資訊。



- 取得執行個體的公有 DNS 名稱。

  您可以從 Amazon EC2 主控台取得執行個體的公有 DNS。檢查執行個體窗格的公有 IPv4 DNS 資料欄。如果此資料欄已隱藏，請選擇畫面右上角的設定圖示 ( )，並選取公有 IPv4 DNS。您也可以在執行個體窗格的執行個體資訊區段中找到公有 DNS。當您在 Amazon EC2 主控台的執行個體窗格中選取執行個體時，該執行個體的相關資訊將顯示在頁面的下半部。在詳細資訊索引標籤下，查找公有 IPv4 DNS。

  如果您願意的話，可以使用 describe-instances (AWS CLI) 或 Get-EC2Instance (AWS Tools for Windows PowerShell) 命令。

  如果未顯示公有 IPv4 DNS，則請確認執行個體狀態為執行中，且您尚未在私有子網路中啟動執行個體。如果您使用啟動執行個體精靈啟動執行個體，則可能已編輯網路設定下的自動指派公有 IP 欄位，並將值變更為停用。如果停用自動指派公有 IP 選項，則執行個體在啟動時不會被指派公有 IP 地址。

- (僅限 IPv6 執行個體) 取得執行個體的 IPv6 地址。

  如果您已將 IPv6 地址指派給執行個體，則您可使用其 IPv6 地址連線至執行個體，而不必使用公有 IPv4 地址或公有 IPv4 DNS 主機名稱。您的本機電腦必須擁有 IPv6 地址，且必須設定以使用

IPv6。您可以從 Amazon EC2 主控台取得執行個體的 IPv6 地址。檢查執行個體窗格的 IPv6 IP 資料欄。或者，您可以在執行個體資訊區段中找到 IPv6 地址。當您在 Amazon EC2 主控台的執行個體窗格中選取執行個體時，該執行個體的相關資訊將顯示在頁面的下半部。在詳細資訊索引標籤下，查找 IPv6 地址。

如果您願意的話，可以使用 describe-instances (AWS CLI) 或 Get-EC2Instance (AWS Tools for Windows PowerShell) 命令。如需有關 IPv6 的詳細資訊，請參閱 IPv6 地址。

- (Linux 執行個體) 取得執行個體的使用者名稱。

  使用您的使用者帳戶的使用者名稱或用於啟動執行個體的 AMI 預設使用者名稱，可連線至執行個體。

  - 取得使用者帳戶的使用者名稱。

    如需如何建立使用者帳戶的詳細資訊，請參閱在您的 Amazon EC2 Linux 執行個體上管理系統使用者。

  - 取得您用於啟動執行個體的 AMI 的預設使用者名稱。
    - Amazon Linux – `ec2-user`
    - CentOS – `centos` 或 `ec2-user`
    - Debian – `admin`
    - Fedora – `fedora` 或 `ec2-user`
    - FreeBSD – `ec2-user`
    - RHEL – `ec2-user` 或 `root`
    - SUSE – `ec2-user` 或 `root`
    - Ubuntu – `ubuntu`
    - Oracle – `ec2-user`
    - Bitnami – `bitnami`
    - Rocky Linux – `rocky`
    - 其它 – 檢查 AMI 提供者

找出私有金鑰並設定許可

您必須知道私有金鑰檔案的位置，才能使用 SSH 或使用 RDP 的 Windows 執行個體與 Linux 執行個體進行初始連線。若為 SSH 連線，則必須設定檔案許可，以便只有您可以讀取金鑰。

如需有關使用 Amazon EC2 時金鑰對如何運作的資訊，請參閱 Amazon EC2 金鑰對和 Amazon EC2 執行個體。

- 找出私有金鑰。

  取得當您啟動執行個體時所指定的金鑰對的 .pem 檔案在您電腦中的位置的完整路徑。如需詳細資訊，請參閱the section called “識別啟動時指定的公有金鑰”。

  如果您找不到私有金鑰檔案，請參閱 我的私有金鑰遺失。如何連線到我的執行個體？

  (Linux 執行個體) 如果使用 Putty 連線至執行個體，且需要將 .pem 檔案轉換為 .ppk，請參閱 使用 PuTTYgen 轉換私有金鑰。

- (Linux 執行個體) 設定私有金鑰的許可，確保只有您可以讀取。
  - 從 macOS 或 Linux 進行連線

    如果您計劃在 macOS 或 Linux 電腦上使用 SSH 用戶端連線至 Linux 執行個體，請使用下列命令設定私有金鑰檔案的許可，以便只有您能夠讀取該檔案。

    ```
    chmod 400 key-pair-name.pem
    ```

    如果您未設定這些許可，則無法使用此金鑰對來連接至您的執行個體。如需詳細資訊，請參閱錯誤：未受保護的私有金鑰檔案。

  - 從 Windows 進行連線

    開啟檔案總管並開啟.pem檔案的內容（按一下滑鼠右鍵）選單。選取屬性 > 安全性索引標籤，然後選擇進階。選擇停用繼承。移除目前使用者以外的所有使用者存取權。

(選用) 取得執行個體指紋

為了保護自己免受中間人攻擊，您可以透過驗證顯示的指紋來驗證即將連線的執行個體真實性。如果已從第三方提供的公有 AMI 啟動您的執行個體，則驗證指紋會很有幫助。

任務概觀

首先，從執行個體取得執行個體指紋。然後，當您連線到執行個體並收到提示來驗證指紋時，請將您在此程序取得的指紋與顯示的指紋進行比較。如果這些指紋不符，可能有人嘗試發動中間人攻擊。如果相符，您就能很有信心地連線至您的執行個體。

取得執行個體指紋的先決條件

- 執行個體必不得處於 pending 狀態。只有在執行個體的第一次開機完成後，才能使用指紋。
- 您必須是執行個體擁有者才能取得主控台輸出。
- 有多種方式可取得執行個體指紋。若要使用 AWS CLI，必須在本機電腦安裝。如需安裝 的資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的 入門 AWS CLI。

取得執行個體指紋

在步驟 1，您會取得主控台輸出，其中包含執行個體指紋。在步驟 2，您可以在主控台輸出找到執行個體指紋。

1. 使用下列方法之一取得主控台輸出。

   Console

   1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
   2. 從左側導覽，選擇執行個體。
   3. 選取執行個體，然後再依序選擇動作、監控和疑難排解和取得系統日誌。

   AWS CLI

   在本機電腦上 (而非所連線的執行個體)，如下所示使用 get-console-output 命令來取得執行個體的指紋。如果輸出很大，可以將輸出傳送至文字檔案，這可能更易於讀取。

   ```
   aws ec2 get-console-output \
       --instance-id i-1234567890abcdef0 \
       --query Output \
       --output text > temp.txt
   ```

   PowerShell

   於本機電腦，使用下面的 Get-EC2ConsoleOutput cmdlet。

   ```
   $encodedOutput = (Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0).Output
   [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($encodedOutput
   ```

2. 在主控台輸出，尋找位於 BEGIN SSH HOST KEY FINGERPRINTS 下的執行個體 (主機) 指紋。可能會有數個執行個體指紋。當連線到執行個體時，它只會顯示其中一個指紋。

具體的輸出可能因作業系統、AMI 版本以及是否使用 AWS 建立的金鑰對而有所不同。以下為範例輸出。

```
ec2:#############################################################
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 256 SHA256:l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY no comment (ECDSA)
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJOIdHG52dFi66EEfQ no comment (ED25519)
ec2: 2048 SHA256:L8l6pepcA7iqW/jBecQjVZClUrKY+o2cHLI0iHerbVc no comment (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #############################################################
```

> **Note**
>
> 您會在連線至執行個體時參考此指紋。

使用 SSH 連線至您的 Linux 執行個體

有多種方法可使用 SSH 連線至 Linux 執行個體。其中的某些方法會因您連線的來源本機電腦作業系統而有所不同。其他方法以瀏覽器為基礎，例如 EC2 Instance Connect 或 AWS Systems Manager Session Manager，並且可以從任何電腦使用。您可以使用 SSH 連線至 Linux 執行個體並執行命令，或在本機電腦與執行個體之間使用 SSH 傳輸檔案。

使用 SSH 連線到您的 Linux 執行個體之前，請先完成下列先決條件：

- 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
- 請確保與執行個體關聯的安全群組允許來自您的 IP 地址的傳入 SSH 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。
- 取得必要的執行個體詳細資訊.
- 找出私有金鑰並設定許可.
- (選用) 取得執行個體指紋.

然後，請選擇以下其中一個選項，以使用 SSH 連線至 Linux 執行個體。

- 使用 SSH 用戶端連線
- 使用 PuTTY 連線

- 使用 SCP 傳輸檔案

如果您無法連線至執行個體，且需要協助進行疑難排解，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

使用 SSH 用戶端連接至 Linux 執行個體

您可以使用 Secure Shell (SSH) 從本機電腦連線至 Linux 執行個體。如需關於其他選項的詳細資訊，請參閱 連線至 EC2 執行個體。

> **Note**
>
> 如果您在嘗試連線至執行個體時收到錯誤訊息，請確認執行個體符合所有 SSH 連線先決條件。如果其符合所有先決條件，但您仍然無法連線至 Linux 執行個體，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

目錄

- SSH 連線先決條件
- 使用 SSH 用戶端連接至 Linux 執行個體

SSH 連線先決條件

在使用 SSH 連線到 Linux 執行個體之前，請完成下列任務。

完成一般先決條件。

- 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
- 取得必要的執行個體詳細資訊.
- 找出私有金鑰並設定許可.
- (選用) 取得執行個體指紋.

允許來自 IP 位址的傳入 SSH 流量。

請確保與執行個體關聯的安全群組允許來自您的 IP 地址的傳入 SSH 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。

視需要在本機電腦安裝 SSH 用戶端。

您的本機電腦可能已預設安裝 SSH 用戶端。您可以在終端機視窗輸入下列命令來驗證這一點。若您的電腦無法識別該命令，您必須安裝 SSH 用戶端。

```
ssh
```

以下是 Windows 的一些可能選項。如果您的電腦執行不同的作業系統，請參閱該作業系統的文件以取得 SSH 用戶端選項。

在 Windows 安裝 OpenSSL

在 Windows 安裝 OpenSSH 後，您可以使用 SSH 從 Windows 電腦連線至 Linux 執行個體。開始之前，請確定符合下列要求。

Windows 版本

電腦上的 Windows 版本必須是 Windows Server 2019 或更新版本。

對於舊版 Windows，請下載並安裝 Win32-OpenSSH。

PowerShell 需求

若要使用 PowerShell 在您的 Windows 作業系統中安裝 OpenSSH，您必須執行 PowerShell 5.1 版或更高版本，並且您的帳戶必須是內建管理員群組的成員。從 PowerShell 執行 `$PSVersionTable.PSVersion` 以檢查您的 PowerShell 版本。

若要檢查您是否是內建管理員群組的成員，請執行下列 PowerShell 命令：

```
(New-Object
 Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent())).Is
```

如果您是內建管理員群組的成員，則輸出為 `True`。

若要使用 PowerShell 安裝適用於 Windows 的 OpenSSH，請執行下列 PowerShell 命令。

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

以下為範例輸出。

```
Path          :
```

```
Online        : True
RestartNeeded : False
```

若要使用 PowerShell 從 Windows 中解除安裝 OpenSSH，請執行下列 PowerShell 命令。

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

以下為範例輸出。

```
Path          :
Online        : True
RestartNeeded : True
```

安裝適用於 Linux 的 Windows 子系統 (WSL)

在 Windows 安裝 WSL 之後，您可以使用 Linux 命令列工具，例如 SSH 用戶端，從 Windows 電腦連線至 Linux 執行個體。

請遵循中的說明進行在 EC2 Windows 執行個體安裝適用於 Linux 的 Windows 子系統 如果您遵循 Microsoft 安裝指南的指示，他們會安裝 Linux 的 Ubuntu 發行版本。您可以視需要安裝不同的 Linux 發行版本。

在 WSL 終端機視窗中，將 .pem 檔案 (針對您在啟動時為執行個體指定的金鑰對) 從 Windows 複製到 WSL。請記下 .pem 檔案在 WSL 上的完整路徑，以便於在連接至執行個體時使用。如需有關如何指定 Windows 硬碟路徑的資訊，請參閱如何存取 C 磁碟機？。

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

如需有關解除安裝適用於 Linux 的 Windows 子系統的資訊，請參閱如何解除安裝 WSL 發行版本？。

使用 SSH 用戶端連接至 Linux 執行個體

請使用下列程序使用 SSH 用戶端連線至您的 Linux 執行個體。

使用 SSH 用戶端連線至執行個體

1. 在電腦開啟終端機視窗。
2. 若要連線至執行個體，請使用 ssh 命令。您需要在先決條件收集之執行個體的詳細資訊。例如，您需要私有金鑰 (.pem 檔案) 的位置、使用者名稱，以及公有 DNS 名稱或 IPv6 地址。下面是範例命令。

- (公有 DNS) 若要使用公有 DNS 名稱，請輸入下列命令。

```
ssh -i /path/key-pair-name.pem instance-user-name@instance-public-dns-name
```

- (IPv6) 或者，如果您的執行個體具有 IPv6 地址，請輸入下列命令以使用 IPv6 地址。

```
ssh -i /path/key-pair-name.pem instance-user-name@2001:db8::1234:5678:1.2.3.4
```

以下是回應範例。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
 can't be established.
ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

3. (選用) 確認安全提醒中的指紋與指紋是否相符。如果這些指紋不符合，可能有人嘗試發動中間人攻擊。如果它們符合，請繼續下一個步驟。如需詳細資訊，請參閱取得執行個體指紋。
4. 輸入 **yes**。

   您會看到如下的回應：

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
 the list of known hosts.
```

使用 PuTTY 連線至您的 Linux 執行個體

您可以使用 PuTTY (適用於 Windows 的免費 SSH 用戶端) 連線到 Linux 執行個體。

如果執行的是 Windows Server 2019 或更高版本，建議您使用 OpenSSH，這是一種透過 SSH 通訊協定進行遠端登入的開放原始碼連線工具。

> **Note**
>
> 如果您在嘗試連線至執行個體時收到錯誤訊息，請確認執行個體符合所有 SSH 連線先決條件。如果其符合所有先決條件，但您仍然無法連線至 Linux 執行個體，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

目錄

- 先決條件
- (選用) 使用 PuTTYgen 轉換私有金鑰
- 連接至您的 Linux 執行個體

先決條件

在使用 PuTTY 連線到 Linux 執行個體之前，請先完成下列任務。

完成一般先決條件。

- 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
- 取得必要的執行個體詳細資訊.
- 找出私有金鑰並設定許可.
- (選用) 取得執行個體指紋.

允許來自 IP 位址的傳入 SSH 流量。

請確保與執行個體關聯的安全群組允許來自您的 IP 地址的傳入 SSH 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。

視需要在本機電腦安裝 PuTTY。

從 PuTTY 下載頁面下載並安裝 PuTTY。如已安裝舊版的 PuTTY，建議您下載最新版本。務必安裝整個套件。

使用 PuTTYgen 將私有金鑰轉換為 PPK 格式。

您必須為啟動執行個體時指定的金鑰對指定私有金鑰。如果您以 .pem 格式建立私有金鑰，則必須將其轉換為 PPK 檔案以與 PuTTY 搭配使用。找到私有金鑰 (.pem 檔案)，然後按照 使用 PuTTYgen 轉換私有金鑰 中的步驟操作。

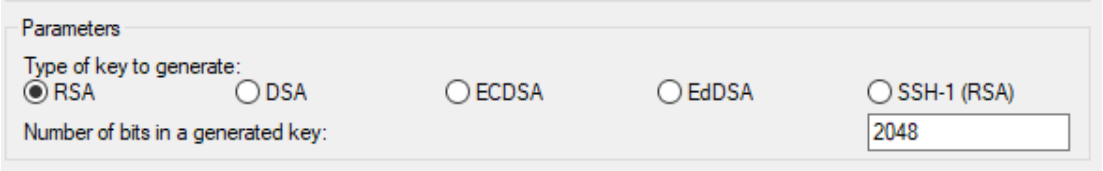
(選用) 使用 PuTTYgen 轉換私有金鑰

PuTTY 原生並不支援 PEM 格式的 SSH 金鑰。PuTTY 提供一套名為 PuTTYgen 的工具，可將 PEM 金鑰轉換為 PuTTY 所需的 PPK 格式。如果您使用 PEM 格式而非 PPK 格式建立金鑰，則必須將私有金鑰 (.pem 檔案) 轉換為此格式 (.ppk 檔案)，才能搭配 PuTTY 使用。

將私有私鑰從 PEM 格式轉換為 PPK 格式

1. 從 Start (開始) 功能表中選擇 All Programs (所有程式)、PuTTY、PuTTYgen。

2. 在 Type of key to generate (要產生的金鑰類型) 下，選擇 RSA (SSH-2 RSA)。如果您的 PuTTYgen 版本不包含此選項，請選擇 SSH-2 RSA。



3. 選擇 Load (載入)。根據預設，PuTTYgen 只會顯示副檔名為 `.ppk` 的檔案。若要尋找您的 `.pem` 檔案，請選擇顯示所有類型之檔案的選項。



4. 為您在啟動執行個體時所指定的金鑰對選取 `.pem` 檔案，然後選取 Open (開啟)。PuTTYgen 會顯示一個通知，指出 `.pem` 檔案已成功匯入。選擇 OK (確定)。
5. 若要以 PuTTY 可使用的格式儲存金鑰，請選擇 Save private key (儲存私有金鑰)。PuTTYgen 會顯示有關儲存沒有密碼短語之金鑰的警告。選擇 Yes (是)。

> **Note**
>
> 私有金鑰上的複雜密碼是額外一層保護。即使有人探索到您的私有金鑰，在沒有複雜密碼的情況下就無法使用您的私有金鑰。使用複雜密碼的缺點是提高自動化的難度，因為需要人力介入登入執行個體，或將檔案複製到執行個體。

6. 為您用於金鑰對的金鑰指定相同名稱 (例如，`key-pair-name`)，然後選擇 Save (儲存)。PuTTY 會自動新增 `.ppk` 副檔名。

您的私有金鑰對現已有正確的格式可供 PuTTY 使用。您現在可以使用 PuTTY 的 SSH 用戶端連線至您的執行個體。

連接至您的 Linux 執行個體

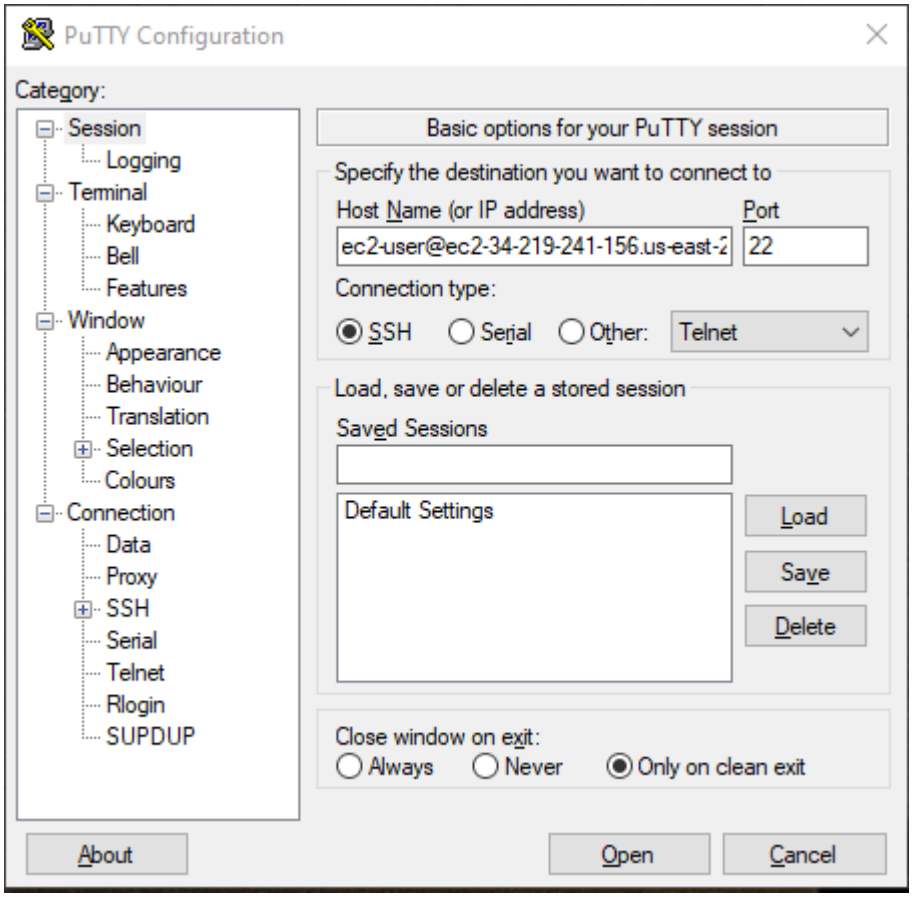
請使用下列程序使用 PuTTY 連線至您的 Linux 執行個體。需要您為私有金鑰建立的 `.ppk` 檔案。如需詳細資訊，請參閱上一節中的(選用) 使用 PuTTYgen 轉換私有金鑰。如果您在嘗試連線至執行個體時遇到錯誤，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

最後測試的版本 – PuTTY .78

使用 PuTTY 連線到您的執行個體

1. 啟動 PuTTY (在開始選單搜尋 PuTTY，然後選擇開啟)。
2. 在 Category (類別) 窗格中選擇 Session (工作階段)，然後填妥下列欄位：
   a. 在 Host Name (主機名稱) 方塊中，執行下列其中一項作業：
      - (公有 DNS) 若要使用執行個體的公有 DNS 名稱連線，請輸入 *instance-user-name*@*instance-public-dns-name*。
      - (IPv6) 或者，如果您的執行個體具有 IPv6 位址，若要使用執行個體的 IPv6 位址進行連線，請輸入 *instance-user-name*@*2001:db8::1234:5678:1.2.3.4*。

      如需有關如何尋找您執行個體的使用者名稱，以及執行個體的公有 DNS 名稱或 IPv6 地址的詳細資訊，請參閱 取得必要的執行個體詳細資訊。
   b. 確保 Port (連接埠) 值是 22。
   c. 在 Connection type (連線類型) 下，選取 SSH。



3. (選用) 您可設定 PuTTY 按定期間隔自動傳送保持連線資料，以保持工作階段持續作用。這對避免您執行個體因工作階段閒置而中斷連線很有幫助。在類別窗格中選擇連線，然後在保持連線之間的秒數中輸入所需間隔。例如，如果您的工作階段在閒置 10 分鐘後中斷連線，請輸入 180 以設定 PuTTY 每 3 分鐘傳送一次保持連線資料。
4. 在類別窗格中，展開連線、展開 SSH，然後選擇身分驗證。選擇憑證。
5. 在要驗證的私有金鑰檔案旁，選擇瀏覽。在選取私有金鑰檔案對話方塊中，選取您為金鑰對產生的 .ppk 檔案。您可以開啟檔案，或在選取私有金鑰檔案對話方塊中選擇開啟。
6. (選用) 如果您打算稍後再次連線到此執行個體，可以儲存工作階段資訊以供日後使用。在類別窗格中，選擇工作階段。在儲存的工作階段中輸入工作階段的名稱，然後選擇儲存。
7. 若要連線至執行個體，請選擇開啟。
8. 如果這是您第一次連線到此執行個體，PuTTY 會顯示安全提醒對話方塊，詢問您是否信任要連線的主機。

    a. (選用) 確認安全提醒對話方塊中的指紋是否符合您之前在(選用) 取得執行個體指紋中取得的指紋。如果這些指紋不符合，可能有人嘗試發動中間人攻擊。如果它們符合，請繼續下一個步驟。

    b. 選擇 Accept (接受)。隨即開啟視窗，您會連線到您的執行個體。

> **Note**
>
> 當您將私有金鑰轉換成 PuTTY 格式時，如已指定複雜密碼，您必須在登入執行個體時提供該複雜密碼。

如果您在嘗試連線至執行個體時遇到錯誤，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。

使用 SCP 將檔案傳輸至 Linux 執行個體

在本機電腦和 Linux 執行個體之間傳輸檔案的一種方式是使用安全複製通訊協定 (SCP)。SCP 是簡單操作的理想選項，例如一次性檔案複製。SCP 會使用您透過 SSH 連線至執行個體時所用的相同 .pem 檔案，以保障檔案傳輸的安全性。若您需要保持檔案同步，或者若檔案很大，則 rsync 比 SCP 更快、效率更高。為了安全起見，請透過 SSH 使用 rsync，因為 rsync 預設會使用純文字傳輸資料。

使用 SCP 連線到您的 Linux 執行個體之前，請先完成下列先決條件：

- 完成一般先決條件。

  - 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
  - 取得必要的執行個體詳細資訊.
  - 找出私有金鑰並設定許可.
  - (選用) 取得執行個體指紋.
- 允許來自 IP 位址的傳入 SSH 流量。

  請確保與執行個體關聯的安全群組允許來自您的 IP 地址的傳入 SSH 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。
- 安裝 SCP 用戶端。

  大部分的 Linux、Unix 和 Apple 電腦預設都包含 SCP 用戶端。如果您的電腦沒有 SCP 用戶端，OpenSSH 專案會提供 SSH 工具完整套件的免費實作，包括 SCP 用戶端。如需詳細資訊，請參閱 https://www.openssh.com。

下列程序會逐步引導您使用 SCP 使用執行個體的公有 DNS 名稱傳輸檔案，或是使用 IPv6 地址 (如果執行個體具備的話) 傳輸檔案。

使用 SCP 在電腦和執行個體之間傳輸檔案

1. 決定來源檔案在電腦上的位置，以及執行個體上的目的地路徑。在下列範例中，私有金鑰檔案的名稱為 `key-pair-name.pem`，要傳輸的檔案為 `my-file.txt`，執行個體的使用者名稱為 ec2-user，執行個體的公用 DNS 名稱為 `instance-public-dns-name`，而執行個體的 IPv6 地址為 `2001:db8::1234:5678:1.2.3.4`。
   - (公用 DNS) 若要將檔案傳輸到執行個體上的目的地，請從您的電腦輸入下列命令。

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@instance-public-dns-name:path/
     ```
   - (IPv6) 若要將檔案傳輸到執行個體上的目的地 (如果執行個體具有 IPv6 地址)，請從您的電腦輸入下列命令。IPv6 位址必須用方括號 ([ ]) 括住，必須逸出 (\)。

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@
     \[2001:db8::1234:5678:1.2.3.4\]:path/
     ```
2. 如果您尚未使用 SSH 連接到執行個體，則會看到如下所示的回應：

   ```
   The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'
   ```

```
can't be established.
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

(選用) 您可以選用驗證安全性警示中的指紋是否符合執行個體指紋。如需詳細資訊，請參閱(選用) 取得執行個體指紋。

輸入 **yes**。

3. 如果傳輸成功，則回應會和下列內容相似：

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
my-file.txt                                   100%   480     24.4KB/s   00:00
```

4. 若要反向 (從 Amazon EC2 執行個體到您的電腦) 傳輸檔案，請反轉主機參數的順序。例如：您可以從 EC2 執行個體傳輸 `my-file.txt` 到本機電腦上的目的地 `my-file2.txt`，如下列範例所示。

   - (公用 DNS) 若要將檔案傳輸到電腦上的目的地，請從您的電腦輸入下列命令。

     ```
     scp -i /path/key-pair-name.pem ec2-user@instance-public-dns-name:path/my-file.txt path/my-file2.txt
     ```

   - (IPv6) 如果執行個體具有 IPv6 地址，欲將檔案傳輸到電腦上的目的地，請從您的電腦輸入下列命令。IPv6 位址必須用方括號 ([ ]) 括住，必須逸出 (\)。

     ```
     scp -i /path/key-pair-name.pem ec2-user@\[2001:db8::1234:5678:1.2.3.4\]:path/my-file.txt path/my-file2.txt
     ```

在您的 Amazon EC2 Linux 執行個體上管理系統使用者

每個 Linux 執行個體都會使用預設 Linux 系統使用者來啟動。您可以將使用者新增至執行個體，並刪除使用者。

針對預設使用者，預設使用者名稱會取決於您啟動執行個體時所指定的 AMI。

> **Note**
>
> 預設情況下，密碼驗證和根登入會停用，而 sudo 會啟用。若要登入執行個體，您必須建立金鑰對。如需有關登入的詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。

您可以允許執行個體的密碼驗證和根登入。如需詳細資訊，請參閱適用於您作業系統的文件。

> **Note**
>
> Linux 系統使用者不應和 IAM 使用者混淆。如需詳細資訊，請參閱「IAM 使用者指南」中的 IAM 使用者。

目錄

- 預設使用者名稱
- 考量事項
- 建立使用者
- 移除使用者

預設使用者名稱

EC2 執行個體的預設使用者名稱取決於您啟動執行個體時指定的 AMI。

預設使用者名稱為：

- 針對 Amazon Linux AMI，使用者名稱為 `ec2-user`。
- 針對 CentOS AMI，使用者名稱為 `centos` 或 `ec2-user`。
- 針對 Debian AMI，使用者名稱為 `admin`。
- 針對 Fedora AMI，使用者名稱為 `fedora` 或 `ec2-user`。
- 針對 FreeBSD AMI，使用者名稱為 `ec2-user`。
- 針對 RHEL AMI，使用者名稱為 `ec2-user` 或 `root`。
- 針對 SUSE AMI，使用者名稱為 `ec2-user` 或 `root`。
- 若為 Ubuntu AMI，使用者名稱為 `ubuntu`。
- 針對 Oracle AMI，使用者名稱為 `ec2-user`。
- 針對 Bitnami AMI，使用者名稱為 `bitnami`。

> **Note**
>
> 若要尋找其他 Linux 發行版本的預設使用者名稱，請向 AMI 供應商確認。

考量事項

許多應用程式都適合使用預設使用者。不過，您可以選擇新增使用者，讓個人能有自己的檔案和工作空間。此外，為新使用者建立使用者也會比授予多名 (可能缺乏經驗) 預設使用者的存取權更安全，因為若預設使用者使用不當，可能會對系統造成重大傷害。

若要讓使用者能夠使用 Linux 系統使用者，透過 SSH 存取您的 EC2 執行個體，您必須與使用者共用 SSH 金鑰。或者，您可以使用 EC2 Instance Connect 為使用者提供存取權限，如此就不需要共用和管理 SSH 金鑰。如需詳細資訊，請參閱使用公有 IP 位址與 EC2 Instance Connect 連線至 Linux 執行個體。

建立使用者

首先建立使用者，然後新增允許使用者連線至並登入執行個體的 SSH 公有金鑰。

> **Important**
>
> 在此程序的步驟 1 中，您會建立新的金鑰配對。由於金鑰配對的功能與密碼類似，因此安全地處理至關重要。如果您為使用者建立金鑰配對，則必須確保將私有金鑰安全地傳送給使用者。或者，使用者可以透過建立專屬金鑰配對，將私人金鑰安全地保存在其機器上，以完成步驟 1 和步驟 2，然後將公有金鑰傳送給您，以完成步驟 3 的流程。

建立使用者

1. 建立新的金鑰對。您必須將 `.pem` 檔案提供給您要為其建立使用者的使用者。他們必須使用此檔案來連接到執行個體。
2. 從您在上一個步驟中建立的金鑰對中，擷取公開金鑰。

   ```
   $ ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
   ```

   此命令會傳回公有金鑰，如下列範例所示。

   ```
   ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih
   ```

```
+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/
i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco
+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3RbBQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

3. 連線到執行個體。
4. 使用 adduser 命令建立使用者並將其新增到系統 (在 /etc/passwd 檔案中具有項目)。此命令也會為使用者建立群組和主目錄。在此範例中，使用者命名為 *newuser*。

   - AL2023 和 Amazon Linux 2

     如果使用 AL2023 和 Amazon Linux 2，則在預設情況下，建立的使用者已停用密碼驗證。

     ```
     [ec2-user ~]$ sudo adduser newuser
     ```

   - Ubuntu

     包含 --disabled-password 參數，以建立已停用密碼驗證的使用者。

     ```
     [ubuntu ~]$ sudo adduser newuser --disabled-password
     ```

5. 切換到新的使用者，以便您建立的目錄和檔案可擁有適當的所有權。

   ```
   [ec2-user ~]$ sudo su - newuser
   ```

   此提示會從 ec2-user 變更為 *newuser*，表示您已將 shell 工作階段切換至新使用者。
6. 將 SSH 公有金鑰新增到使用者。首先在使用者的主目錄中建立 SSH 金鑰檔案的目錄、然後建立金鑰檔案，最後將公有金鑰貼到金鑰檔案中，如下列子步驟中所示。

   a. 在 .ssh 主目錄中建立 *newuser* 目錄，然後將其檔案許可變更為 700 (只有擁有者能讀取、寫入和開啟目錄)。

      ```
      [newuser ~]$ mkdir .ssh
      ```

      ```
      [newuser ~]$ chmod 700 .ssh
      ```

> **Important**
>
> 若沒有這些明確的檔案許可，使用者將無法登入。

b. 在 .ssh 目錄中建立名為 authorized_keys 的檔案，然後將其檔案許可變更為 600 (只有擁有者能讀取和寫入檔案)。

```
[newuser ~]$ touch .ssh/authorized_keys
```

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

> **Important**
>
> 若沒有這些明確的檔案許可，使用者將無法登入。

c. 使用您喜愛的文字編輯器 (例如 vim 或 nano) 來開啟 authorized_keys 檔案。

```
[newuser ~]$ nano .ssh/authorized_keys
```

將您在步驟 2 中擷取到的公開金鑰貼到檔案中，並儲存這些變更。

> **Important**
>
> 請確定將此公開金鑰貼成連續一行。此公開金鑰絕對不可分成多行。

使用者現在應該能使用對應到您為 authorized_keys 檔案新增之公有金鑰的私有金鑰，在您的執行個體上登入 *newuser* 使用者。如需連接至 Linux 執行個體之不同方法的詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。

移除使用者

若不再需要使用者，您可以移除該使用者，使該使用者無法再被使用。

使用 userdel 命令來移除系統的使用者。當指定 -r 參數時，也會同時刪除使用者的主目錄和郵件多工緩衝處理。若要保留使用者的主目錄和郵件多工緩衝處理，請忽略 -r 參數。

```
[ec2-user ~]$ sudo userdel -r olduser
```

使用 RDP 連線至您的 Windows 執行個體

您可以使用遠端桌面，連接至從多數 Windows Amazon Machine Image (AMI) 建立的 Amazon EC2 執行個體。遠端桌面使用遠端桌面協定 (RDP)，可讓您以一般使用電腦的方式連接和使用執行個體，就像您使用坐在您面前的電腦 (本機電腦) 一樣。其適用於多數的 Windows 版本，也適用於 Mac OS。

Windows Server 作業系統的授權允許為了管理用途而同時有兩個遠端連線。Windows Server 的授權包含在您的 Windows 執行個體價格中。如果您同時需要超過兩個的遠端連線，就必須購買 Remote Desktop Services (RDS) 授權。如果您嘗試第三個連線，就會出現錯誤。

> **Tip**
>
> 如果您需要連線至執行個體，以針對在 AWS Nitro 系統上建置的執行個體進行開機、網路組態和其他問題的疑難排解，您可以使用 EC2 Serial Console for 執行個體。

目錄

- 使用 RDP 用戶端連線至您的 Windows 執行個體。
- 使用 Fleet Manager 連線到 Windows 執行個體
- 使用 RDP 將檔案傳輸至 Windows 執行個體

使用 RDP 用戶端連線至您的 Windows 執行個體。

您可以使用 RDP 用戶端連線至您的 Windows 執行個體，如下所示。

> **Tip**
>
> 或者，您可以使用 Systems Manager Fleet Manager 或 EC2 Instance Connect Endpoint 連線到 Windows 執行個體。

先決條件

您必須符合下列先決條件，才能使用 RDP 用戶端連線至 Windows 執行個體。

- 完成一般先決條件。

  - 請確認您的執行個體已通過狀態檢查。執行個體可能需要幾分鐘的時間準備就緒，以接受連線要求。如需詳細資訊，請參閱檢視狀態檢查。
  - 取得必要的執行個體詳細資訊.
  - 找出私有金鑰並設定許可.
  - (選用) 取得執行個體指紋.
- 安裝 RDP 用戶端。
  - (Windows) Windows 預設包含 RDP 用戶端。若要確認，請在命令提示視窗下輸入 mstsc。若您的電腦無法辨識此命令，請從 Microsoft Store 下載 Microsoft Remote Desktop 應用程式。
  - (macOS X) 從 Mac App Store 下載 Windows App for Mac (原名 Microsoft Remote Desktop)。
  - (Linux) 使用 Remmina。
- 允許來自您 IP 位址的傳入 RDP 流量。

  請確認與執行個體關聯的安全群組允許來自您 IP 位址的傳入 RDP 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。

擷取管理員密碼

如果您已將執行個體加入網域，您可以使用來自 的網域登入資料連線至執行個體 Directory Service。在遠端桌面登入畫面上，不要使用本機電腦名稱和產生的密碼，而是使用系統管理員的合格使用者名稱 (例如 **`corp.example.com\Admin`**) 和此帳戶的密碼。

若要使用 RDP 連線到 Windows 執行個體，您必須擷取初始管理員密碼，然後在連線到執行個體時輸入此密碼。在執行個體啟動後，需要幾分鐘的時間處理才能提供密碼。您的帳戶必須擁有呼叫 GetPasswordData 動作的權限。如需詳細資訊，請參閱控制 Amazon EC2 API 存取的政策範例。

管理員帳戶的預設使用者名稱取決於 AMI 包含的作業系統 (OS) 的語言。若要確定正確的使用者名稱，請識別作業系統的語言，然後選擇對應的使用者名稱。例如，對於英文作業系統，使用者名稱為 `Administrator`，對於法文作業系統，使用者名稱為 `Administrateur`，對於葡萄牙文作業系統，使用者名稱為 `Administrador`。如果作業系統的語言版本沒有相同語言的使用者名稱，請選擇使用者名稱 `Administrator (Other)`。如需詳細資訊，請參閱 Microsoft 網站中的管理員帳戶在 Windows 中的當地語系化名稱。

擷取初始管理員密碼

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。

3. 選取執行個體，然後選取 Connect (連線)。
4. 在連線至執行個體頁面，選擇 RDP 用戶端索引標籤。
5. 對於使用者名稱，選擇管理員帳戶的預設使用者名稱。您選擇的使用者名稱必須與您用於啟動執行個體的 AMI 所包含的作業系統 (OS) 的語言相符。若無與作業系統相同語言的使用者名稱，請選擇管理員 (其他)。
6. 選擇取得密碼。
7. 在取得 Windows 密碼頁面，執行下列動作：

   a. 選擇上傳私有金鑰檔案，然後導覽至您在啟動執行個體時指定的私有金鑰 (`.pem`) 檔案。選取檔案並選取 Open (開啟)，將檔案的完整內容複製至此視窗。

   b. 選擇解密密碼。取得 Windows 密碼頁面隨即關閉，執行個體的預設管理員密碼會顯示在密碼下，取代先前顯示的取得密碼連結。

   c. 複製密碼並將其保存在安全處。必須使用此密碼來連接至執行個體。

連線至 Windows 執行個體

以下程序使用適用於 Windows 的遠端桌面連線用戶端 (MSTSC)。如果您使用其他 RDP 用戶端，請下載 RDP 檔案，然後查看 RDP 用戶端的文件，了解建立 RDP 連線的步驟。

使用 RDP 用戶端連線至 Windows 執行個體。

1. 在連線至執行個體頁面，選擇下載遠端桌面檔案。檔案下載完成後，請選擇取消以返回執行個體頁面。RDP 檔案已下載至您的 `Downloads` 資料夾。
2. 執行 `mstsc.exe` 以開啟 RDP 用戶端。
3. 展開顯示選項，選擇開啟，然後從 `Downloads` 資料夾中選取 .rdp 檔案。
4. 依預設，電腦是執行個體的公有 IPv4 DNS 名稱，使用者名稱是管理員帳戶。若要改用 IPv6 連線至執行個體，請將執行個體的公有 IPv4 DNS 名稱替換為其 IPv6 地址。檢視預設設定並視需要進行變更。
5. 選擇連線。如果您收到遠端連線發布者不明的警告，請選擇連線以繼續。
6. 輸入先前儲存的密碼，然後選擇確定。
7. 由於自我簽署憑證的性質，您可能會收到安全憑證無法驗證的警告。執行以下任意一項：
   - 如果您信任憑證，請選擇是，以連線至您的執行個體。
   - [Windows] 在繼續之前，請將憑證指紋與系統日誌的值進行比較，以確認遠端電腦身分。選擇檢視憑證，然後從詳細資料索引標籤選擇指紋。將此值與動作、監控和疑難排解、取得系統日誌中的 `RDPCERTIFICATE-THUMBPRINT` 值進行比較。

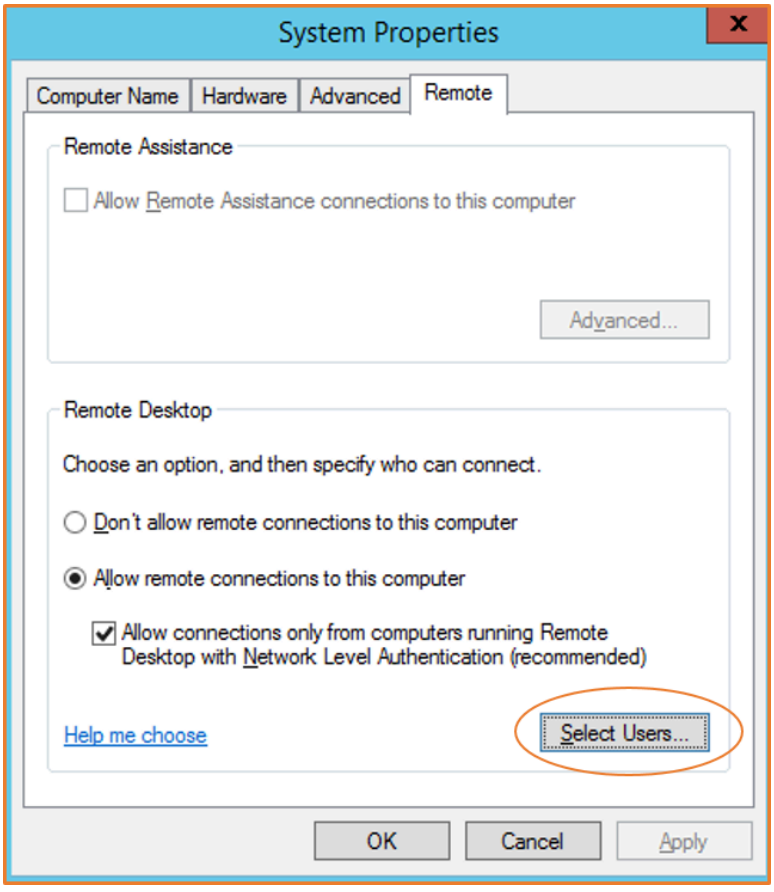
- [Mac OS X] 在繼續之前，請將憑證指紋與系統日誌的值進行比較，以確認遠端電腦身分。選擇顯示憑證、展開詳細資訊，然後選擇 SHA1 指紋。將此值與動作、監控和疑難排解、取得系統日誌中的 `RDPCERTIFICATE-THUMBPRINT` 值進行比較。

8. 如果 RDP 連線成功，RDP 用戶端會顯示 Windows 登入畫面，然後顯示 Windows 桌面。如果您收到錯誤訊息，請參閱 the section called “遠端桌面無法連線到遠端電腦”。完成 RDP 連線後，即可關閉 RDP 用戶端。

設定使用者帳戶

透過 RDP 連線至執行個體之後，建議您執行下列任務：

- 變更預設的管理員密碼。您可以在登入執行個體後變更密碼，就像使用任何其他執行 Windows 伺服器的電腦一樣。
- 在執行個體上建立其他擁有管理員權限的使用者。如果您忘記管理員密碼或管理員帳戶發生問題，這可作為保護措施。新使用者必須具備可遠端存取執行個體的許可。在 Windows 桌面或檔案總管上，以滑鼠右鍵按一下This PC (本機)，選取內容，以開啟系統內容。選取遠端設定，然後選取選取使用者，將使用者新增至遠端桌面使用者群組。



使用 Fleet Manager 連線到 Windows 執行個體

您可以使用 Fleet Manager 功能 AWS Systems Manager，使用遠端桌面通訊協定 (RDP) 連線至 Windows 執行個體，並在 的相同頁面上顯示最多四個 Windows 執行個體 AWS 管理主控台。您可以從 Amazon EC2 主控台的執行個體頁面中直接連線至 Fleet Manager 遠端桌面的第一個執行個體。如需有關 Fleet Manager 的詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的使用遠端桌面連線到受管執行個體。

如果您使用 Fleet Manager 進行連線，則不需要特別允許來自您 IP 地址的傳入 RDP 流量。Fleet Manager 會為您處理。

先決條件

在嘗試使用 Fleet Manager 連線至執行個體之前，您必須設定環境。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的設定您的環境。

使用 Fleet Manager 連線到 Windows 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽窗格中選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Connect (連線)。
4. 在 RDP 用戶端索引標籤的連線類型，選擇使用 Fleet Manager 連線。
5. 選擇 Fleet Manager 遠端桌面。這將開啟 AWS Systems Manager 主控台中的 Fleet Manager Remote Desktop (Fleet Manager 遠端桌面) 頁面。
6. 輸入您的憑證，然後選擇連線。
7. 如果 RDP 連線成功，Fleet Manager 將顯示 Windows 桌面。當您完成工作階段時，請選擇動作、結束工作階段。

如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的使用遠端桌面連線至 Windows Server 受管執行個體。

使用 RDP 將檔案傳輸至 Windows 執行個體

您可用與任何 Windows 伺服器相同的使用方式來使用 Windows 執行個體。例如，您可使用 Microsoft Remote Desktop Connection (RDP) 的本機檔案共用功能在 Windows 執行個體和本機電腦之間傳輸檔案。本機檔案可以位在硬碟、DVD 光碟、可攜式媒體磁碟機和映射的網路磁碟機。

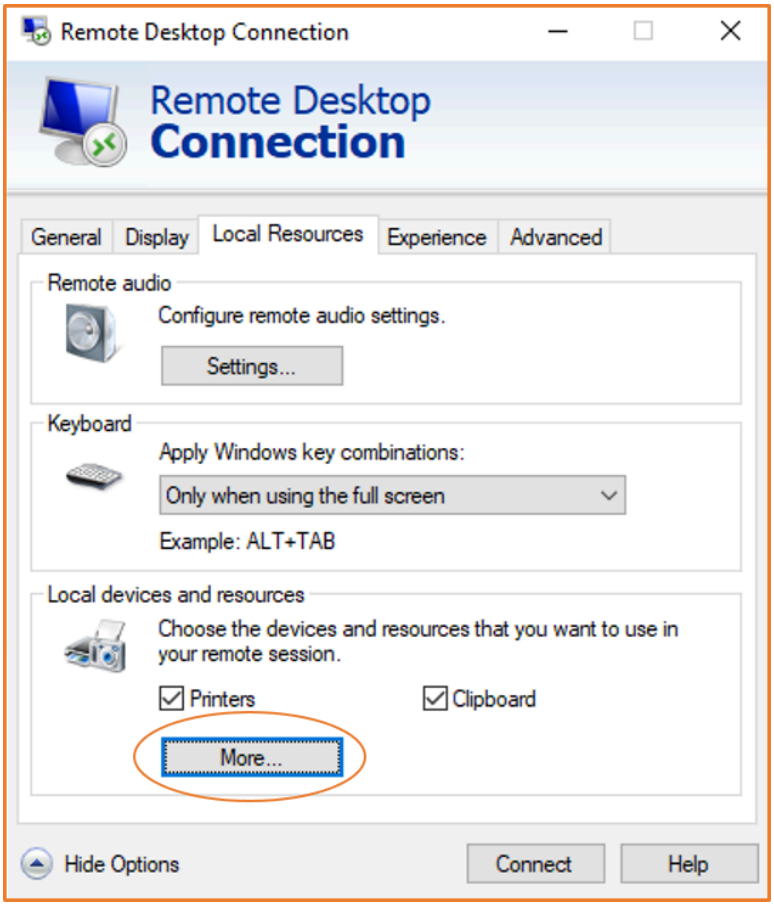
如要從 Windows 執行個體存取本機檔案，您必須透過將遠端工作階段磁碟機映射至您本機磁碟機來啟用本機檔案共享功能。這些步驟略有不同，取決於您的本機電腦作業系統是 Windows 還是 macOS X。

如需使用 RDP 連線先決條件的詳細資訊，請參閱 先決條件。

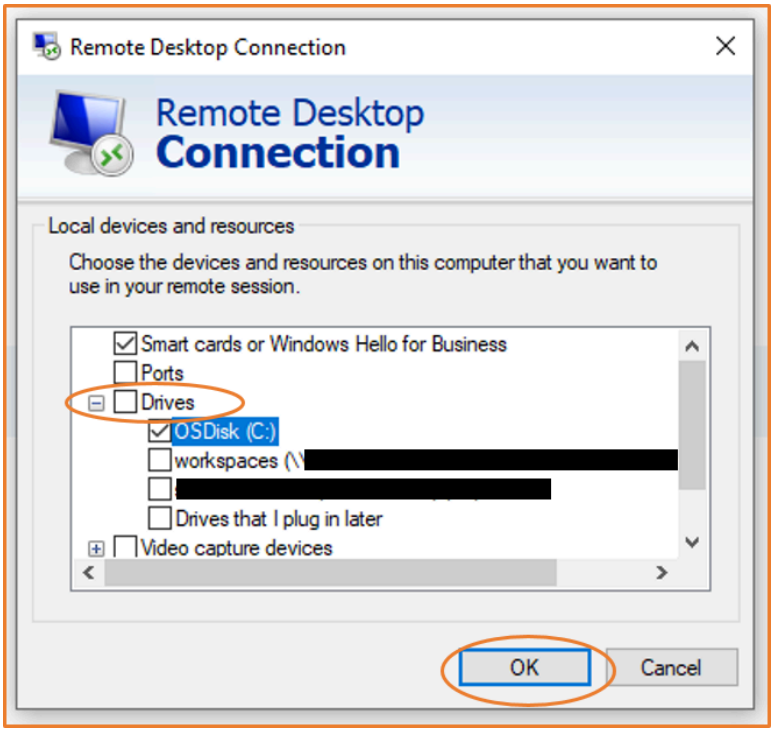
Windows

如要將遠端工作階段磁碟機映射至您本機 Windows 電腦的本機磁碟機

1. 開啟遠端桌面連線用戶端。
2. 選擇 Show Options (顯示選項)。
3. 將執行個體主機名稱新增至電腦欄位，並將使用者名稱新增至使用者名稱欄位，如下所示：
   a. 在 Connection settings (連線設定) 下，選擇 Open... (開啟...)，然後瀏覽您從 Amazon EC2 主控台下載的 RDP 捷徑檔案。該檔案包含標識執行個體的公有 IPv4 DNS 主機名稱和管理員使用者名稱。
   b. 選取檔案，然後選取 Open (開啟)。Computer (電腦) 和 User name (使用者名稱) 欄位使用 RDP 捷徑檔案的值進行填入。
   c. 選擇儲存。
4. 選擇 Local Resources (本機資源) 標籤。
5. 在 Local Devices and resources (本機裝置和資源) 下，選擇 More... (更多...)



6. 開啟 Drives (磁碟機) 並選取本機磁碟機，以映射到 Windows 執行個體。
7. 選擇 OK (確定)。

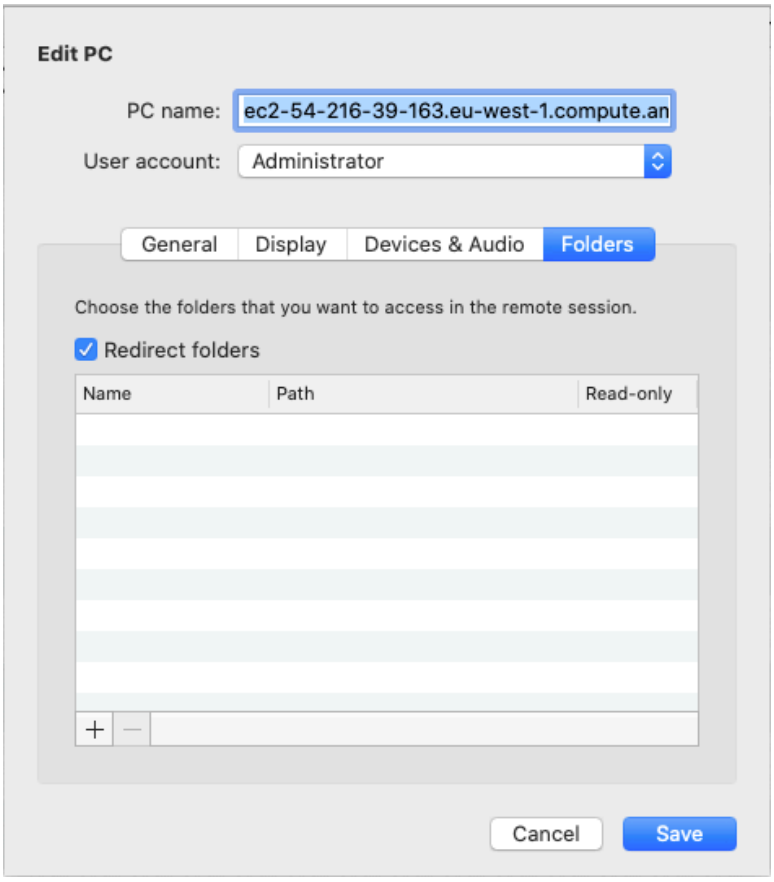


8. 選擇 Connect (連接) 以連接到您的 Windows 執行個體。

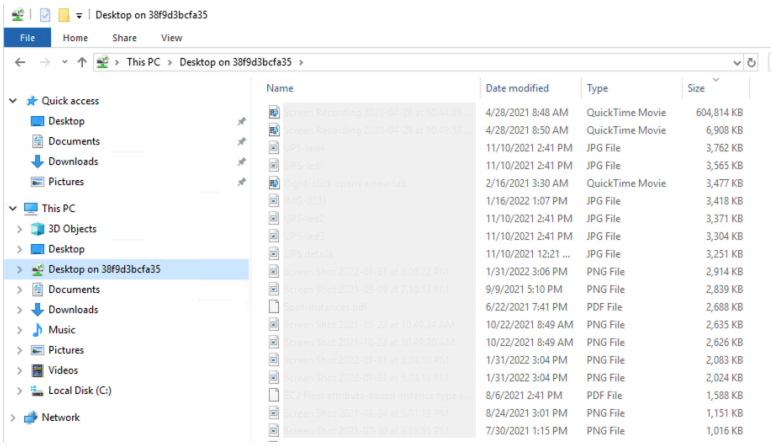
macOS X

如要將遠端工作階段磁碟機映射至本機 macOS X 電腦上的本機資料夾

1. 開啟遠端桌面連線用戶端。
2. 瀏覽至您從 Amazon EC2 主控台下載的 RDP 檔案 (在您最初連接到執行個體時)，然後將其拖曳至遠端桌面連線用戶端。
3. 開啟 RDP 檔案的內容 （ 按一下滑鼠右鍵 ） 選單，然後選擇編輯。
4. 選取資料夾索引標籤，然後選取重新引導資料夾核取方塊。



5. 選擇左下方的 + 圖示，瀏覽至要映射的資料夾，然後選擇 Open (開啟)。為每個要映射的資料夾重複此步驟。
6. 選擇儲存。
7. 選擇 Connect (連接) 以連接到您的 Windows 執行個體。系統會提示您輸入密碼。
8. 在執行個體上的檔案總管中，展開 This PC (此 PC)，然後找出您可從中存取本機檔案的共用資料夾。於下列螢幕擷取畫面中，本機電腦上的 Desktop (桌面) 資料夾已映射至執行個體上的遠端工作階段磁碟機。



如需在 Mac 電腦上使本機裝置可用於遠端工作階段的詳細資訊，請參閱 Mac 用戶端入門。

使用 Session Manager 連線至 Amazon EC2 執行個體

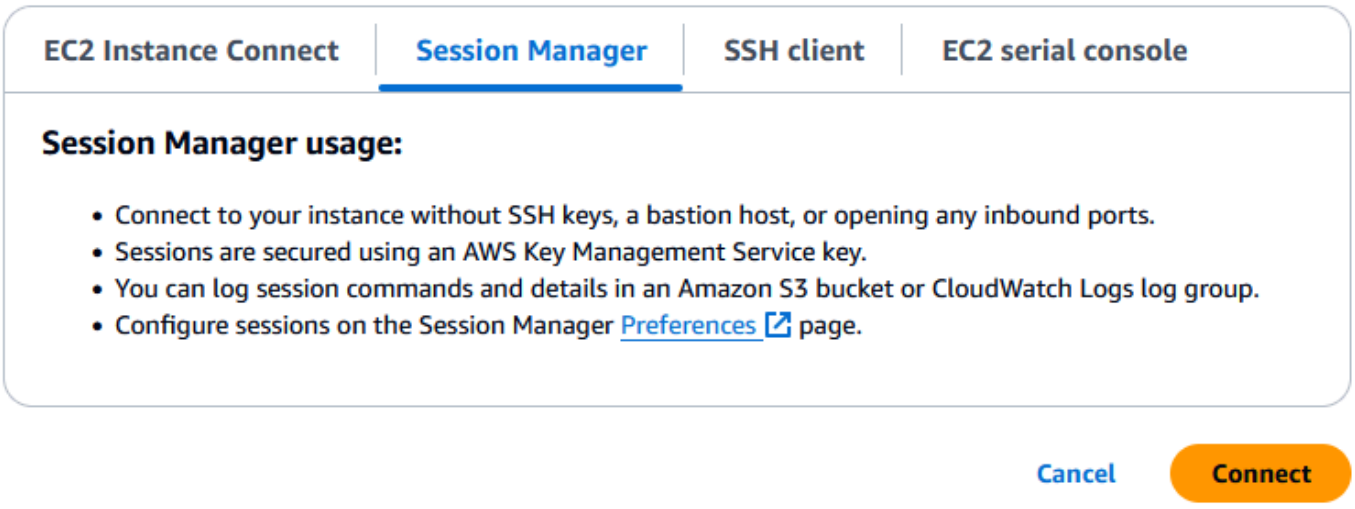
Session Manager 是一種全受管 AWS Systems Manager 功能，可透過互動式瀏覽器型 shell 或透過管理您的 Amazon EC2 執行個體 AWS CLI。您可以使用工作階段管理員，搭配您帳戶中的執行個體啟動工作階段。工作階段啟動之後，您可以像對任何其他連線類型一樣在執行個體執行互動式命令。如需工作階段管理員的詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 AWS Systems Manager 工作階段管理員。

先決條件

在嘗試使用 Session Manager 連線至執行個體之前，您必須完成必要的設定步驟。例如，執行個體必須由 SSM 管理，並且必須附加具有 AmazonSSMManagedInstanceCore 政策的 IAM 角色。如需詳細資訊，請參閱設定工作階段管理員。

在 Amazon EC2 主控台透過 Session Manager 連接到 Amazon EC2 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Connect (連線)。
4. 對於連線方法，選擇工作階段管理員。
5. 選取連線即可啟動工作階段。



疑難排解

如果您收到未獲授權執行一或多個 Systems Manager 動作 (`ssm:command-name`) 的錯誤，則必須更新政策以允許您從 Amazon EC2 主控台啟動工作階段。如需詳細資訊和指示，請參閱「AWS Systems Manager 使用者指南」中的工作階段管理員的預設 IAM 政策快速入門。

使用公有 IP 位址與 EC2 Instance Connect 連線至 Linux 執行個體

Amazon EC2 Instance Connect 提供了一種透過 Secure Shell (SSH) 連線至 Linux 執行個體的安全方式。使用 EC2 Instance Connect，您可以使用 AWS Identity and Access Management (IAM) 政策和主體來控制執行個體的 SSH 存取，無需共用和管理 SSH 金鑰。所有使用 EC2 Instance Connect 的連線請求會記錄在 AWS CloudTrail，這樣您就可以稽核連接請求。

可以使用 Amazon EC2 主控台或您選擇的 SSH 用戶端，透過 EC2 Instance Connect 連線到執行個體。

當您使用 EC2 Instance Connect 連接至執行個體時，EC2 Instance Connect API 會將 SSH 公有金鑰推送至執行個體中繼資料，此金鑰可保留 60 秒。連接至使用者的 IAM 政策會授權使用者將公有金鑰推送至執行個體中繼資料。SSH 常駐程式使用安裝 EC2 Instance Connect 時設定的 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，從執行個體中繼資料查詢身分驗證所用的公有金鑰並將您連接至執行個體。

> **Tip**
>
> EC2 Instance Connect 是連線到 Linux 執行個體的其中一個選項。如需有關其他選項的詳細資訊，請參閱 使用 SSH 連線至您的 Linux 執行個體。連線至 Windows 執行個體，請參閱 使用 RDP 連線至您的 Windows 執行個體。

定價

EC2 Instance Connect 可免費使用。

區域可用性

EC2 Instance Connect 適用於所有 ， AWS 區域 但 除外 AWS European Sovereign Cloud (Germany)。本地區域不支援。

目錄

- 教學課程：使用 EC2 Instance Connect 完成連線執行個體所需的設定
- EC2 Instance Connect 的先決條件
- 為 EC2 Instance Connect 授予 IAM 許可

- 在 EC2 執行個體上安裝 EC2 Instance Connect
- 使用 EC2 Instance Connect 連線至 Linux 執行個體
- 解除安裝 EC2 Instance Connect

如需討論如何使用 EC2 Instance Connect 來改善堡壘主機安全性的部落格文章，請參閱使用 Amazon EC2 Instance Connect 保護您的堡壘主機。

教學課程：使用 EC2 Instance Connect 完成連線執行個體所需的設定

若要在 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體，您必須先完成先決條件設定，才能成功連線至執行個體。本教學課程的目的是指導您完成任務以完成先決條件設定。

教學課程概觀

在本教學課程中，您將完成下列四項任務：

- 任務 1：授予使用 EC2 Instance Connect 所需的許可

  首先，您將建立包含 IAM 許可的 IAM 政策，允許您將公有金鑰推送至執行個體中繼資料。您可以將此政策連接至您的 IAM 身分 (使用者、使用者群組或角色)，以便您的 IAM 身分取得這些許可。
- 任務 2：允許從 EC2 Instance Connect 服務到執行個體的傳入流量

  然後，您將建立安全群組，允許從 EC2 Instance Connect 服務到執行個體的流量。當您在 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體時，必須執行此動作。
- 任務 3：啟動您的執行個體

  然後，您將使用預先安裝 EC2 Instance Connect 的 AMI 啟動 EC2 執行個體，並新增上一個步驟所建立的安全群組。
- 任務 4：連線到您的執行個體

  最後，您將在 Amazon EC2 主控台使用 EC2 Instance Connect 來連線至執行個體。如果您可以連線，則可以確定您在任務 1、2 和 3 完成的先決條件組態已成功。

任務 1：授予使用 EC2 Instance Connect 所需的許可

當您使用 EC2 Instance Connect 連接至執行個體時，EC2 Instance Connect API 會將 SSH 公有金鑰推送至執行個體中繼資料，此金鑰可保留 60 秒。您需要將 IAM 政策連接至 IAM 身分 (使用者、使用者群組或角色)，才能授予您將公有金鑰推送至執行個體中繼資料所需的許可。

任務目標

您將建立 IAM 政策，授予將公有金鑰推送至執行個體的許可。要允許的特定動作是 `ec2-instance-connect:SendSSHPublicKey`。您還必須允許 `ec2:DescribeInstances` 動作，以便在 Amazon EC2 主控台檢視及選取執行個體。

建立政策後，您將政策連接至您的 IAM 身分 (使用者、使用者群組或角色)，以便您的 IAM 身分取得許可。

您要建立的政策設定如下：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

Important

本教學課程建立的 IAM 政策是高度寬鬆政策，可讓您使用任何 AMI 使用者名稱連線到任何執行個體。本教學課程使用此高度寬鬆政策，讓教學課程保持簡單，並專注於本教學課程所教導的特定組態。不過，在生產環境，建議將 IAM 政策設定為提供最低權限許可。如需 IAM 政策的範例，請參閱為 EC2 Instance Connect 授予 IAM 許可。

建立並連接可讓您使用 EC2 Instance Connect 連線至執行個體的 IAM 政策

1. 首先建立 IAM 政策

a. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。

b. 在導覽窗格中，選擇政策。

c. 選擇 Create policy (建立政策)。

d. 在新增許可頁面，執行以下作業：

   i. 針對服務類別，請選擇 EC2 Instance Connect。

   ii. 在允許的動作下方，在搜尋欄位輸入 **send** 以顯示相關動作，然後選取 SendSSHPublicKey。

   iii. 在資源，選擇全部。對於生產環境，我們建議您依其 ARN 指定執行個體，但在本教學課程，您允許所有執行個體。

   iv. 選擇新增其他許可。

   v. 針對服務，選擇 EC2。

   vi. 在允許的動作下，在搜尋欄位中開始輸入 **describein** 以顯示相關動作，然後選取 DescribeInstances。

   vii. 選擇下一步。

e. 在檢閱和建立頁面，執行以下作業：

   i. 針對 Policy Name (政策名稱)，請輸入政策的名稱。

   ii. 選擇建立政策。

2. 然後將政策連接至您的身分

a. 在 IAM 主控台的導覽窗格中，選擇 Policies (政策)。

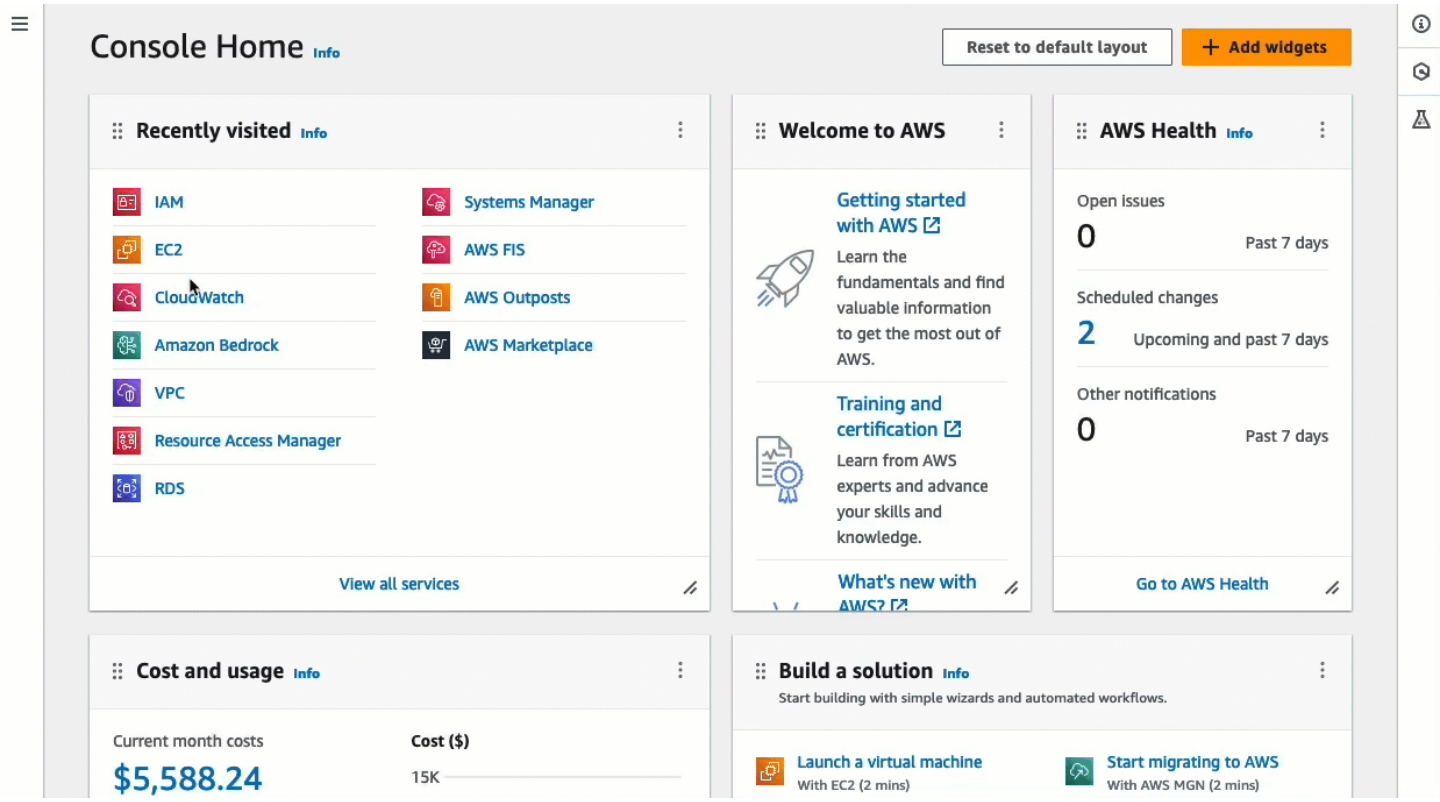
b. 在政策清單中，選取您建立的政策名稱旁的選項按鈕。您可以使用搜尋方塊來篩選政策清單。

c. 選擇 Actions (動作)、Attach (連接)。

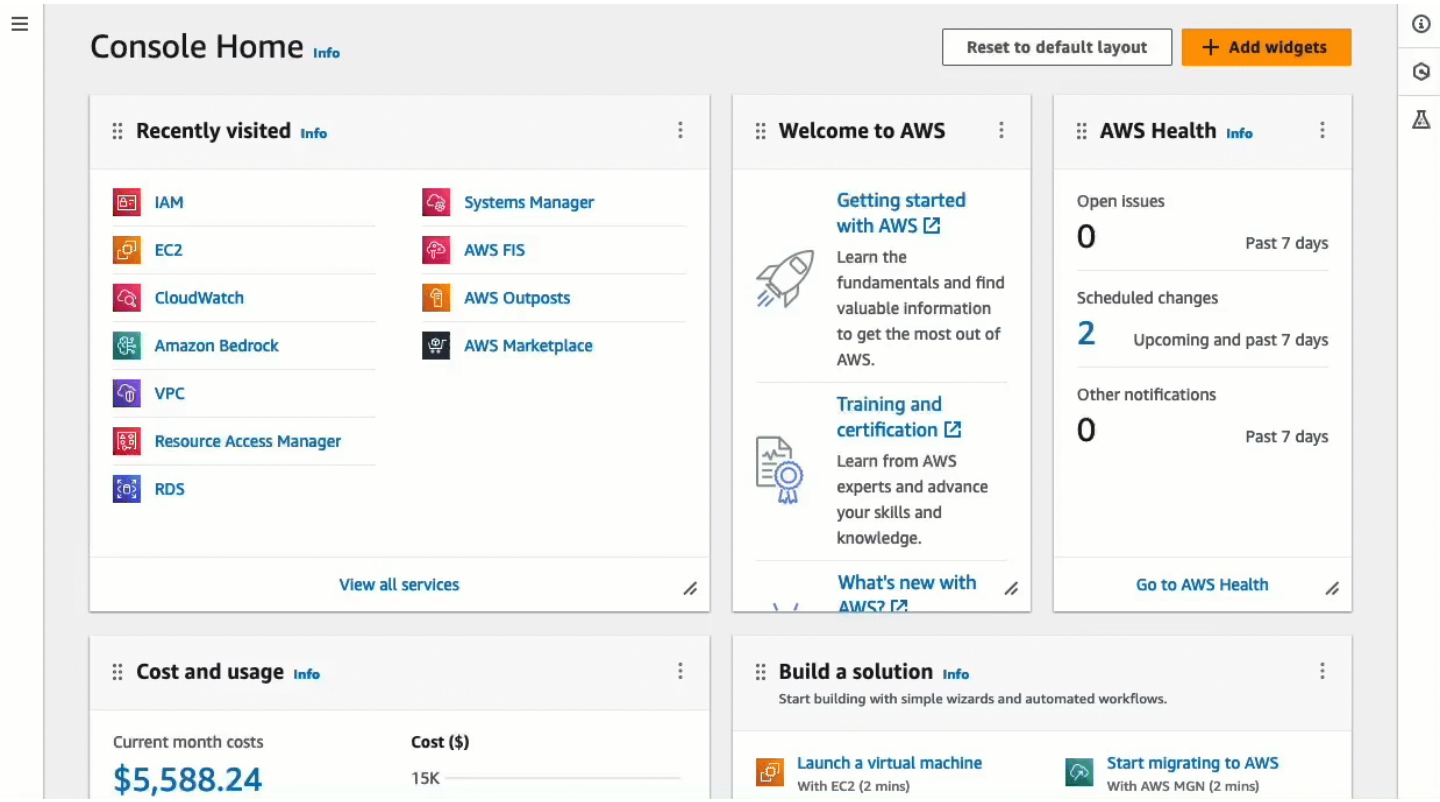
d. 在 IAM 實體，選取身分 (使用者、使用者群組或角色) 旁的核取方塊。您可以使用搜尋方塊來篩選實體清單。

e. 選擇連接政策。

檢視動畫：建立 IAM 政策



檢視動畫：連接 IAM 政策



任務 2：允許從 EC2 Instance Connect 服務到執行個體的傳入流量

當您在 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體時，必須允許到達執行個體的流量來自 EC2 Instance Connect 服務。這與從本機電腦連線到執行個體不同；在這種情況下，您必須允許從本機電腦到執行個體的流量。若要允許來自 EC2 Instance Connect 服務的流量，您必須建立安全群組，允許來自 EC2 Instance Connect 服務的 IP 位址範圍內的傳入 SSH 流量。

AWS 使用字首清單來管理 IP 地址範圍。EC2 Instance Connect 字首清單的名稱如下所示，並以區域代碼取代*##*：

- IPv4 字首清單名稱：`com.amazonaws.region.ec2-instance-connect`
- IPv6 字首清單名稱：`com.amazonaws.region.ipv6.ec2-instance-connect`

任務目標

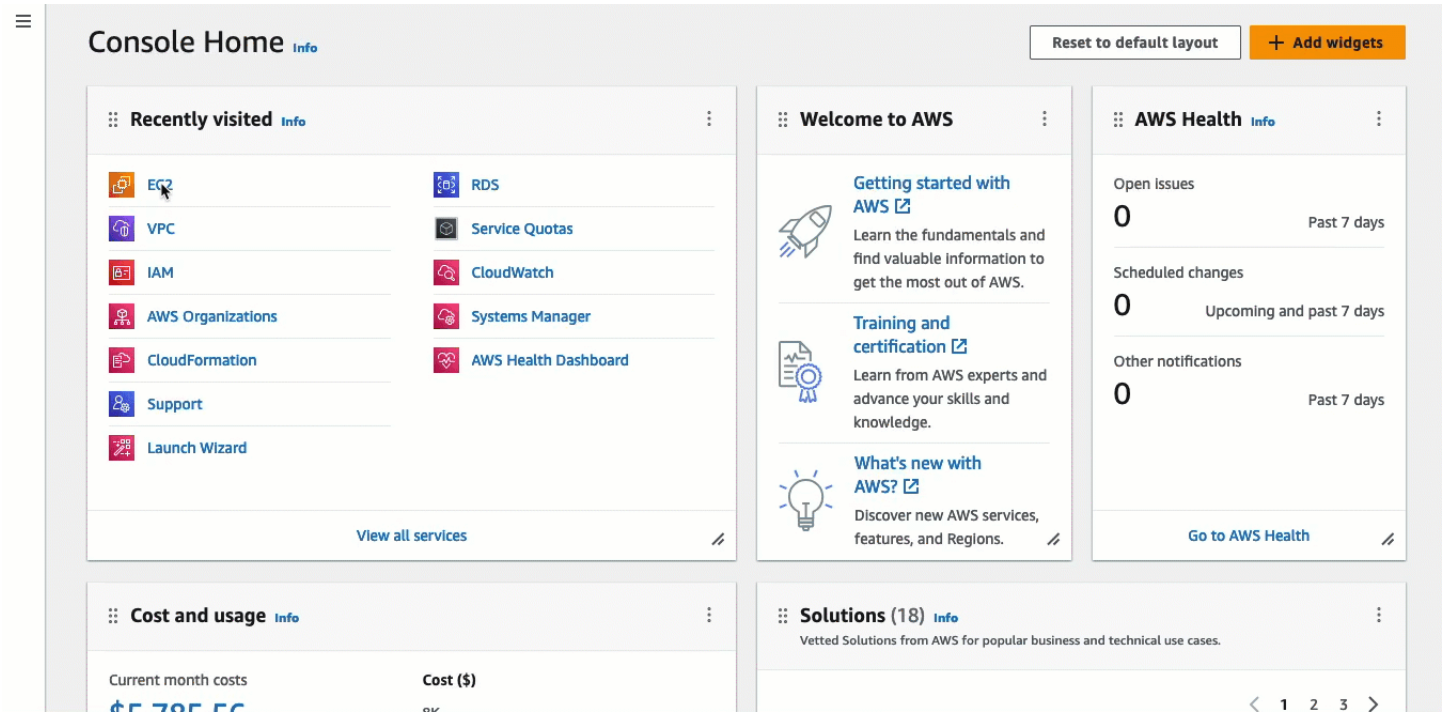
您將建立安全群組，允許來自執行個體所在區域 IPv4 字首清單的連接埠 22 上的傳入 SSH 流量。

建立安全群組，允許從 EC2 Instance Connect 服務到執行個體的傳入流量

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選擇 Create Security Group (建立安全群組)。
4. 在 Basic details (基本詳細資料) 下，執行下列動作：
   a. 在安全群組名稱，輸入有意義的安全群組名稱。
   b. 為描述輸入有意義的安全群組說明。
5. 在輸入規則，執行下列動作：
   a. 選擇新增規則。
   b. 針對 Type (類型)，選擇 SSH。
   c. 對於來源，請離開自訂。
   d. 在來源旁的欄位，選取 EC2 Instance Connect 的字首清單。

      例如，如果您的執行個體位於美國東部 (維吉尼亞北部) (`us-east-1`) 區域，且您的使用者將連線到其公有 IPv4 地址，請選擇下列字首清單：com.amazonaws.us-east-1.ec2-instance-connect
6. 選擇建立安全群組。

檢視動畫：建立安全群組



任務 3：啟動您的執行個體

當啟動執行個體時，您必須指定包含啟動執行個體所需資訊的 AMI。您可以選擇啟動預先安裝或未預先安裝 EC2 Instance Connect 的執行個體。在此任務中，您可以指定預先安裝 EC2 Instance Connect 的 AMI。

如果您在未預先安裝 EC2 Instance Connect 的情況啟動執行個體，並且想要使用 EC2 Instance Connect 連線到執行個體，則需要執行其他組態步驟。這些步驟已超出本教學課程範圍。

任務目標

您將使用 Amazon Linux 2023 AMI 啟動執行個體，該 AMI 已預先安裝 EC2 Instance Connect。您也將指定您先前建立的安全群組，以便可以在 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體。由於您將使用 EC2 Instance Connect 連線至執行個體，將公有金鑰推送至執行個體的中繼資料，因此啟動執行個體時不需要指定 SSH 金鑰。

啟動可在 Amazon EC2 主控台使用 EC2 Instance Connect 進行連線的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面頂端的導覽列中，會顯示目前的 AWS 區域（例如愛爾蘭)。選取要在其中啟動執行個體的區域。此選項非常重要，因為您建立了允許特定區域流量的安全群組，因此您必須選取要啟動執行個體的相同區域。

3. 從 Amazon EC2 主控台儀表板選擇 Launch Instance (啟動執行個體)。
4. (選用) 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入執行個體的描述性名稱。
5. 在應用程式和作業系統映像 (Amazon Machine Image) 下，選擇快速啟動。預設會選取 Amazon Linux。在 Amazon Machine Image (AMI)，預設選取 Amazon Linux 2023 AMI。保留此任務的預設選擇。
6. 在執行個體類型，針對執行個體類型，保留預設選擇，或選擇不同的執行個體類型。
7. 在金鑰對 (登入) 中，針對金鑰對名稱，選擇不使用金鑰對繼續 (不建議)。當使用 EC2 Instance Connect 連線到執行個體時，EC2 Instance Connect 會將金鑰對推送到執行個體的中繼資料，而這是用於連線的金鑰對。
8. 在Network settings (網路設定) 中，執行下列操作：

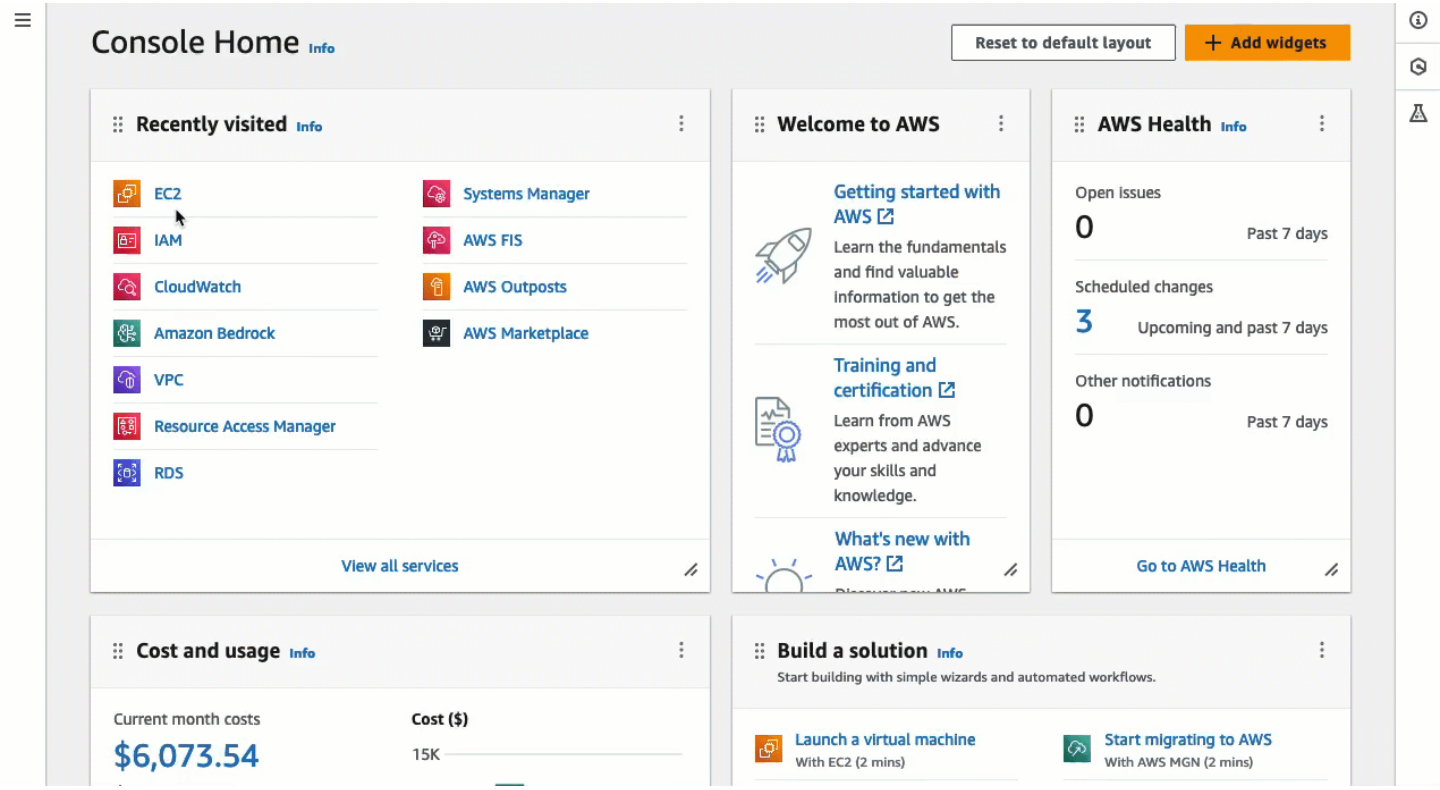
   a. 對於自動指派公有 IP，保留啟用。

   > **Note**
   >
   > 若要在 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體，執行個體必須具有公有 IPv4 或 IPv6 位址。

   b. 針對防火牆 (安全群組)，選擇選取現有的安全群組。

   c. 在一般安全群組下方，選擇您先前建立的安全群組。
9. 在 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

檢視動畫：啟動執行個體



任務 4：連線到您的執行個體

當您使用 EC2 Instance Connect 連接至執行個體時，EC2 Instance Connect API 會將 SSH 公有金鑰推送至執行個體中繼資料，此金鑰可保留 60 秒。SSH 常駐程式使用 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser` 從執行個體中繼資料查詢身分驗證所用的公有金鑰並將您連線至執行個體。

任務目標

在此任務，您將使用 Amazon EC2 主控台的 EC2 Instance Connect 連線到執行個體。如果您已完成先決任務 1、2 和 3，則連線應會成功。

連結執行個體的步驟

使用下列步驟連線至您的執行個體。若要檢視步驟動畫，請參閱 檢視動畫：連線至您的執行個體。

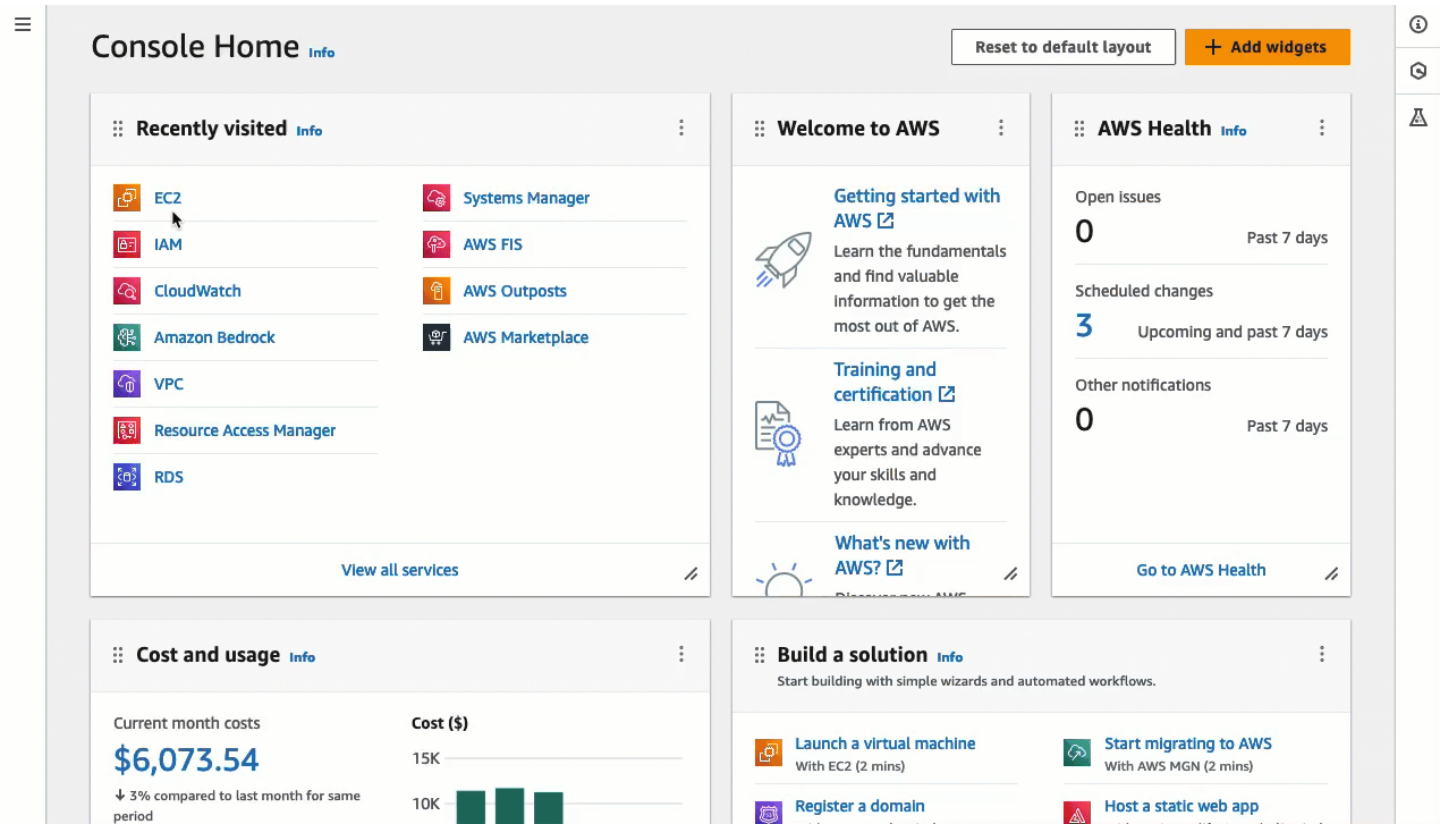
在 Amazon EC2 主控台使用 EC2 Instance Connect 連線執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在畫面頂端的導覽列中，會顯示目前的 AWS 區域 （例如愛爾蘭)。選取執行個體所在的區域。
3. 在導覽窗格中，選擇執行個體。

4. 選取執行個體，然後選擇連線。
5. 選擇 EC2 Instance Connect 索引標籤。
6. 請選擇使用公有 IP 建立連線。
7. 選擇連線。

   瀏覽器將開啟終端機視窗，並且您會連線到您的執行個體。

檢視動畫：連線至您的執行個體



EC2 Instance Connect 的先決條件

以下是安裝和使用 EC2 Instance Connect 的先決條件：

- 安裝 EC2 Instance Connect
- 確保網路連線
- 允許傳出流量到 EC2 Instance Connect 代理
- 允許傳入 SSH 流量
- 授予許可
- 在本機電腦安裝 SSH 用戶端

- 符合使用者名稱需求

安裝 EC2 Instance Connect

若要使用 EC2 Instance Connect 連線至執行個體，請務必安裝 EC2 Instance Connect。您可以使用預先安裝 EC2 Instance Connect 的 AMI 啟動執行個體，也可以在使用支援的 AMI 啟動之執行個體上安裝 EC2 Instance Connect。如需詳細資訊，請參閱在 EC2 執行個體上安裝 EC2 Instance Connect。

確保網路連線

執行個體可設定為允許使用者透過網際網路或執行個體的私有 IP 地址連線至您的執行個體。視您的使用者透過 EC2 Instance Connect 連線到執行個體的方式而定，您必須設定以下網路存取：

- 如果使用者將透過網際網路連線到執行個體，則該執行個體必須具有公有 IPv4 或 IPv6 位址，並且必須位於具有到網際網路路由的公有子網路。如果您尚未修改預設公有子網路，則它包含僅針對 IPv4 的網際網路路由，而不是針對 IPv6。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的使用網際網路閘道啟用 VPC 網際網路存取權。
- 如果您的使用者將透過執行個體的私有 IPv4 地址連線到您的執行個體，則您必須建立與 VPC 的私有網路連線，例如使用 AWS Direct Connect AWS Site-to-Site VPN或 VPC 對等互連，以便您的使用者可以連線到執行個體的私有 IP 地址。

如果您的執行個體沒有公有 IPv4 或 IPv6 位址，且您不想按上述說明設定網路存取權，則可將 EC2 Instance Connect 端點視為 EC2 Instance Connect 的替代方案。透過 EC2 Instance Connect 端點，您可以使用 SSH 或 RDP 連線到執行個體，即使執行個體沒有公有 IPv4 或 IPv6 位址。如需詳細資訊，請參閱使用 Amazon EC2 主控台連線至 Linux 執行個體。

允許傳出流量到 EC2 Instance Connect 代理

當您使用 Amazon EC2 主控台連線至執行個體時，您的瀏覽器會在連接埠 443 上建立 WebSocket 連線，以在區域特定的端點連線至 EC2 Instance Connect Proxy 服務。

端點格式取決於分割區：

- 商業 AWS 區域和 AWS GovCloud (US) 區域使用端點格式：`prod.region.oneclickv2-proxy.ec2.aws.dev`。
- 中國區域使用端點格式：`prod.region.oneclickv2-proxy.ec2.a2z.org.cn`。

下列範例顯示每個分割區的端點：

- 對於美國東部（維吉尼亞北部）區域，端點為 `prod.us-east-1.oneclickv2-proxy.ec2.aws.dev`。
- 對於中國（北京）區域，端點為 `prod.cn-north-1.oneclickv2-proxy.ec2.a2z.org.cn`。

> **Note**
>
> 您的瀏覽器必須能夠到達此端點。如果防火牆或 VPN 封鎖連線，連線會失敗。

允許傳入 SSH 流量

當使用 Amazon EC2 主控台連線至執行個體時

當使用者使用 Amazon EC2 主控台連線至執行個體時，必須允許到達執行個體的流量來自 EC2 Instance Connect 服務。該服務由特定的 IP 地址範圍識別，該範圍透過字首清單進行 AWS 管理。您必須建立允許來自 EC2 Instance Connect 服務的傳入 SSH 流量的安全群組。若要設定此項目，請針對傳入規則，在來源旁的欄位選取 EC2 Instance Connect 字首清單。

AWS 為每個 區域的 IPv4 和 IPv6 地址提供不同的受管字首清單。EC2 Instance Connect 字首清單的名稱如下所示，並以區域代碼取代*##*：

- IPv4 字首清單名稱：`com.amazonaws.region.ec2-instance-connect`
- IPv6 字首清單名稱：`com.amazonaws.region.ipv6.ec2-instance-connect`

有關建立安全群組的說明，請參閱 任務 2：允許從 EC2 Instance Connect 服務到執行個體的傳入流量。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的可用受 AWS管字首清單。

當使用 CLI 或 SSH 連線到執行個體時

請確認與執行個體關聯的安全群組在連接埠 22 上允許來自您 IP 地址或網路的傳入 SSH 流量。VPC 的預設安全群組根據預設並不允許傳入 SSH 流量。啟動執行個體精靈建立的安全群組預設會允許傳入 SSH 流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。

授予許可

您必須將所需許可授予給每個將使用 EC2 Instance Connect 連線到執行個體的 IAM 使用者。如需詳細資訊，請參閱為 EC2 Instance Connect 授予 IAM 許可。

在本機電腦安裝 SSH 用戶端

如果您的使用者將使用 SSH 進行連線，則必須確保其本機電腦具有 SSH 用戶端。

使用者的本機電腦很可能預設已安裝 SSH 用戶端。他們可以在命令列鍵入 ssh 來檢查 SSH 用戶端。若本機電腦無法識別該命令，可以安裝 SSH 用戶端。如需在 Linux 或 macOS X 上安裝 SSH 用戶端的相關資訊，請參閱 http://www.openssh.com。如需在 Windows 10 上安裝 SSH 用戶端的相關資訊，請參閱 Windows 中的 OpenSSH。

如果使用者僅使用 Amazon EC2 主控台連線至執行個體，就無需在本機電腦安裝 SSH 用戶端。

符合使用者名稱需求

使用 EC2 Instance Connect 連線至執行個體時，使用者名稱必須符合下列先決條件：

- 第一個字元：必須是字母 (A-Z、a-z)、數字 (0-9) 或底線 (_)
- 後續字元：可以是字母 (A-Z、a-z)、數字 (0-9) 或 @ . _ - 這些字元
- 長度下限：1 個字元
- 長度上限：31 個字元

為 EC2 Instance Connect 授予 IAM 許可

若要使用 EC2 Instance Connect 連線至執行個體，請務必建立 IAM 政策來授予使用者執行下列動作和條件的許可：

- `ec2-instance-connect:SendSSHPublicKey` 動作 – 准許將公有金鑰推送至執行個體。
- `ec2:osuser` 條件 – 指定可將公有金鑰推送至執行個體的 OS 使用者名稱。使用針對啟動執行個體的 AMI 的預設使用者名稱。AL2023 和 Amazon Linux 2 的預設使用者名稱為 `ec2-user`，而 Ubuntu 的名稱為 `ubuntu`。
- `ec2:DescribeInstances` 動作 – 使用 EC2 主控台時必需，因為包裝函數會呼叫此動作。使用者可能已有從另一個政策呼叫此動作的許可。
- `ec2:DescribeVpcs` 動作 – 連線至 IPv6 地址時為必要。

考慮限制對特定 EC2 執行個體的存取。否則，具有 `ec2-instance-connect:SendSSHPublicKey` 動作許可的所有 IAM 主體都可以連線到所有 EC2 執行個體。您可以指定資源 ARN 或將資源標籤做為條件索引鍵使用，來限制存取。

如需詳細資訊，請參閱適用於 Amazon EC2 Instance Connect 的動作、資源和條件金鑰。

如需有關建立和編輯 IAM 政策的資訊，請參閱「IAM 使用者指南」中的建立 IAM 政策。

允許使用者連線至特定執行個體

下列 IAM 政策會授予連線至特定執行個體的許可，由其資源 ARN 識別。

在下列 IAM 政策範例中，指定了下列動作和條件：

- 此 ec2-instance-connect:SendSSHPublicKey 動作准許使用者連線至兩個執行個體，由資源 ARN 指定。若要准許使用者連線至所有 EC2 執行個體，請以 * 萬用字元取代資源 ARN。
- 只有在連線時指定 *ami-username*，ec2:osuser 條件才會准許連線至執行個體。
- 指定 ec2:DescribeInstances 動作，准許使用主控台的使用者連線至執行個體。如果使用者僅使用 SSH 用戶端連線至執行個體，則可以省略 ec2:DescribeInstances。注意，ec2:Describe* API 動作不支援資源層級許可。因此，在 Resource 元素中必須包含 * 萬用字元。
- 指定 ec2:DescribeVpcs 動作，准許使用主控台透過 IPv6 地址連線至執行個體。如果您的使用者只會使用公有 IPv4 地址，可以省略 ec2:DescribeVpcs。注意，ec2:Describe* API 動作不支援資源層級許可。因此，在 Resource 元素中必須包含 * 萬用字元。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

允許使用者連線至具有特定標籤的執行個體

屬性型存取控制 (ABAC) 是一種授權策略，可根據可連接到使用者 AWS 和資源的標籤定義許可。您可使用資源標籤來控制對執行個體的存取。如需使用標籤控制對 AWS 資源的存取的詳細資訊，請參閱《IAM 使用者指南》中的控制對 AWS 資源的存取。

在下列 IAM 政策範例中，ec2-instance-connect:SendSSHPublicKey 動作准許使用者連線至任何執行個體 (由資源 ARN 中的 * 萬用字元表示)，條件是執行個體的資源標記中 key=tag-key 和 value=tag-value。

指定 ec2:DescribeInstances 動作，准許使用主控台的使用者連線至執行個體。如果使用者僅使用 SSH 用戶端連線至執行個體，則可以省略 ec2:DescribeInstances。注意，ec2:Describe* API 動作不支援資源層級許可。因此，在 Resource 元素中必須包含 * 萬用字元。

指定 ec2:DescribeVpcs 動作，准許使用主控台透過 IPv6 地址連線至執行個體。如果您的使用者只會使用公有 IPv4 地址，可以省略 ec2:DescribeVpcs。注意，ec2:Describe* API 動作不支援資源層級許可。因此，在 Resource 元素中必須包含 * 萬用字元。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/tag-key": "tag-value"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

在 EC2 執行個體上安裝 EC2 Instance Connect

若要使用 EC2 Instance Connect 連線至 Linux 執行個體，則該執行個體必須安裝 EC2 Instance Connect。安裝 EC2 Instance Connect 會在執行個體上設定 SSH 常駐程式。

如需 EC2 Instance Connect 套件的詳細資訊，請參閱 GitHub 網站上的aws/aws-ec2-instance-connect-config。

> **Note**
>
> 如果您對 SSH 身分驗證設定 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，則 EC2 Instance Connect 安裝不會進行更新。因此，您無法使用 EC2 Instance Connect。

安裝先決條件

在安裝 EC2 Instance Connect 之前，請確定您符合下列先決條件。

- 確認執行個體使用下列其中一項：
  - Amazon Linux 2 在 2.0.20190618 之前的版本 *
  - AL2023 最小 AMI 或 Amazon ECS 最佳化 AMI
  - CentOS Stream 8 和 9
  - 14.2.1 之前的 macOS Sonoma、13.6.3 之前的 Ventura，以及 12.7.2 之前的 Monterey *
  - Red Hat Enterprise Linux (RHEL) 8 和 9
  - Ubuntu 16.04 和 18.04 *

> **Tip**
>
> * 針對 Amazon Linux 2、macOS 與 Ubuntu：若使用比上述更新的版本來啟動執行個體，EC2 Instance Connect 已預先安裝，無須手動安裝。

- 驗證 EC2 Instance Connect 的一般先決條件。

  如需詳細資訊，請參閱EC2 Instance Connect 的先決條件。
- 確認在本機電腦上使用 SSH 用戶端連線至執行個體的先決條件。

  如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
- 取得執行個體的 ID。

  您可以使用 Amazon EC2 主控台取得執行個體的 ID (從執行個體 ID 資料欄中)。如果您願意的話，可以使用 describe-instances (AWS CLI) 或 Get-EC2Instance (AWS Tools for Windows PowerShell) 命令。

手動安裝 EC2 Instance Connect

> **Note**
>
> 如果您使用下列其中一個 AMI 啟動執行個體，則會預先安裝 EC2 Instance Connect，您可以略過此程序：
>
> - AL2023 標準 AMI
> - Amazon Linux 2 2.0.20190618 或更新版本
> - macOS Sonoma 14.2.1 或更新版本
> - macOS Ventura 13.6.3 或更新版本
> - macOS Monterey 12.7.2 或更新版本
> - Ubuntu 20.04 或更新版本

根據執行個體的作業系統，使用以下其中一種程序來安裝 EC2 Instance Connect。

Amazon Linux 2

若要在使用 Amazon Linux 2 啟動的執行個體上安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。

   將以下命令中的範例值取代為您自己的值。使用在您啟動執行個體時指派至該執行個體的 SSH 金鑰對與您在啟動執行個體所用的 AMI 預設使用者名稱。Amazon Linux 2 的預設使用者名稱為 `ec2-user`。

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. 在執行個體上安裝 EC2 Instance Connect 套件。

   ```
   [ec2-user ~]$ sudo yum install ec2-instance-connect
   ```

   您應會在 `/opt/aws/bin/` 資料夾中看到三個新指令碼：

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   eic_run_authorized_keys
   ```

3. (選用) 確認 EC2 Instance Connect 是否已順利安裝在您的執行個體上。

   ```
   [ec2-user ~]$ sudo less /etc/ssh/sshd_config
   ```

   如果 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser` 行包含以下值，則 EC2 Instance Connect 的安裝已順利完成：

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - `AuthorizedKeysCommand` 會設定 `eic_run_authorized_keys` 指令碼來從執行個體中繼資料查閱金鑰
   - `AuthorizedKeysCommandUser` 會將系統使用者設為 `ec2-instance-connect`

> **Note**
>
> 如果您先前已設定 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，EC2 Instance Connect 安裝不會變更這些值且您將無法使用 EC2 Instance Connect。

CentOS

若要在使用 CentOS 啟動的執行個體上安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。

   將以下命令中的範例值取代為您自己的值。使用在您啟動執行個體時指派至該執行個體的 SSH 金鑰對與您在啟動執行個體所用的 AMI 預設使用者名稱。若使用 CentOS，使用者名稱預設為 `centos` 或 `ec2-user`。

   ```
   $ ssh -i my_ec2_private_key.pem centos@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. 如果您使用 HTTP 或 HTTPS 代理伺服器，則必須在目前的 Shell 工作階段中設定 `http_proxy` 或 `https_proxy` 環境變數。

   如果您不使用代理伺服器，則可以略過此步驟。

   - 若使用 HTTP 代理伺服器，請執行下列命令：

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

   - 若使用 HTTPS 代理伺服器，請執行下列命令：

     ```
     $ export http_proxy=https://hostname:port
     $ export https_proxy=https://hostname:port
     ```

3. 執行下列命令，在執行個體上安裝 EC2 Instance Connect 套件。

適用於 CentOS 的 EC2 執行個體 Connect 組態檔案是以 Red Hat Package Manager (RPM) 套件提供，搭配以下項目適用的不同 RPM 套件：CentOS 8 和 CentOS 9，以及在 Intel /AMD (x86_64) 或 ARM (AArch64) 上執行的執行個體類型。

為您的作業系統和 CPU 架構使用命令區塊。

- CentOS 8

  Intel/AMD (x86_64)

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
  west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
  connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
  west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
  selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
  selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

  ARM (AArch64)

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
  west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
  connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
  west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
  selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
  selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

- CentOS 9

  Intel/AMD (x86_64)

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  ```

```
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

您應會在 /opt/aws/bin/ 資料夾中看到以下新指令碼：

```
eic_run_authorized_keys
```

4. (選用) 確認 EC2 Instance Connect 是否已順利安裝在您的執行個體上。

   - CentOS 8：

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-
     connect.conf
     ```

   - CentOS 9：

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，則 EC2 Instance Connect 的安裝已順利完成：

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand 會設定 eic_run_authorized_keys 指令碼來從執行個體中繼資料查閱金鑰
- AuthorizedKeysCommandUser 會將系統使用者設為 ec2-instance-connect

> **Note**
>
> 如果您先前已設定 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，EC2 Instance Connect 安裝不會變更這些值且您將無法使用 EC2 Instance Connect。

macOS

若要在使用 macOS 啟動的執行個體上安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。

   將以下命令中的範例值取代為您自己的值。使用在您啟動執行個體時指派至該執行個體的 SSH 金鑰對與您在啟動執行個體所用的 AMI 預設使用者名稱。macOS 執行個體的預設使用者名稱為 ec2-user。

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. 使用以下命令更新 Homebrew。更新將列出 Homebrew 已知的軟體。EC2 Instance Connect 套件是透過 macOS 執行個體上的 Homebrew 提供。如需詳細資訊，請參閱 更新 Amazon EC2 Mac 執行個體上的作業系統和軟體。

```
[ec2-user ~]$ brew update
```

3. 在執行個體上安裝 EC2 Instance Connect 套件。這會安裝軟體，並設定 sshd 以使用該軟體。

```
[ec2-user ~]$ brew install ec2-instance-connect
```

您應會在 /opt/aws/bin/ 資料夾中看到以下新指令碼：

```
eic_run_authorized_keys
```

4. (選用) 確認 EC2 Instance Connect 是否已順利安裝在您的執行個體上。

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
```

如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，則 EC2 Instance Connect 的安裝已順利完成：

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand 會設定 eic_run_authorized_keys 指令碼來從執行個體中繼資料查閱金鑰
- AuthorizedKeysCommandUser 會將系統使用者設為 ec2-instance-connect

> **Note**
>
> 如果您先前已設定 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，EC2 Instance Connect 安裝不會變更這些值且您將無法使用 EC2 Instance Connect。

RHEL

若要在使用 Red Hat Enterprise Linux (RHEL) 啟動的執行個體上安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。

將以下命令中的範例值取代為您自己的值。使用在您啟動執行個體時指派至該執行個體的 SSH 金鑰對與您在啟動執行個體所用的 AMI 預設使用者名稱。RHEL 的預設使用者名稱為 ec2-user 或 root。

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. 如果您使用 HTTP 或 HTTPS 代理伺服器，則必須在目前的 Shell 工作階段中設定 http_proxy 或 https_proxy 環境變數。

   如果您不使用代理伺服器，則可以略過此步驟。

   - 若使用 HTTP 代理伺服器，請執行下列命令：

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

   - 若使用 HTTPS 代理伺服器，請執行下列命令：

     ```
     $ export http_proxy=https://hostname:port
     $ export https_proxy=https://hostname:port
     ```

3. 執行下列命令，在執行個體上安裝 EC2 Instance Connect 套件。

   適用於 RHEL 的 EC2 執行個體 Connect 組態檔案是以 Red Hat Package Manager (RPM) 套件提供，搭配以下項目適用的不同 RPM 套件：RHEL 8 和 RHEL 9，以及在 Intel /AMD (x86_64) 或 ARM (AArch64) 上執行的執行個體類型。

   為您的作業系統和 CPU 架構使用命令區塊。

   - RHEL 8

     Intel/AMD (x86_64)

     ```
     [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
     [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
     west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
     connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
     connect.rpm
     ```

```
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- RHEL 9

  Intel/AMD (x86_64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

  ARM (AArch64)

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
```

```
connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

您應會在 /opt/aws/bin/ 資料夾中看到以下新指令碼：

```
eic_run_authorized_keys
```

4. (選用) 確認 EC2 Instance Connect 是否已順利安裝在您的執行個體上。

   - 針對 RHEL 8：

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-
     connect.conf
     ```

   - 針對 RHEL 9：

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，則 EC2 Instance Connect 的安裝已順利完成：

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - AuthorizedKeysCommand 會設定 eic_run_authorized_keys 指令碼來從執行個體中繼資料查閱金鑰
   - AuthorizedKeysCommandUser 會將系統使用者設為 ec2-instance-connect

> **Note**
>
> 如果您先前已設定 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，EC2 Instance Connect 安裝不會變更這些值且您將無法使用 EC2 Instance Connect。

Ubuntu

若要在使用 Ubuntu 16.04 或更新版本啟動的執行個體上安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。

   將以下命令中的範例值取代為您自己的值。使用在您啟動執行個體時指派至該執行個體的 SSH 金鑰對並使用您在啟動執行個體所用的 AMI 預設使用者名稱。若為 Ubuntu AMI，使用者名稱為 `ubuntu`。

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. (選用) 請確認您的執行個體具有最新的 Ubuntu AMI。

   執行下列命令以更新您的執行個體上的所有套件。

   ```
   ubuntu:~$ sudo apt-get update
   ```

   ```
   ubuntu:~$ sudo apt-get upgrade
   ```

3. 在執行個體上安裝 EC2 Instance Connect 套件。

   ```
   ubuntu:~$ sudo apt-get install ec2-instance-connect
   ```

   您應會在 `/usr/share/ec2-instance-connect/` 資料夾中看到三個新指令碼：

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   ```

```
eic_run_authorized_keys
```

4. (選用) 確認 EC2 Instance Connect 是否已順利安裝在您的執行個體上。

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，則 EC2 Instance Connect 的安裝已順利完成：

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %
%u %%f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand 會設定 eic_run_authorized_keys 指令碼來從執行個體中繼資料查閱金鑰
- AuthorizedKeysCommandUser 會將系統使用者設為 ec2-instance-connect

> **Note**
>
> 如果您先前已設定 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，EC2 Instance Connect 安裝不會變更這些值且您將無法使用 EC2 Instance Connect。

使用 EC2 Instance Connect 連線至 Linux 執行個體

下列指示說明如何透過 Amazon EC2 主控台 AWS CLI、 或 SSH 用戶端，使用 EC2 Instance Connect 連線至 Linux 執行個體。 Amazon EC2

當您透過主控台或 使用 EC2 Instance Connect 連線到執行個體時 AWS CLI，EC2 Instance Connect API 會自動將 SSH 公有金鑰推送至執行個體中繼資料，並在其中保留 60 秒。連接至您使用者的 IAM 政策會授權此動作。如果您偏好使用自己的 SSH 金鑰，則可以使用 SSH 用戶端，並使用 EC2 Instance Connect 將 SSH 金鑰明確推送至執行個體。

考量事項

在使用 EC2 Instance Connect 連線至執行個體之後，SSH 工作階段終止之前，連線會持續存在。連線的持續時間並不是由 IAM 憑證的持續時間決定。若 IAM 憑證到期，連線會繼續存在。使用 EC2

Instance Connect 主控台體驗時，若 IAM 憑證到期，可藉由關閉瀏覽器頁面來終止連線。若使用您自己的 SSH 用戶端與 EC2 Instance Connect 來推送金鑰，您可設定 SSH 逾時值，以自動終止 SSH 工作階段。

要求

在開始之前，請務必檢閱先決條件。

連線選項

- 使用 Connect Amazon EC2 主控台進行連線
- 使用 連線 AWS CLI
- 使用您自己的金鑰和 SSH 用戶端來進行連接
- 疑難排解

使用 Connect Amazon EC2 主控台進行連線

您可以透過 Amazon EC2 主控台使用 EC2 Instance Connect 連線到執行個體。

要求

若要使用 Amazon EC2 主控台進行連線，執行個體必須具有公用 IPv4 或 IPv6 位址。如果執行個體只有私有 IPv4 地址，您可以使用 ec2-instance-connect AWS CLI 進行連線。

要使用 Amazon EC2 主控台連線至執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Connect (連線)。
4. 選擇 EC2 Instance Connect 索引標籤。
5. 請選擇使用公有 IP 建立連線。
6. 如果有選擇，請選取要連線的 IP 位址。否則，系統會自動選取 IP 位址。
7. 針對使用者名稱，請驗證使用者名稱。
8. 選擇連線以建立連線。瀏覽器內的終端機視窗隨即開啟。

使用 連線 AWS CLI

您可以使用 ec2-instance-connect AWS CLI 透過 SSH 用戶端連線至執行個體。EC2 Instance Connect 會根據指定的連線類型，嘗試以預先定義的順序使用可用的 IP 位址建立連線。如果 IP 位址無法使用，系統會自動嘗試排序的下一個 IP 位址。

連線類型

`auto` (default)

EC2 Instance Connect 會嘗試依下列順序使用執行個體的 IP 位址並使用對應的連線類型進行連線：

1. 公有 IPv4：`direct`
2. 私有 IPv4：`eice`
3. IPv6：`direct`

`direct`

EC2 Instance Connect 會嘗試依下列順序使用執行個體的 IP 位址進行連線：

1. 公有 IPv4
2. IPv6
3. 私有 IPv4 (不會透過 EC2 Instance Connect 端點連線)

`eice`

EC2 Instance Connect 會嘗試使用執行個體的私有 IPv4 地址及 EC2 Instance Connect 端點進行連線。

> **Note**
>
> 未來，`auto`連線類型的行為可能會變更。若要確保使用所需的連線類型，請明確`--connection-type`將 設定為 `direct`或 `eice`。

要求

您必須使用第 2 AWS CLI 版。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI。

使用執行個體 ID 連線到執行個體

如果您只知道執行個體 ID，並希望讓 EC2 Instance Connect 決定連線到執行個體時要使用的連線類型，則請使用執行個體 ID 並搭配 ec2-instance-connect ssh CLI 命令。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example
```

使用執行個體 ID 和 EC2 Instance Connect 端點連線至執行個體

如果要透過 EC2 Instance Connect 端點連線到執行個體，則請使用上述命令並指定具有 `eice` 值的 `--connection-type` 參數。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --connection-type eice
```

使用執行個體 ID 和您自己的私有金鑰檔案連線至執行個體

如果想透過 EC2 Instance Connect 端點使用自己的私有金鑰連線到執行個體，則請指定執行個體 ID 和私有金鑰檔案的路徑。請勿在路徑中包含 *file://*；下列範例會無效：*file:///path/to/key*。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --private-key-file /
path/to/key.pem
```

> **Tip**
>
> 如果您在使用這些命令時發生錯誤，請確定您使用的是第 2 AWS CLI 版，因為該`ssh`命令僅適用於此主要版本。我們也建議定期更新至 AWS CLI 版本 2 的最新次要版本，以存取最新功能。如需相關資訊，請參閱「AWS Command Line Interface 使用者指南」中的關於 AWS CLI 第 2 版。

使用您自己的金鑰和 SSH 用戶端來進行連接

當您使用 EC2 Instance Connect API 時，您可以使用自己的 SSH 金鑰和從您選擇的 SSH 用戶端連接執行個體。這可讓您從 EC2 Instance Connect 發佈公有的金鑰至執行個體。此連線方法適用於具有公有和私有 IP 地址的執行個體。

要求

- 金鑰對的需求
  - 支援的類型：RSA (OpenSSH 和 SSH2) 和 ED25519

  - 支援的長度：2048 和 4096
  - 如需詳細資訊，請參閱使用第三方工具來建立金鑰對，然後將公有金鑰匯入 Amazon EC2。
- 連線到僅具有私有 IP 位址的執行個體時，從中起始 SSH 工作階段的本機電腦必須具有連線到 EC2 Instance Connect 服務端點的能力 (以將 SSH 公有金鑰推送至執行個體)，以及透過網路連線到執行個體私有 IP 位址的能力，以建立 SSH 工作階段。EC2 Instance Connect 服務端點可透過網際網路或 Direct Connect 公有虛擬界面存取。若要連線到執行個體的私有 IP 地址，您可以使用服務，例如 Direct Connect、AWS Site-to-Site VPN 或 VPC 互連。

使用您自己的金鑰和任何 SSH 用戶端來連接制執行個體

1. (選用) 產生新 SSH 私有和公有金鑰

   您可以使用以下命令來產生新 SSH 私有和公有金鑰 (my_key 和 my_key.pub)：

   ```
   ssh-keygen -t rsa -f my_key
   ```

2. 將 SSH 公有金鑰推送至執行個體

   使用 send-ssh-public-key 命令將 SSH 公開金鑰推送至執行個體。如果使用 AL2023 或 Amazon Linux 2 啟動執行個體，AMI 的預設使用者名稱為 ec2-user。如果您是使用 Ubuntu 啟動執行個體，AMI 的預設使用者名稱為 ubuntu。

   下列範例將公開金鑰推送至特定可用區域的特定執行個體，若要認證 ec2-user。

   ```
   aws ec2-instance-connect send-ssh-public-key \
       --region us-west-2 \
       --availability-zone us-west-2b \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --instance-os-user ec2-user \
       --ssh-public-key file://my_key.pub
   ```

3. 使用私有金鑰連線至執行個體

   使用 ssh 命令，來使用私有金鑰連接至執行個體，之後系統會將公有金鑰從執行個體中繼資料中移除 (在系統移除金鑰前您有 60 秒的時間)。指定對應於公有金鑰應的私有金鑰、用於啟動執行個體之 AMI 的預設使用者名稱，以及執行個體的公有 DNS 名稱 (如果透過私有網路連線，請指定私有 DNS 名稱或 IP 位址)。新增該 IdentitiesOnly=yes 選項以確保連線只使用 ssh 組態中的檔案和指定的金鑰。

```
ssh -o "IdentitiesOnly=yes" -i my_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

下列範例使用 `timeout 3600` 將 SSH 工作階段設定為 1 小時後終止。工作階段終止之後，工作階段期間啟動的程序可能會在執行個體上繼續執行。

```
timeout 3600 ssh -o “IdentitiesOnly=yes” -i my_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

疑難排解

如果您在嘗試連線至執行個體時遇到錯誤，請參閱下文：

- 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解
- 如何對使用 EC2 Instance Connect 連線至 EC2 執行個體的問題進行疑難排解？

解除安裝 EC2 Instance Connect

若要停用 EC2 Instance Connect，連接至您的 Linux 執行個體並解除安裝在 OS 上的 `ec2-instance-connect` 套件。如果 `sshd` 組態與您安裝在 EC2 Instance Connect 設定的內容相符，解除安裝 `ec2-instance-connect` 也會移除 `sshd` 組態。如果您在安裝 EC2 Instance Connect 之後修改 `sshd` 組態，您必須手動更新。

Amazon Linux

您可以在 AL2023 和 Amazon Linux 2 2.0.20190618 或更新版本中 (EC2 Instance Connect 已預先設定) 解除安裝 EC2 Instance Connect。

若要在使用 Amazon Linux 啟動的執行個體解除安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。指定啟動執行個體時所使用的 SSH 金鑰對，以及 AL2023 或 Amazon Linux 2 AMI 的預設使用者名稱，即 `ec2-user`。

   例如，以下 ssh 命令會使用 `my_ec2_private_key.pem` 金鑰對連接到公有 DNS 名稱為 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com` 的執行個體。

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

2. 使用 yum 指令，解除安裝 ec2-instance-connect 套件。

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

Ubuntu

若要在使用 Ubuntu AMI 啟動的執行個體上解除安裝 EC2 Instance Connect

1. 使用 SSH 連接至您的執行個體。當您啟動且為 Ubuntu AMI 為預設使用者名稱時，為您的執行個體指定您所使用的 SSH 金鑰對 ubuntu。

   例如，以下 ssh 命令會使用 my_ec2_private_key.pem 金鑰對連接到公有 DNS 名稱為 ec2-a-b-c-d.us-west-2.compute.amazonaws.com 的執行個體。

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

2. 使用 apt-get 指令，解除安裝 ec2-instance-connect 套件。

```
ubuntu:~$ sudo apt-get remove ec2-instance-connect
```

使用私有 IP 位址與 EC2 Instance Connect 端點連線至您的執行個體

EC2 Instance Connect 端點可讓您安全地從網際網路連線至執行個體，而無需使用堡壘主機，您的虛擬私有雲端 (VPC) 也無須具備直接網際網路連線。

優勢

- 您可以連線到執行個體，而無需執行個體擁有公有 IPv4 或 IPv6 地址。 會收取所有公有 IPv4 地址 AWS 的費用，包括與執行中執行個體相關聯的公有 IPv4 地址和彈性 IP 地址。如需詳細資訊，請參閱 Amazon VPC 定價頁面中的公有 IPv4 地址。
- 您可以從網際網路連線至執行個體，VPC 無須具有直接網際網路連線的網際網路閘道。
- 您可以控制對建立和使用 EC2 Instance Connect 端點的存取，以連線至使用 IAM 政策和許可的執行個體。

- 連線到您執行個體的所有嘗試 (無論是成功還是失敗) 都會記錄到 CloudTrail。

定價

使用 EC2 Instance Connect 端點無需額外費用。如果使用 EC2 Instance Connect 端點連線至位於不同可用區域中的執行個體，則跨可用區域的資料傳輸需額外付費。

目錄

- 運作方式
- 考量事項
- 授予許可以使用 EC2 Instance Connect 端點
- EC2 Instance Connect 端點的安全群組
- 建立 EC2 Instance Connect 端點
- 修改 EC2 Instance Connect 端點
- 刪除 EC2 Instance Connect 端點
- 使用 EC2 Instance Connect 端點連線至 Amazon EC2 執行個體
- 記錄透過 EC2 Instance Connect 端點建立的連線
- EC2 Instance Connect 端點的服務連結角色
- EC2 Instance Connect 端點配額

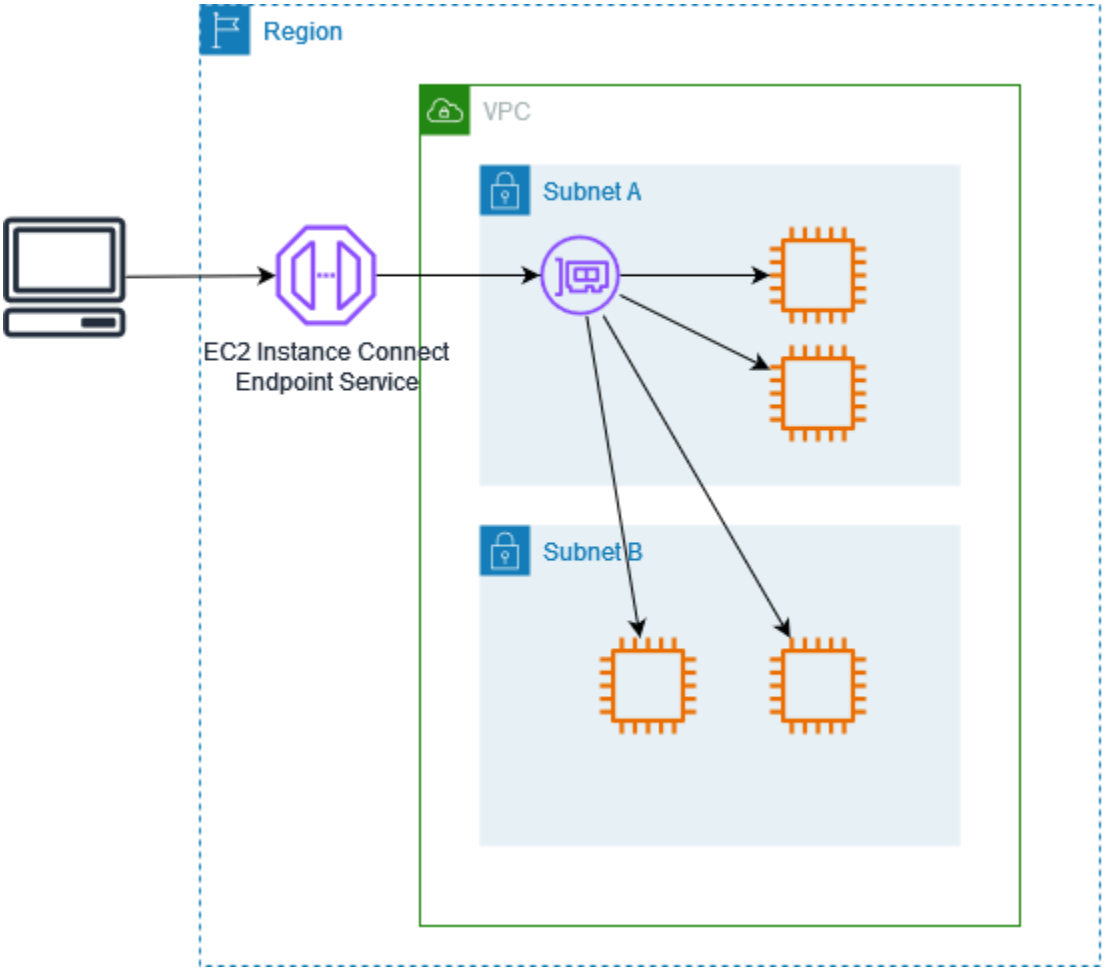
運作方式

EC2 Instance Connect 端點是身分感知的 TCP 代理。EC2 Instance Connect 端點服務會使用 IAM 實體的登入資料，從您的電腦建立連接至端點的私有通道。流量會在到達 VPC 之前經過身分驗證和授權。

您可以設定其他安全群組規則，以限制傳入至執行個體的流量。例如，您可以使用傳入規則，以允許來自 EC2 Instance Connect 端點的管理連接埠上的流量。

您可以設定路由表規則，以允許端點連線到 VPC 任何子網路中的任何執行個體。

下圖顯示使用者如何使用 EC2 Instance Connect 端點從網際網路連線至其執行個體。首先，在子網路 A 中建立 EC2 Instance Connect 端點。我們將為子網路中的端點建立網路介面，並以該介面做為流量進入點，目的地是您 VPC 中執行個體。如果子網路 B 的路由表允許來自子網路 A 的流量，則您可以使用端點來連接子網路 B 中的執行個體。



考量事項

開始之前，請考慮以下項目。

- EC2 Instance Connect 端點特別適用於管理流量使用案例，而不適用於大量資料傳輸。系統會限流大量資料傳輸。
- 您可建立 EC2 Instance Connect 端點，以便為擁有私有 IPv4 位址或 IPv6 位址的執行個體的流量提供支援。端點的 IP 位址類型必須與執行個體的 IP 位址相符。您可建立支援全部 IP 位址類型的端點。
- (Linux 執行個體) 若使用自己的金鑰對，則可以使用任何 Linux AMI。否則，您的執行個體必須安裝 EC2 Instance Connect。如需哪些 AMI 包含 EC2 Instance Connect，以及如何在其他支援的 AMI 上進行安裝的相關資訊，請參閱安裝 EC2 Instance Connect。
- 您可指派安全群組至 EC2 Instance Connect 端點。否則，VPC 將使用我們的預設安全群組。EC2 Instance Connect 端點的安全群組必須允許傳出流量到目的地執行個體。如需詳細資訊，請參閱EC2 Instance Connect 端點的安全群組。

- 您可以設定 EC2 Instance Connect 端點，以便在請求路由至執行個體時，保留用戶端的來源 IP 位址。否則，網路介面的 IP 位址會成為所有傳入流量的用戶端 IP 位址。
  - 如果您開啟用戶端 IP 保留，執行個體的安全群組必須允許來自用戶端的流量。此外，執行個體必須與 EC2 Instance Connect 端點位於相同的 VPC 中。
  - 如果您關閉用戶端 IP 保留，執行個體的安全群組必須允許來自 VPC 的流量。這是預設值。
  - 僅在 IPv4 EC2 Instance Connect 端點上支援用戶端 IP 保留。如需使用用戶端 IP 保留，EC2 Instance Connect 端點的 IP 位址類型必須為 IPv4。若 IP 位址類型為雙重堆疊或 IPv6，則用戶端 IP 保留不受支援。
  - 以下執行個體類型不支援用戶端 IP 保留：C1、CG1、CG2、G1、HI1、M1、M2、M3 和 T1。如果您開啟用戶端 IP 保留，並嘗試使用 EC2 Instance Connect 端點連線到以上類型的執行個體，連線會失敗。
  - 當透過傳輸閘道傳送流量時，不支援用戶端 IP 保留。
- 當您建立 EC2 Instance Connect Endpoint 時，會自動為 AWS Identity and Access Management (IAM) 中的 Amazon EC2 服務建立服務連結角色。Amazon EC2 使用服務連結角色在網路介面中佈建帳戶，這在建立 EC2 Instance Connect 端點時是必需的。如需詳細資訊，請參閱EC2 Instance Connect 端點的服務連結角色。
- 每個 VPC 和每個子網路只能建立 1 個 EC2 Instance Connect 端點。如需詳細資訊，請參閱EC2 Instance Connect 端點配額。如果您需要在相同 VPC 的不同可用區域中建立另一個 EC2 Instance Connect 端點，您必須先刪除現有的 EC2 Instance Connect 端點。否則，您將收到配額錯誤。
- 每個 EC2 執行個體連線端點最多可支援 20 個並行連線。
- 已建立 TCP 連線的持續時間上限是 1 小時 (3600 秒)。您可以在 IAM 政策中指定允許的持續時間上限，上限為 3600 秒。如需詳細資訊，請參閱使用 EC2 Instance Connect 端點連線到執行個體所需的許可。

  連線的持續時間並不是由 IAM 憑證的持續時間決定。若 IAM 憑證到期，達到指定持續時間上限之前，連線會繼續存在。若使用 EC2 Instance Connect 端點主控台體驗連線至執行個體，則將通道持續時間上限 (秒) 設定為小於 IAM 憑證持續時間的值。若 IAM 憑證提前到期，可藉由關閉瀏覽器頁面來終止執行個體連線。

授予許可以使用 EC2 Instance Connect 端點

依預設，IAM 實體沒有建立、描述或修改 EC2 Instance Connect 端點的許可。IAM 管理員可以建立 IAM 政策，授予在所需資源上執行特定動作所需的權限。

如需有關建立和編輯 IAM 政策的資訊，請參閱「IAM 使用者指南」中的建立 IAM 政策。

下列範例政策示範您如何控制使用者對 EC2 Instance Connect 端點所需的許可。

範例

- 建立、描述、修改和刪除 EC2 Instance Connect 端點所需的許可
- 使用 EC2 Instance Connect 端點連線到執行個體所需的許可
- 僅從特定 IP 位址範圍連線的許可

建立、描述、修改和刪除 EC2 Instance Connect 端點所需的許可

若要建立及修改 EC2 Instance Connect 端點，使用者需要下列動作的許可：

- `ec2:CreateInstanceConnectEndpoint`
- `ec2:CreateNetworkInterface`
- `ec2:CreateTags`
- `ec2:ModifyInstanceConnectEndpoint`
- `iam:CreateServiceLinkedRole`

若要描述和刪除 EC2 Instance Connect 端點，使用者需要下列動作的許可：

- `ec2:DescribeInstanceConnectEndpoints`
- `ec2:DeleteInstanceConnectEndpoint`

您可以建立一個政策，以准許在所有子網路中建立、描述、修改和刪除 EC2 Instance Connect 端點。或者，僅將子網路 ARN 指定為允許的 `Resource` 或使用 `ec2:SubnetID` 條件索引鍵，來限制指定子網路的動作。您也可以使用 `aws:ResourceTag` 條件索引鍵來明確允許或拒絕具有特定標籤的端點建立。如需詳細資訊，請參閱「IAM 使用者指南」中的 IAM 中的政策和許可。

IAM 政策範例

在下列 IAM 政策範例中，`Resource` 區段准許在所有子網路中建立、修改和刪除端點，以星號 (*) 指定。`ec2:Describe*` API 動作不支援資源層級許可。因此，在 `Resource` 元素中必須包含 * 萬用字元。

使用 EC2 Instance Connect 端點連線到執行個體所需的許可

`ec2-instance-connect:OpenTunnel` 動作准許建立執行個體的 TCP 連線，以便透過 EC2 Instance Connect 端點進行連線。您可以指定要使用的 EC2 Instance Connect 端點。或者，帶有星號

(*) 的 Resource 允許使用者使用任何可用的 EC2 Instance Connect 端點。您也可以根據是否存在作為條件索引鍵的資源標籤來限制執行個體的存取。

條件

- ec2-instance-connect:remotePort – 執行個體上可用來建立 TCP 連線的連接埠。使用此條件索引鍵時，嘗試連線到政策中指定的連接埠以外的任何其他連接埠上的執行個體會導致失敗。
- ec2-instance-connect:privateIpAddress – 與您要建立 TCP 連線的執行個體相關聯的目的地私有 IP 位址。您可以指定單一 IP 位址 (例如 10.0.0.1/32) 或透過 CIDR 指定 IP 範圍 (例如 10.0.1.0/28)。使用此條件索引鍵時，嘗試連線至具有不同私有 IP 位址或超出 CIDR 範圍的執行個體會導致失敗。
- ec2-instance-connect:maxTunnelDuration – 已建立的 TCP 連線的持續時間上限。單位為秒，持續時間範圍為至少 1 秒至最多 3600 秒 (1 小時)。如果未指定條件，則預設持續時間會設為 3600 秒 (1 小時)。嘗試連線至執行個體的時間超過 IAM 政策中指定的持續時間，或超過預設最大值時，會導致失敗。連線會在指定的持續時間後中斷。

  如果在 IAM 政策中指定 maxTunnelDuration，且指定的值少於 3,600 秒 (預設值)，則您必須在連線至執行個體時在命令中指定 --max-tunnel-duration。如需有關如何連線至執行個體的資訊，請參閱 使用 EC2 Instance Connect 端點連線至 Amazon EC2 執行個體。

您也可以根據 EC2 Instance Connect 端點上的資源標籤存在情況，授予使用者建立執行個體連線的存取權。如需詳細資訊，請參閱「IAM 使用者指南」中的 IAM 中的政策和許可。

針對 Linux 執行個體，ec2-instance-connect:SendSSHPublicKey 動作准許將公有金鑰推送至執行個體。ec2:osuser 條件指定可將公有金鑰推送至執行個體的 OS (作業系統) 使用者名稱。使用針對啟動執行個體的 AMI 的預設使用者名稱。如需詳細資訊，請參閱為 EC2 Instance Connect 授予 IAM 許可。

IAM 政策範例

以下 IAM 政策範例允許 IAM 主體僅使用指定的 EC2 Instance Connect 端點(由指定的端點 ID eice-123456789abcdef 識別) 連線到執行個體。必須滿足所有條件，連線才會成功建立。

> **Note**
>
> ec2:Describe* API 動作不支援資源層級許可。因此，在 Resource 元素中必須包含 * 萬用字元。

Linux

此範例會評估連接埠 22 (SSH) 是否已與執行個體建立連線，如果執行個體的私有 IP 位址位於 10.0.1.0/31 (介於 10.0.1.0和 之間10.0.1.1) 的範圍內，且 maxTunnelDuration 小於或等於 3600 秒。連線會在 3600 秒 (1 小時) 後中斷。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Windows

此範例會評估連接埠 3389 (RDP) 是否已與執行個體建立連線，如果執行個體的私有 IP 位址位於 10.0.1.0/31 (介於 10.0.1.0和 之間10.0.1.1) 的範圍內，且 maxTunnelDuration 小於或等於 3600 秒。連線會在 3600 秒 (1 小時) 後中斷。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "3389"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
```

```
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

僅從特定 IP 位址範圍連線的許可

下列 IAM 政策範例允許 IAM 主體連線到執行個體，條件是它們從政策中指定的 IP 位址範圍內的 IP 位址進行連線。如果 IAM 主體從不在 192.0.2.0/24 範圍內的 IP 位址呼叫 OpenTunnel (此政策中的範例 IP 位址範圍)，則回應為 Access Denied。如需詳細資訊，請參閱「IAM 使用者指南」中的 aws:SourceIp。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                }
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Resource": "*"
        }
    ]
}
```

EC2 Instance Connect 端點的安全群組

安全群組負責控制允許到達和離開其關聯資源的流量。例如，除非與執行個體相關聯的安全群組特別允許，否則我們會拒絕往返 Amazon EC2 執行個體的流量。

下列範例示範如何設定 EC2 Instance Connect 端點和目標執行個體的安全群組規則。

範例

- EC2 Instance Connect 端點安全群組規則
- 目標執行個體安全群組規則

EC2 Instance Connect 端點安全群組規則

EC2 Instance Connect 端點的安全群組規則必須允許目標執行個體的傳出流量離開端點。您可以指定執行個體安全群組或 VPC 的 IPv4 或 IPv6 位址範圍做為目的地。

到端點的流量來自 EC2 Instance Connect 端點服務，無論端點安全群組的傳入規則為何，都會獲得允許。若要控制誰可以使用 EC2 Instance Connect 端點連線到執行個體，請使用 IAM 政策。如需詳細資訊，請參閱使用 EC2 Instance Connect 端點連線到執行個體所需的許可。

傳出規則範例：安全群組參考

下列範例使用安全群組參考，這表示目的地是與目標執行個體相關聯的安全群組。此規則允許從端點向使用此安全群組的所有執行個體的傳出流量。

通訊協定	目標	連接埠範圍	Comment
TCP	*######### ID*	22	允許與執行個體安全群組相關的所有執行個體的傳出 SSH 流量

傳出規則範例：IPv4 地址範圍

下列範例允許傳出流量到指定的 IPv4 地址範圍。執行個體的 IPv4 地址是從其子網路指派，因此您可以使用 VPC 的 IPv4 地址範圍。

通訊協定	目標	連接埠範圍	Comment
TCP	*VPC A IPv4 CIDR*	22	允許傳出 SSH 流量到 VPC

傳出規則範例：IPv6 地址範圍

下列範例允許傳出流量到指定的 IPv6 地址範圍。執行個體的 IPv6 地址是從其子網路指派，因此您可以使用 VPC 的 IPv6 地址範圍。

通訊協定	目標	連接埠範圍	Comment
TCP	*VPC A IPv6 CIDR*	22	允許傳出 SSH 流量到 VPC

目標執行個體安全群組規則

目標執行個體的安全群組規則必須允許來自 EC2 Instance Connect 端點的傳入流量。您可以指定端點安全群組或 IPv4 或 IPv6 位址範圍做為來源。如果您指定 IPv4 地址範圍，來源將取決於用戶端 IP 保留是否關閉或開啟。如需詳細資訊，請參閱考量事項。

由於安全群組具有狀態，因此無論執行個體安全群組的傳出規則為何，回應流量都可以離開 VPC。

傳入規則範例：安全群組參考

下列範例使用安全群組參考，這表示來源是與端點相關聯的安全群組。此規則允許從端點到所有使用此安全群組的執行個體的傳入 SSH 流量，不論用戶端 IP 保留為開啟或關閉。如果沒有 SSH 的其他傳入安全群組規則，則執行個體只會接受來自端點的 SSH 流量。

通訊協定	來源	連接埠範圍	Comment
TCP	*####### ID*	22	允許來自與端點安全群組相關聯資源的傳入 SSH 流量

傳入規則範例：用戶端 IP 保留關閉

下列範例允許傳入 SSH 流量自指定的 IPv4 地址範圍。由於用戶端 IP 保留已關閉，來源 IPv4 地址會是端點網路介面的地址。端點網路介面的地址是由其子網路指派，因此您可以使用 VPC 的 IPv4 地址範圍來允許與 VPC 中所有執行個體的連線。

通訊協定	來源	連接埠範圍	Comment
TCP	*VPC A IPv4 CIDR*	22	允許來自 VPC 的傳入 SSH 流量。

傳入規則範例：用戶端 IP 保留開啟

下列範例允許傳入 SSH 流量自指定的 IPv4 地址範圍。由於用戶端 IP 保留已開啟，來源 IPv4 地址會是用戶端的地址。

通訊協定	來源	連接埠範圍	Comment
TCP	*## IPv4 ####*	22	允許傳入流量自指定的用戶端 IPv4 地址範圍

建立 EC2 Instance Connect 端點

您可以建立 EC2 Instance Connect 端點，以允許安全連線至您的執行個體。

考量事項

- 共用子網路 – 您可以在與您共用的子網路中建立 EC2 Instance Connect 端點。然而，您不能在與您共用的子網路中使用 VPC 擁有者建立的 EC2 Instance Connect 端點。
- IP 位址類型 – EC2 Instance Connect 端點支援以下位址類型，且必須與您的子網路相容：
  - `ipv4` – 僅透過私有 IPv4 位址連接至 EC2 執行個體。
  - `dualstack` – 透過私有 IPv4 位址或 IPv6 位址連線至 EC2 執行個體。
  - `ipv6` – 僅透過私有 IPv6 位址連接至 EC2 執行個體。

先決條件

您必須具有必要的 IAM 許可才能建立 EC2 Instance Connect 端點。如需詳細資訊，請參閱建立、描述、修改和刪除 EC2 Instance Connect 端點所需的許可。

Console

建立 EC2 Instance Connect 端點

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左側導覽窗格中選擇 Endpoints (端點)。
3. 選擇建立端點，然後指定端點設定，如下所示：

   a. (選用) 針對名稱標籤，請輸入端點的名稱。

   b. 針對類別，請選擇 EC2 Instance Connect 端點。

   c. 在網路設定下，針對 VPC，選取包含目標執行個體的 VPC。

   d. (選用) 若要保留用戶端 IP 位址，展開額外設定，然後選取保留用戶端 IP 核取方塊。否則，預設為使用端點網路介面做為用戶端 IP 位址。

   > **Note**
   >
   > 僅當端點的 IP 位址類型設定為 IPv4 時，方可使用此選項。

   e. (選用) 針對安全群組，選取要與端點產生關聯的安全群組。否則，會依預設使用 VPC 的預設安全群組。如需詳細資訊，請參閱EC2 Instance Connect 端點的安全群組。

   f. 針對子網路，請選取要在其中建立端點的子網路。

   g. 針對 IP 位址類型，選擇端點的 IP 位址類型。若需要同時支援執行個體的 IPv4 與 IPv6 連線，選擇 Dualstack。若需要支援用戶端 IP 保留，選擇 IPv4。

h. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤的鍵和值。

4. 檢閱您的設定，然後選擇 Create endpoint (建立端點)。

   端點的最初狀態是 Pending (擱置中)。在使用此端點連線至執行個體之前，您必須等待端點狀態變為可用。這可能需要幾分鐘的時間。

5. 若要使用端點連線至執行個體，請參閱 連接至執行個體。

AWS CLI

建立 EC2 Instance Connect 端點

使用 create-instance-connect-endpoint 命令。

```
aws ec2 create-instance-connect-endpoint \
    --subnet-id subnet-0123456789example
```

如需指定端點支援的流量類型，可納入 `--ip-address-type` 參數。有效值為 `ipv4`、`dualstack` 或 `ipv6`。子網路必須支援您指定的 IP 位址類型。若省略 `--ip-address-type` 參數，預設值由子網路支援的 IP 位址類型確定。

```
aws ec2 create-instance-connect-endpoint \
    --subnet-id subnet-0123456789example \
    --ip-address-type ipv4
```

以下為範例輸出。

```
{
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "create-complete",
        "StateMessage": "",
        "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "NetworkInterfaceIds": [
            "eni-0123abcd"
        ],
```

```
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1a",
        "AvailabilityZoneId": "use1-az4",
        "CreatedAt": "2023-04-07T15:43:53.000Z",
        "SubnetId": "subnet-0123abcd",
        "PreserveClientIp": false,
        "SecurityGroupIds": [
            "sg-0123abcd"
        ],
        "Tags": [],
        "IpAddressType": "ipv4"
}
```

如要監控建立狀態

State 欄位的初始值為 create-in-progress。在您可使用此端點連線至執行個體之前，請等待狀態變為 create-complete。使用 describe-instance-connect-endpoints 命令來監控 EC2 Instance Connect 端點的狀態。--query 參數可篩選 State 欄位的結果。

```
aws ec2 describe-instance-connect-endpoints --instance-connect-endpoint-
ids eice-0123456789example --query InstanceConnectEndpoints[*].State --output text
```

以下為範例輸出。

```
create-complete
```

PowerShell

如需建立 EC2 Instance Connect 端點

使用 New-EC2InstanceConnectEndpoint cmdlet。

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example
```

如需指定端點支援的流量類型，可納入 -IpAddressType 參數。有效值為 ipv4、dualstack 或 ipv6。子網路必須支援您指定的 IP 位址類型。若省略 -IpAddressType 參數，預設值由子網路支援的 IP 位址類型確定。

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example -
IpAddressType ipv4
```

以下為範例輸出。

```
OwnerId                     : 111111111111
InstanceConnectEndpointId   : eice-0123456789example
InstanceConnectEndpointArn  : arn:aws:ec2:us-east-1:111111111111:instance-connect-
endpoint/eice-0123456789example
State                       : create-complete
StateMessage                :
DnsName                     : eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com
FipsDnsName                 : eice-0123456789example.0123abcd.fips.ec2-instance-
connect-endpoint.us-east-1.amazonaws.com
NetworkInterfaceIds         : {eni-0123abcd}
VpcId                       : vpc-0123abcd
AvailabilityZone            : us-east-1a
AvailabilityZoneId          : use1-az4
CreatedAt                   : 4/7/2023 3:43:53 PM
SubnetId                    : subnet-0123abcd
PreserveClientIp            : False
SecurityGroupIds            : {sg-0123abcd}
Tags                        : {}
IpAddressType               : ipv4
```

如要監控建立狀態

State 欄位的初始值為 create-in-progress。在您可使用此端點連線至執行個體之前，請等待狀態變為 create-complete。使用 Get-EC2InstanceConnectEndpoint cmdlet 來監控 EC2 Instance Connect 端點的狀態。.State.Value 可篩選 State 欄位的結果。

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

以下為範例輸出。

```
create-complete
```

修改 EC2 Instance Connect 端點

您可以使用 主控台 AWS CLI或 SDK 修改現有的 EC2 Instance Connect Endpoints。

開始之前，您必須擁有必要的 IAM 許可。如需詳細資訊，請參閱建立、描述、修改和刪除 EC2 Instance Connect 端點所需的許可。

您可修改的參數

您可修改下面的 EC2 Instance Connect 端點參數：

安全群組

您可針對 EC2 Instance Connect 端點指定新安全群組。新安全群組可取代目前的安全群組。

若修改安全群組，必須指定：

- 至少一個安全群組，即使該安全群組只是 VPC 中的預設安全群組。
- 安全群組的 ID，而非名稱。

IP 地址類型

您可針對 EC2 Instance Connect 端點，指定一個新 IP 位址類型。

有效值：`ipv4` | `dualstack` | `ipv6`

保留用戶端 IP 設定

您可指定是否將用戶端 IP 位址保留為來源。

> **Note**
>
> 僅在 IPv4 EC2 Instance Connect 端點上支援保留用戶端 IP。若啟用 `PreserveClientIp`，端點的現有 IP 位址類型必須為 `ipv4`，或者，若在同一請求中修改 IP 位址類型，新的值必須為 `ipv4`。

Console

如需修改 EC2 Instance Connect 端點

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左側導覽窗格中選擇 Endpoints (端點)。
3. 選取端點。
4. 選擇動作、修改端點設定。
5. 變更端點設定。

6. 選擇儲存變更。

   端點進入待定狀態。修改完成時，端點會返回可用狀態。

AWS CLI

如需修改 EC2 Instance Connect 端點

使用 modify-instance-connect-endpoint 命令並指定要修改的 EC2 Instance Connect 端點與參數。在下面的範例中，修改了單一請求中的全部參數。

```
aws ec2 modify-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-0123456789example \
    --security-group-ids sg-0123456789example \
    --ip-address-type dualstack \
    --no-preserve-client-ip
```

以下為範例輸出。

```
{
    "Return": true
}
```

如需監控更新狀況

修改期間，EC2 Instance Connect 端點狀態將變更為 update-in-progress。更新程序以非同步方式執行，且以 update-complete 或 update-failed 狀態完成。狀態變更為 update-complete 之前，端點使用其舊的組態。

請使用 describe-instance-connect-endpoints 命令來監控更新狀態。--query 參數可篩選 State 欄位的結果。

```
aws ec2 describe-instance-connect-endpoints \
    --instance-connect-endpoint-ids eice-0123456789example \
    --query InstanceConnectEndpoints[*].State --output text
```

以下為範例輸出。

```
update-complete
```

PowerShell

如需修改 EC2 Instance Connect 端點

請使用 Edit-EC2InstanceConnectEndpoint cmdlet，然後指定 EC2 Instance Connect 端點及待修改參數。在下面的範例中，修改了單一請求中的全部參數。

```
Edit-EC2InstanceConnectEndpoint `
    -InstanceConnectEndpointId eice-0123456789example `
    -SecurityGroupIds sg-0123456789example `
    -IpAddressType dualstack `
    -PreserveClientIp $false
```

以下為範例輸出。

```
True
```

如需監控更新狀況

修改期間，EC2 Instance Connect 端點狀態將變更為 update-in-progress。更新程序以非同步方式執行，且以 update-complete 或 update-failed 狀態完成。狀態變更為 update-complete 之前，端點使用其舊的組態。

請使用 Get-EC2InstanceConnectEndpoint 命令來監控更新狀態。.State.Value 可篩選 State 欄位的結果。

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

以下為範例輸出。

```
update-complete
```

刪除 EC2 Instance Connect 端點

EC2 Instance Connect 端點結束使用後即可刪除。

您必須具有必要的 IAM 許可才能建立 EC2 Instance Connect 端點。如需詳細資訊，請參閱建立、描述、修改和刪除 EC2 Instance Connect 端點所需的許可。

使用主控台刪除 EC2 Instance Connect 端點時，其會進入刪除狀態。如果刪除成功，刪除的端點將不會再出現。若刪除失敗，則狀態為 delete-failed，且狀態訊息將提供失敗原因。

當您使用 刪除 EC2 Instance Connect Endpoint 時 AWS CLI，它會進入 `delete-in-progress` 狀態。如果刪除成功，則會進入 `delete-complete` 狀態。如果刪除失敗，則狀態為 `delete-failed`，而 `StateMessage` 將提供失敗原因。

Console

刪除 EC2 Instance Connect 端點

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左側導覽窗格中選擇 Endpoints (端點)。
3. 選取端點。
4. 選擇 Actions (動作)、Delete VPC endpoints (刪除 VPC 端點)。
5. 出現確認提示時，請按一下 **delete**。
6. 選擇 刪除。

AWS CLI

刪除 EC2 Instance Connect 端點

使用 delete-instance-connect-endpoint 命令並指定要刪除的 EC2 Instance Connect 端點的 ID。

```
aws ec2 delete-instance-connect-endpoint --instance-connect-endpoint-
id eice-03f5e49b83924bbc7
```

以下為範例輸出。

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
```

```
        "AvailabilityZoneId": "use1-az2",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

PowerShell

刪除 EC2 Instance Connect 端點

使用 Remove-EC2InstanceConnectEndpoint cmdlet 並指定要刪除的 EC2 Instance Connect 端點的 ID。

```
Remove-EC2InstanceConnectEndpoint -InstanceConnectEndpointId eice-03f5e49b83924bbc7
```

以下為範例輸出。

```
@{
    InstanceConnectEndpoint = @{
        OwnerId = "111111111111"
        InstanceConnectEndpointId = "eice-0123456789example"
        InstanceConnectEndpointArn = "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example"
        State = "delete-in-progress"
        StateMessage = ""
        NetworkInterfaceIds = @()
        VpcId = "vpc-0123abcd"
        AvailabilityZone = "us-east-1d"
        AvailabilityZoneId = "use1-az2"
        CreatedAt = "2023-02-07T12:05:37+00:00"
        SubnetId = "subnet-0123abcd"
    }
}
```

使用 EC2 Instance Connect 端點連線至 Amazon EC2 執行個體

您可以使用 EC2 Instance Connect 端點連線到支援 SSH 或 RDP 的 Amazon EC2 執行個體。

先決條件

- 您必須具有必要的 IAM 許可才能連線至 EC2 Instance Connect 端點。如需詳細資訊，請參閱使用 EC2 Instance Connect 端點連線到執行個體所需的許可。

- EC2 Instance Connect 端點必須為以下狀態之一：
  - 新端點的 create-complete
  - 正在修改現有端點的 update-in-progress、update-complete 或 update-failed。修改端點時，狀態變更為 update-complete 之前，端點會繼續使用其原始組態。

    若沒有 VPC 的 EC2 Instance Connect 端點，您可建立一個。如需詳細資訊，請參閱建立 EC2 Instance Connect 端點。
- EC2 Instance Connect 端點 IP 位址類型必須與執行個體的 IP 位址類型相容。若端點 IP 位址類型為雙重堆疊，則可同時適用於 IPv4 與 IPv6 位址。
- (Linux 執行個體) 若要使用 Amazon EC2 主控台連線至您的執行個體，或使用 CLI 來連接，並讓 EC2 Instance Connect 處理暫時性金鑰，則您的執行個體必須安裝 EC2 Instance Connect。如需詳細資訊，請參閱安裝 EC2 Instance Connect。
- 確保執行個體的安全群組允許來自 EC2 Instance Connect 端點的傳入 SSH 流量。如需詳細資訊，請參閱目標執行個體安全群組規則。

連線選項

- 使用 Amazon EC2 主控台連線至 Linux 執行個體
- 使用 SSH 連線至您的 Linux 執行個體
- 使用 以執行個體 ID 連線到您的 Linux 執行個體 AWS CLI
- 使用 RDP 連線至您的 Windows 執行個體
- 疑難排解

使用 Amazon EC2 主控台連線至 Linux 執行個體

您可以使用 Amazon EC2 主控台 (瀏覽器型用戶端) 連線到執行個體，如下所示。

要使用 Amazon EC2 主控台連線至執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 Connect (連線)。
4. 選擇 EC2 Instance Connect 索引標籤。
5. 針對連線類型，請選擇使用私有 IP 建立連線。

6. 選擇私有 IPv4 位址還是 IPv6 位址。依據指派給您的執行個體的 IP 位址提供選項。若選項灰顯，則表示執行個體沒有指派給該類型的 IP 位址。
7. 針對 EC2 Instance Connect 端點，請選擇 EC2 Instance Connect 端點的 ID。

> **Note**
>
> EC2 Instance Connect 端點必須與您在上一步驟中選擇的 IP 位址相容。若端點 IP 位址類型為雙重堆疊，則可同時適用於 IPv4 與 IPv6 位址。如需詳細資訊，請參閱建立 EC2 Instance Connect 端點。

8. 針對使用者名稱，如果您用來啟動執行個體的 AMI 使用 `ec2-user` 以外的使用者名稱，請輸入正確的使用者名稱。
9. 針對通道持續時間上限 (秒)，輸入允許的 SSH 連線持續時間上限。

   持續時間必須符合 IAM 政策中指定的任何 `maxTunnelDuration` 條件。如果您無權存取 IAM 政策，請聯絡您的管理員。
10. 選擇連線。這會開啟執行個體的終端視窗。

使用 SSH 連線至您的 Linux 執行個體

您可以使用 SSH 連線到 Linux 執行個體，並使用 `open-tunnel` 命令建立私有通道。可以在單一連線或多重連線模式下使用 `open-tunnel`。您可指定執行個體 ID、私有 IPv4 位址，或者 IPv6 位址。

如需使用 AWS CLI 以使用 SSH 連線至執行個體的詳細資訊，請參閱 使用 連線 AWS CLI。

以下範例使用 OpenSSH。您可以使用支援代理模式的任何其他 SSH 用戶端。

單一連接

使用 SSH 和 **`open-tunnel`** 命令僅允許單一連線至執行個體

使用 `ssh`和 open-tunnel AWS CLI 命令，如下所示。`-o` 代理命令包含 `open-tunnel` 命令，它可建立執行個體的私有通道。

```
ssh -i my-key-pair.pem ec2-user@i-1234567890abcdef0 \
    -o ProxyCommand='aws ec2-instance-connect open-tunnel --instance-
id i-1234567890abcdef0'
```

對於：

- `-i` – 指定用來啟動執行個體的金鑰對。
- *`ec2-user`*`@`*`i-1234567890abcdef0`* – 指定用來啟動執行個體的 AMI 使用者名稱，以及執行個體 ID。若是使用 IPv6 位址的執行個體，必須指定 IPv6 位址，而非執行個體 ID。
- `--instance-id` – 指定要連線之執行個體的 ID。或者，指定 `%h`，其可從使用者中擷取執行個體 ID。若是使用 IPv6 位址的執行個體，使用 `--instance-id` *`i-1234567890abcdef0`* 來取代 `--private-ip-address` *`2001:db8::1234:5678:1.2.3.4`*。

多重連線

若要允許執行個體的多個連線，請先執行 open-tunnel AWS CLI 命令以開始接聽新的 TCP 連線，然後使用 `ssh` 建立新的 TCP 連線和執行個體的私有通道。

允許使用 SSH 和 **`open-tunnel`** 命令多重連線至執行個體

1. 執行下列命令來開始偵聽本機電腦上指定連接埠上的新 TCP 連線。

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-1234567890abcdef0 \
    --local-port 8888
```

預期的輸出結果：

```
Listening for connections on port 8888.
```

2. 在新終端視窗中，執行下列 `ssh` 命令，為執行個體建立新的 TCP 連線和私有通道。

```
ssh -i my-key-pair.pem ec2-user@localhost -p 8888
```

預期輸出 – 在第一個終端視窗中，您會看到以下內容：

```
[1] Accepted new tcp connection, opening websocket tunnel.
```

您可能會看到以下內容：

```
[1] Closing tcp connection.
```

使用 以執行個體 ID 連線到您的 Linux 執行個體 AWS CLI

如果您只知道執行個體 ID，您可以使用 ec2-instance-connect ssh AWS CLI 命令，使用 SSH 用戶端連線到執行個體。如需詳細資訊，請參閱使用 連線 AWS CLI。

先決條件

- 安裝第 2 AWS CLI 版，並使用您的登入資料進行設定。如需詳細資訊，請參閱「AWS Command Line Interface 使用者指南」中的安裝或更新 AWS CLI的最新版本和設定 AWS CLI。
- 或者，在預先驗證的 shell 中開啟 AWS CloudShell 和執行 AWS CLI 命令。

使用執行個體 ID 和 EC2 Instance Connect 端點連線至執行個體

如果您僅知道執行個體 ID，請使用 ec2-instance-connect ssh CLI 命令，並使用 `eice` 值來指定 `ssh` 命令、執行個體 ID 和 `--connection-type` 參數，以使用 EC2 Instance Connect 端點。若執行個體僅擁有 IPv6 位址，還必須連同 IPv6 位址一起納入 `--instance-ip` 參數。

- 若執行個體擁有私有 IPv4 位址 (也可有 IPv6 位址)，請使用下面的命令與參數：

```
aws ec2-instance-connect ssh \
    --instance-id i-1234567890example \
    --os-user ec2-user \
    --connection-type eice
```

- 若執行個體僅擁有 IPv6 位址，請在 IPv6 位址中納入 `--instance-ip` 參數：

```
aws ec2-instance-connect ssh \
    --instance-id i-1234567890example \
    --instance-ip 2001:db8::1234:5678:1.2.3.4 \
    --os-user ec2-user \
    --connection-type eice
```

> **Tip**
>
> 如果您收到錯誤，請確定您使用的是第 2 AWS CLI 版。`ssh` 參數僅適用於第 2 AWS CLI 版。如需相關資訊，請參閱「AWS Command Line Interface 使用者指南」中的關於 AWS CLI 第 2 版。

使用 RDP 連線至您的 Windows 執行個體

您可以透過 EC2 Instance Connect 端點，使用遠端桌面通訊協定 (RDP)，在沒有公有 IPv4 地址或公有 DNS 名稱的情況下連線到 Windows 執行個體。

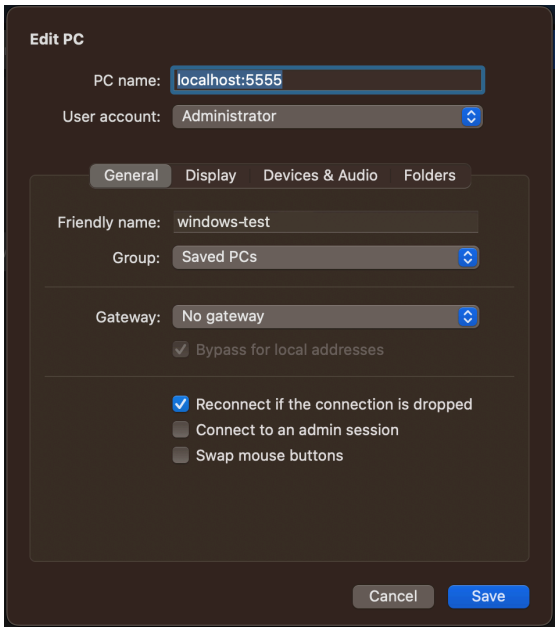
使用 RDP 用戶端連線至您的 Windows 執行個體。

1. 完成使用 RDP 連線至 Windows 執行個體中的步驟 1 – 8。在步驟 8 下載 RDP 桌面檔案之後，您會收到無法連線訊息，這是預期訊息，因為您的執行個體沒有公有 IP 位址。
2. 執行下列命令，為執行個體所在的 VPC 建立私有通道。`--remote-port` 必須為 3389，因為 RDP 預設會使用連接埠 3389。

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-1234567890abcdef0 \
    --remote-port 3389 \
    --local-port any-port
```

3. 在下載資料夾中，尋找您下載的 RDP 桌面檔案，然後將其拖曳至 RDP 用戶端視窗。
4. 開啟 RDP 桌面檔案的內容 （按一下滑鼠右鍵） 選單，然後選擇編輯。
5. 在編輯電腦視窗中，針對電腦名稱 (要連線的執行個體)，輸入 `localhost:local-port`，其中 *local-port* 使用您在步驟 2 中所指定的值，然後選擇儲存。

   請注意，編輯電腦視窗的以下螢幕截取畫面來自 Mac 中的 Microsoft 遠端桌面。如果使用 Windows 用戶端，則該視窗可能會有所不同。



6. 在 RDP 用戶端中，開啟 PC 的內容（按一下滑鼠右鍵）選單（您剛設定的內容），然後選擇連線以連線至執行個體。
7. 出現提示時，輸入管理員帳戶的解密密碼。

疑難排解

請使用以下資訊來協助您診斷並修正使用 EC2 Instance Connect 端點連線執行個體時可能遇到的問題。

無法連線至執行個體

以下是您可能無法連線至執行個體的常見原因。

- 安全群組：檢查指派給 EC2 Instance Connect 端點和執行個體的安全群組。有關所需安全群組規則的詳細資訊，請參閱 EC2 Instance Connect 端點的安全群組。
- 執行個體狀態：驗證您的執行個體處於 `running` 狀態。
- 金鑰對：如果用來連線的命令需要私有金鑰，請確認執行個體具有公有金鑰，且您擁有對應的私有金鑰。
- IAM 許可：確認您具有所需的 IAM 許可。如需詳細資訊，請參閱授予許可以使用 EC2 Instance Connect 端點。

如需 Linux 執行個體的更多疑難排解秘訣，請參閱 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解。如需 Windows 執行個體的疑難排解秘訣，請參閱 the section called “Windows 執行個體 RDP 問題”。

錯誤代碼：AccessDeniedException

如果您收到 `AccessDeniedException` 錯誤，且 IAM 政策中已指定 `maxTunnelDuration` 條件，則請務必在連線至執行個體時指定 `--max-tunnel-duration` 參數。如需有關此參數的詳細資訊，請參閱「AWS CLI 命令參考」中的 open-tunnel。

記錄透過 EC2 Instance Connect 端點建立的連線

您可以使用 日誌記錄透過 EC2 Instance Connect Endpoint 建立的資源操作和稽核連線 AWS CloudTrail 。

如需 AWS CloudTrail 搭配 Amazon EC2 使用 的詳細資訊，請參閱 使用 記錄 Amazon EC2 API 呼叫 AWS CloudTrail。

使用 記錄 EC2 Instance Connect Endpoint API 呼叫 AWS CloudTrail

EC2 Instance Connect 端點資源操作會以管理事件的形式記錄到 CloudTrail。進行下列 API 呼叫時，活動會作為 CloudTrail 事件記錄在事件歷史記錄中：

- `CreateInstanceConnectEndpoint`
- `DescribeInstanceConnectEndpoints`
- `DeleteInstanceConnectEndpoint`

您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的 使用 CloudTrail 事件歷史記錄檢視事件。

使用 AWS CloudTrail 來稽核使用 EC2 Instance Connect Endpoint 連線至執行個體的使用者

透過 EC2 Instance Connect Endpoint 對執行個體的連線嘗試會記錄在事件歷史記錄中的 CloudTrail 中。當透過 EC2 Instance Connect 端點啟動執行個體的連線時，連線會記錄為 CloudTrail 管理事件，具有 `OpenTunnel` 的 `eventName`。

您可以建立將 CloudTrail 事件路由到目標的 Amazon EventBridge 規則。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

以下是 CloudTrail 中記錄的 `OpenTunnel` 管理事件範例。

```
{
     "eventVersion": "1.08",
     "userIdentity": {
         "type": "IAMUser",
         "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
         "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
         "accountId": "123456789012",
         "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
         "userName": "IAM-friendly-name"
     },
     "eventTime": "2023-04-11T23:50:40Z",
     "eventSource": "ec2-instance-connect.amazonaws.com",
     "eventName": "OpenTunnel",
     "awsRegion": "us-east-1",
     "sourceIPAddress": "1.2.3.4",
     "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
     "requestParameters": {
         "instanceConnectEndpointId": "eici-0123456789EXAMPLE",
```

```
        "maxTunnelDuration": "3600",
        "remotePort": "22",
        "privateIpAddress": "10.0.1.1"
    },
    "responseElements": null,
    "requestID": "98deb2c6-3b3a-437c-a680-03c4207b6650",
    "eventID": "bbba272c-8777-43ad-91f6-c4ab1c7f96fd",
    "readOnly": false,
    "resources": [{
        "accountId": "123456789012",
        "type": "AWS::EC2::InstanceConnectEndpoint",
        "ARN": "arn:aws:ec2:us-east-1:123456789012:instance-connect-endpoint/
eici-0123456789EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


EC2 Instance Connect 端點的服務連結角色

Amazon EC2 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 Amazon EC2 的一種特殊 IAM 角色類型。服務連結角色是由 Amazon EC2 預先定義，並包含 Amazon EC2 AWS 服務 代表您呼叫其他 所需的所有許可。如需詳細資訊，請參閱「IAM 使用者指南」中的服務連結角色。

EC2 Instance Connect 端點的服務連結角色許可

Amazon EC2 使用 AWSServiceRoleForEC2InstanceConnect 來在您的帳戶中建立和管理 EC2 Instance Connect 端點所需的網路介面。

AWSServiceRoleForEC2InstanceConnect 服務連結角色信任下列服務擔任該角色：

- `ec2-instance-connect.amazonaws.com`

AWSServiceRoleForEC2InstanceConnect 服務連結角色使用下列受管政策：

- Ec2InstanceConnectEndpoint

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 Ec2InstanceConnectEndpoint。

建立 EC2 Instance Connect 端點的服務連結角色

您不需要手動建立這個服務連結角色。當您建立EC2 Instance Connect 端點時，Amazon EC2 會為您建立服務連結角色。

編輯 EC2 Instance Connect 端點的服務連結角色

EC2 Instance Connect 端點不允許您編輯 AWSServiceRoleForEC2InstanceConnect 服務連結角色。

刪除 EC2 Instance Connect 端點的服務連結角色

如果不再需要使用 EC2 Instance Connect 端點，建議您刪除 AWSServiceRoleForEC2InstanceConnect 服務連結角色。

您必須刪除所有 EC2 Instance Connect 端點資源，才可以刪除服務連結角色。

若要刪除服務連結角色，請參閱《IAM 使用者指南》中的刪除服務連結角色。

您必須設定許可，IAM 實體 (使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

EC2 Instance Connect 端點配額

您的 AWS 帳戶 具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定的。

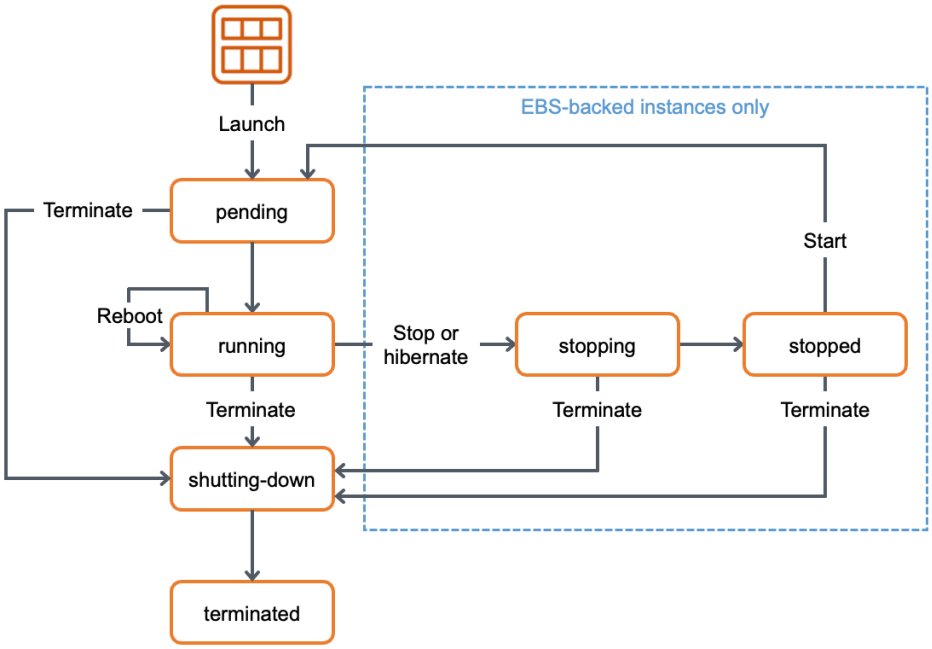
您的 AWS 帳戶 具有下列與 EC2 Instance Connect Endpoint 相關的配額。

名稱	預設	可調整
AWS 帳戶 每個 的 EC2 Instance Connect 端點數量上限 AWS 區域	5	否
每個 VPC 的 EC2 Instance Connect 端點數目上限	1	否
每個子網路的 EC2 Instance Connect 端點數目上限	1	否
每個 EC2 Instance Connect 端點的並行連線數量上限	20	否

Amazon EC2 執行個體狀態變更

從啟動到終止的那一刻起，Amazon EC2 執行個體會轉換到不同的狀態。

下圖顯示執行個體狀態之間的轉換。



您可以在執行個體變更狀態時收到通知。如需詳細資訊，請參閱the section called “狀態變更事件”。

依執行個體狀態計費

下表提供每個執行個體狀態的簡要說明，並指出執行個體用量的付款狀態。無論執行個體的狀態為何，Amazon EBS 磁碟區和彈性 IP 地址等部分 AWS 資源都會產生費用。如需詳細資訊，請參閱 AWS Billing 使用者指南中的避免非預期的費用。

執行個體狀態	描述	執行個體用量計費
pending	執行個體準備進入 running 狀態。當執行個體啟動時，或者在處於 stopped 狀態之後啟動時，將進入 pending 狀態。	不計費

執行個體狀態	描述	執行個體用量計費
running	執行個體正在執行中且可供使用。	計費
stopping	執行個體已準備停止。	不計費 **Note** 如果讓執行個體休眠，則會在執行個體處於 stopping 狀態時向您計費。
stopped	執行個體已關閉且無法使用。執行個體可以隨時啟動。	不計費
shutting-down	執行個體準備終止。	不計費
terminated	執行個體已永久刪除且無法啟動。	不計費 **Note** 適用於已終止執行個體的預留執行個體，根據其付款選項計費至期滿。如需詳細資訊，請參閱Amazon EC2 的預留執行個體概觀

待定執行個體

當您啟動執行個體時，其將進入 pending 狀態。您在啟動時指定的執行個體類型決定託管執行個體的電腦硬體。您在啟動時指定的 Amazon Machine Image (AMI) 會用來啟動執行個體。執行個體便已準備就緒，可供使用時，其將變為 running 狀態。接著您可連接到執行中的執行個體，並以一般使用電腦的方式使用。

您的執行個體轉換為 running 狀態後，會針對該執行個體執行的每秒計費 (最少為一分鐘)，即使執行個體保持閒置而且您未連接它仍會計費。

停止的執行個體

如果您的執行個體未能通過狀態檢查或未如預期執行您的應用程式，而且執行個體的根磁碟區為 Amazon EBS 磁碟區，您可以停止和啟動執行個體，以嘗試修正此問題。

停止執行個體時，其會變成 stopping 狀態，然後再變成 stopped 狀態。執行個體若為 stopped，您不需支付使用量或資料傳輸費用。任何 Amazon EBS 磁碟區的儲存都會產生費用。當您的執行個體變為 stopped 狀態時，您可修改執行個體的部分屬性，包括執行個體類型。

當您啟動執行個體時，它會進入 pending 狀態，執行個體會移至新的主機電腦 (但有時會保留在目前的主機上)。停止並啟動執行個體時，連接原本主機電腦的執行個體儲存體磁碟區內的所有資料可能都會遺失。

您的執行個體會保留其私有 IPv4 地址，這表示與該私有 IPv4 地址或網路介面相關聯的彈性 IP 地址會與您的執行個體保留關聯。如果您的執行個體有 IPv6 地址，則會保留 IPv6 地址。

每次將執行個體從 stopped 轉換為 running 時，只要執行個體在執行就會按秒收費，每次啟動執行個體至少會收費一分鐘。

如需停止和啟動執行個體的詳細資訊，請參閱：停止和啟動 Amazon EC2 執行個體。

休眠的執行個體

當您讓執行個體休眠時，系統會發出訊號以執行休眠 (suspend-to-disk)，將執行個體記憶體 (RAM) 中的內容儲存至您的 Amazon EBS 根磁碟區。執行個體的 Amazon EBS 根磁碟區和任何連接的 Amazon EBS 資料磁碟區都會保留。當您啟動執行個體時，Amazon EBS 根磁碟區會還原至其先前的狀態，並重新載入 RAM 的內容。先前連接的資料磁碟區會重新連接，且執行個體會保留其執行個體 ID。

執行個體休眠時會變成 stopping 狀態，然後再變成 stopped 狀態。當執行個體處於 stopped 狀態時，您不需要支付休眠執行個體的使用費，但在處於 stopping 狀態時會向您收取費用，與停止執行個體而不休眠時不同。您不需要支付資料傳輸費用，但您需要支付任何 Amazon EBS 磁碟區的儲存費用，包括 RAM 資料的儲存費用。

當您啟動休眠的執行個體時，它會進入 pending 狀態，而執行個體會移至新的主機電腦 （不過在某些情況下，會保留在目前的主機上）。

您的執行個體會保留其私有 IPv4 地址，這表示與該私有 IPv4 地址或網路介面相關聯的彈性 IP 地址仍會與您的執行個體相關聯。如果您的執行個體擁有 IPv6 地址，其將保留 IPv6 地址。

如需詳細資訊，請參閱讓 Amazon EC2 執行個體進入休眠。

重新啟動執行個體

您可以使用 Amazon EC2 主控台、命令列工具和 Amazon EC2 API 來重新啟動執行個體。建議您使用 Amazon EC2 重新啟動執行個體，而不是從執行個體執行作業系統重新啟動命令。

重新啟動執行個體等同於重新啟動作業系統。執行個體會留在相同的主機電腦上，並保有其公有 DNS 名稱、私有 IP 地址和執行個體存放區磁碟區上的所有資料。重新啟動通常需要幾分鐘時間才能完成，但重新啟動實際需要時間仍取決於執行個體組態。

重新啟動執行個體不會啟動新的執行個體計費期間；會繼續以秒計費，但不會收取其他的最少一分鐘收費。

如需詳細資訊，請參閱重新啟動您的 Amazon EC2 執行個體。

終止的執行個體

當您決定不再需要執行個體時，您可終止執行個體。執行個體的狀態變更為 `shutting-down` 或 `terminated` 時刻起，該執行個體便停止收取費用。

如果啟用終止保護，您將無法以主控台、CLI 或 API 終止執行個體。

終止執行個體後，仍會短暫顯示於主控台，之後該項目會自動刪除。您也可以用 CLI 和 API 來描述終止的執行個體。資源 (例如標籤) 會逐漸與該終止的執行個體取消關聯，過一段時間後，將不再顯示於該已終止的執行個體。您無法連接或復原已終止的執行個體。

每個 Amazon EBS 後端執行個體均支援 `InstanceInitiatedShutdownBehavior` 屬性，其控制當您在執行個體內部啟動關機時執行個體將停止或終止 (例如，在 Linux 上使用 shutdown 命令時)。預設行為是停止執行個體。您可在執行個體運作中或停止時，修改此屬性設定。

每個 Amazon EBS 磁碟區均支援 `DeleteOnTermination` 屬性，其控制當您終止磁碟區所連接的執行個體時要刪除或保留磁碟區。預設值為刪除根磁碟區，並保留任何其他的 EBS 磁碟區。

如需詳細資訊，請參閱終止 Amazon EC2 執行個體。

執行個體狀態之間的差異

下表摘要列出正在重新啟動、正在停止、正在休眠和正在終止執行個體之間的主要差異。

特性	重新開機	停止/啟動 (僅適用於 Amazon EBS 後端執行個體)	休眠 (僅限 Amazon EBS 後端的執行個體)	終止
主機電腦	執行個體保持在同一個主機電腦上。	執行個體會移至新的主機電腦（但在某些情況下，仍會保留在目前的主機上）。	執行個體會移至新的主機電腦（但在某些情況下，仍會保留在目前的主機上）。	無
私有 IPv4 地址	執行個體將保留其私有 IPv4 地址。	執行個體將保留其私有 IPv4 地址。	執行個體將保留其私有 IPv4 地址。	無
公有 IPv4 地址	執行個體將保留其公有 IPv4 位址。	執行個體會取得新的公有 IPv4 位址，除非其具有次要網路介面或與彈性 IP 位址相關聯的次要私有 IPv4 位址。	執行個體會取得新的公有 IPv4 位址，除非其具有次要網路介面或與彈性 IP 位址相關聯的次要私有 IPv4 位址。	無
彈性 IP 地址 (IPv4)	彈性 IP 地址仍與執行個體關聯	彈性 IP 地址仍與執行個體關聯	彈性 IP 地址仍與執行個體關聯	彈性 IP 地址將從執行個體解除關聯
IPv6 地址	執行個體將保留其私有 IPv6 地址	執行個體將保留其私有 IPv6 地址	執行個體將保留其私有 IPv6 地址	無
執行個體存放區磁碟區	資料將保留	資料將清除	資料將清除	資料將清除
根磁碟區	磁碟區將保留	磁碟區將保留	磁碟區將保留	磁碟區預設為刪除
RAM (記憶體內容)	將清除 RAM	將清除 RAM	RAM 會儲存到根磁碟區的檔案	將清除 RAM

特性	重新開機	停止/啟動 (僅適用於 Amazon EBS 後端執行個體)	休眠 (僅限 Amazon EBS 後端的執行個體)	終止
帳單	執行個體計費小時不變	執行個體的狀態變更為 stopping 時刻起，該執行個體便停止收取費用。每次執行個體從 stopped 轉移為 running 時，我們都會啟動新的執行個體計費期間，每次啟動執行個體時最少收費一分鐘。	執行個體的狀態為 stopping 時會產生費用，但執行個體的狀態為 stopped 時即停止產生費用。每次執行個體從 stopped 轉移為 running 時，我們都會啟動新的執行個體計費期間，每次啟動執行個體時最少收費一分鐘。	執行個體的狀態變更為 shutting-down 的時刻起，該執行個體便停止收取費用

作業系統關機命令每次都會終止具有執行個體儲存體根磁碟區的執行個體。您可控制作業系統關機命令要停止或終止具有 EBS 根磁碟區的執行個體。如需詳細資訊，請參閱變更執行個體啟動的關機行為。

停止和啟動 Amazon EC2 執行個體

如果執行個體使用 Amazon EBS 磁碟區做為其根磁碟區，則您可以停止和啟動此執行個體。當您停止執行個體時，執行個體將會關閉。當您啟動執行個體時，其通常會移轉至新的基礎主機電腦，並指派新的公有 IPv4 地址。

執行個體停止可由使用者啟動（您手動停止執行個體）或由 啟動 AWS（當 AWS 偵測到執行個體的基礎主機發生無法修復的故障時，回應排程的停止事件）。

對於使用者發起的停止，建議您使用 Amazon EC2 主控台、CLI 或 API，而非從執行個體中執行作業系統停止命令。使用 Amazon EC2 時，若執行個體未在幾分鐘內正常關機，Amazon EC2 會執行強制關機。此外， 會 AWS CloudTrail 建立執行個體停止時的 API 記錄。

本主題說明如何執行使用者發起的停止。如需 執行的停止資訊 AWS，請參閱 管理已排程為停止或淘汰的 Amazon EC2 執行個體。

當您停止執行個體時，該執行個體不會刪除。如果您決定不再需要使用執行個體時，可將其終止。如需詳細資訊，請參閱終止 Amazon EC2 執行個體。如果要讓執行個體休眠以儲存執行個體記憶體 (RAM)

中的內容，請參閱：讓 Amazon EC2 執行個體進入休眠。有關執行個體生命週期操作之間的區別，請參閱 :執行個體狀態之間的差異。

目錄

- EC2 執行個體的停止和啟動方式
- 停止執行個體的方法
- 手動停止和啟動您的執行個體
- 自動停止及啟動執行個體
- 尋找所有執行中和停止的執行個體
- 尋找初始和最新的啟動時間
- 為您的 EC2 執行個體啟用停止保護

EC2 執行個體的停止和啟動方式

當您停止 Amazon EC2 執行個體時，變更會在執行個體的作業系統 (OS) 層級註冊，而有些資源會遺失，有些則會保留。啟動執行個體時，會在執行個體層級註冊變更。

主題

- 若您停止執行個體會如何
- 啟動執行個體時會出現的情況
- 測試應用程式對停止和啟動的回應
- 停止和啟動執行個體的相關成本

若您停止執行個體會如何

下列說明使用預設停止方法停止執行個體時通常會發生的情況。請注意，部分細節可能會根據您使用的停止方法而有所不同。

在作業系統層級註冊的變更

- API 請求會將按鈕事件傳送給訪客。
- 由於按鈕事件而停止各種系統服務。作業系統正常關機由從 Hypervisor 按下 ACPI 關機按鈕事件觸發。
- 啟動 ACPI 關機。
- 當正常關機程序結束時，執行個體就會作業系統關機。OS 關機時間無法設定。

- 如果執行個體作業系統在幾分鐘內沒有完全關閉，便會執行硬關機。
- 執行個體會停止執行。
- 執行個體的狀態會變成 `stopping` 然後再變成 `stopped`。
- [自動擴展] 如果您的執行是 Auto Scaling 群組，當執行個體的狀態處於 `running` 以外的任何 Amazon EC2 狀態，或是狀態檢查的狀態變為 `impaired`，則 Amazon EC2 Auto Scaling 會認定此執行個體運作狀態不佳，並將其取代。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的針對 Auto Scaling 群組中的執行個體進行運作狀態檢查。
- [Windows 執行個體] 當您停止和啟動 Windows 執行個體時，啟動代理程式會對執行個體執行任務，例如針對任何已連結的 Amazon EBS 磁碟區變更其磁碟代號。如需這些預設值以及如何變更這些預設值的詳細資訊，請參閱 the section called “EC2Launch v2”。

資源遺失

- 存放在 RAM 的資料。
- 存放在執行個體儲存體磁碟區的資料。
- Amazon EC2 在啟動或開始時自動指派給執行個體的公有 IPv4 地址。若要保留永遠不變的公有 IPv4 地址，可將彈性 IP 地址與您的執行個體建立關聯。

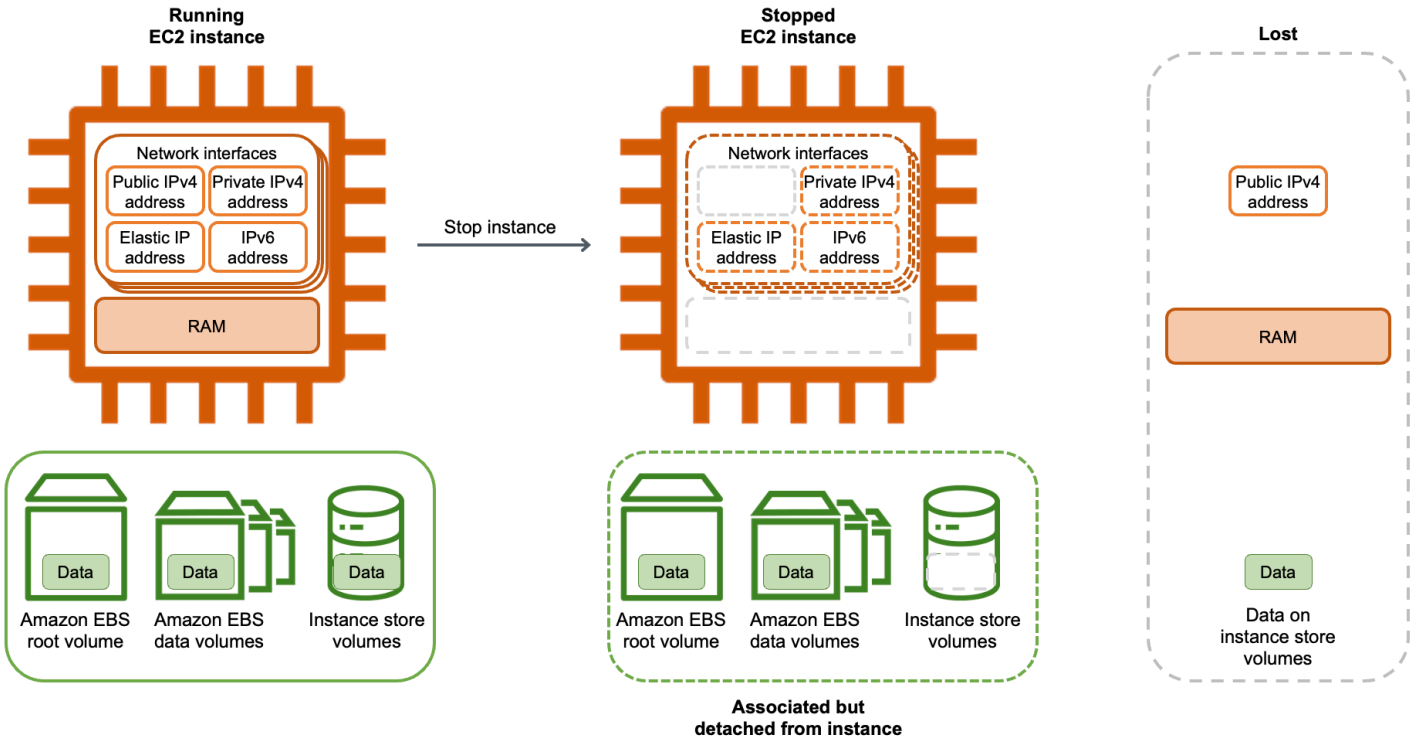
持續存在的資源

- 所有連接的 Amazon EBS 根和資料磁碟區。
- Amazon EBS 磁碟區中存放的資料。
- 任何已連接的網路介面。

  網路介面包含下列資源，這些資源也會持續存在：

  - 私有 IPv4 地址。
  - IPv6 地址。
  - 與執行個體有關聯的彈性 IP 地址。請注意，當執行個體停用時，系統會針對相關聯的彈性 IP 地址向您收費。

下圖顯示 EC2 執行個體停止時持續存在及遺失的內容。此圖表分為三部分：第一個部分標示為執行中的 EC2 執行個體，顯示處於 `running` 狀態的執行個體及其資源。第二個部分標示為已停止的 EC2 執行個體，顯示處於 `stopped` 狀態的執行個體及其持續存在的資源。第三部分標示為已遺失，顯示執行個體停止時遺失的資源。



如需停止 Mac 執行個體時所發生狀況的詳細資訊，請參閱 停止或終止 Amazon EC2 Mac 執行個體。

啟動執行個體時會出現的情況

- 大多數情況下，執行個體會遷移至新的基礎主機電腦 (不過有時候會保留在目前的主機上，例如當執行個體是配置到專用執行個體組態中的主機時)。
- 關聯的 EBS 磁碟區和網路介面會重新連接至執行個體。
- 若執行個體已設定為接收公有 IPv4 位址，Amazon EC2 會為其指派新的公有 IPv4 位址，除非其具有次要網路介面或與彈性 IP 位址相關聯的次要私有 IPv4 位址。
- 如果先停止置放群組中的執行個體，然後再啟動，此執行個體仍會在置放群組中執行。不過，如果執行個體可用的容量不足，起始作業將會失敗。如果您在已執行執行個體的置放群組中啟動執行個體時收到容量錯誤，請停止置放群組中的所有執行個體，然後再次啟動所有執行個體。啟動執行個體的動作，可能會讓這些執行個體轉移到容量足夠的硬體，能夠提供容量給所有請求的執行個體使用。

測試應用程式對停止和啟動的回應

您可以使用 AWS Fault Injection Service 來測試您的應用程式在執行個體停止和啟動時如何回應。如需詳細資訊，請參閱「AWS Fault Injection Service 使用者指南」。

停止和啟動執行個體的相關成本

停止和啟動執行個體的相關成本如下。

停止 - 一旦執行個體狀態變更為 shutting-down 或 terminated，執行個體就不會再產生費用。您不需為已停止的執行個體支付使用量或資料傳輸費用。存放 Amazon EBS 儲存磁碟區會產生費用。

啟動 - 每次啟動已停止的執行個體時，我們會收取至少一分鐘的使用費。如果超過一分鐘，只會依使用的秒數收費。例如，如果您執行執行個體 20 秒便停止，我們會收取完整一分鐘的費用。如果您執行執行個體 3 分 40 秒，則只會收取 3 分 40 秒的使用費。

停止執行個體的方法

使用者發起的停止操作有四種方式：預設停止、跳過作業系統關機的停止、強制停止，以及跳過作業系統關機的強制停止。下表會比較停止方式之間的關鍵差異。

停止方式	主要用途	使用案例	CLI 命令
預設停止	嘗試作業系統正常關機的一般執行個體關機。	一般的執行個體停止。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0`
跳過作業系統關機的停止	停止執行個體時繞過作業系統正常關機。	需要繞過作業系統正常關機時。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`
強制停止	處理停滯的執行個體。先嘗試預設停止；若執行個體無法停止，則強制停止執行個體。	執行個體卡在 stopping 狀態時。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \`

停止方式	主要用途	使用案例	CLI 命令
			`--force`
強制跳過作業系統關機的停止	停止執行個體時強制停止並繞過作業系統正常關機。	需要強制停止並繞過作業系統正常關機時。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

各方法的使用指示，請參閱下列內容：

- 在作業系統正常關機的情況下停止執行個體
- 停止執行個體並繞過作業系統正常關機
- 強制停止執行個體

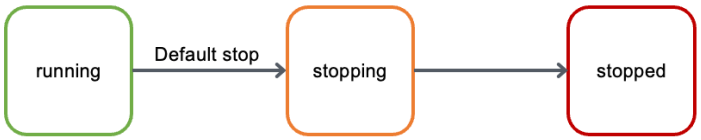
停止方式：

- 預設停止
- 跳過作業系統關機的停止
- 強制停止
- 強制跳過作業系統關機的停止

下列章節提供有關四種不同使用者發起停止方法的詳細資訊。

預設停止

預設停止方法是停止執行個體的標準方式。在您發送 StopInstances 命令時，執行個體會從 `running` 狀態轉換為 `stopping`，最後轉換為 `stopped`，如下圖所示：



用途：嘗試作業系統正常關機的一般執行個體關機。

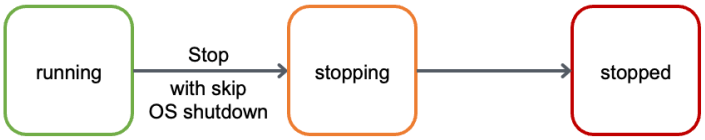
資料影響：保留 EBS 根磁碟區與資料磁碟區上的資料。執行個體儲存體磁碟區上的資料遺失。

使用時機：一般停止操作的首次嘗試。

> **Note**
>
> 若您已嘗試跳過作業系統關機的停止操作，則在同一狀態轉換工作階段期間的後續預設停止嘗試，將不會執行作業系統正常關機。繞過作業系統正常關機在執行個體的目前工作階段無法復原。

跳過作業系統關機的停止

若需要繞過作業系統正常關機，可使用跳過作業系統關機的停止方法來停止執行個體並繞過作業系統正常關機，如下圖所示：



> **Warning**
>
> 繞過作業系統正常關機可能導致資料遺失或損毀 (例如，記憶體內容未寫入磁碟、正在處理的 IO 遺失) 或跳過關機指令碼。

用途：停止執行個體時繞過作業系統正常關機。

資料影響：可能會導致資料遺失或損毀。記憶體內容可能未寫入磁碟，且正在處理的 IO 可能遺失。可能會跳過關機指令碼。

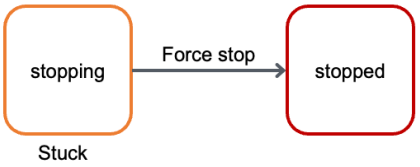
使用時機：需要跳過作業系統正常關機時。若在執行含作業系統正常關機的預設停止操作期間使用，將跳過作業系統正常關機。

> **Note**
>
> 對於執行個體目前的狀態轉換工作階段，繞過作業系統正常關機是不可復原的。在此工作階段期間，後續的預設停止嘗試不會嘗試作業系統正常關機。

強制停止

強制停止方法用於處理卡在 stopping 狀態的執行個體。執行個體通常會因為基礎硬體問題而卡住 (由失敗的系統狀態檢查表示)。

強制停止方法會先嘗試預設停止。若執行個體仍然卡在 stopping 狀態，force 參數會強制關閉執行個體，並將執行個體轉換為 stopped 狀態，如下圖所示：



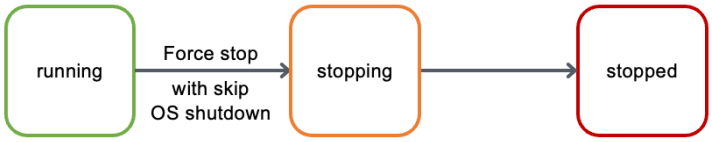
用途：處理卡在 stopping 狀態的執行個體。先嘗試預設停止。若執行個體無法停止，則強制關閉執行個體。

資料影響：先嘗試預設停止，但若強制停止繼續，則可能會導致資料遺失或損毀。在極少數情況下，可能導致停止後向 EBS 磁碟區或其他共用資源寫入資料。

使用時機：預設停止後執行個體仍卡住時的第二次停止嘗試。如需詳細資訊，請參閱對 Amazon EC2 執行個體停止問題進行疑難排解。

強制跳過作業系統關機的停止

若需要強制停止和繞過作業系統正常關機，可使用跳過作業系統關機方法的強制停止將執行個體轉換至 stopped 狀態，如下圖所示：



用途：結合強制停止與在停止執行個體時繞過作業系統正常關機。

資料影響：跳過作業系統關機可能會導致資料遺失或損毀。記憶體內容可能未寫入磁碟，且正在處理的 IO 可能遺失。可能會跳過關機指令碼。若強制停止繼續，則可能會導致額外的資料遺失或損毀。在極少數情況下，可能導致停止後向 EBS 磁碟區或其他共用資源寫入資料。

使用時機：當您想要確定執行個體會停止，並且想要繞過作業系統正常關機時。若在執行含作業系統正常關機的預設停止操作期間使用，將跳過作業系統正常關機。

手動停止和啟動您的執行個體

您可以停止和啟動 Amazon EBS 支援的執行個體 (使用 EBS 根磁碟區的執行個體)。您無法使用執行個體儲存體根磁碟區停止和啟動執行個體。

使用預設方法停止執行個體時，會嘗試執行作業系統 (OS) 正常關機。您可以繞過作業系統正常關機；但這可能會危及資料完整性。

> **Warning**
>
> 停止執行個體時，在任何執行個體儲存體磁碟區的資料都會清除。停止執行個體之前，請確認您已將所需的任何資料從執行個體儲存體磁碟區複製到永久性儲存，例如 Amazon EBS 或 Amazon S3。

[Linux 執行個體] 從執行個體使用 halt 命令不會啟動關機作業。如果您使用 halt 命令，執行個體不會終止，而是會將 CPU 放入 HLT，從而暫停 CPU 作業。執行個體會維持執行中狀態。

您可以使用作業系統 shutdown 或 poweroff 命令啟動關機作業。使用 OS 命令時，執行個體預設會停止。您可變更此行為。如需詳細資訊，請參閱變更執行個體啟動的關機行為。

> **Note**
>
> 如果停止了 Amazon EBS 支援的執行個體，而該執行個體似乎「凍結」於 `stopping` 狀態，則您可以強制將其停止。如需詳細資訊，請參閱對 Amazon EC2 執行個體停止問題進行疑難排解。

停止與啟動方法

- 在作業系統正常關機的情況下停止執行個體
- 停止執行個體並繞過作業系統正常關機
- 啟動執行個體

在作業系統正常關機的情況下停止執行個體

您可以使用預設停止方法停止執行個體，其中包括嘗試作業系統正常關機。如需詳細資訊，請參閱預設停止。

Console

使用預設停止方法來停止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選取執行個體，然後選取一個執行個體。
3. 選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。如果此選項已停用，則執行個體已停止或其根磁碟區是執行個體儲存體磁碟區。
4. 出現確認提示時，請選擇 Stop (停止)。停止執行個體可能需要幾分鐘。

AWS CLI

使用預設停止方法來停止執行個體

使用 stop-instances 命令。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

使用預設停止方法來停止執行個體

使用 Stop-EC2Instance cmdlet。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

停止執行個體並繞過作業系統正常關機

您可以在停止執行個體時繞過作業系統正常關機。如需詳細資訊，請參閱跳過作業系統關機的停止。

> **Warning**
>
> 繞過作業系統正常關機可能導致資料遺失或損毀 (例如，記憶體內容未寫入磁碟、正在處理的 IO 遺失) 或跳過關機指令碼。

Console

停止執行個體並繞過作業系統正常關機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取該執行個體。
3. 選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
4. 在跳過作業系統關機下，選取跳過作業系統關機核取方塊。若您在主控台中未看見此選項，表示目前區域的主控台尚未提供此功能。不過，您可以透過 AWS CLI 或 SDK 存取此功能，或在主控台中嘗試其他區域。
5. 選擇停止。

AWS CLI

停止執行個體並繞過作業系統正常關機

搭配使用 stop-instances 命令與 `--skip-os-shutdown`。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

停止執行個體並繞過作業系統正常關機

搭配使用 Stop-EC2Instance cmdlet 與 `-SkipOsShutdown $true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -SkipOsShutdown $true
```

啟動執行個體

您可啟動已停止的執行個體。

Console

啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取執行個體，並選取執行個體狀態、啟動執行個體。

   執行個體進入 `running` 狀態可能需要數分鐘。

AWS CLI

啟動執行個體

使用 start-instances 命令。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

啟動執行個體

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

自動停止及啟動執行個體

您可以使用下列服務自動停止及啟動執行個體：

上的執行個體排程器 AWS

您可以在 上使用執行個體排程器 AWS 來自動啟動和停止 EC2 執行個體。如需詳細資訊，請參閱 How do I use Instance Scheduler with CloudFormation to schedule EC2 instances? (如何將執行個體排程器與 CloudFormation 結合使用來設定 EC2 執行個體的排程？) 請注意，這會額外收費。

AWS Lambda 和 Amazon EventBridge 規則

您可以使用 Lambda 和 EventBridge 規則，依排程停用和啟動執行個體。如需詳細資訊，請參閱如何使用 Lambda 定期停用和啟動 Amazon EC2 執行個體？

Amazon EC2 Auto Scaling

為確保您有正確數量的 Amazon EC2 執行個體可用來處理應用程式的負載，請建立 Auto Scaling 群組。 Amazon EC2 Auto Scaling 可確保您的應用程式始終具有處理流量需求的適當容量，且只在需要執行個體時啟動執行個體來節省成本。請注意， Amazon EC2 Auto Scaling 會終止不需要的執行個體，而不是停止。若要設定 Auto Scaling 群組，請參閱開始使用 Amazon EC2 Auto Scaling。

尋找所有執行中和停止的執行個體

您可以使用 Amazon EC2 Global View 在單一頁面上找到所有執行 AWS 區域 中和已停止的執行個體。此功能對清查和查找遺忘的執行個體尤其有用。如需有關如何建立全域檢視的資訊，請參閱 使用 AWS 全域檢視跨區域檢視資源。

或者，您可在擁有執行個體的每個區域中執行命令或 cmdlet。

AWS CLI

取得區域中 EC2 執行個體數量

使用下列 describe-instances 命令來計算目前區域的執行個體數量。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "length(Reservations[].Instances[])"
```

以下為範例輸出。

```
27
```

取得區域中 EC2 執行個體的摘要資訊

使用下面的 describe-instances 命令。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "Reservations[].Instances[].[InstanceId,InstanceType,PrivateIpAddress]"
 \
```

```
    --output table
```

以下為範例輸出。

```
-------------------------------------------------------------
|                   DescribeInstances                       |
+----------------------+---------------+-----------------+
|  i-0e3e777f4362f1bf7|  t2.micro     |  10.0.12.9      |
|  i-09453945dcf1529e9|  t2.micro     |  10.0.143.213   |
|  i-08fd74f3f1595fdbd|  m7i.4xlarge  |  10.0.1.103     |
+----------------------+---------------+-----------------+
```

PowerShell

取得區域中 EC2 執行個體數量

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -Region us-east-2).Instances.Length
```

以下為範例輸出。

```
27
```

取得區域中 EC2 執行個體的摘要資訊

使用 Get-EC2Instance cmdlet。您必須在擁有執行個體的每個區域中執行此命令。

```
(Get-EC2Instance).Instances | Select InstanceId, InstanceType, PrivateIpAddress
```

以下為範例輸出。

```
InstanceId          InstanceType PrivateIpAddress
----------          ------------ ----------------
i-0e3e777f4362f1bf7 t2.micro     10.0.12.9
i-09453945dcf1529e9 t2.micro     10.0.143.213
i-08fd74f3f1595fdbd m7i.4xlarge  10.0.1.103
```

尋找初始和最新的啟動時間

當您描述執行個體時，執行個體的啟動時間是最新的啟動時間。在您停止和啟動執行個體之後，啟動時間會反映新執行個體的啟動時間。若要尋找執行個體的初始啟動時間，即使是在停止和啟動執行個體之後，請檢視主要網路介面連接至執行個體的時間。

Console

尋找最新啟動時間

在詳細資訊索引標籤的執行個體詳細資訊下方選取執行個體，並尋找啟動時間。

尋找初始啟動時間

選取執行個體，然後在聯網索引標籤的網路介面下尋找主要網路介面 (裝置索引為 0)。

AWS CLI

尋找初始和最新的啟動時間

使用下列 describe-instances 命令，以顯示指定執行個體的初始啟動時間和最新啟動時間。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query 'Reservations[].Instances[].
{InstanceID:InstanceId,InitialLaunch:NetworkInterfaces[0].Attachment.AttachTime,LastLaunch:L
```

以下為範例輸出。

```
[
    {
        "InstanceID": "i-1234567890abcdef0",
        "InitialLaunch": "2024-04-19T00:47:08+00:00",
        "LastLaunch": "2024-05-27T06:24:06+00:00"
    }
]
```

PowerShell

尋找最新啟動時間

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.LaunchTime
```

以下為範例輸出。

```
Monday, May 27, 2024 6:24:06 AM
```

尋找初始啟動時間

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -
InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces.Attachment.AttachTime
```

以下為範例輸出。

```
Friday, April 19, 2024 12:47:08 AM
```

為您的 EC2 執行個體啟用停止保護

若要避免執行個體意外停止，您可啟用執行個體的停止保護功能。停止保護還可以保護您的執行個體免遭意外終止。

Amazon EC2 `ModifyInstanceAttribute` API 的 `DisableApiStop` 屬性會控制是否可以使用 Amazon EC2 主控台 AWS CLI、 或 Amazon EC2 API 來停止執行個體。您可在執行個體啟動時、運作中或停止時，設定此屬性值。

考量事項

- 啟用停止保護功能無法防止您在使用作業系統命令 (例如 shutdown 或 poweroff) 從執行個體啟動關機作業，而意外停止執行個體。
- 當有排程事件停止執行個體時，啟用停止保護不會 AWS 阻止 停止執行個體。
- 啟用停止保護不會阻止 Amazon EC2 Auto Scaling 在執行個體狀態不良時或在縮減事件期間終止執行個體。您可以使用執行個體停止保護功能，控制 Auto Scaling 群組是否能在縮減時終止特定的執行個體。
- 停止保護不僅可以防止執行個體意外停止，還可以防止在使用主控台 AWS CLI或 API 時意外終止。但是此功能不會自動設定 `DisableApiTermination` 屬性。請注意，當`DisableApiStop`屬性設定為 時`false`，`DisableApiTermination`屬性設定會決定是否可以使用主控台 AWS CLI或 API 終止執行個體。如需更多資訊，請參閱終止 Amazon EC2 執行個體。

- 您無法為具有執行個體儲存體根磁碟區的執行個體啟用停止保護。
- 您無法為 Spot 執行個體啟用停止保護。
- 當您啟用或停用停止保護時，Amazon EC2 API 會遵循最終一致性模式。這表示執行命令來設定停止保護屬性的結果，可能不會立即顯示在您執行的所有後續命令中。如需詳細資訊，請參閱「Amazon EC2 API 參考」中的最終一致性。

停止保護任務

- 在執行個體啟動時啟用停止保護
- 針對執作中或已停止的執行個體啟用停止保護
- 針對執作中或已停止的執行個體停用停止保護

在執行個體啟動時啟用停止保護

您可以在執行個體啟動時，啟用停止保護。

Console

在執行個體啟動時啟用停止保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在儀表板上，選擇啟動執行個體。
3. 在新版啟動執行個體精靈中設定執行個體。
4. 在精靈中，在進階詳細資訊下的停止保護選擇啟用，即可啟用停止保護功能。

AWS CLI

在執行個體啟動時啟用停止保護

使用 run-instances 命令來啟動執行個體。新增下列參數。

```
--disable-api-stop
```

PowerShell

在執行個體啟動時啟用停止保護

使用 New-EC2Instance cmdlet 來啟動執行個體。新增下列參數。

```
-DisableApiStop $true
```

針對執作中或已停止的執行個體啟用停止保護

您可以在執行個體為執行中或已停止狀態時，啟用停止保護。

Console

為執行個體啟用停止保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取執行個體，然後選擇動作、執行個體設定、變更停止保護。
4. 選取啟用核取方塊，然後選擇儲存。

AWS CLI

為執行個體啟用停止保護

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-stop
```

PowerShell

為執行個體啟用停止保護

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $true
```

針對執作中或已停止的執行個體停用停止保護

您可以使用下列其中一種方法，來對執行中或已停止的執行個體停用停止保護。

Console

針對執行中或已停止的執行個體停用停止保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取執行個體，然後選取 Actions (動作)、Instance Settings (執行個體設定)、Change Stop Protection (變更停止保護)。
4. 清除啟用核取方塊，然後選擇儲存。

AWS CLI

針對執行中或已停止的執行個體停用停止保護

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-stop
```

PowerShell

停用執行個體的停止保護

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $false
```

讓 Amazon EC2 執行個體進入休眠

當您讓執行個體休眠時，Amazon EC2 訊號會對作業系統執行休眠 (暫停磁碟機)。休眠會將執行個體記憶體 (RAM) 的內容儲存於您的 Amazon Elastic Block Store (Amazon EBS) 根磁碟區。Amazon EC2 會保留執行個體的 EBS 根磁碟區，以及任何已連接的 EBS 資料磁碟區。啟動執行個體時：

- EBS 根磁碟區會還原至其先前的狀態
- 重新載入 RAM 的內容

- 恢復先前在執行個體上執行的程序
- 先前連接的資料磁碟區會重新連接，且執行個體會保留其執行個體 ID

只有當您的執行個體已啟用休眠並符合休眠必要條件，才能讓執行個體進入休眠。

如果執行個體或應用程式需要花費長時間來引導並建置記憶體使用量以完全發揮功效，則您可以使用休眠來「預先培養」執行個體。若要「預先培養」執行個體，您需要執行下列作業：

1. 在已啟用休眠的情況下將其啟動。
2. 使其進入所需的狀態。
3. 讓其休眠，以便準備好可在需要時恢復為所需狀態。

若執行個體處於 `stopped` 狀態而休眠，或 RAM 的內容要傳輸到 EBS 根磁碟區，這兩種情況的資料傳輸費用都不需要您支付。不過您需支付任何 EBS 磁碟區的儲存費用，包括 RAM 內容的儲存。

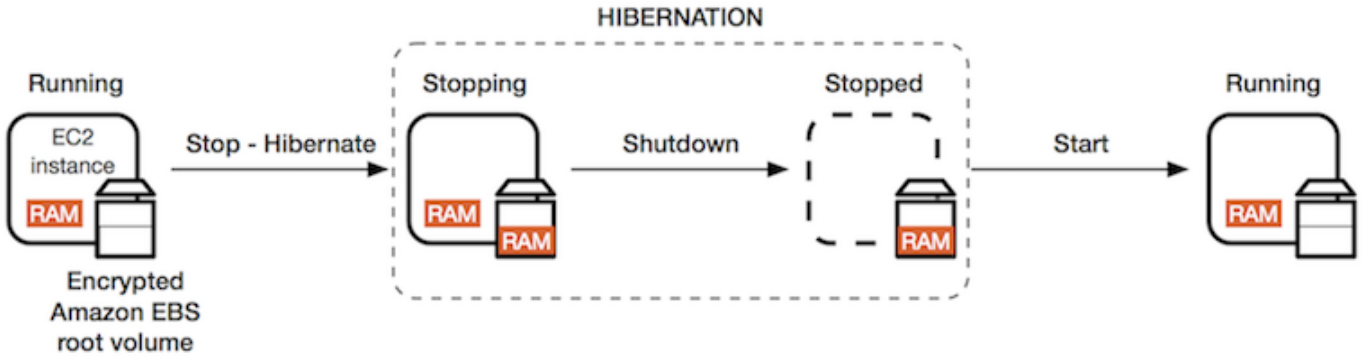
如果您認為不再需要某個執行個體，您可以隨時予以終止，即使該執行個體處於 `stopped` (休眠) 狀態也可以終止。如需詳細資訊，請參閱終止 Amazon EC2 執行個體。

目錄

- Amazon EC2 執行個體休眠的運作方式
- EC2 執行個體休眠的先決條件
- 將 Linux AMI 設定為支援休眠
- 對 Amazon EC2 執行個體啟用休眠
- 在執行個體上停用 KASLR (僅限 Ubuntu)
- 讓 Amazon EC2 執行個體進入休眠
- 啟動休眠的 Amazon EC2 執行個體
- 對 Amazon EC2 執行個體休眠進行疑難排解

Amazon EC2 執行個體休眠的運作方式

下列圖表顯示 EC2 執行個體休眠程序的基本概觀。



若您讓執行個體進入休眠會發生什麼情況

當您讓執行個體進入休眠時，會發生下列情況：

- 執行個體會移至 `stopping` 狀態。Amazon EC2 會對作業系統發出執行休眠 (暫停磁碟機) 的訊號。休眠會凍結所有程序、將執行個體記憶體 (RAM) 的內容儲存於 EBS 根磁碟區，然後執行一般關機。
- 完成關機之後，執行個體會進入 `stopped` 狀態。
- 所有的 EBS 磁碟區仍會繼續連接至執行個體，其中的資料也會持續存在，包括儲存在 RAM 上的內容。
- 任何 Amazon EC2 執行個體存放區磁碟區都會維持連接到執行個體的狀態，但執行個體存放區磁碟區上的資料會遺失。
- 在大部分的情況中，執行個體會在啟動時轉移到新的基礎主機電腦。這也是您在停止並啟動執行個體時會發生的情況。
- 當您啟動執行個體時，執行個體會開機，作業系統會從 EBS 根磁碟區讀取 RAM 的內容，然後將程序恢復為先前的狀態。
- 執行個體會保留其私有 IPv4 位址及任何彈性 IPv6 位址。當您啟動執行個體時，執行個體會繼續保留其私有 IPv4 位址和任何 IPv6 位址。
- Amazon EC2 會釋出公用 IPv4 位址。當您啟動執行個體時，Amazon EC2 會將新公用 IPv4 位址指派給執行個體。
- 執行個體仍會保持與彈性 IP 地址的關聯。我們會針對和已休眠執行個體建立關聯的任何彈性 IP 地址向您收費。

如需休眠與重新開機、停止和終止之間差異的資訊，請參閱執行個體狀態之間的差異。

限制

- 讓執行個體休眠時，任何執行個體存放區磁碟區上的資料都會遺失。

- (Linux 執行個體) 不能讓 RAM 大於 150 GiB 的執行個體休眠。
- (Windows 執行個體) 不能讓 RAM 大於 16 GiB 的執行個體休眠。
- 當您的執行個體休眠時，您無法進行修改。這與未休眠的已停止執行個體不同，您可在其中修改特定屬性，例如執行個體類型或大小。
- 如果您從休眠或已啟用休眠的執行個體中建立快照或 AMI，您可能無法連線至從 AMI 或從快照建立的 AMI 啟動的新執行個體。
- (僅限 Spot 執行個體) 如果 Amazon EC2 讓您的 Spot 執行個體休眠，則只有 Amazon EC2 可以恢復您的執行個體。如果是您讓 Spot 執行個體休眠 (使用者啟動的休眠)，您可以恢復執行個體。只有在有可用的容量，且 Spot 價格低於或等於您指定的最高價格時，才能恢復休眠的 Spot 執行個體。
- 您無法讓位於 Auto Scaling 群組或由 Amazon ECS 使用的執行個體進入休眠。若您的執行個體位於 Auto Scaling 群組中，且您嘗試讓其進入休眠，Amazon EC2 Auto Scaling 服務會將停止的執行個體標記為運作狀態不良，且可能會將其終止並啟動取代用執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的針對 Auto Scaling 群組中的執行個體進行運作狀態檢查。
- 您無法讓設定為在 UEFI 模式下且啟用 UEFI 安全開機進行開機的執行個體進入休眠。
- 如果您休眠啟動到 容量預留 的執行個體，容量預留 不確保休眠執行個體可以在您嘗試啟動之後恢復。
- 如果啟用了聯邦資訊處理標準 (FIPS) 模式，則無法將使用 5.10 以下核心的執行個體休眠。
- 我們不支援讓執行個體休眠超過 60 天。若要保留執行個體 60 天以上，您必須啟動休眠的執行個體、停止執行個體並將其啟動。
- 我們持續透過升級與安全修補程式來更新我們的平台，可能會與現有的已休眠執行個體衝突。在進行需要啟動已休眠執行個體的重大更新時，我們會通知您，以便我們執行關機或重新啟動來套用必要的升級和安全修補程式。

讓 Spot 執行個體休眠的考量

- 如果「您」讓 Spot 執行個體休眠，只要容量可用且 Spot 價格低於或等於您指定的最高價格，就可以重新啟動該執行個體。
- 如果「Amazon EC2」讓您的 Spot 執行個體休眠：
  - 只有 Amazon EC2 才能恢復您的執行個體。
  - 當 Spot 價格低於或等於您所指定的最高價格，而且有可用的容量時，Amazon EC2 會恢復已休眠的 Spot 執行個體。
  - 在 Amazon EC2 讓 Spot 執行個體休眠之前，您會在休眠開始前兩分鐘收到中斷通知。

  如需詳細資訊，請參閱Spot 執行個體中斷。

EC2 執行個體休眠的先決條件

您在啟動隨需執行個體或 Spot 執行個體時，可以啟用休眠支援。您無法在現有執行個體 (無論為執行中或已停止) 啟用休眠。如需詳細資訊，請參閱啟用執行個體休眠。

讓執行個體休眠的要求

- AWS 區域
- AMI
- 執行個體系列
- 執行個體 RAM 大小
- 根磁碟區類型
- 根磁碟區大小
- 根磁碟區加密
- EBS volume type (EBS 磁碟區類型)
- Spot 執行個體請求

AWS 區域

所有 執行個體都可以使用休眠 AWS 區域。

AMI

您必須使用支援休眠的 HVM AMI。下列 AMI 支援休眠：

Linux AMI

Intel 和 AMD 執行個體類型的 AMI

- 2023.09.20 或之後發行的 AL2023 AMI [1]
- 2019.08.29 或更新版本發行的 Amazon Linux 2 AMI。
- 2018.11.16 或更新版本發行的 Amazon Linux AMI 2018.03。
- CentOS 版本 8 AMI [2] (需要設定其他組態)
- Fedora 版本 34 或更新版本 AMI [2] (需要設定其他組態)
- Red Hat Enterprise Linux (RHEL) 9 AMI [2] (需要設定其他組態)
- Red Hat Enterprise Linux (RHEL) 8 AMI [2] (需要設定其他組態)
- Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI，發行序號為 20230303 或更新版本 [3]

- Ubuntu 20.04 LTS (Focal Fossa) AMI，發行序號為 20210820 或更新版本 [3]
- Ubuntu 18.04 LTS (Bionic Beaver) AMI，發行序號為 20190722.1 或更新版本 [3] [5]
- Ubuntu 16.04 LTS (Xenial Xerus) AMI [3] [4] [5] (需要設定其他組態)

Graviton 執行個體類型的 AMI

- 2024.07.01 或之後發行的 AL2023 AMI (64 位元 Arm) [1]
- 2024.06.20 或之後發行的 Amazon Linux 2 AMI (64 位元 Arm)
- Ubuntu 22.04.2 LTS (64 位元 Arm) (Jammy Jellyfish) AMI，發行序號為 20240701 或更新版本 [3]
- Ubuntu 20.04 LTS (64 位元 Arm) (Focal Fossa) AMI，發行序號為 20240701 或更新版本 [3]

[1] 對於 AL2023 最小 AMI，需要額外組態。

[2] 對於 CentOS、Fedora 和 Red Hat Enterprise Linux，只有以 Nitro 為基礎的執行個體才支援休眠。

[3] 建議您在採用 Ubuntu 22.04.2 LTS (Jammy Jellyfish)、Ubuntu 20.04 LTS (Focal Fossa)、Ubuntu 18.04 LTS (Bionic Beaver) 和 Ubuntu 16.04 LTS (Xenial Xerus) 的執行個體上停用 KASLR。如需詳細資訊，請參閱在執行個體上停用 KASLR (僅限 Ubuntu)。

[4] 對於 Ubuntu 16.04 LTS (Xenial Xerus) AMI，`t3.nano` 執行個體類型上不支援休眠。由於 Ubuntu (Xenial Xerus) 已於 2021 年 4 月終止支援，因此不會提供任何修補程式。如果想要使用 `t3.nano` 執行個體類型，則建議升級至 Ubuntu 22.04.2 LTS (Jammy Jellyfish)、Ubuntu 20.04 LTS (Focal Fossa) AMI 或 Ubuntu 18.04 LTS (Bionic Beaver) AMI。

[5] 對 Ubuntu 18.04 LTS (Bionic Beaver) 和 Ubuntu 16.04 LTS (Xenial Xerus) 的支援已接近尾聲。

若要設定您自己的 AMI 以支援休眠，請參閱將 Linux AMI 設定為支援休眠。

即將推出其他版本的 Ubuntu 及其他作業系統的支援。

Windows AMI

- 2023.09.13 或更新版本的 Windows Server 2022 AMI
- 2019.09.11 或更新版本的 Windows Server 2019 AMI
- 2019.09.11 或更新版本的 Windows Server 2016 AMI
- 2019.09.11 或更新版本的 Windows Server 2012 R2 AMI

- 2019.09.11 或更新版本的 Windows Server 2012 AMI

執行個體系列

您必須使用支援休眠的執行個體系列。不過，不支援裸機執行個體。

- 一般用途：
  M3、M4、M5、M5a、M5ad、M5d、M6a、M6g、M6gd、M6i、M6id、M6idn、M6in、M7a、M7g、M7gd
  flex、M8a、M8azn、M8g、M8gb、M8gd、M8gn、M8i、M8i-
  flex、M8in、M8idn、M8ib、M8idb、M9g、M9gd、T2、T3、T3a、T4g
- 運算最佳化：
  C3、C4、C5、C5d、C6a、C6g、C6gd、C6gn、C6i、C6id、C6in、C7a、C7g、C7gd、C7gn、C7i、C7
  flex、C8a、C8g、C8gb、C8gd、C8gn、C8i、C8i-flex、C8in、C8ib、C9g、C9gd
- 記憶體最佳化：
  R3、R4、R5、R5a、R5ad、R5d、R6a、R6g、R6gd、R6idn、R6in、R7a、R7gd、R7i、R7iz、R8a、R8
  flex、R8in、R8idn、R8ib、R8idb、R9g、R9gd、X2gd、X8aedz、X8i
- 儲存最佳化：I3、I3en、I4g、I7i、I7ie、I8g、I8ge、Im4gn、Is4gen

Console

取得支援休眠的執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instance Types (執行個體類型)。
3. 新增篩選條件隨需休眠支援 =true=。
4. (選用) 新增篩選條件以進一步限定到感興趣的特定執行個體類型。

AWS CLI

取得支援休眠的執行個體類型

使用 describe-instance-types 命令。請注意，可用的執行個體類型因區域而異。

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

PowerShell

取得支援休眠的執行個體類型

使用 Get-EC2InstanceType cmdlet。請注意，可用的執行個體類型因區域而異。

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-
Object
```

執行個體 RAM 大小

Linux 執行個體 – 必須小於 150 GiB。

Windows 執行個體 – 必須小於或是等於 16 GiB。若要休眠 T3 或 T3a Windows 執行個體，建議使用至少 1 GiB 的 RAM。

根磁碟區類型

根磁碟區必須是 EBS 磁碟區，而非執行個體存放區。

根磁碟區大小

根磁碟區必須大到足以存放 RAM 的內容，並能容納您預期的使用量，例如作業系統或應用程式。如果您啟用休眠，則會於啟動時在根磁碟區上配置空間以存放 RAM。

根磁碟區加密

根磁碟區必須加密，以保護休眠期間記憶體中的敏感內容。您可以使用 主控台 AWS CLI或 API，透過 Amazon EC2 為執行個體啟用休眠。在執行個體啟動之前，Amazon EC2 會驗證根磁碟區是否已加密。由於 EBS 根磁碟區已加密，因此在休眠期間寫入其中的 RAM 資料也會加密。

⚠ 只有 Amazon EC2 API 會強制encryption-at-rest休眠

Amazon EC2 只會驗證您從 Amazon EC2 開始的休眠先決條件，包括根磁碟區加密。這包括主控台 AWS CLI、 和 API。

如果您在訪客作業系統內啟動休眠，例如 Linux 或 Windows suspend-to-disk，則這些檢查不適用。在這種情況下，請確保 EBS 根磁碟區已加密。作業系統會保留此磁碟區的記憶體內容，再停止執行個體。在共同責任模型下，您負責加密在訪客作業系統中執行之動作的靜態資料。如需詳細資訊，請參閱安全支柱 - AWS Well-Architected Framework 中的共同責任模型和保護靜態資料。

使用下列三個選項中的一個來確保根磁碟區是加密的 EBS 磁碟區：

- 根據預設，EBS 加密 – 您可以根據預設啟用 EBS 加密，以確保在您 AWS 帳戶中建立的所有新 EBS 磁碟區都已加密。以此方式，您可以為您的執行個體啟用休眠，而不在執行個體啟動時指定加密意圖。如需詳細資訊，請參閱預設啟用加密。
- EBS「單一步驟」加密 - 您可以從未加密的 AMI 啟動加密的 EBS 後端 EC2 執行個體，並可同時啟用休眠。如需詳細資訊，請參閱 對 EBS 後端 AMI 使用加密。
- 加密 AMI - 您可以使用加密的 AMI 來啟動您的執行個體，藉此啟用 EBS 加密。如果您的 AMI 沒有加密的根快照，您可以將資料複製到新的 AMI，然後請求加密。如需詳細資訊，請參閱複製期間加密未加密的映像及複製 AMI。

EBS volume type (EBS 磁碟區類型)

EBS 磁碟區必須使用下列其中一個 EBS 磁碟區類型：

- 一般用途 SSD (gp2 和 gp3)
- 佈建 IOPS SSD (io1 和 io2)

如果您選擇佈建 IOPS SSD 磁碟區類型，必須使用適當的 IOPS 佈建 EBS 磁碟區，才能獲得休眠的最佳效能。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 磁碟區類型。

Spot 執行個體請求

針對 Spot 執行個體，適用下列需求：

- Spot 執行個體請求類型必須為 `persistent`。
- 您無法在 Spot 執行個體請求中指定啟動群組。

將 Linux AMI 設定為支援休眠

下列 Linux AMI 可支援讓 Amazon EC2 執行個體進入休眠，但前提是您完成本節中所述的其他組態步驟。

需要設定其他組態：

- 2023.09.20 或之後發行的 AL2023 最小 AMI
- 2019.08.29 或之後發佈的 Amazon Linux 2 最低 AMI

- 2019.08.29 前發佈的 Amazon Linux 2
- 2018.11.16 前發佈的 Amazon Linux
- CentOS 版本 8 或更新版本
- Fedora 版本 34 或更新版本
- Red Hat Enterprise Linux 版本 8 或 9
- Ubuntu 20.04 LTS (Focal Fossa) 在序號 20210820 前發佈的版本
- Ubuntu 18.04 (Bionic Beaver) 在序號 20190722.1 前發佈的版本
- Ubuntu 16.04 (Xenial Xerus)

如需支援休眠且不需要額外組態的 Linux 和 Windows AMI，請參閱 AMI。

如需詳細資訊，請參閱在 Amazon Linux 2 執行個體上更新執行個體軟體。

2023.09.20 或之後發行的 AL2023 最小 AMI

設定 2023.09.20 或之後發行的 AL2023 最小 AMI 以支援休眠

1. 從儲存庫安裝 `ec2-hibinit-agent` 套件。

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

2. 重新啟動服務。

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

2019.08.29 或之後發佈的 Amazon Linux 2 最低 AMI

設定 2019.08.29 或之後發佈的 Amazon Linux 2 最低 AMI 以支援休眠

1. 從儲存庫安裝 `ec2-hibinit-agent` 套件。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

2. 重新啟動服務。

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

2019.08.29 前發佈的 Amazon Linux 2

將 2019.08.29 前發佈的 Amazon Linux 2 AMI 設定為支援休眠

1. 更新核心至 4.14.138-114.102 或更新版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

4. 確認核心版本已更新為 4.14.138-114.102 或更新版本。

```
[ec2-user ~]$ uname -a
```

5. 停止執行個體，並建立 AMI。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

2018.11.16 前發佈的 Amazon Linux

將 2018.11.16 前發佈的 Amazon Linux AMI 設定為支援休眠

1. 更新核心至 4.14.77-70.59 或更新版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

4. 確認核心版本已更新為 4.14.77-70.59 或更高版本。

```
[ec2-user ~]$ uname -a
```

5. 停止執行個體，並建立 AMI。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

CentOS 版本 8 或更新版本

將 CentOS 版本 8 或更新版本的 AMI 設定為支援休眠

1. 更新核心至 4.18.0-305.7.1.el8_4.x86_64 或更新版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 安裝 Fedora Extra Packages for Enterprise Linux (EPEL) 儲存庫。

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

3. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 啟用休眠代理程式以在開機時開始。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

6. 確認核心版本已更新為 4.18.0-305.7.1.el8_4.x86_64 或更新版本。

```
[ec2-user ~]$ uname -a
```

Fedora 版本 34 或更新版本

將 Fedora 版本 34 或更新版本的 AMI 設定為支援休眠

1. 更新核心至 5.12.10-300.fc34.x86_64 或更新版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

3. 啟用休眠代理程式以在開機時開始。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

4. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

5. 確認核心版本已更新為 5.12.10-300.fc34.x86_64 或更新版本。

```
[ec2-user ~]$ uname -a
```

Red Hat Enterprise Linux 版本 8 或 9

設定 Red Hat Enterprise Linux 8 或 9 AMI 以支援休眠

1. 更新核心至 4.18.0-305.7.1.el8_4.x86_64 或更新版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 安裝 Fedora Extra Packages for Enterprise Linux (EPEL) 儲存庫。

   RHEL 版本 8：

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   RHEL 版本 9：

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

3. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 啟用休眠代理程式以在開機時開始。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

6. 確認核心版本已更新為 4.18.0-305.7.1.el8_4.x86_64 或更新版本。

```
[ec2-user ~]$ uname -a
```

Ubuntu 20.04 LTS (Focal Fossa) 在序號 20210820 前發佈的版本

若要將序號 20210820 前發佈的 Ubuntu 20.04 LTS (Focal Fossa) AMI 設定為支援休眠

1. 將 linux-aws-kernel 更新為 5.8.0-1038.40 或更新版本，並將 grub2 為 2.04-1ubuntu26.13 或更新版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

3. 確認核心版本已更新為 5.8.0-1038.40 或更新版本。

```
[ec2-user ~]$ uname -a
```

4. 確認 grub2 版本已更新為 2.04-1ubuntu26.13 或更新版本。

```
[ec2-user ~]$ dpkg --list | grep grub2-common
```

Ubuntu 18.04 (Bionic Beaver) 在序號 20190722.1 前發佈的版本

將序號 20190722.1 前發佈的 Ubuntu 18.04 LTS AMI 設定為支援休眠

1. 更新核心至 4.15.0-1044 或更新版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

4. 確認核心版本已更新為 4.15.0-1044 或更新版本。

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 (Xenial Xerus)

若要將 Ubuntu 16.04 LTS 設為支援休眠，您需要安裝 linux-aws-hwe 核心套件版本 4.15.0-1058-aws 或更新版本，以及 ec2-hibinit-agent。

> **Important**
>
> Canonical 支援 linux-aws-hwe 核心套件。對 Ubuntu 16.04 LTS 的標準支援已於 2021 年 4 月結束，且軟體套件不會再接收定期更新。但是，此套件會收到其他安全性更新，直到 2024 年結束 Extended Security Maintenance 支援為止。如需詳細資訊，請參閱 Canonical Ubuntu 部落格上的 Amazon EC2 Hibernation for Ubuntu 16.04 LTS 現已推出。
> 建議您升級至 Ubuntu 20.04 LTS (Focal Fossa) AMI 或 Ubuntu 18.04 LTS (Bionic Beaver) AMI。

將 Ubuntu 16.04 LTS AMI 設定為支援休眠

1. 更新核心至 4.15.0-1058-aws 或更新版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

2. 從儲存庫安裝 ec2-hibinit-agent 套件。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

4. 確認核心版本已更新為 4.15.0-1058-aws 或更新版本。

```
[ec2-user ~]$ uname -a
```

對 Amazon EC2 執行個體啟用休眠

若要讓執行個體休眠，必須在啟動執行個體時先為其啟用休眠。

> **Important**
> 啟動執行個體之後，就不能對它啟用或停用休眠。

任務

- 為隨需執行個體啟用休眠功能
- 為 Spot 執行個體啟用休眠功能
- 檢視執行個體是否已啟用休眠

為隨需執行個體啟用休眠功能

您可為隨需執行個體啟用休眠功能。

Console

為隨需執行個體啟用休眠功能

1. 遵循程序啟動執行個體，但在完成下列步驟以啟用休眠之前，請勿啟動執行個體。
2. 若要啟用休眠功能，請在啟動執行個體精靈中設定下列欄位：
   a. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 中，選取支援休眠的 AMI。如需詳細資訊，請參閱AMI。

b. 在 Instance type (執行個體類型) 中，選取支援的執行個體類型。如需詳細資訊，請參閱執行個體系列。

c. 在 Configure storage (設定儲存) 中，選擇 Advanced (進階) (在右側)，並指定根磁碟區的下列資訊：

- 針對 Size (GiB) (大小 (GiB))，輸入 EBS 根磁碟區大小。磁碟區必須夠大，才能存放 RAM 內容並容納您預期的使用量。
- 針對 Volume type (磁碟區類型)，選取支援的 EBS 磁碟區類型：一般用途 SSD (gp2 和 gp3) 或佈建 IOPS SSD (io1 和 io2)。
- 針對 Encrypted (加密)，選擇 Yes (是)。如果您在此 AWS 區域中預設啟用加密，則會選取是。
- 針對 KMS key (KMS 金鑰)，選取磁碟區的加密金鑰。如果您在此 AWS 區域中預設啟用加密，則會選取預設加密金鑰。

如需根磁碟區必要條件的詳細資訊，請參閱EC2 執行個體休眠的先決條件。

d. 展開 Advanced details (進階詳細資訊)，針對 Stop - Hibernate behavior (停止 - 休眠行為)，選擇 Enable (啟用)。

3. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

為隨需執行個體啟用休眠功能

使用 run-instances 命令來啟動執行個體。使用 `--block-device-mappings file://mapping.json` 參數指定 EBS 根磁碟區參數，然後使用 `--hibernation-options Configured=true` 參數啟用休眠。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type m5.large \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
```

在 `mapping.json` 中指定下列內容。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName 的值必須符合與 AMI 相關聯的根設備名稱。若要尋找根設備名稱，請使用 describe-images 命令。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

如果您在此 AWS 區域中預設啟用加密，您可以省略 "Encrypted": true。

PowerShell

為隨需執行個體啟用休眠功能

使用 New-EC2Instance 命令來啟動執行個體。先定義區塊型設備映射，然後使用 -BlockDeviceMappings 參數將其新增至命令，以指定 EBS 根磁碟區。使用 -HibernationOptions_Configured $true 參數啟用休眠。

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair
```

DeviceName 的值必須符合與 AMI 相關聯的根設備名稱。若要尋找根設備名稱，請使用 Get-EC2Image 命令。

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

如果您在此 AWS 區域中預設啟用加密，您可以從Encrypted = $true區塊型設備映射中省略 。

為 Spot 執行個體啟用休眠功能

您可為 Spot 執行個體啟用休眠功能。如需中斷時讓 Spot 執行個體休眠的詳細資訊，請參閱 Spot 執行個體中斷。

Console

為 Spot 執行個體啟用休眠功能

1. 遵循程序使用啟動執行個體精靈要求 Spot 執行個體，但在完成下列步驟以啟用休眠之前，請勿啟動執行個體。
2. 若要啟用休眠功能，請在啟動執行個體精靈中設定下列欄位：
   a. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 中，選取支援休眠的 AMI。如需詳細資訊，請參閱AMI。
   b. 在 Instance type (執行個體類型) 中，選取支援的執行個體類型。如需詳細資訊，請參閱執行個體系列。
   c. 在 Configure storage (設定儲存) 中，選擇 Advanced (進階) (在右側)，並指定根磁碟區的下列資訊：
      - 針對 Size (GiB) (大小 (GiB))，輸入 EBS 根磁碟區大小。磁碟區必須夠大，才能存放 RAM 內容並容納您預期的使用量。
      - 針對 Volume type (磁碟區類型)，選取支援的 EBS 磁碟區類型：一般用途 SSD (gp2 和 gp3) 或佈建 IOPS SSD (io1 和 io2)。
      - 針對 Encrypted (加密)，選擇 Yes (是)。如果您在此 AWS 區域中預設啟用加密，則會選取是。
      - 針對 KMS key (KMS 金鑰)，選取磁碟區的加密金鑰。如果您在此 AWS 區域中預設啟用加密，則會選取預設加密金鑰。

      如需根磁碟區必要條件的詳細資訊，請參閱EC2 執行個體休眠的先決條件。

d. 展開進階詳細資訊，除了用於設定 Spot 執行個體的欄位之外，請執行下列操作：

i. 對於要求類型，請選擇持續。

ii. 對於中斷行為，請選擇休眠。或者，對於停止 - 休眠行為，請選擇啟用。這兩個欄位都能在 Spot 執行個體上啟用休眠。您只須設定其中之一。

3. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

為 Spot 執行個體啟用休眠功能

使用 run-instances 命令要求 Spot 執行個體。使用 `--block-device-mappings file://mapping.json` 參數指定 EBS 根磁碟區參數，然後使用 `--hibernation-options Configured=true` 參數啟用休眠。Spot 要求類型 (`SpotInstanceType`) 必須是 `persistent`。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c4.xlarge \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
    --instance-market-options
        {
           "MarketType":"spot",
           "SpotOptions":{
              "MaxPrice":"1",
              "SpotInstanceType":"persistent"
            }
        }
```

在 `mapping.json` 中指定 EBS 根磁碟區參數，如下所示。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
```

```
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName 的值必須符合與 AMI 相關聯的根設備名稱。若要尋找根設備名稱，請使用 describe-images 命令。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

如果您在此 AWS 區域中預設啟用加密，您可以省略 "Encrypted": true。

PowerShell

為 Spot 執行個體啟用休眠功能

使用 New-EC2Instance 命令來要求 Spot 執行個體。先定義區塊型設備映射，然後使用 -BlockDeviceMappings 參數將其新增至命令，以指定 EBS 根磁碟區。使用 -HibernationOptions_Configured $true 參數啟用休眠。

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair `
    -InstanceMarketOption @(
        MarketType = spot;
        SpotOptions @{
        MaxPrice = 1;
        SpotInstanceType = persistent}
    )
```

DeviceName 的值必須符合與 AMI 相關聯的根設備名稱。若要尋找根設備名稱，請使用 Get-EC2Image 命令。

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

如果您在此 AWS 區域中預設啟用加密，您可以從Encrypted = $true區塊型設備映射中省略 。

檢視執行個體是否已啟用休眠

您可檢查執行個體是否啟用休眠。

Console

檢視執行個體是否已啟用休眠

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後在 Instance details (執行個體詳細資訊) 區段的 Details (詳細資料) 標籤上，檢查 Stop-hibernate behavior (停止休眠行為)。Enabled (啟用) 表示執行個體已啟用休眠。

AWS CLI

檢視執行個體是否已啟用休眠

使用 describe-instances 命令，並指定 --filters "Name=hibernation-options.configured,Values=true" 參數來篩選已啟用休眠的執行個體。

```
aws ec2 describe-instances \
    --filters "Name=hibernation-options.configured,Values=true"
```

輸出中的下列欄位表示執行個體已啟用休眠。

```
"HibernationOptions": {
    "Configured": true
}
```

PowerShell

檢視執行個體是否已啟用休眠

使用 Get-EC2Instance cmdlet，並篩選已啟用休眠的執行個體。

```
(Get-EC2Instance `
    -Filter @{Name="hibernation-options.configured"; Values="true"}).Instances
```

在執行個體上停用 KASLR (僅限 Ubuntu)

若要在配備 Ubuntu 16.04 LTS (Xenial Xerus)、Ubuntu 18.04 LTS (Bionic Beaver) (發行序號為 20190722.1 或更新版本) 或 Ubuntu 20.04 LTS (Focal Fossa) (發行序號為 20210820 或更新版本) 的新啟動執行個體上執行休眠，建議停用 KASLR (Kernel Address Space Layout Randomization)。在 Ubuntu 16.04 LTS、Ubuntu 18.04 LTS 或 Ubuntu 20.04 LTS 上，KASLR 預設為啟用。

KASLR 是標準的 Linux 核心安全功能，可以將核心的基底位址值隨機化，有助於儘量避免受到尚未發現的記憶體存取漏洞所拖累和影響。在啟用 KASLR 的情況下，執行個體在休眠之後可能不會恢復。

若要進一步了解 KASLR，請參閱 Ubuntu 功能。

在使用 Ubuntu 啟動的執行個體上停用 KASLR

1. 使用 SSH 連線到您的執行個體。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
2. 在您選擇的編輯器中開啟 `/etc/default/grub.d/50-cloudimg-settings.cfg` 檔案。編輯 `GRUB_CMDLINE_LINUX_DEFAULT` 行，在行尾附加 `nokaslr` 選項，如下列範例所示。

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
    nvme_core.io_timeout=4294967295 nokaslr"
   ```
3. 儲存檔案並結束您的編輯器。
4. 執行下列命令來重建 grub 組態。

   ```
   sudo update-grub
   ```
5. 重新啟動執行個體。

   ```
   sudo reboot
   ```
6. 執行以下命令來確認已新增 `nokaslr`。

   ```
   cat /proc/cmdline
   ```

命令的輸出應該會包含 `nokaslr` 選項。

讓 Amazon EC2 執行個體進入休眠

如果執行個體是 EBS 支援的執行個體、啟用休眠功能並符合休眠必要條件，您可以在隨需執行個體或 Spot 執行個體上啟動休眠。如果無法讓執行個體進入休眠，則會以一般方式關機。

Console

讓執行個體休眠

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Instance state (執行個體狀態)、Hibernate instance (休眠執行個體)。如果停用了 Hibernate instance (休眠執行個體)，則已進入休眠或已停止的執行個體將無法進入休眠。如需詳細資訊，請參閱EC2 執行個體休眠的先決條件。
4. 出現確認提示時，請選擇 Hibernate (休眠)。讓執行個體進入休眠可能需要幾分鐘。執行個體狀態會先變更為 Stopping (正在停止)，然後在執行個體已進入休眠時變更為 Stopped (已停止)。

AWS CLI

讓執行個體休眠

使用 stop-instances 命令，並指定 `--hibernate` 參數。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

PowerShell

讓執行個體休眠

使用 Stop-EC2Instance cmdlet。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
```

```
    -Hibernate $true
```

您可以檢查，是否已在執行個體上啟動休眠。

Console

檢視執行個體是否已啟動休眠

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後在詳細資訊索引標籤的執行個體詳細資訊區段中，查看狀態轉換訊息的值。

   Client.UserInitiatedHibernate: User initiated hibernate 表示您已在隨需執行個體或 Spot 執行個體上啟動休眠。

AWS CLI

檢視執行個體是否已啟動休眠

使用 describe-instances 命令並指定 `state-reason-code` 篩選條件，以查看已初始化休眠的執行個體。

```
aws ec2 describe-instances \
    --filters "Name=state-reason-code,Values=Client.UserInitiatedHibernate"
```

輸出中的下列欄位表示隨需執行個體或 Spot 執行個體已啟動休眠。

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
}
```

PowerShell

檢視執行個體是否已啟動休眠

使用 Get-EC2Instance cmdlet 並指定 `state-reason-code` 篩選條件，以查看已初始化休眠的執行個體。

```
Get-EC2Instance `
```

```
    -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

啟動休眠的 Amazon EC2 執行個體

透過與啟動已停止執行個體的相同方式，來啟動休眠的執行個體。

對於 Spot 執行個體，如果是 Amazon EC2 讓執行個體休眠，則只有 Amazon EC2 可以恢復執行個體。只有是「您」讓 Spot 執行個體休眠時，您才能恢復該執行個體。只有在容量可用，且 Spot 價格低於或等於您指定的最高價格時，才能恢復 Spot 執行個體。

Console

啟動休眠的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取休眠的執行個體，並選取 Instance state (執行個體狀態)、Start instance (啟動執行個體)。執行個體進入 `running` 狀態可能需要數分鐘。在此期間內，執行個體的狀態檢查會顯示執行個體處於失敗狀態，直到執行個體啟動。

AWS CLI

啟動休眠的執行個體

使用 start-instances 命令。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

啟動休眠的執行個體

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

對 Amazon EC2 執行個體休眠進行疑難排解

使用此資訊來協助您診斷並修正讓執行個體進入休眠時可能遇到的問題。

休眠問題

- 無法在啟動後立即進入休眠
- 從 stopping 轉換為 stopped 花費太長時間，且記憶體狀態未在啟動後復原
- 執行個體「卡」在stopping 狀態。
- 休眠後無法立即啟動 Spot 執行個體
- 繼續執行 Spot 執行個體失敗

無法在啟動後立即進入休眠

如果您在啟動執行個體後太快嘗試讓執行個體進入休眠，則會出現錯誤。

在啟動後，Linux 執行個體必須等待約兩分鐘，Windows 執行個體必須等待約五分鐘，才能進入休眠狀態。

從 stopping 轉換為 stopped 花費太長時間，且記憶體狀態未在啟動後復原

如果休眠的執行個體花費長時間才能從 `stopping` 狀態轉換為 `stopped`，且如果記憶體狀態未在啟動後復原，則可能表示休眠未正確設定。

Linux 執行個體

請檢查執行個體系統日誌，尋找與休眠相關的訊息。若要存取系統日誌，請連線到執行個體，或使用 get-console-output 命令。從 `hibinit-agent` 尋找日誌行。如果日誌行指出失敗或日誌行不存在，則很可能在啟動時有錯誤的休眠設定。

例如，下列訊息表示執行個體根磁碟區不夠大：`hibinit-agent: Insufficient disk space. Cannot create setup for hibernation. Please allocate a larger root device.`

如果 `hibinit-agent` 最後的日誌行是 `hibinit-agent: Running: swapoff /swap`，則表示休眠已成功設定。

如果您沒有在程序中看到任何日誌，則您的 AMI 可能不支援休眠。如需支援的 AMI 資訊，請參閱EC2 執行個體休眠的先決條件。如果您使用自己的 Linux AMI，請確保遵循 將 Linux AMI 設定為支援休眠 的指示。

Windows Server 2016 及更新版本

請檢查 EC2 啟動日誌，尋找與休眠相關的訊息。若要存取 EC2 啟動日誌，連線至執行個體並在文字編輯器中開啟 `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\Ec2Launch.log` 檔案。如果使用 EC2Launch v2，請開啟 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`。

> **Note**
>
> 根據預設，Windows 會將檔案和資料夾隱藏在 `C:\ProgramData` 下。若要檢視 EC2 啟動目錄和檔案，您必須在 Windows 檔案總管中輸入路徑或變更資料夾屬性，才可顯示隱藏的檔案和資料夾。

尋找休眠的日誌行。如果日誌行指出失敗或日誌行不存在，則很可能在啟動時有錯誤的休眠設定。

例如，下列訊息表示無法設定休眠：`Message: Failed to enable hibernation.`。如果錯誤訊息包含十進位 ASCII 值，您可以將 ASCII 值轉換為純文字，以便讀取完整的錯誤訊息。

如果日誌行包含 `HibernationEnabled: true`，則表示休眠已成功設定。

Windows Server 2012 R2 和舊版

請檢查 EC2 設定日誌，尋找與休眠相關的訊息。若要存取 EC2 設定日誌，請連線至執行個體，接著在文字編輯器中開啟 `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt` 檔案。尋找 `SetHibernateOnSleep` 的日誌行。如果日誌行指出失敗或日誌行不存在，則很可能在啟動時有錯誤的休眠設定。

例如，下列訊息表示執行個體根磁碟區不夠大：`SetHibernateOnSleep: Failed to enable hibernation: Hibernation failed with the following error: There is not enough space on the disk.`

如果日誌行是 `SetHibernateOnSleep: HibernationEnabled: true`，則表示休眠已成功設定。

Windows 執行個體大小

如果您使用的是具有少於 1 GiB RAM 的 T3 或 T3a Windows 執行個體，請嘗試將執行個體的大小增加為具有至少 1 GiB RAM 的執行個體大小。

執行個體「卡」在stopping 狀態。

如果您讓執行個體進入休眠，但該執行個體「卡」在stopping 狀態，您可以強制將其停止。如需詳細資訊，請參閱對 Amazon EC2 執行個體停止問題進行疑難排解。

休眠後無法立即啟動 Spot 執行個體

如果您嘗試在休眠後兩分鐘內啟動 Spot 執行個體，可能會出現下列錯誤：

```
You failed to start the Spot Instance because the associated Spot Instance
request is not in an appropriate state to support start.
```

Linux 執行個體須等待大約兩分鐘，Windows 執行個體須等待大約五分鐘，然後重試啟動執行個體。

繼續執行 Spot 執行個體失敗

如果您的 Spot 執行個體已成功休眠，但是無法繼續執行，而是會重新啟動 (未保留休眠狀態而直接重新啟動)，可能是因為使用者資料包含下列指令碼：

```
/usr/bin/enable-ec2-spot-hibernation
```

從啟動範本的使用者資料欄位中移除此指令碼，然後請求新的 Spot 執行個體。

請注意，即使執行個體無法繼續執行，在不保留休眠狀態的狀態下，仍可使用與從 `stopped` 狀態啟動的相同方式啟動執行個體。

重新啟動您的 Amazon EC2 執行個體

重新啟動執行個體等同於重新啟動作業系統。在大多數情況下，將執行個體重新開機只需要幾分鐘的時間。

重新啟動執行個體時，其會保留下列項目：

- 公有 DNS 名稱 (IPv4)
- 私有 IPv4 地址
- 公有 IPv4 地址
- IPv6 地址 (如適用)
- 其執行個體儲存體磁碟區上的任何資料

有別於停用和啟動執行個體 (會開啟新帳單週期，最少收費一分鐘)，重新啟動執行個體不會啟動新的執行個體帳單期間。

執行個體重新啟動可以由使用者啟動 （其中您手動重新啟動執行個體） 或由 啟動 AWS （用於自動執行個體復原，或回應排程的重新啟動事件以進行必要的維護，例如套用需要重新啟動的更新）。

對於使用者啟動的重新啟動，建議您使用 Amazon EC2 主控台、CLI 或 API，而非從執行個體中執行作業系統重新啟動命令。使用 Amazon EC2 時，若執行個體未在幾分鐘內正常關機，Amazon EC2 會執行強制重新啟動。此外， 會在執行個體重新啟動時 AWS CloudTrail 建立 的 API 記錄。

本主題說明如何執行使用者啟動的重新啟動。如需 所執行重新啟動的相關資訊 AWS，請參閱 自動執行個體復原功能和 管理已排程進行重新啟動的 Amazon EC2 執行個體。

> **Important**
>
> 如果正在執行個體上安裝更新，建議您不要在所有更新安裝完畢之前，使用 Amazon EC2 主控台或命令列來重新啟動或關機執行個體。當您使用 Amazon EC2 主控台或命令列重新開機或關機執行個體時，會有硬式重新開機執行個體的風險。正在安裝更新時的硬式重新開機可能會讓執行個體進入不穩定狀態。

Console

重新啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 Instance state (執行個體狀態)、Reboot instance (將執行個體重新開機)。
4. 出現確認提示時，選擇重新開機。

   執行個體會保持在 `running` 狀態。

AWS CLI

重新啟動執行個體

使用 reboot-instances 命令。

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef0
```

PowerShell

重新啟動執行個體

使用 Restart-EC2Instance cmdlet。

```
Restart-EC2Instance -InstanceId i-1234567890abcdef0
```

若要執行受控的錯誤注入實驗

您可以使用 AWS Fault Injection Service 來測試重新啟動執行個體時應用程式如何回應。如需詳細資訊，請參閱「AWS Fault Injection Service 使用者指南」。

終止 Amazon EC2 執行個體

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

當您不再需要執行個體時，可以將它刪除。這就是終止您的執行個體。執行個體的狀態變更為 `shutting-down` 或 `terminated` 時刻起，該執行個體便停止收取費用。

執行個體終止後，您將無法連接或啟動該執行個體。然而，您可使用相同 AMI 來啟動新的執行個體。

若您希望停止執行個體，或讓其進入休眠，請參閱 停止和啟動 Amazon EC2 執行個體 或 讓 Amazon EC2 執行個體進入休眠。如需詳細資訊，請參閱執行個體狀態之間的差異。

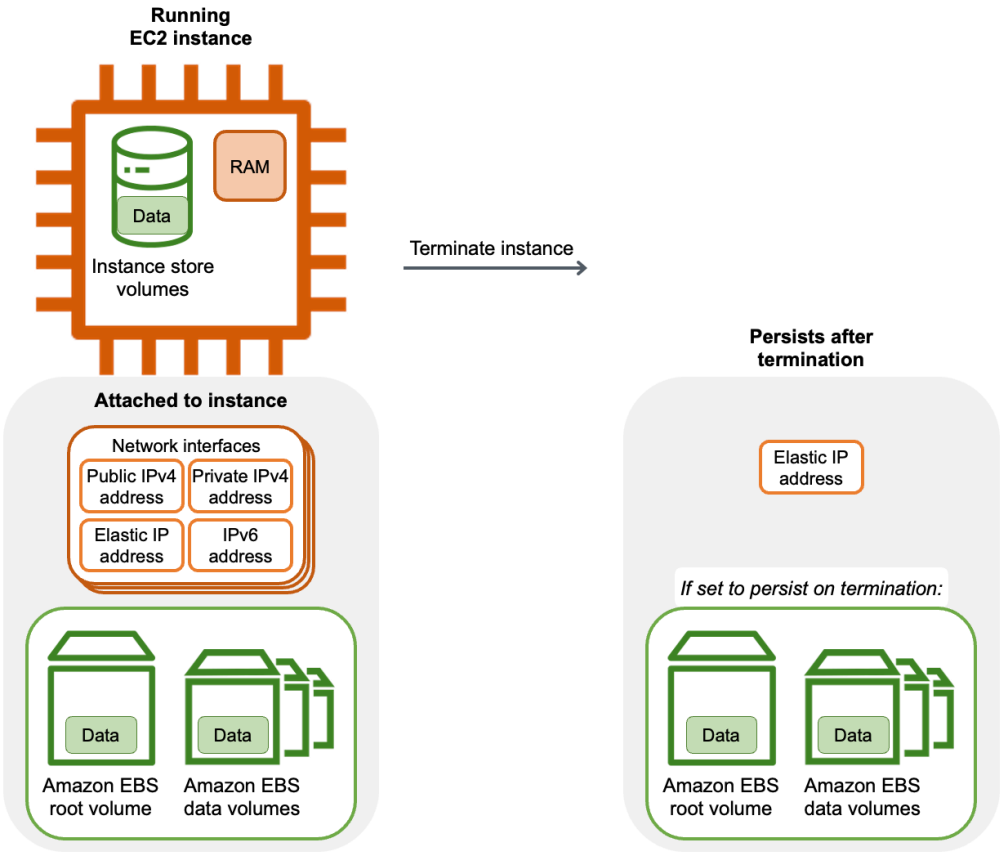
目錄

- 終止執行個體的運作方式
- 終止執行個體的方法
- 在作業系統正常關機的情況下終止執行個體
- 終止執行個體並繞過作業系統正常關機
- 對執行個體終止進行故障診斷
- 變更執行個體終止保護
- 變更執行個體啟動的關機行為
- 在執行個體終止時保留資料

終止執行個體的運作方式

當您終止執行個體時，變更會在執行個體的作業系統 (OS) 層級註冊，而有些資源會遺失，有些則會保留。

下圖顯示 Amazon EC2 執行個體終止時遺失及持續存在的內容。當執行個體終止時，任何執行個體儲存體磁碟區上的資料和存放執行個體 RAM 的資料都會遭到清除。已停止連結與執行個體相關聯的任何彈性 IP 位址。針對 Amazon EBS 根磁碟區和資料磁碟區，結果取決於磁碟區的終止時刪除設定。



考量事項

- 資料持久性
  - 執行個體儲存體磁碟區：當執行個體終止時，所有資料都會永久刪除。
  - EBS 根磁碟區：
    - 若在啟動時連接，預設會在執行個體終止時刪除。
    - 若啟動後連接，預設會在執行個體終止時持續存在。
  - EBS 資料磁碟區：
    - 若使用主控台在啟動時連接：執行個體終止時，預設會持續存在。

  - 若使用 CLI 在啟動時連接：預設在執行個體終止時刪除。
  - 若使用主控台或 CLI 在啟動後連接：執行個體終止時，預設會持續存在。

> **Note**
>
> 執行個體終止時未刪除的任何磁碟區將繼續產生費用。您可變更設定，以在執行個體終止時刪除或保留磁碟區。如需詳細資訊，請參閱在執行個體終止時保留資料。

- 防止意外終止
  - 若要避免遭人員意外終止執行個體，可啟用執行個體的終止保護。
  - 若要控制執行個體從執行個體觸發關機時，執行個體是否停止或終止，請變更執行個體觸發的關機行為。
- 關閉指令碼 - 如果您在執行個體終止時執行指令碼，則執行個體可能會發生異常終止，因為我們無法確保關閉指令碼執行。Amazon EC2 會嘗試完全關閉執行個體，並執行任何系統關閉指令碼；不過，某些事件 (例如硬體故障) 可能會使這些系統關閉指令碼無法執行。
- 裸機執行個體 – x86 裸機執行個體不支援協作關機。

若您終止執行個體會如何

在作業系統層級註冊的變更

- API 請求會將按鈕事件傳送給訪客。
- 由於按鈕事件而停止各種系統服務。系統的順利關機是由 systemd (Linux) 或系統流程 (Windows) 提供。順利關機由從 Hypervisor 按下 ACPI 關機按鈕事件觸發。
- 啟動 ACPI 關機。
- 當順利關機程序結束後，執行個體就會關機。OS 關機時間無法設定。執行個體仍會短暫顯示於主控台，之後該項目會自動刪除。

資源遺失

- 存放在執行個體儲存體磁碟區的資料。
- 若 `DeleteOnTermination` 屬性設定為 `true`，則為 EBS 根磁碟區。
- 若 `DeleteOnTermination` 屬性設定為 `true`，EBS 資料磁碟區 (在啟動時或之後連接)。

持續存在的資源

- 若 DeleteOnTermination 屬性設定為 false，則為 EBS 根磁碟區。
- 若 DeleteOnTermination 屬性設定為 false，EBS 資料磁碟區 (在啟動時或之後連接)。

測試應用程式對執行個體終止的回應

您可以使用 AWS Fault Injection Service 來測試您的應用程式在執行個體終止時如何回應。如需詳細資訊，請參閱「AWS Fault Injection Service 使用者指南」。

終止執行個體的方法

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

使用者發起的執行個體終止有四種方式：預設終止、跳過作業系統關機的終止、強制終止，以及跳過作業系統關機的強制終止。下表比較各終止方法之間的核心差異：

> **Note**
>
> 如果開啟終止保護，則無法終止執行個體。如需詳細資訊，請參閱變更執行個體終止保護。

終止方法	主要用途	使用案例	CLI 命令
預設終止	嘗試作業系統正常關機的一般執行個體關機。	一般執行個體終止。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0`

終止方法	主要用途	使用案例	CLI 命令
透過跳過作業系統關機來終止	終止執行個體時繞過作業系統正常關機。	需要繞過作業系統正常關機時。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`
強制終止	處理停滯的執行個體。會先嘗試預設終止；若執行個體無法終止，則強制終止執行個體。	執行個體卡在 shutting-down 狀態時。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force`
在跳過作業系統關機時強制終止	終止執行個體時，強制終止並繞過作業系統正常關機。	需要強制終止並繞過作業系統正常關機時。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

各方法的使用指示，請參閱下列內容：

- 在作業系統正常關機的情況下終止執行個體
- 終止執行個體並繞過作業系統正常關機
- 強制終止執行個體

在作業系統正常關機的情況下終止執行個體

您可以使用預設終止方法終止執行個體，其中包括嘗試作業系統正常關機。如需詳細資訊，請參閱終止執行個體的方法。

Console

使用預設終止方法以終止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選取執行個體狀態、終止 (刪除) 執行個體。
4. 出現確認提示時，請選擇終止 (刪除)。
5. 在執行個體終止後，仍會以 `terminated` 狀態短暫顯示。

   如果終止失敗，或終止的執行個體已顯示超過數小時，請參閱 仍顯示已終止的執行個體。

AWS CLI

使用預設終止方法以終止執行個體

使用 terminate-instances 命令。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

使用預設終止方法以終止執行個體

使用 Remove-EC2Instance cmdlet。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

終止執行個體並繞過作業系統正常關機

您可以在終止執行個體時繞過作業系統正常關機。如需詳細資訊，請參閱終止執行個體的方法。

Console

終止執行個體並繞過作業系統正常關機

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選取執行個體狀態、終止 (刪除) 執行個體。
4. 在跳過作業系統關機下，選取跳過作業系統關機核取方塊。若您在主控台中未看見此選項，表示目前區域的主控台尚未提供此功能。不過，您可以使用 AWS CLI 或 SDK 存取此功能，或在主控台中嘗試其他區域。
5. 選擇終止 (刪除)。
6. 在執行個體終止後，仍會以 `terminated` 狀態短暫顯示。

   如果終止失敗，或終止的執行個體已顯示超過數小時，請參閱 仍顯示已終止的執行個體。

AWS CLI

終止執行個體並繞過作業系統正常關機

搭配使用 terminate-instances 命令與 `--skip-os-shutdown`。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

終止執行個體並繞過作業系統正常關機

搭配使用 Remove-EC2Instance cmdlet 與`-SkipOsShutdown $true`。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -SkipOsShutdown $true
```

對執行個體終止進行故障診斷

請求者必須具有呼叫 `ec2:TerminateInstances` 的許可。如需詳細資訊，請參閱使用執行個體的範例政策。

如果終止您的執行個體，並啟動另一個執行個體，很可能您已經透過像 EC2 Fleet 或 Amazon EC2 Auto Scaling 這樣的功能來設定自動擴展。如需詳細資訊，請參閱執行個體自動啟動或終止。

> **Note**
>
> 如果開啟終止保護，則無法終止執行個體。如需詳細資訊，請參閱變更執行個體終止保護。

若您的執行個體處於 `shutting-down` 狀態的時間比平常更長，您可以嘗試強制終止。若您保持在 `shutting-down` 狀態，應該會由 Amazon EC2 服務內的自動化程序清理 (終止)。如需詳細資訊，請參閱延遲的執行個體終止。

變更執行個體終止保護

若要防止執行個體透過 Amazon EC2 API 被意外終止，無論您是直接呼叫 `TerminateInstances`，還是使用 Amazon EC2 主控台等其他介面，請為該執行個體啟用終止保護。`DisableApiTermination` 屬性可控制執行個體是否可終止。您執行個體的終止保護預設為停用。您可在執行個體啟動時、運作中或停止時，設定此屬性值。

在 `InstanceInitiatedShutdownBehavior` 屬性設定為 `terminate` 時，`DisableApiTermination` 屬性不會讓您無法從執行個體啟動關機 (例如，使用作業系統命令進行系統關機) 來終止該執行個體。如需詳細資訊，請參閱變更執行個體啟動的關機行為。

考量事項

- 當有終止執行個體的排程事件時，啟用終止保護不會 AWS 阻止 終止執行個體。
- 啟用終止保護不會阻止 Amazon EC2 Auto Scaling 在執行個體狀態不良時或在縮減事件期間終止執行個體。您可以使用執行個體停止保護功能，控制 Auto Scaling 群組是否能在縮減時終止特定的執行個體。您可以暫停 ReplaceUnhealthy 縮減程序，以控制 Auto Scaling 群組是否能終止運作狀態不佳的執行個體。
- 您無法為 競價型執行個體啟用終止保護。

Console

欲在執行個體啟動時啟用終止保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在儀表板上，選擇啟動執行個體。

3. 展開 Advanced Details (進階詳細資訊)。對於終止保護，選取啟用。
4. 完成執行個體的詳細資訊的指定後，請選擇啟動執行個體。

變更執行個體的終止保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)。
3. 選取執行個體。
4. 選擇動作、執行個體設定、變更終止保護。
5. 對於終止保護，請選取或清除啟用。
6. 選擇儲存。

AWS CLI

欲在執行個體啟用終止保護

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-termination
```

停用執行個體的終止保護

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-termination
```

PowerShell

欲在執行個體啟用終止保護

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
```

```
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $true
```

停用執行個體的終止保護

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $false
```

使用終止保護來終止多個執行個體

在相同需求中，若您跨多個可用區域中終止多個執行個體，並啟用一個或多個指定的執行個體以進行終止保護，則要求會失敗，並顯示下列結果：

- 不會終止與受保護執行個體位於相同可用區域中的指定執行個體。
- 位於不同可用區域中的指定執行個體 (其中沒有其他指定執行個體受到保護) 會成功終止。

範例

假設您在兩個可用區域中有以下四個執行個體。

執行個體	可用區域	終止保護
執行個體 1	AZ A	Disabled
執行個體 2		Disabled
執行個體 3	AZ B	Enabled
執行個體 4		Disabled

如果您嘗試在同一個請求中終止所有這些執行個體，請求會報告失敗，結果如下：

- 執行個體 1 和執行個體 2 已成功終止，因為兩個執行個體皆未啟用終止保護。
- 執行個體 3 和執行個體 4 無法終止，因為執行個體 3 已啟用終止保護。

變更執行個體啟動的關機行為

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

根據預設，從 Amazon EBS 後端執行個體起始關閉動作時 (使用 shutdown 或 poweroff 等命令)，該執行個體會停止。您可變更此行為，來讓執行個體終止，而無需變更執行個體的 `InstanceInitiatedShutdownBehavior` 屬性。您可在執行個體運作中或停止時，變更此屬性。

halt 命令不會啟動關機。如果使用，則執行個體不會終止；而是會將 CPU 放入 HLT，而且執行個體仍會繼續執行。

> **Note**
>
> `InstanceInitiatedShutdownBehavior` 屬性僅適用從執行個體的作業系統執行關機，不適用使用 `StopInstances` API 或 Amazon EC2 主控台來停止執行個體。

Console

變更執行個體啟動的關機行為

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取實例。
4. 依序選擇 Actions (動作)、Instance settings (執行個體設定)、Change shutdown behavior (變更關機行為)。

   關機行為會顯示目前的行為。
5. 若要變更此行為，對於關機行為，選擇停止或終止。
6. 選擇儲存。

AWS CLI

變更執行個體啟動的關機行為

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-initiated-shutdown-behavior terminate
```

PowerShell

變更執行個體啟動的關機行為

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -InstanceInitiatedShutdownBehavior terminate
```

在執行個體終止時保留資料

當 Amazon EC2 執行個體終止時，您可能會想要保留執行個體儲存體磁碟區或 Amazon EBS 磁碟區上的資料。本主題說明如何確保您的資料在執行個體終止後仍能保留。

執行個體終止如何影響根磁碟區與資料磁碟區

執行個體儲存體磁碟區

執行個體終止時，執行個體儲存體磁碟區上的任何資料會自動刪除，其中的資料也會遺失。要將此資料保留超過執行個體的存留期，在終止執行個體之前，手動將該資料複製到更持久的儲存體，例如 Amazon EBS 磁碟區、Amazon S3 儲存貯體或 Amazon EFS 檔案系統。如需詳細資訊，請參閱Amazon EC2 執行個體的儲存選項。

Amazon EBS 磁碟區

執行個體終止時，EBS 磁碟區會被刪除或保留，取決於每個磁碟區的 `DeleteOnTermination` 屬性值：

- 是 (主控台) / `true` (CLI) – 磁碟區會在執行個體終止時遭到刪除。
- 否 (主控台) / `false` (CLI) – 磁碟區會在執行個體終止時保留。保留的磁碟區會持續產生費用。

> **Note**
>
> 執行個體終止後，您可拍攝已保留磁碟區的快照，並將其連接到另一個執行個體。要避免產生費用，您必須刪除磁碟區。

EBS 磁碟區的預設刪除行為

預設 `DeleteOnTermination` 值會根據磁碟區類型、磁碟區是在啟動時還是之後連接，以及用來連接磁碟區的方法 (主控台或 CLI) 而有所不同：

磁碟區類型	連接時間	連接方法	執行個體終止時的預設行為
根磁碟區	啟動時	主控台或 CLI	刪除
根磁碟區	After launch (啟動後)	主控台或 CLI	保留
資料量	啟動時	主控台	保留
資料量	啟動時	CLI	刪除
資料量	After launch (啟動後)	主控台和 CLI	保留

檢查磁碟區持久性設定

EBS 磁碟區在啟動時的預設值取決於 AMI 上設定的 `DeleteOnTermination` 屬性。您可以在啟動執行個體時變更此值，覆寫 AMI 的設定。啟動執行個體後，建議您驗證 `DeleteOnTermination` 屬性的預設設定。

檢查執行個體終止時是否會刪除 Amazon EBS 磁碟區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇儲存分頁。
5. 在區塊型儲存設備下，向右捲動以檢查終止時刪除欄。

- 如果為是，磁碟區會在執行個體終止時遭到刪除。
- 如果為否，磁碟區不會在執行個體終止時遭到刪除。未刪除的所有磁碟區都會持續產生費用。

在啟動時將根磁碟區變更為保留

您可以在啟動執行個體時變更 EBS 根磁碟區的 `DeleteOnTermination` 屬性。您也可對資料磁碟區使用下列程序。

Console

欲在啟動時將執行個體的根磁碟區變更為保留

1. 遵循程序啟動執行個體，但在完成下列步驟以將根磁碟區變更為保留之前，請勿啟動執行個體。
2. 在設定儲存窗格中，選擇進階。
3. 在 EBS 磁碟區下方，展開根磁碟區下的資訊。
4. 針對在終止時刪除，選擇否。
5. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

欲在啟動時將執行個體的根磁碟區變更為保留

使用 run-instances 命令，變更區塊型儲存設備映射中的 `DeleteOnTermination` 值。

新增 `--block-device-mappings` 選項：

```
--block-device-mappings file://mapping.json
```

在 `mapping.json` 中，指定裝置名稱，例如 `/dev/sda1` 或者 `/dev/xvda`，針對 `DeleteOnTermination`，指定 `false`。

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
```

```
    }
  }
]
```

PowerShell

欲在啟動時將執行個體的根磁碟區變更為保留

使用 New-EC2Instance cmdlet，變更區塊型儲存設備映射中的 DeleteOnTermination 值。

新增 -BlockDeviceMapping 選項：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定裝置名稱，例如 /dev/sda1 或者 /dev/xvda，針對 DeleteOnTermination，指定 false。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

將執行中執行個體的根磁碟區變更為保留

您可以將執行中執行個體的 EBS 根磁碟區變更為保留。您也可對資料磁碟區使用下列程序。

AWS CLI

將根磁碟區變更為持續存在

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0  \
    --block-device-mappings file://mapping.json
```

在 mapping.json 中，指定裝置名稱，例如 /dev/sda1 或者 /dev/xvda，針對 --DeleteOnTermination，指定 false。

```
[
```

```
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

將根磁碟區變更為持續存在

使用 Edit-EC2InstanceAttribute cmdlet。

新增 `-BlockDeviceMapping` 選項：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定裝置名稱，例如 `/dev/sda1` 或者 `/dev/xvda`，針對 `DeleteOnTermination`，指定 `false`。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

執行個體淘汰

當 AWS 偵測到託管執行個體的基礎硬體發生無法修復的故障時，執行個體會排程淘汰。執行個體根磁碟區類型會決定執行個體淘汰的行為：

- 如果您的執行個體根磁碟區是 Amazon EBS 磁碟區，執行個體會停止，且您可以隨時再次將其啟動。啟動停止的執行個體會將其遷移至新硬體。
- 如果您的執行個體根磁碟區是執行個體儲存體磁碟區，執行個體會終止，且無法再次使用。

如需執行個體事件類型的詳細資訊，請參閱Amazon EC2 執行個體的排程事件。

內容

- 識別排程淘汰的執行個體
- 針對已排程淘汰的 EBS 後端執行個體採取的動作
- 針對排程淘汰的執行個體存放區支援執行個體採取的動作

識別排程淘汰的執行個體

如果已排程淘汰您的執行個體，則您會在事件執行之前收到一封內含執行個體 ID 及淘汰日期的電子郵件。您也可以檢查排程為淘汰的執行個體。

> **Important**
>
> 如果執行個體已排程停用，建議您儘快採取動作，因為執行個體可能已經無法連線。如需詳細資訊，請參閱Check if your instance is reachable。

確定排程淘汰的執行個體的選項

- 監控帳戶聯絡人的電子郵件
- 檢查您的執行個體

監控帳戶聯絡人的電子郵件

若執行個體已排定淘汰，帳戶的主要聯絡人和營運聯絡人會在事件發生前收到電子郵件。此電子郵件包含了執行個體 ID 和排定的淘汰日期。如需詳細資訊，請參閱《 AWS 帳戶管理 參考指南》中的更新 AWS 帳戶的主要聯絡人和更新 AWS 帳戶的替代聯絡人。

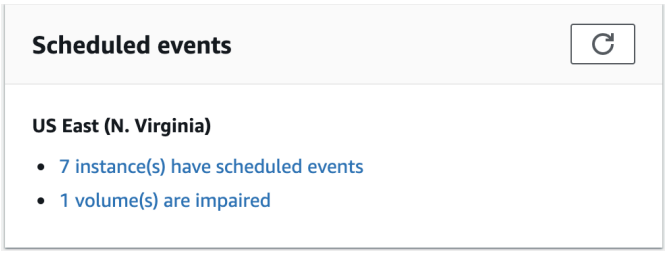
檢查您的執行個體

若您使用的電子郵件帳戶未定期檢查，可能會錯過執行個體淘汰通知。您隨時可以檢查是否有任何執行個體已排定淘汰。

Console

確定排程淘汰的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 EC2 Dashboard (EC2 儀表板)。在排程事件下，您可以看到與 Amazon EC2 執行個體和磁碟區相關聯的事件，依區域組織。



3. 如果您有一個已列出排程事件的執行個體，請在區域名稱下方選取其連結以前往 Events (事件) 頁面。
4. 事件頁面列出所有資源，以及與其相關聯的事件。若要檢視已排定淘汰的執行個體，請從第一個篩選條件清單中選取 Instance resources (執行個體資源)，然後從第二個篩選條件清單中選取 Instance stop or retirement (停止或淘汰執行個體)。
5. 如果篩選條件結果顯示執行個體已排定淘汰，請將其選取，然後記下詳細資訊窗格中 Start time (開始時間) 欄位中的日期和時間。這是您的執行個體淘汰日期。

AWS CLI

找到排程淘汰的執行個體

使用下列 describe-instance-status 命令。在您有執行中執行個體的每個區域重複執行。

```
aws ec2 describe-instance-status --filters Name=event.code,Values=instance-
retirement
```

PowerShell

找到排程淘汰的執行個體

使用 Get-EC2InstanceStatus cmdlet。在您有執行中執行個體的每個區域重複執行。

```
Get-EC2InstanceStatus -Filter @{Name="event.code"; Values="instance-retirement"}
```

針對已排程淘汰的 EBS 後端執行個體採取的動作

若要保留已淘汰執行個體上的資料，您可以執行下列其中一個動作。在執行個體淘汰日期之前採取此動作非常重要，以防止不可預期的停機時間和資料遺失。

對於 Linux 執行個體，如果您不確定您的執行個體是否為 EBS 或執行個體儲存體的後端，請參閱 Amazon EC2 執行個體的根磁碟區。

檢查您的執行個體是否可存取

當您收到執行個體已排程停用的通知時，我們建議您儘快採取下列動作：

- 透過連線或 ping 您的執行個體來檢查您的執行個體是否可以存取。
- 如果可取得您的執行個體，您應該計劃在排程的停止日期之前的適當時間停止/啟動執行個體，這影響最小。如需停止和啟動執行個體的詳細資訊，以及執行個體停止時會發生何事，例如與該執行個體相關聯的公有、私有和彈性 IP 地址的影響，請參閱停止和啟動 Amazon EC2 執行個體。請注意，當您停止並啟動執行個體時，執行個體存放區磁碟區上的資料會遺失。
- 如果您的執行個體無法連線，您應立即採取動作並執行停止/啟動來恢復您的執行個體。
- 或者，如果您想要終止執行個體，請計劃盡快執行個體，以便停止對執行個體產生費用。

為您的執行個體建立備份

從您的執行個體建立 EBS 後端 AMI，以便有備份。若要確保資料完整性，請在建立 AMI 之前停止執行個體。您可等候預定的淘汰日期 – 當執行個體停止時 – 或在淘汰日期之前自行停止執行個體。您可以隨時再次啟動該執行個體。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

啟動替代執行個體

從執行個體建立 AMI 之後，您就可以使用 AMI 來啟動替代用的執行個體。從 Amazon EC2 主控台中，選取新 AMI，然後選擇從 AMI 啟動執行個體。設定執行個體的參數，然後選擇啟動執行個體。如需每個欄位的詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

針對排程淘汰的執行個體存放區支援執行個體採取的動作

若要保留已淘汰執行個體上的資料，您可以執行下列其中一個動作。在執行個體淘汰日期之前採取此動作非常重要，以防止不可預期的停機時間和資料遺失。

> **Warning**
>
> 如果您的執行個體擁有執行個體儲存體根磁碟區且超其淘汰日期，則將會終止，且您無法復原該執行個體或儲存在其中的任何資料。無論執行個體的根磁碟區類型是哪種，執行個體儲存體磁碟區上的資料在執行個體淘汰時都會遺失，即使磁碟區已連接至具有 EBS 根磁碟區的執行個體。

檢查您的執行個體是否可存取

當您收到執行個體已排程停用的通知時，我們建議您儘快採取下列動作：

- 透過連線或 ping 您的執行個體來檢查您的執行個體是否可以存取。
- 如果您的執行個體無法連線，則復原執行個體的可能性很少。如需詳細資訊，請參閱 對無法連線的 Amazon EC2 執行個體進行疑難排解。 AWS 將在排定的淘汰日期終止執行個體，因此，對於無法連線的執行個體，您可以自行立即終止執行個體。

啟動替代執行個體

使用 AMI 工具從您的執行個體建立 Amazon S3 支援的 AMI，如 建立 Amazon S3 支援的 AMI 所述。從 Amazon EC2 主控台中，選取新 AMI，然後選擇從 AMI 啟動執行個體。設定執行個體的參數，然後選擇啟動執行個體。如需每個欄位的詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

將您的執行個體轉換為 EBS 支援的執行個體

將您的資料傳輸到 EBS 磁碟區、建立磁碟區快照，然後從該快照建立 AMI。您可以從新的 AMI 啟動替代用的執行個體。如需詳細資訊，請參閱將 Amazon S3 支援的 AMI 轉換為 EBS 支援的 AMI。

自動執行個體復原功能

> **Important**
>
> 本節描述如何在 EC2 執行個體上主動設定復原機制。當 AWS 偵測到導致系統狀態檢查失敗的基礎硬體或軟體問題時，這些復原機制可用於還原執行個體可用性。如果您目前在存取執行個體時遇到問題，請參閱對 EC2 執行個體進行疑難排解。

如果 AWS 偵測到執行個體因基礎硬體或軟體問題而無法使用，有兩種機制可以自動還原執行個體可用性：簡化的自動復原和 Amazon CloudWatch 動作型復原。還原執行個體可用性亦稱為執行個體復原。

在執行個體復原程序期間， AWS 會嘗試將執行個體從具有基礎硬體或軟體問題的主機移至不同的主機。如果成功，執行個體復原程序為非計劃的重新啟動。您可驗證執行個體復原是否已發生。

若復原程序未成功，執行個體可能會在存在基礎硬體或軟體問題的主機上繼續執行。在此情形下，需手動介入處理。如果執行個體無法連線，或系統狀態檢查持續未通過，我們建議您以手動方式，來停止並啟動執行個體。當您啟動執行個體時，其通常會移轉至新的基礎主機電腦。不過，與自動執行個體復原 (執行個體會保留其公有 IPv4 位址) 不同，重新啟動的執行個體會獲得新的公有 IPv4 位址，除非其設定了彈性 IP 位址。

若要享受自動復原機制的益處，必須在系統狀態檢查失敗前，預先在執行個體上設定這些機制。您啟動執行個體期間，預設會啟用簡化的自動復原。您可在啟動後選擇性地設定基於 Amazon CloudWatch 動作的復原。設定其中一個機制可提升您執行個體的彈性。

簡化自動復原和基於 Amazon CloudWatch 動作的復原僅適用於支援的執行個體。如需詳細資訊，請參閱啟用簡化自動復原的需求及基於 CloudWatch 動作的復原的需求。

> **Warning**
>
> 當 因基礎硬體或軟體問題而 AWS 復原執行個體時，請注意下列後果：儲存在揮發性記憶體 (RAM) 中的資料將會遺失，而且作業系統的執行時間會從零開始。此外，透過基於 CloudWatch 動作的復原，執行個體儲存體磁碟區上的資料也會遺失。為協助防範資料遺失，建議您定期建立重要資料的備份。如需有關 EC2 執行個體備份與復原最佳實務的詳細資訊，請參閱 Amazon EC2 的最佳實務。
>
> 自動執行個體復原機制是為個別執行個體設計的。如需有關建置彈性系統的指引，請參閱 建置彈性系統。

主題

- 自動執行個體復原的核心概念
- 簡化自動復原與基於 CloudWatch 動作的復原之間的差異
- 建置彈性系統
- 驗證是否發生自動執行個體復原
- 在 Amazon EC2 執行個體上設定簡化的自動復原
- 在 EC2 執行個體上設定基於 CloudWatch 動作的復原

自動執行個體復原的核心概念

自動執行個體復原是 Amazon EC2 的一項功能，可在發生基礎硬體或軟體故障時自動還原執行個體可用性，提升 EC2 執行個體的彈性和可靠性。

以下是自動執行個體復原的核心概念：

組態選項

可設定兩種機制來支援自動執行個體復原：

- 簡化的自動復原：支援的執行個體預設會啟用。

- 基於 CloudWatch 動作的復原：在支援的執行個體上需手動設定。

系統狀態檢查

系統狀態檢查會自動監控 EC2 執行個體執行所在的 AWS 基礎設施。

- 如果系統狀態檢查失敗， 會 AWS 啟動自動執行個體復原，嘗試將受影響的執行個體遷移到不同的硬體。
- 系統狀態檢查失敗表示主機的硬體或軟體存在問題，而非執行個體本身的問題。自動執行個體復原可復原系統狀態檢查失敗的執行個體。但若只有執行個體狀態檢查失敗，則自動執行個體復原不會運作。
- 有關執行個體和系統狀態檢查之間的差異，請參閱狀態檢查的類型。

基礎硬體或軟體問題的範例

可能會導致系統狀態檢查失敗的硬體或軟體問題包括：網路連線中斷、系統供電中斷、實體主機上的軟體問題，以及影響網路可達性的實體主機硬體問題。

復原後執行個體的特性

復原後執行個體與原始執行個體相同，但不包含遺失的元素。

保留的元素：

- 執行個體 ID
- 公有、私有和彈性 IP 位址
- 執行個體中繼資料
- 置放群組
- 已連接 EBS 磁碟區
- 可用區域

遺失的元素：

- 儲存於揮發性記憶體 (RAM) 中的資料
- 儲存於執行個體儲存體磁碟區的資料 (僅適用於基於 CloudWatch 動作的復原)
- 作業系統運行時間重設為零

使用 CloudWatch 監控系統狀態檢查

CloudWatch 中的 StatusCheckFailed_System 指標會指出，系統狀態檢查是否通過。

指標值：

- 0 – 系統狀態檢查通過。

- 1 – 系統狀態檢查失敗。

中的事件 Health 儀板表

在自動執行個體復原嘗試期間， Health 儀板表 會根據設定的復原機制及其結果，將事件 AWS 傳送至您的 ：

- 簡化的自動復原
  - 成功事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS
  - 失敗事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE
- 基於 CloudWatch 動作的復原
  - 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
  - 失敗事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

簡化自動復原與基於 CloudWatch 動作的復原之間的差異

下表比較簡化自動復原與基於 CloudWatch 動作的復原之間的核心差異。

比較項目	簡化的自動復原	基於 CloudWatch 動作的復原
組態	預設在支援的執行個體上啟用	需要手動設定 CloudWatch 警示與動作
彈性	修正 管理的復原行為 AWS	可自訂的動作與條件
通知	透過 的基本通知 Health 儀板表	透過 SNS 可自訂通知
裸機執行個體大小	已排除	包含
執行個體儲存體磁碟區只會在啟動時連接	不支援在啟動時連接執行個體儲存體磁碟區的執行個體	在所選取執行個體類型上受支援。請注意，執行個體復原期間，執行個體儲存體磁碟區上的資料會遺失。
復原時間	標準復原嘗試	復原嘗試比簡化自動復原更快
主機問題會在移轉期間解決	移轉可能會取消，執行個體會保留在原始主機上	移轉可能會取消，執行個體會保留在原始主機上

比較項目	簡化的自動復原	基於 CloudWatch 動作的復原
成本	無額外費用	可能產生 CloudWatch 費用

建置彈性系統

雖然簡化的自動復原和 CloudWatch 動作型復原可有效維持個別執行個體的可用性，但 AWS 建議實作高可用性架構，允許流量容錯移轉至運作狀態良好的執行個體。

若要達成此目的，請考慮使用 Elastic Load Balancing（將傳入流量分配到多個 EC2 執行個體）和 Amazon EC2 Auto Scaling（根據需求和運作狀態自動調整執行個體數量）等 AWS 服務。

有關使用 EC2 執行個體建置具備彈性、容錯系統的詳細資訊，請參閱下列資源：

- 在 AWS YouTube 頻道上的返回基本概念：使用 EC2 設計故障
- 架構AWS 部落格網站上的 雲端中的災難復原 (DR) 架構 AWS，第 I 部分：復原策略
- Application Load Balancer 使用者指南
- 《Amazon EC2 Auto Scaling 使用者指南》
- REL11-BP02 容錯移轉至可靠性支柱 Well-Architected Framework 中的運作狀態良好的資源 AWS

驗證是否發生自動執行個體復原

若您的執行個體似乎已離線，然後意外重新啟動，則可能已進行自動執行個體復原，以回應基礎硬體或軟體問題。您可透過檢查 Health 儀板表中的自動執行個體復原事件進行驗證。您也可透過檢查 StatusCheckFailed_System Amazon CloudWatch 指標，確認執行個體是否偵測到基礎硬體或軟體問題。

檢查 中的事件 Health 儀板表

發生自動執行個體復原嘗試時， 會將事件 AWS 傳送至您的 Health 儀板表。具體事件取決於設定的復原機制，以及嘗試是否成功。

在 中檢查自動執行個體復原事件 Health 儀板表

1. 在 Health 儀板表 https：//https://phd.aws.amazon.com/phd/home#/ 開啟 。
2. 尋找與自動執行個體復原關聯的事件。這些事件的存在可確認是否發生自動執行個體復原的嘗試及其相應結果。

- 簡化的自動復原
  - 成功事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS
  - 失敗事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE
- 基於 CloudWatch 動作的復原
  - 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
  - 失敗事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

透過 CloudWatch 監控系統狀態檢查

您可透過檢查 CloudWatch 中的 StatusCheckFailed_System 指標，以驗證執行個體是否偵測到基礎硬體或軟體問題。指標值指出系統狀態檢查是通過 (沒有硬體或軟體問題) 還是失敗 (硬體或軟體問題)。

驗證是否偵測到基礎硬體或軟體問題

1. 透過 https://console.aws.amazon.com/cloudwatch/home?#metricsV2 開啟 CloudWatch 主控台指標頁面。
2. 確認您位於 EC2 執行個體所在的同一區域。
3. 在指標搜尋欄位中貼上下列指標，然後按 Enter 鍵。

   ```
   StatusCheckFailed_System
   ```

4. 選擇 EC2 > 每個執行個體指標。
5. 在表格中，選取要檢查的執行個體旁的核取方塊。
6. 將查詢期間變更為您懷疑發生了復原事件的時間。
7. 選擇圖形化指標索引標籤，並針對 StatusCheckFailed_System，請執行下列動作：
   a. 對於統計資料，選擇平均值、最大值或最小值。
   b. 對於期間，選擇 1 分鐘。
8. 檢查 StatusCheckFailed_System 的數值。
   - 數值為 0：系統狀態檢查通過，表示沒有基礎硬體或軟體問題。
   - 數值為 1：系統狀態檢查失敗，表示存在基礎硬體或軟體問題。

如需詳細資訊，請參閱自動執行個體復原功能。

在 Amazon EC2 執行個體上設定簡化的自動復原

> **Important**
>
> 本節描述如何在 EC2 執行個體上主動設定復原機制。當 AWS 偵測到導致系統狀態檢查失敗的基礎硬體或軟體問題時，這些復原機制可用於還原執行個體可用性。如果您目前在存取執行個體時遇到問題，請參閱對 EC2 執行個體進行疑難排解。

如果 AWS 偵測到執行個體因基礎硬體或軟體問題而無法使用，簡化的自動復原可以透過將具有基礎問題的執行個體從主機移至不同的主機，自動還原執行個體可用性。

如果發生簡化的自動復原， 會根據結果 AWS Health 儀板表， AWS 將下列其中一個事件傳送至您的：

- 成功事件：`AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
- 失敗事件：`AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`

若要接收這些事件的通知，您可以設定通知功能。如需詳細資訊，請參閱《AWS 使用者通知 使用者指南》中的在 AWS 使用者通知中建立您的第一個通知組態。您還可以使用 Amazon EventBridge 規則，利用以下事件代碼監控簡化的自動復原事件：

啟動執行個體期間，預設會在所有支援的執行個體上啟用簡化的自動復原。不過，只有在執行個體處於 `running` 狀態、 中未列出任何服務事件，且執行個體類型有可用的容量時 AWS Health 儀板表，才能運作。某些情況下，例如重大中斷，容量限制可能會導致復原嘗試失敗。如需詳細資訊，請參閱the section called “對簡化的自動復原失敗進行疑難排解”。

您可在啟動期間或之後停用簡化的自動復原，並在需要時重新啟用。

> **Warning**
>
> 當 因基礎硬體或軟體問題而 AWS 復原執行個體時，請注意下列後果：儲存在揮發性記憶體 (RAM) 中的資料將會遺失，而且作業系統的執行時間會從零開始。為協助防範資料遺失，建議您定期建立重要資料的備份。如需有關 EC2 執行個體備份與復原最佳實務的詳細資訊，請參閱 Amazon EC2 的最佳實務。
>
> 自動執行個體復原機制是為個別執行個體設計的。如需有關建置彈性系統的指引，請參閱 建置彈性系統。

目錄

- 啟用簡化自動復原的需求
- 設定簡化的自動復原
- 對簡化的自動復原失敗進行疑難排解

啟用簡化自動復原的需求

可在符合下列條件的執行個體上啟用簡化自動復原：

執行個體類型

- 一般用
  途：A1、M3、M4、M5、M5a、M5n、M5zn、M6a、M6g、M6i、M6in、M7a、M7g、M7i、M7i-flex、M8a、M8azn、M8g、M8gb、M8gn、M8i、M8i-flex、M8in、M8ine、M8ib、M9g、T1、T2、T3、T3a、T4g
- 運算最佳
  化：C3、C4、C5、C5a、C5n、C6a、C6g、C6gn、C6i、C6in、C7a、C7g、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gn、C8i、C8i-flex、C8in、C8ine、C8ib、C9g
- 記憶體最佳
  化：R3、R4、R5、R5a、R5b、R5n、R6a、R6g、R6i、R6in、R7a、R7g、R7i、R7iz、R8a、R8g、R8g-flex、R8in、R8ib、R9g、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb1、U7i-6tb、U7i-12tb、U7inh-32tb X1 X1e X2iezn X8g X8i
- 加速運算：G3、G5g、Inf1、P3、VT1
- 高效能運算：Hpc6a、Hpc7a、Hpc7g、Hpc8a

租用

- 共同
- Dedicated Instance

  如需詳細資訊，請參閱Amazon EC2 專用執行個體。

限制

如果執行個體具有以下特性，則此執行個體不支援簡化的自動復原程序：

- 執行個體大小：`metal` 執行個體

- 租用：專用主機。對於專用主機，改用專用主機自動復原。
- 儲存：具有執行個體儲存體磁碟區的執行個體
- 聯網：使用 Elastic Fabric Adapter 的執行個體
- 自動擴展：屬於 Auto Scaling 群組的執行個體
- 維護：目前正在進行排程維護事件的執行個體

設定簡化的自動復原

當您啟動支援的執行個體時，預設會啟用簡化的自動復原。您可在啟動執行個體期間或之後將自動復原行為設定為 `disabled`。

`default` 組態僅會針對支援的執行個體啟用簡化的自動復原。

Console

啟動時停用簡化的自動復原程序

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選擇 Launch Instance (啟動執行個體)。
3. 在進階詳細資訊區段，針對執行個體自動復原，選擇已停用。
4. 根據需要進行剩餘的執行個體啟動設定，然後啟動執行個體。

啟動後停用簡化的自動復原程序

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 動作、執行個體設定、變更自動復原行為。
4. 選擇 Off (關閉)，然後選擇 Save (儲存)。

在啟動後啟用簡化自動復原

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選取 動作、執行個體設定、變更自動復原行為。
4. 選擇預設，然後選擇儲存。

AWS CLI

啟動時停用簡化的自動復原程序

請使用 `--maintenance-options` 選項並搭配 run-instances 命令。

```
--maintenance-options AutoRecovery=Disabled
```

啟動後停用簡化的自動復原程序

使用 modify-instance-maintenance-options 命令。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery disabled
```

在啟動後啟用簡化自動復原

使用 modify-instance-maintenance-options 命令。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery default
```

PowerShell

啟動時停用簡化的自動復原程序

使用 New-EC2Instance cmdlet。

```
-MaintenanceOptions_AutoRecovery Disabled
```

啟動後停用簡化的自動復原程序

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Disabled
```

在啟動後啟用簡化自動復原

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Enabled
```

對簡化的自動復原失敗進行疑難排解

若簡化自動復原未能復原您的執行個體，請考量下列問題：

- AWS 服務事件正在執行

  簡化的自動復原不會在 AWS Health 儀板表中的服務事件期間運作。您可能不會收到此類事件的復原故障通知。如需最新的服務可用性資訊，請參閱服務運作狀態頁面。

- 容量不足

  暫時沒有足夠的替換硬體可用於移轉執行個體。

- 已達每日復原嘗試次數上限

  執行個體已達到每日復原嘗試的最高限額。如果自動復原失敗，而原始系統狀態檢查未通過的原因判定為硬體劣化，則執行個體可能隨後會遭到淘汰。

如果執行個體系統狀態檢查在多次復原嘗試後持續未通過，請參閱對狀態檢查失敗的執行個體進行疑難排解以取得其他指引。

在 EC2 執行個體上設定基於 CloudWatch 動作的復原

> **Important**
>
> 本節描述如何在 EC2 執行個體上主動設定復原機制。當 AWS 偵測到導致系統狀態檢查失敗的基礎硬體或軟體問題時，這些復原機制可用於還原執行個體可用性。如果您目前在存取執行個體時遇到問題，請參閱對 EC2 執行個體進行疑難排解。

如果 AWS 偵測到執行個體因基礎硬體或軟體問題而無法使用，CloudWatch 動作型復原可以透過將具有基礎問題的執行個體從主機移至不同的主機，自動還原執行個體可用性。

如果發生 CloudWatch 動作型復原， 會根據結果 AWS Health 儀板表將下列其中一個事件 AWS 傳送至您的 ：

- 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
- 失敗事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

您可以設定基於 CloudWatch 動作的復原，將復原動作新增至 Amazon CloudWatch 警示。CloudWatch 動作型復原使用 StatusCheckFailed_System 指標。CloudWatch 動作型復原提供即時復原回應時間的精細程度，以及復原動作和結果的 Amazon Simple Notification Service (Amazon SNS) 通知。與簡化的自動復原相比，這些組態選項允許更快速的復原嘗試，並且更精密地控制系統狀態的檢查失敗事件回應。如需可用 CloudWatch 選項的詳細資訊，請參閱執行個體的狀態檢查。

不過，只有在執行個體處於 running 狀態、 中未列出任何服務事件，且執行個體類型有可用的容量時 AWS Health 儀板表，CloudWatch 動作型復原才能運作。某些情況下，例如重大中斷，容量限制可能會導致復原嘗試失敗。如需詳細資訊，請參閱the section called “疑難排解”。

> **Warning**
>
> 當 因基礎硬體或軟體問題 AWS 而復原執行個體時，請注意下列後果：儲存在揮發性記憶體 (RAM) 和執行個體儲存體磁碟區上的資料將會遺失，而且作業系統的執行時間會從零開始。為協助防範資料遺失，建議您定期建立重要資料的備份。如需有關 EC2 執行個體備份與復原最佳實務的詳細資訊，請參閱 Amazon EC2 的最佳實務。
> 自動執行個體復原機制是為個別執行個體設計的。如需有關建置彈性系統的指引，請參閱 建置彈性系統。

目錄

- 基於 CloudWatch 動作的復原的需求
- 設定 CloudWatch 動作型復原
- 對基於 CloudWatch 動作的復原失敗進行疑難排解

基於 CloudWatch 動作的復原的需求

可在符合下列條件的執行個體上啟用基於 CloudWatch 動作的復原：

執行個體類型

- 一般用途：A1、M3、M4、M5、M5a、M5n、M5zn、M6a、M6g、M6i、M6in、M7a、M7g、M7i、M7i-

flex、M8a、M8azn、M8g、M8gb、M8gn、M8i、M8i-flex、M8in、M8ine、M8ib、M9g、T1、T2、T3、T3a、T4g

- 運算最佳化：C3、C4、C5、C5a、C5n、C6a、C6g、C6gn、C6i、C6in、C7a、C7g、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gn、C8i、C8i-flex、C8in、C8ine、C8ib、C9g
- 記憶體最佳化：R3、R4、R5、R5a、R5b、R5n、R6a、R6g、R6i、R6in、R7a、R7g、R7i、R7iz、R8a、R8g、R8g-flex、R8in、R8ib、R9g、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb1、U7i-6tb、U7i-12tb、U7in-32tb U7inh-32tb X1 X1e X2idn X2iedn X2iezn X8g X8i
- 加速運算：G3、G5g、Inf1、P3、VT1
- 高效能運算：Hpc6a、Hpc7a、Hpc7g、Hpc8a
- 裸機執行個體：上述任何具有裸機執行個體大小的執行個體類型。
- 若在啟動時新增執行個體儲存體磁碟區：則僅支援下列執行個體類型：M3、C3、R3、X1、X1e、X2idn、X2iedn

租用

- 共同
- Dedicated Instance

如需詳細資訊，請參閱Amazon EC2 專用執行個體。

限制

具有下列特性的執行個體不支援基於 CloudWatch 動作的復原：

- 租用：專用主機。對於專用主機，改用專用主機自動復原。
- 聯網：使用 Elastic Fabric Adapter 的執行個體
- 自動擴展：屬於 Auto Scaling 群組的執行個體
- 維護：目前正在進行排程維護事件的執行個體

尋找支援的執行個體類型

您可以檢視支援基於 CloudWatch 動作的復原的執行個體類型。

Console

檢視支援基於 CloudWatch 動作的復原的執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 Instance Types (執行個體類型)。
3. 在篩選條件列中，新增篩選條件支援自動復原程序：是。執行個體類型表格會顯示支援基於 CloudWatch 動作的復原的所有執行個體類型。
4. (選用) 新增篩選條件以進一步限定到感興趣的特定執行個體類型。

AWS CLI

檢視支援基於 CloudWatch 動作的復原的執行個體類型

使用 describe-instance-types 命令並搭配 `auto-recovery-supported` 篩選條件。

```
aws ec2 describe-instance-types \
    --filters Name=auto-recovery-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

檢視支援基於 CloudWatch 動作的復原的執行個體類型

使用 Get-EC2InstanceType cmdlet 搭配 `auto-recovery-supported` 篩選條件。

```
Get-EC2InstanceType `
    -Filter @{Name="auto-recovery-supported";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

設定 CloudWatch 動作型復原

若要為 EC2 執行個體設定基於 CloudWatch 動作的復原，請建立 CloudWatch 警示，用於監控指定執行個體的 `StatusCheckFailed_System` 指標。將警示設定為在指標值為 1 時觸發，此數值表示系統狀態檢查失敗。設定警示動作，使其在觸發時自動復原執行個體。

您可以使用 Amazon EC2 主控台或 CloudWatch 主控台設定警示。有關說明，請參閱本使用者指南中的 將復原動作新增至 Amazon CloudWatch 警示，或《Amazon CloudWatch 使用者指南》中的將復原動作新增至 Amazon CloudWatch 警示。

對基於 CloudWatch 動作的復原失敗進行疑難排解

若基於 CloudWatch 動作的復原未能復原您的執行個體，請考量下列問題：

- AWS 服務事件正在執行

  CloudWatch 動作型復原不會在 AWS Health 儀板表的服務事件期間運作。您可能不會收到此類事件的復原故障通知。如需最新的服務可用性資訊，請參閱服務運作狀態頁面。
- 容量不足

  暫時沒有足夠的替換硬體可用於移轉執行個體。
- 已達每日復原嘗試次數上限

  執行個體已達到每日復原嘗試的最高限額。如果自動復原失敗，而原始系統狀態檢查未通過的原因判定為硬體劣化，則執行個體可能隨後會遭到淘汰。

如果執行個體系統狀態檢查在多次復原嘗試後持續未通過，請參閱對狀態檢查失敗的執行個體進行疑難排解以取得其他指引。

使用執行個體中繼資料管理您的 EC2 執行個體

執行個體中繼資料 是關於您執行個體的資料，您可以用來設定或管理執行中的執行個體。執行個體中繼資料包括下列項目：

執行個體中繼資料屬性

執行個體中繼資料屬性會分為幾個類別，例如主機名稱、事件及安全群組。

動態資料

動態資料為執行個體啟動時產生的中繼資料，例如執行個體身分文件。如需詳細資訊，請參閱動態資料分類。

使用者資料

您也可以使用執行個體中繼資料，存取您在啟動執行個體時指定的使用者資料。例如，您可以指定設定您執行個體的參數，或是包含簡單的指令碼。您也可以建立一般 AMI 並使用使用者資料，修

改啟動時提供的組態檔案。例如，如果您執行不同小型企業的 Web 伺服器，它們全部都可以使用相同的一般 AMI，並從您在啟動時於使用者資料中指定的 Amazon S3 儲存貯體擷取其內容。若要隨時新增客戶，為客戶建立儲存貯體、新增其內容，以及利用使用者資料中提供給程式碼的唯一儲存貯體名稱來啟動 AMI。如果您使用相同的 `RunInstances` 呼叫啟動多個執行個體，則該次保留的所有執行個體均可取得使用者資料。屬於相同保留的每個執行個體都有唯一的 `ami-launch-index` 號碼，可讓您寫入能控制執行個體作業的程式碼。例如，第一個主機可能會選取本身做為叢集中的原始節點。如需詳細的 AMI 啟動範例，請參閱 識別在單一請求中啟動的每個執行個體。

Important

雖然您只能在執行個體內部存取執行個體中繼資料和使用者資料，資料並未受到驗證或密碼編譯法保護。可直接存取執行個體的任何人，以及可能在該執行個體上執行的任何軟體，都能檢視其中繼資料。因此，您不應該將敏感性資料 (例如密碼或長期加密金鑰) 儲存為使用者資料。

目錄

- 執行個體中繼資料分類
- 動態資料分類
- 存取 EC2 執行個體的執行個體中繼資料
- 設定執行個體中繼資料服務選項
- 當使用使用者資料輸入啟動 EC2 執行個體時執行命令
- 識別在單一請求中啟動的每個執行個體

執行個體中繼資料分類

執行個體中繼資料屬性分為數種類別。若要擷取執行個體中繼資料屬性，請在請求中指定類別，回應中即會傳回中繼資料。

發佈新類別時，將會使用新版本編號建立新的執行個體中繼資料建置。在下表中，Version when category was released (類別發佈時的版本) 資料欄指定了發佈執行個體中繼資料類別時的建置版本。為避免每次 Amazon EC2 發佈新的執行個體中繼資料建置時都必須更新您的程式碼，請在中繼資料請求中使用 `latest`，而不是版本編號。如需詳細資訊，請參閱 取得執行個體中繼資料的可用版本。

Amazon EC2 發行新的執行個體中繼資料類別時，新類別的執行個體中繼資料可能無法用於現有的執行個體。使用 Nitro 型執行個體時，您只能擷取啟動時可用類別的執行個體中繼資料。對於使用 Xen hypervisor 的執行個體，您可以停止然後啟動執行個體，以更新執行個體可用的類別。

下表列出執行個體中繼資料的分類。下表中的某些類別名稱包含對您的執行個體而言唯一的資料預留位置。例如，*mac* 代表網路介面的 MAC 地址。擷取執行個體中繼資料時，必須使用實際的值取代預留位置。

類別	描述	類別發佈時的版本
ami-id	用於啟動執行個體的 AMI ID。	1.0
ami-launch-index	如果您使用相同的 RunInstances 呼叫啟動多個執行個體，此值會指出每個執行個體的啟動順序。第一個啟動的執行個體值為 0。如果您使用自動擴展或 EC2 機群啟動執行個體，此值一律為 0。	1.0
ami-manifest-path	指向 Amazon S3 中 AMI 資訊清單檔案的路徑。若您使用 Amazon EBS 後端 AMI 啟動執行個體，傳回的結果將會是 unknown。	1.0
ancestor-ami-ids	任何為了建立此 AMI 重新組合之執行個體的 AMI ID。此值只有在 AMI 資訊清單檔案包含 ancestor-amis 鍵時才會存在。	2007-10-10
autoscaling/target-lifecycle-state	顯示 Auto Scaling 執行個體正在轉換到的目標 Auto Scaling 生命週期狀態的值。當執行個體在 2022 年 3 月 10 日之後轉換到目標生命週期狀態之一時存在。可能的值如下：Detached | InService | Standby | Terminated | Warmed:Hibernated | Warmed:Running | Warmed:Stopped | Warmed:Terminated 。請參閱「Amazon EC2 Auto Scaling 使用	2021-07-15

類別	描述	類別發佈時的版本
	者指南」中的透過執行個體中繼資料擷取目標生命週期狀態。	
block-device-mapping/ami	包含根/開機檔案系統的虛擬裝置。	2007-12-15
block-device-mapping/ebsN	與任何 Amazon EBS 磁碟區相關聯的虛擬裝置。Amazon EBS 磁碟區只在啟動時或在執行個體上次啟動時存在，才能用於中繼資料。N 表示 Amazon EBS 磁碟區的索引 (例如 ebs1 或 ebs2)。	2007-12-15
block-device-mapping/ephemeral N	用於任何非 NVMe 執行個體存放區磁碟區的虛擬裝置。N 表示每個磁碟區的索引。區塊型設備映射內的執行個體存放區磁碟區數量，可能與執行個體的執行個體存放區磁碟區實際數量不同。執行個體類型決定了執行個體存放區磁碟區可用執行個體的數量。如果區塊型設備映射內執行個體存放區磁碟區的數量，超過可用執行個體，請忽略額外的執行個體存放區磁碟區。	2007-12-15
block-device-mapping/root	與根設備相關聯的虛擬裝置或分割區，或虛擬裝置上的分割區，其中根 (/ 或 C:) 檔案系統與指定的執行個體相關聯。	2007-12-15
block-device-mapping/swap	與 swap 相關聯的虛擬裝置。並非總是存在。	2007-12-15
events/maintenance/history	如果執行個體有已完成或已取消的維護事件，請讓 JSON 字串包含事件的相關資訊。	2018-08-17

類別	描述	類別發佈時的版本
events/maintenance/scheduled	如果執行個體有使用中的維護事件，請讓 JSON 字串包含事件的相關資訊。如需詳細資訊，請參閱 檢視會影響 Amazon EC2 執行個體的排程事件。	2018-08-17
events/recommendations/rebalance	針對執行個體發出 EC2 執行個體重新平衡建議通知的大約時間 (UTC)。以下是此類別中繼資料的範例：{"noticeTime": "2020-11-05T08:22:00Z"} 。此類別只有在發出通知後才可用。如需詳細資訊，請參閱EC2 執行個體重新平衡建議。	2020-10-27
hostname	如果 EC2 執行個體使用的是 IP 型命名 (IPBN)，則這是執行個體的私有 IPv4 DNS 主機名稱。如果 EC2 執行個體使用的是資源型命名 (RBN)，則這是 RBN。若有多個網路介面，則其代表 eth0 裝置 (裝置編號為 0 的裝置)。如需 IPBN 和 RBN 的詳細資訊，請參閱 EC2 執行個體主機名稱與網域。	1.0
iam/info	若有任何與執行個體相關聯的 IAM 角色，則包含最近一次執行個體描述檔更新的時間，其中又包含執行個體的 LastUpdated 日期、InstanceProfileArn 和 InstanceProfileId。否則不會存在。	2012-01-12

類別	描述	類別發佈時的版本
iam/security-credentials/role-name	若有和執行個體相關聯的 IAM 角色，*role-name* 即為角色的名稱，且 *role-name* 包含與角色相關聯的暫時安全登入資料 (如需詳細資訊，請參閱從執行個體中繼資料中擷取安全登入資料)。否則不會存在。	2012-01-12
identity-credentials/ec2/info	identity-credentials/ec2/security-credentials/ec2-instance 中有關憑證的資訊。	2018-05-23
identity-credentials/ec2/security-credentials/ec2-instance	執行個體身分角色的登入資料，允許執行個體上的軟體向 自我識別 AWS ，以支援 EC2 Instance Connect 和 AWS Systems Manager 預設主機管理組態等功能。這些登入資料未連接任何政策，因此除了識別 AWS 該功能的執行個體之外，他們沒有額外的 AWS API 許可。如需詳細資訊，請參閱Amazon EC2 執行個體的執行個體身分角色。	2018-05-23
instance-action	通知執行個體應重新開機以準備綁定。有效值：none | shutdown | bundle-pending 。	2008-09-01
instance-id	此執行個體的 ID。	1.0
instance-life-cycle	此執行個體的購買選項。如需詳細資訊，請參閱Amazon EC2 帳單與購買選項。	2019-10-01

類別	描述	類別發佈時的版本
instance-type	執行個體的類型。如需詳細資訊，請參閱Amazon EC2 執行個體類型。	2007-08-29
ipv6	執行個體的 IPv6 地址。若有多個網路介面，則其代表 eth0 裝置 (裝置編號為 0 的裝置) 網路介面，以及第一個指派的 IPv6 地址。如果網路介面 [0] 上沒有 IPv6 地址，則不會設定此項目，並會產生 HTTP 404 回應。	2021-01-03
kernel-id	使用此執行個體啟動的核心 ID (若適用的話)。	2008-02-01
local-hostname	若有多個網路介面，則其代表 eth0 裝置 (裝置編號為 0 的裝置)。如果 EC2 執行個體使用的是 IP 型命名 (IPBN)，則這是執行個體的私有 IPv4 DNS 主機名稱。如果 EC2 執行個體使用的是資源型命名 (RBN)，則這是 RBN。如需 IPBN、RBN 和 EC2 執行個體命名的詳細資訊，請參閱 EC2 執行個體主機名稱與網域。	2007-01-19
local-ipv4	執行個體的私有 IPv4 地址。若有多個網路介面，則其代表 eth0 裝置 (裝置編號為 0 的裝置)。如果這是僅限 IPv6 執行個體，則不會設定此項目，並會產生 HTTP 404 回應。	1.0
mac	執行個體的媒體存取控制 (MAC) 地址。若有多個網路介面，則其代表 eth0 裝置 (裝置編號為 0 的裝置)。	2011-01-01

類別	描述	類別發佈時的版本
metrics/vhostmd	不再提供使用。	2011-05-01
network/interfaces/macs/mac/device-number	與該執行個體相關聯的唯一裝置編號。裝置編號對應至裝置名稱；例如，device-number 為 2 即為 eth2 裝置。此類別對應到 Amazon EC2 API 和 AWS CLI的 EC2 命令使用的 DeviceIndex 和 device-index 欄位。	2011-01-01
network/interfaces/macs/mac/interface-id	網路介面的 ID。	2011-01-01
network/interfaces/macs/mac/ipv4-associations/public-ip	與每個公有 IP 地址相關聯並指派給該介面的私有 IPv4 地址。	2011-01-01
network/interfaces/macs/mac/ipv6s	指派給介面的 IPv6 地址。	2016-06-30
network/interfaces/macs/mac/ipv6-prefix	指派給網路介面的 IPv6 字首。	
network/interfaces/macs/mac/local-hostname	網路界面的私有 IPv4 DNS 主機名稱。完全沒有 IP 地址的介面不會設定此值，例如僅限 EFA 的介面。如果這是IPv6-only的界面，則這是以資源為基礎的名稱。如需 IPBN 和 RBN 的詳細資訊，請參閱 EC2 執行個體主機名稱與網域。	2007-01-19
network/interfaces/macs/mac/local-ipv4s	與介面相關聯的私有 IPv4 地址。如果這是僅限 IPv6 網路介面，則不會設定此項目，並會產生 HTTP 404 回應。	2011-01-01

類別	描述	類別發佈時的版本
network/interfaces/macs/mac/mac	執行個體的 MAC 地址。	2011-01-01
network/interfaces/macs/*mac*/network-card	網路卡的索引。某些執行個體類型支援多個網路卡。	2020-11-01
network/interfaces/macs/mac/owner-id	網路介面擁有者的 ID。在多介面環境中，介面僅能由第三方 (例如 Elastic Load Balancing) 連接。介面上的流量一律由介面的擁有者付費。	2011-01-01
network/interfaces/macs/mac/public-hostname	介面的公有 DNS (IPv4)。只在 enableDnsHostnames 屬性設為 true 時才會傳回此類別。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。如果執行個體只有公有 IPv6 地址且沒有公有 IPv4 地址，則不會設定此項目，並會產生 HTTP 404 回應。	2011-01-01
network/interfaces/macs/mac/public-ipv4s	與界面相關聯的公有 IP 地址與彈性 IP 地址。執行個體上可能會有多個 IPv4 地址。	2011-01-01
network/interfaces/macs/mac/security-groups	網路介面所屬的安全群組。	2011-01-01
network/interfaces/macs/mac/security-group-ids	網路介面所屬的安全群組 ID。	2011-01-01
network/interfaces/macs/mac/subnet-id	介面所在的子網 ID。	2011-01-01

類別	描述	類別發佈時的版本
network/interfaces/macs/mac/subnet-ipv4-cidr-block	介面所在子網的 IPv4 CIDR 區塊。	2011-01-01
network/interfaces/macs/mac/subnet-ipv6-cidr-blocks	介面所在子網的 IPv6 CIDR 區塊。	2016-06-30
network/interfaces/macs/mac/vpc-id	介面所在的 VPC ID。	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-block	VPC 的主要 IPv4 CIDR 區塊。	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	VPC 的 IPv4 CIDR 區塊。	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	介面所在 VPC 的 IPv6 CIDR 區塊。	2016-06-30
placement/availability-zone	啟動執行個體的可用區域。	2008-02-01
placement/availability-zone-id	啟動的執行個體所在的靜態可用區域 ID。可用區域 ID 會在各帳戶間保持一致。但是，它可能與可用區域不同，視帳戶而異。	2019-10-01
placement/group-name	在啟動執行個體的位置放置群組名稱。	2020-08-24
placement/host-id	啟動執行個體位置的主機 ID。僅適用於 專用執行個體。	2020-08-24

類別	描述	類別發佈時的版本
placement/partition-number	啟動執行個體位置的分割區編號。	2020-08-24
placement/region	啟動執行個體 AWS 的區域。	2020-08-24
product-codes	AWS Marketplace 與執行個體相關聯的產品代碼，如果有的話。	2007-03-01
public-hostname	執行個體的公有 DNS (IPv4)。只在 enableDnsHostnames 屬性設為 true 時才會傳回此類別。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。如果執行個體只有公有 IPv6 地址且沒有公有 IPv4 地址，則不會設定此項目，並會產生 HTTP 404 回應。	2007-01-19
public-ipv4	公有 IPv4 地址。若彈性 IP 地址與執行個體相關聯，傳回的值即為彈性 IP 地址。	2007-01-19
public-keys/0/openssh-key	公有金鑰。只有在執行個體啟動階段提供時才可用。	1.0
ramdisk-id	啟動階段指定之 RAM 磁碟的 ID (若適用的話)。	2007-10-10
reservation-id	保留 ID。	1.0
security-groups	套用至執行個體的安全群組名稱。 在啟動之後，您可以變更執行個體的安全群組。這類變更除了會在此反映，也會在 network/interfaces/macs/*mac*/security-groups 中反映。	1.0

類別	描述	類別發佈時的版本
services/domain	區域的 AWS 資源網域。	2014-02-25
services/partition	資源所在的分割區。對於標準 AWS 區域，分割區為 aws。如果您有其他分割區的資源，則該分割區為 aws-*partitionname* 。例如，中國 (北京) 區域的資源分割區，即為 aws-cn。	2015-10-20
spot/instance-action	動作 (休眠、停止或終止) 和動作發生的約略時間 (UTC)。只有在 Spot 執行個體已標記為休眠、停止或終止時，才會顯示此項目。如需詳細資訊，請參閱instance-action。	2016-11-15
spot/termination-time	Spot 執行個體的作業系統接收關機訊號的近似時間 (UTC)。此項目只有在 Spot 執行個體已由 Amazon EC2 標記為終止時，才會存在且包含時間值 (例如，2015-01-05T18:02:00Z)。若您自行終止 Spot 執行個體，則 termination-time 項目便不會設為時間。如需詳細資訊，請參閱termination-time。	2014-11-05
system	執行個體的基礎虛擬化類型 (Hypervisor)。	2022-09-24
tags/instance	與執行個體相關聯的執行個體標籤。僅當您明確允許存取執行個體中繼資料中的標籤時才可用。如需詳細資訊，請參閱 在執行個體中繼資料中啟用標籤存取權。	2021-03-23

動態資料分類

下表列出動態資料的分類。

類別	描述	類別發佈時的版本
`fws/instance-monitoring`	顯示客戶是否在 CloudWatch 中啟用詳細一分鐘監控的值。有效值：`enabled | disabled`	2009-04-04
`instance-identity/document`	包含執行個體屬性的 JSON，例如 instance-id、私有 IP 地址等。請參閱 Amazon EC2 執行個體的執行個體身分文件。	2009-04-04
`instance-identity/pkcs7`	用來使用簽章確認文件的真確性和其內容。請參閱 Amazon EC2 執行個體的執行個體身分文件。	2009-04-04
`instance-identity/signature`	可由其他廠商用來確認其來源和真確性的資料。請參閱 Amazon EC2 執行個體的執行個體身分文件。	2009-04-04

存取 EC2 執行個體的執行個體中繼資料

您可以從執行個體本身內部或 EC2 主控台、API、 SDK 或 AWS CLI，存取 EC2 執行個體中繼資料。若要從主控台或命令列取得執行個體目前的執行個體中繼資料設定，請參閱 查詢現有執行個體的執行個體中繼資料選項。

您也可以修改具有 EBS 根磁碟區之執行個體的使用者資料。執行個體必須處於停止狀態。如需主控台操作指示，請參閱更新執行個體使用者資料。如需使用 的 Linux 範例 AWS CLI，請參閱 modify-instance-attribute。如需使用 Tools for Windows PowerShell 的 Windows 範例，請參閱 the section called “使用者資料與 Tools for Windows PowerShell”。

> **Note**
>
> 您無須為用來擷取執行個體中繼資料和使用者資料的 HTTP 請求支付費用。

執行個體中繼資料存取考量

若要避免執行個體中繼資料的問題，請考慮下列事項。

IMDSv2 強制執行導致執行個體啟動失敗 (**`HttpTokensEnforced=enabled`**)

在啟用 IMDSv2 強制執行之前，您需要執行個體上的所有軟體來支援 IMDSv2，之後您可以將預設值變更為停用 IMDSv1 (`httpTokens=required`)，之後即可啟用強制執行。如需詳細資訊，轉換為使用 執行個體中繼資料服務第 2 版。

命令格式

命令格式不同，根據您使用的是執行個體中繼資料服務版本 1 (IMDSv1) 或執行個體中繼資料服務版本 2 (IMDSv2)，命令格式有所不同。根據預設，您可以同時使用兩個版本的執行個體中繼資料服務。欲要求使用 IMDSv2，請參閱 使用執行個體中繼資料服務存取執行個體中繼資料。

如果需要 IMDSv2，則 IMDSv1 將無法運作

如果您使用 IMDSv1 且未收到回應，則可能需要 IMDSv2。若要檢查是否需要 IMDSv2，請選取執行個體以檢視其詳細資訊。IMDSv2 值表示必要 (您必須使用 IMDSv2) 或選用 (您可以使用 IMDSv2 或 IMDSv1)。

(IMDSv2) 使用 /latest/api/token 擷取字符

向任何版本特定的路徑發出 PUT 請求 (例如 `/2021-03-23/api/token`)，將導致中繼資料服務傳回 403 Forbidden 錯誤。這是預期行為。

中繼資料版本

為避免每次 Amazon EC2 發佈新的執行個體中繼資料建置時都必須更新您的程式碼，我們建議您在路徑中使用 `latest`，而不是版本編號。

IPv6 支援

若要使用 IPv6 地址擷取執行個體中繼資料，請確定啟用及使用 IMDS 的 IPv6 地址 `[fd00:ec2::254]`，而不是 IPv4 地址 `169.254.169.254`。執行個體必須是在支援 IPv6 的子網路中啟動的 Nitro 型執行個體。

(Windows) 使用 Windows Sysprep 建立自訂 AMIs

若要在您從自訂 Windows AMI 啟動執行個體確保 IMDS 正常運作，AMI 必須是使用 Windows Sysprep 建立的標準化映像。否則，IMDS 將無法運作。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。

在容器環境中，考慮重新設定或將跳轉限制增加到 2

根據預設 AWS SDKs 會使用 IMDSv2 呼叫。如果 IMDSv2 呼叫未收到回應，則 AWS SDKs會重試呼叫，如果仍然失敗，請使用 IMDSv1。這可能會導致延遲，特別是在容器環境中。對於需要 IMDSv2 AWS SDKs，如果跳轉限制在容器環境中為 1，則呼叫可能完全不會收到回應，因為前往容器會被視為額外的網路跳轉。

若要緩解容器環境中的這些問題，請考慮將組態變更為將設定 （例如 AWS 區域) 直接傳遞至容器，或考慮將跳轉限制增加至 2。如需關於跳轉限制的資訊，請參閱透過 EC2 執行個體中繼資料服務的增強功能，提高開放式防火牆、反向代理伺服器及 SSRF (伺服器端請求偽造) 弱點的防禦能力。如需變更跳轉限制的資訊，請參閱 變更 PUT 回應跳轉限制。

每秒封包數 (PPS) 的限制

使用連結本機位址的服務有每秒 1024 個封包 (PPS) 限制。此限制包括 Route 53 Resolver DNS 查詢、執行個體中繼資料服務 (IMDS) 請求、Amazon Time Service Network Time Protocol (NTP) 請求和 Windows Licensing Service (適用於 Microsoft Windows 型執行個體) 請求的彙總。

使用者資料存取的其他考量

- 使用者資料被視為不透明資料：您指定的內容在擷取時會依原樣傳回。由執行個體自行解讀使用者資料並對其採取行動。
- 使用者資料必須為 base64 編碼形式。根據您使用的工具或 SDK，可能會自動為您執行 Base64 編碼。例如：
  - Amazon EC2 主控台可為您執行 base64 編碼，或是接受 base64 編碼輸入。
  - AWS CLI 第 2 版預設會為您執行二進位參數的 base64 編碼。第 1 AWS CLI 版會為您執行`--user-data`參數的 base64 編碼。
  - 會為您 適用於 Python (Boto3) 的 AWS SDK 執行 `UserData` 參數的 base64 編碼。
- 使用者資料在 base64 編碼之前限制為 16 KB (原始形式)。長度為 n 的字串在 base64 編碼之後的大小為 ceil(n/3)*4。
- 使用者資料在擷取時必須以 base64 解碼。如果您使用執行個體中繼資料或主控台擷取資料，將會自動為您解碼資料。
- 若您停止執行個體，請修改其使用者資料，然後啟動執行個體，則更新後的使用者資料便不會在您啟動執行個體時自動執行。您可以透過 Windows 執行個體設定組態，使更新後的使用者資料指令碼在您啟動執行個體，或是每一次重新開機或啟動執行個體時執行一次。
- 使用者資料是執行個體屬性。如果您從執行個體建立 AMI，執行個體使用者資料不會包含在 AMI 中。

從 EC2 執行個體存取執行個體中繼資料

因為您的執行個體中繼資料可從您執行中的執行個體取得，您無須使用 Amazon EC2 主控台或 AWS CLI。若您正在撰寫要從您的執行個體執行的指令碼，這將會很有幫助。例如，您可以存取從執行個體中繼資料存取您執行個體的本機 IP 地址，管理與外部應用程式的連線。

以下所有項目都被視為執行個體中繼資料，但存取方法各有不同。選擇代表要存取之執行個體中繼資料類型的標籤，以查看更多資訊。

Metadata

執行個體中繼資料屬性分為數種類別。對於每個執行個體中繼資料類別的說明，請參閱 執行個體中繼資料分類。

若要從執行中的執行個體存取中繼資料屬性，請從下列 IPv4 或 IPv6 URI 取得資料。這些 IP 位址是連結本機位址且僅在執行個體中有效。如需詳細資訊，請參閱Link-local 地址。

IPv4

```
http://169.254.169.254/latest/meta-data/
```

IPv6

```
http://[fd00:ec2::254]/latest/meta-data/
```

Dynamic data

若要從執行中的執行個體擷取動態資料，請使用下列其中一個 URI。

IPv4

```
http://169.254.169.254/latest/dynamic/
```

IPv6

```
http://[fd00:ec2::254]/latest/dynamic/
```

範例：使用 cURL 存取

下列範例使用 cURL 擷取高層級的執行個體身分類別。

IMDSV2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/
instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

範例：使用 PowerShell 存取

下列範例使用 PowerShell 擷取高層級的執行個體身分類別。

IMDSV2

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-
identity/
document
rsa2048
```

```
pkcs7
signature
```

如需動態資料的詳細資訊及如何擷取它的範例，請參閱Amazon EC2 執行個體的執行個體身分文件。

User data

若要從執行個體擷取使用者資料，請使用下列其中一個 URI。若要使用 IPv6 地址擷取使用者資料，您必須先啟用並確保該執行個體是 Nitro 型執行個體，且位於支援 IPv6 的子網路中。

IPv4

```
http://169.254.169.254/latest/user-data
```

IPv6

```
http://[fd00:ec2::254]/latest/user-data
```

使用者資料的請求會直接傳回資料 (內容類型 `application/octet-stream`)。如果執行個體沒有任何使用者資料，則請求會傳回 `404 - Not Found`。

範例：使用 cURL 存取以擷取逗號分隔文字

下列範例使用 cURL 擷取指定為逗號分隔文字的使用者資料。

IMDSV2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

範例：使用 PowerShell 存取以擷取逗號分隔文字

下列範例使用 PowerShell 擷取指定為逗號分隔文字的使用者資料。

IMDSV2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = Invoke-RestMethod -Headers
 @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
-Method PUT -Uri http://169.254.169.254/latest/api/token} -Method GET -uri
 http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

範例：使用 cURL 存取以擷取指令碼

下列範例使用 cURL 擷取指定為指令碼的使用者資料。

IMDSV2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-
data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

範例：使用 PowerShell 存取以擷取指令碼

下列範例使用 PowerShell 擷取指定為指令碼的使用者資料。

IMDSV2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

IMDSv1

```
Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

查詢現有執行個體的執行個體中繼資料選項

您可以查詢現有執行個體的執行個體中繼資料選項。

Console

查詢現有執行個體的執行個體中繼資料選項

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體並檢查下列欄位：
   - IMDSv2 – 值為必要或是選用。
   - 允許執行個體中繼資料中的標籤 – 值為已啟用或是已停用。
4. 在已選取執行個體的情況下，選擇動作、執行個體設定、修改執行個體中繼資料選項。

此對話方塊會顯示所選執行個體的執行個體中繼資料服務是啟用還是停用。

AWS CLI

查詢現有執行個體的執行個體中繼資料選項

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-id i-1234567898abcdef0 \
    --query 'Reservations[].Instances[].MetadataOptions'
```

PowerShell

若要查詢現有執行個體的執行個體中繼資料選項，請使用 Tools for PowerShell

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567898abcdef0).Instances.MetadataOptions
```

回應及錯誤訊息

所有執行個體中繼資料都會以文字傳回 (HTTP 內容類型 `text/plain`)。

特定中繼資料資源的請求會傳回適當的值。若資源無法使用，則會傳回 `404 - Not Found` HTTP 錯誤碼。

一般中繼資料資源的請求 (URI 的結尾為 /) 會傳回可用資源的清單，或是 `404 - Not Found` HTTP 錯誤碼 (若沒有該資源的話)。清單項目會位於不同列，以換行字元 (ASCII 10) 終止。

如果 IMDSv1 請求未收到回應，則可能需要 IMDSv2。

對於使用 IMDSv2 提出的請求，可傳回下列 HTTP 錯誤碼：

- `400 - Missing or Invalid Parameters` –PUT 請求無效。
- `401 - Unauthorized` – GET 請求使用的字符無效。建議動作會產生新字符。
- `403 - Forbidden` – 不允許請求，或 IMDS 已關閉。
- `404 - Not Found` – 資源無法使用，或是沒有此類資源。

- 503：請求無法完成。重試 請求。

如果 IMDS 傳回錯誤，curl 會在輸出中列印錯誤訊息，並傳回成功狀態碼。錯誤訊息會儲存在 TOKEN 變數中，進而造成使用字符的 curl 命令失敗。如果您使用 -f 選項呼叫 curl，則會在 HTTP 伺服器出現錯誤時傳回錯誤狀態碼。如果您啟用錯誤處理，Shell 可能會攔截錯誤並停止指令碼。

查詢調節

我們會根據每個執行個體來調整對 IMDS 的查詢，並且我們會限制從執行個體到 IMDS 的同時連線數。

如果您使用 IMDS 擷取 AWS 安全登入資料，請避免在每次交易期間或從大量執行緒或程序同時查詢登入資料，因為這可能會導致限流。相反的，我們建議您快取登入資料，直到其接近到期時間為止。如需有關 IAM 角色以及與該角色相關聯的安全憑證的詳細資訊，請參閱 從執行個體中繼資料中擷取安全登入資料。

若您在存取 IMDS 時遭到限流，請使用指數退避策略重試您的查詢。

使用執行個體中繼資料服務存取執行個體中繼資料

您可以使用以下其中一個方法，從執行中的執行個體存取執行個體中繼資料：

- 執行個體中繼資料服務第 2 版 (IMDSv2) – 工作階段導向方法

  如需範例，請參閱 IMDSv2 的範例。
- 執行個體中繼資料服務第 1 版 (IMDSv1) – 請求/回應方法

  如需範例，請參閱 IMDSv1 的範例。

依預設，您可以使用 IMDSv1 或 IMDSv2，或兩者。

您可以在每個執行個體上設定執行個體中繼資料服務 (IMDS)，只接受 IMDSv2 呼叫，這會導致 IMDSv1 呼叫失敗。如需如何設定執行個體以使用 IMDSv2 的資訊，請參閱 設定執行個體中繼資料服務選項。

PUT 或 GET 標頭對於 IMDSv2 來說是唯一的。如果這些標頭存在於請求中，則該請求適用於 IMDSv2。如果不存在相應標頭，則假設該請求適用於 IMDSv1。

如需深入了解 IMDSv2，請參閱 透過 EC2 執行個體中繼資料服務的增強功能，提高開放式防火牆、反向代理伺服器及 SSRF (伺服器端請求偽造) 弱點的防禦能力。

主題

- 執行個體中繼資料服務第 2 版 的運作方式
- 使用支援的 AWS SDK
- IMDSv2 的範例
- IMDSv1 的範例

執行個體中繼資料服務第 2 版 的運作方式

IMDSv2 會使用工作階段導向請求。使用工作階段導向請求，您就能建立定義工作階段持續期間的工作階段字符，此期間最短 1 秒，最長可達 6 小時。在指定持續期間，您可以將相同的工作階段字符用於後續請求。在指定持續期間到期之後，您必須建立新的工作階段字符，才能使用未來請求。

> **Note**
>
> 本節中的範例使用執行個體中繼資料服務 (IMDS) 的 IPv4 地址：`169.254.169.254`。如果您要透過 IPv6 地址擷取 EC2 執行個體的執行個體中繼資料，請確定您啟用並改用 IPv6 地址：`[fd00:ec2::254]`。IMDS 的 IPv6 地址與 IMDSv2 命令相容。IPv6 地址僅在 IPv6 支援的子網路 (僅支援雙重堆疊或 IPv6) 中的 Nitro 型執行個體可存取。

下列範例會使用 Shell 指令碼和 IMDSv2，擷取頂層的執行個體中繼資料項目。每個範例：

- 使用 PUT 請求，建立持續 6 小時 (21,600 秒) 的工作階段字符
- 將工作階段字符標頭儲存在名為 `TOKEN` (Linux 執行個體) 或 `token` (Windows 執行個體) 的變數中
- 使用字符請求上層中繼資料項目

Linux 範例

您可以執行兩個單獨的命令，或將它們合併。

單獨命令

首先，使用以下命令產生字符。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

然後，使用字符產生使用下列命令的頂層中繼資料項目。

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/
```

組合命令

您可以存放字符並組合命令。下面的範例合併了上述兩個命令，並將工作階段字符標頭存放在一個名為 TOKEN 的變數中。

> **Note**
>
> 如果在建立字符時發生錯誤，變數中會存放錯誤消息而非有效字符，命令也不會發揮作用。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

建立字符之後，您可以重複使用直到到期為止。在下列範例命令中，其中會取得用於啟動執行個體的 AMI ID，會重複使用前一個範例中 $TOKEN 中儲存的字符。

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/ami-id
```

Windows 範例

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -
Uri http://169.254.169.254/latest/meta-data/
```

建立字符之後，您可以重複使用直到到期為止。在下列範例命令中，其中會取得用於啟動執行個體的 AMI ID，會重複使用前一個範例中 $token 中儲存的字符。

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
```

```
-Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

當您使用 IMDSv2 請求執行個體中繼資料時，請求必須包含下列項目：

1. 使用 PUT 請求，在執行個體中繼資料服務中起始工作階段。PUT 請求會傳回一個字符，其必須包含在執行個體中繼資料服務的後續 GET 請求中。字符必須使用 IMDSv2 存取中繼資料。
2. 將字符包含在 IMDS 的所有 GET 請求中。字符使用方式設定為 `required` 時，不包含有效字符或字符已到期的請求會收到 `401 - Unauthorized` HTTP 錯誤碼。
   - 字符是執行個體特定金鑰。字符在其他 EC2 執行個體上無效，而且如果您嘗試在產生字符的執行個體外部使用該字符，則會遭到拒絕。
   - PUT 請求必須包含指定字符存留時間 (TTL)，時間會以秒數表示且最長可達 6 小時 (21,600 秒)。字符會代表邏輯工作階段。TTL 會指定字符有效的時間長度，也就是工作階段的持續期間。
   - 在字符到期之後，若要繼續存取執行個體中繼資料，您必須使用另一個 PUT 建立新的工作階段。
   - 您可以選擇重複使用字符或使用每個請求來建立新字符。對於少量請求，您每次需要存取 IMDS 時，就能更輕鬆地產生和立即使用字符。但為了提升效率，您可以為該字符指定時間更長的持續期間，然後再重複使用該字符，而不需要在每次要請求執行個體中繼資料時寫入 PUT。並行字符數量沒有實際限額，每個都代表它自己的工作階段。但 IMDS2 仍會受到正常 IMDS 連線和調節限額的限制。如需詳細資訊，請參閱查詢調節。

HTTP GET 及 HEAD 方法可在 IMDSv2 執行個體中繼資料請求中使用。如果 PUT 請求包含 X-Forwarded-For 標頭，則會遭到拒絕。

依預設，PUT 請求在 IP 通訊協定等級的 1 回應跳轉限額 (存留時間)。如果您需要更大的跳轉限制，您可以使用 modify-instance-metadata-options AWS CLI 命令進行調整。例如，您可以需要更大的跳轉限制，才能與執行個體上執行之容器服務回溯相容。如需詳細資訊，請參閱修改現有執行個體的執行個體中繼資料選項。

使用支援的 AWS SDK

若要使用 IMDSv2，您的 EC2 執行個體必須使用支援使用 IMDSv2 的 AWS SDK 版本。所有 AWS SDKs 的最新版本都支援使用 IMDSv2。

> **Important**
>
> 建議您保持最新的 SDK 版本，以確保使用最新功能、安全性更新和基礎相依性。不建議您繼續使用不受支援的開發套件版本，由您自行決定。如需詳細資訊，請參閱「AWS 開發套件和工具參考指南」中的 AWS 開發套件及工具維護政策。

以下是支援使用 IMDSv2 的最低版本：

- AWS CLI— 1.16.289
- AWS Tools for Windows PowerShell – 4.0.1.0
- 適用於 .NET 的 AWS SDK – 3.3.634.1
- 適用於 C++ 的 AWS SDK – 1.7.229
- 適用於 Go 的 AWS SDK – 1.25.38
- AWS 適用於 Go 的 SDK v2 – 0.19.0
- 適用於 Java 的 AWS SDK – 1.11.678
- AWS SDK for Java 2.x – 2.10.21
- AWS Node.js 中適用於 JavaScript 的 SDK – 2.722.0
- 適用於 Kotlin 的 AWS SDK – 1.1.4
- 適用於 PHP 的 AWS SDK – 3.147.7
- AWS 適用於 Python 的 SDK (Botocore) – 1.13.25
- 適用於 Python (Boto3) 的 AWS SDK – 1.12.6
- 適用於 Ruby 的 AWS SDK – 3.79.0

IMDSv2 的範例

在 Amazon EC2 執行個體上執行下列範例，以擷取 IMDSv2 的執行個體中繼資料。

在 Windows 執行個體上，您可以使用 Windows PowerShell 或安裝 cURL 或 wget。如果您在 Windows 執行個體上安裝第三方工具，請確定您已仔細閱讀過隨附的文件，因為呼叫和輸出可能與此處所述的內容不同。

範例

- 取得執行個體中繼資料的可用版本
- 取得上層中繼資料項目
- 取得中繼資料項目的值
- 取得可用公有金鑰清單
- 顯示可使用公有金鑰 0 的格式
- 取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)
- 取得執行個體的子網 ID

- 取得執行個體的執行個體標籤

取得執行個體中繼資料的可用版本

此範例會取得執行個體中繼資料的可用版本。每個版本會參照在發佈新執行個體中繼資料類別時的執行個體中繼資料建置。執行個體中繼資料建置版本與 Amazon EC2 API 版本無關。若您有依存於先前版本中結構和資訊的指令碼，您也可以取得先前版本。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/
1.0
2007-01-19
```

```
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

取得上層中繼資料項目

此範例會取得最上層的中繼資料項目。如需有關回應中項目的詳細資訊，請參閱 執行個體中繼資料分類。

請注意，標籤僅在您允許存取時才會包含在此輸出中。如需詳細資訊，請參閱the section called “在執行個體中繼資料中啟用標籤存取權”。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
```

```
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
```

```
security-groups
services/
tags/
```

取得中繼資料項目的值

這些範例會取得從先前範例取得之一些頂層中繼資料項目的值。這些請求會使用先前範例之命令建立的儲存字符。字符不得過期。

cURL

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/local-hostname
```

```
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

取得可用公有金鑰清單

此範例會取得可用公有金鑰的清單。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

顯示可使用公有金鑰 0 的格式

此範例會顯示可使用公有金鑰 0 的格式。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
openssh-key
```

取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)

此範例會取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

```
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

取得執行個體的子網 ID

此範例會取得執行個體的子網 ID。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -
Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

取得執行個體的執行個體標籤

如果開啟了在執行個體中繼資料中存取執行個體標籤的功能，您可以從執行個體中繼資料中取得該執行個體的標籤。如需詳細資訊，請參閱從執行個體中繼資料擷取標籤。

IMDSv1 的範例

在 Amazon EC2 執行個體上執行下列範例，以擷取 IMDSv1 的執行個體中繼資料。

在 Windows 執行個體上，您可以使用 Windows PowerShell 或安裝 cURL 或 wget。如果您在 Windows 執行個體上安裝第三方工具，請確定您已仔細閱讀過隨附的文件，因為呼叫和輸出可能與此處所述的內容不同。

範例

- 取得執行個體中繼資料的可用版本
- 取得上層中繼資料項目
- 取得中繼資料項目的值
- 取得可用公有金鑰清單
- 顯示可使用公有金鑰 0 的格式
- 取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)
- 取得執行個體的子網 ID
- 取得執行個體的執行個體標籤

取得執行個體中繼資料的可用版本

此範例會取得執行個體中繼資料的可用版本。每個版本會參照在發佈新執行個體中繼資料類別時的執行個體中繼資料建置。執行個體中繼資料建置版本與 Amazon EC2 API 版本無關。若您有依存於先前版本中結構和資訊的指令碼，您也可以取得先前版本。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
```

```
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

取得上層中繼資料項目

此範例會取得最上層的中繼資料項目。如需有關回應中項目的詳細資訊，請參閱 執行個體中繼資料分類。

請注意，標籤僅在您允許存取時才會包含在此輸出中。如需詳細資訊，請參閱the section called “在執行個體中繼資料中啟用標籤存取權”。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
```

```
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

取得中繼資料項目的值

這些範例會取得從先前範例取得之一些頂層中繼資料項目的值。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-
id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-
hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

取得可用公有金鑰清單

此範例會取得可用公有金鑰的清單。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/ 0=my-public-key
```

顯示可使用公有金鑰 0 的格式

此範例會顯示可使用公有金鑰 0 的格式。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
openssh-key
```

取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)

此範例會取得公有金鑰 0 (格式為 OpenSSH 金鑰格式)。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

取得執行個體的子網 ID

此範例會取得執行個體的子網 ID。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/network/
interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

取得執行個體的執行個體標籤

如果開啟了在執行個體中繼資料中存取執行個體標籤的功能，您可以從執行個體中繼資料中取得該執行個體的標籤。如需詳細資訊，請參閱從執行個體中繼資料擷取標籤。

轉換為使用 執行個體中繼資料服務第 2 版

如果您想要將執行個體設定為只接受執行個體中繼資料服務第 2 版 (IMDSv2) 呼叫，建議您使用下列工具和轉換路徑。

主題

- 轉換至 IMDSv2 的工具
- 需要 IMDSv2 的建議路徑

轉換至 IMDSv2 的工具

下列工具可協助您識別、監控和管理軟體從 IMDSv1 到 IMDSv2 的轉換。如需如何使用這些工具的指示，請參閱 需要 IMDSv2 的建議路徑。

AWS 軟體

最新版本的 AWS CLI 和 AWS SDKs 支援 IMDSv2。若要使用 IMDSv2，請更新您的 EC2 執行個體以使用最新版本。如需支援 IMDSv2 的最低 AWS SDK 版本，請參閱 使用支援的 AWS SDK。

所有 Amazon Linux 2 和 Amazon Linux 2023 軟體套件都支援 IMDSv2。Amazon Linux 2023 預設會停用 IMDSv1。

IMDS Packet Analyzer

IMDS Packet Analyzer 是一種開放原始碼工具，可識別並記錄執行個體開機階段和執行時間操作期間的 IMDSv1 呼叫。透過分析這些日誌，您可以精確識別在執行個體上進行 IMDSv1 呼叫的軟體，並判斷需要更新哪些項目才能僅在執行個體上支援 IMDSv2。您可以從命令列執行 IMDS Packet Analyzer，或將其安裝為服務。如需詳細資訊，請參閱 GitHub 上的 AWS ImdsPacketAnalyzer。

CloudWatch

CloudWatch 提供下列兩個指標來監控您的執行個體：

`MetadataNoToken` – IMDSv2 使用字符後端工作階段，而 IMDSv1 不使用。`MetadataNoToken` 指標會追蹤對使用 IMDSv1 的執行個體中繼資料服務 (IMDS) 的呼叫數量。透過追蹤此指標至零，您可以決定是否要升級所有軟體以使用 IMDSv2 及升級時間。

`MetadataNoTokenRejected` – 停用 IMDSv1 之後，您可以使用 `MetadataNoTokenRejected` 指標來追蹤 IMDSv1 呼叫嘗試和拒絕的次數。透過追蹤此指標，您可以確定自己的軟體是否需要更新才能使用 IMDSv2。

對於每個 EC2 執行個體，這些指標是互斥的。啟用 IMDSv1 時 `MetadataNoToken` (`httpTokens = optional`)，只會發出 。當 IMDSv1 停用時 `MetadataNoTokenRejected` (`httpTokens = required`)， 只會發出。如需何時使用這些指標，請參閱 需要 IMDSv2 的建議路徑。

如需詳細資訊，請參閱執行個體指標。

啟動 APIs

新執行個體：使用 RunInstances API 啟動需要使用 IMDSv2 的新執行個體。如需詳細資訊，請參閱設定新執行個體的執行個體中繼資料選項。

現有執行個體：使用 ModifyInstanceMetadataOptions API 要求在現有執行個體上使用 IMDSv2。如需詳細資訊，請參閱修改現有執行個體的執行個體中繼資料選項。

Auto Scaling 群組啟動的新執行個體：若要要求在 Auto Scaling 群組啟動的所有新執行個體上使用 IMDSv2，Auto Scaling 群組可以使用啟動範本或啟動組態。當您建立啟動範本或建立啟動組態時，必須設定 `MetadataOptions` 參數以要求使用 IMDSv2。Auto Scaling 群組會使用新啟動範本或啟動組態來啟動新執行個體，但現有的執行個體不受影響。

Auto Scaling 群組中的現有執行個體：使用 ModifyInstanceMetadataOptions API 要求在現有執行個體上使用 IMDSv2，或終止執行個體，Auto Scaling 群組將使用新啟動範本或啟動組態中定義的執行個體中繼資料選項設定來啟動新的替換執行個體。

AMI

設定為 `ImdsSupport` 參數的 AMIs 預設`v2.0`會啟動需要 IMDSv2 的執行個體。Amazon Linux 2023 已使用 設定`ImdsSupport = v2.0`。

新 AMIs：建立新的 AMI `v2.0`時，使用 register-image CLI 命令將 `ImdsSupport` 參數設定為 。

現有 AMIs：使用 modify-image-attribute CLI 命令，在修改現有 AMI `v2.0`時將 `ImdsSupport` 參數設定為 。

如需詳細資訊，請參閱設定 AMI。

帳戶層級控制項

您可以在帳戶層級設定所有執行個體中繼資料選項的預設值。當您啟動執行個體時，會自動套用預設值。如需詳細資訊，請參閱將 IMDSv2 設定為帳戶的預設值。

您也可以強制執行在帳戶層級使用 IMDSv2 的需求。啟用 IMDSv2 強制執行時：

- 新執行個體：設定為在啟用 IMDSv1 的情況下啟動的執行個體將無法啟動
- 停用 IMDSv1 的現有執行個體：將防止嘗試在現有執行個體上啟用 IMDSv1。
- 已啟用 IMDSv1 的現有執行個體：已啟用 IMDSv1 的現有執行個體不會受到影響。

如需詳細資訊，請參閱在帳戶層級強制執行 IMDSv2。

IAM 政策和 SCP

您可以使用 IAM 政策 AWS Organizations 或服務控制政策 (SCP) 來控制使用者，如下所示：

- 除非執行個體設定為使用 IMDSv2，否則無法使用 RunInstances API 啟動執行個體。
- 無法使用 ModifyInstanceMetadataOptions API 修改現有執行個體，以重新啟用 IMDSv1。

IAM 政策或 SCP 必須包含下列 IAM 條件索引鍵：

- `ec2:MetadataHttpEndpoint`
- `ec2:MetadataHttpPutResponseHopLimit`
- `ec2:MetadataHttpTokens`

如果 API 或 CLI 呼叫中的參數不符合包含條件金鑰的政策中指定的狀態，則 API 或 CLI 呼叫會失敗並顯示 `UnauthorizedOperation` 回應。

此外，您可以選擇另外一個保護層，以強制執行從 IMDSv1 變更為 IMDSv2 的作業。在與透過 EC2 角色登入資料呼叫APIs 相關的存取管理層中，您可以在 IAM 政策 AWS Organizations 或服務控制政策 (SCPs) 中使用條件金鑰。具體而言，在 IAM 政策中使用數值為 `2.0` 的條件索引鍵 `ec2:RoleDelivery`，從 IMDSv1 取得之 EC2 角色憑證產生的 API 呼叫都會收到 `Unauthorized0peration` 回應。也能利用 SCP 要求的條件進行更廣泛範圍的作業。這可確保透過 IMDSv1 傳遞的登入資料無法實際用於呼叫 APIs因為任何不符合指定條件的 API 呼叫都會收到`Unauthorized0peration`錯誤。

如需 IAM 政策的範例，請參閱使用執行個體中繼資料。如需 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。

宣告政策

使用宣告政策 (的功能 AWS Organizations) 在整個組織中集中設定 IMDS 帳戶預設值，包括 IMDSv2 強制執行。如需範例政策，請參閱AWS Organizations 《 使用者指南》中支援的宣告政策一節中的執行個體中繼資料索引標籤。

需要 IMDSv2 的建議路徑

使用上述工具，我們建議將下列路徑轉換為 IMDSv2：

- 步驟 1：使用 IMDSv2=選用和稽核 IMDSv1 用量來識別執行個體
- 步驟 2：將軟體更新至 IMDSv2
- 步驟 3：執行個體需要 IMDSv2
- 步驟 4：將 IMDSv2=必要設為預設值
- 步驟 5：強制執行個體要求 IMDSv2

步驟 1：使用 IMDSv2=選用和稽核 IMDSv1 用量來識別執行個體

若要評估 IMDSv2 遷移範圍，請識別設定為允許 IMDSv1 或 IMDSv2 的執行個體，並稽核 IMDSv1 呼叫。

1. 識別設定為允許 IMDSv1 或 IMDSv2 的執行個體：

   Amazon EC2 console

   1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
   2. 在導覽窗格中，選擇執行個體。
   3. 若要只查看設定為允許 IMDSv1 或 IMDSv2 的執行個體，請新增篩選條件 IMDSv2 = 選用。

4. 或者，若要查看所有執行個體的 IMDSv2 是否為選用或必要，請開啟偏好設定視窗（齒輪圖示），開啟 IMDSv2，然後選擇確認。這會將 IMDSv2 資料欄新增至執行個體資料表。

AWS CLI

使用 describe-instances 命令並依 篩選`metadata-options.http-tokens = optional`，如下所示：

```
aws ec2 describe-instances --filters "Name=metadata-options.http-
tokens,Values=optional" --query "Reservations[*].Instances[*].[InstanceId]" --
output text
```

2. 在每個執行個體上稽核 IMDSv1 呼叫：

使用 CloudWatch 指標 `MetadataNoToken`。此指標會顯示對執行個體上 IMDS 的 IMDSv1 呼叫次數。如需詳細資訊，請參閱執行個體指標。

3. 在進行 IMDSv1 呼叫的執行個體上識別軟體：

使用開放原始碼 IMDS Packet Analyzer 在執行個體的開機階段和執行時間操作期間識別和記錄 IMDSv1 呼叫。使用此資訊來識別要更新的軟體，讓您的執行個體準備好僅使用 IMDSv2。您可以從命令列執行 IMDS Packet Analyzer，或將其安裝為服務。

步驟 2：將軟體更新至 IMDSv2

將執行個體上使用角色登入資料的所有 SDKs、CLIs 和軟體更新為 IMDSv2-compatible版本。如需更新 CLI 的詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的安裝或更新至 AWS CLI的最新版本。

步驟 3：執行個體需要 IMDSv2

透過 `MetadataNoToken` 指標確認零 IMDSv1 呼叫後，請將您現有的執行個體設定為需要 IMDSv2。此外，將所有新執行個體設定為需要 IMDSv2。換句話說，在所有現有和新的執行個體上停用 IMDSv1。

1. 將現有執行個體設定為需要 IMDSv2：

Amazon EC2 console

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體。
4. 選擇動作、執行個體設定、修改執行個體中繼資料選項。
5. 針對 IMDSv2，請選擇必要。
6. 選擇儲存。

AWS CLI

使用 modify-instance-metadata-options CLI 命令來指定僅使用 IMDSv2。

> **Note**
>
> 您可以在執行中的執行個體上修改此設定。變更會立即生效，而不需要重新啟動執行個體。

如需詳細資訊，請參閱需要使用 IMDSv2。

2. 在停用 IMDSv1 後監控問題：
   1. 使用 `MetadataNoTokenRejected` CloudWatch 指標追蹤嘗試和拒絕 IMDSv1 呼叫的次數。
   2. 如果`MetadataNoTokenRejected`指標在遇到軟體問題的執行個體上記錄 IMDSv1 呼叫，這表示軟體需要更新才能使用 IMDSv2。
3. 設定新執行個體以要求 IMDSv2：

   Amazon EC2 console

   1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
   2. 依照步驟啟動執行個體。
   3. 展開進階詳細資訊，對於中繼資料版本，僅選擇 V2 （需要金鑰）。
   4. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。

      如需詳細資訊，請參閱啟動時設定執行個體。

AWS CLI

AWS CLI：使用 run-instances 命令並指定需要 IMDSv2。

步驟 4：將 IMDSv2=必要設為預設值

您可以在帳戶或組織層級將 IMDSv2=必要設定為預設組態。這可確保所有新啟動的執行個體都會自動設定為需要 IMDSv2。

1. 設定帳戶層級預設值：

   Amazon EC2 console

   1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
   2. 在導覽窗格中，選擇 Dashboard (儀表板)。
   3. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。
   4. 在 IMDS 預設值下，選擇管理。
   5. 針對執行個體中繼資料服務 ，選擇已啟用。
   6. 對於中繼資料版本，選擇僅 V2 （需要金鑰）。
   7. 選擇更新。

   AWS CLI

   使用 modify-instance-metadata-defaults CLI 命令並指定 `--http-tokens required`和 `--http-put-response-hop-limit 2`。

   如需詳細資訊，請參閱將 IMDSv2 設定為帳戶的預設值。
2. 或者，使用宣告政策設定組織層級預設值：

   使用宣告政策將 IMDSv2 的組織預設值設定為必要。如需範例政策，請參閱AWS Organizations《 使用者指南》中支援的宣告政策一節中的執行個體中繼資料索引標籤。

步驟 5：強制執行個體要求 IMDSv2

在您確認任何執行個體上沒有 IMDSv1 的相依性之後，建議您在所有新執行個體上強制執行 IMDSv2。

使用下列其中一個選項來強制執行 IMDSv2：

1. 使用帳戶屬性強制執行 IMDSv2

   您可以在每個帳戶層級強制使用 IMDSv2 AWS 區域。強制執行時，執行個體只能在設定為需要 IMDSv2 時啟動。無論執行個體或 AMI 的設定方式為何，此強制執行都適用。如需詳細資訊，請參閱在帳戶層級強制執行 IMDSv2。若要在組織層級套用此設定，請設定宣告政策。如需範例政策，請參閱AWS Organizations 《 使用者指南》中支援的宣告政策一節中的執行個體中繼資料索引標籤。

   若要防止強制撤銷，您應該使用 IAM 政策來防止存取 ModifyInstanceMetadataDefaults API。如需詳細資訊，請參閱使用 IAM 政策。

   > **Note**
   >
   > 此設定不會變更現有執行個體的 IMDS 版本，但會封鎖在目前已停用 IMDSv1 的現有執行個體上啟用 IMDSv1。

   > **Warning**
   >
   > 如果已啟用 IMDSv2 強制執行`httpTokens`，且在啟動時執行個體組態、帳戶設定或 AMI 組態`required`中未設定為 ，則執行個體啟動將會失敗。如需故障診斷資訊，請參閱啟動 IMDSv1-enabled的執行個體失敗。

2. 或者，使用以下 IAM 或 SCP 條件金鑰強制執行 IMDSv2：

   - `ec2:MetadataHttpTokens`
   - `ec2:MetadataHttpPutResponseHopLimit`
   - `ec2:MetadataHttpEndpoint`

   這些條件金鑰控制 RunInstances 和 ModifyInstanceMetadataOptions APIs和對應 CLIs。如果已建立政策，而且 API 呼叫中的參數與使用條件金鑰之政策中指定的狀態不相符，則 API 或 CLI 會失敗並顯示 `UnauthorizedOperation` 回應。

   如需 IAM 政策的範例，請參閱使用執行個體中繼資料。

限制對執行個體中繼資料服務的存取

您可以考慮使用本機防火牆規則，以停用從一些或所有流程對於執行個體中繼資料服務 (IMDS) 的存取。

對於 Nitro 型執行個體，當 VPC 內的網路設備 (例如虛擬路由器) 將封包轉送至 IMDS 位址，且停用執行個體上預設的來源/目的地檢查，可從您自己的網路中連線到 IMDS。若要防止 VPC 外部的來源連線到 IMDS，建議您修改網路設備的組態，以捨棄 IMDS 目的地 IPv4 地址為 169.254.169.254 的封包；在啟用 IPv6 端點的情況下，則捨棄 IMDS 目的地 IPv6 地址 [fd00:ec2::254] 的封包。

限制 Linux 執行個體的 IMDS 存取

使用 iptables 限制存取

下列範例會使用 Linux iptables 及其 owner 模組以防止 Apache Web 伺服器 (依據 apache 的預設安裝使用者 ID) 存取 169.254.169.254。其使用拒絕規則以拒絕所有執行個體中繼資料請求 (無論是 IMDSv1 或 IMDSv2) 以該使用者的身分執行任何程序。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner
 --uid-owner apache --jump REJECT
```

或者，您可以使用允許規則，考慮只允許存取特定使用者或群組。就安全觀點而言，允許規則可能更加簡單，因為它們會要求您做出關於那些軟體需要存取執行個體中繼資料的決策。如果您使用允許規則，即使您稍後變更執行個體上的軟體或組態，您也不太可能意外地允許軟體存取中繼資料服務 (亦即您並非有意存取)。您也可以使用允許規則來組合群組使用方式，如此您就能新增並從允許的群組移除使用者，而不需要變更防火牆規則。

下列範例可防止所有程序 (在使用者帳戶 trustworthy-user 中執行的程序除外) 存取 IMDS。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match
 owner ! --uid-owner trustworthy-user --jump REJECT
```

> **Note**
>
> - 若要使用本機防火牆規則，您需要採用之前的範例命令，才能滿足您的需求。
> - 依預設，iptables 規則在系統重新啟動時不會持續存在。但可使用 OS 功能將之設定為持續存在，但此處並不會敘述此內容。

- 如果此群組是指定本機使用者的主要群組，iptables owner 模組僅會比對群組成員資格。而不會比對其他群組。

使用 PF 或 IPFW 限制存取

如果您使用 FreeBSD 或 OpenBSD，您也可以考慮使用 PF 或 IPFW。下列範例將 IMDS 的存取權僅限於根使用者。

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

> **Note**
>
> PF 及 IPFW 命令的順序非常重要。PF 預設會試最後一個比對規則，而 IPFW 預設是第一個比對規則。

限制 Windows 執行個體的 IMDS 存取

使用 Windows 防火牆以限制存取

下列 PowerShell 範例會使用內建 Windows 防火牆以防止 Internet Information Server Webs 伺服器 (依據 NT AUTHORITY\IUSR 的預設安裝使用者 ID) 存取 169.254.169.254。其使用拒絕規則以拒絕所有執行個體中繼資料請求 (無論是 IMDSv1 或 IMDSv2) 以該使用者的身分執行任何程序。

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount ("NT
 AUTHORITY\IUSR")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
```

```
PS C:\> $BlockPrincipalSDDL = "D:(A;;CC;;;$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service from IIS" -Action
 block -Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $BlockPrincipalSDDL
```

或者，您可以使用允許規則，考慮只允許存取特定使用者或群組。就安全觀點而言，允許規則可能更加簡單，因為它們會要求您做出關於那些軟體需要存取執行個體中繼資料的決策。如果您使用允許規則，即使您稍後變更執行個體上的軟體或組態，您也不太可能意外地允許軟體存取中繼資料服務 (亦即您並非有意存取)。您也可以使用允許規則來組合群組使用方式，如此您就能新增並從允許的群組移除使用者，而不需要變更防火牆規則。

下列範例會防止以變數 `blockPrincipal` 中指定 OS 群組身分執行的所有程序存取執行個體中繼資料 (在此範例中，Windows 群組 `Everyone`)，但 `exceptionPrincipal` 中指定的程序除外 (在此範例中，是名為 `trustworthy-users` 的群組)。您必須指定拒絕和允許原則，因為 Windows 防火牆 (與 Linux iptables 中的 `! --uid-owner trustworthy-user` 規則不同) 不會提供快速鍵機制，以透過拒絕所有其他規則而僅允許特定原則 (使用者或群組)。

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("Everyone")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $exceptionPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("trustworthy-users")
PS C:\> $ExceptionPrincipalSID =
 $exceptionPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $PrincipalSDDL = "O:LSD:(D;;CC;;;$ExceptionPrincipalSID)(A;;CC;;;
$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service for
 $($blockPrincipal.Value), exception: $($exceptionPrincipal.Value)" -Action block -
Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $PrincipalSDDL
```

> **Note**
>
> 若要使用本機防火牆規則，您需要採用之前的範例命令，才能滿足您的需求。

使用 netsh 規則限制存取

您可以使用 `netsh` 而考慮封鎖所有軟體，但那些規則較不彈性。

```
C:\> netsh advfirewall firewall add rule name="Block metadata service altogether"
 dir=out protocol=TCP remoteip=169.254.169.254 action=block
```

> **Note**
>
> - 若要使用本機防火牆規則，您需要採用之前的範例命令，才能滿足您的需求。
> - 必須從提升權限的命令提示設定 `netsh`，但不可設定為拒絕或允許特定原則。

設定執行個體中繼資料服務選項

執行個體中繼資料服務 (IMDS) 會在每個 EC2 執行個體本機上執行。執行個體中繼資料選項是指一組能控制 IMDS 在 EC2 執行個體上的可存取性和行為的祖態。

您可以在每個執行個體上設定下列執行個體中繼資料選項：

執行個體中繼資料服務 (IMDS)：`enabled | disabled`

> 您可以在執行個體上啟用或停用 IMDS。停用時，您或任何程式碼都無法在執行個體上存取執行個體中繼資料。
>
> IMDS 在執行個體上有兩個端點：IPv4 (`169.254.169.254`) 和 IPv6 (`[fd00:ec2::254]`)。當您啟用 IMDS 時，IPv4 端點會自動啟用。如果您想要啟用 IPv6 端點，則需要明確啟用。

IMDS IPv6 端點：`enabled | disabled`

> 您可以在執行個體上明確啟用 IPv6 IMDS 端點。啟用 IPv6 端點時，IPv4 端點會保持啟用狀態。IPv6 端點僅在 IPv6 支援的子網路 (僅支援雙重堆疊或 IPv6) 中的 Nitro 型執行個體受到支援。

中繼資料版本：`IMDSv1 or IMDSv2 (token optional) | IMDSv2 only (token required)`

> 要求執行個體中繼資料時，IMDSv2 呼叫需要字符。IMDSv1 呼叫不需要字符。您可以設定執行個體以允許 IMDSv1 或 IMDSv2 呼叫 (其中字符為選用項目)，或者僅允許 IMDSv2 呼叫 (其中字符為必要項目)。

中繼資料回應跳轉限制：1–64

> 跳轉限制是允許 PUT 回應進行的網路跳轉數量。您可以將跳轉限制設定為最小 1及最大 64。在容器環境中，1 的跳轉限制可能會導致問題。如需如何緩解這些問題的資訊，請參閱 執行個體中繼資料存取考量 下容器環境的相關資訊。

存取執行個體中繼資料中的標籤：`enabled | disabled`

您可以從執行個體中繼資料啟用或停用執行個體標籤的存取權。如需詳細資訊，請參閱使用執行個體中繼資料檢視 EC2 執行個體的標籤。

若要檢視執行個體的目前組態，請參閱 查詢現有執行個體的執行個體中繼資料選項。

設定執行個體中繼資料選項的位置

執行個體中繼資料選項可以在不同層級設定，如下所示：

- 帳戶：您可以在帳戶層級為每個 AWS 區域設定執行個體中繼資料選項的預設值。啟動執行個體時，執行個體中繼資料選項會自動設定為帳戶層級值。您可以在啟動時變更這些值。帳戶層級預設值不會影響現有的執行個體。
- AMI：註冊或修改 AMI 時，可以將 `imds-support` 參數設定為 `v2.0`。使用此 AMI 啟動執行個體時，執行個體中繼資料版本會自動設定為 IMDSv2 且跳轉限制設定為 2。
- 執行個體：您可以在啟動時變更所有在執行個體上的執行個體中繼資料選項，並覆寫預設設定。您也可以在執行中或已停止的執行個體啟動後，變更執行個體中繼資料選項。請注意，IAM 或 SCP 政策可能會限制變更。

如需詳細資訊，請參閱設定新執行個體的執行個體中繼資料選項及修改現有執行個體的執行個體中繼資料選項。

執行個體中繼資料選項的優先順序

每個執行個體中繼資料選項的值都是在執行個體啟動時，依照階層優先順序來決定。階層結構如下，從最上方的最高優先權開始：

- 優先順序 1：啟動時的執行個體組態：值可以在啟動範本或執行個體組態中指定。此處指定的任何值都會覆寫在帳戶層級或 AMI 中指定的值。
- 優先順序 2：帳戶設定 – 如果在執行個體啟動時未指定值，則由帳戶層級設定（針對每個設定 AWS 區域) 決定。帳戶層級設定會包括每個中繼資料選項的值，或完全不表示任何偏好設定。
- 優先順序 3：AMI 組態：如果未在執行個體啟動或帳戶層級指定值，則由 AMI 組態決定。這僅會套用到 `HttpTokens` 和 `HttpPutResponseHopLimit`。

每個中繼資料選項都會分別進行評估。執行個體可以混合使用直接執行個體組態、帳戶層級預設值，以及來自 AMI 的組態加以設定。

您可以在執行中或已停止的執行個體啟動後，變更任何中繼資料選項的值，除非變更內容受限於 IAM 或 SCP 政策。

> **Note**
>
> 帳戶層級 IMDSv2 強制執行設定會在優先順序決定執行個體的 IMDS 設定之後進行評估。啟用 IMDSv2 強制執行時，使用 IMDSv1 啟用的執行個體將會失敗。如需強制執行的詳細資訊，請參閱 在帳戶層級強制執行 IMDSv2。

> **Warning**
>
> 如果已啟用 IMDSv2 強制執行，且未在啟動時的執行個體組態、帳戶設定或 AMI 組態`required`中`httpTokens`設定為 ，則您的啟動將會失敗。

範例 1 – 判斷中繼資料選項的值

在此範例中，EC2 執行個體會在將 `HttpPutResponseHopLimit` 以帳戶層級設定為 1 的區域中啟動。指定的 AMI 已將 `ImdsSupport` 設定為 `v2.0`。啟動時，不會直接在執行個體上指定中繼資料選項。執行個體會使用下列中繼資料選項啟動：

```
"MetadataOptions": {
    ...
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    ...
```

這些值的判斷方式如下所示：

- 啟動時未指定中繼資料選項：在執行個體啟動期間，未在執行個體啟動參數或啟動範本中提供中繼資料選項的特定值。
- 帳戶設定的優先順序為次要：如果啟動時未指定特定值，則以區域內帳戶層級的設定為準。這表示會套用在帳戶層級設定的預設值。在此情況下，已將 `HttpPutResponseHopLimit` 設定為 1。
- AMI 設定的優先順序為最後：如果沒有在啟動時或 `HttpTokens` (執行個體中繼資料版本) 之帳戶層級指定的特定值，則套用 AMI 設定。在此情況下，AMI 設定 `ImdsSupport: v2.0` 決定 `HttpTokens` 設定為 `required`。請注意，雖然 AMI 設定 `ImdsSupport:`

v2.0 旨在設定 HttpPutResponseHopLimit: 2，但優先順序較高的帳戶層級設定 HttpPutResponseHopLimit: 1 已將其覆寫。

範例 2 – 判斷中繼資料選項的值

在此範例中，EC2 執行個體的啟動設定與範例 1 相同，但 HttpTokens 在啟動時執行在執行個體上設定為 optional。執行個體會使用下列中繼資料選項啟動：

```
"MetadataOptions": {
    ...
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    ...
```

HttpPutResponseHopLimit 的值採用與範例 1 相同的決定方法。但是，HttpTokens 的值決定方法如下：啟動時，在執行個體上設定的中繼資料選項為優先。即使 AMI 使用 ImdsSupport: v2.0 設定 (也就是，將 HttpTokens 設定為 required)，啟動時在執行個體上指定的值 (將 HttpTokens 設定為 optional) 仍為優先。

範例 3 – 判斷啟用 HttpTokensEnforced 的中繼資料選項的值

在此範例中， 區域中的帳戶具有 HttpTokens = required和 HttpTokensEnforced = enabled。

請考慮下列 EC2 執行個體啟動嘗試：

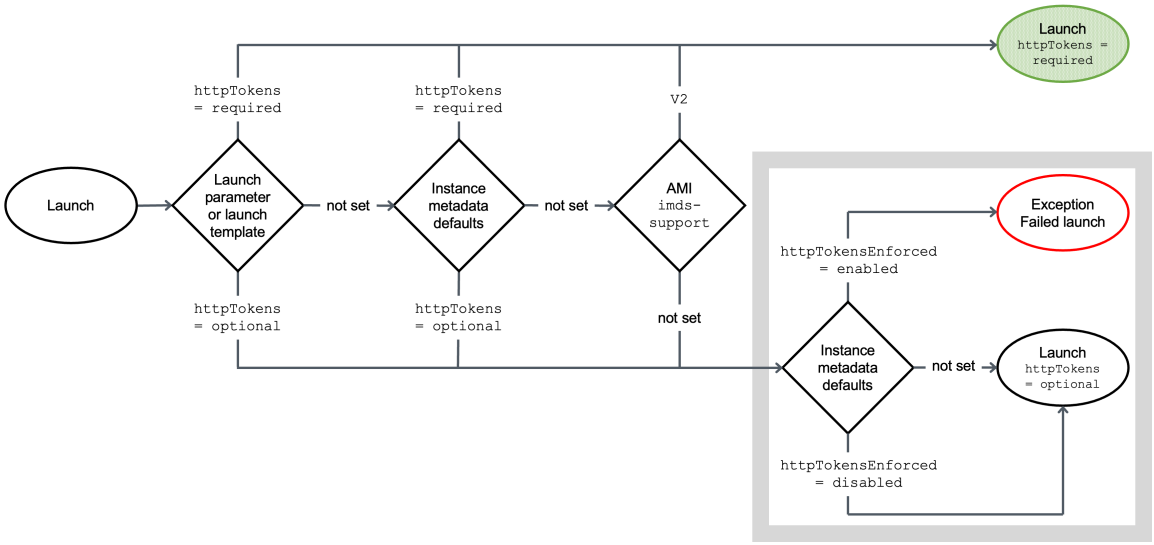
- 啟動嘗試HttpTokens設定為 optional – 啟動失敗，因為已啟用帳戶層級強制執行 (HttpTokensEnforced = enabled)，且啟動參數優先於帳戶預設值。
- HttpTokens 將 設定為 的啟動嘗試 required – 啟動成功，因為它符合帳戶層級強制執行。
- 未指定HttpTokens任何值的啟動嘗試 – 啟動成功，因為根據required帳戶設定，該值預設為 。

設定執行個體中繼資料版本

啟動執行個體時，執行個體中繼資料版本的值為 IMDSv1 或 IMDSv2 （金鑰選用） (httpTokens=optional) 或僅限 IMDSv2 （需要金鑰） (**httpTokens=required**)。

在執行個體啟動時，您可以手動指定中繼資料版本的值，或使用預設值。如果您手動指定值，即會覆寫任何預設值。如果您選擇不手動指定值，則會由預設設定的組合決定。

下列流程圖顯示啟動時執行個體的中繼資料版本如何取決於不同組態層級的設定，以及強制執行的評估位置。以下資料表提供每個層級的特定設定。



此表格顯示啟動時的執行個體中繼資料版本 (如欄 4 中產生的執行個體組態所示) 如何由不同組態層級的設定決定。優先順序從左到右，其中第一欄採用最高優先順序，如下所示：

- 欄 1：啟動參數：代表您在啟動時手動指定之執行個體上的設定。
- 欄 2：帳戶層級預設值：代表帳戶的設定。
- 欄 3：AMI 預設值：代表 AMI 上的設定。

啟動參數	帳戶層級預設值	AMI 預設值	產生的執行個體組態
僅限 V2 (需要字符)	無偏好設定	僅限 V2	僅限 V2
僅限 V2 (需要字符)	僅限 V2	僅限 V2	僅限 V2
僅限 V2 (需要字符)	V1 或 V2	僅限 V2	僅限 V2
V1 或 V2 (字符為選用項目)	無偏好設定	僅限 V2	V1 或 V2
V1 或 V2 (字符為選用項目)	僅限 V2	僅限 V2	V1 或 V2
V1 或 V2 (字符為選用項目)	V1 或 V2	僅限 V2	V1 或 V2

啟動參數	帳戶層級預設值	AMI 預設值	產生的執行個體組態
未設定	無偏好設定	僅限 V2	僅限 V2
未設定	僅限 V2	僅限 V2	僅限 V2
未設定	V1 或 V2	僅限 V2	V1 或 V2
僅限 V2 (需要字符)	無偏好設定	null	僅限 V2
僅限 V2 (需要字符)	僅限 V2	null	僅限 V2
僅限 V2 (需要字符)	V1 或 V2	null	僅限 V2
V1 或 V2 (字符為選用項目)	無偏好設定	null	V1 或 V2
V1 或 V2 (字符為選用項目)	僅限 V2	null	V1 或 V2
V1 或 V2 (字符為選用項目)	V1 或 V2	null	V1 或 V2
未設定	無偏好設定	null	V1 或 V2
未設定	僅限 V2	null	僅限 V2
未設定	V1 或 V2	null	V1 或 V2

使用 IAM 條件索引鍵限制執行個體中繼資料選項

您可以在 IAM 政策或 SCP 中使用 IAM 條件索引鍵，如下所示：

- 只有在執行個體設定為需要使用 IMDSv2 時才允許啟動該執行個體
- 限制允許的轉跳數目
- 關閉對執行個體中繼資料的存取

任務

- 設定新執行個體的執行個體中繼資料選項

- 修改現有執行個體的執行個體中繼資料選項

> **Note**
>
> 您應該謹慎繼續執行和仔細進行測試，然後再進行任何變更。請記得下列事項：
>
> - 如果您強制使用 IMDSv2，使用 IMDSv1 來存取執行個體中繼資料的應用程式或代理程式便會中斷。
> - 如果您關閉對執行個體中繼資料的所有存取，則需要存取執行個體中繼資料才能正常運作的應用程式或代理程式便會中斷。
> - 對於 IMDSv2，在檢索字符時必須使用 `/latest/api/token`。
> - (僅限 Windows) 如果您的 PowerShell 版本早於 4.0，您必須更新至 Windows Management Framework 4.0，才能使用 IMDSv2。

設定新執行個體的執行個體中繼資料選項

您可以為新執行個體設定下列執行個體中繼資料選項。

選項

- 需要使用 IMDSv2
- 啟用 IMDS IPv4 和 IPv6 端點
- 關閉對執行個體中繼資料的存取
- 允許存取執行個體中繼資料中的標籤

> **Note**
>
> 這些選項的設定是在帳戶層級所設定，可直接在帳戶中設定，或使用宣告式政策來設定。它們必須在每個您要設定執行個體中繼資料選項的 AWS 區域 中設定。您可使用宣告式政策同時在多個區域及多個帳戶套用設定。使用宣告式政策時，您無法直接在帳戶中修改設定。本主題說明如何直接在帳戶內配置設定。如需使用宣告式政策的相關資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

需要使用 IMDSv2

您可以使用下列方法來要求在新執行個體上使用 IMDSv2。

需要 IMDSv2

- 將 IMDSv2 設定為帳戶的預設值
- 在帳戶層級強制執行 IMDSv2
- 啟動時設定執行個體
- 設定 AMI
- 使用 IAM 政策

將 IMDSv2 設定為帳戶的預設值

您可以在帳戶層級為每個執行個體中繼資料服務 (IMDS) 設定預設版本 AWS 區域。這表示在啟動新的執行個體時，執行個體中繼資料版本會自動設定為帳戶層級的預設值。但是，您可以在啟動時或啟動後手動覆寫該值。如需有關帳戶層級設定和手動覆寫如何影響執行個體的詳細資訊，請參閱 執行個體中繼資料選項的優先順序。

> **Note**
>
> 設定帳戶層級預設值不會重設現有的執行個體。例如，如果您將帳戶層級預設值設定為 IMDSv2，則任何設為 IMDSv1 的現有執行個體都不會受到影響。若要變更現有執行個體的值，您必須手動變更執行個體本身的值。

您可以將執行個體中繼資料版本的帳戶預設值設定為 IMDSv2，因此所有在帳戶啟動中的新執行個體都需要 IMDSv2，且 IMDSv1 會停用。當您使用此帳戶預設值啟動執行個體時，下列是執行個體的預設值：

- 主控台：中繼資料版本僅設定為 V2 (需要字符)，且中繼資料回應跳轉限制設定為 2。
- AWS CLI：將 `HttpTokens` 設定為 `required`，並將 `HttpPutResponseHopLimit` 設定為 2。

> **Note**
>
> 在將帳戶預設值設定為 IMDSv2 之前，請確定您的執行個體不依賴於 IMDSv1。如需詳細資訊，請參閱需要 IMDSv2 的建議路徑。

Console

將 IMDSv2 設定為指定區域的帳戶預設值

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 若要變更 AWS 區域，請使用頁面右上角的區域選擇器。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。
5. 在 IMDS 預設值旁邊，選擇管理。
6. 在管理 IMDS 預設值頁面上，執行下列操作：
   a. 在執行個體中繼資料服務中，選擇啟用。
   b. 針對 Metadata version (中繼資料版本)，選擇 V2 only (token required) (僅限 V2 (需要權杖))。
   c. 在中繼資料回應跳轉限制中，如果您的執行個體將託管容器，請指定 2。否則，請選取無偏好設定。如果未指定偏好設定，則在啟動時，AMI 具有設定 `ImdsSupport: v2.0` 的情況下預設值為 2；否則，預設值為 1。
   d. 選擇更新。

AWS CLI

將 IMDSv2 設定為指定區域的帳戶預設值

使用 modify-instance-metadata-defaults 命令，並指定要在其中修改 IMDS 帳戶層級設定的區域。如果您的執行個體將託管容器，請將 `--http-tokens` 設定為 `required` 並將 `--http-put-response-hop-limit` 設定為 2。否則，請指定 `-1` 以表示沒有偏好設定。如果指定 `-1` (無偏好設定)，則在啟動時，AMI 具有設定 `ImdsSupport: v2.0` 的情況下預設值為 2；否則，預設值為 1。

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
    --http-tokens required \
    --http-put-response-hop-limit 2
```

以下為範例輸出。

```
{
```

```
    "Return": true
}
```

針對指定區域檢視執行個體中繼資料選項的預設帳戶設定

使用 get-instance-metadata-defaults 命令並指定區域。

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

以下為範例輸出。

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 2
    },
    "ManagedBy": "account"
}
```

ManagedBy 欄位會指出進行相關設定的實體。在此範例中，account 會指出直接在帳戶中進行的設定。值為 declarative-policy 表示設定是透過宣告式政策來配置。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

將 IMDSv2 設定為所有區域的帳戶預設值

使用 modify-instance-metadata-defaults 命令，以修改所有區域的 IMDS 帳戶層級設定。如果您的執行個體將託管容器，請將 --http-tokens 設定為 required 並將 --http-put-response-hop-limit 設定為 2。否則，請指定 -1 以表示沒有偏好設定。如果指定 -1 (無偏好設定)，則在啟動時，AMI 具有設定 ImdsSupport: v2.0 的情況下預設值為 2；否則，預設值為 1。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
```

```
            --http-tokens required \
            --http-put-response-hop-limit 2 \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

以下為範例輸出。

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

針對所有區域檢視執行個體中繼資料選項的預設帳戶設定

使用 get-instance-metadata-defaults 命令。

```
echo -e "Region   \t Level          Hops    HttpTokens" ; \
echo -e "-------------- \t ------------   ----    ----------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下為範例輸出。

```
Region          Level          Hops    HttpTokens
--------------  ------------   ----    ----------
ap-south-1      ACCOUNTLEVEL   2       required
eu-north-1      ACCOUNTLEVEL   2       required
```

```
eu-west-3        ACCOUNTLEVEL    2       required
...
```

PowerShell

將 IMDSv2 設定為指定區域的帳戶預設值

使用 Edit-EC2InstanceMetadataDefault cmdlet，並指定要在其中修改 IMDS 帳戶層級設定的區域。如果您的執行個體將託管容器，請將 -HttpToken 設定為 required 並將 -HttpPutResponseHopLimit 設定為 2。否則，請指定 -1 以表示沒有偏好設定。如果指定 -1 (無偏好設定)，則在啟動時，AMI 具有設定 ImdsSupport: v2.0 的情況下預設值為 2；否則，預設值為 1。

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

以下為範例輸出。

```
True
```

針對指定區域檢視執行個體中繼資料選項的預設帳戶設定

使用 Get-EC2InstanceMetadataDefault cmdlet 並指定區域。

```
Get-EC2InstanceMetadataDefault -Region us-east-1 | Format-List
```

以下為範例輸出。

```
HttpEndpoint            :
HttpPutResponseHopLimit : 2
HttpTokens              : required
InstanceMetadataTags    :
```

將 IMDSv2 設定為所有區域的帳戶預設值

使用 Edit-EC2InstanceMetadataDefault cmdlet 修改所有區域的 IMDS 帳戶層級設定。如果您的執行個體將託管容器，請將 -HttpToken 設定為 required 並將 -HttpPutResponseHopLimit

設定為 2。否則，請指定 -1 以表示沒有偏好設定。如果指定 -1 (無偏好設定)，則在啟動時，AMI 具有設定 ImdsSupport: v2.0 的情況下預設值為 2；否則，預設值為 1。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

預期的輸出結果

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
...
```

針對所有區域檢視執行個體中繼資料選項的預設帳戶設定

使用 Get-EC2InstanceMetadataDefault cmdlet。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region = $_
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

範例輸出

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

在帳戶層級強制執行 IMDSv2

您可以在每個帳戶層級強制使用 IMDSv2 AWS 區域。強制執行時，執行個體只能在設定為需要 IMDSv2 時啟動。無論執行個體或 AMI 的設定方式為何，此強制執行都適用。

> **Note**
>
> 在帳戶層級啟用 IMDSv2 強制執行之前，請確定您的應用程式和 AMIs 支援 IMDSv2。如需詳細資訊，請參閱需要 IMDSv2 的建議路徑。如果已啟用 IMDSv2 強制執行`httpTokens`，且在啟動時執行個體組態、帳戶設定或 AMI 組態`required`中未設定為 ，則執行個體啟動將會失敗。如需故障診斷資訊，請參閱啟動IMDSv1-enabled的執行個體失敗。

> **Note**
>
> 此設定不會變更現有執行個體的 IMDS 版本，但會封鎖在目前已停用 IMDSv1 的現有執行個體上啟用 IMDSv1。

Console

為指定區域中的帳戶強制執行 IMDSv2

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 若要變更 AWS 區域，請使用頁面右上角的區域選擇器。
3. 在導覽窗格中，選擇 Dashboard (儀表板)。
4. 在帳戶屬性卡片的設定下，選擇資料保護和安全性。
5. 在 IMDS 預設值旁邊，選擇管理。
6. 在管理 IMDS 預設值頁面上，執行下列操作：

a. 針對 Metadata version (中繼資料版本)，選擇 V2 only (token required) (僅限 V2 (需要權杖))。

b. 針對強制 IMDSv2，選擇已啟用。

c. 選擇更新。

AWS CLI

為指定區域中的帳戶強制執行 IMDSv2

使用 modify-instance-metadata-defaults 命令，並指定要在其中強制執行 IMDSv2 的區域。

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
    --http-tokens required \
    --http-tokens-enforced enabled
```

以下為範例輸出。

```
{
"Return": true
}
```

檢視特定區域中帳戶的 IMDSv2 強制執行設定

使用 get-instance-metadata-defaults 命令並指定區域。

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

以下為範例輸出。

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpTokensEnforced": "enabled"
    },
    "ManagedBy": "account"
}
```

ManagedBy 欄位會指出進行相關設定的實體。在此範例中，account 會指出直接在帳戶中進行的設定。值為 declarative-policy 表示設定是透過宣告式政策來配置。如需詳細資訊，請參閱 AWS Organizations 使用者指南中的宣告政策。

為所有區域強制執行帳戶的 IMDSv2

使用 modify-instance-metadata-defaults 命令，在所有區域中強制執行 IMDSv2。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens-enforced enabled \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

以下為範例輸出。

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

檢視所有區域中帳戶的 IMDSv2 強制執行設定

使用 get-instance-metadata-defaults 命令。

```
echo -e "Region   \t Level           HttpTokensEnforced" ; \
echo -e "-------------- \t ------------   ----------------" ; \
for region in $(
```

```
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --query 'AccountLevel.HttpTokensEnforced' \
            --output text)
        echo -e "$region \t ACCOUNTLEVEL $output"
    );
done
```

以下為範例輸出。

```
Region           Level          HttpTokensEnforced
--------------   ------------   ------------------
ap-south-1       ACCOUNTLEVEL   enabled
eu-north-1       ACCOUNTLEVEL   enabled
eu-west-3        ACCOUNTLEVEL   enabled
...
```

PowerShell

為指定區域中的帳戶強制執行 IMDSv2

使用 Edit-EC2InstanceMetadataDefault cmdlet，並指定要在其中強制執行 IMDSv2 的區域。

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

以下為範例輸出。

```
@{
    Return = $true
}
```

檢視特定區域中帳戶的 IMDSv2 強制執行設定

使用 Get-EC2InstanceMetadataDefault 命令並指定區域。

```
Get-EC2InstanceMetadataDefault -Region us-east-1
```

以下為範例輸出。

```
@{
    AccountLevel = @{
        HttpTokens = "required"
        HttpTokensEnforced = "enabled"
    }
    ManagedBy = "account"
}
```

ManagedBy 欄位會指出進行相關設定的實體。在此範例中，account 會指出直接在帳戶中進行的設定。值為 declarative-policy 表示設定是透過宣告式政策來配置。如需詳細資訊，請參閱 AWS Organizations 使用者指南中的宣告政策。

為所有區域強制執行帳戶的 IMDSv2

使用 modify-instance-metadata-defaults 命令，在所有區域中強制執行 IMDSv2。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens-enforced enabled \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

以下為範例輸出。

```
Region                  Modified
```

```
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

將 IMDSv2 設定為所有區域的帳戶預設值

使用 Edit-EC2InstanceMetadataDefault cmdlet 修改所有區域的 IMDS 帳戶層級設定。如果您的執行個體將託管容器，請將 `-HttpToken` 設定為 `required` 並將 `-HttpPutResponseHopLimit` 設定為 2。否則，請指定 `-1` 以表示沒有偏好設定。如果指定 `-1` (無偏好設定)，則在啟動時，AMI 具有設定 `ImdsSupport: v2.0` 的情況下預設值為 2；否則，預設值為 1。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

預期的輸出結果

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
...
```

針對所有區域檢視執行個體中繼資料選項的預設帳戶設定

使用 Get-EC2InstanceMetadataDefault cmdlet。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
```

```
        Region = $_
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

範例輸出

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

啟動時設定執行個體

啟動執行個體時，您可以設定下列欄位，將執行個體設定為需要使用 IMDSv2：

- Amazon EC2 主控台：將 Metadata version (中繼資料版本) 設為 V2 only (token required) (僅 V2 (需要權杖))。
- AWS CLI：將 `HttpTokens` 設為 `required`。

當您指定需要 IMDSv2 時，也必須透過將可存取中繼資料設定為已啟用 (主控台) 或將 `HttpEndpoint` 設為 `enabled` (AWS CLI) 來啟用執行個體中繼資料服務 (IMDS) 端點。

在容器環境中，如果需要 IMDSv2，建議將跳轉限制設定為 2。如需詳細資訊，請參閱執行個體中繼資料存取考量。

Console

要求在新執行個體上使用 IMDSv2

- 在 Amazon EC2 主控台中啟動新執行個體時，展開 Advanced details (進階詳細資訊)，並執行下列動作：
  - 針對 Metadata accessible (可存取中繼資料)，選擇 Enabled (已啟用)。

- 針對 Metadata version (中繼資料版本)，選擇 V2 only (token required) (僅限 V2 (需要權杖))。
- (容器環境) 在中繼資料回應跳轉限制中，選擇 2。

如需詳細資訊，請參閱進階詳細資訊。

AWS CLI

要求在新執行個體上使用 IMDSv2

下列 run-instances 範例會啟動 c6i.large 設定為 --metadata-options 的HttpTokens=required 執行個體。當您為 HttpTokens 指定值時，您也必須將 HttpEndpoint 設定為 enabled。由於中繼資料擷取請求的安全權杖標頭設定為 required，因此在要求執行個體中繼資料時需要執行個體使用 IMDSv2。

在容器環境中，如果需要 IMDSv2，建議使用 HttpPutResponseHopLimit=2 將跳轉限制設定為 2。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
 ...
    --metadata-options
 "HttpEndpoint=enabled,HttpTokens=required,HttpPutResponseHopLimit=2"
```

PowerShell

要求在新執行個體上使用 IMDSv2

下列 New-EC2Instance cmdlet 範例會啟動c6i.large執行個體enabled，並將 MetadataOptions_HttpTokens 參數MetadataOptions_HttpEndpoint設定為 required。當您為 HttpTokens 指定值時，您也必須將 HttpEndpoint 設定為 enabled。由於中繼資料擷取請求的安全權杖標頭設定為 required，因此在要求執行個體中繼資料時需要執行個體使用 IMDSv2。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
```

```
    -MetadataOptions_HttpTokens required
```

CloudFormation

若要使用 指定執行個體的中繼資料選項 CloudFormation，請參閱AWS CloudFormation 《 使用者指南》中的 AWS::EC2::LaunchTemplate MetadataOptions 屬性。

設定 AMI

註冊新 AMI 或修改現有 AMI 時，可以將 `imds-support` 參數設定為 `v2.0`。從此 AMI 啟動的執行個體會將中繼資料版本設定為僅 V2 (需要權杖) (主控台) 或將 `HttpTokens` 設為 `required` (AWS CLI)。透過這些設定，執行個體要求在請求執行個體中繼資料時使用 IMDSv2。

請注意，當您將 `imds-support` 設為 `v2.0` 時，從這個 AMI 啟動的執行個體也將有 Metadata response hop limit (中繼資料回應跳轉限制) (主控台) 或將 `http-put-response-hop-limit` (AWS CLI) 設為 2。

> **Important**
>
> 除非 AMI 軟體支援 IMDSv2，否則請勿使用此參數。將值設為 `v2.0` 後，將無法復原。「重設」AMI 的唯一方法是從基礎快照中建立一個新的 AMI。

設定適用於 IMDSv2 的新 AMI

使用下列其中一種方法來為 IMDSv2 設定新的 AMI。

AWS CLI

以下 register-image 範例使用指定的 EBS 根磁碟區快照作為裝置 `/dev/xvda` 來註冊 AMI。為 `imds-support` 參數指定 `v2.0`，以便從此 AMI 啟動的執行個體在請求執行個體中繼資料時要求使用 IMDSv2。

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0123456789example} \
    --architecture x86_64 \
```

```
    --imds-support v2.0
```

PowerShell

下列 Register-EC2Image cmdlet 範例會使用 EBS 根磁碟區的指定快照，將 AMI 註冊為裝置 /dev/xvda。為 ImdsSupport 參數指定 v2.0，以便從此 AMI 啟動的執行個體在請求執行個體中繼資料時要求使用 IMDSv2。

```
Register-EC2Image `
    -Name 'my-image' `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping  (
    New-Object `
        -TypeName Amazon.EC2.Model.BlockDeviceMapping `
        -Property @{
        DeviceName = '/dev/xvda';
        EBS        = (New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property
 @{
                SnapshotId = 'snap-0123456789example'
                VolumeType = 'gp3'
                } )
        }  ) `
    -Architecture X86_64 `
    -ImdsSupport v2.0
```

設定適用於 IMDSv2 的現有 AMI

使用下列其中一種方法來為 IMDSv2 設定現有的 AMI。

AWS CLI

下列 modify-image-attribute 範例只會修改 IMDSv2 的現有 AMI。為 imds-support 參數指定 v2.0，以便從此 AMI 啟動的執行個體在請求執行個體中繼資料時要求使用 IMDSv2。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --imds-support v2.0
```

PowerShell

下列 Edit-EC2ImageAttribute cmdlet 範例只會修改 IMDSv2 的現有 AMI。為 imds-support 參數指定 v2.0，以便從此 AMI 啟動的執行個體在請求執行個體中繼資料時要求使用 IMDSv2。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ImdsSupport 'v2.0'
```

使用 IAM 政策

您可以建立執行下列其中一項的 IAM 政策：

- 防止使用者啟動新執行個體，除非他們在新執行個體上需要 IMDSv2。
- 防止使用者呼叫 ModifyInstanceMetadataOptions API 來變更執行中執行個體的中繼資料選項。限制對 ModifyInstanceMetadataOptions httpTokens 屬性的存取，以防止執行中執行個體的意外更新。
- 防止使用者呼叫 ModifyInstanceMetadataDefaults API 來變更 httpTokens 和 httpTokensEnforced 的帳戶預設設定。限制存取這兩個屬性可確保只有授權的角色才能修改帳戶預設值。

透過使用 IAM 政策，在所有新執行個體上強制使用 IMDSv2

若要確保使用者在請求執行個體中繼資料時只能啟動需要使用 IMDSv2 的執行個體，請執行下列動作：

- 限制對 `ModifyInstanceMetadataOptions`和 `ModifyInstanceMetadataDefaults` API 的存取，特別是 `httpTokens`和 `httpTokensEnforced` 屬性。
- 然後，將帳戶預設為 `httpTokens = required`和 `httpTokensEnforced = enabled`。

  如需 IAM 政策範例，請參閱使用執行個體中繼資料。

啟用 IMDS IPv4 和 IPv6 端點

IMDS 在執行個體上有兩個端點：IPv4 (`169.254.169.254`) 和 IPv6 (`[fd00:ec2::254]`)。當您啟用 IMDS 時，IPv4 端點會自動啟用。即使您在僅限 IPv6 的子網路中啟動執行個體，IPv6 端點仍會保持停用狀態。若要啟用 IPv6 端點，則需要明確啟用。當您啟用 IPv6 端點時，IPv4 端點會保持啟用狀態。

您可以在執行個體啟動時或啟動之後啟用 IPv6 端點。

啟用 IPv6 端點的需求

- 選取的執行個體類型是 Nitro 型執行個體。

- 選取的子網路支援 IPv6，其中子網路僅支援雙重堆疊或 IPv6。

使用下列任一種方法來啟動已啟用 IMDS IPv6 端點的執行個體。

Console

在執行個體啟動時啟用 IMDS IPv6 端點

- 在 Amazon EC2 主控台中啟動執行個體，並在 Advanced details (進階詳細資訊) 下指定以下內容：
  - 在中繼資料 IPv6 端點中，選擇已啟用。

如需詳細資訊，請參閱進階詳細資訊。

AWS CLI

在執行個體啟動時啟用 IMDS IPv6 端點

以下 run-instances 範例會啟動已啟用執行個體中繼資料服務 IPv6 端點的 c6i.large 執行個體。若要啟用 IPv6 端點，對於 --metadata-options 參數，請指定 HttpProtocolIpv6=enabled。當您為 HttpProtocolIpv6 指定值時，您也必須將 HttpEndpoint 設定為 enabled。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=enabled,HttpProtocolIpv6=enabled"
```

PowerShell

在執行個體啟動時啟用 IMDS IPv6 端點

下列 New-EC2Instance cmdlet 範例會啟動c6i.large執行個體，並為 IMDS 啟用 IPv6 端點。若要啟用 IPv6 端點，請將 MetadataOptions_HttpProtocolIpv6 指定為 enabled。當您為 MetadataOptions_HttpProtocolIpv6 指定值時，您也必須將 MetadataOptions_HttpEndpoint 設定為 enabled。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
```

```
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpProtocolIpv6 enabled
```

關閉對執行個體中繼資料的存取

可以在啟動執行個體時停用 IMDS 來關閉對執行個體中繼資料的存取。稍後透過重新啟用 IMDS 可以開啟存取。如需詳細資訊，請參閱開啟對執行個體中繼資料的存取。

> **Important**
>
> 可以選擇在啟動時或啟動後停用 IMDS。如果在啟動時停用 IMDS，下列項目可能無法運作：
>
> - 您可能沒有執行個體的 SSH 存取權。public-keys/0/openssh-key 是執行個體的公有 SSH 金鑰，它將不可存取，因為通常會從 EC2 執行個體中繼資料中提供並存取該金鑰。
> - EC2 使用者資料將無法使用，且不會在執行個體啟動時執行。EC2 使用者資料託管於 IMDS。如果停用 IMDS，可以有效地關閉對使用者資料的存取。
>
> 若要存取此功能，可以在啟動後重新啟用 IMDS。

Console

啟動時關閉對執行個體中繼資料的存取

- 在 Amazon EC2 主控台中啟動執行個體，並在 Advanced details (進階詳細資訊) 下指定以下內容：
  - 針對 Metadata accessible (可存取中繼資料)，選擇 Disabled (已停用)。

如需詳細資訊，請參閱進階詳細資訊。

AWS CLI

啟動時關閉對執行個體中繼資料的存取

將 `--metadata-options` 設為 `HttpEndpoint=disabled` 來啟動執行個體。

```
aws ec2 run-instances \
```

```
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=disabled"
```

PowerShell

啟動時關閉對執行個體中繼資料的存取

下列 New-EC2Instance cmdlet 範例會啟動將 MetadataOptions_HttpEndpoint 設為 的執行個體disabled。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint disabled
```

CloudFormation

若要使用 指定執行個體的中繼資料選項 CloudFormation，請參閱CloudFormation 《 使用者指南》中的 AWS::EC2::LaunchTemplate MetadataOptions 屬性。

允許存取執行個體中繼資料中的標籤

根據預設，無法存取執行個體中繼資料中的執行個體標籤。對於每個執行個體，您必須明確允許存取。如果允許存取，則執行個體標籤金鑰必須符合特定字元限制，否則執行個體將無法啟動。如需詳細資訊，請參閱在執行個體中繼資料中啟用標籤存取權。

修改現有執行個體的執行個體中繼資料選項

您可以修改現有執行個體的執行個體中繼資料選項。

您也可以建立 IAM 政策，以防止使用者修改現有執行個體上的執行個體中繼資料選項。若要控制哪些使用者可以修改執行個體中繼資料選項，請指定一個政策，以防止具有指定角色的使用者以外的所有使用者使用 ModifyInstanceMetadataOptions API。如需 IAM 政策範例，請參閱使用執行個體中繼資料。

> **Note**
>
> 如果使用宣告式政策來設定執行個體中繼資料選項，則您無法直接在帳戶內進行修改。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

需要使用 IMDSv2

使用下列其中一種方法來修改現有執行個體上的執行個體中繼資料選項，以便在請求執行個體中繼資料時要求 IMDSv2。當要求 IMDSv2 時，就無法使用 IMDSv1。

> **Note**
>
> 在要求使用 IMDSv2 之前，請確定執行個體並未進行 IMDSv1 呼叫。`MetadataNoToken` CloudWatch 指標會追蹤 IMDSv1 呼叫。當 `MetadataNoToken` 將執行個體的 IMDSv1 使用量記錄為零時，則執行個體準備好需要 IMDSv2。

Console

要求在現有執行個體上使用 IMDSv2

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體。
4. 選擇動作、執行個體設定、修改執行個體中繼資料選項。
5. 在修改執行個體中繼資料選項對話方塊中，執行下列動作：
   a. 針對執行個體中繼資料服務，請選取啟用。
   b. 針對 IMDSv2，請選擇必要。
   c. 選擇儲存。

AWS CLI

要求在現有執行個體上使用 IMDSv2

使用 modify-instance-metadata-options CLI 命令，並將 `http-tokens` 參數設定為 `required`。當您為 `http-tokens` 指定值時，您也必須將 `http-endpoint` 設定為 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

PowerShell

要求在現有執行個體上使用 IMDSv2

使用 Edit-EC2InstanceMetadataOption cmdlet，並將 HttpTokens 參數設定為 required。當您為 HttpTokens 指定值時，您也必須將 HttpEndpoint 設定為 enabled。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens required `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

還原使用 IMDSv1

當執行個體需要 IMDSv2 時，使用 IMDSv1 請求將會失敗。當 IMDSv2 為選用時，IMDSv2 和 IMDSv1 都可以運作。因此，若要還原 IMDSv1，請使用下列其中一種方法將 IMDSv2 設定為選用 (httpTokens = optional)。

httpTokensEnforced IMDS 屬性也會防止嘗試在現有執行個體上啟用 IMDSv1。為區域中的帳戶啟用 時，嘗試httpTokens將 設定為 optional會導致UnsupportedOperation例外狀況。如需詳細資訊，請參閱 疑難排解。

> **Important**
>
> 如果您的執行個體啟動因 IMDSv2 強制執行而失敗，您有兩個選項可以讓啟動成功：
>
> - 將執行個體啟動為IMDSv2-only – 如果在執行個體上執行的軟體僅使用 IMDSv2 （不依賴 IMDSv1)，則您只能將執行個體啟動為僅限 IMDSv2。若要執行此作業，請在啟動參數httpTokens = required中設定 IMDSv2，或在區域中帳戶的中繼資料預設值中設定 IMDSv2。
> - 停用強制執行 – 如果您的軟體仍然依賴 IMDSv1，disabled請將 區域中帳戶的 httpTokensEnforced設為 。如需詳細資訊，請參閱在帳戶層級強制執行 IMDSv2。

Console

在執行個體上還原使用 IMDSv1

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體。
4. 選擇動作、執行個體設定、修改執行個體中繼資料選項。
5. 在修改執行個體中繼資料選項對話方塊中，執行下列動作：

   a. 針對執行個體中繼資料服務，請確定已選取啟用。

   b. 針對 IMDSv2，請選擇選用。

   c. 選擇儲存。

AWS CLI

在執行個體上還原使用 IMDSv1

您可使用 modify-instance-metadata-options CLI 命令並將 `http-tokens` 設定為 `optional`，以在要求執行個體中繼資料時還原使用 IMDSv1。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens optional \
    --http-endpoint enabled
```

PowerShell

在執行個體上還原使用 IMDSv1

您可以將 Edit-EC2InstanceMetadataOption cmdlet 與 `HttpTokens` 設定為 搭配使用，`optional`以在請求執行個體中繼資料時還原 IMDSv1 的使用。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens optional `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

變更 PUT 回應跳轉限制

對於現有執行個體，您可以變更 PUT 回應跳轉限制的設定。

目前只有 AWS CLI 和 AWS SDKs 支援變更 PUT 回應跳轉限制。

AWS CLI

如何變更 PUT 回應跳轉限制

使用 modify-instance-metadata-options CLI 命令，並將 `http-put-response-hop-limit` 參數設定為必要的跳轉數目。在下列範例中，跳轉限制設定為 3。請注意，為 `http-put-response-hop-limit` 指定數值時，您還必須將 `http-endpoint` 設定為 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-put-response-hop-limit 3 \
    --http-endpoint enabled
```

PowerShell

如何變更 PUT 回應跳轉限制

使用 Edit-EC2InstanceMetadataOption cmdlet，並將 `HttpPutResponseHopLimit` 參數設定為所需的跳轉次數。在下列範例中，跳轉限制設定為 3。請注意，為 `HttpPutResponseHopLimit` 指定數值時，您還必須將 `HttpEndpoint` 設定為 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpPutResponseHopLimit 3 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

啟用 IMDS IPv4 和 IPv6 端點

IMDS 在執行個體上有兩個端點：IPv4 (`169.254.169.254`) 和 IPv6 (`[fd00:ec2::254]`)。當您啟用 IMDS 時，IPv4 端點會自動啟用。即使您在僅限 IPv6 的子網路中啟動執行個體，IPv6 端點仍會保持停用狀態。若要啟用 IPv6 端點，則需要明確啟用。當您啟用 IPv6 端點時，IPv4 端點會保持啟用狀態。

您可以在執行個體啟動時或啟動之後啟用 IPv6 端點。

啟用 IPv6 端點的需求

- 選取的執行個體類型是 Nitro 型執行個體。
- 選取的子網路支援 IPv6，其中子網路僅支援雙重堆疊或 IPv6。

目前只有 AWS CLI 和 AWS SDKs 支援在執行個體啟動後啟用 IMDS IPv6 端點。

AWS CLI

為您的執行個體啟用 IMDS IPv6 端點

使用 modify-instance-metadata-options CLI 命令，並將 `http-protocol-ipv6` 參數設定為 `enabled`。請注意，為 `http-protocol-ipv6` 指定數值時，您還必須將 `http-endpoint` 設定為 `enabled`。

```
aws ec2 modify-instance-metadata-options \
 --instance-id i-1234567890abcdef0 \
 --http-protocol-ipv6 enabled \
 --http-endpoint enabled
```

PowerShell

為您的執行個體啟用 IMDS IPv6 端點

使用 Edit-EC2InstanceMetadataOption cmdlet，並將 `HttpProtocolIpv6` 參數設定為 `enabled`。請注意，為 `HttpProtocolIpv6` 指定數值時，您還必須將 `HttpEndpoint` 設定為 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpProtocolIpv6 enabled `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

開啟對執行個體中繼資料的存取

無論使用哪個版本的 IMDS，都可以啟用執行個體上 IMDS 的 HTTP 端點，來開啟對執行個體中繼資料的存取。您可以停用 HTTP 端點，以隨時反轉這項變更。

使用下列其中一種方法來開啟對執行個體上執行個體中繼資料的存取。

Console

開啟對執行個體中繼資料的存取

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。

3. 選取執行個體。
4. 選擇動作、執行個體設定、修改執行個體中繼資料選項。
5. 在修改執行個體中繼資料選項對話方塊中，執行下列動作：

   a. 針對執行個體中繼資料服務，請選取啟用。

   b. 選擇儲存。

AWS CLI

開啟對執行個體中繼資料的存取

使用 modify-instance-metadata-options CLI 命令，並將 `http-endpoint` 參數設定為 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint enabled
```

PowerShell

開啟對執行個體中繼資料的存取

使用 Edit-EC2InstanceMetadataOption cmdlet，並將 `HttpEndpoint` 參數設定為 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

關閉對執行個體中繼資料的存取

無論使用哪個版本的 IMDS，都可以停用執行個體上 IMDS 的 HTTP 端點，來關閉對執行個體中繼資料的存取。您可以啟用 HTTP 端點，以隨時反轉這項變更。

使用下列其中一種方法來關閉對執行個體上執行個體中繼資料的存取。

Console

關閉對執行個體中繼資料的存取

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。

3. 選取執行個體。
4. 選擇動作、執行個體設定、修改執行個體中繼資料選項。
5. 在修改執行個體中繼資料選項對話方塊中，執行下列動作：
   a. 針對執行個體中繼資料服務，清除啟用。
   b. 選擇儲存。

AWS CLI

關閉對執行個體中繼資料的存取

使用 modify-instance-metadata-options CLI 命令，並將 `http-endpoint` 參數設定為 `disabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint disabled
```

PowerShell

關閉對執行個體中繼資料的存取

使用 Edit-EC2InstanceMetadataOption cmdlet，並將 `HttpEndpoint` 參數設定為 `disabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint disabled).InstanceMetadataOptions
```

允許存取執行個體中繼資料中的標籤

您可以允許存取執行中或已停止執行個體上執行個體中繼資料中的標籤。對於每個執行個體，您必須明確允許存取。如果允許存取，則執行個體標籤金鑰必須符合特定字元限制，否則將收到錯誤訊息。如需詳細資訊，請參閱在執行個體中繼資料中啟用標籤存取權。

疑難排解

修改IMDSv1-enabled的執行個體失敗

說明

您會收到下列錯誤訊息：

```
You can't launch instances with IMDSv1 because httpTokensEnforced
is enabled for this account. Either launch the instance with
httpTokens=required or contact your account owner to disable
httpTokensEnforced using the ModifyInstanceMetadataDefaults API or the
account settings in the EC2 console.
```

原因

當您嘗試在 EC2 帳戶設定或 AWS Organization 宣告政策強制使用 IMDSv2 () 的帳戶中將現有執行個體修改為啟用 IMDSv1 () 時，就會擲出此錯誤`httpTokensEnforced = enabled`。`httpTokens = optional` IMDSv2

解決方案

如果您在現有執行個體上需要 IMDSv1 支援，則需要停用 區域中帳戶的 IMDSv2 強制執行。若要停用 IMDSv2 強制執行，請將 `HttpTokensEnforced`設定為 `disabled`。如需詳細資訊，請參閱《Amazon EC2 API 參考》中的 ModifyInstanceMetadataDefaults。如果您偏好使用 主控台設定此設定，請參閱 在帳戶層級強制執行 IMDSv2。

我們建議您僅使用 IMDSv2 (`httpTokens=required`)。如需詳細資訊，請參閱轉換為使用 執行個體中繼資料服務第 2 版。

當使用使用者資料輸入啟動 EC2 執行個體時執行命令

當您啟動 Amazon EC2 執行個體時，您可以將使用者資料傳遞到用於執行自動化設定任務或在執行個體啟動後執行指令碼的執行個體。

如果您對更複雜的自動化案例感興趣，可以考慮 CloudFormation。如需詳細資訊，請參閱「AWS CloudFormation 使用者指南」中的使用 CloudFormation在 Amazon EC2 部署應用程式。

在 Linux 執行個體，您可以將兩種類型的使用者資料傳遞給 Amazon EC2：shell 指令碼和 cloud-init 指示詞。您也可以將這項資料以純文字、檔案 (在使用命令列工具啟動執行個體時很有用) 或 Base64 編碼文字 (適用於 API 呼叫) 傳遞給啟動執行個體精靈。

在 Windows 執行個體，啟動代理程式會處理您的使用者資料指令碼。

考量事項

- 使用者資料或會做為不透明資料處理：您給予的即為您取回的。是否要解譯則取決於執行個體。

- 使用者資料必須為 base64 編碼形式。Amazon EC2 主控台可為您執行 base64 編碼，或是接受 base64 編碼輸入。如果您使用執行個體中繼資料或主控台擷取使用者資料，將會自動為您進行 base64 解碼。
- 使用者資料在 base64 編碼之前限制為 16 KB (原始形式)。長度為 n 的字串在 base64 編碼之後的大小為 ceil(n/3)*4。
- 使用者資料是執行個體屬性。如果您從執行個體建立 AMI，執行個體使用者資料不會包含在 AMI 中。

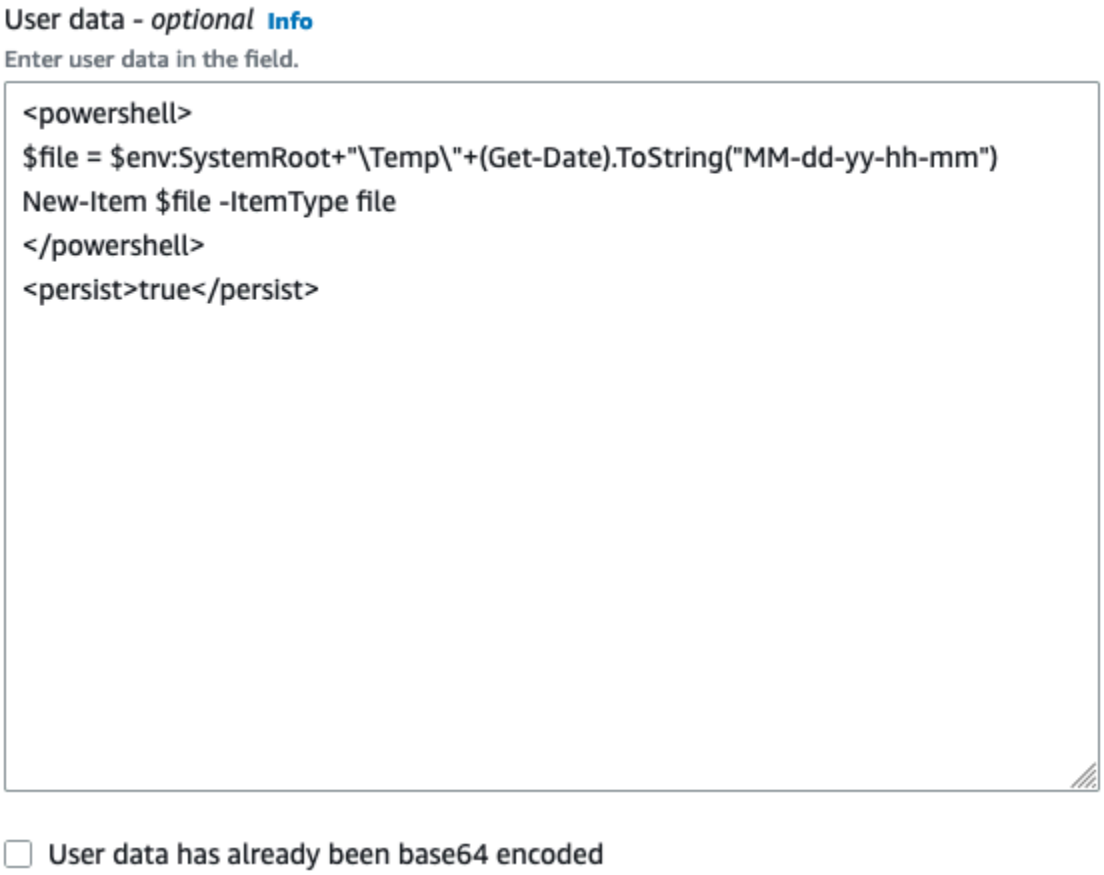
中的使用者資料 AWS 管理主控台

您可以在啟動執行個體時指定執行個體使用者資料。若執行個體的根磁碟區為 EBS 磁碟區，您也可以停止執行個體並更新其使用者資料。

使用啟動精靈在啟動時指定執行個體使用者資料

您可以在使用 EC2 主控台的啟動精靈啟動執行個體時指定使用者資料。若要在啟動時指定使用者資料，請遵循啟動執行個體的程序。此 User data (使用者資料) 欄位位於啟動執行個體精靈的 進階詳細資訊 區段中。在使用者資料欄位輸入 PowerShell 指令碼，然後完成執行個體啟動程序。

在以下使用者資料欄位的螢幕擷取畫面中，範例指令碼會在 Windows 暫時資料夾中建立一個檔案，並在檔名中使用目前日期和時間。當您包括 `<persist>true</persist>` 時，指令碼會在您每次重新開機或啟動執行個體時執行。如果您將使用者資料已使用 base64 編碼核取方塊留白，Amazon EC2 主控台會為您執行 base64 編碼。



如需詳細資訊，請參閱使用啟動精靈在啟動時指定執行個體使用者資料。如需使用 的 Linux 範例 AWS CLI，請參閱 the section called “使用者資料和 AWS CLI”。如需使用 Tools for Windows PowerShell 的 Windows 範例，請參閱 the section called “使用者資料與 Tools for Windows PowerShell”。

檢視及更新執行個體使用者資料

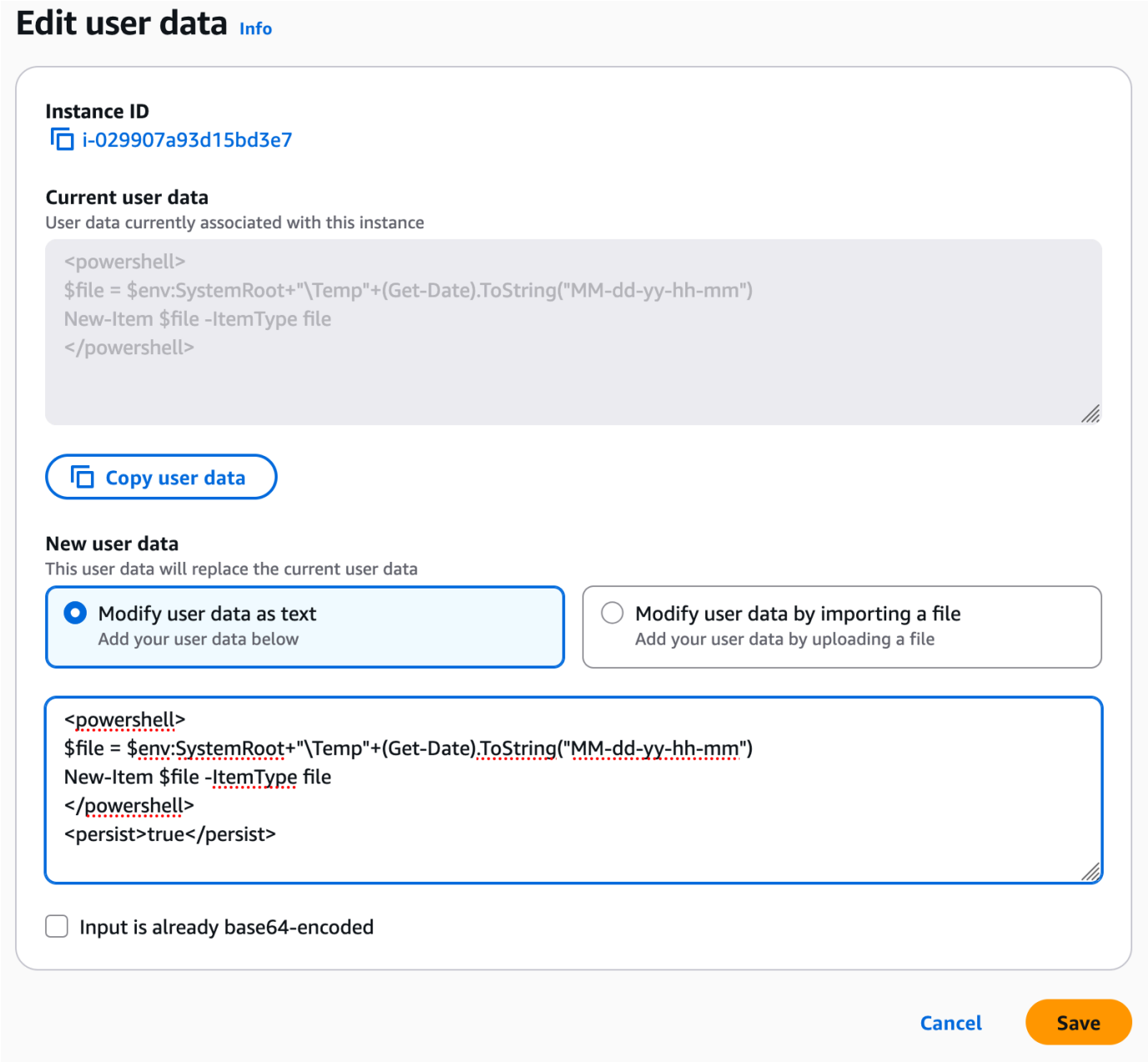
您可以檢視任何執行個體的執行個體使用者資料，並更新已終止之執行個體的執行個體使用者資料。

使用主控台更新執行個體的使用者資料

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，並選取 Actions (動作)、Instance state (執行個體狀態)、Stop instance (停止執行個體)。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

4. 出現確認提示時，請選擇 Stop (停止)。停止執行個體可能需要幾分鐘。
5. 在仍然選取執行個體的情況下，選取動作 > 執行個體設定 > 編輯使用者資料。您無法在執行個體仍在執行中時變更使用者資料，但您可以檢視它。
6. 在 Edit user data (編輯使用者資料) 對話方塊中，更新使用者資料，然後選擇 Save (儲存)。若要在每次重新開機或啟動執行個體時執行使用者資料指令碼，請新增 `<persist>true</persist>`，如下範例所示：



7. 啟動實例。若您已為後續的重新開機或啟動啟用使用者資料執行，即會在執行個體啟動程序期間執行更新的使用者資料指令碼。

Amazon EC2 如何處理 Linux 執行個體的使用者資料

以下範例利用使用者資料來執行命令，可在啟用執行個體時設定 LAMP 伺服器。在每個範例中，都會執行以下任務：

- 更新分佈軟體套件。

- 安裝 Web 伺服器、php 和 mariadb 套件。
- 啟動並開啟 httpd 服務。
- 使用者 ec2-user 即會新增至 apache 群組。
- 接著便會為 Web 目錄和其中包含的檔案設定適當的所有權和檔案許可。
- 一的簡單的網頁便會建立，用於測試 Web 伺服器和 PHP 引擎。

目錄

- 先決條件
- 使用者資料與 Shell 指令碼
- 更新執行個體使用者資料
- 使用者資料與 cloud-init 指示詞
- 使用者資料和 AWS CLI
- 結合 shell 指令碼與 cloud-init 指令

先決條件

此主題中的範例假設如下內容：

- 您的執行個體具有公有 DNS 名稱，且可從網際網路連線。
- 與執行個體相關聯的安全群組已設定為允許 SSH (連接埠 22) 流量，以便您可以連線到執行個體以檢視輸出日誌檔案。
- 您的執行個體使用 Amazon Linux 2 AMI 啟動。命令與指令可能不適用於其他 Linux 發行版本。如需其他發行版本的詳細資訊，例如支援 cloud-init 的情形，請參閱特定發行版本的相關文件。

使用者資料與 Shell 指令碼

如果您熟悉 Shell 指令碼，這是在啟動時將指令傳送至執行個體的最簡單且最完整的方式。在開機階段新增這些任務也會增加開機執行個體所需要的時間長度。任務完成需要多花幾分鐘的時間，接著您便可以測試使用者指令碼是否已成功完成。

> **Important**
>
> 依預設，使用者資料指令碼和 cloud-init 指令只會在您第一次啟動執行個體時的開機週期過程中執行。您可以更新設定，以確保每次重新啟動執行個體時，您的使用者資料指令碼和 cloud-

init 指令都會執行。如需詳細資訊，請參閱如何利用使用者資料，在每次重新啟動 Amazon EC2 Linux 執行個體時自動執行指令碼？ AWS 知識中心的 。

使用者資料 shell 指令碼必須以 #! 字元，以及您希望讀取指令碼 (通常是 /bin/bash)) 的解譯器路徑做為開頭。如需 shell 指令碼的介紹，請參閱 GNU 作業系統網站的 Bash 參考手冊。

做為使用者資料輸入的指令碼會以根使用者的身分執行，因此請不要在指令碼中使用 sudo 命令。請記得，您建立的任何檔案都會由根使用者擁有；若您需要讓非超級使用者擁有檔案存取權，建議您根據需求在指令碼中修改許可。此外，因為指令碼不會以互動方式執行，您無法包含需要使用者意見回饋的命令 (例如不帶有 -y 標記的 yum update)。

如果您在使用者資料指令碼中使用 AWS API，包括 AWS CLI，則必須在啟動執行個體時使用執行個體描述檔。執行個體描述檔提供使用者資料指令碼發出 API 呼叫所需的適當 AWS 登入資料。如需詳細資訊，請參閱《IAM 使用者指南》中的使用執行個體設定檔。您指派給 IAM 角色的許可取決於您使用 API 呼叫的服務。如需詳細資訊，請參閱Amazon EC2 的 IAM 的角色。

cloud-init 輸出日誌檔案會擷取主控台輸出，因此，若執行個體在啟動後未以您想要的方式運作，則可輕鬆地對您的指令碼進行除錯。若要檢視日誌檔案，連線到執行個體並開啟 `/var/log/cloud-init-output.log`。

處理使用者資料指令碼時，會將其複製到 `/var/lib/cloud/instances/instance-id/` 並從中執行。在執行指令碼之後，不會將其刪除。務必從 `/var/lib/cloud/instances/instance-id/` 中刪除使用者資料指令碼，然後再從執行個體中建立 AMI。否則，從 AMI 啟動任何執行個體時，指令碼將存在於這個目錄中。

更新執行個體使用者資料

若要更新執行個體使用者資料，您必須先停止執行個體。如果執行個體正在執行，您可以檢視使用者資料，但無法進行修改。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

修改執行個體使用者資料

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，並選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。如果此選項已停用，則執行個體已停止或其根磁碟區是執行個體儲存體磁碟區。
4. 出現確認提示時，請選擇 Stop (停止)。停止執行個體可能需要幾分鐘。
5. 在仍然選取執行個體的情況下，選取動作 > 執行個體設定 > 編輯使用者資料。
6. 視需要修改使用者資料，然後選擇儲存。
7. 啟動實例。新的使用者資料便會在執行個體啟動之後出現在執行個體上；但不會執行使用者資料指令碼。

使用者資料與 cloud-init 指示詞

cloud-init 套件會在啟動時設定新 Amazon Linux 執行個體的特定部分；最明顯的是，它會設定 ec2-user 的 `.ssh/authorized_keys` 檔案，讓您可以使用自己的私有金鑰登入。如需有關 cloud-init 套件為 Amazon Linux 執行個體執行的組態任務的詳細資訊，請參閱下列文件：

- Amazon Linux 2023 – 自訂 cloud-init
- Amazon Linux 2 – 使用 Amazon Linux 2 上的 cloud-init

cloud-init 使用者指示詞可在啟動時，以傳遞指令碼相同的方式傳遞到執行個體，雖然語法不同。如需 cloud-init 的詳細資訊，請參閱 https：//https://cloudinit.readthedocs.org/en/latest/index.html。

> **Important**
>
> 依預設，使用者資料指令碼和 cloud-init 指令只會在您第一次啟動執行個體時的開機週期過程中執行。您可以更新設定，以確保每次重新啟動執行個體時，您的使用者資料指令碼和 cloud-init 指令都會執行。如需詳細資訊，請參閱如何利用使用者資料在每次重新啟動 Amazon EC2 Linux 執行個體時自動執行指令碼？ AWS 知識中心的 。

在開機階段新增這些任務也會增加開機執行個體所需要的時間長度。任務完成需要多花幾分鐘的時間，接著您便可以測試您的使用者資料指示詞是否已完成。

如需傳遞 cloud-init 指令至 Amazon Linux 執行個體

1. 請遵循啟動執行個體的程序。此 User data (使用者資料) 欄位位於啟動執行個體精靈的 進階詳細資訊 區段中。在 User data (使用者資料) 欄位中輸入您的 cloud-init 指示詞文字，然後完成執行個體啟動程序。

在以下範例中，指令會在 Amazon Linux 上建立及設定 Web 伺服器。需要最上方的#cloud-config行，才能將命令識別為 cloud-init 指令。

AL2023

```
#cloud-config
package_update: true
package_upgrade: all

packages:
- httpd
- mariadb105-server
- php8.1
- php8.1-mysqlnd

runcmd:
- systemctl start httpd
- systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

AL2

```
#cloud-config
package_update: true
package_upgrade: all

packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
- systemctl start httpd
- systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
```

```
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. 給予執行個體充分的時間啟動及執行您使用者資料中的指示詞，然後檢查您的指示詞是否已完成您預計執行的任務。

   針對此範例，在 Web 瀏覽器中，輸入指示詞建立之 PHP 測試檔案的 URL。此 URL 為您執行個體的公有 DNS 地址，其後跟隨斜線和檔案名稱。

   ```
   http://my.public.dns.amazonaws.com/phpinfo.php
   ```

   您現在應該會看見 PHP 資訊頁面。如果您未看見 PHP 資訊頁面，請檢查您使用的安全群組是否包含允許 HTTP (連接埠 80) 流量的規則。如需詳細資訊，請參閱設定安全群組規則。

3. (選用) 若您的指令並未完成您預期執行的任務，或您只希望確認您的指令已順利完成，而沒有發生任何錯誤，請連線到執行個體，檢查 cloud-init 輸出日誌檔案 (`/var/log/cloud-init-output.log`)，並在輸出中查看是否有錯誤訊息。如需其他偵錯資訊，您可以為您的指示詞新增下行：

   ```
   output : { all : '| tee -a /var/log/cloud-init-output.log' }
   ```

   這個指示詞會將 runcmd 的輸出傳送到 `/var/log/cloud-init-output.log`。

使用者資料和 AWS CLI

您可以使用 AWS CLI 來指定、修改和檢視執行個體的使用者資料。如需使用執行個體中繼資料檢視您執行個體中使用者資料的資訊，請參閱存取 EC2 執行個體的執行個體中繼資料。

在 Windows 上，您可以使用 ， AWS Tools for Windows PowerShell 而不是使用 AWS CLI。如需詳細資訊，請參閱 使用者資料與 Tools for Windows PowerShell。

範例：在啟動時指定使用者資料

若要在啟動您的執行個體時指定使用者資料，請搭配 `--user-data` 參數使用 run-instances 命令。使用 run-instances， 會為您 AWS CLI 執行使用者資料的 base64 編碼。

以下範例顯示如何在命令列上將指令碼指定為字串：

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
```

```
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data echo user data
```

以下範例顯示如何使用文字檔案指定指令碼。請確認使用 `file://` 前綴來指定檔案。

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data file://my_script.txt
```

以下是具有 shell 指令碼的範例文字檔案。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

範例：修改已停止執行個體的使用者資料

您可以使用 modify-instance-attribute 命令修改已停止執行個體的使用者資料。使用 時modify-instance-attribute， AWS CLI 不會為您執行使用者資料的 base64 編碼。

- 在 Linux 電腦上，請使用 base64 命令來編碼使用者資料。

  ```
  base64 my_script.txt >my_script_base64.txt
  ```

- 在 Windows 電腦上，請使用 certutil 命令編碼使用者資料。您必須先移除第一行 (BEGIN CERTIFICATE) 和最後一行 (END CERTIFICATE) AWS CLI，才能將此檔案與 搭配使用。

  ```
  certutil -encode my_script.txt my_script_base64.txt
  notepad my_script_base64.txt
  ```

請使用 `--attribute` 和 `--value` 參數，使用已編碼的文字檔案來指定使用者資料。請確認使用 `file://` 前綴來指定檔案。

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData --value file://my_script_base64.txt
```

範例：清除已停止執行個體的使用者資料

若要刪除現有的使用者資料，請使用 modify-instance-attribute 命令，如下所示：

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

範例：檢視使用者資料

若要擷取執行個體的使用者資料，請使用 describe-instance-attribute 命令。使用 時describe-instance-attribute， AWS CLI 不會為您執行使用者資料的 base64 解碼。

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData
```

下列為已使用 base64 編碼之使用者資料的範例輸出。

```
{
    "UserData": {
        "Value":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQpzZXJ2aWNlIGh0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvbg=="
    },
    "InstanceId": "i-1234567890abcdef0"
}
```

- 在 Linux 上，請使用 `--query` 選項取得已編碼的使用者資料，並使用 base64 命令將其解碼。

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" | base64 --decode
  ```

- 在 Windows 電腦上，使用 `--query` 選項以取得已編碼的使用者資料，並使用 certutil 命令將其解碼。請注意，編碼輸出會存放在檔案中，解碼輸出則會存放在另一個檔案中。

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" >my_output.txt
  certutil -decode my_output.txt my_output_decoded.txt
  type my_output_decoded.txt
  ```

以下為範例輸出。

```
#!/bin/bash
yum update -y
```

```
service httpd start
chkconfig httpd on
```

結合 shell 指令碼與 cloud-init 指令

根據預設，您一次只能在使用者資料中包含一種內容類型。不過，您可以使用 MIME 多部分檔案中的 `text/cloud-config` 和 `text/x-shellscript` 內容類型，以在使用者資料中包含 shell 指令碼和 cloud-init 指令。

以下展示 MIME 多部分格式。

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

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud-init directives

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
shell script commands
--//--
```

例如，下列使用者資料包含 cloud-init 指令和 bash shell 指令碼。cloud-init 指令會建立檔案 (`/test-cloudinit/cloud-init.txt`)，並將 `Created by cloud-init` 寫入到該檔案。bash shell 指令碼會建立檔案 (`/test-userscript/userscript.txt`) 並將 `Created by bash shell script` 寫入到該檔案。

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

--//
```

```
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
runcmd:
- [ mkdir, /test-cloudinit ]
write_files:
- path: /test-cloudinit/cloud-init.txt
content: Created by cloud-init

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
mkdir test-userscript
touch /test-userscript/userscript.txt
echo "Created by bash shell script" >> /test-userscript/userscript.txt
--//--
```

Amazon EC2 如何處理 Windows 執行個體的使用者資料

在 Windows 執行個體，啟動代理程式會執行與使用者資料相關的任務。如需詳細資訊，請參閱下列內容：

- EC2Launch v2
- EC2Launch
- EC2Config 服務

如需 CloudFormation 範本中`UserData`屬性組合的範例，請參閱 Base64 編碼 UserData 屬性和 Base64 編碼 UserData 屬性與 AccessKey 和 SecretKey。

有關與 lifecycle hook 搭配運作之 Auto Scaling 群組內的執行個體上執行命令的範例，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的教學課程：設定使用者資料來透過執行個體中繼資料擷取目標生命週期狀態。

目錄

- 使用者資料指令碼
- 壓縮版使用者資料
- 使用者資料執行
- 使用者資料與 Tools for Windows PowerShell

使用者資料指令碼

若要讓 EC2Config 或 EC2Launch 執行指令碼，您必須在新增指令碼至使用者資料時將其括在特殊標籤內。該使用何種標籤，取決於您是在命令提示視窗 (批次命令) 中執行命令，或是使用 Windows PowerShell 來執行命令。

如果您同時指定了批次指令碼和 Windows PowerShell 指令碼，則不論在執行個體使用者資料中的出現順序為何，都會先執行批次指令碼，接著才執行 Windows PowerShell 指令碼。

如果您在使用者資料指令碼中使用 AWS API AWS CLI，包括 ，則必須在啟動執行個體時使用執行個體描述檔。執行個體描述檔提供使用者資料指令碼進行 API 呼叫所需的適當 AWS 登入資料。如需詳細資訊，請參閱執行個體描述檔。您指派給 IAM 角色的許可取決於您使用 API 呼叫的服務。如需詳細資訊，請參閱Amazon EC2 的 IAM 的角色。

指令碼類型

- 批次指令碼的語法
- Windows PowerShell 指令碼的語法
- YAML 組態指令碼的語法
- Base64 編碼

批次指令碼的語法

使用 script 標籤指定批次指令碼。使用換行符號分隔命令，如下列範例所示。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

在預設情況下，使用者資料指令碼僅會在您啟動執行個體時執行一次。若要在每次重新開機或啟動執行個體時執行使用者資料指令碼，請將 <persist>true</persist> 新增至使用者資料。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

EC2Launch v2 代理程式

若要在 UserData 階段將 XML 使用者資料指令碼做為分離的程序與 EC2Launch v2 executeScript 任務一起執行，請將 <detach>true</detach> 新增至使用者資料。

> **Note**
>
> 舊版啟動代理程式不支援 detach 標籤。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<detach>true</detach>
```

Windows PowerShell 指令碼的語法

AWS Windows AMIs 包含 AWS Tools for Windows PowerShell，因此您可以在使用者資料中指定這些 cmdlet。如果您將 IAM 角色與執行個體建立關聯，則不需要指定 cmdlet 的登入資料，因為在執行個體上執行的應用程式會使用角色的登入資料來存取 AWS 資源（例如 Amazon S3 儲存貯體）。

使用 <powershell> 標籤指定 Windows PowerShell 指令碼。使用分行符號來區隔命令。<powershell> 標籤區分大小寫。

例如：

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
```

在預設情況下，使用者資料指令碼僅會在您啟動執行個體時執行一次。若要在每次重新開機或啟動執行個體時執行使用者資料指令碼，請將 <persist>true</persist> 新增至使用者資料。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

您可以使用 <powershellArguments> 標籤指定一或多個 PowerShell 引數。如果未傳遞引數，EC2Launch 和 EC2Launch v2 預設會新增下列引數：-ExecutionPolicy Unrestricted。

範例：

```
<powershell>
    $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

EC2Launch v2 代理程式

若要在 UserData 階段將 XML 使用者資料指令碼做為分離的程序與 EC2Launch v2 executeScript 任務一起執行，請將 <detach>true</detach> 新增至使用者資料。

> **Note**
>
> 舊版啟動代理程式不支援 detach 標籤。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

YAML 組態指令碼的語法

如果您使用 EC2Launch v2 來執行指令碼，則可以使用 YAML 格式。若要檢視 EC2Launch v2 的組態任務、詳細資料和範例，請參閱 EC2Launch v2 任務組態。

指定具有 executeScript 任務的 YAML 指令碼。

執行 PowerShell 指令碼的 YAML 語法範例

```
version: 1.0
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
      $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
      New-Item $file -ItemType file
```

執行批次指令碼的 YAML 語法範例

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: batch
    runAs: localSystem
    content: |-
      echo Current date and time >> %SystemRoot%\Temp\test.log
      echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
```

Base64 編碼

如果您使用 Amazon EC2 API 或不會為您執行使用者資料的 base64 編碼的工具，就必須自行編碼使用者資料。若否，系統就會記錄一則錯誤，指出找不到要執行的 `script` 或 `powershell` 標籤。以下是使用 Windows PowerShell 的編碼範例。

```
$UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

以下是使用 PowerShell 的解碼範例。

```
$Script =
 [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

如需 base64 編碼的詳細資訊，請參閱 https：//https://www.ietf.org/rfc/rfc4648.txt。

壓縮版使用者資料

EC2Launch v2 支援壓縮版使用者資料，做為提交超過 IMDS 所設 16 KB 限制的使用者資料的方法。如需使用此功能，請將您的使用者資料指令碼壓縮成 `.zip` 封存，然後傳遞至 EC2 執行個體。若 EC2Launch v2 偵測到壓縮版使用者資料，可自動對壓縮版使用者資料指令碼解壓縮，然後執行該指令碼。

對於標準使用者資料，如果您使用 Amazon EC2 API 或不會為您執行使用者資料的 base64 編碼的工具，就必須自行編碼壓縮版使用者資料。若要了解使用者資料大小限制及 base64 編碼的相關資訊，請參閱 存取 EC2 執行個體的執行個體中繼資料。

使用者資料執行

根據預設，所有 AWS Windows AMIs都會為初始啟動啟用使用者資料執行。您可以指定在下一次重新開機或重新啟動執行個體時執行使用者資料指令碼。或者，您可以指定每次重新開機或重新啟動執行個體時都執行使用者資料指令碼。

> **Note**
>
> 在預設情況下，初始啟動後不會執行使用者資料。若要在重新開機或啟動執行個體後執行使用者資料，請參閱 在後續重新啟動或啟動期間執行指令碼。

產生隨機密碼時，會透過本機管理員帳戶來執行使用者資料指令碼。否則，會透過系統帳戶執行使用者資料指令碼。

執行個體啟動指令碼

執行個體使用者資料中的指令碼僅會在執行個體初次啟動時執行。如果找到 `persist` 標籤，即會為後續的重新開機或啟動啟用使用者資料執行。EC2Launch v2、EC2Launch 和 EC2Config 的日誌檔案包含來自標準輸出和標準錯誤串流的輸出。

EC2Launch v2

EC2Launch v2 的記錄檔是 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`。

> **Note**
>
> `C:\ProgramData` 資料夾可能隱藏不見。若要檢視此資料夾，您必須顯示隱藏的檔案和資料夾。

以下資訊會在執行使用者資料時記錄：

- `Info: Converting user-data to yaml format` - 如果使用者資料是以 XML 格式提供
- `Info: Initialize user-data state` - 開始執行使用者資料時
- `Info: Frequency is: always` - 如果使用者資料工作在每次開機時執行
- `Info: Frequency is: once` - 如果使用者資料任務只執行一次
- `Stage: postReadyUserData execution completed` - 使用者資料執行結束時

EC2Launch

EC2Launch 的日誌文件是 `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log`。

`C:\ProgramData` 資料夾可能隱藏不見。若要檢視此資料夾，您必須顯示隱藏的檔案和資料夾。

以下資訊會在執行使用者資料時記錄：

- `Userdata execution begins` - 開始執行使用者資料時
- `<persist> tag was provided: true` - 如果找到 persist 標籤
- `Running userdata on every boot` - 如果找到 persist 標籤
- `<powershell> tag was provided.. running powershell content` - 如果找到 powershell 標籤
- `<script> tag was provided.. running script content` - 如果找到指令碼標籤
- `Message: The output from user scripts` - 如果執行使用者資料指令碼，則會記錄其輸出

EC2Config

EC2Config 的日誌文件是 `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2Config.log`。以下資訊會在執行使用者資料時記錄：

- `Ec2HandleUserData: Message: Start running user scripts` - 開始執行使用者資料時
- `Ec2HandleUserData: Message: Re-enabled userdata execution` - 如果找到 persist 標籤
- `Ec2HandleUserData: Message: Could not find <persist> and </persist>` - 如果沒有找到 persist 標籤

- `Ec2HandleUserData: Message: The output from user scripts` - 如果執行使用者資料指令碼，則會記錄其輸出

在後續重新啟動或啟動期間執行指令碼

若更新執行個體使用者資料，下次您重新啟動或啟動執行個體時，執行個體中繼資料會自動反映更新的使用者資料內容。然而，視乎安裝的啟動代理程式，可能需要額外組態，方可設定使用者資料指令碼在後續重新啟動或啟動時執行。

如果您選擇透過 Sysprep 關閉選項，使用者資料指令碼會在下次執行個體啟動或重新開機時執行，即使您並沒有為後續的重新開機或啟動啟用使用者資料執行亦同。

若要了解啟用使用者資料執行的說明，選取與您的啟動代理程式相符的標籤。

EC2Launch v2

與 EC2Launch v1 不同，EC2Launch v2 在每次啟動時都會評估使用者資料任務。您不必手動排程使用者資料任務。依據包括的頻率或保存選項來執行使用者資料。

若是 XML 使用者資料指令碼

如需在每次啟動時執行使用者資料指令碼，可新增 `<persist>true</persist>` 標記至使用者資料。若不包括保存標記，則僅在初始啟動時執行使用者資料指令碼。

若是 YAML 使用者資料

- 如需初始啟動時在使用者資料中執行任務，可設定任務 `frequency` 為 `once`。
- 如需每次啟動時在使用者資料中執行工作，可設定任務 `frequency` 為 `always`。

EC2Launch

1. 連接至 Windows 執行個體。
2. 開啟 PowerShell 命令視窗並執行下列其中一項命令：

   執行一次

   如需下次啟動時執行使用者資料一次，可使用 `-Schedule` 標記。

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   Schedule
   ```

在所有後續啟動時執行

如需在所有後續啟動時執行使用者資料，可使用 -SchedulePerBoot 標記。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

3. 中斷連線您的 Windows 執行個體。若要在下一次啟動執行個體時執行更新的指令碼，請停止執行個體並更新使用者資料。

EC2Config

1. 連接至 Windows 執行個體。
2. Open C:\Program Files\Amazon\Ec2ConfigService \Ec2ConfigServiceSetting.exe.
3. 針對 User Data (使用者資料)，選取 Enable UserData execution for next service start (啟用下一次服務啟動時的使用者資料執行)。
4. 中斷連線您的 Windows 執行個體。若要在下一次啟動執行個體時執行更新的指令碼，請停止執行個體並更新使用者資料。

使用者資料與 Tools for Windows PowerShell

您可以使用 Tools for Windows PowerShell 指定、修改和檢視您執行個體的使用者資料。如需使用執行個體中繼資料檢視您執行個體中使用者資料的資訊，請參閱存取 EC2 執行個體的執行個體中繼資料。如需使用者資料和 的相關資訊 AWS CLI，請參閱 使用者資料和 AWS CLI。

範例：在啟動時指定執行個體使用者資料

使用執行個體使用者資料建立文字檔案。若要在每次重新開機或啟動執行個體時執行使用者資料指令碼，請新增 <persist>true</persist>，如下範例所示。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

若要在啟動您的執行個體時指定執行個體使用者資料，請使用 New-EC2Instance 命令。此命令不會為您執行使用者資料的 base64 編碼。使用以下命令，將使用者資料編碼至名為 script.txt 的文字檔案：

```
PS C:\> $Script = Get-Content -Raw script.txt
PS C:\> $UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

使用 -UserData 參數，將使用者資料傳遞到 New-EC2Instance 命令。

```
PS C:\> New-EC2Instance -ImageId ami-abcd1234 -MinCount 1 -MaxCount 1 -
InstanceType m3.medium \
    -KeyName my-key-pair -SubnetId subnet-12345678 -SecurityGroupIds sg-1a2b3c4d \
    -UserData $UserData
```

範例：更新已停止之執行個體的執行個體使用者資料

您可以使用 Edit-EC2InstanceAttribute 命令修改已停止之執行個體的使用者資料。

使用新的指令碼建立文字檔案。使用以下命令，將使用者資料編碼至名為 new-script.txt 的文字檔案：

```
PS C:\> $NewScript = Get-Content -Raw new-script.txt
PS C:\> $NewUserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($NewScript))
```

使用 -UserData 和 -Value 參數指定使用者資料。

```
PS C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute userData -
Value $NewUserData
```

範例：檢視執行個體使用者資料

若要擷取執行個體的使用者資料，請使用 Get-EC2InstanceAttribute 命令。

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute
 userData).UserData
```

下列為範例輸出。請注意使用者資料已編碼。

```
PHBvd2Vyc2hlbGw
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlci1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```

使用以下命令，將已編碼的使用者資料存放至變數中，然後加以解碼。

```
PS C:\> $UserData_encoded = (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -
Attribute userData).UserData
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData_encoded)
```

下列為範例輸出。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

範例：重新命名執行個體以符合標籤值

使用 Get-EC2Tag 命令讀取標籤值，在第一次開機時重新命名執行個體以符合標籤值，然後重新開機。若要成功執行此命令，您必須有連接至執行個體的角色具備 ec2:DescribeTags 許可，因為由 API 呼叫擷取標籤資訊。如需使用 IAM 角色設定許可的詳細資訊，請參閱 將 IAM 角色連接至執行個體。

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
    $instanceId = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" =
 $token} -Method GET -Uri 'http://169.254.169.254/latest/meta-data/instance-id' -
UseBasicParsing
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
```

```
        {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
    Catch
        {$ErrorMessage = $_.Exception.Message
        Write-Output "Rename failed: $ErrorMessage"}}
 Else
    {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $instanceId = (Invoke-WebRequest http://169.254.169.254/latest/meta-data/instance-
id -UseBasicParsing).content
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
    {Try
        {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
    Catch
        {$ErrorMessage = $_.Exception.Message
        Write-Output "Rename failed: $ErrorMessage"}}
 Else
    {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

如果您的執行個體設定為從執行個體中繼資料存取標籤，則還可以使用執行個體中繼資料中的標籤重新命名執行個體，。如需詳細資訊，請參閱使用執行個體中繼資料檢視 EC2 執行個體的標籤。

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
 $nameValue = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
 -Method GET -Uri 'http://169.254.169.254/latest/meta-data/tags/instance/Name' -
UseBasicParsing
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
```

```
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $nameValue = Get-EC2InstanceMetadata -Path /tags/instance/Name
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

識別在單一請求中啟動的每個執行個體

此範例會示範如何使用使用者資料和執行個體中繼資料，設定您的 Amazon EC2 執行個體。

> **Note**
>
> 本節中的範例使用 IMDS 的 IPv4 地址：169.254.169.254。如果您要透過 IPv6 地址擷取 EC2 執行個體的執行個體中繼資料，請確定您啟用並改用 IPv6 地址：[fd00:ec2::254]。IMDS 的 IPv6 地址與 IMDSv2 命令相容。IPv6 地址僅在 IPv6 支援的子網路 (僅支援雙重堆疊或 IPv6) 中的 Nitro 型執行個體可存取。

Alice 想要啟動四個她最愛的資料庫 AMI 的執行個體，其中第一個為原始執行個體，其餘三個則為複本。當她啟動這些執行個體時，她想要為每個複本新增關於複寫策略的使用者資料。她知道資料將會在四個執行個體上可用，因此她需要使用允許每個執行個體識別其適用部分的方式架構使用者資料。她可以使用 ami-launch-index 執行個體中繼資料值執行此作業，每一個執行個體的該值都是唯一的。如果您同時啟動一個以上的執行個體，ami-launch-index 表示執行個體的啟動順序。第一個啟動的執行個體的值為 0。

下列是 Alice 建構的使用者資料。

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

replicate-every=1min 資料定義第一個複本的組態，replicate-every=5min 定義第二個複本的組態，以此類推。Alice 決定以 ASCII 字串提供此資料，其中以直立線符號 (|) 分隔不同執行個體的資料。

Alice 使用 run-instances 命令啟動四個執行個體，指定使用者資料。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 4 \
    --instance-type t2.micro \
    --user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

在啟動它們之後，所有執行個體都會擁有使用者資料的複本，並且常見的中繼資料會顯示於此：

- AMI ID：ami-0abcdef1234567890
- 保留 ID：r-1234567890abcabc0
- 公有金鑰：無
- 安全群組名稱：預設
- 執行個體類型：t2.micro

但是，每個執行個體都有唯一的中繼資料，如下表所示。

中繼資料	Value
instance-id	i-1234567890abcdef0
ami-launch-index	0

中繼資料	Value
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

中繼資料	Value
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

中繼資料	Value
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

中繼資料	Value
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice 可以使用 `ami-launch-index` 值判斷使用者資料中的哪一個部分適用於特定執行個體。

1. 她連線到其中一個執行個體，擷取該執行個體的 `ami-launch-index`，確保它是其中一個複本：

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/
   token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
   data/ami-launch-index
   2
   ```

   對於後續步驟，IMDSv2 請求會從之前的 IMDSv2 命令來使用儲存的字符，但前提是字符並未過期。

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
   2
   ```

2. 她將 `ami-launch-index` 儲存為變數。

   IMDSv2

   ```
   [ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
    http://169.254.169.254/latest/meta-data/ami-launch-index`
   ```

IMDSv1

```
[ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-
launch-index`
```

3. 她將使用者資料儲存為變數。

   IMDSv2

   ```
   [ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
    http://169.254.169.254/latest/user-data`
   ```

   IMDSv1

   ```
   [ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data`
   ```

4. 最後，Alice 使用 cut 命令擷取適用於該執行個體的使用者資料部分。

   IMDSv2

   ```
   [ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
   replicate-every=5min
   ```

   IMDSv1

   ```
   [ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
   replicate-every=5min
   ```

偵測主機是否為 EC2 執行個體

您可能需要知道應用程式或網站是否在 EC2 執行個體上執行，特別是如果您是在混合運算環境。您可以使用下列其中一個選項，判斷應用程式或網站的主機是否為 EC2 執行個體。

選項

- 檢查 執行個體身分文件
- 檢查系統 UUID
- 檢查系統虛擬機器產生識別符

檢查 執行個體身分文件

每個執行個體都有已簽署的執行個體身分文件，您可以用密碼編譯方式進行驗證。您可以使用執行個體中繼資料服務 (IMDS) 來尋找這些文件。

如需詳細資訊，請參閱執行個體身分文件。

檢查系統 UUID

您可以取得系統 UUID，並在 UUID 的開頭八位元組查看 EC2 (在 Linux 中，這可能是小寫 ec2)。這是一項快速方法，但有可能會不準確，因為不是 EC2 執行個體的系統，也有可能會具有以這些字元開頭的 UUID。此外，某些版本的 SMBIOS 使用小端序格式，其在 UUID 開頭並不包含 EC2。對於使用 SMBIOS 2.4 for Windows 的 EC2 執行個體，或具有自己 SMBIOS 實作的 Amazon Linux 以外的 Linux 發行版本，可能會是這種情況。

Linux 範例：從 DMI 取得 UUID (僅限 HVM AMI)

運用下列命令來透過桌面管理介面 (DMI) 取得 UUID：

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

在下列的範例輸出中，UUID 是以「EC2」開頭，這代表系統可能是 EC2 執行個體。

```
EC2E1916-9099-7CAF-FD21-012345ABCDEF
```

在下列的範例輸出中，是以小端序格式來表示 UUID。

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

或者，對於建立在 Nitro 系統上的執行個體，您可以使用下列命令：

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

如果輸出為執行個體 ID，如下列範例輸出，則系統為 EC2 執行個體：

```
i-0af01c0123456789a
```

Linux 範例：從 hypervisor 取得 UUID (僅限 PV AMI)

使用下列命令從 hypervisor 取得 UUID：

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

在下列的範例輸出中，UUID 是以「ec2」開頭，這代表系統可能是 EC2 執行個體。

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Windows 範例：使用 WMI 或 Windows PowerShell 來取得 UUID

使用 Windows Management Instrumentation (Windows 管理規範) 命令列 (WMIC)，如下所示：

```
wmic path win32_computersystemproduct get uuid
```

或者，如果使用 Windows PowerShell，請使用 Get-WmiObject cmdlet，如下所示：

```
PS C:\> Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" | Select
 UUID
```

在下列的範例輸出中，UUID 是以「EC2」開頭，這代表系統可能是 EC2 執行個體。

```
EC2AE145-D1DC-13B2-94ED-012345ABCDEF
```

如果是使用 SMBIOS 2.4 的執行個體，可能會以小端序格式來表示 UUID，例如：

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

檢查系統虛擬機器產生識別符

虛擬機器產生識別符由 128 位元的專屬緩衝區組成，該緩衝區解譯為加密隨機整數識別符。您可以擷取虛擬機器產生識別符以識別您的 Amazon Elastic Compute Cloud 執行個體。產生識別符透過 ACPI 表項目在執行個體的訪客作業系統中公開。若您的電腦被複製或匯入到 AWS(例如使用 VM Import/Export)，那麼值就會改變。

範例：從 Linux 擷取虛擬機器產生識別符

您可使用以下命令從執行 Linux 的執行個體中擷取虛擬機器產生識別符。

Amazon Linux 2

1. 根據需要，使用以下命令更新現有軟體套件：

```
sudo yum update
```

2. 如有必要，使用下列命令以 busybox 套件作為來源：

```
sudo curl https://www.rpmfind.net/linux/epel/next/8/Everything/x86_64/Packages/
b/busybox-1.35.0-2.el8.next.x86_64.rpm --output busybox.rpm
```

3. 如有必要，使用下列命令安裝必要條件套件：

```
sudo yum install busybox.rpm iasl -y
```

4. 執行下列 `iasl` 命令從 ACPI 表產生輸出結果：

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

5. 執行下列命令，以檢閱 `iasl` 命令的輸出結果：

```
cat SSDT2.dsl
```

輸出結果應產生擷取虛擬機器產生識別符所需的地址空間：

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
```

```
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
    }
}
}
```

6. (選用) 使用下列命令提升其餘步驟的終端許可：

```
sudo -s
```

7. 使用下列命令存放先前收集的地址空間：

```
VMGN_ADDR=0xFED01000
```

8. 使用下列命令重複執行地址空間並建置虛擬機器產生識別符：

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

9. 使用下列命令從輸出檔案中擷取虛擬機器產生識別符：

```
cat vmgenid ; echo
```

您的輸出應類似以下內容：

```
EC2F335D979132C4165896753E72BD1C
```

Ubuntu

1. 根據需要，使用以下命令更新現有軟體套件：

```
sudo apt update
```

2. 如有必要，使用下列命令安裝必要條件套件：

```
sudo apt install busybox iasl -y
```

3. 執行下列 `iasl` 命令從 ACPI 表產生輸出結果：

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

4. 執行下列命令，以檢閱 `iasl` 命令的輸出結果：

```
cat SSDT2.dsl
```

輸出結果應產生擷取虛擬機器產生識別符所需的地址空間：

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
```

```
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
```

```
        })
    }
}
}
```

5. (選用) 使用下列命令提升其餘步驟的終端許可：

```
sudo -s
```

6. 使用下列命令存放先前收集的地址空間：

```
VMGN_ADDR=0xFED01000
```

7. 使用下列命令重複執行地址空間並建置虛擬機器產生識別符：

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

8. 使用下列命令從輸出檔案中擷取虛擬機器產生識別符：

```
cat vmgenid ; echo
```

您的輸出應類似以下內容：

```
EC2F335D979132C4165896753E72BD1C
```

範例：從 Windows 擷取虛擬機器產生識別符

您可建立範例應用程式，從執行 Windows 的執行個體中擷取虛擬機器產生識別符。如需詳細資訊，請參閱 Microsoft 文件集中的「取得虛擬機器產生識別符」。

Amazon EC2 執行個體的執行個體身分文件

您啟動的每個執行個體都有一個提供執行個體本身相關資訊的 執行個體身分文件。您可以使用 執行個體身分文件 來驗證執行個體的屬性。

執行個體停止、開始、重新開始或啟動時，會產生執行個體身分文件。您可以透過執行個體中繼資料服務 (IMDS) 存取執行個體的執行個體身分文件。如需指示，請參閱擷取執行個體身分文件。

執行個體身分文件使用純文字 JSON 格式。其包含下列資訊。

資料	說明
accountId	啟動執行個體 AWS 的帳戶 ID。
architecture	用於啟動執行個體 (i386 | x86_64 | arm64) 的 AMI 架構。
availabilityZone	執行個體執行的可用區域的名稱。例如 us-east-1 。請記住，可用區域名稱可能因 AWS 帳戶而異。
billingProducts	執行個體的計費產品。
devpayProductCodes	已廢除。
imageId	用來啟動執行個體的 AMI ID。
instanceId	執行個體的 ID。
instanceType	執行個體的執行個體類型。
kernelId	與執行個體相關聯的核心 ID (如適用)。
marketplaceProductCodes	用來啟動執行個體的 AMI AWS Marketplace 產品代碼。
pendingTime	執行個體啟動的日期和時間。
privateIp	執行個體的私有 IP 地址。對於僅 IPv4 和雙堆疊執行個體，這包含 IPv4 位址。對於僅 IPv6 的執行個體，這包含 IPv6 位址。
ramdiskId	與此執行個體相關聯的 RAM 磁碟 ID (如適用)。
region	執行個體執行時所在區域。
version	執行個體身分文件 格式的版本。

擷取 EC2 執行個體的執行個體身分文件

Amazon EC2 執行個體的執行個體身分文件是使用純文字 JSON 格式。如需執行個體身分文件內容的說明，請參閱 the section called “執行個體身分文件”。

執行個體身分文件會存放在執行個體的執行個體中繼資料中 (位於 instance-identity/document 動態資料類別中)。您可以連線至執行個體並從執行個體中繼資料擷取執行個體，藉此存取執行個體的執行個體身分文件。

您可以使用 IPv4 地址 169.254.169.254 或 IPv6 地址 fd00:ec2::254 來存取執行個體中繼資料。這些是 Link-local 地址，這表示您只能從執行個體加以存取。本頁面的範例使用 IMDS 的 IPv4 地址：169.254.169.254。若要透過 IPv6 擷取 EC2 執行個體的執行個體中繼資料，請使用 fd00:ec2::254。

要在擷取執行個體身分文件之後，驗證執行個體身分文件的真實性，請參閱 驗證執行個體身分文件。

IMDSv2

Linux

若要擷取執行個體身分文件，請從您的 Linux 執行個體執行下列命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

若要擷取執行個體身分文件，請從您的 Windows 執行個體執行下列 cmdlet。

```
[string]$token = (Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token-ttl-seconds' = '21600'} `
    -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -UseBasicParsing).Content
```

```
(Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $token} `
    -Uri 'http://169.254.169.254/latest/dynamic/instance-identity/document' -UseBasicParsing).Content
```

IMDSv1

Linux

若要擷取執行個體身分文件，請從您的 Linux 執行個體執行下列命令。

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

若要擷取執行個體身分文件，請從您的 Windows 執行個體執行下列 cmdlet。

```
(Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/
document).Content
```

以下為範例輸出。

```
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

驗證 Amazon EC2 執行個體的執行個體身分文件

如果您打算將 執行個體身分文件 的內容用於重要目的，您應該在使用它之前驗證其內容與真實性。

純文字 執行個體身分文件 會伴隨著三個雜湊和加密的簽章。您可以使用這些簽章來驗證 執行個體身分文件 的來源和真實性及其包含的資訊。提供的簽章如下：

- Base64 編碼的簽章 – 此為使用 RSA 金鑰對加密之執行個體身分文件的 base64 編碼 SHA256 雜湊。
- PKCS7 簽章 – 此為使用 DSA 金鑰對加密的 執行個體身分文件 SHA1 雜湊。
- RSA-2048 簽章 – 此為使用 RSA-2048 金鑰對加密的執行個體身分文件 SHA256 雜湊。

每個簽章都可以在執行個體中繼資料的不同端點上使用。您可以根據您的雜湊和加密需求使用任何一個簽章。若要驗證簽章，您必須使用對應的 AWS 公有憑證。

選項

- 選項 1：使用 PKCS7 簽章來驗證執行個體身分文件
- 選項 2：使用 Base64 編碼簽章來驗證執行個體身分文件
- 選項 3：使用 RSA-2048 簽章來驗證執行個體身分文件

選項 1：使用 PKCS7 簽章來驗證執行個體身分文件

本主題說明如何使用 PKCS7 簽章和 AWS DSA 公有憑證來驗證執行個體身分文件。

Linux 執行個體

使用 PKCS7 簽章和 AWS DSA 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 PKCS7 簽章，並將其與所需的標頭和頁尾新增至名為 pkcs7 的新檔案。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

   IMDSv1

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

3. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 DSA 公有憑證，並將內容新增至名為 certificate 的新檔案。

4. 使用 OpenSSL smime 命令以驗證簽章。包含指出需要驗證簽章的 `-verify` 選項，以及指出憑證不需要驗證的 `-noverify` 選項。

```
$ openssl smime -verify -in pkcs7 -inform PEM -certfile certificate -noverify | tee
 document
```

如果簽章有效，則會出現 `Verification successful` 訊息。

此命令也會將執行個體身分文件內容寫入名為 `document` 的新檔案。您可以使用下列命令，將來自執行個體中繼資料的執行個體身分文件內容與此檔案內容比較。

```
$ openssl dgst -sha256 < document
```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

如果無法驗證簽章，請聯絡 支援。

Windows 執行個體

先決條件

此程序需要 `System.Security` Microsoft .NET Core 類別。若要將該類別新增至 PowerShell 工作階段，請執行下列命令。

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 此命令會將類別只新增至目前的 PowerShell 工作階段。如果您啟動新的工作階段，您必須再次執行命令。

使用 PKCS7 簽章和 AWS DSA 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 PKCS7 簽章，將其轉換為位元組陣列，然後再新增至名為 `$Signature` 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

IMDSv2

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-ec2-
metadata-token-ttl-seconds' = '21600'} -Uri http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
dynamic/instance-identity/pkcs7).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Uri
 http://169.254.169.254/latest/dynamic/instance-identity/pkcs7).Content)
```

3. 從執行個體中繼資料擷取純文字執行個體身分文件，將其轉換為位元組陣列，然後新增至名為 $Document 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
   dynamic/instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 DSA 公有憑證，並將內容新增至名為 certificate.pem 的新檔案。

5. 從憑證檔案擷取憑證，並將其存放在名為 $Store 的變數中。

   ```
   PS C:\> $Store =
    [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
   Path certificate.pem)))
   ```

6. 驗證簽章。

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

如果簽章有效，則命令不會傳回任何輸出。如果無法驗證簽章，則命令會傳回 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`。如果無法驗證您的簽章，請聯絡 AWS 支援。

7. 驗證執行個體身分文件的內容。

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

如果執行個體身分文件的內容有效，則命令會傳回 `True`。如果無法驗證執行個體身分文件，請聯絡 AWS 支援。

選項 2：使用 Base64 編碼簽章來驗證執行個體身分文件

本主題說明如何使用 base64 編碼的簽章和 AWS RSA 公有憑證來驗證執行個體身分文件。

Linux 執行個體

使用 base64 編碼簽章和 AWS RSA 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 base64 編碼的簽章，將其轉換為二進位，然後新增至名為 `signature` 的檔案。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/signature | base64 -d >> signature
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/signature |
 base64 -d >> signature
```

3. 從執行個體中繼資料擷取純文字 執行個體身分文件，並新增至名為 document 的檔案。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   $ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
   metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
   dynamic/instance-identity/document >> document
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document
    >> document
   ```

4. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 RSA 公有憑證，並將內容新增至名為 certificate 的新檔案。

5. 從 AWS RSA 公有憑證擷取公有金鑰，並將其儲存至名為 的檔案key。

   ```
   $ openssl x509 -pubkey -noout -in certificate >> key
   ```

6. 使用 OpenSSL dgst 命令來驗證 執行個體身分文件。

   ```
   $ openssl dgst -sha256 -verify key -signature signature document
   ```

   如果簽章有效，則會出現 Verification successful 訊息。

   此命令也會將執行個體身分文件內容寫入名為 document 的新檔案。您可以使用下列命令，將來自執行個體中繼資料的執行個體身分文件內容與此檔案內容比較。

   ```
   $ openssl dgst -sha256 < document
   ```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

如果無法驗證簽章，請聯絡 支援。

Windows 執行個體

使用 base64 編碼簽章和 AWS RSA 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 base64 編碼的簽章，將其轉換為位元組陣列，然後新增至名為 $Signature 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/signature).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
   ```

3. 從執行個體中繼資料擷取純文字執行個體身分文件，將其轉換為位元組陣列，然後新增至名為 $Document 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 RSA 公有憑證，並將內容新增至名為 `certificate.pem` 的新檔案。
5. 驗證執行個體身分文件。

```
PS C:\> [Security.Cryptography.X509Certificates.X509Certificate2]::new((Resolve-
Path certificate.pem)).PublicKey.Key.VerifyData($Document, 'SHA256', $Signature)
```

如果簽章有效，則命令會傳回 `True`。如果無法驗證簽章，請聯絡 支援。

選項 3：使用 RSA-2048 簽章來驗證執行個體身分文件

本主題說明如何使用 RSA-2048 簽章和 AWS RSA-2048 公有憑證來驗證執行個體身分文件。

Linux 執行個體

使用 RSA-2048 簽章和 AWS RSA-2048 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 RSA-2048 簽章，並將其與所需的標頭和頁尾新增至名為 `rsa2048` 的檔案。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

IMDSv2

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
 && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/rsa2048 >> rsa2048 \
 && echo "" >> rsa2048 \
 && echo "-----END PKCS7-----" >> rsa2048
```

IMDSv1

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
```

```
&& curl -s http://169.254.169.254/latest/dynamic/instance-identity/rsa2048
>> rsa2048 \
&& echo "" >> rsa2048 \
&& echo "-----END PKCS7-----" >> rsa2048
```

3. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 RSA-2048 公有憑證，並將內容新增至名為 `certificate` 的新檔案。
4. 使用 OpenSSL smime 命令以驗證簽章。包含指出需要驗證簽章的 `-verify` 選項，以及指出憑證不需要驗證的 `-noverify` 選項。

```
$ openssl smime -verify -in rsa2048 -inform PEM -certfile certificate -noverify |
 tee document
```

如果簽章有效，則會出現 `Verification successful` 訊息。如果無法驗證簽章，請聯絡 支援。

Windows 執行個體

先決條件

此程序需要 `System.Security` Microsoft .NET Core 類別。若要將該類別新增至 PowerShell 工作階段，請執行下列命令。

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 此命令會將類別只新增至目前的 PowerShell 工作階段。如果您啟動新的工作階段，您必須再次執行命令。

使用 RSA-2048 簽章和 AWS RSA-2048 公有憑證驗證執行個體身分文件

1. 連線到執行個體。
2. 從執行個體中繼資料擷取 RSA-2048 簽章，將其轉換為位元組陣列，然後再新增至名為 `$Signature` 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

IMDSv2

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/rsa2048).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/rsa2048).Content)
```

3. 從執行個體中繼資料擷取純文字執行個體身分文件，將其轉換為位元組陣列，然後新增至名為 $Document 的變數。根據執行個體所使用的 IMDS 版本，使用下列其中一個命令。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. 在 AWS 執行個體身分文件簽章的公有憑證 尋找您所在區域的 RSA-2048 公有憑證，並將內容新增至名為 certificate.pem 的新檔案。

5. 從憑證檔案擷取憑證，並將其存放在名為 $Store 的變數中。

   ```
   PS C:\> $Store =
    [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
   Path certificate.pem)))
   ```

6. 驗證簽章。

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

如果簽章有效，則命令不會傳回任何輸出。如果無法驗證簽章，則命令會傳回 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`。如果無法驗證您的簽章，請聯絡 AWS 支援。

7. 驗證執行個體身分文件的內容。

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

如果執行個體身分文件的內容有效，則命令會傳回 `True`。如果無法驗證執行個體身分文件，請聯絡 AWS 支援。

AWS 執行個體身分文件簽章的公有憑證

下列 AWS 公有憑證可用於驗證執行個體身分文件的內容，如中所述驗證執行個體身分文件。

請依據您的所在區域以及您正在使用的驗證程序，確定您使用正確的憑證。如果您正在驗證 PKCS7 簽章，請使用 DSA 憑證。如果您正在驗證 base64 編碼簽章，請使用 RSA 憑證。如果您正在驗證 RSA-2048 簽章，請使用 RSA-2048 憑證。

展開下方的每個區域，以檢視特定區域的憑證。

美國東部 (維吉尼亞北部) — us-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
```

```
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUE1y2NIKCU+Rg4uu4u32koG9QEYIwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MzQwMVoXDTI5MDQyODE3MzQwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUE1y2NIKC
U+Rg4uu4u32koG9QEYIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAlxSmwcWnhT4uAeSinJuz+1BTcKhVSWb5jT8pYjQb8ZoZkXXRGb09mvYeU
NeqOBr27rvRAnaQ/9LUQf72+SahDFuS4CMI8nwowytqbmwquqFr4dxA/SDADyRiF
ea1UoMuNHTY49J/1vPomqsVn7mugTp+TbjqCfOJTpu0temHcFA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALFpzEAVWaQZMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU5MTJaGA8yMTk1MDExNzA4NTkxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
```

```
CgKCAQEAjS2vqZu9mEOhOq+0bRpAbCUiapbZMFNQqRg7kTlr7Cf+gDqXKpHPjsng
SfNz+JHQd8WPI+pmNs+q0Z2aTe23klmf2U52KH9/j1k8RlIbap/yFibFTSedmegX
E5r447GbJRsHUmuIIfZTZ/oRlpuIIO5/Vz7SOj22tdkdY2ADp7caZkNxhSP915fk
2jJMTBUOzyXUS2rBU/ulNHbTTeePjcEkvzVYPahD30TeQ+/A+uWUu89bHSQOJR8h
Um4cFApzZgN3aD5j2LrSMu2pctkQwf9CaWyVznqrsGYjYOY66LuFzSCXwqSnFBfv
fFBAFsjCgY24G2DoMyYkF3MyZlu+rwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUrynSPp4uqSECwy+PiO4qyJ8TWSkwgY4GA1UdIwSBhjCBg4AUrynS
Pp4uqSECwy+PiO4qyJ8TWSmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALFpzEAVWaQZMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADW/s8lXijwdP6NkEoH1m9XLrvK4YTqkNfR6
er/uRRgTx2QjFcMNrx+g87gAml11z+D0crAZ5LbEhDMs+JtZYR3ty0HkDk6SJM85
haoJNAFF7EQ/zCp1EJRIkLLsC7bcDL/Eriv1swt78/BB4RnC9W9kSp/sxd5svJMg
N9a6FAplpNRsWAnbP8JBlAP93oJzblX2LQXgykTghMkQO7NaY5hg/H5o4dMPclTK
lYGqlFUCH6A2vdrxmpKDLmTn5//5pujdD2MN0df6sZWtxwZ0osljV4rDjm9Q3VpA
NWIsDEcp3GUB4proOR+C7PNkY+VGODitBOw09qBGosCBstwyEqY=
-----END CERTIFICATE-----
```

美國東部 (俄亥俄) — us-east-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUVJTc+hOU+8Gk3JlqsX438Dk5c58wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MTE0OVoXDTI5MDQyODE3MTE0OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUVJTc+hOU
+8Gk3JlqsX438Dk5c58wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAywJQaVNWJqW0R0T0xVOSoN1GLk9x9kKEuN67RN9CLin4dA97qa7Mr5W4P
FZ6vnh5CjOhQBRXV9xJUeYSdqVItNAUFK/fEzDdjf1nUfPlQ3OJ49u6CV01NoJ9m
usvY9kWcV46dqn2bk2MyfTTgvmeqP8fiMRPxxnVRkSzlldP5Fg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM07oeX4xevdMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA2MTAx
MjU4MThaGA8yMTk1MTExNDEyNTgxOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA6v6kGMnRmFDLxBEqXzP4npnL65OO0kmQ7w8YXQygSdmNIoScGSU5wfh9
mZdcvCxCdxgALFsFqPvH8fqiE9ttI0fEfuZvHOs8wUsIdKr0Zz0MjSx3cik4tKET
ch0EKfMnzKOgDBavraCDeX1rUDU0Rg7HFqNAOry3uqDmnqtk00XC9GenS3z/7ebJ
fIBEPAam5oYMVFpX6M6St77WdNE8wEU8SuerQughiMVx9kMB07imeVHBiELbMQ0N
lwSWRL/61fA02keGSTfSp/0m3u+lesf2VwVFhqIJs+JbsEscPxOkIRlzy8mGd/JV
ONb/DQpTedzUKLgXbw7KtO3HTG9iXQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU2CTGYE5fTjx7gQXzdZSGPEWAJY4wgY4GA1UdIwSBhjCBg4AU2CTG
YE5fTjx7gQXzdZSGPEWAJY6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM07oeX4xevdMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANdqkIpVypr2PveqUsAKke1wKCOSuw1UmH9k
xX1/VRoHbrI/UznrXtPQOPMmHA2LKSTedwsJuorUn3cFH6qNs8ixBDrl8pZwfKOY
```

```
IBJcTFBbI1xBEFkZoO3wczzo5+8vPQ60RVqAaYb+iCa1HFJpccC3Ovajfa4GRdNb
n6FYnluIcDbmpcQePoVQwX7W3oOYLB1QLN7fE6H1j4TBIsFdO3OuKzmaifQlwLYt
DVxVCNDabpOr6Uozd5ASm4ihPPoEoKo7Ilp0fOT6fZ41U2xWA4+HF/89UoygZSo7
K+cQ90xGxJ+gmlYbLFR5rbJOLfjrgDAb2ogbFy8LzHo2ZtSe60M=
-----END CERTIFICATE-----
```

美國西部 (加利佛尼亞北部) — us-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUK2zmY9PUSTR7rc1k2OwPYu4+g7wwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MDI0M1oXDTI5MDQyODE3MDI0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
```

```
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUK2zmY9PU
STR7rc1k2OwPYu4+g7wwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQA1Ng4QmN4n7iPh5CnadSOc0ZfM7by0dBePwZJyGvOHdaw6P6E/vEk76KsC
Q8p+akuzVzVPkU4kBK/TRqLp19wEWoVwhhTaxHjQ1tTRHqXIVlrkw4JrtFbeNM21
GlkSLonuzmNZdivn9WuQYeGe7nUD4w3q9GgiF3CPorJe+UxtbA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJANNPkIpcyEtIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAzMDdaGA8yMTk1MDQwMzA5MDMwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApHQGvHvq3SVCzDrC7575BW7GWLzcj8CLqYcL3YY7JffupzZ7OjcftO57Z
4fo5Pj0CaS8DtPzh8+8vdwUSMbiJ6cDd3ooio3MnCq6DwzmsY+pY7CiI3UVG7KcH
4TriDqr1Iii7nB5MiPJ8wTeAqX89T3SYaf6Vo+4GCb3LCDGvnkZ9TrGcz2CHkJsj
AIGwgopFpwhIjVYm7obmuIxSIUv+oNH0wXgDL029Zd98SnIYQd/njiqkzE+lvXgk
4h4Tu17xZIKBgFcTtWPky+POGu81DYFqiWVEyR2JKKm2/iR1dL1YsT39kbNg47xY
aR129sS4nB5Vw3TRQA2jL0ToTIxzhQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUgepyiONs8j+q67dmcWu+mKKDa+gwgY4GA1UdIwSBhjCBg4AUgepy
iONs8j+q67dmcWu+mKKDa+ihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJANNPkIpcyEtIMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGLFWyutf1u0xcAc+kmnMPqtc/Q6b79VIX0E
tNoKMI2KR8lcV8ZElXDb0NC6v8UeLpe1WBKjaWQtEjL1ifKg9hdY9RJj4RXIDSK7
33qCQ8juF4vep2U5TTBd6hfWxt1Izi88xudjixmbpUU4YKr8UPbmixldYR+BEx0u
B1KJi9l1lxvuc/Igy/xeHOAZEjAXzVvHp8Bne33VVwMiMxWECZCiJxE4I7+Y6fqJ
pLLSFFJKbNaFyXlDiJ3kXyePEZSc1xiWeyRB2ZbTi5eu7vMG4i3AYWuFVLthaBgu
lPfHafJpj/JDcqt2vKUKfur5edQ6j1CGdxqqjawhOTEqcN8m7us=
-----END CERTIFICATE-----
```

美國西部 (奧勒岡) — us-west-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
```

```
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFx8PxCkbHwpD31bOyCtyz3GclbgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE3MjM1OVoXDTI5MDQyODE3MjM1OVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFx8PxCkb
HwpD31bOyCtyz3GclbgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBzOl+9Xy1+UsbUBI95HO9mbbdnuX+aMJXgG9uFZNjgNEbMcvx+h8P9IMko
z7PzFdheQQ1NLjsHH9mSR1SyC4m9ja6BsejH5nLBWyCdjfdP3muZM4O5+r7vUa1O
dWU+hP/T7DUrPAIVMOE7mpYa+WPWJrN6BlRwQkKQ7twm9kDalA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALZL3lrQCSTMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
```

```
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAxMzJaGA8yMTk1MDExNzA5MDEzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA02Y59qtAA0a6uzo7nEQcnJ26OKF+LRPwZfixBH+EbEN/Fx0gYy1jpjCP
s5+VRNg6/WbfqAsV6X2VSjUKN59ZMnMY9ALA/Ipz0n00Huxj38EBZmX/NdNqKm7C
qWu1q5kmIvYjKGiadfboU8wLwLcHo8ywvfgI6FiGGsEO9VMC56E/hL6Cohko11LW
dizyvRcvg/IidazVkJQCN/4zC9PUOVyKdhW33jXy8BTg/QH927QuNk+ZzD7HH//y
tIYxDhR6TIZsSnRjz3bOcEHxt1nsidc65mY0ejQty4hy7ioSiapw316mdbtE+RTN
fcH9FPIFKQNBpiqfAW5Ebp3Lal3/+wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU7coQx8Qnd75qA9XotSWT3IhvJmowgY4GA1UdIwSBhjCBg4AU7coQ
x8Qnd75qA9XotSWT3IhvJmqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALZL3lrQCSTMMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFZ1e2MnzRaXCaLwEC1pW/f0oRG8nHrlPZ9W
OYZEWbh+QanRgaikBNDtVTwARQcZm3z+HWSkaIx3cyb6vM0DSkZuiwzm1LJ9rDPc
aBm03SEt5v8mcc7sXWvgFjCnUpzosmky6JheCD4O1Cf8k0olZ93FQnTrbg62OK0h
83mGCDeVKU3hLH97FYoUq+3N/IliWFDhvibAYYKFJydZLhIdlCiiB99AM6Sg53rm
oukS3csyUxZyTU2hQfdjyo1nqW9yhvFAKjnnggiwxNKTTPZzstKW8+cnYwiiTwJN
QpVoZdt0SfbuNnmwRUMi+QbuccXweav29QeQ3ADqjgB0CZdSRKk=
-----END CERTIFICATE-----
```

非洲 (開普敦) — af-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7DCCAqwCCQCncbCtQbjuyzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA2MDQxMjQ4MDVaFw00
NTA2MDQxMjQ4MDVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQC12Nr1gMrHcFSZ7S/A
pQBSCMHWmn2qeoQTMVWqe50fnTd0zGFxDdIjKxUK58/8zjWG5uR4TXRzmZpGpmXB
bSufAR6BGqud2LnT/HIWGJAsnX2uOtSyNfCoJigqwhea5w+CqZ6I7iBDdnB4TtTw
qO6TlnExHFVj8LMkylZgiaE1CQIVAIhdobse4K0QnbAhCL6R2euQzloXAoGAV/21
WUuMz/79Ga0JvQcz1FNy1sT0pU9rU4TenqLQIt5iccn/7EIfNtvVO5TZKulIKq7J
gXZr0x/KIT8zsNweetLOaGehPIYRMPX0vunMMR7hN7qA7W17WZv/76adywIsnDKq
ekfe15jinaX8MsKUdyDK7Y+ifCG4PVhoM4+W2XwDgYQAAoGAIxOKbVgwLxbn6Pi2
6hBOihFv16jKxAQI0hHzXJLV0Vyv9QwnqjJJRfOCy3dB0zicLXiIxeIdYfvqJr+u
hlN8rGxEZYYJjEUKMGvsc0DW85jonXz0bNfcP0aaKH0lKKVjL+OZi5n2kn9wgdo5
```

```
F3CVnMl8BUra8A1Tr2yrrE6TVZ4wCQYHKoZIzjgEAwMvADAsAhQfa7MCJZ+/TEY5
AUr0J4wm8VzjoAIUSYZVu2NdRJ/ERPmDfhW5EsjHlCA=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjcw
NzE0MDVaGA8yMTk5MDUwMjA3MTQwNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFd571nUzVtke3rPyRkYfvs3jh0C0EMzzG72boyUNjnfw1+m0TeFraTLKb9T6F
7TuB/ZEN+vmlYqr2+5Va8U8qLbPF0bRH+FdaKjhgWZdYXxGzQzU3ioy5W5ZM1VyB
7iUsxEAlxsybC3ziPYaHI42UiTkQNahmoroNeqVyHNnBpQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAAJLylWyElEgOpW4B1XPyRVD4pAds8Guw2+krgqkY0HxLCdjosuH
RytGDGN+q75aAoXzW5a7SGpxLxk6Hfv0xp3RjDHsoeP0i1d8MD3hAC5ezxS4oukK
s5gbPOnokhKTMPXbTdRn5ZifCbWlx+bYN/mTYKvxho7b5SVg2o1La9aK
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAIFI+O5A6/ZIMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA2MDQx
MjQ4MDRaGA8yMTk4MTEwNzEyNDgwNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy7/WHBBHOrk+20aumT07g8rxrSM0UXgki3eYgKauPCG4Xx//vwQbuZwI
oeVmR9nqnhfij2wOcQdbLandh0EGtbxerete3IoXzd1KXJb11PVmzrzyu5SPBPuP
iCeV4qdjjkXo2YWM6t9YQ911hcG96YSp89TBXFYUh3KLxfqAdTVhuC0NRGhXpyii
j/czo9njofHhqhTr7UEyPun8NVS2QWctLQ86N5zWR3Q0GRoVqqMrJs0cowHTrVw2
9Qr7QBjjBOVbyYmtYxm/DtiKprYV/e6bCAVok015X1sZDd3oCOQNoGlv5XbHJe2o
JFD8GRRy2rkWO/lNwVFDcwec6zC3QwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCE
goqzjpCpmMgCpszFHwvRaSMbspKtK7wNImUjrSBOfBJsfFulyg1Zgn2nDCK7kQhx
jMJmNIvXbps3yMqQ2cHUkKcKf5t+WldfeT4Vk1Rz6HSA8sd0kgVcIesIaoy2aaXU
VEB/oQziRGyKdN1d4TGYVZXG44CkrzSDvlbmfiTq5tL+kAieznVF3bzHgPZW6hKP
EXC3G/IXrXicFEe6YyE1Rakl62VncYSXiGe/i2XvsiNH3Qlmnx5XS7W0SCN0oAxW
EH9twibauv82DVg1WOkQu8EwFw8hFde9X0Rkiu0qVcuU8lJgFEvPWMDFU5sGB6ZM
gkEKTzMvlZpPbBhg99Jl
-----END CERTIFICATE-----
```

亞太區域 (香港) — ap-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCO7MJe5Y3VLjAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDMwMjIxMjFaFw00
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfX
blCvCoVb9957OkLGn/04CowHXJ+vTBR7eyIa6AoXltsQXBOmrJswToFKKxT4gbuw
jK7s9QQX4CmTRWcEgO2RXtZSVjOhsUQMh+yf7Ht4OVL97LWnNfGsX2cwjcRWHYgI
7lvnuBNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOCG
eSNmxpW4QFu4pIlAykm6EnTZKKHT87gdXkAkfoC5fAfOxxhnE2HezZHp9Ap2tMV5
8bWNvoPHvoKCQqwfm+OUBlAxC/3vqoVkKL2mG1KgUH9+hrtpMTkwO3RREnKe7I5O
x9qDimJpOihrL4I0dYvy9xUOoz+DzFAW8+ylWVYpA4GFAAKBgQDbnBAKSxWr9QHY
6Dt+EFdGz6lAZLedeBKpaP53Z1DTO34J0C55YbJTwBTFGqPtOLxnUVDlGiD6GbmC
80f3jvogPR1mSmGsydbNbZnbUEVWrRhe+y5zJ3g9qs/DWmDW0deEFvkhWVnLJkFJ
9pdOu/ibRPH1lE2nz6pK7GbOQtLyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eaygLTtFpFJqzWHc=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICSzCCAbQCCQDtQvkVxRvK9TANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJV
UzETMBEGA1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2VhdHRsZTEYMBYGA1UE
ChMPQW1hem9uLmNvbSBJbmMuMRowGAYDVQQDExFlYzIuYW1hem9uYXdzLmNvbTAe
Fw0xOTAyMDMwMzAwMDZaFw0yOTAyMDIwMzAwMDZaMGoxCzAJBgNVBAYTAlVTMRMw
EQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdTZWF0dGxlMRgwFgYDVQQKEw9B
bWF6b24uY29tIEluYy4xGjAYBgNVBAMTEWVjMi5hbWF6b25hd3MuY29tMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1kkHXYTfc7gY5Q55JJhjTieHAgacaQkiR
Pity9QPDE3b+NXDh4UdP1xdIw73JcIIG3sG9RhWiXVCHh6KkuCTqJfPUknIKk8vs
M3RXflUpBe8Pf+P92pxqPMCz1Fr2NehS3JhhpkCZVGxxwLC5gaG0Lr4rFORubjYY
Rh84dK98VwIDAQABMA0GCSqGSIb3DQEBCwUAA4GBAA6xV9f0HMqXjPHuGILDyaNN
dKcvplNFwDTydVg32MNubAGnecoEBtUPtxBsLoVYXCOb+b5/ZMDubPF9tU/vSXuo
TpYM5Bq57gJzDRaBOntQbX9bgHiUxw6XZWaTS/6xjRJDT5p3S1E0mPI3lP/eJv4o
Ezk5zb3eIf10/sqt4756
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMoxixvs3YssMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA3MjAw
ODQ0NDRaGA8yMTk3MTIyMzA4NDQ0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA4T1PNsOg0FDrGlWePoHeOSmOJTA3HCRy5LSbYD33GFU2eBrOIxoU/+SM
rInKu3GghAMfH7WxPW3etIAZiyTDDU5RLcUq2Qwdr/ZpXAWpYocNc/CEmBFtfbxF
z4uwBIN3/drM0RSbe/wP9EcgmNUGQMMZWeAji8sMtwpOblNWAP9BniUG0Flcz6Dp
uPovwDTLdAYT3TyhzlohKL3f6O48TR5yTaV+3Ran2SGRhyJjfh3FRpP4VC+z5LnT
WPQHN74Kdq35UgrUxNhJraMGCzznolUuoR/tFMwR93401GsM9fVA7SW3jjCGF81z
PSzjy+ArKyQqIpLW1YGWDFk3sf08FQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDK
2/+C3nPMgtyOFX/I3Cyk+Pui44IgOwCsIdNGwuJysdqp5VIfnjegEu2zIMWJSKGO
lMZoQXjffkVZZ97J7RNDW06oB7kj3WVE8a7U4WEOfnO/CbMUf/x99CckNDwpjgW+
K8V8SzAsQDvYZs2KaE+18GFfLVF1TGUYK2rPSZMHyX+v/TIlc/qUceBycrIQ/kke
jDFsihUMLqgmOV2hXKUpIsmiWMGrFQV4AeV0iXP8L/ZhcepLf1t5SbsGdUA3AUY1
3If8s81uTheiQjwY5t9nM0SY/1Th/tL3+RaEI79VNEVfG1FQ8mgqCK0ar4m0oZJl
tmmEJM7xeURdpBBx36Di
-----END CERTIFICATE-----
```

亞太區域 (海德拉巴) — ap-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXjrQ4+XMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAJCKGBBoxIUxqBk94JHhwZZbgvbP0DAOoHENQWxp/98lI7/
YOfYJOVMJS22aCnHDurofmo5rvNIkgXi7Rztbhu
+lko9rK6DgpmpUwBU0WZtf34aZ2IWNBwHaVhHvWAQf9/46u18dMa2YucK1Wi+Vc+M
+KldrvgXmhym6ErNlzhJyMAkGByqGSM44BAMDLwAwLAIUaaPKxaOHoYvwz709xXpsQueIq+UCFFa/
GpzoDOSokl1057NU/2hnsiW4
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAPw1mjt5QguhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yNjA1Mjgx
MDQzMjlaFw0zMTA1MjcxMDQzMjlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
08B7tDQ/rBXKG6473AGJtDzUQ/H+EdYwM76C7dvqhnjkwMjtR7bY+nJrGDdwD9YT
eYmLqaRYmLJdnjT2vwg9FYNAQHwxZVea846QNI+0yWm1YeJI+yb7Q/sWDIusBEAO
MekjzjWODnmdvvKPx0vJ7nKVg1XT0TunIllJTIY5g5sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFM4QY6Q5jpIb3pUP7EBEDUVlvGgsMIGOBgNVHSME
gYYwgYOAFM4QY6Q5jpIb3pUP7EBEDUVlvGgsoWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQD8NZo7eUILoTASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAGPVdLHdjH2PVcfoK41bRQCp
vzwwOB0XpQ1AqQBYZXmm6+Ggc3nQLoOKLG1oRlc38psWtxBjGRL+ktvyAjgwk+BL
xM7Yx+wDf8m2nfrvIdaMHk0i+U7bWaiFod3cD6XICfwg/tFz50auWTtlIPcNOXAU
qPuHbYlFgMUlNuNp78nB
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAIVWfPw/X82fMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MDQx
NDMwMjhaGA8yMjAxMTIwODE0MzAyOFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAg29QEFriG+qFEjYW/v62nN70lMJY/Hevx5TtmU/VIYBPQa3HUGTBAbbI
2Tmy8UMpa8kZeaYeI3RAfiQWt0Ws7wUrBuO2Pdp518WDPaJUH7RWEuu1BDDkyZRW
NAMNPCn3ph7Od243IFcLGku7HVekel5poqRpSfojrMasjlf+CvixUeAJbmFoxUHK
kh5unzG2sZyO4wHXcJPQkRf5a8zSTPe9YZP1kXPPEv4p/jTSggaYPxXyS6QVaTlV
zLeLFZ0fesLPMeil3KYQtV7IKLQiEA2F6dxWnxNWQlyMHtdq6PucfEmVx17i/Xza
yNBRo0azY8WUNVkEXrRhp/pU8Nh3GQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU9AO1aZk9RLXk2ZvRVoUxYvQy9uwwgY4GA1UdIwSBhjCBg4AU9AO1
aZk9RLXk2ZvRVoUxYvQy9uyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAIVWfPw/X82fMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBADexluMRQRftqViahCnauEWGdMvLCBr8A+Yr
6hJq0guoxEk/lahxR137DnfMPuSbi1Rx5QKo7oBrWfG/zsgQUnF2IwHTzwD+i/2m
```

```
XCane6FiS5RpK3lGdILq8ZmlhQk+6iI8yoZLr0LCfTh+CLgIKHOknfR5lFzgzAiF
SI8/Q9mm+uvYtSTZECI6Zh57QZPoETAG/yl+9jiOy2lAelqa/k1i+Qo8gMfOc+Pm
dwY7o6fV+oucgRlsdey6VM45LeyILQqv0RXtVzjuowanzmCCFMjgqiO9oZAWu4Oh
+F3unijELo01vZJs8s2N3KGlo3/jtUFTX6RTKShZlAPLwBi5GMI=
-----END CERTIFICATE-----
```

亞太區域 (雅加達) — ap-southeast-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCArCgAwIBAgIGAXbVDEikMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24g
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYUAAoGBAPjuiEx05N3JQ6cVwntJie67D8OuNo4jGRn
+crEtL7YO0jSVB9zGE1ga
+UgRPIaYETL293S8rTJTVgXAqdpBwfaHC6NUzre8U8iJ8FMNnlP9Gw1oUIlgQBjORyynVJexoB31TDZM
+/52g9O/bpq1QqNyKbeIgyBBlc1dAtr1QLnsMAkGByqGSM44BAMDLwAwLAIUK8E6RDIRtwK+9qnaTOBhvO/
njuQCFFocyT1OxK+UDR888oNsdgtif2Sf
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUMDbUg+reaqO28n34vFUuk6gFWbgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI2MDQyMjEyMDY0MFoXDTMxMDQyMTEyMDY0MFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCnCS/Vbt0gQ1ebWcur2hSO7PnJifE4OPxQ7RgSAlc4/spJp1sD
P+ZrS0LO1ZJfKhXf1R9S3AUwLnsc7b+IuVXdY5LK9RKqu64nyXP5dx170zoL8loE
yCSuRR2fs+04i2QsWBVP+KFNAn7P5L1EHRjkgTO8kjNKviwRV+OkP9ab5wIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUsnsEPOUYPQO0Gv9t4arIC2NN
kHQwgZkGA1UdIwSBkTCBjoAUsnsEPOUYPQO0Gv9t4arIC2NNkHShYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUMDbUg+re
aqO28n34vFUuk6gFWbgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
```

```
AAOBgQCFibNMMdZF13osIpWxjb7uXrqpDYFat+196JVSdq+WMYLfcQtudZOcTclo
Zh+OXR49nRv9x/Q4XRfMWGCdptp4q/6NCScEUPBbY2+7CJyH0gyC6vYcGmLfJbUi
kMB27EWJ4tm18a/qZockIeGgJgqt5/KSZb5iN5CuxRHRDGZI+Q==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMtdyRcH51j9MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MDgx
MjM5MTZaGA8yMjAxMDkxMjEyMzkxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvUsKCxoH6KXRYJLeYTWAQfaBQeCwhJaR56mfUeFHJE4g8aFjWkiN4uc1
TvOyYNnIZKTHWmzmulmdinWNbwP0GiROHb/i7ro0Hhvnptyycgt8ag8affiIbx5X
7ohdwSN2KJ6G0IKflIx7f2NEI0oAMM/9k+T1eVF+MVWzpZoiDp8frLNkqp8+RAgz
ScZsbRfwv3u/if5xJAvdg2nCkIWDMSHEVPoz0lJo7v0ZuDtWWsL1LHnL5ozvsKEk
+ZJyEi23r+U1hIT1NTBdp4yoigNQexedtwCSr7q36oOdDwvZpqYlkLi3uxZ4ta+a
01pzOSTwMLgQZSbKWQrpMvsIAPrxoQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU1GgnGdNpbnL3lLF30Jomg7Ji9hYwgY4GA1UdIwSBhjCBg4AU1Ggn
GdNpbnL3lLF30Jomg7Ji9hahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMtdyRcH51j9MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACVl00qQlatBKVeiWMrhpczsJroxDxlZTOba
6wTMZk7c3akb6XMOSZFbGaifkebPZqTHEhDlrClM2j9AIlYcCx6YCrTf4cuhn2mD
gcJN33143eOWSaeRY3ee4j+V9ne98y3kO2wLz95VrRgclPFR8po2iWGzGhwUi+FG
q8dXeCH3N0DZgQsSgQWwmdNQXZZej6RHLU/8In5trHKLY0ppnLBjn/UZQbeTyW5q
RJB3GaveXjfgFUWj2qOcDuRGaikdS+dYaLsi5z9cA3FolHzWxx9MOs8io8vKqQzV
XUrLTNWwuhZy88cOlqGPxnoRbw7TmifwPw/cunNrsjUUOgs6ZTk=
-----END CERTIFICATE-----
```

亞太地區 (馬來西亞) — ap-southeast-5

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQC5X6U+vgoLEDAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMxMjU3NTRaGA8y
MDUwMDEwMzEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
```

```
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAIzEMPCIPFfOYCg4
BCjKGyl60wOfmwHPzSOXZ3Z2wS/LYNHUtHGwtVNePSTyCu/CuZF6gC9n/wBORtQp
+SSkn+weGc/BmUA1mp/vrN7v+aSCgKJoO+sgpa1PPOgNvUaMw6O5odsZWQCMSjkU
6RTo/PL2v/tMfiCocF4ghvyRC6hvAhUA0VoObKC2IXzXgVvRRupo4qHbcm8CgYAe
bbNuawh3rAxkFvUs9FPzW5E+xllG16Z//61PENKqonmk+zBiBdiIlS1F6ZqmTqkI
z5+qfSt1m3pb3j2W0NT7lEDFvy8Gr6Y2vohCHmL+T1u1Yy4PeqbgrFwcen7y7WoO
/KCV7Y9/ODQMMyuAzT3h5wJNweT7L5MUN8JYpZSi3QOBhAACgYBqaDuG2u6V91Qj
K2wEAElxaaRaNo/ewg/wWDmHYqoeHOR0HfuFCYgASE9f7ULqYtX1VURcgcjw9XN4
BDmPilXvfi04INPTnw4IxFJKDzzC0kVH7esVas982Po8v3megH32H9Rl87r7UG1c
ZEbkSkKVX6YKYg1PR3rfjXgdwVZv/zAJBgcqhkjOOAQDAzAAMC0CFFWeRe2fYW2i
6mMd26Wzbx87Y0DXAhUAoPCnF+5hGJw0jT9aL7QsgcfLi9Y=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAMuBl6rhZCJKMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFuKydxZsordNH7bLwIluEGOkX7/CdLdpeqkDKEhQkFwzpRxaX4EAlkGh2/o7D
8qneC9cGQhqSG5WVVBrmZG7sfkFOM4m1AtY++kfv+MYto1VFgLk1xJbkpq1r4YeQ
Ul+ZsJYsZpyX/t+g8s7rW0OVcBsYx4L75bf34z38mwK8PQIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBADD9C4pWL8RUvF1CJW8kExj35xmozlFlmrKs8Zpi8+Eg6q+W9dgd
xMdH95tgZtmVMDqlvVR+DK0iO1BNpqPjrqWkk2tTLivpS+sGzCE/jCl18Q28Rk71
/A3gLD7Rtbq5TKNvuFCHwYMjrTDHI6aBjIaAlDm4e2/j/OxVtHyZGTre
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANC3xtbPhQ2GMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDAxMDMx
MjU3NTRaGA8yMjAzMDYwOTEyNTc1NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAt3aMy7Hsp4ySG3mlfi+pdTcZw6H6XNU1Y36fNdi4c+MzinQQbnqMPyt7
QLgU+XCWmcWsVo7GQF6n9N01Rh+UXXUZU4jcXlFocQPCWf9O+IIIPXkd67kFMUV
HAxCELjfxHbC+I8e7dwOJhmdF4Bfi52Ty8zz0HdE8JDypPkTDlXuGvTgDyW7NP56
I/vlQaXLoYSbcQe5pv2a9gyBaaCM1QoeqwAHAeCNXb9Nuj9ZX3GHGJb3TuqAeKCD
```

```
5i9TscCB9XjY6Fx+zfSAobjBZwgLEtLOwJhbZnKmx4gJMaaNFipAjVT2FSS3+yev
eTYBoa1dvhk0ivQyQIPpHmihrmkWuwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBC
fdgyI8GjmCqiHAlh+L1bj0LdNql9zl7RXmOEzsuRdtdMumkxYXX88UtROy3fdi1i
VaEwHdAK8ThzRkesgHza/cXzqCMewaYxujSI6p6G7x99FFeGif1x0FJdj8AoeTL7
4h9bmS/6l4/NL7DJI9G7ovES/hoUA9v9TDhv+vauxXlgfrp0MPecprxBYlrc+DH2
adGCdKCp2lQ2YDKOD9TCEjYIli8XSoyevoWHUjfDYrCrCp8l4s/p7HOgYr8fJBAs
EuVy8211LVz1/X4EMBRNtNjXK9sklsxAOX14NDfBFSS0toxl3K6Tf9t/PviBl95d
hncyDAcFgDCK4w8LLlVW
-----END CERTIFICATE-----
```

亞太區域 (墨爾本) — ap-southeast-4

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjWF7P2MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAPRXSsQP9E3dw8QXKlrgBgEVCprLHdK/bbrMas0XMu1EhOD
+q
+0PcTr8+iwbtoXlY5MCeatWIp1GrXQjVqsF8vQqxlEuRuYKbR3nq4mWwaeGlx9AG5EjQHRa3GQ44wWHOdof0M3NRI1MP
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUUOsUa3Aj7pphfmDjsZAFNWODCUowDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI2MDYwMzEwNDM1M1oXDTMxMDYwMjEwNDM1M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDHezwQr2VQpQSTW5TXNefiQrP+qWTGAbGsPeMX4hBMjAJUKys2
NIRcRZaLM/BCew2FIPVjNtlaj6Gwn9ipU4Mlz3zIwAMWi1AvGMSreppt+wV6MRtf
Ojh0Dvj/veJe88aEZJMozNgkJFRS+WFWsckQeL56tf6kY6QTlNo8V/0CsQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUVCy+PqHsYEzvAHcGK+lrlknS
UvowgZkGA1UdIwSBkTCBjoAUVCy+PqHsYEzvAHcGK+lrlknSUvqhYKReMFwxCzAJ
```

```
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUUOsUa3Aj
7pphfmDjsZAFNWODCUowEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBRI72kM+Q6PCU584L6UZ8rk6uERCeq5g8dDzlLZMaMpV4n/s2MJbFP2hO+
ovahJnxLTrs7jP3VXZeVj5Eb0dfYneg0+zNtycnHe+9jH+GTKS8+QkwGj3zmwxqG
mZ9V6XoGwtO/aEz24Yp7CRf8etCYEnVI2ChD7xClRV3DJwliEA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAN4GTQ64zVs8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTMx
MzMzMDBaGA8yMjAxMTIxNzEzMzMwMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA2BYgeCr+Rk/jIAEDOHS7wJql62vc83QEwjuzk0qOFEReIZzlN1fBRNXK
g0T178Kd3gLYcE59wEFbTe/X5yOA1Lo95x1anSAo7R+Cisf9C2HQuJp+gVb+zx71
lniPF7gHziGpm0M8DdAU/IW+wkZwGbP4z7Hq9+bJ0P2ltvPJ5yxSgkFuDsI9VBHa
CLoprHsCHh2VdP8KcMgQQMmHe1NmBpyTk0ul/aLmQkCQEX6ZIRGOeq228fwlh/t+
Ho+jv87duihVKic6MrL32SlD+maX0LSDUydWdaOLLTGkh7oV7+bFuH6msrXUu+Ur
ZEP1r/MidCWMhfgrFzeTBzOHA97qxQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUcHMd1cHqzmsQ5hpUK3EMLhHdsi4wgY4GA1UdIwSBhjCBg4AUcHMd
1cHqzmsQ5hpUK3EMLhHdsi6hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAN4GTQ64zVs8MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAI4PFyVN+7EGSObioiPnv0LL0f70SSzUZJ8p
XO90d4rWea7jIbgZ2AKb+ErynkU9xVg7XQQ5k6KDWgp/4jYFL2dqnt/YAY4PS0un
RSrYElawxLT0BcLn4rcSDC79vQe1xGC5//wDdV6b399COAHRAK6axWYy5w32u9PL
uw0cIp3Ch8JoNwcgTHKRRGzePmBeR4PNqhHTArG4/dJk6/aUO4OpX0WzI6L67CGY
6Nex3dau+gkLCK93dTEkrXtyXHu4wB0J9zd1w+iQ0SEa9eKc78/NjEsF/FZdGrWC
t57lIMOOXJhQ1kRgSwNeZdQWV1dRakvO6sfcvVYkfj1wAvZvvAw=
-----END CERTIFICATE-----
```

亞太區域 (孟買) — ap-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
```

```
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUDLA+x6tTAP3LRTr0z6nOxfsozdMwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MTMwMVoXDTI5MDQyODE0MTMwMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUDLA+x6tT
AP3LRTr0z6nOxfsozdMwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAZ7rYKoAwwiiH1M5GJbrT/BEk3OO2VrEPw8ZxgpqQ/EKlzMlOs/0Cyrmp7
UYyUgYFQe5nq37Z94rOUSeMgv/WRxaMwrLlLqD78cuF9DSkXaZIX/kECtVaUnjk8
BZx0QhoIHOpQocJUSlm/dLeMuE0+0A3HNR6JVktGsUdv9ulmKw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPRYyD8TtmC0MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjAzMDcx
MDQ1MDFaGA8yMTk1MDgxMTEwNDUwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA0LSS5I/eCT2PM0+qusorBx67QL26BIWQHd/yF6ARtHBb/1DdFLRqE5Dj
07Xw7eENC+T79mOxOAbeWg91KaODOzw6i9I/2/HpK0+NDEdD6sPKDA1d45jRra+v
CqAjI+nV9Vw91wv7HjMk3RcjWGziM8/hw+3YNIutt7aQzZRwIWlBpcqx3/AFd8Eu
2UsRMSHgkGUW6UzUF+h/U8218XfrauKNGmNKDYUhtmyBrHT+k6J0hQ4pN7fe6h+Z
w9RVHm24BGhlLxLHLmsOIxvbrF277uX9Dxu1HfKfu5D2kimTY7xSZDNLR2dt+kNY
/+iWdIeEFpPT0PLSILt52wP6stF+3QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBI
E6w+WWC2gCfoJO6c9HMyGLMFEpqZmz1n5IcQt1h9iyO7Vkm1wkJiZsMhXpk73zXf
TPxuXEacTX3SOEa07OIMCFwkusO5f6leOyFTynHCzBgZ3U0UkRVZA3WcpbNB6Dwy
h7ysVlqyT9WZd7EOYm5j5oue2G2xdei+6etgn5UjyWm6liZGrcOF6WPTdmzqa6WG
ApEqanpkQd/HM+hUYex/ZS6zEhd4CCDLgYkIjlrFbFb3pJ1OVLztIfSN5J4Oolpu
JVCfIq5u1NkpzL7ys/Ub8eYipbzI6P+yxXiUSuF0v9b98ymczMYjrSQXIf1e8In3
OP2CclCHoZ8XDQcvvKAh
-----END CERTIFICATE-----
```

亞太區域 (紐西蘭) – ap-southeast-6

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq4CCQCcP+D4eLFOajAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMxMjEzMDZaGA8y
MDUwMTIwNDEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBANBMmM/LbAX13IeB
3yr2pMMpiTW1b1go/axNLfcxEmj0fThKk8y0FAT1frN/UPdI9qRInsyBdEZQLiIH
4WHCXk8IidQLh4DED3G9DvSFFC4Gb8ax+UIQY11wfHPWO4+wA9q8liYTg/FxdJMM
tZS9tZAZxjH6+8yfV8Bt0wUd2Y+lAhUAwg0CxCmZPinHtKbFnpMQfKnzjCMCgYAf
i+D7nGVFOWJ8EYOpRYbG0pHkrpwNy+h01csn6IAwQbIfeOG+SF4N3LeuOAC9vXGY
VGh9XNxuasvqkRAzgRmJ/EYM5Xa/YN62EhddRuN8wCOg66MeczUDh3RTb+Tu6HK0
UR6oTSOQ6rxA5aA1qTc7TqI0+pRZv+QYIZK/ozsZzgOBhAACgYAbVm+rOX8J38cG
LxKjUXVM0KB69SksGLEqjO4ps80EcZ/0W3g1hGLnSX/RjYfAS60hbKK1GX9PpcaO
4mDzCTm3dZ6shbae5t8bBU+mUi8BmaK+Pdu3z5J77y4zv22/yXte8ulZ5J0zKP9K
QP3p3pa7XI9NaXpKGT3yZsJ0BWY2xTAJBgcqhkjOOAQDAzAAMC0CFBs4KQmrxHjQ
OHh0r8JqB8UiwlutAhUAwgxid7wT6FNm8Zwq9hioQ2v2K0A=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAMwLuv2/tk9eMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMx
MjEzMDZaGA8yMjA0MDUwOTEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQC9AsBNOUfT8QpguHqMa9KIXP/YgZOmfRAgEmnLawXTC+40bXUq/fwkx+cebv3x
CvUx6d07/iDlD0dm2Hf2HhPFi3OaCI7vc8c3lqVBvcn6yzmwv35aSEvtVm2NHC3j
4/q6vN2tR2d4a1Mf1QhmOVOJ7XiTEbCy2Yv+pmDbk29V6QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAHatyQIy612kio8neI4r5VBg8pgKwW+mFGpAzOQ2NOyLCNoOeD1D
lPRaSipBN7WW6jpIozbz7JG7Kqc3N+piPGGzfjH9gZjqUqfOGrNHB5nsxvMHI2iL
e+DwsOcV7qnEVRgYbmLYOVKJv25wCsM/Tcn/6f6scEX2x+H0hAh7rwiP
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAI/xUfWwSQOtMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEyMDMx
MjEzMDZaGA8yMjA0MDUwOTEyMTMwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArbid9Tnunu0pWCQUqZCkGeiF8hWixi+LBS1AOew2UsT9keAt2rXXGlU/
wSMriKHsvcrfKtuVKj5chArsOyFvagszH5lIKhI+GfWVQEd2aZiOvAJ+iRUQq1+R
gOr2gyjWlf3znZXTcyhHcJoXDhbn5e+kEa3pwS4yKNRpM3P61V4M2702xis08hjP
qMDxwXMa1Fiy4b7nAyNQifgrwb3bCA5k5prmkl/iajaWa1zMMFgLI9vA6UwUSbTK
H9iOuL/xNfMX+Xzq4CCaQYiJ8zdi56kqnB/lrA05NSjOPLskg6sUGkyiDxGK4gzC
DFrO6oygcdlBaxnyAECGdJcOM6TOmwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBg
8yVPNpctrvSvTazf/Ij8qC6+q6df1SptdYKQkOYWeUVZR7+3uTZ9Yiyq+Bh8Hbyf
NcR0eR6UqhgiB3p5slYNhTvLkpuMuqTpVJFvQtzZycnmKXAQ35G5zCYV0+pkF59L
ch26SukWk0+VP3kIM9VmaIZQKJEBlBLcnbDZyXlHUDnZAYPcxxKQ8uFPkW0GOFMr
ekFCGOUmFSItFXOHy5hD2xmctKNk3l12m6cKydYvONtk/3DovKNnw3c5k40WHQW/
Zdh/LWd4/Jrkan+kQnqm4rkrXF+x3aF0qrKTtHMICvpkZ8g7/588hBs4mZnCmFLa
q/g9WqIcd9SMmBkT6KZ8
-----END CERTIFICATE-----
```

亞太區域 (大阪) — ap-northeast-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUHTRhxHhBZFOGvTFKxHoy9+f5Hl8wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NTQwN1oXDTI5MDQyODE2NTQwN1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQ4IUHTRhxHhB
ZFOGvTFKxHoy9+f5Hl8wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAUZx7DcYbhWNTD4BNGhr5beruT2OUoGHH9J73UKxwdqeb9bH1LIWhIZO0X
/1mjn3bWBgCwfoS8gjZwsVB6fZbNBRy8urdBZJ87xF/4JPBjt7S9oGx/zthDUYrC
yK0Y0v4G0PgiS81CvYLg09LpmYhLSJbXENlkC04v5yxdKxZxyg==
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAMn1yPk22ditMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA3MTkx
MTEyNThaGA8yMTk2MTIyMjExMTI1OFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArznEYef8IjhrJoazI0QGZkmlmHm/4rEbyQbMNifxjsDE8YWtHNwaM91z
zmyK6Sk/tKlWxcnl3g31iq305ziyFPEewe5Qbwf1iz2cMsvfNBcTh/E6u+mBPH3J
gvGanqUJt6c4IbipdEouIjjnynyVWd4D6erLl/ENijeR1OxVpaqSW5SBK7jms49E
pw3wtbchEl3qsE42Ip4IYmWxqjgaxB7vps91n4kfyzAjUmklcqTfMfPCkzmJCRgp
Vh1C79vRQhmriVKD6BXwfZ8tG3a7mijeDn7kTsQzgO07Z2SAE63PIO48JK8HcObH
tXORUQ/XF1jzi/SIaUJZT7kq3kWl8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBj
ThtO9dLvU2QmKuXAhxXjsIdlQgGG3ZGh/Vke4If1ymgLx95v2Vj9Moxk+gJuUSRL
BzFte3TT6b3jPolbECgmAorjj8NxjC17N8QAAI1d0S0gI8kqkG7V8iRyPIFekv+M
pcai1+cIv5IV5qAz8QOMGYfGdYkcoBjsgiyvMJu/2N2UbZJNGWvcEGkdjGJUYYOO
NaspCAFm+6HA/K7BD9zXB1IKsprLgqhiIUgEaW3UFEbThJT+z8UfHG9fQjzzfN/J
nT6vuY/0RRu1xAZPyh2gr5okN/s6rnmh2zmBHU1n8cbCc64MVfXe2g3EZ9Glq/9n
izPrI09hMypJDP04ugQc
-----END CERTIFICATE-----
```

亞太區域 (首爾) — ap-northeast-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
```

```
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUbBSn2UIO6vYk4iNWV0RPxJJtHlgwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEzMzg0NloXDTI5MDQyODEzMzg0NlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUbBSn2UIO
6vYk4iNWV0RPxJJtHlgwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAmjTjalG8MGLqWTC2uYqEM8nzI3px1eo0ArvFRsyqQ3fgmWcQpxExqUqRy
l3+2134Kv8dFab04Gut5wlfRtc2OwPKKicmv/IXGN+9bKFnQFjTqif08NIzrDZch
aFT/uvxrIiM+oN2YsHq66GUhO2+xVRXDXVxM/VObFgPERbJpyA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANuCgCcHtOJhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTQx
NTU3NDRaGA8yMTk1MDIxNzE1NTc0NFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA66iNv6pJPmGM20W8HbVYJSlKcAg2vUGx8xeAbzZIQdpGfkabVcUHGB6m
Gy59VXDMDlrJckDDk6dxUOhmcX9z785TtVZURq1fua9QosdbTzX4kAgHGdp4xQEs
mO6QZqg5qKjBP6xr3+PshfQ1rB8Bmwg0gXEm22CC7o77+7N7Mu2sWzWbiUR7vil4
9FjWS8XmMNwFTlShp4l1TDTevDWW/uYmC30RThM9S4QPvTZ0rAS18hHVam8BCTxa
LHaVCH/Yy52rsz0hM/FlghnSnK105ZKj+b+KIp3adBL8OMCjgc/Pxi0+j3HQLdYE
32+FaXWU84D2iP2gDT28evnstzuYTQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC1
```

```
mA4q+12pxy7By6g3nBk1s34PmWikNRJBwOqhF8ucGRv8aiNhRRye9lokcXomwo8r
KHbbqvtK85l0xUZp/Cx4sm4aTgcMvfJP29jGLclDzeqADIvkWEJ4+xncxSYVlS9x
+78TvF/+8h9U2LnSl64PXaKdxHy2IsHIVRN4GtoaP2Xhpa1S0M328Jykq/571nfN
1WRD1c/fQf1edgzRjhQ4whcAhv7WRRF+qTbfQJ/vDxy8lkiOsvU9XzUaZ0fZSfXX
wXxZamQbONvFcxVHY/0PSiM8nQoUmkkBQuKleDwRWvkoJKYKyr3jvXK7HIWtMrO4
jmXe0aMy3thyK6g5sJVg
-----END CERTIFICATE-----
```

亞太區域 (新加坡) — ap-southeast-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUSqP6ih+++5KF07NXngrWf26mhSUwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE0MzAxNFoXDTI5MDQyODE0MzAxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
```

```
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUSqP6ih++
+5KF07NXngrWf26mhSUwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQAw13BxW11U/JL58j//Fmk7qqtrZTqXmaz1qm2WlIpJpW750MOcP4ux1uPy
eM0RdVZ4jHSMv5gtLAv/PjExBfw9n6vNCk+5GZG4Xec5DoapBZHXmfMo93sjxBFP
4x9rWn0GuwAVO9ukjYPevq2Rerilrq5VvppHtbATVNY2qecXDA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAJVMGw5SHkcvMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
ODU3MTlaGA8yMTk1MDQwMzA4NTcxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAlaSSLfBl7OgmikjLReHuNhVuvM20dCsVzptUyRbut+KmIEEc24wd/xVy
2RMIrydGedkW4tUjkUyOyfET5OAyT43jTzDPHZTkRSVkYjBdcYbe9o/0Q4P7IVS3
XlvwrUu0qo9nSID0mxMnOoF1l8KAqnn10tQ0W+lNSTkasW7QVzcb+3okPEVhPAOq
MnlY3vkMQGI8zX4iOKbEcSVIzf6wuIffXMGHVC/JjwihJ2USQ8fq6oy686g54P4w
ROg415kLYcodjqThmGJPNUpAZ7MOc5Z4pymFuCHgNAZNvjhZDA842Ojecqm62zcm
Tzh/pNMNeGCRYq2EQX0aQtYOIj7bOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU6SSB+3qALorPMVNjToM1Bj3oJMswgY4GA1UdIwSBhjCBg4AU6SSB
+3qALorPMVNjToM1Bj3oJMuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAJVMGw5SHkcvMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAF/0dWqkIEZKg5rca8o0P0VS+tolJJE/FRZO
atHOeaQbWzyac6NEwjYeeV2kY63skJ+QPuYbSuIBLM8p/uTRIvYM4LZYImLGUvoO
IdtJ8mAzq8CZ3ipdMs1hRqF5GRp8lg4w2QpX+PfhnW47iIOBiqSAUkIr3Y3BDaDn
EjeXF6qS4iPIvBaQQ0cvdddNh/pE33/ceghbkZNTYkrwMyBkQlRTTVKXFN7pCRUV
+L9FuQ9y8mP0BYZa5e1sdkwebydU+eqVzsil98ntkhpjvRkaJ5+Drs8TjGaJWlRw
5WuOr8unKj7YxdL1bv7//RtVYVVi296ldoRUYv4SCvJF11z0OdQ=
-----END CERTIFICATE-----
```

亞太區域 (雪梨) — ap-southeast-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFxWyAdk4oiXIOC9PxcgjYYh71mwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MjE0M1oXDTI5MDQyODE1MjE0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUFxWyAdk4
oiXIOC9PxcgjYYh71mwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQByjeQe6lr7fiIhoGdjBXYzDfkX0lGGvMIhRh57G1bbceQfaYdZd7Ptc0jl
bpycKGaTvhUdkpMOiV2Hi9dOOYawkdhyJDstmDNKu6P9+b6Kak8He5z3NU1tUR2Y
uTwcz7Ye8Nldx//ws3raErfTI7D6s9m63OX8cAJ/f8bNgikwpw==
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL2bOgb+dq9rMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTAwNTdaGA8yMTk1MDQwMzA5MDA1N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAmRcyLWraysQS8yDC1b5Abs3TUaJabjqWu7d5gHik5Icd6dKl8EYpQSeS
vz6pLhkgO4xBbCRGlgE8LS/OijcZ5HwdrxBiKbicR1YvIPaIyEQQvF5sX6UWkGYw
Ma5IRGj4YbRmJkBybw+AAV9Icb5LJNOMWPi34OWM+2tMh+8L234v/JA6ogpdPuDr
sM6YFHMZ0NWo58MQ0FnEj2D7H58Ti//vFPl0TaaPWaAIRF85zBiJtKcFJ6vPidqK
f2/SDuAvZmyHC8ZBHg1moX9bR5FsU3QazfbW+c+JzAQWHj2AaQrGSCITxCMlS9sJ
l51DeoZBjnx8cnRe+HCaC4YoRBiqIQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU/wHIo+r5U31VIsPoWoRVsNXGxowwgY4GA1UdIwSBhjCBg4AU/wHI
o+r5U31VIsPoWoRVsNXGxoyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL2bOgb+dq9rMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACobLvj8IxlQyORTz/9q7/VJL509/p4HAeve
92riHp6+Moi0/dSEYPeFTgdWB9W3YCNc34Ss9TJq2D7t/zLGGlbI4wYXU6VJjL0S
hCjWeIyBXUZOZKFCb0DSJeUElsTRSXSFuVrZ9EAwjLvHni3BaC9Ve34iP71ifr75
8Tpk6PEj0+JwiijFH8E4GhcV5chB0/iooU6ioQqJrMwFYnwo1cVZJD5v6D0mu9bS
TMIJLJKv4QQQqPsNdjiB7G9bfkB6trP8fUVYLHLsVlIy5lGx+tgwFEYkG1N8IOO/
2LCawwaWm8FYAFd3IZl04RImNs/IMG7VmH1bf4swHOBHgCN1uYo=
-----END CERTIFICATE-----
```

亞太區域 (台北) – ap-east-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQDrb+HArMxssTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQxNDU2MDZaGA8y
MDUwMDUyNTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtjCCASsGByqGSM44BAEwggEeAoGBAPFb5R5TBUs2N0YW
aJcOjxr4UCDrpVT3x6EeqoGFNAc1yiZw9iv++ff6wsXm0VG5XV1QkWMQcjPfK/2I
TvXh9XEOhsG7vaBHoWkb2gmIxE6C2u8TfoZu0ZmiMceFK7aYz5w/q2J9CmRs38dT
```

```
I9qgTq0LYX+VkiNYOQnn7sQXQ8CjAhUAp2Lue9SfYTmubbpH+3j91Sivc+cCgYAi
DiyGU5R5VgiNV3mhuAyiHCft3LEmTIxos/hx/rHzgm+vCMH55tYejIrv9n7YGrIZ
MejOt8bGYxqSf0Ir/aoj3ZfHihxUzwyfyzwH7D8kAodXJNsPUTKi0O0pGl/e+nVD
JhTIFV3AwC8Tt0r2hljbjhOiyhdYy5hxWHa/WokLAQOBhAACgYAkGGHXbFr/JXol
itpS1qfiETWSRkXGHrdSmUkRumgti2y7c57TgXPwCyNlP1sTr1duy/kYDpwSwyqi
qEYfko0b/xWpDEQQPsWBgolpEYP7RIT85IxrTCrC9ohz85n7v6m9oF6BiBQGtZdk
GBN7iqemg85UaF6mXfHvl3tUYjsB2jAJBgcqhkjOOAQDAy8AMCwCFED9Ybn6fEVF
wLNjDvULmki5bPBWAhR64pxsHFRHz0ZQi+SYMWdIpEs5RQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAJp99iy2CTXhMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDVdclgpcIIsyGG51ag8srK1UMfZ+Ioh9E4PLP+cIWQcdh31fv2kV/whKaqfmif
NefBIuHrTPl6L+mJQbsfeA+TiEx1Jamg4lwQfhgQKQCmjZVMaHtK+AY57TVN6FBf
HXxT2MXzktsKxAQRGeSTFQYrHRzSYvu+GkA+37lqyWRfcwIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBABM+WL9MSb7z3bNOTMkM7ufPucvj+LQ+zEAJiuG0rcB8GI2DhGB5
8ohDPUGlq1cP7vYmDkWG/FrLqx7qgIzkbE3zfb62JpdOuX73oHgCmikMfq8ifoS1
CUxMQ36PDZBFNPzGZxv5kEORqhmGBkYCoIUCQ43jh3aALR/A/rfzfij8
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJt5hgMuWbpMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA1MjQx
NDU2MDZaGA8yMjAzMTAyOTE0NTYwNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA91FXs9Wk9kMI3lkNhjC69C6RTBpSvdLWSohPDlJKXFoaGYIDtyW1ViwO
FORAKz0koarVBx1D4fNKCxf+mhYAf//4Pxt31GzjehHNs2YVGimwhq/SNr+3AuIO
PibA+qScXamaiknE1Pxb4jisbBu2PMAG9OxU2dESNwiqBMvdJvg30YNTUITZ6r0e
MPjR1Y/EFVx/SLHW6wdvwqhNi9hz8HHdThzjCbCQPvmoLBJIKXIfosKqvauSezt1
KJSoS2jMx1OjJB2bBlQTZC8vF0WFNPoiH4jXyJ6+wMUMNSSd5b3uL5stVTMzuIz4
39Hd5YW41rmAy+6jmpGp2eXn6Td0rQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAs
gcgv+Xr67Cj08jKdAUT4GuCoGxveOFAjSDjs5y0cD+kXIT/Yj/+HKFOx5cwThT8F
```

```
TxNQ/oVuo3bGeuFEX799H0rT9Jupy0WiE1Vx81mk2M/1EDtns437edM4ofiWR8tR
Q1VNiRgz6Crr0nFgNoQa9HmR6/M+bImxevHJkWB3skvnxtA+lpKGwJs8jo6fA7AW
jr00opK0r731iBd1z/5hjIjkHn5NizAdiJEh3lVY/A3WS82aygW8LvYnKZeaBfZJ
J4NgxeflxocLFKBJD+hyf5u1hhvl7Mn/aGr5VI8/0F2oUqYo+xsHLT3keNghywO4
NNcyLcD9u16HVXhRhGvo
-----END CERTIFICATE-----
```

亞太區域 (泰國) – ap-southeast-7

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8TCCAq8CCQCOEEMWiJIJpTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIxNTI4NTZaGA8y
MDUwMDQxMzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASsGByqGSM44BAEwggEeAoGBAJWKfrIlz9+/U4wm
NDJPX00avUTdksBKX5RwHkY4o0///2G2HWwiBbxw8F1U+RzCXr80gSoe4+/SCSg3
uDuhJyzN5r4jr7c590CAgsiGdWERs7l4gLQCP504feU4TEjJoq9A8MfRZeqMj7Ug
fT5InTUW7S1/r98ddG/rBHMsJwEbAhUAneIckZ1YeyGcCeL32lujMhj+g68CgYA+
RsyTXCy3Tug5aHun51IfcG2d+pn5K/tv/N3WUR18RplVctrLhwIOUoAsDWPWtxNV
sODetezAyo759CK43JAmYGzXKbRFUHm3n46jP88tSdhuSeJhc/D4l5/0+2L7ndXp
L3+W4N05NdiKSGI8e8t452wTZv/RODivPZ3RtuCDywOBhQACgYEAjVS1Huwzdn1J
+kpd2Rcbe1BAGkmvk5sUu0KhyttqIB1kxTxeWZgsdO8REZSC5gJOwkcGFvXnb3DY
+Ms45r0e0s0rx2FFYjqMLwyRpK9wUjJfSXeJMa9iLQEXuyzBz6zPgfemXbS3zNq/
eoJ9ztIwjB9DMoKL+E1vSLsTGhehqRowCQYHKoZIzjgEAwMxADAuAhUAmljDM3c9
hf0j4XbMjjpnzrx1xhkCFQCASR9pgFNgLK8y6Kojj+P1KJkrSQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAIuIHAhL0xWcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCajgAeOauwvqGDLrHvxujnZ1BnkMzwjrycMUTkj8jqNtWoDQWUJVNPZJILosEU
VwK2I3oNkEsx/ryl9XfXcNNceoYfVEPzkTzozrZyuOG66FWtUU1LKeJ7h9/rX0Zd
9lZEokrdr6dLPt9FsHWaK5ExlUnWBjNltcQLkkKqoeYaFwIDAQABMA0GCSqGSIb3
```

```
DQEBCwUAA4GBAE4G5G+FvKTsX3T7BEcId7f5LSCc2J9gZRDiWn2oTr40CrBMOzJT
KsWr9W89YXW3gaGWltzc0WCwYQbJZgAkuEAZItJjbhdnns87ZbsFO+NZhc6gDtjA
WC3dPlSB9b6rfVoVW9O6Xwa7iNXZo8ddYVJ/ZOIv/totUz9qJt4DmmKk
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANaQIrYcijxaMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDA0MTIx
NTI4NTZaGA8yMjAzMDkxNzE1Mjg1NlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAxtMGK4r6WerVtMfpRRCY3DMP9Q/s4jnRIqolPaxeT99BAFp03HXyOrTz
WjggEHArRGGeDowFkL7Vj+QNY7ran2lnYVlAQ7Ow6OLl6WttT6lMsgsLUIcsED6G
eJ1Ko+ovT1qvmuush1U5R2pHkcCJ1IT2s6uQff85l066K7dCFpaoA1QHUk9zACHj
i0DtWuTAWBXirAZSGlmtP6uj1Aw6y5kACDTjIvZJeW/kmswfApTKaJ56eNkJsUBU
NpKwt3um1BMZduy0cjUv0Sc3dIbLNyfOdyPn8tX5u5ck0EPdtBB5WjEh71IXdZJD
oaBREV7AHg5ERPQsm0BqSvMQt09KiwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
qJyZbv/GUQnm8dEyVYgilQr3Yu35n8sfTwwZ1OJzIVce7NBPWePMfYii/tvGoKNp
KwFa/Vugbn1EwqpTEKqkPE1dOZvnUa97XxhYkoW5U7sgdKCANi1KWjywn7MiJ3Eg
j4gdqYK8wxvHi21ppqr572UO77uA8YMBoBT/CyQzWYUSmbqwKnnzaQBAZeO2iAk
VHuNU+9UsntNB676gRl9ag3Wfxq3yx5Ee1CeQf+US3HJn/pKk1H8dExXmBHvHwO6
GKUVNPN1rxfjTiaSt8wuO80uElAnyHIM3VOR8rJO7PKsobyEeJV0WI1hURO+wxpL
h3IsW7iBrFVvhX5xx7ZU
-----END CERTIFICATE-----
```

亞太區域 (東京) — ap-northeast-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
```

```
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULgwDh7TiDrPPBJwscqDwiBHkEFQwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTEyMjMxMFoXDTI5MDQyODEyMjMxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULgwDh7Ti
DrPPBJwscqDwiBHkEFQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBtjAglBde1t4F9EHCZOj4qnY6Gigy07Ou54i+lR77MhbpzE8V28Li9l+YT
QMIn6SzJqU3/fIycIro1OVY1lHmaKYgPGSEZxBenSBHfzwDLRmC9oRp4QMe0BjOC
gepj1lUoiN7OA6PtA+ycNlsP0oJvdBjhvayLiuM3tUfLTrgHbw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAL9KIB7Fgvg/MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTAwMjVaGA8yMTk1MDExNzA5MDAyNVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAz0djWUcmRW85C5CiCKPFiTIvj6y2OuopFxNE5d3Wtab10bm06vnXVKXu
tz3AndG+Dg0zIL0gMlU+QmrSR0PH2PfV9iejfLak9iwdm1WbwRrCEAj5VxPe0Q+I
KeznOtxzqQ5Wo5NLE9bA61sziUAFNVsTFUzphEwRohcekYyd3bBC4v/RuAjCXHVx
```

```
40z6AIksnAOGN2VABMlTeMNvPItKOCIeRLlllSqXX1gbtL1gxSW40JWdF3WPB68E
e+/1U3F7OEr7XqmNODOL6yh92QqZ8fHjG+afOL9Y2Hc4g+P1nk4w4iohQOPABqzb
MPjK7B2Rze0f9OEc51GBQu13kxkWWQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU5DS5IFdU/QwYbikgtWvkU3fDwRgwgY4GA1UdIwSBhjCBg4AU5DS5
IFdU/QwYbikgtWvkU3fDwRihYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAL9KIB7Fgvg/MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAG/N7ua8IE9IMyno0n5T57erBvLTOQ79fIJN
Mf+mKRM7qRRsdg/eumFft0rLOKo54pJ+Kim2cngCWNhkzctRHBV567AJNt4+ZDG5
hDgV0IxWO1+eaLE4qzqWP/9VrO+p3reuumgFZLVpvVpwXBBeBFUf2drUR14aWfI2
L/6VGINXYs7uP8v/2VBS7r6XZRnPBUy/R4hv5efYXnjwA9gq8+a3stC2ur8m5ySl
faKSwE4H320yAyaZWH4gpwUdbUlYgPHtm/ohRtiWPrN7KEG5Wq/REzMIjZCnxOfS
6KR6PNjlhxBsImQhmBvz6j5PLQxOxBZIpDoiK278e/1Wqm9LrBc=
-----END CERTIFICATE-----
```

加拿大 (中部) — ca-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUIrLgixJJB5C4G8z6pZ5rB0JU2aQwDQYJKoZIhvcNAQEL
```

```
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1MzU0M1oXDTI5MDQyODE1MzU0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUIrLgixJJ
B5C4G8z6pZ5rB0JU2aQwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBHiQJmzyFAaSYs8SpiRijIDZW2RIo7qBKb/pI3rqK6yOWDlPuMr6yNI81D
IrKGGftg4Z+2KETYU4x76HSf0s//vfH3QA57qFaAwddhKYy4BhteFQl/Wex3xTlX
LiwI07kwJvJy3mS6UfQ4HcvZy219tY+0iyOWrz/jVxwq7TOkCw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAJNKhJhaJOuMMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA3Mjkx
MTM3MTdaGA8yMTk2MDEwMjExMzcxN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAhDUh6j1ACSt057nSxAcwMaGr8Ez87VA2RW2HyY8l9XoHndnxmP50Cqld
+26AJtltlqHpI1YdtnZ6OrVgVhXcVtbvte0lZ3ldEzC3PMvmISBhHs6A3SWHA9ln
InHbToLX/SWqBHLOX78HkPRaG2k0COHpRy+fG9gvz8HCiQaXCbWNFDHZev9OToNI
xhXBVzIa3AgUnGMalCYZuh5AfVRCEeALG60kxMMC8IoAN7+HG+pMdqAhJxGUcMO0
LBvmTGGeWhi04MUZWfOkwn9JjQZuyLg6B1OD4Y6s0LB2P1MovmSJKGY4JcF8Qu3z
xxUbl7Bh9pvzFR5gJN1pjM2n3gJEPwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAJ
UNKM+gIIHNk0G0tzv6vZBT+o/vt+tIp8lEoZwaPQh1121iw/I7ZvhMLAigx7eyvf
IxUt9/nf8pxWaeGzi98RbSmbap+uxYRynqe1p5rifTamOsguuPrhVpl12OgRWLcT
rjg/K60UMXRsmg2w/cxV45pUBcyVb5h6Op5uEVAVq+CVns13ExiQL6kk3guG4+Yq
LvP1p4DZfeC33a2Rfre2IHLsJH5D4SdWcYqBsfTpf3FQThH0l0KoacGrXtsedsxs
9aRd7OzuSEJ+mBxmzxSjSwM84Ooh78DjkdpQgv967p3d+8NiSLt3/n7MgnUy6WwB
KtDujDnB+ttEHwRRngX7
-----END CERTIFICATE-----
```

加拿大西部 (卡加利) — ca-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAYPouptUMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAMITzTJUa6cBsIfdHN69zW/
aHjUB4r1ZfKb1FMhIp9EZtEf5n+06oXjUG2+dKRSlFQeEK333ehNZsPd6uqey6TYKtHpFb5XRLS8BpqB
+7gnbAd0CBZM5o4NWesSQ1GLnTdQcGZkYG/
QESkbadoCXQTifCujJE682hTDLIVt1d4ewwCQYHKoZIzjgEAwMvADAsAhRJc4gRS/HWTkCR2MESaQEe/
jOMNQIUNoTwLvuPrmGPupPlGiHe0veZiO8=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUCVFIhsz9lHA88TjOEChaY+KhHjkwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI2MDYxNzA5MjE0M1oXDTMxMDYxNjA5MjE0M1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDK1kIcG5Q6adBXQM75GldfTSiXl7tn54p10TnspI0ErDdb2B6q
2Ji/v4XBVH13ZCMgqlRHMqV8AWI5iO6gFn2A9sN3AZXTMqwtZeiDdebq3k6Wt7ie
YvpXTg0qvgsjQIovRZWaBDBJy9x8C2hW+w9lMQjFHkJ7Jy/PHCJ69EzebQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUwCJ9341XAe3MNiXb/yszt4om
ccIwgZkGA1UdIwSBkTCBjoAUwCJ9341XAe3MNiXb/yszt4omccKhYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUCVFIhsz9
lHA88TjOEChaY+KhHjkwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCuorFtp2WcEreUMNlL44zF8JDSkju0ik27J4OdKRow6tPvqYDmyGTvybWh
N4GtOuNEhWzmj4E7eMHzRslxaHXNhxIa8l+M/GRaq9jJwAb6aU9YWcVH9FT/6Jr0
q6myzukFBNLZ9zCrrclp8dOFdGf0310RDpmGmF3qzU0MlHh+Iw==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALyTn5IHrIZjMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMzEyMDcx
NTM3MDFaGA8yMjAzMDUxMzE1MzcwMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1GP5os424BJmGPCKOSgOclP7lzUiB85du03M4hfjzSOszsBpmBGFDLz1
owYHtIxlq3+Vi1Lt5Qlx3id/ov1QyaBPFWXVeklHVXy9vieCcI3TdjGjTl1W/8MM
m3X26QPcsnHM/Kk2wJ7s186MrqmdSsp3SCPpxv4vEG2Q9yR2bXY41hpc2rWlW8qU
D0JGXluvmmAdFnto20l1XWZ6xFen1h60DRugek/ufCbN+ljKy0xLqPoavH0Ybjsb
UpsAsBs7phaoN+X/5hIERfbp5LfVnqq54pNG5KNu4KynfW9+kA/WS4cJ6FTTN5t+
y0P1HvcL+BL2RuDy6T2bB21xw5WqtQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQURTvu/Dd4zDnmS5G5CfVLnmUBNOswgY4GA1UdIwSBhjCBg4AURTvu
/Dd4zDnmS5G5CfVLnmUBNOuhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALyTn5IHrIZjMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFt523A3Aug6/F8xxyITgA8gkU0btFhlXNSP
U4U2OQ9n0tWI9WqnKNWH3KBxwY5EPitU6b3LM4xc9lDWpz7h2Pto+WhxP9LVKe6f
r8r7teTLCVZ7cfYZHzHg+flZjVpAgzE5BVfrRlj3QKpv0hYT3J1wMtI++Vorq5Nf
aPjzedehJLhmZVALwnfqfLrgv6/gmraP9Vmoa8U4D6AljNiQGYaLwyoPoRm3bUs2
v1Mh9GkEQ1b9+1pFXcqgzJJTGRuiPCyPbECI79FAnx5JM/CkGJV8HlOmjIW1qkK1
Y2qT7wzErrKLJyB53Pw15BdIM1onbDAQreZb0yZQLdoEl/tx7Uk=
-----END CERTIFICATE-----
```

中國 (北京) — cn-north-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
```

```
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAOtrM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvVBz+WQNdPiM9S+aUULOQEriTmNDUrjLWLr7SfaOJScBzis5D5ju0jh1
+qJdkbuGKtFX5OTWTm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
```

```
sgEi6+LMlCN9TwnRKOToEabmDKorss4zFl7VSsbQJwcBSfOcIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9KxsOrcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh4ObXlFN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kcijZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWYhyYIEShFlP1qDHs7F4L46/5lc4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcROll5ag8d/JXGzcTCnlDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEIOtbH1P+g5yll3bI2leYE6Tm8
LKbyfK/532xJPqO9abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

中國 (寧夏) — cn-northwest-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
```

```
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPu4ssY3BlzcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsOiGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBoL3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWMd90qvTfiNrT6vewP813QdJ3EENZOx4ERcf/Wd22tV72kxD
yw1Q3I1OMH4bOItGQAxU5OtXCjBZEEUZooOkU8RoUQOU2Pql4NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKKtf/CsSJ1F
w3qXqFJQA0VWsqjFyHXFI32I/GOupwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSQsuS8E0COpcFJFxP2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uXls35
qQraczUJ9EXDhrv7VmngIk9H3YsxYrlDGEqh/oz4Ze4ULOgnfkauanHikk+BUEsg
/jsTD+7e+niEzJPihHdsvKFDlud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXcOokwu6l6kfzigGkJBxkcq4gre3szZFdCQcUioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
-----END CERTIFICATE-----
```

歐洲 (法蘭克福) — eu-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUFD5GsmkxRuecttwsCG763m3u63UwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE1NTUyOVoXDTI5MDQyODE1NTUyOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIEExMQ4IUFD5Gsmkx
RuecttwsCG763m3u63UwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBBh0WaXlBsW56Hqk588MmJxsOrvcKfDjF57RgEDgnGnQaJcStCVWDO9UYO
JX2tdsPw+E7AjDqjsuxYaotLn3Mr3mK0sNOXq9BljBnWD4pARg89KZnZI8FN35HQ
O/LYOVHCknuPL123VmVRNs51qQA9hkPjvw21UzpDLxaUxt9Z/w==
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAKD+v6LeR/WrMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
OTA4MTlaGA8yMTk1MDExNzA5MDgxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAka8FLhxs1cSJGK+Q+q/vTf8zVnDAPZ3U6oqppOW/cupCtpwMAQcky8DY
Yb62GF7+C6usniaq/9W6xPn/3o//wti0cNt6MLsiUeHqNl5H/4U/Q/fR+GA8pJ+L
npqZDG2tFi1WMvvGhGgIbScrjR4VO3TuKy+rZXMYvMRk1RXZ9gPhk6evFnviwHsE
jV5AEjxLz3duD+u/SjPp1vloxe2KuWnyC+EKInnka909sl4ZAUh+qIYfZK85DAjm
GJP4W036E9wTJQF2hZJrzsiB1MGyC1WI9veRISd30izZZL6VVXLXUtHwVHnVASrS
zZDVpzj+3yD5hRXsvFigGhY0FCVFnwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUxC2l6pvJaRflgu3MUdN6zTuP6YcwgY4GA1UdIwSBhjCBg4AUxC2l
6pvJaRflgu3MUdN6zTuP6YehYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAKD+v6LeR/WrMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAIK+DtbUPppJXFqQMv1f2Gky5/82ZwgbbfXa
HBeGSii55b3tsyC3ZW5ZlMJ7Dtnr3vUkiWbV1EUaZGOUlndUFtXUMABCb/coDndw
CAr53XTv7UwGVNe/AFO/6pQDdPxXn3xBhF0mTKPrOGdvYmjZUtQMSVb9lbMWCFfs
w+SwDLnm5NF4yZchIcTs2fdpoyZpOHDXy0xgxO1gWhKTnYbaZOxkJvEvcckxVAwJ
obF8NyJla0/pWdjhlHafEXEN8lyxyTTyOa0BGTuYOBD2cTYYynauVKY4fqHUkr3v
Z6fboaHEd4RFamShM8uvSu6eEFD+qRmvqlcodbpsSOhuGNLzhOQ=
-----END CERTIFICATE-----
```

歐洲 (愛爾蘭) — eu-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
```

```
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUakDaQ1Zqy87Hy9ESXA1pFC116HkwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MTgxMFoXDTI5MDQyODE2MTgxMFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUakDaQ1Zq
y87Hy9ESXA1pFC116HkwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQADIkn/MqaLGPuK5+prZZ5Ox4bBZLPtreO2C7r0pqU2kPM2lVPyYYydkvP0
lgSmmsErGu/oL9JNztDe2oCA+kNy17ehcsf8cw0uP861czNFKCeU8b7FgBbL+sIm
qi33rAq6owWGi/5uEcfCR+JP7W+oSYVir5r/yDmWzx+BVH5S/g==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOrmqHuaUt0vMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEwMjkw
OTA2MTlaGA8yMTk1MDQwMzA5MDYxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjE7nVu+aHLtzp9FYV25Qs1mvJ1JXD7J0iQ1Gs/RirW9a5ZECCtc4ssnf
zQHq2JRVr0GRchvDrbm1HaP/avtFQR/Thvfltwu9AROVT22dUOTvERdkNzveoFCy
```

```
hf52Rqf0DMrLXG8ZmQPPXPDFAv+sVMWCDftcChxRYZ6mP9O+TpgYNT1krD5PdvJU
7HcXrkNHDYqbsg8A+Mu2hzl0QkvUET83Csg1ibeK54HP9w+FsD6F5W+6ZSHGJ88l
FI+qYKs7xsjJQYgXWfEt6bbckWs1kZIaIOyMzYdPF6ClYzEec/UhIe/uJyUUNfpT
VIsI5OltBbcPF4c7Y20jOIwwI2SgOQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUF2DgPUZivKQR/Zl8mB/MxIkjZDUwgY4GA1UdIwSBhjCBg4AUF2Dg
PUZivKQR/Zl8mB/MxIkjZDWhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOrmqHuaUt0vMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGm6+57W5brzJ3+T8/XsIdLTuiBSe5ALgSqI
qnO5usUKAeQsa+kZIJPyEri5i8LEodh46DAF1RlXTMYgXXxl0YggX88XPmPtok17
l4hib/D9/lu4IaFIyLzYNSzsETYWKWoGVe7ZFz60MTRTwY2u8YgJ5dec7gQgPSGj
avB0vTIgoW41G58sfw5b+wjXCsh0nROon79RcQFFhGnvup0MZ+JbljyhZUYFzCli
31jPZiKzqWa87xh2DbAyvj2KZrZtTe2LQ48Z4G8wWytJzxEeZdREe4NoETf+Mu5G
4CqoaPR05KWkdNUdGNwXewydb3+agdCgfTs+uAjeXKNdSpbhMYg=
-----END CERTIFICATE-----
```

歐洲 (倫敦) — eu-west-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
```

```
MIIDITCCAoqgAwIBAgIUCgCV/DPxYNND/swDgEKGiC5I+EwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MjkxNFoXDTI5MDQyODE2MjkxNFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUCgCV/DPx
YNND/swDgEKGiC5I+EwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQATPu/sOE2esNa4+XPEGKlEJSgqzyBSQLQc+VWo6FAJhGG9fp7D97jhHeLC
5vwfmtTAfnGBxadfAOT3ASkxnOZhXtnRna460LtnNHm7ArCVgXKJo7uBn6ViXtFh
uEEw4y6p9YaLQna+VC8Xtgw6WKq2JXuKzuhuNKSFaGGw9vRcHg==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANBx0E2bOCEPMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNjA4MTEx
NDU2NDJaGA8yMTk2MDExNTE0NTY0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEArYS3mJLGaMrh2DmiPLbqr4Z+xWXTzBWCjOwpsuHE9H6dWUUyl2Bgnu+Z
d8QvW306Yleec45M4F2RA3J4hWHtShzsMlOJVRt+YulGeTf9OCPr26QmIFfs5nD4
fgsJQEry2MBSGA9Fxq3Cw6qkWcrOPsCR+bHOU0XykdKl0MnIbpBf0kTfciAupQEA
dEHnM2J1L2iI0NTLBgKxy5PXLH9weX20BFauNmHH9/J07OpwL20SN5f8TxcM9+pj
Lbk8h1V4KdIwVQpdWkbDL9BCGlYjyadQJxSxz1J343NzrnDM0M4h4HtVaKOS7bQo
Bqt2ruopLRCYgcuFHck/1348iAmbRQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBG
wujwU1Otpi3iBgmhjMClgZyMMn0aQIxMigoFNqXMUNx1Mq/e/Tx+SNaOEAu0n2FF
aiYjvY0/hXOx75ewzZvM7/zJWIdLdsgewpUqOBH4DXFhbSk2TxggSPb0WRqTBxq5
Ed7F7+7GRIeBbRzdLqmISDnfqey8ufW0ks51XcQNomDIRG5s9XZ5KHviDCar8FgL
HngBCdFI04CMagM+pwTO9XN1Ivt+NzUj208ca3oP1IwEAd5KhIhPLcihBQA5/Lpi
h1s3170z1JQ1HZbDrH1pgp+8hSI0DwwDVb3IIH8kPR/J0Qn+hvOl2HOpaUg2Ly0E
pt1RCZe+W7/dF4zsbqwK
-----END CERTIFICATE-----
```

歐洲 (米蘭) — eu-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAqwCCQCMElHPdwG37jAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTA0MjkyMDM1MjJaFw00
NTA0MjkyMDM1MjJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbYwggErBgcqhkjOOAQBMIIBHgKBgQDAkoL4YfdMI/MrQ0oL
NPfeEk94eiCQA5xNOnU7+2eVQtEqjFbDADFENh1p3sh9Q9OoheLFH8qpSfNDWn/0
ktCS909ApTY6Esx1ExjGSeQq/U+SC2JSuuTT4WFMKJ63a/czMtFkEPPnVIjJJJmT
HJSKSsVUgpdDIRvJXuyB0zdB+wIVALQ3OLaVGdlPMNfS1nD/Yyn+32wnAoGAPBQ3
7XHg5NLOS4326eFRUT+4ornQFjJjP6dp3pOBEzpImNmZTtkCNNUKE4Go9hv5T4lh
R0pODvWv0CBupMAZVBP9ObplXPCyEIZtuDqVa7ukPOUpQNgQhLLAqkigTyXVOSmt
ECBj9tu5WNP/x3iTZTHJ+g0rhIqpgh012UwJpKADgYQAAoGAV1OEQPYQUg5/M3xf
6vE7jKTxxyFWEyjKfJK7PZCzOIGrE/swgACy4PYQW+AwcUweSlK/Hx2OaZVUKzWo
wDUbeu65DcRdw2rSwCbBTU342sitFo/iGCV/Gjf+BaiAJtxniZze7J1ob8vOBeLv
uaMQmgOYeZ5e0fl04GtqPl+lhcQwCQYHKoZIzjgEAwMwADAtAhQdoeWLrkm0K49+
AeBK+j6m2h9SKQIVAIBNhS2a8cQVABDCQXVXrc0tOmO8
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAOZ3GEIaDcugMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTEwMjQx
NTE5MDlaGA8yMTk5MDMyOTE1MTkwOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCjiPgW3vsXRj4JoA16WQDyoPc/eh3QBARaApJEc4nPIGoUolpAXcjFhWplo2O+
ivgfCsc4AU9OpYdAPha3spLey/bhHPRi1JZHRNqScKP0hzsCNmKhfnZTIEQCFvsp
DRp4zr91/WS06/flJFBYJ6JHhp0KwM81XQG59lV6kkoW7QIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAGLLrY3P+HH6C57dYgtJkuGZGT2+rMkk2n81/abzTJvsqRqGRrWv
XRKRXlKdM/dfiuYGokDGxiC0Mg6TYy6wvsR2qRhtXW1OtZkiHWcQCnOttz+8vpew
wx8JGMvowtuKB1iMsbwyRqZkFYLcvH+Opfb/Aayi20/ChQLdI6M2R5VU
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
```

```
MIIDOzCCAiOgAwIBAgIJAO/+DgYF78KwMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTA0Mjky
MDM1MjJaGA8yMTk4MTAwMjIwMzUyMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAv1ZLV+Z/P6INq+R1qLkzETBg7sFGKPiwHekbpuB6lrRxKHhj8V9vaReM
lnv1Ur5LAPpMPYDsuJ4WoUbPYAqVqyMAo7ikJHCCM1cXgZJefgN6z9bpS+uA3YVh
V/0ipHh/X2hc2S9wvxKWiSHu6Aq9GVpqL035tJQD+NJuqFd+nXrtcw4yGtmvA6wl
5Bjn8WdsP3xOTKjrByYY1BhXpP/f1ohU9jE9dstsRXLa+XTgTPWcWdCS2oRTWPGR
c5Aeh47nnDsyQfP9gLxHeYeQItV/BD9kU/2Hn6mnRg/B9/TYH8qzlRTzLapXp4/5
iNwusrTNexGl8BgvAPrfhjDpdgYuTwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQB7
5ya11K/hKgvaRTvZwVV8GlVZt0CGPtNvOi4AR/UN6TMm51BzUB5nurB4z0R2MoYO
Uts9sLGvSFALJ4otoB77hyNpH3drttU1CVVwal/yK/RQLSon/IoUkaGEbqalu+mH
nYad5IG4tEbmepX456XXcO58MKmnczNbPyw3FRzUZQtI/sf94qBwJ1Xo6XbzPKMy
xjL57LHIZCssD+XPifXay69OFlsCIgLim11HgPkRIHEOXLSf3dsW9r+4CjoZqB/Z
jj/P4TLCxbYCLkvglwaMjgEWF40Img0fhx7yT2X92MiSrs3oncv/IqfdVTiN8OXq
jgnq1bf+EZEZKvb6UCQV
-----END CERTIFICATE-----
```

歐洲 (巴黎) — eu-west-3

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUaC9fX57UDr6u1vBvsCsECKBZQyIwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MzczOFoXDTI5MDQyODE2MzczOFowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUaC9fX57U
Dr6u1vBvsCsECKBZQyIwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCARv1bQEDaMEzYI0nPlu8GHcMXgmgA94HyrXhMMcaIlQwocGBs6VILGVhM
TXP2r3JFaPEpmXSQNQHvGA13clKwAZbni8wtzv6qXb4L4muF34iQRHF0nYrEDoK7
mMPR8+oXKKuPO/mv/XKo6XAV5DDERdSYHX5kkA2R9wtvyZjPnQ==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALWSfgHuT/ARMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNzA1MzEx
MTE4MTZaGA8yMTk2MTEwMzExMTgxNlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy5V7KDqnEvF3DrSProFcgu/oL+QYD62b1U+Naq8aPuljJe127Sm9WnWA
EBdOSASkOaQ9fzjCPoG5SGgWKxYoZjsevHpmzjVv9+Ci+F57bSuMbjgUbvbRIFUB
bxQojVoXQPHgK5v433ODxkQ4sjRyUbf4YV1AFdfU7zabC698YgPVOExGhXPlTvco
8mlc631ubw2g52j0lzaozUkHPSbknTomhQIvO6kUfX0e0TDMH4jLDG2ZIrUB1L4r
OWKG4KetduFrRZyDHF6ILZu+s6ywiMicUd+2UllDFC6oas+a8D11hmO/rpWU/ieV
jj4rWAFrsebpn+Nhgy96iiVUGS2LuQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDE
iYv6FQ6knXCg+svlcaQG9q59xUC5z8HvJZ1+SxzPKKC4PKQdKvIIfE8GxVXqlZG1
cl5WKTFDMapnzb9RV/DTaVzWx3cMYT77vm1Hl1XGjhx611CGcENH1egI3lOTILsa
+KfopuJEQQ9TDMAIkGjhA+KieU/U5Ctv9fdej6d0GC6OEuwKkTNzPWue6UMq8d4H
2xqJboWsE1t4nybEosvZfQJcZ8jyIYcYBnsG13vCLM+ixjuU5MVVQNMY/gBJzqJB
V+U0QiGiuT5cYgY/QihxdHt99zwGaE0ZBC7213NKrlNuLSrqhDI2NLu8NsExqOFy
OmY0v/xVmQUQl26jJXaM
```

```
-----END CERTIFICATE-----
```

歐洲 (西班牙) — eu-south-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCAq
+gAwIBAgIGAXjwLk46MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAGG2m8EKmaf5qQqj3Z
+rzSaTaXE3B/R/4A2VuGqRYR7MljPtwdmU6/3CPjCACcZmTIcOAKbFiDHqadQgBZXfzGpzw8Zo
+eYmmk5fXycgnj57PYH1dIWU6I7mCbAah5MZMcmHaTmIsomGrhcnWB8d8qOU7oZ0UWK4lbiAQs1MihoUwCQYHKoZIzjg
WmbaU7YM5GwCFCvIJOesO5hZ8PHC52dAR8WWC6oe
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUbtiu7xW+c3P65VHqW/gxSFWHT8kwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI2MDYwOTA4NDMzMVoXDTMxMDYwODA4NDMzMVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDB/VvR1+45Aey5zn3vPk6xBm5o9grSDL6D2iAuprQnfVXn8CIb
SDbWFhA3fi5ippjKkh3sl8VyCvCOUXKdOaNrYBrPRkrdHdBuL2Tc84RO+3m/rxIU
Z2IK1fDlC6sWAjddf6sBrV2w2a78H0H8EwuwiSgttURBjwJ7KPPJCqaqrQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUcZuHrkKFfuCOW/tfRxCPB7Ro
XQQwgZkGA1UdIwSBkTCBjoAUcZuHrkKFfuCOW/tfRxCPB7RoXQShYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUbtiu7xW+
c3P65VHqW/gxSFWHT8kwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQANhLKjQrxeWPzqlTkLK8b44iyITsV0E+o8+gJN6tKmC0Cgapp1+rCcYDzt
4R3uglYdoRDRrEPJVLjS8XcX5cefteIJRKJGm06d+fTvgzVY11IXo77tnzJF356m
Nwunpudg2r1diQCWQKKftqbdTsQxPGPVTVOFv+tXyW1s1TYj9Q==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALWSmO6DvSpQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
MzU4NDNaGA8yMjAxMTIyMjEzNTg0M1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuAAhuSpsHC00/fD2zNlBDpNLRndi9qbHsNeuz3WqN7Samj2aSrM2hS+i
hUxx0BspZj0tZCOsbpPZ+i74NOEQtFeqQoEGvKhB1nJiF4y5I81HDhs5qHvoIivm
7rbbik3zgm1PqS/DmDjVQaXPcD31Rd9ILwBmWEwJqHigyNVlxYtCzTQcrlBrvNZM
dnNgCDAdX/HBEFxx9O12xeu0bSt0s+PJWZ1RTbYrNe7LIH6ntUqHxP/ziQ5trXEZ
uqy7aWk1L8uK4jmyNph0lbaqBa3Y6pYmU1nC27UE4i3fnPB0LSiAr+SrwVvX1g4z
ilo8kr+tbIF+JmcgYLBvO8Jwp+EUqQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUwvGzKJL9A5LReJ4Fxo5K6I20xcowgY4GA1UdIwSBhjCBg4AUwvGz
KJL9A5LReJ4Fxo5K6I20xcqhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALWSmO6DvSpQMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAJAZd31jyoTGLawAD2+v/vQsaB9vZIx5EImi
G8YGkd61uFWeNhAmtrwyE/i6FDSIphDrMHBkvw/D3BsqK+Ev/JOK/VYuaYDx/8fp
H4cwp9jC57CXzdIDREWNf6M9PsHFg2WA9XNNtClOZL5WJiJwel8eDSg+sqJUxE0l
MW+QChq/20F6niyaRK4bXrZq14as7h+F9u3A9xHEOVP7Zk9C2ehrBXzCMLSDt3GV
fEuMea2RxMhozwz34Hkdb6j18qoCfygubulovRNQjKw/cEmgPR16KfZPP5caILVt
9qkYPvePmbiVswZDee73cDymJYxLqILpOZwyXvUH8StiH42FHZQ=
-----END CERTIFICATE-----
```

歐洲 (斯德哥爾摩) — eu-north-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
```

```
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUN1c9U6U/xiVDFgJcYKZB4NkH1QEwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2MDYwM1oXDTI5MDQyODE2MDYwM1owXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUN1c9U6U/
xiVDFgJcYKZB4NkH1QEwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBTIQdoFSDRHkpqNPUbZ9WXR2O5v/9bpmHojMYZb3Hw46wsaRso7STiGGX/
tRqjIkPUIXsdhZ3+7S/RmhFznmZc8e0bjU4n5vi9CJtQSt+1u4E17+V2bF+D3h/7
wcfE0l3414Q8JaTDtfEf/aF3F0uyBvr4MDMd7mFvAMmDmBPSlA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALc/uRxg++EnMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
NDAwMTFaGA8yMTk3MDkxMzE0MDAxMVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzwCGJEJIxqtr2PD2a1mA6LhRzKhTBa1AZsg3eYfpETXIVlrpojMfvVoN
qHvGshWLgrGTT6os/3gsaADheSaJKavxwX3X6tJA8fvEGqr3a1C1MffH9hBWbQqC
LbfUTAbkwis4GdTUwOwPjT1Cm3u9R/VzilCNwkj7iQ65AFAI8Enmsw3UGldEsop4
yChKB3KW3WI0FTh0+gD0YtjrqqYJxpGOYBpJp5vwdd3fZ4t1vidmDMs7liv4f9Bx
```

```
p0oSmUobU4GUlFhBchK1DukICVQdnOVzdMonYm7s+HtpFbVHR8yf6QoixBKGdSal
mBf7+y0ixjCn0pnC0VLVooGo4mi17QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQDG
4ONZiixgk2sjJctwbyD5WKLTH6+mxYcDw+3y/F0fWz561YORhP2FNnPOmEkf0Sl/
Jqk4svzJbCbQeMzRoyaya/46d7UioXMHRZam5IaGBhOdQbi97R4VsQjwQj0RmQsq
yDueDyuKTwWLK9KnvI+ZA6e6bRkdNGflK4N8GGKQ+fBhPwVELkbT9f16OJkezeeN
S+F/gDADGJgmPXfjogICb4Kvshq0H5Lm/xZlDULF2g/cYhyNY6EOI/eS5m1I7R8p
D/m6WoyZdpInxJfxW616OMkxQMRVsruLTNGtby3u1g6ScjmpFtvAMhYejBSdzKG4
FEyxIdEjoeO1jhTsck3R
-----END CERTIFICATE-----
```

歐洲 (蘇黎世) — eu-central-2

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXiKJnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAYNjaCNg/
cfgQOO11BUj5ClUulqwZ9Q+SfDzPZh9D2C0VbiRANiZoxrV8RdgmzzC5T7VcriVwjwvta2Ch//
b+sZ86E5h0XWWr+BeEjD9cu3eDj12XB5sWEbNHNx49p5Tmtu5r2LDtlL8X/
Rpfalu2Z2OJgjFJWGf7hRwxe456n
+lowCQYHKoZIzjgEAwMvADAsAhRChsLcj4U5CVb2cp5M0RE1XbXmhAIUeGSnH+aiUQIWmPEFja+itWDufIk=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAK+7ZflHYRFQMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yNjA1MjYx
MDU0MDBaFw0zMTA1MjUxMDU0MDBaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
tpnRnabOUc9o82HBjbIBE0xlLYSUa1akp1FiV05VWiG5BjyyjujQ+wH1ihadrIES
fLy7I6AVUeXBElSu3LitcCLY7ycov6KSmQFLrCcdoaYFN6VB4ZafQE4f1XlQ+fNo
03fSGPvLeTA2umd3xpYgchIl89CxB3YvDAWnC0RLtU8CAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFDFYNKA6vm8m8PtCqgIoU4I3+EpSMIGOBgNVHSME
```

```
gYYwgYOAFDFYNKA6vm8m8PtCqgIoU4I3+EpSoWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQCvu2X5R2ERUDASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBADsVuZ+wv4qEfSnGx+MH8s7p
oBWc/w6DTMCltWl7+q6a7C7uTIymgfsbug31ClwB7O3dhy26Ae9gNzXfpRnWN7iK
chhz057HHESzeogH1rTHD2RkJE5z6KCC0F8JgEg+JQSSCuTFKzUuBkW4S8WMQ3Tj
kCniK/KM2bEKllCqvINx
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJALvT012pxTxNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTgx
NTEyMDdaGA8yMjAxMTIyMjE1MTIwN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAyn+Lsnq1ykrfYlZkk6aAAYNReNd9Iw8AUwCBkgOr2eBiBBepYxHwU85N
++moQ+j0EV2VaahBeTLShGZZS1HsyK8+cYT2QzpghIoamcYhrPXyIxlWiRQlaqSg
OFiE9bsqL3rCF5Vz+tOiTe5W/7ojfOFls6++g7ZpobwJlpMbuJepqyeHMPyjvO5A
age81lJewc4bxo2ntaW0HCqNksqfYB78j6X6kn3PFpX7FaYAwZA+Xx6C7UCY7rNi
UdQzfAo8htfJi4chz7frpUdQ9kl3IOQrsLshBB5fFUjl09NiFipCGBwi+8ZMeSn1
5qwBI01BWXPfG7WX6Owyjhmh6JtE1wIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU8HN4vvJrsZgPQeksMBgJb9xR1yYwgY4GA1UdIwSBhjCBg4AU8HN4
vvJrsZgPQeksMBgJb9xR1yahYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJALvT012pxTxNMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAGlHYDtcHpfBvdHx9HeQE8HgNugJUPdEqxun
t9U33p8VFrs+uLPtrOd9HDJEGvvs5h84EUie/oGJxRt7V1Vlid1PvHf6cRmpjgqY
YdggAVkZtY/PnFVmzf2bMVlSQPrqCl7U0zaw2Kvnj4zgX0rZyCetgrRZSUSxotyp
978Wy9ccXwVSeYG/YAr5rJpS6ZH7eRQvUY0IzwFNeaOPgOTEVpcjWlV6+MQEvsEx
W85q+s6AVr49eppEx8SLJsl0C23yB+L+t32tAveQImRWtJMpzZ5cxh/sYgDVeoC0
85H1NK/7H9fAzT1cPu1oHSnB0xYzzHGOAmXmusMfwUk8fL1RQkE=
-----END CERTIFICATE-----
```

以色列 (特拉維夫) — il-central-1

DSA

```
-----BEGIN CERTIFICATE-----
```

```
MIIC7zCCAq+gAwIBAgIGAX0QPi
+9MAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYX
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/
hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAbazCL5XXyPmcw3+oMYQUF5/9YogW6D0FZbYuyPgjOoUwWdl6fjlzWca
pq+l1ezuK2DFOzNTEyPEwwCQYHKoZIzjgEAwMvADAsAhRt1jkpXsvrS
+xTo2M9h2s2uLAhEQIUOZ2FcnTSrshF2EIdixZZwtNv66Q=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUPgjnsnqT7OaGnZgGF7/43ihPh+0wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI2MDYxMjA5MjQyOVoXDTMxMDYxMTA5MjQyOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDrc24u3AgFxnoPgzxR6yFXOamcPuxYXhYKWmapb+S8vOy5hpLo
Re4RkOrY0cM3bN07GdEMlin5mU0y1t8y3ct4YewvmkgT42kTyMM+t1K4S0xsqjXx
xS716uGYh7eWtkxrCihj8AbXN/6pa095h+7TZyl2n83keiNUzM2KoqQVMwIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU+dSlCYS89nZaff8nPOdiQJje
2PkwgZkGA1UdIwSBkTCBjoAU+dSlCYS89nZaff8nPOdiQJje2PmhYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUPgjnsnqT
7OaGnZgGF7/43ihPh+0wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQDOq6yTQkIXzAZffpLDBss1R8VZzRoLvRzsFa1xpOcRnPqIm0untEHsOGX/
q34FiGQBOVY6btNaZb7aQXDlZGX+sGvst6BYC2u6H6Zpyo0qy0mz6AGA3eVEonJ2
lKpvAwBkrGajRC8JW0Gx31UZDmAcn3gjQi5zMWFead9mJJJA3g==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAOVp1h2I9wW7MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA3MTUx
MjQ0MTJaGA8yMjAxMTIxOTEyNDQxMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
```

```
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13PkyWvl61iV/SYfO1UFO76UpDfPm2SF/Rz/o33cm699X++EYPxTnoEc
vmWeS0I7eDXc4OCUiToG0sEx0klE0CX1Z1tK6qJ+zgWQLZ9SZEC9H0NsSA6LhrHu
Nq0dzeK3LjhdfcX46/4GqdiptpdTuM4m/hOQ5yx4JMQ/n1sdpv4M5VLRWwWW9Lem
ufb79Id7O9SispxgRsz1KXIjp7N9S4BY7itSXz97uSyzTqEjWZ6mDUhTu3t21GKC
6flALGTTrG2yghEhz53rkvLsvwzjPSSlT6LIf0mrRPzHaf+EdaKoasElE1SHh+ZH
9mI81HywpE+HZ+W+5hBCvjYp90Y1fwIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQU58tN2JO+yEGq5JbIXxGi4vRVPyIwgY4GA1UdIwSBhjCBg4AU58tN
2JO+yEGq5JbIXxGi4vRVPyKhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAOVp1h2I9wW7MBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBANBN0e1EqNy4+IU2yQzMJ+Wy5ZIOtTP6GSBR
7muVY1bDeAwtNTEOpwgrZVlC7/xq5Q0LC1y0Z70hHXEf8au7qStaAoUtxzvhTAZI
NC0lwoFU56UFw4N0vZII17iqEfoqRC4PpI3OxqEJHFy0VLlvAzJoKB4QLLqDAYVA
LXCiOLoVT+y9tRYsxw5MyO0Bi6fxQIIADl2bE9xkunTN1Jkkwqo3LxNy/ryz4QWR
8K7jHUItifv4h/hxBKpHEquN8CkdvM9oeGl7I8PFrSFEpGr1euDXY0euZzzYiDBV
m6GpTJgzpVsEuIX52dPcPemwQncoIfZyhWDW85MJUnby2WTEcFo=
-----END CERTIFICATE-----
```

墨西哥 (中部) – mx-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq8CCQD4QwfTExrgxTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDExMTQwOTQ4MjRaGA8y
MDUwMTExNTA5NDgyNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASwGByqGSM44BAEwggEfAoGBAK+pJEOtD2RFlCbf
oAzONffZa42FB8G6OA/30DT81xWvB2Hnmgw7LfiELYqgfuck3YSDffNDcmiFTJuX
AU9u0NtbbxcOytcdUq4HInpHiB85I6WWWKYB66aGbvYowUPpqBZkASb1i/pYAh5W
nNYvx408008tTzqpMfZDchJHzWqBAhUA0ogoJzTw2/4pKhz9aqTsRCzRVPsCgYEA
qklRUcuU0du/bT/M6kWxYvrTGhO9KXQe+7RtbaIq4dWRsCBnO4smDY/GmI9H8ew1
LRJ9AcLGmxDm795CvVZkHNcht7PDAREagWmz2YvhLA+ev6U0RpfdlBXCck2p1CxQ
LMtoFO7DThksHIQBtnlQdHDvguKEZQz/Iobhne6Kb3wDgYQAAoGAQSTlOqFq6RtR
Jvp406XhGl+eO9SQUpevpzG3Dzbdy6EQ8PBJDOHHJvxbebvpnQssh0CnQfTkSw26
jwPy9V5zRC0ZezwnRTyGSJwiErUKDGeVekEAoNJL1USy8jKkgBajlskZrR+0JHDN
Jv3UwYIPplc4ZS2f2E7btrtlaWt/P70wCQYHKoZIzjgEAwMvADAsAhRXljPWKWYf
61DkDVdgPPHsS2/LuwIUejcGV9WuS3uPvJ6lmn4opxUGBZw=
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICPzCCAaigAwIBAgIUCmzpTTMBQYItpMC2VDYsZfIAS7IwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcMB1NlYXR0bGUxIDAeBgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQwMjEyNTAzNloXDTI5MDQwMTEyNTAzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcMB1NlYXR0bGUxIDAe
BgNVBAoMF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDfvOnCzm1iN58Nm7k6ehoy6v0lnnFI617D6CY3bfuq01RCdEQL
96+pYawJieTH8JAQKj02CAa3AeaqdXTE/pDhI/YKLreeMb4K68WMn24Wjjs6oxjB
bAmsKXtt9ihKHGBFNUhgFrNFYyA2i7ieJviwpHjQ/XgXiG2u1/t/4VydUwIDAQAB
MA0GCSqGSIb3DQEBCwUAA4GBAL5+vvj4lhaE+J5tuCqV3XJzDd97lsD4le2O2uGw
P0sGdUcRAdxzU3Bwq/hhtzNWnfwo0aCEQKmLM7xyd3nUa0VvKXLq+DDuayipWINr
OATnNxFRe99d38qHTR1dgkjZdKbbtnl6O4fgM57tVEuQJd/N4ILl9jaRcJ5Ip+9t
3y5t
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALxuE00HoJomMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEwMzEx
MTE0MzNaGA8yMjA0MDQwNjExMTQzM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA13xWucALo8M/TXbZJgHrqFqFON91dSLPE/eLbmeIJbP1pb9ICd33qKAx
HlKSrXI9b9YS3U1Pl0bF3ZgfeE/x4Y0KDDZwzpf07H8IgRittULJoNLYVKCJXWPq
Ky1qvDJX3653dUbUu9eAdVCTRgk7eKpPBLAmW27+pgAGzEYrVV3u2AvqNTonvfTU
sPgEVnAL1J490pNM85KtFynxFTWGigHkd3BHidxmLrTH4I4eRxNZ9q/3gsDW+zKt
jQlpM7JzZa2OqxsF5YQDh1fF52Emqsr+ufLeGqDLOgT1QWcqpz57AX8GqZpgZULo
itCRNXbQDzZY9FxIgpiFJv3y/qYYDQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCt
MTeH4EgqjJEjb1qm5tzXYurLprVrEVQ+PhGXJFJd3xAZyeaDVYy7kio08E2xhmHd
HtkBDty2KnOHsTQmeAJCci7d4tYXZ/1qe341wwm9OoFcO8jhIndx6FXJCgQUY4dL
AAr9HQJFWG5dMZgbi1Zuhxdio3sSoOBjL2p7QIsGNkITvCDIs/H0/szpJnyyyIqu
wmUhSel5hdy5MwOsyUKVGnAdaS5Vd9oL4kLszS9nBZ7ny6BC9odIkFAdGqQ5vM4z
vcbfOq14hjatQmJgJhksN/0Dpl78Gheq0pIhP8LTkAOEG2832nQLzCa3oxSk8otG
GJXkzzyQjse+13r8+yNJ
-----END CERTIFICATE-----
```

中東 (巴林) — me-south-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1QliO0B
8n6cLZ38VE7ZmrjZ9OQV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PHlP1WGL8IZ34BUgRTtG4TVolvp0smjkMvyRu5hIdKtzjV93Ccx15gVgyk+o1IEG
fZ2Kbw/Dd8JfoPS7KaSCmJKxXQIVAIZbIaDFRGa2qcMkW2HWASyNDl7bAoGBANtz
IdhfMq+l2I5iofY2oj3HI21Kj3LtZrWEg3W+/4rVhL3lTm0Nne1rl9yGujrjQwy5
Zp9V4A/w9w2O10Lx4K6hj34Eefy/aQnZwNdNhv/FQP7Az0fju+Yl6L13OOHQrL0z
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAAKBgQCVS7m77nuNAlZ8
wvUqcooxXMPkxJFl54NxAsAul9KP9KN4svm0O3Zrb7t2FOtXRM8zU3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGWm5EDl
86ayV3c1PEDukgQIAhQow38rQkN/VwHVeSW9DqEshXHjuQ==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDPDCCAqWgAwIBAgIJAMl6uIV/zqJFMA0GCSqGSIb3DQEBCwUAMHIxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0dGxlMSAw
HgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwRZWMyLmFt
YXpvbmF3cy5jb20wIBcNMTkwNDI2MTQzMjQ3WhgPMjE5ODA5MjkxNDMyNDdaMHIx
CzAJBgNVBAYTAlVTMRMwEQYDVQQIDApXYXNoaW5ndG9uMRAwDgYDVQQHDAdTZWF0
dGxlMSAwHgYDVQQKDBdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzEaMBgGA1UEAwwR
ZWMyLmFtYXpvbmF3cy5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALVN
CDTZEnIeoX1SEYqq6k1BV0ZlpY5y3KnoOreCAE589TwS4MX5+8Fzd6AmACmugeBP
Qk7Hm6b2+g/d4tWycyxLaQlcq81DB1GmXehRkZRgGeRge1ePWd1TUA0I8P/QBT7S
gUePm/kANSFU+P7s7u1NNl+vynyi0wUUrw7/wIZTAgMBAAGjgdcwgdQwHQYDVR0O
BBYEFILtMd+T4YgH1cgc+hVsVOV+480FMIGkBgNVHSMEgZwwgZmAFILtMd+T4YgH
1cgc+hVsVOV+480FoXakdDByMQswCQYDVQQGEwJVUzETMBEGA1UECAwKV2FzaGlu
Z3RvbjEQMA4GA1UEBwwHU2VhdHRsZTEgMB4GA1UECgwXQW1hem9uIFdlYiBTZXJ2
aWNlcyBMTEMxGjAYBgNVBAMMEWVjMi5hbWF6b25hd3MuY29tggkAyXq4hX/OokUw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOBgQBhkNTBIFgWFd+ZhC/LhRUY
4OjEiykmbEp6hlzQ79T0Tfbn5A4NYDI2icBP0+hmf6qSnIhwJF6typyd1yPK5Fqt
NTpxxcXmUKquX+pHmIkK1LKDO8rNE84jqxrxRsfDi6by82fjVYf2pgjJW8R1FAw+
```

```
mL5WQRFexbfB5aXhcMo0AA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANZkFlQR2rKqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTAyMDUx
MzA2MjBaGA8yMTk4MDcxMTEzMDYyMFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAy4Vnit2eBpEjKgOKBmyupJzJAiT4fr74tuGJNwwa+Is2vH12jMZn9Il1
UpvvEUYTIboIgISpf6SJ5LmV5rCv4jT4a1Wm0kjfNbiIlkUi8SxZrPypcw24m6ke
BVuxQZrZDs+xDUYIZifTmdgD50u5YE+TLg+YmXKnVgxBU6WZjbuK2INohi71aPBw
2zWUR7Gr/ggIpf635JLU3KIBLNEmrkXCVSnDFlsK4eeCrB7+UNak+4BwgpuykSGG
Op9+2vsuNqFeU1l9daQeG9roHR+4rIWSPa0opmMxv5nctgypOrE6zKXx2dNXQldd
VULv+WH7s6Vm4+yBeG8ctPYH5GOo+QIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBs
ZcViiZdFdpcXESZP/KmZNDxB/kktlIEIhsQ+MNn29jayE5oLmtGjHj5dtA3XNKlr
f6PVygVTKbtQLQqunRT83e8+7iCZMKI5ev7pITUQVvTUwI+Fc01JkYZxRFlVBuFA
WGZO+98kxCS4n6tTwVt+nSuJr9BJRVC17apfHBgSS8c5OWna0VU/Cc9ka4eAfQR4
7pYSDU3wSRE01cs30q34lXZ629IyFirSJ5TTOIc0osNL7vwMQYj8HOn4OBYqxKy8
ZJyvfXsIPh0Na76PaBIs6ZlqAOflLrjGzxBPiwRM/XrGmF8ze4KzoUqJEnK13O6A
KHKgfiigQZ1+gv5FlyXH
-----END CERTIFICATE-----
```

中東 (阿拉伯聯合大公國) — me-central-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7zCCAq
+gAwIBAgIGAXjXhqnnMAkGByqGSM44BAMwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgMEFdhc2hpbmd0b24gU3RhdGUxED
U4EddRIpUt9KnC7s5Of2EbdSPO9EAMMeP4C2USZpRV1AIlH7WT2NWPq/
xfW6MPbLm1Vs14E7gB00b/JmYLdrmVClpJ+f6AR7ECLCT7up1/63xhv4O1fnxqimFQ8E
+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAl2BQjxUjC8yykrmCouuEC/
BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT
+ZxBxCBgLRJFnEj6EwoFhO3zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx
+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAW+csuHsWp/7/
pv8CTKFwxsYudxuR6rbWaHCykIeAydXL9AWnphK6yp1ODEMBFl68Xq8Hp23sOWyf8moOhqCom9+0+ovuUFdpvCie86bp
TOZU568Ty1ff3dDWbdRzeNQRHodRG+XEQSizMkAreeWt4kBa+PUwCQYHKoZIzjgEAwMvADAsAhQD3Z
+XGmzKmgaLgGcVX/Qf1+Tn4QIUH1cgksBSVKbWj81tovBMJeKgdYo=
```

```
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICMzCCAZygAwIBAgIGAXjRrnDjMA0GCSqGSIb3DQEBBQUAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIDBBXYXNoaW5n
KyA6zyruJQrYy00a6wqLA7eeUzk3bMiTkLsTeDQfrkaZMfBAjGaaOymRo1C3qzE4rIenmahvUplu9ZmLwL1idWXMRX2R
+d2SeoKOKQWoc2UOFZMHYxDue7zkyk1CIRaBukTeY13/
RIrlc6X61zJ5BBtZXlHwayjQIDAQABMA0GCSqGSIb3DQEBBQUAA4GBABTqTy3R6RXKPW45FA+cgo7YZEj/
Cnz5YaoUivRRdX2A83BHuBTvJE2+WX00FTEj4hRVjameE1nENoO8Z7fUVloAFDlDo69fhkJeSvn51D1WRrPnoWGgEfr1
B+Wqm3kVEz/QNcz6npmA6
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAM4h7b1CVhqqMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yMjA0MTEx
MDE1MDNaGA8yMjAxMDkxNTEwMTUwM1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApybTWfMOhSoMpqPo72eqAmnn1dXGZM+G8EoZXzwHwT/+IHEXNB4q5N6k
tudYLre1bJxuzEw+iProSHjmb9bB9YscRTofjVhBlt35Fc+i8BaMeH94SR/eE8QO
mll8gnLNW3d62lyuhzuyv1e5wVlRqzYw+X2zRH4/wRDOCOpzjKoHIgyPKsMgwsw5
aTZhNMsGxZN9dbkf0iCGeQLDytwU/JTh/HqvSr3VfUOapTJJiyAxoCtZWgp1/7wC
RvOCSMRJobpUqxZgl/VsttwNkikSFz1wGkcYeSQvk+odbnYQckA8tdddoVI56eD4
qtREQvfpMAX5v7fcqLexl5d5vH8uZQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUOadrbTs+OhzwoAgUJ7RqQNdwufkwgY4GA1UdIwSBhjCBg4AUOadr
bTs+OhzwoAgUJ7RqQNdwufmhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAM4h7b1CVhqqMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAICTdAOGEOnII8HaGCpCB8us/hGFaLptJaAf
D5SJAyVy66/mdfjGzE1BKkKxnbxemEVUIzbRidOnyilB+pKwN3edAjTZtWdpVA0V
R/G/qQPmcVljtycBz4VC6SuOUYflGzLH1GZ6GJWbuDtFzw8r7HGdRN1wrEPe3UF2
sMpuVezqnRUdvVRoVQP4jFgNsE7kNvtn2NiPhb/CtrxpcwIQ7r6YeoHcBSheuV1Z
xZDHynC3KUprQGx1+Z9QqPrDf180MaoqAlTl4+W6Pr2NJYrVUFGS/ivYshMg574l
CPU6r4wWZSKwEUXq4BInYX6z6iclp/p/J5QnJp2mAwyi6M+I13Y=
-----END CERTIFICATE-----
```

南美洲（聖保羅）— sa-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUX4Bh4MQ86Roh37VDRRX1MNOB3TcwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDQyOTE2NDYwOVoXDTI5MDQyODE2NDYwOVowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCHvRjf/0kStpJ248khtIaN8qkDN3tkw4VjvA9nvPl2anJO+eIB
UqPfQG09kZlwpWpmyO8bGB2RWqWxCwuB/dcnIob6w420k9WY5C0IIGtDRNauN3ku
vGXkw3HEnF0EjYr0pcyWUvByWY4KswZV42X7Y7XSS13hOIcL6NLA+H94/QIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQUJdbMCBXKtvCcWdwUUizvtUF2
UTgwgZkGA1UdIwSBkTCBjoAUJdbMCBXKtvCcWdwUUizvtUF2UTihYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUX4Bh4MQ8
6Roh37VDRRX1MNOB3TcwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBnhocfH6ZIX6F5K9+Y9V4HFk8vSaaKL5ytw/P5td1h9ej94KF3xkZ5fyjN
URvGQv3kNmNJBoNarcP9I7JIMjsNPmVzqWawyCEGCZImoARxSS3Fc5EAs2PyBfcD
9nCtzMTaKO09Xyq0wqXVYn1xJsE5d5yBDsGrzaTHKjxo61+ezQ==
```

```
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJAMcyoxx4U0xxMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQw
ODU4MDJaGA8yMTk1MDExNzA4NTgwMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAw45IhGZVbQcy1fHBqzROhO8CsrDzxj/WP4cRbJo/2DAnimVrCCDs5O86
FA39Zo1xsDuJHDlwMKqeXYXkJXHYbcPWc6EYYAnR+PlLG+aNSOGUzsy202S03hT0
B20hWPCqpPp39itIRhG4id6nbNRJOzLm6evHuepMAHR4/OV7hyGOiGaV/v9zqiNA
pMCLhbh2xk0PO35HCVBuWt3HUjsgeks2eEsu9Ws6H3JXTCfiqp0TjyRWapM29OhA
cRJfJ/d/+wBTz1fkWOZ7TF+EWRIN5ITEadlDTPnF1r8kBRuDcS/lIGFwrOOHLo4C
cKoNgXkhTqDDBDu6oNBb2rS0K+sz3QIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUqBy7D847Ya/w321Dfr+rBJGsGTwwgY4GA1UdIwSBhjCBg4AUqBy7
D847Ya/w321Dfr+rBJGsGTyhYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAMcyoxx4U0xxMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBACOoWSBf7b9AlcNrl4lr3QWWSc7k90/tUZal
PlT0G3Obl2x9T/ZiBsQpbUvs0lfotG0XqGVVHcIxF38EbVwbw9KJGXbGSCJSEJkW
vGCtc/jYMHXfhx67Szmftm/MTYNvnzsyQQ3v8y3Rdah+xe1NPdpFrwmfL6xe3pFF
cY33KdHA/3PNLdn9CaEsHmcmj3ctaaXLFIzZhQyyjtsrgGfTLvXeXRokktvsLDS/
YgKedQ+jFjzVJqgr4NjfY/Wt7/8kbbdhzaqlB5pCPjLLzv0zp/XmO6k+JvOePOGh
JzGk5t1QrSju+MqNPFk3+1O7o910Vrhqw1QRB0gr1ExrviLbyfU=
-----END CERTIFICATE-----
```

AWS GovCloud （美國東部） — us-gov-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
```

```
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIULVyrqjjwZ461qelPCiShB1KCCj4wDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE1MjIzNloXDTI5MDUwNjE1MjIzNlowXDELMAkGA1UEBhMCVVMx
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IULVyrqjjw
Z461qelPCiShB1KCCj4wEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQBfAL/YZv0y3zmVbXjyxQCsDloeDCJjFKIu3ameEckeIWJbST9LMto0zViZ
puIAf05x6GQiEqfBMk+YMxJfcTmJB4Ebaj4egFlslJPSHyC2xuydHlr3B04INOH5
Z2oCM68u6GGbj0jZjg7GJonkReG9N72kDva/ukwZKgq8zErQVQ==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJALPB6hxFhay8MA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xODA0MTAx
MjMyNDlaGA8yMTk3MDkxMzEyMzI0OVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAva9xsI9237KYb/SPWmeCVzi7giKNron8hoRDwlwwMC9+uHPd53UxzKLb
pTgtJWAPkZVxEdl2Gdhwr3SULoKcKmkqE6ltVFrVuPT33La1UufguT9k8ZDDuO9C
```

```
hQNHUdSVEuVrK3bLjaSsMOS7Uxmnn7lYT990IReowvnBNBsBlcabfQTBV04xfUG0
/m0XUiUFjOxDBqbNzkEIblW7vK7ydSJtFMSljga54UAVXibQt9EAIF7B8k9l2iLa
mu9yEjyQy+ZQICTuAvPUEWe6va2CHVY9gYQLA31/zU0VBKZPTNExjaqK4j8bKs1/
7dOV1so39sIGBz21cUBec1o+yCS5SwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBt
hO2W/Lm+Nk0qsXW6mqQFsAou0cASc/vtGNCyBfoFNX6aKXsVCHxq2aq2TUKWENs+
mKmYu1lZVhBOmLshyllh3RRoL3Ohp3jCwXytkWQ7ElcGjDzNGc0FArzB8xFyQNdK
MNvXDi/ErzgrHGSpcvmGHiOhMf3UzChMWbIr6udoDlMbSIO7+8F+jUJkh4Xl1lKb
YeN5fsLZp7T/6YvbFSPpmbn1YoE2vKtuGKxObRrhU3h4JHdp1Zel1pZ6lh5iM0ec
SD11SximGIYCjfZpRqI3q50mbxCd7ckULz+UUPwLrfOds4VrVVSj+x0ZdY19Plv2
9shw5ez6Cn7E3IfzqNHO
-----END CERTIFICATE-----
```

AWS GovCloud（美國西部）— us-gov-west-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNl
cnZpY2VzIExMQzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5OO6kK/n1Lzllr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nYl4hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tkqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBcJl/U
hhy1KHVpCGl9fueQ2s6IL0CaO/buycU1CiYQk40KNHCcHfNiZbdlx1E9rpUp7bnF
lRa2v1ntMX3caRVDdbtPEWmdxSCYsYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
MXrs3IgIb6+hUIB+S8dz8/mmO0bpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIziqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDITCCAoqgAwIBAgIUe5wGF3jfb7lUHzvDxmM/ktGCLwwwDQYJKoZIhvcNAQEL
BQAwXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAO
BgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExD
MB4XDTI0MDUwNzE3MzAzMloXDTI5MDUwNjE3MzAzMlowXDELMAkGA1UEBhMCVVMx
```

```
GTAXBgNVBAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAe
BgNVBAoTF0FtYXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQCpohwYUVPH9I7Vbkb3WMe/JB0Y/bmfVj3VpcK445YBRO9K80al
esjgBc2tAX4KYg4Lht4EBKccLHTzaNi51YEGX1aLNrSmxhz1+WtzNLNUsyY3zD9z
vwX/3k1+JB2dRA+m+Cpwx4mjzZyAeQtHtegVaAytkmqtxQrSCexBxvqRqQIDAQAB
o4HfMIHcMAsGA1UdDwQEAwIHgDAdBgNVHQ4EFgQU1ZXneBYnPvYXkHVlVjg7918V
gE8wgZkGA1UdIwSBkTCBjoAU1ZXneBYnPvYXkHVlVjg7918VgE+hYKReMFwxCzAJ
BgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdT
ZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQ4IUe5wGF3jf
b7lUHzvDxmM/ktGCLwwwEgYDVR0TAQH/BAgwBgEB/wIBADANBgkqhkiG9w0BAQsF
AAOBgQCbTdpx1Iob9SwUReY4exMnlwQlmkTLyA8tYGWzchCJOJJEPfsW0ryy1A0H
YIuvyUty3rJdp9ib8h3GZR71BkZnNddHhy06kPs4p8ewF8+d8OWtOJQcI+ZnFfG4
KyM4rUsBrljpG2aOCm12iACEyrvgJJrS8VZwUDZS6mZEnn/lhA==
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJANCOF0Q6ohnuMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA5MTAx
OTQyNDdaGA8yMTk1MDIxMzE5NDI0N1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAzIcGTzNqie3f1olrrqcfzGfbymSM2QfbTzDIOG6xXXeFrCDAmOq0wUhi
3fRCuoeHlKOWAPu76B9os71+zgF22dIDEVkpqHCjBrGzDQZXXUwOzhm+PmBUI8Z1
qvbVD4ZYhjCujWWzrsX6Z4yEK7PEFjtf4M4W8euw0RmiNwjy+knIFa+VxK6aQv94
lW98URFP2fD84xedHp6ozZlr3+RZSIFZsOiyxYsgiwTbesRMI0Y7LnkKGCIHQ/XJ
OwSISWaCddbu59BZeADnyhl4f+pWaSQpQQ1DpXvZAVBYvCH97J1oAxLfH8xcwgSQ
/se3wtn095VBt5b7qTVjOvy6vKZazwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA/
S8+a9csfASkdtQUOLsBynAbsBCH9Gykq2m8JS7YE4TGvqlpnWehz78rFTzQwmz4D
fwq8byPkl6DjdF9utqZ0JUo/Fxelxom0h6oievtBlSkmZJNbgc2WYm1zi6ptViup
Y+4S2+vWZyg/X1PXD7wyRWuETmykk73uEyeWFBYKCHWsO9sI+62O4Vf8Jkuj/cie
1NSJX8fkervfLrZSHBYhxLbL+actVEo00tiyZz8GnhgWx5faCY38D/k4Y/j5Vz99
7lUX/+fWHT3+lTL8ZZK7fOQWh6NQpI0wTP9KtWqfOUwMIbgFQPoxkP00TWRmdmPz
WOwTObEf9ouTnjG9OZ20
-----END CERTIFICATE-----
```

AWS 歐洲主權雲端 — eusc-de-east-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIC8DCCAq8CCQDyckpYH0omIDAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTAlVTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEwMTgwOTM5MzdaGA8y
MDUwMTAxOTA5MzkzN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgTEFdhc2hpbmd0
b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXpvbiBXZWIg
U2VydmljZXMgTExDMIIBtzCCASsGByqGSM44BAEwggEeAoGBAJR1zx7FzSGhXd3m
U1PfwH33QEK4CHnFSnp/DUse0SdLwsvLkHw7Tq3M7IX4E+TZ6OzYZZeoVZttqbyq
OZ8ROUS7CzDWcQ0503de/ya51cTs4BiO4nv2awm1WEoV9XtQyA2JLn0Jv7g2uja0
317krS+Dk3kjs9RpWvkmiyTMu1YhAhUA69U+A26BWLVpNIZZ0uWMZJSNlIkCgYBw
w1cgrHFGEqJNhqN7Sjs2YwTP+jpp1GvhxKyce5aOh5ErVvufb5gVVoKahrrHEBms
TrBXxnfcPaTupwdmEZ9eM8a+9RS2vye//KU1kGIOMgafPEzBEXs9m7okmAQpYXnG
cVOTtx2gd/E/VfKRiMSAmam178qwvcbcijNqRdLjRQOBhQACgYEAiEFVGYVCntmy
va8c2oU5/owdMws+NZ6u9blat0ymPr/8Q+DR77CgUPJW+K0k41GdgWpVlrsjiiPw
iygzw4DW7Vsl018KcIFUTiSwpzv2in+tKNUkO5Ly0q6arbSxrXZffYe7aOZKtCCH
XWDMKOH2iCvZptMpG7fDDSc7Be8v4AgwCQYHKoZIzjgEAwMwADAtAhQyvCF3q437
RvdPXMFcMc6qrD0svAIVAMvsN3NwOY4KRYlFv9Dnok8D/ixI
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIICNjCCAZ+gAwIBAgIJAPybReXZqUSNMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNDEwMTgw
OTM5MzdaGA8yMjA0MDMyNDA5MzkzN1owXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDShbskmohorgcjN7gfApw89CEsg7praCDggaUVXF6gy9JdJ5INcv7+DkwKaAxd
NQWDsWZnxIOqPuPfyAYVP3D2/exP3Rfxj31UomLduu7S83zgiJ8VVL/GzlDELv5/
K2rTcd+580NjLSa6FYDcXDip2ir6TXxRYBYHP65dwniE+wIDAQABMA0GCSqGSIb3
DQEBCwUAA4GBAKu9t3CAtZt7c+k4bCZlmxrwgps1668p24x7QRmLxgXnwj5A2MOM
qEtxA9Q3q3WxUc0nWKaCTvRrTfWW917xsLcvVL8rI6xCjg2bb5FJ73xzcfSxHr9I
Q1mThJgLaWVIbllGJK7+EogR7gfqlPhc0Ql5bYuaozdt+r69xNuJlyR
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
```

```
MIIEEjCCAvqgAwIBAgIJAODRxRi5VXBbMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0yNTAyMjEx
MjQ0MTRaGA8yMjA0MDcyODEyNDQxNFowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAr24OkDrpRU1ffx1PAcHZbKu7fZkhfT/e/PDhSkp2FvOzJEqDUT23E2wp
K/44mhJ3d8blYtgBiRXjqRsAFu8ZbzXLc0gaQ9A5woxfx5b7iOMwneCajS9xS5qr
cE0MMEY9XwG8MCp9YBoDJqDJh74oQqcPROBq16L4k2Ys22n1khus4JpwT7VSRE1I
nnt7fGIYsUmNGM+T2E6ubG3FvkY0HbLvNCcGRMCnDYqloo2BthuSG58KbMAku7VD
hMxzzJte5yMuRpHBojQr9N0hZcSJCtSFo6FhlxCzxGzwoSKzqm99ULYL4lf+IORe
DaKoATZJvbnX0LYAKLtPr1Pz9SD9pQIDAQABo4HUMIHRMAsGA1UdDwQEAwIHgDAd
BgNVHQ4EFgQUHpdxehvJwYL2o25SXyxsSndDW48wgY4GA1UdIwSBhjCBg4AUHpdx
ehvJwYL2o25SXyxsSndDW4+hYKReMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQ4IJAODRxRi5VXBbMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDQYJKoZIhvcNAQELBQADggEBAFGcArNlGB3oRBaXohpnhrnVblhJJa6U7r1/
M5TAJCKCkziSz3lOZmZZ1YgSnQC1blhUD2rhuS8a0Mw1Ml48nLjIYMMK0tkJRKCD
LPxxEZp4EzogYj6PqfSubzm59MmBenIcJPaJ9y2uY1U4Q3PMfU6M1JqqQ0Lzplq0
n9eWpJK3+7UbtuyAzigO4DRWEBMxJBUU7f2LfY+qFjskoeHopX20cmn1vc1TGyAF
Qf8xdt1SaPKC26zJsrCkiMZJNm4aa8KqvhqBHVjV4NrGeVGCcdlmSDiL71Egk9kh
744eGQxNnhZNvePW2kVcSuh0HpAo4Q4vfpjN+jz2Ves3kjFsrgc=
-----END CERTIFICATE-----
```

EC2 執行個體的 STIG 合規

安全技術實作指南 (STIG) 是由國防資訊系統管理局 (DISA) 建立的組態強化標準，用於保護資訊系統和軟體。若要讓您的系統符合 STIG 標準，您必須安裝、設定和測試各種安全設定。

會根據下列漏洞分類套用 STIG。當您建立 STIG 強化映像或將 STIG 設定套用至現有執行個體時，可以選擇要套用強化的層級。較高層級包括所有較低層級的 STIG 設定。例如，高 (類別 I) 層級包含來自中低類別的設定。

合規層級

- 高 （類別 I)

  最嚴重的風險 包含任何可能導致機密性、可用性或完整性遺失的漏洞。

- 中 （類別 II)

  包含任何可能導致機密性、可用性或完整性喪失的漏洞，但可以減輕風險。

- 低 （類別 III)

  包含任何會降低防範機密性、可用性或完整性遺失之措施的漏洞。

Amazon EC2 提供下列方法來建立 STIG 強化執行個體：

- 您可以從專門的公有 Windows AMIs 啟動針對 STIG 強化預先設定的 AWS Windows 執行個體。如需詳細資訊，請參閱《AWS Windows AMI 參考》中的 STIG 強化 AWS Windows Server AMI。
- 可以建立自訂 AMI，以啟動使用 EC2 Image Builder STIG 強化元件建置的預先設定執行個體。如需詳細資訊，請參閱《Image Builder 使用者指南》中的 STIG 強化元件。
- 可以使用 AWSEC2-ConfigureSTIG Systems Manager 命令文件，將 STIG 設定套用至現有的 EC2 執行個體。Systems Manager STIG 命令文件會掃描組態錯誤，並執行修復指令碼來安裝和更新 DoD 憑證，以及移除不必要的憑證以維持 STIG 合規。使用 STIG Systems Manager 文件無需額外費用。

主題

- EC2 執行個體的 STIG 強化設定
- STIG 強化指令碼下載
- 使用 AWS Systems Manager 將 STIG 設定套用至您的執行個體

EC2 執行個體的 STIG 強化設定

STIG 設定根據您的執行個體作業系統和您選擇的合規層級而有所不同。下列內容包含每個合規層級套用的 STIG 的系統特定詳細資訊。

目錄

- EC2 Linux 執行個體的 STIG 設定
- EC2 Windows 執行個體的 STIG 強化設定

EC2 Linux 執行個體的 STIG 設定

本節包含 Amazon EC2 支援的 Linux STIG 強化設定的相關資訊。強化指令碼會根據 Linux 發行版本，將支援的 STIG 設定套用至基礎設施。如果 Linux 發行版本沒有自己的 STIG 強化設定，Amazon EC2 會使用 RHEL 設定，如下所示。

- Red Hat Enterprise Linux (RHEL) 7 STIG 設定

  - RHEL 7
  - CentOS 7
  - Amazon Linux 2 (AL2)
- RHEL 8 STIG 設定
  - RHEL 8
  - CentOS 8
- RHEL 9 STIG 設定
  - RHEL 9
  - CentOS Stream 9

Linux STIG Low (類別 III)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需完整清單，請參閱 STIG 文件庫。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

RHEL 7 STIG 版本 3 第 15 版

- RHEL 7/CentOS 7/AL2

  V-204452, V-204576 和 V-204605

RHEL 8 STIG 第 2 版第 6 版

- RHEL 8/CentOS 8

  V-230241, V-230269, V-230270, V-230281, V-230285, V-230346, V-230381, V-230395, V-230468, V-230469, V-230485, V-230486, V-230491, V-230494, V-230495, V-230496, V-230497, V-230498, V-230499和 V-244527

RHEL 9 STIG 第 2 版第 7 版

- RHEL 9/CentOS Stream 9

V-257782、V-257795、V-257796、V-257824、V-257880、V-257946、V-257947、V-258037、V-258067 和 V-258173

RHEL 10 STIG 第 1 版第 1 版

V-280995, V-281203 和 V-281292

Amazon Linux 2023 STIG 版本 1 第 1 版

V-274141

SLES 12 STIG 第 3 版第 4 版

V-217108、V-217113、V-217140、V-217198、V-217209、V-217211、V-217212、V-217213、V-217214、和 V-255915

SLES 15 STIG 第 2 版第 6 版

V-234811, V-234850, V-234868, V-234873, V-234905, V-234907, V-234908, V-234909, V-234933, V-234934, V-234935, V-234936, V-234955, V-234963, V-234967 和 V-255921

Ubuntu 18.04 STIG 版本 2 第 15 版

V-219163, V-219164, V-219165, V-219172, V-219173, V-219174, V-219175, V-219178, V-219179, V-219180, V-219210, V-219301, V-219327, V-219332 和 V-219333

Ubuntu 20.04 STIG 第 2 版第 4 版

V-238202, V-238203, V-238221, V-238222, V-238223, V-238224, V-238226, V-238234, V-238235, V-238237, V-238308, V-238323, V-238357, V-238362 和 V-238373

Ubuntu 22.04 STIG 第 2 版第 7 版

V-260472, V-260476, V-260479, V-260480, V-260481, V-260520, V-260521, V-260549, V-260550, V-260551, V-260552, V-260581 和 V-260596

Ubuntu 24.04 STIG 第 1 版第 4 版

V-270645, V-270646, V-270664, V-270677, V-270690, V-270695, V-270706, V-270710, V-270734, V-270749, V-270752, V-270818 和 V-270820

Linux STIG Medium (類別 II)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需完整清單，請參閱 STIG 文件庫。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

> **Note**
>
> 除了 Amazon EC2 針對類別 II 漏洞支援的 STIG 強化設定之外，Linux STIG Medium 類別還包括適用於 Linux STIG Low (類別 III) 的所有列出的 STIG 強化設定。

RHEL 7 STIG 版本 3 第 15 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 7/CentOS 7/AL2

  V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204416, V-204417, V-204418, V-204420, V-204422, V-204423, V-204426, V-204427, V-204431, V-204434, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204490, V-204491, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204514, V-204515, V-204516, V-204517, V-204521, V-204524, V-204531, V-204536, V-204537, V-204538, V-204539, V-204540, V-204541, V-204542, V-204543, V-204544, V-204545, V-204546, V-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204579, V-204584, V-204585, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204596, V-204597, V-204598, V-204599, V-204600, V-204601, V-204602, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204616, V-204617, V-204619, V-204622, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, V-255925, V-255927, V-255928 V-256970

RHEL 8 STIG 第 2 版第 6 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 8/CentOS 8

V-230222, V-230228, V-230231, V-230233, V-230236, V-230237, V-230238, V-230239, V-230240, V-230240, V-230243, V-230244, V-230245, V-230246, V-230247, V-230248, V-230249, V-230250, V-230255, V-230256, V-230257, V-230258, V-230259, V-230260, V-230261, V-230262, V-230266, V-230267, V-230268, V-230271, V-230273, V-230275, V-230276, V-230277, V-230278, V-230279, V-230280, V-230282, V-230282, V-230286, V-230287, V-230288, V-230290, V-230291, V-230296, V-230298, V-230310, V-230311, V-230312, V-230313, V-230314, V-230315, V-230316, V-230318, V-230319, V-230320, V-230321, V-230322, V-230324, V-230325, V-230326, V-230327, V-230330, V-230332, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230345, V-230347, V-230348, V-230352, V-230353, V-230354, V-230356, V-230357, V-230358, V-230359, V-230360, V-230361, V-230362, V-230363, V-230365, V-230366, V-230368, V-230369, V-230370, V-230372, V-230373, V-230375, V-230376, V-230377, V-230378, V-230380, V-230382, V-230383, V-230385, V-230386, V-230387, V-230389, V-230390, V-230392, V-230393, V-230394, V-230396, V-230397, V-230398, V-230399, V-230400, V-230401, V-230402, V-230403, V-230404, V-230405, V-230406, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230413, V-230418, V-230419, V-230421, V-230422, V-230423, V-230424, V-230425, V-230426, V-230427, V-230428, V-230429, V-230430, V-230431, V-230432, V-230433, V-230434, V-230435, V-230436, V-230437, V-230438, V-230439, V-230444, V-230446, V-230447, V-230448, V-230449, V-230455, V-230456, V-230462, V-230463, V-230464, V-230465, V-230466, V-230467, V-230470, V-230471, V-230472, V-230473, V-230474, V-230475, V-230478, V-230480, V-230481, V-230482, V-230483, V-230488, V-230489, V-230493, V-230502, V-230503, V-230505, V-230506, V-230507, V-230523, V-230524, V-230525, V-230526, V-230527, V-230532, V-230535, V-230536, V-230537, V-230538, V-230539, V-230540, V-230541, V-230542, V-230543, V-230544, V-230545, V-230546, V-230547, V-230548, V-230549, V-230550, V-230555, V-230556, V-230557, V-230559, V-230560, V-230561, V-237640, V-237642, V-237643, V-244519, V-244523, V-244524, V-244525, V-244526, V-244528, V-244531, V-244533, V-244535, V-244536, V-244538, V-244539, V-244542, V-244543, V-244544, V-244545, V-244547, V-244550, V-244551, V-244552, V-244553, V-244554, V-250315, V-250315, V-250315, V-250316, V-250316, V-250317, V-251707, V-251708, V-251709, V-251710, V-251711, V-251713, V-251714, V-251715, V-251716, V-251717, V-251718, V-256974, V-257258, V-257258, V-274877, V-279929, V-279930 V-279931

RHEL 9 STIG 第 2 版第 7 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 9/CentOS Stream 9

V-257780, V-257781, V-257783, V-257786, V-257788, V-257790, V-257791, V-257792, V-257793, V-257794, V-257797, V-257798, V-257799, V-257800, V-257801, V-257802, V-257803, V-257804, V-257805, V-257806, V-257807, V-257808, V-257809, V-257810, V-257811, V-257812, V-257813, V-257814, V-257815, V-257816, V-257817, V-257818, V-257825, V-257827, V-257828, V-257829, V-257830, V-257831, V-257832, V-257833, V-257834, V-257836, V-257838, V-257839, V-257840, V-257841, V-257842, V-257849, V-257882, V-257883, V-257884, V-257885, V-257886, V-257887, V-257888, V-257889, V-257890, V-257891, V-257892, V-257893, V-257894, V-257895, V-257896, V-257897, V-257898, V-257899, V-257900, V-257901, V-257902, V-257903, V-257904, V-257905, V-257906, V-257907, V-257908, V-257909, V-257910, V-257911, V-257912, V-257913, V-257914, V-257915, V-257916, V-257917, V-257918, V-257919, V-257920, V-257921, V-257922, V-257923, V-257924, V-257925, V-257926, V-257927, V-257928, V-257929, V-257930, V-257933, V-257934, V-257935, V-257936, V-257939, V-257940, V-257942, V-257943, V-257944, V-257948, V-257951, V-257952, V-257953, V-257954, V-257957, V-257958, V-257959, V-257960, V-257961, V-257962, V-257963, V-257964, V-257965, V-257966, V-257967, V-257968, V-257969, V-257970, V-257971, V-257972, V-257973, V-257974, V-257975, V-257976, V-257977, V-257978, V-257979, V-257980, V-257982, V-257983, V-257985, V-257987, V-257988, V-257992, V-257993, V-257994, V-257995, V-257996, V-257997, V-257998, V-257999, V-258000, V-258001, V-258002, V-258003, V-258004, V-258005, V-258006, V-258007, V-258008, V-258009, V-258010, V-258011, V-258028, V-258034, V-258035, V-258038, V-258039, V-258040, V-258041, V-258043, V-258046, V-258049, V-258052, V-258054, V-258055, V-258056, V-258057, V-258060, V-258063, V-258064, V-258065, V-258066, V-258068, V-258070, V-258071, V-258072, V-258073, V-258074, V-258075, V-258077, V-258079, V-258080, V-258081, V-258082, V-258083, V-258084, V-258085, V-258088, V-258089, V-258090, V-258091, V-258092, V-258093, V-258095, V-258097, V-258098, V-258099, V-258100, V-258101, V-258102, V-258103, V-258104, V-258105, V-258107, V-258108, V-258109, V-258110, V-258111, V-258112, V-258113, V-258114, V-258115, V-258116, V-258117, V-258118, V-258119, V-258120, V-258121, V-258122, V-258123, V-258124, V-258125, V-258126, V-258128, V-258129, V-258130, V-258133, V-258137, V-258140, V-258141, V-258142, V-258144, V-258145, V-258146, V-258147, V-258148, V-258150, V-258151, V-258152, V-258153, V-258154, V-258156, V-258157, V-258158, V-258159, V-258160, V-258161, V-258162, V-258163, V-258164, V-258165, V-258166, V-258167, V-258168, V-258169, V-258170, V-258171, V-258172, V-258175, V-258176, V-258177, V-258178, V-258179, V-258180, V-258181, V-258182, V-258183, V-258184, V-258185, V-258186, V-258187, V-258188, V-258189, V-258190, V-258191, V-258192, V-258193, V-258194, V-258195, V-258196, V-258197, V-258198, V-258199, V-258200, V-258201, V-258202, V-258203, V-258204, V-258205, V-258206, V-258207, V-258208, V-258209, V-258210, V-258211, V-258212, V-258213, V-258214, V-258215, V-258216, V-258217, V-258218, V-258219, V-258220, V-258221, V-258222, V-258223,

V-258224, V-258225, V-258226, V-258227, V-258228, V-258229, V-258232, V-258233, V-258234, V-258237, V-258239, V-258240, V-270174, V-270175, V-270176, V-270177, V-272488 V-279936

RHEL 10 STIG 第 1 版第 1 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-280094, V-280942, V-280943, V-280945, V-280946, V-280947, V-280948, V-280950, V-280952, V-280953, V-280954, V-280955, V-280956, V-280958, V-280959, V-280960, V-280961, V-280962, V-280963, V-280964, V-280965, V-280966, V-280967, V-280968, V-280969, V-280970, V-280972, V-280973, V-280974, V-280975, V-280976, V-280977, V-280981, V-280983, V-280984, V-280986, V-280987, V-280988, V-280989, V-280990, V-280991, V-280992, V-280993, V-280994, V-280996, V-280997, V-280999, V-281000, V-281001, V-281002, V-281003, V-281005, V-281006, V-281017, V-281018, V-281019, V-281020, V-281021, V-281022, V-281023, V-281024, V-281025, V-281026, V-281027, V-281028, V-281029, V-281030, V-281031, V-281032, V-281033, V-281034, V-281035, V-281036, V-281037, V-281038, V-281039, V-281040, V-281041, V-281042, V-281043, V-281044, V-281045, V-281047, V-281048, V-281049, V-281050, V-281051, V-281052, V-281053, V-281054, V-281055, V-281056, V-281057, V-281058, V-281059, V-281060, V-281061, V-281062, V-281063, V-281064, V-281065, V-281066, V-281067, V-281068, V-281069, V-281070, V-281071, V-281072, V-281073, V-281074, V-281075, V-281076, V-281077, V-281078, V-281079, V-281080, V-281081, V-281082, V-281083, V-281084, V-281085, V-281086, V-281087, V-281094, V-281095, V-281096, V-281097, V-281098, V-281099, V-281100, V-281101, V-281102, V-281103, V-281104, V-281105, V-281107, V-281108, V-281109, V-281110, V-281111, V-281113, V-281114, V-281115, V-281116, V-281117, V-281118, V-281119, V-281120, V-281121, V-281122, V-281123, V-281124, V-281125, V-281126, V-281127, V-281128, V-281129, V-281130, V-281131, V-281132, V-281133, V-281134, V-281135, V-281136, V-281137, V-281138, V-281139, V-281140, V-281141, V-281142, V-281143, V-281144, V-281145, V-281146, V-281147, V-281148, V-281149, V-281150, V-281151, V-281152, V-281153, V-281154, V-281155, V-281156, V-281157, V-281158, V-281159, V-281160, V-281161, V-281162, V-281163, V-281164, V-281165, V-281168, V-281169, V-281171, V-281174, V-281175, V-281177, V-281178, V-281179, V-281180, V-281181, V-281182, V-281183, V-281184, V-281185, V-281187, V-281188, V-281189, V-281190, V-281191, V-281192, V-281193, V-281195, V-281196, V-281197, V-281198, V-281200, V-281201, V-281204, V-281205, V-281206, V-281209, V-281210, V-281212, V-281214, V-281215, V-281217, V-281218, V-281219, V-281220, V-281224, V-281225, V-281226, V-281249, V-281251, V-281252, V-281253, V-281254, V-281255, V-281256, V-281257, V-281258, V-281259, V-281260, V-281261, V-281262, V-281263, V-281264, V-281265, V-281266, V-281269, V-281270, V-281271, V-281272, V-281273, V-281274, V-281276, V-281277, V-281278, V-281279, V-281280, V-281281, V-281282, V-281284, V-281285, V-281286, V-281287, V-281288,

V-281289, V-281290, V-281291, V-281293, V-281295, V-281296, V-281297, V-281300, V-281301, V-281302, V-281303, V-281304, V-281305, V-281306, V-281308, V-281309, V-281310, V-281311, V-281312, V-281313, V-281314, V-281315, V-281316, V-281317, V-281318, V-281319, V-281320, V-281322, V-281323, V-281324, V-281325, V-281327, V-281328, V-281331, V-281333, V-281334, V-281335, V-281337, V-281338, V-281340, V-281341, V-281342, V-281343, V-281344, V-281345, V-281346, V-281347, V-281348, V-281349, V-281350, V-281351, V-281352, V-281353, V-281354, V-281355, V-281356, V-281357, V-281358, V-281359, V-281360, V-281361, V-281363, V-281364 V-281365

Amazon Linux 2023 STIG 第 1 版第 2 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-273995, V-274000, V-274001, V-274002, V-274003, V-274004, V-274005, V-274006, V-274008, V-274009, V-274010, V-274011, V-274012, V-274013, V-274014, V-274017, V-274018, V-274019, V-274020, V-274021, V-274022, V-274023, V-274024, V-274026, V-274027, V-274028, V-274030, V-274031, V-274032, V-274033, V-274034, V-274035, V-274036, V-274037, V-274040, V-274041, V-274042, V-274044, V-274045, V-274047, V-274048, V-274049, V-274050, V-274051, V-274053, V-274054, V-274059, V-274061, V-274062, V-274069, V-274070, V-274071, V-274072, V-274073, V-274074, V-274075, V-274076, V-274077, V-274078, V-274079, V-274081, V-274082, V-274083, V-274084, V-274085, V-274086, V-274087, V-274088, V-274089, V-274090, V-274091, V-274092, V-274093, V-274094, V-274095, V-274096, V-274097, V-274098, V-274099, V-274100, V-274101, V-274102, V-274103, V-274104, V-274105, V-274106, V-274107, V-274108, V-274109, V-274110, V-274111, V-274112, V-274113, V-274114, V-274115, V-274116, V-274117, V-274119, V-274120, V-274121, V-274122, V-274123, V-274124, V-274125, V-274126, V-274127, V-274128, V-274129, V-274130, V-274131, V-274132, V-274133, V-274134, V-274135, V-274136, V-274137, V-274138, V-274139, V-274140, V-274142, V-274143, V-274144, V-274145, V-274147, V-274149, V-274151, V-274152, V-274154, V-274155, V-274156, V-274157, V-274160, V-274161, V-274162, V-274163, V-274164, V-274165, V-274166, V-274167, V-274168, V-274169, V-274170, V-274173, V-274177, V-274181, V-274182, V-274185 V-274187

SLES 12 STIG 第 3 版第 4 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-217102, V-217105, V-217105, V-217106, V-217106, V-217110, V-217116, V-217117, V-217118, V-217119, V-217120, V-217121, V-217122, V-217124, V-217125, V-217126, V-217127, V-217128, V-217130, V-217134, V-217138, V-217143, V-217147, V-217152, V-217153, V-217154, V-217155, V-217156, V-217158, V-217161, V-217163, V-217166, V-217167, V-217168, V-217169, V-217170,

V-217171, V-217182, V-217183, V-217188, V-217190, V-217191, V-217194, V-217195, V-217196, V-217197, V-217200, V-217201, V-217202, V-217203, V-217204, V-217205, V-217206, V-217207, V-217208, V-217210, V-217217, V-217218, V-217223, V-217227, V-217230, V-217240, V-217241, V-217242, V-217243, V-217244, V-217245, V-217246, V-217247, V-217248, V-217249, V-217250, V-217251, V-217252, V-217253, V-217254, V-217255, V-217257, V-217258, V-217260, V-217265, V-217266, V-217267, V-217269, V-217272, V-217273, V-217274, V-217275, V-217276, V-217277, V-217278, V-217279, V-217280, V-217283, V-217284, V-217286, V-217287, V-217288, V-217289, V-217290, V-217291, V-217292, V-217293, V-217294, V-217295, V-217296, V-217299, V-217300, V-217301, V-217302, V-233308, V-237605, V-237606, V-237607, V-237608, V-237609, V-237610, V-237611, V-237612, V-237613, V-237614, V-237615, V-237616, V-237617, V-237618, V-237619, V-237620, V-237621, V-237622, V-237623, V-251720, V-251722, V-255914 V-256981

SLES 15 STIG 第 2 版第 6 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-234802, V-234807, V-234808, V-234809, V-234813, V-234815, V-234817, V-234821, V-234822, V-234823, V-234825, V-234827, V-234828, V-234829, V-234830, V-234832, V-234833, V-234834, V-234835, V-234836, V-234837, V-234838, V-234839, V-234840, V-234841, V-234842, V-234843, V-234844, V-234845, V-234848, V-234854, V-234855, V-234856, V-234857, V-234858, V-234861, V-234862, V-234863, V-234869, V-234870, V-234875, V-234878, V-234880, V-234881, V-234882, V-234883, V-234884, V-234885, V-234886, V-234887, V-234888, V-234889, V-234891, V-234895, V-234896, V-234897, V-234899, V-234900, V-234901, V-234902, V-234903, V-234904, V-234906, V-234910, V-234911, V-234912, V-234913, V-234914, V-234918, V-234924, V-234928, V-234932, V-234937, V-234938, V-234939, V-234940, V-234941, V-234942, V-234943, V-234944, V-234945, V-234946, V-234947, V-234948, V-234949, V-234950, V-234951, V-234952, V-234954, V-234956, V-234957, V-234958, V-234959, V-234961, V-234962, V-234964, V-234966, V-234969, V-234973, V-234975, V-234976, V-234977, V-234978, V-234979, V-234981, V-234982, V-234983, V-234991, V-235002, V-235003, V-235007, V-235008, V-235009, V-235010, V-235013, V-235014, V-235015, V-235016, V-235017, V-235018, V-235019, V-235020, V-235021, V-235022, V-235023, V-235024, V-235025, V-235026, V-235028, V-235029, V-235030, V-251724, V-255920, V-256983 V-274879

Ubuntu 18.04 STIG 版本 2 第 15 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-219149、V-219155、V-219156、V-219160、V-219166、V-219168、V-219176、V-219181、V-219184、和 V-255906

Ubuntu 20.04 STIG 第 2 版第 4 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-238200、V-238205、V-238207、V-238209、V-238210、V-238211、V-238212、V-238213、V-238220、和 V-274853

Ubuntu 22.04 STIG 第 2 版第 7 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-260471, V-260473, V-260474, V-260475, V-260475, V-260477, V-260478, V-260485, V-260486, V-260487, V-260488, V-260489, V-260490, V-260491, V-260492, V-260493, V-260494, V-260495, V-260496, V-260497, V-260498, V-260499, V-260500, V-260505, V-260506, V-260507, V-260508, V-260509, V-260510, V-260511, V-260512, V-260513, V-260514, V-260522, V-260527, V-260528, V-260530, V-260533, V-260534, V-260535, V-260537, V-260538, V-260540, V-260542, V-260543, V-260545, V-260546, V-260547, V-260553, V-260554, V-260555, V-260556, V-260557, V-260560, V-260561, V-260562, V-260563, V-260564, V-260565, V-260566, V-260567, V-260569, V-260572, V-260573, V-260574, V-260575, V-260576, V-260582, V-260584, V-260585, V-260586, V-260588, V-260589, V-260590, V-260591, V-260594, V-260597, V-260598, V-260599, V-260600, V-260601, V-260602, V-260603, V-260604, V-260605, V-260606, V-260607, V-260608, V-260609, V-260610, V-260611, V-260612, V-260613, V-260614, V-260615, V-260616, V-260617, V-260618, V-260619, V-260620, V-260621, V-260622, V-260623, V-260624, V-260625, V-260626, V-260627, V-260628, V-260629, V-260630, V-260631, V-260632, V-260633, V-260634, V-260635, V-260636, V-260637, V-260638, V-260639, V-260640, V-260641, V-260642, V-260643, V-260644, V-260645, V-260646, V-260647, V-260648, V-260649, V-274862, V-274864 V-274866

Ubuntu 24.04 STIG 第 1 版第 4 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-270649、V-270651、V-270652、V-270653、V-270654、V-270656、V-270657、V-270659、V-270660、和 V-274873

Linux STIG High (類別 I)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需完整清單，請參閱 STIG 文件庫。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

> **Note**
>
> 除了 Amazon EC2 針對類別 I 漏洞支援的 STIG 強化設定之外，Linux STIG High 類別還包括適用於 Linux STIG Medium 和 Low 類別的所有列出的 STIG 強化設定。

RHEL 7 STIG 版本 3 第 15 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 7/CentOS 7/AL2

  V-204424, V-204425, V-204442, V-204443, V-204447, V-204448, V-204455, V-204462, V-204497, V-204497, V-204502, V-204594, V-204620 和 V-204621

RHEL 8 STIG 第 2 版第 6 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 8/CentOS 8

  V-230223, V-230264, V-230283, V-230284, V-230487, V-230492, V-230533, V-230558, V-244540, V-279933, V-230265, V-230226, V-230530, V-268322, V-230529 和 V-230531

RHEL 9 STIG 第 2 版第 7 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

- RHEL 9/CentOS Stream 9

  V-257820, V-257821, V-257826, V-257835, V-257955, V-257956, V-258059, V-258230, V-258238, V-257984, V-257986, V-258078, V-258094, V-258235, V-257784 和 V-257785

RHEL 10 第 1 版第 1 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-280932, V-280933, V-280934, V-280944, V-280949, V-280951, V-280978, V-281007, V-281014, V-281015, V-281216, V-281222, V-281223, V-281267, V-281268, V-281275, V-281298, V-281299和 V-281307

Amazon Linux 2023 STIG 第 1 版第 2 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-273996、V-273997、V-273999、V-274007、V-274038、V-274039、V-274046、V-274052、V-274057 和 V-274153

SLES 12 STIG 第 3 版第 4 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-217101, V-217139, V-217141, V-217142, V-217159, V-217160, V-217164, V-217264, V-217268, V-222386 和 V-251721

SLES 15 STIG 第 2 版第 6 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-234800, V-234804, V-234818, V-234852, V-234859, V-234860, V-234898, V-234984, V-234985, V-234988, V-234989, V-234990, V-235031, V-235032 和 V-251725

Ubuntu 18.04 STIG 版本 2 第 15 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-219157, V-219158, V-219177, V-219212, V-219308, V-219314, V-219316 和 V-251507

Ubuntu 20.04 STIG 第 2 版第 4 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-238201, V-238218, V-238219, V-238326, V-238327, V-238380 和 V-251504

Ubuntu 22.04 STIG 第 2 版第 7 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-260469, V-260482, V-260483, V-260523, V-260524, V-260526, V-260529, V-260539, V-260570, V-260571, V-260579 和 V-279937

Ubuntu 24.04 STIG 第 1 版第 4 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-270647, V-270648, V-270665, V-270666, V-270708, V-270711, V-270712, V-270713, V-270714, V-270717, V-270736, V-270738 和 V-279938

EC2 Windows 執行個體的 STIG 強化設定

Amazon EC2 支援下列 Windows 作業系統版本的 STIG 基準：

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

本節列出 Amazon EC2 為您的 Windows 基礎設施支援的目前 STIG 設定，後接版本歷史記錄日誌。

可以套用低、中或高 STIG 設定。

Windows STIG Low (類別 III)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需 Windows STIG 的完整清單，請參閱 STIGs Document Library (STIG 文件庫)。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

- Windows Server 2025 STIG 第 1 版第 1 版

  V-278082, V-278083, V-278084, V-278085, V-278098, V-278104, V-278110 和 V-278231
- Windows Server 2022 STIG 第 2 版第 7 版

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363 和 V-254481
- Windows Server 2019 STIG 第 3 版第 7 版

  V-205691、V-205819、V-205858、V-205859、V-205860、V-205870、V-205871 以及 V-205923
- Windows Server 2016 STIG 第 2 版第 10 版

  V-224916、V-224917、V-224918、V-224919、V-224931、V-224942 以及 V-225060
- Windows Server 2012 R2 MS STIG 第 3 版第 5 版

  V-225250、V-225318、V-225319、V-225324、V-225327、V-225328、V-225330、V-225331、V-225332 和 V-225537

- Microsoft .NET Framework 4.0 STIG 第 2 版第 7 版

  對於類別 III 漏洞，沒有 STIG 設定會套用至 Microsoft .NET Framework。
- Windows Firewall STIG 第 2 版第 2 版

  V-241994、V-241995、V-241996、V-241999、V-242000、V-242001、V-242006、V-242007 和 V-242008
- Internet Explorer 11 STIG 第 2 版第 6 版

  V-223056 和 V-223078
- Microsoft Edge STIG 第 2 版第 4 版 （ 僅限 Windows Server 2022 和 2025)

  V-235727、V-235731、V-235751、V-235752 和 V-235765

Windows STIG Medium (類別 II)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需 Windows STIG 的完整清單，請參閱 STIGs Document Library (STIG 文件庫)。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

> **Note**
>
> 除了 Amazon EC2 針對類別 II 漏洞支援的 STIG 強化設定之外，Windows STIG Medium 類別還包括適用於 Windows STIG Low (類別 III) 的所有列出的 STIG 強化設定。

- Windows Server 2025 STIG 第 1 版第 1 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-278015, V-278016, V-278019, V-278020, V-278021, V-278022, V-278023, V-278024, V-278025, V-278026, V-278033, V-278034, V-278035, V-278036, V-278037, V-278038, V-278039, V-278047, V-278048, V-278049, V-278050, V-278051, V-278052, V-278053, V-278054, V-278055, V-278056, V-278057, V-278058, V-278059, V-278060, V-278061, V-278062, V-278063, V-278064, V-278065, V-278066, V-278067, V-278068, V-278069, V-278070, V-278071, V-278072, V-278073, V-278074, V-278075, V-278076, V-278077, V-278078, V-278079, V-278080, V-278086, V-278088, V-278089,

V-278091, V-278092, V-278093, V-278094, V-278095, V-278096, V-278097, V-278102, V-278103, V-278105, V-278106, V-278107, V-278108, V-278109, V-278111, V-278112, V-278113, V-278114, V-278115, V-278116, V-278117, V-278118, V-278119, V-278120, V-278122, V-278123, V-278124, V-278126, V-278127, V-278129, V-278130, V-278131, V-278165, V-278168, V-278169, V-278170, V-278171, V-278174, V-278180, V-278181, V-278182, V-278183, V-278184, V-278185, V-278187, V-278188, V-278189, V-278192, V-278193, V-278194, V-278195, V-278198, V-278199, V-278200, V-278201, V-278202, V-278203, V-278204, V-278205, V-278206, V-278209, V-278210, V-278211, V-278212, V-278213, V-278214, V-278218, V-278220, V-278221, V-278222, V-278223, V-278226, V-278227, V-278228, V-278229, V-278230, V-278232, V-278233, V-278234, V-278235, V-278236, V-278237, V-278238, V-278239, V-278240, V-278241, V-278243, V-278244, V-278245, V-278247, V-278248, V-278249, V-278251, V-278252, V-278253, V-278254, V-278255, V-278256, V-278257, V-278258, V-278259, V-278260, V-278261, V-278262, V-279916, V-279917, V-279918, V-279919, V-279920, V-279921, V-279922 V-279923

- Windows Server 2022 STIG 第 2 版第 7 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-254247, V-254269, V-254270, V-254271, V-254272, V-254273, V-254274, V-254275, V-254276, V-254277, V-254278, V-254285, V-254286, V-254287, V-254288, V-254289, V-254290, V-254291, V-254292, V-254296, V-254297, V-254298, V-254299, V-254300, V-254301, V-254302, V-254303, V-254304, V-254305, V-254307, V-254309, V-254311, V-254312, V-254313, V-254314, V-254315, V-254316, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, V-254326, V-254327, V-254328, V-254329, V-254330, V-254331, V-254332, V-254333, V-254334, V-254339, V-254341, V-254342, V-254344, V-254345, V-254346, V-254347, V-254348, V-254349, V-254350, V-254355, V-254356, V-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254365, V-254366, V-254367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254375, V-254376, V-254377, V-254379, V-254380, V-254382, V-254383, V-254384, V-254431, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254440, V-254442, V-254443, V-254444, V-254445, V-254447, V-254448, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254476, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483, V-254484, V-254485, V-254486, V-254487, V-254488, V-254489, V-254491, V-254493, V-254494, V-254495, V-254497, V-254498, V-254499, V-254501, V-254502, V-254503, V-254504, V-254505, V-254506, V-254507, V-254508, V-254509, V-254510, V-254511, V-254512, V-278942, V-278943, V-278944, V-278945, V-278946, V-278947, V-278948 V-278949

- Windows Server 2019 STIG 第 3 版第 7 版

包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

V-205625, V-205626, V-205627, V-205629, V-205630, V-205633, V-205634, V-205635, V-205636, V-205637, V-205638, V-205639, V-205640, V-205641, V-205642, V-205643, V-205644, V-205648, V-205649, V-205650, V-205651, V-205652, V-205655, V-205656, V-205659, V-205660, V-205662, V-205671, V-205672, V-205673, V-205675, V-205676, V-205678, V-205679, V-205680, V-205681, V-205682, V-205683, V-205684, V-205685, V-205686, V-205687, V-205688, V-205689, V-205690, V-205692, V-205693, V-205694, V-205697, V-205698, V-205708, V-205709, V-205712, V-205714, V-205716, V-205717, V-205718, V-205719, V-205720, V-205722, V-205730, V-205731, V-205733, V-205747, V-205748, V-205749, V-205751, V-205752, V-205754, V-205755, V-205756, V-205758, V-205759, V-205760, V-205761, V-205762, V-205763, V-205764, V-205765, V-205766, V-205767, V-205768, V-205769, V-205770, V-205771, V-205772, V-205773, V-205774, V-205775, V-205776, V-205777, V-205778, V-205779, V-205780, V-205781, V-205782, V-205783, V-205784, V-205795, V-205796, V-205797, V-205798, V-205801, V-205808, V-205809, V-205810, V-205811, V-205812, V-205813, V-205814, V-205815, V-205816, V-205817, V-205821, V-205822, V-205823, V-205824, V-205825, V-205826, V-205827, V-205828, V-205830, V-205832, V-205833, V-205835, V-205836, V-205837, V-205838, V-205842, V-205861, V-205863, V-205865, V-205866, V-205867, V-205868, V-205869, V-205872, V-205873, V-205874, V-205909, V-205910, V-205911, V-205912, V-205915, V-205916, V-205917, V-205918, V-205920, V-205921, V-205922, V-205925, V-257503, V-278934, V-278935, V-278936, V-278937, V-278938, V-278939, V-278940 V-278941

- Windows Server 2016 STIG 第 2 版第 10 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-224850, V-224851, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859, V-224866, V-224867, V-224868, V-224869, V-224870, V-224871, V-224872, V-224873, V-224877, V-224878, V-224879, V-224880, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224888, V-224890, V-224892, V-224893, V-224894, V-224895, V-224896, V-224897, V-224900, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, V-224910, V-224911, V-224912, V-224913, V-224914, V-224915, V-224920, V-224921, V-224922, V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224935, V-224936, V-224937, V-224938, V-224939, V-224940, V-224941, V-224943, V-224944, V-224945, V-224946, V-224947, V-224948, V-224949, V-224951, V-224952, V-224953, V-224955, V-224956, V-224957, V-224959, V-224960, V-224962, V-224963, V-224965, V-224966, V-224967, V-224968, V-224969, V-224987, V-224988, V-224989, V-224995, V-224996, V-224997, V-224998, V-224999, V-225000, V-225001, V-225002, V-225003, V-225004, V-225005, V-225008, V-225009, V-225010, V-225011, V-225013, V-225014, V-225015, V-225016, V-225017, V-225018, V-225019, V-225020,

V-225021, V-225022, V-225023, V-225024, V-225026, V-225027, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225058, V-225059, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225070, V-225072, V-225073, V-225074, V-225076, V-225077, V-225078, V-225080, V-225081, V-225082, V-225083, V-225084, V-225085, V-225086, V-225087, V-225088, V-225089, V-225092, V-225093 V-257502

- Windows Server 2012 R2 MS STIG 第 3 版第 5 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-225239、V-225259、V-225260、V-225261、V-225263、V-225264、V-225265、V-225266、V-225267 和 V-225574

- Microsoft .NET Framework 4.0 STIG 第 2 版第 7 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-225223, V-225230, V-225235 和 V-225238

- Windows Firewall STIG 第 2 版第 2 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-241989, V-241990, V-241991, V-241993, V-241998, V-242003, V-242004 和 V-242005

- Internet Explorer 11 STIG 第 2 版第 6 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-223015, V-223017, V-223018, V-223019, V-223020, V-223021, V-223022, V-223023, V-223024, V-223025, V-223026, V-223027, V-223028, V-223029, V-223030, V-223031, V-223032, V-223033, V-223034, V-223035, V-223036, V-223037, V-223038, V-223039, V-223040, V-223041, V-223042, V-223043, V-223044, V-223045, V-223046, V-223048, V-223049, V-223050, V-223051, V-223052, V-223053, V-223054, V-223055, V-223057, V-223058, V-223059, V-223060, V-223061, V-223062, V-223063, V-223064, V-223065, V-223066, V-223067, V-223068, V-223069, V-223070, V-223071, V-223072, V-223073, V-223074, V-223075, V-223076, V-223077, V-223079, V-223080, V-223081, V-223082, V-223083, V-223084, V-223085, V-223086, V-223087, V-223088, V-223089, V-223090, V-223091, V-223092, V-223093, V-223094, V-223095, V-223096, V-223097, V-223098, V-223099, V-223100, V-223101, V-223102, V-223103, V-223104, V-223105, V-223106, V-223107, V-223108, V-223109, V-223110, V-223111, V-223112, V-223113, V-223114, V-223115, V-223116, V-223117,

V-223118, V-223119, V-223120, V-223121, V-223123, V-223125, V-223126, V-223127, V-223128, V-223129, V-223130, V-223131, V-223132, V-223133, V-223134, V-223135, V-223136, V-223137, V-223138, V-223139, V-223140, V-223141, V-223142, V-223143, V-223144, V-223145, V-223146, V-223147, V-223148, V-223149, V-250540 V-250541

- Microsoft Edge STIG 第 2 版第 4 版 （ 僅限 Windows Server 2022 和 2025)

  V-235720、V-235721、V-235723、V-235724、V-235725、V-235726、V-235728、V-235729、V-235730 和 V-246736

- Microsoft Defender STIG 第 2 版第 7 版

  包括 Amazon EC2 針對類別 III (低) 漏洞支援的所有 STIG 強化設定，以及：

  V-213427, V-213429, V-213430, V-213431, V-213432, V-213433, V-213434, V-213435, V-213436, V-213437, V-213438, V-213439, V-213440, V-213441, V-213442, V-213443, V-213444, V-213445, V-213446, V-213447, V-213448, V-213449, V-213450, V-213451, V-213454, V-213455, V-213456, V-213457, V-213458, V-213459, V-213460, V-213461, V-213462, V-213463, V-213464, V-213465, V-213466, V-278647, V-278648, V-278649, V-278650, V-278651, V-278652, V-278653, V-278654, V-278655, V-278656, V-278658, V-278659, V-278660, V-278661, V-278662, V-278668, V-278669, V-278672, V-278674, V-278675, V-278676, V-278677, V-278678, V-278679, V-278680 V-278863

Windows STIG High (類別 I)

下列清單包含 Amazon EC2 對基礎設施支援的 STIG 設定。如果支援的設定不適用於您的基礎設施，Amazon EC2 會跳過該設定並繼續進行。例如，某些 STIG 強化設定可能不適用於獨立伺服器。組織特定的政策也可能會影響適用的設定，例如要求管理員檢閱文件設定。

如需 Windows STIG 的完整清單，請參閱 STIGs Document Library (STIG 文件庫)。如需有關如何檢視完整清單的詳細資訊，請參閱 STIG 檢視工具。

> **Note**
>
> 除了 Amazon EC2 針對類別 I 漏洞支援的 STIG 強化設定之外，Windows STIG High 類別還包括適用於 Windows STIG Medium 和 Low 類別的所有列出的 STIG 強化設定。

- Windows Server 2025 STIG 第 1 版第 1 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-278040, V-278099, V-278100, V-278101, V-278121, V-278125, V-278128, V-278196, V-278215, V-278216, V-278217, V-278219, V-278225, V-278242, V-278246 和 V-278250

- Windows Server 2022 STIG 第 2 版第 7 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-254250, V-254293, V-254352, V-254353, V-254354, V-254374, V-254378, V-254381, V-254446, V-254466, V-254467, V-254469, V-254474, V-254475, V-254492, V-254496 和 V-254500

- Windows Server 2019 STIG 第 3 版第 7 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-205653, V-205654, V-205663, V-205711, V-205713, V-205724, V-205725, V-205750, V-205753, V-205757, V-205802, V-205804, V-205805, V-205806, V-205849, V-205908, V-205914 和 V-205919

- Windows Server 2016 STIG 第 2 版第 10 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-224831, V-224874, V-224932, V-224933, V-224934, V-224954, V-224958, V-224961, V-225025, V-225045, V-225046, V-225048, V-225053, V-225054, V-225071, V-225079 和 V-225091

- Windows Server 2012 R2 MS STIG 第 3 版第 5 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-225274、V-225354、V-225364、V-225365、V-225366、V-225390、V-225396、V-225399、V-225444 和 V-225556

- Microsoft .NET Framework 4.0 STIG 第 2 版第 7 版

  包括 Amazon EC2 針對 Microsoft .NET Framework 的類別 II 和 III (中和低) 漏洞支援的所有 STIG 強化設定。對於類別 I 漏洞，沒有套用其他 STIG 設定。

- Windows Firewall STIG 第 2 版第 2 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-241992, V-241997 和 V-242002

- Internet Explorer 11 STIG 第 2 版第 6 版

包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

V-252910

- Microsoft Edge STIG 第 2 版第 4 版 （ 僅限 Windows Server 2022 和 2025)

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-235758 和 V-235759

- Microsoft Defender STIG 第 2 版第 7 版

  包括 Amazon EC2 針對類別 II 和 III (Medium 和 Low) 漏洞支援的所有 STIG 強化設定，以及：

  V-213426, V-213428, V-213452 和 V-213453

STIG 強化指令碼下載

Amazon 會將 STIG 強化指令碼分組到每個版本的作業系統相關套件中。套件是封存檔案，適用於在其中進行下載和執行的目標作業系統。Linux 指令碼套件會儲存為 TAR 檔案 (.tgz 副檔名)。Windows 指令碼套件會儲存為 ZIP 檔案 (.zip 副檔名)。

Amazon 會將指令碼套件存放在每個 AWS 區域的 EC2 Windows S3 `STIG` 儲存貯體中。Linux 套件沒有單獨的儲存貯體。使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。

主題

- STIG 下載套件詳細資訊
- Linux STIG 版本歷史記錄
- Windows STIG 版本歷史記錄

STIG 下載套件詳細資訊

> **Important**
>
> 除了少數例外，Systems Manager 文件下載的 STIG 強化指令碼不會安裝第三方套件。如果執行個體上已安裝第三方套件，或 `InstallPackage` 參數設定為 `yes`，則會套用 Amazon EC2 為該套件支援的相關 STIG。

當您從 執行下列命令時 AWS CLI，Amazon S3 會從儲存貯體下載最新的 STIG 強化指令碼套件檔案。

```
aws s3 cp s3://aws-windows-downloads-region/STIG/operating system/Latest/bundle-
name destination-directory
```

範例：下載至暫時目錄

此範例顯示已下載至 /tmp 目錄的 Linux 套件

```
aws s3 cp s3://aws-windows-downloads-us-east-1/STIG/Linux/
Latest/LinuxAWSConfigureSTIG.tgz /tmp
```

下載檔案儲存路徑和套件檔案名稱的模式和範例如下：

下載檔案儲存路徑

s3://aws-windows-downloads-*<region>*/STIG/*<operating system>*/Latest/*<bundle file name>*

下載路徑變數

region

AWS 區域（每個區域都有自己的下載儲存貯體。）

operating system

在其中套用 STIG 之執行個體的作業系統平台 – Linux 或 Windows。

bundle file name

格式為 *<os bundle name>*.*<file extension>*。

os bundle name

作業系統套件的標準名稱字首為 LinuxAWSConfigureSTIG 或 AWSConfigureSTIG。為了維持向後相容性，Windows 的下載不包含平台字首。

file extension

壓縮檔案格式 tgz (Linux) 或 zip (Windows)。

套件檔案名稱範例

- LinuxAWSConfigureSTIG.tgz
- AWSConfigureSTIG.zip

Linux STIG 版本歷史記錄

本節記錄每季更新的 Linux 指令碼套件版本歷史記錄。若要查看季度的變更和已發佈的版本，請選擇標題以展開資訊。如果該季度沒有變更，會在標題中有所反映。

2026 年Q1變更 - 04/07/2026：

新增對 RHEL 10 作業系統的支援，並更新下列 STIG 版本，並針對所有合規層級 (low/medium/high) 的 2026 年第一季版本套用 STIGS：

STIG-Build-Linux 1.0.x 版

- RHEL 8 STIG 第 2 版第 6 版
- RHEL 9 STIG 第 2 版第 7 版
- RHEL 10 STIG 第 1 版第 1 版
- Amazon Linux 2023 STIG 第 1 版第 2 版
- SLES 12 STIG 第 3 版第 4 版
- SLES 15 STIG 第 2 版第 6 版
- Ubuntu 20.04 STIG 第 2 版第 4 版
- Ubuntu 22.04 STIG 第 2 版第 7 版
- Ubuntu 24.04 STIG 第 1 版第 4 版

2025 年第 3 季度變更 - 2025/09/04：

新增對 SUSE Linux Enterprise Server (SLES) 作業系統和 Amazon Linux 2023 的支援。已更新下列 STIG 版本，並將 STIGS 套用於 2025 年第三季發佈的所有合規等級 (低/中/高)：

STIG-Build-Linux 1.0.x 版

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 4 版
- RHEL 9 STIG 版本 2 第 5 版
- Amazon Linux 2023 STIG 版本 1 第 1 版
- SLES 12 STIG 版本 3 第 3 版
- SLES 15 STIG 版本 2 第 5 版
- Ubuntu 18.04 STIG 版本 2 第 15 版

- Ubuntu 20.04 STIG 版本 2 第 3 版
- Ubuntu 22.04 STIG 版本 2 第 5 版
- Ubuntu 24.04 STIG 版本 1 第 2 版

2025 年第 2 季度變更 - 2025/06/26：

已更新下列 STIG 版本，並針對 2025 年第二季版本套用 STIGS：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 3 版
- RHEL 9 STIG 版本 2 第 4 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 4 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 3 版
- RHEL 9 STIG 版本 2 第 4 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 4 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 3 版
- RHEL 9 STIG 版本 2 第 4 版
- Ubuntu 18.04 STIG 版本 2 第 15 版

- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 4 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

2025 年第 1 季度變更 - 2025/04/11：

已更新下列 STIG 版本，為 2025 年第一季版本套用 STIGS，並新增對 Ubuntu 24.04 的支援：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 2 版
- RHEL 9 STIG 版本 2 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 3 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 2 版
- RHEL 9 STIG 版本 2 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 3 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 2 版
- RHEL 9 STIG 版本 2 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 15 版

- Ubuntu 20.04 STIG 版本 2 第 2 版
- Ubuntu 22.04 STIG 版本 2 第 3 版
- Ubuntu 24.04 STIG 版本 1 第 1 版

2024 年第 4 季度變更 - 2024/12/10：

已更新下列 STIG 版本，為 2024 年第四季版本套用 STIGS，並新增了 Linux 元件兩個新輸入參數的相關資訊：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 1 版
- RHEL 9 STIG 版本 2 第 2 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 1 版
- Ubuntu 22.04 STIG 版本 2 第 2 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 1 版
- RHEL 9 STIG 版本 2 第 2 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 1 版
- Ubuntu 22.04 STIG 版本 2 第 2 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 15 版
- RHEL 8 STIG 版本 2 第 1 版
- RHEL 9 STIG 版本 2 第 2 版
- Ubuntu 18.04 STIG 版本 2 第 15 版
- Ubuntu 20.04 STIG 版本 2 第 1 版

- Ubuntu 22.04 STIG 版本 2 第 2 版

2024 年第 3 季度變更 - 2024/10/04 (無變更)：

2024 年第三季發行版本的 Linux 元件 STIGS 沒有變更。

2024 年第 2 季度變更 - 2024/05/10：

已更新 STIG 版本，並針對 2024 年第二季版本套用 STIGS。也新增了對 RHEL 9、CentOS Stream 9 和 Ubuntu 22.04 的支援，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 14 版
- RHEL 9 STIG 版本 1 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 14 版
- Ubuntu 20.04 STIG 版本 1 第 12 版
- Ubuntu 22.04 STIG 版本 1 第 1 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 14 版
- RHEL 9 STIG 版本 1 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 14 版
- Ubuntu 20.04 STIG 版本 1 第 12 版
- Ubuntu 22.04 STIG 版本 1 第 1 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 14 版
- RHEL 9 STIG 版本 1 第 3 版
- Ubuntu 18.04 STIG 版本 2 第 14 版

- Ubuntu 20.04 STIG 版本 1 第 12 版
- Ubuntu 22.04 STIG 版本 1 第 1 版

2024 年第 1 季度變更 - 2024/02/06：

已更新 STIG 版本並針對 2024 年第一季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 13 版
- Ubuntu 18.04 STIG 版本 2 第 13 版
- Ubuntu 20.04 STIG 版本 1 第 11 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 13 版
- Ubuntu 18.04 STIG 版本 2 第 13 版
- Ubuntu 20.04 STIG 版本 1 第 11 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 14 版
- RHEL 8 STIG 版本 1 第 13 版
- Ubuntu 18.04 STIG 版本 2 第 13 版
- Ubuntu 20.04 STIG 版本 1 第 11 版

2023 年第 4 季度變更 - 2023/12/07：

已更新 STIG 版本並針對 2023 年第四季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 13 版
- RHEL 8 STIG 版本 1 第 12 版

- Ubuntu 18.04 STIG 版本 2 第 12 版
- Ubuntu 20.04 STIG 版本 1 第 10 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 13 版
- RHEL 8 STIG 版本 1 第 12 版
- Ubuntu 18.04 STIG 版本 2 第 12 版
- Ubuntu 20.04 STIG 版本 1 第 10 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 13 版
- RHEL 8 STIG 版本 1 第 12 版
- Ubuntu 18.04 STIG 版本 2 第 12 版
- Ubuntu 20.04 STIG 版本 1 第 10 版

2023 年第 3 季度變更 - 2023/10/04：

已更新 STIG 版本並針對 2023 年第三季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 12 版
- RHEL 8 STIG 版本 1 第 11 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 9 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 12 版
- RHEL 8 STIG 版本 1 第 11 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 9 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 12 版
- RHEL 8 STIG 版本 1 第 11 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 9 版

2023 年第 2 季度變更 - 2023/05/03：

已更新 STIG 版本並針對 2023 年第二季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 11 版
- RHEL 8 STIG 版本 1 第 10 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 8 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 11 版
- RHEL 8 STIG 版本 1 第 10 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 8 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 11 版
- RHEL 8 STIG 版本 1 第 10 版
- Ubuntu 18.04 STIG 版本 2 第 11 版
- Ubuntu 20.04 STIG 版本 1 第 8 版

2023 年第 1 季度變更 - 2023/03/27：

已更新 STIG 版本並針對 2023 年第一季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 10 版
- RHEL 8 STIG 版本 1 第 9 版
- Ubuntu 18.04 STIG 版本 2 第 10 版
- Ubuntu 20.04 STIG 版本 1 第 7 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 10 版
- RHEL 8 STIG 版本 1 第 9 版
- Ubuntu 18.04 STIG 版本 2 第 10 版
- Ubuntu 20.04 STIG 版本 1 第 7 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 10 版
- RHEL 8 STIG 版本 1 第 9 版
- Ubuntu 18.04 STIG 版本 2 第 10 版
- Ubuntu 20.04 STIG 版本 1 第 7 版

2022 年第 4 季度變更 - 2023/02/01：

已更新 STIG 版本並針對 2022 年第四季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 9 版
- RHEL 8 STIG 版本 1 第 8 版
- Ubuntu 18.04 STIG 版本 2 第 9 版
- Ubuntu 20.04 STIG 版本 1 第 6 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 9 版

- RHEL 8 STIG 版本 1 第 8 版
- Ubuntu 18.04 STIG 版本 2 第 9 版
- Ubuntu 20.04 STIG 版本 1 第 6 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 9 版
- RHEL 8 STIG 版本 1 第 8 版
- Ubuntu 18.04 STIG 版本 2 第 9 版
- Ubuntu 20.04 STIG 版本 1 第 6 版

2022 年第 3 季度變更 - 2022/09/30 (無變更)：

2022 年第三季發行版本的 Linux 元件 STIGS 沒有變更。

2022 年第 2 季度變更 - 2022/08/02：

已推出 Ubuntu 支援，更新 STIG 版本，並針對 2022 年第二季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 7 版
- RHEL 8 STIG 版本 1 第 6 版
- Ubuntu 18.04 STIG 版本 2 第 6 版 (新)
- Ubuntu 20.04 STIG 版本 1 第 4 版 (新)

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 7 版
- RHEL 8 STIG 版本 1 第 6 版
- Ubuntu 18.04 STIG 版本 2 第 6 版 (新)
- Ubuntu 20.04 STIG 版本 1 第 4 版 (新)

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 7 版

- RHEL 8 STIG 版本 1 第 6 版
- Ubuntu 18.04 STIG 版本 2 第 6 版 (新)
- Ubuntu 20.04 STIG 版本 1 第 4 版 (新)

2022 年第 1 季度變更 - 2022/04/26：

重構以包含更佳的容器支援。將先前的 AL2 指令碼與 RHEL 7 結合。已更新 STIG 版本並針對 2022 年第一季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 6 版
- RHEL 8 STIG 版本 1 第 5 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 6 版
- RHEL 8 STIG 版本 1 第 5 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 6 版
- RHEL 8 STIG 版本 1 第 5 版

2021 年第 4 季度變更 - 2021/12/20：

已更新 STIG 版本並針對 2021 年第四季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 5 版
- RHEL 8 STIG 版本 1 第 4 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 5 版
- RHEL 8 STIG 版本 1 第 4 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 5 版
- RHEL 8 STIG 版本 1 第 4 版

2021 年第 3 季度變更 - 2021/09/30：

已更新 STIG 版本並針對 2021 年第三季版本套用 STIGS，如下所示：

Linux STIG Low (類別 III)

- RHEL 7 STIG 版本 3 第 4 版
- RHEL 8 STIG 版本 1 第 3 版

Linux STIG Medium (類別 II)

- RHEL 7 STIG 版本 3 第 4 版
- RHEL 8 STIG 版本 1 第 3 版

Linux STIG High (類別 I)

- RHEL 7 STIG 版本 3 第 4 版
- RHEL 8 STIG 版本 1 第 3 版

Windows STIG 版本歷史記錄

本節記錄每季更新的 Windows 指令碼套件的版本歷史記錄。若要查看季度的變更和已發佈的版本，請選擇標題以展開資訊。如果該季度沒有變更，會在標題中有所反映。

2026 年Q1變更 - 04/07/2026：

新增對 Windows Server 2025 作業系統的支援，並更新下列 STIG 版本，並針對所有合規層級 (low/medium/high) 套用 2026 年第一季版本的 STIGS：

STIG-Build-Windows 1.0.x 版

- Windows Server 2025 STIG 第 1 版第 1 版
- Windows Server 2022 STIG 第 2 版第 7 版

- Windows Server 2019 STIG 第 3 版第 7 版
- Microsoft .NET Framework 4.0 STIG 第 2 版第 7 版
- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 6 版
- Microsoft Edge STIG 第 2 版第 4 版 （ 僅限 Windows Server 2022 和 2025)

2025 年Q4 季變更 - 12/10/2025 （ 無變更 ）：

2025 年第 4 季發行版本的 Windows 元件 STIGS 沒有變更。

2025 年第 3 季度變更 - 2025/9/04 (無變更)：

2025 年第三季發行版本的 Windows 元件 STIGS 沒有變更。

2025 年第 2 季度變更 - 2025/06/26：

已更新 STIG 版本並針對 2025 年第 2 季度版本套用 STIGS，如下所示：

Windows STIG Low (類別 III)

- Windows Server 2022 STIG 第 2 版第 4 版
- Windows Server 2019 STIG 版本 3 第 4 版
- Windows Server 2016 STIG 第 2 版第 10 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 6 版
- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)

Windows STIG Medium (類別 II)

- Windows Server 2022 STIG 第 2 版第 4 版
- Windows Server 2019 STIG 版本 3 第 4 版
- Windows Server 2016 STIG 第 2 版第 10 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 6 版

- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版

Windows STIG High (類別 I)

- Windows Server 2022 STIG 第 2 版第 4 版
- Windows Server 2019 STIG 版本 3 第 4 版
- Windows Server 2016 STIG 第 2 版第 10 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 6 版
- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版

2025 年第 1 季度變更 - 2025/05/04：

針對 2025 年第一季發布的所有 STIG 元件，更新了 STIGS for Internet Explorer 11 STIG 版本 2 第 5 版。

2024 年第 4 季度變更 - 2024/12/10：

已更新 STIG 版本並針對 2024 年第 4 季度版本套用 STIGS，如下所示：

Windows STIG Low (類別 III)

- Windows Server 2022 STIG 第 2 版第 2 版
- Windows Server 2019 STIG 版本 3 第 2 版
- Windows Server 2016 STIG 第 2 版第 9 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 第 2 版第 2 版

- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)

Windows STIG Medium (類別 II)

- Windows Server 2022 STIG 第 2 版第 2 版
- Windows Server 2019 STIG 版本 3 第 2 版
- Windows Server 2016 STIG 第 2 版第 9 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版

Windows STIG High (類別 I)

- Windows Server 2022 STIG 第 2 版第 2 版
- Windows Server 2019 STIG 版本 3 第 2 版
- Windows Server 2016 STIG 第 2 版第 9 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 第 2 版第 2 版
- Internet Explorer 11 STIG 第 2 版第 5 版
- Microsoft Edge STIG 版本 2 第 2 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版

2024 年第 3 季度變更 - 2024/10/04 (無變更)：

2024 年第三季發行版本的 Windows 元件 STIGS 沒有變更。

2024 年第 2 季度變更 - 2024/05/10 (無變更)：

2024 年第二季發行版本的 Windows 元件 STIGS 沒有變更。

2024 年第 1 季度變更 - 2024/02/23 (無變更)：

2024 年第一季發行版本的 Windows 元件 STIGS 沒有變更。

2023 年第 4 季度變更 - 2023/12/07 (無變更)：

2023 年第四季發行版本的 Windows 元件 STIGS 沒有變更。

2023 年第 3 季度變更 - 2023/10/04 (無變更)：

2023 年第三季發行版本的 Windows 元件 STIGS 沒有變更。

2023 年第 2 季度變更 - 2023/05/03 (無變更)：

2023 年第二季發行版本的 Windows 元件 STIGS 沒有變更。

2023 年第 1 季度變更 - 2023/03/27 (無變更)：

2023 年第一季發行版本的 Windows 元件 STIGS 沒有變更。

2022 年第 4 季度變更 - 2023/02/01：

已更新 STIG 版本並針對 2022 年第 4 季度版本套用 STIGS，如下所示：

Windows STIG Low (類別 III)

- Windows Server 2022 STIG 版本 1 第 1 版
- Windows Server 2019 STIG 版本 2 第 5 版
- Windows Server 2016 STIG 版本 2 第 5 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 版本 2 第 3 版
- Microsoft Edge STIG 版本 1 第 6 版 (僅限 Windows Server 2022)

Windows STIG Medium (類別 II)

- Windows Server 2022 STIG 版本 1 第 1 版
- Windows Server 2019 STIG 版本 2 第 5 版
- Windows Server 2016 STIG 版本 2 第 5 版

- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 版本 2 第 3 版
- Microsoft Edge STIG 版本 1 第 6 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版 (僅限 Windows Server 2022)

Windows STIG High (類別 I)

- Windows Server 2022 STIG 版本 1 第 1 版
- Windows Server 2019 STIG 版本 2 第 5 版
- Windows Server 2016 STIG 版本 2 第 5 版
- Windows Server 2012 R2 MS STIG 版本 3 第 5 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 2 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 版本 2 第 3 版
- Microsoft Edge STIG 版本 1 第 6 版 (僅限 Windows Server 2022)
- Defender STIG 版本 2 第 4 版 (僅限 Windows Server 2022)

2022 年第 3 季度變更 - 2022/09/30 (無變更)：

2022 年第三季發行版本的 Windows 元件 STIGS 沒有變更。

2022 年第 2 季度變更 - 2022/08/02：

已更新 STIG 版本，並將 STIGS 套用於 2022 年第 2 季發行版本。

Windows STIG Low (類別 III)

- Windows Server 2019 STIG 第 2 版第 4 版
- Windows Server 2016 STIG 第 2 版第 4 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG Medium (類別 II)

- Windows Server 2019 STIG 第 2 版第 4 版
- Windows Server 2016 STIG 第 2 版第 4 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG High (類別 I)

- Windows Server 2019 STIG 第 2 版第 4 版
- Windows Server 2016 STIG 第 2 版第 4 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

2022 年第 1 季度變更 - 2022/08/02 (無變更)：

2022 年第一季發行版本的 Windows 元件 STIGS 沒有變更。

2021 年第 4 季度變更 - 2021/12/20：

已更新 STIG 版本，並將 STIGS 套用於 2021 年第四季發行版本。

Windows STIG Low (類別 III)

- Windows Server 2019 STIG 第 2 版第 3 版
- Windows Server 2016 STIG 第 2 版第 3 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG Medium (類別 II)

- Windows Server 2019 STIG 第 2 版第 3 版
- Windows Server 2016 STIG 第 2 版第 3 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG High (類別 I)

- Windows Server 2019 STIG 第 2 版第 3 版
- Windows Server 2016 STIG 第 2 版第 3 版
- Windows Server 2012 R2 MS STIG 第 3 版第 3 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 2 第 1 版
- Internet Explorer 11 STIG 第 1 版第 19 版

2021 年第 3 季度變更 - 2021/09/30：

已更新 STIG 版本，並將 STIGS 套用於 2021 年第三季發行版本。

Windows STIG Low (類別 III)

- Windows Server 2019 STIG 第 2 版第 2 版
- Windows Server 2016 STIG 第 2 版第 2 版
- Windows Server 2012 R2 MS STIG 第 3 版第 2 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 1 第 7 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG Medium (類別 II)

- Windows Server 2019 STIG 第 2 版第 2 版

- Windows Server 2016 STIG 第 2 版第 2 版
- Windows Server 2012 R2 MS STIG 第 3 版第 2 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 1 第 7 版
- Internet Explorer 11 STIG 第 1 版第 19 版

Windows STIG High (類別 I)

- Windows Server 2019 STIG 第 2 版第 2 版
- Windows Server 2016 STIG 第 2 版第 2 版
- Windows Server 2012 R2 MS STIG 第 3 版第 2 版
- Microsoft .NET Framework 4.0 STIG 版本 2 第 1 版
- Windows Firewall STIG 版本 1 第 7 版
- Internet Explorer 11 STIG 第 1 版第 19 版

使用 AWS Systems Manager 將 STIG 設定套用至您的執行個體

可以使用 `AWSEC2-ConfigureSTIG` Systems Manager 命令文件，將 STIG 設定套用至現有的 EC2 執行個體。您必須從其更新的執行個體中執行命令文件。命令文件會根據其執行所在的執行個體之作業系統和組態，套用適當的設定。

此頁面包含 `AWSEC2-ConfigureSTIG` 命令文件的詳細資訊，包括輸入參數，以及如何在 Systems Manager 主控台或在 AWS CLI中使用 `send-command` 來執行它。

AWSEC2-ConfigureSTIG 輸入參數

您可以提供下列輸入參數，以指定命令文件應如何將 STIG 設定套用至執行個體。

Level (字串，必填)

指定要套用的 STIG 嚴重性類別。有效值包括下列項目：

- 高
- 中
- 低

如果不指定值，系統會預設為 `High`。

InstallPackages (字串，選用 – 僅限 Linux)

如果值為 No，指令碼不會安裝任何其他軟體套件。如果值為 Yes，指令碼會安裝最高合規所需的其他軟體套件。預設值為 No。

SetDoDConsentBanner (字串，選用 – 僅限 Linux)

如果值為 No，則當連接至已安裝 Linux STIG 指令碼的執行個體時，不會顯示 DoD 同意橫幅。如果值為 Yes，則當連接至已安裝其中一個 STIG Linux 指令碼的執行個體時，登入前會顯示 DoD 同意橫幅。必須先確認橫幅，才能登入。預設值為 No。

如需同意橫幅的範例，請參閱當您存取 DLA Document Services 網站時顯示的國防部隱私權免責聲明和同意通知。

執行 AWSEC2-ConfigureSTIG 命令文件

若要執行 AWSEC2-ConfigureSTIG 文件，請依照適用於您偏好之環境的步驟執行。

Console

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 從導覽窗格中選取執行命令。這會顯示您帳戶中目前正在執行的命令清單 (若適用)。
3. 選擇執行命令。這會開啟執行命令對話方塊，並顯示您有權存取的命令文件清單。
4. 從命令文件清單中選取 AWSEC2-ConfigureSTIG。若要簡化結果，可輸入全部或部分文件名稱。您也可以依擁有者、平台類型或標籤進行篩選。

   在您選取命令文件時，系統會將詳細資訊填入清單下方。
5. 從文件版本清單中選取 Default version at runtime。
6. 設定命令參數以定義 AWSEC2-ConfigureSTIG 如何安裝指令碼套件，並執行它來更新執行個體。如需參數詳細資訊，請參閱 AWSEC2-ConfigureSTIG 輸入參數。
7. 針對目標選取，手動指定標籤或選取執行個體，以識別要執行這項操作的執行個體。

> **Note**
>
> 如果您手動選取執行個體，但預期看到的執行個體並未出現在清單中，請參閱我的執行個體在哪裡？，以取得疑難排解秘訣。

8. 如需定義 Systems Manager Run Command 行為的其他參數 (例如速率控制)，請依照從主控台執行命令中的說明輸入值。
9. 選擇執行。

   如果成功，命令文件會安裝指令碼並設定執行個體。如果命令執行失敗，請檢視 Systems Manager 命令輸出的詳細資訊，以了解執行失敗的原因。

AWS CLI

範例 1：使用預設值執行

執行下列命令來安裝 STIG 指令碼，並使用預設值執行該指令碼。如需有關輸入參數的詳細資訊，請參閱 AWSEC2-ConfigureSTIG 輸入參數。

```
aws ssm send-command \
 --document-name "AWSEC2-ConfigureSTIG" \
 --instance-ids "i-1234567890abcdef0"'
```

範例 2：在執行個體上設定中級 STIG 設定

執行下列命令來安裝 STIG 指令碼，並使用設定為 Medium 的 Level 輸入參數來執行它。如需有關輸入參數的詳細資訊，請參閱 AWSEC2-ConfigureSTIG 輸入參數。

```
aws ssm send-command \
 --document-name "AWSEC2-ConfigureSTIG" \
 --instance-ids "i-1234567890abcdef0"
 --parameters '{"Level":"Medium"}'
```

如果成功，命令文件會安裝指令碼並設定執行個體。如果命令執行失敗，請檢視命令輸出的詳細資訊，以了解執行失敗的原因。

PowerShell

範例 1：使用預設值執行

執行下列命令來安裝 STIG 指令碼，並使用預設值執行該指令碼。如需有關輸入參數的詳細資訊，請參閱 AWSEC2-ConfigureSTIG 輸入參數。

```
Send-SSMCommand -DocumentName "AWSEC2-ConfigureSTIG" -InstanceId
 "i-1234567890abcdef0"}
```

範例 2：在執行個體上設定中級 STIG 設定

執行下列命令來安裝 STIG 指令碼，並使用設定為 `Medium` 的 `Level` 輸入參數來執行它。如需有關輸入參數的詳細資訊，請參閱 AWSEC2-ConfigureSTIG 輸入參數。

```
Send-SSMCommand -DocumentName "AWSEC2-ConfigureSTIG" -InstanceId
 "i-1234567890abcdef0" -Parameter @{'Level'='Medium'}
```

如果成功，命令文件會安裝指令碼並設定執行個體。如果命令執行失敗，請檢視命令輸出的詳細資訊，以了解執行失敗的原因。

EC2 執行個體上的精確時鐘和時間同步

對許多伺服器任務和流程而言，Amazon EC2 執行個體上一致且精準的時間參考至關重要。在識別問題發生的時間和事件的時間順序方面，系統日誌中的時間戳記扮演著至關重要的角色。當您使用 AWS CLI 或 AWS 開發套件從執行個體發出請求時，這些工具會代表您簽署請求。如果您執行個體的日期和時間設定不正確，可能會導致簽章中的日期與請求日期之間的差異，進而 AWS 拒絕您的請求。

為了解決這個重要層面，Amazon 提供 Amazon Time Sync Service，此服務可從所有 EC2 執行個體存取和提供不同 AWS 服務使用。此服務在每個 中使用衛星連接和原子參考時鐘機群 AWS 區域 ，以提供國際標準時間 (UTC) 全球標準的準確和目前時間讀數。

為獲得最佳效能，建議您在 EC2 執行個體上使用本機 Amazon Time Sync Service。對於備份到執行個體上的本機 Amazon Time Sync Service，以及將 Amazon EC2 以外的資源連線至 Amazon Time Sync Service，您可以使用位於 `time.aws.com` 的公有 Amazon Time Sync Service。公有 Amazon Time Sync Service (例如本機 Amazon Time Sync Service)，會自動消彌加至 UTC 的任何閏秒。公有 Amazon Time Sync Service 受到我們每個叢集中衛星連線和原子參考時鐘的全球支援 AWS 區域。

硬體封包時間戳記

您可在執行個體上啟用硬體封包時間戳記功能，為每個傳入的網路封包新增 64 位元奈秒精確度的時間戳記。由於硬體封包時間戳記在硬體層級執行，早於封包抵達核心、通訊端或應用程式層級，因此可規避軟體時間戳記帶來的任何延遲。硬體時間戳記的基礎參考時鐘是 Amazon Time Sync Service 的 PTP 硬體時鐘。

優勢

硬體封包時間戳記提供以下優點：

- 改善事件排序，也可用於判斷封包到達 EC2 執行個體的實際順序，以確保公平的封包處理。
- 測量單向網路延遲。
- 與大多數內部部署解決方案相比，透過更高的精確度與準確度提升分散式交易速度。

先決條件與組態

若要啟用硬體封包時間戳記，您的執行個體必須符合以下先決條件：

- 必須是 Linux 執行個體。
- 滿足支援 PTP 硬體時鐘的要求。

有關組態指示，請參閱 GitHub 上彈性網路介面卡 (ENA) 系列 Linux 核心驅動程式頁面上的硬體封包時間戳記。

閏秒

閏秒於 1972 年推出，偶爾會針對 UTC 時間進行一秒的調整，以考量地球旋轉中的不規則性，以適應國際原子時間 (TAI) 和太陽時間 (Ut1) 之間的差異。為了代表客戶管理閏秒，我們在 Amazon Time Sync Service 中設計了閏秒消彌機制。如需詳細資訊，請參閱閏秒搶先報：即將到來的閏秒和 AWS。

國際度量衡局第 27 屆大會決議，將在 2035 年之前取消閏秒，我們全力支持這項決議。

為了支援這次轉變，我們仍計劃透過本機 NTP 連線或公有 NTP 集區 (`time.aws.com`) 存取 Amazon Time Sync Service 時，在閏秒事件期間消彌時間。不過 PTP 硬體時鐘不提供消彌時間選項。如果發生閏秒，PTP 硬體時鐘會依照 UTC 標準加上閏秒。在大多數情況下，消彌的閏秒和閏秒時間來源相同。但由於閏秒事件期間會有所不同，所以不建議在閏秒事件期間，在您的時間用戶端組態中同時使用消彌和未消彌的時間來源。

主題

- 設定 EC2 執行個體的時間參考以使用本機 Amazon Time Sync Service
- 設定 EC2 執行個體上的時間參考，或將任何網際網路連線裝置設定為使用公用的 Amazon Time Sync Service
- 比較 Linux 執行個體的時間戳記
- 變更執行個體的時區

相關資源

- AWS 運算部落格：這是關於時間：Amazon EC2 執行個體上的微秒準確時鐘
- AWS 雲端操作與遷移部落格：使用 Amazon Time Sync Service 和 Amazon CloudWatch 管理 Amazon EC2 執行個體時鐘準確性 – 第 1 部分 Amazon CloudWatch
- (Linux) https://chrony-project.org/

設定 EC2 執行個體的時間參考以使用本機 Amazon Time Sync Service

Amazon Time Sync Service 提供數種方法，讓您的 Amazon EC2 執行個體同步至本機時間來源。首先，任何 Amazon EC2 執行個體都可以透過網路時間通訊協定 (NTP) 連線到本機時間來源。此外，增強型 Amazon Time Sync Service 為支援的 Amazon EC2 執行個體提供更精確的本機時間來源。在具有 `precision-time`策略的置放群組中啟動支援的執行個體，以達到更精確的 NTP 來源。最後，在精確時間置放群組中啟動的 Linux 執行個體可以存取 PTP 硬體時鐘 (PHC) 裝置，並能夠擷取硬體封包時間戳記。

任何 Amazon EC2 執行個體都可以存取本機 NTP 來源。您可以透過連結本機 IP 地址來連接 NTP 來源，這會將此流量限制在您的 VPC 內，而不需要特定的 VPC 組態變更。您的 AMI 可能已將時鐘同步常駐程式設定為預設使用本機 NTP 來源。此 NTP 來源可透過下列 IP 地址使用：

- IPv4：`169.254.169.123`
- IPv6：`fd00:ec2::123` (僅在 Nitro 型執行個體上存取。)

支援的 Amazon EC2 執行個體可存取增強型 Amazon Time Sync Service。若要存取增強型 Amazon Time Sync Service，請使用 `precision-time`策略在置放群組中啟動支援的執行個體。如果您使用 NTP 連結本機 IP 地址，則不需要將執行個體設定為受益於此改進。任何作業系統都可以使用此增強功能。您可以使用您選擇的 NTP 用戶端，驗證您是否受益於增強型 NTP 來源。

在支援的執行個體系列上執行的 Linux 型 AMIs 具有從 PHC 裝置取得時間的額外選項。ENA 驅動程式讓此裝置可供使用。增強版 NTP 來源和 PHC 裝置都使用相同的高度準確時間來源。PHC 時間來源的存取已最佳化，因此可更準確地同步 Amazon EC2 執行個體。

考量事項

- NTP 時間來源提供 UTC 時間範圍的閏點貼圖檢視，而 PHC 不會貼圖時間。如需詳細資訊，請參閱閏秒。

- 只有 Linux 執行個體可以存取本機 PTP 硬體時鐘。Windows 執行個體必須使用 NTP 來存取增強型 Amazon Time Sync Service。
- 使用連結本機位址的服務有每秒 1024 個封包 (PPS) 限制。此限制包括 Route 53 Resolver DNS 查詢、執行個體中繼資料服務 (IMDS) 請求、Amazon Time Sync Service Network Time Protocol (NTP) 請求和 Windows Licensing Service （適用於 Microsoft Windows 型執行個體）請求的彙總。

目錄

- 存取 Amazon Time Sync Service 的 IPv4 端點
- 存取 Amazon Time Sync Service 的 IPv6 端點
- 存取增強型 Amazon Time Sync Service
- 存取 PTP 硬體時鐘 (PHC)

存取 Amazon Time Sync Service 的 IPv4 端點

您的 AMI 可能依預設已設定 Amazon Time Sync Service。否則，請使用下列程序，將您的執行個體設定為透過 IPv4 端點使用本機 Amazon Time Sync Service。

如需對問題進行故障診斷的協助，請參閱針對 Linux 執行個體上的 NTP 同步問題進行故障診斷，或針對 Windows 執行個體上的時間問題進行故障診斷。

Amazon Linux

預設情況下，AL2023 和最新版的 Amazon Linux 2 設定為使用 Amazon Time Sync Service IPv4 端點。若您確認您的執行個體已設定，可以略過下列程序。

驗證 chrony 是否設定為使用 IPv4 端點

執行下列命令。在輸出中，以 ^* 開頭的行表示偏好的時間來源。

```
[ec2-user ~]$ chronyc sources -v | grep -F ^*
```

```
^* 169.254.169.123               3   4   377    13  -4325ns[-9201ns] +/-  401us
```

設定 chrony，以連線至舊版 Amazon Linux 2 上的 IPv4 端點

1. 連線到您的執行個體，然後解除安裝 NTP 服務。

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. 安裝 chrony 套裝服務。

```
[ec2-user ~]$ sudo yum install chrony
```

3. 使用文字編輯器 (例如 /etc/chrony.conf 或 vim) 開啟 nano 檔案。將下列程式碼新增在檔案中已有之任何其他 server 或 pool 的陳述式之前，然後儲存您的變更：

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

4. 重新啟動 chrony 協助程式 (chronyd)。

```
[ec2-user ~]$ sudo service chronyd restart
```

```
Starting chronyd:                                          [  OK  ]
```

> **Note**
>
> 在 RHEL 和 CentOS (最新為 6 版) 上，服務名稱為 chrony，而不是 chronyd。

5. 使用 chkconfig 命令設定 chronyd 在每次系統開機時啟動。

```
[ec2-user ~]$ sudo chkconfig chronyd on
```

6. 確認 chrony 是否使用 169.254.169.123 IPv4 端點來同步時間。

```
[ec2-user ~]$ chronyc sources -v | grep -F ^*
```

在輸出中，^* 表示偏好的時間來源。

```
^* 169.254.169.123               3   6    17    43    -30us[ -226us] +/-  287us
```

7. 確認時間同步指標是否由 chrony 回報。

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
```

```
Stratum         : 4
Ref time (UTC)  : Wed May 06 00:39:14 2026
System time     : 0.000002191 seconds fast of NTP time
Last offset     : +0.000002164 seconds
RMS offset      : 0.000082968 seconds
Frequency       : 3.710 ppm slow
Residual freq   : +0.002 ppm
Skew            : 0.504 ppm
Root delay      : 0.000362541 seconds
Root dispersion : 0.000225028 seconds
Update interval : 16.1 seconds
Leap status     : Normal
```

Ubuntu

設定 chrony，以連線至 Ubuntu 上的 IPv4 端點

1. 連線至您的執行個體，然後使用 apt 來安裝 chrony 套件。

```
ubuntu:~$ sudo apt install chrony
```

> **Note**
>
> 如有必要，請執行 sudo apt update 先更新您的執行個體。

2. 使用文字編輯器 (例如 /etc/chrony/chrony.conf 或 vim) 開啟 nano 檔案。將下列程式碼新增在檔案中已有之任何其他 server 或 pool 的陳述式之前，然後儲存您的變更：

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. 重新啟動 chrony 服務。

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
Restarting chrony (via systemctl): chrony.service.
```

4. 確認 chrony 是使用 169.254.169.123 IPv4 端點來同步時間。

```
ubuntu:~$ chronyc sources -v | grep -F ^*
```

在輸出中，以 ^* 開頭的行表示偏好的時間來源。

```
^* 169.254.169.123               3   6    17    12    +15us[  +57us] +/-  320us
```

5. 確認時間同步指標是否由 chrony 回報。

```
ubuntu:~$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed May 06 00:39:14 2026
System time     : 0.000002191 seconds fast of NTP time
Last offset     : +0.000002164 seconds
RMS offset      : 0.000082968 seconds
Frequency       : 3.710 ppm slow
Residual freq   : +0.002 ppm
Skew            : 0.504 ppm
Root delay      : 0.000362541 seconds
Root dispersion : 0.000225028 seconds
Update interval : 16.1 seconds
Leap status     : Normal
```

SUSE Linux

從 SUSE Linux Enterprise Server 15 開始，chrony 是 NTP 的預設實作。

設定 chrony，以連線至 SUSE Linux 上的 IPv4 端點

1. 使用文字編輯器 (例如 /etc/chrony.conf 或 vim) 開啟 nano 檔案。
2. 確認檔案是否包含下列程式碼行：

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

如果此程式碼不存在，請新增它。

3. 將任何其他伺服器或集區的程式碼行變更為註解。
4. 開啟 YaST 並啟用 chrony 服務。

Windows

從 2018 年 8 月版本開始，Windows AMI 預設會使用 Amazon Time Sync Service。從這些 AMI 啟動的執行個體不需要進一步的設定，您可以略過下列程序。

如果您使用的 AMI 未依預設設定 Amazon Time Sync Service，請先驗證您目前的 NTP 組態。如果您的執行個體已在使用 Amazon Time Sync Service 的 IPv4 端點，則無需進一步設定。如果您的執行個體不是使用 Amazon Time Sync Service，請完成程序，將 NTP 伺服器變更為使用 Amazon Time Sync Service。

確認 NTP 組態

1. 從執行個體中，開啟命令提示視窗。
2. 輸入下列命令，以取得目前 NTP 組態：

   ```
   w32tm /query /configuration
   ```

   此命令會傳回 Windows 執行個體目前的組態設定，並顯示是否已連接至 Amazon Time Sync Service。
3. (選用) 輸入下列命令，以取得目前組態的狀態：

   ```
   w32tm /query /status
   ```

   此命令會傳回資訊，例如執行個體最後一次與 NTP 伺服器同步的時間以及輪詢間隔。

將 NTP 伺服器改為使用 Amazon Time Sync Service

1. 從命令提示視窗，執行下列命令：

   ```
   w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
   ```
2. 使用下列命令，確認新設定：

   ```
   w32tm /query /configuration
   ```

   在傳回的輸出中，確認 `NtpServer` 顯示 `169.254.169.123` IPv4 端點。

Amazon Windows AMI 的預設 NTP 設定

Amazon Machine Images (AMIs) 通常依循立即可用的預設值，但需要變更才能在 EC2 基礎設施上運作的狀況除外。下列設定已確認在虛擬化環境中運作良好，以及可將任何時鐘漂移保持在一秒的準確度內：

- 更新間隔 – 規範時間服務將系統時間調整為準確性的頻率。 會將更新間隔 AWS 設定為每兩分鐘發生一次。
- NTP 伺服器：從 2018 年 8 月發行版本開始，AMI 預設使用 Amazon Time Sync Service。這次的服務可從 AWS 區域 169.254.169.123 IPv4 端點的任何 存取。此外，0x9 標記表示時間服務正作為用戶端，並使用 `SpecialPollInterval` 來判定以設定的時間伺服器進行查入的頻率。
- 類型 – “NTP” 表示服務將充當獨立式 NTP 用戶端，而不是充當網域的一部分。
- Enabled 和 InputProvider：時間服務已啟用，並提供時間給作業系統。
- 特殊輪詢間隔：每 900 秒 (15 分鐘) 檢查一次已設定的 NTP 伺服器。

> **Note**
>
> 對於 Windows Server 2025 AMIs，`SpecialPollInterval`值為 1024 秒，而不是 900 秒。

登錄檔路徑	金鑰名稱	資料
HKLM:\System\CurrentControlSet\services\w32time\Config	UpdateInterval	120
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	類型	NTP

登錄檔路徑	金鑰名稱	資料
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	Enabled	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900 (Windows Server 2016、2019 和 2022) 或 1024 (Windows Server 2025)

存取 Amazon Time Sync Service 的 IPv6 端點

本節說明如要設定執行個體以透過 IPv6 端點使用本機 Amazon Time Sync Service，步驟與 存取 Amazon Time Sync Service 的 IPv4 端點 所述有何不同。它不會說明整個 Amazon Time Sync Service 組態程序。

IPv6 端點僅在 Nitro 型執行個體上可供存取。

我們不建議同時使用 IPv4 和 IPv6 端點項目。IPv4 和 IPv6 NTP 封包來自您的執行個體的相同本機伺服器。同時設定 IPv4 和 IPv6 端點是不必要的做法，而且不會改善執行個體的時間準確性。

Linux

根據您使用的 Linux 發行版本，當您到達編輯 `chrony.conf` 檔案的步驟時，您將使用 Amazon Time Sync Service 的 IPv6 端點 (`fd00:ec2::123`)，而不是 IPv4 端點 (`169.254.169.123`)：

```
server fd00:ec2::123 prefer iburst minpoll 4 maxpoll 4
```

儲存檔案，並確認 chrony 是否使用 `fd00:ec2::123` IPv6 端點來同步時間：

```
[ec2-user ~]$ chronyc sources -v
```

在輸出中，如果有看到 `fd00:ec2::123` IPv6 端點，表示組態已完成。

Windows

當您到達將 NTP 伺服器變更為使用 Amazon Time Sync Service 的步驟時，您會使用 Amazon Time Sync Service (`fd00:ec2::123`) 的 IPv6 端點，而不是 IPv4 端點 (`169.254.169.123`)：

```
w32tm /config /manualpeerlist:fd00:ec2::123 /syncfromflags:manual /update
```

確認您的新設定正在使用 `fd00:ec2::123` IPv6 端點來同步時間：

```
w32tm /query /configuration
```

在輸出中，確認 `NtpServer` 顯示 `fd00:ec2::123` IPv6 端點。

存取增強型 Amazon Time Sync Service

增強型 Amazon Time Sync Service 為支援的 Amazon EC2 執行個體提供更準確的本機時間來源。在具有`precision-time`策略的置放群組中啟動的執行個體可以存取這些本機來源。對於需要更準確時間的應用程式，我們建議從連結本機 NTP 來源或 Linux 執行個體上的 PTP 硬體時鐘 (PHC) 裝置放置群組。當您在精確時間置放群組中啟動執行個體時， 會將執行個體 AWS 放在支援硬體上，並直接存取 AWS 基礎設施中的高精確時間來源。

主要優點

- 預設已改善 NTP 來源 — 如果您使用如上節所述的 NTP 連結本機 IP 地址，您的執行個體可以立即存取增強型本機 NTP 時間來源。
- 微秒精確時鐘同步 — 將您的 Amazon EC2 Linux 執行個體設定為使用 PTP 硬體時鐘裝置，並實現微秒精確時鐘同步。
- 簡化部署：單一置放策略可確保所有執行個體都具有精確的時間功能。
- 硬體封包時間戳記 — 存取網路測量的低階封包時間戳記。
- 無需額外費用 — 無需額外費用即可使用精密時間置放群組。

規則與限制

- 精確時間置放群組適用於所有 AWS 商業區域。
- 精確時間置放群組支援下列 Gen7 和更新版本的 Amazon EC2 執行個體系列：
  - 一般用途：M7a、M7g、M7gd、M7i、M7i-flex、M8a、M8g

  - 運算最佳化：C7a、C7gd、C7i、C7i-flex、C8g、C8gd
  - 記憶體最佳化：R7a、R7g、R7i、R8g、X8g
  - 儲存最佳化：I8g、I8ge
- 如果您在精確時間置放群組中啟動或啟動執行個體，且硬體不足，無法存取增強型 Amazon Time Sync Service，則請求會失敗。
- 如果您在精確的時間置放群組中停止執行個體，然後再次啟動它，它仍會在置放群組中執行。不過，如果硬體不足，無法授予增強型 Amazon Time Sync Service 的存取權，則啟動可能會失敗。
- Amazon EC2 會持續新增支援增強型 Amazon Time Sync Service 的硬體。如果您的請求因容量不足而失敗，請稍後重試或嘗試不同的可用區域。如需詳細資訊，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。
- 適用置放群組的規則和限制。如需詳細資訊，請參閱Amazon EC2 執行個體的置放群組

建立精確的時間置放群組

您可以使用 AWS CLI、 AWS 管理主控台或 AWS SDKs，透過指定 `precision-time`策略來建立精確的時間置放群組。

使用 AWS CLI

執行以下命令：

```
aws ec2 create-placement-group \
    --group-name my-precision-time-pg \
    --strategy precision-time
```

命令會傳回您在建立容量保留時使用的置放群組 ARN，以及您在啟動執行個體和連結置放群組時使用的群組名稱或群組 ID。

啟動執行個體

建立精確時間置放群組之後，請在啟動執行個體以存取增強型 Amazon Time Sync Service 時指定該群組：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7g.2xlarge \
    --placement GroupId=pg-0aaa1111111111111
```

驗證對增強型 Amazon Time Sync Service 的存取

在精確的時間置放群組中啟動執行個體後，您會受益於增強的 NTP 時間來源。

例如，如果您在執行個體上使用 chronyd 協助程式，您可以驗證 NTP 時間來源現在稱為 Stratum 1，並已改善時鐘準確度指標：

```
[ec2-user ~]$ chronyc sources
```

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 169.254.169.123               1   4   377     3  +3477ns[+4689ns] +/-   91us
```

確認時間同步指標是否由 chrony 回報。

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
Stratum         : 2
Ref time (UTC)  : Wed May 06 01:33:43 2026
System time     : 0.000000276 seconds fast of NTP time
Last offset     : +0.000000331 seconds
RMS offset      : 0.000001929 seconds
Frequency       : 2.870 ppm fast
Residual freq   : +0.000 ppm
Skew            : 0.031 ppm
Root delay      : 0.000107584 seconds
Root dispersion : 0.000036476 seconds
Update interval : 16.2 seconds
Leap status     : Normal
```

存取 PTP 硬體時鐘 (PHC)

PTP 硬體時鐘 (PHC) 是 AWS Nitro 系統的一部分。您可以直接在支援在精確時間置放群組中啟動的裸機和虛擬化 Amazon EC2 執行個體上存取。PHC 裝置目前只能在 Linux 執行個體上存取。下列各節說明如何在 Linux 執行個體上設定和驗證 PHC 裝置。

要求

- 在支援的作業系統上安裝 ENA 驅動程式 2.10.0 版或更新版本。

- 執行 Linux 的 Amazon EC2 執行個體。如需支援作業系統的詳細資訊，請參閱 GitHub 上的驅動程式先決條件。
- 在精確時間置放群組中啟動的 Amazon EC2 執行個體（請參閱 存取增強型 Amazon Time Sync Service)。

> **Note**
>
> 在以下區域和特定執行個體系列中，增強型 Amazon Time Sync Service 和 PHC 裝置無需精確的時間置放群組即可存取。為了獲得最佳體驗，建議您改為在精確置放群組中啟動 Amazon EC2 執行個體。
>
> - 支援傳統存取 AWS 區域：美國東部（維吉尼亞北部）、美國東部（俄亥俄）、亞太區域（馬來西亞）、亞太區域（泰國）、亞太區域（東京）和歐洲（斯德哥爾摩）
> - 舊版存取支援的本地區域：美國東部（紐約市）
> - 舊版存取支援的執行個體系列：
>   - 一般用途：M7a、M7g、M7i
>   - 記憶體最佳化：R7a、R7g、R7i
>   - 儲存最佳化：I8g、I8ge

使用 PHC 支援編譯和啟用 ENA 驅動程式

請一律檢閱 GitHub 上 ENA 驅動程式文件中最新 Elastic Network Adapter (ENA) 驅動程式版本的指示，以取得您作業系統的特定資訊。以下是 Amazon Linux 步驟的概觀。

開始之前，請確定您的執行個體符合 要求。

使用 PHC 支援編譯和啟用 ENA 驅動程式

1. 安裝先決條件。

```
[ec2-user ~]$ sudo yum update
[ec2-user ~]$ sudo yum install kernel-devel-$(uname -r) git
[ec2-user ~]$ sudo reboot
```

2. 載入所需的 ptp 模組。

```
[ec2-user ~]$ sudo modprobe ptp
```

3. 擷取最新的 ENA 驅動程式版本 (2.10.0 版或更新版本) 。

```
[ec2-user ~]$ git clone https://github.com/amzn/amzn-drivers.git /tmp/amzn-drivers
```

4. ena.ko 使用 PHC 支援建置 ENA 驅動程式。

```
[ec2-user ~]$ cd /tmp/amzn-drivers/kernel/linux/ena
[ec2-user ~]$ ENA_PHC_INCLUDE=1 make
```

5. 重新載入 ENA 驅動程式並啟用 PHC 裝置。

```
[ec2-user ~]$ sudo rmmod ena && sudo insmod ena.ko phc_enable=1
```

6. 驗證載入的 ENA 驅動程式具有 PHC 支援。

```
[ec2-user ~]$ modinfo ena | grep -E "phc_enable"
```

```
parm:           phc_enable:Enable PHC.
```

7. 驗證是否已使用 ENA 驅動程式啟用 PHC 支援。

```
[ec2-user ~]$ cat /sys/module/ena/parameters/phc_enable
```

```
1
```

如需在重新啟動時安裝驅動程式和啟用 phc_enable 選項的說明，請參閱 GitHub 上的 ENA 驅動程式 README。

驗證 PTP 裝置組態

確認 ENA PTP 硬體時鐘裝置顯示在執行個體上。

```
[ec2-user ~]$ for file in /sys/class/ptp/*; do echo -n "$file: "; cat "$file/
clock_name"; done
```

預期的輸出結果

```
/sys/class/ptp/ptp<index>: ena-ptp-<PCI slot>
```

其中：

- *index* 為核心註冊的 PTP 硬體時鐘索引。
- *PCI slot* 為 ENA 乙太網路控制器 PCI 插槽。此插槽與 `lspci | grep ENA` 中顯示的插槽相同。

範例輸出

```
/sys/class/ptp/ptp0: ena-ptp-05
```

如果 `ena-ptp-<PCI slot>` 不在輸出中，表示 ENA 驅動程式未正確安裝。檢閱 中的步驟使用 PHC 支援編譯和啟用 ENA 驅動程式。

設定 PTP 符號連結

PTP 裝置通常命名為 /dev/ptp0、/dev/ptp1 等，其索引取決於硬體初始化順序。建立符號連結可確保 chrony 等應用程式能一致參考正確的裝置，不受索引變更影響。

最新的 Amazon Linux 2023 AMI 包含建立 /dev/ptp_ena 符號連結的 udev 規則，指向與 ENA 主機相關聯的正確 /dev/ptp 項目。

首先，執行下列命令，檢查符號連結是否存在。

```
[ec2-user ~]$ ls -l /dev/ptp*
```

範例輸出

```
crw------- 1 root root 245, 0 Jan 31 2025 /dev/ptp0
lrwxrwxrwx 1 root root      4 Jan 31 2025 /dev/ptp_ena -> ptp0
```

其中：

- /dev/ptp*<index>* 為 PTP 裝置的路徑。
- /dev/ptp_ena 為固定的符號連結，指向同一個 PTP 裝置。

如果/dev/ptp_ena符號連結存在，請跳至 設定 chronyd 協助程式以使用 PHC 裝置。若遺失，請執行下列動作：

建立 PTP 符號連結

1. 新增下列 udev 規則。

```
[ec2-user ~]$ echo "SUBSYSTEM==\"ptp\", ATTR{clock_name}==\"ena-ptp-*\", SYMLINK += \"ptp_ena\"" | sudo tee -a /etc/udev/rules.d/53-ec2-network-interfaces.rules
```

2. 透過重新啟動執行個體或執行下列命令，重新載入 udev 規則。

```
[ec2-user ~]$ sudo udevadm control --reload-rules && sudo udevadm trigger
```

設定 chronyd 協助程式以使用 PHC 裝置

計時時鐘同步協助程式必須設定為使用 PHC 裝置作為額外的時間來源，並使用/dev/ptp_ena符號連結來識別裝置。

設定 chronyd 協助程式以使用 PHC 裝置

1. /etc/chrony.conf 使用文字編輯器編輯並新增以下行：

```
refclock PHC /dev/ptp_ena poll 0 delay 0.000010 prefer
```

2. 重新啟動 chrony。

```
[ec2-user ~]$ sudo systemctl restart chronyd
```

3. 確認 chrony 使用 PTP 硬體時鐘。PHC0 裝置必須是 Stratum 0 值的偏好時間來源。本機 NTP 時間來源（如果已設定）必須具有 Stratum 1 值。

```
[ec2-user ~]$ chronyc sources
```

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PHC0                          0   0   377     0   +184ns[ +198ns] +/- 5032ns
^- 169.254.169.123               1   4   377     8    -18us[  -18us] +/-  115us
```

驗證硬體封包時間戳記

載入 PHC 支援的 ENA 驅動程式可讓您存取硬體封包時間戳記。您可以使用 `ethtool -T` *`interface`*命令來驗證對此功能的支援。

如需使用硬體封包時間戳記的詳細資訊，請參閱封包時間戳記 Linux 文件。

```
[ec2-user ~]$ sudo ethtool -T ens5
```

```
Time stamping parameters for ens5:
Capabilities:
        software-transmit
        hardware-receive
        software-receive
        software-system-clock
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes: none
Hardware Receive Filter Modes:
        none
        all
```

如果輸出顯示在`hardware-receive`功能清單中，則您的執行個體可以使用硬體封包時間戳記。

設定 EC2 執行個體上的時間參考，或將任何網際網路連線裝置設定為使用公用的 Amazon Time Sync Service

您可以將執行個體或任何連接網際網路的裝置 (例如本機電腦或內部部署伺服器) 設定為使用可透過網際網路從 `time.aws.com` 存取的公有 Amazon Time Sync Service。您可以使用公有 Amazon Time Sync Service 做為本機 Amazon Time Sync Service 的備份，並將 外部的資源連線至 AWS Amazon Time Sync Service。

> **Note**
>
> 為取得最佳效能，建議您在執行個體上使用本機 Amazon Time Sync Service，並僅使用公有 Amazon Time Sync Service 作為備份。

請遵循於執行個體或裝置的作業系統指示。

Linux

若要將 Linux 執行個體或裝置設定為使用公有 Amazon Time Sync Service，請使用 chrony 或 ntpd

1. 使用文字編輯器編輯 `/etc/chrony.conf` (若使用 chrony) 或 `/etc/ntp.conf` (若使用 ntpd)，如下所示：

   a. 若要防止執行個體或裝置嘗試混合已消彌和未消彌的伺服器，請刪除以 `server` 開頭的行或變成註解，但連接本機 Amazon Time Sync Service 的任何現有連線除外。

   > **Important**
   >
   > 如果要將 EC2 執行個體設定為連接到公有 Amazon Time Sync Service，請勿移除以下行，這一行會將執行個體設定為連接到本機 Amazon Time Sync Service。本機 Amazon Time Sync Service 是更直接的連線，可提供更高的時鐘準確度。公有 Amazon Time Sync Service 只能當成備案。
   >
   > ```
   > server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
   > ```

   b. 新增以下行以連接至公有 Amazon Time Sync Service。

   ```
   pool time.aws.com iburst
   ```

2. 使用以下其中一個命令重新啟動常駐程式。

   - chrony

     ```
     sudo service chronyd force-reload
     ```

   - ntpd

     ```
     sudo service ntp reload
     ```

macOS

將您的 macOS 執行個體或裝置設定為使用公有 Amazon Time Sync Service

1. 開啟 System Preferences (系統偏好設定)。
2. 選擇 Date & Time (日期與時間)，然後選擇 Date & Time (日期與時間) 標籤。

3. 若要進行變更，請選擇鎖定圖示，然後在出現提示時輸入您的密碼。
4. 對於自動設定日期與時間，輸入 **time.aws.com**。

Windows

將您的 Windows 執行個體或裝置設定為使用公有 Amazon Time Sync Service

1. 開啟 Control Panel (主控台)。
2. 選擇 Date and Time (日期與時間) 圖示。
3. 選擇 Internet Time (網際網路時間) 標籤。如果您的 PC 是網域的一部分，就無法使用此索引標籤。在這種情況下，系統會與網域控制器同步時間。您可以將控制器設定為使用公有 Amazon Time Sync Service。
4. 選擇 Change settings (變更設定)。
5. 選取與網際網路時間伺服器同步的核取方塊。
6. 在伺服器旁，輸入 **time.aws.com**。

將您的 Windows Server 執行個體或裝置設定為使用公有 Amazon Time Sync Service

- 遵照 Microsoft 的指示，更新您的登錄檔。

比較 Linux 執行個體的時間戳記

如果您使用的是 Amazon Time Sync Service，則可以將 Amazon EC2 Linux 執行個體上的時間戳記與 ClockBound 進行比較，來判斷事件的真實時間。ClockBound 會測量 EC2 執行個體的時鐘準確度，並允許您檢查給定的時間戳記相對於執行個體的目前時鐘是過去還是未來。此資訊對於判斷 EC2 執行個體之間事件和交易的順序和一致性非常有用，而且與每個執行個體的地理位置無關。

ClockBound 是一種開放原始碼常駐程式和程式庫。若要進一步了解 ClockBound (包括安裝指示)，請參閱 GitHub 上的 ClockBound。

ClockBound 僅支援 Linux 執行個體。

如果您正在使用連接 PTP 硬體時鐘的直接 PTP 連線，您的時間常駐程式 (例如 chrony) 將低估時鐘誤差範圍。這是因為 PTP 硬體時鐘不會像 NTP 那樣傳遞正確的誤差範圍資訊至 chrony。因此，時鐘同步處理常駐程式會假設時鐘是根據 UTC 調整準確度，因此會有誤差範圍 0。為了測量完整的誤差範圍，Nitro 系統會計算 PTP 硬體時鐘的誤差範圍，並透過 ENA 驅動程式 `sysfs` 檔案系統提供給 EC2 執行個體。您可以直接將此值讀取為值，以奈秒為單位。

擷取 PTP 硬體時鐘錯誤繫結

1. 首先使用下列其中一個命令，取得 PTP 硬體時鐘裝置的正確位置。命令中的路徑會根據用來啟動執行個體的 AMI 而有所不同。

   - 針對 Amazon Linux 2：

     ```
     cat /sys/class/net/eth0/device/uevent | grep PCI_SLOT_NAME
     ```

   - 針對 Amazon Linux 2023：

     ```
     cat /sys/class/net/ens5/device/uevent | grep PCI_SLOT_NAME
     ```

   輸出是 PCI 插槽名稱，也就是 PTP 硬體時鐘裝置的位置。在此範例中，位置為 `0000:00:03.0`。

   ```
   PCI_SLOT_NAME=0000:00:03.0
   ```

2. 若要擷取 PTP 硬體時鐘錯誤繫結，請執行下列命令。包含上一個步驟的 PCI 插槽名稱。

   ```
   cat /sys/bus/pci/devices/0000:00:03.0/phc_error_bound
   ```

   輸出即為 PTP 硬體時鐘的時鐘誤差範圍 (以奈秒為單位)。

若要計算使用直接 PTP 連線連接 PTP 硬體時鐘時，特定時間點的正確時鐘誤差範圍，您必須在 chrony 輪詢 PTP 硬體時鐘時，從 chrony 或 ClockBound 新增時鐘誤差範圍。如需有關測量和監控時鐘準確度的詳細資訊，請參閱使用 Amazon Time Sync Service 和 Amazon CloudWatch 管理 Amazon EC2 執行個體時鐘準確度：第 1 部分。

變更執行個體的時區

Amazon EC2 執行個體預設為 UTC (國際標準時間) 時區。您可以將執行個體上的時間變更為當地時區或網路中的另一個時區。

請遵循於執行個體作業系統的指示。

Linux

> **Important**
>
> 此資訊適用於 Amazon Linux。如需其他分發的詳細資訊，請參閱其特定文件。

變更 Amazon Linux 的時區

1. 檢視系統目前的時區設定。

```
[ec2-user ~]$ timedatectl
```

2. 列出可用的時區。

```
[ec2-user ~]$ timedatectl list-timezones
```

3. 設定選擇的時區。

```
[ec2-user ~]$ sudo timedatectl set-timezone America/Vancouver
```

4. (選用) 再次執行 timedatectl 命令確認目前的時區已更新為新的時區。

```
[ec2-user ~]$ timedatectl
```

Windows

變更 Windows 執行個體的時區

1. 從執行個體中，開啟命令提示視窗。
2. 找出執行個體使用的時區。若要取得時區清單，請使用下列命令：

```
tzutil /l
```

此命令會傳回所有可用時區的清單，格式如下：

```
display name
time zone ID
```

3. 找出要指派給執行個體的時區 ID。
4. 範例：指派 UTC 時區：

```
tzutil /s "UTC"
```

範例：指派太平洋標準時間：

```
tzutil /s "Pacific Standard Time"
```

當您變更 Windows 執行個體上的時區時，必須確定在系統重新啟動之後仍然會持續保留時區。否則，執行個體重新啟動時，會回復成使用 UTC 時間。您可以新增 RealTimeIsUniversal 登錄金鑰來保留時區設定。依預設，此機碼會在所有最新一代的執行個體上設定。若要驗證是否已設定RealTimeIsUniversal登錄機碼，請參閱下列程序中的步驟 3。如果未設定機碼，請從頭開始執行下列步驟。

設定 RealTimeIsUniversal 登錄機碼

1. 從執行個體中，開啟命令提示視窗。
2. 使用下列命令，新增登錄機碼：

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /
v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. (選用) 使用下列命令，確認執行個體已順利儲存機碼：

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
\TimeZoneInformation" /s
```

此命令會傳回 TimeZoneInformation 登錄機碼的子機碼。您應該會在清單底端看到 RealTimeIsUniversal 機碼，類似以下：

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation
    Bias                            REG_DWORD     0x1e0
    DaylightBias                    REG_DWORD     0xfffffffc4
    DaylightName                    REG_SZ        @tzres.dll,-211
    DaylightStart                   REG_BINARY    00000300020002000000000000000000
    StandardBias                    REG_DWORD     0x0
    StandardName                    REG_SZ        @tzres.dll,-212
```

```
    StandardStart                   REG_BINARY    00000B0001000200000000000000000
    TimeZoneKeyName                 REG_SZ        Pacific Standard Time
    DynamicDaylightTimeDisabled     REG_DWORD     0x0
    ActiveTimeBias                  REG_DWORD     0x1a4
    RealTimeIsUniversal             REG_DWORD     0x1
```

EC2 容量管理器

Amazon EC2 Capacity Manager 可協助您監控、分析和管理從單一位置跨隨需執行個體、Spot 執行個體和容量預留的容量使用量。容量管理器透過統一的儀表板，能夠每小時進行重新整理及最佳化來彙總您的用量資料，進而可簡化容量追蹤。

容量管理器專門針對大型企業與多帳戶組織設計，無須從多個來源手動收集資料。此工具可為您的 提供深入的洞見 AWS 區域。您能夠深入探究詳細的容量指標，然後立即執行動作來最佳化您的雲端資源。

當您啟用 AWS Organizations 整合時，您可以從單一儀表板檢視和分析組織中所有成員帳戶的容量資料。如果沒有 Organizations 整合，Capity Manager 只會監控啟用 AWS 帳戶 資源的個別內資源。

> **Note**
>
> - AWS 區域 每個帳戶只能啟用一個容量管理員。
> - 容量管理器中顯示的全部費用是依據發布的隨需定價估計的費用。這些估計費用不含 Savings Plans 或預留執行個體等折扣。實際費用可能與這些估計費用有所差異。

主要功能

- 儀表板 — 提供所有隨需容量保留、隨需和 Spot 用量的高階概觀，包括有助於改善容量狀態的關鍵指標。
- 跨帳戶和跨區域彙總：從單一儀表板檢視組織中所有成員帳戶的容量使用量。Capacity Manager 會將分割區中所有區域的資料彙總到您已啟用 Capacity Manager 的單一區域，包括標籤資料。對於跨帳戶資料，容量管理員也會提供帳戶名稱做為維度。
- 標籤型分組和篩選 — 從 Amazon EC2 資源 （ 例如 `environment`或 `team`) 啟用標籤金鑰，以在分組和篩選容量指標時用作維度。根據預設，會包含容量管理員提供的 EC2 Auto Scaling 群組、EKS 叢集名稱、EKS Kubernetes 節點集區和 Karpenter 節點集區的標籤。如需詳細資訊，請參閱管理受監控的標籤金鑰。

- 資料匯出 — 以 CSV 或 Parquet 格式將容量資料匯出至 Amazon S3，以進行進一步分析和自訂報告。
- APIs — 使用 `GetCapacityManagerMetricData`和 以程式設計方式查詢容量指標`GetCapacityManagerMetricDimensions`。
- 日期選擇器 — 分析日期範圍內的容量資料，從一小時到 90 天不等。
- 超過 30 個指標 — 使用超過 30 個指標跨容量保留、隨需執行個體和 Spot 執行個體追蹤容量。

啟用 EC2 容量管理器

視乎您的要求，能夠在兩個不同層級啟用容量管理器。您可以讓單一 AWS 帳戶 監控該帳戶中的資源，或將其與 AWS Organizations 整合，以集中查看組織中所有成員帳戶。

- 組織層級：透過單一儀表板，檢視及分析組織中所有會員帳戶的容量資料。
- 帳戶層級：監控單一 AWS 帳戶中的容量用量。

主題

- 使用 AWS Organizations 啟用 EC2 Capacity Manager
- 在帳戶層級啟用 EC2 容量管理器
- 註冊委派管理員
- 使用 EC2 容量管理器的服務連結角色

使用 AWS Organizations 啟用 EC2 Capacity Manager

您可以使用 AWS Organizations 啟用 EC2 Capacity Manager，以跨所有成員帳戶的組織層級可見性和管理容量。藉助此整合，您可在一個中央位置來監控、分析及管理容量用量。

管理帳戶負責在整個組織，啟用組織層級的存取權及管理容量。

使用 AWS Organizations 啟用容量管理員可提供下列優點：

- 集中式容量可見度 – 在單一儀表板中，透過跨帳戶及跨區域彙總，檢視您組織中所有會員帳戶的容量用量。
- 在整個組織實現最佳化 – 確定您組織中所有帳戶未使用的容量保留及最佳化機會。
- 委派管理員 – 在確保適當存取控制的同時，允許特定會員帳戶來管理組織的容量管理器。

如果您未啟用與 AWS Organizations 的整合，則只能監控您啟用 Capacity Manager AWS 帳戶 之個別中的資源。

先決條件

- 您必須使用管理帳戶和一或多個成員帳戶來設定 AWS Organizations。如需帳戶類型的詳細資訊，請參閱《AWS Organizations 使用者指南》中的術語與概念。
- 管理帳戶必須擁有執行以下 IAM 動作的許可：
  - `organizations:EnableAwsServiceAccess`
  - `organizations:RegisterDelegatedAdministrator` (若使用委派管理)
  - `iam:CreateServiceLinkedRole`
- 您必須使用 AWSEC2CapacityManagerServiceRolePolicy 使用案例建立服務連結角色，以允許 AWS Organization 存取。若要了解詳細資訊，請參閱針對容量管理器建立服務連結角色。

使用 AWS Organizations 啟用容量管理員

使用管理帳戶，在容量管理器中啟用組織存取權。

Console

在容量管理器中啟用組織存取權

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇 Settings (設定) 標籤。
4. 在可信存取權區段，選擇管理可信存取權。
5. 在顯示的提示中，選取啟用可信存取權。然後選擇 Save (儲存)。

AWS CLI

在容量管理器中啟用組織存取權

1. 建立服務連結角色

```
aws iam create-service-linked-role --aws-service-name
 ec2.capacitymanager.amazonaws.com
```

2. 啟用 AWS 組織存取

```
aws organizations enable-aws-service-access --service-principal
 ec2.capacitymanager.amazonaws.com
```

3. 使用 AWS Organization 啟用容量管理員

```
aws ec2 enable-capacity-manager --organizations-access
```

如需更新現有容量管理器的組織存取權，請執行以下命令：

```
aws ec2 update-capacity-manager-organizations-access --organizations-access
```

PowerShell

在容量管理器中啟用組織存取權

1. 使用 New-IAMServiceLinkedRole cmdlet 建立服務連結角色。

```
New-IAMServiceLinkedRole -AWSServiceName "ec2.capacitymanager.amazonaws.com"
```

2. 使用 Enable-ORGAWSServiceAccess cmdlet 啟用 AWS 組織存取。

```
Enable-ORGAWSServiceAccess -ServicePrincipal "ec2.capacitymanager.amazonaws.com"
```

3. 使用 Enable-EC2CapacityManager cmdlet 透過 AWS Organization 啟用容量管理員。 Enable-EC2CapacityManager

```
Enable-EC2CapacityManager -OrganizationsAccess $true
```

若要更新現有 Capacity Manager 的組織存取權，請執行下列 Update-EC2CapacityManagerOrganizationsAccess cmdlet：

```
Update-EC2CapacityManagerOrganizationsAccess -OrganizationsAccess $true
```

確認您的組織是否啟用容量管理器

Console

如需確認您的組織是否啟用容量管理器

1. 在容量管理器主控台中，選擇設定。
2. 在可信存取權區段，確認組織存取權顯示為已啟用。
3. 檢查組織 ID 是否顯示您組織的 ID。

AWS CLI

如需確認您的組織是否啟用容量管理器

執行以下命令：

```
aws ec2 get-capacity-manager-attributes
```

輸出應顯示：

```
{
    "CapacityManagerStatus": "enabled",
    "OrganizationsAccess": true,
    "IngestionStatus": "initial-ingestion-pending",
    "IngestionStatusMessage": "Capacity Manager is collecting historical data from
 2025-10-01T00:00:00Z. Data collection is in progress and may take several hours to
 complete."
}
```

PowerShell

如需確認您的組織是否啟用容量管理器

使用 Get-EC2CapacityManagerAttribute cmdlet。

```
Get-EC2CapacityManagerAttribute
```

輸出應顯示：

```
CapacityManagerStatus      : enabled
DataExportCount            : 0
EarliestDatapointTimestamp :
IngestionStatus            : initial-ingestion-in-progress
IngestionStatusMessage     : Capacity Manager is collecting historical data from
                               2026-03-17T16:00:00Z. Data collection is in progress
 and may take
                               several hours to complete.
LatestDatapointTimestamp   :
OrganizationsAccess        : True
```

考量事項

- 服務連結角色建立：若透過主控台啟用組織存取權，容量管理器可在所有會員帳戶中自動建立 AWSServiceRoleForEC2CapacityManager 服務連結角色。如果您透過 啟用 AWS CLI，則必須`createServiceLinkedRole`手動呼叫 。
- 資料彙總：啟用組織存取之後，容量管理員會回填所有成員帳戶 14 天的歷史資料。完成此程序通常需要幾分鐘時間。
- 區域限制：每個組織只能在一個區域中啟用容量管理員，但它將從所有商業區域彙總資料。
- 許可：會員帳戶無須執行任何動作。容量管理器可使用服務連結角色，自動探索跨所有帳戶的資源。

在帳戶層級啟用 EC2 容量管理器

在帳戶層級啟用容量管理員，以監控和分析單一 AWS 帳戶中的 EC2 容量使用量。啟用容量管理器之後，容量管理器可收集您的隨需執行個體、Spot 執行個體與容量保留的相關資料，進而協助您確定最佳化機會及追蹤用量模式。

在帳戶層級啟用容量管理器

Console

如需為您的帳戶啟用容量管理器

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 在容量管理器頁面，選擇在區域中啟用。

AWS CLI

如需為您的帳戶啟用容量管理器

執行以下命令：

```
aws ec2 enable-capacity-manager
```

PowerShell

如需為您的帳戶啟用容量管理器

使用 Enable-EC2CapacityManager cmdlet。

```
Enable-EC2CapacityManager
```

> **Note**
>
> - 啟用容量管理器之後，容量管理器可收集與彙整 14 天的歷史資料。此程序可能需要幾小時的時間。
> - 收集歷史資料時，會顯示 `initial-ingestion-in-progress` 狀態。在收集期間，您可能會觀察到歷史資料存在缺口。資料收集完成時，會顯示 `ingestion-complete` 狀態。

註冊委派管理員

您可為容量管理器註冊委派管理員。這可讓成員帳戶管理您 AWS 組織的容量管理員。只有管理帳戶才可在您的組織內註冊或移除委派管理員。

> **Note**
>
> 當有已註冊的委派管理員時，您無法停用組織的容量管理員。

主題

- 先決條件
- 註冊委派管理員
- 移除委派管理員

先決條件

您的管理帳戶必須已啟用 Capacity Manager with AWS Organizations。如需詳細資訊，請參閱使用 AWS Organizations 啟用 EC2 Capacity Manager。

註冊委派管理員

您可以使用 Amazon EC2 主控台 AWS CLI、 或 PowerShell 註冊委派管理員。

Console

註冊 委派管理員

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇 Settings (設定) 標籤。
4. 在委派管理員區段，選擇新增。
5. 在顯示的提示中，輸入您想要新增為委派管理員的組織會員的帳戶 ID。
6. 選擇 Add delegated administrator (新增委派管理員)。

AWS CLI

註冊 委派管理員

執行以下命令：

```
aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal ec2.capacitymanager.amazonaws.com
```

PowerShell

註冊 委派管理員

使用 Register-ORGDelegatedAdministrator cmdlet。

```
Register-ORGDelegatedAdministrator `
    -AccountId "123456789012" `
    -ServicePrincipal "ec2.capacitymanager.amazonaws.com"
```

移除委派管理員

您可以使用 Amazon EC2 主控台 AWS CLI、 或 PowerShell 移除委派管理員。

Console

如需移除委派管理員

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇 Settings (設定) 標籤。
4. 在委派管理員區段，選擇管理。
5. 在顯示的提示中，選擇移除委派管理員。

AWS CLI

如需移除委派管理員

執行以下命令：

```
aws organizations deregister-delegated-administrator \
```

```
    --account-id 123456789012 \
    --service-principal ec2.capacitymanager.amazonaws.com
```

PowerShell

如需移除委派管理員

使用 Unregister-ORGDelegatedAdministrator cmdlet。

```
Unregister-ORGDelegatedAdministrator `
    -AccountId "123456789012" `
    -ServicePrincipal "ec2.capacitymanager.amazonaws.com"
```

使用 EC2 容量管理器的服務連結角色

EC2 Capacity Manager 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至容量管理器的特殊 IAM 角色類型。服務連結角色由 Capacity Manager 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定容量管理員，因為您不必手動新增必要的許可。容量管理器定義其服務連結角色的許可，除非另有定義，否則僅有容量管理器可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。如此可保護您的容量管理器資源，避免您不小心移除資源的存取許可。

如需有關支援服務連結角色的其他 服務的資訊，請參閱服務連結角色欄中與 AWS IAM 搭配使用的 服務，並尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

容量管理器的服務連結角色許可

Capacity Manager 使用名為 AWSServiceRoleForEC2CapacityManager 的服務連結角色，可讓您代表您管理容量資源並與 AWS Organizations 整合。

AWSServiceRoleForEC2CapacityManager 服務連結角色信任下列服務擔任該角色：

- `ec2.capacitymanager.amazonaws.com`

名稱為 AWSEC2CapacityManagerServiceRolePolicy 的角色許可政策允許容量管理器完成下列動作：

- organizations:DescribeOrganization
- organizations:ListAccounts
- organizations:ListChildren
- organizations:ListAWSServiceAccessForOrganization
- organizations:ListDelegatedAdministrators

您必須設定許可，以允許您的使用者、群組或角色建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

為容量管理器建立服務連結角色

您可使用 IAM 主控台透過 AWSEC2CapacityManagerServiceRolePolicy 使用案例建立服務連結角色。在 AWS CLI 或 AWS API 中，使用服務名稱建立ec2.capacitymanager.amazonaws.com服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的「建立服務連結角色」。如果您刪除此服務連結角色，您可以使用此相同的程序以再次建立該角色。

為容量管理器編輯服務連結角色

容量管理器不允許您編輯 AWSServiceRoleForEC2CapacityManager 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的編輯服務連結角色。

為容量管理器刪除服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

> **Note**
>
> 若容量管理器服務在您試圖刪除資源時正在使用該角色，刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

如需移除 AWSServiceRoleForEC2CapacityManager 使用的容量管理器資源

1. 移除組織存取權之前，所有委派管理員必須先停用其容量管理器。
2. 停用容量管理器之前，必須先刪除任何作用中資料匯出項。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForEC2CapacityManager 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

容量管理器服務連結角色的支援區域

容量管理器支援在所有提供該服務的區域中使用服務連結角色。如需詳細資訊，請參閱 AWS Regions and endpoints。

容量管理器不支援在提供該服務的每個區域中使用服務連結角色。您可在下列區域使用 AWSServiceRoleForEC2CapacityManager 角色。

區域名稱	區域身分	容量管理器中的支援
美國東部 (維吉尼亞北部)	us-east-1	是
美國東部 (俄亥俄)	us-east-2	是
美國西部 (加利佛尼亞北部)	us-west-1	是
美國西部 (奧勒岡)	us-west-2	是
Africa (Cape Town)	af-south-1	否
亞太地區 (香港)	ap-east-1	否
亞太地區 (雅加達)	ap-southeast-3	否
亞太區域 (孟買)	ap-south-1	是
亞太區域 (大阪)	ap-northeast-3	是
亞太區域 (首爾)	ap-northeast-2	是
亞太區域 (新加坡)	ap-southeast-1	是
亞太區域 (雪梨)	ap-southeast-2	是
亞太區域 (東京)	ap-northeast-1	是
加拿大 (中部)	ca-central-1	是

區域名稱	區域身分	容量管理器中的支援
歐洲 (法蘭克福)	eu-central-1	是
歐洲 (愛爾蘭)	eu-west-1	是
歐洲 (倫敦)	eu-west-2	是
歐洲 (米蘭)	eu-south-1	否
Europe (Paris)	eu-west-3	是
Europe (Stockholm)	eu-north-1	是
Middle East (Bahrain)	me-south-1	否
中東 (阿拉伯聯合大公國)	me-central-1	否
南美洲 (聖保羅)	sa-east-1	是
AWS GovCloud (US-East)	us-gov-east-1	否
AWS GovCloud （美國西部）	us-gov-west-1	否

在容量管理器中整理資料

容量管理器可使用指標、資料點、維度、日期範圍與期間的組合來整理容量資料。這有助您分析用量模式，以及做出資源方面的明智決策。

指標與資料點

指標是按時間順序排列的資料集點。舉例來說，若您想要監控 vCPU 中的 Spot 用量，可使用 `SpotTotalUsageHrsVcpu` 指標。

指標每小時會生成時間戳記資料點，包括 Spot 用量 (vCPU 小時)。舉例來說，若您在上午 10:00 的時段內使用 100 vCPU，容量管理器則會建立時間戳記為上午 10:00 且值為 100 的資料點。

若要了解容量管理器分析的完整指標清單，請參閱 EC2 容量管理器指標。

Dimensions (尺寸)

維度是名稱值對，有助您確定指標的不同層面及對其分類。舉例來說，容量管理器中一個維度的名稱為 AccountID，其中值為實際帳戶 ID。容量管理器可提供各種維度來區隔資料並對其分組，例如，執行個體系列、保留 ARN、保留類型，以及租用。

除了內建維度，您還可以從 Amazon EC2 資源啟用標籤索引鍵，以用作自訂維度。例如，如果您的執行個體使用 `environment`或 標記`team`，您可以啟用這些標籤索引鍵，然後依其值分組和篩選容量指標。

Capacity Manager 也提供 Capacity Manager 提供的標籤，例如 EC2 Auto Scaling 群組 (`aws:autoscaling:groupName`)、EKS 叢集名稱 (`aws:eks:cluster-name`)、EKS Kubernetes 節點集區 (`eks:kubernetes-node-pool-name`) 和 Karpenter 節點集區 (`karpenter.sh/nodepool`)，這些標籤預設為可用，而不會計入您的標籤金鑰限制。

若要了解完整的維度清單，請參閱 EC2 容量管理器指標。如需啟用和管理標籤維度的資訊，請參閱 管理受監控的標籤金鑰。

日期範圍與期間

日期範圍指定您想要分析的時長，從一小時到 90 天。此期間確定了容量管理器在一段時間彙整資料的方式及要傳回的資料點數目。舉例來說，若日期範圍為一週，期間為 1 天，容量管理器則會傳回 7 個資料點。每個資料點代表一天的彙整資料。期間必須為一小時的間隔時間，且平均劃分為日期範圍。

主題

- EC2 容量管理器指標
- 分組及篩選資料
- 管理受監控的標籤金鑰

EC2 容量管理器指標

容量管理器提供全面的指標選擇，用於追蹤不同資源類型的容量。您可依據分析需求，以各種不同的單位來測量指標。

容量管理器中的指標名稱採用四種不同的字首，對待測容量類型進行分類：

- `Reservation` – 容量保留本身，包括預留容量總計、使用率、未使用容量，以及保留計數。
- `Reserved` – 容量保留涵蓋的隨需執行個體用量。
- `Unreserved` – 任何容量保留之外執行的隨需執行個體用量。

- Spot – 尤其針對 Spot 執行個體用量，包括執行時期與估計費用。這些指標與保留型容量分開。

下表亦提供了每項指標的可用維度。維度類別細分如下：

- 一般容量維度 — 帳戶 ID、帳戶名稱、區域、執行個體系列、可用區域、執行個體類型、平台和租用
- 容量保留維度 – 保留 ID (容量保留 ID)、保留 ARN、未使用財務擁有者、保留類型 (ODCR/容量區塊)、建立時間戳記、開始時間戳記、結束時間戳記、狀態，以及執行個體比對條件。
- 預留用量維度 – 保留 ID (CRID)、保留 ARN、保留類型
- 標籤維度 — 客戶管理的標籤索引鍵和 Capacity Manager 提供的標籤。您必須先啟用標籤金鑰，才能將其用作維度。如需詳細資訊，請參閱管理受監控的標籤金鑰。

指標	說明	可用維度	可用單位
ReservationAvgCommittedSize	合約期間處於作用中或排程狀態的平均總容量。可跨維度對大小加總，以及在一段時間後計算平均值。	一般容量與容量保留維度	vCPU、執行個體
ReservationAvgFutureSize	在選取期間，排程為未來開始，但尚未在作用中的容量保留平均量。可跨維度對大小加總，以及在一段時間後計算平均值。	一般容量與容量保留維度	vCPU、執行個體
ReservationAvgUtilization	在選取期間，使用的預留容量平均百分比。	一般容量與容量保留維度	vCPU、執行個體
ReservationMaxCommittedSize	合約期間處於作用中或排程狀態的總容量上限。可跨維度對大小加總，以及計算該期間的最大值。	一般容量與容量保留維度	vCPU、執行個體
ReservationMaxFutureSize	在選取期間，排程為未來開始，但尚未在作用中的容量保留量上限。可跨維度對大小加總，以及計算該期間的最大值。	一般容量與容量保留維度	vCPU、執行個體

指標	說明	可用維度	可用單位
ReservationMaxSize	在選取期間，任何時間點達到容量保留的大小上限。保留 ID 為必要項目。	一般容量與容量保留維度	vCPU、執行個體
ReservationMaxUnusedSize	在選取期間，任何時間點容量保留中未使用容量的大小上限。保留 ID 為必要項目。	一般容量與容量保留維度	vCPU、執行個體
ReservationMaxUtilization	在選取期間，任何時間點容量保留的使用率百分比上限。保留 ID 為必要項目。	一般容量與容量保留維度	
ReservationMinCommittedSize	合約期間處於作用中或排程狀態的總容量下限。可跨維度對大小加總，以及計算該期間的最小值。	一般容量與容量保留維度	vCPU、執行個體
ReservationMinFutureSize	在選取期間，排程為未來開始，但尚未在作用中的容量保留量下限。可跨維度對大小加總，以及計算該期間的最小值。	一般容量與容量保留維度	vCPU、執行個體
ReservationMinSize	在選取期間，任何時間點達到容量保留的大小下限。保留 ID 為必要項目。	一般容量與容量保留維度	vCPU、執行個體
ReservationMinUnusedSize	在選取期間，任何時間點容量保留中未使用容量的大小下限。保留 ID 為必要項目。	一般容量與容量保留維度	vCPU、執行個體
ReservationMinUtilization	在選取期間，任何時間點容量保留的使用率百分比下限。保留 ID 為必要項目。	一般容量與容量保留維度	

指標	說明	可用維度	可用單位
ReservationTotalCapacityHrs	在選取期間，透過容量保留來預留的總容量。	一般容量與容量保留維度	vCPU、執行個體
ReservationTotalCount	在選取期間，您擁有的不同容量保留總數。	一般容量與容量保留維度	
ReservationTotalEstimatedCost	在選取期間，預留總容量時數的估計費用。	一般容量與容量保留維度	
ReservationUnusedTotalCapacityHrs	在選取期間，您擁有但未使用的預留容量總量。	一般容量與容量保留維度	vCPU、執行個體
ReservationUnusedTotalEstimatedCost	在選取期間，您擁有但未使用的預留容量估計費用 (使用隨需費率計算)。	一般容量與容量保留維度	
ReservedTotalEstimatedCost	在選取期間，容量保留涵蓋的隨需執行個體用量估計費用。這不包括 Spot 用量。	一般容量與預留用量維度	
ReservedTotalUsageHrs	在選取期間，容量保留涵蓋的隨需執行個體用量總時數。這不包括 Spot 用量。	一般容量與預留用量維度	vCPU、執行個體
SpotAvgRunTimeBeforeInterruption	在所選期間內中斷之執行個體的平均執行時間，以小時為單位。	僅限區域、可用區域與帳戶 ID 維度	執行個體
SpotInterruptionRate	在所選期間內中斷的執行中 Spot 執行個體百分比。	僅限區域、可用區域與帳戶 ID 維度	vCPU、執行個體

指標	說明	可用維度	可用單位
SpotMaxRunTimeBeforeInterruption	在所選期間內中斷之執行個體的最大執行時間，以小時為單位。	僅限區域、可用區域與帳戶 ID 維度	執行個體
SpotMinRunTimeBeforeInterruption	所選期間內中斷之執行個體的最短執行時間，以小時為單位。	僅限區域、可用區域與帳戶 ID 維度	執行個體
SpotTotalCount	在所選期間內執行的 Spot 執行個體或 vCPUs數量。	僅限區域、可用區域與帳戶 ID 維度	vCPU、執行個體
SpotTotalEstimatedCost	在選取期間，Spot 執行個體用量估計費用 (使用發布的 Spot 費率計算)。	一般容量維度	
SpotTotalInterruptions	在所選期間內中斷的 Spot 執行個體或 vCPUs數量。	僅限區域、可用區域與帳戶 ID 維度	vCPU、執行個體
SpotTotalUsageHrs	在選取期間，Spot 執行個體用量總時數。	一般容量維度	vCPU、執行個體
UnreservedTotalEstimatedCost	在選取期間，容量保留未涵蓋的隨需執行個體用量估計費用。這不包括 Spot 用量。	一般容量維度	
UnreservedTotalUsageHrs	在選取期間，容量保留未涵蓋的隨需執行個體用量總時數。這不包括 Spot 用量。	一般容量維度	vCPU、執行個體

> **Note**
>
> 若單位中包括執行個體，建議您在維度中亦包括執行個體類型。

標籤維度

除了上一節的內建維度之外，容量管理員還支援標籤維度。標籤維度可讓您使用來自 Amazon EC2 資源的標籤索引鍵來分組和篩選指標。

客戶受管標籤維度

您最多可以啟用五個標籤索引鍵，以用作維度。啟用並處於 `activated` 狀態後，標籤維度可用於支援一般容量維度的指標。如需維度類別的完整清單，請參閱 EC2 容量管理器指標。

維度	說明	範例值
標籤索引鍵名稱	來自 Amazon EC2 資源的客戶受管標籤金鑰。	`environment` ，`team`, `cost-center`

Capacity Manager 提供的標籤維度

根據預設，容量管理員會提供下列容量管理員提供的標籤。Capacity Manager 提供的標籤一律可用，且不計入您的標籤金鑰限制。

維度	說明
`tag:aws:autoscaling:groupName`	與執行個體相關聯的 EC2 Auto Scaling 群組名稱。
`tag:aws:eks:cluster-name`	與執行個體相關聯的 EKS 叢集名稱。
`tag:eks:kubernetes-node-pool-name`	與執行個體相關聯的 EKS Kubernetes 節點集區。
`tag:karpenter.sh/nodepool`	與執行個體相關聯的 Karpenter 節點集區。

> **Note**
>
> 當您依標籤維度分組時，沒有該標籤值的資源會包含在具有空字串值的個別儲存貯體中。這可確保所有 資源的總計帳戶。

分組及篩選資料

容量管理器可依據您選擇的維度與日期期間來彙整您的指標。若未選擇維度，容量管理器將會彙整資料，並且在日期範圍的每個期間傳回一個資料點。您可以透過新增維度將資料分組為較小的彙總，包括來自 Amazon EC2 資源的標籤維度。

- 分組 — 依區域、執行個體系列、帳戶 ID 或標籤索引鍵等維度細分容量資料。您可依多個維度來對指標分組，以便進一步細分資料。舉例來說，若依區域及可用區域分組，可獲得每個產生用量的區域及可用區域組合的資料點。
- 篩選 – 僅顯示您選取維度的特定子集。舉例來說，若依執行個體系列分組，可獲得產生用量的所有系列的資料點。然而，若同時依 p5 篩選，可僅查看 p5 執行個體系列的資料點。您也可以依標籤值篩選 - 例如，依 篩選`environment=prod`，只查看生產資源。若要篩選沒有所選標籤值的資源，請依空值篩選。
- 指標單位 – 依 vCPU、執行個體或估計費用等不同單位來檢視結果。例如，在依區域分組和依特定執行個體系列篩選之後，您可以在使用的總 vCPUs、執行中的執行個體數量或預估成本之間切換。

如何在主控台對資料分組及篩選資料

如需在容量管理器中對資料分組及篩選資料

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇您想要分析的資源類型索引標籤：用量、保留或 Spot。
4. 在日期篩選條件區段，選擇日期範圍、時區，以及時間間隔。
5. 在維度篩選條件區段，從選取維度下拉式選單中選擇維度。每個資源類型的維度選項都不同。如需詳細資訊，請參閱EC2 容量管理器指標。

   若要依標籤維度分組，請從下拉式清單中選擇標籤索引鍵 （ 例如，`environment`)。只有處於 `activated` 狀態的標籤索引鍵可用。如需啟用標籤索引鍵的詳細資訊，請參閱 管理受監控的標籤金鑰。

   您可以新增多個維度，包括多個標籤維度，以建立更精細的分組。
6. 如需依您選取的維度篩選，從依維度篩選下拉式選單中，選擇一個篩選條件選項。

   對於標籤維度，篩選條件會顯示資料中存在的標籤值。若要篩選沒有所選標籤值的資源，請選擇空白值選項。
7. 在彙整區段，選擇依 vCPU、執行個體，或者估計費用來檢視結果的單位。

> **Note**
>
> 當您依標籤維度分組時，沒有該標籤值的資源會包含在具有空值的單獨群組中。這可確保 總計考慮所有資源，而不只是標記的資源。

管理受監控的標籤金鑰

Amazon EC2 Capacity Manager 可讓您從 Amazon EC2 資源 （ 例如 `environment`或 `team`) 中選取標籤金鑰，以在分析容量資料時用作維度。啟用標籤金鑰後，您可以依該標籤的值來分組和篩選指標，就像使用區域、執行個體類型或可用區域一樣。

每個帳戶最多可監控五個標籤金鑰。Capacity Manager 提供的標籤不會計入此限制。

主題

- Capacity Manager 提供的標籤
- 標籤生命週期
- 啟用和停用受監控的標籤金鑰
- 檢視受監控的標籤索引鍵
- 使用標籤維度查詢指標
- 資料匯出中的標籤
- 組織和委派管理員
- 考量事項

Capacity Manager 提供的標籤

Capacity Manager 包含每個帳戶一組預設標籤。這些不會計入您的標籤金鑰限制。Capacity Manager 提供的標籤代表常用的分組維度，並包含：

- `aws:autoscaling:groupName` — EC2 Auto Scaling 群組
- `aws:eks:cluster-name` — EKS 叢集名稱
- `eks:kubernetes-node-pool-name` — EKS Kubernetes 節點集區
- `karpenter.sh/nodepool` — Karpenter 節點集區

Capacity Manager 提供的標籤會在 中顯示`GetCapacityManagerMonitoredTagKeys`，並將 `CapacityManagerProvided` 設定為 `true`，且無法由客戶啟用或停用。第一次啟用容量管理員時，

容量管理員提供的標籤會開始處於 activating 狀態，並在容量管理員收到包含容量管理員提供的標籤的第一個資料點activated之後轉換為 ，通常在一到兩個小時內。

標籤生命週期

受監控的標籤金鑰會繼續進行下列狀態：

狀態	說明
activating	標籤金鑰已註冊。Capacity Manager 正在準備收集此標籤的資料。您無法使用此狀態的標籤查詢指標。
activated	正在擷取標籤資料，並可透過指標 APIs 和資料匯出進行查詢。
suspended	標籤索引鍵已超過 100，000 個唯一標籤值的閾值。標籤仍會計入您的限制，但容量管理員不會再為其擷取資料。如果您的標籤值用量低於持續期間的閾值，則標籤會自動重新啟用。
deactivating	正在移除標籤金鑰。GetCapacityManagerMonitoredTagKeys 停用完成後，它不會再出現在 中。

當標籤處於 suspended 狀態時， GetCapacityManagerMonitoredTagKeys會傳回下列狀態訊息：「標籤因標籤值太多而暫停。減少標籤用量或停用標籤。」

> **Note**
>
> 如果您停用標籤金鑰，然後重新啟用相同的金鑰，則只有重新啟用後擷取的資料才能查詢。無法存取先前啟用的歷史資料。會在每次新的啟用時EarliestDatapointTimestamp重設 。

啟用和停用受監控的標籤金鑰

您可以啟用標籤索引鍵，開始將它們監控為維度，或停用不再需要的標籤索引鍵。啟用是非同步的 — 標籤進入 activating 狀態，並在容量管理員開始接收該標籤的資料activated之後轉換為 。停用會從受監控的集合中移除標籤金鑰。

> **Note**
>
> 啟用標籤金鑰時，請僅輸入金鑰名稱（例如，`environment`)。Capacity Manager 會自動將其用作分組和篩選指標的維度。

Console

啟用或停用受監控的標籤金鑰

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇 Settings (設定) 標籤。
4. 在監控標籤金鑰區段中，選擇管理標籤金鑰。
5. 若要啟用標籤金鑰，請輸入標籤金鑰名稱，然後選擇新增。若要停用標籤金鑰，請選取標籤金鑰，然後選擇移除。
6. 選擇儲存變更。

AWS CLI

啟用標籤索引鍵

使用下列命令來啟用一或多個標籤索引鍵：

```
aws ec2 update-capacity-manager-monitored-tag-keys \
    --activate-tag-keys "environment" "teamId"
```

輸出會顯示標籤索引鍵及其初始狀態：

```
{
    "CapacityManagerTagKeys": [
        {
            "TagKey": "environment",
            "Status": "activating"
        },
        {
```

```
            "TagKey": "teamId",
            "Status": "activating"
        }
    ]
}
```

停用標籤索引鍵

使用以下命令停用一或多個標籤索引鍵：

```
aws ec2 update-capacity-manager-monitored-tag-keys \
    --deactivate-tag-keys "project"
```

輸出會顯示更新的狀態：

```
{
    "CapacityManagerTagKeys": [
        {
            "TagKey": "project",
            "Status": "deactivating"
        }
    ]
}
```

您可以在相同的請求中啟用和停用標籤金鑰：

```
aws ec2 update-capacity-manager-monitored-tag-keys \
    --activate-tag-keys "environment" "teamId" \
    --deactivate-tag-keys "project"
```

檢視受監控的標籤索引鍵

您可以檢視您帳戶的所有受監控標籤金鑰，包括其目前狀態和可用的資料的最早時間戳記。

Console

檢視受監控的標籤索引鍵

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇 Settings (設定) 標籤。
4. 在監控標籤金鑰區段中，檢視您的標籤金鑰、其狀態，以及它們是否為 Capacity Manager 提供的標籤。

AWS CLI

檢視受監控的標籤索引鍵

執行以下命令：

```
aws ec2 get-capacity-manager-monitored-tag-keys
```

輸出會列出所有標籤索引鍵，包括 Capacity Manager 提供的標籤和客戶管理的標籤：

```
{
    "CapacityManagerTagKeys": [
        {
            "TagKey": "aws:autoscaling:groupName",
            "Status": "activated",
            "CapacityManagerProvided": true,
            "earliestDatapointTimestamp": "2026-04-08T00:00:00"
        },
        {
            "TagKey": "environment",
            "Status": "activated",
            "CapacityManagerProvided": false,
            "earliestDatapointTimestamp": "2025-08-11T22:00:00"
        }
    ]
}
```

CapacityManagerProvided 欄位指出標籤是容量管理員提供的標籤 (true) 還是客戶管理的標籤 ()false。Capacity Manager 提供的標籤不會計入您的標籤金鑰限制。EarliestDatapointTimestamp 指出可供該標籤索引鍵使用資料的最早時間點。

使用標籤維度查詢指標

標籤金鑰達到activated狀態後，您可以在 GetCapacityManagerMetricDimensions和 中將其用作維度GetCapacityManagerMetricData。

查詢依標籤維度分組的指標

使用下列命令：

```
aws ec2 get-capacity-manager-metric-dimensions \
    --group-by tag:environment account-id \
    --filter-by
 'DimensionCondition={Dimension=tag:environment,Comparison=equals,Values=[prod]}'
```

當您依標籤維度分組時，結果會包含您帳戶中的所有資源，而不只是具有標籤的資源。沒有標籤值的資源會分組為具有空字串值的個別儲存貯體。例如，如果您的帳戶在指定期間內使用 800 個 vCPU 時數，而且只有其中一些資源具有 environment標籤，則依environment標籤索引鍵分組可能會傳回：

- prod — 300 個 vCPU 時數
- staging — 200 個 vCPU 時數
- ""（空字串）— 300 個 vCPU 小時來自沒有 environment標籤的資源

這可確保所有儲存貯體的總計計入您的完整用量。您可以透過傳遞空字串做為篩選條件值，明確篩選未標記的資源：

```
--filter-by
 'DimensionCondition={Dimension=tag:environment,Comparison=equals,Values=[""]}'
```

> **Note**
>
> 如果您查詢的標籤索引鍵仍處於 `activating` 狀態，則會以 400 錯誤拒絕查詢。等待標籤的狀態變更為 `activated`再查詢。您可以使用 檢查狀態`GetCapacityManagerMonitoredTagKeys`。

> **Note**
>
> 在任何提供的標籤維度的 之前，具有開始時間`EarliestDatapointTimestamp`的查詢會遭到拒絕。使用 `GetCapacityManagerMonitoredTagKeys`檢查資料何時可供每個標籤使用。

資料匯出中的標籤

當您啟用標籤監控時，資料匯出會包含已啟用的標籤金鑰和容量管理員提供的標籤，做為額外的資料欄。標籤欄會出現在具有 `tag:environment`和 等標頭的所有標準欄之後`tag:team`。標籤欄會依字母順序排序。

匯出僅包含`activated`處於 狀態的標籤。Capacity Manager 會排除 `activating`、`deactivating`或 `suspended` 狀態的標籤。

> **Note**
>
> 如果您啟用新的標籤金鑰，現有的資料匯出不會自動包含新的標籤。您必須建立新的資料匯出，以包含新啟用的標籤金鑰做為資料欄。

組織和委派管理員

當您的帳戶是啟用 AWS 組織層級容量管理員的組織的一部分時，每個帳戶 （組織管理員和委派管理員） 都可以獨立啟用、停用和查詢標籤金鑰。每個帳戶都會維護自己的標籤狀態、`EarliestDatapointTimestamp`和標籤索引鍵限制。

帳戶只能查詢帳戶本身已啟用之標籤索引鍵的指標資料。如果組織管理員和委派管理員都啟用相同的標籤金鑰 （例如 `environment`)，每個帳戶都會獨立追蹤自己的啟用狀態和資料可用性。

當委派管理員停用標籤金鑰時，委派管理員就無法再查詢該標籤的資料，即使組織管理員仍然啟用相同的標籤金鑰。

考量事項

- 標籤值更新：新資源的標籤值和新套用至現有資源的標籤可在幾個小時內使用。如果您變更資源上現有標籤的值，更新後的值最多可能需要 24 小時才能反映在容量管理員中。
- 啟用時間：啟用標籤金鑰後，最多可能需要 24 小時才能將標籤轉換為`activated`狀態，且資料可查詢。`EarliestDatapointTimestamp` 代表資料何時可用，而不是在標籤啟用時。
- 標籤金鑰限制：每個帳戶最多可監控五個標籤金鑰。Capacity Manager 提供的標籤不會計入此限制。
- 標籤索引鍵字元需求：標籤索引鍵可包含 Unicode 字母、數字、空格和下列字元：`_ . : / = + @ -`。標籤索引鍵不得超過 128 個字元。
- 重新啟用：如果您停用並重新啟用相同的標籤金鑰，則只有新資料可用。會在每次啟用時`EarliestDatapointTimestamp`重設 。
- 暫停的標籤：每個標籤索引鍵最多支援 100，000 個唯一標籤值。如果標籤索引鍵超過此閾值，則會移至 `suspended` 狀態。標籤仍會計入您的限制，但不再擷取資料。減少標籤的唯一值數量，或將其停用以釋放另一個標籤索引鍵的空間。

在 AWS 主控台中導覽容量管理員

在容量管理器主控台中，整理了可提供不同容量資料檢視的索引標籤：

- 儀表板 — 提供所有隨需容量保留、隨需和 Spot 用量的高階概觀，包括有助於改善容量狀態的關鍵指標。
- 用量 – 提供隨需執行個體與 Spot 執行個體的執行個體用量模式概觀。依容量保留分析涵蓋範圍，以及透過靈活的分組及篩選來確定最佳化機會。
- 保留 – 提供跨帳戶與區域的容量保留使用率、管理功能，以及詳細保留指標等分析。
- Spot – 監控 Spot 用量模式，以及針對跨帳戶與區域的 Spot 執行個體提供費用分析。
- 資料匯出 – 管理 Amazon S3 的資料匯出組態，包括排程、格式化，以及範本選取。
- 設定 — 提供服務組態選項，包括組織存取、區域設定和受監控的標籤金鑰管理。您可以從 Amazon EC2 資源啟用標籤金鑰，以在分組和篩選容量指標時用作維度。如需詳細資訊，請參閱管理受監控的標籤金鑰。

在「用量與保留」索引標籤內，容量管理器提供了階層式導覽結構，讓您能夠透過高階概觀，深入探究詳細的資源資訊。了解此導覽模式，有助您高效分析容量資料，以及確定最佳化機會。

主題

- 導覽階層
- 檢視用量與保留明細及詳細資訊

導覽階層

容量管理器使用三層導覽結構來呈現用量與保留資訊：

1. 概觀頁面 – 包含彙整指標的高階摘要。
2. 明細頁面 – 包含篩選及分組選項的詳細分析
3. 詳細資訊頁面：
   - 用量詳細資訊 – 選取維度組合的相關資訊，提供了統計資料與時間序列資料，有助您了解用量模式。
   - 保留詳細資訊 – 特定容量保留的相關資訊，包括使用率統計資料、一段時間的用量模式，以及組態詳細資訊。

> **Note**
>
> Spot 遵循簡化的結構，僅包含概觀頁面。

檢視用量與保留明細及詳細資訊

「用量與保留」索引標籤遵循相同的三層導覽結構，您可從概觀頁面進一步查看明細頁面及詳細資訊頁面。存取明細頁面及詳細資訊頁面的程序相似，導覽連結位於每個索引標籤中的位置僅有細小差別。

如需存取資源細分頁面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇您想要分析的資源類型索引標籤：用量或保留。
4. 在彙總區段，尋找細分連結：
   - 用量：在動作欄，選擇檢視明細。

> **Note**
>
> 視乎螢幕大小或您套用的維度數目，可能需要在頁面中水平捲動來尋找「檢視明細」連結。

- 保留：在保留欄，選擇您想要檢視的保留編號 (編號為連結)。

如需存取詳細資訊頁面

1. 在明細頁面，導覽至相關區段：
   - 用量：不重複的維度組合區段。
   - 保留：保留區段。
2. 在您選擇的資源相關區段，存取詳細資訊頁面。
   - 用量：在動作欄，選擇檢視詳細資訊。

> **Note**
>
> 視乎螢幕大小或您套用的維度數目，可能需要在頁面中水平捲動來尋找「檢視詳細資訊」連結。

   - 保留：在保留 ID 欄，選擇您想要檢視的保留。

匯出容量管理器資料

您可以將容量資料從 EC2 Capacity Manager 匯出至 Amazon S3，以啟用進一步分析、建立自訂報告，或與其他 AWS 服務整合。您能夠以 CSV 或 Parquet 格式來匯出資料。在下面的區段中，您可找到如何匯出容量管理器資料的相關資訊。

> **Note**
>
> 容量管理器僅允許每個 AWS 帳戶一個資料匯出項。

> **Note**
>
> 在極少數情況下，由於資料問題而需要重新驅動資料匯出，新檔案會覆寫該小時的現有檔案。

主題

- 針對容量管理器資料匯出設定 Amazon S3 儲存貯體
- 針對容量管理器資料建立資料匯出

針對容量管理器資料匯出設定 Amazon S3 儲存貯體

若要接收容量管理員資料匯出，您必須在 中擁有 Amazon S3 儲存貯體 AWS 帳戶 ，才能接收和儲存匯出檔案。在容量管理器主控台建立資料匯出時，您可選擇一個擁有的現有 Amazon S3 儲存貯體，或者建立一個新的儲存貯體。

在任一情況中，都必須套用所需的儲存貯體政策，以便允許容量管理器交付匯出檔案。在 Amazon S3 主控台編輯此政策，或者在您建立資料匯出之後變更儲存貯體擁有者，皆會阻止容量管理器交付匯出。

如需建立 Amazon S3 儲存貯體，請參閱 Amazon Simple Storage Service 使用者指南中的建立 S3 儲存貯體。

必須將下面的政策套用至 S3 儲存貯體，以便允許容量管理器交付資料匯出：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.capacitymanager.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:capacity-
manager-data-export/*"
                }
            }
        }
    ]
}
```

藉助此儲存貯體政策，可確保安全地交付容量管理器資料匯出檔案至儲存貯體。具體而言：

- 每次傳送容量管理員資料匯出時， AWS 請先確認儲存貯體是否仍由設定匯出的帳戶所擁有。若儲存貯體擁有權已變更，則不會交付匯出。這樣可協助確保容量管理器資料的安全性。此儲存貯體政策允許 AWS ("Effect": "Allow") 檢查擁有儲存貯體 () 的帳戶"Action": ["s3:ListBucket"]。
- 政策授予容量管理器服務 ("Service": "ec2.capacitymanager.amazonaws.com") 許可，以便寫入匯出檔案 ("Action": "s3:PutObject") 及讀取物件 ("Action": "s3:GetObject")，進而複製資料到您的儲存貯體。

針對容量管理器資料建立資料匯出

如需建立資料匯出，您可在容量管理器主控台或 AWS CLI中使用資料匯出頁面。

先決條件

您必須建立 Amazon Simple Storage Service (Amazon S3) 儲存貯體。您必須確認以下各項：

- 您的 S3 儲存貯體必須與您啟用 Capacity Manager AWS 區域 的位置相同。
- S3 儲存貯體設定有必要的許可政策，以便容量管理器交付檔案。

如需詳細資訊，請參閱針對容量管理器資料匯出設定 Amazon S3 儲存貯體。

程序

您可以使用 AWS 主控台 AWS CLI、 或 PowerShell 匯出您的容量管理員資料。

Console

如需建立資料匯出

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇容量管理器。
3. 選擇資料匯出索引標籤。
4. 選擇建立匯出。
5. 設定匯出屬性、交付位置和標籤（選用）。
6. 選擇建立。

AWS CLI

如需建立資料匯出

使用以下命令來建立包含指定組態的資料匯出：

```
aws ec2 create-capacity-manager-data-export \
    --s3-bucket-name my-exports-bucket \
    --s3-bucket-prefix capacity-data-exports \
    --schedule hourly \
    --output-format parquet/CSV \
    --tag-specifications 'ResourceType=capacity-manager-data-
export,Tags=[{Key=environment,Value=production}]'
```

> **Note**
>
> 上述命令中的 `--tag-specifications` 參數會將資源標籤套用至資料匯出資源本身（例如，用於成本分配或組織）。這些與監控的標籤索引鍵分開，這會決定匯出資料中包含的標籤欄。如需受監控標籤金鑰的詳細資訊，請參閱 管理受監控的標籤金鑰。

PowerShell

如需建立資料匯出

使用 New-EC2CapacityManagerDataExport cmdlet。

```
New-EC2CapacityManagerDataExport `
    -S3BucketName "my-exports-bucket" `
    -S3BucketPrefix "capacity-data-exports" `
    -Schedule "hourly" `
    -OutputFormat "parquet" `
    -TagSpecification @([Amazon.EC2.Model.TagSpecification]@{
        ResourceType = "capacity-manager-data-export"
        Tags = @([Amazon.EC2.Model.Tag]@{
            Key   = "environment"
            Value = "production"
        })
    })
```

資料匯出中的標籤欄

新建立的資料匯出包含標籤值，做為已啟用受監控標籤金鑰和 Capacity Manager 提供之標籤的額外資料欄。如需詳細資訊，請參閱 管理受監控的標籤金鑰 中的 資料匯出中的標籤。

> **Note**
>
> 如果您尚未啟用任何受監控的標籤金鑰，匯出仍會包含容量管理員提供的標籤資料欄（例如 `aws:autoscaling:groupName`、`eks:kubernetes-node-pool-name`、`aws:eks:cluster-name`和 `karpenter.sh/nodepool`）。

管理 EC2 執行個體的裝置驅動程式

裝置驅動程式是與 Amazon EC2 執行個體的虛擬化硬體進行通訊的軟體元件。為防止系統錯誤、效能問題和其他非預期行為，將驅動程式保持在最新狀態至關重要。這在一些驅動程式上更是如此，例如網路、圖形和儲存裝置的驅動程式，因為其表現會根據使用量而對系統效能產生強大影響。新的驅動程式版本可能包含缺陷修復或引入擴充功能，您可能會希望將這些更新運用在目前執行中的執行個體。

網路驅動程式

Linux 發行版本可以在核心中整合網路功能，例如彈性網路介面卡 (ENA) 或 Elastic Fabric Adapter (EFA)。不過，在不同發行版本中實作核心驅動程式功能的時間可能會有所不同。

ENA 和 EFA Linux 核心驅動程式可從 Amazon Drivers GitHub 儲存庫中取得。如需可用驅動程式的詳細資訊和連結，請參閱 GitHub 上的 Amazon 驅動程式。

如需 ENA 驅動程式的相關資訊，請參閱 在 EC2 執行個體上使用 ENA 啟用增強型網路。如需有關 ENA 驅動程式的詳細資訊，請參閱本指南 適用於 Amazon EC2 上的 AI/ML 和 HPC 工作負載的 Elastic Fabric Adapter 一節中的 入門主題。

若要在 Windows 執行個體上安裝或更新網路驅動程式，請參閱下列主題：

- 在 Windows 上安裝 ENA 驅動程式
- 安裝最新的 AWS PV 驅動程式

  如需詳細資訊，請參閱 Windows 執行個體的全虛擬驅動程式。

> **Note**
>
> EFA 在 Windows 執行個體上不受支援。

圖形驅動程式

若要安裝或更新圖形驅動程式，請參閱下列主題：

- 適用於 EC2 執行個體的 AMD 驅動程式
- Amazon EC2 執行個體的 NVIDIA 驅動程式

儲存裝置驅動程式

若要安裝或更新儲存驅動程式，請參閱下列主題：

- 針對 Linux 執行個體，請參閱「Amazon EBS 使用者指南」中的安裝或升級 NVMe 驅動程式。
- 針對 Windows 執行個體，請參閱 NVMe 驅動程式。

其他 Windows 裝置驅動程式

若要安裝或更新其他 Windows 裝置驅動程式，請參閱下列主題：

- 其他 Windows 裝置驅動程式

適用於 EC2 執行個體的 AMD 驅動程式

已連接 AMD GPU 的執行個體 (例如 G4ad 執行個體) 必須安裝適當的 AMD 驅動程式。根據您的需求，您可以使用預先安裝驅動程式的 AMI，或從 Amazon S3 中下載驅動程式。

若要在已連接 NVIDIA GPU 的執行個體 (例如 G4dn 執行個體) 上安裝 NVIDIA 驅動程式，請參閱 NVIDIA 驅動程式。

內容

- 企業驅動程式的 AMD Radeon Pro 軟體
- 安裝 AMD 驅動程式的 AMI
- AMD 驅動程式下載

企業驅動程式的 AMD Radeon Pro 軟體

企業驅動程式的 AMD Radeon Pro 軟體專為提供專業級圖形使用案例支援而打造。使用驅動程式，您可以為每個 GPU 設定兩個 4K 顯示執行個體。

支援的 API

- OpenGL、OpenCL
- Vulkan
- AMD 進階媒體架構
- 影像加速 API
- DirectX 9 和更新版本
- Microsoft 硬體媒體基礎轉換

安裝 AMD 驅動程式的 AMI

AWS 提供安裝了 AMD 驅動程式的不同 Amazon Machine Image (AMIs)。開啟具有 AMD 驅動程式的 Marketplace 產品。

AMD 驅動程式下載

如果使用的 AMI 沒有安裝 AMD 驅動程式，您可以下載 AMD 驅動程式並將其安裝在您的執行個體上。僅以下作業系統版本支援 AMD 驅動程式：

- 具有核心版本 5.10 的 Amazon Linux 2

- Ubuntu 24.04
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

這些下載僅供 AWS 客戶使用。下載即表示您同意下載的軟體，只會用來開發 AMIs，以搭配 AMD Radeon Pro V520 硬體使用，而不做其他用途。安裝軟體時，您必須遵循 AMD 雲端最終使用者許可協議的條款。

在 Amazon Linux 2 Linux 執行個體上安裝 AMD 驅動程式

1. 連線至您的 Linux 執行個體。
2. 在 Linux 執行個體 AWS CLI 上安裝 並設定預設登入資料。如需詳細資訊，請參閱「AWS Command Line Interface 使用者指南」中的安裝 AWS CLI。

   > **Important**
   >
   > 使用者或角色必須擁有授予的許可，它包含 AmazonS3ReadOnlyAccess 政策。如需詳細資訊，請參閱「Amazon Simple Storage Service 使用者指南」中的 AWS 受管政策：AmazonS3ReadOnlyAccess。

3. 如果尚未安裝 gcc、 make和 dialog，請進行安裝。

   ```
   $ sudo yum install -y gcc make dialog
   ```

4. 更新套件快取，並取得執行個體的套件更新。

   ```
   $ sudo amazon-linux-extras install epel -y
   $ sudo yum update -y
   ```

5. 重新啟動執行個體。

   ```
   $ sudo reboot
   ```

6. 重新啟動後重新連線至執行個體。
7. 下載最新的 AMD 驅動程式。

   ```
   $ aws s3 cp --recursive s3://ec2-amd-linux-drivers/latest/ .
   ```

8. 解壓縮檔案。

```
$ tar -xzf amdgpu-pro-*rhel*.tar.gz
```

9. 安裝 amdgpu-install 套裝服務。 以解壓縮驅動程式時建立的版本目錄名稱取代版本。

```
$ cd rhel_7/version/x86_64/a/
$ sudo yum install -y ./amdgpu-install-*.el7.noarch.rpm
```

10. 編輯 /etc/yum.repos.d/amdgpu.repo. baseurl 設定為擷取的x86_64目錄 （ 例如 baseurl=file:///home/ec2-user/rhel_7/*version*/x86_64/)。設定 gpgcheck=0。
11. 編輯 /etc/yum.repos.d/amdgpu-proprietary.repo. baseurl 設定為相同的目錄。設定 gpgcheck=0和 enabled=1。
12. 編輯 /etc/yum.repos.d/rocm.repo. baseurl 設定為相同的目錄。設定 gpgcheck=0。
13. 安裝適用於 Amazon Linux 2 的 AMD GPU 驅動程式。

```
$ sudo amdgpu-install --usecase=workstation --vulkan=pro --accept-eula -y
```

14. 重新啟動執行個體。

```
$ sudo reboot
```

15. 確認驅動程式可正常運作。

```
$ sudo dmesg | grep amdgpu
```

回應應如下所示：

```
Initialized amdgpu
```

在 Ubuntu Linux 執行個體上安裝 AMD 驅動程式

1. 連線至您的 Linux 執行個體。
2. 更新套件快取，並取得執行個體的套件更新。

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

3. 安裝 Linux 韌體和核心模組。

```
$ sudo apt install -y linux-firmware linux-modules-extra-$(uname -r)
```

4. 重新啟動執行個體。

```
$ sudo reboot
```

5. 重新啟動後重新連線至執行個體。
6. 從 AMD 網站上的適用於 AMD Radeon」圖形的 Linux® Drivers 下載適用於 Ubuntu 24.04 的最新 AMD 驅動程式套件。在下列命令中，將 *##*取代為下載檔案名稱的版本字串 （例如 `31.30.313000-1_all`)。

```
$ sudo apt install -y ./amdgpu-install_version.deb
```

7. 安裝適用於 Ubuntu 的 AMD GPU 驅動程式。

```
$ sudo amdgpu-install --usecase=graphics --vulkan=radv -y
```

8. 重新啟動執行個體。

```
$ sudo reboot
```

9. 確認驅動程式可正常運作。

```
$ sudo dmesg | grep amdgpu
```

回應應如下所示：

```
Initialized amdgpu
```

在您的 Windows 執行個體上安裝 AMD 驅動程式

1. 連接至 Windows 執行個體，然後開啟 PowerShell 視窗。
2. 在 Windows 執行個體 AWS Tools for Windows PowerShell 上設定 的預設登入資料。如需詳細資訊，請參閱 AWS Tools for PowerShell 使用者指南中 AWS Tools for Windows PowerShell入門。

> **Important**
>
> 使用者或角色必須擁有授予的許可，它包含 AmazonS3ReadOnlyAccess 政策。如需詳細資訊，請參閱「Amazon Simple Storage Service 使用者指南」中的 AWS 受管政策：AmazonS3ReadOnlyAccess。

3. 根據您的 Windows 版本設定金鑰字首：

   - Windows 10 及 Windows 11

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS10"
   ```

   - Windows Server 2016

   ```
   $KeyPrefix = "archives"
   ```

   - Windows Server 2019

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K19" # use "archives" for Windows Server
    2016
   ```

   - Windows Server 2022

   ```
   $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K22"
   ```

4. 使用下列 PowerShell 命令從 Amazon S3 下載驅動程式至您的桌面。

```
$Bucket = "ec2-amd-windows-drivers"
$LocalPath = "$home\Desktop\AMD"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
    }
}
```

5. 解壓縮下載的驅動程式檔案，然後使用下列 PowerShell 命令執行安裝程式。

```
Expand-Archive $LocalFilePath -DestinationPath "$home\Desktop\AMD\$KeyPrefix" -
Verbose
```

現在，請檢查新目錄的內容。可以使用 `Get-ChildItem` PowerShell 命令來擷取目錄名稱。

```
Get-ChildItem "$home\Desktop\AMD\$KeyPrefix"
```

輸出格式應類似以下內容：

```
Directory: C:\Users\Administrator\Desktop\AMD\latest

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----       10/13/2021  12:52 AM                210414a-365562C-Retail_End_User.2
```

安裝驅動程式：

```
pnputil /add-driver $home\Desktop\AMD\$KeyPrefix\*.inf /install /subdirs
```

6. 依照指示安裝驅動程式，並視需要將執行個體重新開機。
7. 若要確認 GPU 是否正常運作，請檢查裝置管理員。您應該會看到列為顯示轉接器的「AMD Radeon Pro V520 MxGPU」。
8. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

Amazon EC2 執行個體的 NVIDIA 驅動程式

已連接 NVIDIA GPU 的執行個體 (例如 P- 或 G- 系列執行個體類型) 必須安裝適當的 NVIDIA 驅動程式。根據執行個體類型，您可以下載公有 NVIDIA 驅動程式、從 Amazon S3 下載僅供 AWS 客戶使用的驅動程式，或使用預先安裝驅動程式的 AWS AMI。

若要在已連接 AMD GPU 的執行個體 (例如 G4ad 執行個體) 上安裝 AMD 驅動程式，請參閱 AMD 驅動程式。

主題

- NVIDIA 驅動程式類型
- 依執行個體類型分類的可用驅動程式

- 安裝選項
- 使用包含 NVIDIA 驅動程式的 AMI
- 安裝 NVIDIA 公有驅動程式
- 安裝 NVIDIA GRID 驅動程式 (G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 執行個體)
- 安裝 NVIDIA 遊戲驅動程式 (G7e、G6、G6e、G5 和 G4dn 執行個體)

NVIDIA 驅動程式類型

以下是可與以 GPU 為基礎之執行個體搭配使用的主要 NVIDIA 驅動程式類型。

Tesla 驅動程式

這些驅動程式主要用於運算工作負載，將 GPU 用於平行化浮點計算等運算任務，以進行機器學習，以及用於高效能運算應用程式的快速 Fourier 轉換。

GRID 驅動程式

這些驅動程式經過認證，可為轉譯 3D 模型或高解析度影片等內容的專業視覺化應用程式提供最佳效能。您可以將 GRID 驅動程式設定為支援兩種模式。Quadro 虛擬工作站可提供每個 GPU 四個 4K 顯示器的存取權限。GRID vApp 可提供 RDSH 應用程式託管功能。

遊戲驅動程式

這些驅動程式包含遊戲的最佳化功能，而且經常更新，可提供效能增強功能。它們支援每個 GPU 一個 4K 顯示器。

設定的模式

在 Windows 上，Tesla 驅動程式是設定為在 Tesla Compute Cluster (TCC) 模式下執行。GRID 和遊戲驅動程式是設定為在 Windows Display Driver Model (WDDM) 模式下執行。在 TCC 模式下，卡片可供運算工作負載專用。在 WDDM 模式下，卡片可同時支援運算和圖形工作負載。

NVIDIA 控制面板

GRID 和遊戲驅動程式不支援 NVIDIA 控制面板。Tesla 驅動程式不支援此面板。

Tesla、GRID 和遊戲驅動程式支援的 API

- OpenCL、OpenGL 和 Vulkan
- NVIDIA CUDA 和相關程式庫 (例如 cuDNN、TensorRT、nvJPEG 和 cuBLAS)

- 適用於影片編碼的 NVENC 以及適用於影片解碼的 NVDEC
- 僅限 Windows 的 API：DirectX、Direct2D、DirectX Video Acceleration、DirectX Raytracing

依執行個體類型分類的可用驅動程式

下表摘述每種 GPU 執行個體類型的支援 NVIDIA 驅動程式。

執行個體類型	Tesla 驅動程式	GRID 驅動程式	遊戲驅動程式
G3	是	是	否
G4dn	是	是	是
G5	是	是	是
G5g	是 [1]	否	否
G6	是	是	是
G6e	是	是	是
G6f	否	是	否
Gr6	是	是	否
Gr6f	否	是	否
G7e	是	是	是
P3	是	否	否
P4d	是	否	否
P4de	是	否	否
P5	是	否	否
P5e	是	否	否
P5en	是	否	否

執行個體類型	Tesla 驅動程式	GRID 驅動程式	遊戲驅動程式
P6-B200	是	否	否
P6e-GB200	是	否	否
P6-B300	是	否	否

[1] 此 Tesla 驅動程式也支援 ARM64 平台專屬的最佳化圖形應用程式。

安裝選項

使用下列其中一個選項，取得 GPU 執行個體所需的 NVIDIA 驅動程式。

選項

1. 使用包含 NVIDIA 驅動程式的 AMI
2. 安裝 NVIDIA 公有驅動程式
3. 安裝 NVIDIA GRID 驅動程式 (G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 執行個體)
4. 安裝 NVIDIA 遊戲驅動程式 (G7e、G6、G6e、G5 和 G4dn 執行個體)

使用包含 NVIDIA 驅動程式的 AMI

AWS 和 NVIDIA 提供安裝了 NVIDIA 驅動程式的不同 Amazon Machine Image (AMIs)。

- 具有 Tesla 公有驅動程式的 Marketplace 產品
- 具有 GRID 驅動程式的 Marketplace 產品
- 具有遊戲驅動程式的 Marketplace 產品

若要檢閱取決於作業系統 (OS) 平台的考量事項，請選取適用於 AMI 的標籤。

Linux

若要使用這些 AMI 的其中之一來更新安裝的驅動程式版本，您必須從執行個體中解除安裝 NVIDIA 套件，以避免發生版本衝突。使用此命令解除安裝 NVIDIA 套件：

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

CUDA 工具組套件與 NVIDIA 驅動程式具有相依性。解除安裝 NVIDIA 套件會清除 CUDA 工具組。在安裝 NVIDIA 驅動程式之後，您必須解除安裝 CUDA 工具組。

Windows

如果您使用其中一個 AWS Marketplace 供應項目建立自訂 Windows AMI，則 AMI 必須是使用 Windows Sysprep 建立的標準化映像，以確保 GRID 驅動程式能夠正常運作。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。

安裝 NVIDIA 公有驅動程式

如果 中所述 AWS Marketplace AMIs 使用包含 NVIDIA 驅動程式的 AMI不符合您的使用案例，您可以安裝公有驅動程式並自帶授權。安裝選項包含下列項目：

- 選項 1：僅安裝驅動程式
- 選項 2：透過 CUDA 工具組安裝 (建議用於 Linux 發行版本)

P6-B200 和 P6-B300 執行個體類型考量

P6-B200 和 P6-B300 平台是唯一的，因為它們以 PCIe 裝置的形式向執行個體公開 Mellanox ConnectX 網路介面卡 NICs)。這些 NICs 不會做為一般網路介面，而是做為 NVSwitch 橋接器運作，提供用於初始化和設定 NVFabric 的控制路徑，這是 GPU 互連的 NVLink 拓撲。

若要完整初始化系統，NVIDIA Fabric Manager 必須設定 `NVFabric` 並建立 NVSwitch 拓撲。如此一來，InfiniBand 核心模組就能與 Mellanox ConnectX NICs通訊。

NVIDIA Fabric Manager 已包含於 CUDA 工具組中。我們建議為此執行個體類型使用 選項 2：透過 CUDA 工具組安裝。

選項 1：僅安裝驅動程式

若要安裝特定驅動程式，請登入您的執行個體，並從 http://www.nvidia.com/Download/Find.aspx：// 下載執行個體類型的 64 位元 NVIDIA 公有驅動程式。對於產品類型、產品系列及產品，請使用下表中顯示的選項。

然後遵循 NVIDIA 驅動程式安裝指南中的本機儲存庫安裝說明。 https://docs.nvidia.com/datacenter/tesla/driver-installation-guide/index.html

> **Note**
>
> P6-B200 和 P6-B300 執行個體類型需要安裝和設定與 NVIDIA CUDA Toolkit 隨附的其他套件。如需進一步資訊，請參閱 選項 2：透過 CUDA 工具組安裝 中適用於您的 Linux 發行版本的說明。

執行個體	產品類型	產品系列	產品	最低驅動程式版本
G3	Tesla	M-Class	M60	--
G4dn	Tesla	T 系列	T4	--
G5	Tesla	A 系列	A10	470.00 或更新版本
G5g[1]	Tesla	T 系列	T4G	470.82.01 或更新版本
G6	Tesla	L 系列	L4	525.0 或更新版本
G6e	Tesla	L 系列	L40S	535.0 或更新版本
Gr6	Tesla	L 系列	L4	525.0 或更新版本
G7	Tesla	RTX 系列	RTX PRO 4500 Blackwell	595 或更新版本
G7e	Tesla	RTX 系列	RTX PRO 6000 Blackwell	575.0 或更新版本
P3	Tesla	V 系列	V100	--
P4d	Tesla	A 系列	A100	--

執行個體	產品類型	產品系列	產品	最低驅動程式版本
P4de	Tesla	A 系列	A100	--
P5	Tesla	H 系列	H100	530 或更新版本
P5e	Tesla	H 系列	H200	550 或更新版本
P5en	Tesla	H 系列	H200	550 或更新版本
P6-B200[2]	Tesla	HGX 系列	B200	570 或更新版本
P6e-GB200	Tesla	HGX 系列	B200	570 或更新版本
P6-B300[2]	Tesla	HGX 系列	B300	580 或更新版本

[1] G5g 執行個體的作業系統為 Linux aarch64。

[2] 對於 P6-B200 和 P6-B300 執行個體類型，設定 NVIDIA Fabric Manager 有額外的安裝需求。

選項 2：透過 CUDA 工具組安裝

安裝說明會因作業系統而略有不同。若要使用 NVIDIA CUDA 工具組在您的執行個體上安裝公有驅動程式，請遵循適用於您的執行個體作業系統的說明。若您的執行個體作業系統未列於此，請遵循 NVIDIA 開發人員網站上的作業系統與執行個體類型架構說明。如需詳細資訊，請參閱 CUDA Toolkit Downloads。

有關執行個體類型架構或其他規格，請參閱《Amazon EC2 執行個體類型參考》中的加速運算規格。

Amazon Linux 2023

本節說明如何在 Amazon Linux 2023 執行個體上安裝 NVIDIA CUDA 工具組。本節中的命令範例基於 x86_64 架構。

如需 arm64-sbsa 命令，請參閱 CUDA Toolkit Downloads，然後選取套用至您的發行版本的選項。完成最終選擇後，顯示對應說明。

先決條件

安裝工具組與驅動程式前，執行下列命令，以確保您具備正確版本的核心標頭與開發套件。

```
[ec2-user ~]$ sudo dnf install kernel-devel-$(uname -r) kernel-headers-$(uname -r) -y
```

下載工具組與驅動程式

選擇適用於您執行個體的安裝類型，並遵循相關步驟。

AL2023 NVIDIA repository

Amazon Linux 2023 透過 AWS 維護的專用儲存庫提供 NVIDIA GPU 驅動程式和 CUDA 工具組套件。 符合此儲存庫與 AL2023 發行候選 AWS 版本的資格，並透過 Amazon Linux 安全中心提供安全建議。建議您對 AL2023 執行個體使用此選項，因為它可簡化安裝，並透過標準`dnf update`工作流程讓驅動程式保持最新狀態。

如需說明，請參閱《Amazon Linux 2023 使用者指南》中的 NVIDIA 驅動程式。

RPM local installation

您可遵循這些說明，將 CUDA 工具組安裝程式儲存庫套件下載至您的執行個體，然後解壓縮並註冊指定套件。

若要在 NVIDIA 開發人員網站上檢視說明，請參閱 CUDA Toolkit Downloads。

```
[ec2-user ~]$ wget https://developer.download.nvidia.com/compute/cuda/13.0.0/
local_installers/cuda-repo-amzn2023-13-0-local-13.0.0_580.65.06-1.x86_64.rpm
[ec2-user ~]$ sudo rpm -i cuda-repo-amzn2023-13-0-
local-13.0.0_580.65.06-1.x86_64.rpm
```

RPM network installation

您可遵循這些說明，在您執行個體的套件管理工具中註冊 CUDA 儲存庫。執行安裝步驟時，套件管理工具僅會下載必要的套件。

若要在 NVIDIA 開發人員網站上檢視說明，請參閱 CUDA Toolkit Downloads。

```
[ec2-user ~]$ sudo dnf config-manager --add-repo https://
developer.download.nvidia.com/compute/cuda/repos/amzn2023/x86_64/cuda-amzn2023.repo
```

RPM 本機和 RPM 網路安裝的剩餘步驟相同。

1. 完成 CUDA 工具組安裝

```
[ec2-user ~]$ sudo dnf clean all
```

```
[ec2-user ~]$ sudo dnf install cuda-toolkit -y
```

2. 安裝驅動程式的開放式核心模組版本

```
[ec2-user ~]$ sudo dnf module install nvidia-driver:open-dkms -y
```

3. 安裝 GPUDirect Storage 與 Fabric Manager

```
[ec2-user ~]$ sudo dnf install nvidia-gds -y
[ec2-user ~]$ sudo dnf install nvidia-fabric-manager -y
```

4. 啟用 Fabric Manager 與驅動程式持久性

```
[ec2-user ~]$ sudo systemctl enable nvidia-fabricmanager
[ec2-user ~]$ sudo systemctl enable nvidia-persistenced
```

5. (僅限 P6-B200 和 P6-B300) 這些執行個體類型需要安裝和設定隨附於 NVIDIA CUDA Toolkit 的其他套件。

   1. 安裝 NVIDIA Link Subnet Manager 與 `ibstat`。

      ```
      [ec2-user ~]$ sudo dnf install nvlink5
      ```

   2. 啟用在啟動時自動載入 Infiniband 模組。

      ```
      [ec2-user ~]$ echo "ib_umad" | sudo tee -a /etc/modules-load.d/modules.conf
      ```

6. 重新啟動執行個體

```
[ec2-user ~]$ sudo reboot
```

Ubuntu 24.04

本節說明如何在 Ubuntu 24.04 執行個體上安裝 NVIDIA CUDA 工具組。本節中的命令範例基於 `x86_64` 架構。

如需 `arm64-sbsa` 命令，請參閱 CUDA Toolkit Downloads，然後選取套用至您的發行版本的選項。完成最終選擇後，顯示對應說明。

先決條件

安裝工具組與驅動程式前，執行下列命令，以確保您具備正確版本的核心標頭與開發套件。

```
$ apt install linux-headers-$(uname -r)
```

下載工具組與驅動程式

選擇適用於您執行個體的安裝類型，並遵循相關步驟。

Deb local installation

您可遵循這些說明，將 CUDA 工具組安裝程式儲存庫套件下載至您的執行個體，然後解壓縮並註冊指定套件。

若要在 NVIDIA 開發人員網站上檢視說明，請參閱 CUDA Toolkit Downloads。

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/
x86_64/cuda-ubuntu2404.pin
$ sudo mv cuda-ubuntu2404.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/13.0.0/
local_installers/cuda-repo-ubuntu2404-13-0-local_13.0.0-580.65.06-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2404-13-0-local_13.0.0-580.65.06-1_amd64.deb
$ sudo cp /var/cuda-repo-ubuntu2404-13-0-local/cuda-*-keyring.gpg /usr/share/
keyrings/
```

Deb network installation

您可遵循這些說明，在您執行個體的套件管理工具中註冊 CUDA 儲存庫。執行安裝步驟時，套件管理工具僅會下載必要的套件。

若要在 NVIDIA 開發人員網站上檢視說明，請參閱 CUDA Toolkit Downloads。

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/
cuda-keyring_1.1-1_all.deb
$ sudo dpkg -i cuda-keyring_1.1-1_all.deb
```

本機安裝與網路安裝的其餘步驟相同。

1. 完成 CUDA 工具組安裝

```
$ sudo apt update
$ sudo apt install cuda-toolkit -y
```

2. 安裝驅動程式的開放式核心模組版本

```
$ sudo apt install nvidia-open -y
```

3. 安裝 GPUDirect Storage 與 Fabric Manager

```
$ sudo apt install nvidia-gds -y
$ sudo apt install nvidia-fabricmanager -y
```

4. 啟用 Fabric Manager 與驅動程式持久性

```
$ sudo systemctl enable nvidia-fabricmanager
$ sudo systemctl enable nvidia-persistenced
```

5. (僅限 P6-B200 和 P6-B300) 這些執行個體類型需要安裝和設定隨附於 NVIDIA CUDA Toolkit 的其他套件。

   1. 安裝最新的InfiniBand 特定裝置驅動程式」和診斷公用程式。

      ```
      $ sudo apt install linux-modules-extra-$(uname -r) -y
      $ sudo apt install infiniband-diags -y
      ```

   2. 安裝 NVIDIA Link Subnet Manager。

      ```
      $ sudo apt install nvlsm -y
      ```

6. 重新啟動執行個體

```
sudo reboot
```

7. 更新路徑並新增下列環境變數。

```
$ export PATH=${PATH}:/usr/local/cuda-13.0/bin
$ export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/cuda-13.0/lib64
```

Windows 作業系統

若要在 Windows 上安裝 NVIDIA 驅動程序，請依照下列步驟操作：

1. 開啟您下載驅動程式的所在資料夾，並啟動安裝檔案。依照指示安裝驅動程式，並視需要將執行個體重新開機。

2. 使用 Device Manager 停用標有警告圖示的名為 Microsoft Basic Display Adapter 的顯示轉接器。安裝這些 Windows 功能：Media Foundation 和 Quality Windows Audio Video Experience (高品質 Windows 音訊/視訊體驗)。

   > **Important**
   >
   > 請勿停用名為 Microsoft Remote Display Adapter 的顯示轉接器。如果停用 Microsoft Remote Display Adapter，您的連線可能會中斷，並在重新開機後會嘗試連線至執行個體可能會失敗。

3. 檢查裝置管理員，確認 GPU 是否正常運作。
4. 若要達到 GPU 的最佳效能，請完成在 Amazon EC2 執行個體最佳化 GPU 設定中的最佳化步驟。

安裝 NVIDIA GRID 驅動程式 (G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 執行個體)

這些下載僅供 AWS 客戶使用。透過下載，為了遵守 NVIDIA GRID 雲端最終使用者授權合約 (EULA) 中提及 AWS 的解決方案要求，您同意使用下載的軟體來開發 AMIs以便與 NVIDIA L4、NVIDIA L40S、NVIDIA A10G、NVIDIA Tesla T4 或 NVIDIA Tesla M60 硬體搭配使用。您可以使用 GRID 驅動程式在 AWS 環境中建立和使用 AMIs。安裝軟體時，您必須遵循 NVIDIA GRID 雲端最終使用者授權合約的條款。如需作業系統之 NVIDIA GRID 驅動程式版本的詳細資訊，請參閱 NVIDIA 網站上的 NVIDIA Virtual GPU (vGPU) 軟體。

考量事項

- G7e 執行個體需要 GRID 19.1 或更新版本，對於 Windows，則需要 19.4 (582.16) 或更新版本。
- G6f 和 Gr6f 執行個體需要 GRID 18.4 到 GRID 19.5。
- G6e 執行個體需要 GRID 17.4 或更高版本。
- G6 和 Gr6 執行個體需要 GRID 17.1 或更高版本。
- G5 執行個體需要 GRID 13.1 或更新版本 (或 GRID 12.4 或更新版本)。
- G3 執行個體需要 AWS 提供 DNS 解析，GRID 授權才能運作。
- 僅 14.0 版或更高版本的 NVIDIA 驅動程式支援 IMDSv2。
- 針對 Windows 執行個體，如果您從自訂 Windows AMI 啟動執行個體，則該 AMI 必須是使用 Windows Sysprep 建立的標準化映像，以確保 GRID 驅動程式能夠正常運作。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。
- GRID 17.0 及更高版本不支援 Windows Server 2019。

- GRID 14.2 及更高版本不支援 Windows Server 2016。
- G3 執行個體不支援 GRID 17.0 及更高版本。
- 對於 Linux 執行個體，若 NVIDIA 安裝程式失敗並顯示錯誤訊息，您可能需要安裝或更新諸如 gcc 等套件。詳細要求取決於作業系統與核心版本。如需進一步資訊，請參閱 NVIDIA 企業支援入口網站。

先決條件

- (Linux) 確認 AWS CLI 已安裝在您的執行個體上，並使用預設登入資料進行設定。如需詳細資訊，請參閱「AWS Command Line Interface 使用者指南」中的安裝 AWS CLI。
- (Windows) 為執行個體 AWS Tools for Windows PowerShell 上的 設定預設登入資料。如需詳細資訊，請參閱 AWS Tools for PowerShell 使用者指南中的 AWS Tools for Windows PowerShell入門。
- 使用者或角色必須擁有授予的許可，它包含 AmazonS3ReadOnlyAccess 政策。

Amazon Linux 2023

在執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo dnf update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝核心標頭套件。

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra
```

6. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

9. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

12. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

Amazon Linux 2

在執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝目前執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

如果您是使用 Amazon Linux 2 搭配核心版本 5.10，請使用下列命令來安裝 GRID 驅動程式。

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

9. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

12. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

CentOS 7 和 Red Hat Enterprise Linux 7

在執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 停用 NVIDIA 圖形卡的 nouveau 開放原始碼驅動程式。

   a. 將 nouveau 新增到 /etc/modprobe.d/blacklist.conf 黑名單檔案。複製下列程式碼區塊並貼至終端機。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. 編輯 /etc/default/grub 檔案並新增下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   c. 重建 Grub 組態。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

   此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

8. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

9. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

10. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
[ec2-user ~]$ nvidia-smi -q | head
```

11. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

12. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

13. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

    c. 安裝 GUI 桌上型電腦/工作站套件。

```
[ec2-user ~]$ sudo yum groupinstall -y "Server with GUI"
```

CentOS Stream 8 和 Red Hat Enterprise Linux 8

在執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

9. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

12. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

    c. 安裝 GUI 工作站套件。

```
[ec2-user ~]$ sudo dnf groupinstall -y workstation
```

Rocky Linux 8

在 Linux 執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

7. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

9. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

12. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

Ubuntu 和 Debian

在執行個體上安裝 NVIDIA GRID 驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
$ sudo apt-get update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
$ sudo apt-get install -y gcc make
```

3. (Ubuntu) 升級 `linux-aws` 套件以接收最新版本。

```
$ sudo apt-get upgrade -y linux-aws
```

(Debian) 升級套件以接收最新版本。

```
$ sudo apt-get upgrade -y
```

4. 將執行個體重新開機以載入最新的核心版本。

```
$ sudo reboot
```

5. 重新開機後，請重新連線至您的執行個體。
6. 安裝目前執行的核心版本的核心標頭套件。

```
$ sudo apt-get install -y linux-headers-$(uname -r) linux-modules-extra-$(uname -r)
```

7. 停用 NVIDIA 圖形卡的 nouveau 開放原始碼驅動程式。

   a. 將 nouveau 新增到 /etc/modprobe.d/blacklist.conf 黑名單檔案。複製下列程式碼區塊並貼至終端機。

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   b. 編輯 /etc/default/grub 檔案並新增下行：

   ```
   GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
   ```

   c. 重建 Grub 組態。

   ```
   $ sudo update-grub
   ```

8. 使用下列命令下載 GRID 驅動程式安裝公用程式：

```
$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

此儲存貯體會保留 GRID 驅動程式的多個版本。您可以使用下列命令查看所有可用版本。

```
$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

9. 請使用以下命令新增許可來執行驅動程式安裝公用程式。

```
$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 如下所述執行自我安裝指令碼，以安裝您已下載的 GRID 驅動程式。例如：

```
$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

11. 確認驅動程式可正常運作。下列命令的回應會列出已安裝的 NVIDIA 驅動程式版本和 GPU 詳細資料。

```
$ nvidia-smi -q | head
```

12. 如果您在 G4dn、G5 或 G5g 執行個體上使用 14.x 版或更高版本的 NVIDIA vGPU 軟體，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

13. 重新啟動執行個體。

```
$ sudo reboot
```

14. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

    a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

    b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

    c. 安裝 GUI 桌上型電腦/工作站套件。

```
$ sudo apt-get install -y lightdm ubuntu-desktop
```

Windows 作業系統

在 Windows 執行個體上安裝 NVIDIA GRID 驅動程式

1. 連接至 Windows 執行個體，然後開啟 PowerShell 視窗。
2. 使用下列 PowerShell 命令，將驅動程式和 NVIDIA GRID 雲端最終使用者授權合約從 Amazon S3 下載至您的桌面。

```
$Bucket = "ec2-windows-nvidia-drivers"
$KeyPrefix = "latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
$LocalFilePath = Join-Path $LocalPath $LocalFileName
Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
}
}
```

此儲存貯體會保留 NVIDIA GRID 驅動程式的多個版本。您可以移除 `-KeyPrefix $KeyPrefix` 選項，來下載儲存貯體中的所有可用 Windows 版本。如需作業系統之 NVIDIA GRID 驅動程式版本的詳細資訊，請參閱 NVIDIA 網站上的 NVIDIA Virtual GPU (vGPU) 軟體。

從 GRID 版本 11.0 開始，您可以針對 G3 和 G4dn 執行個體使用 `latest` 下的驅動程式。我們不會將 11.0 之後的版本新增至 `g4/latest`，但會將版本 11.0 和 G4dn 專用的舊版保留在 `g4/latest` 下。

G5 執行個體需要 GRID 13.1 或更新版本 (或 GRID 12.4 或更新版本)。

3. 導覽至桌面並開啟安裝檔案以啟動它 （選擇與您的執行個體作業系統版本對應的驅動程式版本）。依照指示安裝驅動程式，並視需要將執行個體重新開機。若要確認 GPU 是否正常運作，請檢查裝置管理員。
4. (選用) 使用下列命令在控制面板中停用授權頁面，以防止使用者意外變更產品類型 (NVIDIA GRID 虛擬工作站預設為啟用)。如需詳細資訊，請參閱「GRID 授權使用者指南」。

   PowerShell

執行下列 PowerShell 命令，以建立登錄值來停用控制面板中的授權頁面。 AWS Windows AMIs AWS Tools for PowerShell 中的 預設為 32 位元版本，且此命令失敗。請改用作業系統隨附的 64 位元版本 PowerShell。

```
New-Item -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name GridLicensing
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" -
Name "NvCplDisableManageLicensePage" -PropertyType "DWord" -Value "1"
```

命令提示

執行下列登錄命令以建立登錄值來停用控制面板中的授權頁面。您可以使用「命令提示字元」視窗或 64 位元版本的 PowerShell 來執行它。

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" /v
 NvCplDisableManageLicensePage /t REG_DWORD /d 1
```

5. (選用) 視您的使用案例而定，您可能會完成下列選用步驟。如果您不需要這項功能，請勿完成這些步驟。

   a. 為了協助善加利用最高 4K 解析度的四個顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

   b. 預設會啟用 NVIDIA Quadro 虛擬工作站。若要針對主控功能的 RDSH 應用程式啟動 GRID 虛擬應用程式，請完成在以 Amazon EC2 GPU 為基礎的執行個體啟用 NVIDIA GRID 虛擬應用程式中的 GRID 虛擬應用程式啟動步驟。

安裝 NVIDIA 遊戲驅動程式 (G7e、G6、G6e、G5 和 G4dn 執行個體)

這些驅動程式僅供 AWS 客戶使用。下載後，即表示您同意僅將下載的軟體用於開發 AMIs，以搭配 RTX PRO 6000 Blackwell、NVIDIA L4、NVIDIA L40S、NVIDIA A10G、NVIDIA Tesla T4 或 NVIDIA Tesla M60 硬體使用。您可以使用 GRID 驅動程式在 AWS 環境中建立和使用 AMIs。安裝軟體時，您必須遵循 NVIDIA GRID 雲端最終使用者授權合約的條款。

考量事項

- G3 執行個體需要 AWS 提供 DNS 解析，GRID 授權才能運作。
- 僅 495.x 版或更高版本的 NVIDIA 驅動程式支援 IMDSv2。

先決條件

- (Linux) 確認 AWS CLI 已安裝在您的執行個體上，並使用預設登入資料進行設定。如需詳細資訊，請參閱「AWS Command Line Interface 使用者指南」中的安裝 AWS CLI。
- 使用者或角色必須擁有授予的許可，它包含 AmazonS3ReadOnlyAccess 政策。

Amazon Linux 2023

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo dnf update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新啟動後，請重新連線至您的執行個體。
5. 安裝核心標頭套件。

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra kernel-devel-
$(uname -r) kernel-headers-$(uname -r) dkms
```

6. 使用下列命令下載遊戲驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. 從下載的 .zip 封存中擷取遊戲驅動程序安裝公用程式。

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令執行安裝程式：

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

10. 使用以下命令建立所需的組態檔案。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下載認證檔案及重新命名。

- 對於 590.48 版或更新版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 對於 575.64 至 580.105.08 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
```

- 對於 460.39 到 575.57 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 對於 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 對於較早版本

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

14. 使用下列命令以確認 NVIDIA 遊戲授權。

```
[ec2-user ~]$ nvidia-smi.exe -q
```

在輸出中，搜尋 `vGPU Software Licensed Product`。

15. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

Amazon Linux 2

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線到您的執行個體。更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新啟動後，請重新連線至您的執行個體。
5. 安裝目前執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 使用下列命令下載遊戲驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. 從下載的 .zip 封存中擷取遊戲驅動程序安裝公用程式。

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令執行安裝程式：

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

如果您是使用 Amazon Linux 2 搭配核心版本 5.10，請使用下列命令來安裝 NVIDIA 遊戲驅動程式。

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

10. 使用以下命令建立所需的組態檔案。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下載認證檔案及重新命名。

- 對於 590.48 版或更新版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 對於 575.64 至 580.105.08 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
```

- 對於 460.39 到 575.57 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 對於 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 對於較早版本

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

14. 使用下列命令以確認 NVIDIA 遊戲授權。

```
[ec2-user ~]$ nvidia-smi.exe -q
```

在輸出中，搜尋 `vGPU Software Licensed Product`。

15. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

CentOS 7 和 Red Hat Enterprise Linux 7

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線至您的 Linux 執行個體。如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝目前執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. 停用 NVIDIA 圖形卡的 nouveau 開放原始碼驅動程式。

   a. 將 nouveau 新增到 /etc/modprobe.d/blacklist.conf 黑名單檔案。複製下列程式碼區塊並貼至終端機。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. 編輯 /etc/default/grub 檔案並新增下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

c. 重建 Grub 組態。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 使用下列命令下載遊戲驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

8. 從下載的 .zip 封存中擷取遊戲驅動程序安裝公用程式。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

9. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

10. 使用以下命令執行安裝程式：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

11. 使用以下命令建立所需的組態檔案。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. 使用以下命令下載認證檔案及重新命名。

- 對於 590.48 版或更新版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 對於 575.64 至 580.105.08 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
```

- 對於 460.39 到 575.57 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 對於 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 對於較早版本

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

13. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

14. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

15. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。如果您不需要此功能，請勿完成此步驟。

CentOS Stream 8 和 Red Hat Enterprise Linux 8

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線至您的 Linux 執行個體。如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝目前執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. 使用下列命令下載遊戲驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. 從下載的 `.zip` 封存中擷取遊戲驅動程序安裝公用程式。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令執行安裝程式：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

10. 使用以下命令建立所需的組態檔案。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
```

```
vGamingMarketplace=2
EOF
```

11. 使用以下命令下載認證檔案及重新命名。

    - 對於 590.48 版或更新版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
      ```

    - 對於 575.64 至 580.105.08 版：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
      ```

    - 對於 460.39 到 575.57 版：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 對於 440.68 到 445.48 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 對於較早版本

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

    ```
    [ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
    ```

    ```
    [ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
    ```

13. 重新啟動執行個體。

    ```
    [ec2-user ~]$ sudo reboot
    ```

14. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

Rocky Linux 8

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線至您的 Linux 執行個體。如果尚未安裝 gcc 和 make，請進行安裝。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新套件快取，並取得執行個體的套件更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 將執行個體重新開機以載入最新的核心版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重新開機後，請重新連線至您的執行個體。
5. 安裝目前執行的核心版本的核心標頭套件。

```
[ec2-user ~]$ sudo dnf install -y unzip elfutils-libelf-devel libglvnd-devel
 kernel-devel-$(uname -r)
```

6. 使用下列命令下載遊戲驅動程式安裝公用程式：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

7. 從下載的 `.zip` 封存中擷取遊戲驅動程序安裝公用程式。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令執行安裝程式：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

10. 使用以下命令建立所需的組態檔案。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下載認證檔案及重新命名。

- 對於 590.48 版或更新版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 對於 575.64 至 580.105.08 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
```

- 對於 460.39 到 575.57 版：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 對於 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 對於較早版本

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

14. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。

Ubuntu 和 Debian

在執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連線至您的 Linux 執行個體。如果尚未安裝 gcc 和 make，請進行安裝。

```
$ sudo apt-get install -y gcc make build-essential
```

2. 更新套件快取，並取得執行個體的套件更新。

```
$ sudo apt-get update -y
```

3. 升級 `linux-aws` 套件以接收最新版本。

```
$ sudo apt-get upgrade -y linux-aws
```

4. 將執行個體重新開機以載入最新的核心版本。

```
$ sudo reboot
```

5. 重新開機後，請重新連線至您的執行個體。
6. 安裝目前執行的核心版本的核心標頭套件。

```
$ sudo apt install -y unzip dkms linux-headers-$(uname -r)
```

7. 停用 NVIDIA 圖形卡的 nouveau 開放原始碼驅動程式。

   a. 將 nouveau 新增到 /etc/modprobe.d/blacklist.conf 黑名單檔案。複製下列程式碼區塊並貼至終端機。

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   b. 編輯 /etc/default/grub 檔案並新增下行：

   ```
   GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
   ```

   c. 重建 Grub 組態。

   ```
   $ sudo update-grub
   ```

8. 使用下列命令下載遊戲驅動程式安裝公用程式：

   ```
   $ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
   ```

   此儲存貯體會保留遊戲驅動程式的多個版本。您可以使用下列命令查看所有可用版本：

   ```
   $ aws s3 ls --recursive s3://nvidia-gaming/linux/
   ```

9. 從下載的 .zip 封存中擷取遊戲驅動程序安裝公用程式。

   ```
   $ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
   ```

10. 請使用以下命令新增許可來執行驅動程式安裝公用程式：

    ```
    $ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
    ```

11. 使用以下命令執行安裝程式：

    ```
    $ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
    ```

出現提示時，請接受授權合約，並依需求指定安裝選項 (您可以接受預設選項)。

12. 使用以下命令建立所需的組態檔案。

```
$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

13. 使用以下命令下載認證檔案及重新命名。

- 對於 590.48 版或更新版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 對於 575.64 至 580.105.08 版：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2025_07_01.cert"
```

- 對於 460.39 到 575.57 版：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 對於 440.68 到 445.48 版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 對於較早版本

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

14. 如果您在 G4dn、G5 或 G5g 執行個體上使用 510.x 版或更高版本的 NVIDIA 驅動程式，請使用以下命令來停用 GSP。如需有關為何需要執行此程序的詳細資訊，請參閱 NVIDIA 的說明文件。

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

15. 重新啟動執行個體。

```
$ sudo reboot
```

16. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。如果您不需要此功能，請勿完成此步驟。

Windows 作業系統

在執行個體上安裝 NVIDIA 遊戲驅動程式之前，除了上述所有遊戲驅動程式的考量事項外，您還必須確保符合以下先決條件。

- 如果您使用自訂的 Windows AMI 啟動 Windows 執行個體，AMI 必須是使用 Sysprep 建立的標準化映像，以確保遊戲驅動程式能夠正常運作。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。
- 在 Windows 執行個體 AWS Tools for Windows PowerShell 上設定 的預設登入資料。如需詳細資訊，請參閱 AWS Tools for PowerShell 使用者指南中 AWS Tools for Windows PowerShell入門。

在 Windows 執行個體上安裝 NVIDIA 遊戲驅動程式

1. 連接至 Windows 執行個體，然後開啟 PowerShell 視窗。
2. 使用下列 PowerShell 命令下載並安裝遊戲驅動程式。

```
$Bucket = "nvidia-gaming"
$KeyPrefix = "windows/latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
$LocalFilePath = Join-Path $LocalPath $LocalFileName
Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
}
}
```

此 S3 儲存貯體會保留 NVIDIA GRID 驅動程式的多個版本。如果將 $KeyPrefix 變數的值從 "windows/latest" 變更為 "windows"，則可以下載儲存貯體中的所有可用版本。

3. 導覽至桌面並開啟安裝檔案以啟動它 （選擇與您的執行個體作業系統版本對應的驅動程式版本）。依照指示安裝驅動程式，並視需要將執行個體重新開機。若要確認 GPU 是否正常運作，請檢查裝置管理員。
4. 使用下列其中一種方法來註冊驅動程式。

   Version 527.27 or above

   使用 64 位元版本的 PowerShell 或命令提示字元視窗來建立下列登錄機碼。

   key：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global

   name (名稱)：vGamingMarketplace

   type (類型)：DWord

   value (值)：2

   PowerShell

   執行下列 PowerShell 命令來建立此登錄值。 AWS Windows AMIs AWS Tools for PowerShell 中的 預設為 32 位元版本，且此命令失敗。請改用作業系統隨附的 64 位元版本 PowerShell。

   ```
   New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global"
    -Name "vGamingMarketplace" -PropertyType "DWord" -Value "2"
   ```

   命令提示

   執行下列登錄命令來建立此登錄值。您可以使用「命令提示字元」視窗或 64 位元版本的 PowerShell 來執行它。

   ```
   reg add "HKLM\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global" /v
    vGamingMarketplace /t REG_DWORD /d 2
   ```

   Earlier versions

   使用 64 位元版本的 PowerShell 或命令提示字元視窗來建立下列登錄機碼。

key：HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global

name (名稱)：vGamingMarketplace

type (類型)：DWord

value (值)：2

PowerShell

執行下列 PowerShell 命令來建立此登錄值。 AWS Windows AMIs AWS Tools for PowerShell 中的 預設為 32 位元版本，且此命令失敗。請改用作業系統隨附的 64 位元版本 PowerShell。

```
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name
 "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

命令提示

執行下列登錄命令，使用「命令提示字元」視窗來建立此登錄機碼。您也可以在 64 位元版本的 PowerShell 中使用此命令。

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global" /v vGamingMarketplace /t
 REG_DWORD /d 2
```

5. 在 PowerShell 中執行下列命令。這會下載認證檔案、重新命名檔案 GridSwCert.txt，並將檔案移至系統磁碟機上的 Public Documents (公用文件) 資料夾。通常，資料夾路徑為 C:\Users\Public\Documents。

   - 對於 591.59 版或更新版本：

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCert_2026_03_02.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 對於 576.80 到 581.80 版：

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCertWindows_2025_07_01.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 對於 460.39 到 576.72 版：

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCertWindows_2024_02_22.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

- 對於 445.87 版本：

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2020_04.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

- 對於較早版本

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

6. 將執行個體重新開機。
7. 在執行個體上尋找 `nvidia-smi.exe` 檔案。

```
Get-ChildItem -Path C:\ -Recurse -Filter "nvidia-smi.exe"
```

使用下列命令以確認 NVIDIA 遊戲授權。將*##*取代為上一個命令的輸出中資料夾的名稱。

```
C:\Windows\System32\DriverStore\FileRepository\path\nvidia-smi.exe -q
```

輸出格式應類似以下內容。

```
vGPU Software Licensed Product
Product Name              : NVIDIA Cloud Gaming
License Status            : Licensed (Expiry: N/A)
```

8. (選用) 為協助善加利用最高 4K 解析度的單一顯示器，請設定高效能顯示器通訊協定 Amazon DCV。如果您不需要此功能，請勿完成此步驟。

在 EC2 Windows 執行個體上安裝 ENA 驅動程式

如果您的執行個體不是以 Amazon 提供的最新 Windows Amazon Machine Image (AMI) 為基礎，則請使用下列程序在執行個體上安裝目前的 ENA 驅動程式。您應該在方便重新啟動執行個體時，執行這項更新。如果安裝指令碼未自動重新啟動執行個體，則建議您重新啟動執行個體作為最後一步。

如果在執行個體執行時使用執行個體儲存體磁碟區來儲存資料，則當停止執行個體時會清除該資料。停止執行個體之前，請確認您已將所需的任何資料從執行個體儲存體磁碟區複製到永久性儲存，例如 Amazon EBS 或 Amazon S3。

先決條件

若要安裝或升級 ENA 驅動程式，Windows 執行個體必須符合以下先決條件：

- 已安裝 PowerShell 3.0 版或更新版本
- 本節所示命令必須在 64 位元版本的 PowerShell 中執行。請勿使用 x86 版本的 PowerShell。此為 32 位元版本的 Shell，不支援這些命令。

步驟 1：備份資料

建議您建立備份 AMI，以防您無法透過 Device Manager 復原變更。若要使用 建立備份 AMI AWS 管理主控台，請遵循下列步驟：

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取需要升級驅動程式的執行個體，並從執行個體狀態選單中選擇停止執行個體。
4. 在停止執行個體後，再次選取執行個體。若要建立備份，請從動作選單中選擇映像和範本，然後選擇建立映像。
5. 若要重新啟動執行個體，請從執行個體狀態選單中選擇啟動執行個體。

步驟 2：安裝或升級 ENA 驅動程式

您可以使用 AWS Systems Manager Distributor 或 PowerShell cmdlets 安裝或升級 ENA 驅動程式。如需進一步指示，請選取與您要使用之方法相符的索引標籤。

Systems Manager Distributor

您可以使用 Systems Manager Distributor 功能，將套件部署至 Systems Manager 管理的節點。您可以使用 Systems Manager Distributor 一次安裝 ENA 驅動程式套件，或使用排程的更新進行安裝。如需有關如何使用 Systems Manager Distributor 安裝 ENA 驅動程式套件 (`AwsEnaNetworkDriver`) 的詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的安裝或更新套件。

PowerShell

本節說明如何使用 PowerShell cmdlet 在執行個體上下載及安裝 ENA 驅動程式套件。

選項 1：下載並解壓縮最新版本

1. 連線到您的執行個體，並以本機管理員登入。
2. 使用 invoke-webrequest cmdlet 下載最新的驅動程式套件：

```
PS C:\> invoke-webrequest https://ec2-windows-drivers-
downloads.s3.amazonaws.com/ENA/Latest/AwsEnaNetworkDriver.zip -
outfile $env:USERPROFILE\AwsEnaNetworkDriver.zip
```

> **Note**
>
> 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```
>
> 或者，可以從執行個體的瀏覽器視窗下載最新的驅動程式套件。

3. 使用 expand-archive cmdlet 來解壓縮下載至執行個體的 zip 封存檔：

```
PS C:\> expand-archive $env:userprofile\AwsEnaNetworkDriver.zip -
DestinationPath $env:userprofile\AwsEnaNetworkDriver
```

選項 2：下載並解壓縮特定版本

1. 連線到您的執行個體，並以本機管理員登入。
2. 從 ENA Windows 驅動程式版本歷史記錄 表格中的版本連結下載您想要的特定版本的 ENA 驅動程式套件。
3. 將 zip 封存檔解壓縮至執行個體。

使用 PowerShell 安裝 ENA 驅動程式

無論您是下載最新的驅動程式還是特定版本，安裝步驟都相同。若要安裝 ENA 驅動程式，請遵循下列步驟。

1. 若要安裝驅動程式，請從執行個體上的 `AwsEnaNetworkDriver` 目錄中執行 `install.ps1` PowerShell 指令碼。如果發生錯誤，則請確認您使用的是 PowerShell 3.0 或更新版本。
2. 如果安裝程式未自動重新啟動執行個體，則請執行 Restart-Computer PowerShell cmdlet。

```
PS C:\> Restart-Computer
```

步驟 3 (選用)：安裝後驗證 ENA 驅動程式版本

若要確保 ENA 驅動程式套件已成功安裝在執行個體上，可以依照下列步驟驗證新版本：

1. 連線到您的執行個體，並以本機管理員登入。
2. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
3. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
4. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
5. 選取名稱，或開啟 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡) 的內容功能表，然後選擇 Properties (屬性)。這會開啟 Amazon 彈性網路介面卡屬性對話方塊。

> **Note**
>
> ENA 介面卡都使用相同的驅動程式。如果您有多個 ENA 介面卡，則可以選取其中一個來更新所有 ENA 介面卡的驅動程式。

6. 若要驗證目前安裝的版本，請開啟驅動程式索引標籤並檢查驅動程式版本。如果目前版本與目標版本不相符，則請參閱 為彈性網路轉接器 Windows 驅動程式進行疑難排解。

復原 ENA 驅動程式安裝

如果安裝出現任何問題，則您可能需要復原驅動程序。請依照下列步驟復原至執行個體上安裝的 ENA 驅動程式的先前版本。

1. 連線到您的執行個體，並以本機管理員登入。
2. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
3. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
4. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
5. 選取名稱，或開啟 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡) 的內容功能表，然後選擇 Properties (屬性)。這會開啟 Amazon 彈性網路介面卡屬性對話方塊。

> **Note**
>
> ENA 介面卡都使用相同的驅動程式。如果您有多個 ENA 介面卡，則可以選取其中一個來更新所有 ENA 介面卡的驅動程式。

6. 若要復原驅動程式，請開啟驅動程式索引標籤並選擇復原驅動程式。這會開啟驅動程式套件復原視窗。

> **Note**
>
> 如果驅動程式索引標籤未顯示復原驅動程式動作，或者動作無法使用，則表示執行個體上的驅動程式存放區不包含先前安裝的驅動程式套件。若要疑難排解此問題，請參閱 故障診斷方案 並展開非預期的 ENA 驅動程式版本已安裝部分。如需有關裝置驅動程式套件選取程序的詳細資訊，請參閱 Microsoft 文件網站上的 Windows 如何為裝置選取驅動程式套件。

追蹤 ENA Windows 驅動程式版本版本

Windows AMI 包含 ENA Windows 驅動程式，用於啟用增強型網路。

對於 Windows Server 2016 版及更新版本，我們建議您使用最新的驅動程式版本。如需舊版 Windows Server，請參閱下表以決定要使用的 ENA 驅動程式版本。

Windows Server 版本	ENA 驅動程式版本
Windows Server 2012 R2	2.6.0 和更舊版本
Windows Server 2012	2.6.0 和更舊版本
Windows Server 2008 R2	2.2.3 和舊版

ENA Windows 驅動程式版本歷史記錄

下表摘要說明各版本的變更。

驅動程式版本	詳細資訊	發行日期
2.11.0	新功能 • 新增動態佇列配置支援，向上擴展至 64 個 I/O 佇列 (取決於執行個體類型限制)。 錯誤修正 • 修復從裝置擷取實體 DMA 位址寬度時，缺少錯誤狀況處理的問題。	2025 年 8 月 1 日
2.10.0	錯誤修正 • 修復佇列初始化期間記憶體配置失敗的處理不當問題，避免未預期的重新啟動。 • 修復裝置功能被忽略時，可能發生的網路卸載設定錯誤問題。	2025 年 6 月 24 日

驅動程式版本	詳細資訊	發行日期
2.9.0	新功能 • 新增裝置起始的非同步重設請求支援。 • 新增處理裝置提供的最大大型 LLQ 深度值支援。 • 在 Windows 事件檢視器中新增事件 ID 58001，以增強對裝置設定錯誤所導致的未預期電源狀態轉換的可見性。 錯誤修正 • 修復裝置初始化期間記憶體配置失敗的處理不當問題，避免未預期的重新啟動。 • 修復中斷服務常式中的問題，該問題可能在裝置停止期間將 DPC 加入佇列，從而避免未預期的重新啟動。	2024 年 12 月 12 日
2.8.0	錯誤修正 • 修正了出口網路緩衝區清單 (NBL) 處理完整流程中的罕見狀況，此狀況可能會因為嘗試釋放已釋放的 NBL 而導致記憶體損毀。 • 修正了停用所有 LSO 與校驗和卸載時 L3 通訊協定的偵測錯誤，此錯誤可能會導致意外行為。	2024 年 9 月 30 日

驅動程式版本	詳細資訊	發行日期
2.7.0	新功能 • 已移除對 Windows Server 2012 (Windows 8) 和 Windows Server 2012 R2 (Windows 8.1) 的支援。這些作業系統版本已達到 AWS的支援期限。Windows Server 2012 及更早版本上的驅動程式安裝將會失敗。 • 新增支援將 IPv6 Tx 檢查總和計算卸載至裝置。 • 新增了寬低延遲佇列 (LLQ) 支援。此功能會根據裝置建議動態啟用。您可以使用新的「WideLLQ」登錄機碼覆寫此設定。 • 新增因 Rx 超載導致封包丟失的報告，這表示傳入封包的 Rx 環路空間不足。 • 已新增對來自裝置的次優組態通知的支援。請參閱 Windows 事件檢視器中的事件 ID 59000。 錯誤修正 • 避免因 Tx 封包的標頭超過最大 Low Latency Queuing (LLQ) 標頭大小而造成不必要的裝置重設。	2024 年 5 月 1 日

驅動程式版本	詳細資訊	發行日期

2.6.0	新功能
	• 為支援 ENA Express 的執行個體類型新增下列網路效能指標。
	  • ena_srd_mode
	  • ena_srd_tx_pkts
	  • ena_srd_eligible_tx_pkts
	  • ena_srd_rx_pkts
	  • ena_srd_resource_utilization
	• 為 Nitro 型執行個體類型新增 conntrack_allowance_available 網路效能指標。
	• 新增由於偵測到 RX 資料損毀而重設新介面卡的原因。
	• 更新驅動程式記錄的基礎設施。
	
	錯誤修正
	• 防止介面卡在 CPU 耗盡導致網路效能指標更新為失敗的情況下進行重設。
	• 防止對裝置心跳的中斷狀況進行錯誤偵測。
	• 修復驅動程式安裝指令碼以支援進行降級操作。
	• 修復接收錯誤計數統計的問題。
		2023 年 6 月 20 日

驅動程式版本	詳細資訊	發行日期
2.5.0	公告 由於無法在 Windows 域控制器上初始化，ENA Windows 驅動程式 2.5.0 版已復原。Windows Client 和 Windows Server 不受影響。	2023 年 2 月 17 日
2.4.0	新功能 • 新增對 Windows Server 2022 的支援。 • 移除對 Windows Server 2008 R2 的支援。 • 將低延遲佇列 (LLQ) 設定為一律開啟，以改善第六代 Amazon EC2 執行個體的效能。 錯誤修正 • 修復將網路效能指標發佈到 Windows 效能計數器 (PCW) 系統的故障問題。 • 修復登錄機碼讀取操作期間的記憶體流失問題。 • 防止在轉接器重設過程中發生無法復原的錯誤時，出現無限重設迴圈。	2022 年 4 月 28 日

驅動程式版本	詳細資訊	發行日期
2.2.4	公告 因第六代 EC2 執行個體的潛在效能下降，ENA Windows 驅動程式 2.2.4 版已回復。我們建議您使用下列其中一個方法，降級您的驅動程式： • 安裝舊版 1. 從此表格中的連結下載舊版套件 (版本 2.2.3)。 2. 執行 install.ps1 PowerShell 安裝指令碼。 如需有關安裝前和安裝後步驟的詳情，請參閱 啟用 Windows 上的增強型網路。 • 使用 Amazon EC2 Systems Manager 進行大量更新 • 透過 SSM 文件 `AWS-ConfigureAWSPackage` 執行大量更新，參數如下： • 名稱：AwsEnaNetworkDriver • 版本：2.2.3	2021 年 10 月 26 日

驅動程式版本	詳細資訊	發行日期
2.2.3	新功能 • 新增對具有最高 400 Gbps 執行個體聯網之新 Nitro 卡的支援。 錯誤修正 • 修正 ENA 驅動程式的系統時間變更與系統時間查詢之間的競爭條件，這會導致硬體無回應的誤判偵測。 公告 Windows ENA 驅動程式 2.2.3 版是支援 Windows Server 2008 R2 的最後一個版本。Windows Server 2008 R2 會繼續支援使用 ENA 的目前可用執行個體類型，且驅動程式可供下載。未來執行個體類型不會支援 Windows Server 2008 R2，而且您無法將 Windows Server 2008 R2 啟動、匯入或遷移到未來執行個體類型。	2021 年 3 月 25 日
2.2.2	新功能 • 針對 CloudWatch 和 Windows 消費者效能計數器所新增的支援，提供查詢網路轉接器效能指標的新功能。 錯誤修正 • 修正裸機執行個體的效能問題。	2020 年 12 月 21 日

驅動程式版本	詳細資訊	發行日期
2.2.1	新功能 • 新增方法以允許主機查詢彈性網路介面卡來取得網路效能指標。	2020 年 10 月 1 日
2.2.0	新功能 • 新增對下一代硬體類型的支援。 • 改善從停止休眠恢復之後的執行個體啟動時間，並消除誤判的 ENA 錯誤訊息。 效能最佳化 • 最佳化傳入流量的處理。 • 改善低資源環境中的共用記憶體管理。 錯誤修正 • 避免在極少數情況下，驅動程式無法重設的 ENA 裝置移除時系統當機。	2020 年 8 月 12 日
2.1.5	錯誤修正 • 修正裸機執行個體上偶發性的網路介面卡初始化失敗。	2020 年 6 月 23 日

驅動程式版本	詳細資訊	發行日期
2.1.4	錯誤修正 • 預防因為來自網路堆疊的 LSO 封包中繼資料毀損而造成的連線問題。 • 預防因為存取已釋放封包記憶體造成罕見競爭條件情況，而導致系統當機。	2019 年 11 月 25 日
2.1.2	新功能 • 將支援新增至開發商 ID 報告以允許作業系統產生以 MAC 為基礎的 UUIDs。 錯誤修正 • 在初始化期間改善 DHCP 網路組態效能。 • 當最大傳輸單元 (MTU) 超過 4K 時，正確計算輸入 IPv6 流量的 L4 檢查總和。 • 一般性改進增加穩定性和次要錯誤修正。	2019 年 11 月 4 日
2.1.1	錯誤修正 • 防止從作業系統抵達的高度分散式 TCP LSO 封包卸下。 • 適當處理 IPv6 網路中 IPSec 內的封裝安全承載 (ESP) 通訊協定。	2019 年 9 月 16 日

驅動程式版本	詳細資訊	發行日期

2.1.0	ENA Windows 驅動程式 2.1 版推出新的 ENA 裝置功能、提供效能加速、加入新功能，並包含多個穩定性改善。
	
	新功能
	
	• 對巨型訊框組態使用標準化的 Windows 登錄機碼。
	• 透過 ENA 驅動程式屬性 GUI 允許 VLAN ID 設定。
	• 改善復原流程
	  • 改善失敗識別機制。
	  • 已新增對可調整復原參數的支援。
	• 對具有 8 個以上 vCPU 的新 EC2 執行個體支援最多 32 個 I/O 佇列。
	• 驅動程式記憶體使用量減少最高達 90%。
	
	效能最佳化
	
	• 減少傳輸路徑延遲。
	• 支援接收檢查總和卸載。
	• 對負載繁重系統的效能最佳化 (鎖定機制的最佳化使用量)。
	• 進一步的增強可減少 CPU 使用量，並改善系統在負載下的回應能力。
		2019 年 7 月 1 日

驅動程式版本	詳細資訊	發行日期
	錯誤修正 • 修正由於非接續 Tx 標頭無效的剖析造成的損毀。 • 修正裸機執行個體上彈性網絡介面分離期間的驅動程式 1.5 版損毀。 • 修正透過 IPv6 的 LSO 虛擬標頭檢查總和計算錯誤。 • 修正初始化失敗時可能的記憶體資源流失。 • 為 IPv4 區段停用 TCP/UDP 檢查總和卸載。 • VLAN 組態的修正。在只應停用 VLAN 優先順序時，不正確地停用了 VLAN。 • 啟用事件檢視器對自訂驅動程式訊息的正確剖析。 • 由於無效的時間戳記處理，修正無法初始化驅動程式。 • 修正資料處理與 ENA 裝置停用之間的競爭條件。	
1.5.0	更新 • 已提升穩定性並修正效能。 • 在 ENA NIC 的 Advanced Properties (進階屬性) 中，您現在可以將 Receive Buffers (接收緩衝區) 的最高值設定為 8192。 • 預設 Receive Buffers (接收緩衝區) 為 1,000。	2018 年 10 月 4 日

驅動程式版本	詳細資訊	發行日期
1.2.3	包含從 Windows Server 2008 R2 到 Windows Server 2016 的可靠性修正並統一支援。	2018 年 2 月 13 日
1.0.8	初始版本。包含適用於 Windows Server 2008 R2、Windows Server 2012 RTM、Windows Server 2012 R2 和 Windows Server 2016 的 AMI。	2016 年 7 月

從 Amazon SNS 訂閱 ENA Windows 驅動程式版本通知

Amazon SNS 會在 EC2 Windows 驅動程式發佈新版本時通知您。使用下列程序訂閱這些通知。

訂閱 EC2 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須選取此區域，因為您要訂閱的 SNS 通知位於本區域中。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選擇 Create subscription (建立訂閱)。
5. 在 Create subscription (建立訂閱) 對話方塊中，執行下列動作：
   a. 針對 TopicARN，複製下列 Amazon Resource Name (ARN)：

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 針對 rotocol (通訊協定)，選擇 `Email`。
   c. 針對端點，輸入您要傳送通知的電子郵件地址。
   d. 選擇 Create subscription (建立訂閱)。
6. 您將會收到確認電子郵件。開啟電子郵件並遵循指示完成訂閱。

只要有新的 EC2 Windows 驅動程式發行，我們就會向訂閱者傳送通知。如果您不想再接收這些通知，請使用下列程序來取消訂閱。

取消訂閱 Amazon EC2 Windows 驅動程式通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 在導覽窗格中，選擇 Subscriptions (訂閱)。
3. 選取訂閱的核取方塊，然後選擇 Actions (動作)、Delete subscriptions (刪除訂閱)。出現確認提示時，請選擇 Delete (刪除)。

Windows 執行個體的全虛擬驅動程式

Windows AMI 包含一組驅動程式，可用來許可存取虛擬化硬體。Amazon EC2 會使用這些驅動程式將執行個體儲存體和 Amazon EBS 磁碟區映射到他們的裝置。下表顯示不同驅動程式的關鍵差異。

	Red Hat PV	Citrix PV	AWS PV
執行個體類型	不支援所有執行個體類型。若您指定不支援的執行個體類型，執行個體便會故障。	支援 Xen 執行個體類型。	支援 Xen 執行個體類型。
已連接磁碟區	支援最多 16 個已連接磁碟區。	支援超過 16 個已連接磁碟區。	支援超過 16 個已連接磁碟區。
網路	驅動程式有已知問題。當發生此問題時，網路連線會在高負載下重設 (例如快速 FTP 檔案傳輸)。		驅動程式會在位於相容的執行個體類型上時，自動在網路介面卡上設定 Jumbo Frame。當執行個體位於叢集置放群組時，這可在叢集置放群組的執行個體間提供更佳的

	Red Hat PV	Citrix PV	AWS PV
			網路效能。如需詳細資訊，請參閱Amazon EC2 執行個體的置放群組。

下列資料表顯示您在 Amazon EC2 之每個版本的 Windows Server 上建議執行的 PV 驅動程式。

Windows Server 版本	PV 驅動程式版本
Windows Server 2025	不支援
Windows Server 2022	AWS PV 最新版本
Windows Server 2019	AWS PV 最新版本
Windows Server 2016	AWS PV 最新版本
Windows Server 2012 R2	AWS PV 8.4.3 版
Windows Server 2012	AWS PV 8.4.3 版
Windows Server 2008 R2	AWS PV 8.3.5 版
Windows Server 2008	Citrix PV 5.9
Windows Server 2003	Citrix PV 5.9

目錄

- AWS PV 驅動程式
- Citrix PV 驅動程式
- Red Hat PV 驅動程式
- 訂閱 通知

- 升級 EC2 Windows 執行個體上的 PV 驅動程式
- 針對 Windows 執行個體上的 PV 驅動程式進行疑難排解

AWS PV 驅動程式

AWS PV 驅動程式會存放在 %ProgramFiles%\Amazon\Xentools目錄中。此目錄也包含公有符號及命令列工具 (xenstore_client.exe)，可讓您存取 XenStore 中的項目。例如，下列 PowerShell 命令會從虛擬化管理程序 (Hypervisor) 傳回目前的時間：

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
 AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

AWS PV 驅動程式元件會列在 下的 Windows 登錄檔中HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services。這些驅動程式元件如下：xenbus、xeniface、xennet、xenvbd 及 xenvif。

AWS PV 驅動程式也有名為 LiteAgent 的 Windows 服務，以使用者模式執行。它會處理 Xen 世代執行個體上來自 AWS APIs關機和重新啟動事件等任務。您可以藉由從命令列執行 Services.msc 來存取及管理服務。在 Nitro 世代執行個體上執行時，不會使用 AWS PV 驅動程式，LiteAgent 服務會從驅動程式 8.2.4 版開始自行停止。更新至最新的 AWS PV 驅動程式也會更新 LiteAgent，並改善所有執行個體世代的可靠性。

安裝最新的 AWS PV 驅動程式

Amazon Windows AMI 包含一組驅動程式，可用來許可存取虛擬化硬體。Amazon EC2 會使用這些驅動程式將執行個體儲存體和 Amazon EBS 磁碟區映射到他們的裝置。我們建議您安裝最新的驅動程式，改善您 EC2 Windows 執行個體的穩定性和效能。

安裝選項

- 使用 AWS Systems Manager 自動更新 PV 驅動程式。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的演練：自動更新 EC2 Windows 執行個體上的 PV 驅動程式。
- 下載 驅動程式套件，然後手動執行安裝程式。如需有關下載和安裝 AWS PV 驅動程式，或是升級網域控制器的資訊，請參閱手動升級 Windows Server 執行個體 (AWS PV 升級)。

如需安裝 AWS PV 驅動程式套件的系統需求，請參閱 AWS PV 驅動程式套件的系統需求。

AWS PV 驅動程式套件歷史記錄

下表顯示每個驅動程式版本的 AWS PV 驅動程式變更。

套件版本	詳細資訊	發行日期
8.6.1 8.6.1	• 已更新安裝程式套件中的執行時間相依性。	2026 年 7 月 21 日
8.6.0	• XenStore 互動的穩定性修正。	2025 年 5 月 27 日
8.5.0	• 穩定性修正，以解決網路裝置分離期間的罕見當機情況。 • 穩定性修正，以解決 EBS 磁碟區分離期間的罕見當機情況。 • 修正套件安裝程式的錯誤。 • 更新 PV 安裝程式以使用 `Pnputil`。	2024 年 10 月 31 日
8.4.3	修正了套件安裝程式中的錯誤，以改善升級體驗。此版本為可在 Windows Server 2012 與 2012 R2 上執行的最後版本。此版本可供下載，然後，由於 Windows Server 2012 與 2012 R2 已終止支援，因此不再支援此版本。	2023 年 1 月 24 日
8.4.2	解決爭用情況的穩定性修復。	2022 年 4 月 13 日
8.4.1	改進的套件安裝程式。	2022 年 1 月 7 日
8.4.0	• 穩定性修正，以解決磁碟 IO 的罕見卡住情況。 • 穩定性修正，以解決 EBS 磁碟區分離期間的罕見當機情況。 • 新增了將負載分散至多個核心的功能，以便工作負載充分利用超過 20,000 個 IOPS，以及應對因瓶頸而遭受降級的情況。若要啟用此功能，請參閱工作負載充分利用超過 20,000 個磁碟 IOPS，以及因 CPU 瓶頸而遭受降級的情況。	2021 年 3 月 2 日
8.3.5	改進的套件安裝程式。	2022 年 1 月 7 日

套件版本	詳細資訊	發行日期
	此版本為可在 Windows Server 2008 R2 上執行的最後版本。此版本可供下載，但不再受支援。Windows Server 2008 R2 已終止使用，且 Microsoft 不再提供支援。	
8.3.4	改善網路裝置連接的可靠性。	2020 年 8 月 4 日
8.3.3	• 更新為 XenStore 對應元件，以避免在錯誤處理路徑期間進行錯誤檢查。 • 更新為儲存元件，以防止提交無效的 SRB 時發生當機。 若要在 Windows Server 2008 R2 執行個體上更新此驅動程式，您必須先驗證已安裝適當的修補程式來處理下列 Microsoft Security Advisory：Microsoft Security Advisory 3033929。	2020 年 2 月 4 日
8.3.2	強化的聯網元件可靠性	2019 年 7 月 30 日
8.3.1	改善儲存體元件的效能和健全性。	2019 年 6 月 12 日
8.2.7	已改善效率，支援遷移至最新一代執行個體類型。	2019 年 5 月 20 日
8.2.6	提升損毀傾印路徑的效率。	2019 年 1 月 15 日
8.2.5	其他安全強化功能。 PowerShell 安裝程式現已在套件中提供。	2018 年 12 月 12 日
8.2.4	可靠性改進功能。	2018 年 10 月 2 日

套件版本	詳細資訊	發行日期
8.2.3	錯誤修正與效能改進。 將 EBS 磁碟區 ID 回報為 EBS 磁碟區的磁碟序號。這可啟用如 S2D 等叢集情境。	2018 年 5 月 29 日
8.2.1	網路及儲存體效能改善及多個強固性修正。 若要確認已安裝此版本，請參閱下列 Windows 登錄值：HKLM\Software\Amazon\PVDriver\Version 8.2.1 。	2018 年 3 月 8 日
7.4.3	新增 Windows Server 2016 支援。 所有支援的 Windows OS 版本穩定性修正。 *AWS PV 驅動程式 7.4.3 版的簽章將於 2019 年 3 月 29 日過期。我們建議您更新至最新的 AWS PV 驅動程式。	2016 年 11 月 18 日
7.4.2	X1 執行個體類型支援的穩定性修正。	2016 年 8 月 2 日
7.4.1	• AWS PV 儲存驅動程式的效能提升。 • AWS PV 儲存驅動程式中的穩定性修正：修正執行個體遇到錯誤檢查碼為 0x0000DEAD 的系統當機問題。 • AWS PV Network 驅動程式中的穩定性修正。 • 新增 Windows Server 2008R2 支援。	2016 年 7 月 12 日
7.3.2	• 改善記錄日誌及診斷。 • AWS PV 儲存驅動程式中的穩定性修正。在某些案例中，磁碟在重新連接到執行個體之後，可能不會在 Windows 中出現。 • 新增 Windows Server 2012 支援。	2015 年 6 月 24 日
7.3.1	TRIM 更新：與 TRIM 請求的相關修正。此修正會穩定執行個體，並改善執行個體管理大量 TRIM 請求時的效能。	

套件版本	詳細資訊	發行日期
7.3.0	TRIM 支援： AWS PV 驅動程式現在會將 TRIM 請求傳送至 Hypervisor。暫時性磁碟會在基礎儲存體支援 TRIM (SSD) 時適當處理 TRIM 請求。請注意，2015 年 3 月時 EBS 式儲存體尚不支援 TRIM。	
7.2.5	• AWS PV 儲存驅動程式中的穩定性修正：在某些情況下，AWS PV 驅動程式可能會反轉無效的記憶體並導致系統故障。 • 產生損毀傾印時的穩定性修正：在某些情況下， AWS PV 驅動程式可能會在寫入損毀傾印時卡在競爭條件中。在此版本之前，僅能透過強制驅動程式停止並重新啟動，遺失記憶體傾印來解決。	
7.2.4	裝置 ID 持久性：此驅動程式修正會遮罩平台 PCI 裝置 ID，強制系統一律呈現相同的裝置 ID，即使執行個體已移動也一樣。更一般的來說，修正會影響 Hypervisor 呈現虛擬裝置的方式。此修正還包含對 AWS PV 驅動程式的共同安裝程式的修改，以便系統持續映射虛擬裝置。	
7.2.2	• 在目錄服務還原模式 (DSRM) 模式中載入 AWS PV 驅動程式：目錄服務還原模式是 Windows Server 網域控制器的安全模式開機選項。 • 在重新連接虛擬網路介面卡裝置時持續裝置 ID：此修正會強制系統檢查 MAC 地址映射並持續裝置 ID。此修正會確保介面卡在重新連接時保留其靜態設定。	
7.2.1	• 於安全模式中執行：修正驅動程式無法在安全模式中載入的問題。先前 AWS PV 驅動程式只會在正常執行系統中執行個體化。 • 新增磁碟至 Microsoft Windows 儲存集區 (Storage Pools)：先前我們會合成 Page 83 查詢。修正會停用 Page 83 支援。請注意，這不會影響目前在叢集環境中使用的儲存集區，因為 PV 磁碟並非有效的叢集磁碟。	
7.2.0	基礎： AWS PV 基礎版本。	

Citrix PV 驅動程式

Citrix PV 驅動程式存放在 `%ProgramFiles%\Citrix\XenTools` (32 位元執行個體) 或 `%ProgramFiles(x86)%\Citrix\XenTools` (64 位元執行個體) 目錄。

Citrix PV 驅動程式元件會在 Windows 登錄中的 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services` 下列出。這些驅動程式元件如下：xenevtchn、xeniface、xennet、Xennet6、xensvc、xenvbd 及 xenvif。

Citrix 也具備名為 XenGuestAgent 的驅動程式元件，該元件會做為 Windows 服務執行。它會處理像是 API 的關機及重新啟動事件等任務。您可以藉由從命令列執行 `Services.msc` 來存取及管理服務。

若您在執行特定工作負載時發生網路錯誤，您可能需要停用 Citrix PV 驅動程式的 TCP 卸載功能。如需詳細資訊，請參閱TCP 卸載。

Red Hat PV 驅動程式

Red Hat 驅動程式支援舊版執行個體；由於驅動程式限制，不建議用於 RAM 大於 12GB 的較新執行個體。RAM 大於 12GB 的執行個體若執行 Red Hat 驅動程式，可能會無法開機而無法存取。我們建議將 Red Hat 驅動程式升級至 Citrix PV 驅動程式，然後將 Citrix PV 驅動程式升級至 AWS PV 驅動程式。

Red Hat 驅動程式的來源檔案位於 `%ProgramFiles%\RedHat` (32 位元執行個體) 或 `%ProgramFiles(x86)%\RedHat` (64 位元執行個體) 目錄中。兩個驅動程式為 Red Hat 全虛擬化網路驅動程式 (`rhelnet`) 及 Red Hat SCSI Miniport 驅動程式 (`rhelscsi`)。

訂閱 通知

Amazon SNS 會在 EC2 Windows 驅動程式發佈新版本時通知您。您可訂閱這些通知。

> **Note**
>
> 必須指定建立 SNS 主題的區域。

只要有新的 EC2 Windows 驅動程式發行，我們就會向訂閱者傳送通知。若您不想再接收這些通知，可取消訂閱。若要了解詳細資訊，請參閱刪除 SNS 主題與訂閱。

Console

訂閱 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。

2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須選取此區域，因為您要訂閱的 SNS 通知位於本區域中。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選擇 Create subscription (建立訂閱)。
5. 在 Create subscription (建立訂閱) 對話方塊中，執行下列動作：
   a. 針對 TopicARN，複製下列 Amazon Resource Name (ARN)：

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 針對 Protocol (通訊協定)，選擇 `Email`。
   c. 針對 Endpoint (端點)，輸入可用於接收通知的電子郵件地址。
   d. 選擇 Create subscription (建立訂閱)。
6. 您將會收到確認電子郵件。開啟電子郵件並遵循指示完成訂閱。

AWS CLI

訂閱 通知

使用下列 命令。

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers \
    --region us-east-1 \
    --protocol email \
    --notification-endpoint YourUserName@YourDomainName.ext
```

PowerShell

訂閱 通知

使用下列 命令。

```
Connect-SNSNotification `
    -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers' `
    -Region us-east-1 `
    -Protocol email `
    -Endpoint "YourUserName@YourDomainName.ext"
```

升級 EC2 Windows 執行個體上的 PV 驅動程式

我們建議您安裝最新的 PV 驅動程式，改善您 EC2 Windows 執行個體的穩定性和效能。此頁面上的指示可協助您下載驅動程式套件並執行安裝程式。

確認 Windows 執行個體使用的驅動程式

開啟裝置管理員，然後檢視網路轉接器。檢查 PV 驅動程式是否為下列項目的其中之一：

- AWS PV 網路裝置
- Citrix PV 乙太網路卡
- Red Hat PV NIC 驅動程式

目錄

- AWS PV 驅動程式套件的系統需求
- 使用 Distributor 升級 Windows Server 執行個體 (AWS PV 升級)
- 手動升級 Windows Server 執行個體 (AWS PV 升級)
- 升級網域控制器 (AWS PV 升級)
- 升級 Windows Server 2008 及 2008 R2 執行個體 (Red Hat 升級至 Citrix PV)
- 升級您的 Citrix Xen 客體代理程式服務

AWS PV 驅動程式套件的系統需求

無論您是透過 Systems Manager 或手動安裝 AWS PV 驅動程式套件，下列需求都適用。

- 套件版本支援的 Windows Server 版本。較舊版本的 Windows Server 需要較舊的套件版本。如需搭配作業系統使用的套件版本，請參閱 Windows 執行個體的全虛擬驅動程式。
- 套件安裝程式所需的最低 .NET Framework 版本，如下表所示。

AWS PV 驅動程式套件版本	最低 .NET Framework 版本
最新	4.7.2
8.4.3	4.5

AWS PV 驅動程式套件版本	最低 .NET Framework 版本
8.3.5	4.5

使用 Distributor 升級 Windows Server 執行個體 (AWS PV 升級)

您可以使用 的 Distributor AWS Systems Manager來安裝或升級 AWS PV 驅動程式套件。安裝或升級可以執行一次，或者您也可以按照排程安裝或更新。此 Distributor 套件不支援安裝類型的`In-place update`選項。

> **Important**
>
> 如果您的執行個體為網域控制站，請參閱升級網域控制器 (AWS PV 升級)。網域控制器執行個體的升級程序與標準版本的 Windows 不同。

1. 我們建議您建立備份，以防您需要復原變更。

   > **Tip**
   >
   > 您可以使用 Systems Manager Automation 來使用 `AWS-CreateImage` 執行手冊建立 AMI，而不是從 Amazon EC2 主控台建立 AMI。如需詳細資訊，請參閱「AWS Systems Manager Automation 執行手冊參考使用者指南」中的 AWS-CreateImage。

   a. 停止執行個體時，在任何執行個體儲存體磁碟區的資料都會清除。停止執行個體之前，請確認您已將所需的任何資料從執行個體儲存體磁碟區複製到永久性儲存，例如 Amazon EBS 或 Amazon S3。
   b. 在導覽窗格中，選擇 Instances (執行個體)。
   c. 選取需要升級驅動程式的執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
   d. 停止執行個體後，請選取執行個體，依序選取 Actions (動作)、Image and templates (映像和範本)，然後選取 Create image (建立映像)。
   e. 選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。
2. 使用遠端桌面連線到執行個體。如需詳細資訊，請參閱the section called “使用 RDP 用戶端連線”。

3. 建議您讓所有非系統磁碟離線，並且在磁碟管理中記下次要磁碟的任何磁碟機代號映射，之後再執行此升級。如果您正在執行 AWS PV 驅動程式的就地更新，則不需要此步驟。我們也建議在服務主控台中將非必要的服務設為 Manual (手動) 啟動。
4. 如需如何使用 Distributor 安裝或升級 AWS PV 驅動程式套件的指示，請參閱AWS Systems Manager 《 使用者指南》中的安裝或更新套件中的程序。
5. 在名稱欄位，選擇 AWSPVDriver。
6. 對於安裝類型，選取解除安裝並重新安裝。
7. 視需要設定套件的其他參數，並使用 Step 4 中的參考程序執行安裝或升級。

   在執行 Distributor 套件後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。
8. 在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查後，透過使用遠端桌面連線到執行個體來驗證是否已安裝新驅動程式。
9. 建立連線之後，請執行下列 PowerShell 命令：

   ```
   Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
   ```
10. 確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱 AWS PV 驅動程式套件歷史記錄。開啟磁碟管理來檢閱任何離線的次要磁碟區，並根據 Step 3 中記下的對應磁碟機代號映射讓它們回到線上。

如果您先前TCP 卸載已停用 Netsh for Citrix PV 驅動程式，建議您在升級至 AWS PV 驅動程式後重新啟用此功能。 AWS PV 驅動程式中不存在 Citrix 驅動程式的 TCP 卸載問題。因此，TCP 卸載可透過 AWS PV 驅動程式提供更好的效能。

如果您先前已將靜態 IP 地址或 DNS 組態套用至網路介面，您可能需要在升級 AWS PV 驅動程式之後重新套用靜態 IP 地址或 DNS 組態。

手動升級 Windows Server 執行個體 (AWS PV 升級)

使用下列程序執行 AWS PV 驅動程式就地升級，或從 Citrix PV 驅動程式升級至 Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2、Windows Server 2016、Windows Server 2019 或 Windows Server 2022 上的 AWS PV 驅動程式。此升級不適用於 Red Hat 驅動程式或其他版本的 Windows Server。

> **Important**
>
> 如果您的執行個體為網域控制站，請參閱升級網域控制器 (AWS PV 升級)。網域控制器執行個體的升級程序與標準版本的 Windows 不同。

手動升級 AWS PV 驅動程式

1. 我們建議您建立備份，以防您需要復原變更。

   > **Tip**
   >
   > 您可以使用 Systems Manager Automation 來使用 `AWS-CreateImage` 執行手冊建立 AMI，而不是從 Amazon EC2 主控台建立 AMI。如需詳細資訊，請參閱「AWS Systems Manager Automation 執行手冊參考使用者指南」中的 AWS-CreateImage。

   a. 停止執行個體時，在任何執行個體儲存體磁碟區的資料都會清除。停止執行個體之前，請確認您已將所需的任何資料從執行個體儲存體磁碟區複製到永久性儲存，例如 Amazon EBS 或 Amazon S3。
   b. 在導覽窗格中，選擇 Instances (執行個體)。
   c. 選取需要升級驅動程式的執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
   d. 停止執行個體後，請選取執行個體，依序選取 Actions (動作)、Image and templates (映像和範本)，然後選取 Create image (建立映像)。
   e. 選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。
2. 使用遠端桌面連線到執行個體。
3. 建議您讓所有非系統磁碟離線，並且在磁碟管理中記下次要磁碟的任何磁碟機代號映射，之後再執行此升級。如果您正在執行 AWS PV 驅動程式的就地更新，則不需要此步驟。我們也建議在服務主控台中將非必要的服務設為 Manual (手動) 啟動。
4. 使用下列其中一個選項下載驅動程式至您的執行個體：
   - 瀏覽器 – 下載 最新的驅動程式套件至執行個體，然後解壓縮 zip 封存。
   - PowerShell – 執行下列命令：

     ```
     Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/AWSPV/Latest/AWSPVDriver.zip -outfile $env:USERPROFILE\pv_driver.zip
     ```

```
Expand-Archive $env:userprofile\pv_driver.zip -DestinationPath $env:userprofile
\pv_drivers
```

如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

5. 執行 AWSPVDriverSetup.msi。

在執行 MSI 後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查之後，您可以使用遠端桌面連線到執行個體，然後執行下列 PowerShell 命令，來確認已安裝新的驅動程式：

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱 AWS PV 驅動程式套件歷史記錄。開啟磁碟管理來檢閱任何離線的次要磁碟區，並根據 Step 3 中記下的對應磁碟機代號映射讓它們回到線上。

如果您先前TCP 卸載已停用 Netsh for Citrix PV 驅動程式，建議您在升級至 AWS PV 驅動程式後重新啟用此功能。 AWS PV 驅動程式中不存在 Citrix 驅動程式的 TCP 卸載問題。因此，TCP 卸載可透過 AWS PV 驅動程式提供更好的效能。

如果您先前已將靜態 IP 地址或 DNS 組態套用至網路介面，您可能需要在升級 AWS PV 驅動程式之後重新套用靜態 IP 地址或 DNS 組態。

升級網域控制器 (AWS PV 升級）

在網域控制器上使用下列程序來執行 AWS PV 驅動程式就地升級，或從 Citrix PV 驅動程式升級至 AWS PV 驅動程式。如需確保 FSMO 角色在升級期間保持運作，建議您先轉移這些角色至其他網域控制器，然後再啟動升級。若要了解詳細資訊，請參閱 Microsoft Learn 網站上的如何檢視與轉移 FSMO 角色。

升級網域控制站

1. 我們建議您建立網域控制站的備份，以防您需要復原變更。不支援使用 AMI 作為備份。若要了解詳細資訊，請參閱 Microsoft 文件中的備份與還原考量。

2. 執行下列命令設定 Windows 開機進入目錄服務還原模式 (DSRM)。

> **Warning**
>
> 在執行此命令前，請確認您知道 DSRM 的密碼。您將需要此資訊，才能在升級完成且執行個體自動重新開機後登入您的執行個體。

```
bcdedit /set {default} safeboot dsrepair
```

PowerShell：

```
PS C:\> bcdedit /set "{default}" safeboot dsrepair
```

系統必須在 DSRM 中開機，因為升級公用程式會移除 Citrix PV 儲存驅動程式，以便安裝 AWS PV 驅動程式。因此，建議您在磁碟管理中記下次要磁碟的任何磁碟機代號和資料夾映射。當 Citrix PV 儲存體驅動程式不存在時，將無法偵測到輔助磁碟機。在輔助磁碟機上使用 NTDS 資料夾的網域控制站將無法開機，因為無法偵測到輔助磁碟。

> **Warning**
>
> 在您執行此命令後，請不要手動重新開機系統。否則將會無法觸及系統，因為 Citrix PV 驅動程式不支援 DSRM。

3. 執行下列命令，將 **`DisableDCCheck`** 新增到登錄：

```
reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t
 REG_SZ /d true
```

4. 下載 最新的驅動程式套件至執行個體，然後解壓縮 zip 封存。
5. 執行 `AWSPVDriverSetup.msi`。

   在執行 MSI 後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。
6. 在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查後，使用遠端桌面連線到執行個體。開啟磁碟管理來檢閱任何離線的輔助磁碟區，並根據稍早記下的對應磁碟機代號和資料夾映射讓磁碟區回到線上狀態。

您必須以下列格式指定使用者名稱來連線到執行個體：hostname\administrator。例如：Win2k12TestBox\administrator。

7. 執行下列命令移除 DSRM 開機組態：

```
bcdedit /deletevalue safeboot
```

8. 重新啟動執行個體。
9. 若要完成升級程序，請確認已安裝新的驅動程式。在裝置管理器中，在儲存體控制器下找到 AWS PV 儲存體主機轉接器。確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱AWS PV 驅動程式套件歷史記錄。
10. 執行下列命令，從登錄刪除 **`DisableDCCheck`**：

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck
```

> **Note**
>
> 如果您之前TCP 卸載停用 Netsh for Citrix PV 驅動程式，建議您在升級至 AWS PV 驅動程式後重新啟用此功能。 AWS PV 驅動程式中不存在 Citrix 驅動程式的 TCP 卸載問題。因此，TCP 卸載可透過 AWS PV 驅動程式提供更好的效能。

升級 Windows Server 2008 及 2008 R2 執行個體 (Red Hat 升級至 Citrix PV)

在您將您的 Red Hat 驅動程式升級至 Citrix PV 驅動程式前，請確認您已執行下列作業：

- 安裝最新版的 EC2Config 服務。如需詳細資訊，請參閱安裝最新版本的 EC2Config。
- 確認您已安裝 Windows PowerShell 3.0。若要確認您已安裝的版本，請在 PowerShell 視窗中執行下列命令：

```
PS C:\> $PSVersionTable.PSVersion
```

  Windows PowerShell 3.0 隨附在 Windows Management Framework (WMF) 3.0 版安裝套件內。如果您需要安裝 Windows PowerShell 3.0，請參閱 Microsoft 下載中心中的 Windows Management Framework 3.0。
- 備份您在執行個體上的重要資訊，或是從執行個體建立 AMI。如需建立 AMI 的詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

> **Tip**
>
> 您可以使用 Systems Manager Automation 來使用 `AWS-CreateImage` 執行手冊建立 AMI，而不是從 Amazon EC2 主控台建立 AMI。如需詳細資訊，請參閱「AWS Systems Manager Automation 執行手冊參考使用者指南」中的 AWS-CreateImage。

若您建立 AMI，請確認執行下列作業：

- 記下您的密碼。
- 請不要手動或使用 EC2Config 服務執行 Sysprep 工具。
- 設定您的乙太網路卡，以自動使用 DHCP 取得 IP 地址。

如需升級 Red Hat 驅動程式

1. 連線到您的執行個體，並以本機管理員登入。如需連線至執行個體的詳細資訊，請參閱使用 RDP 連線至您的 Windows 執行個體。
2. 在您的執行個體中，下載 Citrix PV 升級套件。
3. 將升級套件的內容解壓縮到您選擇的位置。
4. 開啟 Upgrade.bat 檔案。若您收到安全警告，請選擇 Run (執行)。
5. 在 Upgrade Drivers (升級驅動程式) 對話方塊中，檢閱資訊並於準備好開始升級時選擇 Yes (是)。
6. 在 Windows Red Hat 全虛擬化 Xen 驅動程式解除安裝程式 對話方塊中，選擇是以移除 Red Hat 軟體。您的執行個體將會重新開機。

> **Note**
>
> 若您沒有看到解除安裝程式對話方塊，請在 Windows 任務列中選擇 Red Hat Paravirtualize (Red Hat 全虛擬化)。

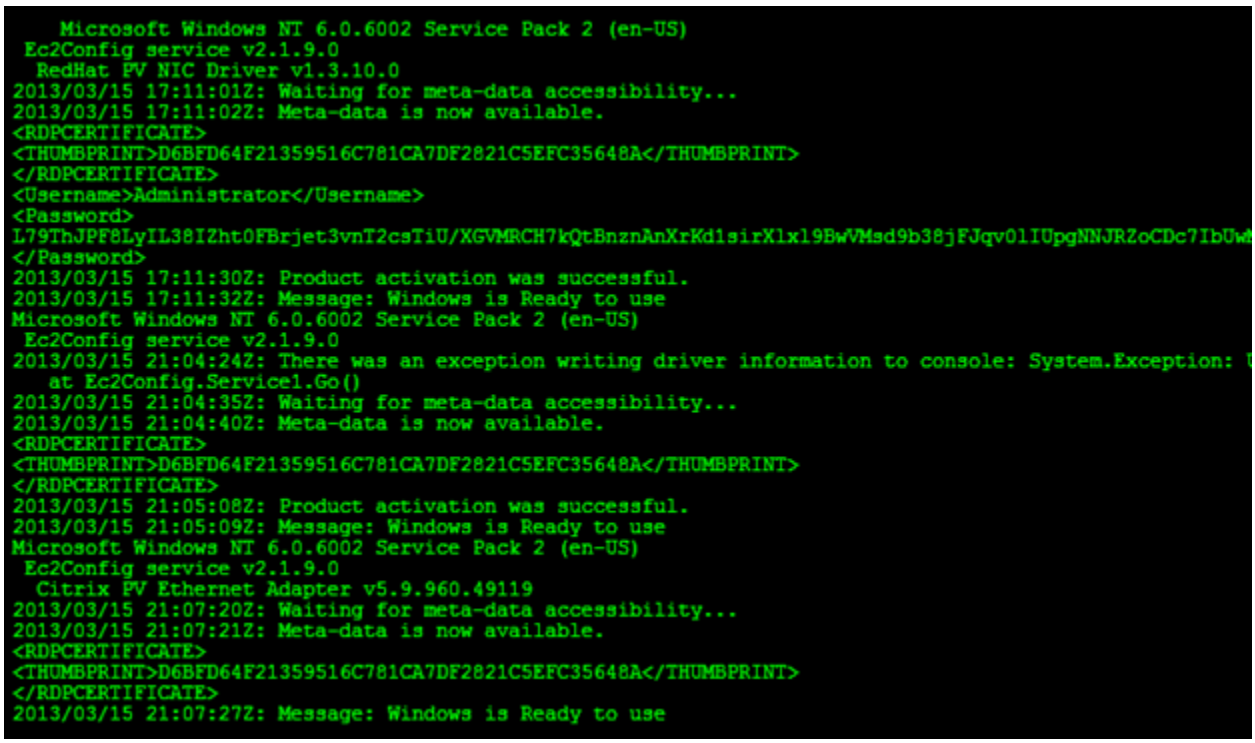


7. 檢查執行個體已重新開機並準備好可供使用。

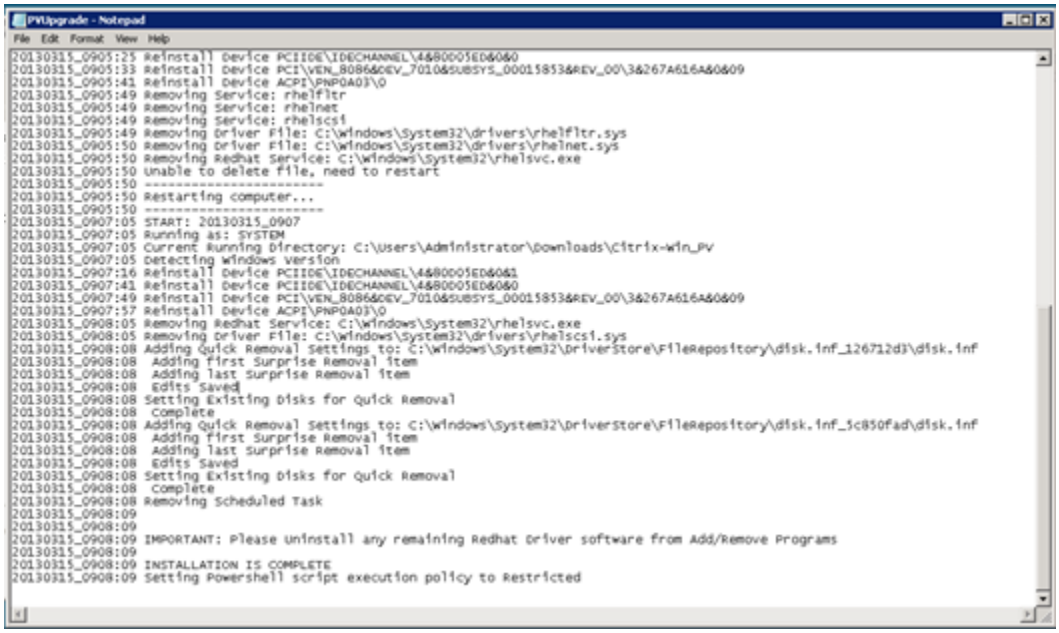
   a. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

b. 在 Instances (執行個體) 頁面上，依序選取 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)，然後選取 Get system log (取得系統記錄檔)。

c. 升級操作應該會重新開機伺服器 3 或 4 次。您可以在日誌檔案中，查看顯示的 `Windows is Ready to use` 次數來了解。



8. 連線到您的執行個體，並以本機管理員登入。
9. 關閉 Red Hat Paravirtualized Xen Drivers for Windows uninstaller (Windows Red Hat 全虛擬化 Xen 驅動程式解除安裝程式) 對話方塊。
10. 確認安裝已完成。導覽至您先前解壓縮的 `Citrix-WIN_PV` 資料夾，開啟 `PVUpgrade.log` 檔案，然後檢查是否有 `INSTALLATION IS COMPLETE` 的文字。



升級您的 Citrix Xen 客體代理程式服務

若您在 Windows Server 上使用 Citrix PV 驅動程式，您可以升級 Citrix Xen 客體代理程式服務。此 Windows 服務會處理像是 API 的關機及重新啟動事件等任務。您可以在任何版本的 Windows Server 上執行此升級套件，只要執行個體是執行 Citrix PV 驅動程式即可。

> **Important**
>
> 對於 Windows Server 2008 R2 和更新版本，我們建議您升級至包含訪客代理程式更新的 AWS PV 驅動程式。

在您開始升級您的驅動程式前，請確認您已備份您在執行個體上的重要資訊，或是從執行個體建立 AMI。如需建立 AMI 的詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

> **Tip**
>
> 您可以使用 Systems Manager Automation 來使用 `AWS-CreateImage` 執行手冊建立 AMI，而不是從 Amazon EC2 主控台建立 AMI。如需詳細資訊，請參閱「AWS Systems Manager Automation 執行手冊參考使用者指南」中的 AWS-CreateImage。

若您建立 AMI，請確認執行下列作業：

- 請不要在 EC2Config 服務中啟用 Sysprep 工具。
- 記下您的密碼。
- 將您的乙太網路卡設為 DHCP。

升級您的 Citrix Xen 客體代理程式服務

1. 連線到您的執行個體，並以本機管理員登入。如需連線至執行個體的詳細資訊，請參閱使用 RDP 連線至您的 Windows 執行個體。
2. 在您的執行個體上，下載 Citrix 升級套件。
3. 將升級套件的內容解壓縮到您選擇的位置。
4. 開啟 Upgrade.bat 檔案。若您收到安全警告，請選擇 Run (執行)。
5. 在 Upgrade Drivers (升級驅動程式) 對話方塊中，檢閱資訊並於準備好開始升級時選擇 Yes (是)。
6. 當升級完成時，將會開啟 `PVUpgrade.log` 檔案，其中包含 `UPGRADE IS COMPLETE` 的文字。
7. 將執行個體重新開機。

針對 Windows 執行個體上的 PV 驅動程式進行疑難排解

對於您使用舊版 Amazon EC2 映像和 PV 驅動程式時可能發生問題，以下是解決方案。

內容

- Windows Server 2012 R2 在執行個體重新開機後遺失網路及儲存體連線能力
- TCP 卸載
- 時間同步
- 工作負載充分利用超過 20,000 個磁碟 IOPS，以及因 CPU 瓶頸而遭受降級的情況

Windows Server 2012 R2 在執行個體重新開機後遺失網路及儲存體連線能力

> **Important**
>
> 僅 2014 年 9 月以前推出的 AMI 才會發生此問題。

2014 年 9 月 10 日之前開放使用的 Windows Server 2012 R2 Amazon Machine Images (AMI) 可能會在執行個體重新開機後遺失網路及儲存體的連線能力。 AWS 管理主控台 系統日誌狀態中的錯誤：「難以偵測主控台輸出的 PV 驅動程式詳細資訊」。連線能力遺失是由「隨插即用清理」功能造成的。

此功能會每 30 天掃描並停用非使用中的系統裝置。功能會不正確的將 EC2 網路裝置識別為非使用中，並從系統移除它。當發生此狀況時，執行個體便會在重新開機後遺失網路連線能力。

針對您懷疑受到此問題影響的系統，您可以下載及執行就地驅動程式升級。若您無法執行就地驅動程式升級，您可以執行協助程式指令碼。指令碼會判斷您的執行個體是否受到影響。若其受到影響，但 Amazon EC2 網路裝置尚未遭到移除，指令碼會停用隨插即用清理掃描。若已移除網路裝置，指令碼會修復裝置，停用隨插即用清理掃描，讓您的執行個體重新開機並啟用網路連線能力。

內容

- 選擇如何修正問題
- 方法 1 - 增強聯網
- 方法 2 - 登錄組態
- 執行補救指令碼

選擇如何修正問題

有兩種方法可以還原受此問題影響執行個體的網路及儲存體連線能力。選擇下列其中一種方法：

方法	必要條件	程序概觀
方法 1 - 增強聯網	增強聯網僅在 Virtual Private Cloud (VPC) 中可用，且需要 C3 執行個體類型。若伺服器目前使用的並非 C3 執行個體類型，您必須暫時變更它。	您將伺服器執行個體類型變更為 C3 執行個體。增強聯網便可讓您連線到受影響的執行個體並修正問題。在修正問題之後，您再將執行個體變更回原始的執行個體類型。此方法通常比方法 2 更快，也較不容易導致使用者錯誤。執行 C3 執行個體時，您將需要支付額外的費用。
方法 2 - 登錄組態	建立或存取第二個伺服器的能力。變更登錄設定的能力。	您將根磁碟區從受影響的執行個體分離，再將其連接到不同的執行個體、連線，並在登錄中進行變更。執行額外的伺服器時，您將需要支付額外的費用。此方法比方法 1 慢，但此

方法	必要條件	程序概觀
		方法可在方法 1 無法解決問題時使用。

方法 1 - 增強聯網

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 尋找受影響的執行個體。選取執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。

   > **Warning**
   >
   > 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

4. 在執行個體停止後，建立備份。選取執行個體，依序選取 Actions (動作)、Image and templates (映像和範本)，然後選取 Create image (建立映像)。
5. 將執行個體類型變更為任何 C3 執行個體類型。
6. 啟動實例。
7. 使用遠端桌面連線至執行個體，然後將 AWS PV 驅動程式升級套件下載至執行個體。
8. 解壓縮資料夾的內容，然後執行 `AWSPVDriverSetup.msi`。

   在執行 MSI 後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。
9. 在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查後，使用遠端桌面連線到執行個體，確認已安裝新的驅動程式。在裝置管理器中，在儲存體控制器下找到 AWS PV 儲存體主機轉接器。確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱AWS PV 驅動程式套件歷史記錄。
10. 停止執行個體，並將執行個體變更回原始的執行個體類型。
11. 啟動執行個體，繼續一般的使用。

方法 2 - 登錄組態

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 尋找受影響的執行個體。選取執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。

   > **Warning**
   >
   > 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

4. 選擇 Launch Instance (啟動執行個體)，在與受影響的執行個體相同的可用區域中建立暫時性的 Windows Server 2008 或 Windows Server 2012 執行個體。請不要建立 Windows Server 2012 R2 執行個體

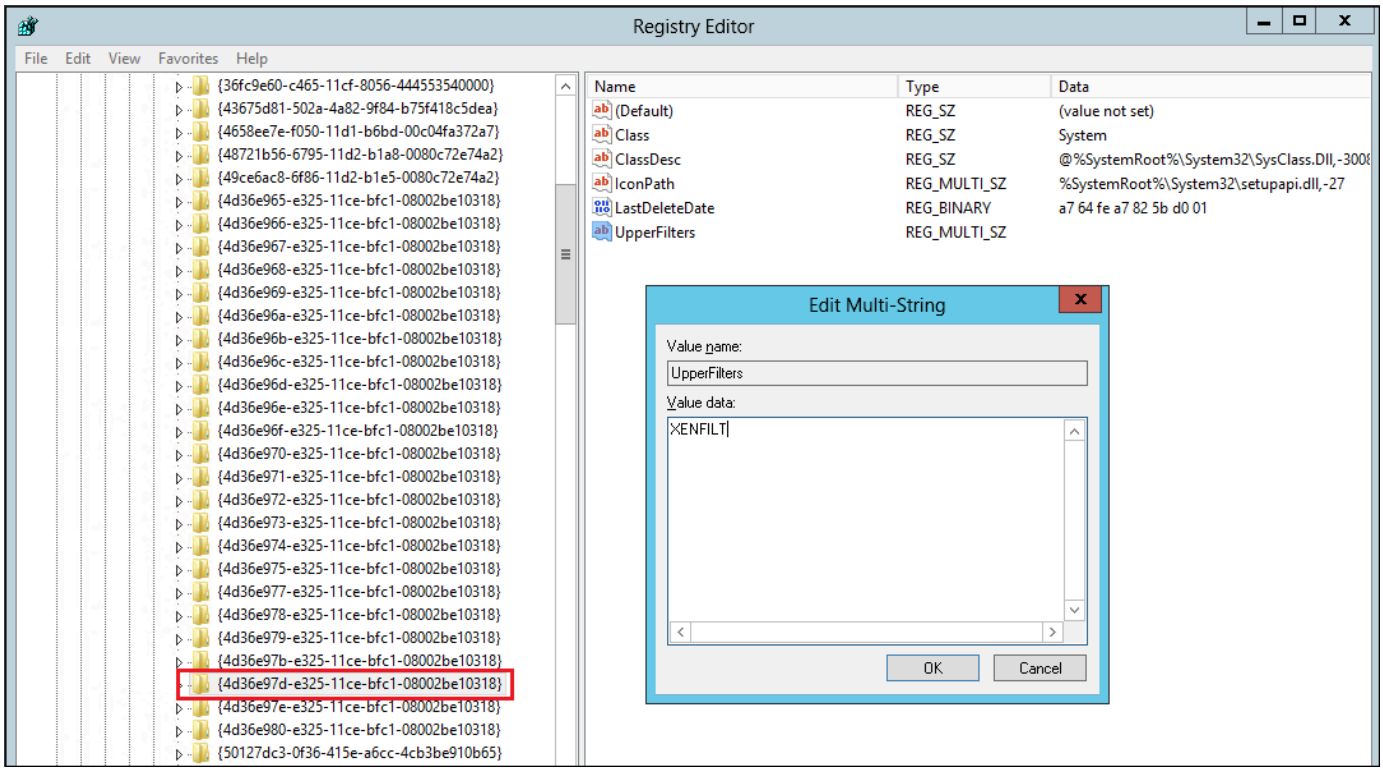
   > **Important**
   >
   > 若您沒有在與受影響的執行個體相同的可用區域內建立執行個體，您將無法將受影響執行個體的根磁碟區連接到新的執行個體。

5. 在導覽窗格中，選擇 Volumes (磁碟區)。
6. 尋找受影響執行個體的根磁碟區。分開磁碟區，再將磁碟區連接至先前建立的暫時執行個體。使用預設裝置名稱 (xvdf) 連接它。
7. 使用遠端桌面連接至暫時性執行個體，然後使用 Disk Management 公用程式將磁碟區變成可用。
8. 在暫時性執行個體上，開啟 Run (執行) 對話方塊，輸入 **`regedit`**，然後按 Enter 鍵。
9. 在登錄編輯程式導覽窗格中，選擇 HKEY_Local_Machine，然後從 File (檔案) 選單選擇 Load Hive (載入 Hive)。
10. 在 Load Hive (載入 Hive) 對話方塊中，導覽至 Affected Volume (受影響的裝置)\Windows\System32\config\System 然後在 Key Name (索引鍵名稱) 對話方塊中輸入暫時性的名稱。例如：輸入 OldSys。
11. 在登錄編輯程式的導覽窗格中，尋找下列索引鍵：

    HKEY_LOCAL_MACHINE***`your_temporary_key_name`**\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. 對於每個金鑰，開啟 UpperFilters，輸入 XENFILT 的值，然後選擇確定。



13. 尋找下列索引鍵：

HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS\Parameters

14. 使用 ActiveDevice 做為名稱來建立新的字串 (REG_SZ)，其值如下：

PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. 尋找下列索引鍵：

HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\XENBUS

16. 將 Count 從 0 變更為 1。
17. 尋找並刪除下列索引鍵：

HKEY_LOCAL_MACHINE*your_temporary_key_name*\ControlSet001\Services\xenvbd\StartOverride

HKEY_LOCAL_MACHINE *your_temporary_key_name*\ControlSet001\Services\xenfilt \StartOverride

18. 在登錄編輯程式導覽窗格中，選擇一開始開啟登錄編輯程式時建立的暫時性機碼。
19. 從 File (檔案) 選單中，選擇 Unload Hive (解除載入 Hive)。
20. 在磁碟管理公用程式中，選擇先前連接的磁碟機，開啟內容 (按右鍵) 選單，然後選擇 Offline (離線)。
21. 在 Amazon EC2 主控台中，從暫時性的執行個體分離受影響的磁碟區，然後以 /dev/sda1 做為裝置名稱，重新連接到您的 Windows Server 2012 R2 執行個體。您必須指定此裝置名稱，將磁碟區指定為根磁碟區。
22. 啟動實例。
23. 使用遠端桌面連線至執行個體，然後將 AWS PV 驅動程式升級套件下載至執行個體。
24. 解壓縮資料夾的內容，然後執行 `AWSPVDriverSetup.msi`。

    在執行 MSI 後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。
25. 在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查後，使用遠端桌面連線到執行個體，確認已安裝新的驅動程式。在裝置管理器中，在儲存體控制器下找到 AWS PV 儲存體主機轉接器。確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱AWS PV 驅動程式套件歷史記錄。
26. 刪除或停止您在此程序中建立的暫時性執行個體。

執行補救指令碼

若您無法執行就地驅動程式升級或遷移至較新的執行個體，您可以執行補救指令碼來修正隨插即用清理任務造成的問題。

執行補救指令碼

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您要執行補救指令碼的執行個體。依序選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

4. 在執行個體停止後，建立備份。選取執行個體，依序選取 Actions (動作)、Image and templates (映像和範本)，然後選取 Create image (建立映像)。
5. 依序選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。
6. 使用遠端桌面連線到執行個體，然後將 RemediateDriverIssue.zip 資料夾 下載到執行個體。
7. 解壓縮資料夾的內容。
8. 根據 Readme.txt 檔案中的說明執行補救指令碼。檔案位於您解壓縮 RemediateDriverIssue.zip 的資料夾中。

TCP 卸載

> **Important**
>
> 此問題不適用於執行 AWS PV 或 Intel 網路驅動程式的執行個體。

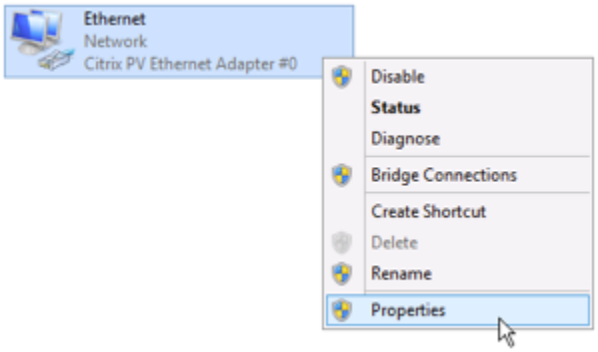
根據預設，Windows AMI 中的 Citrix PV 驅動程式會啟用 TCP 卸載。若您發生傳輸層級的錯誤或封包傳輸錯誤 (在 Windows 效能監視器中可見)—例如當您執行特定 SQL 工作負載時—您可能需要停用此功能。

> **Warning**
>
> 停用 TCP 卸載可能會降低您執行個體的網路效能。

停用 Windows Server 2012 和 2008 的 TCP 卸載

1. 連線到您的執行個體，並以本機管理員登入。
2. 若您使用 Windows Server 2012，請按 Ctrl+Esc 存取 Start (開始) 畫面，然後選擇 Control Panel (控制台)。若您使用 Windows Server 2008，請選取 Start (開始)，然後選取 Control Panel (控制台)。

3. 依序選擇 Network and Internet (網路和網際網路) 和 Network and Sharing Center (網路和共用中心)。
4. 選擇 Change adapter settings (變更介面卡設定)。
5. 開啟 Citrix PV 乙太網路轉接器 #0 的內容 （ 按一下滑鼠右鍵 ） 選單，然後選取屬性。



6. 在 Local Area Connection Properties (區域連線屬性) 對話方塊中，選擇 Configure (設定) 以開啟 Citrix PV Ethernet Adapter #0 Properties (Citrix PV 乙太網路卡 #0 屬性) 對話方塊。
7. 在 Advanced (進階) 標籤上，停用每個屬性，但 Correct TCP/UDP Checksum Value (修正 TCP/UDP 總和檢查碼值) 除外。若要停用屬性，請從 Property (屬性) 中選取，然後從 Value (值) 選取 Disabled (停用)。
8. 選擇 OK (確定)。
9. 從命令列視窗中執行下列命令。

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. 重新啟動執行個體。

時間同步

在 2013.02.13 Windows AMI 發行前，Citrix Xen 客體代理程式可能會不正確的設定系統時間。此可能導致您的 DHCP 租用到期。若您在連線到您的執行個體時發生問題，您可能需要更新代理程式。

若要判斷您是否已具有更新後的 Citrix Xen 客體代理程式，請檢查 `C:\Program Files\Citrix\XenGuestAgent.exe` 檔案是否來自 2013 年 3 月。若此檔案的日期在該日期之前，請更新 Citrix Xen 客體代理程式服務。如需詳細資訊，請參閱升級您的 Citrix Xen 客體代理程式服務。

工作負載充分利用超過 20,000 個磁碟 IOPS，以及因 CPU 瓶頸而遭受降級的情況

如果您使用的 Windows 執行個體執行 AWS PV 驅動程式，其充分利用超過 20,000 個 IOPS，並且遇到錯誤檢查碼 `0x9E: USER_MODE_HEALTH_MONITOR`，則可能會受此問題的影響。

AWS PV 驅動程式中的磁碟讀取和寫入 (IOs) 分為兩個階段：IO 準備和 IO 完成。依預設，準備階段會在單一任意核心上執行。完成階段會在核心 `0` 上執行。處理 IO 所需的運算量因其大小和其他屬性而有所差異。某些 iOS 在準備階段使用更多運算，其他則在完成階段如此。當執行個體驅動超過 20,000 個 IOPS 時，準備或完成階段可能會導致瓶頸，其執行所在的 CPU 佔用 100% 的容量。準備或完成階段是否成為瓶頸，取決於應用程式使用的 IOs 屬性。

從 AWS PV 驅動程式 8.4.0 開始，準備階段和完成階段的負載可以分散到多個核心，消除瓶頸。每個應用程式使用不同的 IO 屬性。因此，套用下列其中一個組態可能會提升、降低或不會影響應用程式的效能。套用任何這些組態之後，請監控應用程式，以確認其符合您所需的效能。

1. 先決條件

   開始此疑難排解程序之前，請先確認下列先決條件：

   - 您的執行個體使用 AWS PV 驅動程式 8.4.0 版或更新版本。若要升級，請參閱升級 EC2 Windows 執行個體上的 PV 驅動程式。
   - 您擁有對執行個體的 RDP 存取權。如需了解使用 RDP 連線至 Windows 執行個體的步驟，請參閱使用 RDP 用戶端連線至您的 Windows 執行個體。。
   - 您擁有執行處理的管理員存取權。
2. 觀察執行個體上的 CPU 負載

   您可以使用 Windows 任務管理員來檢視每個 CPU 上的負載，以確定磁碟 IO 的潛在瓶頸。

   1. 確認您的應用程式正在執行，並處理與生產工作負載類似的流量。
   2. 使用 SSH 連線至您的執行個體。
   3. 選擇執行個體上的 Start (開始) 功能表。
   4. 請在 Start (開始) 功能表中輸入 `Task Manager`，以開啟 Task Manager (任務管理員)。
   5. 如果 Task Manager (任務管理員) 顯示 Summary View (摘要檢視)，請選擇 More details (更多詳細資訊)，以展開詳細檢視。
   6. 選擇 Performance (效能) 索引標籤。
   7. 在左側窗格中選取 CPU。
   8. 在主窗格中開啟圖形的內容（按一下滑鼠右鍵）選單，然後選取將圖形變更為>邏輯處理器以顯示每個核心。

9. 根據您的執行個體上有多少核心，您可能會看到隨時間顯示 CPU 負載的行，或者可能只會看到一個數字。
   - 如果您看到隨時間顯示負載的圖形，請尋找方塊幾乎完全著色的 CPU。
   - 如果您在每個核心上看到一個數字，請尋找一致顯示 95% 以上的核心。

   10請注意核心 0 或不同的核心正在經歷繁重的負載。

3. 選擇要套用的組態

組態名稱	套用此組態的時間	備註
Default configuration	工作負載驅動少於 20,000 個 IOPS，或其他組態並未改善效能或穩定性。	針對此組態，IO 會發生在幾個核心上，這可能會藉由增加快取區域性並減少內容切換，從較小工作負載獲益。
Allow driver to choose whether to distribute completion	工作負載驅動超過 20,000 個 IOPS，核心 0 上觀察到中度或高負載。	對於使用 PV 8.4.0 或更新版本並利用超過 20，000 IOPS 的所有 Xen 執行個體，無論是否遇到問題，建議使用此組態。
Distribute both preparation and completion	工作負載驅動超過 20,000 個 IOPS，並且允許驅動程式選擇分發不改善效能，或是 0 之外的核心正在遇到高負載。	此組態會啟用 IO 準備和 IO 完成的分發。

> **Note**
>
> 未同時分發 IO 完成 (設定`DpcRedirection` 而不設定`NotifierDistributed`) 的情況下，我們建議您不要分發 IO 準備，因為當準備階段平行執行時，完成階段會對準備階段過載較為敏感。

登錄金鑰值

- NotifierDistributed

值 0 或不存在 – 完成階段將在核心 0 上執行。

值 1 – 驅動程式選擇執行完成階段或核心 0，或者每個連接磁碟有另一個核心。

值 2 – 驅動程式會在每個連接磁碟的另一個核心上執行完成階段。

- DpcRedirection

  值 0 或不存在 – 準備階段將會在單一、任意核心上執行。

  值 1 – 準備階段分發在多個核心之間。

預設組態

使用 8.4.0 之前的 AWS PV 驅動程式版本套用預設組態，或在套用本節中的其他組態後觀察到效能或穩定性降低。

1. 使用 SSH 連線至您的執行個體。
2. 以系統管理員身分開啟新的 PowerShell 命令提示。
3. 執行下列命令以移除 `NotifierDistributed` 和 `DpcRedirection` 登錄金鑰。

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name NotifierDistributed
```

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name DpcRedirection
```

4. 將執行個體重新開機。

允許驅動程式選擇是否要分配完成

設定`NotiferDistributed`登錄機碼，以允許 PV 儲存驅動程式選擇是否要分配 IO 完成。

1. 使用 SSH 連線至您的執行個體。
2. 以系統管理員身分開啟新的 PowerShell 命令提示。
3. 執行下列命令以設定 `NotiferDistributed` 登錄金鑰。

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name NotifierDistributed
```

4. 將執行個體重新開機。

分配準備和完成

設定 NotifierDistributed 和 DpcRedirection 登錄金鑰，以始終分發發準備和完成階段。

1. 使用 SSH 連線至您的執行個體。
2. 以系統管理員身分開啟新的 PowerShell 命令提示。
3. 執行下列命令以設定 NotifierDistributed 和 DpcRedirection 登錄金鑰。

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000002 -Name NotifierDistributed
```

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name DpcRedirection
```

4. 將執行個體重新開機。

NVMe 驅動程式

Amazon EBS 磁碟區和執行個體儲存體會公開為 Nitro 型執行個體上的 NVMe 區塊型儲存設備。若要將 Amazon EBS 功能的效能和功能完整用於公開為 NVMe 區塊裝置的磁碟區，執行個體必須安裝 AWS NVMe 驅動程式。所有目前世代的 AWS Windows AMI 都隨附預設安裝的 AWS NVMe 驅動程式。 AMIs

如需 EBS 和 NVMe 的詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 和 NVMe。如需 SSD 執行個體儲存體和 NVMe 的詳細資訊，請參閱 EC2 執行個體的 SSD 執行個體儲存體磁碟區。

Linux 執行個體

下列 AMI 包含下列必要 NVMe 驅動程式：

- Amazon Linux 2

- Amazon Linux AMI 2018.03
- Ubuntu 14.04 或更新版本 (帶 `linux-aws` 核心)

> **Note**
>
> AWS Graviton 型執行個體類型需要具有核心的 Ubuntu 18.04 `linux-aws` 或更新版本

- Red Hat Enterprise Linux 7.4 或更新版本
- SUSE Linux Enterprise Server 12 SP2 或更新版本
- CentOS 7.4.1708 或更新版本
- FreeBSD 11.1 或更新版本
- Debian GNU/Linux 9 或更新版本

確認您的執行個體具有 NVMe 驅動程式

您可以使用以下命令來確認您的執行個體具有 NVMe 驅動程式。

- Amazon Linux、RHEL、CentOS 和 SUSE Linux Enterprise Server

```
$ modinfo nvme
```

  如果執行個體具有 NVMe 驅動程式，命令會傳回驅動程式的相關資訊。

- Amazon Linux 2 和 Ubuntu

```
$ ls /sys/module/ | grep nvme
```

  如果執行個體具有 NVMe 驅動程式，命令會傳回已安裝的驅動程式。

更新 NVMe 驅動程式

如果您的執行個體具有 NVMe 驅動程式，您可以使用下列程序將驅動程式更新為最新版本。

1. 連線到您的 執行個體。
2. 更新套件快取，以取得如下的必要套件更新。
   - 若是 Amazon Linux 2、Amazon Linux、CentOS 與 Red Hat Enterprise Linux：

```
[ec2-user ~]$ sudo yum update -y
```

- 若是 Ubuntu 和 Debian：

```
[ec2-user ~]$ sudo apt-get update -y
```

3. Ubuntu 16.04 和更新版本包含 `linux-aws` 套件，其中包含 Nitro 型執行個體所需的 NVMe 和 ENA 驅動程式。升級 `linux-aws` 套件以接收如下的最新版本：

```
[ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
```

若要 Ubuntu 14.04，您可以安裝如下的最新 `linux-aws` 套件：

```
[ec2-user ~]$ sudo apt-get install linux-aws
```

4. 將執行個體重新開機以載入最新的核心版本。

```
sudo reboot
```

5. 重新開機後，請重新連線至您的執行個體。

Windows 執行個體

PowerShell

如果您未從 Amazon 提供的最新 AWS Windows AMIs 之一啟動執行個體，請使用下列程序在您的執行個體上安裝 current AWS NVMe 驅動程式。在進行此安裝時需要重新啟動。安裝指令碼會重新啟動您的執行個體，或是您必須重新啟動，以作為最後一個步驟。

先決條件

- 已安裝 PowerShell 3.0 版或更新版本
- 本節所示命令必須在 64 位元版本的 PowerShell 中執行。請勿使用 x86 版本的 PowerShell。此為 32 位元版本的 Shell，不支援這些命令。

下載並安裝 latest AWS NVMe 驅動程式

1. 我們建議您根據以下方式建立 AMI 作為備份，以防您需要復原變更。

a. 停止執行個體時，在任何執行個體儲存體磁碟區的資料都會清除。停止執行個體之前，請確認您已將所需的任何資料從執行個體儲存體磁碟區複製到永久性儲存，例如 Amazon EBS 或 Amazon S3。

b. 在導覽窗格中，選擇 Instances (執行個體)。

c. 選取需要升級驅動程式的執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。

d. 停止執行個體後，請選取執行個體，依序選取 Actions (動作)、Image and templates (映像和範本)，然後選取 Create image (建立映像)。

e. 選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。

2. 連線到您的執行個體，並以本機管理員登入。

3. 使用下列其中一個選項下載驅動程式至您的執行個體：

   - 瀏覽器 – 下載 最新的驅動程式套件至執行個體，然後解壓縮 zip 封存。
   - PowerShell – 執行下列命令：

     ```
     Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/
     NVMe/Latest/AWSNVMe.zip -outfile $env:USERPROFILE\nvme_driver.zip
     Expand-Archive $env:userprofile\nvme_driver.zip -DestinationPath
      $env:userprofile\nvme_driver
     ```

     如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：

     ```
     [Net.ServicePointManager]::SecurityProtocol =
      [Net.SecurityProtocolType]::Tls12
     ```

4. 透過執行 `nvme_driver` 目錄 (`.\install.ps1`) 中的 `install.ps1` PowerShell 指令碼，將驅動程式安裝到您的執行個體。如果發生錯誤，請確認您使用的是 PowerShell 3.0 或更新版本。

   a. （選用）從 AWS NVMe 版本 開始`1.5.0`，Windows Server 2016 和更新版本支援小型電腦系統界面 (SCSI) 持久性保留。此功能新增對具有共享 Amazon EBS 儲存體的 Windows Server 容錯移轉叢集的支援。依預設，安裝期間系統不會啟用此功能。

      您可以在執行 `install.ps1` 指令碼以安裝驅動程式時啟用此功能，方法是將 `EnableSCSIPersistentReservations` 參數指定為 `$true` 值。

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $true
```

您可以在執行 install.ps1 指令碼以安裝驅動程式時停用此功能，方法是將 EnableSCSIPersistentReservations 參數指定為 $false 值。

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $false
```

b. 從 AWS NVMe 開始1.5.0，install.ps1指令碼一律使用驅動程式安裝ebsnvme-id工具。

(選用) 對於版本 1.4.0、1.4.1 和 1.4.2，install.ps1 指令碼允許您指定 ebsnvme-id 工具是否應與驅動程式一起安裝。

i. 若要安裝 ebsnvme-id 工具，請指定 InstallEBSNVMeIdTool 'Yes'。

ii. 如果您不想安裝工具，請指定 InstallEBSNVMeIdTool 'No'。

如果您未指定 InstallEBSNVMeIdTool，並且工具已存在於 C:\ProgramData\Amazon\Tools，則套件預設會升級此工具。如果工具不存在，install.ps1 預設不會升級此工具。

如果您不想將工具安裝為套件的一部分，而且想要稍後安裝它，則可以在驅動程式套件中找到最新版本或工具。或者，您可以從 Amazon S3 下載版本 1.0.0：

下載 ebsnvme-id 工具。

5. 如果安裝程式並未重新啟動您的執行個體，請重新啟動執行個體。

Distributor

您可以使用 Distributor 的功能 AWS Systems Manager，安裝 NVMe 驅動程式套件一次，或搭配排定的更新。

安裝 latest AWS NVMe 驅動程式

1. 如需如何使用 Distributor 安裝 NVMe 驅動程式套件的指示，請參閱 Amazon EC2 Systems Manager 使用者指南中的安裝或更新套件。
2. 對於安裝類型，選取解除安裝並重新安裝。
3. 對於名稱，選擇 AWSNVMe。

4. (選用) 對於其他引數，您可指定值來自訂安裝。值必須使用有效 JSON 語法進行格式化。如需如何傳遞 aws configure 套件額外引數的範例，請參閱命令文件外掛程式參考。

   a. 從 AWS NVMe 開始1.5.0，驅動程式支援 Windows Server 2016 和更新版本的 SCSI 持久性保留。依預設，安裝期間系統不會啟用此功能。

      - 若要啟用此功能，請指定 {"SSM_EnableSCSIPersistentReservations": "true"}。
      - 若您不想啟用此功能，請指定 {"SSM_EnableSCSIPersistentReservations": "false"}。

   b. 從 AWS NVMe 開始1.5.0，install.ps1指令碼將一律安裝ebsnvme-id工具。

      (選用) 對於版本 1.4.0、1.4.1 和 1.4.2，install.ps1 指令碼允許您指定 ebsnvme-id 工具是否應與驅動程式一起安裝。

      - 若要安裝 ebsnvme-id 工具，請指定 {"SSM_InstallEBSNVMeIdTool": "Yes"}。
      - 如果您不想安裝工具，請指定 {"SSM_InstallEBSNVMeIdTool": "No"}。

        如果您未針對 Additional Arguments (其他引數) 指定 SSM_InstallEBSNVMeIdTool，並且工具已存在於 C:\ProgramData\Amazon\Tools，則套件預設會升級此工具。如果工具不存在，套件預設不會升級此工具。

        如果您不想將工具安裝為套件的一部分，而且想要稍後安裝它，則可以在驅動程式套件中找到最新版本的工具。或者，您可以從 Amazon S3 下載版本 1.0.0：

        下載 ebsnvme-id 工具。

5. 如果安裝程式並未重新啟動您的執行個體，請重新啟動執行個體。

為 Windows 執行個體設定 SCSI 持續保留

已安裝 After AWS NVMe 驅動程式版本 1.5.0或更新版本後，您可以使用 Windows Server 2016 和更新版本的 Windows 登錄來啟用或停用 SCSI 持久性保留。您必須重新啟動這些登錄檔的執行個體，變更才會生效。

您可以使用以下命令 (將 EnableSCSIPersistentReservations 設定為值 1)，啟用 SCSI 持續保留。

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
```

```
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 1
```

您可以使用以下命令 (將 EnableSCSIPersistentReservations 設定為值 0)，停用 SCSI 持續保留。

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 0
```

AWS NVMe Windows 驅動程式版本歷史記錄

下表顯示哪些 AWS NVMe 驅動程式會在 Amazon EC2 上的每個 Windows Server 版本上執行。

Windows Server 版本	AWS NVMe 驅動程式版本
Windows Server 2025	最新版本
Windows Server 2022	最新版本
Windows Server 2019	最新版本
Windows Server 2016	最新版本
Windows Server 2012 R2	1.5.1 版和較舊版本
Windows Server 2012	1.5.1 版和較舊版本
Windows Server 2008 R2	1.3.2 版和較舊版本
Windows Server 2008	1.3.2 版和較舊版本

下表說明 AWS NVMe 驅動程式的發行版本。

套件版本	驅動程式版本	詳細資訊	發行日期
1.8.2	1.8.2	• 透過改善中斷處理，修正啟動時的錯誤檢查。	2026 年 6 月 15 日

套件版本	驅動程式版本	詳細資訊	發行日期
1.8.1	1.8.1	• 修正磁碟序號產生的回溯相容性問題。	2026 年 2 月 24 日
1.8.0	1.8.0	• 新增支援 IOCTLs以使用實作動態命名空間管理的裝置。如需詳細資訊，請參閱 Microsoft 文件中的 IOCTL_SCSI_MINIPORT IOCTL。 • Get Log Page 命令和序號產生中的錯誤修正。	2026 年 1 月 16 日
1.7.0	1.7.0	• 新增了對 NVMe Get Log Page 命令的支援。 • 新增了對 EBS 與 EC2 執行個體儲存體磁碟區詳細效能統計資料的支援。 • 確定命名空間命令的錯誤修正。	2025 年 9 月 17 日
1.6.0	1.6.0	• 已更新安裝指令碼以使用 PnPUtil。 • 已更新 ebsnvme-id.exe 以使用 NVMe IOCTL。	2024 年 10 月 25 日
1.5.1	1.5.0	修正安裝指令碼以為 `ebsnvme-id` 工具建立資料夾 (如果不存在)。	2023 年 11 月 17 日
1.5.0	1.5.0	已新增對執行 Windows Server 2016 及更新版本之執行個體的 Small Computer System Interface (SCSI) 持續保留的支援。ebsnvme-id 工具 (`ebsnvme-id.exe`) 現在預設安裝。	2023 年 8 月 31 日
1.4.2	1.4.2	已修正 AWS NVMe 驅動程式 不支援 D3 執行個體上執行個體存放磁碟區的錯誤。	2023 年 3 月 16 日

套件版本	驅動程式版本	詳細資訊	發行日期
1.4.1	1.4.1	針對支援命名空間偏好寫入精細程度 (NPGW) 這項選用性 NVMe 功能的 EBS 磁碟區，報告命名空間偏好寫入精細程度。有關詳細資訊，請參閱 NVMe Base Specification, version 1.4 (NVMe 基本規格 1.4 版) 中的第 8.25 節 "Improving Performance through I/O Size and Alignment Adherence" (透過 I/O 大小和匹配依從性提高效能)。	2022 年 5 月 20 日
1.4.0	1.4.0	• 已新增對 iOCTLS 的支援，允許應用程式與 NVMe 設備互動。此支援可讓應用程式從 NVMe 設備中取得 `IdentifyController` 、`IdentifyNamespace` 和 `NameSpace` 清單。如需詳細資訊，請參閱 Microsoft 文件中的通訊協定特定查詢。 • 1.4.0 驅動程式版本和最新的 `ebsnvme-id` 工具 (`ebsnvme-id.exe`) 會組合在單一套件中。這種組合可讓您從單一套件同時安裝驅動程式和工具。如需詳細資訊，請參閱NVMe 驅動程式。 • 錯誤修正與可靠性改進。	2021 年 11 月 23 日
1.3.2	1.3.2	修正主動處理 IO 之 EBS 磁碟區時可能導致資料毀損的問題。未修改線上 EBS 磁碟區 (例如，調整大小或變更類型) 的客戶不受影響。 此版本為可在 Windows Server 2008 與 2008 R2 上執行的最後版本。此版本可供下載，但不再受支援。Windows Server 2008 與 2008 R2 已終止使用，且 Microsoft 不再提供支援。	2019 年 9 月 10 日
1.3.1	1.3.1	可靠性改善。	2019 年 5 月 21 日

套件版本	驅動程式版本	詳細資訊	發行日期
1.3.0	1.3.0	裝置最佳化改善。	2018 年 8 月 31 日
1.2.0	1.2.0	在所有支援的執行個體上改善 AWS NVMe 裝置的效能和可靠性，包括裸機執行個體。	2018 年 6 月 13 日
>1.0.0	>1.0.0	執行 Windows Server 的支援執行個體類型的AWS NVMe 驅動程式。	2018 年 2 月 12 日

訂閱 通知

Amazon SNS 會在 EC2 Windows 驅動程式發佈新版本時通知您。使用下列程序訂閱這些通知。

從主控台訂閱 EC2 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須選取此區域，因為您要訂閱的 SNS 通知位於本區域中。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選擇 Create subscription (建立訂閱)。
5. 在 Create subscription (建立訂閱) 對話方塊中，執行下列動作：
   a. 針對 TopicARN，複製下列 Amazon Resource Name (ARN)：

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 針對 Protocol (通訊協定)，選擇 `Email`。
   c. 針對 Endpoint (端點)，輸入可用於接收通知的電子郵件地址。
   d. 選擇 Create subscription (建立訂閱)。
6. 您將會收到確認電子郵件。開啟電子郵件並遵循指示完成訂閱。

只要有新的 EC2 Windows 驅動程式發行，我們就會向訂閱者傳送通知。如果您不想再接收這些通知，請使用下列程序來取消訂閱。

取消訂閱 Amazon EC2 Windows 驅動程式通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 在導覽窗格中，選擇 Subscriptions (訂閱)。
3. 選取訂閱的核取方塊，然後選擇 Actions (動作)、Delete subscriptions (刪除訂閱)。出現確認提示時，請選擇 Delete (刪除)。

使用 訂閱 EC2 通知 AWS CLI

若要使用 訂閱 EC2 通知 AWS CLI，請使用下列命令。

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --
protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

使用 訂閱 EC2 通知 AWS Tools for Windows PowerShell

若要使用 訂閱 EC2 通知 AWS Tools for Windows PowerShell，請使用下列命令。

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-
drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

其他 Windows 裝置驅動程式

AWS VMClock 裝置是一種虛擬時鐘裝置，存在於 Nitro 型執行個體上。裝置會以 的形式出現在系統裝置下的 Device Manager 中`AWS VMClock Device`。 AWS 提供適用於 Windows 的 null （無函數）驅動程式套件，可讓作業系統辨識裝置。

支援的執行個體

AWS VMClock 裝置可在 Nitro 型執行個體類型上使用。如需 Nitro 型執行個體類型的清單，請參閱 Nitro 型執行個體。

支援的作業系統

驅動程式套件支援下列版本的 Windows Server：

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

- Windows Server 2025

驅動程式安裝

如果 AWS VMClock 裝置在 Device Manager 中顯示警告圖示，表示未安裝驅動程式，您可以手動安裝 null 驅動程式套件。

安裝 AWS VMClock 驅動程式

1. 從 Amazon S3 下載並擷取驅動程式套件。

```
Invoke-WebRequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/
AWSVMClock/Latest/AWSVMClock.zip -OutFile $env:USERPROFILE\AWSVMClock.zip
Expand-Archive $env:USERPROFILE\AWSVMClock.zip -DestinationPath $env:USERPROFILE
\AWSVMClock
```

2. 使用 安裝驅動程式pnputil。

```
pnputil /add-driver $env:USERPROFILE\AWSVMClock\aws_vmclock.inf /install
```

3. 在 Device Manager 中驗證裝置狀態。您不需要重新啟動執行個體。

設定您的 Amazon EC2 Windows 執行個體

啟動 Windows 執行個體之後，您即可以管理員身分登入，以執行 Windows 功能和系統設定的其他組態。EC2 Windows 疑難排解公用程式 可協助您對執行個體的問題進行疑難排解。

您可以設定 Windows 啟動代理程式和其他 Windows 特定的功能，如下所示。

Windows 啟動代理程式

每個 AWS Windows AMI （以及 上提供的許多其他 AMIs AWS Marketplace) 都包含預先設定預設設定的 Windows 啟動代理程式。啟動代理程式會在執行個體啟動期間執行任務，並在執行個體停止並稍後啟動或重新啟動時執行。

EC2 Fast Launch for Windows

每個 Amazon EC2 Windows 執行個體都必須經過標準的 Windows 作業系統 (OS) 啟動步驟，其中包括多次重新啟動，通常需要 15 分鐘或更長時間才能完成。已啟用 EC2 快速啟動功能的 Amazon EC2 Windows Server AMI 會完成其中一些步驟並預先重新開機，以減少啟動執行個體所需的時間。

Windows 特定的系統設定

下列清單包含一些僅適用於 Windows 作業系統的系統設定：

變更 Windows 管理員密碼

當您連線至 Windows 執行個體時，您必須指定使用者帳戶和有存取執行個體許可的密碼。您第一次連線至執行個體時，必須使用管理員帳戶並提供預設密碼。當您第一次連線至執行個體時，建議您變更預設的管理員密碼。

新增 Windows 系統元件

Windows Server 作業系統包含許多選用元件。在每個 AWS Windows Server AMI 中包含所有選用元件並不實際。反之，我們會提供安裝媒體 EBS 快照，其中包含在您的 Windows 執行個體上設定或安裝元件所需要的檔案。

在 Windows 安裝 WSL

Windows Subsystem for Linux (WSL) 是免費下載軟體，可以安裝在 Windows 執行個體上。透過安裝 WSL，您可以直接在 Windows 執行個體上執行原生 Linux 命令列工具，並使用 Linux 工具與傳統 Windows 桌面來編寫指令碼。您可以在單一 Windows 執行個體上輕鬆地在 Linux 和 Windows 之間進行切換，這在開發環境中可能很有用。

AWS Windows 執行個體的裝置驅動程式

您可以更新 Windows 執行個體 AWS 的裝置驅動程式。如需詳細資訊，請參閱the section called “管理裝置驅動程式”。

下表摘要說明 Windows 版 Nitro 型執行個體支援的驅動程式。

版本	儲存體驅動程式	增強型網路驅動程式
Windows Server 2025	AWS NVMe 最新版本	ENA 最新版本
Windows Server 2022	AWS NVMe 最新版本	ENA 最新版本
Windows Server 2019	AWS NVMe 最新版本	ENA 最新版本
Windows Server 2016	AWS NVMe 最新版本	ENA 最新版本
Windows Server 2012 R2	AWS NVMe 1.5.1 版	ENA 2.6.0 版

版本	儲存體驅動程式	增強型網路驅動程式
Windows Server 2008 R2	AWS NVMe 1.3.2 版	ENA 2.2.3 版

下表摘要說明 Windows 版 Xen 型執行個體支援的驅動程式。

版本	儲存體驅動程式	增強型網路驅動程式
Windows Server 2022	AWS PV 最新版本	• ENA 最新版本 [1] • Intel VF [2] • AWS PV 最新版本 [3]
Windows Server 2019	AWS PV 最新版本	• ENA 最新版本 [1] • Intel VF [2] • AWS PV 最新版本 [3]
Windows Server 2016	AWS PV 最新版本	• ENA 最新版本 [1] • Intel VF [2] • AWS PV 最新版本 [3]
Windows Server 2012 R2	AWS PV 8.4.3 版	• ENA 2.6.0 版 [1] • Intel VF [2] • AWS PV 版本 8.4.3 [3]
Windows Server 2008 R2	AWS PV 8.3.5 版	• ENA 2.2.3 版 [1] • Intel VF [2] • AWS PV 版本 8.3.5 [3]

[1] 針對執行個體類型 G3、H1、I3、`m4.16xlarge`、P3、P3dn 和 R4。

[2] 針對 C3、C4、D2、I2、M4 (不含 `m4.16xlarge`) 和 R3 執行個體類型。

[3] 針對 C1、M1、M2、M3、T1、T2、X1 和 X1e 執行個體類型。

Amazon EC2 Windows 執行個體的 Windows 啟動代理程式

每個 AWS Windows AMI 都包含預先設定預設設定的 Windows 啟動代理程式。啟動代理程式會在執行個體啟動期間執行任務，並在執行個體停止並稍後啟動或重新啟動時執行。如需特定代理程式的詳細資訊，請參閱下列清單的詳細資訊頁面。

如需 AWS Windows AMIs的詳細資訊，請參閱 AWS Windows AMI 參考。

- 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v2 代理程式執行任務
- 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v1 代理程式執行任務
- 在 EC2 舊版 Windows 作業系統執行個體啟動期間，使用 EC2Config 服務執行任務

內容

- 比較 Amazon EC2 啟動代理程式
- 設定 EC2 Windows 啟動代理程式的 DNS 尾碼
- 訂閱 EC2 Windows 啟動代理程式通知
- EC2Launch v2 和 EC2Config 代理程式的 Windows 服務管理

比較 Amazon EC2 啟動代理程式

下表顯示了 EC2Config、EC2Launch v1 和 EC2Launch v2 之間的主要功能差異。

功能	EC2Config	EC2Launch v1	EC2Launch v2
Run as (執行身分)	Windows 服務	PowerShell 指令碼	Windows 服務
支援	僅限舊版作業系統	Windows Server 版本： • 2016 • 2019 (LTSC 與 SAC)	Windows Server 版本： • 2016 • 2019 (LTSC 與 SAC) • 2022

功能	EC2Config	EC2Launch v1	EC2Launch v2
			• 2025
組態檔案	XML	JSON	JSON/YAML
設定管理員使用者名稱	否	否	是
壓縮版使用者資料	否	否	是
在 AMI 上預製的本機使用者資料	否	否	是，不可設定。
使用者資料中的任務組態	否	否	是
可設定桌布	否	否	是
自訂任務執行順序	否	否	是
可設定的工作	15	9	啟動時為 20 個
支援 Windows 事件檢視器	是	否	是
事件檢視器事件類型的數目	2	0	30

> **Note**
>
> EC2Config 文件為歷史資訊，僅供參考。Microsoft 不再支援其執行的作業系統版本。我們強烈建議您升級至最新的啟動服務。

設定 EC2 Windows 啟動代理程式的 DNS 尾碼

使用 Amazon EC2 啟動代理程式，您可以設定 Windows 執行個體用於網域名稱解析的 DNS 尾碼清單。啟動代理程式會將下列值新增至 DNS 尾碼搜尋清單，以覆寫 `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` 登錄機碼的標準 Windows 設定：

- 執行個體的網域。
- 因執行個體網域的轉移所導致的尾碼
- NV 網域
- 每個網路介面卡指定的網域

所有啟動代理程式都支援 DNS 尾碼組態。如需詳細資訊，請參閱特定啟動代理程式版本：

- 如需 `setDnsSuffix` 任務以及如何在 EC2Launch v2 設定 DNS 尾碼的詳細資訊，請參閱 setDnsSuffix。
- 如需 DNS 尾碼清單設定以及如何啟用或停用 EC2Launch v1 轉移相關資訊，請參閱 在 Windows 執行個體設定 EC2Launch v1 代理程式。
- 如需 DNS 尾碼清單設定以及如何啟用或停用 EC2Config 轉移相關資訊，請參閱 EC2Config 設定檔。

網域名稱轉移

網域名稱轉移是一種 Active Directory 行為，允許子網域的電腦存取父網域的資源，而無需使用完整網域名稱。根據預設，網域名稱轉移會繼續進行，直到網域名稱進展只剩下兩個節點為止。

如果執行個體已連線至網域，啟動代理程式會對網域名稱執行轉移，並將結果新增至 **`System\CurrentControlSet\Services\Tcpip\Parameters\SearchList`** 登錄機碼維護的 DNS 尾碼搜尋清單。代理程式使用下列登錄機碼的設定來判斷轉移行為。

- **`System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution`**
  - 未設定時，將停用轉移
  - 設為 `1` 時，將啟用轉移 (預設)
  - 設為 `0` 時，會停用轉移
- **`System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel`**
  - 未設定時，請使用 2 層級 (預設)
  - 設為 3 或更高時，請用值來設定層級

當停用轉移或將轉移設定變更為更高層級時，System\CurrentControlSet\Services\Tcpip\Parameters\SearchList 登錄機碼仍包含先前新增的尾碼。它們不會自動移除。您可以手動更新清單，也可以清除清單，讓代理程式執行設定新清單的程序。

> **Note**
>
> 若要從登錄檔清除 DNS 尾碼清單，您可以執行下列命令。
>
> ```
> PS C:\> Invoke-CimMethod -ClassName Win32_NetworkAdapterConfiguration -
> MethodName "SetDNSSuffixSearchOrder" -Arguments @{ DNSDomainSuffixSearchOrder =
>  $null } | Out-Null
> ```

轉移範例

下列範例顯示網域名稱在轉移過程的進展。

corp.example.com

- 進展至 example.com

locale.region.corp.example.com

1. 進展至 region.corp.example.com
2. 進展至 corp.example.com
3. 進展至 example.com

locale.region.corp.example.com 設定為 DomainNameDevolutionLevel=3

1. 進展至 region.corp.example.com
2. 進展至 corp.example.com。由於層級設定，進展在此處停止。

訂閱 EC2 Windows 啟動代理程式通知

當新版本的 EC2 啟動代理程式發佈時，Amazon SNS 會通知您。使用下列程序訂閱這些通知。

訂閱 EC2Config 通知

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 如有必要，請在導覽列中將「區域」變更為美國東部 (維吉尼亞北部)。您必須選取此區域，因為您要訂閱的 SNS 通知已在本區域中建立完成。
3. 在導覽窗格中，選擇 Subscriptions (訂閱)。
4. 選擇 Create subscription (建立訂閱)。
5. 在 Create subscription (建立訂閱) 對話方塊中，執行下列動作：
   a. 針對主題 ARN，請使用以下與您希望接收通知的代理程式相符的 Amazon Resource Name (ARN)：
      - EC2Launch v2：

        ```
        arn:aws:sns:us-east-1:309726204594:amazon-ec2launch-v2
        ```

      - EC2Launch 或 EC2Config：

        ```
        arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
        ```

   b. 針對 rotocol (通訊協定)，選擇 `Email`。
   c. 針對端點，輸入您想要接收通知的電子郵件地址。
   d. 選擇 Create subscription (建立訂閱)。
6. 您會收到要求確認訂閱的電子郵件。開啟電子郵件並遵循指示完成訂閱。

每當發佈新版本的啟動代理程式時，我們都會向訂閱者發送通知。如果您不想再接收這些通知，請使用下列程序來取消訂閱。

取消訂閱啟動代理通知

1. 開啟 Amazon SNS 主控台。
2. 在導覽窗格中，選擇 Subscriptions (訂閱)。
3. 選取訂閱，然後選取 Actions (動作)、Delete subscriptions (刪除訂閱)。出現確認提示時，請選擇 Delete (刪除)。

EC2Launch v2 和 EC2Config 代理程式的 Windows 服務管理

如果您以具有管理權限的使用者身分登入執行個體，您可以像任何其他 Windows 服務一樣管理 EC2Launch v2 和 EC2Config 啟動代理程式。EC2Launch v1 是一組 PowerShell 指令碼，預設會透過排程任務進行管理。本節涵蓋 EC2Launch v2 和 EC2Config 的服務管理。

若要將更新的設定套用至執行個體，您可以從服務的 Microsoft 管理主控台 (MMC) 介面停止並重新啟動 EC2Launch v2 代理程式或 EC2Config 服務啟動代理程式。同樣地，當您安裝新版本的啟動代理程式時，必須先停止代理程式，然後在安裝完成時重新啟動。

> **Note**
>
> 您必須以管理員身分開啟 MMC 服務介面，才能選取這些動作。若要執行此動作，您可以從內容功能表選取以管理員身分執行。或者，若要使用鍵盤開啟介面，請依照下列步驟進行：
>
> 1. 使用 Tab 鍵或方向鍵，從管理工具功能表選取服務功能表項目。
> 2. 使用以下鍵盤組合以管理員身分開啟：Ctrl + Shift + Enter。

下列程序列出在執行個體停止及啟動啟動代理程式的步驟。

停止啟動代理程式

1. 啟動和連線至 Windows 執行個體。
2. 從 Windows 開始功能表選取管理工具。
3. 如本節開頭所述，以管理員身分開啟服務主控台。
4. 在服務清單，選取執行個體執行的代理程式 (EC2Launch 或 EC2Config)，然後從動作功能表選擇停止。或者，您可以使用內容功能表來停止代理程式。

重新啟動啟動代理

1. 啟動和連線至 Windows 執行個體。
2. 從 Windows 開始功能表選取管理工具。
3. 如本節開頭所述，以管理員身分開啟服務主控台。
4. 在服務清單，選取在執行個體執行的代理程式 (EC2Launch 或 EC2Config)，然後從動作功能表選擇開始或重新啟動。或者，您可以使用內容功能表重新啟動代理程式。

如果您不需要更新組態設定、建立自己的 AMI 或使用 AWS Systems Manager，則可以刪除或解除安裝啟動代理程式。

刪除

移除服務將移除其登錄子機碼。

解除安裝

解除安裝服務將移除檔案、登錄子機碼和任何服務捷徑。

刪除啟動代理程式

1. 啟動和連線至 Windows 執行個體。
2. 啟動 Windows 命令提示字元視窗。
3. 執行下列其中一個命令來刪除啟動代理程式。
   - 執行下列命令來刪除 EC2Launch 或 EC2Launch v2：

     ```
     sc delete ec2launch
     ```

   - 執行下列命令來刪除 EC2Config 服務：

     ```
     sc delete ec2config
     ```

解除安裝啟動代理程式。

1. 啟動和連線至 Windows 執行個體。
2. 從 Windows 開始功能表，依序選取 Windows 系統和控制台。
3. 選擇程式和功能以開啟安裝在執行個體的程式清單。
4. 從清單選取啟動代理程式 (Amazon EC2Launch 或 EC2ConfigService)，然後從檔案功能表選擇解除安裝。或者，您可以使用內容功能表。

> **Note**
>
> 您可以在版本欄位查看已安裝的啟動代理程式版本。

在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v2 代理程式執行任務

根據預設，從 AWS Windows Server 2022 和 Windows Server 2025 AMI 啟動的所有支援 Amazon EC2 執行個體都包含 EC2Launch v2 啟動代理程式 (`EC2Launch.exe`)。 AMIs 我們亦提供將 EC2Launch v2 安裝為預設啟動代理程式的 Windows Server 2016 和 2019 AMI。除了包含 EC2Launch v1 的 Windows Server 2016 和 2019 AMI 之外，亦提供了這些 AMI。您可以從 Amazon EC2 主控台中的 AMI 頁面，在搜尋中輸入以下字首來搜尋預設包含 EC2Launch v2 的 Windows AMI：`EC2LaunchV2-Windows_Server-*`。

若要比較啟動代理程式版本功能，請參閱 比較 Amazon EC2 啟動代理程式。

EC2Launch v2 在執行個體啟動期間執行任務，並在執行個體停止並稍後啟動或重新啟動時執行。EC2Launch v2 也可依據需求執行任務。其中部分作業為自動啟用，其他作業則必須手動啟用。EC2Launch v2 服務支援所有 EC2Config 和 EC2Launch 功能。

此服務會使用組態檔來控制其操作。您可以使用圖形化工具或直接將組態檔編輯為單一 .yml 檔案 (`agent-config.yml`) 來更新組態檔。若要了解關於日誌檔案位置的詳細資訊，請參閱 EC2Launch v2 目錄結構。

EC2Launch v2 會發佈 Windows 事件日誌，協助您疑難排解錯誤並設定觸發程序。如需詳細資訊，請參閱Windows 事件日誌。

支援的作業系統版本

EC2Launch v2 代理程式支援下列 Windows Server 作業系統 (OS) 版本：

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019 (長期服務通道和半年通道)
- Windows Server 2016

預設會執行的任務

EC2Launch v2 代理程式在初始執行個體啟動期間，預設僅會執行以下任務一次。依據任務在其啟動階段執行的順序對其進行整理。

`Boot` 階段

- extendRootPartition

`PreReady` 階段

- activateWindows
- setDnsSuffix
- setAdminAccount
- setWallpaper

`PostReady` 階段

- startSsm

EC2Launch v2 概念

考慮 EC2Launch v2 時，下列概念有助於理解。

agent-config

`agent-config` 是位於 EC2Launch v2 組態資料夾中的檔案。包括開機、網路、預備就緒和後備就緒階段的組態。此檔案用於指定執行個體組態，設定應在 AMI 第一次啟動或後續啟動時執行的任務。

依預設，EC2Launch v2 安裝會安裝一個 `agent-config` 檔案，它包含標準 Amazon Windows AMI 中使用的推薦組態。您可以更新該組態檔案，以變更 EC2Launch v2 指定的 AMI 的預設開機經驗。若要了解關於日誌檔案位置的詳細資訊，請參閱 EC2Launch v2 目錄結構。

Frequency (頻率)

任務頻率確定何時執行任務，這取決於開機內容。大多數任務只有一個允許的頻率。您可以指定 `executeScript` 任務的頻率。

您將在 EC2Launch v2 任務組態 中看到以下頻率。

- 一次 – 當 AMI 已初次啟動 (完成 Sysprep)，執行任務一次。
- 始終 – 每次啟動代理程式執行時，任務都會執行。啟動代理程式會執行於下列時間：
  - 執行個體啟動或重新啟動
  - 執行 EC2Launch 服務
  - `EC2Launch.exe run` 被叫用

階段

階段是 EC2Launch v2 代理程式所執行任務的邏輯群組。某些工作只能在特定階段中執行。其他人可以在多個階段運行。使用 `agent-config.yml` 時，必須指定階段清單，以及要在每個階段中執行的任務清單。

此服務會以下列順序執行各階段：

階段 1：開機

階段 2：網路

階段 3：預備就緒

Windows 已準備就緒

PreReady 階段完成後，服務會將 `Windows is ready` 訊息傳送到 Amazon EC2 主控台。

階段 4：後備就緒

使用者資料會在 PostReady 階段執行。部分指令碼版本會在 `agent-config.yml` 檔案後備就緒階段之前執行，有些則會在之後執行，如下所示：

`agent-config.yml` 之前

- YAML 使用者資料版本 1.1
- XML 使用者資料

`agent-config.yml` 之後

- YAML 使用者資料 1.0 版 (舊版供回溯相容性之用)

如需階段和任務的範例，請參閱 範例：`agent-config.yml`。

使用使用者資料時，您必須指定代理程式要執行的任務清單。此階段是隱含的。如需任務範例，請參閱 範例：使用者資料。

EC2Launch v2 會依照您在 `agent-config.yml` 和使用者資料中指定的順序來執行任務清單。階段按順序執行。下一個階段會在前一個階段完成之後開始。任務也會依序執行。

任務

您可以調用任務以在執行個體上執行動作。您可以在 `agent-config.yml` 檔案中或透過使用者資料來設定任務。如需 EC2Launch v2 可用任務的清單，請參閱 EC2Launch v2 任務。如需了解任務組態結構描述和詳細資訊，請參閱 EC2Launch v2 任務組態。

使用者資料

使用者資料是啟動執行個體時可設定的資料。您可以更新使用者資料，以動態變更自訂 AMI 或快速入門 AMI 的設定方式。EC2Launch v2 支援 60 kB 的使用者資料輸入長度。使用者資料只包含 UserData 階段，因此在 `agent-config` 檔案之後執行。您可以在使用啟動執行個體精靈啟動執行個體時輸入使用者資料，也可以從 EC2 主控台修改使用者資料。如需處理使用者資料的資訊，請參閱 Amazon EC2 如何處理 Windows 執行個體的使用者資料。

EC2Launch v2 任務概觀

EC2Launch v2 可以在每次開機時執行下列任務：

- 設定新的和選擇性的自訂桌布來呈現執行個體的相關資訊。
- 設定在本機電腦上建立的系統管理員帳戶的屬性。
- 將 DNS 尾碼新增至搜尋尾碼清單。只有不存在的尾碼才會加入到清單中。
- 為任何額外的磁碟區設定磁碟機代號，並將它們延伸至使用可用空間。
- 將檔案從組態寫入磁碟。
- 執行 EC2Launch v2 組態檔案指定的指令碼，或從 `user-data` 執行指令碼。`user-data` 的指令碼可以是純文字或壓縮，並以 base64 格式提供。
- 執行具有指定參數的程式。
- 設定電腦名稱。
- 將執行個體資訊傳送到 Amazon EC2 主控台。
- 傳送 RDP 憑證指紋到 Amazon EC2 主控台。
- 動態擴充作業系統的分割區，以包含任何尚未分割的空間。
- 執行使用者資料。如需指定使用者資料的詳細資訊，請參閱EC2Launch v2 任務組態。
- 設定非持久性靜態路由以到達中繼資料服務和 AWS KMS 伺服器。
- 將非開機磁碟分割設定為 `mbr` 或 `gpt`。
- 在 Sysprep 後啟動 Systems Manager 服務。
- 最佳化 ENA 設定。
- 為更新的 Windows 版本啟用開啟 SSH。
- 啟用巨型訊框。
- 將 Sysprep 設定為在 EC2Launch v2 中執行。

- 發佈 Windows 事件日誌。

EC2Launch v2 目錄結構

EC2Launch v2 應該安裝在以下目錄中：

- 服務二進位檔案： `%ProgramFiles%\Amazon\EC2Launch`
- 服務資料 (設定、日誌檔和狀態檔)：`%ProgramData%\Amazon\EC2Launch`

> **Note**
>
> 根據預設，Windows 會將檔案和資料夾隱藏在 `C:\ProgramData` 下。若要檢視 EC2Launch v2 目錄和檔案，您必須在 Windows 檔案總管中輸入路徑或變更資料夾屬性，才可顯示隱藏的檔案和資料夾。

`%ProgramFiles%\Amazon\EC2Launch` 目錄包含二進位檔案和支援的程式庫。它包括以下子目錄：

- `settings`
  - `EC2LaunchSettingsUI.exe` – 用於修改 `agent-config.yml` 檔案的使用者介面
  - `YamlDotNet.dll` – 用於支援使用者介面中的某些操作的 DLL
- `tools`
  - `ebsnvme-id.exe` – 用於檢查執行個體上 EBS 磁碟區工具
  - `AWSAcpiSpcrReader.exe` – 用於確定要使用的正確 COM 連接埠工具
  - `EC2LaunchEventMessage.dll` – 用於支援 EC2Launch 的 Windows 事件日誌記錄的 DLL
- `service`
  - `EC2LaunchService.exe` – 啟動代理程式以服務方式執行時，所啟動的 Windows 服務可執行檔
- `EC2AgentTelemetry.dll` – 用於支援 EC2 代理程式遙測的 DLL
- `EC2Launch.exe` – 主 EC2Launch 可執行檔
- `EC2LaunchAgentAttribution.txt` – EC2 Launch 中使用的代碼屬性

`%ProgramData%\Amazon\EC2Launch` 目錄包含下列子目錄。服務產生的所有資料，包括日誌、組態和狀態，都會儲存在此目錄中。

- `config` – 組態

服務組態檔案儲存在此目錄中為 agent-config.yml。此檔案可以更新，以修改、新增或移除服務所執行的預設任務。在此目錄中建立檔案的許可限制為系統管理員帳戶，以防止權限提升。

- log – 執行個體日誌

  服務 (agent.log) 和遙測 (telemetry.log) 的日誌會存放在此目錄中。當 agent.log 達到 1 MB 大小時，會自動輪換，並使用時間戳記格式建立備份檔案 （ 例如 agent-2026-03-02T18-56-39.188.log)。一次只會維護一個備份日誌檔案。

- state – 服務狀態資料

  服務用來判斷應該執行哪些任務的狀態會儲存在此處。有一個 .run-once 檔案會指出服務是否已在 Sysprep 之後執行 (因此，下次執行時會略過頻率為一次的任務)。此子目錄包含 state.json 和 previous-state.json 以追蹤每個任務的狀態。

- sysprep – Sysprep

  此目錄包含的檔案，可用來判斷 Sysprep 在建立可重複使用的自訂 Windows AMI 時要執行哪些作業。

- wallpaper – 桌布

  此桌布映像會存放在此目錄。

遙測

遙測是其他資訊，有助於 AWS 更了解您的需求、診斷問題和提供功能，以改善您使用 的體驗 AWS 服務。

EC2Launch v2 版本 2.1.592 及更新版本會收集遙測，例如用量指標和錯誤。這些資料是從執行 EC2Launch v2 的 Amazon EC2 執行個體中收集的資料。這包括 擁有的所有 Windows AMIs AWS。

EC2Launch v2 會收集下列類型的遙測：

- 用量資訊 – 代理程式命令、安裝方法和排程的執行頻率。
- 錯誤和診斷資訊 – 代理程式安裝錯誤代碼、執行錯誤代碼和錯誤呼叫堆疊。

從版本 2.0.592 到 2.1.1 所收集資料的範例：

```
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
```

```
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentCommandErrorCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentErrorLocation=addroutes.go:49
```

自版本 2.2.63 起，EC2 代理程式遙測資料將格式化為 JSON 物件：

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

預設為啟用遙測。您可隨時停用遙測收集。

停用執行個體的遙測

若要停用單一執行個體的遙測，您可以設定系統環境變數，或使用 MSI 來修改安裝。

若要藉由設定系統環境變數來停用遙測，請以管理員身分執行下列命令。

```
setx /M EC2LAUNCH_TELEMETRY 0
```

若要使用 MSI 來停用遙測，請在下載 MSI 後執行以下命令。

```
msiexec /i ".\AmazonEC2Launch.msi" Remove="Telemetry" /q
```

EC2Launch v2 的更多主題

- 安裝最新版本的 EC2Launch v2
- 設定 Windows 執行個體的 EC2Launch v2 設定
- EC2Launch v2 啟動任務的任務定義
- 疑難排解使用 EC2Launch v2 代理程式的問題
- EC2Launch v2 版本歷程記錄

安裝最新版本的 EC2Launch v2

您可使用以下其中一種方法，在 EC2 執行個體上安裝 EC2Launch v2 代理程式：

- 從 Amazon S3 下載代理程式，再使用 Windows PowerShell 進行安裝。如需下載 URL，請參閱 在 Amazon S3 上下載 EC2Launch v2。
- 透過 SSM Distributor 進行安裝。
- 當您建立自訂映像時，請從 EC2 Image Builder 元件安裝 。

- 從已預先安裝 EC2Launch v2 的 AMI 啟動執行個體。

> **Warning**
>
> AmazonEC2Launch.msi 會解除安裝先前版本的 EC2 啟動服務，例如 EC2Launch (v1) 和 EC2Config。

如需了解安裝步驟，請選取符合您偏好方法的索引標籤。

PowerShell

若要使用 Windows PowerShell 安裝最新版本的 EC2Launch v2 代理程式，請依照下列步驟執行。

1. 建立您的本機目錄。

```
New-Item -Path "$env:USERPROFILE\Desktop\EC2Launchv2" -ItemType Directory
```

2. 為您的下載位置設定 URL。使用您將使用的 Amazon S3 URL 執行以下命令。如需下載 URL，請參閱 在 Amazon S3 上下載 EC2Launch v2

```
$Url = "Amazon S3 URL/AmazonEC2Launch.msi"
```

3. 使用以下複合命令來下載和安裝代理程式：

```
$DownloadFile = "$env:USERPROFILE\Desktop\EC2Launchv2\" + $(Split-Path -Path
 $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadFile
msiexec /i "$DownloadFile"
```

> **Note**
>
> 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

4. 此 msiexec 命令會在 Windows Server 執行個體的下列位置安裝 EC2Launch v2：`%ProgramFiles%\Amazon\EC2Launch`。若要驗證安裝是否執行，您可以檢查執行個體上的本機檔案系統。

AWS Systems Manager Distributor

若要使用 AWS Systems Manager 快速設定來設定 EC2Launch v2 的自動更新，請參閱 使用 Distributor 快速設定自動安裝和更新。

您也可以從 AWS Systems Manager Distributor 執行一次性安裝 `AWSEC2Launch-Agent` 套件。如需有關如何從 Systems Manager Distributor 安裝套件的指示，請參閱「AWS Systems Manager 使用者指南」中的安裝或更新套件。

EC2 Image Builder component

您可以在使用 EC2 Image Builder 建置自訂映像時，安裝 `ec2launch-v2-windows` 元件。如需如何使用 EC2 Image Builder 建置自訂映像的指示，請參閱 EC2 Image Builder 使用者指南中的使用 EC2 Image Builder 主控台精靈建立映像管道。

AMI

根據預設，EC2Launch v2 預先安裝在適用於 Windows Server 2022 及更高作業系統的 AMI 上：

- Windows_Server-*version*-English-Full-Base
- Windows_Server-*version*-English-Core-Base
- Windows_Server-*version*-English-Core-EKS_Optimized
- 所有其他語言的 Windows Server *version* AMI
- 安裝 SQL 的 Windows Server *version* AMI

EC2Launch v2 也已預先安裝在下列 Windows Server AMI 上。您可以在 Amazon EC2 主控台找到這些 AMI，或使用下列搜尋字首： AWS CLI 中的 `EC2LaunchV2-`。

- EC2LaunchV2-Windows_Server-2019-English-Core-Base
- EC2LaunchV2-Windows_Server-2019-English-Full-Base
- EC2LaunchV2-Windows_Server-2016-English-Core-Base
- EC2LaunchV2-Windows_Server-2016-English-Full-Base

使用 AWS Systems Manager Distributor Quick Setup 自動安裝和更新 EC2Launch v2

透過 AWS Systems Manager Distributor Quick Setup，您可以設定 EC2Launch v2 的自動更新。下列流程會在您的執行個體上設定 Systems Manager Association，以您指定的頻率自動更新 EC2Launch v2 代理程式。Distributor Quick Setup 建立的關聯可以包含 AWS 帳戶 和 區域內的執行個體，或 AWS 組織內的執行個體。如需詳細資訊，請參閱「AWS Organizations 使用指南」中的教學課程：建立和設定組織。

開始之前，請確定您的執行個體符合所有先決條件。

先決條件

若要使用 Distributor 快速設定設定自動更新，您的執行個體必須符合下列先決條件。

- 您至少有一個執行中執行個體支援 EC2Launch v2。參閱 EC2Launch v2 的支援作業系統。
- 您已在執行個體上執行 Systems Manager 設定任務。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的設定工作階段管理員。
- EC2Launch v2 必須是唯一安裝在執行個體上的啟動代理程式。如果您安裝了多個啟動代理程式，您的 Distributor 快速設定組態將會失敗。使用 Distributor 快速設定設定 EC2Launch v2 之前，若系統中已安裝 EC2Config 或 EC2Launch v1 啟動代理程式，請先將其解除安裝。

設定 EC2Launch v2 的經銷商快速設定

若要使用 Distributor 快速設定建立 EC2Launch v2 的組態，請在完成 Distributor 套件部署的步驟時使用下列設定：

- 軟體套件：Amazon EC2Launch v2 代理程式。
- 更新頻率：從清單中選取頻率。
- 目標：從可用的部署選項中進行選擇。

若要檢查組態的狀態，請瀏覽至 AWS 管理主控台中的 Systems Manager 快速設定組態索引標籤。

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇快速設定。
3. 在組態索引標籤中，選取與您建立之組態相關聯的資料列。組態索引標籤會列出組態，並包含金鑰詳細資訊的摘要，例如區域、部署狀態和關聯狀態。

> **Note**
>
> 每個 EC2Launch v2 Distributor 組態的關聯名稱以下列前置詞開頭：`AWS-QuickSetup-Distributor-EC2Launch-Agent-`。

4. 若要檢視詳細資訊，請選取組態，然後選擇檢視詳細資訊。

如需詳細資訊和疑難排解步驟，請參閱「AWS Systems Manager 使用者指南」中的疑難排解快速設定結果。

在 Amazon S3 上下載 EC2Launch v2

若要安裝最新版本 EC2Launch v2，請從下列位置下載安裝程式：

- https://s3.amazonaws.com/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi

設定安裝選項

在安裝或升級 EC2Launch v2 時，您可以使用 EC2Launch v2 安裝對話方塊或命令列 shell 中的 msiexec 命令來設定安裝選項。

首次在執行個體上執行 EC2Launch v2 安裝程式時，該安裝程式會初始化執行個體上的啟動代理程式設定，如下所示：

- 安裝程式會建立本機路徑，並將啟動代理程式檔案寫入該路徑。這有時被稱為全新安裝。
- 安裝程式會建立 `EC2LAUNCH_TELEMETRY` 環境變數 (若尚無)，並根據您的組態進行設定。

如需有關設定的詳細資訊，請選取符合您將使用的設定方法的索引標籤。

Amazon EC2Launch Setup dialog

在安裝或升級 EC2Launch v2 時，您可以透過 EC2Launch v2 安裝對話方塊來設定以下安裝選項。

基本安裝選項

傳送遙測

在您將此功能納入安裝對話方塊中時，安裝程式會將 `EC2LAUNCH_TELEMETRY` 環境變數的值設定為 `1`。若您停用傳送遙測，安裝程式會將環境變數設定為 `0`。

在 EC2Launch v2 代理程式執行時，其會讀取 EC2LAUNCH_TELEMETRY 環境變數，決定是否要上傳遙測資料。若值等於 1，則會上傳資料。若不是，則不會上傳。

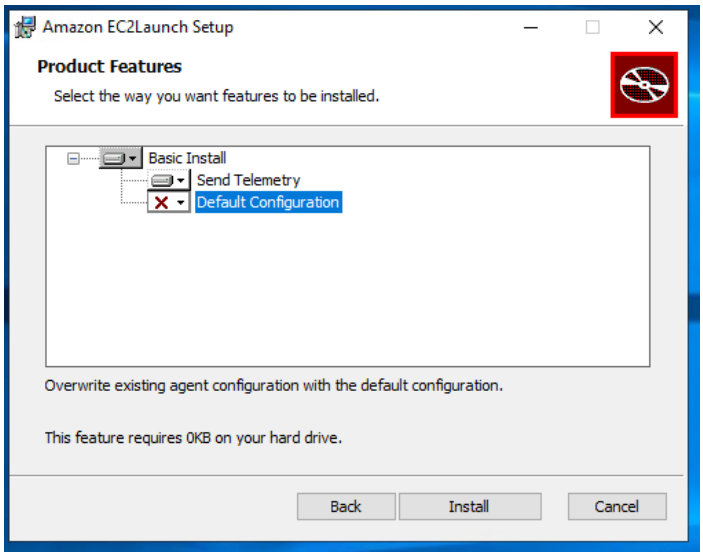
預設組態

EC2Launch v2 的預設組態為覆寫本機啟動代理程式 (若已具備)。首次在執行個體上執行安裝時，預設組態會執行全新安裝。若您在初始安裝時停用預設組態，則安裝會失敗。

若再次在執行個體上執行安裝，您可以停用預設組態，執行不會取代 `%ProgramData%/Amazon/EC2Launch/config/agent-config.yml` 檔案的升級。

範例：使用遙測來升級 EC2Launch v2

以下範例會顯示設定為升級目前安裝及啟用遙測的 EC2Launch v2 設定對話方塊。此組態會在不取代代理程式組態檔案的情況下執行安裝，並將 EC2LAUNCH_TELEMETRY 環境變數設定為 1。



Command line

在安裝或升級 EC2Launch v2 時，您可以使用命令列 shell 中的 msiexec 命令來設定以下安裝選項。

ADDLOCAL 參數值

基本 (必填)

安裝啟動代理程式。如果 ADDLOCAL 參數中不存在此值，則安裝結束。

全新

當您在 ADDLOCAL 參數中納入 Clean 值時，安裝程式會將代理程式組態檔案寫入以下位置：%ProgramData%/Amazon/EC2Launch/config/agent-config.yml。如果代理程式組態檔案已存在，則會覆寫檔案。

當您將 Clean 值放在 ADDLOCAL 參數之外時，安裝程式會執行不會取代代理程式組態檔案的升級。

遙測

當您在 ADDLOCAL 參數中納入 Telemetry 值時，安裝程式會將 EC2LAUNCH_TELEMETRY 環境變數設為 1。

當您將 Telemetry 值放在 ADDLOCAL 參數之外時，安裝程式會將環境變數設為 0。

在 EC2Launch v2 代理程式執行時，其會讀取 EC2LAUNCH_TELEMETRY 環境變數，決定是否要上傳遙測資料。若值等於 1，則會上傳資料。若不是，則不會上傳。

範例：使用遙測安裝 EC2Launch v2

```
& msiexec /i "C:\Users\Administrator\Desktop\EC2Launchv2\AmazonEC2Launch.msi"
 ADDLOCAL="Basic,Clean,Telemetry" /q
```

確認 EC2Launch v2 版本

請用下列程序之一確認執行個體上安裝的 EC2Launch v2 版本。

PowerShell

使用 Windows PowerShell 確認已安裝的 EC2Launch v2 版本，如下所示。

1. 從 AMI 啟動執行個體，並連線至執行個體。
2. 在 PowerShell 中執行如下命令可確認已安裝的 EC2Launch 版本。

```
& "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" version
```

Windows Control Panel

在 Windows 控制台中確認已安裝的 EC2Launch v2 版本，如下所示。

1. 從 AMI 啟動執行個體，並連線至執行個體。
2. 開啟 Windows 控制台，然後選擇程式和功能。
3. 從已安裝程式清單中，找出 Amazon EC2Launch。其版本編號將出現在 Version (版本) 欄中。

若要檢視 AWS Windows AMIs 的最新更新，請參閱 Windows AMI 參考中的 Windows AMI 版本歷史記錄。 AWS

如需 EC2Launch v2 的最新版本，請參閱 EC2Launch v2 版本歷史記錄。

您會在 EC2Launch v2 服務有新版本發佈時收到通知。如需詳細資訊，請參閱訂閱 EC2 Windows 啟動代理程式通知。

設定 Windows 執行個體的 EC2Launch v2 設定

本節包含有關如何進行 EC2Launch v2 設定的資訊。

主題包括：

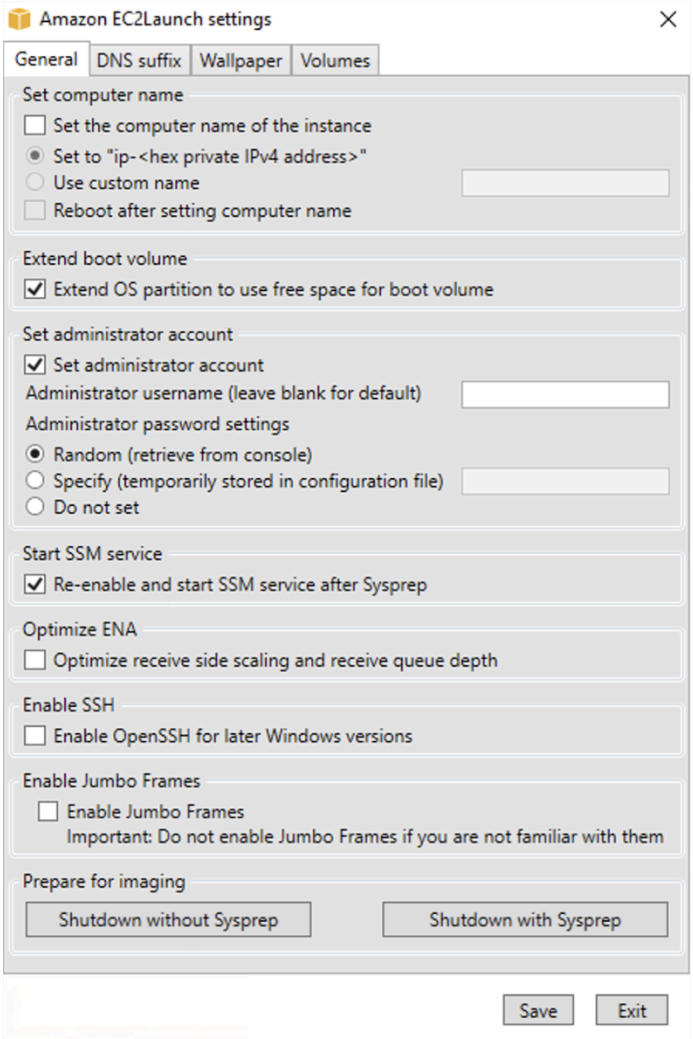
- 使用 EC2Launch v2 設定對話方塊變更設定
- 使用 CLI 設定 EC2Launch v2
- EC2Launch v2 任務組態
- EC2Launch v2 結束程式碼和重新啟動
- EC2Launch v2 和 Sysprep

使用 EC2Launch v2 設定對話方塊變更設定

下列程序說明如何使用 EC2Launch v2 對話方塊啟用或停用設定。

> **Note**
>
> 如果您在 agent-config.yml 檔案中不當設定自訂任務，並嘗試開啟 Amazon EC2Launch 設定對話方塊，您將收到錯誤。有關架構範例，請參閱 範例：agent-config.yml。

1. 啟動和連線至 Windows 執行個體。
2. 從「開始」功能表選擇所有程式,，然後瀏覽至 EC2Launch 設定。選擇透過 Sysprep 關閉或不透過 Sysprep 關閉之前，確認在執行關閉時儲存您想要套用的任何變更。



3. 在 Ec2 服務屬性對話方塊的一般索引標籤中，啟用或停用下列設定。

   a. Set Computer Name (設定電腦名稱)

      如果啟用此設定 (預設為停用)，則會在每次開機時將目前主機名稱與所需主機名稱進行比較。如果主機名稱不相符，則會重設主機名稱，然後系統會選擇性地重新開機以取得新的主機名稱。如果未指定自訂主機名稱，則會使用十六進位格式的私有 IPv4 地址產生名稱，例如：`ip-AC1F4E6`。若要避免現有的主機名稱遭到修改，請勿啟用此設定。

   b. 延伸開機磁碟區

      此設定將動態擴充 `Disk 0/Volume 0`，以包含任何尚未分割的空間。此設定在執行個體是從自訂大小的根磁碟區啟動時很有用。

   c. 設定管理員帳戶

啟用後，您可以為本機電腦上建立的管理員帳戶設定使用者名稱和密碼屬性。如果未啟用此功能，遵循 Sysprep 的系統不會建立系統管理員帳戶。只有在 `adminPassword` 是 `adminPasswordtype` 時，才要在 `Specify` 中提供密碼。

密碼類型的定義如下：

i. `Random`

EC2Launch 會產生密碼並以使用者金鑰將其加密。系統會在執行個體啟動後停用此設定，以便執行個體重新開機或停止再啟動時，能持續使用這個密碼。

ii. `Specify`

EC2Launch 會使用您在 `adminPassword` 中指定的密碼 如果該密碼不符合系統要求，EC2Launch 會改為產生隨機密碼。系統會將該密碼以純文字存放於 `agent-config.yml`，並在 Sysprep 設定管理員密碼後予以刪除。EC2Launch 會以使用者金鑰加密該密碼。

iii. `Do not set`

EC2Launch 會使用您在未使用的 .xml 檔案中指定的密碼。如果您未在 unattend.xml 中指定密碼，會停用管理員帳戶。

d. 啟動 SSM 服務

選取此選項時，Systems Manager 服務會啟用，以啟動以下 Sysprep。EC2Launch v2 將執行先前所述的所有任務，而且 SSM Agent 將處理 Systems Manager 功能的請求，例如執行命令和狀態管理員。

您可使用執行命令升級現有執行個體，以使用 EC2Launch v2 服務和 SSM Agent 的最新版本。若要了解詳細資訊，請參閱 AWS Systems Manager 使用者指南中的使用執行命令更新 SSM 代理程式。

e. 最佳化 ENA

選取時，會設定 ENA 設定，以確保針對 ENA 接收端擴展和接收佇列深度設定進行最佳化 AWS。如需詳細資訊，請參閱設定接收端擴展 CPU 親和性。

f. 啟用 SSH

此設定會啟用適用於較新版本 Windows 的 OpenSSH，以允許遠端系統管理。

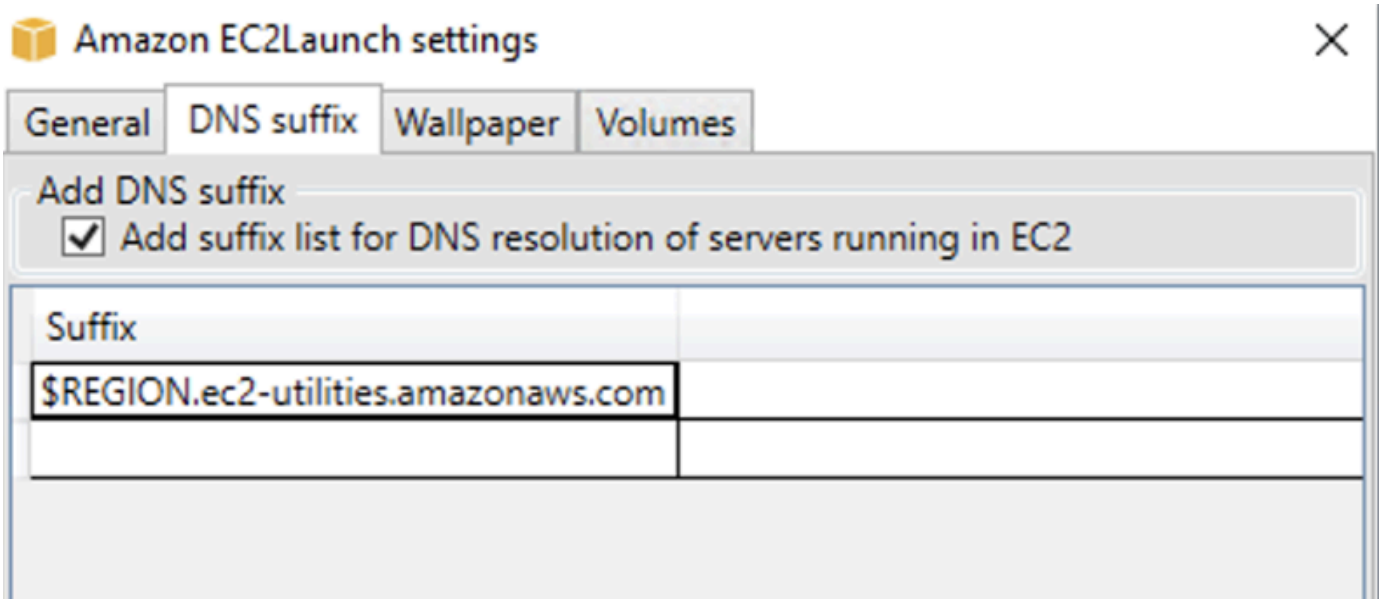
g. 啟用巨型訊框

選取以啟用「巨型訊框」。「巨型訊框」可能會對您的網路通訊產生意想不到的影響，因此在啟用之前，請務必了解「巨型訊框」對您的系統的影響。如需「巨型訊框」的詳細資訊，請參閱 巨型訊框 (9001 MTU)。

h. 準備建立映像

選取是否希望 EC2 執行個體在使用 Sysprep 的情況下關閉。當您想要使用 EC2Launch v2 執行 Sysprep 時，請選擇 Shutdown with Sysprep (使用 Sysprep 關機)。

4. 在DNS 尾碼標籤上，您可以選取是否要為 EC2 中執行之伺服器的 DNS 解析新增 DNS 尾碼清單，而不需提供完整網域名稱。DNS 尾碼可以包含變數 `$REGION` 和 `$AZ`。只有不存在的尾碼才會新增到清單中。



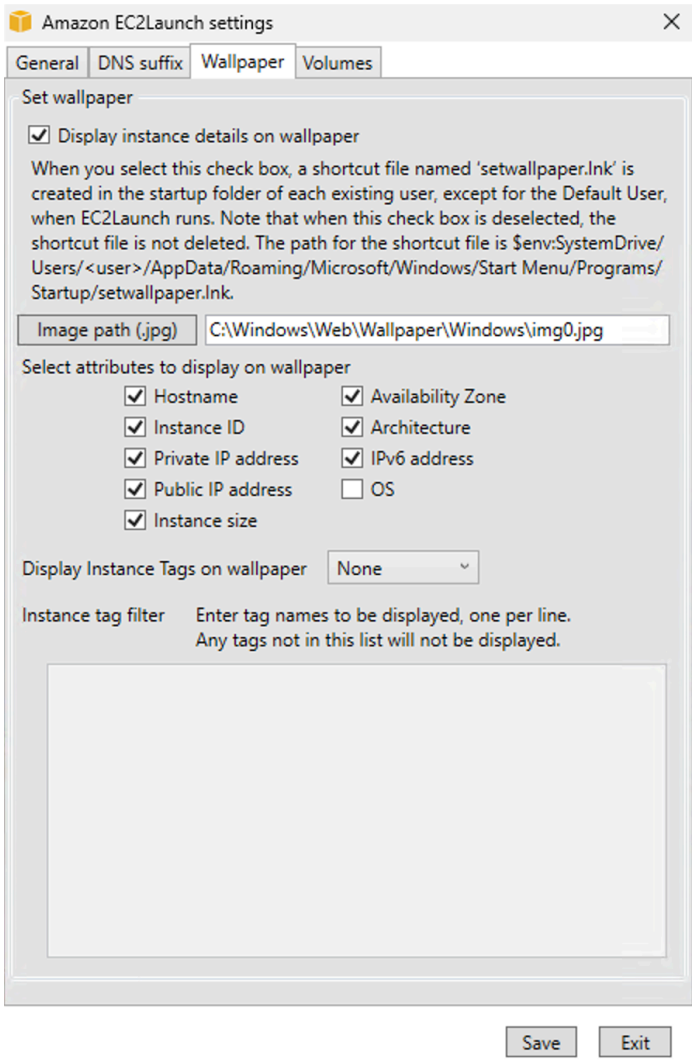
5. 在桌布索引標籤中，您可以使用背景圖片來設定執行個體桌布，並指定要顯示之桌布的執行個體詳細資訊。Amazon EC2 會在您每次登入時產生詳細資訊。

您可以使用下列控制項來設定桌布。

- 在桌布上顯示執行個體詳細資訊 - 此核取方塊會啟用或停用桌布上的執行個體詳細資訊顯示。
- 圖片路徑 (.jpg) - 指定要用作桌布背景的圖片路徑。
- 選取要在桌布上顯示的屬性 - 選取要顯示在桌布上之執行個體詳細資訊的核取方塊。清除要從桌布中移除的先前選取的執行個體詳細資訊的核取方塊。
- 在桌布上顯示執行個體標籤 - 選取下列其中一個設定以在桌布上顯示執行個體標籤：
  - 無 - 不在桌布上顯示任何執行個體標籤。
  - 顯示全部 - 在桌布上顯示所有執行個體標籤。

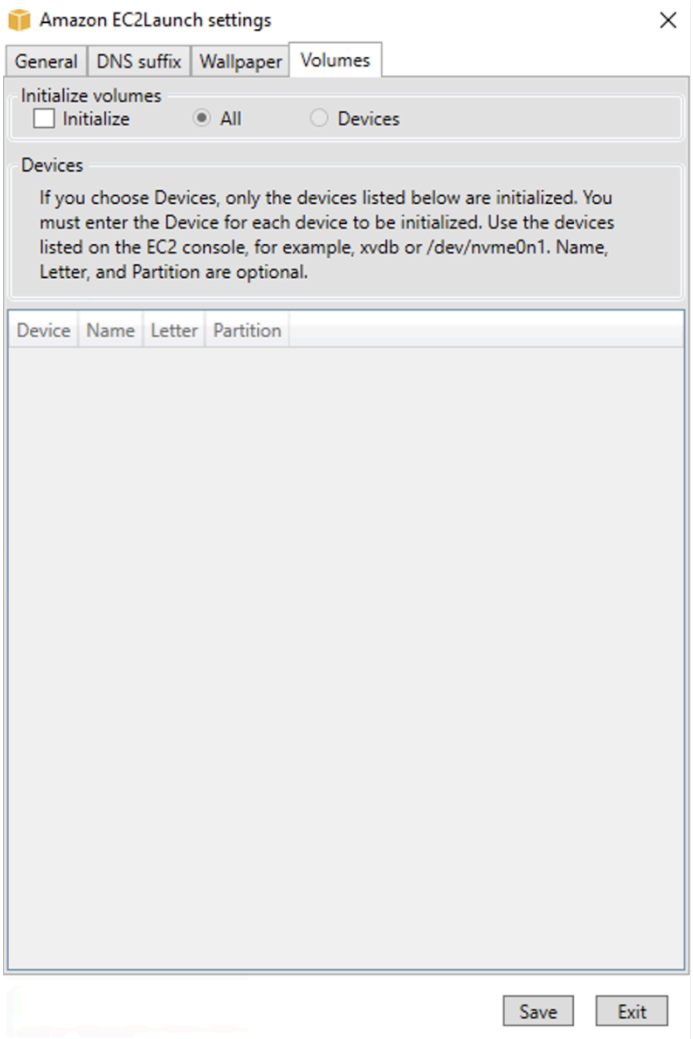
- 顯示篩選結果 - 在桌布上顯示指定的執行個體標籤。當您選取此設定時，可以在執行個體標籤篩選方塊中新增您想要顯示在桌布上的執行個體標籤。

> **Note**
>
> 您必須在中繼資料中啟用標籤，才能在桌布上顯示標籤。如需有關執行個體標籤和中繼資料的詳細資訊，請參閱 使用執行個體中繼資料檢視 EC2 執行個體的標籤。



6. 在磁碟區標籤上，選取是否要初始化連線至執行個體的磁碟區。啟用會設定任何額外磁碟區的磁碟機代號，並將它們延伸到使用可用空間。如果您選取全部，則會初始化所有儲存磁碟區。如果您選取裝置，則只會初始化清單中指定的裝置。您必須為每個要初始化的裝置輸入裝置。使用 EC2 主控台上列出的裝置，例如，xvdb 或 /dev/nvme0n1。下拉式清單會顯示連接到執行個體的儲存磁碟區。若要輸入未連接到執行個體的裝置，請在文字欄位中輸入該裝置。

名稱、字母和分割區是選填欄位。如果沒有為磁碟分割區指定任何值，則會以 gpt 磁碟分割區類型初始化大於 2 TB 的儲存磁碟區，而小於 2 TB 的磁碟分割區會以 mbr 磁碟分割區類型初始化。如果已設定裝置，且非 NTFS 裝置包含磁碟分割區表格，或磁碟的前 4 KB 會包含資料，接著系統會略過磁碟並記錄動作。



使用 CLI 設定 EC2Launch v2

您可以使用命令列介面 (CLI) 來配置 EC2Launch 設定並管理服務。下節包含您可用來管理 EC2Launch v2 之 CLI 命令的說明和使用資訊。

> **Note**
>
> EC2Launch v2 CLI 可執行檔 `EC2Launch.exe`位於 中`C:\Program Files\Amazon\EC2Launch\`。根據預設，此目錄不在 Windows PATH 中。本節範例中顯示的`ec2launch`

*command*表單是殼層無法直接辨識的速記。若要執行命令，請使用可執行檔的完整路徑，或先變更安裝目錄。下列 PowerShell 範例使用呼叫運算子的完整路徑：

```
& "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" status
```

命令

- collect-logs
- get-agent-config
- list-volumes
- 重設
- run
- status
- sysprep
- validate
- version
- 桌布

collect-logs

收集 EC2Launch 的日誌、壓縮檔案，並將它們置於指定的目錄中。

範例

```
ec2launch collect-logs -o C:\Mylogs.zip
```

用途

```
ec2launch collect-logs [flags]
```

Flags

```
-h, --help
```

```
collect-logs 的說明
```

`-o, --output string`

壓縮輸出日誌的路徑

get-agent-config

以指定的格式 (JSON 或 YAML) 列印 `agent-config.yml`。如果未指定任何格式，則會以先前指定的格式列印 `agent-config.yml`。

範例

```
ec2launch get-agent-config -f json
```

用途

`ec2launch get-agent-config [flags]`

Flags

`-h, --help`

`get-agent-config` 的說明

`-f, --format string`

`agent-config` 檔案的輸出格式：`json`、`yaml`

list-volumes

列出連接到執行個體的所有儲存磁碟區，包括暫時磁碟區和 EBS 磁碟區。

範例

```
ec2launch list-volumes
```

用途

`ec2launch list-volumes`

Flags

`-h, --help`

`list-volumes` 的說明

重設

此任務的主要目標是在下次代理程式執行時重設該代理程式。為此，reset 命令會從本機 `EC2Launch` 目錄刪除 EC2Launch v2 的所有代理程式狀態資料 (請參閱：EC2Launch v2 目錄結構)。選擇性重設會刪除服務和 Sysprep 日誌。

指令碼行為取決於代理程式執行指令碼的模式 (內嵌或分離)。

內嵌 (預設)

EC2Launch v2 代理程式會一次執行一個指令碼 (`detach: false`)。這是預設設定。

> **Note**
>
> 當內嵌指令碼發出 reset 或 sysprep 命令時，它會立即執行並重設代理程式。目前任務完成後，代理程式就會關閉，而不執行任何後續任務。
> 例如，如果發出命令的任務後緊接著一個 `startSsm` 任務 (預設情況下會在使用者資料執行之後包含)，則任務不會執行，而且 Systems Manager 服務永遠不會啟動。

Detached

EC2Launch v2 代理程式會與其他任務同時執行指令碼 (`detach: true`)。

> **Note**
>
> 當分離的指令碼發出 reset 或 sysprep 命令時，這些命令會等待代理程式完成再執行。executeScript 之後的任務仍會執行。

範例

```
ec2launch reset -c
```

用途

`ec2launch reset [flags]`

Flags

`-c, --clean`

在 `reset` 前清除執行個體日誌

`-h, --help`

`reset` 的說明

run

執行 EC2Launch v2。

範例

```
ec2launch run
```

用途

`ec2launch run [flags]`

Flags

`-h, --help`

`run` 的說明

status

取得 EC2Launch v2 代理程式的狀態。選擇性地封鎖處理程序，直到代理程式完成為止。處理程序結束程式碼會決定代理程式狀態：

- `0` – 代理程式已執行且成功。
- `1` – 代理程式已執行且失敗。
- `2` – 代理程式仍在執行中。
- `3` – 代理程式處於未知狀態。代理程式狀態為未執行或已停止。
- `4` – 嘗試擷取代理程式狀態時發生錯誤。
- `5` – 代理程式未執行，且上次已知執行的狀態未知。這可能表示以下其中一項：
  - `state.json` 和 `previous-state.json` 均被刪除。

- previous-state.json 已毀損。

這是執行 reset 命令後的代理程式狀態。

範例：

```
ec2launch status -b
```

> **Note**
>
> status 命令會將短訊息寫入主控台（例如 agent ran successfully)。不過，列出的程序結束代碼值先前會決定客服人員狀態，而不是該訊息。若要在 PowerShell 中檢視結束程式碼，請在執行命令後讀取$LASTEXITCODE自動變數，例如：
>
> ```
> & "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" status; "ExitCode=
> $LASTEXITCODE"
> ```
>
> 的輸出ExitCode=0表示代理程式已成功執行。

用途

ec2launch status [flags]

Flags

-b,--block

封鎖處理程序，直到代理程式完成執行

-h,--help

status 的說明

sysprep

此任務的主要目標是在下次代理程式執行時重設該代理程式。為此，sysprep 命令會重設代理程式狀態、更新 unattend.xml 檔案、停用 RDP，然後執行 Sysprep。

指令碼行為取決於代理程式執行指令碼的模式 (內嵌或分離)。

內嵌 (預設)

EC2Launch v2 代理程式會一次執行一個指令碼 (`detach: false`)。這是預設設定。

> **Note**
>
> 當內嵌指令碼發出 reset 或 sysprep 命令時，它會立即執行並重設代理程式。目前任務完成後，代理程式就會關閉，而不執行任何後續任務。
> 例如，如果發出命令的任務後緊接著一個 `startSsm` 任務 (預設情況下會在使用者資料執行之後包含)，則任務不會執行，而且 Systems Manager 服務永遠不會啟動。

Detached

EC2Launch v2 代理程式會與其他任務同時執行指令碼 (`detach: true`)。

> **Note**
>
> 當分離的指令碼發出 reset 或 sysprep 命令時，這些命令會等待代理程式完成再執行。executeScript 之後的任務仍會執行。

範例：

```
ec2launch sysprep
```

用途

`ec2launch sysprep [flags]`

Flags

`-c,--clean`

在 `sysprep` 前清除執行個體日誌

`-h,--help`

Sysprep 說明

`-s,--shutdown`

在 sysprep 後關閉執行個體

validate

驗證 agent-config 檔案 C:\ProgramData\Amazon\EC2Launch\config\agent-config.yml。

範例

```
ec2launch validate
```

用途

ec2launch validate [flags]

Flags

-h , --help

validate 的說明

version

取得可執行版本。

範例

```
ec2launch version
```

用途

ec2launch version [flags]

Flags

-h, --help

version 的說明

桌布

將新桌布設定為提供的桌布路徑 (.jpg 檔案)，並顯示選取的執行個體詳細資料。

語法

```
ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--all-tags ^
--
attributes=hostName,instanceId,privateIpAddress,publicIpAddress,ipv6Address,instanceSize,availa
```

輸入

Parameters

--allowed-tags [*tag-name-1*, *tag-name-n*]

(選用) 要顯示在桌布上的 Base64 編碼 JSON 執行個體標籤名稱陣列。您可以使用此標籤或 `--all-tags`，但不能同時使用兩者。

--attributes *attribute-string-1*, *attribute-string-n*

(選用) 要將設定套用至桌布的 `wallpaper` 屬性字串清單 (以逗號分隔)。

[--path | -p] *path-string*

(必要) 指定 `wallpaper` 背景圖片檔案路徑。

旗標

--all-tags

(選用) 在桌布上顯示所有執行個體標籤。您可以使用此標籤或 `--allowed-tags`，但不能同時使用兩者。

[--help | -h]

顯示 wallpaper 命令的說明。

EC2Launch v2 任務組態

本節包含 `agent-config.yml` 和使用者資料的組態結構描述任務、詳細資訊及範例。

任務與範例

- 結構描述：agent-config.yml

- 設定啟動或重新啟動期間執行的 EC2Launch v2 使用者資料指令碼

結構描述：**agent-config.yml**

agent-config.yml 檔案的結構如下所示。請注意，不能在同一階段重複任務。有關任務屬性，請參閱後面的任務說明。

文件結構：agent-config.yml

JSON

```
{
 "version": "1.1",
 "config": [
  {
   "stage": "string",
   "tasks": [
    {
     "task": "string",
     "inputs": {
      ...
     }
    },
    ...
   ]
  },
  ...
 ]
}
```

YAML

```
version: 1.1
config:
- stage: string
  tasks:
  - task: string
 inputs:
   ...
  ...
...
```

範例：**agent-config.yml**

下列範例顯示 agent-config.yml 組態檔的設定。

```
version: 1.1
config:
- stage: boot
  tasks:
  - task: extendRootPartition
- stage: preReady
  tasks:
  - task: activateWindows
    inputs:
      activation:
        type: amazon
  - task: setDnsSuffix
    inputs:
      suffixes:
      - $REGION.ec2-utilities.amazonaws.com
  - task: setAdminAccount
    inputs:
      password:
        type: random
  - task: setWallpaper
    inputs:
      path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
      attributes:
      - hostName
      - instanceId
      - privateIpAddress
      - publicIpAddress
      - instanceSize
      - availabilityZone
      - architecture
- stage: postReady
  tasks:
  - task: startSsm
```

設定啟動或重新啟動期間執行的 EC2Launch v2 使用者資料指令碼

下列 JSON 和 YAML 範例顯示使用者資料的文件結構。Amazon EC2 會剖析您在文件中指定的 tasks 陣列中命名的每個任務。每個任務都有自己的屬性集和要求集。如需詳細資訊，請參閱 EC2Launch v2 啟動任務的任務定義。

> **Note**
>
> 任務只能在使用者資料任務陣列中出現一次。

文件結構：使用者資料

JSON

```
{
 "version": "1.1",
 "tasks": [
  {
   "task": "string",
   "inputs": {
    ...
   },
  },
  ...
 ]
}
```

YAML

```
version: 1.1
tasks:
- task: string
  inputs:
    ...
...
```

範例：使用者資料

如需使用者資料的詳細資訊，請參閱 Amazon EC2 如何處理 Windows 執行個體的使用者資料。

下列 YAML 文件範例顯示 EC2Launch v2 執行以建立檔案的 PowerShell 指令碼 (作為使用者資料)。

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
```

```
    type: powershell
    runAs: localSystem
    content: |-
      New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

針對與舊版啟動代理程式相容的使用者資料，您可以使用 XML 格式。EC2Launch v2 會將指令碼作為 UserData 階段中的 executeScript 任務執行。為了符合 EC2Launch v1 和 EC2Config 行為，預設情況下使用者資料指令碼會作為連接/內嵌的程序執行。

您可以新增選用標籤來自訂指令碼的執行方式。例如，除了在執行個體啟動時執行一次使用者資料指令碼之外，若還想在執行個體重新啟動時執行，您可以使用下列標記：

<persist>true</persist>

範例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

您可以使用 <powershellArguments> 標籤指定一或多個 PowerShell 引數。如果未傳遞引數，EC2Launch v2 預設會新增下列引數：-ExecutionPolicy Unrestricted。

範例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

若要將 XML 使用者資料指令碼作為分離的程序執行，請將下列標籤新增至您的使用者資料。

<detach>true</detach>

範例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
```

```
  New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

> **Note**
>
> 舊版啟動代理程式不支援分離標籤。

變更日誌：使用者資料

下表列出使用者資料的變更，並將其交叉參考至適用的 EC2Launch v2 代理程式版本。

使用者資料版本	詳細資訊	引入於
1.1	• 使用者資料任務會在代理程式組態檔案中的 `PostReady` 階段之前執行。 • 在啟動 Systems Manager Agent 之前執行使用者資料 (與 EC2Launch v1 和 EC2Config 的行為相同)。*	EC2Launch v2 2.0.1245 版
1.0	• 將被棄用。 • 使用者資料任務會在代理程式組態檔案中的 `PostReady` 階段之後執行。這不能與 EC2Launch v1 回溯相容。 • 受 Systems Manager Agent 啟動與使用者資料任務之間的競爭條件影響。	EC2Launch v2 2.0.0 版

* 與預設 `agent-config.yml` 檔案搭配使用時。

EC2Launch v2 結束程式碼和重新啟動

您可以使用 EC2Launch v2 定義指令碼處理結束程式碼的方式。根據預設，在指令碼中執行的最後一個命令的結束程式碼會報告為整個指令碼的結束程式碼。例如，如果指令碼包含三個命令，而第一個命令失敗，但後續命令成功，則會因最終命令成功而將執行狀態報告為 `success`。

如果您希望指令碼重新啟動執行個體，則必須在指令碼中指定 `exit 3010`，即使重新啟動是指令碼中的最後一個步驟也是如此。`exit 3010` 會指示 EC2Launch v2 重新啟動執行個體並再次呼叫指令碼，直到傳回不是 `3010` 的結束程式碼，或直到達到最大重新啟動計數為止。EC2Launch v2 允許每個任務最多 5 次重新啟動。如果您嘗試使用不同的機制 (例如 `Restart-Computer`) 從指令碼重新啟動執行個體，則指令碼執行狀態將不一致。例如，狀態可能會卡在重新啟動迴圈或不執行重新啟動。

如果您使用與舊版代理程式相容的 XML 使用者資料格式，則使用者資料執行的次數可能會超出您的預期。如需詳細資訊，請參閱故障診斷一節中的 服務會多次執行使用者資料。

EC2Launch v2 和 Sysprep

EC2Launch v2 服務將執行 Sysprep，此為 Microsoft 的工具，可讓您建立自訂且可重複使用的 Windows AMI。EC2Launch v2 呼叫 Sysprep 時，會使用 `%ProgramData%\Amazon\EC2Launch` 中的檔案來決定要執行的操作。您可以使用 EC2Launch 設定 對話方塊間接編輯這些檔案，或直接使用 YAML 編輯器或文字編輯器來編輯這些檔案。但有些進階設定無法在 EC2Launch 設定對話方塊中使用，因此您必須直接編輯這些項目。

如果您在更新執行個體設定後從執行個體建立 AMI，新設定將套用到從新 AMI 啟用的任何執行個體。如需有關建立 AMI 的詳細資訊，請參閱 建立 Amazon EBS 支援的 AMI。

EC2Launch v2 啟動任務的任務定義

EC2Launch v2 在啟動或統計期間執行的每個任務都有專屬的一組屬性和要求。任務詳細資訊包含以下設定：任務執行的頻率 (一次或是永久執行)、該任務在代理程式啟動流程中的階段、其語法以及 YAML 文件範例。如需詳細資訊，請檢閱此參考中顯示的任務詳細資訊。

EC2Launch v2 任務

- activateWindows
- enableJumboFrames
- enableOpenSsh
- executeProgram
- executeScript
- extendRootPartition
- initializeVolume
- optimizeEna
- setAdminAccount
- setDnsSuffix

- setHostName
- setWallpaper
- startSsm
- sysprep
- writeFile

activateWindows

針對一組 AWS KMS 伺服器啟用 Windows。如果偵測到執行個體為自帶授權 (BYOL)，則會略過啟動。

Frequency — 一次

AllowedStages — `[PreReady]`

Inputs —

`activation`：(映射)

`type`：(字串) 要使用的啟動類型，設定為 `amazon`

範例

```
task: activateWindows
  inputs:
    activation:
    type: amazon
```

enableJumboFrames

啟用「巨型訊框」，以增加網路介面卡的最大傳輸單位 (MTU)。如需詳細資訊，請參閱巨型訊框 (9001 MTU)。

Frequency – 始終

AllowedStages — `[PostReady, UserData]`

Inputs — 無

範例

```
task: enableJumboFrames
```

enableOpenSsh

啟用 Windows OpenSSH，並將執行個體的公開金鑰新增至授權的金鑰資料夾。

Frequency — 一次

AllowedStages — `[PreReady, UserData]`

Inputs — 無

範例

以下範例示範如何在執行個體上啟用 OpenSSH，以及如何將執行個體的公有金鑰新增至授權的金鑰資料夾。此組態僅適用於執行 Windows Server 2019 及更新版本的執行個體。

```
task: enableOpenSsh
```

executeProgram

執行具有可選引數和指定頻率的程式。

階段：您可以在 `PreReady`、`PostReady` 和 `UserData` 階段執行 `executeProgram` 任務。

頻率：可設定，請參閱輸入。

輸入

本節包含要執行之 executeProgram 任務的一或多個程式 (輸入)。每個輸入可以包含下列可配置設定：

頻率 (字串)

(必要) 僅指定下列其中一個值：

- `once`
- `always`

路徑 (字串)

(必要) 要執行的可執行檔的檔案路徑。

引數 (字串清單)

(選用) 以逗號分隔的引數清單，作為輸入提供給程式。

runAs (字串)

(必要) 必須設定為 `localSystem`

Output

所有任務都會將日誌檔案條目寫入 `agent.log` 檔案。`executeProgram` 任務的其他輸出會分別儲存在動態命名的資料夾中，如下所示：

`%LocalAppData%\Temp\`*`EC2Launch#########`*`\`*`outputfilename.tmp`*

輸出檔案的確切路徑包含在 `agent.log` 檔案中，例如：

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\ExecuteProgramInputs.tmp
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

`executeProgram` 任務的輸出檔案

`ExecuteProgramInputs.tmp`

包含可執行檔的路徑，以及 `executeProgram` 任務在執行時傳送給其所有輸入參數。

`Output.tmp`

包含 `executeProgram` 任務執行之程式的執行期輸出。

`Err.tmp`

包含 `executeProgram` 任務執行之程式的執行期錯誤訊息。

範例

下列範例說明如何透過 `executeProgram` 任務，從執行個體上的本機目錄執行可執行檔。

範例 1：具有一個引數的設定可執行檔

此範例顯示在靜默模式下執行設定可執行檔的 `executeProgram` 任務。

```
task: executeProgram
```

```
  inputs:
    - frequency: always
      path: C:\Users\Administrator\Desktop\setup.exe
      arguments: ['-quiet']
```

範例 2：具有兩個引數的 VLC 可執行檔

此範例顯示執行 VLC 可執行檔的 `executeProgram` 任務，其中兩個引數作為輸入參數傳遞。

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\vlc-3.0.11-win64.exe
      arguments: ['/L=1033','/S']
      runAs: localSystem
```

executeScript

執行具有可選引數和指定頻率的指令碼。指令碼行為取決於代理程式執行指令碼的模式 (內嵌或分離)。

內嵌 (預設)

EC2Launch v2 代理程式會一次執行一個指令碼 (`detach: false`)。這是預設設定。

> **Note**
>
> 當內嵌指令碼發出 reset 或 sysprep 命令時，它會立即執行並重設代理程式。目前任務完成後，代理程式就會關閉，而不執行任何後續任務。
> 例如，如果發出命令的任務後緊接著一個 `startSsm` 任務 (預設情況下會在使用者資料執行之後包含)，則任務不會執行，而且 Systems Manager 服務永遠不會啟動。

Detached

EC2Launch v2 代理程式會與其他任務同時執行指令碼 (`detach: true`)。

> **Note**
>
> 當分離的指令碼發出 reset 或 sysprep 命令時，這些命令會等待代理程式完成再執行。executeScript 之後的任務仍會執行。

階段：您可以在 PreReady、PostReady 和 UserData 階段執行 executeScript 任務。

頻率：可設定，請參閱輸入。

輸入

本節包含要執行之 executeScript 任務的一或多個指令碼 (輸入)。每個輸入可以包含下列可配置設定：

頻率 (字串)

(必要) 僅指定下列其中一個值：

- once
- always

類型 (字串)

(必要) 僅指定下列其中一個值：

- batch
- powershell

引數 (字串清單)

(選用) 傳遞給 Shell (非 PowerShell 指令碼) 的字串引數清單。type: batch 不支援此參數。如果未傳遞引數，EC2Launch v2 預設會新增下列引數：-ExecutionPolicy Unrestricted。

內容 (字串)

(必要) 指令碼內容。

runAs (字串)

(必要) 僅指定下列其中一個值：

- admin
- localSystem

分離 (布林值)

(選用) EC2Launch v2 代理程式預設為一次執行一個指令碼 (detach: false)。若要與其他任務同時執行指令碼，請將值設定為 true (detach: true)。

> **Note**
>
> 當 detach 設定為 true 時，指令碼結束代碼 (包括 3010) 沒有任何作用。

Output

所有任務都會將日誌檔案條目寫入 agent.log 檔案。executeScript 任務執行之指令碼的其他輸出會分別儲存在動態命名的資料夾中，如下所示：

%LocalAppData%\Temp*EC2Launch#########**outputfilename.ext*

輸出檔案的確切路徑包含在 agent.log 檔案中，例如：

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\UserScript.ps1
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

executeScript 任務的輸出檔案

UserScript.*ext*

包含 executeScript 任務執行的指令碼。檔案的副檔名取決於您在 executeScript 任務的 type 參數中指定的指令碼類型，如下所示：

- 如果類型為 batch，則檔案副檔名為 .bat。
- 如果類型為 powershell，則檔案副檔名為 .ps1。

Output.tmp

包含 executeScript 任務執行之指令碼的執行期輸出。

Err.tmp

包含 executeScript 任務執行之指令碼的執行期錯誤訊息。

範例

下列範例示範如何使用 executeScript 任務執行內嵌指令碼。

範例 1：Hello World 輸出文字檔案

此範例顯示了 executeScript 任務，其執行 PowerShell 指令碼以在 C: 磁碟機上建立「Hello World」文字檔案。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: admin
      content: |-
        New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
        Set-Content 'C:\PowerShellTest.txt' "Hello world"
```

範例 2：執行兩個指令碼

此範例顯示 executeScript 任務可以執行多個指令碼，而指令碼類型不一定需要相符。

第一個指令碼 (type: powershell) 會將執行個體上目前執行的程序摘要寫入位於 C: 磁碟機上的文字檔案。

第二個指令碼 (batch) 會將系統資訊寫入 Output.tmp 檔案。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |
        Get-Process | Out-File -FilePath C:\Process.txt
    - frequency: always
      type: batch
      runAs: localSystem
      content: |
        systeminfo
```

範例 3：具有重新啟動功能的等冪系統組態

此範例顯示 executeScript 任務，其執行等冪指令碼以執行下列系統組態，並在每個步驟之間重新啟動：

- 重新命名電腦。

- 將電腦加入網域。
- 啟用 Telnet。

該指令碼確保每個操作只執行一次。這可以防止重新啟動循環，並讓指令碼具有等冪性。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |-
        $name = $env:ComputerName
        if ($name -ne $desiredName) {
          Rename-Computer -NewName $desiredName
          exit 3010
        }
        $domain = Get-ADDomain
        if ($domain -ne $desiredDomain)
        {
          Add-Computer -DomainName $desiredDomain
          exit 3010
        }
        $telnet = Get-WindowsFeature -Name Telnet-Client
        if (-not $telnet.Installed)
        {
          Install-WindowsFeature -Name "Telnet-Client"
          exit 3010
        }
```

extendRootPartition

延伸根磁碟區以使用磁碟上的所有可用空間。

Frequency — 一次

AllowedStages — [Boot]

Inputs — 無

範例

```
task: extendRootPartition
```

initializeVolume

初始化連接到執行個體的空白磁碟區，以便啟動和分割這些磁碟區。如果啟動代理程式偵測到磁碟區並非空白，就會略過初始化。如果磁碟區的前 4 KiB 空白，或磁碟區沒有 Windows 可辨識的磁碟機配置，磁碟區會視為空白。

不論磁碟機是否已經初始化，在此任務執行時永遠都會套用 `letter` 輸入參數。

`initializeVolume` 任務會執行下列動作。

- 將磁碟屬性 `offline` 和 `readonly` 設為 false。
- 建立分割區。如果未在 `partition` 輸入參數中指定分割區類型，會套用下列預設值：
  - 如果磁碟大小小於 2 TB，請將分割區類型設為 `mbr`。
  - 如果磁碟大小大於等於 2 TB，請將分割區類型設為 `gpt`。
- 將磁碟區格式化為 NTFS。
- 將磁碟區標籤設定如下：
  - 如果有指定，請使用 `name` 輸入參數的值。
  - 如果磁碟區為暫時性，且未指定名稱，請將磁碟區標籤設為 `Temporary Storage Z`。
- 如果磁碟區為暫時性 (SSD 或 HDD，而非 Amazon EBS)，請在磁碟區的根目錄建立 `Important.txt` 檔案，其中包含下列內容：

```
This is an 'Instance Store' disk and is provided at no additional charge.

*This disk offers increased performance since it is local to the host
*The number of Instance Store disks available to an instance vary by instance type
*DATA ON THIS DRIVE WILL BE LOST IN CASES OF IMPAIRMENT OR STOPPING THE INSTANCE.
 PLEASE ENSURE THAT ANY IMPORTANT DATA IS BACKED UP FREQUENTLY

For more information, please refer to: EC 2 ##################.
```

- 將磁碟機代號設定為 `letter` 輸入參數中指定的值。

階段：您可以在 `PostReady` 和 `UserData` 階段執行 `initializeVolume` 任務。

頻率：始終。

輸入

您可以設定執行期參數，如下所示：

devices (映射清單)

(視情況而定) 啟動代理程式初始化之每個裝置的組態。initialize 輸入參數設為 devices 時，此為必要項目。

- device (字串，必要) – 在建立執行個體期間識別裝置。例如，xvdb、xvdf 或 \dev \nvme0n1。
- letter (字串，選用) – 一個字元。要指派的磁碟機代號。
- name (字串，選用) – 要指派的磁碟區名稱。
- partition (字串，選用) – 為要建立的分割區類型指定下列任一個值，或根據磁碟區大小讓啟動代理程式設定預設值：
  - mbr
  - gpt

initialize (字串)

(必要) 僅指定下列其中一個值：

- all
- devices

範例

下列範例說明 initializeVolume 任務的輸入組態範例。

範例 1：在執行個體上初始化兩個磁碟區

此範例說明在執行個體上初始化兩個次要磁碟區的 initializeVolume 任務。範例中的裝置 DataVolume2 為暫時性。

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: DataVolume1
    letter: D
    partition: mbr
  - device: /dev/nvme0n1
    name: DataVolume2
    letter: E
```

```
    partition: gpt
```

範例 2：初始化連接至執行個體的 EBS 磁碟區

此範例說明 `initializeVolume` 任務，這會初始化所有連接至執行個體的空白 EBS 磁碟區。

```
task: initializeVolume
inputs:
  initialize: all
```

optimizeEna

根據目前的執行個體類型最佳化 ENA 設定；可能會將執行個體重新開機。

Frequency — 永遠

AllowedStages — `[PostReady, UserData]`

Inputs — 無

範例

```
task: optimizeEna
```

setAdminAccount

為在本機電腦上建立的預設管理員帳戶設定屬性。

Frequency — 一次

AllowedStages — `[PreReady]`

Inputs —

`name`：(字串) 管理員帳戶的名稱

`password`：(映射)

`type`：(字符) 策略來設置密碼，無論是 `static`、`random` 或 `doNothing`

`data`：(字串) 儲存資料，如果 `type` 欄位是靜態的話

範例

```
task: setAdminAccount
inputs:
  name: Administrator
  password:
  type: random
```

setDnsSuffix

將 DNS 尾碼新增至搜尋尾碼清單。只有不存在的尾碼才會加入到清單中。如需啟動代理程式如何設定 DNS 尾碼的詳細資訊，請參閱 設定 EC2 Windows 啟動代理程式的 DNS 尾碼。

Frequency – 始終

AllowedStages — `[PreReady]`

Inputs —

`suffixes`：(字串清單) 一或多個有效 DNS 尾碼清單；有效的替代變數為 `$REGION` 和 `$AZ`

範例

```
task: setDnsSuffix
inputs:
  suffixes:
  - $REGION.ec2-utilities.amazonaws.com
```

setHostName

將電腦的主機名稱設定為自訂字串，或者如果 `hostName` 未指定，則設定私有 IPv4 地址。

Frequency — 永遠

AllowedStages — `[PostReady, UserData]`

Inputs —

`hostName`：(字串) 選擇性的主機名稱，必須格式化如下。

- 必須等於或少於 15 個字元
- 必須只包含英數字元 (a-z、A-Z、0-9) 和連字號 (-) 字元。
- 不得完全由數字字元組成。

reboot：(布林值) 表示當主機名被更改時是否允許重新啟動

範例

```
task: setHostName
inputs:
  reboot: true
```

setWallpaper

除了 Default User 之外，在每個現有使用者的啟動資料夾中建立 setwallpaper.lnk 捷徑檔案。使用者在執行個體啟動後第一次登入時，此捷徑檔案會執行。這會使用顯示執行個體屬性的自訂桌布，設定執行個體。

捷徑檔案路徑為：

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/
Startup/setwallpaper.lnk
```

> **Note**
>
> 移除 setWallpaper 任務時，不會刪除此捷徑檔案。如需詳細資訊，請參閱setWallpaper 任務未啟用，但桌布會在重新啟動時重設。

階段：您可以在 PreReady 和 UserData 階段設定桌布。

頻率：always

桌布組態

您可以使用下列設定來配置桌布。

輸入

您提供的輸入參數，以及可以設定桌布的屬性：

路徑 (字串)

(必要) 要用於桌布圖片的本機 .jpg 格式圖片檔案的檔案名稱路徑。

屬性 (字串清單)

(選用) 您可以將以下一個或多個屬性新增至桌布：

- architecture
- availabilityZone
- hostName
- instanceId
- instanceSize
- privateIpAddress
- publicIpAddress
- ipv6Address

instanceTags

(選用) 您可以使用下列其中一個選項來進行此設定。

- AllTags (字串) – 將所有執行個體標籤新增至桌布。

```
instanceTags: AllTags
```

- instanceTags (字串清單) – 指定要新增至桌布的執行個體標籤名稱清單。例如：

```
instanceTags:
  - Tag 1
  - Tag 2
```

範例

下列範例顯示設定桌布背景圖片檔案路徑的桌布組態輸入，以及名為 Tag 1 和 Tag 2 的執行個體標籤，以及包含執行個體的主機名稱、執行個體 ID 及私有和公有 IP 地址的屬性。

```
task: setWallpaper
inputs:
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
  - hostName
  - instanceId
  - privateIpAddress
  - publicIpAddress
  instanceTags:
  - Tag 1
  - Tag 2
```

> **Note**
>
> 您必須在中繼資料中啟用標籤，才能在桌布上顯示標籤。如需有關執行個體標籤和中繼資料的詳細資訊，請參閱 使用執行個體中繼資料檢視 EC2 執行個體的標籤。

startSsm

依照 Sysprep 開始 Systems Manager (SSM) 服務。

Frequency — 永遠

AllowedStages — `[PostReady, UserData]`

Inputs — 無

範例

```
task: startSsm
```

sysprep

重設服務狀態、更新 `unattend.xml`、停用 RDP，以及執行 Sysprep。只有在所有其他任務完成後才會執行此任務。

Frequency — 一次

AllowedStages — `[UserData]`

Inputs —

`clean`：(布林值) 在執行 Sysprep 之前清除執行個體日誌

`shutdown`：(布林值) 在執行 Sysprep 後關閉執行個體

範例

```
task: sysprep
inputs:
clean: true
shutdown: true
```

writeFile

將檔案寫入目的地。

Frequency — 請參閱 Inputs

AllowedStages — `[PostReady, UserData]`

Inputs —

`frequency`：(字符) 其中一個 `once` 或 `always`

`destination`：(字符) 要寫入內容的路徑

`content`：(字符) 要寫入文本的目的地

範例

```
task: writeFile
inputs:
  - frequency: once
  destination: C:\Users\Administrator\Desktop\booted.txt
  content: Windows Has Booted
```

疑難排解使用 EC2Launch v2 代理程式的問題

本節說明 EC2Launch v2 的常見故障診斷案例、有關檢視 Windows 事件日誌的資訊，以及主控台日誌輸出和訊息。

故障診斷主題

- 常見的疑難排解案例：
- Windows 事件日誌
- EC2Launch v2 主控台日誌輸出

常見的疑難排解案例：

本節說明常見的疑難排解案例和解決步驟。

案例

- 服務無法設定桌布

- 服務無法執行使用者資料
- 服務只執行一次任務
- 服務無法執行任務
- 服務會多次執行使用者資料
- 在移轉至 EC2Launch v2 後，來自 EC2Launch v1 的排程任務無法執行
- 服務會初始化不為空的 EBS 磁碟區
- setWallpaper 任務未啟用，但桌布會在重新啟動時重設
- 服務卡在執行狀態
- 無效的 agent-config.yml 導致無法開啟 EC2Launch v2 設定對話方塊
- task:executeScript should be unique and only invoked once

服務無法設定桌布

Resolution

1. 檢查 `%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\setwallpaper.lnk` 是否存在。
2. 檢查 `%ProgramData%\Amazon\EC2Launch\log\agent.log` 是否發生任何錯誤。

服務無法執行使用者資料

可能原因：服務可能在執行使用者資料之前失敗。

Resolution

1. 檢查 `%ProgramData%\Amazon\EC2Launch\state\previous-state.json`。
2. 查看 `boot`、`network`、`preReady` 和 `postReadyLocalData` 是否已全部標記為成功。
3. 如果其中一個階段失敗，請檢查 `%ProgramData%\Amazon\EC2Launch\log\agent.log` 是否有特定的錯誤。

服務只執行一次任務

Resolution

1. 檢查任務的頻率。

2. 如果服務已經在 Sysprep 之後執行，且任務頻率設定為 once，則任務將不會再次執行。
3. 如果您希望 EC2Launch v2 每次執行時都能執行任務，請將任務的頻率設定為 always。

服務無法執行任務

Resolution

1. 檢查 %ProgramData%\Amazon\EC2Launch\log\agent.log 中的最新項目。
2. 如果沒有發生錯誤，請嘗試從 "%ProgramFiles%\Amazon\EC2Launch\EC2Launch.exe" run 手動執行服務，以查看任務是否成功。

服務會多次執行使用者資料

Resolution

EC2Launch v1 和 EC2Launch v2 處理使用者資料的方式不同。當 persist 設定為 true 時，EC2Launch v1 會將使用者資料視為執行個體上的排程任務來執行。如果將 persist 設定為 false，即使任務由於重新啟動而退出或在執行時中斷，也不會進行排程。

EC2Launch v2 會將使用者資料視為代理程式任務執行，並追蹤其執行狀態。如果使用者資料導致電腦重新啟動，或是使用者資料在執行時中斷，執行狀態將一直保持為 pending，且使用者資料將會在下次執行個體啟動時再次執行。如果您想防止使用者資料指令碼多次執行，請使指令碼等冪。

下列範例等冪指令碼會設定電腦名稱並加入網域。

```
<powershell>
  $name = $env:computername
  if ($name -ne $desiredName) {
 Rename-Computer -NewName $desiredName
  }
  $domain = Get-ADDomain
  if ($domain -ne $desiredDomain)
  {
 Add-Computer -DomainName $desiredDomain
  }
  $telnet = Get-WindowsFeature -Name Telnet-Client
  if (-not $telnet.Installed)
  {
 Install-WindowsFeature -Name "Telnet-Client"
  }
</powershell>
```

```
<persist>false</persist>
```

在移轉至 EC2Launch v2 後，來自 EC2Launch v1 的排程任務無法執行

Resolution

遷移工具不會偵測任何連結至 EC2Launch v1 指令碼的排程任務；因此，不會自動在 EC2Launch v2 中設定這些任務。若要設定這些任務，請編輯 `agent-config.yml` 檔案，或使用 EC2Launch v2 設定對話方塊。例如，如果執行個體有執行 `InitializeDisks.ps1` 的排程任務，則執行遷移工具後，您必須在 EC2Launch v2 設定對話方塊中指定要初始化的磁碟區。請參閱 使用 EC2Launch v2 設定對話方塊變更設定 程序的步驟 6。

服務會初始化不為空的 EBS 磁碟區

Resolution

在初始化磁碟區之前，EC2Launch v2 會嘗試偵測磁碟區是否為空。如果磁碟區不為空，則會略過初始化程序。偵測為非空白的任何磁碟區都不會初始化。如果磁碟區的前 4 KiB 為空，或磁碟區沒有 Windows 可辨識的磁碟機配置，磁碟區會被視為空。在 Linux 系統上初始化並格式化的磁碟區沒有 Windows 可辨識的磁碟機配置，例如 MBR 或 GPT。因此，該磁碟區將視為空磁碟區且已初始化。如果您想要保留此資料，請勿依賴 EC2Launch v2 空白磁碟機偵測。而是在 EC2Launch v2 設定對話方塊 (請參閱步驟 6) 或 `agent-config.yml` 中指定要初始化的磁碟區。

`setWallpaper` 任務未啟用，但桌布會在重新啟動時重設

`setWallpaper` 任務會在每個現有使用者啟動資料夾中，建立 `setwallpaper.lnk` 捷徑檔案。使用者在執行個體啟動後第一次登入時，此捷徑檔案會執行。這會使用顯示執行個體屬性的自訂桌布，設定執行個體。移除 `setWallpaper` 任務不會刪除此捷徑檔案。您必須手動刪除此檔案，或使用指令碼將其刪除。

捷徑路徑為：

`$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/setwallpaper.lnk`

Resolution

手動刪除此檔案，或使用指令碼將其刪除。

刪除捷徑檔案的範例 PowerShell 指令碼

```
foreach ($userDir in (Get-ChildItem "C:\Users" -Force -Directory).FullName)
```

```
{
 $startupPath = Join-Path $userDir -ChildPath "AppData\Roaming\Microsoft\Windows\Start
 Menu\Programs\Startup"
 if (Test-Path $startupPath)
 {
  $wallpaperSetupPath = Join-Path $startupPath -ChildPath "setwallpaper.lnk"
  if (Test-Path $wallpaperSetupPath)
  {
   Remove-Item $wallpaperSetupPath -Force -Confirm:$false
  }
 }
}
```

服務卡在執行狀態

Description

EC2Launch v2 會被封鎖，其日誌訊息 (agent.log) 內容如下：

```
2022-02-24 08:08:58 Info:
 ********************************************************************************
2022-02-24 08:08:58 Info: EC2Launch Service starting
2022-02-24 08:08:58 Info: Windows event custom log exists: Amazon EC2Launch
2022-02-24 08:08:58 Info: ACPI SPCR table not supported. Bailing Out
2022-02-24 08:08:58 Info: Serial port is in use. Waiting for Serial Port...
2022-02-24 08:09:00 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:02 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:04 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:06 Info: ACPI SPCR table not supported. Use default console port.
```

可能的原因

SAC 處於啟用狀態，並使用序列連接埠。如需詳細資訊，請參閱使用 SAC 疑難排解您的 Windows 執行個體。

Resolution

請嘗試下列步驟來解決此問題：

- 停用正在使用序列連接埠的服務。
- 如果希望服務繼續使用序列連接埠，請編寫自訂指令碼來執行啟動代理程式任務，並將其做為排程任務呼叫。

無效的 **agent-config.yml** 導致無法開啟 EC2Launch v2 設定對話方塊

說明

EC2Launch v2 設定將嘗試剖析 agent-config.yml 檔案，然後開啟對話方塊。如果 YAML 組態檔案不遵循支援的結構描述，對話方塊將顯示以下錯誤：

```
Unable to parse configuration file agent-config.yml. Review configuration file. Exiting application.
```

Resolution

1. 驗證組態檔案是否遵循支援的結構描述。
2. 如果要從頭開始，請將預設組態檔案複製到 agent-config.yml。您可以使用「任務組態」區段中提供的範例 agent-config.yml。
3. 您也可以透過刪除 agent-config.yml 重新開始。EC2Launch v2 設定產生空的組態檔案。

task:executeScript should be unique and only invoked once

說明

不能在同一階段重複任務。

Resolution

某些任務必須作為陣列輸入，如 executeScript 和 executeProgram。如需將指令碼編寫為陣列的範例，請參閱 executeScript。

Windows 事件日誌

EC2Launch v2 會針對重要事件發佈 Windows 事件日誌，例如服務啟動、Windows 已就緒，以及任務成功和失敗。事件 ID 會唯一識別特定事件。每個事件都包含階段、任務和層級資訊，以及說明。您可以使用事件 ID 來設定特定事件的觸發程序。

事件 ID 提供事件的相關資訊，並唯一識別某些事件。事件 ID 的最低有效位元顯示事件的嚴重性。

事件	最低有效位元
Success	. . .0
Informational	. . .1

事件	最低有效位元
Warning	. . .2
Error	. . .3

服務啟動或停止時產生的服務相關事件包含單一數字事件識別符。

事件	單一數字識別符
Success	0
Informational	1
Warning	2
Error	3

事件的 EC2LaunchService.exe 事件訊息以 Service: 為開頭。事件的 EC2Launch.exe 事件訊息不是以 Service: 為開頭。

四位數的事件 ID 包括關於階段、任務和事件嚴重性的資訊。

主題

- 事件 ID 格式
- 事件 ID 範例
- Windows 事件日誌結構描述

事件 ID 格式

下表顯示 EC2Launch v2 事件識別符的格式。

3	2 1	0
S	T	L

3	2 1	0

表格中的字母和數字代表下列事件類型和定義。

Event type	定義
S (階段)	0 - 服務層級訊息 1 - 開機 2 - 網路 3 - 預備就緒 5 - Windows 已準備就緒 6 - 後備就緒 7 - 使用者資料
T (任務)	由對應的兩個值所代表的任務對每個階段都不同。若要檢視完整的事件清單，請參閱 Windows 事件日誌結構描述。
L (活動等級)	0 - 成功 1 - 資訊 2 - 警告 3 - 錯誤

事件 ID 範例

以下是範例事件 ID 。

- 5000 - Windows 已做好使用準備

- 3010 - 在預備就緒階段啟動視窗任務成功
- 6013 - 在後備就緒本機資料階段設置桌布任務時發生錯誤

Windows 事件日誌結構描述

訊息/事件 ID	事件訊息
. . .0	Success
. . .1	Informational
. . .2	Warning
. . .3	Error
x	EC2Launch service-level logs
0	EC2Launch service exited successfully
1	EC2Launch service informational logs
2	EC2Launch service warning logs
3	EC2Launch service error logs
10	Replace state.json with previous-state.json
100	Serial Port
200	Sysprep
300	PrimaryNic
400	Metadata
x000	Stage (1 digit), Task (2 digits), Status (1 digit)

訊息/事件 ID	事件訊息
1000	Boot
1010	Boot - extend_root_partition
2000	Network
2010	Network - add_routes
3000	PreReady
3010	PreReady - activate_windows
3020	PreReady - install_egpu_manager
3030	PreReady - set_monitor_on
3040	PreReady - set_hibernation
3050	PreReady - set_admin_account
3060	PreReady - set_dns_suffix
3070	PreReady - set_wallpaper
3080	PreReady - set_update_schedule
3090	PreReady - output_log
3100	PreReady - enable_open_ssh
5000	Windows is Ready to use
6000	PostReadyLocalData
7000	PostReadyUserData
6010/7010	PostReadyLocal/UserData - set_wallpaper

訊息/事件 ID	事件訊息
6020/7020	PostReadyLocal/UserData - set_update_schedule
6030/7030	PostReadyLocal/UserData - set_hostname
6040/7040	PostReadyLocal/UserData - execute_program
6050/7050	PostReadyLocal/UserData - execute_script
6060/7060	PostReadyLocal/UserData - manage_package
6070/7070	PostReadyLocal/UserData - initialize_volume
6080/7080	PostReadyLocal/UserData - write_file
6090/7090	PostReadyLocal/UserData - start_ssm
7100	PostReadyUserData - enable_open_ssh
6110/7110	PostReadyLocal/UserData - enable_jumbo_frames

EC2Launch v2 主控台日誌輸出

本節包含 EC2Launch v2 的範例主控台日誌輸出，並列出所有 EC2Launch v2 主控台日誌錯誤訊息，以協助您針對問題進行故障診斷。如需執行個體主控台輸出及存取方式的詳細資訊，請參閱 the section called “執行個體主控台輸出”。

輸出

- EC2Launch v2 主控台日誌輸出
- EC2Launch v2 主控台日誌訊息

EC2Launch v2 主控台日誌輸出

以下是 EC2Launch v2 的範例主控台日誌輸出。本範例中的部分值已取代為置於大括號內的代表性文字。

```
2025/07/22 21:26:53Z: Windows sysprep configuration complete.
2025/07/22 21:26:53Z: Message: Waiting for access to metadata...
2025/07/22 21:26:53Z: Message: Meta-data is now available.
2025/07/22 21:26:53Z: AMI Origin Version: 2024.12.13
2025/07/22 21:26:53Z: AMI Origin Name: Windows_Server-2022-English-Full-Base
2025/07/22 21:26:53Z: OS: Microsoft Windows NT 10.0.20348
2025/07/22 21:26:53Z: OsVersion: 10.0
2025/07/22 21:26:53Z: OsProductName: Windows Server 2022 Datacenter
2025/07/22 21:26:53Z: OsBuildLabEx: 20348.1.amd64fre.fe_release.210507-1500
2025/07/22 21:26:53Z: OsCurrentBuild: 20348
2025/07/22 21:26:53Z: OsReleaseId: 2009
2025/07/22 21:26:53Z: Language: en-US
2025/07/22 21:26:53Z: TimeZone: UTC
2025/07/22 21:26:53Z: Offset: UTC +0000
2025/07/22 21:26:53Z: Launch: EC2 Launch v2.2.63
2025/07/22 21:26:53Z: AMI-ID: ami-1234567890abcdef1
2025/07/22 21:26:53Z: Instance-ID: i-1234567890abcdef0
2025/07/22 21:26:54Z: Instance Type: t3.xlarge
2025/07/22 21:26:54Z: Driver: AWS NVMe Driver v1.6.0.35
2025/07/22 21:26:54Z: SubComponent: 1.6.0.35; EnableSCSIPersistentReservations: 0
2025/07/22 21:26:54Z: Driver: AWS PV Driver Package v8.5.0
2025/07/22 21:26:55Z: Driver: Amazon Elastic Network Adapter v2.8.0.0
2025/07/22 21:26:55Z: Driver: EC2 Windows Utility Device v3.1.0.25
2025/07/22 21:26:55Z: HOSTNAME: EC2AMAZ-9FJG5CC
2025/07/22 21:26:55Z: RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
2025/07/22 21:26:55Z: RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
2025/07/22 21:26:56Z: SSM: Amazon SSM Agent v3.3.2746.0
2025/07/22 21:26:57Z: User data format: no_user_data
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentCommandErrorCode=1
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentErrorLocation=execute_windows.go:410
```

```
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/22 21:26:57Z: Message: Windows is Ready to use
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
```

EC2Launch v2 主控台日誌訊息

以下是所有 EC2Launch v2 主控台日誌訊息的清單。

```
Error EC2Launch service is stopping. {error message}
```

已停止服務的錯誤詳細資訊：

- `Error setting up EC2Launch agent folders`
- `See instance logs for detail`
- `Error stopping service`
- `Error initializing service`

```
Windows sysprep configuration complete
```

```
Invalid administrator username: {invalid username}
```

```
Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
```

下列訊息是包含 AMI 詳細資訊的資訊區塊：

```
AMI Origin Version: {amiVersion}
AMI Origin Name: {amiName}
Microsoft Windows NT {currentVersion}.{currentBuildNumber}
OsVersion: {currentVersion}
OsProductName: {productName}
OsBuildLabEx: {buildLabEx}
OsCurrentBuild: {currentBuild}
OsReleaseId: {releaseId}
Language: {language}
```

```
TimeZone: {timeZone}
Offset: UTC {offset}
Launch agent: EC2Launch {BuildVersion}
AMI-ID: {amiId}
Instance-ID: {instanceId}
Instance Type: {instanceType}
HOSTNAME: {computer name}
RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
SqlServerBilling: {sql billing}
SqlServerInstall: {sql patch leve, edition type}
Driver: AWS NVMe Driver {version}
Driver: Inbox NVMe Driver {version}
Driver: AWS PV Driver Package {version}
Driver: EC2 Windows Utility Device {version}
SSM: Amazon SSM Agent {version}
AWS VSS Version: {version}
```

```
Windows sysprep configuration complete.
Windows is being configured. 'SysprepState is {state}'
Windows is still being configured. 'SysprepState is {state}'
Windows is Ready to use
Waiting for access to metadata...
Meta-data is now available.
Metadata is not available for this instance.
Timed out waiting for access to metadata.
User data format: {format}
```

EC2Launch v2 遙測訊息包含了啟動遙測屬性值。自版本 2.2.63 起，EC2 代理程式遙測資料會格式化為 JSON 物件。

```
EC2LaunchTelemetry: {telemetry property}
```

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

EC2Launch v2 版本歷程記錄

版本歷程記錄

- EC2Launch v2 版本歷史記錄

EC2Launch v2 版本歷史記錄

要確保您已安裝最新的啟動代理程式，請參閱 安裝最新版本的 EC2Launch v2。您會在 EC2Launch v2 代理程式有新版本發佈時收到通知。如需詳細資訊，請參閱訂閱 EC2 Windows 啟動代理程式通知。

下表說明 EC2Launch v2 的發行版本。

版本	詳細資訊	發行日期
2.5.2	• 修正自訂憑證和群組政策部署憑證的 RDP 憑證驗證失敗的問題。	2026 年 8 月 5 日
2.5.1	• 改善啟用休眠時的錯誤偵測。	2026 年 6 月 3 日
2.5.0	• 更新序列連接埠邏輯，以在序列連接埠無法使用時繼續代理程式任務。 • 已將 EC2 Windows 公用程式裝置驅動程式版本新增至執行個體主控台輸出。 • 已移除未使用的凳子日誌檔案。 • 寫入執行個體主控台輸出時最佳化驅動程式查詢。	2026 年 5 月 4 日
2.4.0	• 透過移除備援 `console.log` 、 `err.log` 檔案來簡化代理程式記錄。 • 透過新增日誌輪換、將日誌大小限制為 1 MB，並將時間精確度提高到毫秒，來增強客服人員日誌記錄功能。 • 已移除 Hyper-V 狀態查詢和相關的執行個體主控台輸出。	2026 年 3 月 5 日
2.3.237	• 修正特定 Visual C++ 執行時間版本與代理程式遙測函數之間不相容所造成的代理程式當機問題。	2026 年 2 月 25 日

版本	詳細資訊	發行日期
2.3.108	• 已將 IPv6 地址顯示新增至桌布。如需記憶體和網路屬性，請參閱 AWS 文件。	2025 年 12 月 29 日
2.3.56	• 修正 EC2Launch v2 未正確處理預先設定持久性網路路由的問題。	2025 年 11 月 4 日
2.3.5	• 新增路由時，透過 IMDS 改進主要網路介面偵測功能。	2025 年 9 月 15 日
2.2.63	• 更新了啟動代理程式，以改進 遙測。	2025 年 7 月 29 日
2.1.1	• 新增對 IPv6 的完整支援。	2025 年 5 月 14 日
2.0.2107	• 增強了路由新增功能，以處理介面上未配置 IPv4 或 IPv6 位址的場景。	2025 年 3 月 27 日
2081 年 2 月 0 日	• 修正未正確擷取或驗證 RDP 憑證資訊的問題。已新增功能，可視需要自動啟動遠端桌面服務。 • 調整了 EC2Launch v2 服務許可，以修正查詢服務狀態時發生的問題。	2025 年 2 月 4 日
2046 年 2 月 0 日	• 更新 `agent-config.yml` 檔案中的桌布路徑，以使用預設作業系統桌布路徑。 • 新增遙測以監控代理程式發生錯誤的位置。 • 更新代理程式日誌訊息。	2024 年 10 月 3 日
2.0.1981	• 已更新針對非管理員使用者的 `EC2Launch.exe` CLI 命令錯誤訊息。	2024 年 8 月 6 日

版本	詳細資訊	發行日期
2.0.1948	• 新增遙測以監控管理員密碼選項的使用情況。 • 已修改 EC2Launch 許可。	2024 年 7 月 1 日
2.0.1924	• 已更新 EC2Launch 設定 UI。 • 已更新桌布 CLI 命令。 • 已更新 EC2Launch 安裝程式。	2024 年 6 月 10 日
2.0.1914	• 新增具有未指定閘道地址的路由 (適用於 IPv4 的 `0.0.0.0` 或適用於 IPv6 的 `::`)。 • 一律新增 IPv4 和 IPv6 路由。 • 修正未指定 `Administrator` 使用者名稱時新增至 `agent-config.yml` 檔案的問題。 • 已修改 EC2Launch v2 許可。	2024 年 6 月 5 日
2.0.1881	• 已將加密密碼選項新增至 `setAdminAccount` 任務。 • 已將 CLI 命令新增至 agent-config.yml 中的加密靜態密碼。 • 已修正 XML 在使用管理員許可執行時，使用者資料未新增 PowerShell 引數的問題。如需詳細資訊，請參閱Amazon EC2 如何處理 Windows 執行個體的使用者資料。 • 已調整當使用 `LocalSystem` 許可執行時，`executeScript` 任務和使用者資料指令碼的 PowerShell 引數。當引數為空時，代理程式會使用下列預設值：`-ExecutionPolicy Unrestricted`。 • 防止將重複的驅動程式版本列印到主控台日誌。	2024 年 5 月 8 日

版本	詳細資訊	發行日期
2.0.1815	• 已調整錯誤處理，以在 sysprep 之前失敗關鍵設定問題。 • 已修正桌布和主機名稱任務在執行個體上使用不正確的 IP 位址，並將多個 IP 位址指派給主要網路介面的問題。 • 桌布和主機名稱工作已變更為先從 IMDS 取得私有 IP，如果 IMDS 已停用，則回到 WMI。 • 已修正 `initializeVolume` 任務因暫時性錯誤導致 `sc1` 磁碟區無法初始化的問題。	2024 年 3 月 6 日
2.0.1739	• 已修正 Windows Administrator 使用者執行的 `executeScript` 工作無法擷取結束代碼的問題。	2024 年 1 月 17 日
2.0.1702	• 將 `Telemetry.log` 許可限制為 `read-execute` ，僅適用於標準使用者。 • 將 EC2Launch Windows 服務設定為在啟動失敗時重新啟動。 • 透過記錄 `route.exe` `stderr` 輸出，使 `add-routes` 故障可操作。 • 修正當路由指標超出 [1, 9999] 範圍時發生的問題。 • 為數種新的執行個體類型新增了桌布支援。 • 修正以 Windows 管理員使用者身分執行並將輸出傳送至 `stderr` 的使用者資料指令碼所造成的問題。	2024 年 1 月 4 日

版本	詳細資訊	發行日期
2.0.1643	• 將 ebsnvme-id.exe 工具更新至 1.1.0.7 版。 • 修正以 'metal-*' 開頭 (例如 metal-48x1) 的金屬執行個體類型的接收端調整 (RSS) 和接收佇列深度設定的問題。 • 已移除對封鎖代理程式之 XML 使用者資料命令進行報告的遙測事件。 • 已更新 setDnsSuffix 任務，以根據登錄項目限制網域名稱轉移：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel 。 • 新增一個公有任務以及可新增網路路由的 CLI。 • 注意 – 這是正式支援 Windows Server 2012 的最新版本。 • 注意 – 這是正式支援 32 位元作業系統的最新版本。	2023 年 10 月 4 日
2.0.1580	• 修改日誌檔案許可時，已變更啟動代理程式處理錯誤的方式。 • 已新增連線至序列埠資源的逾時時間。如果序列埠正在使用中，則逾時允許啟動代理程式繼續執行。	2023 年 9 月 5 日

版本	詳細資訊	發行日期

2.0.1521	• 已棄用 EC2Launch.exe 、reset 和 sysprep 命令的 –block 旗標。
	• EC2Launch.exe 已更新，可偵測並處理內嵌 executeScript 任務中使用的 reset 和 sysprep 命令。這些命令會導致代理程式在 executeScript 任務執行後停止執行。
	• 已更新 XML 使用者資料指令碼，預設情況下會內嵌執行。
	• 啟用 XML 使用者資料指令碼，以使用新的 detach 標籤在分離狀態下執行。如需詳細資訊，請參閱使用者資料指令碼。
	• 對代理程式日誌進行了下列變更。
	  • 更新代理程式日誌訊息。
	  • 從代理程式日誌中移除 executeScript 內容和輸出。
	  • 從代理程式日誌中移除 executeProgram 引數和輸出。
	• 對主控台日誌進行了下列變更。
	  • 為主控台日誌新增 EnableSCSIPersistentReservations 值。
		2023 年 7 月 3 日

版本	詳細資訊	發行日期
2.0.1303	• 新增網絡路由時新增了額外的錯誤處理和日誌行。 • 在 PreReady 階段中已允許 `executeScript` 和 `executeProgram` 任務。 • 已更新 `executeProgram` 任務，以產生類似於 executeScript 任務輸出的輸出檔案。如需詳細資訊，請參閱executeProgram。 • 已新增遙測，以監控 XML 使用者資料中封鎖代理程式命令的使用情況。	2023 年 5 月 3 日
2.0.1245	• 以純文字形式記錄當機呼叫堆疊，改善當機的可見性。 • 已新增 EventLog 服務作為啟動相依性，以在 Amazon EC2Launch 服務啟動速度比 EventLog 服務更快時修正當機問題。 • 讓 XML 使用者資料在 PostReady 階段之前從代理程式組態檔案 (例如 EC2Launch v1 和 EC2Config) 中執行。 • 已新增 YAML 使用者資料 1.1 版，讓使用者資料在 PostReady 階段之前從代理程式組態檔案中執行 (YAML 使用者資料 1.0 版在 PostReady 階段之後從代理程式組態檔案中執行)。	2023 年 3 月 8 日
2.0.1173	• 新增一個可選功能以在桌布上顯示執行個體標籤。如需詳細資訊，請參閱setWallpaper。 • 未正確設定 Elastic Graphics 的安全群組時，會新增錯誤處理。 • 修復未啟用執行個體中繼資料服務時的逾時。	2023 年 2 月 6 日

版本	詳細資訊	發行日期
2.0.1121	• 修正未指派公有 IPv4 地址時會將 404 錯誤列印至桌布的問題。 • 修復當裝置的磁碟機代號設為 D 時磁碟區檔案系統格式為 RAW 而非 NTFS 的問題。 • 修復 NVMe SSD 磁碟區錯誤識別為 EBS 磁碟區的問題。 • 修復在停用 IMDS 後啟動 Windows 時的錯誤。	2023 年 1 月 4 日
2.0.1082	• 修正停用 IMDS 時 `setWallpaper :privateIpAddress` 欄位為空白的問題。 • 修正停用 IMDS 時將主機名稱設定為私有 IPv4 地址的問題。 • 修正在 Windows Server 2012 上初始化磁碟區的問題。 • 修正設定巨型訊框的問題。 • 修正執行個體啟動時未指定 SSH 金鑰時的錯誤。 • 修正 Windows 沒有 'ReleaseId' 登錄機碼時 Windows Server 2012 上的錯誤。	2022 年 12 月 7 日
2.0.1011	• 修正 PnPDeviceID 為空白時尋找網路轉接器的邏輯。	2022 年 11 月 11 日
2.0.1009	• 使用 PCI 區段資訊來選取主控台連接埠。	2022 年 11 月 8 日

版本	詳細資訊	發行日期
2.0.982	• 新增重試邏輯以獲取 RDP 資訊。 • 修正 `d2.8xlarge` 執行個體的磁碟區初始化期間的錯誤。 • 修正在重新啟動後可能會選取錯誤網路轉接器的問題。 • 當 ACPI SPCR 無法使用時，移除誤報錯誤訊息。	2022 年 10 月 31 日
2.0.863	• 更新 IMDS 等待邏輯，以僅傳出 IMDSv2 請求。 • 新增邏輯，將磁碟機代號指派給已初始化但未掛載的磁碟區。 • 當 key pair type 不受支援時，印出更詳細的錯誤訊息。 • 修復 3010 重新啟動的程式碼錯誤。 • 新增檢查無效 base64 編碼的使用者資料功能。	2022 年 7 月 6 日
2.0.698	• 修復了執行指令碼時日誌輸出中的錯字。	2022 年 1 月 30 日
2.0.674	• 遙測會上傳已啟用/已停用的隱私權控制。 • 修正 `index out of bounds` 錯誤。 • 在 `sysprep` 期間移除桌布捷徑。	2021 年 11 月 15 日
2.0.651	• 新增邏輯以在安裝 EC2Launch v2 期間，解除安裝舊式代理程式。 • 修正根磁碟區未列為磁碟區 0 時的 `list-volume` CLI 問題。	2021 年 10 月 7 日

版本	詳細資訊	發行日期
2.0.592	• 修正錯誤以正確報告階段狀態。 • 當日誌檔關閉時，移除錯誤的警示錯誤訊息。 • 新增遙測。	2021 年 8 月 31 日
2.0.548	• 為十六進位 IP 主機名稱新增開頭的零。 • 修正 `enableOpenSsh` 任務的檔案許可。 • 修正 sysprep 命令當機。	2021 年 8 月 4 日
2.0.470	• 修正在網路階段中等待 DHCP 指派 IP 給執行個體的錯誤。 • 修正 `SearchList` 登錄機碼不存在時的 `setDnsSuffix` 錯誤。 • 修正 `setDnsSuffix` 中 DNS 轉移邏輯中的錯誤。 • 在中繼重新開機後新增網路路由。 • 允許 `initializeVolume` 重新用字母標示現有磁碟區。 • 從版本子命令中移除額外的資訊。	2021 年 7 月 20 日
2.0.285	• 在分離的程序中新增了執行使用者指令碼的選項。 • 舊式使用者資料 (XML 使用者資料) 現在會在分離的程序中執行，這與之前啟動代理程式類似的行為。 • 將 CLI 標記新增至 `sysprep` 和 `reset` 命令，這可讓其在服務停止之前進行封鎖。 • 限制組態資料夾許可。	2021 年 3 月 8 日

版本	詳細資訊	發行日期
2.0.207	• 將選擇性 hostName 欄位新增至 setHostName 工作。 • 修正重新開機錯誤。重新啟動任務 executeScript 並將 executeProgram 標記為執行中。 • 在狀態命令中新增更多傳回碼。 • 新增引導服務，以修正在執行 t2.nano 執行個體類型時的啟動問題。 • 修正全新安裝模式，以移除安裝程式未追蹤的檔案。	2021 年 2 月 2 日
2.0.160	• 修正 validate 命令，以偵測無效的階段名稱。 • 在 addroutes 任務中新增 w32tm resync 命令。 • 修正變更 DNS 尾碼搜尋順序的問題。 • 新增檢查條件，以更詳細地報告無效的使用者資料。	2020 年 12 月 4 日
2.0.153	在 UserData 中新增 Sysprep 功能。	2020 年 11 月 3 日

版本	詳細資訊	發行日期
2.0.146	• 修正在非英文 AMI 上的 RootExtend 的問題。 • 授與使用者群組寫入日誌檔案的權限。 • 為 GPT 磁碟區建立 MS 預留的磁碟分割。 • 在 Amazon EC2Launch 設定中，新增清單磁碟區命令和磁碟區下拉式清單。 • 新增 get-agent-config 命令，用於列印 yaml 或 json 格式的 agent-config.yml 檔案。 • 如果沒有偵測到公開金鑰，則清除靜態密碼。	2020 年 10 月 6 日
2.0.124	• 新增在桌布上顯示作業系統版本的選項。 • 初始化加密的 EBS 磁碟區。 • 為沒有本機 DNS 名稱的 VPC 新增路由。	2020 年 9 月 10 日
2.0.104	• 如果不存在，則建立 DNS 尾碼搜尋清單。 • 如果沒有要求，則可略過休眠。	2020 年 8 月 12 日
2.0.0	初始版本。	2020 年 6 月 30 日

在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v1 代理程式執行任務

適用於 Windows Server 2016 與 2019 的 Amazon 受管 AMI 包括名稱為 EC2Launch 的 Windows Powershell 指令碼集。EC2Launch 會在初始執行個體開機期間執行任務。如需有關包含在 AWS Windows AMIs 中的 EC2Launch 版本的資訊，請參閱 AWS Windows AMI 參考。

> **Note**
>
> Windows Server 2016 及更新作業系統版本的最新啟動代理程式為 EC2Launch v2，可取代 EC2Config 和 EC2Launch，並預先安裝於 AWS Windows Server 2016 與 2019 AMI，且名詞以 `EC2LaunchV2-Windows_Server-*` 開始。您亦可在 Windows Server 2016 與 2019 上手動安裝及設定代理程式。如需詳細資訊，請參閱安裝 EC2Launch v2。
> 為了搭配使用 EC2Launch 與 IMDSv2，版本必須為 1.3.2002730 或以上版本。

您可以使用下列 Windows PowerShell 命令可確認已安裝的 EC2Launch 版本。

```
Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1" | Select Version
```

EC2Launch 任務

根據預設，EC2Launch 會在初始執行個體開機期間執行下列任務：

- 設定呈現執行個體資訊的新桌面。
- 將電腦名稱設定為執行個體私有 IPv4 地址。
- 傳送執行個體資訊到 Amazon EC2 主控台。
- 傳送 RDP 憑證指紋到 EC2 主控台。
- 為管理員帳戶設定隨機密碼。
- 新增 DNS 尾碼。
- 動態擴充作業系統分割區，以包含任何未分割的空間。
- 執行使用者資料 (若指定)。如需指定使用者資料的詳細資訊，請參閱當使用使用者資料輸入啟動 EC2 執行個體時執行命令。
- 設定持續性靜態路由以到達中繼資料服務和 AWS KMS 伺服器。

> **Important**
>
> 如果從這個執行個體建立自訂 AMI，則會擷取這些路由做為作業系統組態的一部分，任何從該 AMI 啟動的新執行個體會保留相同路由，而不論子網放置何處。若要更新路由，請參閱啟動自訂 AMI 時更新 Server 2016 和更新版本的中繼資料/KMS 路由。

下列任務可協助維護與 EC2Config 服務回溯相容的功能。您也可以設定 EC2Launch 在啟動期間執行這些任務：

- 初始化輔助 EBS 磁碟區。
- 傳送 Windows 事件日誌到 EC2 主控台日誌。
- 傳送「Windows 已可使用」訊息到 EC2 主控台。

EC2Launch 目錄結構

根據預設，EC2Launch 會安裝於根目錄 `C:\ProgramData\Amazon\EC2-Windows\Launch` 中的 Windows Server 2016 和更新版本 AMI。

> **Note**
>
> 根據預設，Windows 會將檔案和資料夾隱藏在 `C:\ProgramData` 下。若要檢視 EC2Launch 目錄和檔案，您必須在 Windows 檔案總管中輸入路徑或變更資料夾屬性，才可顯示隱藏的檔案和資料夾。

`Launch` 目錄包含下列子目錄。

- `Scripts` – 含有構成 EC2Launch 的 PowerShell 指令碼。
- `Module` – 含有建置 Amazon EC2 相關指令碼的模組。
- `Config` – 含有您可以自訂的指令碼組態檔案。
- `Sysprep` – 含有 Sysprep 資源。
- `Settings` – 含有 Sysprep 圖形使用者介面的應用程式。
- `Library` - 包含 EC2 啟動代理程式的共用程式庫。
- `Log` – 包含指令碼的子目錄與指令碼產生的日誌檔案。

遙測

遙測是額外資訊，有助於 AWS 更了解您的需求、診斷問題和提供功能，以改善您的 AWS 服務體驗。

EC2Launch 版本 `1.3.2003498` 及更新版本會收集遙測，例如用量指標和錯誤。這些資料是從執行 EC2Launch 的 Amazon EC2 執行個體中收集的資料。這包括 擁有的所有 Windows AMIs AWS。

EC2Launch 會收集下列類型的遙測：

- 用量資訊 – 代理程式命令、安裝方法和排程的執行頻率。
- 錯誤和診斷資訊 – 代理程式安裝和執行錯誤代碼。

收集資料的範例：

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

預設為啟用遙測。您可隨時停用遙測收集。如果啟用遙測，EC2Launch 會傳送遙測資料，而不會傳送其他通知給客戶。

您選擇啟用或停用遙測收集。

您可以選擇加入或退出遙測收集。您選取加入或退出遙測收集，以確保我們遵守您的遙測選項。

遙測可見性

啟用遙測時，它會在 Amazon EC2 主控台輸出中顯示，如下所示：

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

停用執行個體的遙測

若要藉由設定系統環境變數來停用遙測，請以管理員身分執行下列命令：

```
setx /M EC2LAUNCH_TELEMETRY 0
```

若要在安裝期間停用遙測，請執行 `install.ps1`，如下所示：

```
. .\install.ps1 -EnableTelemetry:$false
```

EC2Launch 的更多主題

- 安裝最新版本的 EC2Launch
- 在 Windows 執行個體設定 EC2Launch v1 代理程式
- EC2Launch 版本歷史記錄

安裝最新版本的 EC2Launch

使用下列程序可在執行個體上下載和安裝 EC2Launch 最新版本。

下載和安裝 EC2Launch 最新版本

1. 如果您已在執行個體上安裝和設定 EC2Launch，請備份 EC2Launch 組態檔案。安裝程序不會保留此檔案的變更內容。根據預設，檔案位於 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。
2. 下載 EC2-Windows-Launch.zip 到執行個體上的目錄。
3. 下載 install.ps1 到您下載 `EC2-Windows-Launch.zip` 的同一個目錄。
4. 執行 `install.ps1`
5. 如果您備份了 EC2Launch 組態檔案，請將其複製到 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。

使用 PowerShell 下載和安裝最新版本 EC2Launch

如果您已在執行個體上安裝和設定 EC2Launch，請備份 EC2Launch 組態檔案。安裝程序不會保留此檔案的變更內容。根據預設，檔案位於 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。

若要使用 PowerShell 安裝最新版本 EC2Launch，請以管理員身分在 PowerShell 視窗執行下列命令：

```
mkdir $env:USERPROFILE\Desktop\EC2Launch
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/EC2-Windows-Launch.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Launch/latest/install.ps1"
$DownloadZipFile = "$env:USERPROFILE\Desktop\EC2Launch\" + $(Split-Path -Path $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
& $env:USERPROFILE\Desktop\EC2Launch\install.ps1
```

> **Note**
>
> 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016，則可能需要為 PowerShell 終端機啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
> ```

檢查啟動代理程式以確認安裝。以管理員身分透過 PowerShell 視窗執行下列命令：

```
Import-Module C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psm1
Import-LocalizedData -BaseDirectory C:\ProgramData\Amazon\EC2-Windows\Launch\Module\ -
FileName 'Ec2Launch.psd1' -BindingVariable moduleManifest
$moduleManifest.Get_Item('ModuleVersion')
```

在 Windows 執行個體設定 EC2Launch v1 代理程式

在您的執行個體第一次初始化後，您可以設定 EC2Launch 再次執行並執行其他啟動任務。

工作

- 設定初始化任務
- 排程 EC2Launch 在每次開機時都執行
- 初始化磁碟機和映射磁碟機代號
- 傳送 Windows 事件日誌到 EC2 主控台。
- 在成功開機後傳送「Windows 已就緒」訊息

設定初始化任務

在 `LaunchConfig.json` 檔案中指定設定以啟用或停用下列初始化任務：

- 將電腦名稱設定為執行個體私有 IPv4 地址。
- 將監視器設定為永遠保持開啟狀態。
- 設定新的桌面。
- 新增 DNS 尾碼清單。

> **Note**
>
> 這會新增下列網域的 DNS 尾碼查詢，並設定其他標準尾碼。如需啟動代理程式如何設定 DNS 尾碼的詳細資訊，請參閱 設定 EC2 Windows 啟動代理程式的 DNS 尾碼。
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- 擴充開機磁碟區大小。
- 設定管理員密碼。

進行初始化設定

1. 在要設定的執行個體上，於文字編輯器中開啟下列檔案：C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json。
2. 視需要更新下列設定並儲存變更。只有在 adminPassword 是 adminPasswordtype 時，才要在 Specify 中提供密碼。

```
{
 "setComputerName": false,
 "setMonitorAlwaysOn": true,
 "setWallpaper": true,
 "addDnsSuffixList": true,
 "extendBootVolumeSize": true,
 "handleUserData": true,
 "adminPasswordType": "Random | Specify | DoNothing",
 "adminPassword": "password that adheres to your security policy (optional)"
}
```

密碼類型的定義如下：

Random

EC2Launch 會產生密碼並以使用者金鑰將其加密。系統會在執行個體啟動後停用此設定，以便執行個體重新開機或停止再啟動時，能持續使用這個密碼。

`Specify`

EC2Launch 會使用您在 `adminPassword` 中指定的密碼。如果該密碼不符合系統要求，EC2Launch 會改為產生隨機密碼。系統會將該密碼以純文字存放於 `LaunchConfig.json`，並在 Sysprep 設定管理員密碼後予以刪除。EC2Launch 會以使用者金鑰加密該密碼。

`DoNothing`

EC2Launch 會使用您在 `unattend.xml` 檔案中指定的密碼。如果您未在 `unattend.xml` 中指定密碼，會停用管理員帳戶。

3. 在 Windows PowerShell 中，執行下列命令以排程指令碼做為 Windows 排程任務執行。該指令碼會在下一次開機期間執行一次，然後停用這些任務，以避免再次執行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

排程 EC2Launch 在每次開機時都執行

您可排程 EC2Launch 在每次開機時都執行，不是只在初次開機時執行。

啟用 EC2Launch 在每次開機時都執行

1. 開啟 Windows PowerShell 並執行下列命令：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

2. 或使用下列命令執行可執行檔：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe
```

然後選取 `Run EC2Launch on every boot`。您可指定 EC2 執行個體 `Shutdown without Sysprep` 或 `Shutdown with Sysprep`。

> **Note**
>
> 當您啟用 EC2Launch 以便在每次開機時執行時，下次 EC2Launch 執行時會發生下列情況：

- 如果 AdminPasswordType 仍然設定為 Random，EC2Launch 將在下次開機時產生新密碼。開機之後，AdminPasswordType 會自動設定為 DoNothing，以防止 EC2Launch 在後續開機時產生新密碼。若要防止 EC2Launch 在第一次開機時產生新密碼，請手動將 AdminPasswordType 設定為 DoNothing，然後再重新開機。
- HandleUserData 會設回 false，除非使用者資料將 persist 設成 true。如需詳細資訊，請參閱the section called “使用者資料指令碼”。

初始化磁碟機和映射磁碟機代號

在 DriveLetterMappingConfig.json 檔案中指定設定，以將磁碟機代號映射到您的 EC2 執行個體上的磁碟區。指令碼會初始化尚未初始化和分割的磁碟機。如需有關在 Windows 中取得磁碟區詳細資訊的更多詳細資訊，請參閱 Microsoft 文件中的 Get-Volume。

將磁碟機代號映射到磁碟區

1. 在文字編輯器中開啟 C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json 檔案。
2. 指定下列磁碟區設定並儲存您的變更：

```
{
 "driveLetterMapping": [
  {
   "volumeName": "sample volume",
   "driveLetter": "H"
  }
 ]
}
```

3. 開啟 Windows PowerShell，並使用下列命令執行初始化磁碟的 EC2Launch 指令碼：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

若要在每次執行個體開機時初始化磁碟，請新增 -Schedule 標記，如下所示：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

傳送 Windows 事件日誌到 EC2 主控台。

在 EventLogConfig.json 檔案中指定設定，以傳送 Windows 事件日誌到 EC2 主控台日誌。

進行設定以傳送 Windows 事件日誌

1. 在執行個體上，於文字編輯器中開啟 C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json 檔案。
2. 進行下列日誌設定並儲存您的變更：

```
{
 "events": [
  {
   "logName": "System",
   "source": "An event source (optional)",
   "level": "Error | Warning | Information",
   "numEntries": 3
  }
 ]
}
```

3. 在 Windows PowerShell 中執行下列命令，讓系統排程指令碼在每次執行個體開機時做為 Windows 排程任務執行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendEventLogs.ps1 -Schedule
```

日誌可能要花費三分鐘 (含) 以上才會顯示在 EC2 主控台日誌中。

在成功開機後傳送「Windows 已就緒」訊息

EC2Config 服務會在每次開機後，傳送「Windows 已就緒」訊息到 EC2 主控台。EC2Launch 只會在初次開機後傳送這個訊息。若要與 EC2Config 服務回溯相容，您可以排程 EC2Launch 在每次開機後傳送這個訊息。在執行個體上，開啟 Windows PowerShell 並執行下列命令。系統會排程指令碼做為 Windows 排程任務執行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -Schedule
```

EC2Launch 版本歷史記錄

若要下載和安裝 EC2Launch 的最新版本，請參閱安裝最新版本的 EC2Launch。

下表說明 EC2Launch 發行版本。

版本	詳細資訊	發行日期
1.4.299	• 用於擷取 EC2WinUtil 驅動程式版本的最佳化邏輯。 • 移除 Hyper-V 狀態查詢可縮短 Windows 的連線時間。	2026 年 3 月 3 日
1.4.183	• 更新安裝程式邏輯，以防止 Windows Server 2022 上不受支援的安裝。 • 更新以將 EC2WinUtil 驅動程式版本列印至執行個體主控台日誌。	2026 年 2 月 4 日
1.4.6	• 更新代理程式日誌訊息。	2025 年 10 月 13 日
1.3.2005119	• 修正在沒有任何參數的情況下呼叫 `Invoke-Userdata` 時出現失敗的問題。	2025 年 2 月 11 日
1.3.2005065	• 修正未正確擷取或驗證 RDP 憑證資訊的問題。已新增功能，可視需要自動啟動遠端桌面服務。	2024 年 10 月 22 日
1.3.2005008	• 如果找不到預設桌布映像，請將 `Set-Wallpaper` 更新回到單色背景。	2024 年 8 月 6 日
1.3.2004959	• 已更新安裝程式邏輯，以防止在 Windows Server 2025 或更新版本進行不受支援的安裝。	2024 年 7 月 2 日
1.3.2004891	• 修正了 `HandleUserData` 未按預期設定為`false` 的問題。 • 已將 `Encrypted` 密碼選項新增至 `LaunchConfig.json` 。 • 根據預設，變更 `Settings` UI 行為以加密使用者指定的密碼。	2024 年 5 月 31 日

版本	詳細資訊	發行日期
	• 新增 SetAdminPasswordConfig.ps1 ，將 Specify 密碼選項轉換為代理程式組態檔案 Encrypted 的密碼選項。	
1.3.2004617	• 修正設定桌布時的錯誤。	2024 年 1 月 15 日
1.3.2004592	• 已更新 install.ps1 為 %ProgramData%\Amazon\EC2-Windows\Launch 設定的存取許可。 • 已將 EC2Launch 資料夾/檔案存取限制為 read-execute，僅適用於標準使用者帳戶。 • 已變更代理程式以停止等待執行個體中繼資料服務 (IMDS) 來初始化 (若沒有為執行個體啟用 IMDS)。 • 在等待 IMDS 初始化時新增了五分鐘的逾時時間。 • 將代理程式變更為在 Windows is Ready 訊息之前 (而非之後) 將遙測寫入執行個體主控台日誌。 • 為數種新的執行個體類型新增了桌布支援。 如需 EC2Launch 目錄的存取許可及使用者帳戶許可詳細資訊，請參閱 the section called “EC2Launch 目錄結構”。	2024 年 1 月 2 日
1.3.2004491	• 新增遙測以監控指定管理員密碼選項的使用情況。	2023 年 11 月 9 日
1.3.2004462	• 在每次寫入序列主控台後新增排清。	2023 年 10 月 18 日

版本	詳細資訊	發行日期
1.3.2004438	• 根據登錄項目限制網域名稱轉移：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel 。 • 僅將 UserdataExecution.log 許可限制給 Administrators 。 • 當日誌初始化失敗時，在 Windows Event Log 中新增錯誤訊息。	2023 年 10 月 4 日
1.3.2004256	• 為主控台日誌新增 EnableSCSIPersistentReservations 值。 • 為 Get-ConsolePort 新增重試功能。	2023 年 7 月 7 日
1.3.2004052	• 修正在執行個體啟動時未指定 SSH 金鑰時出現的錯誤。 • 已更新為在失敗時重試啟動 AmazonSSMAgent Windows 服務。 • 已更新為如果 BeforeSysprep.cmd 失敗並顯示非零結束代碼，則 SysprepInstance.ps1 失敗。	2023 年 3 月 8 日
1.3.2003975	• 修正影響 Packer AMI 建置 (SysprepInstance.ps1 傳回 1 的 $LastErrorCode) 的問題。	2022 年 12 月 24 日

版本	詳細資訊	發行日期
1.3.2003961	• 已修正快速啟動執行個體上明確指定的管理員密碼會被隨機密碼覆寫的問題。 • 已修正 SSM Agent 無法在較小的執行個體類型上啟動的問題。 • 已修正執行個體主控台日誌包含 RDPCERTIFICATE-THUMBPRINT: 0000000000000000000000000 而非有效 RDP 憑證指紋值的問題。	2022 年 12 月 6 日
1.3.2003923	• 修正 PnPDeviceID 為空白時尋找網路轉接器的邏輯。	2022 年 11 月 9 日
1.3.2003919	• 已更新 Get-ConsolePort 以使用 PCI 區段資訊。 • 已修正在重新啟動後可能會選取錯誤網路轉接器的問題。 • 已修正 start-SSM-Agent 逾時邏輯。 • 已修正 Send-AdminCredentials 函數別名的回溯相容性。	2022 年 11 月 8 日
1.3.2003857	• 選取主要網路轉接器時，應優先考慮具有預設閘道的轉接器。 • 擴充了記憶體內密碼加密。	2022 年 10 月 3 日
1.3.2003824	• 已修正 setComputerName 期間的錯誤。 • 新增在偵測到 BYOL 帳單代碼時略過 Windows 啟用的邏輯。 • 新增記憶體內密碼加密。 • 已修正 m6id.4xlarge 磁碟區初始化期間的錯誤。	2022 年 8 月 30 日

版本	詳細資訊	發行日期
1.3.2003691	• 更新了 IMDS 等待邏輯，以僅發出 IMDSv2 請求。 • 修正了影響 eGPU 安裝的錯誤。	2022 年 6 月 21 日
1.3.2003639	• 新增了網路轉接器等待邏輯，以防止在初始化之前使用。 • 已修正輕微的問題。	2022 年 5 月 10 日
1.3.2003498	• 新增的遙測。 • 新增了設定 UI 的捷徑。 • 格式化 PowerShell 指令碼。 • 修復了在完成 BeforeSysprep.cmd 之前發生的關機問題。	2022 年 1 月 31 日
1.3.2003411	已變更密碼產生邏輯，以排除複雜性低的密碼。	2021 年 8 月 4 日
1.3.2003364	更新了支援 IMDSv2 的 Install-EgpuManager。	2021 年 6 月 7 日
1.3.2003312	• 在 `setMonitorAlwaysOn` 設定前後新增日誌行。 • 已將 AWS Nitro Enclaves 套件版本新增至主控台日誌。	2021 年 5 月 4 日
1.3.2003284	透過更新將使用者資料存放至 `LocalAppData` 的位置，來改善許可模型。	2021 年 3 月 23 日
1.3.2003236	• 已更新設定使用者密碼的方法於 `Set-AdminAccount` 和 `Randomize-LocalAdminPassword` 中。 • 修正 `InitializeDisks` ，以在將磁碟設定為可寫入之前檢查是否已設定為唯讀。	2021 年 2 月 11 日

版本	詳細資訊	發行日期
1.3.2003210	`install.ps1` 的本地化修正程式。	2021 年 1 月 7 日
1.3.2003205	更新 `%ProgramData%AmazonEC2-Windows\Launch\ModuleScripts` 目錄權限的 `install.ps1` 安全性修正程式。	2020 年 12 月 28 日
1.3.2003189	新增路由後新增 `w32tm resync`。	2020 年 12 月 4 日
1.3.2003155	已更新執行個體類型資訊。	2020 年 8 月 25 日
1.3.2003150	新增 `OsCurrentBuild` 和 `OsReleaseId` 至主控台輸出。	2020 年 4 月 22 日
1.3.2003040	修正 IMDS 第 1 版備用邏輯。	2020 年 4 月 7 日
1.3.2002730	已新增對 IMDS V2 的支援。	2020 年 3 月 3 日
1.3.2002240	已修正輕微的問題。	2019 年 10 月 31 日
1.3.2001660	修正在第一次執行 Sysprep 後未使用密碼的使用者自動登入的問題。	2019 年 7 月 2 日
1.3.2001360	已修正輕微的問題。	2019 年 3 月 27 日
1.3.2001220	所有 PowerShell 指令碼已簽署。	2019 年 2 月 28 日
1.3.2001200	已修正下列 InitializeDisks.ps1 問題：在 Windows Server 容錯移轉叢集的節點上執行指令碼，將格式化遠端節點上的磁碟機，而其磁碟機代號符合本機磁碟機代號。	2019 年 2 月 27 日

版本	詳細資訊	發行日期
1.3.2001160	已修正 Windows 2019 中遺漏桌布的問題。	2019 年 2 月 22 日
1.3.2001040	• 新增將顯示器設定為永遠不關閉的外掛程式，以修復 ACPI 問題。 • 寫入主控制台的 SQL Server 版本。	2019 年 1 月 21 日
1.3.2000930	修正在已啟用 ipv6 的 ENI 上新增路由到中繼資料的問題。	2019 年 1 月 2 日
1.3.2000760	• RSS 的新增預設組態和 ENA 裝置的接收佇列設定。 • Sysprep 時停用休眠。	2018 年 12 月 5 日
1.3.2000630	• DNS 伺服器新增路由 169.254.169.253/32。 • 設定管理使用者的新增篩選條件 • 執行個體休眠改進。 • 排程 EC2Launch 在每次開機時都執行的新增選項。	2018 年 11 月 9 日
1.3.2000430.0	• 已新增路由 169.254.169.123/32 至 AMZN 時間服務。 • 已新增路由 169.254.169.249/32 至 GRID 授權服務。 • 嘗試啟動 Systems Manager 時新增了 25 秒的逾時。	2018 年 9 月 19 日
1.3.200039.0	• 已修正 EBS NVME 磁碟區不正確的磁碟機代號。 • 已新增 NVME 驅動程式版本的其他記錄。	2018 年 8 月 15 日
1.3.2000080	已修正輕微的問題。	

版本	詳細資訊	發行日期
1.3.610	已修正將輸出和錯誤重新導向到使用者資料之檔案的問題。	
1.3.590	• 在桌面中新增了缺少的執行個體類型。 • 已修正磁碟機代號映射和磁碟安裝的問題。	
1.3.580	• 已修正 `Get-Metadata` ，以對 Web 請求使用預設系統代理設定。 • 在磁碟初始化中新增了 NVMe 的特殊案例。 • 已修正輕微的問題。	
1.3.550	新增了 `-NoShutdown` 選項以啟用 Sysprep 而不關機。	
1.3.540	已修正輕微的問題。	
1.3.530	已修正輕微的問題。	
1.3.521	已修正輕微的問題。	
1.3.0	• 已修正電腦名稱變更的十六進位長度問題。 • 已修正電腦名稱變更可能發生重新開機迴圈的問題。 • 已修正桌面設定中的問題。	
1.2.0	• 更新以顯示有關 EC2 系統日誌中安裝之作業系統 (OS) 的資訊。 • 更新以顯示 EC2 系統日誌中的 EC2Launch 和 SSM Agent 版本。 • 已修正輕微的問題。	

版本	詳細資訊	發行日期
1.1.2	• 更新以顯示 EC2 系統日誌中的 ENA 驅動程式資訊。 • 更新以從主要 NIC 篩選邏輯排除 Hyper-V。 • 已將 AWS KMS 伺服器和連接埠新增至 KMS 啟用的登錄機碼。 • 已改善多位使用者的桌面設定。 • 更新以清除來自持久性存放區的路由。 • 更新以從 DNS 尾碼清單中的可用區域移除 z。 • 更新以處理使用者資料中 <runAsLocalSystem> 標籤的問題。	
1.1.1	初始版本.	

在 EC2 舊版 Windows 作業系統執行個體啟動期間，使用 EC2Config 服務執行任務

> **Note**
>
> EC2Config 已結束支援。Microsoft 不再支援其執行的作業系統版本。強烈建議您升級至最新的啟動代理程式。
> Windows Server 2022 和更新作業系統版本的最新啟動代理程式為 EC2Launch v2，可取代 EC2Config 和 EC2Launch，並預先安裝在 AWS Windows Server 2022 和 2025 AMIs 上。您亦可在 Windows Server 2016 與 2019 上手動安裝及設定代理程式。如需詳細資訊，請參閱安裝 EC2Launch v2。

適用於 Windows Server 2016 之前的 Windows Server 版本的 Windows AMI 包含一項選用服務，即 EC2Config 服務 (`EC2Config.exe`)。EC2Config 會在啟動期間啟動執行個體及執行作業時及每次您停止或啟動執行個體時啟動。EC2Config 也可依據需求執行作業。其中部分作業為自動啟用，其他作業則必須手動啟用。雖然此項服務為選用，其中可存取專屬的進階功能。此服務在 LocalSystem 帳戶內執行。

EC2Config 服務將執行 Sysprep，此為 Microsoft 的工具，可讓您建立自訂且可重複使用的 Windows AMI。EC2Config 呼叫 Sysprep 時，會使用 `%ProgramFiles%\Amazon\EC2ConfigService\Settings` 中的檔案來決定要執行的操作。您可間接用 EC2 服務屬性系統對話方塊編輯這些檔案，或直接用 XML 編輯器或文字編輯器編輯這些檔案。但有些進階設定無法在 EC2 服務屬性系統對話方塊中使用，因此您必須直接編輯這些項目。

如果您在更新執行個體設定後從執行個體建立 AMI，新設定將套用到從新 AMI 啟用的任何執行個體。如需有關建立 AMI 的詳細資訊，請參閱 建立 Amazon EBS 支援的 AMI。

EC2Config 使用設定檔案控制其運作。您可使用圖形工具或直接編輯 XML 檔案，以更新這些設定檔案。服務的二進位檔案和其他檔案包含在 `%ProgramFiles%\Amazon\EC2ConfigService` 目錄中。

目錄

- EC2Config 和 AWS Systems Manager
- EC2Config 作業
- EC2Config 設定檔
- 安裝最新版本的 EC2Config
- 設定 EC2Config 服務的 .NET 代理程式設定
- 從 EC2 Windows 執行個體的系統對話方塊設定 EC2Config 服務屬性
- 疑難排解使用 EC2Config 啟動代理程式的問題
- EC2Config 版本記錄

EC2Config 和 AWS Systems Manager

EC2Config 服務會處理 Windows Server 2016 先前 Windows Server 版本 (2016 年 11 月之前發佈) 之 AMI 所建立執行個體的 Systems Manager 請求。

2016 年 11 月之後發佈的 Windows Server 2016 先前 Windows Server 版本的 AMI 所建立的執行個體，包含 EC2Config 服務和 SSM Agent。EC2Config 將執行先前所述的所有作業，而且 SSM Agent 將處理 Systems Manager 功能的請求，例如執行命令和狀態管理員。

您可使用執行命令，將使用的現有執行個體升級到 EC2Config 服務和 SSM Agent 的最新版本。若要了解詳細資訊，請參閱 AWS Systems Manager 使用者指南中的使用執行命令更新 SSM 代理程式。

EC2Config 作業

EC2Config 會在執行個體初次啟動時執行初始啟動作業，接著停用執行個體。若要重新執行這些作業，您必須先明確啟用執行個體，然後再關閉執行個體，或手動執行 Sysprep。這些作業如下：

- 設定管理員帳戶的隨機加密密碼。
- 產生並安裝遠端桌面連線所用的主機憑證。
- 動態擴充作業系統的分割區，以包含任何尚未分割的空間。
- 執行指定的使用者資料 (和 Cloud-Init，如果尚未安裝)。如需指定使用者資料的詳細資訊，請參閱當使用使用者資料輸入啟動 EC2 執行個體時執行命令。

EC2Config 會在執行個體每次啟動時執行下列作業：

- 變更主機名稱以符合十六進位表示法中的私有 IP 地址 （此任務預設為停用，且必須在執行個體啟動時啟用才能執行）。
- 設定金鑰管理伺服器 (AWS KMS)，檢查 Windows 啟用狀態，並視需要啟動 Windows。
- 掛載所有的 Amazon EBS 磁碟區和執行個體儲存體磁碟區，然後將磁碟區名稱對應至磁碟機代號。
- 將事件日誌項目寫入主控台以協助故障診斷 （此任務預設為停用，且必須在執行個體啟動時啟用才能執行）。
- 將 Windows 已就緒的狀態寫入主控台。
- 新增自訂路由表至主要網路轉接器，以在連接單個 NIC 或多個 NIC 時啟用下列的 IP 地址：`169.254.169.250`、`169.254.169.251` 和 `169.254.169.254`。這些地址供 Windows 啟用和當您要存取執行個體中繼資料時使用。

> **Note**
>
> 如果 Windows 作業系統設定為使用 IPv4，則可以使用這些 IPv4 link-local 地址。如果 Windows 作業系統已停用 IPv4 網路通訊協定堆疊，並改用 IPv6，請新增 `[fd00:ec2::250]` 取代 `169.254.169.250` 和 `169.254.169.251`。然後新增 `[fd00:ec2::254]` 取代 `169.254.169.254`。

EC2Config 會在每次有使用者登入時執行下列作業：

- 在桌面背景顯示桌布資訊。

執行個體執行時，您可要求 EC2Config 隨需執行下列作業：

- 執行 Sysprep 並關閉執行個體，如此才能從中建立 AMI。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。

EC2Config 設定檔

設定檔可控制 EC2Config 服務的操作。這些檔案位在 `C:\Program Files\Amazon\Ec2ConfigService\Settings` 目錄：

- `ActivationSettings.xml` - 使用金鑰管理伺服器 (AWS KMS) 控制產品啟用。
- `AWS.EC2.Windows.CloudWatch.json`—控制要傳送至 CloudWatch 的效能計數器和要傳送至 CloudWatch Logs 的日誌。
- `BundleConfig.xml`—控制 EC2Config 如何準備執行個體後端執行個體以用於建立 AMI。
- `Config.xml`—控制主要設定。
- `DriveLetterConfig.xml`—控制磁碟機代號映射。
- `EventLogConfig.xml`—控制執行個體開機時顯示在主控台上的事件日誌資訊。
- `WallpaperSettings.xml`—控制顯示在桌面背景上的資訊。

ActivationSettings.xml

此檔案包含控制產品啟用的設定。Windows 開機時，EC2Config 服務會檢查 Windows 是否已啟動。如果 Windows 尚未啟動，服務會搜尋指定的 AWS KMS 伺服器，嘗試啟動 Windows。

- `SetAutodiscover` - 表示是否自動偵測 AWS KMS 。
- `TargetKMSServer`- 儲存 的私有 IP 地址 AWS KMS。 AWS KMS 和您的執行個體必須在相同的區域。
- `DiscoverFromZone`- 從指定的 DNS 區域探索 AWS KMS 伺服器。
- `ReadFromUserData`- 從 UserData 取得 AWS KMS 伺服器。
- `LegacySearchZones`- 從指定的 DNS 區域探索 AWS KMS 伺服器。
- `DoActivate`—使用區段中的指定設定嘗試啟用。此值可以為 `true` 或 `false`。
- `LogResultToConsole`—將結果顯示在主控台。

BundleConfig.xml

此檔案包含控制 EC2Config 如何準備執行個體以用於建立 AMI 的設定。

- AutoSysprep—指出是否自動使用 Sysprep。變更值為 Yes 將使用 Sysprep。
- SetRDPCertificate—設定遠端桌面伺服器的自我簽署憑證。這可安全地傳送 RDP 至執行個體。如果新執行個體應取得該憑證，請變更值為 Yes。

  此設定不適用於作業系統版本早於 Windows Server 2016 的執行個體，因為這些作業系統可以產生自己的憑證。
- SetPasswordAfterSysprep—在新啟動的執行個體上設定隨機密碼，以使用者啟動金鑰加密，然後將加密的密碼輸出到主控台。如果新執行個體不應設定為隨機加密密碼，請變更此設定的值為 No。

Config.xml

外掛程式

- Ec2SetPassword—每次啟動執行個體時產生隨機加密密碼。此功能自初次啟動後將預設停用，因此重新啟動本執行個體不會變更使用者設定的密碼。變更此設定為 Enabled，將繼續在每次啟動執行個體時產生密碼。

  如果您計劃要從執行個體建立 AMI，此設定極為重要。
- Ec2SetComputerName—依執行個體的 IP 地址將執行個體的主機名稱設為唯一名稱，並重新啟動執行個體。若要自行設定主機名稱，或避免現有的主機名稱遭到修改，您必須停用此設定。
- Ec2InitializeDrives—在啟動期間初始化及格式化所有磁碟區。此功能預設為啟用。
- Ec2EventLog—在主控台顯示事件日誌項目。根據預設，會顯示系統事件日誌中最近的三個錯誤項目。若要指定要顯示的事件日誌項目，請編輯 EventLogConfig.xml 目錄內的 EC2ConfigService\Settings 檔案。若要了解此檔案中設定的相關資訊，請參閱 Eventlog 金鑰。
- Ec2ConfigureRDP—在執行個體上設定自我簽署憑證，讓使用者可使用遠端桌面安全存取執行個體。此設定不適用於作業系統版本早於 Windows Server 2016 的執行個體，因為這些作業系統可以產生自己的憑證。
- Ec2OutputRDPCert—顯示遠端桌面憑證資訊至主控台，讓使用者可對照指紋進行確認。
- Ec2SetDriveLetter—依使用者定義的設定來設定掛載磁碟區的磁碟機代號。根據預設，Amazon EBS 磁碟區掛載至執行個體時，可用執行個體上的磁碟機代號掛載。若要指定磁碟機代號映射，請編輯 DriveLetterConfig.xml 目錄內的 EC2ConfigService\Settings 檔案。

- `Ec2WindowsActivate`—外掛程式將處理 Windows 啟用。外掛程式會檢查 Windows 是否啟動。如果沒有，它會更新 AWS KMS 用戶端設定，然後啟用 Windows。

  若要修改 AWS KMS 設定，請編輯位於 `EC2ConfigService\Settings`目錄中`ActivationSettings.xml`的檔案。
- `Ec2DynamicBootVolumeSize`—擴充 Disk 0/Volume 0，以包含任何尚未分割的空間。
- `Ec2HandleUserData`—在 Sysprep 執行後建立並執行使用者於初次啟動執行個體時建立的指令碼。指令碼標籤括住的命令將儲存至批次檔，PowerShell 標籤括住的命令將儲存至 .ps1 檔案 (對應至 Ec2 服務屬性系統對話方塊上的使用者資料核取方塊)。
- `Ec2ElasticGpuSetup`—若執行個體與彈性 GPU 相關聯，則安裝彈性 GPU 軟體套件。
- `Ec2FeatureLogging`—將 Windows 功能安裝和對應的服務狀態傳送至主控台。僅支援 Microsoft Hyper-V 功能和對應的 vmms 服務。

全域設定

- `ManageShutdown` – 確定 Amazon S3 支援的 AMI 啟動的執行個體不會在執行 Sysprep 時終止。
- `SetDnsSuffixList`—設定 Amazon EC2 網路轉接器的 DNS 尾碼。如此可對執行於 Amazon EC2 的伺服器進行 DNS 解析，而不用提供完全合格的網域名稱。

> **Note**
>
> 這會新增下列網域的 DNS 尾碼查詢，並設定其他標準尾碼。如需啟動代理程式如何設定 DNS 尾碼的詳細資訊，請參閱 設定 EC2 Windows 啟動代理程式的 DNS 尾碼。
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- `WaitForMetaDataAvailable`—確定 EC2Config 服務將等待直到中繼資料變成可存取且網路變成可用，然後再繼續開機。此檢查會確定 EC2Config 可從中繼資料取得用於啟用和其他外掛程式的資訊。
- `ShouldAddRoutes`—新增自訂路由表至主要網路轉接器，以在連接多個 NIC 時啟用下列的 IP 地址：169.254.169.250、169.254.169.251 和 169.254.169.254。這些地址供 Windows 啟用和當您要存取執行個體中繼資料時使用。
- `RemoveCredentialsfromSyspreponStartup`—下次服務啟動時從 `Sysprep.xml` 移除管理員密碼。若要確保此密碼保持一致，請編輯此設定。

DriveLetterConfig.xml

此檔案包含控制磁碟機代號映射的設定。根據預設，磁碟區可映射至任何可用的磁碟機代號。您可將磁碟區掛載至特定的磁碟機代號，如下所示。

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- VolumeName—磁碟區標籤。例如 *My Volume*。若要指定執行個體儲存體磁碟區的映射，請使用 Temporary Storage X 標籤，其中的 X 為 0 至 25 的數字。
- DriveLetter—磁碟機代號。例如 *M:*。假如磁碟機代號已在使用中，映射便會失敗。

EventLogConfig.xml

此檔案包含控制執行個體開機時顯示在主控台上之事件日誌資訊的設定。根據預設，會顯示系統事件日誌中最近的三個錯誤項目。

- Category—傳送至主控台的事件日誌金鑰。
- ErrorType——事件類型 (例如 Error、Warning、Information。)
- NumEntries—此類別存放的事件數量。
- LastMessageTime—為避免重複推送相同的訊息，服務會在每次推送訊息時更新此值。
- AppName—事件來源或記錄事件的應用程式。

WallpaperSettings.xml

此檔案包含控制顯示在桌面背景上之資訊的設定。預設顯示下列資訊。

- Hostname—顯示電腦名稱。
- Instance ID—顯示執行個體的 ID。

- Public IP Address—顯示執行個體的公有 IP 地址。
- Private IP Address—顯示執行個體的私有 IP 地址。
- Availability Zone—顯示執行個體正在執行的可用區域。
- Instance Size—顯示執行個體的類型。
- Architecture—顯示 PROCESSOR_ARCHITECTURE 環境變數的設定。

移除此項目，將移除預設顯示的任何資訊。您可新增額外要顯示的執行個體中繼資料，如下所示。

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

您可新增額外要顯示的系統環境變數，如下所示。

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

此檔案包含控制 EC2Config 如何初始化磁碟機的設定。

根據預設，EC2Config 會初始化作業系統未將其設為上線的磁碟機。您可自訂外掛程式，如下所示。

```
<InitializeDrivesSettings>
    <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

使用設定群組指定初始化磁碟機的方式：

FormatWithTRIM

在格式化磁碟機時啟用 TRIM 命令。磁碟機完成格式化和初始化後，系統將還原 TRIM 組態。

自 EC2Config 3.18 版起，TRIM 命令在磁碟格式化操作期間預設為停用，如此將可改善格式化的時間。針對 EC2Config 3.18 版和更新版本，請使用此設定啟用磁碟格式化操作期間的 TRIM 命令。

FormatWithoutTRIM

格式化磁碟機時停用 TRIM 命令，以改善 Windows 的格式化時間。磁碟機完成格式化和初始化後，系統將還原 TRIM 組態。

DisableInitializeDrives

停用新磁碟機的格式化。請用此設定手動初始化磁碟機。

安裝最新版本的 EC2Config

> **Note**
>
> Windows Server 2022 及更新作業系統版本的最新啟動代理程式為 EC2Launch v2，可取代 EC2Config 和 EC2Launch。EC2Launch v2 已預先安裝在 AWS Windows Server 2022 和 2025 AMIs 上。您亦可在 Windows Server 2016 與 2019 上手動安裝及設定代理程式。如需詳細資訊，請參閱安裝 EC2Launch v2。

如需如何接收 EC2Config 更新通知的詳細資訊，請參閱訂閱 EC2 Windows 啟動代理程式通知。如需各版本變更的詳細資訊，請參閱EC2Config 版本記錄。

開始之前

- 確認您擁有 .NET Framework 3.5 SP1 或更新版本。
- 根據預設，安裝程式會在安裝期間將您的設定檔換成預設設定，並在安裝完成後重新啟動 EC2Config 服務。如果變更 EC2Config 服務設定，請從 `%Program Files%\Amazon\Ec2ConfigService\Settings` 目錄複製 `config.xml` 檔案。更新 EC2Config 服務後，您可還原此檔案，以保留設定變更。

確認 EC2Config 版本

請用下列程序確認執行個體上安裝的 EC2Config 版本。

確認安裝的 EC2Config 版本

1. 從 AMI 啟動執行個體，並連線至執行個體。
2. 從 Control Panel (控制台) 中，選取 Programs and Features (程式和功能)。
3. 從已安裝程式清單中，找出 `Ec2ConfigService`。其版本編號將出現在 Version (版本) 欄中。

更新 EC2Config

請用下列程序在執行個體上下載並安裝最新版的 EC2Config。

下載並安裝最新版的 EC2Config

1. 下載並解壓縮 EC2Config 安裝程式。
2. 執行 `EC2Install.exe`。如需完整的選項清單，請執行 `EC2Install` 並加入 `/?` 選項。根據預設，設定程式會顯示提示。若要執行命令且不加入提示，請使用 `/quiet` 選項。

   > **Important**
   >
   > 若要保留所儲存之 `config.xml` 檔案中的自訂設定，請執行 `EC2Install` 並加入 `/norestart` 選項，還原設定，然後手動重新啟動 EC2Config 服務。

3. 如果您執行的是 EC2Config 4.0 版或更新版本，您必須從 Microsoft Services 內嵌在執行個體上重新開始 SSM Agent。

   > **Note**
   >
   > 您重新啟動或者停止並啟動執行個體後，執行個體系統日誌或 Trusted Advisor 檢查中才會顯示更新的 EC2Config 版本資訊。

若要使用 PowerShell 下載並安裝最新版本 EC2Config

若要使用 PowerShell 下載、解壓縮並安裝最新版本 EC2Config，請在 PowerShell 視窗執行下列命令：

```
$Url = "https://s3.amazonaws.com/ec2-downloads-windows/EC2Config/EC2Install.zip"
$DownloadZipFile = "$env:USERPROFILE\Desktop\" + $(Split-Path -Path $Url -Leaf)
$ExtractPath = "$env:USERPROFILE\Desktop\"
Invoke-WebRequest -Uri $Url -OutFile $DownloadZipFile
$ExtractShell = New-Object -ComObject Shell.Application
$ExtractFiles = $ExtractShell.Namespace($DownloadZipFile).Items()
$ExtractShell.NameSpace($ExtractPath).CopyHere($ExtractFiles)
Start-Process $ExtractPath
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\EC2Install.exe `
    -ArgumentList "/S"
```

> **Note**
>
> 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
> ```

查看 `C:\Program Files\Amazon\` 是否有 `Ec2ConfigService` 目錄以驗證安裝。

設定 EC2Config 服務的 .NET 代理程式設定

您可以將 EC2Config 服務設定為使用下列其中一種方法透過代理通訊：適用於 .NET 的 AWS SDK、`system.net`元素或 Microsoft 群組政策和 Internet Explorer。使用適用於 .NET 的 AWS SDK 是偏好的方法，因為您可以指定登入憑證。

方法

- 使用 適用於 .NET 的 AWS SDK （偏好） 設定代理設定
- 使用 system.net 元素設定代理伺服器設定
- 使用 Microsoft 群組政策和 Microsoft Internet Explorer 設定代理伺服器設定

使用 適用於 .NET 的 AWS SDK （偏好） 設定代理設定

您可指定 `proxy` 檔案中的 `Ec2Config.exe.config` 元素，以設定 EC2Config 服務的 proxy 設定。如需詳細資訊，請參閱適用於 .NET 的 AWS SDK 組態檔案參考。

指定 Ec2Config.exe.config 中的代理伺服器元素

1. 在要讓 EC2Config 服務透過 proxy 通訊的執行個體上，編輯 `Ec2Config.exe.config` 檔案。根據預設，檔案位在下列目錄：`%ProgramFiles%\Amazon\Ec2ConfigService`。
2. 將以下 `aws` 元素新增到 `configSections`。請勿將此元素新增到任何現有的 `sectionGroups`。

   針對 EC2Config 3.17 版或更早版本

   ```
   <configSections>
   ```

```
   <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
</configSections>
```

針對 EC2Config 3.18 版或更新版本

```
<configSections>
     <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
</configSections>
```

3. 將以下 aws 元素新增到 Ec2Config.exe.config 檔案。

```
<aws>
   <proxy
     host="string value"
     port="string value"
     username="string value"
     password="string value" />
</aws>
```

4. 儲存您的變更。

使用 system.net 元素設定代理伺服器設定

您可在 system.net 檔案的 Ec2Config.exe.config 元素中指定 proxy 的設定。若要了解詳細資訊，請參閱 defaultProxy 元素 (網路設定)。

指定 Ec2Config.exe.config 中的 system.net 元素

1. 在要讓 EC2Config 服務透過 proxy 通訊的執行個體上，編輯 Ec2Config.exe.config 檔案。根據預設，檔案位在下列目錄：%ProgramFiles%\Amazon\Ec2ConfigService。
2. 新增 defaultProxy 項目至 system.net。若要了解詳細資訊，請參閱 defaultProxy 元素 (網路設定)。

   例如，下列設定將路由所有流量，以使用 Internet Explorer 目前設定的 proxy，但不包含中繼資料和授權流量，這些流量將繞過代理伺服器。

```
<defaultProxy>
    <proxy usesystemdefault="true" />
    <bypasslist>
        <add address="169.254.169.250" />
        <add address="169.254.169.251" />
```

```
        <add address="169.254.169.254" />
        <add address="[fd00:ec2::250]" />
        <add address="[fd00:ec2::254]" />
    </bypasslist>
</defaultProxy>
```

3. 儲存您的變更。

使用 Microsoft 群組政策和 Microsoft Internet Explorer 設定代理伺服器設定

EC2Config 服務在 Local System 使用者帳戶下執行。在執行個體上變更群組政策設定後，您可在 Internet Explorer 中為此帳戶指定執行個體範圍的代理伺服器設定。

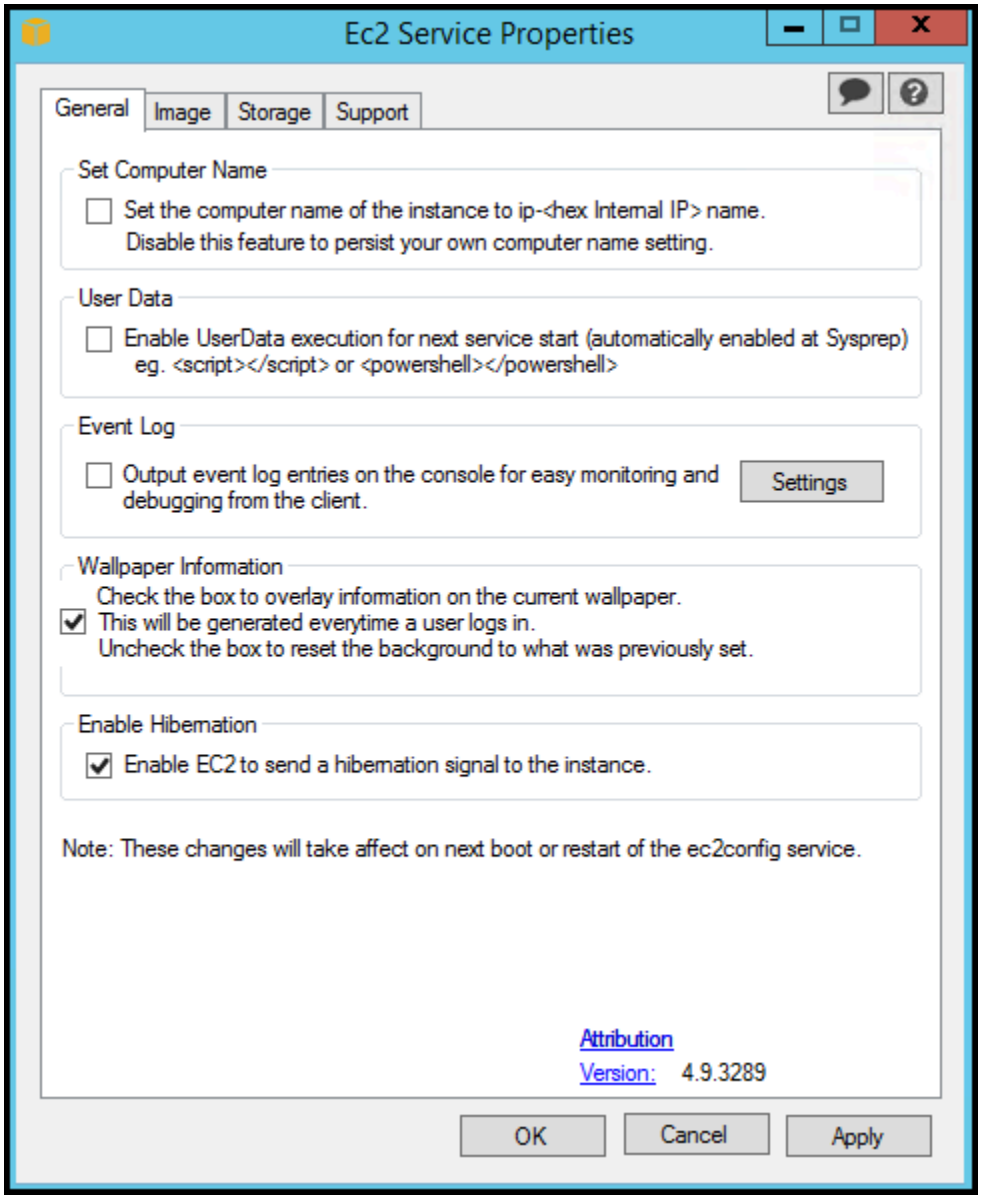
使用群組政策和 Internet Explorer 設定代理伺服器設定

1. 在要讓 EC2Config 服務透過 proxy 通訊的執行個體上，以管理員身分開啟命令提示，輸入 **`gpedit.msc`** 並按 Enter 鍵。
2. 在本機群組政策編輯器中的 Local Computer Policy (本機電腦政策) 底下，選擇 Computer Configuration (電腦設定)、Administrative Templates (系統管理範本)、Windows Components (Windows 元件)、Internet Explorer。
3. 在右側窗格中，選擇 Make proxy settings per-machine (rather than per-user) (依每一部電腦設定 proxy 設定 (而不是依每一位使用者))，然後選擇 Edit policy setting (編輯政策設定)。
4. 選擇 Enabled (已啟用)，然後選擇 Apply (套用)。
5. 開啟 Internet Explorer，然後選擇 Tools (工具) 按鈕。
6. 選擇 Internet Option (網際網路選項)，然後選擇 Connections (連線) 索引標籤。
7. 選擇 LAN settings (LAN 設定)。
8. 在 Proxy server (代理伺服器) 底下，選擇 Use a proxy server for your LAN (為您的 LAN 使用代理伺服器) 選項。
9. 指定地址和連接埠資訊，然後選擇 OK (確定)。

從 EC2 Windows 執行個體的系統對話方塊設定 EC2Config 服務屬性

下列程序說明如何使用 EC2 服務屬性系統對話方塊啟用或停用設定。

1. 啟動和連線至 Windows 執行個體。
2. 從開始功能表中，選擇所有程式，然後選擇 EC2ConfigService 設定。



3. 在 EC2 服務屬性系統對話方塊的一般索引標籤，啟用或停用下列設定。

   Set Computer Name (設定電腦名稱)

   如果啟用此設定 (預設為停用)，每次開機時會將主機名稱與目前的內部 IP 地址比對；如果主機名稱與內部 IP 地址不符，將重設主機名稱以包含內部 IP 地址，接著系統會重新啟動，以選用新的主機名稱。若要自行設定主機名稱，或避免現有的主機名稱遭到修改，請勿啟用此設定。

   使用者資料

   使用者資料執行可讓您指定執行個體中繼資料中的指令碼。根據預設，這些指令碼會在初始啟動時執行。您也可將其設定為在下次重新啟動或啟動執行個體時，或每次重新啟動或啟動執行個體時執行。

如果指令碼較大，建議透過使用者資料下載指令碼，然後執行指令碼。

如需詳細資訊，請參閱使用者資料執行。

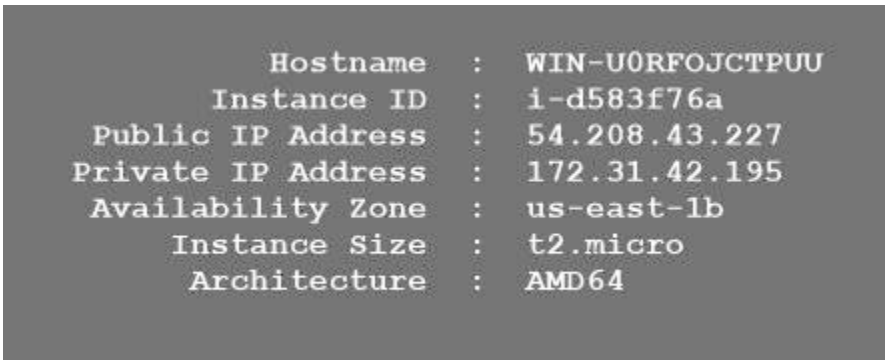
事件日誌

使用此設定可於開機時在主控台上顯示事件日誌項目，以方便進行監控和除錯。

選擇設定，為傳送至主控台的日誌項目指定篩選條件。預設篩選條件會傳送系統事件日誌中最近的三個錯誤項目至主控台。

Wallpaper Information (桌布資訊)

使用此設定可在桌面背景顯示系統資訊。以下是顯示在桌面背景上的資訊範例。

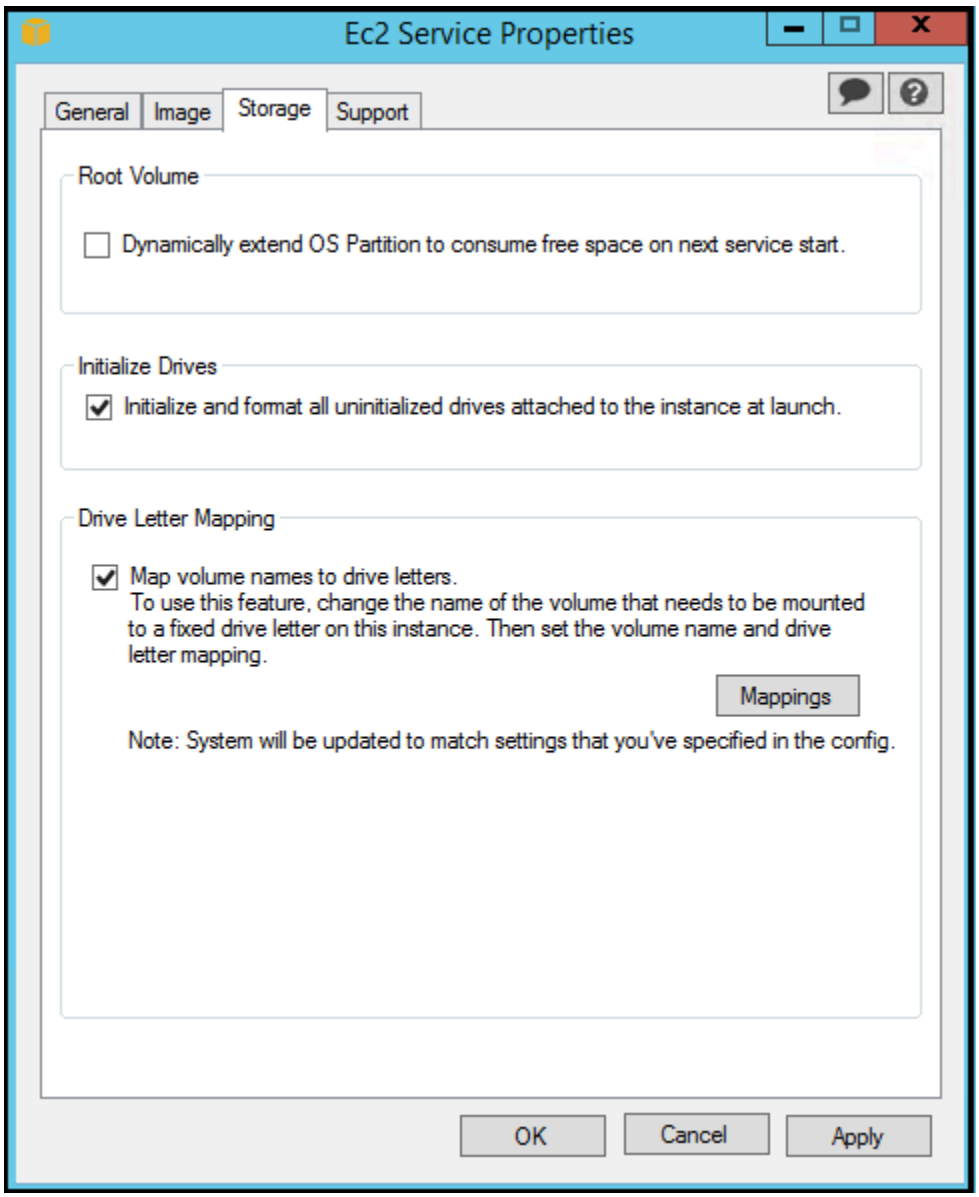


顯示在桌面背景上的資訊由設定檔 `EC2ConfigService\Settings\WallpaperSettings.xml` 控制。

啟用休眠

使用此設定，允許 EC2 向作業系統發出執行休眠的信號。

4. 選擇儲存分頁。您可以啟用或停用以下設定。



Root Volume (根磁碟區)

此設定將動態擴充 Disk 0/Volume 0，以包含任何尚未分割的空間。此設定在執行個體是從自訂大小的根磁碟區啟動時很有用。

Initialize Drives (初始化磁碟機)

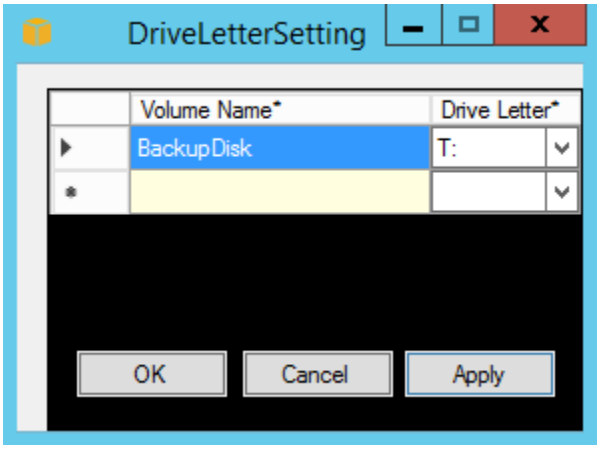
此設定會在啟動時格式化及掛載連接至執行個體的所有磁碟區。

Drive Letter Mapping (磁碟機代號映射)

系統會對應連接至執行個體的磁碟區與磁碟機代號。如為 Amazon EBS 磁碟區，預設為指派 D: 到 Z: 的磁碟機代號。對於執行個體存放磁碟區，預設值取決於驅動程式。 AWS PV 驅動程

式和 Citrix PV 驅動程式會指派從 Z： 到 A： 的執行個體存放磁碟區磁碟機代號。Red Hat 驅動程式會為執行個體儲存體磁碟區指派 D: 到 Z: 的磁碟機代號。

若要選擇磁碟區的磁碟機代號，請選擇映射。在 DriveLetterSetting 對話方塊中，指定每個磁碟區的磁碟區名稱和磁碟機代號值，選擇套用，然後選擇確定。建議選取不會與可能正在使用中的磁碟機代號衝突的代號，例如選取較中間的英文字母做為磁碟機代號。



指定磁碟機代號映射，並連接含有與您指定的其中一個磁碟區名稱相同標籤的磁碟區後，EC2Config 便會自動指派指定的磁碟機代號給該磁碟區。但假如磁碟機代號已在使用中，磁碟機代號映射便會失敗。請注意，EC2Config 不會變更在您指定磁碟機代號映射時已連接之磁碟區的磁碟機代號。

5. 若要儲存設定並在稍後繼續處理，請選擇確定以關閉 EC2 Service Properties 系統對話方塊。如果已完成執行個體的自訂且想從該執行個體建立 AMI，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。

疑難排解使用 EC2Config 啟動代理程式的問題

以下資訊有助於排解 EC2Config 服務的疑難問題。

在無法使用的執行個體上更新 EC2Config

請用下列程序更新使用遠端桌面無法存取之 Windows Server 執行個體上的 EC2Config 服務。

更新無法連線之 Amazon EBS 後端 Windows 執行個體上的 EC2Config

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 尋找受影響的執行個體。選取執行個體，並依序選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

4. 選擇 Launch instances (啟動執行個體)，並在與受影響執行個體相同的可用區域中，建立暫時性的 `t2.micro` 執行個體。使用與用來啟動受影響之執行個體不同的 AMI。

> **Important**
>
> 若您沒有在與受影響的執行個體相同的可用區域內建立執行個體，您將無法將受影響執行個體的根磁碟區連接到新的執行個體。

5. 在 EC2 主控台中，選擇 Volumes (磁碟區)。
6. 尋找受影響執行個體的根磁碟區。分開磁碟區，再將磁碟區連接至先前建立的暫時執行個體。使用預設裝置名稱 (xvdf) 連接它。
7. 使用遠端桌面連接至暫時性執行個體，然後使用 Disk Management 公用程式將磁碟區變成可用。
8. 下載最新版的 EC2Config 服務。將 `.zip` 檔案中的檔案解壓縮到所連接之磁碟機上的 `Temp` 目錄中。
9. 在暫時性執行個體上，開啟 Run (執行) 對話方塊，輸入 **`regedit`**，然後按 Enter 鍵。
10. 選擇 `HKEY_LOCAL_MACHINE`。從 File (檔案) 功能表中，選擇 Load Hive (載入 Hive)。選擇磁碟機，然後導覽並開啟下列檔案：`Windows\System32\config\SOFTWARE`。出現提示時，指定機碼名稱。
11. 選取剛載入的機碼並導覽至 `Microsoft\Windows\CurrentVersion`。選擇 `RunOnce` 機碼。如果該機碼不存在，從內容 (按一下右鍵) 選單中選擇 `CurrentVersion`，選擇 New (新增)，然後選擇 Key (機碼)。將機碼命名為 `RunOnce`。
12. 從內容 (按一下右鍵) 選單中選擇 `RunOnce` 機碼選擇 New (新增)，然後選擇 String Value (字串值)。在名稱輸入 `Ec2Install`，在資料輸入 `C:\Temp\Ec2Install.exe /quiet`。
13. 選擇 `HKEY_LOCAL_MACHINE\`*`specified key name`*`\Microsoft\Windows NT\CurrentVersion\Winlogon` 機碼。從內容 (按一下右鍵) 選單中選擇 New (新增)，然後選擇 String Value (字串值)。在名稱輸入 **`AutoAdminLogon`**，在數值資料輸入 **`1`**。
14. 選擇 `HKEY_LOCAL_MACHINE\`*`specified key name`*`\Microsoft\Windows NT\CurrentVersion\Winlogon>` 機碼。從內容 (按一下右鍵) 選單中選擇 New (新增)，然後選擇 String Value (字串值)。在名稱輸入 **`DefaultUserName`**，在數值資料輸入 **`Administrator`**。

15. 選擇 HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT\CurrentVersion\Winlogon 機碼。從內容 (按一下右鍵) 選單中選擇 New (新增)，然後選擇 String Value (字串值)。在名稱輸入 **DefaultPassword**，在數值資料輸入密碼。
16. 在 Registry Editor (登錄編輯程式) 導覽窗格中，選擇一開始開啟登錄編輯程式時建立的暫時性機碼。
17. 從 File (檔案) 選單中，選擇 Unload Hive (解除載入 Hive)。
18. 在 Disk Management 公用程式中，選擇先前連接的磁碟機，開啟內容 (按一下滑鼠右鍵) 選單，然後選擇 Offline (離線)。
19. 在 Amazon EC2 主控台中，將受影響的磁碟區與暫時性執行個體分離，然後將其重新連接至裝置名稱為 /dev/sda1 的執行個體。您必須指定此裝置名稱，將磁碟區指定為根磁碟區。
20. 停止和啟動 Amazon EC2 執行個體執行個體。
21. 執行個體啟動後，檢查系統日誌並確認出現 Windows is ready to use (Windows 已可開始使用) 的訊息。
22. 開啟登錄編輯程式，然後選擇 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon。刪除先前建立的字串索引鍵值：AutoAdminLogon、DefaultUserName 和 DefaultPassword。
23. 刪除或停止您在此程序中建立的暫時性執行個體。

EC2Config 版本記錄

下表說明 EC2Config 的發佈版本。如需 SSM Agent更新的詳細資訊，請參閱 Systems Manager SSM Agent 版本備註。

> **Important**
>
> EC2Config 已結束支援。僅最新版 EC2Config 代理程式可供下載。舊版將標示為私有。

版本	詳細資訊	發行日期
4.9.5777	• 修正某些執行個體類型的 RSS 組態設定不正確的問題。 • SSM Agent 新版本 3.3.484.0 。	2024 年 6 月 17 日

版本	詳細資訊	發行日期
4.9.5554	• 根據登錄項目限制網域名稱轉移：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel 。 • SSM Agent 新版本 3.2.1630.0 。	2023 年 10 月 4 日
4.9.5467	• 已新增重試功能，可探索主控台連接埠。 • SSM Agent 新版本 3.1.2282.0 。	2023 年 8 月 1 日
4.9.5288	• 已將 AWS Core SDK 更新為 3.7.103.23 版。 • 修正 AWS-UpdateEC2Config SSM 文件無法在僅啟用 IMDSv2 的執行個體上更新 EC2Config 的問題。 • SSM Agent 新版本 3.1.2144.0 。	2023 年 3 月 8 日
4.9.5231	• SSM Agent 新版本 3.1.1927.0。	2023 年 2 月 14 日
4.9.5103	• 已修正 r5d 和 i4i 執行個體系列上無法正確識別暫時性磁碟區的問題。 • SSM Agent 新版本 3.1.1856.0。	2022 年 12 月 5 日
4.9.5064	• 已更新為使用 PCI 區段資訊來選取主控台連接埠。 • 已簽署 PowerShell 指令碼，並已新增著作權標題。 • 已修正主要網路轉接器選取邏輯。 • SSM Agent 新版本 3.1.1732.0。	2022 年 11 月 16 日

版本	詳細資訊	發行日期
4.9.4588	• 更新了 IMDS 等待邏輯，以僅發出 IMDSv2 請求。 • 新增 libec2launch.dll 啟動代理程式共用庫。 • SSM Agent 3.1.1188.0 新版本。	2022 年 5 月 31 日
4.9.4556	• 新增等待邏輯以確保 NIC 在使用前完全初始化。 • 新版本的 Log4Net 2.0.14.0 收取了安全修補程式。 • 新版本的 SSM Agent 3.1.1045.0 收取了安全修補程式。	2022 年 3 月 1 日
4.9.4536	• 修復了 Temp 資料夾遺失時使用者資料損毀的問題。 • SSM Agent 新版本 3.1.804.0。	2022 年 1 月 31 日
4.9.4508	• 修正正確運算 diskpart 指令碼路徑的問題。 • SSM Agent 新版本 3.1.338.0。	2021 年 10 月 6 日
4.9.4500	• 已更新具有 IMDS v2 支援的 `Install-EgpuManagerConfig` 。 • 已更新 Web 連結以使用 https。 • SSM Agent 新版本 3.1.282.0	2021 年 9 月 7 日
4.9.4419	• 修正 IMDS 第 1 版後援邏輯 • 已將 Windows 臨時目錄的所有使用更新為 EC2config 臨時目錄 • SSM Agent 新版本 3.0.1124.0	2021 年 6 月 2 日

版本	詳細資訊	發行日期
4.9.4381	• 在 EC2ConfigUpdater 中新增了對 SSM 文件結構描述 2.2 版的支援 • 新增 AWS Nitro Enclaves 套件版本至主控台日誌 • SSM Agent 新版本 3.0.529.0	2021 年 5 月 4 日
4.9.4326	• 移除了設定 UI 中的所有連結 • 這是支援 Windows Server 2008 的最後一個 EC2Config 版本。	2021 年 3 月 3 日
4.9.4279	• 已修正與 `Ec2ConfigMonitor` 排程任務相關的安全性問題 • 修正磁碟機代號對應問題和不正確的暫時磁碟計數 • 新增 `OsCurrentBuild` 和 `OsReleaseId` 至主控台輸出 • SSM Agent 新版本 2.3.871.0	2020 年 12 月 11 日
4.9.4222	• 修正 IMDS 第 1 版後援邏輯 • SSM Agent 新版本 2.3.842.0	2020 年 4 月 7 日
4.9.4122	• 已新增對 IMDS v2 的支援。 • SSM Agent 新版本 2.3.814.0	2020 年 3 月 4 日
4.9.3865	• 修正 Windows Server 2008 R2 在裸機執行個體上偵測 COM 連接埠的問題 • SSM Agent 新版本 2.3.722.0	2019 年 10 月 31 日
4.9.3519	• SSM Agent 新版本 2.3.634.0	2019 年 6 月 18 日
4.9.3429	• SSM Agent 新版本 2.3.542.0	2019 年 4 月 25 日
4.9.3289	• SSM Agent 新版本 2.3.444.0	2019 年 2 月 11 日

版本	詳細資訊	發行日期
4.9.3270	• 新增了將顯示器設定為永遠不關閉的外掛程式，以修復 ACPI 問題。 • 寫入主控制台的 SQL Server 版本。 • SSM Agent 新版本 2.3.415.0	2019 年 1 月 22 日
4.9.3230	• 已更新磁碟機代號對應說明，以更切合功能。 • SSM Agent 新版本 2.3.372.0	2019 年 1 月 10 日
4.9.3160	• 主要 NIC 的等待時間增加 • RSS 的新增預設組態和 ENA 裝置的接收佇列設定 • Sysprep 時停用休眠 • SSM Agent 新版本 2.3.344.0 • 升級 AWS SDK 至 3.3.29.13	2018 年 12 月 15 日
4.9.3067	• 執行個體休眠改進 • SSM Agent 新版本 2.3.235.0	2018 年 11 月 8 日
4.9.3034	• DNS 伺服器新增路由 169.254.169.253/32 • SSM Agent 新版本 2.3.193.0	2018 年 10 月 24 日
4.9.2986	• 所有 EC2Config 相關二進位檔的新增簽署 • SSM Agent 新版本 2.3.136.0	2018 年 10 月 11 日
4.9.2953	SSM Agent 新版本 (2.3.117.0)	2018 年 10 月 2 日
4.9.2926	SSM Agent 新版本 (2.3.68.0)	2018 年 9 月 18 日
4.9.2905	• SSM Agent 新版本 (2.3.50.0) • 已新增路由 169.254.169.123/32 至 AMZN 時間服務 • 已新增路由 169.254.169.249/32 至 GRID 授權服務 • 已修正造成 EBS NVMe 磁碟區被標示為暫時性的問題	2018 年 9 月 17 日

版本	詳細資訊	發行日期
4.9.2854	SSM Agent 新版本 (2.3.13.0)	2018 年 8 月 17 日
4.9.2831	SSM Agent 新版本 (2.2.916.0)	2018 年 8 月 7 日
4.9.2818	SSM Agent 新版本 (2.2.902.0)	2018 年 7 月 31 日
4.9.2756	SSM Agent 新版本 (2.2.800.0)	2018 年 6 月 27 日
4.9.2688	SSM Agent 新版本 (2.2.607.0)	2018 年 5 月 25 日
4.9.2660	SSM Agent 新版本 (2.2.546.0)	2018 年 5 月 11 日
4.9.2644	SSM Agent 新版本 (2.2.493.0)	2018 年 4 月 26 日
4.9.2586	SSM Agent 新版本 (2.2.392.0)	2018 年 3 月 28 日
4.9.2565	• SSM Agent 新版本 (2.2.355.0) • 修正 M5 和 C5 執行個體的問題 (找不到 PV 驅動程式) • 新增執行個體類型、最新 PV 驅動程式和 NVMe 驅動程式的主控台記錄	2018 年 3 月 13 日
4.9.2549	SSM Agent 新版本 (2.2.325.0)	2018 年 3 月 8 日
4.9.2461	SSM Agent 新版本 (2.2.257.0)	2018 年 2 月 15 日

版本	詳細資訊	發行日期
4.9.2439	SSM Agent 新版本 (2.2.191.0)	2018 年 2 月 6 日
4.9.2400	SSM Agent 新版本 (2.2.160.0)	2018 年 1 月 16 日
4.9.2327	• SSM Agent 新版本 (2.2.120.0) • 新增 Amazon EC2 裸機執行個體上的 COM 連接埠探索 • 新增 Amazon EC2 裸機執行個體上的 Hyper-V 狀態記錄	2018 年 1 月 2 日
4.9.2294	SSM Agent 新版本 (2.2.103.0)	2017 年 12 月 4 日
4.9.2262	SSM Agent 新版本 (2.2.93.0)	2017 年 11 月 15 日
4.9.2246	SSM Agent 新版本 (2.2.82.0)	2017 年 11 月 11 日
4.9.2218	SSM Agent 新版本 (2.2.64.0)	2017 年 10 月 29 日
4.9.2212	SSM Agent 新版本 (2.2.58.0)	2017 年 10 月 23 日
4.9.2203	SSM Agent 新版本 (2.2.45.0)	2017 年 10 月 19 日
4.9.2188	SSM Agent 新版本 (2.2.30.0)	2017 年 10 月 10 日
4.9.2180	• SSM Agent 新版本 (2.2.24.0) • 新增 GPU 執行個體的 Elastic GPU 外掛程式	2017 年 10 月 5 日
4.9.2143	SSM Agent 新版本 (2.2.16.0)	2017 年 10 月 1 日

版本	詳細資訊	發行日期
4.9.2140	SSM Agent 新版本 (2.1.10.0)	
4.9.2130	SSM Agent 新版本 (2.1.4.0)	
4.9.2106	SSM Agent 新版本 (2.0.952.0)	
4.9.2061	SSM Agent 新版本 (2.0.922.0)	
4.9.2047	SSM Agent 新版本 (2.0.913.0)	
4.9.2031	SSM Agent 新版本 (2.0.902.0)	
4.9.2016	• SSM Agent 新版本 (2.0.879.0) • 修正 Windows Server 2003 的 CloudWatch Logs 目錄路徑	
4.9.1981	• SSM Agent 新版本 (2.0.847.0) • 修正 EBS 磁碟區內所產生之 `important.txt` 的問題。	
4.9.1964	SSM Agent 新版本 (2.0.842.0)	
4.9.1951	• SSM Agent 新版本 (2.0.834.0) • 修正暫時性硬碟機之磁碟機代號 Z: 無法對應的問題。	
4.9.1925	• SSM Agent 新版本 (2.0.822.0) • (錯誤) 此版本不是 SSM Agent v4.9.1775 的有效更新目標。	
4.9.1900	SSM Agent 新版本 (2.0.805.0)	
4.9.1876	• SSM Agent 新版本 (2.0.796.0) • 修正管理使用者資料執行的輸出/錯誤重新導向問題。	

版本	詳細資訊	發行日期
4.9.1863	• SSM Agent 新版本 (2.0.790.0) • 修正連接多個 EBS 磁碟區至 Amazon EC2 執行個體時發生的問題。 • 改善 CloudWatch 為接受設定路徑，保持向下相容性。	
4.9.1791	SSM Agent 新版本 (2.0.767.0)	
4.9.1775	SSM Agent 新版本 (2.0.761.0)	
4.9.1752	SSM Agent 新版本 (2.0.755.0)	
4.9.1711	SSM Agent 新版本 (2.0.730.0)	
4.8.1676	SSM Agent 新版本 (2.0.716.0)	
4.7.1631	SSM Agent 新版本 (2.0.682.0)	
4.6.1579	• SSM Agent 新版本 (2.0.672.0) • 修正 v4.3、v4.4 和 v4.5 的代理程式更新問題	
4.5.1534	SSM Agent 新版本 (2.0.645.1)	
4.4.1503	SSM Agent 新版本 (2.0.633.0)	
4.3.1472	SSM Agent 新版本 (2.0.617.1)	
4.2.1442	SSM Agent 新版本 (2.0.599.0)	
4.1.1378	SSM Agent 新版本 (2.0.558.0)	

版本	詳細資訊	發行日期
4.0.1343	• 執行命令、狀態管理員、CloudWatch 代理程式和網域加入支援已移至另一稱為 SSM Agent 的代理程式。SSM Agent 將在 EC2Config 升級時一併安裝。如需詳細資訊，請參閱 EC2Config 和 AWS Systems Manager。 • 如果您已在 EC2Config 中設定代理伺服器，則您需要先更新 SSM Agent 的代理伺服器，然後再升級。如果未更新代理伺服器設定，將無法使用執行命令管理執行個體。為避免此問題，請在更新至新版本前先參閱下列資訊：「AWS Systems Manager 使用者指南」中的在 Windows 執行個體上安裝和設定 SSM Agent。 • 如果先前已使用本機設定檔 (`AWS.EC2.Windows.CloudWatch.json`) 在執行個體上啟用 CloudWatch 整合，您需要設定該檔案，以結合 SSM Agent 使用。	
3.19.1153	• 為具有舊 AWS KMS 組態的執行個體重新啟用外掛程式。跳過 BYOL 使用者的啟動。 • 將磁碟格式化操作期間的預設 TRIM 行為變更為停用，並新增 FormatWithTRIM，以使用者資料覆寫 InitializeDisks 外掛程式。	
3.18.1118	• 修正以可靠地新增路由表至主要網路轉接器。 • 更新以改善 AWS 服務的支援。	
3.17.1032	• 修正篩選條件設為相同類別時出現的重複系統日誌。 • 修正以避免磁碟初始化時發生停滯。	
3.16.930	新增在啟動時將「Window is Ready to use」(Window 已可開始使用) 事件記錄至 Windows 事件日誌的支援。	

版本	詳細資訊	發行日期
3.15.880	修正以允許將 Systems Manager 執行命令輸出上傳至名稱含「.」字元的 S3 儲存貯體。	
3.14.786	新增覆寫 InitializeDisks 外掛程式設定的支援。例如：若要加速 SSD 磁碟初始化，您可在使用者資料中指定下列設定以暫時停用 TRIM： <InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings	
3.13.727	Systems Manager RunCommand - 修正以在 Windows 重新開機後可靠地處理命令。	
3.12.649	• 修正以在執行命令/指令碼時從容處理重新開機。 • 修正以從容取消執行命令。 • 透過 Systems Manager Run Command 安裝應用程式時，新增支援（選用）將 MSI 日誌上傳至 S3。	
3.11.521	• 修正以啟用 Windows Server 2003 的 RDP 指紋產生。 • 修正以將時區和 UTC 位移包含在 EC2Config 日誌行內。 • Systems Manager 支援平行執行執行命令的命令。 • 還原先前的變更，將已分割的磁碟設為上線。	
3.10.442	• 修正安裝 MSI 應用程式時的 Systems Manager 設定錯誤。 • 修正以可靠地將儲存磁碟設為上線。 • 更新以改善 AWS 服務的支援。	

版本	詳細資訊	發行日期
3.9.359	• 修正後期 Sysprep 指令碼，讓 Windows 更新設定保持為預設狀態。 • 修正密碼產生外掛程式，以改善取得 GPO 密碼政策設定的可靠性。 • 限制僅本機管理員群組擁有 EC2Config/SSM 日誌資料夾權限。 • 更新以改善 AWS 服務的支援。	
3.8.294	• 修正 CloudWatch 非主要磁碟機上的日誌無法上傳的問題。 • 加入重試邏輯，改善磁碟初始化流程。 • 新增改善建立 AMI 時 SetPassword 外掛程式有時會發生錯誤的錯誤處理能力。 • 更新以改善 AWS 服務的支援。	
3.7.308	• 改善 ec2config-cli 公用程式，以在執行個體內進行 config 測試與故障診斷。 • 避免在 OpenVPN 轉接器上新增 AWS KMS 和中繼資料服務的靜態路由。 • 修正使用者資料執行未執行「persist」標籤的問題。 • 改善無法登入 EC2 主控台時的錯誤處理能力。 • 更新以改善 AWS 服務的支援。	

版本	詳細資訊	發行日期
3.6.269	• Windows 啟用可靠性修正為先使用連結本機地址 169.254.0.250/251 透過 啟用 Windows AWS KMS • 改善 Systems Manager、Windows 啟動和網域加入等狀況下的代理伺服器處理能力 • 修正重複的使用者帳戶行新增到 Sysprep 回答檔案的問題	
3.5.228	• 解決 CloudWatch 外掛程式在讀取 Windows 事件日誌時可能消耗過多 CPU 和記憶體的狀況 • 在 EC2Config 設定 UI 中，新增 CloudWatch 組態文件的連結	
3.4.212	• 修正 EC2Config 搭配 VM-Import 使用時的問題。 • 修正 WiX 安裝程式內的服務命名問題。	
3.3.174	• 改善 Systems Manager 和網域加入問題的例外處理能力。 • 支援 Systems Manager SSM 結構描述版本控制的變更。 • 修正 Win2K3 上的暫時性磁碟格式化問題。 • 支援設定超過 2TB 磁碟大小的變更。 • 設定 GC 模式為預設值，減少虛擬記憶體用量。 • 支援從 `aws:psModule` 和 `aws:application` 外掛程式的 UNC 路徑下載成品。 • 改善 Windows 啟用外掛程式的記錄功能。	

版本	詳細資訊	發行日期
3.2.97	• 延遲載入 Systems Manager SSM 組件，以改善效能。 • 改善不正確之 sysprep2008.xml 的例外處理能力。 • Systems Manager「套用」設定的命令列支援。 • 支援有電腦重新命名等待時加入網域的變更。 • 支援 `aws:applications` 外掛程式中的選擇性參數。 • 支援 `aws:psModule` 外掛程式中的命令陣列。	
3.0.54	• 啟用 Systems Manager 的支援。 • 透過 Systems Manager 自動將 EC2 Windows 執行個體加入 AWS 目錄的網域。 • 透過 Systems Manager 設定和上傳 CloudWatch 日誌/指標。 • 透過 Systems Manager 安裝 PowerShell 模組。 • 透過 Systems Manager 安裝 MSI 應用程式。	
2.4.233	• 新增排程任務，從服務啟動故障中復原 EC2Config。 • 改善主控台日誌錯誤訊息。 • 更新以改善 AWS 服務的支援。	

版本	詳細資訊	發行日期
2.3.313	• 修正啟用 CloudWatch Logs 功能時某些情況下會消耗大量記憶體的問題。 • 修正升級錯誤，允許 ec2config 2.1.19 以下版本現在可升級至最新版。 • 更新 COM 連接埠開啟例外，使其在日誌中更方便好用。 • Ec2configServiceSettings UI 停用調整規模，並修正 UI 中的屬性和版本顯示位置。	
2.2.12	• 處理為判斷 Windows Sysprep 狀態而查詢登錄機碼有時傳回 null 的 NullPointerException 問題。 • 發佈 finally 區塊中的非受管資源。	
2.2.11	修正 CloudWatch 外掛程式處理空白日誌行的問題。	
2.2.10	• 移除透過 UI 設定 CloudWatch Logs 設定。 • 讓使用者在 `%ProgramFiles%\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` 檔案中定義 CloudWatch Logs 設定，以允許未來的增強功能。	
2.2.9	修正未處理的例外狀況並新增記錄功能。	
2.2.8	• 修正 EC2Config 安裝程式中的 Windows OS 版本檢查，以支援 Windows Server 2003 SP1 和更新版本。 • 修正為更新 Sysprep 組態檔而讀取登錄機碼時的 null 值處理能力。	

版本	詳細資訊	發行日期
2.2.7	• 新增 Windows 2008 和更新版本上 EC2Config 在 Sysprep 執行期間執行的支援。 • 改善例外處理和記錄功能，以提升診斷效率	
2.2.6	• 減少上傳日誌事件時對執行個體和 CloudWatch Logs 的負載。 • 解決 CloudWatch Logs 外掛程式無法持續保持啟用的升級問題	
2.2.5	• 新增上傳日誌到 CloudWatch Logs 服務的支援。 • 修正 Ec2OutputRDPCert 外掛程式中的競爭條件問題 • 變更 EC2Config 服務復原選項，從 TakeNoAction 變為 Restart • 新增更多 EC2Config 當機時的例外狀況資訊	
2.2.4	• 修正 PostSysprep.cmd 中的錯別字 • 修正 EC2Config 在 OS2012+ 中無法釘選至開始選單的錯誤	

版本	詳細資訊	發行日期
2.2.3	• 新增在安裝時未立即啟動服務之情況下安裝 EC2Config 的選項。若要使用，請從命令提示執行「Ec2Install.exe start=false」 • 新增桌布外掛程式中用於控制新增/移除桌布的參數。若要使用，請從命令提示執行「Ec2WallpaperInfo.exe set」或「Ec2WallpaperInfo.exe revert」 • 新增檢查 RealTimeIsUniversal 機碼，輸出不正確的 RealTimeIsUniveral 登錄機碼設定至主控台 • 移除 EC2Config 對 Windows 暫存資料夾的相依性 • 移除 UserData 執行對 .Net 3.5 的相依性	
2.2.2	• 新增對服務停止行為的檢查，以檢查資源是否發佈 • 修正加入網域時執行時間過長的問題	
2.2.1	• 更新安裝程式，以允許從舊版升級 • 修正僅發生於 .Net4.5 環境下的 Ec2WallpaperInfo 錯誤 • 修正間歇性的驅動程式偵測錯誤 • 新增靜音安裝選項。執行 Ec2Install.exe 並加入「-q」選項。例如：「Ec2Install.exe -q」	
2.2.0	• 新增對僅限於 .Net4 和 .Net4.5 環境的支援 • 更新安裝程式	

版本	詳細資訊	發行日期
2.1.19	• 新增使用 Intel 網路驅動程式時的暫時性磁碟標籤支援 (例如 C3 執行個體類型)。如需詳細資訊，請參閱增強 Amazon EC2 執行個體的聯網能力。 • 新增 AMI Origin Version 和 AMI Origin Name 支援至主控台輸出 • 變更主控台輸出，以執行一致的格式化/剖析 • 更新說明檔案	
2.1.18	• 新增 EC2Config WMI Object for Completion 通知 (-Namespace root\Amazon -Class EC2_ConfigService) • 改善含大量事件日誌之啟動 WMI 查詢的效能；避免其在初始執行時長時間大量消耗 CPU	
2.1.17	• 修正標準輸出和標準錯誤緩衝填入的 UserData 執行問題 • 修正 w2k8 OS 以上版本主控台輸出有時出現不正確 RDP 指紋的問題 • Windows 2008+ 的主控台輸出現在包含「RDPCERTIFICATE-SubjectName:」，其包含機器名稱值 • 新增 D:\ 至磁碟機代號映射下拉式清單 • 移動說明按鈕至右上角並變更外觀/使用體驗 • 新增意見回饋問卷連結至右上角	

版本	詳細資訊	發行日期
2.1.16	• 一般索引標籤包含新版本的 EC2Config 下載頁面連結 • 桌面桌布疊加層現在存放在 Users Local Appdata 資料夾，而非 My Documents，以支援 MyDoc 重新導向 • MSSQLServer 名稱與 Post-Sysprep 指令碼中的系統同步 (2008+) • 重新排序應用程式資料夾 (將檔案移至 Plugin 目錄並移除重複的檔案) • 變更系統日誌輸出 (主控台)： • *改為日期、名稱、值格式，以方便剖析 (請開始將相依性遷移至新格式) • *新增「Ec2SetPassword」外掛程式狀態 • *新增 Sysprep 開始和結束時間 • 修正非英文版作業系統上暫時性磁碟未標籤為「Temporary Storage」的問題 • 修正執行 Sysprep 後的 EC2Config 解除安裝問題	

版本	詳細資訊	發行日期
2.1.15	• 將中繼資料服務請求最佳化 • 中繼資料現在將繞過代理伺服器設定 • 暫時性磁碟已標籤為「Temporary Storage」，且找到的 Important.txt 將放入磁碟區 (僅限 Citrix PV 驅動程式)。如需詳細資訊，請參閱升級 EC2 Windows 執行個體上的 PV 驅動程式。 • 暫時性磁碟將指派 Z 到 A 的代號 (僅限 Citrix PV 驅動程式) - 指派可用磁碟機代號映射外掛程式覆寫，覆寫為磁碟區標籤「Temporary Storage X」(其中 X 為 0-25 的數字) • UserData 現在會在「Windows 就緒」後立即執行	
2.1.14	桌面桌布修正	
2.1.13	• 桌面桌布預設將顯示主機名稱 • 移除對 Windows 時間服務的相依性 • 針對單一介面獲指派多重 IP 的狀況新增路由表	
2.1.11	• 變更 Ec2Activation 外掛程式 • -每隔 30 天檢查一次啟用狀態 • -如果寬限期仍有 90 天 (共 180 天)，則重新嘗試啟用	

版本	詳細資訊	發行日期
2.1.10	• 桌面桌布疊加層不再與 Sysprep 或不使用 Sysprep 關機一致 • 在開頭為 <persist>true</persist> 的每個服務上執行的使用者資料選項 • 變更 /DisableWinUpdate.cmd 的位置和名稱為 /Scripts/PostSysprep.cmd • /Scripts/PostSysprep.cmd 中的管理員密碼預設為不過期 • 解除安裝會將 EC2Config PostSysprep 指令碼從 c:\windows\setup\script\CommandComplete.cmd 中移除 • 新增路由表對自訂介面指標的支援	
2.1.9	UserData 執行不再侷限為 3851 個字元	

版本	詳細資訊	發行日期
2.1.7	• OS 版本和語言識別符寫入至主控台 • EC2Config 版本寫入至主控台 • PV 驅動程式版本寫入至主控台 • 偵測錯誤檢查，並在找到錯誤後於下次開機時輸出至主控台 • 新增選項至 config.xml，以維持 Sysprep 登入資料 • 新增啟動時無法使用 ENI 情況下的路由重試邏輯 • 使用者資料執行 PID 寫入至主控台 • 從 GPO 擷取的最小產生密碼長度 • 設定服務啟動重試 3 次 • 新增 S3_DownloadFile.ps1 和 S3_Upload file.ps1 範例至 / Scripts 資料夾	

版本	詳細資訊	發行日期
2.1.6	• 新增版本資訊至 General (一般) 索引標籤 • Bundle 索引標籤重新命名為 Image (影像) • 簡化指定密碼的流程，並將密碼相關 UI 從 General (一般) 索引標籤移至 Image (影像) 索引標籤 • Disk Settings (磁碟設定) 索引標籤重新命名為 Storage (儲存) • 新增 Support (支援) 索引標籤，並提供可執行故障排除的常用工具 • Windows Server 2003 `sysprep.ini` 設為預設擴充 OS 分割區 • 新增私有 IP 地址至桌布 • 私有 IP 地址將在桌布上顯示 • 新增主控台輸出的重試邏輯 • 修正中繼資料存取功能的通訊連接埠例外狀況 -- 其會導致 EC2Config 在顯示主控台輸出之前終止 • 每次開機時檢查啟用狀態 -- 視需要啟動 • 修正相關路徑的問題 -- 發生在手動從啟動資料夾執行桌布捷徑時；指向管理員/日誌 • 修正 Windows Server 2003 使用者 (管理員除外) 的預設背景色彩	

版本	詳細資訊	發行日期
2.1.2	• UTC (Zulu) 的主控台時間戳記 • 移除 Sysprep 索引標籤上的超連結顯示 • 新增在 Windows 2008+ 第一次開機時動態擴充根磁碟區的功能 • 啟用設定密碼時，現在允取 EC2Config 設定密碼 • EC2Config 會在執行 Sysprep 之前檢查啟用狀態 (若未啟動將出現警告) • Windows Server 2003 `Sysprep.xml` 現在預設為 UTC 時區，而非亞太地區 • 隨機啟動伺服器 • Drive Mapping (磁碟機映射) 索引標籤重新命名為 Disk Settings (磁碟設定) • Initialize Drives UI (初始化磁碟機 UI) 項目從 General (一般) 索引標籤移至 Disk Settings (磁碟設定) 索引標籤 • 說明按鈕現在指向 HTML 說明檔案 • 更新 HTML 說明檔案的變更 • 更新磁碟機代號映射的「備註」文字 • 新增 InstallUpdates.ps1 至 /Scripts 資料夾，在 Sysprep 之前自動執行修補程式和清理	

版本	詳細資訊	發行日期
2.1.0	• 桌面桌布預設在第一次登入 (未中斷/重新連接) 後顯示執行個體資訊 • PowerShell 可從使用者資料執行，只要用 <powershell></powershell> 括住程式碼	

為您的 Windows 執行個體使用 EC2 Fast Launch

當您設定適用於 EC2 Fast Launch 的 Windows Server AMI 時，Amazon EC2 會建立一組預先佈建的快照，以便加快啟動速度，如下所示。

1. Amazon EC2 會依據您的設定啟動一組暫時 t3 執行個體。
2. 當每個臨時執行個體完成標準啟動步驟時，Amazon EC2 會建立預先佈建的執行個體快照。它會將快照存放在您的 Amazon S3 儲存貯體中。
3. 快照準備就緒時，Amazon EC2 會終止相關聯的 t3 執行個體，以盡可能降低資源成本。
4. 下一次 Amazon EC2 從啟用 EC2 Fast Launch 的 AMI 中啟動執行個體時，會使用其中一個快照來大幅度縮短啟動時間。

Amazon EC2 會自動補充您現有的快照，因為它會使用這些快照從啟用 EC2 Fast Launch 的 AMI 啟動執行個體。

任何有權存取 AMI 且啟用了 EC2 Fast Launch 的帳戶，皆可因啟動時間縮短而受益。當 AMI 擁有者授予您啟動執行個體的權限時，預先佈建的快照來自 AMI 擁有者的帳戶。

如果有人與您共用支援 EC2 Fast Launch 功能的 AMI，您可以自行在共用 AMI 上啟用或停用更快的啟動速度。若您為共用 AMI 啟用 EC2 Fast Launch 功能，Amazon EC2 會直接在您的帳戶中建立預先佈建快照。如果您耗盡帳戶中的快照，仍可使用 AMI 擁有者帳戶中的快照。

> **Note**
>
> 當某個啟動使用預先佈建的快照時，EC2 Fast Launch 會立即刪除這些快照，從而將儲存成本降至最低並防止重複使用。不過，如果刪除的快照符合保留規則，資源回收筒會自動保留它們。建議您檢閱資源回收筒保留規則的範圍，以免發生此類情況。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的資源回收筒。

此功能與 EBS 快速快照還原不同。您必須針對每個快照確實啟用 EBS 快速快照還原，並有其相關費用。

以下影片示範如何設定您的 Windows AMI 以快速概觀相關關鍵術語及其定義來加快啟動速度：啟動 EC2 Windows 執行個體的速度最多加快 65% AWS。

資源成本

為 Windows AMI 設定 EC2 Fast Launch 無需支付服務費。不過，標準定價適用於 Amazon EC2 使用的任何基礎 AWS 資源。若要進一步了解相關資源成本以及如何管理這些成本，請參閱 管理 EC2 Fast Launch 基礎資源的成本。

目錄

- 重要用語
- 適用於 Windows 的 EC2 Fast Launch 的先決條件
- 為您的 Amazon EC2 Windows Server AMI 設定 EC2 Fast Launch 設定
- 檢視啟用了 EC2 Fast Launch 的 AMI
- 管理 EC2 Fast Launch 基礎資源的成本
- 監控 EC2 Fast Launch
- 用於 EC2 Fast Launch 的服務連結角色
- 對 Windows EC2 Fast Launch 進行故障診斷

重要用語

EC2 Fast Launch 功能使用下列重要用語：

已預先佈建的快照

執行個體的快照，此執行個體是從啟用了 EC2 Fast Launch 功能的 Windows AMI 啟動，且已完成以下 Windows 啟動步驟並根據需要重新啟動。

- Sysprep 特製化
- Windows 開箱即用體驗 (OOBE)

這些步驟完成後，EC2 Fast Launch 將停止執行個體，並建立快照，以便稍後用於依據組態從 AMI 更快速啟動。

啟動頻率

控制 Amazon EC2 可在指定時間範圍內啟動的預先佈建快照的數量。為 AMI 啟用 EC2 Fast Launch 時，Amazon EC2 會在背景建立一組初始的預先佈建快照。例如，如果啟動頻率設定為每小時五次啟動 (預設值)，則 EC2 Fast Launch 將建立一組由五個預先佈建快照組成的初始快照。

當 Amazon EC2 從啟用了 EC2 Fast Launch 的 AMI 中啟動執行個體時，會使用其中一個預先佈建快照來縮短啟動時間。使用快照時，它們會自動補充，直到達到啟動頻率指定的數量。

如果您預計從 AMI 啟動的執行個體數量會出現峰值 (例如，在特殊事件期間)，則可以預先增加啟動頻率，以涵蓋所需的其他執行個體。當您的啟動速率恢復正常時，則可以將頻率降低回原始值。

當您遇到啟動次數高於預期時，則可能會用到所有可用的預先佈建快照。這不會引發任何啟動失敗。但是，它可能會使某些執行個體經歷標準啟動過程，直到快照得到補充為止。

目標資源計數

為啟用 EC2 Fast Launch 功能的 Amazon EC2 Windows Server AMI 所保留的預先佈建快照數量。

最大平行啟動數

控制 Amazon EC2 可同時啟動多少個執行個體，以便為 EC2 Fast Launch 功能建立預先佈建的快照。如果您的目標資源計數高於設定的最大平行啟動數，Amazon EC2 會啟動開始建立快照之最大平行啟動數設定所指定的執行個體數。當這些執行個體完成此程序時，Amazon EC2 會擷取快照並停止執行個體。然後，它會繼續啟動更多執行個體，直到可用的快照總數達到目標資源計數為止。最大平行啟動數的值必須為 6 或更大。

適用於 Windows 的 EC2 Fast Launch 的先決條件

在設定 EC2 Fast Launch 之前，請確認您已符合下列先決條件，以便為您 AWS 帳戶中的 AMI 建立快照:

- 若您在設定 EC2 Fast Launch 時提供自訂啟動範本，此服務會使用您在啟動範本中定義的 VPC 和其他組態設定。如需詳細資訊，請參閱在設定 EC2 Fast Launch 時使用啟動範本。
- 若您未使用自訂啟動範本來設定該設定，在啟用 EC2 Fast Launch 前，必須將 EC2FastLaunchFullAccess 政策連接至您目前的 IAM 角色。然後，服務會自動在 中建立具有下列資源的 CloudFormation 堆疊 AWS 帳戶。
  - 虛擬私有雲端 (VPC)
  - 跨多個可用區域的私有子網路

  - 已設定執行個體中繼資料服務第 2 版 (IMDSv2) 的啟動範本
  - 無傳入或傳出規則的安全群組
- 私有 EC2 Fast Launch AMI 必須支援使用者資料指令碼執行。
- 若要為 AMI 設定 EC2 Fast Launch，必須使用帶關機選項的 Sysprep 來建立 AMI。EC2 Fast Launch 功能目前不支援執行中執行個體建立的 AMI。

  若要使用 Sysprep 建立 AMI，請參閱 使用 Windows Sysprep 建立 Amazon EC2 AMI。
- 若要為使用客戶自管金鑰加密的加密 AMI 啟用 EC2 Fast Launch，必須授予 EC2 Fast Launch 的服務連結角色使用該 CMK 的許可。如需詳細資訊，請參閱the section called “存取客戶自管金鑰”。
- AWS 帳戶 中所有 AMI 的最大平行啟動數預設配額為每個區域 40。您可為自己的帳戶請求增加 Service Quotas，如下所示。

  1. 開啟 Service Quotas 主控台，網址為 https://console.aws.amazon.com/servicequotas/。
  2. 在導覽窗格中，選擇 AWS 服務。
  3. 在搜尋列中輸入 **`EC2 Fast Launch`**，然後選取結果。
  4. 選取平行執行個體啟動連結，以開啟服務配額詳細資訊頁面。
  5. 選擇在帳戶層級請求增加。

  如需詳細資訊，請參閱「Service Quotas 使用者指南」中的請求提高配額。

為您的 Amazon EC2 Windows Server AMI 設定 EC2 Fast Launch 設定

您可以從 API、 AWS 管理主控台 SDK、CloudFormation 或 AWS Command Line Interface () 為您所擁有AMIs 或與您共用SDKs AMIs 設定 EC2 Fast Launch AWS CLI。設定 EC2 Fast Launch 之前，請確認 AMI 符合建立預先佈建快照所需的所有先決條件。如需詳細資訊，請參閱適用於 Windows 的 EC2 Fast Launch 的先決條件。

當您為 Windows 執行個體啟用更快速的啟動時，Amazon EC2 會進行檢查，以確保您擁有從指定的 AMI 和啟動範本 (如果提供) 啟動執行個體所需的許可，包括加密 AMI 的許可。為了防止執行個體啟動程序期間發生錯誤，服務會在啟用 EC2 Fast Launch 之前驗證您的許可。如果您沒有所需的許可，服務會傳回錯誤，且不會啟用 EC2 Fast Launch。

EC2 Fast Launch 與 EC2 Image Builder 整合，協助您建立啟用 EC2 Fast Launch 的自訂映像。如需詳細資訊，請參閱「EC2 Image Builder 使用者指南」中的為啟用 EC2 Fast Launch 的 Windows AMI (AWS CLI) 建立分佈設定。

啟用 EC2 Fast Launch

在變更這些設定之前，請確保您的 AMI 和在其中執行的區域滿足所有 適用於 Windows 的 EC2 Fast Launch 的先決條件。

Console

啟用 EC2 Fast Launch

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Images (映像) 下，選擇 AMIs (AIM)。
3. 選取要更新的 AMI，方法選取相應 Name (名稱) 旁邊的核取方塊。
4. 從 AMI 清單上方的動作功能表中選擇設定快速啟動。這將開啟設定快速啟動頁面，您可以在其中設定 EC2 Fast Launch 的設定。
5. 若要開始使用預先佈建的快照以更快地從 Windows AMI 啟動執行個體，請選取啟用 Fast Launch 核取方塊。
6. 從 Set anticipated launch frequency (設定預期啟動頻率) 下拉式清單中，選擇一個值以指定為涵蓋預期執行個體啟動磁碟區而建立和維護的快照數量。
7. 修改完成後，請選擇 Save changes (儲存變更)。

> **Note**
>
> 如果您需要用啟動範本來指定 VPC 或 IMDSv2 的中繼資料設定，請參閱 在設定 EC2 Fast Launch 時使用啟動範本。

AWS CLI

啟用 EC2 Fast Launch

使用下列 enable-fast-launch 命令為指定的 AMI 啟用 EC2 Fast Launch，啟動六個平行執行個體進行預先佈建。

```
aws ec2 enable-fast-launch \
    --image-id ami-0abcdef1234567890  \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

以下為範例輸出。

```
{
 "ImageId": "ami-0abcdef1234567890",
 "ResourceType": "snapshot",
 "SnapshotConfiguration": {
     "TargetResourceCount": 10
 },
 "LaunchTemplate": {},
 "MaxParallelLaunches": 6,
 "OwnerId": "0123456789123",
 "State": "enabling",
 "StateTransitionReason": "Client.UserInitiated",
 "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

PowerShell

啟用 EC2 Fast Launch

使用 Enable-EC2FastLaunch cmdlet 為指定的 AMI 啟用 EC2 Fast Launch，啟動六個平行執行個體進行預先佈建。

```
Enable-EC2FastLaunch `
 -ImageId ami-0abcdef1234567890 `
 -MaxParallelLaunch 6 `
 -Region us-west-2 `
 -ResourceType snapshot
```

以下為範例輸出。

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate        :
 MaxParallelLaunches   : 6
 OwnerId               : 0123456789123
 ResourceType          : snapshot
 SnapshotConfiguration : Amazon.EC2.Model.FastLaunchSnapshotConfigurationResponse
 State                 : enabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime   : 2/25/2022 12:24:11 PM
```

停用 EC2 Fast Launch

在變更這些設定之前，請確保您的 AMI 和在其中執行的區域滿足所有 適用於 Windows 的 EC2 Fast Launch 的先決條件。

Console

停用 EC2 Fast Launch

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Images (映像) 下，選擇 AMIs (AIM)。
3. 選取要更新的 AMI，方法選取相應 Name (名稱) 旁邊的核取方塊。
4. 從 AMI 清單上方的動作功能表中選擇設定快速啟動。這將開啟設定快速啟動頁面，您可以在其中設定 EC2 Fast Launch 的設定。
5. 清除啟用 EC2 Fast Launch 核取方塊以停用 Windows 快速啟動並移除預先佈建的快照。這會導致 AMI 將繼續為每個執行個體使用標準啟動流程。

> **Note**
>
> 停用 Windows 映像最佳化時，將自動刪除所有現存的預先佈建快照。必須完成此步驟，然後才能再次開始使用該功能。

6. 修改完成後，請選擇 Save changes (儲存變更)。

AWS CLI

停用 EC2 Fast Launch

使用下列 disable-fast-launch 命令在指定 AMI 上停用 EC2 Fast Launch，並清除現有的預先佈建快照。

```
aws ec2 disable-fast-launch --image-id ami-01234567890abcedf
```

以下為範例輸出。

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
```

```
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

PowerShell

停用 EC2 Fast Launch

使用 Disable-EC2FastLaunch cmdlet 在指定 AMI 上停用 EC2 Fast Launch，並清除現有的預先佈建快照。

```
Disable-EC2FastLaunch -ImageId ami-0abcdef1234567890
```

以下為範例輸出。

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
 MaxParallelLaunches   : 6
 OwnerId               : 0123456789123
 ResourceType          : snapshot
 SnapshotConfiguration :
 State                 : disabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime   : 2/25/2022 1:10:08 PM
```

在設定 EC2 Fast Launch 時使用啟動範本

透過啟動範本，您可以設定 Amazon EC2 每次從該範本啟動執行個體時使用的一組啟動參數。您可以指定 AMI 等項目，以用於基礎映像、執行個體類型、儲存體、網路設定等。

啟動範本是選擇性的，但下列特定情況除外；當您設定更快速啟動功能時，您必須為 Windows AMI 使用啟動範本：

- 您必須使用啟動範本為 Windows AMI 指定現有 VPC。若您使用 AWS 帳戶的預設 VPC，則此不適用。
- 如果您的帳戶包含對 Amazon EC2 執行個體強制實施 IMDSv2 的政策，您必須建立指定中繼資料組態強制執行 IMDSv2 的啟動範本。

  使用包含來自 EC2 主控台中繼資料組態的啟動範本，或在 中執行 enable-fast-launch 命令時 AWS CLI，或呼叫 EnableFastLaunch API 動作。

當您使用啟動範本時，EC2 Fast Launch 不支援下列組態。如果針對 EC2 Fast Launch 使用啟動範本，則不得指定以下項目：

- 使用者資料指令碼
- 終止保護
- 已停用中繼資料
- Spot 選項
- 終止執行個體的關機行為
- 網路介面、彈性圖形或 Spot 執行個體請求的資源標籤

指定 VPC

步驟 1：建立啟動範本

建立指定 Windows 執行個體下列詳細資訊的啟動範本：

- VPC 子網路。
- `t3.xlarge` 的執行個體類型。

如需詳細資訊，請參閱建立 Amazon EC2 啟動範本。

步驟 2：指定 EC2 Fast Launch AMI 的啟動範本

Console

為 EC2 Fast Launch 指定啟動範本

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Images (映像) 下，選擇 AMIs (AIM)。

3. 選取要更新的 AMI，方法選取相應 Name (名稱) 旁邊的核取方塊。
4. 從 AMI 清單上方的動作功能表中選擇設定快速啟動。這將開啟設定快速啟動頁面，您可以在其中設定 EC2 Fast Launch 的設定。
5. 此啟動範本方框會執行篩選搜尋，在您帳戶目前區域中搜尋與您輸入文字相符的啟動範本。在方框中指定所有或部分的啟動範本名稱或 ID，以比對出相符的啟動範本清單。例如，如果您在方框中輸入 `fast`，Amazon EC2 會在您帳戶目前的區域當中，搜尋所有名稱中包含「快速」的啟動範本。

   若要新建一個啟動範本，請選擇 Create launch template (建立啟動範本)。
6. 當您選取啟動範本時，Amazon EC2 會在 Source template version (來源範本版本) 方框中列出該範本的預設版本。若要指定其他版本，請反白要替換的預設版本，然後在方框中輸入所需的版本編號。
7. 修改完成後，請選擇 Save changes (儲存變更)。

AWS CLI

為 EC2 Fast Launch 指定啟動範本

使用 enable-fast-launch 命令並搭配 `--launch-template` 選項，指定啟動範本的名稱或 ID。

```
--launch-template LaunchTemplateName=my-launch-template
```

PowerShell

為 EC2 Fast Launch 指定啟動範本

使用 Enable-EC2FastLaunch cmdlet 搭配 `-LaunchTemplate_LaunchTemplateId` 或 `-LaunchTemplate_LaunchTemplateName` 參數。

```
-LaunchTemplate_LaunchTemplateName my-launch-template
```

如需 EC2 啟動範本的詳細資訊，請參閱 將執行個體啟動參數存放在 Amazon EC2 啟動範本中。

EC2 Fast Launch 的許可檢查

當您使用啟動範本啟用 EC2 Fast Launch 時，您可以指定範本的編號版本。您也可以使用 `$Latest`或 `$Default`版本。如果您使用其中一個別名，別名可以覆寫您打算限制的 `ec2:RunInstances`和 `iam:PassRole`許可。

當您啟用 EC2 Fast Launch 時，Amazon EC2 會根據目前的啟動範本版本檢查您的許可。 `$Latest`和 `$Default` 會在當時解析為特定版本。此檢查會在 Amazon EC2 啟用此功能之前執行。它會驗證啟動執行個體所需的許可，例如 `ec2:RunInstances`和 `iam:PassRole`許可。若要這樣做，EC2 Fast Launch 會發出`RunInstances`試轉呼叫，檢查您的許可，而不啟動執行個體。如果您的許可不允許必要的動作，Amazon EC2 會失敗請求，並傳回描述缺少許可的錯誤。Amazon EC2 不會啟用 EC2 Fast Launch。

在此初始檢查完成後，EC2 Fast Launch 會在背景中建立預先佈建的快照。然後，它會使用其服務連結角色的許可來啟動執行個體。每次啟動執行個體時，都會使用當時 `$Latest`或 `$Default` 解析的版本。即使版本在初次檢查後已變更，也是如此。它不會重新檢查您的許可。因此，可以更新啟動範本的人員可以將 IAM 角色或執行個體描述檔傳遞至執行個體。即使他們沒有該角色的 `iam:PassRole` 許可，也可能發生這種情況。

為了確保 Amazon EC2 在您啟用此功能時一律使用經過驗證的版本，請指定編號的啟動範本版本。不要使用 `$Latest` 或 `$Default`。

如果您將 EC2 Fast Launch 設定為使用 `$Latest`或 `$Default`啟動範本版本，我們建議您限制誰可以建立和管理啟動範本版本。使用 IAM 政策來限制對 `ec2:CreateLaunchTemplateVersion`和 等動作的存取`ec2:ModifyLaunchTemplate`。

檢視啟用了 EC2 Fast Launch 的 AMI

您可以使用 中的 describe-fast-launch-images 命令 AWS CLI，或 Get-EC2FastLaunchImage Tools for PowerShell Cmdlet，以取得已啟用 EC2 Fast Launch AMIs 詳細資訊。

Amazon EC2 為結果中傳回的每個 Windows AMI 提供以下詳細資訊：

- 已啟用 EC2 Fast Launch 的 AMI 的映像 ID。
- 用於預先佈建相關 Windows AMI 的資源類型。支援的值為：`snapshot`。
- 快照組態是一組參數，使用快照預先佈建關聯的 Windows AMI。
- 啟動範本資訊，包括相關 AMI 從預先佈建快照啟動 Windows 執行個體時使用的啟動範本的 ID、名稱和版本。
- 為建立資源而同時啟動的最大執行個體數。
- 關聯 AMI 的擁有者 ID。系統不會為與您共享的 AMI 填入此項目。
- 關聯 AMI 的 EC2 Fast Launch 的目前狀態。支援的值包括：`enabling | enabling-failed | enabled | enabled-failed | disabling | disabling-failed`。

> **Note**
>
> 您還可以在 EC2 主控台的 Manage image optimization (管理映像最佳化) 頁面中，看到目前狀態顯示為 Image optimization state (映像最佳化狀態)。

- 關聯 AMI 的 EC2 Fast Launch 變更為目前狀態的原因。
- 關聯 AMI 的 EC2 Fast Launch 變更為目前狀態的時間。

AWS CLI

尋找已為 EC2 Fast Launch 設定的 AMI

使用下列 describe-fast-launch-images 命令說明帳戶中為 EC2 Fast Launch 設定的每個 AMI 的詳細資訊。在此範例中，帳戶中只有一個 AMI 已設定了 EC2 Fast Launch。

```
aws ec2 describe-fast-launch-images
```

以下為範例輸出。

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

PowerShell

尋找已為 EC2 Fast Launch 設定的 AMI

使用下列 Get-EC2FastLaunchImage cmdlet 說明帳戶中為 EC2 Fast Launch 設定的每個 AMI 的詳細資訊。在此範例中，帳戶中只有一個 AMI 已設定了 EC2 Fast Launch。

```
Get-EC2FastLaunchImage -ImageId ami-0abcdef1234567890
```

以下為範例輸出。

```
ImageId               : ami-0abcdef1234567890
LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
OwnerId               : 012345678912
ResourceType          : snapshot
SnapshotConfiguration :
State                 : enabled
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 12:54:43 PM
```

管理 EC2 Fast Launch 基礎資源的成本

為 Windows AMI 設定 EC2 Fast Launch 無需支付服務費。不過，當您為 Amazon EC2 Windows AMI 啟用 EC2 Fast Launch 時，標準定價適用於 Amazon EC2 用於準備和存放預先佈建快照的基礎 AWS 資源。 Amazon EC2 您可以設定成本分配標籤，以協助追蹤和管理與 EC2 Fast Launch 資源相關聯的成本。如需如何設定成本分配標籤的詳細資訊，請參閱 追蹤帳單中的 EC2 Fast Launch 費用。

下列範例示範如何分配與 EC2 Fast Launch 快照成本相關聯的成本。

範例情境：AtoZ Example 公司擁有一個具有 50 GiB EBS 根磁碟區的 Windows AMI。它們為 AMI 啟用了 EC2 Fast Launch，並將目標資源計數設定為 5。在一個月的時間裡，為其 AMI 使用 EC2 Fast Launch 的成本約為 5.00USD，成本明細如下：

1. 當 AtoZ Example 啟用 EC2 Fast Launch 時，Amazon EC2 將啟動五個小型執行個體。每個執行個體都會執行 Sysprep 和 OOBE Windows 啟動步驟，並根據需要重新啟動。每個執行個體都需要幾分鐘的時間 (取決於該區域或可用區域 (AZ) 的忙碌程度以及 AMI 的大小，時間可能會有所不同)。

成本

- 執行個體執行時間成本 (或最低執行時間，如果適用)：五個執行個體
- 磁碟區成本：五個 EBS 根磁碟區

2. 預先佈建過程完成後，Amazon EC2 將產生執行個體的快照，並將其存放在 Amazon S3 中。快照通常會存放 4 到 8 小時，然後才會啟動使用。在這種情況下，每個快照的成本大約為 0.02 到 0.05 USD。

   成本

   - 快照儲存 (Amazon S3)：五個快照

3. Amazon EC2 拍攝快照後，即會停止執行個體。此時，執行個體將不再累計成本。然而，EBS 磁碟區成本繼續累計。

   成本

   - EBS 磁碟區：相關 EBS 根磁碟區的成本繼續存在。

> **Note**
>
> 此處顯示的費用僅供示範用途。您的成本會有所不同，具體取決於您的 AMI 組態和定價計劃。

追蹤帳單中的 EC2 Fast Launch 費用

成本分配標籤可協助您整理 AWS 帳單，以反映與 EC2 Fast Launch 相關的成本。您可使用 Amazon EC2 為 EC2 Fast Launch 準備和存放預先佈建的快照時，於其建立之資源中新增的下列標籤：

標籤索引鍵：`CreatedBy`，值：`EC2 Fast Launch`

在帳單與成本管理主控台中啟用標籤並設定詳細的帳單報告後，`user:CreatedBy` 欄出現在報告中。此欄包含來自所有服務的值。不過，如果您下載了 CSV 檔案，則可以將資料匯入試算表，並在值中篩選 `EC2 Fast Launch`。啟用標籤 AWS Cost and Usage Report 時，此資訊也會出現在 中。

步驟 1：啟用使用者定義的成本分配標籤

若要在成本報告中包含資源標籤，您必須先在帳單與成本管理主控台中啟用該標籤。如需詳細資訊，請參閱「AWS 帳單與成本管理 使用者指南」中的啟用使用者定義的成本分配標籤。

> **Note**
>
> 啟用最多可能需要 24 小時。

步驟 2：設定成本報告

如果您已設定成本報告，則下次啟用完成後執行報告時，標籤的資料欄會出現。若是第一次設定成本報告，請選擇以下其中一項。

- 請參閱「AWS 帳單與成本管理 使用者指南」中的設定每月成本分配報告。
- 請參閱「AWS Cost and Usage Report 使用者指南」中的建立成本和用量報告。

> **Note**
>
> 最多可能需要 24 小時 AWS 才能開始將報告交付到您的 S3 儲存貯體。

您可以使用 Amazon EC2 主控台、API、SDK、CloudFormation 或 AWS CLI中的 ec2 命令，為自己擁有的 Windows AMI 設定 EC2 Fast Launch。下列各節涵蓋 Amazon EC2 主控台和 的組態步驟 AWS CLI。

您也可以建立自訂 Windows AMI，這些 AMI 已使用 EC2 Image Builder 設定為實現 EC2 Fast Launch。如需詳細資訊，請參閱為啟用 EC2 Fast Launch 的 Windows AMI (AWS CLI) 建立分佈設定。

監控 EC2 Fast Launch

本節介紹如何在您的帳戶中監控已啟用 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI。

透過 EventBridge 監控 EC2 Fast Launch 狀態變更

當 Windows AMI 的狀態變更且啟用了 EC2 Fast Launch 時，Amazon EC2 會產生一個 `EC2 Fast Launch State-change Notification` 事件。然後 Amazon EC2 會將狀態變更事件傳送至 Amazon EventBridge (之前稱為 Amazon CloudWatch Events)。

您可以建立 EventBridge 規則，以便觸發一個或多個動作以回應狀態變更事件。例如，您可以建立 EventBridge 規則，以偵測何時啟用 EC2 Fast Launch，並執行下列動作：

- 將訊息傳送至 Amazon SNS 主題，通知其訂閱用戶。

- 叫用執行某些動作的 Lambda 函數。
- 將狀態變更資料傳送至 Amazon Data Firehose 進行分析。

如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的建立對事件做出反應的 Amazon EventBridge 規則。

狀態變更事件

EC2 Fast Launch 功能會盡力發出 JSON 格式的狀態變更事件。Amazon EC2 會以幾乎即時的速度將事件傳送至 EventBridge。本節說明事件欄位，並顯示事件格式範例。

EC2 Fast Launch State-change Notification

imageId

識別具有 EC2 Fast Launch 狀態變更的 AMI。

resourceType

用於預先佈建的資源類型。支援的值為：`snapshot`。預設值為 `snapshot`。

state

指定 AMI 的 EC2 Fast Launch 功能的目前狀態。有效值包括下列項目：

- 正在啟用 — 您已為 AMI 啟用 EC2 Fast Launch 功能，而 Amazon EC2 已開始為預先佈建程序建立快照。
- 啟用失敗 — 發生錯誤，導致第一次啟用 AMI 的 EC2 Fast Launch 時，預先佈建程序失敗。這可能會在預先佈建程序期間隨時發生。
- 已啟用 — EC2 Fast Launch 功能已啟用。Amazon EC2 為新啟用的 EC2 Fast Launch AMI 建立第一個預先佈建的快照後，狀態就會變更為 `enabled`。如果 AMI 已啟用並再次進行預先佈建，狀態變更會立即發生。
- 啟用失敗 — 此狀態只有在 EC2 Fast Launch AMI 不是第一次進行預先佈建程序時才適用。如果 EC2 Fast Launch 功能停用，稍後再次啟用，或者在第一次完成預先佈建後發生組態變更或其他錯誤，就會發生這種情況。
- 正在停用 — AMI 擁有者已關閉 AMI 的 EC2 Fast Launch 功能，而且 Amazon EC2 已開始清理程序。
- 已停用 — EC2 Fast Launch 功能已停用。Amazon EC2 完成清理程序後，狀態就會變更為 `disabled`。

- 禁用失敗 — 出現問題，導致清理程序失敗。這意味著某些預先佈建的快照可能仍會保留在帳戶中。

stateTransitionReason

EC2 Fast Launch AMI 狀態變更的原因。

> **Note**
>
> 此事件訊息中的所有欄位都是必填欄位。

下列範例顯示新啟用的 EC2 Fast Launch AMI，它已啟動第一個執行個體以開始預先佈建程序。此時，狀態為 `enabling`。Amazon EC2 建立第一個預先佈建的快照後，狀態會變更為 `enabled`。

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-012345678901",
 "detail-type": "EC2 Fast Launch State-change Notification",
 "source": "aws.ec2",
 "account": "123456789012",
 "time": "2022-08-31T20:30:12Z",
 "region": "us-east-1",
 "resources": [
  "arn:aws:ec2:us-east-1:123456789012:image/ami-123456789012"
 ],
 "detail": {
  "imageId": "ami-123456789012",
  "resourceType": "snapshot",
  "state": "enabling",
  "stateTransitionReason": "Client.UserInitiated"
 }
}
```

使用 CloudWatch 監控 EC2 Fast Launch 指標

啟用 EC2 Fast Launch 功能的 Amazon EC2 AMI 將指標傳送到 Amazon CloudWatch。您可以使用 AWS 管理主控台、 AWS CLI或 API 來列出 EC2 Fast Launch 傳送至 CloudWatch 的指標。`AWS/EC2` 命名空間包含下列 EC2 Fast Launch 指標：

指標	Description
NumberOfAvailableFastLaunchSnapshots	每個已啟用 EC2 Fast Launch 的 AMI 可用的預先佈建快照數量。
NumberOfInstancesFastLaunched	每個已啟用 EC2 Fast Launch 之 AMI 的執行個體數目，AMI 已從預先佈建快照啟動。
NumberOfInstancesNotFastLaunched	每個已啟用 EC2 Fast Launch 之 AMI 的執行個體數目，此數量會由於在啟動期間缺少可用的預先佈建快照而導致冷開機。
FastLaunchSnapshotUsedToRefillStartTime	Amazon EC2 從啟用 EC2 Fast Launch 之 AMI 啟動新映像時的時間戳記，以在使用現有快照後建立另一個快照。
FastLaunchSnapshotCreationTime	測量 Amazon EC2 啟動執行個體以及啟用 EC2 Fast Launch 之 AMI 建立快照所花費的時間。

用於 EC2 Fast Launch 的服務連結角色

Amazon EC2 使用許可的服務連結角色，它需要代表您呼叫其他 AWS 服務 。服務連結角色是直接連結至 的唯一 IAM 角色類型 AWS 服務。服務連結角色提供將許可委派給 的安全方式， AWS 服務 因為只有連結的服務可以擔任服務連結角色。如需 Amazon EC2 如何使用 IAM 角色的詳細資訊，請參閱 Amazon EC2 的 IAM 的角色。

Amazon EC2 使用名為 AWSServiceRoleForEC2FastLaunch 的服務連結角色建立和管理一組預先佈建的快照，從而縮短從 Windows AMI 啟動執行個體所需的時間。

AWSServiceRoleForEC2FastLaunch 授予的許可

AWSServiceRoleForEC2FastLaunch 服務連結角色信任下列服務來擔任此角色：

- `ec2fastlaunch.amazonaws.com`

Amazon EC2 使用 EC2FastLaunchServiceRolePolicy 受管政策來完成下列動作：

- AWS CloudFormation – 允許 EC2 Fast Launch 取得關聯 CloudFormation 堆疊的描述。

- Amazon CloudWatch – 將與 EC2 Fast Launch 相關聯的指標資料發佈到 Amazon EC2 命名空間。
- Amazon EC2 – 授予 EC2 Fast Launch 執行下列動作的存取權：
  - 從已啟用 Amazon EC2 EC2 Windows Server AMI 啟動執行個體，以執行佈建步驟。另需指定資源模式，允許對與 License Manager 相關聯的 AMI 執行 `ec2:RunInstances`。
  - 在建立預先佈建的快照後，停止並終止由 EC2 Fast Launch 啟動的執行個體。
  - 描述用於從已啟用 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 啟動執行個體的映像與執行個體類型資源，並從中建立快照。
  - 描述啟動範本資源，然後從啟動範本啟動執行個體。
  - 描述執行個體、執行個體屬性與執行個體狀態、磁碟區與磁碟區屬性。
  - 描述網路介面
  - 刪除 EC2 Fast Launch 建立的資源，包括快照、啟動範本、磁碟區與網路介面。
  - 為 EC2 Fast Launch 建立的資源加標籤，這些資源用於啟動和預先佈建 Windows 執行個體，並建立供最終啟動程序使用的快照。
- Amazon EventBridge – 包含建立 EventBridge 事件規則、擷取其建立規則的詳細資訊或刪除這些規則的存取權。EC2 Fast Launch 亦可取得接收根據事件規則轉送之 EC2 Fast Launch 事件的目標服務清單，並在其建立的事件規則中新增或移除目標服務。
- IAM – 允許 EC2 Fast Launch 建立 `EC2FastLaunchServiceRolePolicy` 服務連結角色、取得並使用名稱包含 `ec2fastlaunch` 的執行個體設定檔，以及使用您啟動範本中的執行個體設定檔代表您啟動執行個體。
- AWS KMS – 包含建立授權、列出由 EC2 Fast Launch 建立且可淘汰的授權的存取權。另可描述或使用金鑰，以加密或解密連接至 EC2 Fast Launch 建立之執行個體的磁碟區，並產生非明文的資料金鑰。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 EC2FastLaunchServiceRolePolicy。

如需使用 Amazon EC2 受管政策的詳細資訊，請參閱 AWS Amazon EC2 的 受管政策。

建立服務連結角色

您不需要手動建立此服務連結角色。當您開始為 AMI 使用 EC2 Fast Launch 時，Amazon EC2 會自動建立服務連結角色 (如果尚不存在此角色)。

如果從您的帳戶中刪除服務連結角色，您可以開始為另一個 Windows AMI 啟用 EC2 Fast Launch，從而在您的帳戶中重新建立此角色。或者，您可以針對目前 AMI 停用 EC2 Fast Launch，然後再次啟用。但是，停用該功能會導致 AMI 對所有新執行個體使用標準啟動過程，而 Amazon EC2 會移除

您的所有預先佈建快照。所有預先佈建的快照都被刪除之後，您可以再次開始為 AMI 啟用 EC2 Fast Launch。

存取客戶自管金鑰

要為使用客戶自管金鑰 (CMK) 加密的加密的 AMI 啟用 EC2 Fast Launch，您必須授予 AWSServiceRoleForEC2FastLaunch 角色使用該 CMK 的許可。為此，呼叫 create-grant 命令 針對 `--grantee-principal`，指定您帳戶中 AWSServiceRoleForEC2FastLaunch 角色的 ARN。對於 `--operations`，請指定 `CreateGrant`。

```
aws kms create-grant \
    --key-id arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2FastLaunch \
    --operations CreateGrant
```

編輯服務連結角色

Amazon EC2 不允許您編輯 AWSServiceRoleForEC2FastLaunch 服務連結角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的編輯服務連結角色。

刪除服務連結角色

您必須先刪除所有相關的資源，才能刪除服務連結角色。這樣可保護與已啟用 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 相關聯的 Amazon EC2 資源，避免無意間移除了資源的存取許可。

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForEC2FastLaunch 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

支援的區域

Amazon EC2 支援所有提供 Amazon EC2 服務的區域中的 EC2 Fast Launch 服務連結角色。

對 Windows EC2 Fast Launch 進行故障診斷

故障診斷方案

下列案例可協助您診斷並解決啟用 EC2 Fast Launch 時可能遇到的常見問題。

無法停止執行個體以建立快照

說明

啟用 EC2 Fast Launch 時，此服務會啟動一組執行個體，用於建立預先佈建的快照。每個執行個體有 30 分鐘時間完成此程序。若任一執行個體順利完成，此服務會將 AMI 的 Fast Launch 狀態設定為 `Enabled`。不過，若執行個體無法在分配的時間內完成程序，且其他任何執行個體均未完成程序，則此服務會終止所有執行個體，將 AMI 的 Fast Launch 狀態設定為 `enabling_failed`，Fast Launch 狀態原因如下：

```
Unable to stop instance ID=i-1234567890abcdef0 for snapshot creation.
```

原因

最常見的原因是，嘗試為從執行中執行個體建立的 Windows AMI，或未符合所有 EC2 Fast Launch 先決條件 的 AMI 啟用 EC2 Fast Launch。

解決方案

確保您使用的 AMI 符合所有的 EC2 Fast Launch 先決條件。

若要為 AMI 設定 EC2 Fast Launch，必須使用帶關機選項的 Sysprep 來建立 AMI。如需詳細資訊，請參閱使用 Windows Sysprep 建立 Amazon EC2 AMI。

您已達到 VPC 限制

說明

如果您不使用啟動範本來指定現有的 VPC，而且您的帳戶沒有定義預設 VPC，則服務會自動建立包含 VPC 和其他資源的 CloudFormation 堆疊，如中所述EC2 Fast Launch 先決條件。

原因

您的 AWS 帳戶 在該區域已達到允許建立的 VPC 數量上限，且未為 EC2 Fast Launch 指定要使用的現有 VPC。這將會導致程序失敗。

解決方案

您可使用下列其中一種選項來解決此問題：

- 您可以要求增加配額

- 您可以提供指定現有 VPC 的啟動範本

若要要求提高您的帳戶每個區域可定義的 VPC 數量，請遵循下列步驟：

1. 開啟 Service Quotas 主控台，網址為 https://console.aws.amazon.com/servicequotas/。
2. 在服務主控台儀表板中，選擇 Amazon Virtual Private Cloud (Amazon VPC)。這會開啟 VPC Service Quotas。
3. 依 `VPCs per Region` 篩選，直接前往該配額頁面。
4. 選取每個區域的 VPC 數，然後選擇在帳戶層級請求增加。

若您有緊急配額請求，或您的配額增加請求遭拒，請聯絡 Support 尋求協助。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的請求提高配額。

用來啟用 EC2 Fast Launch 的許可不足

說明

首次啟用 EC2 Fast Launch 且未指定啟動範本時，EC2 Fast Launch 會使用服務預設資源建立服務專屬的 CloudFormation 堆疊。但若您的 IAM 主體 (角色或使用者) 缺乏必要許可，CloudFormation 範本將無法部署。

日誌訊息可能如下所示：

```
Can't enable EC2 Fast Launch. The IAM credentials that you are using do not have
 sufficient permissions. Attach EC2FastLaunchFullAccess in the IAM console.
```

原因

您的 IAM 使用者或角色缺乏啟用 EC2 Fast Launch 的必要許可。

解決方案

確認用於啟用 EC2 Fast Launch 的 IAM 主體 (使用者或角色) 已連接 `EC2FastLaunchFullAccess` 政策。此 AWS 受管政策授予所有 EC2 Fast Launch 資源的完整存取權。若要檢視此政策的許可，請參閱《AWS 受管政策參考》中的 EC2FastLaunchFullAccess 政策。

變更 Amazon EC2 執行個體的 Windows 管理員密碼

如果您從 AWS Windows AMI 啟動執行個體，預先安裝的啟動代理程式會設定預設密碼，如下所示：

- 對於 Windows Server 2022 及更新版本，EC2Launch v2 會產生預設密碼。
- 對於 Windows Server 2016 及 2019，EC2Launch 代理程式會產生預設密碼。
- 對於 Windows Server 2012 R2 及較早版本，EC2Config 服務 會產生預設密碼。

> **Note**
>
> 對於 Windows Server 2016 及更高版本的 AMI，本機管理員會停用 `Password never expires`。對於 Windows Server 2016 之前的 AMI 版本，`Password never expires` 會為本機管理員啟用。

連線後變更管理員密碼

當您第一次連線至執行個體時，建議您變更預設的管理員密碼。使用下列程序變更 Windows 執行個體的管理員密碼。

> **Important**
>
> 將新密碼存放於安全處。您無法使用 Amazon EC2 主控台擷取新密碼。主控台只能擷取預設的密碼。如果您在變更密碼後嘗試使用預設密碼連線執行個體，您會收到 "Your credentials did not work" (您的登入資料無法作用) 錯誤。

變更本機管理員密碼

1. 連線至執行個體並開啟命令提示。
2. 執行下列命令。如果您的新密碼包含特殊字元，請務必以雙引號括住密碼。

```
net user Administrator "new_password"
```

3. 將新密碼存放於安全處。

變更遺失或過期的密碼

如果您的密碼遺失或過期，您可產生新的密碼。如需密碼重設程序，請參閱重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼。

將選用的 Windows Server 元件新增至 Amazon EC2 Windows 執行個體

若要存取及安裝選用元件，您必須尋找屬於您版本 Windows Server 的正確 EBS 快照，從快照建立磁碟區，然後將磁碟區連接到您的執行個體。

開始之前

使用 AWS 管理主控台 或命令列工具來取得執行個體的執行個體 ID 和可用區域。您必須在與您執行個體相同的可用區域中建立您的 EBS 磁碟區。

使用下列程序之一將 Windows Server 元件新增至您的執行個體。

Console

如需為您的執行個體新增 Windows 元件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 從 Filter (篩選條件) 列中，選擇 Public Snapshots (公有快照)。
4. 新增 Owner Alias (擁有者別名) 篩選條件，然後選擇 amazon。
5. 新增 Description (描述) 篩選條件，然後輸入 **`Windows`**。
6. 按 Enter
7. 選取符合您系統架構及語言偏好設定的快照。例如，若執行個體執行 Windows Server 2019，請選取 Windows 2019 English Installation Media (Windows 2019 英文版安裝媒體)。
8. 選擇 Actions (動作)、Create volume from snapshot (從快照建立磁碟區)。
9. 針對可用區域，選取符合您 Windows 執行個體的可用區域。選擇 Add tag (新增標籤)，然後為標籤索引鍵指定 **`Name`**，並為標籤值指定描述性名稱。選擇建立磁碟區。
10. 在 Volume Successfully Created (磁碟區建立成功) 訊息 (綠色橫幅) 中，選擇您剛剛建立的磁碟區。
11. 選擇 Actions (動作)、Attach Volume (連接磁碟區)。
12. 從 Instance (執行個體) 中，選取執行個體 ID。
13. 對於 Device name (裝置名稱)，輸入配件的裝置名稱。如需裝置名稱的說明，請參閱 Amazon EC2 執行個體上磁碟區的裝置名稱。
14. 選擇 Attach volume (連接磁碟區)。
15. 連線到您的執行個體，然後讓磁碟區可用。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的讓 Amazon EBS 磁碟區可供使用。

> **Important**
>
> 請不要初始化磁碟區。

16. 開啟 Control Panel (控制面板)，Programs and Features (程式和功能)。選擇 Turn Windows features on or off (開啟或關閉 Windows 功能)。若您收到提示要求安裝媒體，請指定使用安裝媒體的 EBS 磁碟區。
17. (選用) 完成媒體安裝後，您可以分離磁碟區。分離磁碟區後，即可將它刪除。

AWS CLI

如需為您的執行個體新增 Windows 元件

1. 使用 describe-snapshots 命令，搭配 `owner-ids` 參數和 `description` 篩選條件，取得可用安裝媒體快照的清單。

```
aws ec2 describe-snapshots \
    --owner-ids amazon \
    --filters Name=description,Values=Windows*
```

2. 在輸出中，注意符合您系統架構和語言偏好設定的快照 ID。例如：

```
{
    "Snapshots": [
    ...
        {
            "OwnerAlias": "amazon",
            "Description": "Windows 2019 English Installation Media",
            "Encrypted": false,
            "VolumeId": "vol-be5eafcb",
            "State": "completed",
            "VolumeSize": 6,
            "Progress": "100%",
            "StartTime": "2019-10-25T20:00:47.000Z",
            "SnapshotId": "snap-22da283e",
            "OwnerId": "123456789012"
        },
    ...
   ]
```

```
}
```

3. 使用 create-volume 命令從快照建立磁碟區。指定與您執行個體相同的可用區域。

```
aws ec2 create-volume \
    --snapshot-id snap-0abcdef1234567890 \
    --volume-type gp2 \
    --availability-zone us-east-1a
```

4. 在輸出中，記下磁碟區 ID。

```
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-01234567890abcdef",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateTime": "2017-04-18T10:33:10.940Z",
    "Size": 6
}
```

5. 使用 attach-volume 命令將磁碟區連接到您的執行個體。

```
aws ec2 attach-volume \
    --volume-id vol-0c98b37f30bcbc290 \
    --instance-id i-01474ef662b89480 \
    --device xvdg
```

6. 連線到您的執行個體，然後讓磁碟區可用。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的讓 Amazon EBS 磁碟區可供使用。

> **Important**
>
> 請不要初始化磁碟區。

7. 開啟 Control Panel (控制面板)，Programs and Features (程式和功能)。選擇 Turn Windows features on or off (開啟或關閉 Windows 功能)。若您收到提示要求安裝媒體，請指定使用安裝媒體的 EBS 磁碟區。

8. (選用) 完成媒體安裝後，請使用 detach-volume 命令將磁碟區與您的執行個體分離。分離磁碟區後，您可以使用 delete-volume 命令來刪除磁碟區。

PowerShell

如需為您的執行個體新增 Windows 元件

1. 使用 Get-EC2Snapshot Cmdlet，搭配 Owner 和 description 篩選條件，取得可用安裝媒體快照的清單。

```
Get-EC2Snapshot `
    -Owner amazon `
    -Filter @{ Name="description"; Values="Windows*" }
```

2. 在輸出中，注意符合您系統架構和語言偏好設定的快照 ID。例如：

```
...
DataEncryptionKeyId :
Description         : Windows 2019 English Installation Media
Encrypted           : False
KmsKeyId            :
OwnerAlias          : amazon
OwnerId             : 123456789012
Progress            : 100%
SnapshotId          : snap-0abcdef1234567890
StartTime           : 10/25/2019 8:00:47 PM
State               : completed
StateMessage        :
Tags                : {}
VolumeId            : vol-01234567890abcdef
VolumeSize          : 6
...
```

3. 使用 New-EC2Volume Cmdlet 從快照建立磁碟區。指定與您執行個體相同的可用區域。

```
New-EC2Volume `
    -AvailabilityZone us-east-1a `
    -VolumeType gp2 `
    -SnapshotId snap-0abcdef1234567890
```

4. 在輸出中，記下磁碟區 ID。

```
Attachments      : {}
AvailabilityZone : us-east-1a
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-0abcdef1234567890
State            : creating
Tags             : {}
VolumeId         : vol-01234567890abcdef
VolumeType       : gp2
```

5. 使用 Add-EC2Volume Cmdlet 將磁碟區連接到您的執行個體。

```
Add-EC2Volume `
    -InstanceId i-1234567890abcdef0 `
    -VolumeId vol-01234567890abcdef `
    -Device xvdh
```

6. 連線到您的執行個體，然後讓磁碟區可用。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的讓 Amazon EBS 磁碟區可供使用。

> **Important**
>
> 請不要初始化磁碟區。

7. 開啟 Control Panel (控制面板)，Programs and Features (程式和功能)。選擇 Turn Windows features on or off (開啟或關閉 Windows 功能)。若您收到提示要求安裝媒體，請指定使用安裝媒體的 EBS 磁碟區。
8. (選用) 完成媒體安裝後，請使用 Dismount-EC2Volume Cmdlet 將磁碟區與您的執行個體分離。分離磁碟區後，您可以使用 Remove-EC2Volume Cmdlet 移除該磁碟區。

在 EC2 Windows 執行個體安裝適用於 Linux 的 Windows 子系統

Windows Subsystem for Linux (WSL) 是 Microsoft Windows 的一項功能。透過在 EC2 Windows 執行個體上安裝 WSL，您可以直接在 Windows 執行個體上執行原生 Linux 工具。

Windows Subsystem for Linux (WSL) 有兩個版本：WSL 1 與 WSL 2。若要了解詳細資訊，請參閱 Microsoft 網站上的 Windows Subsystem for Linux 文件。

要求

- 作業系統必須為 Windows Server 2019 或更新版本。
- 您只能在虛擬化 Windows 執行個體上安裝 WSL 1 （執行個體大小不是.metal或不支援巢狀虛擬化）。
- 您可以在支援巢狀虛擬化並啟用 NestedVirtualization CPU 選項的執行個體上安裝 WSL 1 或 WSL 2。
- 您可在裸機執行個體上安裝 WSL 1 或 WSL 2 (執行個體大小為 .metal)。根據預設，裸機執行個體提供巢狀虛擬化所需的支援。

如需 EC2 巢狀虛擬化的詳細資訊，請參閱 使用巢狀虛擬化在 Amazon EC2 執行個體中執行 Hypervisor。

在 Windows 執行個體上安裝 WSL

若要安裝 WSL 1

1. 安裝 WSL。您將使用的程序取決於在執行個體上執行的 Windows Server 版本。
   - Windows Server 2022 及更新版本 - 在 EC2 執行個體上執行以下標準安裝命令。

     ```
     wsl --install --enable-wsl1 --no-launch
     ```

   - Windows Server 2019 - 啟用 WSL，然後依據 Microsoft 網站上的在舊版 Windows Server 上安裝 WSL 所述指示來安裝 WSL。
2. 重新啟動 EC2 執行個體。

   ```
   shutdown -r -t 20
   ```

3. 若要設定 WSL 以使用 WSL 1，請在執行個體上執行下列命令。虛擬化執行個體需要此步驟（執行個體大小未.metal設定為巢狀虛擬化）。

   ```
   wsl --set-default-version 1
   ```

4. 安裝預設發行版本。

```
wsl --install
```

安裝 WSL 2 (.metal 或啟用巢狀虛擬化的執行個體)

請在 EC2 執行個體上執行下列標準安裝命令。依預設，會安裝 WSL 2。

```
wsl --install
```

EC2 Windows 疑難排解公用程式

EC2WinUtil 驅動程式能為您的 Windows 執行個體提供以下類型的疑難排解支援。

損毀呼叫堆疊

EC2WinUtil 會從執行個體收集基本當機資訊，並將其寫入序列主控台。下列清單包含公用程式寫入主控台的一些關鍵詳細資訊。

- 識別產生故障的模組。
- 與此 Windows 事件相關的錯誤代碼。
- 最近呼叫的堆疊追蹤。

透過這些詳細資訊，您可以執行初始根本原因分析，並判斷是否需要進一步分析。輸出至序列主控台也可讓 AWS 追蹤 Amazon EC2 驅動程式的當機趨勢，並診斷大規模當機事件。

> **Note**
>
> EC2WinUtil 不會在其當機呼叫堆疊中收集任何客戶資料。

如需驅動程式版本備註，請參閱 EC2 Windows 公用程式驅動程式版本歷史記錄。

EC2 Windows 公用程式驅動程式版本歷史記錄

下列資料表顯示 Amazon EC2 之每個版本的 Windows Server 上執行的 EC2WinUtil 驅動程式。早期版本的作業系統使用預先安裝在 AWS Windows Server AMI 上的驅動程式，而執行個體則從該 Windows Server AMI 啟動。與您共用或您透過 訂閱的 AMIs AWS Marketplace 未預先安裝驅動程式。

Windows Server 版本	EC2WinUtil 驅動程式版本
Windows Server 2025	最新版本
Windows Server 2022	最新版本
Windows Server 2019	最新版本
Windows Server 2016	最新版本

> **Note**
>
> 在驅動程式 3.0.0 版之前，EC2WinUtil 驅動程式無法下載以供手動安裝。舊版僅適用於 Windows AWS AMIs 預先安裝的驅動程式。

下表說明 EC2WinUtil 驅動程式的發行版本。

套件版本	驅動程式版本	詳細資訊	發行日期
3.2.0	3.2.0	用於 EC2Launch v2 代理程式的增強型主控台記錄機制。	2026 年 6 月 2 日
3.1.1	3.1.1	記錄到主控台輸出時增加通話堆疊長度。	2026 年 3 月 3 日
3.1.0	3.1.0	改善電源管理事件處理。	2026 年 2 月 4 日
3.0.0	3.0.0	針對 Windows 10 將驅動程式現代化，並新增支援以原始驅動程式安裝。	2024 年 6 月 13 日
2.0.0	2.0.0	已針對 metal 執行個體類型新增對 MMIO 序列埠輸出的支援。同時改善當機剖析並更新輸出格式。	2018 年 8 月 23 日
1.0.1	1.0.1	由於命名空間與 Amazon Inspector 衝突，因此將驅動程式名稱變更為 EC2WinUtil 。包含數個錯誤修正。	2018 年 3 月 1 日

套件版本	驅動程式版本	詳細資訊	發行日期
1.0.0	1.0.0	初始版本。驅動程式最初稱為 AwsAgent。	2017 年 11 月 28 日

將 EC2 Windows 執行個體升級至較新版本的 Windows Server

您可使用以下其中一種方法，在該升級時，從舊版升級 EC2 Windows 執行個體上的 Windows Server 作業系統。

就地升級

就地升級會在現有執行個體上執行。僅作業系統檔案會在此過程期間受到影響，而您的設定、伺服器角色和資料皆會維持不變。

遷移 (也稱為並排升級)

遷移會涉及擷取設定、組態和資料，並將這些移植到全新 EC2 Windows 執行個體上的較新作業系統。您可以從您從 訂閱的公有或私有 Windows AMI AWS Marketplace，或與您共用的 AMI 啟動執行個體。您也可以使用 EC2 Image Builder 建立自訂 AMI。如需詳細資訊，請參閱 Image Builder 使用者指南。

> **Note**
>
> AWS 為在 EC2 執行個體上執行的 Windows Server 版本提供一組公開可用的 Amazon Machine Image (AMIs)。這些 AMI 會每月更新。如需最新的 Windows AMI 相關資訊，請參閱 AWS Windows AMI 參考。

Microsoft 傳統上都建議遷移到較新版本的 Windows Server，而非就地升級。遷移可能會減少升級錯誤或問題，但相較於就地升級可能需要更長的時間，因為需要佈建新的執行個體、計劃及移植應用程式，以及調整新執行個體上的組態設定。就地升級的速度較快，但軟體不相容可能會產生錯誤。

目錄

- 在 EC2 Windows 執行個體上執行就地升級
- 使用自動化執行手冊來升級 EC2 Windows 執行個體
- 將 EC2 Windows 執行個體遷移至 Nitro 型執行個體類型

- 對 EC2 Windows 執行個體上的作業系統升級進行疑難排解

在 EC2 Windows 執行個體上執行就地升級

在您執行就地升級前，您必須判斷執行個體執行的網路驅動程式為何。PV 網路驅動程式可讓您使用遠端桌面存取您的執行個體。執行個體使用 AWS PV、Intel 網路轉接器或增強型網路驅動程式。如需詳細資訊，請參閱Windows 執行個體的全虛擬驅動程式。

在您開始就地升級之前

在您開始您的就地升級之前，請完成下列任務並記下以下重要詳細資訊。

- 閱讀 Microsoft 文件，了解升級需求、已知問題和限制。同時也請檢閱升級的正式說明。
  - Upgrade Options for Windows Server 2012
  - Upgrade Options for Windows Server 2012 R2
  - Windows Server 2016 及更新版本的升級或轉換選項
  - Windows Server 升級
- 我們建議在至少配備 2 個 vCPU 或 4GB RAM 的執行個體上執行作業系統升級。必要時，您可以將執行個體變更為相同類型但更大的執行個體 (例如，將 t2.small 變更為 t2.large)、執行升級，然後再調整回原始大小。如果您需要保留執行個體大小，您可以使用執行個體主控台螢幕擷取畫面來監控進度。如需詳細資訊，請參閱Amazon EC2 執行個體類型變更。
- 確認您 Windows 執行個體的根磁碟區具有足夠的可用磁碟空間。Windows 安裝程序可能不會警告您的磁碟空間不足。如需升級至特定作業系統所需磁碟空間的資訊，請參閱 Microsoft 文件。若磁碟區沒有足夠的空間，您可以擴展它。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 彈性磁碟區。
- 決定您的升級途徑。您必須將作業系統升級至相同的架構。例如，您必須將 32 位元系統升級至 32 位元系統。Windows Server 2008 R2 及更新版本為僅限 64 位元。
- 停用防毒軟體、反間諜軟體及防火牆。這些軟體類型可能會和升級程序衝突。請在升級完成之後再次啟用防毒軟體、反間諜軟體及防火牆。
- 如 將 EC2 Windows 執行個體遷移至 Nitro 型執行個體類型 所述更新至最新的驅動程式。
- 升級協助程式服務僅支援執行 Citrix PV 驅動程式的執行個體。若執行個體執行的是 Red Hat 驅動程式，您必須先手動升級那些驅動程式。

使用 AWS PV、Intel 網路轉接器或增強型網路驅動程式，就地升級執行個體

透過下列程序，使用 AWS PV、Intel 網路介面卡或增強聯網網路驅動程式升級 Windows Server 執行個體。

執行就地升級

1. 為備份或測試用途，建立您計劃升級之系統的 AMI。您接著便能在複本上執行升級，模擬測試環境。若升級完成，您便可以將流量切換至此執行個體，而無須經歷太多停機時間。若升級失敗，您可以回復至備份。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。
2. 確認您的 Windows Server 執行個體使用最新的網路驅動程式。
   a. 若要更新您的 AWS PV 驅動程式，請參閱 升級 EC2 Windows 執行個體上的 PV 驅動程式。
   b. 若要更新您的 ENA 驅動程式，請參閱 在 EC2 Windows 執行個體上安裝 ENA 驅動程式。
   c. 若要更新您的 Intel 驅動程式，請參閱 已使用 Intel 82599 VF 介面來強化網路。
3. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
4. 在導覽窗格中，選擇 Instances (執行個體)。尋找執行個體。記下執行個體的執行個體 ID 和可用區域。您在此程序的後續步驟中將需要此資訊。
5. 若您要從 Windows Server 2012 或 2012 R2 升級至 Windows Server 2016 或更新版本，請在您的執行個體上執行下列操作再繼續。
   a. 解除安裝 EC2Config 服務。如需詳細資訊，請參閱EC2Launch v2 和 EC2Config 代理程式的 Windows 服務管理。
   b. 安裝 EC2Launch v1 或 EC2Launch v2 代理程式。如需詳細資訊，請參閱 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v1 代理程式執行任務和 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v2 代理程式執行任務。
   c. 安裝 AWS Systems Manager SSM 代理程式。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的在適用於 Windows Server 的 Amazon EC2 上手動安裝 SSM Agent。
6. 從 Windows Server 安裝媒體快照建立新的磁碟區。
   a. 在左側導覽窗格中，於 Elastic Block Store 下方，選擇 Snapshots (快照)。
   b. 從篩選條件列中，選擇公有快照。
   c. 在搜尋列中，指定下列篩選條件：
      - 選擇所有者別名，然後選擇 =，最後選擇 amazon。

- 選擇說明，然後開始輸入 **Windows**。選取與您要升級到的系統架構及語言偏好設定相符的 Windows 篩選條件。例如，選擇 Windows 2019 英文版安裝媒體以升級至 Windows Server 2019。

d. 選取與您要升級到的系統架構和語言偏好設定相符的快照旁邊的核取方塊，然後選擇動作、從快照建立磁碟區。

e. 在建立磁碟區頁面中，選擇符合 Windows 執行個體的可用區域，然後選擇建立磁碟區。

7. 在頁面頂端的成功建立的磁碟區 vol-***1234567890example*** 橫幅中，選擇您剛才建立的磁碟區 ID。
8. 選擇 Actions (動作)、Attach Volume (連接磁碟區)。
9. 在附接磁碟區頁面中，對於執行個體，選取 Windows 執行個體的執行個體 ID，然後選擇附接磁碟區。
10. 依照使 Amazon EBS 磁碟區可使用中的下列步驟，使新的磁碟區可供使用。

> **Important**
>
> 請勿初始化磁碟，因為這樣做會刪除現有資料。

11. 在 Windows PowerShell，切換到新的磁碟區磁碟機。透過開啟連接到執行個體的安裝媒體磁碟區，開始升級。

a. 若您要升級至 Windows Server 2016 或更新版本，請執行下列項目：

```
.\setup.exe /auto upgrade /dynamicupdate disable
```

> **Note**
>
> 在將 `/dynamicupdate` 選項設定為停用的情況下執行 setup.exe 會阻止 Windows 在 Windows Server 升級程序期間中安裝更新，因為在升級期間中安裝更新可能會導致失敗。升級完成後，您可以使用 Windows Update 安裝更新。

若您要升級至較早版本的 Windows Server，請執行下列項目：

```
Sources\setup.exe
```

b. 針對選取您要安裝的作業系統，選取適用於 Windows Server 執行個體的完整安裝選項，然後選擇下一步。

c. 針對 Which type of installation do you want? (您希望的安裝類型為何？)，選擇 Upgrade (升級)。

d. 完成協助程式。

Windows Server 安裝程式會複製並處理檔案。在幾分鐘之後，您的遠端桌面工作階段便會關閉。升級所需要花費的時間取決於在您 Windows Server 執行個體上執行之應用程式和伺服器角色的數目。升級程序最少可能需要花費 40 分鐘至數小時。在升級過程中，執行個體可能無法通過一個或多個狀態檢查。升級完成後，所有狀態檢查都會通過。您可以檢查主控台輸出的系統日誌，或使用磁碟和 CPU 活動的 Amazon CloudWatch 指標判斷升級是否正在進行中。

> **Note**
>
> 若升級至 Windows Server 2019，在升級完成之後，若需要的話，您可以手動變更桌面背景，移除先前作業系統的名稱。

若執行個體在數小時之後仍無法通過所有狀態檢查，請參閱 對 EC2 Windows 執行個體上的作業系統升級進行疑難排解。

升級後任務

1. 登入執行個體，初始化 .NET Framework 的升級，然後在提示後重新開機系統。
2. 如果您尚未在先前步驟中執行此操作，請安裝 EC2Launch v1 或 EC2Launch v2 代理程式。如需詳細資訊，請參閱 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v1 代理程式執行任務和 在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v2 代理程式執行任務。
3. 如果您升級至 Windows Server 2012 R2，我們建議您將 PV 驅動程式升級至 AWS PV 驅動程式。若您升級以 Nitro 為基礎的執行個體，我們建議您安裝或升級 NVME 和 ENA 驅動程式。如需詳細資訊，請參閱 NVMe 驅動程式 或 啟用 Windows 上的增強型網路 。
4. 重新啟用防毒軟體、反間諜軟體及防火牆。

使用自動化執行手冊來升級 EC2 Windows 執行個體

您可以使用 AWS Systems Manager Automation Runbook 在 上執行 AWS Windows 和 SQL Server 執行個體的自動升級。

目錄

- 相關服務
- 執行選項
- 升級 Windows Server
- 升級 SQL Server

相關服務

自動升級程序會使用下列 AWS 服務：

- AWS Systems Manager. AWS Systems Manager 是功能強大、統一的界面，可集中管理 AWS 資源。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」。
- AWS Systems Manager Agent (SSM Agent) 是可在 Amazon EC2 執行個體、內部部署伺服器或虛擬機器 (VM) 上安裝和設定的 Amazon 軟體。SSM Agent 讓 Systems Manager 能夠更新、管理和設定這些資源。此代理程式會處理 AWS Cloud 中來自 Systems Manager 服務的申請，然後依照要求中的指定執行。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的使用 SSM Agent。
- AWS Systems Manager SSM Runbook。SSM Runbook 定義 Systems Manager 在受管執行個體上執行的動作。SSM Runbook 使用 JavaScript 物件標記法 (JSON) 或 YAML，其中包括您指定的步驟和參數。本主題使用兩份 Systems Manager SSM Runbook 來進行自動化。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 AWS Systems Manager Automation runbook 參考。

執行選項

當您在 Systems Manager 主控台選取 Automation (自動化) 時，請選取 Execute (執行)。在您選取自動化文件後，系統會提示您選取自動化執行選項。請從以下選項中選擇。在本主題稍後所提供途徑的步驟中，我們使用 Simple execution (簡易執行) 選項。

簡易執行

如果您想要更新單一執行個體，但不想經歷每個自動化步驟來稽核結果，請選擇此選項。稍後的升級步驟中會進一步詳述此選項。

Rate control (比率控制)

如果您要對多個執行個體套用升級，請選擇此選項。請定義以下設定。

- Parameter (參數)

此設定 (也在 Multi-Account and Region (多帳戶和區域) 設定中) 定義自動化如何展開。

- 目標

  選取您要套用自動化的目標。此設定也在 Multi-Account and Region (多帳戶和區域) 設定中。

- Parameter Values (參數值)

  使用自動化文件參數中定義的值。

- Resource Group (資源群組)

  在 中 AWS，資源是您可以使用的實體。範例包括 Amazon EC2 執行個體、 AWS CloudFormation 堆疊或 Amazon S3 儲存貯體。如果您使用多個資源，將它們作為群組進行管理可能很有用，而不是為每個任務從一個 AWS 服務移至另一個服務。在某些情況下，您可能需要管理大量的相關資源，例如組成應用程式層的 EC2 執行個體。在此情況下，您在這些資源上可能需要一次執行大量動作。

- Tags (標籤)

  標籤可協助您以不同的方式分類 AWS 資源，例如依用途、擁有者或環境。此分類在您擁有許多相同類型的資源時很有用。您可以使用指派的標籤快速識別特定資源。

- Rate Control (比率控制)

  Rate Control (比率控制) 也在 Multi-Account and Region (多帳戶和區域) 設定中。當您設定比率控制參數時，請以目標計數或機群的百分比，定義機群中有多少要套用自動化。

Multi-Account and Region (多帳戶和區域)

除了 Rate Control (比率控制) 下指定的參數 (也用在 Multi-Account and Region (多帳戶和區域) 設定中)，另外還有兩個設定：

- Accounts and organizational units (OUs) (帳戶和組織單位 (OU))

  指定您要執行自動化的多個帳戶。

- AWS 區域

  指定您要執行自動化的多個 AWS 區域 位置。

手動執行

此選項類似於 Simple execution (簡易執行)，但可讓您逐步完成每個自動化步驟並稽核結果。

升級 Windows Server

AWSEC2-CloneInstanceAndUpgradeWindows Runbook 從您帳戶中的 Windows Server 執行個體建立 Amazon Machine Image (AMI)，並將此 AMI 升級到您選擇的支援版本。此多步驟程序需要最多兩個小時才會完成。

自動化升級程序中包含兩個 AMI：

- 目前執行中的執行個體。第一個 AMI 是目前執行中的執行個體 (尚未升級)。此 AMI 用來啟動另一個執行個體，以執行就地升級。當程序完成時，將從您的帳戶中刪除此 AMI，除非您明確要求保留原始執行個體。此設定由參數 KeepPreUpgradeImageBackUp 處理 (預設值為 false，表示預設會刪除 AMI)。
- 已升級的 AMI。此 AMI 是自動化程序的成果。

最終成果是一個 AMI，即已升級的 AMI 執行個體。

當升級完成時，您可以在 Amazon VPC 中啟動新的 AMI，以測試應用程式功能。測試後，在執行另一次升級前，請先安排應用程式停機時間，再完全切換至已升級的執行個體。

先決條件

若要使用 AWS Systems Manager 自動化文件自動化 Windows Server 升級，您必須執行下列任務：

- 搭配指定的 IAM 政策建立 IAM 角色，以允許 Systems Manager 在您的 Amazon EC2 執行個體上執行自動化任務，並驗證您是否符合使用 Systems Manager 的先決條件。如需詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的建立角色以將許可委派給 AWS 服務。
- 選取您希望如何執行自動化的選項。執行選項包括 Simple execution (簡易執行)、Rate control (比率控制)、Multi-account and Region (多帳戶和區域) 及 Manual execution (手動執行)。如需關於這些選項的詳細資訊，請參閱 執行選項。
- 確認 SSM Agent 安裝於您的執行個體上。如需詳細資訊，請參閱在適用於 Windows Server 的 Amazon EC2 執行個體上安裝與設定 SSM Agent。
- Windows PowerShell 3.0 或更新版本必須安裝在您的執行個體上。
- 對於加入 Microsoft Active Directory 網域的執行個體，建議您指定沒有連線到您的網域控制站的 SubnetId，以協助避免主機名稱衝突。
- 執行個體子網路必須具有網際網路的傳出連線，以便存取 AWS 服務 Amazon S3 和從 Microsoft 下載修補程式。如果子網路是公有子網路且執行個體具有公有 IP 地址，或子網路是私有子網路且具有將網際網路流量傳送至公有 NAT 裝置的路由，則符合此需求。

- 此自動化適用於執行 Windows Server 2008 R2、Windows Server 2012 R2、Windows Server 2016 和 Windows Server 2019 的執行個體。
- 確認執行個體在開機磁碟中有 20 GB 的可用磁碟空間。
- 如果執行個體未使用 提供的 Windows 授權 AWS，請指定包含 Windows Server 2012 R2 安裝媒體的 Amazon EBS 快照 ID。若要執行此作業：
  1. 確認 Amazon EC2 執行個體執行的是 Windows Server 2012 R2 或更新版本。
  2. 在執行個體執行的相同可用區域中建立 6 GB 的 Amazon EBS 磁碟區。將磁碟區連結到執行個體。例如，將其掛載為 D 磁碟機。
  3. 開啟 ISO 的內容（按一下滑鼠右鍵）選單，並將其掛載到執行個體，例如磁碟機 E。
  4. 從磁碟機 E:\ 將 ISO 的內容複製到磁碟機 D:\
  5. 從上述步驟 2 建立的 6 GB 磁碟區建立 Amazon EBS 快照。

Windows Server 升級限制

此自動化不支援升級 Windows 網域控制站、叢集或 Windows 桌面作業系統。此自動化也不支援安裝了下列角色的適用於 Windows Server 的 Amazon EC2 執行個體：

- 遠端桌面工作階段主機 (RDSH)
- 遠端桌面連線代理人 (RDCB)
- 遠端桌面虛擬化主機 (RDVH)
- 遠端桌面 Web 存取 (RDWA)

執行 Windows Server 自動升級的步驟

使用 AWSEC2-CloneInstanceAndUpgradeWindows Automation Runbook，按照以下步驟升級您的 Windows Server 執行個體。

1. 從 AWS 管理主控台中開啟 Systems Manager。
2. 從左側導覽窗格中，在 Change Management (變更管理) 下選擇 Automation (自動化)。
3. 選擇 Execute automation (執行自動化)。
4. 搜尋稱為 `AWSEC2-CloneInstanceAndUpgradeWindows` 的自動化文件。
5. 當文件名稱出現時，請選取它。選取此文件時，將會出現文件詳細資訊。
6. 選擇 Execute automation (執行自動化) 以輸入此文件的參數。在頁面頂端，保持選取 Simple execution (簡易執行)。

7. 根據下列指導，輸入所要求的參數。

   - `InstanceID`

     類型：字串

     (必要) 執行 Windows Server 2008 R2、2012 R2、2016 或 2019 的執行個體，並且已安裝 SSM Agent。

   - `InstanceProfile.`

     類型：字串

     (必要) IAM 執行個體描述檔。這是用於對 Amazon EC2 執行個體和 AWS AMIs 執行 Systems Manager 自動化的 IAM 角色。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的設定 EC2 執行個體許可。

   - `TargetWindowsVersion`

     類型：字串

     (必要) 選取目標 Windows 版本。

   - `SubnetId`

     類型：字串

     (必要) 這是升級程序的子網，也是您的 EC2 執行個體所在地方。確認子網路具有對 AWS 服務的傳出連線，包括 Amazon S3，以及對 Microsoft 的傳出連線 （以下載修補程式）。

   - `KeepPreUpgradedBackUp`

     類型：字串

     (選用) 如果此參數設為 `true`，自動化會保留從執行個體建立的映像。預設設定為 `false`。

   - `RebootInstanceBeforeTakingImage`

     類型：字串

     (選用) 預設為 `false` (不重新啟動)。如果此參數設為 `true`，Systems Manager 在為升級建立 AMI 之前會重新啟動執行個體。

8. 輸入參數後，選擇 Execute (執行)。自動化開始時，您可以監控執行進度。

9. 當自動化完成時，您會看到 AMI ID。您可以啟動 AMI，以驗證 Windows 作業系統已升級。

> **Note**
>
> 自動化沒有必要執行所有步驟。這些步驟取決於自動化和執行個體的行為。Systems Manager 可能略過有些非必要的步驟。
> 此外，某些步驟可能會逾時。Systems Manager 會嘗試升級和安裝所有最新的修補程式。不過，根據特定步驟可定義的逾時設定，修補程式有時會逾時。發生此情況時，Systems Manager 自動化會繼續下一個步驟，以確保內部作業系統升級到目標 Windows Server 版本。

10. 自動化完成之後，您可以使用 AMI ID 啟動 Amazon EC2 執行個體，以檢閱您的升級。如需如何從 AWS AMI 建立 Amazon EC2 執行個體的詳細資訊，請參閱如何從自訂 AMI 啟動 EC2 執行個體？

升級 SQL Server

AWSEC2-CloneInstanceAndUpgradeSQLServer 指令碼會從您帳戶中執行 SQL Server 的 Amazon EC2 執行個體建立 AMI，然後將 AMI 升級到更新版本的 SQL Server。此多步驟程序需要最多兩個小時才會完成。

自動化會從執行個體建立 AMI，接著在您提供的子網中啟動新的 AMI。自動化接著會執行 SQL Server 的就地升級。升級完成之後，自動化在終止已升級的執行個體之前會建立新的 AMI。

自動化升級程序中包含兩個 AMI：

- 目前執行中的執行個體。第一個 AMI 是目前執行中的執行個體 (尚未升級)。此 AMI 用來啟動另一個執行個體，以執行就地升級。當程序完成時，將從您的帳戶中刪除此 AMI，除非您明確要求保留原始執行個體。此設定由參數 `KeepPreUpgradeImageBackUp` 處理 (預設值為 `false`，表示預設會刪除 AMI)。
- 已升級的 AMI。此 AMI 是自動化程序的成果。

最終成果是一個 AMI，即已升級的 AMI 執行個體。

當升級完成時，您可以在 Amazon VPC 中啟動新的 AMI，以測試應用程式功能。測試後，在執行另一次升級前，請先安排應用程式停機時間，再完全切換至已升級的執行個體。

先決條件

若要使用 AWS Systems Manager 自動化文件自動化 SQL Server 升級，您必須執行下列任務：

- 搭配指定的 IAM 政策建立 IAM 角色，以允許 Systems Manager 在您的 Amazon EC2 執行個體上執行自動化任務，並驗證您是否符合使用 Systems Manager 的先決條件。如需詳細資訊，請參閱「AWS Identity and Access Management 使用者指南」中的建立角色以委派許可給 AWS 服務。
- 選取您希望如何執行自動化的選項。執行選項包括 Simple execution (簡易執行)、Rate control (比率控制)、Multi-account and Region (多帳戶和區域) 及 Manual execution (手動執行)。如需關於這些選項的詳細資訊，請參閱 執行選項。
- Amazon EC2 執行個體必須使用 Windows Server 2008 R2 (或更新版本) 和 SQL Server 2008 或更新版本。
- 確認 SSM Agent 安裝於您的執行個體上。如需詳細資訊，請參閱在適用於 Windows Server 的 Amazon EC2 執行個體上使用 SSM Agent。
- 確認執行個體具有足夠的可用磁碟空間：
  - 若您要從 Windows Server 2008 R2 升級至 2012 R2，或從 Windows Server 2012 R2 升級至更新版本的作業系統，請確認您的執行個體開機磁碟中有 20 GB 的可用磁碟空間。
  - 若您要從 Windows Server 2008 R2 升級至 2016 或更新版本，請確認執行個體開機磁碟中有 40 GB 的可用磁碟空間。
- 針對使用自有授權 (BYOL) 的 SQL Server 版本執行個體，適用下列額外的事前準備：
  - 提供 Amazon EBS 快照 ID，其中包含目標 SQL Server 安裝媒體。若要執行此作業：
    1. 確認 Amazon EC2 執行個體執行的是 Windows Server 2008 R2 或更新版本。
    2. 在執行個體執行的相同可用區域中建立 6 GB 的 Amazon EBS 磁碟區。將磁碟區連結到執行個體。例如，將其掛載為 D 磁碟機。
    3. 開啟 ISO 的內容（按一下滑鼠右鍵）選單，並將其掛載到執行個體，例如磁碟機 E。
    4. 從磁碟機 E:\ 將 ISO 的內容複製到磁碟機 D:\
    5. 建立步驟 2 中所建立 6 GB 磁碟區的 Amazon EBS 快照。

SQL Server 自動升級限制

使用 AWSEC2-CloneInstanceAndUpgradeSQLServer Runbook 來執行自動升級時，如下限制適用：

- 升級僅能在使用 Windows 身分驗證的 SQL Server 上執行。
- 確認執行個體上沒有待定的安全性修補程式更新。開啟 Control Panel (控制面板)，接著選擇 Check for updates (檢查更新)。
- 不支援在 HA 和鏡像模式中的 SQL Server 部署。

執行 SQL Server 自動升級的步驟

使用 AWSEC2-CloneInstanceAndUpgradeSQLServer Automation Runbook，按照以下步驟升級您的 SQL Server。

1. 下載 SQL Server 2016 .iso 檔案並掛載到來源伺服器 (如果您尚未這麼做)。
2. 掛載 .iso 檔案之後，將所有元件檔案複製到您選擇的任何磁碟區。
3. 建立磁碟區的 Amazon EBS 快照，並將快照 ID 複製到剪貼簿，供稍後使用。如需詳細資訊，請參閱 Amazon EBS 使用者指南中的建立 Amazon EBS 快照。
4. 將執行個體描述檔連接至 Amazon EC2 來源執行個體。這可讓 Systems Manager 與 EC2 執行個體通訊，並在新增至 AWS Systems Manager 服務之後對其執行命令。在此範例中，我們將角色命名為 `SSM-EC2-Profile-Role`，並將 `AmazonSSMManagedInstanceCore` 政策連接到角色。
5. 在 AWS Systems Manager 主控台的左側導覽窗格中，選擇受管執行個體。確認 EC2 執行個體在受管執行個體的清單中。如果幾分鐘後沒有看見您的執行個體，請參閱 AWS Systems Manager 使用者指南中的我的執行個體在哪裡？。
6. 在左側導覽窗格中，在 Change Management (變更管理) 下選擇 Automation (自動化)。
7. 選擇 Execute automation (執行自動化)。
8. 搜尋稱為 `AWSEC2-CloneInstanceAndUpgradeSQLServer` 的自動化文件。
9. 選擇 `AWSEC2-CloneInstanceAndUpgradeSQLServer` SSM 文件，然後選擇 Next (下一步)。
10. 確保選取 Simple execution (簡易執行) 選項。
11. 根據下列指導，輸入所要求的參數。

    - `InstanceId`

      類型：字串

      (必要) 執行 SQL Server 2008 R2 (或更新版本) 的執行個體。

    - `IamInstanceProfile`

      類型：字串

      (必要) IAM 執行個體描述檔。

    - `SQLServerSnapshotId`

      類型：字串

(必要) 目標 SQL Server 安裝媒體的快照 ID。包含 SQL Server 授權的執行個體不需要此參數。

- SubnetId

  類型：字串

  (必要) 這是升級程序的子網，也是您的 EC2 執行個體所在地方。確認子網路具有對 AWS 服務的傳出連線，包括 Amazon S3，以及對 Microsoft 的傳出連線 （以下載修補程式）。

- KeepPreUpgradedBackUp

  類型：字串

  (選用) 如果此參數設為 true，自動化會保留從執行個體建立的映像。預設設定為 false。

- RebootInstanceBeforeTakingImage

  類型：字串

  (選用) 預設為 false (不重新啟動)。如果此參數設為 true，Systems Manager 在為升級建立 AMI 之前會重新啟動執行個體。

- TargetSQLVersion

  類型：字串

  (選用) 目標 SQL Server 版本。預設值為 2016。

12. 輸入參數後，選擇 Execute (執行)。自動化開始時，您可以監控執行進度。
13. 當 Execution Status (執行狀態) 顯示 Success (成功) 時，請展開 Outputs (輸出) 來檢視 AMI 資訊。針對您選擇的 VPC，您可以使用 AMI ID 來啟動您的 SQL Server 執行個體。
14. 開啟 Amazon EC2 主控台。在左側導覽窗格中選擇 AMI。您應該會看到新的 AMI。
15. 若要確認已成功安裝 SQL Server 新版本，請選擇新的 AMI 並選擇 Launch (啟動)。
16. 選擇您要用於 AMI 的執行個體類型、您要部署到的 VPC 和子網，以及您要使用的儲存體。因為您是從 AMI 啟動新的執行個體，將會出現磁碟區，供您選擇加入您啟動的新 EC2 執行個體中。您可以移除任何這些磁碟區，或新增磁碟區。
17. 新增標籤以協助識別您的執行個體。
18. 將一或多個安全群組新增至執行個體。
19. 選擇 Launch Instance (啟動執行個體)。
20. 選取執行個體的標籤名稱，然後在 Actions (動作) 下拉式清單下，選取 Connect (連接)。
21. 驗證 SQL Server 新版本是新執行個體上的新資料庫引擎。

將 EC2 Windows 執行個體遷移至 Nitro 型執行個體類型

AWS Windows AMIs 是以 Microsoft 安裝媒體使用的預設設定進行設定，並具有一些自訂。自訂內容會包含支援 Nitro 型執行個體的驅動程式和組態，例如 M5 和 C5。

在從 Xen 型執行個體遷移至 Nitro 型執行個體 (包括裸機執行個體) 時，我們建議您在下列情況下遵循本主題中的步驟：

- 如果您是從自訂的 Windows AMI 啟動執行個體
- 如果您正從 Amazon 在 2018 年 8 月之前建立並提供的 Windows AMI 啟動執行個體

或者，您可以使用 `AWSSupport-UpgradeWindowsAWSDrivers` 自動化文件，以自動執行第 1 部分、第 2 部分及第 3 部分中所述的程序。如果您選擇使用自動程序，請參閱 （替代） 使用 升級 AWS PV、EMA 和 NVMe 驅動程式 AWS Systems Manager，然後繼續進行第 4 部分及第 5 部分。

如需詳細資訊，請參閱 Amazon EC2 Update – Additional Instance Types, Nitro System, and CPU Options。

> **Note**
>
> 下列遷移程序可以在 Windows Server 2016 版和更新版本上執行。先前已達生命週期結束的作業系統版本不會接受測試，且可能無法與最新的執行個體類型相容。
> 若要遷移 Linux 執行個體，請參閱 the section called “執行個體類型變更”。

內容

- 第 1 部分：安裝和升級 AWS PV 驅動程式
- 第 2 部分：安裝和升級 ENA
- 第 3 部分：Update AWS NVMe 驅動程式
- 第 4 部分：更新 EC2Config 與 EC2Launch
- 第 5 部分：為 Bare Metal 執行個體安裝序列埠驅動程式
- 第 6 部分：更新電源管理設定
- 第 7 部分：更新新執行個體類型的 Intel 晶片組驅動程式
- （替代） 使用 升級 AWS PV、EMA 和 NVMe 驅動程式 AWS Systems Manager

開始之前

此程序假繪設您有一個 Xen 型執行個體 (例如 M4 或 C4)，且正在遷移至 Nitro 型執行個體。

您必須使用 PowerShell 第 3.0 版或更新版本，才能順利執行升級。

> **Note**
>
> 遷移時，現有網路介面卡上的靜態 IP 或自訂 DNS 網路設定可能會遺失，因為執行個體預設使用新的增強型聯網轉接器裝置。

在遵循此程序中的步驟之前，我們建議您建立該執行個體的備份。從 EC2 主控台中，選擇需要遷移的執行個體、開啟內容 (按一下右鍵) 功能表、選擇 Instance State (執行個體狀態)，然後選擇 Stop (停止)。

> **Warning**
>
> 停止執行個體時，在任何執行個體存放區磁碟區的資料都會清除。若要保留執行個體存放區磁碟區上的任何資料，請確定您將資料備份至持久性儲存。

在 EC2 console (EC2 主控台) 中開啟執行個體的內容 (按一下右鍵) 功能表、選擇 Image (映像)，然後選擇 Create Image (建立映像)。

> **Note**
>
> 在遷移或變更執行個體類型後，即可完成此說明的第 4 與第 5 部分。但是，如果您要遷移到裸機執行個體類型，我們建議在遷移開始前先行完成相關設定。

第 1 部分：安裝和升級 AWS PV 驅動程式

雖然 Nitro 系統不會使用 AWS PV 驅動程式，但如果您使用舊版的 Citrix PV 或 AWS PV，您仍應升級它們。最新的 AWS PV 驅動程式會解決之前版本驅動程式中的錯誤，這些錯誤可能會在您使用 Nitro 系統，或者需要遷移回 Xen 執行個體時出現。最佳實務是，建議您一律更新為 Windows 執行個體的最新驅動程式 AWS。

使用下列程序執行 AWS PV 驅動程式就地升級，或從 Citrix PV 驅動程式升級至 Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2、Windows Server 2016 或 Windows Server

2019 上的 AWS PV 驅動程式。如需詳細資訊，請參閱升級 EC2 Windows 執行個體上的 PV 驅動程式。

若要升級網域控制站，請參閱升級網域控制器 (AWS PV 升級) 。

執行將 或 升級至 AWS PV 驅動程式

1. 使用遠端桌面連線到執行個體並為執行個體準備升級。在執行升級前將所有非系統磁碟離線。如果您正在執行 AWS PV 驅動程式的就地更新，則不需要此步驟。在服務主控台中將非必要的服務設為手動啟動。
2. 將最新的驅動程式套件下載到執行個體。
3. 解壓縮資料夾的內容，然後執行 `AWSPVDriverSetup.msi`。

在執行 MSI 之後，執行個體會自動重新開機並升級驅動程式。執行個體最多可能會在 15 分鐘內無法使用。

在升級完成且執行個體在 Amazon EC2 主控台中通過兩項運作狀態檢查後，使用遠端桌面連線到執行個體，確認已安裝新的驅動程式。在裝置管理器中，在儲存體控制器下找到 AWS PV 儲存體主機轉接器。確認驅動程式版本與驅動程式版本歷史記錄表格中列出的最新版本相同。如需詳細資訊，請參閱AWS PV 驅動程式套件歷史記錄。

第 2 部分：安裝和升級 ENA

升級到最新的彈性網路轉接器驅動程式，以確保支援所有網路功能。如果您啟動執行個體，且其尚未啟用增強型聯網，您必須在執行個體上下載並安裝必要的網路轉接器驅動程式。然後，設定 enaSupport 執行個體屬性，以啟用增強型聯網。您只能在支援的執行個體類型上啟用此屬性，且需先安裝 ENA 驅動程式。如需詳細資訊，請參閱在 EC2 執行個體上使用 ENA 啟用增強型網路。

1. 將最新的驅動程式下載到執行個體。如果您需要舊版的驅動程式，請參閱 ENA Windows 驅動程式版本歷史記錄。
2. 將 zip 封存檔解壓縮。
3. 從擷取的資料夾執行 `install.ps1` PowerShell script 以安裝驅動程式。

> **Note**
>
> 請以管理員身分執行 `install.ps1` 指令碼，以避免發生安裝錯誤。

4. 請檢查您的 AMI 是否已啟動 enaSupport。若否，請遵循在 EC2 執行個體上使用 ENA 啟用增強型網路中的文件繼續進行。

第 3 部分：Update AWS NVMe 驅動程式

AWS NVMe 驅動程式用於與在 Nitro 系統中公開為 NVMe 區塊裝置的 Amazon EBS 和 SSD 執行個體存放區磁碟區互動，以獲得更好的效能。

> **Important**
>
> 當您在 Xen 型執行個體上安裝或升級 AWS NVMe 時，會特別修改下列指示，目的是將執行個體遷移至 Nitro 型執行個體。

1. 將最新的驅動程式套件下載到執行個體。

   如果您需要舊版的驅動程式，請參閱 AWS NVMe Windows 驅動程式版本 以了解支援的版本。
2. 將 zip 封存檔解壓縮。
3. 按照 `Readme.txt` 中的說明來安裝驅動程式。
4. 開啟 PowerShell 工作階段，並執行下列命令：

   ```
   PS C:\> start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
   ```

   > **Note**
   >
   > 若要套用命令，必須以系統管理員身分執行 PowerShell 工作階段。PowerShell (x86) 版本會導致錯誤。
   > 此命令僅在裝置驅動程式上執行 sysprep。它不會執行完整的 sysprep 準備。
5. Windows Server 2008 R2 和 Windows Server 2012 請關閉執行個體，變更並開始啟動執行個體，然後繼續進行第 4 部分。如果您在遷移至 Nitro 型執行個體類型之前，先在 Xen 型執行個體類型上再次啟動執行個體，則其將不會開機。對於其他支援的 Windows AMI，您可以在裝置 sysprep 之後隨時變更執行個體類型。

第 4 部分：更新 EC2Config 與 EC2Launch

對 Windows 執行個體而言，當最新的 EC2Config 與 EC2Launch 公用程式在 Nitro 系統 (包括 EC2 Bare Metal) 上執行時，其將提供其他功能與資訊。根據預設，EC2Config 包含在 Windows Server 2016 之前版本的 AMI 內。EC2Launch 將取代 Windows Server 2016 和更新版本 AMI 上的 EC2Config。

更新 EC2Config 與 EC2Launch 服務時， AWS 的新 Windows AMI 將包含最新版的服務。不過，您必須更新自己的 Windows AMI 和執行個體，獲得最新版的 EC2Config 與 EC2Launch。

安裝或更新 EC2Config

1. 下載並解壓縮 EC2Config 安裝程式。
2. 執行 `EC2Install.exe`。如需完整的選項清單，請執行 `EC2Install` 並加入 `/?` 選項。根據預設，設定程式會顯示提示。若要執行命令且不加入提示，請使用 `/quiet` 選項。

如需詳細資訊，請參閱安裝最新版本的 EC2Config。

安裝或更新 EC2Launch

1. 如果您已在執行個體上安裝和設定 EC2Launch，請備份 EC2Launch 組態檔案。安裝程序不會保留此檔案的變更內容。根據預設，檔案位於 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。
2. 下載 EC2-Windows-Launch.zip 到執行個體上的目錄。
3. 下載 install.ps1 到您下載 `EC2-Windows-Launch.zip` 的同一個目錄。
4. 執行 `install.ps1`。

   > **Note**
   >
   > 請以管理員身分執行 `install.ps1` 指令碼，以避免發生安裝錯誤。

5. 如果您備份了 EC2Launch 組態檔案，請將其複製到 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目錄。

如需詳細資訊，請參閱在 EC2 Windows 執行個體啟動期間，使用 EC2Launch v1 代理程式執行任務。

第 5 部分：為 Bare Metal 執行個體安裝序列埠驅動程式

`i3.metal` 執行個體類型會使用 PCI 型序列裝置，而非 I/O 連接埠型序列裝置。最新的 Windows AMI 會自動使用 PCI 型序列裝置並安裝好序列埠驅動程式。如果您不是使用 Amazon 於 2018 年 4 月 11 日或之後發布的 Windows AMI 所啟動的執行個體，您必須安裝序列埠驅動程式，為 EC2 功能 (例如密碼產生或主控台輸出) 啟用序列裝置。最新的 EC2Config 與 EC2Launch 公用程式也支援 i3.metal 並提供其他功能。若您尚未這樣做，請遵循第 4 部分中的步驟。

安裝序列埠驅動程式

1. 將序列驅動程式套件下載到執行個體。
2. 擷取資料夾內容、針對 `aws_ser.INF` 開啟內容 (按一下右鍵) 功能表，然後選擇 install (安裝)。
3. 選擇 Okay (確定)。

第 6 部分：更新電源管理設定

電源管理設定的下列更新會設定永不關閉顯示器，讓 Nitro 系統上的作業系統能正常關機。Amazon 提供的所有 Windows AMI，自 2018 年 11 月 28 日起皆有此預設組態。

1. 開啟命令提示或 PowerShell 工作階段。
2. 執行下列命令：

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

第 7 部分：更新新執行個體類型的 Intel 晶片組驅動程式

`u-6tb1.metal`、`u-9tb1.metal` 和 `u-12tb1.metal` 執行個體類型必須使用晶片組驅動程式先前未安裝在 Windows AMI 上的硬體。如果您不是使用從 Amazon 於 2018 年 11 月 19 日或之後提供之 Windows AMI 啟動的執行個體，您必須使用 Intel 晶片組 INF 公用程式安裝驅動程式。

安裝晶片組驅動程式

1. 執行個體的晶片組 INF 公用程式。

2. 將檔案解壓縮。
3. 執行 `SetupChipset.exe`。
4. 接受 Intel 軟體授權合約，並安裝晶片組驅動程式。
5. 重新啟動執行個體。

（替代）使用 升級 AWS PV、EMA 和 NVMe 驅動程式 AWS Systems Manager

`AWSSupport-UpgradeWindowsAWSDrivers` 自動化文件可以自動執行第 1 部分、第 2 部分及第 3 部分中所述的步驟。此方法也可以修復驅動程式升級失敗的執行個體。

`AWSSupport-UpgradeWindowsAWSDrivers` 自動化文件會升級或修復指定 EC2 執行個體上的儲存和網路 AWS 驅動程式。文件會呼叫 AWS Systems Manager 代理程式 (SSM 代理程式），嘗試線上安裝最新版本的 AWS 驅動程式。如果 SSM Agent 無法聯絡，如果明確請求，文件可以執行 AWS 驅動程式的離線安裝。

> **Note**
>
> 此程序將無法在網域控制器上進行。若要更新網域控制器上的驅動程式，請參閱升級網域控制器 (AWS PV 升級）。

使用 自動升級 AWS PV、EMA 和 NVMe 驅動程式 AWS Systems Manager

1. 開啟 https://console.aws.amazon.com/systems-manager 中的 Systems Manager 主控台。
2. 選擇 Automation (自動化)、Execute Automation (執行自動化)。
3. 搜尋然後選取 AWSSupport-UpgradeWindowsAWSDrivers 自動化文件，接著請選擇執行自動化。
4. 在輸入參數區段中，設定以下選項：

   執行個體 ID

   輸入要升級之執行個體的唯一 ID。

   AllowOffline

   (選用) 選擇下列其中一個選項：

   - `True` – 選擇此選項來執行離線安裝。升級過程中執行個體會停止並重新啟動。

> **Warning**
>
> 停止執行個體時，在任何執行個體存放區磁碟區的資料都會清除。若要保留執行個體存放區磁碟區上的任何資料，請確定您將資料備份至持久性儲存。

- `False` – (預設值) 若要執行線上安裝，請將此選項保持選取狀態。升級過程中執行個體會重新啟動。

> **Important**
>
> 線上和離線升級會在嘗試升級作業之前建立 AMI。在自動化完成之後，AMI 會持續存在。保護您對 AMI 的存取，或刪除它 (若不再需要它的話)。

SubnetId

(選用) 輸入下列其中一個值：

- `SelectedInstanceSubnet` – (預設值)升級程序會將 helper 執行個體啟動至與要升級之執行個體相同的子網路。子網必須允許與 Systems Manager 端點 (`ssm.*`) 進行通訊。
- `CreateNewVPC` – 升級程序會將 helper 執行個體啟動至新的 VPC。如果您不確定目標執行個體的子網是否允許與 `ssm.*` 端點進行通訊，請使用此選項。您的使用者必須擁有建立 VPC 的許可。
- 特定子網路 ID – 指定要將 helper 執行個體啟動至其中之特定子網路的 ID。子網必須與要升級的執行個體位於相同的可用區域，而且它必須允許與 `ssm.*` 端點進行通訊。

5. 選擇 Execute (執行)。
6. 允許升級以完成。完成線上升級最多可能需要 10 分鐘，而完成離線升級最需要多 25 分鐘。

對 EC2 Windows 執行個體上的作業系統升級進行疑難排解

AWS 提供升級協助程式服務問題的升級支援，此 AWS 公用程式可協助您執行涉及 Citrix PV 驅動程式的就地升級。

在升級之後，當 .NET 執行時間最佳化服務最佳化 .NET Framework 時，執行個體的平均 CPU 用量可能會暫時增加。這是預期的行為。

若執行個體在數小時之後仍無法通過所有狀態檢查，請檢查下列內容。

- 若您升級到 Windows Server 2008，並且所有狀態檢查在數小時之後仍然失敗，則升級可能已失敗，並會顯示提示，讓您按一下確認以確認還原。因為無法在此狀態存取主控台，沒有任何方式能按一下按鈕。若要解決此問題，請透過 Amazon EC2 主控台或 API 執行重新啟動。初始化重新開機可能需要十分鐘以上。執行個體可能會在 25 分鐘可用。
- 從伺服器移除應用程式或伺服器角色並再試一次。

若執行個體在從伺服器移除應用程式或伺服器角色後仍然無法通過所有狀態檢查，請執行下列作業。

- 停止執行個體，並將根磁碟區連接到另一個執行個體。如需詳細資訊，請參閱等待中繼資料服務中如何停止並將根磁碟區連接到另一個執行個體的說明。
- 分析 Windows 安裝日誌檔案及事件日誌，查看故障。

針對其他作業系統升級或遷移的問題，我們建議檢閱在您開始就地升級之前中列出的文章。

教學課程：將 Amazon EC2 執行個體連線至 Amazon RDS 資料庫

教學課程目標

本教學課程的目的是了解如何使用 AWS 管理主控台來設定 Amazon EC2 執行個體和 Amazon RDS 資料庫之間的連線。

設定連線有不同的選項。在本教學課程中，我們將探討以下三個選項：

- 選項 1：使用 EC2 主控台自動將執行個體連線至 RDS 資料庫

  使用 EC2 主控台中的自動連線功能，自動設定 EC2 執行個體和 RDS 資料庫之間的連線，以允許 EC2 執行個體和 RDS 資料庫之間的流量。
- 選項 2：使用 RDS 主控台自動將執行個體連線至 RDS 資料庫

  使用 RDS 主控台中的自動連線功能，自動設定 EC2 執行個體和 RDS 資料庫之間的連線，以允許 EC2 執行個體和 RDS 資料庫之間的流量。
- 選項 3：透過建立安全群組，以手動方式將執行個體連線至 RDS 資料庫

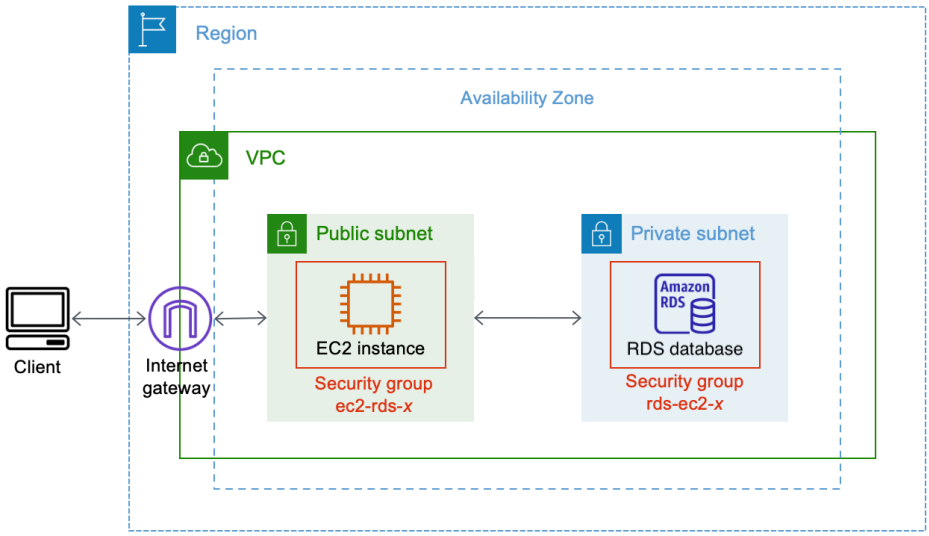
  透過手動設定和指派安全群組來複製選項 1 和選項 2 中自動連線功能自動建立的組態，以設定 EC2 執行個體與 RDS 資料庫之間的連線。

Context

至於為什麼要在 EC2 執行個體和 RDS 資料庫之間設定連線，讓我們一起思考下面這種情況：您的網站向使用者提供表單以供填寫。您需要在資料庫中擷取表單資料。您可以在已被設定為 Web 伺服器的 EC2 執行個體上託管您的網站，並且可以在 RDS 資料庫中擷取表單資料。EC2 執行個體和 RDS 資料庫必須彼此相連，以便表單資料可以從 EC2 執行個體傳送到 RDS 資料庫。本教學課程會說明如何設定該連線。請注意，這只是連線 EC2 執行個體和 RDS 資料庫的其中一個使用案例範例。

Architecture

下圖展示了所建立的資源，以及完成本教學課程中的所有步驟後產生的架構組態。



此圖展示了您將建立的下列資源：

- 您將在相同的 AWS 區域 VPC 和可用區域中建立 EC2 執行個體和 RDS 資料庫。
- 您將在公有子網路中建立 EC2 執行個體。
- 您將在私有子網路中建立 RDS 資料庫。

  當您使用 RDS 主控台建立 RDS 資料庫並自動連線 EC2 執行個體時，會自動選取資料庫的 VPC、資料庫子網路群組和公開存取設定。在與 EC2 執行個體相同之 VPC 內的私有子網路中自動建立 RDS 資料庫。
- 網際網路使用者可以透過網際網路閘道使用 SSH 或 HTTP/HTTPS 連線到 EC2 執行個體。
- 網際網路使用者無法直接連線至 RDS 資料庫；只有 EC2 執行個體可連線至 RDS 資料庫。
- 當您使用自動連線功能允許 EC2 執行個體和 RDS 資料庫之間的流量時，會自動建立並新增下列安全群組：

- 建立安全群組 ec2-rds-*x* 並新增至 EC2 執行個體。它有一個傳出規則，該規則引用 rds-ec2-*x* 安全群組作為其目的地。這允許來自 EC2 執行個體的流量透過 rds-ec2-*x* 安全群組到達 RDS 資料庫。
- 建立安全群組 rds-ec2-*x* 並新增至 RDS 資料庫。它有一個傳入規則，該規則引用 ec2-rds-*x* 安全群組作為其來源。這允許來自 EC2 執行個體的流量透過 ec2-rds-*x* 安全群組到達 RDS 資料庫。

透過使用單獨的安全群組 (一個用於 EC2 執行個體，一個用於 RDS 資料庫)，您可以更好地控制執行個體和資料庫的安全性。如果您要在執行個體和資料庫上使用相同的安全群組，然後修改安全群組以適應資料庫，則修改會同時影響執行個體和資料庫。換句話說，如果您使用一個安全群組，您可能會無意中修改資源 (執行個體或資料庫) 的安全性，因為您忘記了安全群組已連接至該資源。

自動建立的安全群組也會遵守最低權限，因為其只允許透過建立特定於工作負載的安全群組對，在資料庫連接埠上進行此工作負載的相互連線。

考量事項

完成本教學課程中的任務後，請考量以下內容：

- 兩個主控台 - 在本教學課程中，您將使用下列兩個主控台：
  - Amazon EC2 主控台 – 您將使用 EC2 主控台啟動執行個體、自動將 EC2 執行個體連線到 RDS 資料庫，以及使用手動選項，透過建立安全群組來設定連線。
  - Amazon RDS 主控台 – 您將使用 RDS 主控台建立 RDS 資料庫，並自動將 EC2 執行個體連線到 RDS 資料庫。
- 一個 VPC - 若要使用自動連線功能，EC2 執行個體和 RDS 資料庫必須位於相同的 VPC 中。

  如果您要手動設定 EC2 執行個體和 RDS 資料庫之間的連線，則可以在一個 VPC 中啟動 EC2 執行個體，並在另一個 VPC 中啟動 RDS 資料庫；不過，您需要設定其他路由和 VPC 組態。這是本教學課程中涵蓋的情況。
- 一 AWS 區域：EC2 執行個體和 RDS 資料庫必須位於相同的 區域。
- 兩個安全群組 - EC2 執行個體與 RDS 資料庫之間的連線由兩個安全群組設定 - 針對 EC2 執行個體的安全群組，以及針對 RDS 資料庫的安全群組。

  當您使用 EC2 主控台或 RDS 主控台中的自動連線功能來設定連線 (本教學課程的選項 1 和選項 2) 時，會自動建立安全群組並指派給 EC2 執行個體和 RDS 資料庫。

  如果您不使用自動連線功能，則需要手動建立並指派安全群組。您可以在本教學課程的選項 3 中執行此操作。

教學課程所需時間

30 分鐘

您可以一次性完成整個教學課程，也可以一次完成一個任務。

成本

完成本教學課程後，您建立 AWS 的資源可能會產生費用。

您可以在免費方案下使用 Amazon EC2，前提是 AWS 您的帳戶符合免費方案的資格，而且您可以根據免費方案要求設定資源。如需詳細資訊，2025 年 7 月 15 日之前和之後的免費方案優惠。

如果您的 EC2 執行個體和 RDS 資料庫位於不同的可用區域，則需支付資料傳輸費用。若要避免產生這些費用，EC2 執行個體和 RDS 資料庫必須位於相同的可用區域。如需有關資料傳輸費用的資訊，請參閱「Amazon EC2 隨需定價」頁面中的資料傳輸。

若要避免在完成教學課程後產生費用，請務必刪除不再需要的資源。如需刪除資源的步驟，請參閱 任務 4 (選用)：清除。

選項 1：使用 EC2 主控台自動將執行個體連線至 RDS 資料庫

選項 1 的目的是探索 EC2 主控台中的自動連線功能，其可自動設定 EC2 執行個體和 RDS 資料庫之間的連線，以允許從 EC2 執行個體到 RDS 資料庫的流量。在選項 3 中，您將了解如何手動設定連線。

任務

- 開始之前
- 任務 1 (選用)：建立 RDS 資料庫
- 任務 2 (選用)：啟動 EC2 執行個體
- 任務 3：自動將 EC2 執行個體連線到 RDS 資料庫
- 任務 4：驗證連線組態
- 任務 5 (選用)：清除

開始之前

完成此教學課程需要以下項目：

- 與 EC2 執行個體處於相同 VPC 的 RDS 資料庫。您可以使用現有的 RDS 資料庫或遵循任務 1 中的步驟來建立新的 RDS 資料庫。
- 與 RDS 資料庫處於相同 VPC 的 EC2 執行個體。您可以使用現有的 EC2 執行個體或遵循任務 2 中的步驟來建立新的 EC2 執行個體。
- 呼叫以下操作的許可：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

任務 1 (選用)：建立 RDS 資料庫

> **Note**
>
> 建立 Amazon RDS 資料庫並不是本教學課程的重點。如果您已有 RDS 資料庫，並且您想在本教學課程中使用該資料庫，則可跳過此任務。
> 如果您使用現有的 RDS 資料庫，請確保其與 EC2 執行個體位於相同的 VPC 中，以便您可以使用自動連線功能。

此任務的目標是建立 RDS 資料庫，以便您可以完成任務 3，您可以在其中設定 EC2 執行個體和 RDS 資料庫之間的連線。此任務中的步驟會設定 RDS 資料庫，如下所示：

- 引擎類型：MySQL
- 範本：免費方案
- 資料庫執行個體識別符：**`tutorial-database-1`**

- 資料庫執行個體類別：db.t3.micro

> **Important**
>
> 在生產環境中，您應設定資料庫以滿足您的特定需求。

若要建立 MySQL RDS 資料庫

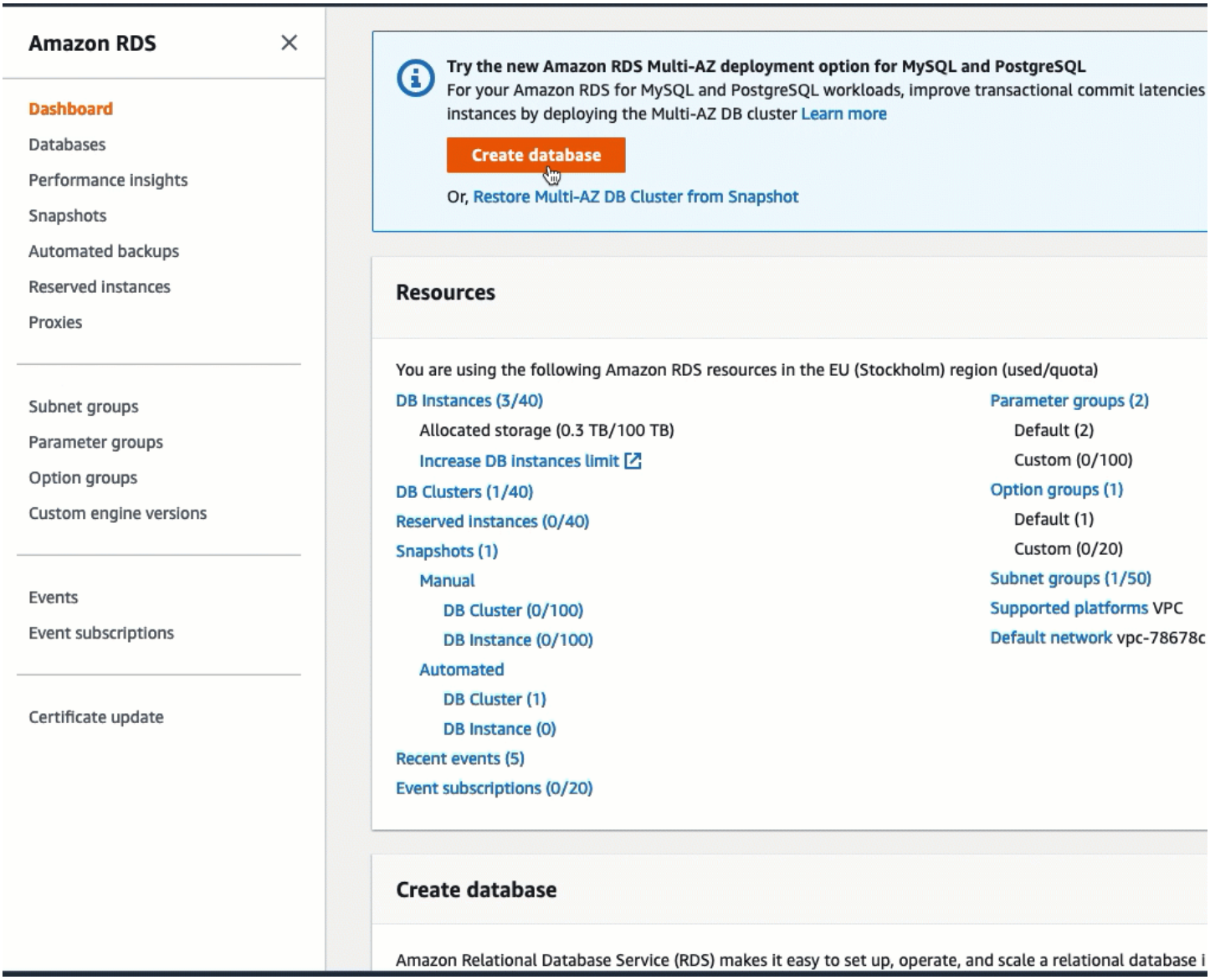
1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 從區域選取器 (位於右上角) 中選取 AWS 區域。資料庫和 EC2 執行個體必須位於相同區域，才能使用 EC2 主控台中的自動連線功能。
3. 在儀表板中，選擇 Create database (建立資料庫)。
4. 在 Choose a database creation method (選取資料庫建立方法) 中，確認選取 Standard Create (標準建立)。如果您選取 Easy create (輕鬆建立)，則無法使用 VPC 選取器。您必須確保您的資料庫與 EC2 執行個體位於相同的 VPC 中，才能使用 EC2 主控台中的自動連線功能。
5. 在 Engine options (引擎選項) 中，針對 Engine type (引擎類型) 選擇 MySQL。
6. 在 Templates (範本) 中，選擇符合您需求的範例範本。在本教學課程中，選擇 Free tier (免費方案) 免費建立 RDS 資料庫。不過，請注意，免費方案僅適用於符合免費方案資格的帳戶。在 Free tier (免費方案) 方塊中選擇 Info (資訊) 連結，可閱讀更多資訊。
7. 在 Settings (設定) 下，執行下列動作：
   a. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入資料庫的名稱。針對本教學，輸入 **tutorial-database-1**。
   b. 對於 Master username (主要使用者名稱)，請保留預設名稱，即 **admin**。
   c. 對於 Master password (主要密碼)，請輸入您在本教學課程中可以記住的密碼，然後在 Confirm password (確認密碼) 中再次輸入密碼。
8. 在執行個體組態下，針對資料庫執行個體類別，請保留預設內容，即 db.t3.micro。如果您的帳戶符合免費方案資格，則可免費使用此資料庫類別。如需詳細資訊，請參閱 AWS 免費方案。
9. 在 Connectivity (連線) 中，對於 Compute resource (運算資源)，選擇 Don't connect to an EC2 compute resource (不連線至 EC2 運算資源)，因為稍後會在任務 3 中連線 EC2 執行個體和 RDS 資料庫。

   (稍後，在本教學課程的選項 2 中，您將透過選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源) 來嘗試 RDS 主控台中的自動連線功能。)

10. 對於 Virtual private cloud (VPC) (虛擬私有雲端 (VPC))，選擇 VPC。VPC 必須具備資料庫子網路群組。若要使用自動連線功能，EC2 執行個體和 RDS 資料庫必須位於相同的 VPC 中。
11. 請保留本頁面中其他欄位的所有預設值。
12. 選擇建立資料庫。

    在 Databases (資料庫) 畫面中，新資料庫的 Status (狀態) 為 Creating (正在建立)，直到資料庫準備就緒可供使用為止。狀態變更為 Available (可用) 時，您便可連線至資料庫。視資料庫類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新資料庫才會可用。

檢視動畫：建立 RDS 資料庫



任務 2 (選用)：啟動 EC2 執行個體

> **Note**
>
> 啟動執行個體並不是本教學課程的重點。如果您已有 Amazon EC2 執行個體，並且您想在本教學課程中使用該執行個體，則可跳過此任務。
> 如果您使用現有的 EC2 執行個體，則請確保其與 RDS 資料庫位於相同的 VPC 中，以便您可以使用自動連線功能。

此任務的目標是啟動 EC2 執行個體，以便您可以完成任務 3，您可以在其中設定 EC2 執行個體和 Amazon RDS 資料庫之間的連線。此任務中的步驟會設定 EC2 執行個體，如下所示：

- 執行個體名稱：**`tutorial-instance-1`**
- AMI：Amazon Linux 2
- 執行個體類型：`t2.micro`
- 自動指派公有 IP：已啟用
- 具有以下三個規則的安全群組：
  - 允許來自您 IP 地址的 SSH
  - 允許來自任何地方的 HTTPS 流量
  - 允許來自任何地方的 HTTP 流量

> **Important**
>
> 在生產環境中，您應設定執行個體以滿足您的特定需求。

啟動 EC2 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從區域選取器 (位於右上角) 中選取 AWS 區域。執行個體和 RDS 資料庫必須位於相同區域，才能使用 EC2 主控台中的自動連線功能。
3. 在 EC2 Dashboard (EC2 儀表板) 中，選擇 Launch Instance (啟動執行個體)。

4. 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入用於識別執行個體的名稱。針對本教學課程，將執行個體命名為 **`tutorial-instance-1`**。雖然執行個體名稱不是強制性的，但是當您在 EC2 主控台中選取執行個體時，名稱可協助您輕鬆識別其。
5. 在 Application and OS Images (應用程式和作業系統映像) 中，選擇符合您 Web 伺服器需求的 AMI。本教學課程使用 Amazon Linux 2。
6. 在 Instance type (執行個體類型) 中，針對 Instance type (執行個體類型)，選取符合您 Web 伺服器需求的執行個體類型。本教學課程使用的是 `t2.micro`。

> **Note**
>
> 視您建立帳戶的時間而定，您可能有資格使用免費方案下的 Amazon EC2。
> 如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。如果某個執行個體類型可用於免費方案，則標記為符合免費方案條件。
> 如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。
> 如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。

7. 在 key pair (login) (金鑰對 (登入)) 中，針對 Key pair name (金鑰對名稱) 選擇您的金鑰對。
8. 在Network settings (網路設定) 中，執行下列操作：

   a. 對於 Network (網路) 和 Subnet (子網路)，如果您尚未變更預設 VPC 或子網路，則可以保留預設設定。

      如果您已變更預設 VPC 或子網路，則請檢查下列項目：

      i. 執行個體和 RDS 資料庫必須位於相同的 VPC 中，才能使用自動連線功能。依預設，您只有一個 VPC。

      ii. 您要在其中啟動執行個體的 VPC 必須連接網際網路閘道，以便您可以從網際網路存取 Web 伺服器。您的預設 VPC 會與網際網路閘道一起自動設定。

      iii. 若要確保您的執行個體接收公有 IP 地址，針對 Auto-assign public IP (自動指派公有 IP)，檢查是否已選取 Enable (啟用)。如果選取 Disable (停用)，則請選取 Edit (編輯) (位於 Network Settings (網路設定) 右側)，然後針對 Auto-assign public IP (自動指派公有 IP) 選取 Enable (啟用)。

b. 若要使用 SSH 連線至執行個體，您需要一個安全群組規則來授權來自您電腦公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。根據預設，當您啟動執行個體時，會建立新的安全群組，其中包含允許從任何地方傳入 SSH 流量的規則。

若要確保只有您的 IP 位址可以連線到您的執行個體，請在防火牆 (安全群組) 下，從允許 SSH 流量來源核取方塊旁的下拉式清單中，選擇我的 IP。

c. 若要允許流量從網際網路傳輸到您的執行個體，請選取下列核取方塊：

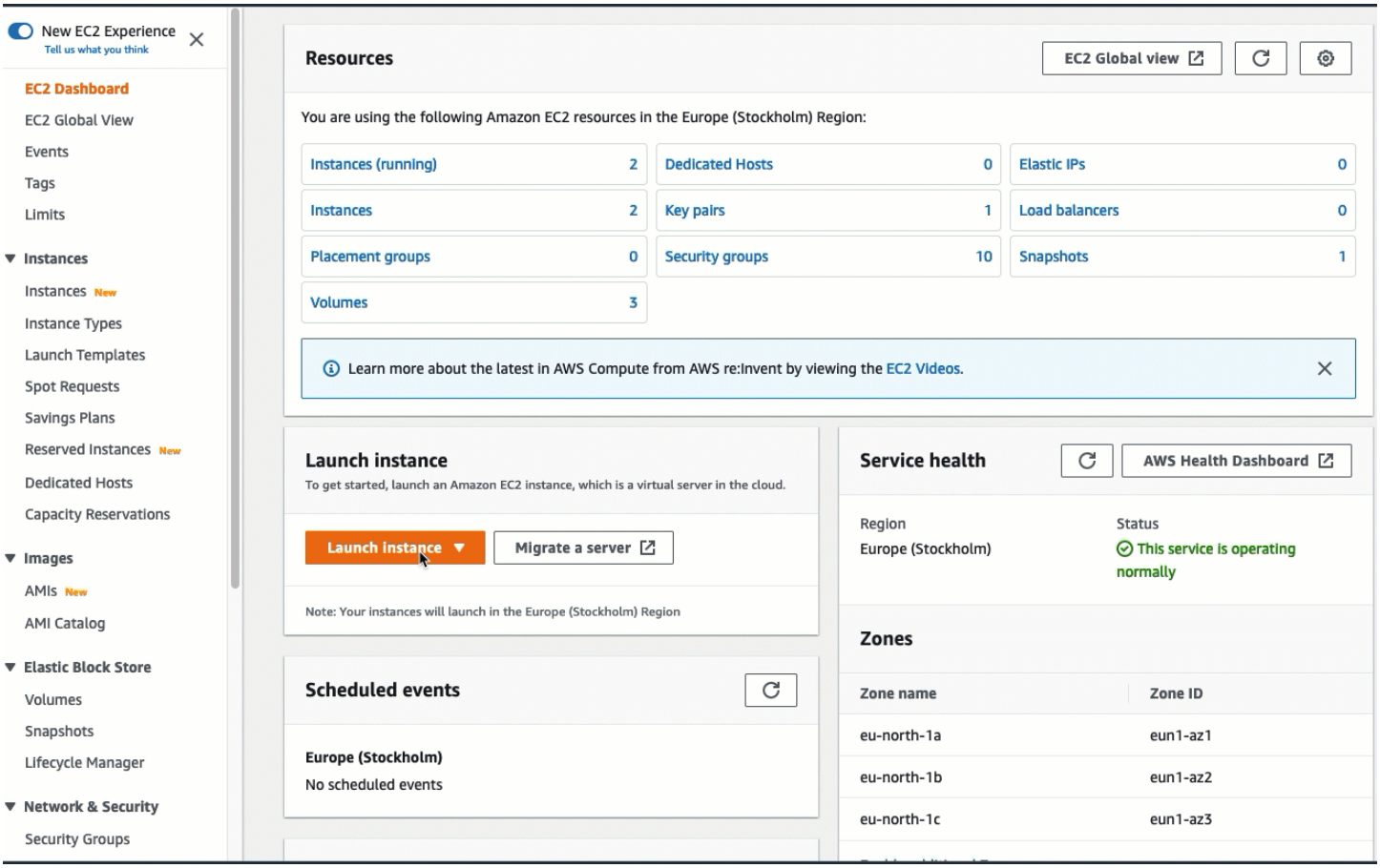
- Allow HTTPs traffic from the internet (允許來自網際網路的 HTTPS 流量)
- Allow HTTP traffic from the internet (允許來自網際網路的 HTTP 流量)

9. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。
10. 保持確認頁面開啟。當您自動將執行個體連線至資料庫時，下一個任務會需要它。

如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

如需有關啟動執行個體的詳細資訊，請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

檢視動畫：啟動 EC2 執行個體



任務 3：自動將 EC2 執行個體連線到 RDS 資料庫

此任務的目標是使用 EC2 主控台中的自動連線功能，自動設定 EC2 執行個體與 RDS 資料庫之間的連線。

若要使用 EC2 主控台自動將 EC2 執行個體連線至 RDS 資料庫

1. 在執行個體啟動確認頁面 (應該從先前的任務中開啟) 中，選擇 Connect an RDS database (連線 RDS 資料庫)。

   如果已關閉確認頁面，則請按照下列步驟操作：

   a. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

   b. 在導覽窗格中，選擇執行個體。

   c. 選取您剛建立的 EC2 執行個體，然後選取 Actions (動作)、Networking (聯網)、Connect RDS database (連線 RDS 資料庫)。

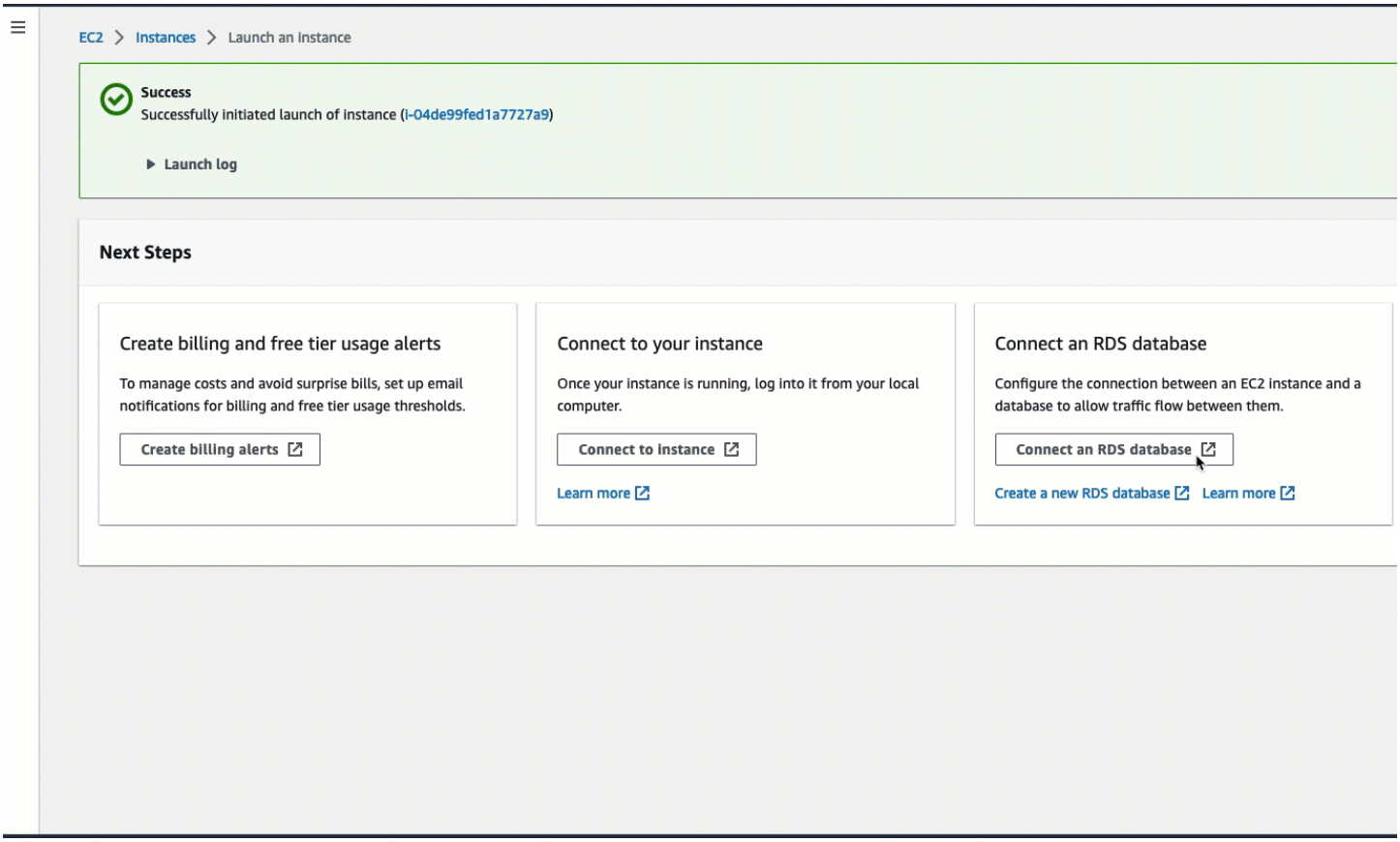
如果無法使用 Connect RDS database (連線 RDS 資料庫)，則請檢查 EC2 執行個體是否處於 Running (執行中) 狀態。

2. 對於 Database role (資料庫角色)，請選擇 Instance (執行個體)。在這種情況下，Instance (執行個體) 是指資料庫執行個體。
3. 對於 RDS database (RDS 資料庫)，請選擇您在任務 1 中建立的 RDS 資料庫。

> **Note**
>
> EC2 執行個體和 RDS 資料庫必須位於相同的 VPC 中，才能互相連接。

4. 選擇連線。

檢視動畫：自動將新啟動的 EC2 執行個體連線至 RDS 資料庫



任務 4：驗證連線組態

此任務的目標是確認已建立兩個安全群組並指派給執行個體和資料庫。

當您使用主控台中的自動連線功能來設定連線時，會自動建立安全群組並指派給執行個體和資料庫，如下所示：

- 建立安全群組 rds-ec2-*x* 並新增至 RDS 資料庫。它有一個傳入規則，該規則引用 ec2-rds-*x* 安全群組作為其來源。這允許來自 EC2 執行個體的流量透過 ec2-rds-*x* 安全群組到達 RDS 資料庫。
- 建立安全群組 ec2-rds-*x* 並新增至 EC2 執行個體。它有一個傳出規則，該規則引用 rds-ec2-*x* 安全群組作為其目的地。這允許來自 EC2 執行個體的流量透過 rds-ec2-*x* 安全群組到達 RDS 資料庫。

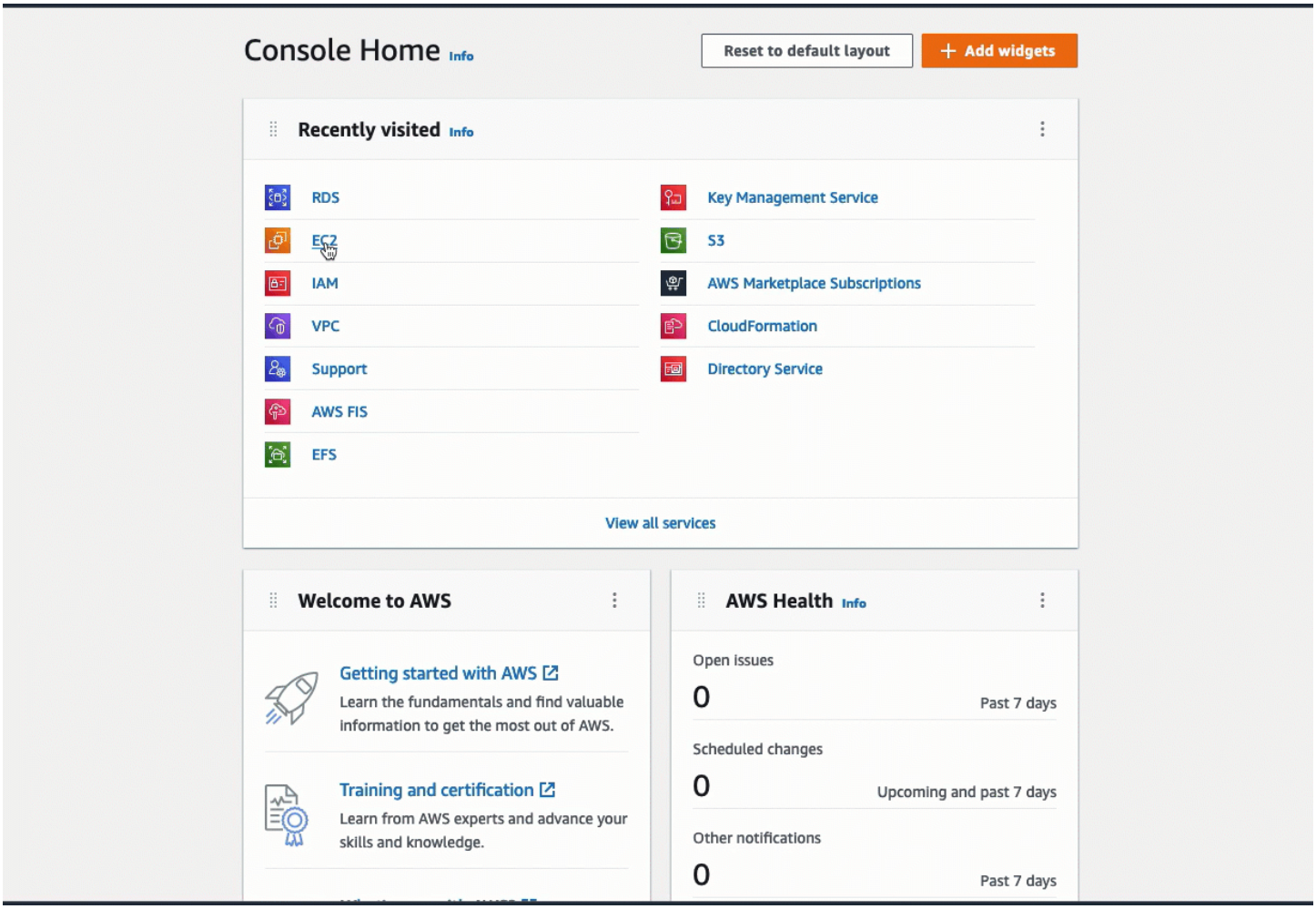
若要使用主控台驗證連線組態

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽頁面中，選擇 Databases (資料庫)。
3. 選擇您為此教學課程建立的 RDS 資料庫。
4. 在連線與安全索引標籤的安全性、VPC 安全群組中，確認會顯示名為 rds-ec2-*x* 的安全群組。
5. 選擇 rds-ec2-*x* 安全群組。EC2 主控台中的 Security Groups (安全群組) 畫面開啟。
6. 選擇 rds-ec2-*x* 安全群組，將其開啟。
7. 選擇 Inbound Rules (傳入規則) 索引標籤。
8. 確認下列安全群組規則存在，如下所示：
   - 類型：MYSQL/Aurora
   - 連接埠範圍：3306
   - 來源：*sg-0987654321example* / ec2-rds-*x* – 這是指派給您在上述步驟中驗證的 EC2 執行個體的安全群組。
   - 說明：允許來自 EC2 執行個體 (連接了 *sg-1234567890example*) 之連線的規則
9. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
10. 在導覽窗格中，選擇執行個體。
11. 選取您在上一個任務中選取要連線到 RDS 資料庫的 EC2 執行個體，然後選取 Security (安全) 索引標籤。
12. 在安全詳細資訊、安全群組中，確認名為 ec2-rds-*x* 的安全群組在清單中。*x* 是一個數字。
13. 選擇 ec2-rds-*x* 安全群組，將其開啟。
14. 選擇 Outbound rules (傳出規則) 索引標籤。
15. 確認下列安全群組規則存在，如下所示：
    - 類型：MYSQL/Aurora

- 連接埠範圍：3306
- 目的地：*sg-1234567890example* / rds-ec2-*x*
- 說明：允許從此安全群組連接的任何執行個體到 **database-tutorial** 之連線的規則

透過驗證這些安全群組和安全群組規則是否存在，並按照本程序所述將其指派給 RDS 資料庫和 EC2 執行個體，您可以使用自動連線功能驗證連線是否已自動設定。

檢視動畫：驗證連線組態



您已完成本教學課程的選項 1。您現在可以完成選項 2，會教您如何使用 RDS 主控台將 EC2 執行個體自動連線到 RDS 資料庫，或者您可完成選項 3，會教您如何手動設定在選項 1 中自動建立的安全群組。

任務 5 (選用)：清除

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

現在您已經完成教學課程，最好清理 (刪除) 您不再想要使用的任何資源。清除 AWS 資源可防止您的帳戶產生任何進一步的費用。

如果您針對本教學課程專門啟動 EC2 執行個體，則可以將其終止以停止產生與其相關的任何費用。

欲使用主控台來終止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您為此教學課程建立的執行個體，然後選取 Instance state (執行個體狀態)、Terminate instance (終止執行個體)。
4. 出現確認提示時，請選擇 Terminate (終止)。

如果您針對本教學課程專門建立了 RDS 資料庫，則可以將其刪除以停止產生與其相關的任何費用。

若要使用主控台刪除 RDS 資料庫

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選取您為此教學課程建立的 RDS 資料庫，然後選取 Actions (動作)、Delete (刪除)。
4. 在方塊中輸入 **`delete me`**，然後選擇 Delete (刪除)。

選項 2：使用 RDS 主控台自動將執行個體連線至 RDS 資料庫

選項 2 的目的是探索 RDS 主控台中的自動連線功能，其可自動設定 EC2 執行個體和 RDS 資料庫之間的連線，以允許從 EC2 執行個體到 RDS 資料庫的流量。在選項 3 中，您將了解如何手動設定連線。

任務

- 開始之前
- 任務 1 (選用)：啟動 EC2 執行個體
- 任務 2：建立 RDS 資料庫並自動將其連線到 EC2 執行個體
- 任務 3：驗證連線組態
- 任務 4 (選用)：清除

開始之前

完成此教學課程需要以下項目：

- 與 RDS 資料庫處於相同 VPC 的 EC2 執行個體。您可以使用現有的 EC2 執行個體或遵循任務 1 中的步驟來建立新執行個體。
- 呼叫以下操作的許可：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

任務 1 (選用)：啟動 EC2 執行個體

> **Note**
>
> 啟動執行個體並不是本教學課程的重點。如果您已有 Amazon EC2 執行個體，並且您想在本教學課程中使用該執行個體，則可跳過此任務。

此任務的目標是啟動 EC2 執行個體，以便您可以完成任務 2，您可以在其中設定 EC2 執行個體和 Amazon RDS 資料庫之間的連線。此任務中的步驟會設定 EC2 執行個體，如下所示：

- 執行個體名稱：**`tutorial-instance-2`**
- AMI：Amazon Linux 2
- 執行個體類型：`t2.micro`
- 自動指派公有 IP：已啟用
- 具有以下三個規則的安全群組：
  - 允許來自您 IP 地址的 SSH
  - 允許來自任何地方的 HTTPS 流量
  - 允許來自任何地方的 HTTP 流量

> **Important**
>
> 在生產環境中，您應設定執行個體以滿足您的特定需求。

啟動 EC2 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在 EC2 Dashboard (EC2 儀表板) 中，選擇 Launch Instance (啟動執行個體)。
3. 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入用於識別執行個體的名稱。針對本教學課程，將執行個體命名為 **`tutorial-instance-2`**。雖然執行個體名稱並非必要名稱，但是當您在 RDS 主控台中選取執行個體時，名稱可協助您輕鬆識別其。
4. 在 Application and OS Images (應用程式和作業系統映像) 中，選擇符合您 Web 伺服器需求的 AMI。本教學課程使用 Amazon Linux。

5. 在 Instance type (執行個體類型) 中，針對 Instance type (執行個體類型)，選取符合您 Web 伺服器需求的執行個體類型。本教學課程使用的是 `t2.micro`。

> **Note**
>
> 視您建立帳戶的時間而定，您可能有資格使用免費方案下的 Amazon EC2。
> 如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。如果某個執行個體類型可用於免費方案，則標記為符合免費方案條件。
> 如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。
> 如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。

6. 在 key pair (login) (金鑰對 (登入)) 中，針對 Key pair name (金鑰對名稱) 選擇您的金鑰對。
7. 在Network settings (網路設定) 中，執行下列操作：

   a. 對於 Network (網路) 和 Subnet (子網路)，如果您尚未變更預設 VPC 或子網路，則可以保留預設設定。

      如果您已變更預設 VPC 或子網路，則請檢查下列項目：

      i. 執行個體必須位於與 RDS 資料庫相同的 VPC 中，才能使用自動連線組態。依預設，您只有一個 VPC。

      ii. 您要在其中啟動執行個體的 VPC 必須連接網際網路閘道，以便您可以從網際網路存取 Web 伺服器。您的預設 VPC 會與網際網路閘道一起自動設定。

      iii. 若要確保您的執行個體接收公有 IP 地址，針對 Auto-assign public IP (自動指派公有 IP)，檢查是否已選取 Enable (啟用)。如果選取 Disable (停用)，則請選取 Edit (編輯) (位於 Network Settings (網路設定) 右側)，然後針對 Auto-assign public IP (自動指派公有 IP) 選取 Enable (啟用)。

   b. 若要使用 SSH 連線至執行個體，您需要一個安全群組規則來授權來自您電腦公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。根據預設，當您啟動執行個體時，會建立新的安全群組，其中包含允許從任何地方傳入 SSH 流量的規則。

若要確保只有您的 IP 位址可以連線到您的執行個體，請在防火牆 (安全群組) 下，從允許 SSH 流量來源核取方塊旁的下拉式清單中，選擇我的 IP。

c. 若要允許流量從網際網路傳輸到您的執行個體，請選取下列核取方塊：

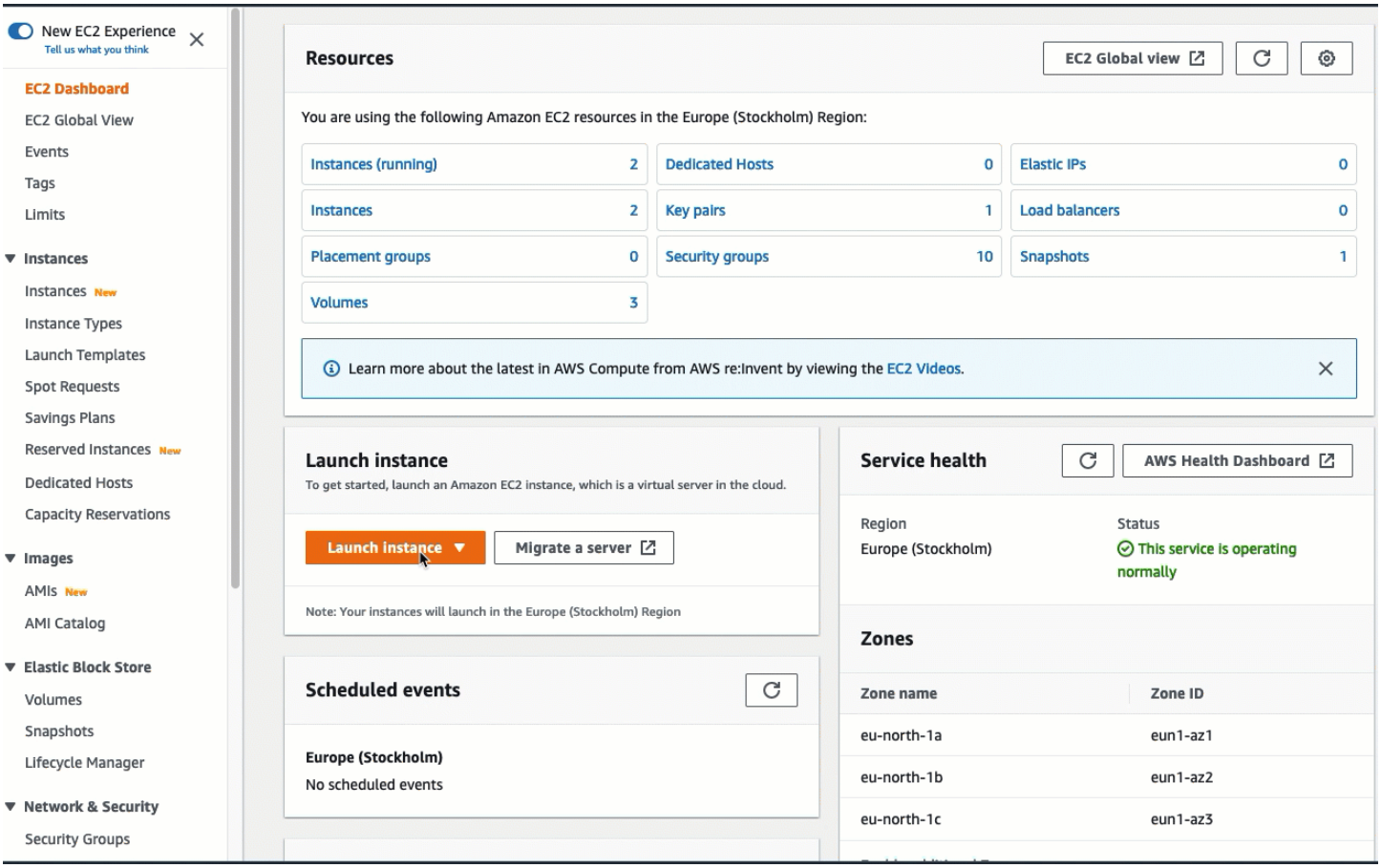
- Allow HTTPs traffic from the internet (允許來自網際網路的 HTTPS 流量)
- Allow HTTP traffic from the internet (允許來自網際網路的 HTTP 流量)

8. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。
9. 選擇 View all instances (檢視所有執行個體)，以關閉確認頁面並返回主控台。您的執行個體首先會處於 `pending` 狀態，然後進入 `running` 狀態。

   如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱[對 Amazon EC2 執行個體啟動問題進行疑難排解](#)。

如需有關啟動執行個體的詳細資訊，請參閱 [使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體](#)。

檢視動畫：啟動 EC2 執行個體



任務 2：建立 RDS 資料庫並自動將其連線到 EC2 執行個體

此任務的目標是建立 RDS 資料庫，並使用 RDS 主控台中的自動連線功能，自動設定 EC2 執行個體與 RDS 資料庫之間的連線。此任務中的步驟會設定資料庫執行個體，如下所示：

- 引擎類型：MySQL
- 範本：免費方案
- 資料庫執行個體識別符：**tutorial-database**
- 資料庫執行個體類別：db.t3.micro

> **Important**
>
> 在生產環境中，您應設定執行個體以滿足您的特定需求。

若要建立 RDS 資料庫並自動將其連線至 EC2 執行個體

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 從區域選擇器 （右上角），選擇您在 AWS 區域 其中建立 EC2 執行個體的 。EC2 執行個體和 RDS 資料庫必須位於相同的區域中。
3. 在儀表板中，選擇 Create database (建立資料庫)。
4. 在 Choose a database creation method (選取資料庫建立方法) 中，確認選取 Standard Create (標準建立)。如果您選擇 Easy create (輕鬆建立)，則無法使用自動連線功能。
5. 在 Engine options (引擎選項) 中，針對 Engine type (引擎類型) 選擇 MySQL。
6. 在 Templates (範本) 中，選擇符合您需求的範例範本。在本教學課程中，選擇 Free tier (免費方案) 免費建立 RDS 資料庫。不過，請注意，只有在您的帳戶符合免費方案資格時，才能使用免費方案。在 Free tier (免費方案) 方塊中選擇 Info (資訊) 連結，可閱讀更多資訊。
7. 在 Settings (設定) 下，執行下列動作：
   a. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入資料庫的名稱。針對本教學，輸入 **tutorial-database**。
   b. 對於 Master username (主要使用者名稱)，請保留預設名稱，即 **admin**。
   c. 對於 Master password (主要密碼)，請輸入您在本教學課程中可以記住的密碼，然後在 Confirm password (確認密碼) 中再次輸入密碼。

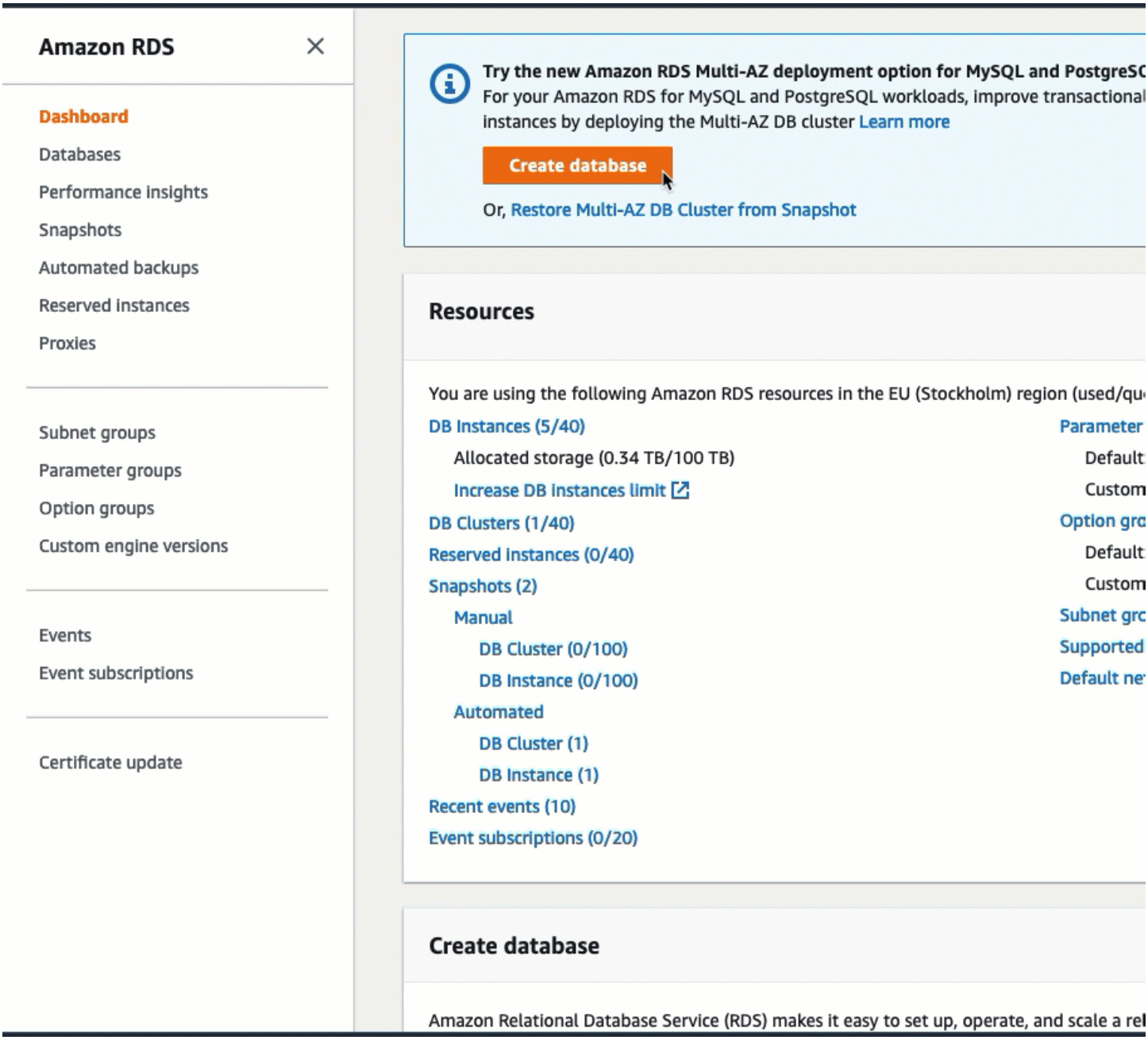
8. 在執行個體組態中，將資料庫執行個體類別保留為預設值，即 db.t3.micro。如果您的帳戶符合免費方案資格，則可以免費使用此執行個體。如需詳細資訊，請參閱 AWS 免費方案。
9. 在 Connectivity (連線) 中，對於 Compute resource (運算資源)，選擇 Connect to an EC2 compute resource (連線至 EC2 運算資源)。這是 RDS 主控台中的自動連線功能。
10. 針對 EC2 instance (EC2 執行個體)，選擇您想要連線的 EC2 執行個體。在本教學課程中，您可以選擇在上一個任務中建立的執行個體 (您已命名為 **`tutorial-instance`**)，或選擇另一個現有執行個體。如果在清單中沒有看到您的執行個體，則請選擇 Connectivity (連線) 右側的重新整理圖示。

    使用自動連線功能時，會將安全群組新增至此 EC2 執行個體，並將另一個安全群組新增至 RDS 資料庫。系統會自動設定安全群組，以允許 EC2 執行個體和 RDS 資料庫之間的流量。在下一個任務中，您將確認安全群組是否已建立並指派給 EC2 執行個體和 RDS 資料庫。
11. 選擇建立資料庫。

    在 Databases (資料庫) 畫面中，新資料庫的 Status (狀態) 為 Creating (正在建立)，直到資料庫準備就緒可供使用為止。狀態變更為 Available (可用) 時，您便可連線至資料庫。視資料庫類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新資料庫才會可用。

若要進一步了解，請參閱「Amazon RDS 使用者指南」中的設定與 EC2 執行個體的自動網路連線。

檢視動畫：建立 RDS 資料庫並自動將其連線至 EC2 執行個體



任務 3：驗證連線組態

此任務的目標是確認已建立兩個安全群組並指派給執行個體和資料庫。

當您使用主控台中的自動連線功能來設定連線時，會自動建立安全群組並指派給執行個體和資料庫，如下所示：

- 建立安全群組 rds-ec2-*x* 並新增至 RDS 資料庫。它有一個傳入規則，該規則引用 ec2-rds-*x* 安全群組作為其來源。這允許來自 EC2 執行個體的流量透過 ec2-rds-*x* 安全群組到達 RDS 資料庫。
- 建立安全群組 ec2-rds-*x* 並新增至 EC2 執行個體。它有一個傳出規則，該規則引用 rds-ec2-*x* 安全群組作為其目的地。這允許來自 EC2 執行個體的流量透過 rds-ec2-*x* 安全群組到達 RDS 資料庫。

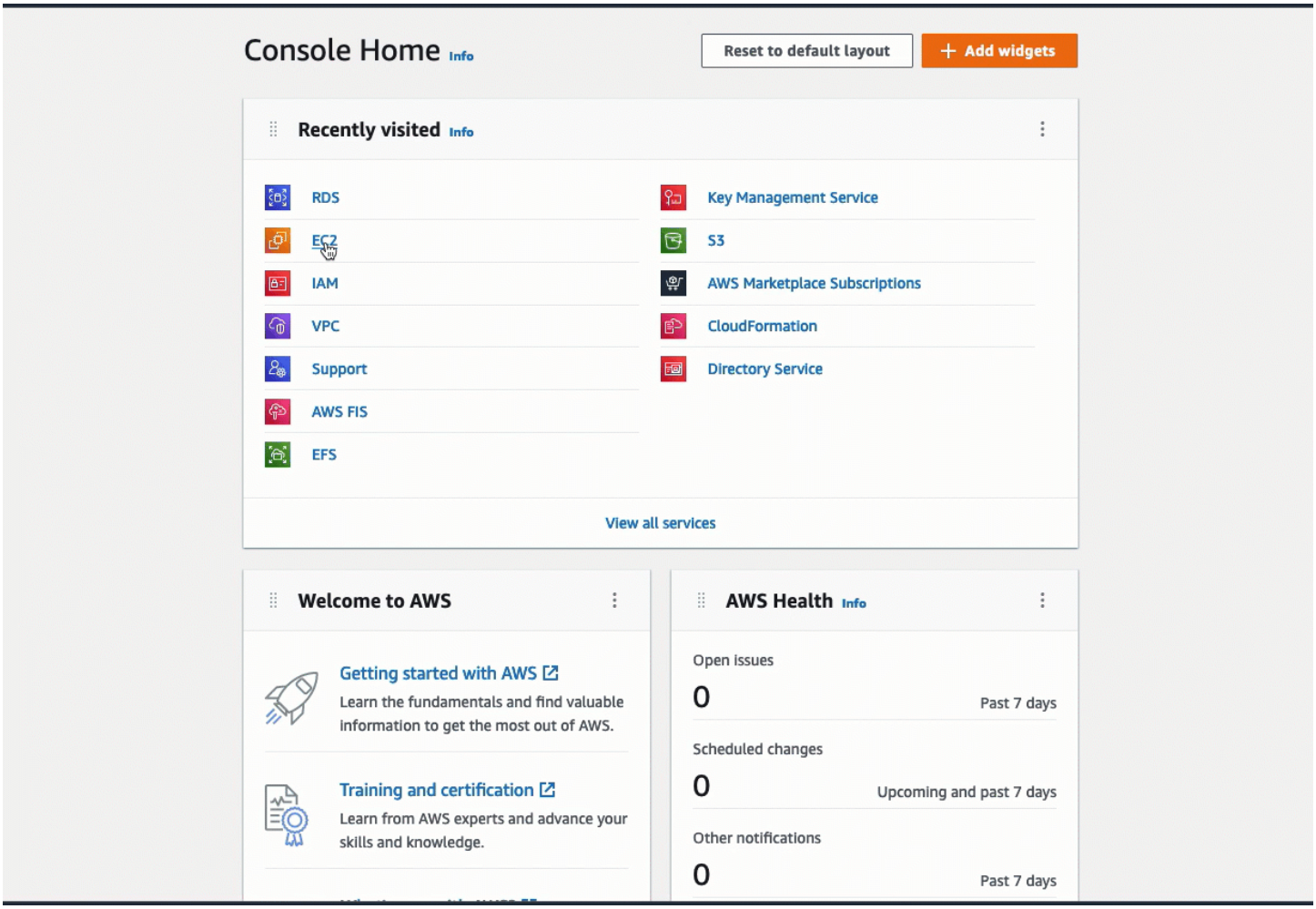
若要使用主控台驗證連線組態

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽頁面中，選擇 Databases (資料庫)。
3. 選擇您為此教學課程建立的 RDS 資料庫。
4. 在連線與安全索引標籤的安全性、VPC 安全群組中，確認會顯示名為 rds-ec2-*x* 的安全群組。
5. 選擇 rds-ec2-*x* 安全群組。EC2 主控台中的 Security Groups (安全群組) 畫面開啟。
6. 選擇 rds-ec2-*x* 安全群組，將其開啟。
7. 選擇 Inbound Rules (傳入規則) 索引標籤。
8. 確認下列安全群組規則存在，如下所示：
   - 類型：MYSQL/Aurora
   - 連接埠範圍：3306
   - 來源：*sg-0987654321example* / ec2-rds-*x* – 這是指派給您在上述步驟中驗證的 EC2 執行個體的安全群組。
   - 說明：允許來自 EC2 執行個體 (連接了 *sg-1234567890example*) 之連線的規則
9. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
10. 在導覽窗格中，選擇執行個體。
11. 選取您在上一個任務中選取要連線到 RDS 資料庫的 EC2 執行個體，然後選取 Security (安全) 索引標籤。
12. 在安全詳細資訊、安全群組中，確認名為 ec2-rds-*x* 的安全群組在清單中。*x* 是一個數字。
13. 選擇 ec2-rds-*x* 安全群組，將其開啟。
14. 選擇 Outbound rules (傳出規則) 索引標籤。
15. 確認下列安全群組規則存在，如下所示：
    - 類型：MYSQL/Aurora
    - 連接埠範圍：3306

- 目的地：***sg-1234567890example*** / rds-ec2-***x***
- 說明：允許從此安全群組連接的任何執行個體到 **`database-tutorial`** 之連線的規則

透過驗證這些安全群組和安全群組規則是否存在，並按照本程序所述將其指派給 RDS 資料庫和 EC2 執行個體，您可以使用自動連線功能驗證連線是否已自動設定。

檢視動畫：驗證連線組態



您已完成本教學課程的選項 2。您現在可以完成選項 3，其會教導您如何手動設定在選項 2 中自動建立的安全群組。

任務 4 (選用)：清除

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。

終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

現在您已經完成教學課程，最好清理 (刪除) 您不再想要使用的任何資源。清除 AWS 資源可防止您的帳戶產生任何進一步的費用。

如果您針對本教學課程專門啟動 EC2 執行個體，則可以將其終止以停止產生與其相關的任何費用。

欲使用主控台來終止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您為此教學課程建立的執行個體，然後選取 Instance state (執行個體狀態)、Terminate instance (終止執行個體)。
4. 出現確認提示時，請選擇 Terminate (終止)。

如果您針對本教學課程專門建立了 RDS 資料庫，則可以將其刪除以停止產生與其相關的任何費用。

若要使用主控台刪除 RDS 資料庫

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選取您為此教學課程建立的 RDS 資料庫，然後選取 Actions (動作)、Delete (刪除)。
4. 在方塊中輸入 **`delete me`**，然後選擇 Delete (刪除)。

選項 3：透過建立安全群組，以手動方式將執行個體連線至 RDS 資料庫

選項 3 的目標是了解如何透過手動複製自動連線功能的組態來手動設定 EC2 執行個體和 RDS 資料庫之間的連線。

任務

- 開始之前
- 任務 1 (選用)：啟動 EC2 執行個體

- 任務 2 (選用)：建立 RDS 資料庫
- 任務 3：透過建立安全群組並將其指派給執行個體，手動將 EC2 執行個體連線至 RDS 資料庫
- 任務 4 (選用)：清除

開始之前

完成此教學課程需要以下項目：

- 與 RDS 資料庫處於相同 VPC 的 EC2 執行個體。您可以使用現有的 EC2 執行個體或遵循任務 1 中的步驟來建立新執行個體。
- 與 EC2 執行個體處於相同 VPC 的 RDS 資料庫。您可以使用現有的 RDS 資料庫或遵循任務 2 中的步驟來建立新資料庫。
- 呼叫以下操作的許可：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

任務 1 (選用)：啟動 EC2 執行個體

> **Note**
>
> 啟動執行個體並不是本教學課程的重點。如果您已有 Amazon EC2 執行個體，並且您想在本教學課程中使用該執行個體，則可跳過此任務。

此任務的目標是啟動 EC2 執行個體，以便您可以完成任務 3，您可以在其中設定 EC2 執行個體和 Amazon RDS 資料庫之間的連線。此任務中的步驟會設定 EC2 執行個體，如下所示：

- 執行個體名稱：**`tutorial-instance`**
- AMI：Amazon Linux 2
- 執行個體類型：`t2.micro`
- 自動指派公有 IP：已啟用
- 具有以下三個規則的安全群組：
  - 允許來自您 IP 地址的 SSH
  - 允許來自任何地方的 HTTPS 流量
  - 允許來自任何地方的 HTTP 流量

> **Important**
>
> 在生產環境中，您應設定執行個體以滿足您的特定需求。

啟動 EC2 執行個體

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/ec2/：// 開啟 Amazon EC2 主控台。
2. 在 EC2 Dashboard (EC2 儀表板) 中，選擇 Launch Instance (啟動執行個體)。
3. 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入用於識別執行個體的名稱。針對本教學課程，將執行個體命名為 **`tutorial-instance-manual-1`**。雖然執行個體名稱不是強制性的，但名稱可協助您輕鬆識別其。
4. 在 Application and OS Images (應用程式和作業系統映像) 中，選擇符合您 Web 伺服器需求的 AMI。本教學課程使用 Amazon Linux。
5. 在 Instance type (執行個體類型) 中，針對 Instance type (執行個體類型)，選取符合您 Web 伺服器需求的執行個體類型。本教學課程使用的是 `t2.micro`。

   > **Note**
   >
   > 視您建立帳戶的時間而定，您可能有資格使用免費方案下的 Amazon EC2。
   > 如果您的 是在 2025 年 7 月 15 AWS 帳戶 日之前建立的，而且還不到 12 個月，您可以在無法使用 t2.micro 的區域中選取 t2.micro 執行個體類型或 t3.micro 執行個體類型，在 免

費方案下使用 Amazon EC2。 請注意，當您啟動 t3.micro 執行個體時，其會預設為無限制模式，且依據 CPU 用量而定，這可能會產生額外費用。如果某個執行個體類型可用於免費方案，則標記為符合免費方案條件。
如果您在 2025 年 7 月 15 日 AWS 帳戶 或之後建立 ，則可以使用 t3.micro、t3.small、t4g.micro、t4g.small、c7i-flex.large 和 m7i-flex.large 執行個體類型 6 個月，或直到您的點數用完為止。
如需詳細資訊，請參閱2025 年 7 月 15 日之前和之後的免費方案優惠。

6. 在 key pair (login) (金鑰對 (登入)) 中，針對 Key pair name (金鑰對名稱) 選擇您的金鑰對。
7. 在Network settings (網路設定) 中，執行下列操作：

   a. 對於 Network (網路) 和 Subnet (子網路)，如果您尚未變更預設 VPC 或子網路，則可以保留預設設定。

      如果您已變更預設 VPC 或子網路，則請檢查下列項目：

      i. 執行個體必須與 RDS 資料庫處於相同的 VPC 中。依預設，您只有一個 VPC。
      ii. 您要在其中啟動執行個體的 VPC 必須連接網際網路閘道，以便您可以從網際網路存取 Web 伺服器。您的預設 VPC 會與網際網路閘道一起自動設定。
      iii. 若要確保您的執行個體接收公有 IP 地址，針對 Auto-assign public IP (自動指派公有 IP)，檢查是否已選取 Enable (啟用)。如果選取 Disable (停用)，則請選取 Edit (編輯) (位於 Network Settings (網路設定) 右側)，然後針對 Auto-assign public IP (自動指派公有 IP) 選取 Enable (啟用)。

   b. 若要使用 SSH 連線至執行個體，您需要一個安全群組規則來授權來自您電腦公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。根據預設，當您啟動執行個體時，會建立新的安全群組，其中包含允許從任何地方傳入 SSH 流量的規則。

      若要確保只有您的 IP 位址可以連線到您的執行個體，請在防火牆 (安全群組) 下，從允許 SSH 流量來源核取方塊旁的下拉式清單中，選擇我的 IP。

   c. 若要允許流量從網際網路傳輸到您的執行個體，請選取下列核取方塊：

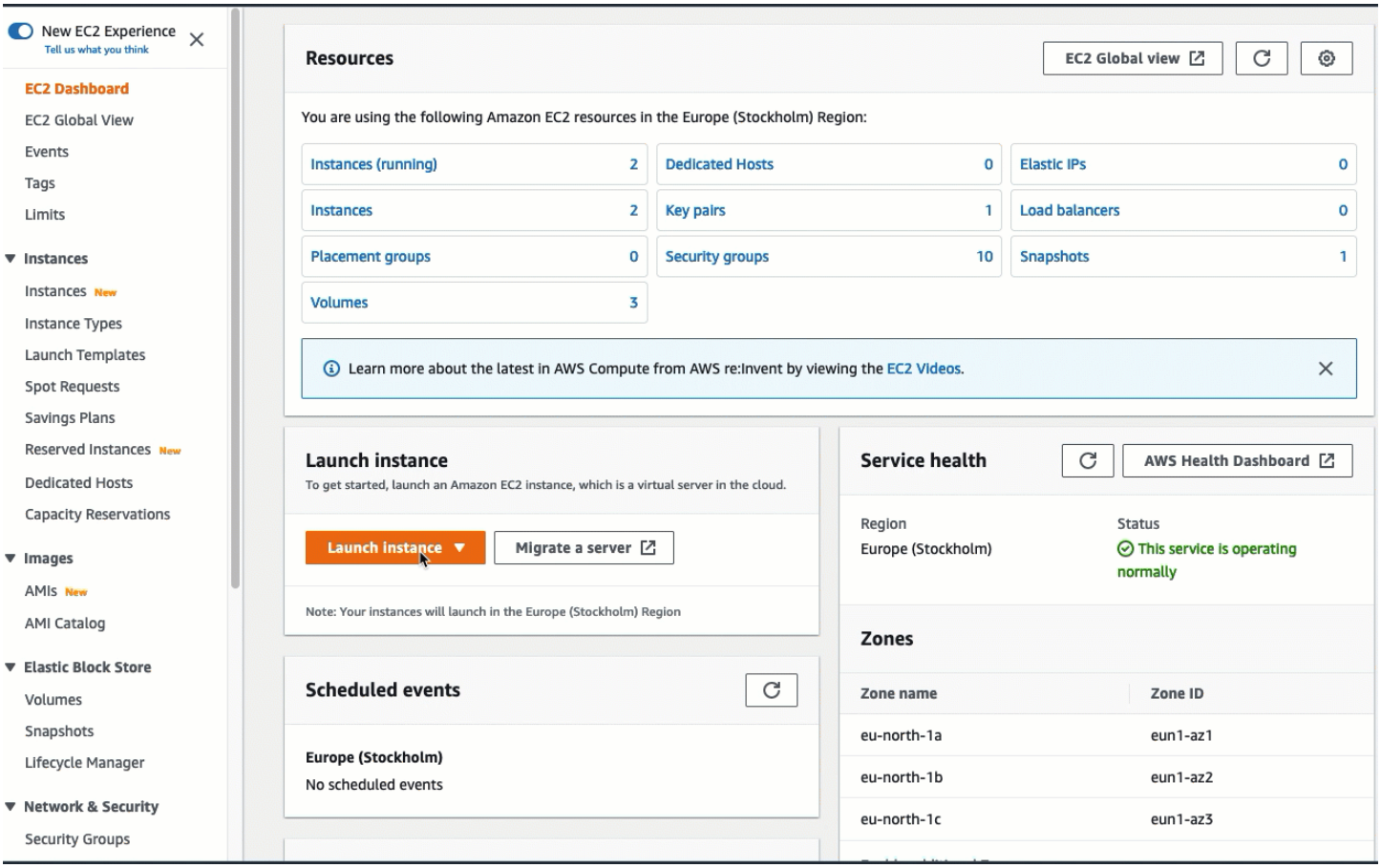
      - Allow HTTPs traffic from the internet (允許來自網際網路的 HTTPS 流量)
      - Allow HTTP traffic from the internet (允許來自網際網路的 HTTP 流量)

8. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。
9. 選擇 View all instances (檢視所有執行個體)，以關閉確認頁面並返回主控台。您的執行個體首先會處於 `pending` 狀態，然後進入 `running` 狀態。

如果執行個體無法啟動或狀態立即進入 `terminated` 而非 `running`，請參閱對 Amazon EC2 執行個體啟動問題進行疑難排解。

如需有關啟動執行個體的詳細資訊，請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

檢視動畫：啟動 EC2 執行個體



任務 2 (選用)：建立 RDS 資料庫

> **Note**
>
> 建立 RDS 資料庫並不是教學課程中本部分的重點。如果您已有 RDS 資料庫，並且您想在本教學課程中使用該資料庫，則可跳過此任務。

此任務的目標是建立 RDS 資料庫。將此執行個體連線至 EC2 執行個體時，您會在任務 3 中使用此執行個體。此任務中的步驟會設定 RDS 資料庫，如下所示：

- 引擎類型：MySQL
- 範本：免費方案
- 資料庫執行個體識別符：**tutorial-database-manual**
- 資料庫執行個體類別：db.t3.micro

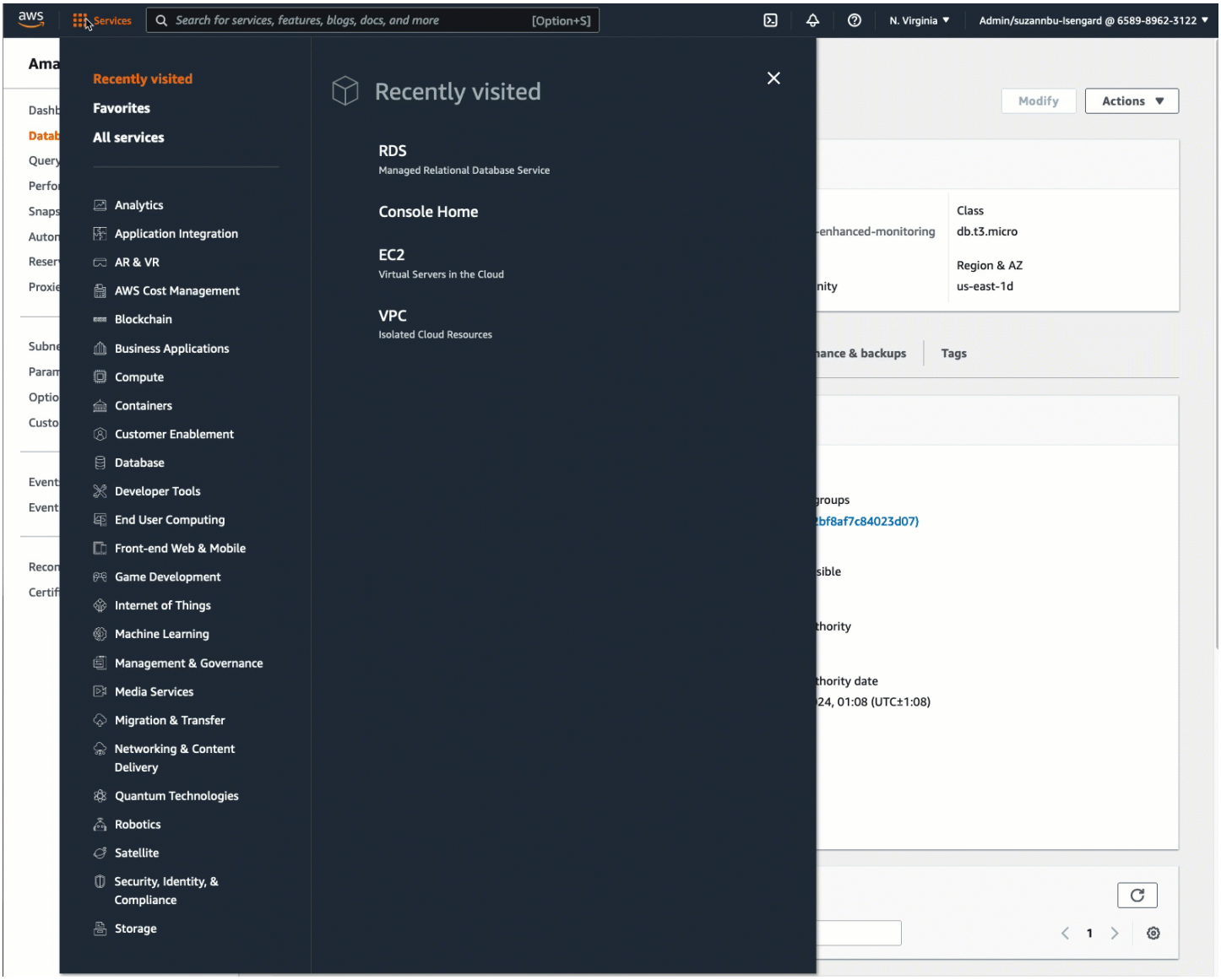
> **Important**
>
> 在生產環境中，您應設定執行個體以滿足您的特定需求。

建立 MySQL 資料庫執行個體

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 從區域選擇器 （右上角），選擇您在 AWS 區域 其中建立 EC2 執行個體的 。EC2 執行個體與資料庫執行個體必須處於相同區域。
3. 在儀表板中，選擇 Create database (建立資料庫)。
4. 在 Choose a database creation method (選擇資料庫建立方法) 中，選擇 Easy create (輕鬆建立)。當您選擇此選項時，無法使用自動設定連線的自動連線功能。
5. 在 Engine options (引擎選項) 中，針對 Engine type (引擎類型) 選擇 MySQL。
6. 在 DB instance size (資料庫執行個體大小) 中，選擇 Free tier (免費方案)。
7. 在 DB instance identifier (資料庫執行個體識別符) 中，輸入 RDS 資料庫的名稱。針對本教學，輸入 **tutorial-database-manual**。
8. 對於 Master username (主要使用者名稱)，請保留預設名稱，即 **admin**。
9. 對於 Master password (主要密碼)，請輸入您在本教學課程中可以記住的密碼，然後在 Confirm password (確認密碼) 中再次輸入密碼。
10. 選擇建立資料庫。

    在 Databases (資料庫) 畫面中，新資料庫執行個體的 Status (狀態) 為 Creating (正在建立)，直到資料庫執行個體準備就緒可供使用為止。狀態變更為 Available (可用) 時，您便能連線至資料庫執行個體。視資料庫執行個體類別和儲存體數量而定，可能需要最多 20 分鐘的時間，新執行個體才會可用。

檢視動畫：建立資料庫執行個體



任務 3：透過建立安全群組並將其指派給執行個體，手動將 EC2 執行個體連線至 RDS 資料庫

此任務的目標是透過手動執行下列動作來複製自動連線功能的連線組態：您建立兩個新的安全群組，然後將每個安全群組新增至 EC2 執行個體和 RDS 資料庫。

建立兩個新的安全群組，並分別指派給 EC2 執行個體和 RDS 資料庫

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 首先建立要新增至 EC2 執行個體的安全群組，如下所示：

a. 在導覽窗格中，選擇 Security Groups (安全群組)。

b. 選擇 Create Security Group (建立安全群組)。

c. 對於 Security group name (安全群組名稱)，輸入安全群組的描述性名稱。針對本教學，輸入 **ec2-rds-manual-configuration**。

d. 對於 Description (描述)，輸入簡短描述。針對本教學，輸入 **EC2 instance security group to allow EC2 instance to securely connect to RDS database**。

e. 選擇建立安全群組。建立 RDS 資料庫安全群組之後，您將回到此安全群組以新增傳出規則。

3. 現在，建立要新增至 RDS 資料庫的安全群組，如下所示：

a. 在導覽窗格中，選擇 Security Groups (安全群組)。

b. 選擇 Create Security Group (建立安全群組)。

c. 對於 Security group name (安全群組名稱)，輸入安全群組的描述性名稱。針對本教學，輸入 **rds-ec2-manual-configuration**。

d. 對於 Description (描述)，輸入簡短描述。針對本教學，輸入 **RDS database security group to allow EC2 instance to securely connect to RDS database**。

e. 在 Inbound rules (傳入規則) 中，選擇 Add rule (新增規則)，然後執行下列動作：

i. 對於 Type (類型)，選擇 MYSQL/Aurora。

ii. 對於 Source (來源)，選擇您在此程序的步驟 2 中建立的 EC2 執行個體安全群組 ec2-rds-manual-configuration。

f. 選擇建立安全群組。

4. 編輯 EC2 執行個體安全群組以新增傳出規則，如下所示：

a. 在導覽窗格中，選擇安全群組。

b. 選取 EC2 執行個體安全群組 (已命名為 **ec2-rds-manual-configuration**)，然後選取 Outbound rules (傳出規則) 索引標籤。

c. 選擇 Edit outbound rules (編輯傳出規則)。

d. 選擇 Add rule (新增規則) 並執行下列動作：

i. 對於 Type (類型)，選擇 MYSQL/Aurora。

ii. 對於目的地，選擇您在此程序的步驟 3 中建立的 RDS 資料庫安全群組 rds-ec2-manual-configuration。

iii. 選擇儲存規則。

5. 將 EC2 執行個體安全群組新增至 EC2 執行個體，如下所示：

   a. 在導覽窗格中，選擇執行個體。

   b. 選取 EC2 執行個體，然後選取 Actions (動作)、Security (安全性)、Change security groups (變更安全群組)。

   c. 在 Associated security groups (關聯的安全群組) 中，選取 Select security groups (選取安全群組) 欄位，選取您先前建立的 ec2-rds-manual-configuration，然後選取 Add security group (新增安全群組)。

   d. 選擇儲存。

6. 將 RDS 資料庫安全群組新增至 RDS 資料庫，如下所示：

   a. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。

   b. 在導覽窗格中，選取 Databases (資料庫)，然後選取您的資料庫。

   c. 選擇 Modify (修改)。

   d. 在 Connectivity (連線) 中，針對 Security group (安全群組) 選擇您先前建立的 rds-ec2-manual-configuration，然後選擇 Continue (繼續)。

   e. 在 Scheduling of modifications (修改排程) 中，選擇 Apply immediately (立即套用)。

   f. 選擇 Modify DB instance (修改資料庫執行個體)。

   您現在已完成手動步驟，這些步驟會模擬當使用自動連線功能時所發生的自動步驟。

您已完成本教學課程的選項 3。如果您已完成選項 1、2 和 3，且不再需要本教學課程中建立的資源，您應該將其刪除，以避免產生不必要的費用。如需詳細資訊，請參閱任務 4 (選用)：清除。

任務 4 (選用)：清除

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

現在您已經完成教學課程，最好清理 (刪除) 您不再想要使用的任何資源。清除 AWS 資源可防止您的帳戶產生任何進一步的費用。

如果您針對本教學課程專門啟動 EC2 執行個體，則可以將其終止以停止產生與其相關的任何費用。

欲使用主控台來終止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您為此教學課程建立的執行個體，然後選取 Instance state (執行個體狀態)、Terminate instance (終止執行個體)。
4. 出現確認提示時，請選擇 Terminate (終止)。

如果您針對本教學課程專門建立了 RDS 資料庫，則可以將其刪除以停止產生與其相關的任何費用。

若要使用主控台刪除 RDS 資料庫

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選取您為此教學課程建立的 RDS 資料庫，然後選取 Actions (動作)、Delete (刪除)。
4. 在方塊中輸入 **`delete me`**，然後選擇 Delete (刪除)。

EC2 機群 和 Spot 機群

EC2 機群和 Spot 機群旨在提供一種實用方式，在單一操作中啟動數十個、數百個或數千個 Amazon EC2 執行個體的機群。機群中的每個執行個體都由啟動範本設定，或是由您在啟動時手動設定的一組啟動參數設定。

主題

- 功能和優勢
- 最好的機群方法是什麼？
- EC2 機群或 Spot 機群的組態選項
- 使用 EC2 機群
- 使用 Spot 機群
- 監控您的 EC2 機群或 Spot 機群
- EC2 機群的教學課程
- EC2 機群的範例 CLI 組態
- 範例 CLI 組態 Spot 機群
- EC2 機群和 Spot 機群的配額

功能和優勢

機群提供下列功能和優點，這些優點可讓您在多個 EC2 執行個體上執行應用程式時，最大限度地節省成本並最佳化可用性和效能。

多種執行個體類型

機群可以啟動多個執行個體類型，進而確保該機群不依賴於任何單一執行個體類型的可用性。這會增加機群中執行個體的整體可用性。

跨可用區域分佈執行個體

機群可在多個可用區域中啟動，協助您降低成本並改善可用性。若您的機群包含 Spot 執行個體，機群會根據您對價格與中斷的偏好設定，自動選取可用區域。

多個購買選項

機群可以啟動多種購買選項 (Spot 執行個體和隨需執行個體)，讓您透過 Spot 執行個體用量來最佳化成本。您也可以將預留執行個體和 Savings Plan 折扣與機群中的隨需執行個體搭配使用，善加利用這些折扣。

自動替換 Spot 執行個體

如果您的機群包含 Spot 執行個體，當您的 Spot 執行個體中斷時，它可以自動請求替換 Spot 容量。透過容量重新平衡，機群也可以監控和主動替換具有較高中斷風險的 Spot 執行個體。

保留隨需容量

機群可以使用隨需容量保留來保留隨需容量。機群也可包括機器學習 (ML) 的容量區塊，讓您在為來保留 GPU 執行個體，以支援短期 ML 工作負載。

最好的機群方法是什麼？

一般而言，我們建議您使用 Amazon EC2 Auto Scaling 啟動 Spot 和隨需執行個體機群，因為它提供了可用來管理機群的其他功能。其他功能清單包括 Spot 和隨需執行個體的自動運作狀態檢查替換、以應用程式為基礎的運作狀態檢查，以及與 Elastic Load Balancing 的整合，可確保應用程式流量均勻分佈到運作良好的執行個體。當您使用 Amazon ECS、Amazon EKS（自我管理節點群組）和 Amazon VPC Lattice 等 AWS 服務時，也可以使用 Auto Scaling 群組。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南。

如果您無法使用 Amazon EC2 Auto Scaling，則可以考慮使用 EC2 機群或 Spot 機群。EC2 機群和 Spot 機群提供相同的核心功能。不過，EC2 機群只能使用命令列，不提供主控台支援。Spot Fleet 提供主控台支援，但使用沒有計劃投資的舊版 API。

請遵循下列表格來協助判斷要使用的機群。

機群方法	何時使用？	使用案例
Amazon EC2 Auto Scaling	• 您需要具有單一組態或混合組態的多個執行個體。 • 您想要自動化執行個體的生命週期管理。	建立 Auto Scaling 群組，管理執行個體的生命週期，同時維護所需的執行個體數量。支援在指定的最小和最大限制之間水平擴展 (新增更多執行個體)。

機群方法	何時使用？	使用案例
EC2 機群	• 您需要具有單一組態或混合組態的多個執行個體。 • 您希望自行管理您的執行個體生命週期。 • 如果您不需要自動擴展，建議您使用 instant 類型 EC2 機群。	在單一操作中建立隨需執行個體和 Spot 執行個體的 instant 機群，並具有隨執行個體類型、AMI、可用區域或子網路而異的多個啟動規格。Spot 執行個體分配策略預設為每單位 lowest-price ，但我們建議您將其變更為 price-capacity-optimized 。
Spot 機群	• 我們強烈建議您不要使用 Spot Fleet，因為它使用沒有計劃投資的舊版 API。 • 如果您想要管理執行個體生命週期，請使用 EC2 機群。 • 如果您不想要管理執行個體生命週期，請使用 Auto Scaling 群組。	只有在您需要主控台支援何時使用 EC2 機群的使用案例時，才使用 Spot 機群。

EC2 機群或 Spot 機群的組態選項

規劃 EC2 機群或 Spot 機群時，建議您在決定如何設定機群時考慮下列選項。

組態選項	問題	文件
機群請求類型	您要建立的機群是否要針對所需的目標容量提交一次性請求？或是隨時間維持目標容量？	EC2 機群和 Spot 機群請求類型
Spot 執行個體	您是否計劃在機群中包含 Spot 執行個體？ 請檢閱 Spot 最佳實務，並在規劃機群時使用這些最佳實務，讓您能以最低價格佈建執行個體。	Amazon EC2 Spot 的最佳實務

組態選項	問題	文件
機群的花費限制	您是否要限制每小時要為機群支付的費用？	設定 EC2 機群或 Spot 機群的花費限制
執行個體類型和屬性型執行個體類型選項	是否要自行指定機群中的執行個體類型，或讓 Amazon EC2 選取符合您應用程式需求的執行個體類型？	指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型
執行個體權重	您是否要為每個執行個體類型指派權重，以代表其運算容量和效能，進而讓 Amazon EC2 選取任何可用的執行個體類型組合，以滿足您想要的目標容量？	使用執行個體權重來管理 EC2 機群或 Spot 機群的成本和效能
分配策略	您是否要自行決定是否針對可用容量、價格或執行個體類型進行最佳化，以用於機群中的 Spot 執行個體和隨需執行個體？	使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量
容量重新平衡	您是否希望機群自動取代具有風險的 Spot 執行個體？	在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體
隨需容量保留	是否要為機群中的隨需執行個體預留容量？	使用容量保留功能來保留 EC2 機群中的隨需容量
多個容量保留類型	您是否要使用單一容量保留資源群組跨多個容量保留類型啟動執行個體：隨需容量保留、ML 容量區塊和中斷容量保留，並選擇性地回復到隨需容量？	教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體

EC2 機群和 Spot 機群請求類型

> **Note**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

EC2 機群或 Spot 機群的請求類型會決定請求是同步還是非同步，以及是否為針對所需目標容量的一次性請求，還是持續努力隨時間維持容量。設定機群時，您必須指定請求類型。

EC2 機群和 Spot 機群都提供兩種請求類型：`request` 和 `maintain`。此外，EC2 機群提供名為 `instant` 的第三個請求類型。

機群請求類型

`instant` (僅限 EC2 機群)

如果您將請求類型設定為 `instant`，EC2 機群會針對您所需容量送出同步的一次性請求。在 API 回應中，會傳回已啟動的執行個體，並為無法啟動的執行個體提供錯誤資訊。如需詳細資訊，請參閱設定 instant 類型的 EC2 機群。

`request`

如果您將請求類型設定為 `request`，機群會針對您所需容量送出非同步的一次性請求。如果容量因為 Spot 中斷而減少，機群不會嘗試補充 Spot 執行個體；如果無法使用容量，也不會在替代的 Spot 容量集區中提交請求。使用主控台建立 `request` 類型 Spot 機群時，請清除維護目標容量核取方塊。

`maintain` (default)

如果您將請求類型設定為 `maintain`，機群會針對您所需容量送出非同步請求，並自動補充所有中斷的 Spot 執行個體以進行維持。使用主控台建立 `maintain` 類型 Spot 機群時，請選擇維護目標容量核取方塊。

設定 instant 類型的 EC2 機群

instant 類型的 EC2 機群是同步的一次性請求，只會嘗試啟動您所需的容量一次。API 回應列出已啟動的執行個體，以及無法啟動的執行個體錯誤。使用 instant 類型的 EC2 機群具有本文描述的多種好處。本文末尾處會提供範例組態。

對於需要僅啟動 API 來啟動 EC2 執行個體的工作負載，您可以使用 RunInstances API。但是，憑藉 RunInstances，您只能在同一請求中啟動隨需執行個體或 Spot 執行個體，但不能同時啟動兩者。此外，當您使用 RunInstances 啟動 Spot 執行個體時，您的 Spot 執行個體請求僅限於一種執行個體類型和一個可用區域。此操作針對單個 Spot 容量集區 (一組具有相同執行個體類型和可用區域的未使用執行個體)。如果 Spot 容量集區沒有足夠的 Spot 執行個體容量來滿足您的請求，則 RunInstances 呼叫將會失敗。

相比使用 RunInstances 啟動 Spot 執行個體，我們建議使用具有設為 `instant` 的 `type` 參數的 CreateFleet API，以獲得以下好處：

- 在一個請求中啟動隨需執行個體和 Spot 執行個體。EC2 機群可以啟動隨需執行個體、Spot 執行個體或兩者皆有。如果有可用容量且您的請求中每個小時的價格上限超過 Spot 價格，則會滿足 Spot 執行個體的請求。
- 增加 Spot 執行個體的可用性。透過使用 `instant` 類型 EC2 機群，您可以按照 Spot 最佳實務啟動 Spot 執行個體，獲得以下好處：
  - Spot 最佳實務：對執行個體類型和可用區域具有彈性。

    好處：透過指定數個執行個體類型和可用區域，您可以增加 Spot 容量集區的數目。這可讓 Spot 服務更有機會尋找並配置您所需的 Spot 運算容量。一個很好的經驗法則是針對每個工作負載的至少 10 個執行個體類型上具有彈性，並確定所有可用區域均已設定為在 VPC 中使用。
  - Spot 最佳實務：使用 price-capacity-optimized 分配策略。

    好處：`price-capacity-optimized` 分配策略會自動從最可用的 Spot 容量集區識別執行個體並會自動從最低價格的集區佈建執行個體。由於您的 Spot 執行個體容量來自具有最佳容量的集區，因此可降低 Amazon EC2 需要回收容量時，中斷 Spot 執行個體的可能性。
- 取得更廣泛的功能組合。對於需要僅啟動 API 的工作負載，並且您更希望管理您的執行個體的生命週期，而不是讓 EC2 機群為您進行管理，請使用 `instant` 類型 EC2 機群，而不是 RunInstances API。EC2 機群提供比 RunInstances 更廣泛的功能組合，如以下範例所示。對於所有其他工作負載，您應使用 Amazon EC2 Auto Scaling，因為它為各種工作負載提供更全面的功能組合，例如 ELB 後端應用程式、容器化工作負載和佇列處理任務。

您可以使用即時類型的 EC2 機群，在容量區塊中啟動執行個體。如需詳細資訊，請參閱教學課程：設定您的 EC2 機群以啟動執行個體至容量區塊。

AWS Amazon EC2 Auto Scaling 和 Amazon EMR 等 服務會使用立即類型的 EC2 機群來啟動 EC2 執行個體。

立即類型 EC2 機群的先決條件

如需建立 EC2 機群的先決條件，請參閱 EC2 Fleet 先決條件。

instant 類型的 EC2 機群如何運作

當使用 `instant` 類型的 EC2 機群時，事件的順序如下所示：

1. 設定：設定 CreateFleet 請求類型為 `instant`。如需詳細資訊，請參閱建立 EC2 Fleet。請注意，在您進行 API 呼叫後，將無法修改它。
2. 請求：在您進行 API 呼叫時，Amazon EC2 機群會針對您所需容量送出同步的一次性請求。

3. 回應：API 回應列出已啟動的執行個體，以及無法啟動的執行個體錯誤。
4. 說明：您可以描述您的 EC2 機群、列出與 EC2 機群相關的執行個體，並檢視 EC2 機群的歷史記錄。
5. 終止執行個體：您隨時可終止執行個體。
6. 刪除機群請求：機群請求可手動或自動刪除：
   - 手動：在執行個體啟動後，您可以刪除機群請求。

     請注意，不支援具有執行中執行個體的已刪除 `instant` 機群。刪除 `instant` 機群時，Amazon EC2 會自動終止其所有執行個體。對於包含超過 1000 個執行個體的 機群，刪除請求可能會失敗。若您的機群有超過 1000 個執行個體，請先手動終止大部分執行個體，留下 1000 個或以下的執行個體。接著刪除機群，剩餘的執行個體將會自動終止。
   - 自動：Amazon EC2 會在以下任一時間後刪除機群請求：
     - 所有執行個體會終止。
     - 機群無法啟動任何執行個體。

範例

以下範例展示如何為不同的使用案例使用 `instant` 類型的 EC2 機群。如需使用 EC2 CreateFleet API 參數的詳細資訊，請參閱 Amazon EC2 API 參考中的 CreateFleet (建立機群)。

範例

- 範例 1：使用容量最佳化分配策略啟動 Spot 執行個體
- 範例 2：使用容量最佳化分配策略啟動單一 Spot 執行個體
- 範例 3：使用執行個體權重啟動 Spot 機群
- 範例 4：在單一可用區域內啟動 Spot 執行個體
- 範例 5：在單一可用區域內啟動單一執行個體類型的 Spot 執行個體
- 範例 6：僅在可啟動最小目標容量時才啟動 Spot 執行個體
- 範例 7：僅當可在單個可用區域中啟動相同執行個體類型的最低目標容量時，才能啟動 Spot 執行個體
- 範例 8：使用多個啟動範本來啟動執行個體
- 範例 9：啟動基於隨需執行個體的 Spot 執行個體
- 範例 10：基於使用容量保留和優先順序分配策略的隨需執行個體，使用容量最佳化分配策略啟動 Spot 執行個體
- 範例 11：使用容量最佳化的優先順序分配策略啟動 Spot 執行個體

- 範例 12：指定 Systems Manager 參數而非 AMI ID

範例 1：使用容量最佳化分配策略啟動 Spot 執行個體

下列範例指定 instant 類型 EC2 機群中所需的參數：啟動範本、目標容量、預設購買選項，以及啟動範本覆寫。

- 啟動範本由其啟動範本名稱和版本編號識別。
- 12 個啟動範本覆寫會指定 4 個不同執行個體類型和 3 個不同的子網路，每個都位於不同的可用區域。每個執行個體類型和子網路組合都定義了一個 Spot 容量集區，從而產生 12 個 Spot 容量集區。
- 機群的目標容量為 20 個執行個體。
- 預設購買選項為 spot，這會導致機群嘗試將 20 個 Spot 執行個體啟動到 Spot 容量集區中，且擁有啟動中執行個體數目之最佳容量。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
          "LaunchTemplateSpecification":{
             "LaunchTemplateName":"ec2-fleet-lt1",
             "Version":"$Latest"
          },
          "Overrides":[
             {
                "InstanceType":"c5.large",
                "SubnetId":"subnet-fae8c380"
             },
             {
                "InstanceType":"c5.large",
                "SubnetId":"subnet-e7188bab"
             },
             {
                "InstanceType":"c5.large",
                "SubnetId":"subnet-49e41922"
             },
             {
                "InstanceType":"c5d.large",
                "SubnetId":"subnet-fae8c380"
```

```
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 2：使用容量最佳化分配策略啟動單一 Spot 執行個體

您可以透過進行 instant 類型多個 EC2 機群的 API 呼叫，將 TotalTargetCapacity 設定為 1，以最佳方式一次啟動一個 Spot 執行個體。

下列範例指定 instant 類型 EC2 機群中所需的參數：啟動範本、目標容量、預設購買選項，以及啟動範本覆寫。啟動範本由其啟動範本名稱和版本編號識別。12 個啟動範本覆寫會擁有 4 個不同執行個體類型和 3 個不同的子網路，每個都位於不同的可用區域。機群的目標容量為 1 個執行個體，並且預設購買選項為 Spot，這會導致機群嘗試根據容量最佳化的分配策略從 12 個 Spot 容量集區之一啟動 Spot 執行個體，以從最可用的容量集區啟動 Spot 執行個體。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
```

```
                "InstanceType":"c5d.large",
                "SubnetId":"subnet-49e41922"
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-49e41922"
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 3：使用執行個體權重啟動 Spot 機群

下列範例使用執行個體權重，這表示價格為每單位小時而非每執行個體小時的價格。假設一個工作負載單位需要 15 GB 的記憶體和 4 個 vCPU，每個啟動組態會列出不同的執行個體類型和不同的權重，具體取決於執行個體上可以執行的工作負載單位數。例如，m5.xlarge (4 個 vCPU 和 16 GB 記憶體) 可以執行一個單位，並且權重為 1，m5.2xlarge (8 個 vCPU 和 32 GB 記憶體) 可以執行 2 個單位，並且權

重為 2，依此類推。目標總容量設定為 40 個單位。預設購買選項為 Spot，且分配策略容量最佳化，這會產生 40 個 m5.xlarge (40 除以 1)、20 個 m5.2xlarge (40 除以 2)、10 個 m5.4xlarge (40 除以 4)、5 個 m5.8xlarge (40 除以 8)，或基於容量最佳化分配策略，權重相加達到所需容量的執行個體類型的混合。

如需詳細資訊，請參閱 使用執行個體權重來管理 EC2 機群或 Spot 機群的成本和效能。

```
{
   "SpotOptions":{
      "AllocationStrategy":"capacity-optimized"
   },
   "LaunchTemplateConfigs":[
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":1
            },
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":2
            },
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":2
            },
```

```
            {
               "InstanceType":"m5.2xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":2
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":4
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":8
            }
         ]
      }
   ],
   "TargetCapacitySpecification":{
      "TotalTargetCapacity":40,
      "DefaultTargetCapacityType":"spot"
   },
   "Type":"instant"
```

```
}
```

範例 4：在單一可用區域內啟動 Spot 執行個體

您可以透過將 Spot 選項 SingleAvailabilityZone 設定為 true，將機群設定為啟動單一可用區域內的所有執行個體。

12 個啟動範本覆寫會擁有不同執行個體類型和子網路 (每個都位於不同的可用區域)，但權重容量相同。總目標容量為 20 個執行個體，預設購買選項為 Spot，Spot 分配策略為容量最佳化。EC2 機群使用啟動規格，從具有最佳容量的 Spot 容量集區啟動全部在單一可用區域中的 20 個 Spot 執行個體。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 5：在單一可用區域內啟動單一執行個體類型的 Spot 執行個體

您可以透過將 SpotOptions SingleInstanceType 和 SingleAvailabilityZone 設定為 true，將機群設定為啟動單一可用區域內，相同執行個體類型的所有執行個體。

12 個啟動範本覆寫會擁有不同執行個體類型和子網路 (每個都位於不同的可用區域)，但權重容量相同。總目標容量為 20 個執行個體，預設購買選項為 Spot，Spot 分配策略為容量最佳化。EC2 機群使用啟動規格，從具有最佳容量的 Spot 容量集區啟動全部在單一可用區域中，相同執行個體類型的 20 個 Spot 執行個體。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
```

```
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-49e41922"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 6：僅在可啟動最小目標容量時才啟動 Spot 執行個體

只有透過將 Spot 選項 MinTargetCapacity 設定為要同時啟動的最小目標容量，以啟動最小目標容量，您才可以將機群設定為啟動執行個體。

指定 MinTargetCapacity 時，必須至少指定下列其中一個參數：SingleInstanceType 或 SingleAvailabilityZone。在此範例中，已指定 SingleInstanceType，因此所有 20 個執行個體都必須使用相同的執行個體類型。

12 個啟動範本覆寫會擁有不同執行個體類型和子網路 (每個都位於不同的可用區域)，但權重容量相同。總目標容量和最小目標容量設定為 20 個執行個體，預設購買選項為 Spot，並且 Spot 配置策略為容量最佳化。只有在能夠同時啟動全部 20 個執行個體時，EC2 機群才能使用啟動範本覆寫，從具有最優容量的 Spot 容量集區啟動 20 個 Spot 執行個體。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
```

```
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-49e41922"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 7：僅當可在單個可用區域中啟動相同執行個體類型的最低目標容量時，才能啟動 Spot 執行個體

只有透過將 Spot 選項 MinTargetCapacity 設定為要與 SingleInstanceType 和 SingleAvailabilityZone 選項同時啟動的最小目標容量，以在單個可用區域中，以單個執行個體類型啟動最小目標容量，您才可以將機群設定為啟動執行個體。

覆寫啟動範本的 12 個啟動規格擁有不同執行個體類型和子網路 (每個都位於不同的可用區域)，但權重容量相同。總目標容量和最小目標容量均設定為 20 個執行個體，預設購買選項為 Spot，Spot 分配策略為容量最佳化，SingleInstanceType 和 SingleAvailabilityZone 為 true。只有在能同時啟動全部 20

個執行個體時，EC2 機群才能使用啟動規格，從具有最佳容量的 Spot 容量集區啟動全部在單一可用區域中，相同執行個體類型的 20 個 Spot 執行個體。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
```

```
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5.4xlarge",
                "SubnetId":"subnet-49e41922"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-fae8c380"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-e7188bab"
            },
            {
                "InstanceType":"m5d.4xlarge",
                "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 8：使用多個啟動範本來啟動執行個體

您可以透過指定多個啟動範本，將機群設定為針對不同執行個體類型或執行個體類型群組，啟動具有不同啟動規格的執行個體。在這個範例中，我們希望針對不同執行個體類型，擁有不同 EBS 磁碟區大小，並且我們在啟動範本 ec2-fleet-lt-4xl、ec2-fleet-lt-9xl 和 ec2-fleet-lt-18xl 中進行設定。

在此範例中，我們根據大小，針對 3 個執行個體類型使用 3 種不同的啟動範本。所有啟動範本上的啟動規格覆寫會根據執行個體類型上的 vCPU，使用執行個體權重。總目標容量為 144 個單位，預設購買選項為 Spot，Spot 分配策略為容量最佳化。EC2 機群可以使用啟動範本 ec2-fleet-4xl 來啟動 9 個 c5n.4xlarge (144 除以 16)，或使用啟動範本 ec2-fleet-9xl 來啟動 4 個 c5n.9xlarge (144 除以 36)，或

使用啟動範本 ec2-fleet-18xl 來啟動 2 個 c5n.18xlarge (144 除以 72)，或基於容量最佳化分配策略，權重相加達到所需容量的執行個體類型的混合。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-18xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":72
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":72
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":72
            }
         ]
      },
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-9xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.9xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":36
            },
            {
```

```
                "InstanceType":"c5n.9xlarge",
                "SubnetId":"subnet-e7188bab",
                "WeightedCapacity":36
             },
             {
                "InstanceType":"c5n.9xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":36
             }
          ]
       },
       {
          "LaunchTemplateSpecification":{
             "LaunchTemplateName":"ec2-fleet-lt-4xl",
             "Version":"$Latest"
          },
          "Overrides":[
             {
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-fae8c380",
                "WeightedCapacity":16
             },
             {
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-e7188bab",
                "WeightedCapacity":16
             },
             {
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":16
             }
          ]
       }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 144,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 9：啟動基於隨需執行個體的 Spot 執行個體

下列範例指定機群 20 個執行個體的總目標容量，以及 5 個隨需執行個體的目標容量。預設購買選項為 Spot。機群依照指定啟動 5 個隨需執行個體，但需要再啟動 15 個執行個體才能實現總目標容量。差值購買選項的計算方式為 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType，這會導致啟動 15 個 Spot 執行個體的機群根據容量最佳化的分配策略，構成 12 個 Spot 容量集區之一。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
```

```
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 10：基於使用容量保留和優先順序分配策略的隨需執行個體，使用容量最佳化分配策略啟動 Spot 執行個體

您可以透過將容量保留的使用策略設為 use-capacity-reservations-first (首先使用容量保留)，來將機群設定為在以預設目標容量類型作為 Spot 啟動隨需執行個體的基礎時，首先使用按需容量保留。因此，如果多個執行個體集區有未使用的容量保留，則會套用所選擇的隨需分配策略。在此範例中，隨需分配策略為優先順序。

在此範例中，有 6 個未使用的容量保留可供使用。這小於機群的目標隨需容量 10 個隨需執行個體。

帳戶在 2 個集區中有以下 6 個未使用的容量保留。每個集區中的容量保留數目由 AvailableInstanceCount 表示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

下列機群組態僅顯示此範例的相關組態。隨需分配策略是優先順序，而容量保留的使用策略是 use-capacity-reservations-first (首先使用容量保留)。Spot 分配策略是容量最佳化。目標總容量為 20、隨需目標容量為 10、預設目標容量類型為 spot。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "OnDemandOptions":{
       "CapacityReservationOptions": {
         "UsageStrategy": "use-capacity-reservations-first"
       },
       "AllocationStrategy":"prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
```

```
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 3.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 4.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 5.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 6.0
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 7.0
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 8.0
            },
            {
```

```
                "InstanceType":"m5.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 9.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 10.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 11.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 12.0
            }
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 10,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

使用上述組態建立 instant 類型機群之後，會啟動以下 20 個執行個體以符合目標容量：

- us-east-1a 中的 7 個 c5.large 隨需執行個體 – us-east-1a 中的 c5.large 是第一優先順序，並有 3 個未使用 c5.large容量保留可供使用。容量保留會先用來啟動 3 個隨需執行個體，再根據隨需分配策略啟動額外的 4 個隨需執行個體，即此範例中的優先順序。
- us-east-1a 中的 3 個 m5.large 隨需執行個體 – us-east-1a 中的 m5.large 為第二優先，並有 3 個未使用的 m5.large 容量保留可用。
- 來自 12 個 Spot 容量集區中的 10 個 Spot 執行個體，該集區根據容量最佳化的分配策略具有最佳容量。

啟動機群後，您可以執行 describe-capacity-reservations，以查看還剩餘多少個未使用的容量保留。在此範例中，您應該會看到下列回應，其中顯示了已使用的所有 c5.large 和 m5.large 容量保留。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

範例 11：使用容量最佳化的優先順序分配策略啟動 Spot 執行個體

下列範例指定 instant 類型 EC2 機群中所需的參數：啟動範本、目標容量、預設購買選項，以及啟動範本覆寫。啟動範本由其啟動範本名稱和版本編號識別。覆寫啟動範本的 12 個啟動規格擁有已指派優先順序的 4 個不同執行個體類型和 3 個不同的子網路，每個都位於不同的可用區域。機群的目標容量是 20 個執行個體，預設購買選項為 Spot，這導致機群嘗試根據容量最佳化優先順序的分配策略，從 12 個 Spot 容量集區之一啟動 20 個 Spot 執行個體，該策略會竭盡全力實作優先順序，但會先針對容量最佳化。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
```

```
            "SubnetId":"subnet-e7188bab",
            "Priority": 1.0
         },
         {
            "InstanceType":"c5.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 1.0
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-fae8c380",
            "Priority": 2.0
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-e7188bab",
            "Priority": 2.0
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 2.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-fae8c380",
            "Priority": 3.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-e7188bab",
            "Priority": 3.0
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-49e41922",
            "Priority": 3.0
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-fae8c380",
            "Priority": 4.0
         },
         {
```

```
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 4.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 4.0
            }
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 12：指定 Systems Manager 參數而非 AMI ID

下列範例使用啟動範本來指定機群中執行個體的組態。在此範例中，針對 `ImageId`，要使用 System Manager 參數參考 AMI，而不是指定 AMI ID。啟動執行個體時，Systems Manager 參數會解析為 AMI ID。

在此範例中，Systems Manager 參數將以有效的格式指定：`resolve:ssm:golden-ami`。Systems Manager 參數還有其他有效的格式。如需詳細資訊，請參閱使用 Systems Manager 參數而非 AMI ID。

> **Note**
>
> 機群類型必須為 `instant`。其他機群類型不支援指定 System Manager 參數替代 AMI ID。

```
{
    "LaunchTemplateData": {
        "ImageId": "resolve:ssm:golden-ami",
        "InstanceType": "m5.4xlarge",
        "TagSpecifications": [{
            "ResourceType": "instance",
            "Tags": [{
                "Key": "Name",
```

```
                "Value": "webserver"
            }]
        }]
    }
}
```

設定 EC2 機群或 Spot 機群的花費限制

您可以設定限制，以指定您願意在 EC2 機群或 Spot 機群中的每小時花費。達到您的花費限制時，即使未達到目標容量，機群也會停止啟動執行個體。

隨需執行個體和 Spot 執行個體有不同的花費限制。

在 EC2 機群中設定隨需執行個體和 Spot 執行個體的花費限制

使用 create-fleet 命令和下列參數：

- 對於隨需執行個體：在 `OnDemandOptions` 結構中，在 `MaxTotalPrice` 欄位中指定您的花費限制。
- 對於 Spot 執行個體：在 `SpotOptions` 結構中，在 `MaxTotalPrice` 欄位中指定您的花費限制。

在 Spot 機群中設定隨需執行個體和 Spot 執行個體的花費限制

您可以使用 Amazon EC2 主控台或 AWS CLI 來設定您的花費限制。

(主控台) 建立 Spot 機群時，請選取設定 Spot 執行個體成本上限核取方塊，然後輸入設定成本上限的值 (每小時)。如需詳細資訊，請參閱 使用已定義的參數建立 Spot 機群請求 中的步驟 6.e.。

(AWS CLI) 使用 request-spot-fleet 命令和下列參數：

- 對於隨需執行個體：在 `OnDemandMaxTotalPrice` 欄位中指定您的花費限制。
- 對於 Spot 執行個體：在 `SpotMaxTotalPrice` 欄位中指定您的花費限制。

範例

以下範例顯示兩個不同的案例。在第一個範例中，機群在達到隨需執行個體 (`OnDemandTargetCapacity`) 的目標容量設定時，會停止啟動隨需執行個體。在第二個範例中，機群會在達到您願意支付的隨需執行個體每小時金額上限 (`MaxTotalPrice`) 時停止啟動隨需執行個體。

範例：達到目標容量時停止啟動隨需執行個體

假設有 m4.large 隨需執行個體 的請求，其中：

- 隨需價格：每小時 0.10 USD
- OnDemandTargetCapacity：10
- MaxTotalPrice：1.50 USD

機群會啟動 10 個隨需執行個體，因為 1.00USD 的總計 (10 個執行個體 x 0.10USD) 未超出隨需執行個體 1.50USD 的 MaxTotalPrice。

範例：達到總價格上限時停止啟動隨需執行個體

假設有 m4.large 隨需執行個體 的請求，其中：

- 隨需價格：每小時 0.10 USD
- OnDemandTargetCapacity：10
- MaxTotalPrice：0.80 USD

如果機群啟動了隨需目標容量 (10 個隨需執行個體)，每個小時的總成本會是 1.00USD。這超出為隨需執行個體指定的 MaxTotalPrice 的金額 (0.80 USD)。為了防止花費超出您願意支付的金額，機群只會啟動 8 個隨需執行個體 (低於隨需目標容量)，因為啟動更多會超出隨需執行個體的 MaxTotalPrice。

爆量效能執行個體

如果您使用爆量效能執行個體啟動 Spot 執行個體，而且您計劃立即使用爆量效能 Spot 執行個體，且在短時間內沒有閒置時間來累積 CPU 額度，建議您以標準模式將其啟用，以避免支付更高的成本。如果您在無限制模式下啟動高載效能 Spot 執行個體，而且 CPU 立即高載，則您將耗費高載的剩餘額度。如果您短暫使用執行個體，則您的執行個體沒有時間產生 CPU 額度來償還剩餘額度，而且當終止執行個體時，會針對剩餘額度向您收費。

只有在執行個體執行夠長時間，可為爆量產生 CPU 額度時，無限制模式才適合用於高載效能 Spot 執行個體。否則，支付剩餘額度會使爆量效能 Spot 執行個體比使用其他執行個體更為昂貴。如需詳細資訊，請參閱使用無限制模式與固定 CPU 的時機。

啟動額度旨在透過提供足夠的運算資源來設定執行個體，創造 T2 執行個體具有生產力的初始啟動體驗。不允許重複啟動 T2 執行個體來獲得新的啟動額度。如果您需要持續的 CPU，可以賺取額度 (藉由在某些期間閒置)、為 T2 Spot 執行個體使用無限制模式，或將執行個體類型與專屬 CPU 搭配使用。

指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型

建立 EC2 機群或 Spot 機群時，您必須指定一或多個執行個體類型，以在機群中設定隨需執行個體和 Spot 執行個體。做為手動指定執行個體類型的替代方式，您可以指定執行個體必須具有的屬性，然後 Amazon EC2 會識別具有這些屬性的所有執行個體類型。這就是所謂的屬性型執行個體類型選項。例如，您可以指定執行個體所需的 vCPU 數目下限和上限，然後機群將使用符合這些 vCPU 需求的任何可用執行個體類型來啟動執行個體。

屬性型執行個體類型選擇非常適合可對其使用哪些執行個體類型具有彈性的工作負載和架構，例如執行容器或 Web 機群、處理大數據，以及實作持續整合和部署 (CI/CD) 工具時。

優勢

屬性型執行個體類型選擇具有下列優勢：

- 輕鬆使用正確的執行個體類型：由於有太多可用的執行個體類型，為您的工作負載尋找適當的執行個體類型可能會非常耗時。當您指定執行個體類型屬性時，執行個體類型會自動具有工作負載所需的屬性。
- 簡化的組態：若要針對機群手動指定多個執行個體類型，您必須為每個執行個體類型建立個別的啟動範本覆寫。但是，有了屬性型執行個體類型選擇，若要提供多個執行個體類型，您只需要在啟動範本或在啟動範本覆寫中指定執行個體屬性。
- 自動使用新的執行個體類型：當您指定執行個體屬性而非執行個體類型時，您的機群可以在新一代的執行個體類型釋出時使用它們，即「為未來提供」機群的組態。
- 執行個體類型彈性：當您指定執行個體屬性而非執行個體類型時，機群可以從用於啟動 Spot 執行個體的廣泛執行個體類型中進行選擇，而這些執行個體類型符合 Spot 執行個體類型靈活性的最佳實務。

主題

- 屬性型執行個體類型選擇的運作方式
- 價格保護
- 效能保護
- 考量事項
- 建立具有屬性型執行個體類型選擇的 EC2 機群
- 建立具有屬性型執行個體類型選擇的 Spot 機群
- 有效和無效的 EC2 機群組態範例

- 有效和無效的 Spot 機群組態範例
- 預覽具有所指定屬性的執行個體類型

屬性型執行個體類型選擇的運作方式

若要在機群組態中使用屬性型執行個體類型選擇，請將執行個體類型清單取代為執行個體所需的執行個體屬性清單。EC2 機群或 Spot 機群將在具有所指定執行個體屬性的任何可用執行個體類型上啟動執行個體。

主題

- 執行個體屬性的類型
- 要在何處設定屬性型執行個體類型選擇
- EC2 機群或 Spot 機群在佈建機群時如何使用屬性型執行個體類型選擇

執行個體屬性的類型

有數個您可以指定來表達運算需求的執行個體屬性，例如：

- vCPU 數量：每個執行個體的 vCPU 數量下限和上限。
- 記憶體：每個執行個體的記憶體 GiB 下限和上限。
- 本機儲存：是否使用本機儲存的 EBS 或執行個體儲存體磁碟區。
- 爆量效能：是否使用 T 執行個體系列，包括 T4g、T3a、T3和 T2 類型。

如需每個屬性和預設值的描述，請參閱 Amazon EC2 API 參考中的 InstanceRequirements。

要在何處設定屬性型執行個體類型選擇

視您使用的是 主控台或 而定 AWS CLI，您可以指定屬性型執行個體類型選擇的執行個體屬性，如下所示：

在主控台中，您可以在下列機群組態元件中指定執行個體屬性：

- 在啟動範本中，則參考機群請求中的啟動範本
- (僅限 Spot 機群) 在機群請求中

在 中 AWS CLI，您可以在下列一或多個機群組態元件中指定執行個體屬性：

- 在啟動範本中，則參考機群請求中的啟動範本
- 在啟動範本覆寫中

  如果想要混合使用不同 AMI 的執行個體，您可以在多個啟動範本覆寫中指定執行個體屬性。例如，不同的執行個體類型可以使用 x86 和 ARM 型處理器。
- (僅限 Spot 機群) 在啟動規格中

EC2 機群或 Spot 機群在佈建機群時如何使用屬性型執行個體類型選擇

EC2 機群或 Spot 機群以下列方式佈建機群：

- EC2 機群可識別具有所指定屬性的執行個體類型。
- EC2 機群會使用價格保護來決定要排除哪些執行個體類型。
- 它會根據具有相符執行個體類型的 AWS 區域或可用區域，決定要考慮從中啟動執行個體的容量集區。
- EC2 機群會套用指定的分配策略，以決定要從哪些容量集區中啟動執行個體。

  請注意，屬性型執行個體類型選擇不會挑選要從中佈建機群的容量集區；這是分配策略的任務。

  如果您指定分配策略，機群將根據指定的分配策略啟動執行個體。
  - 對於 Spot 執行個體，屬性型執行個體類型選擇支援價格容量最佳化、容量最佳化和最低價格分配策略。請注意，我們不建議最低價格的 Spot 配置策略，因為該策略具有最高的 Spot 執行個體中斷風險。
  - 若為隨需執行個體，屬性型執行個體類型選擇支援最低價格分配策略。
- 如果具有所指定執行個體屬性的執行個體類型沒有容量，則無法啟動任何執行個體，而且機群會傳回錯誤。

價格保護

價格保護是一項功能，可防止 EC2 機群或 Spot 機群使用您認為過於昂貴的執行個體類型，即使它們剛好符合您指定的屬性也一樣。若要使用價格保護，您必須設定價格閾值。隨後，當 Amazon EC2 選取具有您屬性的執行個體類型時，其會排除定價高於閾值的執行個體類型。

Amazon EC2 計算價格閾值的方式如下：

- Amazon EC2 會先從符合您屬性的執行個體類型中，識別價格最低的執行個體類型。

- 然後，Amazon EC2 會取得您為價格保護參數指定的值 (以百分比表示)，並將其乘以已識別執行個體類型的價格。結果即為價格閾值的價格。

隨需執行個體和 Spot 執行個體有不同的價格閾值。

當您使用屬性型執行個體類型選擇建立機群時，價格保護將預設啟用。您可以保留預設值，也可以指定自己的值。

您也可以關閉價格保護。若要表示沒有價格保護閾值，請指定高百分比值，例如 999999。

主題

- 如何識別價格最低的執行個體類型
- 隨需執行個體價格保護
- Spot 執行個體價格保護
- 指定價格保護閾值

如何識別價格最低的執行個體類型

Amazon EC2 會從符合您指定屬性的執行個體中，找出價格最低的執行個體類型，從而決定價格門檻所依據的價格。Amazon EC2 會透過下列方式執行此操作：

- 首先，查看符合您屬性的目前世代 C、M 或 R 執行個體類型。如果找到任何相符項目，則會識別價格最低的執行個體類型。
- 如果沒有相符項目，則會查看任何符合您屬性的目前世代執行個體類型。如果找到任何相符項目，則會識別價格最低的執行個體類型。
- 如果沒有相符項目，則會查看任何符合您屬性的上一代執行個體類型，並識別價格最低的執行個體類型。

隨需執行個體價格保護

隨需執行個體類型的價格保護閾值的計算方式，是高於已識別最低價格隨需執行個體類型 (`OnDemandMaxPricePercentageOverLowestPrice`) 的百分比。您可以指定您願意支付的百分比。如果您未指定此參數，則會使用 `20` 的預設值來計算比已識別價格高 20% 的價格保護閾值。

例如，如果識別的隨需執行個體價格為 `0.4271`，而您指定 `25`，則價格閾值比 `0.4271` 多 25%。計算方式如下：`0.4271 * 1.25 = 0.533875`。計算的價格是您願意為隨需執行個體支付的最高金額，在此範例中，Amazon EC2 會排除任何超過 `0.533875` 的隨需執行個體類型。

Spot 執行個體價格保護

根據預設，Amazon EC2 會自動套用最佳 Spot 執行個體價格保護，以一致方式從各種執行個體類型中選擇。您也可以自行手動設定價格保護。不過，若您讓 Amazon EC2 為您執行此操作，則可以提高 Spot 容量被滿足的可能性。

您可以使用下列其中一個選項手動指定價格保護。如果您手動設定價格保護，我們建議您使用第一個選項。

- 已識別價格最低的隨需執行個體類型百分比 [MaxSpotPriceAsPercentageOfOptimalOnDemandPrice]

  例如，如果識別的隨需執行個體類型價格為 0.4271，而您指定 60，則價格閾為 0.4271 的 60%。計算方式如下：0.4271 * 0.60 = 0.25626。計算的價格是您願意為 Spot 執行個體支付的最高金額，在此範例中，Amazon EC2 會排除任何超過 0.25626 的 Spot 執行個體類型。
- 高於已識別價格最低的 Spot 執行個體類型百分比 [SpotMaxPricePercentageOverLowestPrice]

  例如，如果識別的 Spot 執行個體類型價格為 0.1808，而您指定 25，則價格閾值比 0.1808 多 25%。計算方式如下：0.1808 * 1.25 = 0.226。計算的價格是您願意為 Spot 執行個體支付的最高金額，在此範例中，Amazon EC2 會排除任何超過 0.266 的 Spot 執行個體類型。我們不建議使用此參數，因為 Spot 價格可能會波動，因此您的價格保護閾值也可能會波動。

指定價格保護閾值

使用 指定價格保護閾值 AWS CLI

使用 建立 EC2 機群或 Spot 機群時 AWS CLI，請針對屬性型執行個體類型選取設定機群，然後執行下列動作：

- 如要指定隨需執行個體價格保護閾值，請於 JSON 組態檔案的 InstanceRequirements 結構中，對於 OnDemandMaxPricePercentageOverLowestPrice，以百分比形式輸入價格保護閾值。
- 若要指定 Spot 執行個體價格保護閾值，請在 JSON 組態檔案中的 InstanceRequirements 結構中，指定下列其中一個參數：
  - 若為 MaxSpotPriceAsPercentageOfOptimalOnDemandPrice，請以百分比形式輸入價格保護閾值。
  - 若為 SpotMaxPricePercentageOverLowestPrice，請以百分比形式輸入價格保護閾值。

如需詳細資訊，請參閱 建立具有屬性型執行個體類型選擇的 EC2 機群 或 建立具有屬性型執行個體類型選擇的 Spot 機群 。

(僅限 Spot機群) 使用主控台指定價格保護閾值

使用主控台建立 Spot 機群時，為屬性型執行個體類型選擇設定機群，然後執行下列作業：

- 如要指定隨需執行個體價格保護閾值，請在其他執行個體屬性下，選擇隨需價格保護，然後選擇新增屬性，然後輸入價格保護閾值 (百分比)。
- 如要指定 Spot 執行個體價格保護閾值，請在其他執行個體屬性，選擇 Spot 價格保護，然後選擇新增屬性，選擇一個基本值，以作為您的價格依據，然後輸入價格保護閾值 (百分比)。

> **Note**
>
> 建立機群時，若您將 `TargetCapacityUnitType` 設定為 `vcpu` 或 `memory-mib`，則價格保護閾值會根據每個 vCPU 或每個記憶體的價格進行套用，而非每個執行個體的價格。

效能保護

效能保護功能可確保 EC2 機群或 Spot 機群使用類似或超過指定效能基準的執行個體類型。若要使用效能保護，您可以將執行個體系列指定為基準參考。指定執行個體系列的功能會建立最低可接受效能層級。當 Amazon EC2 為您的機群選取執行個體類型時，會考慮您指定的屬性和效能基準。低於效能基準的執行個體類型，將自動排除在選取範圍之外，即使這些類型符合您的其他指定屬性。這可確保所有選取的執行個體類型所提供的效能，與指定執行個體系列所建立的基準類似或更好。Amazon EC2 將使用此基準來引導執行個體類型選擇，但選取的執行個體類型不一定會超過每個應用程式的基準。

目前，此功能僅支援以 CPU 效能作為基準效能因素。指定執行個體系列的 CPU 處理器的 CPU 效能，可以做為效能基準，確保選取的執行個體類型類似於或超過此基準。具有相同 CPU 處理器的執行個體系列會產生相同的篩選結果，即使其網路或磁碟效能不同。例如，指定 `c6in` 或 `c6i` 作為基準參考，會產生相同的效能型篩選結果，因為兩個執行個體系列都使用相同的 CPU 處理器。

不支援的執行個體系列

效能保護不支援下列執行個體系列：

- 一般用途：Mac1 | Mac2 | Mac2-m1ultra | Mac2-m2 | Mac2-m2pro | M1 | M2 | T1
- 運算最佳化：C1

- 記憶體最佳化：U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | u-24tb1 | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb
- 加速運算：G3 | G3s | P3dn | P4d | P5
- 高效能運算：Hpc7g

如果您透過指定支援的執行個體系列來啟用效能保護，則傳回的執行個體類型將排除上述不支援的執行個體系列。

如果您將不支援的執行個體系列指定為基準效能值，API 會傳回 GetInstanceTypesFromInstanceRequirements 的空白回應，並傳回 CreateFleet、RequestSpotFleet、ModifyFleet 和 ModifySpotFleetRequest 的例外狀況。

範例：設定 CPU 效能基準

在下面的示例中，執行個體需求是使用執行個體類型啟動，該執行個體類型有與 c6i 執行個體系列相同的 CPU 核心性能。這將篩選效能較低的 CPU 處理器的執行個體類型，即使它們符合您的其他指定執行個體需求，例如 vCPU 數量。例如，如果您指定的執行個體屬性包含 4 個 vCPU 和 16GB 的記憶體，則具有這些屬性但 CPU 效能低於 c6i 的執行個體類型將被排除在選擇之外。

```
"BaselinePerformanceFactors": {
        "Cpu": {
            "References": [
                {
                    "InstanceFamily": "c6i"
                }
            ]
        }
```

考量事項

- 您可以在 EC2 機群或 Spot 機群中指定執行個體類型或執行個體屬性，但不能同時指定兩者。

  使用 CLI 時，啟動範本覆寫將覆寫啟動範本。例如，如果啟動範本包含執行個體類型，且啟動範本覆寫包含執行個體屬性，則執行個體屬性所識別的執行個體將覆寫啟動範本中的執行個體類型。
- 使用 CLI 時，若將執行個體屬性指定為覆寫，您也無法指定權重或優先順序。
- 您最多可以在請求組態中指定四個 InstanceRequirements 結構。

建立具有屬性型執行個體類型選擇的 EC2 機群

您可將 EC2 機群設定為使用屬性型執行個體類型選擇。無法使用 Amazon EC2 主控台建立 EC2 機群。

屬性型執行個體類型選取的屬性在 `InstanceRequirements` 結構中予以指定。當 `InstanceRequirements` 併入機群組態中時，必須排除 `InstanceType` 和 `WeightedCapacity`；其無法與執行個體屬性同時決定機群組態。

在 AWS CLI 和 PowerShell 範例中，會指定下列屬性：

- `VCpuCount` – 最少 2 個 vCPU，最多 4 個 vCPU。若無需設定上限，可省略最大值。
- `MemoryMiB` – 最小 8 GiB 的記憶體，最大 16 GiB。若無需設定上限，可省略最大值。

此組態可識別所有具備 2 至 4 個 vCPU 與 8 至 16 GiB 記憶體的執行個體類型。但是，當 EC2 機群佈建機群時，價格保護和分配策略可能會排除某些執行個體類型。

AWS CLI

若要建立具有屬性型執行個體類型選取的 EC2 機群

使用 create-fleet 命令建立 EC2 機群。指定 JSON 檔案中的機群組態。

```
aws ec2 create-fleet \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

下列範例 *file_name*.json 檔案包含將 EC2 機群設定為使用屬性型執行個體類型選取的參數。

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized"
    },
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
```

```
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

PowerShell

若要建立具有屬性型執行個體類型選取的 EC2 機群

使用 New-EC2Fleet cmdlet。

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRangeRequest
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiBRequest
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirementsRequest
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object
 Amazon.EC2.Model.FleetLaunchTemplateSpecificationRequest
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.FleetLaunchTemplateOverridesRequest
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.FleetLaunchTemplateConfigRequest
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

New-EC2Fleet `
```

```
    -SpotOptions_AllocationStrategy "price-capacity-optimized" `
    -LaunchTemplateConfig @($launchTemplateConfig) `
    -TargetCapacitySpecification_DefaultTargetCapacityType "spot" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -Type "instant"
```

建立具有屬性型執行個體類型選擇的 Spot 機群

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

您可以將機群設定為使用屬性型執行個體類型選擇。

屬性型執行個體類型選取的屬性在 `InstanceRequirements` 結構中予以指定。當 `InstanceRequirements` 併入機群組態中時，必須排除 `InstanceType` 和 `WeightedCapacity`；其無法與執行個體屬性同時決定機群組態。

在 AWS CLI 和 PowerShell 範例中，會指定下列屬性：

- `VCpuCount` – 最少 2 個 vCPU，最多 4 個 vCPU。若無需設定上限，可省略最大值。
- `MemoryMiB` – 最小 8 GiB 的記憶體，最大 16 GiB。若無需設定上限，可省略最大值。

此組態可識別所有具備 2 至 4 個 vCPU 與 8 至 16 GiB 記憶體的執行個體類型。但是，當 Spot 機群佈建機群時，價格保護和分配策略可能會排除某些執行個體類型。

Console

若要為屬性型執行個體類型選取設定 Spot 機群

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Spot Requests (Spot 請求)，然後選擇 Request Spot Instances (請求 Spot 執行個體)。
3. 遵循步驟來建立 Spot 機群。如需詳細資訊，請參閱 使用已定義的參數建立 Spot 機群請求。

   在建立 Spot 機群時，為屬性型執行個體類型選擇設定機群，如下所示：

a. 對於 Instance type requirements (執行個體類型請求)，選擇 Specify instance attributes that match your compute requirements (指定符合運算需求的執行個體屬性)。

b. 對於 vCPUs，輸入所需的 vCPU 數量下限和上限。若要指定無限制，請選取 No minimum (無下限)、No maximum (無上限)，或兩者。

c. 對於 Memory (GiB) (記憶體 (GiB))，輸入所需記憶體數量的下限和上限。若要指定無限制，請選取 No minimum (無下限)、No maximum (無上限)，或兩者。

d. (選用) 對於 Additional instance attributes (其他執行個體屬性)，您可以選擇性地指定一或多個屬性，以更詳細地表達您的運算需求。每個額外屬性都會將進一步的限制新增至您的請求。

e. (選用) 展開 Preview matching instance types (預覽相符的執行個體類型)，以檢視具有所指定屬性的執行個體類型。

AWS CLI

若要為屬性型執行個體類型選取設定 Spot 機群

使用 request-spot-fleet 命令建立 Spot 機群。指定 JSON 檔案中的機群組態。

```
aws ec2 request-spot-fleet \
    --region us-east-1 \
    --spot-fleet-request-config file://file_name.json
```

下列範例 *file_name*.json 檔案包含將 Spot 機群設定為使用屬性型執行個體類型選取的參數。

```
{
    "AllocationStrategy": "priceCapacityOptimized",
    "TargetCapacity": 20,
    "Type": "request",
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
```

```
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }]
}
```

PowerShell

若要為屬性型執行個體類型選取設定 Spot 機群

使用 Request-EC2SpotFleet cmdlet。

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRange
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiB
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirements
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object Amazon.EC2.Model.FleetLaunchTemplateSpecification
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

Request-EC2SpotFleet `
    -SpotFleetRequestConfig_AllocationStrategy "PriceCapacityOptimized" `
    -SpotFleetRequestConfig_TargetCapacity 20 `
    -SpotFleetRequestConfig_Type "Request" `
    -SpotFleetRequestConfig_LaunchTemplateConfig $launchTemplateConfig
```

有效和無效的 EC2 機群組態範例

如果您使用 AWS CLI 建立 EC2 機群，則必須確保您的機群組態有效。下列範例顯示有效和無效的組態。

當組態包含下列項目時，視為無效：

- 同時具有 `Overrides` 和 `InstanceRequirements` 的單一 `InstanceType`
- 兩個 `Overrides` 結構，一個具有 `InstanceRequirements`，而另一個具有 `InstanceType`
- 兩個在同一 `InstanceRequirements` 內具有重疊屬性值的 `LaunchTemplateSpecification` 結構

範例組態

- 有效組態：具有覆寫的單一啟動範本
- 有效組態：具有多個 InstanceRequirements 的單一啟動範本
- 有效組態：兩個啟動範本，每個都具有覆寫
- 有效組態：僅限指定的 InstanceRequirements，沒有重疊的屬性值
- 組態無效：Overrides 包含 InstanceRequirements 和 InstanceType
- 組態無效：兩個 Overrides 包含 InstanceRequirements 和 InstanceType
- 組態無效：重疊屬性值

有效組態：具有覆寫的單一啟動範本

下列組態有效。其包含一個啟動範本和一個 `Overrides` 結構 (其中包含一個 `InstanceRequirements` 結構)。範例組態的文字說明如下。

```
{
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
```

```
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5000,
        "DefaultTargetCapacityType": "spot",
        "TargetCapacityUnitType": "vcpu"
        }
    }
}
```

InstanceRequirements

若要使用屬性型執行個體選擇，您必須在組態中包含 InstanceRequirements 結構，並為機群中的執行個體指定所需的屬性。

在上述範例中，指定了下列執行個體屬性：

- VCpuCount – 執行個體類型必須具有最少 2 個，最多 8 個的 vCPU。
- MemoryMiB – 執行個體類型必須具有最多 10240 MiB 的記憶體。最小值若為 0，表示沒有最小限制。
- MemoryGiBPerVCpu – 執行個體類型必須具有每個 vCPU 最多 10,000 MiB 的記憶體。Min 為選用參數。若省略它，表示沒有最小限制。

TargetCapacityUnitType

TargetCapacityUnitType 參數會指定目標容量的單位。在範例中，目標容量為 5000，而目標容量單位類型為 vcpu，它們一起指定了所需的目標容量，即 5,000 個 vCPU。EC2 機群將啟動足夠的執行個體，以便機群中的 vCPU 總數為 5,000 個 vCPU。

有效組態：具有多個 InstanceRequirements 的單一啟動範本

下列組態有效。其包含一個啟動範本和一個 Overrides 結構 (其中包含兩個 InstanceRequirements 結構)。InstanceRequirements 中指定的屬性是有效的，因為這些值不重疊—第一個 InstanceRequirements 結構會指定 0-2 個 vCPU 的VCpuCount，而第二個 InstanceRequirements 結構則會指定 4-8 個 vCPU。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
```

```
        }
    }
}
```

有效組態：兩個啟動範本，每個都具有覆寫

下列組態有效。其包含兩個啟動範本，每個都具有一個 Overrides 結構 (其中包含一個 InstanceRequirements 結構)。這個組態有助於同一個機群中的 arm 和 x86 架構支援。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
```

```
                    }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

有效組態：僅限指定的 **InstanceRequirements**，沒有重疊的屬性值

下列組態有效。其包含兩個 LaunchTemplateSpecification 結構，每個都有一個啟動範本和一個 Overrides 結構 (其中包含一個 InstanceRequirements 結構)。InstanceRequirements 中指定的屬性是有效的，因為這些值不重疊—第一個 InstanceRequirements 結構會指定 0-2 個 vCPU 的VCpuCount，而第二個 InstanceRequirements 結構則會指定 4-8 個 vCPU。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
```

```
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

組態無效：Overrides 包含 InstanceRequirements 和 InstanceType

下列組態無效。Overrides 結構同時包含 InstanceRequirements 和 InstanceType。針對 Overrides，您可以指定 InstanceRequirements 或 InstanceType，但不能同時指定兩者。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
```

```
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

組態無效：兩個 Overrides 包含 InstanceRequirements 和 InstanceType

下列組態無效。Overrides 結構同時包含 InstanceRequirements 和 InstanceType。您可以指定 InstanceRequirements 或 InstanceType，但不能同時指定兩者，即使它們處於不同的 Overrides 結構也一樣。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
```

```
            ]
        },
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyOtherLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceType": "m5.large"
            }
          ]
        }
    ],
     "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
  }
}
```

組態無效：重疊屬性值

下列組態無效。兩個 InstanceRequirements 結構每個都包含 "VCpuCount": {"Min": 0, "Max": 2}。這些屬性的值重疊，這會導致容量集區重複。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceRequirements": {
                    "VCpuCount": {
                        "Min": 0,
                        "Max": 2
                    },
                    "MemoryMiB": {
                        "Min": 0
                    }
```

```
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

有效和無效的 Spot 機群組態範例

如果您使用 AWS CLI 建立 Spot 機群，則必須確保您的機群組態有效。下列範例顯示有效和無效的組態。

當組態包含下列項目時，視為無效：

- 同時具有 `Overrides` 和 `InstanceRequirements` 的單一 `InstanceType`
- 兩個 `Overrides` 結構，一個具有 `InstanceRequirements`，而另一個具有 `InstanceType`
- 兩個在同一 `InstanceRequirements` 內具有重疊屬性值的 `LaunchTemplateSpecification` 結構

範例組態

- 有效組態：具有覆寫的單一啟動範本
- 有效組態：具有多個 InstanceRequirements 的單一啟動範本
- 有效組態：兩個啟動範本，每個都具有覆寫
- 有效組態：僅限指定的 InstanceRequirements，沒有重疊的屬性值

- 組態無效：Overrides 包含 InstanceRequirements 和 InstanceType
- 組態無效：兩個 Overrides 包含 InstanceRequirements 和 InstanceType
- 組態無效：重疊屬性值

有效組態：具有覆寫的單一啟動範本

下列組態有效。其包含一個啟動範本和一個 Overrides 結構 (其中包含一個 InstanceRequirements 結構)。範例組態的文字說明如下。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
        "TargetCapacity": 5000,
```

```
            "OnDemandTargetCapacity": 0,
            "TargetCapacityUnitType": "vcpu"
    }
}
```

InstanceRequirements

若要使用屬性型執行個體選擇，您必須在組態中包含 InstanceRequirements 結構，並為機群中的執行個體指定所需的屬性。

在上述範例中，指定了下列執行個體屬性：

- VCpuCount – 執行個體類型必須具有最少 2 個，最多 8 個的 vCPU。
- MemoryMiB – 執行個體類型必須具有最多 10240 MiB 的記憶體。最小值若為 0，表示沒有最小限制。
- MemoryGiBPerVCpu – 執行個體類型必須具有每個 vCPU 最多 10,000 MiB 的記憶體。Min 為選用參數。若省略它，表示沒有最小限制。

TargetCapacityUnitType

TargetCapacityUnitType 參數會指定目標容量的單位。在範例中，目標容量為 5000，而目標容量單位類型為 vcpu，它們一起指定了所需的目標容量，即 5,000 個 vCPU。Spot 機群將啟動足夠的執行個體，以便機群中的 vCPU 總數為 5,000 個 vCPU。

有效組態：具有多個 InstanceRequirements 的單一啟動範本

下列組態有效。其包含一個啟動範本和一個 Overrides 結構 (其中包含兩個 InstanceRequirements 結構)。InstanceRequirements 中指定的屬性是有效的，因為這些值不重疊—第一個 InstanceRequirements 結構會指定 0-2 個 vCPU 的VCpuCount，而第二個 InstanceRequirements 結構則會指定 4-8 個 vCPU。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
```

```
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

有效組態：兩個啟動範本，每個都具有覆寫

下列組態有效。其包含兩個啟動範本，每個都具有一個 Overrides 結構 (其中包含一個 InstanceRequirements 結構)。這個組態有助於同一個機群中的 arm 和 x86 架構支援。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
```

```
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
```

```
}
```

有效組態：僅限指定的 **InstanceRequirements**，沒有重疊的屬性值

下列組態有效。其包含兩個 LaunchTemplateSpecification 結構，每個都有一個啟動範本和一個 Overrides 結構 (其中包含一個 InstanceRequirements 結構)。InstanceRequirements 中指定的屬性是有效的，因為這些值不重疊—第一個 InstanceRequirements 結構會指定 0-2 個 vCPU 的VCpuCount，而第二個 InstanceRequirements 結構則會指定 4-8 個 vCPU。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
```

```
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

組態無效：**Overrides** 包含 **InstanceRequirements** 和 **InstanceType**

下列組態無效。Overrides 結構同時包含 InstanceRequirements 和 InstanceType。針對 Overrides，您可以指定 InstanceRequirements 或 InstanceType，但不能同時指定兩者。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
```

```
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

組態無效：兩個 Overrides 包含 InstanceRequirements 和 InstanceType

下列組態無效。Overrides 結構同時包含 InstanceRequirements 和 InstanceType。您可以指定 InstanceRequirements 或 InstanceType，但不能同時指定兩者，即使它們處於不同的 Overrides 結構也一樣。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
```

```
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

組態無效：重疊屬性值

下列組態無效。兩個 InstanceRequirements 結構每個都包含 "VCpuCount": {"Min": 0, "Max": 2}。這些屬性的值重疊，這會導致容量集區重複。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
```

```
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                    }
                ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

預覽具有所指定屬性的執行個體類型

您可以使用 get-instance-types-from-instance-requirements 命令，來預覽符合您指定之屬性的執行個體類型。這特別有助於解決在不啟動任何執行個體的情況下要在請求組態中指定哪些屬性。請注意，該命令不會考慮可用容量。

使用 指定屬性來預覽執行個體類型的清單 AWS CLI

1. (選用) 若要產生可指定的所有可能屬性，請使用 get-instance-types-from-instance-requirements 命令和 `--generate-cli-skeleton` 參數。您可以選擇性地將輸出導向至檔案來儲存它，方法是使用 `input > attributes.json`。

```
aws ec2 get-instance-types-from-instance-requirements \
```

```
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

預期的輸出結果

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "i386"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "included",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
            "Min": 0,
            "Max": 0
```

```
        },
        "LocalStorage": "included",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "gpu"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "nvidia"
        ],
        "AcceleratorNames": [
            "a100"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "NetworkBandwidthGbps": {
            "Min": 0.0,
            "Max": 0.0
        },
        "AllowedInstanceTypes": [
            ""
        ]
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

2. 使用上一個步驟的輸出來建立 JSON 組態檔，並將其設定如下：

> **Note**
>
> 您必須提供 ArchitectureTypes、VirtualizationTypes、VCpuCount 和 MemoryMiB 的值。您可以省略其他屬性；省略時，就會使用預設值。
> 如需每個屬性及其預設值的說明，請參閱 get-instance-types-from-instance-requirements。

a. 針對 ArchitectureTypes，指定一或多種類型的處理器架構。

b. 針對 VirtualizationTypes，指定一或多種類型的虛擬化。

c. 針對 VCpuCount，指定 vCPU 數目下限和上限。若要指定沒有下限，請針對 Min，指定 0。若要指定沒有上限，請省略 Max 參數。

d. 針對 MemoryMiB，指定記憶體的數量下限和上限 (以 MiB 為單位)。若要指定沒有下限，請針對 Min，指定 0。若要指定沒有上限，請省略 Max 參數。

e. 您可以選擇性地指定一或多個其他屬性，以進一步限制傳回的執行個體類型清單。

3. 若要預覽具有您在 JSON 檔案中指定之屬性的執行個體類型，請使用 get-instance-types-from-instance-requirements 命令，並使用 --cli-input-json 參數指定 JSON 檔案的名稱和路徑。您可以選擇性地格式化要以表格格式出現的輸出。

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

範例 *attributes.json* 文件

在此範例中，JSON 檔案中包含必要的屬性。它們是 ArchitectureTypes、VirtualizationTypes、VCpuCount 和 MemoryMiB。此外，選用 InstanceGenerations 屬性也包含在內。請注意，對於 MemoryMiB，可以省略 Max，以表示沒有限制。

```
{

    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
```

```
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

範例輸出

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||             InstanceTypes             ||
|+-------------------------------------+|
||             InstanceType              ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
||  c5n.xlarge                         ||
||  d2.xlarge                          ||
...
```

4. 在識別符合您需求的執行個體類型之後，請記下您使用的執行個體屬性，以便在設定機群請求時可以使用它們。

使用執行個體權重來管理 EC2 機群或 Spot 機群的成本和效能

執行個體加權，可讓您為 EC2 機群或 Spot 機群中的每個執行個體類型指派權重，以代表其運算容量和效能。根據權重，機群可以使用指定執行個體類型的任意組合，只要機群可以滿足所需的目標容量即可。這可協助您管理機群的成本和效能。

權重代表執行個體類型為總目標容量貢獻的容量單位。

範例：使用執行個體加權進行效能管理

假設您的機群有兩種執行個體類型，您可以為每個執行個體類型指派不同的權重，以反映每個執行個體類型達到相同的效能所需的權重，如下所示：

- m5.large – 權重：1
- m5.2xlarge – 權重：4

指派以上權重，表示您需要 4 個 m5.large 執行個體才能達到與 1 個 m5.2xlarge 相同的效能。

若要計算指定目標容量需要多少執行個體類型的執行個體，請使用下列公式：

```
target capacity / weight = number of instances
```

如果您的目標容量為 8 個單位，機群可以使用 m5.large 或 m5.2xlarge 達成目標容量，或兩者混合，如下所示：

- 8 個 m5.large 執行個體 (容量為 8/權重為 1 = 8 個執行個體)
- 2 個 m5.2xlarge 執行個體 (容量為 8/權重為 4 = 2 個執行個體)
- 4 個 m5.large 和 1 個 m5.2xlarge

範例：使用執行個體權重進行成本管理

根據預設，您指定的價格是每執行個體小時。使用執行個體權重功能時，您指定的價格是每單位小時。您可以透過將執行個體類型價格除以它所代表的單位數，來計算每單位小時的價格。機群會將目標容量除以執行個體的權重，來計算出要啟動的執行個體數量。如果結果不是整數，則機群會將其捨入到下一個整數，以便您的機群大小不低於其目標容量。即使所啟動的執行個體容量超過所請求的目標容量，該機群也可以選取您在啟動規格中指定的任何集區。

下表包含決定目標容量為 10 的機群每單位價格之計算範例。

執行個體類型	執行個體權重	目標容量	已啟動的執行個體數目	每個執行個體小時的價格	每個單位小時的價格
r3.xlarge	2	10	5	0.05 USD	$0.025

執行個體類型	執行個體權重	目標容量	已啟動的執行個體數目	每個執行個體小時的價格	每個單位小時的價格
			(10 除以 2)		(.05 除以 2)
r3.8xlarge	8	10	2 (10 除以 8，將結果四捨五入)	0.10 USD	$0.0125 (.10 除以 8)

按照下列方式，在實現請求時使用機群執行個體權重，以在完成要求時，使用每單位最低價在集區中佈建所需目標容量：

1. 在您所選擇的執行個體 (預設) 或裝置中 (例如 vCPU、記憶體、儲存空間或輸送量)，設定機群的目標容量。
2. 設定每單位價格
3. 為每個啟動規格指定權重，即執行個體類型代表目標容量的單位數。

執行個體權重範例

請考慮下列組態的機群請求：

- 目標容量為 24
- 執行個體類型為 r3.2xlarge 且權重為 6 的啟動規格
- 執行個體類型為 c3.xlarge 且權重為 5 的啟動規格

權重表示執行個體類型代表目標容量的單位數。如果第一個啟動規格提供最低的每單位價格 (r3.2xlarge 的每小時執行個體價格除以 6)，機群將啟動其中四個執行個體 (24 除以 6)。

如果第二個啟動規格提供了最低的每單位價格 (c3.xlarge 的每小時執行個體價格除以 5)，則機群會啟動這些執行個體的其中 5 個 (24 除以 5，將結果四捨五入)。

執行個體權重與分配策略

請考慮下列組態的機群請求：

- 目標容量為 30 個 Spot 執行個體
- 執行個體類型為 `c3.2xlarge` 且權重為 8 的啟動規格
- 執行個體類型為 `m3.xlarge` 且權重為 8 的啟動規格
- 執行個體類型為 `r3.xlarge` 且權重為 8 的啟動規格

機群會啟動 4 個執行個體 (30 除以 8，將結果四捨五入)。採用 `diversified` 策略，機群會在三個集區的每一個集區啟動一個執行個體，並且三個集區任一個中的第四個執行個體會提供最低的每單位價格。

使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量

當您在 EC2 機群或 Spot 機群中使用多個容量集區 (每個集區都包含執行個體類型和可用區域) 時，您可以使用配置策略來管理 Amazon EC2 如何從這些集區達成 Spot 和隨需容量。配置策略可以最佳化可用容量、價格，以及要使用的執行個體類型。Spot 執行個體和隨需執行個體有不同的配置策略。

主題

- Spot 執行個體的分配策略
- 隨需執行個體的配置策略
- 選擇適當的 Spot 分配策略
- 維持 Spot 執行個體的目標容量
- 針對隨需容量排定執行個體類型的優先順序

Spot 執行個體的分配策略

您的啟動組態會決定可啟動 Spot 執行個體的 EC2 機群或 Spot 機群，及其所有可能 Spot 容量集區 (執行個體類型和可用區域)。但是，啟動執行個體時，機群會使用您指定的配置策略，從所有可能的集區中挑選特定的集區。

> **Note**
>
> (僅 Linux 執行個體) 如果設定 Spot 執行個體為在開啟 AMD SEV-SNP 的情形下啟動，則需要支付額外的小時用量費，相當於所選執行個體類型隨需小時費率的 10%。如果配置策略使用價格作為輸入，則機群不會包含此額外費用；僅使用 Spot 價格。

您可以為 Spot 執行個體指定下列其中一個配置策略：

最佳化的價格容量 (建議)

機群會識別擁有啟動中執行個體數目之最高容量可用性的集區。這表示會從很可能在短期內中斷可能性最低的集區請求 Spot 執行個體。機群接著會從這些最低價集區中請求 Spot 執行個體。

最佳化的價格容量分配策略是大多數 Spot 工作負載的最佳選擇，例如無狀態容器化應用程式、微服務、Web 應用程式、資料和分析作業以及批次處理。

如果您使用的是 AWS CLI，則參數名稱`price-capacity-optimized`適用於 EC2 機群，`priceCapacityOptimized`適用於 Spot 機群。

容量最佳化

機群會識別擁有啟動中執行個體數目之最高容量可用性的集區。這表示會從很可能在短期內中斷可能性最低的集區請求 Spot 執行個體。

使用 Spot 執行個體，定價會依據長期的供需趨勢隨時間慢慢改變，但是容量會即時波動。容量最佳化策略會查看即時容量資料並預測哪些資料為最多可用，從而自動將 Spot 執行個體啟動到最可用的集區。這適用於可能具有較高的重新啟動工作中斷成本的工作負載，例如持續整合 (CI)、影像和媒體轉譯、深度學習以及可能具有較高的重新啟動工作中斷成本的高效能運算 (HPC) 工作負載。容量最佳化策略可實現盡可能少的中斷，因此可降低您工作負載的整體成本。

或者，您可以使用具有優先順序參數的容量最佳化優先順序分配策略，以從最高到最低的優先順序對執行個體類型排序。您可以為不同的執行個體類型設定相同的優先順序。機群會先針對容量進行最佳化，但會盡力遵守執行個體類型的優先順序 (例如，如果遵守優先順序不會顯著影響機群佈建最佳容量的能力)。對於必須將中斷可能性降至最低的工作負載來說，這是一個很好的選擇，而且某些執行個體類型的偏好也很重要。請注意，當您為 Spot 容量設定執行個體類型的優先順序時，如果隨需執行個體分配策略設定為優先順序，則相同的優先順序也會套用在隨需執行個體上。使用 Spot 機群時，只有在您的機群使用啟動範本時，才支援使用優先順序。

如果您使用的是 AWS CLI，則 EC2 機群`capacity-optimized-prioritized`的參數名稱為 `capacity-optimized`和 ，Spot 機群`capacityOptimizedPrioritized`的參數名稱為 `capacityOptimized`和 。

多樣化

Spot 執行個體會分佈在所有 Spot 容量集區。如果您使用的是 AWS CLI，則參數名稱`diversified`適用於 EC2 機群和 Spot 機群。

最低價格 (不建議)

> **Warning**
> 我們不建議最低價格的配置策略，因為該策略具有最高的 Spot 執行個體中斷風險。

Spot 執行個體來自價格最低且還有可用容量的集區。使用 時 AWS CLI，這是預設策略。不過，建議您透過指定最佳化的價格容量分配策略來覆寫預設。

使用最低價格策略時，如果價格最低的集區沒有可用容量，Spot 執行個體來自下一個價格最低且還有可用容量的集區。如果集區在滿足您的所需容量之前用完容量，機群將繼續從下一個價格最低的集區中選取集區來滿足您的請求。為確保達到所需的容量，您可能會從數個集區中接收 Spot 執行個體。

由於此策略僅考慮執行個體價格，而不考慮容量可用性，因此可能會導致高中斷率。

只有在使用 AWS CLI時，才能使用最低價格分配策略。EC2 機群參數名稱為 `lowest-price`，而 Spot 機群則是 `lowestPrice`。

要使用的集區數量

要配置目標 Spot 容量的 Spot 集區數量。僅在配置策略設定為最低價格時有效。機群會選取價格最低的 Spot 集區，並將目標 Spot 容量平均配置到您指定數量的 Spot 集區上。

請注意，機群會盡最大努力從您指定的集區數量中提取 Spot 執行個體。如果集區在滿足您的目標容量之前用完 Spot 容量，則機群將透過從下一個價格最低的集區中進行提取來繼續滿足您的請求。為確保達到目標容量，您可能會從超過您指定的集區數量中接收 Spot 執行個體。同樣地，如果大多數集區沒有 Spot 容量，您可能會從少於您指定的集區數量中接收完整目標容量。

此參數僅在指定最低價格分配策略時可用，且僅在使用 AWS CLI時可用。EC2 機群和 Spot 機群的參數名稱皆為 `InstancePoolsToUseCount`。

隨需執行個體的配置策略

您的啟動組態會決定 EC2 機群或 Spot 機群可從中啟動隨需執行個體的所有可能容量集區 (執行個體類型和可用區域)。但是，啟動執行個體時，機群會使用您指定的配置策略，從所有可能的集區中挑選特定的集區。

您可以為隨需執行個體指定下列其中一個配置策略：

最低價格

隨需執行個體個體來自價格最低且還有可用容量的集區。這是預設策略。

如果價格最低的集區沒有可用容量，隨需執行個體來自下一個價格最低且具有可用容量的集區。

如果集區在滿足您的所需容量之前用完容量，機群將繼續從下一個價格最低的集區中選取集區來滿足您的請求。為確保達到所需的容量，您可能會從數個集區中接收隨需執行個體。

優先順序

機群會使用您指派給每個啟動範本覆寫的優先順序，從優先順序最高的項目開始啟動執行個體類型。此策略無法與屬性型執行個體類型選取搭配使用。如需如何使用此配置策略的範例，請參閱 針對隨需容量排定執行個體類型的優先順序。

選擇適當的 Spot 分配策略

您可以選擇適當的 Spot 分配策略，進而針對您的使用案例最佳化機群。

平衡最低價格與容量可用性

為了平衡價格最低的 Spot 容量集區與具有最高容量可用性的 Spot 容量集區之間的利弊，建議您使用最佳化的價格容量分配策略。此策略會根據集區的價格和這些集區中 Spot 執行個體的容量可用性，決定要從哪些集區請求 Spot 執行個體的集區。這表示 Spot 執行個體是從很可能在短期內中斷可能性最低的集區請求，同時仍然考慮價格。

如果您的機群執行彈性且無狀態的工作負載，包括容器化應用程式、微服務、Web 應用程式、資料和分析作業以及批次處理，則使用最佳化的價格容量分配策略，以達到最佳的成本節省和容量可用性。

如果您的機群執行的工作負載有較高的重新啟動工作中斷的成本，則您應該實作檢查點作業，以便應用程式可以在中斷時從該點重新啟動。透過使用檢查點作業，您可以讓最佳化的價格容量分配策略非常適合這些工作負載，因為其可以從價格最低的集區中分配容量，而這些集區也會提供較低的 Spot 執行個體中斷率。

如需使用價格容量最佳化配置策略的 JSON 組態範例，請參閱下列內容：

- EC2 機群：範例 10：在 price-capacity-optimized 機群中啟動 Spot 執行個體
- Spot 機群：範例 11：在 priceCapacityOptimized 機群中啟動 Spot 執行個體

當工作負載具有較高的中斷成本時

如果您執行的工作負載使用價格相似的執行個體類型，或是中斷成本非常之高，以致於任何成本節省都不足以彌補中斷的邊際增加，則您可以選擇是否使用該容量最佳化策略。該策略可從最可用的 Spot 容量集區分配容量，而這些集區會提供較少中斷的可能性，進而降低您工作負載的整體成本。

當必須將中斷可能性降至最低，而且對於某些執行個體類型的偏好也很重要時，您可以使用容量最佳化優先順序分配策略，然後設定執行個體類型的順序，來表現您的集區優先順序，以從最高到最低優先順序使用。

請注意，當您設定容量最佳化優先順序的優先順序時，如果隨需執行個體設定為優先順序，則相同的優先順序也會套用至您的隨需執行個體。另請注意，使用 Spot 機群時，只有在您的機群使用啟動範本時，才支援使用優先順序。

如需使用容量最佳化配置策略的 JSON 組態範例，請參閱以下內容：

- EC2 機群：範例 8：在容量最佳化機群中啟動 Spot 執行個體
- Spot 機群：範例 9：在容量最佳化機群中啟動 Spot 執行個體

如需使用容量最佳化優先順序配置策略的 JSON 組態範例，請參閱以下內容：

- EC2 機群：範例 9：使用優先順序在容量最佳化機群中啟動 Spot 執行個體
- Spot 機群：範例 10：使用優先順序在容量最佳化機群中啟動 Spot 執行個體

當您的工作負載具有時間彈性且容量可用性並非相關因素時

如果您的機群很小或執行時間很短，您仍可在考慮容量可用性的同時，使用價格最佳化優先順序來實現成本節省最大化。

當您的機群很龐大或執行時間較長時

如果您的機群很龐大或執行時間較長，則可以使用多樣化策略，透過將多個 Spot 執行個體分散到多個集區中來提高機群的可用性。例如，如果機群指定了 10 個集區和 100 個執行個體的目標容量，則該機群會在每個集區中啟動 10 個 Spot 執行個體。如果某個集區的 Spot 價格超過該集區的最高價格，則只有 10% 的機群會受到影響。隨著時間，使用這種策略也會使您的機群對任何一個集區 Spot 價格上漲較不敏感。使用多樣化策略，Spot 機群不會將 Spot 執行個體啟動至任何等於或高於隨需價格的 Spot 價格集區中。

維持 Spot 執行個體的目標容量

在 Spot 執行個體因 Spot 價格或 Spot 容量集區可用容量的變更而終止之後，類型為 maintain 的機群會啟動替代 Spot 執行個體。分配策略會確定從中啟動替代執行個體的集區，如下所示：

- 如果分配策略是最佳化的價格容量，則機群會在擁有最多 Spot 執行個體容量可用性的集區中啟動替代執行個體，同時將價格納入考慮並識別價格最低且具有高容量可用性的執行個體。
- 如果分配策略為最佳化容量，則機群會在擁有最多 Spot 執行個體容量可用性的集區中啟動替代執行個體。
- 如果分配策略為多樣化，則機群會將替代 Spot 執行個體分配到剩餘的集區中。

針對隨需容量排定執行個體類型的優先順序

EC2 機群或 Spot 機群嘗試實現您的隨需容量時，它會預設為先啟動價格最低的執行個體類型。若隨需分配策略設為優先，則機群會使用優先順序來決定在實現隨需容量時要先使用哪一種執行個體類型。優先順序會指派至啟動範本覆寫，優先順序最高的將會先啟動。

範例：排定執行個體類型的優先順序

在本例中，您設定三個啟動範本覆寫，每個範本均具有不同的執行個體類型。

執行個體類型的隨需價格在價格範圍內。以下是本範例中使用的執行個體類型，依價格順序列出，從最便宜的執行個體類型開始：

- m4.large – 最便宜
- m5.large
- m5a.large

若您不使用優先順序來決定順序，則機群會以最便宜的執行個體類型開始來實現隨需容量。

但是，假設您有要首先使用且未使用的 m5.large 預留執行個體。您可設定啟動範本覆寫，以便依優先順序使用執行個體類型，如下所示：

- m5.large – 優先順序 1
- m4.large – 優先順序 2
- m5a.large – 優先順序 3

在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體

透過容量重新平衡，您的 EC2 機群或 Spot 機群可以主動取代具有中斷風險的 Spot 執行個體，進而維持所需的 Spot 容量。當 Spot 執行個體的中斷風險提高時，Amazon EC2 會傳送重新平衡建議。如果啟用容量重新平衡，重新平衡建議會在風險執行個體中斷之前，觸發新的 Spot 執行個體啟動。

容量重新平衡可協助您維持工作負載可用性，方法是在執行中的執行個體受到 Amazon EC2 的中斷之前，使用新的 Spot 執行個體主動擴增機群。

設定 EC2 機群，以使用容量重新平衡，進而啟動取代 Spot 執行個體

使用 create-fleet 命令和 `MaintenanceStrategies` 結構中的相關參數。如需範例 JSON 組態檔，請參閱 範例 7：設定容量重新平衡，以啟動取代 Spot 執行個體。

設定 Spot 機群，以使用容量重新平衡，進而啟動取代 Spot 執行個體

您可以使用 Amazon EC2 主控台或 AWS CLI 來設定容量重新平衡。

(主控台) 在您建立 Spot 機群時，選取容量重新平衡核取方塊。如需詳細資訊，請參閱 使用已定義的參數建立 Spot 機群請求 中的步驟 6.d.。

(AWS CLI) 使用 request-spot-fleet 命令和 `SpotMaintenanceStrategies` 結構中的相關參數。如需範例 JSON 組態檔，請參閱 範例 8：設定容量重新平衡，以啟動取代 Spot 執行個體。

主題

- 限制
- 組態選項
- 考量事項

限制

- 容量重新平衡僅適用於類型為 `maintain` 的機群。
- 機群執行時，您無法修改容量重新平衡設定。若要變更容量重新平衡設定，您必須刪除機群並建立新的機群。

組態選項

EC2 機群和 Spot 機群的 `ReplacementStrategy` 支援以下兩個值：

`launch-before-terminate`

Amazon EC2 會在新的替代 Spot 執行個體啟動之後，終止收到重新平衡通知的 Spot 執行個體。當指定 `launch-before-terminate` 時，您亦須指定 `termination-delay` 的值。在啟動新的替代執行個體之後，Amazon EC2 會等待 `termination-delay` 的持續時間，然後終止舊的執行個體。對於 `termination-delay`，下限為 120 秒 (2 分鐘)，上限則為 7200 秒 (2 小時)。

建議僅在您可以預測執行個體關閉程序將需要多長時間才能完成時，才使用 `launch-before-terminate`。這將確保只在關機程序完成之後，才會終止舊執行個體。請注意，Amazon EC2 可以中斷舊的執行個體，並在 `termination-delay` 之前提供兩分鐘警告。

強烈建議您不要使用 `lowest-price` (EC2 機群) 或 `lowestPrice` (Spot 機群) 分配策略與 `launch-before-terminate` 結合，以避免替代 Spot 執行個體也處於提高的中斷風險。

`launch`

針對現有 Spot 執行個體發出重新平衡通知時，Amazon EC2 會啟動替代 Spot 執行個體。Amazon EC2 不會自動終止收到重新平衡建議的執行個體。您可以終止舊的執行個體，也可以讓它們保持執行中狀態。所有執行個體在執行時，您需要支付所有執行個體的費用。

考量事項

如果您設定容量重新平衡的 EC2 機群或 Spot 機群，請考慮下列事項：

盡可能在請求中提供更多 Spot 容量集區

設定您的機群，以使用多個執行個體類型和可用區域。這提供了在各種 Spot 容量集區中啟動 Spot 執行個體的彈性。如需詳細資訊，請參閱 對執行個體類型和可用區域具有彈性。

避免中斷替代 Spot 執行個體的風險提高

為了避免提高中斷風險，我們強烈建議使用 `capacity-optimized` 或 `capacity-optimized-prioritized` 配置策略。這些策略可確保在最佳 Spot 容量集區中啟動替代 Spot 執行個體，因此不太可能在不久的將來中斷。如需詳細資訊，請參閱使用價格和容量最佳化分配策略。

如果您使用 `lowest-price` 配置策略，替代 Spot 執行個體的中斷風險可能會提高。這是因為 Amazon EC2 總是會在當時具有可用容量的最低價集區中啟動執行個體，即使您的替代 Spot 執行個體在啟動後很快就會中斷也一樣。

只有在可用性相同或更好時，Amazon EC2 才會啟動新執行個體

容量重新平衡的其中一個目標是改善 Spot 執行個體的可用性。如果現有的 Spot 執行個體收到重新平衡建議，只有新執行個體提供與現有執行個體相同或更好的可用性時，Amazon EC2 才會啟動新

執行個體。如果新執行個體的中斷風險比現有執行個體更大，則 Amazon EC2 將不會啟動新執行個體。但是，Amazon EC2 將繼續評估 Spot 容量集區，並在可用性改善時啟動新執行個體。

如果 Amazon EC2 不主動啟動新執行個體，現有執行個體可能會中斷。發生這種情況時，無論新執行個體是否存在高中斷風險，Amazon EC2 都會嘗試啟動新執行個體。

容量重新平衡不會增加 Spot 執行個體中斷率

當您啟用容量重新平衡時，不會增加 Spot 執行個體中斷率 (當 Amazon EC2 需要恢復容量時回收的 Spot 執行個體數量)。但是，如果容量重新平衡偵測到執行個體有中斷風險，Amazon EC2 將立即嘗試啟動新的執行個體。因此，與在有風險的執行個體中斷後等待 Amazon EC2 啟動新的執行個體相比，可能會取代更多執行個體。

雖然您可以在啟用容量重新平衡的情況下取代更多執行個體，但您可以從主動式而非被動式中受益，因為在執行個體中斷之前您有更多時間採取動作。透過 Spot 執行個體中斷通知，您通常最多只有兩分鐘時間來正常關閉您的執行個體。透過容量重新平衡預先啟動新的執行個體，可讓現有程序更有可能在有風險的執行個體上完成，您可以啟動執行個體關閉程序，並防止在有風險的執行個體上排程新的工作。您也可以開始準備新啟動的執行個體以接管應用程式。透過容量重新平衡的主動式取代，您可以從正常連續性中獲益。

作為示範使用容量重新平衡的風險和益處的理論範例，請考慮下列案例：

- 下午 2:00 – 收到執行個體 A 的重新平衡建議，Amazon EC2 會立即開始嘗試啟動取代執行個體 B，讓您有時間開始關機程序。*
- 下午 2:30 – 收到執行個體 B 的重新平衡建議，會取代為執行個體 C，讓您有時間開始關機程序。*
- 下午 2:32 – 如果未啟用容量重新平衡，並且如果在下午 2:32 收到執行個體 A 的 Spot 執行個體中斷通知，則您最多只有兩分鐘時間採取動作，但是執行個體 A 會一直執行到目前。

* 如果指定了 `launch-before-terminate`，Amazon EC2 將在取代執行個體連線後終止有風險的執行個體。

Amazon EC2 可以啟動新的替代 Spot 執行個體，直到滿足的容量為目標容量雙倍

設定機群為容量重新平衡時，機群會嘗試為每個收到重新平衡建議的 Spot 執行個體啟動新的替代 Spot 執行個體。Spot 執行個體收到重新平衡建議後，該執行個體不再計入以滿足容量的一部分。根據替代策略，Amazon EC2 會在預先設定的終止延遲之後終止執行個體，或讓執行個體繼續執行。這讓您有機會在執行個體上執行重新平衡動作。

如果您的機群達到目標容量的兩倍，即使替代執行個體本身收到重新平衡建議，它仍會停止啟動新的替代執行個體。

例如，您建立目標容量為 100 個 Spot 執行個體的機群。所有 Spot 執行個體都會收到重新平衡建議，這會導致 Amazon EC2 啟動 100 個替代 Spot 執行個體。這會將已滿足 Spot 執行個體的數目提高至 200 個，這是目標容量的兩倍。某些替代執行個體會收到重新平衡建議，但不會再啟動替代執行個體，因為機群不能超過目標容量的兩倍。

請注意，所有執行個體在執行時都需支付這些執行個體的費用。

建議您將機群設定為終止收到重新平衡建議的 Spot 執行個體

如果您設定機群進行容量重新平衡，建議您只在可以預測執行個體關閉程序需要多長時間才能完成時，才選擇具有適當終止延遲的 `launch-before-terminate`。這將確保只在關機程序完成之後，才會終止舊執行個體。

如果您選擇自行終止建議用於重新平衡的執行個體，建議您監控機群中 Spot 執行個體收到的重新平衡建議訊號。透過監控訊號，您可以在 Amazon EC2 中斷受影響的執行個體之前，快速對受影響的執行個體執行重新平衡動作，然後手動終止它們。如果您未終止執行個體，則會在執行時繼續為這些執行個體付費。Amazon EC2 不會自動終止收到重新平衡建議的執行個體。

您可以使用 Amazon EventBridge 或執行個體中繼資料來設定通知。如需詳細資訊，請參閱監控重新平衡建議訊號。

機群在縮減或擴展期間計算已滿足容量時，不會計算收到重新平衡建議的執行個體

如果您的機群已設定容量重新平衡，且您將目標容量變更為縮減或擴展，則機群不會將標示要重新平衡的執行個體計算為已滿足容量一部分，如下所示：

- 縮減 – 如果您減少所需的目標容量，Amazon EC2 會終止未標示要重新平衡的執行個體，直到達到所需容量為止。標示要重新平衡的執行個體不會計入已滿足的容量。

  例如，您建立目標容量為 100 個 Spot 執行個體的機群。10 個執行個體會收到重新平衡建議，因此 Amazon EC2 會啟動 10 個新的替代執行個體，從而產生 110 個執行個體的滿足容量。然後，您可以將目標容量減少為 50 (縮減)，但已滿足容量實際上為 60 個執行個體，因為標示要重新平衡的 10 個執行個體不會被 Amazon EC2 終止。您需要手動終止這些執行個體，或者您也可以讓它們保持運行。
- 橫向擴展 – 如果您增加想要的目標容量，Amazon EC2 會啟動新執行個體，直到達到所需容量為止。標示要重新平衡的執行個體不會計入已滿足的容量。

  例如，您建立目標容量為 100 個 Spot 執行個體的機群。10 個執行個體會收到重新平衡建議，因此機群會啟動 10 個新的替代執行個體，從而產生 110 個執行個體的滿足容量。然後，您將目標容量增加至 200 (擴展)，但已滿足容量實際上是 210 個執行個體，因為標示要重新平衡的 10 個執行個體不會被機群計為目標容量的一部分。您需要手動終止這些執行個體，或者您也可以讓它們保持運行。

使用容量保留功能來保留 EC2 機群中的隨需容量

隨需容量保留可讓您在特定的可用區域中，為隨需執行個體預訂任何期間的運算容量。您可將 EC2 機群設定為在啟動隨需執行個體時先使用容量保留。

隨需容量保留僅適用於請求類型設為 `instant` 的 EC2 機群。

容量保留設定為 `open` 或 `targeted`。EC2 機群可以按照 `open` 或 `targeted` 的容量保留啟動隨需執行個體：

- 如果容量保留為 `open`，則具有相符屬性的隨需執行個體會以預留容量自動執行。
- 如果容量保留為 `targeted`，則隨需執行個體必須特別針對它使用預留容量執行。這對於用完特定容量保留或控制何時該使用特定容量保留的情況下十分有用。

如果您在 EC2 機群中使用 `targeted` 容量保留，必須有足夠的容量保留以滿足目標隨需容量，否則啟動會失敗。為了避免啟動失敗，最好將 `targeted` 容量保留新增至資源群組，然後針對資源群組為目標來執行。資源群組不需要有足夠的容量保留；如果在滿足目標隨需容量之前用完容量保留，則機群可以將剩餘的目標容量啟動為一般隨需容量。

在 EC2 機群中使用容量保留

1. 將機群配置為 `instant` 類型。您無法針對其他類型的機群使用容量保留。
2. 將容量保留的使用策略配置為 `use-capacity-reservations-first`。
3. 在啟動範本的 Capacity reservation (容量保留) 中，選擇 Open (開放) 或 Target by group (依群組分類的目標)。如果您選擇 Target by group (依群組分類的目標)，請指定容量保留資源群組 ID。

當機群嘗試滿足隨需容量時，如果發現多個執行個體集區具有相符的未使用容量保留，則會根據隨需分配策略 (`lowest-price` 或 `prioritized`) 決定啟動隨需執行個體的集區。

相關資源

- 如需設定機群以使用容量保留來滿足隨需容量的 CLI 範例，請參閱 EC2 機群的範例 CLI 組態 中的範例 5 至 7。
- 如需教學課程，引導您完成建立容量保留、在機群中使用，以及檢視剩餘容量保留數量的步驟，請參閱 教學課程：使用目標容量保留設定 EC2 機群啟動隨需執行個體
- 您也可以設定 EC2 機群，使用單一容量保留資源群組跨多個容量保留類型啟動執行個體：隨需容量保留、ML 容量區塊和中斷容量保留，並選擇性地回復至隨需容量。若要執行此作業，請使

用 ReservedCapacityOptions搭配DefaultTargetCapacityType設定為 的 reserved-capacity。如需教學，請參閱教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體。

- 如需設定容量保留的相關資訊，請參閱 使用 EC2 隨需容量保留來保留運算容量 與隨需容量保留常見問答集。

使用 EC2 機群

若要開始使用 EC2 機群，請建立請求，其中包含總目標容量、隨需容量、Spot 容量和指定機群中執行個體組態的啟動範本。您可以選擇性地指定其他參數，或讓機群使用預設值。您在建立機群時，也可以標記機群請求及其執行個體和磁碟區。

機群會在有可用容量時啟動隨需執行個體，並在最高價格超過 Spot 價格且容量可用時，啟動 Spot 執行個體。

機群啟動後，您可以描述機群請求、機群中的執行個體，以及任何機群事件。您也可以視需要指派其他標籤。

如果您需要變更任何機群參數 (例如總目標容量)，您可以修改機群，前提是其已設定為維持容量。提交之後，您就無法修改一次性請求的容量。

機群請求會一直保持作用中狀態，直到過期或您將其刪除為止。當您刪除機群請求時，可以終止執行個體，或讓執行個體保持執行中狀態。如果您選擇讓執行個體保持執行中狀態，隨需執行個體會執行直到您將其終止為止，而 Spot 執行個體則會一直執行直到中斷或您將其終止為止。

主題

- EC2 Fleet 請求狀態
- EC2 Fleet 先決條件
- 建立 EC2 Fleet
- 標記新的或現有的 EC2 機群請求，及其啟動的執行個體和磁碟區
- 描述 EC2 機群、其執行個體，及其事件
- 修改 EC2 Fleet
- 刪除 EC2 機群請求和機群中的執行個體

EC2 Fleet 請求狀態

EC2 機群請求可以是各種狀態之一，每個狀態都會指出請求生命週期的不同階段，以及其執行個體管理。

EC2 Fleet 請求可以是下列任一狀態：

`submitted`

正在評估 EC2 Fleet 請求且 Amazon EC2 正準備啟動執行個體的目標數量。超出機群限制的請求會立即刪除。

`active`

EC2 Fleet 請求已驗證且 Amazon EC2 正在嘗試維持執行中的執行個體目標數目。此請求會一直維持在該狀態，直到修改或刪除。

`modifying`

正在修改 EC2 Fleet 請求。此請求會一直維持在該狀態，直到已完成修改或已刪除請求。只有 `maintain` 機群類型可以修改。此狀態不適用於其他請求類型。

`deleted_running`

EC2 機群請求已刪除，且不會啟動其他 Spot 執行個體。其現有執行個體將繼續執行直到中斷或手動終止。此請求會一直維持在該狀態，直到所有執行個體中斷或終止。刪除 EC2 Fleet 請求後，只有類型 `maintain` 或 `request` 的 EC2 Fleet 可具有執行中的執行個體。不支援具有執行中執行個體的已刪除 `instant` 機群。此狀態不適用於 `instant` 機群。

`deleted_terminating`

EC2 Fleet 請求已刪除，且其執行個體已終止。此請求會一直維持在該狀態，直到所有執行個體終止。

`deleted`

EC2 機群 請求已刪除，且無執行中的執行個體。請求會在其執行個體終止之後兩天刪除。

EC2 Fleet 先決條件

若要建立 EC2 Fleet，必須具備下列必要條件：

- 啟動範本

- EC2 Fleet 的服務連結角色
- 授予客戶受管金鑰存取權，以便與加密的 AMI 和 EBS 快照搭配使用
- EC2 Fleet 使用者的許可

啟動範本

啟動範本會指定所要啟動執行個體的組態資訊，例如執行個體類型和可用區域。如需啟動範本的詳細資訊，請參閱 將執行個體啟動參數存放在 Amazon EC2 啟動範本中。

EC2 Fleet 的服務連結角色

`AWSServiceRoleForEC2Fleet` 角色會授與 EC2 Fleet 許可，以代表您請求、啟動、終止和標記執行個體。Amazon EC2 會使用此服務連結角色來完成下列動作：

- `ec2:RunInstances` – 啟動執行個體。
- `ec2:RequestSpotInstances` – 請求 競價型執行個體。
- `ec2:TerminateInstances` – 終止執行個體。
- `ec2:DescribeImages` - 描述執行個體的 Amazon Machine Images (AMI)。
- `ec2:DescribeInstanceStatus` - 描述執行個體的狀態。
- `ec2:DescribeSubnets` - 描述執行個體的子網絡。
- `ec2:CreateTags` – 將標籤新增至 EC2 Fleet、執行個體和磁碟區。

在使用 AWS CLI 或 API 建立 EC2 機群之前，請確定此角色存在。

> **Note**
>
> `instant` EC2 Fleet 不需要此角色。

若要建立角色，請依照下列所示使用 IAM 主控台。

為 EC2 Fleet 建立 AWSServiceRoleForEC2Fleet 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。

3. 選擇 Create Role (建立角色)。
4. 在 Select trusted entity (選取信任的實體) 頁面上，執行以下作業：

   a. 針對信任的實體類型，請選擇 AWS 服務。

   b. 在使用案例下，針對服務或使用案例，選擇 EC2 - 機群。

   > **Tip**
   >
   > 請務必選擇 EC2 - 機群。如果您選擇 EC2，則 EC2 - 機群使用案例不會出現在使用案例清單中。EC2 - 機群使用案例會自動建立具有必要 IAM 許可的政策，並將建議 AWSServiceRoleForEC2Fleet 做為角色名稱。

   c. 選擇下一步。
5. 在 Add permissions (新增許可) 頁面上，選擇 Next (下一步)。
6. 在命名、檢閱和建立頁面上，選擇建立角色。

如果您不再需要使用 EC2 Fleet，建議您刪除 AWSServiceRoleForEC2Fleet 角色。從您的帳戶中刪除此角色後，如果您建立其他機群，則可以再次建立此角色。

如需詳細資訊，請參閱「IAM 使用者指南」中的服務連結角色。

授予客戶受管金鑰存取權，以便與加密的 AMI 和 EBS 快照搭配使用

如果您在 EC2 機群中指定加密的 AMI 或加密的 Amazon EBS 快照，並使用 AWS KMS 金鑰進行加密，則必須授予 AWSServiceRoleForEC2Fleet 角色使用客戶受管金鑰的許可，以便 Amazon EC2 可以代表您啟動執行個體。若要執行此動作，您必須將授予新增至客戶受管金鑰，如下列程序所示。

提供許可時，授與為金鑰政策的替代方案。如需詳細資訊，請參閱「AWS Key Management Service 開發人員指南」中的使用授與和在 AWS KMS中使用金鑰政策。

若要授予 AWSServiceRoleForEC2Fleet 角色許可以使用客戶受管金鑰

- 使用 create-grant 命令將授予新增至客戶受管金鑰，並指定獲授予許可能夠執行授予所允許作業的主體 (AWSServiceRoleForEC2Fleet 服務連結角色)。客戶受管金鑰由 `key-id` 參數和客戶受管金鑰的 ARN 指定。委託人是透過 `grantee-principal` 參數和 AWSServiceRoleForEC2Fleet 服務連結角色的 ARN 指定。

```
aws kms create-grant \
    --region us-east-1 \
```

```
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

EC2 Fleet 使用者的許可

如果使用者將會建立或管理 EC2 Fleet，請務必授予這些使用者所需的許可。

為 EC2 Fleet 建立政策

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇政策。
3. 選擇 Create policy (建立政策)。
4. 在 Create policy (建立政策) 頁面上，選擇 JSON 索引標籤，接著使用下列內容來取代其文字，然後選擇 Review policy (檢閱政策)。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
              "iam:ListRoles",
              "iam:PassRole",
              "iam:ListInstanceProfiles"
            ],
            "Resource":"arn:aws:iam::123456789012:role/DevTeam*"
        }
```

```
    ]
}
```

ec2:* 會授與使用者權限，以呼叫所有 Amazon EC2 API 動作。若要將使用者限制於特定的 Amazon EC2 API 動作，請改為指定這些動作。

使用者必須具有呼叫下列動作的權限：iam:ListRoles 動作 (以列舉現有的 IAM 角色)、iam:PassRole 動作 (以指定 EC2 Fleet 角色) 和 iam:ListInstanceProfiles 動作 (以列舉現有的執行個體設定檔)。

(選用) 若要讓使用者能夠使用 IAM 主控台建立角色或執行個體設定檔，您也必須將下列動作新增至政策中：

- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:GetRole
- iam:ListPolicies

5. 在 Review policy (檢閱政策) 頁面上，輸入政策名稱和說明，然後選擇 Create policy (建立政策)。
6. 若要提供存取權，請新增權限至您的使用者、群組或角色：

   - 中的使用者和群組 AWS IAM Identity Center：

     建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。

   - 透過身分提供者在 IAM 中管理的使用者：

     建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。

   - IAM 使用者：

     - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
     - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

建立 EC2 Fleet

若要建立 EC2 機群，請在 JSON 檔案中定義機群組態，並使用 create-fleet 命令參考檔案。在 JSON 檔案中，您必須指定機群的總目標容量、Spot 執行個體和隨需執行個體的個別目標容量，以及定義機群中執行個體組態的啟動範本，例如 AMI、執行個體類型、子網路或可用區域，以及一個或多個安全群組。您可以選擇指定其他組態，例如覆寫啟動範本組態的參數、從 EC2 容量集區中選取 Spot 執行個體和隨需執行個體的配置策略，以及您針對機群所願意支付的最高金額。如需詳細資訊，請參閱EC2 機群或 Spot 機群的組態選項。

EC2 機群會在容量可用時啟動隨需執行個體，並在您的最高價格超過 Spot 價格且容量可用時，啟動 Spot 執行個體。

如果您的機群包含 Spot 執行個體，且類型為 `maintain`，則 Amazon EC2 會在 Spot 執行個體中斷時，嘗試維護機群目標容量。

EC2 Fleet 限制

下列限制適用於 EC2 Fleet：

- 僅能透過 Amazon EC2 API、AWS CLI、AWS SDK 和 CloudFormation 建立 EC2 機群。
- EC2 機群請求無法跨越 AWS 區域。您需要為每個區域建立個別的 EC2 Fleet。
- EC2 Fleet 請求不能橫跨同一個可用區域的不同子網。

建立 EC2 Fleet

若您要使用 EC2 機群啟動執行個體機群，只需在機群請求中指定下列參數，機群將使用其他參數的預設值：

- `LaunchTemplateId` 或 `LaunchTemplateName` – 指定要使用的啟動範本 (其中包含要啟動的執行個體參數，例如執行個體類型和可用區域)
- `TotalTargetCapacity` – 指定機群的總目標容量
- `DefaultTargetCapacityType` – 指定預設購買選項是隨需還是 Spot

若要覆寫啟動範本中指定的參數，您可以指定一或多個覆寫。每個覆寫可能因執行個體類型、可用區域、子網路和最高價格而異，且可能包含不同的加權容量。做為指定執行個體類型的替代方式，您可以指定執行個體必須具有的屬性，然後 Amazon EC2 會識別具有這些屬性的所有執行個體類型。如需詳細資訊，請參閱 指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型。

對於類型為 的 EC2 機群instant，您也可以覆寫下列啟動範本參數：

- IamInstanceProfile – 要與執行個體建立關聯的 IAM 執行個體描述檔。如需詳細資訊，請參閱Amazon EC2 的 IAM 的角色。
- KeyName – 用於執行個體的金鑰對名稱。如需詳細資訊，請參閱Amazon EC2 金鑰對和 Amazon EC2 執行個體。
- MetadataOptions – 執行個體的執行個體中繼資料服務組態。如需詳細資訊，請參閱使用執行個體中繼資料服務存取執行個體中繼資料。

對於類型為 instant 的 EC2 機群，您可以指定 Systems Manager 參數，而不是 AMI ID。您可以在覆寫或啟動範本中，指定 Systems Manager 參數。如需詳細資訊，請參閱使用 Systems Manager 參數而非 AMI ID。

對於類型為 的 EC2 機群instant，您也可以使用 LaunchTemplateSpecificationUserData 參數在啟動範本規格層級指定使用者資料。使用此參數，您可以提供 base64 編碼的使用者資料，而無需將其包含在啟動範本本身中。在 base64 編碼之前，使用者資料不得超過 16 KB。

您可以指定 JSON 檔案中的機群參數。如需您可以指定的所有可能參數相關資訊，請參閱 檢視所有 EC2 機群組態選項。

如需機群組態的範例，請參閱 EC2 機群的範例 CLI 組態。

目前沒有主控台支援建立 EC2 Fleet。

建立 EC2 機群

使用 create-fleet 命令來建立機群，並指定包含機群組態參數的 JSON 檔案。

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

以下是類型 request 或 maintain 的叢集輸出範例。

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

以下是啟動目標容量之類型 instant 的叢集輸出範例。

回應包含每組已啟動執行個體的可用區域 ID、可用區域名稱和子網路 ID。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ],
      "InstanceType": "c5.large",
      "Platform": null,
      "AvailabilityZoneId": "use1-az1",
      "AvailabilityZone": "us-east-1a",
      "SubnetId": "subnet-0123456789abcdefEXAMPLE"
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-5678901234abcdef0",
        "i-5432109876abcdef9"
      ],
      "AvailabilityZoneId": "use1-az1",
```

```
      "AvailabilityZone": "us-east-1a",
      "SubnetId": "subnet-0123456789abcdefEXAMPLE"
    }
  ]
}
```

以下是啟動部分目標容量並附有未啟動之執行個體錯誤之類型 instant 的叢集輸出範例。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientInstanceCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
```

```
      ]
  ]
}
```

以下是未啟動任何執行個體之類型 instant 的叢集輸出範例。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": []
}
```

建立取代 Spot 執行個體運作狀態不佳的 EC2 機群

EC2 Fleet 會每兩分鐘定期檢查機群中執行個體的運作狀態。執行個體的運作狀態會是 `healthy` 或 `unhealthy`。

EC2 Fleet 會根據 Amazon EC2 提供的狀態檢查資訊，來判定執行個體的運作狀態。如果執行個體狀態檢查或系統狀態檢查的運作狀態檢查，出現連續三次為 `impaired`，則該執行個體即為 `unhealthy`。如需詳細資訊，請參閱Amazon EC2 執行個體的狀態檢查。

您可以設定機群以取代運作狀態不佳的 競價型執行個體。`ReplaceUnhealthyInstances` 設為 `true` 以後，在報告為 `unhealthy` 時，則 Spot 執行個體會被取代。當取代一個狀態不佳的 Spot 執行個體時，機群可能會低於其目標容量長達幾分鐘。

要求

- 運作狀態檢查取代僅支援維持目標容量的 EC2 Fleet (類型 `maintain` 的機群)，不支援類型 `request` 或 `instant` 的機群。
- 運作狀態檢查取代僅支援 競價型執行個體。此功能不支援 隨需執行個體。
- 您可以設定 EC2 Fleet 只在建立執行個體時取代狀況不良的執行個體。
- 使用者只有在擁有呼叫 `ec2:DescribeInstanceStatus` 動作的許可時，才能使用運作狀態檢查取代功能。

設定 EC2 Fleet 以取代運作狀態不佳的 競價型執行個體

1. 使用資訊在 建立 EC2 Fleet 中建立 EC2 機群。
2. 若要設定機群以取代運作狀態不佳的 Spot 執行個體，在 JSON 檔案中，針對 `ReplaceUnhealthyInstances` 請指定 `true`。

檢視所有 EC2 機群組態選項

若要檢視 EC2 Fleet 組態參數的完整清單，您可以產生 JSON 檔案。如需每個參數的描述，請參閱 create-fleet。

產生所有可能 EC2 機群參數的 JSON 檔案

使用 create-fleet (AWS CLI) 命令和 `--generate-cli-skeleton` 參數來產生 EC2 Fleet JSON 檔案，並將輸出導向至檔案以儲存它：

```
aws ec2 create-fleet \
    --generate-cli-skeleton input > ec2createfleet.json
```

以下為範例輸出。

```
{
    "DryRun": true,
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch"
            }
        },
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        },
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": "",
                "Version": ""
            },
            "Overrides": [
                {
                    "InstanceType": "r5.metal",
```

```
                    "MaxPrice": "",
                    "SubnetId": "",
                    "AvailabilityZone": "",
                    "WeightedCapacity": 0.0,
                    "Priority": 0.0,
                    "Placement": {
                        "AvailabilityZone": "",
                        "Affinity": "",
                        "GroupName": "",
                        "PartitionNumber": 0,
                        "HostId": "",
                        "Tenancy": "dedicated",
                        "SpreadDomain": "",
                        "HostResourceGroupArn": ""
                    },
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 0
                        },
                        "MemoryMiB": {
                            "Min": 0,
                            "Max": 0
                        },
                        "CpuManufacturers": [
                            "amd"
                        ],
                        "MemoryGiBPerVCpu": {
                            "Min": 0.0,
                            "Max": 0.0
                        },
                        "ExcludedInstanceTypes": [
                            ""
                        ],
                        "InstanceGenerations": [
                            "previous"
                        ],
                        "SpotMaxPricePercentageOverLowestPrice": 0,
                        "OnDemandMaxPricePercentageOverLowestPrice": 0,
                        "BareMetal": "included",
                        "BurstablePerformance": "required",
                        "RequireHibernateSupport": true,
                        "NetworkInterfaceCount": {
                            "Min": 0,
```

```
                        "Max": 0
                    },
                    "LocalStorage": "excluded",
                    "LocalStorageTypes": [
                        "ssd"
                    ],
                    "TotalLocalStorageGB": {
                        "Min": 0.0,
                        "Max": 0.0
                    },
                    "BaselineEbsBandwidthMbps": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorTypes": [
                        "inference"
                    ],
                    "AcceleratorCount": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorManufacturers": [
                        "amd"
                    ],
                    "AcceleratorNames": [
                        "a100"
                    ],
                    "AcceleratorTotalMemoryMiB": {
                        "Min": 0,
                        "Max": 0
                    }
                }
            }
        ]
    }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 0,
    "OnDemandTargetCapacity": 0,
    "SpotTargetCapacity": 0,
    "DefaultTargetCapacityType": "on-demand",
    "TargetCapacityUnitType": "memory-mib"
},
"TerminateInstancesWithExpiration": true,
```

```
    "Type": "instant",
    "ValidFrom": "1970-01-01T00:00:00",
    "ValidUntil": "1970-01-01T00:00:00",
    "ReplaceUnhealthyInstances": true,
    "TagSpecifications": [
        {
            "ResourceType": "fleet",
            "Tags": [
                {
                    "Key": "",
                    "Value": ""
                }
            ]
        }
    ],
    "Context": ""
}
```

標記新的或現有的 EC2 機群請求，及其啟動的執行個體和磁碟區

為便於將 EC2 機群請求及其啟動的執行個體和磁碟區分類並進行管理，您可以利用自訂中繼資料來標記這些請求。您可以在建立 EC2 Fleet 請求時，或建立之後，將標籤指派給該請求。同樣地，您可以在機群啟動執行個體和磁碟區時或之後，將標籤指派給執行個體和磁碟區。

當您標記機群請求時，由機群啟動的執行個體和磁碟區不會自動加上標記。您需要明確標記由機群啟動的執行個體和磁碟區。您可以選擇僅將標籤指派給機群請求，或僅指派給機群啟動的執行個體，或僅指派給連接至機群啟動之執行個體的磁碟區，或僅指派給機群建立的網路介面，或指派給所有執行個體。

> **Note**
>
> 針對 `instant` 機群類型，您可以為連接至 隨需執行個體 和 競價型執行個體 的磁碟區加標籤。您也可以標記`instant`機群類型的網路介面。針對 `request` 或 `maintain` 機群類型，您只可以為連接至 隨需執行個體 的磁碟區加標籤。

如需有關標籤運作方式的詳細資訊，請參閱標記您的 Amazon EC2 資源。

必要條件

授予使用者標記資源的許可。如需詳細資訊，請參閱範例：標籤資源。

授予使用者標記資源的許可

建立包含下列項目的 IAM 政策：

- `ec2:CreateTags` 動作。這會授予使用者建立標籤的許可。
- `ec2:CreateFleet` 動作。這會授予使用者建立 EC2 Fleet 請求的許可。
- 針對 `Resource`，建議您指定 `"*"`。這可讓使用者標記所有資源類型。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

標記新的 EC2 Fleet 請求

若要在建立 EC2 Fleet 請求時加以標記，請在建立機群時所用的 JSON 檔案中指定金鑰值對。`ResourceType` 的值必須為 `fleet`。若您指定其他值，機群請求將會失敗。

標記由 EC2 Fleet 啟動的執行個體和磁碟區

若要在機群啟動執行個體時標記這些執行個體和磁碟區，請在 EC2 Fleet 請求中參考的啟動範本中指定標籤。

> **Note**
>
> 您無法標記連接至 競價型執行個體、由 `request` 或 `maintain` 機群類型啟動的磁碟區。

若要標記現有的 EC2 機群請求、執行個體和磁碟區

使用建立標籤命令來標記現有資源。

```
aws ec2 create-tags \
```

```
    --resources fleet-12a34b55-67cd-8ef9-
ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \
    --tags Key=purpose,Value=test
```

描述 EC2 機群、其執行個體，及其事件

您可說明您的 EC2 機群組態、您 EC2 機群中的執行個體，及您 EC2 機群的事件歷史記錄。

任務

- 說明您的 EC2 機群
- 說明 EC2 機群中的所有執行個體
- 說明 EC2 機群的事件歷史記錄

說明您的 EC2 機群

AWS CLI

如要說明您的 EC2 機群

使用 describe-fleets 命令。

```
aws ec2 describe-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

以下為範例輸出。

```
{
    "Fleets": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2022-02-09T03:35:52+00:00",
            "FleetId": "fleet-364457cd-3a7a-4ed9-83d0-7b63e51bb1b7",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 2.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template",
```

```
                        "Version": "$Latest"
                    }
                }
            ],
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "TerminateInstancesWithExpiration": false,
            "Type": "maintain",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "AllocationStrategy": "capacity-optimized",
                "InstanceInterruptionBehavior": "terminate"
            },
            "OnDemandOptions": {
                "AllocationStrategy": "lowestPrice"
            }
        }
    ]
}
```

PowerShell

如要說明您的 EC2 機群

使用 Get-EC2FleetList cmdlet。

```
Get-EC2FleetList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

說明 EC2 機群中的所有執行個體

執行中執行個體的傳回清單會定期重新整理且可能過期。

AWS CLI

說明適用於指定 EC2 機群的執行個體。

使用 describe-fleet-instances 命令。

```
aws ec2 describe-fleet-instances \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

以下為範例輸出。

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-09cd595998cb3765e",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-86k84j6p"
        },
        {
            "InstanceId": "i-09cf95167ca219f17",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-dvxi7fsm"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

PowerShell

說明適用於指定 EC2 機群的執行個體。

使用 Get-EC2FleetInstanceList cmdlet。

```
Get-EC2FleetInstanceList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

說明 EC2 機群的事件歷史記錄

如需有關事件歷史記錄中事件的詳細資訊，請參閱 EC2 機群 事件類型。

AWS CLI

描述指定 EC2 機群的事件

使用 describe-fleet-history 命令。

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2020-06-01T00:00:00Z
```

以下為範例輸出。

```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
```

```
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2020-06-01T00:00:00.000Z"
}
```

PowerShell

描述指定 EC2 機群的事件

使用 Get-EC2FleetHistory cmdlet。

```
Get-EC2FleetHistory `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE `
    -UtcStartTime 2020-06-01T00:00:00Z
```

修改 EC2 Fleet

您可以修改 EC2 機群的總目標容量、Spot 容量和隨需容量。如果機群新的總目標容量減少到低於機群的目前大小，您也可以修改是否應終止執行中的執行個體。

考量事項

修改 EC2 機群時，請考慮下列事項：

- 機群類型 – 您僅可以修改 `maintain` 類型的 EC2 機群。您無法修改 `request` 或 `instant` 類型的 EC2 機群。
- 機群參數 – 您可以修改 EC2 機群的下列參數：
  - `target-capacity-specification` – 增加或減少以下的目標容量：
    - `TotalTargetCapacity`
    - `OnDemandTargetCapacity`
    - `SpotTargetCapacity`
  - `excess-capacity-termination-policy` – 如果機群的總目標容量減少到低於 EC2 Fleet 的目前大小，是否應終止執行中的執行個體。有效的值如下：

  - `no-termination`
  - `termination`
- 增加總目標容量時的機群行為 – 當您增加總目標容量時，EC2 機群會根據為 `DefaultTargetCapacityType` 指定的執行個體購買選項 (即隨需執行個體或 Spot 執行個體) 並根據指定的配置策略啟動其他執行個體。
- 減少 Spot 目標容量時的機群行為 – 當您降低 Spot 目標容量時，EC2 機群會刪除超過新目標容量的任何開放請求。您可以請求機群終止 Spot 執行個體，直到機群大小達到新的目標容量。當 EC2 Fleet 因為目標容量減少而終止 Spot 執行個體時，執行個體會收到 Spot 執行個體中斷通知。

  會根據配置策略選取要終止的執行個體：

  - `capacity-optimized` – 從可用容量最少的集區中終止執行個體。
  - `price-capacity-optimized` – 使用價格與可用容量的組合：從可用容量最少，且在這些集區中價格最高的集區終止執行個體。
  - `diversified` – 在所有集區中終止執行個體。
  - `lowest-price` – 從價格最高的集區終止執行個體。

  或者，您可以請求該 EC2 機群將機群保持為目前的大小，但不能取代任何中斷的或您手動終止的 Spot 執行個體。
- 機群狀態 – 您可以修改處於 `submitted` 或 `active` 狀態中的 EC2 機群。當您修改機群時，機群會進入 `modifying` 狀態。

修改 EC2 機群的命令

AWS CLI

修改 EC2 機群的總目標容量

使用 modify-fleet 命令。

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=20
```

如果您要降低目標容量，但希望保持目前的大小，則可以按照下列方式修改前一個範例。

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
```

```
    --target-capacity-specification TotalTargetCapacity=10 \
    --excess-capacity-termination-policy no-termination
```

PowerShell

修改 EC2 機群的總目標容量

使用 Edit-EC2Fleet cmdlet。

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20
```

如果您要降低目標容量，但希望保持目前的大小，則可以按照下列方式修改前一個範例。

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

刪除 EC2 機群請求和機群中的執行個體

如果您不再需要某個 EC2 機群請求，可以將其刪除。刪除機群請求後，也會取消與機群相關的所有 Spot 請求，因此不會啟動新的 Spot 執行個體。

您在刪除 EC2 機群時，必須指定是否也要終止其所有執行個體。這包含隨需執行個體和 Spot 執行個體。對於 `instant` 機群，EC2 機群必須在機群刪除時終止執行個體。不支援具有執行中執行個體的已刪除 `instant` 機群。

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

如果指定在刪除機群請求時必須終止執行個體，則機群請求會進入 `deleted_terminating` 狀態。否則會進入 `deleted_running` 狀態，且執行個體會一直執行直到中斷或您手動終止。

限制

- 您可以在單一操作中刪除多達 25 個 instant 類型的機群。
- 您可以在單一操作中刪除多達 100 個 maintain 或 request 類型的機群。
- 您可以在單一操作中刪除最多 125 個機群，前提是您不超過以上所指定每個機群類型的配額。
- 如果您超過要刪除的指定機群數量，則不會刪除任何機群。
- 不支援具有執行中執行個體的已刪除 instant 機群。刪除 instant 機群時，Amazon EC2 會自動終止其所有執行個體。對於包含超過 1000 個執行個體的 instant 機群，刪除請求可能會失敗。若您的機群有超過 1000 個執行個體，請先手動終止大部分執行個體，留下 1000 個或以下的執行個體。接著刪除機群，剩餘的執行個體將會自動終止。

AWS CLI

若要刪除 EC2 機群請求並終止其執行個體

使用 delete-fleets 命令並搭配 --terminate-instances 選項。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

以下為範例輸出。

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
        }
    ]
}
```

若要刪除 EC2 機群請求但不終止其執行個體

請改用 --no-terminate-instances 選項來修改之前的範例。請注意，--no-terminate-instances 不支援 instant 機群。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

以下為範例輸出。

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
        }
    ]
}
```

PowerShell

若要刪除 EC2 機群請求並終止其執行個體

使用 Remove-EC2Fleet cmdlet 搭配 `-TerminateInstance` 參數。

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

若要刪除 EC2 機群請求但不終止其執行個體

透過變更 `-TerminateInstance` 參數的值來修改之前的範例。

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

當機群無法刪除時進行故障診斷

如果刪除 EC2 機群請求刪除失敗，則輸出中的 `UnsuccessfulFleetDeletions` 會傳回的 EC2 機群 ID、錯誤碼和錯誤訊息。

錯誤代碼為：

- ExceededInstantFleetNumForDeletion
- fleetIdDoesNotExist
- fleetIdMalformed
- fleetNotInDeletableState
- NoTerminateInstancesNotSupported
- UnauthorizedOperation
- unexpectedError

ExceededInstantFleetNumForDeletion 疑難排解

如果您在單一請求中嘗試刪除超過 25 個 instant 機群，則會傳回 ExceededInstantFleetNumForDeletion 錯誤。下列為此錯誤的範例輸出。

```
{
    "UnsuccessfulFleetDeletions": [
     {
          "FleetId": " fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
          "Error": {
                 "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                 "Code": "ExceededInstantFleetNumForDeletion"
           }
     },
     {
           "FleetId": "fleet-9a941b23-0286-5bf4-2430-03a029a07e31",
           "Error": {
                 "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                 "Code": "ExceededInstantFleetNumForDeletion"
            }
     }
     .
     .
     .
     ],
     "SuccessfulFleetDeletions": []
}
```

NoTerminateInstancesNotSupported 疑難排解

如果您指定在刪除 instant 機群時不得終止該機群中的執行個體，則會傳回 NoTerminateInstancesNotSupported 錯誤。--no-terminate-instances 不支援 instant 機群。下列為此錯誤的範例輸出。

```
{
      "UnsuccessfulFleetDeletions": [
            {
                  "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
                  "Error": {
                          "Message": "NoTerminateInstances option is not supported for
 instant fleet",
                          "Code": "NoTerminateInstancesNotSupported"
                   }
            }
       ],
       "SuccessfulFleetDeletions": []
}
```

UnauthorizedOperation 疑難排解

如果沒有終止執行個體的許可，您在刪除必須終止其執行個體的機群時，會收到 UnauthorizedOperation 錯誤。以下是錯誤回應。

```
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not
 authorized to perform this
operation. Encoded authorization failure message: VvuncIxj7Z_CPGNYXWqnuFV-
YjByeAU66Q9752NtQ-I3-qnDLWs6JLFd
KnSMMiq5s6cGqjjPtEDpsnGHzzyHasFHOaRYJpaDVravoW25azn6KNkUQQlFwhJyujt2dtNCdduJfrqcFYAjlEiRMkfDHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNtqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdhlONg6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmrm2m01-
EMhekLFZeJLr
DtYOpYcEl4_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1PxODZuqWHs23YXWVywzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheurOGpkprPIC
</Message></Error></Errors><RequestID>89b1215c-7814-40ae-a8db-41761f43f2b0</
RequestID></Response>
```

若要解決錯誤，您必須將 ec2:TerminateInstances 動作新增至 IAM 政策，如下列範例所示。

使用 Spot 機群

> **Important**
>
> 我們強烈建議您使用 Spot Fleet，因為它使用沒有計劃投資的舊版 API。如果您想要管理執行個體生命週期，請改用 EC2 機群。如果您不想管理執行個體生命週期，請改用 Auto Scaling 群組。只有在您需要主控台支援使用 EC2 機群的使用案例時，才使用 Spot 機群。如需詳細資訊，請參閱最好的機群方法是什麼？。

若要開始使用 Spot 機群，請建立包含 Spot 執行個體總目標容量的請求、選用隨需部分，並手動指定 AMI 和金鑰對，或是指定包含機群中執行個體組態的啟動範本。您可以選擇性地指定其他參數，或讓機群使用預設值。您在建立機群時，也可以標記機群請求及其執行個體和磁碟區。

機群會在有可用容量時啟動隨需執行個體，並在最高價格超過 Spot 價格且容量可用時，啟動 Spot 執行個體。

機群啟動後，您可以描述機群請求、機群中的執行個體，以及任何機群事件。您也可以視需要指派其他標籤。

如果您需要變更任何機群參數 (例如總目標容量)，您可以修改機群，前提是其已設定為維持容量。提交之後，您就無法修改一次性請求的容量。

機群請求會一直保持作用中狀態，直到過期或您將其取消 (刪除) 為止。當您取消機群請求時，可以終止執行個體，或讓執行個體保持執行中狀態。如果您選擇讓執行個體保持執行中狀態，隨需執行個體會執行直到您將其終止為止，而 Spot 執行個體則會一直執行直到中斷或您將其終止為止。

主題

- Spot Fleet 請求狀態
- Spot Fleet 許可
- 建立 Spot 機群
- 標記新的或現有的 Spot 機群請求及其啟動的執行個體和磁碟區
- 說明 Spot 機群請求、其執行個體和事件歷史記錄
- 修改 Spot Fleet 請求
- 取消 (刪除) Spot 機群請求
- 了解 Spot 機群的自動擴展

Spot Fleet 請求狀態

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

Spot 機群請求可以是各種狀態之一，每個狀態都會指出請求生命週期的不同階段，以及其執行個體管理。

Spot Fleet 請求可以是下列任一狀態：

`submitted`

正在評估 Spot 機群請求且 Amazon EC2 正準備啟動執行個體的目標數量。超出 Spot 機群配額的請求會立即取消。

`active`

已驗證 Spot 機群且 Amazon EC2 正在嘗試維持執行中的 Spot 執行個體目標數量。此請求會一直維持在該狀態，直到修改或取消。

`modifying`

正在修改 Spot 機群請求。請求會一直維持在此狀態，直到已完成修改或是已取消請求。只有 `maintain` 機群類型可以修改。此狀態不適用於一次性 `request` 機群類型。

`cancelled_running`

Spot 機群已取消 (已刪除)，且不會啟動額外的 Spot 執行個體。其現有執行個體將繼續執行直到中斷或手動終止。此請求會一直維持在該狀態，直到所有執行個體中斷或終止。

`cancelled_terminating`

Spot 機群已取消 (已刪除)，且其執行個體已終止。此請求會一直維持在該狀態，直到所有執行個體終止。

`cancelled`

Spot 機群已取消 (已刪除)，且無執行中的執行個體。請求會在其執行個體終止之後兩天刪除。

Spot Fleet 許可

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

如果使用者將會建立或管理 Spot 機群，您必須授予這些使用者所需的許可。

如果使用 Amazon EC2 主控台來建立 Spot Fleet，它會建立名為 `AWSServiceRoleForEC2SpotFleet` 和 `AWSServiceRoleForEC2Spot` 的兩個服務連結角色，以及名為 `aws-ec2-spot-fleet-tagging-role` 的角色來授與 Spot Fleet 許可，以代表您請求、啟動、終止和標記資源。如果您使用 AWS CLI 或 API，您必須確保這些角色存在。

請使用下列指示，授與必要的許可並建立角色。

許可和角色

- 將 Spot 機群的許可授予使用者
- Spot Fleet 的服務連結角色
- Spot 執行個體的服務連結角色
- 標記 Spot Fleet 的 IAM 角色

將 Spot 機群的許可授予使用者

如果使用者將會建立或管理 Spot 機群，請務必授予這些使用者所需的許可。

為 Spot 機群建立政策

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇政策、建立政策。
3. 在建立政策頁面上，選擇 JSON，使用下列內容來替代其文字。

   JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:CreateTags",
                "ec2:RequestSpotFleet",
                "ec2:ModifySpotFleetRequest",
                "ec2:CancelSpotFleetRequests",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequestHistory"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/aws-ec2-spot-fleet-tagging-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

上述範例政策會向使用者授予大多數 Spot Fleet 使用案例所需的許可。若要將使用者限制於特定的 API 動作，請改為僅指定這些 API 動作。

需要的 EC2 和 IAM API

必須在原則中包含下列 API：

- `ec2:RunInstances` – 在 Spot Fleet 中啟動執行個體所需
- `ec2:CreateTags` – 標記 Spot Fleet 請求、執行個體或磁碟區所需
- `iam:PassRole` – 指定 Spot Fleet 角色所需

- `iam:CreateServiceLinkedRole` – 建立服務連結角色所需
- `iam:ListRoles` – 列舉現有 IAM 角色所需
- `iam:ListInstanceProfiles` – 列舉現有執行個體描述檔所需

> **Important**
>
> 如果您在啟動規格或啟動範本中為 IAM 執行個體設定檔指定角色，您必須授予使用者許可，以將角色傳遞到服務。若要執行此操作，請在 IAM 政策中包含 `"arn:aws:iam::*:role/IamInstanceProfile-role"`，做為 `iam:PassRole` 動作的資源。如需詳細資訊，請參閱《IAM 使用者指南》中的授予使用者將角色傳遞至 AWS 服務的許可。

Spot Fleet API

視需要將下列 Spot Fleet API 動作新增至您的政策：

- `ec2:RequestSpotFleet`
- `ec2:ModifySpotFleetRequest`
- `ec2:CancelSpotFleetRequests`
- `ec2:DescribeSpotFleetRequests`
- `ec2:DescribeSpotFleetInstances`
- `ec2:DescribeSpotFleetRequestHistory`

選用的 IAM API

(選用) 若要讓使用者能夠使用 IAM 主控台建立角色或執行個體設定檔，您必須將下列動作新增至政策中：

- `iam:AddRoleToInstanceProfile`
- `iam:AttachRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:CreateRole`
- `iam:GetRole`

- `iam:ListPolicies`

4. 選擇檢閱政策。
5. 在 Review policy (檢閱政策) 頁面上，輸入政策名稱和說明，然後選擇 Create policy (建立政策)。
6. 若要提供存取權，請新增權限至您的使用者、群組或角色：

   - 中的使用者和群組 AWS IAM Identity Center：

     建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。

   - 透過身分提供者在 IAM 中管理的使用者：

     建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。

   - IAM 使用者：
     - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
     - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

Spot Fleet 的服務連結角色

Amazon EC2 使用許可的服務連結角色，它需要代表您呼叫其他 AWS 服務。服務連結角色是直接連結至 AWS 服務的唯一 IAM 角色類型。服務連結角色提供將許可委派給 AWS 服務的安全方式，因為只有連結的服務可以擔任服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的服務連結角色。

Amazon EC2 使用名為 AWSServiceRoleForEC2SpotFleet 的服務連結角色來代您管理執行個體。

> **Important**
>
> 如果您在 Spot 機群中指定加密的 AMI 或加密的 Amazon EBS 快照，則必須授予 AWSServiceRoleForEC2SpotFleet 角色許可才能使用 CMK，使得 Amazon EC2 可以代表您啟動執行個體。如需詳細資訊，請參閱授與 CMK 的存取權，以便與加密的 AMI 和 EBS 快照搭配使用。

AWSServiceRoleForEC2SpotFleet 授與的許可

AWSServiceRoleForEC2SpotFleet 角色會授予 Spot 機群許可，以代表您請求、啟動、終止及標記執行個體。Amazon EC2 會使用此服務連結角色來完成下列動作：

- `ec2:RequestSpotInstances` - 請求 競價型執行個體
- `ec2:RunInstances` - 啟動執行個體
- `ec2:TerminateInstances` - 終止執行個體
- `ec2:DescribeImages` - 描述 Amazon Machine Images (AMI) 的執行個體
- `ec2:DescribeInstanceStatus` - 監控執行個體的狀態
- `ec2:DescribeSubnets` - 描述執行個體的子網絡
- `ec2:CreateTags` - 將標籤新增至 Spot Fleet 請求、執行個體和磁碟區
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - 將指定執行個體新增到指定的負載平衡器
- `elasticloadbalancing:RegisterTargets` - 向指定的目標群組登錄指定的目標

建立服務連結角色

在大部分情況下，您不需要手動建立服務連結角色。您第一次使用主控台建立 Spot Fleet 時，Amazon EC2 會建立 AWSServiceRoleForEC2SpotFleet 服務連結角色。

如果您在 2017 年 10 月之前擁有使用中的 Spot Fleet 請求，則當 Amazon EC2 開始支援此服務連結角色時，Amazon EC2 會在您的 AWS 帳戶中建立 AWSServiceRoleForEC2SpotFleet 角色。如需詳細資訊，請參閱《IAM 使用者指南》中的我的 AWS 帳戶中出現的新角色。

如果您使用 AWS CLI 或 API 來建立 Spot 機群，您必須先確保此角色存在。

使用主控台為 Spot 機群建立 AWSServiceRoleForEC2SpotFleet 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 選擇 Create Role (建立角色)。
4. 在 Select trusted entity (選取信任的實體) 頁面上，執行以下作業：
   a. 針對信任的實體類型，請選擇 AWS 服務。
   b. 在使用案例下，針對服務或使用案例，選擇 EC2。
   c. 針對使用案例，請選擇 EC2 - Spot 機群。

> **Note**
>
> EC2 - Spot 機群使用案例會自動建立具有必要 IAM 許可的政策，並將 AWSEC2SpotFleetServiceRolePolicy 建議做為角色名稱。

d. 選擇下一步。

5. 在 Add permissions (新增許可) 頁面上，選擇 Next (下一步)。
6. 在命名、檢閱和建立頁面上，選擇建立角色。

使用 建立 Spot Fleet 的 AWSServiceRoleForEC2SpotFleet 角色 AWS CLI

使用如下所示的 create-service-linked-role 命令。

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

如果您不再需要使用 Spot Fleet，建議您刪除 AWSServiceRoleForEC2SpotFleet 角色。從您的帳戶中刪除此角色後，如果您使用主控台請求 Spot Fleet，Amazon EC2 會再次建立此角色。如需詳細資訊，請參閱「IAM 使用者指南」中的刪除服務連結角色。

授與 CMK 的存取權，以便與加密的 AMI 和 EBS 快照搭配使用

如果您在 Spot 機群請求中指定加密的 AMI 或加密的 Amazon EBS 快照，並且使用客戶受管金鑰進行加密，則您必須授予 AWSServiceRoleForEC2SpotFleet 角色許可才能使用 CMK，使得 Amazon EC2 可以代表您啟動執行個體。若要執行此動作，您必須新增授與至 CMK，如以下列程序所示。

提供許可時，授與為金鑰政策的替代方案。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的使用授予和在 AWS KMS中使用金鑰政策。

授與 AWSServiceRoleForEC2SpotFleet 角色許可以使用 CMK

- 使用 create-grant 命令來新增授與至 CMK，並指定獲授與許可能夠執行授與所允許作業的委託人 (AWSServiceRoleForEC2SpotFleet 服務連結角色)。CMK 是透過 `key-id` 參數和 CMK 的 ARN 指定。委託人是透過 `grantee-principal` 參數和 AWSServiceRoleForEC2SpotFleet 服務連結角色的 ARN 指定。

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
spotfleet.amazonaws.com/AWSServiceRoleForEC2SpotFleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

Spot 執行個體的服務連結角色

Amazon EC2 使用名為 AWSServiceRoleForEC2Spot 的服務連結角色來代您管理 競價型執行個體。如需詳細資訊，請參閱Spot 執行個體請求的服務連結角色。

標記 Spot Fleet 的 IAM 角色

`aws-ec2-spot-fleet-tagging-role` IAM 角色授予 Spot Fleet 許可，以標記 Spot Fleet 請求、執行個體和磁碟區。如需詳細資訊，請參閱標記新的或現有的 Spot 機群請求及其啟動的執行個體和磁碟區。

> **Important**
>
> 如果您選擇標記機群中的執行個體，並也選擇維護目標容量 (Spot 機群請求為類型 `maintain`)，為使用者與 `IamFleetRole` 所設定的許可差異可能會導致機群中執行個體標記行為的不一致。如果 `IamFleetRole` 不包含 `CreateTags` 許可，則可能無法標記由機群啟動的某些執行個體。雖然我們正在努力修正此不一致，但為了確保標記由機群啟動的所有執行個體，建議您針對 `IamFleetRole` 使用 `aws-ec2-spot-fleet-tagging-role` 角色。或者，若要使用現有角色，請將 `AmazonEC2SpotFleetTaggingRole` AWS 受管政策連接至現有角色。否則，您需要手動將 `CreateTags` 許可新增至現有政策。

若要建立用於標記 Spot Fleet 的 IAM 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 選擇 Create Role (建立角色)。
4. 在 Select trusted entity (選取受信任實體) 頁面，於 Trusted entity type (受信任實體的類型) 下，選擇 AWS service (服務)。
5. 在 Use case (使用案例) 下，從 Use cases for other AWS services (其他服務的使用案例)，選擇 EC2，然後選擇 EC2 - Spot Fleet Tagging (EC2 - Spot 機群標記)。

6. 選擇下一步。
7. 在 Add permissions (新增許可) 頁面上，選擇 Next (下一步)。
8. 在 Name, review, and create (名稱，檢閱和建立) 頁面上，針對 Role name (角色名稱) 輸入角色的名稱 (例如 **aws-ec2-spot-fleet-tagging-role**)。
9. 檢視頁面上的資訊，然後選擇 Create role (建立角色)。

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多權限的實體執行該動作。若要限制 Spot 機群為資源提供另一項服務的許可，我們建議在 `aws-ec2-spot-fleet-tagging-role` 信任政策中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰。

將 aws:SourceArn 和 aws:SourceAccount 條件金鑰新增至 **aws-ec2-spot-fleet-tagging-role** 信任政策

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 尋找您先前建立的 `aws-ec2-spot-fleet-tagging-role`，並選擇連結 (而非核取方塊)。
4. 在 Summary (摘要) 下，選擇 Trust relationships (信任關係) 索引標籤，然後選擇 Edit trust policy (編輯信任政策)。
5. 在 JSON 陳述式中，加入一個 `Condition` 元素，其中包含 `aws:SourceAccount` 和 `aws:SourceArn` 全域條件內容金鑰來防止混淆代理問題，如下所式：

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/
sfr-*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
```

> **Note**
>
> 如果 aws:SourceArn 值包含帳戶 ID，且同時使用這兩個全域條件內容金鑰，則在相同政策陳述式中使用 aws:SourceAccount 值和 aws:SourceArn 值中的帳戶時，必須使用相同的帳戶 ID。

最終信任政策如下所示：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "spotfleet.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/sfr-*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
}
```

6. 選擇更新政策。

下表提供了 aws:SourceArn 的潛在值，以不同程度地限制 aws-ec2-spot-fleet-tagging-role 的範圍。

API 操作	呼叫的服務	Scope (範圍)	**aws:SourceArn**
RequestSpotFleet	AWS STS (AssumeRole)	將 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限制為指定帳戶中的 spot-fleet-requests。	arn:aws:ec2:*: *123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	AWS STS (AssumeRole)	將 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限制為指定帳戶和指定區域中的 spot-fleet-requests。請注意，此角色在其他區域中無法使用。	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	AWS STS (AssumeRole)	將 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限制為僅限影響機群 sfr-11111111-1111-1111-1111-111111111111 的動作。請注意，此角色可能無法用於其他 Spot 機群。此外，此角色無法用於透過 request-spot-fleet 啟動任何新的 Spot 機群。	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*11111111-1111-1111-1111-111111111111*

建立 Spot 機群

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

您可以使用 AWS 管理主控台，透過僅選擇 AMI 和所需的總目標容量，快速建立 Spot 機群請求。Amazon EC2 會設定最符合您的需求且遵循 Spot 最佳實務的機群。或者，您可以修改任何預設設定。

如果您想要在機群中包含隨需執行個體，則必須在請求中指定啟動範本，並指定所需的隨需容量。

機群會在有可用容量時啟動隨需執行個體，並在您的最高價格超過 Spot 價格且容量可用時，啟動 Spot 執行個體。

如果您的機群包含 Spot 執行個體，且類型為 `maintain`，則 Amazon EC2 會在 Spot 執行個體中斷時，嘗試維護機群目標容量。

所需的許可

如需詳細資訊，請參閱the section called “Spot Fleet 許可”。

任務

- 快速建立 Spot 機群請求
- 使用已定義的參數建立 Spot 機群請求
- 建立 Spot 機群，可取代運作狀態不佳的 Spot 執行個體

快速建立 Spot 機群請求

依照這些步驟使用 Amazon EC2 主控台，快速建立 Spot 機群請求。

若要使用建議的設定建立 Spot 機群請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 如果您是第一次使用 Spot，將會顯示歡迎頁面；請選擇開始使用。否則，選擇建立 Spot 機群請求。

4. 在 Launch parameters (啟動參數) 下，選擇 Manually configure launch parameters (手動設定啟動參數)。
5. 對於 AMI，選擇 AMI。
6. 在 Target capacity (目標容量) 下， 針對 Total target capacity (總目標容量)，指定要請求的單位數目。對於單位類型，您可以選擇執行個體、vCPU 或記憶體 (GiB)。
7. 在您的機群請求一覽，檢閱您的機群組態，然後選擇啟動。

使用已定義的參數建立 Spot 機群請求

您可以使用您定義的參數建立 Spot Fleet。

Console

使用已定義的參數建立 Spot 機群請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 如果您是第一次使用 Spot，將會顯示歡迎頁面；請選擇開始使用。否則，選擇建立 Spot 機群請求。
4. 對於啟動參數，您可以手動設定啟動參數，也可以使用啟動範本，如下所示：

   a. [手動設定] 若要在 Amazon EC2 主控台中定義啟動參數，請選擇手動設定啟動參數，然後執行下列動作：

      i. 針對 AMI，請選擇 提供的基本 AMIs AWS，或選擇搜尋 AMI 以使用來自我們使用者社群 AWS Marketplace、 或您自己的 AMI。

      > **Note**
      >
      > 如果取消註冊或停用啟動參數中指定的 AMI，則無法從 AMI 啟動新的執行個體。對於設定為維持目標容量的機群，將不會維持目標容量。

      ii. (選用) 針對金鑰對名稱，選擇現有的金鑰對或建立新的金鑰對。

      [現有的金鑰對] 選擇金鑰對。

      [新的金鑰對] 選擇建立新的金鑰對，以前往金鑰對頁面。當您完成時，請返回 Spot Requests (Spot 請求) 頁面並重新整理清單。

iii. (選用) 展開 Additional launch parameters (其他啟動參數)，並執行下列動作：

A. (選用) 若要啟用 Amazon EBS 最佳化，請針對 EBS-optimized (EBS 最佳化)，選取 Launch EBS-optimized instances (啟動 EBS 最佳化的執行個體)。

B. (選用) 若要新增執行個體的暫時區塊層級儲存，請針對執行個體存放區，選擇連接啟動。

C. (選用) 若要新增儲存空間，請選擇 Add new volume (新增磁碟區)，並指定其他的執行個體存放區磁碟區或 Amazon EBS 磁碟區，取決於執行個體類型。

D. (選用) 執行個體預設為啟用基本監控。若要啟用詳細的監控，請針對 Monitoring (監控)，選取 Enable CloudWatch detailed monitoring (啟用 CloudWatch 詳細監控)。

E. (選用) 若要執行專用 Spot 執行個體，對於租用，請選擇 Dedicated - run a dedicated instance (專用 - 執行專用執行個體)。

F. (選用) 針對安全群組，選擇一或多個安全群組或建立新的安全群組。

[現有的安全群組] 選擇一或多個安全群組。

[新的安全群組] 選擇 Create new security group (建立新安全群組) 以前往 Security Groups (安全群組) 頁面。當您完成時，請返回 Spot Requests (Spot 請求) 並重新整理清單。

G. (選用) 若要從網際網路連線到您的執行個體，請針對自動指派 IPv4 公有 IP，選擇啟用。

H. (選用) 若要啟動包含 IAM 角色的 競價型執行個體，請針對 IAM 執行個體描述檔，選擇該角色。

I. (選用) 若要執行啟動指令碼，請將指令碼複製到使用者資料。

J. (選用) 若要新增標籤，請選擇 Create tag (建立標籤) 並輸入標籤的鍵和值，然後選擇 Create (建立)。針對每個標籤，重複執行上述動作。

對於每個標籤，若要使用相同標籤標記執行個體和 Spot Fleet 請求，請確定同時選取 Instances (執行個體) 和 Fleet (機群)。若只要標記由機群啟動的執行個體，請清除 Fleet (機群)。若只要標記 Spot Fleet 請求，請清除 Instances (執行個體)。

b. [啟動範本] 若要使用您在啟動範本中建立的組態，請選擇使用啟動範本，對於啟動範本，請選擇一個啟動範本。

> **Note**
>
> 如果您想要 Spot 機群中的隨需容量，則必須指定啟動範本。

5. 針對其他請求詳細資訊，執行下列操作：

   a. 檢閱其他的請求詳細資訊。若要進行變更，請清除套用預設值。

   b. (選用) 針對 IAM 叢集角色，您可使用預設角色或選擇不同的角色。若要在變更角色後使用預設角色，請選擇使用預設角色。

   c. (選用) 若要建立僅在特定期間內有效的請求，請編輯請求有效性起始日期和請求有效性結束日期。

   d. (選用) 根據預設，當 Spot 機群請求過期時，Amazon EC2 會終止您的 Spot 執行個體。若要在請求過期後繼續執行這些執行個體，請清除在請求過期時終止執行個體。

   e. (選用) 若要登錄 競價型執行個體與負載平衡器，請選擇從一個或多個負載平衡器接收流量，然後選擇一個或多個 Classic Load Balancer 或目標群組。

6. 針對 Target capacity (目標容量)，執行下列動作：

   a. 針對 Total target capacity (總目標容量)，指定目標容量要請求的單位數目。對於單位類型，您可以選擇 Instances (執行個體)、vCPUs, 或 Memory (MiB) (記憶體 (MiB))。若要指定目標容量為 0，以便稍後可新增容量，您必須先選取維持目標容量。

   b. (選用) 針對 Include On-Demand base capacity (包含隨需基礎容量)，指定要請求的隨需單位數量。數量必須小於總目標容量。Amazon EC2 會計算差額，並將差額分配給要請求的 Spot 單位。

   > **Important**
   >
   > 若要指定選用的隨需容量，您必須先選擇啟動範本。

   c. (選用) Amazon EC2 預設會在 Spot 執行個體中斷時終止。若要維持目標容量，請選取維持目標容量。然後您就可以在 Spot 執行個體中斷時，指定 Amazon EC2 終止、停止或休眠。若要這麼做，請在中斷行為中選擇對應的選項。

   > **Note**
   >
   > 如果取消註冊或停用啟動參數中指定的 AMI，則無法從 AMI 啟動新的執行個體。在此情況下，對於設定為維持目標容量的機群，將不會維持目標容量。

d. (選用) 若要允許 Spot Fleet 在機群中為現有的 Spot 執行個體發出執行個體重新平衡通知時啟動替代 Spot 執行個體，請選取 Capacity rebalance (容量重新平衡)，然後選擇執行個體取代策略。如果選擇終止前啟動，請指定 Amazon EC2 終止舊執行個體之前的延遲 (以秒為單位)。如需詳細資訊，請參閱在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體。

e. (選用) 若要控制您為了在機群中所有的 Spot 執行個體所支付的每小時金額，請選取 Set maximum cost for Spot Instances (設定 Spot 執行個體的最大成本)，然後輸入您願意支付的每小時金額上限。達到總額上限時，Spot Fleet 會停止啟動 Spot 執行個體，即使它未符合目標容量亦然。如需詳細資訊，請參閱設定 EC2 機群或 Spot 機群的花費限制。

7. 針對 Network (網路)，請執行下列動作：

a. 針對網路，選擇現有的 VPC 或建立新的 VPC。

[現有的 VPC] 選擇 VPC。

[新的 VPC] 選擇建立新的 VPC 以進入 Amazon VPC 主控台。當您完成時，請返回此畫面，並重新整理清單。

b. (選用) 對於可用區域，讓 Amazon EC2 為您的 Spot 執行個體選擇可用區域，或指定一個或多個可用區域。

如果您在可用區域中擁有一或多個子網，請從子網選擇適當的子網。若要新增子網，請選擇建立新的子網，以進入 Amazon VPC 主控台。當您完成時，請返回此畫面，並重新整理清單。

8. 針對 Instance type requirements (執行個體類型需求)，您可以指定執行個體屬性，並讓 Amazon EC2 利用這些屬性識別最的執行個體類型，或者您可以指定執行個體清單。如需詳細資訊，請參閱指定屬性，為 EC2 機群或 Spot 機群選取執行個體類型。

a. 如果選擇 Specify instance attributes that match your compute requirements (指定符合運算需求的執行個體屬性)，請指定您的執行個體屬性，如下所示：

i. 對於 vCPUs，輸入所需的 vCPU 數量下限和上限。若要指定無限制，請選取無下限和/或無上限。

ii. 對於 Memory (GiB) (記憶體 (GiB))，輸入所需記憶體數量的下限和上限。若要指定無限制，請選取無下限和/或無上限。

iii. (選用) 對於其他執行個體屬性，您可以選擇性地指定一或多個屬性，以更詳細地表達您的運算需求。每個額外屬性都會將進一步的限制新增至您的請求。您可以省略額外

屬性；省略時，就會使用預設值。如需每個屬性及其預設值的描述，請參閱 get-spot-placement-scores。

iv. (選用) 若要檢視具有所指定屬性的執行個體類型，請展開 Preview matching instance types (預覽相符的執行個體類型)。若要排除執行個體類型，使其不在請求中使用，請選取執行個體，然後選擇 Exclude selected instance types (排除選取的執行個體類型)。

b. 如果選擇 Manually select instance types (手動選取執行個體類型)，Spot Fleet 會提供預設的執行個體類型清單。若要選取更多執行個體類型，請選擇 Add instance types (新增執行個體類型)、選取要在請求中使用的執行個體類型，然後選擇 Select (選取)。若要刪除執行個體類型，請選取執行個體類型並選擇 Delete (刪除)。

9. 對於配置策略，選擇符合您需求的 Spot 配置策略和隨需配置策略。如需詳細資訊，請參閱使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量。
10. 針對 Your fleet request at a glance (您的機群請求一覽)，檢閱您的機群配置，並視需要進行任何調整。
11. （選用）若要下載啟動組態的副本以搭配 使用 AWS CLI，請選擇 JSON 組態。
12. 當您準備好啟動 Spot 機群時，選擇啟動。

    Spot 機群請求類型為 `fleet`。請求處理完成時，會新增類型為 `instance` 的請求，其狀態為 `active` 和 `fulfilled`。

AWS CLI

建立 Spot 機群請求

使用 request-spot-fleet 命令。

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

如需組態檔案範例，請參閱 範例 CLI 組態 Spot 機群。

PowerShell

建立 Spot 機群請求

使用 Request-EC2SpotFleet cmdlet。下列範例在容量最佳化機群中啟動 Spot 執行個體。

```
Request-EC2SpotFleet `
```

```
    -SpotFleetRequestConfig_TargetCapacity 50 `
    -SpotFleetRequestConfig_AllocationStrategy "CapacityOptimized" `
    -SpotFleetRequestConfig_IamFleetRole "arn:aws:iam::123456789012:role/my-spot-
fleet-role" `
    -SpotFleetRequestConfig_LaunchTemplateConfig @($launchConfig)
```

如下所示，定義啟動組態，設定所需啟動範本與覆寫屬性。如需組態範例，請參閱 範例 CLI 組態 Spot 機群。

```
$lcSpec = Amazon.EC2.Model.FleetLaunchTemplateSpecification
# To do - Set FleetLaunchTemplateSpecification properties
$lcOverrides = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
# To do - Set LaunchTemplateOverrides properties
$launchConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchConfig.LaunchTemplateSpecification $lcSpec
$launchConfig.Overrides @($lcOverrides)
```

建立 Spot 機群，可取代運作狀態不佳的 Spot 執行個體

Spot Fleet 會每兩分鐘定期檢查機群中 Spot 執行個體的運作狀態。執行個體的運作狀態會是 `healthy` 或 `unhealthy`。

Spot Fleet 會根據 Amazon EC2 提供的狀態檢查資訊，來判定執行個體的運作狀態。如果執行個體狀態檢查或系統狀態檢查等運作狀態檢查，出現連續三次為 `unhealthy`，則該執行個體即為 `impaired`。如需詳細資訊，請參閱 Amazon EC2 執行個體的狀態檢查。

您可以設定機群以取代運作狀態不佳的 競價型執行個體。啟用運作狀態檢查取代後，Spot 執行個體會在被報告為 `unhealthy` 時被取代。當取代一個狀態不佳的 Spot 執行個體時，機群可能會低於其目標容量長達幾分鐘。

要求

- 運作狀態檢查替代僅支援維持目標容量的 Spot Fleets (類型 `maintain` 的機群)，不支援一次性的 Spot Fleets (類型 `request` 的機群)。
- 運作狀態檢查取代僅支援 競價型執行個體。此功能不支援 隨需執行個體。
- 您可以設定 Spot Fleet，只在建立執行個體時取代狀況不良的執行個體。
- 使用者只有在擁有呼叫 `ec2:DescribeInstanceStatus` 動作的許可時，才能使用運作狀態檢查取代功能。

Console

設定 Spot 機群以取代運作狀態不佳的 Spot 執行個體

1. 遵循下列步驟，在 使用已定義的參數建立 Spot 機群請求 中建立 Spot 機群。
2. 若要設定機群以取代運作狀態不佳的 Spot 執行個體，請展開其他啟動參數，然後在運作狀態檢查下，選取取代運作狀態不佳的執行個體。若要啟用此選項，首先必須選擇維護目標容量。

AWS CLI

設定 Spot 機群以取代運作狀態不佳的 Spot 執行個體

使用 request-spot-fleet 命令並搭配 `SpotFleetRequestConfig` 的 `ReplaceUnhealthyInstances` 屬性。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "TargetCapacity": 10,
        "ReplaceUnhealthyInstances": true
    }
}
```

PowerShell

設定 Spot 機群請求以取代運作狀態不佳的 Spot 執行個體

使用 Request-EC2SpotFleet cmdlet 搭配 `-SpotFleetRequestConfig_ReplaceUnhealthyInstance` 參數。

```
-SpotFleetRequestConfig_ReplaceUnhealthyInstance $true
```

標記新的或現有的 Spot 機群請求及其啟動的執行個體和磁碟區

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

為便於將 Spot 機群請求及其啟動的執行個體和磁碟區分類並進行管理，您可以利用自訂中繼資料來標記這些請求。您可以在建立 Spot Fleet 請求時或之後將標記指派給請求。同樣地，您可以在機群啟動執行個體和磁碟區時或之後，將標籤指派給執行個體和磁碟區。

當您標記機群請求時，由機群啟動的執行個體和磁碟區不會自動加上標記。您需要明確標記由機群啟動的執行個體和磁碟區。您可以選擇僅將標籤指派給機群請求，或僅指派給機群啟動的執行個體，或僅指派給連接至機群啟動之執行個體的磁碟區，或指派給前述所有項目。

> **Note**
>
> 您只能為連接至隨需執行個體的磁碟區加標籤。您無法為連接至 競價型執行個體 的磁碟區加標籤。

您可以使用 Amazon EC2 主控台或命令列工具來指派標籤。

如需標籤運作方式的詳細資訊，請參閱標記您的 Amazon EC2 資源。

目錄

- 必要條件
- 標記新的 Spot Fleet 以及執行個體和其啟動的磁碟區
- 標記現有 Spot Fleet
- 檢視 Spot Fleet 請求標籤

必要條件

授予使用者標記資源的許可。如需詳細資訊，請參閱範例：標籤資源。

授予使用者標記資源的許可

建立包含下列項目的 IAM 政策：

- ec2:CreateTags 動作。這會授予使用者建立標籤的許可。
- ec2:RequestSpotFleet 動作。這會授予使用者建立 Spot 機群請求的許可。
- 對於 Resource，您必須指定 "*"。這可讓使用者標記所有資源類型。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotFleetRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:RequestSpotFleet"
            ],
            "Resource": "*"
        }
    ]
}
```

Important

資源目前不支援spot-fleet-request資源層級許可。如果您指定 spot-fleet-request 作為資源，當您嘗試標記機群時，會收到未經授權的例外狀況。下列範例說明如何不要設定政策。

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags",
        "ec2:RequestSpotFleet"
    ],
    "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/*"
}
```

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

標記新的 Spot Fleet 以及執行個體和其啟動的磁碟區

若要標記新的 Spot 機群請求及其使用主控台啟動的執行個體和磁碟區

1. 遵循使用已定義的參數建立 Spot 機群請求程序。
2. 新增標籤的方式取決於您是手動設定機群，還是使用啟動範本。

   - 如果您是手動設定機群，請執行下列動作：

     若要新增標籤，請展開其他啟動參數，選擇新增標籤，然後輸入標籤鍵和值。針對每個標籤，重複執行上述動作。

     對於每個標籤，您可以標記 Spot Fleet 請求和具有相同標籤的執行個體。若要標記兩者，請確定已同時選取執行個體和機群。若只要標記 Spot Fleet 請求，請清除 Instances (執行個體)。若只要標記由機群啟動的執行個體，請清除 Fleet (機群)。

     > **Note**
     >
     > 當您手動設定機群時，並沒有標記磁碟區的選項。磁碟區標籤僅支援連接至 隨需執行個體 的磁碟區。當您手動設定機群時，無法指定隨需執行個體。

   - 如果您是使用啟動範本，請執行下列動作：

     若要將標籤新增至機群請求，請在標籤下選擇建立標籤，然後輸入標籤鍵和值。針對每個標籤，重複執行上述動作。

     若要標記機群中的資源，您必須在啟動範本中指定標籤。

使用 標記新的 Spot Fleet 請求及其啟動的執行個體和磁碟區 AWS CLI

若要在建立 Spot Fleet 請求時加上標記，並在機群啟動執行個體和磁碟區時給它們加上標記，請依下列方式設定 Spot Fleet 請求組態：

Spot Fleet 請求標籤：

- 在 SpotFleetRequestConfig 中指定 Spot Fleet 請求的標籤。
- 對於 ResourceType，請指定 spot-fleet-request。若指定其他值，機群請求將會失敗。
- 對於 Tags，指定金鑰值對。您可以指定多個金鑰值對。

執行個體標籤：

- 在 LaunchSpecifications 中指定執行個體的標籤。
- 對於 ResourceType，請指定 instance。若指定其他值，機群請求將會失敗。
- 對於 Tags，指定金鑰值對。您可以指定多個金鑰值對。

  或者，您可以在 Spot Fleet 請求中參照的啟動範本中，指定執行個體的標籤。

磁碟區標籤：

- 在 Spot Fleet 請求中參照的啟動範本中，指定磁碟區的標籤。不支援在 LaunchSpecifications 中標記磁碟區。

在下面的例子中，Spot Fleet 請求標記了兩個標籤：Key=Environment and Value=Production 和 Key=Cost-Center and Value=123。由機群啟動的執行個體標記了一個標籤 (這與 Spot Fleet 請求的標籤之一相同)：Key=Cost-Center and Value=123。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
```

```
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value":"Production"
                    },
                    {
                        "Key": "Cost-Center",
                        "Value":"123"
                    }
                ]
            }
        ]
    }
}
```

使用 標記 Spot 機群啟動的執行個體 AWS CLI

若要在機群啟動執行個體時加上標記，您可以在 Spot Fleet 請求中參照的啟動範本中指定標籤，或者您可以指定 Spot Fleet 請求組態中的標籤，如下所示：

- 在 `LaunchSpecifications` 中指定執行個體的標籤。

- 對於 ResourceType，請指定 instance。若指定其他值，機群請求將會失敗。
- 對於 Tags，指定金鑰值對。您可以指定多個金鑰值對。

在下列範例中，機群啟動的執行個體均標記了一個標籤：Key=Cost-Center and Value=123。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1
    }
}
```

使用 標記連接至 Spot 機群啟動之隨需執行個體的磁碟區 AWS CLI

若要在機群建立磁碟區時標記磁碟區，請在 Spot Fleet 請求中參考的啟動範本中指定標籤。

> **Note**
>
> 磁碟區標籤僅支援連接至 隨需執行個體 的磁碟區。您無法為連接至 競價型執行個體 的磁碟區加標籤。
> 不支援在 `LaunchSpecifications` 中標記磁碟區。

標記現有 Spot Fleet

若要使用主控台標記現有的 Spot Fleet 請求

建立 Spot 機群請求後，您可以使用主控台為機群請求新增標籤。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇標籤索引標籤，然後選擇建立標籤。

使用 標記現有的 Spot Fleet 請求 AWS CLI

使用 create-tags 命令來標記現有資源。在下面的例子中，現有的 Spot Fleet 請求會使用 Key=purpose 和 Value=test 標籤。

```
aws ec2 create-tags \
    --resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
    --tags Key=purpose,Value=test
```

檢視 Spot Fleet 請求標籤

若要使用主控台檢視 Spot Fleet 請求標籤

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求並選擇 Tags (標籤)。

若要描述 Spot Fleet 請求標籤

使用 describe-tag 命令檢視指定資源的標籤。在下面的範例中，會描述指定 Spot Fleet 請求的標籤。

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
    "Tags": [
        {
            "Key": "Environment",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Production"
        },
        {
            "Key": "Another key",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Another value"
        }
    ]
}
```

您也可以透過描述 Spot Fleet 請求來檢視 Spot Fleet 請求的標籤。

使用 describe-spot-fleet-requests 命令來檢視指定 Spot Fleet 請求的組態，這包括針對機群請求指定的任何標籤。

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```

```
{
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-
tagging-role",
                "LaunchSpecifications": [
```

```
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",
                    "Value": "Another value"
                }
            ]
        }
    ]
}
```

說明 Spot 機群請求、其執行個體和事件歷史記錄

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

您可以說明您的 Spot 機群組態、您 Spot 機群中的執行個體，以及您 Spot 機群的事件歷史記錄。

Console

說明您的 Spot 機群

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。ID 開頭為 sfr-。若要查看組態的詳細資訊，請選擇描述。
4. 若要列出 Spot Fleet 的 Spot 執行個體，請選擇 Instances (執行個體)。
5. 若要檢視 Spot Fleet 的歷史記錄，請選擇 History (歷史記錄)。

AWS CLI

說明 Spot 機群請求標籤

使用 describe-spot-fleet-requests 命令。

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

說明指定 Spot 機群請求的執行中執行個體

使用 describe-fleet-instances 命令。

```
aws ec2 describe-spot-fleet-instances \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

說明指定 Spot 機群請求的事件歷史記錄

使用 describe-spot-fleet-request-history 命令。

```
aws ec2 describe-spot-fleet-request-history \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2024-05-18T00:00:00Z
```

PowerShell

說明 Spot 機群請求標籤

使用 Get-EC2SpotFleetRequest cmdlet。

```
Get-EC2SpotFleetRequest
```

說明指定 Spot 機群請求的執行中執行個體

使用 Get-EC2SpotFleetInstance cmdlet。

```
Get-EC2SpotFleetInstance `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

說明指定 Spot 機群請求的事件歷史記錄

使用 Get-EC2SpotFleetRequestHistory cmdlet。

```
Get-EC2SpotFleetRequestHistory `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -UtcStartTime 2024-05-18T00:00:00Z
```

修改 Spot Fleet 請求

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

您可以修改作用中的 Spot Fleet 請求，來完成下列任務：

- 增加總目標容量和隨需部分
- 減少總目標容量和隨需部分

當您增加總目標容量時，Spot 機群會根據其 Spot 機群請求的配置策略，啟動額外的 Spot 執行個體。當您增加隨需部分時，Spot Fleet 會啟動額外的隨需執行個體。

當您減少總目標容量時，Spot 機群會取消超出新目標容量的所有開放請求。您可以請求 Spot Fleet 終止 Spot 執行個體，直到機群大小達到新的目標容量。如果分配策略為 `diversified`，則 Spot Fleet 會在集區中終止執行個體。或者，您可以請求該 Spot Fleet 將機群保持為目前的大小，但不能取代任何中斷的或您手動終止的 Spot 執行個體。

考量事項

- 您無法修改一次性的 Spot Fleet 請求。當您建立 Spot Fleet 請求時，如果已選取 Maintain target capacity (維護目標容量)，則只能修改 Spot Fleet 請求。

- 當 Spot Fleet 因為目標容量減少而終止執行個體時，執行個體會收到 Spot 執行個體中斷通知。

Console

修改 Spot 機群請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇動作和修改目標容量。
5. 在修改目標容量中，執行下列動作：

   a. 輸入新的目標容量和隨需部分。
   b. (選用) 如果您正在減少目標容量，但希望叢集保持目前的大小，請清除終止執行個體。
   c. 選擇提交。

AWS CLI

修改 Spot 機群請求

請使用 modify-spot-fleet-request 命令更新指定 Spot Fleet 請求的目標容量。

```
aws ec2 modify-spot-fleet-request \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity 20
```

如果要減少所指定 Spot Fleet 的目標容量，而不會因此終止任何 Spot 執行個體，您可以如下修改先前的命令：

```
aws ec2 modify-spot-fleet-request \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity 10 \
    --excess-capacity-termination-policy NoTermination
```

PowerShell

修改 Spot 機群請求

使用 Edit-EC2SpotFleetRequest cmdlet，更新指定 Spot 機群請求的目標容量。

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20
```

如果要減少所指定 Spot Fleet 的目標容量，而不會因此終止任何 Spot 執行個體，您可以如下修改先前的命令：

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

取消 (刪除) Spot 機群請求

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

如果不再需要 Spot 機群，則可取消 Spot 機群請求，即會刪除請求。取消機群請求後，也會取消與機群相關的所有 Spot 請求，因此不會啟動新的 Spot 執行個體。

刪除時 Spot 機群請求時，必須指定是否也要終止其全部執行個體。這包含隨需執行個體和 Spot 執行個體。

> **Warning**
>
> 終止執行個體的操作為永久性且無法復原。
> 終止執行個體後，您將無法再連線至該執行個體，且無法復原該執行個體。所有設定為終止時刪除的已連接 Amazon EBS 磁碟區，也會被永久刪除且無法復原。儲存在執行個體儲存體磁碟區的所有資料將永久遺失。如需詳細資訊，請參閱終止執行個體的運作方式。
> 在終止執行個體之前，確認您已將終止使用後需要保留的全部資料備份至持久性儲存體。

如果指定在刪除機群請求時必須終止執行個體，則機群請求會進入 `cancelled_terminating` 狀態。否則會進入 `cancelled_running` 狀態，且執行個體會一直執行直到中斷或您手動終止。

限制

- 您可以在單一請求中取消多達 100 個機群。如果您超過指定的數量，則不會取消任何機群。

Console

取消 (刪除) Spot 機群請求

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇動作、取消請求。
5. 在取消 Spot 請求對話方塊中，執行下列操作：
   a. 若要在取消 Spot 機群請求的同時終止關聯的執行個體，請將終止執行個體核取方塊保持為選取狀態。若要取消 Spot 機群請求，而不終止關聯的執行個體，請取消勾選終止執行個體核取方塊。
   b. 選擇確認。

AWS CLI

取消 (刪除) Spot 機群請求並終止其執行個體

使用 cancel-spot-fleet-requests 命令並搭配 `--terminate-instances` 選項。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

以下為範例輸出。

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
```

```
    "UnsuccessfulFleetRequests": []
}
```

取消 (刪除) Spot 機群請求，而不終止其執行個體

請改用 `--no-terminate-instances` 選項來修改之前的範例。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

以下為範例輸出。

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

PowerShell

取消 (刪除) Spot 機群請求並終止其執行個體

使用 Stop-EC2SpotFleetRequest cmdlet 搭配 `-TerminateInstance` 參數。

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

取消 (刪除) Spot 機群請求，而不終止其執行個體

透過變更 `-TerminateInstance` 參數的值來修改之前的範例。

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

了解 Spot 機群的自動擴展

> **Important**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

自動擴展讓您的 Spot 機群可根據需求增加或減少其目標容量。Spot 機群可以利用自動擴展，在指定的範圍內啟動執行個體 (擴展) 或終止執行個體 (縮減)，以回應一個或多個擴展政策。

透過結合 Amazon EC2、Amazon CloudWatch 和 Application Auto Scaling API 可實現 Spot Fleet 的自動擴展。Spot Fleet 請求是使用 Amazon EC2 建立的，警示是使用 CloudWatch 建立的，而擴展政策是使用 Application Auto Scaling 建立的。

自動擴展的類型

Spot Fleet 支援以下類型的自動擴展：

- 目標追蹤擴展 – 透過為特定指標設定目標值，增加或減少機群目前的容量。這類似於您家中溫控器維持溫度的方式 — 您只要選擇所需溫度，溫控器就會完成其餘的動作。
- 步進擴展 – 根據一組擴展調整 (也稱為步進調整)，來增加或減少機群目前的容量，其會根據警示違規的程度而有不同。
- 排程的擴展 – 根據日期和時間，來增加或減少機群目前的容量。

考量事項

針對您的 Spot 機群使用自動擴展時，請考慮下列事項：

- 執行個體權重 – 如果您要使用執行個體權重，請記住，Spot 機群可視需要超過目標容量。已實現的容量可以是浮點數，但目標容量必須是整數，因此 Spot Fleet 會四捨五入為下一個整數。如果檢視警示觸發時擴展政策的結果，您必須將這些行為列入考量。例如，假設目標容量為 30、已實現的容量為 30.1，則規模調整政策會減掉 1。當警示觸發時，自動擴展程序會從 30.1 減掉 1 而得出 29.1，然後再四捨五入為 30，因此不需採取規模調整動作。在另一個範例中，假設選擇了 2、4 和 8 的執行個體權重，而目標容量為 10，但沒有可用的權重 2 執行個體，因此 Spot Fleet 針對已實現的容量 12 佈建權重為 4 和 8 的執行個體。如果規模調整政策將目標容量減少了 20%，而且警示已觸發，則自動擴展程序會從 12 減掉 12*0.2 而得出 9.6，然後再四捨五入為 10，因此不需採取規模調整動作。

- 冷卻時間 – 您為 Spot 機群建立的擴展政策支援冷卻時間。這是在擴展活動完成之後，先前的觸發相關擴展活動還能影響未來擴展活動的秒數。如果使用向外擴展政策，則在冷卻時間仍有效時，由先前起始冷卻時間的向外擴展事件所新增的容量，會在計算時列入下次向外擴展所需容量的一部分。這種做法的目的是連續的向外擴展 (但並非過度)。如果使用向內擴展政策，在冷卻時間尚未到期之前，會利用此期間來封鎖請求中後續的擴展。這種做法的目的是保守地進行擴展，以保障應用程式的可用性。不過，如果在向內擴展後的冷卻時間內，有另一個警示觸發了向外擴展政策，則自動擴展功能會立即擴展可擴展的目標。
- 使用詳細監控 – 建議您依據每分鐘更新的執行個體指標進行擴展，因為這樣能更快速地響應使用率變更。5 分鐘頻率的擴展指標可能會導致較慢的回應時間和用到過時的指標資料進行擴展。若要將執行個體的指標資料每隔 1 分鐘傳送至 CloudWatch，您必須指定啟用執行個體的詳細監控。如需詳細資訊，請參閱管理 EC2 執行個體的詳細監控及使用已定義的參數建立 Spot 機群請求。
- AWS CLI – 如果您使用 AWS CLI 來設定 Spot Fleet 的擴展，您將使用應用程式自動擴展命令。

Spot Fleet 自動擴展所需的 IAM 許可

透過結合 Amazon EC2、Amazon CloudWatch 和 Application Auto Scaling API 可實現 Spot Fleet 的自動擴展。Spot Fleet 請求是使用 Amazon EC2 建立的，警示是使用 CloudWatch 建立的，而擴展政策是使用 Application Auto Scaling 建立的。除了 Spot 機群和 Amazon EC2 所需的 IAM 許可之外，存取機群擴展設定的使用者還必須擁有支援自動擴展服務的適當許可。

若要針對 Spot 機群使用自動擴展，使用者必須具備許可，才能使用下列範例政策中所顯示的各項動作。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
```

```
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

您也可以建立自己的 IAM 政策，以便具有更精細的 Application Auto Scaling API 呼叫許可。如需詳細資訊，請參閱「Application Auto Scaling 使用者指南」https://docs.aws.amazon.com/autoscaling/application/userguide/auth-and-access-control.html中適用於 Application Auto Scaling 的 Identity and Access Management。

Application Auto Scaling 服務也需要描述 Spot Fleet 和 CloudWatch 警示的許可，以及代表您修改 Spot Fleet 目標容量的許可。如果您為 Spot Fleet 啟用自動擴展，則會建立名為 `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest` 的服務連結角色。此服務連結角色會授與 Application Auto Scaling 許可以描述政策的警示、監控機群目前的容量，以及修改機群的容量。Application Auto Scaling 的原始受管 Spot Fleet 角色為 `aws-ec2-spot-fleet-autoscale-role`，但已不再需要。服務連結的角色預設為 Application Auto Scaling。如需詳細資訊，請參閱「Application Auto Scaling 使用者指南」中的適用於 Application Auto Scaling 的服務連結角色。

目標追蹤擴展：透過為特定指標設定目標值，擴展 Spot 機群

您可以使用目標追蹤擴展，藉由選取指標及設定目標值來建立目標追蹤擴展政策。接著，Spot 機群會建立及管理可觸發擴展政策的 CloudWatch 警示，並根據選擇的指標和目標值來計算擴展。擴展政策會視需要新增或移除執行個體來調整容量，讓指標保持在等於或接近指定目標值。目標追蹤政策不僅會讓指標保持在接近目標值，還會進行調整，以配合指標中因負載模式波動而造成的波動，並將快速容量波動降到最低。

您可以為 Spot 機群建立多個目標追蹤擴展政策，但前提是每個政策都是使用不同的指標。機群會根據指定最大機群容量的政策來進行擴展。這可讓您涵蓋多個案例，以確保有足夠的容量可供您的應用程式工作負載使用。

為了確保應用程式的可用性，機群可以配合指標按比例快速地擴展規模，但是以更漸進的方式縮減規模。

當 Spot 機群因為目標容量減少而終止 Spot 執行個體時，執行個體會收到 Spot 執行個體中斷通知。

> **Note**
>
> 請勿編輯或刪除 CloudWatch 警示，Spot Fleet 管理它們用於目標追蹤擴展政策。當您刪除目標追蹤擴展政策時，Spot Fleet 會自動刪除警示。

先決條件

- Spot 機群請求必須具有 `maintain` 的請求類型。類型 `request` 的請求不支援自動擴展功能。
- 設定 Spot Fleet 自動擴展所需的 IAM 許可。
- 請參閱考量事項。

設定目標追蹤政策

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的自動擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至自動擴展區段。
5. 如果未設定自動擴展，請選擇設定。
6. 利用容量擴展的範圍來設定機群的最低和最高容量。自動擴展功能不會將機群的規模擴展到小於最低容量或大於最高容量。
7. 針對 Policy Name (政策名稱)，請輸入政策的名稱。
8. 選擇目標指標。
9. 輸入指標的目標值。
10. 針對冷卻時間，請指定一個新值 (以秒為單位) 或保留預設值。
11. (選用) 若要略過根據目前的組態建立縮減政策，請選取停用縮減。您可以使用不同的組態來建立向內擴展政策。
12. 選擇儲存。

使用 設定目標追蹤政策 AWS CLI

1. 使用 register-scalable-target 命令，將 Spot Fleet 請求註冊為可擴充的目標。
2. 使用 put-scaling-policy 命令，來建立擴展政策。

步驟擴展：使用步驟擴展政策來擴展 Spot 機群

如果使用步進規模調整政策，您可以指定 CloudWatch 警示來觸發規模調整程序。例如，如果想在 CPU 的利用率達到某個水準時擴展，請使用 Amazon EC2 所提供的 `CPUUtilization` 指標來建立警示。

建立步進規模調整政策時，您必須指定下列規模調整類型的其中一種：

- 新增 – 根據指定的容量單位數量或指定的目前容量百分比，來增加機群的目標容量。
- 移除 – 根據指定的容量單位數量或指定的目前容量百分比，來減少機群的目標容量。
- 設定為 – 將機群的目標容量設定為指定的容量單位數量。

警示觸發時，自動擴展程序會利用已實現的容量和規模調整政策，來計算出新的目標容量，然後據以更新目標容量。例如，假設目標容量為和已實現的容量為 10，而且規模調整政策會增加 1。當警示觸發時，自動擴展程序會將 10 加上 1 而得出 11，因此 Spot Fleet 會啟動 1 個執行個體。

當 Spot 機群因為目標容量減少而終止 Spot 執行個體時，執行個體會收到 Spot 執行個體中斷通知。

先決條件

- Spot 機群請求必須具有 `maintain` 的請求類型。類型 `request` 的請求不支援自動擴展功能。
- 設定 Spot Fleet 自動擴展所需的 IAM 許可。
- 思考哪些 CloudWatch 指標對您的應用程式是重要的。您可以根據 提供的指標 AWS 或您自己的自訂指標來建立 CloudWatch 警示。
- 對於您將在擴展政策中使用的 AWS 指標，如果提供指標的服務預設不會啟用 CloudWatch 指標集合。
- 請參閱考量事項。

建立 CloudWatch 警示

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，展開警示，然後選擇所有警示。

3. 選擇建立警示。
4. 在指定指標和條件頁面上，選擇選取指標。
5. 選擇 EC2 Spot、機群請求指標，選取指標 (例如，TargetCapacity)，然後選擇選取指標。

   指定指標和條件頁面隨即出現，顯示您選取指標的圖形及其他資訊。
6. 針對期間，選擇警示的評估期間，例如 1 分鐘。評估警示時，每個期間都會彙整為一個資料點。

   > **Note**
   >
   > 期間越短會建立更敏感的警示。

7. 針對條件，透過定義閾值條件來定義警示。例如，您可以定義一個閾值，設定每當指標值大於或等於 80% 時，即會觸發警示。
8. 在其他組態 下，針對要產生警示的資料點，指定多少資料點 (評估期間) 必須處於 ALARM 狀態才會觸發警示，例如，1 評估期間或 3 個中的 2 個評估期間。這樣建立的警示會在連續超過這麼多期間時進入 ALARM 狀態。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的評估警示。
9. 對於遺失資料處理，選擇其中之一個選項 (或保留預設的 將遺失資料視為遺失)。如需詳細資訊，請參閱「Amazon CloudWatch 使用者指南」中的設定 CloudWatch 警示如何處理遺失資料。
10. 選擇下一步。
11. (選用) 若要接收擴展事件的通知，針對通知，您可以選擇或建立您要用於接收通知的 Amazon SNS 主題。否則，您可以立即刪除通知，並根據需要在之後新增通知。
12. 選擇下一步。
13. 在新增名稱和描述下，輸入警示的名稱和描述，然後選擇下一步。
14. 選擇 Create alarm (建立警示)。

為您的 Spot 機群設定步進擴展政策

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的自動擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至自動擴展區段。
5. 如果未設定自動擴展，請選擇設定。

6. 利用容量擴展的範圍來設定機群的最低和最高容量。擴展政策不會將機群的規模擴展到小於最低容量或大於最高容量。
7. 在擴展政策下，針對政策類型，選擇步驟擴展政策。
8. 一開始，擴展政策包含名為 ScaleUp 和 ScaleDown 的步驟擴展政策。您可以完成這些政策，或是選擇移除政策來刪除政策。您也可以選擇新增政策。
9. 若要定義政策，請執行下列的動作：

   a. 針對 Policy Name (政策名稱)，請輸入政策的名稱。

   b. 針對政策觸發器，選擇現有的警示，或是選擇建立警示來開啟 Amazon CloudWatch 主控台，然後建立警示。

   c. 針對修改容量，請定義要擴展的量以及步驟調整的下限與上限。您可以新增或移除特定數量的執行個體，或現有機群大小的百分比，或將機群設為確切大小。

      例如，若要建立可將機群容量增加 30% 的步驟擴展政策，請選擇新增，在下一個欄位中輸入 30，然後選擇百分比。根據預設，新增政策的下限為警示閾值，而上限為無限大正數 (+) 值。根據預設，移除政策的上限為警示閾值，而下限為無限小負數 (-) 值。

   d. (選用) 若要新增另一個步驟，請選擇新增步驟。

   e. 針對冷卻時間，請指定一個新值 (以秒為單位) 或保留預設值。
10. 選擇儲存。

使用 設定 Spot 機群的步驟擴展政策 AWS CLI

1. 使用 register-scalable-target 命令，將 Spot Fleet 請求註冊為可擴充的目標。
2. 使用 put-scaling-policy 命令，來建立擴展政策。
3. 使用 put-metric-alarm 命令，來建立觸發擴展政策的警示。

排程擴展：依排程擴展 Spot 機群

按照排程擴展機群，讓您能夠因應可預測的需求變化，據以擴展您的應用程式。透過建立排程動作，您可以指示 Spot 機群在特定時間執行擴展活動。若要建立排程的動作，您必須指定現有的 Spot 機群、應該發生擴展活動的時間，以及所需的最小和最大容量。排程動作可設定為擴展一次，或依週期性排程擴展。如果您需要變更，可以編輯或刪除排程的動作。

先決條件

- 只能針對現有的 Spot 機群建立排程的動作。您在建立 Spot 機群時，無法建立排程的動作。

- Spot 機群請求必須具有 `maintain` 的請求類型。類型 `request` 的請求不支援自動擴展功能。
- 設定 Spot Fleet 自動擴展所需的 IAM 許可。
- 請參閱考量事項。

建立一次性排程動作

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的排程擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至排程擴展區段。
5. 選擇建立排程動作。
6. 在名稱 中，為排程動作指定名稱。
7. 為容量下限、容量上限或兩者輸入一值。
8. 針對重複，選擇一次。
9. (選用) 針對開始時間、結束時間或兩者選擇日期和時間。
10. 選擇建立。

若要建立週期性排程動作

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的排程擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至排程擴展區段。
5. 在名稱 中，為排程動作指定名稱。
6. 為容量下限、容量上限或兩者輸入一值。
7. 針對重複，選擇其中一個預先定義的排程 (例如，每天)，或選擇 自訂，然後輸入 Cron 表達式。如需排程擴展所支援 Cron 表達式的詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的 Cron 表達式。
8. (選用) 針對開始時間、結束時間或兩者選擇日期和時間。

9. 選擇提交。

編輯排程動作

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的排程擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至排程擴展區段。
5. 選取排程動作，然後選擇動作、編輯。
6. 執行需要的變更，然後選擇提交。

刪除排程動作

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在瀏覽窗格中，選擇 Spot 請求。
3. 選取您的 Spot Fleet 請求。
4. 選擇畫面底部附近的排程擴展索引標籤。如果您已選取 Spot 機群的連結，則沒有索引標籤；而是向下捲動至排程擴展區段。
5. 選取排程動作，然後選擇動作、刪除。
6. 出現確認提示時，請選擇 Delete (刪除)。

使用 管理排程擴展 AWS CLI

使用下列命令：

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

監控您的 EC2 機群或 Spot 機群

EC2 機群或 Spot 機群的有效監控，是維持最佳效能並確保可靠性的必要條件。有各種工具可協助您達成此目標，包括本主題所涵蓋的 Amazon CloudWatch 和 Amazon EventBridge。

您可以使用 CloudWatch 收集和追蹤指標、設定警示及自動對機群狀態的變更做出反應。

您可以使用 EventBridge 透過程式設計方式監控和回應機群發出的事件。透過在 EventBridge 中定義規則，您可以自動化對特定機群事件的回應，例如執行個體終止或機群狀態變更，從而提高您的營運效率。

主題

- 使用 CloudWatch 監控 EC2 機群或 Spot 機群
- 使用 Amazon EventBridge 監控和以程式設計方式回應 EC2 機群或 Spot 機群發出的事件

使用 CloudWatch 監控 EC2 機群或 Spot 機群

您可以使用本節所述的 Amazon CloudWatch 指標來監控 EC2 機群或 Spot 機群。

> **Important**
>
> 為確保準確性，建議您在使用這些指標時，啟用詳細監控功能。如需詳細資訊，請參閱管理 EC2 執行個體的詳細監控。

如需使用 CloudWatch 的詳細資訊，請參閱 使用 CloudWatch 監控您的執行個體。

EC2 機群和 Spot 機群指標

AWS/EC2Spot 命名空間包含您機群的下列指標，加上機群中 Spot 執行個體的 CloudWatch 指標。如需詳細資訊，請參閱執行個體指標。

指標	說明
AvailableInstancePoolsCount	機群請求中指定的 Spot 容量集區。 單位：Count
BidsSubmittedForCapacity	Amazon EC2 已提交機群請求的容量。 單位：Count
EligibleInstancePoolCount	在機群請求中指定的 Spot 容量集區，Amazon EC2 可在其中實現請求。在您願意為 Spot 執行個體支付的最高價格低

指標	說明
	於 Spot 價格或 Spot 價格高於隨需執行個體價格的集區中，Amazon EC2 不會實現請求。 單位：計數
FulfilledCapacity	Amazon EC2 已履行的容量。 單位：Count
MaxPercentCapacityAllocation	在機群請求中指定的所有機群集區的 PercentCapacityAllocation 的最大值。 單位：百分比
PendingCapacity	TargetCapacity 與 FulfilledCapacity 之間的差異。 單位：計數
PercentCapacityAllocation	已為指定維度的 Spot 容量集區配置的容量。若要取得所有 Spot 容量集區所記錄的最大值，請使用 MaxPercentCapacityAllocation 。 單位：百分比
TargetCapacity	機群請求的目標容量。 單位：Count
TerminatingCapacity	由於佈建容量而終止的容量大於目標容量。 單位：計數

如果指標的測量單位是 Count，則最有用的統計資訊是 Average。

EC2 機群和 Spot 機群維度

若要篩選機群的資料，請使用下列維度。

維度	描述
AvailabilityZone	依可用區域篩選資料。
FleetRequestId	依機群請求篩選資料。
InstanceType	依執行個體類型篩選資料。

檢視 EC2 機群或 Spot 機群的 CloudWatch 指標

您可以使用 Amazon CloudWatch 主控台來檢視機群的 CloudWatch 指標。這些指標會以監控圖表的形式顯示。如果機群正在使用中，這些圖表會顯示資料點。

指標會先依據命名空間進行分組，然後再根據各命名空間內不同的維度組合來分組。例如，您可以按機群請求 ID、執行個體類型或可用區域來檢視所有機群指標或機群指標群組。

檢視機群指標

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，擴充指標，然後選擇所有指標。
3. 選擇 EC2 Spot 命名空間。

> **Note**
>
> 若未顯示 EC2 Spot 命名空間，這有兩個原因。您可能從未在 區域中使用過 EC2 機群或 Spot 機群 - 只有您使用 AWS 的服務才會將指標傳送至 Amazon CloudWatch。或是因為您過去內兩週未曾在該區域使用 EC2 機群或 Spot 機群，因此沒有顯示此命名空間。

4. 若要根據維度來篩選指標，請選擇下列其中一個選項：
   - 機群請求指標 – 依機群請求分組
   - 根據可用區域 – 依機群請求和可用區域分組
   - 根據執行個體類型 – 依機群請求和執行個體類型分組
   - 根據可用區域/執行個體類型 – 依機群請求、可用區域和執行個體類型分組
5. 若要檢視指標的資料，請勾選指標旁的核取方塊。

使用 Amazon EventBridge 監控和以程式設計方式回應 EC2 機群或 Spot 機群發出的事件

當 EC2 機群或 Spot 機群的狀態發生變化時，機群會發出通知。通知會以傳送至 Amazon EventBridge (先前稱為 Amazon CloudWatch Events) 的事件形式提供使用。盡可能發出事件。

使用 Amazon EventBridge，您可以建立觸發程式設計動作以回應事件的規則。例如，您可以建立兩個 EventBridge 規則：一個是在機群狀態變更時觸發的規則，另一個是在機群中的執行個體終止時觸發的規則。在此例中，如果機群狀態變化，您可以設定第一個規則，以便規則可叫用 SNS 主題來向您傳送電子郵件通知。如果機群中的執行個體終止，您可以設定第二個規則，以便規則可叫用 Lambda 函數來啟動新執行個體。

> **Note**
>
> 只有類型 `maintain` 和 `request` 的機群會發出事件。類型 `instant` 的機群不會發出事件，因為它們提交一次性的同步請求，而且在回應中會立即知道機群的狀態。若要使用 Amazon EventBridge 監控機群事件，請求類型必須為 `maintain` 或 `request`。

如需如何描述機群事件歷史記錄的說明，請參閱 說明 EC2 機群的事件歷史記錄。

主題

- 建立 Amazon EventBridge 規則來監控 EC2 機群或 Spot 機群事件
- EC2 機群 事件類型
- Spot 機群事件類型

建立 Amazon EventBridge 規則來監控 EC2 機群或 Spot 機群事件

當發出 EC2 機群或 Spot 機群的狀態變化通知時，該通知會以 JSON 檔案，以事件的形式傳送至 Amazon EventBridge。如果 EventBridge 偵測到事件模式符合規則中定義的模式，EventBridge 會叫用規則中指定的目標 (或多個目標)。

您可以撰寫 EventBridge 規則，根據相符的事件模式自動執行動作。

事件中的下列欄位會形成規則中定義的事件模式：

`"source": "aws.ec2fleet"`

識別事件來自 EC2 機群。

`"detail-type": "EC2 Fleet State Change"`

識別事件類型。

`"detail": { "sub-type": "submitted" }`

識別事件子類型。

如需 EC2 機群和 Spot 機群事件和範例事件資料的清單，請參閱 EC2 機群 事件類型 和 Spot 機群事件類型。

範例

- 建立傳送通知的 EventBridge 規則
- 建立一個 EventBridge 規則來觸發 Lambda 函數

建立傳送通知的 EventBridge 規則

下列範例會建立 EventBridge 規則，以在每次 Amazon EC2 發出 EC2 機群狀態變更通知時傳送電子郵件、文字訊息或行動推送通知。將發出此範例中的訊號作為 `EC2 Fleet State Change` 事件，這會觸發規則定義的動作。

先決條件

建立 EventBridge 規則之前，您必須針對電子郵件、文字訊息或行動推送通知建立 Amazon SNS 主題。

建立 EventBridge 規則以在 EC2 機群狀態變更時傳送通知

1. 在 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 選擇 Create rule (建立規則)。
3. 針對 Define rule detail (定義規則詳細資訊) 執行下列動作：
   a. 輸入規則的Name (名稱)，或者輸入描述。

      在同一個區域和同一個事件匯流排上，規則不能與另一個規則同名。
   b. 針對事件匯流排選擇預設值。當您帳戶中 AWS 的服務產生事件時，一律會前往您帳戶的預設事件匯流排。
   c. 針對規則類型，選擇具有事件模式的規則。
   d. 選擇下一步。

4. 針對 Build event pattern (建置事件模式) 執行下列動作：

   a. 在 Event source (事件來源) 欄位中，選擇 AWS events or EventBridge partner events (事件或 EventBridge 合作夥伴事件)。

   b. 針對此範例的 Event pattern (事件模式)，您需指定下列事件模式，以便與 `EC2 Fleet Instance Change` 事件相符。

```
{
 "source": ["aws.ec2fleet"],
 "detail-type": ["EC2 Fleet Instance Change"]
}
```

      若要新增事件模式，您可以透過選擇 Event pattern form (事件模式表單) 使用範本，或者透過選擇 Custom pattern (JSON editor) (自定義模式 (JSON 編輯器)) 指定自有模式，如下：

      i. 若要使用範本建立事件模式，請執行下列動作：

         A. 選擇 Event pattern form (事件模式表單)。

         B. 在 Event source (事件來源) 欄位中，選擇 AWS services (服務)。

         C. 針對 AWS Service (服務)，請選擇 EC2 Fleet (EC2 機群)。

         D. 針對 Event type (事件類型)，選擇 EC2 Fleet Instance Change (EC2 機群執行個體變更)。

         E. 若要自訂範本，請選擇 Edit pattern (編輯模式) 並進行變更以與範例事件模式相符。

      ii. (替代) 若要指定自訂事件模式，請執行下列動作：

         A. 選擇 Custom pattern (JSON editor) (自訂模式 (JSON 編輯器))。

         B. 在 Event pattern (事件模式) 方塊中，為此範例新增事件模式。

   c. 選擇下一步。

5. 針對 Select target(s) (選取目標) 執行下列動作：

   a. 在 Target types (目標類型) 欄位中，選擇 AWS service (服務)。

   b. 針對 Select a target (選取目標) 選取 SNS topic (SNS 主題)，以便在事件發生時傳送電子郵件、簡訊或行動裝置推送通知。

   c. 針對 Topic (主題)，請選擇現有的主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

d. (選用) 在 Additional settings (其他設定) 下，您可以選擇性地設定其他設定。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的建立對事件做出反應的 Amazon EventBridge 規則 (步驟 16)。

e. 選擇下一步。

6. (選用) 針對 Tags (標籤)，您可以選擇性地將一或多個標籤指派給您的規則，然後選擇 Next (下一步)。

7. 針對 Review and create (檢閱和建立) 執行下列動作：

a. 檢閱規則的詳細資訊，然後視需求進行修改。

b. 選擇建立規則。

如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 規則和 Amazon EventBridge 事件模式。

建立一個 EventBridge 規則來觸發 Lambda 函數

下列範例會建立 EventBridge 規則，以在每次 Amazon EC2 因執行個體啟動時發出 EC2 機群執行個體變更通知時觸發 Lambda 函數。將發出此範例中的訊號作為 `EC2 Fleet Instance Change` 事件 (子類型 `launched`)，這會觸發規則定義的動作。

在建立 EventBridge 規則之前，您必須建立 Lambda 函數。

建立要在 EventBridge 規則中使用的 Lambda 函數

1. 在 https：//https://console.aws.amazon.com/lambda/ 開啟 AWS Lambda 主控台。

2. 選擇 Create function (建立函式)。

3. 輸入函數的名稱、設定程式碼，然後選擇 Create function (建立函數)。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的建立您的第一個 Lambda 函式。

建立 EventBridge 規則以在執行個體處於 EC2 機群變更狀態時觸發 Lambda 函數

1. 在 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。

2. 選擇 Create rule (建立規則)。

3. 針對 Define rule detail (定義規則詳細資訊) 執行下列動作：

a. 輸入規則的Name (名稱)，或者輸入描述。

在同一個區域和同一個事件匯流排上，規則不能與另一個規則同名。

b. 針對事件匯流排選擇預設值。當您帳戶中 AWS 的服務產生事件時，一律會前往您帳戶的預設事件匯流排。

c. 針對規則類型，選擇具有事件模式的規則。

d. 選擇下一步。

4. 針對 Build event pattern (建置事件模式) 執行下列動作：

a. 在 Event source (事件來源) 欄位中，選擇 AWS events or EventBridge partner events (事件或 EventBridge 合作夥伴事件)。

b. 針對此範例的 Event pattern (事件模式)，您需指定下列事件模式，以便與 `EC2 Fleet Instance Change` 事件和 `launched` 子類型相符。

```
{
 "source": ["aws.ec2fleet"],
 "detail-type": ["EC2 Fleet Instance Change"],
 "detail": {
   "sub-type": ["launched"]
}
```

若要新增事件模式，您可以透過選擇 Event pattern form (事件模式表單) 使用範本，或者透過選擇 Custom pattern (JSON editor) (自定義模式 (JSON 編輯器)) 指定自有模式，如下：

i. 若要使用範本建立事件模式，請執行下列動作：

A. 選擇 Event pattern form (事件模式表單)。

B. 在 Event source (事件來源) 欄位中，選擇 AWS services (服務)。

C. 針對 AWS Service (服務)，請選擇 EC2 Fleet (EC2 機群)。

D. 針對 Event type (事件類型)，選擇 EC2 Fleet Instance Change (EC2 機群執行個體變更)。

E. 選擇 Edit pattern (編輯模式)，並新增 `"detail": {"sub-type": ["launched"]` 以符合範例事件模式。若要獲得正確的 JSON 格式，請在前方括號 (`]`) 之後插入一個逗號 (`,`)。

ii. (替代) 若要指定自訂事件模式，請執行下列動作：

A. 選擇 Custom pattern (JSON editor) (自訂模式 (JSON 編輯器))。

B. 在 Event pattern (事件模式) 方塊中，為此範例新增事件模式。

c. 選擇下一步。

5. 針對 Select target(s) (選取目標) 執行下列動作：

   a. 在 Target types (目標類型) 欄位中，選擇 AWS service (服務)。

   b. 針對 Select a target (選取目標) 選取 SNS topic (SNS 主題)，以便在事件發生時傳送電子郵件、簡訊或行動裝置推送通知。

   c. 針對 Topic (主題)，請選擇 Lambda function (Lambda 函數)，然後在 Function (函數) 中選擇您建立的函數，以便在事件發生時回應。

   d. (選用) 在 Additional settings (其他設定) 下，您可以選擇性地設定其他設定。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」中的建立對事件做出反應的 Amazon EventBridge 規則 (步驟 16)。

   e. 選擇下一步。

6. (選用) 針對 Tags (標籤)，您可以選擇性地將一或多個標籤指派給您的規則，然後選擇 Next (下一步)。

7. 針對 Review and create (檢閱和建立) 執行下列動作：

   a. 檢閱規則的詳細資訊，然後視需求進行修改。

   b. 選擇建立規則。

如需如何建立 Lambda 函數和執行 Lambda 函數的 EventBridge 規則的教學課程，請參閱 AWS Lambda 開發人員指南中的教學課程：使用 EventBridge 記錄 Amazon EC2 執行個體的狀態。

EC2 機群 事件類型

有五種 EC2 機群 事件類型。每個事件類型都有數個子類型。

Event types (事件類型)

- EC2 機群狀態變更
- EC2 機群Spot 執行個體請求變更
- EC2 機群執行個體變更
- EC2 機群資訊
- EC2 機群錯誤

EC2 機群狀態變更

EC2 機群會在 EC2 機群變更狀態時將 `EC2 Fleet State Change` 事件傳送至 Amazon EventBridge。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "715ed6b3-b8fc-27fe-fad6-528c7b8bf8a2",
    "detail-type": "EC2 Fleet State Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:20Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
        "sub-type": "active"
    }
}
```

`sub-type` 的可能值為：

`active`

EC2 機群請求已驗證且 Amazon EC2 正在嘗試維持執行中的執行個體目標數目。

`deleted`

EC2 機群 請求已刪除，且無執行中的執行個體。EC2 機群 會在其執行個體終止之後兩天刪除。

`deleted_running`

EC2 機群 請求已刪除，且不會啟動其他執行個體。其現有執行個體將繼續執行直到中斷或終止。此請求會一直維持在該狀態，直到所有執行個體中斷或終止。

`deleted_terminating`

EC2 機群請求已刪除，且其執行個體已終止。此請求會一直維持在該狀態，直到所有執行個體終止。

`expired`

EC2 機群請求已過期。如果請求是使用 `TerminateInstancesWithExpiration` 集建立的，後續 `terminated` 事件會指出已終止執行個體。

`modify_in_progress`

正在修改 EC2 機群請求。請求會一直維持在此狀態，直到已完成修改。

`modify_succeeded`

EC2 機群請求已修改。

`submitted`

正在評估 EC2 機群請求且 Amazon EC2 正準備啟動執行個體的目標數量。

`progress`

EC2 機群請求正在履行的過程中。

EC2 機群Spot 執行個體請求變更

EC2 機群會在機群中的 Spot 執行個體請求變更狀態時將 `EC2 Fleet Spot Instance Request Change` 事件傳送至 Amazon EventBridge。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "19331f74-bf4b-a3dd-0f1b-ddb1422032b9",
    "detail-type": "EC2 Fleet Spot Instance Request Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/
fleet-83fd4e48-552a-40ef-9532-82a3acca5f10"
    ],
    "detail": {
        "spot-instance-request-id": "sir-rmqske6h",
        "description": "SpotInstanceRequestId sir-rmqske6h, PreviousState:
 cancelled_running",
```

```
        "sub-type": "cancelled"
    }
}
```

sub-type 的可能值為：

active

Spot 執行個體請求已履行，並具有關聯的 Spot 執行個體。

cancelled

您已取消 Spot 執行個體請求，或 Spot 執行個體請求已過期。

disabled

您已停用 Spot 執行個體。

submitted

已提交 Spot 執行個體請求。

EC2 機群執行個體變更

EC2 機群會在機群中的執行個體變更狀態時將 EC2 Fleet Instance Change 事件傳送至 Amazon EventBridge。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "542ce428-c8f1-0608-c015-e8ed6522c5bc",
    "detail-type": "EC2 Fleet Instance Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:23Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
        "instance-id": "i-0c594155dd5ff1829",
```

```
        "description": "{\"instanceType\":\"c5.large\",\"image\":\"ami-6057e21a\",
\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1d\"}",
        "sub-type": "launched"
    }
}
```

sub-type 的可能值為：

launched

已啟動新執行個體。

terminated

已終止執行個體。

termination_notified

Amazon EC2 在縮減規模期間終止 Spot 執行個體，也就是機群的目標容量向下修改 (例如，從目標容量 4 變為 3) 時，系統會傳送執行個體終止通知。

EC2 機群資訊

EC2 機群會在履行期間發生錯誤時傳送 EC2 Fleet Information 事件至 Amazon EventBridge。此資訊事件不會封鎖機群嘗試履行其目標容量。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
    "detail-type": "EC2 Fleet Information",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T08:17:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-
bb9e-415d-8f54-3fa5a8628b91"
    ],
    "detail": {
        "description": "c4.xlarge, ami-0947d2ba12ee1ff75, Linux/UNIX, us-east-1a,
 Spot price in either SpotFleetRequestConfigData or SpotFleetLaunchSpecification or
 LaunchTemplate or LaunchTemplateOverrides is less than Spot market price $0.0619",
```

```
        "sub-type": "launchSpecUnusable"
    }
}
```

sub-type 的可能值為：

fleetProgressHalted

每個啟動規格中的價格無效，因為價格低於 Spot 價格 (所有啟動規格都已產生 launchSpecUnusable 事件)。如果 Spot 價格變動，啟動規格則可能會變成有效。

launchSpecTemporarilyBlacklisted

組態無效，且多次嘗試啟動執行個體失敗。如需詳細資訊，請參閱事件的描述。

launchSpecUnusable

啟動規格中的價格無效，因為價格低於 Spot 價格。

registerWithLoadBalancersFailed

嘗試在負載平衡器失敗時註冊執行個體。如需詳細資訊，請參閱事件的描述。

EC2 機群錯誤

EC2 機群會在履行期間發生錯誤時傳送 EC2 Fleet Error 事件至 Amazon EventBridge。錯誤事件會阻止機群嘗試履行其目標容量。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "69849a22-6d0f-d4ce-602b-b47c1c98240e",
    "detail-type": "EC2 Fleet Error",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-10-07T01:44:24Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-9bb19bc6-60d3-4fd2-ae47-
d33e68eafa08"
    ],
    "detail": {
```

```
        "description": "m3.large, ami-00068cd7555f543d5, Linux/UNIX: IPv6 is not
 supported for the instance type 'm3.large'. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

sub-type 的可能值為：

iamFleetRoleInvalid

EC2 機群沒有啟動或終止執行個體所需的許可。

allLaunchSpecsTemporarilyBlacklisted

沒有任何組態是有效的，且多次嘗試啟動執行個體失敗。如需詳細資訊，請參閱事件的描述。

spotInstanceCountLimitExceeded

您已達到可啟動 Spot 執行個體的數目上限。

spotFleetRequestConfigurationInvalid

組態無效。如需詳細資訊，請參閱事件的描述。

Spot 機群事件類型

> **Note**
>
> Spot Fleet 使用沒有計劃投資的舊版 API。我們建議您改用 EC2 Fleet 或 Auto Scaling 群組。如需詳細資訊，請參閱最好的機群方法是什麼？。

有五種 Spot 機群事件類型。每個事件類型都有數個子類型。

Event types (事件類型)

- EC2 Spot 機群狀態變更
- EC2 Spot 機群Spot 執行個體請求變更
- EC2 Spot 機群執行個體變更
- EC2 Spot 機群資訊
- EC2 Spot 機群錯誤

EC2 Spot 機群狀態變更

Spot 機群會在 Spot 機群變更狀態時將 `EC2 Spot Fleet State Change` 事件傳送至 Amazon EventBridge 。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "d1af1091-6cc3-2e24-203a-3b870e455d5b",
    "detail-type": "EC2 Spot Fleet State Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:57:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-4b6d274d-0cea-4b2c-
b3be-9dc627ad1f55"
    ],
    "detail": {
        "sub-type": "submitted"
    }
}
```

`sub-type` 的可能值為：

`active`

已驗證 Spot 機群請求，且 Amazon EC2 正在嘗試維持執行中的執行個體目標數目。

`cancelled`

Spot 機群請求已取消，且無執行中的執行個體。Spot 機群將會在其執行個體終止後兩天刪除。

`cancelled_running`

Spot 機群請求已取消，不會啟動其他執行個體。其現有執行個體將繼續執行直到中斷或終止。此請求會一直維持在該狀態，直到所有執行個體中斷或終止。

`cancelled_terminating`

Spot 機群請求已取消，其執行個體正在終止。此請求會一直維持在該狀態，直到所有執行個體終止。

`expired`

Spot 機群請求已過期。如果請求是使用 `TerminateInstancesWithExpiration` 集建立的，後續 `terminated` 事件會指出已終止執行個體。

`modify_in_progress`

正在修改 Spot 機群請求。請求會一直維持在此狀態，直到已完成修改。

`modify_succeeded`

已修改 Spot 機群請求。

`submitted`

正在評估 Spot 機群請求且 Amazon EC2 正準備啟動執行個體的目標數量。

`progress`

正在履行 Spot 機群請求。

EC2 Spot 機群Spot 執行個體請求變更

Spot 機群會在機群中的 Spot 執行個體請求變更狀態時將 `EC2 Spot Fleet Spot Instance Request Change` 事件傳送至 Amazon EventBridge。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "cd141ef0-14af-d670-a71d-fe46e9971bd2",
    "detail-type": "EC2 Spot Fleet Spot Instance Request Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:53:21Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-
a98d2133-941a-47dc-8b03-0f94c6852ad1"
    ],
    "detail": {
        "spot-instance-request-id": "sir-a2w9gc5h",
        "description": "SpotInstanceRequestId sir-a2w9gc5h, PreviousState:
 cancelled_running",
        "sub-type": "cancelled"
    }
```

```
}
```

sub-type 的可能值為：

active

Spot 執行個體請求已履行，並具有關聯的 Spot 執行個體。

cancelled

您已取消 Spot 執行個體請求，或 Spot 執行個體請求已過期。

disabled

您已停用 Spot 執行個體。

submitted

已提交 Spot 執行個體請求。

EC2 Spot 機群執行個體變更

Spot 機群會在機群中的執行個體變更狀態時將 EC2 Spot Fleet Instance Change 事件傳送至 Amazon EventBridge 。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "11591686-5bd7-bbaa-eb40-d46529c2710f",
    "detail-type": "EC2 Spot Fleet Instance Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T07:25:02Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-c8a764a4-bedc-4b62-
af9c-0095e6e3ba61"
    ],
    "detail": {
        "instance-id": "i-08b90df1e09c30c9b",
        "description": "{\"instanceType\":\"r4.2xlarge\",\"image\":
\"ami-032930428bf1abbff\",\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":
\"us-east-1a\"}",
        "sub-type": "launched"
```

```
    }
}
```

sub-type 的可能值為：

launched

已啟動新執行個體。

terminated

已終止執行個體。

termination_notified

Amazon EC2 在縮減規模期間終止 Spot 執行個體，也就是機群的目標容量向下修改 (例如，從目標容量 4 變為 3) 時，系統會傳送執行個體終止通知。

EC2 Spot 機群資訊

當履行期間發生錯誤時，Spot 機群將 EC2 Spot Fleet Information 事件傳送至 Amazon EventBridge。此資訊事件不會封鎖機群嘗試履行其目標容量。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "73a60f70-3409-a66c-635c-7f66c5f5b669",
    "detail-type": "EC2 Spot Fleet Information",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-08T20:56:12Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-2531ea06-
af18-4647-8757-7d69c94971b1"
    ],
    "detail": {
        "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot
 bid price is less than Spot market price $0.5291",
        "sub-type": "launchSpecUnusable"
    }
}
```

sub-type 的可能值為：

`fleetProgressHalted`

每個啟動規格中的價格無效，因為價格低於 Spot 價格 (所有啟動規格都已產生 `launchSpecUnusable` 事件)。如果 Spot 價格變動，啟動規格則可能會變成有效。

`launchSpecTemporarilyBlacklisted`

組態無效，且多次嘗試啟動執行個體失敗。如需詳細資訊，請參閱事件的描述。

`launchSpecUnusable`

啟動規格中的價格無效，因為價格低於 Spot 價格。

`registerWithLoadBalancersFailed`

嘗試在負載平衡器失敗時註冊執行個體。如需詳細資訊，請參閱事件的描述。

EC2 Spot 機群錯誤

當履行期間發生錯誤時，Spot 機群將 `EC2 Spot Fleet Error` 事件傳送至 Amazon EventBridge。錯誤事件會阻止機群嘗試履行其目標容量。

以下是此事件的範例資料。

```
{
    "version": "0",
    "id": "10adc4e7-675c-643e-125c-5bfa1b1ba5d2",
    "detail-type": "EC2 Spot Fleet Error",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T06:56:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/
sfr-38725d30-25f1-4f30-83ce-2907c56dba17"
    ],
    "detail": {
        "description": "r4.2xlarge, ami-032930428bf1abbff, Linux/UNIX: The
 associatePublicIPAddress parameter can only be specified for the network interface
 with DeviceIndex 0. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
```

```
}
```

sub-type 的可能值為：

`iamFleetRoleInvalid`

Spot 機群沒有啟動或終止執行個體所需的許可。

`allLaunchSpecsTemporarilyBlacklisted`

沒有任何組態是有效的，且多次嘗試啟動執行個體失敗。如需詳細資訊，請參閱事件的描述。

`spotInstanceCountLimitExceeded`

您已達到可啟動 Spot 執行個體的數目上限。

`spotFleetRequestConfigurationInvalid`

組態無效。如需詳細資訊，請參閱事件的描述。

EC2 機群的教學課程

有不同的方法可以設定 EC2 機群。您選擇的組態取決於您的特定使用案例。

下列教學課程涵蓋一些可能的使用案例，並提供實作這些案例所需的任務。

使用案例	教學課程連結
使用執行個體權重來管理 EC2 機群的可用性和效能。 執行個體加權，可讓您為 EC2 機群中的每個執行個體類型指派權重，以代表其運算容量和效能。根據權重，機群可以使用指定執行個體類型的任意組合，只要機群可以滿足所需的目標容量即可。	教學課程：使用執行個體權重設定 EC2 機群
使用隨需容量以確保尖峰時段的可用性，但可受益於以較低的成本獲得額外的 Spot 容量。 將 EC2 機群設定為使用隨需執行個體做為主要容量，以確保尖峰時段的可用容量。此外，將一	教學課程：將 EC2 機群設定為使用隨需執行個體做為主要容量

使用案例	教學課程連結
些容量配置給 Spot 執行個體，以受益於折扣定價，同時請記住，如果 Amazon EC2 需要恢復容量，Spot 執行個體可能會中斷。	
使用容量保留，為您的隨需執行個體保留運算容量。 將 EC2 機群設定為在啟動隨需執行個體時，先使用 `targeted` 容量保留。如果您有嚴格的容量需求，並且正在執行需要特定層級長期或短期容量保證的業務關鍵工作負載，建議您建立容量保留，以確保只要在需要時，即可有 Amazon EC2 容量存取權。	教學課程：使用目標容量保留設定 EC2 機群啟動隨需執行個體
使用容量區塊為您的 ML 工作負載保留需求高的 GPU 執行個體。 設定您的 EC2 機群以啟動執行個體至容量區塊。	教學課程：設定您的 EC2 機群以啟動執行個體至容量區塊
使用可中斷的容量保留來暫時重新利用整個 AWS 組織的閒置容量並節省成本。 將 EC2 機群設定為在可中斷的容量保留中啟動執行個體，其代表 AWS 組織內的備用容量。容量擁有者可以隨時回收容量。回收後，EC2 會在 2 分鐘通知後終止執行個體。	教學課程：設定您的 EC2 機群，在可中斷的容量保留中啟動執行個體
使用單一容量保留資源群組，在多個容量保留類型中啟動執行個體。 將 EC2 機群設定為使用單一容量保留資源群組，跨多個容量保留類型啟動執行個體：隨需容量保留、ML 的容量區塊和中斷容量保留。您可以依特定順序排定容量保留類型的優先順序，並在預留容量不足以滿足您的目標時，選擇性地回復為隨需容量。	教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體

教學課程：使用執行個體權重設定 EC2 機群

本教學課程使用一間稱為 Example Corp 的虛構公司，說明使用執行個體權重來請求 EC2 Fleet 的程序。

目標

Example Corp 是一間製藥公司，希望利用 Amazon EC2 的計算能力篩選可能用於抗癌的化合物。

規劃

Example Corp 首先檢視了 Spot 最佳實務。接下來，Example Corp 決定他們的 EC2 Fleet 需求。

執行個體類型

Example Corp 擁有運算和記憶體密集型應用程式，該應用程式在至少 60 GB 記憶體和八個虛擬 CPU (vCPU) 情況下效能最佳。他們希望以最低的價格為應用程式最大限度利用這些資源。Example Corp 認為以下任一個 EC2 執行個體類型都能滿足他們的需求：

執行個體類型	記憶體 (GiB)	vCPU
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

目標容量 (單位)

透過執行個體權重，目標容量可以等於多個執行個體 (預設) 或多個因素如核心 (vCPU)、記憶體 (GiB) 和儲存空間 (GB) 的組合。Example Corp 將其應用程式基礎 (60 GB RAM 和 8 個 vCPU) 做為一個單位考慮，Example Corp 決定此數量的 20 倍將可滿足他們的需求。因此，該公司將其 EC2 機群請求的目標容量設定為 20 個單位。

執行個體權重

Example Corp 決定目標容量後，即計算執行個體權重。為了計算每個執行個體類型的執行個體權重，將決定達成目標容量所需的每個執行個體類型單位，如下所示：

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 單位 (共 20 單位)
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 單位 (共 20 單位)
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 單位 (共 20 單位)

因此，Example Corp 會針對其 EC2 Fleet 請求中的各個啟動組態，指派 1、2 和 4 的執行個體權重。

每個單位小時的價格

Example Corp 會使用每執行個體小時的隨需價格，來做為其價格的起點。他們也可以使用最近的 Spot 價格，或兩者的組合。為計算每單位小時價格，他們將每個執行個體小時的起始價格除以權重。例如：

執行個體類型	隨需價格	執行個體權重	每個單位小時的價格
r3.2xLarge	0.7 USD	1	0.7 USD
r3.4xLarge	1.4 USD	2	0.7 USD
r3.8xLarge	$2.8	4	0.7 USD

Example Corp 可以使用每單位小時 0.7 USD 的全球價格，並且對於所有三種執行個體類型都具有競爭力。在 `r3.8xlarge` 啟動規格中，他們也可以使用每單位小時 0.7 USD 全局價格和 0.9 USD 每單位小時特定價格。

驗證許可

Example Corp 在建立 EC2 Fleet 之前，驗證其具有所需許可的 IAM 角色。如需詳細資訊，請參閱EC2 Fleet 先決條件。

建立啟動範本

接下來，範例公司會建立一個啟動範本。啟動範本 ID 會在下列步驟中使用。如需詳細資訊，請參閱建立 Amazon EC2 啟動範本。

建立 EC2 Fleet

Example Corp 會建立一個檔案 `config.json`，且其 EC2 Fleet 組態如下。在下列範例中，以您自己的資源識別符取代資源識別符。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r3.2xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "r3.4xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 2
                },
                {
                    "InstanceType": "r3.8xlarge",
                    "MaxPrice": "0.90",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 4
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example Corp 使用以下 create-fleet 命令來建立 EC2 Fleet。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

如需詳細資訊，請參閱 建立 EC2 Fleet。

履行

分配策略決定您的 競價型執行個體 來自哪個 Spot 容量集區。

採用 lowest-price 策略 (預設的策略)，競價型執行個體 來自實現時每單位價格最低的集區。為提供 20 個單位的容量，EC2 Fleet 將啟動 20 個 r3.2xlarge 執行個體 (20 除以 1)、10 個 r3.4xlarge 執行個體 (20 除以 2) 或 5 個 r3.8xlarge 執行個體 (20 除以 4)。

如果 Example Corp 使用 diversified 策略，競價型執行個體 將來自所有三個集區。EC2 Fleet 將啟動 6 個 r3.2xlarge 執行個體 (提供 6 個單位)、3 個 r3.4xlarge 執行個體 (提供 6 個單位) 和 2 個 r3.8xlarge 執行個體 (提供 8 個單位)，共計 20 個單位。

教學課程：將 EC2 機群設定為使用隨需執行個體做為主要容量

本教學課程使用一間稱為 ABC Online 的虛構公司來說明請求 EC2 Fleet 的隨需做為主要容量的程序，以及 Spot 容量 (如果可用)。

目標

餐飲外送公司 ABC Online 致力於跨 EC2 執行個體類型佈建 Amazon EC2 容量和購買選項，以實現該公司所要的規模、效能和成本。

計畫

ABC Online 需要固定容量以應付尖峰期，但希望以更低的價格獲得額外容量。該公司決定下列的 EC2 機群需求：

- 隨需執行個體容量 – ABC Online 需要 15 個隨需執行個體來確保其可以在尖峰期間容納流量。
- Spot 執行個體容量 – ABC Online 打算佈建 5 個 Spot 執行個體，以更低的價格提高效能。

驗證許可

ABC Online 在建立 EC2 Fleet 之前，驗證其具有所需許可的 IAM 角色。如需詳細資訊，請參閱EC2 Fleet 先決條件。

建立啟動範本

接下來，ABC Online 會建立啟動範本。啟動範本 ID 會在下列步驟中使用。如需詳細資訊，請參閱建立 Amazon EC2 啟動範本。

建立 EC2 Fleet

ABC Online 會建立一個檔案 config.json，且其 EC2 Fleet 組態如下。在下列範例中，以您自己的資源識別符取代資源識別符。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "2"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity":15,
        "DefaultTargetCapacityType": "spot"
    }
}
```

ABC Online 使用以下 create-fleet 命令來建立 EC2 Fleet。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

如需詳細資訊，請參閱 建立 EC2 Fleet。

履行

分配策略決定隨需容量能始終實現，而如果有可用容量，則目標容量的剩餘部分將做為 Spot 實現。

教學課程：使用目標容量保留設定 EC2 機群啟動隨需執行個體

本教學課程將逐步引導您完成必須執行的所有步驟，以便您的 EC2 Fleet 啟動隨需執行個體至 `targeted` 容量預留。

您將會了解到如何設定機群，以在啟動隨需執行個體時先使用 `targeted` 隨需容量預留。您也會了解到如何設定機群，以便機群在隨需目標容量總計超過可用的未使用容量預留數目時，使用指定的分配策略來選取要在其中啟動剩餘目標容量的執行個體集區。

EC2 Fleet 組態

本教學課程機群設定如下所示：

- 目標容量：10 個隨需執行個體

- 總計未使用的 targeted 容量保留：6 (小於機群的隨需目標容量：10 個隨需執行個體)
- 容量保留集區數目：2 (us-east-1a 和 us-east-1b)
- 每個集區的容量保留數目：3
- 隨需分配策略：lowest-price (當未使用的容量保留數目小於隨需目標容量時，機群會根據隨需分配策略決定要用來啟動剩餘隨需容量的集區。)

  請注意，您也可以使用 prioritized 分配策略，而不使用 lowest-price 分配策略。

若要按照 targeted 容量預留來啟動隨需執行個體，必須執行以下幾個步驟：

- 步驟 1：建立容量預留
- 步驟 2：建立容量預留資源群組
- 步驟 3：將容量預留新增至容量預留資源群組
- (選用) 步驟 4：檢視資源群組中的容量預留
- 步驟 5：建立啟動範本，此範本指定容量預留以特定資源群組為目標
- (選用) 步驟 6：描述啟動範本
- 步驟 7：建立 EC2 Fleet
- (選用) 步驟 8：檢視剩餘未使用的容量預留數目

步驟 1：建立容量預留

使用 create-capacity-reservation 命令來建立容量預留，其中三個用於 us-east-1a，另外三個用於 us-east-1b。除了可用區域之外，容量預留的其他屬性皆相同。

在 **us-east-1a** 中的 3 個容量預留

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1a \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

最後產生的容量預留 ID 範例

```
cr-1234567890abcdef1
```

在 **us-east-1b** 中的 3 個容量預留

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1b \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

最後產生的容量預留 ID 範例

```
cr-54321abcdef567890
```

步驟 2：建立容量預留資源群組

使用 `resource-groups` 服務和 create-group 命令來建立容量預留資源群組。在此範例中，資源群組會命名為 `my-cr-group`。如需為何必須建立資源群組的相關資訊，請參閱 使用容量保留功能來保留 EC2 機群中的隨需容量。

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

步驟 3：將容量預留新增至容量預留資源群組

使用 `resource-groups` 服務和 group-resources 命令，將您在步驟 1 中建立的容量預留新增至容量預留資源群組。請注意，您必須參考隨需容量預留的 ARN。

```
aws resource-groups group-resources \
    --group my-cr-group \
    --resource-arns \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

範例輸出

```
{
   "Failed": [],
```

```
    "Succeeded": [
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
    ]
}
```

(選用) 步驟 4：檢視資源群組中的容量預留

使用 `resource-groups` 服務和 list-group-resources 命令可選擇性地描述資源群組以檢視其容量預留。

```
aws resource-groups list-group-resources --group my-cr-group
```

範例輸出

```
{
    "ResourceIdentifiers": [
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1"
        },
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-54321abcdef567890"
        }
    ]
}
```

步驟 5：建立啟動範本，此範本指定容量預留以特定資源群組為目標

使用 create-launch-template 命令來建立啟動範本，使用此範本來指定要使用的容量預留。在此範例中，機群會使用新增至資源群組的 `targeted` 容量預留。因此，啟動範本資料會指定容量預留以特定資源群組為目標。在此範例中，啟動範本會命名為 `my-launch-template`。

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --launch-template-data \
        '{"ImageId": "ami-0123456789example",
          "CapacityReservationSpecification":
```

```
            {"CapacityReservationTarget":
                { "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-1:123456789012:group/my-cr-group" }
            }
        }'
```

(選用) 步驟 6：描述啟動範本

使用 describe-launch-template-versions 命令可選擇性地描述啟動範本，以檢視其組態。

```
aws ec2 describe-launch-template-versions --launch-template-name my-launch-template
```

範例輸出

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-01234567890example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2021-01-19T20:50:19.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0947d2ba12ee1ff75",
                "CapacityReservationSpecification": {
                    "CapacityReservationTarget": {
                        "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:us-east-1:123456789012:group/my-cr-group"
                    }
                }
            }
        }
    ]
}
```

步驟 7：建立 EC2 Fleet

建立 EC2 Fleet，該機群會指定將啟動之執行個體的組態資訊。下列 EC2 Fleet 組態僅顯示此範例的相關組態。啟動範本 `my-launch-template` 是您在步驟 5 中建立的啟動範本。有兩個執行個體集區，每個集區內的執行個體類型相同 (`c5.xlarge`)，但使用不同的可用區域 (`us-east-1a` 和 `us-`

east-1b)。執行個體集區的價格相同，因為定價是根據區域界定，而不是根據可用區域界定的。總目標容量為 10，預設目標容量類型為 on-demand。隨需分配策略為 lowest-price。容量預留的使用策略為 use-capacity-reservations-first。

> **Note**
>
> 機群類型必須為 instant。其他機群類型不支援 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
    },
    "Type": "instant"
}
```

使用上述組態建立 instant 機群之後，會啟動以下 10 個執行個體以符合目標容量：

- 會先使用容量保留來啟動 6 個隨需執行個體，如下所示：
  - 3 個隨需執行個體已在 us-east-1a 中啟動至 3 個 c5.xlarge targeted 容量保留
  - 3 個隨需執行個體已在 c5.xlarge 中啟動至 3 個 targeted us-east-1b 容量保留
- 為了滿足目標容量，會根據隨需分配策略 (在本例中為 lowest-price) 使用一般隨需容量來啟動 4 個額外的隨需執行個體。不過，由於集區的價格相同 (因為價格是按區域而非可用區域計價)，因此機群會使用任一個集區來啟動剩餘的 4 個隨需執行個體。

(選用) 步驟 8：檢視剩餘未使用的容量預留數目

啟動機群後，您也可以執行 describe-capacity-reservations，以查看還剩餘多少個未使用的容量預留。在此範例中，您應該會看到下列回應，其中顯示所有集區中的所有容量保留都已使用。

```
{ "CapacityReservationId":  "cr-111",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}

 { "CapacityReservationId":  "cr-222",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}
```

教學課程：設定您的 EC2 機群以啟動執行個體至容量區塊

本教學課程將逐步引導您完成必須執行的所有步驟，以便 EC2 Fleet 在容量區塊中啟動您的執行個體。

在大多數情況下，EC2 機群請求的目標容量應小於或等於您所鎖定容量區塊的保留可用容量。超出容量區塊預留上限的目標容量請求將無法滿足。如果目標容量請求超出容量區塊保留上限，您會收到 Insufficient Capacity Exception，告知您請求超出容量區塊的保留上限。

> **Note**
>
> 對於容量區塊，EC2 Fleet 將不會撤回至針對所需的目標剩餘容量啟動隨需執行個體。

如果 EC2 Fleet 無法在可用的預留容量區塊中滿足所要求的目標容量，EC2 Fleet 會盡可能滿足容量需求，並傳回能夠啟動的執行個體。您可以再次重複呼叫 EC2 Fleet，直到佈建所有執行個體為止。

設定 EC2 Fleet 請求後，您必須等待至容量區塊預留的開始日期。如果您請求 EC2 機群啟動至尚未啟動的容量區塊，則會收到 `Insufficient Capacity Error`。

預留的容量區塊啟用後，您就可以執行 EC2 Fleet 的 API 呼叫，並根據您選取的參數將執行個體佈建到容量區塊中。在容量區塊中執行的執行個體會繼續執行，直到您手動停止或終止這些執行個體，或 Amazon EC2 在容量區塊保留結束時終止這些執行個體。

如需關於容量區塊的詳細資訊，請參閱 ML 的容量區塊。

考量事項

- 在容量區塊中啟動執行個體時，僅支援 `instant` 類型的 EC2 機群請求。如需詳細資訊，請參閱設定 instant 類型的 EC2 機群。
- 同一 EC2 機群請求不支援多個容量區塊。
- 不支援在使用 `OnDemandTargetCapacity` 或 `SpotTargetCapacity` 時，同時將 `capacity-block` 設為 `DefaultTargetCapacity`。
- 如果將 `DefaultTargetCapacityType` 設為 `capacity-block`，則無法提供 `OnDemandOptions::CapacityReservationOptions`。一旦提供，便會發生例外狀況。

設定 EC2 機群以啟動執行個體至容量區塊

1. 建立啟動範本。

   在啟動範本中，執行下列動作：

   - 針對 `InstanceMarketOptionsRequest`，將 `MarketType` 設定為 `capacity-block`。
   - 若要鎖定容量區塊保留，請為 `CapacityReservationID` 指定容量區塊保留 ID。

   請記下啟動範本名稱和版本。在下一個步驟中，您將用到這項資訊。

   如需建立啟動範本的詳細資訊，請參閱 建立 Amazon EC2 啟動範本。
2. 設定 EC2 機群。

   建立檔案 `config.json`，並為您的 EC2 機群採用以下組態。在下列範例中，以您自己的資源識別符取代資源識別符。

   如需有關設定 EC2 機群的詳細資訊，請參閱 建立 EC2 Fleet。

   ```
   {
   ```

```
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "CBR-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "p5.48xlarge",
                    "AvailabilityZone": "us-east-1a"
                },
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "capacity-block"
    },
    "Type": "instant"
}
```

3. 啟動機群。

   使用以下 create-fleet 命令。

   ```
   aws ec2 create-fleet --cli-input-json file://config.json
   ```

   如需詳細資訊，請參閱建立 EC2 Fleet。

教學課程：設定您的 EC2 機群，在可中斷的容量保留中啟動執行個體

本教學課程會逐步解說您必須執行的步驟，讓 EC2 機群在可中斷的容量保留中啟動執行個體。

可中斷容量預留代表 AWS 組織內隨需容量預留擁有者提供給您的備用容量。這些保留適用於可中斷的工作負載，因為容量擁有者可以隨時透過 2 分鐘的中斷通知回收容量，之後 EC2 會終止執行個體。如需可中斷容量保留的詳細資訊，請參閱 可中斷的容量保留。

EC2 機群請求的目標容量應小於或等於您要鎖定的可中斷容量保留的可用容量。如果目標容量請求超過可中斷容量保留的可用容量，EC2 機群會盡可能啟動執行個體，並在 API 回應中報告啟動的執行個體。

在可中斷容量保留中執行的執行個體會繼續執行，直到您手動停止或終止，或容量擁有者回收容量為止。當容量擁有者回收容量時，Amazon EC2 會在終止執行個體前 2 分鐘傳送 Amazon EventBridge 通知。

考量事項

- 只有 類型的 EC2 機群請求`instant`支援在可中斷容量保留中啟動執行個體。
- `DefaultTargetCapacityType` 不支援使用 `OnDemandTargetCapacity`或 `SpotTargetCapacity`，同時將 `reserved-capacity` 設定為 。
- 當您指定多個啟動範本時，每個範本都以不同的可中斷容量保留為目標，EC2 機群會跨所有相符的保留佈建執行個體。
- 對於可中斷的容量保留，EC2 機群不會回復為在所需目標容量的剩餘時間內啟動隨需執行個體或 Spot 執行個體。

驗證許可

在建立 EC2 機群之前，請確認您擁有具有必要許可的 IAM 角色。如需詳細資訊，請參閱EC2 Fleet 先決條件。

若要在可中斷容量保留中啟動執行個體，您必須執行下列步驟：

步驟 1：建立啟動範本

使用 create-launch-template 命令建立啟動範本，以指定要鎖定的可中斷容量保留。在啟動範本中，將 `MarketType` 設定為 ，`interruptible-capacity-reservation`並指定可中斷容量保留`CapacityReservationId`的 。

啟動範本組態範例：

```
{
    "LaunchTemplateName": "interruptible-cr-launch-template",
    "LaunchTemplateData": {
        "InstanceType": "m5.large",
        "ImageId": "ami-0abcdef1234567890",
        "CapacityReservationSpecification": {
            "CapacityReservationTarget": {
                "CapacityReservationId": "cr-0123456789abcdef0"
            }
        },
```

```
        "InstanceMarketOptions": {
            "MarketType": "interruptible-capacity-reservation"
        }
    }
}
```

範例輸出

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-0123456789example",
        "LaunchTemplateName": "interruptible-cr-launch-template",
        "CreateTime": "2026-03-12T10:00:00.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

如需詳細資訊，請參閱建立 Amazon EC2 啟動範本。

步驟 2：設定 EC2 機群

為 EC2 機群建立指定啟動範本和目標容量的組態檔案。下列組態使用`interruptible-cr-launch-template`您在步驟 1 中建立的啟動範本。

使用 `reserved-capacity`做為 `interruptible-capacity-reservation`時，您必須指定 `ReservedCapacityOptions` ，並將 `ReservationType`設定為 `DefaultTargetCapacityType`。

建立名為 `config.json` 且具有下列內容的檔案：

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "interruptible-cr-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
```

```
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a"
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "reserved-capacity"
    },
    "ReservedCapacityOptions": {
        "ReservationType": ["interruptible-capacity-reservation"]
    },
    "Type": "instant"
}
```

金鑰組態參數：

參數	說明
DefaultTargetCapacityType	設定為 reserved-capacity ，表示執行個體應該以預留容量啟動。
ReservedCapacityOptions	指定預留容量的類型。對於可中斷的容量保留，請將 ReservationType 設定為 ["interruptible-capacity-reservation"] 。
Type	必須設定為 instant。可中斷容量保留僅支援即時機群。

步驟 3：啟動機群並檢視結果

使用 create-fleet 命令來建立機群：

```
aws ec2 create-fleet \
    --cli-input-json file://config.json
```

使用上述組態建立機instant群之後，EC2 機群會在可中斷容量保留中啟動 10 個執行個體，以符合目標容量。

> **Note**
>
> 如果機群無法滿足完整目標容量，回應會包含已啟動的執行個體，以及未滿足容量的任何錯誤。

範例輸出

```
{
    "FleetId": "fleet-12345678-1234-1234-1234-123456789012",
    "Instances": [
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a"
                }
            },
            "Lifecycle": "interruptible-capacity-reservation",
            "InstanceIds": [
                "i-0123456789example1",
                "i-0123456789example2",
                "i-0123456789example3",
                "i-0123456789example4",
                "i-0123456789example5",
                "i-0123456789example6",
                "i-0123456789example7",
                "i-0123456789example8",
                "i-0123456789example9",
                "i-0123456789example0"
            ],
            "InstanceType": "m5.large",
            "Platform": "Linux/UNIX"
        }
    ],
    "Errors": []
}
```

如需詳細資訊，請參閱建立 EC2 Fleet。

清除

若要停止產生費用，請在不再需要執行個體時將其終止。請注意，EC2 也會在容量擁有者回收容量時，自動終止在可中斷容量保留中啟動的執行個體。

相關資源

- 可中斷的容量保留
- 使用 EC2 隨需容量保留來保留運算容量
- 使用 EC2 機群
- 建立 EC2 Fleet
- 將執行個體啟動參數存放在 Amazon EC2 啟動範本中

教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體

本教學課程將逐步引導您設定 EC2 機群，以使用單一容量保留資源群組，跨多個容量保留類型啟動執行個體：隨需容量保留 (ODCRs)、ML 的容量區塊和可中斷容量保留。

您將了解如何建立包含不同容量保留類型的容量保留資源群組、設定 EC2 機群以特定順序排定容量保留類型的優先順序，以及當預留容量不足以滿足您的目標時，選擇性地回復到隨需容量。

考量事項

- 使用 時`instant`，僅支援 類型的 EC2 機群請求`ReservedCapacityOptions`。
- 您必須明確指定要在`ReservationTypes`清單中鎖定的容量保留類型。EC2 機群只會在您指定的容量保留類型中啟動執行個體。
- `ReservedCapacityOptions` 與 互斥`OnDemandOptions.CapacityReservationOptions`。您無法在相同的 EC2 機群請求中使用兩者。

本教學假設您的帳戶中已有下列容量保留：

- `p5.48xlarge` 中 的 ODCR (`cr-1234567890abcdef1`) `us-east-1a`
- `p4d.48xlarge` 中 的 ODCR (`cr-abcdef1234567890a`) `us-east-1b`
- `p5.48xlarge` 中 的容量區塊 (`cr-0123456789abcdef0`) `us-east-1a`

- p5.48xlarge 中 的可中斷容量保留 (cr-9876543210fedcba9)us-east-1a，由組織中的另一個帳戶與您共用

驗證許可

在建立 EC2 機群之前，請確認您擁有具有必要許可的 IAM 角色。如需詳細資訊，請參閱EC2 Fleet 先決條件。

步驟 1：建立容量保留資源群組

建立容量保留資源群組，以保留 EC2 機群鎖定的容量保留。設定 群組，使其接受任何執行個體類型的容量保留和容量保留類型，這可讓群組混合保留容量保留類型。如需step-by-step說明，請參閱容量保留群組。

> **Note**
>
> 當您建立 群組時，請勿在AWS::EC2::CapacityReservationPool組態中指定 instance-type或 reservation-type 參數。省略這些參數可讓 群組接受任何執行個體類型的容量保留和容量保留類型。

步驟 2：將容量保留新增至群組

將您的容量保留新增至容量保留資源群組。您可以新增 ODCRs、ML 的容量區塊、可中斷的容量保留和 UltraServer 容量區塊的任意組合。在本教學課程中，新增本教學課程開頭列出的四個容量保留。如需step-by-step說明，請參閱容量保留群組。

步驟 3：建立啟動範本

使用 create-launch-template 命令來建立以容量保留資源群組為目標的啟動範本。當您ReservedCapacityOptions在 EC2 機群組態中使用 時，您不需要在啟動範本MarketType中設定 - EC2 機群會為每個容量保留類型自動設定此值。

```
aws ec2 create-launch-template \
    --launch-template-name my-cr-resource-group-template \
    --launch-template-data \
        '{"ImageId": "ami-0123456789example",
          "CapacityReservationSpecification":
            {"CapacityReservationTarget":
```

```
                {"CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-1:123456789012:group/my-cr-resource-group"}
            }
        }'
```

步驟 4：建立 EC2 機群

建立 EC2 機群，以容量保留資源群組中的多個容量保留類型為目標。金鑰組態為 ReservedCapacityOptions，指定下列項目：

- ReservationTypes – 要鎖定的容量保留類型的排序清單。EC2 機群可依此順序處理類型。有效值為 on-demand-capacity-reservation、capacity-block 和 interruptible-capacity-reservation。
- AllocationStrategy – 執行個體類型在每個容量保留類型中的優先順序。設定為 prioritized 以遵循執行個體類型覆寫順序。
- ReservedCapacityFallbackOptions – MarketTypes設定為 ，["on-demand"]以在預留容量不足時啟動隨需執行個體。如果您未指定備用選項，EC2 機群不會在指定的容量保留類型用盡之後回復為任何其他市場類型。

建立名為 config.json 且具有下列內容的檔案。在下列範例中，以您自己的資源識別符取代資源識別符。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-cr-resource-group-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "p5.48xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "Priority": 1
                },
                {
                    "InstanceType": "p4d.48xlarge",
                    "AvailabilityZone": "us-east-1b",
                    "Priority": 2
                }
```

```
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 100,
        "DefaultTargetCapacityType": "reserved-capacity"
    },
    "ReservedCapacityOptions": {
        "ReservationTypes": ["on-demand-capacity-reservation", "capacity-block",
 "interruptible-capacity-reservation"],
        "ReservedCapacityFallbackOptions": {
            "MarketTypes": ["on-demand"]
        },
        "AllocationStrategy": "prioritized"
    },
    "Type": "instant"
}
```

使用 create-fleet 命令啟動機群：

```
aws ec2 create-fleet --cli-input-json file://config.json
```

如需詳細資訊，請參閱建立 EC2 Fleet。

EC2 機群如何解決容量保留

根據上述組態，EC2 機群會嘗試以下列順序啟動執行個體：

1. ODCRs – 先p5.48xlarge保留（最高優先順序的執行個體類型），然後再p4d.48xlarge保留。在每個執行個體類型中，如果存在多個 ODCRs，則會隨機選取一個。
2. 容量區塊 – 如果所有 ODCRs 中沒有可用的容量，EC2 機群會鎖定容量區塊。 會先p5.48xlarge保留，然後在每個執行個體類型內隨機p4d.48xlarge選取 。
3. 可中斷的容量保留 – 如果所有容量區塊中沒有可用的容量，EC2 機群會鎖定可中斷的容量保留。 會先p5.48xlarge保留，然後在每個執行個體類型內隨機p4d.48xlarge選取 。
4. 隨需備用 – 如果預留容量不足以達到 100 個執行個體的目標，則剩餘容量會以隨需執行個體的形式啟動。EC2 機群會根據您在 中指定的配置策略啟動這些隨需執行個體OnDemandOptions。

> **Note**
>
> EC2 機群會先解析容量保留類型，然後解析每個容量保留類型中的執行個體類型。在 EC2 機群移至排序清單中的下一個類型之前，每個容量保留類型都已完全耗盡。

（選用）步驟 5：驗證執行個體屬性

使用 describe-instances 命令來驗證每個執行個體使用的容量保留類型。機群啟動的執行個體會顯示下列屬性。

屬性	隨需容量保留值	容量區塊值	可中斷的容量保留值	隨需值
instance-lifecycle	null	capacity-block	interruptible-capacity-reservation	null
capacity-reservation-id	使用的容量保留 ID	使用的容量保留 ID	使用的容量保留 ID	不存在
capacity-reservation-specification	群組 ARN	群組 ARN	群組 ARN	不存在

您可以使用下列命令，依容量保留資源群組篩選執行個體：

```
aws ec2 describe-instances \
    --filters "Name=capacity-reservation-specification.capacity-reservation-target.capacity-reservation-resource-group-arn,Values=arn:aws:resource-groups:us-east-1:123456789012:group/my-cr-resource-group"
```

清除

若要停止產生費用，請在不再需要執行個體時終止執行個體。請注意，在容量區塊中啟動的執行個體會在容量區塊保留結束時自動終止。如果容量擁有者回收容量，則啟動至可中斷容量預留的執行個體會自動終止。

相關資源

- 容量保留群組
- ML 的容量區塊
- 可中斷的容量保留
- 使用 EC2 隨需容量保留來保留運算容量
- 使用 EC2 機群
- 建立 EC2 Fleet
- 將執行個體啟動參數存放在 Amazon EC2 啟動範本中

EC2 機群的範例 CLI 組態

您可以在 JSON 檔案中定義 EC2 機群組態，然後以 create-fleet 指令引用該檔案，以建立您的機群，如下所示：

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

下列範例說明各種 EC2 機群使用案例的啟動組態。如需組態參數的詳細資訊，請參閱 create-fleet。

範例

- 範例 1：啟動 Spot 執行個體做為預設購買選項
- 範例 2：啟動 隨需執行個體 做為預設購買選項
- 範例 3：啟動 隨需執行個體 做為主要容量
- 範例 4：使用多個容量保留來啟動隨需執行個體
- 範例 5：當總目標容量超過未使用的容量保留數目時，使用容量保留來啟動隨需執行個體
- 範例 6：使用目標容量保留啟動隨需執行個體
- 範例 7：設定容量重新平衡，以啟動取代 Spot 執行個體
- 範例 8：在容量最佳化機群中啟動 Spot 執行個體
- 範例 9：使用優先順序在容量最佳化機群中啟動 Spot 執行個體
- 範例 10：在 price-capacity-optimized 機群中啟動 Spot 執行個體
- 範例 11：設定屬性型執行個體類型選取

- 範例 12：使用容量保留資源群組在多個容量保留類型中啟動執行個體

如需類型 instant 機群的更多 CLI 範例，請參閱 設定 instant 類型的 EC2 機群。

範例 1：啟動 Spot 執行個體做為預設購買選項

下列範例指定 EC2 機群中所需的最小參數：啟動範本、目標容量和預設購買選項。啟動範本由其啟動範本 ID 和版本編號識別。機群的目標容量為 2 個執行個體，預設購買選項為 spot，會讓機群啟動 2 個 Spot 執行個體。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

範例 2：啟動 隨需執行個體 做為預設購買選項

下列範例指定 EC2 機群中所需的最小參數：啟動範本、目標容量和預設購買選項。啟動範本由其啟動範本 ID 和版本編號識別。機群的目標容量為 2 個執行個體，預設購買選項為 on-demand，會讓機群啟動 2 個 隨需執行個體。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
```

```
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```

範例 3：啟動 隨需執行個體 做為主要容量

下列範例指定機群 2 個執行個體的總目標容量，以及 1 個隨需執行個體的目標容量。預設購買選項為 spot。機群依照指定啟動 1 個隨需執行個體，但需要再啟動一個執行個體才能實現總目標容量。其差值的購買選項計算式為 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType，這會讓機群啟動 1 個 Spot 執行個體。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

範例 4：使用多個容量保留來啟動隨需執行個體

您可以將容量保留的使用策略設為 use-capacity-reservations-first，以將機群設定為在啟動隨需執行個體時首先使用隨需容量保留。本例示範當容量保留比所需目標容量更多時，機群如何選取要使用的容量保留。

本例使用的機群組態如下：

- 目標容量：12 個隨需執行個體
- 總計未使用的容量保留：15 (大於機群的目標容量：12 個隨需執行個體)

- 容量保留集區數目：3 (m5.large、m4.xlarge 和 m4.2xlarge)
- 每個集區的容量保留數目：5
- 隨需分配策略：lowest-price (當多個執行個體集區中有多個未使用的容量保留時，機群會根據隨需分配策略決定要用來啟動隨需執行個體的集區。)

  請注意，您也可以使用 prioritized 分配策略，而不使用 lowest-price 分配策略。

Capacity Reservations

帳戶在 3 個不同的集區中有以下 15 個未使用的容量保留。每個集區中的容量保留數目由 AvailableInstanceCount 表示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

機群組態

下列機群組態僅顯示此範例的相關組態。總目標容量為 12，預設目標容量類型為 on-demand。隨需分配策略為 lowest-price。容量保留的使用策略為 use-capacity-reservations-first。

在此範例中， 隨需執行個體 價格如下：

- m5.large – 每小時 0.096 美元
- m4.xlarge – 每小時 0.20 美元
- m4.2xlarge – 每小時 0.40 美元

> **Note**
>
> 機群類型必須為 instant。其他機群類型不支援 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-abc1234567example",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]
```

```
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant"
}
```

使用上述組態建立 instant 叢集之後，會啟動以下 12 個執行個體以符合目標容量：

- 5 個 m5.large 隨需執行個體在 us-east-1a 中 (us-east-1a 中的 m5.large 價格最低)，並且有 5 個可用的未使用 m5.large容量保留。
- 5 個 m4.xlarge 隨需執行個體在 us-east-1a 中 – us-east-1a 中的 m4.xlarge 價格次低，並且有 5 個可用的未使用 m4.xlarge容量保留。
- 2 個 m4.2xlarge 隨需執行個體在 us-east-1a 中 – us-east-1a 中的 m4.2xlarge 價格第三低，並且有 5 個可用的未使用 m4.2xlarge容量保留，只需要其中 2 個就可滿足目標容量

啟動機群後，您可以執行 describe-capacity-reservations，以查看還剩餘多少個未使用的容量保留。在本例中，您會看到下面回應，這顯示所有的 m5.large 和 m4.xlarge 容量保留都已使用，只剩 3 個 m4.2xlarge容量保留未使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
```

```
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 3
}
```

範例 5：當總目標容量超過未使用的容量保留數目時，使用容量保留來啟動隨需執行個體

您可以將容量保留的使用策略設為 use-capacity-reservations-first，以將機群設定為在啟動隨需執行個體時首先使用隨需容量保留。本例示範當總計目標容量超過可用的未使用容量保留數目時，機群會如何選取用來啟動隨需執行個體的執行個體集區。

本例使用的機群組態如下：

- 目標容量：16 個隨需執行個體
- 總計未使用的容量保留：15 (小於機群的目標容量：16 個隨需執行個體)
- 容量保留集區數目：3 (m5.large、m4.xlarge 和 m4.2xlarge)
- 每個集區的容量保留數目：5
- 隨需分配策略：lowest-price (當未使用的容量保留數目小於隨需目標容量時，機群會根據隨需分配策略決定要用來啟動剩餘隨需容量的集區。)

  請注意，您也可以使用 prioritized 分配策略，而不使用 lowest-price 分配策略。

Capacity Reservations

帳戶在 3 個不同的集區中有以下 15 個未使用的容量保留。每個集區中的容量保留數目由 AvailableInstanceCount 表示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

```
{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

機群組態

下列機群組態僅顯示此範例的相關組態。總目標容量為 16，預設目標容量類型為 on-demand。隨需分配策略為 lowest-price。容量保留的使用策略為 use-capacity-reservations-first。

在此範例中， 隨需執行個體 價格如下：

- m5.large – 每小時 0.096 USD
- m4.xlarge – 每小時 0.20 USD
- m4.2xlarge – 每小時 0.40 USD

> **Note**
>
> 機群類型必須為 instant。其他機群類型不支援 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
```

```
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

使用上述組態建立 instant 叢集之後，會啟動以下 16 個執行個體以符合目標容量：

- 6 個 m5.large 隨需執行個體在 us-east-1a 中 – us-east-1a 中的 m5.large 價格最低，並且有 5 個可用的未使用 m5.large容量保留。會先使用容量保留來啟動 5 個隨需執行個體。在使用了其餘的 m4.xlarge 和 m4.2xlarge 容量保留之後，為了滿足目標容量，會根據隨需分配策略啟動額外的隨需執行個體，在本例中為 lowest-price。

- 5 個 m4.xlarge 隨需執行個體在 us-east-1a 中 – us-east-1a 中的 m4.xlarge 價格次低，並且有 5 個可用的未使用 m4.xlarge容量保留。
- 5 個 m4.2xlarge 隨需執行個體在 us-east-1a 中 – us-east-1a 中的 m4.2xlarge 價格第三低，並且有 5 個可用的未使用 m4.2xlarge容量保留。

啟動機群後，您可以執行 describe-capacity-reservations，以查看還剩餘多少個未使用的容量保留。在此範例中，您應該會看到下列回應，其中顯示所有集區中的所有容量保留都已使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```

範例 6：使用目標容量保留啟動隨需執行個體

您可以將容量保留的使用策略設為 use-capacity-reservations-first，以將機群設定為在啟動隨需執行個體 時首先使用 targeted 隨需容量保留。本例示範如何使用 targeted 容量保留來啟動隨需執行個體，其中容量保留的屬性皆相同，但其可用區域 (us-east-1a 和 us-east-1b) 不同。本例亦示範當總計目標容量超過可用的未使用容量保留數目時，機群會如何選取用來啟動隨需執行個體的執行個體集區。

本例使用的機群組態如下：

- 目標容量：10 個隨需執行個體
- 總計未使用的 targeted 容量保留：6 (小於機群的隨需目標容量：10 個隨需執行個體)
- 容量保留集區數目：2 (us-east-1a 和 us-east-1b)

- 每個集區的容量保留數目：3
- 隨需分配策略：lowest-price (當未使用的容量保留數目小於隨需目標容量時，機群會根據隨需分配策略決定要用來啟動剩餘隨需容量的集區。)

  請注意，您也可以使用 prioritized 分配策略，而不使用 lowest-price 分配策略。

如需完成此範例所須執行程序的逐步演練，請參閱 教學課程：使用目標容量保留設定 EC2 機群啟動隨需執行個體。

Capacity Reservations

帳戶在 2 個不同的集區中有以下 6 個未使用的容量保留。在本例中，集區的可用區域有所不同。每個集區中的容量保留數目由 AvailableInstanceCount 表示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1b",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

機群組態

下列機群組態僅顯示此範例的相關組態。總目標容量為 10，預設目標容量類型為 on-demand。隨需分配策略為 lowest-price。容量保留的使用策略為 use-capacity-reservations-first。

在本例中，us-east-1 中的 c5.xlarge 隨需執行個體價格為每小時 0.17 美元。

> **Note**
>
> 機群類型必須為 instant。其他機群類型不支援 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
    },
    "Type": "instant"
}
```

使用上述組態建立 instant 機群之後，會啟動以下 10 個執行個體以符合目標容量：

- 會先使用容量保留來啟動 6 個隨需執行個體，如下所示：
  - 3 個隨需執行個體已在 us-east-1a 中啟動至 3 個 c5.xlarge targeted 容量保留

  - 3 個隨需執行個體已在 c5.xlarge 中啟動至 3 個 targeted us-east-1b 容量保留
- 為了滿足目標容量，會根據隨需分配策略 (在本例中為 lowest-price) 使用一般隨需容量來啟動 4 個額外的隨需執行個體。不過，由於集區的價格相同 (因為價格是按區域而非可用區域計價)，因此機群會使用任一個集區來啟動剩餘的 4 個隨需執行個體。

啟動機群後，您可以執行 describe-capacity-reservations，以查看還剩餘多少個未使用的容量保留。在此範例中，您應該會看到下列回應，其中顯示所有集區中的所有容量保留都已使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}
```

範例 7：設定容量重新平衡，以啟動取代 Spot 執行個體

下列範例會設定 EC2 機群，以便 Amazon EC2 在機群中對 Spot 執行個體發出重新平衡建議時啟動替代 Spot 執行個體。若要為 ReplacementStrategy 設定 Spot 執行個體的自動替代，請指定 launch-before-terminate。若要設定從啟動新替代 Spot 執行個體到自動刪除舊 Spot 執行個體的時間延遲，請針對 termination-delay，指定一值 (以秒為單位)。如需詳細資訊，請參閱 組態選項。

> **Note**
>
> 建議您僅在可以預測執行個體關閉程序需要多長時間才能完成時，才使用 launch-before-terminate，以便只在這些程序完成後才會終止舊執行個體。所有執行個體在執行時，您需要支付所有執行個體的費用。

容量重新平衡策略的有效性取決於 EC2 機群請求中指定的 Spot 容量集區數目。我們建議您使用多樣化的執行個體類型和可用區域的集合來設定機群，若要 AllocationStrategy，請指定 capacity-

optimized。如需設定 EC2 機群容量重新平衡時應考量哪些項目的詳細資訊，請參閱 在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體。

```
{
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "LaunchTemplate",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c4.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c5.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       }
                ]
          }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "MaintenanceStrategies": {
```

```
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

範例 8：在容量最佳化機群中啟動 Spot 執行個體

下列範例示範如何使用可最佳化容量的 Spot 分配策略來設定 EC2 機群。若要最佳化容量，您必須將 `AllocationStrategy` 設為 `capacity-optimized`。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。目標容量為 50 個 Spot 執行個體。EC2 機群嘗試將 50 個 Spot 執行個體啟動到 Spot 容量集區中，且擁有啟動中執行個體數目之最佳容量。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "r4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2a"
                           },
                      },
                       {
                           "InstanceType": "m4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           },
                       },
                       {
                           "InstanceType": "c5.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
```

```
                        }
                    }
                ]
            }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"

    }
}
```

範例 9：使用優先順序在容量最佳化機群中啟動 Spot 執行個體

下列範例示範在盡最大努力使用優先順序時，如何使用可最佳化容量的 Spot 分配策略來設定 EC2 機群。

當您使用 capacity-optimized-prioritized 分配策略時，您可以使用 Priority 參數來指定 Spot 容量集區的優先順序，其中數字越小，優先順序越高。如果您對它們一視同仁，也可以為數個 Spot 容量集區設定相同的優先順序。如果您沒有設定集區的優先順序，集區將被視為最後一個優先順序。

若要排定 Spot 容量集區的優先順序，您必須將 AllocationStrategy 設為 capacity-optimized-prioritized。EC2 機群將首先針對容量進行最佳化，但會盡力實現優先順序 (例如，若實現優先順序不會顯著影響 EC2 機群佈建最佳容量的能力)。對於必須將中斷可能性降至最低的工作負載來說，這是一個很好的選擇，而且某些執行個體類型的偏好也很重要。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。每個集區都有優先順序，其中數字越小，優先順序越高。目標容量為 50 個 Spot 執行個體。EC2 機群嘗試以最高優先順序在 Spot 容量集區中啟動 50 個 Spot 執行個體，但首先針對容量進行最佳化。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
```

```
                        {
                            "InstanceType": "r4.2xlarge",
                            "Priority": 1,
                            "Placement": {
                                "AvailabilityZone": "us-west-2a"
                            },
                        },
                        {
                            "InstanceType": "m4.2xlarge",
                            "Priority": 2,
                            "Placement": {
                                "AvailabilityZone": "us-west-2b"
                            },
                        },
                        {
                            "InstanceType": "c5.2xlarge",
                            "Priority": 3,
                            "Placement": {
                                "AvailabilityZone": "us-west-2b"
                            }
                        }
                    ]
                }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"
}
```

範例 10：在 price-capacity-optimized 機群中啟動 Spot 執行個體

下列範例示範如何使用可最佳化容量且價格最低的 Spot 分配策略來設定 EC2 機群。若要在將價格納入考慮的同時最佳化容量，您必須將 Spot `AllocationStrategy` 設定為 `price-capacity-optimized`。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。目標容量為 50 個 Spot 執行個體。EC2 機群嘗試將 50 個 Spot 執行個體啟動到 Spot 容量集區中，且擁有啟動中執行個體數目之最佳容量，同時還會選擇價格最低的集區。

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
```

```
        "MinTargetCapacity": 2,
        "SingleInstanceType": true
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "r4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2a"
                           },
                      },
                       {
                           "InstanceType": "m4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           },
                       },
                       {
                           "InstanceType": "c5.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           }
                       }
                  ]
              }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 50,
        "OnDemandTargetCapacity":0,
        "SpotTargetCapacity":50,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

範例 11：設定屬性型執行個體類型選取

下列範例示範如何將 EC2 機群設定為使用屬性型執行個體類型選取來識別執行個體類型。若要指定必要的執行個體屬性，您可以在 InstanceRequirements 結構中指定屬性。

在下列範例中，指定了兩個執行個體屬性：

- VCpuCount – 至少指定 2 個 vCPU。由於未指定最大值，因此沒有最大限制。
- MemoryMiB – 至少指定 4 MiB 的記憶體。由於未指定最大值，因此沒有最大限制。

將會識別任何具有 2 個或更多 vCPU 和 4 MiB 或更多記憶體的執行個體類型。但是，當 EC2 機群佈建機群時，價格保護和分配策略可能會排除某些執行個體類型。

如需您可指定之所有可能屬性的清單和說明，請參閱 Amazon EC2 API 參考中的InstanceRequirements。

```
{
 "SpotOptions": {
  "AllocationStrategy": "price-capacity-optimized"
 },
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }],
 "TargetCapacitySpecification": {
  "TotalTargetCapacity": 20,
  "DefaultTargetCapacityType": "spot"
 },
 "Type": "instant"
```

```
}
```

範例 12：使用容量保留資源群組在多個容量保留類型中啟動執行個體

下列範例會使用單一容量保留資源群組，將 instant EC2 機群設定為跨多個容量保留類型啟動執行個體：隨需容量保留、ML 的容量區塊和可中斷的容量保留。您可以在 ReservedCapacityOptions結構中指定這些偏好設定，並將 DefaultTargetCapacityType設定為 reserved-capacity。

於下列範例中：

- ReservationTypes – 要鎖定的容量保留類型的排序清單。EC2 機群可依此順序處理類型：on-demand-capacity-reservation，然後是 capacity-block，然後是 interruptible-capacity-reservation。
- AllocationStrategy – 設定為 prioritized 以根據您的執行個體類型覆寫順序，排定每個容量保留類型內的執行個體類型優先順序。
- ReservedCapacityFallbackOptions – MarketTypes 設定為 ["on-demand"]，因此機群會針對無法從預留容量滿足的任何目標容量啟動隨需執行個體。如果您省略此選項，在指定的容量保留類型用盡之後，機群不會回到任何其他市場類型。

啟動範本以容量保留資源群組為目標。當您使用 時ReservedCapacityOptions，您不需要在啟動範本MarketType中設定 - EC2 機群會為每個容量保留類型自動設定。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-cr-resource-group-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "p5.48xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "Priority": 1
                },
                {
                    "InstanceType": "p4d.48xlarge",
                    "AvailabilityZone": "us-east-1b",
                    "Priority": 2
                }
```

```
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 100,
        "DefaultTargetCapacityType": "reserved-capacity"
    },
    "ReservedCapacityOptions": {
        "ReservationTypes": ["on-demand-capacity-reservation", "capacity-block",
 "interruptible-capacity-reservation"],
        "ReservedCapacityFallbackOptions": {
            "MarketTypes": ["on-demand"]
        },
        "AllocationStrategy": "prioritized"
    },
    "Type": "instant"
}
```

如需step-by-step教學課程，請參閱 教學課程：設定您的 EC2 機群，使用容量保留資源群組在多個容量保留類型中啟動執行個體。

範例 CLI 組態 Spot 機群

您可以在 JSON 檔案中定義 Spot 機群組態，然後使用 request-spot-fleet AWS CLI 命令參考該檔案來建立機群，如下所示：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://file_name.json
```

下列範例說明各種 Spot 機群使用案例的啟動組態。如需組態參數的詳細資訊，請參閱 request-spot-fleet。如需有關建立 Spot 機群的詳細資訊，請參閱 建立 Spot 機群。

> **Note**
>
> 對於 Spot 機群，您無法在啟動範本或啟動規格中指定網路介面 ID。確保省略啟動範本或啟動規格中的 `NetworkInterfaceID` 參數。

範例

- 範例 1：使用區域中最低價格的可用區域來啟動 Spot 執行個體

- 範例 2：使用清單中最低價格的可用區域來啟動 Spot 執行個體
- 範例 3：使用清單中最低價格的執行個體類型來啟動 Spot 執行個體
- 範例 4。覆寫請求的價格
- 範例 5：使用多樣化分配策略來啟動 Spot 機群
- 範例 6：使用執行個體權重來啟動 Spot 機群
- 範例 7：使用隨需容量來啟動 Spot 機群
- 範例 8：設定容量重新平衡，以啟動取代 Spot 執行個體
- 範例 9：在容量最佳化機群中啟動 Spot 執行個體
- 範例 10：使用優先順序在容量最佳化機群中啟動 Spot 執行個體
- 範例 11：在 priceCapacityOptimized 機群中啟動 Spot 執行個體
- 範例 12：設定屬性型執行個體類型選取

範例 1：使用區域中最低價格的可用區域來啟動 Spot 執行個體

下列的範例指定了單一啟動規格，其中未包含可用區域或子網路。Spot 機群會在具有預設子網路的最低價格可用區域中，啟動執行個體。您所支付的價格不會超過隨需價格。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

範例 2：使用清單中最低價格的可用區域來啟動 Spot 執行個體

下列的範例指定了兩種啟動規格，各包含不同的可用區域或子網路，但具有相同的執行個體類型和 AMI。

可用區域

Spot 機群會在您所指定最低價格可用區域的預設子網路中，啟動執行個體。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "Placement": {
              "AvailabilityZone": "us-west-2a, us-west-2b"
          },
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

子網路

您可以指定預設子網路或非預設子網路，而非預設子網路可以是來自預設 VPC 或非預設 VPC。Spot 服務會在最低價格可用區域的任何可用子網路中，啟動執行個體。

您不能在 Spot 機群請求中指定來自同一個可用區域的不同子網路。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
```

```
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

如果執行個體是在預設的 VPC 中啟動，則其預設會接收公有 IPv4 位址。如果執行個體是在非預設的 VPC 中啟動，則其預設不會接收公有 IPv4 位址。在預設規格中啟動網路介面，來指派公有 IPv4 位址給在非預設 VPC 中啟動的執行個體。指定網路介面時，必須加入使用網路介面的子網路 ID 和安全群組 ID。

```
  ...
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "InstanceType": "m3.medium",
          "NetworkInterfaces": [
              {
                  "DeviceIndex": 0,
                  "SubnetId": "subnet-1a2b3c4d",
                  "Groups": [ "sg-1a2b3c4d" ],
                  "AssociatePublicIpAddress": true
              }
          ],
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ...
```

範例 3：使用清單中最低價格的執行個體類型來啟動 Spot 執行個體

下列的範例指定了兩種啟動組態，各具有不同的執行個體類型，但包含相同的 AMI 和可用區域或子網路。Spot 機群會使用具有最低價格的指定執行個體類型，來啟動執行個體。

可用區域

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "c5.4xlarge",
          "Placement": {
            "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "r3.8xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      }
  ]
}
```

子網路

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
```

```
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "r3.8xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      }
  ]
}
```

範例 4。覆寫請求的價格

我們建議您使用預設的最高價格，也就是隨需價格。如果想要的話，您可以指定機群請求的最高價格，和個別啟動規格的最高價格。

下列的範例指定了機群請求的最高價格，和三種啟動規格中其中兩種的最高價格。對於未指定最高價格的任何啟動規格，則會使用機群請求的最高價格。Spot 機群會使用具有最低價格的執行個體類型，來啟動執行個體。

可用區域

```
{
  "SpotPrice": "1.00",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.2xlarge",
```

```
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.10"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.4xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.20"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

子網路

```
{
  "SpotPrice": "1.00",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "SpotPrice": "0.10"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.4xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "SpotPrice": "0.20"
      },
      {
```

```
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

範例 5：使用多樣化分配策略來啟動 Spot 機群

下列範例使用 diversified 分配策略。這些啟動規格各具有不同的執行個體類型，但包含相同的 AMI 和可用區域或子網路。Spot 機群會將 30 個執行個體分配給 3 種啟動規格，如此每種規格就有 10 個執行個體。如需詳細資訊，請參閱使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量。

可用區域

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      }
```

```
  ]
}
```

子網路

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

最佳實務為增加 EC2 容量在其中一個可用區域停機時可以履行 Spot 請求的機會。此最佳做法可以讓區域之間多元化。針對此案例，請將可供您使用的各可用區域納入啟動規格之中。不要每次都使用相同的子網路，而是使用三個唯一的子網路 (各個都映射至不同的區域)。

可用區域

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
```

```
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2a"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2c"
            }
        }
    ]
}
```

子網路

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-2a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
```

```
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-3a2b3c4d"
        }
    ]
}
```

範例 6：使用執行個體權重來啟動 Spot 機群

下列範例使用執行個體權重，這表示價格為每單位小時而非每執行個體小時的價格。每個啟動組態都列出了不同的執行個體類型和不同的權重。Spot 機群會使用具有最低每單位小時價格的執行個體類型。Spot 機群會將目標容量除以執行個體的權重，來計算出要啟動的 Spot 機群數量。如果結果不是整數，則 Spot 機群會將其捨入到下一個整數，以便您的機群大小不低於其目標容量。

如果 `r3.2xlarge` 的請求成功，Spot 會佈建 4 個這類執行個體。將 20 除以 6，會得出總共 3.33 個執行個體，接著再四捨五入為 4 個執行個體。

如果 `c3.xlarge` 的請求成功，Spot 會佈建 7 個這類執行個體。將 20 除以 3，會得出總共 6.66 個執行個體，接著再四捨五入為 7 個執行個體。

如需詳細資訊，請參閱使用執行個體權重來管理 EC2 機群或 Spot 機群的成本和效能。

可用區域

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
```

```
            "WeightedCapacity": 3
        }
    ]
}
```

子網路

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 3
      }
  ]
}
```

範例 7：使用隨需容量來啟動 Spot 機群

為確保永遠有執行個體容量，您可以在 Spot 機群請求中加入對隨需容量的請求。如果有容量，則隨需請求一律已履行。如有容量和可用性，則目標容量的餘額已履行做為 Spot 使用。

下列範例將所要的目標容量指定為 10，其中 5 個必須為隨需容量。未指定 Spot 容量；它暗含在目標容量減去隨需容量的餘額中。如果有可用的 Amazon EC2 容量和可用性，Amazon EC2 會啟動 5 個容量單位作為隨需容量，5 個容量單位 (10-5=5) 作為 Spot 容量。

```
{
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 10,
  "SpotPrice": null,
  "ValidFrom": "2018-04-04T15:58:13Z",
```

```
  "ValidUntil": "2019-04-04T15:58:13Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [],
  "Type": "maintain",
  "OnDemandTargetCapacity": 5,
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
        "Version": "2"
      },
      "Overrides": [
        {
          "InstanceType": "t2.medium",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-d0dc51fb"
        }
      ]
    }
  ]
}
```

範例 8：設定容量重新平衡，以啟動取代 Spot 執行個體

下列範例會設定 Spot 機群，以便 Amazon EC2 在機群中對 Spot 執行個體發出重新平衡建議時啟動替代 Spot 執行個體。若要為 `ReplacementStrategy` 設定 Spot 執行個體的自動替代，請指定 `launch-before-terminate`。若要設定從啟動新替代 Spot 執行個體到自動刪除舊 Spot 執行個體的時間延遲，請針對 `termination-delay`，指定一值 (以秒為單位)。如需詳細資訊，請參閱組態選項。

> **Note**
>
> 建議僅在您可以預測執行個體關閉程序將需要多長時間才能完成時，才使用 `launch-before-terminate`。這可確保只在關機程序完成之後，才會終止舊執行個體。所有執行個體在執行時，您需要支付所有執行個體的費用。

容量重新平衡策略的有效性取決於 Spot 機群請求中指定的 Spot 容量集區數量。我們建議您使用多樣化的執行個體類型和可用區域的集合來設定機群，若要 `AllocationStrategy`，請指定 `capacityOptimized`。如需當設定 Spot 機群進行容量重新平衡時應考量哪些項目的詳細資訊，請參閱 在 EC2 機群和 Spot 機群中使用容量重新平衡，來取代具有風險的 Spot 執行個體。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "LaunchTemplate",
                    "Version": "1"
                },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c4.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c5.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       }
                ]
          }
    ],
        "TargetCapacity": 5,
        "SpotMaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
```

```
    }
}
```

範例 9：在容量最佳化機群中啟動 Spot 執行個體

下列範例示範如何使用可最佳化容量的 Spot 分配策略來設定 Spot 機群。若要最佳化容量，您必須將 `AllocationStrategy` 設為 `capacityOptimized`。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。目標容量為 50 個 Spot 執行個體。Spot 機群嘗試將 50 個 Spot 執行個體啟動到 Spot 容量集區中，且擁有啟動中執行個體數目之最佳容量。

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                    "AvailabilityZone": "us-west-2a"
                },
                {
                    "InstanceType": "m4.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

範例 10：使用優先順序在容量最佳化機群中啟動 Spot 執行個體

下列範例示範在盡最大努力使用優先順序時，如何使用可最佳化容量的 Spot 分配策略來設定 Spot 機群。

當您使用 `capacityOptimizedPrioritized` 分配策略時，您可以使用 `Priority` 參數來指定 Spot 容量集區的優先順序，其中數字越小，優先順序越高。如果您對它們一視同仁，也可以為數個 Spot 容量集區設定相同的優先順序。如果您沒有設定集區的優先順序，集區將被視為最後一個優先順序。

若要排定 Spot 容量集區的優先順序，您必須將 `AllocationStrategy` 設為 `capacityOptimizedPrioritized`。Spot 機群會先針對容量進行最佳化，但會盡力遵守優先順序 (例如，如果遵守優先順序不會顯著影響 Spot 機群佈建最佳容量的能力)。對於必須將中斷可能性降至最低的工作負載來說，這是一個很好的選擇，而且某些執行個體類型的偏好也很重要。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。每個集區都有優先順序，其中數字越小，優先順序越高。目標容量為 50 個 Spot 執行個體。Spot 機群嘗試以最高優先順序在 Spot 容量集區中啟動 50 個 Spot 執行個體，但首先針對容量進行最佳化。

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimizedPrioritized"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                   "Priority": 1,
                    "AvailabilityZone": "us-west-2a"
                },
                {
                          "InstanceType": "m4.2xlarge",
                          "Priority": 2,
                          "AvailabilityZone": "us-west-2b"
                },
                {
```

```
                            "InstanceType": "c5.2xlarge",
                            "Priority": 3,
                            "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

範例 11：在 priceCapacityOptimized 機群中啟動 Spot 執行個體

下列範例示範如何使用可最佳化容量且價格最低的 Spot 分配策略來設定 Spot 機群。若要在將價格納入考慮的同時最佳化容量，您必須將 Spot `AllocationStrategy` 設定為 `priceCapacityOptimized`。

在下列範例中，三個啟動規格指定三個 Spot 容量集區。目標容量為 50 個 Spot 執行個體。Spot 機群嘗試將 50 個 Spot 執行個體啟動到 Spot 容量集區中，且擁有啟動中執行個體數目之最佳容量，同時還會選擇價格最低的集區。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "OnDemandAllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111111111111:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": [
                     {
                            "InstanceType": "r4.2xlarge",
                            "AvailabilityZone": "us-west-2a"
                      },
                      {
                            "InstanceType": "m4.2xlarge",
                            "AvailabilityZone": "us-west-2b"
                      },
                      {
```

```
                        "InstanceType": "c5.2xlarge",
                        "AvailabilityZone": "us-west-2b"
                    }
                ]
            }
        ],
        "TargetCapacity": 50,
        "Type": "request"
    }
}
```

範例 12：設定屬性型執行個體類型選取

下列範例示範如何將 Spot 機群設定為使用屬性型執行個體類型選取來識別執行個體類型。若要指定必要的執行個體屬性，您可以在 `InstanceRequirements` 結構中指定屬性。

在下列範例中，指定了兩個執行個體屬性：

- `VCpuCount` – 至少指定 2 個 vCPU。由於未指定最大值，因此沒有最大限制。
- `MemoryMiB` – 至少指定 4 MiB 的記憶體。由於未指定最大值，因此沒有最大限制。

將會識別任何具有 2 個或更多 vCPU 和 4 MiB 或更多記憶體的執行個體類型。但是，當 Spot 機群佈建機群時，價格保護和分配策略可能會排除某些執行個體類型。

如需您可指定之所有可能屬性的清單和說明，請參閱 Amazon EC2 API 參考中的InstanceRequirements。

```
{
 "AllocationStrategy": "priceCapacityOptimized",
 "TargetCapacity": 20,
 "Type": "request",
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
```

```
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }]
}
```

EC2 機群和 Spot 機群的配額

您的 AWS 帳戶 具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定的。

通常的 Amazon EC2 配額適用於由 EC2 機群或 Spot 機群啟動的執行個體，例如 Spot 執行個體限制和磁碟區限制。

此外，您的 AWS 帳戶 具有下列與 EC2 機群和 Spot 機群相關的配額：

配額說明	配額
每個區域屬於類型 `maintain` 和 `request`，以及 `active`、`deleted_running` 和 `cancelled_running` 狀態的 EC2 機群和 Spot 機群數目	1,000 [1] [2] [3]
`instant` 類型的 EC2 機群數量	無限制
EC2 機群和 Spot 機群類型 `maintain` 和 `request` 的 Spot 容量集區數目 (執行個體類型和子網路的唯一組合)	300 [1]
EC2 機群類型 `instant` 的 Spot 容量集區數目 (執行個體類型和子網路的唯一組合)	無限制
啟動規格中使用者資料的大小	16 KB [2]
每 EC2 機群或 Spot 機群的目標容量	10,000
使用 時，每個 EC2 機群請求的容量預留數量 `ReservedCapacityOptions`	5，000 2

配額說明	配額
區域中涵蓋所有 EC2 機群和 Spot 機群的目標容量	100,000 [1]
EC2 機群請求或 Spot 機群請求不能跨區域。	
EC2 機群請求或 Spot 機群請求不能橫跨同一個可用區域的不同子網路。	

[1] 這些配額同時適用於您的 EC2 機群和您的 Spot 機群。

[2] 這些是硬性配額。您不能對這些配額請求提高配額。

[3] 刪除 EC2 機群或取消 Spot 機群請求後，如果您指定該機群不得在刪除或取消請求時終止其 Spot 執行個體，則機群請求會進入 `deleted_running` (EC2 機群) 或 `cancelled_running` (Spot 機群) 狀態，而執行個體會繼續執行直至中斷，或直至您手動將其終止。如果終止執行個體，則機群請求會進入 `deleted_terminating` (EC2 機群) 或 `cancelled_terminating` (Spot 機群) 狀態，而不會計入此配額。如需詳細資訊，請參閱刪除 EC2 機群請求和機群中的執行個體及取消 (刪除) Spot 機群請求。

請求增加目標容量的配額

如果您需要的容量超過目標容量的預設配額，則可以請求增加配額。

請求增加目標容量的配額

1. 開啟 支援 中心建立案例表單。
2. 選擇提高服務限制。
3. 對於 Limit type (限制類型)，請選擇 EC2 Fleet (EC2 機群)。
4. 針對區域，選擇要請求提高配額 AWS 的區域。
5. 對於 Limit (限制)，請選擇 Target Fleet Capacity per Fleet (in units) (每個機群的目標機群容量 (單位))，或者 Target Fleet Capacity per Region (in units) (每個區域的目標機群容量 (單位))，具體取決於您要增加的配額。
6. 針對 New limit value (新的限值)，輸入新的配額值。
7. 若要請求增加另一個配額，請選擇 Add another request (新增另一個請求)，然後重複步驟 4–6。
8. 對於 Use case description (使用案例描述)，輸入您請求增加配額的原因。

9. 在 Contact options (聯絡選項)，請指定您偏好的聯絡語言和聯絡方式。
10. 選擇提交。

在 Amazon EC2 中聯網

Amazon VPC 可讓您在帳戶的專用虛擬網路中啟動 AWS 資源，例如 Amazon EC2 執行個體 AWS ，稱為虛擬私有雲端 (VPC)。啟動執行個體時，您可以從 VPC 中選取子網。執行個體設定為主要網路介面，這是邏輯虛擬網路卡。執行個體會從子網的 IPv4 地址接收主要私人 IP 地址，並將其指派給主要網路介面。

您可以控制執行個體是否從 Amazon 的公有 IP 地址集區接收公有 IP 地址。執行個體的公有 IP 地址只會與您的執行個體關聯，直到執行個體停止或終止為止。如果您需要持久性公有 IP 地址，您可以為 AWS 您的帳戶配置彈性 IP 地址，並將其與執行個體或網路界面建立關聯。在您釋放彈性 IP 地址之前，彈性 IP 地址仍會與 AWS 您的帳戶保持關聯，而且您可以視需要將其從一個執行個體移至另一個執行個體。您可以將自己的 IP 地址範圍帶到您的 AWS 帳戶中，它會顯示為地址集區，然後從您的地址集區配置彈性 IP 地址。

若要提高網路效能並減少延遲，您可以在配置群組中啟動執行個體。使用增強型聯網，您可以獲得大幅提高的每秒封包數 (PPS) 效能。您可以使用 Elastic Fabric Adapter (EFA)，即彈性架構轉接器來加速高效能運算和機器學習應用程式，這是您可以連接至支援的執行個體類型的網路裝置。

功能

- 區域 (Region) 和區域 (Zone)
- Amazon EC2 執行個體 IP 定址
- EC2 執行個體主機名稱與網域
- 將自有 IP 位址 (BYOIP) 用於 Amazon EC2
- 彈性 IP 位址
- 彈性網路介面
- Amazon EC2 執行個體網路頻寬
- 增強 Amazon EC2 執行個體的聯網能力
- 適用於 Amazon EC2 上的 AI/ML 和 HPC 工作負載的 Elastic Fabric Adapter
- Amazon EC2 拓撲
- Amazon EC2 執行個體的置放群組
- EC2 執行個體的網路最大傳輸單位 (MTU)
- EC2 執行個體的虛擬私有雲
- 次要網路

區域 (Region) 和區域 (Zone)

Amazon EC2 託管於全球的多個地點。這些位置由 AWS 區域、可用區域 AWS Outposts、本機區域和 Wavelength 區域組成。

- 區域是獨立的地理區域。
- 可用區域是每個區域內的多個獨立位置。
- Local Zones 可讓您將資源 (例如運算和儲存) 放置在靠近最終使用者的多個位置。
- Wavelength Zones 能夠讓您建置可為 5G 裝置與最終使用者提供極低延遲的應用程式。Wavelength 會將標準 AWS 運算和儲存服務部署到電信業者的 5G 網路邊緣。
- AWS Outposts 將原生 AWS 服務、基礎設施和操作模型帶入幾乎所有資料中心、主機代管空間或內部部署設施。

AWS 操作state-of-the-art、高可用性的資料中心。儘管故障極為少見，但仍可能影響相同位置內執行個體的可用性。若您將所有執行個體都託管於單一位置，一旦該位置受故障影響，所有執行個體都將無法使用。

如需詳細資訊，請參閱 AWS 全球基礎設施。

目錄

- 大區 (Regions)
- 可用區域
- 本機區域
- Wavelength 區域
- AWS Outposts

大區 (Regions)

每個區域皆設計為與其他區域隔離。如此可達到最高的容錯能力與穩定性。

啟動執行個體時，選取可讓執行個體接近特定客戶，或符合法律或您的其他要求的區域。可以在多個區域中啟動執行個體。

當檢視資源時，您只會看到與您指定的區域繫結的資源。這是因為區域彼此隔離，而且我們不會自動跨區域複寫資源。

可用的區域

如需可用區域的清單，請參閱 AWS 區域。

Amazon EC2 的區域端點

使用命令列介面或 API 動作操作執行個體時，您必須指定其區域端點。如需 Amazon EC2 的區域與端點的詳細資訊，請參閱「Amazon EC2 Developer 指南」中的 Amazon EC2 服務端點。

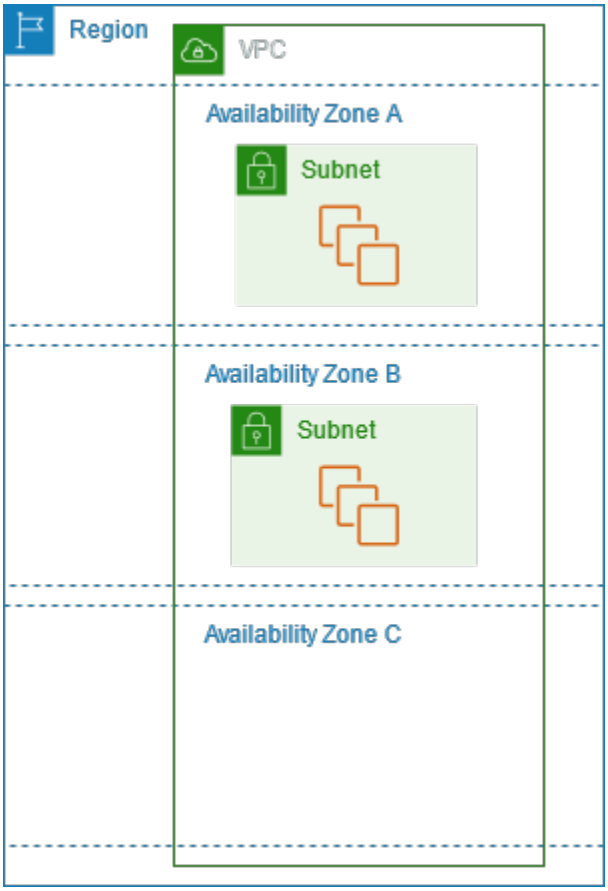
如需詳細資訊，請參閱《AWS 區域 和可用區域使用者指南》中的 AWS 區域。

可用區域

每個區域擁有多個隔離位置，稱為可用區域。可用區域的代碼為其區域代碼，後續跟著一個字母識別符。例如 `us-east-1a`。

透過在多個可用區域中啟動 EC2 執行個體，可保護應用程式免受區域中單個位置故障的影響。

下圖說明 區域中的多個可用區域 AWS 。可用區域 A 和可用區域 B 各有一個子網路，每個子網路都有 EC2 執行個體。可用區域 C 沒有子網，因此您無法在此可用區域中啟動執行個體。



如需詳細資訊，請參閱EC2 執行個體的虛擬私有雲。

按地區劃分的可用區域

如需按地區劃分的可用區域清單，請參閱 AWS 可用區域。

可用區域的執行個體

啟動執行個體時，可以選取區域和虛擬私有雲端 (VPC)。然後，可以從其中一個可用區域中選取子網路，或讓我們為您選擇子網路。啟動初始執行個體時，建議讓我們根據系統運作狀態和可用容量為您選取可用區域。如果您啟動其他的執行個體，則只有當您的新執行個體必須接近或隔離於現有執行個體時，您才能指定可用區域。

如果您要將執行個體分散至多個可用區域，但其中一個執行個體出現故障，則您可設計應用程式，讓其他可用區域內的執行個體來處理請求。

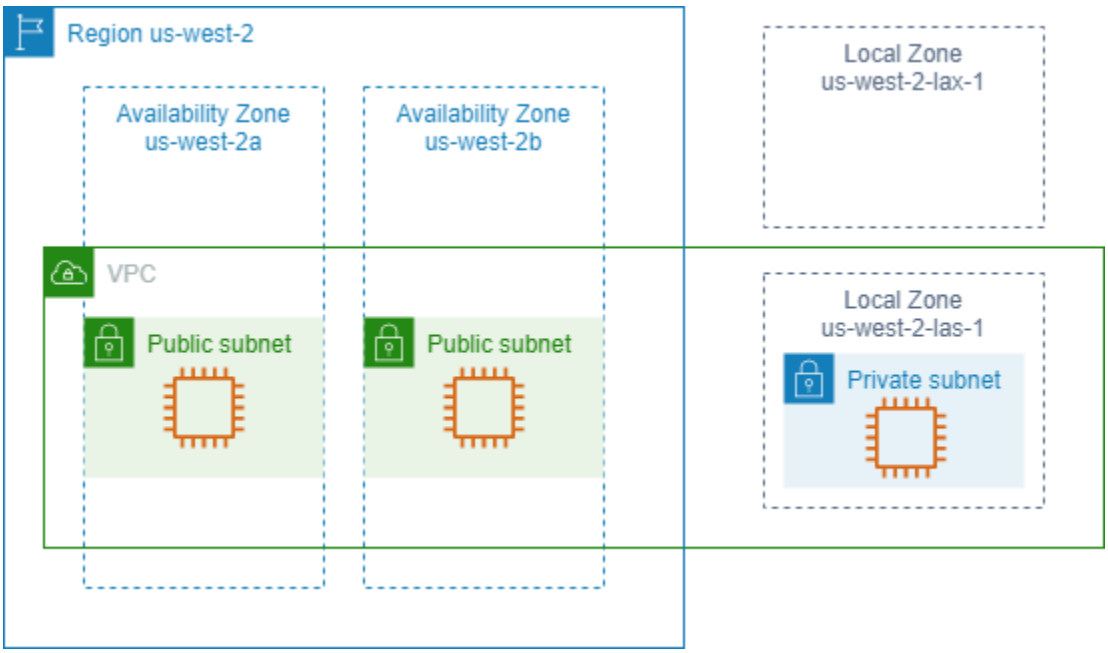
如需詳細資訊，請參閱《AWS 區域 和可用區域使用者指南》中的 AWS 可用區域。

本機區域

Local Zone 是 AWS 區域的延伸，其地理位置接近您的使用者。Local Zones 有自己的網際網路連線和支援 Direct Connect，因此在 Local Zone 中建立的資源可以為本機使用者提供低延遲的通訊。如需詳細資訊，請參閱什麼是 AWS Local Zones？ 在 AWS Local Zones 使用者指南中。

本機區域的代碼為其區域代碼，後續跟著一個字母識別符，代表其實體位置。例如：`us-west-2-lax-1` 位於洛杉磯。

下圖說明 AWS 區域 `us-west-2`、兩個可用區域，以及兩個本機區域。VPC 跨越可用區域和其中一個本機區域。VPC 中的每個區域都有一個子網，而每個子網都有一個執行個體。



可用 Local Zones

如需可用 Local Zones 的清單，請參閱AWS Local Zones 使用者指南中的可用 Local Zones。如需公告的 Local Zones 清單，請參閱 AWS Local Zones 位置。

Local Zones 中的執行個體

若要使用 Local Zone，首先您必須啟用它。在本機區域中建立子網路。您可以在啟動執行個體時指定 Local Zone 子網路，這會將其放置在 Local Zone 的 Local Zone 子網路中。

當您在 Local Zone 中啟動執行個體時，您也可以從網路邊界群組配置 IP 位址。網路邊界群組是一組唯一的可用區域、本機區域或 Wavelength 區域，用於 AWS 公告 IP 地址，例如 `us-west-2-lax-1a`。您可以從網路邊界群組配置下列 IP 地址：

- Amazon 提供的彈性 IPv4 地址
- Amazon 提供的 IPv6 VPC 地址 (僅適用於洛杉磯地區)

如需如何在 Local Zones 中啟動執行個體的詳細資訊，請參閱 AWS Local Zones 使用者指南中的 Local Zones 入門。 AWS

Wavelength 區域

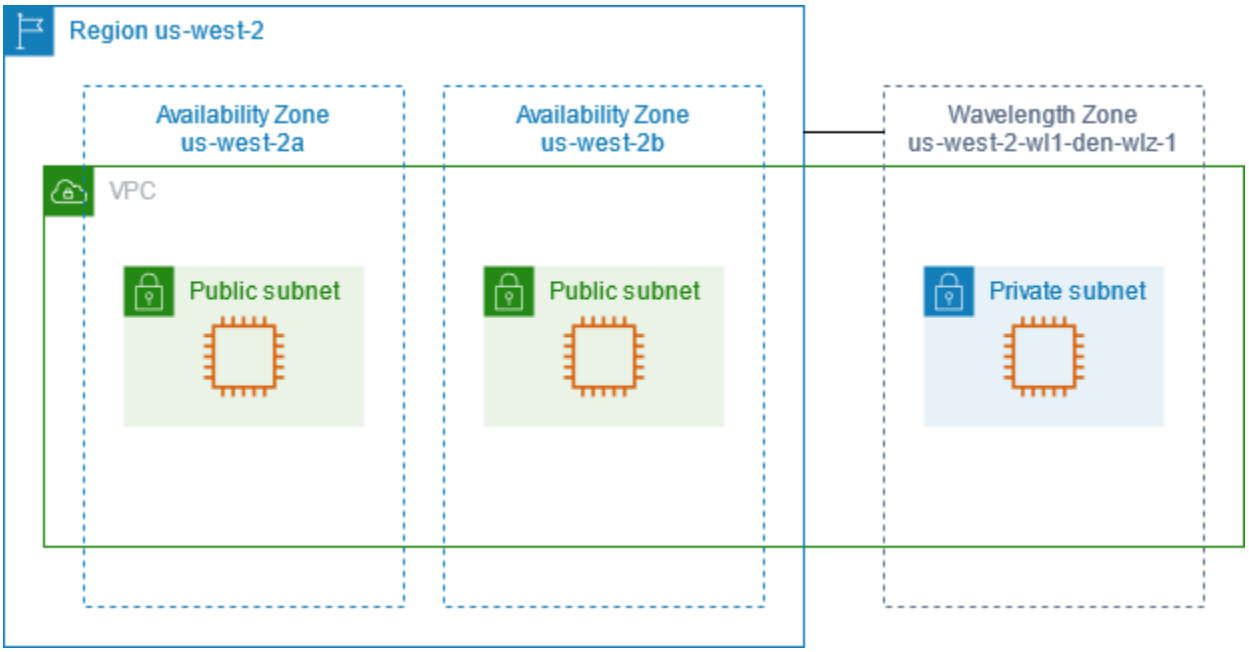
AWS Wavelength 可讓開發人員建置應用程式，為行動裝置和最終使用者提供極低延遲。Wavelength 會將標準 AWS 運算和儲存服務部署到電信業者的 5G 網路邊緣。開發人員可以將虛擬私有雲端 (VPC)

擴展到一或多個 Wavelength 區域，然後使用 AWS Amazon EC2 執行個體之類的資源來執行需要超低延遲和區域中 AWS 服務連線的應用程式。

Wavelength 區域是部署 Wavelength 基礎設施之電信業者位置的隔離區域。Wavelength 區域繫結至區域。Wavelength 區域是區域的邏輯延伸，並且由區域中的控制平面管理。

Wavelength 區域的代碼為其區域代碼，後續跟著一個字母識別符，代表其實體位置。例如：`us-east-1-wl1-bos-wlz-1` 位於波士頓。

下圖說明 AWS 區域 `us-west-2`、兩個可用區域和一個 Wavelength 區域。VPC 跨越可用區域和該 Wavelength 區域。VPC 中的每個區域都有一個子網，而每個子網都有一個執行個體。



並非每個區域都有 Wavelength 區域。如需支援 Wavelength 區域的區域相關資訊，請參閱 AWS Wavelength 開發人員指南中的可用 Wavelength 區域。

可用 Wavelength 區域

如需可用 Wavelength 區域的清單，請參閱「AWS Wavelength 開發人員指南」中的可用 Wavelength 區域。

Wavelength 區域中的執行個體

若要使用 Wavelength 區域，您必須先選擇加入「Zone (區域)」。在 Wavelength 區域中建立子網路。您可以在啟動執行個體時指定 Wavelength 子網路。您也可以從網路邊界群組配置電信業者 IP 地址，該網路邊界群組是一組唯一的可用區域、Local Zones 或 Wavelength 區域， AWS 會從中公告 IP 地址，例如，`us-east-1-wl1-bos-wlz-1`。

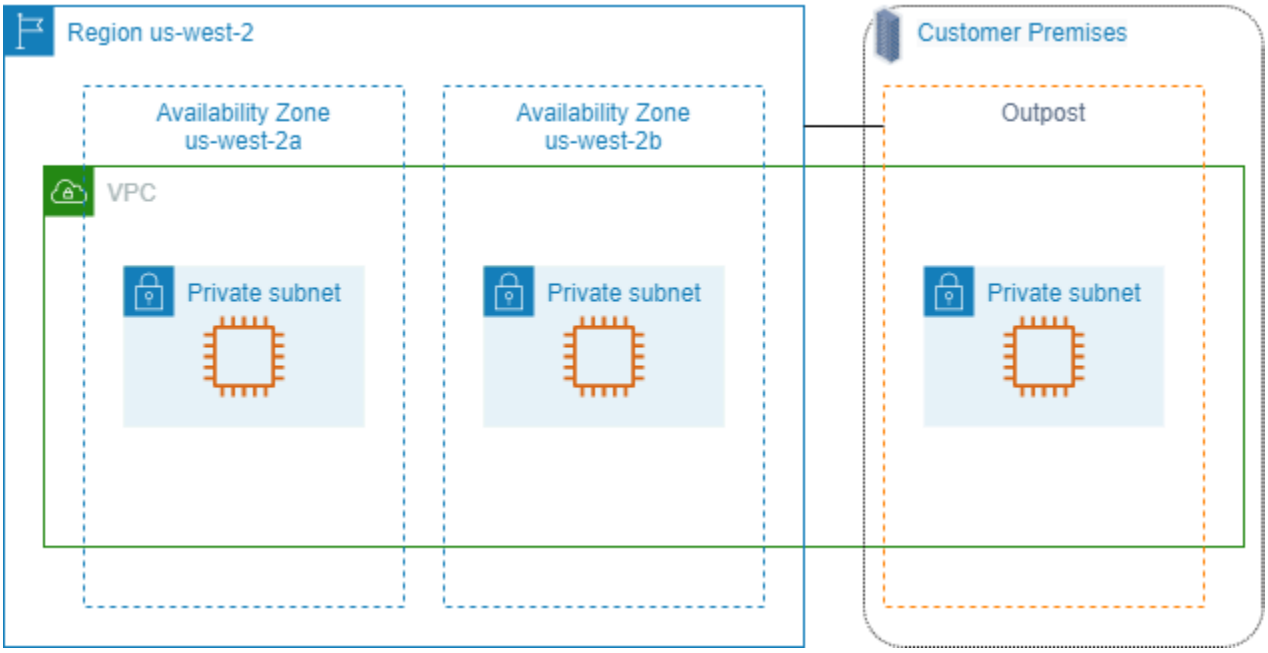
關於在 Wavelength 區域中啟動執行個體的逐步說明，請參閱「AWS Wavelength 開發人員指南」中的AWS Wavelength入門。

AWS Outposts

AWS Outposts 是一種全受管服務，可將 AWS 基礎設施、服務、APIs 和工具延伸到客戶內部部署。透過提供 AWS 受管基礎設施的本機存取權， AWS Outposts 可讓客戶使用與 AWS 區域中相同的程式設計界面，在內部部署中建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。

Outpost 是部署在客戶站點的 AWS 運算和儲存容量集區。 會在 AWS 區域中 AWS 操作、監控和管理此容量。您可以在 Outpost 上建立子網路，並在建立 AWS 資源時指定子網路。Outpost 子網路中的執行個體會使用私有 IP 地址與 AWS 區域中的其他執行個體通訊，全部都在相同的 VPC 內。

下圖說明 AWS 區域 `us-west-2`、兩個可用區域和一個 Outpost。VPC 跨越可用區域和 Outpost。Outpost 位於內部部署客戶資料中心。VPC 中的每個區域都有一個子網，而每個子網都有一個執行個體。



Outpost 上的執行個體

若要開始使用 AWS Outposts，您必須建立 Outpost 並訂購 Outpost 容量。 AWS Outposts 提供兩種外形規格：Outposts 機架和 Outposts 伺服器。如需 Outposts 組態的詳細資訊，請參閱AWS Outposts 系列。安裝 Outpost 設備之後，當您在 Outpost 上啟動 EC2 執行個體時，即可使用運算和儲存容量。

若要啟動 EC2 執行個體，您必須建立 Outpost 子網路。安全群組可控制 Outpost 子網路中執行個體的傳入與傳出流量，就像可用區域子網路中的執行個體一樣。若要使用 SSH 來連線到 Outpost 子網路中的執行個體，可在將其啟動時指定金鑰對，就像您對可用區域子網路中的執行個體所執行的操作一樣。

如需詳細資訊，請參閱開始使用 Outposts 機架或開始使用 Outposts 伺服器。

Outposts 機架上的磁碟區

如果您的容量位於 Outpost 機架上，您可以在您建立的 Outpost 子網路中建立 EBS 磁碟區。建立磁碟區時，請指定 Outpost 的 Amazon Resource Name (ARN)。

以下 create-volume 命令會在指定的 Outpost 上建立空的 50 GB 磁碟區。

```
aws ec2 create-volume --availability-zone us-east-2a --outpost-arn arn:aws:outposts:us-east-2:123456789012:outpost/op-03e6fecad652a6138 --size 50
```

您可以動態地修改 Amazon EBS gp2 磁碟區的大小，而不將其分離。如需有關在不分離磁碟區的情況下修改磁碟區的詳細資訊，請參閱「Amazon EBS 使用者指南」中的請求修改 EBS 磁碟區。

我們建議您將 Outpost 機架上執行個體的根磁碟區限制為 30 GiB 或更小。您可以在 AMI 或執行個體的區塊型設備映射中指定資料磁碟區，以提供額外的儲存空間。若要修剪開機磁碟區中未使用的區塊，請參閱 AWS 合作夥伴網路部落格中的如何建立稀疏 EBS 磁碟區。

我們建議您增加根磁碟區的 NVMe 逾時。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 I/O 操作逾時。

Outposts 伺服器上的磁碟區

Outpost 伺服器上的執行個體提供執行個體儲存體磁碟區，但不支援 EBS 磁碟區。選擇僅具有單一 EBS 快照的 Amazon EBS 支援的 AMI。選擇具有足夠執行個體儲存空間的執行個體尺寸，以滿足您的應用程式需求。如需詳細資訊，請參閱執行個體儲存體磁碟區限制。

Amazon EC2 執行個體 IP 定址

Amazon EC2 和 Amazon VPC 同時支援 IPv4 和 IPv6 定址通訊協定。Amazon VPC 預設會使用 IPv4 定址通訊協定；您無法停用此行為。當您建立 VPC 時，必須指定 IPv4 CIDR 區塊 (私有 IPv4 地址的範圍)。您可以選擇將 IPv6 CIDR 區塊指派給 VPC，並將 IPv6 地址從該區塊指派給子網中的執行個體。

當您啟動 EC2 執行個體時，可以指定 VPC 與子網路。執行個體會接收子網路 CIDR 範圍中的私有 IPv4 位址。可以選擇使用公有 IPv4 位址和 IPv6 位址來設定執行個體。如果不同 VPCs中的 EC2 執行個體使用公有 IP 地址通訊，流量會保留在 AWS 私有全域網路中，而不會周遊公有網際網路。

目錄

- 私有 IPv4 地址
- 公有 IPv4 地址
- 公有 IPv4 地址最佳化
- IPv6 地址
- 多個 IP 地址
- EC2 執行個體主機名稱
- Link-local 地址
- 管理 EC2 執行個體的 IPv4 地址
- 管理 EC2 執行個體的 IPv6 地址
- EC2 執行個體的次要 IP 位址
- 為 Windows 執行個體設定次要私有 IPv4 地址

私有 IPv4 地址

私有 IPv4 地址是無法透過網際網路存取的 IP 地址。您可以在相同 VPC 的執行個體之間使用私有 IPv4 地址進行通訊。如需私有 IPv4 地址標準和規格的詳細資訊，請參閱 RFC 1918。我們會使用 DHCP 將私有 IPv4 地址配置給執行個體。

> **Note**
>
> 您可以建立具有可公開路由 CIDR 區塊的 VPC，而此 CIDR 區塊不在 RFC 1918 所指定的私有 IPv4 地址範圍內。不過，基於本文件的目的，我們會將私有 IPv4 地址 (或「私有 IP 地址」) 參照為 VPC 之 IPv4 CIDR 範圍內的 IP 地址。

VPC 子網可以是下列其中一個類型：

- 僅限 IPv4 子網路：您只能在已獲指派 IPv4 地址的這些子網路中建立資源。
- 僅限 IPv6 子網路：您只能在已獲指派 IPv6 地址的這些子網路中建立資源。
- IPv4 和 IPv6 子網路：您可以在已獲指派 IPv4 和 IPv6 地址的這些子網路中建立資源。

當您將 EC2 執行個體啟動至僅限 IPv4 或雙堆疊 (IPv4 和 IPv6) 子網時，此執行個體會從子網的 IPv4 地址範圍接收主要私有 IP 地址。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 IP 定址。如果您未在啟動執行個體時指定主要私有 IP 地址，我們會在子網的 IPv4 範圍中選取可用 IP 地址。每個執行個體皆會有獲指派之主要私有 IPv4 地址的預設網路介面 (index 0)。您也可以指定其他私有 IPv4 地址 (稱為次要私有 IPv4 地址)。與主要私有 IP 地址不同，輔助私有 IP 地址可從原執行個體重新指派至別的執行個體。如需詳細資訊，請參閱多個 IP 地址。

私有 IPv4 地址，無論是主要或次要地址，在執行個體停止和啟動時或休眠和啟動時都會維持與網路介面的關聯，而會在執行個體終止時予以釋出。

公有 IPv4 地址

公有 IP 地址是可從網際網路存取的 IPv4 地址。您可以使用公有地址，在執行個體與網際網路之間通訊。

當您在預設 VPC 中啟動執行個體時，預設會將公有 IP 地址指派給它。當您將執行個體啟動到非預設 VPC 時，子網的屬性可判定啟動到該子網的執行個體是否接收公有 IPv4 地址集區中的公有 IP 地址。根據預設，我們不會將公有 IP 地址指派給非預設子網中啟動的執行個體。

您可以控制您的執行個體是否接收公有 IP 地址，如下所示：

- 修改子網路的公有 IP 尋址屬性。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的修改子網路的公有 IPv4 定址屬性。
- 在啟動期間啟用或停用公有 IP 尋址功能。這會覆寫子網路的公有 IP 尋址屬性。如需詳細資訊，請參閱在啟動時指派公有 IPv4 位址。
- 啟動後從執行個體中取消指派公有 IP 位址。如需詳細資訊，請參閱the section called “管理 IP 位址”。

公有 IP 地址會從 Amazon 的公有 IPv4 地址集區指派給您的執行個體，而且不會與 AWS 您的帳戶建立關聯。取消公有 IP 地址與您執行個體的關聯後，會將之釋出回公有 IPv4 地址集區，且您將無法重複使用之。

在下列情況下，我們會從執行個體中釋出公有 IP 位址，並指派新的公有 IP 位址：

- 當執行個體停止、休眠或終止時，我們會釋出公有 IP 位址。當您啟動已停止或休眠的執行個體時，我們會指派新的公有 IP 位址。
- 當您將彈性 IP 位址與執行個體建立關聯時，我們會釋出公有 IP 位址。當您取消彈性 IP 位址與執行個體的關聯時，我們會指派新的公有 IP 位址。

- 如果我們釋出執行個體的公有 IP 位址，且它具有次要網路介面，則我們不會指派新的公有 IP 位址。
- 如果我們釋出執行個體的公有 IP 位址，且它沒有與彈性 IP 位址相關聯的次要私有 IP 位址，則我們不會指派新的公有 IP 位址。

如果您需要可建立和取消與執行個體之關聯的持久性公有 IP 地址，請改用彈性 IP 地址。

如果您使用動態 DNS 將現有 DNS 名稱映射至新執行個體的公有 IP 地址，則最多可能需要 24 個小時，IP 地址才會傳播到整個網際網路。因此，已終止的執行個體還會持續收到請求，而新的執行個體可能尚未開始接收流量。若要解決此問題，請使用彈性 IP 地址。您可以配置自己的彈性 IP 地址，並建立地址與您執行個體的關聯。如需詳細資訊，請參閱彈性 IP 位址。

如果您使用 Amazon VPC IP Address Manager (IPAM)，您可以從 取得連續的公有 IPv4 地址區塊，AWS 並使用它將彈性 IP 地址配置給 AWS 資源。使用連續 IPv4 地址區塊可大幅降低安全存取控制清單的管理開銷，並簡化企業在 上擴展的 IP 位址配置和追蹤 AWS。如需詳細資訊，請參閱「Amazon VPC IPAM 使用者指南」中的從 IPAM 集區配置循序彈性 IP 位址。

考量事項

- AWS 收取所有公有 IPv4 地址的費用，包括與執行中執行個體相關聯的公有 IPv4 地址和彈性 IP 地址。如需詳細資訊，請參閱 Amazon VPC 定價頁面中的公有 IPv4 地址。
- 透過公有 NAT IP 地址存取其他執行個體的執行個體，會根據其是否位在相同的區域，而收取地區或網際網路資料傳輸的費用。

公有 IPv4 地址最佳化

AWS 收取所有公有 IPv4 地址的費用，包括與執行中執行個體相關聯的公有 IPv4 地址和彈性 IP 地址。如需詳細資訊，請參閱 Amazon VPC 定價頁面中的公有 IPv4 地址。

下列清單包含您可以採取的動作，以最佳化您使用的公有 IPv4 地址數量：

- 使用彈性負載平衡器將流量載入 EC2 執行個體，並在指派給執行個體的主要 ENI 上停用自動指派公有 IP。負載平衡器使用單一公有 IPv4 地址，因此可減少公有 IPv4 地址計數。您可能也想要合併現有的負載平衡器，以進一步減少公有 IPv4 地址計數。
- 如果使用 NAT 閘道的唯一原因是 SSH 進入私有子網路中的 EC2 執行個體以進行維護或緊急狀況，請考慮改用 EC2 Instance Connect Endpoint。EC2 Instance Connect Endpoint 允許您連線至執行個體，而不需要執行個體擁有公有 IPv4 地址。
- 如果您的 EC2 執行個體位於公有子網路中，並配置公有 IP 位址給它們，請考慮將執行個體移至私有子網路、移除公有 IP 位址，並使用公有 NAT 閘道允許存取和存取您的 EC2 執行個體。使用 NAT

閘道有成本考量。使用此計算方法來判斷 NAT 閘道是否具有成本效益。您可以建立 AWS 帳單成本和用量報告，以取得此計算Number of public IPv4 addresses所需的 。

```
NAT gateway per hour + NAT gateway public IPs + NAT gateway transfer / Existing
 public IP cost
```

其中：

- NAT gateway per hour = $0.045 * 730 hours in a month * Number of Availability Zones the NAT gateways are in
- NAT gateway public IPs = $0.005 * 730 hours in a month * Number of IPs associated with your NAT gateways
- NAT gateway transfer = $0.045 * Number of GBs that will go through the NAT gateway in a month
- Existing public IP cost = $0.005 * 730 hours in a month * Number of public IPv4 addresses

如果總計小於 1，NAT 閘道會比公有 IPv4 地址便宜。

- 使用 AWS PrivateLink私下連線至其他 AWS 帳戶託管的 AWS 服務或 服務，而不是使用公有 IPv4 地址和網際網路閘道。
- 將您自己的 IP 位址範圍 (BYOIP) 帶到 AWS ，並使用公有 IPv4 地址的範圍，而不是使用 Amazon 擁有的公有 IPv4 地址。
- 關閉為啟動到子網路中的執行個體會自動指派公用 IPv4 位址。建立子網路時，VPCs 通常會預設為停用此選項，但您應該檢查現有的子網路，以確保其已停用。
- 如果您有不需要公有 IPv4 地址的 EC2 執行個體，請檢查連接至執行個體的網路介面是否已停用自動指派公有 IP。
- 在 AWS Global Accelerator中為私有子網路中的 EC2 執行個體設定加速器端點，讓網際網路流量可以直接流向 VPCs中的端點，而不需要公有 IP 位址。您也可以將自己的地址帶到 AWS Global Accelerator,並使用自己的 IPv4 地址做為加速器的靜態 IP 位址。

IPv6 地址

IPv6 地址是全域不重複的地址，可設定為維持在私有不公開的狀態，或可透過網際網路來連線。公有和私有 IPv6 定址都可在 中使用 AWS：

- 私有 IPv6： AWS 考慮私有 IPv6 會處理未公告且無法在網際網路上公告的 IPv6 AWS。

- 公有 IPv6： AWS 考慮公有 IPv6 處理在網際網路上公告的 IPv6 AWS。

如需公有和私有 IPv6 地址的詳細資訊，請參閱「Amazon VPC 使用者指南」中的 IPv6 地址。

所有執行個體類型都支援 IPv6 地址，但下列項目除外：C1、M1、M2、M3 和 T1。

如果 IPv6 CIDR 區塊與 VPC 和子網相關聯，以及符合下列其中一項，則您的 EC2 執行個體會收到 IPv6 地址：

- 在啟動期間，您的子網會設定為自動將 IPv6 地址指派給執行個體。如需詳細資訊，請參閱修改子網路的公有 IP 定址屬性。
- 在啟動期間，您將 IPv6 地址指派給您的執行個體。
- 在啟動之後，您將 IPv6 地址指派給執行個體的主要網路介面。
- 在啟動之後，您將 IPv6 地址指派給相同子網中的網路介面，並將網路介面連接至執行個體。

當您的執行個體在啟動期間收到 IPv6 地址時，會建立該地址與執行個體之主要網路介面 (index 0) 的關聯。您可以透過下列方式，管理執行個體主要網路介面的 IPv6 地址：

- 從網路介面指派和取消指派 IPv6 位址 您可指派給網路介面的 IPv6 地址數目以及您可連接至執行個體的網路介面數目，會根據執行個體類型而不同。如需詳細資訊，請參閱每個網路介面的 IP 位址上限。
- 啟用主要 IPv6 位址。主要 IPv6 位址可讓您避免中斷傳送至執行個體或 ENI 的流量。如需詳細資訊，請參閱 為您的 EC2 執行個體建立網路介面 或 管理網路介面的 IP 位址： 。

IPv6 地址會在您停止和啟動或休眠和啟動執行個體時持續保留，並在您終止執行個體時予以釋放。若 IPv6 地址已經指派給另一個網路介面，您無法重新指派該 IPv6 地址 (您必須先將之取消指派)。

您可以透過控制子網路的路由或使用安全群組和網路 ACL 規則來控制執行個體是否可以透過其 IPv6 地址存取。如需更多資訊，請參閱「Amazon VPC 使用者指南」中的「網路間流量隱私權」。

如需預留 IPv6 地址範圍的詳細資訊，請參閱 IANA IPv6 Special-Purpose Address Registry 和 RFC4291。

多個 IP 地址

您可以指定執行個體的多個私有 IPv4 和 IPv6 地址。您可指定給執行個體的網路介面數目以及私有 IPv4 和 IPv6 地址數目取決於執行個體類型。如需詳細資訊，請參閱每個網路介面的 IP 位址上限。

使用案例

- 在單一伺服器上使用多個 SSL 憑證，以及建立每個憑證與特定 IP 地址的關聯，以在單一伺服器上託管多個網站。
- 操作網路應用裝置 (例如防火牆或負載平衡器)，其中的每個網路介面有多個 IP 地址。
- 將輔助 IP 地址重新指派給待命執行個體，以在執行個體失敗時將內部流量重新導向至備用執行個體。

多個 IP 地址運作方式

- 您可以將輔助私有 IPv4 地址指派給任何網路介面。
- 您可以將多個 IPv6 地址指派給網路介面，而網路介面位於具有相關聯 IPv6 CIDR 區塊的子網中。
- 您必須從網路介面之子網的 IPv4 CIDR 區塊範圍選擇輔助 IPv4 位址。
- 您必須從網路介面之子網的 IPv6 CIDR 區塊範圍選擇 IPv6 地址。
- 您可以將安全群組與網路介面 (而非個別的 IP 地址) 建立關聯。因此，您在網路介面中指定的每個 IP 地址會受其網路介面安全群組的規範。
- 多個 IP 地址可以指派和取消指派至連接至執行中或已停止執行個體的網路介面。
- 已指派給網路介面的輔助私有 IPv4 地址，可以經您的明確允許而重新指派給另一個網路介面。
- 無法將 IPv6 地址重新指派給另一個網路介面；您必須先從現有網路介面取消指派 IPv6 地址。
- 使用命令列工具或 API 將多個 IP 地址指派給網路介面時，如果無法指派其中一個 IP 地址，則整個操作會失敗。
- 輔助網路介面與執行個體分離或連接至執行個體時，網路介面會保有主要私有 IPv4 地址、輔助私有 IPv4 地址、彈性 IP 地址和 IPv6 地址。
- 雖然您無法從執行個體分離主要網路介面，但是您可以將主要網路介面的輔助私有 IPv4 地址重新指派給另一個網路介面。

如需詳細資訊，請參閱the section called “次要 IP 位址”。

EC2 執行個體主機名稱

當您建立 EC2 執行個體時， AWS 會建立該執行個體的主機名稱。如需主機名稱類型及其佈建方式的詳細資訊 AWS，請參閱 EC2 執行個體主機名稱與網域。Amazon 所提供的 DNS 伺服器可將 Amazon 提供的主機名稱解析為 IPv4 和 IPv6 地址。Amazon DNS 伺服器位於 VPC 網路範圍基礎加 2。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。

Link-local 地址

Amazon EC2 會為只能從 EC2 執行個體存取的服務保留特定的 IPv4 連結本機地址。這些服務會在基礎主機上執行，而不是在執行個體上執行。當您存取這些服務的預留連結本機地址時，您會與 Xen Hypervisor 或 Nitro 控制器通訊。基礎主機會處理這些預留地址的流量，而不是根據客戶設定的路由轉送。安全群組和網路 ACLs 不會篩選此流量。

AWS 不會保留 中的每個 IPv4 地址`169.254.0.0/16`。此範圍內非預留地址的流量遵循適用的 VPC 路由。您的安全群組和網路 ACLs可以篩選此流量。

Link-local 地址範圍

- IPv4 – 169.254.0.0/16 (169.254.0.0 至 169.254.255.255)
- IPv6 – fe80::/10

使用 link-local 地址存取的服務

- 執行個體中繼資料服務 – `169.254.169.254`
- Amazon Route 53 Resolver （也稱為 Amazon DNS 伺服器） – `169.254.169.253`
- Amazon Time Sync Service – `169.254.169.123`
- AWS KMS 伺服器

管理 EC2 執行個體的 IPv4 地址

您可以在啟動執行個體時，將公有 IPv4 地址指派給該執行個體。您可以在主控台中透過 Instances (執行個體) 頁面或 Network Interfaces (網路介面) 頁面檢視您執行個體的 IPv4 地址。

任務

- 在啟動時指派公有 IPv4 位址
- 在啟動時指派私有 IPv4 位址
- 檢視主要 IPv4 位址。
- 使用執行個體中繼資料來檢視 IPv4 位址

在啟動時指派公有 IPv4 位址

每一個子網都會有一個屬性，以判定是否將公有 IP 地址指派至啟動到該子網的執行個體。非預設子網預設會將此屬性設定為 false，而預設子網會將此屬性設定為 true。當您啟動執行個體時，也可以使用公有 IPv4 定址功能控制是否將公有 IPv4 地址指派至執行個體；您可以覆寫子網之 IP 定址屬性的預設行為。公有 IPv4 地址會從 Amazon 的公有 IPv4 地址集區指派，並會指派給具有裝置索引 eth0 的網路介面。此功能取決於啟動執行個體時的特定條件。

考量事項

- 您可以在啟動後透過管理與網路介面相關聯的 IP 位址，從執行個體取消指派公有 IP 位址。如需公有 IPv4 地址的詳細資訊，請參閱公有 IPv4 地址。
- 如果指定多個網路介面，則無法自動指派公有 IP 位址。此外，如果您為 eth0 指定現有網路介面，則無法使用自動指派公有 IP 功能覆寫子網路設定。
- 不過，不論您是否在啟動期間將公有 IP 位址指派給執行個體，在啟動執行個體之後，都可以建立彈性 IP 位址與執行個體的關聯。如需詳細資訊，請參閱彈性 IP 位址。您也可以修改子網的公有 IPv4 定址行為。如需詳細資訊，請參閱修改子網路的公有 IPv4 定址屬性。

Console

若要在啟動時指派公有 IPv4 位址

遵循程序啟動執行個體，當您設定 Network Settings (網路設定) 時，選擇選項以 Auto-assign Public IP (自動指派公有 IP)。

AWS CLI

若要在啟動時指派公有 IPv4 位址

請使用 `--associate-public-ip-address` 選項並搭配 run-instances 命令。

```
--associate-public-ip-address
```

PowerShell

若要在啟動時指派公有 IPv4 位址

使用 New-EC2Instance cmdlet 搭配 `-AssociatePublicIp` 參數。

```
-AssociatePublicIp $true
```

在啟動時指派私有 IPv4 位址

您可以指定子網路 IPv4 位址範圍中的 IPv4 位址，或讓 Amazon EC2 為您選擇一個。此地址會指派給主要網路介面。

若要在啟動之後指派 IPv4 位址，請參閱 the section called “將次要 IP 位址指派給執行個體”。

Console

若要在啟動時指派私有 IPv4 位址

請遵循該程序來啟動執行個體。當您設定網路設定時，請展開進階網路組態，然後輸入主要 IP 的值。

AWS CLI

若要在啟動時指派私有 IPv4 位址

請使用 `--private-ip-address` 選項並搭配 run-instances 命令。

```
--private-ip-addresses 10.251.50.12
```

若要讓 Amazon EC2 選擇 IP 位址，請忽略此選項。

PowerShell

若要在啟動時指派私有 IPv4 位址

使用 New-EC2Instance cmdlet 搭配 `-PrivateIpAddress` 參數。

```
-PrivateIpAddress 10.251.50.12
```

若要讓 Amazon EC2 選擇 IP 位址，請忽略此參數。

檢視主要 IPv4 位址。

公有 IPv4 地址會顯示為主控台中網路介面的屬性，但會透過 NAT 映射至主要私有 IPv4 地址。因此，如果您檢查執行個體上網路介面的屬性 (例如，透過 `ifconfig` (Linux) 或 `ipconfig` (Windows))，則不會顯示公有 IPv4 地址。

Console

若要檢視執行個體的 IPv4 位址

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 在聯網索引標籤中，尋找公有 IPv4 位址和私有 IPv4 位址。
5. (選用) 聯網索引標籤也包含執行個體的網路介面和彈性 IP 位址。

AWS CLI

若要檢視執行個體的主要 IPv4 位址

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].PrivateIpAddress" \
    --output text
```

以下為範例輸出。

```
10.251.50.12
```

PowerShell

若要檢視執行個體的主要 IPv4 位址

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.PrivateIpAddress
```

以下為範例輸出。

```
10.251.50.12
```

使用執行個體中繼資料來檢視 IPv4 位址

可以透過擷取執行個體中繼資料來取得執行個體的 IPv4 位址。如需詳細資訊，請參閱使用執行個體中繼資料管理您的 EC2 執行個體。

若要使用執行個體中繼資料來檢視 IPv4 位址

1. 連線到您的執行個體。如需詳細資訊，請參閱連線至 EC2 執行個體。
2. 請執行下列其中一個命令：

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/local-ipv4
```

Windows

透過 Windows 執行個體中執行以下命令。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/local-ipv4
```

Windows

透過 Windows 執行個體中執行以下命令。

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. 使用下列其中一個命令來存取公有 IP 位址。如果有與執行個體相關聯的彈性 IP 位址，命令會傳回彈性 IP 位址。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
 "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/public-ipv4
```

Windows

透過 Windows 執行個體中執行以下命令。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

Windows

透過 Windows 執行個體中執行以下命令。

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/public-ipv4
```

管理 EC2 執行個體的 IPv6 地址

如果已建立 VPC 和子網與 IPv6 CIDR 區塊的關聯，您可以在啟動期間或之後將 IPv6 地址指派給執行個體。您可以在主控台中透過執行個體頁面或網路介面頁面存取執行個體的 IPv6 位址。下列任務會設定執行個體的 IP 位址。若要改為設定網路介面的 IP 位址，請參閱 the section called “管理 IP 位址”。

任務

- 將 IPv6 地址指派給執行個體
- 檢視執行個體的 IPv6 位址
- 使用執行個體中繼資料來檢視 IPv6 位址
- 從執行個體取消指派 IPv6 地址

將 IPv6 地址指派給執行個體

您可以指定子網路 IPv6 位址範圍中的 IPv6 位址，或讓 Amazon EC2 為您選擇一個。此地址會指派給主要網路介面。請注意，下列執行個體類型不支援 IPv6 地址：C1、M1、M2、M3 和 T1。

Console

若要在啟動時指派 IPv6 位址

請遵循該程序來啟動執行個體。當您設定網路設定時，請選擇自動指派 IPv6 IP 的選項。如果沒有看到此選項，則選取的子網路沒有相關聯的 IPv6 CIDR 區塊。

在啟動之後指派 IPv6 地址

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Actions (動作)、Networking (聯網)、Manage IP addresses (管理 IP 地址)。
4. 展開網路介面。在 IPv6 addresses (IPv6 地址) 下選擇 Assign new IP address (指派新 IP 地址)。

5. 輸入子網路範圍中的 IPv6 位址，或將欄位保留空白，讓 Amazon EC2 為您選擇 IPv6 位址。如果沒有看到此選項，則執行個體子網路沒有相關聯的 IPv6 CIDR 區塊。
6. 選擇儲存。

AWS CLI

若要在啟動時指派 IPv6 位址

請使用 `--ipv6-addresses` 選項並搭配 run-instances 命令。下列範例會指派兩個 IPv6 位址。

```
--ipv6-addresses Ipv6Address=2001:db8::1234:5678:1.2.3.4
 Ipv6Address=2001:db8::1234:5678:5.6.7.8
```

若要讓 Amazon EC2 選擇 IPv6 位址，請改用 `--ipv6-address-count` 選項。下列範例會指派兩個 IPv6 位址。

```
--ipv6-address-count 2
```

在啟動之後指派 IPv6 地址

使用 assign-ipv6-addresses 命令。下列範例會指派兩個 IPv6 位址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 2001:db8::1234:5678:5.6.7.8
```

若要讓 Amazon EC2 選擇 IPv6 位址，請改用 `--ipv6-address-count` 選項。下列範例會指派兩個 IPv6 位址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-address-count 2
```

PowerShell

若要在啟動時指派 IPv6 位址

使用 New-EC2Instance cmdlet 搭配 `-Ipv6Address` 參數。下列範例會指派兩個 IPv6 位址。

```
-Ipv6Address $ipv6addr1,$ipv6addr2
```

定義 IPv6 位址，如下所示。

```
$ipv6addr1 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
$ipv6addr2 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr2.Ipv6Address = "2001:db8::1234:5678:5.6.7.8"
```

若要讓 Amazon EC2 選擇 IPv6 位址，請改用 `-Ipv6AddressCount` 參數。下列範例會指派兩個 IPv6 位址。

```
-Ipv6AddressCount 2
```

在啟動之後指派 IPv6 地址

使用 Register-EC2Ipv6AddressList cmdlet。下列範例會指派兩個 IPv6 位址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address "2001:db8::1234:5678:1.2.3.4","2001:db8::1234:5678:5.6.7.8"
```

若要讓 Amazon EC2 選擇 IPv6 位址，請改用 `-Ipv6AddressCount` 參數。下列範例會指派兩個 IPv6 位址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6AddressCount 2
```

檢視執行個體的 IPv6 位址

您可檢視執行個體的 IPv6 位址。

Console

若要檢視執行個體的 IPv6 位址

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取實例。

4. 在 Networking (聯網) 標籤上，找出 IPv6 addresses (IPv6 地址)。

AWS CLI

若要檢視執行個體的 IPv6 位址

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].Ipv6Address" \
    --output text
```

以下為範例輸出。

```
2001:db8::1234:5678:1.2.3.4
```

PowerShell

若要檢視執行個體的 IPv6 位址

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.Ipv6Address
```

以下為範例輸出。

```
2001:db8::1234:5678:1.2.3.4
```

使用執行個體中繼資料來檢視 IPv6 位址

連線至執行個體後，可以使用執行個體中繼資料來擷取 IPv6 位址。首先，必須從 `http://169.254.169.254/latest/meta-data/network/interfaces/macs/` 中取得執行個體的 MAC 位址。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

透過 Windows 執行個體執行下面的 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

透過 Windows 執行個體中執行下列 cmdlet。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

從執行個體取消指派 IPv6 地址

您可以隨時從執行個體取消指派 IPv6 地址。

Console

從執行個體取消指派 IPv6 地址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Actions (動作)、Networking (聯網)、Manage IP addresses (管理 IP 地址)。
4. 展開網路介面。在 IPv6 addresses (IPv6 地址) 下，選擇 IPv6 地址旁的 Unassign (取消指派)。
5. 選擇儲存。

AWS CLI

從執行個體取消指派 IPv6 地址

使用 unassign-ipv6-addresses 命令。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4
```

PowerShell

從執行個體取消指派 IPv6 地址

使用 Unregister-EC2Ipv6AddressList cmdlet。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8::1234:5678:1.2.3.4
```

EC2 執行個體的次要 IP 位址

指派給網路介面的第一個 IPv4 位址稱為主要 IP 位址。次要 IP 位址是指派給網路介面的其他 IPv4 位址。如需詳細資訊，請參閱the section called “多個 IP 地址”。

也可以向執行個體指派多個 IPv6 位址。如需詳細資訊，請參閱the section called “IPv6 地址”。

任務

- 將次要 IP 位址指派給執行個體
- 設定作業系統以使用次要 IP 位址
- 從執行個體中取消指派次要 IP 位址

將次要 IP 位址指派給執行個體

可以在啟動執行個體時或在執行個體執行之後，將次要 IP 位址指派給執行個體的網路介面。

Console

若要在啟動時指派次要 IP 位址

1. 請遵循該程序來啟動執行個體。配置網路設定時，請展開進階網路組態。
2. 針對次要 IP，選擇自動指派，然後輸入要為 Amazon EC2 指派的 IP 位址數量。或者，選擇手動指派並輸入 IPv4 位址。
3. 完成其餘步驟以啟動執行個體。

若要在啟動後指派次要 IP 位址

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 Actions (動作)、Networking (聯網)、Manage IP addresses (管理 IP 地址)。
4. 展開網路介面。
5. 若要新增 IPv4 位址，請在 IPv4 位址下，選擇指派新的 IP 位址。輸入子網路範圍中的 IPv4 位址，或將欄位保留空白，讓 Amazon EC2 為您選擇一個。
6. 選擇儲存。

AWS CLI

若要在啟動時指派次要 IP 位址

請使用 `--secondary-private-ip-addresses` 選項並搭配 run-instances 命令。

```
--secondary-private-ip-addresses 10.251.50.12
```

若要讓 Amazon EC2 選擇 IP 位址，請改用 `--secondary-private-ip-address-count` 選項。下列範例會指派一個次要 IP 位址。

```
--secondary-private-ip-address-count 1
```

或者，可以建立網路介面。如需詳細資訊，請參閱the section called “建立網路介面”。

若要在啟動後指派次要 IP 位址

使用具有 `--private-ip-addresses` 選項的 assign-private-ip-addresses 命令。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --private-ip-addresses 10.251.50.12
```

若要讓 Amazon EC2 選擇 IPv4 位址，請改用 `--secondary-private-ip-address-count` 參數。下列範例會指派一個 IPv4 位址。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --secondary-private-ip-address-count 1
```

PowerShell

若要在啟動時指派次要 IP 位址

您必須建立網路介面。如需詳細資訊，請參閱the section called “建立網路介面”。

若要在啟動後指派次要 IP 位址

使用具有 `-PrivateIpAddress` 參數的 Register-EC2PrivateIpAddress cmdlet。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

若要讓 Amazon EC2 選擇 IPv4 位址，請改用 `-SecondaryPrivateIpAddressCount` 參數。下列範例會指派一個 IPv4 位址。

```
Register-EC2PrivateIpAddress `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SecondaryPrivateIpAddressCount 1
```

設定作業系統以使用次要 IP 位址

在將次要 IP 位址指派給執行個體之後，必須在執行個體上設定作業系統，以識別其他私有 IPv4 位址。

Linux 執行個體

- 如果您使用的是 Amazon Linux，ec2-net-utils 套件可以為您處理此步驟。套件會設定您在執行個體執行時所連接的額外網路介面、在 DHCP 租賃續約期間重新整理輔助 IPv4 地址，以及更新相關路由規則。根據您的系統，可以使用下列其中一個命令來立即重新整理介面清單：`sudo systemctl restart systemd-networkd` (AL2023) 或 `sudo service network restart` (Amazon Linux 2)。可以使用下列命令來檢視最新清單：`ip addr li`。如果您需要手動控制網路組態，則可以移除 ec2-net-utils 套件。如需更多資訊，請查看使用 ec2-net-utils 設定您的網路介面。
- 如果您使用的是其他 Linux 發行版本，請參閱 Linux 發行版本的文件。搜尋有關設定額外網路介面和輔助 IPv4 地址的資訊。如果執行個體在相同的子網上有兩個或以上的介面，請搜尋有關使用路由規則暫時處理非對稱路由的資訊。

Windows 執行個體

如需詳細資訊，請參閱為 Windows 執行個體設定次要私有 IPv4 地址。

從執行個體中取消指派次要 IP 位址

如果不再需要次要 IP 位址，您可以從執行個體或網路介面中取消指派。從網路介面取消指派輔助私有 IPv4 地址時，也會取消關聯彈性 IP 地址 (若其存在)。

Console

從執行個體取消指派輔助私有 IPv4 地址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取動作、聯網、管理 IP 地址。
4. 展開網路介面。在 IPv4 地址下，為要取消指派的 IPv4 地址選擇取消指派。

5. 選擇儲存。

AWS CLI

若要取消指派次要私有 IP 位址

使用 unassign-private-ip-addresses 命令。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface eni-1234567890abcdef0\
    --private-ip-addresses 10.251.50.12
```

PowerShell

若要取消指派次要私有 IP 位址

使用 Unregister-EC2PrivateIpAddress cmdlet。

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterface eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

為 Windows 執行個體設定次要私有 IPv4 地址

您可以指定執行個體的多個私有 IPv4 地址。將輔助私有 IPv4 地址指派給執行個體之後，必須設定執行個體的作業系統，以辨識輔助私有 IPv4 地址。

> **Note**
>
> 這些說明是以 Windows Server 2022 為基礎。根據 Windows 執行個體的作業系統，這些步驟的實作可能會不同。

任務

- 先決條件
- 步驟 1：在您的執行個體上設定靜態 IP 定址
- 步驟 2：設定您的執行個體的次要私有 IP 地址

- 步驟 3：設定應用程式使用次要私有 IP 地址

先決條件

- 將次要私有 IPv4 地址指派給執行個體的網路介面。您可以在啟動執行個體時或在執行個體執行之後，指派次要私有 IPv4 地址。如需詳細資訊，請參閱將次要 IP 位址指派給執行個體。

步驟 1：在您的執行個體上設定靜態 IP 定址

若要讓 Windows 執行個體使用多個 IP 地址，您必須設定執行個體使用靜態 IP 定址，而不是 DHCP 伺服器。

> **Important**
>
> 當您在執行個體上設定靜態 IP 定址時，IP 地址必須完全符合主控台、CLI 或 API 中所顯示的 IP 位址。如果您不正確地輸入這些 IP 地址，執行個體可能會變成無法存取。

在 Windows 執行個體上設定靜態 IP 定址

1. 連線到您的執行個體。
2. 執行下列步驟，尋找執行個體的 IP 地址、子網遮罩和預設閘道地址：
   - 在 PowerShell 中執行下列命令：

     ```
     ipconfig /all
     ```

     檢閱輸出並記下網路介面的 IPv4 地址、子網路遮罩、預設閘道和 DNS 伺服器值。輸出應類似以下範例：

     ```
     ...

     Ethernet adapter Ethernet 4:

        Connection-specific DNS Suffix  . : us-west-2.compute.internal
        Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
        Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
        DHCP Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
     ```

```
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, April 8, 2024 12:19:29 PM
   Lease Expires . . . . . . . . . . : Monday, April 8, 2024 4:49:30 PM
   Default Gateway . . . . . . . . . : 10.200.0.1
   DHCP Server . . . . . . . . . . . : 10.200.0.1
   DHCPv6 IAID . . . . . . . . . . . : 151166011
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-
E7
   DNS Servers . . . . . . . . . . . : 10.200.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

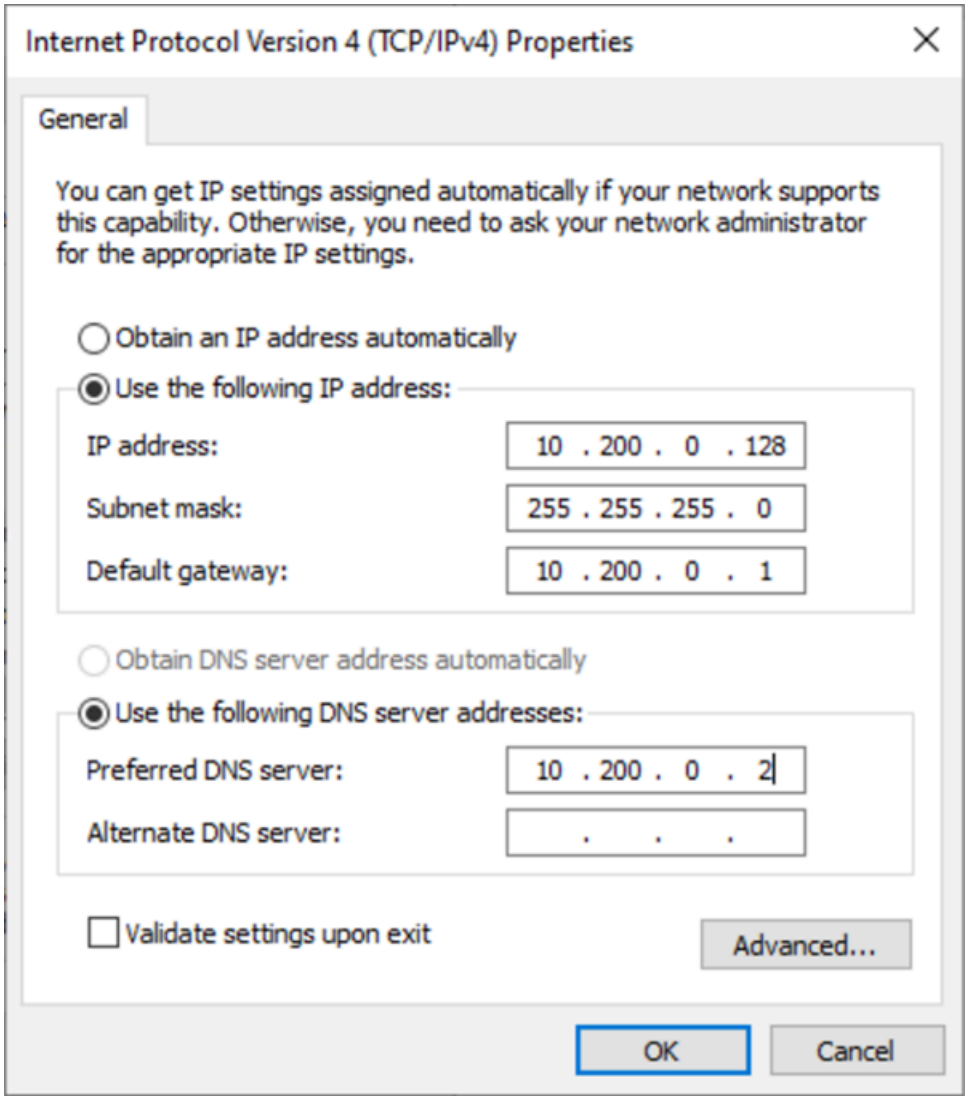
3. 在 PowerShell 執行以下命令開啟網路和共用中心：

```
& $env:SystemRoot\system32\control.exe ncpa.cpl
```

4. 開啟網路介面 (區域連線或乙太網路) 的內容 (右鍵) 選單，然後選擇內容。
5. 選擇網際網路通訊協定第 4 版 (TCP/IPv4)、內容。
6. 在網際網路通訊協定第 4 版 (TCP/IPv4) - 內容對話方塊中，選擇使用下列 IP 地址，並輸入下列值，然後選擇確定。

欄位	Value
IP 地址	前述步驟 2 中取得的 IPv4 地址。
子網遮罩	前述步驟 2 中取得的子網遮罩。
預設閘道	前述步驟 2 中取得的預設閘道地址。
慣用 DNS 伺服器	前述步驟 2 中取得的 DNS 伺服器。
其他 DNS 伺服器	前述步驟 2 中取得的其他 DNS 伺服器。如果未列出其他 DNS 伺服器，請將此欄位保留空白。

> **Important**
>
> 如果您將 IP 地址設定為目前 IP 地址以外的任何值，則會中斷與執行個體的連線。



您將會中斷與 Windows 執行個體的 RDP 連線數秒，而執行個體會從使用 DHCP 轉換成靜態定址。執行個體會保留先前的 IP 地址資訊，但現在此資訊是靜態的，且並非透過 DHCP 管理。

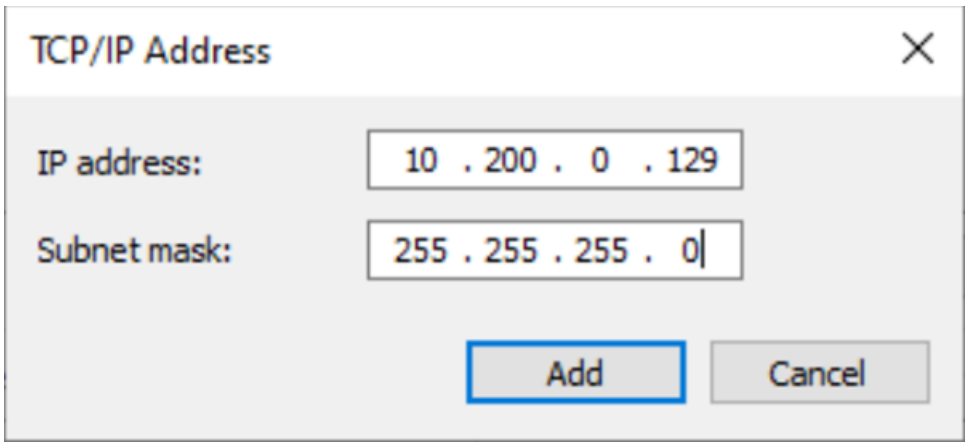
步驟 2：設定您的執行個體的次要私有 IP 地址

在您設定 Windows 執行個體的靜態 IP 定址之後，即可準備第二個私有 IP 地址。

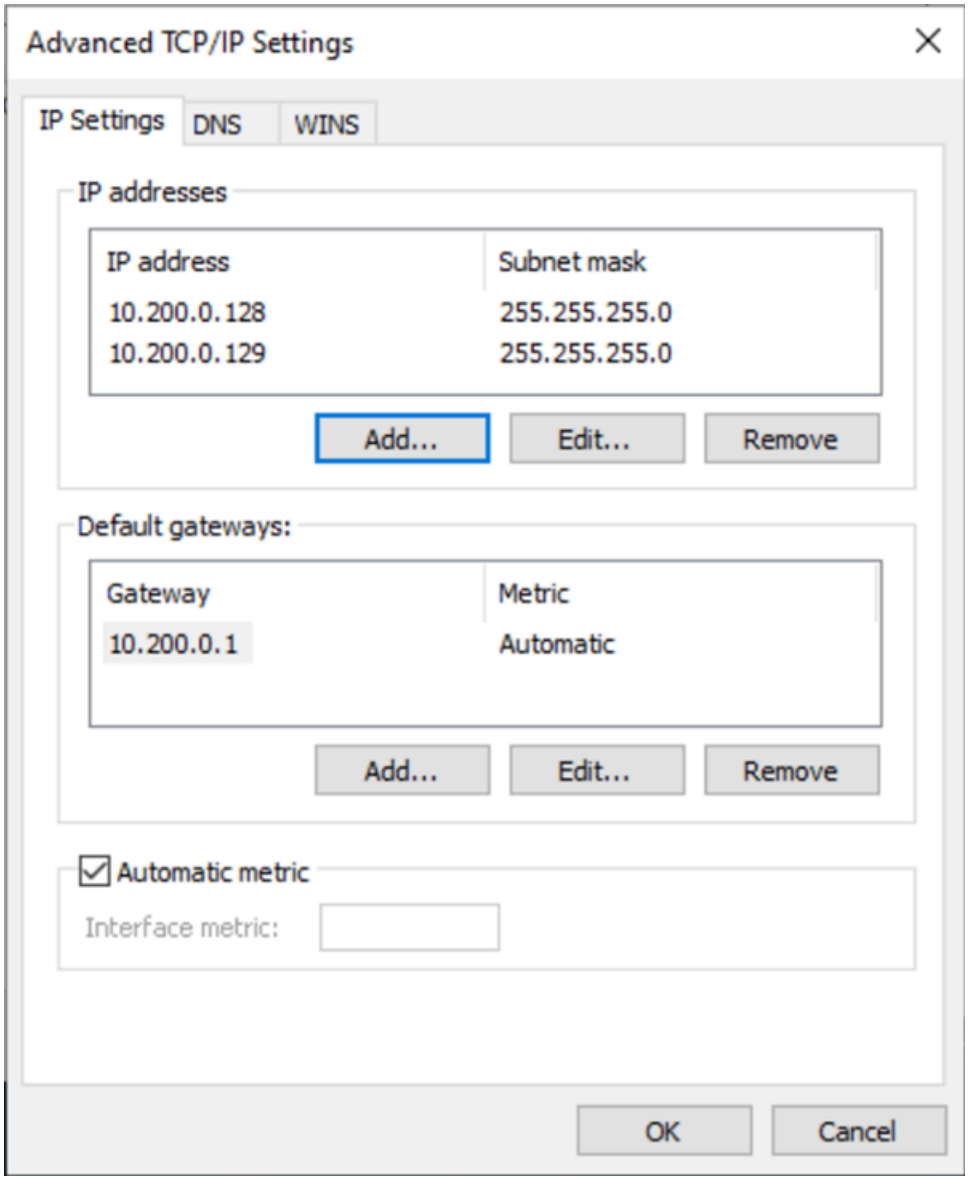
設定次要 IP 地址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 在 Networking (聯網) 標籤上，記下輔助 IP 地址。

4. 連線到您的執行個體。
5. 在 Windows 執行個體上，選擇開始、控制台。
6. 選擇網路和網際網路、網路和共用中心。
7. 選取網路介面 (區域連線或乙太網路)，然後選取內容。
8. 在區域連線內容頁面上，選擇網際網路通訊協定第 4 版 (TCP/IPv4)、內容、進階。
9. 選擇 Add (新增)。
10. 在 TCP/IP 地址對話方塊中，針對 IP 地址輸入輔助私有 IP 地址。針對子網遮罩，輸入您在步驟 1：在您的執行個體上設定靜態 IP 定址中於主要私有 IP 地址輸入的相同子網遮罩，然後選擇新增。



11. 確認 IP 地址設定，然後選擇確定。



12. 選擇確定、關閉。
13. 若要確認次要 IP 位址已新增至作業系統，請在 PowerShell 執行 `ipconfig /all` 命令。輸出應類似以下內容：

```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
   Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
```

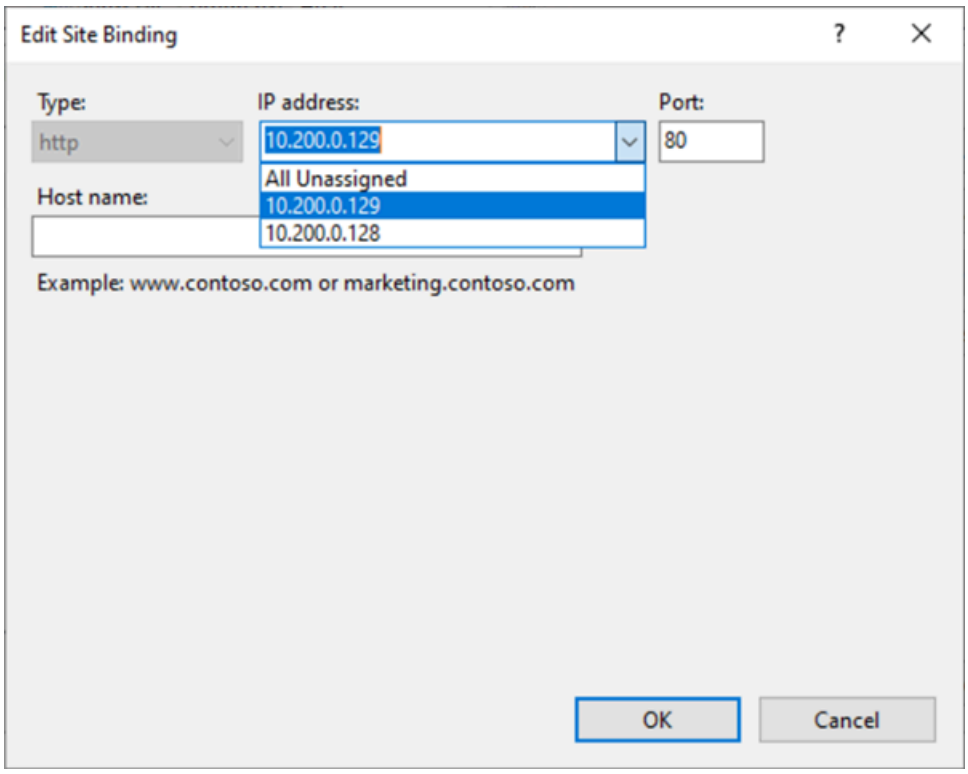
```
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.200.0.129(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.0.1
   DHCPv6 IAID . . . . . . . . . . . : 151166011
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-E7
   DNS Servers . . . . . . . . . . . : 10.200.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

步驟 3：設定應用程式使用次要私有 IP 地址

您可以設定任何應用程式使用輔助私有 IP 地址。例如，如果您的執行個體在 IIS 上執行網站，即可設定 IIS 使用輔助私有 IP 地址。

設定 IIS 使用輔助私有 IP 地址

1. 連線到您的執行個體。
2. 開啟 Internet Information Services (IIS) 管理員。
3. 在連線窗格中，展開網站。
4. 開啟您網站的內容 (右鍵) 選單，然後選擇編輯繫結。
5. 在網站繫結對話方塊中，針對類型，選擇 http、編輯。
6. 在編輯網站繫結對話方塊中，針對 IP 地址，選取輔助私有 IP 地址。(根據預設，每個網站都接受來自所有 IP 地址的 HTTP 請求)。



7. 選擇確定、關閉。

EC2 執行個體主機名稱與網域

了解 EC2 執行個體主機名稱與網域，對於有效管理和存取您的 Amazon EC2 執行個體至關重要。每個 EC2 執行個體可擁有不同類型的主機名稱 (私有與公有)，其用途各異且遵循特定命名慣例。

本主題說明 EC2 執行個體主機名稱的結構、建構方式，以及可用的不同主機名稱類型。您將了解如何檢視和修改主機名稱設定、了解何時使用每種類型，以及了解 AWS 環境中主機名稱管理的最佳實務。

目錄

- 了解 EC2 執行個體主機名稱與網域
- 主機名稱類型

了解 EC2 執行個體主機名稱與網域

EC2 執行個體位址由不同元件組成。以下是使用執行個體私有 IPv4 位址的 EC2 執行個體位址範例：

```
IP address         Domain name
#--------# #------------------------#
```

```
ip-10-24-34-0.us-west-2.compute.internal
#-----------#
  Hostname
#-------------------------------------#
    Fully qualified domain name (FQDN)
```

其中：

- IP 位址：與執行個體相關聯的主要網路介面的主要 IPv4 位址。
- 主機名稱：特定 EC2 執行個體的本機名稱 (供作業系統使用，用於本機網路識別)
- 網域名稱： AWS 提供的 FQDN 部分
- 完整網域名稱 (FQDN)：包含主機名稱和網域名稱的完整位址。這是用於跨網路存取您執行個體的完整、全球唯一識別碼。

根據您為執行個體或連接至執行個體的主要網路介面選擇的主機名稱類型，主機名稱與網域名稱格式會與上述範例不同。本節介紹主機名稱類型選項。

主機名稱類型

AWS 提供兩種類型的主機名稱：私有和公有。下表比較私有與公有主機名稱之間的主要差異，包括解析方式、設定方式，以及各類型的適用場景。

	私有主機名稱	公有主機名稱
DNS 解析	私有主機名稱支援無法從公有網際網路存取的私有 FQDN。私有主機名稱僅允許請求解析至 VPC 內私有 IPv4 與 IPv6 GUA 位址。	公有主機名稱可啟用可從公有網際網路存取的公有 FQDN。公有主機名稱可讓請求解析至 VPC 內的私有 IPv4 與 IPv6 GUA，以及來自網際網路的公有 IP (水平分割 DNS)。
組態	私有主機名稱在執行個體層級設定。	公有主機名稱在網路介面層級設定。
使用情況	• 對於僅在 VPC 內解析的主機名稱，用於內部通訊 • 確保內部命名結構描述的私密性 • 簡化內部資源管理，無需考量外部 DNS	• 從網際網路透過 IPv4 與 IPv6 存取執行個體 • 實現從 IPv4 順利移轉至 IPv6 環境

目錄

- 公有主機名稱
- 私有主機名稱

公有主機名稱

您可以使用公有主機名稱存取 EC2 執行個體，這類主機名稱會解析至執行個體的公有 IPv4 或 IPv6 位址，協助您順利移轉至 IPv6。

公有主機名稱可讓您：

- 透過 IPv4 和 IPv6 存取 EC2 執行個體，讓您能更靈活地連線到資源。
- 按照自身節奏從 IPv4 移轉至 IPv6 環境。例如，您可以分離資料庫與應用程式的移轉作業，降低複雜度與風險。
- 使用自動解析至適當 IP 位址的多個主機名稱選項 (僅 IPv4、IPv6-only 和雙堆疊)。
- 透過水平分割 DNS 提升安全性，從您的 VPC 內查詢時，會將主機名稱解析至私人 IP 位址。

目錄

- 公有主機名稱類型與適用場景
- 檢視公有主機名稱
- 修改公有主機名稱類型

公有主機名稱類型與適用場景

要使用公有主機名稱，必須修改現有的網路介面。本節說明三個公有主機名稱類型選項，協助您選擇合適的類型：

- 雙堆疊 - IP 型名稱
  - 若您要移轉或計劃從 IPv4 移轉至 IPv6，這是最佳選項。此選項允許透過 IPv4 和 IPv6 進行連線，為可能使用任一通訊協定的用戶端提供彈性，並讓用戶端在整個移轉至 IPv6 的過程中保持相同主機名稱。
  - 來自 VPC 內的請求會解析至網路介面的私有 IPv4 位址與 IPv6 全域單播位址 (GUA)。來自網際網路的請求會同時解析至網路介面的公有 IPv4 與 IPv6 GUA 位址。
  - 範例
    - 當您選擇此選項時，會為此網路介面產生雙堆疊 FQDN。以下是將產生的 FQDN 範例：

    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO.ap-southeast-2.ip.aws
  - 其中：
    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO 是主機名稱，由網路介面上主要公有 IPv6 位址的 base36 表示法 (f5lnz-0khrm-nt2u3-gyqqt-nbdl5) 與主要公有 IPv4 位址的 base36 表示法 (q3cdpO) 組成。
    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 由 Amazon DNS 解析程式解析至 IPv6 GUA 位址 FFFF:1407:4:f000:81d:2689:1066:4489。這是指派給網路介面的 IPv6 GUA。
    - q3cdpO 會解析至 IPv4 位址 52.54.55.56。這是連接至主要網路介面的公有 IPv4 位址。
    - ap-southeast-2 是網路介面所屬的子網路區域。
    - ip AWS.aws 是由 提供的網域。
- IPv6 - IP 型名稱
  - 若您已經移轉至 IPv6 且僅需透過 IPv6 進行連線，這是合適的選擇。
  - 來自 VPC 內或來自網際網路的請求會解析至網路介面的 IPv6 GUA。
  - 範例
    - 當您選擇此選項時，會為此網路介面產生 FQDN。以下是將產生的 FQDN 範例：
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
    - 其中：
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 是主機名稱，為網路介面上主要公有 IPv6 位址的 base36 表示法。
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 由 Amazon DNS 解析程式解析至 IPv6 GUA 位址 FFFF:1407:4:f000:81d:2689:1066:4489。這是指派給網路介面的 IPv6 GUA。
      - ap-southeast-2 是網路介面所屬的子網路區域。
      - ip AWS.aws 是由 提供的網域。
- IPv4 - IP 型名稱
  - 若使用此網路介面的執行個體需要在移轉至 IPv6 期間維持 IPv4 存取，或在執行個體上執行的應用程式或系統僅支援 IPv4，這是合適的選項。若您僅需維持 IPv4 連線，且工作負載不需要 IPv6 支援，這是最佳選項。例如，若您要移轉至 IPv6，您可能會決定讓部分應用程式保留在 IPv4，其餘則移轉至 IPv6。
  - VPC 內的請求會解析至網路介面的私有主要 IPv4 位址。來自網際網路的請求會解析至網路介面的公有 IPv4 位址。
  - 範例

- 若您選擇此選項，將會為此網路介面產生啟用 IPv4 的公有主機名稱。以下是將產生的 DNS 名稱範例：
  - ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
- 其中：
  - ec2-52-54-55-66 是衍生自網路介面主要公有 IPv4 地址的主機名稱，格式為 ec2-，後面接著以連字號取代的點 IP 地址。
  - ec2-52-54-55-66 解析為 IPv4 地址 52.54.55.66。這是連接至主要網路介面的公有 IPv4 位址。
  - ap-southeast-2 是網路介面所屬的子網路區域。
  - compute.amazonaws.com 是 AWS 為 IPv4-based公有主機名稱提供的網域。

> **Important**
>
> 從上述範例中可見，主機名稱透過 IP 位址產生。若您變更指派給網路介面的主要私有 IPv4 位址或第一個 IPv6 GUA，主機名稱中對應 IP 位址的部分將會變更，且先前產生的公有主機名稱將不再有效。此外，變更主要 IPv4 公有位址會強制在中下游重新整理《Amazon EC2 使用者指南》中的執行個體中繼資料服務 (IMDS)，因此 EC2 執行個體的 IMDS 中繼資料會自動更新。

檢視公有主機名稱

若網路介面所屬的 VPC 未同時啟用 EnableDnsHostnames 和 EnableDnsSupport，則不會定義或產生主機名稱類型。

Console

您可檢視執行個體或主要網路介面的公有主機名稱。

檢視執行個體的主機名稱類型與 DNS 名稱

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體的核取方塊。
4. 在網路索引標籤的主機名稱和 DNS 下，找到下列項目：
   - 公有主機名稱類型

- 公有 DNS
- 僅 IPv4 IP 型名稱
- 僅 IPv6 - IP 型名稱
- 雙堆疊 - IP 型名稱

檢視網路介面的主機名稱類型和 DNS 名稱

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 在搜尋欄位中輸入執行個體的 ID。選取網路介面 ID，開啟其詳細資訊頁面。
4. 在主機名稱和 DNS 下，尋找下列項目：
   - 公有主機名稱類型
   - 公有 DNS 名稱
   - 公有 IPv4 DNS 名稱
   - 公有 IPv6 DNS 名稱
   - 公有雙堆疊 DNS 名稱

AWS CLI

檢視網路介面的主機名稱類型和 DNS 名稱

使用 describe-network-interfaces 命令。

```
aws ec2 describe-network-interfaces \
    --network-interface-id eni-1234567890abcdef0 \
    --query NetworkInterfaces[].PublicIpDnsNameOptions
```

以下為範例輸出。由於主機名稱類型為 public-dual-stack-dns-name，DNS 主機名稱與 PublicDualStackDnsName 相同。

```
[
     {
          "DnsHostnameType": "public-dual-stack-dns-name",
          "PublicIpv4DnsName": "ec2-52-54-55-66.ap-
southeast-2.compute.amazonaws.com",
```

```
          "PublicIpv6DnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-
southeast-2.ip.aws",
          "PublicDualStackDnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-
southeast-2.ip.aws"
     }
]
```

PowerShell

檢視網路介面的主機名稱類型和 DNS 名稱

使用 Get-EC2NetworkInterface cmdlet。

```
(Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0).PublicIpDnsNameOptions
```

以下為範例輸出。由於主機名稱類型為 public-dual-stack-dns-name，DNS 主機名稱與 PublicDualStackDnsName 相同。

```
DnsHostnameType        : public-dual-stack-dns-name
PublicDualStackDnsName : f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-southeast-2.ip.aws
PublicIpv4DnsName      : ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
PublicIpv6DnsName      : f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
```

修改公有主機名稱類型

公有主機名稱類型選項取決於與網路介面相關聯的 IP 位址：

- 若網路介面僅有公有 IPv4 位址，主機名稱類型必須為 IPv4 - IP 型名稱。
- 若網路介面僅有 IPv6 位址，主機名稱類型必須為 IPv6 - IP 型名稱。
- 若網路介面同時具有公有 IPv4 位址與 IPv6 位址，主機名稱類型可為雙堆疊 - IP 型名稱。

先決條件

- 網路介面必須具有相關聯的公有 IPv4 位址或 IPv6 位址。
- 網路介面所在的 VPC 必須已啟用 EnableDnsHostnames 和 EnableDnsSupport。請參閱《Amazon VPC 使用者指南》中的檢視和更新 VPC 的 DNS 屬性。

Console

修改公有主機名稱類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 在搜尋欄位中輸入執行個體的 ID。選取網路介面的核取方塊。

   或者，從執行個體詳細資訊頁面，選擇聯網索引標籤，並選取裝置索引 0 的網路介面 ID。
4. 選擇動作、修改公有主機名稱類型。
5. 選擇一個選項：
   - 雙堆疊 - IP 型名稱：網路介面的雙堆疊公有主機名稱。來自 VPC 內的請求會解析至網路介面的私有 IPv4 位址與 IPv6 全域單播位址。來自網際網路的請求會同時解析至網路介面的公有 IPv4 與 IPv6 GUA 位址。
   - IPv4 - IP 型名稱：網路介面的啟用 IPv4 公有主機名稱。VPC 內的請求會解析至網路介面的私有主要 IPv4 位址。來自網際網路的請求會解析至網路介面的公有 IPv4 位址。
   - IPv6 - IP 型名稱：網路介面的啟用 IPv6 公有主機名稱。來自 VPC 內或來自網際網路的請求會解析至網路介面的 IPv6 GUA。
6. 選擇 Modify (修改)。

AWS CLI

修改公有主機名稱類型

使用 modify-public-ip-dns-name-options 命令。

```
aws ec2 modify-public-ip-dns-name-options \
    --network-interface-id eni-1234567890abcdef0 \
    --hostname-type public-dual-stack-dns-name
```

以下為範例輸出。

```
{
    "Successful": true
}
```

PowerShell

修改公有主機名稱類型

使用 Edit-EC2PublicIpDnsNameOption cmdlet。

```
Edit-EC2PublicIpDnsNameOption `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -HostNameType public-dual-stack-dns-name
```

私有主機名稱

本節說明將執行個體啟動至 VPC 子網路時可用的 Amazon EC2 執行個體私有主機名稱。

私有主機名稱可區分您網路上的 EC2 執行個體。例如，如果您希望執行指令碼，與網路上的部分或全部執行個體通訊，則可以使用執行個體的私有主機名稱。

目錄

- 私有主機名稱類型
- 如何尋找資源名稱和 IP 名稱
- 在資源名稱和 IP 名稱之間進行選擇
- 變更 Amazon EC2 的資源型命名的選項

私有主機名稱類型

在 VPC 中啟動 EC2 執行個體時，客體作業系統主機名稱有兩種私有主機名稱類型：

- IP name (IP 名稱)：舊式命名結構描述，其中當您啟動執行個體時，執行個體的私有 IPv4 地址會包含在執行個體的主機名稱中。IP 名稱存在於 EC2 執行個體的生命週期中。用作私有 DNS 主機名時，它只會傳回私有 IPv4 地址 (A 記錄)。
- 資源名稱：當您啟動執行個體時，EC2 執行個體 ID 會包含在執行個體的主機名稱中。資源名稱存在於 EC2 執行個體的生命週期中。用作私有 DNS 主機名稱時，它可以同時傳回私有 IPv4 地址 (A 記錄) 和/或 IPv6 全域單點傳送地址 (AAAA 記錄)。

EC2 執行個體客體作業系統主機名稱類型取決於子網路設定：

- 如果執行個體啟動至僅限 IPv4 子網路，您可以選取 IP 名稱或資源名稱。

- 如果執行個體啟動至雙堆疊 (IPv4+IPv6) 子網路，您可以選取 IP 名稱或資源名稱。
- 如果執行個體啟動至僅限 IPv6 子網路，則會自動使用資源名稱。

目錄

- IP 名稱
- 資源名稱
- IP 名稱與資源名稱之間的區別

IP 名稱

當您使用 Hostname type (主機名稱類型) 的 IP name (IP 名稱) 啟動 EC2 執行個體時，客體作業系統主機名稱會設定為使用私有 IPv4 地址。

- us-east-1 中執行個體的格式：*private-ipv4-address*.ec2.internal
- 範例：*ip-10-24-34-0*.ec2.internal
- 任何其他 AWS 區域中執行個體的格式： *private-ipv4-address.region*.compute.internal
- 範例：*ip-10-24-34-0.us-west-2*.compute.internal

資源名稱

在僅限 IPv6 子網路中啟動 EC2 執行個體時，預設會將 Hostname type (主機名稱類型) 選取為 Resource name (資源名稱)。在僅限 IPv4 或雙堆疊 (IPv4+IPv6) 子網路中啟動執行個體時，Resource name (資源名稱) 是可選取的選項。在啟動執行個體之後，您可以管理主機名稱組態。如需詳細資訊，請參閱變更 Amazon EC2 的資源型命名的選項。

當您使用 Hostname type (主機名稱類型) 的 Resource name (資源名稱) 啟動 EC2 執行個體時，客體作業系統主機名稱會設定為使用 EC2 執行個體 ID。

- us-east-1 中執行個體的格式：*ec2-instance-id*.ec2.internal
- 範例：*i-0123456789abcdef*.ec2.internal
- 任何其他 AWS 區域中執行個體的格式： *ec2-instance-id.region*.compute.internal
- 範例：*i-0123456789abcdef.us-west-2*.compute.internal

IP 名稱與資源名稱之間的區別

IP 名稱與資源名稱的 DNS 查詢共存，以確保向後相容性並允許您從 IP 型主機命名遷移至資源型命名。若為以 IP 名稱為基礎的私有 DNS 主機名稱，您無法設定是否要回應執行個體的 DNS A 記錄查詢。DNS A 記錄查詢始終會回應，而不管客體作業系統主機名稱設定如何。相反地，若為以資源名稱為基礎的私有 DNS 主機名稱，您可以設定是否要回應執行個體的 DNS A 和/或 DNS AAAA 查詢。您可以在啟動執行個體或修改子網路時設定回應行為。如需詳細資訊，請參閱變更 Amazon EC2 的資源型命名的選項。

如何尋找資源名稱和 IP 名稱

您可以在 Amazon EC2 主控台中查看主機名稱類型、資源名稱和 IP 名稱。

目錄

- 建立 EC2 執行個體時
- 檢視現有 EC2 執行個體的詳細資訊時

建立 EC2 執行個體時

當您建立 EC2 執行個體時，根據您選取的子網路類型，Hostname type (主機名稱類型) 的 Resource name (資源名稱) 可能是可用的，或者可能已選取且無法修改。本節介紹您查看主機名稱類型資源名稱和 IP 名稱的案例。

案例 1

您可以在精靈中建立 EC2 執行個體 (請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體)，並在設定詳細資訊時，選擇已設定為僅 IPv6 的子網路。

在此情況下，Hostname type (主機名稱類型) 的 Resource name (資源名稱) 會自動選取且無法修改。Enable IP name IPv4 (A record) DNS requests (啟用 IP 名稱 IPv4 (A 記錄) DNS 請求) 和 Enable resource-based IPv4 (A record) DNS requests (啟用以資源為基礎的 IPv4 (A 記錄) DNS 請求) 的 DNS Hostname (DNS 主機名稱) 選項將自動取消選取，並且不可修改。預設會選取 Enable resource-based IPv6 (AAAA record) DNS requests (啟用以資源為基礎的 IPv6 (AAAA 記錄) DNS 請求)，但可以修改。如果選取此項，對資源名稱的 DNS 請求將解析為此 EC2 執行個體的 IPv6 地址 (AAAA 記錄)。

案例 2

您可以在精靈中建立 EC2 執行個體 (請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體)，並在設定詳細資訊時，選擇以 IPv4 CIDR 區塊設定的子網路，或同時以 IPv4 和 IPv6 CIDR 區塊 (「雙堆疊」) 設定的子網路。

在此情況下，會自動選取 Enable IP name IPv4 (A record) DNS requests (啟用 IP 名稱 IPV4 (A 記錄) DNS 請求) 且無法變更。這表示對 IP 名稱的請求將解析為此 EC2 執行個體的 IPv4 地址 (A 記錄)。

選項預設為子網路的組態，但您可以修改此執行個體的選項，取決於子網路設定：

- Hostname type (主機名稱類型)：決定要將 EC2 執行個體的客體作業系統主機名稱做為資源名稱還是 IP 名稱。預設值為 IP name (IP 名稱)。
- Enable resource-based IPv4 (A record) DNS requests (啟用資源型 IPv4 (A 記錄) DNS 請求)：決定對資源名稱的請求是否解析為此 EC2 執行個體的私有 IPv4 地址 (A 記錄)。預設不會選取此選項。
- Enable resource-based IPv6 (AAAA record) DNS requests (啟用資源型 IPv6 (AAAA 記錄) DNS 請求)：決定對資源名稱的請求是否解析為此 EC2 執行個體的 IPv6 GUA 地址 (AAAA 記錄)。預設不會選取此選項。

檢視現有 EC2 執行個體的詳細資訊時

您可以在 EC2 執行個體的 Details (詳細資訊) 標籤中查看現有 EC2 執行個體的主機名稱值：

- Hostname type (主機名稱類型)：IP 名稱或資源名稱格式的主機名稱。
- Private IP DNS name (IPv4 only) (私有 IP DNS 名稱 (僅限 IPv4))：IP 名稱，其一律會解析為執行個體的私有 IPv4 地址。
- Private resource DNS name (私有資源 DNS 名稱)：資源名稱，其會解析為針對此執行個體選取的 DNS 記錄。
- Answer private resource DNS name (回答私有資源 DNS 名稱)：資源名稱，其會解析為 IPv4 (A)、IPv6 (AAA) 或 IPv4 和 IPv6 (A 和 AAA) DNS 記錄。

此外，如果直接透過 SSH 連線至 EC2 執行個體，並輸入 `hostname` 命令，您會看到採用 IP 名稱或資源名稱格式。

在資源名稱和 IP 名稱之間進行選擇

啟動 EC2 執行個體時 (請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體)，如果您將 Hostname type (主機名稱類型) 選擇為 Resource name (資源名稱)，EC2 執行個體將以資源名稱格式的主機名稱啟動。在這種情況下，此 EC2 執行個體的 DNS 記錄也可以指向資源名稱。這讓您能夠靈活地選擇該主機名稱是解析為執行個體的 IPv4 地址、IPv6 地址，還是同時解析為 IPv4 和 IPv6 地址。如果您計劃在將來使用 IPv6，或者如果您目前正在使用雙堆疊子網路，則最好針對 Hostname type (主機名稱類型) 使用 Resource name (資源名稱)，以便您變更執行個體主機名稱的 DNS 解析，而無需

對 DNS 記錄本身做出任何變更。資源名稱允許您在 EC2 執行個體上新增和移除 IPv4 和 IPv6 DNS 解析。

而如果您將 Hostname type (主機名稱類型) 選擇為 IP name (IP 名稱)，並將其用作 DNS 主機名稱，則只能解析為執行個體的 IPv4 地址。即使執行個體同時具有 IPv4 地址和與之關聯的 IPv6 地址，也不會解析為執行個體的 IPv6 地址。

變更 Amazon EC2 的資源型命名的選項

您可以變更子網路的主機名稱類型和 DNS 主機名稱組態，這會影響該主體中的所有後續執行個體啟動，也可以在啟動 EC2 執行個體之後對其進行變更。

資源型命名的選項

- 主機名稱類型：決定將 EC2 執行個體 (已在子網路中啟動) 之客體作業系統主機名稱的預設設定。此名稱可為資源名稱或 IP 名稱
- 啟用 DNS 主機名稱 IPV4 (A 記錄) 請求：決定對資源名稱的 DNS 請求/查詢是否解析為此 EC2 執行個體的私有 IPv4 地址 (A 記錄)。
- 啟用 DNS 主機名稱 IPV6 (AAAA 記錄) 請求：決定對資源名稱的 DSN 請求/查詢是否解析至此 EC2 執行個體的 IPv6 地址 (AAAA 記錄)。

子網路

變更子網路設定不會變更已在子網路中啟動的 EC2 執行個體的組態。

Console

修改子網路的選項

開啟 Amazon VPC 主控台並選取子網路。選擇動作、編輯子網路設定。視需要修改設定並儲存變更。

AWS CLI

修改子網路的選項

使用 modify-subnet-attribute 命令。

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-0abcdef1234567890 \
    --private-dns-hostname-type-on-launch resource-name \
```

```
    --enable-resource-name-dns-a-record-on-launch \
    --enable-resource-name-dns-aaaa-record-on-launch
```

PowerShell

修改子網路的選項

使用 Edit-EC2SubnetAttribute cmdlet。

```
Edit-EC2SubnetAttribute `
    -SubnetId subnet-0abcdef1234567890 `
    -PrivateDnsHostnameTypeOnLaunch ResourceName `
    -EnableResourceNameDnsAAAARecordOnLaunch $true `
    -EnableResourceNameDnsARecordOnLaunch $true
```

EC2 執行個體

考量事項

- 若要變更主機名稱類型，您必須先停止執行個體。變更其他兩個選項無需停止執行個體。
- 由於無法停止具有執行個體儲存體根磁碟區的執行個體，因此只能在執行個體啟動時設定主機名稱類型與 DNS 主機名稱選項。只有下列執行個體類型支援執行個體儲存體根磁碟區：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

Console

修改執行個體的主機名稱類型和 DNS 主機名稱選項

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 如果您即將變更 Use resource based naming as guest OS hostname (使用資源型命名作為客體作業系統主機名稱) 設定，首先停止 EC2 執行個體。否則，請跳過這個步驟。

   若要停止執行個體，請選取執行個體，然後選取 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
3. 選取執行個體，然後選取 Actions (動作)、Instance settings (執行個體設定)、Change resource based naming options (變更資源型命名選項)。

   - Use resource based naming as guest OS hostname (使用資源型命名作為客體作業系統主機名稱)：決定要將 EC2 執行個體的客體作業系統主機名稱做為資源名稱還是 IP 名稱。

- Answer DNS hostname IPv4 (A record) requests (回答 DNS 主機名稱 IPV4 (A 記錄) 請求)：決定對資源名稱的 DNS 請求/查詢是否解析為此 EC2 執行個體的私有 IPv4 地址。
- Answer DNS hostname IPv6 (AAAA record) requests (回答 DNS 主機名稱 IPV6 (AAAA 記錄) 請求)：決定對資源名稱的 DSN 請求/查詢是否解析為此 EC2 執行個體的 IPv6 地址 (AAAA 記錄)。

4. 選擇儲存。
5. 如果已停止執行個體，請重新啟動它。

AWS CLI

修改執行個體的主機名稱類型和 DNS 主機名稱選項

使用 modify-private-dns-name-options 命令。

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --private-dns-hostname-type resource-name \
    --enable-resource-name-dns-a-record \
    --enable-resource-name-dns-aaaa-record
```

PowerShell

修改執行個體的主機名稱類型和 DNS 主機名稱選項

使用 Edit-EC2PrivateDnsNameOption cmdlet。

```
Edit-EC2PrivateDnsNameOption `
    -InstanceId i-1234567890abcdef0 `
    -PrivateDnsHostnameType ResourceName `
    -EnableResourceNameDnsAAAARecord $true`
    -EnableResourceNameDnsARecord $true
```

將自有 IP 位址 (BYOIP) 用於 Amazon EC2

您可以將內部部署網路的一部分或全部公有可路由 IPv4 或 IPv6 地址範圍用於您的 AWS 帳戶。您可以繼續控制地址範圍，也可以透過網際網路公告地址範圍 AWS。將地址範圍帶入 Amazon EC2 之後，其會在 中 AWS 帳戶 顯示為地址集區。

> **Note**
>
> 本文件說明如何使用您自己的 IP 位址範圍僅供在 Amazon EC2 中使用。若要使用您自己的 IP 地址範圍， AWS Global Accelerator請參閱《 AWS Global Accelerator 開發人員指南》中的使用您自己的 IP 地址 (BYOIP)。若要使用您自己的 IP 地址範圍， Amazon VPC IP Address Manager請參閱《Amazon VPC IPAM 使用者指南》中的教學課程：將您的 IP 地址帶到 IPAM。

當您將 IP 地址範圍帶至 時 AWS， 會 AWS 驗證您是否控制 IP 地址範圍。您可以使用兩種方法來證明您控制此範圍：

- 如果您的 IP 位址範圍已向支援 RDAP 的網際網路註冊機構 (例如 ARIN、RIPE 和 APNIC) 註冊，您可以透過此頁面上的流程，使用 X.509 憑證驗證網域的控制。憑證必須僅在佈建流程期間內有效。您可以在佈建完成後，從 RIR 記錄中移除憑證。
- 無論您的網際網路註冊是否支援 RDAP，您都可以使用 Amazon VPC IPAM 來驗證 DNS TXT 記錄對網域的控制。該流程記錄在「Amazon VPC IPAM 使用者指南」 中的教學課程：將您的 IP 位址用於 IPAM。

如需詳細資訊，請參閱 AWS Online Tech talk Deep Dive on Bring Your Own IP。

目錄

- BYOIP 定義
- 需求與配額
- 區域可用性
- Local Zone 可用性
- Amazon EC2 中 BYOIP 的先決條件
- 加入您的位址範圍以用於 Amazon EC2
- 在 Amazon EC2 中使用您的 BYOIP 位址範圍

BYOIP 定義

- X.509 自我簽署的憑證 - 最常用來加密和驗證網路中資料的憑證標準。這是 用來 AWS 驗證從 RDAP 記錄控制 IP 空間的憑證。若要取得有關 X.509 憑證的更多資訊，請參閱 RFC 3280。

- 自治系統編號 (ASN) – 這是一個全域唯一識別符，可定義由一個或多個網路業者執行的 IP 字首群組，而這些網路業者維護單一且明確定義的路由政策。
- 區域網際網路註冊管理機構 (RIR) – 管理全球某個地區內 IP 地址和 ASN 之配置與註冊的組織。
- 註冊資料訪問協定 (RDAP)：用來查詢 RIR 目前註冊資料的唯讀協定。查詢的 RIR 資料庫中的項目稱為「RDAP 記錄」。客戶必須透過 RIR 提供的機制更新特定記錄類型。會查詢這些記錄 AWS ，以驗證 RIR 中地址空間的控制。
- 路由來源授權 (ROA) - 由 RIR 建立的物件，可供客戶於特定自治系統中驗證 IP 公告。如需概觀，請參閱 ARIN 網站上的路由來源授權 (ROAS)。
- 本地網際網路註冊機構 (LIR) – 從 RIR 為其客戶分配 IP 地址區塊的網際網路服務供應商等組織。

需求與配額

- 必須在區域網際網路註冊管理機構 (RIR) 註冊位址範圍。有關地理區域的任何政策，請參閱您的 RIR。BYOIP 目前支援在美洲網際網路號碼註冊管理機構 (ARIN) 或歐洲 IP 網路資源協調中心 (RIPE) 或亞太區域資訊中心 (APNIC) 進行註冊。它必須以商業或機構實體註冊，而無法以個人身分註冊。
- 您可以使用的最特定 IPv4 位址範圍是 /24。
- 您可以帶來的最具體 IPv6 地址範圍是 /48 代表可公開公告CIDRs，以及 /60 代表不可公開公告CIDRs。
- 對於不可公開公告的 CIDR 範圍，不需要 ROA，但仍需要更新 RDAP 記錄。
- 您可以將每個地址範圍一次帶到一個 AWS 區域。
- 您可以將每個 AWS 區域總共五個 BYOIP IPv4 和 IPv6 地址範圍帶到 AWS 您的帳戶。您無法使用 Service Quotas 主控台調整 BYOIP CIDRs 的配額，但您可以聯絡 AWS Support Center 請求增加配額，如 中的AWS 服務配額中所述AWS 一般參考。
- 除非您使用 Amazon VPC IP Address Manager (IPAM) 並將 IPAM 與 AWS Organizations 整合，AWS RAM 否則無法使用 與其他帳戶共用您的 IP 地址範圍。如需詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的將 IPAM 與 AWS Organizations 整合。
- IP 地址範圍中的地址都必須有良好的歷史記錄。我們會調查 IP 地址範圍的評價，如果 IP 地址範圍包含的 IP 地址評價不佳或與惡意行為有關，我們保留拒絕該範圍的權利。
- 舊版地址空間 (即在區域網際網路註冊管理機構 (RIR) 系統形成之前，由網際網路號碼分配局 (IANA) 的中央登錄檔所分配的 IPv4 地址空間) 仍然需要對應的 ROA 物件。
- 對於 LIR，其通常使用手動程序來更新其記錄。這可能需要幾天時間才能部署，依 LIR 而定。

- 大型 CIDR 區塊需要單一的 ROA 物件和 RDAP 記錄。您可以使用單一物件和記錄，從該範圍將多個較小的 CIDR 區塊帶到 AWS多個 AWS 區域。
- Wavelength 區域或 上的 不支援 BYOIP AWS Outposts。
- 不要在 RADb 或任何其他 LIR 中對 BYOIP 進行任何手動變更。BYOIP 會自動更新 RADb。任何包含 BYOIP ASN 的手動變更將會導致 BYOIP 佈建操作失敗。
- 將 IPv4 地址範圍帶入目的地之後 AWS，您可以使用範圍內的所有 IP 地址，包括第一個地址 （網路地址） 和最後一個地址 （廣播地址）。

區域可用性

BYOIP 功能目前適用於所有商業 AWS 區域 (中國地區除外)。

Local Zone 可用性

Local Zone 是 AWS 區域的延伸，其地理位置接近您的使用者。Local Zones 會分組為「網路邊界群組」。在 中 AWS，網路邊界群組是可用區域 (AZs)、本機區域或 Wavelength 區域的集合，從中 AWS 公告公有 IP 地址。本機區域的網路邊界群組可能與 AWS 區域中AZs 不同，以確保 AWS 網路與存取這些區域中資源的客戶之間的最低延遲或實體距離。

您可以使用 `--network-border-group` 選項，在下列 Local Zone 網路邊界群組中佈建 BYOIPv4 地址範圍，並公告這些地址範圍：

- af-south-1-los-1
- ap-northeast-1-tpe-1
- ap-south-1-ccu-1
- ap-south-1-del-1
- ap-southeast-1-bkk-1
- ap-southeast-1-mnl-1
- ap-southeast-2-akl-1
- ap-southeast-2-per-1
- eu-central-1-ham-1
- eu-central-1-ist-1
- eu-central-1-waw-1
- eu-north-1-cph-1
- eu-north-1-hel-1

- me-south-1-mct-1
- us-east-1-atl-2
- us-east-1-bos-1
- us-east-1-bue-1
- us-east-1-chi-2
- us-east-1-dfw-2
- us-east-1-iah-2
- us-east-1-lim-1
- us-east-1-mci-1
- us-east-1-mia-2
- us-east-1-msp-1
- us-east-1-nyc-1
- us-east-1-nyc-2
- us-east-1-phl-1
- us-east-1-qro-1
- us-east-1-scl-1
- us-west-2-den-1
- us-west-2-hnl-1
- us-west-2-las-1
- us-west-2-lax-1
- us-west-2-pdx-1
- us-west-2-phx-2
- us-west-2-sea-1

如果您已啟用 Local Zones (請參閱啟用 Local Zone)，您可以在佈建和公告 BYOIPv4 CIDR 時為 Local Zones 選擇網路邊界群組。仔細選擇網路邊界群組，因為 EIP 與其相關聯的 AWS 資源必須位於相同的網路邊界群組中。

> **Note**
>
> 您目前無法在 Local Zones 中佈建或公告 BYOIPv6 地址範圍，但 除外`eu-central-1-ist-1`，它同時支援 BYOIPv4 和 BYOIPv6。

在 eu-central-1-ist-1 Local Zone 中，公告和撤回是非同步操作，最多可能需要五個工作天才能完成。當您公告地址範圍時，其狀態會從 變更為 provisioned pending-advertising，然後在advertised操作完成後變更為 。當您將其撤回時，狀態會從 變更為 advertised pending-withdrawal，然後變更回 provisioned。您可以透過 AWS 主控台或 CLI 監控這些操作的進度。

如果您使用自己的 ASN (BYOASN)，您可以在公告地址範圍時指定它。您無法在公告後變更與 CIDR 相關聯的 ASN。若要變更 ASN，請撤銷 CIDR 並重新公告。

> **Important**
>
> 您無法取消待定的廣告或撤銷。您必須等待操作完成，才能啟動下一個操作。

Amazon EC2 中 BYOIP 的先決條件

BYOIP 的加入過程分為兩個階段，您必須執行三個步驟。這些步驟與下圖中說明的步驟相對應。我們在本文件中包含手動步驟，但您的 RIR 可能會提供受管服務來協助您完成這些步驟。

> **Tip**
>
> 本節中的任務需要 Linux 終端，而且可以使用 Linux、AWS CloudShell 或 Windows Subsystem for Linux 執行。

目錄

- 概要
- 建立私密金鑰並產生 X.509 憑證
- 將 X.509 憑證上傳至 RIR 中的 RDAP 記錄
- 在 RIR 中建立一個 ROA 物件

概要

準備階段

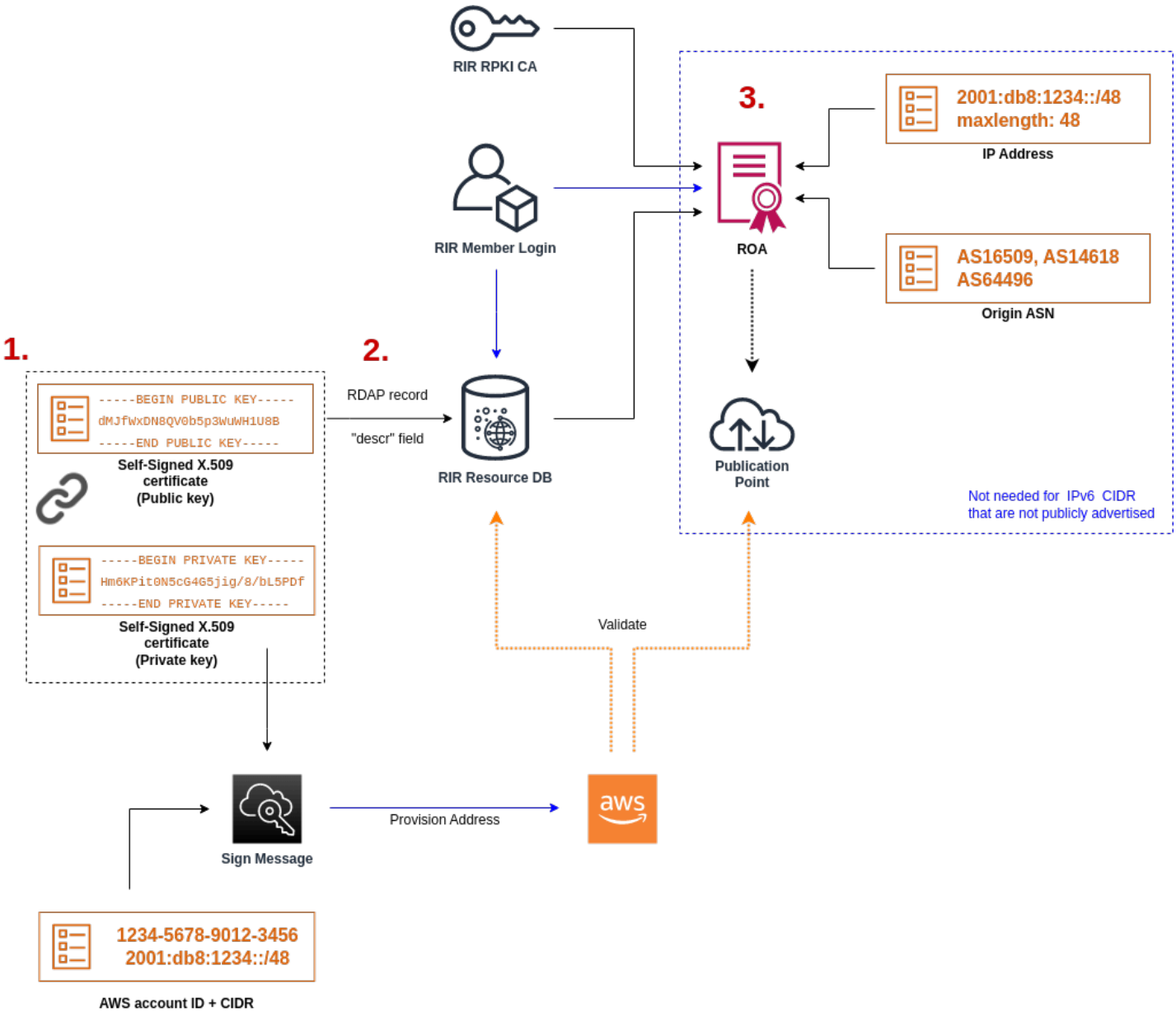
[1] 建立私密金鑰，然後使用其來產生自我簽署的 X.509 憑證進行身分驗證。此憑證僅在佈建階段使用。您可以在下列佈建階段完成後，從 RIR 記錄中移除憑證。

RIR 組態階段

[2] 上傳自我簽署的憑證至您 RDAP 記錄註解。

[3] 在 RIR 中建立一個 ROA 物件。ROA 定義了所需的地址範圍、允許公告地址範圍的自治系統編號 (ASN) 以及向您 RIR 的資源公有金鑰基礎設施 (RPKI) 註冊的到期日期。

> **Note**
>
> 不可公開公告的 IPv6 地址空間不需要 ROA。



如要引入多個不連續的地址範圍，您必須針對每個地址範圍重複此程序。不過，如果將連續區塊分割到數個不同的 AWS 區域，則不需要重複準備和 RIR 組態步驟。

引入地址範圍不會影響您先前引入的任何地址範圍。

建立私密金鑰並產生 X.509 憑證

使用以下程序建立自簽 X.509 憑證，並將其加入到您 RIR 的 RDAP 記錄中。此金鑰對可用來向 RIR 驗證地址範圍。openssl 命令需要 OpenSSL 1.0.2 版或更新版本。

複製下列命令，並只取代預留位置值 (以彩色斜體文字表示)。

此程序會遵循加密私有 RSA 金鑰並需要密碼短語才能進行存取的最佳實務。

1. 產生 RSA 2048 位元私密金鑰，如下所示。

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out
 private-key.pem
```

-aes256 參數會指定用來加密私有金鑰的演算法。此命令會傳回下列輸出，包括設定密碼短語的提示：

```
......+++
.+++
Enter PEM pass phrase: xxxxxxx
Verifying - Enter PEM pass phrase: xxxxxxx
```

您可以使用下列命令來檢查金鑰：

```
$ openssl pkey -in private-key.pem -text
```

這會傳回密碼短語提示和金鑰內容，應類似於以下內容：

```
Enter pass phrase for private-key.pem: xxxxxxx
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDFBXHRI4HVKAhh
3seiciooizCRTbJe1+YsxNTja4XyKypVGIFWDGhZs44FCHlPOOSVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zHOSEpNmY2fMxISBxewlxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv5l0tJKk+a91K4+tO3UdDR7Sno5WXExfsBrW3
g1ydo3TBsx8i5/YiVOcNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRsOlAhJ
DnZPNeweboo+K3Q3lwbgbmOKD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGrMSn2
```

```
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxfox2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPli1SXnpzvkdU4Hyco4zgbhXFsE
RNYjYfOGzTPwdBLpNMB6k3Tp4RHse6dNrlH0jDhpioL8cQEBdBJyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULMlWiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjGO59W
jfZjzTX5pQtVVH68ruciH88DTZCwjCkjBhxg+OIkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjwlC/3jxp8zJy6P8o
JQKv7TdvMwUj4VSWOHZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1NrONnSz5QsndQ04Z6ihAQlPmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDdc23yNmT3+Jyptf0fKjEvONK+xwUKzi9c
L/OzBq5yOIC1Pz2T85gOe1i8kwZws+xlpG6uBT6lmIJELd0k59FyupNu4dPvX5SD
6GGqdx4jk9KvI74usGeOBohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGgO28jjr
LpaHNZ/MXQKBgQDfLNcnS0LzpsS2aK0tzyZU8SMyqVHOGMxj7quhneBq2T6FbiLD
T9TVlYaGNZ0j71vQaLI19qOubWymbautH0Op5KV8owdf4+bf1/NJaPIOzhDUSIjD
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycaAW9lItu8aBrMndnQKBgQDb
nNp/JyRwqjOrNljk7DHEs+SD39kHQzzCfqd+dnTPv2sc06+cpym3yulQcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWMwHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXbWdIhYyI3QKBgD+F/6wcZ85QW8nAUykA
3WrSIx/3cwDGdm4NRGct8ZOZjTHjiy9ojMOD1L7iMhRQ/3k3hUsin5LDMp/ryWGG
x4uIaLat40kiC7T4I66DM7P59euqdz3w0PD+VU+h7GSivvsFDdySUt7bNK0AUVLh
dMJfWxDN8QV0b5p3WuWH1U8B
-----END PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
    00:c5:05:71:d1:23:81:d5:28:08:61:de:c7:a2:72:
    2a:28:8b:30:91:4d:b2:5e:d7:e6:2c:c4:d4:e3:6b:
    85:f2:2b:2a:55:18:81:56:0c:68:59:b3:8e:05:08:
    79:4f:38:e4:95:27:e3:6a:3f:be:30:f7:aa:0c:ec:
    33:d2:df:1a:3d:91:a4:32:64:11:67:d9:81:29:d8:
    40:6a:e6:f7:f7:d3:b2:87:35:19:99:65:49:a4:9f:
    4c:c7:39:21:29:36:66:36:7c:cc:48:48:1c:5e:c2:
    5c:51:14:09:e2:c2:64:9d:ff:c4:c3:bc:72:4c:63:
    d1:6f:00:8b:d6:b9:3b:2f:e6:5d:2d:24:a9:3e:6b:
    dd:4a:e3:eb:4e:dd:47:43:47:b4:a7:a3:95:97:13:
    17:ec:06:b5:b7:83:5c:9d:a3:74:c1:b3:1f:22:e7:
    f6:22:54:e7:0d:02:9c:bb:81:ed:bf:16:2c:18:dd:
    a0:97:24:1e:ab:ea:7b:85:e8:7f:26:46:02:38:af:
    8b:e4:31:1b:0e:94:08:49:0e:76:4f:35:ec:1e:6e:
    8a:3e:2b:74:37:97:06:e0:6e:63:8a:0f:fc:fd:b2:
    f9:3c:37:ff:a1:51:30:6d:21:7d:1f:46:d6:c6:f8:
    f2:c8:c3:7c:56:44:71:ab:31:29:f6:07:3b:0f:56:
    e0:cb
publicExponent: 65537 (0x10001)
privateExponent:
    0a:22:54:8f:68:5f:26:42:af:e3:b0:dc:dd:eb:37:
```

```
    65:ec:7a:ec:0e:6e:0d:58:d7:9b:17:e8:c7:65:e1:
    76:ea:67:7c:07:0d:a8:0a:6d:57:a7:d7:b7:44:8f:
    50:d6:e1:53:16:c1:28:d6:ec:86:82:46:b9:f1:70:
    5c:f9:62:d5:25:e7:a7:3b:e4:75:4e:07:c9:ca:38:
    ce:06:e1:5c:5b:04:44:d6:23:61:f3:86:cd:33:f0:
    74:12:e9:34:c0:7a:93:74:e9:e1:11:ec:7b:a7:4d:
    ae:51:f4:8c:38:69:8a:82:fc:71:01:01:74:12:72:
    54:5e:57:d3:0c:a6:11:b9:95:98:2d:23:80:7f:cc:
    c6:c0:40:3d:65:ba:64:a8:9c:83:d5:0b:32:55:a2:
    01:9d:cc:44:06:4f:8c:71:e0:a5:89:00:02:c5:16:
    28:06:c2:07:05:50:71:58:c6:3b:9f:56:8d:f6:63:
    cd:35:f9:a5:0b:55:54:7e:bc:ae:e7:22:1f:cf:03:
    4d:90:b0:8c:29:23:06:1c:60:f8:e2:24:24:12:c4:
    e7:09:21:f3:68:c8:1d:28:af:67:ad:df:97:02:f0:
    cf:e1:34:f8:78:44:2d:26:49:ae:7d:8c:63:a2:71:
    9a:29:37:a8:d3:54:38:5f:d9:fb:79:ac:76:3d:a5:
    b9
prime1:
    00:e3:c2:50:bf:de:3c:69:f3:32:72:e8:ff:28:25:
    02:af:ed:37:6f:33:05:23:e1:54:96:38:76:41:1c:
    bb:f8:7a:f2:5a:6a:26:b4:b9:08:c8:a3:55:03:6b:
    c0:18:8a:da:a1:5f:53:66:08:27:a1:18:7f:32:b9:
    78:ff:bf:a5:77:0b:33:0a:0e:49:91:af:53:6b:38:
    d9:d2:cf:94:2c:9d:d4:34:e1:9e:a2:84:04:25:3e:
    62:7d:ea:0e:30:2a:d8:28:0b:b0:18:a7:23:f4:83:
    56:be:e3:fb:23:6f:5f:a8:dd:84:08:e2:90:ff:17:
    bd:5c:fa:a6:b3:b4:7e:cf:47
prime2:
    00:dd:73:6d:f2:36:64:f7:f8:9c:a9:b5:fd:1f:2a:
    31:2f:38:d2:be:c7:05:0a:ce:2f:5c:2f:f3:b3:06:
    ae:72:38:80:b5:3f:3d:93:f3:98:0e:7b:58:bc:93:
    06:70:b3:ec:65:a4:6e:ae:05:3e:a5:98:82:44:2d:
    dd:24:e7:d1:72:ba:93:6e:e1:d3:ef:5f:94:83:e8:
    61:aa:77:1e:23:93:d2:af:23:be:2e:b0:67:8e:06:
    88:66:17:4a:61:4c:79:2b:58:a0:71:5e:2c:93:d2:
    84:bc:ce:39:c9:94:49:fc:ca:c2:29:1a:03:b6:f2:
    38:eb:2e:96:87:35:9f:cc:5d
exponent1:
    00:df:2c:d7:27:4b:42:f3:a6:c4:b6:68:ad:2d:cf:
    26:54:f1:23:32:a9:51:ce:18:cc:63:ee:ab:a1:9d:
    e0:6a:d9:3e:85:6e:22:c3:4f:d4:d5:95:86:86:35:
    9d:23:ef:5b:d0:68:b2:35:f6:a3:ae:6d:6c:a6:6d:
    ab:ad:1f:43:a9:e4:a5:7c:a3:07:5f:e3:e6:df:d7:
    f3:49:68:f2:0e:ce:10:d4:48:88:c3:42:8d:35:59:
```

```
    6d:f5:67:d5:c3:49:18:4a:15:39:d6:ce:60:a3:05:
    d7:88:71:a8:f2:cd:fd:74:60:ab:32:71:a0:16:f6:
    52:2d:bb:c6:81:ac:c9:dd:9d
exponent2:
    00:db:9c:da:7f:27:24:70:aa:33:ab:36:58:e4:ec:
    31:c4:b3:e4:83:df:d9:07:43:3c:c2:7e:a7:7e:76:
    74:cf:bf:6b:1c:d3:af:9c:a7:29:b7:ca:e9:50:71:
    ba:24:50:ba:72:7e:64:68:dd:b8:a7:fe:9b:c9:43:
    76:99:5f:f0:5d:87:dc:28:4d:7a:a1:5c:37:6b:ad:
    2c:16:22:75:58:31:03:f2:3e:4f:1f:fc:3f:66:20:
    e2:69:e4:55:16:33:01:c3:53:ec:21:21:94:b1:b0:
    47:84:fa:3b:62:c6:55:ad:85:e2:91:62:44:26:cd:
    06:57:6d:67:48:85:8c:88:dd
coefficient:
    3f:85:ff:ac:1c:67:ce:50:5b:c9:c0:53:29:00:dd:
    6a:d2:23:1f:f7:73:00:c6:76:6e:0d:44:67:2d:f1:
    93:99:8d:31:e3:8b:2f:68:8c:c3:83:d4:be:e2:32:
    14:50:ff:79:37:85:4b:22:9f:92:c3:32:9f:eb:c9:
    61:86:c7:8b:88:68:b6:ad:e3:49:22:0b:b4:f8:23:
    ae:83:33:b3:f9:f5:eb:aa:77:3d:f0:d0:f0:fe:55:
    4f:a1:ec:64:a2:be:fb:05:0d:dc:92:52:de:db:34:
    ad:00:51:52:e1:74:c2:5f:5b:10:cd:f1:05:74:6f:
    9a:77:5a:e5:87:d5:4f:01
```

不使用私有金鑰時，請將它保管在安全位置。

2. 使用在上一個步驟中建立的私密金鑰產生 X.509 憑證。在此範例中，憑證會在 365 天後過期；在這段時間之後，就無法再信任此憑證。務必設定適當的過期期限。憑證必須僅在佈建流程期間內有效。您可以在佈建完成後，從 RIR 記錄中移除憑證。`tr -d "\n"` 命令從輸出中刪除換行符號字元 (換行符)。出現提示時，您需要提供 Common Name (通用名稱)，但其他欄位可以保留空白。

```
$ openssl req -new -x509 -key private-key.pem -days 365 | tr -d "\n" >
 certificate.pem
```

此輸出結果類似以下內容：

```
Enter pass phrase for private-key.pem: xxxxxxx
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:example.com
Email Address []:
```

> **Note**
>
> AWS 佈建不需要通用名稱。其可為任何內部或公有網域名稱。

您可以使用下列命令來檢查憑證：

```
$ cat certificate.pem
```

輸出應該是一個 PEM 編碼的長字串，沒有換行符，前面為 `-----BEGIN CERTIFICATE-----`，後面為 `-----END CERTIFICATE-----`。

將 X.509 憑證上傳至 RIR 中的 RDAP 記錄

將您先前建立的憑證新增到 RIR 的 RDAP 記錄。請務必包含 `-----BEGIN CERTIFICATE-----` 和 `-----END CERTIFICATE-----` 字串之前和之後的編碼部分。所有這些內容都必須在單獨的長線上。更新 RDAP 的程序取決於您的 RIR：

- 若是 ARIN，請使用 客戶經理入口網站，在代表您位址範圍的「網路資訊」物件的「Public Comments」區段中新增憑證。請勿將其新增至您組織的評論區段。
- 若是 RIPE，請將憑證作為新的「descr」欄位新增至代表您位址範圍的「inetnum」或「inet6num」物件。您通常可以在 RIPE 資料庫入口網站 的「我的資源」區段找到這些內容。請勿將其新增至組織的註解區段，或上述物件的「備註」欄位。
- 對於 APNIC，透過編輯 "inetnum" 或 "inet6num" 記錄上的備註來新增憑證。

您可以在下列佈建階段完成後，從 RIR 記錄中移除憑證。

在 RIR 中建立一個 ROA 物件

建立 ROA 物件來授權 Amazon ASN 16509 和 14618 以宣傳您的地址範圍，以及目前獲授權宣傳地址範圍的 ASN。對於 AWS GovCloud (US) Regions，僅授權 ASN 8987。對於 AWS 歐洲主權雲端，請授權 ASNs 16509 和 214101。您必須將長度上限設定為您要使用的 CIDR 大小。您可以使用的最特定 IPv4 字首為 /24。針對可公開公告的 CIDR，您可以使用的最特定 IPv6 地址範圍是 /48，而針對不可公開公告的 CIDR，則是 /60。

Important

如果您要為 Amazon VPC IP Address Manager (IPAM) 建立 ROA 物件，則當您建立 ROA 時，您必須針對 IPv4 CIDR 將 IP 地址字首的長度上限設定為 /24。若要將 IPv6 CIDR 新增至可公告集區，則 IP 地址字首的長度上限必須為 /48。這可確保您擁有將公有 IP 地址跨 AWS 區域分割的完整彈性。IPAM 會強制執行您設定的長度上限。如需有關 BYOIP 地址到 IPAM 的詳細資訊，請參閱《Amazon VPC IPAM 使用者指南》中的教學課程：BYOIP 地址 CIDR 到 IPAM。

可能最多需要 24 小時 Amazon 才能使用 ROA。如需詳細資訊，請參閱 RIR：

- ARIN - ROA Requests
- RIPE - Managing ROAs
- APNIC - 路由管理

當您將廣告從現場部署工作負載遷移到 時 AWS，您必須先為現有的 ASN 建立 ROA，才能為 Amazon 的 ASNs 建立 ROAs。否則，可能會對現有路由和廣告造成影響。

Important

若要讓 Amazon 公告並繼續公告您的 IP 地址範圍，您使用 Amazon ASN 的 ROA 必須符合上述準則。如果您的 ROA 無效或不符合上述準則，Amazon 保留停止公告您 IP 位址範圍的權利。

Note

不可公開公告的 IPv6 地址空間不需要此步驟。

加入您的位址範圍以用於 Amazon EC2

依據您的需求，BYOIP 的加入程序包括下列任務：

任務

- 在 中佈建可公開公告的地址範圍 AWS
- 佈建不可公開公告的 IPv6 地址範圍
- 透過 公告地址範圍 AWS
- 解除佈建地址範圍
- 驗證您的 BYOIP

在 中佈建可公開公告的地址範圍 AWS

當您佈建地址範圍以搭配 使用時 AWS，即表示您確認控制地址範圍，並授權 Amazon 公告該地址範圍。我們也會透過簽署的授權訊息驗證您確實控制該地址範圍。此訊息是以您使用 X.509 憑證更新 RDAP 記錄時使用的自我簽署 X.509 金鑰對簽署。 AWS 需要密碼編譯簽署的授權訊息，其會呈現給 RIR。RIR 會根據新增至 RDAP 的憑證來驗證簽章，並根據 ROA 檢查授權詳細資訊。

佈建地址範圍

1. 編寫訊息

   撰寫純文字授權訊息。訊息格式如下所示，其中的日期是訊息的過期日期：

   ```
   1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
   ```

   用您自己的值取代帳戶號碼、地址範圍和到期日，建立類似下列內容的訊息：

   ```
   text_message="1|aws|0123456789AB|198.51.100.0/24|20211231|SHA256|RSAPSS"
   ```

   這不能與具有類似特征的 ROA 訊息混淆。

2. 簽署訊息

   使用您先前建立的私有金鑰來簽署純文字訊息。此命令傳回的簽章是一個長字串，您需要在下一個步驟中使用它。

> **Important**
>
> 建議您複製並貼上此命令。除訊息內容外，請勿修改或取代任何值。

```
signed_message=$( echo -n $text_message | openssl dgst -sha256 -sigopt
 rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private-key.pem -keyform PEM
 | openssl base64 | tr -- '+=/' '-_~' | tr -d "\n")
```

3. 佈建地址

使用 AWS CLI provision-byoip-cidr 命令來佈建地址範圍。`--cidr-authorization-context` 選項會使用您先前建立的訊息和簽章字串。

> **Important**
>
> 如果 BYOIP AWS 範圍與設定 AWS CLI 不同，您必須指定應佈建的區域`Default region name`。

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message" --region us-east-1
```

佈建地址範圍是一種非同步操作；系統會立即傳回呼叫，但地址範圍要等到其狀態從 `pending-provision` 變更為 `provisioned` 時，才能開始使用。

4. 監控進度

雖然大多數佈建會在兩小時內完成，但公開公告範圍的佈建流程可能需要長達一週的時間才能完成。使用 describe-byoip-cidrs 命令監控進度，如下例所示：

```
aws ec2 describe-byoip-cidrs --max-results 5 --region us-east-1
```

如果在佈建期間發生問題且狀態變成 `failed-provision`，您必須在問題解決後再次執行 `provision-byoip-cidr` 命令。

佈建不可公開公告的 IPv6 地址範圍

根據預設，可向網際網路公開公告所佈建的地址範圍。您可以佈建不可公開公告的 IPv6 地址範圍。對於不可公開公告的路由，佈建程序通常會在幾分鐘內完成。當您將來自非公有地址範圍的 IPv6 CIDR 區塊與 VPC 產生關聯時，只能透過支援 IPv6 的混合連線選項存取該 IPv6，例如Direct Connect、AWS Site-to-Site VPN，或 Amazon VPC 傳輸閘道

佈建非公有位址範圍不需要 ROA。

> **Important**
>
> - 您只能指定在佈建期間可否公開公告地址範圍。您稍後無法變更可公告狀態。
> - Amazon VPC 不支援唯一本機地址 (ULA) CIDR。所有 VPC 都必須具有唯一的 IPv6 CIDR。兩個 VPC 不能具有相同的 IPv6 CIDR 範圍。

若要佈建不可公開公告的 IPv6 地址範圍，請使用下列 provision-byoip-cidr 命令。

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message" --no-publicly-advertisable --
region us-east-1
```

透過 公告地址範圍 AWS

佈建好地址範圍之後，即可將其公告。佈建好地址範圍之後，您必須公布。對於已經佈建好的地址範圍，您不能必須只公布一部分。

如果您佈建了不會公開公告的 IPv6 位址範圍，則不需要完成此步驟。

我們建議您先停止從其他位置公告地址範圍或範圍的任何部分，然後再公告地址範圍 AWS。如果您持續從其他位置公告您的 IP 位址範圍或其任何部分，我們就無法穩定地支援或疑難排解問題。具體來說，我們無法保證該位址範圍或其部分的流量可以進入我們的網路。

若要將停機時間降至最低，您可以將 AWS 資源設定為在公告地址集區之前使用地址集區中的地址，然後同時停止從目前位置公告該地址，並開始公告該地址集區 AWS。如需從您的地址集區配置彈性 IP 地址的詳細資訊，請參閱配置彈性 IP 位址。

限制

- 即使您每次都指定不同的地址範圍，最多仍只能每 10 秒執行一次 advertise-byoip-cidr 命令。

- 即使您每次都指定不同的地址範圍，最多仍只能每 10 秒執行一次 withdraw-byoip-cidr 命令。

若要公告地址範圍，請使用下列 advertise-byoip-cidr 命令。

```
aws ec2 advertise-byoip-cidr --cidr address-range --region us-east-1
```

若要停止公告地址範圍，請使用下列 withdraw-byoip-cidr 命令。

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

> **Note**
>
> 在 eu-central-1-ist-1 Local Zone 中，公告和撤回是非同步的。在操作完成之前，地址範圍會進入 pending-advertising或 pending-withdrawal 狀態。如需詳細資訊，請參閱本機區域可用性。

解除佈建地址範圍

若要停止搭配 使用您的地址範圍 AWS，請先釋出任何彈性 IP 地址，並取消任何仍從地址集區配置的 IPv6 CIDR 區塊的關聯。然後停止公告位址範圍，最後取消佈建位址範圍。

您無法取消佈建一部分的位址範圍。如果您想要使用更具體的地址範圍 AWS，請取消佈建整個地址範圍，並佈建更具體的地址範圍。

(IPv4) 若要釋出每個彈性 IP 地址，請使用下列 release-address 命令。

```
aws ec2 release-address --allocation-id eipalloc-12345678abcabcabc --region us-east-1
```

(IPv6) 若要解除 IPv6 CIDR 區塊的關聯，請使用下列 disassociate-vpc-cidr-block 命令。

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
 --region us-east-1
```

若要停止公告地址範圍，請使用下列 withdraw-byoip-cidr 命令。

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

若要解除佈建地址範圍，請使用下列 deprovision-byoip-cidr 命令。

```
aws ec2 deprovision-byoip-cidr --cidr address-range --region us-east-1
```

取消佈建位址範圍最多可能需要一天的時間。

驗證您的 BYOIP

1. 驗證自我簽署的 x.509 金鑰對

   透過 whois 命令驗證憑證是否已上傳且有效。

   若為 ARIN，請使用 `whois -h whois.arin.net r + 2001:0DB8:6172::/48` 來查詢您地址範圍的 RDAP 記錄。檢查命令輸出中 NetRange (網路範圍)的 Public Comments 部分。該憑證應新增至地址範圍的 Public Comments 部分中。

   您可使用下列命令來檢查包含憑證的 Public Comments：

   ```
   whois -h whois.arin.net r + 2001:0DB8:6172::/48 | grep Comments | grep BEGIN
   ```

   此會傳回含有金鑰內容的輸出，應類似於下列內容：

   ```
   Public Comments:
   -----BEGIN CERTIFICATE-----
   MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
   LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
   hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
   wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
   NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
   rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
   VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
   W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
   R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
   prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
   mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
   Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
   xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
   3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
   AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
   QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
   08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
   Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
   UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
   ```

```
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

若為 RIPE，請使用 `whois -r -h whois.ripe.net` *`2001:0DB8:7269::/48`* 來查詢您地址範圍的 RDAP 記錄。檢查命令輸出中 `inetnum` 物件 (網路範圍)的 `descr` 部分。該憑證應新增為地址範圍的新 `descr` 欄位。

您可使用下列命令來檢查包含憑證的 `descr`：

```
whois -r -h whois.ripe.net 2001:0DB8:7269::/48 | grep descr | grep BEGIN
```

此會傳回含有金鑰內容的輸出，應類似於下列內容：

```
descr:
-----BEGIN CERTIFICATE-----MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8
RDAHSP+I1TowDQYJKoZIhvcNAQELBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAg
MCEF1Y2tsYW5kMREwDwYDVQQHDAhBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIF
dlYiBTZXJ2aWNlczETMBEGA1UECwwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvbiBXZWIgU2VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW
8xEzARBgNVBAMMCkJZT0lQIERlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3
stsI5QesHVRwOaXUdprAnndaTugmDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wXlAqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdeIW48QaYjhMlUEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
GlHWkJsbhr0VEUyAGu1bwkgcdww3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2glHpGt0XGF7GbGTAfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0
XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6Y
Lhz521lfyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyL
xngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9
wySL507XQz76Uk5cFypBOzbnk35UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8
mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGC
vRDl/qdO/GIDJi77dmZWkh/ic90MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoN
PyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

若為 APNIC，請使用 `whois -h whois.apnic.net 2001:0DB8:6170::/48` 來查詢您 BYOIP 地址範圍的 RDAP 記錄。檢查命令輸出中 `inetnum` 物件 (網路範圍)的 `remarks` 部分。該憑證應新增為地址範圍的新 `remarks` 欄位。

您可使用下列命令來檢查包含憑證的 `remarks`：

```
whois -h whois.apnic.net 2001:0DB8:6170::/48 | grep remarks | grep BEGIN
```

此會傳回含有金鑰內容的輸出，應類似於下列內容：

```
remarks:
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

2. 驗證 ROA 物件的建立

使用 RIPEstat Data API，驗證是否成功建立 ROA 物件。請務必依據 Amazon ASN 16509 和 14618，及目前授權公布地址範圍的 ASN 來測試您的地址範圍。對於 AWS GovCloud (US) Regions，請針對 ASN 8987 進行測試。對於 AWS 歐洲主權雲端，請針對 ASNs16509 和 214101 進行測試。

您可使用以下命令，檢查具有您地址範圍不同 Amazon ASN 的 ROA 物件：

```
curl --location --request GET "https://stat.ripe.net/data/rpki-validation/data.json?
resource=ASN&prefix=CIDR
```

於此範例輸出中，Amazon ASN 16509 的回應結果為 "status": "valid"。這表示地址範圍的 ROA 物件已成功建立：

```
{
    "messages": [],
    "see_also": [],
    "version": "0.3",
    "data_call_name": "rpki-validation",
    "data_call_status": "supported",
    "cached": false,
    "data": {
        "validating_roas": [
            {
                "origin": "16509",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "valid"
            },
            {
                "origin": "14618",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            },
            {
                "origin": "64496",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            }
        ],
        "status": "valid",
        "validator": "routinator",
        "resource": "16509",
        "prefix": "2001:0DB8::/32"
    },
    "query_id": "20230224152430-81e6384e-21ba-4a86-852a-31850787105f",
```

```
    "process_time": 58,
    "server_id": "app116",
    "build_version": "live.2023.2.1.142",
    "status": "ok",
    "status_code": 200,
    "time": "2023-02-24T15:24:30.773654"
}
```

“unknown” 的狀態表示尚未建立地址範圍的 ROA 物件。“invalid_asn” 的狀態表示未成功建立地址範圍的 ROA 物件。

在 Amazon EC2 中使用您的 BYOIP 位址範圍

您可以檢視和使用已在帳戶中佈建的 IPv4 和 IPv6 地址範圍。如需詳細資訊，請參閱the section called “加入您的位址範圍”。

IPv4 位址範圍

您可以從 IPv4 地址集區建立彈性 IP 地址，並將其與 AWS 資源搭配使用，例如 EC2 執行個體、NAT 閘道和 Network Load Balancer。

若要檢視您在帳戶中佈建之 IPv4 位址集區的相關資訊，請使用下列 describe-public-ipv4-pools 命令。

```
aws ec2 describe-public-ipv4-pools --region us-east-1
```

若要從您的 IPv4 位址集區建立彈性 IP 地址，請使用 allocate-address 命令。您可以使用 `--public-ipv4-pool` 選項來指定 `describe-byoip-cidrs` 傳回的位址集區 ID。或者，您可以使用 `--address` 選項來指定您佈建的位址範圍。

IPv6 位址範圍

若要檢視您在帳戶中佈建之 IPv6 位址集區的相關資訊，請使用下列 describe-ipv6-pools 命令。

```
aws ec2 describe-ipv6-pools --region us-east-1
```

若要建立 VPC 並指定來自 IPv6 位址集區的 IPv6 CIDR，請使用下列 create-vpc 命令。若要讓 Amazon 從您的 IPv6 位址集區中選擇 IPv6 CIDR，請省略 `--ipv6-cidr-block` 選項。

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --ipv6-cidr-block ipv6-cidr --ipv6-
pool pool-id --region us-east-1
```

若要將來自您的 IPv6 位址集區的 IPv6 CIDR 區塊與 VPC 建立關聯，請使用下列 associate-vpc-cidr-block 命令。若要讓 Amazon 從您的 IPv6 位址集區中選擇 IPv6 CIDR，請省略 `--ipv6-cidr-block` 選項。

```
aws ec2 associate-vpc-cidr-block --vpc-id vpc-123456789abc123ab --ipv6-cidr-block ipv6-cidr --ipv6-pool pool-id --region us-east-1
```

若要檢視您的 VPC 和相關聯的 IPv6 位址集區資訊，請使用 describe-vpcs 命令。若要檢視來自特定 IPv6 位址集區的關聯 IPv6 CIDR 區塊的相關資訊，請使用下列取得 get-associated-ipv6-pool-cidrs 命令。

```
aws ec2 get-associated-ipv6-pool-cidrs --pool-id pool-id --region us-east-1
```

如果您取消 IPv6 CIDR 區塊與 VPC 的關聯，即會將它釋出回您的 IPv6 位址集區中。

彈性 IP 位址

彈性 IP 位址是針對動態雲端運算設計的靜態 IPv4 地址。彈性 IP 地址會配置給 AWS 您的帳戶，而且在您釋出之前都是您的地址。透過彈性 IP 位址，您可以快速地將地址重新映射至帳戶中的另一個執行個體，以遮罩執行個體或軟體的故障。或者，您也可以在您網域的 DNS 紀錄中指定彈性 IP 位址，以便您的網域指向您的執行個體。如需詳細資訊，請參閱您的網域註冊文件。

彈性 IP 位址是可從網際網路存取的公有 IPv4 地址。如果您需要連線至沒有公有 IPv4 地址的執行個體，則可將彈性 IP 位址與執行個體建立關聯，進而啟用與網際網路通訊的功能。

目錄

- 彈性 IP 位址定價
- 彈性 IP 位址基本概念
- 彈性 IP 位址配額
- 將彈性 IP 位址與執行個體建立關聯
- 在 之間傳輸彈性 IP 地址 AWS 帳戶
- 釋出彈性 IP 位址
- 在 Amazon EC2 上建立電子郵件的反向 DNS 記錄

彈性 IP 位址定價

無論是處於使用中 (配置給資源，例如 EC2 執行個體) 還是閒置狀態 (已在帳戶中建立但未配置)，所有彈性 IP 位址都會產生費用。

AWS 收取所有公有 IPv4 地址的費用，包括與執行中執行個體相關聯的公有 IPv4 地址和彈性 IP 地址。如需詳細資訊，請參閱 Amazon VPC 定價頁面中的公有 IPv4 地址。

彈性 IP 位址基本概念

以下是彈性 IP 位址的基本特性：

- 彈性 IP 位址是靜態的地址；不會隨著時間而變更。
- 彈性 IP 位址僅適用於特定區域，無法移至不同區域。
- 彈性 IP 位址來自 Amazon 的 IPv4 地址集區，或來自您已用於 AWS 帳戶帳戶的自訂 IPv4 地址集區。我們不支援 IPv6 的彈性 IP 位址。
- 若要使用彈性 IP 位址，您可以先將一個地址配置給帳戶，再將其與您的執行個體或網路介面建立關聯。
- 當您將彈性 IP 位址與執行個體建立關聯時，它也會與執行個體的主要網路介面相關聯。當您將彈性 IP 位址與連接到執行個體的網路介面建立關聯時，它也會與執行個體相關聯。
- 當您將彈性 IP 位址與執行個體或其主要網路介面建立關聯時，如果執行個體已有與其相關聯的公有 IPv4 地址，則該公有 IPv4 地址會釋出回 Amazon 的公有 IPv4 地址集區，而彈性 IP 位址會改為與該執行個體建立關聯。您無法重複使用先前與執行個體關聯的公有 IPv4 地址，也無法將該公有 IPv4 地址轉換為彈性 IP 位址。如需詳細資訊，請參閱公有 IPv4 地址。
- 您可以取消彈性 IP 位址與資源的關聯，並將其與不同的資源重新關聯。若要避免非預期的行為，請確保在變更前，已關閉與現有關聯中命名之資源的所有啟用連線。將彈性 IP 位址關聯至不同資源之後，您可以重新開啟與新關聯資源的連線。
- 在您明確將其釋放之前，取消關聯的彈性 IP 位址會維持配置給您的帳戶。您的帳戶中的所有彈性 IP 位址都會向您收取費用，無論它們是否與執行個體相關聯。如需詳細資訊，請參閱 Amazon VPC 定價頁面中的公有 IPv4 地址。
- 當您將彈性 IP 位址與先前具有公有 IPv4 地址的執行個體建立關聯時，執行個體的公有 DNS 主機名稱會變更以符合彈性 IP 位址。
- 我們會將公有 DNS 主機名稱解析為執行個體網路外部之執行個體的公有 IPv4 地址或彈性 IP 位址，以及解析為執行個體網路內執行個體的私有 IPv4 地址。
- 當您從已帶到 AWS 帳戶的 IP 地址集區配置彈性 IP 地址時，不會計入您的彈性 IP 地址限制。如需詳細資訊，請參閱彈性 IP 位址配額。

- 當您配置彈性 IP 位址時，可以將彈性 IP 位址與網路邊界群組建立關聯。這是我們公告 CIDR 區塊的位置。設定網路邊界群組會將 CIDR 區塊限制在此群組。如果您未指定網路邊界群組，我們會設定包含區域中所有可用區域的邊界群組 (例如，us-west-2)。
- 彈性 IP 位址只能用於特定網路邊界群組。

彈性 IP 位址配額

根據預設，每個區域都有五 (5) 個彈性 IP 地址的配額，因為公 AWS 帳戶 有 (IPv4) 網際網路地址是很少的公有資源。我們強烈建議您將彈性 IP 位址主要用在發生執行個體故障時將地址重新映射至另一個執行個體，並針對所有其他節點間通訊使用 DNS 主機名稱。

如果您認為自己的架構需要更多的彈性 IP 位址，您可以直接從 Service Quotas 主控台請求提高配額。若要請求增加配額，請選擇在帳戶層級請求增加。如需詳細資訊，請參閱Amazon EC2 服務配額。

將彈性 IP 位址與執行個體建立關聯

配置彈性 IP 地址之後，您可以將其與 AWS 資源建立關聯，例如 EC2 執行個體、NAT 閘道或 Network Load Balancer。若要稍後將彈性 IP 地址與不同的 AWS 資源建立關聯，您可以取消其與其目前資源的關聯，然後將其與新資源建立關聯。

任務

- 配置彈性 IP 位址
- 建立彈性 IP 位址的關聯
- 取消彈性 IP 位址的關聯

配置彈性 IP 位址

配置用於某區域的彈性 IP 位址。無論是處於使用中 (與資源關聯，例如 EC2 執行個體) 還是閒置狀態 (已在帳戶中建立但未關聯)，所有彈性 IP 位址都會產生費用。

Console

配置彈性 IP 地址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，於 Network & Security (網路與安全) 中，選擇 Elastic IPs (彈性 IP)。
3. 選擇 Allocate Elastic IP address (配置彈性 IP 位址)。

4. (選用) 配置彈性 IP 地址 (EIP) 時，您可以選擇要配置 EIP 的網路邊界群組。網路邊界群組是可用區域 (AZs)、本機區域或 Wavelength 區域的集合，可從中 AWS 公告公有 IP 地址。Local Zones 和 Wavelength Zones 的網路邊界群組可能與區域中AZs 不同，以確保 AWS 網路和存取這些區域中資源的客戶之間的最低延遲或實體距離。

   > **Important**
   >
   > 您必須在與 EIP 相關聯的 AWS 資源相同的網路邊界群組中配置 EIP。一個網路邊界群組中的 EIP 只能在該網路邊界群組中的區域中進行公告，而不能在其他網路邊界群組所代表的任何其他區域中進行公告。

   如果您已啟用 Local Zones 或 Wavelength Zones (如需詳細資訊，請參閱啟用 Local Zone 或啟用 Wavelength Zones)，您可以為 AZ、Local Zones 或 Wavelength Zones 選擇網路邊界群組。請謹慎選擇網路邊界群組，因為 EIP 及其關聯的 AWS 資源必須位於相同的網路邊界群組中。您可以使用 EC2 主控台檢視可用區域、Local Zones 或 Wavelength Zones 所在的網路邊界群組。一般而言，區域中的所有可用區域屬於相同的網路邊界群組，而 Local Zones 或 Wavelength Zones 則屬於其各自的網路邊界群組。

   如果您沒有啟用 Local Zones 或 Wavelength Zones，則當您配置 EIP 時，代表該區域所有 AZ 的網路邊界群組 (例如 `us-west-2`) 會為您預先定義，且您無法予以變更。這表示您配置給此網路邊界群組的 EIP 會在您所在區域的所有 AZ 中進行公告。

5. 對於公有 IPv4 地址集區，請選擇下列其中一項：

   - Amazon 的 IPv4 地址集區—若您要從 Amazon IPv4 地址集區配置一個 IPv4 地址。
   - 您帶入 AWS 帳戶的公有 IPv4 位址 - 如果您想要從您的 AWS 帳戶中的 IP 位址集區中指派不連續 (非連續) 的公用 IPv4 位址。如果您沒有任何 IP 位址集區，則會停用此選項。如需將您自己的 IP 地址範圍帶入 AWS 帳戶的詳細資訊，請參閱 將自有 IP 位址 (BYOIP) 用於 Amazon EC2。
   - 客戶擁有的 IPv4 地址集區 - 如果您想要從內部部署網路建立的集區配置 IPv4 地址，以便與 AWS Outpost 搭配使用。如果您沒有 AWS Outpost，則會停用此選項。
   - 使用 IPAM IPv4 集區配置：如果您想要從 IPAM 集區中的連續公有 IPv4 區塊配置循序彈性 IP 位址。配置循序彈性 IP 位址可大幅降低安全存取控制清單的管理開銷，並簡化企業在 上擴展的 IP 位址配置和追蹤 AWS。如需詳細資訊，請參閱「Amazon VPC IPAM 使用者指南」中的從 IPAM 集區配置循序彈性 IP 位址。

6. (選用) 若要新增標籤，請選擇新增標籤，然後輸入標籤金鑰和標籤值。

AWS CLI

配置彈性 IP 位址

使用 allocate-address AWS CLI 命令。

在以下範圍中，Amazon EC2 會從 Amazon 的地址集區中選取地址。

```
aws ec2 allocate-address
```

在下列範例中，Amazon EC2 會從您使用 AWS BYOIP 帶到 的指定集區中選取地址。

```
aws ec2 allocate-address \
    --public-ipv4-pool ipv4pool-ec2-012345abcdef67890
```

下列範例指定來自指定 IPv4 IPAM 集區的地址。

```
aws ec2 allocate-address \
    --ipam-pool-id ipam-pool-1234567890abcdef0 \
    --address 192.0.2.0
```

PowerShell

配置彈性 IP 位址

使用 New-EC2Address cmdlet。

在以下範圍中，Amazon EC2 會從 Amazon 的地址集區中選取地址。

```
New-EC2Address
```

在下列範例中，Amazon EC2 會從您使用 AWS BYOIP 帶到 的指定集區中選取地址。

```
New-EC2Address `
    -PublicIpv4Pool ipv4pool-ec2-012345abcdef67890
```

下列範例指定來自指定 IPv4 IPAM 集區的地址。

```
New-EC2Address `
```

```
    -IpamPoolId ipam-pool-1234567890abcdef0 `
    -Address 192.0.2.0
```

建立彈性 IP 位址的關聯

如果您想要將彈性 IP 位址與執行個體建立關聯來啟用與網際網路通訊的功能，您也必須確定執行個體位於公有子網路中。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的使用網際網路閘道啟用網際網路存取。

Console

建立彈性 IP 位址與執行個體的關聯

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取要關聯的彈性 IP 位址並依序選取 Actions (動作)、Associate Elastic IP address (關聯彈性 IP 位址)。
4. 對於 Resource type (資源類型)，選擇 Instance (執行個體)。
5. 例如，選擇要與彈性 IP 位址建立關聯的執行個體。您也可以輸入文字來搜尋特定執行個體。
6. (選用) 對於 Private IP address (私人 IP 位址)，指定要與彈性 IP 位址建立關聯的私人 IP 位址。
7. 選擇 Associate (關聯)。

建立彈性 IP 位址與網路介面的關聯。

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取要關聯的彈性 IP 位址並依序選取 Actions (動作)、Associate Elastic IP address (關聯彈性 IP 位址)。
4. 對於資源類型，選擇 Network interface (網路介面)。
5. 在 Network interface (網路介面) 中，選擇要與彈性 IP 位址建立關聯的網路介面。您也可以輸入文字來搜尋特定網路介面。
6. (選用) 對於 Private IP address (私人 IP 位址)，指定要與彈性 IP 位址建立關聯的私人 IP 位址。
7. 選擇 Associate (關聯)。

AWS CLI

建立彈性 IP 位址的關聯

使用 associate-address AWS CLI 命令。

```
aws ec2 associate-address \
    --instance-id i-0b263919b6498b123 \
    --allocation-id eipalloc-64d5890a
```

PowerShell

建立彈性 IP 位址的關聯

使用 Register-EC2Address cmdlet。

```
Register-EC2Address `
    -InstanceId i-0b263919b6498b123 `
    -AllocationId eipalloc-64d5890a
```

取消彈性 IP 位址的關聯

您可以隨時解除彈性 IP 位址與執行個體或網路介面的關聯。解除與彈性 IP 位址的關聯後，您可以將它與其他資源建立關聯。

Console

解除關聯並重新關聯彈性 IP 位址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取要解除彈性 IP 位址的關聯，請依序選取 Actions (動作)、Disassociate Elastic IP address (解除彈性 IP 位址的關聯)。
4. 選擇 Disassociate (取消關聯)。

AWS CLI

取消與彈性 IP 位址的關聯

使用 disassociate-address AWS CLI 命令。

```
aws ec2 disassociate-address --association-id eipassoc-12345678
```

PowerShell

取消與彈性 IP 位址的關聯

使用 Unregister-EC2Address cmdlet。

```
Unregister-EC2Address -AssociationId eipassoc-12345678
```

在 之間傳輸彈性 IP 地址 AWS 帳戶

您可以將彈性 IP 地址從一個轉移到 AWS 帳戶 另一個。此問題的解決方案如下：

- 災難復原 – 發生緊急事故時，使用彈性 IP 位址轉移可快速重新對應 IP，以處理公開網站的工作負載。
- 組織重組 – 快速將工作負載從一個工作負載移至 AWS 帳戶 另一個工作負載。地址轉移可避免需要等待安全群組和網路 ACLs 允許新的彈性 IP 位址。
- 集中式安全管理 – 使用集中式 AWS 安全帳戶來追蹤和轉移已經過安全合規審核的彈性 IP 地址。

定價

轉移彈性 IP 位址無需付費。

任務

- 啟用彈性 IP 位址轉移
- 接受轉移後的彈性 IP 位址
- 停用彈性 IP 位址轉移

啟用彈性 IP 位址轉移

本節旨在說明如何接受轉移後的彈性 IP 位址。請注意下列與啟用彈性 IP 位址轉移的相關限制：

- 您可以將彈性 IP 地址從任何 AWS 帳戶 （來源帳戶） 轉移到相同 AWS 區域 （轉移帳戶） AWS 帳戶 中的任何其他 。您無法將彈性 IP 位址轉移到不同的區域。

- 當您轉移彈性 IP 位址時， AWS 帳戶之間會發生兩步驟交握。來源帳戶開始轉移時，轉移帳戶有七天時間可以接受彈性 IP 位址轉移。在這七天內，來源帳戶可以檢視待處理的轉移 （例如，在 AWS 主控台中或使用 describe-address-transfers 命令）。七天後轉移將到期，屆時彈性 IP 位址的擁有權會回到來源帳戶。
- 接受的轉移會在接受轉移後 14 天內顯示給來源帳戶 （例如，在 AWS 主控台中或使用 describe-address-transfers 命令）。
- AWS 不會通知轉移帳戶有關待定的彈性 IP 地址轉移請求。來源帳戶的擁有者必須通知轉移帳戶的擁有者，有必須接受的彈性 IP 位址轉移的請求。
- 轉移完成時，會重設與要轉移之彈性 IP 位址相關的所有標籤。
- 您無法從公有 IPv4 地址集區中配置的彈性 IP 地址傳輸到 AWS 帳戶 – 通常稱為使用自有 IP (BYOIP) 地址集區。
- 您無法從 Amazon 提供的連續公有 IPv4 Amazon VPC IP 位址管理器 (IPAM) 集區傳輸配置的彈性 IP 位址。相反地，IPAM 可讓您透過將 IPAM 與 AWS Organizations 整合並使用 來跨 AWS 帳戶共用 IPAM 集區 AWS RAM。如需詳細資訊，請參閱「Amazon VPC IPAM 使用者指南」中的從 IPAM 集區配置循序彈性 IP 位址。
- 如果您嘗試轉移的彈性 IP 位址，具有與其相關聯的反向 DNS 記錄，您可以開始轉移程序，但轉移帳戶無法接受轉移，除非移除相關聯的 DNS 記錄。
- 如果您已啟用並設定 AWS Outposts，您可能已從客戶擁有的 IP 地址集區 (CoIP) 配置彈性 IP 地址。您無法轉移從 CoIP 配置的彈性 IP 位址。不過，您可以使用 與其他帳戶 AWS RAM 共用 CoIP。如需詳細資訊，請參閱AWS Outposts 使用者指南中的客戶擁有的 IP 位址。
- 您可使用 Amazon VPC IPAM 來追蹤彈性 IP 位址在 AWS Organizations組織中的帳戶之間轉移的情況。如需詳細資訊，請參閱 View IP address history (檢視 IP 位址歷程記錄)。如果彈性 IP 地址轉移到組織 AWS 帳戶 外部的 ，則彈性 IP 地址的 IPAM 稽核歷史記錄會遺失。

這些步驟必須由來源帳戶完成。

Console

啟用彈性 IP 位址轉移

1. 請確定您使用的是來源 AWS 帳戶。
2. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
4. 選取要啟用轉移的一或多個彈性 IP 位址，然後選擇 Actions (動作)、Enable transfer (啟用轉移)。

5. 如果要轉移多個彈性 IP 位址，您會看到 Transfer type (轉移類型) 選項。請選擇下列其中一個選項：
   - 如果您要將彈性 IP 地址轉移到單一帳戶，請選擇單一 AWS 帳戶。
   - 如果您要將彈性 IP 地址轉移到多個帳戶，請選擇多個 AWS 帳戶。
6. 在 Transfer account ID (轉移帳戶 ID) 底下，輸入您要將彈性 IP 位址轉移至哪個 AWS 帳戶 ID。
7. 在文字方塊中輸入 **enable**，以便確認轉移。
8. 選擇提交。
9. 若要接受轉移，請參閱接受轉移後的彈性 IP 位址。若要停用轉移，請參閱停用彈性 IP 位址轉移。

AWS CLI

啟用彈性 IP 位址轉移

使用 enable-address-transfer 命令。

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

PowerShell

啟用彈性 IP 位址轉移

使用 Enable-EC2AddressTransfer cmdlet。

```
Enable-EC2AddressTransfer `
    -AllocationId eipalloc-09ad461b0d03f6aaf `
    -TransferAccountId 123456789012
```

接受轉移後的彈性 IP 位址

本節旨在說明如何接受轉移後的彈性 IP 位址。

當您轉移彈性 IP 位址時， AWS 帳戶之間會發生兩步驟交握。來源帳戶開始轉移時，轉移帳戶有七天時間可以接受彈性 IP 位址轉移。在這七天內，來源帳戶可以檢視待處理的轉移 （例如，在 AWS 主控

台中或使用 describe-address-transfers 命令）。七天後轉移將到期，屆時彈性 IP 位址的擁有權會回到來源帳戶。

接受轉移時，請注意下列可能發生的異常情況，以及如何解決這些異常問題：

- AddressLimitExceeded：如果轉移帳戶超過彈性 IP 位址配額，則來源帳戶仍可啟用彈性 IP 位址轉移，但是如果轉移帳戶嘗試接受轉移，就會發生此異常情況。根據預設，所有 AWS 帳戶每個區域限制為 5 個彈性 IP 地址。如需提高限制的指示，請參閱彈性 IP 位址配額。
- InvalidTransfer.AddressCustomPtrSet：如果您或組織中的某個人已將您嘗試轉移的彈性 IP 位址設定成使用反向 DNS 查詢，則來源帳戶仍可啟用該彈性 IP 位址的轉移，但會在轉移帳戶嘗試接受轉移時發生此異常情況。若要解決這個問題，來源帳戶必須移除該彈性 IP 位址的 DNS 記錄。如需詳細資訊，請參閱在 Amazon EC2 上建立電子郵件的反向 DNS 記錄。
- InvalidTransfer.AddressAssociated：如果彈性 IP 位址與 ENI 或 EC2 執行個體有關，則來源帳戶仍可啟用該彈性 IP 位址的轉移，但是如果轉移帳戶嘗試接受轉移，就會發生此異常情況。若要解決這個問題，來源帳戶必須取消與該彈性 IP 位址的關聯。如需詳細資訊，請參閱取消彈性 IP 位址的關聯。

如有任何其他異常情況，請聯絡 支援。

這些步驟必須由轉移帳戶完成。

Console

接受彈性 IP 位址轉移

1. 確保您使用的是轉移帳戶。
2. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
4. 選擇 Actions (動作)、Accept transfer (接受轉移)。
5. 當您接受轉移時，與該彈性 IP 位址有關的所有標籤都不會隨著該彈性 IP 位址轉移。如果您要為接受的彈性 IP 位址定義 Name (名稱) 標籤，請選取 Create a tag with a key of 'Name' and a value that you specify (建立索引鍵為 'Name' 並具有您指定值的標籤)。
6. 輸入您要轉移的彈性 IP 位址。
7. 如果您要接受多個轉移的彈性 IP 位址，請選擇 Add address (新增位址)，以輸入其他彈性 IP 位址。
8. 選擇提交。

AWS CLI

若要接受彈性 IP 位址轉移

使用 accept-address-transfer 命令。

```
aws ec2 accept-address-transfer --address 100.21.184.216
```

PowerShell

若要接受彈性 IP 位址轉移

使用 Approve-EC2AddressTransfer cmdlet。

```
Approve-EC2AddressTransfer -Address 100.21.184.216
```

停用彈性 IP 位址轉移

本節旨在說明如何在啟用轉移後停用彈性 IP 轉移。

這些步驟必須由啟用轉移的來源帳戶完成。

Console

停用彈性 IP 位址轉移

1. 確保您使用的是來源 AWS 帳戶。
2. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
3. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
4. 在彈性 IP 的資源清單中，確保您已啟用顯示 Transfer status (轉移狀態) 欄位的屬性。
5. 選取 Transfer status (轉移狀態) 為 Pending (待定) 的一或多個彈性 IP 位址，然後選擇 Actions (動作)、Disable transfer (停用轉移)。
6. 在文字方塊中輸入 **disable**，以便進行確認。
7. 選擇提交。

AWS CLI

若要停用彈性 IP 位址轉移

使用 disable-address-transfer 命令。

```
aws ec2 disable-address-transfer --allocation-id eipalloc-09ad461b0d03f6aaf
```

PowerShell

若要停用彈性 IP 位址轉移

使用 Disable-EC2AddressTransfer cmdlet。

```
Disable-EC2AddressTransfer -AllocationId eipalloc-09ad461b0d03f6aaf
```

釋出彈性 IP 位址

如果您不再需要彈性 IP 位址，我們建議您將其釋出。要發行的彈性 IP 地址目前不得與 AWS 資源相關聯。

Console

釋出彈性 IP 位址

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取要釋出的彈性 IP 位址，然後依序選取 Actions (動作)、Release Elastic IP addresses (釋出彈性 IP 位址)。
4. 選擇 Release (釋出)。

AWS CLI

釋出彈性 IP 位址

使用 release-address AWS CLI 命令。

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

PowerShell

釋出彈性 IP 位址

使用 Remove-EC2Address cmdlet。

```
Remove-EC2Address -AllocationId eipalloc-64d5890a
```

如果您釋出您的彈性 IP 位址，您也許能夠予以復原。適用的規定如下：

- 如果彈性 IP 位址已配置給其他 AWS 帳戶，或復原之後會導致您超過彈性 IP 位址限制，則您無法復原此彈性 IP 位址。
- 您無法復原與彈性 IP 位址相關聯的標籤。

AWS CLI

復原彈性 IP 位址

使用 allocate-address 命令。

```
aws ec2 allocate-address \
    --domain vpc \
    --address 203.0.113.3
```

PowerShell

復原彈性 IP 位址

使用 New-EC2Address cmdlet。

```
New-EC2Address `
    -Address 203.0.113.3 `
    -Domain vpc `
    -Region us-east-1
```

在 Amazon EC2 上建立電子郵件的反向 DNS 記錄

如果您打算從 EC2 執行個體傳送電子郵件給第三方，我們建議您佈建一或多個彈性 IP 位址，並將靜態反向 DNS 紀錄指派給用於傳送電子郵件的彈性 IP 位址。這可協助您避免某些反垃圾郵件組織將您的電子郵件標記為垃圾郵件。 與 ISPs和網際網路反垃圾郵件組織 AWS 合作，以降低從這些地址傳送的電子郵件被標記為垃圾郵件的機會。

考量事項

- 在建立反向 DNS 記錄之前，您必須設定對應的轉送 DNS 記錄（記錄類型 A），以解析至您帳戶中的彈性 IP 地址。最佳實務是包含您要在轉送 DNS 記錄中建立反向 DNS 記錄的所有彈性 IP 地址。
- 如果反向 DNS 記錄與彈性 IP 位址相關聯，該彈性 IP 位址會鎖定到您的帳戶，而且在移除記錄之前都無法從您的帳戶釋放。
- 如果您聯絡 支援 為彈性 IP 地址設定反向 DNS，您可以移除反向 DNS，但無法釋出彈性 IP 地址，因為它已被 鎖定 支援。若要解除彈性 IP 位址的鎖定，請聯絡 AWS 支援。彈性 IP 位址解除鎖定後，您可以將其釋放。
- 您無法使用主控台或AWS GovCloud (US) Region建立反向 DNS 記錄。 AWS 必須為您指派靜態反向 DNS 記錄。開啟支援案例以移除反向 DNS 和電子郵件傳送限制。必須提供彈性 IP 位址和反向 DNS 記錄。

建立反向 DNS 記錄

您可以建立彈性 IP 位址的反向 DNS 紀錄，如下所示。

Console

若要建立反向 DNS 記錄

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取彈性 IP 位址，然後選取 Actions (動作)、Update reverse DNS (更新反向 DNS)。
4. 針對 Reverse DNS domain name (反向 DNS 網域名稱)，輸入網域名稱。
5. 輸入 **`update`** 以確認。
6. 選擇 Update (更新)。

AWS CLI

若要建立反向 DNS 記錄

使用 modify-address-attribute 命令。

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
```

```
    --domain-name example.com
```

以下為範例輸出。

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net.",
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

若要建立反向 DNS 記錄

使用 Edit-EC2AddressAttribute cmdlet。

```
Edit-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -DomainName 'example.com' |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

以下為範例輸出。

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.net.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.com.
```

移除反向 DNS 記錄

您可以從彈性 IP 位址移除反向 DNS 記錄，如下所示。

如果您收到下列錯誤，您可以向 提交移除電子郵件傳送限制的請求 支援 ，以尋求協助。

```
The address cannot be released because it is locked to your account.
```

Console

若要移除反向 DNS 記錄

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Elastic IPs (彈性 IP)。
3. 選取彈性 IP 位址，然後選取 Actions (動作)、Update reverse DNS (更新反向 DNS)。
4. 針對 Reverse DNS domain name (反向 DNS 網域名稱)，清除相應的網域名稱。
5. 輸入 **update** 以確認。
6. 選擇 Update (更新)。

AWS CLI

若要移除反向 DNS 記錄

使用 reset-address-attribute 命令。

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

以下為範例輸出。

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com.",
            "PtrRecordUpdate": {
                "Value": "example.net.",
```

```
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

若要移除反向 DNS 記錄

使用 Reset-EC2AddressAttribute cmdlet。

```
Reset-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -Attribute domain-name |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

以下為範例輸出。

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.com.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.net.
```

彈性網路介面

彈性網絡介面是代表虛擬網路卡之 VPC 中的邏輯網路元件。您可建立並設定網路介面，然後將它們連接到同一可用區域中啟用的執行個體。網路介面的屬性在它連接到執行個體或與執行個體分離，以及重新連接到另一個執行個體時，會一直跟著它。當您將網路介面從一個執行個體移至另一個執行個體時，網路流量會從原版執行個體重新導向至新的執行個體。

請注意，此 AWS 資源在 AWS 管理主控台 和 Amazon EC2 API 中稱為網路介面。因此，我們在本文件中使用「網路介面」而非「彈性網絡介面」。本文件中的「網路介面」一詞一律表示「彈性網絡介面」。

網路介面屬性

網路界面可包含下列屬性：

- 子網路 IPv4 地址範圍中的主要私有 IPv4 地址
- 子網路 IPv6 地址範圍中的主 IPv6 地址
- 子網路 IPv4 地址範圍中的輔助私有 IPv4 地址
- 每個私有 IPv4 地址一個彈性 IP 位址 (IPv4)
- 一個公有 IPv4 地址
- 次要 IPv6 地址
- Security groups (安全群組)
- 一個 MAC 地址
- 一個來源/目標檢查標記
- 一項描述

監控流量

您可在您的網路介面上啟用 VPC 流量日誌，擷取出入網路介面的流量資訊。建立流量日誌之後，您可以在 Amazon CloudWatch Logs 中檢視及擷取其資料。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的VPC 流程日誌。

目錄

- 網路介面概念
- 網路卡
- 每個網路介面的 IP 位址上限
- 為您的 EC2 執行個體建立網路介面
- EC2 執行個體的網路介面附件
- 管理網路介面的 IP 位址：
- 修改網路介面屬性
- Amazon EC2 執行個體的多個網路介面
- 申請者受管的網路介面
- Amazon EC2 網路介面的字首委派

- 刪除網路介面

網路介面概念

以下是開始使用網路介面時需要了解的重要概念。

主要網路介面

每個執行個體都有預設的網路介面，稱為主要網路介面。您無法分離主要網路介面和執行個體。

次要網路介面

您可以將次要網路介面建立並連接至執行個體。最大網路介面數因執行個體類型而異。如需詳細資訊，請參閱每個網路介面的 IP 位址上限。

網路介面的 IPv4 地址

當您將 EC2 執行個體啟動至僅限 IPv4 或雙重堆疊子網路時，此執行個體會從子網的 IPv4 地址範圍接收主要私有 IP 位址。您也可以指定其他私有 IPv4 地址 (稱為次要私有 IPv4 地址)。與主要私有 IP 位址不同，輔助私有 IP 位址可從原執行個體重新指派至別的執行個體。

網路介面的公有 IPv4 地址

所有子網路都有可修改的屬性，決定在該子網路中建立的網路介面 (因此執行個體在該子網路中啟動) 是否獲派公有 IPv4 地址。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的子網路設定。當您啟動執行個體時，IP 位址將會指派給主網路介面。如果您在啟動執行個體時指定現有網路介面作為主網路接口，則公有 IPv4 地址將由此網路介面決定。

當您建立網路介面時，它會繼承來自子網路的公有 IPv4 定址屬性。以後若要修改子網路的公有 IPv4 定址屬性，網路介面會保持建立時生效的設定。

當執行個體停止、休眠或終止時，我們會釋出公有 IP 位址。我們會在您啟動已停止或休眠的執行個體時指派新的公有 IP 位址，除非它具有次要網路介面或與彈性 IP 位址相關聯的次要私有 IPv4 位址。

網路介面的 IPv6 地址

如果您建立 IPv6 CIDR 區塊與您 VPC 及子網路的關聯，您可以將子網路範圍內的一或多個 IPv6 地址指派給網路介面。每個 IPv6 地址皆可指派給一個網路介面。

所有子網路都有可修改的屬性，決定在該子網路中建立的網路介面 (因此執行個體在該子網路中啟動) 是否自動獲派子網路範圍內 IPv6 地址。當您啟動執行個體時，IPv6 地址是指派給已建立的主要網路介面。

適用於網路介面的彈性 IP 位址

如有彈性 IP 位址，您可建立它與網路介面其中一個私有 IPv4 地址的關聯。您可建立一個彈性 IP 位址與每個私有 IPv4 地址的關聯。如果您取消彈性 IP 位址與網路介面的關聯，您可以將其釋放，或與不同的執行個體建立關聯。

終止行為

您可設定連接至執行個體之網路介面的終止行為。您可指定網路介面是否應該在您終止它所連接的執行個體時自動刪除。

來源/目標檢查

您可以啟用或停用來源/目的地檢查，這可確保執行個體是其接收之任何流量的來源或目的地。來源/目的地檢查預設為啟用。如果執行個體執行網路位置轉譯、路由或防火牆等服務，則必須停用來源/目標檢查。

申請者受管的網路介面

這些網路介面由 建立和管理 AWS 服務 ，讓您能夠使用一些資源和服務。您無法自行管理這些網路介面。如需詳細資訊，請參閱申請者受管的網路介面。

字首委派

字首是保留的私有 IPv4 或 IPv6 CIDR 範圍，您可以將該範圍配置給與執行個體相關聯的網路介面的自動或手動指派。透過使用委派字首，您就可以將 IP 位址範圍指派為單一字首，以更快速地啟動服務。

受管網路介面

受管網路介面由服務供應商管理，例如 Amazon EKS Auto Mode。您無法直接修改受管網路介面的設定。受管網路介面由受管欄位中的 true 值識別。如需詳細資訊，請參閱Amazon EC2 受管執行個體。

網路卡

多數執行個體類型都支援一張網路卡。支援多張網路卡的執行個體類型可提供更佳的網路效能，包括超過 100 Gbps 的頻寬功能和經過改善的封包速率效能。當您將網路介面連接到支援多個網路卡的執行個體時，您可以選取網路介面的網路卡。主要網路介面必須指派給網路卡索引 0。

EFA 和僅限 EFA 的網路介面會計為網路介面。每個網路卡只能指派一個 EFA 或僅限 EFA 的網路介面。主網路介面不能是僅 EFA 的網路介面。

下列執行個體類型支援多張網路卡。如需執行個體類型支援的網路介面數量資訊，請參閱每個網路介面的 IP 位址上限。

執行個體類型	網路卡數量
c6in.32xlarge	2
c6in.metal	2
c8gb.48xlarge	2
c8gb.metal-48xl	2
c8gn.48xlarge	2
c8gn.metal-48xl	2
c8in.96xlarge	2
c8in.metal-96xl	2
c8ib.96xlarge	2
c8ib.metal-96xl	2
dl1.24xlarge	4
g6e.24xlarge	2
g6e.48xlarge	4
g7.48xlarge	2
g7e.24xlarge	2
g7e.48xlarge	4
hpc6id.32xlarge	2
hpc7a.12xlarge	2
hpc7a.24xlarge	2

執行個體類型	網路卡數量
hpc7a.48xlarge	2
hpc7a.96xlarge	2
hpc8a.96xlarge	2
i7ie.48xlarge	2
i7ie.metal-48xl	2
i8ge.48xlarge	2
i8ge.metal-48xl	2
m6idn.32xlarge	2
m6idn.metal	2
m6in.32xlarge	2
m6in.metal	2
m8gb.48xlarge	2
m8gb.metal-48xl	2
m8gn.48xlarge	2
m8gn.metal-48xl	2
m8in.96xlarge	2
m8in.metal-96xl	2
m8idn.96xlarge	2
m8idn.metal-96xl	2
m8ib.96xlarge	2

執行個體類型	網路卡數量
m8ib.metal-96xl	2
m8idb.96xlarge	2
m8idb.metal-96xl	2
p4d.24xlarge	4
p4de.24xlarge	4
p5.48xlarge	32
p5e.48xlarge	32
p5en.48xlarge	16
p6-b200.48xlarge	8
p6-b300.48xlarge	17
p6e-gb200.36xlarge	17
r8gb.48xlarge	2
r8gb.metal-48xl	2
r8gn.48xlarge	2
r8gn.metal-48xl	2
r8in.96xlarge	2
r8in.metal-96xl	2
r8idn.96xlarge	2
r8idn.metal-96xl	2
r8ib.96xlarge	2

執行個體類型	網路卡數量
r8ib.metal-96xl	2
r8idb.96xlarge	2
r8idb.metal-96xl	2
r6idn.32xlarge	2
r6idn.metal	2
r6in.32xlarge	2
r6in.metal	2
trn1.32xlarge	8
trn1n.32xlarge	16
trn2.48xlarge	16
trn2u.48xlarge	16
u7in-16tb.224xlarge	2
u7in-24tb.224xlarge	2
u7in-32tb.224xlarge	2
u7inh-32tb.480xlarge	2

每個網路介面的 IP 位址上限

每個執行個體類型都支援最大數量的網路介面、每個網路介面的私有 IPv4 地址數量上限，以及每個網路介面的 IPv6 地址數量上限。每個網路介面的 IPv6 地址限制和私有 IPv4 地址限制是分開的。請注意，所有執行個體類型都支援 IPv6 定址，但下列項目除外：C1、M1、M2、M3 和 T1。

可用的網路介面

Amazon EC2 執行個體類型指南提供每個執行個體類型可用的網路介面相關資訊。如需詳細資訊，請參閱下列內容：

- 網路規格 – 一般用途
- 網路規格 – 運算最佳化
- 網路規格 – 記憶體最佳化
- 網路規格 – 儲存最佳化
- 網路規格 – 加速運算
- 網路規格 – 高效能運算
- 網路規格 – 上一代

Console

若要擷取網路介面上限

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instance Types (執行個體類型)。
3. 新增篩選條件以指定執行個體類型 (Instance type=c5.12xlarge) 或執行個體系列 (Instance family=c5)。
4. （選用）選擇偏好設定圖示，然後開啟網路介面的數量上限。此資料欄指出每個執行個體類型的網路介面數量上限。
5. (選用) 選擇執行個體類型。在聯網索引標籤上，尋找網路介面的數量上限。

AWS CLI

若要擷取網路介面上限

可以使用 describe-instance-types 命令來顯示執行個體類型的相關資訊，例如：支援的網路介面和每個介面的 IP 位址。下列範例顯示所有 C8i 執行個體的此資訊。

```
{ echo -e "InstanceType\tMaximumNetworkInterfaces\tIpv4AddressesPerInterface"; \
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c8i.*" \
```

```
    --query 'InstanceTypes[*].[InstanceType, NetworkInfo.MaximumNetworkInterfaces,
 NetworkInfo.Ipv4AddressesPerInterface]' \
    --output text | sort -k2 -n; } | column -t
```

以下為範例輸出。

```
InstanceType    MaximumNetworkInterfaces  Ipv4AddressesPerInterface
c8i.large       3                         20
c8i.2xlarge     4                         30
c8i.xlarge      4                         30
c8i.4xlarge     8                         50
c8i.8xlarge     10                        50
c8i.12xlarge    12                        50
c8i.16xlarge    16                        64
c8i.24xlarge    16                        64
c8i.32xlarge    24                        64
c8i.48xlarge    24                        64
c8i.96xlarge    24                        64
c8i.metal-48xl  24                        64
c8i.metal-96xl  24                        64
```

PowerShell

若要擷取網路介面上限

您可以使用 Get-EC2 Instance Type PowerShell 命令來顯示執行個體類型的相關資訊，例如其支援的網路介面和每個介面的 IP 位址。下列範例顯示所有 C8i 執行個體的此資訊。

```
Get-EC2InstanceType -Filter @{Name="instance-type"; Values="c8i.*"} |
Select-Object `
    InstanceType,
    @{Name='MaximumNetworkInterfaces';
 Expression={$_.NetworkInfo.MaximumNetworkInterfaces}},
    @{Name='Ipv4AddressesPerInterface';
 Expression={$_.NetworkInfo.Ipv4AddressesPerInterface}} |
Sort-Object MaximumNetworkInterfaces |
Format-Table -AutoSize
```

以下為範例輸出。

```
InstanceType   MaximumNetworkInterfaces Ipv4AddressesPerInterface
------------   ------------------------ -------------------------
```

c8i.large	3	20
c8i.xlarge	4	30
c8i.2xlarge	4	30
c8i.4xlarge	8	50
c8i.8xlarge	10	50
c8i.12xlarge	12	50
c8i.24xlarge	16	64
c8i.16xlarge	16	64
c8i.96xlarge	24	64
c8i.48xlarge	24	64
c8i.metal-96xl	24	64
c8i.32xlarge	24	64
c8i.metal-48xl	24	64

為您的 EC2 執行個體建立網路介面

您可以建立網路介面以供 EC2 執行個體使用。當您建立網路介面時，您可以指定其建立所在的子網路。建立網路介面之後，無法將其移至另一個子網路。必須將網路介面連接到同一可用區域中的執行個體。您可以將次要網路介面從某個執行個體分離，然後將其附加到相同可用區域內的其他執行個體上。您無法分離主要網路介面和執行個體。如需詳細資訊，請參閱the section called “網路介面連接”。

Console

若要建立網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選擇 Create network interface (建立網路介面)。
4. (選用) 針對描述，輸入描述性名稱。
5. 針對子網路，選取子網路。後續步驟中可用的選項取決於您選取的子網路類型 (僅限 IPv4、僅限 IPv6 或雙堆疊 (IPv4 和 IPv6))。
6. 針對介面類型，選擇以下其中一項：
   - ENA：高效能網路介面，旨在處理 TCP/IP 通訊協定的高輸送量和每秒封包速率，同時最大限度減少 CPU 使用。這是預設值。如需關於 ENA 的詳細資訊，請參閱彈性網路介面卡。
   - EFA 與 ENA：支援 ENA 和 EFA 裝置的網路介面，用於傳統 TCP/IP 型傳輸以及 SRD 型傳輸。如果選擇 EFA 搭配 ENA，則要與其連接的執行個體必須支援 EFA。如需 EFA 的詳細資訊，請參閱 Elastic Fabric Adapter。

- 僅限 EFA：高效能網路介面，旨在處理 SRD 型傳輸的高輸送量、低延遲節點間通訊，同時繞過作業系統堆疊。如果選擇此選項，則要與其連接的執行個體必須支援 EFA。僅限 EFA 的網路介面不支援 IP 位址。如需 EFA 的詳細資訊，請參閱 Elastic Fabric Adapter。

7. 針對 Private IPv4 address (私有 IPv4 地址)，執行下列其中一項操作：
   - 選取 Auto-assign (自動指派)，以允許 Amazon EC2 從子網路選取 IPv4 地址。
   - 選取 Custom (自訂)，然後輸入您從子網路選取的 IPv4 地址。
8. (僅限使用 IPv6 地址的子網路) 針對 IPv6 地址，執行以下其中一個動作：
   - 如果您不想將 IPv6 地址指派給網路介面，則選擇 None (無)。
   - 選取 Auto-assign (自動指派)，以允許 Amazon EC2 從子網路選取 IPv6 地址。
   - 選取 Custom (自訂)，然後輸入您從子網路選取的 IPv6 地址。
9. (選用) 如果您要在雙堆疊或僅限 IPv6 的子網路中建立網路介面，則可以選擇指派主要 IPv6 IP。這會將主要 IPv6 全域單播位址 (GUA) 指派給網路介面。指派主要 IPv6 地址可讓您避免中斷執行個體或 ENI 的流量。如果要連接此 ENI 的執行個體倚賴其 IPv6 地址而不變更，請選擇啟用。 AWS 會自動將與連接至執行個體的 ENI 相關聯的 IPv6 地址指派為主要 IPv6 地址。將 IPv6 GUA 地址啟用為主要 IPv6 之後，您就無法將其停用。當您啟用 IPv6 GUA 地址作為主要 IPv6 地址時，第一個 IPv6 GUA 地址會成為主要 IPv6 地址，直到執行個體終止或網路介面中斷連接為止。如果有多個與執行個體連接之 ENI 相關聯的 IPv6 地址，並且啟用主要 IPv6 地址，則與 ENI 相關聯的第一個 IPv6 GUA 地址會成為主要 IPv6 地址。
10. (選用) 若要建立 Elastic Fabric Adapter，選擇Elastic Fabric Adapter、Enable (啟用)。
11. (選用) 在進階設定下，您可以選擇設定 IP 字首委派。如需詳細資訊，請參閱字首委派。
    - 自動指派 — 從子網路的 IPv4 或 IPv6 CIDR 區塊 AWS 中選擇字首，並將其指派給網路介面。
    - 自訂 — 您可以從子網路的 IPv4 或 IPv6 CIDR AWS 區塊指定字首，並驗證字首尚未指派給其他資源，然後再指派給網路介面。
12. (選用) 在進階設定下的閒置連線追蹤逾時，修改預設閒置連線逾時。如需詳細資訊，請參閱閒置連線追蹤逾時。
    - 已建立的 TCP 逾時：處於已建立狀態的閒置 TCP 連線逾時 (以秒為單位)。
      - 最小值：60 秒
      - 最大值：432000 秒
      - 預設：Nitro v6 執行個體類型的350秒數 (P6e-GB200 除外）；所有其他執行個體類型的432000秒數（包括 P6e-GB200)。
      - 建議：432000 秒以下。

- UDP 逾時：只出現單一方向或單一請求回應交易流量的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：30 秒
  - 最大值：60 秒
  - 預設：30 秒
- UDP 串流逾時：分類為出現多個請求回應交易串流的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：60 秒
  - 最大值：180 秒
  - 預設：180 秒

13. 針對 Security groups (安全群組)，選取一個或多個安全群組。
14. (選用) 針對每個標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤金鑰和選用標籤值。
15. 選擇 Create network interface (建立網路介面)。

AWS CLI

範例 1：若要建立具有 Amazon EC2 所選 IP 位址的網路介面

使用下列 create-network-interface 命令。此範例建立具有 Amazon EC2 所選公有 IPv4 位址和 IPv6 位址的網路介面。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-1234567890abcdef0
```

範例 2：建立具有特定 IP 位址的網路介面

使用下列 create-network-interface 命令。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --private-ip-address 10.251.50.12 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 \
    --groups sg-1234567890abcdef0
```

範例 3：建立具有次要 IP 位址計數的網路介面

使用下列 create-network-interface 命令。在此範例中，Amazon EC2 會同時選擇主要 IP 位址和次要 IP 位址。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --secondary-private-ip-address-count 2 \
    --groups sg-1234567890abcdef0
```

範例 4：建立具有特定次要 IP 位址的網路介面

使用下列 create-network-interface 命令。此範例指定主要 IP 位址和次要 IP 位址。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --private-ip-addresses PrivateIpAddress=10.0.1.30,Primary=true \
                           PrivateIpAddress=10.0.1.31,Primary=false
    --groups sg-1234567890abcdef0
```

PowerShell

範例 1：若要建立具有 Amazon EC2 所選 IP 位址的網路介面

使用 New-EC2NetworkInterface cmdlet。此範例建立具有 Amazon EC2 所選公有 IPv4 位址和 IPv6 位址的網路介面。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -Ipv6AddresCount 1 `
    -Group sg-1234567890abcdef0
```

範例 2：建立具有特定 IP 位址的網路介面

使用 New-EC2NetworkInterface cmdlet。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
```

```
    -PrivateIpAddress 10.251.50.12 `
    -Ipv6Address $ipv6addr `
    -Group sg-1234567890abcdef0
```

定義 IPv6 位址，如下所示。

```
$ipv6addr = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
```

範例 3：建立具有次要 IP 位址計數的網路介面

使用 New-EC2NetworkInterface cmdlet。在此範例中，Amazon EC2 會同時選擇主要 IP 位址和次要 IP 位址。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -SecondaryPrivateIpAddressCount 2 `
    -Group sg-1234567890abcdef0
```

範例 4：建立具有特定次要 IP 位址的網路介面

使用 New-EC2NetworkInterface cmdlet。此範例指定主要 IP 位址和次要 IP 位址。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -PrivateIpAddresses @($primary, $secondary) `
    -Group sg-1234567890abcdef0
```

定義次要位址，如下所示。

```
$primary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$primary.PrivateIpAddress = "10.0.1.30"
$primary.Primary = $true
$secondary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$secondary.PrivateIpAddress = "10.0.1.31"
$secondary.Primary = $false
```

EC2 執行個體的網路介面附件

您可以建立網路介面，以供 EC2 執行個體做為主要或次要網路介面使用。如果 EC2 執行個體與網路介面位於相同可用區域，則必須將網路介面連接到 EC2 執行個體。執行個體的執行個體類型決定您可以連接到執行個體的網路介面數量。如需詳細資訊，請參閱the section called “每個執行個體類型每網路介面的 IP 位址”。

考量事項

- 您可在執行個體執行時 (熱連接)、停止時 (暖連接) 或正在啟動時 (冷連接) 將網路介面連接到執行個體。
- 您可在執行個體執行時或停止時分離輔助網路介面。但是您不能分離主要網路介面。
- 您可以將次要網路介面從一個執行個體分離，並將其連接至另一個執行個體。
- 使用 CLI、API 或開發套件啟動執行個體時，您可以指定主要網路介面和其他網路介面。請注意，如果在啟動期間新增次要網路介面，則無法啟用公有 IPv4 位址的自動指派。
- 以多個網路介面啟動 Amazon Linux 或 Windows Server 執行個體，會自動設定執行個體作業系統的介面、私有 IPv4 地址和路由表。
- 其他網路介面的暖或熱連接可能需要您手動帶出第二個介面、設定私有 IPv4 地址，以及相應修改路由表。執行 Amazon Linux 或 Windows Server 的執行個體會自動辨識暖或熱連接並自行設定。
- 您不能將另一個網路介面連接到執行個體 (例如，NIC 小組組態)，以增加或加倍雙目錄執行個體傳入或傳出的網路頻寬。
- 如果您將相同子網路中兩個或以上的網路介面連接到執行個體，您可能會發生聯網問題，例如非對稱路由。如果可能，請改在主要網路介面上使用輔助私有 IPv4 地址。
- 對於僅限 IPv6 子網路中的 EC2 執行個體，如果您將次要網路介面連接到執行個體，則第二個網路介面的私有 DNS 主機名稱將解析為執行個體第一個網路介面上的第一個 IPv6 地址。
- [Windows 執行個體] – 如果向執行個體新增多個網路介面，則必須將網路介面設定為使用靜態路由。

連接網路介面

使用 Amazon EC2 主控台的執行個體或網路介面頁面，可將網路介面連接到與該網路介面相同的可用區域內的任何執行個體。或者，可以在啟動執行個體時指定現有網路介面。

如已釋放您執行個體上的公有 IPv4 地址，如有多個網路界面連接到執行個體，就不會收到新的地址。如需公有 IPv4 地址行為的詳細資訊，請參閱公有 IPv4 地址。

Console

若要使用「執行個體」頁面來連接網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體的核取方塊。
4. 選擇 Actions (動作)、Networking (網路)、Attach network interface (連接網路介面)。
5. 選擇 VPC。網路介面可以位於與執行個體相同的 VPC 或您擁有的不同 VPC，只要網路介面與執行個體位於相同的可用區域即可。如此一來，您就能在不同 VPC 建立多重主目錄執行個體，分別使用不同的網路和安全組態。
6. 選取網路介面。如果執行個體支援多張網路卡，您可以選擇網路卡。
7. 選擇 Attach (連接)。

若要使用「網路介面」頁面來連接網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面的核取方塊。
4. 選擇 Actions (動作)、Attach (連接)。
5. 選擇執行個體。如果執行個體支援多張網路卡，您可以選擇網路卡。
6. 選擇 Attach (連接)。

AWS CLI

若要連接網路介面

使用下列 attach-network-interface 命令。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

PowerShell

若要連接網路介面

使用 Add-EC2NetworkInterface cmdlet。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1
```

分離網路介面

您可以使用 Amazon EC2 主控台的 Instances (執行個體) 或 Network Interfaces (網路介面) 頁面，隨時分離連接至 EC2 執行個體的輔助網路介面。

如果您嘗試分開連接到其他服務 (例如 Elastic Load Balancing 負載平衡器、Lambda 函數、WorkSpace 或 NAT 閘道) 之資源的網路介面，您會收到錯誤，指示您沒有存取資源的許可。若要找出哪個服務建立了連接至網路介面的資源，請檢查網路介面的描述。如果您刪除資源，則會刪除其網路介面。

Console

若要使用「執行個體」頁面來分離網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體的核取方塊。檢查 Networking (網路) 標籤的 Network interfaces (網路介面) 區段，確認網路介面已連接至執行個體作為次要網路介面。
4. 選擇 Actions (動作)、Networking (網路)、Detach network interface (分離網路介面)。
5. 選取網路介面，然後選取 Detach (分離)。

若要使用「網路介面」頁面來分離網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面的核取方塊。檢查 Networking (網路) 標籤的 Instance details (執行個體詳細資訊) 區段，確認網路介面已連接至執行個體作為次要網路介面。
4. 選擇 Actions (動作)、Detach (分離)。
5. 當出現確認提示時，選擇 Detach (分離)。

6. 如果網路介面無法從執行個體分離，請選擇 Force detachment (強制分離)、Enable (啟用)，然後再試一次。我們建議強制分離只作為最後的手段。強制分離可以防止您在相同索引上連接不同的網路介面，直到重新啟動執行個體為止。它也可以防止執行個體中繼資料反映網路介面已分離，直到您重新啟動執行個體為止。

AWS CLI

若要分離網路介面

使用下列 detach-network-interface 命令。

```
aws ec2 detach-network-interface --attachment-id eni-attach-016c93267131892c9
```

PowerShell

若要分離網路介面

使用 Dismount-EC2NetworkInterface cmdlet。

```
Dismount-EC2NetworkInterface -AttachmentId eni-attach-016c93267131892c9
```

管理網路介面的 IP 位址：

您可以管理以下網路介面的 IP 位址：

- 彈性 IP 位址 (每個私有 IPv4 地址各一個)
- IPv4 地址
- IPv6 地址

Console

若要管理網路介面的 IP 位址

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面的核取方塊。
4. 若要管理 IPv4 和 IPv6 地址，請執行下列動作：

a. 選擇 Actions (動作)、Manage IP addresses (管理 IP 位址)。
b. 展開網路介面。
c. 針對 IPv4 位址，視需要修改 IP 位址。若要指派其他 IPv4 地址，請選擇指派新的 IP 地址，然後從子網路範圍指定 IPv4 地址，或讓 為您 AWS 選擇一個地址。
d. (雙堆疊或僅限 IPv6) 對於 IPv6 位址，視需要編輯 IP 位址。若要指派其他 IPv6 位址，請選擇指派新的 IP 位址，然後從子網路範圍指定 IPv6 位址或讓 AWS 為您選擇。
e. 若要指派或取消指派公有 IPv4 地址至網路介面，請選擇自動指派公有 IP。可以針對任何網路介面啟用或停用此功能，但它只會影響主要網路介面。
f. (雙堆疊或僅限 IPv6) 在指派主要 IPv6 IP 中，選擇啟用以指派主要 IPv6 位址。將與網路介面相關聯的第一個 GUA 選為主要 IPv6 位址。指派主要 IPv6 位址之後，就無法變更。此位址是主要 IPv6 位址，直到執行個體終止或網路介面分離為止。
g. 選擇儲存。

5. 若要建立彈性 IP 位址的關聯，請執行下列操作：
   a. 選擇 Actions (動作)、Associate address (建立與地址的關聯)。
   b. 針對 Elastic IP address (彈性 IP 位址)，選取彈性 IP 位址。
   c. 針對 Private IPv4 address (私有 IPv4 地址)，選取要與彈性 IP 位址建立關聯的私有 IPv4 地址。
   d. (選用) 如果網路介面目前與另一個執行個體或網路介面建立關聯，選擇 Allow the Elastic IP address to be reassociated (允許與彈性 IP 位址重新建立關聯)。
   e. 選擇 Associate (關聯)。
6. 若要取消彈性 IP 位址的關聯，請執行下列操作：
   a. 選擇 Actions (動作)、Disassociate address (取消與地址的關聯)。
   b. 針對 Public IP address (公有 IP 位址)，選取彈性 IP 位址。
   c. 選擇 Disassociate (取消關聯)。

AWS CLI

若要管理 IPv4 位址

使用下列 assign-private-ip-addresses 命令來指派 IPv4 位址。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
```

```
    --private-ip-addresses 10.0.0.82
```

使用下列 unassign-private-ip-addresses 命令來取消指派 IPv4 位址。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

若要管理 IPv6 位址

使用下列 assign-ipv6-addresses 命令來指派 IPv6 位址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

使用下列 unassign-ipv6-addresses 命令來取消指派 IPv6 位址。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

若要管理主要私有 IPv4 位址的彈性 IP 位址

使用下列 associate-address 命令，將彈性 IP 位址與主要私有 IPv4 位址進行關聯。

```
aws ec2 associate-address \
    --allocation-id eipalloc-0b263919b6EXAMPLE \
    --network-interface-id eni-1234567890abcdef0
```

使用以下 disassociate-address 命令，取消彈性 IP 位址與主要私有 IPv4 位址的關聯。

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745a1EXAMPLE
```

PowerShell

若要管理 IPv4 位址

使用 Register-EC2PrivateIpAddress cmdlet 來指派 IPv4 位址。

```
Register-EC2PrivateIpAddress `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

使用 Unregister-EC2PrivateIpAddress cmdlet 取消指派 IPv4 位址。

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

若要管理 IPv6 位址

使用 Register-EC2Ipv6AddressList cmdlet 來指派 IPv6 位址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

使用 Unregister-EC2Ipv6AddressList cmdlet 來取消指派 IPv6 位址。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

若要管理主要私有 IPv4 位址的彈性 IP 位址

使用 Register-EC2Address cmdlet，將彈性 IP 位址與主要私有 IPv4 位址建立關聯。

```
Register-EC2Address `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -AllocationId eipalloc-0b263919b6EXAMPLE
```

使用 Unregister-EC2Address cmdlet，取消彈性 IP 位址與主要私有 IPv4 位址的關聯。

```
Unregister-EC2Address -AssociationId eipassoc-2bebb745a1EXAMPLE
```

修改網路介面屬性

您可以變更下列網路介面屬性：

- 描述
- 安全群組
- 在終止時刪除
- 來源/目標檢查
- 閒置連線追蹤逾時

考量事項

您無法變更請求者管理的網路介面。

Console

若要修改網路介面屬性

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面的核取方塊。
4. 若要變更描述，請執行下列操作：
   a. 選擇 Actions (動作)、Change description (變更描述)。
   b. 在描述中，輸入描述。
   c. 選擇儲存。
5. 如要變更安全群組，請執行下列操作：
   a. 選擇 Actions (動作)、Change security groups (變更安全群組)。
   b. 對於關聯的安全群組，視需要新增和移除安全群組。必須為同一個 VPC 建立安全群組和網路介面。
   c. 選擇儲存。
6. 若要變更終止行為，請執行下列動作：
   a. 選擇 Actions (動作)、Change termination behavior (變更終止行為)。
   b. 依次選取或清除終止時刪除、啟用。
   c. 選擇儲存。
7. 若要變更來源/目的地檢查，請執行下列動作：
   a. 選擇 Actions (動作)、Change source/dest check (變更來源/目的地檢查)。

b. 依次選取或清除來源/目的地檢查、啟用。

c. 選擇儲存。

8. 若要變更閒置連線追蹤逾時，請執行下列動作：

a. 依次選擇動作、修改閒置連線追蹤逾時。

b. 視需要修改逾時值。如需詳細資訊，請參閱閒置連線追蹤逾時。

- 已建立的 TCP 逾時：處於已建立狀態的閒置 TCP 連線逾時 (以秒為單位)。
  - 最小值：60 秒
  - 最大值：432000 秒
  - 預設：Nitro v6 執行個體類型的350秒數 (P6e-GB200 除外）；所有其他執行個體類型的432000秒數（包括 P6e-GB200)。
  - 建議：432000 秒以下。
- UDP 逾時：只出現單一方向或單一請求回應交易流量的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：30 秒
  - 最大值：60 秒
  - 預設：30 秒
- UDP 串流逾時：分類為出現多個請求回應交易串流的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：60 秒
  - 最大值：180 秒
  - 預設：180 秒

c. 選擇儲存。

AWS CLI

Example範例：若要修改描述

使用下列 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --description "my updated description"
```

Example範例：若要修改安全群組

使用下列 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

Example範例：若要修改終止行為

使用下列 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --attachment AttachmentId=eni-attach-43348162abEXAMPLE,DeleteOnTermination=false
```

Example範例：若要啟用來源/目的地檢查

使用下列 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --source-dest-check
```

Example範例：若要修改閒置連線追蹤逾時

使用下列 modify-network-interface-attribute 命令。如需詳細資訊，請參閱閒置連線追蹤逾時。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --connection-tracking-specification
 TcpEstablishedTimeout=172800,UdpStreamTimeout=90,UdpTimeout=60
```

PowerShell

Example範例：若要修改描述

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
```

```
    -Description "my updated description"
```

Example範例：若要修改安全群組

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

Example範例：若要修改終止行為

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Attachment_AttachmentId eni-attach-43348162abEXAMPLE `
    -Attachment_DeleteOnTermination $false
```

Example範例：若要啟用來源/目的地檢查

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SourceDestCheck $true
```

Example範例：若要修改閒置連線追蹤逾時

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。如需詳細資訊，請參閱閒置連線追蹤逾時。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -ConnectionTrackingSpecification_TcpEstablishedTimeout 172800 `
    -ConnectionTrackingSpecification_UdpStreamTimeout 90 `
    -ConnectionTrackingSpecification_UdpTimeout 60
```

Amazon EC2 執行個體的多個網路介面

當您想要執行下列作業時，將多個網路介面連接到一個執行個體會很有用：

- 網路管理。
- 網路和安全設備。
- 在不同子網路或 VPCs 中具有工作負載的雙主目錄執行個體。
- 建立低預算、高可用性的解決方案。

網路管理

下列概觀說明使用多個網路介面建立的管理網路。

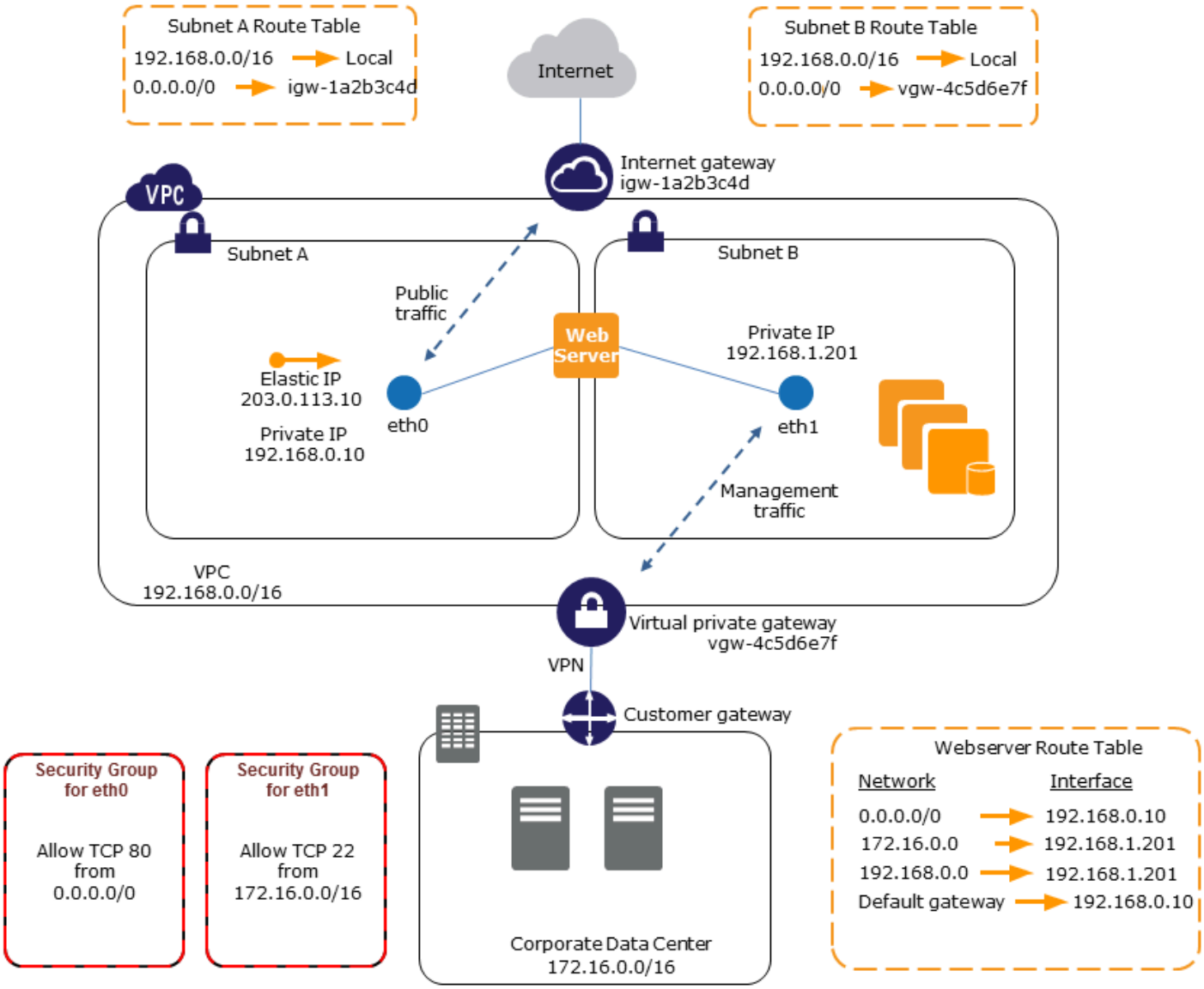
條件

- 執行個體上的主要網路介面 (例如：eth0) 會處理公有流量。
- 執行個體上的次要網路介面 (例如：eth1) 會處理後端管理流量。它連線到另一個子網路，具有更嚴格的存取控制，並與主要網路介面位於同一個可用區域 (AZ)。

設定

- 主要網路介面可能在負載平衡器後面，也可能不在負載平衡器後面，具有關聯的安全群組，可允許從網際網路存取伺服器。例如，允許從 0.0.0.0/0 或負載平衡器存取 TCP 連接埠 80 和 443。
- 次要網路介面具有關聯的安全群組，僅允許從下列其中一個位置起始的 SSHRDP 存取：
  - 允許的 IP 位址範圍 (VPC 內或來自網際網路)。
  - 與主要網路介面位於相同 AZ 中的私有子網路。
  - 虛擬私有閘道。

> **Note**
>
> 為確保容錯移轉功能，請考慮使用輔助私有 IPv4 處理網路介面的傳入流量。執行個體故障時，您可將執行個體及/或輔助私有 IPv4 地址移至待命的執行個體。



網路和安全設備

有些網路和安全設備，如負載平衡器、網路位置轉譯 (NAT) 伺服器和代理伺服器等，最好能設定多個網路介面。您可建立輔助網路介面，並將它們連接到正在執行這些應用程式類型的執行個體，然後使用它們自己的公有和私有 IP 位址、安全群組和來源/目標檢查來設定其他介面。

在不同子網路中具有工作負載的雙主目錄執行個體

您可在連線至應用程式伺服器所在之中間層網路的每一個 Web 伺服器上放置網路介面。應用程式伺服器也可以是資料庫伺服器所在之後端網路 (子網路) 的雙目錄。每個雙目錄執行個體都會在前端收到及

處理請求、起始後端連線，然後將請求傳送至後端網路的伺服器，而非透過雙目錄執行個體路由網路套件。

相同帳戶中不同 VPCs中具有工作負載的雙主目錄執行個體

您可以在一個 VPC 中啟動 EC2 執行個體，並將另一個 VPC (位於同一個可用區域) 的次要 ENI 連接到執行個體。如此一來，您就能在不同 VPC 建立多重主目錄執行個體，分別使用不同的網路和安全組態。您無法在不同 AWS 帳戶中跨 VPCs 建立多主目錄執行個體。

在下列使用案例中，您可以跨 VPC 使用雙主目錄執行個體：

- 克服兩個無法對等的 VPC 之間的 CIDR 重疊問題：您可以在 VPC 中運用次要 CIDR，並允許執行個體跨兩個非重疊的 IP 範圍進行通訊。
- 在單一帳戶內連接多個 VPC：啟用個別資源之間的通訊，這些資源通常會以 VPC 界限隔開。

低預算、高可用性解決方案

如果服務特定函數的其中一個執行個體失敗，其網路介面可以連接到為相同角色預先設定的替代或熱待命執行個體，以快速復原服務。例如，您可使用網路介面做為重大服務的主要或輔助網路介面，例如資料庫執行個體或 NAT 執行個體。如果執行個體故障，您 (或更有可能是代您執行的程式碼) 可將網路介面連接到熱待命執行個體。因為執行個體維持其私有 IP 位址、彈性 IP 位址和 MAC 地址，所以您一旦將網路介面連接到替代執行個體，網路流量即開始流向待命的執行個體。使用者從執行個體故障到網路介面連接到待命執行個體這段時間內，會短暫遺失連線能力，但不需要變更路由表或您的 DNS 伺服器。

申請者受管的網路介面

請求者受管的網路介面是 AWS 服務 代表您在您的 VPC 中建立的網路介面。網路介面與其他服務的資源相關聯，例如來自 Amazon RDS 的資料庫執行個體、NAT 閘道或來自 AWS PrivateLink的介面 VPC 端點。

考量事項

- 您可以檢視帳戶中的申請者受管網路介面，也可以新增或移除標籤。不過，您無法在申請者管理的網路界面上執行下列動作：
  - 修改網路介面屬性
  - 連接或分離網路界面
  - 關聯或取消關聯彈性 IP 地址

- 指派或取消指派私有 IP 地址
- 啟動執行個體時指定網路界面

即使您無法修改，您仍然可以重設網路介面屬性。

- 當您刪除與申請者受管網路介面相關聯的資源時， 會 AWS 服務 分離網路介面並將其刪除。如果服務分離網路介面，但沒有將其刪除，則可以刪除分離的網路介面。

Console

若要檢視請求者管理的網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，依序選擇 Network & Security (網路與安全) 和 Network Interfaces (網路介面)。
3. 選取網路介面 ID，開啟其詳細資訊頁面。
4. 下列為可用來確定網路介面用途的重點欄位：
   - Description (說明)：由建立介面的 AWS 服務提供的說明。例如，"VPC Endpoint Interface vpce 089f2123488812123"。
   - 申請者受管：指出網路界面是否由 管理 AWS。
   - 申請者 ID：建立網路界面的委託人或服務別名或 AWS 帳戶 ID。如果您建立了網路界面，這是您的 AWS 帳戶 ID。否則，其他主體或服務將會加以建立。

AWS CLI

若要檢視請求者管理的網路介面

使用 describe-network-interfaces 命令，如下所示。

```
aws ec2 describe-network-interfaces \
    --filters "Name=requester-managed,Values=true" \
    --query "NetworkInterfaces[*].[Description, InterfaceType]" \
    --output table
```

下列為顯示可用來確定網路介面用途之重點欄位的範例輸出：`Description` 和 `InterfaceType`。

```
-------------------------------------------------------------------------------
|                          DescribeNetworkInterfaces                          |
+-----------------------------------------------------+-------------------------+
|  VPC Endpoint Interface: vpce-0f00567fa8477a1e6     |  interface              |
|  VPC Endpoint Interface vpce-0d8ddce4be80e4474      |  interface              |
|  VPC Endpoint Interface vpce-078221a1e27d1ea5b      |  vpc_endpoint           |
|  Resource Gateway Interface rgw-0bba03f3d56060135   |  interface              |
|  VPC Endpoint Interface: vpce-0cc199f605eaeace7     |  interface              |
|  VPC Endpoint Interface vpce-019b90d6f16d4f958      |  interface              |
+-----------------------------------------------------+-------------------------+
```

PowerShell

若要檢視請求者管理的網路介面

使用 Get-EC2NetworkInterface cmdlet，如下所示。

```
Get-EC2NetworkInterface -Filter @{Name="requester-managed"; Values="true"} | Select
 Description, InterfaceType
```

下列為輸出範例，它顯示可用來確定網路介面用途之重點欄位：Description 和 InterfaceType。

```
Description                                       InterfaceType
-----------                                       -------------
VPC Endpoint Interface: vpce-0f00567fa8477a1e6    interface
VPC Endpoint Interface vpce-0d8ddce4be80e4474     interface
VPC Endpoint Interface vpce-078221a1e27d1ea5b     vpc_endpoint
Resource Gateway Interface rgw-0bba03f3d56060135  interface
VPC Endpoint Interface: vpce-0cc199f605eaeace7    interface
VPC Endpoint Interface vpce-019b90d6f16d4f958     interface
```

Amazon EC2 網路介面的字首委派

您可以自動或手動將私有 IPv4 或 IPv6 CIDR 範圍指派給網路介面。透過指派字首，您就可以擴展和簡化應用程式的管理，包括需要在執行個體上有多個 IP 地址的容器和聯網應用程式。如需 IPv4 和 IPv6 地址的詳細資訊，請參閱 Amazon EC2 執行個體 IP 定址。

以下為可用的指派選項：

- 自動指派 — AWS 選擇字首並將其指派給您的網路界面。如果網路介面的子網路具有 `prefix` 類型的子網路 CIDR 預留，我們會從子網路 CIDR 預留中選取字首。否則，我們會從子網路 CIDR 範圍中進行選取。
- 手動指派 — 您可以指定字首，並 AWS 驗證它尚未指派給其他資源，然後再指派給您的網路介面。

指派字首有下列優點：

- 增加網路介面上的 IP 地址 – 當您使用字首時，會指派 IP 地址區塊，而非個別 IP 地址。這會增加網路介面的 IP 地址數量。
- 簡化容器的 VPC 管理 – 在容器應用程式中，每個容器都需要唯一的 IP 地址。為執行個體指派字首可簡化 VPC 的管理，因為您可以啟動和終止容器，而無需為個別 IP 指派呼叫 Amazon EC2 API。

內容

- 基本概念
- 考量事項
- 網路介面字首
  - 在建立網路介面期間指派字首
  - 為現有的網路介面指派字首
  - 從網路介面移除字首

基本概念

- 您可以為新的或現有的網路介面指派字首。
- 若要使用字首，請先為網路介面指派字首，然後將網路介面連接至執行個體，接著設定作業系統。
- 當您選擇指定字首的選項時，該字首必須符合下列需求：
  - 您可以指定的 IPv4 字首是 `/28`。
  - 您可以指定的 IPv6 字首是 `/80`。
  - 字首位於網路介面的子網 CIDR 中，且不會與指派給子網中現有資源的其他字首或 IP 地址重疊。
- 您可以將字首指派給主要或次要網路介面。
- 您可以將彈性 IP 地址指派給具有指派字首的網路介面。
- 您也可以將彈性 IP 地址指派給已指派字首的 IP 地址部分。
- 我們會將執行個體的私有 DNS 主機名稱解析為主要私有 IPv4 位址。

- 我們會使用以下格式，為網路介面指派每個私有 IPv4 地址，包括字首的私有 IPv4 地址：
  - us-east-1 區域

```
ip-private-ipv4-address.ec2.internal
```

  - 所有其他區域

```
ip-private-ipv4-address.region.compute.internal
```

考量事項

使用字首時，請考慮下列事項：

- 以 Nitro 為基礎的執行個體支援帶有字首的網路介面。
- 網路介面的字首必須使用 IPv6 地址和私有 IPv4 地址。
- 您可以指派給網路介面的 IP 地址數量上限會因執行個體類型而異。指派給網路介面的每個字首都計為一個 IP 地址。例如，c5.large 執行個體每個網路介面的 IPv4 地址數量上限為 10 個。此執行個體的每個網路介面都具備主要 IPv4 地址。如果網路介面沒有次要 IPv4 地址，則最多可以為網路介面指派 9 個字首。為網路介面多指派一個 IPv4 地址，可為網路介面指派的字首數量便少一個。如需詳細資訊，請參閱每個網路介面的 IP 位址上限。
- 來源/目標檢查中包含字首。
- 您必須使用字首設定使用網路介面的作業系統。注意下列事項：
  - 有些 Amazon Linux AMIs 包含由 安裝的其他指令碼 AWS，稱為 ec2-net-utils。這些指令碼會選擇性自動化您網路介面的組態。這些指令碼僅供 Amazon Linux 使用。
  - 如果您不是使用 Amazon Linux，則可以使用容器網路介面 (CNI) 作為 Kubernetes 外掛程式，或 dockerd (如果您使用 Docker 來管理容器)。

網路介面字首

當您將字首指派給網路介面時，可以選擇讓我們自動指派字首，還是指定自訂字首。如果您讓我們自動指派字首，且網路介面的子網路具有類型 prefix 的子網路 CIDR 預留，我們會從子網路 CIDR 預留中選取字首。否則，我們會從子網路 CIDR 範圍中進行選取。

任務

- 在建立網路介面期間指派字首

- 為現有的網路介面指派字首
- 從網路介面移除字首

在建立網路介面期間指派字首

您可以在建立網路介面時指派自動或自訂字首。

Console

在建立網路介面期間指派自動字首

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選擇 Create network interface (建立網路介面)。
4. 指定網路介面的描述，選取要在其中建立網路介面的子網路，然後設定私有 IPv4 和 IPv6 地址。
5. 展開 Advanced settings (進階設定)。
6. 對於 IPv4 字首委派，請執行下列其中一項操作：
   - 若要自動指派 IPv4 字首，選擇自動指派。然後對於 IPv4 字首的數目，指定要指派的字首數目。
   - 若要指派特定的 IPv4 字首，請選擇自訂。然後選擇新增字首，並輸入要使用的字首。
7. 對於 IPv6 字首委派，請執行下列其中一項操作：
   - 若要自動指派 IPv6 字首，選擇自動指派。然後對於 IPv6 字首的數目，指定要指派的字首數目。
   - 若要指派特定的 IPv6 字首，請選擇自訂。然後選擇新增字首，並輸入要使用的字首。

   > **Note**
   >
   > IPv6 prefix delegation (IPv6 字首委派) 僅在為 IPv6 啟用選定的子網才會出現。

8. 選取要與網路介面建立關聯的安全群組，並視需要指派資源標籤。
9. 選擇 Create network interface (建立網路介面)。

AWS CLI

在建立網路介面期間指派自動 IPv4 字首

使用 create-network-interface 命令，並將 --ipv4-prefix-count 設定為 AWS 要指派的 IPv4 字首數目。在下列範例中， AWS 會指派一個 IPv4 字首。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 automatic example" \
    --ipv4-prefix-count 1
```

在建立網路介面期間指派特定的 IPv4 字首

使用 create-network-interface 命令，並將 --ipv4-prefixes 設定為字首。 AWS 會從此範圍中選取 IPv4 地址。在下列範例中，字首 CIDR 是 10.0.0.208/28。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 manual example" \
    --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

在建立網路介面期間指派自動 IPv6 字首

使用 create-network-interface 命令，並將 --ipv6-prefix-count 設定為 AWS 要指派的 IPv6 字首數目。在下列範例中， AWS 會指派一個 IPv6 字首。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 automatic example" \
    --ipv6-prefix-count 1
```

在建立網路介面期間指派特定的 IPv6 字首

使用 create-network-interface 命令，並將 --ipv6-prefixes 設定為字首。 AWS 會從此範圍中選取 IPv6 地址。在下列範例中，字首 CIDR 是 2600:1f13:fc2:a700:1768::/80。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 manual example" \
    --ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80
```

PowerShell

在建立網路介面期間指派自動 IPv4 字首

使用 New-EC2NetworkInterface cmdlet，並將 Ipv4PrefixCount 設定為 AWS 要指派的 IPv4 字首數量。在下列範例中， AWS 會指派一個 IPv4 字首。

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 automatic example' `
    -Ipv4PrefixCount 1
```

在建立網路介面期間指派特定的 IPv4 字首

使用 New-EC2NetworkInterface cmdlet 並Ipv4Prefix設定為字首。從此範圍 AWS 中選取 IPv4 地址。在下列範例中，字首 CIDR 是 10.0.0.208/28。

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 manual example' `
    -Ipv4Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv4PrefixSpecificationRequest `
        -Property @{Ipv4Prefix = '10.0.0.208/28'})
```

在建立網路介面期間指派自動 IPv6 字首

使用 New-EC2NetworkInterface cmdlet，並將 Ipv6PrefixCount 設定為 AWS 要指派的 IPv6 字首數量。在下列範例中， AWS 會指派一個 IPv6 字首。

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 automatic example' `
    -Ipv6PrefixCount 1
```

在建立網路介面期間指派特定的 IPv6 字首

使用 New-EC2NetworkInterface cmdlet 並Ipv6Prefixes設定為字首。從此範圍 AWS 中選取 IPv6 地址。在下列範例中，字首 CIDR 是 2600:1f13:fc2:a700:1768::/80。

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 manual example' `
    -Ipv6Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv6PrefixSpecificationRequest `
```

```
    -Property @{Ipv6Prefix = '2600:1f13:fc2:a700:1768::/80'})
```

為現有的網路介面指派字首

您可以為現有網路介面指派自動或自訂字首。

Console

將自動字首指派給現有網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取要指派字首的網路介面，然後選取 Actions (動作)、Manage prefixes (管理字首)。
4. 對於 IPv4 字首委派，請執行下列其中一項操作：
   - 若要自動指派 IPv4 字首，選擇自動指派。然後對於 IPv4 字首的數目，指定要指派的字首數目。
   - 若要指派特定的 IPv4 字首，請選擇自訂。然後選擇新增字首，並輸入要使用的字首。
5. 對於 IPv6 字首委派，請執行下列其中一項操作：
   - 若要自動指派 IPv6 字首，選擇自動指派。然後對於 IPv6 字首的數目，指定要指派的字首數目。
   - 若要指派特定的 IPv6 字首，請選擇自訂。然後選擇新增字首，並輸入要使用的字首。

   > **Note**
   >
   > IPv6 prefix delegation (IPv6 字首委派) 僅在為 IPv6 啟用選定的子網才會出現。

6. 選擇儲存。

AWS CLI

使用 assign-ipv6-addresses 命令來指派 IPv6 字首，並使用 assign-private-ip-addresses 命令將 IPv4 字首指派給現有的網路介面。

將自動 IPv4 字首指派給現有網路介面

使用 assign-private-ip-addresses 命令，並將 `--ipv4-prefix-count` 設定為 AWS 要指派的 IPv4 字首數目。在下列範例中， AWS 會指派一個 IPv4 字首。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefix-count 1
```

若要將特定 IPv4 字首指派給現有的網路介面

使用 assign-private-ip-addresses 命令並將 `--ipv4-prefixes` 設定為字首。 AWS 會從此範圍選取 IPv4 位址。在下列範例中，字首 CIDR 是 10.0.0.208/28。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefixes 10.0.0.208/28
```

將自動 IPv6 字首指派給現有網路介面

使用 assign-ipv6-addresses 命令，並將 `--ipv6-prefix-count` 設定為 AWS 要指派的 IPv6 字首數目。在下列範例中， AWS 會指派一個 IPv6 字首。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefix-count 1
```

若要將特定 IPv6 字首指派給現有的網路介面

使用 assign-ipv6-addresses 命令並`--ipv6-prefixes`設定為字首。從此範圍 AWS 中選取 IPv6 地址。在下列範例中，字首 CIDR 是 2600:1f13:fc2:a700:18bb::/80。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

將自動 IPv4 字首指派給現有網路介面

使用 Register-EC2PrivateIpAddress cmdlet，並將 `Ipv4PrefixCount` 設定為 AWS 要指派的 IPv4 字首數目。在下列範例中， AWS 會指派一個 IPv4 字首。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
```

```
    -Ipv4PrefixCount 1
```

若要將特定 IPv4 字首指派給現有的網路介面

使用 Register-EC2PrivateIpAddress cmdlet 並Ipv4Prefix設定為字首。從此範圍 AWS 中選取 IPv4 地址。在下列範例中，字首 CIDR 是 10.0.0.208/28。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

將自動 IPv6 字首指派給現有網路介面

使用 Register-EC2Ipv6AddressList cmdlet，並將 Ipv6PrefixCount 設定為 AWS 要指派的 IPv4 字首數目。在下列範例中， AWS 會指派一個 IPv6 字首。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6PrefixCount 1
```

若要將特定 IPv6 字首指派給現有的網路介面

使用 Register-EC2Ipv6AddressList cmdlet 並Ipv6Prefix設定為字首。從此範圍 AWS 中選取 IPv6 地址。在下列範例中，字首 CIDR 是 2600:1f13:fc2:a700:18bb::/80。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

從網路介面移除字首

您可以從現有的網路介面移除字首。

Console

從網路介面移除字首

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面。

4. 選擇動作、管理字首。
5. 對於 IPv4 字首委派，若要移除特定字首，請選擇要移除字首旁的取消指派。若要移除所有字首，請選擇不指派。
6. 對於 IPv6 字首委派，若要移除特定字首，請選擇要移除字首旁的取消指派。若要移除所有字首，請選擇不指派。

> **Note**
>
> IPv6 prefix delegation (IPv6 字首委派) 僅在為 IPv6 啟用選定的子網才會出現。

7. 選擇儲存。

AWS CLI

您可以使用 unassign-ipv6-addresses 命令來移除 IPv6 字首，並使用 unassign-private-ip-address 命令，從現有的網路介面移除 IPv4 字首。

從網路介面移除 IPv4 字首

使用 unassign-private-ip-address 命令，並將 `--ipv4-prefix` 設定為字首 CIDR 以移除。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefixes 10.0.0.176/28
```

從網路介面移除 IPv6 字首

使用 unassign-ipv6-addresses 命令，並將 `--ipv6-prefix` 設定為字首 CIDR 以移除。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

從網路介面移除 IPv4 字首

使用 Unregister-EC2PrivateIpAddress cmdlet，並將 `Ipv4Prefix` 設定字首 CIDR 以移除。

```
Unregister-EC2PrivateIpAddress `
```

```
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

從網路介面移除 IPv6 字首

使用 Unregister-EC2Ipv6AddressList cmdlet，並將 Ipv6Prefix 設定字首 CIDR 以移除。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

刪除網路介面

刪除網路介面會釋放所有與介面關聯的屬性，並釋放其他執行個體要使用的所有私有 IP 位址或彈性 IP 位址。

您無法刪除正在使用的網路介面。首先，您必須分開網路介面。

Console

刪除網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面的核取方塊，然後選取 Actions (動作)、Delete (刪除)。
4. 出現確認提示時，請選擇 Delete (刪除)。

AWS CLI

若要刪除網路介面

使用下列 delete-network-interface 命令。

```
aws ec2 delete-network-interface --network-interface-id eni-1234567890abcdef0
```

PowerShell

若要刪除網路介面

使用 Remove-EC2NetworkInterface cmdlet。

```
Remove-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0
```

Amazon EC2 執行個體網路頻寬

執行個體頻寬規格適用於執行個體的傳入和傳出流量。舉例來說，如果某個執行個體指定的頻寬最多為 10 Gbps，這意味著傳入流量最高可達 10 Gbps 的頻寬，同時傳出流量最高可達 10 Gbps 的頻寬。EC2 執行個體可用的網路頻寬取決於數項因素，如下所示。

多流流量

當流量通過網際網路閘道或本機閘道時，多流量流量的可用頻寬取決於執行個體類型。

執行個體類型	可用的頻寬
少於 32 個 vCPUs執行個體類型	限制為 5 Gbps
超過 32 個 vCPUs執行個體類型	限制為執行個體類型可用頻寬的 50%
C8in、C8ine、M8in、M8ine、M8idn、R8in、R8idn 執行個體	僅限於執行個體類型的基準頻寬

單一流量

當執行個體不在相同的叢集置放群組中時，單一流量流量的頻寬限制為 5 Gbps。若要降低延遲並增加單一流量頻寬，請嘗試下列其中一項：

- 使用集群放置群組可為相同置放群組中的執行個體實現最高 10 Gbps 的頻寬。
- 在兩個端點之間設定多個路徑，以使用多重路徑 TCP (MPTCP) 達到更高的頻寬。
- 針對相同可用區域中的合格執行個體設定 ENA Express，以在這些執行個體之間達到最高 25 Gbps 的頻寬。

> **Note**
>
> 單一流程會視為唯一的 5 個元組 TCP 或 UDP 流程。對於 IP 標頭後面的其他通訊協定，例如 GRE 或 IPsec，會使用來源 IP、目的地 IP 和下一個通訊協定的 3 個元組來定義流程。

可用執行個體頻寬

執行個體的可用網路頻寬取決於其擁有的 vCPU 數量。例如，m5.8xlarge 執行個體具有 32 個 vCPU 和 10 Gbps 的網路頻寬，而 m5.16xlarge 執行個體具有 64 個 vCPU 和 20 Gbps 的網路頻寬。然而，如果執行個體超過執行個體層級的網路限額 (例如每秒封包數或追蹤連線數)，則執行個體可能無法達到此頻寬。流量可以使用多少可用頻寬取決於 vCPUs 數量和目的地。例如，m5.16xlarge執行個體有 64 vCPUs，因此區域中另一個執行個體的流量可以使用可用的完整頻寬 (20 Gbps)。不過，經過網際網路閘道或本機閘道的流量只能使用 50% 的可用頻寬 (10 Gbps)。

通常，具有 16 個或更少 vCPU 的執行個體 (大小為 4xlarge 和更小) 會被記錄為具有「高達」指定的頻寬。例如，「高達 10 Gbps」。這些執行個體具有基準頻寬。若要滿足其他需求，他們可使用網路輸入/輸出額度機制，以實現超越基準頻寬的高載頻寬。執行個體可以在有限的時間內使用高載頻寬 (通常是 5 到 60 分鐘)，具體取決於執行個體大小。

執行個體在啟動時會收到最大的網路輸入/輸出額度數。如果執行個體耗盡其網路輸入/輸出額度，則會返回其基準頻寬。每當執行中的執行個體使用的網路頻寬比其基準頻寬少，就會獲得網路輸入/輸出額度。已停止的執行個體不會獲得網路輸入/輸出額度。由於高載頻寬是共用資源，所以即使執行個體具有可用額度，執行個體高載仍只會是盡量實現。

傳入和傳出流量有單獨的網路 I/O 額度儲存貯體。

基本和高載網路效能

下列 Amazon EC2 執行個體類型指南說明每一個執行個體類型的網路效能，以及可使用高載頻寬的執行個體可用的基準網路頻寬。如需詳細資訊，請參閱下列內容：

- 網路規格 – 一般用途
- 網路規格 – 運算最佳化
- 網路規格 – 記憶體最佳化
- 網路規格 – 儲存最佳化
- 網路規格 – 加速運算
- 網路規格 – 高效能運算
- 網路規格 – 上一代

或者，您可以使用命令列工具來取得資訊。Amazon EC2 主控台不會顯示執行個體類型的基準網路頻寬。

AWS CLI

您可以使用 describe-instance-types 命令來顯示執行個體類型的相關資訊。下列範例會顯示所有 C5 執行個體的網路效能資訊。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].[InstanceType, NetworkInfo.NetworkPerformance,
 NetworkInfo.NetworkCards[0].BaselineBandwidthInGbps] | sort_by(@,&[2])" \
    --output table
```

以下為範例輸出。如果您的輸出缺少基準頻寬，請更新至最新版本的 AWS CLI。

```
-----------------------------------------------
|           DescribeInstanceTypes             |
+--------------+--------------------+---------+
|  c5.large    |  Up to 10 Gigabit  |  0.75   |
|  c5.xlarge   |  Up to 10 Gigabit  |  1.25   |
|  c5.2xlarge  |  Up to 10 Gigabit  |  2.5    |
|  c5.4xlarge  |  Up to 10 Gigabit  |  5.0    |
|  c5.9xlarge  |  12 Gigabit        |  12.0   |
|  c5.12xlarge |  12 Gigabit        |  12.0   |
|  c5.18xlarge |  25 Gigabit        |  25.0   |
|  c5.24xlarge |  25 Gigabit        |  25.0   |
|  c5.metal    |  25 Gigabit        |  25.0   |
+--------------+--------------------+---------+
```

PowerShell

您可以使用 Get-EC2InstanceType PowerShell 命令來顯示執行個體類型的相關資訊。下列範例會顯示所有 C5 執行個體的網路效能資訊。

```
Get-EC2InstanceType -Filter @{Name = "instance-type"; Values = "c5.*" } | `
    Select-Object `
    InstanceType,
    @{Name = 'NetworkPerformance'; Expression =
 {($_.Networkinfo.NetworkCards.NetworkPerformance)}},
    @{Name = 'BaselineBandwidthInGbps'; Expression =
 {($_.Networkinfo.NetworkCards.BaselineBandwidthInGbps)}} | `
Format-Table -AutoSize
```

以下為範例輸出。

```
InstanceType NetworkPerformance BaselineBandwidthInGbps
------------ ------------------ ----------------------
c5.4xlarge   Up to 10 Gigabit                    5.00
c5.xlarge    Up to 10 Gigabit                    1.25
c5.12xlarge  12 Gigabit                         12.00
c5.9xlarge   12 Gigabit                         12.00
c5.24xlarge  25 Gigabit                         25.00
c5.metal     25 Gigabit                         25.00
c5.2xlarge   Up to 10 Gigabit                    2.50
c5.large     Up to 10 Gigabit                    0.75
c5.18xlarge  25 Gigabit                         25.00
```

EC2 執行個體頻寬加權組態

有些執行個體類型支援可設定的頻寬加權，您可以在其中選取有利於網路處理或 EBS 操作的基準頻寬加權。基準頻寬的預設設定取決於執行個體類型。您可以在啟動期間設定頻寬加權，或使用下列加權偏好設定來修改執行個體設定：

- default – 此選項會針對執行個體類型使用標準頻寬組態。
- vpc-1 – 此選項會增加可用於聯網的基準頻寬，並減少 EBS 操作的基準頻寬。
- ebs-1 – 此選項會增加 EBS 操作可用的基準頻寬，並減少聯網的基準頻寬。

頻寬加權考量

以下是可能會影響頻寬加權策略的一些考量。

- 設定頻寬加權偏好設定只會影響頻寬規格。每秒網路封包數 (PPS) 和每秒讀寫次數 (IOPS) 規格不會變更。
- 聯網和 EBS 之間的合併頻寬規格不會變更。選取頻寬加權組態時，所選選項可用的基準頻寬會增加，其餘選項的基準頻寬則會減少相同的絕對量。對於 Flex 執行個體以外的所有執行個體，您所選選項的可用高載頻寬保持不變，其餘選項則會減少。對於最大 4xlarge 的 Flex 執行個體，高載頻寬保持不變。對於 8xlarge 和更大的 Flex 執行個體，高載頻寬的增加量與基準頻寬相同。
- 請務必了解頻寬配置的變更如何影響 EBS 的 I/O 效能。對於具有 `vpc-1` 組態 (網路頻寬增加) 的 EC2 執行個體，如果在達到 IOPS 限制之前達到 EBS 頻寬限制，則 EBS 磁碟區可能會遇到較低的 IOPS。這在較大的 I/O 大小中更為明顯。

例如，在通常支援 240,000 IOPS 且大小為 16 KiB I/O 的執行個體類型上，如果選取 vpc-1 加權，可能會因為調整後的 EBS 基準頻寬限制而降低可達到的 IOPS。

規劃工作負載時，請考慮 I/O 大小和模式。較小的 I/O 大小不太可能受到頻寬限制的影響，而較大的 I/O 大小或序列工作負載可能會因為頻寬變更而受到更多影響。請務必測試您的特定工作負載，以確保所選組態的最佳效能。

- 經過網際網路閘道或本機閘道之流量的聯網多流量頻寬規格會視情況調整為已設定選項基準頻寬的 50% 或 5 Gbps。如需詳細資訊，請參閱Amazon EC2 執行個體網路頻寬。

  下列範例是以預設基準頻寬為 40 Gbps 且預設邊界頻寬為 20 Gbps 的執行個體類型為基礎。如果選擇此執行個體的 vpc-1 頻寬加權，則加權基準頻寬會變更為 50 Gbps，邊界頻寬則會變更為 25 Gbps。
- 此功能適用於所有商業區域，並與 EC2 執行個體可用性和支援保持一致。
- 此功能不會對 EC2 執行個體增加額外費用。

頻寬加權支援的執行個體類型

下列執行個體系列中的執行個體類型支援可設定的頻寬加權。

- 一般用途：M8a、M8g、M8gd、M8i、M8id、M8i-flex、M9g、M9gd
- 運算最佳化：C8a、C8g、C8gd、C8i、C8id、C8i-flex、C9g、C9gd
- 記憶體最佳化：R8a、R8g、R8gd、R8i、R8id、R8i-flex、R9g、R9gd、X8g、X8aedz、X8i

檢查目前的頻寬設定

若要查看執行個體目前的頻寬設定，請選取其中一個索引標籤以取得指示。

Console

若要取得執行個體的頻寬設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 從清單中選擇要檢查的執行個體，然後導覽至聯網索引標籤。目前的設定會顯示在設定頻寬欄位中。如果頻寬未設定為特定值，Amazon EC2 會使用執行個體類型的預設設定。

AWS CLI

若要取得執行個體的頻寬設定

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].NetworkPerformanceOptions.BandwidthWeighting
 \
    --output text
```

以下為範例輸出。

```
default
```

此範例會針對較高的聯網頻寬，列出頻寬加權偏好設定設為 vpc-1 的所有執行個體。

```
aws ec2 describe-instances \
    --filters "Name=network-performance-options.bandwidth-weighting,Values=vpc-1" \
    --query Reservations[].Instances[].InstanceId \
    --output text
```

PowerShell

若要取得執行個體的頻寬設定

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.NetworkPerformanceOptions.BandwidthWeighting.Value
```

以下為範例輸出。

```
default
```

此範例會針對較高的聯網頻寬，列出頻寬加權偏好設定設為 vpc-1 的所有執行個體。

```
(Get-EC2Instance `
```

```
    -Filter @{Name="network-performance-options.bandwidth-
weighting";Values="vpc-1"}).Instances.InstanceId
```

設定執行個體的頻寬權重

可以在啟動時或透過從 EC2 主控台、API/SDK 或 CLI 修改現有執行個體來設定頻寬加權。

啟動執行個體時設定頻寬權重

若要在啟動執行個體時配置頻寬設定，請選取其中一個索引標籤以取得指示。

也可以在啟動範本中指定頻寬加權。若要建立啟動範本，請參閱 建立 Amazon EC2 啟動範本。要設定的參數所在位置與直接從主控台啟動執行個體時的位置相同。展開進階詳細資訊區段，並設定執行個體頻寬組態。

若要使用啟動範本啟動執行個體，請參閱 使用啟動範本啟動 EC2 執行個體。

Console

若要使用可設定的頻寬加權來啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選擇啟動執行個體。這會開啟啟動執行個體頁面。視您的偏好而定，有幾種其他方式可以進入啟動對話方塊。例如，可以直接從 AMI 或從 Amazon EC2 儀表板本身啟動執行個體。
4. 您從中進行啟動的 Amazon Machine Image (AMI) 必須基於 `Arm` 架構。許多 Quick Start 映像都支援 x86 和 `Arm`架構。選擇執行個體的作業系統後，請從架構清單中選取 `Arm` 選項。
5. 執行個體類型必須是此功能的其中一個 支援的執行個體類型。
6. 當您展開進階詳細資訊區段時，可以向下捲動以尋找執行個體頻寬組態設定。選取執行個體的頻寬組態選項。
7. 照常設定執行個體的所有其他設定，然後選擇啟動執行個體。

AWS CLI

若要使用可設定的頻寬加權來啟動執行個體

使用具有下列選項的 run-instances 命令，啟動專為更高網路頻寬加權而設定的執行個體。

```
--network-performance-options BandwidthWeighting=vpc-1
```

使用具有下列選項的 run-instances 命令，啟動專為更高 EBS 頻寬加權而設定的執行個體。

```
--network-performance-options BandwidthWeighting=ebs-1
```

PowerShell

若要使用可設定的頻寬加權來啟動執行個體

使用具有下列參數的 New-EC2Instance cmdlet，啟動專為更高網路頻寬加權而設定的執行個體。

```
-NetworkPerformanceOptions_BandwidthWeighting vpc-1
```

使用具有下列參數的 New-EC2Instance cmdlet，啟動專為更高 EBS 頻寬加權而設定的執行個體。

```
-NetworkPerformanceOptions_BandwidthWeighting ebs-1
```

更新現有執行個體的頻寬加權

若要更新現有執行個體的頻寬加權，執行個體必須處於 Stopped 狀態。

Console

若要更新頻寬加權

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 從清單中選取要更新的執行個體。
4. 變更頻寬組態之前，執行個體必須處於 Stopped 狀態。如果執行個體正在執行中，請從執行個體狀態功能表中選取停止執行個體。
5. 從動作 > 聯網功能表中選擇管理頻寬。這會開啟管理頻寬對話方塊。

> **Note**
>
> 如果執行個體類型不支援頻寬加權的組態，則會停用該功能表項目。

6. 選取該選項以更新執行個體，然後選擇變更以儲存設定。

AWS CLI

若要更新頻寬加權

使用 modify-instance-network-performance-options 命令，為指定的執行個體設定更高的網路頻寬加權。

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=vpc-1
```

下列範例會為指定的執行個體設定較高的 EBS 頻寬加權。

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=ebs-1
```

PowerShell

若要更新頻寬加權

使用 Edit-EC2InstanceNetworkPerformanceOption cmdlet，為指定的執行個體設定更高的網路頻寬加權。

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting vpc-1
```

下列範例會為指定的執行個體設定較高的 EBS 頻寬加權。

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting ebs-1
```

網路效能

下表顯示可透過 `default`、`vpc-1` 和 `ebs-1` 組態實現的聯網效能 (以 Gbps 為單位)。

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
c8a.medium	0.52 / 12.5	0.65 / 12.5	0.438 / 10.0
c8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
c8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8a.8xlarge	15.0	18.75	12.5
c8a.12xlarge	22.5	28.125	18.75
c8a.16xlarge	30.0	37.5	25.0
c8a.24xlarge	40.0	50.0	32.5
c8a.48xlarge	75.0	93.75	60.0
c8a.metal-24xl	40.0	50.0	32.5
c8a.metal-48xl	75.0	93.75	60.0
c8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
c8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8g.8xlarge	15.0	18.75	12.5
c8g.12xlarge	22.5	28.125	18.75

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
c8g.16xlarge	30.0	37.5	25.0
c8g.24xlarge	40.0	50.0	32.5
c8g.48xlarge	50.0	62.5	40.0
c8g.metal-24xl	40.0	50.0	32.5
c8g.metal-48xl	50.0	62.5	40.0
c8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
c8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8gd.8xlarge	15.0	18.75	12.5
c8gd.12xlarge	22.5	28.125	18.75
c8gd.16xlarge	30.0	37.5	25.0
c8gd.24xlarge	40.0	50.0	32.5
c8gd.48xlarge	50.0	62.5	40.0
c8gd.metal-24xl	40.0	50.0	32.5
c8gd.metal-48xl	50.0	62.5	40.0
c8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
c8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8i.8xlarge	15.0	18.75	12.5
c8i.12xlarge	22.5	28.125	18.75
c8i.16xlarge	30.0	37.5	25.0
c8i.24xlarge	40.0	50.0	32.5
c8i.32xlarge	50.0	62.5	40.0
c8i.48xlarge	75.0	93.75	60.0
c8i.96xlarge	100.0	125.0	80.0
c8i.metal-48xl	75.0	93.75	60.0
c8i.metal-96xl	100.0	125.0	80.0
c8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
c8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8id.8xlarge	15.0	18.75	12.5
c8id.12xlarge	22.5	28.125	18.75
c8id.16xlarge	30.0	37.5	25.0
c8id.24xlarge	40.0	50.0	32.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8id.32xlarge	50.0	62.5	40.0
c8id.48xlarge	75.0	93.75	60.0
c8id.96xlarge	100.0	125.0	80.0
c8id.metal-48xl	75.0	93.75	60.0
c8id.metal-96xl	100.0	125.0	80.0
c8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
c8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
c8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
c8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
c8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
c8i-flex.16xlarge	15.0 / 30.0	18.75 / 37.5	12.5 / 25.0
c9g.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
c9g.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
c9g.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
c9g.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0
c9g.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
c9g.8xlarge	17.0	21.25	14.0
c9g.12xlarge	25.0	31.25	20.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c9g.16xlarge	34.0	42.5	28.0
c9g.24xlarge	50.0	62.5	41.0
c9g.48xlarge	100.0	125.0	82.0
c9g.metal-48xl	100.0	125.0	82.0
c9gd.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
c9gd.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
c9gd.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
c9gd.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0
c9gd.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
c9gd.8xlarge	17.0	21.25	14.0
c9gd.12xlarge	25.0	31.25	20.5
c9gd.16xlarge	34.0	42.5	28.0
c9gd.24xlarge	50.0	62.5	41.0
c9gd.48xlarge	100.0	125.0	82.0
c9gd.metal-48xl	100.0	125.0	82.0
m8a.medium	0.52 / 12.5	0.65 / 12.5	0.438 / 10.0
m8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8a.8xlarge	15.0	18.75	12.5
m8a.12xlarge	22.5	28.125	18.75
m8a.16xlarge	30.0	37.5	25.0
m8a.24xlarge	40.0	50.0	32.5
m8a.48xlarge	75.0	93.75	60.0
m8a.metal-24xl	40.0	50.0	32.5
m8a.metal-48xl	75.0	93.75	60.0
m8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
m8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
m8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8g.8xlarge	15.0	18.75	12.5
m8g.12xlarge	22.5	28.125	18.75
m8g.16xlarge	30.0	37.5	25.0
m8g.24xlarge	40.0	50.0	32.5
m8g.48xlarge	50.0	62.5	40.0
m8g.metal-24xl	40.0	50.0	32.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8g.metal-48xl	50.0	62.5	40.0
m8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
m8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
m8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8gd.8xlarge	15.0	18.75	12.5
m8gd.12xlarge	22.5	28.125	18.75
m8gd.16xlarge	30.0	37.5	25.0
m8gd.24xlarge	40.0	50.0	32.5
m8gd.48xlarge	50.0	62.5	40.0
m8gd.metal-24xl	40.0	50.0	32.5
m8gd.metal-48xl	50.0	62.5	40.0
m8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8i.8xlarge	15.0	18.75	12.5
m8i.12xlarge	22.5	28.125	18.75

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8i.16xlarge	30.0	37.5	25.0
m8i.24xlarge	40.0	50.0	32.5
m8i.32xlarge	50.0	62.5	40.0
m8i.48xlarge	75.0	93.75	60.0
m8i.96xlarge	100.0	125.0	80.0
m8i.metal-48xl	75.0	93.75	60.0
m8i.metal-96xl	100.0	125.0	80.0
m8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8id.8xlarge	15.0	18.75	12.5
m8id.12xlarge	22.5	28.125	18.75
m8id.16xlarge	30.0	37.5	25.0
m8id.24xlarge	40.0	50.0	32.5
m8id.32xlarge	50.0	62.5	40.0
m8id.48xlarge	75.0	93.75	60.0
m8id.96xlarge	100.0	125.0	80.0
m8id.metal-48xl	75.0	93.75	60.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8id.metal-96xl	100.0	125.0	80.0
m8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
m8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
m8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
m8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
m8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
m8i-flex.16xlarge	15.0 / 30.0	18.75 / 37.5	12.5 / 25.0
m9g.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
m9g.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
m9g.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
m9g.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0
m9g.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
m9g.8xlarge	17.0	21.25	14.0
m9g.12xlarge	25.0	31.25	20.5
m9g.16xlarge	34.0	42.5	28.0
m9g.24xlarge	50.0	62.5	41.0
m9g.48xlarge	100.0	125.0	82.0
m9g.metal-48xl	100.0	125.0	82.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m9gd.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
m9gd.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
m9gd.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
m9gd.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0
m9gd.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
m9gd.8xlarge	17.0	21.25	14.0
m9gd.12xlarge	25.0	31.25	20.5
m9gd.16xlarge	34.0	42.5	28.0
m9gd.24xlarge	50.0	62.5	41.0
m9gd.48xlarge	100.0	125.0	82.0
m9gd.metal-48xl	100.0	125.0	82.0
r8a.medium	0.52 / 12.5	0.65 / 12.5	0.438 / 10.0
r8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8a.8xlarge	15.0	18.75	12.5
r8a.12xlarge	22.5	28.125	18.75
r8a.16xlarge	30.0	37.5	25.0

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
r8a.24xlarge	40.0	50.0	32.5
r8a.48xlarge	75.0	93.75	60.0
r8a.metal-24xl	40.0	50.0	32.5
r8a.metal-48xl	75.0	93.75	60.0
r8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
r8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8g.8xlarge	15.0	18.75	12.5
r8g.12xlarge	22.5	28.125	18.75
r8g.16xlarge	30.0	37.5	25.0
r8g.24xlarge	40.0	50.0	32.5
r8g.48xlarge	50.0	62.5	40.0
r8g.metal-24xl	40.0	50.0	32.5
r8g.metal-48xl	50.0	62.5	40.0
r8gd.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
r8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8gd.8xlarge	15.0	18.75	12.5
r8gd.12xlarge	22.5	28.125	18.75
r8gd.16xlarge	30.0	37.5	25.0
r8gd.24xlarge	40.0	50.0	32.5
r8gd.48xlarge	50.0	62.5	40.0
r8gd.metal-24xl	40.0	50.0	32.5
r8gd.metal-48xl	50.0	62.5	40.0
r8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8i.8xlarge	15.0	18.75	12.5
r8i.12xlarge	22.5	28.125	18.75
r8i.16xlarge	30.0	37.5	25.0
r8i.24xlarge	40.0	50.0	32.5
r8i.32xlarge	50.0	62.5	40.0
r8i.48xlarge	75.0	93.75	60.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8i.96xlarge	100.0	125.0	80.0
r8i.metal-48xl	75.0	93.75	60.0
r8i.metal-96xl	100.0	125.0	80.0
r8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8id.8xlarge	15.0	18.75	12.5
r8id.12xlarge	22.5	28.125	18.75
r8id.16xlarge	30.0	37.5	25.0
r8id.24xlarge	40.0	50.0	32.5
r8id.32xlarge	50.0	62.5	40.0
r8id.48xlarge	75.0	93.75	60.0
r8id.96xlarge	100.0	125.0	80.0
r8id.metal-48xl	75.0	93.75	60.0
r8id.metal-96xl	100.0	125.0	80.0
r8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
r8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
r8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
r8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
r8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
r8i-flex.16xlarge	15.0 / 30.0	18.75 / 37.5	12.5 / 25.0
r9g.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
r9g.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
r9g.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
r9g.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0
r9g.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
r9g.8xlarge	17.0	21.25	14.0
r9g.12xlarge	25.0	31.25	20.5
r9g.16xlarge	34.0	42.5	28.0
r9g.24xlarge	50.0	62.5	41.0
r9g.48xlarge	100.0	125.0	82.0
r9g.metal-48xl	100.0	125.0	82.0
r9gd.medium	0.55 / 15.0	0.688 / 15.0	0.455 / 12.0
r9gd.large	1.0 / 15.0	1.25 / 15.0	0.81 / 12.0
r9gd.xlarge	2.1 / 15.0	2.625 / 15.0	1.725 / 12.0
r9gd.2xlarge	4.25 / 17.0	5.313 / 17.0	3.5 / 14.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r9gd.4xlarge	8.5 / 17.0	10.625 / 17.0	7.0 / 14.0
r9gd.8xlarge	17.0	21.25	14.0
r9gd.12xlarge	25.0	31.25	20.5
r9gd.16xlarge	34.0	42.5	28.0
r9gd.24xlarge	50.0	62.5	41.0
r9gd.48xlarge	100.0	125.0	82.0
r9gd.metal-48xl	100.0	125.0	82.0
x8g.medium	0.52 / 12.5	0.65 / 12.5	0.441 / 10.0
x8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
x8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
x8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
x8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
x8g.8xlarge	15.0	18.75	12.5
x8g.12xlarge	22.5	28.125	18.75
x8g.16xlarge	30.0	37.5	25.0
x8g.24xlarge	40.0	50.0	32.5
x8g.48xlarge	50.0	62.5	40.0
x8g.metal-24xl	40.0	50.0	32.5
x8g.metal-48xl	50.0	62.5	40.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
x8aedz.large	1.562 / 18.75	1.953 / 18.75	1.249 / 15.0
x8aedz.xlarge	3.125 / 18.75	3.907 / 18.75	2.5 / 15.0
x8aedz.3xlarge	9.375 / 18.75	11.719 / 18.75	7.5 / 15.0
x8aedz.6xlarge	18.75	23.438	15.0
x8aedz.12xlarge	37.5	46.875	30.0
x8aedz.24xlarge	75.0	93.75	60.0
x8aedz.metal-12xl	37.5	46.875	30.0
x8aedz.metal-24xl	75.0	93.75	60.0
x8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
x8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
x8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
x8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
x8i.8xlarge	15.0	18.75	12.5
x8i.12xlarge	22.5	28.125	18.75
x8i.16xlarge	30.0	37.5	25.0
x8i.24xlarge	40.0	50.0	32.5
x8i.32xlarge	50.0	62.5	40.0
x8i.48xlarge	75.0	93.75	60.0
x8i.64xlarge	80.0	100.0	62.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
x8i.96xlarge	100.0	125.0	80.0
x8i.metal-48xl	75.0	93.75	60.0
x8i.metal-96xl	100.0	125.0	80.0

Amazon EBS 效能

下表顯示可透過 `default`、`vpc-1` 和 `ebs-1` 組態實現的 Amazon EBS 效能 (以 Gbps 為單位)。

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
c8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8a.8xlarge	10.0	6.25	12.5
c8a.12xlarge	15.0	9.375	18.75
c8a.16xlarge	20.0	12.5	25.0
c8a.24xlarge	30.0	20.0	37.5
c8a.48xlarge	60.0	41.25	75.0
c8a.metal-24xl	30.0	20.0	37.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8a.metal-48xl	60.0	41.25	75.0
c8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8g.8xlarge	10.0	6.25	12.5
c8g.12xlarge	15.0	9.375	18.75
c8g.16xlarge	20.0	12.5	25.0
c8g.24xlarge	30.0	20.0	37.5
c8g.48xlarge	40.0	27.5	50.0
c8g.metal-24xl	30.0	20.0	37.5
c8g.metal-48xl	40.0	27.5	50.0
c8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8gd.8xlarge	10.0	6.25	12.5

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
c8gd.12xlarge	15.0	9.375	18.75
c8gd.16xlarge	20.0	12.5	25.0
c8gd.24xlarge	30.0	20.0	37.5
c8gd.48xlarge	40.0	27.5	50.0
c8gd.metal-24xl	30.0	20.0	37.5
c8gd.metal-48xl	40.0	27.5	50.0
c8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8i.8xlarge	10.0	6.25	12.5
c8i.12xlarge	15.0	9.375	18.75
c8i.16xlarge	20.0	12.5	25.0
c8i.24xlarge	30.0	20.0	37.5
c8i.32xlarge	40.0	27.5	50.0
c8i.48xlarge	60.0	41.25	75.0
c8i.96xlarge	80.0	55.0	100.0
c8i.metal-48xl	60.0	41.25	75.0
c8i.metal-96xl	80.0	55.0	100.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8id.8xlarge	10.0	6.25	12.5
c8id.12xlarge	15.0	9.375	18.75
c8id.16xlarge	20.0	12.5	25.0
c8id.24xlarge	30.0	20.0	37.5
c8id.32xlarge	40.0	27.5	50.0
c8id.48xlarge	60.0	41.25	75.0
c8id.96xlarge	80.0	55.0	100.0
c8id.metal-48xl	60.0	41.25	75.0
c8id.metal-96xl	80.0	55.0	100.0
c8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
c8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
c8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
c8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
c8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
c9g.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
c9g.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
c9g.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0
c9g.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
c9g.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
c9g.8xlarge	12.0	7.75	15.0
c9g.12xlarge	18.0	11.75	22.5
c9g.16xlarge	24.0	15.5	30.0
c9g.24xlarge	36.0	23.5	45.0
c9g.48xlarge	72.0	47.0	90.0
c9g.metal-48xl	72.0	47.0	90.0
c9gd.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
c9gd.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
c9gd.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0
c9gd.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
c9gd.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
c9gd.8xlarge	12.0	7.75	15.0
c9gd.12xlarge	18.0	11.75	22.5

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
c9gd.16xlarge	24.0	15.5	30.0
c9gd.24xlarge	36.0	23.5	45.0
c9gd.48xlarge	72.0	47.0	90.0
c9gd.metal-48xl	72.0	47.0	90.0
m8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
m8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8a.8xlarge	10.0	6.25	12.5
m8a.12xlarge	15.0	9.375	18.75
m8a.16xlarge	20.0	12.5	25.0
m8a.24xlarge	30.0	20.0	37.5
m8a.48xlarge	60.0	41.25	75.0
m8a.metal-24xl	30.0	20.0	37.5
m8a.metal-48xl	60.0	41.25	75.0
m8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
m8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8g.8xlarge	10.0	6.25	12.5
m8g.12xlarge	15.0	9.375	18.75
m8g.16xlarge	20.0	12.5	25.0
m8g.24xlarge	30.0	20.0	37.5
m8g.48xlarge	40.0	27.5	50.0
m8g.metal-24xl	30.0	20.0	37.5
m8g.metal-48xl	40.0	27.5	50.0
m8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
m8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8gd.8xlarge	10.0	6.25	12.5
m8gd.12xlarge	15.0	9.375	18.75
m8gd.16xlarge	20.0	12.5	25.0
m8gd.24xlarge	30.0	20.0	37.5
m8gd.48xlarge	40.0	27.5	50.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8gd.metal-24xl	30.0	20.0	37.5
m8gd.metal-48xl	40.0	27.5	50.0
m8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8i.8xlarge	10.0	6.25	12.5
m8i.12xlarge	15.0	9.375	18.75
m8i.16xlarge	20.0	12.5	25.0
m8i.24xlarge	30.0	20.0	37.5
m8i.32xlarge	40.0	27.5	50.0
m8i.48xlarge	60.0	41.25	75.0
m8i.96xlarge	80.0	55.0	100.0
m8i.metal-48xl	60.0	41.25	75.0
m8i.metal-96xl	80.0	55.0	100.0
m8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m8id.8xlarge	10.0	6.25	12.5
m8id.12xlarge	15.0	9.375	18.75
m8id.16xlarge	20.0	12.5	25.0
m8id.24xlarge	30.0	20.0	37.5
m8id.32xlarge	40.0	27.5	50.0
m8id.48xlarge	60.0	41.25	75.0
m8id.96xlarge	80.0	55.0	100.0
m8id.metal-48xl	60.0	41.25	75.0
m8id.metal-96xl	80.0	55.0	100.0
m8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
m8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
m8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
m8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
m8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
m9g.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
m9g.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
m9g.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
m9g.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
m9g.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
m9g.8xlarge	12.0	7.75	15.0
m9g.12xlarge	18.0	11.75	22.5
m9g.16xlarge	24.0	15.5	30.0
m9g.24xlarge	36.0	23.5	45.0
m9g.48xlarge	72.0	47.0	90.0
m9g.metal-48xl	72.0	47.0	90.0
m9gd.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
m9gd.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
m9gd.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0
m9gd.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
m9gd.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
m9gd.8xlarge	12.0	7.75	15.0
m9gd.12xlarge	18.0	11.75	22.5
m9gd.16xlarge	24.0	15.5	30.0
m9gd.24xlarge	36.0	23.5	45.0
m9gd.48xlarge	72.0	47.0	90.0
m9gd.metal-48xl	72.0	47.0	90.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
r8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8a.8xlarge	10.0	6.25	12.5
r8a.12xlarge	15.0	9.375	18.75
r8a.16xlarge	20.0	12.5	25.0
r8a.24xlarge	30.0	20.0	37.5
r8a.48xlarge	60.0	41.25	75.0
r8a.metal-24xl	30.0	20.0	37.5
r8a.metal-48xl	60.0	41.25	75.0
r8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8g.8xlarge	10.0	6.25	12.5
r8g.12xlarge	15.0	9.375	18.75

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8g.16xlarge	20.0	12.5	25.0
r8g.24xlarge	30.0	20.0	37.5
r8g.48xlarge	40.0	27.5	50.0
r8g.metal-24xl	30.0	20.0	37.5
r8g.metal-48xl	40.0	27.5	50.0
r8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8gd.8xlarge	10.0	6.25	12.5
r8gd.12xlarge	15.0	9.375	18.75
r8gd.16xlarge	20.0	12.5	25.0
r8gd.24xlarge	30.0	20.0	37.5
r8gd.48xlarge	40.0	27.5	50.0
r8gd.metal-24xl	30.0	20.0	37.5
r8gd.metal-48xl	40.0	27.5	50.0
r8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8i.8xlarge	10.0	6.25	12.5
r8i.12xlarge	15.0	9.375	18.75
r8i.16xlarge	20.0	12.5	25.0
r8i.24xlarge	30.0	20.0	37.5
r8i.32xlarge	40.0	27.5	50.0
r8i.48xlarge	60.0	41.25	75.0
r8i.96xlarge	80.0	55.0	100.0
r8i.metal-48xl	60.0	41.25	75.0
r8i.metal-96xl	80.0	55.0	100.0
r8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8id.8xlarge	10.0	6.25	12.5
r8id.12xlarge	15.0	9.375	18.75
r8id.16xlarge	20.0	12.5	25.0
r8id.24xlarge	30.0	20.0	37.5

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
r8id.32xlarge	40.0	27.5	50.0
r8id.48xlarge	60.0	41.25	75.0
r8id.96xlarge	80.0	55.0	100.0
r8id.metal-48xl	60.0	41.25	75.0
r8id.metal-96xl	80.0	55.0	100.0
r8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
r8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
r8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
r8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
r8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
r9g.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
r9g.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
r9g.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0
r9g.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
r9g.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
r9g.8xlarge	12.0	7.75	15.0
r9g.12xlarge	18.0	11.75	22.5

執行個體類型	default (基準/高載)	vpc-1 (基準/高載)	ebs-1 (基準/高載)
r9g.16xlarge	24.0	15.5	30.0
r9g.24xlarge	36.0	23.5	45.0
r9g.48xlarge	72.0	47.0	90.0
r9g.metal-48xl	72.0	47.0	90.0
r9gd.medium	0.38 / 12.0	0.242 / 7.75	0.475 / 12.0
r9gd.large	0.76 / 12.0	0.51 / 7.75	0.95 / 12.0
r9gd.xlarge	1.5 / 12.0	0.975 / 7.75	1.875 / 12.0
r9gd.2xlarge	3.0 / 12.0	1.937 / 7.75	3.75 / 12.0
r9gd.4xlarge	6.0 / 12.0	3.875 / 7.75	7.5 / 12.0
r9gd.8xlarge	12.0	7.75	15.0
r9gd.12xlarge	18.0	11.75	22.5
r9gd.16xlarge	24.0	15.5	30.0
r9gd.24xlarge	36.0	23.5	45.0
r9gd.48xlarge	72.0	47.0	90.0
r9gd.metal-48xl	72.0	47.0	90.0
x8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
x8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
x8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
x8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
x8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
x8g.8xlarge	10.0	6.25	12.5
x8g.12xlarge	15.0	9.375	18.75
x8g.16xlarge	20.0	12.5	25.0
x8g.24xlarge	30.0	20.0	37.5
x8g.48xlarge	40.0	27.5	50.0
x8g.metal-24xl	30.0	20.0	37.5
x8g.metal-48xl	40.0	27.5	50.0
x8aedz.large	1.25 / 15.0	0.859 / 10.312	1.563 / 15.0
x8aedz.xlarge	2.5 / 15.0	1.718 / 10.312	3.125 / 15.0
x8aedz.3xlarge	7.5 / 15.0	5.156 / 10.312	9.375 / 15.0
x8aedz.6xlarge	15.0	10.312	18.75
x8aedz.12xlarge	30.0	20.625	37.5
x8aedz.24xlarge	60.0	41.25	75.0
x8aedz.metal-12xl	30.0	20.625	37.5
x8aedz.metal-24xl	60.0	41.25	75.0
x8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
x8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
x8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0

執行個體類型	**default** (基準/高載)	**vpc-1** (基準/高載)	**ebs-1** (基準/高載)
x8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
x8i.8xlarge	10.0	6.25	12.5
x8i.12xlarge	15.0	9.375	18.75
x8i.16xlarge	20.0	12.5	25.0
x8i.24xlarge	30.0	20.0	37.5
x8i.32xlarge	40.0	27.5	50.0
x8i.48xlarge	60.0	41.25	75.0
x8i.64xlarge	70.0	50.0	87.5
x8i.96xlarge	80.0	55.0	100.0
x8i.metal-48xl	60.0	41.25	75.0
x8i.metal-96xl	80.0	55.0	100.0

監控執行個體頻寬

您可以使用 CloudWatch 指標來監控執行個體的網路頻寬，以及傳送和接收的封包。當流量超過 Amazon EC2 在執行個體層級定義的網路允許時，可以使用彈性網路轉接器 (ENA) 驅動程式提供的網路效能指標進行監控。

您可以設定 Amazon EC2 是否每一分鐘或每五分鐘將執行個體的指標資料傳送到 CloudWatch。網路效能指標可能會顯示超過允許值並且捨棄封包，而 CloudWatch 執行個體指標則不會顯示。當執行個體對網路資源的需求短暫突增 (稱為微爆量)，但 CloudWatch 指標不足以反映這些微秒級峰值時，就會發生這種情況。

進一步了解

- 執行個體指標

- 監控網路效能

增強 Amazon EC2 執行個體的聯網能力

增強型聯網使用單一根目錄 I/O 虛擬化 (SR-IOV) 在支援的執行個體類型上提供高效能聯網功能。SR-IOV 是一種相較於傳統虛擬網路介面可提高 I/O 效能及降低 CPU 使用率的裝置虛擬化方式。增強聯網提供更高的頻寬、更高的每秒封包數 (PPS) 效能，以及一致較低的執行個體間延遲。使用增強型聯網無需額外收費。

如需每種執行個體類型的支援網路速度資訊，請參閱 Amazon EC2 執行個體類型。

您可以使用下列其中一種機制來啟用增強型網路：

彈性網路轉接器 (ENA)

彈性網路轉接器 (ENA) 在支援的執行個體類型上支援高達 100 Gbps 的網路速度。

所有以 Nitro 系統為基礎的執行個體均使用 ENA 以提供增強型網路功能。此外，下列 Xen 型執行個體使用 ENA：H1、I3、G3、`m4.16xlarge`、P3、P3dn 和 R4。

如需詳細資訊，請參閱在 EC2 執行個體上使用 ENA 啟用增強型網路。

Intel 82599 虛擬功能 (VF) 介面

Intel 82599 虛擬功能介面在支援的執行個體類型上支援高達 10 Gbps 的網路速度。

下列執行個體類型使用 Intel 82599 VF 介面來增強網路功能：C3、C4、D2、I2、M4 (不包括 m4.16xlarge) 和 R3。

如需詳細資訊，請參閱已使用 Intel 82599 VF 介面來強化網路。。

目錄

- 在 EC2 執行個體上使用 ENA 啟用增強型網路
- 使用 ENA Express 提高 EC2 執行個體之間的網路效能
- 已使用 Intel 82599 VF 介面來強化網路。
- 監控 EC2 執行個體的 ENA 設定網路效能
- 改善基於 Linux 的 EC2 執行個體的網路延遲
- Nitro 系統效能調校的考量因素
- 在 EC2 Windows 執行個體上最佳化網路效能

在 EC2 執行個體上使用 ENA 啟用增強型網路

Amazon EC2 會透過彈性網路介面卡 (ENA) 提供增強型聯網功能。若要使用增強型聯網，您必須使用包含所需 ENA 驅動程式的 AMI，或手動安裝它。然後，您可以在執行個體上啟用 ENA 支援。

若要檢閱 ENA 驅動程式的版本備註或安裝指示，請參閱符合您執行個體作業系統平台的標籤。

Linux

您可以在 GitHub 上檢閱下列文件：

- 在 GitHub 上檢閱 ENA Linux 核心驅動程式版本備註。
- 如需 ENA Linux 核心驅動程式的概述，包括安裝說明，請參閱 GitHub 上彈性網路介面卡 (ENA) 系列的 Linux 核心驅動程式。

Windows

您可以從本指南的管理裝置驅動程式區段檢閱下列文件：

- 追蹤 ENA Windows 驅動程式版本版本.
- 在 EC2 Windows 執行個體上安裝 ENA 驅動程式.

對於 Nitro 型執行個體，增強型聯網功能會因執行個體類型實作的 Nitro 版本而異。

若要檢閱執行個體的網路規格，請選擇執行個體類型的執行個體系列連結。如果您不確定適用的執行個體系列，請參閱「Amazon EC2 執行個體類型指南」中的命名慣例。

- 加速運算執行個體的網路規格
- 運算最佳化執行個體的網路規格
- 一般用途執行個體的網路規格
- 高效能運算執行個體的網路規格
- 記憶體最佳化的執行個體的網路規格
- 儲存最佳化執行個體的網路規格

目錄

- 使用 ENA 增強型聯網的先決條件
- 測試是否已啟用增強型網路

- 在執行個體上啟用增強型網路
- ENA 佇列
- 對 Linux 上的 ENA 核心驅動程式進行疑難排解
- 為彈性網路轉接器 Windows 驅動程式進行疑難排解

使用 ENA 增強型聯網的先決條件

若要準備使用 ENA 介面的增強型聯網，請依下列方式設定您的執行個體：

- 啟動 Nitro 型執行個體。
- 確定該執行個體具有網際網路連線能力。
- 如果您的執行個體上有想要保留的重要資料，您現在應從執行個體建立 AMI，以備份這些資料。更新 ENA 核心驅動程式並啟用 `enaSupport` 屬性，可能會導致無法存取不相容的執行個體或作業系統。如果您有較新的備份，發生這些狀況時資料便能獲得保留。
- Linux 執行個體：使用支援版本的 Linux 核心和支援的發行版本，啟動執行個體，即可讓執行個體自動啟用 ENA 增強型聯網。如需詳細資訊，請參閱 ENA Linux 核心驅動程式版本備註。
- Windows 執行個體 – 如果執行個體正在執行 Windows Server 2008 R2 SP1，請確保其具有 SHA-2 程式碼簽署支援更新。
- AWS CloudShell 從 使用 AWS 管理主控台，或在您選擇的AWS Tools for Windows PowerShell任何電腦上安裝和設定 AWS CLI或 ，最好是本機桌上型電腦或筆記型電腦。如需詳細資訊，請參閱 存取 Amazon EC2 和 AWS CloudShell 使用者指南。增強型聯網無法從 Amazon EC2 主控台管理。

測試是否已啟用增強型網路

您可以測試您的執行個體或 AMI 中是否已啟用增強型聯網。

執行個體屬性

檢查 `enaSupport` 執行個體屬性的值。

AWS CLI

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].EnaSupport"
```

若已啟用增強型聯網，則輸出如下所示。

```
[
    true
]
```

PowerShell

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

若已啟用增強型聯網，則輸出如下所示。

```
True
```

映像屬性

檢查 enaSupport 影像屬性的值。

AWS CLI

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query "Images[].EnaSupport"
```

若已啟用增強型聯網，則輸出如下所示。

```
[
    true
]
```

PowerShell

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).EnaSupport
```

若已啟用增強型聯網，則輸出如下所示。

```
True
```

Linux 介面驅動程式

請用下列命令確認 ena 核心驅動程式用於特定介面，並將介面名稱換成您想要檢查的介面。如果您使用單一介面 (預設)，則這會是 eth0。如果您的 Linux 發行版本支援可預測的網路名稱，其名稱可能類似於 ens5。如需詳細資訊，請展開 在執行個體上啟用增強型網路 中 RHEL、SUSE 和 CentOS 的區段。

在下列範例中，不會載入 ena 核心驅動程式，因為列出的驅動程式為 vif。

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

在此範例中，ena 核心驅動程式已載入且為最低建議版本。此執行個體已正確設定增強型網路。

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

在執行個體上啟用增強型網路

您使用的程序會依執行個體的作業系統而定。

Amazon Linux

Amazon Linux 的 AMI 包含已安裝 ENA 之增強型聯網所需的核心驅動程式，並且已啟用 ENA 支援。因此，如果您在支援的執行個體類型上使用 Amazon Linux HVM 版本啟動執行個體，則您的執行個體已啟用增強型聯網。如需詳細資訊，請參閱測試是否已啟用增強型網路。

Ubuntu

最新的 Ubuntu HVM AMI 包含已安裝 ENA 之增強型聯網所需的核心驅動程式，並且已啟用 ENA 支援。因此，如果您在支援的執行個體類型上使用最新的 Ubuntu HVM AMI 來啟動執行個體，則您的執行個體已啟用增強型聯網。如需詳細資訊，請參閱 測試是否已啟用增強型網路。

如果您使用較舊的 AMI 啟動執行個體，且其尚未啟用增強型聯網，您可以安裝 `linux-aws` 核心套件以取得最新的增強型聯網驅動程式，並更新必要的屬性。

安裝 **`linux-aws`** 核心套件 (Ubuntu 16.04 或更新版本)

Ubuntu 16.04 和 18.04 隨附 Ubuntu 自訂核心 (`linux-aws` 核心套件)。若要使用不同的核心，請聯絡支援。

安裝 **`linux-aws`** 核心套件 (Ubuntu Trusty 14.04)

1. 連線到您的 執行個體。
2. 更新套件快取和套件。

   ```
   ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
   ```

   > **Important**
   >
   > 如果更新程序期間提示您安裝 `grub`，請用 `/dev/xvda` 安裝 `grub`，然後選擇保留目前版本的 `/boot/grub/menu.lst`。

3. [EBS 支援的執行個體] 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來停止執行個體：stop-instances (AWS CLI) 或 Stop-EC2Instance (AWS Tools for Windows PowerShell)。

   [執行個體存放區後端執行個體] 您無法停止執行個體來修改屬性。請改為執行此程序：在 Ubuntu 上啟用增強型聯網 (執行個體後端執行個體)。
4. 在本機電腦上使用下列其中一個命令啟用增強型聯網屬性。

   - modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
```

- Edit-EC2InstanceAttribute (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
```

5. (選擇性) 從執行個體建立 AMI，如 建立 Amazon EBS 支援的 AMI 所述。AMI 將從執行個體繼承增強型聯網的 `enaSupport` 屬性。因此，您預設可用此 AMI 啟動其他啟用增強型聯網的執行個體。
6. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。

在 Ubuntu 上啟用增強型聯網 (執行個體後端執行個體)

遵循先前的程序，一直進行到您停止執行個體的那個步驟。如 建立 Amazon S3 支援的 AMI 中所述建立新的 AMI，且務必在註冊 AMI 時啟用增強型聯網屬性。

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

RHEL、SUSE、CentOS

適用於 Red Hat Enterprise Linux 、SUSE Linux Enterprise Server 以及 CentOS 最新 AMI 包含使用 ENA 之增強型聯網所需的核心驅動程式，並且已啟用 ENA 支援。因此，如果您在支援的執行個體類型上使用最新的 AMI 來啟動執行個體，則您的執行個體已啟用增強型聯網。如需詳細資訊，請參閱 測試是否已啟用增強型網路。

以下程序提供在 Amazon Linux AMI 或 Ubuntu 以外的其他 Linux 發行版本上啟用增強型聯網的一般步驟。如需像是詳細命令語法、檔案位置或套件和工具支援等詳細資訊，請參閱 Linux 發行版本的文件。

在 Linux 上啟用增強型聯網

1. 連線到您的執行個體。

2. 在您的執行個體上從 GitHub (網址為 https://github.com/amzn/amzn-drivers) 中複製 ena 核心驅動程式的原始程式碼。(SUSE Linux Enterprise Server 12 SP2 和更新版本預設包含 ENA 2.02，因此您不需要下載和編譯 ENA 驅動程式。若為 SUSE Linux Enterprise Server 12 SP2 和更新版本，您應該提出請求，將您想要的驅動程式版本新增至庫存核心)。

```
git clone https://github.com/amzn/amzn-drivers
```

3. 在您的執行個體上編譯及安裝 ena 核心驅動程式。這些步驟視 Linux 發行版本而定。如需在 Red Hat Enterprise Linux 上編譯核心驅動程式的詳細資訊，請參閱如何在執行 RHEL 的 Amazon EC2 執行個體上安裝最新的 ENS 驅動程式，以取得增強型網路支援？

4. 執行 sudo depmod 命令更新核心驅動程式相依性。

5. 在執行個體上更新 initramfs，確定開機時載入新核心驅動程式。例如，如果您的發行版本支援 dracut，您可以使用下列命令。

```
dracut -f -v
```

6. 判斷系統是否依預設使用可預測的網路介面名稱。使用 systemd 或 udev 197 或以上版本的系統可重新命名乙太網路裝置，但不保證單一網路介面會命名為 eth0。該行為會造成連線至執行個體時發生問題。如需詳細資訊及查看其他的組態選項，請參閱 freedesktop.org 網站上的可預測的網路介面名稱。

   a. 您可用下列命令在 RPM 系統上檢查 systemd 或 udev 的版本。

   ```
   rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
   systemd-208-11.el7_0.2.x86_64
   ```

   在此 Red Hat Enterprise Linux 7 範例中，systemd版本為 208，因此必須停用可預測的網路介面名稱。

   b. 若要停用可預測網路介面名稱，請將 net.ifnames=0 選項新增至 GRUB_CMDLINE_LINUX 的 /etc/default/grub 行中。

   ```
   sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /etc/default/
   grub
   ```

   c. 重建 grub 組態檔案。

   ```
   sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

7. [EBS 支援的執行個體] 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來停止執行個體：stop-instances (AWS CLI)、Stop-EC2Instance (AWS Tools for Windows PowerShell)。

   [執行個體存放區後端執行個體] 您無法停止執行個體來修改屬性。請改為執行此程序：在 Linux 上啟用增強型聯網 (執行個體存放區後端執行個體)。

8. 在本機電腦上使用下列其中一個命令啟用增強型聯網的 `enaSupport` 屬性：

   - modify-instance-attribute (AWS CLI)

     ```
     aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
     ```

   - Edit-EC2InstanceAttribute (Tools for Windows PowerShell)

     ```
     Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
     ```

9. (選擇性) 從執行個體建立 AMI，如 建立 Amazon EBS 支援的 AMI 所述。AMI 將從執行個體繼承增強型聯網的 `enaSupport` 屬性。因此，您預設可用此 AMI 啟動其他啟用增強型聯網的執行個體。

   如果您執行個體的作業系統包含 `/etc/udev/rules.d/70-persistent-net.rules` 檔案，您必須先刪除檔案，然後再建立 AMI。此檔案包含原始執行個體乙太網路卡的 MAC 地址。如果其他執行個體使用此檔案開機，作業系統將無法找到設備，且 `eth0` 可能會失敗，因而造成開機問題。此檔案會在下次開機週期時重新產生，且從 AMI 啟動的任何執行個體將會建立自己的檔案版本。

10. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。
11. (選擇性) 連線到執行個體，並確認核心驅動程式已安裝。

    如果啟用增強型聯網後無法連接至您的執行個體，請參閱 對 Linux 上的 ENA 核心驅動程式進行疑難排解。

在 Linux 上啟用增強型聯網 (執行個體存放區後端執行個體)

遵循先前的程序，一直進行到您停止執行個體的那個步驟。如 建立 Amazon S3 支援的 AMI 中所述建立新的 AMI，且務必在註冊 AMI 時啟用增強型聯網屬性。

- register-image (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport ...
```

Ubuntu 搭配 DKMS

此方法僅用於測試和意見回饋用途。不適用於生產部署。如需生產部署，請參閱Ubuntu。

> **Important**
>
> 使用 DKMS 會使訂閱的支持協議無效。它不應該用於生產部署。

在 Ubuntu 上使用 ENA 啟用增強型聯網 (EBS 後端執行個體)

1. 請遵循Ubuntu中的步驟 1 和 2。
2. 安裝 `build-essential` 套件，以編譯核心驅動程式和 `dkms` 套件，使 `ena` 核心驅動程式在每次核心更新時重建。

   ```
   ubuntu:~$ sudo apt-get install -y build-essential dkms
   ```

3. 在您的執行個體上從 GitHub (網址為 https://github.com/amzn/amzn-drivers) 中複製 `ena` 核心驅動程式的原始碼。

   ```
   ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
   ```

4. 將 `amzn-drivers` 套件移至 `/usr/src/` 目錄，讓 DKMS 可以找到套件並在每次核心更新時建置。附加原始碼版本編號 (目前的版本編號可在版本備註中找到) 到目錄名稱。例如，下列範例顯示 `1.0.0` 版本。

   ```
   ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
   ```

5. 建立包含下列值的 DKMS 組態檔案，並換成您的 `ena` 版本。

   建立檔案。

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

編輯檔案並新增下列值。

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

6. 使用 DKMS 在您的執行個體上新增、建置及安裝 ena 核心驅動程式。

   將核心驅動程式新增至 DKMS。

   ```
   ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
   ```

   使用 dkms 命令建置核心驅動程式。

   ```
   ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
   ```

   使用 dkms 安裝核心驅動程式。

   ```
   ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
   ```

7. 重建 `initramfs`，以在開機時載入正確的核心驅動程式。

   ```
   ubuntu:~$ sudo update-initramfs -u -k all
   ```

8. 使用 測試是否已啟用增強型網路 中的 modinfo ena 命令確認 ena 核心驅動程式已安裝。

   ```
   ubuntu:~$ modinfo ena
   filename:    /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
   version:  1.0.0
   license:  GPL
   description: Elastic Network Adapter (ENA)
   ```

```
author:   Amazon.com, Inc. or its affiliates
srcversion:  9693C876C54CA64AE48F0CA
alias:    pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:    3.13.0-74-generic SMP mod_unload modversions
parm:     debug:Debug level (0=none,...,16=all) (int)
parm:     push_mode:Descriptor / header push mode (0=automatic,1=disable,3=enable)
    0 - Automatically choose according to device capability (default)
    1 - Don't push anything to device memory
    3 - Push descriptors and header buffer to device memory (int)
parm:     enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1) (int)
parm:     enable_missing_tx_detection:Enable missing Tx completions. (default=1)
 (int)
parm:     numa_node_override_array:Numa node override map
(array of int)
parm:     numa_node_override:Enable/Disable numa node override (0=disable)
(int)
```

9. 繼續執行Ubuntu中的步驟 3。

啟用 Windows 上的增強型網路

如果您啟動執行個體，且其尚未啟用增強型聯網，您必須在執行個體上下載並安裝必要的網路轉接器驅動程式，然後設定 enaSupport 執行個體屬性，以啟動增強型聯網。

啟用增強型聯網

1. 連線到您的執行個體，並以本機管理員登入。
2. [僅限 Windows Server 2016 和 2019 版] 安裝驅動程式後，執行下列 EC2Launch PowerShell 指令碼，以設定執行個體。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

3. 在執行個體上，安裝驅動程式，如下所示：
   a. 將最新的驅動程式下載到執行個體。
   b. 將 zip 封存檔解壓縮。

c. 執行 `install.ps1` PowerShell script 來安裝驅動程式。

> **Note**
>
> 如果您收到執行政策錯誤，請將政策設定為 `Unrestricted` (預設設定為 `Restricted` 或 `RemoteSigned`)。在命令列中，執行 `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`，然後再次執行 `install.ps1` PowerShell 指令碼。

4. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來停止執行個體：stop-instances (AWS CLI) 或 Stop-EC2Instance) (AWS Tools for Windows PowerShell)。
5. 在執行個體上啟用 ENA 支援，如下所示：

   a. 在本機電腦上，執行下列其中一個命令，查看執行個體上的 EC2 執行個體 ENA 支援屬性。如果此屬性未啟用，輸出會是「[]」或空白。`EnaSupport` 預設會設定為 `false`。

   - describe-instances (AWS CLI)

   ```
   aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
    "Reservations[].Instances[].EnaSupport"
   ```

   - Get-EC2Instance (Tools for Windows PowerShell)

   ```
   (Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
   ```

   b. 若要啟用 ENA 支援，請執行下列其中一項命令：

   - modify-instance-attribute (AWS CLI)

   ```
   aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-
   support
   ```

   - Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

   ```
   Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
   ```

   若重新啟動執行個體時發生問題，您也可使用下列其中一項命令來停用 ENA 支援。

   - modify-instance-attribute (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --no-ena-support
```

- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $false
```

c. 如先前所示，使用 `true` 或 describe-instances，確認屬性已設定為 Get-EC2Instance。您現在應該會看到下列輸出：

```
[
 true
]
```

6. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。
7. 在執行個體上，確認 ENA 驅動程式已安裝並啟用，如下所示：
   a. 開啟網路圖示的內容（按一下滑鼠右鍵）選單，然後選擇開啟網路和共用中心。
   b. 選擇乙太網路卡 (例如，Ethernet 2)。
   c. 請選擇 Details (詳細資訊)。針對 Network Connection Details (網路連線詳細資料)，檢查 Description (描述) 是否為 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡)。
8. (選擇性) 從執行個體建立 AMI。AMI 將從執行個體繼承 `enaSupport` 屬性。因此，您預設可用此 AMI 啟動其他啟用 ENA 的執行個體。

ENA 佇列

彈性網路轉接器 (ENA) 佇列會處理 Amazon EC2 網路介面的封包處理。每個執行個體類型都有跨其彈性網路界面 (ENIs) 共用的一組 ENA 佇列數量。根據預設，每個 ENI 都會根據執行個體類型和大小取得固定佇列計數。

透過可設定的 ENA 佇列配置，您可以選擇每個 ENI 取得的佇列數量。每個 ENI 和每個執行個體的總計仍為上限。這可協助您將封包處理與工作負載配對。例如，您可以為具有大量流量的 ENI 提供更多佇列，並為具有少量流量的 ENI 提供更多佇列。

主題

- EC2 執行個體類型的 ENA 佇列規格

- 設定 ENA 佇列配置

EC2 執行個體類型的 ENA 佇列規格

下表列出支援可設定 ENA 佇列配置的執行個體類型，以及其預設和最大佇列值。

> **Note**
>
> 如需所有執行個體類型的 ENA 佇列值，包括不支援可設定 ENA 佇列配置的類型，請參閱 Amazon EC2 執行個體類型指南中的網路規格表。

General purpose

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m6i.large	2	2	6
m6i.xlarge	4	4	16
m6i.2xlarge	8	8	32
m6i.4xlarge	8	16	64
m6i.8xlarge	8	32	64
m6i.12xlarge	8	32	64
m6i.16xlarge	8	32	120
m6i.24xlarge	8	32	120
m6i.32xlarge	8	32	120
m6id.large	2	2	6
m6id.xlarge	4	4	16

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m6id.2xlarge	8	8	32
m6id.4xlarge	8	16	64
m6id.8xlarge	8	32	64
m6id.12xlarge	8	32	64
m6id.16xlarge	8	32	120
m6id.24xlarge	8	32	120
m6id.32xlarge	8	32	120
m6idn.large	2	2	6
m6idn.xlarge	4	4	16
m6idn.2xlarge	8	8	32
m6idn.4xlarge	8	16	64
m6idn.8xlarge	16	32	128
m6idn.12xlarge	16	32	128
m6idn.16xlarge	16	32	240
m6idn.24xlarge	32	32	480
m6idn.32xlarge	32	32	512 *
m6in.large	2	2	6
m6in.xlarge	4	4	16
m6in.2xlarge	8	8	32

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m6in.4xlarge	8	16	64
m6in.8xlarge	16	32	128
m6in.12xlarge	16	32	128
m6in.16xlarge	16	32	240
m6in.24xlarge	32	32	480
m6in.32xlarge	32	32	512 *
m8a.medium	1	1	3
m8a.large	2	2	6
m8a.xlarge	4	4	16
m8a.2xlarge	8	8	32
m8a.4xlarge	8	16	64
m8a.8xlarge	8	32	128
m8a.12xlarge	16	64	192
m8a.16xlarge	16	64	256
m8a.24xlarge	16	128	384
m8a.48xlarge	32	128	768
m8a.metal-24xl	16	128	384
m8a.metal-48xl	32	128	768
m8azn.medium	1	1	3

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8azn.large	2	2	8
m8azn.xlarge	4	4	16
m8azn.3xlarge	4	16	48
m8azn.6xlarge	8	32	96
m8azn.12xlarge	8	64	192
m8azn.24xlarge	16	128	384
m8azn.metal-12xl	8	64	192
m8azn.metal-24xl	16	128	384
m8gb.medium	1	1	2
m8gb.large	2	2	6
m8gb.xlarge	4	4	16
m8gb.2xlarge	8	8	32
m8gb.4xlarge	8	16	64
m8gb.8xlarge	8	32	128
m8gb.12xlarge	16	64	192
m8gb.16xlarge	16	64	256
m8gb.24xlarge	16	128	384
m8gb.48xlarge	32	128	768 *
m8gb.metal-24xl	32	128	768

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8gb.metal-48xl	32	128	768 *
m8gn.medium	1	1	2
m8gn.large	2	2	6
m8gn.xlarge	4	4	16
m8gn.2xlarge	8	8	32
m8gn.4xlarge	8	16	64
m8gn.8xlarge	8	32	128
m8gn.12xlarge	16	64	192
m8gn.16xlarge	16	64	256
m8gn.24xlarge	16	128	384
m8gn.48xlarge	32	128	768 *
m8gn.metal-24xl	32	128	768
m8gn.metal-48xl	32	128	768 *
m8i.large	2	2	6
m8i.xlarge	4	4	16
m8i.2xlarge	8	8	32
m8i.4xlarge	8	16	64
m8i.8xlarge	8	32	128
m8i.12xlarge	16	64	192

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8i.16xlarge	16	64	256
m8i.24xlarge	16	128	384
m8i.32xlarge	16	128	512
m8i.48xlarge	32	128	768
m8i.96xlarge	32	128	1536
m8i.metal-48xl	32	128	768
m8i.metal-96xl	32	128	1536
m8id.large	2	2	6
m8id.xlarge	4	4	16
m8id.2xlarge	8	8	32
m8id.4xlarge	8	16	64
m8id.8xlarge	8	32	128
m8id.12xlarge	16	64	192
m8id.16xlarge	16	64	256
m8id.24xlarge	16	128	384
m8id.32xlarge	16	128	512
m8id.48xlarge	32	128	768
m8id.96xlarge	32	128	1536
m8id.metal-48xl	32	128	768

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8id.metal-96xl	32	128	1536
m8i-flex.large	1	1	3
m8i-flex.xlarge	2	2	8
m8i-flex.2xlarge	4	4	16
m8i-flex.4xlarge	4	8	32
m8i-flex.8xlarge	4	16	64
m8i-flex.12xlarge	8	32	96
m8i-flex.16xlarge	8	32	128
m8in.large	2	2	8
m8in.xlarge	4	4	16
m8in.2xlarge	8	8	32
m8in.4xlarge	8	16	64
m8in.8xlarge	16	32	128
m8in.12xlarge	16	64	192
m8in.16xlarge	16	64	256
m8in.24xlarge	16	128	256
m8in.32xlarge	32	128	512
m8in.48xlarge	32	128	768
m8in.96xlarge	32	128	1536 *

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8in.metal-48xl	32	128	768
m8in.metal-96xl	32	128	1536 *
m8idn.large	2	2	8
m8idn.xlarge	4	4	16
m8idn.2xlarge	8	8	32
m8idn.4xlarge	8	16	64
m8idn.8xlarge	16	32	128
m8idn.12xlarge	16	64	192
m8idn.16xlarge	16	64	256
m8idn.24xlarge	16	128	256
m8idn.32xlarge	32	128	512
m8idn.48xlarge	32	128	768
m8idn.96xlarge	32	128	1536 *
m8idn.metal-48xl	32	128	768
m8idn.metal-96xl	32	128	1536 *
m8ine.large	2	2	8
m8ine.xlarge	4	4	16
m8ine.2xlarge	8	8	32
m8ine.4xlarge	16	16	128

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8ine.8xlarge	32	32	256
m8ine.12xlarge	32	64	384
m8ib.large	2	2	8
m8ib.xlarge	4	4	16
m8ib.2xlarge	8	8	32
m8ib.4xlarge	8	16	64
m8ib.8xlarge	16	32	128
m8ib.12xlarge	16	64	192
m8ib.16xlarge	16	64	256
m8ib.24xlarge	16	128	256
m8ib.32xlarge	32	128	512
m8ib.48xlarge	32	128	768
m8ib.96xlarge	32	128	1536 *
m8ib.metal-48xl	32	128	768
m8ib.metal-96xl	32	128	1536 *
m8idb.large	2	2	8
m8idb.xlarge	4	4	16
m8idb.2xlarge	8	8	32
m8idb.4xlarge	8	16	64

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m8idb.8xlarge	16	32	128
m8idb.12xlarge	16	64	192
m8idb.16xlarge	16	64	256
m8idb.24xlarge	16	128	256
m8idb.32xlarge	32	128	512
m8idb.48xlarge	32	128	768
m8idb.96xlarge	32	128	1536 *
m8idb.metal-48xl	32	128	768
m8idb.metal-96xl	32	128	1536 *
m9g.medium	1	1	2
m9g.large	2	2	6
m9g.xlarge	4	4	16
m9g.2xlarge	8	8	32
m9g.4xlarge	8	16	64
m9g.8xlarge	8	32	128
m9g.12xlarge	16	64	192
m9g.16xlarge	16	64	256
m9g.24xlarge	16	128	384
m9g.48xlarge	32	128	768

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
m9g.metal-48xl	32	128	768
m9gd.medium	1	1	2
m9gd.large	2	2	6
m9gd.xlarge	4	4	16
m9gd.2xlarge	8	8	32
m9gd.4xlarge	8	16	64
m9gd.8xlarge	8	32	128
m9gd.12xlarge	16	64	192
m9gd.16xlarge	16	64	256
m9gd.24xlarge	16	128	384
m9gd.48xlarge	32	128	768
m9gd.metal-48xl	32	128	768

Compute optimized

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c6i.large	2	2	6
c6i.xlarge	4	4	16
c6i.2xlarge	8	8	32

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c6i.4xlarge	8	16	64
c6i.8xlarge	8	32	64
c6i.12xlarge	8	32	64
c6i.16xlarge	8	32	120
c6i.24xlarge	8	32	120
c6i.32xlarge	8	32	120
c6id.large	2	2	6
c6id.xlarge	4	4	16
c6id.2xlarge	8	8	32
c6id.4xlarge	8	16	64
c6id.8xlarge	8	32	64
c6id.12xlarge	8	32	64
c6id.16xlarge	8	32	120
c6id.24xlarge	8	32	120
c6id.32xlarge	8	32	120
c6in.large	2	2	6
c6in.xlarge	4	4	16
c6in.2xlarge	8	8	32
c6in.4xlarge	8	16	64

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c6in.8xlarge	16	32	128
c6in.12xlarge	16	32	128
c6in.16xlarge	16	32	240
c6in.24xlarge	32	32	480
c6in.32xlarge	32	32	512 *
c8a.medium	1	1	3
c8a.large	2	2	6
c8a.xlarge	4	4	16
c8a.2xlarge	8	8	32
c8a.4xlarge	8	16	64
c8a.8xlarge	8	32	128
c8a.12xlarge	16	64	192
c8a.16xlarge	16	64	256
c8a.24xlarge	16	128	384
c8a.48xlarge	32	128	768
c8a.metal-24xl	16	128	384
c8a.metal-48xl	32	128	768
c8gb.medium	1	1	2
c8gb.large	2	2	6

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c8gb.xlarge	4	4	16
c8gb.2xlarge	8	8	32
c8gb.4xlarge	8	16	64
c8gb.8xlarge	8	32	128
c8gb.12xlarge	16	64	192
c8gb.16xlarge	16	64	256
c8gb.24xlarge	16	128	384
c8gb.48xlarge	32	128	768 *
c8gb.metal-24xl	32	128	768
c8gb.metal-48xl	32	128	768 *
c8gn.medium	1	1	2
c8gn.large	2	2	6
c8gn.xlarge	4	4	16
c8gn.2xlarge	8	8	32
c8gn.4xlarge	8	16	64
c8gn.8xlarge	8	32	128
c8gn.12xlarge	16	64	192
c8gn.16xlarge	16	64	256
c8gn.24xlarge	16	128	384

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c8gn.48xlarge	32	128	768 *
c8gn.metal-24xl	32	128	768
c8gn.metal-48xl	32	128	768 *
c8i.large	2	2	6
c8i.xlarge	4	4	16
c8i.2xlarge	8	8	32
c8i.4xlarge	8	16	64
c8i.8xlarge	8	32	128
c8i.12xlarge	16	64	192
c8i.16xlarge	16	64	256
c8i.24xlarge	16	128	384
c8i.32xlarge	16	128	512
c8i.48xlarge	32	128	768
c8i.96xlarge	32	128	1536
c8i.metal-48xl	32	128	768
c8i.metal-96xl	32	128	1536
c8id.large	2	2	6
c8id.xlarge	4	4	16
c8id.2xlarge	8	8	32

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c8id.4xlarge	8	16	64
c8id.8xlarge	8	32	128
c8id.12xlarge	16	64	192
c8id.16xlarge	16	64	256
c8id.24xlarge	16	128	384
c8id.32xlarge	16	128	512
c8id.48xlarge	32	128	768
c8id.96xlarge	32	128	1536
c8id.metal-48xl	32	128	768
c8id.metal-96xl	32	128	1536
c8i-flex.large	1	1	3
c8i-flex.xlarge	2	2	8
c8i-flex.2xlarge	4	4	16
c8i-flex.4xlarge	4	8	32
c8i-flex.8xlarge	4	16	64
c8i-flex.12xlarge	8	32	96
c8i-flex.16xlarge	8	32	128
c8in.large	2	2	8
c8in.xlarge	4	4	16

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c8in.2xlarge	8	8	32
c8in.4xlarge	8	16	64
c8in.8xlarge	16	32	128
c8in.12xlarge	16	64	192
c8in.16xlarge	16	64	256
c8in.24xlarge	16	128	256
c8in.32xlarge	32	128	512
c8in.48xlarge	32	128	768
c8in.96xlarge	32	128	1536 *
c8in.metal-48xl	32	128	768
c8in.metal-96xl	32	128	1536 *
c8ine.large	2	2	8
c8ine.xlarge	4	4	16
c8ine.2xlarge	8	8	32
c8ine.4xlarge	16	16	128
c8ine.8xlarge	32	32	256
c8ine.12xlarge	32	64	384
c8ib.large	2	2	8
c8ib.xlarge	4	4	16

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c8ib.2xlarge	8	8	32
c8ib.4xlarge	8	16	64
c8ib.8xlarge	16	32	128
c8ib.12xlarge	16	64	192
c8ib.16xlarge	16	64	256
c8ib.24xlarge	16	128	256
c8ib.32xlarge	32	128	512
c8ib.48xlarge	32	128	768
c8ib.96xlarge	32	128	1536 *
c8ib.metal-48xl	32	128	768
c8ib.metal-96xl	32	128	1536 *
c9g.medium	1	1	2
c9g.large	2	2	6
c9g.xlarge	4	4	16
c9g.2xlarge	8	8	32
c9g.4xlarge	8	16	64
c9g.8xlarge	8	32	128
c9g.12xlarge	16	64	192
c9g.16xlarge	16	64	256

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
c9g.24xlarge	16	128	384
c9g.48xlarge	32	128	768
c9g.metal-48xl	32	128	768
c9gd.medium	1	1	2
c9gd.large	2	2	6
c9gd.xlarge	4	4	16
c9gd.2xlarge	8	8	32
c9gd.4xlarge	8	16	64
c9gd.8xlarge	8	32	128
c9gd.12xlarge	16	64	192
c9gd.16xlarge	16	64	256
c9gd.24xlarge	16	128	384
c9gd.48xlarge	32	128	768
c9gd.metal-48xl	32	128	768

Memory optimized

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r6i.large	2	2	6

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r6i.xlarge	4	4	16
r6i.2xlarge	8	8	32
r6i.4xlarge	8	16	64
r6i.8xlarge	8	32	64
r6i.12xlarge	8	32	64
r6i.16xlarge	8	32	120
r6i.24xlarge	8	32	120
r6i.32xlarge	8	32	120
r6id.large	2	2	6
r6id.xlarge	4	4	16
r6id.2xlarge	8	8	32
r6id.4xlarge	8	16	64
r6id.8xlarge	8	32	64
r6id.12xlarge	8	32	64
r6id.16xlarge	8	32	120
r6id.24xlarge	8	32	120
r6id.32xlarge	8	32	120
r6idn.large	2	2	6
r6idn.xlarge	4	4	16

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r6idn.2xlarge	8	8	32
r6idn.4xlarge	8	16	64
r6idn.8xlarge	16	32	128
r6idn.12xlarge	16	32	128
r6idn.16xlarge	16	32	240
r6idn.24xlarge	32	32	480
r6idn.32xlarge	32	32	512 *
r6in.large	2	2	6
r6in.xlarge	4	4	16
r6in.2xlarge	8	8	32
r6in.4xlarge	8	16	64
r6in.8xlarge	16	32	128
r6in.12xlarge	16	32	128
r6in.16xlarge	16	32	240
r6in.24xlarge	32	32	480
r6in.32xlarge	32	32	512 *
r8a.medium	1	1	3
r8a.large	2	2	6
r8a.xlarge	4	4	16

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8a.2xlarge	8	8	32
r8a.4xlarge	8	16	64
r8a.8xlarge	8	32	128
r8a.12xlarge	16	64	192
r8a.16xlarge	16	64	256
r8a.24xlarge	16	128	384
r8a.48xlarge	32	128	768
r8a.metal-24xl	16	128	384
r8a.metal-48xl	32	128	768
r8gb.medium	1	1	2
r8gb.large	2	2	6
r8gb.xlarge	4	4	16
r8gb.2xlarge	8	8	32
r8gb.4xlarge	8	16	64
r8gb.8xlarge	8	32	128
r8gb.12xlarge	16	64	192
r8gb.16xlarge	16	64	256
r8gb.24xlarge	16	128	384
r8gb.48xlarge	32	128	768 *

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8gb.metal-24xl	32	128	768
r8gb.metal-48xl	32	128	768 *
r8gn.medium	1	1	2
r8gn.large	2	2	6
r8gn.xlarge	4	4	16
r8gn.2xlarge	8	8	32
r8gn.4xlarge	8	16	64
r8gn.8xlarge	8	32	128
r8gn.12xlarge	16	64	192
r8gn.16xlarge	16	64	256
r8gn.24xlarge	16	128	384
r8gn.48xlarge	32	128	768 *
r8gn.metal-24xl	32	128	768
r8gn.metal-48xl	32	128	768 *
r8i.large	2	2	6
r8i.xlarge	4	4	16
r8i.2xlarge	8	8	32
r8i.4xlarge	8	16	64
r8i.8xlarge	8	32	128

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8i.12xlarge	16	64	192
r8i.16xlarge	16	64	256
r8i.24xlarge	16	128	384
r8i.32xlarge	16	128	512
r8i.48xlarge	32	128	768
r8i.96xlarge	32	128	1536
r8i.metal-48xl	32	128	768
r8i.metal-96xl	32	128	1536
r8id.large	2	2	6
r8id.xlarge	4	4	16
r8id.2xlarge	8	8	32
r8id.4xlarge	8	16	64
r8id.8xlarge	8	32	128
r8id.12xlarge	16	64	192
r8id.16xlarge	16	64	256
r8id.24xlarge	16	128	384
r8id.32xlarge	16	128	512
r8id.48xlarge	32	128	768
r8id.96xlarge	32	128	1536

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8id.metal-48xl	32	128	768
r8id.metal-96xl	32	128	1536
r8i-flex.large	1	1	3
r8i-flex.xlarge	2	2	8
r8i-flex.2xlarge	4	4	16
r8i-flex.4xlarge	4	8	32
r8i-flex.8xlarge	4	16	64
r8i-flex.12xlarge	8	32	96
r8i-flex.16xlarge	8	32	128
r8in.large	2	2	8
r8in.xlarge	4	4	16
r8in.2xlarge	8	8	32
r8in.4xlarge	8	16	64
r8in.8xlarge	16	32	128
r8in.12xlarge	16	64	192
r8in.16xlarge	16	64	256
r8in.24xlarge	16	128	256
r8in.32xlarge	32	128	512
r8in.48xlarge	32	128	768

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8in.96xlarge	32	128	1536 *
r8in.metal-48xl	32	128	768
r8in.metal-96xl	32	128	1536 *
r8idn.large	2	2	8
r8idn.xlarge	4	4	16
r8idn.2xlarge	8	8	32
r8idn.4xlarge	8	16	64
r8idn.8xlarge	16	32	128
r8idn.12xlarge	16	64	192
r8idn.16xlarge	16	64	256
r8idn.24xlarge	16	128	256
r8idn.32xlarge	32	128	512
r8idn.48xlarge	32	128	768
r8idn.96xlarge	32	128	1536 *
r8idn.metal-48xl	32	128	768
r8idn.metal-96xl	32	128	1536 *
r8ib.large	2	2	8
r8ib.xlarge	4	4	16
r8ib.2xlarge	8	8	32

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8ib.4xlarge	8	16	64
r8ib.8xlarge	16	32	128
r8ib.12xlarge	16	64	192
r8ib.16xlarge	16	64	256
r8ib.24xlarge	16	128	256
r8ib.32xlarge	32	128	512
r8ib.48xlarge	32	128	768
r8ib.96xlarge	32	128	1536 *
r8ib.metal-48xl	32	128	768
r8ib.metal-96xl	32	128	1536 *
r8idb.large	2	2	8
r8idb.xlarge	4	4	16
r8idb.2xlarge	8	8	32
r8idb.4xlarge	8	16	64
r8idb.8xlarge	16	32	128
r8idb.12xlarge	16	64	192
r8idb.16xlarge	16	64	256
r8idb.24xlarge	16	128	256
r8idb.32xlarge	32	128	512

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r8idb.48xlarge	32	128	768
r8idb.96xlarge	32	128	1536 *
r8idb.metal-48xl	32	128	768
r8idb.metal-96xl	32	128	1536 *
r9g.medium	1	1	2
r9g.large	2	2	6
r9g.xlarge	4	4	16
r9g.2xlarge	8	8	32
r9g.4xlarge	8	16	64
r9g.8xlarge	8	32	128
r9g.12xlarge	16	64	192
r9g.16xlarge	16	64	256
r9g.24xlarge	16	128	384
r9g.48xlarge	32	128	768
r9g.metal-48xl	32	128	768
r9gd.medium	1	1	2
r9gd.large	2	2	6
r9gd.xlarge	4	4	16
r9gd.2xlarge	8	8	32

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
r9gd.4xlarge	8	16	64
r9gd.8xlarge	8	32	128
r9gd.12xlarge	16	64	192
r9gd.16xlarge	16	64	256
r9gd.24xlarge	16	128	384
r9gd.48xlarge	32	128	768
r9gd.metal-48xl	32	128	768
x8aedz.large	2	2	8
x8aedz.xlarge	4	4	16
x8aedz.3xlarge	4	16	48
x8aedz.6xlarge	8	32	96
x8aedz.12xlarge	8	64	192
x8aedz.24xlarge	16	128	384
x8aedz.metal-12xl	8	64	192
x8aedz.metal-24xl	16	128	384
x8i.large	2	2	6
x8i.xlarge	4	4	16
x8i.2xlarge	8	8	32
x8i.4xlarge	8	16	64

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
x8i.8xlarge	8	32	128
x8i.12xlarge	16	64	192
x8i.16xlarge	16	64	256
x8i.24xlarge	16	128	384
x8i.32xlarge	16	128	512
x8i.48xlarge	32	128	768
x8i.64xlarge	32	128	1024
x8i.96xlarge	32	128	1536
x8i.metal-48xl	32	128	768
x8i.metal-96xl	32	128	1536

Storage optimized

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
i8ge.large	2	2	6
i8ge.xlarge	4	4	16
i8ge.2xlarge	8	8	32
i8ge.3xlarge	8	16	48
i8ge.6xlarge	8	32	96

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
i8ge.12xlarge	16	64	192
i8ge.18xlarge	16	128	288
i8ge.24xlarge	16	128	384
i8ge.48xlarge	32	128	1536 *

Accelerated computing

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
g7.2xlarge	8	8	32
g7.4xlarge	8	16	64
g7.8xlarge	8	32	80
g7.12xlarge	16	64	192
g7.24xlarge	16	128	384
g7.48xlarge	32	128	768 *

High performance computing

執行個體類型	每個介面的預設 ENA 佇列數量	每個介面的 ENA 佇列數量上限	每個執行個體的 ENA 佇列數量上限
hpc8a.96xlarge	32	128	512 *

> **Note**
>
> * 這些執行個體類型具有多張網路卡。其他執行個體類型具備單張網路卡。如需詳細資訊，請參閱網路卡。

設定 ENA 佇列配置

在支援的執行個體類型上，您可以設定每個 ENI 的 ENA 佇列計數。使用 AWS 管理主控台 AWS CLI、 或 PowerShell。在 中 AWS 管理主控台，此設定會出現在每個網路界面下方。

> **Note**
>
> 設定 ENA 佇列之前，請注意下列需求：
>
> - 您必須先停止執行個體，才能設定 ENA 佇列的數量。
> - ENA 佇列的值必須為 2 的功率，例如 1、2、4、8、16 或 32。
> - 任何單一 ENI 上的佇列數目不能超過執行個體上的 vCPUs 數目。

變更佇列計數之前，請使用下列命令檢查目前的計數。

AWS CLI

```
aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

PowerShell

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces |
  Select-Object NetworkInterfaceId,
    @{N='DeviceIndex';E={$_.Attachment.DeviceIndex}},
    @{N='AttachmentId';E={$_.Attachment.AttachmentId}},
    @{N='EnaQueueCount';E={$_.Attachment.EnaQueueCount}}
```

使用 ENA 佇列連接網路介面

在下列範例中，在 ENI 上設定了 16 個 ENA 佇列。

AWS CLI

使用 ENA 佇列連接網路介面

使用 attach-network-interface 命令。

```
aws ec2 attach-network-interface \
  --network-interface-id eni-abcdef01234567890 \
  --instance-id i-1234567890abcdef0 \
  --device-index 1 \
  --ena-queue-count 16
```

PowerShell

使用 ENA 佇列連接網路介面

使用 Add-EC2NetworkInterface cmdlet。

```
Add-EC2NetworkInterface `
  -NetworkInterfaceId eni-abcdef01234567890 `
  -InstanceId i-1234567890abcdef0 `
  -DeviceIndex 1 `
  -EnaQueueCount 16
```

使用 ENA 佇列啟動執行個體

在下列範例中，會在 3 個 ENIs 上設定 16 個 ENA 佇列。

AWS CLI

使用 ENA 佇列啟動執行個體

使用 run-instances 命令。

```
aws ec2 run-instances \
  --image-id ami-1234567890abcdef0 \
  --instance-type c8i.4xlarge \
  --network-interfaces \
    "[{\"DeviceIndex\":0,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16},
      {\"DeviceIndex\":1,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16},
```

```
      {\"DeviceIndex\":2,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16}]"
```

PowerShell

使用 ENA 佇列啟動執行個體

使用 New-EC2Instance cmdlet。

```
New-EC2Instance `
  -ImageId ami-1234567890abcdef0 `
  -InstanceType c8i.4xlarge `
  -NetworkInterface @(
    @{DeviceIndex=0; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16},
    @{DeviceIndex=1; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16},
    @{DeviceIndex=2; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16}
  )
```

修改現有網路界面上的 ENA 佇列

在下列範例中，在 ENI 上設定了 16 個 ENA 佇列。

AWS CLI

在網路界面上修改 ENA 佇列

使用 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
  --network-interface-id eni-1234567890abcdef0 \
  --attachment AttachmentId=eni-attach-1234567890abcdef0,EnaQueueCount=16
```

PowerShell

在網路界面上修改 ENA 佇列

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
  -NetworkInterfaceId eni-1234567890abcdef0 `
  -Attachment_AttachmentId eni-attach-1234567890abcdef0 `
  -Attachment_EnaQueueCount 16
```

下列範例中，ENA 計數已重設為預設值。

AWS CLI

```
aws ec2 modify-network-interface-attribute \
  --network-interface-id eni-1234567890abcdef0 \
  --attachment AttachmentId=eni-attach-1234567890abcdef0,DefaultEnaQueueCount=true
```

PowerShell

```
Edit-EC2NetworkInterfaceAttribute `
  -NetworkInterfaceId eni-1234567890abcdef0 `
  -Attachment_AttachmentId eni-attach-1234567890abcdef0 `
  -Attachment_DefaultEnaQueueCount $true
```

對 Linux 上的 ENA 核心驅動程式進行疑難排解

彈性網路轉接器 (ENA) 設計旨在提高作業系統運作狀態，並減少由於預料之外硬體行為及/或故障而導致長期中斷之可能性。ENA 架構盡可能使裝置或驅動程式的故障對系統保持透明。本主題提供 ENA 的故障診斷資訊。

如果您無法連線至您的執行個體，請由 疑難排解連線問題 章節開始。

如果您在遷移至第六代執行個體類型後遇到效能降低的情況，請參閱文章 What do I need to do before migrating my EC2 instance to a sixth generation instance to make sure that I get maximum network performance? (在將 EC2 執行個體遷移至第六代執行個體之前，我需要做些什麼來確保獲得最高網路效能？)

如果您能夠連線至您的執行個體，則可以使用本主題後面章節中說明的故障診斷和復原機制來收集診斷資訊。

內容

- 疑難排解連線問題
- 保持有效機制
- 註冊讀取逾時
- 統計資料
- syslog 中的驅動程式錯誤日誌
- 非最佳化組態通知

疑難排解連線問題

如果您在啟用增強聯網時失去連線，則 ena 模組可能與您的執行個體目前執行之核心不相容。如果您為特定核心版本 (不含 dkms 或具有設定不正確的 dkms.conf 檔案) 安裝模組並更新執行個體核心，則可能會發生此情況。如果在開機階段載入的執行個體核心沒有正確安裝 ena 模組，則您的執行個體將無法識別網路轉接器，且將無法連接到您的執行個體。

如果您為 PV 執行個體或 AMI 啟用增強聯網，也可能導致無法連接到您的執行個體。

如果您的執行個體在啟用增強聯網與 ENA 後即無法連接，您可以為您的執行個體停用 `enaSupport` 屬性，以將其回復到原始網路轉接器。

停用增強聯網與 ENA (EBS 後端執行個體)

1. 從本機電腦中，使用 Amazon EC2 主控台、stop-instances 命令 (AWS CLI) 或 Stop-EC2Instance cmdlet (AWS Tools for PowerShell) 來停止執行個體。
2. 從本機電腦中，搭配使用 modify-instance-attribute 命令與 `--no-ena-support` 選項或具有 `-EnaSupport $false` 參數的 Edit-EC2InstanceAttribute cmdlet 來停用增強型聯網屬性。
3. 從本機電腦中，使用 Amazon EC2 主控台、start-instances 命令或 Start-EC2Instance cmdlet 來啟動執行個體。
4. (選用) 請連線到您的執行個體，並嘗試遵循 ena 中的步驟使用目前核心版本來重新安裝 在 EC2 執行個體上使用 ENA 啟用增強型網路 模組。

停用增強聯網與 ENA (執行個體後端執行個體)

1. 如 建立 Amazon S3 支援的 AMI 中所述，建立新的 AMI。
2. 當您註冊 AMI 時，請務必在 stop-instances 命令 (AWS CLI) 中包含 `--no-ena-support` 選項，或在 Register-EC2Image cmdlet 中包含 `-EnaSupport $false` 參數。

保持有效機制

ENA 裝置以固定速率張貼保持有效事件 (通常每秒一次)。ENA 驅動程式實作監視程式機制，該機制會檢查這些保持有效訊息的存在。如果有訊息存在，監視程式會重新啟動，否則驅動程式會認為裝置出現故障並執行下列作業：

- 傾印目前統計資料至 syslog
- 重設 ENA 裝置

- 重設 ENA 驅動程式狀態

上述重設程序可能會在短時間內導致一些流量遺失 (TCP 連線應該能夠復原)，但不應影響使用者。

ENA 裝置也可以透過不傳送保持有效通知來間接請求裝置重設程序，例如，如果 ENA 裝置在載入無法恢復的組態之後成為未知狀態。

以下是重設程序的範例：

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog
 process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current
 statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.......
........
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the
 end of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The
 driver begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1
 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date
 [Wed Apr  6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
```

```
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset
 process is complete
```

註冊讀取逾時

ENA 架構暗示記憶體映射之 I/O (MMIO) 讀取操作的限量使用。ENA 裝置驅動程式僅在其初始化程序中存取 MMIO 註冊。

如果驅動程式日誌 (在 dmesg 輸出中可用) 指出讀取操作失敗，則可能是由不相容或編譯錯誤的驅動程式、忙碌的硬體裝置或硬體故障引起的。

表示讀取操作失敗的間歇日誌項目不應視為問題；驅動程式會在此情況下將其重試。但是，包含讀取失敗的一系列日誌項目則會指出驅動程式或硬體問題。

以下是驅動程序日誌項目的範例，指出由於逾時而導致的讀取操作失敗：

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[1] offset[88] actual: req id[57006] offset[0]
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[2] offset[8] actual: req id[57007] offset[0]
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

統計資料

如果您遇到網路效能不足或延遲問題，則應擷取並檢查裝置統計資料。您可以使用 ethtool 來獲得這些統計資料，如下所示。

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
 tx_timeout: 0
 suspend: 0
 resume: 0
 wd_expired: 0
 interface_up: 1
 interface_down: 0
 admin_q_pause: 0
 bw_in_allowance_exceeded: 0
 bw_out_allowance_exceeded: 0
 pps_allowance_exceeded: 0
 conntrack_allowance_available: 450878
```

```
  conntrack_allowance_exceeded: 0
  linklocal_allowance_exceeded: 0
  queue_0_tx_cnt: 4329
  queue_0_tx_bytes: 1075749
  queue_0_tx_queue_stop: 0
  ...
```

以下說明下列命令輸出參數：

`tx_timeout`：*N*

Netdev 監視程式啟用的次數。

`suspend`：*N*

驅動程式執行暫停操作的次數。

`resume`：*N*

驅動程式執行繼續操作的次數。

`wd_expired`：*N*

驅動程式在前 3 秒內未收到保持有效事件的次數。

`interface_up`：*N*

ENA 介面出現的次數。

`interface_down`：*N*

ENA 介面關閉的次數。

`admin_q_pause`：*N*

在執行中狀態下找不到管理員佇列的次數。

`bw_in_allowance_exceeded`：*N*

因傳入的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。

`bw_out_allowance_exceeded`：*N*

因傳出的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。

`pps_allowance_exceeded`：*N*

因雙向 PPS 超過執行個體的上限而排入佇列或丟棄的封包數目。*

`conntrack_allowance_available : N`

在達到該執行個體類型的追蹤連線限額之前，執行個體可建立的追蹤連線數目。僅適用於 Nitro 型執行個體。不支援 FreeBSD 執行個體或 DPDK 環境。

`conntrack_allowance_exceeded : N`

因為連線追蹤超過執行個體的上限且無法建立新的連線，而丟棄的封包數目。這可能會導致傳送或傳回執行個體流量的封包遺失。

`linklocal_allowance_exceeded : N`

由於本機代理伺服器服務的流量 PPS 超過網路介面上限而丟棄的封包數目。這會影響 Amazon DNS 服務、執行個體中繼資料服務和 Amazon Time Sync Service 的流量，但不會影響自訂 DNS 解析器的流量。

`queue_N_tx_cnt : N`

此佇列的已傳輸封包數目。

`queue_N_tx_bytes : N`

此佇列的已傳輸位元組數目。

`queue_N_tx_queue_stop : N`

佇列 *N* 已滿並停止的次數。

`queue_N_tx_queue_wakeup : N`

佇列 *N* 在停止後恢復的次數。

`queue_N_tx_dma_mapping_err : N`

直接記憶體存取錯誤計數。如果此值為 0，則指出系統資源不足。

`queue_N_tx_linearize : N`

此佇列的 SKB 線性化嘗試次數。

`queue_N_tx_linearize_failed : N`

此佇列的 SKB 線性化失敗次數。

`queue_N_tx_napi_comp : N`

`napi` 處理常式為此佇列呼叫 `napi_complete` 的次數。

queue_*N*_tx_tx_poll：*N*

napi 處理常式為此佇列排程的次數。

queue_*N*_tx_doorbells：*N*

此佇列的傳輸門鈴數目。

queue_*N*_tx_prepare_ctx_err：*N*

此佇列的 ena_com_prepare_tx 失敗次數。

queue_*N*_tx_bad_req_id：*N*

此佇列的無效 req_id。有效的 req_id 為零，減去 queue_size，減去 1。

queue_*N*_tx_llq_buffer_copy：*N*

其標頭大小大於此佇列 llq 項目的封包數目。

queue_*N*_tx_missed_tx：*N*

此佇列未完成的封包數目。

queue_*N*_tx_unmask_interrupt：*N*

此佇列 tx 中斷取消遮罩的次數。

queue_*N*_rx_cnt：*N*

此佇列接收的封包數目。

queue_*N*_rx_bytes：*N*

此佇列接收的位元組數目。

queue_*N*_rx_rx_copybreak_pkt：*N*

此佇列的 rx 佇列接收到小於 rx_copybreak 封包大小之封包的次數。

queue_*N*_rx_csum_good：*N*

此佇列的 rx 佇列接收到檢查總和已檢查且正確之封包的次數。

queue_*N*_rx_refil_partial：*N*

驅動程式未成功以此佇列緩衝區重新填滿 rx 佇列之空白部分的次數。如果值不為 0，則指出記憶體資源不足。

queue_*N*_rx_bad_csum：*N*

rx 佇列對此佇列有不良檢查總和的次數 (僅當支援 rx 檢查總和卸載時)。

queue_*N*_rx_page_alloc_fail：*N*

此佇列頁面配置失敗的次數。如果值不為 0，則指出記憶體資源不足。

queue_*N*_rx_skb_alloc_fail：*N*

此佇列 SKB 配置失敗的次數。如果值不為 0，則指出系統資源不足。

queue_*N*_rx_dma_mapping_err：*N*

直接記憶體存取錯誤計數。如果此值為 0，則指出系統資源不足。

queue_*N*_rx_bad_desc_num：*N*

每個封包的緩衝區過多。如果值不為 0，則表示使用非常小的緩衝區。

queue_*N*_rx_bad_req_id：*N*

此佇列的 req_id 無效。有效的 req_id 是從 [0, queue_size - 1] 開始。

queue_*N*_rx_empty_rx_ring：*N*

此佇列的 rx 佇列為空白的次數。

queue_*N*_rx_csum_unchecked：*N*

此佇列的 rx 佇列接收到檢查總和尚未檢查之封包的次數。

queue_*N*_rx_xdp_aborted：*N*

XDP 封包歸類為 XDP_ABORT 的次數。

queue_*N*_rx_xdp_drop：*N*

XDP 封包歸類為 XDP_DROP 的次數。

queue_*N*_rx_xdp_pass：*N*

XDP 封包歸類為 XDP_PASS 的次數。

queue_*N*_rx_xdp_tx：*N*

XDP 封包歸類為 XDP_TX 的次數。

queue_*N*_rx_xdp_invalid：*N*

封包的 XDP 傳回碼無效的次數。

queue_*N*_rx_xdp_redirect：*N*

XDP 封包歸類為 XDP_REDIRECT 的次數。

queue_*N*_xdp_tx_cnt：*N*

此佇列的已傳輸封包數目。

queue_*N*_xdp_tx_bytes：*N*

此佇列的已傳輸位元組數目。

queue_*N*_xdp_tx_queue_stop：*N*

此佇列已滿並停止的次數。

queue_*N*_xdp_tx_queue_wakeup：*N*

此佇列在停止後恢復的次數。

queue_*N*_xdp_tx_dma_mapping_err：*N*

直接記憶體存取錯誤計數。如果此值為 0，則指出系統資源不足。

queue_*N*_xdp_tx_linearize：*N*

此佇列的 XDP 緩衝區線性化嘗試次數。

queue_*N*_xdp_tx_linearize_failed：*N*

此佇列的 XDP 緩衝區線性化失敗次數。

queue_*N*_xdp_tx_napi_comp：*N*

napi 處理常式為此佇列呼叫 napi_complete 的次數。

queue_*N*_xdp_tx_tx_poll：*N*

napi 處理常式為此佇列排程的次數。

queue_*N*_xdp_tx_doorbells：*N*

此佇列的傳輸門鈴數目。

queue_*N*_xdp_tx_prepare_ctx_err：*N*

此佇列的 ena_com_prepare_tx 失敗次數。此值應始終為 0；如不是，請查看驅動程式日誌。

queue_*N*_xdp_tx_bad_req_id : *N*

此佇列的 req_id 無效。有效的 req_id 是從 [0, queue_size - 1] 開始。

queue_*N*_xdp_tx_llq_buffer_copy : *N*

此佇列使用 llq 緩衝區複本複製其標頭的封包數目。

queue_*N*_xdp_tx_missed_tx : *N*

tx 佇列項目錯過此佇列完成逾時的次數。

queue_*N*_xdp_tx_unmask_interrupt : *N*

此佇列 tx 中斷取消遮罩的次數。

ena_admin_q_aborted_cmd : *N*

管理命令中止次數。此情況通常發生在自動復原程序中。

ena_admin_q_submitted_cmd : *N*

管理佇列門鈴數目。

ena_admin_q_completed_cmd : *N*

管理佇列完成數目。

ena_admin_q_out_of_space : *N*

驅動程式嘗試提交新管理命令但佇列已滿的次數。

ena_admin_q_no_completion : *N*

驅動程式未取得命令的管理完成之次數。

syslog 中的驅動程式錯誤日誌

在系統開機期間，ENA 驅動程式會將日誌訊息寫入至 syslog。如果您遇到問題，可以檢查這些日誌以尋找錯誤。以下是系統開機時，ENA 驅動程式在 syslog 中記錄的資訊，以及選取訊息的一些註釋範例。

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.416939] [ENA_COM:
 ena_com_validate_version] ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.420915] [ENA_COM:
 ena_com_validate_version] ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.256831] ena 0000:00:03.0: Device
 watchdog is Enabled
```

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.672947] ena 0000:00:03.0: creating 8 io
 queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.680885] [ENA_COM:
 ena_com_init_interrupt_moderation] Feature 20 isn't supported  // Interrupt moderation
 is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.691609] [ENA_COM:
 ena_com_get_feature_ex] Feature 10 isn't supported // RSS HASH function configuration
 is not supported by the
 device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.694583] [ENA_COM:
 ena_com_get_feature_ex] Feature 18 isn't supported //RSS HASH input source
 configuration is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.697433] [ENA_COM:
 ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.701064] ena 0000:00:03.0 (unnamed
 net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.704917] ena 0000:00:03.0: Elastic
 Network Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  480.805037] EXT4-fs (xvda1): re-mounted.
 Opts: (null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  481.025842] NET: Registered protocol family
 10
```

我可以忽略哪些錯誤？

彈性網路轉接器可以忽略下列可能出現在系統錯誤日誌中的警告：

Set host attribute isn't supported

此裝置不支援主機屬性。

failed to alloc buffer for rx queue

此為可復原的錯誤，指出在錯誤拋出時可能存在記憶體壓力問題。

Feature *X* isn't supported

參考的功能不受彈性網路轉接器支援。*X* 可能的值包括：

- 10：此裝置不支援 RSS 雜湊功能組態。
- 12：此裝置不支援 RSS 間接資料表組態。
- 18：此裝置不支援 RSS 雜湊輸入組態。
- 20：此裝置不支援中斷管制。

- 27：彈性網路轉接器驅動程式不支援從 snmpd 中輪詢乙太網路功能。

Failed to config AENQ

彈性網路轉接器不支援 AENQ 組態。

Trying to set unsupported AENQ events

此錯誤指出嘗試設定彈性網路轉接器不支援的 AENQ 事件群組。

非最佳化組態通知

ENA 裝置會偵測驅動程式中您可以變更的非最佳化組態設定。裝置會通知 ENA 驅動程式，並將警告記錄到主控台。下列範例顯示警告訊息的格式。

```
Sub-optimal configuration notification code: 1. Refer to AWS ENA documentation for
 additional details and mitigation options.
```

下列清單顯示通知程式碼詳細資訊，以及非最佳化組態調查結果的建議動作。

- 程式碼 1：不建議使用帶寬 LLQ 組態的 ENA Express

  ENA Express ENI 設定有寬 LLQ。此組態不是最佳化的，可能會影響 ENA Express 的效能。我們建議您在使用 ENA Express ENI 時停用寬 LLQ 設定，如下所示。

  ```
  sudo rmmod ena && sudo modprobe ena force_large_llq_header=0
  ```

  如需 ENA Express 最佳化組態的詳細資訊，請參閱 使用 ENA Express 提高 EC2 執行個體之間的網路效能。
- 代碼 2：不建議使用具有次佳 Tx 佇列深度的 ENA Express ENI

  設定 ENA Express ENI 為具有次佳 Tx 佇列深度。此組態可能會影響 ENA Express 的效能。使用 ENA Express ENI 時，建議您將所有 Tx 佇列擴大至網路介面的最大值，如下所示。

  您可以執行以下 ethtool 命令來調整 LLQ 大小。若要進一步了解如何控制、查詢和啟用寬型 LLQ，請參閱「Amazon Drivers GitHub 儲存庫」中適用於 ENA 文件的 Linux 核心驅動程式的大型低延遲佇列 (大型 LLQ) 主題。

  ```
  ethtool -g interface
  ```

  將 Tx 佇列設定至深度上限：

```
ethtool -G interface tx depth
```

如需 ENA Express 最佳化組態的詳細資訊，請參閱 使用 ENA Express 提高 EC2 執行個體之間的網路效能。

- 程式碼 3：具有一般 LLQ 大小和 Tx 封包流量的 ENA 會超過受支援的標頭大小上限

  根據預設，ENA LLQ 可支援最多 96 個位元組的 Tx 封包標頭大小。如果封包標頭大小大於 96 個位元組，系統會捨棄該封包。若要減少此問題，建議您啟用寬型 LLQ，這會將受支援的 Tx 封包標頭大小增加至最多 224 個位元組。

  然而，在您啟用寬型 LLQ 時，Tx 環的大小上限會從 1000 個項目縮減為 512 個項目。根據預設，系統會為所有 Nitro v4 和較新版本的執行個體類型啟用寬型 LLQ。

  - Nitro v4 執行個體類型的寬型 LLQ Tx 環大小上限預設為 512 個項目，且無法變更。
  - Nitro v5 執行個體類型預設的寬 LLQ Tx 環大小為 512 個項目，最多可增加至 1000 個項目。

  您可以執行以下 ethtool 命令來調整 LLQ 大小。若要進一步了解如何控制、查詢和啟用寬型 LLQ，請參閱「Amazon Drivers GitHub 儲存庫」中適用於 ENA 文件的 Linux 核心驅動程式的大型低延遲佇列 (大型 LLQ) 主題。

  尋找 Tx 佇列的深度上限：

  ```
  ethtool -g interface
  ```

  將 Tx 佇列設定至深度上限：

  ```
  ethtool -G interface tx depth
  ```

為彈性網路轉接器 Windows 驅動程式進行疑難排解

彈性網路轉接器 (ENA) 設計旨在提高作業系統運作狀態，並減少可能中斷 Windows 執行個體操作的預料之外硬體行為或故障。ENA 架構盡可能使裝置或驅動程式的故障對作業系統保持透明。

收集有關執行個體的診斷資訊

開啟 Windows 作業系統 (OS) 工具的步驟會有所不同，具體取決於您執行個體上安裝的作業系統版本。在以下各部分，我們使用 Run (執行) 對話方塊開啟工具，該工具在所有作業系統版本的操作方式均相同。但是，您可以使用偏好的任何方法來存取這些工具。

存取 Run (執行) 對話方塊

- 使用 Windows 標誌鍵組合：`Windows + R`
- 使用搜尋列：
  - 在搜尋列中，輸入 `run`。
  - 從搜尋結果中選擇 Run (執行) 應用程式。

某些步驟需要內容功能表才能存取屬性或內容相關動作。根據您的作業系統版本和硬體，可透過多種方式執行此作業。

存取內容功能表

- 使用滑鼠：在項目上按一下滑鼠右鍵以顯示其內容功能表。
- 使用鍵盤：
  - 根據您的作業系統版本，使用 `Shift + F10` 或 `Ctrl + Shift + F10`。
  - 如果鍵盤上有內容鍵 (方塊中帶有三條水平線)，請選擇所需項目，然後按內容鍵。

如果可以連接到自己的執行個體，請使用以下技術收集診斷資訊以進行故障診斷。

檢查 ENA 裝置狀態

若要使用 Windows Device Manager 檢查 ENA Windows 驅動程式的狀態，請按照下列步驟操作：

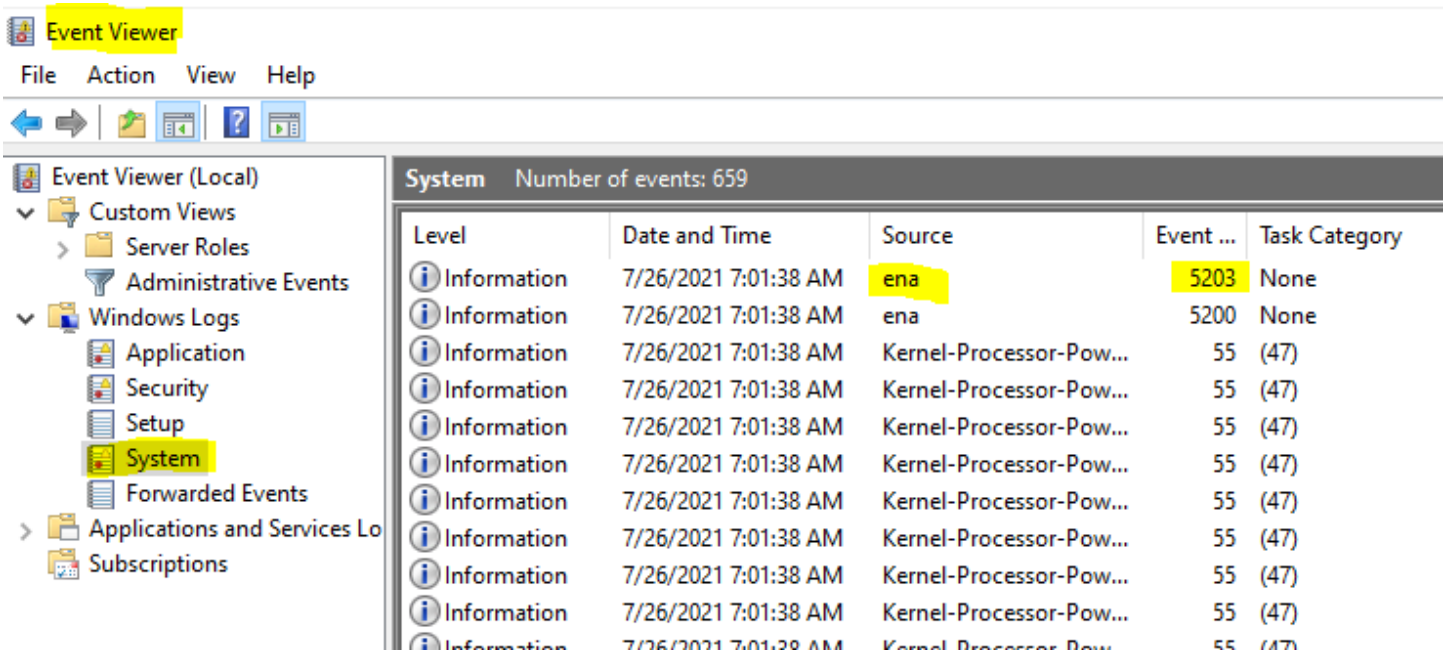
1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
3. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
4. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
5. 選取名稱，或開啟 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡) 的內容功能表，然後選擇 Properties (屬性)。這會開啟 Amazon 彈性網路介面卡屬性對話方塊。
6. 確認一般索引標籤中的訊息顯示「此裝置運作正常」。

調查驅動程式事件訊息

若要使用 Windows 事件檢視器檢視 ENA Windows 驅動程式事件日誌，請按照下列步驟操作：

1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows 事件檢視器，請在 Run (執行) 對話方塊中輸入 `eventvwr.msc`。
3. 選擇確定。Event Viewer (事件檢視器) 視窗隨即開啟。
4. 展開 Windows Logs (Windows 日誌) 功能表，然後選擇 System (系統)。
5. 在右上方面板中的 Action (動作) 下，選擇 Filter Current Log (篩選當前日誌)。此時將顯示篩選對話方塊。
6. 在 Event sources (事件來源) 方塊中，輸入 `ena`。這會將結果限製為由 ENA Windows 驅動程式產生的事件。
7. 選擇確定。此時將在視窗的詳細資訊部分顯示篩選的事件日誌結果。
8. 要深入查看詳細資訊，請從清單中選擇一條事件訊息。

下面的範例顯示了 Windows 事件檢視器系統事件清單中的 ENA 驅動程式事件：



事件訊息摘要

如下資料表顯示了 ENA Windows 驅動程式產生的事件訊息。

Input

事件 ID	ENA 驅動程式事件描述	Type
5001	硬體資源不足	錯誤

事件 ID	ENA 驅動程式事件描述	Type
5002	轉接器偵測到硬體錯誤	錯誤
5005	對於未及時完成的 NDIS 操作，轉接器存在逾時	錯誤
5032	轉接器未能重設裝置	錯誤
5200	已初始化轉接器	資訊
5201	已停止轉接器	資訊
5202	已暫停轉接器	資訊
5203	已重新啟動轉接器	資訊
5204	已關閉轉接器	資訊
5205	已重設轉接器	錯誤
5206	已意外刪除轉接器	錯誤
5208	轉接器初始化常式失敗	錯誤
5210	轉接器遇到內部問題並已成功恢復	錯誤

檢閱效能指標

ENA Windows 驅動程式會從啟用其的執行個體發佈網路效能指標。您可以使用本機效能監控應用程式檢視和啟用執行個體上的指標。如需 ENA Windows 驅動程式所產生指標的詳細資訊，請參閱 監控 EC2 執行個體的 ENA 設定網路效能。

在啟用 ENA 指標且安裝 Amazon CloudWatch 代理程式的執行個體上，CloudWatch 會收集與 Windows 效能監控中的計數器關聯的指標以及一些適用於 ENA 的進階指標。除了在 EC2 執行個體上預設啟用的指標以外，還會收集以下這些指標。如需這些指標的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的 CloudWatch 代理程式收集的指標。

> **Note**
>
> 效能指標適用於 ENA 驅動程式 2.4.0 版及更新版本 (也適用於 2.2.3 版)。因第六代 EC2 執行個體的潛在效能下降，ENA 驅動程式 2.2.4 版已回復。建議您升級至驅動程式的目前版本，確保您擁有最新的更新項目。

使用效能指標的一些方式包括：

- 對執行個體效能問題執行故障診斷。
- 為工作負載選擇合適的執行個體大小。
- 主動規劃擴展活動。
- 基準化應用程式，以確定它們是否最大限度地提高了執行個體上可用的效能。

重新整理速率

根據預設，驅動程式使用 1 秒的間隔重新整理指標。但是，擷取指標的應用程式可能會使用不同的間隔進行輪詢。您可以使用驅動程式的進階屬性，在 Device Manager (裝置管理員)中變更重新整理間隔。

若要變更 ENA Windows 驅動程式的指標重新整理間隔，請按照下列步驟操作：

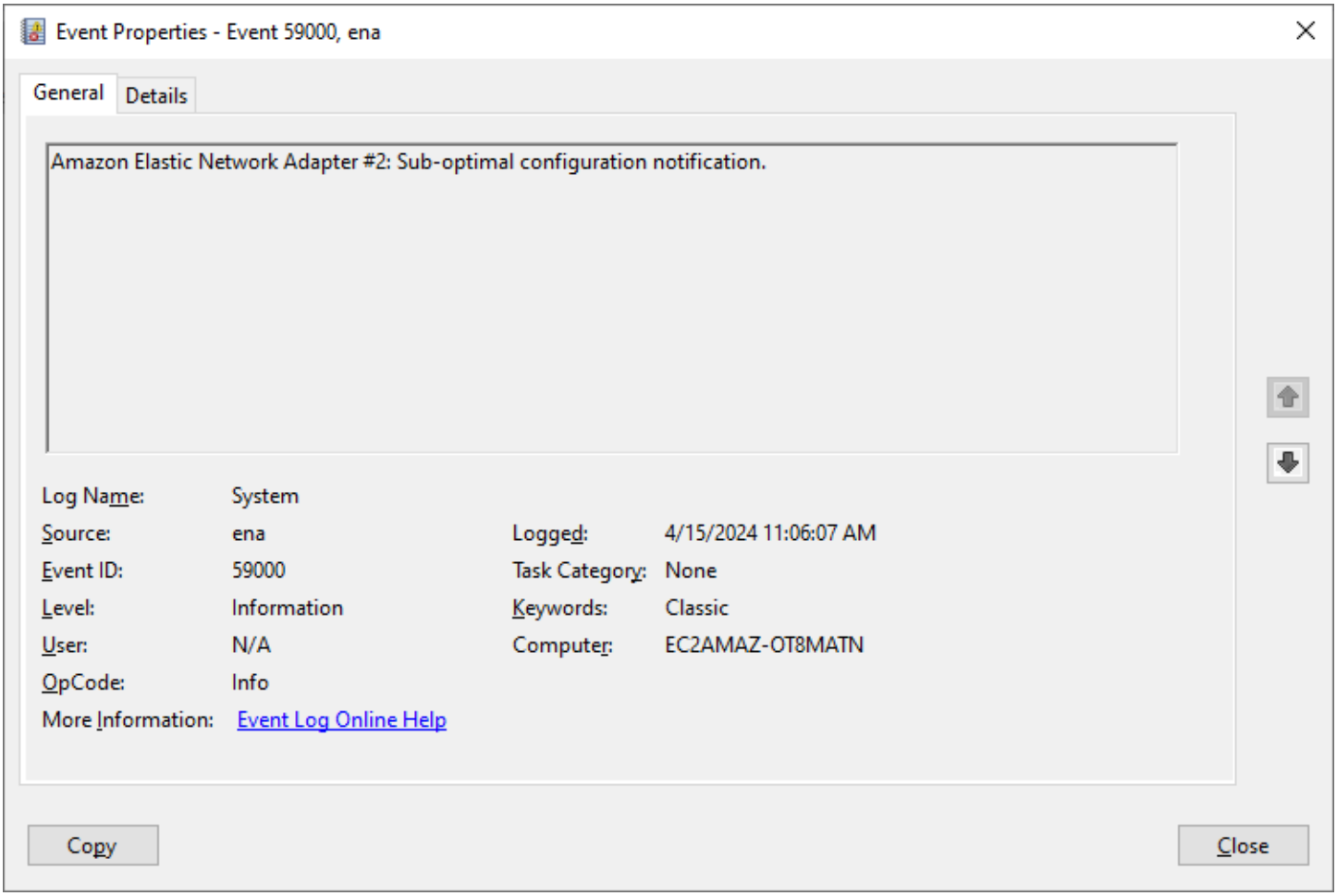
1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
3. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
4. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
5. 選取名稱，或開啟 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡) 的內容功能表，然後選擇 Properties (屬性)。這會開啟 Amazon 彈性網路介面卡屬性對話方塊。
6. 在彈出式視窗中開啟 Advanced (進階) 索引標籤。
7. 從 Property (屬性) 清單中，選擇 Metrics Refresh Interval (指標重新整理間隔) 以變更值。
8. 完成後，選擇 OK (確定)。

調查非最佳化組態通知

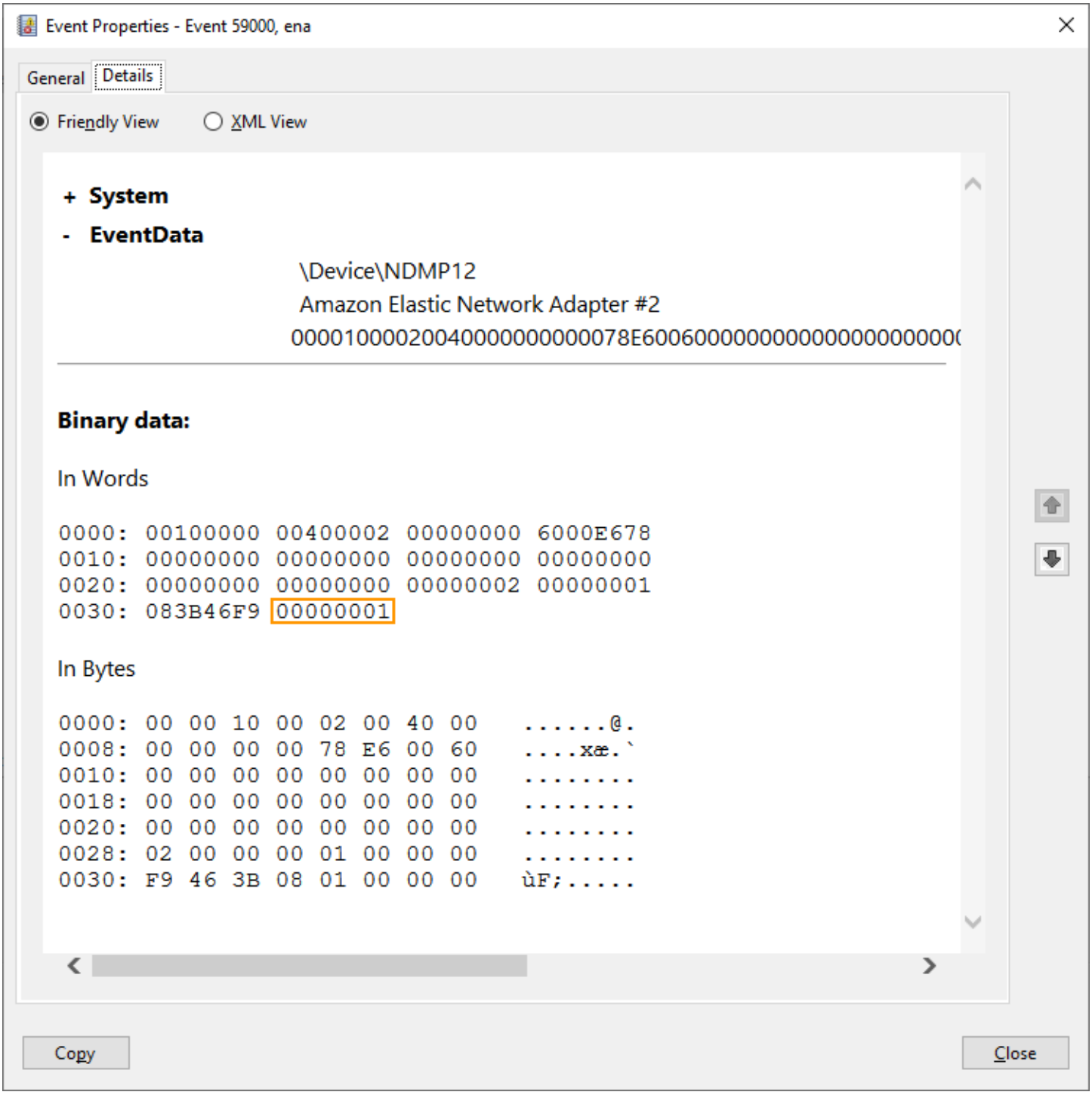
ENA 裝置會偵測驅動程式中您可以變更的非最佳化組態設定。裝置會通知 ENA 驅動程式，並記錄事件通知。在 Windows 事件檢視器中檢閱非最佳化事件

1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows 事件檢視器，請在 Run (執行) 對話方塊中輸入 `eventvwr.msc`。
3. 選擇確定。Event Viewer (事件檢視器) 視窗隨即開啟。
4. 展開 Windows Logs (Windows 日誌) 功能表，然後選擇 System (系統)。
5. 在右上方面板中的 Action (動作) 下，選擇 Filter Current Log (篩選當前日誌)。此時將顯示篩選對話方塊。
6. 在 Event sources (事件來源) 方塊中，輸入 `ena`。這會將結果限製為由 ENA Windows 驅動程式產生的事件。
7. 選擇確定。此時將在視窗的詳細資訊部分顯示篩選的事件日誌結果。

具有 ID `59000` 的事件會通知您非最佳化的組態調查結果。開啟事件的內容（按一下滑鼠右鍵）選單，然後選擇事件屬性以開啟詳細檢視，或從檢視選單中選取預覽窗格以查看相同的詳細資訊。



開啟詳細資訊索引標籤以查看事件代碼。在二進位資料：單字區段中，最後一個單字為程式碼。



下列清單顯示通知程式碼詳細資訊，以及非最佳化組態調查結果的建議動作。

- 程式碼 **1**：不建議使用帶寬 LLQ 組態的 ENA Express

  ENA Express ENI 設定有寬 LLQ。此組態不是最佳化的，可能會影響 ENA Express 的效能。我們建議您在使用 ENA Express ENI 時停用寬 LLQ 設定，如下所示。

1. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
2. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
3. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
4. 開啟 `Amazon Elastic Network Adapter` 的裝置屬性。
5. 從該處開始，請開啟進階索引標籤以進行變更。
6. 選取 LLQ 標頭大小政策屬性，並將其值設定為 `Normal (128 Bytes)`。
7. 選擇 OK (確定) 以儲存變更。

- 代碼 **2**：不建議使用具有次佳 Tx 佇列深度的 ENA Express ENI

  設定 ENA Express ENI 為具有次佳 Tx 佇列深度。此組態可能會影響 ENA Express 的效能。使用 ENA Express ENI 時，建議您將所有 Tx 佇列擴大至網路介面的最大值，如下所示。

  請依照這些步驟，將 Tx 佇列擴大至最大深度：

  1. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
  2. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
  3. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
  4. 開啟 `Amazon Elastic Network Adapter` 的裝置屬性。
  5. 從該處開始，請開啟進階索引標籤以進行變更。
  6. 選取傳輸緩衝區屬性，並將其值設定為受支援的最大值。
  7. 選擇 OK (確定) 以儲存變更。

ENA 轉接器重設

當 ENA Windows 驅動程式偵測到轉接器上的錯誤並將轉接器標記為運作狀態不良時，重設程序就會啟動。驅動程式無法自行重設，因此它依賴於作業系統來檢查轉接器的運作狀態，並叫用 ENA Windows 驅動程式的重設處理常式。重設程序可能會導致短時間內出現流量遺失。但是，TCP 連接應該能夠復原。

ENA 轉接器也可能透過傳送保持有效通知失敗而間接請求裝置重設程序。例如，如果 ENA 轉接器在載入不可恢復的組態之後達到未知狀態，它可能會停止傳送保持有效通知。

ENA 轉接器重設的常見原因

- 缺少保持有效訊息

  ENA 轉接器以固定速率張貼保持有效事件 (通常每秒一次)。ENA Windows 驅動程式實作監視程式機制，該機制會檢查這些保持有效訊息的存在。如果自上次檢查以來偵測到一條或多條新訊息，它就會記錄成功結果。否則，驅動程式得出裝置出現故障的結論，並啟動重設序列。

- 封包卡在傳輸佇列中

  ENA 轉接器驗證封包是否按預期方式通過傳輸佇列。ENA Windows 驅動程式偵測封包是否卡住，如果已卡住，則啟動重設序列。

- 記憶體映射 I/O (MMIO) 暫存器的讀取逾時

  若要限制記憶體映射 I/O (MMIO) 讀取操作，ENA Windows 驅動程式僅在初始化和重設過程中存取 MMIO 暫存器。如果驅動程式偵測到逾時，它將執行以下動作之一，具體取決於正在執行的程序：

  - 如果在初始化過程中偵測到逾時，它會讓流程失敗，從而導致驅動程式在 Windows Device Manager (裝置管理員) 中的 ENA 轉接器上顯示黃色的驚歎號。
  - 如果在重設過程中偵測到逾時，則會讓流程失敗。然後，作業系統會啟動 ENA 轉接器的意外刪除，同時透過停止並啟動已刪除的轉接器來進行復原。有關意外刪除網路界面卡 (NIC) 的詳細資訊，請參閱 Microsoft Windows 硬體開發人員文件中的處 NIC 的意外刪除。

故障診斷方案

下列方案可以幫助您對 ENA Windows 驅動程式可能發生的問題進行故障診斷：如果您沒有最新版本，我們建議您從升級 ENA 驅動程式開始操作。若要尋找 Windows 作業系統版本的最新驅動程式，請參閱 追蹤 ENA Windows 驅動程式版本版本。

非預期的 ENA 驅動程式版本已安裝

說明

在完成安裝特定版本 ENA 驅動程式的步驟之後，Windows Device Manager 會顯示 Windows 安裝了不同版本的 ENA 驅動程式。

原因

當您執行驅動程式套件的安裝時，Windows 會在開始安裝之前，對本機驅動程式存放區中指定裝置的全部有效驅動程式套件進行排名。然後其會選取最低等級值的套件作為最佳相符項目。這可能與您要

安裝的套件不同。如需有關裝置驅動程式套件選取程序的詳細資訊，請參閱 Microsoft 文件網站上的 Windows 如何為裝置選取驅動程式套件。

解決方案

若要確保 Windows 會安裝您選擇的驅動程式套件版本，可以使用 PnPUtil 命令列工具從驅動程式存放區移除排名較低的驅動程式套件。

請遵循下列步驟更新 ENA 驅動程式：

1. 連線到您的執行個體，並以本機管理員登入。
2. 開啟 Device Manager (裝置管理員) 屬性視窗，如 檢查 ENA 裝置狀態 部分中所述。這會開啟 Amazon 彈性網路介面卡屬性視窗的一般索引標籤。
3. 開啟 Driver (驅動程式) 索引標籤。
4. 選擇 Update Driver (更新驅動程式)。這會開啟更新驅動程式軟體 – Amazon 彈性網路介面卡對話方塊。
   a. 在您要如何搜尋驅動程式軟體？區段中，選擇瀏覽電腦上的驅動程式軟體。
   b. 在瀏覽電腦上的驅動程式軟體頁面中，選擇位於搜尋欄下方的讓我從電腦上的裝置驅動程式清單挑選。
   c. 在選取您要為此硬體安裝的裝置驅動程式頁面中，選擇擁有磁碟...。
   d. 在從磁碟安裝視窗中，選擇下拉式清單中檔案位置旁邊的瀏覽...。
   e. 導覽至下載目標 ENA 驅動程式套件的位置。選取名為 `ena.inf` 的檔案，然後選擇開啟。
   f. 若要開始安裝，請選擇確定，然後選擇下一步。
5. 如果安裝程式未自動重新啟動執行個體，則請執行 Restart-Computer PowerShell cmdlet。

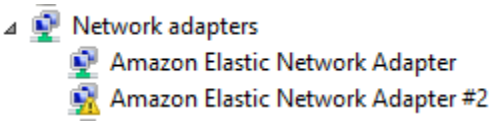
```
PS C:\> Restart-Computer
```

ENA 驅動程式的裝置警告

說明

Device Manager (裝置管理員) 的 Network adapters (網路轉接器) 部分中的 ENA 轉接器圖示顯示一個警告符號 (內部帶有驚歎號的黃色三角形)。

下面的範例顯示了在 Windows Device Manager (裝置管理員) 中帶有警告圖示的 ENA 轉接器：



原因

此裝置警告通常由環境問題引起，此問題可能需要進行更多的研究，並且通常需要執行消除程序來確定根本原因。如需完整的設備錯誤清單，請參閱 Microsoft 文件中的設備管理器錯誤訊息。

解決方案

此裝置警告的解決方案取決於根本原因。此處介紹的消除程序包括幾個基本步驟，用於幫助確定和解決可能具有簡單解決方案的最常見問題。如果這些步驟無法解決問題，則需要進行其他根本原因分析。

請按照以下步驟幫助確定和解決常見問題：

1. 停止並啟動裝置。

   開啟 Device Manager (裝置管理員) 屬性視窗，如 檢查 ENA 裝置狀態 部分中所述。這將開啟 Amazon Elastic Network Adapter Properties (Amazon 彈性網路轉接器屬性) 視窗中的 General (常規) 索引標籤，其中 Device status (裝置狀態) 顯示錯誤代碼和一條簡短訊息。

   a. 開啟 Driver (驅動程式) 索引標籤。
   b. 選擇 Disable Device (停用裝置)，在顯示的警告訊息中回應 Yes (是)。
   c. 選擇 Enable Device (啟用裝置)。
2. 停止並啟動 EC2 執行個體

   如果轉接器仍在 Device Manager (裝置管理員) 中顯示警告圖示，則下一步是停止並啟動 EC2 執行個體。在大多數情況下，這會在不同硬體上重新啟動執行個體。
3. 調查可能的執行個體資源問題

   如果您已停止並啟動 EC2 執行個體，但問題仍然存在，這可能表明您的執行個體存在資源問題，例如記憶體不足。

轉接器重設時連接逾時 (錯誤代碼 5007、5205)

說明

Windows 事件檢視器顯示 ENA 轉接器的適配器逾時以及結合其發生的事件。消息類似以下範例：

- 事件 ID 5007：Amazon 彈性網路轉接器：操作過程中逾時。

- 事件 ID 5205：Amazon 彈性網路轉接器：轉接器重設已啟動。

轉接器重設會導致最小規模的流量中斷。即使執行多個重設，它們也很少會導致任何嚴重的網路中斷。

原因

此事件序列表示 ENA Windows 驅動程式為無回應的 ENA 轉接器啟動了重設。但是，裝置驅動程式用於偵測此問題的機制受到 CPU 0 不足導致的誤報約束。

解決方案

如果這種錯誤組合頻繁發生，請檢查您的資源分配，以了解調整可能發揮作用的位置。

1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows Resource Monitor (資源監視器)，請在 Run (執行) 對話方塊中輸入 `resmon`。
3. 選擇確定。這將開啟 Resource Monitor (資源監視器) 視窗。
4. 開啟 CPU 索引標籤。每個 CPU 的用量圖表顯示在 Resource Monitor (資源監視器) 視窗的右側。
5. 檢查 CPU 0 的用量級別，了解它們是否太高。

我們建議您設定 RSS 以排除較大執行個體類型 (超過 16 個 vCPU) 上適用於 ENA 轉接器的 CPU 0。對於較小的執行個體類型，設定 RSS 可能會改善體驗，但由於可用核心數量較少，因此必須進行測試，以確保限制 CPU 核心不會對效能產生負面影響。

使用 Set-NetAdapterRss 命令為 ENA 轉接器設定 RSS，如下列範例所示。

```
Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_.InterfaceDescription -like
 "*Elastic*"}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1
```

遷移到第六代執行個體基礎設施會影響效能或配件

說明

如果遷移到第六代 EC2 執行個體，並且您尚未更新 ENA Windows 驅動程式版本，則可能會遇到效能下降或 ENA 配件故障。

原因

依據執行個體作業系統 (OS)，第六代 EC2 執行個體類型需要以下 ENA Windows 驅動程式的最低版本。

最低版本

Windows Server 版本	ENA 驅動程式版本
Windows Server 2008 R2	2.2.3 或 2.4.0
Windows Server 2012 及更新版本	2.2.3 和更新版本
Windows 工作站	2.2.3 和更新版本

解決方案

在升級至第六代 EC2 執行個體之前，請確定您用來啟動的 AMI 具有相容驅動程式 (依據執行個體作業系統)，如上表所示。如需詳細資訊，請參閱 AWS re:Post 知識中心文章在將 EC2 執行個體遷移至第六代執行個體之前，我需要做些什麼來確保獲得最高網路效能？。

彈性網路介面效能不佳

說明

ENA 介面未按預期發揮效能。

原因

效能問題的根本原因分析是一種消除程序。其中涉及太多的變數，無法指定常見原因。

解決方案

根本原因分析的第一步是檢閱未按預期發揮效能的執行個體的診斷資訊，以確定是否存在可能導致問題的錯誤。如需詳細資訊，請參閱 收集有關執行個體的診斷資訊 一節。

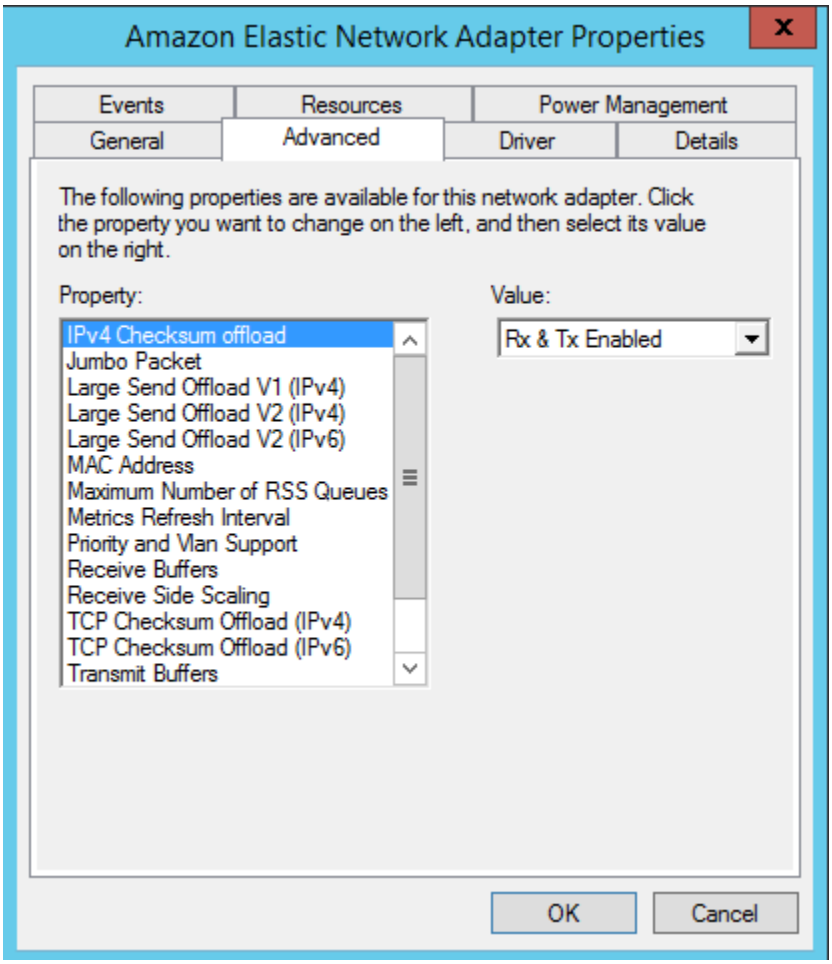
為了在執行個體上達到使用增強型聯網的最大網路效能，您可能需要修改預設的作業系統組態。根據預設，某些最佳化 (例如開啟檢查總和卸載和啟用 RSS) 已經在官方 Windows AMI 上設定。有關可套用至 ENA 轉接器的其他最佳化，請參閱 ENA 轉接器效能調整 中所示的效能調整。

我們建議您謹慎進行，並將裝置屬性調整限制在本節中列出的項目，或 AWS 支援團隊建議的特定變更。

若要變更 ENA 轉接器屬性，請按照下列步驟操作：

1. 使用上一部分中描述的其中一種方法開啟 Run (執行) 對話方塊。
2. 若要開啟 Windows Device Manager (Windows 裝置管理員)，請在 Run (執行) 對話方塊中輸入 `devmgmt.msc`。
3. 選擇確定。這將開啟 Device Manager (裝置管理員) 視窗。
4. 選取 Network adapters (網路轉接器) 左側的箭頭以展開清單。
5. 選取名稱，或開啟 Amazon Elastic Network Adapter (Amazon 彈性網路介面卡) 的內容功能表，然後選擇 Properties (屬性)。這會開啟 Amazon 彈性網路介面卡屬性對話方塊。
6. 若要進行變更，請開啟進階索引標籤。
7. 完成後，請選擇確定以儲存變更。

下面的範例顯示了 Windows Device Manager (裝置管理員) 中的 ENA 轉接器屬性：



ENA 轉接器效能調整

下表包含可調整以提高 ENA 介面效能的屬性。

Input

屬性	說明	預設值	調整
接收緩衝區	控制軟體接收佇列中的項目數目。	1024	可提高至上限 8192。
接收端調整 (RSS)	支援在多處理器系統中跨多個 CPU 高效分配網路接收處理。	已啟用	您可以將負載分散到多個處理器中。如需詳細資訊，請參閱 在 EC2 Windows 執行個體上最佳化網路效能。
最大 RSS 佇列數	設定在 RSS 已啟用時允許的 RSS 佇列數上限。	32	RSS 佇列的數量是在驅動程式初始化過程中確定的，其包括以下限制 (除其他限制外)： • 由此屬性設定的 RSS 佇列限制 • 執行個體限制 (vCPU 計數) • 硬體世代限制 (ENAv1 中最多 8 個 RSS 佇列，ENAv2 中最多 32 個 RSS 佇列) 根據執行個體和硬體世代限制，您可以設

屬性	說明	預設值	調整
			定 1-32 的值。如需詳細資訊，請參閱 在 EC2 Windows 執行個體上最佳化網路效能。
巨型封包	允許使用巨型乙太網路訊框 (超過 1500 個位元組的承載)。	已停用 (這將承載限制為 1500 個位元組或更少)	值可以設定為最高 9015，即轉換為 9001 個位元組的承載。這是巨型乙太網路訊框的最大承載。請參閱 使用巨型乙太網路訊框的考量事項。

使用巨型乙太網路訊框的考量事項

巨型訊框可增加單一封包的承載大小，這會增加不屬於封包成本的封包比例，藉此允許超過 1500 位元組的資料。傳送相同數量的可用資料，所需的封包數即會減少。但是，在下列情況下，流量限制為 MTU 最大為 1500：

- EC2 Classic 指定 AWS 區域外的流量。
- 單一 VPC 外的流量。
- 跨區域 VPC 對等互連的流量。
- VPN 連線的流量。
- 網際網路閘道的流量。

> **Note**
>
> 超過 1500 個位元組的封包會分段。如果 IP 標頭設有 `Don't Fragment` 旗標，這些封包將被捨棄。
>
> 巨型訊框應謹慎用於進出網路的流量，或任何離開 VPC 的流量。封包由中介系統切割，因此拖累流量速度。要在 VPC 內使用巨型訊框而不影響離開 VPC 的傳出流量，請嘗試以下選項之一：

- 按路由設定 MTU 大小。
- 使用具有不同 MTU 大小和不同路由的多個網路介面。

巨型訊框的建議使用案例

巨型訊框對於 VPC 內部和之間的流量非常有用。我們建議在以下使用案例中使用巨型訊框：

- 以配置於叢集置放群組內的執行個體而言，巨型訊框有助實現最大網路傳輸量。如需詳細資訊，請參閱Amazon EC2 執行個體的置放群組。
- 您可以針對透過 Direct Connect的 VPC 和現場部署網路之間的流量，使用 Jumbo Frame。如需有關使用 Direct Connect和驗證巨型訊框功能的詳細資訊，請參閱Direct Connect 《 使用者指南》中的私有虛擬介面或傳輸虛擬介面的 MTU。
- 如需傳輸閘道的支援 MTU 大小的詳細資訊，請參閱 Amazon VPC 傳輸閘道中的傳輸閘道的配額。

使用 ENA Express 提高 EC2 執行個體之間的網路效能

ENA Express 採用可擴展可靠資料包 AWS (SRD) 技術。SRD 是一種高效能的網路傳輸通訊協定，其使用動態路由來增加輸送量並將網路流量尾延遲降至最低。使用 ENA Express，您可以在相同可用區域中的兩個 EC2 執行個體之間或相同區域內的跨可用區域進行通訊。

ENA Express 的優勢

- 將單一流程可使用的最大頻寬從相同區域內的 5 Gbps 增加到 25 Gbps，直到彙總執行個體限制為止。
- 減少相同可用區域中 EC2 執行個體之間的網路流量尾端延遲，尤其是在高網路負載期間。
- 偵測並避免擁塞的網路路徑。
- 直接在網路層中處理某些任務，例如接收端的封包重新排序，以及大多數需要的重新傳輸。這會釋放應用程式層進行其他工作。

> **Note**
>
> - 如果您的應用程式具有高packets-per-second數需求，且需要在不擁塞的期間最佳化延遲，則 增強型網路 可能更適合。
> - ENA Express 流量無法在本機區域中傳送。

- 南美洲（聖保羅）、中東（巴林）和中東（阿拉伯聯合大公國）不提供可用區域之間流量的 ENA Express 支援。

在執行個體上為網路介面附件啟用 ENA Express 之後，傳送執行個體會啟動與接收執行個體的通訊，而 SRD 會偵測 ENA Express 是否同時在傳送執行個體和接收執行個體上運作。如果 ENA Express 正在運作，則通訊可以使用 SRD 傳輸。如果 ENA Express 尚未運作，則通訊會回復為標準 ENA 傳輸。

在網路流量較輕的期間，當封包使用 ENA Express 時，您可能會注意到封包延遲中位數（數十微秒）稍微增加。在這段時間內，優先考慮特定網路效能特性的應用程式可以從 ENA Express 中受益，如下所示：

- 程序可以受益於將單一流量頻寬上限從同一區域內的 5 Gbps 增加到 25 Gbps，直到彙總執行個體限制為止。例如，如果特定執行個體類型最高支援 12.5 Gbps，則單一流量頻寬也會限制為 12.5 Gbps。
- 在相同可用區域中執行時間較長的程序，會在網路擁塞期間降低尾部延遲。
- 程序可以從更順暢且更標準的網路回應時間分佈中獲益。

主題

- ENA Express 的工作原理
- ENA Express 支援的執行個體類型
- 調校 Linux 執行個體上 ENA Express 設定的效能
- 檢閱 EC2 執行個體的 ENA Express 設定
- 設定 EC2 執行個體的 ENA Express 設定

ENA Express 的工作原理

ENA Express 採用可擴展可靠資料包 AWS (SRD) 技術。它會將每個網路流程的封包分配到不同的 AWS 網路路徑，並在偵測到擁塞跡象時動態調整分佈。它還管理接收端的封包重新排序。

若要確保 ENA Express 可以按預期管理網路流量，傳送和接收執行個體及其之間的通訊必須符合下列所有要求：

- 支援傳送和接收執行個體類型。如需詳細資訊，請參閱 ENA Express 支援的執行個體類型 資料表。
- 傳送和接收執行個體都必須設定 ENA Express。如果組態有任何差異，您可能會遇到流量預設為標準 ENA 傳輸的情況。下列情境說明會發生的情況。

情況：組態的差異

執行個體	ENA Express 已啟用	UDP 使用 ENA Express
執行個體 1	是	是
執行個體 2	是	否

在這種情況下，兩個執行個體之間的 TCP 流量可以使用 ENA Express，因為這兩個執行個體皆已將其啟用。但是，由於其中一個執行個體並未將 ENA Express 用於 UDP 流量，因此這兩個執行個體透過 UDP 之間的通訊會使用標準 ENA 傳輸。

- 傳送和接收執行個體必須在相同的 區域中執行。
- 執行個體之間的網路路徑不得包含中介軟體方塊。ENA Express 目前不支援中介軟體。
- (僅 Linux 執行個體) 若要充分利用全部頻寬潛力，請使用 2.2.9 或更新版本的驅動程式。
- (僅 Linux 執行個體) 若要產生指標，請使用 2.8 或更新版本的驅動程式。

如果未滿足任意要求，執行個體會使用標準 TCP/UDP 通訊協定但不帶 SRD 進行通訊。

若要確保您的執行個體網路驅動程式設定為最佳效能，請參閱適用於 ENA 驅動程式的建議最佳實務。這些最佳實務也適用於 ENA Express。如需詳細資訊，請參閱 GitHub 網站上的 ENA Linux 驅動程式最佳實務與效能最佳化指南。

> **Note**
>
> Amazon EC2 是指執行個體與以附件形式連接到執行個體的網路介面之間的關係。ENA Express 設定會套用至附件。如果網路介面從執行個體中斷連接，則附件將不再存在，而套用至該執行個體的 ENA Express 設定將不再有效。執行個體終止時也是如此，即使網路介面仍然存在。

在傳送執行個體和接收執行個體上啟用網路介面附件的 ENA Express 之後，您可以使用 ENA Express 指標來協助確保您的執行個體充分利用 SRD 技術提供的效能改進。如需 ENA Express 指標的詳細資訊，請參閱 ENA Express 的指標。

ENA Express 支援的執行個體類型

下列執行個體類型支援 ENA Express。

General purpose

執行個體類型	Architecture
m6a.12xlarge	x86_64
m6a.16xlarge	x86_64
m6a.24xlarge	x86_64
m6a.32xlarge	x86_64
m6a.48xlarge	x86_64
m6a.metal	x86_64
m6i.8xlarge	x86_64
m6i.12xlarge	x86_64
m6i.16xlarge	x86_64
m6i.24xlarge	x86_64
m6i.32xlarge	x86_64
m6i.metal	x86_64
m6id.8xlarge	x86_64
m6id.12xlarge	x86_64
m6id.16xlarge	x86_64
m6id.24xlarge	x86_64
m6id.32xlarge	x86_64

執行個體類型	Architecture
m6id.metal	x86_64
m6idn.8xlarge	x86_64
m6idn.12xlarge	x86_64
m6idn.16xlarge	x86_64
m6idn.24xlarge	x86_64
m6idn.32xlarge	x86_64
m6idn.metal	x86_64
m6in.8xlarge	x86_64
m6in.12xlarge	x86_64
m6in.16xlarge	x86_64
m6in.24xlarge	x86_64
m6in.32xlarge	x86_64
m6in.metal	x86_64
m7a.12xlarge	x86_64
m7a.16xlarge	x86_64
m7a.24xlarge	x86_64
m7a.32xlarge	x86_64
m7a.48xlarge	x86_64
m7a.metal-48xl	x86_64
m7g.12xlarge	arm64

執行個體類型	Architecture
m7g.16xlarge	arm64
m7g.metal	arm64
m7gd.12xlarge	arm64
m7gd.16xlarge	arm64
m7gd.metal	arm64
m7i.12xlarge	x86_64
m7i.16xlarge	x86_64
m7i.24xlarge	x86_64
m7i.48xlarge	x86_64
m7i.metal-24xl	x86_64
m7i.metal-48xl	x86_64
m8a.16xlarge	x86_64
m8a.24xlarge	x86_64
m8a.48xlarge	x86_64
m8a.metal-24xl	x86_64
m8a.metal-48xl	x86_64
m8azn.12xlarge	x86_64
m8azn.24xlarge	x86_64
m8azn.metal-12xl	x86_64
m8azn.metal-24xl	x86_64

執行個體類型	Architecture
m8g.8xlarge	arm64
m8g.12xlarge	arm64
m8g.16xlarge	arm64
m8g.24xlarge	arm64
m8g.48xlarge	arm64
m8g.metal-24xl	arm64
m8g.metal-48xl	arm64
m8gb.4xlarge	arm64
m8gb.8xlarge	arm64
m8gb.12xlarge	arm64
m8gb.16xlarge	arm64
m8gb.24xlarge	arm64
m8gb.48xlarge	arm64
m8gb.metal-24xl	arm64
m8gb.metal-48xl	arm64
m8gd.8xlarge	arm64
m8gd.12xlarge	arm64
m8gd.16xlarge	arm64
m8gd.24xlarge	arm64
m8gd.48xlarge	arm64

執行個體類型	Architecture
m8gd.metal-24xl	arm64
m8gd.metal-48xl	arm64
m8gn.4xlarge	arm64
m8gn.8xlarge	arm64
m8gn.12xlarge	arm64
m8gn.16xlarge	arm64
m8gn.24xlarge	arm64
m8gn.48xlarge	arm64
m8gn.metal-24xl	arm64
m8gn.metal-48xl	arm64
m8i.8xlarge	x86_64
m8i.12xlarge	x86_64
m8i.16xlarge	x86_64
m8i.24xlarge	x86_64
m8i.32xlarge	x86_64
m8i.48xlarge	x86_64
m8i.96xlarge	x86_64
m8i.metal-48xl	x86_64
m8i.metal-96xl	x86_64
m8id.8xlarge	x86_64

執行個體類型	Architecture
m8id.12xlarge	x86_64
m8id.16xlarge	x86_64
m8id.24xlarge	x86_64
m8id.32xlarge	x86_64
m8id.48xlarge	x86_64
m8id.96xlarge	x86_64
m8id.metal-48xl	x86_64
m8id.metal-96xl	x86_64
m8in.8xlarge	x86_64
m8in.12xlarge	x86_64
m8in.16xlarge	x86_64
m8in.24xlarge	x86_64
m8in.32xlarge	x86_64
m8in.48xlarge	x86_64
m8in.96xlarge	x86_64
m8in.metal-48xl	x86_64
m8in.metal-96xl	x86_64
m8idn.8xlarge	x86_64
m8idn.12xlarge	x86_64
m8idn.16xlarge	x86_64

執行個體類型	Architecture
m8idn.24xlarge	x86_64
m8idn.32xlarge	x86_64
m8idn.48xlarge	x86_64
m8idn.96xlarge	x86_64
m8idn.metal-48xl	x86_64
m8idn.metal-96xl	x86_64
m8ib.8xlarge	x86_64
m8ib.12xlarge	x86_64
m8ib.16xlarge	x86_64
m8ib.24xlarge	x86_64
m8ib.32xlarge	x86_64
m8ib.48xlarge	x86_64
m8ib.96xlarge	x86_64
m8ib.metal-48xl	x86_64
m8ib.metal-96xl	x86_64
m8idb.8xlarge	x86_64
m8idb.12xlarge	x86_64
m8idb.16xlarge	x86_64
m8idb.24xlarge	x86_64
m8idb.32xlarge	x86_64

執行個體類型	Architecture
m8idb.48xlarge	x86_64
m8idb.96xlarge	x86_64
m8idb.metal-48xl	x86_64
m8idb.metal-96xl	x86_64
m9g.8xlarge	arm64
m9g.12xlarge	arm64
m9g.16xlarge	arm64
m9g.24xlarge	arm64
m9g.48xlarge	arm64
m9g.metal-48xl	arm64
m9gd.8xlarge	arm64
m9gd.12xlarge	arm64
m9gd.16xlarge	arm64
m9gd.24xlarge	arm64
m9gd.48xlarge	arm64
m9gd.metal-48xl	arm64

Compute optimized

執行個體類型	Architecture
c6a.12xlarge	x86_64

執行個體類型	Architecture
c6a.16xlarge	x86_64
c6a.24xlarge	x86_64
c6a.32xlarge	x86_64
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6gn.4xlarge	arm64
c6gn.8xlarge	arm64
c6gn.12xlarge	arm64
c6gn.16xlarge	arm64
c6i.8xlarge	x86_64
c6i.12xlarge	x86_64
c6i.16xlarge	x86_64
c6i.24xlarge	x86_64
c6i.32xlarge	x86_64
c6i.metal	x86_64
c6id.8xlarge	x86_64
c6id.12xlarge	x86_64
c6id.16xlarge	x86_64
c6id.24xlarge	x86_64
c6id.32xlarge	x86_64

執行個體類型	Architecture
c6id.metal	x86_64
c6in.8xlarge	x86_64
c6in.12xlarge	x86_64
c6in.16xlarge	x86_64
c6in.24xlarge	x86_64
c6in.32xlarge	x86_64
c6in.metal	x86_64
c7a.12xlarge	x86_64
c7a.16xlarge	x86_64
c7a.24xlarge	x86_64
c7a.32xlarge	x86_64
c7a.48xlarge	x86_64
c7a.metal-48xl	x86_64
c7g.12xlarge	arm64
c7g.16xlarge	arm64
c7g.metal	arm64
c7gd.12xlarge	arm64
c7gd.16xlarge	arm64
c7gd.metal	arm64
c7gn.4xlarge	arm64

執行個體類型	Architecture
c7gn.8xlarge	arm64
c7gn.12xlarge	arm64
c7gn.16xlarge	arm64
c7gn.metal	arm64
c7i.12xlarge	x86_64
c7i.16xlarge	x86_64
c7i.24xlarge	x86_64
c7i.48xlarge	x86_64
c7i.metal-24xl	x86_64
c7i.metal-48xl	x86_64
c8a.16xlarge	x86_64
c8a.24xlarge	x86_64
c8a.48xlarge	x86_64
c8a.metal-24xl	x86_64
c8a.metal-48xl	x86_64
c8g.8xlarge	arm64
c8g.12xlarge	arm64
c8g.16xlarge	arm64
c8g.24xlarge	arm64
c8g.48xlarge	arm64

執行個體類型	Architecture
c8g.metal-24xl	arm64
c8g.metal-48xl	arm64
c8gb.4xlarge	arm64
c8gb.8xlarge	arm64
c8gb.12xlarge	arm64
c8gb.16xlarge	arm64
c8gb.24xlarge	arm64
c8gb.48xlarge	arm64
c8gb.metal-24xl	arm64
c8gb.metal-48xl	arm64
c8gd.8xlarge	arm64
c8gd.12xlarge	arm64
c8gd.16xlarge	arm64
c8gd.24xlarge	arm64
c8gd.48xlarge	arm64
c8gd.metal-24xl	arm64
c8gd.metal-48xl	arm64
c8gn.4xlarge	arm64
c8gn.8xlarge	arm64
c8gn.12xlarge	arm64

執行個體類型	Architecture
c8gn.16xlarge	arm64
c8gn.24xlarge	arm64
c8gn.48xlarge	arm64
c8gn.metal-24xl	arm64
c8gn.metal-48xl	arm64
c8i.8xlarge	x86_64
c8i.12xlarge	x86_64
c8i.16xlarge	x86_64
c8i.24xlarge	x86_64
c8i.32xlarge	x86_64
c8i.48xlarge	x86_64
c8i.96xlarge	x86_64
c8i.metal-48xl	x86_64
c8i.metal-96xl	x86_64
c8id.8xlarge	x86_64
c8id.12xlarge	x86_64
c8id.16xlarge	x86_64
c8id.24xlarge	x86_64
c8id.32xlarge	x86_64
c8id.48xlarge	x86_64

執行個體類型	Architecture
c8id.96xlarge	x86_64
c8id.metal-48xl	x86_64
c8id.metal-96xl	x86_64
c8in.8xlarge	x86_64
c8in.12xlarge	x86_64
c8in.16xlarge	x86_64
c8in.24xlarge	x86_64
c8in.32xlarge	x86_64
c8in.48xlarge	x86_64
c8in.96xlarge	x86_64
c8in.metal-48xl	x86_64
c8in.metal-96xl	x86_64
c8ib.8xlarge	x86_64
c8ib.12xlarge	x86_64
c8ib.16xlarge	x86_64
c8ib.24xlarge	x86_64
c8ib.32xlarge	x86_64
c8ib.48xlarge	x86_64
c8ib.96xlarge	x86_64
c8ib.metal-48xl	x86_64

執行個體類型	Architecture
c8ib.metal-96xl	x86_64
c9g.8xlarge	arm64
c9g.12xlarge	arm64
c9g.16xlarge	arm64
c9g.24xlarge	arm64
c9g.48xlarge	arm64
c9g.metal-48xl	arm64
c9gd.8xlarge	arm64
c9gd.12xlarge	arm64
c9gd.16xlarge	arm64
c9gd.24xlarge	arm64
c9gd.48xlarge	arm64
c9gd.metal-48xl	arm64

Memory optimized

執行個體類型	Architecture
r6a.12xlarge	x86_64
r6a.16xlarge	x86_64
r6a.24xlarge	x86_64
r6a.32xlarge	x86_64

執行個體類型	Architecture
r6a.48xlarge	x86_64
r6a.metal	x86_64
r6i.8xlarge	x86_64
r6i.12xlarge	x86_64
r6i.16xlarge	x86_64
r6i.24xlarge	x86_64
r6i.32xlarge	x86_64
r6i.metal	x86_64
r6id.8xlarge	x86_64
r6id.12xlarge	x86_64
r6id.16xlarge	x86_64
r6id.24xlarge	x86_64
r6id.32xlarge	x86_64
r6id.metal	x86_64
r6idn.8xlarge	x86_64
r6idn.12xlarge	x86_64
r6idn.16xlarge	x86_64
r6idn.24xlarge	x86_64
r6idn.32xlarge	x86_64
r6idn.metal	x86_64

執行個體類型	Architecture
r6in.8xlarge	x86_64
r6in.12xlarge	x86_64
r6in.16xlarge	x86_64
r6in.24xlarge	x86_64
r6in.32xlarge	x86_64
r6in.metal	x86_64
r7a.12xlarge	x86_64
r7a.16xlarge	x86_64
r7a.24xlarge	x86_64
r7a.32xlarge	x86_64
r7a.48xlarge	x86_64
r7a.metal-48xl	x86_64
r7g.12xlarge	arm64
r7g.16xlarge	arm64
r7g.metal	arm64
r7gd.12xlarge	arm64
r7gd.16xlarge	arm64
r7gd.metal	arm64
r7i.12xlarge	x86_64
r7i.16xlarge	x86_64

執行個體類型	Architecture
r7i.24xlarge	x86_64
r7i.48xlarge	x86_64
r7i.metal-24xl	x86_64
r7i.metal-48xl	x86_64
r7iz.8xlarge	x86_64
r7iz.12xlarge	x86_64
r7iz.16xlarge	x86_64
r7iz.32xlarge	x86_64
r7iz.metal-16xl	x86_64
r7iz.metal-32xl	x86_64
r8a.16xlarge	x86_64
r8a.24xlarge	x86_64
r8a.48xlarge	x86_64
r8a.metal-24xl	x86_64
r8a.metal-48xl	x86_64
r8g.8xlarge	arm64
r8g.12xlarge	arm64
r8g.16xlarge	arm64
r8g.24xlarge	arm64
r8g.48xlarge	arm64

執行個體類型	Architecture
r8g.metal-24xl	arm64
r8g.metal-48xl	arm64
r8gb.4xlarge	arm64
r8gb.8xlarge	arm64
r8gb.12xlarge	arm64
r8gb.16xlarge	arm64
r8gb.24xlarge	arm64
r8gb.48xlarge	arm64
r8gb.metal-24xl	arm64
r8gb.metal-48xl	arm64
r8gd.8xlarge	arm64
r8gd.12xlarge	arm64
r8gd.16xlarge	arm64
r8gd.24xlarge	arm64
r8gd.48xlarge	arm64
r8gd.metal-24xl	arm64
r8gd.metal-48xl	arm64
r8gn.4xlarge	arm64
r8gn.8xlarge	arm64
r8gn.12xlarge	arm64

執行個體類型	Architecture
r8gn.16xlarge	arm64
r8gn.24xlarge	arm64
r8gn.48xlarge	arm64
r8gn.metal-24xl	arm64
r8gn.metal-48xl	arm64
r8i.8xlarge	x86_64
r8i.12xlarge	x86_64
r8i.16xlarge	x86_64
r8i.24xlarge	x86_64
r8i.32xlarge	x86_64
r8i.48xlarge	x86_64
r8i.96xlarge	x86_64
r8i.metal-48xl	x86_64
r8i.metal-96xl	x86_64
r8id.8xlarge	x86_64
r8id.12xlarge	x86_64
r8id.16xlarge	x86_64
r8id.24xlarge	x86_64
r8id.32xlarge	x86_64
r8id.48xlarge	x86_64

執行個體類型	Architecture
r8id.96xlarge	x86_64
r8id.metal-48xl	x86_64
r8id.metal-96xl	x86_64
r8in.8xlarge	x86_64
r8in.12xlarge	x86_64
r8in.16xlarge	x86_64
r8in.24xlarge	x86_64
r8in.32xlarge	x86_64
r8in.48xlarge	x86_64
r8in.96xlarge	x86_64
r8in.metal-48xl	x86_64
r8in.metal-96xl	x86_64
r8idn.8xlarge	x86_64
r8idn.12xlarge	x86_64
r8idn.16xlarge	x86_64
r8idn.24xlarge	x86_64
r8idn.32xlarge	x86_64
r8idn.48xlarge	x86_64
r8idn.96xlarge	x86_64
r8idn.metal-48xl	x86_64

執行個體類型	Architecture
r8idn.metal-96xl	x86_64
r8ib.8xlarge	x86_64
r8ib.12xlarge	x86_64
r8ib.16xlarge	x86_64
r8ib.24xlarge	x86_64
r8ib.32xlarge	x86_64
r8ib.48xlarge	x86_64
r8ib.96xlarge	x86_64
r8ib.metal-48xl	x86_64
r8ib.metal-96xl	x86_64
r8idb.8xlarge	x86_64
r8idb.12xlarge	x86_64
r8idb.16xlarge	x86_64
r8idb.24xlarge	x86_64
r8idb.32xlarge	x86_64
r8idb.48xlarge	x86_64
r8idb.96xlarge	x86_64
r8idb.metal-48xl	x86_64
r8idb.metal-96xl	x86_64
r9g.8xlarge	arm64

執行個體類型	Architecture
r9g.12xlarge	arm64
r9g.16xlarge	arm64
r9g.24xlarge	arm64
r9g.48xlarge	arm64
r9g.metal-48xl	arm64
r9gd.8xlarge	arm64
r9gd.12xlarge	arm64
r9gd.16xlarge	arm64
r9gd.24xlarge	arm64
r9gd.48xlarge	arm64
r9gd.metal-48xl	arm64
u7i-6tb.112xlarge	x86_64
u7i-8tb.112xlarge	x86_64
u7i-12tb.224xlarge	x86_64
u7in-16tb.224xlarge	x86_64
u7in-24tb.224xlarge	x86_64
u7in-32tb.224xlarge	x86_64
u7inh-32tb.480xlarge	x86_64
x2idn.16xlarge	x86_64
x2idn.24xlarge	x86_64

執行個體類型	Architecture
x2idn.32xlarge	x86_64
x2idn.metal	x86_64
x2iedn.8xlarge	x86_64
x2iedn.16xlarge	x86_64
x2iedn.24xlarge	x86_64
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64
x8g.8xlarge	arm64
x8g.12xlarge	arm64
x8g.16xlarge	arm64
x8g.24xlarge	arm64
x8g.48xlarge	arm64
x8g.metal-24xl	arm64
x8g.metal-48xl	arm64
x8aedz.24xlarge	x86_64
x8aedz.metal-24xl	x86_64
x8i.8xlarge	x86_64
x8i.12xlarge	x86_64
x8i.16xlarge	x86_64
x8i.24xlarge	x86_64

執行個體類型	Architecture
x8i.32xlarge	x86_64
x8i.48xlarge	x86_64
x8i.64xlarge	x86_64
x8i.96xlarge	x86_64
x8i.metal-48xl	x86_64
x8i.metal-96xl	x86_64

Accelerated computing

執行個體類型	Architecture
g6.48xlarge	x86_64
g6e.12xlarge	x86_64
g6e.24xlarge	x86_64
g6e.48xlarge	x86_64
g7.24xlarge	x86_64
g7.48xlarge	x86_64
g7e.12xlarge	x86_64
g7e.24xlarge	x86_64
g7e.48xlarge	x86_64
p5.4xlarge	x86_64
p5.48xlarge	x86_64

執行個體類型	Architecture
p5e.48xlarge	x86_64
p5en.48xlarge	x86_64
p6-b200.48xlarge	x86_64
p6-b300.48xlarge	x86_64
p6e-gb200.36xlarge	arm64

Storage optimized

執行個體類型	Architecture
i4g.4xlarge	arm64
i4g.8xlarge	arm64
i4g.16xlarge	arm64
i4i.8xlarge	x86_64
i4i.12xlarge	x86_64
i4i.16xlarge	x86_64
i4i.24xlarge	x86_64
i4i.32xlarge	x86_64
i4i.metal	x86_64
i7i.12xlarge	x86_64
i7i.16xlarge	x86_64
i7i.24xlarge	x86_64

執行個體類型	Architecture
i7i.48xlarge	x86_64
i7i.metal-24xl	x86_64
i7i.metal-48xl	x86_64
i7ie.6xlarge	x86_64
i7ie.12xlarge	x86_64
i7ie.18xlarge	x86_64
i7ie.24xlarge	x86_64
i7ie.48xlarge	x86_64
i7ie.metal-24xl	x86_64
i7ie.metal-48xl	x86_64
i8g.8xlarge	arm64
i8g.12xlarge	arm64
i8g.16xlarge	arm64
i8g.24xlarge	arm64
i8g.48xlarge	arm64
i8g.metal-24xl	arm64
i8g.metal-48xl	arm64
i8ge.6xlarge	arm64
i8ge.12xlarge	arm64
i8ge.18xlarge	arm64

執行個體類型	Architecture
i8ge.24xlarge	arm64
i8ge.48xlarge	arm64
i8ge.metal-24xl	arm64
i8ge.metal-48xl	arm64
im4gn.4xlarge	arm64
im4gn.8xlarge	arm64
im4gn.16xlarge	arm64

調校 Linux 執行個體上 ENA Express 設定的效能

為了確保 ENA Express 可以有效運作，您的 Linux 執行個體必須符合數個網路組態要求。

您可以從 Amazon GitHub 儲存庫下載並執行 ENA Express 設定檢查指令碼，而不是手動設定每個設定。指令碼會根據 ENA Express 的必要和建議設定來驗證您的執行個體，並輸出確切的命令來修正其發現的任何問題。

https://github.com/amzn/amzn-ec2-ena-utilities/blob/main/ena-express/check-ena-express-settings.sh

指令碼會檢查下列設定和組態：

- MTU 大小 – ENA Express 需要低於預設值的 MTU，才能容納其他 AWS SRD 標頭。新建立的 TCP 連線會自動固定 MSS 以緩解此問題，但 UDP 流量仍需要較低的 MTU。
- TCP 輸出佇列大小限制 – 檢查每個通訊端的傳輸中位元組限制是否足以維持高輸送量。網路延遲增加的環境需要更高的限制。
- 位元組佇列限制 – 確認網路界面上已停用位元組佇列限制 (BQL)。BQL 可以限制為裝置層級傳輸排入佇列的資料量，這會限制 ENA Express 效能。

> **Note**
>
> Amazon Linux 發行版本的 ENA 驅動程式預設會停用位元組佇列限制。

- TCP Autocorking – 檢查是否停用 TCP Autocorking。停用自動追蹤可以減少某些 ENA Express TCP 流量模式的延遲，例如請求回應工作負載。這可能會導致封包處理額外負荷的最小增加。
- TX 佇列大小和大型 LLQ – 驗證網路界面的傳輸佇列大小是否足夠大，以獲得最佳效能。指令碼也會檢查 ENA 模組參數是否明確停用大型低延遲佇列（大型 LLQ) 功能，因為它可以減少可用的 TX 佇列深度。如需大型 LLQ 及其對 TX 佇列大小影響的詳細資訊，請參閱 GitHub 上的大型低延遲佇列（大型 LLQ)。
- RX 佇列大小 – 檢查網路界面的接收環形緩衝區是否足夠大，以有效率地處理傳入流量，並避免封包在負載下下降。
- TCP 和網路通訊端緩衝區大小 – 驗證 TCP 接收和傳送緩衝區大小上限，以及核心網路通訊端緩衝區預設值和上限是否足夠大，以維持高輸送量。這些設定在網路延遲增加的環境中很重要，您需要較大的緩衝區才能使用連線。
- TCP 擁塞控制 – 驗證 TCP 擁塞控制組態已最佳化，可在網路延遲增加的環境中與 ENA Express 搭配使用。

指令碼也會報告其他診斷資訊，包括 ENA 驅動程式版本、EMA SRD 統計資料、中斷管制設定、佇列組態和通訊端緩衝區大小。此資訊有助於對 ENA Express 效能問題進行故障診斷。

為了確保您的執行個體網路驅動程式已設定為最佳效能，請同時檢閱 GitHub 上的 ENA Linux 驅動程式最佳實務和效能最佳化指南。

檢閱 EC2 執行個體的 ENA Express 設定

您可以依執行個體或網路介面驗證 ENA Express 設定。要更新 ENA Express 設定，請參閱 the section called “設定執行個體設定”。

Console

檢視網路介面的 ENA Express 設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取網路介面以查看該執行個體的詳細資訊。您可以選擇 Network interface ID (網路介面 ID) 連結，以開啟詳細資訊頁面；或選取清單左側的核取方塊，以在頁面底部的詳細資訊窗格中檢視詳細資訊。
4. 在詳細資訊索引標籤或詳細資訊頁面的網路介面附件區段中，檢視 ENA Express 和 ENA Express UDP 的設定。

檢視執行個體的 ENA Express 設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取執行個體以查看該執行個體的詳細資訊。您可以選擇 Instance ID (執行個體 ID) 連結，以開啟詳細資訊頁面；或選取清單左側的核取方塊，以在頁面底部的詳細資訊窗格中檢視詳細資訊。
4. 在 Networking (聯網) 標籤的 Network interfaces (網路介面) 區段中，向右捲動以檢視 ENA Express 和 ENA Express UDP 的設定。

AWS CLI

取得執行個體的 ENA Express 設定

使用 describe-instances 命令。此命令範例會傳回連接到 `--instance-ids` 參數所指定之每個執行中執行個體，以及其網路介面的 ENA Express 組態清單。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7 \
    --query 'Reservations[*].Instances[*].[InstanceId,
 NetworkInterfaces[*].Attachment.EnaSrdSpecification]'
```

以下為範例輸出。

```
[
    [
        [
            "i-1234567890abcdef0",
            [
                {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ],
    [
        [
```

```
            "i-0598c7d356eba48d7",
            [
            {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ]
]
```

取得網路介面的 ENA Express 設定

使用 describe-network-interfaces 命令。

```
aws ec2 describe-network-interfaces \
    --network-interface-ids eni-1234567890abcdef0 \
    --query NetworkInterfaces[].[NetworkInterfaceId,Attachment.EnaSrdSpecification]
```

以下為範例輸出。

```
[
    [
        "eni-1234567890abcdef0",
        {
            "EnaSrdEnabled": true,
            "EnaSrdUdpSpecification": {
                "EnaSrdUdpEnabled": false
            }
        }
    ]
]
```

PowerShell

取得網路介面的 ENA Express 設定

使用 Get-EC2NetworkInterface cmdlet。

```
Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 | `
```

```
Select-Object `
    Association,
    NetworkInterfaceId,
    OwnerId,
    @{Name = 'AttachTime'; Expression = { $_.Attachment.AttachTime } },
    @{Name = 'AttachmentId'; Expression = { $_.Attachment.AttachmentId } },
    @{Name = 'DeleteOnTermination'; Expression =
 { $_.Attachment.DeleteOnTermination } },
    @{Name = 'NetworkCardIndex'; Expression = { $_.Attachment.NetworkCardIndex } },
    @{Name = 'InstanceId'; Expression = { $_.Attachment.InstanceId } },
    @{Name = 'InstanceOwnerId'; Expression = { $_.Attachment.InstanceOwnerId } },
    @{Name = 'Status'; Expression = { $_.Attachment.Status } },
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled } },
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled } }
```

以下為範例輸出。

```
Association         :
NetworkInterfaceId  : eni-0d1234e5f6a78901b
OwnerId             : 111122223333
AttachTime          : 6/11/2022 1:13:11 AM
AttachmentId        : eni-attach-0d1234e5f6a78901b
DeleteOnTermination : True
NetworkCardIndex    : 0
InstanceId          : i-1234567890abcdef0
InstanceOwnerId     : 111122223333
Status              : attached
EnaSrdEnabled       : True
EnaSrdUdpEnabled    : False
```

設定 EC2 執行個體的 ENA Express 設定

您可以針對支援的 EC2 執行個體類型設定 ENA Express，而無需安裝任何其他軟體。如需詳細資訊，請參閱the section called “ENA Express 支援的執行個體類型”。

Console

管理網路介面的 ENA Express

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在左側導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取要連接到執行個體的網路介面。您可以選擇 Network interface ID (網路介面 ID) 連結，以開啟詳細資訊頁面；或選取清單左側的核取方塊。
4. 從頁面右上角的 Action (動作) 功能表中選擇 Manage ENA Express (管理 ENA Express)。此操作會開啟 Manage ENA Express (管理 ENA Express) 對話方塊，並顯示選取的網路介面 ID 和目前設定。

   如果您選取的網路介面未連接至執行個體，則此動作不會顯示在功能表中。
5. 若要使用 ENA Express，請選取啟用核取方塊。
6. 啟用 ENA Express 時，您可以設定 UDP 設定。若要使用 ENA Express UDP，請選取啟用核取方塊。
7. 若要儲存設定，請選擇 Save (儲存)。

管理執行個體的 ENA Express

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取您要管理的執行個體。您可以選擇 Instance ID (執行個體 ID) 連結，以開啟詳細資訊頁面；或選取清單左側的核取方塊。
4. 選取要為您的執行個體設定的 Network interface (網路介面)。
5. 從頁面右上角的 Action (動作) 功能表中選擇 Manage ENA Express (管理 ENA Express)。
6. 若要為連接至執行個體的網路介面設定 ENA Express，請從 Network interface (網路介面) 清單中進行選取。
7. 若要將 ENA Express 用於選取的網路介面附件，請選取啟用核取方塊。
8. 啟用 ENA Express 時，您可以設定 UDP 設定。若要使用 ENA Express UDP，請選取啟用核取方塊。
9. 若要儲存設定，請選擇 Save (儲存)。

當您連接網路介面時設定 ENA Express

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中，選擇 Network Interfaces (網路介面)。
3. 選取未連接到執行個體的網路介面 (Status (狀態) 為 Available (可用)。您可以選擇 Network interface ID (網路介面 ID) 連結，以開啟詳細資訊頁面；或選取清單左側的核取方塊。

4. 選取您要連接的 Instance (執行個體)。
5. 若要在您將網路介面連接至執行個體後使用 ENA Express，請選取啟用核取方塊。
6. 啟用 ENA Express 時，您可以設定 UDP 設定。若要使用 ENA Express UDP，請選取啟用核取方塊。
7. 若要將網路介面連接至執行個體並儲存 ENA Express 設定，請選擇 Attach (連接)。

AWS CLI

當您連接網路介面時設定 ENA Express

使用 attach-network-interface 命令，如下列範例所示。

範例 1：將 ENA Express 用於 TCP 流量而非 UDP 流量

此範例將 EnaSrdEnabled 設定為 true，並允許 EnaSrdUdpEnabled 預設為 false。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

範例 2：將 ENA Express 用於 TCP 流量和 UDP 流量

在此範例中，將 EnaSrdEnabled 和 EnaSrdUdpEnabled 都設定為 true。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

更新網路介面附件的 ENA Express 設定

使用 modify-network-interface-attribute 命令，如下列範例所示。

範例 1：將 ENA Express 用於 TCP 流量而非 UDP 流量

此範例會將 EnaSrdEnabled 設定為 true，並且允許將 EnaSrdUdpEnabled 預設為 false，如果之前並未設定。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

範例 2：將 ENA Express 用於 TCP 流量和 UDP 流量

在此範例中，將 EnaSrdEnabled 和 EnaSrdUdpEnabled 都設定為 true。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

範例 3：停止將 ENA Express 用於 UDP 流量

此範例會將 EnaSrdUdpEnabled 設定為 false。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdUdpSpecification={EnaSrdUdpEnabled=false}'
```

PowerShell

當您連接網路介面時設定 ENA Express

使用 Add-EC2NetworkInterface cmdlet，如以下範例所示。

範例 1：將 ENA Express 用於 TCP 流量而非 UDP 流量

此範例將 EnaSrdEnabled 設定為 true，並允許 EnaSrdUdpEnabled 預設為 false。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true
```

範例 2：將 ENA Express 用於 TCP 流量和 UDP 流量

在此範例中，將 EnaSrdEnabled 和 EnaSrdUdpEnabled 都設定為 true。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdUdpSpecification_EnaSrdUdpEnabled $true
```

設定網路介面附件的 ENA Express 設定

使用 Edit-EC2NetworkInterfaceAttribute cmdlet，如以下範例所示。

範例 1：將 ENA Express 用於 TCP 流量而非 UDP 流量

此範例會將 EnaSrdEnabled 設定為 true，並且允許將 EnaSrdUdpEnabled 預設為 false，如果之前並未設定。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

範例 2：將 ENA Express 用於 TCP 流量和 UDP 流量

在此範例中，將 EnaSrdEnabled 和 EnaSrdUdpEnabled 都設定為 true。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
```

```
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

範例 3：停止將 ENA Express 用於 UDP 流量

此範例會將 EnaSrdUdpEnabled 設定為 false。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-0123f4567890a1b23 `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $false ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

在啟動時設定 ENA Express

啟動執行個體時，您可以使用下列其中一種方法來直接設定 ENA Express。指定的連結會參考這些方法 AWS 管理主控台 的說明。

- 啟動執行個體精靈 – 您可以使用啟動執行個體精靈在啟動時設定 ENA Express。如需詳細資訊，請參閱啟動執行個體精靈的 網路設定 中的「進階網路組態」。
- 啟動範本 – 當您使用啟動範本時，可以在啟動時設定 ENA Express。如需詳細資訊，請參閱 建立 Amazon EC2 啟動範本 頁面，然後展開「網路設定」區段，並檢閱「進階網路組態」。

已使用 Intel 82599 VF 介面來強化網路。

在 Xen 型執行個體方面，Intel 82599 虛擬功能 (VF) 介面會提供增強型聯網功能。介面會使用 Intel ixgbevf 驅動程式。

以下標籤會顯示如何驗證為執行個體作業系統安裝的網路轉接器驅動程式。

Linux

Linux 介面驅動程式

請用下列命令確認模組使用於特定介面，並將介面名稱換成您想要檢查的介面。如果您使用單一介面 (預設)，則這會是 eth0。如果作業系統支援可預測的網路名稱，則名稱可能像是 ens5。

在下列範例中，不會載入 ixgbevf 模組，因為列出的驅動程式為 vif。

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

在此範例中，將載入 ixgbevf 模組。此執行個體已正確設定增強型網路。

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

Windows

Windows 網路轉接器

若要確定已安裝驅動程式，請連線到執行個體，並開啟裝置管理員。您應會看到網路轉接器下列出的 Intel(R) 82599 Virtual Function。

目錄

- 準備您的執行個體以取得增強型聯網
- 測試是否已啟用增強型網路
- 在執行個體上啟用增強型網路

- 疑難排解連線問題

準備您的執行個體以取得增強型聯網

若要準備使用 Intel 82599 VF 介面的增強型聯網，請依下列方式設定您的執行個體：

- 驗證執行個體類型為以下其中一項：C3、C4、D2、I2、M4 (不含 `m4.16xlarge`) 和 R3。
- 確定該執行個體具有網際網路連線能力。
- 如果您的執行個體上有想要保留的重要資料，您現在應從執行個體建立 AMI，以備份這些資料。更新核心與核心模組，以及啟用 `sriovNetSupport` 屬性，可能使執行個體變得不相容或使作業系統變得無法連線。如果您有較新的備份，發生這些狀況時資料便能獲得保留。
- Linux 執行個體 – 從使用 Linux 核心版本 2.6.32 或更新版本的 HVM AMI 啟動執行個體。最新的 Amazon Linux HVM AMI 擁有安裝增強型聯網所需的模組和必要的屬性集。因此，如果您使用目前的 Amazon Linux HVM AMI 啟動 Amazon EBS 支援的增強型聯網所支援的執行個體，則您的執行個體已啟用增強型聯網。

  > **Warning**
  >
  > 增強型聯網僅支援於 HVM 執行個體。啟用 PV 執行個體的增強型聯網可能使其變為無法連線。在沒有正確的模組或模組版本下設定此屬性，也可能使執行個體變為無法連線。

- Windows 執行個體 – 從64位元 HVM AMI 啟動執行個體。您無法在 Windows Server 2008 上啟用增強型聯網。Windows Server 2012 R2 和 Windows Server 2016 和更新版本 AMI 已啟用增強型聯網。Windows Server 2012 R2 內含 Intel 驅動程式 1.0.15.3，因此我們建議您使用 Pnputil.exe 公用程式將驅動程式升級至最新版本。
- AWS CloudShell 從 使用 AWS 管理主控台，或在您選擇的AWS Tools for Windows PowerShell任何電腦上安裝和設定 AWS CLI或 ，最好是本機桌上型電腦或筆記型電腦。如需詳細資訊，請參閱 存取 Amazon EC2 和 AWS CloudShell 使用者指南。增強型聯網無法從 Amazon EC2 主控台管理。

測試是否已啟用增強型網路

確認已在執行個體或映像上設定 `sriovNetSupport` 屬性。

AWS CLI

若要檢查執行個體屬性 (sriovNetSupport)

使用下列 describe-instance-attribute 命令。如果已設定屬性，則值為 `simple`。

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute sriovNetSupport
```

若要檢查映像屬性 (sriovNetSupport)

使用下列 describe-images 命令。如果已設定屬性，則值為 simple。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query "Images[].SriovNetSupport"
```

PowerShell

若要檢查執行個體屬性 (sriovNetSupport)

使用 Get-EC2InstanceAttribute cmdlet。如果已設定屬性，則值為 simple。

```
Get-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Attribute sriovNetSupport
```

若要檢查映像屬性 (sriovNetSupport)

使用下列 describe-images 命令。如果已設定屬性，則值為 simple。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).SriovNetSupport
```

在執行個體上啟用增強型網路

您使用的程序會依執行個體的作業系統而定。

> **Warning**
>
> 增強型聯網屬性一經啟用後，便無法停用。

Amazon Linux

最新的 Amazon Linux HVM AMI 擁有安裝增強型網路所需的 `ixgbevf` 模組和必要的 `sriovNetSupport` 屬性集。因此，如果您使用目前的 Amazon Linux HVM AMI 啟動執行個體類型，則您的執行個體已啟用增強型聯網。如需詳細資訊，請參閱 測試是否已啟用增強型網路。

如果您使用較舊的 Amazon Linux AMI 啟動執行個體，且其尚未啟用增強型聯網，請用下列程序啟用增強型聯網。

啟用增強型聯網

1. 連線到您的執行個體。
2. 在執行個體上，執行下列命令，將您的執行個體更新為最新的核心與核心模組，包括 `ixgbevf`：

```
[ec2-user ~]$ sudo yum update
```

3. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來重新啟動執行個體：reboot-instances (AWS CLI) 或 Restart-EC2Instance (AWS Tools for Windows PowerShell)。
4. 再次連接至執行個體，並使用 `ixgbevf` 中的 modinfo ixgbevf 命令確認 測試是否已啟用增強型網路 模組已安裝且為最低建議版本。
5. [EBS 支援的執行個體] 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來停止執行個體：stop-instances (AWS CLI) 或 Stop-EC2Instance (AWS Tools for Windows PowerShell)。

   [執行個體存放區後端執行個體] 您無法停止執行個體來修改屬性。相反地，請跳到下一個程序。
6. 在本機電腦上使用下列其中一個命令啟用增強型聯網屬性。

   AWS CLI

   使用 modify-instance-attribute 命令，如下所示。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

   PowerShell

   使用 Edit-EC2InstanceAttribute cmdlet，如下所示。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -SriovNetSupport "simple"
```

7. (選擇性) 從執行個體建立 AMI，如 建立 Amazon EBS 支援的 AMI 所述。AMI 將從執行個體繼承增強型聯網屬性。因此，您預設可用此 AMI 啟動其他啟用增強型聯網的執行個體。
8. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。
9. 連接至執行個體，並使用 `ixgbevf` 中的 ethtool -i eth***n*** 命令確認 測試是否已啟用增強型網路 模組已在網路介面上安裝及載入。

啟用增強型聯網 (執行個體後端執行個體)

遵循先前的程序，一直進行到您停止執行個體的那個步驟。如 建立 Amazon S3 支援的 AMI 中所述建立新的 AMI，且務必在註冊 AMI 時啟用增強型聯網屬性。

AWS CLI

使用 register-image 命令，如下所示。

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

使用 Register-EC2Image，如下所示。

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu

開始之前，請先檢查您的執行個體是否啟用增強型聯網。

Quick Start Ubuntu HVM AMI 包含增強型聯網必要的驅動程式。如果您有 2.16.4 版以前的 `ixgbevf`，則可以安裝 `linux-aws` 核心套件來取得最新的增強型聯網驅動程式。

以下程序提供在 Ubuntu 執行個體上編譯 `ixgbevf` 模組的一般步驟。

安裝 `linux-aws` 核心套件

1. 連線到您的 執行個體。
2. 更新套件快取和套件。

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> 如果更新程序期間提示您安裝 grub，請用 /dev/xvda 安裝 grub，然後選擇保留目前版本的 /boot/grub/menu.lst。

其他 Linux 發行版本

開始之前，請先檢查您的執行個體是否啟用增強型聯網。最新的 Quick Start HVM AMI 包含增強型聯網必要的驅動程式，因此您不需要執行其他步驟。

如果您需要在 Amazon Linux 或 Ubuntu 以外的其他 Linux 發行版本上使用 Intel 82599 VF 介面啟用增強型聯網，以下程序提供一般步驟。如需像是詳細命令語法、檔案位置或套件和工具支援等詳細資訊，請參閱 Linux 發行版本的特定文件。

在 Linux 上啟用增強型聯網

1. 連線到您的 執行個體。
2. 在您的執行個體上從 Sourceforge (https://sourceforge.net/projects/e1000/files/ixgbevf%20stable/) 下載 ixgbevf 模組來源。

   2.16.4 之前版本的 ixgbevf，包括 2.14.2 版，無法在某些 Linux 發行版本上正確建置，包括特定版本的 Ubuntu。
3. 在您的執行個體上編譯及安裝 ixgbevf 模組。

   > **Warning**
   >
   > 如果您為目前核心編譯 ixgbevf 模組，接著升級核心，但未針對新核心重建驅動程式，系統可能在下次重新啟動時還原為發行版本專屬的 ixgbevf 模組。如果發行版本專屬版本與增強型聯網不相容，則系統可能變得無法連線。
4. 執行 sudo depmod 命令更新模組相依性。
5. 在執行個體上更新 initramfs，確定開機時載入新模組。
6. 判斷系統是否依預設使用可預測的網路介面名稱。使用 systemd 或 udev 197 或以上版本的系統可重新命名乙太網路裝置，但不保證單一網路介面會命名為 eth0。該行為會造成連線至執行個體時

發生問題。如需詳細資訊及查看其他的組態選項，請參閱 freedesktop.org 網站上的可預測的網路介面名稱。

a. 您可用下列命令在 RPM 系統上檢查 systemd 或 udev 的版本：

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
    systemd-208-11.el7_0.2.x86_64
```

在上述的 Red Hat Enterprise Linux 7 範例中，systemd 版本為 208，因此可預測的網路介面名稱已停用。

b. 若要停用可預測網路介面名稱，請將 `net.ifnames=0` 選項新增至 `GRUB_CMDLINE_LINUX` 的 `/etc/default/grub` 行中。

```
[ec2-user ~]$ sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /
etc/default/grub
```

c. 重建 grub 組態檔案。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS 支援的執行個體] 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來停止執行個體：stop-instances (AWS CLI) 或 Stop-EC2Instance (AWS Tools for Windows PowerShell)。

   [執行個體存放區後端執行個體] 您無法停止執行個體來修改屬性。相反地，請跳到下一個程序。

8. 在本機電腦上使用下列其中一個命令啟用增強型聯網屬性。

   AWS CLI

   使用 modify-instance-attribute 命令，如下所示。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 -\
    -sriov-net-support simple
```

   PowerShell

   使用 Edit-EC2InstanceAttribute cmdlet，如下所示。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -SriovNetSupport "simple"
```

9. (選擇性) 從執行個體建立 AMI，如 建立 Amazon EBS 支援的 AMI 所述。AMI 將從執行個體繼承增強型聯網屬性。因此，您預設可用此 AMI 啟動其他啟用增強型聯網的執行個體。

   如果您執行個體的作業系統包含 `/etc/udev/rules.d/70-persistent-net.rules` 檔案，您必須先刪除檔案，然後再建立 AMI。此檔案包含原始執行個體乙太網路卡的 MAC 地址。如果其他執行個體使用此檔案開機，作業系統將無法找到設備，且 `eth0` 可能會失敗，因而造成開機問題。此檔案會在下次開機週期時重新產生，且從 AMI 啟動的任何執行個體將會建立自己的檔案版本。

10. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。
11. (選擇性) 連線到執行個體，並確認模組已安裝。

啟用增強型聯網 (執行個體存放區支援的執行個體)

遵循先前的程序，一直進行到您停止執行個體的那個步驟。如 建立 Amazon S3 支援的 AMI 中所述建立新的 AMI，且務必在註冊 AMI 時啟用增強型聯網屬性。

AWS CLI

使用 register-image 命令，如下所示。

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

使用 Register-EC2Image，如下所示。

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Windows

如果您啟動執行個體，且其尚未啟用增強型聯網，您必須在執行個體上下載並安裝必要的網路轉接器驅動程式，然後設定 `sriovNetSupport` 執行個體屬性，以啟動增強型聯網。您只能在支援的執行個體類型上啟用此屬性。如需詳細資訊，請參閱增強 Amazon EC2 執行個體的聯網能力。

> **Important**
>
> 若要在 Windows AMIs 中檢視最新的驅動程式更新，請參閱「AWS Windows AMI 參考」中的 Windows AMI 版本歷史記錄。

啟用增強型聯網

1. 連線到您的執行個體，並以本機管理員登入。
2. [Windows Server 2016 和更新版本] 安裝驅動程式後，執行下列 EC2 Launch PowerShell 指令碼，以設定執行個體。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

> **Important**
>
> 當您啟用初始化執行個體 EC2 Launch 指令碼時，管理員密碼將重設。您可以修改組態檔來停用管理員密碼重設，方法為在初始化任務的設定中指定它。

3. 從執行個體中，下載適用於您的作業系統的 Intel 網路轉接器驅動程式：

   - Windows Server 2022

     請造訪下載頁面並下載Wired_driver_*version*_x64.zip。
   - Windows Server 2019 (包括 Server 版本 1809 和更新版本*)

     請造訪下載頁面並下載Wired_driver_*version*_x64.zip。
   - Windows Server 2016 (包括 Server 版本 1803 和更早版本*)

     請造訪下載頁面並下載Wired_driver_*version*_x64.zip。
   - Windows Server 2012 R2

     請造訪下載頁面並下載Wired_driver_*version*_x64.zip。
   - Windows Server 2012

     請造訪下載頁面並下載Wired_driver_*version*_x64.zip。
   - Windows Server 2008 R2

請造訪下載頁面並下載PROWinx64Legacy.exe。

* Intel 驅動程式與軟體頁面並未特別說明伺服器版本 1803 和更早版本，以及 1809 和更新版本。

4. 安裝適用於您的作業系統的 Intel 網路轉接器驅動程式。

   - Windows Server 2008 R2

     1. 在 Downloads (下載) 資料夾中，找出 PROWinx64Legacy.exe 檔案並將其重新命名為 PROWinx64Legacy.zip。
     2. 解壓縮 PROWinx64Legacy.zip 檔案的內容。
     3. 開啟命令列，導覽至解壓縮的資料夾，並執行下面的命令，使用 pnputil 公用程式新增及安裝驅動程式存放區中的 INF 檔案。

     ```
     C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
     ```

   - Windows Server 2022、Windows Server 2019、Windows Server 2016、Windows Server 2012 R2 和 Windows Server 2012

     1. 在 Downloads (下載) 資料夾中，解壓縮 Wired_driver_*version*_x64.zip 檔案的內容。
     2. 開啟命令列，導覽至解壓縮的資料夾，並執行下面其中一個命令，使用 pnputil 公用程式新增及安裝驅動程式存放區中的 INF 檔案。

        - Windows Server 2022

        ```
        pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
        ```

        - Windows Server 2019

        ```
        pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
        ```

        - Windows Server 2016

        ```
        pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
        ```

        - Windows Server 2012 R2

        ```
        pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
        ```

        - Windows Server 2012

```
pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
```

5. 在本機電腦上使用下列其中一個命令啟用增強型聯網屬性。

   AWS CLI

   使用 modify-instance-attribute 命令，如下所示。

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --sriov-net-support simple
   ```

   PowerShell

   使用 Edit-EC2InstanceAttribute cmdlet，如下所示。

   ```
   Edit-EC2InstanceAttribute `
       -InstanceId i-1234567890abcdef0 `
       -SriovNetSupport "simple"
   ```

6. (選擇性) 從執行個體建立 AMI，如 建立 Amazon EBS 支援的 AMI 所述。AMI 將從執行個體繼承增強型聯網屬性。因此，您預設可用此 AMI 啟動其他啟用增強型聯網的執行個體。
7. 從本機電腦中，使用 Amazon EC2 主控台或下列其中一個命令來啟動執行個體：start-instances (AWS CLI) 或 Start-EC2Instance (AWS Tools for Windows PowerShell)。

疑難排解連線問題

如果您在啟用增強型聯網時失去連線，則 `ixgbevf` 模組可能與核心不相容。請嘗試安裝您執行個體適用之 Linux 發行版本隨附的 `ixgbevf` 模組版本。

如果您為 PV 執行個體或 AMI 啟用增強聯網，可能導致無法連線到您的執行個體。

如需詳細資訊，請參閱如何開啟及設定 EC2 執行個體的增強型聯網？

監控 EC2 執行個體的 ENA 設定網路效能

彈性網絡介面 (ENA) 驅動程式會從啟用它們的執行個體發行網路效能指標。您可以使用這些指標來進行執行個體效能的故障診對、為工作負載選擇正確的執行個體大小、主動計劃擴展活動，以及基準化分析應用程式，以判斷它們是否能夠最大化執行個體上的可用效能。

Amazon EC2 在執行個體層級定義網路上限，以確保高品質的聯網體驗，包括跨執行個體大小的一致網路效能。 為每個執行個體 AWS 提供下列項目的上限：

- 頻寬能力 - 每個 EC2 執行個體都具有最大頻寬，用於彙總傳入和傳出流量 (根據執行個體類型和大小)。一些執行個體使用網路 I/O 額度機制，根據平均頻寬使用率來配置網路頻寬。Amazon EC2 也具有最大頻寬，可用於傳入 Direct Connect 和網際網路的流量。如需詳細資訊，請參閱Amazon EC2 執行個體網路頻寬。
- 每秒封包數 (PPS) 效能 - 根據執行個體類型和大小，每個 EC2 執行個體都具有 PPS 效能上限。
- 追蹤的連線 - 安全性群組會追蹤每個建立的連線，以確保傳回封包如預期般交付。每個執行個體可追蹤的連線數目上限。如需詳細資訊，請參閱Amazon EC2 安全群組連線追蹤
- 連結本機服務存取 - Amazon EC2 可為 Amazon DNS 服務、執行個體中繼資料服務和 Amazon Time Sync Service 等本機代理服務的流量提供每個網路介面的 PPS 上限。

當執行個體的網路流量超過上限時， 會將超過上限的流量 AWS 排入佇列，然後捨棄網路封包。您可以使用網路效能指標監控流量何時超過上限。這些指標可即時通知您網路流量的影響，以及可能的網路效能問題。

目錄

- 要求
- ENA 驅動程式的指標
- 檢視 執行個體的網路效能指標
- ENA Express 的指標
- 適用於 ENA 的網路效能度量 (具有 DPDK 驅動程式)
- 執行 FreeBSD 執行個體的指標

要求

Linux 執行個體

- 安裝 ENA 驅動程式 2.2.10 或更高版本。若要確認已安裝的版本，請使用 ethtool 命令。在下列範例中，版本符合最低需求。

```
[ec2-user ~]$ ethtool -i eth0 | grep version
version: 2.2.10
```

若要升級您的 ENA 驅動程式，請參閱增強型聯網。

- 若要將這些指標匯入至 Amazon CloudWatch，請安裝 CloudWatch 代理程式。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的收集網路效能指標。
- 若要支援 `conntrack_allowance_available` 指標，請安裝 ENA 驅動程式 2.8.1 版或更新版本。
- 若要覆寫輸出片段 PPS 限制 1024，請安裝 ENA 驅動程式 2.13.3 版或更新版本。

Windows 執行個體

- 安裝 ENA 驅動程式 2.2.2 版或更高版本。若要確認已安裝的版本，請使用如下所示的 Device Manager (裝置管理員)。
  1. 執行 `devmgmt.msc` 以開啟 Device Manager (裝置管理員)。
  2. 展開 Network Adapters (網路介面卡)。
  3. 選擇 Amazon Elastic Network Adapter (Amazon 彈性網絡介面卡)、Properties (屬性) 。
  4. 在 Driver (驅動程式)索引標籤上，找出 Driver Version (驅動程式版本)

  若要升級您的 ENA 驅動程式，請參閱增強型聯網。
- 若要將這些指標匯入至 Amazon CloudWatch，請安裝 CloudWatch 代理程式。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的 收集進階網路指標。

ENA 驅動程式的指標

ENA 驅動程式會即時將下列指標交付至執行個體。它們提供自上次重設驅動程式後，每個網路介面上排入佇列或丟棄的累積封包數目。

指標	說明	支援於
`bw_in_allowance_exceeded`	因傳入的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。	所有執行個體類型
`bw_out_allowance_exceeded`	因傳出的彙總頻寬超過執行個體的上限而排入佇列或丟棄的封包數目。	所有執行個體類型

指標	說明	支援於
conntrack_allowance_exceeded	因為連線追蹤超過執行個體的上限且無法建立新的連線，而丟棄的封包數目。這可能會導致傳送或傳回執行個體流量的封包遺失。	所有執行個體類型
conntrack_allowance_available	在達到該執行個體類型的追蹤連線限額之前，執行個體可建立的追蹤連線數目。	僅限 Nitro 型執行個體
linklocal_allowance_exceeded	由於本機代理伺服器服務的流量 PPS 超過網路介面上限而丟棄的封包數目。這會影響 Amazon DNS 服務、執行個體中繼資料服務和 Amazon Time Sync Service 的流量，但不會影響自訂 DNS 解析器的流量。	所有執行個體類型
pps_allowance_exceeded	因雙向 PPS 超過執行個體的上限而排入佇列或丟棄的封包數目。*	所有執行個體類型

* 根據 ENA Linux 驅動程式 2.13.3 版或更新版本的片段代理模式設定，此限制也可以包括網路介面的超過 1024 PPS 的輸出片段捨棄。如果 Linux 驅動程式已啟用片段代理模式，則輸出片段捨棄會繞過通常適用的 1024 PPS 限制，並在標準 PPS 允許範圍內計數。預設停用片段代理模式。

檢視 執行個體的網路效能指標

您使用的程序會依執行個體的作業系統而定。

Linux 執行個體

您可以將指標發佈至您最愛的工具，以視覺化方式呈現指標資料。例如，您可以使用 CloudWatch 代理程式將指標發佈至 Amazon CloudWatch。代理程式可讓您選取個別指標並控制發佈。

您也可以使用 ethtool 以擷取每個網路介面的指標，例如下方所示的 eth0。

```
[ec2-user ~]$ ethtool -S eth0
```

```
    bw_in_allowance_exceeded: 0
    bw_out_allowance_exceeded: 0
    pps_allowance_exceeded: 0
    conntrack_allowance_exceeded: 0
    linklocal_allowance_exceeded: 0
    conntrack_allowance_available: 136812
```

Windows 執行個體

您可以使用任何 Windows 效能計數器的取用者來檢視指標。資料可以根據EnaPerfCounters 資訊清單進行剖析。這是定義效能計數器提供者及其計數器集的 XML 檔案。

安裝資訊清單

如果您使用包含 ENA 驅動程式 2.2.2 或更高版本的 AMI 啟動執行個體，或使用 ENA 驅動程式 2.2.2 驅動程式套件中的安裝指令碼，則資訊清單已安裝。若要手動安裝資訊清單，請使用下列步驟：

1. 使用下列命令移除現有的資訊清單：

   ```
   unlodctr /m:EnaPerfCounters.man
   ```

2. 將資訊清單檔案 EnaPerfCounters.man 從驅動程式安裝套件複製到 %SystemRoot%\System32\drivers。
3. 使用下列命令安裝新的資訊清單：

   ```
   lodctr /m:EnaPerfCounters.man
   ```

使用效能監控以檢視指標

1. 開啟效能監控器。
2. 按 Ctrl+N 以新增計數器。
3. 從清單中選擇 ENA Packets Shaping (ENA 封包成形)。
4. 選取要監控的執行個體，然後選取 Add (新增)。
5. 選擇 OK (確定)。

ENA Express 的指標

ENA Express 採用可擴展可靠資料包 AWS (SRD) 技術。SRD 是一種高效能的網路傳輸通訊協定，其使用動態路由來增加輸送量並將網路流量尾延遲降至最低。若您已在傳送執行個體和接收執行個體上啟用網路介面附件的 ENA Express，您可以使用 ENA Express 指標來協助確保您的執行個體充分利用 SRD 技術提供的效能改進。例如：

- 評估您的資源，以確保其有足夠的容量來建立更多 SRD 連線。
- 識別存在防止合格傳出封包使用 SRD 的潛在問題的位置。
- 計算對執行個體使用 SRD 的傳出流量百分比。
- 計算對執行個體使用 SRD 的傳入流量百分比。

> **Note**
>
> 若要產生指標，請使用 2.8 或更新版本的驅動程式。

若要查看針對 ENA Express 篩選的 Linux 執行個體指標清單，請遵循適用於網路介面的 ethtool 命令 (在這裡顯示為 `eth0`)：請記下 `ena_srd_mode` 指標的值。

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
 ena_srd_mode: 1
 ena_srd_tx_pkts: 0
 ena_srd_eligible_tx_pkts: 0
 ena_srd_rx_pkts: 0
 ena_srd_resource_utilization: 0
```

下列指標適用於所有已啟用 ENA Express 的執行個體。

ena_srd_mode

描述已啟用哪些 ENA Express 功能。相關值如下：

- `0` = ENA Express 關閉，UDP 關閉
- 1 = ENA Express 開啟，UDP 關閉
- 2 = ENA Express 關閉，UDP 開啟

> **Note**
>
> 僅在原先啟用 ENA Express，且將 UDP 設定為使用它時，才會發生這種情況。會保留 UDP 流量的先前值。

- 3 = ENA Express 開啟，UDP 開啟

ena_srd_eligible_tx_pkts

網路數量如下：

- 支援傳送和接收執行個體類型。如需詳細資訊，請參閱 ENA Express 支援的執行個體類型 資料表。
- 傳送和接收執行個體都必須設定 ENA Express。
- 傳送和接收執行個體必須在相同的 區域中執行。
- 執行個體之間的網路路徑不得包含中介軟體方塊。ENA Express 目前不支援中介軟體。

> **Note**
>
> ENA Express 資格指標涵蓋來源和目的地需求，以及兩個端點之間的網路。在計算後，合格封包仍可能會被取消資格。例如，如果合格封包超過最大傳輸單位 (MTU) 限制，則其會回復為標準 ENA 傳輸，儘管封包在計數器中仍會反映為合格。

ena_srd_tx_pkts

在給定時間段內傳輸的 SRD 封包數目。

ena_srd_rx_pkts

在給定時間段內接收的 SRD 封包數目。

ena_srd_resource_utilization

執行個體使用的並行 SRD 連線允許的最大記憶體使用率百分比。

若要確認封包傳輸是否使用 SRD，您可以比較指定期間內的合格封包數目 (`ena_srd_eligible_tx_pkts` 指標) 與傳輸的 SRD 封包數目 (`ena_srd_tx_pkts` 指標)。

出口流量 (傳出封包)

若要確保您的出口流量如預期般使用 SRD，請比較 SRD 合格封包數目 (ena_srd_eligible_tx_pkts) 與指定期間內傳送的 SRD 封包數目 (ena_srd_tx_pkts)。

合格封包數目與傳送的 SRD 封包數目之間的顯著差異通常是資源使用率問題所造成的。當連接至執行個體的網路卡已用完其最大資源，或封包超過 MTU 限制時，合格封包無法透過 SRD 傳輸，且必須回到標準 ENA 傳輸。在即時遷移或即時伺服器更新期間，封包也可能會歸於此差距中。需要額外的疑難排解，才能判斷根本原因。

> **Note**
>
> 您可以忽略合格封包數目和 SRD 封包數目之間偶爾的細微差異。例如，當您的執行個體為 SRD 流量建立與另一個執行個體的連線時，就可能會發生這種情況。

若要了解指定期間內的總出口流量使用 SRD 的百分比，請比較傳出的 SRD 封包數目 (ena_srd_tx_pkts) 與該執行個體在該時間內傳送的封包總數 (NetworkPacketOut)。

輸入流量 (傳入封包)

若要了解傳入流量使用 SRD 的百分比，請比較指定期間內接收的 SRD 封包數目 (ena_srd_rx_pkts) 與該執行個體在該時間內接收的封包總數 (NetworkPacketIn)。

資源使用率

資源使用率是根據單一執行個體在指定時間可以保留的並行 SRD 連線數目而定。資源使用率指標 (ena_srd_resource_utilization) 會追蹤您目前執行個體的使用率。當使用率接近 100% 時，您預期將會看到效能問題。ENA Express 從 SRD 回復為標準 ENA 傳輸，且捨棄封包的可能性增加。高資源使用率表示是時候擴展執行個體，以改善網路效能。

> **Note**
>
> 當執行個體的網路流量超過上限時， 會將超過上限的流量 AWS 排入佇列，然後捨棄網路封包。

Persistence

在啟用執行個體的 ENA Express 時，會累積輸出和輸入指標。如果停用 ENA Express，則指標就會停止累積，但只要執行個體仍在執行，就會持續存在。如果執行個體重新啟動或終止，或者如果網路介面已從執行個體中斷連接，就會重設指標。

適用於 ENA 的網路效能度量 (具有 DPDK 驅動程式)

ENA 驅動程式 2.2.0 版及更高版本支援網路指標報告。DPDK 20.11 包含 ENA 驅動程式 2.2.0，是第一個支援此功能的 DPDK 版本。

DPDK 驅動程式 25.03 版或更新版本支援片段代理模式。如果 DPDK 驅動程式已啟用片段代理模式，則輸出片段捨棄會繞過通常適用的 1024 PPS 限制，並在標準 PPS 允許範圍內計數。預設停用片段代理模式。

您可以使用範例應用程式來檢視 DPDK 統計資料。若要啟動範例應用程式的互動式版本，請執行下列命令。

```
./app/dpdk-testpmd -- -i
```

在此互動式工作階段中，您可以輸入命令來擷取連接埠的延伸統計資料。下列範例命令會擷取連接埠 0 的統計資料。

```
show port xstats 0
```

以下是包含 DPDK 範例應用程式的互動式工作階段範例。

```
[root@ip-192.0.2.0 build]# ./app/dpdk-testpmd -- -i
        EAL: Detected 4 lcore(s)
        EAL: Detected 1 NUMA nodes
        EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
        EAL: Selected IOVA mode 'PA'
        EAL: Probing VFIO support...
        EAL:   Invalid NUMA socket, default to 0
        EAL:   Invalid NUMA socket, default to 0
        EAL: Probe PCI driver: net_ena (1d0f:ec20) device: 0000:00:06.0
(socket 0)
        EAL: No legacy callbacks, legacy socket not created
        Interactive-mode selected

        Port 0: link state change event
        testpmd: create a new mbuf pool <mb_pool_0>: n=171456,
size=2176, socket=0
        testpmd: preferred mempool ops selected: ring_mp_mc

        Warning! port-topology=paired and odd forward ports number, the
last port will pair with itself.
```

```
        Configuring Port 0 (socket 0)
        Port 0: 02:C7:17:A2:60:B1
        Checking link statuses...
        Done
        Error during enabling promiscuous mode for port 0: Operation
not supported - ignore
        testpmd> show port xstats 0
        ###### NIC extended statistics for port 0
        rx_good_packets: 0
        tx_good_packets: 0
        rx_good_bytes: 0
        tx_good_bytes: 0
        rx_missed_errors: 0
        rx_errors: 0
        tx_errors: 0
        rx_mbuf_allocation_errors: 0
        rx_q0_packets: 0
        rx_q0_bytes: 0
        rx_q0_errors: 0
        tx_q0_packets: 0
        tx_q0_bytes: 0
        wd_expired: 0
        dev_start: 1
        dev_stop: 0
        tx_drops: 0
        bw_in_allowance_exceeded: 0
        bw_out_allowance_exceeded: 0
        pps_allowance_exceeded: 0
        conntrack_allowance_exceeded: 0
        linklocal_allowance_exceeded: 0
        rx_q0_cnt: 0
        rx_q0_bytes: 0
        rx_q0_refill_partial: 0
        rx_q0_bad_csum: 0
        rx_q0_mbuf_alloc_fail: 0
        rx_q0_bad_desc_num: 0
        rx_q0_bad_req_id: 0
        tx_q0_cnt: 0
        tx_q0_bytes: 0
        tx_q0_prepare_ctx_err: 0
        tx_q0_linearize: 0
        tx_q0_linearize_failed: 0
        tx_q0_tx_poll: 0
        tx_q0_doorbells: 0
```

```
        tx_q0_bad_req_id: 0
        tx_q0_available_desc: 1023
        testpmd>
```

如需範例應用程式及使用它來擷取延伸統計資料的詳細資訊，請參閱 DPDK 文件中的 Testpmd 應用程式使用者指南。

執行 FreeBSD 執行個體的指標

從 2.3.0 版開始，ENA FreeBSD 驅動程式支援收集執行 FreeBSD 時，執行個體的網路效能指標。若要啟用收集 FreeBSD 指標，請輸入下列命令，並將*##*設定為介於 1 到 3600 之間的值。這會指定收集 FreeBSD 指標的頻率 (以秒為單位)。

```
sysctl dev.ena.network_interface.eni_metrics.sample_interval=interval
```

例如，下列命令會將驅動程式設定為每 10 秒在網路介面 1 上收集一次 FreeBSD 指標：

```
sysctl dev.ena.1.eni_metrics.sample_interval=10
```

若要關閉收集 FreeBSD 指標，您可以執行上述命令並指定 `0` 作為*##*。

啟用收集 FreeBSD 指標後，您可以執行下列命令來擷取最新收集的指標集。

```
sysctl dev.ena.network_interface.eni_metrics
```

改善基於 Linux 的 EC2 執行個體的網路延遲

網路延遲是指資料封包從其來源傳送到目的地所需的時間。透過網路傳送資料的應用程式仰賴及時回應為使用者提供正面體驗。高網路延遲可能會導致各種問題，例如：

- 網頁載入時間緩慢
- 影片串流延遲
- 難以存取線上資源

本節針對在 Linux 上執行的 Amazon EC2 執行個體，列出可改善網路延遲的步驟。若要達到最佳延遲，請依照下列步驟設定執行個體、核心和 ENA 驅動程式設定。如需其他設定指導方針，請參閱 GitHub 上「ENA Linux 驅動程式最佳實務與效能最佳化指南」。

> **Note**
>
> 視網路硬體、啟動執行個體的 AMI 以及應用程式使用案例而定，步驟和設定可能會略有不同。變更之前，請徹底測試並監控網路效能，確保可獲得理想結果。

減少資料封包的網絡跳轉次數

資料封包在從路由器之間移動時所經過的每個躍點都會增加網路延遲。通常，流量必須經過多個躍點才能到達目的地。有兩種方法可以減少 Amazon EC2 執行個體經過的網路躍點：

- 集群放置群組 – When 若指定集群放置群組，Amazon EC2 會啟動彼此非常靠近的執行個體，其實際位置就在同一個可用區域 (AZ) 內，封裝更緊密。群組中執行個體的實體接近性可讓其充分利用高速連線，進而實現低延遲和高單流輸送量。
- 專用執行個體–專用執行個體是專供您使用的實體伺服器。使用專用執行個體，可以啟動執行個體在相同實體伺服器上執行。在相同專用執行個體上執行的執行個體之間的通訊可以不需要額外的躍點。

Linux 核心組態如何影響延遲

Linux 核心組態可以增加或減少網路延遲。若要實現延遲最佳化目標，請務必根據工作負載的特定需求對 Linux 核心組態進行微調。

Linux 核心的許多組態選項可能有助於降低網路延遲。最具影響力的選項如下所示。

- 啟用忙碌輪詢模式 — 忙碌輪詢模式可減少網路接收路徑的延遲。若啟用忙碌輪詢模式，通訊端層代碼可以直接輪詢網路裝置的接收佇列。忙碌輪詢的缺點是主機中的 CPU 使用率較高，因為要在緊密迴圈中輪詢新資料。有兩個全域設定可控制等待所有介面封包的微秒數。

  `busy_read`

  通訊端讀取的低延遲忙碌輪詢逾時。這可控制等待通訊端層讀取裝置佇列上封包的微秒數。若要使用 sysctl 命令全域啟用此功能，Linux 核心組織建議使用 50 微秒的值。如需詳細資訊，請參閱 Linux 核心使用者和管理員指南中的 busy_read。

  ```
  [ec2-user ~]$ sudo sysctl -w net.core.busy_read=50
  ```

busy_poll

輪詢和選取的低延遲忙碌輪詢逾時。這可控制等待事件的微秒數。建議值介於 50-100 微秒之間，具體取決於要輪詢的通訊端數目。新增的通訊端越多，數值越高。

```
[ec2-user ~]$ sudo sysctl -w net.core.busy_poll=50
```

- 設定 CPU 電源狀態 (C-state) – C-state 可控制核心未使用時要進入的休眠層級。建議您控制 C-state 來微調系統的延遲與效能。在更深層的 C-state 中，CPU 基本上處於「休眠」狀態，在喚醒並轉換回作用中狀態之前無法回應請求。核心進入休眠狀態需要時間，而且雖然休眠核心讓其他核心更有餘裕加速至更高頻率，但是欲喚醒該休眠核心並開始執行工作也需要時間。

  例如，若負責處理網路封包中斷的核心處於休眠狀態，則處理此類中斷的服務就可能受到延遲。您可以設定系統，使其不會使用更深層的 C-state。不過，雖然此組態可能會減少處理器反應延遲，但同時也減少了 Turbo Boost 的其他核心可用預留空間。

  若要減少處理器反應延遲，您可以限制更深層的 C-state。如需詳細資訊，請參閱「Amazon Linux 2 使用者指南」中的限制深層的 C-state 達到高效能與低延遲。

插斷仲裁

ENA 網路驅動程式可啟用執行個體與網路之間的通訊。驅動程式會處理網路封包，並將它們傳遞至網路堆疊或 Nitro 卡。網路封包傳入時，Nitro 卡會產生中斷，讓 CPU 通知軟體有事件發生。

插斷

中斷是裝置或應用程式傳送給處理器的訊號。中斷會通知處理器已發生事件或已滿足需要立即注意的條件。中斷可以處理時間敏感任務，例如從網絡介面接收資料、處理硬體事件或服務來自其他裝置的請求。

插斷仲裁

中斷調節是一種技術，可以透過彙總或延遲來減少裝置產生的中斷次數。中斷調節的目的是透過減少處理大量中斷所產生的額外負荷來改善系統效能。中斷次數過多會增加 CPU 使用率，對輸送量造成不利影響，而中斷次數太少會增加延遲。

動態中斷調節

動態中斷調節是一種增強型的中斷調節形式，可根據目前的系統負載和流量模式動態調整中斷速率。其目標是在減少中斷負荷和每秒封包數或頻寬之間取得平衡。

> **Note**
>
> 在某些 AMI 中預設啟用動態中斷調節 (但在所有 AMI 中都可啟用或停用)。

為了將網路延遲降到最低，可能需要停用中斷調節。但是這也可能會增加插斷處理的開銷。在減少延遲和盡可能減少開銷之間找到適當的平衡點非常重要。ethtool 命令可協助您設定插斷仲裁。依預設，rx-usecs 設定為 20，tx-usecs 設定為 64。

若要取得目前的插斷仲裁設定，請使用下列命令。

```
[ec2-user ~]$ ethtool -c interface | egrep "rx-usecs:|tx-usecs:|Adaptive RX"
Adaptive RX: on  TX: off
rx-usecs: 20
tx-usecs: 64
```

若要停用中斷修改和動態中斷調節，請使用下列命令。

```
[ec2-user ~]$ sudo ethtool -C interface adaptive-rx off rx-usecs 0 tx-usecs 0
```

Nitro 系統效能調校的考量因素

Nitro System 是由 建置的硬體和軟體元件集合 AWS ，可實現高效能、高可用性和高安全性。Nitro 系統提供的類裸機功能可免除虛擬化開銷，並支援需完整存取主機硬體的工作負載。如需更多詳細資訊，請參閱 AWS Nitro 系統。

所有目前世代的 EC2 執行個體類型都會對 EC2 Nitro 卡進行網路封包處理。本主題涵蓋 Nitro 卡上的高階封包處理、影響封包處理效能的網路架構和組態的常見層面，以及達到 Nitro 型執行個體峰值效能所需採取的動作。

Nitro 卡會處理所有輸入和輸出 (I/O) 介面，例如虛擬私有雲端 (VPCs) 所需的介面。對於透過網路傳送或接收資訊的所有元件，Nitro 卡可做為 I/O 流量的獨立運算裝置，與客戶工作負載所執行的系統主機板在物理上是分開的。

Nitro 卡上的網路封包流程

在 Nitro 系統上建置的 EC2 執行個體具有硬體加速功能，可更快速地進行封包處理，其輸送量是以每秒封包數 (PPS) 的指標進行測量。當 Nitro 卡執行新流程的初始評估時，該卡會儲存流程中全部封包所

具有的相同資訊，例如安全群組、存取控制清單和路由表項目。處理相同流程的其他封包時，可以使用儲存的資訊來減少這些封包的負荷。

您的連線速率是以每秒連線數 (CPS) 的指標進行測量。每個新連線都將產生額外的處理負荷，這些額外處理開銷必須納入工作負載能力預估。設計工作負載時，同時考慮 CPS 和 PPS 指標是至關重要的。

如何建立連線

當 Nitro 型執行個體與另一個端點之間建立連線時，Nitro 卡會評估兩個端點之間傳送或接收的第一個封包的完整流程。相同流程的後續封包通常不需要完整重新評估。不過，此規則也有例外狀況。如需例外狀況的詳細資訊，請參閱 不使用硬體加速的封包。

下列屬性將定義兩個端點和兩端之間的封包流程。這五個屬性合稱為 5 元組流程。

- 來源 IP
- 來源連接埠
- 目標 IP
- 目標連接埠
- 通訊協定

封包流程的方向稱為「傳入」和「傳出」。以下是端對端網路封包流程的高階概述。

- 「傳入」 – 當 Nitro 卡處理傳入網路封包時，會根據具狀態防火牆規則和存取控制清單來評估封包。該卡會追蹤連線、進行計量，並執行其他適用的動作。然後，該卡會將封包轉送到主機 CPU 上的目的地。
- 「輸出」 – 當 Nitro 卡處理傳出網路封包時，會查詢遠端介面目的地、評估各種 VPC 函數、套用速率限制，以及執行其他適用的動作。然後，該卡會將封包轉送到網路中的下一個中轉目的地。

透過網路設計達到最佳效能

若要發揮 Nitro 系統的效能，您必須了解網路處理的需求，以及這些需求如何影響 Nitro 資源的工作負載。然後，您就可以透過設計，讓網路環境達到最佳效能。您的基礎架構設定、應用程式工作負載設計及其組態，都可能會影響封包處理和連線速率。例如，如果您的應用程式有較高的連線建立率，例如 DNS 服務、防火牆或虛擬路由器，其利用連線建立後才能加速硬體的機會就較低。

您透過應用程式和基礎架構設定，來簡化工作負載並改善網路效能。不過，並非所有封包都符合加速資格。Nitro 系統會針對新連線以及不符合加速資格的封包使用完整的網路流程。

本節的其餘部分將著重於應用程式和基礎設施設計考量因素，以協助確保封包盡可能在加速路徑內流動。

Nitro 系統的網路設計考量因素

為執行個體設定網路流量時，需要考量的層面非常廣，這些層面都可能對 PPS 效能造成影響。建立流程後，大多數定期傳入或傳出的封包都符合加速資格。不過，此規則也有例外狀況，這些例外的目的是確保基礎架構設計和封包流程持續符合通訊協定標準。

若要獲得 Nitro 卡的最佳效能，您應該仔細考量基礎架構和應用程式的下列組態，及其詳細資訊與優缺點。

基礎架構考量因素

您的基礎架構組態可能會影響封包流程和處理效率。要考慮的重要因素包含下列項目：

非對稱的網路介面組態

安全群組使用連線追蹤，以追蹤流入和流出執行個體流量的資訊。非對稱路由，也就是當流量透過一個網路介面進入執行個體，並透過不同的網路介面離開，可能導致執行個體在追蹤流程時，可達的峰值效能降低。如需安全群組連線追蹤、未追蹤的連線和自動追蹤連線的詳細資訊，請參閱 Amazon EC2 安全群組連線追蹤。

網路驅動程式

網路驅動程式會定期更新和釋出。過時的驅動程式可能會大幅降低效能。請將驅動程式保持於最新狀態，以確保您獲得最新的修補程式，並充分發揮效能改善功能，例如，加速路徑功能僅適用於最新一代的驅動程式。舊版驅動程式不支援加速路徑功能。

若要利用加速路徑功能，建議您在執行個體上安裝最新的 ENA 驅動程式。

Linux 執行個體 – ENA Linux 驅動程式 2.2.9 或更新版本。若要從 Amazon Drivers GitHub 儲存庫安裝或更新 ENA Linux 驅動程式，請參閱讀我檔案的驅動程式編譯一節。

Windows 執行個體 – ENA Windows 驅動程式 2.0.0 或更新版本。若要安裝或更新 ENA Windows 驅動程式，請參閱 在 EC2 Windows 執行個體上安裝 ENA 驅動程式。

端點之間的距離

由於應用程式層的 TCP 時段調整，相比於跨區域連線，在同一可用區域中的兩個執行個體之間連線，每秒可以處理更多的封包，這決定了在任意給定時間可傳輸的資料量。執行個體之間若距離較長，則延遲將增加，端點可處理的封包數量也將隨之減少。

位元組佇列限制 (BQL)

BQL 是一項功能，可限制傳遞至 Nitro 卡的位元組數，以減少佇列。在 ENA 驅動程式、Amazon Linux 作業系統及大多數 Linux 發行版本中，BQL 預設為停用狀態。若同時啟用 BQL 與片段代理覆寫，在處理所有片段之前限制傳遞給 Nitro 的位元組數，可能會導致效能限制。

應用程式設計考量因素

應用程式設計和組態有幾個層面，可能會對您的處理效率造成影響。要考慮的重要因素包含下列項目：

封包大小

若封包大小較大，則執行個體可在網路上傳送和接收的資料輸送量會增加。Amazon EC2 支援 9001 位元組的巨型訊框，但其他服務可能會強制執行不同限制。若封包較小，則可提高封包處理速率，但當封包數量超過 PPS 額度時，可能造成可達到的最大頻寬降低。

如果封包的大小超過網絡跳轉的最大傳輸單位 (MTU)，路徑上的路由器可能會將其分段。產生的封包片段會被視為例外狀況，並以標準速率正常處理 (非加速)。這可能會導致效能發生變化。但您可透過片段代理模式設定，覆寫傳出分段封包的標準行為。如需詳細資訊，請參閱最大化 Nitro 系統的網路效能。建議您在設定 MTU 時評估您的拓撲。

通訊協定權衡

TCP 等可靠的通訊協定比 UDP 等不可靠的通訊協定具有更高的額外負荷。UDP 傳輸通訊協定具有較低額外負荷，以及簡化的網路處理，可能會產生更高的 PPS 速率，但封包交付的可靠性將遭到犧牲。如果封包交付的可靠性對您的應用程式來說並非關鍵，則 UDP 可能是不錯的選擇。

微爆量

當流量短時間內超過限額，而不是平均分佈時，就會發生微爆量。這通常會以微秒為單位進行。

例如，假設您的執行個體最多可以傳送 10Gbps，而您的應用程式卻在半秒內傳送整整 10Gb。此微爆量在前半秒便已超過額度，導致剩餘的秒內沒有任何剩餘額度。即使您在 1 秒內傳送 10Gb，前半秒的額度可能會導致封包排入佇列或遭到捨棄。

您可以使用 Linux 流量控制等網路排程器來協助調整輸送量，並避免因微量爆量而造成佇列或捨棄封包。

流程數量

單一流程限制為 5Gbps，除非其位於支援高達 10Gbps 的叢集置放群組內，或者使用支援高達 25Gbps 的 ENA Express。

同樣地，Nitro 卡可以處理多個流程的更多封包，而非使用單一流程。為了達到每個執行個體的尖峰封包處理速率，我們建議在總頻寬為 100Gbps 或更高的執行個體上至少 100 個流程。隨著彙總頻寬的增加，達到峰值處理速率所需的流程數量也會增加。基準測試將協助您判斷在網路上達到尖峰速率所需的組態。

彈性網路介面卡 (ENA) 佇列

ENA (彈性網路介面卡) 使用多個接收 (Rx) 與傳輸 (Tx) 佇列 (ENA 佇列)，以提升 EC2 執行個體的網路效能與可擴展性。這些佇列會透過在可用佇列間對傳送與接收的資料進行負載平衡，有效管理網路流量。

如需詳細資訊，請參閱the section called “ENA 佇列”。

特徵處理負荷

流量鏡射和 ENA Express 等功能可能導致處理負荷增加，進而降低絕對封包處理效能。您可以限制功能使用，或停用功能，以提高封包處理速率。

保持狀態的連線追蹤

您的安全群組使用連線追蹤來儲存流入和流出執行個體流量的資訊。連線追蹤會將規則套用至每個個別的網路流量流程，以判斷流量被允許還是被拒絕。Nitro 卡會使用流程追蹤來維持流程的狀態。套用的安全群組規則更多，評估流程便需要更多工作。

> **Note**
>
> 並非所有的網路流量流程都會追蹤。如果使用 未追蹤的連線 設定安全群組規則，則在有多個有效的回覆路徑時，除了自動追蹤連線以確保對稱路由之外，不需要額外的工作。

不使用硬體加速的封包

並非所有封包都可以利用硬體加速。處理這些例外狀況時，通常需要一些處理負荷，以確保網路流程的運作狀態。網路流程必須可靠地符合通訊協定標準、符合 VPC 設計中的變更，以及僅將封包路由至允許的目的地。不過，額外負荷將導致您的效能降低。

「封包片段」

如「應用程式設計考量因素」所述，超過網路 MTU 的封包所產生的封包片段，將被視為例外正常處理，無法利用硬體加速。但根據您的驅動程式版本，您可透過片段代理模式繞過輸出片段限制。如需詳細資訊，請參閱下方 最大化 Nitro 系統的網路效能 區段中可採取的動作。

「閒置連線」

當連線有一段時間沒有活動時，即使連線尚未達到逾時限制，系統也可以取消其優先順序。然後，如果資料在取消優先順序之後傳入，系統需要將其視為例外處理才能重新連線。

若要管理您的連線，您可以使用連線追蹤逾時來關閉閒置連線。您也可以使用 TCP 保持連線，讓閒置連線保持開啟。如需詳細資訊，請參閱閒置連線追蹤逾時。

VPC 變動

安全群組、路由表和存取控制清單的更新，皆需要在處理路徑中重新評估，以確保路由項目和安全群組規則的套用仍符合預期。

ICMP 流程

網路控制訊息通訊協定 (ICMP) 是一個網路層通訊協定，可供網路裝置診斷網路通訊問題。這些封包一律使用完整流程。

非對稱 L2 流程

在 NitroV3 及更早版本的平台上，若同一子網路中的兩個 ENI 中，一個使用預設閘道路由器而另一個未使用，則這兩個 ENI 之間的流量不會使用硬體加速。在此案例中，NitroV4 和更新版本平台會使用硬體加速。若要在 NitroV3 或更早版本的平台上獲得更佳效能，請確保兩個 ENI 使用的預設閘道路由器一致，或將這些 ENI 置於不同子網路中。

最大化 Nitro 系統的網路效能

您可透過調整網路設定，最大化 Nitro 系統的網路效能。

主題

- 考量事項
- 調校 PPS 效能
- 設定 ENA 佇列配置
- 監控 Linux 執行個體的效能

考量事項

在您做出任何設計決策，或調整執行個體上的任何網路設定之前，建議您採取下列步驟，以協助確保您獲得最佳結果：

1. 了解您可以藉由檢閱 Nitro 系統的網路設計考量因素 來改善效能之動作的優缺點。

   如需 Linux 執行個體組態的更多考量和最佳實務，請參閱 GitHub 上的 ENA Linux 驅動程式最佳實務和效能最佳化指南。
2. 以峰值活動流量計數作為工作負載的基準，以確定應用程式效能的基準。透過效能基準，您可以在設定或應用程式設計中測試變化，以了解哪些考量因素將產生最大影響，特別是如果您計劃向上擴展或橫向擴充。

調校 PPS 效能

下列清單包含您可以根據系統需求調整 PPS 效能的動作。

- 減少兩個執行個體之間的實體距離。若傳送和接收執行個體位於相同的可用區域，或使用叢集置放群組時，您可以減少封包從一個端點移動到另一個端點所需的跳轉次數。
- 請使用 未追蹤的連線。
- 針對網路流量使用 UDP 通訊協定。
- 對於彙總頻寬為 100Gbps 或更高的 EC2 執行個體，請將工作負載分散到 100 個以上的個別流程，以將工作平均分散到 Nitro 卡。
- 若要突破 EC2 執行個體的輸出片段 PPS 限制，可啟用片段代理模式 (取決於您的驅動程式版本)。此設定允許在處理路徑中評估分段封包，進而突破 1024 的輸出 PPS 限制。載入驅動程式時，執行下列其中一項命令來啟用或停用片段代理模式：

  啟用片段代理模式

  ```
  sudo insmod ena.ko enable_frag_bypass=1
  ```

  停用片段代理模式

  ```
  sudo insmod ena.ko enable_frag_bypass=0
  ```

設定 ENA 佇列配置

在支援的執行個體類型上，您可在彈性網路介面 (ENI) 之間動態配置這些佇列。彈性的 ENA 佇列配置可最佳化資源分佈，實現 vCPU 使用率最大化。通常情況下，高網路效能工作負載需要多個 ENA 佇列。如需詳細資訊，請參閱ENA 佇列。

監控 Linux 執行個體的效能

您可以在 Linux 執行個體上使用 Ethtool 指標，以監控執行個體聯網效能指標，例如頻寬、封包速率和連線追蹤。如需詳細資訊，請參閱 監控 EC2 執行個體的 ENA 設定網路效能。

在 EC2 Windows 執行個體上最佳化網路效能

為了在您的 Windows 執行個體上達到使用增強型聯網的最大網路效能，您可能需要修改預設的作業系統組態。建議您針對需要高網路效能的應用程式進行下列組態變更。其他最佳化 (例如開啟檢查總和卸載與啟用 RSS) 已經在官方 Windows AMI 上配置。

> **Note**
>
> 大多數使用案例都應該停用 TCP 煙囪卸載；此功能在 Windows Server 2016 中已被移除。

除了這些作業系統最佳化以外，您也應該考慮網路流量的最大傳輸單位 (MTU)，並根據您的工作負載和網路架構來調整。如需詳細資訊，請參閱EC2 執行個體的網路最大傳輸單位 (MTU)。

AWS 在 50us 叢集置放群組中啟動的執行個體與 99.9 百分位數的尾端延遲之間，定期測量平均往返延遲。如果您的應用程式需要始終很低的延遲，我們建議在 Nitro 系統中建構的固定效能執行個體上使用最新版本的 ENA 驅動程式。

設定接收端擴展 CPU 親和性

接收端調整 (RSS) 可用來將網路流量 CPU 負載分配至多個處理器。根據預設，Amazon 官方 Windows AMI 會設為啟用 RSS。ENA 彈性網路介面最多可提供八個 RSS 佇列。藉由為 RSS 佇列以及其他系統處理程序定義 CPU 親和性，您可以將 CPU 負載分配至多核心系統，以便處理更多網路流量。在具有超過 16 個 vCPUs執行個體類型上，我們建議您使用 `Set-NetAdapterRSS` PowerShell cmdlet，這會從所有彈性網路介面的 RSS 組態中手動排除開機處理器 （啟用超執行緒時的邏輯處理器 0 和 1)，以防止與各種系統元件發生爭用。

Windows 會察覺超執行緒，並確保單一網路介面卡 (NIC) 的 RSS 佇列總是置於不同的實體核心。因此，除非停用超執行緒，否則為了完全防止與其他 NICs 爭用，請將每個 NIC 的 RSS 組態分散在 16 個邏輯處理器範圍內。`Set-NetAdapterRss` cmdlet 可讓您定義 BaseProcessorGroup、BaseProcessorNumber、MaxProcessingGroup、MaxProcessorNumber 和 NumaNode (選用) 的值，以定義有效邏輯處理器的各 NIC 範圍。如果無足夠的實體核心可以完全消除彈性網路介面間的競爭，視見面預期的工作負載而定，請盡量減少重疊範圍，或減少 ENI 範圍中的邏輯處理器數量 (換言之，少量管理網路介面可能不需要許多已指派的 RSS 佇列)。此外，如先前所述，

各種元件必須在 CPU 0 上執行，因此當可使用足夠的 vCPU 時，我們建議從所有 RSS 組態中加以排除。

例如，當有 2 個 NUMA 節點並啟用超執行緒的 72 vCPU 執行個體出現三個彈性網路介面時，下列命令會將網路負載分配在兩顆 CPU 之間，而不會重疊並避免完全使用核心 0。

```
Set-NetAdapterRss -Name NIC1 -BaseProcessorGroup 0 -BaseProcessorNumber 2 -
MaxProcessorNumber 16
Set-NetAdapterRss -Name NIC2 -BaseProcessorGroup 1 -BaseProcessorNumber 0 -
MaxProcessorNumber 14
Set-NetAdapterRss -Name NIC3 -BaseProcessorGroup 1 -BaseProcessorNumber 16 -
MaxProcessorNumber 30
```

注意，這些設定對於每個網路轉接器都是持續有效。如果有個執行個體調整大小成不同 vCPU 數的執行個體，您應針對每個已啟用彈性網路介面的執行個體重新評估 RSS 組態。此 cmdlet 的完整 Microsoft 文件可參閱：Set-NetAdapterRss。

SQL 工作負載的特別注意事項：我們還建議您檢查輸入/輸出執行緒親和性設定和彈性網路介面 RSS 組態，以盡量減少相同 CPU 的輸入/輸出和網路爭用。請參閱伺服器組態：親和性遮罩。

適用於 Amazon EC2 上的 AI/ML 和 HPC 工作負載的 Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) 是可連接到 Amazon EC2 執行個體的網路裝置，用以加速人工智慧 (AI)、機器學習 (ML) 和高效能運算 (HPC) 應用程式。EFA 可讓您實現內部部署的 AI/ML 或 HPC 叢集的應用程式效能，並具有 AWS 雲端提供的可擴展性、靈活性和彈性。

EFA 比雲端式 HPC 系統中傳統上使用的 TCP 傳輸，提供更低和更一致的延遲及更高的傳輸量。它可增強執行個體間通訊的效能，在擴展 HPC 和機器學習應用程式時不可或缺。它經過最佳化，可在現有的 AWS 網路基礎設施上運作，並且可以根據應用程式需求進行擴展。

EFA 與 Libfabric 整合，並支援適用於 AI 和 ML 應用程式的 Nvidia Collective Communications Library (NCCL) 和 NVIDIA Inference Xfer Library (NIXL)，以及適用於 HPC 應用程式的 Open MPI 4.1 和更新版本，以及 Intel MPI 2019 Update 5 和更新版本。NCCL 和 MPI 與 Libfabric 1.7.0 及更新版本整合。NIXL 與 Libfabric 1.21.0 及更新版本整合。

在擁有 Nitro 第 4 版及更新版本的大多數受支援執行個體類型上，EFA 支援 RDMA (遠端直接記憶體存取) 寫入。擁有 Nitro 第 4 版及更新版本的所有執行個體都支援 RDMA 讀取。如需詳細資訊，請參閱支援的執行個體類型。

目錄

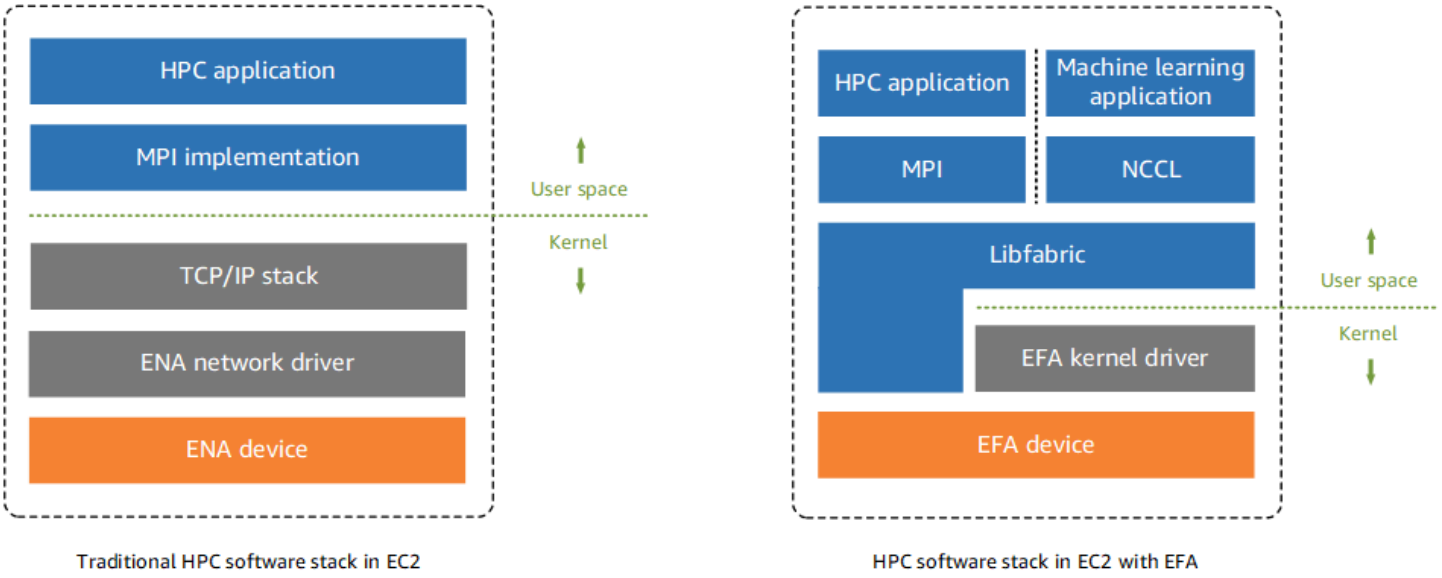
- EFA 基本概念
- 支援的介面和程式庫
- 支援的執行個體類型
- 支援的作業系統
- EFA 限制
- EFA 定價
- Amazon EC2 上 HPC 工作負載的 EFA 和 MPI 入門
- Amazon EC2 上的適用於 ML 工作負載的 EFA 和 NCCL 入門
- 開始使用適用於 Amazon EC2 上推論工作負載的 EFA 和 NIXL
- 使用多個網路卡最大化 Amazon EC2 執行個體的網路頻寬
- 在容器中安裝 EFA 軟體
- 建立 Elastic Fabric Adapter 並將其連接至 Amazon EC2 執行個體
- 從 Amazon EC2 執行個體分離和刪除 EFA
- 在 Amazon EC2 上監控彈性布料轉接器
- 使用檢查總和驗證 EFA 安裝程式
- Elastic Fabric Adapter 版本備註

EFA 基本概念

EFA 裝置可以透過兩種方式連接到 EC2 執行個體：

1. 使用傳統的 EFA 介面，也稱為 EFA 搭配 ENA，這會同時建立 EFA 裝置和 ENA 裝置。
2. 使用僅限 EFA 介面，僅建立 EFA 裝置。

EFA 裝置透過可擴展可靠資料包 (SRD) 通訊協定提供內建 OS-bypass 和擁塞控制等功能。EFA 裝置功能可實現低延遲、可靠的傳輸功能，讓 EFA 介面為 Amazon EC2 上的 HPC 和 ML 應用程式提供更好的應用程式效能。雖然 ENA 裝置提供傳統 IP 網路。如需根據您的使用案例使用哪些執行個體組態的指引，請參閱 最大化網路頻寬。



傳統上，AI/ML 應用程式使用 NCCL 和 NIXL （用於分解推論）。HPC 應用程式使用訊息傳遞界面 (MPI) 與系統的網路傳輸界面。在 AWS 雲端中，這表示應用程式會與 NCCL、NIXL 或 MPI 連接，然後會使用作業系統的 TCP/IP 堆疊和 ENA 裝置驅動程式來啟用執行個體之間的網路通訊。

使用傳統 EFA (EFA 搭配 ENA) 或僅限 EFA 界面時，AI/ML 應用程式會使用 NCCL 和 NIXL （用於分解推論）。HPC 應用程式使用 MPI，直接與 Libfabric API 連接。Libfabric API 會繞過作業系統的核心，直接與 EFA 裝置通訊來將封包放到網路上。這減少了額外負荷，使 AI/ML 和 HPC 應用程式能夠更有效率地執行。

> **Note**
>
> Libfabric 是 OpenFabrics Interfaces (OFI) 框架的核心元件，定義並匯出 OFI 的使用者空間 API。如需詳細資訊，請參閱 Libfabric OpenFabrics 網站。

ENA、EFA 和僅限 EFA 網路介面之間的差異

Amazon EC2 提供兩種類型的網路介面：

- ENA 介面提供支援 VPC IP 聯網所需的所有傳統 IP 聯網和路由功能。如需詳細資訊，請參閱在 EC2 執行個體上使用 ENA 啟用增強型網路。
- EFA (EFA 搭配 ENA) 介面提供用於 IP 聯網的 ENA 裝置和用於低延遲、高輸送量通訊的 EFA 裝置。
- 僅 EFA 介面僅支援 EFA 裝置功能，沒有傳統 IP 網路的 ENA 裝置。

下表提供 ENA、EFA 和僅限 EFA 網路介面的比較。

	ENA	EFA (EFA 搭配 ENA)	僅限 EFA
支援 IP 網路功能	是	是	否
可以指派 IPv4 或 IPv6 地址	是	是	否
可用作執行個體的主要網路介面	是	是	否
計入執行個體的 ENI 連接限制	是	是	是
執行個體類型支援	支援所有 Nitro 型執行個體類型	支援的執行個體類型	支援的執行個體類型
EC2 APIs 中的參數命名	`interface`	`efa`	`efa-only`
EC2 主控台中的欄位命名	沒有選取項目	EFA 搭配 ENA	僅限 EFA

支援的介面和程式庫

EFA 支援下列介面和程式庫：

- 開啟 MPI 4.1 和更新版本
- Intel MPI 2019 Update 5 及更新版本
- NVIDIA Collective Communications Library (NCCL) 2.4.2 和更新版本
- NVIDIA Inference Xfer Library (NIXL) 1.0.0 及更新版本
- AWS Neuron SDK 2.3 版及更新版本

支援的執行個體類型

下列所有執行個體類型都支援 EFA。此外，資料表還指示支援執行個體類型的 RDMA 讀取和 RDMA 寫入。

Nitro v6 (EFA v4)

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
一般用途		
m8a.48xlarge	Yes	Yes
m8a.metal-48xl	Yes	Yes
m8azn.24xlarge	Yes	Yes
m8azn.metal-24xl	Yes	Yes
m8gb.16xlarge	Yes	Yes
m8gb.24xlarge	Yes	Yes
m8gb.48xlarge	Yes	Yes
m8gb.metal-24xl	Yes	Yes
m8gb.metal-48xl	Yes	Yes
m8gn.16xlarge	Yes	Yes
m8gn.24xlarge	Yes	Yes
m8gn.48xlarge	Yes	Yes
m8gn.metal-24xl	Yes	Yes
m8gn.metal-48xl	Yes	Yes
m8i.48xlarge	Yes	Yes
m8i.96xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
m8i.metal-48xl	Yes	Yes
m8i.metal-96xl	Yes	Yes
m8id.48xlarge	Yes	Yes
m8id.96xlarge	Yes	Yes
m8id.metal-48xl	Yes	Yes
m8id.metal-96xl	Yes	Yes
m8in.48xlarge	Yes	Yes
m8in.96xlarge	Yes	Yes
m8in.metal-48xl	Yes	Yes
m8in.metal-96xl	Yes	Yes
m8idn.48xlarge	Yes	Yes
m8idn.96xlarge	Yes	Yes
m8idn.metal-48xl	Yes	Yes
m8idn.metal-96xl	Yes	Yes
m8ib.48xlarge	Yes	Yes
m8ib.96xlarge	Yes	Yes
m8ib.metal-48xl	Yes	Yes
m8ib.metal-96xl	Yes	Yes
m8idb.48xlarge	Yes	Yes
m8idb.96xlarge	Yes	Yes
m8idb.metal-48xl	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
m8idb.metal-96xl	Yes	Yes
m9g.48xlarge	Yes	Yes
m9g.metal-48xl	Yes	Yes
m9gd.48xlarge	Yes	Yes
m9gd.metal-48xl	Yes	Yes
運算最佳化		
c8a.48xlarge	Yes	Yes
c8a.metal-48xl	Yes	Yes
c8gb.16xlarge	Yes	Yes
c8gb.24xlarge	Yes	Yes
c8gb.48xlarge	Yes	Yes
c8gb.metal-24xl	Yes	Yes
c8gb.metal-48xl	Yes	Yes
c8gn.16xlarge	Yes	Yes
c8gn.24xlarge	Yes	Yes
c8gn.48xlarge	Yes	Yes
c8gn.metal-24xl	Yes	Yes
c8gn.metal-48xl	Yes	Yes
c8i.48xlarge	Yes	Yes
c8i.96xlarge	Yes	Yes
c8i.metal-48xl	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
c8i.metal-96xl	Yes	Yes
c8id.48xlarge	Yes	Yes
c8id.96xlarge	Yes	Yes
c8id.metal-48xl	Yes	Yes
c8id.metal-96xl	Yes	Yes
c8in.48xlarge	Yes	Yes
c8in.96xlarge	Yes	Yes
c8in.metal-48xl	Yes	Yes
c8in.metal-96xl	Yes	Yes
c8ib.48xlarge	Yes	Yes
c8ib.96xlarge	Yes	Yes
c8ib.metal-48xl	Yes	Yes
c8ib.metal-96xl	Yes	Yes
c9g.48xlarge	Yes	Yes
c9g.metal-48xl	Yes	Yes
c9gd.48xlarge	Yes	Yes
c9gd.metal-48xl	Yes	Yes
記憶體最佳化		
r8a.48xlarge	Yes	Yes
r8a.metal-48xl	Yes	Yes
r8gb.16xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
r8gb.24xlarge	Yes	Yes
r8gb.48xlarge	Yes	Yes
r8gb.metal-24xl	Yes	Yes
r8gb.metal-48xl	Yes	Yes
r8gn.16xlarge	Yes	Yes
r8gn.24xlarge	Yes	Yes
r8gn.48xlarge	Yes	Yes
r8gn.metal-24xl	Yes	Yes
r8gn.metal-48xl	Yes	Yes
r8i.48xlarge	Yes	Yes
r8i.96xlarge	Yes	Yes
r8i.metal-48xl	Yes	Yes
r8i.metal-96xl	Yes	Yes
r8id.48xlarge	Yes	Yes
r8id.96xlarge	Yes	Yes
r8id.metal-48xl	Yes	Yes
r8id.metal-96xl	Yes	Yes
r8in.48xlarge	Yes	Yes
r8in.96xlarge	Yes	Yes
r8in.metal-48xl	Yes	Yes
r8in.metal-96xl	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
r8idn.48xlarge	Yes	Yes
r8idn.96xlarge	Yes	Yes
r8idn.metal-48xl	Yes	Yes
r8idn.metal-96xl	Yes	Yes
r8ib.48xlarge	Yes	Yes
r8ib.96xlarge	Yes	Yes
r8ib.metal-48xl	Yes	Yes
r8ib.metal-96xl	Yes	Yes
r8idb.48xlarge	Yes	Yes
r8idb.96xlarge	Yes	Yes
r8idb.metal-48xl	Yes	Yes
r8idb.metal-96xl	Yes	Yes
r9g.48xlarge	Yes	Yes
r9g.metal-48xl	Yes	Yes
r9gd.48xlarge	Yes	Yes
r9gd.metal-48xl	Yes	Yes
x8aedz.24xlarge	Yes	Yes
x8aedz.metal-24xl	Yes	Yes
x8i.48xlarge	Yes	Yes
x8i.64xlarge	Yes	Yes
x8i.96xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
x8i.metal-48xl	Yes	Yes
x8i.metal-96xl	Yes	Yes
儲存優化		
i8ge.48xlarge	Yes	Yes
i8ge.metal-48xl	Yes	Yes
加速運算		
g7.8xlarge	Yes	Yes
g7.12xlarge	Yes	Yes
g7.24xlarge	Yes	Yes
g7.48xlarge	Yes	Yes
g7e.8xlarge	Yes	Yes
g7e.12xlarge	Yes	Yes
g7e.24xlarge	Yes	Yes
g7e.48xlarge	Yes	Yes
p6-b200.48xlarge	Yes	Yes
p6-b300.48xlarge	Yes	Yes
高效能運算		
hpc8a.96xlarge	Yes	Yes

Nitro v5 (EFA v3)

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
一般用途		
m8g.24xlarge	Yes	Yes
m8g.48xlarge	Yes	Yes
m8g.metal-24xl	Yes	Yes
m8g.metal-48xl	Yes	Yes
m8gd.24xlarge	Yes	Yes
m8gd.48xlarge	Yes	Yes
m8gd.metal-24xl	Yes	Yes
m8gd.metal-48xl	Yes	Yes
運算最佳化		
c7gn.16xlarge	Yes	No
c7gn.metal	Yes	No
c8g.24xlarge	Yes	Yes
c8g.48xlarge	Yes	Yes
c8g.metal-24xl	Yes	Yes
c8g.metal-48xl	Yes	Yes
c8gd.24xlarge	Yes	Yes
c8gd.48xlarge	Yes	Yes
c8gd.metal-24xl	Yes	Yes
c8gd.metal-48xl	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
記憶體最佳化		
r8g.24xlarge	Yes	Yes
r8g.48xlarge	Yes	Yes
r8g.metal-24xl	Yes	Yes
r8g.metal-48xl	Yes	Yes
r8gd.24xlarge	Yes	Yes
r8gd.48xlarge	Yes	Yes
r8gd.metal-24xl	Yes	Yes
r8gd.metal-48xl	Yes	Yes
x8g.24xlarge	Yes	Yes
x8g.48xlarge	Yes	Yes
x8g.metal-24xl	Yes	Yes
x8g.metal-48xl	Yes	Yes
儲存優化		
i7ie.48xlarge	Yes	Yes
i7ie.metal-48xl	Yes	Yes
i8g.48xlarge	Yes	Yes
i8g.metal-48xl	Yes	Yes
加速運算		
p5en.48xlarge	Yes	Yes
p6e-gb200.36xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
trn2.3xlarge	Yes	Yes
trn2.48xlarge	Yes	Yes
trn2u.48xlarge	Yes	Yes
高效能運算		
hpc7g.4xlarge	Yes	No
hpc7g.8xlarge	Yes	No
hpc7g.16xlarge	Yes	No

Nitro v4 (EFA v2)

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
一般用途		
m6a.48xlarge	Yes	Yes
m6a.metal	Yes	Yes
m6i.32xlarge	Yes	Yes
m6i.metal	Yes	Yes
m6id.32xlarge	Yes	Yes
m6id.metal	Yes	Yes
m6idn.32xlarge	Yes	Yes
m6idn.metal	Yes	Yes
m6in.32xlarge	Yes	Yes
m6in.metal	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
m7a.48xlarge	Yes	Yes
m7a.metal-48xl	Yes	Yes
m7g.16xlarge	Yes	Yes
m7g.metal	Yes	Yes
m7gd.16xlarge	Yes	Yes
m7gd.metal	Yes	Yes
m7i.48xlarge	Yes	Yes
m7i.metal-48xl	Yes	Yes
運算最佳化		
c6a.48xlarge	Yes	Yes
c6a.metal	Yes	Yes
c6gn.16xlarge	Yes	Yes
c6i.32xlarge	Yes	Yes
c6i.metal	Yes	Yes
c6id.32xlarge	Yes	Yes
c6id.metal	Yes	Yes
c6in.32xlarge	Yes	Yes
c6in.metal	Yes	Yes
c7a.48xlarge	Yes	Yes
c7a.metal-48xl	Yes	Yes
c7g.16xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
c7g.metal	Yes	Yes
c7gd.16xlarge	Yes	Yes
c7gd.metal	Yes	Yes
c7i.48xlarge	Yes	Yes
c7i.metal-48xl	Yes	Yes
記憶體最佳化		
r6a.48xlarge	Yes	Yes
r6a.metal	Yes	Yes
r6i.32xlarge	Yes	Yes
r6i.metal	Yes	Yes
r6id.32xlarge	Yes	Yes
r6id.metal	Yes	Yes
r6idn.32xlarge	Yes	Yes
r6idn.metal	Yes	Yes
r6in.32xlarge	Yes	Yes
r6in.metal	Yes	Yes
r7a.48xlarge	Yes	Yes
r7a.metal-48xl	Yes	Yes
r7g.16xlarge	Yes	Yes
r7g.metal	Yes	Yes
r7gd.16xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
r7gd.metal	Yes	Yes
r7i.48xlarge	Yes	Yes
r7i.metal-48xl	Yes	Yes
r7iz.32xlarge	Yes	Yes
r7iz.metal-32xl	Yes	Yes
u7i-6tb.112xlarge	Yes	Yes
u7i-8tb.112xlarge	Yes	Yes
u7i-12tb.224xlarge	Yes	Yes
u7in-16tb.224xlarge	Yes	Yes
u7in-24tb.224xlarge	Yes	Yes
u7in-32tb.224xlarge	Yes	Yes
u7inh-32tb.480xlarge	Yes	Yes
x2idn.32xlarge	Yes	Yes
x2idn.metal	Yes	Yes
x2iedn.32xlarge	Yes	Yes
x2iedn.metal	Yes	Yes
儲存優化		
i4g.16xlarge	Yes	Yes
i4i.32xlarge	Yes	Yes
i4i.metal	Yes	Yes
i7i.24xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
i7i.48xlarge	Yes	Yes
i7i.metal-48xl	Yes	Yes
im4gn.16xlarge	Yes	Yes
加速運算		
f2.48xlarge	Yes	Yes
g6.8xlarge	Yes	Yes
g6.12xlarge	Yes	Yes
g6.16xlarge	Yes	Yes
g6.24xlarge	Yes	Yes
g6.48xlarge	Yes	Yes
g6e.8xlarge	Yes	Yes
g6e.12xlarge	Yes	Yes
g6e.16xlarge	Yes	Yes
g6e.24xlarge	Yes	Yes
g6e.48xlarge	Yes	Yes
gr6.8xlarge	Yes	Yes
p5.4xlarge	Yes	Yes
p5.48xlarge	Yes	Yes
p5e.48xlarge	Yes	Yes
trn1.32xlarge	Yes	Yes
trn1n.32xlarge	Yes	Yes

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
高效能運算		
hpc6a.48xlarge	Yes	Yes
hpc6id.32xlarge	Yes	Yes
hpc7a.12xlarge	Yes	Yes
hpc7a.24xlarge	Yes	Yes
hpc7a.48xlarge	Yes	Yes
hpc7a.96xlarge	Yes	Yes

Nitro v3 (EFA v1)

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
一般用途		
m5dn.24xlarge	No	No
m5dn.metal	No	No
m5n.24xlarge	No	No
m5n.metal	No	No
m5zn.12xlarge	No	No
m5zn.metal	No	No
運算最佳化		
c5n.9xlarge	No	No
c5n.18xlarge	No	No
c5n.metal	No	No

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
記憶體最佳化		
r5dn.24xlarge	No	No
r5dn.metal	No	No
r5n.24xlarge	No	No
r5n.metal	No	No
x2iezn.12xlarge	No	No
x2iezn.metal	No	No
儲存優化		
i3en.12xlarge	No	No
i3en.24xlarge	No	No
i3en.metal	No	No
加速運算		
dl2q.24xlarge	No	No
g4dn.8xlarge	No	No
g4dn.12xlarge	No	No
g4dn.16xlarge	No	No
g4dn.metal	No	No
g5.8xlarge	No	No
g5.12xlarge	No	No
g5.16xlarge	No	No
g5.24xlarge	No	No

執行個體類型	支援 RDMA 讀取	支援 RDMA 寫入
g5.48xlarge	No	No
inf1.24xlarge	No	No
p3dn.24xlarge	No	No
p4d.24xlarge	Yes	No
p4de.24xlarge	Yes	No
vt1.24xlarge	No	No
上一代		
p3dn.24xlarge	No	No

查看特定區域中支援 EFA 的可用執行個體類型

可用的執行個體類型因區域而異。若要查看區域中支援 EFA 的可用執行個體類型，請使用 describe-instance-types 命令，搭配 `--region` 參數。包含 `--filters` 參數以將結果範圍限定為支援 EFA 的執行個體類型，以及包含 `--query` 參數以將輸出範圍限定為 `InstanceType` 的值。

```
aws ec2 describe-instance-types \
    --region us-east-1  \
    --filters Name=network-info.efa-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]"  \
    --output text | sort
```

支援的作業系統

作業系統支援因處理器類型而異。下表顯示受支援作業系統的規則套件可用性。

作業系統	Intel/AMD (**x86_64**) 執行個體類型	AWS Graviton (**arm64**) 執行個體類型
Amazon Linux 2023	✓	✓

作業系統	Intel/AMD (**x86_64**) 執行個體類型	AWS Graviton (**arm64**) 執行個體類型
RHEL 8、9 和 10	✓	✓
Debian 11、12 和 13	✓	✓
Rocky Linux 8 和 9	✓	✓
Ubuntu 22.04、24.04 和 26.04	✓	✓
SUSE Linux Enterprise 15 SP2 及更新版本	✓	✓

> **Note**
>
> Intel MPI 可能不支援列出的某些作業系統。如果您使用的是 Intel MPI，請參閱 Intel MPI 文件來驗證對作業系統的支援。

EFA 限制

EFA 具有下列限制：

- 並非所有執行個體類型都支援 RDMA 寫入。如需詳細資訊，請參閱支援的執行個體類型。
- 目前不支援 P4d/P4de/DL1 執行個體與其他執行個體類型之間的 EFA 流量[1]。
- 支援多個網路卡的執行個體類型可以設定為每個網路卡一個 EFA。所有其他支援的執行個體類型每個執行個體只支援一個 EFA。
- 在連接 EFA 時，不支援 `c7g.16xlarge`、`m7g.16xlarge` 和 `r7g.16xlarge` 專用執行個體和專用主機。
- EFA 流量[1]無法跨可用區域或 VPC。這不適用於來自 EFA 介面 ENA 裝置的一般 IP 流量。
- EFA 流量[1]不可路由傳送。來自 EFA 介面 ENA 裝置的正常 IP 流量仍然可以路由。

- 不支援 EFA AWS Outposts。
- 只有以 AWS Cloud Digital Interface 軟體開發套件 (AWS CDI SDK) 為基礎的應用程式，Windows 執行個體才支援 EFA (EFA 搭配 ENA) 界面的 EFA 裝置。如果您將 EFA (EFA 搭配 ENA) 介面連接至非 CDI SDK 型應用程式的 Windows 執行個體，則其可做為 ENA 介面運作，無需新增 EFA 裝置功能。Windows 或 Linux 上的 AWS CDI 應用程式不支援僅限 EFA 界面。如需詳細資訊，請參閱AWS Cloud Digital Interface 軟體開發套件 (AWS CDI SDK) 使用者指南。

[1]EFA 流量是指透過 EFA (EFA 搭配 ENA) 或僅限 EFA 介面的 EFA 裝置傳輸的流量。

EFA 定價

EFA 是選用的 Amazon EC2 聯網功能，您可以在任何支援的執行個體上啟用該功能，而無需額外付費。

Amazon EC2 上 HPC 工作負載的 EFA 和 MPI 入門

本教學課程協助您啟動 EFA 和 MPI 啟用的執行個體叢集用於 HPC 工作負載。

> **Note**
>
> `u7i-12tb.224xlarge`、`u7in-16tb.224xlarge`、`u7in-24tb.224xlarge` 和 `u7in-32tb.224xlarge`執行個體可以使用 Open MPI 執行最多 128 個平行 MPI 程序，或使用 Intel MPI 執行最多 256 個平行 MPI 程序。

任務

- 步驟 1：準備具備 EFA 功能的安全群組
- 步驟 2：啟動暫時執行個體
- 步驟 3：安裝 EFA 軟體
- 步驟 4：(選用) 啟用 Open MPI 5
- 步驟 5：(選用) 安裝 Intel MPI
- 步驟 6：停用 ptrace 保護
- 步驟 7. 確認安裝
- 步驟 8：安裝 HPC 應用程式
- 步驟 9：建立已啟用 EFA 的 AMI
- 步驟 10：在集群放置群組中啟動已啟用 EFA 的執行個體

- 步驟 11：終止暫時執行個體
- 步驟 12：啟用無密碼 SSH

步驟 1：準備具備 EFA 功能的安全群組

EFA 需要有安全群組來允許往返安全群組自身的所有傳入和傳出流量。下列程序會建立一個安全群組，該安全群組允許所有傳入和傳出流量，並允許來自任何 IPv4 地址的傳入 SSH 流量以進行 SSH 連線。

> **Important**
>
> 此安全群組僅供測試之用。對於生產環境，我們建議您建立傳入 SSH 規則，該規則僅允許來自連線 IP 位址的流量，例如電腦的 IP 位址或區域網路中的一系列 IP 位址。

如需其他案例，請參閱不同使用案例的安全群組規則。

建立具備 EFA 功能的安全群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)，然後選擇 Create Security Group (建立安全群組)。
3. 在 Create Security Group (建立安全群組) 視窗中，執行下列動作：
   a. 對於 Security group name (安全群組名稱)，輸入安全群組的描述性名稱，例如 `EFA-enabled security group`。
   b. (選用) 對於 Description (描述)，輸入安全群組的簡短描述。
   c. 對於 VPC，選取您打算讓具備 EFA 功能的執行個體在其中啟動的 VPC。
   d. 選擇 Create Security Group (建立安全群組)。
4. 選取您建立的安全群組，在 Details (詳細資料) 索引標籤上，複製 Security group ID (安全群組 ID)。
5. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit inbound rules (編輯傳入規則)，然後執行下列動作。
   a. 選擇新增規則。
   b. 針對類型，選擇所有流量。

c. 若為 Source type (來源類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。

d. 選擇新增規則。

e. 針對 Type (類型)，選擇 SSH。

f. 針對 Source type (來源類別)，選擇 Anywhere-IPv4(隨處 - IPv4)。

g. 選擇儲存規則。

6. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit outbound rules (編輯傳出規則)，然後執行下列動作。

a. 選擇新增規則。

b. 針對類型，選擇所有流量。

c. 若為 Destination type (目的地類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。

d. 選擇儲存規則。

> **Important**
>
> 自我參考傳入和傳出規則（允許進出安全群組本身的所有流量）是 EFA 運作的必要條件。如果沒有這些規則，執行個體之間的 EFA 流量將遭到封鎖。

步驟 2：啟動暫時執行個體

啟動暫時執行個體，以用來安裝和設定 EFA 軟體元件。您可以使用此執行個體來建立具備 EFA 功能的 AMI，再從中啟動具備 EFA 功能的執行個體。

啟動暫時執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，`EFA-instance`。該名稱將指派作為執行個體的資源標籤 (Name=*`EFA-instance`*)。
4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，為其中一個支援的作業系統選取 AMI。

5. 在 Instance type (執行個體類型) 區段中，選取支援的執行個體類型。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：

   a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。

   > **Important**
   >
   > 您必須選取子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。

   b. 針對 Firewall (security groups) (防火牆 (安全群組))，選取 Select existing security group (選取現有的安全群組)，接著選取您在前一個步驟中建立的安全群組。

   c. 展開進階設定區段。

      針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

      (選用) 如果您使用的是多卡執行個體類型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，則針對所需的每個額外網路介面，選擇新增網路介面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1，介面類型 = EFA 搭配 ENA 或僅限 EFA。
8. 在儲存區段中，根據需求設定磁碟區。
9. 在右邊的 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

> **Note**
>
> 請考慮要求將 IMDSv2 用於暫時執行個體，以及您在步驟 9 中建立的 AMI，除非您已將 IMDSv2 設定為帳戶的預設值。如需 IMDSv2 設定步驟的相關資訊，請參閱 設定新執行個體的執行個體中繼資料選項。

步驟 3：安裝 EFA 軟體

在暫存執行個體上安裝支援 EFA 所需的具備 EFA 功能的核心、EFA 驅動程式、Libfabric 和 Open MPI 堆疊。

該步驟會根據您是否打算將 EFA 與 Open MPI、Intel MPI 或 Open MPI 與 Intel MPI 搭配使用而有所不同。

> **Note**
>
> Intel MPI 可能不支援某些作業系統。如果您使用的是 Intel MPI，請參閱 Intel MPI 文件來驗證對作業系統的支援。

安裝 EFA 軟體

1. 連接至您啟動的執行個體。如需詳細資訊，請參閱 使用 SSH 連線至您的 Linux 執行個體。
2. 為確保所有軟體套件皆為最新版本，請對您的執行個體執行快速軟體更新。此程序可能需要幾分鐘的時間。

   - Amazon Linux 2023、RHEL 8/9、Rocky Linux 8/9

     ```
     $ sudo yum update -y
     ```

   - Ubuntu 和 Debian

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

   - SUSE Linux Enterprise

     ```
     $ sudo zypper update -y
     ```

3. 重新啟動執行個體並重新連線至其中。
4. 下載 EFA 軟體安裝檔案。軟體安裝檔案已封裝成壓縮 tarball (`.tar.gz`) 檔案。若要下載最新穩定版本，請使用下列命令：

   您也可以在之前的命令中以 `latest` 取代版本號碼，以取得最新版本。

   ```
   $ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.50.0.tar.gz
   ```

5. (選用) 驗證 EFA tarball (`.tar.gz`) 檔案的真確性及完整性。

   我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。如果您不想驗證 tarball 檔案，請略過此步驟。

> **Note**
>
> 或者，如果您偏好使用 MD5 或 SHA256 檢查總和來驗證 tarball 檔案，請參閱使用檢查總和驗證 EFA 安裝程式。

a. 下載公有 GPG 金鑰並匯入至您的 keyring。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
import aws-efa-installer.key
```

命令應傳回金鑰值。請記下金鑰的值，因為下一個步驟將需要它。

b. 驗證 GPG 金鑰的指紋。執行以下命令，並指定上一步驟中的金鑰值。

```
$ gpg --fingerprint key_value
```

命令應傳回與 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC 相同的指紋。如果指紋不相符，請勿執行 EFA 安裝指令碼，並聯絡 支援。

c. 下載簽章檔案並驗證 EFA tarball 檔案的簽章。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.50.0.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.50.0.tar.gz.sig
```

下面顯示了範例輸出。

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

如果結果包含Good signature，而指紋與上一個步驟中傳回的指紋相符，請繼續下一個步驟。如果指紋不相符，請勿執行 EFA 安裝指令碼，並聯絡 支援。

6. 從壓縮的 .tar.gz 檔案中解壓縮檔案，然後導覽至解壓縮的目錄。

```
$ tar -xf aws-efa-installer-1.50.0.tar.gz && cd aws-efa-installer
```

7. (選用) 在安裝期間驗證個別套件簽章。

   從 EFA 安裝程式 1.48.0 開始，安裝程式包含 GPG 簽署的個別 RPM 和 DEB 套件。若要在安裝期間驗證每個個別套件的真實性和完整性，請使用 `--check-signatures`旗標。當您啟用此旗標時，安裝程式會先驗證所有套件簽章，而且只有在每個套件通過驗證時才會繼續安裝。如果任何套件驗證失敗，安裝程式會立即結束，而無需安裝任何項目。

   a. 下載 GPG 公有金鑰。

   ```
   $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
   ```

   b. 匯出金鑰路徑。然後，在下一個步驟中，將 附加`--check-signatures`到安裝命令，並使用 `sudo -E`而不是 sudo 來保留環境變數。

   ```
   $ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
   ```

   在 RPM 型系統上 (Amazon Linux、RHEL、Rocky Linux 和 SUSE)，安裝程式會使用 驗證每個 RPM`rpm --checksig`。在 DEB 型系統 (Ubuntu、Debian) 上，安裝程式會使用 GPG 簽章驗證來驗證每個 DEB。

   如果任何套件的驗證失敗，安裝會立即中止，保護您的系統免於損壞或惡意的套件。

   > **Note**
   >
   > `--check-signatures` 旗標為選用。如果沒有，安裝程式不會執行個別簽章驗證。

8. 安裝 EFA 軟體。根據使用案例而定，執行以下其中一個命令。

   > **Note**
   >
   > 如果您已完成上一個設定套件簽章驗證的選用步驟，請附加`--check-signatures`至安裝命令並使用 `sudo -E`，而非 sudo。例如：`sudo -E ./efa_installer.sh -y --check-signatures`。

> **Note**
>
> EFA 不支援使用 SUSE Linux 的 NVIDIA GPUDirect。如果您使用的是 SUSE Linux，您必須另外指定 `--skip-kmod` 選項來防止 kmod 安裝。根據預設，SUSE Linux 不允許樹狀結構外的核心模組。

Open MPI and Intel MPI

如果您打算將 EFA 與 Open MPI 和 Intel MPI 搭配使用，您必須使用 Libfabric 和 Open MPI 安裝 EFA 軟體，且必須完成步驟 5：安裝 Intel MPI。

若要使用 Libfabric 和 Open MPI 安裝 EFA 軟體，請執行下列命令。

> **Note**
>
> 從 EFA 1.30.0 開始，預設會同時安裝 Open MPI 4.1 和 Open MPI 5。您可以選擇是否指定您要安裝的 Open MPI 版本。若要僅安裝 Open MPI 4.1，請包含 `--mpi=openmpi4`。若僅要安裝 Open MPI 5，請包括 `--mpi=openmpi5`。若要同時安裝這兩種版本，請忽略 `--mpi` 選項。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 會安裝到 `/opt/amazon/efa`。Open MPI 4.1 已安裝至 。 `/opt/amazon/openmpi`Open MPI 5 會安裝到 `/opt/amazon/openmpi5`。

Open MPI only

如果您打算將 EFA 與 Open MPI 搭配使用，您必須使用 Libfabric 和 Open MPI 安裝 EFA 軟體，且必須跳過步驟 5：安裝 Intel MPI。若要使用 Libfabric 和 Open MPI 安裝 EFA 軟體，請執行下列命令。

> **Note**
>
> 從 EFA 1.30.0 開始，預設會同時安裝 Open MPI 4.1 和 Open MPI 5。您可以選擇是否指定您要安裝的 Open MPI 版本。若要僅安裝 Open MPI 4.1，請包含 --

mpi=openmpi4。若僅要安裝 Open MPI 5，請包括 --mpi=openmpi5。若要同時安裝這兩種版本，請忽略 --mpi 選項。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 會安裝到 /opt/amazon/efa。Open MPI 4.1 已安裝至 。 /opt/amazon/openmpiOpen MPI 5 會安裝到 /opt/amazon/openmpi5。

Intel MPI only

如果您打算只將 EFA 與 Intel MPI 搭配使用，您無需使用 Libfabric 和 Open MPI 即可安裝 EFA 軟體。在這種情況下，Intel MPI 會使用其內嵌 Libfabric。如果您選擇這麼做，您必須完成步驟 5：安裝 Intel MPI。

若要在不使用 Libfabric 和 Open MPI 的情況下安裝 EFA 軟體，請執行下列命令。

```
$ sudo ./efa_installer.sh -y --minimal
```

9. 如果 EFA 安裝程式提示您重新啟動執行個體，請執行這項操作，然後重新連線至執行個體。否則，請登出執行個體，然後重新登入以完成安裝。
10. 刪除未壓縮的 tarball 和 tarball 本身。否則，其中會包含在您建立的已啟用 EFA 的 AMI 中，從而增加其大小。

步驟 4：(選用) 啟用 Open MPI 5

> **Note**
>
> 只有在您打算使用 Open MPI 5 時，才執行此步驟。

從 EFA 1.30.0 開始，預設會同時安裝 Open MPI 4.1 和 Open MPI 5。或者，您可以選擇僅安裝 Open MPI 4.1 或 Open MPI 5。

如果您在步驟 3：安裝 EFA 軟體中選擇安裝且有意使用 Open MPI 5，則必須執行下列步驟以啟用 Open MPI 5。

啟用 Open MPI 5

1. 將 Open MPI 5 新增至 PATH 環境變數。

```
$ module load openmpi5
```

2. 確認是否已啟用 Open MPI 5 以供使用。

```
$ which mpicc
```

此命令應會傳回 Open MPI 5 安裝目錄 - /opt/amazon/openmpi5。

3. (選用) 若要確保在每次執行個體啟動時，Open MPI 5 都會新增至 PATH 環境變數，請執行下列動作：

bash shell

將 `module load openmpi5` 新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

csh and tcsh shells

將 `module load openmpi5` 新增至 /home/*username*/.cshrc。

如果您需要從 PATH 環境變數移除 Open MPI 5，請執行下列命令，並從 shell 啟動指令碼中移除該命令。

```
$ module unload openmpi5
```

步驟 5：(選用) 安裝 Intel MPI

> **Important**
> 只有在您打算使用 Intel MPI 時，才執行此步驟。如果您打算使用 Open MPI，請略過此步驟。

Intel MPI 需要額外的安裝和環境變數組態。

先決條件

確保執行下列步驟的使用者具有 sudo 許可。

安裝 Intel MPI

1. 如要下載 Intel MPI 安裝指令碼，請執行下列步驟

   a. 請造訪 Intel 網站。

   b. 在該網頁的 Intel MPI Library 區段中，選擇 Intel MPI Library for Linux Offline 安裝程式。

2. 執行您在上一個步驟中下載的安裝指令碼。

```
$ sudo bash installation_script_name.sh
```

3. 在安裝程式中，選擇 Accept & install (接受並安裝)。
4. 閱讀 Intel 改進計畫，選擇適當的選項，然後選擇 Begin Installation (開始安裝)。
5. 完成安裝時，請選擇 Close (關閉)。
6. 根據預設，Intel MPI 會使用其內嵌 (內部) Libfabric。您可以將 Intel MPI 設定為使用 EFA 安裝程式隨附的 LibFabric。通常，EFA 安裝程式隨附的 Libfabric 版本比 Intel MPI 版本更高。在某些情況下，EFA 安裝程式隨附的 Libfabric 比 Intel MPI 效能更佳。若要將 Intel MPI 設定為使用 EFA 安裝程式隨附的 Libfabric，請根據您的 shell 執行下列其中一項操作。

   bash shells

   將以下陳述式新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export I_MPI_OFI_LIBRARY_INTERNAL=0
```

   csh and tcsh shells

   將以下陳述式新增至 /home/*username*/.cshrc。

```
setenv I_MPI_OFI_LIBRARY_INTERNAL 0
```

7. 新增下列 source 命令到您的 shell 指令碼，從安裝目錄中獲取 `vars.sh` 的指令碼，從而在每次啟動執行個體時設定編譯器環境。根據 shell 而定，執行以下其中一個命令。

   bash shells

   將以下陳述式新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
source /opt/intel/oneapi/mpi/latest/env/vars.sh
```

csh and tcsh shells

將以下陳述式新增至 /home/*username*/.cshrc。

```
source /opt/intel/oneapi/mpi/latest/env/vars.csh
```

8. 根據預設，如果因為設定錯誤而無法使用 EFA，Intel MPI 會預設使用 TCP/IP 網路堆疊，這可能會導致應用程式效能降低。透過將 I_MPI_OFI_PROVIDER 設定為 efa，可防止此情況發生。如果無法使用 EFA，這會導致 Intel MPI 失敗，並出現下列錯誤：

```
Abort (XXXXXX) on node 0 (rank 0 in comm 0): Fatal error in PMPI_Init: OtherMPI
 error,
MPIR_Init_thread (XXX).......:
MPID_Init (XXXX).............:
MPIDI_OFI_mpi_init_hook (XXXX):
open_fabric (XXXX)...........:
find_provider (XXXX).........:
OFI fi_getinfo() failed (ofi_init.c:2684:find_provider:
```

根據 shell 而定，執行以下其中一個命令。

bash shells

將以下陳述式新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export I_MPI_OFI_PROVIDER=efa
```

csh and tcsh shells

將以下陳述式新增至 /home/*username*/.cshrc。

```
setenv I_MPI_OFI_PROVIDER efa
```

9. 根據預設，Intel MPI 不會列印偵錯資訊。您可以指定不同的詳細資訊等級來控制偵錯資訊。可能的值 (依提供的詳細資訊量排序) 為：0 (預設)、1、2、3、4、5。等級 1 和更高等級會列印 libfabric version 和 libfabric provider。使用 libfabric version 檢查 Intel MPI

使用的是內部 LibFabric 還是 EFA 安裝程式隨附的 LibFabric。如果它使用的是內部 Libfabric，則版本字尾為 `impi`。使用 `libfabric provider` 檢查 Intel MPI 是否的是 EFA 還是或 TCP/IP 網絡。如果使用的是 EFA，則值為 `efa`。如果使用的是 TCP/IP，則值為 `tcp;ofi_rxm`。

若要啟用偵錯資訊，請根據您的 shell 執行以下其中一個命令。

bash shells

將以下陳述式新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export I_MPI_DEBUG=value
```

csh and tcsh shells

將以下陳述式新增至 /home/*username*/.cshrc。

```
setenv I_MPI_DEBUG value
```

10. 根據預設，Intel MPI 會使用作業系統的共用記憶體 (shm) 進行節點內通訊，且僅使用 Libfabric (ofi) 進行節點間通訊。通常，此組態可提供最佳效能。不過，在某些情況下，Intel MPI shm 結構可能會導致某些應用程式無限期中止。

    若要解決此問題，您可以強制 Intel MPI 使用 Libfabric 進行節點內和節點間通訊。為此，請根據您的 shell，執行以下其中一個命令。

    bash shells

    將以下陳述式新增至 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

    ```
    export I_MPI_FABRICS=ofi
    ```

    csh and tcsh shells

    將以下陳述式新增至 /home/*username*/.cshrc。

    ```
    setenv I_MPI_FABRICS ofi
    ```

> **Note**
>
> EFA Libfabric 提供者會使用作業系統的共用記憶體進行節點內通訊。這表示將 `I_MPI_FABRICS` 設定為 `ofi` 可產生與預設 `shm:ofi` 組態類似的效能。

11. 登出執行個體，然後重新登入。

如果您不要再使用 Intel MPI，請從 shell 啟動指令碼中移除環境變數。

步驟 6：停用 ptrace 保護

為了改善 HPC 應用程式的效能，在相同執行個體上執行處理序時，Libfabric 會使用執行個體的本機記憶體進行處理序間通訊。

共用記憶體功能使用 Cross Memory Attach (CMA)，其未受 ptrace 保護支援。如果您使用的 Linux 發行套件依預設已啟用 ptrace 防護，例如 Ubuntu，您必須停用該保護。如果您的 Linux 發行套件依預設未啟用 ptrace 保護，請略過此步驟。

停用 ptrace 保護

請執行下列其中一項：

- 如果出於測試用途要暫時停用 ptrace 保護，請執行下列命令。

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- 若要永久停用 ptrace 保護，請將 `kernel.yama.ptrace_scope = 0` 新增至 `/etc/sysctl.d/10-ptrace.conf`，並重新啟動執行個體。

步驟 7. 確認安裝

確認已成功安裝

1. 若要確認已成功安裝 MPI，請執行下列命令：

```
$ which mpicc
```

   - 若為 Open MPI，傳回的路徑應包括 `/opt/amazon/`

- 若為 Intel MPI，傳回的路徑應包括 `/opt/intel/`。如果沒有得到預期輸出，請確保已獲取 Intel MPI `vars.sh` 指令碼。

2. 若要確認是否已成功安裝 EFA 軟體元件和 Libfabric，請執行下列命令。

```
$ fi_info -p efa -t FI_EP_RDM
```

此命令應該會傳回 Libfabric EFA 介面的相關資訊。以下範例顯示命令輸出。

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
```

步驟 8：安裝 HPC 應用程式

在暫時執行個體上安裝 HPC 應用程式。安裝程序依特定的 HPC 應用程式而異。如需詳細資訊，請參閱《Amazon Linux 2023 使用者指南》中的管理作業系統更新。

> **Note**
>
> 如需安裝說明，請參閱 HPC 應用程式的文件。

步驟 9：建立已啟用 EFA 的 AMI

安裝所需的軟體元件之後，您需要建立 AMI，以重複用來啟動具備 EFA 功能的執行個體。

從暫時執行個體建立 AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選取 Actions (動作)、Image (映像)、Create image (建立映像)。
4. 對於 Create image (建立映像)，執行下列動作：

   a. 對於 Image name (映像名稱)，輸入 AMI 的描述性名稱。

b. (選用) 對於 Image description (映像描述)，輸入 AMI 的簡短描述。

c. 選擇建立映像。

5. 在導覽窗格中，選擇 AMIs。

6. 找到您在清單中建立的 AMI。等待狀態從pending變為available，再繼續進行下一個步驟。

步驟 10：在集群放置群組中啟動已啟用 EFA 的執行個體

使用具備 EFA 功能的 AMI (在步驟 7 中建立) 和具備 EFA 功能的安全群組 (在步驟 1 中建立)，在叢集置放群組中啟動具備 EFA 功能的執行個體。

> **Note**
>
> - 在叢集置放群組中啟動具備 EFA 的執行個體並非絕對必要。不過，建議在叢集置放群組中執行具備 EFA 功能的執行個體，因為這樣會在單一可用區域的低延遲群組中啟動執行個體。
> - 為了確保在擴展叢集的執行個體時有可用的容量，您可以為叢集置放群組建立容量保留。如需詳細資訊，請參閱將容量保留與置放群組搭配使用。

啟動執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。

3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，EFA-instance。該名稱將指派作為執行個體的資源標籤 (Name=*EFA-instance*)。

4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，選取 My AMIs (我的 AMI)，接著選取您在前一個步驟中建立的 AMI。

5. 在 Instance type (執行個體類型) 區段中，選取支援的執行個體類型。

6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。

7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：

a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。

> **Important**
>
> 您必須選取子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。

b. 針對 Firewall (security groups) (防火牆 (安全群組))，選取 Select existing security group (選取現有的安全群組)，接著選取您在前一個步驟中建立的安全群組。

c. 展開進階設定區段。

針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

(選用) 如果您使用的是多卡執行個體類型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，則針對所需的每個額外網路介面，選擇新增網路介面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1，介面類型 = EFA 搭配 ENA 或僅限 EFA。

8. (選用) 在儲存區段中，根據需求設定磁碟區。
9. 在進階詳細資訊區段中，針對 配置群組名稱，選取要在其中啟動執行個體的叢集置放群組。如果您需要建立新的叢集置放群組，請選取建立新的配置群組。
10. 在右邊的摘要面板，針對執行個體的數目，輸入要啟動的具備 EFA 功能的執行個體數量，接著選擇啟動執行個體。

步驟 11：終止暫時執行個體

此時，您不再需要您在步驟 2 中啟動的執行個體。您可以終止執行個體，以停止產生費用。

終止暫時執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您建立的暫時執行個體，然後選擇動作、執行個體狀態、終止 (刪除) 執行個體。
4. 出現確認提示時，請選擇終止 (刪除)。

步驟 12：啟用無密碼 SSH

若要讓您的應用程式能夠跨越叢集中的所有執行個體執行，您必須啟用從領導節點到成員節點的無密碼 SSH 存取。領導節點是您在其中執行應用程式的執行個體。叢集中的其餘執行個體為成員節點。

在叢集的執行個體之間啟用無密碼 SSH

1. 選取叢集中的一個執行個體作為領導節點，並與它連線。
2. 在領導節點上停用 strictHostKeyChecking 並啟用 ForwardAgent。使用您偏好的文字編輯器開啟 ~/.ssh/config，然後新增下列命令。

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. 產生 RSA 金鑰對。

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

在 $HOME/.ssh/ 目錄中建立金鑰對。

4. 變更領導節點上私有金鑰的許可。

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. 使用您偏好的文字編輯器開啟 ~/.ssh/id_rsa.pub 並複製該金鑰。
6. 對於叢集中的每個成員節點，執行以下操作：

   a. 連線到執行個體。

   b. 使用您偏好的文字編輯器開啟 ~/.ssh/authorized_keys，然後新增您先前複製的公有金鑰。

7. 若要測試無密碼的 SSH 是否如預期運作，請連線至領導節點並執行下列命令。

```
$ ssh member_node_private_ip
```

您應該連線至成員節點，而不會提示您輸入金鑰或密碼。

Amazon EC2 上的適用於 ML 工作負載的 EFA 和 NCCL 入門

NVIDIA Collective Communications Library (NCCL) 是一種標準集體溝通常式程式庫，用於多個 GPU 橫跨單一節點或多個節點。您可以搭配 EFA、Libfabric 和 MPI 使用 NCCL，以支援各種機器學習工作負載。如需詳細資訊，請參閱 NCCL 網站。

要求

- 支援的執行個體類型包括 EFA 支援的 P 系列和 G 系列執行個體類型。如需詳細資訊，請參閱 Amazon EC2 加速運算執行個體。
- 支援的基本 AMIs：Amazon Linux 2023、Ubuntu 26.04、Ubuntu 24.04、Ubuntu 22.04、Debian 12 和 RHEL 10。
- EFA 僅支援 NCCL 2.4.2 和更新版本。

如需使用 搭配 EFA 和 NCCL 執行機器學習工作負載的詳細資訊 AWS 深度學習 AMIs，請參閱《AWS 深度學習 AMIs 開發人員指南》中的在 DLAMI 上使用 EFA。

步驟

- 步驟 1：準備具備 EFA 功能的安全群組
- 步驟 2：啟動暫時執行個體
- 步驟 3：安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN
- 步驟 4：安裝 GDRY
- 步驟 5：安裝 EFA 軟體
- 步驟 6：安裝 NCCL
- 步驟 7：安裝 NCCL Test
- 步驟 8：測試您的 EFA 和 NCCL 組態
- 步驟 9：安裝您的機器學習應用程式
- 步驟 10：建立 EFA 和 NCCL 啟用的 AMI
- 步驟 11：終止暫時執行個體
- 步驟 12：在叢集置放群組中啟動 EFA 和具備 NCCL 功能的執行個體
- 步驟 13：啟用無密碼 SSH

步驟 1：準備具備 EFA 功能的安全群組

EFA 需要有安全群組來允許往返安全群組自身的所有傳入和傳出流量。下列程序會建立一個安全群組，該安全群組允許所有傳入和傳出流量，並允許來自任何 IPv4 地址的傳入 SSH 流量以進行 SSH 連線。

> **Important**
>
> - 此安全群組僅供測試之用。對於生產環境，我們建議您建立傳入 SSH 規則，該規則僅允許來自連線 IP 位址的流量，例如電腦的 IP 位址或區域網路中的一系列 IP 位址。
> - 自我參考傳入和傳出規則（允許進出安全群組本身的所有流量）是 EFA 運作的必要條件。如果沒有這些規則，執行個體之間的 EFA 流量將會遭到封鎖，且 NCCL 通訊將會失敗。

如需其他案例，請參閱不同使用案例的安全群組規則。

建立具備 EFA 功能的安全群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)，然後選擇 Create Security Group (建立安全群組)。
3. 在 Create Security Group (建立安全群組) 視窗中，執行下列動作：
   a. 對於 Security group name (安全群組名稱)，輸入安全群組的描述性名稱，例如 `EFA-enabled security group`。
   b. (選用) 對於 Description (描述)，輸入安全群組的簡短描述。
   c. 對於 VPC，選取您打算讓具備 EFA 功能的執行個體在其中啟動的 VPC。
   d. 選擇 Create Security Group (建立安全群組)。
4. 選取您建立的安全群組，在 Details (詳細資料) 索引標籤上，複製 Security group ID (安全群組 ID)。
5. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit inbound rules (編輯傳入規則)，然後執行下列動作。
   a. 選擇新增規則。
   b. 針對類型，選擇所有流量。
   c. 若為 Source type (來源類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。
   d. 選擇新增規則。
   e. 針對 Type (類型)，選擇 SSH。
   f. 針對 Source type (來源類別)，選擇 Anywhere-IPv4(隨處 - IPv4)。
   g. 選擇儲存規則。

6. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit outbound rules (編輯傳出規則)，然後執行下列動作。

   a. 選擇新增規則。

   b. 針對類型，選擇所有流量。

   c. 若為 Destination type (目的地類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。

   d. 選擇儲存規則。

步驟 2：啟動暫時執行個體

啟動暫時執行個體，以用來安裝和設定 EFA 軟體元件。您可以使用此執行個體來建立具備 EFA 功能的 AMI，再從中啟動具備 EFA 功能的執行個體。

啟動暫時執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，`EFA-instance`。該名稱將指派作為執行個體的資源標籤 (Name=*`EFA-instance`*)。
4. 在應用程式和作業系統映像區段中，為其中一個支援的作業系統選取 AMI。
5. 在執行個體類型區段中，選取支援的執行個體類型。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：

   a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。

   > **Important**
   >
   > 您必須選取子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。

   b. 針對 Firewall (security groups) (防火牆 (安全群組))，選取 Select existing security group (選取現有的安全群組)，接著選取您在前一個步驟中建立的安全群組。

   c. 展開進階設定區段。

      針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

(選用) 如果您使用的是多卡執行個體類型，則針對所需的每個額外網路界面，選擇新增網路界面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1 和界面類型 = EFA 搭配 ENA 或僅限 EFA。

8. 在儲存區段中，根據需求設定磁碟區。

> **Note**
>
> 您必須為 NVIDIA CUDA Toolkit 佈建額外的 10 到 20 GiB 儲存空間。如果您未佈建足夠的儲存空間，當您嘗試安裝 NVIDIA 驅動程式和 CUDA 工具組時，會收到`insufficient disk space`錯誤。

9. 在右邊的 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

步驟 3：安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN

Amazon Linux 2023

安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN

1. 為確保所有軟體套件皆為最新版本，請對您的執行個體執行快速軟體更新。

```
$ sudo dnf upgrade -y && sudo reboot
```

重新啟動執行個體後，請重新與它連線。

2. 安裝安裝 NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 所需的公用程式。

```
$ sudo dnf groupinstall 'Development Tools' -y && sudo dnf install -y dkms
 kernel-devel-$(uname -r) kernel-headers-$(uname -r)
```

3. 停用 nouveau 開放原始碼驅動程式。

   a. 針對您目前執行的核心版本，安裝需要的公用程式和核心標頭套件。

```
$ sudo yum install -y wget kernel-devel-$(uname -r) kernel-headers-$(uname -
r)
```

   b. 將 nouveau新增至`/etc/modprobe.d/blacklist.conf`拒絕清單檔案。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

c. 附加GRUB_CMDLINE_LINUX="rdblacklist=nouveau"至 grub 檔案並重新建置 GRUB 組態。

```
$ echo 'GRUB_CMDLINE_LINUX="rdblacklist=nouveau"' | sudo tee -a /etc/
default/grub \
&& sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 重新啟動執行個體並重新連線至其中。
5. 新增 CUDA 網路儲存庫。

```
$ sudo yum-config-manager --add-repo https://developer.download.nvidia.com/
compute/cuda/repos/rhel8/x86_64/cuda-rhel8.repo
```

6. 下載並安裝 NVIDIA GPU 驅動程式。

```
$ wget https://us.download.nvidia.com/tesla/580.167.08/NVIDIA-Linux-
x86_64-580.167.08.run \
&& sudo sh NVIDIA-Linux-x86_64-580.167.08.run -m kernel-open --no-drm --disable-
nouveau --dkms --silent
```

7. 安裝 NVIDIA CUDA Toolkit 和 cuDNN。

```
$ sudo dnf install -y cuda-toolkit-13-0 libcudnn9-cuda-13 libcudnn9-devel-
cuda-13
```

8. 重新啟動執行個體並重新連線至其中。
9. （使用 NVSwitch 的執行個體，例如 P 系列多 GPU 執行個體）安裝和啟動 NVIDIA Fabric Manager。G 系列執行個體不使用 NVSwitch，也不需要 Fabric Manager。

```
$ sudo dnf install -y https://developer.download.nvidia.com/compute/cuda/repos/
rhel8/x86_64/nvidia-fabricmanager-580.167.08-1.el8.x86_64.rpm \
```

```
&& sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
fabricmanager
```

10. 確認每次執行個體啟動時都已設定 CUDA 路徑。

    - 針對 bash shell，新增以下陳述式到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

      ```
      export PATH=/usr/local/cuda/bin:$PATH
      export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
      lib64:$LD_LIBRARY_PATH
      ```

    - 針對 tcsh shell，新增以下陳述式到 /home/*username*/.cshrc。

      ```
      setenv PATH=/usr/local/cuda/bin:$PATH
      setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
      lib64:$LD_LIBRARY_PATH
      ```

11. 若要確認 NVIDIA GPU 驅動程式是否正常運作，請執行下列命令。

    ```
    $ nvidia-smi -q | head
    ```

    此命令應會傳回 NVIDIA GPUs、NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 的相關資訊。

Ubuntu 26.04, Ubuntu 24.04, and Ubuntu 22.04

安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN

1. 為確保所有軟體套件皆為最新版本，請對您的執行個體執行快速軟體更新。

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

2. 安裝安裝 NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 所需的公用程式。

   ```
   $ sudo apt-get update && sudo apt-get install build-essential -y
   ```

3. 若要使用 NVIDIA GPU 驅動程式，您必須先停用nouveau開放原始碼驅動程式。

   a. 針對您目前執行的核心版本，安裝需要的公用程式和核心標頭套件。

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

b. 將 nouveau新增至/etc/modprobe.d/blacklist.conf拒絕清單檔案。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

c. 使用您偏好的文字編輯器開啟 /etc/default/grub，然後新增下列命令。

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

d. 重建 GRUB 組態。

```
$ sudo update-grub
```

4. 重新啟動執行個體並重新連線至其中。
5. 新增 CUDA 儲存庫並安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN。

- Ubuntu 26.04

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2604/
x86_64/cuda-keyring_1.1-1_all.deb \
&& sudo dpkg -i cuda-keyring_1.1-1_all.deb \
&& sudo add-apt-repository -y 'deb [trusted=yes] https://
developer.download.nvidia.com/compute/cuda/repos/ubuntu2604/x86_64 /' \
&& sudo apt-get update
```

```
$ sudo apt-get install -y nvidia-open cuda-toolkit-13-3 libcudnn9-cuda-13
 libcudnn9-dev-cuda-13
```

- Ubuntu 24.04

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/
x86_64/cuda-keyring_1.1-1_all.deb \
&& sudo dpkg -i cuda-keyring_1.1-1_all.deb \
```

```
&& sudo add-apt-repository -y 'deb [trusted=yes] https://
developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64 /' \
&& sudo apt-get update
```

```
$ sudo apt-get install -y nvidia-open-580 cuda-toolkit-13-0 libcudnn9-cuda-13
 libcudnn9-dev-cuda-13
```

- Ubuntu 22.04

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/
x86_64/cuda-keyring_1.1-1_all.deb \
&& sudo dpkg -i cuda-keyring_1.1-1_all.deb \
&& sudo DEBIAN_FRONTEND=noninteractive add-apt-repository -y "deb https://
developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/ /" \
&& sudo apt-get update
```

```
$ sudo apt-get install -y nvidia-open-580 cuda-toolkit-13-0 libcudnn9-cuda-13
 libcudnn9-dev-cuda-13
```

6. 重新啟動執行個體並重新連線至其中。
7. （使用 NVSwitch 的執行個體，例如 P 系列多 GPU 執行個體）安裝和啟動 NVIDIA Fabric Manager。G 系列執行個體不使用 NVSwitch，也不需要 Fabric Manager。

   - Ubuntu 26.04

   ```
   $ sudo apt-get install -y nvidia-fabricmanager \
   && sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
   fabricmanager
   ```

   - Ubuntu 24.04 和 Ubuntu 22.04

   ```
   $ sudo apt-get install -y nvidia-fabricmanager-580 \
   && sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
   fabricmanager
   ```

8. 確認每次執行個體啟動時都已設定 CUDA 路徑。

   - 針對 bash shell，新增以下陳述式到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export PATH=/usr/local/cuda/bin:$PATH
   ```

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

- 針對 tcsh shell，新增以下陳述式到 /home/*username*/.cshrc。

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

9. 若要確認 NVIDIA GPU 驅動程式是否正常運作，請執行下列命令。

```
$ nvidia-smi -q | head
```

此命令應會傳回 NVIDIA GPUs、NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 的相關資訊。

Debian 12

安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN

1. 安裝安裝 NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 所需的公用程式。

```
$ sudo apt-get install -y build-essential gcc make linux-headers-$(uname -r)
 dkms
```

2. 停用 nouveau 開放原始碼驅動程式。

```
$ sudo sed -i 's/GRUB_CMDLINE_LINUX=""/
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"/' /etc/default/grub && sudo update-grub
```

3. 重新啟動執行個體並重新連線至其中。
4. 新增 CUDA 儲存庫並安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN。

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/debian12/x86_64/
cuda-keyring_1.1-1_all.deb \
&& sudo dpkg -i cuda-keyring_1.1-1_all.deb \
&& sudo DEBIAN_FRONTEND=noninteractive add-apt-repository -y "deb https://
developer.download.nvidia.com/compute/cuda/repos/debian12/x86_64/ /" \
&& sudo apt-get update
```

```
$ sudo apt-get install -y nvidia-open cuda-toolkit-13-0 libcudnn9-cuda-13
 libcudnn9-dev-cuda-13
```

5. 設定 NVIDIA UVM 核心模組。

```
$ uvm_ko=$(find /lib/modules/$(uname -r) -name 'nvidia*uvm*.ko*' 2>/dev/null
 | head -1) && if [ -n "$uvm_ko" ]; then real=$(basename "$uvm_ko"); real=
${real%.ko*}; echo "$real" | sudo tee /etc/modules-load.d/nvidia-uvm.conf; if
 [ "$real" != "nvidia-uvm" ]; then echo "alias nvidia-uvm $real" | sudo tee /
etc/modprobe.d/nvidia-uvm.conf; fi; sudo modprobe "$real" || true; fi && sudo
 modprobe nvidia
```

6. 重新啟動執行個體並重新連線至其中。
7. （使用 NVSwitch 的執行個體，例如 P 系列多 GPU 執行個體）安裝和啟動 NVIDIA Fabric Manager。G 系列執行個體不使用 NVSwitch，也不需要 Fabric Manager。

   a. 判斷 NVIDIA 核心模組的版本。

   ```
   $ cat /proc/driver/nvidia/version | grep "Kernel Module"
   ```

   以下為範例輸出。

   ```
   NVRM version: NVIDIA UNIX x86_64 Kernel Module  610.43.02  ...
   ```

   在上述範例中，已安裝610核心模組的主要版本。

   b. 使用上一個步驟中識別的主要版本安裝 NVIDIA Fabric Manager。

   ```
   $ sudo apt-get install -y nvidia-fabricmanager-major_version_number \
   && sudo systemctl enable nvidia-fabricmanager && sudo systemctl start
    nvidia-fabricmanager
   ```

8. 確認每次執行個體啟動時都已設定 CUDA 路徑。

   - 針對 bash shell，新增以下陳述式到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export PATH=/usr/local/cuda/bin:$PATH
   export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
   lib64:$LD_LIBRARY_PATH
   ```

- 針對 tcsh shell，新增以下陳述式到 /home/*username*/.cshrc。

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

9. 若要確認 NVIDIA GPU 驅動程式是否正常運作，請執行下列命令。

```
$ nvidia-smi -q | head
```

此命令應會傳回 NVIDIA GPUs、NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 的相關資訊。

RHEL 10

安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN

1. 為確保所有軟體套件皆為最新版本，請對您的執行個體執行快速軟體更新。

```
$ sudo dnf upgrade -y && sudo reboot
```

重新啟動執行個體後，請重新與它連線。

2. 安裝安裝 NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 所需的公用程式。

```
$ sudo dnf groupinstall 'Development Tools' -y \
&& sudo dnf install -y dkms kernel-devel-$(uname -r) \
&& sudo dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-
latest-10.noarch.rpm
```

3. 停用 nouveau 開放原始碼驅動程式。

a. 將 nouveau新增至/etc/modprobe.d/blacklist.conf拒絕清單檔案。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
```

```
EOF
```

b. 附加GRUB_CMDLINE_LINUX="rdblacklist=nouveau"至 grub 檔案並重新建置 GRUB 組態。

```
$ echo 'GRUB_CMDLINE_LINUX="rdblacklist=nouveau"' | sudo tee -a /etc/
default/grub \
&& sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 重新啟動執行個體並重新連線至其中。
5. 新增 CUDA 儲存庫並安裝 NVIDIA GPU 驅動程式、NVIDIA CUDA Toolkit 和 cuDNN。

```
$ sudo yum-config-manager --add-repo https://developer.download.nvidia.com/
compute/cuda/repos/rhel10/x86_64/cuda-rhel10.repo \
&& sudo dnf install -y nvidia-open-580.167.08 cuda-toolkit-13-0 libcudnn9-
cuda-13 libcudnn9-devel-cuda-13
```

6. 重新啟動執行個體並重新連線至其中。
7. （使用 NVSwitch 的執行個體，例如 P 系列多 GPU 執行個體）安裝和啟動 NVIDIA Fabric Manager。G 系列執行個體不使用 NVSwitch，也不需要 Fabric Manager。

```
$ sudo dnf install -y https://developer.download.nvidia.com/compute/cuda/repos/
rhel10/x86_64/nvidia-fabricmanager-580.167.08-1.x86_64.rpm \
&& sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
fabricmanager
```

8. 確認每次執行個體啟動時都已設定 CUDA 路徑。

- 針對 bash shell，新增以下陳述式到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

- 針對 tcsh shell，新增以下陳述式到 /home/*username*/.cshrc。

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

9. 若要確認 NVIDIA GPU 驅動程式是否正常運作，請執行下列命令。

```
$ nvidia-smi -q | head
```

此命令應會傳回 NVIDIA GPUs、NVIDIA GPU 驅動程式和 NVIDIA CUDA Toolkit 的相關資訊。

步驟 4：安裝 GDRY

安裝 GDRCopy 以提高 Libfabric 的效能。如需更多關於 GDRCopy 的詳細資訊，請參閱 GDRCopy 儲存庫。

Amazon Linux 2023

若要安裝 GDRCopy

1. 安裝所需的依存項目。

```
$ sudo yum -y install dkms rpm-build make check check-devel
```

2. 下載並解壓縮 GDRCopy 套件。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.5.2.tar.gz \
&& tar xf v2.5.2.tar.gz && cd gdrcopy-2.5.2/packages
```

3. 建置 GDRCopy RPM 套件。

```
$ CUDA=/usr/local/cuda ./build-rpm-packages.sh
```

4. 安裝 GDRCopy RPM 套件。

```
$ sudo rpm -Uvh gdrcopy-kmod-2.5.2*dkms*.rpm \
&& sudo rpm -Uvh gdrcopy-2.5.2*.rpm \
&& sudo rpm -Uvh gdrcopy-devel-2.5.2*.rpm
```

Ubuntu 26.04, Ubuntu 24.04, and Ubuntu 22.04

若要安裝 GDRCopy

1. 安裝所需的依存項目。

```
$ sudo apt-get install -y build-essential devscripts debhelper fakeroot pkg-
config dkms
```

> **Note**
>
> 在 Ubuntu 22.04 上，也安裝下列其他相依性：

```
$ sudo apt-get install -y check libsubunit-dev
```

2. 下載並擷取 GDRCopy 套件，然後建置套件。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.5.2.tar.gz \
&& tar xf v2.5.2.tar.gz \
&& cd gdrcopy-2.5.2/packages \
&& CUDA=/usr/local/cuda ./build-deb-packages.sh
```

3. 安裝 GDRCopy DEB 套件。

```
$ sudo dpkg -i gdrdrv-dkms_2.5.2-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.5.2-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.5.2-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.5.2-1_amd64.*.deb
```

Debian 12

若要安裝 GDRCopy

1. 安裝所需的依存項目。

```
$ sudo apt-get install -y build-essential devscripts debhelper fakeroot pkg-
config dkms
```

2. 下載並解壓縮 GDRCopy 套件。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.5.2.tar.gz \
&& tar xf v2.5.2.tar.gz && cd gdrcopy-2.5.2/packages
```

3. 套用 Debian 版本修補程式並建置套件。

```
$ sed -i 's/(2.5.2)/(2.5.2-1)/g' debian-lib/changelog \
&& sed -i 's/(2.5.2)/(2.5.2-1)/g' debian-tests/changelog \
&& sed -i 's/(2.5.2)/(2.5.2-1)/g' dkms/debian/changelog \
&& sed -i 's/(2.5.2)/(2.5.2-1)/g' debian-meta/changelog \
&& sed -i 's/FULL_VERSION="${VERSION}"/FULL_VERSION="${VERSION}-
${DEBIAN_VERSION}"/g' build-deb-packages.sh
```

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. 安裝 GDRCopy DEB 套件。

```
$ sudo dpkg -i gdrdrv-dkms_2.5.2*.deb \
&& sudo dpkg -i libgdrapi_2.5.2*.deb \
&& sudo dpkg -i gdrcopy-tests_2.5.2*.deb \
&& sudo dpkg -i gdrcopy_2.5.2*.deb
```

RHEL 10

若要安裝 GDRCopy

1. 安裝所需的依存項目。

```
$ sudo yum -y install dkms rpm-build make check check-devel
```

2. 下載並解壓縮 GDRCopy 套件。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.5.2.tar.gz \
&& tar xf v2.5.2.tar.gz && cd gdrcopy-2.5.2/packages
```

3. 建置 GDRCopy RPM 套件。

```
$ CUDA=/usr/local/cuda ./build-rpm-packages.sh
```

4. 安裝 GDRCopy RPM 套件。

```
$ sudo rpm -Uvh gdrcopy-kmod-2.5.2*dkms*.rpm \
&& sudo rpm -Uvh gdrcopy-2.5.2*.rpm \
&& sudo rpm -Uvh gdrcopy-devel-2.5.2*.rpm
```

步驟 5：安裝 EFA 軟體

在執行個體上安裝支援 EFA 所需的具備 EFA 功能的核心、EFA 驅動程式、Libfabric、aws-ofi-nccl 外掛程式和 Open MPI 堆疊。

安裝 EFA 軟體

1. 連接至您啟動的執行個體。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
2. 下載 EFA 軟體安裝檔案。軟體安裝檔案做為壓縮的 tarball (`.tar.gz`) 檔案。若要下載最新穩定版本，請使用下列命令：

   ```
   $ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.50.0.tar.gz
   ```

   您也可以在之前的命令中以 `latest` 取代版本號碼，以取得最新版本。
3. (選用) 驗證 EFA tarball (`.tar.gz`) 檔案的真確性及完整性。

   我們建議您執行這項操作來確認軟體發布者的身分，並檢查檔案自發行以來並未遭到變更或損毀。如果您不想驗證 tarball 檔案，請略過此步驟。

   > **Note**
   >
   > 或者，如果您偏好使用 MD5 或 SHA256 檢查總和來驗證 tarball 檔案，請參閱使用檢查總和驗證 EFA 安裝程式。

   a. 下載公有 GPG 金鑰並匯入至您的 keyring。

      ```
      $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
      import aws-efa-installer.key
      ```

      命令應傳回金鑰值。請記下金鑰的值，因為下一個步驟將需要它。

   b. 驗證 GPG 金鑰的指紋。執行以下命令，並指定上一步驟中的金鑰值。

```
$ gpg --fingerprint key_value
```

命令應傳回與 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC 相同的指紋。如果指紋不相符，請勿執行 EFA 安裝指令碼，並聯絡 支援。

c. 下載簽章檔案並驗證 EFA tarball 檔案的簽章。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.50.0.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.50.0.tar.gz.sig
```

下面顯示了範例輸出。

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

如果結果包含Good signature，而指紋與上一個步驟中傳回的指紋相符，請繼續下一個步驟。如果指紋不相符，請勿執行 EFA 安裝指令碼，並聯絡 支援。

4. 從壓縮的 .tar.gz 檔案中解壓縮檔案，然後導覽至解壓縮的目錄。

```
$ tar -xf aws-efa-installer-1.50.0.tar.gz && cd aws-efa-installer
```

5. (選用) 在安裝期間驗證個別套件簽章。

從 EFA 安裝程式 1.48.0 開始，安裝程式包含 GPG 簽署的個別 RPM 和 DEB 套件。若要在安裝期間驗證每個個別套件的真實性和完整性，請使用 --check-signatures旗標。當您啟用此旗標時，安裝程式會先驗證所有套件簽章，而且只有在每個套件通過驗證時才會繼續安裝。如果任何套件驗證失敗，安裝程式會立即結束，而無需安裝任何項目。

a. 下載 GPG 公有金鑰。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
```

b. 匯出金鑰路徑。然後，在下一個步驟中，將 附加--check-signatures到安裝命令，並使用 sudo -E而不是 sudo 來保留環境變數。

```
$ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
```

在以 RPM 為基礎的系統上 (Amazon Linux 2023、RHEL、Rocky Linux 和 SUSE)，安裝程式會使用 驗證每個 RPM`rpm --checksig`。在 DEB 型系統 (Ubuntu、Debian) 上，安裝程式會使用 GPG 簽章驗證來驗證每個 DEB。如果任何套件的驗證失敗，安裝會立即中止。

> **Note**
>
> `--check-signatures` 旗標為選用。如果沒有，安裝程式不會執行個別簽章驗證。

6. 執行 EFA 軟體安裝指令碼。

> **Note**
>
> 如果您已完成上一個設定套件簽章驗證的選用步驟，請附加`--check-signatures`至安裝命令並使用 `sudo -E`，而非 `sudo`。例如：`sudo -E ./efa_installer.sh -y --mpi=openmpi5 --check-signatures`。

> **Note**
>
> 從 EFA 1.30.0 開始，預設會同時安裝 Open MPI 4.1 和 Open MPI 5。除非您需要 Open MPI 4.1，否則請僅安裝 Open MPI 5。下列命令只會安裝 Open MPI 5。如果您想要安裝 Open MPI 4.1 和 Open MPI 5，請移除 `--mpi=openmpi5`。

```
$ sudo ./efa_installer.sh -y --mpi=openmpi5
```

Libfabric 安裝在 `/opt/amazon/efa` 目錄中。aws-ofi-nccl 外掛程式安裝在 `/opt/amazon/ofi-nccl` 目錄中。Open MPI 安裝在 `/opt/amazon/openmpi` 目錄中。

7. 如果 EFA 安裝程式提示您重新啟動執行個體，請執行這項操作，然後重新連線至執行個體。否則，請登出執行個體，然後重新登入以完成安裝。
8. 確認已成功安裝 EFA 軟體。

```
$ fi_info -p efa -t FI_EP_RDM
```

此命令應該會傳回 Libfabric EFA 介面的相關資訊。

步驟 6：安裝 NCCL

安裝 NCCL。如需更多關於 NCCL 詳細資訊，請參閱 NCCL 存放庫。

若要安裝 NCCL

1. 請前往 /opt 目錄。

```
$ cd /opt
```

2. 複製官方的 NCCL 存放庫至執行個體，並導覽至本地複製的存放庫。

```
$ sudo git clone https://github.com/NVIDIA/nccl.git -b v2.30.4-1 && cd nccl
```

3. 建立並安裝 NCCL 並指定 CUDA 安裝目錄。

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda-13
```

步驟 7：安裝 NCCL Test

安裝 NCCL test。NCCL test 可讓您確認 NCCL 已正確安裝，並如預期中執行。如需更多關於 NCCL test 的詳細資訊，請參閱 nccl-test 存放庫。

若要安裝 NCCL test

1. 導覽至您的主目錄。

```
$ cd $HOME
```

2. 複製官方的 nccl-tests 存放庫至執行個體，並導覽至本地複製的存放庫。

```
$ git clone https://github.com/NVIDIA/nccl-tests.git && cd nccl-tests
```

3. 新增 Libfabric 目錄至 LD_LIBRARY_PATH 變數。

- Amazon Linux 2023

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

- Ubuntu 和 Debian

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
```

- RHEL 10

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

4. 安裝 NCCL test 並指定 MPI、NCCL 和 CUDA 安裝目錄。

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=/opt/nccl/build CUDA_HOME=/usr/
local/cuda-13
```

步驟 8：測試您的 EFA 和 NCCL 組態

執行測試以確保已對 EFA 和 NCCL 正確設定暫時執行個體。

若要測試您的 EFA 和 NCCL 組態

1. 建立主機檔案指定要執行測試的主機。下列命令建立名稱為 my-hosts 的主機檔案，包含執行個體本身的參考資料。

   IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
 "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
meta-data/local-ipv4 >> my-hosts
```

   IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-
hosts
```

2. 執行 NCCL 測試。下列命令假設您每個執行個體有 8 個 GPUs。根據您的執行個體類型調整 -n和 -N值。

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_EFA_USE_DEVICE_RDMA=1 \
-x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/lib:/
opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
--hostfile my-hosts -n 8 -N 8 \
--mca pml ^cm --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none
 \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

3. 列印 NCCL_DEBUG 記錄時，您可以確認 EFA 是積極作為 NCCL 的基礎提供者。

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Selected Provider is efa*
```

使用 p4d.24xlarge 執行個體時會顯示下列其他資訊。

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Running on P4d platform, Setting
 NCCL_TOPO_FILE environment variable to /home/ec2-user/install/plugin/share/aws-
ofi-nccl/xml/p4d-24xl-topo.xml
```

步驟 9：安裝您的機器學習應用程式

在暫時執行個體上安裝機器學習應用程式。安裝程序依特定的機器學習應用程式而異。如需在 Linux 執行個體上安裝軟體的詳細資訊，請參閱《Amazon Linux 2023 使用者指南》中的管理作業系統更新。

> **Note**
>
> 參考機器學習應用程式的文件，以取得安裝指示。

步驟 10：建立 EFA 和 NCCL 啟用的 AMI

安裝所需的軟體元件之後，您需要建立 AMI，以重複用來啟動具備 EFA 功能的執行個體。

從暫時執行個體建立 AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選取 Actions (動作)、Image (映像)、Create image (建立映像)。
4. 對於 Create image (建立映像)，執行下列動作：
   a. 對於 Image name (映像名稱)，輸入 AMI 的描述性名稱。
   b. (選用) 對於 Image description (映像描述)，輸入 AMI 的簡短描述。
   c. 選擇建立映像。
5. 在導覽窗格中，選擇 AMIs。
6. 找到您在清單中建立的 AMI。等待狀態從 `pending` 變為 `available`，再繼續進行下一個步驟。

步驟 11：終止暫時執行個體

此時，您不再需要您啟動的暫時執行個體。您可以終止執行個體，以停止產生費用。

終止暫時執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選取動作、執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇終止。

步驟 12：在叢集置放群組中啟動 EFA 和具備 NCCL 功能的執行個體

使用具備 EFA 功能的 AMI 和具備 EFA 功能的安全群組，在叢集置放群組中啟動具備 EFA 和 NCCL 功能的執行個體。

> **Note**
>
> - 在叢集置放群組中啟動具備 EFA 的執行個體並非絕對必要。不過，建議在叢集置放群組中執行具備 EFA 功能的執行個體，因為這樣會在單一可用區域的低延遲群組中啟動執行個體。
> - 要確保在擴展叢集執行個體時容量可用，您可以為叢集置放群組建立容量保留。如需詳細資訊，請參閱將容量保留與置放群組搭配使用。

New console

啟動暫時執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，`EFA-instance`。該名稱將指派作為執行個體的資源標籤 (Name=*`EFA-instance`*)。
4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，選取 My AMIs (我的 AMI)，接著選取您在前一個步驟中建立的 AMI。
5. 在 Instance type (執行個體類型) 區段中，選取 `p3dn.24xlarge` 或者 `p4d.24xlarge`。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：
   a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。
   b. 針對 Firewall (security groups) (防火牆 (安全群組))，選取 Select existing security group (選取現有的安全群組)，接著選取您在前一個步驟中建立的安全群組。
   c. 展開進階設定區段。

      針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

      (選用) 如果您使用的是多卡執行個體類型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，則針對所需的每個額外網路介面，選擇新增網路介面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1，介面類型 = EFA eith ENA 或僅限 EFA。
8. (選用) 在儲存區段中，根據需求設定磁碟區。
9. 在進階詳細資訊區段中，針對叢集置放群組，選取要在其中啟動執行個體的叢集置放群組。如果您需要建立新的叢集置放群組，請選取建立新的配置群組。
10. 在右邊的摘要面板，針對執行個體的數目，輸入要啟動的具備 EFA 功能的執行個體數量，接著選擇啟動執行個體。

Old console

在叢集置放群組中啟動具備 EFA 和 NCCL 功能的執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 Launch Instance (啟動執行個體)。
3. 在 Choose an AMI (選取 AMI) 頁面上，選取 My AMIs (我的 AMI)，找出您之前建立的 AMI，然後選取 Select (選取)。
4. 在 Choose an Instance Type (選取執行個體類型) 頁面上，選取 p3dn.24xlarge，然後選取 Next: Configure Instance Details (下一步：設定執行個體詳細資訊)。
5. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面上，執行下列操作：
   a. 對於 Number of instances (執行個體的數目)，輸入要啟動的具備 EFA 和 NCCL 功能的執行個體數。
   b. 對於 Network (網路) 和 Subnet (子網路)，選取要在其中啟動執行個體的 VPC 和子網路。
   c. 對於 Placement group (置放群組)，選取 Add instance to placement group (將執行個體新增至置放群組)。
   d. 對於 Placement group name (置放群組名稱)，選取 Add to a new placement group (新增至新的置放群組)，然後輸入置放群組的描述性名稱。然後為 Placement group strategy (置放群組策略) 選取 cluster (叢集)。
   e. 對於 EFA，選擇 Enable (啟用)。
   f. 在 Network Interfaces (網路介面) 區段中，針對裝置 eth0，選擇 New network interface (新網路介面)。您可以選擇性指定一個主要 IPv4 地址，以及一或多個次要 IPv4 地址。如果您在有相關聯 IPv6 CIDR 區塊的執行個體中啟動執行個體，您可以選擇性指定一個主要 IPv6 地址，以及一或多個次要 IPv6 地址。
   g. 選擇 Next: Add Storage (下一步：新增儲存體)。
6. 在 Add Storage (新增儲存體) 頁面上，除了 AMI 指定的磁碟區 (例如根設備磁碟區)，指定要連接到執行個體的磁碟區。然後選擇 Next: Add Tags (下一步：新增標籤)。
7. 在 Add Tags (新增標籤) 頁面上，為執行個體指定標籤 (例如使用者易記的名稱)，然後選擇 Next: Configure Security Group (下一步：設定安全群組)。
8. 在 Configure Security Group (設定安全群組) 頁面上，針對 Assign a security group (指派安全群組)，選取 Select an existing security group (選取現有安全群組)，然後選取您之前建立的安全群組。
9. 選擇 Review and Launch (檢閱和啟動)。

10. 在 Review Instance Launch (檢閱執行個體啟動) 頁面上，檢閱設定，然後選擇 Launch (啟動)，以選擇金鑰對並啟動執行個體。

步驟 13：啟用無密碼 SSH

> **Note**
>
> 預設 SSH 使用者因作業系統而異：ubuntu適用於 Ubuntu、admin適用於 Debian、ec2-user適用於 Amazon Linux 和 RHEL。

若要讓您的應用程式能夠跨越叢集中的所有執行個體執行，您必須啟用從領導節點到成員節點的無密碼 SSH 存取。領導節點是您在其中執行應用程式的執行個體。叢集中的其餘執行個體為成員節點。

在叢集的執行個體之間啟用無密碼 SSH

1. 選取叢集中的一個執行個體作為領導節點，並與它連線。
2. 在領導節點上停用 strictHostKeyChecking 並啟用 ForwardAgent。使用您偏好的文字編輯器開啟 ~/.ssh/config，然後新增下列命令。

   ```
   Host *
       ForwardAgent yes
   Host *
       StrictHostKeyChecking no
   ```

3. 產生 RSA 金鑰對。

   ```
   $ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
   ```

   在 $HOME/.ssh/ 目錄中建立金鑰對。
4. 變更領導節點上私有金鑰的許可。

   ```
   $ chmod 600 ~/.ssh/id_rsa
   chmod 600 ~/.ssh/config
   ```

5. 使用您偏好的文字編輯器開啟 ~/.ssh/id_rsa.pub 並複製該金鑰。
6. 對於叢集中的每個成員節點，執行以下操作：

   a. 連線到執行個體。

b. 使用您偏好的文字編輯器開啟 ~/.ssh/authorized_keys，然後新增您先前複製的公有金鑰。

7. 若要測試無密碼的 SSH 是否如預期運作，請連線至領導節點並執行下列命令。

```
$ ssh member_node_private_ip
```

您應該連線至成員節點，而不會提示您輸入金鑰或密碼。

開始使用適用於 Amazon EC2 上推論工作負載的 EFA 和 NIXL

NVIDIA Inference Xfer Library (NIXL) 是一種高輸送量、低延遲的通訊程式庫，專為分解的推論工作負載而設計。NIXL 可以與 EFA 和 Libfabric 搭配使用，以支援預先填充和解碼節點之間的 KV-cache 傳輸，並可在各種儲存層之間實現高效的 KV-cache 移動。如需詳細資訊，請參閱 NIXL 網站。

要求

- 僅支援 Ubuntu 24.04 和 Ubuntu 22.04 基本 AMIs。
- EFA 僅支援 NIXL 1.0.0 及更新版本。

步驟 1：準備具備 EFA 功能的安全群組

EFA 需要有安全群組來允許往返安全群組自身的所有傳入和傳出流量。下列程序會建立一個安全群組，該安全群組允許所有傳入和傳出流量，並允許來自任何 IPv4 地址的傳入 SSH 流量以進行 SSH 連線。

> **Important**
>
> 此安全群組僅供測試之用。對於生產環境，我們建議您建立傳入 SSH 規則，該規則僅允許來自連線 IP 位址的流量，例如電腦的 IP 位址或區域網路中的一系列 IP 位址。

如需其他案例，請參閱不同使用案例的安全群組規則。

建立具備 EFA 功能的安全群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)，然後選擇 Create Security Group (建立安全群組)。

3. 在 Create Security Group (建立安全群組) 視窗中，執行下列動作：

   a. 對於 Security group name (安全群組名稱)，輸入安全群組的描述性名稱，例如 `EFA-enabled security group`。

   b. (選用) 對於 Description (描述)，輸入安全群組的簡短描述。

   c. 對於 VPC，選取您打算讓具備 EFA 功能的執行個體在其中啟動的 VPC。

   d. 選擇 Create Security Group (建立安全群組)。

4. 選取您建立的安全群組，在 Details (詳細資料) 索引標籤上，複製 Security group ID (安全群組 ID)。

5. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit inbound rules (編輯傳入規則)，然後執行下列動作。

   a. 選擇新增規則。

   b. 針對類型，選擇所有流量。

   c. 若為 Source type (來源類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。

   d. 選擇新增規則。

   e. 針對 Type (類型)，選擇 SSH。

   f. 針對 Source type (來源類別)，選擇 Anywhere-IPv4(隨處 - IPv4)。

   g. 選擇儲存規則。

6. 在仍然選取安全群組的情況下，依序選取 Actions (動作) 和 Edit outbound rules (編輯傳出規則)，然後執行下列動作。

   a. 選擇新增規則。

   b. 針對類型，選擇所有流量。

   c. 若為 Destination type (目的地類型)，選擇 Custom (自訂)，然後將您複製的安全群組 ID 貼到欄位中。

   d. 選擇儲存規則。

> **Important**
>
> 自我參考傳入和傳出規則（允許進出安全群組本身的所有流量）是 EFA 運作的必要條件。如果沒有這些規則，執行個體之間的 EFA 流量將遭到封鎖。

步驟 2：啟動暫時執行個體

啟動暫時執行個體，以用來安裝和設定 EFA 軟體元件。您可以使用此執行個體來建立具備 EFA 功能的 AMI，再從中啟動具備 EFA 功能的執行個體。

啟動暫時執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，`EFA-instance`。該名稱將指派作為執行個體的資源標籤 (Name=*`EFA-instance`*)。
4. 在應用程式和作業系統映像區段中，為其中一個支援的作業系統選取 AMI。您也可以在 DLAMI 版本備註頁面上選取支援的 DLAMI。
5. 在執行個體類型區段中，選取支援的執行個體類型。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：

   a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。

   b. 針對 Firewall (security groups) (防火牆 (安全群組))，選取 Select existing security group (選取現有的安全群組)，接著選取您在前一個步驟中建立的安全群組。

   c. 展開進階設定區段。

      針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

      (選用) 如果您使用的是多卡執行個體類型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，則針對所需的每個額外網路介面，選擇新增網路介面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1，介面類型 = EFA 搭配 ENA 或僅限 EFA。
8. 在儲存區段中，根據需求設定磁碟區。

> **Note**
>
> 您必須為 Nvidia CUDA 工具組佈建額外的 10 到 20 GitB 儲存空間。如果您沒有佈建足夠的儲存空間，您會在嘗試安裝 Nvidia 驅動程式和 CUDA 工具組時，收到 `insufficient disk space` 錯誤。

9. 在右邊的 Summary (摘要) 面板中，選擇 Launch instance (啟動執行個體)。

> **Important**
>
> 如果您的 AMI 已包含 Nvidia GPU 驅動程式、CUDA 工具組和 cuDNN，或者您使用非 GPU 執行個體，請略過步驟 3。

步驟 3：安裝 Nvidia GPU 驅動程式、Nvidia CUDA 工具組和 cuDNN

安裝 Nvidia GPU 驅動程式、Nvidia CUDA 工具組和 cuDNN

1. 為確保所有軟體套件皆為最新版本，請對您的執行個體執行快速軟體更新。

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

2. 請安裝 Nvidia GPU 驅動程式和 Nvidia CUDA 工具組需要的公用程式。

```
$ sudo apt-get install build-essential -y
```

3. 若要使用 Nvidia GPU 驅動程式，您必須先停用 nouveau 開源驅動程式。

   a. 針對您目前執行的核心版本，安裝需要的公用程式和核心標頭套件。

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

   b. 新增 nouveau 至 /etc/modprobe.d/blacklist.conf 拒絕清單檔案。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   c. 使用您偏好的文字編輯器開啟 /etc/default/grub，然後新增下列命令。

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   d. 重建 Grub 組態。

```
$ sudo update-grub
```

4. 重新啟動執行個體並重新連線至其中。
5. 新增 CUDA 儲存庫並安裝 Nvidia GPU 驅動程式、NVIDIA CUDA 工具組和 cuDNN。

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/
compute/machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/repos/
ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-dkms-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535 cuda-
toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

6. 重新啟動執行個體並重新連線至其中。
7. (僅限 p4d.24xlarge 和 p5.48xlarge) 安裝 Nvidia Fabric Manager。

   a. 您安裝的 Nvidia Fabric Manager 版本必須與您在上一個步驟中安裝的 Nvidia 核心模組版本一致。

      執行以下命令來判斷 Nvidia 核心模組的版本。

      ```
      $ cat /proc/driver/nvidia/version | grep "Kernel Module"
      ```

      以下為範例輸出。

      ```
      NVRM version: NVIDIA UNIX x86_64 Kernel Module  450.42.01  Tue Jun 15 21:26:37
       UTC 2021
      ```

      在上述範例中，已安裝核心模組主要版本 450。這表示您需要安裝 Nvidia Fabric Manager 版本 450。

b. 安裝 Nvidia Fabric Manager。執行以下命令，並指定在上一步驟中使用的主要版本。

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-major_version_number
```

例如，如果已安裝核心模組的主要版本 450，請使用下列命令來安裝同樣版本的 Nvidia Fabric Manager。

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-450
```

c. 啟動服務，並確保服務會在執行個體啟動時自動啟動。NV Switch Management 需要使用 Nvidia Fabric Manager。

```
$ sudo systemctl start nvidia-fabricmanager && sudo systemctl enable nvidia-
fabricmanager
```

8. 確認每次執行個體啟動時都已設定 CUDA 路徑。

- 針對 bash shell，新增以下陳述式到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:
$LD_LIBRARY_PATH
```

- 針對 tcsh shell，新增以下陳述式到 /home/*username*/.cshrc。

```
setenv PATH=/usr/local/cuda/bin:$PATH
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:
$LD_LIBRARY_PATH
```

9. 若要確認 Nvidia GPU 驅動程式正確運作，請執行下列命令。

```
$ nvidia-smi -q | head
```

命令應該回傳 Nvidia GPU、Nvidia GPU 驅動程式和 Nvidia CUDA 工具組的資訊。

> **Important**
>
> 如果您的 AMI 已包含 GDRCopy，或者您使用非 GPU 執行個體，請略過步驟 4。

步驟 4：安裝 GDRY

安裝 GDRCopy 以改善 GPU 型平台上 Libfabric 的效能。如需更多關於 GDRCopy 的詳細資訊，請參閱 GDRCopy 儲存庫。

若要安裝 GDRCopy

1. 安裝所需的依存項目。

```
$ sudo apt -y install build-essential devscripts debhelper check libsubunit-dev
 fakeroot pkg-config dkms
```

2. 下載並解壓縮 GDRCopy 套件。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz \
&& cd gdrcopy-2.4/packages
```

3. 建置 GDRCopy DEB 套件。

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. 安裝 GDRCopy DEB 套件。

```
$ sudo dpkg -i gdrdrv-dkms_2.4-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.4-1_amd64.*.deb
```

> **Important**
>
> 如果您的 AMI 已包含最新的 EFA 安裝程式，請略過步驟 5。

步驟 5：安裝 EFA 軟體

安裝執行個體上支援 EFA 所需的已啟用 EFA 的核心、EFA 驅動程式和 Libfabric 堆疊。

安裝 EFA 軟體

1. 連接至您啟動的執行個體。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
2. 下載 EFA 軟體安裝檔案。軟體安裝檔案已封裝成壓縮 tarball (`.tar.gz`) 檔案。若要下載最新穩定版本，請使用下列命令：

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.50.0.tar.gz
```

3. 從壓縮`.tar.gz`檔案解壓縮檔案，刪除 tarball，然後導覽至解壓縮的目錄。

```
$ tar -xf aws-efa-installer-1.50.0.tar.gz && rm -rf aws-efa-installer-1.50.0.tar.gz
 && cd aws-efa-installer
```

4. (選用) 在安裝期間驗證個別套件簽章。

   從 EFA 安裝程式 1.48.0 開始，安裝程式包含 GPG 簽署的個別 RPM 和 DEB 套件。若要在安裝期間驗證每個個別套件的真實性和完整性，請使用 `--check-signatures`旗標。當您啟用此旗標時，安裝程式會先驗證所有套件簽章，而且只有在每個套件通過驗證時才會繼續安裝。如果任何套件驗證失敗，安裝程式會立即結束，而無需安裝任何項目。

   a. 下載 GPG 公有金鑰。

   ```
   $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
   ```

   b. 匯出金鑰路徑。然後，在下一個步驟中，將 附加`--check-signatures`到安裝命令，並使用 `sudo -E`而不是 `sudo` 來保留環境變數。

   ```
   $ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
   ```

   在 RPM 型系統上 (Amazon Linux、RHEL、Rocky Linux 和 SUSE)，安裝程式會使用 驗證每個 RPM`rpm --checksig`。在 DEB 型系統 (Ubuntu、Debian) 上，安裝程式會使用 GPG 簽章驗證來驗證每個 DEB。

   如果任何套件的驗證失敗，安裝會立即中止，保護您的系統免於損壞或惡意的套件。

> **Note**
>
> `--check-signatures` 旗標為選用。如果沒有，安裝程式不會執行個別簽章驗證。

5. 執行 EFA 軟體安裝指令碼。

> **Note**
>
> 如果您已完成上一個設定套件簽章驗證的選用步驟，請附加`--check-signatures`至安裝命令並使用 `sudo -E`，而非 `sudo`。例如：`sudo -E ./efa_installer.sh -y --check-signatures`。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 安裝在 `/opt/amazon/efa` 目錄中。

6. 如果 EFA 安裝程式提示您重新啟動執行個體，請執行這項操作，然後重新連線至執行個體。否則，請登出執行個體，然後重新登入以完成安裝。
7. 確認已成功安裝 EFA 軟體元件。

```
$ fi_info -p efa -t FI_EP_RDM
```

此命令應該會傳回 Libfabric EFA 介面的相關資訊。以下範例顯示命令輸出。

- `p3dn.24xlarge` 具有單一網路介面

```
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-rdm
version: 2.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

- 具有多個網路介面的 `p4d.24xlarge` 和 `p5.48xlarge`

```
provider: efa
fabric: EFA-fe80::c6e:8fff:fef6:e7ff
domain: efa_0-rdm
```

```
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c34:3eff:feb2:3c35
domain: efa_1-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c0f:7bff:fe68:a775
domain: efa_2-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::ca7:b0ff:fea6:5e99
domain: efa_3-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

步驟 6：安裝 NIXL

安裝 NIXL。如需 NIXL 的詳細資訊，請參閱 NIXL 儲存庫。

Pre-built distributions

使用 PyPI 安裝 NIXL

1. 安裝所需的依存項目。

```
$ sudo apt install pip
```

2. 安裝 NIXL。

```
$ pip install nixl
```

Build from source

從來源建置和安裝 NIXL

1. 安裝所需的依存項目。

```
$ sudo apt install cmake pkg-config meson pybind11-dev libaio-dev nvidia-cuda-
toolkit pip libhwloc-dev \
&& pip install meson ninja pybind11
```

2. 導覽至您的主目錄。

```
$ cd $HOME
```

3. 將官方 NIXL 儲存庫複製到執行個體，然後導覽至本機複製的儲存庫。

```
$ sudo git clone https://github.com/ai-dynamo/nixl.git && cd nixl
```

4. 建置並安裝 NIXL，並指定 Libfabric 安裝目錄的路徑。

```
$ sudo meson setup . nixl --prefix=/usr/local/nixl -Dlibfabric_path=/opt/amazon/
efa
$ cd nixl && sudo ninja && sudo ninja install
```

步驟 7：安裝 NIXL 基準，並測試您的 EFA 和 NIXL 組態

安裝 NIXL Benchmark 並執行測試，以確保您的暫時執行個體已針對 EFA 和 NIXL 正確設定。NIXL Benchmark 可讓您確認 NIXL 已正確安裝，並正常運作。如需詳細資訊，請參閱 nixlbench 儲存庫。

NIXL Benchmark (nixlbench) 需要 ETCD 才能協調用戶端和伺服器。若要搭配 NIXL 使用 ETCD，需要 ETCD 伺服器和用戶端，以及 ETCD CPP API。

Build from Docker

使用 Docker 安裝和測試 NIXL 基準

1. 將官方 NIXL 儲存庫複製到執行個體，然後導覽至 nixlbench 建置目錄。

```
$ git clone https://github.com/ai-dynamo/nixl.git
$ cd nixl/benchmark/nixlbench/contrib
```

2. 建置容器。

```
$ ./build.sh
```

如需 Docker 建置選項的詳細資訊，請參閱 nixlbench 儲存庫。

3. 安裝 Docker.

```
$ sudo apt install docker.io -y
```

4. 啟動 ETCD 伺服器以進行協調。

```
$ docker run -d --name etcd-server \
    -p 2379:2379 -p 2380:2380 \
    quay.io/coreos/etcd:v3.5.18 \
    /usr/local/bin/etcd \
    --data-dir=/etcd-data \
    --listen-client-urls=http://0.0.0.0:2379 \
    --advertise-client-urls=http://0.0.0.0:2379 \
    --listen-peer-urls=http://0.0.0.0:2380 \
    --initial-advertise-peer-urls=http://0.0.0.0:2380 \
    --initial-cluster=default=http://0.0.0.0:2380
```

5. 驗證 ETCD 伺服器是否正在執行。

```
$ curl -L http://localhost:2379/health
```

預期的輸出結果：

```
{"health":"true"}
```

6. 開啟執行個體的兩個終端機。在兩個終端機上執行下列命令來驗證安裝。此命令在相同的執行個體上使用 ETCD 伺服器、使用 Libfabric 做為後端，並使用 GPU 記憶體操作。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM \
    --target_seg_type VRAM
```

> **Note**
>
> 對於非 GPU 執行個體，請使用 值DRAM，而非 VRAM 。

Build from source

> **Important**
>
> 只有在您在步驟 6 中選擇從來源建置時，才遵循此索引標籤。

安裝 NIXL 基準

1. 安裝所需的系統相依性。

```
$ sudo apt install libgflags-dev
```

2. 安裝 ETCD 伺服器和用戶端。

```
$ sudo apt install -y etcd-server etcd-client
```

3. 安裝 ETCD CPP API。

   a. 安裝 ETCD CPP API 所需的相依性。

```
$ sudo apt install libboost-all-dev libssl-dev libgrpc-dev libgrpc++-dev
 libprotobuf-dev protobuf-compiler-grpc libcpprest-dev
```

   b. 複製並安裝 ETCD CPP API。

```
$ cd $HOME
$ git clone https://github.com/etcd-cpp-apiv3/etcd-cpp-apiv3.git
$ cd etcd-cpp-apiv3
$ mkdir build && cd build
$ cmake ..
$ sudo make -j$(nproc) && sudo make install
```

4. 建置並安裝 nixlbench。

```
$ sudo meson setup . $HOME/nixl/benchmark/nixlbench -Dnixl_path=/usr/local/nixl/
$ sudo ninja && sudo ninja install
```

測試您的 EFA 和 NIXL 組態

1. 在執行個體上啟動 ETCD 伺服器。

```
$ etcd --listen-client-urls "http://0.0.0.0:2379" \
    --advertise-client-urls "http://localhost:2379" &
```

2. 驗證 ETCD 伺服器是否正在執行。

```
$ curl -L http://localhost:2379/health
```

預期的輸出結果：

```
{"health":"true"}
```

3. 開啟執行個體的兩個終端機。在兩個終端機上，完成下列步驟以執行 nixlbench。

   a. 導覽至安裝 nixlbench 的目錄。

   ```
   $ cd /usr/local/nixlbench/bin/
   ```

   b. 執行測試並指定後端、ETCD 伺服器的地址和啟動器區段類型。下列命令使用相同執行個體上的 ETCD 伺服器、使用 Libfabric 做為後端，並使用 GPU 記憶體操作。環境變數會設定下列項目：

   - NIXL_LOG_LEVEL=INFO — 啟用詳細的偵錯輸出。您也可以指定 僅WARN接收錯誤訊息。
   - LD_LIBRARY_PATH — 設定 NIXL 程式庫的路徑。

   如需 NIXL Benchmark 引數的詳細資訊，請參閱官方 nixlbench 儲存庫中的 NIXLbench README。

   ```
   $ export NIXL_LOG_LEVEL=INFO
   $ export LD_LIBRARY_PATH=/usr/local/nixl/lib/$(gcc -dumpmachine):
   $LD_LIBRARY_PATH
   ```

```
$ nixlbench --etcd-endpoints 'http://localhost:2379' \
    --backend 'LIBFABRIC' \
    --initiator_seg_type 'VRAM' \
    --target_seg_type 'VRAM'
```

> **Note**
>
> 對於非 GPU 執行個體，請使用 值DRAM，而非 VRAM 。

步驟 8：安裝您的機器學習應用程式

在暫時執行個體上安裝機器學習應用程式。安裝程序依特定的機器學習應用程式而異。

> **Note**
>
> 如需安裝說明，請參閱機器學習應用程式的文件。

步驟 9：建立啟用 EFA 和 NIXL 的 AMI

安裝所需的軟體元件之後，您需要建立 AMI，以重複用來啟動具備 EFA 功能的執行個體。

從暫時執行個體建立 AMI

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選取 Actions (動作)、Image (映像)、Create image (建立映像)。
4. 對於 Create image (建立映像)，執行下列動作：
   a. 對於 Image name (映像名稱)，輸入 AMI 的描述性名稱。
   b. (選用) 對於 Image description (映像描述)，輸入 AMI 的簡短描述。
   c. 選擇建立映像。
5. 在導覽窗格中，選擇 AMIs。
6. 找到您在清單中建立的 AMI。等待狀態從`pending`變為`available`，再繼續進行下一個步驟。

步驟 10：終止暫時執行個體

此時，您不再需要您啟動的暫時執行個體。您可以終止執行個體，以停止產生費用。

終止暫時執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取您建立的暫時執行個體，然後選取動作、執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇終止。

步驟 11：啟動啟用 EFA 和 NIXL 的執行個體

使用您在步驟 9 中建立的已啟用 EFA 的 AMI，以及您在步驟 1 中建立的已啟用 EFA 的安全群組，來啟動已啟用 EFA 和 NIXL 的執行個體。

啟動啟用 EFA 和 NIXL 的執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch Instances (啟動執行個體) 來開啟新的啟動執行個體精靈。
3. (選用) 在 Name and tags (名稱和標籤) 區段中，為執行個體命名，例如，`EFA-instance`。該名稱將指派作為執行個體的資源標籤 (Name=*`EFA-instance`*)。
4. 在 Application and OS Images (應用程式和作業系統映像) 區段中，選取 My AMIs (我的 AMI)，接著選取您在前一個步驟中建立的 AMI。
5. 在執行個體類型區段中，選取支援的執行個體類型。
6. 在 Key pair (金鑰對) 區段中，選取要用於執行個體的金鑰對。
7. 在 Network settings (網路設定) 區段，選擇 Edit (編輯)，接著執行下列動作：

   a. 對於 Subnet (子網路)，請選擇要在其中啟動執行個體的子網路。若您未選取子網路，便無法啟用 EFA 的執行個體。

   b. 針對防火牆 （ 安全群組 ），選擇選取現有的安全群組，然後選取您在步驟 1 中建立的安全群組。

   c. 展開進階設定區段。

      針對網路介面 1，選取網路卡索引 = 0、裝置索引 = 0，以及介面類型 = EFA 搭配 ENA。

(選用) 如果您使用的是多卡執行個體類型，例如 p4d.24xlarge 或 p5.48xlarge，則針對所需的每個額外網路介面，選擇新增網路介面，針對網路卡索引，選擇下一個未使用的索引，然後選擇裝置索引 = 1，介面類型 = EFA 搭配 ENA 或僅限 EFA。

8. (選用) 在儲存區段中，根據需求設定磁碟區。
9. 在右邊的摘要面板，針對執行個體的數目，輸入要啟動的具備 EFA 功能的執行個體數量，接著選擇啟動執行個體。

步驟 12：啟用無密碼 SSH

若要讓您的應用程式能夠跨越叢集中的所有執行個體執行，您必須啟用從領導節點到成員節點的無密碼 SSH 存取。領導節點是您在其中執行應用程式的執行個體。叢集中的其餘執行個體為成員節點。

在叢集的執行個體之間啟用無密碼 SSH

1. 選取叢集中的一個執行個體作為領導節點，並與它連線。
2. 在領導節點上停用 strictHostKeyChecking 並啟用 ForwardAgent。使用您偏好的文字編輯器開啟 ~/.ssh/config，然後新增下列命令。

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. 產生 RSA 金鑰對。

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

在 $HOME/.ssh/ 目錄中建立金鑰對。

4. 變更領導節點上私有金鑰的許可。

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. 使用您偏好的文字編輯器開啟 ~/.ssh/id_rsa.pub 並複製該金鑰。
6. 對於叢集中的每個成員節點，執行以下操作：

   a. 連線到執行個體。

b. 使用您偏好的文字編輯器開啟 ~/.ssh/authorized_keys，然後新增您先前複製的公有金鑰。

7. 若要測試無密碼的 SSH 是否如預期運作，請連線至領導節點並執行下列命令。

```
$ ssh member_node_private_ip
```

您應該連線至成員節點，而不會提示您輸入金鑰或密碼。

> **Important**
>
> 只有在遵循步驟 7 時，才遵循步驟 13。

步驟 13：跨執行個體測試 EFA 和 NIXL 組態

執行測試，以確保您的執行個體已針對 EFA 和 NIXL 正確設定。

Build from Docker

使用 Docker 跨執行個體測試 EFA 和 NIXL 組態

1. 選取兩個主機來執行 nixlbench 基準測試。使用第一個主機的 IP 地址做為中繼資料交換的 ETCD 伺服器 IP。
2. 在主機 1 上啟動 ETCD 伺服器。

```
$ docker run -d --name etcd-server \
    -p 2379:2379 -p 2380:2380 \
    quay.io/coreos/etcd:v3.5.18 \
    /usr/local/bin/etcd \
    --data-dir=/etcd-data \
    --listen-client-urls=http://0.0.0.0:2379 \
    --advertise-client-urls=http://0.0.0.0:2379 \
    --listen-peer-urls=http://0.0.0.0:2380 \
    --initial-advertise-peer-urls=http://0.0.0.0:2380 \
    --initial-cluster=default=http://0.0.0.0:2380
```

3. 驗證 ETCD 伺服器是否正在執行。

```
$ curl -L http://localhost:2379/health
```

```
{"health":"true"}
```

4. 在主機 1 上執行 nixlbench 基準測試。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

5. 在主機 2 上執行 nixlbench 基準測試。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://ETCD_SERVER_IP:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

Build from source

> **Important**
>
> 只有在您在步驟 6 中選擇從來源建置時，才遵循此索引標籤。

跨執行個體測試 EFA 和 NIXL 組態

1. 選取兩個主機來執行 nixlbench 基準測試。使用第一個主機的 IP 地址做為中繼資料交換的 ETCD 伺服器 IP。
2. 在主機 1 上啟動 ETCD 伺服器。

```
$ etcd --listen-client-urls "http://0.0.0.0:2379" \
    --advertise-client-urls "http://localhost:2379" &
```

3. 驗證 ETCD 伺服器是否正在執行。

```
$ curl -L http://localhost:2379/health
```

```
{"health":"true"}
```

4. 在主機 1 上執行 nixlbench 基準測試。

```
$ export NIXL_LOG_LEVEL=INFO
$ export LD_LIBRARY_PATH=$HOME/nixl/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH

$ nixlbench \
    --etcd-endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

5. 在主機 2 上執行 nixlbench 基準測試。

```
$ export NIXL_LOG_LEVEL=INFO
$ export LD_LIBRARY_PATH=$HOME/nixl/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH

$ nixlbench \
    --etcd-endpoints http://ETCD_SERVER_IP:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

步驟 14：測試透過 vLLM 提供的分解推論 (選用)

安裝 NIXL 之後，您可以透過 LLM 推論使用 NIXL，並提供 vLLM、SGLang 和 TensorRT-LLM 等架構。

使用 vLLM 為您的推論工作負載提供服務

1. 安裝 vLLM。

```
$ pip install vllm
```

2. 使用 NIXL 啟動 vLLM 伺服器。下列範例命令會為 NIXL 交握連線、KV 連接器、KV 角色和傳輸後端建立一個預填充（生產者）和一個解碼（消費者）執行個體。如需詳細範例和指令碼，請參閱 NIXLConnector 使用者指南。

   若要搭配 EFA 使用 NIXL，請根據您的設定和使用案例設定環境變數。

   - 生產者（預填器）組態

```
$ vllm serve your-application \
    --port 8200 \
```

```
    --enforce-eager \
    --kv-transfer-config
 '{"kv_connector":"NixlConnector","kv_role":"kv_both","kv_buffer_device":"cuda","kv_conne
{"backends":["LIBFABRIC"]}}'
```

- 取用者（解碼器）組態

```
$ vllm serve your-application \
    --port 8200 \
    --enforce-eager \
    --kv-transfer-config
 '{"kv_connector":"NixlConnector","kv_role":"kv_both","kv_buffer_device":"cuda","kv_conne
{"backends":["LIBFABRIC"]}}'
```

上述範例組態會設定下列項目：

- kv_role 到 kv_both，這會啟用對稱功能，其中連接器可以同時做為生產者和消費者。這可為未預先決定角色差異的實驗設定和案例提供靈活性。
- kv_buffer_device 到 cuda，這可讓您使用 GPU 記憶體。
- NIXL 後端至 LIBFABRIC，可讓 NIXL 流量通過 EFA。

使用多個網路卡最大化 Amazon EC2 執行個體的網路頻寬

許多支援 EFA 的執行個體類型也具有多個網路卡。如需詳細資訊，請參閱網路卡。如果您計劃搭配其中一個執行個體類型使用 EFA，我們建議您使用下列基本組態：

- 針對主要網路介面（網路卡索引 0、裝置索引 0)，建立 ENA 介面。您不能使用僅限 EFA 的網路介面做為主要網路介面。
- 如果網路卡索引 0 支援 EFA，請為網路卡索引 0、裝置索引 建立僅限 EFA 的網路介面1。
- 對於每個額外的網路界面，請根據您的使用案例，使用下一個未使用的網路卡索引0、裝置索引、僅限 EFA 的網路界面和/或 1 ENA 網路界面的裝置索引，例如 ENA 頻寬需求或 IP 地址空間。如需使用案例範例，請參閱P5 和 P5e 執行個體的 EFA 組態。

> **Note**
>
> P5 執行個體需要以特定方式設定網路介面，才能啟用最大網路頻寬。如需詳細資訊，請參閱P5 和 P5e 執行個體的 EFA 組態。

下列範例示範如何根據這些建議啟動執行個體。

Instance launch

使用啟動執行個體精靈，在中啟動執行個體

1. 在網路設定區段中選擇編輯。
2. 展開 進階網路組態。
3. 針對主要網路介面 （ 網路卡索引 0、裝置索引 0)，建立 ENA 介面。您不能使用僅限 EFA 的網路介面做為主要網路介面。
4. 如果網路卡索引 0 支援 EFA，請為網路卡索引 0、裝置索引 建立僅限 EFA 的網路介面1。
5. 對於每個額外的網路界面，請根據您的使用案例，使用下一個未使用的網路卡索引0、裝置索引、僅限 EFA 的網路界面和/或 1 ENA 網路界面的裝置索引，例如 ENA 頻寬需求或 IP 地址空間。如需使用案例範例，請參閱P5 和 P5e 執行個體的 EFA 組態。

使用 run-instances 命令在執行個體啟動期間指定 EFAs

針對 `--network-interfaces`，指定所需的網路介面數量。對於主要網路介面，針對 `NetworkCardIndex=0` 和 `DeviceIndex=0` 指定 `InterfaceType=interface`。如果網路卡索引 0 支援 EFA，請指定 `NetworkCardIndex=0`、 `DeviceIndex=1`和 `InterfaceType=efa-only`。針對任何其他網路介面，針對 `NetworkCardIndex`指定下一個未使用的索引、`DeviceIndex=0`針對 `InterfaceType=efa-only`和/或`DeviceIndex=1`針對 `InterfaceType=interface`。

下列範例命令程式碼片段顯示具有 32 個 EFA 網路介面的請求。

```
$ aws ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami-0abcdef1234567890 \
```

```
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
 \
"NetworkCardIndex=0,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=17,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=18,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=19,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
```

```
"NetworkCardIndex=20,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=21,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=22,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=23,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=24,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=25,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=26,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=27,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=28,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=29,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=30,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=31,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only"
...
```

Launch templates

使用 Amazon EC2 主控台將 EFAs 新增至啟動範本

1. 在 Network settings (網路設定) 下，展開 Advanced network configuration (進階網路組態)。
2. 針對主要網路介面 （ 網路卡索引 0、裝置索引 0)，建立 ENA 介面。您不能使用僅限 EFA 的網路介面做為主要網路介面。
3. 如果網路卡索引 0 支援 EFA，請為網路卡索引 0、裝置索引 建立僅限 EFA 的網路介面1。
4. 對於每個額外的網路界面，請根據您的使用案例，使用下一個未使用的網路卡索引0、裝置索引、僅限 EFA 的網路界面和/或 1 ENA 網路界面的裝置索引，例如 ENA 頻寬需求或 IP 地址空間。如需使用案例範例，請參閱P5 和 P5e 執行個體的 EFA 組態。

使用 create-launch-template 命令將 EFAs 新增至啟動範本

針對 NetworkInterfaces，指定所需的網路介面數量。對於主要網路介面，針對 NetworkCardIndex=0 和 DeviceIndex=0 指定 InterfaceType=interface。如果網路卡索引 0 支援 EFA，請指定 NetworkCardIndex=0、 DeviceIndex=1和 InterfaceType=efa-only。針對任何其他網路介面，針對 NetworkCardIndex指定下一個未使用的索引、DeviceIndex=0針對 InterfaceType=efa-only和/或DeviceIndex=1針對 InterfaceType=interface。

下面的程式碼片段顯示了一個範例，其中包含可能的 32 個網路介面中的 3 個網路介面。

```
"NetworkInterfaces":[
{
  "NetworkCardIndex":0,
  "DeviceIndex":0,
  "InterfaceType": "interface",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 0,
  "DeviceIndex": 1,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 1,
  "DeviceIndex": 0,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 2,
  "DeviceIndex": 0,
```

```
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 3,
  "DeviceIndex": 0,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
}
...
```

P5 和 P5e 執行個體的 EFA 組態

p5.48xlarge 和 p5e.48xlarge執行個體支援 32 張網路卡，總網路頻寬容量為 3，200 Gbps，其中高達 800 Gbps 可用於 IP 網路流量。由於 EFA 和 IP 網路流量共用相同的基礎資源，因此一個使用的頻寬將減少另一個可用的頻寬。這表示只要總頻寬不超過 3,200 Gbps，且 IP 頻寬不超過 800 Gbps，您就可以在 EFA 流量和 IP 流量之間以任何組合分配網路頻寬。例如，如果您使用 400 Gbps 的 IP 頻寬，則可以同時達到高達 2,800 Gbps 的 EFA 頻寬。

使用案例 1：儲存 IP 位址並避免潛在的 Linux IP 問題

此組態提供高達 3,200 Gbps 的 EFA 網路頻寬，以及高達 100 Gbps 的 IP 網路頻寬，並具有一個私有 IP 位址。此組態也有助於避免潛在的 Linux IP 問題，例如不允許自動指派公有 IP 位址和 IP 路由挑戰 (主機名稱對 IP 位址映射問題和來源 IP 位址不相符)，如果執行個體具有多個網路介面，則可能會發生這種情況。

- 對於主要網路介面 （網路卡索引 0、裝置索引 0)，請使用 ENA 介面。
- 針對網路卡索引 0、裝置索引 1，建立僅限 EFA 的網路介面。
- 對於剩餘的網路介面 （網路卡索引 1-31、裝置索引 0)，請使用僅限 EFA 的網路介面。

使用案例 2：最大 EFA 和 IP 網路頻寬

此組態提供高達 3,200 Gbps 的 EFA 網路頻寬，以及高達 800 Gbps 的 IP 網路頻寬和 8 個私有 IP 位址。您無法使用此組態自動指派公有 IP 位址。但是，在啟動進行網際網路連線之後，可以將彈性 IP 位址連接至主要網路介面 (NetworkCardIndex=0、DeviceIndex=0)。

- 對於主要網路界面 （ 網路卡索引 0、裝置索引 0)，請使用 ENA 網路界面。
- 對於其餘的介面，請執行下列動作：
  - 在網路卡索引 0 裝置索引 1 上指定僅限 EFA 的網路介面，對於網路卡索引 1、2 和 3，請使用裝置索引 0。
  - 在下列每個網路卡索引子集中指定一個 ENA 網路介面和四個僅限 EFA 網路介面，並將裝置索引 1 用於 ENA 網路介面，並將裝置索引 0 用於僅限 EFA 網路介面：
    - [4、5、6、7]
    - [8、9、10、11]
    - [12、13、14、15]
    - [16、17、18、19]
    - [20、21、22、23]
    - [24、25、26、27]
    - [28、29、30、31]

下圖說明了此設定。

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami_id \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=0,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
```

```
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=17,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=18,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=19,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=20,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
```

```
"NetworkCardIndex=21,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=22,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=23,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=24,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=25,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=26,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=27,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=28,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=29,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=30,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=31,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
...
```

P6-B200 執行個體的 EFA 組態

P6-B200 執行個體的總網路頻寬容量為 3，200 Gbps，其中最高可達 1600 Gbps 可用於 ENA。其中有 8 個 GPU 和 8 個網路卡，其中每個網路卡支援高達 400 Gbps EFA 頻寬和 200 Gbps ENA 頻寬。由於 EFA 和 ENA 流量共用相同的基礎資源，因此一個使用的頻寬將減少另一個可用的頻寬。

使用案例 1：儲存 IP 位址

此組態每個執行個體至少耗用一個私有 IP 位址，並支援高達 3200 Gbps 的 EFA 頻寬和 200 Gbps 的 ENA 頻寬。

- 對於主要網路介面（網路卡索引 0、裝置索引 0)，請使用 ENA 介面。
- 針對網路卡索引 0、裝置索引 1，建立僅限 EFA 的網路介面。
- 對於其餘 7 個網路卡（網路卡索引 1-7、裝置索引 0)，請使用僅限 EFA 的網路介面。

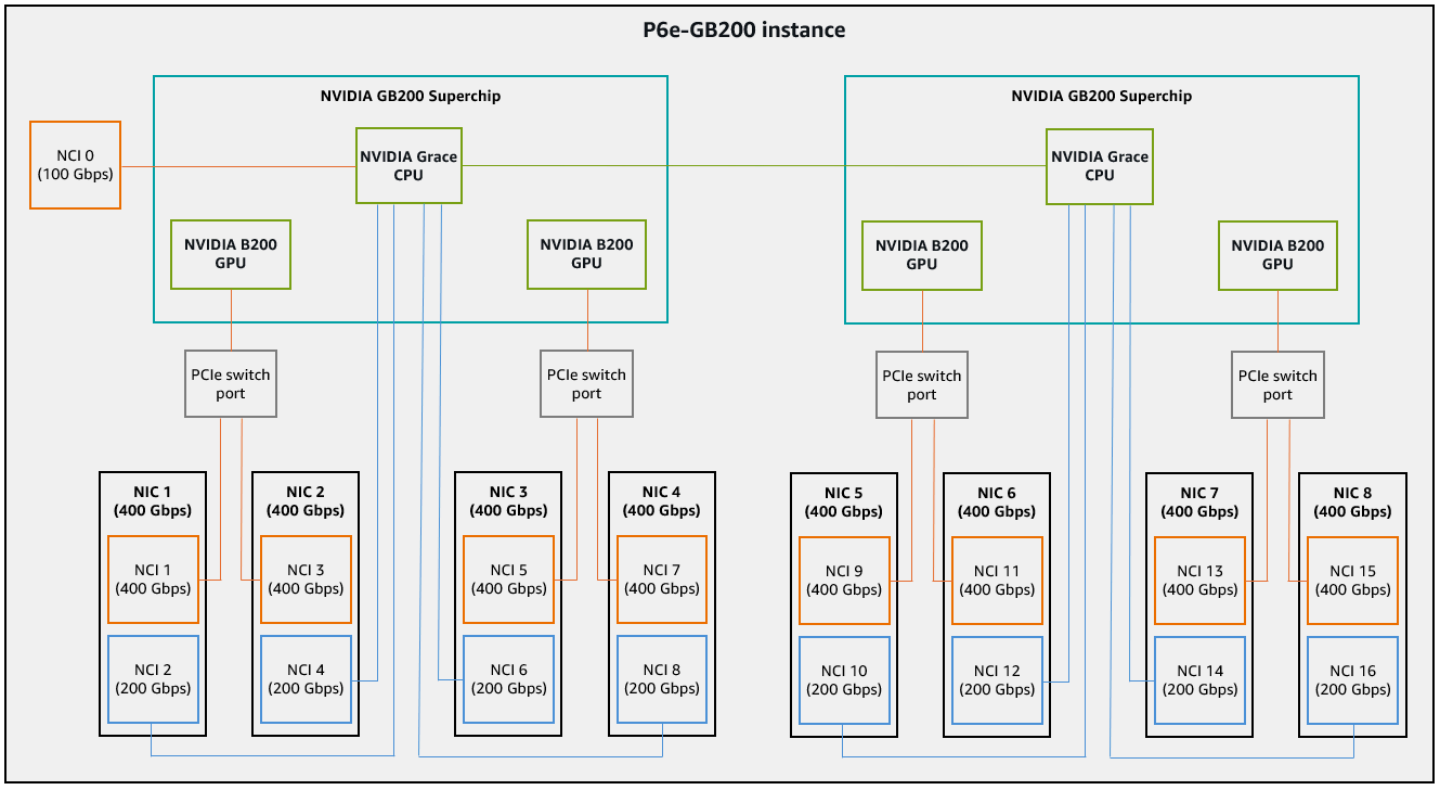
使用案例 2：最大 EFA 和 ENA 頻寬

此組態每個執行個體至少耗用 8 個私有 IP 位址，並支援高達 3200 Gbps 的 EFA 頻寬和 1600 Gbps 的 ENA 頻寬。

- 對於主要網路介面 （網路卡索引 0、裝置索引 0)，請使用 ENA 介面。
- 針對網路卡索引 0、裝置索引 1，建立僅限 EFA 的網路介面。
- 對於其餘 7 張網路卡 （網路卡索引 1-7)，請在裝置索引 0 上建立僅限 EFA 的網路介面，並在裝置索引 1 上建立 ENA 網路介面。

P6e-GB200 執行個體的 EFA 組態

P6e-GB200 執行個體最多可設定 17 個網路卡。下圖顯示 P6e-GB200 執行個體的實體網路介面卡 (NIC) 配置，以及網路卡索引 (NCI) 的映射。



主要 NCI (索引 0) 支援高達 100 Gbps 的 ENA 頻寬。具有下列索引的 NCI 僅支援 EFA 網路介面和 400 Gbps EFA 頻寬：[1, 3, 5, 7, 9, 11, 13, 15]。具有下列索引的 NCI 支援高達 200 Gbps 的 ENA 或 EFA 頻寬：[2, 4, 6, 8, 10, 12, 14, 16]。

下列群組中的 NCI 在主機上共用基礎實體 NIC：

- [1 和 2]
- [3 和 4]
- [5 和 6]
- [7 和 8]
- [9 和 10]
- [11 和 12]
- [13 和 14]
- [15 和 16]

每個實體 NIC 最多支援 400 Gbps 的頻寬。由於這些群組中的 NCI 共用相同的基礎實體 NIC，因此其中一個使用的頻寬將減少另一個可使用的頻寬。例如，如果 NCI 2 使用 200 Gbps 的 ENA 頻寬，則 NCI 1 同時可使用最多 200 Gbps 的 EFA 頻寬。

主機上的每個基礎 GPU 都可以直接透過下列 NCI 對來傳送流量：

- [1 和 3]
- [5 和 7]
- [9 和 11]
- [13 和 15]

每個 GPU 支援高達 400 Gbps 的 EFA 頻寬。由於這些群組中的網路卡共用相同的 GPU，因此其中一個使用的頻寬將減少另一個可使用的頻寬。例如，如果 NCI 1 使用 200 Gbps 的 EFA 頻寬，則 NCI 3 同時可使用最多 200 Gbps 的 EFA 頻寬。因此，為了達到最大的 EFA 效能，建議執行下列其中一項操作，以達到總共 1,600 Gbps 的 EFA 頻寬：

- 將僅限 EFA 的網路介面新增至每個群組中的一個 NIC，以達到每個網路介面 400 Gbps (4 個 EFA 網路介面 x 400 Gbps)。
- 將僅限 EFA 的網路界面新增至每個群組中的每個 NIC，以達到每個網路介面 200 Gbps (8 個 EFA 網路介面 x 200 Gbps)。

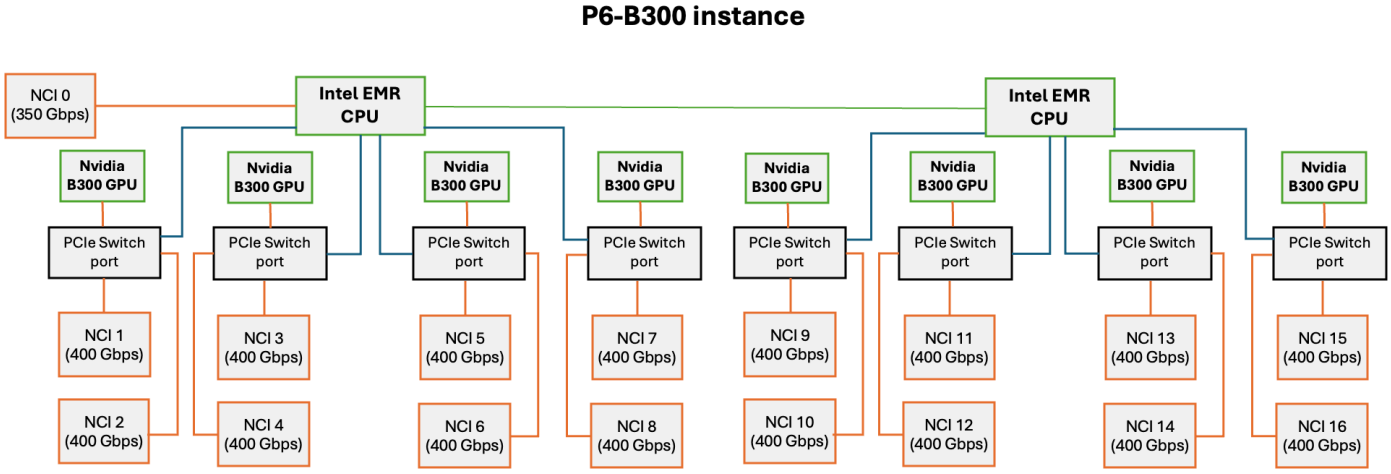
例如，下列組態在每個 NCI 群組中使用單一僅限 EFA 網路介面，提供高達 1,600 Gbps 的 EFA 頻寬，使用僅限主要 NCI (索引 0) 提供高達 100 Gbps 的 ENA 網路頻寬。

- 對於主要 NCI (網路卡索引 0，裝置索引 0)，請使用 ENA 網路介面。
- 將僅限 EFA 的網路介面新增至下列項目：

- NCI 1，裝置索引 0
- NCI 5，裝置索引 0
- NCI 9，裝置索引 0
- NCI 13，裝置索引 0

P6-B300 執行個體的 EFA 組態

P6-B300 執行個體的總網路頻寬容量高達 6400 Gbps，對於 ENA 流量則高達 3870 Gbps。它們有 8 GPUs 和 17 張網路卡，其中主要網路卡僅支援頻寬高達 350 Gbps 的 ENA 網路介面。次要網路卡支援高達 400 Gbps EFA 和高達 220 Gbps 的 ENA 頻寬。由於 EFA 和 ENA 流量共用相同的基礎資源，因此一個使用的頻寬將減少另一個可用的頻寬。



使用案例 1：儲存 IP 位址

此組態每個執行個體至少耗用一個私有 IP 地址，並支援高達 6400 Gbps 的 EFA 頻寬和高達 350 Gbps 的 ENA 頻寬。

- 對於主要網路界面 （網路卡索引 0、裝置索引 0)，請使用 ENA 網路界面。
- 對於剩餘的網路卡 （網路卡索引 1-16、裝置索引 0)，請使用僅限 EFA 的網路介面。

```
--network-interfaces \
"NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
```

```
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
```

使用案例 2：最大 EFA 和 ENA 頻寬

此組態每個執行個體至少耗用 17 個私有 IP 地址，並支援高達 6400 Gbps 的 EFA 頻寬和高達 3870 Gbps 的 ENA 頻寬。

- 對於主要網路介面（網路卡索引 0、裝置索引 0)，請使用 ENA 網路介面。
- 對於剩餘的網路卡，請建立僅限 EFA 的介面（網路卡索引 1-16 裝置索引 0) 和 ENA 介面網路卡索引 1-16 裝置索引 1)。

```
--network-interfaces \
```

```
"NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
```

```
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
```

在容器中安裝 EFA 軟體

本節說明如何在容器化環境中使用 EFA 安裝程式。您可以將命令新增至 Dockerfile 或直接在執行中容器的 shell 中執行，以安裝 EFA 軟體。

使用 shell 命令在容器中安裝 EFA 軟體

1. 更新作業系統套件。此步驟可確保您有最新的套件。

   - 針對 Debian 作業系統：

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

   - 對於以 RHEL 為基礎的作業系統：

     ```
     yum update -y
     ```

2. 下載 EFA 安裝程式。我們建議您使用最新的 EFA 安裝程式。如需最新版本的資訊，請參閱 Elastic Fabric Adapter 版本備註。

```
mkdir /tmp/efa \
&& cd /tmp/efa \
&& curl -O https://efa-installer.amazonaws.com/aws-efa-installer-version.tar.gz
```

（選用）驗證 EFA 安裝程式套件的簽章。

```
wget https://efa-installer.amazonaws.com/aws-efa-installer.key \
&& gpg --import aws-efa-installer.key \
&& cat aws-efa-installer.key | gpg --fingerprint \
&& wget https://efa-installer.amazonaws.com/aws-efa-installer-version.tar.gz.sig \
&& gpg --verify ./aws-efa-installer-version.tar.gz.sig
```

3. 安裝 EFA 軟體。下列命令使用這些引數：

   - `--skip-kmod` — 略過核心模組安裝。容器環境無法修改核心模組。
   - `--skip-limit-conf` — 略過限制組態。您無法在容器`ulimit`內設定 。改為將此設定為容器建立的一部分。
   - `--no-verify` — 略過 EFA 裝置偵測。裝置偵測使用 `lspci`，無法在容器化環境中運作。

```
tar -xf aws-efa-installer-version.tar.gz \
&& cd aws-efa-installer \
&& ./efa_installer.sh -y --skip-kmod --skip-limit-conf --no-verify
```

> **Note**
>
> 對於 NGC 容器，使用 EFA 安裝程式 1.49.0 版或更新版本，您可以使用 `--enable-ngc`旗標，而不是上述三個旗標。此旗標會自動處理所有容器特定的組態：
>
> ```
> ./efa_installer.sh -y --enable-ngc
> ```

4. 清除暫時安裝檔案。

```
rm -rf /tmp/efa
```

建立 Elastic Fabric Adapter 並將其連接至 Amazon EC2 執行個體

您可以建立 EFA 並將其連接到 Amazon EC2 執行個體，就像 Amazon EC2 中的任何其他彈性網路介面一樣。不過，與彈性網路介面不同，EFA 不能與處於 `running` 狀態的執行個體連接或分離。

考量事項

- 您可以變更與 EFA 相關聯的安全群組。若要啟用 OS-bypass 功能，EFA 必須是安全群組的成員，而該安全群組允許往返於其本身的所有傳入和傳出流量。如需詳細資訊，請參閱步驟 1：準備具備 EFA 功能的安全群組。

  就像變更與彈性網絡介面相關聯的安全群組，您以同樣的方式變更與 EFA 相關聯的安全群組。如需詳細資訊，請參閱the section called “修改網路介面屬性”。
- 將彈性 IP (IPv4) 和 IPv6 地址指派給 EFA (EFA 搭配 ENA) 的方式，就像將 IP 位址指派給彈性網絡介面一樣。如需詳細資訊，請參閱管理 IP 位址。

  您無法將 IP 位址指派給僅限 EFA 的網路介面。

任務

- 建立 EFA
- 將 EFA 連接至已停止的執行個體
- 啟動執行個體時連接 EFA
- 將 EFA 新增至啟動範本

建立 EFA

您可以在 VPC 的子網路中建立 EFA。EFA 建立後即無法移至另一個子網路，而且只能連接到同一可用區域中已停止的執行個體。

Console

若要建立 EFA (EFA 搭配 ENA 或僅限 ENA) 網路介面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Network Interfaces (網路介面)，然後選擇 Create network interface (建立網路介面)。
3. 針對 Description (描述)，輸入 EFA 的描述性名稱。

4. 針對 Subnet (子網路)，選取要在其中建立 EFA 的子網路。
5. 針對介面類型，選擇以下其中一項：
   - EFA 搭配 ENA：建立同時支援 ENA 和 EFA 裝置的網路介面。
   - 僅限 EFA：僅使用 EFA 裝置建立網路介面。
6. (僅限 EFA 搭配 ENA) 設定網路介面的 IP 位址和字首指派。您可以指派的 IP 位址和字首類型取決於選取的子網路。對於僅限 IPv4 的子網路，只能指派 IPv4 IP 位址和字首。對於僅限 IPv6 的子網路，只能指派 IPv6 IP 位址和字首。對於雙堆疊子網路，可以同時指派 IPv4 和 IPv6 IP 位址和字首。

   > **Note**
   >
   > 您無法將 IP 位址指派給僅限 EFA 的網路介面。

   a. 針對私有 IPv4 位址和/或 IPv6 位址，選擇自動指派，讓 Amazon EC2 從選取的子網路中自動指派 IP 位址，或選擇自訂以手動指定要指派的 IP 位址。

   b. 如果指派 IPv6 位址，可以選擇啟用指派主要 IPv6 IP。這樣做會將主要 IPv6 全域單播位址 (GUA) 指派給網路介面。指派主要 IPv6 地址可讓您避免中斷執行個體或 ENI 的流量。如需詳細資訊，請參閱 IPv6 位址。

   c. 對於 IPv4 字首委派和/或 IPv6 字首委派，選擇自動指派，讓 Amazon EC2 從子網路的 CIDR 區塊中自動指派字首，或選擇自訂以從子網路的 CIDR 區塊中手動指定字首。如果您指定字首， 會 AWS 驗證它尚未指派給另一個資源。如需詳細資訊，請參閱Amazon EC2 網路介面的字首委派

   d. (選用) 設定閒置連線追蹤逾時設定。如需詳細資訊，請參閱閒置連線追蹤逾時

      - TCP 建立逾時：處於已建立狀態的閒置 TCP 連線的逾時時間 (以秒為單位)。最小值：60秒。最大值：432000秒 (5天)。預設：432000秒。建議使用：432000秒以下。
      - UDP 逾時：只出現單一方向或單一請求回應交易流量的閒置 UDP 流量的逾時時間 (以秒為單位)。最小值：30秒。最大值：60秒。預設：30秒。
      - UDP 串流逾時：分類為出現多個請求回應交易串流的閒置 UDP 流量的逾時時間 (以秒為單位)。最小值：60秒。最大值：180秒 (3分鐘)。預設：180秒。
7. 針對 Security groups (安全群組)，選取一個或多個安全群組。
8. 選擇 Create network interface (建立網路介面)。

AWS CLI

若要建立 EFA

使用 create-global-network 命令。對於 `--interface-type`，請為 EFA 網路介面指定 `efa`，或為僅限 EFA 網路介面指定 `efa-only`。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --interface-type efa \
    --description "my efa"
```

PowerShell

若要建立 EFA

使用 New-EC2NetworkInterface cmdlet。對於 `-InterfaceType`，請為 EFA 網路介面指定 `efa`，或為僅限 EFA 網路介面指定 `efa-only`

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -InterfaceType efa `
    -Description "my efa"
```

將 EFA 連接至已停止的執行個體

您可以將 EFA 連接至任何處於 `stopped` 狀態的受支援執行個體。您無法將 EFA 連接至處於 `running` 狀態的執行個體。如需支援的執行個體類型的詳細資訊，請參閱支援的執行個體類型。

就像將網路介面連接至執行個體，您以同樣的方式將 EFA 連接至執行個體。如需詳細資訊，請參閱 連接網路介面。

啟動執行個體時連接 EFA

AWS CLI

啟動執行個體時連接現有的 EFA

請使用 `--network-interfaces` 選項並搭配 run-instances 命令。對於主網路介面，指定 EFA 網路接口和 `NetworkCardIndex=0`，`DeviceIndex=0`。若要連接多個 EFA 網路介面，請參閱最大化網路頻寬。

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    NetworkInterfaceId=eni-1234567890abcdef0, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

啟動執行個體時連接新的 EFA

請使用 `--network-interfaces` 選項並搭配 run-instances 命令。對於主要網路介面，針對 `NetworkCardIndex=0` 和 `DeviceIndex=0` 指定 `InterfaceType=efa`。如果您要連接多個 EFA 網路介面，請參閱最大化網路頻寬。

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    InterfaceType=efa, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

PowerShell

啟動執行個體時連接現有的 EFA

使用 New-EC2Instance cmdlet 搭配 `-NetworkInterfaces` 參數。

```
-NetworkInterface $networkInterface
```

定義網路介面，如下所示。

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.NetworkInterfaceId = "eni-1234567890abcdef0"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

啟動執行個體時連接新的 EFA

使用 New-EC2Instance cmdlet 搭配 `-NetworkInterfaces` 參數。

```
-NetworkInterface $networkInterface
```

定義網路介面，如下所示。

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.InterfaceType = "efa"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

將 EFA 新增至啟動範本

您可以建立啟動範本，其中包含啟動具備 EFA 功能的執行個體所需的組態資訊。您可以在啟動範本中指定 EFA 和僅限 EFA 的網路介面。若要建立具備 EFA 功能的啟動範本，請建立新的啟動範本，並指定支援的執行個體類型、具備 EFA 功能的 AMI，以及具備 EFA 功能的安全群組。針對 `NetworkInterfaces`，指定要連接的 EFA 網路介面。對於主要網路介面，針對 `NetworkCardIndex=0` 和 `DeviceIndex=0` 指定 `InterfaceType=efa`。如果您要連接多個 EFA 網路介面，請參閱使用多個網路卡最大化 Amazon EC2 執行個體的網路頻寬。

您可以利用啟動範本，透過其他 AWS 服務啟動已啟用 EFA 的執行個體，例如 AWS Batch或 AWS ParallelCluster。

如需建立啟動範本的詳細資訊，請參閱建立 Amazon EC2 啟動範本。

從 Amazon EC2 執行個體分離和刪除 EFA

您可以將 EFA 從 Amazon EC2 執行個體分離，並以與 Amazon EC2 中任何其他彈性網路介面相同的方式刪除。

分離 EFA

若要從執行個體分離 EFA，您必須先停止執行個體。您無法從處於執行中狀態的執行個體分離 EFA。

就像從執行個體分離彈性網絡介面，您以同樣的方式從執行個體分離 EFA。如需詳細資訊，請參閱分離網路介面。

刪除 EFA

若要刪除 EFA，您必須先從執行個體分離它。您無法刪除仍連接至執行個體的 EFA。

就像刪除彈性網絡介面一樣，您以同樣的方式刪除 EFAs。如需詳細資訊，請參閱刪除網路介面。

在 Amazon EC2 上監控彈性布料轉接器

您可以使用下列功能來監控 Elastic Fabric Adapter 的效能。

主題

- Amazon EC2 執行個體的 EFA 驅動程式指標
- Amazon VPC 流程日誌
- Amazon CloudWatch

Amazon EC2 執行個體的 EFA 驅動程式指標

Elastic Fabric Adapter (EFA) 驅動程式會從已連接 EFA 介面的執行個體中實時發佈多個指標。可以使用這些指標來進行執行個體效能和聯網問題的故障診對、為工作負載選擇正確的叢集大小、主動計劃擴展活動，以及基準化分析應用程式，以判斷其是否能夠最大化執行個體上的可用效能。

主題

- 可用的 EFA 驅動程式指標
- 擷取執行個體的 EFA 驅動程式指標

可用的 EFA 驅動程式指標

ENA 驅動程式會即時將下列指標交付至執行個體。其提供從執行個體啟動或上次驅動程式重設以來，所連接之 EFA 裝置傳送、接收、傳輸或捨棄的累計錯誤、連線事件和封包或位元組數量。

指標	說明	支援的執行個體類型
`tx_bytes`	已傳輸的位元組數目。 單位：位元組	支援 EFA 的所有執行個體類型
`rx_bytes`	接收的位元組數量。 單位：位元組	支援 EFA 的所有執行個體類型
`tx_pkts`	代理程式傳輸的封包數目。	支援 EFA 的所有執行個體類型

指標	說明	支援的執行個體類型
	單位：計數	
rx_pkts	接收的封包數量。 單位：計數	支援 EFA 的所有執行個體類型
rx_drops	收到然後捨棄的封包數量。 單位：計數	支援 EFA 的所有執行個體類型
send_bytes	使用傳送操作傳送的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
recv_bytes	傳送操作收到的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
send_wrs	使用傳送操作傳送的封包數量。 單位：計數	支援 EFA 的所有執行個體類型
recv_wrs	傳送操作收到的封包數量。 單位：計數	支援 EFA 的所有執行個體類型
rdma_write_wrs	已完成的 rdma 寫入操作數目。 單位：計數	支援 EFA 的所有執行個體類型
rdma_read_wrs	已完成的 rdma 讀取操作數目。 單位：計數	支援 EFA 的所有執行個體類型

指標	說明	支援的執行個體類型
rdma_write_bytes	其他執行個體使用 rdma 寫入操作寫入其中的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
rdma_read_bytes	使用 rdma 讀取操作收到的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
rdma_write_wr_err	發生本機或遠端錯誤的 rdma 寫入操作數目。 單位：計數	支援 EFA 的所有執行個體類型
rdma_read_wr_err	發生本機或遠端錯誤的 rdma 讀取操作數目。 單位：計數	支援 EFA 的所有執行個體類型
rdma_read_resp_bytes	回應 rdma 讀取操作而傳送的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
rdma_write_recv_bytes	rdma 寫入操作收到的位元組數。 單位：位元組	支援 EFA 的所有執行個體類型
retrans_bytes	重新傳輸的 EFA SRD 位元組數。 單位：計數	支援 EFA 的 Nitro v4 和更新版本的執行個體類型
retrans_pkts	重新傳輸的 EFA SRD 封包數目。 單位：位元組	支援 EFA 的 Nitro v4 和更新版本的執行個體類型

指標	說明	支援的執行個體類型
retrans_timeout_events	EFA SRD 流量逾時並導致網路路徑變更的次數。 單位：計數	支援 EFA 的 Nitro v4 和更新版本的執行個體類型
impaired_remote_conn_events	EFA SRD 連線進入受損狀態的次數，導致輸送量速率限制降低。 單位：計數	支援 EFA 的 Nitro v4 和更新版本的執行個體類型
unresponsive_remote_events	EFA SRD 遠端連線沒有回應的次數。 單位：計數	支援 EFA 的 Nitro v4 和更新版本的執行個體類型

如需有關支援 EFA 的執行個體類型詳細資訊，請參閱 支援的執行個體類型。

擷取執行個體的 EFA 驅動程式指標

您可以使用 rdma-tool 命令列工具來擷取連接至執行個體之所有 EFA 介面的指標，如下所示：

```
$ rdma -p statistic show
link rdmap0s31/1
    tx_bytes 0
    tx_pkts 0
    rx_bytes 0
    rx_pkts 0
    rx_drops 0
    send_bytes 0
    send_wrs 0
    recv_bytes 0
    recv_wrs 0
    rdma_read_wrs 0
    rdma_read_bytes 0
    rdma_read_wr_err 0
    rdma_read_resp_bytes 0
    rdma_write_wrs 0
    rdma_write_bytes 0
```

```
    rdma_write_wr_err 0
    retrans_bytes 0
    retrans_pkts 0
    retrans_timeout_events 0
    unresponsive_remote_events 0
    impaired_remote_conn_events 0
```

或者，您可以使用下列命令，從 sys 檔案擷取連接至執行個體之每個 EFA 介面的指標。

```
$ more /sys/class/infiniband/device_number/ports/port_number/hw_counters/* | cat
```

例如

```
$ more /sys/class/infiniband/rdmap0s31/ports/1/hw_counters/* | cat
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/lifespan
::::::::::::::
12
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_resp_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_recv_bytes
::::::::::::::
0
::::::::::::::
```

```
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_drops
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_pkts
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_pkts
::::::::::::::
0
::::::::::::::
```

```
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_bytes
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_pkts
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_timeout_events
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/unresponsive_remote_events
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/impaired_remote_conn_events
::::::::::::::
0
```

Amazon VPC 流程日誌

您可以建立 Amazon VPC 流程日誌，以擷取進出 EFA 的流量的相關資訊。現在流程日誌資料可發佈至 Amazon CloudWatch Logs 與 Amazon S3。建立流量日誌之後，您可以在選擇的目的地中擷取及檢視其資料。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的 VPC 流程日誌。

就像為彈性網絡介面建立流程日誌，您以同樣的方式為 EFA 建立流程日誌。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的建立流量日誌。

在流程日誌項目中，EFA 流量是以 `srcAddress` 和 `destAddress` 來識別，兩者的格式都是 MAC 地址，如下列範例所示。

```
version accountId  eniId        srcAddress        destAddress       sourcePort destPort
 protocol packets bytes start      end        action log-status
2       3794735123 eni-10000001 01:23:45:67:89:ab 05:23:45:67:89:ab -          -
 -        9       5689  1521232534 1524512343 ACCEPT OK
```

Amazon CloudWatch

如果您在 Amazon EKS 叢集中使用 EFA，您可以使用 CloudWatch Container Insights 監控 EFAs。Amazon CloudWatch Container Insights 支援所有 EFA 驅動程式指標，但 `retrans_bytes`、`retrans_pkts`、`retrans_timeout_events`、`unresponsive_remote_events` 和 `impaired_remote_conn_events` 除外。

如需詳細資訊，請參閱「Amazon CloudWatch 使用者指南」中的 Amazon ECS Container Insights 指標。

使用檢查總和驗證 EFA 安裝程式

可以選擇使用 MD5 或 SHA256 檢查總和來驗證 EFA tarball (`.tar.gz` 檔案)。我們建議您執行這項操作來確認軟體發行者的身分識別，並檢查應用程式自發行以來並未遭到變更或損毀。

驗證 tarball 的步驟

使用 md5sum 公用程式做為 MD5 檢查總和，或使用 Sha256sum 公用程式做為 SHA256 檢查總和，並指定 tarball 檔案名稱。您必須從儲存 tarball 檔案的目錄執行命令。

- MD5

```
$  md5sum tarball_filename.tar.gz
```

- SHA256

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

這些命令應該會傳回下列格式的檢查總和值。

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

將命令傳回的檢查總和值與下表所提供的檢查總和值進行比較。如果檢查總和相符，則可以安全地執行安裝指令碼。如果檢查總和不相符，請勿執行安裝指令碼，然後聯絡 支援。

例如，下列指令會使用 SHA256 檢查總和來驗證 EFA 1.9.4 tarball。

```
$  sha256sum aws-efa-installer-1.9.4.tar.gz
```

輸出範例：

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

下表列出 EFA 最新版本的檢查總和。

版本	檢查總和
EFA 1.50.0	MD5: e5a5178944b1f1112f3b2eb3b15ca5a7 SHA256：fa6dff8593d866866c13cb4640d9059835cd4efa427971f100ab40c97bef2841
EFA 1.49.0	MD5: bb10de6856e08a7e7e8a6b52bea6f681 SHA256：cf2e9281a2328a243c76f911a490faed43ca0fecfe4733c25e34b2e92a32c309
EFA 1.48.0	MD5: d30960f1876fb16ac8238193a26c6b29 SHA256：f3a8aadc2b94288694a9b9375bcad84a82f0f843e0eefafda7b158b2200549cc
EFA 1.47.0	MD5: c81d4caf24dabc04a6e4818590620f5f SHA256：2df4201e046833c7dc8160907bee7f52b76ff80ed147376a2d0ed8a0dd66b2db
EFA 1.46.0	MD5：a88bbd9b71624d7ca401b54bc2fc0c19 SHA256：8302bd7849afb95c903a875d7dcb6f85b3d7629e9a8b67d020031cfc6f4d0ee1
EFA 1.45.1	MD5：91c3c87e16bbcaca1513252c38b771bb

版本	檢查總和
	SHA256：9aeb20c645135b6039 cc08986d8f14e63280f7839e882 a74df5e83627ffeaa17
EFA 1.45.0	MD5：800aeddfa9d9c5f139a7b8f7c4fec627 SHA256：25ba26a0877fe33173 90dc126aad2f23e27fc461cf0b9 40004f032cb342fa539
EFA 1.44.0	MD5:d024f6bebe080db42745103b84ca7c43 SHA256：f129a5b44a49d593d2 47e55a59eb9bcb57121566e1c2e 42b832a4e794fa83d8a
EFA 1.43.3	MD5: 4dbc6eeecc516760253c10cbedb 6319d SHA256：6c470ebce254c71653 47b5048895ac2996c8856727164 2297f4c597738300652
EFA 1.43.2	MD5: 7287b25a07c9747c0d4001e8fc5 f59b2 SHA256：de15c5bdbc83b952af bde876110830c604ad0796680e5 157c05f7c1979a41069
EFA 1.43.1	MD5: 7cfafc8debaea51dd4966fa0b2b ba673 SHA256:54211eda0c193138ee8ed09b5fb 41c41fc76fe0a77935fa4ec8d989466342740

版本	檢查總和
EFA 1.43.0	MD5: f2b3dd7dc8670b541f7c23fd58e5e503 SHA256：786df3458c499237be33bb8e50ffd4da7c18c20e254380ffc80fb90833d8cc73
EFA 1.42.0	MD5: 94b2b1db09da1dde08ec049db1f24370 SHA256：4114fe612905ee05083ae5cb391a00a012510f3abfecc642d86c9a5ae4be9008
EFA 1.41.0	MD5: 086181c3ee3f8da512fc6e1c795e8936 SHA256：3506354cdfbe31ff552fe75f5d0d9bb7efd29cf79bd99457347d29c751c38f9f
EFA 1.40.0	MD5: f3ec6f73fbeaccba082327507581157c SHA256：30491b0fe7c3470d4439594538855c981b05fa69862d74f8c05eb9b97912368a
EFA 1.39.0	MD5: c223d5954a85a7fbcd248c942b866e43 SHA256：2cbc028c03064633bb990782b47c36156637769e2f48704417a9c700a7a32101

版本	檢查總和
EFA 1.38.1	MD5: f112569e828ab65187777f794bab542c SHA256：83923374afd388b1cfcf4b3a21a2b1ba7cf46a01a587f7b519b8386cb95e4f81
EFA 1.38.0	MD5: 43a2a446b33a2506f40853d55059f1ea SHA256：4f436954f35ad53754b4d005fd8d0be63de3b4184de41a695b504bdce0fecb22
EFA 1.37.0	MD5: 6328070192bae920eca45797ad4c1db1 SHA256：2584fc3c8bb99f29b3285e275747ff09d67c18e162c2a652e36c976b72154bfb
EFA 1.36.0	MD5: 1bec83180fbffb23452ab6469ca21dfa SHA256：de183f333cfb58aeb7908a67bf9106985ba3ccb7f8638b851d2a0d8dbfacaec4
EFA 1.35.0	MD5: 252f03c978dca5f8e8d9f34e488b256e SHA256：432b6ad4368ba0cd8b902729d14a908a97be7a3dcc5239422ea994a47f35a5e1

版本	檢查總和
EFA 1.34.0	MD5: 5cd4b28d27a31677c16139b54c9acb45 SHA256：bd68839e741b0afd3ec2e37d50603803cfa7a279c120f0a736cc57c2ff2d7fdc
EFA 1.33.0	MD5: e2f61fccbcaa11e2ccfddd3660522276 SHA256：0372877b87c6a7337bb7791d255e1053b907d030489fb2c3732ba70069185fce
EFA 1.32.0	MD5: db8d65cc028d8d08b5a9f2d88881c1b1 SHA256：5f7233760be57f6fee6de8c09acbfbf59238de848e06048dc54d156ef578fc66
EFA 1.31.0	MD5: 856352f12bef2ccbadcd75e35aa52aaf SHA256：943325bd37902a4300ac9e5715163537d56ecb4e7b87b37827c3e547aa1897bf
EFA 1.30.0	MD5: 31f48e1a47fe93ede8ebd273fb747358 SHA256：876ab9403e07a0c3c91a1a34685a52eced890ae052df94857f6081c5f6c78a0a

版本	檢查總和
EFA 1.29.1	MD5: e1872ca815d752c1d7c2b5c175e52a16 SHA256：178b263b8c25845b63dc93b25bcdff5870df5204ec509af26f43e8d283488744
EFA 1.29.0	MD5: 39d06a002154d94cd982ed348133f385 SHA256：836655f87015547e733e7d9f7c760e4e24697f8bbc261bb5f3560abd4206bc36
EFA 1.28.0	MD5: 9dc13b7446665822605e66febe074035 SHA256：2e625d2d6d3e073b5178e8e861891273d896b66d03cb1a32244fd56789f1c435
EFA 1.27.0	MD5: 98bfb515ea3e8d93f554020f3837fa15 SHA256：1d49a97b0bf8d964d91652a79ac851f2550e33a5bf9d0cf86ec9357ff6579aa3
EFA 1.26.1	MD5: 884e74671fdef4725501f7cd2d451d0c SHA256：c616994c924f54ebfabfab32b7fe8ac56947fae00a0ff453d975e298d174fc96

版本	檢查總和
EFA 1.26.0	MD5: f8839f12ff2e3b9ba09ae8a82b30e663 SHA256：bc1abc1f76e97d204d3755d2a9ca307fc423e51c63141f798c2f15be3715aa11
EFA 1.25.1	MD5: 6d876b894547847a45bb8854d4431f18 SHA256：d2abc553d22b89a4ce92882052c1fa6de450d3a801fe005da718b7d4b9602b06
EFA 1.25.0	MD5: 1993836ca749596051da04694ea0d00c SHA256：98b7b26ce031a2d6a93de2297cc71b03af647194866369ca53b60d82d45ad342
EFA 1.24.1	MD5: 211b249f39d53086f3cb0c07665f4e6f SHA256：120cfeec233af0955623ac7133b674143329f9561a9a8193e473060f596aec62
EFA 1.24.0	MD5: 7afe0187951e2dd2c9cc4b572e62f924 SHA256：878623f819a0d9099d76ecd41cf4f569d4c3aac0c9bb7ba9536347c50b6bf88e

版本	檢查總和
EFA 1.23.1	MD5: 22491e114b6ee7160a8290145dca0c28 SHA256：5ca848d8e0ff4d1571cd443c36f8d27c8cdf2a0c97e9068ebf000c303fc40797
EFA 1.23.0	MD5: 38a6d7c1861f5038dba4e441ca7683ca SHA256：555d497a60f22e3857fdeb3dfc53aa86d05926023c68c916d15d2dc3df6525bd
EFA 1.22.1	MD5: 600c0ad7cdbc06e8e846cb763f92901b SHA256：f90f3d5f59c031b9a964466b5401e86fd0429272408f6c207c3f9048254e9665
EFA 1.22.0	MD5: 8f100c93dc8ab519c2aeb5dab89e98f8 SHA256：f329e7d54a86a03ea51da6ea9a5b68fb354fbae4a57a02f9592e21fce431dc3a
EFA 1.21.0	MD5: 959ccc3a4347461909ec02ed3ba7c372 SHA256：c64e6ca34ccfc3ebe8e82d08899ae8442b3ef552541cf5429c43d11a04333050

版本	檢查總和
EFA 1.20.0	MD5: 7ebfbb8e85f1b94709df4ab3db47913b SHA256：aeefd2681ffd5c4c631d1502867db5b831621d6eb85b61fe3ec80df983d1dcf0
EFA 1.19.0	MD5: 2fd45324953347ec5518da7e3fefa0ec SHA256：99b77821b9e72c8dea015cc92c96193e8db307deee05b91a58094cc331f16709
EFA 1.18.0	MD5: fc2571a72f5d3c7b7b576ce2de38d91e SHA256：acb18a0808aedb9a5e485f1469225b9ac97f21db9af78e4cd6939700debe1cb6
EFA 1.17.3	MD5: 0517df4a190356ab559235147174cafd SHA256：5130998b0d2883bbae189b21ab215ecbc1b01ae0231659a9b4a17b0a33ebc6ca
EFA 1.17.2	MD5: a329dedab53c4832df218a24449f4c9a SHA256：bca1fdde8b32b00346e175e597ffab32a09a08ee9ab136875fb38283cc4cd099

版本	檢查總和
EFA 1.17.1	MD5: 733ae2cfc9d14b52017eaf0a2ab6b0ff SHA256：f29322640a88ae9279805993cb836276ea240623820848463ca686c8ce02136f
EFA 1.17.0	MD5: d430fc841563c11c3805c5f82a4746b1 SHA256：75ab0cee4fb6bd38889dce313183f5d3a83bd233e0a6ef6205d8352821ea901d
EFA 1.16.0	MD5: 399548d3b0d2e812d74dd67937b696b4 SHA256：cecec36495a1bc6fdc82f97761a541e4fb6c9a3cbf3cfcb145acf25ea5dbd45b
EFA 1.15.2	MD5: 955fea580d5170b05823d51acde7ca21 SHA256：84df4fbc1b3741b6c073176287789a601a589313accc8e6653434e8d4c20bd49
EFA 1.15.1	MD5: c4610267039f72bbe4e35d7bf53519bc SHA256：be871781a1b9a15fca342a9d169219260069942a8bda7a8ad06d4baeb5e2efd7

版本	檢查總和
EFA 1.15.0	MD5: 9861694e1cc00d884fadac07d22898be SHA256：b329862dd5729d2d098d0507fb486bf859d7c70ce18b61c302982234a3a5c88f
EFA 1.14.1	MD5: 50ba56397d359e57872fde1f74d4168a SHA256：c7b1b48e86fe4b3eaa4299d3600930919c4fe6d88cc6e2c7e4a408a3f16452c7
EFA 1.14.0	MD5: 40805e7fd842c36ececb9fd7f921b1ae SHA256：662d62c12de85116df33780d40e0533ef7dad92709f4f613907475a7a1b60a97
EFA 1.13.0	MD5: c91d16556f4fd53becadbb345828221e SHA256：ad6705eb23a3fce44af3afc0f7643091595653a723ad0374084f4f2b715192e1
EFA 1.12.3	MD5: 818aee81f097918cfaebd724eddea678 SHA256：2c225321824788b8ca3fbc118207b944cdb096b847e1e0d1d853ef2f0d727172

版本	檢查總和
EFA 1.12.2	MD5: 956bb1fc5ae0d6f0f87d2e481d49fccf SHA256: 083a868a2c212a5a4fcf3e4d732b685ce39cceb3ca7e5d50d0b74e7788d06259
EFA 1.12.1	MD5: f5bfe52779df435188b0a2874d0633ea SHA256: 5665795c2b4f09d5f3f767506d4d4c429695b36d4a17e5758b27f033aee58900
EFA 1.12.0	MD5: d6c6b49fafb39b770297e1cc44fe68a6 SHA256: 28256c57e9ecc0b0778b41c1f777a9982b4e8eae782343dfe1246079933dca59
EFA 1.11.2	MD5: 2376cf18d1353a4551e35c33d269c404 SHA256: a25786f98a3628f7f54f7f74ee2b39bc6734ea9374720507d37d3e8bf8ee1371
EFA 1.11.1	MD5: 026b0d9a0a48780cc7406bd51997b1c0 SHA256: 6cb04baf5ffc58ddf319e956b5461289199c8dd805fe216f8f9ab8d102f6d02a

版本	檢查總和
EFA 1.11.0	MD5: 7d9058e010ad65bf2e14259214a36949 SHA256: 7891f6d45ae33e822189511c4ea1d14c9d54d000f6696f97be54e915ce2c9dfa
EFA 1.10.1	MD5: 78521d3d668be22976f46c6fecc7b730 SHA256: 61564582de7320b21de319f532c3a677d26cc46785378eb3b95c636506b9bcb4
EFA 1.10.0	MD5: 46f73f5a7afe41b4bb918c81888fefa9 SHA256: 136612f96f2a085a7d98296da0afb6fa807b38142e2fc0c548fa986c41186282
EFA 1.9.5	MD5: 95edb8a209c18ba8d250409846eb6ef4 SHA256: a4343308d7ea4dc943ccc21bcebed913e8868e59bfb2ac93599c61a7c87d7d25
EFA 1.9.4	MD5: f26dd5c350422c1a985e35947fa5aa28 SHA256: 1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14

版本	檢查總和
EFA 1.9.3	MD5: 95755765a097802d3e6d5018d1a5d3d6 SHA256: 46ce732d6f3fcc9edf6a6e9f9df0ad136054328e24675567f7029edab90c68f1
EFA 1.8.4	MD5: 85d594c41e831afc6c9305263140457e SHA256：0d974655a09b213d7859e658965e56dc4f23a0eee2dc44bb41b6d039cc5bab45

Elastic Fabric Adapter 版本備註

下表說明 Elastic Fabric Adapter 軟體的版本歷史記錄和變更日誌。

版本	變更	版本日期
1.50.0	• 停止對 Amazon Linux 2 的支援。支援 AL2 的最後一個 EFA 安裝程式版本是 1.49.0。 • 升級到 libfabric 2.6.0amzn1.0 • 升級到 EFA 驅動程式 3.3.0 • 升級到 rdma-core 64.0 • 升級到 AWS OFI NCCL 外掛程式 1.21.1	2026 年 8 月 21 日
1.49.0	• 升級到 libfabric 2.4.0amzn5.0 • 升級到 EFA 驅動程式 3.1.0	2026 年 6 月 27 日

版本	變更	版本日期

	• 升級到 rdma-core 63.0
	• 升級到 AWS OFI NCCL 外掛程式 1.20.0
	• 停止對 openSUSE Leap 的支援。
	• NGC 安裝更新和錯誤修正：
	  • 使用 時，安裝遺失`--build-ngc` 的 libfabric-aws-bin/dev 套件。
	  • 移除 build-ngc 使用案例的自動偵測，並移除相關聯的`--disable-build-ngc` 選項。
	  • `--skip-rdma-core` 指定 時，不會再隱含 `--build-ngc` 。
	  • 錯誤修正：使用 `--build-ngc` 且未指定 `--skip-rdma-core` 時，現在會如預期安裝 rdma-core。
	  • 所有 NGC 安裝使用案例都避免完全安裝。
	  • `--skip-mpi` 和 現在`--skip-rdma-core` 是全域選擇，不限於 NGC 使用案例。	

版本	變更	版本日期
1.48.0	• 升級到 libfabric 2.4.0amzn3.0 • 最佳化 RDM 操作中的 shm 地址擷取 • 修正 DC 封包完成、CQ 耗盡、AH 單元測試和 MR 記錄的使用use-after-free錯誤 • 修正 efa_rdm_ep_get_peer 和 QP 鎖定中的競爭條件 • 修正未初始化的描述項陣列，造成 FI_EFA_ENABLE_SHM_TRANSFER=0 的分隔錯誤 • 修正 fi_info、err_buf、EFA 裝置銷毀和單元測試中的記憶體流失 • 在 CQ 輪詢中新增 base_ep 的 null 檢查，以處理銷毀QPs • 在 efa_qp_create 中設定 qp->base_ep，如果 efa_mr_reg_ibv_mr 傳回 NULL，則傳回錯誤 • 將 QP 資料表從網域移至具有裝置層級鎖定的裝置 • 將 QP 產生新增至資料路徑直接請求 ID，並驗證 CQE 上的 QP 編號 • 直接為資料路徑停止設定 qp->ibv_qp_ex->wr_id • 在接收接收完成時，正確從 longcts_send 清單中移除 txe • 重構並清除 efa_cq_handle_error • 將 MR 註冊計數器轉換為原子操作 • 遷移 shm MR 註冊碼並引入內部 mr regv 函數 • 在 ASAN 下預設停用 MR 快取	2026 年 4 月 14 日

版本	變更	版本日期
	• 只有在請求 FI_HMEM 但 p2p 不可用時，才停用 zcpy_rx • 請勿追蹤非偵錯建置的 tx pkt 集區 • 改善 wait_send 程序與模擬函數 • 調整 efa-direct CQ err_data 和 efa_show_help 的日誌層級 • 在 ASAN 下定義保護區巨集並停用機制 • 新增 ep 啟用錯誤路徑和錯誤完成處理的單元測試 • 升級到 libnccl-ofi 1.19.0 • 修正處理由 ROCm/HSA 建立的 dma-buf 控點的錯誤，並在 ROCm 平台上使用 HSA 執行期時新增 dma-buf 支援偵測 • 修正潛在的記憶體流失 • 已修正在 Docker 容器中產生 GB200 的 NCCL 拓撲，其中 NUMA 節點與套件節點中斷連線會導致產生不正確的拓撲 • 初始化期間 NIC 0 上的 QP 使用率降低 • 改善 P6-B200 和 P6-B300 執行個體類型的調校器演算法選擇 • 支援在 NGC 容器上安裝 libnccl-ofi • 新增個別 RPM/DEB 套件簽章驗證支援 • 新增對 RHEL 10 的支援 • 在 Debian 12 和 RHEL 10 上新增 OFI NCCL 外掛程式	

版本	變更	版本日期

1.47.0	• 升級到 libfabric 2.4.0amzn1.0
	  • 重構對等管理：將 endpoint→對等雜湊貼圖移至 AV 層級
	  • 新增封包項目產生計數器以防止 ABA 問題
	  • 使用等待物件實作封鎖 CQ 讀取支援 (fi_cq_sread)
	  • 修正 eager/mulreq/longread RTM 封包處理中的無兩次修正
	  • 修正以裝置 RDMA 功能為基礎的 MR 註冊 BV 存取旗標
	  • 停用 FI_OPT_EFA_SENDRECV_IN_ORDER_ALIGNED_128_BYTES （傳回 -FI_EOPNOTSUPP)
	  • 最佳化封包項目以容納兩個 x86 快取行 (128 位元組）
	  • 一律在 RTM/RTA 錯誤時成功執行 efa_rdm_srx_start 和滑動接收時段
	  • 在端點清除期間釋出 RX 封包項目之前取消連結
	• 升級到 libnccl-ofi 1.18.0
	  • P6-B300 支援：新增 P6-B300 的自訂調校器決策
	  • 透過減少較小訊息大小的頻道計數，改善 P6-B200 上的 PAT 效能
	  • 已變更通訊協定預設值，預設為 Trn1 上的 RDMA 通訊協定，預設為 g7e.8xlarge 上的 SENDRECV 通訊協定
	  • 動態平台選擇：新增了功能，可根據是否存在 AWS NICs 在執行時間啟用 AWS 最佳化。這可讓單一外掛程式二進位檔案同時用於 AWS 和非AWS 平台。在編譯時， AWS optimizations 仍可停用。
		2026 年 1 月 29 日

版本	變更	版本日期
	• 已修正對 RDMA 通訊協定中非 FI_MR_VIRT_ADDR 供應商的支援 • 改善向 NCCL 報告的 NIC PCIe 連結速度和寬度 • 重新設計執行緒模型以支援多執行緒應用程式，而無需為每個執行緒使用單獨的 Libfabric 網域。 • 已修正對 FI_MR_ENDPOINT 供應商（僅支援 SENDRECV 通訊協定）的支援，方法是以正確順序清除資源 • 升級到 rdma-core 61.0 • 如果提供執行緒網域，請使用單一執行緒 CQ • 驗證 CQE 程序上的 QP 編號 • 升級到 efa 驅動程式 3.0.0 • 改善管理員錯誤處理 • 完成輪詢時檢查 QP 號碼是否正確 • 移除最佳頁面大小問題清單的無條件後端連接埠 • 列印錯誤指標的 errno 字串 • 簡化 CQ 建立流程中的程式碼 • 使用 umem 建立流程在 CQ 中向後移植上游變更 • 調整新 DKMS 版本的 DKMS 組態 • 升級以開啟 MPI 5.0.9amzn1 • 錯誤修正：根據 MPI 執行緒支援設定網域執行緒層級 •	

版本	變更	版本日期
	錯誤修正：請求 FI_COMPLETION 旗標，以確保為所有資料傳輸事件產生完成 • 在 BTL 和 MTL 之間共用網域，以減少為修復高核心計數系統的資源耗盡而建立的網域總數	
1.46.0	• 新增對 Debian 13 的支援 • 升級到 efa-nv-peermem 1.2.3 • 使用新的核心和 DKMS 修正建置錯誤和警告 • 升級到 libfabric 2.3.1amzn4.0 • 啟用 efa-rdm 通訊協定路徑的直接資料路徑 • 啟用 ROCr HMEM 支援 • 新的 FI_OPT_EFA_USE_UNSOLICITED_WRITE_RECV 選項，用於停用主動寫入接收 • 最佳化連線建立時間 • 在 RHEL 9、Rocky Linux 9 和 Debian 13 上新增 libnccl-ofi	2025 年 12 月 12 日
1.45.1	• 升級到 libfabric 2.3.1amzn3.0 • 修正當提供者公告支援 4 個位元組時，將 cq_data 截斷為 2 個位元組的錯誤	2025 年 11 月 26 日

版本	變更	版本日期

1.45.0	• 升級到 rdma-core 60.amzn0
	  • 擴展 DV 查詢 CQ 以傳回門鈴
	• 升級到 libfabric 2.3.1amzn2.0
	  • 支援 efa-direct 的 FI_RX_CQ_DATA 模式
	  • 最佳化資料路徑直接路徑中的 WQE 貼文
	  • 修正 cntr 進度中的競爭條件
	  • 改善警告訊息的記錄
	  • 正確處理已移除對等的 RMA 完成
	  • 達到 AH 限制時，從隱含 AV 中移出 AH 項目
	• 升級到 libnccl-ofi 1.17.2
	  • 修正 Libfabric 初始化失敗時 NCCL v2.28.x 的當機問題
	  • 新增對 g7e 執行個體系列的支援
	  • 修正 NCCL 在支援 DMA-BUF 的平台上可能錯誤嘗試使用 GPUDirect RDMA 路徑的問題
	  • 修正需要每個端點記憶體註冊NICs 的關機順序問題 (Cray Slingshot)
		2025 年 11 月 17 日

版本	變更	版本日期

1.44.0	• 升級到 rdma-core 59.amzn0
	  • 新增支援以將 QP 和 CQ 綁定至執行緒網域
	  • 新增支援以配置父系網域
	  • 新增支援以配置執行緒網域
	  • 新增直接動詞查詢 QP 和 CQ
	  • 新增選項以使用外部記憶體建立 CQ`
	  • 在 QP 損毀後修正 CQ 輪詢
	• 升級到 libfabric 2.3.1amzn1.0
	• 升級到 Open MPI 5.0.8amzn1
	  • 功能增強
	  • 錯誤修正
	• 升級到 libnccl-ofi 1.17.1
	  • 支援 NCCL v2.28.3-1，同時保持與 NCCL v2.17.1 和更新版本的向後相容性。
	  • 使用 platform-aws 進行建置需要 Libfabric v1.22.0amzn4.0 或更高版本。使用 Libfabric v2.3.1amzn1.0 之前的版本進行了測試。
	  • 修正以支援 CUDA12 與 CUDA13 之間的相容性。
		2025 年 10 月 29 日

版本	變更	版本日期

1.43.3	• 升級到 efa 驅動程式 2.17.3
	  • 使用 NVIDIA 580 驅動程式支援 P2P
	  • 調整 6.17 及以上核心的 MR 註冊
		2025 年 10 月 01 日

1.43.2	• 升級到 libnccl-ofi 1.16.3
	  • 支援 NCCL v2.27.7-1，同時保持與 NCCL v2.17.1 和更新版本的向後相容性。
	  • 使用 platform-aws 進行建置需要 Libfabric v1.22.0amzn4.0 或更高版本。使用 Libfabric 2.1.0amzn5.0 之前的版本進行了測試。
	  • 在所有 Amazon EC2 執行個體類型上預設啟用每個執行緒的網域，以改善 NCCL 建立多個代理執行緒所在的一些應用程式的效能
		2025 年 8 月 15 日

1.43.1	• 升級到 libnccl-ofi 1.16.2
	  • 支援 NCCL v2.27.6-1，同時保持與 NCCL v2.17.1 和更新版本的向後相容性。
	  • 使用 platform-aws 進行建置需要 Libfabric v1.22.0amzn4.0 或更高版本。使用 Libfabric 2.1.0amzn4.0 之前的版本進行了測試。
	  • 新增了平台組態，以支援在 Amazon EC2 p5.4xlarge 執行個體類型上使用 OFI NCCL 外掛程式
	• 升級到 libfabric 2.1.0amzn5.0
		2025 年 7 月 31 日

版本	變更	版本日期

1.43.0	• 升級到 libnccl-ofi 1.16.1
	  • 支援 NCCL v2.27.6-1，同時保持與 NCCL v2.17.1 和更新版本的向後相容性。
	  • 使用 platform-aws 進行建置需要 Libfabric v1.22.0amzn4.0 或更高版本。使用 Libfabric 2.1.0amzn3 之前的版本進行了測試。
	  • 更新拓撲檔案中報告的 PCI 連結速度格式，以符合核心 5.7+
	  • 新增了 SKIP_NICS_WITHOUT_ACCEL_AT_SAME_PCI_LEVEL，以略過在相同 pci 層級沒有加速器的 libfabric nic
	• 升級到 efa 驅動程式 2.17.2
	• 升級到 efa-nv-peermem 1.2.2
	• 升級到 libfabric 2.1.0amzn4.0
	• 升級到 rdma-core 58.amzn0
	  • 修正工作請求索引雙重用途
	  • 將 WQE 長度新增至 post_send 追蹤點
	• 啟用 Graviton 平台的最佳化
	• 停止對 Ubuntu 20.04 的支援
		2025 年 7 月 25 日

版本	變更	版本日期

1.42.0	• 升級到 efa 驅動程式 2.15.3
	• 升級到 efa-nv-peermem 1.2.1
	• 升級到 rdma-core 57.amzn1
	  • 修正工作請求索引雙重用途
	• 升級到 libfabric 2.1.0amzn3.0
	• 升級到 libnccl-ofi 1.15.0
	  • 支援 NCCL 2.26.6-1，同時保持與 NCCL v2.17.1 和更新版本的向後相容性。
	  • 使用 platform-aws 進行建置需要 Libfabric v1.22.0amzn4.0 或更高版本。使用 Libfabric 2.1.0amzn3 之前的版本進行了測試。
	  • 建置系統和平台支援
	  • 新增了 Amazon EC2 P6-B200 平台支援
	  • 將預設外掛程式的程式庫名稱變更為 libnccl-net-ofi.so，並依預設建立從 libnccl-net-ofi.so 至 libnccl-net.so 的 symlink，以維持向後相容性。這可讓使用者設定 NCCL_NET_PLUGIN=ofi，強制 NCCL 使用 OFI 外掛程式進行通訊。指定要設定的 --disable-nccl-net-symlink 會跳過 symlink，允許在相同的容器中安裝多個外掛程式。
	  • 調校和效能改善
	  • 新增 P6-B200 上適用於 0x0 和 0x7 位元遮罩的 AllReduce、AllGather 和 ReduceScatter 區域的調頻器支援
	  • 根據經驗結果和分析，更新 P5en 和 P6-B200 平台的預設延遲
	  • 進行更新，使用具有 trafficClass 參數的 NCCL v10 API，以支援未來的流量優先級
	  • 將外掛程式的程式碼庫從 C 移轉到 C++
		2025 年 6 月 6 日

版本	變更	版本日期
	• 新增對工作的支援，其中每個 GPU 的 NIC 數量因系統而異。如需詳細資訊，請參閱 OFI_NCCL_FORCE_NUM_RAILS 執行時期環境變數文件。	
1.41.0	• 升級到 rdma-core 57.amzn0	2025 年 5 月 16 日
1.40.0	• 升級到 libfabric 2.1.0amzn2.0 • 升級到 rdma-core 56.0 • 升級到 efa-config 1.18。 • 棄用對 Debian 10 的支援 • 升級到 libnccl-ofi 1.14.2 • 在適用於 Ubuntu 和 Amazon Linux 2023 的 ARM 上啟用 Libfabric 中的 CUDA 支援 • 在 Ubuntu 和 Amazon Linux 2023 的 ARM 上新增 libnccl-ofi	2025 年 5 月 2 日
1.39.0	• 升級到 libfabric 2.1.0 • 升級到 Open MPI 5.0.6 • 升級到 libnccl-ofi 1.14.1 • 升級到 efa 驅動程式 2.15.0	2025 年 4 月 16 日
1.38.1	• 更新至 `Libfabric 1.22.0amzn5.0`	2025 年 3 月 3 日
1.38.0	• 加 `libnccl-ofi 1.13.2-1`	2025 年 1 月 8 日
1.37.0	• 升級至 `libfabric 1.22.0amzn4.0` • 升級至 `rdma-core 54.amzn0`	2024 年 11 月 18 日

版本	變更	版本日期
1.36.0	• 新增對 Debian 12 的支援 • 在 /opt/amazon/efa/bin 中以公用程式指令碼形式分佈 efa_test.sh • 升級到 efa 驅動程式 2.13.0 • 升級至 libfabric 1.22.0amzn3.0 • 升級至 rdma-core 54.0 • 升級到 Open mpi 4.1.7	2024 年 11 月 7 日
1.35.0	• 升級到 Open MPI 5.0.5 • 升級到 PRRTE 3.0.6 並將 RPM 套件重新命名為 prrte-aws • 將 OpenPMIx RPM 套件重新命名為 pmix-aws • 設定 dpkg 和 RPM 建置的建置選項 • 升級至 Libfabric 1.22.0amzn2.0 • 升級到 efa 驅動程式 2.12.1	2024 年 10 月 14 日
1.34.0	• 停止對 CentOS 7 和 RHEL 7 的支援 • 擷取 Libfabric 1.22.0amzn1.0 • 將 efa-config 套件更新至 1.17	2024 年 8 月 6 日
1.33.0	• 升級到 efa 驅動程式 2.10.0 • 升級至 rdma-core 52.0 • 升級到 PRRTE 3.0.5 • 升級到 Open MPI 5.0.3 • 新增對 Ubuntu 24.04 LTS 的支援 • 移除 OpenPMIx、PRRTE 和 Open MPI HTML 文件。使用者應參閱官方網站。 • 停止對 openSUSE 15.4 或舊版本的支援	2024 年 6 月 20 日

版本	變更	版本日期
1.32.0	• 升級到 efa 驅動程式 2.8.0 • 升級到 efa-nv-peermem 驅動程式 1.1.1 • 將 efa-config 套件升級至 1.16 • 將 efa-profile 套件升級至 1.7 • 升級至 `rdma-core 50.0` • 升級至 `libfabric 1.21.0amzn1.0` • 透過 `fi_pingpong` 連接埠選取來增強 `efa_test.sh` • 視需要從系統安裝較新的 rdma-core	2024 年 4 月 18 日
1.31.0	• 升級到 OpenPMIx 4.2.8 • 升級到 PRRTE 3.0.3 • 升級到 Open MPI 5.0.2，並將 MCA 元件轉換為 DSO。 • 升級至 `Libfabric 1.20.1amzn1.0`	2024 年 3 月 7 日
1.30.0	• 包含 OpenPMIx 4.2.7，預設安裝在 `/opt/amazon/pmix` • 包含 PRRTE 3.0.2，預設安裝在 `/opt/amazon/prrte` • 包含 Open MPI 5.0.0，預設安裝在 `/opt/amazon/openmpi5` • 將 efa-profile 更新至 1.6	2023 年 12 月
1.29.1	• 擷取 `libfabric 1.19.0amzn4.0`	2023 年 12 月
1.29.0	• 擷取 efa kernel 驅動程式 2.6.0 • 擷取 `libfabric 1.19.0amzn3.0`	2023 年 11 月
1.28.0	• 將 Open MPI 升級至 4.1.6 • 擷取 `libfabric 1.19.0amzn2.0`	2023 年 10 月
1.27.0	• 將 `Libfabric 1.19.0amzn1.0` 新增至安裝程式 • 為 Open MPI 4 啟用 GCC 內建原子	2023 年 9 月
1.26.1	• `efa_test.sh` ：修正無限重試錯誤	2023 年 9 月

版本	變更	版本日期
1.26.0	• 新增對 RHEL 9 的支援 • 修正多個分佈上的偵錯建置 • 擷取 `libfabric 1.18.2amzn1.0` • `efa_test.sh` ：新增 `fi_pingpong` 的重試	2023 年 9 月
1.25.1	• 擷取 `libfabric 1.18.1amzn1.0` • 在 EFA 安裝程式指令碼中更新逾時	2023 年 9 月
1.25.0	• 新增對 Amazon Linux 2023 的支援 • 安裝後測試中的錯誤修正	2023 年 7 月
1.24.1	• 將 libfabric 升級到 1.18.1 • 將 efa 驅動程式升級至 2.5.0	2023 年 7 月
1.24.0	• 擷取 `rdma-core 46.0` • 擷取 efa 驅動程式 2.4.1 • 支援 Debian 11	2023 年 6 月
1.23.1	• 擷取 `libfabric 1.18.0amzn2.0`	2023 年 6 月
1.23.0	• 新增對 Debian 10 的支援 • 停止對 Ubuntu 18.04 LTS 的支援 • 將 efa-config 套件升級至 1.14 • 擷取 `libfabric 1.18.0amzn1.0`	2023 年 5 月
1.22.1	• 將 libfabric 升級到 1.17.1	2023 年 3 月
1.22.0	• 將 Open MPI 升級至 4.1.5 • 將 libfabric 升級到 1.17.0 • 將 efa-config 套件升級至 1.13	2023 年 2 月

版本	變更	版本日期
1.21.0	• 新增對 Rocky Linux 9 作業系統的支援 • 擷取 efa 驅動程式 2.1.1 • 擷取 `libfabric 1.16.1amzn3.0` • 將 efa-config 套件升級至 1.12	2022 年 12 月
1.20.0	• 新增對 Rocky Linux 8 作業系統的支援。 • 擷取 efa 驅動程式 2.1.0。 • 擷取 `rdma-core 43.0`。 • 擷取 `libfabric 1.16.1amzn1.0`	2022 年 11 月
1.19.0	• 擷取 `libfabric 1.16.0` • 使用 `--enable-orterun-prefix-by-default` 建置 Open MPI	2022 年 10 月
1.18.0	• 新增對 Ubuntu22.04 的支援	2022 年 8 月
1.17.3	• 將 libfabric 更新至 `1.16.0~amzn4.0` 。~ 指示它是 libfabric 1.16.0 的發行前版本。 • 將安裝後 Pingpong 測試逾時延長至 20 秒	2022 年 8 月
1.17.2	• 將 libfabric 更新至 `1.16.0~amzn3.0` 。~ 指示它是 libfabric 1.16.0 的發行前版本。	2022 年 7 月
1.17.1	• 將 libfabric 更新至 `1.16.0~amzn2.0` 。~ 指示它是 libfabric 1.16.0 的發行前版本。 • 在建置 libfabric 時停用實驗網路提供者	2022 年 7 月
1.17.0	• 將 `rdma-core` 更新至 v41.0 • 將 Open MPI 更新至 4.1.4 • 將 libfabric 更新至 `1.16.0~amzn1.0` 。~ 指示它是 libfabric 1.16.0 的發行前版本。	2022 年 7 月

版本	變更	版本日期
1.16.0	• 將 libfabric 更新至 `1.15.1amzn1.0` ，包含神經元程式庫名稱變更 • 將 efa-config 升級至 1.10 • 在 libfabric 建置中排除 opx 和 rxd 提供者	2022 年 6 月
1.15.2	• 將 libfabric 更新至 1.14.1	2022 年 5 月
1.15.1	• 將 libfabric 更新至 `1.14.0amzn1.0`	2022 年 3 月
1.15.0	• 修正導致 Open SuSE 15.3 安裝失敗的錯誤 • 停止支援 Open SuSE 15.2 (當 Open SuSE 15.2 達到生命週期結束時) • 停止支援 CentOS 8 (當 CentOS 8 達到生命週期結束時) • 將 libfabric 更新至 1.14.0 • 將 efa 核心驅動程式更新至 1.16.0 • 將 `rdma-core` 更新至 v39.0 • 將 Open MPI 更新至 4.1.2 版。	2022 年 2 月
1.14.1	• 將 libfabric 更新至 `1.13.2amzn1.0` 。	2021 年 10 月
1.14.0	• 擷取 efa 核心驅動程式 1.14.2。 • 將 `efa_installer.sh` 中的 `-g, --enable-gdr` 設為無操作選項，因為最新的 efa 核心驅動程式預設會啟用 GDR 支援。 • 擷取 `rdma-core v37.0`。 • 擷取 `libfabric 1.13.2`。 • 新增套件清單並比較 RPM/DEB 以在安裝期間列出，從而防止未知的套件安裝。 • 在安裝程式指令碼中新增休眠，以等待在 EFA 驅動程式重新載入後套用 udev 規則。	2021 年 10 月

版本	變更	版本日期
1.13.0	• 將 `rdma-core` 更新至 v35.0。 • 將 `libfabric` 更新至 v1.13.0amzn1.0。 • 在 Gravition2 平台上新增對 CentOS/RHEL 8 的 EFA 支援。 • 在安裝程式指令碼中新增版本比較邏輯，以便在較高版本系統中安裝本機套件時跳過本機套件安裝。	2021 年 8 月
1.12.3	• 將 EFA 核心模組更新至 1.13.0。 • 將 efa-config 套件更新至 1.9 版。改進巨型頁面保留的計算，以處理大型預設巨型頁面大小。	2021 年 7 月
1.12.2	• 將 EFA 核心模組更新至 1.12.3。 • 使用 `--with-libevent=external` 和 `--with-hwloc=external` 建置 Open MPI debian 套件。 • 將 Open MPI rpm 建置 ID 設為 2，以修正 CentOS 8 上 HWLOC 的向後相容性問題。 • 移除 SLES15SP2 和 openSUSE 15.2 上 kernel-devel 和 kernel-source 套件的安裝。	2021 年 6 月
1.12.1	• 將 Libfabric 更新至 1.11.2amzon1.1 版。 • 將 EFA 核心模組更新至 1.12.1 版。	2021 年 5 月

版本	變更	版本日期
1.12.0	• 將 Open MPI 更新至 4.1.1 版。 • 將 Libfabric 更新至 1.11.2amzn1.0 版。 • 使用與 AL2 rdma-core 相同的封裝組態，為 Amazon Linux 2 建置 rdma-core。 • 建置 Open MPI RPM 時，請勿強制執行 `-Wl,--enable-new-dtags` 。 • 使用 hwloc 和 libevent 的系統程式庫建置 Open MPI。 • 將 EFA 核心模組更新至 1.12.0 版 • 將 efa-config 套件更新至 1.8 版。改進長期執行個體的巨型頁面保留計算。 • 將 efa-profile 套件更新至 1.5 版。移除 Open MPI 集體調校檔案，它可作為修正 P4d 上的 Open MPI 4.1.0 當機的解決方法。 • 將 `rdma-core` 更新至 v32.1。 • 停止對 Amazon Linux 1 和 Ubuntu 16.04 的支援。	2021 年 5 月
1.11.2	• 透過組態檔案來變更用於實作 MPI_BARRIER 的預設演算法，修復在 P4d 上使用 Open MPI 時遇到的 Open MPI 當機問題。 • 透過 `--disable-builtin-atomics` ，在 ARM 上的 Open MPI 中停用內建原子來解決編譯器問題。	2021 年 2 月
1.11.1	• 將 Open MPI 更新至 4.1.0 版。 • 將 efa-config 套件更新至 1.7 版。改進巨型頁面保留計數的計算。 • 將 efa-profile 套件更新至 1.3 版。由於使用 Open MPI 4.1.0，移除不需要的集體決策檔案。	2020 年 12 月
1.11.0	• 新增對 Gravition2 平台的支援。 • 將 `rdma-core` 更新至 31.2amzn 版。 • 將 Libfabric 更新至 1.11.1amzn1.0 版。 • 將 efa-config 更新至 1.6 版。 • 將 efa-profile 更新至 1.2 版。	2020 年 12 月

版本	變更	版本日期
1.10.1	• 新增對 CentOS/RHEL 8 的支援。 • 新增對 Ubuntu 20.04 的支援 • 新增對 SUSE Linux Enterprise 15 的支援。	2020 年 11 月
1.10.0	• 新增 P4d 平台的 GPUDirect RDMA 支援。使用 `--enable-gdr` 安裝程式選項來安裝 GDR 感知核心模組和使用者空間。 • 將 EFA 核心模組更新至 1.10.2 版。 • 將 `rdma-core` 更新至 31.amzn0 版。 • 將 Libfabric 更新至 1.11.1 版。 • 將 Open MPI 更新至 4.0.5 版。 • 將 efa-config 更新至 1.5 版。 • 將 efa-profile 更新至 1.1 版。包括改善的 Open MPI 集體決策檔案。	2020 年 10 月
1.9.5	• 將 efa-config 更新至 1.4 版。修正 Open MPI 集體決策檔案中的錯誤。	2020 年 9 月
1.9.4	• 將 Open MPI 更新至 4.0.3 版。 • 將 Libfabric 更新至 1.10.1amazon1.1 版。 • 將 `rdma-core` 更新至 28.amzn0 版。	2020 年 7 月
1.9.3	• 將 EFA 核心模組更新至 1.6.0 版。 • 將 `rdma-core` 更新至 28.amzn0 版。 • 將 Libfabric 更新至 1.10.1amzn1.1 版。 • 將 efa-config 更新至 1.3 版。新增 Open MPI 的集體調校檔案。 • 如果已安裝 dkms，則略過安裝。 • 修正 `--skip-kmod` 安裝模式以實際運作。	2020 年 6 月
1.8.4	• 將組態檔案移至 efa-config 和 efa-profile 套件中，以便作業系統套件管理員進行追蹤。 • 將 Open MPI 更新至 4.0.3 版。	2020 年 4 月

版本	變更	版本日期
1.8.3	• 將 EFA 核心模組更新至 1.5.1 版。 • 在某些平台上分佈 DKMS，而不是依賴 EPEL 儲存庫來提高安裝可靠性。 • 在 RHEL 7 上，安裝建置在 CentOS 7 上的 RPM，而不是建置在 Amazon Linux 2 上的 RPM。	2020 年 2 月
1.8.2	• 由於核心模組和 `rdma-core` 之間的裝置命名不相符，請將 `rdma-core` 還原至版本 25。	2020 年 1 月
1.8.1	• 將 Libfabric 更新至 1.9.0amzn1.1 版。	2020 年 1 月
1.8.0	• 將 `rdma-core` 更新至 27.0 版 • 將 EFA 核心模組更新至 1.5.0 版。 • 將 Libfabric 更新至 1.9.0amzn1.0 版。 • 新增選項 `--minimal` 以僅安裝 EFA 核心模組和 `rdma-core` 。	2019 年 12 月
1.7.1	• 將 Libfabric 更新至 1.8.1amzn1.3 版。	2019 年 12 月
1.7.0	• 新增 Libfabric 模組檔案。 • 將 Libfabric 更新至 1.8.1amzn1.1 版。	2019 年 11 月
1.6.2	• 將 Open MPI 更新至 1.6.2 版。	2019 年 10 月
1.6.1	• 將 Libfabric 更新至 1.8.1amzn1.0 版。 • 將 Open MPI 更新至 4.0.1 版。 • 將 `rdma-core` 更新至 26.0 版	2019 年 10 月
1.5.4	• 將 EFA 核心模組更新至 1.4.1 版。	2019 年 9 月
1.5.3	• 將 EFA 核心模組更新至 1.3.1 版。 • 除非安裝核心驅動程式，否則請避免安裝 `kernel-devel` 或 `linux-headers` 套件。	2019 年 9 月
1.5.1	• 僅在存在 EFA 裝置時設定巨型頁面。	2019 年 8 月

版本	變更	版本日期
1.5.0	• 將 Libfabric 更新至 1.8.0amzn1.1 版。 • 將 `rmda-core` 更新至 25.0 版。	2019 年 8 月
1.4.1	• 將 Libfabric 和 Open MPI 程式庫路徑 (`/opt/amazon/efa/lib64` 和 `/opt/amazon/efa/openmpi/lib`) 新增至 `/etc/ld.so.conf.d/efa.conf`，以確保 Open MPI 和 Libfabric 共用程式庫的位置正確。	2019 年 7 月
1.4.0	• 將 EFA 核心模組更新至 1.3.0 版。 • 將 Libfabric 更新至 1.8.0amzn1.0 版 • 支援 Intel MPI 2019 Update 4 的第一個版本。	2019 年 7 月

Amazon EC2 拓撲

Amazon EC2 拓撲提供運算容量的相對鄰近性的階層檢視。您可以使用此資訊大規模管理高效能運算 (HPC)、機器學習 (ML) 和生成式 AI 運算基礎設施。

可用的 API

Amazon EC2 提供兩個 API 來了解 EC2 拓撲：

- DescribeInstanceTopology
  - 顯示正在執行的執行個體在網路階層中相對於彼此的位置。
  - 協助最佳化在現有執行個體上執行工作負載的位置。
- DescribeCapacityReservationTopology
  - 顯示在您啟動執行個體之前，預留容量在網路階層中相對於彼此的位置。
  - 在啟動執行個體之前，讓您了解預留容量的位置，以協助規劃容量。

主要優點

EC2 拓撲具有以下主要優點：

- 容量管理 – 最佳化資源使用率。
- 任務排程 – 做出有關工作負載置放的明智決策。

- 節點排名 – 了解緊密耦合執行個體上效能最佳化的相對鄰近性。

考量事項

- 拓撲檢視僅適用於：
  - 處於 `running` 狀態的執行個體
  - 處於 `pending` 或 `active` 狀態的容量保留
- 每個拓撲檢視都是唯一的 AWS 帳戶。
- AWS 管理主控台 不支援檢視拓撲。
- 雖然拓撲資訊可協助您了解執行個體置放，但您無法使用它來啟動實際靠近現有執行個體的新執行個體。若要影響執行個體置放，您可以在叢集置放群組中建立容量保留。

定價

描述 EC2 拓撲無需額外費用。

目錄

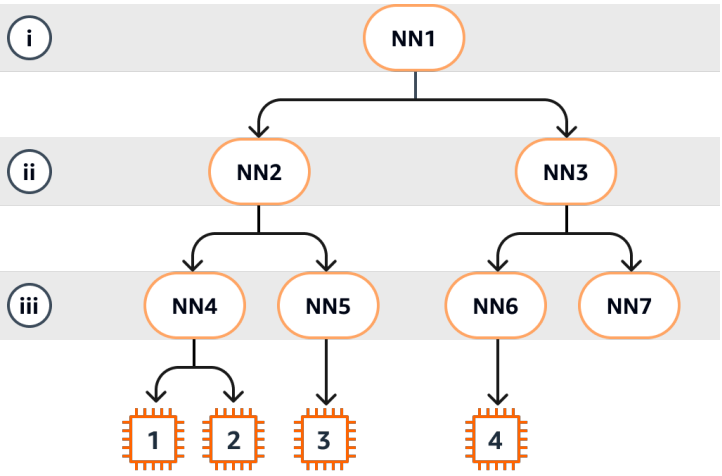
- Amazon EC2 拓撲如何運作
- Amazon EC2 拓撲的先決條件
- Amazon EC2 執行個體拓撲範例

Amazon EC2 拓撲如何運作

AWS 網路以圖層階層排列。EC2 執行個體會連接至第三層或第三層以下的網路，具體取決於執行個體類型。執行個體的拓撲由一組節點描述，網路的每一層都有一個節點。DescribeInstanceTopology 或 DescribeCapacityReservationTopology API 回應中的節點集可提供網路階層的由上而下檢視，並將底部節點連接到執行個體。

> **Note**
>
> 有些執行個體類型在代表網路 4 層的節點集中包含 4 個網路節點，而其他則包含代表網路 3 層的 3 個網路節點。如需支援的執行個體類型，請參閱 執行個體類型。
> 根據容量保留的類型，可能只會看到 1、2 或 3 個網路節點。

下圖的視覺化表示法有助於了解 EC2 拓撲。網路節點識別為 NN1 – NN7。數字 i、ii 和 iii 在此範例中代表不同的網路層。數字 1、2、3 和 4 可標識 EC2 執行個體。執行個體連接到底層中的節點，在下列範例中以 iii 標識。可有多個執行個體連接到同一個節點。



在此範例中：

- 圖中的執行個體 1 連接到第 iii 層中的網路節點 4 (NN4)。NN4 連接到第 ii 層中的網路節點 2 (NN2)，NN2 連接到第 i 層中的網路節點 1 (NN1)，也就是此範例中的網路階層頂端。網路節點集由 NN1、NN2 和 NN4 組成，以上層到底層的階層表示。
- 執行個體 2 也連接至網路節點 4 (NN4)。執行個體 1 和執行個體 2 有共同的網路節點集：NN1、NN2 和 NN4。
- 執行個體 3 連接至網路節點 5 (NN5)。NN5 連接到 NN2，而 NN2 連接到 NN1。執行個體 3 的網路節點集為 NN1、NN2 和 NN5。
- 執行個體 4 連接至網路節點 6 (NN6)。其網路節點集為 NN1、NN3 和 NN6。

考量執行個體 1、2 和 3 的鄰近性時，執行個體 1 和 2 彼此更接近，因為兩者連接到相同的網路節點 (NN4)，而執行個體 3 連接至不同的網路節點 (NN5) 因此較遠。

考量此圖中所有執行個體的鄰近性時，比起與執行個體 4 之間的距離，執行個體 1、2 和 3 彼此的距離更接近，因為這幾個執行個體的網路節點集中都有 NN2。

一般而言，如果連接到任何兩個執行個體的網路節點相同，這兩個執行個體實際位置彼此相近，就像執行個體 1 和 2 的情況一樣。此外，網路節點之間的躍點數越少，執行個體彼此之間的距離就越接近。例如，執行個體 1 和 3 到共同網路節點 (NN2) 的躍點，少於與執行個體 4 的共同網路節點 (NN1) 之間的躍點，因此它們之間的距離比執行個體 4 更近。

在此範例中，沒有在網路節點 7 (NN7) 下執行的執行個體，因此 API 輸出不會包含 NN7。

如何解譯 DescribeInstanceTopology 輸出

可以使用 DescribeInstanceTopology API 來描述執行個體拓撲。API 的輸出提供執行個體基礎網路拓撲的階層式檢視。

下列輸出範例對應上圖中四個執行個體的網路拓撲資訊。為了說明本範例，輸出範例中會包含註解。

輸出中的以下資訊非常重要：

- NetworkNodes 說明單個執行個體的網路節點集。
- 在每個網路節點集中，網路節點會以由上至下的階層順序列出。
- 連接至執行個體的網路節點是清單中的最後一個網路節點 (底層)。
- 若要找出哪些執行個體彼此靠近，請先在底層中尋找一般網路節點。如果底層沒有共同網路節點，請往上層尋找共同網路節點。

在下列範例中，輸出i-1111111111example和i-2222222222example位於與此範例中的其他執行個體相較之間最接近的位置，因為它們在底層nn-4444444444example中具有共同的網路節點。

> **Note**
>
> 回應包含 3 個或更多網路節點。如需每個支援執行個體類型之回應中網路節點數量的資訊，請參閱 執行個體類型。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",  //Corresponds to instance 1
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 in layer i
                "nn-2222222222example",           //Corresponds to NN2 in layer ii
                "nn-4444444444example"            //Corresponds to NN4 in layer iii -
 bottom layer, connected to the instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
```

```
        {
            "InstanceId": "i-2222222222example",  //Corresponds to instance 2
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-4444444444example"            //Corresponds to NN4 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",  //Corresponds to instance 3
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-5555555555example"            //Corresponds to NN5 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-444444444example",  //Corresponds to instance 4
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",          //Corresponds to NN1 - layer i
                "nn-3333333333example",          //Corresponds to NN3 - layer ii
                "nn-6666666666example"           //Corresponds to NN6 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

如何解譯 DescribeCapacityReservationTopology 輸出

可以使用 DescribeCapacityReservationTopology API 來描述容量保留拓撲。輸出提供預留容量的基礎網路拓撲的階層式檢視。

下列輸出範例對應上圖中的網路拓撲資訊。為了說明本範例，輸出範例中會包含註解。

輸出中的以下資訊非常重要：

- NetworkNodes 說明單一容量保留的網路節點集。
- 在每個網路節點集中，網路節點會以由上至下的階層順序列出。
- 連接至容量保留的網路節點是清單中的最後一個網路節點 (底層)。
- 若要判斷容量保留是否會彼此接近，請先在輸出中的底層找到共同的網路節點。如果底層沒有共同網路節點，請往上層尋找共同網路節點。

在下列範例輸出中，cr-1111111111example 位於 nn-2222222222example，而 cr-2222222222example 位於 nn-3333333333example。由於容量保留位於 layer ii 中的不同網路節點上，因此從一個容量保留中的執行個體到另一個容量保留中的執行個體的通訊效率將會降低。

> **Note**
>
> 回應包含 1、2 或 3 個網路節點，具體取決於容量保留的類型。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1111111111example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",      //Corresponds to NN1 - layer i
                "nn-2222222222example"       //Corresponds to NN2 - layer ii
                // Visibility of additional nodes requires an instance launch and
                // the DescribeInstanceTopology API
            ],
            "AvailabilityZone": "us-west-2a"
        },
```

```
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",      //Corresponds to NN1 - layer i
                "nn-3333333333example"       //Corresponds to NN3 - layer ii
                // Visibility of additional nodes requires an instance launch and
                // the DescribeInstanceTopology API
            ],
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

DescribeInstanceTopology 和 DescribeCapacityReservationTopology 之間的差異

下表比較 DescribeInstanceTopology 和 DescribeCapacityReservationTopology API 之間的主要差異：

比較項目	DescribeInstanceTopology	DescribeCapacityReservation Topology
使用階段	啟動後 (執行模式)	啟動前 (規劃和管理模式)
主要用途	在執行中的執行個體上最佳化工作負載	執行個體啟動前的容量規劃和容量保留管理 (合併、分割、指派)
網路節點數量	顯示執行中執行個體的所有節點。如果執行個體位於容量保留中，則首批節點將符合對應的容量保留拓撲，後面跟著要連線至執行個體的其他節點。	顯示部分節點集合，這些節點會根據容量保留狀態 (pending 或 active) 和類型而有所不同。*
State	執行個體必須處於 running 狀態	容量保留必須處於 pending 或 active 狀態
使用案例	• 工作負載最佳化	• 容量規劃

比較項目	DescribeInstanceTopology	DescribeCapacityReservationTopology
	• 效能調校 • 執行時期拓撲分析	• 容量保留管理 (merge/split/assign) • 啟動前拓撲評估

* 對於 Ultraservers 的容量區塊，描述 `active` 容量保留或其執行中執行個體的拓撲時，網路節點集相同。

Amazon EC2 拓撲的先決條件

若要描述 Amazon EC2 拓撲，請確保執行個體和容量保留符合下列先決條件。

先決條件：

- AWS 區域
- 執行個體類型
- State
- IAM 許可

AWS 區域

支援 AWS 區域：

- 美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)
- 非洲 (開普敦)
- 亞太區域 (雅加達)、亞太區域 (香港)、亞太區域 (海德拉巴)、亞太區域 (墨爾本)、亞太地區 (孟買)、亞太區域 (大阪)、亞太地區 (首爾)、亞太地區 (新加坡)、亞太地區 (雪梨)、亞太地區 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (巴黎)、歐洲 (西班牙)、歐洲 (斯德哥爾摩)、歐洲 (蘇黎世)
- 以色列 (特拉維夫)
- 中東 (巴林)、中東 (阿拉伯聯合大公國)
- 南美洲 (聖保羅)

- AWS GovCloud （美國西部）
- AWS European Sovereign Cloud (Germany)

以色列（特拉維夫）和 AWS GovCloud （美國西部）不支援 DescribeCapacityReservationTopology API。

執行個體類型

支援的執行個體類型：

- 在回應中傳回 3* 個網路節點：
  - g6e.xlarge | g6e.2xlarge | g6e.4xlarge | g6e.8xlarge | g6e.12xlarge | g6e.16xlarge | g6e.24xlarge | g6e.48xlarge | g7e.2xlarge | g7e.4xlarge | g7e.8xlarge | g7e.12xlarge | g7e.24xlarge | g7e.48xlarge
  - hpc6a.48xlarge | hpc6id.32xlarge | hpc7g.4xlarge | hpc7g.8xlarge | hpc7g.16xlarge | hpc7a.12xlarge | hpc7a.24xlarge | hpc7a.48xlarge | hpc7a.96xlarge | hpc8a.96xlarge
  - p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | p5e.48xlarge | p5en.48xlarge | p6e-gb200.36xlarge
  - trn1.2xlarge | trn1.32xlarge | trn1n.32xlarge | trn2.48xlarge | trn2u.48xlarge
- 在回應中傳回 4* 個網路節點：
  - p6-b200.48xlarge | p6-b300.48xlarge

* 使用 DescribeInstanceTopology API 時，傳回的網路節點數量才適用。對於 DescribeCapacityReservationTopology API，傳回的網路節點數量有所不同，具體取決於容量保留的類型和狀態。

可用的執行個體類型因區域而異。如需詳細資訊，請參閱依區域劃分的 Amazon EC2 執行個體類型。

State

- 對於 DescribeInstanceTopology – 執行個體必須處於 running 狀態。
- 對於 DescribeCapacityReservationTopology – 容量保留必須處於 pending 或 active 狀態。

您無法獲取處於任何其他狀態的執行個體或容量保留的拓撲資訊。

IAM 許可

您的 IAM 身分 (使用者、使用者群組或角色) 需要下列許可：

- `ec2:DescribeInstanceTopology`
- `ec2:DescribeCapacityReservationTopology`

Amazon EC2 執行個體拓撲範例

可以使用 describe-instance-topology 命令來描述 EC2 執行個體的拓撲。也可以使用 describe-capacity-reservation-topology 命令來描述容量保留的拓撲。

使用沒有參數或篩選條件的 `describe-instance-topology` 或 `describe-capacity-reservation-topology` 命令時，回應會包含指定區域中符合此命令所支援執行個體類型的所有執行個體或容量保留 (取決於使用的命令)。您可以藉由包含 `--region` 參數或設定預設區域來指定區域。如需設定預設區域的詳細資訊，請參閱：為您的 Amazon EC2 資源選取區域。

您可以加入參數以傳回符合指定執行個體或容量保留 ID 或放置群組名稱的執行個體或容量保留。您也可以加入篩選條件，以傳回符合指定執行個體類型或執行個體系列的執行個體或容量保留，或是指定可用區域或本地區域中的執行個體或容量保留。您可以包含單一參數或篩選條件，或是參數和篩選條件的組合。

輸出會分頁，預設每頁最多 20 個執行個體或容量保留。您可以使用 `--max-results` 參數指定每頁最多 100 個執行個體或容量保留。

如需詳細資訊，請參閱describe-instance-topology及describe-reservation-topology-topology。

所需的許可

需要以下許可：

- `ec2:DescribeInstanceTopology` – 用於描述執行個體拓撲
- `ec2:DescribeCapacityReservationTopology` – 用於描述容量保留拓撲

範例

- 範例 1：DescribeInstanceTopology - 執行個體 ID
- 範例 2：DescribeInstanceTopology - 置放群組名稱參數
- 範例 3：DescribeInstanceTopology - 執行個體類型篩選條件

  - 範例 3a – 指定執行個體類型的完全相符篩選條件
  - 範例 3b – 執行個體系列的萬用字元篩選條件
  - 範例 3c – 合併執行個體系列和完全相符篩選條件
- 範例 4：DescribeInstanceTopology - 區域 ID 篩選條件
  - 範例 4a – 可用區域篩選條件
  - 範例 4b – Local Zone 篩選條件
  - 範例 4c – 合併可用區域和本機區域篩選條件
- 範例 5：DescribeInstanceTopology - 執行個體類型和區域 ID 篩選條件
- 範例 6：DescribeCapacityReservationTopology - 容量保留 ID
- 範例 7：DescribeCapacityReservationTopology - 執行個體類型篩選條件

範例 1：DescribeInstanceTopology - 執行個體 ID

AWS CLI

若要說明特定執行個體的拓撲

使用具有 `--instance-ids` 參數的 describe-instance-topology 命令。輸出僅包含符合指定執行個體 ID 的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --instance-ids i-1111111111example i-2222222222example
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
```

```
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明特定執行個體的拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -InstanceId i-1111111111example, i-2222222222example
```

範例 2：DescribeInstanceTopology - 置放群組名稱參數

AWS CLI

若要描述特定置放群組中執行個體的拓撲

使用具有 group-names 參數的 describe-instance-topology 命令。輸出僅包含位於任一指定置放群組中的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --group-names ML-group HPC-group
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要描述特定置放群組中執行個體的拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -GroupName ML-group, HPC-group
```

範例 3：DescribeInstanceTopology - 執行個體類型篩選條件

您可以依指定的執行個體類型進行篩選 (完全相符)，或依執行個體系列進行篩選 (使用萬用字元)。您也可以結合指定的執行個體類型篩選器和執行個體族群篩選。

範例 3a – 指定執行個體類型的完全相符篩選條件

AWS CLI

若要說明具有特定執行個體類型的執行個體拓撲

使用具有 `instance-type` 篩選條件的 describe-instance-topology 命令。輸出僅包含具有指定執行個體類型的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1n.32xlarge
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明具有特定執行個體類型的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1n.32xlarge"}
```

範例 3b – 執行個體系列的萬用字元篩選條件

AWS CLI

若要說明具有特定執行個體系列的執行個體拓撲

使用具有 instance-type 篩選條件的 describe-instance-topology 命令。輸出僅包含具有指定執行個體系列的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1*
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
```

```
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明具有特定執行個體系列的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1*"}
```

範例 3c – 合併執行個體系列和完全相符篩選條件

AWS CLI

若要說明具有執行個體系列或執行個體類型的執行個體拓撲

使用具有 `instance-type` 篩選條件的 describe-instance-topology 命令。輸出僅包含符合指定條件的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
```

```
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge"
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-4343434343example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明具有執行個體系列或執行個體類型的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"}
```

範例 4：DescribeInstanceTopology - 區域 ID 篩選條件

您可以使用 zone-id 篩選條件來依可用區域或本地區域進行篩選。您也可以結合可用區域篩選條件和本地區域篩選條件。

範例 4a – 可用區域篩選條件

AWS CLI

若要說明特定可用區域中的執行個體拓撲

使用具有 zone-id 篩選條件的 describe-instance-topology 命令。輸出僅包含指定可用區域中的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明特定可用區域中的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1"}
```

範例 4b – Local Zone 篩選條件

AWS CLI

若要說明特定本地區域中的執行個體拓撲

使用具有 zone-id 篩選條件的 describe-instance-topology 命令。輸出僅包含指定本地區域中的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-atl2-az1
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明特定本地區域中的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-atl2-az1"}
```

範例 4c – 合併可用區域和本機區域篩選條件

AWS CLI

若要說明特定區域中的執行個體拓撲

使用具有 zone-id 篩選條件的 describe-instance-topology 命令。輸出僅包含位於任一指定區域中的執行個體。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1,use1-atl2-az1
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
```

```
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明特定區域中的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

範例 5：DescribeInstanceTopology - 執行個體類型和區域 ID 篩選條件

可將篩選條件合併到單一命令中。

AWS CLI

若要說明具有特定執行個體類型、執行個體系列和區域的執行個體拓撲

使用具有 `instance-type` 和 `zone-id` 篩選條件的 describe-instance-topology 命令。回應包含具有任一指定執行個體類型且位於任一指定區域的任何執行個體。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge" \
              "Name=zone-id,Values=use1-az1,use1-atl2-az1"
```

以下為範例輸出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
```

```
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要說明具有特定執行個體類型、執行個體系列和區域的執行個體拓撲

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"} `
            @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

範例 6：DescribeCapacityReservationTopology - 容量保留 ID

AWS CLI

若要描述特定容量保留的拓撲

使用具有 `capacity-reservation-id` 參數的 describe-capacity-reservation-topology 命令。輸出僅包含符合指定容量保留 ID 的容量保留。

```
aws ec2 describe-capacity-reservation-topology \
    --region us-east-1 \
    --capacity-reservation-id cr-1111111111example cr-2222222222example
```

以下為範例輸出。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1111111111example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        },
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5en.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要描述特定容量保留的拓撲

使用 Get-EC2CapacityReservationTopology cmdlet。

```
Get-EC2CapacityReservationTopology `
    -CapacityReservationId cr-1111111111example cr-2222222222example
```

範例 7：DescribeCapacityReservationTopology - 執行個體類型篩選條件

您可以依指定的執行個體類型進行篩選 (完全相符)，或依執行個體系列進行篩選 (使用萬用字元)。您也可以結合指定的執行個體類型篩選器和執行個體族群篩選。

AWS CLI

若要描述具有特定執行個體類型的容量保留拓撲

使用具有 `instance-type` 篩選條件的 describe-capacity-reservation-topology 命令。回應包含具有指定執行個體類型的任何執行個體。

```
aws ec2 describe-capacity-reservation-topology \
    --region us-east-1 \
    --filters Name=instance-type,Values=p5en.48xlarge
```

以下為範例輸出。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5en.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

若要描述具有特定執行個體類型的容量保留拓撲

使用 Get-EC2CapacityReservationTopology cmdlet。

```
Get-EC2CapacityReservationTopology `
```

```
    -Filter @{Name="instance-type"; Values="p5en.48xlarge"}
```

Amazon EC2 執行個體的置放群組

為了符合您的工作負載需求，您可以將一組相互依存的 EC2 執行個體啟動到置放群組中以影響其置放。

視工作負載類型而定，您可以採取下列其中一個置放策略來建立置放群組：

- 叢集：會將執行個體緊密地包裝在一個可用區域內。此策略可讓工作負載達到高效能運算 (HPC) 應用程式典型的緊密耦合node-to-node通訊所需的低延遲網路效能。
- 分割區 – 將您的執行個體分散到邏輯分割區，以便一個分割區中的執行個體群組不會與不同分割區中的執行個體群組共用基礎硬體。大量分散和複寫的工作負載 (例如 Hadoop、Cassandra 和 Kafka) 通常採取此策略。
- 分散：嚴格地將一小組執行個體分散到不同的底層硬體，以減少相互關聯的故障。
- 精確時間 – 將執行個體放置在支援硬體上，可直接存取 AWS 基礎設施中的高精確度時間來源。

置放群組是選用的程序。如果您未將執行個體啟動至置放群組中，EC2 會嘗試以將所有執行個體分散於底層硬體上的方式置放執行個體，使相互關聯的故障減到最少。

定價

建立置放群組無需收費。

規則與限制

在使用置放群組之前，請記住下列規則：

- 執行個體一次可以放置在一個置放群組中；您無法將執行個體放在多個置放群組中。
- 置放群組無法合併。
- 隨需容量保留 和 zonal 預留執行個體 會在特定的可用區域中，為 EC2 執行個體提供容量保留。當您啟動執行個體時，如果執行個體屬性符合隨需容量保留或區域預留執行個體指定的屬性，則執行個體會自動使用預留容量。如果您在置放群組中啟動執行個體，也是如此。
- 您無法在置放群組中啟動專用執行個體。
- 您無法啟動在置放群組中設定為中斷時停止或休眠的 Spot 執行個體。

目錄

- 置放群組的置放策略
- 為您的 EC2 執行個體建立置放群組
- 變更 EC2 執行個體的放置位置
- 刪除置放群組
- 共用放置群組
- 上的置放群組 AWS Outposts

置放群組的置放策略

您可以使用下列其中一種置放策略為 EC2 執行個體建立置放群組。

置放策略

- 集群放置群組
- 分區置放群組
- 分散放置群組
- 精確時間置放群組

集群放置群組

集群放置群組是單一可用區域中的執行個體邏輯分組。執行個體不會隔離到單一機架。集群放置群組可以跨越相同區域中互連的虛擬私有網路 (VPC)。相同集群放置群組中的執行個體可享受 TCP/IP 流量更高的每個流程傳輸量限制，並且放置在相同的網路高等分頻寬區段中。

下列映像會顯示放入集群放置群組中的執行個體。



能夠受惠於低網路延遲、高網路傳輸量或這兩者的應用，建議採用集群放置群組。當大部分的網路流量都在群組中的執行個體之間時，也建議使用這些群組。若要為置放群組提供最低延遲和最高的網路效能 (每秒封包數)，請選擇支援增強型聯網功能的執行個體類型。如需詳細資訊，請參閱增強型聯網。

您可以在建立叢集置放群組時指定父系精確度時間置放群組，以確保在叢集置放群組中啟動的執行個體放置在具有精確時間功能的硬體上。當您建立叢集置放群組時，請使用 `--parent-group-id` 參數來建立此關係。

我們建議您以下列方式啟動執行個體：

- 使用單一啟動要求，在置放群組中啟動您所需的執行個體數目。
- 對置放群組中的所有執行個體使用相同的執行個體類型。

如果之後您試著在置放群組中加入更多執行個體，或是在置放群組中啟動超過一種執行個體類型，將會提高出現容量不足錯誤的機率。

如果先停止置放群組中的執行個體，然後再啟動，此執行個體仍會在置放群組中執行。不過，如果執行個體可用的容量不足，起始作業將會失敗。

若置放群組中已經擁有正在運作的執行個體，當您在此置放群組中啟動執行個體時，如果收到容量錯誤，請停止和啟動該置放群組中的所有執行個體，然後試著再次啟動。啟動執行個體可能會將其遷移到具有所有請求執行個體容量的硬體。

規則與限制

集群放置群組適用的規則如下：

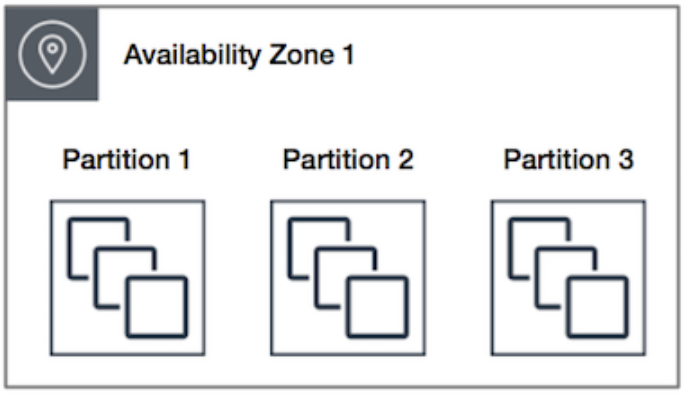
- 支援下列執行個體類型：
  - 最新一代執行個體，但爆量效能執行個體 (例如 T2)、Mac1 執行個體和 M7i-flex 執行個體除外。
  - 下列上一代執行個體：A1、C3、C4、I2、M4、R3 和 R4。
- 集群放置群組不能涵跨多個可用區域。
- 在集群放置群組中，兩個執行個體之間傳送資料的最高網路傳輸速度，會受限於這兩個執行個體之中較慢者的速度。針對需要高傳輸量的應用程式，請選擇符合您需求之網路連線能力的執行個體類型。
- 下列規則適用於為增強聯網所啟用的執行個體：
  - 集群放置群組內的執行個體，單一傳輸流的速度最多可達 10 Gbps。不在集群放置群組內的執行個體，單一傳輸流的速度最高可達 5 Gbps。
  - 在同一個區域中，Amazon S3 儲存貯體之間透過公有 IP 地址空間或 VPC 端點的往來流量，可以使用所有可用的執行個體總頻寬。

- 您可以在集群放置群組中啟動多種執行個體類型。不過，這會降低啟動成功所需容量可用的機率。我們建議在集群放置群組中，針對所有執行個體使用相同的執行個體類型。
- 您可以在叢集置放群組中建立隨需容量保留，以明確預留該置放群組中的容量。請注意，您無法使用區域預留執行個體預留容量，因為它們無法在置放群組中明確預留容量。
- 容量區塊不支援置放群組。如需詳細資訊，請參閱ML 的容量區塊。
- 叢集置放群組的網際網路和內部部署資源 Direct Connect 連線的網路流量限制為 5 Gbps。

分區置放群組

分區置放群組有助於降低應用程式發生相關硬體故障的可能性。使用分區置放群組時，Amazon EC2 會將每個群組分割成稱為分區的邏輯區段。Amazon EC2 會確保置放群組內的每個分區有其自己的一組機架。每個機架有其自己的網路和電源。一個置放群組內不會有兩個分割區共用相同的機架，這樣可讓您隔離硬體故障在應用程式內造成的影響。

下圖顯示在單一可用區域中分區置放群組的簡單視覺化呈現。其中顯示放入分區置放群組中的執行個體，而該群組具有三個分區 - 分區 1、分區 2 和分區 3。每個分區包含多個執行個體。一個分割區中的執行個體不會與其他分割區中的執行個體共用機架，可讓您將單一硬體故障的影響僅限縮在相關聯的分割區中。



分區置放群組可用於跨不同機架來部署大型分散式和複寫式工作負載，例如 HDFS、HBase 和 Cassandra。將執行個體啟動到分區置放群組時，Amazon EC2 會嘗試在指定的分區數中均勻分佈執行個體。您也可以將執行個體啟動到特定的分區，讓您能夠更有效控制放置執行個體的位置。

分區置放群組的分割區可以在同一區域中的多個可用區域。分區置放群組在每個可用區域最多可有 7 個分區。可啟動到分區置放群組中的執行個體數，僅會受到帳戶限制的限制。

此外，分割區置放群組提供分割區的可見性 — 您可以查看哪些執行個體位於哪些分割區中。您可以與拓撲感知應用程式 (例如 HDFS、HBase 和 Cassandra) 共用此資訊。這些應用程式會使用此資訊來進行智能資料複寫決策，以提高資料可用性與持久性。

如果開始或啟動分區置放群組中的執行個體，但是沒有足夠的唯一硬體來滿足請求，則請求會失敗。Amazon EC2 會隨著時間釋出更多可用的不同硬體，因此您可以之後再嘗試進行請求。

規則與限制

分區置放群組適用的規則如下：

- 分區置放群組在每個可用區域最多支援 7 個分區。您可以在分區置放群組中啟動的執行個體數，僅會受到帳戶限制的限制。
- 將執行個體啟動到分區置放群組時，Amazon EC2 會嘗試在所有分區中均勻分配執行個體。Amazon EC2 不保證會在所有分區中均勻分配執行個體。
- 具有 專用執行個體 的分區置放群組最多可有 2 個分區。
- 容量保留不能在分區置放群組中預留容量。

分散放置群組

分散置放群組是一組放在不同硬體上的執行個體。

若應用程式具有應該分開保存的少量重要執行個體，建議使用分散置放群組。透過分散放置群組來啟動執行個體，可降低同時發生故障的風險，這種情況可能會在執行個體共用相同設備時發生。分散置放群組可讓您存取不同的硬體，因此適合混合執行個體類型或隨時間啟動執行個體。

如果開始或啟動分散置放群組中的執行個體，但是沒有足夠的唯一硬體來滿足請求，則請求會失敗。Amazon EC2 會隨著時間釋出更多可用的不同硬體，因此您可以之後再嘗試進行請求。

分散置放群組可以將執行個體分散到機架或主機。如需詳細資訊，請參閱the section called “上的置放群組 AWS Outposts”。

下圖顯示單一可用區域中分散置放群組中的七個執行個體。每個執行個體都放置在不同的機架上，其中每個機架都有自己的網路和電源。



規則與限制

分散置放群組適用的規則如下：

- AWS 區域 和 支援機架層級分散置放群組 AWS Outposts。
- 僅支援主機層級分散置放群組 AWS Outposts。
- 機架層級分散置放群組可以跨越相同區域中的多個可用區域，但在每個可用區域中最多支援七個執行中的執行個體。

  例如，在具有三個可用區域的區域中，您可以在分散置放群組中執行總共 21 個執行個體，每個可用區域中有 7 個執行個體。如果您嘗試在已有七個執行個體的可用區域中啟動第八個執行個體，則執行個體不會啟動。

  如果您在一個可用區域中需要七個以上的執行個體，建議使用多重分散置放群組。使用多個分散置放群組並不保證執行個體會分散在群組中，但確實會確保執行個體分散在每個群組中，這會限制特定失敗類別的影響。
- 在 上 AWS Outposts，機架層級分散置放群組可以容納與 Outpost 部署中機架數量相同的執行個體，而主機層級分散置放群組可以容納與主機數量相同的執行個體。
- 專用執行個體 不支援分散置放群組。
- 容量保留不能在分散置放群組中預留容量。

精確時間置放群組

精確時間置放群組會將執行個體放置在支援的硬體上，可直接存取 AWS 基礎設施中的高精確時間來源。在精確時間置放群組中啟動的執行個體可以存取增強型 Amazon Time Sync Service，相較於所有 Amazon EC2 執行個體可用的標準本機 NTP 端點，該服務可提供更精確的本機時間來源。

對於需要微秒準確時鐘同步的應用程式，建議使用精確時間置放群組，例如需要交易排序和一致性的分散式資料庫、需要精確時間戳記的金融服務，以及使用本機時鐘讀取排序事件的分散式系統。

精確時間置放群組中的執行個體受益於下列項目：

- 增強的本機 NTP 時間來源，提高了準確性，適用於所有支援的作業系統，無需額外的組態。
- 存取 PTP 硬體時鐘 (PHC) 裝置以進行微秒精確同步處理 （僅限 Linux 執行個體）。
- 奈米秒解析度網路測量的硬體封包時間戳記 （僅限 Linux 執行個體）。

如需設定執行個體以使用增強型時間來源的相關資訊，請參閱 設定 EC2 執行個體的時間參考以使用本機 Amazon Time Sync Service。

下列規則適用於精確時間置放群組：

規則與限制

- 如需目前支援的執行個體系列清單，請參閱 設定 EC2 執行個體的時間參考以使用本機 Amazon Time Sync Service。
- 如果您在精確時間置放群組中啟動或啟動執行個體，且硬體不足，無法提供增強型 Amazon Time Sync Service 的存取權，則請求會失敗。請稍後再試，或嘗試不同的可用區域。
- 您無法刪除做為叢集置放群組父項的精確時間置放群組。
- 使用精確度時間置放群組不收取額外費用。

為您的 EC2 執行個體建立置放群組

您可以使用置放群組來控制執行個體相對於彼此的置放。請先建立配置群組，便可以在該配置群組中啟動執行個體。

限制

您可以在每個區域的每個帳戶建立最多 500 個放置群組。

Console

若要建立置放群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Placement Groups (置放群組)。
3. 選擇建立放置群組。
4. 針對名稱，指定群組的名稱。
5. 針對置放策略，選擇群組的置放策略：叢集、分散、分割區或精確度時間。
   a. 如果您選擇分散，對於分散層級，請選擇主機或機架。
   b. 如果您選擇分割區，請在分割區數量中輸入群組的分割區數量。
   c. 如果您選擇叢集，您可以選擇指定父系精確時間置放群組，以確保叢集置放群組中的執行個體在具備精確時間功能的硬體上啟動。

6. （選用）若要新增標籤，請在標籤下選擇新增標籤，然後輸入索引鍵和值。
7. 選擇建立群組。

AWS CLI

使用 create-placement-group 命令。

建立叢集置放群組

下列範例會建立一個放置群組，它使用 cluster 放置策略，並且套用一個標籤以及 purpose 的索引鍵和 production 的數值。

```
aws ec2 create-placement-group \
    --group-name my-cluster \
    --strategy cluster \
    --tag-specifications 'ResourceType=placement-
group,Tags={Key=purpose,Value=production}'
```

建立具有精確時間置放群組父系的叢集置放群組

下列範例會建立叢集置放群組，使用 --parent-group-id 參數指定父系精確度時間置放群組。

```
aws ec2 create-placement-group \
    --group-name my-cluster \
    --strategy cluster \
    --parent-group-id pg-0aaa1111111111111
```

建立分區放置群組

下列範例會建立使用置放策略的partition置放群組，並使用 --partition-count 參數指定五個分割區。

```
aws ec2 create-placement-group \
    --group-name HDFS-Group-A \
    --strategy partition \
    --partition-count 5
```

建立精確的時間置放群組

下列範例會建立使用置放策略的precision-time置放群組。

```
aws ec2 create-placement-group \
    --group-name my-precision-time-pg \
    --strategy precision-time
```

PowerShell

使用 New-EC2PlacementGroup cmdlet。

建立叢集置放群組

下列範例會建立叢集置放群組。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy cluster
```

建立具有精確時間置放群組父系的叢集置放群組

下列範例會建立叢集置放群組，以使用 -ParentGroupId 參數指定父系精確度時間置放群組。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy cluster `
    -ParentGroupId pg-0aaa1111111111111
```

建立分區放置群組

下列範例會建立分區放置群組。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy partition `
    -PartitionCount 5
```

建立精確的時間置放群組

下列範例會建立精確的時間置放群組。

```
New-EC2PlacementGroup `
    -GroupName my-precision-time-pg `
    -Strategy precision-time
```

變更 EC2 執行個體的放置位置

您可以使用下列任一方式變更執行個體的置放群組：

- 新增執行個體移到放置群組
- 在置放群組之間移動執行個體
- 從置放群組中移除執行個體

需求

在變更執行個體的置放群組之前，該執行個體必須處於 `stopped` 狀態。

Console

變更執行個體置放

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇 Actions (動作)、Instance settings (執行個體設定)、Modify instance placement (修改執行個體的配置)。
5. 對於放置群組，執行下列其中一項操作：
   - 對於置放群組，選擇要做為移動執行個體目的地的放置群組。
   - 若要將執行個體從放置群組移至另一放置群組，請選擇放置群組。
   - 若要從置放群組中移除執行個體，請選擇無。
6. 選擇儲存。

AWS CLI

將執行個體移到置放群組

使用以下 modify-instance-placement 命令。

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name MySpreadGroup
```

從置放群組中移除執行個體

使用以下 modify-instance-placement 命令。當您為置放群組名稱指定空字串時，會從其目前的置放群組中移除執行個體。

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

PowerShell

將執行個體移到置放群組

使用具有置放群組名稱的 Edit-EC2InstancePlacement cmdlet。

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName MySpreadGroup
```

從置放群組中移除執行個體

使用具有置放群組名稱空字串的 Edit-EC2InstancePlacement cmdlet。

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName ""
```

刪除置放群組

如果需要更換或不再需要某個置放群組，可以刪除該置放群組。置放群組不得包含任何執行個體，您才可予以刪除。您可以終止執行個體，將其移動到另一個放置群組，或將其從放置群組中移除。

您無法刪除屬於叢集置放群組父項的置放群組。首先刪除叢集置放群組。

Console

刪除置放群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Placement Groups (置放群組)。
3. 選取置放群組並依序選取 Actions (動作)、Delete (刪除)。

4. 出現確認提示時，請輸入 **Delete**，然後選擇 Delete (刪除)。

AWS CLI

刪除置放群組

使用 delete-placement-group 命令。

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

刪除置放群組

使用 Remove-EC2PlacementGroup cmdlet。

```
Remove-EC2PlacementGroup -GroupName my-cluster
```

共用放置群組

放置群組共用可讓您影響不同 AWS 帳戶帳戶所擁有的相互依存執行個體的放置。擁有者可以跨多個 AWS 帳戶 或其組織內共用置放群組。參與者可以在與其帳戶共用的置放群組中啟動執行個體。

置放群組擁有者可與下列項目共用置放群組：

- 組織 AWS 帳戶 內部或外部的特定
- 組織內的組織單位
- 整個 組織

您可以使用 VPC 對等互連來連接由個別 擁有的執行個體， AWS 帳戶 並取得共用叢集置放群組提供的完整延遲優勢。

目錄

- 規則與限制
- 所需的許可
- 跨可用區域共用
- 置放群組共用

- 置放群組取消共用

規則與限制

當您共用置放群組或與您共用置放群組時，下列規則和限制適用。

- 若要共用置放群組，您必須在 中擁有該群組 AWS 帳戶。您無法將已共用給您的放置群組再共用出去。
- 共用分割區或分散置放群組時，置放群組限制不會變更。共用分割區置放群組在每個可用區域最多可有七個分割區，而共用分散置放群組在每個可用區域最多可支援七個執行中的執行個體。
- 若要與您的組織或組織中的組織部門共用放置群組，您必須啟用與 AWS Organizations的共用。如需詳細資訊，請參閱共用您的 AWS 資源。
- 使用 AWS 管理主控台 啟動執行個體時，您可以選取與您共用的任何置放群組。使用 AWS CLI 啟動執行個體時，您必須依 ID 而非名稱指定共用置放群組。只有當您是共用展示位置群組的擁有者時，才可以使用展示位置群組的名稱。
- 您必須負責管理共用置放群組中您所擁有的執行個體。
- 您無法檢視或修改與共用放置群組相關聯但非您擁有的執行個體和容量保留。
- 置放群組的 Amazon Resource Name (ARN) 包含擁有置放群組的帳戶 ID。您可以使用置放群組 ARN 的帳戶 ID 部分來識別與您共用的置放群組擁有者。

所需的許可

若要共用置放群組，使用者必須具有下列動作的許可：

- `ec2:PutResourcePolicy`
- `ec2:DeleteResourcePolicy`

跨可用區域共用

為確保資源分配至區域中的所有可用區域，可用區域會獨立對應至各個帳戶的名稱。這可能導致帳戶之間的可用區域命名出現差異。例如，您 `us-east-1a`的可用區域 AWS 帳戶 可能沒有與`us-east-1a`另一個區域相同的位置 AWS 帳戶。

若要基於您的帳戶來識別專用主機的相對位置，您必須使用可用區域 ID (AZ ID)。AZ ID 是可用區域在所有 AWS 帳戶之間唯一且一致的識別符。例如，`use1-az1` 是 `us-east-1` 區域的可用區域 ID，它在每一個 AWS 帳戶的位置都相同。如需詳細資訊，請參閱 AZ ID。

置放群組共用

若要共用置放群組，您必須將它新增至資源共用。資源共用是可讓您跨 共用資源 AWS RAM 的資源 AWS 帳戶。資源共享指定要共用的資源，以及共用它們的消費者。

如果您是 組織的一部分 AWS Organizations，則會啟用組織內的共用，且組織中的取用者會獲得共用置放群組的存取權。

如果置放群組與組織 AWS 帳戶 外部的 共用， AWS 帳戶 擁有者會收到加入資源共用的邀請。他們可以在接受邀請後存取共用置放群組。

您可以在 AWS 帳戶 使用 之間共用置放群組 AWS Resource Access Manager。如需詳細資訊，請參閱「AWS RAM 使用者指南」中的建立資源共用。

置放群組取消共用

置放群組擁有者可以隨時取消共用已共用置放群組。當您取消共用已共用放置群組時，下列變更將會發生：

- 與 AWS 帳戶 置放群組共用的 無法再啟動執行個體或保留容量。
- 在共用置放群組中執行的任何執行個體都會與置放群組取消關聯，但會繼續在您的 中執行 AWS 帳戶。
- 共用置放群組中的任何容量保留都會與置放群組取消關聯，但仍可供您在 中使用 AWS 帳戶。

如需詳細資訊，請參閱「AWS RAM 使用者指南」中的刪除資源共用。

上的置放群組 AWS Outposts

AWS Outposts 是一種全受管服務，可將 AWS 基礎設施、服務、APIs 和工具延伸到客戶內部部署。透過提供 AWS 受管基礎設施的本機存取權， AWS Outposts 可讓客戶使用與 AWS 區域相同的程式設計界面，在內部部署中建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。

Outpost 是在客戶站點部署的 AWS 運算和儲存容量集區。 會在 AWS 區域中 AWS 操作、監控和管理此容量。

您可以在帳戶中建立的 Outposts 中建立置放群組。這樣，您就可以在站點上的 Outpost 中跨基礎硬體分散執行個體。在 Outposts 中建立和使用置放群組的方式，與在一般可用區域中建立和使用置放群組的方式相同。在 Outpost 中建立具有分散策略的置放群組時，可以選擇讓置放群組在主機層級或是機架層級分散執行個體。在主機之間分散執行個體可讓您搭配單一機架 Outpost 使用分散策略。

考量

- 機架層級分散置放群組可以容納與 Outpost 部署中機架一樣多的執行個體。
- 主機層級分散置放群組可以容納與 Outpost 部署中主機一樣多的執行個體。

先決條件

您的站點必須安裝 Outpost。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立 Outpost 並訂購 Outpost 容量。

在 Outpost 中使用置放群組

1. 在 Outpost 上建立子網。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的建立子網。
2. 在 Outpost 的關聯區域中建立置放群組。如果您使用分散策略建立置放群組，您可以選擇主機層級或機架層級分散，以判斷群組如何在 Outpost 的基礎硬體之間分散執行個體。如需詳細資訊，請參閱the section called “建立置放群組”。
3. 在置放群組中啟動執行個體。在 Subnet (子網路) 欄位中，選取您在步驟 1 建立的子網路，在 Placement group name (置放群組名稱) 欄位中，選取您在步驟 2 建立的置放群組。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的在 Outpost 上啟動執行個體。

EC2 執行個體的網路最大傳輸單位 (MTU)

網路連線的最大傳輸單位 (MTU) 係允許通過該連線的最大封包大小 (以位元組為單位)。連線的 MTU 越大，單一封包能傳遞的資料也越多。乙太網幀內含封包 (或您實際傳送的資料) 和環繞的網路額外負荷資訊。

乙太網路訊框具有不同格式，最常見的格式是標準乙太網路 v2 訊框格式。其可支援 1500 MTU，此為多數網路均支援的最大乙太網路封包大小。執行個體支援的最大 MTU 因執行個體類型而異。

所有 EC2 執行個體類型都支援 1500 MTU。

目錄

- 巨型訊框 (9001 MTU)
- 路徑 MTU 探索
- 為您的 Amazon EC2 執行個體設定 MTU
- 疑難排解

巨型訊框 (9001 MTU)

透過巨型訊框，可增加每個封包的承載大小，這會增加不屬於封包成本的封包比例。透過巨型訊框，只需要較少的封包，即可傳送相同數量的可用資料。不過，某些類型的流量會受到下列最大承載的限制：

MTU 限制 1500 個位元組

- 網際網路閘道的流量
- VPN 連線的流量
- AWS 區域之間的流量，除非使用傳輸閘道

MTU 限制 8500 個位元組

- 跨區域 VPC 對等互連的流量

如果封包超過其 MTU 限制，會將其切割為片段，若其 IP 標頭設有 `Don't Fragment` 標記，則會將其捨棄。

巨型訊框應謹慎用於進出網路的流量，或任何離開 VPC 的流量。封包由中介系統切割，因此拖累流量速度。欲在 VPC 中使用巨型訊框但不減緩流出 VPC 的流量，您可依據路由設定 MTU 大小，或使用多個具備不同 MTU 大小與路由的彈性網絡介面。

以配置於集群放置群組內的執行個體而言，巨型訊框有助實現最大網路傳輸量，此情況建議使用巨型訊框。如需詳細資訊，請參閱Amazon EC2 執行個體的置放群組。

您可以針對透過 Direct Connect的 VPC 和現場部署網路之間的流量，使用 Jumbo Frame。如需詳細資訊並了解如何驗證巨型訊框功能，請參閱《Direct Connect 使用者指南》中的私有虛擬界面或傳輸虛擬界面的 MTU。

所有目前一代的執行個體類型都支援巨型訊框。下列上一代執行個體類型均支援巨型訊框：A1、C3、I2、M3 和 R3。

相關資源

- 如需 NAT 閘道相關資訊，請參閱「Amazon VPC 使用者指南」中的 NAT 閘道基本概念。
- 如需傳輸閘道相關資訊，請參閱「Amazon VPC 傳輸閘道使用者指南」中的 MTU。
- 如需 Local Zones 相關資訊，請參閱「AWS Local Zones 使用者指南」中的考量。
- 如需 AWS Wavelength，請參閱AWS Wavelength 《 使用者指南》中的最大傳輸單位。
- 對於 Outpost，請參閱 AWS Outposts 使用者指南中的服務連結最大傳輸單位需求。

路徑 MTU 探索

路徑 MTU 探索 (PMTUD) 可用於確認兩個裝置間的路徑 MTU。路徑 MTU 是原始主機和接收主機之間的路徑上支援的最大封包尺寸。在兩個主機之間的網路中，當 MTU 大小出現差異時，PMTUD 會允許接收主機使用 ICMP 訊息回應原始主機。此 ICMP 訊息指示原始主機沿著網路路徑，使用最低的 MTU 大小來重新傳送請求。如果不這樣協議，可能會因為請求太大，使得接收端主機無法接受，而發生封包捨棄。

針對 IPv4，若主機傳送的封包大小大於接收主機的 MTU，或是大於路徑上裝置的 MTU，則接收主機或裝置便會丟棄封包，然後傳回下列 ICMP 訊息：`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (類型 3，代碼 4)。這會指示傳輸主機將承載分割成多個較小的封包，然後重新傳輸它們。

IPv6 通訊協定不支援網路中的分段。若主機傳送的封包大小大於接收主機的 MTU，或是大於路徑上裝置的 MTU，則接收主機或裝置便會丟棄封包，然後傳回下列 ICMP 訊息：`ICMPv6 Packet Too Big (PTB)` (類型 2)。這會指示傳輸主機將承載分割成多個較小的封包，然後重新傳輸它們。

透過某些元件 (例如 NAT 閘道和負載平衡器) 建立的連線會自動追蹤。這意味著系統會自動為您的傳出連線嘗試啟用安全群組追蹤。如果自動追蹤連線、取消追蹤連線，或您的安全群組規則允許傳入 ICMP 流量，您可以接收 PMTUD 回應。

請注意，即使在安全群組層級允許流量，也可以封鎖 ICMP 流量，例如，如果您有拒絕進入子網路的 ICMP 流量的網路存取控制清單項目。

> **Important**
>
> 路徑 MTU 探索並不保證某些路由器不會捨棄巨型訊框。您 VPC 內的網際網路閘道僅能轉送至多 1500 位元組的封包。網際網路流量建議使用 1500 MTU 封包。

如需透過 NAT 閘道的 MTU 規則，請參閱「Amazon VPC 使用者指南」中的最大傳輸單位 (MTU)。如需透過傳輸閘道的 MTU 規則，請參閱 AWS Transit Gateway 使用者指南中的傳輸單位上限 (MTU)。

為您的 Amazon EC2 執行個體設定 MTU

網路連線的最大傳輸單位 (MTU) 係允許通過該連線的最大封包大小 (以位元組為單位)。所有 Amazon EC2 執行個體都支援標準影格 (1500 MTU)，而所有目前世代的執行個體類型都支援巨型影格 (9001 MTU)。

您可以檢視 Amazon EC2 執行個體的 MTU、檢視執行個體與其他主機之間的路徑 MTU，以及將執行個體設定為使用標準或巨型訊框。

任務

- 檢查兩個主機間的路徑 MTU
- 檢查執行個體的 MTU
- 為您的執行個體設定 MTU

檢查兩個主機間的路徑 MTU

您可以檢查您 EC2 執行個體和另一主機間的路徑 MTU。可使用 DNS 名稱或 IP 位址做為目的地。若目的地為另一個 EC2 執行個體，請確認其安全群組允許傳入的 UDP 流量。

您使用的程序會依執行個體的作業系統而定。

Linux 執行個體

在執行個體上執行 tracepath 命令，以檢查 EC2 執行個體與指定目的地之間的路徑 MTU。此命令是 `iputils` 套件的一部分，依預設可在許多 Linux 發行版本中使用。

此範例會檢查 EC2 執行個體和 `amazon.com` 間的路徑 MTU。

```
[ec2-user ~]$ tracepath amazon.com
```

在此範例中，路徑 MTU 為 1500。

```
 1?: [LOCALHOST]      pmtu 9001
 1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)   0.187ms pmtu 1500
 1:  no reply
 2:  no reply
 3:  no reply
 4:  100.64.16.241 (100.64.16.241)                              0.574ms
 5:  72.21.222.221 (72.21.222.221)                             84.447ms asymm 21
 6:  205.251.229.97 (205.251.229.97)                           79.970ms asymm 19
 7:  72.21.222.194 (72.21.222.194)                             96.546ms asymm 16
 8:  72.21.222.239 (72.21.222.239)                             79.244ms asymm 15
 9:  205.251.225.73 (205.251.225.73)                           91.867ms asymm 16
...
31:  no reply
```

```
    Too many hops: pmtu 1500
    Resume: pmtu 1500
```

Windows 執行個體

使用 mturoute.exe 檢查路徑 MTU

1. 從 https://elifulkerson.com/projects/mturoute.php：// 下載mturoute.exe到您的 EC2 執行個體。
2. 開啟命令提示字元視窗，並變更至您下載 mturoute.exe 所在的目錄。
3. 使用下列命令來檢查您 EC2 執行個體與指定目的地之間的路徑 MTU。此範例會檢查 EC2 執行個體和 www.elifulkerson.com 間的路徑 MTU。

   ```
   .\mturoute.exe www.elifulkerson.com
   ```

   在此範例中，路徑 MTU 為 1500。

   ```
   * ICMP Fragmentation is not permitted. *
   * Speed optimization is enabled. *
   * Maximum payload is 10000 bytes. *
   + ICMP payload of 1472 bytes succeeded.
   - ICMP payload of 1473 bytes is too big.
   Path MTU: 1500 bytes.
   ```

檢查執行個體的 MTU

您可以檢查執行個體的 MTU 值。部分執行個體設定為使用巨型訊框，其他則使用標準訊框大小。

您使用的程序會依執行個體的作業系統而定。

Linux 執行個體

檢查 Linux 執行個體上的 MTU 設定

在 EC2 執行個體上執行下列 ip 命令。如果主要網路介面不是 eth0，請將取代 eth0 為您的網路介面。

```
[ec2-user ~]$ ip link show eth0
```

請注意，在此範例輸出中，*mtu 9001* 表示此執行個體使用巨型訊框。

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode
 DEFAULT group default qlen 1000
    link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

Windows 執行個體

您使用的程序取決於執行個體上的驅動程式。

ENA driver

2.1.0 版及更新版本

若要取得 MTU 值，請在 EC2 執行個體上使用下列 Get-NetAdapterAdvancedProperty 命令。使用萬用字元 (星號) 取得所有乙太網路名稱。檢查介面名稱 *JumboPacket 的項目。9015 值表示已啟用巨型訊框。巨型訊框預設為停用。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

1.5 版和較舊版本

若要取得 MTU 值，請在 EC2 執行個體上使用下列 Get-NetAdapterAdvancedProperty 命令。檢查介面名稱 MTU 的項目。9001 值表示已啟用巨型訊框。巨型訊框預設為停用。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Intel SRIOV 82599 driver

若要取得 MTU 值，請在 EC2 執行個體上使用下列 Get-NetAdapterAdvancedProperty 命令。檢查介面名稱 *JumboPacket 的項目。9014 值表示已啟用巨型訊框。(請注意，MTU 大小包含標頭和承載。) 巨型訊框預設為停用。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

AWS PV driver

若要取得 MTU 值，請在 EC2 執行個體上使用下列命令。介面名稱可能有所不同。請在輸出中尋找具備名稱 "Ethernet"、"Ethernet 2" 或 "Local Area Connection" 的項目。您將需要介面名稱來啟用或停用巨型訊框。9001 值表示已啟用巨型訊框。

```
netsh interface ipv4 show subinterface
```

為您的執行個體設定 MTU

建議在您 VPC 內的網路流量使用巨型訊框，或在網際網路流量使用標準訊框。無論您的使用案例為何，我們都建議您確認執行個體會依預期運作。

您使用的程序會依執行個體的作業系統而定。

Linux 執行個體

設定 Linux 執行個體上的 MTU 值

1. 在執行個體上執行以下 ip 命令。下列命令會將所需的 MTU 值設定為 1500，但您可改用 9001。如果主要網路介面不是 eth0，請將 eth0 取代為實際網路介面。

   ```
   [ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
   ```

2. (選用) 欲在重新啟動後保留您的網路 MTU 設定，請依據您的作業系統類型修改下列組態檔案。

   - Amazon Linux 2023 – 修改組態檔案的 [Link] 區段。預設組態檔案為 /usr/lib/systemd/network/80-ec2.network，或者可以更新在 /run/systemd/network/ 中建立的任何自訂組態檔案，其中檔案名稱為 *priority-interface*.network。如需詳細資訊，請參閱 Amazon Linux 文件中的聯網服務。

     ```
     MTUBytes=1500
     ```

   - Amazon Linux 2 – 請將下列行新增至 /etc/sysconfig/network-scripts/ifcfg-*eth0* 檔案：

     ```
     MTU=1500
     ```

     將下行新增至 /etc/dhcp/dhclient.conf 檔案：

     ```
     request subnet-mask, broadcast-address, time-offset, routers, domain-name,
      domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-
     servers;
     ```

   - 其他 Linux 發行版本 - 請參閱其特定文件。

3. (可選) 重新啟動執行個體，並確認 MTU 設定正確。

Windows 執行個體

您使用的程序取決於執行個體上的驅動程式。

ENA driver

您可使用裝置管理員或 Set-NetAdapterAdvancedProperty 命令來變更 MTU 設定。

2.1.0 版及更新版本

使用下列命令來啟用巨型訊框。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9015
```

使用下列命令停用。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

1.5 版和較舊版本

使用下列命令來啟用巨型訊框。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 9001
```

使用下列命令停用。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 1500
```

Intel SRIOV 82599 driver

您可使用裝置管理員或 Set-NetAdapterAdvancedProperty 命令來變更 MTU 設定。

使用下列命令來啟用巨型訊框。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

使用下列命令停用。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

AWS PV driver

您可以使用執行個體上的 netsh 命令來變更 MTU。您可使用裝置管理員變更 MTU 設定。

使用下列命令來啟用巨型訊框。

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

使用下列命令停用。

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500
```

疑難排解

使用巨型訊框時，若 EC2 執行個體和 Amazon Redshift 叢集之間出現連線問題，請參閱《Amazon Redshift 管理指南》中的查詢似乎沒有回應且有時無法觸達叢集。

EC2 執行個體的虛擬私有雲

Amazon Virtual Private Cloud (Amazon VPC) 可讓您在 AWS 雲端內的邏輯隔離區域中定義虛擬網路，稱為虛擬私有雲端或 VPC。您可以在 VPC 的子網路中建立 AWS 資源，例如 Amazon EC2 執行個體。VPC 近似於您在自有資料中心內運作的傳統網路，卻能提供 AWS可擴展性基礎設施的效益。您可以設定您的 VPC；您可以選取其 IP 地址範圍、建立子網，以及設定路由表、網路閘道與安全設定。您可以將 VPC 中的執行個體連線至網際網路或連線至自己的資料中心。

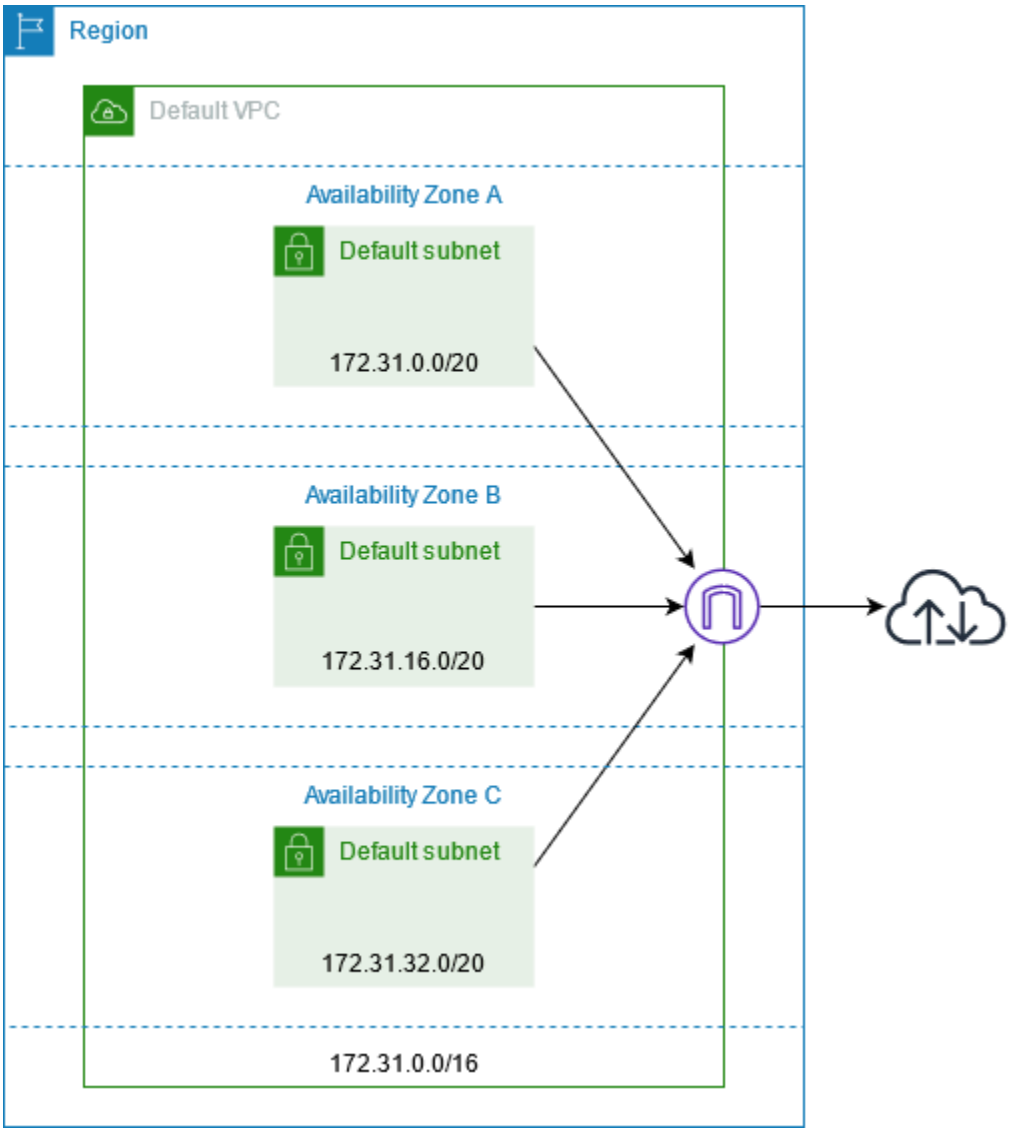
目錄

- 您的預設 VPC
- 非預設 VPC
- 網際網路存取
- 共用子網路

- 僅限 IPv6 的子網路

您的預設 VPC

當您建立 AWS 帳戶時，我們會在每個區域中建立預設 VPC。預設 VPC 是已設定好可供您使用的 VPC。例如，每個預設 VPC 的每個可用區域都有預設子網、連接到 VPC 的網際網路閘道，且在主路由表中有路由，將所有流量 (0.0.0.0/0) 傳送至網際網路閘道。您可以視需要修改預設 VPC 的設定。例如，您可以新增子網路和路由表。



非預設 VPC

您可以建立自己的 VPC，如 Amazon VPC 使用者指南中的建立 VPC 所述，而不是為您的資源使用預設 VPC。

以下是建立 EC2 執行個體 VPC 時需要考慮的一些事項。

- 您可以使用 IPv4 CIDR 區塊的預設建議，或輸入您的應用程式或網路所需的 CIDR 區塊。
- 在多個可用區域中使用子網路，以確保高可用性。
- 如果必須可從網際網路存取執行個體，請執行以下其中一項操作：
  - 如果您的執行個體可以在公有子網路中，請新增公有子網路。維持兩個 DNS 選項為啟用狀態。您可以選擇現在或稍後新增私有子網路。
  - 如果您的執行個體必須位於私有子網路中，則僅新增私有子網路。您可以新增 NAT 閘道，以提供對私有子網路中執行個體的網際網路存取。如果執行個體跨可用區域傳送或接收大量流量，則每個可用區域建立 1 個 NAT 閘道。否則，您可以在其中一個可用區域中建立 NAT 閘道，並在與 NAT 閘道相同的可用區域中啟動傳送或接收跨區域流量的執行個體。

網際網路存取

在預設 VPC 的預設子網路中啟動的執行個體可以存取網際網路，因為預設 VPC 設定為指派公有 IP 位址和 DNS 主機名稱，而主路由表設定為路由至連接到 VPC 的網際網路閘道。

對於您在非預設子網路和 VPC 中建立的執行個體，您可以使用下列任一選項，確保您在這些子網路中啟動的執行個體可以存取網際網路：

- 設定網際網路閘道。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的使用網際網路閘道連接至網際網路。
- 設定公有 NAT 閘道。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的從私有子網存取網際網路。

共用子網路

將 EC2 執行個體啟動到共用 VPC 子網路時，請注意以下事項：

- 參與者可以透過指定共用子網路的 ID，以在共用子網路中執行執行個體。參與者必須擁有他們指定的任何網路介面。
- 參與者可以啟動、停止、終止和描述自己在共用子網路中建立的執行個體。參與者無法啟動、停止、終止或描述 VPC 擁有者在共用子網路中建立的執行個體。
- VPC 擁有者無法啟動、停止、終止或描述參與者在共用子網路中建立的執行個體。

- 參與者可以使用 EC2 Instance Connect Endpoint 連線到共用子網路中的執行個體。參與者必須在共用子網路中建立 EC2 Instance Connect Endpoint。參與者無法使用 VPC 擁有者在共用子網路中建立的 EC2 Instance Connect Endpoint。

如需有關共用 Amazon EC2 資源的詳細資訊，請參閱下列內容：

- the section called “管理與組織或 OU 的 AMI 共用”
- the section called “共用容量保留”
- the section called “共用放置群組”
- 跨帳戶 Amazon EC2 專用主機共用

如需有關共用子網路的詳細資訊，請參閱「Amazon VPC 使用者指南」中的與其他帳戶共用 VPC。

僅限 IPv6 的子網路

在僅限 IPv6 之子網路中啟動的 EC2 執行個體會收到 IPv6 地址，而非 IPv4 地址。您在僅限 IPv6 子網路中啟動的任何執行個體，都必須是 Nitro 型執行個體。

次要網路

次要網路是專用於特殊聯網使用案例的虛擬網路。這些網路在邏輯上隔離在 AWS 雲端的分割區內。您可以建立資源，例如次要網路中的次要子網路。次要網路與 Amazon VPCs 緊密結合，因此特定執行個體會進行多重主目錄，並同時在 VPC 和次要網路中啟動。

次要網路目前適用於特定執行個體類型，以及透過具有長期承諾的容量保留。如果您認為次要網路可能有利於您的工作負載，請聯絡您的客戶團隊以取得更多資訊。

目錄

- 什麼是次要網路？
- 重要概念
- Architecture
- 其他考量
- 開始使用
- 管理次要網路資源
- 網路設計最佳實務
- 疑難排解

- 配額和限制

什麼是次要網路？

次要網路提供邏輯、隔離的網路，與 VPC 網路搭配使用，因此執行個體會多主目錄至兩個獨立網路。次要網路的優點包括：

- 適用於特殊使用案例和通訊協定的高效能聯網，例如 ML 工作負載的東西連線
- 多租戶支援邏輯隔離
- 執行個體與 VPCs AWS 和服務無縫整合

重要概念

次要網路

區域聯網建構，提供具有 IPv4 CIDR 區塊的邏輯第 3 層網路 （ 從 /28 到 /12)。次要網路的運作獨立於實體分割網路基礎設施上的 VPCs。

次要子網路

次要網路中的可用區域特定建構，類似於 VPC 子網路。次要子網路支援從 /28 到 /12 的 CIDR 區塊。

次要界面

連接到次要網路卡的網路介面，在次要子網路中提供東西連線。這些界面在實體和邏輯上都與彈性網路界面 (ENIs) 分開。

Architecture

支援次要網路的 EC2 執行個體是多重主目錄，這表示他們可以同時在 VPC 和次要網路內通訊：

- VPC：為 AWS 服務、儲存、資料庫、聯網服務和網際網路提供南北 TCP/IP 連線
- 次要網路：在支援的專用執行個體之間提供東西連線

其他考量

- 次要介面是透過 RunInstances 管理，無法獨立建立或刪除。

- 執行個體啟動後，無法連接/刪除次要介面。
- 啟動後就無法變更次要介面 IP 地址。
- 次要網路不支援安全群組、NACLs、流程日誌等 VPC 功能。

開始使用

先決條件

使用次要網路啟動執行個體之前，請確定您也已在目標區域中設定 VPC，並在 EC2 容量的目標可用區域中設定子網路。

步驟 1：建立次要網路

在與 VPC 相同的區域中建立次要網路。這是區域資源，可為 RDMA 流量提供邏輯隔離。

```
aws ec2 create-secondary-network \
  --network-type rdma \
  --ipv4-cidr-block 172.31.0.0/16 \
  --region us-east-2
```

參數：

- `--network-type`：網路類型（目前僅支援 rdma)
- `--ipv4-cidr-block`：介於 /28 和 /12 之間的 IPv4 CIDR 區塊
- `--region`： AWS 區域 (US-East-2)

> **Note**
>
> 最佳實務：選擇與您的 VPC CIDR 不重疊的 CIDR 區塊，以簡化執行個體層級的路由。

步驟 2：建立次要子網路

在與 VPC 子網路相同的可用區域中建立次要子網路。這是 AZ 特定的資源。

```
aws ec2 create-secondary-subnet \
  --secondary-network-id sn-1234567890abcdef0 \
  --ipv4-cidr-block 172.31.24.0/24 \
```

```
  --availability-zone us-east-2a
```

> **Note**
>
> IP 地址保留：如同 VPC 子網路，Amazon 會保留每個次要子網路中的前 4 個 IP 地址和最後一個 IP 地址以供內部使用。

步驟 3：啟動執行個體

在 VPC 子網路和次要子網路中啟動執行個體。執行個體將具備與兩個網路連線的多重主目錄。

```
aws ec2 run-instances \
  --image-id ami-12345678 \
  --count 1 \
  --instance-type <instance> \
  --key-name MyKeyPair \
  --instance-market-options '{"MarketType": "capacity-block"}' \
  --capacity-reservation-specification '{"CapacityReservationTarget": \
  {"CapacityReservationId": "cr-1234567890abcdef0"}}' \
  --network-interfaces \
    "NetworkCardIndex=0,DeviceIndex=0,Groups=sg-12345678,\
    SubnetId=subnet-0987654321fedcba0,InterfaceType=interface" \
  --secondary-interfaces \
    "NetworkCardIndex=1,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=2,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=3,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=4,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=5,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=6,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=7,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=8,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true"
```

金鑰參數：

- `--network-interfaces`：指定 VPC 連線的主要 Nitro ENI （網路卡索引 0)
- `--secondary-interfaces`：指定次要子網路中東西連線的 8 個次要介面 （網路卡索引 1-8)
- `InterfaceType=secondary`：表示次要界面

管理次要網路資源

描述次要網路

檢視次要網路的詳細資訊：

```
aws ec2 describe-secondary-networks \
  --secondary-network-id sn-1234567890abcdef0
```

描述次要子網路

檢視次要子網路的詳細資訊：

```
aws ec2 describe-secondary-subnets \
  --secondary-subnet-id ss-98765421yxz
```

描述次要界面

檢視連接至執行個體的次要網路界面詳細資訊：

```
aws ec2 describe-secondary-interfaces \
  --filters "Name=attachment.instance-id,Values=i-1234567890abcdef0"
```

刪除資源

刪除次要子網路：

```
aws ec2 delete-secondary-subnet \
  --secondary-subnet-id ss-98765421yxz
```

刪除次要網路：

```
aws ec2 delete-secondary-network \
  --secondary-network-id sn-1234567890abcdef0
```

> **Important**
>
> 您必須先終止所有執行個體並刪除所有次要子網路，才能刪除次要網路。

網路設計最佳實務

CIDR 規劃

避免重疊 CIDRs：雖然次要網路與 VPCs 實體隔離，但使用非重疊 CIDR 區塊可簡化執行個體作業系統層級的路由組態。

> **Note**
>
> Amazon 會為每個子網路保留 5 個 IP 地址。

流量隔離

依次要網路隔離：為不同的專案、團隊或安全界限建立個別的次要網路。次要網路提供執行個體之間的邏輯隔離。執行個體無法跨不同的次要網路進行通訊。

使用多個子網路：在次要網路中，使用多個次要子網路依 GPU 索引、可用區域或工作負載類型分割流量。例如，常見的架構模式是部署具有 4 或 8 個次要子網路的單一次要網路，其中每個次要子網路都與一組常見索引的 GPUs 對齊。

疑難排解

執行個體啟動失敗

問題：執行個體無法使用次要網路介面啟動。

解決方案：

- 確認您的 AMI 包含適當的驅動程式支援
- 確保您的次要子網路具有足夠的可用 IP 地址
- 確認您的容量保留處於「作用中」狀態
- 檢查您的次要子網路是否與 VPC 子網路位於相同的可用區域

連線問題

問題：無法在執行個體之間建立 RDMA 連線。

解決方案：

- 確認所有執行個體都位於相同的次要網路和次要子網路中
- 檢查次要介面驅動程式是否已正確載入執行個體
- 確保您的應用程式已繫結至正確的網路介面
- 可透過直接路由存取相同次要子網路內的執行個體。跨子網路通訊可透過 DHCP 提供的靜態路由進行。

API 錯誤

問題：次要網路和次要子網路操作的 API 呼叫失敗。

解決方案：

- 驗證 ec2：CreateSecondaryNetwork、ec2：CreateSecondarySubnet 的 IAM 許可。
- 檢查 CIDR 區塊是否在支援的範圍內 (/28 到 /12)
- 確認您使用的是正確的區域和可用區域

配額和限制

若要請求提高配額，請使用 AWS Service Quotas或聯絡 AWS Support。

次要網路配額和限制

資源	限制	可調整
每個區域的次要網路	5	是
每個次要網路的次要子網路	200	是
CIDR 區塊大小	/28 到 /12	否

Amazon EC2 中的安全

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

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

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 Amazon EC2 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端內部的安全 – 您的責任包含下列領域：
  - 控制對執行個體的網路存取，例如：透過設定 VPC 和安全群組。如需詳細資訊，請參閱 控制網路流量。
  - 管理用於連線到執行個體的登入資料。
  - 管理部署至訪客作業系統的訪客作業系統和軟體，包括更新和安全性修補程式。如需詳細資訊，請參閱 Amazon EC2 執行個體的更新管理。
  - 設定連接至執行個體的 IAM 角色以及與這些角色相關聯的許可。如需詳細資訊，請參閱 Amazon EC2 的 IAM 的角色。

本文件有助於您了解如何在使用 Amazon EC2 時套用共同責任模型。它會示範如何設定 Amazon EC2 以符合您的安全性和合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 Amazon EC2 資源。

目錄

- Amazon EC2 中的資料保護
- Amazon EC2 中的基礎設施安全
- Amazon EC2 的復原功能
- Amazon EC2 的合規驗證
- Amazon EC2 的 Identity and Access Management
- Amazon EC2 執行個體的更新管理
- Windows 執行個體的安全性最佳做法
- Amazon EC2 金鑰對和 Amazon EC2 執行個體
- EC2 執行個體的 Amazon EC2 安全群組

- Amazon EC2 執行個體的 NitroTPM
- Amazon EC2 執行個體驗證
- Windows 執行個體的 Credential Guard
- 使用介面 VPC 端點存取 Amazon EC2

Amazon EC2 中的資料保護

共同責任模型適用於 AWS Amazon Elastic Compute Cloud 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。 如需歐洲資料保護的詳細資訊，請參閱一般資料保護規則 (GDPR) 中心。

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

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

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 Amazon EC2 或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

目錄

- Amazon EBS 資料安全
- 靜態加密
- 傳輸中加密

Amazon EBS 資料安全

Amazon EBS 磁碟區是以原始、未格式化的區塊型儲存設備型式提供給您。這些裝置是在 EBS 基礎設施上建立的邏輯裝置，Amazon EBS 服務可確保裝置在客戶進行任何使用或重複使用之前在邏輯上是空的 (也就是說，原始區塊為零或其包含加密虛擬隨機資料)。

若您的程序要求在使用後或使用前 (或兩者)，使用特定方法清除所有資料，例如 DoD 5220.22-M (國家工業安全計畫操作手冊) 或 NIST 800-88 (媒體清理準則)，您可在 Amazon EBS 上執行此作業。該區塊層級的活動將反映至 Amazon EBS 服務中的基礎儲存媒體。

靜態加密

EBS 磁碟區

Amazon EBS 加密是適用於 EBS 磁碟區和快照的加密解決方案。它使用 AWS KMS keys。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 加密。

[Windows 執行個體] 您也可以使用 Microsoft EFS 和 NTFS 許可，進行資料夾和檔案層級加密。

執行個體儲存體磁碟區

NVMe 執行個體儲存體磁碟區上的資料會以執行個體上的硬體模組中實作的 XTS-AES-256 區塊編碼器來加密。用於加密寫入本機連接 NVMe 儲存裝置之資料的金鑰按照每個客戶和每個磁碟區提供。這些金鑰由硬體模組產生且僅駐留在其中， AWS 人員無法存取此模組。這些加密金鑰會在執行個體停止或終止時銷毀，且無法復原。您無法停用此加密，也無法提供您自己的加密金鑰。

H1、D3 和 D3en 執行個體上的 HDD 執行個體儲存體磁碟區中的資料會使用 XTS-AES-256 和一次性金鑰來加密。

當您讓執行個體停止、休眠或終止時，執行個體儲存體磁碟區中的所有儲存區塊都會重設。因此，資料無法透過另一個執行個體的執行個體儲存體存取。

記憶體

下列執行個體會啟用記憶體加密：

- 具有 AWS Graviton2 或更新版本 AWS Graviton 處理器的執行個體支援永遠開啟的記憶體加密。加密金鑰會在主機系統內安全地產生，不能離開主機系統，並在主機重新啟動或關閉電源時銷毀。如需詳細資訊，請參閱 AWS Graviton 處理器。
- 搭載第三代 Intel Xeon 可擴充處理器 (Ice Lake) 的執行個體 (例如 M6i 執行個體) 以及搭載第四代 Intel Xeon 可擴充處理器 (Sapphire Rapids) 的執行特體 (例如 M7i 執行個體)。這些處理器支援使用 Intel 總記憶體加密 (TME) 的永遠開啟記憶體加密。

- 搭載第三代 AMD EPYC 處理器 (Milan) 的執行個體 (例如 M6a 執行個體) 以及搭載第四代 AMD EPYC 處理器 (Genoa) 的執行個體 (例如 M7a 執行個體)。這些處理器使用 AMD 安全記憶體加密 (SME)，可支援永遠開啟的記憶體加密。
- AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) 支援某些 AMD 型執行個體類型。如需詳細資訊，請參閱尋找支援 AMD SEV-SNP 的 EC2 執行個體類型。

傳輸中加密

在實體層加密

在離開 AWS 安全設施之前，透過 AWS 全球網路跨 AWS 區域傳輸的所有資料都會在實體層自動加密。AZ 之間的所有流量都會加密。額外的加密層 (包括本節所列的加密層) 可能會提供額外的保護。

Amazon VPC 對等互連和 Transit Gateway 跨區域對等互連提供的加密

所有使用 Amazon VPC 對等互連和 Transit Gateway 對等互連的跨區域流量都會在離開區域時自動進行批量加密。在離開 AWS 安全設施之前，會在實體層為所有流量自動提供額外的加密層，如本節先前所述。

執行個體間的加密

AWS 可在所有類型的 EC2 執行個體之間提供安全且私有的連線。此外，某些執行個體類型使用基礎 Nitro System 硬體的卸載功能，以自動加密執行個體之間的傳輸中流量。此加密機制使用帶有關聯資料的認證加密 (AEAD) 演算法 (採用 256 位元加密)。這對網路效能沒有影響。若要支援執行個體之間額外的傳輸中流量加密，必須符合下列要求：

- 執行個體使用下列執行個體類型：
  - 一般用途：
    M5dn、M5n、M5zn、M6a、M6i、M6id、M6idn、M6in、M7a、M7g、M7gd、M7i、M7i-flex、M8a、M8azn、M8g、M8gb、M8gd、M8gn、M8i、M8id、M8i-flex、M8in、M8idn、M8ine、M8ib、M8idb、M9g、M9gd、Mac-m4、Mac-m4pro
  - 運算最佳化：C5n、C6a、C6gn、C6i、C6id、C6in、C7a、C7g、C7gd、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gd、C8gn、C8i、C8id、C8i-flex、C8in、C8ine、C8ib、C9g、C9gd
  - 記憶體最佳化：R5dn、R5n、R6a、R6i、R6id、R6idn、R6in、R7a、R7g、R7gd、R7i、R7iz、R8a、R8g、R8gb flex、R8in、R8idn、R8ib、R8idb、R9g、R9gd、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb U7i-12tb U7in-16tb U7in-24tb U7in-32tb U7inh-32tb X2idn X2iedn X2iezn X8g X8aedz X8i

  - 儲存最佳化：D3、D3en、I3en、I4g、I4i、I7i、I7ie、I8g、I8ge、Im4gn、Is4gen
  - 加速運算：DL1、DL2q、F2、G4ad、G4dn、G5、G6、G6e、G6f、Gr6、Gr6f、G7、G7e、Inf1、Inf2、P3dn、P4d、P4de、P5、P5e、P5en、P6-B200、P6-B300、P6e-GB200、Trn1、Trn1n、Trn2、Trn2u、VT1
  - 高效能運算：Hpc6a、Hpc6id、Hpc7a、Hpc7g、Hpc8a
- 這些執行個體位於相同的區域中。
- 這些執行個體位於相同的 VPC 或對等 VPC 中，且流量不會經過虛擬網路裝置或服務，例如負載平衡器或傳輸閘道。

在離開 AWS 安全設施之前，會在實體層為所有流量自動提供額外的加密層，如本節先前所述。

若要檢視使用 加密執行個體之間傳輸中流量的執行個體類型 AWS CLI

使用下列 describe-instance-types 命令。

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

加密往返 AWS Outposts

Outpost 會建立稱為服務連結的特殊網路連線，以連接至其 AWS 主要區域，並選擇性地建立與您指定之 VPC 子網路的私有連線。透過這些連線的所有流量都會完整加密。如需詳細資訊，請參閱 AWS Outposts 使用者指南中的透過服務連結連線和傳輸中加密。

遠端存取加密

SSH 及 RDP 通訊協定為遠端存取執行個體提供安全的通訊頻道，無論是直接存取還是透過 EC2 Instance Connect。使用 AWS Systems Manager Session Manager 或 Run Command 對執行個體的遠端存取會使用 TLS 1.2 加密，而建立連線的請求會使用 SigV4 簽署，並由 驗證和授權AWS Identity and Access Management。

使用諸如 Transport Layer Security (TLS) 等加密通訊協定是您的責任，以便加密在用戶端和您的 Amazon EC2 執行個體之間傳輸的敏感資料。

(Windows 執行個體) 務必只允許 EC2 執行個體與 AWS API 端點或其他敏感的遠端網路服務之間的加密連線。您可以透過傳出安全群組或 Windows 防火牆規則來強制執行此操作。

Amazon EC2 中的基礎設施安全

Amazon Elastic Compute Cloud 是受管服務，受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon EC2。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

如需詳細資訊，請參閱 Security Pillar – AWS Well-Architected Framework 中的基礎設施保護。

網路隔離

虛擬私有雲端 (VPC) 是 AWS 雲端中您自己的邏輯隔離區域中的虛擬網路。使用不同的 VPC，依工作負載或組織實體來隔離基礎設施。

子網是您的 VPC 中的 IP 地址範圍。啟動執行個體時，您會在 VPC 的子網中啟動它。使用子網來隔離單一 VPC 內的應用程式層 (例如，Web、應用程式及資料庫)。如果不應該從網際網路直接存取，則針對您的執行個體使用私有子網。

若要使用私有 IP 地址從 VPC 呼叫 Amazon EC2 API，請使用 AWS PrivateLink。如需詳細資訊，請參閱使用介面 VPC 端點存取 Amazon EC2。

實體主機上的隔離

已經將相同實體主機上的不同 EC2 執行個體彼此隔離，如同他們位於不同實體主機一樣。Hypervisor 會隔離 CPU 和記憶體，而且執行個體式提供的虛擬化磁碟，而非存取原始磁碟裝置。

當您停止或終止執行個體時，Hypervisor 會先清除配置到該執行個體的記憶體 (設定為零)，然後再將之配置到新執行個體，而且會重設儲存體的每個區塊。這可確保不會蓄意地向其他執行個體公開您的資料。

網路基礎設施會將 AWS 網路 MAC 地址動態指派給執行個體。IP 地址會由 AWS 網路基礎設施動態地指派給執行個體，或是由 EC2 管理員透過驗證的 API 請求進行指派。 AWS 網路允許執行個體僅從指派給他們的 MAC 和 IP 地址傳送流量。否則，流量會遭到捨棄。

依預設，執行個體無法接收不是特別定址給它的流量。如果您需要在執行個體上執行網路位置轉譯 (NAT)、路油或防火牆服務，您可以針對網路介面停用來源/目的地檢查。

控制網路流量

請考慮下列選項來控制 EC2 執行個體的網路流量：

- 使用安全性群組限制您執行個體的存取權限。設定允許最低所需網路流量的規則。例如，您可以允許僅來自於企業網路位址範圍或特定協定 (如 HTTPS) 的流量。對於 Windows 執行個體，允許 Windows 管理流量和最少輸出連線。
- 運用安全群組作為控制 Amazon EC2 執行個體網路存取的主要機制。必要時，請謹慎使用網路 ACL 來提供無狀態、粗糙的網路控制。安全群組比網路 ACL 更多用途，因為它們能夠執行有狀態封包篩選，並建立參考其他安全群組的規則。不過，網路 ACL 可以有效地作為次要控制項，以拒絕特定流量子集或提供高階子網防護護欄。此外，因為網路 ACL 會套用至整個子網，所以執行個體若在沒有正確安全群組的情況下意外啟動，它們可以用作深度防禦。
- [Windows 執行個體] 使用群組政策物件 (GPO) 集中管理 Windows 防火牆設定，可進一步增強網路控制。客戶通常會使用 Windows 防火牆進一步了解網路流量，以及補充安全群組篩選條件，以建立進階規則以封鎖特定應用程式存取網路，或篩選來自子集 IP 地址的流量。例如，Windows 防火牆可以限制只有特定使用者或應用程式才能存取 EC2 中繼資料服務 IP 地址。或者，面向公眾的服務可能會使用安全群組來限制特定連接埠的流量，並使用 Windows 防火牆來維護明確封鎖之 IP 地址的清單。
- 如果不應該從網際網路直接存取，則針對您的執行個體使用私有子網。使用堡壘主機或 NAT 閘道，以取得來自私人子網中執行個體的網際網路存取權限。
- [Windows 執行個體] 使用安全管理通訊協定，例如透過 SSL/TLS 進行 RDP 封裝。遠端桌面閘道 Quick Start 提供部署遠端桌面閘道的最佳實務，包括將 RDP 設定為使用 SSL/TLS。
- 【Windows 執行個體】 使用 Active Directory 或 Directory Service 來緊密且集中地控制和監控互動式使用者和群組對 Windows 執行個體的存取，並避免本機使用者許可。亦請避免使用網域管理員，而是建立更精細、以應用程式特定角色為基礎的帳戶。Just Enough Administration (JEA) 允許管理 Windows 執行個體的變更，無需互動或管理員存取。此外，JEA 還可讓組織鎖定執行個體管理所需之 Windows PowerShell 命令子集的管理存取權。如需詳細資訊，請參閱安全性、身分和合規的最佳實務。
- [Windows 執行個體] 系統管理員應該使用 Windows 帳戶搭配有限存取權來執行日常活動，並且只在必要時才提升存取權，以執行特定的組態變更。此外，只有在絕對必要時才能直接存取 Windows 執行個體。反之，運用 EC2 執行命令、系統中心組態管理員 (SCCM)、Windows PowerShell DSC 或 Amazon EC2 Systems Manager (SSM) 這類的中央組態管理系統，將變更推送至 Windows 伺服器。

- 設定必要網路路由最少的 Amazon VPC 子網路由表。例如，只將要求直接存取網際網路的 Amazon EC2 執行個體放入路由至網際網路閘道的子網路中，並僅將需要直接存取內部網路的 Amazon EC2 執行個體放入路由至虛擬私有閘道的子網路中。
- 請考慮使用其他安全群組或網路界面，來控制和稽核 Amazon EC2 執行個體管理流量，並與一般應用程式流量分開。這種方法可讓客戶實作變更控制的特殊 IAM 政策，讓稽核安全群組規則或自動規則驗證指令碼的變更變得更輕鬆。使用多個網路介面也提供其他控制網路流量的選項，包括能夠建立以主機為基礎的路由政策，或能夠根據網路介面的指定子網路運用不同的 VPC 子網路路由規則。
- 使用 AWS Virtual Private Network 或 Direct Connect 建立從遠端網路到 VPCs私有連線。如需詳細資訊，請參閱 Network-to-Amazon VPC 連線選項。
- 使用 VPC 流程日誌監控到達您執行個體的流量。
- 使用 GuardDuty 惡意軟體防護來識別執行個體上可能危害工作負載的惡意軟體之可疑行為、重新利用資源進行惡意使用，以及未經授權存取您的資料。
- 使用 GuardDuty 執行時期監控來識別和回應針對執行個體的潛在威脅。如需詳細資訊，請參閱執行時期監控如何搭配 Amazon EC2 執行個體運作。
- 使用 AWS Security Hub CSPM、Reachability Analyzer 或網路存取分析器來檢查執行個體中是否有非預期的網路可存取性。
- 使用 EC2 Instance Connect 連線至您的執行個體，其會使用 Secure Shell (SSH)，而不需要分享和管理 SSH 金鑰。
- 使用 AWS Systems Manager Session Manager 遠端存取您的執行個體，而不是開啟輸入 SSH 或 RDP 連接埠，以及管理金鑰配對。
- 使用AWS Systems Manager 執行命令來自動執行常見的管理任務，而不是連線至您的執行個體。
- [Windows 執行個體] 許多 Windows 作業系統角色和 Microsoft 商務應用程式也提供增強的功能，例如 IIS 內的 IP 位址範圍限制、Microsoft SQL Server 中的 TCP/IP 篩選政策，以及 Microsoft Exchange 中的連線篩選政策。應用程式層內的網路限制功能可為重要的商務應用程式伺服器提供額外的防禦層。

Amazon VPC 支援其他網路安全性控制，例如閘道、代理伺服器和網路監控選項。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的控制網路流量。

Amazon EC2 的復原功能

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。透過可用區域，您可以設計與操作的應用

程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如果您需要在更大的地理距離內複製資料或應用程式，請使用 AWS Local Zones。Local AWS Zone 是 AWS 區域的延伸，其地理位置接近您的使用者。Local Zones 有自己的網際網路連線，並支援 Direct Connect。如同所有 AWS 區域， AWS 本機區域與其他區域完全隔離 AWS 。

如果您需要在 AWS 本機區域中複寫資料或應用程式， AWS 建議您使用下列其中一個區域做為容錯移轉區域：

- 另一個 Local Zone
- 區域中不是父區域的可用區域。您可以使用 describe-availability-zones 命令來檢視父區域。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

除了 AWS 全球基礎設施之外，Amazon EC2 還提供下列功能來支援您的資料彈性：

- 跨區域複製 AMI
- 跨區域複製 EBS 快照
- 使用 Amazon Data Lifecycle Manager 自動化 EBS 後端 AMI
- 使用 Amazon Data Lifecycle Manager 自動化 EBS
- 使用 Amazon EC2 Auto Scaling 維護機群的運作狀態及可用性
- 使用 Elastic Load Balancing 將輸入流量分配到單一可用區域或多個可用區域中的多個執行個體。

Amazon EC2 的合規驗證

我們的新 AWS 註冊體驗並非針對受管制的工作負載而設計。如果您正在使用我們新的 AWS 註冊體驗，但想要將 AWS 用於受管制的工作負載，您可以註冊 AWS （進階） 或為您的 AWS 環境啟用進階功能。

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 AWS 安全文件。

Amazon EC2 的 Identity and Access Management

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制哪些人員可進行身分驗證 (登入) 並獲得授權 (具有許可) 以使用 Amazon EC2 資源。IAM 是您可以免費使用 AWS 服務 的 。

您的安全登入資料會將您識別為 中的 服務， AWS 並授予您存取 AWS 資源的權限，例如 Amazon EC2 資源。您可使用 Amazon EC2 和 IAM 的功能來允許其他使用者、服務和應用程式使用您的 Amazon EC2 資源，但不共用您的安全憑證。您可以使用 IAM 來控制其他使用者如何使用 中的資源 AWS 帳戶，也可以使用安全群組來控制對 Amazon EC2 執行個體的存取。您可選擇授予 Amazon EC2 資源的完全使用或有限制使用的權限。

如果您是開發人員，您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式所需的安全憑證。將 IAM 角色連接至執行個體後，在執行個體上執行的應用程式可以從執行個體中繼資料服務 (IMDS) 擷取登入資料。

如需使用 IAM 保護您的 AWS 資源的最佳實務，請參閱《IAM 使用者指南》中的 IAM 中的安全最佳實務。

目錄

- Amazon EC2 身分型政策
- 控制 Amazon EC2 API 存取的政策範例
- 控制 Amazon EC2 主控台存取的政策範例
- AWS Amazon EC2 的 受管政策
- Amazon EC2 的 IAM 的角色

Amazon EC2 身分型政策

依預設，使用者無權建立或修改 Amazon EC2 資源，或使用 Amazon EC2 API、Amazon EC2 主控台或 CLI 執行任務。若要允許使用者建立或修改資源並執行任務，您必須建立授予使用者許可的 IAM 政策，以使用他們需要的特定資源和 API 動作，然後將這些政策連接到需要這些許可的使用者、群組或 IAM 角色。

將政策連接到使用者、使用者群組或角色時，政策會允許或拒絕使用者在特定資源上執行特定任務的許可。如需 IAM 政策的一般詳細資訊，請參閱「IAM 使用者指南」中 IAM 中的政策和許可。如需管理及建立 IAM 政策的詳細資訊，請參閱管理 IAM 政策。

IAM 政策必須授予或拒絕使用一或多個 Amazon EC2 動作的許可。政策中還必須指定可用於動作的資源，可為所有資源，或在某些案例中為特定的資源。政策也可以包含您套用到資源的條件。

若要開始使用，您可以檢查 Amazon EC2 的 AWS 受管政策是否符合您的需求。否則，您可以建立自己的自訂政策。如需詳細資訊，請參閱the section called “AWS 受管政策”。

目錄

- 政策語法
- Amazon EC2 動作
- Amazon EC2 API 動作支援的資源層級許可
- 適用於 Amazon EC2 的 Amazon Resource Name (ARN)
- Amazon EC2 的條件金鑰
- 使用屬性型存取控制存取
- 向使用者、群組和角色授予許可
- 檢查使用者是否擁有必要的許可

政策語法

IAM 政策為包含一或多個陳述式的 JSON 文件。每個陳述式的結構如下所示。

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

陳述式由各種元素組成：

- Effect (效果)：效果 可以是 `Allow` 或 `Deny`。根據預設， 使用者沒有使用資源和 API 動作的許可，因此所有請求均會遭到拒絕。明確允許覆寫預設值。明確拒絕覆寫任何允許。

- Action (動作)：動作 是您授予或拒絕許可的特定 API 動作。若要了解指定動作，請參閱Amazon EC2 動作。
- Resource (資源)：受動作影響的資源。有些 Amazon EC2 API 動作可讓您在政策中包含可由動作建立或修改的特定資源。您可以使用 Amazon Resource Name (ARN) 來指定資源，或是使用萬用字元 (*) 來指定陳述式套用到所有資源。如需詳細資訊，請參閱Amazon EC2 API 動作支援的資源層級許可。
- Condition (條件)：條件為選擇性。您可以使用它們來控制何時政策開始生效。如需指定 Amazon EC2 條件的詳細資訊，請參閱 Amazon EC2 的條件金鑰。

如需這些欄位的詳細資訊，請參閱「IAM 使用者指南」中的 IAM JSON 政策元素參考。如需 Amazon EC2 的範例 IAM 政策陳述式，請參閱 控制 Amazon EC2 API 存取的政策範例。

Amazon EC2 動作

在 IAM 政策陳述式中，您可以從任何支援 IAM 的服務指定任何 API 動作。針對 Amazon EC2，請在 API 動作名稱使用下列字首：`ec2:`。例如：`ec2:RunInstances` 和 `ec2:CreateImage`。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

```
"Action": ["ec2:action1", "ec2:action2"]
```

您也可以使用萬用字元指定多個動作。例如，您可以指定名稱開頭有「Describe」文字的所有動作，如下所示：

```
"Action": "ec2:Describe*"
```

> **Note**
>
> 目前，Amazon EC2 Describe* API 動作並不支援資源層級許可。如需 Amazon EC2 資源層級許可的相關詳細資訊，請參閱 Amazon EC2 身分型政策。

若要指定所有的 Amazon EC2 API 動作，請使用 * 萬用字元，如下所示：

```
"Action": "ec2:*"
```

若要查看 Amazon EC2 動作的清單，請參閱服務授權參考中的 Amazon EC2 定義的動作。

Amazon EC2 API 動作支援的資源層級許可

資源層級許可能夠讓您指定使用者可執行動作的資源。Amazon EC2 支援部分的資源層級許可。這表示針對特定 Amazon EC2 動作，您可以根據應滿足的條件，或允許使用者使用特定的資源，來控制使用者何時可以使用那些動作。例如，您可以授予使用者啟動執行個體的許可，但僅限特定類型，且僅能藉由特定的 AMI 來達成。

若要在 IAM 政策陳述式中指定資源，請使用它的 Amazon Resource Name (ARN)。如需指定 ARN 值的詳細資訊，請參閱 適用於 Amazon EC2 的 Amazon Resource Name (ARN)。如果 API 動作不支援個別 ARN，您必須使用萬用字元 (*) 來指定該動作可以影響所有資源。

若要查看識別哪些 Amazon EC2 API 動作支援資源層級許可的表格，以及您可以在政策中使用的 ARN 和條件金鑰，請參閱 動作、資源與條件金鑰 Amazon EC2。

您可以在用於 Amazon EC2 API 動作的 IAM 政策中，套用以標籤為基礎的資源層級許可。這可讓您更有效地控制使用者可以建立、修改或使用的資源。如需詳細資訊，請參閱在建立期間授予標籤 Amazon EC2 資源的許可。

適用於 Amazon EC2 的 Amazon Resource Name (ARN)

每個 IAM 政策陳述式都會套用到您使用其 ARN 指定的資源。

ARN 一般的語法如下所示：

```
arn:aws:[service]:[region]:[account-id]:resourceType/resourcePath
```

服務

服務 (例如，ec2)。

region

資源的區域 (例如，us-east-1)。

account-id

AWS 帳戶 ID，不含連字號 （ 例如 123456789012)。

resourceType

資源類型 (例如，instance)。

resourcePath

識別資源的路徑。您可以在路徑中使用 * 萬用字元。

例如，您可以在陳述式中使用特定執行個體 (i-1234567890abcdef0) 的 ARN 來加以指定，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

您也可以使用 * 萬用字元指定所有屬於特定帳戶的執行個體，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

您也可以使用 * 萬用字元指定屬於特定帳戶的所有 Amazon EC2 資源，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*"
```

若要指定所有資源，或如果特定的 API 動作不支援 ARN，請在 Resource 元素中使用 * 萬用字元，如下所示。

```
"Resource": "*"
```

許多 Amazon EC2 API 動作都涉及多個資源。例如，AttachVolume 會將 Amazon EBS 磁碟區連接至執行個體，所以使用者必須具備該磁碟區與執行個體的使用許可。若要在單一陳述式中指定多項資源，請用逗號分隔其 ARN，如下所示。

```
"Resource": ["arn1", "arn2"]
```

如需 Amazon EC2 資源的 ARN 清單，請參閱 由 Amazon EC2 定義的資源類型 。

Amazon EC2 的條件金鑰

在政策陳述式中，您可以選擇性的指定生效時發揮控制效果的條件。每個條件都包含一或多個索引鍵/值對。條件鍵不區分大小寫。我們已定義 AWS 全域條件索引鍵，以及額外的服務特定條件索引鍵。

如需 Amazon EC2 的服務特定條件金鑰清單，請參閱 Amazon EC2 的條件金鑰。Amazon EC2 也會實作 AWS 全域條件金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的可用於所有請求的資訊。

所有 Amazon EC2 操作都支援 aws:RequestedRegion 和 ec2:Region 條件索引鍵。如需詳細資訊，請參閱範例：限制特定區域的存取。

若要在 IAM 政策中使用條件金鑰，請使用 Condition 陳述式。例如，下列政策會授與使用者新增及移除任何安全群組之傳入和傳出規則的許可。它會使用 ec2:Vpc 條件金鑰，來指定這些動作只能在特定 VPC 中的安全群組上執行。

JSON

```
{
"Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action": [
       "ec2:AuthorizeSecurityGroupIngress",
       "ec2:AuthorizeSecurityGroupEgress",
       "ec2:RevokeSecurityGroupIngress",
       "ec2:RevokeSecurityGroupEgress"],
     "Resource": "arn:aws:ec2:us-east-1:111122223333:security-group/*",
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333:vpc/
vpc-11223344556677889"
        }
      }
    }
  ]
}
```

若您指定多個條件，或在單一條件中指定多個索引鍵，我們會使用邏輯 AND 操作加以評估。若您針對單一索引鍵使用多個值指定單一條件，我們會使用邏輯 OR 操作評估條件。若要授予許可，必須符合所有的條件。

您也可以在指定條件時使用預留位置。如需詳細資訊，請參閱「IAM 使用者指南」中的 IAM 政策元素：變數和標籤。

> **Important**
>
> 許多條件鍵均為資源專屬，而且部分 API 動作使用多個資源。若您撰寫的政策具備條件鍵，請使用陳述式的 Resource 元素來指定套用該條件鍵的資源。若未這麼做，該政策可能會導致使

用者完全無法執行該動作，因為未套用條件金鑰之資源的條件檢查會失敗。若您不希望指定資源，或者若已撰寫政策的 `Action` 元素來納入多個 API 動作，則您必須使用 `...IfExists` 條件類型，以確保未使用該條件金鑰的資源，忽略該條件金鑰。如需詳細資訊，請參閱IAM 使用者指南中的 ...IfExists 條件。

條件索引鍵

- ec2:Attribute 條件金鑰
- ec2:ResourceID 條件索引鍵
- ec2:SourceInstanceARN 條件金鑰

ec2:Attribute 條件金鑰

`ec2:Attribute` 條件金鑰可用於按資源屬性篩選存取的條件。

條件金鑰僅支援屬於基本資料類型 (例如字串或整數) 的屬性，或僅具有 Value 屬性 (例如 ModifyImageAttribute API 動作的 Description 或 ImdsSupport 物件) 的複雜 AttributeValue 物件。條件金鑰不能用於具有多個屬性的複雜物件，例如 ModifyImageAttribute API 動作的 LaunchPermission 物件。

例如，以下政策使用 `ec2:Attribute/Description` 條件金鑰，按 ModifyImageAttribute API 動作的複雜 Description (描述) 物件篩選存取。條件金鑰僅允許將映像描述修改為 `Production` 或 `Development` 的請求。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute/Description": [
            "Production",
            "Development"
```

```
          ]
        }
      }
    }
  ]
}
```

以下範例政策使用 ec2:Attribute 條件金鑰，按 ModifyImageAttribute API 動作的基本 Attribute (屬性) 屬性篩選存取。條件金鑰會拒絕嘗試修改映像描述的所有請求。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute": "Description"
        }
      }
    }
  ]
}
```

ec2:ResourceID 條件索引鍵

使用以下 ec2:*Resource*ID 條件索引鍵搭配指定的 API 動作時，條件索引鍵值會用於指定由 API 動作建立之產生的資源。ec2:*Resource*ID 條件索引鍵不能用於指定 API 請求中指定的來源資源。如果使用下列其中一項 ec2:*Resource*ID 條件索引鍵搭配指定的 API，您必須一律指定萬用字元 (*)。如果指定不同的值，則該條件在執行期期間一律會解析為 *。例如，若要使用 ec2:ImageId 條件索引鍵搭配 CopyImage API，您必須按如下方式指定條件索引鍵：

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:ImageID": "*"
        }
      }
    }
  ]
}
```

我們建議您避免使用 條件索引鍵搭配這些 API 動作。

- ec2:DhcpOptionsID – CreateDhcpOptions
- ec2:ImageID–CopyImage、CreateImage、ImportImage 和 RegisterImage
- ec2:InstanceID–RunInstances 和 ImportInstance
- ec2:InternetGatewayID – CreateInternetGateway
- ec2:NetworkAclID – CreateNetworkAcl
- ec2:NetworkInterfaceID – CreateNetworkInterface
- ec2:PlacementGroupName – CreatePlacementGroup
- ec2:RouteTableID – CreateRouteTable
- ec2:SecurityGroupID – CreateSecurityGroup
- ec2:SnapshotID–CopySnapshot、CreateSnapshot、CreateSnapshots 和 ImportSnapshots
- ec2:SubnetID – CreateSubnet
- ec2:VolumeID–CreateVolume 和 ImportVolume
- ec2:VpcID – CreateVpc
- ec2:VpcPeeringConnectionID – CreateVpcPeeringConnection

相反地，如果需要根據特定資源 ID 篩選存取權，建議您按如下方式使用 Resource 政策元素：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-01234567890abcdef"
    }
  ]
}
```

ec2:SourceInstanceARN 條件金鑰

使用 ec2:SourceInstanceARN 指定提出請求之執行個體的 ARN。這是一個AWS 全域條件金鑰，這表示您可以將它與 Amazon EC2 以外的服務搭配使用。如需政策範例，請參閱範例：允許特定執行個體檢視其他服務中的資源 AWS。

使用屬性型存取控制存取

當您建立授予使用者使用 EC2 資源之許可的 IAM 政策時，您可以在政策的 Condition 元素中包含標籤資訊，以根據標籤控制存取。這稱為以屬性型存取控制 (ABAC)。ABAC 對於使用者可以修改、使用或刪除哪些資源提供了更佳的控制。如需詳細資訊，請參閱 AWS的 ABAC 是什麼？

例如：您可以建立一個政策，允許使用者終止執行個體，但如果執行個體具有標籤 environment=production，則拒絕該動作。若要這樣做，您可以使用 aws:ResourceTag 條件金鑰，根據連接至資源的標籤允許或拒絕存取資源。

```
"StringEquals": { "aws:ResourceTag/environment": "production" }
```

若要了解 Amazon EC2 API 動作是否支援使用 aws:ResourceTag 條件金鑰控制存取，請參閱 Amazon EC2 的動作、資源和條件金鑰。由於 Describe 動作不支援資源層級許可，您必須在不同的陳述式中指定它們，無需條件。

如需 IAM 政策的範例，請參閱控制 Amazon EC2 API 存取的政策範例。

如果您允許或拒絕使用者根據標籤存取資源，請務必考慮明確拒絕使用者將這些標籤新增至相同資源或從中移除的能力。否則，使用者可能透過修改標籤來避開您的限制，並取得資源的存取。

向使用者、群組和角色授予許可

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

檢查使用者是否擁有必要的許可

建立 IAM 政策之後，我們建議您檢查該政策是否授予使用者許可，允許他們使用所需要的特定 API 動作和資源，再讓政策生效。

首先，建立一個用於測試的使用者，然後將建立的 IAM 政策連接到該測試使用者。接著，以測試使用者的身分提出請求。

若您要測試的 Amazon EC2 動作會建立或修改資源，您應使用 `DryRun` 參數進行請求 (或使用 `--dry-run` 選項執行 AWS CLI 命令)。在此案例中，呼叫會完成授權檢查，但不會完成操作。例如，您可以檢查使用者是否可以終止特定執行個體，而無須實際終止它。若測試使用者具有必要的許可，請求會傳回 `DryRunOperation`。否則，會傳回 `UnauthorizedOperation`。

如果政策未授予使用者預期的許可，或授予過多許可，您可以視需要調整政策並重新測試，直到您取得所要的結果。

> **Important**
>
> 政策變更的散佈可能需要幾分鐘時間才能生效。因此，我們建議您等待五分鐘，然後再測試您的政策更新。

如果授權檢查失敗，請求將傳回包含診斷資訊的編碼訊息。您可使用 `DecodeAuthorizationMessage` 動作將訊息解碼。如需詳細資訊，請參閱 AWS Security Token Service API 參考中的 DecodeAuthorizationMessage 以及 decode-authorization-message。

控制 Amazon EC2 API 存取的政策範例

您可以使用 IAM 政策向使用者授予使用 Amazon EC2 所需的權限。如需逐步說明，請參閱「IAM 使用者指南」中的建立 IAM 政策。

以下範例顯示了可用於授予使用者使用 Amazon EC2 的權限的政策陳述式。這些政策專為使用 AWS CLI 或 AWS SDK 提出的請求而設計。在下列範例中，將每個*********取代為您自己的資訊。

範例

- 範例：唯讀存取
- 範例：限制特定區域的存取
- 使用執行個體
- 啟動執行個體 (RunInstances)
- 使用 競價型執行個體
- 範例：使用 預留執行個體
- 範例：標籤資源
- 範例：使用 IAM 角色
- 範例：使用路由表
- 範例：允許特定執行個體檢視其他服務中的資源 AWS
- 範例：使用啟動範本
- 使用執行個體中繼資料
- 使用 Amazon EBS 磁碟區和快照

如需可在 Amazon EC2 主控台中使用的政策範例，請參閱控制 Amazon EC2 主控台存取的政策範例。

範例：唯讀存取

下列政策可授予使用者使用名稱開頭為 `Describe` 之所有 Amazon EC2 API 動作的許可。`Resource` 元素使用萬用字元，表示使用者可以為這些 API 動作指定所有資源。如果 API 動作不支援資源層級許可，也需要使用 * 萬用字元。如需可與 Amazon EC2 API 動作搭配使用的 ARN 詳細資訊，請參閱 Amazon EC2 的動作、資源與條件金鑰。

因為根據預設會拒絕使用者使用 API 動作的許可，所以使用者沒有對資源執行任何動作的許可 (除非其他陳述式授予他們該許可)。

JSON

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

範例：限制特定區域的存取

下列政策拒絕使用所有 Amazon EC2 API 動作的使用者許可，除非該區域為歐洲 (法蘭克福)。它使用全域條件金鑰 `aws:RequestedRegion`，所有 Amazon EC2 API 動作都支援它。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "eu-central-1"
        }
      }
    }
  ]
}
```

或者，您可以使用條件金鑰 `ec2:Region`，它為 Amazon EC2 獨有並受所有 Amazon EC2 API 動作支援。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "ec2:Region": "eu-central-1"
        }
      }
    }
  ]
}
```

使用執行個體

範例

- 範例：描述、啟動、停止、開始及終止所有執行個體
- 範例：描述所有執行個體，並只停止、啟動及終止特定執行個體

範例：描述、啟動、停止、開始及終止所有執行個體

下列政策可授予使用者使用在 `Action` 元素中指定之 API 動作的許可。`Resource` 元素使用 * 萬用字元，表示使用者可以為這些 API 動作指定所有資源。如果 API 動作不支援資源層級許可，也需要使用 * 萬用字元。如需可與 Amazon EC2 API 動作搭配使用的 ARN 詳細資訊，請參閱 Amazon EC2 的動作、資源與條件金鑰。

因為根據預設會拒絕使用者使用 API 動作的許可，所以使用者沒有使用任何其他 API 動作的許可 (除非其他陳述式授予他們該許可)。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeImages",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "*"
    }
   ]
}
```

範例：描述所有執行個體，並只停止、啟動及終止特定執行個體

下列政策允許使用者描述所有執行個體、只開始及停止執行個體 i-1234567890abcdef0 和 i-0598c7d356eba48d7，並只終止 us-east-1 區域中資源標籤為 "purpose=test" 的執行個體。

第一個陳述式的 Resource 元素使用 * 萬用字元，表示使用者可以為該動作指定所有資源；在本例中，他們可以列出所有執行個體。如果 API 動作不支援資源層級許可 (在本例中為 ec2:DescribeInstances)，也需要使用 * 萬用字元。如需可與 Amazon EC2 API 動作搭配使用的 ARN 詳細資訊，請參閱 Amazon EC2 的動作、資源與條件金鑰。

第二個陳述式的 StopInstances 和 StartInstances 動作使用資源層級許可。特定的執行個體在 Resource 元素中以其 ARN 表示。

第三個陳述式允許使用者終止 us-east-1區域中屬於指定 AWS 帳戶的所有執行個體，但僅限於執行個體具有標籤 的情況下"purpose=test"。Condition 元素可限定政策陳述式生效的條件。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

啟動執行個體 (RunInstances)

RunInstances API 動作會啟動一個或多個 隨需執行個體 或 競價型執行個體。RunInstances 需要 AMI 並建立一個執行個體。使用者可以在請求中指定金鑰對和安全性群組。啟動至 VPC 需要子網，並會建立網路介面。從由 Amazon EBS 支援的 AMI 啟動會建立磁碟區。因此，使用者必須具有使用這些

Amazon EC2 資源的許可。您可以建立需要使用者在 RunInstances 上指定選用參數，或限制使用者使用特定參數值的政策陳述式。

如需啟動執行個體所需的資源層級許可詳細資訊，請參閱 Amazon EC2 的動作、資源及條件金鑰。

根據預設，使用者不具描述、開始、停止或終止所產生執行個體的許可。將管理所產生執行個體之許可授予使用者的一個方式，就是為每個執行個體建立特定標籤，再建立讓他們管理具有該標籤之執行個體的陳述式。如需詳細資訊，請參閱使用執行個體。

Resources

- AMI
- 執行個體類型
- 子網
- EBS 磁碟區
- Tags (標籤)
- 啟動範本中的標籤
- Elastic GPU
- 啟動範本

AMI

下列政策只允許使用者使用指定的 AMI ami-9e1670f7 和 ami-45cf5c3c 啟動執行個體。使用者無法使用其他 AMI 啟動執行個體 (除非其他陳述式授予使用者該許可)。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:us-east-1::image/ami-9e1670f7",
        "arn:aws:ec2:us-east-1::image/ami-45cf5c3c",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
```

```
        "arn:aws:ec2:us-east-1:111122223333:security-group/*",
        "arn:aws:ec2:us-east-1:111122223333:subnet/*",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ]
    }
  ]
}
```

或者，下列政策允許使用者從 Amazon 或某些受信任和已驗證的合作夥伴擁有的所有 AMI 啟動執行個體。第一個陳述式的 Condition 元素會測試 ec2:Owner 是否為 amazon。使用者無法使用其他 AMI 啟動執行個體 (除非其他陳述式授予使用者該許可)。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
         {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
```

```
   ]
}
```

執行個體類型

下列政策只允許使用者使用 t2.micro 或 t2.small 執行個體類型啟動執行個體，您可能會為了控制成本而這樣做。因為第一個陳述式的 Condition 元素會測試 ec2:InstanceType 是否為 t2.micro 或 t2.small，所以使用者無法啟動更大的執行個體。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
   ]
}
```

或者，您可以建立政策來拒絕使用者啟動 t2.micro 和 t2.small 執行個體類型以外之任何執行個體的許可。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
        {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringNotEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
   ]
}
```

子網

下列政策只允許使用者使用指定的子網 subnet-*12345678* 啟動執行個體。群組無法將執行個體啟動至任何其他子網 (除非其他陳述式授予使用者該許可)。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:subnet/subnet-12345678",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1::image/ami-*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:us-east-1:111122223333:security-group/*"
      ]
    }
   ]
}
```

或者，您可以建立拒絕使用者將執行個體啟動至任何其他子網之許可的政策。陳述式執行此作業的方式是拒絕建立網路介面的許可，但指定子網 subnet-*12345678* 的介面除外。此拒絕會覆寫為允許將執行個體啟動至其他子網所建立的任何其他政策。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
         {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-12345678"
```

```
            }
         }
      },
      {
         "Effect": "Allow",
         "Action": "ec2:RunInstances",
         "Resource": [
            "arn:aws:ec2:us-east-1::image/ami-*",
            "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-1:111122223333:security-group/*"
            ]
         }
      ]
}
```

EBS 磁碟區

下列政策只允許使用者在執行個體的 EBS 磁碟區已加密時啟動執行個體。使用者必須從使用加密快照建立的 AMI 啟動執行個體，以確保根磁碟區經過加密。使用者在啟動期間連接至執行個體的任何其他磁碟區也必須經過加密。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

Tags (標籤)

建立時為執行個體加上標籤

下列政策允許使用者在建立期間啟動執行個體並為執行個體套用標籤。針對套用標籤的資源建立動作，使用者必須具有使用 CreateTags 動作的許可。第二個陳述式使用 ec2:CreateAction 條件鍵限制使用者在 RunInstances 的條件下才可建立標籤，且僅限為執行個體建立。使用者無法為現有資源套用標籤，也無法使用 RunInstances 請求為磁碟區套用標籤。

如需詳細資訊，請參閱在建立期間授予標籤 Amazon EC2 資源的許可。

JSON

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

```
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

使用特定標籤建立時為執行個體和磁碟區加上標籤

下列政策包含 aws:RequestTag 條件鍵，需要使用者使用標籤 RunInstances 和 environment=production 為 purpose=webserver 建立的任何執行個體和磁碟區套用標籤。如果使用者未傳遞這些特定標籤，或完全未指定標籤，請求會失敗。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1::image/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
```

```
      ],
      "Resource": [
          "arn:aws:ec2:us-east-1:111122223333:volume/*",
          "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/environment": "production" ,
             "aws:RequestTag/purpose": "webserver"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

使用至少一個特定標籤建立時為執行個體和磁碟區加上標籤

下列政策在 ForAnyValue 條件的部分使用 aws:TagKeys 修飾詞，表示至少必須在請求中指定一個標籤，而且它必須包含 environment 鍵或 webserver 鍵。此標籤必須同時套用至執行個體和磁碟區。使用者可以在請求中指定任何標籤值。

JSON

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

```
            "ec2:RunInstances"
        ],
        "Resource": [
            "arn:aws:ec2:us-east-1::image/*",
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "arn:aws:ec2:us-east-1:111122223333:security-group/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
            "ec2:RunInstances"
        ],
        "Resource": [
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:instance/*"
        ],
        "Condition": {
            "ForAnyValue:StringEquals": {
                "aws:TagKeys": ["environment","webserver"]
            }
         }
      },
      {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction" : "RunInstances"
            }
         }
      }
    ]
}
```

如果執行個體在建立時被標籤，則必須使用特定標籤來加以標籤

在下列政策中，使用者不需要在請求中指定標籤，但若需指定，標籤必須是 purpose=test。不允許其他標籤。使用者可以套用標籤至 RunInstances 請求中任何可套用標籤的資源。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/purpose": "test",
             "ec2:CreateAction" : "RunInstances"
          },
          "ForAllValues:StringEquals": {
              "aws:TagKeys": "purpose"
          }
       }
    }
  ]
}
```

不允許任何人為 RunInstances 建立時呼叫標籤

JSON

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

```
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

只允許 spot-instances-request 的特定標籤。不一致的數字 2 會意外在此發揮作用。在一般情況下，不指定任何標籤會導致「未驗證」狀態。如果是 spot-instances-request，則不會評估此政策，如果沒有 spot-instances-request 標記，因此 non-tag Spot on Run 請求將會成功。

啟動範本中的標籤

在下列範例中，使用者可以啟動執行個體，但前提是他們要使用特定的啟動範本 (`lt-09477bcd97b0d310e`)。`ec2:IsLaunchTemplateResource` 條件鍵可防止使用者覆寫在啟動範本中指定的任何資源。陳述式的第二部分允許使者在建立時為執行個體套用標籤 – 如果在啟動範本中指定執行個體的標籤，則需要陳述式的這個部分。

JSON

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

```
  "Statement": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/lt-09477bcd97b0d310e"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

Elastic GPU

在下列政策中，使用者可以啟動執行個體，並指定要連接至執行個體的 Elastic GPU。使用者可以在任何區域中啟動執行個體，但他們只可在 us-east-2 區域中於啟動期間連接彈性 GPU。

ec2:ElasticGpuType 條件索引鍵可確保執行個體使用 eg1.medium 或 eg1.large 彈性 GPU 類型。

JSON

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

```
    "Statement": [
             {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:elastic-gpu/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2",
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:111122223333:network-interface/*",
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ec2:*:111122223333:subnet/*",
                "arn:aws:ec2:*:111122223333:volume/*",
                "arn:aws:ec2:*:111122223333:key-pair/*",
                "arn:aws:ec2:*:111122223333:security-group/*"
            ]
        }
    ]
}
```

啟動範本

在下列範例中，使用者可以啟動執行個體，但前提是他們要使用特定的啟動範本 (lt-09477bcd97b0d310e)。使用者可以透過在 RunInstances 動作中指定參數，來覆寫啟動範本中的任何參數。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/lt-09477bcd97b0d310e"
          }
       }
    }
  ]
}
```

在此範例中，使用者只有在使用啟動範本時才能啟動執行個體。此政策會使用 ec2:IsLaunchTemplateResource 條件索引鍵，以防止使用者覆寫啟動範本中預先存在的任何 ARN。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
```

```
        }
      }
   ]
}
```

下列政策範例允許使用者啟動執行個體，但前提是他們要使用啟動範本。使用者無法覆寫請求中的子網和網路介面參數；這些參數只能在啟動範本中予以指定。陳述式的第一部分使用 NotResource 元素來允許子網和網路介面以外的所有其他資源。陳述式的第二部分允許子網和網路介面資源，但前提是它們來自啟動範本。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                      "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          }
       }
    },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                   "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
```

```
    }
  ]
}
```

下列範例只允許使用者在使用啟動範本且該啟動範本具有標籤 Purpose=Webservers 時，啟動執行個體。使用者無法覆寫 RunInstances 動作中的任何啟動範本參數。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
         "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
       "StringEquals": {
           "aws:ResourceTag/Purpose": "Webservers"
        }
       }
     }
  ]
}
```

使用 競價型執行個體

您可以使用 RunInstances 動作來建立競價型執行個體請求，並在建立時標記競價型執行個體請求。要為 RunInstances 指定的資源是 `spot-instances-request`。

`spot-instances-request` 資源會在 IAM 政策中進行評估，如下所示：

- 如果您沒有在建立時標記 Spot 執行個體請求，則 Amazon EC2 不會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。
- 如果您在建立時標記 Spot 執行個體請求，Amazon EC2 會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。

因此，對於 `spot-instances-request` 資源，下列規則適用於 IAM 政策：

- 如果您使用 RunInstances 來建立 Spot 執行個體請求，而且您不打算在建立時標記 Spot 執行個體請求，則不需要明確允許 `spot-instances-request` 資源；呼叫將會成功。
- 如果您使用 RunInstances 來建立 Spot 執行個體請求，並打算在建立時標記 Spot 執行個體請求，則必須在 RunInstances 允許陳述式中包含 `spot-instances-request` 資源，否則呼叫將會失敗。
- 如果您使用 RunInstances 建立 Spot 執行個體請求，並打算在建立時標記 Spot 執行個體請求，則必須在 CreateTag 允許陳述式中指定 `spot-instances-request` 資源或 * 萬用字元，否則呼叫將會失敗。

您可以使用 RunInstances 或 RequestSpotInstances 請求 競價型執行個體。下列範例 IAM 政策僅適用於使用 RunInstances 請求 競價型執行個體。

範例：使用 RunInstances 請求 競價型執行個體

下列政策可讓使用者使用 RunInstances 動作來請求 競價型執行個體。由 RunInstances 建立的 `spot-instances-request` 資源會請求 競價型執行個體。

> **Note**
>
> 若要使用 RunInstances 來建立 Spot 執行個體請求，如果您不打算在建立時標記 Spot 執行個體請求，則可以從 `Resource` 清單中省略 `spot-instances-request`。這是因為如果 Spot 執行個體請求未在建立時標記，則 Amazon EC2 不會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        }
    ]
}
```

Warning

不支援的 – 範例：拒絕使用者 競價型執行個體 使用 RunInstances 請求的權限 `spot-instances-request` 資源不支援下列政策。
下列政策旨在提供使用者啟動 隨需執行個體 的許可，但拒絕使用者請求 競價型執行個體 的許可。由 RunInstances 建立的 `spot-instances-request` 資源是請求 競價型執行個體 的資源。第二個陳述式是為了拒絕 `spot-instances-request` 資源的 RunInstances 動作。但系統不支援此條件，因為如果 Spot 執行個體請求未在建立時標記，則 Amazon EC2 不會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。

JSON

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

```
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequestsNOTSUPPORTEDDONOTUSE",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
        }
    ]
}
```

範例：在建立時標記 Spot 執行個體請求

下列政策可讓使用者標記執行個體啟動期間建立的所有資源。第一個陳述式可讓 RunInstances 建立列出的資源。由 RunInstances 建立的 `spot-instances-request` 資源是請求 競價型執行個體 的資源。第二個陳述式提供 * 萬用字元，允許在執行個體啟動時建立所有資源時加上標記。

> **Note**
>
> 如果您在建立時標記 Spot 執行個體請求，Amazon EC2 會評估 RunInstances 陳述式中的 `spot-instances-request` 資源。因此，您必須明確允許 RunInstances 動作的 `spot-instances-request` 資源，否則呼叫將會失敗。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

範例：拒絕在建立時標記 Spot 執行個體請求

下列政策拒絕使用者標記執行個體啟動期間建立的資源的許可。

第一個陳述式可讓 RunInstances 建立列出的資源。由 RunInstances 建立的 `spot-instances-request` 資源是請求 競價型執行個體 的資源。第二個陳述式會提供 * 萬用字元，以拒絕在執行個體啟動時建立所有標記的資源。如果在建立時標記 `spot-instances-request` 或任何其他資源，RunInstances 呼叫將會失敗。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenyTagResources",
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Warning

不支援 - 範例：只有系統指派特定標籤時，才允許建立 Spot 執行個體請求

`spot-instances-request` 資源不支援下列政策。

下列政策旨在授與 RunInstances 建立請求的許可，只有在 Spot 執行個體請求標記為特定標籤時才會建立請求。

第一個陳述式可讓 RunInstances 建立列出的資源。

第二個陳述式向使用者授予許可，以便僅在請求具有 `environment=production` 標籤時才建立 Spot 執行個體請求。如果此條件套用至 RunInstances 所建立的其他資源，則不指定任

何標籤會導致 Unauthenticated 錯誤。不過，如果沒有為 Spot 執行個體請求指定標籤，則 Amazon EC2 不會評估 RunInstances 陳述式中的 spot-instances-request 資源，這會導致 RunInstances 建立非標籤的 Spot 執行個體請求。
請注意，指定 environment=production 以外的其他標記會導致 Unauthenticated 錯誤，因為如果使用者標記 Spot 執行個體請求，Amazon EC2 會評估 RunInstances 陳述式中的 spot-instances-request 資源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
        {
            "Sid":
 "RequestSpotInstancesOnlyIfTagIsEnvironmentProductionNOTSUPPORTEDDONOTUSE",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/
*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
```

```
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }

    ]
}
```

範例：如果系統指派特定標籤，則拒絕建立 Spot 執行個體請求

如果請求已標記為 `environment=production`，下列政策會拒絕 RunInstances 建立 Spot 執行個體請求的許可。

第一個陳述式可讓 RunInstances 建立列出的資源。

如果請求具有 `environment=production` 標籤，第二個陳述式會拒絕使用者建立 Spot 執行個體請求的許可。指定 `environment=production` 為標籤會導致 `Unauthenticated` 錯誤。指定其他標籤或不指定標籤將導致建立 Spot 執行個體請求。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
```

```
        },
        {
            "Sid": "DenySpotInstancesRequests",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

範例：使用 預留執行個體

下列政策可提供使用者檢視、修改及購買您帳戶中 預留執行個體 的許可。

您無法針對個別的 預留執行個體 設定資源層級的權限。此政策表示使用者可存取帳戶中的所有 預留執行個體。

Resource 元素使用 * 萬用字元，表示使用者可以為此動作指定所有資源；在本例中，他們可以列出及修改帳戶中的所有預留執行個體。他們也可以使用帳戶登入資料購買 預留執行個體。如果 API 動作不支援資源層級許可，也需要使用 * 萬用字元。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
```

```
        "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
    }
   ]
}
```

用以允許使用者檢視及修改您帳戶中的預留執行個體，但不允許購買新的預留執行個體。

JSON

```
{
  "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    }
  ]
}
```

範例：標籤資源

下列政策只允許使用者在標籤包含 `CreateTags` 鍵和 `environment` 值時，使用 `production` 動作將標籤套用至執行個體。不允許其他標籤，並且使用者無法將標籤用於任何其他資源類型。

JSON

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

```
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
    ]
}
```

下列政策允許使用者套用標籤至任何已有 owner 金鑰和使用者名稱值之標籤的可套用標籤資源。此外，使用者也必須在請求中使用 anycompany:environment-type 鍵和 test 值或 prod 值指定標籤。使用者可以在請求中指定其他標籤。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type":
 ["test","prod"],
                    "aws:ResourceTag/owner": "${aws:username}"
                }
            }
        }
    ]
}
```

您可以建立允許使用者為資源刪除特定標籤的 IAM 政策。例如，下列政策允許使用者在請求中指定的標籤鍵為 environment 或 cost-center 時，刪除磁碟區的標籤。您可以為標籤指定任何值，但標籤鍵必須符合指定的任一鍵。

> **Note**
>
> 如果您刪除資源，也會刪除與該資源相關聯的所有標籤。使用者不需要使用 ec2:DeleteTags 動作的許可，也能刪除具有標籤的資源；他們只需要執行刪除動作的許可。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Effect": "Allow",
      "Action": "ec2:DeleteTags",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment","cost-center"]
        }
      }
    }
  ]
}
```

此政策只允許使用者刪除任何資源上的 environment=prod 標籤，而且僅限資源已套用 owner 金鑰和使用者名稱值之標籤的情況。使用者無法刪除資源的任何其他標籤。

JSON

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

```
      "Action": [
        "ec2:DeleteTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "prod",
          "aws:ResourceTag/owner": "${aws:username}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment"]
        }
      }
    }
  ]
}
```

範例：使用 IAM 角色

下列政策允許使用者對具有標籤 `department=test` 的執行個體連接、取代及分離 IAM 角色。取代或分離 IAM 角色需要關聯 ID，因此政策也會授予使用者使用 `ec2:DescribeIamInstanceProfileAssociations` 動作的許可。

使用者必須具有使用 `iam:PassRole`動作將角色傳遞給執行個體的許可。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation",
        "ec2:DisassociateIamInstanceProfile"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department":"test"
```

```
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeIamInstanceProfileAssociations",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/DevTeam*"
    }
  ]
}
```

下列政策允許使用者對任何執行個體連接或取代 IAM 角色。使用者只能連接或取代名稱開頭為 TestRole- 的 IAM 角色。針對 iam:PassRole 動作，請確定您指定 IAM 角色的名稱，而不是執行個體描述檔 (如果名稱不同)。如需詳細資訊，請參閱執行個體描述檔。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::111122223333:role/TestRole-*"
        }
    ]
}
```

範例：使用路由表

下列政策允許使用者新增、移除及取代只與 VPC vpc-ec43eb89 相關聯之路由表的路由。若要針對 ec2:Vpc 條件鍵指定 VPC，您必須指定 VPC 的完整 ARN。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-
ec43eb89"
                }
            }
        }
    ]
}
```

範例：允許特定執行個體檢視其他服務中的資源 AWS

下列是您可連接至 IAM 角色的政策範例。此政策允許執行個體檢視各種 AWS 服務中的資源。它使用 ec2:SourceInstanceARN 條件鍵指定發出請求的執行個體必須是執行個體

i-093452212644b0dd6。如果同一個 IAM 角色與另一個執行個體相關聯，其他執行個體就無法執行這些動作。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "s3:ListAllMyBuckets",
                "dynamodb:ListTables",
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ArnEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:us-
east-1:111122223333:instance/i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

範例：使用啟動範本

下列政策允許使用者建立啟動範本版本並修改啟動範本，但僅限特定啟動範本 (lt-*09477bcd97b0d3abc*)。使用者無法使用其他啟動範本。

JSON

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

```
      "Action": [
        "ec2:CreateLaunchTemplateVersion",
        "ec2:ModifyLaunchTemplate"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/
lt-09477bcd97b0d3abc"
    }
  ]
}
```

下列政策允許使用者刪除任何啟動範本和啟動範本版本，但前提是啟動範本具有標籤 Purpose=Testing。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Action": [
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

使用執行個體中繼資料

下列政策可確保使用者只能使用 執行個體中繼資料服務第 2 版 (IMDSv2) 擷取 執行個體中繼資料 。您可以將以下四個政策合併為一個包含四個陳述式的政策。當合併為一個政策時，您可以將政策用作為服

務控制政策 (SCP)。它同樣可用作為套用至現有 IAM 政策 (取消和限制現有許可) 的拒絕政策，或者用作為跨帳戶、組織單位 (OU) 或整個組織全域套用的 SCP。

> **Note**
>
> 下列 RunInstances 中繼資料選項政策必須與授予委託人許可使用 RunInstances 啟動執行個體的政策搭配使用。如果委託人同樣不具有 RunInstances 許可，則無法啟動執行個體。如需詳細資訊，請參閱使用執行個體和啟動執行個體 (RunInstances)中的政策。

> **Important**
>
> 如果您使用 Auto Scaling 群組，而且需要在所有新執行個體上要求使用 IMDSv2，您的 Auto Scaling 群組必須使用啟動範本。
> 當 Auto Scaling 群組使用啟動範本時，會在建立新的 Auto Scaling 群組時，檢查 IAM 主體的 `ec2:RunInstances` 許可。當現有的 Auto Scaling 群組更新為使用新的啟動範本或新版本的啟動範本時，也會檢查這些許可。
> 只有在建立或更新使用啟動範本的 Auto Scaling 群組時，才會對 `RunInstances` 檢查 IAM 主體使用 IMDSv1 的限制。對於設為使用 `Latest` or `Default` 啟動範本的 Auto Scaling 群組，建立新版本的啟動範本時，不會檢查許可。對於要檢查的許可，您必須將 Auto Scaling 群組設為使用特定版本的啟動範本。
>
> 若要在 Auto Scaling 群組啟動的執行個體上強制使用 IMDSv2，需要以下額外步驟：
>
> 1. 藉由為建立的新委託人使用服務控制政策 (SCP) 或 IAM 許可界限，對組織中的所有帳戶停用啟動組態。對於具有 Auto Scaling 群組許可的 IAM 主體，使用此條件金鑰更新其相關的政策。若要停用啟動組態，請使用 `"autoscaling:LaunchConfigurationName"` 條件金鑰 (其值指定為 `null`)，建立或修改相關的 SCP、許可界限或 IAM 政策。
> 2. 對於新的啟動範本，請在啟動範本中設定執行個體中繼資料選項。對於現有的啟動範本，請建立新版本的啟動範本，並在新版本中設定執行個體中繼資料選項。
> 3. 在授予任何委託人許可使用啟動範本的政策中，藉由指定 `"autoscaling:LaunchTemplateVersionSpecified": "true"` 來限制 `$latest` 和 `$default` 的關聯。藉由限制使用特定版本的啟動範本，可以確保使用已設定執行個體中繼資料選項的版本來啟動新執行個體。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling API 參考中的 LaunchTemplateSpecification，指定 `Version` 參數。
> 4. 對於使用啟動組態的 Auto Scaling 群組，將啟動組態取代為啟動範本。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的移轉 Auto Scaling 群組至啟動範本。

5. 對於使用啟動範本的 Auto Scaling 群組，請確保其將新啟動範本與設定的執行個體中繼資料選項搭配使用，或將新版本的目前啟動範本與設定的執行個體中繼資料選項搭配使用。如需詳細資訊，請參閱 update-auto-scaling-group。

範例

- 需要使用 IMDSv2
- 拒絕選擇退出 IMDSv2
- 指定最大跳轉限制
- 限制可修改執行個體中繼資料選項的人員
- 需要從 IMDSv2 擷取角色登入資料

需要使用 IMDSv2

下列政策指出，除非也選擇加入執行個體以要求使用 IMDSv2 (由 "ec2:MetadataHttpTokens": "required" 指示)，否則您無法呼叫 RunInstances API。如果您未指定執行個體需要 IMDSv2，則在呼叫 RunInstances API 時會出現 UnauthorizedOperation 錯誤。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringNotEquals": {
                    "ec2:MetadataHttpTokens": "required"
                }
            }
        }
    ]
}
```

拒絕選擇退出 IMDSv2

下列政策規定您無法呼叫 ModifyInstanceMetadataOptions API，也無法允許 IMDSv1 或 IMDSv2 選項。如果您呼叫 ModifyInstanceMetadataOptions API，則必須將 HttpTokens 屬性設定為 required。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "DenyIMDSv1HttpTokensModification",
        "Effect": "Deny",
        "Action": "ec2:ModifyInstanceMetadataOptions",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringNotEquals": {
                "ec2:Attribute/HttpTokens": "required"
            },
            "Null": {
                "ec2:Attribute/HttpTokens": false
            }
        }
    }]
}
```

指定最大跳轉限制

下列政策指出，除非您同時指定躍點限制，且躍點限制不能超過 3，否則您無法呼叫 RunInstances API。如果不這麼做，則在呼叫 RunInstances API 時會出現 UnauthorizedOperation 錯誤。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "MaxImdsHopLimit",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
```

```
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "NumericGreaterThan": {
                    "ec2:MetadataHttpPutResponseHopLimit": "3"
                }
            }
        }
    ]
}
```

限制可修改執行個體中繼資料選項的人員

下列政策只允許具有角色 `ec2-imds-admins` 的使用者對執行個體中繼資料選項進行變更。如果 `ec2-imds-admins` 角色以外的任何委託人嘗試呼叫 ModifyInstanceMetadataOptions API，將出現 `Unauthorized0peration` 錯誤。這個陳述式可用來控制 ModifyInstanceMetadataOptions API 的使用；ModifyInstanceMetadataOptions API 目前沒有細微存取控制 (條件)。

需要從 IMDSv2 擷取角色登入資料

下列政策指出，如果此政策套用至角色，且 EC2 服務假定該角色，並將產生的登入資料用於簽署請求，則必須由從 IMDSv2 擷取的 EC2 角色登入資料簽署該請求。否則，其所有 API 呼叫將得到 `Unauthorized0peration` 錯誤。一般情況下，可套用此陳述式/政策，因為如果請求未由 EC2 角色登入資料簽署，則不具任何作用。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "RequireAllEc2RolesToUseV2",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:RoleDelivery": "2.0"
                }
            }
        }
    ]
```

```
}
```

使用 Amazon EBS 磁碟區和快照

如需使用 Amazon EBS 磁碟區和快照的範例政策，請參閱 Amazon EBS 的身分型政策範例。

控制 Amazon EC2 主控台存取的政策範例

您可以使用 IAM 政策向使用者授予使用 Amazon EC2 所需的權限。如需逐步說明，請參閱「IAM 使用者指南」中的建立 IAM 政策。

主控台會針對其功能使用其他的 API 動作，所以這些政策可能不會如預期般運作。例如，使用者的權限如果只能使用 `DescribeVolumes` API 動作，則當該使用者試著在主控台中檢視磁碟區時，就可能會遇到錯誤。本節中所示範的政策，可讓使用者使用主控台的特定部分。如需為 Amazon EC2 主控台建立政策的詳細資訊，請參閱下列 AWS 安全部落格文章：授予使用者在 Amazon EC2 主控台中的工作許可。

以下範例顯示了可用於授予使用者使用 Amazon EC2 的權限的政策陳述式。將每個*########*替換為自己的資訊。這些政策專為使用 AWS 管理主控台開發套件發出的請求所設計。Amazon EC2 主控台可能會呼叫多個 API 動作來顯示單一資源，而且在使用者嘗試任務且主控台顯示錯誤之前可能並不明顯。如需詳細資訊，請參閱下列 AWS 安全部落格文章：授予使用者在 Amazon EC2 主控台中的工作許可。

範例

- 範例：唯讀存取
- 範例：使用 EC2 啟動執行個體精靈
- 範例：使用安全群組
- 範例：使用彈性 IP 地址
- 範例：使用 預留執行個體

為了協助您確定在主控台中執行任務所需的 API 操作，您可以使用記錄呼叫的服務，例如 AWS CloudTrail。如果政策並未授予權限來建立或修改特定的資源，主控台會顯示編碼的訊息，其中包含診斷資訊。您可以使用 的 DecodeAuthorizationMessage API 動作 AWS STS或 中的 decode-authorization-message 命令來解碼訊息 AWS CLI。

範例：唯讀存取

若要允許使用者檢視 Amazon EC2 主控台中的所有資源，您可以使用與下列範例相同的政策：範例：唯讀存取。除非有另一個陳述式授予使用者此種權限，否則使用者無法對其他資源執行任何動作，也無法建立新資源。

檢視執行個體、AMI 和快照

或者，您可以提供對資源子集的唯讀存取權限。若要這麼做，請針對每項資源，使用特定的 `ec2:Describe` 動作，來取代 `ec2:Describe` API 動作中的 * 萬用字元。下列的政策可讓使用者檢視 Amazon EC2 主控台中的所有執行個體、AMI 和快照。`ec2:DescribeTags` 動作可讓使用者檢視公用 AMI。主控台需要標記資訊來顯示公用 AMI；不過，您可以移除這項動作，來讓使用者只檢視私有 AMI。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeImages",
         "ec2:DescribeTags",
         "ec2:DescribeSnapshots"
      ],
      "Resource": "*"
   }
   ]
}
```

> **Note**
>
> Amazon EC2 `ec2:Describe*` API 動作不支援資源層級的許可，因此您無法控制使用者可以在主控台中檢視哪些個別資源。因此，在上述的陳述式中，`Resource` 元素必須包含 * 萬用字元。如需可與 Amazon EC2 API 動作搭配使用的 ARN 詳細資訊，請參閱 Amazon EC2 的動作、資源與條件金鑰。

檢視執行個體和 CloudWatch 指標

下列的政策可讓使用者在 Amazon EC2 主控台中檢視執行個體，以及在 Instances (執行個體) 頁面的 Monitoring (監控) 索引標籤中，檢視 CloudWatch 警示和指標。Amazon EC2 主控台使用 CloudWatch API 來顯示警示和指標，因此您必須授予使用者許可以執行 `cloudwatch:DescribeAlarms`、`cloudwatch:DescribeAlarmsForMetric`、`cloudwatch:ListMet` 和 `cloudwatch:GetMetricData` 動作。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "cloudwatch:DescribeAlarms",
         "cloudwatch:DescribeAlarmsForMetric",
         "cloudwatch:ListMetrics",
         "cloudwatch:GetMetricStatistics",
         "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
   }
   ]
}
```

範例：使用 EC2 啟動執行個體精靈

Amazon EC2 啟動執行個體精靈是一個內含選項的畫面，可用來設定和啟動執行個體。政策必須包含使用 API 動作的權限，來讓使用者能夠使用精靈的選項。如果政策未包含使用這些動作的權限，可能無法適當地載入精靈中的某些項目，使用者就無法完成啟動。

基本啟動執行個體精靈存取權

若要順利地完成啟動，必須授予使用者權限，來使用 `ec2:RunInstances` API 動作和至少下列的 API 動作：

- `ec2:DescribeImages`：檢視和選取 AMI。

- ec2:DescribeInstanceTypes：檢視和選取執行個體類型。
- ec2:DescribeVpcs：檢視可用的網路選項。
- ec2:DescribeSubnets：檢視所選擇 VPC 的所有可用子網。
- ec2:DescribeSecurityGroups 或 ec2:CreateSecurityGroup：檢視和選取現有的安全群組，或建立新的安全群組。
- ec2:DescribeKeyPairs 或 ec2:CreateKeyPair：選取現有的金鑰對，或建立新的金鑰對。
- ec2:AuthorizeSecurityGroupIngress：新增傳入規則。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateKeyPair"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        }
    ]
}
```

您可以在政策中加入 API 動作，來為使用者提供更多選項，例如：

- `ec2:DescribeAvailabilityZones`：檢視並選取特定的可用區域。
- `ec2:DescribeNetworkInterfaces`：針對選取的子網，檢視並選取現有的網路介面。
- 若要在 VPC 安全群組中新增外傳規則，必須授予使用者權限來使用 `ec2:AuthorizeSecurityGroupEgress` API 動作。若要修改或刪除現有的規則，必須授予使用者權限，以使用相關的 `ec2:RevokeSecurityGroup*` API 動作。
- `ec2:CreateTags`：標記由 `RunInstances` 所建立的資源。如需詳細資訊，請參閱在建立期間授予標籤 Amazon EC2 資源的許可。如果使用者不具有使用此動作的許可，而且試著在 啟動執行個體精靈的標記頁面上使用標籤，則啟動會失敗。

> **Important**
>
> 啟動執行個體時指定 Name (名稱) 可建立標籤，並且需要 `ec2:CreateTags` 動作。請小心授與使用者使用 `ec2:CreateTags` 動作的許可，因為這樣做會限制您使用 `aws:ResourceTag` 條件金鑰來限制使用者使用其他資源的能力。如果您授予使用者使用 `ec2:CreateTags`動作的許可，他們可以變更資源的標籤以略過這些限制。如需詳細資訊，請參閱使用屬性型存取控制存取。

- 若要在選取 AMI 時使用 Systems Manager 參數，您必須在政策中新增 `ssm:DescribeParameters` 和 `ssm:GetParameters`。`ssm:DescribeParameters` 會授予使用者檢視和選取 Systems Manager 參數的許可。`ssm:GetParameters` 會授與使用者取得 Systems Manager 參數值的許可。您也可以限制對特定 Systems Manager 參數的存取。如需詳細資訊，請參閱本節稍後的 Restrict access to specific Systems Manager parameters (限制特定 Systems Manager 參數的存取)。

目前，Amazon EC2 `Describe*` API 動作並不支援資源層級的許可，因此您無法限制使用者可以在啟動執行個體精靈中檢視哪些個別資源。不過，您可以對 `ec2:RunInstances` API 動作套用資源層級的權限，來限制使用者可以使用哪些資源來啟動執行個體。如果使用者選取了自己無使用授權的選項，啟動會失敗。

Restrict access to a specific instance type, subnet, and Region (限制對特定執行個體類型、子網和區域的存取)

下列政策可讓使用者使用 Amazon 所擁有的 AMI，來啟動 `t2.micro` 執行個體，而且只啟動至指定的子網 (`subnet-1a2b3c4d`)。使用者只能在指定區域中啟動。如果使用者選取了不同的區域，或是在啟動執行個體精靈中選取了不同的執行個體類型、AMI 或子網路，啟動會失敗。

第一個陳述式授與使用者許可來檢視啟動執行個體精靈中的選項，或建立新的選項。如上列的範例所解釋。第二個陳述式授予使用者許可，來針對 ec2:RunInstances 動作使用網路介面、磁碟區、金鑰對、安全群組和子網資源，如果要將執行個體啟動到 VPC，就必須執行此項動作。如需關於使用 ec2:RunInstances 動作的詳細資訊，請參閱 啟動執行個體 (RunInstances)。第三和第四個陳述式分別授予使用者使用執行個體和 AMI 資源的許可，但前提是執行個體為 t2.micro 執行個體，並且 AMI 為 Amazon 或某些受信任和已驗證的合作夥伴所有。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeImages",
         "ec2:DescribeInstanceTypes",
         "ec2:DescribeKeyPairs",
         "ec2:CreateKeyPair",
         "ec2:DescribeVpcs",
         "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups",
         "ec2:CreateSecurityGroup",
         "ec2:AuthorizeSecurityGroupIngress"
   ],
   "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action":"ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-2:111122223333:volume/*",
         "arn:aws:ec2:us-east-2:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-2:111122223333:security-group/*",
         "arn:aws:ec2:us-east-2:111122223333:subnet/subnet-1a2b3c4d"
      ]
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:instance/*"
```

```
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": "t2.micro"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
            "arn:aws:ec2:us-east-2::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   }
   ]
}
```

Restrict access to specific Systems Manager parameters (限制對特定 Systems Manager 參數的存取)

下列政策授與使用具有特定名稱之 Systems Manager 參數的存取權。

第一個陳述式會授與使用者許可，以在啟動執行個體精靈中選取 AMI 時檢視 Systems Manager 參數。第二個陳述式會授與使用者只使用命名為 `prod-*` 參數的許可。

JSON

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

```
      "Action": [
         "ssm:GetParameters"
      ],
     "Resource": "arn:aws:ssm:us-east-2:123456123456:parameter/prod-*"
   }
   ]
}
```

範例：使用安全群組

檢視安全群組，以及新增和移除規則

下列的政策會授予使用者許可，讓使用者可在 Amazon EC2 主控台中檢視安全群組，並針對具有 Department=Test 標籤的現有安全群組新增和移除傳入與傳出規則，以及列出和修改規則說明。

在第一個陳述式中，ec2:DescribeTags 動作可讓使用者在主控台中檢視標籤，如此就能更容易找出使用者獲得允許修改的安全群組。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups",
         "ec2:DescribeSecurityGroupRules",
         "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
```

```
      ],
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/Department": "Test"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
         "ec2:ModifySecurityGroupRules"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:security-group-rule/*"
      ]
   }
]}
```

使用 Create Security Group (建立安全群組) 對話方塊

您可以建立政策，允許使用者在 Amazon EC2 主控台中，使用 Create Security Group (建立安全群組) 對話方塊。若要使用此對話方塊，必須授予使用者權限，來至少使用下列的 API 動作：

- `ec2:CreateSecurityGroup`：建立新的安全群組。
- `ec2:DescribeVpcs`：在 VPC 清單中，檢視現有 VPC 的清單。

使用者可利用這些權限來順利建立新的安全群組，但無法新增任何規則到安全群組。若要在 Create Security Group (建立安全群組) 對話方塊中使用規則，您可以將下列的 API 動作加入政策：

- `ec2:AuthorizeSecurityGroupIngress`：新增傳入規則。
- `ec2:AuthorizeSecurityGroupEgress`：將外傳規則加入 VPC 安全群組。
- `ec2:RevokeSecurityGroupIngress`：修改或刪除現有的傳入規則。這是一項實用的動作，可讓使用者在主控台中使用 Copy to new (複製到新的) 功能。這項功能會開啟 Create Security Group (建立安全群組) 對話方塊，並使用與所選取安全群組相同的規則，來填入規則。
- `ec2:RevokeSecurityGroupEgress`：修改或刪除 VPC 安全群組的外傳規則。這是一項實用的動作，可讓使用者修改或刪除預設的外傳規則 (此規則允許所有外傳流量)。

- ec2:DeleteSecurityGroup：適用於無法儲存個別規則的情況。主控台會先建立安全群組，然後再加入指定的規則。如果規則無效，動作會失敗，而主控台會試著刪除安全群組。使用者仍在使用 Create Security Group (建立安全群組) 對話方塊，因此可以修正無效的規則，然後再次試著建立安全群組。此 API 動作並非必要，但如果使用者並未獲得使用該動作的權限，並試著建立包含無效規則的安全群組，則會建立未包含任何規則的安全群組，使用者必須在之後加入規則。
- ec2:UpdateSecurityGroupRuleDescriptionsIngress：新增或更新傳入 (輸入) 安全群組規則的描述。
- ec2:UpdateSecurityGroupRuleDescriptionsEgress：新增或更新傳出 (輸出) 安全群組規則的描述。
- ec2:ModifySecurityGroupRules：修改安全群組規則。
- ec2:DescribeSecurityGroupRules：列出安全群組規則。

下列政策授予使用者許可來使用 Create Security Group (建立安全群組) 對話方塊，並針對和指定 VPC (vpc-1a2b3c4d) 具有關聯的安全群組，建立其傳入和傳出規則。使用者可以建立 VPC 的安全群組，但無法將任何規則新增至這些安全群組。同樣地，使用者也無法將任何的規則，加入與 VPC vpc-1a2b3c4d 無關聯的現有安全群組。使用者也會獲得權限，來檢視主控台中的所有安全群組。如此，使用者就能更輕鬆地找出可以加入傳入規則的安全群組。此政策也會授予使用者權限，來刪除和 VPC vpc-1a2b3c4d 具有關聯的安全群組。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress"
```

```
      ],
      "Resource": "arn:aws:ec2:us-east-2:111122223333:security-group/*",
      "Condition":{
         "ArnEquals": {
            "ec2:Vpc": "arn:aws:ec2:us-east-2:111122223333:vpc/vpc-1a2b3c4d"
         }
      }
    }
   ]
}
```

範例：使用彈性 IP 地址

若要讓使用者在 Amazon EC2 主控台中檢視彈性 IP 地址，您必須授予使用者許可來使用 `ec2:DescribeAddresses` 動作。

若要讓使用者使用彈性 IP 地址，您可以將下列的動作加入政策。

- `ec2:AllocateAddress`：配置彈性 IP 地址。
- `ec2:ReleaseAddress`：釋出彈性 IP 地址。
- `ec2:AssociateAddress`：建立彈性 IP 地址與執行個體或網路介面的關聯。
- `ec2:DescribeNetworkInterfaces` 和 `ec2:DescribeInstances`：使用 Associate address (關聯地址) 畫面。此畫面會顯示可以和彈性 IP 地址建立關聯的執行個體或網路介面。
- `ec2:DisassociateAddress`：取消彈性 IP 地址與執行個體或網路介面的關聯。

下列政策可讓使用者檢視和配置彈性 IP 地址，並建立這些地址與執行個體的關聯。使用者不能建立彈性 IP 地址與網路介面的關聯、取消彈性 IP 地址的關聯，也不能釋出這些地址。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAddresses",
                "ec2:AllocateAddress",
```

```
                "ec2:DescribeInstances",
                "ec2:AssociateAddress"
            ],
            "Resource": "*"
        }
    ]
}
```

範例：使用 預留執行個體

以下政策允許使用者檢視和修改帳戶中的預留執行個體，以及在 AWS 管理主控台中購買新的預留執行個體。

此政策允許使用者檢視帳戶中的所有預留執行個體，以及帳戶中的所有隨需執行個體。您無法針對個別的預留執行個體設定資源層級的許可。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeReservedInstances",
         "ec2:ModifyReservedInstances",
         "ec2:PurchaseReservedInstancesOffering",
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "ec2:DescribeAvailabilityZones",
         "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
   }
   ]
}
```

若要確保 Amazon EC2 主控台能夠針對您可在其中購買預留執行個體的可用區域顯示相關的資訊，則必須執行 `ec2:DescribeAvailabilityZones` 動作。`ec2:DescribeInstances` 動作則非必要，但可確保使用者能夠檢視帳戶中的執行個體，並購買保留，以配合目前的規格。

您可以調整 API 動作來限制使用者的存取，例如移除 `ec2:DescribeInstances` 和 `ec2:DescribeAvailabilityZones` 代表使用者具有唯讀存取權限。

AWS Amazon EC2 的 受管政策

若要新增許可給使用者、群組和角色，使用 AWS 受管政策比自行撰寫政策更容易。建立 IAM 客戶受管政策需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的 AWS 受管政策。這些政策涵蓋常見的使用案例，並且可在您的帳戶中使用 AWS 。如需 AWS 受管政策的詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 服務會維護和更新 AWS 受管政策。您無法變更 AWS 受管政策中的許可。服務偶爾會在 AWS 受管政策中新增其他許可以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和角色)。當新功能啟動或新操作可用時，服務很可能會更新 AWS 受管政策。服務不會從 AWS 受管政策中移除許可，因此政策更新不會破壞您現有的許可。

此外， AWS 支援跨多個 服務之任務函數的受管政策。例如，ReadOnlyAccess AWS 受管政策提供所有 AWS 服務和資源的唯讀存取權。當服務啟動新功能時， 會為新操作和資源 AWS 新增唯讀許可。如需任務職能政策的清單和說明，請參閱 IAM 使用者指南中有關任務職能的AWS 受管政策。

AWS 受管政策： AmazonEC2FullAccess

您可以將 `AmazonEC2FullAccess` 政策連接到 IAM 身分。此政策授與允許完全存取 Amazon EC2的許可。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AmazonEC2FullAccess。

AWS 受管政策： AmazonEC2ReadOnlyAccess

您可以將 `AmazonEC2ReadOnlyAccess` 政策連接到 IAM 身分。此政策授予允許對 Amazon EC2 進行唯讀存取的許可。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AmazonEC2ReadOnlyAccess。

AWS 受管政策： AmazonEC2ImageReferencesAccessPolicy

您可將 `AmazonEC2ImageReferencesAccessPolicy` 政策連接到 IAM 身分。此政策授予使用 EC2 DescribeImageReferences API 所需的許可，包括檢視 EC2 執行個體、啟動範本、Systems Manager 參數，以及 Image Builder 配方的許可。政策支援 `IncludeAllResourceTypes`旗標，並在 AWS 新增對新資源類型的支援時繼續運作，不再需要未來的政策更新。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AmazonEC2ImageReferencesAccessPolicy。

AWS 受管政策： AWSEC2CapacityReservationFleetRolePolicy

此政策連接至名為 AWSServiceRoleForEC2CapacityReservationFleet 的服務連結角色，可讓服務代您在容量保留機群中建立、修改，以及取消容量保留。如需詳細資訊，請參閱使用容量保留機群的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2CapacityReservationFleetRolePolicy。

AWS 受管政策： AWSEC2FleetServiceRolePolicy

此政策連接到名為 AWSServiceRoleForEC2Fleet 的服務連結角色，允許 EC2 Fleet 代表您請求、啟動、終止和標記執行個體。如需詳細資訊，請參閱EC2 Fleet 的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2FleetServiceRolePolicy。

AWS 受管政策： AWSEC2SpotFleetServiceRolePolicy

此政策連接到名為 AWSServiceRoleForEC2SpotFleet 的服務連結角色，允許 Spot Fleet 代表您啟動和管理執行個體。如需詳細資訊，請參閱Spot Fleet 的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2SpotFleetServiceRolePolicy。

AWS 受管政策： AWSEC2SpotServiceRolePolicy

此政策連接到名為 AWSServiceRoleForEC2Spot 的服務連結角色，允許 Amazon EC2 代表您啟動和管理 Spot 執行個體。如需詳細資訊，請參閱Spot 執行個體請求的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2SpotServiceRolePolicy。

AWS 受管政策： AWSEC2VssSnapshotPolicy

您可以將此受管政策連接至您用於 Amazon EC2 Windows 執行個體的 IAM 執行個體設定檔角色。政策授予許可，允許 Amazon EC2 代表您建立和管理 VSS 快照。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2VssSnapshotPolicy。

AWS 受管政策： DeclarativePoliciesEC2Report

此政策可連接至名稱為 `AWSServiceRoleForDeclarativePoliciesEC2Report` 的服務連結角色，以便提供生成宣告式政策的帳戶狀態報告所需的唯讀 API 存取權。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 DeclarativePoliciesEC2Report。

AWS 受管政策： EC2FastLaunchFullAccess

您可以將 `EC2FastLaunchFullAccess` 政策連接至執行個體設定檔或其他 IAM 角色。此政策授予 EC2 Fast Launch 動作的完整存取權，以及目標許可，如下所示。

許可詳細資訊

- EC2 快速啟動 – 授予管理存取權，讓角色可以啟用或停用 EC2 快速啟動，並描述 EC2 快速啟動映像。
- Amazon EC2 – 授予 Amazon EC2 RunInstances、CreateTags、描述，以及建立和修改啟動範本作業的存取權。此外還會授予建立網路及安全群組資源、授權傳入規則，以及刪除 EC2 Fast Launch 所建立資源的存取權。
- IAM – 授予存取權，以取得和使用名稱包含 `ec2fastlaunch` 的執行個體描述檔來建立 EC2FastLaunchServiceRolePolicy 服務連結角色。
- CloudFormation – 授予 EC2 Fast Launch 描述及建立 CloudFormation 堆疊，以及刪除所建立堆疊的存取權。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 EC2FastLaunchFullAccess。

AWS 受管政策： AWSEC2CapacityManagerServiceRolePolicy

此政策會連接到名為 的服務連結角色AWSServiceRoleForEC2CapacityManager，以允許 EC2 Capacity Manager 代表您管理容量資源並與 AWS Organizations 整合。如需詳細資訊，請參閱 EC2 容量管理器的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSEC2CapacityManagerServiceRolePolicy。

AWS 受管政策： EC2FastLaunchServiceRolePolicy

此政策連接至名為 AWSServiceRoleForEC2FastLaunch 的服務連結角色，以允許 Amazon EC2 建立和管理一組預先佈建的快照，從而減少啟用 EC2 Fast Launch 的 AMI 啟動執行個體所需的時間。如需詳細資訊，請參閱the section called “服務連結角色”。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 EC2FastLaunchServiceRolePolicy。

AWS 受管政策： Ec2InstanceConnect

您可將 `Ec2InstanceConnect` 政策連接到 IAM 身分。此政策授予許可，允許客戶呼叫 EC2 Instance Connect，將暫時性金鑰發佈至其 EC2 執行個體，並透過 SSH 或 EC2 Instance Connect CLI 連線。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 Ec2InstanceConnect。

AWS 受管政策： Ec2InstanceConnectEndpoint

此政策會連接到名為 AWSServiceRoleForEC2InstanceConnect 的服務連結角色，它允許 EC2 Instance Connect 端點代表您執行動作。如需詳細資訊，請參閱EC2 Instance Connect 端點的服務連結角色。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 Ec2InstanceConnectEndpoint。若要了解此政策更新的相關說明，請參閱 AWS 受管政策的 Amazon EC2 更新。

AWS 受管政策的 Amazon EC2 更新

檢視自此服務開始追蹤 Amazon EC2 AWS 受管政策更新以來的詳細資訊。

變更	描述	日期
AWSEC2CapacityManagerServiceRolePolicy – 新政策	Amazon EC2 新增了此政策，可讓您代表您管理容量資源並與 AWS Organizations 整合。	2025 年 10 月 15 日
AmazonEC2ImageReferencesAccessPolicy – 新政策	Amazon EC2 新增了此政策，以便提供掃描 EC2 DescribeImageReferences API 支援的全部資源類型的許可。	2025 年 8 月 26 日
Ec2InstanceConnectEndpoint - 更新的政策	為支援修改現有的 Instance Connect 端點，Amazon EC2 更新了此政策，以新增在 EC2 Instance Connect 端點建立的網路介面上指派及取消指派 IPv6 位址，以及修改安全群組的許可。Amazon EC2 亦更新了此政策，以便能使用 `StringLike` 條件運算子來取代 `Null` 條件運算子。	2025 年 7 月 31 日
EC2FastLaunchServiceRolePolicy - 更新的政策	為協助防止資源孤立，Amazon EC2 更新了此政策，以新增	2025 年 7 月 17 日

變更	描述	日期
	描述磁碟區、磁碟區屬性及網路介面，以及刪除 EC2 Fast Launch 所建立磁碟區及網路介面的許可。	
EC2FastLaunchFullAccess - 更新的政策	Amazon EC2 更新了此政策，以便包括建立及修改啟動範本作業、建立網路及安全群組資源、授權傳入規則，以及刪除 EC2 Fast Launch 所建立的資源。另外，該服務還可描述及建立 CloudFormation 堆疊，以及刪除 EC2 Fast Launch 建立的堆疊。	2025 年 5 月 14 日
EC2FastLaunchServiceRolePolicy - 更新的政策	Amazon EC2 更新了此政策，以便新增 Amazon EventBridge 建立及管理 EC2 Fast Launch 事件規則的存取權。此外，EC2 Fast Launch 現在可以描述 CloudFormation 堆疊、從與 相關聯的 AMI 啟動執行個體 AWS License Manager、取得其建立的可淘汰的 AWS KMS 授予清單，以及刪除其建立的啟動範本。	2025 年 5 月 14 日
AWSEC2CapacityReservationFleetRolePolicy – 更新的許可	Amazon EC2 更新了 `AWSEC2CapacityReservationFleetRolePolicy` 受管政策，可使用 `ArnLike` 條件運算子而非 `StringLike` 條件運算子。	2025 年 3 月 3 日

變更	描述	日期
AmazonEC2ReadOnlyAccess：新增許可	Amazon EC2 新增了可讓您使用 `GetSecurityGroupsForVpc` 作業來擷取安全群組的許可。	2024 年 12 月 27 日
EC2FastLaunchFullAccess – 新政策	Amazon EC2 新增此政策，以從執行個體執行與 EC2 快速啟動功能相關的 API 動作。政策可以連接到從已啟用 EC2 快速啟動的 AMI 啟動之執行個體的執行個體設定檔。	2024 年 5 月 14 日
AWSEC2VssSnapshotPolicy – 新政策	Amazon EC2 已新增 `AWSEC2VssSnapshotPolicy` 政策，其中包含建立標籤並將其新增至 Amazon Machine Image (AMIs) 和 EBS 快照的許可。	2024 年 3 月 28 日
Ec2InstanceConnectEndpoint – 新政策	Amazon EC2 新增了 `Ec2InstanceConnectEndpoint` 政策。此政策可連接至 AWSServiceRoleForEC2InstanceConnect 服務連結角色，以便您建立 EC2 Instance Connect 端點時，允許 Amazon EC2 代表您執行操作。	2023 年 1 月 24 日
EC2FastLaunchServiceRolePolicy – 新政策	Amazon EC2 新增了 EC2 Fast Launch 功能，其透過建立一組預先佈建的快照，讓 Windows AMI 能夠更快速地啟動執行個體。	2021 年 11 月 26 日

變更	描述	日期
Amazon EC2 開始追蹤變更	Amazon EC2 開始追蹤其 AWS 受管政策的變更	2021 年 3 月 1 日

Amazon EC2 的 IAM 的角色

應用程式必須使用 AWS 登入資料簽署其 API 請求。因此，若您是應用程式開發人員，對於在 EC2 執行個體上運作的應用程式，您會需要其登入資料的管理策略。例如，您可安全將 AWS 登入資料分配給執行個體，讓在這些執行個體上運作的應用程式能夠使用此登入資料來簽署請求，同時避免其他使用者取得您的登入資料。不過，將登入資料安全地分發給每個執行個體是一項挑戰，尤其是代表您 AWS 建立的執行個體，例如 Spot 執行個體或 Auto Scaling 群組中的執行個體。當您輪換登入資料時，也必須能夠更新每個執行個體上的 AWS 登入資料。

我們設計的 IAM 角色，讓您的應用程式能夠從執行個體安全提出 API 請求，您無須管理應用程式使用的安全登入資料。您可以委派許可使用 IAM 角色提出 API 請求，而不是建立和分發您的 AWS 登入資料，如下所示：

1. 建立 IAM 角色。
2. 定義哪些帳戶 AWS 或服務可以擔任該角色。
3. 定義應用程式在角色擔任後可使用的 API 動作與資源。
4. 啟動執行個體時指定該角色，或將該角色連接至現有執行個體。
5. 讓應用程式擷取一組暫時登入資料並加以使用。

例如，若執行於您執行個體上的應用程式需要使用 Amazon S3 內的儲存貯體，您可使用 IAM 角色為其授予許可。您可建立 JSON 格式的政策，藉此指定 IAM 角色的許可。這些政策類似於您為使用者建立的政策。若您變更角色，此變更會傳播至所有執行個體。

> **Note**
>
> Amazon EC2 IAM 角色登入資料不受角色中設定的工作階段持續時間上限限制。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

建立 IAM 角色時，請與最低權限 IAM 政策建立關聯，以限制對應用程式所需之特定 API 呼叫的存取。對於 Windows 至 Windows 通訊，請使用妥善定義且妥善記載的 Windows 群組和角色，授與

Windows 執行個體之間的應用程式層級存取權。群組和角色可讓客戶定義最低權限應用程式和 NTFS 資料夾層級許可，以限制對應用程式特定需求的存取。

雖然您只能將一個 IAM 角色連接到一個執行個體中，但您可以將相同的角色連接到多個執行個體中。如需建立 IAM 角色的詳細資訊，請參閱IAM 使用者指南中的角色。

您可將資源層級許可套用至 IAM 政策，藉此控制使用者連接、取代或分離執行個體的 IAM 角色的能力。如需詳細資訊，請參閱 Amazon EC2 API 動作支援的資源層級許可及下列範例：範例：使用 IAM 角色。

內容

- 執行個體描述檔
- 許可使用案例
- 從執行個體中繼資料中擷取安全登入資料
- 授予 IAM 使用者將 IAM 角色傳遞至執行個體的許可
- 將 IAM 角色連接至執行個體
- Amazon EC2 執行個體的執行個體身分角色

執行個體描述檔

Amazon EC2 會使用執行個體描述檔做為 IAM 角色的容器。使用 IAM 主控台建立 IAM 角色時，主控台會自動建立執行個體描述檔，並將其命名為與對應角色相同的名稱。若您使用 Amazon EC2 主控台來啟動具備 IAM 角色的執行個體，或使用該主控台將 IAM 角色連接至執行個體，您須根據執行個體描述檔名稱清單來選擇角色。

如果您使用 AWS CLI、 API 或 AWS SDK 來建立角色，則可以將角色和執行個體描述檔建立為個別動作，名稱可能不同。如果您接著使用 AWS CLI、 API 或 AWS 開發套件來啟動具有 IAM 角色的執行個體，或將 IAM 角色連接至執行個體，請指定執行個體設定檔名稱。

執行個體描述檔內僅有一個 IAM 角色。您可在多個執行個體設定檔中包括 IAM 角色。

如需更新執行個體的許可，請取代其執行個體設定檔。不建議從執行個體設定檔中移除角色，因為最多需要延遲一小時，此變更才生效。

如需詳細資訊，請參閱 IAM 使用者指南 中的使用執行個體設定檔。

許可使用案例

當您第一次為應用程式建立 IAM 角色時，有時可能會授予超出所需的許可。在生產環境中啟動應用程式之前，您可以根據 IAM 角色的存取活動產生 IAM 政策。IAM Access Analyzer 會檢閱您的 AWS CloudTrail 日誌，並產生政策範本，其中包含角色在指定日期範圍內已使用的許可。您可以使用範本建立具有精細許可的受管政策，然後將其連接至 IAM 角色。如此一來，您只會授予角色與特定使用案例 AWS 的資源互動所需的許可。這有助於您符合授予最低權限的最佳實務。如需更多資訊，請參閱「IAM 使用者指南」中的 IAM Access Analyzer 政策驗證。

從執行個體中繼資料中擷取安全登入資料

執行個體上的應用程式會從執行個體中繼資料項目 `iam/security-credentials/role-name`，擷取角色提供的安全登入資料。如此即授予該應用程式許可，能夠執行您在建立角色與安全登入資料的關聯時所定義的角色動作與資源。這些安全登入資料為暫時的，我們會定期更換。舊的登入資料過期前至少五分鐘，我們就會提供新的登入資料。

如需執行個體中繼資料的詳細資訊，請參閱使用執行個體中繼資料管理您的 EC2 執行個體。

> **Warning**
>
> 若您使用的服務會運用具備 IAM 角色的執行個體中繼資料，請確認該服務代您發出 HTTP 呼叫時不會暴露您的登入資料。可能會暴露登入資料的服務類型包括 HTTP 代理、HTML/CSS 驗證器服務和支援納入 XML 的 XML 處理器。

對於 Amazon EC2 工作負載，建議您使用下述方法擷取工作階段憑證。這些憑證應該可讓您的工作負載發起 AWS API 請求，而無需使用 `sts:AssumeRole` 擔任已與執行個體相關聯的相同角色。除非您需要為屬性型存取控制 (ABAC) 傳遞工作階段標籤，或需要傳遞工作階段政策以進一步限制角色權限，否則此類角色擔任呼叫是不必要的，因為它們會建立一組新的相同的臨時角色工作階段憑證。

如果您的工作負載使用角色來擔任它自己，則必須建立信任政策，明確允許該角色擔任它自己。如果您不建立信任政策，會得到 `AccessDenied` 錯誤。如需了解詳細資訊，請參閱 IAM 使用者指南中的更新角色信任政策。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令，可擷取 IAM 角色的安全憑證。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

透過 Windows 執行個體執行下面的 cmdlet，可擷取 IAM 角色的安全憑證。

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令，可擷取 IAM 角色的安全憑證。

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

透過 Windows 執行個體執行下面的 cmdlet，可擷取 IAM 角色的安全憑證。

```
Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

以下為範例輸出。如果您無法擷取安全登入資料，請參閱「IAM 使用者指南」中的無法存取 EC2 執行個體上的臨時安全登入資料。

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
```

```
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

對於在執行個體上執行的應用程式 AWS CLI、 和 Tools for Windows PowerShell 命令，您不需要明確取得臨時安全登入資料 - Windows PowerShell AWS SDKs AWS CLI、 和 Tools 會自動從 EC2 執行個體中繼資料服務取得登入資料並使用它們。欲使用暫時安全登入資料在執行個體外部進行呼叫 (如為了測試 IAM 政策)，您必須提供存取金鑰、私有金鑰和工作階段字符。如需詳細資訊，請參閱《IAM 使用者指南》中的使用暫時安全登入資料來請求存取 AWS 資源。

授予 IAM 使用者將 IAM 角色傳遞至執行個體的許可

IAM 使用者等 中的身分必須具有特定許可 AWS 帳戶，才能使用 IAM 角色啟動 Amazon EC2 執行個體、將 IAM 角色連接至執行個體、取代執行個體的 IAM 角色，或從執行個體分離 IAM 角色。您必須視需要授予使用下列 API 動作的許可：

- `iam:PassRole`
- `ec2:AssociateIamInstanceProfile`
- `ec2:DisassociateIamInstanceProfile`
- `ec2:ReplaceIamInstanceProfileAssociation`

> **Note**
>
> 如果您將 `iam:PassRole` 資源指定為 `*`，這將授予您的任何 IAM 角色傳遞給執行個體的存取權限。若要遵循最低權限的最佳實務，請使用 `iam:PassRole` 指定特定 IAM 角色ARNs，如以下範例政策所示。

程式設計存取的範例政策

下列 IAM 政策授予許可，以使用 IAM 角色啟動執行個體、將 IAM 角色連接至執行個體，或使用 AWS CLI 或 Amazon EC2 API 取代執行個體的 IAM 角色。

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances",
         "ec2:AssociateIamInstanceProfile",
         "ec2:DisassociateIamInstanceProfile",
         "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/DevTeam*"
    }
  ]
}
```

主控台存取的其他要求

若要授予使用 Amazon EC2 主控台完成相同任務的許可，您還必須包含 `iam:ListInstanceProfiles` API 動作。

將 IAM 角色連接至執行個體

您可建立 IAM 角色，並在啟動期間或之後將其連接至執行個體。您也可以取代或分離 IAM 角色。

在執行個體啟動期間建立和連接 IAM 角色（建議）

1. 在 EC2 執行個體啟動期間，展開進階詳細資訊。
2. 在 IAM 執行個體設定檔區段中，選擇建立新的 IAM 角色。
3. 會開啟內嵌角色建立表單，讓您：

   - 指定角色名稱（例如 `EC2-S3-Access-Role`)
   - 透過選取 AWS 受管政策或為您的執行個體建立自訂政策來定義許可

     例如，若要授予 S3 存取權，請選取 `AmazonS3ReadOnlyAccess` 受管政策
   - 檢閱允許 `ec2.amazonaws.com`擔任角色的信任政策

- 新增中繼資料的選用標籤

4. 選擇建立角色。

   系統會自動選取新建立的角色，並在執行個體啟動時透過執行個體描述檔連接至您的執行個體。

> **Note**
>
> 當您在執行個體啟動期間使用主控台建立角色時，會自動建立與角色同名的執行個體描述檔。執行個體描述檔是在啟動時將 IAM 角色資訊傳遞給執行個體的容器。

> **Important**
>
> - 雖然您只能將一個 IAM 角色連接到一個執行個體中，但您可以將相同的角色連接到多個執行個體中。
> - 關聯限制存取應用程式所需特定 API 呼叫的最低權限 IAM 政策。

如需建立 IAM 角色的詳細資訊，請參閱IAM 使用者指南中的角色。

在執行個體啟動期間連接現有的 IAM 角色

若要使用 Amazon EC2 主控台在啟動時將現有的 IAM 角色連接至執行個體，請展開進階詳細資訊。針對 IAM 執行個體描述檔，從下拉式清單中選取 IAM 角色。

> **Note**
>
> 若您已使用主控台建立 IAM 角色，則會為您建立執行個體設定檔，且其名稱會與該角色相同。如果您使用 AWS CLI、 API 或 AWS SDK 建立 IAM 角色，您可能已為執行個體描述檔指定與該角色不同的名稱。

您可以將 IAM 角色連接至正在執行或已停止的執行個體。如果執行個體已連接 IAM 角色，您必須將其取代為新的 IAM 角色。

Console

將 IAM 角色連接至執行個體 (主控台)

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇 Actions (動作)、Security (安全性)、Modify IAM role (修改 IAM 角色)。
5. 為 IAM 角色，選擇 IAM 執行個體設定檔。
6. 選擇更新 IAM 角色。

AWS CLI

將 IAM 角色連接至執行個體 (主控台)

使用 associate-iam-instance-profile 命令，將 IAM 角色連接至執行個體。當您指定執行個體設定檔時，您可以使用執行個體設定檔的 Amazon Resource Name (ARN)，也可以使用其名稱。

```
aws ec2 associate-iam-instance-profile \
    --instance-id i-1234567890abcdef0 \
    --iam-instance-profile Name="TestRole-1"
```

PowerShell

將 IAM 角色連接至執行個體 (主控台)

使用 Register-EC2IamInstanceProfile cmdlet。

```
Register-EC2IamInstanceProfile `
    -InstanceId i-1234567890abcdef0 `
    -IamInstanceProfile_Name TestRole-1
```

若要取代已擁有連接 IAM 角色之執行個體上的 IAM 角色，則執行個體必須處於執行中狀態。如果您想要變更執行個體上的 IAM 角色，但不中斷現有的角色，您可以這麼做。例如，您可以這麼做確保執行個體上應用程式執行的 API 動作不會中斷。

Console

取代執行個體的 IAM 角色

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇 Actions (動作)、Security (安全性)、Modify IAM role (修改 IAM 角色)。
5. 為 IAM 角色，選擇 IAM 執行個體設定檔。
6. 選擇更新 IAM 角色。

AWS CLI

取代執行個體的 IAM 角色

1. 視需要使用 describe-iam-instance-profile-associations 命令來獲取關聯 ID。

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. 請使用 replace-iam-instance-profile-association 命令。指定現有執行個體設定檔的關聯 ID，以及新的執行個體設定檔的 ARN 或名稱。

```
aws ec2 replace-iam-instance-profile-association \
    --association-id iip-assoc-0044d817db6c0a4ba \
    --iam-instance-profile Name="TestRole-2"
```

PowerShell

取代執行個體的 IAM 角色

1. 視需要使用 Get-EC2IamInstanceProfileAssociation cmdlet 來獲取關聯 ID。

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. 使用 Set-EC2IamInstanceProfileAssociation cmdlet。指定現有執行個體設定檔的關聯 ID，以及新的執行個體設定檔的 ARN 或名稱。

```
Set-EC2IamInstanceProfileAssociation `
    -AssociationId iip-assoc-0044d817db6c0a4ba `
    -IamInstanceProfile_Name TestRole-2
```

您可透過執行中或已停止的執行個體來分離 IAM 角色。

Console

分離 IAM 角色和執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇 Actions (動作)、Security (安全性)、Modify IAM role (修改 IAM 角色)。
5. 針對 IAM role (IAM 角色)，選擇 No IAM Role (無 IAM 角色)。
6. 選擇更新 IAM 角色。
7. 出現確認提示時，請輸入分離，然後選擇分離。

AWS CLI

分離 IAM 角色和執行個體

1. 視需要使用 describe-iam-instance-profile-associations，來獲取 IAM 執行個體設定檔欲分離的關聯 ID。

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. 請使用 disassociate-iam-instance-profile 命令。

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-0044d817db6c0a4ba
```

PowerShell

分離 IAM 角色和執行個體

1. 視需要使用 Get-EC2IamInstanceProfileAssociation，來獲取 IAM 執行個體設定檔欲分離的關聯 ID。

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. 使用 Unregister-EC2IamInstanceProfile cmdlet。

```
Unregister-EC2IamInstanceProfile -AssociationId iip-assoc-0044d817db6c0a4ba
```

Amazon EC2 執行個體的執行個體身分角色

您啟動的每個 Amazon EC2 執行個體都有一個代表其身分的執行個體身分角色。執行個體身分角色是一種 IAM 角色。整合使用執行個體身分角色的服務 AWS 和功能，可以使用它來識別服務的執行個體。

您可以從 /identity-credentials/ec2/security-credentials/ec2-instance 的執行個體中繼資料服務 (IMDS) 存取執行個體身分角色憑證。登入資料包含 AWS 臨時存取金鑰對和工作階段字符。它們用於向使用執行個體身分角色 AWS 的服務簽署 AWS Sigv4 請求。無論執行個體上是否已啟用使用執行個體身分角色的服務或功能，憑證都會顯示在執行個體中繼資料中。

執行個體身分角色在執行個體啟動時自動建立，沒有角色信任政策文件，且不受任何身分或資源政策的約束。

支援的服務

下列 AWS 服務使用執行個體身分角色：

- Amazon Bedrock AgentCore – AgentCore 執行期執行個體使用執行個體身分角色在啟動期間設定執行個體。
- Amazon EC2 – EC2 執行個體連線使用執行個體身分角色來更新 Linux 執行個體的主機金鑰。
- Amazon GuardDuty – GuardDuty 執行時期監控會使用執行個體身分角色，允許執行時期代理程式將安全性遙測資料傳送至 GuardDuty VPC 端點。
- AWS Lambda – Lambda 受管執行個體會將執行個體身分角色用於生命週期掛鉤、遙測和成品分佈。

- AWS Security Token Service (AWS STS) – 執行個體身分角色登入資料可與 AWS STS `GetCallerIdentity`動作搭配使用。
- AWS Systems Manager – 使用預設主機管理組態時， AWS Systems Manager 會使用執行個體身分角色提供的身分來註冊 EC2 執行個體。識別執行個體之後，Systems Manager 可將 `AWSSystemsManagerDefaultEC2InstanceManagementRole` IAM 角色傳遞給執行個體。

執行個體身分角色無法與其他 AWS 服務或功能搭配使用，因為它們沒有與執行個體身分角色的整合。

執行個體身分角色 ARN

執行個體身分角色 ARN 採用下列格式：

```
arn:aws-partition:iam::account-number:assumed-role/aws:ec2-instance/instance-id
```

例如：

```
arn:aws:iam::0123456789012:assumed-role/aws:ec2-instance/i-1234567890abcdef0
```

如需 ARN 的詳細資訊，請參閱「IAM 使用者指南」中的 Amazon Resource Name (ARN)。

Amazon EC2 執行個體的更新管理

建議您定期修補、更新和保護 EC2 執行個體上的作業系統與應用程式。您可以使用 AWS Systems Manager Patch Manager 會針對作業系統及應用程式自動化安裝安全性相關更新的程序。

對於 Auto Scaling 群組中的 EC2 執行個體，您可以使用 AWS-PatchAsgInstance 執行手冊協助避免正在進行修補的執行個體遭到取代。或者，您也可以使用任何自動更新服務或建議的程序，以安裝應用程式廠商提供的更新。

Resources

- AL2023 – Amazon Linux AL2023中的更新 AL2023
- AL2 – Amazon Linux 2 使用者指南中的管理 Amazon Linux 2 執行個體上的軟體
- Windows 執行個體 – the section called “更新管理”

Windows 執行個體的安全性最佳做法

我們建議您遵循這些 Windows 執行個體的安全最佳實務。

目錄

- 高階安全最佳做法
- 更新管理
- 組態管理
- 變更管理
- Amazon EC2 Windows 執行個體的稽核和責任

高階安全最佳做法

您應該為您的 Windows 執行個體遵守以下進階安全最佳做法：

- 最低存取權：僅授與信任和預期之系統和位置的存取權。這適用於所有 Microsoft 產品，例如 Active Directory、Microsoft 商務生產力伺服器，以及基礎設施服務，例如遠端桌面服務，反向代理伺服器、IIS Web 伺服器等。使用 Amazon EC2 執行個體安全群組、網路存取控制清單 (ACLs) 和 Amazon VPC 公有/私有子網路等 AWS 功能，在 架構中的多個位置分層安全性。在 Windows 執行個體內，客戶可以使用 Windows 防火牆，在其部署內進一步將深度防禦策略進行分層。僅安裝系統依設計運作所需的作業系統元件和應用程式。將基礎設施服務 (例如 IIS) 設定為在服務帳戶下執行，或使用應用程式集區身分這類功能，在本機和遠端存取整個基礎設施中的資源。
- 最低權限 – 決定執行個體和帳戶執行其函數所需的最低權限集。限制這些伺服器和使用者只允許這些已定義的許可。使用角色型存取控制這類技術來減少管理帳戶的表面區域，並建立最受限的角色來完成任務。使用作業系統功能 (例如 NTFS 內的加密檔案系統 (EFS))，來加密敏感的敏感資料，並控制應用程式和使用者對它的存取。
- 組態管理：建立基準伺服器組態，其中納入最新的安全性修補程式，以及主機型保護套件，包括防毒、反惡意軟體、入侵偵測/預防，以及檔案完整性監控。根據目前記錄的基準來評定每部伺服器，以識別並標示任何偏差。確保每部伺服器都已設定為產生並安全地存放適當的日誌和稽核資料。
- 變更管理：建立程序來控制伺服器組態基準的變更，並朝著完全自動化的變更程序運作。此外，運用 Just Enough Administration (JEA) 與 Windows PowerShell DSC，將管理存取權限制為最低必要功能。
- 修補程式管理：執行定期修補、更新和保護您 EC2 執行個體上的作業系統與應用程式的程序。
- 稽核日誌：稽核 Amazon EC2 執行個體的存取和所有變更，以驗證伺服器的完整性，並確保僅進行授權的變更。利用 增強型記錄 IIS 等功能來增強預設記錄功能。 AWS 和 等功能 AWS CloudTrail 也可用於稽核網路存取，包括允許/拒絕的請求和 API 呼叫。

更新管理

為了確保在 Amazon EC2 上執行 Windows Server 時獲得最佳結果，我們建議您實施以下最佳做法：

- Configure Windows Update
- Update drivers
- Use the latest Windows AMIs
- Test performance before migration
- Update launch agents
- 您可在安裝更新後重新開機 Windows 執行個體。如需詳細資訊，請參閱重新啟動您的 Amazon EC2 執行個體。

如需如何將 Windows 執行個體升級或遷移至較新版 Windows Server 的資訊，請參閱將 EC2 Windows 執行個體升級至較新版本的 Windows Server。

設定 Windows Update

根據預設，從 AWS Windows Server AMIs 啟動的執行個體不會透過 Windows Update 接收更新。

更新 Windows 驅動程式

在所有 Windows EC2 執行個體上維護最新的驅動程式，以確保在機群中套用最新的問題修正和效能增強。根據您的執行個體類型，您應該更新 AWS PV、Amazon ENA 和 AWS NVMe 驅動程式。

- 使用 SNS 主題接收新驅動程式的版本更新。
- 使用 AWS Systems Manager Automation Runbook AWSSupport-UpgradeWindowsAWSDrivers 輕鬆將更新套用至您的執行個體。

使用最新 Windows AMI 來啟動執行個體

AWS 每月發行新的 Windows AMIs，其中包含最新的作業系統修補程式、驅動程式和啟動代理程式。在啟動新執行個體或建構自己的自訂映像時，您應運用最新的 AMI。

- 若要檢視每個 AWS Windows AMIs 版本的更新，請參閱 AWS Windows AMI 版本歷史記錄。
- 若要建構最新的可用 AMI，請參閱使用 Systems Manager 參數存放區查詢最新的 Windows AMI。
- 如需可用於啟動資料庫執行個體和合規強化使用案例的專門 Windows AMIs 詳細資訊，請參閱 Windows AMIs 參考中的專門「AWS Windows AMI」。

在遷移前測試系統/應用程式效能

將企業應用程式遷移至 AWS 可以涉及許多變數和組態。一律對 EC2 解決方案執行效能測試以確保：

- 已正確配置執行個體類型，包括執行個體大小、增強型聯網和租用 (共用或專用)。
- 執行個體拓撲適用於工作負載，並在必要時運用高效能功能，例如專用租用、置放群組、執行個體儲存體磁碟區、裸機。

安裝最新版本的 EC2Launch v2

安裝最新的 EC2Launch v2 代理程式，以確保您的機群中套用最新的增強功能。如需詳細資訊，請參閱安裝 EC2Launch v2。

如果您有混合機群，或想要繼續使用 EC2Launch (Windows Server 2016 和 2019) 或 EC2 Config (僅限舊版作業系統) 代理程式，則請更新至各自代理程式的最新版本。

下列 Windows Server 版本和啟動代理程式組合支援自動更新。您可以在 Amazon EC2 啟動代理程式下的 SSM 快速設定主機管理主控台中選擇自動更新。

Windows 版本	EC2Launch v1	EC2Launch v2
2016	✓	✓
2019	✓	✓
2022		✓

- 如需 EC2Launch v2 的詳細資訊，請參閱 the section called “安裝 EC2Launch v2”。
- 如需手動更新 EC2Config 的資訊，請參閱the section called “安裝 EC2Config”。
- 如需手動更新 EC2Launch 的資訊，請參閱the section called “安裝 EC2Launch”。

組態管理

Amazon Machine Image (AMI) 為 Amazon EC2 執行個體提供初始配置，其中包括 Windows 作業系統和選用的客戶特定自訂，例如應用程式和安全控制。建立包含自訂安全組態基準的 AMI 目錄，以確保所有 Windows 執行個體都搭配標準安全控制來啟動。安全基準可以分為 AMI、啟動 EC2 執行個體時動態引導，或封裝為透過 AWS Service Catalog 產品組合統一分佈的產品。如需保護 AMI 的詳細資訊，請參閱建置 AMI 的最佳實務。

每個 Amazon EC2 執行個體都應該遵守組織安全標準。不要勿安裝任何不需要的 Windows 角色和功能，並安裝軟體以防範惡意程式碼 (防毒、反惡意軟體、漏洞防護)、監控主機完整性，以及執行入侵偵測。設定安全軟體以監控和維護作業系統安全設定、保護重要作業系統檔案的完整性，以及在偏離安全基準時發出提醒。考慮實作由 Microsoft、網際網路安全中心 (CIS) 或國家標準技術研究所 (NIST) 所發佈的建議安全組態基準。考慮使用其他適用於特定應用程式伺服器的 Microsoft 工具，例如 Best Practice Analyzer for SQL Server。

AWS 客戶也可以執行 Amazon Inspector 評估，以提高部署在 Amazon EC2 執行個體上應用程式的安全性和合規性。Amazon Inspector 會自動評定應用程式是否有漏洞或偏離最佳實務，並包含一個知識庫，其中有數百個映射至常見安全合規性標準 (例如 PCI DSS) 和漏洞定義的規則。內建規則的範例包括檢查是否已啟用遠端根登入，或是否已安裝易受攻擊的軟體版本。 AWS 安全研究人員會定期更新這些規則。

保護 Windows 執行個體時，我們建議您實作 Active Directory Domain Services，為分散式位置啟用可擴展、安全且可管理的基礎設施。此外，從 Amazon EC2 主控台或使用 Amazon EC2 佈建工具啟動執行個體後 AWS CloudFormation，最好使用原生作業系統功能，例如 Microsoft Windows PowerShell DSC，以在發生組態偏離時維持組態狀態。

變更管理

在啟動時將初始安全基準套用至 Amazon EC2 執行個體之後，系統會控制持續的 Amazon EC2 變更以維護虛擬機器的安全。建立變更管理程序，以授權和納入 AWS 資源 （例如安全群組、路由表和網路 ACLs) 以及作業系統和應用程式組態 （例如 Windows 或應用程式修補、軟體升級或組態檔案更新） 的變更。

AWS 提供數種工具來協助管理 AWS 資源的變更，包括 Systems Center Operations Manager 和 System Center Virtual Machine Manager 的 AWS CloudTrail AWS Config CloudFormation AWS Elastic Beanstalk、 和 和管理套件。請注意，Microsoft 會在每個月的第二個星期二 （或視需要） 發行 Windows 修補程式，並在 Microsoft 發行修補程式後五 AWS 天內 AWS 更新由 管理的所有 Windows AMIs。因此，請務必持續修補所有基準 AMIs、使用最新的 AMI IDs 更新 CloudFormation 範本和 Auto Scaling 群組組態，並實作工具來自動化執行中的執行個體修補程式管理。

Microsoft 提供了數個選項，用於管理 Windows 作業系統和應用程式變更。例如，SCCM 提供環境修改的完整生命週期涵蓋範圍。選取處理商務需求的工具，並控制變更如何影響應用程式 SLA、容量、安全性和災難復原程序。避免手動變更，而是運用自動化組態管理軟體或命令列工具 (例如 EC2 執行命令或 Windows PowerShell)，來實作以指令碼執行且可重複的變更程序。為了協助滿足此需求，請將具有增強記錄功能的堡壘主機用於與 Windows 執行個體的所有互動，以確保自動記錄所有事件和任務。

Amazon EC2 Windows 執行個體的稽核和責任

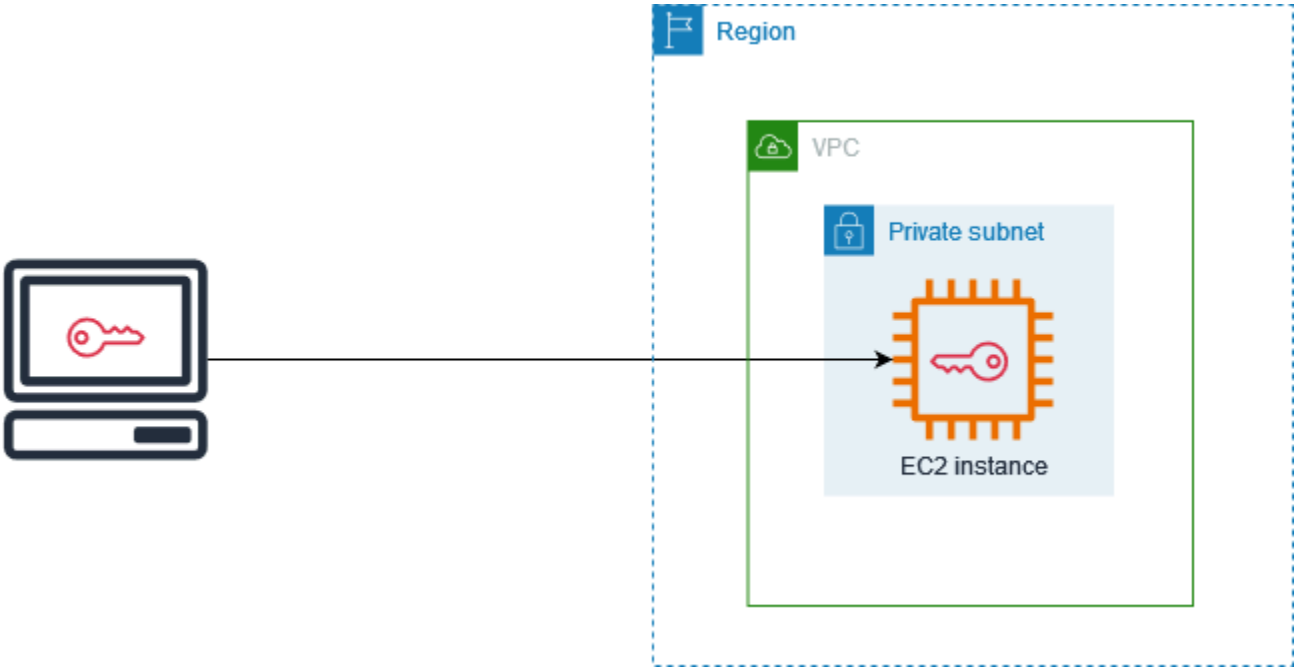
AWS CloudTrail AWS Config， AWS Config 規則 並提供稽核和變更追蹤功能，以稽核 AWS 資源變更。將 Windows 事件日誌設定為將本機日誌檔傳送至集中式日誌管理系統，來保留日誌資料以進行安全性和操作行為分析。Microsoft System Center Operations Manager (SCOM) 可彙總部署到 Windows 執行個體之 Microsoft 應用程式的相關資訊，並根據應用程式角色和服務套用預先設定和自訂的規則集。System Center 管理套件建置在 SCOM 上，以提供應用程式特定的監控和設定指引。這些管理套件支援 Windows Server Active Directory、SharePoint Server 2013、Exchange Server 2013、Lync Server 2013、SQL Server 2014，以及許多其他的伺服器和技術。

除了 Microsoft 系統管理工具之外，客戶還可以使用 Amazon CloudWatch 來監控執行個體 CPU 使用率、磁碟效能、網路 I/O，以及執行主機和執行個體狀態檢查。EC2Config、EC2Launch 和 EC2Launch v2 啟動代理程式提供對 Windows 執行個體的附加進階功能的存取。例如，它們可以將 Windows 系統、安全性、應用程式和 Internet Information Services (IIS) 日誌匯出至 CloudWatch Logs，然後與 Amazon CloudWatch 指標和警示整合。客戶也可以建立將 Windows 效能計數器匯出至 Amazon CloudWatch 自訂指標的指令碼。

Amazon EC2 金鑰對和 Amazon EC2 執行個體

金鑰對 (由公有金鑰和私有金鑰組成) 是一組安全登入資料，您可以在連線至 Amazon EC2 執行個體時用來證明身分。針對 Linux 執行個體，私有金鑰讓您可安全地將 SSH 傳送至執行個體。針對 Windows 執行個體，需要私有金鑰才能解密之後要用來連接執行個體的管理員密碼。

Amazon EC2 會在執行個體上儲存公有金鑰，而您可存放私有金鑰，如以下圖表所示。請務必將私有金鑰存放在安全之處，因為擁有私有金鑰的任何人都可連線至使用金鑰對的執行個體。



在您啟動執行個體時，可以指定金鑰對，以便使用需要金鑰對的方法來連線至執行個體。根據您管理安全性的方式，您可以為所有執行個體指定相同的金鑰對，也可以指定不同的金鑰對。

針對 Linux 執行個體，在您的執行個體第一次啟動時，啟動時所指定的公有金鑰會放置在 `~/.ssh/authorized_keys` 內某個條目中的 Linux 執行個體中。當您使用 SSH 連線到 Linux 執行個體時，若要登入，則必須指定與公有金鑰相對應的私有金鑰。

如需連線至 EC2 執行個體的詳細資訊，請參閱 連線至 EC2 執行個體。

> **Important**
>
> 由於 Amazon EC2 不會保留您私有金鑰的複本，因此，如果私有金鑰遺失了，就無法恢復。但是，仍然有一種方法可以連線到遺失私有金鑰的執行個體。如需詳細資訊，請參閱我的私有金鑰遺失。如何連線到我的執行個體？

做為金鑰對的替代方案，您可以使用 AWS Systems Manager Session Manager以互動式瀏覽器型 shell 或 AWS Command Line Interface () 連線至執行個體AWS CLI。

目錄

- 為您的 Amazon EC2 執行個體建立金鑰對
- 描述金鑰對
- 刪除您的金鑰對

- 在執行個體上新增或取代公有金鑰
- 驗證金鑰對的指紋

為您的 Amazon EC2 執行個體建立金鑰對

您可以使用 Amazon EC2 來建立金鑰對，或使用第三方工具產生金鑰對，然後再將其匯入 Amazon EC2。

Amazon EC2 可支援 Linux 和 Windows 執行個體的 2048位元 SSH-2 RSA 金鑰。Amazon EC2 也支援 Linux 執行個體上的 ED25519 金鑰。

如需有關如何在建立金鑰對後連線至執行個體的說明，請參閱 the section called “使用 SSH 連線至您的 Linux 執行個體” 和 the section called “使用 RDP 連線至您的 Windows 執行個體”。

目錄

- 使用 Amazon EC2 建立金鑰對
- 使用 建立金鑰對 AWS CloudFormation
- 使用第三方工具來建立金鑰對，然後將公有金鑰匯入 Amazon EC2

使用 Amazon EC2 建立金鑰對

使用 Amazon EC2 建立金鑰對時，公有金鑰會儲存在 Amazon EC2 中，而您可儲存私有金鑰。

每個區域最多可建立 5,000個金鑰對。若要申請增加，請建立支援案例。如需詳細資訊，請參閱 支援使用者指南中的建立支援案例。

Console

使用 Amazon EC2 建立金鑰對

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 請在導覽窗格的 Network & Security (網路與安全性) 下方，選擇 Key Pairs (金鑰對)。
3. 選擇 Create key pair (建立金鑰對)。
4. 對於 Name (名稱)，輸入金鑰對的描述性名稱。Amazon EC2 會將公有金鑰與您指定為金鑰名稱的名稱相關聯。金鑰名稱最多可包含 255 個 ASCII 字元。它不能包括前置或結尾空格。
5. 選取適合您作業系統的金鑰對類型：

   (Linux 執行個體) 針對金鑰對類型，選擇 RSA 或 ED25519。

(Windows 執行個體) 針對金鑰對類型，選擇 RSA。Windows 執行個體不支援 ED25519 金鑰。

6. 針對 Private key file format (私有金鑰檔案格式)，選擇私有金鑰的儲存格式。若要將私有金鑰儲存為可與 OpenSSH 搭配使用的格式，請選擇 pem。若要將私有金鑰儲存為可與 PuTTY 搭配使用的格式，請選擇 ppk。
7. 若要向公有金鑰新增標籤，請選擇 Add tag (新增標籤)，然後輸入標籤的金鑰和值。針對每個標籤，重複執行上述動作。
8. 選擇 Create key pair (建立金鑰對)。
9. 您的瀏覽器會自動下載私有金鑰檔案。基礎檔案名稱是您所指定的金鑰對名稱，而副檔名是根據您選擇的檔案格式所決定。將私有金鑰檔案存放在安全的地方。

> **Important**
>
> 這是您儲存私有金鑰檔案的唯一機會。

10. 如果您計劃在 macOS 或 Linux 電腦上使用 SSH 用戶端連線至 Linux 執行個體，請使用下列命令設定私有金鑰檔案的許可，以便只有您能夠讀取該檔案。

```
chmod 400 key-pair-name.pem
```

如果您未設定這些許可，則無法使用此金鑰對來連接至您的執行個體。如需詳細資訊，請參閱錯誤：未受保護的私有金鑰檔案。

AWS CLI

使用 Amazon EC2 建立金鑰對

1. 按如下說明，使用 create-key-pair 命令，產生金鑰對並將私有金鑰儲存到 .pem 檔案中。使用 --query 選項可將私有金鑰材料列印到輸出。使用 --output 選項可將私有金鑰資料儲存於特定檔案。視乎金鑰格式，副檔名應為 .pem 或 .ppk。私有金鑰名稱可與公有金鑰名稱不同，但為了方便使用，請使用相同的名稱。

```
aws ec2 create-key-pair \
    --key-name my-key-pair \
    --key-type rsa \
    --key-format pem \
    --query "KeyMaterial" \
```

```
    --output text > my-key-pair.pem
```

2. 如果您計劃在 macOS 或 Linux 電腦上使用 SSH 用戶端連線至 Linux 執行個體，請使用下列命令設定私有金鑰檔案的許可，以便只有您能夠讀取該檔案。

```
chmod 400 key-pair-name.pem
```

如果您未設定這些許可，則無法使用此金鑰對來連接至您的執行個體。如需詳細資訊，請參閱錯誤：未受保護的私有金鑰檔案。

PowerShell

使用 Amazon EC2 建立金鑰對

按如下說明，使用 New-EC2KeyPair cmdlet，產生金鑰並將其儲存到 .pem 或 .ppk 檔案中。使用 Out-File cmdlet 可將私有金鑰資料儲存在有特定副檔名的檔案中。視乎金鑰格式，副檔名應為 .pem 或 .ppk。私有金鑰名稱可與公有金鑰名稱不同，但為了方便使用，請使用相同的名稱。

```
(New-EC2KeyPair `
    -KeyName "my-key-pair" `
    -KeyType "rsa" `
    -KeyFormat "pem").KeyMaterial | Out-File -Encoding ascii -FilePath C:\path\my-key-pair.pem
```

使用 建立金鑰對 AWS CloudFormation

當您使用 建立新的金鑰對時 CloudFormation，私有金鑰會儲存至 AWS Systems Manager 參數存放區。參數名稱具有以下格式：

```
/ec2/keypair/key_pair_id
```

若要了解詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 AWS Systems Manager 參數存放區。

使用 建立金鑰對 CloudFormation

1. 在範本中指定 AWS::EC2::KeyPair 資源。

```
Resources:
```

```
  NewKeyPair:
    Type: 'AWS::EC2::KeyPair'
    Properties:
      KeyName: new-key-pair
```

2. 按如下說明，使用 describe-key-pairs 命令以取得金鑰對 ID。

```
aws ec2 describe-key-pairs --filters Name=key-name,Values=new-key-pair --query
 KeyPairs[*].KeyPairId --output text
```

以下為範例輸出。

```
key-05abb699beEXAMPLE
```

3. 按如下說明，使用 get-parameter 命令以取得金鑰參數，並將金鑰材料儲存在 .pem 檔案中。

```
aws ssm get-parameter --name /ec2/keypair/key-05abb699beEXAMPLE --with-decryption
 --query Parameter.Value --output text > new-key-pair.pem
```

所需的 IAM 許可

若要讓 代表您 CloudFormation 管理參數存放區參數， CloudFormation 或您的使用者擔任的 IAM 角色必須具有下列許可：

- `ssm:PutParameter` – 授予許可，以建立私有金鑰材料的參數。
- `ssm:DeleteParameter` - 准許刪除用來存放私有金鑰材料的參數。無論金鑰對是否是由 CloudFormation匯入或建立，都需要此許可。

當 CloudFormation 刪除由堆疊建立或匯入的金鑰對時，它會執行許可檢查，以判斷您是否具有刪除參數的許可，即使它僅在建立金鑰對時 CloudFormation 建立參數，而不是在匯入金鑰對時建立參數。 會使用與您帳戶中任何參數不相符的建構參數名稱來 CloudFormation 測試所需的許可。因此，您可能會在 `AccessDeniedException` 錯誤訊息中看到虛構參數名稱。

使用第三方工具來建立金鑰對，然後將公有金鑰匯入 Amazon EC2

您可以不使用 Amazon EC2 來建立金鑰對，而改用第三方工具來建立 RSA 或 ED25519 金鑰對，然後再將公有金鑰匯入 Amazon EC2。

金鑰對的需求

- 支援的類型：
  - (Linux 與 Windows) RSA
  - (僅限 Linux) ED25519

    > **Note**
    >
    > Windows 執行個體不支援 ED25519 金鑰。

  - Amazon EC2 不接受 DSA 金鑰。
- 支援的格式：
  - OpenSSH 公有金鑰格式 (若為 Linux，`~/.ssh/authorized_keys` 中的格式)。
  - (僅限 Linux) 如果您在使用 EC2 Instance Connect API 時以 SSH 連線，則還支援 SSH2 格式。
  - SSH 私有金鑰檔案格式必須是 PEM 或 PPK
  - (僅 RSA) Base64 編碼的 DER 格式
  - (僅 RSA) 如 RFC 4716 中所指定的 SSH 公有金鑰檔案格式
- 支援的長度：
  - 1024、2048 和 4096。
  - (僅限 Linux) 如果您在使用 EC2 Instance Connect API 時以 SSH 連線，則支援的長度為 2048 和 4096。

使用第三方工具來產生金鑰對

1. 使用您自選的第三方工具來產生金鑰對 例如：您可以使用 ssh-keygen (標準 OpenSSH 安裝隨附的工具)。或者，Java、Ruby、Python 和其他多種程式設計語言提供了標準程式庫，您可以使用這些程式庫來產生金鑰對。

   > **Important**
   >
   > 私有金鑰必須是 PEM 或 PPK 格式。例如：使用 `ssh-keygen -m PEM` 來產生 PEM 格式的 OpenSSH 金鑰。

2. 將公有金鑰儲存到本機檔案。例如，`~/.ssh/my-key-pair.pub` (Linux、macOS) 或 `C:\keys\my-key-pair.pub` (Windows)。此檔案的副檔名不重要。

3. 將私有金鑰儲存到具有 .pem 或 .ppk 副檔名的本機檔案。例如，~/.ssh/my-key-pair.pem 或 ~/.ssh/my-key-pair.ppk (Linux、macOS) 或者 C:\keys\my-key-pair.pem 或 C:\keys\my-key-pair.ppk(Windows)。副檔名很重要，因為根據您用來連線至執行個體的工具而定，您將需要特定的檔案格式。OpenSSH 會需要 .pem 檔案，而 PuTTY 則需要 .ppk 檔案。

> **Important**
>
> 將私有金鑰檔案存放在安全的地方。您在啟動執行個體時需要提供公有金鑰的名稱，且每次連線至執行個體時也需要提供對應的私有金鑰。

建立金鑰對之後，請使用以下其中一個方法，將您的公有金鑰匯入 Amazon EC2。

Console

將公有金鑰匯入至 Amazon EC2

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Key Pairs (金鑰對)。
3. 選擇 Import key pair (匯入金鑰對)。
4. 在 Name (名稱) 處，輸入公有金鑰的描述性名稱。名稱最多可包含 255 個 ASCII 字元。它不能包括前置或結尾空格。

   > **Note**
   >
   > 當您從 EC2 主控台連線到執行個體時，主控台會為您的私有金鑰檔案名稱建議此名稱。

5. 選取 Browse (瀏覽) 以導覽並選取您的公有金鑰，或將公有金鑰的內容貼到 Public key contents (公有金鑰內容) 欄位。
6. 選擇 Import key pair (匯入金鑰對)。
7. 確認您匯入的公開金鑰出現在金鑰對清單中。

AWS CLI

將公有金鑰匯入至 Amazon EC2

使用 import-key-pair 命令。

```
aws ec2 import-key-pair \
    --key-name my-key-pair \
    --public-key-material fileb://path/my-key-pair.pub
```

確認是否已成功匯入金鑰對

使用 describe-key-pairs 命令。

```
aws ec2 describe-key-pairs --key-names my-key-pair
```

PowerShell

將公有金鑰匯入至 Amazon EC2

使用 Import-EC2KeyPair cmdlet。

```
$publickey=[Io.File]::ReadAllText("C:\Users\TestUser\.ssh\id_rsa.pub")
Import-EC2KeyPair `
    -KeyName my-key-pair `
    -PublicKey $publickey
```

確認是否已成功匯入金鑰對

使用 Get-EC2KeyPair cmdlet。

```
Get-EC2KeyPair -KeyName my-key-pair
```

描述金鑰對

您可以描述存放在 Amazon EC2 中的金鑰對。您也可以擷取公有金鑰材料，並辨識啟動時指定的公有金鑰。

任務

- 描述金鑰對
- 擷取公有金鑰資料
- 識別啟動時指定的公有金鑰

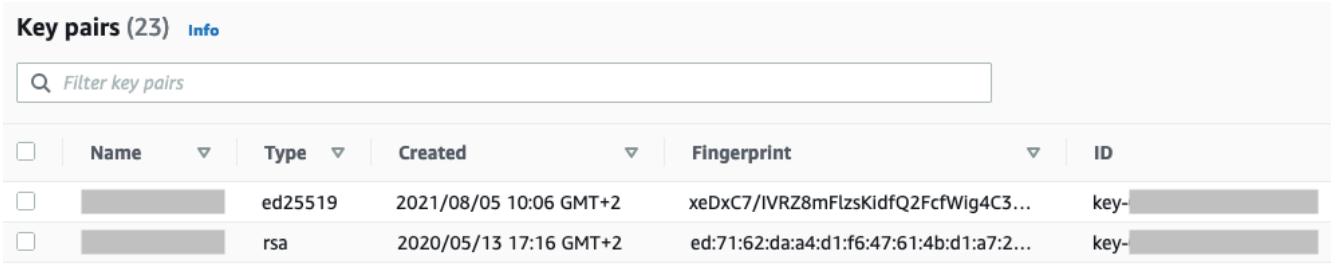
描述金鑰對

您可以查看存放於 Amazon EC2 中公有金鑰的以下資訊：公有金鑰名稱、ID、金鑰類型、指紋、公有金鑰資料、Amazon EC2 建立金鑰的日期和時間 (UTC 時區) (若金鑰是由第三方工具所建立，則為該金鑰匯入 Amazon EC2 的日期及時間)，以及任何與公有金鑰相關聯的標籤。

Console

如需檢視您的金鑰對的相關資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽器中，選擇 Key Pairs (金鑰對)。
3. 您可以在金鑰對表中檢視每個公有金鑰的相關資訊。



4. 若要檢視公有金鑰標籤，請勾選金鑰旁的核取方塊，然後依序選取動作、管理標籤。

AWS CLI

如需檢視金鑰對的相關資訊

使用 describe-key-pairs 命令。

```
aws ec2 describe-key-pairs --key-names key-pair-name
```

PowerShell

如需檢視金鑰對的相關資訊

使用 Get-EC2KeyPair cmdlet。

```
Get-EC2KeyPair -KeyName key-pair-name
```

擷取公有金鑰資料

您可獲取金鑰對的公有金鑰資料。下方所示為範例公有金鑰。請注意，新增了換行符號以方便閱讀。

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Private key

如需使用 ssh-keygen (Linux) 來擷取公有金鑰資料

請在本機 Linux 或 macOS 電腦使用 ssh-keygen 命令。指定已將私有金鑰 (.pem 檔案) 下載至其中的路徑。

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

若此 ssh-keygen 命令失敗，執行下面的 chmod 命令，確認私有金鑰檔案擁有必要的許可。

```
chmod 400 key-pair-name.pem
```

如需使用 PuTTYgen 來擷取公有金鑰資料 (Windows)

請在本機 Windows 電腦啟動 PuTTYgen。選擇載入。選取 .ppk 或 .pem 私有金鑰檔案。PuTTYgen 會在 Public key for pasting into OpenSSH authorized_keys file (貼到 OpenSSH authorized_keys 檔案的公有金鑰) 下顯示公有金鑰。您也可以選擇 Save public key (儲存公有金鑰)、指定檔案名稱、儲存檔案，然後開啟檔案，以檢視公有金鑰。

AWS CLI

如需擷取公有金鑰資料

請使用下面的 describe-key-pairs 命令並指定 --include-public-key 選項。

```
aws ec2 describe-key-pairs \
    --key-names key-pair-name \
    --include-public-key \
    --query "KeyPairs[].PublicKey"
```

PowerShell

如需擷取公有金鑰資料

請使用 Get-EC2KeyPair 命令。

```
(Get-EC2KeyPair -KeyName key-pair-name -IncludePublicKey $true).PublicKey
```

IMDSv2

Linux

透過執行個體執行下面的命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
```

Windows

透過 Windows 執行個體執行下面的 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令。

```
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

透過 Windows 執行個體中執行下列 cmdlet。

```
Invoke-RestMethod -uri  http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

識別啟動時指定的公有金鑰

如果您在啟動執行個體時指定了公有金鑰，執行個體會記錄公有金鑰名稱。即使您變更執行個體上的公有金鑰，或新增公有金鑰，執行個體的公有金鑰名稱也不會變更。

Console

如需確定啟動時指定的公有金鑰

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 請前往詳細資訊索引標籤，在執行個體詳細資訊下尋找啟動時指派的金鑰對。

AWS CLI

如需確定啟動時指定的公有金鑰

使用下面的 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].KeyName" \
    --output text
```

以下為範例輸出。

```
key-pair-name
```

PowerShell

如需確定啟動時指定的公有金鑰

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances | Select KeyName
```

以下為範例輸出。

```
KeyName
-------
key-pair-name
```

刪除您的金鑰對

您可以刪除金鑰對，這會移除存放在 Amazon EC2 中的公有金鑰。刪除金鑰對不會刪除相符的私有金鑰。

使用下列方法刪除公有金鑰時，只會刪除您在建立或匯入金鑰對時儲存在 Amazon EC2 中的公有金鑰。無論在啟動執行個體時或啟動之後，移除公有金鑰都不會從您新增金鑰的任何執行個體中移除公有金鑰。它也不會刪除您本機電腦上的私有金鑰。只要您仍然擁有私有金鑰 (`.pem`) 檔案，即可繼續使用從 Amazon EC2 刪除的公有金鑰來連線至已啟動的執行個體。

> **Important**
>
> 如果使用 Auto Scaling 群組 (例如，在 Elastic Beanstalk 環境中)，請確保在相關聯的啟動範本或啟動組態中並未指定您要刪除的公有金鑰。如果 Amazon EC2 Auto Scaling 偵測到運作狀態不良的執行個體，則會啟動替代執行個體。不過，如果找不到公有金鑰，執行個體啟動就會失敗。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 使用者指南中的啟動範本。

Console

刪除 Amazon EC2 上的公有金鑰

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Key Pairs (金鑰對)。
3. 選取要刪除的金鑰對，然後選取 Actions (動作)、Delete (刪除)。
4. 在確認欄位中，輸入 `Delete`，然後選擇 Delete (刪除)。

AWS CLI

刪除 Amazon EC2 上的公有金鑰

使用 delete-key-pair 命令。

```
aws ec2 delete-key-pair --key-name my-key-pair
```

PowerShell

刪除 Amazon EC2 上的公有金鑰

使用 Remove-EC2KeyPair cmdlet。

```
Remove-EC2KeyPair -KeyName my-key-pair
```

在執行個體上新增或取代公有金鑰

說明

如果您遺失私有金鑰，便無法存取任何使用金鑰對的執行個體。如需詳細資訊，了解使用與啟動時指定之金鑰對不同的金鑰對來連線至執行個體，請參閱我遺失了私有金鑰。

當您啟動執行個體時，您可指定金鑰對。若您在啟動時指定金鑰對，當執行個體第一次啟動時，公有金鑰材料會放置在 ~/.ssh/authorized_keys 內項目的 Linux 執行個體上。首次使用 SSH 連線至 Linux 執行個體時，您將指定與存放於 Linux 執行個體的公有金鑰相對應的預設使用者與私有金鑰。

在連線至執行個體之後，您可變更用來存取執行個體預設系統帳戶的金鑰對，方法是在執行個體上新增公有金鑰，或是取代執行個體上的公有金鑰 (刪除現有公有金鑰並在執行個體上新增一個)。您也可以從執行個體中移除所有公有金鑰。

以下是您可以新增或取代金鑰對的理由：

- 如果組織中的使用者需要使用不同的金鑰對來存取系統使用者，您可以將該共有金鑰新增到執行個體。
- 如果某些人擁有私有金鑰的複本 (.pem 檔案)，而您希望阻止他們連線到您的執行個體 (例如，如果他們已離開您的組織)，則您可刪除執行個體上的公有金鑰，並以新的金鑰取代。
- 如果您從執行個體中建立 Linux AMI，則公有金鑰材料會從執行個體複製到 AMI。如果從 AMI 中啟動執行個體，則新的執行個體會包含來自原始執行個體的公有金鑰。若要防止擁有私有金鑰的某些人連線到新執行個體，請從原始執行個體中移除公有金鑰，然後再建立 AMI。

使用以下程序來修改預設使用者 (例如 ec2-user) 的金鑰對。如需有關將使用者新增至執行個體的詳細資訊，請參閱執行個體上作業系統相關文件。

新增或替換金鑰對

1. 使用 Amazon EC2 主控台或第三方工具來建立新的金鑰對。
2. 從新的金鑰對擷取公有金鑰。如需詳細資訊，請參閱擷取公有金鑰資料。
3. 連線到您的執行個體。
4. 在執行個體上，使用您自選的文字編輯器開啟 .ssh/authorized_keys 檔案。貼上新金鑰對的公有金鑰資訊 (在現有公有金鑰資訊的下方)，然後儲存該檔案。
5. 中斷連線您的執行個體。測試您可使用新金鑰對的私有金鑰檔案，來連線至您的執行個體。
6. 若您將使用 Auto Scaling、EC2 機群，或者啟動範本來啟動執行個體，檢查在您的啟動範本或啟動組態中是否已指定您將要取代的金鑰對。否則，執行個體將無法啟動。
7. (選用) 如果要替換現有的金鑰對，請連線到執行個體，然後從 .ssh/authorized_keys 檔案，刪除原始金鑰對的公有金鑰資訊。

若要從執行個體中移除公有金鑰

1. 連線到您的執行個體。
2. 使用您自選的文字編輯器，在執行個體上開啟 .ssh/authorized_keys 檔案。刪除公有金鑰資訊，然後儲存檔案。

> **Warning**
>
> 若從執行個體中移除所有公有金鑰，並從執行個體中斷連線，除非您已設定另一種登入方式，否則無法再次連線執行個體。

驗證金鑰對的指紋

若要驗證金鑰對的指紋，請將 Amazon EC2 主控台金鑰對頁面上顯示的指紋，或由 describe-key-pairs 命令傳回的指紋，與您使用本機電腦上的私密金鑰產生的指紋進行比較。這些指紋應該相符。

當 Amazon EC2 計算指紋時，Amazon EC2 可能會使用 = 字元向指紋附加填補。其他工具，例如 ssh-keygen，可能會忽略此填補。

如果您要嘗試驗證 Linux EC2 執行個體的指紋，而不是金鑰對的指紋，請參閱取得執行個體指紋。

如何計算指紋

Amazon EC2 使用不同的雜湊函數來計算 RSA 和 ED25519 金鑰對的指紋。此外，對於 RSA 金鑰對，Amazon EC2 使用不同的雜湊函數來計算指紋，具體取決於金鑰對是由 Amazon EC2 建立或是匯入至 Amazon EC2。

下表列出了針對由 Amazon EC2 建立和匯入至 Amazon EC2 的 RSA 和 ED25519 金鑰對，用來計算其指紋的雜湊函數。

(Linux 執行個體) 用於計算指紋的雜湊函數

金鑰對來源	RSA 金鑰對 (Windows 和 Linux)	ED25519 金鑰對 (Linux)
由 Amazon EC2 建立	SHA-1	SHA-256
匯入至 Amazon EC2	MD5[1]	SHA-256

[1] 如果您將公有 RSA 金鑰匯入至 Amazon EC2，則會使用 MD5 雜湊函數來計算指紋。無論您如何建立金鑰對 (例如透過使用第三方工具，或從使用 Amazon EC2 建立的現有私有金鑰產生新的公有金鑰)，都是如此。

在不同區域中使用相同的金鑰對

如果您計劃使用相同的金鑰對來連接到不同 中的執行個體 AWS 區域，則必須將公有金鑰匯入您將使用該金鑰的所有 區域。如果您使用 Amazon EC2 來建立金鑰對，則可以 擷取公有金鑰資料，以便將公有金鑰匯入至其他區域。

> **Note**
>
> - 如果您使用 Amazon EC2 建立 RSA 金鑰對，然後使用 Amazon EC2 私有金鑰產生公有金鑰，則匯入的公有金鑰將具有與原始公有金鑰不同的指紋。這是因為使用 Amazon EC2 建立的原始 RSA 金鑰的指紋是使用 SHA-1 雜湊函數計算，而匯入的 RSA 金鑰的指紋則使用 MD5 雜湊函數計算。
> - 對於 ED25519 金鑰對，因為使用相同的 SHA-256 雜湊函數來計算指紋，所以無論是由 Amazon EC2 建立或是匯入至 Amazon EC2，指紋均相同。

從私有金鑰產生指紋

使用下列命令之一從本機電腦上的私有金鑰產生指紋。

如果您使用的是 Windows 本機電腦，您可以使用 Windows Subsystem for Linux (WSL) 來執行下列命令。使用如何透過 WSL 在 Windows 上安裝 Linux 中的指示，來安裝 WSL 和 Linux 發行版本。指示中的範例是安裝 Linux 的 Ubuntu 發行版本，但您可以安裝任何發行版本。您將會收到提示，要求您重新啟動電腦以讓變更生效。

- 如果是使用 Amazon EC2 建立金鑰對

  如下列範例所示，使用 OpenSSL 工具來產生指紋。

  對於 RSA 金鑰對：

  ```
  openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt |
   openssl sha1 -c
  ```

  (Linux 執行個體) 針對 ED25519 金鑰對：

  ```
  ssh-keygen -l -f path_to_private_key
  ```

- (僅限 RSA 金鑰對) 如果您已將公有金鑰匯入 Amazon EC2

  無論您如何建立金鑰對 (例如透過使用第三方工具，或從使用 Amazon EC2 建立的現有私有金鑰產生新的公有金鑰)，您都可以遵循此程序

  如下列範例所示，使用 OpenSSL 工具來產生指紋。

  ```
  openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
  ```

- 如果您已使用 OpenSSH 7.8 或更新版本建立 OpenSSH 金鑰對，並將公有金鑰匯入至 Amazon EC2

  如下列範例所示，使用 ssh-keygen 來產生指紋。

  對於 RSA 金鑰對：

  ```
  ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER
   | openssl md5 -c
  ```

  (Linux 執行個體) 針對 ED25519 金鑰對：

```
ssh-keygen -l -f path_to_private_key
```

EC2 執行個體的 Amazon EC2 安全群組

安全群組會做為您 EC2 執行個體的虛擬防火牆，控制傳入及傳出流量。傳入規則會控制傳入至您的執行個體的流量，以及傳出規則會控制從您的執行個體傳出的流量。啟動執行個體時，您會為其指派一個或多個安全群組。如果您未指定安全群組，Amazon EC2 便會使用 VPC 的預設安全群組。在您啟動執行個體之後，您可變更其安全群組。

安全性是 AWS 與您之間的共同責任。如需詳細資訊，請參閱 Amazon EC2 中的安全。 AWS 提供安全群組做為保護執行個體的工具之一，而且您需要設定這些群組以符合您的安全需求。若您有安全群組無法完全滿足的需求，您可以在使用安全群組的同時在您的任何執行個體上維持您自己的防火牆。

定價

使用安全群組無需額外收費。

目錄

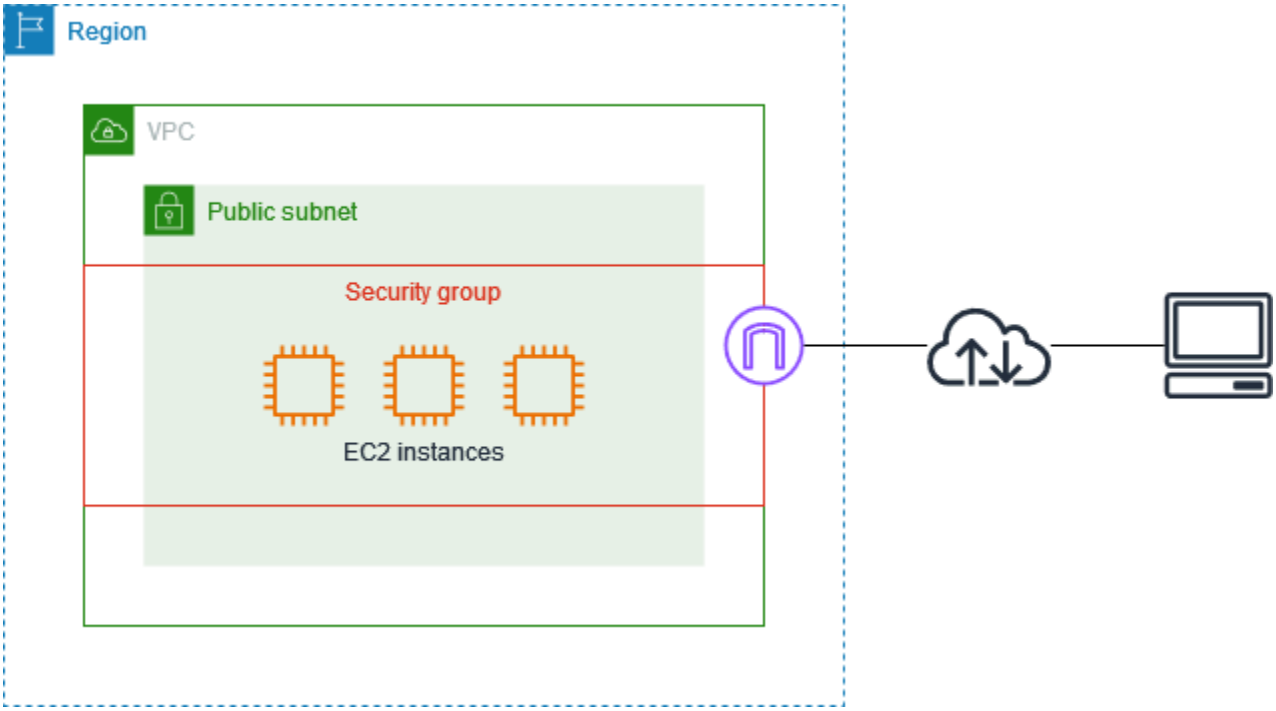
- 概要
- 為您的 Amazon EC2 執行個體建立安全群組。
- 變更 Amazon EC2 執行個體的安全群組。
- 刪除一個 Amazon EC2 安全群組
- Amazon EC2 安全群組連線追蹤
- 不同使用案例的安全群組規則

概要

您可以將每個執行個體與多個安全群組建立關聯，也可以將每個安全群組與多個執行個體建立關聯。您在各個安全群組新增規則，允許流量往返於建立關聯的執行個體。您可以隨時修改安全群組的規則。新的、修改過的規則會自動套用至與安全群組建立關聯的所有執行個體。當 Amazon EC2 決定是否要允許流量觸達執行個體時，我們會評估所有與執行個體建立關聯之安全群組的所有規則。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的安全群組規則。

下圖顯示具有子網路、網際網路閘道和安全群組的 VPC。子網路包含 EC2 執行個體。要與執行個體建立關聯的安全群組。僅有安全群組規則允許的流量才能到達執行個體。例如，如果安全群組包含允許

ICMP 流量從您的網路傳輸至執行個體的規則，您即可從電腦 SSH 執行個體。如果安全群組包含的規則允許來自與其相關聯的資源的所有流量，則每個執行個體都可以接收從其他執行個體傳送的任何流量。



安全群組是具有狀態的群組—若您從執行個體傳送請求，該請求的回應流量將允許流入，與對內安全群組規則無關。無論傳出規則為何，針對允許傳入流量的回應都會允許傳出。如需詳細資訊，請參閱連線追蹤。

為您的 Amazon EC2 執行個體建立安全群組。

安全群組就像是防火牆，用於關聯的執行個體，可在執行個體層級控制傳入及傳出流量。您可以將規則新增至安全群組，讓您能夠使用 SSH (Linux 執行個體) 或 RDP (Windows 執行個體) 連線至執行個體。您也可以新增允許用戶端流量的規則，例如，目的地為 Web 伺服器的 HTTP 和 HTTPS 流量。

您可以在啟動執行個體時，為執行個體指派安全群組。當您新增或移除規則時，這些變更會自動套用到所有您已指派該安全群組的執行個體。

在您啟動執行個體之後，您便無法變更指派給它的安全群組。如需詳細資訊，請參閱變更 Amazon EC2 執行個體的安全群組。。

您可以在建立安全群組時新增傳入和傳出安全群組規則，也可以稍後再新增這些規則。如需詳細資訊，請參閱設定安全群組規則。如需可新增至安全群組的規則範例，請參閱不同使用案例的安全群組規則。

考量事項

- 新的安全群組一開始只有允許流量離開資源的傳出規則。您必須新增規則啟用任何傳入流量，或是限制傳出流量。
- 為允許 SSH 或 RDP 存取執行個體的規則設定來源時，請勿允許從任何地方存取，因為它會允許從網際網路上的所有 IP 位址存取執行個體。通常在測試環境中短暫進行此操作是沒有問題的，但用在生產環境則不安全。
- 若特定連接埠有超過一個規則，Amazon EC2 會套用最寬鬆的規則。例如，若您有一個規則，允許 IP 位址 203.0.113.1 存取 TCP 連接埠 22 (SSH)；另一個規則允許所有人存取 TCP 連接埠 22，則所有人皆能存取 TCP 連接埠 22。
- 您可以將多個安全性群組指派給執行個體。因此，執行個體可以有數百個適用的規則。這可能會在您存取執行個體時產生問題。但建議您盡可能緊縮您的規則。
- 當您將安全群組指定為規則的來源或目標時，規則會影響所有與安全群組相關聯的執行個體。傳入流量會根據與來源安全群組相關聯之執行個體的私有 IP 地址允許 (而非公有 IP 或彈性 IP 地址)。如需有關 IP 地址的詳細資訊，請參閱 Amazon EC2 執行個體 IP 定址。
- 請注意，根據預設 Amazon EC2 會封鎖連接埠 25 上的流量。如需詳細資訊，請參閱使用通訊埠 25 傳送的電子郵件限制。

Console

建立安全群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Security Groups (安全群組)。
3. 選擇 Create Security Group (建立安全群組)。
4. 輸入安全性群組的描述性名稱和簡短描述。您無法在建立安全群組之後變更安全群組的名稱和說明。
5. 針對 VPC，選擇您要在其中執行 Amazon EC2 執行個體的 VPC。
6. (選用) 若要新增傳入規則，請選擇傳入規則。針對每個規則，選擇新增規則並指定通訊協定、連接埠和來源。例如，若要允許 SSH 流量，請選擇 SSH for Type，並為來源指定電腦或網路的公有 IPv4 地址。
7. (選用) 若要新增傳出規則，請選擇傳出規則。針對每個規則，選擇新增規則並指定通訊協定、連接埠和目的地。在 Outbound (傳出) 索引標籤上，保留允許所有傳出流量的預設規則。
8. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤金鑰和值。

9. 選擇建立安全群組。

AWS CLI

建立安全群組

使用下面的 create-security-group 命令。

```
aws ec2 create-security-group \
    --group-name my-security-group \
    --description "my security group" \
    --vpc-id vpc-1234567890abcdef0
```

若要了解新增規則的範例，請參閱 the section called “設定安全群組規則”。

PowerShell

建立安全群組

使用 New-EC2SecurityGroup cmdlet。

```
New-EC2SecurityGroup `
    -GroupName my-security-group `
    -Description "my security group" `
    -VpcId vpc-1234567890abcdef0
```

若要了解新增規則的範例，請參閱 the section called “設定安全群組規則”。

變更 Amazon EC2 執行個體的安全群組。

您可以在啟動 Amazon EC2 執行個體時為其指定安全群組。在您啟動執行個體之後，您可變更其安全群組。您也可以隨時新增、移除或編輯關聯安全群組的安全群組規則。

安全群組與網路介面關聯。新增或移除安全群組會變更與主要網路介面相關聯的安全群組。您也可以變更與任何其他網路介面關聯的安全群組。如需詳細資訊，請參閱修改網路介面屬性。

任務

- 新增或移除 VPC 安全群組。
- 設定安全群組規則

新增或移除 VPC 安全群組。

啟動執行個體後，您可以從相關聯的安全群組清單中新增或移除安全群組。當您將多個安全群組與執行個體建立關聯時，每個安全群組的規則都會有效彙總並建立一組規則。Amazon EC2 會使用這一組規則判斷是否要允許存取。

要求

- 執行個體必須處於 `running` 或 `stopped` 狀態。

Console

變更執行個體的安全群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體，然後選取 動作、安全性、變更安全群組。
4. 對於關聯的安全群組，請從清單中選取安全群組，然後選取新增安全群組。

   若要移除關聯的安全群組，請為該安全群組選擇移除。
5. 選擇儲存。

AWS CLI

變更執行個體的安全群組

請使用下面的 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

PowerShell

變更執行個體的安全群組

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -Group sg-1234567890abcdef0
```

設定安全群組規則

建立安全群組之後，您可以新增、更新和刪除其安全群組規則。當您更新規則時，更新的規則會自動套用至與安全群組相關聯的任何資源。

如需可新增至安全群組的規則範例，請參閱不同使用案例的安全群組規則。

所需的許可

開始之前，請務必備妥必要的許可。如需詳細資訊，請參閱範例：使用安全群組。

通訊協定與連接埠

- 在主控台中選取預先定義的類型後，系統會為您指定通訊協定與連接埠範圍。如需輸入連接埠範圍，必須選取以下自訂類型之一：自訂 TCP 或自訂 UDP。
- 使用 AWS CLI，您可以使用 `--protocol`和 `--port`選項新增具有單一連接埠的單一規則。如需新增多項規則，或具有連接埠範圍的規則，請改用 `--ip-permissions` 選項。

資料來源和目的地

- 透過主控台，可指定下列內容作為傳入規則的來源，或傳出規則的目的地：
  - 自訂 – IPv4 CIDR 區塊、IPv6 CIDR 區塊、安全群組或字首清單。
  - Anywhere-IPv4 – 0.0.0.0/0 IPv4 CIDR 區塊。
  - Anywhere-IPv6 – ::/0 IPv6 CIDR 區塊。
  - 我的 IP：自動新增您本機電腦的公有 IPv4 地址。
- 透過 AWS CLI，您可以使用 `--cidr`選項指定 IPv4 CIDR 區塊，或使用 `--source-group`選項指定安全群組。如需指定字首清單或 IPv6 CIDR 區塊，請使用 `--ip-permissions` 選項。

> **Warning**
>
> 如果您要為連接埠 22 (SSH) 或 3389 (RDP) 新增規則，則應只授權特定 IP 位址或特定範圍的地址來存取您的執行個體。如果選擇 Anywhere-IPv4，則可讓所有 IPv4 地址使用指定的通訊協定存取您的執行個體。如果選擇 Anywhere-IPv6，則可讓所有 IPv6 地址使用指定的通訊協定存取您的執行個體。

Console

設定安全群組規則

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇安全群組。
3. 選取安全群組。
4. 若要編輯傳入規則，請從動作或傳入規則索引標籤中選擇編輯傳入規則。
   a. 若要新增規則，請選擇新增規則，並輸入規則的類型、通訊協定、連接埠和來源。

      針對 TCP 或 UDP，您必須輸入要允許的連接埠範圍。針對自訂 ICMP，則必須從 Protocol (通訊協定) 中選擇 ICMP 類型名稱，然後再從 Port Range (連接埠範圍) 中選擇代碼名稱 (若適用)。如果是任何其他類型，則系統會自動為您設定通訊協定和連接埠範圍。
   b. 若要更新規則，請視需要變更其通訊協定、描述和來源。不過，您無法變更來源類型。例如，如果來源是 IPv4 CIDR 區塊，則無法指定 IPv6 CIDR 區塊、字首清單或安全群組。
   c. 若要刪除規則，請選擇其刪除按鈕。
5. 若要編輯傳出規則，請從動作或傳出規則索引標籤中選擇編輯傳出規則。
   a. 若要新增規則，請選擇新增規則，並輸入規則的類型、通訊協定、連接埠和目的地。您也可以選擇輸入描述。

      針對 TCP 或 UDP，您必須輸入要允許的連接埠範圍。針對自訂 ICMP，則必須從 Protocol (通訊協定) 中選擇 ICMP 類型名稱，然後再從 Port Range (連接埠範圍) 中選擇代碼名稱 (若適用)。如果是任何其他類型，則系統會自動為您設定通訊協定和連接埠範圍。
   b. 若要更新規則，請視需要變更其通訊協定、描述和來源。不過，您無法變更來源類型。例如，如果來源是 IPv4 CIDR 區塊，則無法指定 IPv6 CIDR 區塊、字首清單或安全群組。
   c. 若要刪除規則，請選擇其刪除按鈕。
6. 選擇儲存規則。

AWS CLI

如需新增安全群組規則

請使用 authorize-security-group-ingress 命令，來新增傳入規則。在下面的範例中，允許從指定字首清單中的 CIDR 區塊傳入 SSH 流量。

```
aws ec2 authorize-security-group-ingress \
```

```
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds=[{PrefixListId=pl-
f8a6439156EXAMPLE}]'
```

請使用 authorize-security-group-egress 命令，來新增傳出規則。在下面的範例中，允許連接埠 80 上的 TCP 流量傳出至指定安全群組的執行個體。

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bb6EXAMPLE}]'
```

如需移除安全群組規則

請使用下面的 revoke-security-group-ingress 命令，來移除傳入規則。

```
aws ec2 revoke-security-group-egress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-09ed298024EXAMPLE
```

請使用下面的 revoke-security-group-egress 命令，來移除傳出規則。

```
aws ec2 revoke-security-group-ingress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-0352250c1aEXAMPLE
```

如需修改安全群組規則

請使用 modify-security-group-rules 命令。在下面的範例中，變更了指定安全群組規則的 IPv4 CIDR 區塊。

```
aws ec2 modify-security-group-rules \
    --group id sg-1234567890abcdef0 \
    --security-group-rules
 'SecurityGroupRuleId=sgr-09ed298024EXAMPLE,SecurityGroupRule={IpProtocol=tcp,FromPort=80,To
```

PowerShell

如需新增安全群組規則

請使用 Grant-EC2SecurityGroupIngress cmdlet，來新增傳入規則。在下面的範例中，允許從指定字首清單中的 CIDR 區塊傳入 SSH 流量。

```
$plid = New-Object -TypeName Amazon.EC2.Model.PrefixListId
$plid.Id = "pl-f8a6439156EXAMPLE"
Grant-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=22; ToPort=22; PrefixListIds=$plid}
```

請使用 Grant-EC2SecurityGroupEgress cmdlet，來新增傳出規則。在下面的範例中，允許連接埠 80 上的 TCP 流量傳出至指定安全群組的執行個體。

```
$uigp = New-Object -TypeName Amazon.EC2.Model.UserIdGroupPair
$uigp.GroupId = "sg-0aad1c26bb6EXAMPLE"
Grant-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=80; ToPort=80; UserIdGroupPairs=
$uigp}
```

如需移除安全群組規則

請使用 Revoke-EC2SecurityGroupIngress cmdlet，來移除傳入規則。

```
Revoke-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-09ed298024EXAMPLE
```

請使用 Revoke-EC2SecurityGroupEgress cmdlet，來移除傳出規則。

```
Revoke-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-0352250c1aEXAMPLE
```

如需修改安全群組規則

使用 Edit-EC2SecurityGroupRule cmdlet。在下面的範例中，變更了指定安全群組規則的 IPv4 CIDR 區塊。

```
$sgrr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleRequest
```

```
$sgrr.IpProtocol = "tcp"
$sgrr.FromPort = 80
$sgrr.ToPort = 80
$sgrr.CidrIpv4 = "0.0.0.0/0"
$sgr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleUpdate
$sgr.SecurityGroupRuleId = "sgr-09ed298024EXAMPLE"
$sgr.SecurityGroupRule = $sgrr
Edit-EC2SecurityGroupRule  `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRule $sgr
```

刪除一個 Amazon EC2 安全群組

當您完成建立用於 Amazon EC2 執行個體的安全群組時，您可以將其刪除。

要求

- 安全群組無法與執行個體或網路介面建立關聯。
- 由另一個安全群組規則參考的安全群組無法刪除。

Console

刪除安全群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. (選用) 若要確認您的安全群組未與執行個體建立關聯，請執行下列動作：
   a. 在導覽窗格中，選擇安全群組。
   b. 複製要刪除的安全群組 ID。
   c. 在導覽窗格中，選擇執行個體。
   d. 在搜尋列中，新增安全群組 IDs 等於篩選條件，並貼上安全群組的 ID。如果沒有結果，則安全群組不會與執行個體建立關聯。否則，您必須先取消安全群組的關聯，才能將其刪除。
3. 在導覽窗格中，選擇安全群組。
4. 選取安全群組，然後選擇動作、刪除安全群組。
5. 如果您選擇多個安全群組，系統會提示您進行確認。如果無法刪除某些安全群組，我們會顯示每個安全群組的狀態，指出是否要刪除。若要確認刪除，請輸入刪除。

6. 選擇 刪除。

AWS CLI

刪除安全群組

請使用下面的 delete-security-group 命令。

```
aws ec2 delete-security-group --group-id sg-1234567890abcdef0
```

PowerShell

刪除安全群組

使用 Remove-EC2SecurityGroup cmdlet。

```
Remove-EC2SecurityGroup -GroupId sg-1234567890abcdef0
```

Amazon EC2 安全群組連線追蹤

您的安全群組使用連線追蹤來追蹤流入和流出執行個體流量的資訊。規則會根據流量的連線狀態套用，以判斷該允許或拒絕該流量。使用這種方法，安全群組是可設定狀態的。這表示傳入流量的回應可從執行個體傳出，與對外安全群組規則無關，反之亦然。

舉例來說，假設您從家用電腦中對執行個體啟動一個命令 (例如 netcat 或類似的命令)，並且您的傳入安全群組規則允許 ICMP 流量。會追蹤連線的相關資訊 (包括連接埠資訊)。來自執行個體對命令的回應流量不會作為新的請求追蹤，而是作為已建立的連線，並且將允許流出執行個體，即使您的傳出安全群組規則限制傳出 ICMP 流量。

針對 TCP、UDP 或 ICMP 之外的通訊協定，只會追蹤 IP 地址和通訊協定號碼。若您的執行個體傳送流量到另一個主機，且該主機也在 600 秒內將相同類型的流量傳送至您的執行個體，無論傳入安全群組規則為何，執行個體的安全群組都會接受該流量。安全群組會接受該流量，因為將其視為原始流量的回應流量。

變更安全群組規則時，其跟蹤的連線不會立即中斷。安全群組將繼續允許封包，直到現有連線逾時為止。若要確保無論追蹤狀態為何，流量皆會立即中斷，或所有流量都受到防火牆規則的限制，您可以為子網使用網路 ACL。網路 ACL 為無狀態，因此不會自動允許回應流量。新增阻止任一方向流量的網路 ACL 會破壞現有連線。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的網路 ACL。

> **Note**
>
> 安全群組對 Route 53 Resolver 接收或發出的 DNS 流量沒有影響，有時稱為「VPC+2 IP 地址」(請參閱「Amazon Route 53 開發人員指南」中的 Amazon Route 53 Resolver 是什麼？)，或稱為 「AmazonProvidedDNS」 (請參閱「Amazon Virtual Private Cloud 使用者指南」中的使用 DHCP 選項集)。如果您想要透過 Route 53 Resolver 篩選 DNS 請求，則可以啟用 Route 53 Resolver DNS Firewall (請參閱「Amazon Route 53 開發人員指南」中的 Route 53 Resolver DNS Firewall)。

未追蹤的連線

並非所有的流量流程都會追蹤。若安全群組規則允許所有流量的 TCP 或 UDP 流 (0.0.0.0/0 或 ::/0)，且另一個方向也有對應的規則，它允許所有連接埠 (0-65535) 的所有回應流量 (0.0.0.0/0 或 ::/0)，則不會追蹤流量，除非其為自動追蹤連線的一部分。接著會依據允許回應流量的傳入或傳出規則允許未追蹤流的回應流量，而非依據追蹤資訊。

若啟用未受追蹤之流量流程的規則遭到移除或修改，則該流程會立即中斷。例如，如果您有一個開啟的 (0.0.0.0/0) 輸出規則，並移除允許所有 (0.0.0.0/0) 輸入 SSH (TCP 連接埠 22) 流量至執行個體的規則 (或進行修改以不再允許連線)，則您現有的 SSH 連線會立即中斷與執行個體的連線。該連線之前未被追蹤，因此該變更會中斷連線。另一方面，如果您的輸入規則範圍較窄，最初允許 SSH 連線 (表示連線已被追蹤)，但會將規則變更為不再允許來自目前 SSH 用戶端地址的新連線，則現有的 SSH 連線將不會因為其遭到追蹤而中斷。

自動追蹤連線

即使安全群組組態未以其他方式要求追蹤，也會自動追蹤透過以下方式所建立的連線。

- 輸出限定網際網路閘道
- Global Accelerator 加速器
- NAT 閘道
- Network Firewall 防火墻端點
- Network Load Balancer
- AWS PrivateLink （介面 VPC 端點）
- AWS Lambda （超平面彈性網路介面）
- DynamoDB 閘道端點 – 每個 DynamoDB 連線皆會取用 2 個 Conntrack 項目。

連線追蹤額度

Amazon EC2 會定義每個執行個體可追蹤的連線數目上限。達到最大值之後，任何傳送或接收的封包都會捨棄，因為無法建立新的連線。發生這種情況時，傳送及接收封包的應用程式無法正確通訊。使用 `conntrack_allowance_available` 網路效能指標來判斷該執行個體類型仍可使用的追蹤連線數目。

若要判斷是否因為執行個體的網路流量超過可追蹤的連線數目上限而捨棄封包，請使用 `conntrack_allowance_exceeded` 網路效能指標。如需詳細資訊，請參閱監控 EC2 執行個體的 ENA 設定網路效能。

使用 Elastic Load Balancing 時，如果超過每個執行個體可追蹤的連線數目上限，建議您使用負載平衡器調整執行個體數目，或使用負載平衡器調整執行個體大小。

連線追蹤最佳實務

非對稱路由，也就是當流量透過一個網路介面進入執行個體，並透過不同的網路介面離開，可能導致執行個體在追蹤流程時，可達的峰值效能降低。

若要在為您的安全群組啟用連線追蹤時維持尖峰效能並最佳化連線管理，我們建議您使用下列組態：

- 如果可能，請避免非對稱路由拓撲。
- 使用網路 ACLs 而不是使用安全群組進行篩選。
- 如果您必須使用安全群組進行連線追蹤，請設定最短的閒置連線追蹤逾時。如需閒置連線追蹤逾時的詳細資訊，請參閱下一節。
- 隨著 Nitrov6 執行個體的預設逾時較短，具有長期連線（例如資料庫連線集區、持久性 HTTP 連線或串流工作負載）的應用程式應在執行個體啟動時設定適當的`TcpEstablishedTimeout`值。
- 對於長期連線，請將 TCP 保持運作狀態設定為以少於 5 分鐘的間隔傳送，以確保連線保持開啟並維持其追蹤狀態。這有助於防止因閒置逾時而捨棄連線，並減少連線重新建立的額外負荷。

如需開啟績效詳情的詳細資訊，請參閱Nitro 系統效能調校的考量因素。

閒置連線追蹤逾時

安全群組會追蹤每個建立的連線，以確保傳回封包如預期交付。每個執行個體可追蹤的連線數目上限。處於閒置狀態的連線可能會導致連線追蹤耗盡，並造成不追蹤連線及捨棄封包。您可以在彈性網路介面上設定閒置連線追蹤的逾時。

> **Note**
>
> 此功能僅適用於 Nitro 型執行個體。在部署到生產環境之前，您應該在 Nitrov6 世代執行個體上測試應用程式，減少350第二個預設連線追蹤逾時。

可設定的逾時有三種：

- 已建立的 TCP 逾時：處於已建立狀態的閒置 TCP 連線逾時 (以秒為單位)。
  - 最小值：60 秒
  - 最大值：432000 秒
  - 預設：Nitro v6 執行個體類型的350秒數 (P6e-GB200 除外）；所有其他執行個體類型的432000秒數 （包括 P6e-GB200)。
  - 建議：432000 秒以下。
- UDP 逾時：只出現單一方向或單一請求回應交易流量的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：30 秒
  - 最大值：60 秒
  - 預設：30 秒
- UDP 串流逾時：分類為出現多個請求回應交易串流的閒置 UDP 流量逾時 (以秒為單位)。
  - 最小值：60 秒
  - 最大值：180 秒
  - 預設：180 秒

您可修改下列任何情況的預設逾時：

- 如果您使用 Amazon EC2 網路效能指標監控追蹤的連線，conntrack_allowance_exceeded 和 conntrack_allowance_available 可用指標可讓您監控丟棄的封包和追蹤連線使用率，以透過縱向擴展或橫向擴展動作來主動管理 EC2 執行個體容量，以協助在丟棄封包之前滿足網路連線需求。如果您在 EC2 執行個體上觀察到 conntrack_allowance_exceeded，將已建立的 TCP 逾時設定得較低可能對您有所幫助，可解決因不正確的用戶端或網路中間盒造成的過時 TCP/UDP 工作階段。
- 一般而言，負載平衡器或防火牆已建立的 TCP 閒置逾時範圍為 60 到 90 分鐘。如果您執行的工作負載預期會處理來自網路防火牆等設備的大量連線 (10 萬以上)，建議您在 EC2 網路介面上設定相似的逾時。
- 如果您執行的工作負載使用非對稱路由拓撲，建議您設定 TCP 建立的閒置逾時 60 秒。

- 如果您執行的工作負載具有大量連線 (例如 DNS、SIP、SNMP、Syslog、Radius 和其他主要使用 UDP 來處理請求的服務)，則將「UDP 串流」逾時設為 60 秒，可為現有容量提供更高的規模/效能，並防止發生灰色故障。
- 系統會追蹤所有經過 Network Load Balancer 的 TCP/UDP 連線。TCP 流量的閒置逾時值為 350 秒，UDP 流量為 120 秒，且會隨介面層級逾時值而異。您可在網路介面層級設定逾時，讓逾時的彈性大於負載平衡器的預設值。

執行下列操作時，您可以選擇設定連線追蹤逾時：

- 建立網路介面
- 修改網路介面屬性
- 啟動 EC2 執行個體
- 建立 EC2 執行個體啟動範本

範例

在下列範例中，安全群組具備允許 TCP 和 ICMP 流量的傳入規則，以及允許所有傳出流量的對外規則。

傳入

通訊協定類型	連接埠號碼	來源
TCP	22 (SSH)	203.0.113.1/32
TCP	80 (HTTP)	0.0.0.0/0
TCP	80 (HTTP)	::/0
ICMP	全部	0.0.0.0/0

傳出

通訊協定類型	連接埠號碼	目的地
全部	全部	0.0.0.0/0
全部	全部	::/0

對於執行個體或網路介面的直接網路連線，追蹤行為如下：

- 會追蹤連接埠 22 (SSH) 上傳入和傳出的 TCP 流量，因為傳入規則僅允許來自 203.0.113.1/32 的流量，而非所有 IP 地址 (0.0.0.0/0)。
- 連接埠 80 (HTTP) 上傳入和傳出的 TCP 流量則不會受到追蹤，因為傳入和傳出規則允許來自所有 IP 地址的流量。
- ICMP 流量一律會受到追蹤。

若您移除 IPv4 流量的傳出規則，則所有傳入和傳出的 IPv4 流量都會受到追蹤，包括位於連接埠 80 (HTTP) 上的流量。如果您移除 IPv6 流量的傳出規則，IPv6 流量也會受到相同影響。

不同使用案例的安全群組規則

您可以建立安全群組並新增可反映與該安全群組相關聯之執行個體角色的規則。例如，設定為 Web 伺服器的執行個體需要允許輸入 HTTP 和 HTTPS 存取的安全性群組規則。同樣，資料庫執行個體需要允許存取資料庫類型的規則，例如 MySQL 透過連接埠 3306 的存取。

以下為您可以針對特定類型存取，為安全群組新增之規則類型的範例。

範例

- Web 伺服器規則
- 資料庫伺服器規則
- 從您電腦連線到執行個體的規則
- 從執行個體使用相同安全群組連線到執行個體的規則
- Ping/ICMP 的規則
- DNS 伺服器規則
- Amazon EFS 規則
- Elastic Load Balancing 規則

如需說明，請參閱 建立安全群組 與 the section called “設定安全群組規則”。

Web 伺服器規則

以下對內規則會允許來自任何 IP 地址的 HTTP 和 HTTPS 存取。若您的 VPC 已啟用 IPv6，您可以新增規則，控制來自 IPv6 地址的對內 HTTP 和 HTTPS 流量。

通訊協定類型	通訊協定號碼	連接埠	來源 IP	備註
TCP	6	80 (HTTP)	0.0.0.0/0	允許來自任何 IPv4 地址的對內 HTTP 存取
TCP	6	443 (HTTPS)	0.0.0.0/0	允許來自任何 IPv4 地址的對內 HTTPS 存取
TCP	6	80 (HTTP)	::/0	允許來自任何 IPv6 地址的傳入 HTTP 存取
TCP	6	443 (HTTPS)	::/0	允許來自任何 IPv6 地址的傳入 HTTPS 存取

資料庫伺服器規則

以下對內規則是您可以為資料庫存取新增之規則的範例，取決於您在執行個體上執行的資料庫類型為何。如需 Amazon RDS 執行個體的詳細資訊，請參閱 Amazon RDS 使用者指南。

針對來源 IP，請指定下列其中一項：

- 您本機網路中的特定 IP 地址或 IP 地址範圍 (CIDR 區塊標記法)
- 存取資料庫之執行個體群組的安全群組 ID

通訊協定類型	通訊協定號碼	連接埠	備註
TCP	6	1433 (MS SQL)	在 Amazon RDS 執行個體上存取 Microsoft SQL Server 資料庫的預設連接埠 (範例)
TCP	6	3306 (MYSQL/Aurora)	在 Amazon RDS 執行個體上存取 MySQL 或 Aurora 資料庫的預設連接埠 (範例)
TCP	6	5439 (Redshift)	存取 Amazon Redshift 叢集資料庫的預設連接埠。

通訊協定類型	通訊協定號碼	連接埠	備註
TCP	6	5432 (PostgreSQL)	在 Amazon RDS 執行個體上存取 PostgreSQL 資料庫的預設連接埠 (範例)
TCP	6	1521 (Oracle)	在 Amazon RDS 執行個體上存取 Oracle 資料庫的預設連接埠 (範例)

您可以選擇性限制來自資料庫伺服器的輸出流量。例如，您可能想允許存取網際網路以取得軟體更新，但限制所有其他類型的流量。您必須先移除允許所有對外流量的預設對外規則。

通訊協定類型	通訊協定號碼	連接埠	目標 IP	備註
TCP	6	80 (HTTP)	0.0.0.0/0	允許前往任何 IPv4 地址的對外 HTTP 存取
TCP	6	443 (HTTPS)	0.0.0.0/0	允許前往任何 IPv4 地址的對外 HTTPS 存取
TCP	6	80 (HTTP)	::/0	(僅限啟用 IPv6 的 VPC) 允許前往任何 IPv6 地址的對外 HTTP 存取
TCP	6	443 (HTTPS)	::/0	(僅限啟用 IPv6 的 VPC) 允許前往任何 IPv6 地址的對外 HTTPS 存取

從您電腦連線到執行個體的規則

若要連線到您的執行個體，您的安全群組必須具有允許 SSH 存取 (適用於 Linux 執行個體) 或 RDP 存取 (適用於 Windows 執行個體) 的對內規則。

通訊協定類型	通訊協定號碼	連接埠	來源 IP
TCP	6	22 (SSH)	您電腦的公有 IPv4 地址，或是您本機網路中的 IP 地址範圍。若您的 VPC

通訊協定類型	通訊協定號碼	連接埠	來源 IP
			已啟用 IPv6，且您的執行個體具有 IPv6 地址，您可以輸入 IPv6 地址或範圍。
TCP	6	3389 (RDP)	您電腦的公有 IPv4 地址，或是您本機網路中的 IP 地址範圍。若您的 VPC 已啟用 IPv6，且您的執行個體具有 IPv6 地址，您可以輸入 IPv6 地址或範圍。

從執行個體使用相同安全群組連線到執行個體的規則

若要允許與相同安全群組相關聯的執行個體彼此之間互相通訊，您必須為此明確新增規則。

> **Note**
>
> 如果您將路由設定為透過中間設備來轉遞不同子網中兩個執行個體之間的流量，則您必須確保兩個執行個體的安全群組均允許流量在執行個體之間流動。每個執行個體的安全群組都必須參考另一個執行個體的私有 IP 地址，或是包含其他執行個體之子網的 CIDR 範圍作為來源。如果您參考另一個執行個體的安全群組作為來源，這不會允許流量在執行個體之間流動。

下表說明可讓相關聯的執行個體互相通訊的安全群組傳入規則。規則允許所有類型的流量。

通訊協定類型	通訊協定號碼	連接埠	來源 IP
-1 (全部)	-1 (全部)	-1 (全部)	安全群組的 ID，或是包含其他執行個體之子網的 CIDR 範圍 (請參閱備註)。

Ping/ICMP 的規則

ping 命令是一種 ICMP 流量。若要 ping 您的執行個體，您必須新增下列任一對內 ICMP 規則。

Type	通訊協定	來源		
自訂 ICMP - IPv4	回聲請求	您電腦的公有 IPv4 地址、特定 IPv4 地址，或任何位置的 IPv4 或 IPv6 地址。		
所有 ICMP - IPv4	IPv4 ICMP (1)	您電腦的公有 IPv4 地址、特定 IPv4 地址，或任何位置的 IPv4 或 IPv6 地址。		

若要使用 ping6 命令 ping 您執行個體的 IPv6 地址，您必須新增下列對內 ICMPv6 規則。

Type	通訊協定	來源		
所有 ICMP - IPv6	IPv6 ICMP (58)	您電腦的 IPv6 地址、特定 IPv4 地址，或任何位置的 IPv4 或 IPv6 地址。		

DNS 伺服器規則

若您已將您的 EC2 執行個體設為 DNS 伺服器，您必須確保 TCP 和 UDP 流量能透過連接埠 53 觸達您的 DNS 伺服器。

針對來源 IP，請指定下列其中一項：

- 網路中的特定 IP 地址或 IP 地址範圍 (CIDR 區塊標記法)
- 您網路中需要存取 DNS 伺服器之執行個體集的安全群組 ID

通訊協定類型	通訊協定號碼	連接埠
TCP	6	53
UDP	17	53

Amazon EFS 規則

若您正在搭配 Amazon EC2 執行個體使用 Amazon EFS 檔案系統，您與 Amazon EFS 掛載目標建立關聯的安全群組必須允許透過 NFS 通訊協定的流量。

通訊協定類型	通訊協定號碼	連接埠	來源 IP	備註
TCP	6	2049 (NFS)	安全群組的 ID	允許來自與此安全群組相關聯之資源 (包括掛載目標) 的對內 NFS 存取。

若要在您的 Amazon EC2 執行個體上掛載 Amazon EFS 檔案系統，您必須連線到您的執行個體。因此，與您的執行個體相關聯的安全群組必須具備允許來自您本機電腦或本機網路對內 SSH 的規則。

通訊協定類型	通訊協定號碼	連接埠	來源 IP	備註
TCP	6	22 (SSH)	您本機電腦的 IP 地址範圍，或是您網路的 IP 地址範圍 (CIDR 區塊標記法)。	允許來自您本機電腦的對內 SSH 存取。

Elastic Load Balancing 規則

若您正在使用負載平衡器註冊 EC2 執行個體，與您負載平衡器相關聯的安全群組必須具備允許和您的執行個體通訊的規則。如需詳細資訊，請參閱 Elastic Load Balancing 說明文件。

- Application Load Balancer 的安全群組

- Network Load Balancer 的安全群組
- 設定您的 Classic Load Balancer 的安全群組

Amazon EC2 執行個體的 NitroTPM

Nitro Trusted Platform Module (NitroTPM) 是由 AWS Nitro System 提供的虛擬裝置，符合 TPM 2.0 規格。它可安全存放用於對執行個體執行身分驗證的成品 (如密碼、憑證或加密金鑰)。NitroTPM 可產生金鑰，並將其用於密碼編譯功能 (例如雜湊、簽署、加密和解密)。

NitroTPM 會採用測量開機過程，在這個過程中，開機載入器和作業系統會建立各開機二進位的密碼編譯雜湊，並將其與 NitroTPM 內部平台組態註冊 (PCR) 中的過往值結合。透過測量開機，您可從 NitroTPM 獲得簽署的 PCR 值，然後使用這些值向遠端實體證明執行個體開機軟體的完整性。這稱為遠端證明。

透過 NitRotTPM，可以使用特定 PCR 值標記金鑰和秘密，這樣便可確保，如果 PCR 值和執行個體完整性發生變化，就再也無法存取這些金鑰和秘密。這種特殊形式的條件式存取被稱為密封和開封。BitLocker 等作業系統技術皆可使用 NitroTPM 來密封磁碟機解密金鑰，這樣便可確保，只有作業系統已正確開機且確定處於良好狀態時才能解密磁碟機。

若要使用 NitroTPM，必須選取已設定 NitroTPM 支援的 Amazon Machine Image (AMI)，然後使用 AMI 來啟動 Nitro 型執行個體。您可以選取一個 Amazon 預先建置的 AMI，或自行建立一個。

定價

使用 NitroTPM 無需額外付費。只需為您使用的基礎資源支付費用。

目錄

- 搭配 Amazon EC2 執行個體使用 NitroTPM 的需求
- 啟用 NitroTPM 的 Linux AMI
- 驗證已為 NitroTPM 啟用 AMI
- 啟用或停止 Amazon EC2 執行個體上的 NitroTPM
- 驗證已為 NitroTPM 啟用 Amazon EC2 執行個體
- 擷取 EC2 執行個體的公有背書金鑰

搭配 Amazon EC2 執行個體使用 NitroTPM 的需求

若要在啟用 NitroTPM 的情況下啟動執行個體，您必須符合以下要求。

主題

- AMI
- 執行個體類型
- 考量事項

AMI

AMI 必須已啟用 NitroTPM。

Linux AMI

無預先設定的 AMI。您必須設定自有 AMI。如需詳細資訊，請參閱啟用 NitroTPM 的 Linux AMI。

Windows AMI

若要尋找針對 NitroTPM 和 UEFI 安全開機搭配 Microsoft 金鑰預先設定的 AWS Windows AMI，請參閱 Windows AMIs參考中的尋找使用 NitroTPM 和 UEFI 安全開機設定的 Windows Server AMI。 AWS AMIs

> **Note**
>
> 作業系統 – AMI 必須包含具有 TPM 2.0 Command Response Buffer (CRB) 驅動程式的作業系統。最新的作業系統會包含 TPM 2.0 CRB 驅動程式。
> UEFI 開機模式 – 必須將 AMI 設定為 UEFI 開機模式。如需詳細資訊，請參閱Amazon EC2 執行個體的 UEFI 安全開機。

執行個體類型

您必須使用下列其中一個虛擬化執行個體類型：

- 一般用途：
  M5、M5a、M5ad、M5d、M5dn、M5n、M5zn、M6a、M6g、M6gd、M6i、M6id、M6idn、M6in、M7a、M
  flex、M8a、M8azn、M8g、M8gb、M8gd、M8gn、M8i、M8id、M8i-
  flex、M8in、M8idn、M8ine、M8ib、M8idb、M9g、M9gd、T3、T3a、T4g
- 運算最佳化：
  C5、C5a、C5ad、C5d、C5n、C6a、C6g、C6gd、C6gn、C6i、C6id、C6in、C7a、C7g、C7gd、C7gn、
  flex、C8a、C8g、C8gb、C8gd、C8gn、C8i、C8id、C8i-flex、C8in、C8ine、C8ib、C9g、C9gd

- 記憶體最佳化：R5、R5a、R5ad、R5b、R5d、R5dn、R5nR6a、R6g、R6gd、R6i、R6id、R6idn、R6in、R7a、R7g、R7gd、R7i、R7iz、R8a、R8g、R8gb、R flex、R8in、R8idn、R8ib、R8idb、R9g、R9gd、U7i-6tb、U7i-8tb、U7i-12tb、U7in-16tb、U7in-24tb、U7 X2idn X2iedn X2iezn X8g X8aedz X8i
- 儲存最佳化：D3、D3en、I3en、I4i、I7i、I7ie、I8g、I8ge、Im4gn
- 加速運算：F2、G4dn、G5、G6、G6e、G6f、Gr6、Gr6f、G7、G7e、Inf1、Inf2、P5、P5e、P5en、P6-B200、P6-B300、P6e-GB200、Trn2、Trn2u
- 高效能運算：Hpc6a、Hpc6id、Hpc8a

考量事項

使用 NitroTPM 時需要注意以下事項：

- 在您使用已啟用 NitroTPM 的 AMI 啟動執行個體後，若想要變更執行個體類型，您所選的新執行個體類型也必須支援 NitroTPM。
- 使用 NitroTPM 型金鑰加密的 BitLocker 磁碟區僅能於原始執行個體上使用。
- NitroTPM 狀態不會顯示在 Amazon EC2 主控台中。
- Amazon EBS 快照中不包含 NitroTPM 狀態。
- VM Import/Export 映像中不包含 NitroTPM 狀態。
- 上 AWS Outposts、本機區域或 Wavelength 區域不支援 NitroTPM。

啟用 NitroTPM 的 Linux AMI

若要為執行個體啟用 NitroTPM，您必須使用具有已啟用 NitroTPM 的 AMI 來啟動執行個體。在註冊時，您必須使用 NitroTPM 支援來設定 Linux AMI。之後便無法再設定 NitroTPM 支援。

如需了解針對 NitroTPM 支援預先設定的 Windows AMI 清單，請參閱 搭配 Amazon EC2 執行個體使用 NitroTPM 的需求。

您必須使用 RegisterImage API 來建立已設定 NitroTPM 的 AMI。您無法使用 Amazon EC2 主控台或 VM Import/Export。

啟用 NitroTPM 的 Linux AMI

1. 使用所需的 Linux AMI 來啟動暫時執行個體。請留意其根磁碟區的 ID，您可在執行個體的儲存體索引標籤上的主控台中找到此 ID。

2. 在執行個體達到 running 狀態後，請建立執行個體根磁碟區的快照。若要了解詳細資訊，請參閱建立 EBS 磁碟區的快照。
3. 將建立的快照註冊為 AMI。在區塊型儲存設備映射中，指定您為根磁碟區建立的快照。

   下方所示為 register-image 命令的範例。在 --tpm-support，請指定 v2.0。在 --boot-mode，請指定 uefi。

```
aws ec2 register-image \
    --name my-image \
    --boot-mode uefi \
    --architecture x86_64 \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0abcdef1234567890} \
    --tpm-support v2.0
```

   下方所示為 Register-EC2Image cmdlet的範例。

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
    -Name my-image `
    -Architecture "x86_64" `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
    -BootMode Uefi `
    -TpmSupport V20
```

4. 終止您在步驟 1 中啟動的暫時執行個體。

驗證已為 NitroTPM 啟用 AMI

若要為執行個體啟用 NitroTPM，您必須使用具有已啟用 NitroTPM 的 AMI 來啟動執行個體。您可對映像進行描述，以確定該映像已啟用 NitroTPM。若您是 AMI 擁有者，可對 tpmSupport 映像屬性進行描述。

Amazon EC2 主控台不會顯示 TpmSupport。

AWS CLI

如需確認是否已啟用 NitroTPM

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].TpmSupport
```

若已為 AMI 啟用 NitroTPM，則輸出中會如下所示。如果未啟用 TPM，則輸出為空白。

```
[
    "v2.0"
]
```

或者，若您是 AMI 擁有者，則可使用 tpmSupport 屬性並搭配 describe-image-attribute 命令。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute tpmSupport
```

以下為範例輸出。

```
{
    "ImageId": "ami-0abcdef1234567890",
    "TpmSupport": {
        "Value": "v2.0"
    }
}
```

如需尋找啟用 NitroTPM 的 AMI

在下面的範例中，列示了啟用 NitroTPM 時您所擁有的 AMI 的 ID。

```
aws ec2 describe-images \
    --owners self \
    --filters Name=tpm-support,Values=v2.0 \
    --query Images[].ImageId
```

PowerShell

如需確認是否已啟用 NitroTPM

使用 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | Select TpmSupport
```

若已為 AMI 啟用 NitroTPM，則輸出中會如下所示。如果未啟用 TPM，則輸出為空白。

```
TpmSupport
----------
v2.0
```

或者，若您是 AMI 擁有者，則可使用 `tpmSupport` 屬性並搭配 Get-EC2ImageAttribute cmdlet。

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute tpmSupport
```

如需尋找啟用 NitroTPM 的 AMI

在下面的範例中，列示了啟用 NitroTPM 時您所擁有的 AMI 的 ID。

```
Get-EC2Image `
    -Owner self `
    -Filter @{Name="tpm-support; Values="v2.0"} | Select ImageId
```

啟用或停止 Amazon EC2 執行個體上的 NitroTPM

您僅可在啟動時為 NitroTPM 啟用 Amazon EC2 執行個體。為 NitroTPM 啟用執行個體之後，您就無法停用它。若您不再需要使用 NitroTPM，就必須設定作業系統以停止使用。

主題

- 啟動執行個體並啟用 NitroTPM
- 停用執行個體的 NitroTPM

啟動執行個體並啟用 NitroTPM

在滿足先決條件的情況下啟動執行個體時，NitroTPM 會在執行個體上自動啟用。您僅可在執行個體啟動時啟用執行個體的 NitroTPM。如需有關啟動執行個體的詳細資訊，請參閱啟動 Amazon EC2 執行個體。

停用執行個體的 NitroTPM

啟動執行個體並啟用其 NitroTPM 後，便無法停用該執行個體的 NitroTPM。但是，您可以使用以下工具，停用執行個體上的 TPM 2.0 裝置驅動程式，從而將作業系統設定為停止使用 NitroTPM：

- 若為 Linux 執行個體，請使用 tpm-tools。
- 若為 Windows 執行個體，請使用 TPM 管理主控台 (tpm.msc)。

如需有關停用裝置驅動程式的詳細資訊，請參閱適用於您作業系統的相關文件。

驗證已為 NitroTPM 啟用 Amazon EC2 執行個體

您可確認 Amazon EC2 執行個體是否已啟用 NitroTPM。若在執行個體上已啟用 NitroTPM 支援，命令將傳回 `"v2.0"`。否則，輸出中不會存在 `TpmSupport` 欄位。

Amazon EC2 主控台不會顯示 `TpmSupport` 欄位。

AWS CLI

驗證是否已為 NitroTPM 啟動執行個體。

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].TpmSupport
```

PowerShell

驗證是否已為 NitroTPM 啟動執行個體。

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.TpmSupport
```

確認您的 Windows 執行個體上的 NitroTPM 存取權

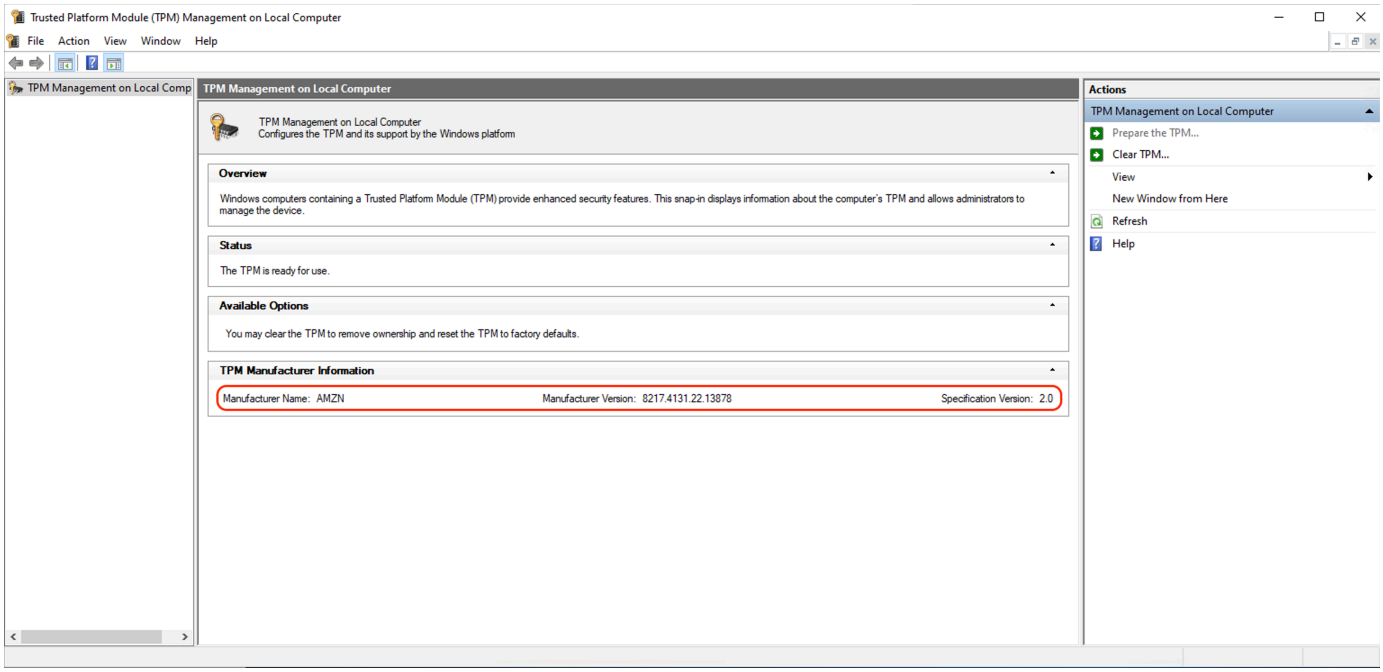
(僅限 Windows 執行個體) 驗證 NitroTPM 是否可存取 Windows

1. 連接至 EC2 Windows 執行個體。

2. 在執行個體上，執行 tpm.msc 程式。

   系統將會開啟 TPM Management on Local Computer (本機電腦上的 TPM 管理) 視窗。

3. 查看 TPM Manufacturer Information (TPM 製造商資訊) 欄位。此欄位中包含製造商的名稱和執行個體上的 NitroTPM 版本。



擷取 EC2 執行個體的公有背書金鑰

您隨時可以安全地擷取執行個體的公有背書金鑰。

AWS CLI

擷取執行個體的公有背書金鑰

請使用 get-instance-tpm-ek-pub 命令。

範例 1

以下範例會取得指定執行個體的 rsa-2048 公有背書金鑰 (格式為 tpmt)。

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format tpmt \
    --key-type rsa-2048
```

下列為範例輸出。

```
{
    "InstanceId": "i-01234567890abcdef",
    "KeyFormat": "tpmt",
    "KeyType": "rsa-2048",
    "KeyValue": "AAEACwADALIAIINxl2dEhLEXAMPLEUal1yT9UtduBlILZPKh2hszFGmqAAYAgABDA
    EXAMPLEAAABAOiRd7WmgtdGNoV1h/AxmW+CXExblG8pEUfNm0LOLiYnEXAMPLERqApiFa/UhvEYqN4
    Z7jKMD/usbhsQaAB1gKA5RmzuhSazHQkax7EXAMPLEzDthlS7HNGuYn5eG7qnJndRcakS+iNxT8Hvf
    0S1ZtNuItMs+Yp4SO6aU28MT/JZkOKsXIdMerY3GdWbNQz9AvYbMEXAMPLEPyHfzgVO0QTTJVGdDxh
    vxtXCOu9GYf0crbjEXAMPLEd4YTbWdDdgOKWF9fjzDytJSDhrLAOUctNzHPCd/92l5zEXAMPLEOIFA
    Ss50C0/802c17W2pMSVHvCCa9lYCiAfxH/vYKovAAE="
}
```

範例 2

以下範例會取得指定執行個體的 rsa-2048 公有背書金鑰 (格式為 der)。

```
aws ec2 get-instance-tpm-ek-pub \
    --instance-id i-1234567890abcdef0 \
    --key-format der \
    --key-type rsa-2048
```

下列為範例輸出。

```
{
    "InstanceId": "i-1234567890abcdef0",
    "KeyFormat": "der",
    "KeyType": "rsa-2048",
    "KeyValue": "MIIBIjANBgEXAMPLEw0BAQEFAAOCAQ8AMIIBCgKCAQEA6JF3taEXAMPLEXWH8DGZb4
    JcTFuUbykRR82bQs4uJifaKSOv5NGoEXAMPLEG8Rio3hnuMowP+6xuGxBoAHWAoDlGbO6FJrMdEXAMP
    LEnYUHvMO2GVLsc0a5ifl4buqcmd1FxqRL6I3FPwe9/REXAMPLE0yz5inhI7ppTbwxP8lmQ4qxch0x6
    tjcZ1Zs1DP0EXAMPLERUYLQ/Id/OBU7RBNMlUZ0PGG/G1cI670Zh/RytuOdx9iEXAMPLEtZ0N2A4pYX
    1+PMPK0lIOGssA5Ry03Mc8J3/3aXnOD2/ASRQ4gUBKznQLT/zTZEXAMPLEJUe8IJr2VgKIB/Ef+9gqi
    8AAQIDAQAB"
}
```

PowerShell

擷取執行個體的公有背書金鑰

使用 Get-EC2InstanceTpmEkPub cmdlet。

範例 1

以下範例會取得指定執行個體的 rsa-2048 公有背書金鑰 (格式為 tpmt)。

```
Get-EC2InstanceTpmEkPub `
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat tpmt `
    -KeyType rsa-2048
```

範例 2

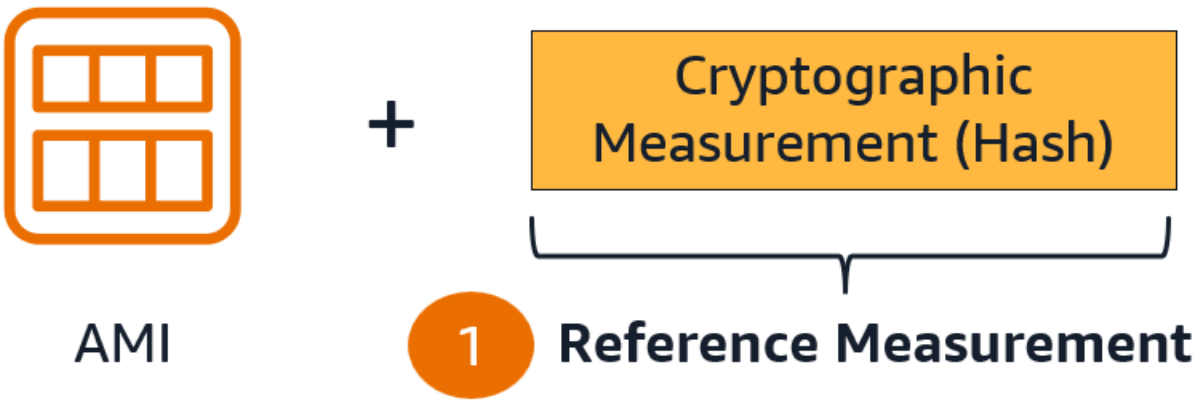
以下範例會取得指定執行個體的 rsa-2048 公有背書金鑰 (格式為 der)。

```
Get-EC2InstanceTpmEkPub `
    -InstanceId i-1234567890abcdef0 `
    -KeyFormat der `
    -KeyType rsa-2048
```

Amazon EC2 執行個體驗證

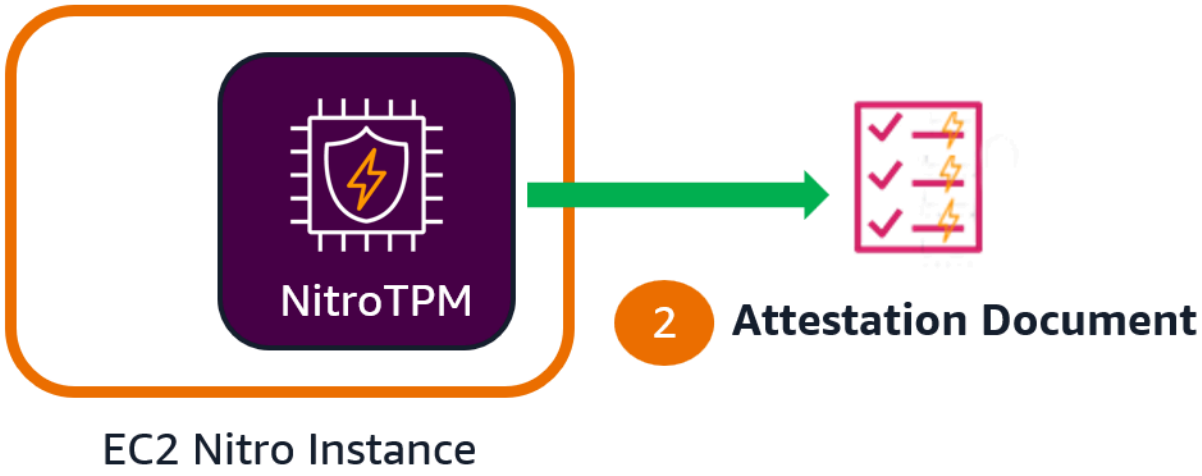
驗證是一個程序，讓您能夠以密碼編譯方式，向任何一方證明僅可信軟體、驅動程式及啟動程序在 Amazon EC2 執行個體上執行。Amazon EC2 執行個體驗證採用 Nitro 可信平台模組 (NitroTPM) 與可驗證的 AMI 技術。

驗證的第一步是建置可驗證的 AMI，以及確定該 AMI 的參考衡量項。可驗證的 AMI 是從頭開始建置以進行驗證的 AMI。參考衡量項是您納入 AMI 的所有軟體與組態的衡量項目。若要了解如何獲取參考衡量項的詳細資訊，請參閱 建置範例映像說明。



下一步是透過可驗證的 AMI，來啟動一個支援 Nitro-TPM 的 EC2 執行個體。在您啟動該執行個體之後，可使用 NitroTPM 工具來生成驗證文件。您隨後可將驗證文件中的 EC2 執行個體實際衡量項與參考衡量項做比較，以便查看執行個體是否使用您信任的軟體與組態。

藉由比較可驗證的 AMI 建立程序期間生成的參考衡量項與執行個體驗證文件中包括的衡量項，您可確認僅受信任的軟體與程式碼會在執行個體上執行。



與 整合 AWS KMS

若要更輕鬆地比較測量結果，您可以使用 AWS Key Management Service (AWS KMS) 做為證明文件的驗證者。使用 時 AWS KMS，您可以建立以認證為基礎的 KMS 金鑰政策，只有在您提供具有符合參考測量之測量的證明文件時，才允許使用 KMS 金鑰進行特定操作。如需執行此操作，可將特定條件索引鍵新增至將參考衡量項用做條件金鑰值的 KMS 金鑰政策，然後指定滿足條件索引鍵時允許哪些 KMS 作業。

當您使用 KMS 金鑰執行 KMS 操作時，您必須將證明文件連接至 KMS 請求。 AWS KMS 然後， 會根據 KMS 金鑰政策中的參考測量驗證證明文件中的測量，並只在測量相符時允許金鑰存取。

另外，若針對執行個體生成驗證文件，必須為您擁有的金鑰對指定公有金鑰。驗證文件內包括指定的公有金鑰。當 AWS KMS 驗證證明文件並允許解密操作時，它會先使用證明文件中包含的公有金鑰自動加密回應，然後再傳回。此操作可確保能夠對回應解密，並且僅可搭配驗證文件內包括的公有金鑰相符的私有金鑰使用。

這樣一來，確保僅在可信軟體與程式碼執行的執行個體，能夠使用 KMS 金鑰來執行密碼編譯作業。

驗證隔離的運算環境

一般來說，您可建置 EC2 執行個體，以及將其設定為隔離的運算環境，這不會提供互動式存取，亦不會設置機制讓管理員及使用者可存取 EC2 執行個體中正在處理的資料。藉助 EC2 執行個體驗證，您可向第三方或服務證明，執行個體以隔離式運算環境執行。如需詳細資訊，請參閱將資料與您自己的運算子隔離。

若要了解範例，請參閱建立隔離式運算環境的範例 Amazon Linux 2023 映像說明。您可將此範例映像說明用做起點，然後自訂說明來滿足您的要求。

AWS 共同責任模型

NitroTPM 與可驗證的 AMI 是建置區塊，能夠協助您在 EC2 執行個體上建立及設定驗證。您負責設定 AMI，以便滿足您各自使用案例的要求。若要了解詳細資訊，請參閱 AWS 共同責任模型。

主題

- 可驗證的 AMI
- AWS KMS 準備證明
- 獲取 NitroTPM 驗證文件
- 與 整合 AWS KMS
- 將資料與您自己的運算子隔離

可驗證的 AMI

可驗證的 AMI 是一種 Amazon Machine Image (AMI)，包含代表其全部內容的對應密碼編譯雜湊。AMI 建立程序期間會生成雜湊，且依據該 AMI 的整個內容進行計算，包括應用程式、程式碼，以及啟動程序。

維護可驗證狀態

執行個體的衡量項基於其初始啟動狀態。啟動之後對執行個體做出任何軟體或程式碼變更，以及重新啟動之後持續存在，都會在重新啟動之後變更執行個體的衡量項。若更改衡量項，則會偏離可驗證的 AMI 的參考衡量項，並且執行個體重新啟動之後，執行個體不再能夠對 AWS KMS 成功進行驗證。因此，為了讓可驗證的 AMI 有用，執行個體需要在重新啟動之後返回其原始啟動狀態。

始終返回原始啟動狀態，能夠確保執行個體在重新啟動之後成功進行驗證。可以使用下面的公用程式，確保執行個體在重新啟動之後仍可驗證：

- `erofs` – 增強了唯讀檔案系統。此公用程式能確保根檔案系統為唯讀。將此公用程式寫入檔案系統，包括 `/etc`、`/run` 及 `/var`，存放於記憶體，以及在執行個體重新啟動時遺失，並讓根檔案系統處於其原始啟動狀態。若要了解詳細資訊，請參閱 erofs 文件。
- `dm-verity` – 針對唯讀根檔案系統提供完整性防護。公用程式可計算檔案系統區塊的雜湊，並將其存放於核心命令列。此操作可讓核心在啟動期間確認檔案系統的完整性。若要了解詳細資訊，請參閱 dm-verity 文件。

建立可驗證的 AMI 的要求

可驗證的 AMI 具有以下要求：

- 基本作業系統 – Amazon Linux 2023 與 NixOS
- 架構 – `x86_64` 或 `arm64` 架構
- TPM 支援 – 必須啟用 NitroTPM。如需詳細資訊，請參閱搭配 Amazon EC2 執行個體使用 NitroTPM 的需求。
- 啟用模式 – 必須啟用 UEFI 啟動模式。

主題

- 建立可驗證的 AMI
- 建置範例 Amazon Linux 2023 映像說明
- 範例 Amazon Linux 2023 映像說明
- 針對您的工作負載，自訂範例 Amazon Linux 2023 映像說明
- 使用自訂 AMI 的運算 PCR 衡量項

建立可驗證的 AMI

如需建立可驗證的 AMI，您將需要使用 KIWI Next Generation (KIWI NG) 並搭配 Amazon Linux 2023。Amazon Linux 2023 提供了使用 KIWI NG 建置可驗證的 AMI 所需的全部軟體及公用程式。

KIWI NG 是一款開放原始碼工具，可用於建置預先設定的基於 Linux 的映像。KIWI NG 使用可定義映像內容的 XML 映像說明。影像描述指定要執行的基本作業系統、軟體、核心組態和指令碼，以針對特定使用案例建置ready-to-use AMI。

AMI 建置期間，您需要使用 nitro-tpm-pcr-compute 公用程式，依據 KIWI NG 生成的統一核心映像 (UKI) 來生成參考衡量項。若要了解使用 nitro-tpm-pcr-compute 公用程式的相關詳細資訊，請參閱 使用自訂 AMI 的運算 PCR 衡量項。

AWS 提供範例 Amazon Linux 2023 映像描述，其中包含在隔離的運算環境中設定 EC2 執行個體所需的所有組態。如需詳細資訊，請參閱建置範例 Amazon Linux 2023 映像說明。

建置範例 Amazon Linux 2023 映像說明

AWS 提供範例 Amazon Linux 2023 映像描述，您可以用作為工作負載建立自訂可認證 AMIs起點。範例映像說明包括做為基本作業系統的 Amazon Linux 2023，以及實現檔案系統不可變性的 dm-verity 與 erofs 組態，並且該說明移除了全部互動式存取權 (例如 SSH、EC2 Instance Connect，以及序列主控台)，以建立隔離式運算環境。若要了解範例映像說明的詳細資訊，請參閱 Github 儲存庫。

範例映像說明可在 /usr/bin/ 目錄的建置映像中，自動安裝 NitroTPM 工具 (nitro-tpm-pcr-compute 與 nitro-tpm-attest)。這樣一來，確保工具會預先安裝在透過 AMI 啟動的執行個體上。

範例影像說明包括一個指令碼 edit_boot_install.sh，其中包括生成參考衡量項所需的命令。指令碼可將 KIWI NG 建立的原始磁碟映像檔案 (.raw) 掛載至迴路裝置，尋找具有 .efi 副檔名的 UKI，然後執行 nitro-tpm-pcr-compute 公用程式來生成 AMI 的參考衡量項。在建置期間，KIWI NG 可自動執行指令碼。

本教學課程向您介紹了如何建置範例映像說明，以便建立可驗證的 AMI。

若要了解建立自己的映像說明的相關詳細資訊，請參閱下面的 KIWI NG 文件：

- Quick Start
- 映像說明
- 範例 Amazon Linux 2023 映像說明

先決條件

您的 IAM 身分必須具有下列許可才能完成本教學課程：

- ebs:CompleteSnapshot、 ebs:StartSnapshot和 ebs:PutSnapshotBlock上的 arn:aws:ec2:*::snapshot/*
- ec2:RegisterImage 在所有資源上

如需使用 KIWI NG 來建置範例 Amazon Linux 2023 映像說明

1. 使用最新的 AL2023 AMI 來啟動 Amazon EC2 執行個體 如需確保執行個體具有足夠的儲存空間以建置 AMI，請確保至少佈建 12 GB 的儲存空間。
2. 安裝所需的依存項目。使用下面的命令可安裝以下公用程式：

   - kiwi-cli
   - veritysetup
   - erofs-utils
   - aws-nitro-tpm-tools

   ```
   sudo dnf install -y kiwi-cli python3-kiwi kiwi-systemdeps-core python3-poetry-core
    qemu-img veritysetup erofs-utils git cargo aws-nitro-tpm-tools
   ```

3. 安裝 coldsnap 公用程式。使用此公用程式，您可透過原始映像資料來建立 Amazon EBS 快照。您將使用此公用程式，透過 KIWI NG 建立的原始磁碟映像檔案來建立 EBS 快照。

   ```
   git clone https://github.com/awslabs/coldsnap.git
   cd coldsnap
   cargo install --locked coldsnap
   cd ..
   ```

4. 獲取範例映像說明檔案。

   ```
   sudo dnf install kiwi-image-descriptions-examples
   ```

   範例映像說明檔案可下載至以下目錄：/usr/share/kiwi-image-descriptions-examples/al2023/attestable-image-example

5. 使用 KIWI NG system build 命令來建置範例映像說明。使用下面的命令可在 ./image 目錄中建立原始磁碟映像檔案。

   ```
   sudo kiwi-ng \
   --color-output \
   --loglevel 0 \
   system build \
   --description /usr/share/kiwi-image-descriptions-examples/al2023/attestable-image-
   example \
   --target-dir ./image
   ```

若要了解詳細資訊，請參閱 kiwi-ng system build 文件。

6. 獲取 AMI 的參考衡量項。nitro-tpm-pcr-compute 公用程式可在上一步驟的映像建置期間生成衡量項。您可在以下檔案中尋找參考衡量項：./image/pcr_measurements.json。

   衡量項以下列 JSON 格式提供：

```
{
  "Measurements": {
    "HashAlgorithm": "SHA384 { ... }",
    "PCR4": "PCR4_measurement",
    "PCR7": "PCR7_measurement",
    "PCR12": "PCR12_measurement"
  }
}
```

7. 使用 coldsnap 公用程式可上傳 KIWI NG 建立的原始磁碟映像至 EBS 快照。使用命令可傳回快照 ID。請記下 ID，您在後續步驟會用到。

```
SNAPSHOT=$(.cargo/bin/coldsnap upload ./image/al2023*.raw)
echo "Created snapshot: $SNAPSHOT"
```

   若要了解 coldsnap 公用程式的相關詳細資訊，請參閱 coldsnap GitHub 儲存庫。

8. 使用上一步驟的快照，在 UEFI 啟動模式下註冊支援 TPM 2.0 的 AMI。若是 --architecture，針對 Intel 指定 x86_64，或針對 Graviton 指定 arm64。

```
aws ec2 register-image \
--name "attestable_isolated_al2023_ami" \
--virtualization-type hvm \
--boot-mode uefi \
--architecture x86_64|arm64 \
--root-device-name /dev/xvda \
--block-device-mappings DeviceName=/dev/xvda,Ebs={SnapshotId=${SNAPSHOT}} \
--tpm-support v2.0 \
--ena-support
```

範例 Amazon Linux 2023 映像說明

範例 Amazon Linux 2023 映像說明有以下特性：

1. 統一核心映像 (UKI) 啟動 – 使用單一且簽署的二進位檔啟動，可將核心、initrd 及啟動參數組合為一個不可變映像。
2. 唯讀根檔案系統 – 使用增強的唯讀檔案系統 (erofs) 搭配 dm-verity 防護，來確保無法修改根檔案系統，以及確保密碼編譯完整性驗證。
3. 臨時堆疊式檔案系統 – 建立允許臨時寫入 /etc、/run 及 /var 等目錄的臨時堆疊式檔案系統。由於此堆疊式檔案系統僅存在於記憶體中，一切變更皆會在執行個體重新啟動後自動遺失，從而確保系統返回到其原始可信狀態。
4. 停用的遠端存取方法 – 移除下面的遠端存取機制以阻止遠端存取：

存取方法	說明	映像說明實作
SSH	排除 OpenSSH 伺服器。讓執行個體本質上不能處理 SSH 流量。	忽略 openssh-server 套件 *
使用者資料	移除 Cloud-init。讓操作員無法向執行個體提供使用者資料，以及在啟動時執行指令碼。	忽略 cloud-init 與 cloud-init-cfg-ec2 套件 *
Chrony	停用 chrony 命令連接埠。阻止運算子在執行中的執行個體上執行 chrony 命令。	忽略 amazon-chrony-config 套件 *
MOTD	移除 MOTD 套件。讓操作員無法在執行中的執行個體上變更訊息或功能。	忽略 update-motd 套件 *
AWS SSM	移除 AWS SSM 代理程式。防止使用 AWS SSM 遠端存取執行中的執行個體。	忽略 amazon-ssm-agent 套件 *
EC2 執行個體連線	移除 EC2 Instance Connect 套件。使用此工具來停用 SSH 存取權。	忽略 ec2-instance-connect 套件 *
序列主控台	停用序列主控台。確保執行中的執行個體無法使用主控台存取權，以及讓操作員無法登入序列主控台。	透過核心命令列參數停用

* 若要了解詳細資訊，請參閱映像說明元素。

針對您的工作負載，自訂範例 Amazon Linux 2023 映像說明

您可自訂範例 Amazon Linux 2023 映像說明，以及包括您的特定工作負載所需的軟體套件、指令碼，以及檔案。在 KIWI NG 映像說明中新增或修改各種元素，藉此來實現自訂。

主題

- 儲存庫管理
- 套件管理
- 新增檔案及目錄
- 新增自訂指令碼

儲存庫管理

範例映像說明預設會包括單一 <repository> 元素，其會指向適用於 Amazon Linux 2023 核心儲存庫的鏡像端點。您可視需要新增參考至其他儲存庫，以便從中安裝您的必要軟體。

範例映像說明使用如 <packagemanager> 元素中所定義的 dnf 套件管理員。

若要了解新增儲存庫的相關詳細資訊，請參閱設定儲存庫。

套件管理

範例映像說明預設會包括所需的一切套件，用於針對包含 erofs 唯讀檔案系統的隔離式運算環境來建立 Amazon Linux 2023 可驗證的 AMI。

您可新增額外的軟體套件至映像說明中的 <packages> 元素，藉此在映像說明中加入這些額外的軟體套件。<packages> 元素可定義應安裝至 AMI 的全部軟體。

您亦可使用 <packages> 元素，來解除安裝或刪除特定軟體套件。

若要了解在映像說明中新增或移除套件的相關詳細資訊，請參閱新增及移除套件。

新增檔案及目錄

範例映像說明包括堆疊式樹狀目錄 (/root/)。堆疊式樹狀目錄是一個包含檔案與目錄的目錄，這些檔案與目錄會在映像建置程序期間複製到映像中。您在堆疊式樹狀目錄中置放的任何檔案及目錄，都會在映像建置程序期間直接複製到映像的根檔案系統。

安裝全部套件之後，堆疊式樹狀目錄即會複製到映像中。檔案即可新增，且會覆寫現有檔案。

新增自訂指令碼

範例映像說明包括單一自訂指令碼 `edit_boot_install.sh`。此指令碼包括執行 `nitro-tpm-pcr-compute` 公用程式所需的命令，這些命令可依據映像內容生成參考衡量項。安裝啟動載入程式之後，即可呼叫此指令碼。

您可視需要在映像說明中加入您自己的自訂指令碼，以便在映像建置程序期間或首次啟動映像時執行任務或組態。使用指令碼，您能夠以單獨使用映像說明無法實現的方式來自訂您的映像。

如需在您的映像說明中加入自訂指令碼，您需要依據指令碼類型來對指令碼正確命名，然後將其新增至與 `appliance.kiwi` 檔案相同的目錄。若對指令碼正確命名且置於正確位置，KIWI NG 無需在映像說明檔案中明確參考這些指令碼，即可自動偵測及執行指令碼。

若要了解 KIWI NG 支援的指令碼相關詳細資訊，請參閱使用者定義的指令碼。

使用自訂 AMI 的運算 PCR 衡量項

使用 `nitro-tpm-pcr-compute` 公用程式，您可依據統一核心映像 (UKI)，在建置期間針對可驗證的 AMI 生成參考衡量項。

範例 Amazon Linux 2023 映像說明可在 `/usr/bin/` 目錄的建置映像中，自動安裝公用程式。範例映像說明還包括指令碼，其中包含映像建置期間執行公用程式生成參考衡量項所需的命令。若使用範例映像說明，則您無須安裝公用程式或手動執行。如需詳細資訊，請參閱建置範例 Amazon Linux 2023 映像說明。

安裝 **nitro-tpm-pcr-compute** 公用程式

若使用 Amazon Linux 2023，則您可如下所示透過 Amazon Linux 儲存庫安裝 `nitro-tpm-pcr-compute` 公用程式。

```
sudo yum install aws-nitro-tpm-tools
```

這些工具皆安裝於 `/usr/bin` 目錄。

使用 **nitro-tpm-pcr-compute** 公用程式

公用程式提供了單一命令 `nitro-tpm-pcr-compute` 來生成參考衡量項。

您在執行命令時可指定以下各項：

- 統一核心映像 (`UKI.efi`) – 標準啟動及 UEFI 的必要項目。

如需生成可驗證的 AMI 的參考衡量項：

請使用下列命令和參數：

```
/usr/bin/nitro-tpm-pcr-compute \
--image UKI.efi
```

公用程式能夠以下列 JSON 格式來傳回參考衡量項：

```
{
  "Measurements": {
    "HashAlgorithm": "SHA384 { ... }",
    "PCR4": "PCR4_measurement",
    "PCR7": "PCR7_measurement",
    "PCR12": "PCR12_measurement"
  }
}
```

若要了解如何使用 `nitro-tpm-pcr-compute` 公用程式的實際範例，請參閱範例 Amazon Linux 2023 映像說明中包括的 `edit_boot_install.sh` 指令碼。

AWS KMS 準備證明

> **Note**
>
> 若要對第三方服務進行驗證，必須建置您自己的自訂機制，以便接收、剖析及確認驗證文件。如需詳細資訊，請參閱確認 NitroTPM 驗證文件。

建立可認證 AMI 之後，您應該有參考測量，可用來驗證來自 Amazon EC2 執行個體的請求。 AWS KMS 提供內建的 NitroTPM 認證支援。

針對您用來加密秘密資料的 AWS KMS 金鑰，請新增金鑰政策，只有在 API 請求包含與您在可認證 AMI 建立程序期間產生之參考測量相符的證明文件時，才允許金鑰存取。針對標準開機使用 PCR4 和 PCR12 測量，或針對安全開機使用 PCR7 測量。這可確保只有來自使用可認證 AMI 啟動之執行個體的請求可以使用 AWS KMS 金鑰執行密碼編譯操作。

AWS KMS 提供 `kms:RecipientAttestation:NitroTPMPCR4`、`kms:RecipientAttestation:NitroTPMPCR7`和

kms:RecipientAttestation:NitroTPMPCR12條件金鑰，可讓您為 NitroTPM KMS 金鑰政策建立以認證為基礎的條件。若要了解詳細資訊，請參閱 NitroTPM 的條件金鑰。

例如，以下 AWS KMS 金鑰政策僅在請求來自連接執行個體描述檔的MyEC2InstanceRole執行個體，以及請求包含具有特定 PCR 4 和 PCR 12 值的證明文件時，才允許金鑰存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow requests from instances with attested AMI only",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyEC2InstanceRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateRandom"
      ],
      "Resource": "*",
      "Condition": {
        "StringEqualsIgnoreCase": {

 "kms:RecipientAttestation:NitroTPMPCR4":"EXAMPLE6b9b3d89a53b13f5dfd14a1049ec0b80a9ae4b159adde4

 "kms:RecipientAttestation:NitroTPMPCR12":"000000000000000000000000000000000000000000000000000
        }
      }
    }
  ]
}
```

獲取 NitroTPM 驗證文件

驗證文件是 NitroTPM 驗證程序的關鍵元件。該文件包含一系列密碼編譯衡量項，可用於確認執行個體身分，以及證明其僅執行可信軟體。您可以搭配 使用 證明文件 AWS KMS，該文件提供 NitroTPM 證明的內建支援，或建置您自己的密碼編譯證明機制。

使用 nitro-tpm-attest 公用程式，您可在執行時期針對 Amazon EC2 執行個體，擷取簽署的 NitroTPM 驗證文件。

範例 Amazon Linux 2023 映像說明可在 `/usr/bin/` 目錄的建置映像中，自動安裝公用程式。這樣一來，確保公用程式會預先安裝在使用 AMI 啟動的執行個體上。您無須手動安裝公用程式。如需詳細資訊，請參閱建置範例 Amazon Linux 2023 映像說明。

主題

- 安裝 nitro-tpm-attest 公用程式
- 使用 nitro-tpm-attest 公用程式
- NitroTPM 驗證文件內容
- 確認 NitroTPM 驗證文件

安裝 `nitro-tpm-attest` 公用程式

若使用 Amazon Linux 2023，則您可如下所示透過 Amazon Linux 儲存庫安裝 `nitro-tpm-attest` 公用程式。

```
sudo yum install aws-nitro-tpm-tools
```

使用 `nitro-tpm-attest` 公用程式

公用程式提供單一命令 `nitro-tpm-attest`，以便擷取驗證文件。命令可傳回以 Concise Binary Object Representation (CBOR) 編碼，以及使用 CBOR 物件簽署及加密 (COSE) 簽署的驗證文件。

當您執行命令時，可以指定下列選用參數：

- `public-key` — 可由 AWS KMS 或外部服務用來在傳回回應資料之前加密回應資料的公有金鑰。該金鑰可確保僅擁有私有金鑰的預期收件者才可對資料解密。例如，如果您使用 進行證明 AWS KMS，服務會使用證明文件中的公有金鑰加密純文字資料，並在回應的 `CiphertextForRecipient` 欄位中傳回產生的加密文字。僅支援 RSA 金鑰。
- `user-data` – 使用者資料可用於傳遞任何額外簽署的資料至外部服務。可使用此使用者資料，完成請求執行個體與外部服務間達成的協定。不適用於 的證明 AWS KMS。
- `nonce` – 可使用 nonce 在執行個體與外部服務間設定挑戰-回應驗證，以便協助阻止模擬攻擊。使用 nonce 可讓外部服務確認其是否與即時執行個體，而非與重複使用舊驗證文件的冒充者互動。不適用於 的證明 AWS KMS。

如需擷取驗證文件

請使用下列命令和選用參數：

```
/usr/bin/nitro-tpm-attest \
--public-key rsa_public_key \
--user-data user_data \
--nonce nonce
```

若要了解展示如何生成 RSA 金鑰對，以及如何透過公有金鑰來請求驗證的完整範例，請參閱 nitro-tpm-attest GitHub 儲存庫。

NitroTPM 驗證文件內容

驗證文件由 NitroTPM 生成，且由 Nitro Hypervisor 簽署。該文件包含一系列與 Amazon EC2 執行個體相關的平台組態暫存器 (PCR) 值。驗證文件包括下面的 PCR：

> **Important**
>
> PCR0 與 PCR1 一般用於測量由 AWS控制的初始啟動碼。如需允許早期啟動程式碼進行安全更新，這些 PCR 始終會含常數值。

- PCR0 – 核心系統韌體可執行程式碼
- PCR1 – 核心系統韌體資料
- PCR2 – 延伸或插入式可執行程式碼
- PCR3 – 延伸或插入式韌體資料
- PCR4 – 啟動管理員程式碼
- PCR5 – 啟動管理員程式碼組態及資料與 GPT 分區資料表
- PCR6 – 主機平台製造商規格
- PCR7 – 安全啟動政策
- PCR8 - 15 – 定義用於靜態作業系統
- PCR16 – 偵錯
- PCR23 – 應用程式支援

PCR4、PCR7 和 PCR12 專門用於驗證執行個體是否使用可證明的 AMI 啟動。PCR4 和 PCR12 可用於使用標準開機進行驗證，而 PCR7 可用於使用安全開機進行驗證。

- PCR4 (啟動管理員程式碼) – 若執行個體啟動，NitroTPM 可建立其 UEFI 環境執行的全部二進位檔的密碼編譯雜湊。藉助可驗證的 AMI，這些啟動二進位檔可嵌入雜湊，以阻止未來載入無相符雜湊的二進位檔。這樣一來，單一啟動二進位雜湊可準確描述執行個體會執行的程式碼。
- PCR7 (安全啟動政策) – 您可透過 UEFI 安全啟動簽署金鑰，來簽署 UEFI 啟動二進位檔。若啟動 UEFI 安全啟動，UEFI 將會阻止執行與設定政策不相符的 UEFI 啟動二進位檔。PCR7 包含執行個體 UEFI 安全啟動政策的雜湊。

  若需要維護跨執行個體更新持續存在的單一 KMS 政策，您可建立一項針對 PCR7 進行驗證的政策，以便確認 UEFI 安全啟動憑證。建立可驗證的 AMI 期間，您隨後可透過憑證來簽署啟動二進位檔，然後做為 AMI 的 UEFI 資料中唯一允許的憑證進行安裝。請謹記，此模型仍要求生成新的憑證、在政策中安裝憑證，以及在您想要阻止透過舊的 (不可信) AMI 啟動的執行個體來傳遞 KMS 政策的情況下更新 AMI。
- PCR12 — 包含傳遞給 UEFI 開機二進位檔的命令列雜湊。與標準開機的 PCR4 搭配使用，以驗證命令列未修改。

確認 NitroTPM 驗證文件

> **Note**
>
> 本主題旨在面向使用第三方金鑰管理服務，以及需要建置其驗證文件確認機制的使用者。

本主題提供了整個 NitroTPM 驗證流程的詳細概觀。它也會討論請求證明文件時 AWS Nitro 系統所產生的內容，並說明金鑰管理服務應如何處理證明文件。

主題

- 驗證文件
- 驗證文件確認

驗證旨在依據執行個體在執行的程式碼與組態，證明執行個體是一個值得信任的實體。執行個體的信任根位於 AWS Nitro 系統中，提供證明文件。

證明文件由 AWS Nitro Attestation Public Key Infrastructure (PKI) 簽署，其中包含可納入任何服務的已發佈憑證授權機構。

驗證文件

驗證文件以 Concise Binary Object Representation (CBOR) 編碼，以及使用 CBOR 物件簽署及加密 (COSE) 簽署。

若要了解 CBOR 的相關詳細資訊，請參閱 RFC 8949：Concise Binary Object Representation (CBOR)。

驗證文件規格

下面所示為驗證文件的結構。

```
AttestationDocument = {
    module_id: text,                    ; issuing Nitro hypervisor module ID
    timestamp: uint .size 8,            ; UTC time when document was created, in
                                        ; milliseconds since UNIX epoch
    digest: digest,                     ; the digest function used for calculating the
                                        ; register values
    nitrotpm_pcrs: { + index => pcr },  ; map of PCRs at the moment the Attestation
 Document was generated
    certificate: cert,                  ; the public key certificate for the public
 key
                                        ; that was used to sign the Attestation
 Document
    cabundle: [* cert],                 ; issuing CA bundle for infrastructure
 certificate
    ? public_key: user_data,            ; an optional DER-encoded key the attestation
                                        ; consumer can use to encrypt data with
    ? user_data: user_data,             ; additional signed user data, defined by
 protocol
    ? nonce: user_data,                 ; an optional cryptographic nonce provided by
 the
                                        ; attestation consumer as a proof of
 authenticity
}

cert = bytes .size (1..1024)       ; DER encoded certificate
user_data = bytes .size (0..1024)
pcr = bytes .size (32/48/64)       ; PCR content
index = 0..31
digest = "SHA384"
```

使用驗證文件 (public_key、user_data 及 nonce) 中的選用參數，可在驗證執行個體與外部服務間建立自訂驗證通訊協定。

驗證文件確認

若透過 Nitro Hypervisor 請求驗證文件，您會收到包含已簽署驗證文件的二進位 Blob。簽署的驗證文件為 CBOR 編碼、COSE 簽署 (使用 COSE_Sign1 簽章結構) 的物件。整個驗證程序包括以下步驟：

1. 解碼 CBOR 物件，以及將其映射至 COSE_Sign1 結構。
2. 透過 COSE_Sign1 結構擷取驗證文件。
3. 確認憑證的鏈結。
4. 確定驗證文件已適當簽署。

證明文件由 AWS Nitro Attestation PKI 簽署，其中包含商業 AWS 分割區的根憑證。根憑證可從 https://aws-nitro-enclaves.amazonaws.com/AWS_NitroEnclaves_Root-G1.zip 下載，並且可使用下面的指紋進行確認。

```
64:1A:03:21:A3:E2:44:EF:E4:56:46:31:95:D6:06:31:7E:D7:CD:CC:3C:17:56:E0:98:93:F3:C6:8F:79:BB:5B
```

根憑證是以 AWS Certificate Manager 私有憑證授權單位 (AWS 私有 CA) 私有金鑰為基礎，生命週期為 30 年。PCA 的主旨採用下列格式。

```
CN=aws.nitro-enclaves, C=US, O=Amazon, OU=AWS
```

主題

- COSE 與 CBOR
- 語意有效性
- 憑證有效性
- 憑證鏈有效性

COSE 與 CBOR

通常，若訊息上僅置放一個簽章，則會使用 COSE_Sign1 簽章結構。處理內容與簽章的參數將會置放於受到保護的標頭中，而非分隔 COSE_Sign。視乎將在其中使用的內容，能夠以加標籤或不加標籤來對結構編碼。加標籤的 COSE_Sign1 結構可透過 CBOR 標籤 18 辨識。

承載內文的 CBOR 物件、簽章，以及內文及簽章的相關資訊稱為 COSE_Sign1 結構。COSE_Sign1 結構為 CBOR 陣列。陣列包括下列欄位。

```
[
  protected:   Header,
  unprotected: Header,
  payload:     This field contains the serialized content to be signed,
  signature:   This field contains the computed signature value.
]
```

在驗證文件的關聯內容中，陣列包括以下各項。

```
18(/* COSE_Sign1 CBOR tag is 18 */
    {1: -35}, /* This is equivalent with {algorithm: ECDS 384} */
    {}, /* We have nothing in unprotected */
    $ATTESTATION_DOCUMENT_CONTENT /* Attestation Document */,
    signature /* This is the signature */
)
```

若要了解 CBOR 的相關詳細資訊，請參閱 RFC 8949：Concise Binary Object Representation (CBOR)。

語意有效性

驗證文件的 CA 套件始終按下面的順序排列。

```
[ ROOT_CERT - INTERM_1 - INTERM_2 .... - INTERM_N]
      0          1          2             N - 1
```

請記住此排序，因為某些現有的工具可能需要以不同順序排列，例如，Java PKI API 程式設計人員指南中的 Java 的 CertPath。

如需驗證憑證，可從驗證文件 CA 套件開始，然後生成必要的憑證鏈，其中 TARGET_CERT 是驗證文件中的憑證。

```
[TARGET_CERT, INTERM_N, ..... , INTERM_2, INTERM_1, ROOT_CERT]
```

憑證有效性

對於憑證鏈中的所有憑證，必須確保目前日期在憑證中指定的有效期間範圍內。

憑證鏈有效性

一般來說，可能需要多個憑證鏈，且包含由一個 CA 簽署的公有金鑰擁有者憑證，以及由其他 CA 簽署的零或多個額外的 CA 憑證。由於公有金鑰使用者僅使用有限數量的保證 CA 公有金鑰來初始化，因此必須提供這類稱為憑證路徑的憑證鏈。網際網路 PKI 的憑證路徑驗證程序基於 X.509 中提供的演算法。憑證路徑處理可確認主體辨別名稱及/或主體替代名稱與主體公有金鑰間的繫結。繫結受限於憑證中指定的約束條件，這些約束條件包含依賴方指定的路徑與輸入。基本約束條件與政策可對延伸進行約束，允許憑證路徑處理邏輯來自動化決策程序。

> **Note**
>
> 執行驗證時，必須停用 CRL。

使用 Java，從根路徑與生成的憑證鏈開始，憑證鏈驗證如下所示。

```
validateCertsPath(certChain, rootCertficate) {
    /* The trust anchor is the root CA to trust */
    trustAnchors.add(rootCertificate);

    /* We need PKIX parameters to specify the trust anchors
     * and disable the CRL validation
     */
    validationParameters = new PKIXParameters(trustAnchors);
    certPathValidator = CertPathValidator.getInstance(PKIX);
    validationParameters.setRevocationEnabled(false);

    /* We are ensuring that certificates are chained correctly */
    certPathValidator.validate(certPath, validationParameters);
}
```

與 整合 AWS KMS

您的執行個體應該有一個應用程式，可以使用從 NitroTPM 擷取的證明文件提出 AWS KMS API 請求。當您使用證明文件提出請求時， AWS KMS 會根據 KMS 金鑰政策中的參考測量，驗證所提供證明文件中的測量。僅當驗證文件中的衡量項與 KMS 金鑰政策中的參考衡量項相符時，方可允許這些請求。

若透過驗證文件呼叫 Decrypt、DeriveSharedSecret、GenerateDataKey、GenerateDataKeyPair 或 GenerateRandom API 作業，這些 API 會加密來自驗證文件公開金鑰回應的純文字，並傳回密文而非純文字。僅當使用執行個體生成的私有金鑰時，才能解密此密文。

若要了解詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 NitroTPM 密碼編譯驗證。

> **Note**
>
> 若要對第三方服務進行驗證，必須建置您自己的自訂機制，以便接收、剖析及確認驗證文件。如需詳細資訊，請參閱確認 NitroTPM 驗證文件。

將資料與您自己的運算子隔離

AWS Nitro 系統具有零操作員存取。任何 AWS 系統或人員都無法登入 Amazon EC2 Nitro 主機、存取 EC2 執行個體的記憶體，或存取存放在本機加密執行個體儲存體或遠端加密 Amazon EBS 磁碟區上的任何客戶資料。

若要處理高度敏感資料，您可能會考慮甚至阻止您自己的操作員存取 EC2 執行個體，藉此來限制該資料的存取權。

您可建立自訂可驗證的 AMI，且設定為提供隔離式運算環境。AMI 組態視乎工作負載及應用程式需求而定。在建置您的 AMI 以建立隔離式運算環境時，考慮這些最佳實務。

- 移除全部互動式存取權，以便阻止操作員或使用者存取執行個體。
- 確認 AMI 中僅包括可信軟體與程式碼。
- 在執行個體內設定網路防火牆以封鎖存取權。
- 確認所有儲存體與檔案系統的唯讀與不可變狀態。
- 限制執行個體存取為經驗證、授權及記錄的 API 呼叫。

更新無互動式存取權的可驗證的 AMI

使用隔離的運算環境 AMI 啟動執行個體後，任何使用者或運算子都無法連線到執行個體。這意味著啟動之後，無法在執行個體上安裝或更新任何軟體。

若要求新的軟體或軟體更新，必須建立新的可驗證的 AMI，其中包括必要的軟體或軟體更新。然後，使用該 AMI 來啟動新的執行個體，或者在原始執行個體上執行根磁碟區取代。對 AMI 做出的任何軟體變更皆會生成新的雜湊。

下面的操作會導致 NitroTPM 驗證文件中參考衡量項發生變更：

- 停止及啟動透過可驗證的 AMI 啟動的執行個體

- 透過不同的 AMI 來執行根磁碟區取代

若執行任何這些操作，必須透過新的參考衡量項來更新驗證服務。例如，如果您使用 AWS KMS 進行證明，則必須將 KMS 金鑰政策更新為新的參考測量。

在整個執行個體生命週期，執行個體將保留其 NitroTPM 金鑰資料，以及透過停止/啟動及根磁碟區取代作業來持續存在。

Windows 執行個體的 Credential Guard

AWS Nitro 系統支援 Amazon Elastic Compute Cloud (Amazon EC2) Windows 執行個體的 Credential Guard。Credential Guard 是基於 Windows 虛擬化的安全性 (VBS) 功能，可讓您建立隔離環境以保護諸如 Windows 使用者憑證和程式碼完整性強制執行等安全性資產，而這些超出了 Windows 核心保護範圍。當您執行 EC2 Windows 執行個體時， Credential Guard 會使用 AWS Nitro 系統來保護 Windows 登入憑證，避免從作業系統的記憶體中擷取。

目錄

- 先決條件
- 啟動支援的執行個體
- 停用記憶體完整性
- 開啟 Credential Guard
- 驗證 Credential Guard 正在執行

先決條件

您的 Windows 執行個體必須符合下列先決條件，才能使用 Credential Guard。

Amazon Machine Images (AMI)

必須預先設定 AMI，才能啟用 NitroTPM 和 UEFI Secure Boot。如需受支援 AMI 的詳細資訊，請參閱 the section called “要求”。

記憶體完整性

不支援記憶體完整性 (也稱為受 Hypervisor 保護的程式碼完整性 (HVCI)) 或 Hypervisor 強制執行的程式碼完整性。開啟 Credential Guard 之前，您必須確定此功能已停用。如需詳細資訊，請參閱停用記憶體完整性。

執行個體類型

除非另有說明，否則下列執行個體類型的所有大小均支援 Credential Guard：C5、C5d、C5n、C6i、C6id、C6in、C7i、C7i-flex、M5、M5d、M5dn、M5n、M5zn、M6i、M6id、M6idn、M6in、M7i、M7i-flex、R5、R5b、R5d、R5dn、R5n、R6i、R6id、R6idn、R6in、R7i、R7iz、T3。

> **Note**
>
> - 雖然 NitRoTPM 有一些共同的必要執行個體類型，但執行個體類型必須是上述執行個體類型的其中一種，才能支援 Credential Guard。
> - Credential Guard 不支援：
>   - 裸機執行個體。
>   - 下列執行個體類型：C7i.48xlarge、M7i.48xlarge 和 R7i.48xlarge。

如需執行個體類型的詳細資訊，請參閱 Amazon EC2 執行個體類型指南。

啟動支援的執行個體

您可以使用 Amazon EC2 主控台或 AWS Command Line Interface (AWS CLI) 來啟動可支援 Credential Guard 的執行個體。您將需要一個相容 AMI ID 來啟動執行個體，該 ID 對於每個 AWS 區域都是唯一的。

> **Tip**
>
> 可以使用下列連結在 Amazon EC2 主控台中探索並啟動具有相容 Amazon 提供之 AMI 的執行個體：
> https://console.aws.amazon.com/ec2/v2/home?#Images:visibility=public-images;v=3;search=:TPM-Windows_Server;ownerAlias=amazon

Console

啟動執行個體

遵循步驟以啟動執行個體，同時指定支援的執行個體類型和預先設定的 Windows AMI。

AWS CLI

啟動執行個體

使用 run-instances 命令，利用支援的執行個體類型和預先設定的 Windows AMI 來啟動執行個體。

```
aws ec2 run-instances \
    --image-id resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base \
    --instance-type c6i.large \
    --region us-east-1 \
    --subnet-id subnet-0abcdef1234567890
    --key-name key-name
```

PowerShell

啟動執行個體

使用 New-EC2Instance 命令，利用支援的執行個體類型和預先設定的 Windows AMI 來啟動執行個體。

```
New-EC2Instance `
    -ImageId resolve:ssm:/aws/service/ami-windows-latest/TPM-Windows_Server-2022-English-Full-Base `
    -InstanceType c6i.large `
    -Region us-east-1 `
    -SubnetId subnet-0abcdef1234567890 `
    -KeyName key-name
```

停用記憶體完整性

在支援的情況下，您可以使用本機群組政策編輯器停用記憶體完整性。下列指引適用於程式碼完整性的虛擬化型保護下的各組態設定：

- 已啟用但未鎖定：將設定修改為已停用以停用記憶體完整性。
- 已透過 UEFI 鎖定啟用：已透過 UEFI 鎖定啟用記憶體完整性。使用 UEFI 鎖定啟用記憶體完整性之後，就無法停用記憶體完整性。建議您建立停用記憶體完整性的新執行個體，並終止未使用且不受支援的執行個體。

使用本機群組政策編輯器停用記憶體完整性

1. 使用遠端桌面通訊協定 (RDP) 以擁有管理員權限的使用者帳戶身分連線至執行個體。如需詳細資訊，請參閱the section called “使用 RDP 用戶端連線”。
2. 開啟「開始」選單並搜尋 **cmd** 以啟動命令提示。
3. 執行以下命令以開啟本機群組政策編輯器：`gpedit.msc`
4. 在本機群組政策編輯器中，選擇電腦設定、管理範本、系統、Device Guard。
5. 選取開啟基於虛擬化的安全性，然後選取編輯政策設定。
6. 開啟虛擬化型程式碼完整性保護的設定下拉式清單，選擇已停用，然後選擇套用。
7. 重新啟動執行個體以套用變更。

開啟 Credential Guard

啟動具有受支援執行個體類型和相容 AMI 的 Windows 執行個體，並確認已停用記憶體完整性之後，可以開啟 Credential Guard。

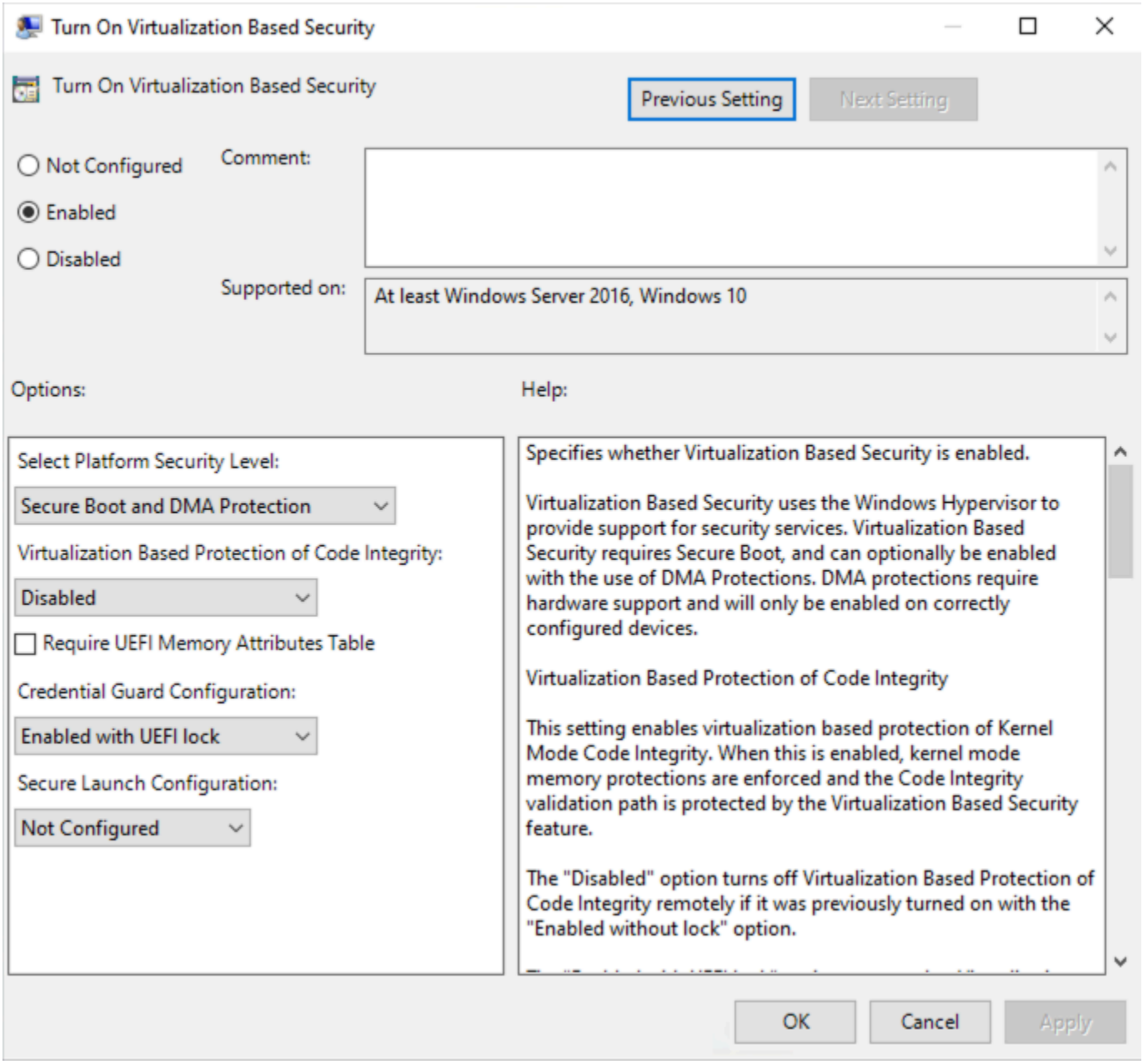
> **Important**
>
> 需要管理員權限執行以下步驟來開啟 Credential Guard。

開啟 Credential Guard

1. 使用遠端桌面通訊協定 (RDP) 以擁有管理員權限的使用者帳戶身分連線至執行個體。如需詳細資訊，請參閱the section called “使用 RDP 用戶端連線”。
2. 開啟「開始」選單並搜尋 **cmd** 以啟動命令提示。
3. 執行以下命令以開啟本機群組政策編輯器：`gpedit.msc`
4. 在本機群組政策編輯器中，選擇電腦設定、管理範本、系統、Device Guard。
5. 選取開啟基於虛擬化的安全性，然後選取編輯政策設定。
6. 在開啟基於虛擬化的安全性選單中選擇已啟用。
7. 針對選取平台安全性層級，選擇安全開機和 DMA 保護。
8. 針對 Credential Guard 組態，選擇已啟用且搭配 UEFI 鎖定。

> **Note**
>
> 其餘的政策設定不需要來啟用 Credential Guard，而且可以保留為未設定。

下圖顯示如先前所述配置的 VBS 設定：



9. 重新啟動執行個體以套用設定。

驗證 Credential Guard 正在執行

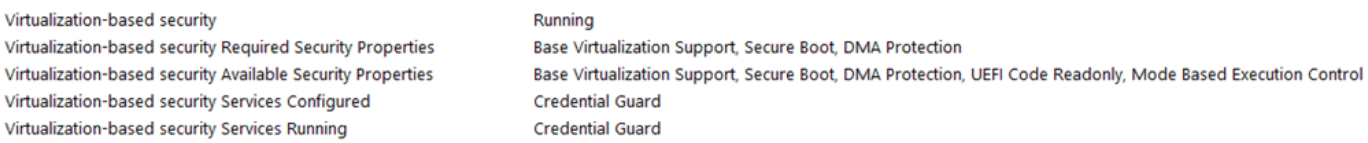
可以使用 Microsoft System Information (`Msinfo32.exe`) 工具來確認 Credential Guard 是否正在執行。

> **Important**
>
> 您必須先重新啟動執行個體，才能完成套用啟用 Credential Guard 所需的政策設定。

驗證 Credential Guard 正在執行

1. 使用遠端桌面通訊協定 (RDP) 連線至執行個體。如需詳細資訊，請參閱the section called “使用 RDP 用戶端連線”。
2. 在執行個體的 RDP 工作階段中，開啟「開始」選單並搜尋 **cmd** 以啟動命令提示。
3. 透過執行以下命令，開啟 System Information：`msinfo32.exe`
4. Microsoft System Information 工具會列出 VBS 組態的詳細資訊。在基於虛擬化的安全性服務旁邊，確認 Credential Guard 顯示為正在執行。

   下圖顯示 VBS 正在執行，如前所述：



使用介面 VPC 端點存取 Amazon EC2

您可在 VPC 的資源與 Amazon EC2 API 之間建立私有連線，藉此改善 VPC 的安全狀態。您可以像在 VPC 中一樣存取 Amazon EC2 API，無需使用網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線。VPC 中的 EC2 執行個體無需公有 IP 地址，即可存取 Amazon EC2 API。

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

目錄

- 建立介面 VPC 端點
- 建立端點政策

建立介面 VPC 端點

使用以下服務名稱為 Amazon EC2 建立介面端點：

- com.amazonaws.***region***.ec2 – 為 Amazon EC2 API 動作建立端點。

如需詳細資訊，請參閱《 指南》中的 AWS 服務 使用介面 VPC 端點存取 。 AWS PrivateLink

建立端點政策

端點政策為 IAM 資源，您可將其連接至您的介面端點。預設端點政策可允許透過介面端點完整存取 Amazon EC2 API。若要控制 VPC 對 Amazon EC2 API 的存取權限，請將自訂端點政策連接至介面端點。

端點政策會指定以下資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

> **Important**
>
> 當非預設政策套用至 Amazon EC2 的介面 VPC 端點時，某些失敗的 API 請求`RequestLimitExceeded`可能不會記錄到 AWS CloudTrail 或 Amazon CloudWatch。

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

下列範例顯示 VPC 端點政策，該政策拒絕建立未加密磁碟區或啟動具有未加密磁碟區之執行個體的許可。範例政策也會授與執行所有其他 Amazon EC2 動作的許可。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Action": "ec2:*",
```

```
        "Effect": "Allow",
        "Resource": "*",
        "Principal": "*"
    },
    {
        "Action": [
            "ec2:CreateVolume"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    },
    {
        "Action": [
            "ec2:RunInstances"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    }]
}
```

Amazon EC2 執行個體的儲存選項

Amazon EC2 針對您的執行個體，提供彈性、具成本效益且易於使用的資料儲存選項。每種選項都有獨特的效能與耐用性組合。這些儲存選項可獨立或組合使用，因應您的要求。

區塊儲存

- Amazon EBS – Amazon EBS 提供可從執行個體連接和分離的耐久區塊層級儲存體磁碟區。您可以將多個 EBS 磁碟區連接至同一個執行個體。EBS 磁碟區的存留與其相關執行個體的生命週期無關。您可以加密 EBS 磁碟區。若要保留資料的備份副本，您可以從 EBS 磁碟區建立快照。快照會存放在 Amazon S3。您可以從快照建立 EBS 磁碟區。
- EC 2 執行個體的執行個體儲存體暫時區塊儲存 – 執行個體儲存體會提供執行個體的暫時區塊層級儲存。執行個體儲存體磁碟區的數量、大小和類型取決於執行個體類型和執行個體大小。執行個體存放區磁碟區上的資料只會在相關聯執行個體的壽命期間持續保留；如果您讓執行個體停止、休眠或終止，則會遺失執行個體存放區磁碟區上的任何資料。

物件儲存體

- Amazon S3 – Amazon S3 會提供對可靠且價廉之資料儲存體基礎設施的存取。其設計可以簡化 Web 規模運算，讓您存放和擷取任意數量的資料，且隨時可從 Amazon EC2 或 Web 上的任意位置執行。例如，您可以使用 Amazon S3 來存放資料和應用程式的備份複本。Amazon EC2 使用 Amazon S3 來存放 EBS 快照和 Amazon S3 支援的 AMI。

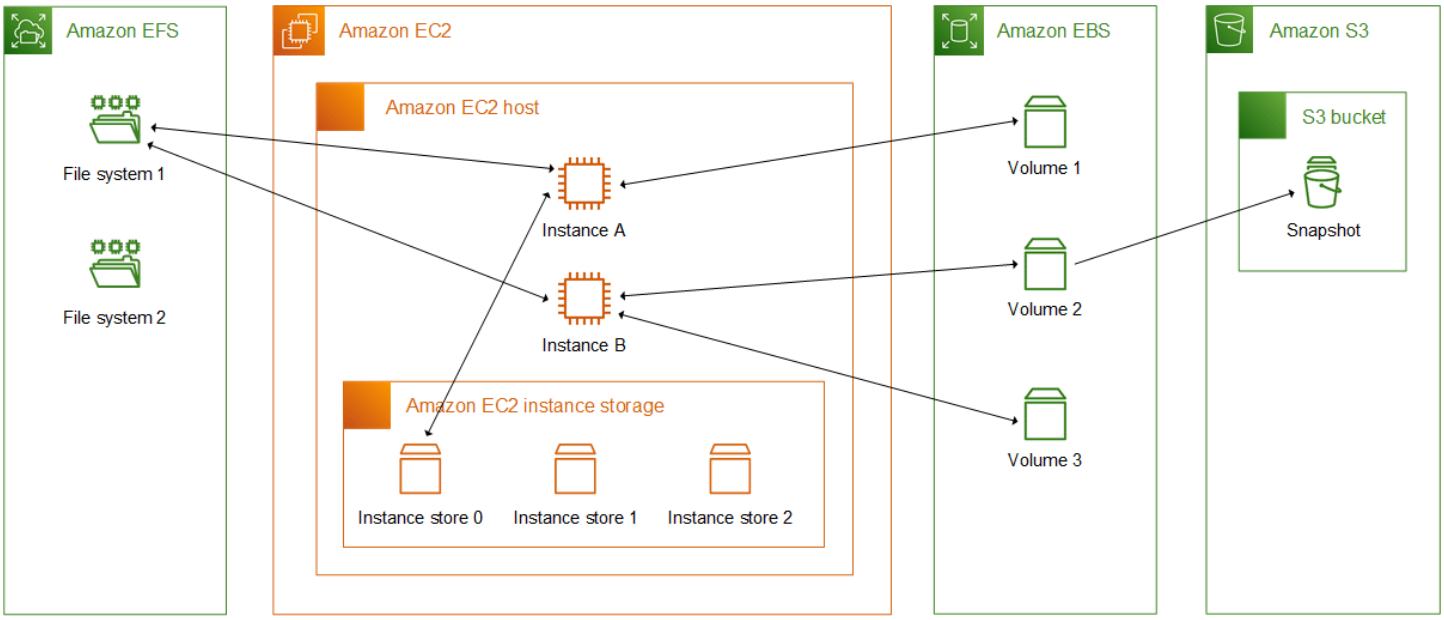
檔案儲存體

- Amazon EFS (僅限 Linux 執行個體) – Amazon EFS 提供可擴展的檔案儲存體，以搭配 Amazon EC2 使用。您可以建立 EFS 檔案系統，並設定執行個體來掛載檔案系統。您可以使用 EFS 檔案系統做為多個執行個體上執行之工作負載和應用程式的共同資料來源。
- Amazon FSx – 透過 Amazon FSx，您可以在雲端啟動、執行和擴展功能豐富的高效能檔案系統。Amazon FSx 是一項全受管服務，支援各種工作負載。您可以在以下廣泛使用的檔案系統之間進行選擇：Lustre、NetApp ONTAP、OpenZFS 和 Windows File Server。

檔案快取

- 搭配 Amazon EC2 執行個體一起使用 Amazon File Cache — Amazon File Cache 在 上提供暫時性、高效能的快取， AWS 以處理檔案資料。快取會提供對 Amazon EC2 上運算工作負載的讀取和寫入資料存取，其延遲低於一毫秒，輸送量高達數百 GB/秒，且 IOPS 高達數百萬。

下圖顯示這些儲存體選項和執行個體之間的關係。



AWS 儲存體定價

開啟AWS 定價，捲動至 AWS 產品的定價，然後選取儲存。選擇儲存產品，開啟其定價頁面。

Amazon EC2 執行個體的 Amazon EBS 持久性區塊儲存

Amazon Elastic Block Store (Amazon EBS) 會提供可擴展且高效能的區塊儲存資源，這些資源可與 Amazon EC2 執行個體搭配使用。您可以使用 Amazon EBS 來建立和管理以下區塊儲存資源：

- Amazon EBS 磁碟區 – 這些是您連接至 Amazon EC2 執行個體的儲存磁碟區。在將磁碟區連接至執行個體後，您可採用與使用區塊儲存相同的方式使用它。執行個體可與此磁碟區互動，就像與本機磁碟機互動一樣。
- Amazon EBS 快照 – 這些是 Amazon EBS 磁碟區的時間點備份，會與磁碟區本身個別保存。您可以建立快照來備份 Amazon EBS 磁碟區上的資料。然後，您隨時可從這些快照還原新磁碟區。

您可以在啟動期間建立 EBS 磁碟區並將其連接至執行個體，且可在啟動之後隨時建立 EBS 磁碟區並連接至執行個體。您也可以增加 EBS 磁碟區的大小或效能，且無需分離磁碟區或重新啟動執行個體。

建立之後，您隨時可從 EBS 磁碟區中建立 EBS 快照。您可以使用 EBS 快照來備份儲存在磁碟區上的資料。然後，您可以使用這些快照來立即還原磁碟區，或跨 AWS 帳戶區域 AWS 或可用區域遷移資料。您可以使用 Amazon Data Lifecycle Manager 或 AWS Backup 自動建立、保留和刪除 EBS 快照。

受管 EBS 磁碟區由服務供應商管理，例如 Amazon EKS Auto Mode。您無法直接修改受管 EBS 磁碟區的設定。受管 EBS 磁碟區會由受管欄位中的 true 值識別。如需詳細資訊，請參閱Amazon EC2 受管執行個體。

如需有關使用磁碟區和快照的詳細資訊，請參閱 Amazon EBS 使用者指南。

Amazon EC2 執行個體的 Amazon EBS 磁碟區限制

可連接到執行個體的 Amazon EBS 磁碟區數目上限，取決於執行個體類型和執行個體大小。如果超過執行個體的磁碟區連接限制，連接請求會失敗並顯示 `AttachmentLimitExceeded` 錯誤。

考慮要連接至執行個體的磁碟區數目時，您應考量是否需要提高 I/O 頻寬或提高儲存體容量。

頻寬和容量

針對一致且可預期的頻寬使用案例，請使用 Amazon EBS 最佳化型執行個體搭配一般用途 SSD 磁碟區或 佈建 IOPS SSD 磁碟區。如需獲得最大效能，請比對您為磁碟區所佈建的 IOPS 與適用於執行個體類型的頻寬。

針對 RAID 組態，您可能會發現大於 8 個磁碟區的陣列會因提高 I/O 額外負荷而降低效能。測試個別應用程式效能，並視需要調整。

內容

- 建置於 Nitro System 上的執行個體磁碟區限制
  - 專用 EBS 磁碟區限制
  - 共用 EBS 磁碟區限制
- Xen 型執行個體的磁碟區限制
  - Linux 執行個體
  - Windows 執行個體

建置於 Nitro System 上的執行個體磁碟區限制

在 Nitro 系統上建置之執行個體的磁碟區限制，將依執行個體類型而定。部分 Nitro 執行個體類型具有專用的 EBS 磁碟區限制，但大多數類型具有共用磁碟區限制。若要檢視每個執行個體類型的磁碟區連接限制，請參閱下列內容：

- Amazon EBS 規格 – 一般用途

- Amazon EBS 規格 – 運算最佳化
- Amazon EBS 規格 – 記憶體最佳化
- Amazon EBS 規格 – 儲存最佳化
- Amazon EBS 規格 – 加速運算
- Amazon EBS 規格 – 高效能運算
- Amazon EBS 規格 – 上一代

專用 EBS 磁碟區限制

下列 Nitro 執行個體類型的專用 EBS 磁碟區限制不盡相同，實際情況將依執行個體的大小而定。此限制不會與其他裝置連接共用。換句話說，您可以連接任意數量的 EBS 磁碟區，最多不超過磁碟區連接限制，而不論連接的裝置 (例如 NVMe 執行個體儲存體磁碟區和網路介面) 數量為何。

- 一般用途：M7a | M7i | M7i-flex | M8a | M8azn | M8g | M8gb | M8gd | M8gn | M8i | M8id | M8i-flex | M8in | M8idn | M8ine | M8ib | M8idb | M9g | M9gd
- 運算最佳化：C7a | C7i | C7i-flex | C8a | C8g | C8gb | C8gd | C8gn | C8i | C8id | C8i-flex | C8in | C8ine | C8ib | C9g | C9gd
- 記憶體最佳化：R7a | R7i | R7iz | R8a | R8g | R8gb R8gd | R8gn | R8i | R8id | R8i-flex | R8in | R8idn | R8ib | R8idb | R9g | R9gd | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X8g | X8aedz | X8ih-32tb
- 儲存最佳化：I7i | I7ie | I8g | I8ge
- 加速運算：F2 | G6 | G6e | G6f | Gr6 | Gr6f | G7 | G7e | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6-B300 | P6e-GB200 | Trn2 | Trn2u
- 高效能運算：Hpc7a | Hpc8a

共用 EBS 磁碟區限制

其餘 Nitro 執行個體類型 (未列在 專用 EBS 磁碟區限制 中) 都有磁碟區連接限制，會在 Amazon EBS 磁碟區、網路介面和 NVMe 執行個體儲存體磁碟區之間共用該限制。您可以連接任意數量的 Amazon EBS 磁碟區 (最多不超過該限制)，減去連接網路介面和 NVMe 執行個體儲存體磁碟區的數量。請記住，每個執行個體必須至少有一個網路介面，而且會在啟動時自動連接 NVMe 執行個體儲存體磁碟區。

大多數的 Nitro 執行個體最多可支援 28個附件。以下範例會示範如何計算您可連接的 EBS 磁碟區數量。

範例

- 若使用僅具有主要網路介面的 `m5.xlarge` 執行個體，您可連接 27個 EBS 磁碟區。

  28 個磁碟區 - 1 個網路介面 = 27
- 若使用具有兩個其他網路介面的 `m5.xlarge` 執行個體，您可連接 25個 EBS 磁碟區。

  28 個磁碟區 - 3 個網路介面 = 25
- 若使用具有兩個其他網路介面的 `m5d.xlarge` 執行個體，您可連接 24個 EBS 磁碟區。

  28 個磁碟區 - 3 個網路介面 - 1 個 NVMe 執行個體儲存體磁碟區 = 24

Xen 型執行個體的磁碟區限制

基於 Xen 的執行個體 (例如 T2) 的磁碟區限制取決於作業系統。

如需詳細資訊，請參閱Xen 型執行個體。

Linux 執行個體

將 40 個以上的磁碟區連接到 Xen 型 Linux 執行個體，可能會導致開機失敗。此數量包括根磁碟區，以及任何已連接的執行個體儲存體磁碟區和 Amazon EBS 磁碟區。

如果具有大量磁碟區的執行個體發生開機問題，請停止執行個體、分離啟動程序非必要的所有磁碟區、啟動執行個體，然後在執行個體執行之後重新連接磁碟區。

> **Important**
>
> Xen 型 Linux 執行個體最佳情況可支援連接超過 40 個磁碟區，但不予保證。

Windows 執行個體

下表根據使用的驅動程式，顯示 Xen 型 Windows 執行個體的磁碟區限制。這些數目包括根磁碟區，以及任何已連接的執行個體儲存體磁碟區和 Amazon EBS 磁碟區。

驅動程式	磁碟區限制
AWS PV	26

驅動程式	磁碟區限制
Citrix PV	26
Red Hat PV	17

我們建議您不要將超過 26 個磁碟區連接至具有 AWS PV 或 Citrix PV 驅動程式的 Xen 型 Windows 執行個體，因為這可能會導致效能問題。若要判定執行個體使用哪些 PV 驅動程式，或將 Windows 執行個體從 Red Hat 升級至 Citrix PV 驅動程式，請參閱the section called “升級 PV 驅動程式”。

> **Important**
>
> Xen 型 Windows 執行個體最佳情況可支援連接超過以下數量的磁碟區，但不予保證。

如需裝置名稱如何與磁碟區相關的詳細資訊，請參閱如何為 Amazon EC2 Windows 執行個體連接和映射磁碟區。

EBS 卡

大多數執行個體類型都支援一張 EBS 卡。支援多個 EBS 卡的執行個體類型可在 EBS 輸送量和 IOPS 中提供更高的 EBS 最佳化效能。您的 Amazon EC2 執行個體最大效能會平均分散到每個 EBS 卡。例如，在支援 2 張 EBS 卡`1,000,000`的總 IOPS 的 EC2 執行個體上，每個 EBS 卡最多可支援 `500,000` IOPS。如需 Amazon EC2 執行個體支援之 Amazon EBS 效能的相關資訊，請參閱 Amazon EBS 最佳化的執行個體類型。

當您將 Amazon EBS 磁碟區連接至支援多個 EBS 卡的執行個體時，您可以透過指定 EBS 卡索引來選取磁碟區的 EBS 卡。根磁碟區必須連接到 EBS 卡索引 0。

具有多個 EBS 卡的執行個體類型

下列執行個體類型支援多個 EBS 卡。如需執行個體類型支援的 Amazon EBS 磁碟區數量的相關資訊，請參閱 Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。

執行個體類型	EBS 卡數量
一般用途	
m8gb.48xlarge	2

執行個體類型	EBS 卡數量
m8gb.metal-48xl	2
m8gn.48xlarge	2
m8gn.metal-48xl	2
m8in.96xlarge	2
m8in.metal-96xl	2
m8idn.96xlarge	2
m8idn.metal-96xl	2
m8ib.96xlarge	2
m8ib.metal-96xl	2
m8idb.96xlarge	2
m8idb.metal-96xl	2
最佳化運算	
c8gb.48xlarge	2
c8gb.metal-48xl	2
c8gn.48xlarge	2
c8gn.metal-48xl	2
c8in.96xlarge	2
c8in.metal-96xl	2
c8ib.96xlarge	2
c8ib.metal-96xl	2

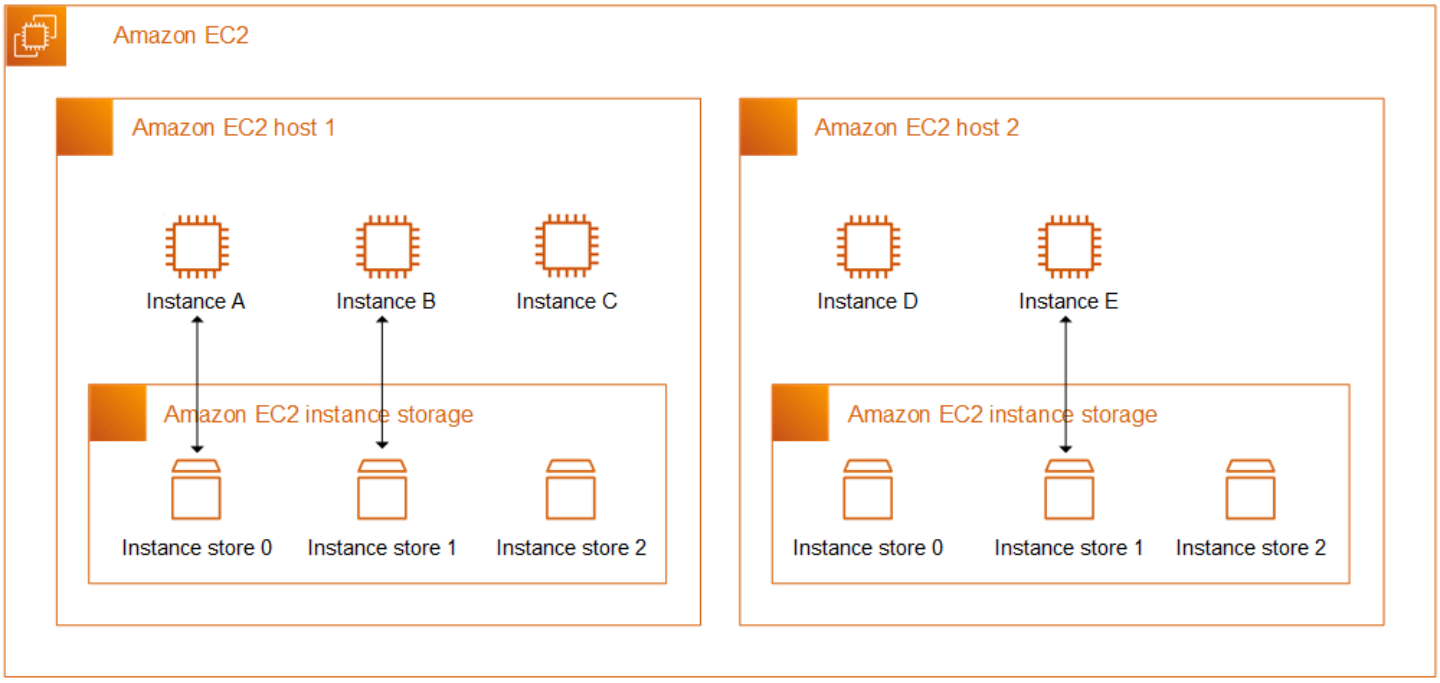
執行個體類型	EBS 卡數量
最佳化記憶體	
r8gb.48xlarge	2
r8gb.metal-48xl	2
r8gn.48xlarge	2
r8gn.metal-48xl	2
r8in.96xlarge	2
r8in.metal-96xl	2
r8idn.96xlarge	2
r8idn.metal-96xl	2
r8ib.96xlarge	2
r8ib.metal-96xl	2
r8idb.96xlarge	2
r8idb.metal-96xl	2
加速運算	
g7.48xlarge	2

EC 2 執行個體的執行個體儲存體暫時區塊儲存

執行個體儲存體為 EC2 執行個體提供暫時的區塊層級儲存空間。這個儲存空間是由實際連接到主機電腦之磁碟所提供。執行個體儲存體非常適合暫時儲存不斷變更的資訊，例如緩衝區、快取、臨時資料及其他暫存的內容。也可用來存放複製到整個執行個體機群的暫時性資料，例如網頁伺服器的負載平衡集區。

執行個體存放區包含顯示為區塊型儲存設備的一個或多個執行個體存放磁碟區。執行個體儲存體的大小和可用裝置的數量，會隨執行個體類型和執行個體大小而有不同。例如，並非所有執行個體類型都會提供執行個體儲存體磁碟區。如需詳細資訊，請參閱EC2 執行個體的執行個體儲存體磁碟區限制。

將根據從 ephemeral0 至 ephemeral23 的順序，針對執行個體儲存體磁碟區的虛擬裝置給予虛擬裝置名稱。例如，針對支援一個執行個體儲存體磁碟區的執行個體類型，一個磁碟區的虛擬裝置名稱為 ephemeral0。針對支援四個執行個體儲存體磁碟區的執行個體類型，四個磁碟區的虛擬裝置名稱如下所示：ephemeral0、ephemeral1、ephemeral2 和 ephemeral3。



執行個體儲存體定價

無需額外付費即可使用為執行個體提供的執行個體儲存體磁碟區。執行個體的使用成本包含執行個體儲存體磁碟區。

目錄

- Amazon EC2 執行個體儲存體磁碟區的資料持久性
- EC2 執行個體的執行個體儲存體磁碟區限制
- EC2 執行個體的 SSD 執行個體儲存體磁碟區
- 將執行個體儲存體磁碟區新增至 EC2 執行個體
- 啟用 M1 和 C1 EC2 執行個體的執行個體儲存體交換磁碟區
- 在 EC2 執行個體上初始化執行個體儲存體磁碟區
- Amazon EC2 執行個體儲存體磁碟區的詳細效能統計資料

Amazon EC2 執行個體儲存體磁碟區的資料持久性

執行個體儲存體磁碟區只會在執行個體啟動時連接。啟動後即無法連接執行個體儲存體磁碟區。您不能將執行個體儲存體磁碟區與某個執行個體中斷連接，然後再連接到不同的執行個體。

執行個體儲存體磁碟區只存在於所連接執行個體的存留期。您無法將執行個體儲存體磁碟區設定為保留超過其關聯執行個體的存留期。

即使將執行個體重新開機，執行個體儲存體磁碟區上的資料仍會存在。但如果停止、休眠或終止執行個體，則資料不會存在。執行個體停止、休眠或終止時，會以密碼編譯方式清除執行個體儲存體磁碟區的每個區塊。

因此，如果是重要的長期資料，請不要只使用執行個體儲存體磁碟區。如果您需要將儲存在執行個體儲存體磁碟區上的資料保留超過執行個體的存留期，必須手動將該資料複製到更持久的儲存體，例如 Amazon EBS 磁碟區、Amazon S3 儲存貯體或 Amazon EFS 檔案系統。

部分事件可能會導致您的資料在執行個體的整個存留期中未持續存在。下表指出在特定事件期間，虛擬化和裸機執行個體的執行個體儲存體磁碟區上的資料是否持續存在。

事件	您的資料會發生什麼事？
使用者啟動的執行個體生命週期事件	
執行個體重新開機	The data persists
執行個體停止	The data does not persist
執行個體休眠	The data does not persist
執行個體終止	The data does not persist
執行個體類型變更	The data does not persist *
從執行個體建立 EBS 支援的 AMI	The data does not persist in the created AMI **
從執行個體中建立基於 Amazon S3 的 AMI (Linux instances)	The data persists in the AMI bundle uploaded to Amazon S3 ***
使用者啟動的作業系統事件	
A shutdown is initiated	The data does not persist †

事件	您的資料會發生什麼事？
A restart is initiated	The data persists
AWS 排程事件	
執行個體停止	The data does not persist
重新啟動執行個體	The data persists
系統重新開機	The data persists
執行個體淘汰	The data does not persist
意外事件	
簡化的自動復原	The data does not persist
基於 CloudWatch 動作的復原	The data does not persist
The underlying disk fails	The data on the failed disk does not persist
Power failure	The data persists upon reboot

* 如果新執行個體類型支援執行個體儲存體，執行個體會具有新執行個體類型支援的執行個體儲存體磁碟區數量，但資料不會傳輸至新執行個體。如果新執行個體類型不支援執行個體儲存體，則執行個體不會取得執行個體儲存體磁碟區。

** 資料不包含在 EBS 支援的 AMI 中，且不包含在連接至該 AMI 啟動之執行個體的執行個體儲存體磁碟區中。

*** 資料包含在上傳至 Amazon S3 的 AMI 套件中。從該 AMI 啟動執行個體時，執行個體會取得 AMI 中綁定的執行個體儲存體，以及建立 AMI 時包含的資料。

† 終止保護和停止保護不會防止執行個體因透過執行個體上的作業系統關機，而停止或終止。儲存在執行個體儲存體磁碟區上的資料不會保留在執行個體停止和終止事件中。

EC2 執行個體的執行個體儲存體磁碟區限制

執行個體儲存體磁碟區的數量、大小和類型取決於執行個體類型。有些執行個體類型，例如 C8i、M8i 和 R8i，不支援執行個體存放磁碟區，而其他執行個體類型，例如 C8id、M8id 和 R8id，則支援執行個

體存放磁碟區。您無法將超過執行個體類型所支援的執行個體儲存體磁碟區數量，連接至執行個體。對於支援執行個體儲存體磁碟區的執行個體類型，執行個體儲存體磁碟區的數量和大小會依執行個體大小而異。例如，`r8id.large`支援 1 x 118 GB 執行個體存放區磁碟區，而 `r8id.32xlarge`支援 2 x 3800 GB 執行個體存放區磁碟區。

對於具有 NVMe 執行個體儲存體磁碟區的執行個體類型，所有支援的執行個體儲存體磁碟區都會在啟動時自動連接至執行個體。對於具有非 NVMe 執行個體儲存體磁碟區的執行個體類型 (如 C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e)，您必須為要在啟動時連接的執行個體儲存體磁碟區，手動指定區塊型儲存設備映射。執行個體啟動後，您必須先格式化並掛載連接的執行個體儲存體磁碟區，才能使用它們。啟動執行個體之後，即無法連接執行個體儲存體磁碟區。

部分執行個體類型使用 NVMe 或 SATA 型固態硬碟 (SSD)，其他執行個體類型則使用 SATA 型硬碟 (HDD)。SSD 提供極低延遲的高隨機 I/O 效能，但您不希望資料在執行個體終止時仍持續存在，或是可以善用容錯架構。如需詳細資訊，請參閱EC2 執行個體的 SSD 執行個體儲存體磁碟區。

NVMe 執行個體存放區磁碟區和某些 HDD 執行個體存放區磁碟區上的資料會靜態加密。如需詳細資訊，請參閱Amazon EC2 中的資料保護。

可用執行個體儲存體磁碟區

「Amazon EC2 執行個體類型指南」提供在每種支援的執行個體類型上，可使用執行個體儲存體磁碟區的數量、尺寸、類型和效能最佳化配置。如需詳細資訊，請參閱下列內容：

- 執行個體儲存體規格 – 一般用途
- 執行個體儲存體規格 – 運算最佳化
- 執行個體儲存體規格 – 記憶體最佳化
- 執行個體儲存體規格 – 儲存最佳化
- 執行個體儲存體規格 – 加速運算
- 執行個體儲存體規格 – 高效能運算
- 執行個體儲存體規格 – 上一代

Console

若要擷取執行個體儲存體磁碟區資訊

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instance Types (執行個體類型)。

3. 新增篩選條件 Local instance storage = true。儲存體資料欄顯示執行個體類型的執行個體儲存體總大小。
4. （選用）選擇偏好設定圖示，然後開啟儲存磁碟計數。此資料欄顯示執行個體儲存體磁碟區的數量。
5. (選用) 新增篩選條件以進一步限定到感興趣的特定執行個體類型。

AWS CLI

若要擷取執行個體儲存體磁碟區資訊

使用 describe-instance-types 命令。下列範例顯示具有執行個體儲存體磁碟區之 R8i 執行個體系列中每個執行個體類型的執行個體儲存體總大小。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r8i*" "Name=instance-storage-
supported,Values=true" \
    --query 'sort_by(InstanceTypes, &InstanceStorageInfo.TotalSizeInGB)[].
{InstanceType:InstanceType,TotalSizeInGB:InstanceStorageInfo.TotalSizeInGB}' \
    --output table
```

以下為範例輸出。

```
-------------------------------------
|        DescribeInstanceTypes       |
+------------------+-----------------+
|   InstanceType   |  TotalSizeInGB  |
+------------------+-----------------+
|  r8id.large      |  118            |
|  r8id.xlarge     |  237            |
|  r8id.2xlarge    |  474            |
|  r8id.4xlarge    |  950            |
|  r8id.8xlarge    |  1900           |
|  r8id.12xlarge   |  2850           |
|  r8id.16xlarge   |  3800           |
|  r8id.24xlarge   |  5700           |
|  r8id.32xlarge   |  7600           |
|  r8id.48xlarge   |  11400          |
|  r8id.metal-48xl |  11400          |
|  r8id.96xlarge   |  22800          |
|  r8id.metal-96xl |  22800          |
```

```
+------------------+-----------------+
```

若要取得執行個體類型的完整執行個體儲存體詳細資訊

使用 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r8id.32xlarge" \
    --query 'InstanceTypes[0].InstanceStorageInfo' \
    --output json
```

範例輸出顯示此執行個體類型有兩個 3800 GB NVMe SSD 磁碟區，總共 7600 GB 的執行個體儲存體。

```
{
    "TotalSizeInGB": 7600,
    "Disks": [
        {
            "SizeInGB": 3800,
            "Count": 2,
            "Type": "ssd"
        }
    ],
    "NvmeSupport": "required",
    "EncryptionSupport": "required"
}
```

PowerShell

若要擷取執行個體儲存體磁碟區資訊

使用 Get-EC2InstanceType cmdlet。下列範例顯示具有執行個體儲存體磁碟區之 R8i 執行個體系列中每個執行個體類型的執行個體儲存體總大小。

```
(Get-EC2InstanceType -Filter `
    @{Name="instance-type"; Values="r8i*"},
    @{Name="instance-storage-supported"; Values="true"}) |
    Sort-Object {$_.InstanceStorageInfo.TotalSizeInGB} |
    Format-Table InstanceType,

 @{Name="Disks.SizeInGB";Expression={$_.InstanceStorageInfo.Disks[0].SizeInGB}},
        @{Name="Disks.Count";Expression={$_.InstanceStorageInfo.Disks[0].Count}},
```

```
        @{Name="TotalSizeInGB";Expression={$_.InstanceStorageInfo.TotalSizeInGB}}
```

以下為範例輸出。

```
InstanceType    Disks.SizeInGB Disks.Count TotalSizeInGB
------------    -------------- ----------- -------------
r8id.large                 118           1           118
r8id.xlarge                237           1           237
r8id.2xlarge               474           1           474
r8id.4xlarge               950           1           950
r8id.8xlarge              1900           1          1900
r8id.12xlarge             2850           1          2850
r8id.16xlarge             3800           1          3800
r8id.24xlarge             2850           2          5700
r8id.32xlarge             3800           2          7600
r8id.48xlarge             3800           3         11400
r8id.metal-48xl           3800           3         11400
r8id.96xlarge             3800           6         22800
r8id.metal-96xl           3800           6         22800
```

若要取得執行個體類型的完整執行個體儲存體詳細資訊

使用 Get-EC2InstanceType cmdlet。

```
(Get-EC2InstanceType `
    -Filter @{Name="instance-type"; Values="r8id.32xlarge"}).InstanceStorageInfo |
    Format-List *,
        @{Name="Disks.Count";Expression={$_.Disks[0].Count}},
        @{Name="Disks.SizeInGB";Expression={$_.Disks[0].SizeInGB}},
        @{Name="Disks.Type";Expression={$_.Disks[0].Type.Value}}
```

範例輸出顯示此執行個體類型有兩個 3800 GB NVMe SSD 磁碟區，總共 7600 GB 的執行個體儲存體。

```
Disks             : {Amazon.EC2.Model.DiskInfo}
EncryptionSupport : required
NvmeSupport       : required
TotalSizeInGB     : 7600
Disks.Count       : 2
Disks.SizeInGB    : 3800
Disks.Type        : ssd
```

EC2 執行個體的 SSD 執行個體儲存體磁碟區

如同其他的執行個體存放磁碟區，您必須在啟動執行個體時，對應執行個體的 SSD 執行個體存放磁碟區。SSD 執行個體磁碟區上的資料，只會存在於其相關執行個體的生命週期期間。如需詳細資訊，請參閱 將執行個體儲存體磁碟區新增至 EC2 執行個體。

NVMe SSD 磁碟區

某些執行個體提供非揮發性記憶體儲存裝置 (NVMe) 固態硬碟 (SSD) 執行個體存放區磁碟區。關於每種執行個體類型所支援的執行個體存放磁碟區類型，詳細資訊請參閱EC2 執行個體的執行個體儲存體磁碟區限制。

NVMe 執行個體儲存體上的資料會以執行個體上的硬體模組中實作的 XTS-AES-256 區塊編碼器來加密。加密金鑰是以硬體模組來產生，且對每個 NVMe 執行個體儲存體設備而言是唯一的。所有加密金鑰會在執行個體停止或終止時銷毀，且無法復原。您無法停用此加密，也無法提供您自己的加密金鑰。

Linux 執行個體

若要存取 NVMe 磁碟區，必須安裝 NVMe 驅動程式。下列 AMI 符合此需求：

- AL2023
- Amazon Linux 2
- Amazon Linux AMI 2018.03 及更新版本
- Ubuntu 14.04 或更新版本 (帶 `linux-aws` 核心)

> **Note**
>
> AWS Graviton 型執行個體類型需要具有核心`linux-aws`的 Ubuntu 18.04 或更新版本

- Red Hat Enterprise Linux 7.4 或更新版本
- SUSE Linux Enterprise Server 12 SP2 或更新版本
- CentOS 7.4.1708 或更新版本
- FreeBSD 11.1 或更新版本
- Debian GNU/Linux 9 或更新版本

- Bottlerocket

在連線到執行個體之後，您可以使用 lspci 指令，來列出 NVMe 裝置。下列是 i3.8xlarge 執行個體的輸出範例，此執行個體支援四種 NVMe 裝置。

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

如果使用支援的作業系統，但是未顯示 NVMe 裝置，請利用下列命令來確認已載入 NVMe 模組。

- Amazon Linux, Amazon Linux 2, Ubuntu 14/16, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS 7

  ```
  $ lsmod | grep nvme
  nvme           48813  0
  ```

- Ubuntu 18

  ```
  $ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
  s/nvme/host/nvme-core.ko
  kernel/drivers/nvme/host/nvme.ko
  kernel/drivers/nvmem/nvmem_core.ko
  ```

NVMe 磁碟區符合 NVMe 1.0e 規格。您可以對 NVMe 磁碟區使用 NVMe 指令。如果使用 Amazon Linux，您可以使用 nvme-cli 命令，從儲存庫安裝 yum install 套件。使用其他支援的 Linux 版本時，如果映像中未提供，您可以下載 nvme-cli 套件。

Windows 執行個體

下列作業系統的最新 AWS Windows AMIs 包含 AWS 用來與公開為 NVMe 區塊型儲存設備之 SSD 執行個體存放磁碟區互動的 NVMe 驅動程式，以提供更好的效能：

- Windows Server 2025

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

連線到執行個體之後，您可以在 Disk Manager 中，確認是否顯示 NVMe 磁碟區。在任務列上開啟 Windows 標誌的內容 (按右鍵) 選單，然後選擇 Disk Management (磁碟管理)。

Amazon 提供的 AWS Windows AMIs 包含 AWS NVMe 驅動程式。如果您不是使用最新的 AWS Windows AMIs，則可以安裝 current AWS NVMe 驅動程式。

非 NVMe SSD 磁碟區

下列執行個體支援使用非 NVMe SSD 的執行個體儲存體磁碟區，以提供高度的隨機 I/O 效能：C3、I2、M3、R3 和 X1。關於每種執行個體類型所支援的執行個體存放磁碟區，詳細資訊請參閱EC2 執行個體的執行個體儲存體磁碟區限制。

SSD 型執行個體儲存體磁碟區 I/O 效能

隨著執行個體的 SSD 執行個體存放區磁碟區越來越滿，可達到的寫入 IOPS 次數將下降。這是因為 SSD 控制器必須執行額外的工作，去尋找可用的空間、重寫現有資料，以及清除未使用的空間以供重寫。這個收集廢棄項目的程序會造成 SSD 的內部寫入放大，也就是 SSD 寫入操作與使用者寫入操作的比例放大。假如寫入操作不是 4,096 位元組的倍數或未符合 4,096 位元組的邊界，效能下降的幅度會更明顯。因為當您寫入少量位元組或未符合的位元組時，SSD 控制器必須讀取週圍的資料並將結果存放到新的位置，此模式會造成寫入放大大幅增加，提高延遲，因而使 I/O 效能巨幅下降。

SSD 控制器利用多項策略來減少寫入放大的影響。其中一項策略，就是在 SSD 執行個體儲存體內保留空間，讓控制器能更有效管理寫入操作可用的空間。此策略稱為過度佈建。提供給執行個體的 SSD 執行個體儲存體磁碟區不會保留任何過度佈建的空間。為了減少寫入放大，建議讓 10% 的磁碟區保留為未分割，如此 SSD 控制器便不會將其用於過度佈建。這樣做雖然會減少可用的儲存空間，但可提高效能，即使磁碟即將用完所有容量。

使用支援 TRIM 的執行個體儲存體磁碟區時，當不再需要已寫入的資料時，可以利用 TRIM 指令來通知 SSD 控制器。這會讓控制器有更多的可用空間能夠使用，可減少寫入放大現象，並提升效能。如需詳細資訊，請參閱執行個體存放區磁碟區 TRIM 支援。

執行個體存放區磁碟區 TRIM 支援

某些執行個體類型支援具有 TRIM 的 SSD 磁碟區。如需詳細資訊，請參閱EC2 執行個體的執行個體儲存體磁碟區限制。

> **Note**
>
> （僅限 Windows 執行個體） 執行 Windows Server 2012 R2 的執行個體支援自 AWS PV 驅動程式 7.3.0 版起的 TRIM。執行較早版本 Windows Server 的執行個體不支援 TRIM。

支援 TRIM 的執行個體存放磁碟區，會在分配給執行個體之前完全停用刪除通知 (執行 trim 指令)。在執行個體啟動時，這些磁碟區不會格式化為任何檔案系統，因此您必須先將這些磁碟區格式化之後，才能掛載和使用這些磁碟區。若要更快速地存取這些磁碟區，應在將這些磁碟區格式化時跳過 TRIM 操作。

(Windows 執行個體) 若要在初始格式化期間暫時停用 TRIM 支援，請使用 `fsutil behavior set DisableDeleteNotify 1` 命令。格式化完成後，請使用 `fsutil behavior set DisableDeleteNotify 0` 重新啟用 TRIM 支援。

使用支援 TRIM 的執行個體存放磁碟區時，當不再需要已寫入的資料時，可以利用 TRIM 指令來通知 SSD 控制器。這會讓控制器有更多的可用空間能夠使用，可減少寫入放大現象，並提升效能。在 Linux 執行個體上，使用 `fstrim` 命令來啟用定期 TRIM。在 Windows 執行個體上，使用 `fsutil behavior set DisableDeleteNotify 0`命令來確保在正常操作期間啟用 TRIM 支援。

將執行個體儲存體磁碟區新增至 EC2 執行個體

對於具有 NVMe 執行個體儲存體磁碟區的執行個體類型，所有支援的執行個體儲存體磁碟區都會在啟動時自動連接至執行個體。執行個體啟動時，系統會自動列舉這些執行個體儲存體磁碟區並指派裝置名稱。

對於具有非 NVMe 執行個體儲存體磁碟區的執行個體類型 (如 C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e)，您必須為要在啟動時連接的執行個體儲存體磁碟區，手動指定區塊型儲存設備映射。您可以在執行個體啟動要求或用來啟動執行個體的 AMI 中，指定區塊型裝置映射。區塊型裝置映射包含裝置名稱和對應的磁碟區。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射

> **Important**
>
> 您只能在啟動執行個體時，將執行個體儲存體磁碟區連接至執行個體。啟動執行個體之後，即無法將執行個體存放磁碟區連結到執行個體。

在啟動執行個體之後，您必須先確定執行個體的執行個體存放磁碟區已經過格式化並掛載，才能使用這些磁碟區。執行個體儲存體根磁碟區會自動掛載。

根磁碟區的考量

區塊型設備映射一律會指定執行個體的根磁碟區。根磁碟區永遠會自動掛載。

Linux 執行個體 – 根磁碟區會是 Amazon EBS 磁碟區或執行個體儲存體磁碟區。執行個體如果具有根磁碟區的執行個體存放磁碟區，則此磁碟區的大小會隨 AMI 而有不同，但其上限為 10 GB。如需詳細資訊，請參閱根磁碟區類型。

Windows 執行個體 – 根磁碟區必須為 Amazon EBS 磁碟區。根磁碟區不支援執行個體儲存體。

目錄

- 將執行個體儲存體磁碟區新增至 Amazon EC2 AMI
- 在啟動期間，將執行個體儲存體磁碟區新增至 EC2 執行個體
- 讓 EC2 執行個體上的執行個體儲存體磁碟區變成可用

將執行個體儲存體磁碟區新增至 Amazon EC2 AMI

您可以建立 AMI，此 AMI 具備包含執行個體存放磁碟區的區塊型設備映射。

如果您啟動執行個體，其使用指定執行個體儲存體磁碟區區塊型裝置映射的 AMI，支援非 NVMe 執行個體儲存體磁碟區，則執行個體包含這些執行個體儲存體磁碟區。如果 AMI 中執行個體儲存體磁碟區區塊型裝置映射的數量，超過執行個體可用的執行個體儲存體磁碟區數量，請忽略額外的執行個體儲存體磁碟區區塊型裝置映射。

如果您啟動執行個體，其使用指定執行個體儲存體磁碟區區塊型裝置映射的 AMI，支援 NVMe 執行個體儲存體磁碟區，則會忽略執行個體儲存體磁碟區區塊型裝置映射。支援 NVMe 執行個體儲存體磁碟區的執行個體會取得所有支援的執行個體儲存體磁碟區，無論執行個體啟動要求和 AMI 中指定的區塊型裝置映射為何。這些磁碟區的裝置映射取決於作業系統列舉這些磁碟區的順序。

考量事項

- 可用執行個體存放磁碟區的數量，會視執行個體類型而定。如需詳細資訊，請參閱the section called "可用執行個體儲存體磁碟區"。
- 必須為每個區塊型儲存設備指定設備名稱。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的裝置名稱。
- 啟動執行個體時，您可以略過在 AMI 區塊型設備映射中指定的非 NVMe 執行個體存放磁碟區，或新增執行個體存放磁碟區。
- 如果是 M3 執行個體，在執行個體的區塊型設備映射中指定執行個體存放區，而非 AMI。Amazon EC2 可能會忽略 AMI 中的執行個體儲存體磁碟區區塊型裝置映射。

Console

若要將執行個體儲存體磁碟區新增至基於 Amazon EBS 的 AMI

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取該執行個體。
3. 選擇 Actions (動作)、Image and templates (映像和範本)、Create image (建立映像)。
4. 在 建立影像 頁面，為您的影像輸入有意義的名稱和說明。
5. 針對每個要新增的執行個體存放區的磁碟區，選取 Add New Volume (新增新磁碟區)、從 Volume Type (磁碟區類型) 選取一個執行個體存放區的磁碟區，並從 Device (裝置) 選取一個裝置名稱。
6. 選擇 Create image (建立映像)。

AWS CLI

若要將執行個體儲存體磁碟區新增至 AMI

使用具有 `--block-device-mappings` 選項的 create-image 命令，指定基於 EBS 的 AMI 的區塊型儲存設備映射。使用具有 `--block-device-mappings` 選項的 register-image 命令，指定基於 iAmazon S3 的 AMI 的區塊型儲存設備映射。

```
--block-device-mappings file://mapping.json
```

下列區塊型儲存設備映射會新增兩個執行個體儲存體磁碟區。

```
[
```

```
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

若要將執行個體儲存體磁碟區新增至 AMI

使用具有 `-BlockDeviceMapping` 參數的 New-EC2Image cmdlet 來指定基於 EBS 的 AMI 的區塊型儲存設備映射。使用具有 `-BlockDeviceMapping` 參數的 Register-EC2Image cmdlet 來指定基於 Amazon S3 的 AMI 的區塊型儲存設備映射。

```
-BlockDeviceMapping $bdm
```

下列區塊型儲存設備映射會新增兩個執行個體儲存體磁碟區。

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

在啟動期間，將執行個體儲存體磁碟區新增至 EC2 執行個體

在您啟動具有非 NVMe 執行個體儲存體磁碟區的執行個體類型 (如 C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e) 時，您必須為要在啟動時連接的執行個體儲存體磁碟區，指定區塊型儲存設備映射。必須在執行個體啟動要求或用於啟動執行個體的 AMI 中，指定區塊型裝置映射。

如果 AMI 包含執行個體儲存體磁碟區的區塊型裝置映射，您不需要在執行個體啟動要求中指定區塊型裝置映射，除非您需要的執行個體儲存體磁碟區數量超過 AMI 中所包含的數量。

如果 AMI 不包括執行個體儲存體磁碟區的區塊型裝置映射，則必須在執行個體啟動請求中指定區塊型裝置映射。

對於具有 NVMe 執行個體儲存體磁碟區的執行個體類型，所有支援的執行個體儲存體磁碟區都會在啟動時自動連接至執行個體。

考量事項

- 可用執行個體存放磁碟區的數量，會視執行個體類型而定。如需詳細資訊，請參閱the section called “可用執行個體儲存體磁碟區”。
- 必須為每個區塊型儲存設備指定設備名稱。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的裝置名稱。
- 如果是 M3 執行個體，即使未在執行個體的區塊型設備映射中指定這些執行個體，您可能也會獲得執行個體存放磁碟區。

Console

若要在執行個體啟動請求中指定區塊型儲存設備映射

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在儀表板中，選擇啟動執行個體。
3. 在 Application and OS Images (應用程式和 OS 映像) 區段中，選取要使用的 AMI。
4. 在設定儲存體區段中，執行個體儲存體磁碟區區段會列出可連結至該執行個體的執行個體儲存體磁碟區。
5. 對於要連接的每個執行個體儲存體磁碟區，針對裝置名稱，請選取要使用的裝置名稱。
6. 根據需要進行剩餘的執行個體設定，然後選擇啟動執行個體。

AWS CLI

若要在執行個體啟動請求中指定區塊型儲存設備映射

請使用 `--block-device-mappings` 選項並搭配 run-instances 命令。

```
--block-device-mappings file://mapping.json
```

下列區塊型儲存設備映射會新增兩個執行個體儲存體磁碟區。

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

若要在執行個體啟動請求中指定區塊型儲存設備映射

使用 New-EC2Instance cmdlet 搭配 -BlockDeviceMapping 選項。

```
-BlockDeviceMapping $bdm
```

下列區塊型儲存設備映射會新增兩個執行個體儲存體磁碟區。

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

讓 EC2 執行個體上的執行個體儲存體磁碟區變成可用

啟動具有連接執行個體儲存體磁碟區的執行個體之後，您必須先掛接磁碟區，才能存取這些磁碟區。

Linux 執行個體

您可在啟動執行個體之後，將磁碟區格式化為自選的檔案系統。

在 Linux 上讓執行個體存放磁碟區變成可用

1. 使用 SSH 用戶端連線到執行個體。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
2. 使用 df -h 指令來檢視磁碟區是否已格式化和掛載。

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   72K  3.8G   1% /dev
tmpfs           3.8G     0  3.8G   0% /dev/shm
/dev/nvme0n1p1  7.9G  1.2G  6.6G  15% /
```

3. 使用 lsblk 來檢視是否有任何磁碟區已在啟動時映射，但尚未格式化和掛載。

```
$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1    0    8G  0 disk
##nvme0n1p1   259:2    0    8G  0 part /
##nvme0n1p128 259:3    0    1M  0 part
nvme1n1       259:0    0 69.9G  0 disk
```

4. 若要只針對映射的執行個體存放磁碟區來進行格式化和掛載，請執行下列動作：

   a. 使用 mkfs 指令來在裝置上建立檔案系統。

   ```
   $ sudo mkfs -t xfs /dev/nvme1n1
   ```

   b. 使用 mkdir 指令來建立用來掛載裝置的目錄。

   ```
   $ sudo mkdir /data
   ```

   c. 使用 mount 指令，將裝置掛載到新建立的目錄上。

   ```
   $ sudo mount /dev/nvme1n1 /data
   ```

Windows 執行個體

如果是 Windows 執行個體，系統會將執行個體儲存體磁碟區重新格式化為 NTFS 檔案系統。

您可以使用 Windows 磁碟管理，來檢視執行個體儲存體磁碟區。如需詳細資訊，請參閱列出非 NVMe 磁碟。

手動掛載執行個體存放區磁碟區

1. 選擇開始，輸入電腦管理，然後按 Enter。
2. 在左側面板中，選擇磁碟管理。
3. 如果系統提示您初始化磁碟區，請選取要初始化的磁碟區，根據您的使用案例選取所需的磁碟分割類型，然後選取 確定。
4. 在磁碟區清單中，開啟要掛載磁碟區的內容（按一下滑鼠右鍵）選單，然後選擇新增簡易磁碟區。
5. 在精靈上，選擇下一步。
6. 在「指定磁碟區大小」畫面上，選擇下一步以使用最大磁碟區大小。或者，選擇介於最小和最大磁碟空間之間的磁碟區大小。
7. 在「指派磁碟機代號或路徑」畫面上，執行下列其中一項作業，然後選擇下一步。
   - 若要使用磁碟機代號掛接磁碟區，請選擇指派下列磁碟機代號，然後選擇要使用的磁碟機代號。
   - 若要將磁碟區掛接為資料夾，請在下列空白 NTFS 資料夾中選取掛接，然後選取瀏覽以建立或選取要使用的資料夾。
   - 若要在沒有磁碟機代號或路徑的情況下掛接磁碟區，請選擇不要指派磁碟機代號或磁碟機路徑。
8. 在格式化分割區畫面上，指定是否要格式化磁碟區。如果您選擇格式化磁碟區，請選擇所需的檔案系統和單位大小，然後指定磁碟區標籤。
9. 依序選擇下一步，完成。

啟用 M1 和 C1 EC2 執行個體的執行個體儲存體交換磁碟區

> **Note**
>
> 本主題僅適用於 `c1.medium` 和 `m1.small` Linux 執行個體。

`c1.medium` 和 `m1.small` 執行個體類型的實體記憶體數量有限。因此，其會在啟動時被授予 900 MiB 交換磁碟區，作為 Linux 系統的虛擬記憶體或交換空間。當系統需要比實體配置更多的記憶體時，可以使用 Linux 的置換空間。啟用置換空間時，Linux 系統可以將不常使用的記憶體分頁，從實體記憶體換到置換空間 (專用磁區或是現有檔案系統中的調換檔)，以釋出空間，給需要高速存取的記憶體分頁使用。

> **Note**
>
> - 使用置換空間來做為記憶體分頁時，其速度和效率並不如使用 RAM。如果工作負載會定期將分頁記憶體換到置換空間中，您應考慮轉而使用較大的執行個體類型和更多的 RAM。如需詳細資訊，請參閱Amazon EC2 執行個體類型變更。
> - 雖然 Linux 核心將此置換空間視為根磁碟區上的分區，但該空間實際上為獨立的執行個體儲存體磁碟區，而不論根磁碟區是何種類型。

Amazon Linux 會自動啟用和使用此置換空間，但 AMI 可能會需要執行一些其他的步驟才能辨識和使用此置換空間。若要查看執行個體是否正在使用置換空間，您可以使用 swapon -s 命令。

```
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

上述的執行個體連結和啟用了 900 MiB 的智慧磁碟區。如果使用此指令未列出置換磁碟區，可能需要啟用該設備的置換空間。使用 lsblk 命令來檢查可用的磁碟。

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk
```

在此處，置換磁碟區 xvda3 可供執行個體使用，但是並未啟用 (請注意，MOUNTPOINT 欄位為空白)。您可以利用 swapon 命令來啟用置換磁碟區。

> **Note**
>
> 您必須在 /dev/ 命令所列出的裝置名稱前方，加上 lsblk。裝置可能需有不同的名稱，例如 sda3、sde3 或 xvde3。在下列的指令中使用系統的裝置名稱。

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

現在置換空間應該會顯示 lsblk 和 swapon -s 輸出中。

```
[ec2-user ~]$ lsblk
```

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk [SWAP]
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

您也需要編輯 /etc/fstab 檔案，讓此一置換空間在每次系統開機時自動啟用。

```
[ec2-user ~]$ sudo vim /etc/fstab
```

將下列的指令行附加到 /etc/fstab 檔案之後 (使用系統的置換裝置名稱)：

```
/dev/xvda3       none    swap    sw  0       0
```

將執行個體存放磁碟區做為置換空間使用

任何執行個體存放磁碟區皆可做為置換空間使用。例如，m3.medium 執行個體類型包含適合做為置換空間的 4 GB SSD 執行個體存放磁碟區。如果執行個體存放磁碟區的容量大上許多 (例如 350 GB)，您可以考慮將磁碟區分割出一個較小的 4 到 8 GB 置換磁區，其他的部分則做為資料磁碟區。

> **Note**
>
> 此程序僅適用於支援執行個體儲存體的執行個體類型。如需支援的執行個體類型清單，請參閱 EC2 執行個體的執行個體儲存體磁碟區限制。

1. 列出連結至執行個體的區塊型儲存設備，以取得執行個體存放磁碟區的裝置名稱。

   ```
   [ec2-user ~]$ lsblk -p
   NAME       MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   /dev/xvdb  202:16   0   4G  0 disk /media/ephemeral0
   /dev/xvda1 202:1    0   8G  0 disk /
   ```

   在此範例中，執行個體存放磁碟區為 /dev/xvdb。因為這是 Amazon Linux 執行個體，因此執行個體存放磁碟區會經過格式化並掛載於 /media/ephemeral0；不是所有的 Linux 作業系統都會自動執行這項動作。

2. (選用) 如果執行個體存放磁碟區已掛載 (會在 MOUNTPOINT 命令輸出中列出 lsblk)，請利用下列指令來卸載該磁碟區。

```
[ec2-user ~]$ sudo umount /dev/xvdb
```

3. 使用 mkswap 命令，在裝置上設定 Linux 置換區。

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swapspace version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. 啟用新的置換空間。

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. 確認新的置換空間正在使用。

```
[ec2-user ~]$ swapon -s
Filename    Type  Size Used Priority
/dev/xvdb                               partition 4188668 0 -1
```

6. 編輯 /etc/fstab 檔案，讓此一置換空間在每次系統開機時自動啟用。

```
[ec2-user ~]$ sudo vim /etc/fstab
```

如果 /etc/fstab 檔案具有 /dev/xvdb (或 /dev/sdb) 的項目，請修改為和下列行相同；如果該檔案不具有此裝置的項目，請將下列行附加於 /etc/fstab 檔案之後 (讓系統使用置換裝置的名稱)：

```
/dev/xvdb       none    swap    sw  0       0
```

> **Important**
>
> 當執行個體停止或休眠時，執行個體存放磁碟區上的資料會遺失；這包括在 Step 3 中所建立的執行個體存放區置換空間格式。如果停止和重新啟動執行個體，而此執行個體已設定使用執行個體存放區置換空間，則您必須在新的執行個體存放磁碟區上，重複執行 Step 1 到 Step 5。

在 EC2 執行個體上初始化執行個體儲存體磁碟區

由於 Amazon EC2 將磁碟虛擬化的方式，在某些執行個體存放磁碟區上，第一次寫入任意位置的動作，其執行速度會比後續的寫入動作慢。對大多數應用程式而言，在執行個體的整個生命週期內分攤此效能減損是可以接受的。不過，如果要求高度磁碟效能，建議在正式生產使用之前，藉由寫入每個磁碟位置一次，來將磁碟初始化。

> **Note**
>
> 執行個體類型具備直連式固態硬碟 (SSD) 和 TRIM 支援，可在啟動時提供最高效能，不需初始化。關於適用於每種執行個體類型的執行個體存放區，詳細資訊請參閱 EC2 執行個體的執行個體儲存體磁碟區限制。

如果需要在延遲或傳輸量方面獲得更高的彈性，建議使用 Amazon EBS。

若要將執行個體存放磁碟區初始化，請根據要初始化的存放區 (例如 dd 或 /dev/sdb)，使用下列 /dev/nvme1n1 指令來進行。

> **Note**
>
> 請務必在執行此指令之前卸載磁碟。
> 初始化作業可能需要很長的時間來完成 (如果是超大型的執行個體，約需要 8 小時)。

若要將執行個體存放磁碟區初始化，請針對 m1.large、m1.xlarge、c1.xlarge、m2.xlarge、m2.2xlarge 和 m2.4xlarge 執行個體類型使用下列指令：

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

若要在所有的執行個體存放磁碟區上同時進行初始化，請使用下列指令：

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

設定 RAID 的磁碟這項動作，會藉由寫入所有磁碟位置來將這些磁碟初始化。設定軟體式 RAID 時，請務必變更最低重建速度：

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

Amazon EC2 執行個體儲存體磁碟區的詳細效能統計資料

Amazon EC2 為連接至 Nitro 型 Amazon EC2 執行個體的 NVMe 執行個體儲存體磁碟區提供即時、高解析度的效能統計資料。這些統計資料會以彙總計數器的形式呈現，並在執行個體的生命週期內保留。統計資料提供有關累計操作數量、傳送和接收的位元組、讀取和寫入 I/O 操作所花費的時間，以及讀取和寫入 I/O 操作的長條圖的詳細資訊。雖然這些統計資料會維持與 Amazon EBS 詳細效能統計資料的一致性，但其中也包含依 I/O 大小細分的詳細延遲長條圖，可讓您更深入了解儲存體效能模式。此增強可見性可仔細確定哪些特定 I/O 大小遇到延遲問題，讓您能夠最佳化應用程式效能，並更有效地對問題進行疑難排解。

最多可以 1 秒的間隔來收集這些統計資料。如果請求的頻率超過 1 秒的間隔，NVMe 驅動程式可能會將請求與其他管理員命令排入佇列，以便稍後處理。

考量事項

- 只有連接至 Nitro 型執行個體的 NVMe 執行個體儲存體磁碟區支援此統計資料。
- 執行個體停止和重新啟動時，計數器不會持續存在。
- 可免費使用統計資料。

統計資料

NVMe 區塊型儲存設備提供下列統計資料：

統計資料名稱	全名	Type	說明
total_rea d_ops	讀取操作總計	計數器	已完成讀取操作的總數。
total_wri te_ops	寫入操作總計	計數器	已完成寫入操作的總數。

統計資料名稱	全名	Type	說明
total_rea d_bytes	讀取位元組總計	計數器	傳輸的讀取位元組總數。
total_wri te_bytes	寫入位元組總計	計數器	傳輸的寫入位元組總數。
total_rea d_time	讀取時間總計	計數器	執行所有已完成讀取操作花費的總時間，以微秒為單位。
total_wri te_time	寫入時間總計	計數器	執行所有已完成寫入操作花費的總時間，以微秒為單位。
instance_ store_vol ume_perfo rmance_ex ceeded_io ps	總體時間需求超過磁碟區的最大 IOPS	計數器	IOPS 請求超過磁碟區最大 IOPS 的總時間 (以微秒為單位)。超過 0 的任何值都表示您的工作負載需要的 IOPS 超過磁碟區可以提供的數量。理想情況下，此指標在兩次快照時間之間的增量計數應最少。
instance_ store_vol ume_perfo rmance_ex ceeded_tp	總時間需求超過磁碟區的最大輸送量	計數器	輸送量請求超過磁碟區輸送量上限的總時間 (以微秒為單位)。超過 0 的任何值都表示您的工作負載需要的輸送量超過磁碟區可以提供的數量。理想情況下，此指標在兩次快照時間之間的增量計數應最少。
volume_qu eue_lengt h	磁碟區佇列長度	時間點	等待完成的讀取和寫入操作數目。
read_io_l atency_hi stogram	讀取 I/O 長條圖	長條圖 *	每個延遲時段內完成的讀取操作數量 (以微秒為單位)。

統計資料名稱	全名	Type	說明
`write_io_latency_histogram`	寫入 I/O 長條圖	長條圖 *	每個延遲時段內完成的寫入操作數量 (以微秒為單位)。

> **Note**
>
> * 直方圖統計資料僅代表已成功完成的 I/O 操作。停滯或受損的 I/O 操作不包含在內，但在 `volume_queue_length` 統計資料中將明顯呈現為時間點統計資料。

存取統計資料

必須直接從連接執行個體儲存體磁碟區的執行個體中存取統計資料。可以使用下列其中一個方法來存取統計資料。

Linux 執行個體

Amazon CloudWatch

可以設定 Amazon CloudWatch 代理程式，從執行個體中收集統計資料，並將其作為 CloudWatch 中的自訂指標使用。然後，可以使用 CloudWatch 中的指標來分析 I/O 模式，追蹤效能趨勢，建立自訂儀表板，並根據效能閾值設定自動警示。

如需有關設定 CloudWatch 代理程式的詳細資訊，請參閱收集 Amazon EC2 執行個體儲存體磁碟區指標。

nvme-cli tool

若要存取統計資料

1. 連接至磁碟區所連接的執行個體。
2. 2025 年 9 月 15 日之後發行的 Amazon Linux 2023 AMI 包含最新版本的 `nvme-cli` 工具。如果您使用的是舊版 Amazon Linux AMI，請更新 `nvme-cli` 工具。

```
sudo yum install nvme-cli
```

3. 執行下列命令，並指定磁碟區的設備名稱。

```
sudo nvme amzn stats /dev/nvme0n1
```

統計資料也提供依 I/O 大小細分的詳細延遲長條圖。若要檢視依 I/O 大小細分的統計資料，請包含 `--details` 選項。例如：

```
sudo nvme amzn stats --details /dev/nvme0n1
```

透過指定 `--help` 選項，可取得有關如何使用工具的詳細資訊。例如：

```
sudo nvme amzn stats --help
```

Windows 執行個體

nvme_amzn.exe tool

若要存取統計資料

1. 連接至磁碟區所連接的執行個體。
2. 請確定您使用的是 AWSNVMe 驅動程式版本 `1.7.0` 或更新版本。如需有關更新 AWSNVMe 驅動程式的詳細資訊，請參閱 AWS NVMe 驅動程式。
3. 取得磁碟區的磁碟編號。如需詳細資訊，請參閱將 Amazon EC2 Windows 執行個體上的 NVMe 磁碟映射至磁碟區。
4. 以管理員身分執行下列命令，並指定磁碟區的磁碟編號。

```
.\nvme_amzn.exe stats disk_number
```

統計資料也提供依 I/O 大小細分的詳細延遲長條圖。若要檢視依 I/O 大小細分的統計資料，請包含 `--details` 選項。例如：

```
.\nvme_amzn.exe stats --details disk_number
```

Amazon EC2 執行個體的根磁碟區

當您啟動執行個體時，系統會為執行個體建立根磁碟區。根磁碟區內含用來啟動執行個體的映像。每個執行個體都有一個根磁碟區。您可以在啟動期間或結束後，將儲存磁碟區新增至執行個體。

您啟動執行個體時使用的 AMI，會判斷根磁碟區的類型。您可以從 Amazon EBS 支援的 AMI (Linux 和 Windows 執行個體) 或Amazon S3 支援的 AMI (僅限 Linux 執行個體) 啟動執行個體。每種類型的 AMI 的用途存在顯著差異。如需這些差異的詳細資訊，請參閱 根磁碟區類型。

建議您使用 Amazon EBS 支援的 AMI，因這些執行個體啟動速度較快且採用持久性儲存。

系統會為根磁碟區保留特定的裝置名稱。如需詳細資訊，請參閱Amazon EC2 執行個體上磁碟區的裝置名稱。

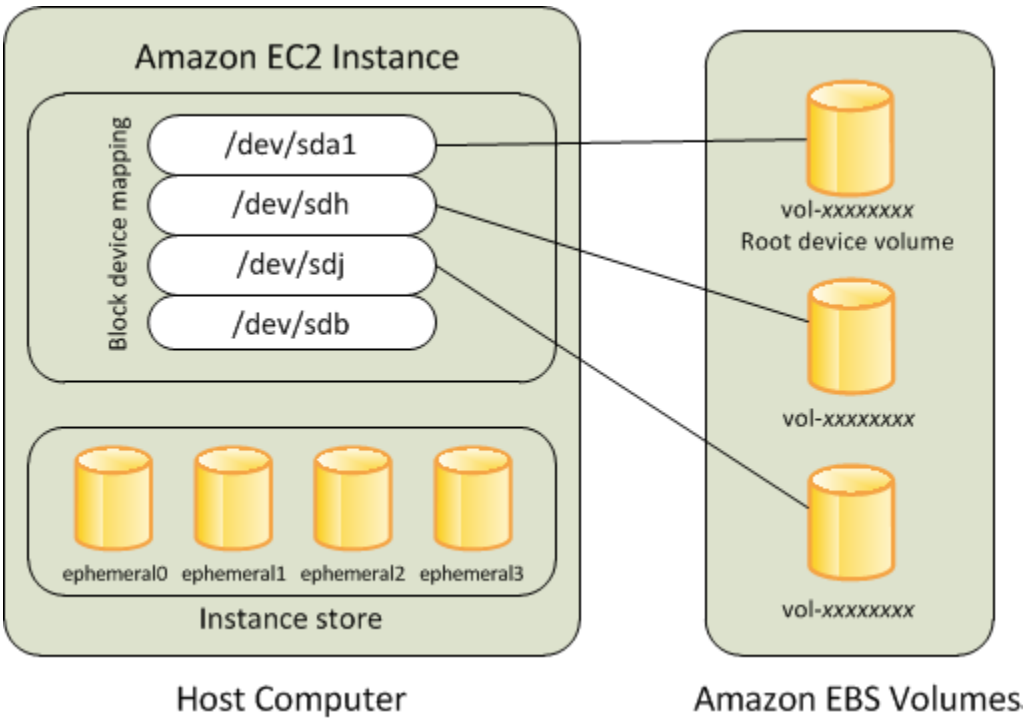
目錄

- 具有 Amazon EBS 根磁碟區的執行個體
- 具有執行個體儲存體根磁碟區的執行個體 （ 僅限 Linux 執行個體 ）
- 在 Amazon EC2 執行個體終止後保留 Amazon EBS 根磁碟區
- 在不停止 Amazon EC2 執行個體的情況下，取代根磁碟區。

具有 Amazon EBS 根磁碟區的執行個體

使用 Amazon EBS 做為根磁碟區的執行個體會自動連接 Amazon EBS 磁碟區。啟動具有 Amazon EBS-AMI 的執行個體時，我們會為 AMI 所參考的各個 Amazon EBS 快照，建立 Amazon EBS 磁碟區。您可根據執行個體類型選擇使用其他 Amazon EBS 磁碟區或執行個體存放區磁碟區。

具有 EBS 根磁碟區的執行個體可停止並在稍後重新啟動，不會影響連接磁碟區內儲存的資料。當具有 EBS 根磁碟區的執行個體處於停止狀態時，您仍可執行多種執行個體和磁碟區的相關任務。例如，您可修改執行個體的屬性、變更其大小、更新其正使用的核心，或者您可將根磁碟區連接至另一個運作中執行個體以進行除錯或其他用途。如需詳細資訊，請參閱 Amazon EBS 磁碟區。



限制

您無法使用 `st1` 或 `sc1` EBS 磁碟區做為根磁碟區。

執行個體失敗

若具有 EBS 根磁碟區的執行個體出現故障，您可採取下列方法之一來還原工作階段：

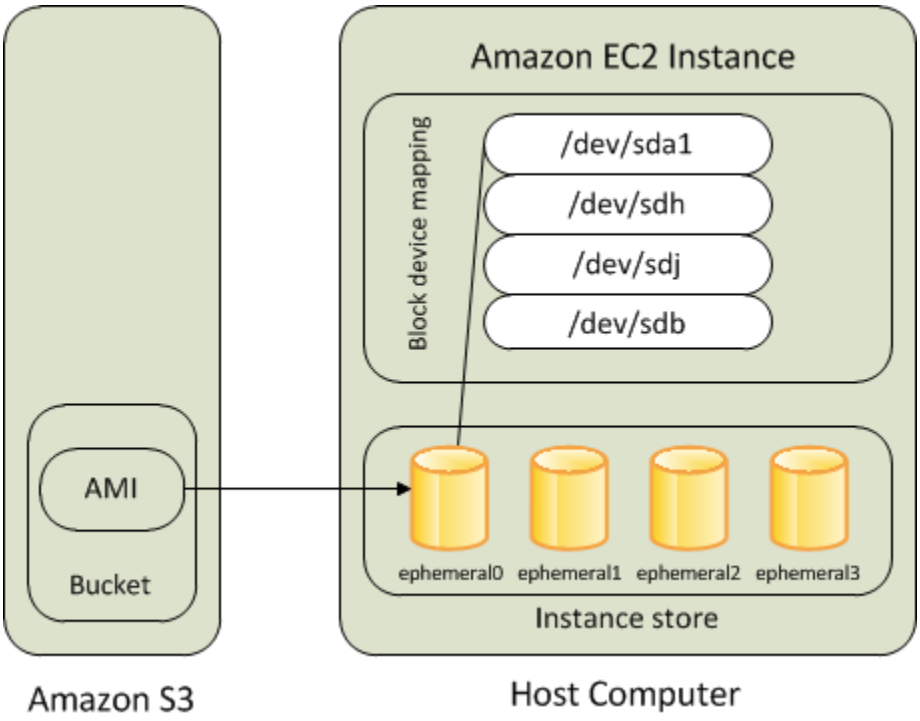
- 停止然後再次啟動 (請先嘗試此方法)。
- 自動拍攝所有相關磁碟區的快照，並建立新的 AMI。如需詳細資訊，請參閱 建立 Amazon EBS 支援的 AMI。
- 採取下列步驟將磁碟區連接至新的執行個體：
  1. 建立根磁碟區的快照。
  2. 使用快照註冊新的 AMI。
  3. 從新的 AMI 啟動新的執行個體。
  4. 將剩餘的 Amazon EBS 磁碟區自舊的執行個體分離。
  5. 將 Amazon EBS 磁碟區重新連接至新的執行個體。

具有執行個體儲存體根磁碟區的執行個體 （僅限 Linux 執行個體）

> **Note**
>
> Windows 執行個體不支援執行個體儲存體的根磁碟區。

將執行個體儲存體用於根磁碟區的執行個體，自動具備一個或多個可用的執行個體儲存體磁碟區，其中一個磁碟區會當成根磁碟區使用。從 Amazon S3 支援的 AMI 啟動執行個體時，該 AMI 會複製到根磁碟區。請注意，您可根據執行個體類型選擇使用其他執行個體存放區磁碟區。

只要執行個體持續運作，執行個體儲存體磁碟區上的資料就會永久保存，但若執行個體被終止 (具有執行個體儲存體根磁碟區的執行個體不支援停止動作) 或故障 (如底層磁碟區出現問題)，這些資料就會被刪除。如需詳細資訊，請參閱EC 2 執行個體的執行個體儲存體暫時區塊儲存。



支援的執行個體類型

只有下列執行個體類型支援執行個體儲存體磁碟區做為根磁碟區：
C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

執行個體失敗

具有執行個體儲存體根磁碟區的執行個體故障或終止後，將無法還原。若您計畫使用具有執行個體儲存體根磁碟區的執行個體，我們強烈建議您將執行個體儲存體磁碟區上的資料分散至多個可用區域。您也應將執行個體存放區磁碟區上的重要資料，定期備份為持久性儲存。

在 Amazon EC2 執行個體終止後保留 Amazon EBS 根磁碟區

根據預設，執行個體終止時，將刪除 Amazon EBS 根磁碟區。您可以變更預設行為，以確保在執行個體終止後 Amazon EBS 根磁碟區持續存在。若要變更此預設行為，請將 `DeleteOnTermination` 屬性設定為 `false`。您可以在執行個體啟動時，或啟動後執行此操作。

任務

- 將根磁碟區設定為執行個體啟動期間持續存在
- 將根磁碟區設定為若為現有執行個體則持續存在
- 確認根磁碟區已設定為持續存在

將根磁碟區設定為執行個體啟動期間持續存在

您可以設定根磁碟區在啟動執行個體期間持續存在。

Console

將根磁碟區設定為啟動執行個體時持續存在

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選擇 Launch instances (啟動執行個體)。
3. 依序選擇 Amazon Machine Image (AMI)、執行個體類型和金鑰對，然後設定網路設定。
4. 針對設定儲存，選擇進階。
5. 展開根磁碟區。
6. 針對在終止時刪除，選擇否。
7. 當您完成對執行個體的設定後，選擇啟動執行個體。

AWS CLI

將根磁碟區設定為啟動執行個體時持續存在

使用 run-instances 命令，並包含下列選項。

```
--block-device-mappings file://mapping.json
```

在 mapping.json 中，指定將 DeleteOnTermination 屬性設定為 false 的區塊型儲存設備映射。

```
[
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "DeleteOnTermination": false
        }
    }
]
```

PowerShell

將根磁碟區設定為啟動執行個體時持續存在

使用 New-EC2Instance cmdlet，並納入下列參數。

```
-BlockDeviceMapping $bdm
```

建立將 DeleteOnTermination 屬性設定為 $false 的區塊型儲存設備映射。

```
$ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "dev/xvda"
$bdm.Ebs = $ebs
```

將根磁碟區設定為若為現有執行個體則持續存在

您可以設定根磁碟區在執行個體執行期間持續存在。請注意，您無法使用 Amazon EC2 主控台完成此任務。

AWS CLI

將根磁碟區設定為若為現有執行個體則持續存在

使用 modify-instance-attribute 命令，搭配將 DeleteOnTermination 屬性設定為 false 的區塊型設備映射。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --block-device-mappings file://mapping.json
```

在 mapping.json 中指定下列內容。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "DeleteOnTermination": false
        }
    }
]
```

PowerShell

將根磁碟區設定為若為現有執行個體則持續存在

使用 Edit-EC2InstanceAttribute cmdlet，搭配將 DeleteOnTermination 屬性設定為 $false 的區塊型儲存設備映射。

```
$ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification
$bdm.DeviceName = "/dev/xvda"
$bdm.Ebs = $ebs
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -BlockDeviceMapping $bdm
```

確認根磁碟區已設定為持續存在

您可確認根磁碟區已設定為持續存在。

Console

確認根磁碟區已設定為持續存在

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選取 Instances (執行個體)，然後選取執行個體。
3. 在 Storage (儲存體) 標籤的 Block devices (區塊型裝置) 下，找出根磁碟區的項目。如果 Delete on termination (終止時刪除) 為 No，則磁碟區會設定為持續存在。

AWS CLI

確認根磁碟區已設定為持續存在

使用 describe-instance 命令，並驗證 `DeleteOnTermination` 屬性是否已設定為 `false`。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].BlockDeviceMappings"
```

以下為範例輸出。

```
[
    [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "AttachTime": "2024-07-12T04:05:33.000Z",
                "DeleteOnTermination": false,
                "Status": "attached",
                "VolumeId": "vol-1234567890abcdef0"

        }
    ]
]
```

PowerShell

確認根磁碟區已設定為持續存在

使用 Get-EC2Instance cmdlet，並驗證 `DeleteOnTermination` 屬性是否已設定為 `False`。

```
(Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

以下為範例輸出。

```
AssociatedResource  :
AttachTime          : 7/12/2024 4:05:33 AM
DeleteOnTermination : False
Operator            :
Status              : attached
VolumeId            : vol-1234567890abcdef0
```

在不停止 Amazon EC2 執行個體的情況下，取代根磁碟區。

Amazon EC2 可讓您取代執行中執行個體的根 Amazon EBS 磁碟區，同時保留下列項目：

- 儲存在執行個體存放區磁碟區上的資料 - 在還原根磁碟區後，執行個體存放區磁碟區仍會保持連接至執行個體。
- 資料 (非根) Amazon EBS 磁碟區上儲存的資料 - 在還原根磁碟區後，非根 Amazon EBS 磁碟區會保持連接到執行個體。
- 網路組態 – 所有網路介面都會保持連接至執行個體，並保留其 IP 地址、識別符和附件 ID。當執行個體可用時，會清除所有擱置中的網路流量。此外，執行個體會保留在相同的實體主機上，因此會保留其公有和私有 IP 地址和 DNS 名稱。
- IAM 政策 – IAM 描述檔和與執行個體相關聯的政策 (例如索引標籤政策) 會保留並強制執行。

目錄

- 取代根磁碟區的運作方式
- 考量事項
- 取代根磁碟區

取代根磁碟區的運作方式

取代執行個體的根磁碟區時，會建立根磁碟區取代任務。原始根磁碟區會從執行個體中分離，新根磁碟區會連接至其位置中的執行個體。執行個體的區塊型裝置映射已更新，以反映取代根磁碟區的 ID。

當您取代執行個體的根磁碟區時，您必須指定新磁碟區的來源。下列為可能選項。

將根磁碟區還原至原始狀態

此選項將目前根磁碟區取代為基於該磁碟區時所使用快照的新磁碟區。

使用啟動狀態的考量事項

取代根磁碟區會獲得與原始根磁碟區相同的類型、大小和「在終止時刪除」屬性。

使用快照取代根磁碟區

此選項會根據您指定的快照，將目前根磁碟區取代為替代磁碟區。例如，您先前在此根磁碟區所建立的特定快照。如果您需要復原客體操作系統中發生的根磁碟機或網路組態錯誤損壞，這個功能非常實用。

取代根磁碟區會獲得與原始根磁碟區相同的類型、大小和「在終止時刪除」屬性。

使用快照時的考量事項

- 您僅能使用直接從執行個體目前或先前根磁碟區建立的快照。
- 您無法使用從根磁碟區擷取的快照中建立的快照複本。
- 在成功取代根磁碟區之後，您依然可以使用從原始根磁碟區中取得的快照來取代新的 (取代) 根磁碟區。

使用 AMI 取代根磁碟區

此選項會使用您指定的 AMI 取代目前的根磁碟區。如果您需要執行作業系統和應用程式修補或升級，此功能非常實用。AMI 必須具有與執行個體相同的產品代碼、帳單資訊、架構類型和虛擬化類型。

如果已對 ENA 或 sriov-net 啟用執行個體，則您必須使用支援這些功能的 AMI。如果未對 ENA 或 sriov-net 啟用執行個體，您可以選擇不包含這些功能支援的 AMI，或者在選擇支援 ENA 或 sriov-net 的 AMI 時，自動新增對這些功能的支援。

如果已對 NitroTPM 啟用執行個體，則您必須使用已啟用 NitroTPM 的 AMI。如果尚未為 NitroTPM 設定執行個體，無論您選擇的 AMI 為何，NitroTPM 支援都不會啟用。

只要執行個體支援 AMI 的開機模式，您就可以選取開機模式與執行個體不同的 AMI。如果執行個體不支援該開機模式，則請求會失敗。如果執行個體支援該開機模式，則新的開機模式會傳播至執行個體，並相應地更新其 UEFI 資料。如果您已手動修改開機順序，或已新增私有 UEFI 安全開機金鑰來載入私有核心模組，則變更會在根磁碟區取代期間遺失。

取代根磁碟區獲得與原始根磁碟區相同的磁碟區類型和終止時刪除屬性。取代根磁碟區的大小是 AMI 根磁碟區區塊型設備映射大小和目前根磁碟區大小的較大者。

根磁碟區替換任務完成後，當您使用主控台 AWS CLI 或 AWS SDKs 描述執行個體時，會反映下列新的和更新的資訊：

- 新的 AMI ID

- 根磁碟區的新磁碟區 ID
- 已更新開機模式組態 (如果由 AMI 變更)
- 已更新 NitroTPM 組態 (如果由 AMI 啟用)
- 已更新 ENA 組態 (如果由 AMI 啟用)
- 已更新 sriov-net 組態 (如果由 AMI 啟用)

新的 AMI ID 也會反映在執行個體中繼資料中。

使用 AMI 時的考量事項

- 如果您使用具有多個區塊型裝置映射的 AMI，則只會使用 AMI 的根磁碟區。其他 (非根) 磁碟區會被忽略。
- 只有在您擁有使用 AMI 及其關聯根磁碟區快照的權限時，才能使用此功能。您無法將此功能與 AWS Marketplace AMIs搭配使用。
- 只有在執行個體沒有產品代碼的情況下，才能使用不含產品代碼的 AMI。
- 取代根磁碟區的大小是 AMI 根磁碟區區塊型設備映射大小和目前根磁碟區大小的較大者。
- 執行個體的執行個體身分文件會自動更新。
- 如果執行個體支援 NitroTPM，則會重設執行個體的 NitroTPM 資料，並產生新的金鑰。

使用現有的 Amazon EBS 磁碟區取代根磁碟區

與建立新磁碟區的快照和 AMI 選項不同，此選項會以您已設定的現有 Amazon EBS 磁碟區取代目前的根磁碟區。這對於資料庫等具狀態工作負載很有用，在您將中繼資料或軟體連接到執行個體之前，您要先在根磁碟區上準備中繼資料或軟體。

使用此選項之前，您必須建立和準備取代的 Amazon EBS 磁碟區。您可以將它作為資料磁碟區連接到執行個體，將所需的資料複製到執行個體，然後分離它，以準備它。磁碟區必須與執行個體位於相同的可用區域。

取代根磁碟區會保留您在建立和準備時所設定的大小、類型和其他屬性。原始根磁碟區的終止時刪除不會轉移到替代根磁碟區。

使用現有 Amazon EBS 磁碟區的考量事項

- 替換磁碟區必須與執行個體位於相同的可用區域。
- 取代磁碟區不得在請求時連接至任何執行個體。

- 替換磁碟區必須處於 `available` 狀態。
- 如果取代磁碟區不符合這些要求，根磁碟區取代請求會失敗。
- 您可以對 AWS Marketplace 執行個體使用此選項，但替代磁碟區上的市集產品代碼和帳單代碼都必須符合執行個體上的代碼。
- 如果原始根磁碟區已加密，則替換磁碟區也必須加密。您可以從未加密的磁碟區移至已加密的磁碟區，但不能從已加密的磁碟區移至未加密的磁碟區。加密替代磁碟區時，您的 IAM 主體必須具有分離原始磁碟區所需的 AWS KMS 許可。您還必須具有連接和解密替換磁碟區的許可。這些是連接加密磁碟區所需的相同許可。

您可以選擇是否要在根磁碟區取代程序完成後保留原始根磁碟區。如果您選擇在取代程序完成之後刪除原始根磁碟區，則原始根磁碟區會自動刪除並變得無法復原。如果您選擇在程序完成後保留原始根磁碟區，則該磁碟區仍會在您的帳戶中佈建；當您不再需要它時必須手動刪除磁碟區。

根磁碟區取代任務將通過以下狀態進行轉換：

- `pending` – 正在建立取代磁碟區。
- `in-progress` – 原始磁碟區正在分離，且取代磁碟區正在連接。
- `succeeded` – 取代磁碟區已成功連接至執行個體，且執行個體可用。
- `failing` – 取代任務處於失敗的程序中。
- `failed` – 取代任務失敗，但仍連接至根磁碟區。
- `failing-detached` - 取代任務正在失敗的過程中，執行個體可能不會連接根磁碟區。
- `failed-detached` – 取代任務失敗，且執行個體沒有連接根磁碟區。

如果根磁碟區取代任務失敗，則執行個體會重新啟動，而原始根磁碟區會保持連接到執行個體。

考量事項

開始之前，請考慮以下項目。

要求

- 執行個體必須處於 `running` 狀態。
- 執行個體會在過程中自動重新啟動。重新開機期間會清除記憶體 (RAM) 的內容。不需要手動重新啟動。
- 如果根磁碟區是執行個體存放區磁碟區，則無法取代根磁碟區。僅支援具有 Amazon EBS 根磁碟區的執行個體。

- 您可取代所有虛擬化執行個體類型和 EC2 Mac 裸機執行個體的根磁碟區。不支援其他裸機執行個體類型。
- 使用快照時，您只能使用直接從執行個體目前或先前的根磁碟區建立的快照。
- 如果帳戶在目前區域中已啟用 Amazon EBS 預設加密，無論指定快照的加密狀態或指定 AMI 的根磁碟區為何，都會一律加密根磁碟區取代任務建立的取代根磁碟區。

加密結果

下表摘要提供了可能的加密結果的摘要。

	原始根磁碟區	指定的快照、AMI 或磁碟區	預設加密	取代根磁碟區	用於取代根磁碟區的加密金鑰
將取代根磁碟區還原至初始啟動狀態	加密	不適用	不考慮	加密	與原始根磁碟區相同的 KMS 金鑰
	未加密	不適用	Disabled	未加密	不適用
	未加密	不適用	已啟用	加密	帳戶的 Amazon EBS 加密的預設 KMS 金鑰
從快照或 AMI 還原取代根磁碟區	加密	未加密	不考慮	加密	與原始根磁碟區相同的 KMS 金鑰
	加密	加密	不考慮	加密	與原始根磁碟區相同的 KMS 金鑰
	未加密	未加密	Disabled	未加密	不適用
	未加密	未加密	已啟用	加密	帳戶的 Amazon

	原始根磁碟區	指定的快照、AMI 或磁碟區	預設加密	取代根磁碟區	用於取代根磁碟區的加密金鑰
					EBS 加密的預設 KMS 金鑰
	未加密	加密	不考慮	加密	如果 AMI 或快照由帳戶擁有，則會使用 AMI 或快照的 KMS 金鑰加密取代磁碟區。如果與帳戶共用 AMI 或快照，則會使用該帳戶的 Amazon EBS 加密的預設 KMS 金鑰來加密取代磁碟區。
從現有的 Amazon EBS 磁碟區還原替換根磁碟區	加密	加密	不適用	加密	指定磁碟區的 KMS 金鑰
	加密	未加密	不適用	不允許。您無法將加密的根磁碟區取代為未加密的磁碟區。Amazon EC2 拒絕取代任務。	不適用
	未加密	未加密	不適用	未加密	不適用

	原始根磁碟區	指定的快照、AMI 或磁碟區	預設加密	取代根磁碟區	用於取代根磁碟區的加密金鑰
	未加密	加密	不適用	加密	指定磁碟區的 KMS 金鑰

取代根磁碟區

取代執行個體的根磁碟區時，會建立根磁碟區取代任務。您可以使用根磁碟區取代任務來監視取代過程的進度和結果。

Console

取代根磁碟區

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取要取代根磁碟區的執行個體，然後依序選取 Actions (動作)、Monitor and troubleshoot (監控與疑難排解)、Replace root volume (取代根磁碟區)。

   > **Note**
   >
   > 如果選定的執行個體不在 `running` 狀態，則會停用取代根磁碟區操作。

4. 針對區域，選擇以下其中一個選項：
   - 啟動狀態 – 從用來建立目前根磁碟區的快照還原取代根磁碟區。
   - 快照 – 將取代的根磁碟區還原至指定的快照。針對快照，選取要使用的快照。
   - 映像 – 使用您指定的 AMI 還原取代根磁碟區。針對映像，選取要使用的 AMI。
5. (選用) 對於磁碟區初始化速率，您可指定磁碟區初始化的 Amazon EBS 佈建速率 (磁碟區初始化速率)，以 MiB/s 為單位，快照區塊將以此速率從 Amazon S3 下載至磁碟區。如需詳細資訊，請參閱使用 Amazon EBS 佈建速率進行磁碟區初始化。要使用預設初始化速率或快速快照還原 (若已針對選取的快照啟用)，請勿指定速率。
6. (選用) 若要刪除要取代的根磁碟區，請選取刪除取代的根磁碟區。
7. 選擇建立取代任務。

8. 若要監控取代任務，請選擇執行個體的儲存體索引標籤，然後展開最近的根磁碟區取代任務。

AWS CLI

若要將取代根磁碟區還原至啟動狀態

使用 create-replace-root-volume-task 指令。若為 `--instance-id`，請指定要對其取代根磁碟區的執行個體 ID。省略 `--snapshot-id` 和 `--image-id` 參數。若要在取代原始根磁碟區之後將其刪除，請包含 `--delete-replaced-root-volume` 並指定 `true`。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 `--volume-initialization-rate` 指定介於 `100` 和 `300` MiB/s 之間的值。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

若要將取代根磁碟區還原至特定快照

使用 create-replace-root-volume-task 指令。若為 `--instance-id`，請指定要對其取代根磁碟區的執行個體 ID。若為 `--snapshot-id`，請指定要使用的快照 ID。若要在取代原始根磁碟區之後將其刪除，請包含 `--delete-replaced-root-volume` 並指定 `true`。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 `--volume-initialization-rate` 指定介於 `100` 和 `300` MiB/s 之間的值。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--snapshot-id snap-9876543210abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

若要使用 AMI 來還原取代根磁碟區

使用 create-replace-root-volume-task 指令。若為 `--instance-id`，請指定要對其取代根磁碟區的執行個體 ID。若為 `--image-id`，請指定要使用的 AMI ID。若要在取代原始根磁碟區之後將其刪除，請包含 `--delete-replaced-root-volume` 並指定 `true`。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 `--volume-initialization-rate` 指定介於 `100` 和 `300` MiB/s 之間的值。

```
aws ec2 create-replace-root-volume-task \
```

```
--instance-id i-1234567890abcdef0 \
--image-id ami-09876543210abcdef \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

使用現有的 Amazon EBS 磁碟區取代根磁碟區

使用 create-replace-root-volume-task 指令。針對 --instance-id，指定要取代根磁碟區的執行個體 ID（例如 i-1234567890abcdef0)。對於 --volume-id，指定要用作新根磁碟區的 Amazon EBS 磁碟區的 ID。若要在取代完成後刪除原始根磁碟區，請包含 --delete-replaced-root-volume。Amazon EC2 接受 --volume-initialization-rate 參數，但會忽略此選項，因為它不會下載快照資料。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--volume-id vol-0123456789abcdef0 \
--delete-replaced-root-volume
```

檢視根磁碟區取代任務的狀態

使用 describe-replace-root-volume-tasks 命令，並指定要檢視的根磁碟區取代任務的 ID。

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-1234567890abcdef0 \
    --query ReplaceRootVolumeTasks[].TaskState
```

以下為範例輸出。

```
[
    "succeeded"
]
```

或者，指定要依執行個體篩選結果的 instance-id 篩選條件。

```
$ aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-1234567890abcdef0
```

PowerShell

若要將取代根磁碟區還原至啟動狀態

使用 New-EC2ReplaceRootVolumeTask 命令。若為 -InstanceId，請指定要對其取代根磁碟區的執行個體 ID。省略 -SnapshotId 和 -ImageId 參數。若要在取代原始根磁碟區之後將其刪除，請包含 -DeleteReplacedRootVolume 並指定 $true。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 -VolumeInitializationRate 指定介於 100 和 300 MiB/s 之間的值。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

若要將取代根磁碟區還原至特定快照

使用 New-EC2ReplaceRootVolumeTask 命令。若為 --InstanceId，請指定要對其取代根磁碟區的執行個體 ID。若為 -SnapshotId，請指定要使用的快照 ID。若要在取代原始根磁碟區之後將其刪除，請包含 -DeleteReplacedRootVolume 並指定 $true。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 -VolumeInitializationRate 指定介於 100 和 300 MiB/s 之間的值。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -SnapshotId snap-9876543210abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

若要使用 AMI 來還原取代根磁碟區

使用 New-EC2ReplaceRootVolumeTask 命令。若為 -InstanceId，請指定要對其取代根磁碟區的執行個體 ID。若為 -ImageId，請指定要使用的 AMI ID。若要在取代原始根磁碟區之後將其刪除，請包含 -DeleteReplacedRootVolume 並指定 $true。要指定快照區塊從 Amazon S3 下載到磁碟區的磁碟區初始化速率，請針對 -VolumeInitializationRate 指定介於 100 和 300 MiB/s 之間的值。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -ImageId ami-0abcdef1234567890 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

使用現有的 Amazon EBS 磁碟區取代根磁碟區

使用 New-EC2ReplaceRootVolumeTask 命令。針對 -InstanceId，指定要取代根磁碟區的執行個體 ID（例如 i-1234567890abcdef0)。對於 -VolumeId，指定要用作新根磁碟區的 Amazon EBS 磁碟區的 ID。若要在取代完成後刪除原始根磁碟區，請包含-DeleteReplacedRootVolume並指定 $true。Amazon EC2 接受 -VolumeInitializationRate 參數，但會忽略此選項，因為它不會下載快照資料。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -VolumeId vol-0123456789abcdef0 `
    -DeleteReplacedRootVolume $true
```

檢視根磁碟區取代任務的狀態

使用 Get-EC2ReplaceRootVolumeTask 命令，並指定要檢視之根磁碟區取代任務的 ID。

```
(Get-EC2ReplaceRootVolumeTask `
    -ReplaceRootVolumeTaskIds replacevol-1234567890abcdef0).TaskState.Value
```

以下為範例輸出。

```
Succeeded
```

或者，指定要依執行個體篩選結果的 instance-id 篩選條件。

```
PS C:\> Get-EC2ReplaceRootVolumeTask -Filters @{Name = 'instance-id'; Values =
 'i-1234567890abcdef0'} | Format-Table
```

Amazon EC2 執行個體上磁碟區的裝置名稱

當您將磁碟區連接到您的執行個體時，您會在其中包含磁碟區的裝置名稱。Amazon EC2 會使用此裝置名稱。執行個體的區塊型儲存設備驅動程式會在掛載磁碟區時指派實際的磁碟區名稱，其指派的名稱會與 Amazon EC2 使用的名稱不同。

您執行個體可以支援的磁碟區數目取決於作業系統。如需詳細資訊，請參閱 Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。

目錄

- 可用裝置名稱
- 裝置名稱考量

可用裝置名稱

Linux 執行個體

Linux 執行個體有兩種可用的虛擬化類型：全虛擬化 (PV) 或硬體虛擬機器 (HVM)。執行個體的虛擬化類型會由用來啟動執行個體的 AMI 決定。所有執行個體類型都支援 HVM AMI。某些前一代執行個體類型支援 PV AMI。請注意您 AMI 的虛擬化類型，因為您可以使用的建議及可用裝置名稱會取決於您執行個體的虛擬化類型。如需詳細資訊，請參閱虛擬化類型。

下表列出您可以在區塊型設備映射中或在連結 EBS 磁碟區時指定的可用裝置名稱。

虛擬化類型	Available	預留供根磁碟區使用	建議用於 EBS 資料磁碟區	執行個體儲存體磁碟區
全虛擬化	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[f-p] /dev/sd[f-p][1-6]	/dev/sd[b-e]
HVM	/dev/sd[a-z] /dev/xvd[a-c][a-z] /dev/xvdd[a-x]	根據 AMI 而不同 /dev/sda1 或 /dev/xvda	/dev/sd[b-z] /dev/xvdb[b-z] *	/dev/sd[b-e] /dev/sd[b-h] (h1.16xlarge) /dev/sd[b-y] (d2.8xlarge) /dev/sd[b-i] (i2.8xlarge) **

* 您在區塊型設備映射中針對 NVMe EBS 磁碟區指定的裝置名稱會使用 NVMe 裝置名稱 (/dev/nvme[0-26]n1) 重新命名。區塊型儲存設備驅動程式指派 NVMe 設備名稱的順序，可能會與您在區塊型設備映射中為磁碟區指定的順序不同。

** NVMe 執行個體存放區磁碟區會自動列舉並指派 NVMe 裝置名稱。

Windows 執行個體

AWS Windows AMIs 使用下列其中一組驅動程式，以允許存取虛擬化硬體：

- AWS PV： Windows 執行個體的全虛擬驅動程式
- AWS NVMe： NVMe 驅動程式

基於 Nitro 的執行個體的設備名稱

下表列出您可以在區塊型儲存設備映射中或在將 EBS 磁碟區連結至基於 Nitro 的執行個體時指定的可用設備名稱。

驅動程式類型	Available	預留供根磁碟區使用	建議用於 EBS 磁碟區	執行個體儲存體磁碟區
AWS NVMe	xvd[a-z] xvd[a-c][a-z] xvdd[a-x] /dev/sda1	/dev/sda1	xvd[b-z] xvdb[b-z]	*

* 會自動列舉 NVMe 執行個體儲存體磁碟區，並指派一個 Windows 磁碟機代號。

基於 Xen 的執行個體的設備名稱

下表列出您可以在區塊型儲存設備映射中或在將 EBS 磁碟區連結至基於 Xen 的執行個體時指定的可用設備名稱。

驅動程式類型	Available	預留供根磁碟區使用	建議用於 EBS 磁碟區	執行個體儲存體磁碟區
AWS PV	xvd[b-z]	/dev/sda1	xvd[f-z]	xvdc[a-x]

驅動程式類型	Available	預留供根磁碟區使用	建議用於 EBS 磁碟區	執行個體儲存體磁碟區
	xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]			xvd[a-e]
Citrix PV (不再支援)	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Red Hat PV (不再支援)	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvdc[a-x] xvd[a-e]

如需執行個體存放區磁碟區的詳細資訊，請參閱EC 2 執行個體的執行個體儲存體暫時區塊儲存。如需 NVMe EBS 磁碟區 (Nitro 型執行個體) 的詳細資訊 (包括如何識別 EBS 裝置)，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 和 NVMe。

裝置名稱考量

當您選取裝置名稱時，請注意下列項目：

- 您所用裝置名稱的結尾部分不應重疊，因為這可能會在您啟動執行個體時造成問題。例如，請避免為連接至相同執行個體的磁碟區使用組合，例如 `/dev/xvdf` 和 `xvdf`。
- 雖然您可以使用用來連接執行個體存放區磁碟區的裝置名稱來連接您的 EBS 磁碟區，但我們強烈建議您不要這麼做，因為該行為可能會無法預期。
- 執行個體的 NVMe 執行個體存放區磁碟區數目取決於執行個體的大小。NVMe 執行個體儲存體磁碟區會自動列舉並指派 NVMe 裝置名稱 (Linux 執行個體) 或 Windows 磁碟機代號 (Windows 執行個體)。

- (Windows 執行個體） AWS Windows AMIs隨附其他軟體，可在第一次開機時準備執行個體。可能是 EC2Config 服務 (Windows Server 2016 前的 Windows AMI) 或 EC2Launch (Windows Server 2016 和更新版本)。在裝置映射到磁碟機後，便會初始化並掛載。根磁碟機會初始化並掛載為 C:\。根據預設，EBS 磁碟區連接至 Windows 執行個體時，可能會在執行個體上顯示任何磁碟機代號。您可以根據您的規格，變更設定以設定磁碟區的磁碟機代號。對於執行個體存放磁碟區，預設值取決於驅動程式。 AWS PV 驅動程式和 Citrix PV 驅動程式會指派從 Z： 到 A： 的執行個體存放磁碟區磁碟機代號。Red Hat 驅動程式會為執行個體儲存體磁碟區指派 D: 到 Z: 的磁碟機代號。如需詳細資訊，請參閱 Amazon EC2 Windows 執行個體的 Windows 啟動代理程式 和 如何為 Amazon EC2 Windows 執行個體連接和映射磁碟區。
- (Linux 執行個體) 視核心的區塊型儲存設備驅動程式而定，裝置可能可以使用和您指定的不同名稱來連接。例如，假設您指定了裝置名稱 /dev/sdh，您的裝置可以重新命名為 /dev/xvdh 或 /dev/hdh。在大部分情況下，結尾的字母會維持相同。在 Red Hat Enterprise Linux (及其變體，例如 CentOS) 的某些版本中，結尾的字母可能會變更 (/dev/sda 可能會變成 /dev/xvde)。在這些情況下，各裝置名稱的結尾字母會以相同的次數遞增。例如，如果 /dev/sdb 重新命名為 /dev/xvdf，則 /dev/sdc 重新命名為 /dev/xvdg。Amazon Linux 會為您指定給已重新命名裝置的名稱建立符號連結。其他作業系統可能會有不同的行為。
- (Linux 執行個體) HVM AMI 不支援在設備名稱上使用尾碼，但 /dev/sda1 (這是預留給根磁碟區) 和 /dev/sda2 除外。可以使用 /dev/sda2 時，不建議使用此裝置映射與 HVM 執行個體搭配。
- (Linux 執行個體) 使用 PV AMI 時，您無法連接共用相同磁碟機代號的磁碟機，無論其後方是否有數字。例如，若您將磁碟區連接為 /dev/sdc，並將另一個磁碟區連接為 /dev/sdc1，執行個體只會看見 /dev/sdc。若要在裝置名稱中使用尾碼，您必須在所有共享相同基礎代號的裝置名稱上使用尾碼 (例如 /dev/sdc1、/dev/sdc2、/dev/sdc3)。
- (Linux 執行個體) 有些自訂核心可能會限制使用 /dev/sd[f-p] 或 /dev/sd[f-p][1-6]。若您在使用 /dev/sd[q-z] 或 /dev/sd[q-z][1-6] 上發生問題，請嘗試切換為 /dev/sd[f-p] 或 /dev/sd[f-p][1-6]。

在您指定所選裝置名稱之前，請先確認該裝置名稱是否可用。否則，您會收到裝置名稱使用中的錯誤。若要檢視磁碟裝置及其掛載點，請使用 lsblk 命令 (Linux 執行個體) 或 Disk Management 公用程式，或使用 diskpart 命令 (Windows 執行個體)。

Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射

您啟動的每個執行個體均有一個相關聯的根磁碟區，這可能是 Amazon EBS 磁碟區或執行個體儲存體磁碟區。您可以使用區塊型儲存設備映射，來指定執行個體啟動時要連接的其他 EBS 磁碟區或執行個體儲存體磁碟區。您也可以將其他的 EBS 磁碟區連接到執行中的執行個體。但是，唯一能將執行個體

儲存體磁碟區連接至執行個體的方式，就是使用區塊型儲存設備映射來連接磁碟區，如同執行個體啟動時一樣。

目錄

- 區塊型儲存設備映射概念
- 將區塊型儲存設備映射新增至 AMI
- 將區塊型儲存設備映射新增至 Amazon EC2 執行個體

區塊型儲存設備映射概念

區塊型儲存設備是一種儲存設備，它可依位元組或位元 (區塊) 的順序移動資料。這些裝置支援隨機存取，且通常使用緩衝 I/O。例如硬碟、CD-ROM 光碟機和快閃磁碟機。區塊型設備可實際連接至電腦，或如同實際連接至電腦一般進行遠端存取。

Amazon EC2 支援兩種類型的區塊型設備：

- 執行個體存放磁碟區 (基礎硬體實際連接到執行個體主機電腦的虛擬設備)
- EBS 磁碟區 (遠端儲存設備)

區塊型設備映射定義要連接到執行個體的區塊型儲存設備 (執行個體存放磁碟區和 EBS 磁碟區)。您可在建立 AMI 時指定區塊型設備映射，如此從 AMI 啟動的所有執行個體都會使用該映射。或者，也可在啟動執行個體時指定區塊型設備映射，則此映射將覆寫啟動執行個體之 AMI 中指定的映射。請注意，某一執行個體類型支援的所有 NVMe 執行個體存放磁碟區會在執行個體啟動時自動列舉，並獲指派設備名稱；將這些磁碟區加入區塊型設備映射中並無作用。

內容

- 區塊型儲存設備映射項目
- 區塊型儲存設備映射執行個體存放區 Caveat
- 區塊型儲存設備映射範例
- 如何在作業系統中將設備變為可用

區塊型儲存設備映射項目

建立區塊型設備映射時，您需為要連接至執行個體的每個區塊型儲存設備指定下列資訊：

- Amazon EC2 內使用的裝置名稱。執行個體的區塊型儲存設備驅動程式，會在掛載磁碟區時指派實際的磁碟區名稱。指派的名稱可能和 Amazon EC2 建議的名稱不一樣。如需詳細資訊，請參閱 Amazon EC2 執行個體上磁碟區的裝置名稱。

對於執行個體存放區磁碟區，您也可以指定下列資訊：

- 虛擬裝置：`ephemeral[0-23]`。請注意，可供執行個體使用的執行個體存放磁碟區數量和大小，根據執行個體類型而異。

對於 NVMe 執行個體存放區磁碟區，也適用下列資訊：

- 這些磁碟區會自動列舉，並獲指派設備名稱；將這些磁碟區加入區塊型設備映射中並無作用。

對於 EBS 磁碟區，您也可以指定下列資訊：

- 用於建立區塊型儲存設備之快照的 ID (snap-xxxxxxxx)。若已指定磁碟區大小，則此值是選用的。無法指定已封存快照的 ID。
- 磁碟區的大小 (GiB)。指定的大小必須大於或等於指定快照的大小。
- 是否在執行個體終止時刪除磁碟區 (`true` 或 `false`)。根磁碟區的預設值為 `true`，連接磁碟區的預設值為 `false`。建立 AMI 時，其區塊型設備映射會從執行個體繼承此設定。啟動執行個體時，其會從 AMI 繼承此設定。
- 磁碟區類型對於一般用途 SSD 為 `gp2` 和 `gp3`，對於佈建 IOPS SSD 為 `io1` 和 `io2`，對於輸送量最佳化 HDD 為 `st1`，對於冷 HDD 為 `sc1`，對於磁帶為 `standard`。
- 磁碟區支援的每秒輸入/輸出操作數 (IOPS)。(僅與 `io1` 和`io2` 磁碟區搭配使用。)
- 有些執行個體類型支援多個 EBS 卡。您可以指定 EBS 卡索引，為要連接的磁碟區選取 EBS 卡。如需詳細資訊，請參閱EBS 卡。

區塊型儲存設備映射執行個體存放區 Caveat

使用區塊型設備映射中有執行個體存放磁碟區的 AMIs 啟動執行個體時，有幾個注意事項。

- 部分執行個體類型包含的執行個體存放磁碟區較多，有些執行個體類型則完全沒有執行個體存放磁碟區。如果您的執行個體類型支援一個執行個體存放磁碟區，而 AMI 有兩個執行個體存放磁碟區的映射，則執行個體將以一個執行個體存放磁碟區啟動。
- 執行個體存放磁碟區只能在啟動時映射。您無法停止沒有執行個體存放磁碟區的執行個體 (例如 `t2.micro`)，請將執行個體變更為支援執行個體存放磁碟區的類型，然後重新啟動含執行個體存放

磁碟區的執行個體。但您可從執行個體建立 AMI，並在支援執行個體存放磁碟區的執行個體類型上啟動，然後將這些執行個體存放磁碟區映射至執行個體。

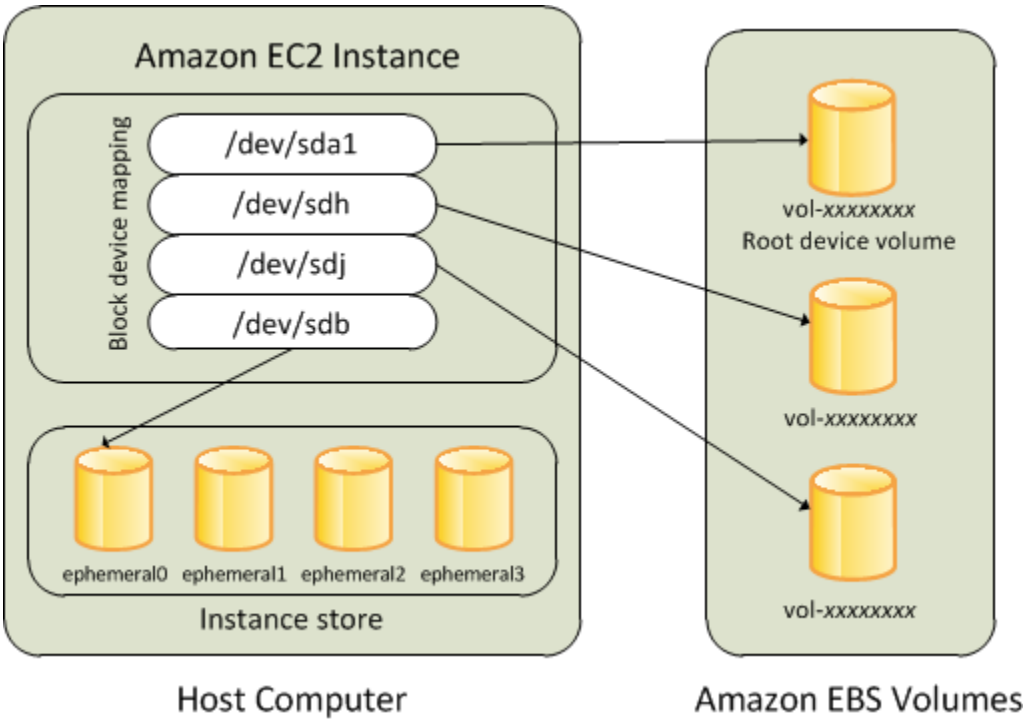
- 如果啟動有執行個體存放磁碟區映射的執行個體，接著停止執行個體，將其變更為執行個體存放磁碟區較少的執行個體類型，並重新啟動執行個體，則一開始啟動的執行個體存放磁碟區映射仍會出現在執行個體中繼資料內。但是，只有該執行個體類型支援的執行個體存放磁碟區數量上限可供執行個體使用。

> **Note**
>
> 停止執行個體時，執行個體存放磁碟區上的所有資料都會遺失。

- 根據啟動時的執行個體存放區容量而定，除非於啟動時指定，否則 M3 執行個體啟動時可能會忽略 AMI 執行個體存放區的區塊型設備映射。您應在啟動時指定執行個體存放區的區塊型設備映射，即使要啟動的 AMI 在 AMI 內具有執行個體存放磁碟區映射，以確保執行個體存放磁碟區在執行個體啟動時可用。

區塊型儲存設備映射範例

下圖顯示 EBS 後端執行個體的區塊型設備映射範例。其將 `/dev/sdb` 映射至 `ephemeral0`，並映射兩個 EBS 磁碟區，一個映射至 `/dev/sdh`，另一個映射至 `/dev/sdj`。亦顯示為根磁碟區 `/dev/sda1` 的 EBS 磁碟區。



請注意，此區塊型設備映射範例使用於本主題的命令和 API 範例中。您可在指定 AMI 的區塊型儲存裝置對應和啟動執行個體時更新區塊型儲存裝置對應中找到建立區塊型設備映射的命令和 API 範例。

如何在作業系統中將設備變為可用

Amazon EC2 使用 /dev/sdh 和 xvdh 等設備名稱來描述區塊型儲存設備。Amazon EC2 使用區塊型設備映射來指定要連接到 EC2 執行個體的區塊型儲存設備。區塊型儲存設備連接至執行個體之後，作業系統必須掛載儲存設備，接著方可供存取。區塊型儲存設備與執行個體分離之後，作業系統會卸載儲存設備，於是便無法再存取。

Linux 執行個體 – 執行個體第一次啟動時，區塊型儲存設備映射中指定的設備名稱會映射至對應的區塊型儲存設備。執行個體類型將決定預設要格式化及掛載的執行個體存放磁碟區。您可在啟動時掛載額外的執行個體存放磁碟區，只要不超過執行個體類型可用的執行個體存放磁碟區數量即可。如需詳細資訊，請參閱 EC 2 執行個體的執行個體儲存體暫時區塊儲存。執行個體的區塊型儲存設備驅動程式決定格式化及掛載磁碟區時要使用的設備。

Windows 執行個體 – 執行個體第一次啟動時，區塊型儲存設備映射中指定的設備名稱會映射至對應的區塊型儲存設備，然後 Ec2Config 服務將初始化並掛載磁碟機。根磁碟區將掛載為 C:\。執行個體存放磁碟區將掛載為 Z:\、Y:\，並以此類推。掛載 EBS 磁碟區時，其可用任何可用的磁碟機代號掛載。不過，您可設定指派磁碟機代號至 EBS 磁碟機的方式；如需詳細資訊，請參閱 the section called “Windows 啟動代理程式”。

將區塊型儲存設備映射新增至 AMI

每個 AMI 皆有區塊型設備映射，其指定從 AMI 啟動時要連接至執行個體的區塊型儲存設備。若要新增更多區塊型儲存設備至 AMI，您必須建立自己的 AMI。

內容

- 指定 AMI 的區塊型儲存裝置對應
- 檢視 AMI 區塊型儲存裝置對應的 EBS 磁碟區

指定 AMI 的區塊型儲存裝置對應

有兩種方法可在建立 AMI 時指定除了根磁碟區以外的其他磁碟區。如果您從執行個體建立 AMI 之前已將磁碟區連接至執行中的執行個體，AMI 的區塊型設備映射將包含這些相同的磁碟區。如為 EBS 磁碟區，現有資料將儲存至新快照，而區塊型設備映射中將指定這個新快照。如為執行個體存放磁碟區，資料將不會保留。

如為 EBS 後端 AMI，您可用區塊型設備映射新增 EBS 磁碟區和執行個體存放磁碟區。如為 Amazon S3 支援的 AMI，您只能在登錄映像時修改映像工作資訊清單檔案中的區塊型儲存設備映射項目，以此方式來新增執行個體儲存體磁碟區。

> **Note**
>
> 如為 M3 執行個體，您必須在啟動時在執行個體的區塊型設備映射中指定執行個體存放磁碟區。啟動 M3 執行個體時，若未指定為執行個體區塊型設備映射，AMI 區塊型設備映射中指定的執行個體存放磁碟區將遭忽略。

Console

將磁碟區新增至 AMI

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取執行個體並選取 Actions (動作)、Image and templates (影像和範本)、Create image (建立影像)。
4. 輸入影像的名稱和說明。
5. 執行個體磁碟區會出現在 Instance volumes (執行個體磁碟區) 下。若要新增其他磁碟區，請選擇 Add volume (新增磁碟區) 。
6. 針對 Volume Type (磁碟區類型)，選擇一種磁碟區類型。對於Device (裝置)，請選擇裝置名稱。對於 EBS 磁碟區，您可以指定其他詳細資訊，例如：快照、磁碟區大小、磁碟區類型、IOPS 和加密狀態。
7. 選擇建立映像。

AWS CLI

將磁碟區新增至 AMI

使用 create-image 命令，指定基於 EBS 的 AMI 的區塊型儲存設備映射。使用 register-image 命令，指定基於 Amazon S3 的 AMI 的區塊型儲存設備映射。

使用 `--block-device-mappings` 參數指定區塊型儲存設備映射。可以直接在命令列指定以 JSON 編碼的引數，或參考 JSON 檔案，如下所示。

```
--block-device-mappings file://mapping.json
```

若要新增執行個體存放磁碟區，請使用下列映射。請注意，會自動新增 NVMe 執行個體儲存體磁碟區。

```
{
    "DeviceName": "device_name",
    "VirtualName": "ephemeral0"
}
```

若要新增空的 100 GiB 磁碟區，請使用下列映射。

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

若要以快照為基礎新增 EBS 磁碟區，請使用下列映射。

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "SnapshotId": "snap-1234567890abcdef0"
    }
}
```

若要省略設備的映射，請使用下列映射。

```
{
    "DeviceName": "device_name",
    "NoDevice": ""
}
```

PowerShell

使用 New-EC2Image cmdlet，指定基於 EBS 的 AMI 的區塊型儲存設備映射。使用 Register-EC2Image cmdlet，指定基於 Amazon S3 的 AMI 的區塊型儲存設備映射。

新增 -BlockDeviceMapping 選項，在 bdm 中指定更新：

```
-BlockDeviceMapping $bdm
```

下列範例根據快照，新增磁碟區。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.SnapshotId = "snap-1234567890abcdef0"
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

以下映射會新增 100 GiB 的空磁碟區。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.VolumeSize = 100
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

檢視 AMI 區塊型儲存裝置對應的 EBS 磁碟區

您可輕鬆列舉 AMI 區塊型設備映射中的 EBS 磁碟區。

Console

若要檢視 AMI 的 EBS 磁碟區

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 AMIs (AMI)。
3. 從 Filter (篩選條件) 清單中選擇 EBS images (EBS 映像)，以顯示 EBS 後端 AMI 的清單。
4. 選取所要的 AMI，然後檢視 Details (詳細資訊) 索引標籤。根磁碟區至少具有以下資訊 (其中根裝置一詞等同於根磁碟區)：

   - Root Device Type (根設備類型)ebs ()
   - Root Device Name (根設備名稱) (例如，/dev/sda1)
   - Block Devices (區塊型儲存設備) (例如，/dev/sda1=snap-1234567890abcdef0:8:true)

如果使用區塊型設備映射建立含額外 EBS 磁碟區的 AMI，Block Devices (區塊型儲存設備) 欄位也會顯示額外磁碟區的映射。(此畫面不會顯示執行個體存放磁碟區。)

AWS CLI

若要檢視 AMI 的 EBS 磁碟區

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Image[0].BlockDeviceMappings
```

PowerShell

若要檢視 AMI 的 EBS 磁碟區

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).BlockDeviceMappings
```

將區塊型儲存設備映射新增至 Amazon EC2 執行個體

根據預設，您所啟動的執行個體包含啟動執行個體之 AMI 區塊型設備映射中指定的任何儲存設備。您可在啟動執行個體時指定變更執行個體的區塊型設備映射，這些更新將覆寫或合併 AMI 的區塊型設備映射。

限制

- 如為根磁碟區，只能修改：磁碟區大小、磁碟區類型和 Delete on Termination (在終止時刪除) 旗標。
- 修改 EBS 磁碟區時，您無法減少其大小。因此，您所指定的快照大小必須等於或大於 AMI 區塊型設備映射中的快照。

任務

- 啟動執行個體時更新區塊型儲存裝置對應
- 更新執行中執行個體的區塊型儲存裝置對應

- 檢視執行個體區塊型儲存裝置對應的 EBS 磁碟區
- 檢視執行個體區塊型儲存裝置對應的執行個體存放區的磁碟區

啟動執行個體時更新區塊型儲存裝置對應

您可在啟動執行個體時新增 EBS 磁碟區和執行個體存放磁碟區至執行個體。但請注意，更新執行個體的區塊型設備映射，不會永久變更啟動執行個體之 AMI 的區塊型設備映射。

Console

若要在啟動時更新執行個體的磁碟區

1. 遵循程序來啟動執行個體，但在完成下列步驟以更新磁碟區之前，請勿啟動執行個體。
2. (選用) 若要新增磁碟區，請選擇設定儲存、新增新磁碟區。選取磁碟區大小和磁碟區類型。
3. (選用) 若要隱藏由 AMI 的區塊型儲存設備映射指定的磁碟區，請選擇設定儲存、移除。
4. (選用) 若要修改 EBS 磁碟區的組態，請在設定儲存窗格中選擇進階。展開磁碟區的資訊，並進行任何所需變更。
5. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

AWS CLI

若要在啟動時更新執行個體的磁碟區

請使用 `--block-device-mappings` 選項並搭配 run-instances 命令。

```
--block-device-mappings file://mapping.json
```

例如，假設 AMI 區塊型儲存設備映射可指定下列內容：

- `/dev/xvda` - EBS 根磁碟區
- `/dev/sdh` - 從 `snap-1234567890abcdef0` 中建立的 EBS 磁碟區
- `/dev/sdj` - 空白 EBS 磁碟區，大小為 `100`
- `/dev/sdb` - 執行個體儲存體磁碟區 `ephemeral0`

假設以下是 `mapping.json` 中的執行個體區塊型儲存設備映射。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 100
        }
    },
    {
        "DeviceName": "/dev/sdj",
        "NoDevice": ""
    },
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 300
        }
    },
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
]
```

執行個體區塊型儲存設備映射會執行下列動作：

- 覆寫根磁碟區的大小，/dev/xvda，將其增加到 100 GiB。
- 防止 /dev/sdj 連接至執行個體。
- 覆寫 /dev/sdh 的大小，將其增加到 300 GiB。請注意，不需要再次指定快照 ID。
- 新增暫時性磁碟區，/dev/sdc。如果執行個體類型不支援多重執行個體儲存體磁碟區，這將沒有效果。如果執行個體類型支援 NVMe 執行個體儲存體磁碟區，會自動列舉這些磁碟區並包含在執行個體區塊型儲存設備映射中，且無法覆寫。

PowerShell

若要在啟動時更新執行個體的磁碟區

搭配使用 -BlockDeviceMapping 參數與 New-EC2Instance cmdlet 以及 -BlockDeviceMapping 參數。

```
-BlockDeviceMapping $bdm
```

假設以下是 $bdm 中的執行個體區塊型儲存設備映射。

```
$bdm = @()

$root = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$root.DeviceName = "/dev/xvda"
$ebs1 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs1.VolumeSize = 100
$root.Ebs = $ebs1
$bdm += $root

$sdj = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdj.DeviceName = "/dev/sdj"
$sdj.NoDevice = ""
$bdm += $sdj

$sdh = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdh.DeviceName = "/dev/sdh"
$ebs2 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs2.VolumeSize = 300
$sdh.Ebs = $ebs2
$bdm += $sdh

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral1"
$bdm += $sdc
```

執行個體區塊型儲存設備映射會執行下列動作：

- 覆寫根磁碟區的大小，/dev/xvda，將其增加到 100 GiB。
- 防止 /dev/sdj 連接至執行個體。
- 覆寫 /dev/sdh 的大小，將其增加到 300 GiB。請注意，不需要再次指定快照 ID。
- 新增暫時性磁碟區，/dev/sdc。如果執行個體類型不支援多重執行個體儲存體磁碟區，這將沒有效果。如果執行個體類型支援 NVMe 執行個體儲存體磁碟區，會自動列舉這些磁碟區並包含在執行個體區塊型儲存設備映射中，且無法覆寫。

更新執行中執行個體的區塊型儲存裝置對應

您不需要為了變更屬性而停止執行個體。

AWS CLI

若要更新執行中執行個體的區塊型儲存設備映射

使用 modify-instance-attribute 命令。

新增 --block-device-mappings 選項：

```
--block-device-mappings file://mapping.json
```

在 mapping.json 中，指定更新。例如，下列更新會將根磁碟區變更為持續存在。

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

若要更新執行中執行個體的區塊型儲存設備映射

使用 Edit-EC2InstanceAttribute cmdlet。

新增 -BlockDeviceMapping 選項：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定更新。例如，下列更新會將根磁碟區變更為持續存在。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

檢視執行個體區塊型儲存裝置對應的 EBS 磁碟區

您可輕鬆列舉映射至執行個體的 EBS 磁碟區。

Console

若要檢視執行個體的 EBS 磁碟區

1. 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並查看儲存體索引標籤中顯示的詳細資訊。根磁碟區至少具有以下資訊 (其中根裝置一詞等同於根磁碟區)：

   - 根裝置類型 (例如： EBS)
   - Root device name (根裝置名稱) (例如：/dev/xvda)
   - 區塊型儲存設備 (例如：/dev/xvda、/dev/sdf 和 /dev/sdj)

   如果執行個體是使用區塊型儲存裝置對應的其他 EBS 磁碟區來啟動，則其會顯示在 Block devices (區塊型儲存裝置)。任何執行個體存放區的磁碟區都不會顯示在此索引標籤上。
4. 若要顯示 EBS 磁碟區的其他資訊，請選擇其磁碟區 ID 以移至磁碟區頁面。

AWS CLI

若要檢視執行個體的 EBS 磁碟區

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[*].Instances[0].BlockDeviceMappings
```

PowerShell

若要檢視執行個體的 EBS 磁碟區

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-0bac57d7472c89bac).Instances.BlockDeviceMappings
```

檢視執行個體區塊型儲存裝置對應的執行個體存放區的磁碟區

執行個體類型會決定執行個體可用執行個體儲存體磁碟區的數量和類型。如果區塊型設備映射中的執行個體存放磁碟區的數量，超過執行個體可用的執行個體存放磁碟區，請忽略額外的磁碟區。若要檢視執行個體的執行個體儲存體磁碟區，請執行 lsblk命令 (Linux 執行個體) 或開啟 Windows Disk Management (Windows 執行個體)。若要了每個解執行個體類型支援多少執行個體儲存體磁碟區，請參閱 Amazon EC2 執行個體類型規格。

檢視執行個體的區塊型設備映射時，您只會看見 EBS 磁碟區，而非執行個體存放磁碟區。您用來檢視執行個體的執行個體存放區磁碟區的方法取決於磁碟區類型。

NVMe 執行個體存放區磁碟區

Linux 執行個體

您可以使用 NVMe 命令列套件 (nvme-cli) 來查詢區塊型設備映射中的 NVMe 執行個體存放區磁碟區。在您的執行個體上下載並安裝套件，然後執行下列命令。

```
[ec2-user ~]$ sudo nvme list
```

以下是執行個體的範例輸出。Model (模型) 欄中的文字指出磁碟區是 EBS 磁碟區還是執行個體存放區磁碟區。在本範例中，/dev/nvme1n1 和 /dev/nvme2n1 是執行個體存放區磁碟區。

```
Node             SN                   Model
 Namespace
---------------- -------------------- ----------------------------------------
 ---------
/dev/nvme0n1     vol06afc3f8715b7a597 Amazon Elastic Block Store               1

/dev/nvme1n1     AWS2C1436F5159EB6614 Amazon EC2 NVMe Instance Storage         1

/dev/nvme2n1     AWSB1F4FF0C0A6C281EA Amazon EC2 NVMe Instance Storage         1
  ...
```

Windows 執行個體

您可以使用磁碟管理或 PowerShell 同時列出 EBS 和執行個體儲存體 NVMe 磁碟區。如需詳細資訊，請參閱the section called “將 NVME 磁碟映射至磁碟區”。

HDD 或 SSD 執行個體存放區磁碟區

您可以使用執行個體中繼資料來查詢區塊型設備映射中的 HDD 或 SSD 執行個體存放區磁碟區。不包含 NVMe 執行個體存放磁碟區。

執行個體中繼資料所有請求的基底 URI 為 `http://169.254.169.254/latest/`。如需詳細資訊，請參閱使用執行個體中繼資料管理您的 EC2 執行個體。

Linux 執行個體

首先，連接至執行中的執行個體。從執行個體使用此查詢，取得其區塊型設備映射。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

回應將包含執行個體的區塊型儲存設備名稱。例如，執行個體存放區支援的 `m1.small` 執行個體的輸出看起來如下。

```
ami
ephemeral0
root
swap
```

`ami` 設備是執行個體看到的根磁碟區。執行個體存放磁碟區將命名為 `ephemeral[0-23]`。`swap` 設備用於分頁檔。如果您也映射了 EBS 磁碟區，其將顯示為 `ebs1`、`ebs2` 等等。

若要取得區塊型設備映射中個別區塊型儲存設備的詳細資訊，請將其名稱附加到先前的查詢，如下所示。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

Windows 執行個體

首先，連接至執行中的執行個體。從執行個體使用此查詢，取得其區塊型設備映射。

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/
```

回應將包含執行個體的區塊型儲存設備名稱。例如，執行個體存放區支援的 m1.small 執行個體的輸出看起來如下。

```
ami
ephemeral0
root
swap
```

ami 設備是執行個體看到的根磁碟區。執行個體存放磁碟區將命名為 ephemeral[0-23]。swap 設備用於分頁檔。如果您也映射了 EBS 磁碟區，其將顯示為 ebs1、ebs2 等等。

若要取得區塊型設備映射中個別區塊型儲存設備的詳細資訊，請將其名稱附加到先前的查詢，如下所示。

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

如何為 Amazon EC2 Windows 執行個體連接和映射磁碟區

> **Note**
>
> 本主題僅適用於 Windows 執行個體。

Windows 執行個體隨附 EBS 磁碟區，此磁碟區可做為根磁碟區使用。如果您的 Windows 執行個體使用 AWS PV 或 Citrix PV 驅動程式，您可以選擇新增最多 25 個磁碟區，總共 26 個磁碟區。如需詳細資訊，請參閱Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。

取決於執行個體的類型，執行個體將可使用 0 到 24 個可能的執行個體存放磁碟區。若要使用執行個體可用的任何執行個體存放磁碟區，您必須在建立 AMI 或啟動執行個體時指定這些磁碟區。您也可以在建立 AMI 或啟動執行個體時新增 EBS 磁碟區，或是在執行個體執行時連結這些磁碟區。

將磁碟區新增到執行個體時，您會指定 Amazon EC2 所使用的裝置名稱。如需詳細資訊，請參閱 Amazon EC2 執行個體上磁碟區的裝置名稱。 AWS Windows Amazon Machine Image (AMI) 包含一組 Amazon EC2 會使用的驅動程式，可用來將執行個體存放區和 EBS 磁碟區對應到 Windows 磁碟和磁碟編號。

將磁碟映射至 EBS 磁碟區的方法

- 將 Amazon EC2 Windows 執行個體上的 NVMe 磁碟映射至磁碟區
- 將 Amazon EC2 Windows 執行個體上的非 NVMe 磁碟映射至磁碟區

將 Amazon EC2 Windows 執行個體上的 NVMe 磁碟映射至磁碟區

若使用 Nitro 型執行個體，EBS 磁碟區會公開為 NVMe 裝置。本主題會說明如何檢視執行個體上 Windows 作業系統可用的 NVMe 磁碟。其也會顯示如何將這些 NVMe 磁碟映射至基礎 Amazon EBS 磁碟區，以及為 Amazon EC2 使用的區塊型儲存設備映射指定的裝置名稱。

主題

- 列出 NVMe 磁碟
- 將 NVMe 磁碟映射至磁碟區

列出 NVMe 磁碟

您可以使用磁碟管理公用程式或 Powershell，來搜尋 Windows 執行個體上的磁碟。

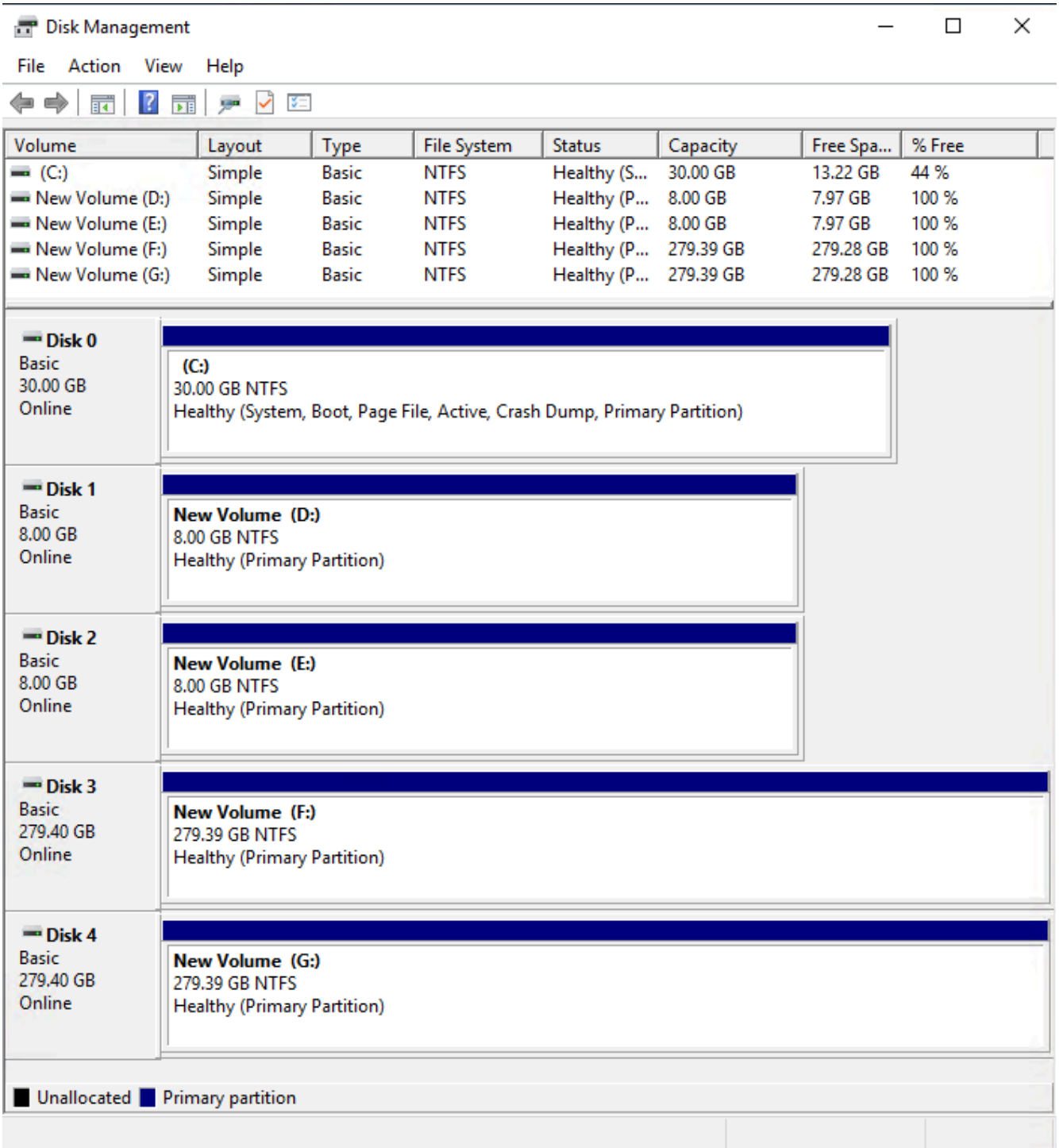
Disk Management

搜尋 Windows 執行個體上的磁碟

1. 使用遠端桌面登入 Windows 執行個體。如需詳細資訊，請參閱 使用 RDP 連線至您的 Windows 執行個體。

2. 啟動磁碟管理公用程式。
3. 檢閱磁碟。根磁碟區是掛載為 `C:\` 的 EBS 磁碟區。如果沒有顯示其他的磁碟，表示當您建立 AMI 或啟動執行個體時，並未指定其他的磁碟區。

   下列範例顯示當您啟動具有兩個額外 EBS 磁碟區的 `r5d.4xlarge` 執行個體時，可以使用的磁碟。



PowerShell

下列的 PowerShell 指令碼，會列出每個磁碟及其對應的裝置名稱和磁碟區。此指令碼適用於與 Nitro 型執行個體一起使用，其會使用 NVMe EBS 和執行個體儲存體磁碟區。

連線到您的 Windows 執行個體並執行下列命令，讓 PowerShell 指令碼可執行。

```
Set-ExecutionPolicy RemoteSigned
```

複製下列指令碼，並在您的 Windows 執行個體上將其儲存為 `mapping.ps1`。

```
# List the disks for NVMe volumes

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function GetEBSVolumeId {
    param($Path)
    $SerialNumber = (Get-Disk -Path $Path).SerialNumber
    if($SerialNumber -clike 'vol*'){
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("vol","vol-")
    }
    else {
       $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("AWS","AWS-")
    }
    return $EbsVolumeId
}

function GetDeviceName{
    param($EbsVolumeId)
    if($EbsVolumeId -clike 'vol*'){

        $Device  = ((Get-EC2Volume -VolumeId $EbsVolumeId ).Attachment).Device
        $VolumeName = ""
    }
     else {
        $Device = "Ephemeral"
        $VolumeName = "Temporary Storage"
    }
    Return $Device,$VolumeName
}
```

```
function GetDriveLetter{
    param($Path)
    $DiskNumber =  (Get-Disk -Path $Path).Number
    if($DiskNumber -eq 0){
        $VirtualDevice = "root"
        $DriveLetter = "C"
        $PartitionNumber = (Get-Partition -DriveLetter C).PartitionNumber
    }
    else
    {
        $VirtualDevice = "N/A"
        $DriveLetter = (Get-Partition -DiskNumber $DiskNumber).DriveLetter
        if(!$DriveLetter)
        {
            $DriveLetter = ((Get-Partition -DiskId $Path).AccessPaths).Split(",")[0]
        }
        $PartitionNumber = (Get-Partition -DiskId $Path).PartitionNumber
    }

    return $DriveLetter,$VirtualDevice,$PartitionNumber

}

$Report = @()
foreach($Path in (Get-Disk).Path)
{
    $Disk_ID = ( Get-Partition -DiskId $Path).DiskId
    $Disk = ( Get-Disk -Path $Path).Number
    $EbsVolumeId  = GetEBSVolumeId($Path)
    $Size =(Get-Disk -Path $Path).Size
    $DriveLetter,$VirtualDevice, $Partition = (GetDriveLetter($Path))
    $Device,$VolumeName = GetDeviceName($EbsVolumeId)
    $Disk = New-Object PSObject -Property @{
      Disk          = $Disk
      Partitions    = $Partition
      DriveLetter   = $DriveLetter
      EbsVolumeId   = $EbsVolumeId
      Device        = $Device
      VirtualDevice = $VirtualDevice
      VolumeName= $VolumeName
    }
 $Report += $Disk
}
```

```
$Report | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, VolumeName
```

執行指令碼，如下所示：

```
PS C:\> .\mapping.ps1
```

以下是具有根磁碟區、兩個 EBS 磁碟區和兩個執行個體存放區磁碟區之執行個體的範例輸出。

```
Disk Partitions DriveLetter EbsVolumeId           Device    VirtualDevice VolumeName
---- ---------- ----------- -----------           ------    ------------- ----------
   0          1 C           vol-03683f1d861744bc7 /dev/sda1 root
   1          1 D           vol-082b07051043174b9 xvdb      N/A
   2          1 E           vol-0a4064b39e5f534a2 xvdc      N/A
   3          1 F           AWS-6AAD8C2AEEE1193F0 Ephemeral N/A           Temporary
 Storage
   4          1 G           AWS-13E7299C2BD031A28 Ephemeral N/A           Temporary
 Storage
```

如果您沒有在 Windows 執行個體上設定適用於 Windows PowerShell 的 工具，指令碼就無法取得 EBS 磁碟區 ID，且會在 `EbsVolumeId` 欄中使用 N/A。

將 NVMe 磁碟映射至磁碟區

可以使用 Get-Disk 命令，將 Windows 磁碟編號映射至 Amazon EBS 磁碟區和 Amazon EC2 執行個體儲存體磁碟區。

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                   HealthStatus
 OperationalStatus      Total Size Partition

                      Style
------ ------------- -------------                   ------------
 -----------------      ---------- ----------
3      NVMe Amazo... AWS6AAD8C2AEEE1193F0_00000001.   Healthy              Online
              279.4 GB MBR
4      NVMe Amazo... AWS13E7299C2BD031A28_00000001.   Healthy              Online
              279.4 GB MBR
2      NVMe Amazo... vol0a4064b39e5f534a2_00000001.   Healthy              Online
                  8 GB MBR
```

```
0      NVMe Amazo... vol03683f1d861744bc7_00000001.   Healthy              Online
                30 GB MBR
1      NVMe Amazo... vol082b07051043174b9_00000001.   Healthy              Online
                 8 GB MBR
```

您也可以執行 ebsnvme-id 命令，將 NVMe 磁碟編號映射至 EBS 磁碟區 ID 和裝置名稱。

```
PS C:\> C:\PROGRAMDATA\Amazon\Tools\ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-03683f1d861744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

將 Amazon EC2 Windows 執行個體上的非 NVMe 磁碟映射至磁碟區

對於從使用 AWS PV 或 Citrix PV 驅動程式的 Windows AMI 啟動的執行個體，您可以使用此頁面所述的關係，將 Windows 磁碟映射到您的執行個體存放區和 EBS 磁碟區。本主題會說明如何檢視執行個體上 Windows 作業系統可用的非 NVMe 磁碟。其也會顯示如何將這些非 NVMe 磁碟映射至基礎 Amazon EBS 磁碟區，以及為 Amazon EC2 使用的區塊型儲存設備映射指定的裝置名稱。

> **Note**
>
> 若您啟動執行個體，且如果 Windows AMI 使用 Red Hat PV 驅動程式，您可以更新執行個體來使用 Citrix 驅動程式。如需詳細資訊，請參閱the section called “升級 PV 驅動程式”。

主題

- 列出非 NVMe 磁碟
- 將非 NVMe 磁碟映射至磁碟區

列出非 NVMe 磁碟

您可以使用磁碟管理公用程式或 PowerShell，來尋找 Windows 執行個體上的磁碟。

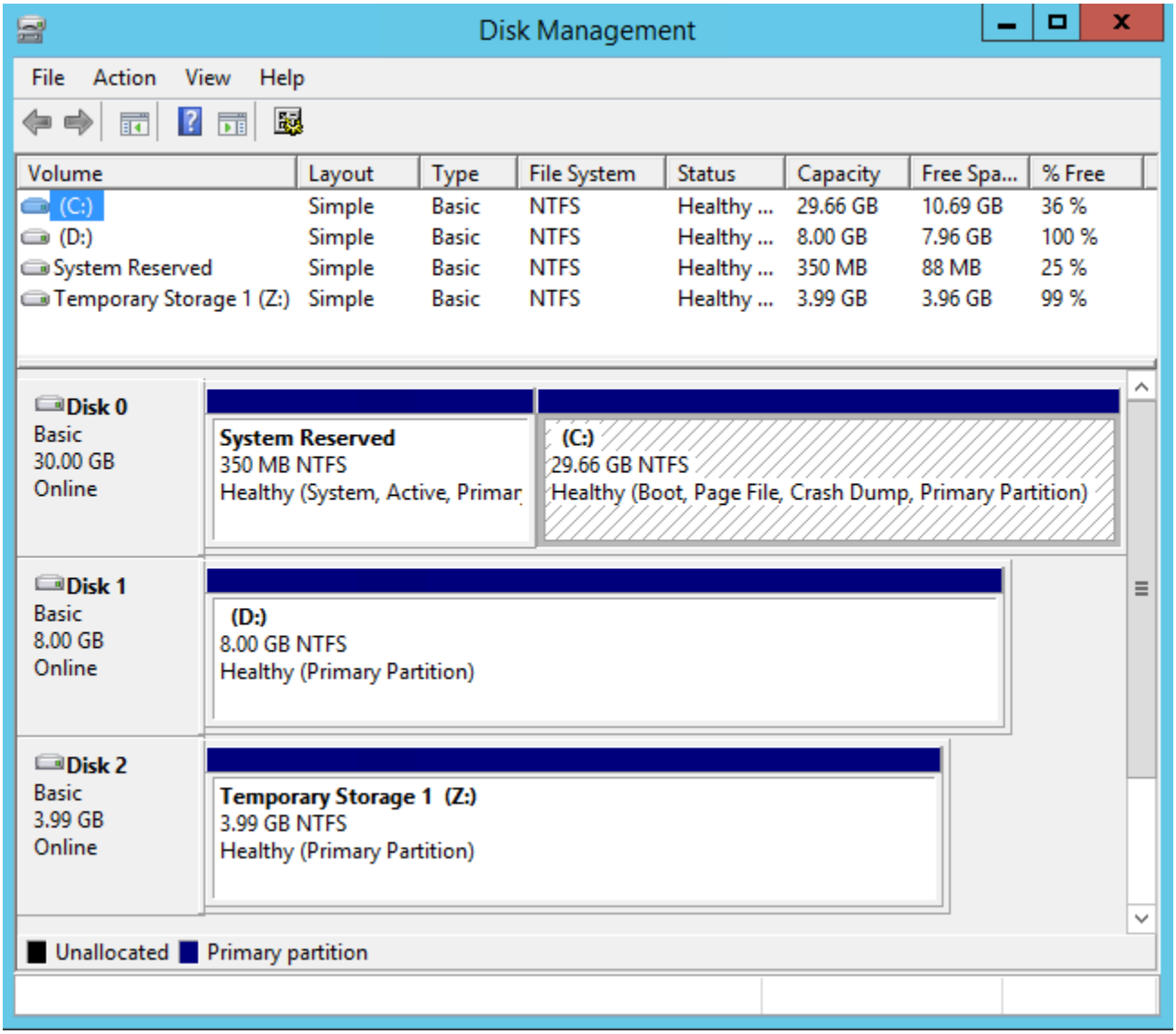
Disk Management

搜尋 Windows 執行個體上的磁碟

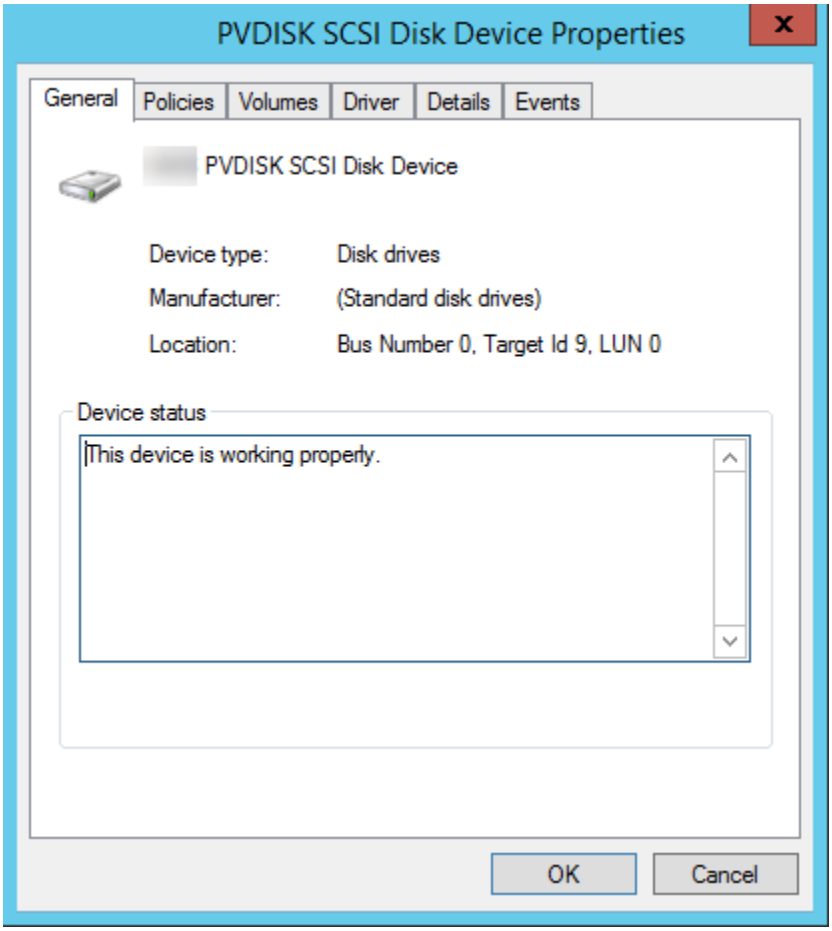
1. 使用遠端桌面登入 Windows 執行個體。如需詳細資訊，請參閱 使用 RDP 連線至您的 Windows 執行個體。
2. 啟動磁碟管理公用程式。

   在任務列上，開啟 Windows 標誌的內容 （按一下滑鼠右鍵） 選單，然後選擇磁碟管理。
3. 檢閱磁碟。根磁碟區是掛載為 C:\ 的 EBS 磁碟區。如果沒有顯示其他的磁碟，表示當您建立 AMI 或啟動執行個體時，並未指定其他的磁碟區。

   下列的範例，會顯示當您使用執行個體存放磁碟區 (磁碟 2) 和其他的 EBS 磁碟區 (磁碟 1) 來啟動 m3.medium 執行個體時，可以使用的磁碟。



4. 開啟標示為磁碟 1 之灰色窗格的內容（按一下滑鼠右鍵）選單，然後選取屬性。記下 Location (位置) 的值，然後在將非 NVMe 磁碟映射至磁碟區的表格中查閱。例如，下列磁碟的位置為匯流排編號 0，目標 Id 9，LUN 0。根據 EBS 磁碟區的對應表，此位置的裝置名稱為 `xvdj`。



PowerShell

下列的 PowerShell 指令碼，會列出每個磁碟及其對應的裝置名稱和磁碟區。

要求與限制

- 需要 Windows Server 2012 或更新版本。
- 需要登入資料才能取得 EBS 磁碟區 ID。您可以使用 Tools for PowerShell 或將 IAM 角色連接到執行個體來設定描述檔。
- 不支援 NVMe 磁碟區。
- 不支援動態磁碟。

連線到您的 Windows 執行個體並執行下列命令，讓 PowerShell 指令碼可執行。

```
Set-ExecutionPolicy RemoteSigned
```

複製下列指令碼，並在您的 Windows 執行個體上將其儲存為 mapping.ps1。

```
# List the disks
function Convert-SCSITargetIdToDeviceName {
  param([int]$SCSITargetId)
  If ($SCSITargetId -eq 0) {
    return "sda1"
  }
  $deviceName = "xvd"
  If ($SCSITargetId -gt 25) {
    $deviceName += [char](0x60 + [int]($SCSITargetId / 26))
  }
  $deviceName += [char](0x61 + $SCSITargetId % 26)
  return $deviceName
}

[string[]]$array1 = @()
[string[]]$array2 = @()
[string[]]$array3 = @()
[string[]]$array4 = @()

Get-WmiObject Win32_Volume | Select-Object Name, DeviceID | ForEach-Object {
  $array1 += $_.Name
  $array2 += $_.DeviceID
}

$i = 0
While ($i -ne ($array2.Count)) {
  $array3 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).SerialNumber) -replace "_[^ ]*$" -replace "vol", "vol-"
  $array4 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).FriendlyName)
  $i ++
}

[array[]]$array = $array1, $array2, $array3, $array4

Try {
  $InstanceId = Get-EC2InstanceMetadata -Category "InstanceId"
  $Region = Get-EC2InstanceMetadata -Category "Region" | Select-Object -
ExpandProperty SystemName
}
Catch {
```

```
  Write-Host "Could not access the instance Metadata using AWS Get-
EC2InstanceMetadata CMDLet.
Verify you have AWSPowershell SDK version '3.1.73.0' or greater installed and
 Metadata is enabled for this instance." -ForegroundColor Yellow
}
Try {
  $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
 $InstanceId).Instances.BlockDeviceMappings
  $VirtualDeviceMap = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping").GetEnumerator() | Where-Object { $_.Key -ne "ami" }
}
Catch {
  Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys or assigned an IAM role with adequate
 permissions." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
  $DriveLetter = $null
  $VolumeName = $null
  $VirtualDevice = $null
  $DeviceName = $_.FriendlyName

  $DiskDrive = $_
  $Disk = $_.Number
  $Partitions = $_.NumberOfPartitions
  $EbsVolumeID = $_.SerialNumber -replace "_[^ ]*$" -replace "vol", "vol-"
  if ($Partitions -ge 1) {
    $PartitionsData = Get-Partition -DiskId $_.Path
    $DriveLetter = $PartitionsData.DriveLetter | Where-object { $_ -notin @("",
 $null) }
    $VolumeName = (Get-PSDrive | Where-Object { $_.Name -in
 @($DriveLetter) }).Description | Where-object { $_ -notin @("", $null) }
  }
  If ($DiskDrive.path -like "*PROD_PVDISK*") {
    $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -Class
 Win32_Diskdrive | Where-Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" +
 $DiskDrive.Number) }).SCSITargetId)
    $BlockDeviceName = "/dev/" + $BlockDeviceName
    $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*"
 + $_.DeviceName + "*" }
    $EbsVolumeID = $BlockDevice.Ebs.VolumeId
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
```

```
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
    $BlockDeviceName = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping")."ephemeral$((Get-WmiObject -Class Win32_Diskdrive | Where-
Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" + $DiskDrive.Number) }).SCSIPort - 2)"
    $BlockDevice = $null
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  ElseIf ($DiskDrive.path -like "*NETAPP*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $EbsVolumeID = "FSxN Volume"
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  Else {
    $BlockDeviceName = $null
    $BlockDevice = $null
  }
```

```
  New-Object PSObject -Property @{
    Disk          = $Disk;
    Partitions    = $Partitions;
    DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
    EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
    Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else
 { $BlockDeviceName };
    VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
    VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
    DeviceName    = If ($DeviceName -eq $null) { "N/A" } Else { $DeviceName };
  }
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, DeviceName, VolumeName
```

執行指令碼，如下所示：

```
PS C:\> .\mapping.ps1
```

下列為範例輸出。

```
Disk  Partitions  DriveLetter   EbsVolumeId             Device      VirtualDevice
 DeviceName              VolumeName
----  ----------  -----------   -----------             ------      -------------
 ----------              ----------
   0           1            C   vol-0561f1783298efedd   /dev/sda1   N/A
 NVMe Amazon Elastic B   N/A
   1           1            D   vol-002a9488504c5e35a   xvdb        N/A
 NVMe Amazon Elastic B   N/A
   2           1            E   vol-0de9d46fcc907925d   xvdc        N/A
 NVMe Amazon Elastic B   N/A
```

如果您未在 Windows 執行個體上提供登入資料，則指令碼無法取得 EBS 磁碟區 ID，並會在 EbsVolumeId 欄中使用 N/A。

將非 NVMe 磁碟映射至磁碟區

執行個體的區塊型儲存設備驅動程式，會在掛載磁碟區時指派實際的磁碟區名稱。

映射項目

- 執行個體儲存體磁碟區

- EBS 磁碟區

執行個體儲存體磁碟區

下表說明 Citrix PV 和 AWS PV 驅動程式如何將非 NVMe 執行個體存放區磁碟區映射至 Windows 磁碟區。可用執行個體存放磁碟區的數量，會取決於執行個體的類型。如需詳細資訊，請參閱 EC2 執行個體的執行個體儲存體磁碟區限制。

位置	裝置名稱
匯流排編號 0，目標 ID 78，LUN 0	xvdca
匯流排編號 0，目標 ID 79，LUN 0	xvdcb
匯流排編號 0，目標 ID 80，LUN 0	xvdcc
匯流排編號 0，目標 ID 81，LUN 0	xvdcd
匯流排編號 0，目標 ID 82，LUN 0	xvdce
匯流排編號 0，目標 ID 83，LUN 0	xvdcf
匯流排編號 0，目標 ID 84，LUN 0	xvdcg
匯流排編號 0，目標 ID 85，LUN 0	xvdch
匯流排編號 0，目標 ID 86，LUN 0	xvdci
匯流排編號 0，目標 ID 87，LUN 0	xvdcj
匯流排編號 0，目標 ID 88，LUN 0	xvdck
匯流排編號 0，目標 ID 89，LUN 0	xvdcl

EBS 磁碟區

下表說明 Citrix PV 和 AWS PV 驅動程式如何將非 NVMe EBS 磁碟區映射至 Windows 磁碟區。

Location	裝置名稱
匯流排編號 0，目標 ID 0，LUN 0	/dev/sda1
匯流排編號 0，目標 ID 1，LUN 0	xvdb
匯流排編號 0，目標 ID 2，LUN 0	xvdc
匯流排編號 0，目標 ID 3，LUN 0	xvdd
匯流排編號 0，目標 ID 4，LUN 0	xvde
匯流排編號 0，目標 ID 5，LUN 0	xvdf
匯流排編號 0，目標 ID 6，LUN 0	xvdg
匯流排編號 0，目標 ID 7，LUN 0	xvdh
匯流排編號 0，目標 ID 8，LUN 0	xvdi
匯流排編號 0，目標 ID 9，LUN 0	xvdj
匯流排編號 0，目標 ID 10，LUN 0	xvdk
匯流排編號 0，目標 ID 11，LUN 0	xvdl
匯流排編號 0，目標 ID 12，LUN 0	xvdm
匯流排編號 0，目標 ID 13，LUN 0	xvdn
匯流排編號 0，目標 ID 14，LUN 0	xvdo
匯流排編號 0，目標 ID 15，LUN 0	xvdp
匯流排編號 0，目標 ID 16，LUN 0	xvdq
匯流排編號 0，目標 ID 17，LUN 0	xvdr
匯流排編號 0，目標 ID 18，LUN 0	xvds
匯流排編號 0，目標 ID 19，LUN 0	xvdt

Location	裝置名稱
匯流排編號 0，目標 ID 20，LUN 0	xvdu
匯流排編號 0，目標 ID 21，LUN 0	xvdv
匯流排編號 0，目標 ID 22，LUN 0	xvdw
匯流排編號 0，目標 ID 23，LUN 0	xvdx
匯流排編號 0，目標 ID 24，LUN 0	xvdy
匯流排編號 0，目標 ID 25，LUN 0	xvdz

在 Amazon EC2 Linux 執行個體上預防未完成的寫入

> **Note**
>
> 僅 Linux 執行個體支援預防未完成的寫入。

預防未完成寫入是 設計的區塊儲存功能 AWS ，可改善 I/O 密集型關聯式資料庫工作負載的效能，並減少延遲，而不會對資料恢復能力造成負面影響。使用 InnoDB 或 XtraDB 作為資料庫引擎的關聯式資料庫，例如 MySQL 和 MariaDB，將受益於預防未完成的寫入。

一般而言，使用大於儲存裝置電源故障不可分割性分頁的關聯式資料庫會使用資料記錄機制來預防未完成的寫入。MariaDB 和 MySQL 使用雙寫緩衝區檔案將資料寫入資料表之前記錄資料。在寫入不完整或未完成的情況下，由於作業系統當機或寫入交易期間斷電，資料庫可以從雙寫緩衝區復原資料。與寫入雙寫緩衝區相關聯的額外 I/O 負荷會影響資料庫效能和應用程式延遲，並減少每秒可處理的交易數量。有關雙寫緩衝區的詳細資訊，請參閱 MariaDB 和 MySQL 文件。

透過預防未完成的寫入，資料會以全有或全無寫入交易的方式寫入儲存區，因此無需使用雙寫緩衝區。這可避免在寫入交易期間發生作業系統當機或斷電時，將部分或未完成的資料寫入儲存區。每秒處理的交易數量最多可增加 30%，且寫入延遲最多可減少 50%，並且不會影響工作負載的彈性。

定價

使用預防未完成的寫入不會產生額外成本。

目錄

- 在 Amazon EC2 上預防未完成的寫入之區塊大小
- 在 Amazon EC2 上使用預防未完成寫入的需求
- 檢查 Amazon EC2 執行個體類型對預防未完成寫入之支援
- 在 Amazon EC2 上設定您的工作負載，以預防未完成的寫入

在 Amazon EC2 上預防未完成的寫入之區塊大小

預防未完成的寫入支援 4 KiB、8 KiB 和 16 KiB 資料區塊的寫入作業。資料區塊起始邏輯區塊定址 (LBA) 必須與 4 KiB、8 KiB 或 16 KiB 的各自區塊邊界大小相符。例如，對於 16 KiB 寫入作業，資料區塊起始 LBA 必須與 16 KiB 的區塊邊界大小相符。

下表顯示各種儲存體和執行個體類型的支援。

	4 KiB 區塊	8 KiB 區塊	16 KiB 區塊
執行個體儲存體磁碟區	所有 NVMe 執行個體儲存體磁碟區都會連接至新一代的 I 系列執行個體。	AWS Nitro SSD 支援的 I4i、Im4gn、Is4gen、I7i、I7ie、I8g 和 I8ge 執行個體。	
Amazon EBS 磁碟區	連接到Nitro 型執行個體的所有 Amazon EBS 磁碟區。		

若要確認您的執行個體和磁碟區是否支援預防未完成的寫入，請查詢執行個體是否支援預防未完成的寫入和其他詳細資訊，例如支援的區塊和邊界大小。如需詳細資訊，請參閱檢查 Amazon EC2 執行個體類型對預防未完成寫入之支援。

在 Amazon EC2 上使用預防未完成寫入的需求

若要預防未完成的寫入功能正常運作，I/O 作業必須符合 NTWPU、NTWGU、NTWBU 欄位中指定的大小、對齊和邊界要求。您必須設定作業系統，以確保特定的儲存子系統 (檔案系統、LVM、RAID 等) 在提交至裝置之前，不會修改儲存體堆疊下的 I/O 屬性，包括區塊合併、分割或區塊定址重新定位。

已透過下列組態測試預防未完成的寫入：

- 支援所需區塊大小的執行個體類型和儲存體類型。

- 核心版本 5.10 或更高版本的 Amazon Linux 2。
- 已啟用 `bigalloc` 且叢集大小為 16 KiB 的 ext4，以及最新的 ext4 公用程式 (e2fsprogs 1.46.5 或更高版本)。
- `O_DIRECT` 檔案存取模式繞過 Linux 核心緩衝區快取。

> **Note**
>
> 您不需要停用 MySQL 和 MariaDB 工作負載的 I/O 合併。

檢查 Amazon EC2 執行個體類型對預防未完成寫入之支援

若要確認您的執行個體和磁碟區是否支援預防未完成的寫入，以及檢視包含預防未完成寫入資訊的 NVMe 命名空間廠商特定資料，請使用下列命令。

```
$ sudo nvme id-ns -v device_name
```

> **Note**
>
> 該命令以十六進位傳回供應商特定的資訊，並帶有 ASCII 解釋。您可能需要在可讀取和剖析輸出的應用程式中建置類似於 `ebsnvme-id` 的工具。

例如，下列命令會傳回 NVMe 命名空間廠商特定的資料，其中包含 的破解寫入預防資訊`/dev/nvme1n1`。

```
$ sudo nvme id-ns -v /dev/nvme1n1
```

如果您的執行個體和磁碟區支援預防未完成的寫入，它會在 NVMe 命名空間廠商特定的資料中傳回下列 AWS 未完成的寫入預防資訊。

> **Note**
>
> 下表中的位元組代表 NVMe命名空間廠商特定資料開頭的位元組位移。

位元組	說明
0:31	例如，裝置配件掛載點的名稱，例如 /dev/xvda 。您可以在磁碟區附件請求期間提供此功能，且 Amazon EC2 執行個體可使用其來建立 NVMe 區塊型儲存裝置 (nvmeXn1) 的符號連結。
32:63	磁碟區 ID。例如 vol01234567890abcdef 。此欄位可用於將 NVMe 裝置對應至連接的磁碟區。
64:255	保留以供日後使用。
256:257	命名空間預防未完成的寫入單位大小 (NTWPU)。此欄位指出在停電或錯誤情況下保證自動寫入 NVM 之寫入作業的命名空間特定大小。此欄位邏輯區塊 (用以零為基礎的值表示) 中指定。
258:259	命名空間預防未完成的寫入精密度 (NTWPG)。此欄位指出在停電或錯誤情況下保證自動寫入 NVM 的寫入作業以下 NTWPU 的命名空間特定大小增量。也就是說，大小應為 NTWPG * n <= NTWPU，其中 n 是正整數。寫入作業 LBA 位移也必須與此欄位相符。此欄位邏輯區塊 (用以零為基礎的值表示) 中指定。
260:263	命名空間預防未完成的寫入邊界大小 (NTWPB)。此欄位指出此 NTWPU 值的不可分割邊界大小。在停電或錯誤情況下，不能保證向此命名空間寫入跨越不可分割邊界的內容會自動寫入 NVM。0h 的值表示停電或錯誤情況沒有不可分割邊界。所有其他值會使用與 NTWPU 欄位相同的編碼方式，以邏輯區塊來指定大小。

在 Amazon EC2 上設定您的工作負載，以預防未完成的寫入

支援的執行個體類型 (含支援的磁碟區) 預設啟用預防未完成的寫入。您不需要設定任何其他設定來啟用預防未完成的寫入。

> **Note**
>
> 不支援預防未完成寫入的工作負載不會影響效能。您不需要對這些工作負載進行任何變更。支援預防未完成寫入但未設定為使用的工作負載，請繼續使用雙寫入緩衝區，而不會獲得任何效能優勢。

若要設定 MySQL 或 MariaDB 軟體堆疊以停用雙寫緩衝區，並使用預防未完成的寫入，請完成以下步驟：

1. 將磁碟區設定為搭配 BigAlloc 選項使用 ext4 檔案系統，並將叢集大小設定為 4 KiB、8 KiB 或 16 KiB。搭配使用 BigAlloc 與大小為 4 KiB、8 KiB 或 16 KiB 的叢集，可確保檔案系統分配檔案與各自邊界相符。

   ```
   $  mkfs.ext4 -O bigalloc -C 4096|8192|16384 device_name
   ```

   > **Note**
   >
   > 對於 MySQL 和 MariaDB，您必須使用 `-C 16384` 來比對資料庫頁面大小。將分配精密度設定為頁面大小倍數以外的值，可能會導致分配與儲存裝置的預防未完成的寫入邊界不符。

   例如：

   ```
   $  mkfs.ext4 -O bigalloc -C 16384 /dev/nvme1n1
   ```

2. 將 InnoDB 設定為使用 `O_DIRECT` 排清方法並關閉 InnoDB 雙寫。使用您偏好的文字編輯器開啟 `/etc/my.cnf`，並更新 `innodb_flush_method` 和 `innodb_doublewrite` 參數，如下所示：

   ```
   innodb_flush_method=O_DIRECT
   innodb_doublewrite=0
   ```

> **Important**
>
> 如果您使用的是邏輯磁碟區管理工具 (LVM) 或其他儲存虛擬化層，請確定磁碟區的起始位移與 16 KiB 倍數相符。這與基礎 NVMe 儲存體相關，可用來說明儲存虛擬化層所使用的中繼資料標題和超級區塊。如果您新增 LVM 實體磁碟區的位移，可能會導致檔案系統分配與 NVMe 裝置的位移不相符，而這會使預防未完成的寫入失效。如需詳細資訊，請參閱 Linux 手冊頁面中的 `--dataalignmentoffset`。

基於 Windows VSS 的應用程式一致性 Amazon EBS 快照

使用 AWS Systems Manager 執行命令，可對所有連接至 Amazon EC2 Windows 執行個體的 Amazon EBS 磁碟區，擷取應用程式一致性快照。快照程序會使用 Windows 磁碟區陰影複製服務 (VSS) 來取得 VSS 感知應用程式的 EBS 磁碟區層級備份。快照包括來自這些應用程式和磁碟之間擱置中交易的資料。若需要備份所有連接的磁碟區，您不需要關閉執行個體或中斷其連結。

使用基於 VSS 的 EBS 快照無需額外付費。您只需為備份程序所建立的 EBS 快照支付費用。如需詳細資訊，請參閱 Amazon EBS 快照收費方式為何？

> **Note**
>
> 基於 Windows VSS 的應用程式一致性快照僅受 Windows 執行個體支援。

目錄

- 什麼是 IVS？
- 基於 VSS 的 Amazon EBS 快照解決方案運作方式
- 建立基於 Windows VSS 的 EBS 快照的先決條件
- 為您的 EC2 Windows 執行個體建立 VSS 型 EBS 快照
- 疑難排解基於 Windows VSS 的 EBS 快照
- 使用 AWS VSS 解決方案還原執行個體的資料
- AWS VSS 解決方案版本歷史記錄

什麼是 IVS？

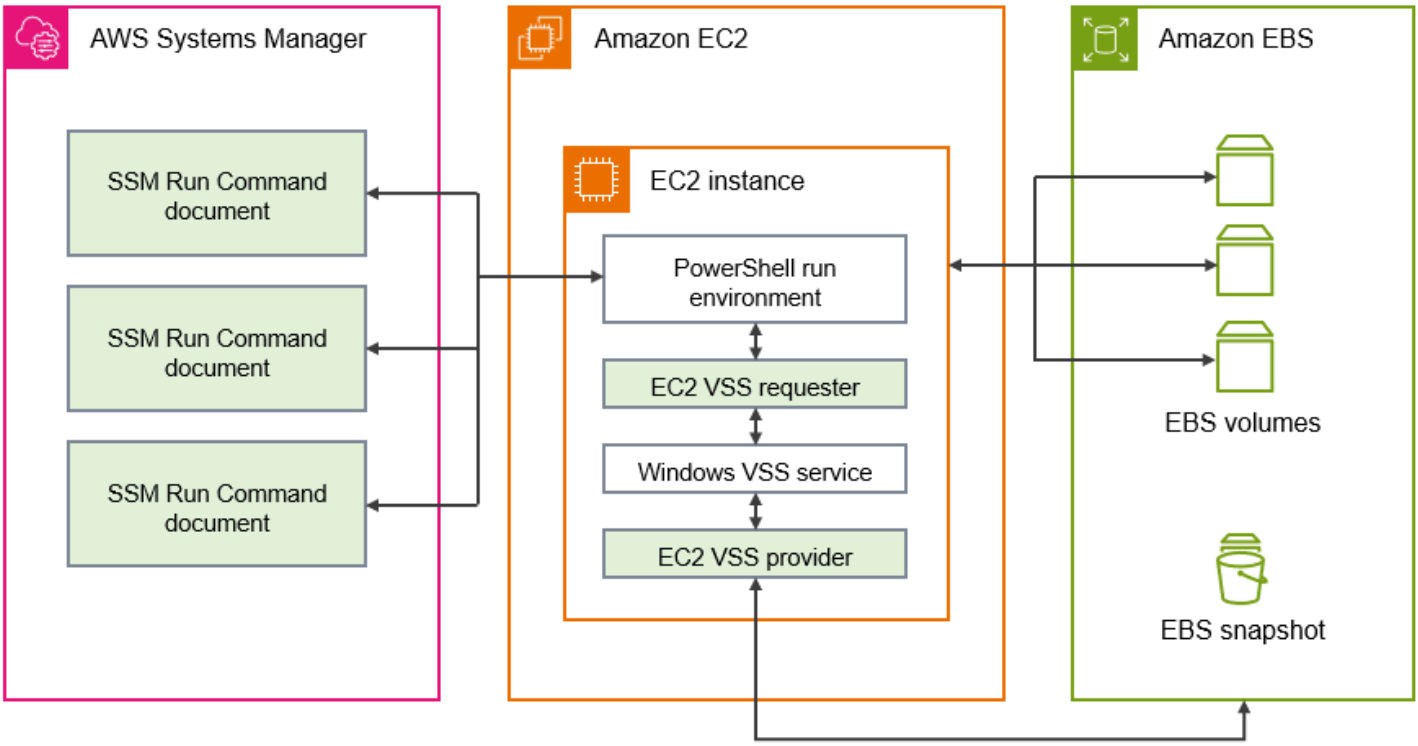
磁碟區陰影複製服務 (VSS) 是一種包含在 Microsoft Windows 中的備份和復原技術。當正在使用電腦檔案或磁碟區時，其可建立其備份複本或快照。如需詳細資訊，請參閱磁碟區陰影複製服務。

若要建立應用程式一致性快照，會涉及以下軟體元件。

- VSS 服務 – Windows 作業系統的一部分
- VSS 請求程式 – 請求建立陰影複製的軟體
- VSS 寫入程式 – 通常是作為應用程式 (例如 SQL Server) 的一部分提供，以確保要備份的資料集一致

- VSS 供應程式 – 建立基礎磁碟區陰影複本的元件

基於 Windows VSS 的 Amazon EBS 快照解決方案包含多個可協助建立備份的 Systems Manager (SSM) Run Command 文件，以及一個稱為 `AwsVssComponents` 的 Systems Manager Distributor 套件，其中包括 EC2 VSS 請求程式和 EC2 VSS 供應程式。此 `AwsVssComponents` 套件必須安裝在 EC2 Windows 執行個體上，才能擷取 EBS 磁碟區的應用程式一致性快照。下圖說明這些軟體元件之間的關係。



基於 VSS 的 Amazon EBS 快照解決方案運作方式

用於讓應用程式一致，且基於 VSS 的 EBS 快照指令碼的程序包含下列步驟。

1. 完成建立基於 Windows VSS 的 EBS 快照的先決條件。
2. 輸入 `AWSEC2-VssInstallAndSnapshot` SSM 文件的參數，然後使用執行命令來執行此文件。如需詳細資訊，請參閱執行 AWSEC2-VssInstallAndSnapshot 命令文件 (建議採納)。
3. 執行個體上的 Windows VSS 服務會針對執行中的應用程式，協調所有正在進行的 I/O 操作。
4. 系統會排清所有 I/O 緩衝區，並暫時暫停所有 I/O 操作。暫停時間最多持續 10 秒。
5. 在暫停期間，系統會為連接至執行個體的所有磁碟區建立快照。
6. 系統會取消暫停，並恢復 I/O 操作。

7. 系統會將所有新建立的快照新增至 EBS 快照清單。此程序成功建立基於 VSS 的 EBS 快照時，系統一律會為其加上 AppConsistent:true 標記。
8. 如果您需要從快照還原，則可採用從快照建立磁碟區的標準 EBS 程序。或者，您能夠透過範例指令碼將所有磁碟區還原至執行個體 (如 使用 AWS VSS 解決方案還原執行個體的資料 中所述)。

建立基於 Windows VSS 的 EBS 快照的先決條件

您可以使用 Systems Manager Run Command AWS Backup或 Amazon Data Lifecycle Manager 建立 VSS 型 EBS 快照。以下先決條件適用所有解決方案。

系統要求

確保您的 EC2 Windows 執行個體符合建立 VSS 型快照的所有系統需求，包括支援的 Windows 作業系統、.NET 架構 AWS Tools for Windows PowerShell、PowerShell 和 AWS Systems Manager Agent 版本。

IAM 許可

附加到 Amazon EC2 Windows 執行個體的 IAM 角色必須擁有許可，才能使用 VSS 建立應用程式一致性快照。若要授予必要許可，您可以將 `AWSEC2VssSnapshotPolicy` 受管政策附加到執行個體設定檔。

VSS 元件

若要在 Windows 作業系統上建立應用程式一致快照，則必須在執行個體上安裝 `AwsVssComponents` 套件。該套件包含執行個體上的 EC2 VSS 代理程式，它可作為 EBS 磁碟區的 VSS 請求程式和 EC2 VSS 供應程式。

系統要求

安裝 Systems Manager Agent

VSS 由 Systems Manager Agent 使用 PowerShell 進行協調。請確保已在 EC2 執行個體上安裝 SSM Agent `3.0.502.0` 版或更高版本。如果已在使用舊版 SSM Agent，則可利用 Run Comman 進行更新。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的針對 Amazon EC2 執行個體設定 Systems Manager 以及在適用於 Windows Server 的 Amazon EC2 執行個體上使用 SSM Agent。

Amazon EC2 Windows 執行個體需求

執行 Windows Server 2016 和更新版本的執行個體皆支援基於 VSS 的 EBS 快照。

.NET Framework 版本

AwsVssComponents 套件需要 .NET Framework 版本 4.6 或更高版本。Windows Server 2016 之前的 Windows 作業系統版本預設為 .NET Framework 的舊版本。如果您的執行個體使用舊版 .NET Framework，則必須使用 Windows 更新安裝版本 4.6 或更高版本。

AWS Tools for Windows PowerShell 版本

確保您的執行個體執行 AWS Tools for Windows PowerShell 版本 3.3.48.0或更新版本。若要查看版本，請在執行個體的 PowerShell 終端中執行以下命令。

```
C:\> Get-AWSPowerShellVersion
```

如果您需要 AWS Tools for Windows PowerShell 更新執行個體，請參閱AWS Tools for PowerShell《 使用者指南》中的安裝 AWS Tools for Windows PowerShell 。

Windows PowerShell 版本

請確保執行個體正在執行 Windows PowerShell 主要版本 3、4 或 5。若要查看版本，請在執行個體的 PowerShell 終端中執行以下命令。

```
C:\> $PSVersionTable.PSVersion
```

PowerShell 語言模式

請確定您的執行個體已將 PowerShell 語言模式設定為 FullLanguage。如需詳細資訊，請參閱 Microsoft 文件中的 about_Language_Modes。

使用 IAM 受管政策，授予 VSS 型快照

AWSEC2VssSnapshotPolicy 受管政策讓 Systems Manager 可在您的 Windows 執行個體上執行以下動作：

- 建立和標記 EBS 快照
- 建立和標記 Amazon Machine Image (AMI)
- 將中繼資料 (例如裝置 ID) 連接至 VSS 建立的預設快照標籤。

本主題涵蓋 VSS 受管政策的許可詳細資訊，以及如何將其連接至您的 EC2 執行個體設定檔 IAM 角色。

目錄

- AWSEC2VssSnapshotPolicy 受管政策詳細資訊
- 將 VSS 快照受管政策連接至執行個體設定檔角色

AWSEC2VssSnapshotPolicy 受管政策詳細資訊

AWS 受管政策是 Amazon 為客戶提供的獨立政策 AWS 。 AWS 受管政策旨在授予常見使用案例的許可。您無法變更 AWS 受管政策中定義的許可。然而，您可以複製政策，並將使用其作為使用案例特定客戶管理政策的基準。

如需 AWS 受管政策的詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

若要使用 AWSEC2VssSnapshotPolicy 受管政策，您可將其連接至已連接至 EC2 Windows 執行個體的 IAM 角色。此政策讓 EC2 VSS 解決方案可建立標籤，並將標籤新增至 Amazon Machine Image (AMI) 和 EBS 快照。若要連接政策，請參閱 將 VSS 快照受管政策連接至執行個體設定檔角色。

AWSEC2VssSnapshotPolicy 授予的許可

AWSEC2VssSnapshotPolicy 政策包含以下 Amazon EC2 許可，以允許 Amazon EC2 代表您建立和管理 VSS 快照。您可將此受管政策連接至用於 EC2 Windows 執行個體的 IAM 執行個體設定檔角色。

- ec2:CreateTags – 將標籤新增至 EBS 快照和 AMIs，協助識別和分類資源。
- ec2:DescribeInstanceAttribute – 擷取連接至目標執行個體的 EBS 磁碟區和對應之區塊型儲存設備映射。
- ec2:CreateSnapshots – 建立 EBS 磁碟區的快照。
- ec2:CreateImage – 從執行中的 EC2 執行個體建立 AMI。
- ec2:DescribeImages – 擷取 EC2 AMI 和快照的資訊。
- ec2:DescribeSnapshots – 判斷快照的建立時間和狀態，以驗證應用程式一致性。

> **Note**
>
> 若要檢視此政策的許可，請參閱「AWS 受管政策參考」中的 AWSEC2VssSnapshotPolicy。

簡化特定使用案例的許可 - 進階

`AWSEC2VssSnapshotPolicy` 受管政策會包含您可建立 VSS 型快照之所有方式的許可。您可以建立僅包含您所需許可的自訂政策。

使用案例：建立 AMI、使用案例：使用 AWS Backup 服務

如果您只使用 CreateAmi選項，或者您只透過 AWS Backup 服務建立 VSS 型快照，則可以簡化政策陳述式，如下所示。

- 省略由以下陳述式 ID (SID) 識別的政策陳述式：
  - CreateSnapshotsWithTag
  - CreateSnapshotsAccessInstance
  - CreateSnapshotsAccessVolume
- 調整 CreateTagsOnResourceCreation 陳述式，如下所示：
  - 從資源中移除 arn:aws:ec2:*:*:snapshot/* 標籤。
  - 從 ec2:CreateAction 條件中移除 CreateSnapshots。
- 調整 CreateTagsAfterResourceCreation 陳述式，以從資源中移除 arn:aws:ec2:*:*:snapshot/*。
- 調整 DescribeImagesAndSnapshots 陳述式，以從陳述式動作中移除 ec2:DescribeSnapshots。

使用案例：僅限快照

若您不使用 CreateAmi 選項，則可簡化政策陳述式，如下所示。

- 省略由以下陳述式 ID (SID) 識別的政策陳述式：
  - CreateImageAccessInstance
  - CreateImageWithTag
- 調整 CreateTagsOnResourceCreation 陳述式，如下所示：
  - 從資源中移除 arn:aws:ec2:*:*:image/* 標籤。
  - 從 ec2:CreateAction 條件中移除 CreateImage。
- 調整 CreateTagsAfterResourceCreation 陳述式，以從資源中移除 arn:aws:ec2:*:*:image/*。
- 調整 DescribeImagesAndSnapshots 陳述式，以從陳述式動作中移除 ec2:DescribeImages。

> **Note**
>
> 為了確保自訂政策會如預期執行，建議您定期檢閱並納入受管政策的更新內容。

將 VSS 快照受管政策連接至執行個體設定檔角色

若要授予 VSS 型快照之許可給 EC2 Windows 執行個體，您可將 AWSEC2VssSnapshotPolicy 受管政策連接至執行個體設定檔角色，如下所示。請務必確保您的執行個體符合所有 系統要求。

> **Note**
>
> 若要使用受管政策，您的執行個體必須安裝 `AwsVssComponents` 套件版本 `2.3.1` 或更新版本。如需版本歷史記錄，請參閱 AwsVssComponents 套件版本。

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色以查看您可存取的 IAM 角色清單。
3. 選取連接至執行個體之角色的角色名稱連結。這會開啟角色詳細資訊頁面。
4. 若要連接受管政策，請選擇位於清單面板右上角的新增許可。接著，請在下拉式清單中，選擇連接政策。
5. 若要簡化結果，請在搜尋列中輸入政策名稱 (`AWSEC2VssSnapshotPolicy`)。
6. 選取要連接之政策名稱旁的核取方塊，然後選擇新增許可。

管理 Windows VSS 型 EBS 快照的 VSS 元件套件

在建立 VSS 型 EBS 快照之前，請確定您的 Windows 執行個體已安裝最新版本的 VSS 元件套件。有幾種方式可以將 `AwsVssComponents` 套件安裝到現有執行個體上，如下所示：

- (建議) 執行 AWSEC2-VssInstallAndSnapshot 命令文件 (建議採納)。這會在每次執行時視需要自動安裝或更新。
- 在 EC2 Windows 執行個體上手動安裝 VSS 元件.
- 更新 EC2 Windows 執行個體上的 VSS 元件套件.

您也可以使用 EC2 Image Builder 建立 AMI，以使用 `aws-vss-components-windows` 受管元件安裝映像的 `AwsVssComponents` 套件。受管元件使用 AWS Systems Manager Distributor 來安裝

套件。Image Builder 建立映像後，您從相關聯 AMI 啟動的每個執行個體都會安裝 VSS 套件。如需如何在安裝 VSS 套件的情況下建立 AMI 的詳細資訊，請參閱 EC2 Image Builder 使用者指南中的 Windows 的 Distributor 套件受管元件。

目錄

- 在 EC2 Windows 執行個體上手動安裝 VSS 元件
- 更新 EC2 Windows 執行個體上的 VSS 元件套件

在 EC2 Windows 執行個體上手動安裝 VSS 元件

您的 EC2 Windows 執行個體須已安裝 VSS 元件，您才能使用 Systems Manager 建立應用程式一致快照。如果您不執行 `AWSEC2-VssInstallAndSnapshot` 命令文件，以便在每次建立應用程式一致快照時自動安裝或更新套件，則必須手動安裝套件。

如果您打算使用以下其中一種方法，從 EC2 執行個體建立應用程式一致快照，也必須手動安裝。

- 使用 建立 VSS 快照 AWS Backup
- 使用 Amazon Data Lifecycle Manager 建立 VSS 快照

如果您需要執行手動安裝，我們建議您使用最新的 AWS VSS 元件套件來改善 EC2 Windows 執行個體上應用程式一致性快照的可靠性和效能。

> **Note**
>
> 若要在建立應用程式一致快照時自動安裝或更新 `AwsVssComponents` 套件，建議您使用 Systems Manager 來執行 `AWSEC2-VssInstallAndSnapshot` 文件。如需詳細資訊，請參閱執行 AWSEC2-VssInstallAndSnapshot 命令文件 (建議採納)。

若要在 Amazon EC2 Windows 執行個體上安裝 VSS 元件，請依照適用您偏好之環境的步驟執行。

Console

使用 SSM Distributor 安裝 VSS 元件

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇 執行命令。

3. 選擇 執行命令。
4. 在 Command document (命令文件) 清單中，選擇 AWS-ConfigureAWSPackage 旁的按鈕。
5. 在 Command parameters (命令參數) 中，執行以下操作：
   a. 驗證 Action (動作) 設為 Install (安裝)。
   b. 對於名稱，輸入 `AwsVssComponents`。
   c. 在版本中，輸入版本或將欄位留空，以便 Systems Manager 安裝最新版本。
6. 在 Targets (目標) 中，手動指定標籤或選取執行個體，以識別您要執行這項操作的執行個體。

   > **Note**
   >
   > 如果您選取手動選取執行個體，但您預期看到的執行個體並未出現在清單上，請參閱 https://docs.aws.amazon.com/systems-manager/latest/userguide/troubleshooting-remote-commands.html#where-are-instances 使用者指南中的AWS Systems Manager 我的執行個體在哪裡？以取得故障診斷秘訣。

7. 對於 Other parameters (其他參數)：
   - (選用) 在 Comment (註解) 中，輸入此命令的相關資訊。
   - 在 Timeout (seconds) (逾時 (秒)) 中，指定在命令執行全面失敗之前，系統要等候的秒數。
8. (選用) 在 Rate control (速率控制) 中：
   - 在 Concurrency (並行) 中，指定可同時執行命令的執行個體數目或百分比。

     > **Note**
     >
     > 如果您已透過選取 Amazon EC2 標籤來選取目標，而且不確定有多少執行個體會使用所選的標籤，請指定百分比來限制可同時執行文件的執行個體數目。

   - 在 Error threshold (錯誤閾值) 中，指定在特定數目或百分比之執行個體上的命令失敗之後，停止在其他執行個體上執行命令。例如，如果您指定三個錯誤，則 Systems Manager 會在收到第四個錯誤時停止傳送命令。仍在處理命令的執行個體也可能會傳送錯誤。
9. (選用) 在 Output options (輸出選項) 區段中，如果您想要將命令輸出儲存至檔案，請選取 Enable writing to an S3 bucket (啟用寫入至 S3 儲存貯體) 旁的按鈕。指定儲存貯體和 (選用) 字首 (資料夾) 名稱。

> **Note**
>
> 授予能力以將資料寫入至 S3 儲存貯體的 S3 許可，會是指派給執行個體之執行個體描述檔的許可，而不是執行此工作之使用者的許可。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的設定 EC2 執行個體許可。

10. (選用) 指定 SNS notifications (SNS 通知) 的選項。

    如需針對執行命令設定 Amazon SNS 通知的資訊，請參閱針對 AWS Systems Manager設定 Amazon SNS 通知。

11. 選擇執行。

AWS CLI

透過以下程序，即可使用 AwsVssComponents 中的執行命令來下載 AWS CLI套件，並在執行個體上進行安裝。該套件安裝兩個元件：VSS 請求程式和 VSS 供應程式。系統會將這些元件複製到執行個體上的目錄，接著註冊供應程式 DLL，以將其做為 VSS 供應程式。

若要安裝 VSS 套件

執行下列命令，藉此下載並安裝 Systems Manager 所需的 VSS 元件。

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
```

PowerShell

透過以下程序，即可使用 Tools for Windows PowerShell 中的執行命令來下載 AwsVssComponents 套件，並在執行個體上進行安裝。該套件安裝兩個元件：VSS 請求程式和 VSS 供應程式。系統會將這些元件複製到執行個體上的目錄，接著註冊供應程式 DLL，以將其做為 VSS 供應程式。

若要安裝 VSS 套件

執行下列命令，藉此下載並安裝 Systems Manager 所需的 VSS 元件。

```
Send-SSMCommand `
```

```
    -DocumentName "AWS-ConfigureAWSPackage" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'action'='Install';'name'='AwsVssComponents'}
```

驗證 AWS VSS 元件上的簽章

請使用下列程序驗證 AwsVssComponents 套件上的簽章。

1. 連接至 Windows 執行個體。如需詳細資訊，請參閱使用 RDP 連線至您的 Windows 執行個體。
2. 導覽至 C:\Program Files\Amazon\AwsVssComponents。
3. 開啟 ec2-vss-agent.exe 的內容選單 (按一下右鍵)，然後選擇屬性。
4. 導覽至數位簽章索引標籤，並確認簽署者的名稱是否為 Amazon Web Services Inc.
5. 使用上述步驟驗證 Ec2VssInstaller 和 Ec2VssProvider.dll 上的簽章。

更新 EC2 Windows 執行個體上的 VSS 元件套件

建議維持將 VSS 元件更新為最新的建議版本。在新版 AwsVssComponents 套件發行時，您可以透過多種不同方式更新元件。

更新方法

- 您可以在發行新版本的 AWS VSS 元件在 EC2 Windows 執行個體上手動安裝 VSS 元件時重複中所述的步驟。
- 您可以將 Systems Manager State Manager 關聯設定為在 AwsVssComponents 套件可用時自動下載並安裝全新或更新後的 VSS 元件。
- 您可以在使用 Systems Manager 執行 AWSEC2-VssInstallAndSnapshot 文件時，隨時在建立應用程式一致快照時自動安裝或更新 AwsVssComponents 套件。

> **Note**
>
> 建議您使用 Systems Manager 執行 AwsVssComponents 命令文件，其會在建立應用程式一致快照時，自動安裝或更新 AWSEC2-VssInstallAndSnapshot 套件。如需詳細資訊，請參閱執行 AWSEC2-VssInstallAndSnapshot 命令文件 (建議採納)。

若要建立 Systems Manager State Manager 關聯，請依照適用於您偏好之環境的步驟執行。

Console

當您建立 Systems Manager State Manager 關聯時，有兩種更新 `AwsVssComponents` 套件的選項，如下所示：

解除安裝並重新安裝

此方法會下載並安裝套件，無需額外的先決條件。

就地更新

這會執行套件的就地更新，並具有下列先決條件：

- 在執行個體上安裝的 SSM Agent 版本必須是版本 `3.3.808.0` 或更新版本。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的在適用於 Windows Server 的 EC2 執行個體上使用 SSM Agent。
- 如果已指定，則 `AwsVssComponents` 套件版本必須是版本 `2.5.0` 或更新版本。早期版本不支援就地更新。

> **Note**
>
> 如果執行個體不符合這些先決條件，就地更新將會失敗。請改用解除安裝並重新安裝選項。

若要建立 State Manager 關聯

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇 State Manager (狀態管理員)。

   或者，如果 Systems Manager 首頁先開啟，請開啟導覽窗格，然後選擇 State Manager。
3. 選擇 Create association (建立關聯)。
4. 在 Name (名稱) 欄位中，輸入描述性名稱。
5. 在文件清單中，選擇 AWS-ConfigureAWSPackage。
6. 在 Parameters (參數) 區段中，從 Action (動作) 清單中選擇 Install (安裝)。
7. 針對安裝類型，選擇解除安裝並重新安裝或就地更新。
8. 在 Name (名稱) 欄位中，輸入 `AwsVssComponents`。可將 Version (版本) 和 Additional Arguments (其他引數) 保留為空白。

9. 在 Targets (目標) 區段中，選擇一個選項。

> **Note**
>
> 如果您選擇使用標籤將執行個體設為目標，且指定映射到 Linux 執行個體的標籤，則關聯會在 Windows 執行個體上執行成功，但在 Linux 執行個體上執行失敗。關聯的整體狀態會顯示為 Failed (失敗)。

10. 在 Specify schedule (指定排程) 區段中，選擇一個選項。
11. 在 Advanced options (進階選項) 區段中，針對 Compliance severity (合規嚴重性)，選擇關聯的嚴重性等級。如需詳細資訊，請參閱了解關聯合規性。針對變更行事曆，請選取預先設定的變更行事曆。如需詳細資訊，請參閱 AWS Systems Manager 變更行事曆。
12. 針對比率控制，請執行下列操作：
    - 在並行中，指定可同時執行命令的受管節點數目或百分比。
    - 在 Error threshold (錯誤閾值) 中，指定在特定數目或百分比之節點上的命令失敗之後，停止在其他受管節點上執行命令。
13. (選用) 針對輸出選項，若要將命令輸出儲存至檔案，請選取啟用將輸出寫入 S3。在方塊中輸入儲存貯體和字首 (資料夾) 名稱。
14. 選擇 Create association (建立關聯)，接著選擇 Close (關閉)。系統會嘗試在執行個體上建立關聯，並立即套用狀態。

> **Note**
>
> 如果 Windows Server 的 EC2 執行個體顯示失敗狀態，請確認 SSM 代理程式正在執行個體上執行，並確認執行個體已設定 Systems Manager 的 AWS Identity and Access Management (IAM) 角色。如需詳細資訊，請參閱設定 AWS Systems Manager。

AWS CLI

使用 create-association 命令，以依照排程更新 Distributor 套件，而無需讓相關聯的應用程式離線。只會取代套件中的新檔案或更新檔案。

若要建立 State Manager 關聯

執行下列命令以建立關聯。--name 值 (文件名稱) 一律為 AWS-ConfigureAWSPackage。以下命令會使用索引鍵 InstanceIds 來指定目標執行個體。

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters action=Install,installationType="Uninstall and
 reinstall",name=AwsVssComponents \
    --targets Key=InstanceIds,Values=i-1234567890abcdef0,i-000011112222abcde
```

PowerShell

若要建立 State Manager 關聯

使用 New-SSMAssociation cmdlet。

```
New-SSMAssociation `
    -Name "AWS-ConfigureAWSPackage" `
    -Parameter  @{
        "action" = "Install"
        "installationType" = "Uninstall and reinstall"
        "name" = "AwsVssComponents"
    } `
    -Target @{
        "Key" = "InstanceIds"
        "Values" = @("i-1234567890abcdef0", "i-000011112222abcde")
    }
```

為您的 EC2 Windows 執行個體建立 VSS 型 EBS 快照

在您符合所有 建立基於 Windows VSS 的 EBS 快照的先決條件 後，便可使用以下任一方法，從 EC2 執行個體建立 VSS 型快照。

AWS Systems Manager 命令文件

使用 Systems Manager 命令文件 以建立 VSS 型快照。

若要自動化備份，您可以建立使用 AWSEC2-VssInstallAndSnapshot命令文件的 AWS Systems Manager 維護時段任務。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 使用維護時段 Windows (主控台)。

AWS Backup

您可以在主控台或 CLI 中啟用 VSS， AWS Backup 以在使用 時建立 VSS 備份。如需詳細資訊，請參閱「AWS Backup 開發人員指南」中的建立 Windows VSS 備份。

> **Note**
>
> AWS Backup 不會在您的執行個體上自動安裝AwsVssComponents套件。您必須在執行個體上執行手動安裝。如需詳細資訊，請參閱在 EC2 Windows 執行個體上手動安裝 VSS 元件。

Amazon Data Lifecycle Manager

您可以在快照生命週期政策中啟用前置和後置指令碼，以使用 Amazon Data Lifecycle Manager 建立 VSS 快照。如需詳細資訊，請參閱「Amazon EBS 使用者指南」中的自動化應用程式一致性快照。

> **Note**
>
> Amazon Data Lifecycle Manager 不會在您的執行個體上自動安裝 AwsVssComponents 套件。您必須在執行個體上執行手動安裝。如需詳細資訊，請參閱在 EC2 Windows 執行個體上手動安裝 VSS 元件。

您可以使用 Systems Manager 命令文件來建立 VSS 型快照

您可以使用 AWS Systems Manager 命令文件來建立 VSS 型快照。下列內容會介紹可用的命令文件，以及文件用來建立快照的執行期參數。

在您使用任何 Systems Manager 命令文件之前，請確保您已符合所有 建立基於 Windows VSS 的 EBS 快照的先決條件。

主題

- Systems Manager VSS 快照文件的參數
- 執行 Systems Manager VSS 快照命令文件

Systems Manager VSS 快照文件的參數

建立 VSS 快照的 Systems Manager 文件全部都會使用下列參數，除非另有說明：

AmiName (字串，選用)

如果 CreateAmi 選項設定為 `True`，請指定備份所建立之 AMI 的名稱。

description (字串，選用)

指定此程序所建立之快照或映像的描述。

CollectDiagnosticLogs (字串，選用)

若要在快照和 AMI 建立步驟期間收集更多資訊，請將此參數設定為「`True`」。此參數的預設值為「`False`」。合併診斷日誌會儲存為 `.zip` 格式封存檔，並位於執行個體上的以下位置：

`C:\ProgramData\Amazon\AwsVss\Logs\`*`timestamp`*`.zip`

CopyOnly (字串，選用)

如果您除了 AWS VSS 使用原生 SQL Server 備份，執行僅限複製備份可防止 AWS VSS 中斷原生差異備份鏈。若要執行僅限複製備份操作，請將此參數設定為 `True`。

此參數的預設值為 `False`，這會導致 AWS VSS 執行完整的備份操作。

CreateAmi (字串，選用)

若要建立 VSS 型 Amazon Machine Image (AMI) 來備份您的執行個體，請將此參數設定為 `True`。此參數的預設值為 `False`，因此會改為使用 EBS 快照備份您的執行個體。

如需從執行個體建立 AMI 的詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

executionTimeout (字串，選用)

指定在執行個體上執行快照建立程序或從執行個體建立 AMI 的時間上限 (以秒為單位)。增加此逾時時間可讓命令等待更長時間，使 VSS 開始凍結並完成標記其建立的資源。此逾時僅適用於快照或 AMI 建立步驟。安裝或更新 `AwsVssComponents` 套件的初始步驟不包含在逾時中。

ExcludeBootVolume (字串，選用)

若您建立快照，此設定會從備份程序中排除開機磁碟區。若要從快照中排除開機磁碟區，請將 ExcludeBootVolume 設定為 `True`，將 CreateAmi 設定為 `False`。

如果您為備份建立 AMI，則應將此參數設定為 `False`。此參數的預設值為 `False`。

NoWriters (字串，選用)

若要從快照程序中排除應用程式 VSS 寫入程式，請將此參數設定為 True。排除應用程式 VSS 寫入程式有助於您解決與第三方 VSS 備份元件的衝突。此參數的預設值為 False。

若 SaveVssMetadata 為 True，則此參數必須設為 False。

SaveVssMetadata (字串，選用)

若要在每個快照期間儲存 VSS 中繼資料檔案，請將此參數設定為 True。預設值為 False。這些檔案有助於深入了解備份操作中包含哪些元件或寫入器，以及每個快照的相關檔案和磁碟區。使用 VSS 還原解決方案還原 SQL 資料庫時，會使用中繼資料檔案。如需如何從 VSS 快照還原 SQL 資料庫的詳細資訊，請參閱使用自動化 Runbook 從 AWS VSS 解決方案快照還原資料庫。

中繼資料檔案之名稱中會具有相關聯的快照集 id。您可以在執行個體的下列位置找到它們：

```
C:\ProgramData\Amazon\AwsVss\VssMetadata\
```

> **Warning**
>
> - 儲存 VSS 中繼資料檔案會需要 2.4.0 版或更新版本的 AwsVssComponents 套件。如果您的執行個體已安裝較舊版本，將 SaveVssMetadata 設定為 True 會導致快照建立失敗。
> - NoWriters 和 SaveVssMetadata 參數互斥。若兩者皆設為 True，則快照建立會失敗。

tags (字串，選用)

我們建議您標記快照和映像，有助於尋找和管理資源，例如從快照清單還原磁碟區。系統會新增 Name 金鑰，並加上空白值，您可在其中指定要套用至輸出快照或映像的名稱。

如果您要指定其他標籤，請在標籤之間使用分號來分隔。例如 Key=Environment,Value=Test;Key=User,Value=TestUser1。

> **Note**
>
> 標籤索引鍵和值只能包含英數字元和下列特殊字元：( ) ./\-"'@_+:={}。

根據預設，系統會為 VSS 型快照和映像新增以下預留標籤。

- 裝置 – 針對 VSS 型快照，這是快照擷取之 EBS 磁碟區的裝置名稱。
- AppConsistent – 此標籤表示成功建立 VSS 型快照或 AMI。
- AwsVssConfig – 這會識別啟用 VSS 時所建立的快照和 AMIs。標籤會包含中繼資訊，例如 `AwsVssComponents` 版本和快照集 ID。

> **Warning**
>
> 在您的參數清單中指定任何這些預留標籤，將會導致錯誤。

VssVersion (字串，選用)

在使用 `AWSEC2-VssInstallAndSnapshot` 文件的情況下，您可以指定 `VssVersion` 參數以在執行個體上安裝特定版本的 `AwsVssComponents` 套件。將此參數保留空白，以安裝建議的預設版本。

如果已安裝指定版本的 `AwsVssComponents` 套件，則指令碼會略過安裝步驟，並繼續進行備份步驟。如需 `AwsVssComponents` 套件版本和操作支援的清單，請參閱AWS VSS 解決方案版本歷史記錄。

執行 Systems Manager VSS 快照命令文件

您可以使用 AWS Systems Manager 命令文件建立 VSS 型 EBS 快照，如下所示。

執行 AWSEC2-VssInstallAndSnapshot 命令文件 (建議採納)

當您使用 AWS Systems Manager 執行`AWSEC2-VssInstallAndSnapshot`文件時，指令碼會執行下列步驟。

1. 該指令碼會先在執行個體上安裝或更新 `AwsVssComponents` 套件，具體取決於是否已安裝執行個體。
2. 指令碼會在完成第一個步驟後，建立應用程式一致快照。

若要執行 `AWSEC2-VssInstallAndSnapshot` 文件，請依照適用於您偏好之環境的步驟執行。

Console

從主控台建立建立 VSS 型 EBS 快照

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 從導覽窗格中選取執行命令。這會顯示您帳戶中目前正在執行的命令清單 (若適用)。
3. 選擇執行命令。這會開啟您具有存取權限的命令文件清單。
4. 從命令文件清單中選取 AWSEC2-VssInstallAndSnapshot。若要簡化結果，可輸入全部或部分文件名稱。您也可以依擁有者、平台類型或標籤進行篩選。

   在您選取命令文件時，系統會將詳細資訊填入清單下方。
5. 從文件版本清單中選取 Default version at runtime。
6. 設定命令參數，以定義 AWSEC2-VssInstallAndSnapshot 如何安裝 AwsVssComponents 套件並使用 VSS 快照或 AMI 來備份。如需參數詳細資訊，請參閱 Systems Manager VSS 快照文件的參數。
7. 針對目標選取，手動指定標籤或選取執行個體，以識別要執行這項操作的執行個體。

   > **Note**
   >
   > 如果您手動選取執行個體，但預期看到的執行個體並未出現在清單中，請參閱我的執行個體在哪裡？，以取得疑難排解秘訣。
8. 如需定義 Systems Manager Run Command 行為的其他參數 (例如速率控制)，請依照從主控台執行命令中的說明輸入值。
9. 選擇 Run (執行)。

   如果操作成功，該命令就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標籤，或是搜尋 AppConsistent，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視 Systems Manager 命令輸出的詳細資訊，以了解執行失敗的原因。若系統成功完成命令，但無法備份特定磁碟區，則您可以在 EBS 磁碟區清單中進行疑難排解。

AWS CLI

您可以在 中執行下列命令 AWS CLI ，以建立 VSS 型 EBS 快照，並取得快照建立的狀態。

建立 VSS 型 EBS 快照

執行下列命令，藉此建立 VSS 型 EBS 快照。若要建立快照，您必須使用 `--instance-ids` 參數來識別執行個體。如需您可以查看之其他參數的詳細資訊，請參閱 Systems Manager VSS 快照文件的參數。

```
aws ssm send-command \
 --document-name "AWSEC2-VssInstallAndSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"],"VssVersion":[""]}'
```

如果操作成功，該命令文件就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標記，或是搜尋 AppConsistent，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視命令輸出的詳細資訊，以了解執行失敗的原因。

取得命令狀態

若要取得快照目前的狀態，請使用從 send-command 傳回的命令 ID 執行下列命令。

```
aws ssm get-command-invocation
 --instance-ids "i-01234567890abcdef" \
 --command-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
 --plugin-name "CreateVssSnapshot"
```

PowerShell

使用 執行下列命令 AWS Tools for Windows PowerShell ，以建立 VSS 型 EBS 快照，並取得目前建立輸出的執行時間狀態。指定前面清單中描述的參數，以修改快照處理程序的行為。

使用適用於 Windows PowerShell 的工具建立 VSS 型 EBS 快照

執行下列命令，以建立 VSS 型 EBS 快照或 AMI。

```
Send-SSMCommand -DocumentName "AWSEC2-VssInstallAndSnapshot" -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value';'VssVersion'=''}
```

取得命令狀態

若要取得快照目前的狀態，請使用從 Send-SSMCommand 傳回的命令 ID 執行下列命令。

```
Get-SSMCommandInvocationDetail -InstanceId "i-01234567890abcdef" -CommandId
 "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" -PluginName "CreateVssSnapshot"
```

如果操作成功，該命令就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標記，或是搜尋 AppConsistent，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視命令輸出的詳細資訊，以了解執行失敗的原因。

執行 AWSEC2-CreateVssSnapshot 命令文件

若要執行 AWSEC2-CreateVssSnapshot 文件，請依照適用於您偏好之環境的步驟執行。

Console

從主控台建立建立 VSS 型 EBS 快照

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 從導覽窗格中選取執行命令。這會顯示您帳戶中目前正在執行的命令清單 (若適用)。
3. 選擇執行命令。這會開啟您具有存取權限的命令文件清單。
4. 從命令文件清單中選取 AWSEC2-CreateVssSnapshot。若要簡化結果，可輸入全部或部分文件名稱。您也可以依擁有者、平台類型或標籤進行篩選。

   在您選取命令文件時，系統會將詳細資訊填入清單下方。
5. 從文件版本清單中選取 Default version at runtime。
6. 設定命令參數，以定義如何使用 VSS 快照或 AMI 來備份 AWSEC2-CreateVssSnapshot。如需參數詳細資訊，請參閱 Systems Manager VSS 快照文件的參數。
7. 針對目標選取，手動指定標籤或選取執行個體，以識別要執行這項操作的執行個體。

   > **Note**
   >
   > 如果您手動選取執行個體，但預期看到的執行個體並未出現在清單中，請參閱我的執行個體在哪裡？，以取得疑難排解秘訣。
8. 如需定義 Systems Manager Run Command 行為的其他參數 (例如速率控制)，請依照從主控台執行命令中的說明輸入值。
9. 選擇 Run (執行)。

如果操作成功，該命令就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標籤，或是搜尋 AppConsistent，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視 Systems Manager 命令輸出的詳細資訊，以了解執行失敗的原因。若系統成功完成命令，但無法備份特定磁碟區，則您可以在 EBS 磁碟區清單中進行疑難排解。

AWS CLI

您可以在 中執行下列命令 AWS CLI ，以建立 VSS 型 EBS 快照。

建立 VSS 型 EBS 快照

執行下列命令，藉此建立 VSS 型 EBS 快照。若要建立快照，您必須使用 --instance-ids 參數來識別執行個體。如需您可以查看之其他參數的詳細資訊，請參閱 Systems Manager VSS 快照文件的參數。

```
aws ssm send-command \
 --document-name "AWSEC2-CreateVssSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"]}'
```

如果操作成功，該命令文件就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標記，或是搜尋 AppConsistent，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視命令輸出的詳細資訊，以了解執行失敗的原因。

PowerShell

使用 執行下列命令 AWS Tools for Windows PowerShell ，以建立 VSS 型 EBS 快照。

使用適用於 Windows PowerShell 的工具建立 VSS 型 EBS 快照

執行下列命令，藉此建立 VSS 型 EBS 快照。若要建立快照，您必須使用 InstanceId 參數來識別執行個體。您可以指定多個執行個體，藉此為其建立快照。如需您可以查看之其他參數的詳細資訊，請參閱 Systems Manager VSS 快照文件的參數。

```
Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value'}
```

如果操作成功，該命令就會將新快照填入 EBS 快照清單中。您可以搜尋指定的標記，或是搜尋 `AppConsistent`，藉此在 EBS 快照清單中找到這些快照。如果命令執行失敗，請檢視 命令輸出的詳細資訊，以了解執行失敗的原因。若系統成功完成命令，但無法備份特定磁碟區，則您可以在 EBS 快照清單中進行疑難排解。

使用共用 EBS 儲存體，為 Windows 容錯移轉叢集執行命令文件

可以使用上一節所述的任何命令列程序來建立 VSS 型快照。命令文件 (`AWSEC2-VssInstallAndSnapshot` 或 `AWSEC2-CreateVssSnapshot`) 必須在叢集中的主節點上執行。次要節點上的文件將會失敗，因為其無法存取共用磁碟。如果您的主要和次要變更是動態變更，您可以在多個節點上執行 AWS Systems Manager Run Command 文件，預期該命令會在主要節點上成功，並在次要節點上失敗。

> **Note**
>
> 若要自動化備份，您可以建立使用 `AWSEC2-VssInstallAndSnapshot` 文件的 AWS Systems Manager 維護時段任務。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的 使用維護時段 Windows (主控台)。

疑難排解基於 Windows VSS 的 EBS 快照

在嘗試任何其他疑難排解步驟之前，建議您先確認下列詳細資料。

- 確保您符合所有 建立基於 Windows VSS 的 EBS 快照的先決條件。
- 確認您使用您作業系統適用的 `AwsVssComponents` 套件最新 支援 Windows OS 版本。您觀察到的問題可能已在較新版本中得到解決。

主題

- 檢查日誌檔
- 收集其他診斷日誌
- 在已設定代理的執行個體上使用 VSS
- 錯誤：Thaw 管道連線逾時、Thaw 發生錯誤、VSS Freeze 等候逾時或其他逾時錯誤
- 錯誤：無法叫用方法。僅在此語言模式下的核心類型上支援方法調用

檢查日誌檔

如果建立基於 VSS 的 EBS 快照時遇到問題或收到錯誤訊息，則可在 Systems Manager 主控台中檢視命令輸出。

對於建立 VSS 快照的 Systems Manager 文件，您可以在執行時期將 `CollectDiagnosticLogs` 參數設定為「`True`」。當 `CollectDiagnosticLogs` 參數設定為「`True`」時，VSS 會收集其他日誌來協助偵錯。如需詳細資訊，請參閱收集其他診斷日誌。

如果您收集診斷日誌，Systems Manager 文件會將其儲存在以下位置的執行個體：`C:\ProgramData\Amazon\AwsVss\Logs\`*`timestamp`*`.zip`。`CollectDiagnosticLogs` 參數的預設值為「`False`」。

> **Note**
>
> 如需其他偵錯說明，您可以將`.zip`檔案傳送到 支援。

無論您是否收集診斷日誌檔，都可以使用下列其他日誌檔：

- `%ProgramData%\Amazon\SSM\InstanceData\`*`InstanceID`*`\document\orchestration\`*`SSMCommandID`*`\awsrunPowerShellScript\runPowerShellScript\stdout`
- `%ProgramData%\Amazon\SSM\InstanceData\`*`InstanceID`*`\document\orchestration\`*`SSMCommandID`*`\awsrunPowerShellScript\runPowerShellScript\stderr`

您也可以開啟事件檢視器 Windows 應用程式，然後選擇 Windows 日誌 、應用程式來檢視其他日誌。若要特別查看 EC2 Windows VSS 供應程式和磁碟區陰影複製服務的事件，請依條款 **`Ec2VssSoftwareProvider`** 和 **`VSS`** 的來源進行篩選。

如果您正透過 VPC 端點使用 Systems Manager，且 Systems Manager send-command API 動作 (主控台的 Run Command) 失敗，請確認已正確設定下列端點：com.amazonaws.***`region`***.ec2。

如果沒有定義 Amazon EC2 端點，則用來列舉所連接 EBS 磁碟區的呼叫會失敗，進而導致 Systems Manager 命令失敗。如需有關使用 Systems Manager 設定 VPC 端點的詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的建立 Virtual Private Cloud 端點。

收集其他診斷日誌

若要在使用 Systems Manager 發送命令執行 VSS 快照文件時，收集其他診斷日誌，請在執行時期將 `CollectDiagnosticLogs` 輸入參數設定為「`True`」。我們建議您在疑難排解時將此參數設定為「`True`」。

若要檢視命令列範例，請選取下列其中一個索引標籤。

AWS CLI

以下範例在 AWS CLI執行 `AWSEC2-CreateVssSnapshot` Systems Manager 文件：

```
aws ssm send-command \
    --document-name "AWSEC2-CreateVssSnapshot" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"description":["Example - create diagnostic logs at
 runtime."],"tags":["Key=tag_name,Value=tag_value"],"CollectDiagnosticLogs":
["True"]}'
```

PowerShell

以下範例在 PowerShell 執行 `AWSEC2-CreateVssSnapshot` Systems Manager 文件：

```
Send-SSMCommand `
    -DocumentName "AWSEC2-CreateVssSnapshot" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'description'='Example - create diagnostic logs at
 runtime.';'tags'='Key=tag_name,Value=tag_value';'CollectDiagnosticLogs'='True'}
```

在已設定代理的執行個體上使用 VSS

如果在使用代理連線 EC2 端點的執行個體上建立基於 VSS 的 EBS 快照時發生問題，請確認執行個體的下列設定：

- 確認代理已設定，以便以 SYSTEM 身分 AWS Tools for Windows PowerShell 執行，以存取執行個體區域和 IMDS 中的 EC2 服務端點。
- 若要支援使用系統設定的 WinHTTP Proxy，請確定您已在執行個體安裝最新 `AwsVssComponents` 版本。如需設定 WinHTTP 代理的詳細資訊，請前往 Microsoft 網站參閱 Windows 超文字傳輸通訊協定 (WINHTTP) 的 Netsh 命令。

錯誤：Thaw 管道連線逾時、Thaw 發生錯誤、VSS Freeze 等候逾時或其他逾時錯誤

EC2 Windows VSS 供應程式可能會因為執行個體上的活動或服務導致基於 VSS 的 EBS 快照無法及時進行而逾時。Windows VSS 架構會提供無法設定的 10 秒時段，暫停與檔案系統的通訊。在這段期間，AWSEC2-CreateVssSnapshot 會對磁碟區執行快照。

下列問題可能造成 EC2 Windows VSS 供應程式在快照期間遇到時間限制：

- 磁碟區的過多 I/O
- 執行個體上的 EC2 API 回應緩慢
- 分段的磁碟區
- 與某些防毒軟體不相容
- VSS 應用程式寫入器的問題
- 當大量 PowerShell 模組啟用模組記錄時，可能會導致 PowerShell 指令碼執行緩慢

執行 AWSEC2-CreateVssSnapshot 命令文件時遇到的大部分逾時問題是與備份時執行個體的工作負載過高相關。下列動作可協助您成功建立快照：

- 重試 AWSEC2-CreateVssSnapshot 命令來查看快照是否成功。若在某些情況下重試成功，減少執行個體負載可能使快照更加成功。
- 等候一段時間，待執行個體上的工作負載減少，再重試 AWSEC2-CreateVssSnapshot 命令。或者，您可以在執行個體處於低壓力時嘗試快照。
- 當系統上的防毒軟體關閉時，請嘗試 VSS 快照。若這樣可解決問題，請見防毒軟體說明，並將其設定為允許 VSS 快照。
- 如果您的帳戶在執行快照的相同區域內出現大量 Amazon EC2 API 呼叫，則 API 限流可能會延遲快照作業。若要減少限流影響，請使用最新 AwsVssComponents 套件。此套件使用 EC2 CreateSnapshots API 動作來減少每個磁碟區快照建立和標記等變動動作的數量。
- 如果您同時執行多個 AWSEC2-CreateVssSnapshot 命令指令碼，您可以採取下列步驟來減少並行問題。
  - 請考慮在 API 活動較低期間安排快照時間。
  - 如果您在 Systems Manager 主控台 (或 API 中的 SendCommand) 使用 Run Command 來執行命令指令碼，您可以使用 Systems Manager 速率控制來減少並行。

    您也可以使用 Systems Manager 速率控制來減少服務並行 AWS Backup ，例如使用 Systems Manager 執行命令指令碼的服務。

- 在 Shell 中執行命令 `vssadmin list writers`，並查看是否有回報系統上任何寫入器的上次錯誤欄位中任何錯誤。如有任何寫入器回報逾時錯誤，請考慮在執行個體負載較少時重試快照。
- 當您使用較小的執行個體類型 (例如 *t2 | t3 | t3a*.nano 或 *t2 | t3 | t3a*.micro)，可能會因記憶體和 CPU 限制而發生逾時。下列動作可能有助於減少逾時問題。
  - 在建立快照之前，請嘗試關閉記憶體或 CPU 密集型應用程式。
  - 嘗試在執行個體活動較低期間建立快照。

錯誤：無法叫用方法。僅在此語言模式下的核心類型上支援方法調用

當 PowerShell 語言模式未設定為 `FullLanguage` 時，您將會遇到此錯誤。`AWSEC2-CreateVssSnapshot` SSM 文件需要將 PowerShell 設定為 `FullLanguage` 模式。

若要確認語言模式，請在 PowerShell 主控台中的執行個體上執行以下命令：

```
$ExecutionContext.SessionState.LanguageMode
```

如需語言模式的詳細資訊，請參閱 Microsoft 文件中的 about_Language_Modes。

使用 AWS VSS 解決方案還原執行個體的資料

您可以從 VSS 解決方案建立的 VSS 型快照還原 Windows 執行個體的 AWS EBS 磁碟區。如果您的 AWS VSS 解決方案快照包含 Microsoft SQL Server 資料庫的備份，您可以使用 `AWSEC2-RestoreSqlServerDatabaseWithVss` AWS Systems Manager Automation Runbook 還原資料庫。

資料庫還原執行手冊可自動執行整個還原程序，包括透過快照建立磁碟區，以及將其連接至執行個體。自動化可善用 VSS 技術來還原資料庫，從而讓您在還原時無需停止 SQL Server 應用程式，或者中斷任何作用中連線。

若要了解如何使用 Microsoft SQL Server 資料庫執行手冊的詳細說明，請參閱 Microsoft SQL Server on Amazon EC2 使用者指南中的透過基於 VSS 的快照進行還原。

自訂指令碼以從 AWS VSS 解決方案快照還原 EBS 磁碟區

您可以使用`RestoreVssSnapshotSampleScript.ps1`指令碼做為模型，建立自己的自訂指令碼，從 AWS VSS 解決方案快照還原 EBS 磁碟區。範例指令碼會執行以下任務：

- 停止執行個體
- 移除執行個體上的所有現有磁碟機 (若已排除開機磁碟區，則其將不會涵蓋在內)

- 從快照建立新的磁碟區
- 使用快照上的裝置 ID 標記來將磁碟區連接至執行個體
- 重新啟動執行個體

> **Important**
>
> 下列指令碼會分離所有連接至執行個體的磁碟區，然後從快照建立新的磁碟區。請確認您已正確地備份執行個體。系統不會刪除舊的磁碟區。如果您想刪除舊磁碟區，則可編輯指令碼。

從基於 VSS 的 EBS 快照還原磁碟區

1. 下載 RestoreVssSnapshotSampleScript.zip 檔案，並解壓縮檔案內容。
2. 在文字編輯器中開啟 `RestoreVssSnapshotSampleScript.ps1`，並編輯指令碼底端的範例呼叫，加上有效的 EC2 執行個體 ID 和 EBS 快照 ID，然後從 PowerShell 執行指令碼。

AWS VSS 解決方案版本歷史記錄

此頁面包含依 AWS VSS 元件套件版本列出的版本備註，以及每個 Windows Server 支援版本的元件和指令碼版本需求。

主題

- AwsVssComponents 套件版本
- 支援 Windows OS 版本

AwsVssComponents 套件版本

下表說明 AWS VSS 元件套件的發行版本。

版本	詳細資訊	發行日期	可下載
2.5.2	• 新增對`aws-eusc`分割區端點的支援。 • 更新 AWS SDK 以納入安全性更新。	2026 年 9 月 2 日	是

版本	詳細資訊	發行日期	可下載
2.5.1	修正了 SQL 資料庫還原在指定目標資料庫參數時可能出現故障的情況。	2025 年 3 月 13 日	是
2.5.0	• 新增了讀取 VSS 中繼資料檔案以及還原執行個體上的 Microsoft SQL Server 資料庫的功能。若要了解詳細資訊，請參閱 Microsoft SQL Server on Amazon EC2 使用者指南中的透過基於 VSS 的快照進行還原。 • 新增了在安裝或升級 `AwsVssComponents` 套件時對就地更新選項的支援。	2025 年 1 月 17 日	是
2.4.0	新增功能以在建立快照時儲存 VSS 中繼資料檔案。若要啟用此功能，請參閱 Systems Manager VSS 快照文件的參數 的 SaveVssMetadata。	2024 年 10 月 7 日	是
2.3.3	已更新 VSS 代理程式，以確保在建立快照期間使用 `Ec2VssProvider` 。	2024 年 6 月 25 日	是
2.3.2	已修正在解除安裝時未移除 VSS 提供者註冊的情況。	2024 年 5 月 9 日	是
2.3.1	新增了新的預設標籤`AwsVssConfig` ，以識別 AWS VSS 建立的快照和 AMIs。	2024 年 3 月 7 日	是

版本	詳細資訊	發行日期	可下載
2.2.1	• 已新增 `DescribeInstanceAttribute` API 使用支援。 • 錯誤修正與可靠性改進。 • 在 Windows Server 2012 和 2012 R2. AWS VSS 元件 2.2.1 版上安裝的 Windows Server 2012 和 2012 R2 已棄用支援將會失敗。 AWS VSS 元件 2.1.0 版是支援 Windows Server 2012 和 2012 R2 的最後一個版本。	2024 年 1 月 18 日	是
2.1.0	已新增 `CreateSnapshots` API 使用支援。	2023 年 11 月 6 日	是
2.0.1	新增 WinHTTP 代理設定的使用支援。	2023 年 10 月 26 日	否
2.0.0	為 AWS VSS 元件新增了建立快照和 AMIs 的功能，可相容於 PowerShell 模組記錄、指令碼區塊記錄和轉錄功能。	2023 年 4 月 28 日	否
1.3.2.0	修復了未正確報告安裝失敗的情況。	2022 年 5 月 10 日	否
1.3.1.0	• 修正快照在網域控制器上失敗且與 NTDS VSS 寫入器記錄有關的錯誤。 • 修正解除安裝 1.0 版 VSS 提供者時的 VSS 代理程式錯誤。	2020 年 2 月 6 日	是

版本	詳細資訊	發行日期	可下載
1.3.00	• 降低不需要的詳細資訊以改善記錄。 • 修正安裝期間的區域化問題。 • 修正某些提供者註冊錯誤狀況的傳回代碼。 • 修正各種安裝問題。	2019 年 3 月 19 日	否
1.2.00	• 為代理程式新增命令列參數 -nw (無寫入器) 和 -copy (僅限複製)。 • 修正記憶體配置呼叫不當所導致的 EventLog 錯誤。	2018 年 11 月 15 日	否
1.1	已修正做為預設 Windows Backup and Restore 提供者不正確使用的 AWS VSS 元件。	2017 年 12 月 12 日	否
1.0	初始版本。	2017 年 11 月 20 日	否

支援 Windows OS 版本

下表顯示您應該在 Amazon EC2 上的每個 Windows Server 版本上執行的 AWS VSS 解決方案版本。

Windows Server 版本	AwsVssComponents 版本	AWSEC2-VssInstallAndSnapshot 版本名稱	AWSEC2-CreateVssSnapshot 版本名稱
Windows Server 2025	預設	預設	預設

Windows Server 版本	AwsVssComponents 版本	AWSEC2-VssInstallAndSnapshot 版本名稱	AWSEC2-CreateVssSnapshot 版本名稱
Windows Server 2022	預設	預設	預設
Windows Server 2019	預設	預設	預設
Windows Server 2016	預設	預設	預設
Windows Server 2012 R2	2.1.0	不支援	2012R2
Windows Server 2012	2.1.0	不支援	2012R2
Windows Server 2008 R2	1.3.1.0	不支援	2008R2

Amazon EC2 上的物件儲存、檔案儲存和檔案快取

雲端檔案儲存是一種在雲端存放資料的方式，提供伺服器和應用程式透過共享檔案系統存取資料。這種相容性使得雲端檔案儲存非常適合用於倚賴共享檔案系統的工作負載，而且提供簡易的整合，無須程式碼變更。

有許多檔案儲存解決方案，包括計算執行個體上的單一節點檔案伺服器 (其使用區塊儲存作為基礎，沒有可擴展性或很少冗餘以保護資料)、自造式叢集解決方案，以及完全受管解決方案。下列內容介紹 提供的一些儲存服務 AWS ，用於 Amazon EC2 執行個體。

目錄

- 使用 Amazon S3 搭配 Amazon EC2 執行個體
- 搭配 Amazon EC2 Linux 執行個體一起使用 Amazon EFS
- 使用 Amazon FSx 搭配 Amazon EC2 執行個體
- 搭配 Amazon EC2 執行個體一起使用 Amazon File Cache

使用 Amazon S3 搭配 Amazon EC2 執行個體

Amazon Simple Storage Service (Amazon S3) 是一項物件儲存服務，提供領先業界的可擴展性、資料可用性、安全性和效能。您可以使用 Amazon S3，從 Amazon EC2 執行個體或透過網路從任何位置，存放和擷取各種使用案例的任何資料量，例如資料湖、網站、備份和大數據分析。如需詳細資訊，請參閱什麼是 Amazon S3？

有兩種方式可從 Amazon EC2 執行個體存取 Amazon S3 資料： Amazon EC2

- 檔案存取 – 使用 Amazon S3 檔案將 S3 儲存貯體掛載為執行個體上的高效能檔案系統。
- 物件存取 – 使用 Amazon S3 API、 AWS CLI、 AWS SDKs 或 wget 等工具來複製物件往返 S3。

使用 Amazon S3 檔案存取檔案

Amazon S3 Files 是一種無伺服器檔案系統，可讓您將 S3 一般用途儲存貯體掛載為運算執行個體上的高效能檔案系統。透過 S3 檔案，您可以使用本機掛載路徑上的讀取和寫入等標準檔案系統操作，以檔案形式存取 S3 物件。

您可以在啟動時或在執行中的執行個體上啟動後，將 S3 檔案系統掛載到 EC2 執行個體。

先決條件

使用 EC2 執行個體設定 S3 檔案之前，請確定您有下列項目：

- S3 檔案系統和至少一個處於可用狀態的掛載目標。如需有關建立 S3 檔案系統的資訊，請參閱《Amazon S3 使用者指南》中的使用 Amazon S3 檔案。 Amazon S3
- 連接執行個體描述檔的 EC2 Linux 執行個體。如需掛載檔案系統所需許可的相關資訊，請參閱《Amazon S3 使用者指南》中的 IAM 角色和政策。

- 允許執行個體與檔案系統掛載目標之間 NFS 流量 （連接埠 2049) 的安全群組。如需所需安全群組設定的資訊，請參閱《Amazon S3 使用者指南》中的安全群組。

使用 EC2 主控台在啟動時將檔案系統掛載至 EC2 執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇啟動執行個體。
3. 在Network settings (網路設定) 中，執行下列操作：
   a. 選擇編輯。
   b. 針對子網路，選取子網路。
   c. 選取預設安全群組，以確保您的 EC2 執行個體可以存取 S3 檔案系統。您無法使用此安全群組透過 Secure Shell (SSH) 存取您的 EC2 執行個體。若要透過 SSH 存取，您可以稍後編輯預設安全群組並新增規則以允許 SSH，或新增允許 SSH 的新安全群組。您可以使用以下設定：
      i. Type (類型)：SSH
      ii. Protocol (通訊協定)：TCP
      iii. Port Range (連接埠範圍)：22
      iv. Source (來源)：Anywhere (任何位置) 0.0.0.0/0
4. 在設定儲存體下，執行下列動作：
   a. 在檔案系統下，選擇 S3 檔案。
   b. 選擇 Add shared file system (新增共用檔案系統)。
   c. 對於 S3 檔案系統，您的檔案系統會根據您在網路設定中選取的子網路顯示在可用區域中。選擇您要掛載的 S3 檔案系統。如果您沒有任何檔案系統，請選擇建立新的檔案系統來建立新的檔案系統。
   d. 在 EC2 執行個體上輸入要掛載檔案系統的本機掛載路徑 （例如 `/mnt/s3files`)。
   e. 將產生命令來掛載檔案系統並將其新增至 fstab。您可以將此命令新增至進階詳細資訊下的使用者資料欄位。然後，您的 EC2 執行個體將設定為在啟動時以及重新啟動時掛載 S3 檔案系統。您也可以在 EC2 執行個體啟動後執行這些命令。
5. 在進階詳細資訊下，將執行個體描述檔連接至您的執行個體。您的 IAM 角色必須具有掛載檔案系統和存取 S3 儲存貯體的許可。如需所需許可的詳細資訊，請參閱《Amazon S3 使用者指南》中的 IAM 角色和政策。
6. 選擇啟動執行個體。

執行個體啟動後，會安裝所需的軟體公用程式，並掛載檔案系統。您可以透過導覽至本機掛載路徑來檢視檔案系統。

在啟動後將檔案系統掛載至 EC2 執行個體

1. 透過 Secure Shell (SSH) 或使用 EC2 主控台中的 EC2 Instance Connect 連線至 EC2 執行個體。
2. 若要掛載 S3 檔案系統，請使用掛載協助程式公用程式 `amazon-efs-utils`。根據您的 Linux 發行版本，使用下列其中一個命令來安裝`amazon-efs-utils`套件：

   a. 如果您使用的是 Amazon Linux，請執行下列命令，從 Amazon 的儲存庫安裝 efs-utils：

   ```
   sudo yum -y install amazon-efs-utils
   ```

   b. 如果您使用其他支援的 Linux 發行版本，請執行下列命令：

   ```
   curl https://amazon-efs-utils.aws.com/efs-utils-installer.sh | sudo sh -s -- --
   install
   ```

   c. 如需其他 Linux 發行版本，請參閱 GitHub 上的 efs-utils 儲存庫。
3. 使用下列命令建立檔案系統掛載點的目錄：

   ```
   sudo mkdir {path/to/mount}
   ```
4. 掛載 S3 檔案系統：

   ```
   FS="{YOUR_FILE_SYSTEM_ID}"
   sudo mount -t s3files $FS:/ {path/to/mount}
   ```
5. 確認檔案系統已掛載：

   ```
   df -h {path/to/mount}
   ```

以檔案形式檢視 S3 儲存貯體中的物件

完成上述程序後，您現在可以使用標準檔案系統操作，在本機掛載路徑上讀取和寫入 S3 物件做為檔案。如果您的 S3 儲存貯體中有物件，您可以使用下列命令將其檢視為檔案：

```
ls {path/to/mount}
```

物件型存取

您可以使用 Amazon S3 S3 之間複製檔案。 AWS CLI AWS SDKs 如果您有必要的許可，您可以使用下列其中一種方法，在 Amazon S3 和執行個體之間複製檔案。

wget

> **Note**
>
> 此方法僅適用於公有物件。如果物件非公有，您會收到 `ERROR 403: Forbidden` 訊息。如果您收到此錯誤，您必須使用 Amazon S3 主控台 AWS CLI、 AWS API、 AWS SDK 或 AWS Tools for Windows PowerShell，而且您必須擁有必要的許可。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 Amazon S3 中的身分和存取管理及下載物件。

wget 公用程式是 HTTP 和 FTP 用戶端，可用來從 Amazon S3 下載公有物件。其預設安裝在 Amazon Linux 和多數其他的發行版本上，也可供 Windows 下載。若要下載 Amazon S3 物件，請用下列命令並更換要下載之物件的 URL。

```
[ec2-user ~]$ wget https://amzn-s3-demo-bucket.s3.amazonaws.com/path-to-file
```

PowerShell

可以使用 AWS Tools for Windows PowerShell，將物件移入和移出 Amazon S3。

使用 Copy-S3Object cmdlet，將 Amazon S3 物件複製至 Windows 執行個體，如下所示。

```
Copy-S3Object `
    -BucketName amzn-s3-demo-bucket `
    -Key path-to-file `
    -LocalFile my_copied_file.ext
```

或者，可以在 Windows 執行個體上使用 Web 瀏覽器來開啟 Amazon S3 主控台。

AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 從 Amazon S3 下載限制項目，以及上傳項目。如需詳細資訊，包括如何安裝與設定工具，請參閱 AWS Command Line Interface 詳細資訊頁面。

aws s3 cp 命令類似於 Unix cp 命令。可讓您從 Amazon S3 複製檔案到執行個體，從執行個體複製檔案到 Amazon S3，以及從某一 Amazon S3 位置複製檔案到其他位置。

使用下列命令，將物件從 Amazon S3 複製到執行個體：

```
aws s3 cp s3://amzn-s3-demo-bucket/my_folder/my_file.ext my_copied_file.ext
```

使用下列命令將物件從執行個體複製回 Amazon S3：

```
aws s3 cp my_copied_file.ext s3://amzn-s3-demo-bucket/my_folder/my_file.ext
```

aws s3 sync 命令可將整個 Amazon S3 儲存貯體同步至本機目錄位置。此功能在下載資料集，以及讓本機複本與遠端資料集保持同步時很有幫助。如果您擁有 Amazon S3 儲存貯體的適當許可，還能將命令中的來源與目的地位置對調，於完成後將本機目錄備份推送至雲端。

使用下列命令，將整個 Amazon S3 儲存貯體下載到執行個體上的本機目錄：

```
aws s3 sync s3://amzn-s3-demo-source-bucket local_directory
```

Amazon S3 API

您可以使用 API 來存取 Amazon S3 中的資料。您可使用此 API 來協助開發應用程式，並將其與其他 API 和軟體開發套件整合。如需詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的使用 SDK 的 Amazon S3 程式碼範例 AWS SDKs。

搭配 Amazon EC2 Linux 執行個體一起使用 Amazon EFS

> **Note**
>
> Windows 執行個體上不支援 Amazon EFS。

Amazon EFS 提供可擴展的檔案儲存體，以搭配 Amazon EC2 使用。您可以使用 EFS 檔案系統做為多個執行個體上執行之工作負載和應用程式的共同資料來源。如需詳細資訊，請參閱 Amazon Elastic File System 產品頁面。

本教學課程會說明如何在執行個體啟動期間，使用 Amazon EFS Quick Create 精靈建立和連接 Amazon EFS 檔案系統。如需有關如何使用 Amazon EFS 主控台建立檔案系統的教學課程，請參閱「Amazon Elastic File System 使用者指南」中的 Amazon Elastic File System 入門。

> **Note**
>
> 當您使用 EFS Quick Create 建立 EFS 檔案系統時，會使用下列服務建議的設定建立檔案系統：
>
> - 啟用自動備份。
> - 在選取的 VPC 中管理掛載目標。
> - 一般用途效能模式。
> - 爆量輸送量模式。
> - 使用預設金鑰為 Amazon EFS (`aws/elasticfilesystem`) 啟用靜態資料加密。
> - 使用 30天政策啟用 Amazon EFS 生命週期管理。

任務

- 使用 Amazon EFS Quick Create 建立 EFS 檔案系統
- 測試 EFS 檔案系統
- 刪除 EFS 檔案系統

使用 Amazon EFS Quick Create 建立 EFS 檔案系統

您可以建立 EFS 檔案系統，並在使用 Amazon EC2 啟動執行個體精靈的 Amazon EFS Quick Create 功能啟動您的執行個體時，將其掛載到您的執行個體。

使用 Amazon EFS Quick Create 建立 EFS 檔案系統

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇啟動執行個體。
3. (選用) 在 Name and tags (名稱和標籤) 下，針對 Name (名稱)，輸入用於識別執行個體的名稱。

4. 在 Application and OS Images (Amazon Machine Image) (應用程式和作業系統映像 (Amazon Machine Image)) 下，請選取 Linux 作業系統，然後針對 Amazon Machine Image (AMI)，選取 Linux AMI。
5. 在 Instance type (執行個體類型) 下，針對 Instance type (執行個體類型)，選取執行個體類型或保留預設值。
6. 在 Key pair (login) (金鑰對 (登入)) 下，針對 Key pair name (金鑰對名稱)，選擇現有的金鑰對或建立新的金鑰對。
7. 在 Network settings (網路設定) 下，選取 Edit (編輯) (在右側)，然後針對 Subnet (子網)，選取一個子網。

   > **Note**
   >
   > 您必須先選取子網，才能新增 EFS 檔案系統。

8. 在 Configure storage (設定儲存) 下，選擇 Edit (編輯) (在右下角)，然後執行下列動作：

   a. 針對檔案系統，請確認已選取 EFS，然後請選擇建立新的共用檔案系統。

   b. 針對檔案系統名稱，請輸入 Amazon EFS 檔案系統的名稱，然後請選擇建立檔案系統。

   c. 針對掛載點，請指定自訂掛載點或保留預設值。

   d. 若要啟用檔案系統的存取權，請選取 Automatically create and attach security groups (自動建立和連接安全群組)。選取此核取方塊後，系統將自動建立以下安全群組，並將其連接至檔案系統的執行個體和掛載目標：

      - 執行個體安全群組 - 包含允許透過 NFS 2049 連接埠流量的傳出規則，但是不包含傳入規則。
      - 檔案系統掛載目標安全群組 - 包含允許來自執行個體安全群組的 NFS 2049 連接埠流量的傳入規則 (如上所述)，以及允許透過 NFS 2049 連接埠流量的傳出規則。

      > **Note**
      >
      > 或者，您可以手動建立並連接安全群組。如果您要手動建立並連接安全群組，請清除 Automatically create and attach the required security groups (自動建立並連接必要的安全群組)。

   e. 若要在執行個體啟動時自動掛載共用檔案系統，請選取 Automatically mount shared file system by attaching required user data script (透過連接所需的使用者指令碼自動掛載共用檔

案系統)。若要檢視自動產生的使用者資料，請展開 Advanced details (進階詳細資訊)，然後向下捲動至 User data (使用者資料)。

> **Note**
>
> 如果您已在選取此核取方塊之前先新增使用者資料，原始使用者資料將會被自動產生的使用者資料覆寫。

9. 視需要進行任何其他執行個體組態設定。
10. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

測試 EFS 檔案系統

您可以連線至執行個體，並確認將檔案系統掛載至您指定的目錄 (例如 /mnt/efs)。

確認檔案系統已掛載

1. 連線到您的執行個體。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。
2. 從執行個體的終端機視窗，執行 df -T 命令，確認 EFS 檔案系統已掛載。

```
$ df -T
Filesystem     Type              1K-blocks    Used          Available Use% Mounted
 on
/dev/xvda1     ext4                8123812 1949800            6073764  25% /
devtmpfs       devtmpfs            4078468      56            4078412   1% /dev
tmpfs          tmpfs               4089312       0            4089312   0% /dev/shm
efs-dns        nfs4       9007199254740992       0   9007199254740992   0% /mnt/efs
```

請注意，檔案系統名稱 (在範例輸出中顯示為 *efs-dns*) 的格式如下。

```
file-system-id.efs.aws-region.amazonaws.com:/
```

3. (選用) 在檔案系統中從執行個體建立檔案，然後確認您可以從另一個執行個體檢視檔案。
   a. 從執行個體中，執行下列命令來建立檔案。

   ```
   $ sudo touch /mnt/efs/test-file.txt
   ```

   b. 從另一個執行個體中，執行下列命令來檢視檔案。

```
$ ls /mnt/efs
test-file.txt
```

刪除 EFS 檔案系統

若您不再需要檔案系統，可以刪除它。

刪除檔案系統

1. 開啟 Amazon Elastic File System 主控台，網址為 https://console.aws.amazon.com/efs/。
2. 選取要刪除的檔案系統。
3. 選擇 Actions (動作)、Delete file system (刪除檔案系統)。
4. 出現確認提示時，請輸入檔案系統 ID，然後選擇 Delete file system (刪除檔案系統)。

使用 Amazon FSx 搭配 Amazon EC2 執行個體

Amazon FSx 系列服務使您可以輕鬆啟動、執行和擴展由常用商業和開放原始碼檔案系統提供支援的共用儲存體。您可以使用新的啟動執行個體精靈，在啟動時自動將以下類型的 Amazon FSx 檔案系統連接到您的 Amazon EC2 執行個體：

- Amazon FSx for NetApp ONTAP 提供 AWS 雲端中全受管共用儲存，具有 NetApp ONTAP 的熱門資料存取和管理功能。
- Amazon FSx for OpenZFS 可提供由常用 OpenZFS 檔案系統支援的全受管共用儲存體，此儲存體具備成本效益。

> **Note**
>
> - 此功能僅在新的啟動執行個體精靈中可用。如需詳細資訊，請參閱 使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體
> - Amazon FSx for Windows File Server 和 Amazon FSx for Lustre 檔案系統無法在啟動時掛載。啟動後，必須手動掛載這些檔案系統。

您可以選擇掛載先前建立的現有檔案系統，也可以建立新的檔案系統，以便在啟動時掛載到執行個體。

主題

- 安全群組和使用者資料指令碼
- 在啟動時掛載 Amazon FSx 檔案系統

安全群組和使用者資料指令碼

當您使用啟動執行個體精靈將 Amazon FSx 檔案系統掛載到執行個體時，您可以選擇是否自動建立並連接啟用存取檔案系統所需的安全群組，以及是否自動包含掛載檔案系統所需的使用者資料指令碼並將其設定為可供使用。

主題

- Security groups (安全群組)
- 使用者資料指令碼

Security groups (安全群組)

如果您選擇自動建立啟用存取檔案系統所需的安全群組，啟動執行個體精靈將建立並連接兩個安全群組：一個安全群組連接到執行個體，另一個連接到檔案系統。有關安全群組要求的詳細資訊，請參閱「適用於 Amazon VPC ONTAP 檔案系統存取控制的 FSx」和「適用於 Amazon VPC 的 OpenZFS 檔案系統存取控制的 FSx」。

我們會將標籤 `Name=instance-sg-1` 新增至建立並附加至執行個體的安全群組。標籤中的值會在每次啟動執行個體精靈為 Amazon FSx 檔案系統建立安全群組時自動遞增。

安全群組包含下列輸出規則，但不包含輸入規則。

傳出規則

通訊協定類型	連接埠號碼	目的地
UDP	111	########
UDP	20001 - 20003	########
UDP	4049	########
UDP	2049	########
UDP	635	########

通訊協定類型	連接埠號碼	目的地
UDP	4045 - 4046	########
TCP	4049	########
TCP	635	########
TCP	2049	########
TCP	111	########
TCP	4045 - 4046	########
TCP	20001 - 20003	########
全部	全部	########

建立並附加至檔案系統的安全群組會以 `Name=fsx-sg-`*1* 標記。標籤中的值會在每次啟動執行個體精靈為 Amazon FSx 檔案系統建立安全群組時自動遞增。

安全群組包括以下規則。

傳入規則

通訊協定類型	連接埠號碼	來源
UDP	2049	########
UDP	20001 - 20003	########
UDP	4049	########
UDP	111	########
UDP	635	########
UDP	4045 - 4046	########
TCP	4045 - 4046	########

通訊協定類型	連接埠號碼	來源
TCP	635	########
TCP	2049	########
TCP	4049	########
TCP	20001 - 20003	########
TCP	111	########

傳出規則

通訊協定類型	連接埠號碼	目的地
全部	全部	0.0.0.0/0

使用者資料指令碼

如果您選擇自動連接使用者資料指令碼，啟動執行個體精靈會將以下使用者資料新增到執行個體中。此指令碼將安裝必要的套件、掛載檔案系統並更新您的執行個體設定，以便在執行個體重新啟動時，自動重新掛載檔案系統。

```
#cloud-config
package_update: true
package_upgrade: true
runcmd:
- yum install -y nfs-utils
- apt-get -y install nfs-common
- svm_id_1=svm_id
- file_system_id_1=file_system_id
- vol_path_1=/vol1
- fsx_mount_point_1=/mnt/fsx/fs1
- mkdir -p "${fsx_mount_point_1}"
- if [ -z "$svm_id_1" ]; then printf "\n${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
 0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
```

```
  nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
  0\n" >> /etc/fstab; fi
- retryCnt=15; waitTime=30; while true; do mount -a -t nfs4 defaults; if [ $? = 0 ] ||
  [ $retryCnt -lt 1 ]; then echo File system mounted successfully; break; fi; echo File
  system not available, retrying to mount.; ((retryCnt--)); sleep $waitTime; done;
```

在啟動時掛載 Amazon FSx 檔案系統

在啟動時掛載新的或現有的 Amazon FSx 檔案系統

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，接著選擇 Launch instance (啟動執行個體) 來開啟啟動執行個體精靈。
3. 在 Application and OS Images (應用程式和 OS 映像) 區段中，選取要使用的 AMI。
4. 在 Instance type (執行個體類型) 區段中，選取執行個體類型。
5. 在 Key pair (金鑰對) 區段中，選取現有的金鑰對或建立新金鑰對。
6. 在 Network settings (網路設定) 區段中，執行下列動作：

   a. 選擇編輯。

   b. 如果您想要掛載現有的檔案系統，請針對 Subnet (子網) 選擇檔案系統的偏好子網。我們推薦您在與檔案系統之偏好子網路相同的可用區域中啟動執行個體，以便最佳化效能。

      如果您想要建立新的檔案系統以掛載到執行個體，請針對 Subnet (子網) 選擇要在其中啟動執行個體的子網。

      > **Important**
      >
      > 您必須選取一個子網才能在新的啟動執行個體精靈中啟用 Amazon FSx 功能。如果沒有選取子網，則無法掛載現有檔案系統或建立新的檔案系統。

7. 在 Storage (儲存) 區段中，執行下列動作：

   a. 根據需要設定磁碟區。

   b. 展開 File systems (檔案系統) 區段，然後選取 FSx。

   c. 選擇 Add shared file system (新增共用檔案系統)。

   d. 針對 File system (檔案系統)，選取要掛載的檔案系統。

> **Note**
>
> 該清單顯示所選區域內您帳戶的所有 Amazon FSx for NetApp ONTAP 和 Amazon FSx for OpenZFS 檔案系統。

e. 若要自動建立並連接啟用存取檔案系統所需的安全群組，請選取 Automatically create and attach security groups (自動建立並連接安全群組)。如果您希望手動建立安全群組，請清除核取方塊。如需詳細資訊，請參閱Security groups (安全群組)。

f. 若要自動連接掛載檔案系統所需的使用者資料指令碼，請選取 Automatically mount shared file system by attaching required user data script (透過連接所需的使用者資料指令碼自動掛載共享檔案系統)。如果您希望手動提供使用者資料指令碼，請清除核取方塊。如需詳細資訊，請參閱使用者資料指令碼。

8. 在 Advanced (進階) 區段中，根據需要編輯其他執行個體設定。
9. 選擇啟動。

搭配 Amazon EC2 執行個體一起使用 Amazon File Cache

Amazon File Cache 在 上提供全受管的高速快取， AWS 無論資料存放在何處，都能更輕鬆地處理檔案資料。Amazon File Cache 可作為暫時性且高效能的儲存位置，適用於存放於內部部署檔案系統、AWS 檔案系統和 Amazon Simple Storage Service (Amazon S3) 儲存貯體中的資料。您可以使用此功能，將分散的資料集提供給 AWS 具有統一檢視的檔案型應用程式，並以高速提供低於毫秒的延遲和高輸送量。如需詳細資訊，請參閱 Amazon File Cache 使用者指南。

Amazon File Cache 可與最熱門的 AMI 搭配使用，並與 x86 型執行個體類型和 Graviton 執行個體類型相容。您可以使用開放原始碼 Lustre 用戶端，從 Amazon EC2 執行個體存取快取。您可以掛載快取，然後使用標準 Linux 命令在快取中使用檔案和目錄。Amazon EC2 執行個體可以從相同虛擬私有雲端 (VPC) 中的其他可用區域存取快取，但前提是您的網路組態允許可在 VPC 內的子網路之間存取。您也可在共用 VPC 中建立快取。

若要開始使用，請參閱「Amazon File Cache 使用者指南」中的 Amazn File Cache 入門。

標記您的 Amazon EC2 資源

資源是指您可以使用的實體。當您使用 服務的功能時，Amazon EC2 會建立 資源。例如，Amazon EC2 資源包括映像、執行個體、機群、金鑰對和安全群組。所有 Amazon EC2 資源類型都包含描述資源的屬性。例如，名稱、描述、資源識別符和 Amazon Resource Name (ARN)。

Amazon EC2 資源專屬於其所在的 AWS 區域或區域。例如，Amazon Machine Image (AMI) 專屬於 AWS 區域，但您從 AMI 啟動的執行個體專屬於您啟動它的區域。您可以使用其 ARN，在許可政策中指定 Amazon EC2 資源。

您的 AWS 帳戶 具有 Amazon EC2 的預設配額。這些配額定義您可以建立的資源數量上限。例如，執行中執行個體的 vCPUs 數量上限有配額。如果啟動執行個體或啟動停止的執行個體會導致您超過配額，操作會失敗。

您可以使用資源 IDs或標籤， AWS 帳戶 依區域搜尋 中的特定資源。若要跨多個區域搜尋特定資源或資源類型，請使用 Amazon EC2 Global View。

目錄

- 為您的 Amazon EC2 資源選取區域
- 尋找您的 Amazon EC2 資源
- 使用 AWS 全域檢視跨區域檢視資源
- 標記您的 Amazon EC2 資源
- Amazon EC2 服務配額

為您的 Amazon EC2 資源選取區域

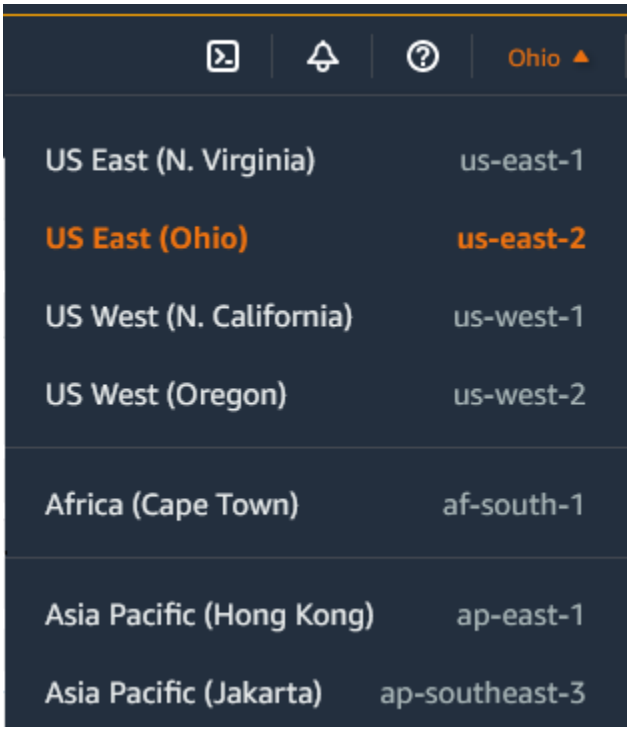
Amazon EC2 資源專屬於其所在的 AWS 區域或區域。每次建立 Amazon EC2 資源時，您可以指定資源的區域。

考量事項

- 有些 AWS 資源可能無法在所有區域中使用。在開始在區域中建立 AWS 資源之前，請確定您可以在區域中建立所需的所有資源。
- 某些區域預設為停用。*您必須先啟用這些區域，才能使用它們。若要了解詳細資訊，請參閱 AWS 區域。

使用主控台指定資源的區域

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽列中，選取 Regions (區域) 選取器，然後選取區域。



3. 區域選擇器包含可用於 的所有資源 AWS 帳戶。選擇清單底部附近的底線文字，以檢視未為您的帳戶啟用的區域。

使用 選取資源的區域 AWS CLI

您可以設定 AWS CLI 以使用預設區域。若未在命令中指定一個區域，則會使用預設區域。如需在特定命令中使用不同區域，可新增下面的選項。

```
--region us-east-1
```

如需 Tools for PowerShell 來為資源選取區域

您可對 Tools for Windows PowerShell 進行設定，以便使用預設區域。若未在命令中指定區域，則會使用預設區域。如需在特定命令中使用不同區域，可新增下面的參數。

```
-Region us-east-1
```

尋找您的 Amazon EC2 資源

您可以使用 Amazon EC2 主控台取得某些資源類型的清單。您可以使用資源對應的命令或 API 動作取得每種資源類型的清單。如果您有許多資源，您可篩選結果，只包含或排除符合特定條件的資源。

內容

- 使用主控台列出和篩選資源
- 使用命令列來列示及篩選
- 使用 AWS 全域檢視跨區域檢視資源

使用主控台列出和篩選資源

內容

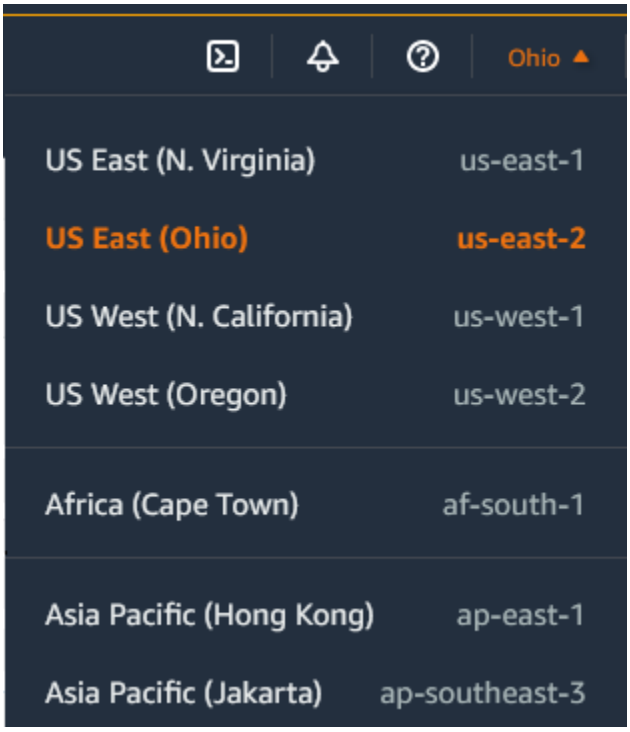
- 使用主控台列出資源
- 使用主控台篩選資源
  - 支援的篩選條件
- 使用主控台儲存篩選條件集
  - 主要功能
  - 建立篩選條件集
  - 修改篩選條件集
  - 刪除篩選條件集

使用主控台列出資源

您可以使用主控台檢視最常見的 Amazon EC2 資源類型。若要檢視其他資源，請使用命令列介面或 API 動作。

使用主控台列出 EC2 資源

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. Amazon EC2 資源專屬於 AWS 區域。從導覽列中，選取區域選取器。

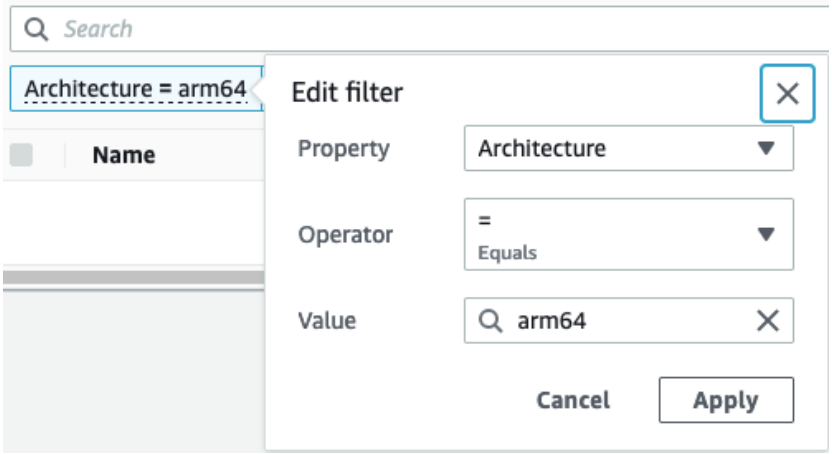


3. 在導覽窗格中選擇對應資源類型的選項。例如，若要列出執行個體，請選擇執行個體。

使用主控台篩選資源

篩選資源清單

1. 在導覽窗格中，選取資源類型 (例如，Instances (執行個體))。
2. 選擇搜尋欄位。
3. 從清單中選取篩選條件。
4. 選取運算子，例如 = (等於)。某些屬性有更多可用的運算子可供選取。請注意，並非所有畫面均支援選取運算子。
5. 選取篩選條件值。
6. 若要編輯選取的篩選條件，請選取篩選條件字符 (藍色框)，進行所需的編輯，然後選取 Apply (套用)。請注意，並非所有畫面均支援編輯選取的篩選條件。



7. 完成後，請移除篩選器。

支援的篩選條件

Amazon EC2 主控台支援兩種篩選類型。

- API 篩選發生在伺服器端。篩選會套用在 API 呼叫上，這會減少伺服器傳回的資源數目。它允許快速篩選大量資源，並且可以減少伺服器和瀏覽器之間傳輸資料的時間和成本。API 篩選支援 = (等於) 和 : (包含) 運算子，並且一律區分大小寫。
- 用戶端篩選發生在用戶端。它可讓您篩選瀏覽器中已經可用的資料 (也就是 API 已傳回的資料)。用戶端篩選與 API 篩選器結合使用效果很好，可在瀏覽器中篩選至較小的資料集。除了 = (等於) 和 : (包含) 運算子，用戶端篩選也可以支援範圍運算子，例如 >= (大於或等於) 和否定 (反向) 運算子，例如 != (不等於)。

Amazon EC2 主控台支援下列類型的搜尋：

以關鍵字搜尋

以關鍵字搜尋是一種任意文字搜尋，可讓您在所有資源屬性或標籤中搜尋值，而不需指定要搜尋的屬性或標籤金鑰。

> **Note**
>
> 所有關鍵字搜尋都使用用戶端篩選。

要以關鍵字搜尋，請在搜尋方塊中輸入或貼上您要找的內容，然後選擇 Enter。例如，搜尋 123 會符合任何屬性中具有 123 的所有執行個體，例如 IP 地址、執行個體 ID、VPC ID 或 AMI ID，或

在其任何標籤中，例如 Name (名稱)。如果任意文字搜尋傳回非預期的相符項目，請套用其他篩選器。

依屬性搜尋

依屬性搜尋可讓您在所有資源中搜尋特定屬性。

> **Note**
>
> 屬性搜尋會使用 API 篩選或用戶端篩選，視選取的屬性而定。執行屬性搜尋時，屬性會相應地分組。

例如，您可以搜尋所有執行個體的 Instance state (執行個體狀態) 屬性，以僅傳回處於 `stopped` 狀態的執行個體。若要執行此作業：

1. 在執行個體畫面的搜尋欄位中，開始輸入 `Instance state`。在您輸入字元時，針對以下執行個體狀態會出現兩種類型的篩選條件：API 篩選條件和用戶端篩選條件。
2. 若要在伺服器側進行搜尋，請在 API filters (API 篩選條件) 下選擇 Instance state (執行個體狀態)。若要在用戶端側搜尋，請在 Client filters (用戶端篩選條件) 下選擇 Instance state (client) (執行個體狀態 (用戶端))。

   此時會顯示所選屬性的可能運算子清單。
3. 選擇 = (等於) 運算子。

   此時會顯示所選屬性和運算子的可能值清單。
4. 從清單中選擇 Stopped (已停止)。

依標籤搜尋

依標籤搜尋，您可以透過標籤金鑰或標籤值篩選目前顯示的表中的資源。

標籤搜尋使用 API 篩選或者用戶端篩選，具體取決於 Preferences (偏好設定) 視窗中的設定。

確保標籤的 API 篩選

1. 開啟 Preferences (偏好設定) 視窗。
2. 清除使用常規運算式比對核取方塊。如果選取此核取方塊，則會執行用戶端篩選。
3. 選取使用區分大小寫比對核取方塊。如果清除此核取方塊，則會執行用戶端篩選。
4. 選擇確認。

依標籤搜尋時，可以使用下列值：

- (empty) ((空白)) – 尋找具有指定標籤金鑰的所有資源，但必須沒有標籤值。
- All values (所有值) – 尋找具有指定標籤金鑰和任何標籤值的所有資源。
- Not tagged (未標記) – 尋找不含指定標籤金鑰的索引資源。
- 顯示的值 – 尋找具有指定標籤金鑰和指定標籤值的所有資源。

您可以使用下列技巧來強化或精簡搜尋：

反向搜尋

反向搜尋可讓您搜尋不符合指定值的資源。在執行個體和 AMI 畫面中，透過選取 != (不等於) 或 !: (不包含) 運算子，然後選取一個值來執行反向搜尋。在其他畫面中，反向搜尋是透過在搜尋關鍵字前面加上驚嘆號 (!) 字元來執行。

> **Note**
>
> 只有用戶端篩選條件上的關鍵字搜尋和屬性搜尋才支援反向搜尋。API 篩選器上的屬性搜尋不支援此功能。

例如，您可以搜尋所有執行個體的 Instance state (執行個體狀態) 屬性，以排除所有處於 `terminated` 狀態的執行個體。若要執行此作業：

1. 在執行個體畫面的搜尋欄位中，開始輸入 `Instance state`。在您輸入字元時，針對以下執行個體狀態會出現兩種類型的篩選條件：API 篩選條件和用戶端篩選條件。
2. 在 Client filters (用戶端篩選條件) 下選擇 Instance state (client) (執行個體狀態 (用戶端))。只有用戶端篩選條件支援反向搜尋。

   此時會顯示所選屬性的可能運算子清單。
3. 選擇 != (不等於)，然後選擇 terminated (終止)。

若要根據執行個體狀態屬性來篩選執行個體，您還可以使用 Instance state (執行個體狀態) 欄中的搜尋圖示 ()。帶有加號 (+) 的搜尋圖示會顯示符合該屬性的所有執行個體。帶有減號 (-) 的搜尋圖示會排除符合該屬性的所有執行個體。

以下是使用反向搜尋的另一個範例：若要列出未獲指派名稱為 launch-wizard-1 之安全群組的所有執行個體，請在 Client filters (用戶端篩選條件) 下，依 Security group name (安全群組名稱) 屬性搜尋，選擇 !=，並在搜尋列中輸入 launch-wizard-1。

部分搜尋

透過部分搜尋，您可以搜尋部分字串值。若要執行部分搜尋，請僅輸入您要搜尋的關鍵字部分。在 Instances (執行個體) 和 AMIs 畫面中，部分搜尋只能使用 : (包含) 運算子執行。在其他畫面上，可以選取用戶端篩選條件屬性，並立即僅輸入您要搜尋的關鍵字部分。例如，在 Instance type (執行個體類型) 畫面上，若要搜尋所有 t2.micro、t2.small 和 t2.medium 執行個體，請依 Instance Type (執行個體類型) 屬性搜尋，並輸入關鍵字 t2。

常規運算式搜尋

若要使用常規運算式搜尋，必須選取偏好設定視窗中的使用常規運算式比對核取方塊。

當您需要將欄位中的值與特定模式匹配，就可以使用常規表達式。例如，若要搜尋開頭為 s 的值，請搜尋 ^s。若要搜尋結尾為 xyz 的值，請搜尋 xyz$。或者，若要搜尋以數字開頭後跟一個或多個字元的值，請搜尋 [0-9]+.*。

> **Note**
>
> 只有用戶端篩選器上的關鍵字搜尋和屬性搜尋才支援常規運算式搜尋。API 篩選器上的屬性搜尋不支援此功能。

區分大小寫的搜尋

若要使用區分大小寫的搜尋，必須選取偏好設定視窗中的使用區分大小寫比對核取方塊。區分大小寫的偏好設定僅適用於用戶端和標籤篩選條件。

> **Note**
>
> API 篩選條件一律區分大小寫。

萬用字元搜尋

使用 * 萬用字元來比對零個或多個字元。使用 ? 萬用字元來比對零個或一個字元。例如，如果您有一個具有值 prod、prods 和 production 的資料集，則搜尋 prod* 會比對出所有值，而 prod? 僅會比對出 prod 和 prods。若要使用常值，請使用反斜線 (\) 將其逸出。例如，"prod*" 會符合 prod*。

> **Note**
>
> 只有 API 篩選器上的屬性和標籤搜尋才支援萬用字元搜尋。用戶端篩選器上的關鍵字搜尋和屬性及標籤搜尋不支援此功能。

組合搜尋

一般而言，具有相同屬性的多個篩選器會自動使用 OR 合併。例如，搜尋 `Instance State : Running` 和 `Instance State : Stopped` 會傳回執行中「或」已停止的所有執行個體。若要使用 AND 合併搜尋，請跨不同的屬性進行搜尋。例如，搜尋 `Instance State : Running` 和 `Instance Type : c4.large` 會傳回只屬於類型 `c4.large`「且」處於執行中狀態的執行個體。

使用主控台儲存篩選條件集

儲存的篩選條件集是自訂的篩選條件群組，您可以建立和重複使用這些篩選條件，以有效率地檢視 Amazon EC2 資源。此功能有助於簡化工作流程，快速存取特定資源檢視。

Amazon EC2 主控台僅支援儲存的篩選條件集，目前僅適用於磁碟區頁面。

主要功能

- 自訂：建立根據您的需求量身打造的篩選條件集。例如，您可以建立篩選條件集，以僅顯示在指定日期之後建立的gp3磁碟區。
- 預設篩選條件：設定頁面的預設篩選條件集，當您導覽至頁面時，預設篩選條件會自動套用。如果未設定預設值，則不會套用篩選條件。
- 簡易應用程式：選取儲存的篩選條件集以立即套用。然後，Amazon EC2 會顯示相關資源，其中包含以藍色字符表示的作用中篩選條件。
- 彈性：視需要建立、修改和刪除篩選條件集。

建立篩選條件集

建立新的篩選條件集

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Volumes (磁碟區)。

> **Note**
>
> 儲存的篩選條件集目前僅可在磁碟區使用。

3. 在搜尋欄位中，選取篩選條件集的篩選條件。
4. 選擇清除篩選條件按鈕旁的箭頭，然後選擇儲存新的篩選條件集。
5. 在儲存篩選條件集視窗中，執行下列動作：
   a. 對於篩選條件集名稱，請輸入篩選條件集的名稱。
   b. (選用) 針對篩選條件集描述，輸入篩選條件集的描述。
   c. (選用) 若要將篩選條件集設為預設篩選條件，請選取設為預設核取方塊。

      > **Note**
      >
      > 每次開啟主控台頁面時，預設篩選條件都會自動套用。

   d. 選擇儲存。

修改篩選條件集

修改篩選條件集

1. 從儲存的篩選條件集清單中，選取要修改的篩選條件。
2. 若要新增篩選條件，請在搜尋欄位中選取要新增至篩選條件集的篩選條件。若要刪除集合中的篩選條件，請選擇篩選條件字符上的 X。
3. 選擇清除篩選條件按鈕旁的箭頭，然後選擇修改篩選條件集。
4. 在修改篩選條件集視窗中，執行下列動作：
   a. (選用) 若要將篩選條件集設為預設篩選條件，請選取設為預設核取方塊。

      > **Note**
      >
      > 每次開啟主控台頁面時，預設篩選條件都會自動套用。

   b. 選擇 Modify (修改)。

刪除篩選條件集

刪除篩選條件集

1. 從儲存的篩選條件集清單中，選取要刪除的篩選條件。
2. 選擇清除篩選條件按鈕旁的箭頭，然後選擇刪除篩選條件集。
3. 在刪除篩選條件集視窗中，檢閱篩選條件以確認這是您要刪除的篩選條件，然後選擇刪除。

使用命令列來列示及篩選

每種資源類型都有對應的 API 動作，可讓您用來描述、列出或取得該類型的資源。產生的資源清單可能很長，因此若要加快速度並提高實用性，請篩選結果，只包含符合特定條件的資源。

篩選考量

- 您可以在單一請求中指定最多 50 個篩選條件和最多 200 個總篩選條件值。
- 篩選條件字串長度最多可達 255 個字元。
- 您可以使用萬用字元搭配篩選條件值。星號 (*) 代表零或更多字元，問號 (?) 代表零或一個字元。
- 篩選條件值區分大小寫。
- 您的搜尋可包含萬用字元的文字值；您只需要在字元前加一個反斜線區隔它們即可。例如，`\*amazon\?\\` 值會搜尋 `*amazon?\` 文字字串。
- 您無法指定 的篩選條件值null。反之，請使用用戶端篩選。例如，下列命令使用 --query選項，並傳回在沒有金鑰對的情況下啟動的執行個體 IDs。

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[?!not_null(KeyName)].InstanceId' \
    --output text
```

AWS CLI

Example範例：指定單一篩選條件

您可使用 describe-instances 列出 Amazon EC2 執行個體。在沒有篩選條件的情況下，回應會包含您所有資源的資訊。您可使用下面的選項，只在輸出中包含執行中的執行個體。

```
--filters Name=instance-state-name,Values=running
```

若只要列出執行中執行個體的執行個體 ID，請如下所示新增 `--query` 選項。

```
--query "Reservations[*].Instances[*].InstanceId"
```

Example範例：指定多個篩選條件或篩選條件值

如果您指定多個篩選條件或多個篩選條件值，資源必須符合所有要包含在輸出中的篩選條件。

您可使用下面的選項，列出類型為 `m5.large` 或 `m5d.large` 的所有執行個體。

```
--filters Name=instance-type,Values=m5.large,m5d.large
```

您可使用下面的選項，列出類型為 `t2.micro` 的所有已停止的執行個體。

```
--filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example範例：在篩選條件值中使用萬用字元

您可使用下面的選項並搭配 describe-snapshots，以便僅傳回描述為「資料庫」的快照。

```
--filters Name=description,Values=database
```

* 萬用字元符合零或更多字元。您可使用下面的選項，以便僅傳回描述中包括字詞 database 的快照。

```
--filters Name=description,Values=*database*
```

? 萬用字元確切符合 1 個字元。您可使用下面的選項，以便僅傳回描述為 "database" 或 "database" 且後接一個字元的快照。

```
--filters Name=description,Values=database?
```

您可使用下面的選項，以便僅傳回描述為「資料庫」且後接最多四個字元的快照。這會排除具有「資料庫」後面接著五個或更多字元的描述。

```
--filters Name=description,Values=database????
```

Example範例：根據日期篩選

透過 AWS CLI，您可以使用 JMESPath 來使用表達式篩選結果。例如，下列describe-snapshots命令會顯示指定日期 AWS 帳戶 之前由 建立之所有快照IDs。如果您未指定擁有者，結果會包含所有公有快照。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime<='2024-03-31')].[SnapshotId]" \
    --output text
```

下面的範例將顯示指定日期範圍中所建立的所有快照的 ID。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

範例：根據標籤篩選

如需如何根據標籤篩選資源清單的範例，請參閱依標籤篩選 Amazon EC2 資源。

PowerShell

Example範例：指定單一篩選條件

您可使用 Get-EC2Instance 列出 Amazon EC2 執行個體。在沒有篩選條件的情況下，回應會包含您所有資源的資訊。您可使用下面的參數，以便僅在輸出中包括執行中的執行個體。

```
-Filter @{Name="instance-state-name"; Values="running"}
```

下面的範例僅會列示執行中執行個體的 ID。

```
(Get-EC2Instance -Filter @{Name="instance-state-name"; Values="stopped"}).Instances
 | Select InstanceId
```

Example範例：指定多個篩選條件或篩選條件值

如果您指定多個篩選條件或多個篩選條件值，資源必須符合所有要包含在結果中的篩選條件。

您可使用下面的選項，列出類型為 `m5.large` 或 `m5d.large` 的所有執行個體。

```
-Filter @{Name="instance-type"; Values="m5.large", "m5d.large"}
```

您可使用下面的選項，列出類型為 t2.micro 的所有已停止的執行個體。

```
-Filter @{Name="instance-state-name"; Values="stopped"}, @{Name="instance-type";
 Values="t2.micro"}
```

使用 AWS 全域檢視跨區域檢視資源

AWS 全域檢視可讓您在單一區域中 AWS ，或在單一主控台中同時跨多個區域檢視和搜尋資源。如需詳細資訊，請參閱使用 AWS 全域檢視跨區域檢視資源。

使用 AWS 全域檢視跨區域檢視資源

AWS 全域檢視可讓您在單一 AWS 區域或單一主控台中跨多個區域檢視部分資源。 AWS 全域檢視也提供全域搜尋功能，可讓您同時搜尋多個區域的特定資源或特定資源類型。

AWS Global View 不會讓您以任何方式修改資源。

支援的資源

使用 AWS 全域檢視，您可以在 AWS 帳戶 啟用 的所有區域中檢視下列資源的全域摘要。

- Auto Scaling 群組
- 可用區域
- 容量保留和容量區塊
- 資料庫叢集
- 資料庫執行個體
- DHCP 選項集
- Amazon Elastic Container Service (Amazon ECS) 叢集
- ECS 服務
- ECS 任務
- 輸出限定網際網路閘道
- 彈性 IP
- 端點服務

- 執行個體
- 網際網路閘道
- 受管理的字首清單
- NAT 閘道
- 網路 ACL
- 網路介面
- Outpost
- 路由表
- S3 儲存貯體
- Security groups (安全群組)
- 子網路
- 磁碟區
- VPC
- VPC 端點
- VPC 對等連線

所需的許可

使用者必須具有下列許可才能使用 AWS Global View。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "autoscaling:DescribeAutoScalingGroups",
      "ec2:DescribeRegions",
      "ec2:DescribeCapacityReservations",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeEgressOnlyInternetGateways",
      "ec2:DescribeAddresses",
      "ec2:DescribeVpcEndpointServices",
```

```
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribePrefixLists",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeVpcEndpointServiceConfigurations",
        "ec2:DescribeManagedPrefixLists",
        "ecs:ListClusters",
        "ecs:DescribeServices",
        "ecs:ListTasks",
        "outposts:ListOutposts",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
    }]
}
```

使用 AWS 全域檢視

登入至 AWS Global View 主控台。

> **Important**
>
> 您無法在 Firefox 中使用私有視窗來存取 AWS Global View。

主控台包含下列項目：

- 區域總管 - 此索引標籤包含下列部分：
  - 摘要 – 提供所有區域資源的高階概觀。

展開 顯示所有資源摘要，指出 AWS 帳戶 已啟用 的區域數量。其餘欄位表示您目前在這些區域中擁有的資源數目。選擇任一連結，即可檢視所有區域中該類型的資源。例如，如果 Instances (執行個體) 標籤下的連結是 29 in 10 Regions (10 個區域中的 29 個)，則表示您目前跨 10 個區域擁有 29 個執行個體。選擇該連結以檢視全部 29 個執行個體的清單。

- 區域總管 – 列出所有 AWS 區域 （包括未啟用您帳戶的那些），並提供每個區域每個資源類型的總計。

  選擇區域名稱，以檢視該特定區域所有類型的所有資源。例如，選擇 Africa (Cape Town) af-south-1 (非洲 (開普敦) af-south-1) 以檢視該區域中的所有 VPC、子網路、執行個體、安全群組、磁碟區和 Auto Scaling 群組。或者，選取一個區域，然後選取 View resources for selected Region (檢視所選區域的資源)。

  選擇特定區域中特定資源類型的值，以僅檢視該區域中該類型的資源。例如，選擇非洲 (開普敦) af-south-1 以僅檢視該區域中的執行個體。

- 全域搜尋 - 此索引標籤可讓您跨單一區域或多個區域搜尋特定資源或特定資源類型。它也可讓您檢視特定資源的詳細資訊。

  若要搜尋資源，請在方格前面的欄位中輸入搜尋條件。您可以依區域、資源類型以及指派給資源的標籤進行搜尋。

  若要檢視特定資源的詳細資訊，請在方格中選取該資源。您也可以選擇要在該各個主控台中開啟的資源 ID。例如，選擇執行個體 ID，以在 Amazon EC2 主控台中開啟執行個體，或選擇子網 ID，以在 Amazon VPC 主控台中開啟子網。

- 地域與區域 – 此頁面可讓您檢視和管理所有可用的地域、可用區域、本機區域和 Wavelength 區域。您可以使用清單檢視/地圖檢視切換來切換清單檢視和地圖檢視。您的檢視偏好設定會保留在各個工作階段中。
  - 地圖檢視 – 在互動式世界地圖上顯示所有區域，依狀態著色。若要查看位置的詳細資訊，請指向該位置或選擇該位置。選擇要採取動作的位置 – 您可以啟用區域、啟用區域或開啟其詳細資訊頁面。您也可以篩選地圖，以僅顯示特定狀態。地圖和詳細資訊清單會保持同步 - 選取其中一個位置會反白另一個位置。
  - 清單檢視 – 顯示資料表中的區域和區域。從區域索引標籤，您可以檢視所有 AWS 區域。狀態欄顯示針對 AWS 帳戶啟用的區域。

  您可在此頁面選取區域，以便：

  - 檢視區域代碼、地理位置，以及每個區域類型的數量等區域詳細資訊。

    您亦可檢視可用區域、本機區域及 Wavelength 區域清單，以及區域中的資源清單。

- 啟用或停用區域。

在每個區域索引標籤，您可檢視該區域類型的清單。在本機區域索引標籤中，您可選擇加入本機區域。

> **Tip**
>
> 如果您只使用特定區域或資源類型，您可以自訂 AWS 全域檢視，以僅顯示這些區域和資源類型。若要自訂顯示的區域和資源類型，請在導覽面板中選擇設定，然後在資源和區域索引標籤上，選取您不想在 AWS 全域檢視中顯示的區域和資源類型。

標記您的 Amazon EC2 資源

為協助您管理您的執行個體、映像和其他 Amazon EC2 資源，您可將您自己的中繼資料，以標籤的形式指派給每個資源。標籤可讓您以不同的方式分類 AWS 資源，例如依用途、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為—您可以依據先前指派的標籤，快速識別特定的資源。本主題說明標籤並示範如何建立它們。

> **Warning**
>
> 許多不同 API 呼叫傳回了標籤金鑰和其值。拒絕存取 `DescribeTags` 不會自動拒絕存取其他 API 傳回的標籤。根據最佳實務，建議您不要在標籤中包含敏感資料。

內容

- 標籤基本概念
- 標記您的 資源
- 標籤限制
- 標籤與存取管理
- 標記您的資源以便計費
- 在建立期間授予標籤 Amazon EC2 資源的許可
- 新增和移除 Amazon EC2 資源的標籤
- 依標籤篩選 Amazon EC2 資源

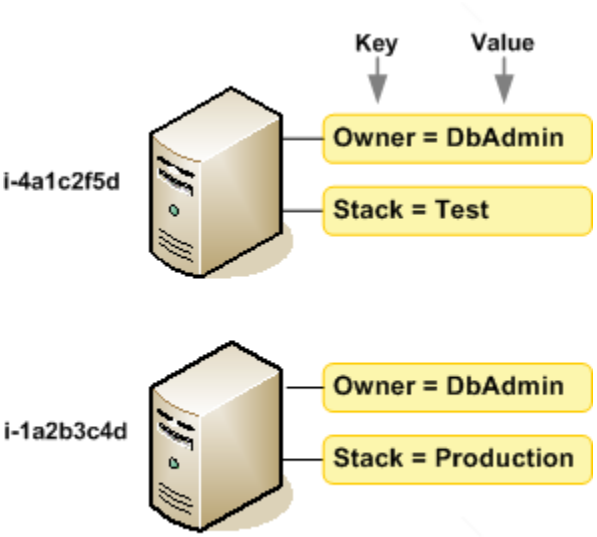
- 使用執行個體中繼資料檢視 EC2 執行個體的標籤

標籤基本概念

標籤是您指派給 AWS 資源的標籤。每個標籤皆包含由您定義的一個金鑰與一個選用值。

標籤可讓您以不同的方式分類 AWS 資源，例如依用途、擁有者或環境。例如，您可以為您帳戶的 Amazon EC2 執行個體定義一組標籤，協助您追蹤各個執行個體的擁有者與堆疊層級。

下圖說明標籤建立的運作方式。在此範例中，您已在每個執行個體上指派兩個標籤 – 一個標籤的金鑰為 `Owner`，另一個標籤的金鑰為 `Stack`。每個標籤也都有一個關聯值。



我們建議您為每種資源類型建立符合您需求的標籤金鑰。使用一致的標籤金鑰組可讓您更輕鬆管理您的資源。您可以根據您新增的標籤搜尋和篩選資源。如需如何實作有效資源標記策略的詳細資訊，請參閱標記最佳實務 AWS 白皮書。

標籤對 Amazon EC2 來說不具有任何語意意義，並會嚴格解譯為字元字串。此外，標籤不會自動指派給您的資源。您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為空白字串，但您無法將標籤的值設為 Null。若您將與現有標籤具有相同鍵的標籤新增到該資源，則新值會覆寫舊值。如果您刪除資源，也會刪除任何該資源所使用的標籤。

> **Note**
>
> 刪除資源之後，其標籤可能會在主控台、API 和 CLI 輸出中保持短時可見。這些標籤將逐漸取消與資源的關聯，並永久刪除。

標記您的 資源

使用 Amazon EC2 主控台時，您可以使用相關資源畫面上的標籤索引標籤將標籤套用至資源，也可以使用 AWS Resource Groups 主控台中的標籤編輯器。有些資源畫面可讓您在建立資源時指定資源的標籤；例如，具有 Name 鍵和您指定之值的標籤。在大多數的案例中，主控台會立即在建立資源後套用標籤 (而非在資源建立過程時)。主控台可能會根據 Name 標籤整理資源，但此標籤對 Amazon EC2 服務來說不具有任何語意意義。

如果您使用的是 Amazon EC2 API AWS CLI、 或 AWS SDK，您可以使用 `CreateTags` EC2 API 動作將標籤套用至現有資源。此外，有些資源建立動作可讓您在建立資源時指定資源的標籤。若標籤無法在資源建立時套用，我們會轉返資源建立程序。這可確保資源不是具有標籤建立，就是不會建立，因此無論何時都不會有不具有標籤的資源。藉由在建立時為資源建立標籤，您可以消除在資源建立後執行自訂標籤指令碼的必要。如需有關讓使用者在建立時為資源加上標籤的詳細資訊，請參閱 在建立期間授予標籤 Amazon EC2 資源的許可。

您可以在您的 IAM 政策中將標籤式的資源層級許可套用到支援在建立時新增標籤的 Amazon EC2 API 動作，以實作可在建立時為資源建立標籤之使用者和群組的細微控制。您的資源從建立時便已獲得適當保全，由於標籤會立即套用到您的資源，因此控制使用資源的任何標籤式資源層級許可都會立即生效。您可以更準確的追蹤和報告您的資源。您可以強制新資源使用標籤，並控制哪些標籤金鑰和值會在您的資源上設定。

您也可以將資源層級許可套用到 IAM 政策中的 `CreateTags` 和 `DeleteTags` Amazon EC2 API 動作，控制哪些標籤金鑰和值會在您現有的資源上設定。如需詳細資訊，請參閱 範例：標籤資源。

如需為您的資源建立標籤以便計費的詳細資訊，請參閱 AWS Billing 使用者指南中的 Using cost allocation tags (使用成本分配標籤)。

標籤限制

以下基本限制適用於標籤：

- 每一資源最多標籤數 – 50
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。
- 索引鍵長度上限 - 128 個 UTF-8 Unicode 字元
- 值的長度上限 - 256 個 UTF-8 Unicode 字元
- 允許的字元

  - 雖然 EC2 允許其標籤中的任何角色，但其他 AWS 服務更嚴格。所有 AWS 服務的允許字元包括：以 UTF-8 表示的字母 (a-z、A-Z)、數字 (0-9) 和空格，以及下列字元：+ - = . _ : / @。
  - 若啟用執行個體中繼資料中的執行個體標籤，則執行個體標籤鍵值只能使用字母 (a-z、A-Z)、數字 (0-9) 和下列字元：+ - = . , _ : @。執行個體標籤鍵不可含有空格或 /，而且不能僅包含 . (一個句點)、.. (兩個句點) 或 _index。如需詳細資訊，請參閱 使用執行個體中繼資料檢視 EC2 執行個體的標籤。
- 標籤鍵與值皆區分大小寫。
- 字aws:首會保留供 AWS 使用。如果標籤具有此字首的標籤金鑰，則您無法編輯或刪除標籤的金鑰或值。具 aws: 字首的標籤，不算在受資源限制的標籤計數內。

您無法僅根據標籤終止、停止或刪除資源。您必須指定資源識別符。例如，若要刪除您套用稱為 DeleteMe 標籤金鑰的快照，您必須搭配快照的資源識別符 (例如 DeleteSnapshots) 使用 snap-1234567890abcdef0 動作。

當您標記公有或共用資源時，您指派的標籤僅適用於 AWS 您的帳戶；其他 AWS 帳戶無法存取這些標籤。對於共用資源的標籤型存取控制，每個 AWS 帳戶都必須指派自己的一組標籤，以控制對資源的存取。

標籤與存取管理

如果您使用的是 AWS Identity and Access Management (IAM)，您可以控制 AWS 帳戶中哪些使用者具有建立、編輯或刪除標籤的許可。如需詳細資訊，請參閱在建立期間授予標籤 Amazon EC2 資源的許可。

您也可以使用資源標籤來實作以屬性為基礎的控制 (ABAC)。您可以建立 IAM 政策，允許根據資源的標籤進行操作。如需詳細資訊，請參閱 使用屬性型存取控制存取。

標記您的資源以便計費

您可以使用標籤來整理 AWS 帳單，以反映您自己的成本結構。若要這樣做，請註冊以取得包含標籤索引鍵值 AWS 的帳戶帳單。如需有關使用標籤設定成本分配報告的詳細資訊，請參閱 AWS Billing 使用者指南中的每月成本分配報告。若想要查看合併資源的成本，您可根據具有相同標籤金鑰值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。如需詳細資訊，請參閱 AWS Billing 使用者指南中的使用成本分配標籤。

> **Note**
>
> 若您才剛啟用報告，目前月份的資料會在 24 小時之後提供檢視。

成本分配標籤可用來表示哪些資源產生成本，但刪除或停用這些資源不一定能降低成本。例如，由另一個快照參考的快照資料會進行保留，即使包含原始資料的快照遭到刪除也不例外。如需詳細資訊，請參閱 AWS Billing 使用者指南中的 Amazon Elastic Block Store 磁碟區及快照。

> **Note**
>
> 已標記的彈性 IP 地址，不會顯示在您的成本分配報告中。

在建立期間授予標籤 Amazon EC2 資源的許可

有些資源建立 Amazon EC2 API 動作可讓您在建立資源時指定標籤。您可以使用資源標籤來實作以屬性為基礎的控制 (ABAC)。如需詳細資訊，請參閱標記您的 資源及使用屬性型存取控制存取。

使用者若要在建立時標記資源，他們必須具備建立資源動作 (如 `ec2:RunInstances` 或 `ec2:CreateVolume`) 的使用許可。若標籤於資源建立動作指定，Amazon 會針對 `ec2:CreateTags` 動作執行其他授權，以確認使用者具備建立標籤的許可。因此，使用者必須同時具備使用 `ec2:CreateTags` 動作的明確許可。

在 `ec2:CreateTags` 動作的 IAM 政策定義中，搭配 `ec2:CreateAction` 條件金鑰使用 `Condition` 元素，將標記許可給與建立資源的動作。

下列範例示範一個政策，其允許使用者啟動執行個體，並在啟動期間將標籤套用至執行個體與磁碟區。使用者沒有標記現有資源的權限 (他們不能直接呼叫 `ec2:CreateTags` 動作)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

同樣的，下列政策允許使用者建立磁碟區，並在磁碟區建立期間將標籤套用至該磁碟區。使用者沒有標記現有資源的權限 (他們不能直接呼叫 ec2:CreateTags 動作)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateVolume"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "CreateVolume"
          }
       }
    }
  ]
}
```

只有在資源建立動作中套用了標籤時，才評估 ec2:CreateTags 動作。因此，在沒有標記條件的情況下，若請求中未指定標籤，則具備資源建立許可的使用者不需要使用 ec2:CreateTags 動作的許可。

然而，若該使用者試圖建立具有標籤的資源卻未具備使用 ec2:CreateTags 動作的許可，則該請求會失敗。

如果在啟動範本中提供標籤，也會評估 ec2:CreateTags 動作。如需政策範例，請參閱 啟動範本中的標籤。

控制對特定標籤的存取

您可以在 IAM 政策的 Condition 元素中使用其他條件，來控制可套用至資源的標籤金鑰索和值。

下列條件金鑰機可與前一節中的範例搭配使用：

- aws:RequestTag：表示請求中必須存在特定標籤金鑰或標籤金鑰與值。請求內亦可指定其他標籤。
  - 搭配 StringEquals 條件運算子使用，以強制結合特定標籤金鑰與值，例如強制執行標籤 cost-center=cc123：

    ```
    "StringEquals": { "aws:RequestTag/cost-center": "cc123" }
    ```

  - 搭配 StringLike 條件運算子使用，以在請求中強制執行特定標籤金鑰，例如強制執行標籤金鑰 purpose：

    ```
    "StringLike": { "aws:RequestTag/purpose": "*" }
    ```

- aws:TagKeys：強制執行請求中使用的標籤金鑰。
  - 搭配 ForAllValues 修飾詞使用，若請求內提供特定標籤金鑰，將強制加以執行 (若請求內指定標籤，則僅允許特定標籤金鑰，不允許其他標籤)。例如，允許標籤金鑰 environment 或 cost-center：

    ```
    "ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
    ```

  - 搭配 ForAnyValue 修飾詞使用，以強制要求請求內至少具有一個指定的標籤金鑰。例如，請求內必須出現至少下列標籤金鑰 environment 或 webserver 之一：

    ```
    "ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }
    ```

這些條件鍵可套用於支援標記的資源建立動作，以及 ec2:CreateTags 和 ec2:DeleteTags 動作。若要了解 Amazon EC2 API 動作是否支援標記，請參閱 Amazon EC2 的動作、資源和條件金鑰。

若要強制使用者在建立資源時指定標籤，您必須在資源建立動作內，搭配 `aws:RequestTag` 修飾詞使用 `aws:TagKeys` 條件金鑰或 `ForAnyValue` 條件金鑰。若使用者未針對資源建立動作指定標籤，則不會評估 `ec2:CreateTags` 動作。

以條件而言，條件金鑰不區分大小寫，而條件值會區分大小寫。因此，欲強制標籤鍵區分大小寫，請使用 `aws:TagKeys` 條件索引鍵，其中標籤鍵指定為條件值。

如需 IAM 政策的範例，請參閱控制 Amazon EC2 API 存取的政策範例。如需了解詳細資訊，請參閱 IAM 使用者指南中的具有多個關聯內容索引鍵或值的條件。

新增和移除 Amazon EC2 資源的標籤

當您建立 Amazon EC2 資源時，例如 Amazon EC2 執行個體，您可以指定要新增至資源的標籤。您也可以使用 Amazon EC2 主控台來顯示特定 Amazon EC2 資源的標籤。您也可以從現有的 Amazon EC2 資源新增或移除標籤。

您可以使用 AWS Resource Groups 主控台中的標籤編輯器來檢視、新增或移除所有區域所有 AWS 資源的標籤。您可以同時從多個資源類型中一次套用或移除標籤。如需詳細資訊，請參閱標記 AWS 資源使用者指南。

任務

- 使用主控台新增標籤
- 使用 新增標籤 AWS CLI
- 使用 PowerShell 來新增標籤
- 使用 CloudFormation 新增標籤

使用主控台新增標籤

您可直接透過資源頁面來新增標籤至現有資源。

如需新增標籤至現有資源

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 請在導覽列中，選取資源所在的區域。
3. 在導覽窗格中，選取資源類型 (例如，Instances (執行個體))。
4. 從清單選取資源。
5. 從標籤索引標籤上，選擇管理標籤。

6. 請選擇新增標籤，然後輸入金鑰和標籤值。
7. 選擇儲存。

使用 新增標籤 AWS CLI

您可在建立資源時新增標籤，或新增標籤至現有資源。

如需在建立資源時新增標籤

請使用 `-tag-specifications` 選項在建立時給資源加標籤。標籤規格要求待加標籤的資源類型、標籤索引鍵，以及標籤值。在下面的範例中會建立標籤，然後將標籤新增至標籤規格。

```
--tag-specifications 'ResourceType=instance,Tags=[{Key=stack,Value=production}]'
```

如需新增標籤至現有資源

在下面的範例中，說明了如何使用 create-tags 命令將標籤新增至現有資源。

Example範例：將標籤新增至資源

此範例會將標籤 **Stack=production** 新增至指定的映像，或覆寫 AMI 的現有標籤，其中標籤金鑰為 stack。如果命令成功，則不會傳回任何輸出。

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 \
    --tags Key=stack,Value=production
```

Example範例：將標籤新增至多個資源

此範例會為 AMI 和執行個體新增 (或覆寫) 兩個標籤。其中一個標籤只包含金鑰 (webserver)，沒有值 (我們會將值設定為空字串)。其他標籤包含金鑰 (stack) 與值 (**Production**)。如果命令成功，則不會傳回任何輸出。

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 i-1234567890abcdef0 \
    --tags Key=webserver,Value=  Key=stack,Value=Production
```

Example範例：新增具有特殊字元的標籤

此範例會將標籤 [Group]=test 新增至執行個體。中括號 ([和]) 是特殊字元，必須將其逸出。

如果您使用 Linux 或 OS X，若要逸出特殊字元，請以雙引號 (") 括住具有特殊字元的元素，然後使用單引號 (') 括住整個金鑰和值結構。

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags 'Key="[Group]",Value=test'
```

如果您使用 Windows，若要逸出特殊字元，請以雙引號 (") 括住具有特殊字元的元素，然後在每個雙引號字元前面加上反斜線 (****)，如下所示：

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

如果您使用 Windows PowerShell，若要逸出特殊字元，請以雙引號 (**"**) 括住具有特殊字元的值，在每個雙引號字元前面加上反斜線 (****)，然後用單引號 (**'**) 括住整個金鑰和值結構，如下所示：

```
aws ec2 create-tags `
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

使用 PowerShell 來新增標籤

您可在建立資源時新增標籤，或新增標籤至現有資源。

如需在建立資源時新增標籤

請使用 `-TagSpecification` 參數在建立時給資源加標籤。標籤規格要求待加標籤的資源類型、標籤索引鍵，以及標籤值。在下面的範例中會建立標籤，然後將標籤新增至標籤規格。

```
$tag = @{Key="stack"; Value="production"}
$tagspec = new-object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance"
$tagspec.Tags.Add($tag)
```

在下面的範例中，在 `-TagSpecification` 參數中指定了此標籤。

```
-TagSpecification $tagspec
```

如需新增標籤至現有資源

使用 New-EC2Tag cmdlet。必須指定資源、標籤索引鍵，以及標籤值。

```
New-EC2Tag `
    -Resource i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="production"}
```

使用 CloudFormation 新增標籤

對於 Amazon EC2 資源類型，您可以使用 Tags 或 TagSpecifications 屬性來指定標籤。

以下範例使用其 **Stack=Production** 屬性，將標籤 Tags 新增到 AWS::EC2::Instance。

Example範例：YAML 中的標籤

```
Tags:
  - Key: "Stack"
    Value: "Production"
```

Example範例：JSON 中的標籤

```
"Tags": [
    {
        "Key": "Stack",
        "Value": "Production"
    }
]
```

以下範例使用其 **Stack=Production** 屬性，將標籤 TagSpecifications 新增到 AWS::EC2::LaunchTemplate LaunchTemplateData。

Example範例：YAML 中的 TagSpecifications

```
TagSpecifications:
  - ResourceType: "instance"
    Tags:
    - Key: "Stack"
      Value: "Production"
```

Example範例：JSON 中的 TagSpecifications

```
"TagSpecifications": [
```

```
    {
        "ResourceType": "instance",
        "Tags": [
            {
                "Key": "Stack",
                "Value": "Production"
            }
        ]
    }
]
```

依標籤篩選 Amazon EC2 資源

新增標籤後，您可以篩選以 Amazon EC2 資源為基礎的標籤索引鍵和標籤值。

Console

如需依標籤篩選資源

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取資源類型 (例如，Instances (執行個體))。
3. 選擇搜尋欄位。
4. 在清單中的標籤下，選擇標籤金鑰。
5. 從清單中選擇對應的標籤值。
6. 完成後，請移除篩選器。

如需有關在 Amazon EC2 主控台中使用篩選條件的詳細資訊，請參閱 尋找您的 Amazon EC2 資源。

AWS CLI

如需透過指定的標籤索引鍵來描述單一類型的資源

新增以下篩選條件至 `describe` 命令，以便透過 Stack 標籤來描述該類型的資源，無論標籤值為何。

```
--filters Name=tag-key,Values=Stack
```

如需透過指定的標籤來描述單一類型的資源

新增以下篩選條件至 `describe` 命令，以便透過標籤 Stack=production 來描述該類型的資源。

```
--filters Name=tag:Stack,Values=production
```

如需透過指定的標籤值來描述單一類型的資源

新增以下篩選條件至 `describe` 命令，以便透過具有值 production 的標籤來描述該類型的資源，無論標籤索引鍵為何。

```
--filters Name=tag-value,Values=production
```

如需透過指定的標籤來描述所有 EC2 資源

新增以下篩選條件至 describe-tags 命令，以便透過標籤 Stack=test 來描述全部 EC2 資源。

```
--filters Name=key,Values=Stack Name=value,Values=test
```

PowerShell

如需依標籤索引鍵來篩選單一類型的資源

新增以下篩選條件至 `Get` cmdlet，以便透過 Stack 標籤來描述該類型的資源，無論標籤值為何。

```
-Filter @{Name="tag-key"; Values="Stack"}
```

如需依標籤來篩選單一類型的資源

新增以下篩選條件至 `Get` cmdlet，以便透過標籤 Stack=production 來描述該類型的資源。

```
-Filter @{Name="tag:Stack"; Values="production"}
```

如需依標籤值來篩選單一類型的資源

新增以下篩選條件至 `Get` cmdlet，以便透過具有值 production 的標籤來描述該類型的資源，無論標籤索引鍵的值為何。

```
-Filter @{Name="tag-value"; Values="production"}
```

如需依標籤篩選全部 EC2 資源

新增以下篩選條件至 Get-EC2Tag cmdlet，以便透過標籤 Stack=test 來描述全部 EC2 資源。

```
-Filter @{Name="tag:Stack"; Values="test"}
```

使用執行個體中繼資料檢視 EC2 執行個體的標籤

您可以從執行個體中繼資料存取執行個體的標籤。透過從執行個體中繼資料存取標籤，您無需再使用 `DescribeInstances` 或 `DescribeTags` API 呼叫來擷取標籤資訊，從而減少每秒的 API 交易數，並可讓您的標籤擷取隨控制的執行個體數量進行擴展。此外，在執行個體上執行的本機程序可以直接從執行個體中繼資料檢視執行個體的標籤資訊。

根據預設，不可從執行個體中繼資料取得標籤；您必須明確允許存取。您可以在執行個體啟動時允許存取，或在執行中或已停止執行個體上啟動後允許存取。您也可以透過在啟動範本中指定此許可來允許標籤的存取。使用範本啟動的執行個體允許存取執行個體中繼資料中的標籤。

若新增或移除執行個體標籤，會在執行個體執行期間更新執行個體中繼資料，而無需停止然後啟動執行個體。

任務

- 在執行個體中繼資料中啟用標籤存取權
- 從執行個體中繼資料擷取標籤
- 在執行個體中繼資料中停用標籤存取權

在執行個體中繼資料中啟用標籤存取權

根據預設，無法存取執行個體中繼資料中的執行個體標籤。針對每個執行個體，您必須明確啟用存取權。

> **Note**
>
> 如果您允許存取執行個體中繼資料中的標籤，執行個體標籤金鑰會受到特定限制。不合規會導致新執行個體的啟動失敗，或現有執行個體的錯誤。這些限制包括：
>
> - 只能包含字母 (a-z、A-Z)、數字 (0-9) 和下列字元：`+ - = . , _ : @`。
> - 不可含有空格或 `/`。
> - 不能只包含 `.` (一個句點)、 `..` (兩個句點) 或 `_index`。

如需詳細資訊，請參閱標籤限制。

Console

如需執行個體啟動期間在執行個體中繼資料中啟用標籤存取權

1. 請遵循該程序來啟動執行個體。
2. 展開進階詳細資訊，對於中繼資料中的允許標籤，請選擇啟用。
3. 在 Summary (摘要) 面板中，檢閱您的執行個體組態，然後選擇 Launch instance (啟動執行個體)。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。

如需執行個體啟動之後在執行個體中繼資料中啟用標籤存取權

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取您的執行個體，然後依次選取動作、執行個體設定、允許執行個體中繼資料中的標籤。
4. 要允許存取執行個體中繼資料中的標籤，請選取允許核取方塊。
5. 選擇儲存。

AWS CLI

如需執行個體啟動期間在執行個體中繼資料中啟用標籤存取權

請使用 run-instances 命令，並新增下列 `--metadata-options` 選項。

```
--metadata-options "InstanceMetadataTags=enabled"
```

如需執行個體啟動之後在執行個體中繼資料中啟用標籤存取權

請使用下面的 modify-instance-metadata-options 命令。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags enabled
```

如需確認在執行個體中繼資料中是否已啟用標籤存取權

請使用 describe-instances 命令並檢查 InstanceMetadataTags 的值。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].MetadataOptions[].InstanceMetadataTags"
```

以下為範例輸出。此值為 enabled 或 disabled。

```
[
    "enabled"
]
```

PowerShell

如需執行個體啟動期間在執行個體中繼資料中啟用標籤存取權

請使用 New-EC2Instance cmdlet 並新增下面的 -MetadataOptions_InstanceMetadataTags 參數。

```
-MetadataOptions_InstanceMetadataTags enabled
```

如需執行個體啟動之後在執行個體中繼資料中啟用標籤存取權

使用 Edit-EC2InstanceMetadataOption cmdlet。

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTags enabled
```

如需確認在執行個體中繼資料中是否已啟用標籤存取權

請使用 Get-EC2Instance cmdlet，然後檢查 InstanceMetadataTags 的值。

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.MetadataOptions.InstanceMetadataTags.Value
```

以下為範例輸出。此值為 enabled 或 disabled。

```
enabled
```

從執行個體中繼資料擷取標籤

如果您允許存取執行個體中繼資料中的執行個體標籤，則可以從執行個體中繼資料存取 `tags/instance` 類別。如需詳細資訊，請參閱[存取 EC2 執行個體的執行個體中繼資料]。

IMDSv2

Linux

透過 Linux 執行個體執行下面的命令，以便列出執行個體的全部標籤索引鍵。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance
```

此範例可取得前面範例中獲得的索引鍵的值。IMDSv2 請求使用在之前範例命令中建立的儲存字符，前提是字符並未過期。字符不得過期。

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

透過 Windows 執行個體執行下面的 cmdlet，以便列出執行個體的全部標籤索引鍵。

```
$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

此範例可取得前面範例中獲得的索引鍵的值。IMDSv2 請求使用在之前範例命令中建立的儲存字符，前提是字符並未過期。字符不得過期。

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

IMDSv1

Linux

透過 Linux 執行個體執行下面的命令，以便列出執行個體的全部標籤索引鍵。

```
curl http://169.254.169.254/latest/meta-data/tags/instance
```

此範例可取得前面範例中獲得的索引鍵的值。

```
curl http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

透過 Windows 執行個體執行下面的 cmdlet，以便列出執行個體的全部標籤索引鍵。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

此範例可取得前面範例中獲得的索引鍵的值。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

在執行個體中繼資料中停用標籤存取權

您可在執行個體中繼資料中停用對執行個體標籤的存取權。您不需要在啟動時停用對執行個體中繼資料上的執行個體標籤的存取權，因為根據預設，它處於關閉狀態。

Console

如需在執行個體中繼資料中停用標籤存取權

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後依次選取 Actions (動作)、Instance settings (執行個體設定)、Allow tags in instance metadata (允許執行個體中繼資料中的標籤)。

4. 要允許存取執行個體中繼資料中的標籤，請選擇允許核取方塊。
5. 選擇儲存。

AWS CLI

如需在執行個體中繼資料中停用標籤存取權

請使用下面的 modify-instance-metadata-options 命令。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags disabled
```

PowerShell

如需在執行個體中繼資料中停用標籤存取權

使用 Edit-EC2InstanceMetadataOption cmdlet。

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTag disabled
```

Amazon EC2 服務配額

當您建立 時 AWS 帳戶，我們會根據您的區域為您的 AWS 資源設定預設配額 （也稱為限制）。如果您嘗試超過資源的配額，請求會失敗。例如，您可以為區域中的隨需執行個體佈建的 Amazon EC2 vCPUs 數量上限。如果您嘗試在 區域中啟動執行個體，而此請求會導致您的用量超過此配額，則請求會失敗。如果發生這種情況，您可以減少資源用量或請求增加配額。

Service Quotas 主控台是一個集中位置，您可以在其中檢視和管理 AWS 服務的配額，並請求增加您使用的許多資源的配額。使用我們提供的配額資訊來管理您的 AWS 基礎設施。計劃在您需要的時候，先行請求提高配額。

相關配額文件

- Amazon EC2 端點與配額
- Amazon EC2 執行個體類型配額
- Amazon EBS 的配額

- Amazon VPC 配額

檢視您目前的配額

可以使用 Service Quotas 主控台檢視每個區域的配額。

使用 Service Quotas 主控台檢視目前的配額

1. 在 https：//https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/ 開啟 Service Quotas 主控台。
2. 從導覽列 (位於畫面頂部) 中，選取一個區域。
3. 在管理配額窗格中選取服務，例如，Amazon Elastic Compute Cloud (Amazon EC2)。選擇檢視配額。
4. 使用篩選欄位，按照資源名稱篩選清單。例如，輸入 **On-Demand** 找出隨需執行個體的配額。
5. 若要檢視詳細資訊，請選擇配額名稱以開啟配額的詳細資訊頁面。

請求增加

您可以請求增加每個區域的配額。

使用 Service Quotas 主控台請求提高

1. 在 https：//https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/ 開啟 Service Quotas 主控台。
2. 從導覽列 (位於畫面頂部) 中，選取一個區域。
3. 使用篩選欄位，按照資源名稱篩選清單。例如，輸入 **On-Demand** 找出隨需執行個體的配額。
4. 如果配額可調整，請選取配額，然後選擇請求提高配額。
5. 對於變更配額值，輸入新配額值。
6. 選擇請求。
7. 若要檢視主控台中任何擱置或最近解決的請求，請從導覽窗格中選擇儀表板。對於擱置的請求，請選擇請求狀態以開啟請求回條。請求的初始狀態為 Pending (待定)。狀態變更為請求的配額後，您會看到案例編號 支援。選擇案例編號，為請求開啟票證。

如需詳細資訊，包括如何使用 AWS CLI 或 SDKs請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

使用通訊埠 25 傳送的電子郵件限制

Amazon EC2 預設僅允許透過連接埠 25 將流量傳出至私有 IPv4 位址。在公有 IPv4 位址與 IPv6 位址上會封鎖透過連接埠 25 傳輸的流量。

您可以請求移除此限制。如需詳細資訊，請參閱如何從我的 Amazon EC2 執行個體或 Lambda 函數中移除連接埠 25 的限制？

此限制不會適用於透過連接埠 25 傳出至下列目的地的流量：

- 存在來源網路介面的 VPC 主 CIDR 區塊中的 IP 地址。
- 在 RFC 1918、RFC 6598 和 RFC 4193 中定義的 CIDR 內的 IP 地址。

監控 Amazon EC2 資源

監控是維護 Amazon EC2 執行個體和 AWS 解決方案可靠性、可用性和效能的重要部分。您應該從 AWS 解決方案中的所有部分收集監控資料，以便在發生多點失敗時更輕鬆地偵錯。

AWS 提供各種可用來監控 Amazon EC2 的工具。Amazon EC2 和 CloudWatch 主控台儀表板可提供 Amazon EC2 環境狀態的快速瀏覽檢視。此外，我們提供下列項目：

- 系統狀態檢查 – 監控使用執行個體所需的 AWS 系統，以確保它們正常運作。這些檢查會偵測執行個體中需要 AWS 參與修復的問題。系統狀態檢查失敗時，您可以選擇等待 AWS 修正問題，也可以自行解決問題 (例如停止並重新啟動，或終止並取代執行個體)。導致系統狀態檢查失敗的問題範例包括：
  - 網路連線中斷
  - 系統電力中斷
  - 實體主機的軟體問題
  - 實體主機上會影響網路連線的硬體問題

  如需詳細資訊，請參閱 Amazon EC2 執行個體的狀態檢查。
- 執行個體狀態檢查 – 監控個別執行個體的軟體和網路組態。這些檢查會偵測需要您介入修復的問題。一般來說，當執行個體狀態檢查失敗時，您需要自行處理問題 (例如重新啟動執行個體，或在作業系統中進行修改)。可能導致執行個體狀態檢查失敗的問題範例包括：
  - 系統狀態檢查失敗
  - 網路或啟動組態的設定不正確
  - 記憶體用盡
  - 檔案系統毀損
  - 核心不相容

  如需詳細資訊，請參閱 Amazon EC2 執行個體的狀態檢查。
- Amazon CloudWatch 警示 – 監看指定時段內的單一指標，並根據與多個時段內給定之閾值相對的指標值來執行一或多個動作。此動作是傳送到 Amazon Simple Notification Service (Amazon SNS) 主題或 Amazon EC2 Auto Scaling 政策的通知。警示僅會針對持續狀態變更呼叫動作。CloudWatch 警示不會只因處於特定狀態就叫用動作，狀態必須已變更並已維持一段指定的時間。如需詳細資訊，請參閱使用 CloudWatch 監控您的執行個體。

- Amazon EventBridge 事件 – 自動化您的 AWS 服務並自動回應系統事件。來自 AWS 服務的事件會以近乎即時的方式交付至 EventBridge，而且您可以指定當事件符合您撰寫的規則時要採取的自動化動作。如需詳細資訊，請參閱the section called “使用 EventBridge 進行自動化”。
- AWS CloudTrail 日誌 – 擷取對 Amazon EC2 API 發出的呼叫的詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些 CloudTrail 日誌來判斷提出了哪些呼叫、提出呼叫的來源 IP 位址、提出呼叫的人員及時間等。如需詳細資訊，請參閱the section called “使用 CloudTrail 記錄 API 呼叫”。
- CloudWatch 代理程式 – 會從 EC2 執行個體和內部部署伺服器上的主機和訪客收集記錄和系統層級度量。如需詳細資訊，請參閱 「Amazon CloudWatch 使用者指南」 中的使用 CloudWatch 代理程式從 Amazon EC2 執行個體和現場部署伺服器收集指標和日誌。

監控您的 Amazon EC2 執行個體狀態

您可以透過檢視您執行個體的狀態檢查和排程事件，監控您執行個體的狀態。

狀態檢查可為您提供由 Amazon EC2 進行之自動化檢查的結果資訊。這些自動化檢查會偵測是否有特定問題正在影響您的執行個體。狀態檢查資訊搭配 Amazon CloudWatch 提供的資料，可給予您每個執行個體的詳細操作可見性。

也可以查看為您的執行個體排程之特定事件的狀態。事件的狀態提供為您的執行個體計畫之近期活動的相關資訊，例如重新開機或淘汰。也提供每個事件的排程開始時間和結束時間。

目錄

- Amazon EC2 執行個體的狀態檢查
- 應用程式狀態檢查
- Amazon EC2 執行個體的狀態變更事件
- Amazon EC2 執行個體的排程事件

Amazon EC2 執行個體的狀態檢查

透過狀態檢查，您可以快速判斷是否有問題可能導致您的執行個體無法執行您的應用程式。Amazon EC2 提供四種類型的狀態檢查：系統、執行個體、連接的 EBS 和應用程式。系統、執行個體和連接的 EBS 狀態檢查由 Amazon EC2 管理，並在每個執行個體上自動執行。應用程式狀態檢查為選擇加入，並監控執行個體上執行之應用程式的 HTTP 或 HTTPS 回應。您可以在 Amazon EC2 主控台 AWS CLI、 或 AWS SDKs 中檢視所有狀態檢查的結果。

狀態檢查會每分鐘執行一次，並會傳回通過或失敗狀態。如果所有檢查都通過，執行個體的整體狀態即為 OK (正常)。若一或多個檢查失敗，整體狀態便會是 impaired (故障)。系統、執行個體和連接的 EBS 狀態檢查由 Amazon EC2 管理，無法停用或刪除。應用程式狀態檢查是選擇加入；您可以自行建立、關聯和管理它們。

當狀態檢查失敗時，狀態檢查對應的 CloudWatch 指標會遞增。如需詳細資訊，請參閱 狀態檢查指標。但是，您可以使用這些指標，建立根據狀態檢查結果而觸發的 CloudWatch 警示。例如，您可以建立警示，在狀態檢查於特定執行個體上失敗時警告您。如需詳細資訊，請參閱 為狀態檢查失敗的 Amazon EC2 執行個體建立 CloudWatch 警示。

您也可以建立 Amazon CloudWatch 提醒以監控 Amazon EC2 執行個體，並且在執行個體因為基礎問題故障時，自動復原執行個體。如需詳細資訊，請參閱 自動執行個體復原功能。

內容

- 狀態檢查類型
- 檢視 Amazon EC2 執行個體的狀態檢查
- 為狀態檢查失敗的 Amazon EC2 執行個體建立 CloudWatch 警示

狀態檢查類型

狀態檢查有四種類型。

- 系統狀態檢查
- 執行個體狀態檢查
- 附接的 EBS 狀態檢查
- 應用程式狀態檢查

系統狀態檢查

系統狀態檢查會監控執行個體執行所在的 AWS 系統。這些檢查會偵測您執行個體需要 AWS 介入修復的基礎問題。當系統狀態檢查失敗時，您可以選擇等待 AWS 來修正問題，也可以自行解決問題。針對 Amazon EBS 後端執行個體，您可以自行停止並啟動執行個體，多數情況下，這會將它遷移到新的主機。針對執行個體儲存體支援的執行個體 (僅支援 Linux 執行個體)，您可以終止並取代執行個體。請注意，執行個體存放區是暫時的，當執行個體停止時，所有資料都會丟失。

下列為可能導致系統狀態檢查失敗的問題範例：

- 網路連線中斷

- 系統電力中斷
- 實體主機的軟體問題
- 實體主機上會影響網路連線的硬體問題

如果系統狀態檢查失敗，則會遞增 StatusCheckFailed_System 指標。

裸機執行個體

如果您在裸機執行個體上執行重新啟動，系統狀態檢查可能會暫時傳回失敗狀態。當執行個體變得可用時，系統狀態檢查應傳回通過狀態。

執行個體狀態檢查

執行個體狀態檢查會監控個別執行個體的軟體和網路連接性。Amazon EC2 會將地址解析協定 (ARP) 請求傳送至網路介面 (NIC)，以便檢查執行個體的運作狀態。這些檢查會偵測需要您介入修復的問題。當執行個體狀態檢查失敗時，通常您需要自行處理問題 (例如重新開機執行個體或對執行個體的組態進行變更)。

> **Note**
>
> 使用 `systemd-networkd` 進行網路設定的最新 Linux 發行版本可能會以不同於早期發行版本的方式回報運作狀態檢查。在開機過程中，此類型的網路可以更早啟動，並有可能在其他可能也會影響執行個體健康的啟動工作之前完成。取決於網路可用性的狀態檢查可在其他工作完成之前，回報運作狀態。

下列為可能導致執行個體狀態檢查失敗的問題範例：

- 系統狀態檢查失敗
- 網路或啟動組態不正確
- 記憶體用盡
- 檔案系統毀損
- 核心不相容
- 在重新啟動期間，執行個體狀態檢查可以報告失敗，直到執行個體再次可用為止。

如果執行個體狀態檢查失敗，則會遞增 StatusCheckFailed_Instance 指標。

裸機執行個體

如果您在裸機執行個體上執行重新啟動，執行個體狀態檢查可能會暫時傳回失敗狀態。當執行個體變得可用時，執行個體狀態檢查應傳回通過狀態。

附接的 EBS 狀態檢查

附接的 EBS 狀態檢查可監控附接至執行個體的 Amazon EBS 磁碟區是否可連線且能夠完成 I/O 操作。此 `StatusCheckFailed_AttachedEBS` 指標是二進位值，如果附接至執行個體的一個或多個 EBS 磁碟區無法完成 I/O 操作，則表示損壞。這些狀態檢查會對運算或 Amazon EBS 基礎設施的潛在問題進行偵測。當連接的 EBS 狀態檢查指標失敗時，您可以等待 AWS 解決問題，也可以採取動作，例如取代受影響的磁碟區，或停止和重新啟動執行個體。

以下是可能導致附接的 EBS 狀態檢查失敗的問題範例：

- EBS 磁碟區之下儲存子系統上的硬體或軟體問題
- 實體主機上會影響 EBS 磁碟區連線的硬體問題
- 執行個體與 EBS 磁碟區之間的連線問題

可以使用 `StatusCheckFailed_AttachedEBS` 指標來協助改善工作負載的彈性。您可以使用此指標，建立根據狀態檢查結果而觸發的 Amazon CloudWatch 警示。例如，偵測到長期影響時，可容錯移轉至次要執行個體或可用區域。或者，可以使用 EBS CloudWatch 指標來監控每個附接磁碟區的 I/O 效能，以偵測並取代受損磁碟區。如果 EBS 狀態檢查指出受損，且您的工作負載並未將 I/O 驅動至連接的 EBS 磁碟區，請停止並啟動執行個體，將其移至新的主機。這可以解決影響 EBS 磁碟區連線能力的基礎主機問題。如需詳細資訊，請參閱 Amazon EBS 的 Amazon CloudWatch 指標。

您也可以設定 Amazon EC2 Auto Scaling 群組來偵測連接的 EBS 狀態檢查失敗，然後將受影響的執行個體取代為新的執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用受損的 Amazon EBS 磁碟區監控和取代自動擴展執行個體。

> **Note**
>
> 附接的 EBS 狀態檢查指標僅適用於 Nitro 執行個體。

應用程式狀態檢查

使用應用程式狀態檢查來監控在 Amazon EC2 執行個體上執行之應用程式的網路連線能力和可用性。透過應用程式狀態檢查，您可以在應用程式層級監控 HTTP 和 HTTPS 回應。應用程式狀態檢查只會在您與其建立關聯的執行個體上執行。

您可以將每個應用程式狀態檢查設定為請求應用程式端點上的 HTTP 路徑，並定義指出良好回應的回應代碼。

您可以在應用程式重新啟動或部署期間隱藏應用程式狀態檢查，以避免在預期的停機時間期間發生誤判失敗。如需詳細資訊，請參閱處理部署、就地修補和替換。

以下是可能導致應用程式狀態檢查失敗的問題範例：

- 應用程式程序當機或停止回應
- 應用程式傳回 HTTP 回應碼，指出內部應用程式服務錯誤或其他問題
- 無法連線應用程式連接埠
- 軟體或基礎硬體問題導致您的執行個體失敗且沒有回應

如果應用程式狀態檢查失敗，我們會遞增 `StatusCheckFailed_Application` 指標。

如需設定應用程式狀態檢查的資訊，請參閱 應用程式狀態檢查。

檢視 Amazon EC2 執行個體的狀態檢查

如果執行個體狀態檢查失敗，通常就需要您自行處理問題 (例如重新啟動執行個體或對執行個體的組態進行變更)。若要自行對系統或執行個體狀態檢查失敗進行故障診斷，請參閱狀態檢查失敗的疑難排解 Amazon EC2 Linux 執行個體。

Console

檢視狀態檢查

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 在執行個體頁面上，狀態檢查資料欄會列出每個執行個體的操作狀態。
4. 若要檢視特定執行個體的狀態，請選取執行個體，然後選擇狀態和警示索引標籤。
5. 若要檢閱狀態檢查的 CloudWatch 指標，請在狀態和警示索引標籤上展開指標，以查看下列指標的圖表：

   - 系統狀態檢查失敗
   - 執行個體狀態檢查失敗
   - 連接的 EBS 狀態檢查失敗
   - 應用程式的狀態檢查失敗

如需詳細資訊，請參閱the section called “狀態檢查指標”。

AWS CLI

檢視狀態檢查

使用 describe-instance-status 命令。

範例：取得所有執行中執行個體的狀態

```
aws ec2 describe-instance-status
```

範例：取得所有執行個體的狀態

```
aws ec2 describe-instance-status --include-all-instances
```

範例：取得單一執行中執行個體的狀態

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

範例：取得狀態為 impaired 的所有執行個體

```
aws ec2 describe-instance-status \
--filters Name=instance-status.status,Values=impaired
```

PowerShell

檢視狀態檢查

使用 Get-EC2InstanceStatus 命令。

範例：取得所有執行中執行個體的狀態

```
Get-EC2InstanceStatus
```

範例：取得所有執行個體的狀態

```
Get-EC2InstanceStatus -IncludeAllInstance $true
```

範例：取得單一執行中執行個體的狀態

```
Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0
```

範例：取得狀態為 impaired 的所有執行個體

```
Get-EC2InstanceStatus \
-Filter @{Name="instance-status.status"; Values="impaired"}
```

為狀態檢查失敗的 Amazon EC2 執行個體建立 CloudWatch 警示

您可以使用狀態檢查指標來建立 CloudWatch 警示，在執行個體狀態檢查失敗時通知您。

如果缺少指標資料點，狀態檢查和狀態檢查警示可能會暫時進入資料不足的狀態。儘管很少見，但當指標報告系統中斷時，即使執行個體狀態良好，也可能會發生這種情況。建議您將此狀態視為遺失資料，而不是狀態檢查失敗或警示違規。這在回應中對執行個體執行停止、終止、重新啟動或復原動作時尤其重要。

Console

此範例會設定警示，它在執行個體未通過狀態檢查時傳送通知。可以選擇性地停止、終止或復原執行個體。

建立狀態檢查警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，選擇狀態檢查標籤，然後選擇動作 > 建立狀態檢查警示。
4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示)頁面上，於 Add or edit alarm (新增或編輯警示)下，選擇 Create an alarm (建立警示)。
5. 針對警示通知，開啟切換開關以設定 Amazon Simple Notification Service (Amazon SNS) 通知。選取現有 Amazon SNS 主題或輸入名稱以建立新主題。

   若將電子郵件地址新增到收件人清單，或是建立新主題，Amazon SNS 會將確認電子郵件傳送至每個新地址。每個收件人都必須選擇電子郵件中的確認連結。僅确认的地址会接收提醒通知。
6. 針對警示動作，開啟切換開關，以指定觸發警示時要採取的動作。選取動作。

7. 針對 Alarm thresholds (警示閾值)，指定警示的指標和準則。

   您可保留範例分組依據 (Average) 和 要做為範例的資料類型 (Status check failed:either) 的預設設定，或者可以根據您的需求進行變更。

   針對連續期間，設定要評估的期間數，然後在期間中，輸入在觸發提醒及傳送電子郵件前的評估期間長度。
8. (選用) 針對 Sample metric data (範例指標資料)，選擇 Add to dashboard (新增至儀表板)。
9. 選擇建立。

若需要變更執行個體狀態警示，您可以編輯它。

編輯狀態檢查警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選擇 Actions (動作)、Monitoring (監控)、Manage CloudWatch alarms (管理 CloudWatch 警示)。
4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示)頁面上，於 Add or edit alarm (新增或編輯警示)下，選擇 Edit an alarm (編輯警示)。
5. 針對 Search for alarm (搜尋警示)，選擇警示。
6. 完成變更後，請選擇 Update (更新)。

AWS CLI

在下列範例中，當執行個體的執行個體檢查或系統檢查在至少兩個連續期間中失敗時，警示便會發佈通知到 SNS 主題。使用的 CloudWatch 指標為 `StatusCheckFailed`。

建立狀態檢查警示

1. 選取現有的 SNS 主題，或建立新主題。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的存取 AWS CLI中的 Amazon SNS。
2. 使用下列 list-metrics 命令檢視 Amazon EC2 可用的 Amazon CloudWatch 指標。

   ```
   aws cloudwatch list-metrics --namespace AWS/EC2
   ```
3. 使用下列 put-metric-alarm 命令建立警示。

```
aws cloudwatch put-metric-alarm \
    --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 \
    --metric-name StatusCheckFailed \
    --namespace AWS/EC2 \
    --statistic Maximum \
    --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
    --unit Count \
    --period 300 \
    --evaluation-periods 2 \
    --threshold 1 \
    --comparison-operator GreaterThanOrEqualToThreshold \
    --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

期間為收集 Amazon CloudWatch 指標的時間範圍，以秒為單位。此範例使用 300，即 60 秒乘以 5 分鐘。評估期間為針對指標的值與閾值進行比較的連續期間數。此範例使用 2。警示動作為觸發此警示時要執行的動作。

PowerShell

建立狀態檢查警示

當執行個體在至少兩個連續時段未通過狀態檢查時，請使用 Write-CWMetricAlarm cmdlet 將通知發佈至 SNS 主題，如下所示。

```
Write-CWMetricAlarm `
    -AlarmName "StatusCheckFailed-Alarm-for-i-1234567890abcdef0" `
    -MetricName "StatusCheckFailed" `
    -Namespace "AWS/EC2" `
    -Statistic "Maximum" `
    -Dimension @{Name="InstanceId"; Values="i-1234567890abcdef0"} `
    -Unit "Count" `
    -Period 300 `
    -EvaluationPeriod 2 `
    -Threshold 1 `
    -ComparisonOperator "GreaterThanOrEqualToThreshold" `
    -AlarmAction "arn:aws:sns:us-west-2:111122223333:my-sns-topic"
```

期間為收集 Amazon CloudWatch 指標的時間範圍，以秒為單位。此範例使用 300，即 60 秒乘以 5 分鐘。評估期間為針對指標的值與閾值進行比較的連續期間數。此範例使用 2。警示動作為觸發此警示時要執行的動作。

應用程式狀態檢查

應用程式狀態檢查可協助您監控在 Amazon EC2 上執行之應用程式的效能和運作狀態。透過應用程式狀態檢查，您可以透過可設定的路徑和連接埠監控應用程式，以偵測和回應應用程式運作狀態受損。例如，您可以使用應用程式狀態檢查來確認您的 Web 伺服器正在接聽其預期的連接埠，並接受新的連線。

應用程式狀態檢查會在可設定的路徑和連接埠監控應用程式的 HTTP 和 HTTPS 回應。它們每 60 秒執行一次並與 Amazon EC2 Auto Scaling 整合，因此您可以自動替換應用程式受損的執行個體。

目錄

- 應用程式狀態檢查的運作方式
- 開始使用應用程式狀態檢查
- 組態選項
- 預設設定
- Amazon EC2 Auto Scaling 整合
- 處理部署、就地修補和替換
- 測試新的應用程式狀態檢查
- 進階聯網
- 最佳實務
- 疑難排解
- 監控應用程式狀態檢查
- 安全與許可
- 定價
- 配額

應用程式狀態檢查的運作方式

應用程式狀態檢查會每 60 秒將 HTTP 或 HTTPS 請求傳送至接聽執行個體上網路連接埠的端點。 會將回應碼與您設定的狀態碼比對器 AWS 進行比較。檢查會在連續多次失敗的請求後標示為受損，並在連續多次成功請求後再次正常運作。兩個計數都預設為 2 且可設定。如需詳細資訊，請參閱評估閾值。

> **Note**
>
> 應用程式狀態檢查會透過 HTTP/2 傳送運作狀態檢查請求。

HTTPS 通訊協定檢查不會驗證伺服器憑證。

在重新啟動期間，應用程式狀態檢查會報告失敗，直到執行個體再次可用為止，因為應用程式在作業系統重新啟動時無法回應運作狀態檢查請求。

網路架構

應用程式狀態檢查源自 Amazon EC2 應用程式狀態檢查服務。若要到達您的執行個體， 會在您的 VPC 中 AWS 建立受管彈性網路界面 (ENI)。 會為每個具有相關聯執行個體的來源子網路和安全群組組合 AWS 建立一個 ENI。當應用程式狀態檢查首先需要該組合時， 會 AWS 建立受管 ENI，並在不需要剩餘的應用程式狀態檢查時將其移除。受管 ENI 不會計入您的執行個體 ENI 限制，但會計入您帳戶的每個區域網路介面配額，該配額會依每個可用區域強制執行。如需詳細資訊，請參閱 Amazon VPC 配額。

根據預設，Amazon EC2 會從主控台隱藏這些受管網路介面，並為在此設定可供使用之前沒有受管資源的帳戶隱藏 API 清單操作。若要變更其可見性，請參閱受管資源可見性設定。

AWS 在關聯的執行個體中，為來源子網路和安全群組的每個組合建立一個受管網路介面。受管界面的數量會隨著受監控執行個體使用的不同子網路和安全群組組合數量而增加。將受監控的執行個體合併為較少的子網路和安全群組組合，可減少受管介面的數量。例如，200 個執行個體分散在 2 個子網路中，所有子網路都使用單一安全群組來產生 2 個受管界面。在這 2 個子網路中使用 3 個安全群組的相同 200 個執行個體會產生最多 6 個受管界面。每個界面對應至一個子網路和安全群組組合。

應用程式狀態檢查會從 VPC 中的私有優勢點到達您的執行個體。範圍說明檢查的來源，而不是執行個體 IP 地址的屬性。 會在 VPC 內的子網路中 AWS 建立受管 ENI，並透過私有網路路徑到達執行個體。

透過 AWS 受管網路路徑，運作狀態檢查流量來自 AWS 與目標執行個體 （或本機區域目標的父可用區域） 位於相同可用區域中的受管 Amazon EC2 執行個體。流量會透過 AWS 內部網路傳輸，而不會周遊公有網際網路。如需詳細資訊，請參閱 Amazon Web Services 網站上的 Amazon VPC FAQs。

使用客戶管理的網路路徑，您可以選擇來源子網路，以便從與目標不同的可用區域執行檢查。如需詳細資訊，請參閱the section called “跨可用區域監控”。

AWS 受管和客戶受管網路路徑

應用程式狀態檢查支援兩種加入模式，可判斷誰會選取運作狀態檢查 ENI 的來源子網路和安全群組，以及目標執行個體的目的地子網路和安全群組。

AWS 受管網路路徑

AWS 會選取運作狀態檢查 ENI 的來源子網路和安全群組，以及目標執行個體的目的地子網路和安全群組。

客戶管理的網路路徑

您可以指定運作狀態檢查 ENI 的來源子網路和安全群組，以及目標執行個體的目的地子網路和安全群組。

當您需要控制哪些子網路和安全群組運作狀態檢查流量源自時，例如當您的 VPC 具有嚴格的網路分割、防火牆規則或合規要求，限制哪些來源可以到達您的應用程式端點時，請使用客戶受管網路路徑。

您可以在建立命令中包含或省略 `--health-check-paths` 參數，以選擇 模式。如果您省略 `--health-check-paths` 參數， 會 AWS 選取來源和目的地子網路和安全群組 (AWS 受管網路路徑）。如果您包含 `--health-check-paths` 參數，您可以管理它們 （客戶管理的網路路徑）。

IP 版本

每個應用程式狀態檢查都與單一 IP 版本 (IPv4 或 IPv6) 相關聯。若要透過 IPv4 和 IPv6 監控執行個體，請建立兩個不同的應用程式狀態檢查，並將兩者與執行個體建立關聯。

檢查從您 VPC 中的 IPv4 和 IPv6 到達您的執行個體。

檢查狀態值

每個個別檢查都會報告下列其中一個狀態：

- `passed`：檢查已成功完成
- `failed`：檢查失敗。回應包含應用程式傳回的 HTTP 狀態碼。如需解釋和修復指引，請參閱 疑難排解。
- `initializing`：檢查尚未完成其第一次評估
- `insufficient-data`：檢查未收到足夠的資料來判斷結果
- `not-applicable`：檢查未與執行個體建立關聯

執行個體報告的整體應用程式狀態會彙總所有個別檢查結果。整體狀態為下列其中一項：

- ok：通過的所有檢查

- `impaired`：一或多個檢查失敗
- `initializing`：一或多個檢查尚未完成其第一次評估
- `insufficient-data`：一或多個檢查報告資料不足
- `not-applicable`：所有相關聯的應用程式狀態檢查都會從彙總中排除
- `suppressed`：執行個體的應用程式狀態檢查評估遭到隱藏

聚合

您可以將每個應用程式狀態檢查標記為包含在執行個體的整體狀態中或從中排除。根據預設，檢查為 `included`。

`included`

檢查有助於執行個體的整體狀態，而 Amazon EC2 Auto Scaling 會使用它。

`excluded`

檢查會報告其個別狀態，但不影響執行個體的整體狀態，而 Amazon EC2 Auto Scaling 不會使用它。使用此設定在生產環境中驗證新的檢查，而不會影響整體狀態或觸發 Amazon EC2 Auto Scaling 替換。這是將檢查新增至現有生產工作負載時的建議工作流程；請參閱 測試新的應用程式狀態檢查。

開始使用應用程式狀態檢查

先決條件

建立應用程式狀態檢查之前，請確定您有下列項目：

- 具有您要監控之執行個體的 VPC。
- 每個執行個體上的應用程式端點，可在您將設定的連接埠和 HTTP 路徑上回應 HTTP 或 HTTPS 請求。
- 每個目的地執行個體上的安全群組，允許檢查連接埠上來自應用程式狀態檢查所用來源安全群組的傳入流量。請參閱 安全與許可。

步驟 1：設定您的 應用程式

設定應用程式端點以回應連接埠上的 HTTP 或 HTTPS 請求，以及您在建立檢查時指定的 HTTP 路徑。傳回狀態碼比對器中包含的回應碼，以指出應用程式運作狀態良好。

確定目的地執行個體的安全群組允許來自應用程式狀態檢查所用來源安全群組之檢查連接埠的傳入流量。對於受管網路路徑， 會在建立檢查時 AWS 提供來源安全群組。對於客戶管理的網路路徑，您可以在建立檢查時指定來源安全群組。

步驟 2：建立檢查定義

使用 AWS CLI 建立應用程式狀態檢查。

Console

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的執行個體下，選擇應用程式狀態檢查。
3. 選擇建立應用程式狀態檢查。
4. 在運作狀態檢查邏輯下，設定下列項目：
   - 通訊協定：選擇 HTTP 或 HTTPS。
   - 連接埠：輸入應用程式接聽的連接埠。
   - 路徑 （選用）：輸入要請求的 HTTP 路徑，例如 `/healthcheck`。
   - IP 版本：選擇 IPv4 或 IPv6。
   - 裝置索引：要檢查的網路介面裝置索引。預設值為 `0`。
5. 在控制項和閾值下，設定逾時，並選擇性地設定狀態碼比對器、失敗閾值、成功閾值和初始化寬限期。檢查間隔固定為 60 秒。
6. 在彙總下，選擇包含檢查以對整體應用程式狀態做出貢獻，並驅動 Amazon EC2 Auto Scaling，或選擇排除以報告檢查，而不會影響整體狀態。
7. 在運作狀態檢查路徑下，保留不要指定網路路徑 （建議/預設），讓 Amazon EC2 將運作狀態檢查網路介面放置在執行個體的子網路中，或選擇指定網路路徑 （進階） 自行定義來源子網路、安全群組和目的地。
8. （選用） 新增名稱標籤和其他標籤。
9. 選擇建立應用程式狀態檢查。

AWS CLI

若要使用 AWS 受管網路路徑，請省略 `--health-check-paths` 參數，並讓 AWS 選取來源和目的地子網路和安全群組。

```
aws ec2 create-application-status-check \
```

```
        --protocol https \
        --port 443 \
        --path "/health" \
        --status-code-matcher "200"
```

若要使用客戶管理的網路路徑，請包含 --health-check-paths 參數。每個運作狀態檢查路徑都包含來源（運作狀態檢查 ENI 的子網路和安全群組）和一或多個目的地（目標執行個體的子網路和安全群組）。

```
aws ec2 create-application-status-check \
        --protocol https \
        --port 443 \
        --path "/health" \
        --status-code-matcher "200" \
        --health-check-paths '[{"Source":
{"SubnetId":"subnet-111","SecurityGroupId":"sg-aaa"},"Destinations":
[{"SubnetId":"subnet-222","SecurityGroupId":"sg-bbb"}]}]'
```

步驟 3：將檢查與執行個體建立關聯

將檢查與您要監控的執行個體建立關聯，無論是依執行個體 ID 或依標籤。

Console

1. 在導覽窗格的執行個體下，選擇應用程式狀態檢查，然後選取檢查。
2. 選擇管理狀態檢查關聯，然後選擇依資源 ID 管理關聯或依標籤管理關聯。
3. 若要與 Auto Scaling 群組中的所有執行個體建立關聯，請選擇依標籤管理關聯，然後輸入 aws:autoscaling:groupName做為標籤索引鍵，輸入您的 Auto Scaling 群組名稱做為值。
4. 選擇關聯。

AWS CLI

依執行個體 ID：

```
aws ec2 associate-application-status-check \
        --application-status-check-id asc-1234567890abcdef0 \
        --instance-ids i-0123456789abcdef0
```

依標籤：

```
aws ec2 associate-application-status-check \
        --application-status-check-id asc-1234567890abcdef0 \
        --target-tag-associations Key=Environment,Value=production
```

若要與 Auto Scaling 群組中的所有執行個體建立關聯，請使用aws:autoscaling:groupName系統標籤：

```
aws ec2 associate-application-status-check \
        --application-status-check-id asc-1234567890abcdef0 \
        --target-tag-associations Key=aws:autoscaling:groupName,Value=my-asg
```

關聯和取消關聯操作會傳回每個執行個體的成功和失敗結果。如果某些執行個體無法建立關聯（例如，因為檢查已建立關聯），這些執行個體會出現在失敗的結果中，並說明原因。

步驟 4：檢視結果

檢視每個執行個體應用程式的運作狀態。

Console

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，然後選擇狀態和警示索引標籤。
4. 在應用程式狀態檢查下，檢閱整體狀態和每個相關聯的檢查個別狀態。

AWS CLI

```
aws ec2 describe-application-status \
        --instance-ids i-0123456789abcdef0
```

回應包含整體應用程式狀態，以及針對每個相關聯的檢查、檢查狀態，以及針對失敗的檢查，包含應用程式傳回的 HTTP 狀態碼。

回應範例：

```
{
        "ApplicationStatuses": [
```

```
            {
                "InstanceId": "i-0123456789abcdef0",
                "ApplicationStatus": {
                    "Status": "ok",
                    "Details": [
                        {
                            "ApplicationStatusCheckId": "asc-1234567890abcdef0",
                            "Status": "passed",
                            "Reason": {
                                "Code": "ResponseCodeMatched",
                                "StatusCode": 200,
                                "Protocol": "HTTP"
                            }
                        }
                    ]
                }
            }
        ]
    }
```

若要檢視檢查定義（非每個執行個體狀態），請使用 describe-application-status-checks。此命令會傳回應用程式狀態檢查的組態，包括通訊協定、連接埠、HTTP 路徑和狀態碼比對器設定。

組態選項

應用程式狀態檢查接受數個組態參數。本節說明從參數名稱無法自我識別行為的參數。如需參數和驗證規則的完整清單，請參閱《Amazon EC2 API 參考》中的 CreateApplicationStatusCheck 和 AssociateApplicationStatusCheck。

評估閾值

`FailureThreshold`

檢查標記為受損之前連續失敗的請求數量。預設：2。

`SuccessThreshold`

在檢查再次標記為正常之前，連續成功的請求數量。預設：2。

`Timeout`

在請求記錄為失敗之前等待回應的秒數。強制執行為強制逾時；如果您的應用程式未在此視窗中回應，無論最終回應為何，請求都會記錄為失敗。預設：6. 有效範圍：1-30。

啟動寬限期

`InitializationGracePeriodSeconds`

在執行個體啟動後， AWS 開始評估檢查的等待秒數。使用此參數讓應用程式有時間在檢查開始之前開始接聽。如果寬限期太短，Amazon EC2 Auto Scaling 可能會在應用程式就緒之前取代新的執行個體。預設：300。有效範圍：1 到 600。

IP 範圍

`IpScope`

應用程式狀態檢查使用`private`範圍；檢查從您的 VPC 內執行。對於 IPv4，這對應到執行個體的私有 IP 地址。對於 IPv6， AWS 不會將地址分類為公有或私有；檢查會接受任何 IPv6 地址，並從 VPC 中評估該地址。

裝置索引

`DeviceIndex`

執行個體上 AWS 評估運作狀態檢查的網路裝置的索引。當您執行個體的主要網路裝置不是您要檢查的裝置時，請變更此選項。預設：0.

彙總、IP 版本和運作狀態檢查路徑（來源和目的地子網路和安全群組）涵蓋在此頁面稍早的自有區段中。

預設設定

透過 AWS 受管網路路徑，應用程式狀態檢查會使用下列預設值。

設定	預設
檢查間隔	60 秒（固定；無法設定）
Failure threshold	連續 2 次失敗
成功閾值	連續 2 次成功
Timeout (逾時)	6 秒

設定	預設
狀態碼比對器	200
HTTP 路徑	/
IP 版本	ipv4
IP 範圍	private
裝置索引	0
初始化寬限期	300 秒
聚合	包含
來源子網路和安全群組	管理者 AWS

Amazon EC2 Auto Scaling 整合

只要檢查包含在彙總中`impaired`，Amazon EC2 Auto Scaling 會自動終止和取代其整體應用程式狀態報告為 的執行個體。除了將應用程式狀態檢查與群組中的執行個體建立關聯之外，不需要 Auto Scaling 群組組態。

Amazon EC2 Auto Scaling 會使用執行個體的整體狀態，而非個別檢查狀態。標記的檢查`excluded`不會驅動 Amazon EC2 Auto Scaling 動作。`suppressed` 狀態中的檢查不會驅動 Amazon EC2 Auto Scaling 動作。

在檢查上使用 `InitializationGracePeriodSeconds` 參數，在應用程式狀態檢查開始之前，讓新的執行個體有時間啟動。如果寬限期太短，Amazon EC2 Auto Scaling 可能會終止新的執行個體，並在其應用程式準備好為流量提供服務之前予以取代。

檢查會`InitializationGracePeriodSeconds`設定執行個體啟動後多久，檢查才會開始評估應用程式。將其設定為涵蓋應用程式的啟動時間，因此檢查不會在應用程式仍在啟動`impaired`時報告。Auto Scaling 群組的運作狀態檢查寬限期是分開的。它會設定執行個體進入服務後多久，Amazon EC2 Auto Scaling 才會因運作狀態檢查失敗而終止它。

如需 Amazon EC2 Auto Scaling 如何使用運作狀態檢查的詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 Auto Scaling 群組中的執行個體運作狀態檢查和搭配 Auto Scaling 群組使用應用程式狀態檢查。 Amazon EC2 Auto Scaling

處理部署、就地修補和替換

部署、就地修補和其他維護操作可能會暫時停止或重新啟動您的應用程式。在此期間，應用程式狀態檢查會報告失敗，因為應用程式無法回應運作狀態檢查請求。如果您的執行個體位於具有包含在彙總中應用程式狀態檢查的 Auto Scaling 群組中，即使預期中斷，Amazon EC2 Auto Scaling 仍可能會終止並取代這些執行個體。

選項 A：隱藏檢查

對您知道持續時間的週框維護時段使用抑制。在執行個體層級強制執行抑制。您可以指定持續時間，或將其省略以隱藏檢查，直到您停用禁止為止。

AWS CLI

```
aws ec2 enable-application-status-check-suppression \
        --instance-ids i-0123456789abcdef0 \
        --duration-seconds 3600
```

回應會傳回每個執行個體的抑制何時開始，以及何時結束。部分成功是可能的；某些執行個體可能無法隱藏，並出現於回應中並說明原因。

若要在禁止時段到期之前繼續檢查：

```
aws ec2 disable-application-status-check-suppression \
        --instance-ids i-0123456789abcdef0
```

抑制時，執行個體的整體應用程式狀態會報告 `suppressed`。Amazon EC2 Auto Scaling 不會對`suppressed`執行個體採取行動。

選項 B：從彙總中排除檢查

如果您希望檢查繼續評估和報告其個別狀態，但不會影響整體狀態或觸發 Amazon EC2 Auto Scaling 動作，請將檢查的彙總設定設為 `excluded`。這適用於更長期的情況，例如推出新的檢查版本或驗證變更而不危及取代，以及您希望遙測繼續而不會對操作造成影響的情況。

如需詳細資訊，請參閱聚合。

選項 C：取消與檢查的關聯

將取消關聯用於更長期或無限期的移除。

```
aws ec2 disassociate-application-status-check \
        --application-status-check-id asc-1234567890abcdef0 \
        --instance-ids i-0123456789abcdef0
```

如果您依標籤建立關聯，請從執行個體中移除標籤以取消關聯。取消關聯後，執行個體的整體應用程式狀態會報告 not-applicable。

部署指引

部署是需要抑制的最常見維護案例。當您的部署工具具有部署前掛鉤和部署後掛鉤時，請使用抑制，以便在部署開始之前隱藏檢查，並在部署完成後停用抑制。

一般模式為：

1. 在預先部署掛鉤中，呼叫執行個體的 enable-application-status-check-suppression，持續時間涵蓋預期的部署時段。
2. 執行部署。
3. 在部署後掛鉤中，呼叫執行個體的 disable-application-status-check-suppression。

如果您的部署工具沒有掛鉤，請從叫用部署的 CI/CD 管道進行磁碟機抑制。

測試新的應用程式狀態檢查

您可以在生產環境中驗證新的應用程式狀態檢查，然後再開始對執行個體層級監控做出貢獻。當您建立檢查excluded時，將彙總設定設為 ，然後確認它報告預期的狀態和 HTTP 回應代碼。當您準備好時，請將設定變更為 ，included讓檢查有助於執行個體的整體狀態，並與 Amazon EC2 Auto Scaling 整合。

1. 建立將彙總設定設為 的檢查excluded。

   ```
   aws ec2 create-application-status-check \
           --protocol https \
           --port 443 \
           --path "/health" \
           --status-code-matcher "200" \
           --aggregation excluded
   ```

2. 將檢查與測試執行個體或生產機群的子集建立關聯。
3. 等待至少兩個檢查間隔（約兩分鐘），以允許檢查完成初始評估。

4. 使用 describe-application-status 來驗證檢查是否報告預期狀態和 HTTP 回應碼。

```
aws ec2 describe-application-status \
        --instance-ids i-0123456789abcdef0
```

5. 如果檢查如預期報告，請將彙總設定更新為 `included` ，讓檢查有助於執行個體的整體狀態，並驅動 Amazon EC2 Auto Scaling 動作。

```
aws ec2 modify-application-status-check \
        --application-status-check-id asc-1234567890abcdef0 \
        --aggregation included
```

進階聯網

應用程式狀態檢查來自您指定的來源子網路和安全群組 （或為您 AWS 選取） 中的受管 ENI。對於需要比單一來源組態更高可用性的工作負載，或在 Local Zones 或 Outposts 中執行的工作負載，請考慮下列模式。

跨可用區域監控

對於可用區域備援，您可以從多個可用區域執行運作狀態檢查。使用客戶管理的網路路徑，您可以使用 `--health-check-paths` 參數，定義其來源位於到達相同目的地執行個體之兩個不同可用區域中的運作狀態檢查路徑。從兩個可用區域進行監控，即使有一個可用區域無法使用，執行個體的運作狀態報告仍會持續運作。

下列範例會建立具有兩個運作狀態檢查路徑的檢查，其來源位於不同的可用區域，兩者都到達相同的目的地執行個體。

```
aws ec2 create-application-status-check \
        --protocol https \
        --port 443 \
        --path "/health" \
        --status-code-matcher "200" \
        --health-check-paths '[{"Source":{"SubnetId":"subnet-source-
az1","SecurityGroupId":"sg-healthcheck"},"Destinations":[{"SubnetId":"subnet-
app-az1","SecurityGroupId":"sg-app"}]},{"Source":{"SubnetId":"subnet-source-
az2","SecurityGroupId":"sg-healthcheck"},"Destinations":[{"SubnetId":"subnet-app-
az2","SecurityGroupId":"sg-app"}]}]'
```

本機區域

對於在 AWS Local Zones 中執行的執行個體，受管彈性網路界面 (ENI) 位於父 AWS 區域，而非 Local Zone。父區域與本地區域執行個體之間的運作狀態檢查流量會周遊本地區域服務連結，這可能會產生額外的資料傳輸費用。

最佳實務

- 設計您的運作狀態端點，以反映在該執行個體上執行之應用程式的運作狀態。當您的端點根據應用程式本身傳回運作狀態時，Amazon EC2 Auto Scaling 只會取代真正受損的執行個體。如果端點的回應也取決於共用資源，例如資料庫或下游服務，則該資源的問題可能會一次在多個執行個體間失敗檢查。這可能會觸發整個機群的替換。如需撰寫運作狀態檢查端點的指引，請參閱 Amazon Builders' Library中的實作運作狀態檢查。
- 防止相互關聯的故障。隨附的檢查會驅動 Amazon EC2 Auto Scaling 替換。在多個執行個體中一次失敗的檢查可能會觸發一波取代。在 Auto Scaling 群組上設定執行個體維護政策，以限制同時取代的執行個體數量。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的執行個體維護政策。
- 執行個體受損計數的警示。在機群的 `StatusCheckFailed_Application` 指標上建立 Amazon CloudWatch 警示。許多執行個體突然上升表示共用相依性，而不是個別執行個體故障，並讓您在取代層疊之前有時間回應。如需詳細資訊，請參閱the section called “監控應用程式狀態檢查”。
- 保持在網路介面配額內。應用程式狀態檢查會建立受管網路介面，以計入每個區域配額的網路介面。AWS 會強制執行每個可用區域的此配額。監控您的用量，讓成長中的機群不會達到配額。如果是這樣， AWS 則無法建立新的介面。如需相關配額警示指引，請參閱 the section called “配額”。
- 將狀態檢查許可視為變更控制。建立、修改、刪除、關聯、取消關聯和隱藏應用程式狀態檢查的 IAM 動作可能會影響執行個體可用性。這些動作會決定驅動 Amazon EC2 Auto Scaling 取代的因素。將 `ec2:CreateApplicationStatusCheck`、`ec2:ModifyApplicationStatusCheck`、`ec2:AssociateApplicationStatusCheck`和 等動作`ec2:EnableApplicationStatusCheckSuppression`視為變更控制，而不是將其視為一般 Amazon EC2 存取的一部分。如需動作的完整清單，請參閱 Amazon EC2 API 參考。

疑難排解

當應用程式狀態檢查報告受損，但您預期應用程式運作狀態良好時，請確認下列各項：

1. 執行個體連線能力。確認執行個體的執行個體和系統狀態檢查為 ok。

2. 安全群組傳入規則。目的地執行個體的安全群組必須允許來自應用程式狀態檢查所用來源安全群組之檢查連接埠的傳入流量。對於 AWS 受管網路路徑， AWS 會提供來源安全群組；對於客戶受管網路路徑，請使用您指定為來源的安全群組。
3. 主機防火牆。執行個體上的任何主機層級防火牆 (iptable、Windows Firewall、第三方主機防火牆）必須允許檢查連接埠上的傳入流量。
4. 應用程式端點。應用程式必須接聽您設定的連接埠和路徑。使用來自執行個體的本機請求進行確認 (`curl http://localhost:PORT/PATH`)。
5. 通訊協定不相符。如果檢查設定為 HTTPS，但端點僅提供 HTTP （反之亦然），則所有呼叫都會失敗。
6. 狀態碼比對程式。確認應用程式的實際回應碼包含在您設定的狀態碼比對程式中。
7. 網路路徑。如果您已設定客戶管理的網路路徑，請確認來源子網路和安全群組已連線至目的地子網路。使用 VPC Reachability Analyzer 追蹤網路路徑。
8. 可用的 ENI quota. AWS 在您的帳戶中為每個來源子網路和安全群組組合建立受管彈性網路界面 (ENI)。確認您的帳戶尚未達到每個區域的網路介面配額，該配額會依可用區域強制執行。如果您的帳戶已達到此配額， AWS 則 無法建立受管 ENI，且無法執行檢查。如需詳細資訊，請參閱 Amazon VPC 配額。

原因代碼

describe-application-status 回應包含每次檢查的原因。原因包含您的應用程式傳回的 HTTP 狀態碼（以數字表示），以及用於檢查的通訊協定。`passed` 如果傳回的狀態碼包含在您的狀態碼比對程式中，則會標記檢查，`failed`否則會標記檢查。

原因還包括原因碼，以及 HTTP 層級結果的通訊協定和傳回的 HTTP 狀態碼。原因包含下列欄位：

`Code`

應用程式狀態檢查結果的原因碼。下列其中一值：

- `ResponseCodeMatched`：運作狀態檢查傳回的 HTTP 狀態碼符合設定的 `StatusCodeMatcher`。
- `ResponseCodeMismatch`：運作狀態檢查傳回的 HTTP 狀態碼不符合設定的 `StatusCodeMatcher`。
- `ConnectionTimeout`：與目標的連線逾時。
- `ResponseTimeout`：等待目標的回應時，運作狀態檢查逾時。
- `ConnectionRefused`：目標拒絕運作狀態檢查連線。

- ConnectionReset：在收到回應之前，已重設運作狀態檢查連線。

對於 ResponseCodeMatched和 ResponseCodeMismatch， StatusCode 欄位包含傳回的 HTTP 狀態碼，而 Protocol 欄位包含用於運作狀態檢查的通訊協定。對於連線錯誤，例如 ConnectionTimeout、ConnectionRefused、 ResponseTimeout和 ConnectionReset，StatusCode和 Protocol 欄位不存在。

Protocol

用於運作狀態檢查的通訊協定。HTTP 或 之一HTTPS。

StatusCode

運作狀態檢查傳回的 HTTP 狀態碼。

使用傳回的 HTTP 狀態碼來識別檢查失敗的原因。一些常見範例：

HTTP 狀態碼	典型意義	常見補救措施
200	應用程式傳回成功的回應。	無。這通常是正常狀態。
301, 302	應用程式傳回重新導向。運作狀態檢查呼叫不會遵循重新導向。	將運作狀態檢查路徑指向重新導向的目的地，或者如果您認為重新導向碼正常運作，請將重新導向碼新增至您的狀態碼比對程式。
401, 403	應用程式需要身分驗證或拒絕存取運作狀態檢查路徑。	將運作狀態檢查路徑設定為未經驗證，或在不需要登入資料的路徑上提供運作狀態檢查。
404	在應用程式上找不到設定的運作狀態檢查路徑。	確認路徑與您的應用程式提供的路由相符。
500	應用程式傳回內部伺服器錯誤。	調查執行個體上的應用程式日誌。
502, 503, 504	應用程式可連線，但會回報上游或容量問題。	調查應用程式運作狀態、相依性和容量。如果您的應用程式在啟動期間傳回這些代碼，請增加 InitializationGracePeriodSeconds 。

如需完整ApplicationStatusReason結構，請參閱《Amazon EC2 API 參考》中的 ApplicationStatusReason。

常見錯誤

- 安全群組不允許來自檢查連接埠上運作狀態檢查來源的傳入流量。
- 應用程式繫結至 網路界面127.0.0.1，而不是接聽。
- 運作狀態檢查路徑會傳回重新導向 (301、302) 而非成功回應，且狀態碼比對程式不包含重新導向碼。
- 檢查是針對 HTTPS 設定，但應用程式只提供 HTTP，反之亦然。
- 應用程式啟動的時間比InitializationGracePeriodSeconds值更長，Amazon EC2 Auto Scaling 會先取代執行個體，再準備就緒。

監控應用程式狀態檢查

您可以透過三種方式監控應用程式狀態檢查：

- Amazon CloudWatch。StatusCheckFailed_Application 指標會反映執行個體的整體應用程式狀態，並可驅動警示。指標會在彙總設定為 的相關聯檢查中，依執行個體彙總included。CloudWatch 也會針對每個名為 的相關聯檢查，發佈每個檢查的指標StatusCheckFailed_Application_*application-status-check-id*。
- describe-instance-status。傳回整體應用程式狀態，以及執行個體的其他狀態資訊。
- describe-application-status。傳回每個執行個體的詳細結果，包括每個相關聯的檢查的個別狀態，以及應用程式傳回的 HTTP 狀態碼。

使用 CloudWatch 指標進行警示驅動型自動化。當您已查詢執行個體狀態describe-instance-status時，請使用 。使用 describe-application-status以取得每個檢查的詳細可見性。

安全與許可

AWS 透過服務連結角色建立和管理用於應用程式狀態檢查的網路介面。服務不需要 IAM 設定即可建立這些 ENIs。服務連結角色使用 EC2ApplicationStatusChecksServiceRolePolicy AWS 受管政策。

若要自行建立、關聯、描述、刪除和隱藏應用程式狀態檢查，您的 IAM 使用者或角色需要對應的 Amazon EC2 許可。如需動作的完整清單，請參閱 Amazon EC2 API 參考。

執行個體的安全群組必須允許來自您設定連接埠上運作狀態檢查來源安全群組的傳入流量。使用 AWS 受管網路路徑時， AWS 會提供來源安全群組；使用客戶受管網路路徑時，請使用您指定為來源的安全群組。

定價

應用程式狀態檢查依下列元件計費：

- 每個可用區域每個受管彈性網路界面 (ENI) 的每小時 0.01 USD。
- 標準 Amazon CloudWatch 定價適用於應用程式狀態檢查指標。

配額

應用程式狀態檢查受 AWS 服務配額約束。如需配額名稱、預設值和說明，請參閱《 AWS 一般參考》中的 Amazon EC2 端點和配額。

除了影響受管網路介面 AWS 的服務配額之外，應用程式狀態檢查還具有下列服務配額。您可以從 Service Quotas 主控台檢視您的用量和請求增加。

在這些配額中，目標是一個運作狀態檢查監控的單一執行個體。如果多個運作狀態檢查監控執行個體，則每個執行個體和運作狀態檢查配對都會計為個別目標。關聯是您與運作狀態檢查建立關聯的單一標籤規則或單一執行個體 ID。每個規則或執行個體 ID 都會計為一個關聯，無論其解析的執行個體數量為何。

配額	預設	可調整
每個帳戶的運作狀態檢查	50	是，自動
每個運作狀態檢查的關聯	50	是，自動
每個帳戶的關聯數	200	是，自動
每個帳戶的目標數	5,000	是，依請求

大多數配額增加會自動核准。增加每個帳戶的目標需要請求和手動核准。

> **Important**
>
> 如果帳戶中的目標數量超過每個帳戶的目標配額，則不會監控超過限制的目標，也不會報告應用程式狀態。為了避免監控的差距，請將目標計數保持在配額內或請求增加。

我們建議您在應用程式狀態檢查配額用量上建立 Amazon CloudWatch 警示，以便在達到配額之前收到通知。Service Quotas 會將用量指標發佈至 CloudWatch 中的AWS/Usage命名空間，您可以使用它來建立警示。

Amazon EC2 執行個體的狀態變更事件

Amazon EC2 會在執行個體狀態變更時將 EC2 Instance State-change Notification 事件傳送至 Amazon EventBridge。

以下是此事件的範例資料。在本範例中，執行個體進入 pending 狀態。

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2021-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":"i-1234567890abcdef0",
      "state":"pending"
   }
}
```

state 的可能值為：

- pending
- running
- stopping
- stopped

- shutting-down
- terminated

啟動或開始執行個體時，其會變成 pending 狀態，然後再變成 running 狀態。停止執行個體時，其會變成 stopping 狀態，然後再變成 stopped 狀態。終止執行個體時，其會變成 shutting-down 狀態，然後再變成 terminated 狀態。如需詳細資訊，請參閱Amazon EC2 執行個體狀態變更。

建立在 Amazon EC2 執行個體變更狀態時傳送電子郵件的警示

若要在執行個體變更狀態時接收電子郵件通知，請建立 Amazon SNS 主題，然後為 EC2 Instance State-change Notification 事件建立 EventBridge 規則。

建立 SNS 主題

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。
2. 在導覽窗格中，選擇主題。
3. 請選擇建立主題。
4. 針對類型，選擇標準。
5. 在 Name (名稱) 中，輸入主題名稱。
6. 請選擇建立主題。
7. 選擇建立訂閱。
8. 對於通訊協定，選擇電子郵件。
9. 在 Endpoint (端點) 中，輸入接收通知的電子郵件地址。
10. 選擇建立訂閱。
11. 您會收到帶有下列主旨行的電子郵件訊息：AWS Notification - Subscription Confirmation。請依照指示來確認訂閱。

若要建立 EventBridge 規則

1. 在 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 選擇 Create rule (建立規則)。
3. 在 Name (名稱) 中，輸入規則名稱。
4. 針對規則類型，選擇具有事件模式的規則。
5. 選擇下一步。

6. 針對 Event pattern (事件模式)，請執行下列動作：
   a. 在 Event source (事件來源)，選擇 AWS 服務。
   b. 針對 AWS 服務，選擇 EC2。
   c. 在 Event Type (事件類型) 中，選擇 EC2 Instance State-change Notification (EC2 執行個體狀態變更通知)。
   d. 根據預設，我們會傳送任何執行個體的任何狀態變更通知。如果願意，可以選擇特定狀態或特定執行個體。
7. 選擇下一步。
8. 如下所示指定目標：
   a. 對於 Target types (目標類型)，選擇 AWS 服務。
   b. 對於 Select a target (選取目標)，選擇 SNS topic (SNS 主題)。
   c. 針對 Topic (主題)，選擇您在先前程序中建立的 SNS 主題。
9. 選擇下一步。
10. (選用) 將標籤新增至您的規則。
11. 選擇下一步。
12. 選擇建立規則。
13. 為了測試您的規則，請啟動狀態變更。例如，開始已停止的執行個體、停止執行中的執行個體或啟動執行個體。您會收到帶有下列主旨行的電子郵件訊息：AWS Notification Message。電子郵件內文包含事件資料。

Amazon EC2 執行個體的排程事件

為了確保基礎設施的可靠性和效能， AWS 可以排程事件以重新啟動、停止和淘汰您的執行個體。這些事件不會頻繁發生。

如果您的其中一個執行個體將受到排程事件的影響， 會使用與您 AWS 帳戶相關聯的電子郵件地址，透過電子郵件提前 AWS 通知您。此電子郵件提供事件的詳細資訊，例如開始和結束日期。根據事件類型，您可能可以採取動作來控制事件的時間。 AWS 也會傳送事件，您可以使用 Amazon EventBridge 來監控和管理該 AWS Health 事件。如需詳細資訊，請參閱使用 Amazon EventBridge 監控 AWS Health 中的事件。

排程事件由 管理 AWS。監控執行個體的排程事件。不過，您可以：

- 檢視執行個體的排程事件

- 自訂排程事件通知，以便在電子郵件通知中包含或移除標籤。
- 重新排程特定排程事件。
- 建立已排程事件的自訂事件時段。
- 在執行個體排定為重新啟動、停止或淘汰時採取動作。

為了確保您收到排程事件的通知，請在帳戶頁面上驗證您的聯絡資訊。

> **Note**
>
> 當執行個體受到排程事件的影響，並且屬於 Auto Scaling 群組時，Amazon EC2 Auto Scaling 最終會在運作狀態檢查期間將其取代，而您無需採取任何進一步動作。如需 Amazon EC2 Auto Scaling 執行運作狀態檢查的詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的 Auto Scaling 群組中執行個體的運作狀態檢查。

排程事件類型

Amazon EC2 可為執行個體建立以下類型的排程事件，其中事件會在排程的時間發生：

事件類型	事件代碼	事件動作
執行個體停止	`instance-stop`	執行個體會在排程時間停止。當您再次啟動時，它會遷移到新的主機。僅適用於具有 Amazon EBS 根磁碟區的執行個體。
執行個體淘汰	`instance-retirement`	在排程時間，如果執行個體具有 Amazon EBS 根磁碟區，則會停止，如果它具有執行個體儲存體根磁碟區，則會終止。
重新啟動執行個體	`instance-reboot`	執行個體會在排程時間重新啟動。執行個體會保留在主機上，在重新啟動期間，主機會進行維護。這稱為就地重新啟動。

事件類型	事件代碼	事件動作
系統重新開機	system-reboot	在排程時間，執行個體會重新啟動並移轉至新主機。這稱為重新啟動移轉。
系統維護	system-maintenance	在排程時間，執行個體可能會因網路維護或電源維護而暫時受到影響。

確定事件類型

您可以檢查為執行個體排程的事件類型。

Console

若要確定事件類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 在資料表中，事件代碼會出現在事件類型資料欄中。
4. 若要篩選資料表以僅顯示執行個體的事件，請在搜尋欄位中，從篩選條件清單中選擇資源類型：執行個體。

AWS CLI

若要確定執行個體的事件類型

使用 describe-instance-status 命令。如果執行個體有相關聯的排程事件，輸出會提供排程事件的相關資訊。

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0 \
    --query InstanceStatuses[].Events
```

以下為範例輸出。排程事件代碼為 system-reboot。

```
[
```

```
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2020-03-14T22:00:00.000Z",
            "NotBefore": "2020-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2020-04-05T11:00:00.000Z"
        }
    ]
]
```

PowerShell

若要確定執行個體的事件類型

使用 Get-EC2InstanceStatus cmdlet。如果執行個體有相關聯的排程事件，輸出會提供排程事件的相關資訊。

```
(Get-EC2InstanceStatus `
    -InstanceId i-1234567890abcdef0).Events
```

以下為範例輸出。排程事件代碼為 system-reboot。

```
Code              : system-reboot
Description       : The instance is scheduled for a reboot
InstanceEventId   : instance-event-0d59937288b749b32
NotAfter          : 2020-03-14T22:00:00.000Z
NotBefore         : 2020-03-14T20:00:00.000Z
NotBeforeDeadline : 2020-04-05T11:00:00.000Z
```

目錄

- 管理已排程為停止或淘汰的 Amazon EC2 執行個體
- 管理已排程進行重新啟動的 Amazon EC2 執行個體
- 管理為維護排程的 Amazon EC2 執行個體
- 檢視會影響 Amazon EC2 執行個體的排程事件
- 自訂 EC2 執行個體的排程事件通知
- 重新排程 EC2 執行個體的排程事件

- 為影響 Amazon EC2 執行個體的排程事件建立自訂事件時段

管理已排程為停止或淘汰的 Amazon EC2 執行個體

當 AWS 偵測到執行個體的基礎主機發生無法修復的故障時，會根據執行個體的根磁碟區類型，將執行個體排程為停止或終止。

- 若執行個體具有 Amazon EBS 根磁碟區，則執行個體會排程為停止。
- 若執行個體具有執行個體儲存體根磁碟區，則執行個體會排程為終止。

如需詳細資訊，請參閱執行個體淘汰。

> **Important**
>
> 任何存放在執行個體存放區磁碟區的資料都會在執行個體停止、休眠或終止時遺失。這包含連接到擁有 EBS 根磁碟區之執行個體的執行個體儲存體磁碟區。請確認您已在執行個體停止、休眠或終止前，從您的執行個體存放區磁碟區儲存您稍後需要的資料。

您可以採取的動作

對於具有 EBS 根磁碟區的執行個體，您可以採取的動作

當您收到已排程的 `instance-stop` 事件通知時，可以採取下列其中一個動作：

- 等待排程的停止：您可以等待執行個體在其排程的維護時段內停止。
- 執行手動停止和啟動：您可以在適合您的時間自行停止和啟動執行個體，這會將其移轉至新主機。這與重新啟動執行個體不同。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。
- 自動停止和啟動：您可以將立即停止和啟動自動化，以回應排程的 `instance-stop` 事件。如需詳細資訊，請參閱「AWS Health 使用者指南」中的在 EC2 執行個體上自動執行操作以回應 AWS Health中的事件。

對於具有執行個體儲存體根磁碟區的執行個體，您可以採取的動作

當您收到已排程的 `system-retirement` 事件通知，並且您想要保留資料時，可以採取下列動作：

1. 從最近的 AMI 中啟動替代執行個體。
2. 在執行個體被排程為終止之前，將所有必要的資料移轉至替代執行個體。

3. 終止原始執行個體，或是等待執行個體依照排程終止。

如需有關您可採取的動作詳細資訊，請參閱執行個體淘汰。

管理已排程進行重新啟動的 Amazon EC2 執行個體

當 AWS 必須執行安裝更新或維護基礎主機等任務時，可以排程執行個體重新啟動。在排程的重新啟動期間，執行個體會保留在相同的主機上，或移轉至不同的主機，具體取決於事件，如下所示：

- `instance-reboot` 事件
  - 在重新啟動期間，執行個體會保留在主機上。這稱為就地重新啟動。
  - 目前的主機會進行維護。
  - 通常在幾秒鐘內完成。
- `system-reboot` 事件
  - 在重新啟動期間，執行個體會移轉至新主機。這稱為重新啟動移轉。
  - 通常在幾分鐘內完成。

若要檢查為您的執行個體排程的事件類型，請參閱確定事件類型。

您可以採取的動作

當您收到排程的 `instance-reboot` 或 `system-reboot` 事件通知時，可以採取下列其中一個動作：

- 等待排程的重新啟動：您可以等待執行個體在其排程的維護時段內重新啟動。
- 對重新啟動進行重新排程：您可以將執行個體重新啟動重新排程為適合您的日期和時間。
- 執行使用者啟動的重新啟動：您可以在適合您的時間自行手動重新啟動執行個體。不過，結果取決於事件：
  - `instance-reboot` 事件 – 您的執行個體會保留在目前的硬體上 (就地重新啟動)，不會進行主機維護，而且事件會保持開啟狀態。
  - `system-reboot` 事件
    - 如果執行個體已啟用重新啟動移轉，則使用者啟動的重新啟動會嘗試將執行個體移轉至新硬體。如果成功，則會清除事件。如果失敗，將發生就地重新啟動，並且事件會按排程進行。
    - 如果執行個體已停用重新啟動移轉，則使用者啟動的重新啟動會將執行個體保留在相同的硬體上 (就地重新啟動)，不會執行主機維護，而且事件會保持排程狀態。當排程事件最終發生時，AWS 會將您的執行個體移至新硬體 （ 重新啟動遷移 ）。

AWS 重新啟動執行個體後

AWS 重新啟動執行個體後，以下適用：

- 清除排程的事件。
- 更新事件描述。
- 對於 `instance-reboot` 事件：
  - 基礎主機的維護已完成。
- 對於 `system-reboot` 事件：
  - 執行個體會移至新主機。
  - 執行個體會保留其 IP 位址和 DNS 名稱。
  - 不會保留本機執行個體儲存體磁碟區上的資料。
- 您可以在執行個體完全啟動後使用它。

替代選項

如果您無法重新排程重新啟動事件或為使用者啟動的重新啟動啟用重新啟動移轉，但想要在排程的維護時段維持正常操作，可以執行下列動作：

- 對於具有 EBS 根磁碟區的執行個體
  - 手動停止並啟動執行個體，將其移轉至新主機。這與手動重新啟動執行個體不同，其中執行個體會保留在相同的主機上。
  - 也可以選擇將立即執行個體停止和啟動自動化，以回應已排程的重新啟動事件。如需詳細資訊，請參閱「AWS Health 使用者指南」中的在 EC2 執行個體上自動執行操作以回應 AWS Health中的事件。

> **Important**
>
> 在執行個體停止時，執行個體儲存體磁碟區上的資料會遺失。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。

- 對於具有執行個體儲存體根磁碟區的執行個體
  1. 從最近的 AMI 中啟動替代執行個體。
  2. 在排程的維護時段之前，將所有必要的資料移轉至取代執行個體。
  3. 終止原始執行個體。

啟用或停用重新啟動移轉

為 system-reboot 事件排程執行個體時，可以在事件之前將其重新啟動。使用者啟動的重新啟動結果取決於執行個體的重新啟動移轉設定：

- 已啟用 – 使用者啟動的重新啟動會嘗試將執行個體移轉至新硬體 (重新啟動移轉)。如果成功，則會清除事件。如果失敗，將發生就地重新啟動，並且事件會按排程進行。請注意，即使已啟用，只有在執行個體符合重新啟動移轉需求時，才能進行重新啟動移轉。
- 已停用 - 使用者啟動的重新啟動會將執行個體保留在相同的硬體上 (就地重新啟動)，不會執行主機維護，而且事件會保持排程狀態。當排程事件最終發生時， AWS 會將您的執行個體移至新硬體 （ 重新啟動遷移 ）。

移轉後重新啟動需要的時間比就地重新啟動更長：

- 就地重新啟動：約 30 秒
- 移轉後重新啟動：數分鐘

> **Note**
>
> 預設會為使用者啟動的重新啟動移轉啟用可接收 system-reboot 事件通知的執行個體。

啟用重新啟動移轉的需求

可以在滿足下列條件的執行個體上啟用重新啟動移轉：

執行個體類型

並非所有執行個體類型都支援啟用重新啟動移轉。您可以檢視支援啟用重新啟動移轉的執行個體類型。

Console

若要檢視支援啟用重新啟動移轉的執行個體類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 Instance Types (執行個體類型)。
3. 在篩選列中，輸入重新啟動移轉支援：支援。當您輸入字元且篩選條件名稱出現時，可以選取它。

執行個體類型資料表會顯示支援啟用重新啟動移轉的所有執行個體類型。

AWS CLI

若要檢視支援啟用重新啟動移轉的執行個體類型

使用 describe-instance-types 命令並搭配 `reboot-migration-support` 篩選條件。

```
aws ec2 describe-instance-types \
    --filters Name=reboot-migration-support,Values=supported \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

若要檢視支援啟用重新啟動移轉的執行個體類型

使用 Get-EC2InstanceType cmdlet 搭配 `reboot-migration-support` 篩選條件。

```
Get-EC2InstanceType `
    -Filter @{Name="reboot-migration-support";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

租用

- 共同
- Dedicated Instance

如需詳細資訊，請參閱Amazon EC2 專用執行個體。

限制

具有下列特性的執行個體不支援重新啟動移轉：

- 平台：在 Xen Hypervisor 上原生執行的執行個體
- 執行個體大小：`metal` 執行個體
- 租用：專用主機。對於專用主機，改用專用主機自動復原。
- 儲存：具有執行個體儲存體磁碟區的執行個體

- 聯網：使用 Elastic Fabric Adapter 的執行個體
- 自動擴展：屬於 Auto Scaling 群組的執行個體

啟用或停用重新啟動移轉的步驟

當執行個體收到 `system-reboot` 事件時，預設會啟用重新啟動移轉。可以停用重新啟動移轉，以便在使用者啟動的重新啟動期間，執行個體會保持在相同的硬體上 (就地重新啟動)。

`default` 組態不會為不支援的執行個體啟用重新啟動移轉。如需詳細資訊，請參閱啟用重新啟動移轉的需求。

您可以在執行中或已停止的執行個體上停用或啟用重新啟動移轉。

AWS CLI

停用重新啟動移轉

使用 modify-instance-maintenance-options 命令，並將 `--reboot-migration` 參數設定為 `disabled`。

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration disabled
```

若要啟用重新啟動移轉

使用 modify-instance-maintenance-options 命令，並將 `--reboot-migration` 參數設定為 `default`。

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration default
```

PowerShell

停用重新啟動移轉

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId  `
    -RebootMigration Disabled
```

若要啟用重新啟動移轉

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -RebootMigration Enabled
```

管理為維護排程的 Amazon EC2 執行個體

當 AWS 必須維護執行個體的基礎主機時，它會排程執行個體進行維護。有兩種維護事件類型：網路維護及電源維護。

- 在網路維護期間，排程執行個體會暫時遺失網路連線能力。維護完成後，便會還原您執行個體的一般網路連線能力。
- 在電源維護期間，排程執行個體會暫時離線，然後重新開機。當執行重新開機時，會保留所有您執行個體的組態設定。

在您的執行個體重新開機後 (通常需要數分鐘)，確認您的應用程式正以預期的方式正常運作。此時，您的執行個體應該不會再有任何相關聯的排程事件，或若是如此，則排程事件描述的開頭為 [Completed] ([已完成])。有時需要最多 1 個小時，才會更新執行個體狀態描述。完成的維護事件會在 Amazon EC2 主控台儀表板上顯示最多一週。

您可以採取的動作

對於具有 EBS 根磁碟區的執行個體，您可以採取的動作

當您收到 `system-maintenance` 事件通知時，可以採取下列其中一個動作：

- 等待已排程的維護：可以等待維護按排程進行。
- 執行手動停止和啟動：可以停止和啟動執行個體，將其移轉到新主機。這與重新啟動執行個體不同。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。

- 自動化停止和啟動：可以將立即停止和啟動進行自動化，以回應排程的維護事件。如需詳細資訊，請參閱「AWS Health 使用者指南」中的在 EC2 執行個體上自動執行操作以回應 AWS Health中的事件。

對於具有執行個體儲存體根磁碟區的執行個體，您可以採取的動作

當您收到 `system-maintenance` 事件通知時，可以採取下列其中一個動作：

- 等待已排程的維護：可以等待維護按排程進行。
- 啟動替代執行個體：如果您想要在排程的維護時段維持正常操作：
  1. 從最近的 AMI 中啟動替代執行個體。
  2. 在排程的維護時段之前，將所有必要的資料移轉至取代執行個體。
  3. 終止原始執行個體。

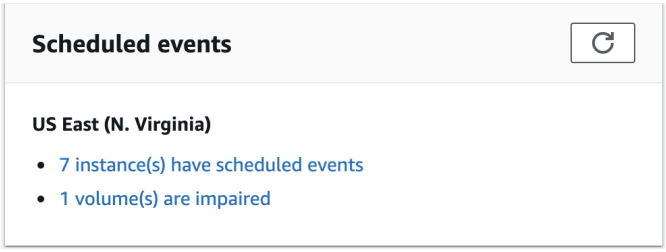
檢視會影響 Amazon EC2 執行個體的排程事件

除了在電子郵件中接收排程事件的通知，您可以檢查排程事件。

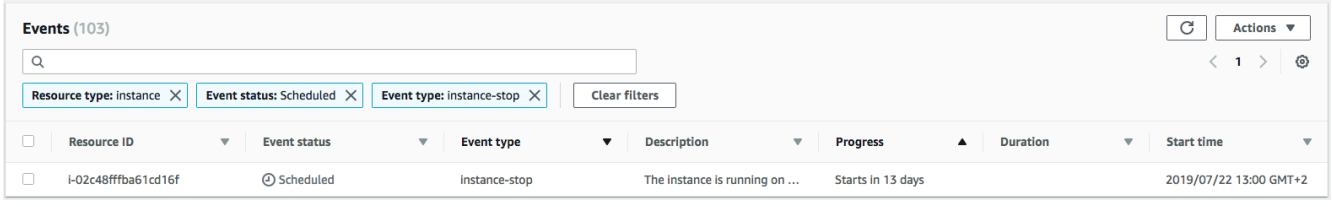
Console

檢視執行個體的排程事件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 儀表板會在排程事件下方顯示任何與事件相關聯的資源。



3. 如需更多詳細資訊，在導覽窗格中選擇事件。此時會顯示任何與事件相關聯的資源。您可以依特性 (例如事件類型、資源類型和可用區域) 進行篩選。



AWS CLI

檢視執行個體的排程事件

使用 describe-instance-status 命令。

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0 \
    --query "InstanceStatuses[].Events"
```

下列範例輸出顯示重新開機事件。

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2019-03-15T22:00:00.000Z",
            "NotBefore": "2019-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
         }

    ]
]
```

下列為顯示執行個體淘汰事件的範例輸出。

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0e439355b779n26",

            "Code": "instance-stop",
            "Description": "The instance is running on degraded hardware",
            "NotBefore": "2015-05-23T00:00:00.000Z"
        }
    ]
]
```

PowerShell

檢視執行個體的排程事件

使用下列 Get-EC2InstanceStatus 命令。

```
(Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0).Events
```

下列為顯示執行個體淘汰事件的範例輸出。

```
Code         : instance-stop
Description  : The instance is running on degraded hardware
NotBefore    : 5/23/2015 12:00:00 AM
```

Instance metadata

使用執行個體中繼資料檢視您執行個體的排程事件

您可以使用執行個體中繼資料服務第 2 版或執行個體中繼資料服務第 1 版，從執行個體中繼資料擷取有關執行個體之作用中維護事件的資訊。

IMDSV2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

以下範例輸出包含排程系統重新開機事件的資訊 (採用 JSON 格式)。

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "active"
  }
```

```
]
```

使用執行個體中繼資料檢視執行個體已完成或已取消事件的事件歷史記錄

您可以使用執行個體中繼資料服務第 2 版或執行個體中繼資料服務第 1 版，從執行個體中繼資料擷取有關已完成或已取消之執行個體事件的資訊。

IMDSV2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/history
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/history
```

以下範例輸出包含已取消之系統重新開機事件和已完成之系統重新開機事件的資訊 (採用 JSON 格式)。

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Canceled] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "canceled"
  },
  {
    "NotBefore" : "29 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Completed] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "29 Jan 2019 09:17:23 GMT",
    "State" : "completed"
  }
]
```

AWS Health

您可以使用 AWS Health 儀板表 來了解可能會影響執行個體的事件。會將問題 Health 儀板表 整理成三個群組：開啟問題、排程變更和其他通知。排定的變更包含進行中或即將進行的項目。

如需詳細資訊，請參閱「指南」中的「AWS Health AWS Health 儀板表入門」。

自訂 EC2 執行個體的排程事件通知

您可以自訂排程事件通知，以便在電子郵件通知中包含標籤。如此可以更輕鬆地識別受影響的資源 (執行個體或 專用執行個體)，並為即將到來的事件排定動作的優先順序。

當您自訂事件通知以包含標籤時，您可以選擇包含：

- 與受影響資源關聯的所有標籤
- 僅與受影響資源關聯的特定標籤

例如，假設您將 `application`、`costcenter`、`project`，和 `owner` 標籤指派給所有執行個體。您可以選擇在事件通知中包含所有標籤。或者，如果您只想在事件通知中看到 `owner` 和 `project` 標籤，則可以選擇只包含這些標籤。

選取要包含的標籤之後，事件通知將包含資源 ID (執行個體 ID 或 專用執行個體 ID)，以及與受影響資源關聯的標籤金鑰和值組。

任務

- 在事件通知中包含標籤
- 從事件通知中移除標籤
- 檢視要包含在事件通知中的標籤

在事件通知中包含標籤

您選擇要包含的標籤會套用至所選區域的所有資源 (執行個體和 專用執行個體)。若要自訂其他區域中的事件通知，請先選取所需的區域，然後執行下列步驟。

Console

在事件通知中包含標籤

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event notifications (管理事件通知)。
4. 開啟在事件通知中包含標籤。
5. 根據您要包含在事件通知中的標籤，執行下列其中一項動作：
   - 若要包含與受影響執行個體或專用執行個體相關聯的所有標籤，請選取包含所有資源標籤。
   - 如需選取要包含的標籤，請選取選擇要包含的標籤，然後選取或輸入標籤索引鍵。
6. 選擇儲存。

AWS CLI

在事件通知中包含所有標籤

使用 register-instance-event-notification-attributes 命令並將 `IncludeAllTagsOfInstance` 參數設定為 `true`。

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=true"
```

在事件通知中包含特定標籤

使用 register-instance-event-notification-attributes 命令，並使用 `InstanceTagKeys` 參數指定要包含的標籤。

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute 'InstanceTagKeys=["tag_key_1", "tag_key_2",
 "tag_key_3"]'
```

PowerShell

在事件通知中包含所有標籤

使用 Register-EC2InstanceEventNotificationAttribute cmdlet。

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $true
```

在事件通知中包含特定標籤

使用 Register-EC2InstanceEventNotificationAttribute cmdlet。

```
Register-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey tag_key_1, tag_key_2, tag_key_3
```

從事件通知中移除標籤

可從事件通知中移除標籤。

Console

從事件通知中移除標籤

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event notifications (管理事件通知)。
4. 若要從事件通知中移除所有標籤，請關閉在事件通知中包含資源標籤。
5. 若要從事件通知中移除特定標籤，請為對應的標籤索引鍵選擇 X)。
6. 選擇儲存。

AWS CLI

從事件通知中移除所有標籤

使用 deregister-instance-event-notification-attributes 命令並將 `IncludeAllTagsOfInstance` 參數設定為 `false`。

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute "IncludeAllTagsOfInstance=false"
```

從事件通知中移除標籤

使用 deregister-instance-event-notification-attributes 命令，並使用 `InstanceTagKeys` 參數指定要移除的標籤。

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute 'InstanceTagKeys=["tag_key_3"]'
```

PowerShell

從事件通知中移除所有標籤

使用 Unregister-EC2InstanceEventNotificationAttribute cmdlet。

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_IncludeAllTagsOfInstance $false
```

從事件通知中移除標籤

使用 Unregister-EC2InstanceEventNotificationAttribute cmdlet。

```
Unregister-EC2InstanceEventNotificationAttribute `
    -InstanceTagAttribute_InstanceTagKey tag_key_3
```

檢視要包含在事件通知中的標籤

可檢視要包含在事件通知中的標籤。

Console

檢視要包含在事件通知中的標籤

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event notifications (管理事件通知)。

AWS CLI

若要檢視要包含在事件通知中的標籤

使用 describe-instance-event-notification-attributes 命令。

```
aws ec2 describe-instance-event-notification-attributes
```

PowerShell

若要檢視要包含在事件通知中的標籤

使用 Get-EC2InstanceEventNotificationAttribute cmdlet。

```
Get-EC2InstanceEventNotificationAttribute
```

重新排程 EC2 執行個體的排程事件

您可以重新排程事件，使其發生在適合您的特定日期和時間。對事件進行重新排程後，可能需要一到兩分鐘才會顯示更新的日期。

限制

- 只能重新排程有事件截止日期的事件。事件最晚可以重新排程到事件截止日期。截止日期資料欄 (主控台) 和 `NotBeforeDeadline` 欄位 (AWS CLI) 會顯示事件截止日期。
- 只能重新排程尚未開始的事件。開始時間資料欄 (主控台) 和 `NotBefore` 欄位 (AWS CLI) 會顯示開始時間。已排程在接下來 5 分鐘內開始的事件不能重新排程。
- 新的事件開始時間與目前時間至少必須相隔 60 分鐘。
- 如果您使用主控台重新排程多個事件，則由具有最早事件截止日期的事件決定事件截止日期。

Console

重新排程事件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 從篩選清單中選擇 Resource type: instance (資源類型：執行個體)。
4. 選取一或多個執行個體，然後選擇 Actions (動作)、Schedule event (排程事件)。

   只能重新排程有事件截止日期 (由 Deadline (截止日期) 的值指出) 的事件。如果其中一個選取的事件沒有截止日期，則會停用 Actions (動作)、Schedule event (排程事件)。
5. 對於 New start time (新增開始時間)，輸入事件的新日期和時間。新的日期和時間必須發生在 Event deadline (事件截止日期) 之前。
6. 選擇儲存。

   更新後的事件開始時間可能需要一兩分鐘才會反映在主控台中。

AWS CLI

重新排程事件

使用 modify-instance-event-start-time 命令。

```
aws ec2 modify-instance-event-start-time \
    --instance-id i-1234567890abcdef0 \
    --instance-event-id instance-event-0d59937288b749b32 \
    --not-before 2020-03-25T10:00:00.000
```

PowerShell

重新排程事件

使用 Edit-EC2InstanceEventStartTime cmdlet。

```
Edit-EC2InstanceEventStartTime `
    -InstanceId i-1234567890abcdef0 `
    -InstanceEventId instance-event-0d59937288b749b32 `
    -NotBefore 2020-03-25T10:00:00.000
```

為影響 Amazon EC2 執行個體的排程事件建立自訂事件時段

您可以為重新啟動、停止或終止 Amazon EC2 執行個體的排程事件定義自訂每週重複的事件時段。您可以將一或多個執行個體關聯到事件時段。如果已計劃這些執行個體的排程事件， AWS 會在相關聯的事件時段中排程事件。

您可以使用事件時段，透過指定在工作負載離峰期間出現的事件時段，將工作負載可用性最大化。您也可以調整事件時段與內部維護排程。

可以透過指定一組時間範圍來定義事件時段。最短時間範圍為 2 小時。合併的時間範圍總計必須至少 4 小時。

您可以使用執行個體 ID 或執行個體標籤，將一或多個執行個體與事件時段建立關聯。您也可以使用主機 ID，將專用執行個體與事件時段建立關聯。

> **Warning**
>
> 事件時段僅適用於停止、重新啟動或終止執行個體的排程事件。
> 事件時段不適用於：
>
> - 快速排程事件和網路維護事件。

- 未排程的維護 (例如自動執行個體復原) 和未計劃的重新啟動。

使用事件時段

- 考量事項
- 建立事件時段
- 將目標與事件時段關聯
- 解除目標與事件時段的關聯
- 修改事件時段
- 刪除事件時段

考量事項

- 所有事件時段的時間皆為 UTC 格式。
- 事件時段可以包含多個時間範圍。雖然每個單獨範圍必須至少為 2 小時，但所有範圍的總持續時間必須至少為 4 小時。
- 只有一個目標類型 (執行個體 ID、專用執行個體 ID 或執行個體標籤) 可以與事件時段關聯。
- 一個目標類型 (執行個體 ID、專用執行個體 ID 或執行個體標籤) 只能與一個事件時段關聯。
- 事件時段最多可與 100 個執行個體 ID 或 50 個專用執行個體 ID 或 50 個執行個體標籤相關聯。執行個體標籤可與任意數量的執行個體關聯。
- 每個 AWS 區域最多可建立 200 個事件時段。
- 與事件時段相關聯的多個執行個體可能會同時發生排程事件。
- 如果 AWS 已排程事件，修改事件時段不會變更排程事件的時間。如果事件有截止日期，您可以重新排程事件。
- 可以在排程事件之前停止和啟動執行個體。這會將執行個體移轉至新主機並清除事件。

建立事件時段

您可以建立一或多個事件時段。對於每個事件時段，您可以指定一個或多個時間段。例如，您可以建立一個事件時段，其時間段為每天凌晨 4 點開始，持續 2 小時。或者，您可以建立一個事件時段，其時間段為星期日凌晨 2 點到凌晨 4 點和星期三凌晨 3 點到凌晨 5 點。

事件時段會每週重複出現，直到您刪除為止。

Console

建立事件時段

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event windows (管理事件時段)。
4. 選擇 Create instance event window (建立執行個體事件時段)。
5. 對於 Event window name (事件時段名稱)，輸入事件時段的描述性名稱。
6. 對於 Event window schedule (事件時段排程)，透過使用 cron 排程建置器或指定時間範圍，進行選擇以在事件時段中指定時間段。

   - 如果選擇 Cron schedule builder (Cron 排程建置器)，請指定以下內容：
     1. 對於 Days (UTC) (天 (UTC))，指定事件時段發生的星期日。
     2. 對於 Start time (UTC) (開始時間 (UTC))，指定事件時段開始的時間。
     3. 對於 Duration (持續時間)，在事件時段中指定時間段的持續時間。每個時間段的最短持續時間為 2 小時。事件時段的最短持續時間總計必須等於或超過 4 小時。所有時間均以 UTC 表示。
   - 如果選擇 Time ranges (時間範圍)，請選擇 Add new time range (新增時間範圍)，然後指定開始日期和時間以及結束日期和時間。對每個時間範圍重複此步驟。每個時間範圍的最短持續時間為 2 小時。所有時間範圍合併的最小持續時間總計必須等於或超過 4 小時。
7. (選用) 如需目標詳細資訊，請將一個或多個執行個體與事件時段建立關聯。使用執行個體 ID 或執行個體標籤來關聯執行個體。使用主機 ID 來關聯專用主機。當對這些目標進行排程以進行維護時，事件將在此事件時段期間出現。

   請注意，您可以建立事件時段，而無需將目標與時段建立關聯。稍後，您可以修改時段以關聯一個或多個目標。
8. (選用) 對於 Event window tags (事件時段標籤)，選擇 Add tag (新增標籤)，然後輸入標籤的金鑰和值。針對每個標籤，重複執行上述動作。
9. 選擇 Create event window (建立事件時段)。

AWS CLI

建立具有時間範圍的事件時段

使用 create-instance-event-window 命令。

```
aws ec2 create-instance-event-window \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

透過 cron 表達式建立事件時段

使用 create-instance-event-window 命令。

```
aws ec2 create-instance-event-window \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

建立具有時間範圍的事件時段

使用 New-EC2InstanceEventWindow cmdlet。

```
$timeRange = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange.StartWeekDay = "monday"
$timeRange.EndWeekDay = "wednesday"
$timeRange.StartHour = 2
$timeRange.EndHour = 8
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec `
    -TimeRange @($timeRange)
```

以下為範例輸出。

```
AssociationTarget     :
CronExpression        :
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
```

```
Tags                 : {key1}
TimeRanges           : {Amazon.EC2.Model.InstanceEventWindowTimeRange}
```

透過 cron 表達式建立事件時段

使用 New-EC2InstanceEventWindow cmdlet。

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec`
    -CronExpression "* 21-23 * * 2,3"
```

以下為範例輸出。

```
AssociationTarget     :
CronExpression        : * 21-23 * * 2,3
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {}
```

將目標與事件時段關聯

建立事件時段後，可以將目標與事件時段建立關聯。只能將一種類型的目標與事件時段建立關聯。可以指定執行個體 ID、專用主機 ID 或執行個體標籤。

Console

若要將目標與事件時段關聯

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選取要修改的事件時段。
4. 選擇動作、修改執行個體事件時段。

AWS CLI

將執行個體標籤與事件時段關聯

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
                    "Value": "v1"
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

若要將執行個體 ID 與事件時段建立關聯

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

將專用執行個體與事件時段關聯

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
```

```
}
```

PowerShell

將執行個體標籤與事件時段關聯

使用 Register-EC2InstanceEventWindow cmdlet。

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1,$tag2)
```

若要將執行個體 ID 與事件時段建立關聯

使用 Register-EC2InstanceEventWindow cmdlet。

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

將專用執行個體與事件時段關聯

使用 Register-EC2InstanceEventWindow cmdlet。

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

解除目標與事件時段的關聯

Console

若要解除目標與事件時段的關聯

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選取要修改的事件時段。
4. 選擇動作、修改執行個體事件時段。

AWS CLI

解除執行個體標籤與事件時段的關聯

使用 disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

若要解除執行個體 ID 與事件時段的關聯

使用 disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
```

```
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

解除專用執行個體與事件時段的關聯

使用 disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

PowerShell

解除執行個體標籤與事件時段的關聯

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
```

```
    -AssociationTarget_InstanceTag @($tag1, $tag2)
```

若要解除執行個體 ID 與事件時段的關聯

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

解除專用執行個體與事件時段的關聯

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

修改事件時段

您可以修改事件時段的所有欄位，但其 ID 除外。例如，當日光節約時間開始時，您可能想要修改事件時段排程。對於現有事件時段，您可能會想要新增或移除目標。

您可以在修改事件時段時修改時間範圍或 cron 表達式，但不能同時修改兩者。

Console

修改事件時段

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event windows (管理事件時段)。
4. 選取要修改的事件時段，然後選擇 Actions (動作)、Modify instance event window (修改執行個體事件時段)。
5. 修改事件時段中的欄位，然後選擇 Modify event window (修改事件時段)。

AWS CLI

修改事件時段的時間範圍

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

修改事件時段的一組時間範圍

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
```

```
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
   {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            },
            {
                "StartWeekDay": "thursday",
                "StartHour": 2,
                "EndWeekDay": "friday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

修改事件時段的 cron 表達式

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

以下為範例輸出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

修改事件時段的時間範圍

使用 Edit-EC2InstanceEventWindow cmdlet。

```
$timeRange1 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange1.StartWeekDay = "monday"
$timeRange1.EndWeekDay = "wednesday"
```

```
$timeRange1.StartHour = 2
$timeRange1.EndHour = 8
$timeRange2 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange2.StartWeekDay = "thursday"
$timeRange2.EndWeekDay = "friday"
$timeRange2.StartHour = 1
$timeRange2.EndHour = 6
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -TimeRange @($timeRange1, $timeRange2)
```

修改事件時段的 cron 表達式

使用 Edit-EC2InstanceEventWindow cmdlet。

```
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -CronExpression "* 21-23 * * 2,3"
```

刪除事件時段

可以一次刪除一個事件時段。

Console

刪除事件時段

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Events (事件)。
3. 選擇 Actions (動作)、Manage event windows (管理事件時段)。
4. 選取要刪除的事件時段，然後選擇 Actions (動作)、Delete instance event window (刪除執行個體事件時段)。
5. 當出現提示時，輸入 **delete**，然後選擇 Delete (刪除)。

AWS CLI

刪除事件時段

使用 delete-instance-event-window 命令，然後指定要刪除的事件時段。

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890
```

若要強制刪除事件時段

使用 `--force-delete` 參數 (如果事件時段目前與目標相關聯)。

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

PowerShell

刪除事件時段

使用 Remove-EC2InstanceEventWindow cmdlet。

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890
```

若要強制刪除事件時段

使用 Remove-EC2InstanceEventWindow cmdlet。

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -ForceDelete $true
```

使用 CloudWatch 監控您的執行個體

您可以使用 Amazon CloudWatch 來監控執行個體，前者會收集來自 Amazon EC2 的原始資料，並處理為可讀且近乎即時的指標。這些統計資料會記錄 15 個月的時間，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。

在預設情況下，Amazon EC2 會每隔 5 分鐘將指標資料傳送到 CloudWatch。若要將執行個體的指標資料每隔 1 分鐘傳送至 CloudWatch，您可啟用執行個體的詳細監控。如需詳細資訊，請參閱 管理 EC2 執行個體的詳細監控。

Amazon EC2 主控台會根據 Amazon CloudWatch 的原始資料顯示一系列圖形。根據需求，您可能偏好從 Amazon CloudWatch 取得執行個體的資料，而非從主控台的圖形取得資料。

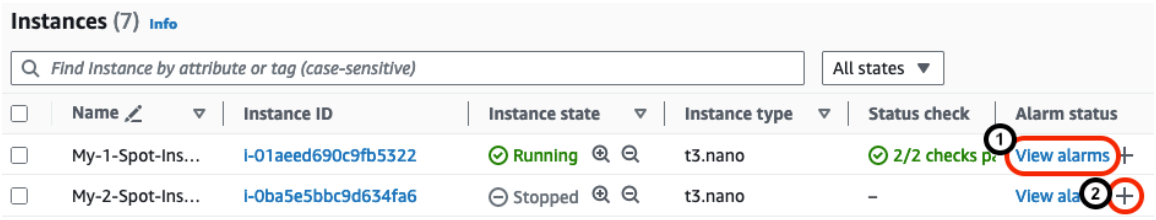
如需 Amazon CloudWatch 帳單與成本資訊，請參閱 Amazon CloudWatch 使用者指南中的 CloudWatch 帳單與成本。

目錄

- 在 Amazon EC2 主控台中管理 EC2 執行個體的 CloudWatch 警示
- 管理 EC2 執行個體的詳細監控
- 您的執行個體可用的 CloudWatch 指標
- 使用 Amazon EC2 主控台安裝和設定 CloudWatch 代理程式，以新增其他指標
- 執行個體 CloudWatch 指標的統計資料。
- 檢視執行個體的監控圖表
- 為執行個體建立 CloudWatch 警示
- 建立警示以停止、終止、重新啟動或復原執行個體

在 Amazon EC2 主控台中管理 EC2 執行個體的 CloudWatch 警示

從 Amazon EC2 主控台的執行個體畫面中，您可以管理執行個體的 Amazon CloudWatch 警示。在執行個體資料表中，警示狀態欄提供兩個主控台控制項：一個用於檢視警示的控制項，另一個用於建立或編輯警示。下列螢幕擷取畫面指出這些主控台控制項，編號為 1 (檢視警示) 和 2 (建立或編輯警示的 + 符號)。



從執行個體畫面中檢視警示

您可以從執行個體畫面中檢視每個執行個體的警示。

從執行個體畫面中檢視執行個體的警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 在執行個體資料表中，針對您選擇的執行個體選擇檢視警示 (在上述螢幕擷取畫面中為編號 1)。
4. 在 ***i-1234567890abcdef0*** 的警示詳細資訊視窗中，選擇警示名稱以在 CloudWatch 主控台中檢視警示。

從執行個體畫面建立警示

您可以從執行個體畫面為每個執行個體建立警示。

從執行個體畫面建立執行個體的警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 在執行個體資料表中，針對您選擇的執行個體，選擇加號 (在上述螢幕擷取畫面中為編號 2)。
4. 在管理 CloudWatch 警示畫面中，建立您的警示。如需詳細資訊，請參閱 為執行個體建立 CloudWatch 警示。

從執行個體畫面編輯警示

您可以從執行個體畫面編輯執行個體的警示。

從執行個體畫面編輯執行個體的警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 在執行個體資料表中，針對您選擇的執行個體，選擇加號 (在上述螢幕擷取畫面中為編號 2)。
4. 在管理 CloudWatch 警示畫面中，編輯您的警示。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的編輯或刪除 CloudWatch 警示。

管理 EC2 執行個體的詳細監控

Amazon CloudWatch 提供兩種類別的監控：基本監控和詳細監控。依預設，設定執行個體以進行基本監控。您可以選擇啟用詳細監控，以協助您更快地識別和處理操作問題。執行個體啟動時，或在執行個體運作或停止後，您都可以啟用或關閉執行個體的詳細監控。

在執行個體上啟用詳細監控並不會影響其連接 EBS 磁碟區的監控。如需詳細資訊，請參閱 Amazon EBS 的 Amazon CloudWatch 指標。

下表重點說明您執行個體的基本監控和詳細監控之間的差異。

監控類型	描述	費用
基本監控	狀態檢查指標在 1 分鐘內可用。所有其他指標均在 5 分鐘內可用。	免費。
詳細監控	如果啟用了執行個體的詳細監控，即可在 1 分鐘內取得指標。 啟用詳細監控之後，您可以在類似的執行個體群組之間彙總資料。	我們會按 Amazon EC2 傳送至 CloudWatch 的指標計費。您不需為資料儲存付費。如需詳細資訊，請參閱 Amazon CloudWatch 定價頁面中的付費方案。

目錄

- 必要許可
- 在啟動時啟用詳細監控
- 管理詳細監控

必要許可

若要啟用執行個體的詳細監控，您的使用者必須具有使用 MonitorInstances API 動作的許可。若要關閉執行個體的詳細監控，您的使用者必須具有使用 UnmonitorInstances API 動作的許可。

在啟動時啟用詳細監控

在啟動時，使用下列程序啟用詳細監控。依預設，執行個體會使用基本監控。

Console

在執行個體啟動時啟用詳細監控

使用 Amazon EC2 主控台啟動執行個體時，請在進階詳細資訊下，選取詳細 CloudWatch 監控核取方塊。

AWS CLI

在執行個體啟動時啟用詳細監控

請使用 `--monitoring` 選項並搭配 run-instances 命令。

```
--monitoring Enabled=true
```

PowerShell

在執行個體啟動時啟用詳細監控

使用 New-EC2Instance cmdlet 搭配 `-Monitoring` 參數。

```
-Monitoring $true
```

管理詳細監控

使用下列程序來管理執行中或已停止之執行個體的詳細監控。

Console

若要管理詳細監控

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體。
4. 選擇動作、監控和疑難排解、管理詳細監控。
5. 在詳細監控頁面上，針對詳細監控，執行下列其中一項操作：
   - 詳細監控 – 選取啟用。
   - 基本監控 – 清除啟用。
6. 選擇確認。

AWS CLI

若要啟用詳細監控

使用下列 monitor-instances 命令。

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

若要停用詳細監控

使用 unmonitor-instances 命令。

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

PowerShell

若要啟用詳細監控

使用 Start-EC2InstanceMonitoring cmdlet。

```
Start-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

若要停用詳細監控

使用 Stop-EC2InstanceMonitoring cmdlet。

```
Stop-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

您的執行個體可用的 CloudWatch 指標

Amazon EC2 將指標傳送至 Amazon CloudWatch。您可以使用 AWS 管理主控台、 AWS CLI或 API 來列出 Amazon EC2 傳送至 CloudWatch 的指標。每個資料點預設會涵蓋執行個體從啟動時間後 5 分鐘內的活動。若您已啟用詳細監控功能，則每個資料點會涵蓋從啟動時間後下一分鐘的活動。請注意，針對 Minimum (最小值)、Maximum (最大值) 和 Average (平均值) 等統計數字，EC2 提供的指標最精細的程度是 1 分鐘。

如需有關如何使用 AWS 管理主控台 或 檢視可用指標的資訊 AWS CLI，請參閱《Amazon CloudWatch 使用者指南》中的檢視可用指標。

如需取得這些指標統計資料的資訊，請參閱執行個體 CloudWatch 指標的統計資料。。

內容

- 執行個體指標
- 加速器指標
- CPU 額度指標
- 專用執行個體指標
- Nitro 型執行個體的 Amazon EBS 指標
- 狀態檢查指標
- 流量鏡像指標
- Auto Scaling 群組指標
- Amazon EC2 指標維度

- Amazon EC2 用量指標

執行個體指標

AWS/EC2 命名空間包含下列執行個體指標。

指標	說明	單位	有意義的統計資料
CPUUtilization	Amazon EC2 用於執行 EC2 執行個體的實體 CPU 時間百分比，其中包括執行使用者程式碼和 Amazon EC2 程式碼所花費的時間。 整體而言，CPUUtilization 是訪客 CPUUtilization 和 Hypervisor CPUUtilization 的總和。 由於舊式裝置模擬、非舊式裝置組態、頻繁中斷的工作負載、即時遷移和即時更新等因素，作業系統中工具顯示的百分比可能與 CloudWatch 不同。	Percent	• Average • Minimum • Maximum
DiskReadOps	在指定期間，執行個體可用之所有執行個體存放區磁碟區的已完成讀取操作。 若要計算該期間的每秒平均 I/O 操作數 (IOPS)，請將該期間的總操作數除以該期間的秒數。 如果沒有執行個體存放區磁碟區，數值可能為 0 或指標未回報。	Count	• Sum • Average • Minimum • Maximum
DiskWriteOps	在指定期間，執行個體可用之所有執行個體存放區磁碟區的已完成寫入操作。 若要計算該期間的每秒平均 I/O 操作數 (IOPS)，請將該期間的總操作數除以該期間的秒數。	Count	• Sum • Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
	如果沒有執行個體存放區磁碟區，數值可能為 0 或指標未回報。		
DiskReadBytes	從執行個體可用之所有執行個體存放區磁碟區讀取的位元組。 此指標用來判斷應用程式從執行個體硬碟中讀取的資料磁碟區。這可以用來判斷應用程式的速度。 所報告的數目是在期間內收到的位元組總數。如果您要使用基本 (5 分鐘) 監控，則可以將此數字除以 300，以找到位元組數/秒。如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME，尋找每秒位元組數。舉例來說，如果您已在 CloudWatch 中將 DiskReadBytes 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以位元組數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。 如果沒有執行個體存放區磁碟區，數值可能為 0 或指標未回報。	Bytes	• Sum • Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
DiskWriteBytes	寫入至執行個體可用之所有執行個體存放區磁碟區的位元組。 此指標用來判斷應用程式寫入至執行個體硬碟中的資料磁碟區。這可以用來判斷應用程式的速度。 所報告的數目是在期間內收到的位元組總數。如果您要使用基本 (5 分鐘) 監控，則可以將此數字除以 300，以找到位元組數/秒。如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME，尋找每秒位元組數。舉例來說，如果您已在 CloudWatch 中將 DiskWriteBytes 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以位元組數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。 如果沒有執行個體存放區磁碟區，數值可能為 0 或指標未回報。	Bytes	• Sum • Average • Minimum • Maximum
MetadataNoToken	已使用不使用字符的方法成功存取執行個體中繼資料服務 (IMDS) 的次數。 此指標用來判斷是否有任何正在使用執行個體中繼資料服務第 1 版 (IMDSv1) 的程序，在未使用字符的情況下存取執行個體中繼資料。如果所有請求都使用字符後端工作階段，也就是執行個體中繼資料服務第 2 版 (IMDSv2)，則值為 0。如需詳細資訊，請參閱轉換為使用 執行個體中繼資料服務第 2 版。	Count	• Sum • Percentiles

指標	說明	單位	有意義的統計資料
MetadataNoTokenRejected	在停用 IMDSv1 後嘗試 IMDSv1 呼叫的次數。 如果出現此指標，則表示已嘗試 IMDSv1 呼叫，並遭拒絕。您可以重新啟用 IMDSv1 或確保所有呼叫都使用 IMDSv2。如需詳細資訊，請參閱轉換為使用 執行個體中繼資料服務第 2 版。	Count	• Sum • Percentiles
NetworkIn	執行個體在所有網路介面上收到的位元組數目。此指標識別流向單一執行個體之傳入網路流量的磁碟區。 所報告的數目是在期間內收到的位元組總數。如果您要使用基本 (5 分鐘) 監控並且統計數字為總和，則可以將此數字除以 300，以找出每秒的位元組數。如果您具有詳細 (1 分鐘) 監控並且統計數字為總和，請將它除以 60。	Bytes	• Sum • Average • Minimum • Maximum
NetworkOut	執行個體在所有網路介面上送出的位元組數目。此指標識別來自單一執行個體之傳出網路流量的磁碟區。 所報告的數目是在期間內送出的位元組總數。如果您要使用基本 (5 分鐘) 監控並且統計數字為總和，則可以將此數字除以 300，以找出每秒的位元組數。如果您具有詳細 (1 分鐘) 監控並且統計數字為總和，請將它除以 60。	Bytes	• Sum • Average • Minimum • Maximum
NetworkPacketsIn	執行個體在所有網路介面上收到的封包數目。此指標識別單一執行個體上的傳入流量 (封包數目)。 此指標僅適用於基本監控 (每隔 5 分鐘)。若要計算執行個體在 5 秒時間每秒接收的封包數 (PPS)，請將 Sum 統計數字值除以 300。	Count	• Sum • Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
NetworkPackets Out	執行個體在所有網路介面上送出的封包數目。此指標識別單一執行個體上的傳出流量 (封包數目)。 此指標僅適用於基本監控 (每隔 5 分鐘)。若要計算執行個體在 5 分鐘時間每秒傳送的封包數 (PPS)，請將 Sum 統計數字值除以 300。	Count	• Sum • Average • Minimum • Maximum

加速器指標

AWS/EC2 命名空間包含加速運算執行個體的下列加速器指標。僅在加速運算執行個體類型的子集上受支援。

指標	說明	單位	有意義的統計資料
GPUPowerUtilization	有效功率用量佔有最大有效功率的百分比。	百分比	• Average • Minimum • Maximum

CPU 額度指標

AWS/EC2 命名空間包含高載效能執行個體的下列 CPU 額度指標。

指標	說明	單位	有意義的統計資料
CPUCreditUsage	執行個體為 CPU 使用率花費的 CPU 額度數量。一個 CPU 額度等於一個 vCPU 以 100% 使用率執行 1 分鐘，或同等的 vCPU、使用率與時間的組合 (例如，一個 vCPU 以 50% 使用率執行 2 分鐘，或兩個 vCPU 以 25% 使用率執行 2 分鐘)。	額度 (vCPU-分鐘)	• Sum • Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
	CPU 額度指標僅提供 5 分鐘頻率。如果您要指定大於 5 分鐘的期間，請使用 Sum 統計資訊代替 Average 統計資訊。		
CPUCreditBalance	自執行個體啟動或開始後，累積獲得的 CPU 額度數量。如果是 T2 Standard，CPUCreditBalance 也包含已產生的啟動額度。 獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。額度餘額有最大值限制，它取決於執行個體大小。到達限制之後，任何獲得的新額度都會遭到捨棄。如果是 T2 Standard，啟動額度不會計入此限制。 CPUCreditBalance 中的額度可供執行個體支付以大幅提升並超越基準 CPU 使用率。 當執行個體執行時，CPUCreditBalance 中的額度不會過期。當 T3 或 T3a 執行個體停止時，CPUCreditBalance 值會持續保存七天。因此，所有累積的額度都會消失。當 T2 執行個體停止時，CPUCreditBalance 值不會持續保存，而且所有累積的額度都將消失。 CPU 額度指標僅提供 5 分鐘頻率。	額度 (vCPU-分鐘)	• Sum • Average • Minimum • Maximum
CPUSurplusCreditBalance	當 unlimited 執行個體的 CPUCreditBalance 值為 0 時，該執行個體已支出的剩餘額度數量。 CPUSurplusCreditBalance 值由獲得的 CPU 額度支付。如果剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量，超過最大值的支出剩餘額度將必須負擔額外的費用。 CPU 額度指標僅提供 5 分鐘頻率。	額度 (vCPU-分鐘)	• Sum • Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
CPUSurplusCreditsCharged	若支出剩餘額度數量未由獲得的 CPU 額度付清，會產生額外的費用。 發生以下任何情況時，將收取支出剩餘額度的費用。 • 支出剩餘額度超過執行個體在 24 小時期間可獲得的最大額度數量。在小時結束時，將收取超過最大值的支出剩餘額度的費用。 • 執行個體已停止或終止。 • 執行個體從 unlimited 切換至 standard。 CPU 額度指標僅提供 5 分鐘頻率。	額度 (vCPU-分鐘)	• Sum • Average • Minimum • Maximum

專用執行個體指標

AWS/EC2 命名空間包含下列 T3 專用執行個體的指標。

指標	說明	單位	有意義的統計資料
DedicatedHostCPUUtilization	專用執行個體上執行的執行個體目前正在使用的已配置運算容量百分比。	Percent	• Sum • Average • Minimum • Maximum

Nitro 型執行個體的 Amazon EBS 指標

AWS/EC2 命名空間包含連接至 Nitro 型執行個體 (不屬於裸機執行個體) 的磁碟區的其它 Amazon EBS 指標。

指標	說明	單位	有意義的統計資料
InstanceEBSIOPSExceededCheck	報告應用程式是否嘗試在最後一分鐘內驅動超過執行個體 EBS IOPS 上限的 IOPS。此指標可以是 0 (未超過 IOPS) 或 1 (超過 IOPS)。	無	• 總和 • 平均數 • 下限 • 上限
InstanceEBSThroughputExceededCheck	報告應用程式是否嘗試在最後一分鐘內驅動超過執行個體 EBS 輸送量上限的輸送量。此指標可以是 0 (未超過輸送量) 或 1 (超過輸送量)。	無	• 總和 • 平均數 • 下限 • 上限
EBSReadOps	在指定期間，從連接至執行個體的所有 Amazon EBS 磁碟區完成讀取的操作數。 若要計算該期間的每秒平均讀取 I/O 操作數 (讀取 IOPS)，請將該期間的總操作數除以該期間的秒數。如果您正使用基本 (5 分鐘) 監控，則可以將此數字除以 300，以計算讀取 IOPS。如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME ，尋找每秒操作數。舉例來說，如果您已在 CloudWatch 中將 EBSReadOps 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以操作數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。	計數	• 總和 • 平均數 • 下限 • 上限
EBSWriteOps	在指定期間，從連接至執行個體的所有 EBS 磁碟區完成寫入的操作數。 若要計算該期間的每秒平均寫入 I/O 操作數 (寫入 IOPS)，請將該期間的總操作數除以該期間的秒數。如果您正使用基本 (5 分鐘) 監控，則可以將此數字除以 300，以計算寫入 IOPS。	計數	• 總和 • 平均數 • 下限 • 上限

指標	說明	單位	有意義的統計資料
	如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME ，尋找每秒操作數。舉例來說，如果您已在 CloudWatch 中將 EBSWriteOps 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以操作數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。		
EBSReadBytes	在指定期間內，從連接至執行個體的所有 EBS 磁碟區所讀取的位元組。 所報告的數目是在該期間內讀取的位元組總數。如果您正使用基本 (5 分鐘) 監控，則可以將此數字除以 300，以得到所讀取的位元組數/秒。如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME ，尋找每秒位元組數。舉例來說，如果您已在 CloudWatch 中將 EBSReadBytes 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以位元組數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。	位元組	• 總和 • 平均數 • 下限 • 上限

指標	說明	單位	有意義的統計資料
EBSWriteBytes	在指定期間內，從所有連接至執行個體的 EBS 磁碟區所寫入的位元組。 所報告的數目是在該期間內寫入的位元組總數。如果您正使用基本 (5 分鐘) 監控，則可以將此數字除以 300，得到所寫入的位元組數/秒。如果您具有詳細 (1 分鐘) 監控，請將它除以 60。您也可以使用 CloudWatch 指標數學函數 DIFF_TIME ，尋找每秒位元組數。舉例來說，如果您已在 CloudWatch 中將 EBSWriteBytes 圖表化為 m1，則指標數學公式 m1/(DIFF_TIME(m1)) 會傳回以位元組數/秒為單位的指標。如需 DIFF_TIME 和其他指標數學函數的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」的使用指標數學。	位元組	• 總和 • 平均數 • 下限 • 上限
EBSIOBalance%	提供叢發儲存貯體中剩餘 I/O 額度百分比資訊。只有基本監控才提供此指標。 此指標僅適用於一些大小為 *.4xlarge 及更小的執行個體，至少每 24 小時維持最佳效能 30 分鐘。 Sum 統計資料不適用於此指標。	百分比	• 下限 • 上限
EBSByteBalance%	提供叢發儲存貯體中剩餘傳輸量額度百分比的資訊。只有基本監控才提供此指標。 此指標僅適用於一些大小為 *.4xlarge 及更小的執行個體，至少每 24 小時維持最佳效能 30 分鐘。 Sum 統計資料不適用於此指標。	百分比	• 下限 • 上限

如需 EBS 磁碟區提供之指標的詳細資訊，請參閱「Amazon EBS 使用者指南」中的 Amazon EBS 磁碟區指標。如需 EC2 Fleet 和 Spot Fleet 指標的詳細資訊，請參閱 使用 CloudWatch 監控 EC2 機群或 Spot 機群。

狀態檢查指標

根據預設，狀態檢查指標的頻率為 1 分鐘，而且免費。對於新啟動的執行個體，只有在執行個體完成初始化狀態 (在執行個體進入 `running` 狀態的幾分鐘內) 之後，才能使用狀態檢查指標資料。如需有關 EC2 狀態檢查的詳細資訊，請參閱Amazon EC2 執行個體的狀態檢查。

`AWS/EC2` 命名空間包含下列狀態檢查指標。

指標	說明	單位	有意義的統計資料
`StatusCheckFailed`	報告執行個體是否在最後一分鐘通過所有狀態檢查。 此指標可以是 `0` (通過) 或 `1` (失敗)。 根據預設，此指標的頻率為 1 分鐘，而且免費。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_Instance`	報告執行個體是否在最後一分鐘通過執行個體狀態檢查。 此指標可以是 `0` (通過) 或 `1` (失敗)。 根據預設，此指標的頻率為 1 分鐘，而且免費。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_System`	報告執行個體是否在最後一分鐘通過系統狀態檢查。 此指標可以是 `0` (通過) 或 `1` (失敗)。 根據預設，此指標的頻率為 1 分鐘，而且免費。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_AttachedEBS`	報告執行個體是否在最後一分鐘通過附接的 EBS 狀態檢查。 此指標可以是 `0` (通過) 或 `1` (失敗)。	Count	• Average • Minimum • Maximum

指標	說明	單位	有意義的統計資料
	根據預設，此指標的頻率為 1 分鐘，而且免費。		

AWS/EBS 命名空間包含下列狀態檢查指標。

指標	說明	單位	有意義的統計資料
VolumeStalledIOCheck	注意：僅適用於 Nitro 執行個體。未針對連接至 Amazon ECS 和 AWS Fargate 任務的磁碟區發佈。 在最後一分鐘報告磁碟區停止的 IO 檢查是通過還是失敗。此指標可以是 0 (通過) 或 1 (失敗)。	無	• Average • Minimum • Maximum

流量鏡像指標

AWS/EC2 命名空間包含鏡像流量的指標。如需詳細資訊，請參閱「Amazon VPC Traffic Mirroring 指南」中的使用 Amazon CloudWatch 監控鏡像流量。

Auto Scaling 群組指標

AWS/AutoScaling 命名空間包含 Auto Scaling 組的指標。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的監控 Auto Scaling 群組和執行個體的 CloudWatch 指標。

Amazon EC2 指標維度

您可以使用下列維度來強化上表中列出的指標。

維度	描述
AutoScalingGroupName	此維度可篩選您為指定容量群組中所有執行個體請求的資料。如果您使用 Auto Scaling，則「Auto Scaling 群組」是您定義的執行個體集合。如果執行個體位於 Auto Scaling 群組中，則此維度僅適用於 Amazon EC2 指標。適用於已啟用詳細或基本監控的執行個體。

維度	描述
ImageId	此維度可篩選您為執行此 Amazon EC2 Amazon Machine Image (AMI) 的所有執行個體請求的資料。適用於已啟用詳細監控的執行個體。
InstanceId	此維度會篩選僅針對已識別執行個體所請求的資料。如此可協助您找出要監控資料的確切執行個體。
InstanceType	此維度可篩選您為以此指定執行個體類型執行的所有執行個體請求的資料。如此可協助您依據執行的執行個體類型將您的資料進行分類。例如，您可以比較來自 m1.small 執行個體與 m1.large 執行個體的資料，判斷哪個執行個體對您的應用程式而言具有較高的商業價值。適用於已啟用詳細監控的執行個體。

Amazon EC2 用量指標

您可以使用 CloudWatch 用量指標來提供您帳戶的資源用量可見度。使用這些指標，以 CloudWatch 圖表和儀表板視覺化目前的服務使用狀況。

Amazon EC2 用量指標對應至 AWS 服務配額。您可以設定警示，在您的用量接近服務配額時發出警示。如需 CloudWatch 與 Service Quotas 整合的詳細資訊，請參閱「Amazon CloudWatch 使用者指南」中的 AWS 用量指標。

Amazon EC2 在 AWS/Usage 命名空間中發佈下列指標。

指標	描述
ResourceCount	您的帳戶中正在執行的特定資源數量。資源由與指標相關聯的維度定義。 此指標最有用的統計資料是 MAXIMUM，它表示 1 分鐘內使用的最大資源數量。

以下維度用於強化 Amazon EC2 發佈的用量指標。

維度	說明
Service	包含資源 AWS 的服務名稱。對於 Amazon EC2 用量指標，此維度的值為 EC2。
Type	正在報告的實體類型。目前，Amazon EC2 用量指標的唯一有效值為 Resource。
Resource	正在執行的資源類型。目前，Amazon EC2 用量指標的唯一有效值為 vCPU，它會傳回正在執行之執行個體的相關資訊。
Class	正在追蹤的資源類別。對於以 vCPU 做為 Resource 維度值的 Amazon EC2 用量指標，有效值為 Standard/OnDemand 、F/OnDemand 、G/OnDemand 、Inf/OnDemand 、P/OnDemand 和 X/OnDemand 。 該維度的值定義指標報告之執行個體類型的第一個字母。例如，Standard/OnDemand 傳回開頭為 A、C、D、H、I、M、R、T 和 Z 之所有執行中執行個體的相關資訊，,而 G/OnDemand 傳回開頭為 G 之所有執行中執行個體的相關資訊。

使用 Amazon EC2 主控台安裝和設定 CloudWatch 代理程式，以新增其他指標

說明
透過 Amazon EC2 主控台安裝和設定 CloudWatch 代理程式目前適用於 Amazon EC2 的 Beta 版，之後可能會有所變更。

根據預設，Amazon CloudWatch 提供基本指標，例如 CPUUtilization和 NetworkIn，用於監控您的 Amazon EC2 執行個體。若要收集其他指標，您可以在 EC2 執行個體上安裝 CloudWatch 代理程式，然後將代理程式設定為發出選取的指標。您可以在每個 EC2 執行個體上手動安裝和設定 CloudWatch 代理程式，而不需要使用 Amazon EC2 主控台來為您執行此操作。

您可使用 Amazon EC2 主控台在執行個體上安裝 CloudWatch 代理程式，並將代理程式設定為發出選取的指標。

或者，若要手動完成此程序，請參閱《Amazon CloudWatch 使用者指南》中的安裝 CloudWatch 代理程式。如需 CloudWatch 代理程式的詳細資訊，請參閱使用 CloudWatch 代理程式收集指標、日誌和追蹤。

目錄

- 先決條件
- 運作方式
- 成本
- 安裝及設定 CloudWatch 代理程式

先決條件

若要使用 Amazon EC2 安裝和設定 CloudWatch 代理程式，您必須符合本節所述的使用者和執行個體先決條件。

> **Tip**
>
> 此功能不適用於所有 AWS 區域。如果此頁面的安裝程序中描述的功能表項目不存在於 Amazon EC2 主控台中，而且您對執行個體的執行位置可以靈活調整，請嘗試其他區域。否則，可以使用《Amazon CloudWatch 使用者指南》中的手動指示來安裝和設定代理程式。

使用者先決條件

若要使用此功能，您的 IAM 主控台使用者或角色必須具有使用 Amazon EC2 和下列 IAM 所需的許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
```

```
            "Resource": "arn:aws:ssm:*:*:parameter/EC2-Custom-Metrics-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:ListCommandInvocations",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

執行個體先決條件

- 執行個體狀態：running
- 支援的作業系統：Linux
- AWS Systems Manager 代理程式 (SSM 代理程式）：已安裝。有關 SSM Agent 的兩個備註：
  - SSM Agent 預先安裝在由 AWS 和信任的第三方提供的一些 Amazon Machine Image AMIs) 上。如需有關支援的 AMIs 和安裝 SSM Agent 指示的詳細資訊，請參閱AWS Systems Manager 使用者指南中的預先安裝 SSM Agent 的 Amazon Machine Images (AMIs)。
  - 如果您遇到 SSM Agent 的問題，請參閱AWS Systems Manager 使用者指南中的 SSM Agent 疑難排解。
- 執行個體的 IAM 許可：必須將下列 AWS 受管政策新增至連接至執行個體的 IAM 角色：
  - AmazonSSMManagedInstanceCore – 讓執行個體使用 Systems Manager 安裝和設定 CloudWatch 代理程式。

- CloudWatchAgentServerPolicy – 讓執行個體使用 CloudWatch 代理程式將資料寫入 CloudWatch。

如需如何將 IAM 許可新增至執行個體的詳細資訊，請參閱《IAM 使用者指南》中的使用執行個體設定檔。

運作方式

在您可以使用 Amazon EC2 控制台安裝和設定 CloudWatch 代理程式之前，您必須先確認您的 IAM 使用者或角色，以及您要新增指標的執行個體符合特定先決條件。然後，您可以使用 Amazon EC2 主控台在選取的執行個體上安裝和設定 CloudWatch 代理程式。

首先符合先決條件

- 您需要必要的 IAM 許可 – 開始使用之前，請確認您的主控台使用者或角色具有使用此功能所需的 IAM 許可。
- 執行個體 – 若要使用此功能，您的 EC2 執行個體必須是 Linux 執行個體、已安裝 SSM Agent、具備所需的 IAM 許可，且正在執行中。

接下來，您即可使用此功能

1. 選取您的執行個體 – 在 Amazon EC2 主控台中，您可以選取要安裝和設定 CloudWatch 代理程式的執行個體。然後，選擇設定 CloudWatch 代理程式來開始流程。
2. 驗證 SSM 代理程式 – Amazon EC2 會檢查 SSM 代理程式是否已在每個執行個體上安裝和啟動。任何未通過此檢查的執行個體都會從流程中排除。在此流程期間，SSM Agent 會用於在執行個體中執行動作。
3. 驗證 IAM 許可 – Amazon EC2 會檢查每個執行個體是否具有此流程所需的 IAM 許可。任何未通過此檢查的執行個體都會從流程中排除。IAM 許可可讓 CloudWatch 代理程式從執行個體收集指標，並與 整合 AWS Systems Manager 以使用 SSM 代理程式。
4. 驗證 CloudWatch 代理程式 – Amazon EC2 會檢查 CloudWatch 代理程式是否已在每個執行個體上安裝並執行。如果任何執行個體未通過此檢查，Amazon EC2 會為您安裝和啟動 CloudWatch 代理程式。此程序完成後，CloudWatch 代理程式會在每個執行個體上收集選取的指標。
5. 選取指標設定 – 您可以從執行個體中選取要讓 CloudWatch 代理程式發出的指標。選取後，Amazon EC2 會將設定檔案存放在 Parameter Store 中，該檔案會保留到流程完成為止。除非流程中斷，否則 Amazon EC2 將從 Parameter Store 中刪除設定檔案。請注意，如果您未選取指標，但先前已將其新增至執行個體，則此流程完成時，將會從執行個體中移除該指標。

6. 更新 CloudWatch 代理程式設定 – Amazon EC2 會將指標設定傳送至 CloudWatch 代理程式。這是流程中的最後一個步驟。如果成功，您的執行個體可以發出選取指標的資料，Amazon EC2 會從 Parameter Store 中刪除設定檔案。

成本

您在此流程期間新增的其他指標會以自訂指標計費。如需 CloudWatch 指標定價的詳細資訊，請參閱 Amazon CloudWatch 定價。

安裝及設定 CloudWatch 代理程式

您可以使用 Amazon EC2 主控台來安裝和設定 CloudWatch 代理程式，以新增其他指標。

> **Note**
>
> 每次執行此流程時，您都會覆寫現有的 CloudWatch 代理程式設定。如果您未選取先前選取的指標，則會從執行個體中移除該指標。

使用 Amazon EC2 主控台安裝和設定 CloudWatch 代理程式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取要安裝和設定 CloudWatch 代理程式的執行個體。
4. 選擇動作、監控和疑難排解、設定 CloudWatch 代理程式。
5. 對於流程中的每個步驟，請閱讀主控台文字，然後選擇下一步。
6. 若要完成流程，請在最後一個步驟中，選擇完成。

若要更新 Amazon EC2 主控台建立的代理程式組態

可以手動自訂 EC2 主控台建立的組態。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的手動建立或編輯 CloudWatch 代理程式組態檔案。

執行個體 CloudWatch 指標的統計資料。

您可以取得執行個體 CloudWatch 指標的統計資料。統計資料是隨著指定期間的指標資料彙總。CloudWatch 會根據自訂資料或在 中由其他服務 AWS 提供給 CloudWatch 的指標資料點提供統計

資料。彙總會使用在您指定期間的命名空間、公制名稱、維度和量測資料點單位來進行。下表說明可用的統計資料。

統計數字	描述
Minimum	在指定期間內觀察到的最低值。您可以使用這個值來判斷適用於您應用程式的低活動磁碟區。
Maximum	在指定期間內觀察到的最高值。您可以使用這個值來判斷適用於您應用程式的高活動磁碟區。
Sum	為符合指標總和提交的所有值。這個統計資料在決定指標的總磁碟區時非常有用。
Average	指定期間內的 Sum/SampleCount 值。透過比較這個統計資料與 Minimum 和 Maximum，您可以確定指標的完整範圍，以及 Minimum 和 Maximum 與平均使用的接近程度。這個比較可協助您知道何時需要增加或減少您的資源。
SampleCount	用於統計資料計算的資料點計數 (數值)。
pNN.NN	指定百分位數的值。您可以指定任何百分位數，最多使用兩位小數 (例如，p95.45)。

目錄

- 取得特定執行個體的統計資料
- 彙總所有執行個體的統計資料
- 依據 Auto Scaling 群組彙總統計資料
- 依據 AMI 的彙總統計資料

取得特定執行個體的統計資料

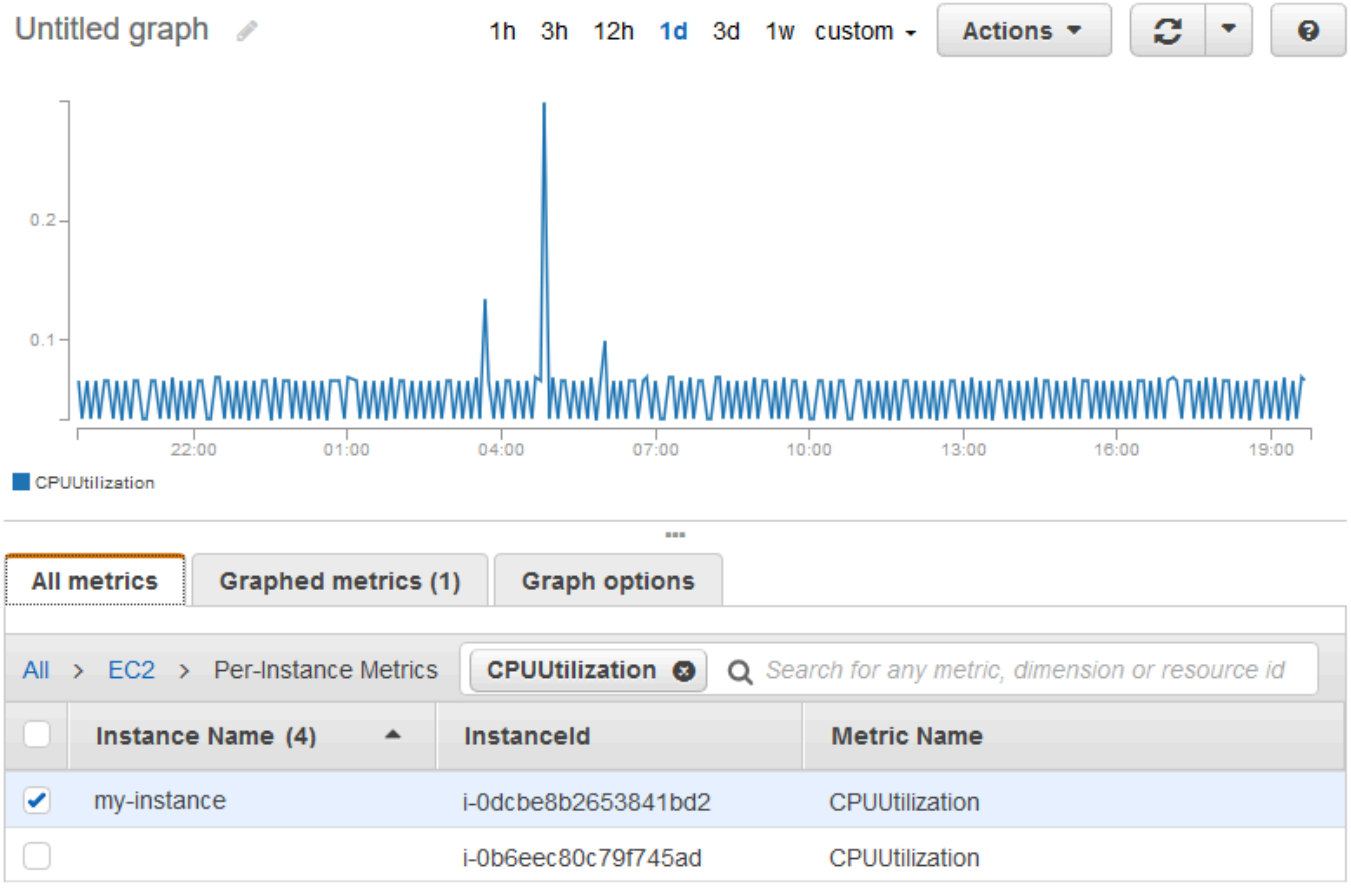
您可以使用 AWS 管理主控台 或 AWS CLI 來取得特定執行個體的統計資料。下列範例示範如何使用 AWS 管理主控台 或 AWS CLI 來判斷特定 EC2 執行個體的 CPU 使用率上限。

要求

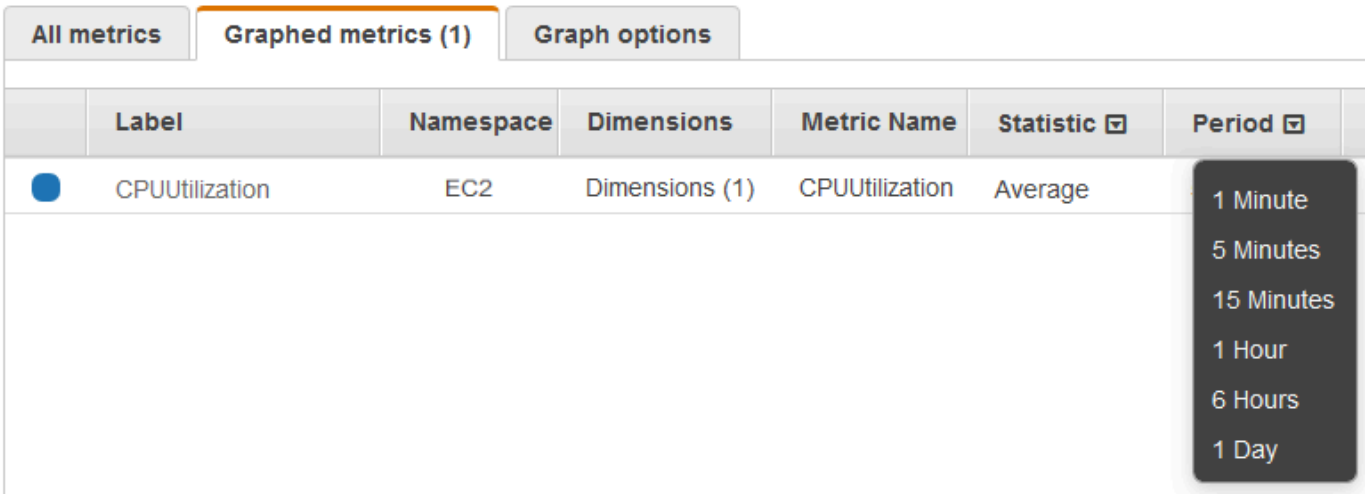
- 您必須擁有該執行個體的 ID。您可以使用 AWS 管理主控台 或 describe-instances 命令以取得執行個體 ID。
- 預設會啟用基本監控，但您可以啟用詳細監控。如需詳細資訊，請參閱管理 EC2 執行個體的詳細監控。

顯示特定執行個體的 CPU 使用率 (主控台)

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇指標。
3. 選擇 EC2 指標命名空間。
4. 選擇每個執行個體指標維度。
5. 在搜尋欄位中，輸入 **CPUUtilization**，然後按 Enter 鍵。選擇特定執行個體的資料列，這會顯示該執行個體 CPUUtilization 指標的圖形。若要為圖形命名，請選擇鉛筆圖示。若要變更時間範圍，請選取一個預先定義的值，或選擇 custom (自訂)。



6. 若要變更指標的統計資料或期間，請選擇 Graphed metrics (圖表化指標) 索引標籤。選擇欄位標題或個別的值，然後選擇不同的值。



取得特定執行個體的 CPU 使用率 (AWS CLI)

透過下列 get-metric-statistics 命令，使用指定的期間和時間間隔來取得指定執行個體的 CPUUtilization 指標：

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2022-10-18T23:18:00 --end-time 2022-10-19T23:18:00
```

下列為範例輸出。每個值代表單一 EC2 執行個體的最大 CPU 使用率百分比。

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T07:18:00Z",
```

```
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

彙總所有執行個體的統計資料

已啟用詳細監控的執行個體可使用彙總統計資料。彙總資料不含使用基本監控的執行個體。在取得所有執行個體的彙總統計資料前，您必須啟用詳細監控 (額外付費)，它以 1 分鐘的間隔時間提供資料。

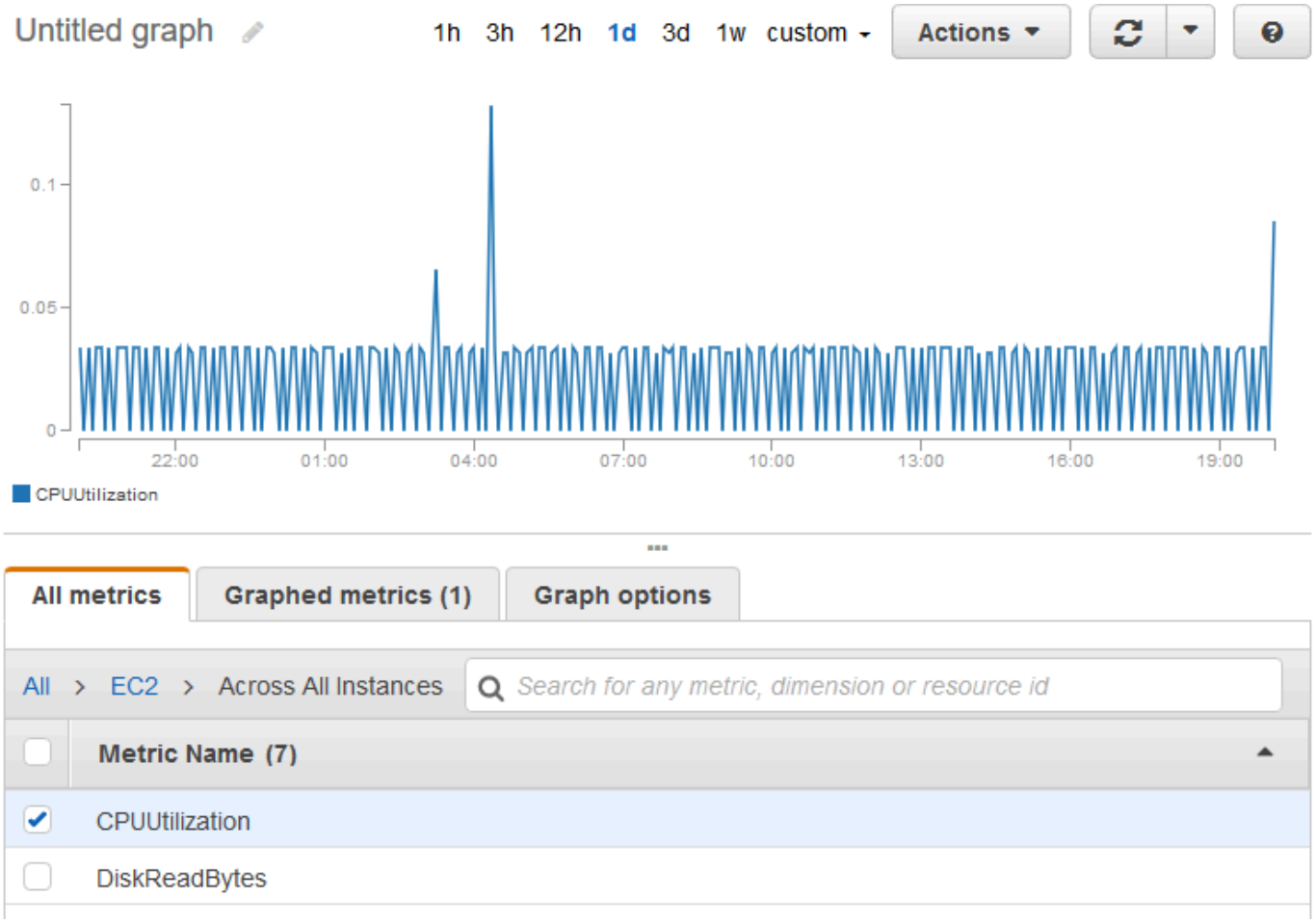
請注意，Amazon CloudWatch 無法跨 AWS 區域彙總資料。區域之間的指標是完全獨立的。

此範例顯示如何使用詳細監控取得 EC2 執行個體的平均 CPU 使用量。由於沒有指定維度，CloudWatch 會傳回 AWS/EC2 命名空間中所有維度的統計資料。

> **Important**
>
> 此擷取 AWS 命名空間中所有維度的技術不適用於您發佈至 Amazon CloudWatch 的自訂命名空間。使用自訂命名空間，您必須指定與任何特定資料點建立關聯的一組完整維度，以擷取包含該資料點的統計資料。

顯示您所有執行個體的平均 CPU 使用率 (主控台)

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇指標。
3. 選擇 EC2 命名空間，然後選擇 Across All Instances (在所有執行個體中)。
4. 選擇包含 CPUUtilization 的資料列，這會顯示您所有 EC2 執行個體指標的圖形。若要為圖形命名，請選擇鉛筆圖示。若要變更時間範圍，請選取一個預先定義的值，或選擇 custom (自訂)。



5. 若要變更指標的統計資料或期間，請選擇 Graphed metrics (圖表化指標) 索引標籤。選擇欄位標題或個別的值，然後選擇不同的值。

取得您所有執行個體的平均 CPU 使用率 (AWS CLI)

如下所示，使用 get-metric-statistics 命令以取得所有執行個體 CPUUtilization 指標的平均值。

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/EC2 \
    --metric-name CPUUtilization \
    --period 3600  --statistics "Average" "SampleCount" \
    --start-time 2022-10-11T23:18:00 \
    --end-time 2022-10-12T23:18:00
```

下列為範例輸出：

```
{
```

```
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2022-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


依據 Auto Scaling 群組彙總統計資料

您可以彙總 Auto Scaling 群組中 EC2 執行個體的統計資料。請注意，Amazon CloudWatch 無法跨 AWS 區域彙總資料。區域之間的指標是完全獨立的。

此範例顯示如何擷取一個 Auto Scaling 群組寫入磁碟的總位元組。總計是計算指定 Auto Scaling 群組中的所有 EC2 執行個體，以 24 小時為間隔取得每隔 1 分鐘的數量。

顯示 Auto Scaling 群組中執行個體的 DiskWriteBytes (主控台)

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇指標。
3. 選擇 EC2 命名空間，然後選擇 By Auto Scaling Group (依據 Auto Scaling 群組)。
4. 選擇 DiskWriteBytes 指標的資料列與指定的 Auto Scaling 群組，這會顯示 Auto Scaling 群組中執行個體指標的圖形。若要為圖形命名，請選擇鉛筆圖示。若要變更時間範圍，請選取一個預先定義的值，或選擇 custom (自訂)。
5. 若要變更指標的統計資料或期間，請選擇 Graphed metrics (圖表化指標) 索引標籤。選擇欄位標題或個別的值，然後選擇不同的值。

顯示 Auto Scaling 群組中執行個體的 DiskWriteBytes (AWS CLI)

使用 get-metric-statistics 命令，如下所示。

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
 --period 360 \
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --
start-time 2022-10-16T23:18:00 --end-time 2022-10-18T23:18:00
```

下列為範例輸出：

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2022-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2022-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

依據 AMI 的彙總統計資料

您可以使用 AMI 為已啟用詳細監控的執行個體彙總統計資料。彙總資料不含使用基本監控的執行個體。在取得所有執行個體的彙總統計資料前，您必須啟用詳細監控 (額外付費)，它以 1 分鐘的間隔時間提供資料。

請注意，Amazon CloudWatch 無法跨 AWS 區域彙總資料。區域之間的指標是完全獨立的。

此範例顯示如何判斷使用特定 Amazon Machine Image (AMI) 之所有執行個體的平均 CPU 使用率。此平均是以一天期間內每 60 秒的時間間隔計算。

依據 AMI 顯示平均 CPU 使用率 (主控台)

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。

2. 在導覽窗格中，選擇指標。
3. 選擇 EC2 命名空間，然後選擇 By Image (AMI) Id (依據映像 (AMI) ID)。
4. 選擇 CPUUtilization 指標與特定 AMI，這會顯示指定 AMI 的指標圖形。若要為圖形命名，請選擇鉛筆圖示。若要變更時間範圍，請選取一個預先定義的值，或選擇 custom (自訂)。
5. 若要變更指標的統計資料或期間，請選擇 Graphed metrics (圖表化指標) 索引標籤。選擇欄位標題或個別的值，然後選擇不同的值。

取得某映像 ID 的平均 CPU 使用率 (AWS CLI)

使用 get-metric-statistics 命令，如下所示。

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-
time 2022-10-10T00:00:00 --end-time 2022-10-11T00:00:00
```

下列為範例輸出。每個值代表執行指定 AMI 之 EC2 執行個體的平均 CPU 使用率百分比。

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

檢視執行個體的監控圖表

啟動執行個體後，您可開啟 Amazon EC2 主控台，在監控索引標籤檢視執行個體的監控圖表。每個圖表都以一個可用的 Amazon EC2 指標為基礎。

下列圖表可供使用：

- 平均 CPU 使用率 (百分比)
- 平均磁碟讀取 (位元組)
- 平均磁碟寫入 (位元組)
- 最大網路輸入 (位元組)
- 最大網路輸出 (位元組)
- 磁碟讀取操作摘要 (計數)
- 磁碟寫入操作摘要 (計數)
- 狀態摘要 (任意)
- 執行個體狀態摘要 (計數)
- 系統狀態摘要 (計數)

如需指標及其向圖表提供資料的詳細資訊，請參閱 您的執行個體可用的 CloudWatch 指標。

使用 CloudWatch 主控台繪製指標圖表

您亦可使用 CloudWatch 主控台，針對 Amazon EC2 及其他 AWS 服務所產生的指標資料製作圖表。如需詳細資訊，請參閱「Amazon CloudWatch 使用者指南」 中的圖形指標。

為執行個體建立 CloudWatch 警示

您可以建立 CloudWatch 警示，以監控其中一個執行個體的 CloudWatch 指標。CloudWatch 會在指標達到您指定的閾值時，自動傳送通知給您。您可以使用 Amazon EC2 主控台或 CloudWatch 主控台提供的更多進階選項，建立 CloudWatch 警示。

使用 CloudWatch 主控台建立警示

如需範例，請參閱 Amazon CloudWatch 使用者指南 中的建立 Amazon CloudWatch 警示。

使用 Amazon EC2 主控台建立警示

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選擇 Actions (動作)、Monitor and troubleshoot (監控和故障診斷)、Manage CloudWatch alarms (管理 CloudWatch 警示)。
4. 在管理 CloudWatch 警示詳細資訊頁面上，於新增或編輯警示 下，選取建立警示。
5. 針對警示通知，選擇是否設定 Amazon Simple Notification Service (Amazon SNS) 通知。輸入現有 Amazon SNS 主題或輸入名稱以建立新主題。
6. 在警示動作選擇是否要指定觸發警示時採取的動作。從清單中選擇一個動作。
7. 針對警示閾值，選取警示的指標和準則。例如，若要建立會在 CPU 使用率達到 80% 並維持 5 分鐘時觸發的警示，請執行下列操作：
   a. 保留範例分群依據 (平均) 和要取樣的資料類型 (CPU 使用率) 的預設設定。
   b. 在警示時機選擇 **>=** 並在百分比輸入 **`0.80`**。
   c. 在連續期間輸入 **`1`**，並在期間選取 5 分鐘。
8. (選用) 針對 範例指標資料，選擇新增至儀表板。
9. 選擇建立。

您可以從 Amazon EC2 主控台或 CloudWatch 主控台編輯您的 CloudWatch 警示設定。如果您想要刪除警示，可以從 CloudWatch 主控台執行。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的編輯或刪除 CloudWatch 警示。

建立警示以停止、終止、重新啟動或復原執行個體

使用 Amazon CloudWatch 警示動作，您可以建立警示以自動停止、終止、重新啟動或復原您的執行個體。當執行個體不再需要執行，您可以使用停止或終止動作以協助您節省成本。如果發生系統受損，您可以使用重新啟動和復原動作，自動重新啟動這些執行個體或將它們復原到新的硬體。

> **Note**
>
> 如需 Amazon CloudWatch 警示帳單與定價資訊，請參閱 Amazon CloudWatch 使用者指南中的 CloudWatch 帳單與成本。

`AWSServiceRoleForCloudWatchEvents` 服務連結角色 AWS 可讓 代表您執行警示動作。在 AWS 管理主控台、 AWS CLI或 IAM API 中第一次建立警示時，CloudWatch 會為您建立服務連結角色。

在許多情況下，您可能想要自動停止或終止您的執行個體。例如，您可能有專門批次處理薪資作業或科學運算任務的執行個體，它們在執行一段時間後完成工作。您不必讓這些執行個體閒置 (及累積費用)，您可以停止或終止它們以協助您節省成本。使用停止和終止警示動作的主要差別是，如果執行個體之後需要重新執行，您將可以輕鬆地啟動已停止的執行個體，而且您可以保持相同的執行個體 ID 和根磁碟區。不過，您無法啟動已終止的執行個體。相反地，您必須啟動新的執行個體。在停止或終止執行個體時，執行個體儲存體磁碟區上的資料會遺失。

您可以將停止、終止、重新啟動或復原動作，新增至設定於 Amazon EC2 每執行個體指標上的任何提醒，包括 Amazon CloudWatch 提供的基本和詳細監控指標 (在 AWS/EC2 命名空間中)，以及包括 InstanceId 維度的任何自訂指標，只要其值代表有效的執行中 Amazon EC2 執行個體。

> **Important**
>
> 如果缺少指標資料點，狀態檢查警示可以暫時進入 INSUFFICIENT_DATA 狀態。儘管很少見，但當指標報告系統中斷時，即使執行個體狀態良好，也可能會發生這種情況。我們建議您將 INSUFFICIENT_DATA 狀態視為遺失資料，而非警示違規，特別是在將警示設定為停止、終止、重新啟動或復原執行個體時。

主控台支援

您可以使用 Amazon EC2 主控台或 CloudWatch 主控台建立警示。本文件中的程序使用 Amazon EC2 主控台。如需使用 CloudWatch 主控台的程序，請參閱「Amazon CloudWatch 使用者指南」中的建立警示，來停止、終止、重新開機或恢復執行個體。

許可

您必須擁有 iam:CreateServiceLinkedRole 才能建立或修改執行 EC2 警示動作的警示。服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱「IAM 使用者指南」中的建立角色以委派許可權給 AWS 服務。

目錄

- 將停止動作新增至 Amazon CloudWatch 警示
- 新增終止動作至 Amazon CloudWatch 警示
- 將重新開機動作新增至 Amazon CloudWatch 警示
- 將復原動作新增至 Amazon CloudWatch 警示
- Amazon CloudWatch 警示動作案例

將停止動作新增至 Amazon CloudWatch 警示

您可以建立警示，在符合特定閾值時停止 Amazon EC2 執行個體。例如，您可以執行開發或測試執行個體，並偶爾忘記關閉它們。您可以建立警示，在平均 CPU 使用率百分比已低於 10% 達 24 小時時觸發，以通知其為閒置且不再使用。您可以調整閾值、持續時間和期間以符合您的需求，而且您可以新增 Amazon Simple Notification Service (Amazon SNS) 通知，當觸發提醒時，您將會收到電子郵件。

使用 Amazon EBS 磁碟區作為根磁碟區的執行個體可以停止或終止，而使用執行個體儲存體作為根磁碟區的執行個體只能終止。在終止或停止執行個體時，執行個體儲存體磁碟區上的資料會遺失。

建立警示以停止閒置的執行個體 (Amazon EC2 主控台)

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選擇 Actions (動作)、Monitor and troubleshoot (監控和故障診斷)、Manage CloudWatch alarms (管理 CloudWatch 警示)。

   或者，您可以在 Alarm status (警示狀態) 資料欄中選擇加號 (+)。
4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示) 頁面上，執行下列動作：

   a. 選擇 Create an alarm (建立警示)。

   b. 若要在警示觸發時接收電子郵件，請在 Alarm notification (警示通知) 選擇現有的 Amazon SNS 主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

   c. 切換 Alarm action (警示動作)，然後選擇 Stop (停止)。

   d. 若要將 Group samples by (分組範例條件) 和 Type of data to sample (要作為範例的資料類型)，請選擇統計資料和指標。在此範例中，選擇 Average (平均) 和 CPU Utilization (CPU 使用率)。

   e. 針對 Alarm When (警示時間) 和 Percent (百分比)，指定指標閾值。在此範例中，指定 <= 和 10%。

   f. 針對 Consecutive period (連續期間) 和Period (期間)，指定警示的評估期間。在此範例中，請指定 1 個每隔 5 Minute (5 分鐘) 的連續期間。

   g. Amazon CloudWatch 會自動為您建立警示名稱。若要變更名稱，請在 Alarm name (警示名稱) 中輸入新名稱。警示名稱只能包含 ASCII 字元。

> **Note**
>
> 您可以在建立警示之前根據自己的需求調整警示組態，也可以稍後再進行編輯。這包括指標、閾值、持續時間、動作和通知設定。不過，在您建立警示之後，以後就無法編輯其名稱。

h. 選擇建立。

新增終止動作至 Amazon CloudWatch 警示

您可以建立警示，在符合特定閾值時自動終止 EC2 執行個體 (前提是執行個體未啟用終止保護)。例如，您可能想要在執行個體完成作業時予以終止，而且不再需要該執行個體。如果您之後還要使用該執行個體，您應該停止而非終止執行個體。在終止執行個體時，執行個體儲存體磁碟區上的資料會遺失。如需為執行個體啟用和停用終止保護的詳細資訊，請參閱 變更執行個體終止保護。

建立警示以終止閒置的執行個體 (Amazon EC2 主控台)

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選擇 Actions (動作)、Monitor and troubleshoot (監控和故障診斷)、Manage CloudWatch alarms (管理 CloudWatch 警示)。

   或者，您可以在 Alarm status (警示狀態) 資料欄中選擇加號 (+)。
4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示) 頁面上，執行下列動作：

   a. 選擇 Create an alarm (建立警示)。

   b. 若要在警示觸發時接收電子郵件，請在 Alarm notification (警示通知) 選擇現有的 Amazon SNS 主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

   c. 切換 Alarm action (警示動作)，然後選擇 Terminate (終止)。

   d. 若要將 Group samples by (分組範例條件) 和 Type of data to sample (要作為範例的資料類型)，請選擇統計資料和指標。在此範例中，選擇 Average (平均) 和 CPU Utilization (CPU 使用率)。

e. 針對 Alarm When (警示時間) 和 Percent (百分比)，指定指標閾值。在此範例中，指定 >= 和 10%。

f. 針對 Consecutive period (連續期間) 和Period (期間)，指定警示的評估期間。在此範例中，請指定 24 個每隔 1 Hour (1 小時) 的連續期間。

g. Amazon CloudWatch 會自動為您建立警示名稱。若要變更名稱，請在 Alarm name (警示名稱) 中輸入新名稱。警示名稱只能包含 ASCII 字元。

> **Note**
>
> 您可以在建立警示之前根據自己的需求調整警示組態，也可以稍後再進行編輯。這包括指標、閾值、持續時間、動作和通知設定。不過，在您建立警示之後，以後就無法編輯其名稱。

h. 選擇建立。

將重新開機動作新增至 Amazon CloudWatch 警示

您可以建立 Amazon CloudWatch 警示來監控 Amazon EC2 執行個體，並自動重新啟動執行個體。重新啟動警示動作建議用於執行個體運作狀態檢查失敗 (相對的，復原警示動作則適用於系統運作狀態檢查失敗)。重新啟動執行個體等同於重新啟動作業系統。在大多數情況下，將執行個體重新開機只需要幾分鐘的時間。當您重新啟動執行個體時，它會維持在相同的實體主機上，所以您的執行個體會保有其公有 DNS 名稱、私有 IP 地址和執行個體存放區磁碟區上的任何資料。

有別於停用和重新開始執行個體，重新啟動執行個體不會啟動新的執行個體計費時段 (最少收費一分鐘)。在重新啟動執行個體時，會保留執行個體儲存體磁碟區上的資料。執行個體儲存體磁碟區必須在重新啟動後重新掛載到檔案系統中。如需詳細資訊，請參閱重新啟動您的 Amazon EC2 執行個體。

> **Important**
>
> 為了避免重新開機和復原動作之間的競爭情況，請避免為重新開機警示和復原警示設定相同的評估期間數。我們建議您將重新開機警示設定為三個各一分鐘的評估期間。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的評估警示。

建立警示以重新啟動執行個體 (Amazon EC2 主控台)

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。

3. 選取執行個體，並選擇 Actions (動作)、Monitor and troubleshoot (監控和故障診斷)、Manage CloudWatch alarms (管理 CloudWatch 警示)。

   或者，您可以在 Alarm status (警示狀態) 資料欄中選擇加號

   ()。

4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示) 頁面上，執行下列動作：

   a. 選擇 Create an alarm (建立警示)。

   b. 若要在警示觸發時接收電子郵件，請在 Alarm notification (警示通知) 選擇現有的 Amazon SNS 主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

   c. 切換 Alarm action (警示動作)，然後選擇 Reboot (重新開機)。

   d. 若要將 Group samples by (分組範例條件) 和 Type of data to sample (要作為範例的資料類型)，請選擇統計資料和指標。在此範例中，選擇 Average (平均) 和 Status check failed: instance (狀態檢查失敗：執行個體)。

   e. 針對 Consecutive period (連續期間) 和Period (期間)，指定警示的評估期間。在此範例中，請輸入 3 個每隔 1 分鐘的連續期間。如果停用 1 分鐘，您必須啟用詳細監控，或者您可以選擇停用 5 分鐘。

   f. Amazon CloudWatch 會自動為您建立警示名稱。若要變更名稱，請在 Alarm name (警示名稱) 中輸入新名稱。警示名稱只能包含 ASCII 字元。

   g. 選擇建立。

將復原動作新增至 Amazon CloudWatch 警示

您可以建立 Amazon CloudWatch 提醒，用來監控 Amazon EC2 執行個體。如果執行個體因為基礎硬體故障或需要 AWS 參與修復的問題而受損，您可以自動復原執行個體。已終止的執行個體無法復原。復原後的執行個體與原始執行個體相同，包括執行個體 ID、私有 IP 地址、彈性 IP 地址及所有執行個體中繼資料。

若執行個體不支援復原動作，則 CloudWatch 會防止您將復原動作新增至此執行個體上的警示。

當 `StatusCheckFailed_System` 提醒觸發且復原動作啟動時，您會收到 Amazon SNS 主題通知，這是您在建立提醒時選擇並與復原動作關聯的通知。在執行個體復原期間，執行個體會在重新啟動期間遷移，記憶體內的任何資料都將遺失。當程序完成時，會將資訊發佈到您為此警示設定的 SNS 主題。

訂閱此 SNS 主題的所有使用者都會收到電子郵件通知，其中包含復原嘗試的狀態和進一步的說明。您會發現執行個體在已復原的執行個體上重新啟動。

> **Note**
>
> 復原動作只能用於 `StatusCheckFailed_System`，而非 `StatusCheckFailed_Instance`。

下列為可能導致系統狀態檢查失敗的問題：

- 網路連線中斷
- 系統電力中斷
- 實體主機的軟體問題
- 實體主機上會影響網路連線的硬體問題

復原動作僅在符合某些特性的執行個體上受到支援：如需詳細資訊，請參閱自動執行個體復原功能。

如果您的執行個體具有公有 IP 地址，它將在復原後保留公有 IP 地址。

> **Important**
>
> 為了避免重新開機和復原動作之間的競爭情況，請避免為重新開機警示和復原警示設定相同的評估期間數。我們建議您將復原警示設定為兩個各一分鐘的評估期間。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的評估警示。

建立警示以復原執行個體 (Amazon EC2 主控台)

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取執行個體，並選擇 Actions (動作)、Monitor and troubleshoot (監控和故障診斷)、Manage CloudWatch alarms (管理 CloudWatch 警示)。

   或者，您可以在 Alarm status (警示狀態) 資料欄中選擇加號 (+)。
4. 在 Manage CloudWatch alarms (管理 CloudWatch 警示) 頁面上，執行下列動作：

a. 選擇 Create an alarm (建立警示)。

b. 若要在警示觸發時接收電子郵件，請在 Alarm notification (警示通知) 選擇現有的 Amazon SNS 主題。您首先需要使用 Amazon SNS 主控台建立 Amazon SNS 主題。如需詳細資料，請參閱「Amazon Simple Notification Service 開發人員指南」中的使用 Amazon SNS 進行應用程式至人員 (A2P) 訊息。

> **Note**
>
> 使用者必須訂閱指定的 SNS 主題，以在警示觸發時收到電子郵件通知。即使未指定 SNS 主題或根使用者未訂閱指定的 SNS 主題， AWS 帳戶根使用者 仍會在發生自動執行個體復原動作時收到電子郵件通知。

c. 切換 Alarm action (警示動作)，然後選擇 Recover (復原)。

d. 若要將 Group samples by (分組範例條件) 和 Type of data to sample (要作為範例的資料類型)，請選擇統計資料和指標。在此範例中，選擇 Average (平均) 和 Status check failed: system (狀態檢查失敗：系統)。

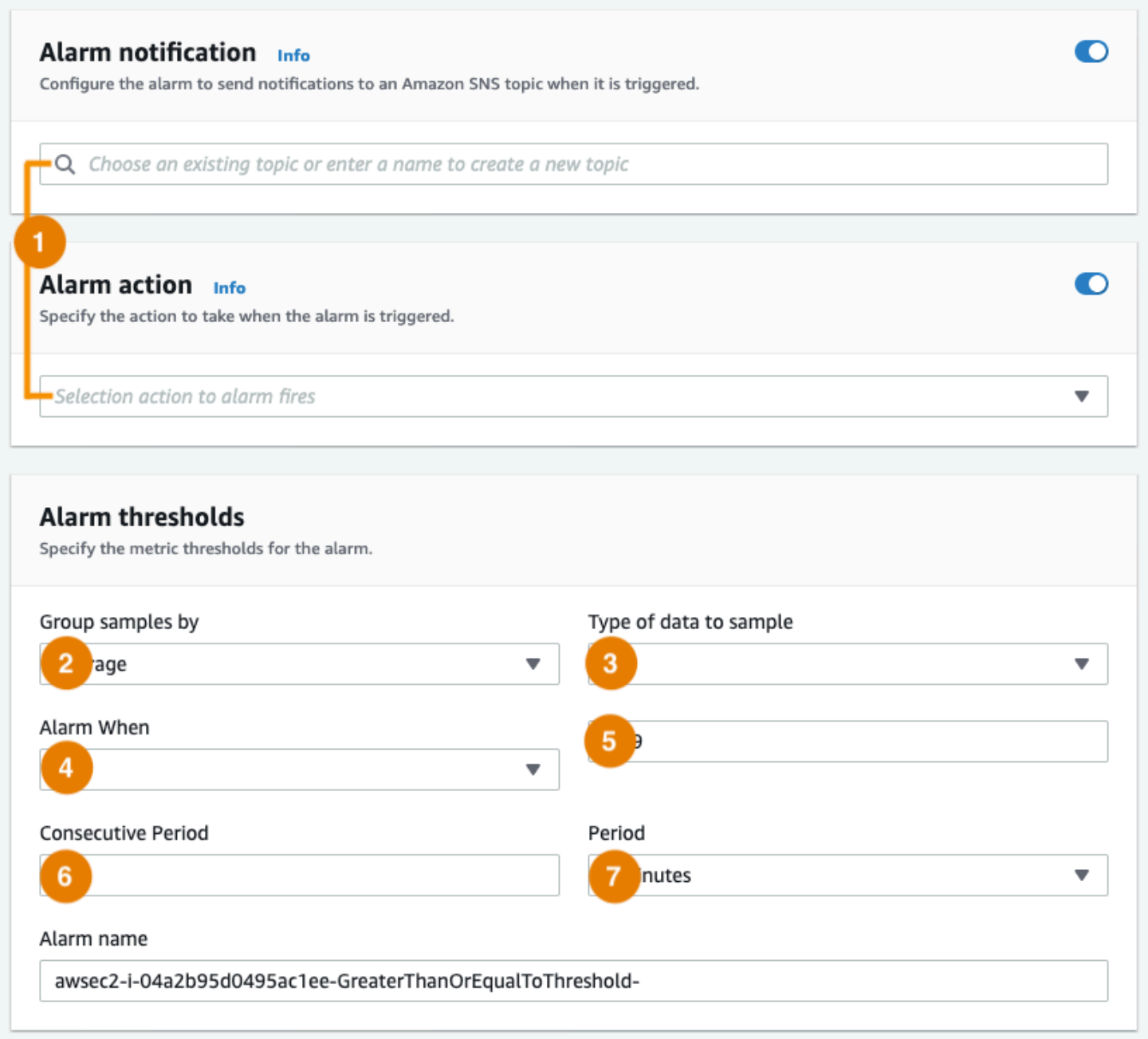
e. 針對 Consecutive period (連續期間) 和Period (期間)，指定警示的評估期間。在此範例中，請輸入 2 個每隔 1 分鐘的連續期間。如果停用 1 分鐘，您必須啟用詳細監控，或者您可以選擇停用 5 分鐘。

f. Amazon CloudWatch 會自動為您建立警示名稱。若要變更名稱，請在 Alarm name (警示名稱) 中輸入新名稱。警示名稱只能包含 ASCII 字元。

g. 選擇建立。

Amazon CloudWatch 警示動作案例

您可以使用 Amazon EC2 主控台來建立提醒動作，在滿足特定條件時停止或終止 Amazon EC2 執行個體。在下列用來設定警示動作的主控台頁面螢幕截圖中，我們為設定加上編號。我們也在隨後的案例中為設定加上編號，協助您建立適當的動作。



案例 1：停止閒置的開發和測試執行個體

建立警示，在用於軟體開發或測試的執行個體已閒置至少一小時後，將其停止。

設定	Value
1	停止
2	最大

設定	Value
3	CPU 使用率
4	<=
5	10%
6	1
7	1 小時

案例 2：停止閒置的執行個體

建立警示，在執行個體已閒置 24 小時後，停止執行個體並傳送電子郵件。

設定	Value
1	停止和寄送電子郵件
2	平均數
3	CPU 使用率
4	<=
5	5%
6	24
7	1 小時

案例 3：針對發生異常高流量的 Web 伺服器傳送電子郵件

建立警示，在執行個體每天超過 10 GB 傳出網路流量後，傳送電子郵件。

設定	Value
1	電子郵件

設定	Value
2	總和
3	網路輸出
4	>
5	10 GB
6	24
7	1 小時

案例 4：停止發生異常高流量的 Web 伺服器

建立警示，在傳出流量每小時超過 1 GB 後，停止執行個體並傳送簡訊 (SMS)。

設定	Value
1	停止並傳送簡訊
2	總和
3	網路輸出
4	>
5	1 GB
6	1
7	1 小時

案例 5：停止受損的執行個體

建立警示，停止未通過三次連續運作狀態檢查的執行個體 (以 5 分鐘間隔執行)。

設定	Value
1	停止
2	平均數
3	狀態檢查失敗：系統
4	-
5	-
6	1
7	15 分鐘

案例 6：批次處理任務完成時終止執行個體

建立警示，當執行批次任務的執行個體不再傳送結果資料時，終止該執行個體。

設定	Value
1	終止
2	最大
3	網路輸出
4	<=
5	100,000 位元組
6	1
7	5 分鐘

使用 EventBridge 自動化 Amazon EC2

您可以使用 Amazon EventBridge 自動化您的 ， AWS 服務 並自動回應系統事件，例如應用程式可用性問題或資源變更。 AWS 服務的事件會以接近即時的方式傳送到 EventBridge。您可建立規則來指示您在意的事件，以及當事件符合規則時執行的動作。可以自動觸發的動作如下：

- 叫用 AWS Lambda 函數
- 叫用 Amazon EC2 執行命令
- 將事件轉送至 Amazon Kinesis Data Streams
- 啟用 AWS Step Functions 狀態機器
- 通知 Amazon SNS 主題
- 通知 Amazon SQS 佇列

以下是如何搭配 Amazon EC2 使用 EventBridge 的範例：

- 每當執行個體進入執行狀態時，啟用 Lambda 函數。
- 建立或修改 Amazon EBS 磁碟區時通知 Amazon SNS 主題。
- 每當其他服務中 AWS 發生特定事件時，使用 Amazon EC2 Run Command 將命令傳送至一或多個 Amazon EC2 執行個體。

如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

Amazon EC2 事件類型

Amazon EC2 支援以下事件類型：

- EC2 AMI 狀態變更
- EC2 快速啟動狀態變更通知
- EC2 機群錯誤
- EC2 機群資訊
- EC2 機群執行個體變更
- EC2 機群 Spot 執行個體請求變更
- EC2 機群狀態變更

- EC2 執行個體重新平衡推薦
- EC2 執行個體狀態 - 變更通知
- EC2 Spot 機群錯誤
- EC2 Spot 機群資訊
- EC2 Spot 機群執行個體變更
- EC2 Spot 機群 Spot 執行個體請求變更
- EC2 Spot 機群狀態變更
- EC2 Spot 執行個體中斷警告
- EC2 Spot 執行個體請求履行
- EC2 ODCR 使用率不足通知

如需有關 Amazon EBS 支援的事件類型的資訊，請參閱 Amazon EBS 的 Amazon EventBridge。

使用 記錄 Amazon EC2 API 呼叫 AWS CloudTrail

Amazon EC2 API 與 AWS CloudTrail 整合，這項服務可提供由使用者、角色或 AWS 服務所採取之動作的記錄。CloudTrail 會將所有 Amazon EC2 API 呼叫擷取為事件。擷取的呼叫包括來自主控台的呼叫。您可以使用 CloudTrail 所收集的資訊，判斷對 Amazon EC2 API 提出的請求、提出請求的 IP 位址、人員或時間等。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是使用根使用者還是使用者憑證提出。
- 請求是否代表 IAM Identity Center 使用者提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務服務提出。

當您建立帳戶 AWS 帳戶 時CloudTrail 會在 中處於作用中狀態，而且您會自動存取 CloudTrail 事件歷史記錄。CloudTrail 事件歷史記錄為 AWS 區域中過去 90 天記錄的管理事件，提供可檢視、可搜尋、可下載且不可變的記錄。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」中的使用 CloudTrail 事件歷史記錄。檢視事件歷史記錄不會產生 CloudTrail 費用。

如需 AWS 帳戶 過去 90 天內持續記錄的事件，請建立線索或 CloudTrail Lake 事件資料存放區。

CloudTrail 追蹤

線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。使用 建立的所有線索 AWS 管理主控台 都是多區域。您可以使用 AWS CLI建立單一或多區域追蹤。建議您建立多區域追蹤，因為您擷取 AWS 區域 帳戶中所有 的活動。如果您建立單一區域追蹤，您只能檢視追蹤 AWS 區域中記錄的事件。如需追蹤的詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的為您的 AWS 帳戶建立追蹤和為組織建立追蹤。

您可以透過建立追蹤，免費將持續管理事件的一個複本從 CloudTrail 傳遞至您的 Amazon S3 儲存貯體，但這樣做會產生 Amazon S3 儲存費用。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。如需 Amazon S3 定價的相關資訊，請參閱 Amazon S3 定價。

CloudTrail Lake 事件資料存放區

CloudTrail Lake 讓您能夠對事件執行 SQL 型查詢。CloudTrail Lake 會將分列式 JSON 格式的現有事件轉換為 Apache ORC 格式。ORC 是一種單欄式儲存格式，針對快速擷取資料進行了最佳化。系統會將事件彙總到事件資料存放區中，事件資料存放區是事件的不可變集合，其依據為您透過套用進階事件選取器選取的條件。套用於事件資料存放區的選取器控制哪些事件持續存在並可供您查詢。如需 CloudTrail Lake 的詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 AWS CloudTrail Lake。

CloudTrail Lake 事件資料存放區和查詢會產生費用。建立事件資料存放區時，您可以選擇要用於事件資料存放區的定價選項。此定價選項將決定擷取和儲存事件的成本，以及事件資料存放區的預設和最長保留期。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。

CloudTrail 中的 Amazon EC2 API 管理事件

管理事件提供有關在 資源上執行的管理操作的資訊 AWS 帳戶。這些也稱為控制平面操作。根據預設，CloudTrail 記錄管理事件。

所有 Amazon EC2 API 動作都會記錄為管理事件。如需 CloudTrail 記錄之 API 動作的清單，請參閱 Amazon EC2 API 參考。例如，對 RunInstances、 DescribeInstances 和 StopInstances動作的呼叫會記錄為管理事件。

Amazon EC2 API 事件範例

一個事件代表任何來源提出的單一請求，並包含請求 API 操作的相關資訊、操作的日期和時間、請求參數等。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此事件不會以任何特定順序出現。

以下日誌檔記錄顯示某位使用者終止了執行個體。

```
{
   "Records":[
      {
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"user"
         },
         "eventTime":"2016-05-20T08:27:45Z",
         "eventSource":"ec2.amazonaws.com",
         "eventName":"TerminateInstances",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"198.51.100.1",
         "userAgent":"aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
         "requestParameters":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d"
               }]
            }
         },
         "responseElements":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d",
                  "currentState":{
                     "code":32,
                     "name":"shutting-down"
                  },
                  "previousState":{
                     "code":16,
                     "name":"running"
                  }
               }]
            }
         },
         "requestID":"be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
         "eventID":"6e12345-2a4e-417c-aa78-7594fEXAMPLE",
         "eventType":"AwsApiCall",
```

```
        "recipientAccountId":"123456789012"
      }
    ]
}
```

如需有關 CloudTrail 記錄內容的詳細資訊，請參閱「AWS CloudTrail 使用者指南」中的CloudTrail 記錄內容。

稽核使用 EC2 Instance Connect 的連線

您可以使用 AWS CloudTrail 來稽核使用 EC2 Instance Connect 連線至執行個體的使用者。

使用 AWS CloudTrail 主控台透過 EC2 Instance Connect 稽核 SSH 活動

1. 前往 https://console.aws.amazon.com/cloudtrail/ 開啟 CloudTrail 主控台。
2. 確認您位於正確的區域。
3. 在導覽窗格中，選擇 Event history (事件歷史記錄)。
4. 對於 Filter (篩選條件)，請選擇 Event source (事件來源)、ec2-instance-connect.amazonaws.com。
5. (選用) 針對 Time range (時間範圍)，選取時間範圍。
6. 選擇 Refresh events (重新整理事件) 圖示。
7. 頁面會顯示與 SendSSHPublicKey API 呼叫對應的事件。使用箭頭展開事件以檢視其他詳細資訊，例如用於進行 SSH 連線的使用者名稱和 AWS 存取金鑰，以及來源 IP 地址。
8. 若要以 JSON 格式顯示完整事件資訊，請選擇 View event (檢視事件)。requestParameters 欄位包含用來進行 SSH 連線的目的地執行個體 ID、OS 使用者名稱和公開金鑰。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"}
```

```
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey ",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNO01234567890EXAMPLE"
        }
     },
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
}
```

如果您已將 AWS 帳戶設定為在 S3 儲存貯體中收集 CloudTrail 事件，您可以以程式設計方式下載和稽核資訊。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」中的取得和檢視 CloudTrail 日誌檔案。

使用 CloudWatch Application Insights 監控 .NET 和 SQL Server 應用程式

CloudWatch Application Insights 可幫助您監控使用 Amazon EC2 執行個體以及其他 AWS 應用程式資源的 .NET 和 SQL Server 應用程式。它會識別和設定金鑰指標日誌，並在所有的應用程式資源和技術堆疊 (例如，Microsoft SQL Server 資料庫、Web (IIS) 和應用程式伺服器、作業系統、負載平衡器和佇列) 中發出警示。並持續監控指標和日誌，以偵測和建立異常及錯誤的關聯。偵測到錯誤和異常時，Application Insights 會產生事件，可用於設定通知或採取動作。為協助故障診斷，它會建立已偵測到問題的自動化儀表板，包括關聯的指標異常和日誌錯誤以及其他洞見，指出可能的根本原因。自動化儀表板可協助您採用靈活的補救動作，讓應用程式保持良好的運作狀態，防止應用程式的最終使用者受到影響。

有關偵測到的問題資訊：

- 問題的簡短摘要
- 問題的開始時間和日期
- 問題嚴重性：高/中/低
- 偵測到的問題狀態：正在進行/已解決
- 洞見：對所偵測問題及可能根本原因自動產生的洞見
- 對洞見的意見回饋：您針對適用於 .NET 和 SQL Server 的 CloudWatch Application Insights 所產生洞見實用性提供的意見回饋
- 相關觀察：詳細檢視與跨各種應用程式元件問題有關的相關日誌指標異常和錯誤程式碼片段

意見回饋

您可以指出針對所偵測問題自動產生的洞見是否有用，提供意見回饋。您對洞見的意見回饋以及您的應用程式診斷 (指標異常和日誌例外狀況)，都會用來提升未來對類似問題的偵測。

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南 中的 CloudWatch Application Insights 文件。

追蹤 Amazon EC2 的免費方案用量

建立 時 AWS 帳戶，您可以使用 AWS 免費方案免費開始使用 Amazon EC2。免費方案優惠會有所不同，具體取決於您是在 2025 年 7 月 15 日之前，還是在 2025 年 7 月 15 日當天或之後建立帳戶。如需詳細資訊，請參閱AWS Billing 《 使用者指南》中的使用 AWS 免費方案探索 AWS 服務。

2025 年 7 月 15 日之前和之後的免費方案優惠

下表會根據 AWS 帳戶 建立日期，比較免費方案優惠：

免費方案優惠	在 2025 年 7 月 15 日之前建立的帳戶	在 2025 年 7 月 15 日當天或之後建立的帳戶
標記為符合免費方案資格的執行個體類型	`t2.micro`, `t3.micro`	`t3.micro`, `t3.small`, `t4g.micro` ，`t4g.small` ，`c7i-flex.large` ，`m7i-flex.large`

免費方案優惠	在 2025 年 7 月 15 日之前建立的帳戶	在 2025 年 7 月 15 日當天或之後建立的帳戶
標記為符合免費方案資格的 Amazon EBS 磁碟區類型	standard, st1, sc1, gp2, 和 gp3	standard, st1, sc1, gp2, 和 gp3
標記為符合免費方案資格的 AMI。	檢查標記為符合免費方案資格的 AMI。	檢查標記為符合免費方案資格的 AMI。
使用限制	受使用限額限制，超過該限制之後，會按使用量付費。	獲得 100 美元的註冊抵用金，並獲得高達 100 美元的額外抵用金。
免費方案持續時間	免費方案自您建立帳戶之日起持續 12 個月。在此期間，如果超過免費方案使用限制，則需要按使用量付費。	免費方案自您建立帳戶之日起持續 6 個月，或當抵用金用完時，以先發生者為準。不能超過免費方案限制。

如需列出符合免費方案資格的執行個體類型

使用 describe-instance-types 命令並搭配 `free-tier-eligible` 篩選條件。

```
aws ec2 describe-instance-types \
    --filters Name=free-tier-eligible,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

如需列出符合免費方案資格的 AMI

使用具有 `free-tier-eligible` 篩選條件的 describe-images 命令。

```
aws ec2 describe-images \
    --filters Name=free-tier-eligible,Values=true \
    --query "Images[*].[ImageId]" \
    --output text | sort
```

追蹤 2025 年 7 月 15 日之前建立之帳戶的免費方案使用情況

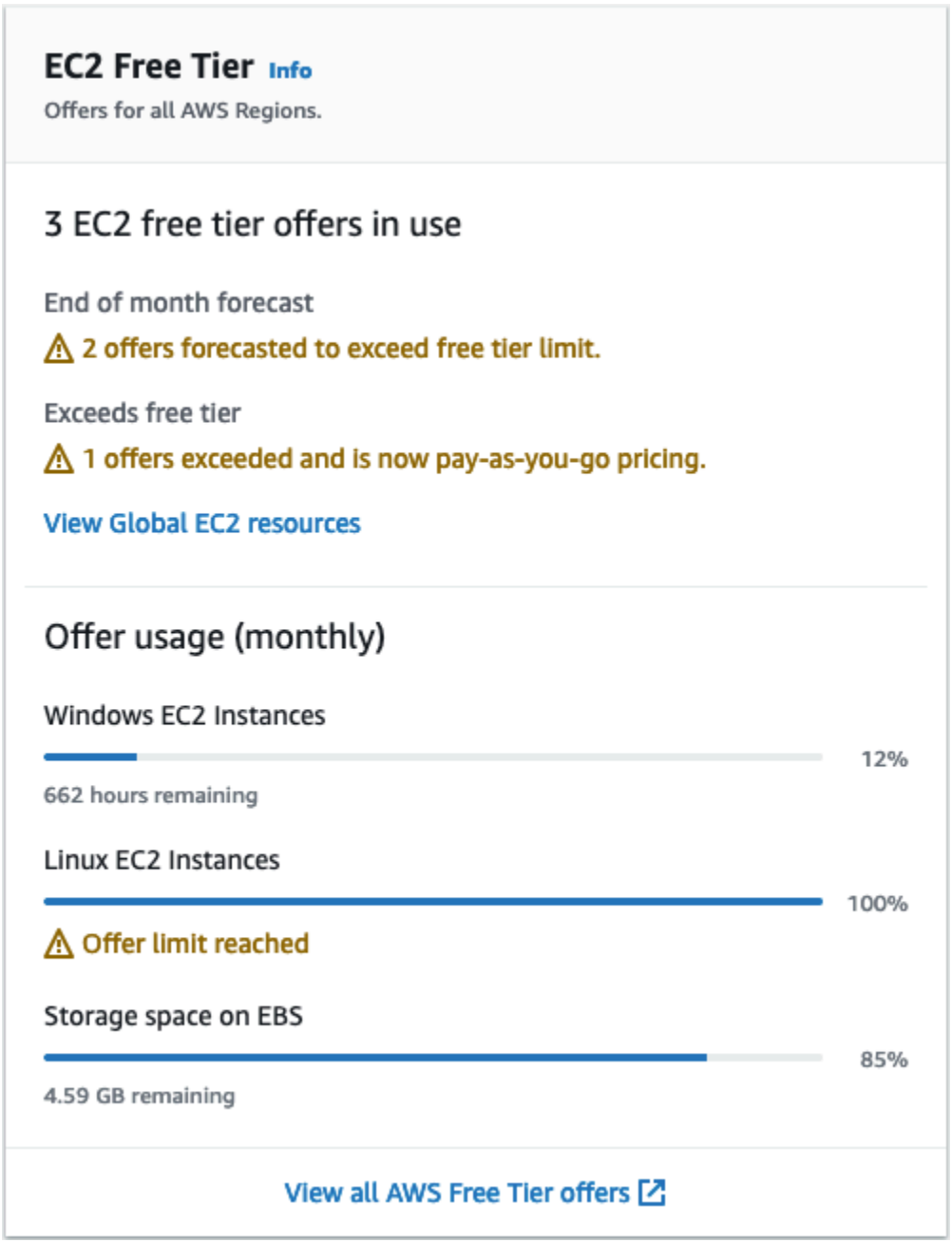
> **Note**
>
> 本節僅適用於 2025 年 7 月 15 AWS 帳戶 日之前建立的免費方案使用者。如果您在 2025 年 7 月 15 日或之後建立帳戶，請參閱 AWS Billing 《使用者指南》中的追蹤您的 AWS 免費方案用量。

如果您在 2025 年 7 月 15 日之前建立帳戶，如果您是 AWS 客戶不到 12 個月且保持在 AWS 免費方案用量限制內，則可以使用 Amazon EC2 而無需支付費用。請務必要追蹤免費方案的用量，以避免出現意外的帳單費用。如果您超過免費方案限制，將會產生標準的依用量計費費用。如需詳細資訊，請參閱AWS 免費方案。

> **Note**
>
> 如果您已成為 AWS 客戶超過 12 個月，則不再符合免費方案用量的資格，而且不會看到下列程序所述的 EC2 免費方案方塊。

追蹤您的免費方案用量

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 EC2 Dashboard (EC2 儀表板)。
3. 尋找 EC2 免費方案方塊 (位於右上角)。



4. 在 EC2 免費方案方塊中，檢查您的免費方案用量，如下所示：

   - 在使用中的 EC2 免費方案優惠下，請注意以下警告：
     - 月底預測 – 這項警告會提醒您，若您繼續目前的用量模式，則本月將會產生費用。
     - 超過免費方案 – 這項警告會提醒您，已超過了免費方案限制且已產生費用。
   - 請根據優惠用量 (每月) 注意您對 Linux 執行個體、Windows 執行個體和 EBS 儲存體的用量。該百分比表示您本月使用了多少免費方案限制。如果您達到 100%，則進一步使用將會產生費用。

> **Note**
>
> 本資訊僅限於在您建立了執行個體之後才會顯示。不過，用量資訊並非即時更新；其更新頻率為一天三次。

5. 為了避免產生進一步的費用，請刪除任何現在產生費用的資源，否則若超過免費方案限制用量，將會產生費用。

   - 如需刪除執行個體的指示，請參閱 終止 Amazon EC2 執行個體。
   - 若要檢查其他區域中是否有可能產生費用的資源，請在 EC2 免費方案方塊中選擇檢視全域 EC2 資源以開啟 EC2 全域檢視。如需詳細資訊，請參閱使用 AWS 全域檢視跨區域檢視資源。

6. 若要檢視 AWS 服務 下所有 的資源用量 AWS 免費方案，請在 EC2 免費方案方塊底部，選擇檢視所有 AWS 免費方案 優惠。如需詳細資訊，請參閱「AWS 帳單使用者指南」中的使用 AWS 免費方案以試用服務。

了解帳單與用量報告中 Amazon EC2 的代碼

當您使用 Amazon EC2 時，我們會在您的 AWS 帳單和用量報告中包含相關代碼。檢閱這些代碼有助於了解 Amazon EC2 的成本與使用模式。追蹤和管理您的支出費用對於成本的最佳化至關重要。

下表說明在帳單和用量報告中顯示的 Amazon EC2 代碼。如需帳單和用量報告中使用的區域代碼清單，請參閱 AWS Region billing codes。

下列項目的帳單代碼：

- 執行個體
- 裸機執行個體
- 專用執行個體
- 專用執行個體
- EBS 最佳化
- Capacity Reservations

相關資源

- the section called “帳單與購買選項”
- the section called “了解 AMI 帳單”

- Amazon EC2 定價

執行個體

Code	Description	單位
region-BoxUsage:*instance-type*	隨需執行個體的執行時間。	小時
region-HostBoxUsage:*instance-type*	專用主機上執行個體的執行時間。	小時
region-SpotUsage:*instance-type*	Spot 執行個體的執行時間。	小時

裸機執行個體

Code	Description	單位
region-BoxUsage:*instance-family*.metal	裸機隨需執行個體的執行時間。	小時
region-HostBoxUsage:*instance-family*.metal	專用主機上裸機執行個體的執行時間。	小時
region-SpotUsage:*instance-family*.metal	裸機 Spot 執行個體的執行時間。	小時

專用執行個體

Code	Description	單位
region-HostUsage :*host-type*	專用主機的佈建時間。	小時
region-ReservedHostUsage :*host-type*	專用主機保留的套用時間。	小時

專用執行個體

Code	Description	單位
region-DedicatedUsage :*instance-type*	專用執行個體的執行時間。	小時 + 每個區域的費用

EBS 最佳化

Code	Description	單位
region-EBSOptimized :*instance-type*	EBS 最佳化的啟用時間。	小時

Capacity Reservations

Code	Description	單位
region-Reservation：*instance-type*	容量保留的預留執行個體時間。	小時
region-UnusedBox：*instance-type*	容量保留的未使用預留執行個體時間。	小時
region-DedicatedRes：*instance-type*	專用容量保留的預留執行個體時間。	小時
region-UnusedDed：*instance-type*	專用容量保留的未使用預留執行個體時間。	小時

建立 EC2 執行個體的庫存

Amazon EC2 提供隨需、可擴展的運算容量。EC2 執行個體是在 AWS 網路中執行的虛擬伺服器。

下表描述了 EC2 執行個體的關鍵特性。系統管理員可運用本指南提供的說明，取得這些特性的相關資訊，並用於配置在內部部署或其他雲端服務供應商網路中執行的功能等效伺服器。

特性	說明
地理位置	Amazon EC2 託管於全球所有 AWS 區域。您可以選取執行個體地點，使其靠近您的客戶。可以在多個地點啟動執行個體。
硬體規格	Amazon EC2 提供各式各樣的最佳化執行個體類型，以滿足不同的使用案例。您為執行個體選取

特性	說明
	的執行個體類型決定其運算、記憶體、聯網和儲存資源。
映像	Amazon Machine Image (AMI) 包含了啟動時要在執行個體上安裝的軟體。其中包括作業系統、軟體套件與自訂組態。
IP 位址與 DNS 主機名稱	執行個體會收到私有 IP 位址與私有 DNS 主機名稱。若您為執行個體設定公有 IP 位址，該執行個體也會收到公有 DNS 主機名稱。
安全群組規則	與執行個體相關聯的安全群組規則會決定允許哪些傳入流量與輸出流量。
使用者資料	使用者資料會在啟動時，提供給執行個體。它包含 shell 指令碼 (Linux) 或 PowerShell 指令碼 (Windows)。

地理位置

Amazon EC2 適用於全球每個 AWS 區域。各個 區域為獨立的地理區域。選擇靠近大多數使用者的區域部署伺服器，可降低網路延遲。

您可使用 Amazon EC2 Global View 列出所有區域的 Amazon EC2 執行個體。使用 AWS 管理主控台和 API，您必須個別列出每個區域的執行個體。

為什麼這很重要

確認執行個體的部署位置後，您可根據需求決定，將功能等效伺服器部署在相同位置或不同位置。

取得全部區域的 EC2 執行個體摘要

1. 前往 https://console.aws.amazon.com/ec2globalview/home 開啟 Amazon EC2 全域檢視主控台。
2. 在區域瀏覽器索引標籤的摘要下，檢查執行個體的資源計數，其中包括執行個體數量與區域數量。選擇底線文字，查看執行個體計數如何分散到各個區域。
3. 在全域搜尋索引標籤上，選取用戶端篩選條件資源類型 = 執行個體。您可以透過指定區域或標籤進一步篩選結果。

使用 取得區域中的 EC2 執行個體數量 AWS CLI

使用下列 describe-instances 命令來計算指定區域的執行個體數量。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "length(Reservations[*].Instances[])"
```

以下為範例輸出。

```
27
```

若要取得執行個體 ID 清單而非執行個體計數，請改用下列 --query 參數。

```
--query "Reservations[*].Instances[].InstanceId"
```

硬體規格

您為 EC2 執行個體指定的執行個體類型，會決定其可用的運算、記憶體、儲存及聯網資源。每個執行個體系列皆會提供不同平衡的運算、記憶體、網路和儲存體資源。如需詳細資訊，請參閱 Amazon EC2 執行個體類型指南。

為什麼這些很重要

確定執行個體的硬體規格後，您可決定功能等效伺服器所需的最低規格。

取得執行個體所用的執行個體類型摘要

使用下面的 describe-instances 命令。sed 命令會移除包含括號的行與空白行。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].InstanceType" | sed 's/[][]//g;/^$/d' | sort |
 uniq -c | sort -nr
```

以下為範例輸出。

```
  20    "c6i.4xlarge",
   5    "t2.micro",
   2    "g6e.2xlarge",
```

取得有關特定稽核執行個體的資訊

若要描述執行個體類型，請使用下列 describe-instance-types 命令。--query 參數僅包含輸出中的指定欄位。這些屬於基本特性。您可加入所需的任何額外欄位，或參閱《Amazon EC2 執行個體類型指南》中的 Amazon EC2 執行個體類型規格。

```
aws ec2 describe-instance-types \
    --instance-types c6i.4xlarge t2.micro g6e.2xlarge \
    --query "InstanceTypes[*].
[InstanceType,VCpuInfo.DefaultVCpus,MemoryInfo.SizeInMiB,NetworkInfo.NetworkPerformance]"
 \
    --output table
```

以下為範例輸出。這些欄位包括執行個體類型、vCPU、記憶體 (GiB) 和網路頻寬 (Gbps)。

```
-------------------------------------------------------
|                DescribeInstanceTypes                |
+--------------+-----+--------+----------------------+
|  t2.micro    |  1  |  1024  |  Low to Moderate     |
|  c6i.4xlarge |  16 |  32768 |  Up to 12.5 Gigabit  |
|  g6e.2xlarge |  8  |  65536 |  Up to 20 Gigabit    |
+--------------+-----+--------+----------------------+
```

映像

Amazon Machine Image (AMI) 提供啟動時安裝在執行個體上的軟體，例如作業系統、服務、開發工具及應用程式。您可將 AMI 匯出為其他格式，以便用於其他伺服器。

為什麼這些很重要

確認您執行個體所使用的 AMI 後，即可規劃功能等效伺服器所需的映像。您可以將 AMI 匯出為可在其他位置使用的格式。或者，您可能需要連線到從每個 AMI 啟動的執行個體，確認已安裝的內容及其設定方式，並確保您擁有所需的一切複本。

取得執行個體 AMI

使用下面的 describe-instances 命令。sed 命令會移除包含括號的行與空白行。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
```

```
    --query "Reservations[*].Instances[].ImageId" | sed 's/[][]//g;/^$/d' | sort | uniq
 -c | sort -nr
```

以下為範例輸出。

```
20     "ami-0a70b9d193ae8a79",
 5     "ami-07d9cf938edb0739b",
 2     "ami-09245d5773578a1d6",
```

取得特定 AMI 的相關資訊

使用下面的 describe-images 命令。--query 參數僅包含輸出中的指定欄位。您可以包含所需的任何其他欄位。

```
aws ec2 describe-images \
    --image-id ami-0a70b9d193ae8a79 ami-07d9cf938edb0739b ami-09245d5773578a1d6 \
    --query "Images[*].
{ID:ImageId,CreateDate:CreationDate,Name:Name,Description:Description}"
```

以下為範例輸出。

```
[
    {
        "ID": "ami-0a70b9d193ae8a799",
        "CreateDate": "2025-03-01T02:22:41.000Z",
        "Name": "web-server-3.1",
        "Description": "Image for web servers"
    },
    {
        "ID": "ami-07d9cf938edb0739b",
        "CreateDate": "2025-02-01T23:59:03.000Z",
        "Name": "awesome-application-11.5",
        "Description": "Image for Awesome Application"
    },
    {
        "ID": "ami-09245d5773578a1d6",
        "CreateDate": "2025-01-31T02:22:41.000Z",
        "Name": "monitoring-4.2",
        "Description": "Monitoring software"
    }
]
```

從 AMI 建立 VM 檔案

使用 export-image 命令。支援的影像格式為 VHD (與 Citrix Xen 和 Microsoft Hyper-V 相容)、VMDK (與 VMware ESX 和 VMware vSphere 相容) 以及原始格式 (與 KVM 和 Xen Hypervisor 相容)。如需 VM Import/Export 需求與限制的相關資訊，請參閱 VM Import/Export 需求。

IP 位址與 DNS 主機名稱

您的使用者會使用其公有 DNS 主機名稱，透過網際網路連線至 EC2 執行個體。EC2 執行個體的公有 DNS 主機名稱會解析至其公有 IP 位址。EC2 執行個體可透過其私有 IP 位址互相通訊。

為什麼這些很重要

IP 位址使裝置能夠透過區域網路或網際網路相互通訊。記錄您執行個體的 IP 位址與 DNS 名稱，可協助確保功能等效伺服器能與您的執行個體存取相同的用戶端或裝置。例如，您可更新負載平衡器組態，或為 DNS 供應商建立的 DNS 記錄。

取得執行個體的 IP 位址和 DNS 主機名稱

使用下面的 describe-instances 命令。您必須在擁有執行個體的每個區域中執行此命令。`--query` 參數僅包含輸出中的指定欄位。您可以包含所需的任何其他欄位。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].[InstanceId,PrivateIpAddress,PublicDnsName]" \
    --output table
```

以下為範例輸出。這些欄是執行個體 ID、私有 IPv4 位址與公有 IPv4 DNS 名稱。

```
----------------------------------------------------------------------------------
|                                DescribeInstances
 |
+---------------------+------------+-------------------------------------------------
+
| i-0bac57d7472c89bac | 10.0.2.139 | ec2-192-0-2-142.us-east-2.compute.amazonaws.com
 |
| i-0fa8b7678975a3fff | 10.0.14.85 | ec2-198-51-100-56.us-east-2.compute.amazonaws.com
 |
  ...
  ...
| i-08fd74f3f1595fdbd | 10.0.1.241 | ec2-203.0.113.13.us-east-2.compute.amazonaws.com
 |
```

```
+--------------------+-----------+-----------------------------------------------------
+
```

安全群組規則

安全群組會做為虛擬防火牆。將安全群組與 EC2 執行個體關聯後，其規則允許 EC2 執行個體透過特定連接埠與通訊協定的傳入及傳出流量。

為什麼這些很重要

確認允許抵達您伺服器的傳入流量與允許離開伺服器的輸出流量後，即可規劃功能等效伺服器所需的防火牆規則。

取得執行個體的安全群組

使用下面的 describe-instances 命令。sed 命令會移除包含括號的行與空白行。您必須在擁有執行個體的每個區域中執行此命令。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].SecurityGroups[].GroupId" | sed 's/[][]//g;/^
$/d' | sort | uniq -c | sort -nr
```

以下為範例輸出。

```
27     "sg-01dd3383691d02f42",
10     "sg-08c77cc910c0b3b2c",
 2     "sg-00f4e409629f1a42d",
```

取得安全群組的傳入規則

使用如下 describe-security-group-rules 命令，顯示指定安全群組的規則，其中 `IsEgress` 為 `false`。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-01dd3383691d02f42 \
    --query 'SecurityGroupRules[?IsEgress==`false`]'
```

以下為範例輸出。第一個規則是預設傳入規則，允許來自指派給此安全群組的所有資源的傳入流量。另一個規則允許來自指定字首清單中 IP 位址的 SSH 流量。要描述字首清單中的 CIDR 區塊，請使用 describe-prefix-lists 命令。

```
[
    {
        "SecurityGroupRuleId": "sgr-06c8b42574a91db1b",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "ReferencedGroupInfo": {
            "GroupId": "sg-01dd3383691d02f42",
            "UserId": "123456789012"
        },
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-west-2:123456789012:security-group-
rule/sgr-06c8b42574a91db1b"
    },
    {
        "SecurityGroupRuleId": "sgr-0886a5d46afcd1758",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "PrefixListId": "pl-f8a6439125e7bf465",
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-east-2:123456789012:security-group-
rule/sgr-0886a5d46afcd1758"
    }
]
```

取得安全群組的傳出規則

使用如下 describe-security-group-rules 命令，顯示指定安全群組的規則，其中 IsEgress 為 true。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-01dd3383691d02f42 \
    --query 'SecurityGroupRules[?IsEgress==`true`]'
```

以下為範例輸出。它僅包含預設傳出規則，允許所有傳出 IPv4 流量。

```
[
```

```
    {
        "SecurityGroupRuleId": "sgr-048f09a719247dce7",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": true,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "CidrIpv4": "0.0.0.0/0",
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-east-2:123456789012:security-group-
rule/sgr-048f09a719247dce7"
    }
]
```

使用者資料

當您啟動 EC2 執行個體時，可藉助使用者資料將 shell 指令碼傳遞至執行個體。請注意，使用者資料是 base64 編碼，因此您需要解碼使用者資料才能讀取指令碼。

為什麼這很重要

若您在啟動時執行命令，做為設定執行個體的一部分，則可能需要在設定功能等效伺服器時執行相同的任務。

檢視執行個體的解碼後使用者資料

使用下列 describe-instance-attribute 命令。base64 命令會對使用者資料進行解碼。

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute userData \
    --output text \
    --query "UserData.Value" | base64 --decode
```

以下為範例輸出。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

相關資源

以下是 EC2 執行個體的其他特性：

- 金鑰對
- 儲存
- Tags (標籤)

您可確認您是否使用下列項目來啟動 EC2 執行個體，或在 EC2 執行個體之間分配流量：

- Auto Scaling 群組
- EC2 機群
- Elastic Load Balancing

針對 Amazon EC2 執行個體的問題進行疑難排解

以下過程和提示可幫助您解決 Amazon EC2 執行個體的問題。

問題

- 對 Amazon EC2 執行個體啟動問題進行疑難排解
- 對 Amazon EC2 執行個體停止問題進行疑難排解
- 對 Amazon EC2 執行個體終止問題進行疑難排解
- 對無法連線的 Amazon EC2 執行個體進行疑難排解
- 對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解
- 狀態檢查失敗的疑難排解 Amazon EC2 Linux 執行個體
- 從錯誤磁碟區對 Amazon EC2 Linux 執行個體開機進行疑難排解
- 對連線至 Amazon EC2 Windows 執行個體的問題進行疑難排解
- 對 Amazon EC2 Windows 執行個體啟動問題進行疑難排解
- 針對 Amazon EC2 Windows 執行個體的問題進行疑難排解
- 透過網路對 Windows 執行個體進行核心除錯
- 重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼
- 針對 Amazon EC2 Windows 執行個體的 Sysprep 問題進行疑難排解
- 使用 EC2Rescue 疑難排解受損的 Amazon EC2 Linux 執行個體
- 使用 EC2Rescue 疑難排解受損的 Amazon EC2 Windows 執行個體
- EC2 Serial Console for 執行個體
- 傳送診斷中斷以偵錯無法連線的 Amazon EC2 執行個體

對 Amazon EC2 執行個體啟動問題進行疑難排解

以下是疑難排解秘訣，可協助您在啟動 Amazon EC2 執行個體時解決問題。

啟動問題

- 無效裝置名稱
- 超過執行個體限制

- 執行個體容量不足
- 目前並不支援要求的組態。請查看文件以了解支援的組態。
- 立刻終止執行個體
- 許可不足
- Windows 啟動後立即出現高 CPU 用量 (僅限 Windows 執行個體)
- 啟動IMDSv1-enabled的執行個體失敗

無效裝置名稱

說明

當您嘗試啟動新的執行個體時，發生 `Invalid device name` *`device_name`* 錯誤。

原因

如果您在嘗試啟動執行個體時收到此錯誤，表示在請求中為一個或多個磁碟區指定的裝置名稱具有無效的裝置名稱。可能的原因包括：

- 選取的 AMI 可能正在使用裝置名稱。
- 裝置名稱可能會保留給根磁碟區。
- 裝置名稱可能會用於請求中的其他磁碟區。
- 裝置名稱可能對作業系統無效。

解決方案

要解決問題：

- 請確保您選取的 AMI 中未使用裝置名稱。執行下列命令以檢視 AMI 使用的裝置名稱。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890 --query
 'Images[*].BlockDeviceMappings[].DeviceName'
```

- 請確保您沒有使用為根磁碟區保留的裝置名稱。如需詳細資訊，請參閱可用裝置名稱。
- 請確保請求中指定的每個磁碟區都有唯一的裝置名稱。
- 請確保您指定的裝置名稱格式正確。如需詳細資訊，請參閱可用裝置名稱。

超過執行個體限制

說明

當您嘗試啟動新的執行個體或重新啟動已停止的執行個體時，發生 `InstanceLimitExceeded` 錯誤。

原因

當您嘗試啓動新的執行個體或重新啟動已停止的執行個體時，如果發生 `InstanceLimitExceeded` 錯誤，就表示已達到您在區域中可啟動的執行個體數目限制。當您建立 AWS 帳戶時，我們會針對每個區域可執行的執行個體數量設定預設限制。

解決方案

您可以要求提高每一區域的執行個體限制。如需詳細資訊，請參閱 Amazon EC2 服務配額。

執行個體容量不足

說明

當您嘗試啟動新的執行個體或重新啟動已停止的執行個體時，發生 `InsufficientInstanceCapacity` 錯誤。

原因

當您嘗試啓動執行個體或重新啟動已停止的執行個體時，如果發生此錯誤，就表示 AWS 目前可用的隨需容量不足，無法滿足您的請求。

解決方案

若要解決問題，請嘗試下列方法：

- 等候幾分鐘，然後再次提交您的請求；容量會頻繁轉移。
- 以降低的執行個體數提交新請求。例如，如果您要提出一次啟動 15 個執行個體的請求，請改嘗試提出 3 個啟動 5 個執行個體的請求，或 15 個啟動 1 個執行個體的請求。
- 如果啟動執行個體，請提交新的請求，而不要指定可用區域。
- 如果啟動執行個體，請使用不同的執行個體類型 (您可以在後面的階段調整大小) 來提交新的請求。如需詳細資訊，請參閱 Amazon EC2 執行個體類型變更。
- 如果透過叢集置放群組來啟動執行個體，可能會出現容量不足的錯誤。

目前並不支援要求的組態。請查看文件以了解支援的組態。

說明

當您嘗試啟動新的執行個體時，由於不支援執行個體組態，因此發生 Unsupported 錯誤。

原因

錯誤訊息會提供其他詳細資訊。例如，指定的區域或可用區域可能不支援執行個體類型或執行個體購買選項。

解決方案

請嘗試不同的執行個體組態。若要搜尋符合您需求的執行個體類型，請參閱 尋找 Amazon EC2 執行個體類型。

立刻終止執行個體

說明

您的執行個體會從 pending 狀態進入 terminated 狀態。

原因

下列是執行個體為什麼可能會立即終止的幾個原因：

- 您已超過 EBS 磁碟區限制。如需詳細資訊，請參閱 Amazon EC2 執行個體的 Amazon EBS 磁碟區限制。
- EBS 快照已損毀。
- 根 EBS 磁碟區已加密，且您沒有存取 KMS 金鑰進行解密的許可。
- AMI 的區塊型設備映射中指定的快照已加密，且您沒有存取 KMS 金鑰進行解密的許可，或您無法存取 KMS 金鑰來加密還原的磁碟區。
- 用來啟動執行個體的 Amazon S3 支援的 AMI 缺少必要部分 (an image.part.xx 檔案)。

如需詳細資訊，請使用下列其中一種方法來查明終止原因。

使用 Amazon EC2 主控台查明終止原因

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取該執行個體。
3. 在第一個索引標籤上，找出狀態轉換原因旁的原因。

使用 取得終止原因 AWS CLI

1. 使用 describe-instances 命令並指定執行個體 ID。

```
aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

2. 檢閱命令傳回的 JSON 回應，並記下 StateReason 回應元素中的值。

   下列程式碼區塊顯示 StateReason 回應元素的範例。

```
"StateReason": {
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",
  "Code": "Server.InternalError"
},
```

使用 取得終止原因 AWS CloudTrail

如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的 使用 CloudTrail 事件歷史記錄檢視事件。

解決方案

根據終止原因，採取下列其中一項動作：

- **`Client.VolumeLimitExceeded: Volume limit exceeded`** — 刪除未使用的磁碟區。您可以提交申請來增加磁碟區容量上限。
- **`Client.InternalError: Client error on launch`** — 確保您擁有存取 AWS KMS keys 用於解密和加密磁碟區之 所需的許可。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 在 AWS KMS中使用金鑰政策。

許可不足

說明

當您嘗試啟動新的執行個體時發生 `"errorMessage": "You are not authorized to perform this operation."` 錯誤，並且啟動失敗。

原因

如果在嘗試啟動執行個體時發生此錯誤，表示您沒有啟動執行個體所需的 IAM 許可。

可能缺少的許可包括：

- `ec2:RunInstances`
- `iam:PassRole`

可能還缺少其他許可。如需啟動執行個體所需的許可清單，請參閱 範例：使用 EC2 啟動執行個體精靈 和 啟動執行個體 (RunInstances) 下方的 IAM 政策範例。

解決方案

要解決問題：

- 如果您以 IAM 使用者身分提出請求，請確認您具有以下許可：
  - `ec2:RunInstances` 使用萬用字元資源 (「*」)
  - `iam:PassRole` 與資源符合角色 ARN (例如，`arn:aws:iam::999999999999:role/ExampleRoleName`)
- 如果沒有上述許可，請編輯與 IAM 角色或使用者關聯的 IAM 政策，以新增缺少的必要許可。

如果您的問題尚未解決，且繼續收到啟動失敗錯誤，可以解碼錯誤中包含的授權失敗訊息。解碼的訊息包含 IAM 政策中缺少的許可。如需詳細資訊，請參閱在 EC2 執行個體啟動期間收到 "UnauthorizedOperation" 錯誤後，如何解碼授權失敗訊息？

Windows 啟動後立即出現高 CPU 用量 (僅限 Windows 執行個體)

> **Note**
>
> 此疑難排解秘訣僅適用於 Windows 執行個體。

如果 Windows Update 設為 Check for updates but let me choose whether to download and install them (檢查是否有更新，但讓我選擇是否下載並安裝) (預設的執行個體設定)，此項檢查會使用執行個體 50 - 99% 的 CPU。如果此 CPU 耗用造成您應用程式的問題，您可在 Control Panel (控制面板) 手動變更 Windows Update 設定，或在 Amazon EC2 user data (Amazon EC2 使用者資料) 欄位中使用下列指令碼：

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v
 AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

當您執行此指令碼時，請指定 /d 的值。預設值為 3。可能的值包括以下：

1. 絕不檢查更新
2. 檢查是否有更新，但讓我選擇是否下載並安裝
3. 下載更新，但讓我選擇是否安裝
4. 自動安裝更新

您修改執行個體的使用者資料之後，就可以執行它。如需詳細資訊，請參閱啟動時，在您的 Windows 執行個體上執行命令。

啟動IMDSv1-enabled的執行個體失敗

說明

您會收到以下訊息的UnsupportedOperation例外狀況：

```
You can't launch instances with IMDSv1 because httpTokensEnforced
is enabled for this account. Either launch the instance with
httpTokens=required or contact your account owner to disable
httpTokensEnforced using the ModifyInstanceMetadataDefaults API or the
account settings in the EC2 console.
```

原因

當您嘗試在 EC2 帳戶設定或 AWS 組織宣告政策強制使用 IMDSv2 () 的帳戶中啟動啟用 IMDSv1 () 的新執行個體時，就會擲出此錯誤`httpTokensEnforced = enabled`。`httpTokens = optional` IMDSv2

解決方案

如果您準備僅使用 IMDSv2，請在停用 IMDSv1 的情況下啟動執行個體 (`httpTokens = required`)。若要檢查您是否已準備好，請參閱 轉換為使用 執行個體中繼資料服務第 2 版。

如果您仍然需要對新的或現有的執行個體支援 IMDSv1，則需要停用 區域中帳戶的 IMDSv2 強制執行。若要停用 IMDSv2 強制執行，請將 `HttpTokensEnforced`設定為 `disabled`。如需詳細資訊，

請參閱《Amazon EC2 API 參考》中的 ModifyInstanceMetadataDefaults。如果您偏好使用 主控台設定此設定，請參閱 在帳戶層級強制執行 IMDSv2。

對 Amazon EC2 執行個體停止問題進行疑難排解

如果 Amazon EBS 後端執行個體凍結在 `stopping` 狀態，這可能是底層主機電腦的問題。

若要解決該問題，請遵循這些步驟：

1. 強制停止執行個體

   使用 Amazon EC2 主控台或 AWS CLI 強制停止執行個體。如需相關步驟，請參閱 強制停止執行個體。

   執行個體將先嘗試正常關閉，包括清空檔案系統快取和中繼資料 (儘管您可選擇忽略正常關閉)。如果無法在逾時期間內完成正常關閉，執行個體會強制關閉，且不會清空檔案系統快取和中繼資料。

2. 強制停止後

   執行檔案系統檢查和修復程序。

   > **Important**
   >
   > 執行這些程序十分重要，因為強制停止可避免清空檔案系統快取和中繼資料。

3. 如果強制停止失敗

   如果 10 分鐘後執行個體仍未停止，請執行以下動作：

   a. 在 AWS re:Post 上發佈協助請求。請在訊息中註明執行個體 ID，並說明您已採取的步驟，以利加速解決問題。
   b. 或者，如果您已購買支援方案，可到 Support 中心建立技術支援案例。
   c. 在等待協助時，您可視需要建立替代執行個體。如需相關步驟，請參閱 (選用) 建立替代執行個體。

當執行個體處於 `stopping` 狀態或 `running` 以外的任何狀態時，將不會收取執行個體使用費。當執行個體處於 `running` 狀態時，您只需支付執行個體使用費。

目錄

- 強制停止執行個體
- (選用) 建立替代執行個體

強制停止執行個體

可強制停止執行個體。如果經過 10 分鐘後，執行個體仍未停止，請在 AWS re:Post 上發布請求協助的訊息。請在訊息中註明執行個體 ID，並說明您已採取的步驟，以利加速解決問題。或者，如果您已購買支援方案，可到 Support 中心建立技術支援案例。

> **Note**
>
> 只有在執行個體僅處於 `stopping` 狀態時，才可以使用主控台強制停止執行個體。只有在執行個體僅處於 `pending`、`running` 或 `stopping` 狀態時，才可以使用 AWS CLI強制停止執行個體。

Console

若要強制停止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取凍結的執行個體。
3. 依次選擇執行個體狀態、強制停止執行個體。

   請注意，唯有在您的執行個體處於 `stopping` 狀態的情況下，才能在主控台使用 Force stop instance (強制停止執行個體)。如果您的執行個體處於其他狀態 (`shutting-down`和 除外`terminated`)，您可以使用 AWS CLI 強制停止執行個體。
4. (選用) 若要在強制停止期間跳過正常的作業系統關閉，請選取跳過作業系統關閉核取方塊。
5. 選擇強制停止。

AWS CLI

若要強制停止執行個體

搭配使用 stop-instances 命令與 `--force` 選項。

```
aws ec2 stop-instances \
```

```
    --instance-ids i-1234567890abcdef0 \
    --force
```

若要在強制停止期間略過正常的作業系統關閉，請包含 `--skip-os-shutdown` 選項。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

若要強制停止執行個體

使用 Stop-EC2Instance cmdlet 並將 `-Enforce` 設為 `true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

若要在強制停止期間略過正常的作業系統關閉，請包含 `-SkipOsShutdown $true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

(選用) 建立替代執行個體

在您等待來自 AWS re:Post 或支援中心協助的同時，可視需求建立替代執行個體。從已凍結執行個體建立 AMI，然後使用新的 AMI 來啟動新的執行個體。

> **Important**
>
> 如果凍結的執行個體僅產生系統狀態檢查，您可以建立替代執行個體，這是因為執行個體狀態檢查將導致 AMI 複製到故障作業系統的確切複本上。在確認狀態訊息後，建立 AMI 並使用新的 AMI 來啟動新執行個體。

Console

若要建立替代執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取凍結的執行個體。
3. 選擇 Actions (動作)、Image and templates (映像和範本)、Create image (建立映像)。
4. 在「Create image (建立映像)」頁面，執行下列動作：

   a. 輸入 AMI 的名稱與說明。

   b. 清除重新開機執行個體。

   c. 選擇建立映像。

   如需詳細資訊，請參閱the section called “從執行個體建立 AMI”。
5. 從 AMI 啟動新的執行個體，並確認新的執行個體可正常運作。
6. 選取凍結的執行個體，然後依序選擇動作、執行個體狀態、終止 (刪除) 執行個體。如果該執行個體在終止時也出現凍結的狀況，Amazon EC2 會自動在幾小時內強制其終止。

如果無法如先前步驟中的說明，從執行個體建立 AMI，您可以依照下列方式來建立替換執行個體：

(替代選項) 使用主控台來建立替換執行個體

1. 選取執行個體，然後依序選擇 Description (說明)、Block devices (區塊型儲存設備)。選取每個磁碟區，並記下其磁碟區 ID。請務必註明哪一個磁碟區是根磁碟區。
2. 在導覽窗格中，選擇 Volumes (磁碟區)。針對執行個體選取每個磁碟區，然後依序選擇 Actions (動作)、Create Snapshot (建立快照)。
3. 在導覽窗格中，選擇 Snapshots (快照)。選擇您剛建立的快照，然後依序選擇 Actions (動作)、Create Volume (建立磁碟區)。
4. 使用與凍結執行個體相同的作業系統來啟動執行個體。請寫下其根磁碟區的磁碟區 ID 和裝置名稱。
5. 在導覽窗格中，選擇「Instances (執行個體)」，選取剛才啟動的執行個體，然後依序選擇「Instance state (執行個體狀態)」、「Stop instance (停止執行個體)」。
6. 在導覽窗格中選擇 Volumes (磁碟區)，選取已停止執行個體的根磁碟區，然後依序選擇 Actions (動作)、Detach Volume (分離磁碟區)。

7. 選取從凍結執行個體建立的根磁碟區，然後依序選擇「Actions (動作)」、「Attach Volume (連結磁碟區)」，接著將該磁碟區連結到新的執行個體，並做為其根磁碟區 (使用您之前記下的裝置名稱)。將其他所有非根磁碟區的磁碟區，連結到執行個體。
8. 在導覽窗格中，選擇 Instances (執行個體)，然後選取替換個體。選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。確認執行個體可正常運作。
9. 選取凍結的執行個體，然後依序選擇執行個體狀態、終止 (刪除) 執行個體。如果該執行個體在終止時也出現凍結的狀況，Amazon EC2 會自動在幾小時內強制其終止。

AWS CLI

若要建立替代執行個體

1. 搭配使用 create-image 命令與 `--no-reboot` 選項，從停滯的執行個體中建立 AMI。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-replacement-ami" \
    --description "AMI for replacement instance" \
    --no-reboot
```

2. 使用 run-instances 命令，從剛建立的 AMI 中啟動新執行個體。
3. 確認新的執行個體可正常運作。
4. 使用 terminate-instances 命令來終止停滯的執行個體。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

若要建立替代執行個體

1. 使用 New-EC2Image cmdlet，從停滯的執行個體中建立 AMI，並將 `-NoReboot` 設定為 `true`。

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-replacement-ami" `
    -Description "AMI for replacement instance" `
    -NoReboot $true
```

2. 使用 New-EC2Instance cmdlet，從剛建立的 AMI 中啟動新執行個體。
3. 確認新的執行個體可正常運作。
4. (選用) 使用 Remove-EC2Instance cmdlet，終止停滯的執行個體。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

對 Amazon EC2 執行個體終止問題進行疑難排解

關閉或刪除執行個體即稱為執行個體終止。下列資訊可協助您對終止執行個體之問題進行疑難排解。

當執行個體未處於 `running` 狀態時，將不會向您收取任何執行個體使用費。換句話說，當您終止執行個體時，只要其狀態變更為 `shutting-down`，即刻起就會停止收取該執行個體所產生的費用。

立刻終止執行個體

若干問題可能會導致執行個體在啟動時立即終止。如需詳細資訊，請參閱「立刻終止執行個體」。

延遲的執行個體終止

如果執行個體保持 `shutting-down` 狀態超過幾分鐘，原因可能如下：

- 執行個體正在執行關機指令碼。
- 基礎主機電腦發生問題。

如果執行個體保持 `shutting-down` 狀態數小時，Amazon EC2 會將其視為已停滯，並強制加以終止。

若要自行解決已停滯的執行個體：

1. 強制終止執行個體

   使用 Amazon EC2 主控台或 AWS CLI 強制終止執行個體。如需相關步驟，請參閱 強制終止執行個體。

   執行個體將先嘗試正常關閉，包括清空檔案系統快取和中繼資料 (儘管您可選擇忽略正常關閉)。如果無法在逾時期間內完成正常關閉，執行個體會強制關閉，且不會清空檔案系統快取和中繼資料。
2. 如果強制終止失敗

如果執行個體在數小時後仍未終止，且似乎終止停滯，請執行下列動作：

a. 在 AWS re:Post 上發佈協助請求。請在訊息中註明執行個體 ID，並說明您已採取的步驟，以利加速解決問題。

b. 或者，如果您已購買支援方案，可到 Support 中心建立技術支援案例。

強制終止執行個體

如果執行個體在終止時出現停滯，您可以強制終止執行個體。如果執行個體在數小時後仍未終止，請向 AWS re:Post 發佈協助請求。請在訊息中註明執行個體 ID，並說明您已採取的步驟，以利加速解決問題。或者，如果您已購買支援方案，可到 Support 中心建立技術支援案例。

Console

若要強制終止執行個體

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取凍結的執行個體。
3. 依序選擇執行個體狀態、強制終止執行個體。

   請注意，唯有執行個體處於 `stopping` 狀態的情況下，才能在主控台使用強制終止執行個體。如果您的執行個體處於其他狀態 (`shutting-down`和 除外`terminated`)，您可以使用 AWS CLI 強制終止您的執行個體。
4. (選用) 若要在強制終止期間跳過正常的作業系統關閉，請選取略過作業系統關機核取方塊。
5. 選擇強制終止。

AWS CLI

若要強制終止執行個體

搭配使用 terminate-instances 命令與 `--force` 選項。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

若要在強制終止期間略過正常的作業系統關閉，請包含 `--skip-os-shutdown` 選項。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

若要強制終止執行個體

使用 Remove-EC2Instance cmdlet 並將 -Enforce 設為 true。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

若要在強制終止期間略過正常的作業系統關閉，請包含 -SkipOsShutdown $true。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

仍顯示已終止的執行個體

在執行個體終止後，仍會短暫顯示，然後才會遭到刪除。狀態會顯示為 terminated。如果經過好幾個小時仍未刪除此項目，請聯絡 Support 中心。

錯誤：執行個體可能無法終止。修改其 'disableApiTermination' 執行個體屬性

如果您嘗試終止執行個體並收到 The instance *i-1234567890abcdef0* may not be terminated. Modify its 'disableApiTermination' instance attribute 錯誤訊息，則表示執行個體已啟用終止保護。終止保護可防止執行個體被意外終止。

您必須先停用終止保護，才能終止執行個體。

如需詳細資訊，請參閱變更執行個體終止保護。

執行個體自動啟動或終止

以下行為通常表示您使用了 Amazon EC2 Auto Scaling、EC2 Fleet 或 Spot Fleet，以根據您所定義的條件來自動擴展運算資源。

- 您終止某個執行個體，且新執行個體自動啟動。
- 您啟動某個執行個體，且其中一個執行個體自動終止。
- 您停止某個執行個體，該執行個體終止，且新執行個體自動啟動。

若要停止自動擴展，請尋找 Auto Scaling 群組或正在啟動執行個體的機群，並將其容量設定為 0 或刪除。

對無法連線的 Amazon EC2 執行個體進行疑難排解

以下資訊可協助您對無法連線的 Amazon EC2 執行個體進行疑難排解。您可擷取螢幕截去畫面或存取主控台輸出資料，以協助診斷問題，並判斷是否應該重新啟動執行個體。針對無法連線的 Windows 執行個體，請檢視服務傳回的螢幕擷取畫面以進行疑難排解。

目錄

- 重新啟動執行個體
- 執行個體主控台輸出
- 擷取無法連線執行個體的螢幕擷取畫面
- 針對無法連線之 Windows 執行個體進行疑難排解的常見螢幕擷取畫面
- 當主機電腦故障時的執行個體復原
- 執行個體離線且意外重新啟動

重新啟動執行個體

對於故障診斷和一般的執行個體管理，能夠重新啟動無法連線的執行個體很重要。

如同您可以按下重設鈕來重設電腦，您也可以使用 Amazon EC2 主控台、CLI 或 API，來重設 EC2 執行個體。如需詳細資訊，請參閱重新啟動您的 Amazon EC2 執行個體。

執行個體主控台輸出

主控台輸出是診斷問題的一項寶貴工具。在針對核心問題和服務組態問題進行故障診斷時 (這些問題可能會造成執行個體終止，或是在其 SSH 協助程式能夠啟動之前較無法連線)，這項工具尤其實用。

- Linux 執行個體 – 執行個體主控台輸出所顯示的資料，就是在連接到電腦的實體顯示器上，通常會顯示的主控台輸出資料。主控台輸出會傳回經過緩衝的資訊，這些資訊會在執行個體轉換狀態 (啟動、停止、重新啟動和終止) 之後不久發布。已發布的輸出不會連續更新，只有在可能具有最高價值時才會更新。
- Windows 執行個體 – 執行個體主控台輸出會包含最近三筆系統事件日誌錯誤。

只有執行個體才能存取主控台輸出資料。

您可在執行個體生命週期內，擷取最新的序列主控台輸出。僅 Nitro 型執行個體支援此選項。

Console

無法取得主控台輸出

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 Instances (執行個體)。
3. 選取執行個體。
4. 選擇 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)、Get system log (取得系統日誌)。

AWS CLI

無法取得主控台輸出

使用 get-console-output 命令。

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

PowerShell

無法取得主控台輸出

使用 Get-EC2ConsoleOutput cmdlet。

```
Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0
```

擷取無法連線執行個體的螢幕擷取畫面

如果您無法連線至執行個體，您可擷取執行個體的螢幕擷取畫面，將其當成影像檢視。此影像可讓您看到執行個體的狀態，有利於更快進行故障診斷。

您可以在執行個體執行中或執行個體當機後產生螢幕擷取畫面。產生的影像格式為 JPG ，大小不超過 100 kb。螢幕擷取畫面不收取資料傳輸費用。

限制

下列項目不支援此功能：

- 裸機執行個體 (`*.metal` 類型的執行個體)
- 執行個體正在使用 NVIDIA GRID 驅動程式
- 採用 ARM 型 Graviton 處理器的執行個體
- 上的 Windows 執行個體 AWS Outposts
- AWS Local Zones 上的 Windows 執行個體

區域支援

此功能在以下區域不可用：

- 亞太區域 (泰國)
- 墨西哥 (中部)
- GovCloud 區域

Console

取得執行個體的螢幕擷取畫面

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在左側導覽窗格中選擇 (執行個體)。
3. 選取要擷取的執行個體。
4. 選擇 動作、監視和故障診斷、取得執行個體螢幕截圖。

5. 選擇下載，或開啟影像的內容（按一下滑鼠右鍵）選單以下載並儲存。

AWS CLI

擷取執行個體的螢幕擷取畫面

使用 get-console-screenshot 命令。輸出內容為 base64 編碼。

```
aws ec2 get-console-screenshot --instance-id i-1234567890abcdef0
```

PowerShell

擷取執行個體的螢幕擷取畫面

使用 Get-EC2ConsoleScreenshot cmdlet。輸出內容為 base64 編碼。

```
Get-EC2ConsoleScreenshot -InstanceId i-1234567890abcdef0
```

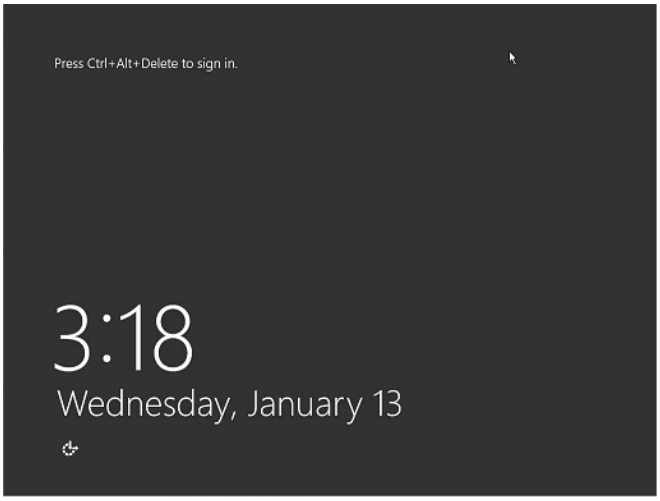
針對無法連線之 Windows 執行個體進行疑難排解的常見螢幕擷取畫面

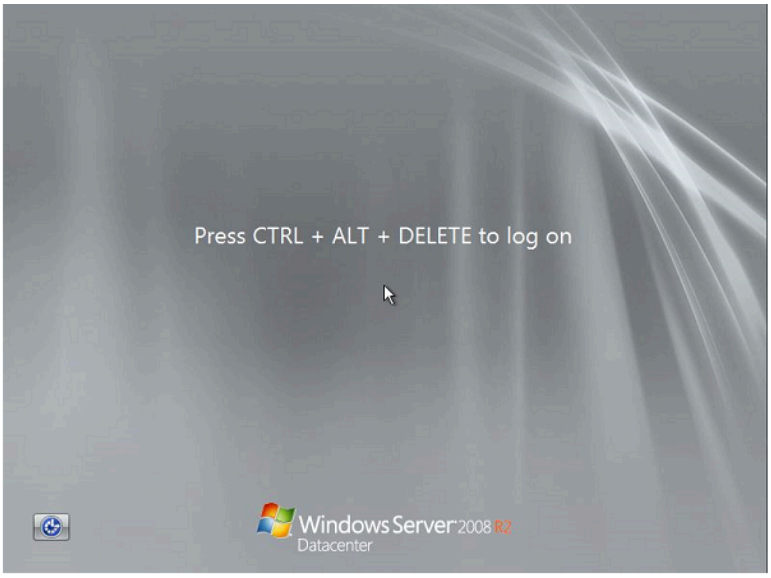
您可根據服務傳回的螢幕截取畫面，使用下列資訊協助您故障診斷無法連線的 Windows 執行個體。

- 登入畫面 (Ctrl+Alt+Delete)
- 復原主控台螢幕
- Windows 開機管理程式畫面
- Sysprep 畫面
- 就緒畫面
- Windows Update 畫面
- Chkdsk

登入畫面 (Ctrl+Alt+Delete)

主控台螢幕截取畫面服務傳回下列內容。





如果在登入期間無法連線某個執行個體，可能是您的網路組態或 Windows 遠端桌面服務發生問題。如有程序正在佔用大量的 CPU，執行個體也可能不回應。

網路組態

使用以下資訊來確認您的、 AWS Microsoft Windows 和本機 （ 或內部部署 ） 網路組態未封鎖對執行個體的存取。

AWS 網路組態

組態	確認
安全群組組態	確認連接埠 3389 已開放給您的安全群組。確認您連線到正確的公有 IP 地址。如果執行個體與彈性 IP 地址無關聯，則公有 IP 地址會在執行個體停止/啟動後變更。如需詳細資訊，請參閱遠端桌面無法連線到遠端電腦。
VPC 組態 (網路 ACL)	確認您 Amazon VPC 的存取控制清單 (ACL) 未封鎖存取。如需相關資訊，請參閱「Amazon VPC 使用者指南」中的網路 ACL。
VPN 組態	如果您使用虛擬私有網路 (VPN) 連線至您的 VPC，請確認 VPN 通道的連線能力。如需詳細資訊，請參閱故障診斷 AWS Client VPN：對 VPC 的通道連線問題。

Windows 網路組態

組態	確認
Windows 防火牆	確認 Windows 防火牆未封鎖對您執行個體的連線。如遠端桌面故障診斷一節遠端桌面無法連線到遠端電腦第 7 項所述，停用 Windows 防火牆。
進階 TCP/IP 組態 (使用靜態 IP)	因為您設定了靜態 IP 地址，所以執行個體可能不回應。若為 VPC，請建立網路介面並連接至執行個體。

本機或內部部署的網路組態

確認本機網路組態未封鎖存取。嘗試連線到與無法連線之執行個體同一 VPC 的另一個執行個體。如果您無法存取另一個執行個體，請與您的本機網路管理員合作，判斷本機政策是否限制存取。

遠端桌面服務問題

如果在登入期間無法連線執行個體，可能是執行個體上的遠端桌面服務 (RDS) 發生問題。

> **Tip**
>
> 您可以使用 `AWSSupport-TroubleshootRDP` 執行手冊，檢查和修改可能影響遠端桌面通訊協定 (RDP) 連線的各種設定。如需詳細資訊，請參閱《AWS Systems Manager Automation 執行手冊參考》中的 `AWSSupport-TroubleshootRDP`。

遠端桌面服務組態

組態	確認
RDS 執行中	確認 RDS 是否正在執行個體上執行。使用 Microsoft Management Console (MMC) 服務嵌入式管理單元連線至執行個體 (`services.msc`)。在服務清單中，確認 Remote Desktop Services (遠端桌面服務) 為 Running (執行中)。如果不是，請啟動它，然後將啟動類型設為 Automatic (自動)。如果使用服務嵌入式管理單元無法連線至執行個體，請分離根磁碟區與執行個體、

組態	確認
	拍攝磁碟區快照或從磁碟區建立 AMI、將原始磁碟區連接到同一可用區域的另一個執行個體做為輔助磁碟區，然後修改 Start 登錄機碼。完成後，重新將根磁碟區連接到原始執行個體。
RDS 已啟用	即使服務已啟動，仍可能停用。分離根磁碟區與執行個體、拍攝磁碟區快照或從磁碟區建立 AMI、將原始磁碟區連接到相同可用區域的另一個執行個體做為輔助磁碟區，然後如 使用遠端登錄在 EC2 執行個體上啟用遠端桌面 所述，修改 Terminal Server 登錄機碼來啟用服務。 完成後，重新將根磁碟區連接到原始執行個體。

高 CPU 使用率

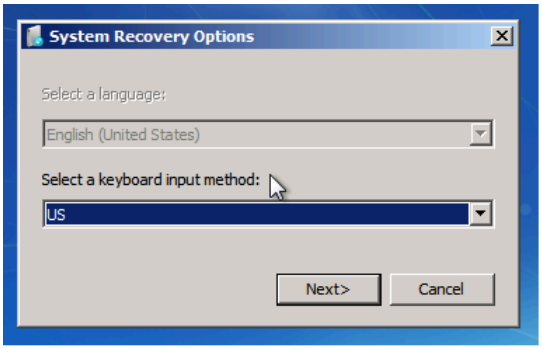
使用 Amazon CloudWatch 檢查您執行個體上的 CPUUtilization (Maximum) 指標。如果 CPUUtilization (Maximum) 的值很高，請等 CPU 下降再嘗試連線。CPU 使用率高的可能的原因如下：

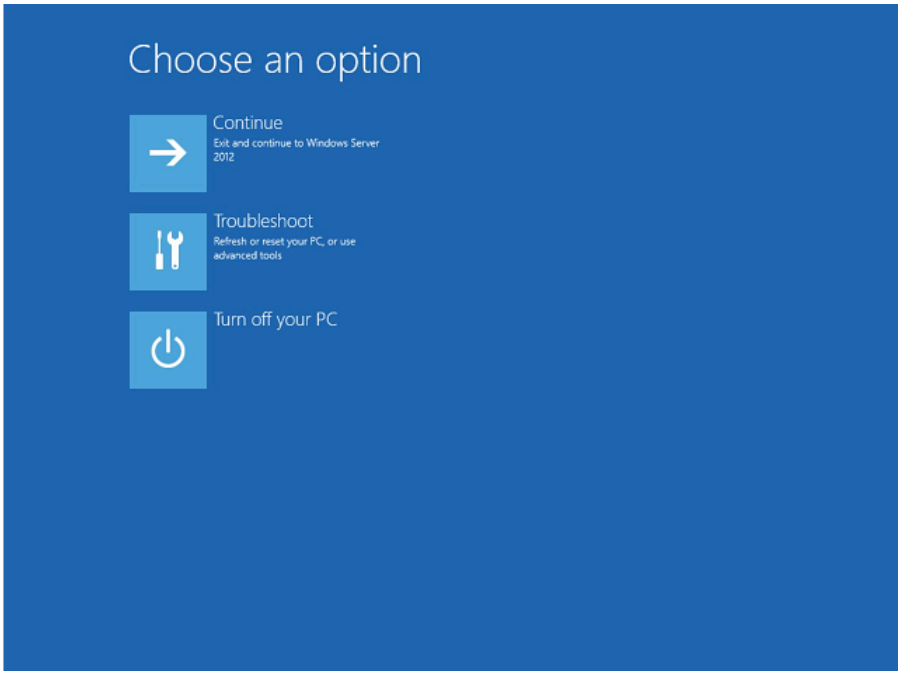
- Windows 更新
- 安全軟體掃描
- 自訂啟動指令碼
- 任務排程

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的 取得特定資源的統計資料。如需其他的故障診斷秘訣，請參閱 Windows 啟動後立即出現高 CPU 用量 (僅限 Windows 執行個體)。

復原主控台螢幕

主控台螢幕截取畫面服務傳回下列內容。





作業系統可能在復原主控台中開機，然後卡在這個狀態，如果 bootstatuspolicy 不是設為 ignoreallfailures。請使用下列程序將 bootstatuspolicy 組態變更為 ignoreallfailures。

根據預設， 提供的公有 Windows AMIs 政策組態 AWS 會設為 ignoreallfailures。

1. 停止無法連線的執行個體。
2. 建立根磁碟區的快照。根磁碟區連接至此執行個體做為 /dev/sda1。

   分離根磁碟區與無法連線的執行個體、拍攝磁碟區快照或從磁碟區建立 AMI，然後將它連接到同一可用區域的另一個執行個體做為輔助磁碟區。

   > **Warning**
   >
   > 如果您的暫時執行個體與原始執行個體使用相同的 AMI 啟動，則您必須完成額外的步驟，否則您無法在還原原始執行個體根磁碟區之後將其啟動，原因是磁碟簽名衝突。如果您必須是使用同一 AMI 建立暫時執行個體，為避免發生磁碟簽章衝突，請完成 磁碟簽章衝突 中的步驟。
   >
   > 或者，請為暫時執行個體選擇不同的 AMI。例如，如果原始執行個體使用適用於 Windows Server 2016 的 AMI，則使用適用於 Windows Server 2019 的 AMI 啟動暫時執行個體。

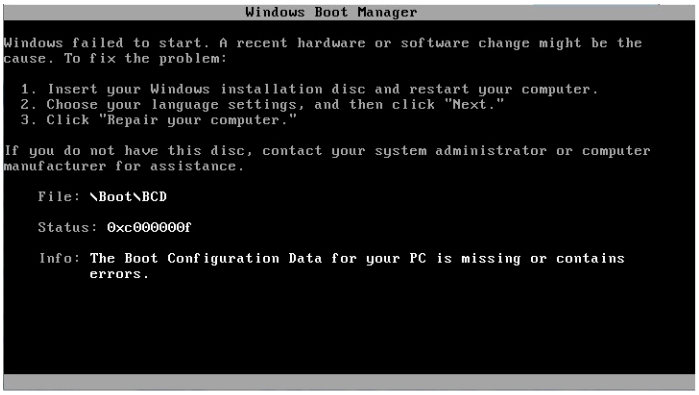
3. 登入執行個體，從命令提示執行下列命令以將 bootstatuspolicy 組態變更為 ignoreallfailures。

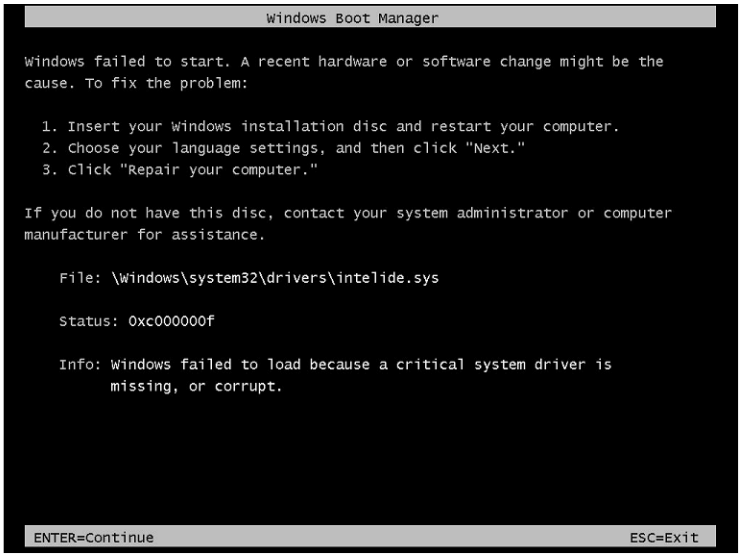
```
bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
 ignoreallfailures
```

4. 將磁碟區重新連接至無法連線的執行個體，然後再次啟動執行個體。

Windows 開機管理程式畫面

主控台螢幕截取畫面服務傳回下列內容。

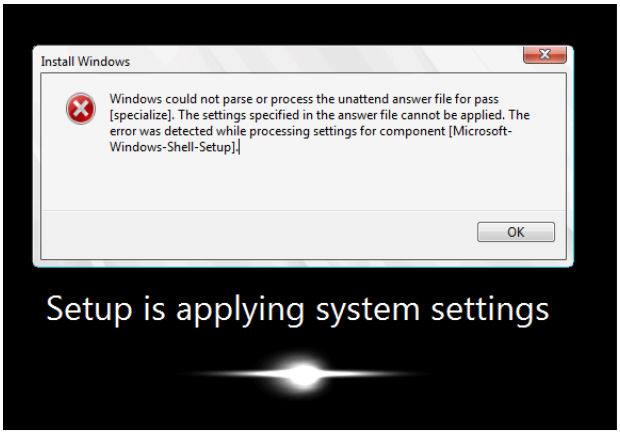


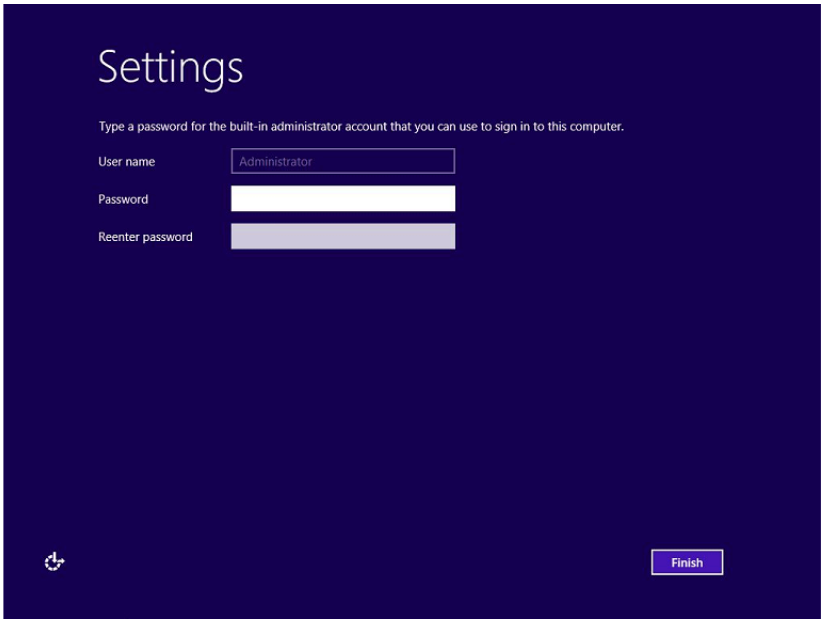


作業系統的系統檔案及/或登錄發生重大損毀。當執行個體卡在這個狀態時，您應該從最近的備份 AMI 復原執行個體或啟動替代執行個體。如果您需要存取此執行個體的資料，請分離所有根磁碟區與無法連線的執行個體、拍攝這些磁碟區的快照或從這些磁碟區建立 AMI，然後將它們連接到同一可用區域的另一個執行個體做為輔助磁碟區。

Sysprep 畫面

主控台螢幕截取畫面服務傳回下列內容。

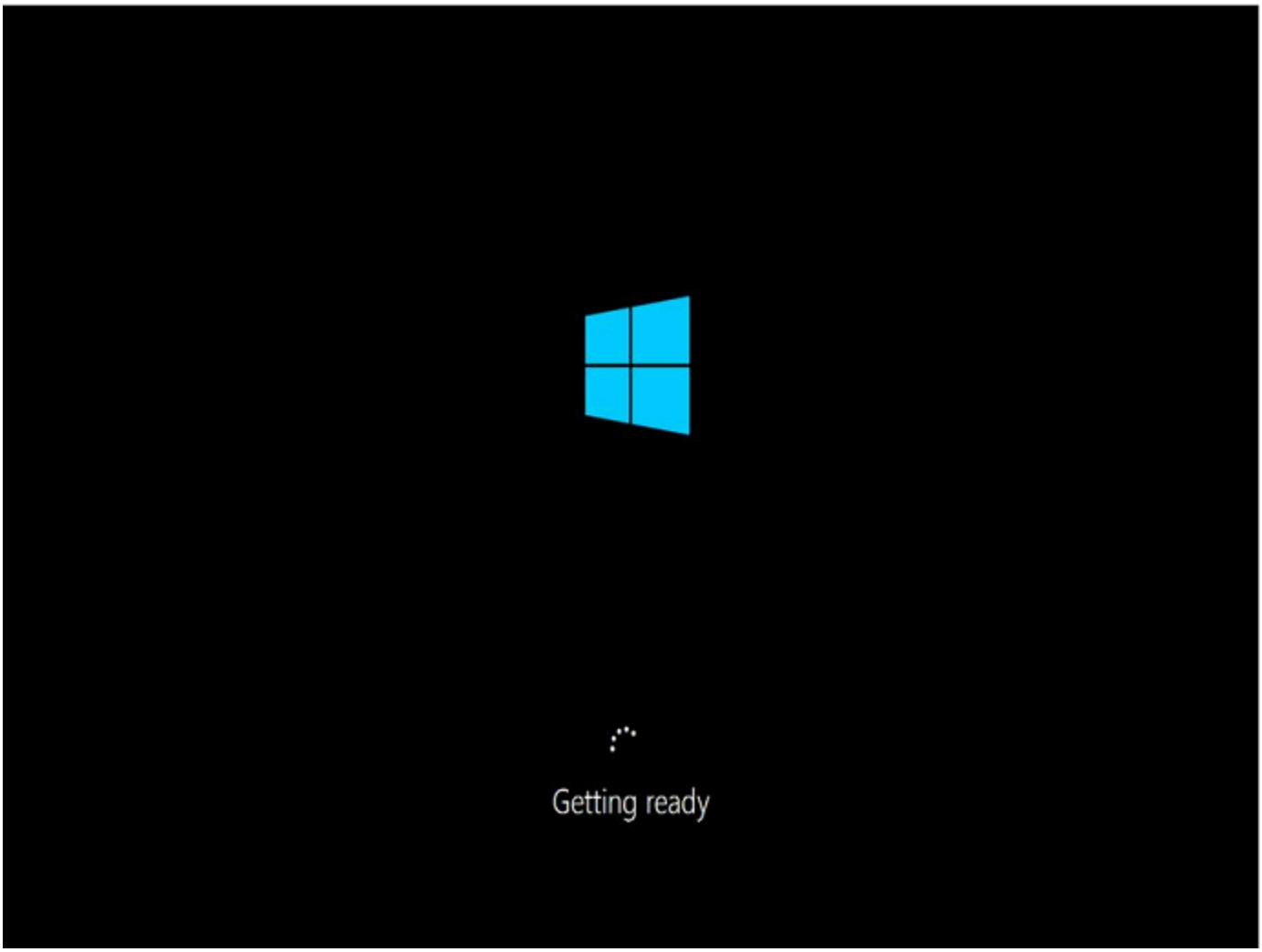




如果您未使用 EC2Config 服務呼叫 sysprep 或如果作業系統在執行 Sysprep 時失敗，您可能會看到這個畫面。您可以使用 EC2Rescue 來重設密碼。否則，請參閱 使用 Windows Sysprep 建立 Amazon EC2 AMI。

就緒畫面

主控台螢幕截取畫面服務傳回下列內容。



重複更新執行個體主控台螢幕擷取畫面服務的畫面，以確認進度環是否旋轉。如果進度環不停旋轉，請等候作業系統啟動。您也可以使用 Amazon CloudWatch 檢查您執行個體上的 CPUUtilization (Maximum) (CPUUtilization (上限)) 指標，查看作業系統是否作用中。如果進度環不旋轉，則執行個體可能卡在開機程序。重新啟動執行個體。如果重新開機不能解決此問題，請從最近的備份 AMI 復原執行個體或啟動替代執行個體。如果您需要存取此執行個體上的資料，請分離根磁碟區與無法連線的執行個體、拍攝磁碟區快照或從磁碟區建立 AMI。然後將它連接到同一可用區域的另一個執行個體做為輔助磁碟區。

Windows Update 畫面

主控台螢幕截取畫面服務傳回下列內容。

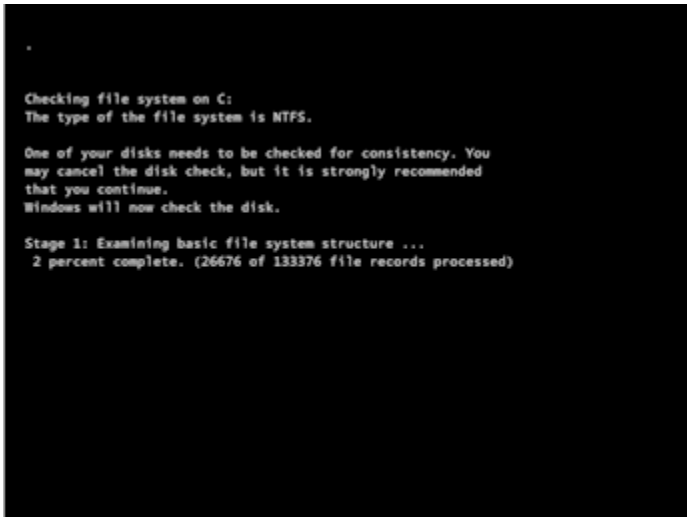


Windows Update 程序正在更新登錄。等候更新完成。不要重新開機或停止執行個體，因為這會在更新期間造成資料損毀。

> **Note**
>
> Windows Update 程序在更新期間會使用伺服器的資源。如果您經常碰到此問題，請考慮使用更快的執行個體類型和更快的 EBS 磁碟區。

Chkdsk

主控台螢幕截取畫面服務傳回下列內容。



Windows 對磁碟機使用 chkdsk 系統工具，以確認檔案系統完整性並修復邏輯檔案系統錯誤。請等候程序完成。

當主機電腦故障時的執行個體復原

如果底層主機電腦的硬體具有無法修復的問題， AWS 可能會排程執行執行個體停止事件。您會事先收到此類事件的電子郵件通知。

在主機電腦故障時復原其上執行的 Amazon EBS 後端執行個體

1. 將執行個體存放磁碟區上的所有重要資料備份到 Amazon EBS 或 Amazon S3。
2. 停止執行個體。
3. 啟動實例。
4. 復原所有重要的資料。

如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。

若要復原在發生故障的主機電上運行執行個體儲存體根磁碟區的執行個體

1. 從執行個體建立 AMI。
2. 將映像上傳到 Amazon S3。
3. 將重要的資料備份到 Amazon EBS 或 Amazon S3。
4. 終止執行個體。
5. 從 AMI 啟動新的執行個體。
6. 將所有重要的資料還原到新的執行個體上。

執行個體離線且意外重新啟動

如果執行個體似乎已離線，然後意外重新啟動，則可能已進行自動執行個體復原。當 AWS 偵測到執行個體因基礎硬體或軟體問題而無法使用，且在執行個體上啟用簡化的自動復原或 CloudWatch 動作型復原時，就會發生這種情況。

在復原程序期間， 會 AWS 遷移到不同的硬體，嘗試還原執行個體的可用性。若要驗證執行個體是否發生自動執行個體復原，請參閱 驗證是否發生自動執行個體復原。

> **Note**
>
> 如果工作負載或應用程式沒有回應，請檢查它是否在執行個體上執行。如果不是，請手動啟動。為了防止未來發生此問題，請實作復原計劃，以確保工作負載或應用程式在執行個體復原後正常運作。

對連線至 Amazon EC2 Linux 執行個體的問題進行疑難排解

下列資訊和常見錯誤可協助您疑難排解連線到執行個體的問題。

連線問題

- 連線問題的常見原因
- 連線到執行個體時發生錯誤：連線時間逾時
- 錯誤：無法載入金鑰 ... 預期為：任一私有金鑰
- 錯誤：伺服器無法辨識使用者金鑰
- 錯誤：許可遭拒或 [執行個體] 連接埠 22 已關閉連線
- 錯誤：未受保護的私有金鑰檔案
- 錯誤：私有金鑰的開頭必須為「-----BEGIN RSA PRIVATE KEY-----」，結尾為「-----END RSA PRIVATE KEY-----」
- 錯誤：主機金鑰驗證失敗
- 錯誤：伺服器拒絕了我們的金鑰，或是沒有可用的支援驗證方法
- 無法對執行個體執行 Ping 動作
- 錯誤：伺服器意外關閉網路連線
- 錯誤：EC2 Instance Connect 的主機金鑰驗證失敗
- 無法使用 EC2 Instance Connect 連線至 Ubuntu 執行個體
- 我的私有金鑰遺失。如何連線到我的執行個體？

連線問題的常見原因

建議您透過確認已正確執行下列任務來開始疑難排解執行個體的連線問題。

確認執行個體的使用者名稱

使用您的使用者帳戶的使用者名稱或用於啟動執行個體的 AMI 預設使用者名稱，可連線至執行個體。

- 取得使用者帳戶的使用者名稱。

  如需如何建立使用者帳戶的詳細資訊，請參閱在您的 Amazon EC2 Linux 執行個體上管理系統使用者。

- 取得您用於啟動執行個體的 AMI 的預設使用者名稱。

用於啟動執行個體的 AMI	預設使用者名稱
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	檢查 AMI 提供者

確認您的安全性群組規則允許流量

請確保與執行個體關聯的安全群組允許來自您的 IP 地址的傳入 SSH 流量。VPC 的預設安全群組根據預設並不允許傳入 SSH 流量。啟動執行個體精靈建立的安全群組根據預設會啟用 SSH 流量。如

需將傳入 SSH 流量規則新增至 Linux 執行個體的步驟，請參閱 從您電腦連線到執行個體的規則。如需驗證步驟，請參閱 連線到執行個體時發生錯誤：連線時間逾時。

確認您的執行個體已就緒

啟動執行個體之後，執行個體可能需要幾分鐘的時間準備就緒，以接受連接要求。檢查您的執行個體，以確保它正在執行並通過其狀態檢查。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 請確認下列內容：

   a. 在 Instance state (執行個體狀態) 欄中，驗證您的執行個體處於 `running` 狀態。
   b. 在狀態檢查欄中，驗證您的執行個體已通過全部狀態檢查。

確認已滿足連線的所有先決條件

確保您備妥連線所需的所有資訊。如需詳細資訊，請參閱使用 SSH 連線至您的 Linux 執行個體。

從 Linux 或 macOS X 進行連線

如果本機電腦作業系統是 Linux 或 macOS X，請檢查特定先決條件的以下項目，以便連線至 Linux 執行個體：

- SSH 用戶端
- EC2 執行個體連線
- AWS Systems Manager 工作階段管理員

從 Windows 進行連線

如果您的本機電腦作業系統是 Windows，請檢查以下連線選項的特定先決條件，以便連線至 Linux 執行個體：

- OpenSSH
- PuTTY
- AWS Systems Manager 工作階段管理員
- 適用於 Linux 的 Windows 子系統

檢查執行個體是否為受管執行個體

不允許使用者啟動連線至受管執行個體。若要判斷執行個體是否受管，請尋找執行個體的受管欄位。若該值為 true，則為受管執行個體。如需詳細資訊，請參閱Amazon EC2 受管執行個體。

連線到執行個體時發生錯誤：連線時間逾時

如果您試著連線至執行個體，並收到錯誤訊息 `Network error: Connection timed out` 或 `Error connecting to [instance], reason: -> Connection timed out: connect`，則請嘗試下列動作：

檢查安全群組規則。

您需要安全群組規則，來允許透過適當的連接埠，從本機電腦公有 IPv4 地址傳入流量。

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 在主控台頁面底部的 Security (安全) 標籤上，檢查 Inbound rules (傳入規則) 下方所選執行個體已生效的規則清單。請確認具備允許從本機電腦到連接埠 22 (SSH) 的流量規則。

   如果安全群組不具有規則來允許從本機電腦傳入流量，請在安全群組中新增規則。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。
4. 針對允許傳入流量的規則，請檢查 Source (來源) 欄位。如果該值是單一 IP 地址，而且 IP 地址不是靜態的，則每次重新啟動電腦時都會指派新的 IP 地址。這將導致規則不會包括您電腦的 IP 地址流量。如果您的電腦位於企業網路中，或者如果您是透過網際網路服務供應商 (ISP) 連線，或者如果您的電腦 IP 地址是動態地址並且在每次重新啟動電腦時都會變更，則 IP 地址可能不是靜態地址。若要確保您的安全群組規則允許從本機電腦傳入流量，請指定用戶端電腦使用的 IP 地址範圍，而不是為 Source (來源) 指定單一 IP 地址。

   如需安全群組規則的詳細資訊，請參閱 Amazon VPC 使用者指南 中的 安全群組規則。

查看子網的路由表。

您需要一個路由路徑，來將目的地位於 VPC 之外的所有流量，傳送到 VPC 的網際網路閘道。

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 在 Networking (網路) 標籤上，請記下 VPC ID 和 Subnet ID (子網 ID) 的值。
4. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
5. 在導覽窗格中，選擇 Internet Gateways (網際網路閘道)。請確認已經有網際網路閘道連結到 VPC。否則請選擇 Create internet gateway (建立網際網路閘道)、輸入網際網路閘道的名稱，然後選擇 Create internet gateway (建立網際網路閘道)。然後，針對您建立的網際網路閘道，選取

Actions (動作)、Attach to VPC (連接至 VPC)、選取您的 VPC，然後選取 Attach internet gateway (連接網際網路閘道)，以將網際網路閘道連接至您的 VPC。

6. 在導覽窗格中，選取 Subnets (子網)，然後選取子網。
7. 在 Route Table (路由表) 索引標籤中，確認有包含 `0.0.0.0/0` 的路由做為目的地，以及有 VPC 的網際網路閘道做為目標。如果您是使用其 IPv6 地址來連線到執行個體，請確認有針對所有 IPv6 流量 (`::/0`) 的路由，來指向網際網路閘道。若否，請執行下列操作：

   a. 選擇路由表 ID (rtb-xxxxxxxx) 以導覽至路由表。
   b. 在 Routes (路由) 標籤中，選擇 Edit routes (編輯路由)。選擇 Add route (新增路由)，使用 `0.0.0.0/0` 做為目的地，並以網際網路閘道為目標。若是 IPv6，請選擇 Add route (新增路由)，使用 `::/0` 做為目的地，並以網際網路閘道為目標。
   c. 選擇 Save routes (儲存路由)。

查看子網的網路存取控制清單 (ACL)。

網路 ACL 必須允許透過通訊埠 22 上您的本機 IP 位址來傳入 SSH 流量。它也必須允許傳送到暫時連接埠 (1024-65535) 的傳出流量。

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中，選擇 Subnets (子網)。
3. 選取您的子網。
4. 在 Network ACL (網路 ACL) 索引標籤中，對於 Inbound rules (傳入規則)，請確認這些規則允許透過所需的連接埠，從您的電腦傳入流量。否則，請刪除或修改會阻止流量的規則。
5. 對於 Outbound Rules (傳出規則)，確認這些規則允許透過暫時連接埠將流量傳出到電腦。否則，請刪除或修改會阻止流量的規則。

如果您的電腦位於企業網路中

請詢問網路管理員，了解內部的防火牆是否允許透過連接埠 22 將流量傳入至您的電腦，或從您的電腦傳出。

如果您的電腦上具有防火牆，請確認此防火牆是否允許透過連接埠 22 號將流量傳入至您的電腦，或從您的電腦傳出。

確認執行個體是否具備公有 IPv4 地址。

如果沒有，您可建立彈性 IP 地址與您執行個體的關聯。如需詳細資訊，請參閱彈性 IP 位址。

檢查執行個體上的 CPU 負載，伺服器可能會過載。

AWS 會自動提供資料，例如 Amazon CloudWatch 指標和執行個體狀態，您可以用來查看執行個體上有多少 CPU 負載，並視需要調整負載的處理方式。如需詳細資訊，請參閱使用 CloudWatch 監控您的執行個體。

- 如果負載會變動不定，您可以利用 Auto Scaling 和 Elastic Load Balancing 功能，來自動擴展或縮減執行個體的規模。
- 如果負載穩定增加，您可以改用較大容量的執行個體類型。如需詳細資訊，請參閱Amazon EC2 執行個體類型變更。

若要使用 IPv6 地址來連線到執行個體，請檢察下列項目：

- 與子網具有關聯的路由表，必須包含讓 IPv6 流量 (`::/0`) 傳到網際網路閘道的路由路徑。
- 安全群組規則必須允許從本機的 IPv6 地址，透過連接埠 22 傳入流量。
- 網路 ACL 必須允許外傳和傳入的 IPv6 流量。
- 如果從舊版的 AMI 啟動執行個體，則執行個體可能未針對 DHCPv6 設定 (在網路介面上不會自動辨識 IPv6 地址)。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的在執行個體上設定 IPv6。
- 您的本機電腦必須擁有 IPv6 地址，且必須設定以使用 IPv6。

錯誤：無法載入金鑰 ... 預期為：任一私有金鑰

如果您嘗試連接到您的執行個體並收到錯誤訊息 `unable to load key ... Expecting: ANY PRIVATE KEY`，則表示存放私有金鑰的檔案設定不正確。如果私有金鑰檔案以 `.pem` 結尾，則表示可能仍設定不正確。私有金鑰檔案設定錯誤的可能原因是憑證遺失。

如果私有金鑰檔案設定不正確，請依照以下步驟解決錯誤

1. 建立新的金鑰對。如需詳細資訊，請參閱 使用 Amazon EC2 建立金鑰對。

   > **Note**
   >
   > 或者，您也可以使用第三方工具來建立新的金鑰對。如需詳細資訊，請參閱 使用第三方工具來建立金鑰對，然後將公有金鑰匯入 Amazon EC2。

2. 將新的金鑰對新增到執行個體。如需詳細資訊，請參閱我的私有金鑰遺失。如何連線到我的執行個體？。

3. 使用新的金鑰對連線至執行個體。

錯誤：伺服器無法辨識使用者金鑰

若如果使用 SSH 連線到執行個體

- 使用 `ssh -vvv`，在連線時取得三重 verbose (詳細模式) 的除錯資訊：

```
ssh -vvv -i path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

下列的輸出範例，示範了在嘗試使用伺服器無法辨識的金鑰來連線到執行個體時，可能會顯示的訊息：

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-
interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
```

```
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

如果使用 PuTTY 連線到您的執行個體

- 確認私有金鑰 (.pem) 檔案是否已轉換為 PuTTY (.ppk) 可以辨識的格式。如需有關轉換私有金鑰的詳細資訊，請參閱使用 PuTTY 連線至您的 Linux 執行個體。

> **Note**
>
> 在 PuTTYgen 中，載入私有金鑰，然後選取 Save Private Key (儲存私有金鑰)，而非 Generate (產生)。

- 驗證您在連線時已使用適當的 AMI 使用者名稱。在 PuTTY Configuration 視窗的 Host name 中，輸入使用者名稱。

用於啟動執行個體的 AMI	預設使用者名稱
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`

用於啟動執行個體的 AMI	預設使用者名稱
Rocky Linux	`rocky`
其他	檢查 AMI 提供者

- 確認已建立傳入安全群組的規則，以允許從適當的連接埠傳入流量。如需詳細資訊，請參閱從您電腦連線到執行個體的規則。

錯誤：許可遭拒或 [執行個體] 連接埠 22 已關閉連線

若使用 SSH 連線至執行個體，並出現下列任一錯誤：`Host key not found in [directory]`、`Permission denied (publickey)`、`Authentication failed, permission denied` 或 `Connection closed by [instance] port 22`，請確認您在連線時使用了適當的 AMI 使用者名稱，且已為執行個體指定正確的私有金鑰 (`.pem`) 檔案。

適合的使用者名稱如下：

用於啟動執行個體的 AMI	預設使用者名稱
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`

用於啟動執行個體的 AMI	預設使用者名稱
Rocky Linux	`rocky`
其他	檢查 AMI 提供者

例如，若要使用 SSH 用戶端來連接至 Amazon Linux 執行個體，請使用下列命令：

```
ssh -i /path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

請確認在啟動執行個體時，所使用的私有金鑰檔案可對應到您所選取的金鑰對。

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 在 Details (詳細資訊) 標籤上，驗證 Instance details (執行個體詳細資訊) 下方中 Key pair name (金鑰對名稱) 的值。
4. 如果在啟動執行個體時，您並未指定金鑰對，可以終止該執行個體並啟動新的執行個體，來確保指定金鑰對。如果這是一直在使用的執行個體，但是您已經沒有金鑰對的 .pem 檔案，可以用新的金鑰對來取代舊的。如需詳細資訊，請參閱我的私有金鑰遺失。如何連線到我的執行個體？。

如果產生自己的金鑰對，請確定金鑰產生器已經過設定，可產生 RSA 金鑰。DSA 金鑰是不被接受的。

如果收到 `Permission denied (publickey)` 錯誤，而上述的所有情況皆不適用 (例如，先前可以連線)，可能是對執行個體家目錄的權限已變更。對 /home/*instance-user-name*/.ssh/authorized_keys 的權限必須限定為只指派給擁有者。

在執行個體上確認權限

1. 停止執行個體並分離根磁碟區。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。
2. 在目前執行個體所在的同一個可用區域中，啟動暫時執行個體 (使用與目前執行個體所用 AMI 類似的或相同的 AMI)，然後將根磁碟區連結到暫時執行個體。
3. 連線到暫時執行個體、建立掛載點，然後掛載已連結的磁碟區。
4. 從暫時執行個體，檢查已連結磁碟區 /home/*instance-user-name*/ 目錄的權限。必要時，請依照下列方式來調整權限：

```
[ec2-user ~]$ chmod 600 mount_point/home/instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name
```

5. 取消掛載磁碟區、將該磁碟區從暫時執行個體分離，然後再重新連結到原始執行個體。請務必為根磁碟區指定正確的裝置名稱，例如 /dev/xvda。
6. 啟動執行個體。如果不再需要暫時執行個體，您可以將其終止。

錯誤：未受保護的私有金鑰檔案

私有金鑰檔案必須受到保護，不讓其他任何使用者讀取或寫入。如果私有金鑰可以讓除了您以外的任何人讀取或寫入，SSH 會忽略您的金鑰，而且會出現下列的警告訊息。

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

如果在您試著登入執行個體時，出現類似的訊息，請檢驗該錯誤訊息的第一行，來確認是否針對執行個體使用了正確的公有金鑰。上列的範例使用私有金鑰 .ssh/my_private_key.pem，此金鑰具有 0777 的檔案權限，可讓所有人讀取或寫入此檔案。這種權限等級非常地不安全，因此 SSH 會忽略此金鑰。

如果從 macOS 或 Linux 連線，請執行下列命令來修正此錯誤，替換私有金鑰檔案的路徑。

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

如果從 Windows 連線至 Linux 執行個體，請在您的本機電腦上執行下列步驟。

1. 導覽至 .pem 檔案。
2. 開啟 .pem 檔案的內容（按一下滑鼠右鍵）選單，然後選取屬性。

3. 選擇 Security (安全) 標籤。
4. 選取 Advanced (進階)。
5. 請確認您是檔案的擁有者。如果不是，請將擁有者變更為您的使用者名稱。
6. 選取 Disable inheritance (停用繼承) 和 Remove all inherited permissions from this object (移除此物件的所有繼承許可)。
7. 選取 Add (新增)、Select a principal (選取委託人)，輸入您的使用者名稱，然後選取 OK (確定)。
8. 從 Permission Entry (許可項目) 視窗中，授予 Read (讀取) 許可，然後選取 OK (確定)。
9. 按一下 Apply (套用)，以確保儲存所有設定。
10. 選取 OK (確定) 關閉 Advanced Security Settings (進階安全設定) 視窗。
11. 選取 OK (確定) 關閉 Properties (屬性) 視窗。
12. 您應能使用 SSH 從 Windows 連線至 Linux 執行個體。

在 Windows 命令提示中執行下列命令。

1. 在命令提示中，導覽至 .pem 檔案的檔案路徑位置。
2. 執行下列命令，重設並移除明確許可：

```
icacls.exe $path /reset
```

3. 執行下列命令，對目前使用者授予讀取許可：

```
icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"
```

4. 執行下列命令以停用繼承並移除繼承的許可。

```
icacls.exe $path /inheritance:r
```

5. 您應能使用 SSH 從 Windows 連線至 Linux 執行個體。

錯誤：私有金鑰的開頭必須為「-----BEGIN RSA PRIVATE KEY-----」，結尾為「-----END RSA PRIVATE KEY-----」

如果您使用第三方工具 (例如 ssh-keygen) 來建立 RSA 金鑰對，則它會以 OpenSSH 金鑰格式產生私有金鑰。當您連線到您的執行個體時，如果以 OpenSSH 格式使用私有金鑰來解密密碼，則您會得到

錯誤 `Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"`。

若要解決錯誤，私有金鑰必須採用 PEM 格式。使用下列命令，以 PEM 格式建立私有金鑰：

```
ssh-keygen -m PEM
```

錯誤：主機金鑰驗證失敗

如果存放在執行個體上的 `known_hosts` 檔案中的主機金鑰與用戶端上的主機金鑰不相符，就會發生此錯誤。例如，如果您使用一個公有 IP 位址連接到執行個體，然後嘗試使用不同的公有 IP 位址再次連接到該執行個體，則可能會發生不相符情況。新增或移除彈性 IP 位址後可能會發生這種情況，因為這樣做會變更執行個體的公有 IP 位址。

若要解決此錯誤，請先確認主機金鑰或執行個體的網路組態有預期變更。在連線到執行個體之前，您可能也想要驗證主機指紋。連線到執行個體後，可以從 `known_hosts` 檔案中移除舊的主機金鑰。如需說明，請參閱執行個體上使用的 Linux 發行版本的文件。

錯誤：伺服器拒絕了我們的金鑰，或是沒有可用的支援驗證方法

如果使用 PuTTY 來連線到執行個體，並出現下列任一錯誤：Error: Server refused our key 或 Error: No supported authentication methods available，請確認您在連線時使用了適當的 AMI 使用者名稱。在 PuTTY 組態視窗的使用者名稱中輸入使用者名稱。

適合的使用者名稱如下：

用於啟動執行個體的 AMI	預設使用者名稱
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`

用於啟動執行個體的 AMI	預設使用者名稱
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	檢查 AMI 提供者

您也應該確認：

- 使用的 PuTTY 是最新版。如需詳細資訊，請參閱 PuTTY 網頁。
- 您的私有金鑰 (.pem) 檔案是否已正確轉換為 PuTTY (.ppk) 可以辨識的格式。如需有關轉換私有金鑰的詳細資訊，請參閱使用 PuTTY 連線至您的 Linux 執行個體。

無法對執行個體執行 Ping 動作

`ping` 命令是一種 ICMP 傳輸流量 — 如果無法對執行個體執行 ping 命令，請確定傳入安全群組的規則，可允許 ICMP 傳輸流量接收從所有來源，或是從您用來發出指令的電腦或執行個體，所傳送的 `Echo Request` 訊息。

如果無法從執行個體發出 `ping` 指令，請確定安全群組外傳的規則，可允許 ICMP 傳輸流量，來讓 `Echo Request` 訊息傳到所有目的地，或是傳到您想要對其執行 ping 的主機。

`Ping` 命令也可能會遭到防火牆封鎖，或因為網路延遲或硬體問題而逾時。如需進一步疑難排解的協助，請洽詢您的區域網路或系統管理員。

錯誤：伺服器意外關閉網路連線

如果您使用 PuTTY 連接到執行個體，並且收到錯誤「伺服器意外關閉網路連線」，請驗證您在 PuTTY 組態的連線頁面上是否已啟用保持連線，以避免中斷連線。部分伺服器在指定的期間內未收到任何資料時會中斷連接用戶端。將 Seconds between keepalives (保持連線之間的秒數) 設為 59 秒。

如果您在啟用保持連線之後仍遇到問題，請在 PuTTY 組態的連線頁面上嘗試停用 Nagle 的演算法。

錯誤：EC2 Instance Connect 的主機金鑰驗證失敗

如果您輪換執行個體主機金鑰，新的主機金鑰不會自動上傳至 AWS 信任的主機金鑰資料庫。因此當您嘗試使用 EC2 Instance Connect 以瀏覽器為基礎的用戶端連線到您的執行個體時，這會導致主機金鑰驗證失敗，而無法連線到您的執行個體。

若要解決此錯誤，您必須在 `eic_harvest_hostkeys` 執行個體上執行指令碼，以將新的主機金鑰上傳至 EC2 Instance Connect。指令碼位於 Amazon Linux 2 執行個體上的 `/opt/aws/bin/` 和 Ubuntu 執行個體上的 `/usr/share/ec2-instance-connect/`。

Amazon Linux 2

解決 Amazon Linux 2 執行個體上的主機金鑰驗證失敗錯誤

1. 使用 SSH 連接至您的執行個體。

   您可以使用 EC2 Instance Connect CLI 或在您啟動執行個體時指派至該執行個體的 SSH 金鑰對，並使用您在啟動執行個體所用的 AMI 預設使用者名稱，來進行連線。Amazon Linux 2 的預設使用者名稱為 `ec2-user`。

   例如，如果執行個體的啟動方式是 Amazon Linux 2，執行個體的公有 DNS 名稱是 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`，金鑰對是 `my_ec2_private_key.pem`，則使用以下命令來透過 SSH 連接至執行個體：

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。

2. 導覽至以下資料夾。

   ```
   [ec2-user ~]$ cd /opt/aws/bin/
   ```

3. 在執行個體上執行以下命令。

   ```
   [ec2-user ~]$ ./eic_harvest_hostkeys
   ```

   請注意，呼叫成功時不會產生輸出。

   您現在可以使用 EC2 Instance Connect 以瀏覽器為基礎的用戶端連線至執行個體。

Ubuntu

解決 Ubuntu 執行個體上的主機金鑰驗證失敗錯誤

1. 使用 SSH 連接至您的執行個體。

   您可以使用 EC2 Instance Connect CLI 或在您啟動執行個體時指派至該執行個體的 SSH 金鑰對，並使用您在啟動執行個體所用的 AMI 預設使用者名稱，來進行連線。若是 Ubuntu，預設使用者名稱為 ubuntu。

   例如，如果執行個體的啟動方式是 Ubuntu，執行個體的公有 DNS 名稱是 ec2-a-b-c-d.us-west-2.compute.amazonaws.com，金鑰對是 my_ec2_private_key.pem，則使用以下命令來透過 SSH 連接至執行個體：

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   如需連線至執行個體的詳細資訊，請參閱使用 SSH 用戶端連接至 Linux 執行個體。
2. 導覽至以下資料夾。

   ```
   [ec2-user ~]$ cd /usr/share/ec2-instance-connect/
   ```
3. 在執行個體上執行以下命令。

   ```
   [ec2-user ~]$ ./eic_harvest_hostkeys
   ```

   請注意，呼叫成功時不會產生輸出。

   您現在可以使用 EC2 Instance Connect 以瀏覽器為基礎的用戶端連線至執行個體。

無法使用 EC2 Instance Connect 連線至 Ubuntu 執行個體

如果您使用 EC2 Instance Connect 連線至 Ubuntu 執行個體，並且在嘗試連線時出現錯誤，可以使用以下資訊嘗試修復問題。

可能的原因

該執行個體的 ec2-instance-connect 套件並非最新版本。

解決方案

如下所示將該執行個體的 ec2-instance-connect 套件更新至最新版本：

1. 使用 EC2 Instance Connect 以外的方法連線至您的執行個體。
2. 在您的執行個體上執行以下命令，將 ec2-instance-connect 套件更新至最新版本。

```
apt update && apt upgrade
```

我的私有金鑰遺失。如何連線到我的執行個體？

如果遺失了 EBS 後端執行個體的私有金鑰，您可以重新獲得對執行個體的存取權限。您必須停止執行個體、中斷其根磁碟區的連結，並將該磁碟區做為資料磁碟區連接至其他執行個體、使用新的公有金鑰修改 authorized_keys 檔案、將磁碟區移回到原始的執行個體，然後重新啟動執行個體。關於啟動、連線到和停用執行個體，詳細資訊請參閱 Amazon EC2 執行個體狀態變更。

僅具有 EBS 根磁碟區的執行個體支援此程序。如果執行個體有一個執行個體儲存體根磁碟區，則無法使用此程序來重新獲取執行個體的存取權；您必須具備私有金鑰，才能連線到執行個體。若要確定執行個體的根磁碟區類型，請開啟 Amazon EC2 主控台，選擇執行個體，選取執行個體，選擇儲存索引標籤，然後在根裝置詳細資訊區段中，檢查根裝置類型的值。

此值為 EBS 或 INSTANCE-STORE。

除了下列步驟之外，遺失私有金鑰時還有其他方法可以連線到 Linux 執行個體。如需詳細資訊，請參閱如果我在初次啟動後遺失了 SSH 金鑰對，該如何連接到 Amazon EC2 執行個體？

使用不同的金鑰對以連線到 EBS 後端執行個體的步驟

- 步驟 1：建立新的金鑰對
- 步驟 2：取得關於原始執行個體及其根磁碟區的資訊
- 步驟 3：停止原始執行個體
- 步驟 4：啟動暫時執行個體
- 步驟 5：將根磁碟區與原始執行個體分離，並將其連接到暫時執行個體
- 步驟 6：在掛載到暫時執行個體的原始磁碟區上，將新的公有金鑰新增至的 authorized_keys
- 步驟 7：取消掛載原始磁碟區，並將該磁碟區從暫時執行個體分離，然後再重新連接到原始執行個體
- 步驟 8：使用新的金鑰對連線到原始執行個體
- 步驟 9：清理

步驟 1：建立新的金鑰對

使用 Amazon EC2 主控台或第三方工具來產生新的金鑰對。如果希望新金鑰對的名稱，和已遺失私有金鑰的相同，您必須先刪除現有的金鑰對。如需建立新金鑰對的詳細資訊，請參閱使用 Amazon EC2 建立金鑰對或使用第三方工具來建立金鑰對，然後將公有金鑰匯入 Amazon EC2。

步驟 2：取得關於原始執行個體及其根磁碟區的資訊

請儲存下列資訊，完成此程序將會需要用到這些資訊。

取得原始執行個體的相關資訊

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中選取 Instances (執行個體)，然後選取您要連線的執行個體。(我們會將此執行個體稱為「原始」執行個體。)
3. 在 Details (詳細資訊) 索引標籤上記下執行個體 ID 和 AMI ID。
4. 在 網路 索引標籤上記下 「可用區域」。
5. 在 Storage (儲存) 索引標籤下的 根裝置名稱，記下根磁碟區的裝置名稱 (例如：`/dev/xvda`)。接著，在 區塊型儲存裝置 下找出這個裝置名稱，並記下磁碟區 ID (例如：vol-0a1234b5678c910de)。

步驟 3：停止原始執行個體

選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。如果此選項已停用，則執行個體已停止或其根磁碟區是執行個體儲存體磁碟區。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

步驟 4：啟動暫時執行個體

啟動暫時執行個體

1. 在導覽窗格中，選擇 Instances (執行個體)，然後選擇 Launch instances (啟動執行個體)。

2. 在 Name and tags (名稱和標籤) 區段，針對 Name (名稱)，輸入 Temporary (暫時)。
3. 在 Application and OS Images (應用程式和作業系統映像) 區段，選取您用來啟動原始執行個體的同一個 AMI。如果此 AMI 無法使用，您可以從已停止的執行個體，建立能夠使用的 AMI。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。
4. 在 Instance type (執行個體類型) 區段，保留預設執行個體類型。
5. 在 Key pair (金鑰對) 區段，針對 Key pair name (金鑰對名稱)，選取要使用的現有金鑰對或建立新的金鑰對。
6. 在 Network settings (網路設定) 區段，選取 Edit (編輯)，然後針對 Subnet (子網)，選取與原始執行個體相同可用區域中的子網。
7. 在 Summary (摘要) 面板中，選擇 Launch (啟動)。

步驟 5：將根磁碟區與原始執行個體分離，並將其連接到暫時執行個體

1. 在導覽窗格中，選擇磁碟區，接著再選取原始執行個體的根磁碟區 (您已在先前的步驟中記下其磁碟區 ID)。選取 Actions (動作)、Detach Volume (分離磁碟區)，然後選取 Detach (分離)。等待磁碟區的狀態變成 `available`。(您可能需要選取 Refresh (重新整理) 圖示。)
2. 在已選取磁碟區的狀態下，選取 Actions (動作)，然後再選取 Attach volume (連接磁碟區)。選取暫時執行個體的執行個體 ID、記下 Device name (裝置名稱) 中的指定裝置名稱 (例如：`/dev/sdf`)，接著再選取 Attach volume (連接磁碟區)。

> **Note**
>
> 如果您從 AWS Marketplace AMI 啟動原始執行個體，且磁碟區包含 AWS Marketplace 代碼，您必須先停止暫時執行個體，才能連接磁碟區。

步驟 6：在掛載到暫時執行個體的原始磁碟區上，將新的公有金鑰新增至的 `authorized_keys`

1. 連線到暫時執行個體。
2. 從暫時執行個體掛載已連結到執行個體的磁碟區，如此您就能存取其檔案系統。例如，如果裝置名稱為 `/dev/sdf`，請使用下列的指令來將磁碟區掛載為 `/mnt/tempvol`。

> **Note**
>
> 在執行個體上所顯示的裝置名稱可能會有不同。例如，掛載為 /dev/sdf 的裝置，在執行個體上可能會顯示為 /dev/xvdf。Red Hat 的某些版本 (或是其變體版本，例如 CentOS) 甚至可能會將結尾的字母增加 4 個字元，其中 /dev/sd*f* 會變成 /dev/xvd*k*。

a. 使用 lsblk 命令來查看磁碟區是否已分割。

```
[ec2-user ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
##xvda1 202:1    0    8G  0 part /
xvdf    202:80   0  101G  0 disk
##xvdf1 202:81   0  101G  0 part
xvdg    202:96   0   30G  0 disk
```

在上述的範例中，/dev/xvda 和 /dev/xvdf 為分割的磁碟區，而 /dev/xvdg 則不是。如果磁碟區已分割，則在接下來的步驟中，您會掛載 (/dev/xvdf1) 而非原始裝置 (/dev/xvdf)。

b. 建立暫時的目錄來掛載磁碟區。

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

c. 使用您先前找出的磁碟區名稱或裝置名稱，將磁碟區 (或分割磁碟區) 掛載至暫時掛載點。必要命令取決於作業系統的檔案系統。請注意，在執行個體上所顯示的設備名稱可能會有不同。如需詳細資訊，請參閱步驟 6 中的 note。

- Amazon Linux、Ubuntu 和 Debian

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

- Amazon Linux 2、CentOS、SUSE Linux 12 和 RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

> **Note**
>
> 如果您收到檔案系統毀損的錯誤，請執行下列命令，使用 fsck 公用程式來檢查檔案系統並修復任何問題：
>
> ```
> [ec2-user ~]$ sudo fsck /dev/xvdf1
> ```

3. 從暫時執行個體使用下列的指令，用暫時執行個體的 authorized_keys 所產生的新公有金鑰，來更新已掛載磁碟區上的 authorized_keys。

> **Important**
>
> 下列的範例使用 Amazon Linux 使用者名稱 ec2-user。您可以需要換成不同的使用者名稱，例如適用於 Ubuntu 執行個體的 ubuntu。

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

如果此項複製動作成功，您就可以進行下一個步驟。

(選用) 否則，如果您沒有編輯 /mnt/tempvol 中的檔案的許可，必須使用 sudo 來更新檔案，然後檢查對於該檔案的許可，以確認您是否能夠登入原始執行個體。使用下列命令以檢查對檔案的許可。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
total 4
-rw------- 1 222 500 398 Sep 13 22:54 authorized_keys
```

在本範例輸出中，*222* 為使用者 ID，而 *500* 為群組 ID。接下來，請使用 sudo 來重新執行先前失敗的複製命令。

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

再次執行下列命令，以確定許可是否已變更。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

如果使用者 ID 和群組 ID 已變更，請使用下列命令來還原這些 ID。

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

步驟 7：取消掛載原始磁碟區，並將該磁碟區從暫時執行個體分離，然後再重新連接到原始執行個體

1. 從暫時執行個體取消掛載已連結的磁碟區，如此您就能將該磁碟區重新連結到原始執行個體。例如，使用以下命令來取消掛載位於 `/mnt/tempvol` 的磁碟區。

   ```
   [ec2-user ~]$ sudo umount /mnt/tempvol
   ```

2. 將磁碟區從暫時執行個體分離 (您在上一步已卸載該磁碟區)：從 Amazon EC2 主控台，選取導覽窗格中的磁碟區，選取原始執行個體的根磁碟區 (您在上一步已記下磁碟區 ID)，選取動作、分離磁碟區，接著再選取分離。等待磁碟區的狀態變成 `available`。(您可能需要選取 Refresh (重新整理) 圖示。)
3. 將磁碟區重新連接到原始執行個體：在已選取磁碟區的狀態下，選取 Actions (動作)、Attach volume (連接磁碟區)。選取原始執行個體的執行個體 ID，指定您先前在步驟 2 中為原始根磁碟區連接 (`/dev/sda1` 或 `/dev/xvda`) 記錄的設備名稱，然後選取連接磁碟區。

   > **Important**
   >
   > 如果您並未指定與原始連結相同的裝置名稱，則無法啟動原始執行個體。Amazon EC2 預期根磁碟區位於 `sda1` 或 `/dev/xvda`。

步驟 8：使用新的金鑰對連線到原始執行個體

選取原始執行個體，然後依序選取 Instance state (執行個體狀態)、Start instance (啟動執行個體)。在執行個體進入 `running` 狀態之後，您可以使用新金鑰對的私有金鑰檔案，來連線到該執行個體。

> **Note**
>
> 如果新金鑰對的名稱和對應的私有金鑰檔案，與原始金鑰對的名稱不同，請務必在連線到執行個體時，指定新私有金鑰檔案的名稱。

步驟 9：清理

(選用) 如果不需再使用時，您可以終止暫時執行個體。選取暫時執行個體，再依序選取執行個體狀態、終止 (刪除) 執行個體。

狀態檢查失敗的疑難排解 Amazon EC2 Linux 執行個體

如果您的 Linux 執行個體無法通過狀態檢查，則以下資訊有助於針對問題進行疑難排解。首先判斷應用程式是否出現任何問題。如果確認執行個體並未如預期執行應用程式，請檢閱狀態檢查資訊和系統日誌。

關於導致狀態檢查失敗的問題範例，請參閱 Amazon EC2 執行個體的狀態檢查。

內容

- 檢閱狀態檢查資訊
- 擷取系統日誌
- 針對 Linux 執行個體，進行系統日誌錯誤的疑難排解
- 記憶體不足：終止程序
- 錯誤：mmu_update 失敗 (記憶體管理更新作業失敗)
- I/O 錯誤 (區塊型儲存設備故障)
- I/O 錯誤：非本機也非遠端磁碟 (故障的分散式區塊型儲存設備)
- request_module：失控迴圈 modprobe (在舊版 Linux 上建立舊核心 modprobe 的迴圈)
- 「嚴重：核心版本太舊」和「fsck：嘗試開啟 /dev 時，找不到此等檔案或目錄」(核心與 AMI 不符)
- 「嚴重：無法載入 /lib/modules」或「BusyBox」(缺少核心模組)
- 錯誤：無效的核心 (與 EC2 不相容的核心)
- fsck：嘗試開啟時找不到此等檔案或目錄... (找不到檔案系統)
- 掛載檔案作業系統的一般錯誤 (掛載失敗)

- VFS：無法在未知的區塊上掛載根 fs (根檔案系統不符)
- 錯誤：無法判定根設備的主/次編號... (根檔案系統/裝置不相符)
- XENBUS：裝置缺少驅動程式...
- ...未進行檢查的天數，強制進行檢查 (需要進行檔案系統檢查)
- fsck 凍結於結束狀態... (缺少裝置)
- GRUB 提示 (grubdom>)
- 帶出界面 eth0：裝置 eth0 的 MAC 地址和預期的不同，略過。(將 MAC 地址直接寫入程式)
- 無法載入 SELinux 政策。機器正處於強制執行模式。現在停止中。(SELinux 組態設定不當)
- XENBUS：連線到裝置的作業逾時 (Xenbus 逾時)

檢閱狀態檢查資訊

使用 Amazon EC2 主控台來調查狀態不健全的執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選取 Instances (執行個體)，然後選取您的執行個體。
3. 選取狀態和警示索引標籤，以查看所有系統狀態檢查、執行個體狀態檢查和連接 EBS 狀態檢查的個別結果。

如果未通過狀態檢查，您可以嘗試下列其中一個選項：

- 建立警示以復原執行個體，以回應失敗的狀態檢查。如需詳細資訊，請參閱建立警示以停止、終止、重新啟動或復原執行個體。
- (執行個體狀態檢查) 如果您已將執行個體類型變更為 Nitro 型執行個體，則從沒有必要 ENA 和 NVMe 驅動程式的執行個體移轉時，狀態檢查會失敗。如需詳細資訊，請參閱變更執行個體類型的相容性。
- 如果是具有 EBS 根磁碟區的執行個體，請停止並重新啟動執行個體。如需詳細資訊，請參閱停止和啟動 Amazon EC2 執行個體。
- 如果是具有執行個體儲存體根磁碟區的執行個體，請終止執行個體，並啟動替代的執行個體。如需詳細資訊，請參閱終止 Amazon EC2 執行個體。
- 請等候 Amazon EC2 解決問題。
- 聯絡 支援 或發佈您的問題以 AWS re：Post。
- 如果您的執行個體位於 Auto Scaling 群組：

- (系統狀態檢查和執行個體狀態檢查) 預設情況下，Amazon EC2 Auto Scaling 會自動啟動替代執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的針對 Auto Scaling 群組中的執行個體進行運作狀態檢查。
- (連接的 EBS 狀態檢查) 您必須設定 Amazon EC2 Auto Scaling 以自動啟動替代執行個體。如需詳細資訊，請參閱「Amazon EC2 Auto Scaling 使用者指南」中的使用受損的 Amazon EBS 磁碟區監控和取代自動擴展執行個體。

- 擷取系統日誌並尋找錯誤。如需詳細資訊，請參閱擷取系統日誌。

擷取系統日誌

如果執行個體狀態檢查未通過，您可以重新啟動執行個體和擷取系統日誌。日誌可能會顯示錯誤，有助於對發生的問題進行故障診斷。重新啟動作業會清除日誌中不必要的資訊。

重新啟動執行個體和擷取系統日誌

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取您的執行個體。
3. 選擇 Instance state (執行個體狀態)、Reboot intance (重新啟動執行個體)。執行個體重新啟動可能需要幾分鐘的時間。
4. 確認問題是否仍持續存在；在某些情況中，重新啟動也許可以解決問題。
5. 當執行個體處於 `running` 狀態時，選擇 Actions (動作)、Monitor and troubleshoot (監視和疑難排解)、Get system log (取得系統日誌檔)。
6. 檢閱畫面上所顯示的日誌，然後使用下列的已知系統日誌錯誤陳述清單，來進行問題的故障診斷。
7. 如果問題無法解決，您可以在 AWS re:Post 中發佈您的問題。

針對 Linux 執行個體，進行系統日誌錯誤的疑難排解

如果 Linux 執行個體未通過執行個體的狀態檢查 (例如執行個體可達性檢查)，請確認是否已按照上述的步驟來擷取系統日誌。下列清單包含了一些常見的系統日誌錯誤和建議的動作，您可以針對每項錯誤採取這些動作，來解決問題。

Memory Errors (記憶體錯誤)

- 記憶體不足：終止程序
- 錯誤：mmu_update 失敗 (記憶體管理更新作業失敗)

Device Errors (裝置錯誤)

- I/O 錯誤 (區塊型儲存設備故障)
- I/O 錯誤：非本機也非遠端磁碟 (故障的分散式區塊型儲存設備)

Kernel Errors (核心錯誤)

- request_module：失控迴圈 modprobe (在舊版 Linux 上建立舊核心 modprobe 的迴圈)
- 「嚴重：核心版本太舊」和「fsck：嘗試開啟 /dev 時，找不到此等檔案或目錄」(核心與 AMI 不符)
- 「嚴重：無法載入 /lib/modules」或「BusyBox」(缺少核心模組)
- 錯誤：無效的核心 (與 EC2 不相容的核心)

File System Errors (檔案系統錯誤)

- fsck：嘗試開啟時找不到此等檔案或目錄... (找不到檔案系統)
- 掛載檔案作業系統的一般錯誤 (掛載失敗)
- VFS：無法在未知的區塊上掛載根 fs (根檔案系統不符)
- 錯誤：無法判定根設備的主/次編號... (根檔案系統/裝置不相符)
- XENBUS：裝置缺少驅動程式...
- ...未進行檢查的天數，強制進行檢查 (需要進行檔案系統檢查)
- fsck 凍結於結束狀態... (缺少裝置)

Operating System Errors (作業系統錯誤)

- GRUB 提示 (grubdom>)
- 帶出界面 eth0：裝置 eth0 的 MAC 地址和預期的不同，略過。(將 MAC 地址直接寫入程式)
- 無法載入 SELinux 政策。機器正處於強制執行模式。現在停止中。(SELinux 組態設定不當)
- XENBUS：連線到裝置的作業逾時 (Xenbus 逾時)

記憶體不足：終止程序

系統日誌記錄項目會顯示記憶體不足的錯誤，類似於下列所示記錄。

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
```

```
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

可能的原因

記憶體用盡

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列其中一項： • 請停止執行個體，並修改執行個體以使用不同的執行個體類型，然後再次啟動該執行個體。例如，較大容量或記憶體最佳化的執行個體類型。 • 重新啟動執行個體，以讓執行個體回到健全狀態。除非您變更執行個體類型，否則問題可能會再次發生。
執行個體存放區後端	請執行下列其中一項： • 終止執行個體並啟動新的執行個體，指定不同的執行個體類型。例如，較大容量或記憶體最佳化的執行個體類型。 • 重新啟動執行個體，以讓執行個體回到健全狀態。除非您變更執行個體類型，否則問題可能會再次發生。

錯誤：mmu_update 失敗 (記憶體管理更新作業失敗)

系統日誌記錄項目會顯示記憶體管理更新作業的失敗，類似於下列所示記錄：

```
...
Press `ESC' to enter the menu... 0   [H[J  Booting 'Amazon Linux 2011.09
 (2.6.35.14-95.38.amzn1.i686)'
```

```
root (hd0)
 Filesystem type is ext2fs, using whole disk
kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us
initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img
ERROR: mmu_update failed with rc=-22
```

可能的原因

Amazon Linux 的問題

建議動作

將您的問題發佈至 AWS re：Post或聯絡 支援。

I/O 錯誤 (區塊型儲存設備故障)

系統日誌記錄項目會顯示輸入/輸出錯誤，類似於下列的範例：

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
```

```
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

可能的原因

執行個體類型	可能的原因
Amazon EBS 後端	故障的 Amazon EBS 磁碟區
執行個體存放區後端	故障的實體磁碟機

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體。 2. 分離磁碟區。 3. 試著將磁碟區復原。 **Note** 經常性地建立 Amazon EBS 磁碟區快照，是建議的理想做法。這可以大幅降低因為故障而造成資料遺失的風險。 4. 將磁碟區重新連結到執行個體。 5. 啟動實例。
執行個體存放區後端	終止執行個體並啟動新的執行個體。 **Note** 資料無法復原。請從備份資料復原。

如果是此種執行個體類型	執行此作業
	Note 理想的做法是使用 Amazon S3 或 Amazon EBS 來進行備份。執行個體存放磁碟區會與單一主機或單一磁碟的故障直接連動。

I/O 錯誤：非本機也非遠端磁碟 (故障的分散式區塊型儲存設備)

系統日誌記錄項目會顯示裝置上的輸入/輸出錯誤，類似於下列的範例：

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056

lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

可能的原因

執行個體類型	可能的原因
Amazon EBS 後端	故障的 Amazon EBS 磁碟區
執行個體存放區後端	故障的實體磁碟機

建議動作

終止執行個體並啟動新的執行個體。

如果是 Amazon EBS 後端執行個體，您可以藉由從最近的快照建立映像，來回復資料。在快照建立後所新增的任何資料，皆無法復原。

request_module：失控迴圈 modprobe (在舊版 Linux 上建立舊核心 modprobe 的迴圈)

系統日誌記錄項目會顯示此狀況，類似於下列所示。使用不穩定的或舊版的 Linux 核心 (例如 2.6.16-xenU)，可能會在啟動時造成無止盡的循環狀況。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

 Xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請使用下列其中一個選項，來使用較新版的核心 (採用 GRUB 或靜態的核心)： 選項 1：終止執行個體並啟動新的執行個體，指定 `-kernel` 和 `-ramdisk` 參數。 選項 2：

如果是此種執行個體類型	執行此作業
	1. 停止執行個體。 2. 修改核心和記憶體虛擬磁碟 (ramdisk) 的屬性，以使用較新版的核心。 3. 啟動實例。
執行個體存放區後端	終止執行個體並啟動新的執行個體，指定 `-kernel` 和 `-ramdisk` 參數。

「嚴重：核心版本太舊」和「fsck：嘗試開啟 /dev 時，找不到此等檔案或目錄」(核心與 AMI 不符)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

可能的原因

不相容的核心和使用者空間

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體。 2. 修改組態，以使用較新版的核心。 3. 啟動實例。
執行個體存放區後端	請執行下列步驟： 1. 建立 AMI，此 AMI 會使用較新版的核心。

如果是此種執行個體類型	執行此作業
	2. 終止執行個體。 3. 從您所建立的 AMI 啟動新的執行個體。

「嚴重：無法載入 /lib/modules」或「BusyBox」(缺少核心模組)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or
 directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No
 such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device.  Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
ALERT! /dev/sda1 does not exist. Dropping to a shell!
```

```
BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

可能的原因

這項問題可能是下列一種或多種狀況所造成：

- 缺少記憶體虛擬磁碟 (ramdisk)
- 記憶體虛擬磁碟 (ramdisk) 缺少正確的模組
- Amazon EBS 根磁碟區未正確地連結為 `/dev/sda1`

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 為 Amazon EBS 磁碟區選擇修正過的記憶體虛擬磁碟 (ramdisk)。 2. 停止執行個體。 3. 先分離然後修復磁碟區。 4. 將磁碟區連結到執行個體。 5. 啟動實例。 6. 修改 AMI 以使用修正過的記憶體虛擬磁碟 (ramdisk)。
執行個體存放區後端	請執行下列步驟： 1. 終止執行個體，並使用正確的記憶體虛擬磁碟 (ramdisk) 來啟動新的執行個體。 2. 使用正確的記憶體虛擬磁碟 (ramdisk) 來建立新的 AMI。

錯誤：無效的核心 (與 EC2 不相容的核心)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
...
root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

可能的原因

這項問題可能是下列狀況中的其中一種或兩種同時造成：

- GRUB 不支援隨附的核心
- 備用核心不存在

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟：

如果是此種執行個體類型	執行此作業
	1. 停止執行個體。 2. 換成可正常運作的核心。 3. 安裝備用核心。 4. 透過修正核心來修改 AMI。
執行個體存放區後端	請執行下列步驟： 1. 終止執行個體，並使用正確的核心來啟動新的執行個體。 2. 使用正確的核心來建立 AMI。 3. (選用) 透過 支援 來尋求資料復原的技術協助。

fsck：嘗試開啟時找不到此等檔案或目錄... (找不到檔案系統)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
  Welcome to Fedora
  Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [  OK  ]

Starting udev: [  OK  ]

Setting hostname localhost:  [  OK  ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[  OK  ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh
```

```
/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]


*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

可能的原因

- ramdisk 檔案系統定義 /etc/fstab 中存在錯誤
- /etc/fstab 中設定錯誤的檔案系統定義
- 找不到磁碟機/磁碟機故障

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體、分離根磁碟區、修復/修改 /etc/fstab 磁碟區、將磁碟區連結到執行個體，以及啟動執行個體。 2. 修正 ramdisk，以納入修改過的 /etc/fstab (如有適用)。 3. 修改 AMI，以使用較新版的 ramdisk。 fstab 中的第 6 個欄位定義了掛載的可用性要求 – 非 0 值暗示 fsck 將在該磁碟區上完成，而且必須成功。在 Amazon EC2 中使用此欄位可

如果是此種執行個體類型	執行此作業
	能會出現問題，因為故障通常會產生互動式主控台提示，而 Amazon EC2 中目前並未提供此提示功能。使用此功能時請特別注意，並閱讀 fstab 的 Linux 手冊頁。
執行個體存放區後端	請執行下列步驟： 1. 終止執行個體並啟動新的執行個體。 2. 分離所有錯誤的 Amazon EBS 磁碟區，並重新啟動執行個體。 3. (選用) 透過 支援 來尋求資料復原的技術協助。

掛載檔案作業系統的一般錯誤 (掛載失敗)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting.  Commit interval 5 seconds

EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
```

```
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

可能的原因

執行個體類型	可能的原因
Amazon EBS 後端	• 已分離或故障的 Amazon EBS 磁碟區。 • 已毀損的檔案系統。 • 不匹配的 ramdisk 和 AMI 組合 (例如 Debian ramdisk 搭配 SUSE AMI)。
執行個體存放區後端	• 故障的磁碟機。 • 已毀損的檔案系統。 • 不匹配的 ramdisk 和組合 (例如 Debian ramdisk 搭配 SUSE AMI)。

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體。 2. 分離根 磁碟區。

如果是此種執行個體類型	執行此作業
	3. 將根磁碟區連結到已知可正常運作的執行個體。 4. 執行檔案系統檢查 (fsck -a /dev/...)。 5. 修正所有錯誤。 6. 將磁碟區從已知可正常運作的執行個體分離。 7. 將磁碟區連結到已停止的執行個體。 8. 啟動實例。 9. 重新檢查執行個體的狀態。
執行個體存放區後端	請嘗試執行下列其中一項操作： • 啟動新的執行個體。 • (選用) 透過 支援 來尋求資料復原的技術協助。

VFS：無法在未知的區塊上掛載根 fs (根檔案系統不符)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
 20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line:  root=/dev/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

可能的原因

執行個體類型	可能的原因
Amazon EBS 後端	• 未正確地連結裝置。 • 根設備未連結到正確的裝置點。

執行個體類型	可能的原因
	• 檔案系統未採用預期的格式。 • 使用舊版的核心 (例如 2.6.16-XenU)。 • 在執行個體上最近更新了核心 (更新作業失敗或出現錯誤)
執行個體存放區後端	硬體裝置故障。

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列其中一項： • 停止並重新啟動執行個體。 • 修改根磁碟區，以掛載於正確的裝置點 (可能是 /dev/sda1 而非 /dev/sda)。 • 停止並進行修改，以使用現代化的核心。 • 請參考您 Linux 版本的文件，以查看已知的更新錯誤。變更或重新安裝核心。
執行個體存放區後端	終止執行個體，並使用現代化的核心來啟動新的執行個體。

錯誤：無法判定根設備的主/次編號... (根檔案系統/裝置不相符)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
```

```
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

可能的原因

- 找不到虛擬區塊型儲存設備驅動程式，或是該驅動程式設定不正確
- 裝置列舉衝突 (sda 對 xvda 或 sda 而非 sda1)
- 所選的執行個體核心不正確

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體。 2. 分離磁碟區。 3. 修正裝置映射問題。 4. 啟動實例。 5. 修正 AMI 以解決裝置映射問題。
執行個體存放區後端	請執行下列步驟： 1. 透過適當的修正來建立新的 AMI (正確地對應區塊型儲存設備)。 2. 終止執行個體，並從您所建立的 AMI 來啟動新的執行個體。

XENBUS：裝置缺少驅動程式...

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

可能的原因

- 找不到虛擬區塊型儲存設備驅動程式，或是該驅動程式設定不正確
- 裝置列舉衝突 (sda 對 xvda)
- 所選的執行個體核心不正確

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止執行個體。 2. 分離磁碟區。 3. 修正裝置映射問題。 4. 啟動實例。 5. 修正 AMI 以解決裝置映射問題。

如果是此種執行個體類型	執行此作業
執行個體存放區後端	請執行下列步驟： 1. 透過適當的修正來建立 AMI (正確地對應區塊型儲存設備)。 2. 終止執行個體，並使用您所建立的 AMI 來啟動新的執行個體。

...未進行檢查的天數，強制進行檢查 (需要進行檔案系統檢查)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

可能的原因

已超過檔案系統檢查時間；強制進行檔案系統檢查。

建議動作

- 請等候檔案系統檢查完成。視根檔案系統的大小而定，檔案系統檢查可能需要一段長時間才會完成。
- 使用 tune2fs 或檔案系統適用的工具，來修改檔案系統，以移除強制進行檔案系統檢查 (fsck) 的功能。

fsck 凍結於結束狀態... (缺少裝置)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
```

```
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

可能的原因

- Ramdisk 尋找缺少的磁碟機
- 強制進行檔案系統一致性檢查
- 磁碟機故障或已分離

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請嘗試執行下列其中一項或多項動作來解決問題： • 停止執行個體、將磁碟區連結到執行中的現有執行個體。 • 手動執行一致性檢查。 • 修正 ramdisk，以加入相關的公用程式。 • 修改檔案系統調校參數，以移除一致性要求 (不建議)。
執行個體存放區後端	請嘗試執行下列其中一項或多項動作來解決問題： • 將 ramdisk 與正確的工具重新綁定為套件。 • 修改檔案系統調校參數，以移除一致性要求 (不建議)。 • 終止執行個體並啟動新的執行個體。 • (選用) 透過 支援 來尋求資料復原的技術協助。

GRUB 提示 (grubdom>)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
    GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

       [ Minimal BASH-like line editing is supported.   For

         the   first   word,  TAB  lists  possible  command

         completions.  Anywhere else TAB lists the possible

         completions of a device/filename. ]

grubdom>
```

可能的原因

執行個體類型	可能的原因
Amazon EBS 後端	• 找不到 GRUB 組態檔案。 • 使用了不正確的 GRUB 映像，預期 GRUB 組態檔案儲存於不同的位置。 • 使用了不支援的檔案系統來存放 GRUB 組態檔案 (例如，將根檔案系統轉換為 GRUB 先前版本不支援的類型)。
執行個體存放區後端	• 找不到 GRUB 組態檔案。 • 使用了不正確的 GRUB 映像，預期 GRUB 組態檔案儲存於不同的位置。 • 使用了不支援的檔案系統來存放 GRUB 組態檔案 (例如，將根檔案系統轉換為 GRUB 先前版本不支援的類型)。

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	選項 1：修改 AMI 並重新啟動執行個體： 1. 修改來源 AMI，以在標準位置建立 GRUB 組態檔案 (/boot/grub/menu.lst)。 2. 確認您的 GRUB 版本支援底層檔案系統的類型，必要時請更新 GRUB。 3. 選擇適合的 GRUB 映像 (hd0-第 1 個磁碟機或 hd00 – 第 1 個磁碟機，第 1 個分割區)。 4. 終止執行個體，並使用您所建立的 AMI 來啟動新的執行個體。 選項 2：修改現有的執行個體： 1. 停止執行個體。 2. 分離根檔案系統。 3. 將根檔案系統連結到已知可正常運作的執行個體。 4. 掛載檔案系統。 5. 建立 GRUB 組態檔案。 6. 確認您的 GRUB 版本支援底層檔案系統的類型，必要時請更新 GRUB。 7. 分離檔案系統。 8. 連結到原始執行個體。 9. 修改核心屬性，以使用適合的 GRUB 映像 (第 1 個磁碟機或第 1 個磁碟機上的第 1 個分割區)。 10啟動實例。
執行個體存放區後端	選項 1：修改 AMI 並重新啟動執行個體：

如果是此種執行個體類型	執行此作業
	1. 使用儲存於標準位置的 GRUB 組態檔案 (/boot/grub/menu.lst)，來建立新的 AMI。 2. 選擇適合的 GRUB 映像 (hd0-第 1 個磁碟機或 hd00 – 第 1 個磁碟機，第 1 個分割區)。 3. 確認您的 GRUB 版本支援底層檔案系統的類型，必要時請更新 GRUB。 4. 終止執行個體，並使用您所建立的 AMI 來啟動新的執行個體。 選項 2：終止執行個體，並啟動新的執行個體 (指定正確的核心)。 Note 若要從現有的執行個體復原資料，請聯絡 支援。

帶出界面 eth0：裝置 eth0 的 MAC 地址和預期的不同，略過。(將 MAC 地址直接寫入程式)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
...
Bringing up loopback interface:  [  OK  ]

Bringing up interface eth0:  Device eth0 has different MAC address than expected,
 ignoring.
[FAILED]

Starting auditd: [  OK  ]
```

可能的原因

在 AMI 組態中有直接寫入程式的界面 MAC

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列其中一項： • 修改 AMI，以移除直接寫入程式的項目，並重新啟動執行個體。 • 修改執行個體，以移除直接寫入程式的 MAC 位址。 或是 請執行下列步驟： 1. 停止執行個體。 2. 分離根 磁碟區。 3. 將磁碟區連結到另一個執行個體，並修改該磁碟區，以移除直接寫入程式的 MAC 位址。 4. 將磁碟區連結到原始執行個體。 5. 啟動實例。
執行個體存放區後端	請執行下列其中一項： • 修改執行個體，以移除直接寫入程式的 MAC 位址。 • 終止執行個體並啟動新的執行個體。

無法載入 SELinux 政策。機器正處於強制執行模式。現在停止中。(SELinux 組態設定不當)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

可能的原因

SELinux 已在啟用時發生錯誤：

- GRUB 不支援隨附的核心
- 備用核心不存在

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列步驟： 1. 停止已故障的執行個體。 2. 分離已故障執行個體的根磁碟區。 3. 將根磁碟區連結到另一個執行中的 Linux 執行個體 (之後稱為復原執行個體)。 4. 連線到復原執行個體，並掛載已故障執行個體的根磁碟區。 5. 將已掛載根磁碟區上的 SELinux 停用。不同 Linux 版本的此項程序也會有所差異，如需詳細資訊，請參閱特定作業系統的文件。 **Note** 在某些系統上，您可以透過設定 `SELINUX=disabled` 檔案中的 `/mount_point /etc/sysconfig/selinux` ，來停用 SELinux，其中 *mount_point* 是您在復原執行個體上掛載磁碟區的位置。 6. 取消掛載根磁碟區、將該磁碟區從復原執行個體分離，然後再重新連結到原始執行個體。 7. 啟動實例。

如果是此種執行個體類型	執行此作業
執行個體存放區後端	請執行下列步驟： 1. 終止執行個體並啟動新的執行個體。 2. (選用) 透過 支援 來尋求資料復原的技術協助。

XENBUS：連線到裝置的作業逾時 (Xenbus 逾時)

系統日誌記錄項目會顯示此狀況，類似於下列所示。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found.  Try passing init= option to kernel.
```

可能的原因

- 區塊型儲存設備未連線到執行個體
- 執行個體使用舊版的執行個體核心

建議動作

如果是此種執行個體類型	執行此作業
Amazon EBS 後端	請執行下列其中一項： • 修改 AMI 和執行個體，以使用現代化的核心並重新啟動執行個體。 • 重新啟動執行個體。
執行個體存放區後端	請執行下列其中一項： • 終止執行個體。

如果是此種執行個體類型	執行此作業
	• 修改 AMI，以使用現代化的核心，並使用此 AMI 來啟動新的執行個體。

從錯誤磁碟區對 Amazon EC2 Linux 執行個體開機進行疑難排解

在某些情況中，除了連結至 /dev/xvda 或 /dev/sda 的磁碟區之外，其他磁碟區會成為 Linux 執行個體的根磁碟區。這可能是因為您已經將另一個執行個體的根磁碟區，或是從根磁碟區快照所建立的磁碟區，連結到包含現有根磁碟區的執行個體。

這是 Linux 中初始記憶體虛擬磁碟 (ramdisk) 的運作方式所造成的。ramdisk 會選擇在 / 中定義為 /etc/fstab 的磁碟區，而在某些發行版本中，這取決於附加至磁碟區分割區的標籤。具體來說，您會發現 /etc/fstab 看起來類似於下列的內容：

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

如果查看這兩個磁碟區的標籤，會發現皆包含 / 標籤：

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

在此範例中，最後可能是由 /dev/xvdf1 成為根磁碟區，您的執行個體會在初始的 ramdisk 執行之後，透過此根設備啟動，而非透過您原本要從中啟動的 /dev/xvda1 磁碟區來啟動。為了解決這項問題，請使用同樣的 e2label 命令，針對您不想從其啟動的已連結磁碟區，來變更其標籤。

在某些情況中，在 /etc/fstab 中指定 UUID 可以解決這項問題。不過，如果兩個磁碟區都來自於相同的快照，或次要的磁碟區是從主要磁碟區的快照建立，兩個磁碟區就會共用一個 UUID。

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

```
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

更改已連結 ext4 磁碟區的標籤

1. 使用 e2label 命令，將磁碟區的標籤變更為除了 / 以外的名稱。

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. 確認磁碟區已具有新的標籤。

```
[ec2-user ~]$ sudo e2label /dev/xvdf1
old/
```

更改已連結 xfs 磁碟區的標籤

- 使用 xfs_admin 命令，將磁碟區的標籤變更為除了 / 以外的名稱。

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
writing all SBs
new label = "old/"
```

在如上述變更磁碟區標籤之後，您應該就能夠重新啟動執行個體，而且讓初始 ramdisk 在執行個體啟動時，選擇正確的磁碟區。

> **Important**
>
> 如果您打算分離具有新標籤的磁碟區，並將該磁碟區傳回給另一個執行個體做為根磁碟區使用，則您必須再次執行上述的程序，以將磁碟區的標籤改回原來的值。否則，其他的執行個體就不會啟動，因為 ramdisk 找不到具有 / 標籤的磁碟區。

對連線至 Amazon EC2 Windows 執行個體的問題進行疑難排解

下列資訊和常見錯誤可協助您針對連線到 Windows 執行個體時的問題進行疑難排解。

連線問題

- 遠端桌面無法連線到遠端電腦

- 使用 macOS RDP 用戶端時發生錯誤
- RDP 顯示黑色螢幕而非桌面
- 無法使用非管理員的使用者從遠端登入至執行個體
- 使用 對遠端桌面問題進行故障診斷 AWS Systems Manager
- 使用遠端登錄在 EC2 執行個體上啟用遠端桌面
- 我的私有金鑰遺失。如何連接到我的 Windows 執行個體？

遠端桌面無法連線到遠端電腦

嘗試下列方法解決執行個體連線相關問題：

- 確認您使用的是正確的公有 DNS 主機名稱。(在 Amazon EC2 主控台，選取執行個體，接著在詳細資訊窗格中勾選 Public DNS (IPv4) (公有 DNS (IPv4))。) 如果您的執行個體是在 VPC 中，而您找不到公有 DNS 名稱，您必須啟用 DNS 主機名稱。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。
- 確認您的執行個體有公有的 IPv4 地址。如果沒有，您可建立彈性 IP 地址與您執行個體的關聯。如需詳細資訊，請參閱 彈性 IP 位址。
- 若要使用 IPv6 地址連線到您的執行個體，請確認您的本機電腦有 IPv6 地址，且設定使用 IPv6。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的在執行個體上設定 IPv6。
- 確認您的安全群組具有規則，允許在連接埠 3389 上進行 RDP 存取。
- 如果您已複製密碼但收到錯誤 `Your credentials did not work`，請嘗試在提示時手動輸入密碼。有可能是您在複製密碼時少了一個字元或多了一個空白字元。
- 請確認執行個體已通過狀態檢查。如需詳細資訊，請參閱Amazon EC2 執行個體的狀態檢查及the section called “Linux 執行個體狀態檢查失敗”。
- 確認子網路的路由表有路由可將目標 VPC 外的所有流量傳送至 VPC 的網際網路閘道。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 建立自訂路由表 (網際網路閘道)。
- 確認 Windows 防火牆或其他防火牆軟體未封鎖執行個體的 RDP 流量。我們建議您停用 Windows 防火牆，並使用安全群組規則控制對您執行個體的存取。嘗試下列其中一個選項：
  - 使用 AWSSupport-TroubleshootRDP 來disable the Windows Firewall profiles using SSM Agent。此選項需要為執行個體設定 AWS Systems Manager。
  - 請使用 AWSSupport-ExecuteEC2Rescue。
  - 停止執行個體、分離根磁碟區並將磁碟區作為資料磁碟區連接至暫時執行個體。連線至暫時執行個體，並讓磁碟區上線。從資料磁碟區中載入登錄 Hive。在 `HKEY_LOCAL_MACHINE`

`\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile` 下方，將 `EnableFirewall` 設定為 0。卸載登錄 Hive，然後讓磁碟區離線。將磁碟區從暫時執行個體分離，並將其作為根磁碟區連接到原始執行個體。啟動原始執行個體。

- 在不屬於 Active Directory 域的執行個體上，確認網路層級身分驗證已停用 (使用 AWSSupport-TroubleshootRDP 以 disable NLA)。
- 確認遠端桌面服務 (TermService) 啟動類型為「自動」，且服務已啟動 (使用 AWSSupport-TroubleshootRDP 以 enable and start the RDP service)。
- 確認您連線至正確的遠端桌面通訊協定連接埠，預設為 3389 (使用 AWSSupport-TroubleshootRDP 以 read the current RDP port 和 change it back to 3389)。
- 確認您的執行個體上允許遠端桌面連線 (使用 AWSSupport-TroubleshootRDP 以 enable Remote Desktop connections)。
- 請確認密碼尚未過期。如果密碼已過期，您可以重設它。如需詳細資訊，請參閱重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼。
- 如果您嘗試使用在此執行個體上建立的使用者進行連線，但收到錯誤 `The user cannot connect to the server due to insufficient access privileges`，請確認您已授權使用者可在本機登入。如需詳細資訊，請參閱授予成員登入本機的權限。
- 如果您嘗試的同時 RDP 工作階段超過允許上限，您的工作階段會終止並顯示訊息：`Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.`。根據預設，您的執行個體可以有兩個同時 RDP 工作階段。

使用 macOS RDP 用戶端時發生錯誤

如果使用 Microsoft 網站所提供的遠端桌面連線用戶端來連線到 Windows Server 執行個體，可能會出現下列錯誤：

```
Remote Desktop Connection cannot verify the identity of the computer that you want to
 connect to.
```

從 Mac App Store 下載 Microsoft Remote Desktop 應用程式，然後使用該應用程式連線到執行個體。

RDP 顯示黑色螢幕而非桌面

請嘗試下列方法解決此問題：

- 如需其他資訊，請參閱主控台輸出。若要使用 Amazon EC2 主控台取得執行個體的主控台輸出，請選取執行個體，然後選擇 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)、Get system log (取得系統記錄檔)。
- 確認您執行的是最新版的 RDP 用戶端。
- 嘗試 RDP 用戶端的預設設定。
- 如果您使用的是遠端桌面連線，請嘗試使用 `/admin` 選項啟動它，如下所示。

```
mstsc /v:instance /admin
```

- 如果伺服器執行全螢幕應用程式，可能已停止回應。請使用 Ctrl+Shift+Esc 啟動 Windows 任務管理員，然後結束應用程式。
- 如果伺服器過度使用，可能已停止回應。若要使用 Amazon EC2 主控台監控執行個體，請選取該執行個體，然後選取 Monitoring (監控) 索引標籤。如果您需要將執行個體類型變更為較大的大小，請參閱 Amazon EC2 執行個體類型變更。

無法使用非管理員的使用者從遠端登入至執行個體

如果您無法使用非管理員帳戶的使用者從遠端登入 Windows 執行個體，請確定您已授予該使用者從本機登入的權限。請參閱 授予使用者或群組權限，以從本機登入網域中的網域控制器。

使用 對遠端桌面問題進行故障診斷 AWS Systems Manager

您可以使用 AWS Systems Manager 來疑難排解使用 RDP 連線至 Windows 執行個體的問題。

AWSSupport-TroubleshootRDP

AWSSupport-TroubleshootRDP 自動化文件可讓使用者檢查或修改目標執行個體上的常用設定，這些設定會影響遠端桌面通訊協定 (RDP) 連線，例如 RDP 連接埠、網路層身分驗證 (NLA) 和 Windows 防火牆設定檔。根據預設，此文件會讀取和輸出這些設定的值。

AWSSupport-TroubleshootRDP 自動化文件可以與 EC2 執行個體、內部部署執行個體和啟用用於 AWS Systems Manager (受管執行個體) 的虛擬機器 (VM) 一起使用。此外，它也可以與未啟用搭配 Systems Manager 使用的 Windows 伺服器之 EC2 執行個體搭配使用。如需啟用執行個體以搭配 使用的相關資訊 AWS Systems Manager，請參閱《 AWS Systems Manager 使用者指南》中的受管節點。

使用 AWSSupport-TroubleshootRDP 文件進行故障診斷

1. 登入 Systems Manager 主控台。
2. 確認您位於受損執行個體所在的同一區域。
3. 從左側導覽窗格中選擇 Documents (文件)。
4. 在 Owned by Amazon (由 Amazon 所有) 標籤中，在搜尋欄位中輸入 `AWSSupport-TroubleshootRDP`。當 `AWSSupport-TroubleshootRDP` 文件出現時，請選取它。
5. 選擇 Execute automation (執行自動化)。
6. 對於 Execution Mode (執行模式)，選擇 Simple execution (簡易執行)。
7. 對於 Input parameters (輸入參數)、InstanceId，啟用 Show interactive instance picker (顯示互動式執行個體選擇器)。
8. 選擇 Amazon EC2 執行個體。
9. 檢閱範例，然後選擇 Execute (執行)。
10. 若要監控執行進度，對於 Execution status (執行狀態)，等待狀態從 Pending (待定) 變成 Success (成功)。展開輸出以檢視結果。若要檢視個別步驟的輸出，請在 Executed Steps (執行的步驟) 中從 Step ID (步驟 ID) 選擇一個項目。

AWSSupport-TroubleshootRDP 範例

下列範例示範如何使用 AWSSupport-TroubleshootRDP 完成常見的故障診斷任務。您可以使用範例 AWS CLI start-automation-execution命令或提供的 連結 AWS 管理主控台。

Example範例：檢查目前 RDP 狀態

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom" --region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example範例：停用 Windows 防火牆

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=i-1234567890abcdef0, Action=Custom, Firewall=Disable" --
region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example範例：停用網路層級身分驗證

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, NLASettingAction=Disable" --
region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example範例：將 RDP 服務啟動類型設為自動並啟動 RDP 服務

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example範例：還原預設 RDP 連接埠 (3389)

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPPortAction=Modify" --
region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example範例：允許遠端連線

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RemoteConnections=Enable" --
region region_code
```

AWS Systems Manager 主控台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

AWSSupport-ExecuteEC2Rescue 自動化文件會利用 Windows Server 的 EC2Rescue 進行自動排解並還原 EC2 執行個體連線和 RDP 問題。如需詳細資訊，請參閱在無法觸達的執行個體上執行 EC2Rescue 工具。

AWSSupport-ExecuteEC2Rescue 自動化文件需要停止執行個體並重新啟動。Systems Manager 會停止執行個體並建立 Amazon Machine Image (AMI)。存放在執行個體存放磁碟區的資料會遺失。如果您不是使用彈性 IP 地址，則公有 IP 地址會變更。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的在無法連線的執行個體上執行 EC2Rescue 工具。

使用 AWSSupport-ExecuteEC2Rescue 文件進行故障診斷

1. 開啟 Systems Manager 主控台。
2. 確認您位於受損 Amazon EC2 執行個體所在的同一區域。
3. 在導覽面板中，選擇文件。

4. 搜尋並選取 AWSSupport-ExecuteEC2Rescue 文件，然後選擇執行自動化。
5. 在 Execution Mode (執行模式) 中，選擇 Simple execution (簡易執行)。
6. 在 Input parameters (輸入參數)區段的 UnreachableInstanceId 中，輸入無法連線的執行個體的 Amazon EC2 執行個體 ID。
7. (選用) 如果您要收集作業系統日誌以進行 Amazon EC2 執行個體故障診斷，請在 LogDestination 中，輸入 Amazon Simple Storage Service (Amazon S3) 儲存貯體名稱。日誌會自動上傳至指定的儲存貯體。
8. 選擇 Execute (執行)。
9. 若要監控執行進度，請在 Execution status (執行狀態) 中，等待狀態從 Pending (待定) 變成 Success (成功)。展開輸出以檢視結果。若要檢視個別步驟的輸出，請在 Executed Steps (執行的步驟) 中選擇 Step ID (步驟 ID)。

使用遠端登錄在 EC2 執行個體上啟用遠端桌面

如果無法連線的執行個體不是由 AWS Systems Manager Session Manager 管理，則可以使用遠端登錄檔來啟用遠端桌面。

1. 從 EC2 控制台，停止無法連線的執行個體。
2. 將無法連線之執行個體的根磁碟區，連接到相同可用區域中的另一個執行個體以做為儲存磁碟區。如果同一可用區域中沒有可連線的執行個體，請啟動一個執行個體。記下無法連線的執行個體上根磁碟區的裝置名稱。
3. 在可連線的執行個體上，開啟「磁碟管理」。在「命令提示」視窗中執行下列命令，即可執行這項操作。

   ```
   diskmgmt.msc
   ```

4. 開啟來自無法連線執行個體之新連接磁碟區的內容（按一下滑鼠右鍵）選單，然後選擇線上。
5. 開啟 Windows 登錄編輯程式。在「命令提示」視窗中執行下列命令，即可執行這項操作。

   ```
   regedit
   ```

6. 在「登錄編輯程式」中，選擇 HKEY_LOCAL_MACHINE，然後依序選擇檔案、載入 Hive。
7. 選取連接磁碟區的磁碟機，導覽至 \Windows\System32\config\，選取 SYSTEM，然後選擇 Open (開啟)。
8. 對於 Key Name (鍵值名稱)，請輸入 hive 的唯一名稱，並選擇 OK (確定)。

9. 請先備份登錄 hive，再對登錄進行任何變更。

   a. 在「登錄編輯程式」主控台樹狀目錄中，選取您載入的 Hive：HKEY_LOCAL_MACHINE*your-key-name*。

   b. 依序選擇檔案、匯出。

   c. 在「匯出登錄檔案」對話方塊中，選擇您要儲存備份複本的位置，然後在 File name (檔案名稱) 欄位中輸入備份檔案的名稱。

   d. 選擇儲存。

10. 在登錄編輯器中，導覽至 ，HKEY_LOCAL_MACHINE*your key name*\ControlSet001\Control\Terminal Server然後在詳細資訊窗格中開啟 fDenyTSConnections。

11. 在 Edit DWORD (編輯 DWORD) 值方塊中，在 Value data (值資料) 欄位中輸入 0。

12. 選擇 OK (確定)。

> **Note**
>
> 如果 Value data (值資料) 欄位中的值為 1，則執行個體將拒絕遠端桌面連線。值 0 允許遠端桌面連線。

13. 在「登錄編輯程式」中，選擇 HKEY_LOCAL_MACHINE*your-key-name*，然後依序選擇檔案、卸載 Hive。

14. 關閉「「登錄編輯程式」」和「磁碟管理」。

15. 在 EC2 主控台中，將磁碟區與可連線的執行個體中斷連接，然後再重新連接到無法連線的執行個體。將磁碟區連接到無法連線的執行個體時，請在裝置欄位中輸入您稍早儲存的裝置名稱。

16. 重新啟動無法連線的執行個體。

我的私有金鑰遺失。如何連接到我的 Windows 執行個體？

當您連線到新啟動的 Windows 執行個體時，使用您在啟動執行個體時所指定金鑰對中的私有金鑰，將管理員帳戶的密碼解密。

如果您遺失了管理員密碼，而且也不再擁有私有金鑰，則必須重設密碼或建立新的執行個體。如需詳細資訊，請參閱重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼。如需使用 Systems Manager 重設密碼之步驟的文件，請參閱「AWS Systems Manager 使用者指南」中的在 EC2 執行個體上重設密碼和 SSH 金鑰。

對 Amazon EC2 Windows 執行個體啟動問題進行疑難排解

以下疑難排解秘訣有助您解決 Amazon EC2 Windows Server 執行個體的密碼和啟用問題。

問題

- 「未提供密碼」
- 「尚未提供密碼」
- 「無法擷取 Windows 密碼」
- 等待中繼資料服務
- 無法啟動 Windows
- Windows 不是正版 (0x80070005)
- 無終端機伺服器的授權伺服器可提供授權
- 「部分設定是由您的組織所管理」

「未提供密碼」

若要使用遠端桌面連線到 Windows 執行個體，您必須指定帳戶和密碼。提供的帳戶和密碼是以您啟動執行個體所用的 AMI 為基礎。您可擷取管理員帳戶自動產生的密碼，或使用 AMI 建立來源之原始執行個體中使用的帳戶和密碼。

您可以為使用自訂 Windows AMI 啟動的執行個體產生管理員帳戶密碼。要產生密碼，您需要在建立 AMI 之前在作業系統中配置一些設定。如需詳細資訊，請參閱建立 Amazon EBS 支援的 AMI。

如果您的 Windows 執行個體未設定產生隨機密碼，當您使用主控台擷取自動產生的密碼時，就會收到以下訊息：

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed. A
password cannot be retrieved for this instance. If you have forgotten your password,
 you can
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

檢查執行個體的主控台輸出，查看您啟動執行個體所用的 AMI，建立時是否停用產生密碼。如果停用產生密碼，主控台輸出會包含以下內容：

```
Ec2SetPassword: Disabled
```

如果停用產生密碼而您不記得原始執行個體的密碼，您可重設此執行個體的密碼。如需詳細資訊，請參閱重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼。

「尚未提供密碼」

若要使用遠端桌面連線到 Windows 執行個體，您必須指定帳戶和密碼。提供的帳戶和密碼是以您啟動執行個體所用的 AMI 為基礎。您可擷取管理員帳戶自動產生的密碼，或使用 AMI 建立來源之原始執行個體中使用的帳戶和密碼。

您應該會在幾分鐘內收到密碼。如果未提供密碼，當您使用主控台擷取自動產生的密碼時，就會收到以下訊息：

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to retrieve
 the
auto-generated password.
```

如果超過四分鐘仍未收到密碼，則可能是執行個體的啟動代理程式未設定為產生密碼。請檢查主控台輸出是否為空以確認。如需詳細資訊，請參閱無法取得主控台輸出。

同時驗證用於存取 Management Portal 的 AWS Identity and Access Management (IAM) 帳戶是否允許 `ec2:GetPasswordData`動作。如需 IAM 許可的詳細資訊，請參閱 什麼是 IAM？

「無法擷取 Windows 密碼」

若要擷取管理員帳戶的自動產生密碼，您必須使用啟動執行個體時所指定之金鑰對的私有金鑰。如果您在啟動執行個體時未指定金鑰對，就會收到下列訊息。

```
Cannot retrieve Windows password
```

您可以使用相同的 AMI，確定指定金鑰對，終止此執行個體並啟動新的執行個體。

等待中繼資料服務

Windows 執行個體必須先從自己的執行個體中繼資料取得資訊，才能自行啟動。根據預設，`WaitForMetaDataAvailable` 設定會確定 EC2Config 服務會先等待執行個體中繼資料變成可存取，才繼續開機程序。如需詳細資訊，請參閱使用執行個體中繼資料管理您的 EC2 執行個體。

如果執行個體的執行個體連線能力測試失敗，請嘗試下列方法解決此問題。

- 檢查您 VPC 的 CIDR 區塊。如果在 IP 地址範圍由 224.0.0.0 到 255.255.255.255 (類別 D 和類別 E IP 地址範圍) 的 VPC 中啟動執行個體，Windows 執行個體即無法正確開機。這些是保留的 IP 地址範圍，不應該指派給主機裝置。我們建議您使用來自私有 (非可公開路由) IP 地址範圍 (如 RFC 1918 中所指定) 的 CIDR 區塊來建立 VPC。
- 系統可能已以靜態 IP 地址設定。請嘗試建立網路介面並連接至執行個體。
- 在您無法連線之 Windows 執行個體上啟用 DHCP

1. 停止受影響的執行個體並分離其根磁碟區。
2. 啟動與受影響執行個體同一可用區域中的暫時執行個體。

> **Warning**
>
> 如果您的暫時執行個體與原始執行個體以相同的 AMI 為基礎，則您必須完成額外的步驟，否則您無法在還原原始執行個體磁碟區之後將其啟動，原因是磁碟簽名衝突。或者，請為暫時執行個體選擇不同的 AMI。例如，如果原始執行個體使用 AWS Windows Server 2016 的 Windows AMI，請使用 AWS Windows Server 2019 的 Windows AMI 啟動暫時執行個體。

3. 從受影響的執行個體將根磁碟區連接到此暫時執行個體。連線至暫時執行個體、開啟 Disk Management (磁碟管理) 公用程式，然後讓磁碟機上線。
4. 從暫時執行個體，開啟 Regedit 並選取 HKEY_LOCAL_MACHINE。從 File (檔案) 功能表中，選擇 Load Hive (載入 Hive)。選取磁碟機、開啟檔案 Windows\System32\config\SYSTEM，然後在提示時指定金鑰名稱 (您可使用任何名稱)。
5. 選取剛載入的機碼並導覽至 ControlSet001\Services\Tcpip\Parameters\Interfaces。每個網路介面均按 GUID 列出。選取正確的網路介面。如已停用 DHCP 並指派了靜態 IP 地址，EnableDHCP 設為 0。若要啟用 DHCP，請將 EnableDHCP 設為 1，然後刪除下列機碼，如果它們存在：NameServer、SubnetMask、IPAddress 和 DefaultGateway。再次選取該金鑰，然後從 File (檔案) 功能表中選擇 Unload Hive (取消載入 Hive)。

> **Note**
>
> 如有多個網路介面，您需要找出正確的介面來啟用 DHCP。若要找出正確的網路界面，請檢閱下列機碼值：NameServer、SubnetMask、IPAddress 和 DefaultGateway。這些值會顯示前一個執行個體的靜態組態。

6. (選用) 如已啟用 DHCP，您可能沒有中繼資料服務的路由。更新 EC2Config 可解決此問題。

   a. 下載並安裝最新版的 EC2Config 服務。如需安裝此服務的詳細資訊，請參閱 the section called “安裝 EC2Config”。

   b. 將 .zip 檔案中的檔案解壓縮到所連接之磁碟機上的 Temp 目錄中。

   c. 開啟 Regedit 並選取 HKEY_LOCAL_MACHINE。從 File (檔案) 功能表中，選擇 Load Hive (載入 Hive)。選取磁碟機、開啟檔案 Windows\System32\config\SOFTWARE，然後在提示時指定金鑰名稱 (您可使用任何名稱)。

   d. 選取剛載入的機碼並導覽至 Microsoft\Windows\CurrentVersion。選取 RunOnce 機碼。(如果此機碼不存在，請在 CurrentVersion 按一下滑鼠右鍵、指向 New (新增)、選取 Key (機碼) 並命名機碼 RunOnce。) 開啟內容 （ 按一下滑鼠右鍵 ） 選單，指向新增，然後選取字串值。在名稱輸入 Ec2Install，在資料輸入 C:\Temp\Ec2Install.exe -q。

   e. 再次選取該金鑰，然後從 File (檔案) 功能表中選擇 Unload Hive (取消載入 Hive)。

7. (選用) 如果您的暫時執行個體與原始執行個體以相同的 AMI 為基礎，則您必須完成下列步驟，否則您無法在還原原始執行個體磁碟區之後將其啟動，原因是磁碟簽名衝突。

> **Warning**
>
> 下列程序說明如何使用登錄編輯程式來編輯 Windows 登錄檔。如果您不熟悉 Windows 登錄檔或如何使用登錄編輯程式來安全地進行變更，請參閱 設定登錄檔。

   a. 開啟命令提示，輸入 regedit.exe，然後按 Enter。

   b. 在登錄編輯程式中，從內容功能表中選擇 HKEY_LOCAL_MACHINE (按一下右鍵)，然後選擇 Find (尋找)。

   c. 輸入 Windows Boot Manager，然後選擇 Find Next (尋找下一個)。

   d. 選擇名為 11000001 的金鑰。此金鑰是您在上一個步驟中找到的金鑰的同級。

   e. 在右窗格中，選擇 Element，然後從內容功能表中選擇 Modify (修改)(按一下右鍵)。

f. 在資料中的位移為 0x38 處找到四位元組的磁碟簽章。反轉位元組以建立磁碟簽章，然後寫下來。例如，下列資料所代表的磁碟簽章是 E9EB3AA5：

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

g. 在命令提示視窗中，執行下列命令以啟動 Microsoft DiskPart。

```
diskpart
```

h. 執行下列 DiskPart 命令以選取磁碟區。(可以使用磁碟管理公用程式，確認磁碟編號為 1)。

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. 執行下列 DiskPart 命令即可取得磁碟簽章。

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. 如果上一個步驟中顯示的磁碟簽章不符合您先前寫下的 BCD 中的磁碟簽章，請使用下列 DiskPart 命令來變更磁碟簽章，使其符合：

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. 使用 Disk Management (磁碟管理) 公用程式讓磁碟機離線。

> **Note**
>
> 如果暫時執行個體執行的作業系統與受影響執行個體的相同，磁碟機會自動離線，因此您不需要手動將其離線。

9. 分離磁碟區與暫時執行個體。如果不再需要使用此暫時執行個體，您可終止它。
10. 將磁碟區連接為 /dev/sda1 來還原受影響執行個體的根磁碟區。
11. 啟動受影響的執行個體。

如果您已連線到此執行個體，請由此執行個體開啟網際網路瀏覽器，然後輸入下列中繼資料伺服器的 URL：

```
http://169.254.169.254/latest/meta-data/
```

如果無法聯絡中繼資料伺服器，請嘗試下列方法解決問題：

- 下載並安裝最新版的 EC2Config 服務。如需安裝此服務的詳細資訊，請參閱 the section called “安裝 EC2Config”。
- 檢查 Windows 執行個體執行的是否為 Red Hat PV 驅動程式。如果是，請更新至 Citrix PV 驅動程式。如需詳細資訊，請參閱the section called “升級 PV 驅動程式”。
- 確認防火牆、IPSec 和代理伺服器設定不會封鎖中繼資料服務 (169.254.169.254) 或 AWS KMS 伺服器的傳出流量 (地址指定在 TargetKMSServer 的 C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml 元素中)。
- 使用下列命令確認您有中繼資料服務的路由 (169.254.169.254)。

  ```
  route print
  ```

- 檢查是否有可能影響您執行個體可用區域的網路問題。請前往 http://status.aws.amazon.com/。

無法啟動 Windows

Windows 執行個體使用 Windows AWS KMS 啟用。如果您的執行個體無法連線 AWS KMS 伺服器A problem occurred when Windows tried to activate. Error Code 0xC004F074，您可以收到此訊息：。Windows 必須每 180 天啟動一次。EC2Config 會在啟用期間到期之前嘗試聯絡 AWS KMS 伺服器，以確保 Windows 保持啟用狀態。

如果您發生 Windows 啟用問題，請使用下列程序予以解決。

EC2Config (Windows Server 2012 R2 AMI 及舊版)

1. 下載並安裝最新版的 EC2Config 服務。如需安裝此服務的詳細資訊，請參閱 the section called “安裝 EC2Config”。
2. 登入執行個體並開啟下列檔案：C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml。
3. 在 config.xml 檔案中找到 Ec2WindowsActivate 外掛程式。將狀態變更為 Enabled (已啟用) 並儲存變更。

4. 在 Windows 服務嵌入式管理單元中，重新啟動 EC2Config 服務或重新開機執行個體。

如果這無法解決啟用問題，請遵循下列其他步驟。

1. 設定 AWS KMS 目標： C:\> slmgr.vbs /skms 169.254.169.250:1688
2. 啟動 Windows：C:\> slmgr.vbs /ato

EC2Launch (Windows Server 2016 AMI 及更新版本)

1. 從具有管理權限的 PowerShell 提示字元中，匯入 EC2Launch 模組：

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1"
```

2. 呼叫新增路由函數以查看新路由清單：

```
PS C:\> Add-Routes
```

3. 呼叫 Set-ActivationSettings 函數：

```
PS C:\> Set-Activationsettings
```

4. 接著執行下列指令碼以啟動 Windows：

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

針對 EC2Config 和 EC2Launch，如果您仍收到啟用錯誤，請確認下列資訊。

- 確認您有通往 AWS KMS 伺服器的路由。開啟 `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` 並找到 `TargetKMSServer` 元素。執行下列命令，並檢查這些 AWS KMS 伺服器的地址是否已列出。

```
route print
```

- 確認已設定 AWS KMS 用戶端金鑰。執行下列命令並檢查輸出。

```
C:\Windows\System32\slmgr.vbs /dlv
```

如果輸出包含錯誤：找不到產品金鑰，則不會設定 AWS KMS 用戶端金鑰。如果未設定 AWS KMS 用戶端金鑰，請如本 Microsoft 文章所述查詢用戶端金鑰：AWS KMS 用戶端啟用和產品金鑰，然後執行下列命令來設定 AWS KMS 用戶端金鑰。

```
C:\Windows\System32\slmgr.vbs /ipk client_key
```

- 確認系統的時間和時區是否正確。如果您使用的時區不是 UTC，請新增下列登錄機碼並將其設定為 1，以確保時間正確：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal。
- 如已啟用 Windows 防火牆，請使用下列命令暫時停用它。

```
netsh advfirewall set allprofiles state off
```

Windows 不是正版 (0x80070005)

Windows 執行個體使用 Windows AWS KMS 啟用。如果執行個體無法完成啟用程序，會回報所用 Windows 不是正版。

請嘗試無法啟動 Windows的建議。

無終端機伺服器的授權伺服器可提供授權

Windows Server 預設透過遠端桌面可同時授權兩名使用者使用。如果您需要透過遠端桌面提供兩位以上的使用者同時存取您的 Windows 執行個體，您可購買遠端桌面服務客戶存取授權 (CAL) 並安裝遠端桌面工作階段主機和遠端桌面工作階段授權伺服器角色。

檢查是否有下列問題：

- 您已超過可同時執行的 RDP 工作階段上限。
- 您已安裝 Windows 遠端桌面服務角色。
- 授權已過期。如果授權已過期，您就無法以使用者身分連線到您的 Windows 執行個體。您可以嘗試下列方法：
  - 使用 /admin 參數從命令列連線至執行個體，例如：

    ```
    mstsc /v:instance /admin
    ```

    如需詳細資訊，請參閱下列 Microsoft 文章：Access Remote Desktop Via Command Line。

- 停止執行個體、分離其 Amazon EBS 磁碟區，然後將它們連接到同一可用區域的另一個執行個體，以復原您的資料。

「部分設定是由您的組織所管理」

從最新 Windows Server AMI 啟動的執行個體可能顯示 Windows Update 對話方塊訊息，其中指出「部分設定是由您的組織所管理」。此訊息是因為 Windows Server 中的變更而出現，不影響 Windows Update 的行為或造成您無法管理更新設定。

如何移除警告

1. 開啟 `gpedit.msc` 並導覽至電腦組態 > 系統管理範本 > Windows 元件 > Windows 更新。編輯設定自動更新，然後將其設為已啟用。
2. 在命令提示字元中，使用 gpupdate /force 更新群組政策。
3. 關閉「Windows Update 設定」再重新開啟。您將看到上述訊息表示您的設定正由您的組織所管理，後面接著「我們將自動下載更新，付費連線的情況除外 (因為可能需要支付費用)。在該種情況下，我們會自動下載保持 Windows 順暢運作所需的這些更新。」
4. 返回 `gpedit.msc`，然後將群組政策設回尚未設定。再次執行 gpupdate /force。
5. 關閉命令提示字元並等待幾分鐘。
6. 重新開啟「Windows Update 設定」。您應該不會看到「部分設定是由您的組織所管理」這個訊息。

針對 Amazon EC2 Windows 執行個體的問題進行疑難排解

下列的疑難排解秘訣可協助您解決 Amazon EC2 Windows 執行個體的相關問題。

問題

- 無法將 AWS Systems Manager Sessions Manager 連線至 Windows Server 2025 執行個體
- EBS 磁碟區不是在 Windows Server 2016 和 2019 和更新版本上初始化
- 在目錄服務還原模式 (DSRM) 中開機 EC2 Windows 執行個體
- 執行個體遺失網路連線能力或排程任務未如預期執行
- 無法取得主控台輸出
- 網路不提供 Windows Server 2012 R2

- 磁碟簽章衝突

無法將 AWS Systems Manager Sessions Manager 連線至 Windows Server 2025 執行個體

您可能會遇到將 AWS Systems Manager Sessions Manager 連線至 Windows Server 2025 執行個體的問題。若要解決此問題，請登入執行個體，然後導覽至 `Settings > Apps > Optional Features`，再新增 `WMIC`。重新啟動 SSM Agent 服務或將執行個體重新開機，且 Sessions Manager 應連線。

您也可使用以下 PowerShell 命令來執行相同動作：

```
Start-Process -FilePath "$env:SystemRoot\system32\Dism.exe" -ArgumentList @('/
Online', '/Add-Capability', '/CapabilityName:WMIC~~~~') -Wait; Restart-Service -Name
 AmazonSSMAgent
```

EBS 磁碟區不是在 Windows Server 2016 和 2019 和更新版本上初始化

從適用於 Windows Server 2016 和 2019 和更新版本的 Amazon Machine Image (AMI) 建立的執行個體使用 EC2Launch v1 代理程式處理各種啟動任務，包括初始化 EBS 磁碟區。根據預設，EC2Launch v1 不會初始化輔助磁碟區。然而，您可以設定 EC2Launch v1 自動初始化這些磁碟，如下所示。

將磁碟機代號映射到磁碟區

1. 連線至執行個體，以設定 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` 檔案並在文字編輯器中開啟。
2. 指定磁碟區設定如下：

   ```
   {
   "driveLetterMapping": [
    {
      "volumeName": "sample volume",
      "driveLetter": "H"
    }]
   }
   ```

3. 儲存您的變更並結束檔案。
4. 開啟 Windows PowerShell，並使用下列命令執行初始化磁碟的 EC2Launch v1 指令碼：

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

若要在每次執行個體開機時初始化磁碟，請新增 `-Schedule` 標記，如下所示：

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

EC2Launch v1 代理程式可以執行執行個體初始化指令碼，例如與 `InitializeInstance.ps1` 指令碼平行處理的 `initializeDisks.ps1`。如果 `InitializeInstance.ps1` 指令碼會重新啟動執行個體，它可能會中斷執行個體啟動時執行的其他排程任務。為了避免任何潛在的衝突，我們建議您將邏輯新增至 `initializeDisks.ps1` 指令碼，以確保執行個體初始化已先完成。

> **Note**
>
> 如果 EC2Launch 指令碼未初始化磁碟區，請確定磁碟區已上線。如果磁碟區離線，請執行下列命令以使所有磁碟上線。
>
> ```
> PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
>  $False
> ```

在目錄服務還原模式 (DSRM) 中開機 EC2 Windows 執行個體

如果執行 Microsoft Active Directory 的執行個體發生系統故障或其他重大問題，您可在稱為目錄服務還原模式(DSRM) 的特殊版本安全模式中開機此執行個體，執行故障診斷。在 DSRM 中，您可修復或復原 Active Directory。

DSRM 的磁碟機支援

啟用 DSRM 和在執行個體中開機的方式取決於執行個體執行的磁碟機。在 EC2 主控台中，您可在系統日誌中檢閱執行個體的磁碟機版本詳細資訊。下表顯示 DSRM 支援的驅動程式。

驅動程式版本	是否支援 DSRM？	後續步驟
Citrix PV 5.9	否	從備份還原執行個體。您無法啟用 DSRM。

驅動程式版本	是否支援 DSRM？	後續步驟
AWS PV 7.2.0	否	雖然此驅動程式不支援 DSRM，但您仍然可以分離根磁碟區與無法連線的執行個體、拍攝磁碟區快照或從磁碟區建立 AMI，然後將它連接到同一可用區域的另一個執行個體做為輔助磁碟區。然後即可啟用 DSRM (如本節中所述)。
AWS PV 7.2.2 及更新版本	是	分離根磁碟區、將它連接到其他執行個體，然後啟用 DSRM (如本節中所述)。
增強型聯網	是	分離根磁碟區、將它連接到其他執行個體，然後啟用 DSRM (如本節中所述)。

如需如何啟用增強型聯網的詳細資訊，請參閱 the section called “彈性網路介面卡 (ENA)”。如需有關升級 AWS PV 驅動程式的資訊，請參閱升級 Windows 執行個體上的 PV 驅動程式。

設定執行個體在 DSRM 中開機

EC2 Windows 執行個體在作業系統執行前沒有網路連線。因此，您無法按鍵盤的 F8 按鈕選取開機選項。您必須使用下列其中一個程序，在 DSRM 中開機 EC2 Windows Server 執行個體。

如果您懷疑 Active Directory 已毀損，但執行個體仍在執行，您可設定此執行個體使用 System Configuration (系統組態) 對話方塊或命令提示在 DSRM 中開機。

使用 System Configuration (系統組態) 對話方塊在 DSRM 中開機線上執行個體

1. 在 Run (執行) 對話方塊中輸入 `msconfig`，然後按 Enter。
2. 選擇 Boot (開機) 標籤。
3. 在 Boot options (開機選項) 下選擇 Safe boot (安全開機)。
4. 選擇 Active Directory repair (Active Directory 修復)，然後選擇 OK (確定)。系統會提示您重新開機伺服器。

使用命令列在 DSRM 中開機線上執行個體

從命令提示視窗執行下列命令：

```
bcdedit /set safeboot dsrepair
```

如果執行個體離線且無法連線，您必須分離根磁碟區，然後將它連接到其他執行個體，以啟用 DSRM 模式。

在 DSRM 中開機離線執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 找出並選取受影響的執行個體。選擇 Instance state (執行個體狀態)、Stop instance (停止執行個體)。
4. 選擇 Launch instances (啟動執行個體)，並在與受影響執行個體相同的可用區域中，建立暫時性的執行個體。選擇使用不同 Windows 版本的執行個體類型。例如，如果您的執行個體是 Windows Server 2016，則選取 Windows Server 2019 執行個體。

   > **Important**
   >
   > 若您沒有在與受影響的執行個體相同的可用區域內建立執行個體，您將無法將受影響執行個體的根磁碟區連接到新的執行個體。

5. 在導覽窗格中，選擇 Volumes (磁碟區)。
6. 尋找受影響執行個體的根磁碟區。分離磁碟區，再將其連接至先前建立的暫時性執行個體。使用預設裝置名稱 (xvdf) 連接它。
7. 使用遠端桌面連接至暫時性執行個體，然後使用 Disk Management 公用程式將磁碟區變成可用。
8. 開啟命令提示並執行下列命令。使用您剛連接之輔助磁碟區的實際磁碟機代號取代 D：

   ```
   bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
   ```

9. 在磁碟管理公用程式中，選擇先前連接的磁碟機，開啟內容 (按右鍵) 選單，然後選擇 Offline (離線)。
10. 在 EC2 主控台中，將受影響的磁碟區與暫時性執行個體分離，然後將其重新連接至裝置名稱為 `/dev/sda1` 的執行個體。您必須指定此裝置名稱，將磁碟區指定為根磁碟區。
11. 啟動實例。
12. 在執行個體通過 EC2 主控台的運作狀態檢查後，使用遠端桌面連線到執行個體並確認它在 DSRM 模式中開機。
13. (選用) 刪除或停止您在此程序中建立的暫時性執行個體。

執行個體遺失網路連線能力或排程任務未如預期執行

如果您重新啟動您的執行個體但它遺失網路連線能力，可能是執行個體的時間發生錯誤。

Windows 執行個體預設使用國際標準時間 (UTC)。如果您將執行個體的時間設至不同的時區然後重新啟動，就會發生時間位移，且執行個體會暫時遺失 IP 地址。執行個體最後會在重新取得網路連線能力，但這可能需要數小時。執行個體重新取得網路連線能力所需要的時間視 UTC 和其他時間之間的差異而定。

此相同的時間問題也會導致排程任務未如預期執行。在本例中，因為執行個體的時間不正確，所以排程的任務未如預期執行。

若要持續使用 UTC 以外的時區，您必須設定 RealTimeIsUniversal 登錄機碼。沒有這個機碼，執行個體會在重新啟動後使用 UTC。

解決造成網路連線能力遺失的時間問題

1. 請務必執行建議的 PV 驅動程式。如需詳細資訊，請參閱the section called “升級 PV 驅動程式”。
2. 確認以下登錄機碼存在且設為 1：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal

無法取得主控台輸出

若為 Windows 執行個體，執行個體主控台會顯示在 Windows 開機程序期間執行的任務輸出。如果 Windows 成功開機，最後記錄的訊息為 `Windows is Ready to use`。您也可以在主控台中顯示事件日誌訊息，但根據您的 Windows 版本不同，系統可能無法預設啟用此功能。如需詳細資訊，請參閱the section called “Windows 啟動代理程式”。

若要使用 Amazon EC2 主控台取得執行個體的主控台輸出，請選取執行個體，然後選擇 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)、Get system log (取得系統記錄檔)。若要使用命令列取得主控台輸出，請使用下列其中一項命令：get-console-output (AWS CLI) 或 Get-EC2ConsoleOutput (AWS Tools for Windows PowerShell)。

若是執行 Windows Server 2012 R2 或更舊版本的執行個體，如果主控台輸出是空的，可能表示 EC2Config 服務發生問題，例如設定錯誤的組態檔或 Windows 無法正確開機。若要修復此問題，請下載並安裝最新版的 EC2Config。如需詳細資訊，請參閱the section called “安裝 EC2Config”。

網路不提供 Windows Server 2012 R2

如需有關對無法在網路上使用之 Windows Server 2012 R2 執行個體進行疑難排解的資訊，請參閱 Windows Server 2012 R2 會在執行個體重新開機後失去網路和儲存連線。

磁碟簽章衝突

您可以使用 EC2Rescue for Windows Server，檢查並解決磁碟簽章衝突。或者，您可以執行下列步驟，手動解決磁碟簽章問題。

> **Warning**
>
> 下列程序說明如何使用登錄編輯程式來編輯 Windows 登錄檔。如果您不熟悉 Windows 登錄檔或如何使用登錄編輯程式來安全地進行變更，請參閱 設定登錄檔。

1. 開啟命令提示，輸入 regedit.exe，然後按 Enter。
2. 在登錄編輯程式中，從內容功能表中選擇 HKEY_LOCAL_MACHINE (按一下右鍵)，然後選擇 Find (尋找)。
3. 輸入 Windows Boot Manager，然後選擇 Find Next (尋找下一個)。
4. 選擇名為 `11000001` 的金鑰。此金鑰是您在上一個步驟中找到的金鑰的同級。
5. 在右窗格中，選擇 `Element`，然後從內容功能表中選擇 Modify (修改)(按一下右鍵)。
6. 在資料中的位移為 0x38 處找到四位元組的磁碟簽章。這是開機組態資料庫簽章 (BCD)。反轉位元組以建立磁碟簽章，然後寫下來。例如，下列資料所代表的磁碟簽章是 `E9EB3AA5`：

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

7. 在命令提示視窗中，執行下列命令以啟動 Microsoft DiskPart。

```
diskpart
```

8. 執行 `select disk` DiskPart 命令，並指定發生磁碟簽章衝突的磁碟區的磁碟編號。

> **Tip**
>
> 若要查看發生磁碟簽章衝突的磁碟區的磁碟編號，請使用磁碟管理公用程式。開啟命令提示，輸入 compmgmt.msc，然後按 Enter。在左側導覽面板中，開啟磁碟管理。在磁碟管理公用程式中，查看發生磁碟簽章衝突的離線磁碟區的磁碟編號。

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. 執行下列 DiskPart 命令即可取得磁碟簽章。

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. 如果上一個步驟中顯示的磁碟簽章不符合您先前寫下的磁碟簽章，請使用下列 DiskPart 命令來變更磁碟簽章，使其相符：

```
DISKPART> uniqueid disk id=E9EB3AA5
```

透過網路對 Windows 執行個體進行核心除錯

KDNET 彈性網路轉接器擴充性模組 (ENA) 是一種硬體驅動程式支援層，可透過 Amazon Elastic Compute Cloud 執行個體上的 ENA 透過網路進行 Windows 核心偵錯。您可以使用擴充模組搭配 Windows Debugger (WinDbg)，在執行 Windows 的 EC2 執行個體上執行核心層級偵錯。

核心偵錯可協助您診斷和疑難排解低階作業系統問題，例如 EC2 Windows 執行個體上的藍螢幕錯誤 (BSODs)、驅動程式故障和開機問題。

目錄

- 先決條件
- 步驟 1：在偵錯主機上安裝 Windows 偵錯工具
- 步驟 2：設定偵錯目標
- 步驟 3：在偵錯主機上啟動偵錯工作階段
- 步驟 4：重新啟動偵錯目標

- 偵錯後清除偵錯設定
- 限制
- 其他備註

先決條件

開始前，請確定您具有下列項目：

兩個 EC2 Windows 執行個體，位於相同的子網路中：

- 除錯主機執行個體 — 執行 Windows Debugger (WinDbg)。
- 偵錯目標執行個體：您要偵錯的執行個體。

如需啟動執行個體的詳細資訊，請參閱 開始教學課程。

主機和目標執行個體的安全群組必須允許用於 KDNET 除錯的連接埠上的傳入和傳出 UDP 流量（建議範圍：50000–50039)。設定此項目最簡單的方法是建立具有傳入規則的安全群組，允許來自自身的 UDP 流量做為來源，然後將該安全群組連接至這兩個執行個體。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的安全群組規則。

除錯目標執行個體必須執行下列其中一個 Windows 版本（或更新版本）：

- 建置編號為 26100.7462 的 Windows Server 2025 (2025 年 12 月修補程式)
- Windows 11 24H2，建置編號為 26100.7309
- Windows 11 25H2，建置編號為 26200.7309

> **Note**
>
> ENA 的 KDNET 擴充性模組是做為 Windows 的一部分分佈，只能透過 Windows 每月累積更新進行更新。我們建議您將最新的 Windows KB 安裝在偵錯目標上，以確保您擁有最新版本的 Windows KB。
> 若要驗證偵錯目標上是否存在模組，請在提升的 PowerShell 工作階段中執行下列命令：
>
> ```
> Test-Path C:\Windows\system32\kd_02_1d0f.dll
> ```
>
> 如果命令傳回 `True`，表示模組可用。

步驟 1：在偵錯主機上安裝 Windows 偵錯工具

在 PowerShell 工作階段中執行下列命令，在偵錯主機執行個體上安裝 Windows 偵錯工具：

```
winget install microsoft.windbg
```

如需詳細安裝說明，請參閱 Microsoft 文件中的安裝 Windows Debugger。

安裝之後，請在 PowerShell 工作階段中執行下列命令，確認偵錯工具是否正常運作：

```
windbgx
```

WinDbg 視窗應開啟。如果安裝成功，您可以關閉視窗。

步驟 2：設定偵錯目標

> **Note**
>
> 當核心偵錯作用中時，用於偵錯工作階段的 ENA 裝置專用於偵錯工具流量。如果您需要在偵錯期間維持偵錯目標執行個體上的網際網路存取，請在開始之前將第二個 ENA 連接到執行個體。

在偵錯目標上，開啟提升的 PowerShell 工作階段，並使用下列步驟設定核心偵錯。

執行下列命令，列出連接至執行個體之 ENA 轉接器的匯流排、裝置和函數編號：

```
Get-NetAdapter -Physical |
    Where-Object -Property PnPDeviceID -Match -Value '^PCI\\VEN_1D0F&DEV_EC2[01]&' |
    Get-NetAdapterHardwareInfo |
    Select-Object InterfaceDescription, BusNumber, DeviceNumber, FunctionNumber |
    Format-List
```

如果多個 ENA 轉接器連接到執行個體

如果執行個體連接多個 ENA 轉接器，請使用下列命令，將每個實體轉接器映射至其私有 IP 地址。您可以在 EC2 → 執行個體 → 【執行個體 ID】 → 網路 → 網路界面 AWS 管理主控台 下的 中交叉參考這些詳細資訊。這可協助您將特定網路介面 IDs、私有 IPs和安全群組與 OS 層級轉接器建立關聯，以進行目標偵錯。

```
Get-NetAdapter -Physical |
    Where-Object PnPDeviceID -Match '^PCI\\VEN_1D0F&DEV_EC2[01]&' |
    ForEach-Object {
        $adapter = $_
        $hwInfo = $adapter | Get-NetAdapterHardwareInfo
        $ipInfo = Get-NetIPAddress -InterfaceIndex $adapter.InterfaceIndex -
AddressFamily IPv4
        [PSCustomObject]@{
            InterfaceDescription = $adapter.InterfaceDescription
            IPAddress            = $ipInfo.IPAddress
            BusNumber            = $hwInfo.BusNumber
            DeviceNumber         = $hwInfo.DeviceNumber
            FunctionNumber       = $hwInfo.FunctionNumber
        }
    } | Format-List
```

請注意用於從輸出除錯的 ENA 轉接器的 `BusNumber`、`DeviceNumber` 和 `FunctionNumber` 值。

執行下列命令來設定核心偵錯。將預留位置值取代為您的特定組態：

```
bcdedit /debug on
bcdedit /set loadoptions FORCEHVTONOTSHAREDEBUGDEVICE
bcdedit /dbgsettings net hostip:host-private-ip port:port-number key:encryption-key
 busparams:b.d.f
```

> **Note**
>
> 執行下列命令`loadoptions`來檢查是否存在 。如果傳回值，請複製該字串並附加`;FORCEHVTONOTSHAREDEBUGDEVICE`至該字串。
>
> ```
> (bcdedit /enum) -match "loadoptions"
> ```

其中：

- *host-private-ip* — 除錯主機執行個體的私有 IPv4 地址。如果您的執行個體是在IPv6-only的子網路中啟動，請參閱 Microsoft 文件中的使用 IPv6 設定 KDNET，以進一步了解搭配 KDNET 使用 IPv6。
- *port-number* — 用於偵錯工作階段的連接埠。建議範圍為 50000–50039（例如 `50000`）。

- *encryption-key* — 用於加密偵錯連線的 256 位元金鑰，指定為以句點分隔的四個 64 位元值。每個 64 位元值的長度上限為 13 個字元，只使用小寫字母 a–z 和數字 0–9。不允許使用特殊字元。加密金鑰範例：1kdnet2keys3.4kdnet5keys6.7kdnet8keys9.10kdnet11ke。
- *b.d.f* — ENA 裝置的匯流排、裝置和函數編號，格式為 bus.device.function（例如 0.3.0），用於偵錯。

> **Tip**
>
> 若要偵錯 Windows 開機程序，請同時執行下列命令：
>
> ```
> bcdedit /bootdebug on
> ```

步驟 3：在偵錯主機上啟動偵錯工作階段

若要允許偵錯主機上的流量，您可以為 WinDbg 應用程式或特定 UDP 連接埠建立防火牆規則。

選項 1：允許 WinDbg 應用程式

執行下列命令來授權 WinDbg 可執行檔：

```
$WinDbgxPath = "$env:LocalAppData\Microsoft\WindowsApps\WinDbgX.exe"
New-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection" -Action Allow -
Program $WinDbgxPath
```

選項 2：允許特定的 UDP 連接埠

或者，您可以允許傳入 UDP 流量到為核心偵錯設定的連接埠。以您選擇的 KDNET 連接埠取代 *port-number*：

```
$DebugPort = port-number
New-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection" -Direction Inbound -
LocalPort $DebugPort -Protocol UDP -Action Allow
```

> **Note**
>
> 防火牆組態可能受到網域群組政策 (GPO) 的限制，或需要更高的管理員許可。如果命令失敗，請聯絡您的網路管理員。

使用符合偵錯目標組態的連接埠和金鑰啟動 WinDbg。您可以指定 Microsoft WinDbg 命令列參考中記錄的其他核心偵錯選項，以符合您的使用案例。

```
windbgx -k net:port=port-number,key=encryption-key
```

WinDbg 會開啟並等待偵錯目標連線。

步驟 4：重新啟動偵錯目標

在偵錯目標上，重新啟動執行個體以啟動偵錯連線：

```
shutdown -r -t 0
```

偵錯目標重新啟動後，偵錯主機上的 WinDbg 會自動連線到目標。您現在可以使用 WinDbg 來檢查核心狀態、設定中斷點，以及診斷問題。

偵錯後清除偵錯設定

在偵錯目標上

完成偵錯後，請從偵錯目標移除核心偵錯組態，以還原正常的開機行為。在偵錯目標上，開啟提升的 PowerShell 工作階段並執行下列命令：

```
bcdedit /debug off
bcdedit /dbgsettings LOCAL
bcdedit /deletevalue loadoptions
```

如果您有 `loadoptions` 以外的現有 `FORCEHVTONOTSHAREDEBUGDEVICE`，您應該`bcdedit /set loadoptions`使用原始 執行 來還原設定`loadoptions`。

如果您已啟用開機偵錯，也請執行：

```
bcdedit /bootdebug off
```

重新啟動執行個體，讓變更生效。

在偵錯主機上

執行下列動作以移除防火牆規則：

```
Remove-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection"
```

限制

ENA 的 KDNET 擴充功能模組目前不支援：

- 第 8 代非金屬 x86_64 執行個體類型（例如 m8a.xlarge)
- 第 7 代 48xlarge 非金屬 x86_64 執行個體類型（例如 m7a.48xlarge)
- u7i 執行個體類型

模組目前不支援啟用安全開機的執行個體。您可以在提升的 PowerShell 工作階段Confirm-SecureBootUEFI中執行 驗證狀態。如果輸出為 True，則安全開機為作用中。請注意，根據預設，具有「TPM」字首的所有 Amazon 提供的映像都已啟用安全開機。

其他備註

如果您在將除錯器連線至目標執行個體時遇到問題，請確認下列事項：

- 符合本指南中列出的所有先決條件，包括所需的 Windows Server 建置版本和可擴展性模組的存在。
- 連接到兩個執行個體的安全群組已正確設定，以允許它們之間在設定的偵錯連接埠上的流量。
- 主機執行個體上的 Windows 防火牆規則不會封鎖已設定連接埠上兩個執行個體之間的網路流量。

如需其他指引，請參閱 Microsoft 文件中的手動設定 KDNET 網路核心偵錯。

重設 Amazon EC2 Windows 執行個體的 Windows 管理員密碼

如果您因為 Windows 管理員密碼遺失或過期，而無法再連線到 Windows Amazon EC2 執行個體，您可以重設密碼。

> **Note**
>
> 有一個 AWS Systems Manager 自動化文件會自動套用重設本機管理員密碼所需的手動步驟。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的在 EC2 執行個體重設密碼和 SSH 金鑰。

手動重設管理員密碼的方法需使用 EC2Launch v2、EC2Config 或 EC2Launch。

- 對於包含 EC2Launch v2 代理程式的所有支援 Windows AMI，請使用 EC2Launch v2。
- 如為 Windows Server 2016 之前的 Windows AMI，請用 EC2Config 服務。
- 如為 Windows Server 2016 和更新版本 AMI，請用 EC2Launch 服務。

這些程序還說明在遺失用來建立執行個體的金鑰對時如何連線至執行個體。Amazon EC2 使用公有金鑰來加密資料片段 (例如密碼)，而私有金鑰則用於解密資料。公有金鑰和私有金鑰稱為金鑰對。在 Windows 執行個體下用金鑰對取得管理員密碼，然後使用 RDP 登入。

> **Note**
>
> 如果您已停用執行個體上的本機管理員帳戶，且已為 Systems Manager 設定您的執行個體，您也可以使用 EC2Rescue 和執行命令來重新啟用和重設本機管理員密碼。如需詳細資訊，請參閱搭配使用 EC2Rescue for Windows Server 與 Systems Manager 執行命令。

目錄

- 使用 EC2Launch v2 重設 EC2 執行個體的 Windows 管理員密碼
- 使用 EC2Launch 重設 EC2 執行個體的 Windows 管理員密碼
- 使用 EC2Config 重設 EC2 執行個體的 Windows 管理員密碼

使用 EC2Launch v2 重設 EC2 執行個體的 Windows 管理員密碼

如果您遺失了 Windows 管理員密碼，且使用包含 EC2Launch v2 代理程式的已支援 Windows AMI，您可以使用 EC2Launch v2 產生新密碼。

如果您使用的是不包含 EC2Launch v2 代理程式的 Windows Server 2016 或更新版本 AMI，請參閱 使用 EC2Launch 重設 EC2 執行個體的 Windows 管理員密碼。

如果您使用的是不包含 EC2Launch v2 代理程式的 Windows Server 2016 之前的 Windows Server AMI，請參閱 使用 EC2Config 重設 EC2 執行個體的 Windows 管理員密碼。

> **Note**
>
> 如果您已停用執行個體上的本機管理員帳戶，且已為 Systems Manager 設定您的執行個體，您也可以使用 EC2Rescue 和執行命令來重新啟用和重設本機管理員密碼。如需詳細資訊，請參閱搭配使用 EC2Rescue for Windows Server 與 Systems Manager 執行命令。

> **Note**
>
> 有一個 AWS Systems Manager 自動化文件會自動套用重設本機管理員密碼所需的手動步驟。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的在 EC2 執行個體重設密碼和 SSH 金鑰。

若要使用 EC2Launch v2 來重設 Windows 管理員密碼，您需要執行下列步驟：

- 步驟 1：確認 EC2Launch v2 代理程式正在執行
- 步驟 2：將根磁碟區從執行個體中分離
- 步驟 3：將磁碟區連接至暫時執行個體
- 步驟 4：刪除 .run-once 檔案
- 步驟 5：重新啟動原始執行個體

步驟 1：確認 EC2Launch v2 代理程式正在執行

嘗試重設管理員密碼之前，請確認 EC2Launch v2 代理程式已安裝且正在執行中。您在本節稍後會用 EC2Launch v2 代理程式重設管理員密碼。

若要確認 EC2Launch v2 代理程式正在執行

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中選擇「Instances (執行個體)」，然後選擇需要重設密碼的執行個體。此執行個體即為本程序中的原始執行個體。
3. 選擇 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)、Get system log (取得系統日誌)。
4. 找出 EC2 Launch 項目，例如 Launch: EC2Launch v2 service v2.0.124。如果您看到此項目，表示 EC2Launch v2 服務正在執行中。

   如果系統日誌輸出為空白，或如果 EC2Launch v2 代理程式未執行，則請用執行個體主控台螢幕擷取畫面服務為執行個體進行故障診斷。如需詳細資訊，請參閱擷取無法連線執行個體的螢幕擷取畫面。

步驟 2：將根磁碟區從執行個體中分離

如果存放密碼的磁碟區連接至執行個體做為根磁碟區，則您無法使用 EC2Launch v2 來重設管理員密碼。您必須將磁碟區從原始執行個體中分離，才能將磁碟區連接至暫時執行個體做為次要磁碟區。

若要將根磁碟區從執行個體中分離

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取需要密碼重設的執行個體，然後選擇執行個體狀態、停止執行個體。執行個體的狀態變更為 Stopped (已停止) 之後，請繼續下一個步驟。
4. (選用) 如果您擁有在啟動此執行個體時已指定的私有金鑰，請繼續下一個步驟。否則，請使用下列步驟將執行個體取代為使用新金鑰對啟動的新執行個體。

   a. 使用 Amazon EC2 主控台來建立新金鑰對。若要使新金鑰對的名稱與已遺失私有金鑰的金鑰對的名稱相同，您必須先刪除現有金鑰對。
   b. 選取要取代的執行個體。請注意執行個體的執行個體類型、VPC、子網路、安全群組和 IAM 角色。
   c. 選擇執行個體後，選擇動作、映像和範本、建立映像。輸入映像的名稱和說明，然後選擇 Create image (建立映像)。
   d. 在導覽窗格中，選擇 AMIs (AMI)。等待映像狀態變更為可用。然後，選擇映像並選擇從 AMI 啟動執行個體。
   e. 填寫欄位以啟動執行個體，確保選擇與要替換的執行個體相同的執行個體類型、VPC、子網路、安全群組和 IAM 角色，然後選擇啟動執行個體。
   f. 出現提示時，請選擇您為新執行個體建立的金鑰對，然後選擇啟動執行個體。
   g. (選用) 如果原始執行個體有關聯的 Elastic IP 地址，將其傳輸至新執行個體。如果原始執行個體除了根磁碟區之外還有 EBS 磁碟區，請將它們傳輸至新執行個體。

5. 將根磁碟區從原始執行個體中分離，如下所示：

   a. 選取原始執行個體，然後選擇儲存索引標籤。請注意根裝置名稱下的根裝置名稱。在區塊型儲存設備下尋找具有此裝置名稱的磁碟區，並記下磁碟區 ID。
   b. 在導覽窗格中，選擇 Volumes (磁碟區)。
   c. 在磁碟區清單中，選擇您記為根裝置的磁碟區，然後選擇動作、分離磁碟區。在將磁碟區狀態變更為 available (可用) 後，繼續下一步驟。

6. 如果您建立了新的執行個體來取代原始執行個體，您現在可以終止原始執行個體。不再需要它。對於此程序的其餘部分，原始執行個體的所有參考都會套用至您建立的新執行個體。

步驟 3：將磁碟區連接至暫時執行個體

接下來，啟動暫時執行個體，並將磁碟區連接至此執行個體做為次要磁碟區。這是您用來修改組態檔案的執行個體。

啟動暫時執行個體並連接磁碟區

1. 啟動暫時執行個體，如下所示：

   a. 在導覽窗格中，選擇 Instances (執行個體)，選擇 Launch instances (啟動執行個體)，然後選擇 AMI。

   > **Important**
   >
   > 若要避免磁碟簽章衝突，您必須為不同版本的 Windows 選取 AMI。例如，如果原始執行個體執行 Windows Server 2019，則使用適用於 Windows Server 2016 的基本 AMI 啟動暫時執行個體。

   b. 保留預設執行個體類型，然後選擇 Next: Configure Instance Details (下一步：設定執行個體的詳細資訊)。

   c. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面中，對於 Subnet (子網路)，選取與原始執行個體相同的可用區域，然後選擇 Review and Launch (檢閱和啟動)。

   > **Important**
   >
   > 暫時執行個體與原始執行個體必須位於相同可用區域。如果您的暫時執行個體位於不同的可用區域，則無法將原始執行個體的根磁碟區與它連接。

   d. 在 Review Instance Launch (檢閱執行個體啟動) 頁面，選擇 Launch (啟動)。

   e. 出現提示時，建立新金鑰對，將其下載到電腦上的安全位置，然後選擇 Launch Instances (啟動執行個體)。

2. 將磁碟區連接到暫時執行個體作為輔助磁碟區，如下所示：

   a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從原始執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。

   b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入暫時執行個體的名稱或 ID，然後從清單中選取執行個體。

   c. 對於 Device (裝置)，輸入 **xvdf** (如果尚未出現在此)，選擇 Attach (連接)。

步驟 4：刪除 .run-once 檔案

現在必須從附接至執行個體的離線磁碟區刪除 `.run-once` 檔案。這會指示 EC2Launch v2 以 once 頻率執行所有工作，其中包括設定管理員密碼。您連接之輔助磁碟區中的檔案路徑會類似 `D:\ProgramData\Amazon\EC2Launch\state\.run-once`。

若要刪除 .run-once 檔案

1. 開啓磁碟管理公用程式，並依照下列說明使磁碟機上線：使 Amazon EBS 磁碟區可供使用。
2. 在已連線的磁碟中找到 `.run-once` 檔案。
3. 刪除 .run-once 檔案。

> **Important**
>
> 設定為執行一次的任何指令碼都會由此動作觸發。

步驟 5：重新啟動原始執行個體

刪除 `.run-once` 檔案之後，將磁碟區重新連接至原始執行個體做為根磁碟區，然後使用金鑰對來連接至執行個體，以擷取管理員密碼。

1. 將磁碟區重新連接到原始執行個體，如下所示：
   a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從暫時執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。
   b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入您的原始執行個體的名稱或 ID，然後選取執行個體。
   c. 對於 Device (裝置)，輸入 **`/dev/sda1`**。
   d. 選擇 Attach (連接)。在磁碟區狀態變更為 `in-use` 之後，繼續下一個步驟。
2. 在導覽窗格中，選擇 Instances (執行個體)。選取原始執行個體，然後依序選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。在執行個體狀態變更為 `Running` 之後，繼續下一個步驟。
3. 使用新金鑰對的私有金鑰擷取新的 Windows 系統管理員密碼，然後連線到執行個體。如需詳細資訊，請參閱 使用 RDP 連線至您的 Windows 執行個體。

> **Important**
>
> 先停止執行個體，接著再啟動後，該執行個體將得到新的公有 IP 地址。務必使用目前的公有 DNS 名稱連接至執行個體。如需詳細資訊，請參閱 Amazon EC2 執行個體狀態變更。

4. (選用) 如果不再使用暫時執行個體，您可以終止它。選取暫時執行個體，再依序選擇 Instance State (執行個體狀態)、Terminate instance (終止執行個體)。

使用 EC2Launch 重設 EC2 執行個體的 Windows 管理員密碼

如果您遺失了 Windows 管理員密碼，且使用 Windows Server 2016 或更新版本 AMI，您可透過 EC2Rescue 工具以利用 EC2Launch 服務產生新密碼。

如果您使用的是不包含 EC2Launch v2 代理程式的 Windows Server 2016 或更新版本 AMI，則可以使用 EC2Launch v2 產生新密碼。

如果使用 Windows Server 2016 之前的 Windows Server AMI，請參閱使用 EC2Config 重設 EC2 執行個體的 Windows 管理員密碼。

> **Warning**
>
> 當您停止執行個體時，執行個體存放磁碟區上的資料會遺失。若要保留此資料，請將其備份至持久性儲存。

> **Note**
>
> 如果您已停用執行個體上的本機管理員帳戶，且已為 Systems Manager 設定您的執行個體，您也可以使用 EC2Rescue 和執行命令來重新啟用和重設本機管理員密碼。如需詳細資訊，請參閱搭配使用 EC2Rescue for Windows Server 與 Systems Manager 執行命令。

> **Note**
>
> 有一個 AWS Systems Manager 自動化文件會自動套用重設本機管理員密碼所需的手動步驟。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的在 EC2 執行個體重設密碼和 SSH 金鑰。

若要使用 EC2Launch 來重設 Windows 管理員密碼，您需要執行下列步驟：

- 步驟 1：將根磁碟區從執行個體中分離
- 步驟 2：將磁碟區連接至暫時執行個體
- 步驟 3：重設管理員密碼
- 步驟 4：重新啟動原始執行個體

步驟 1：將根磁碟區從執行個體中分離

如果存放密碼的磁碟區連接至執行個體做為根磁碟區，則您無法使用 EC2Launch 來重設管理員密碼。您必須將磁碟區從原始執行個體中分離，才能將磁碟區連接至暫時執行個體做為次要磁碟區。

若要將根磁碟區從執行個體中分離

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取需要密碼重設的執行個體，然後選擇執行個體狀態、停止執行個體。執行個體的狀態變更為 Stopped (已停止) 之後，請繼續下一個步驟。
4. (選用) 如果您擁有在啟動此執行個體時已指定的私有金鑰，請繼續下一個步驟。否則，請使用下列步驟將執行個體取代為使用新金鑰對啟動的新執行個體。
   a. 使用 Amazon EC2 主控台來建立新金鑰對。若要使新金鑰對的名稱與已遺失私有金鑰的金鑰對的名稱相同，您必須先刪除現有金鑰對。
   b. 選取要取代的執行個體。請注意執行個體的執行個體類型、VPC、子網路、安全群組和 IAM 角色。
   c. 選擇執行個體後，選擇動作、映像和範本、建立映像。輸入映像的名稱和說明，然後選擇 Create image (建立映像)。
   d. 在導覽窗格中，選擇 AMIs (AMI)。等待映像狀態變更為可用。然後，選擇映像並選擇從 AMI 啟動執行個體。
   e. 填寫欄位以啟動執行個體，確保選擇與要替換的執行個體相同的執行個體類型、VPC、子網路、安全群組和 IAM 角色，然後選擇啟動執行個體。
   f. 出現提示時，請選擇您為新執行個體建立的金鑰對，然後選擇啟動執行個體。
   g. (選用) 如果原始執行個體有關聯的 Elastic IP 地址，將其傳輸至新執行個體。如果原始執行個體除了根磁碟區之外還有 EBS 磁碟區，請將它們傳輸至新執行個體。
5. 將根磁碟區從原始執行個體中分離，如下所示：

a. 選取原始執行個體，然後選擇儲存索引標籤。請注意根裝置名稱下的根裝置名稱。在區塊型儲存設備下尋找具有此裝置名稱的磁碟區，並記下磁碟區 ID。

b. 在導覽窗格中，選擇 Volumes (磁碟區)。

c. 在磁碟區清單中，選擇您記為根裝置的磁碟區，然後選擇動作、分離磁碟區。在將磁碟區狀態變更為 available (可用) 後，繼續下一步驟。

6. 如果您建立了新的執行個體來取代原始執行個體，您現在可以終止原始執行個體。不再需要它。對於此程序的其餘部分，原始執行個體的所有參考都會套用至您建立的新執行個體。

步驟 2：將磁碟區連接至暫時執行個體

接下來，啟動暫時執行個體，並將磁碟區連接至此執行個體做為次要磁碟區。這是您用來執行 EC2Launch 的執行個體。

啟動暫時執行個體並連接磁碟區

1. 啟動暫時執行個體，如下所示：

a. 在導覽窗格中，選擇 Instances (執行個體)，選擇 Launch instances (啟動執行個體)，然後選擇 AMI。

> **Important**
>
> 若要避免磁碟簽章衝突，您必須為不同版本的 Windows 選取 AMI。例如，如果原始執行個體執行 Windows Server 2019，則使用適用於 Windows Server 2016 的基本 AMI 啟動暫時執行個體。

b. 保留預設執行個體類型，然後選擇 Next: Configure Instance Details (下一步：設定執行個體的詳細資訊)。

c. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面中，對於 Subnet (子網路)，選取與原始執行個體相同的可用區域，然後選擇 Review and Launch (檢閱和啟動)。

> **Important**
>
> 暫時執行個體與原始執行個體必須位於相同可用區域。如果您的暫時執行個體位於不同的可用區域，則無法將原始執行個體的根磁碟區與它連接。

d. 在 Review Instance Launch (檢閱執行個體啟動) 頁面，選擇 Launch (啟動)。

e. 出現提示時，建立新金鑰對，將其下載到電腦上的安全位置，然後選擇 Launch Instances (啟動執行個體)。

2. 將磁碟區連接到暫時執行個體作為輔助磁碟區，如下所示：

   a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從原始執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。

   b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入暫時執行個體的名稱或 ID，然後從清單中選取執行個體。

   c. 對於 Device (裝置)，輸入 **xvdf** (如果尚未出現在此)，選擇 Attach (連接)。

步驟 3：重設管理員密碼

接下來，連接至暫時執行個體，並使用 EC2Launch 來重設管理員密碼。

重設管理員密碼

1. 連接至暫時執行個體，並在執行個體上使用 EC2Rescue for Windows Server 工具重設管理員密碼，如下所示：

   a. 下載 EC2Rescue for Windows Server zip 檔案，解壓縮內容，然後執行 EC2Rescue.exe (EC2Rescue.exe)。

   b. 在 License Agreement (授權合約) 畫面上閱讀授權合約，如果接受條款，請選擇 I Agree (我同意)。

   c. 在 Welcome to EC2Rescue for Windows Server (歡迎使用 EC2Rescue for Windows Server) 畫面上，選擇 Next (下一步)。

   d. 在 Select mode (選取模式) 畫面上，選擇 Offline instance (離線執行個體)。

   e. 在 Select a disk (選取磁碟) 畫面上選取 xvdf 設備，然後選擇 Next (下一步)。

   f. 確認選取磁碟，然後選擇 Yes (是)。

   g. 載入磁碟區之後，選擇 OK (確定)。

   h. 在 Select Offline Instance Option (選取離線執行個體選項) 畫面上，選擇 Diagnose and Rescue (診斷與救援)。

   i. 在 Summary (摘要) 畫面上檢閱資訊，然後選擇 Next (下一步)。

   j. 在 Detected possible issues (偵測到的可能問題) 畫面上，選取 Reset Administrator Password (重設管理員密碼)，然後選擇 Next (下一步)。

   k. 在 Confirm (確認) 畫面上選擇 Rescue (救援)、OK (確定)。

l. 在 Done (完成) 畫面上選擇 Finish (完成)。

m. 關閉 EC2Rescue for Windows Server 工具，中斷與暫時執行個體的連線，然後返回 Amazon EC2 主控台。

2. 將次要 (xvdf) 磁碟區從暫時執行個體中分離，如下所示：

a. 在導覽窗格中選擇 Instances (執行個體)，然後選取暫時執行個體。

b. 在暫時執行個體的「Storage (儲存)」標籤上，記下列為 xvdf 的 EBS 磁碟區識別碼。

c. 在導覽窗格中，選擇 Volumes (磁碟區)。

d. 在磁碟區清單中，請勾選之前步驟中所記下的磁碟區，並選擇 Actions (動作)、Detach Volume (分離磁碟區)。在將磁碟區狀態變更為 available (可用) 後，繼續下一步驟。

步驟 4：重新啟動原始執行個體

使用 EC2Launch 來重設管理員密碼之後，將磁碟區重新連接至原始執行個體做為根磁碟區，然後使用金鑰對來連接至執行個體，以擷取管理員密碼。

重新啟動原始執行個體

1. 將磁碟區重新連接到原始執行個體，如下所示：

a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從暫時執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。

b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入您的原始執行個體的名稱或 ID，然後選取執行個體。

c. 對於 Device (裝置)，輸入 **/dev/sda1**。

d. 選擇 Attach (連接)。在磁碟區狀態變更為 `in-use` 之後，繼續下一個步驟。

2. 在導覽窗格中，選擇 Instances (執行個體)。選取原始執行個體，然後依序選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。在執行個體狀態變更為 `Running` 之後，繼續下一個步驟。

3. 使用新金鑰對的私有金鑰擷取新的 Windows 系統管理員密碼，然後連線到執行個體。如需詳細資訊，請參閱 使用 RDP 連線至您的 Windows 執行個體。

4. (選用) 如果不再使用暫時執行個體，您可以終止它。選取暫時執行個體，再依序選擇 Instance State (執行個體狀態)、Terminate instance (終止執行個體)。

使用 EC2Config 重設 EC2 執行個體的 Windows 管理員密碼

如果您遺失了 Windows 管理員密碼，且使用 Windows Server 2016 之前的 Windows AMI，您可以使用 EC2Config 代理程式產生新密碼。

如果您使用的是 Windows Server 2016 或更新版本 AMI，請參閱 使用 EC2Launch 重設 EC2 執行個體的 Windows 管理員密碼，或您可透過 EC2Rescue 工具以利用 EC2Launch 服務產生新密碼。

> **Note**
>
> 如果您已停用執行個體上的本機管理員帳戶，且已為 Systems Manager 設定您的執行個體，您也可以使用 EC2Rescue 和執行命令來重新啟用和重設本機管理員密碼。如需詳細資訊，請參閱搭配使用 EC2Rescue for Windows Server 與 Systems Manager 執行命令。

> **Note**
>
> 有一個 AWS Systems Manager 自動化文件會自動套用重設本機管理員密碼所需的手動步驟。如需詳細資訊，請參閱「AWS Systems Manager 使用者指南」中的在 EC2 執行個體重設密碼和 SSH 金鑰。

若要使用 EC2Config 來重設 Windows 管理員密碼，您需要執行下列步驟：

- 步驟 1：確認 EC2Config 服務在執行中
- 步驟 2：將根磁碟區從執行個體中分離
- 步驟 3：將磁碟區連接至暫時執行個體
- 步驟 4：修改組態檔案
- 步驟 5：重新啟動原始執行個體

步驟 1：確認 EC2Config 服務在執行中

嘗試重設管理員密碼之前，請確認 EC2Config 服務已安裝且正執行中。您在本節稍後會用 EC2Config 服務重設管理員密碼。

確認 EC2Config 服務在執行中

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中選擇「Instances (執行個體)」，然後選擇需要重設密碼的執行個體。此執行個體即為本程序中的原始執行個體。
3. 選擇 Actions (動作)、Monitor and troubleshoot (監視和故障診斷)、Get system log (取得系統日誌)。
4. 找出 EC2 Agent 項目，例如：EC2 Agent: Ec2Config service v3.18.1118。如果找到此項目，表示 EC2Config 服務正在執行中。

   如果系統日誌輸出為空白，或如果 EC2Config 服務未執行，請用執行個體主控台螢幕擷取畫面服務為執行個體進行故障診斷。如需詳細資訊，請參閱擷取無法連線執行個體的螢幕擷取畫面。

步驟 2：將根磁碟區從執行個體中分離

如果存放密碼的磁碟區連接至執行個體做為根磁碟區，則您無法使用 EC2Config 來重設管理員密碼。您必須將磁碟區從原始執行個體中分離，才能將磁碟區連接至暫時執行個體做為次要磁碟區。

若要將根磁碟區從執行個體中分離

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取需要密碼重設的執行個體，然後選擇執行個體狀態、停止執行個體。執行個體的狀態變更為 Stopped (已停止) 之後，請繼續下一個步驟。
4. (選用) 如果您擁有在啟動此執行個體時已指定的私有金鑰，請繼續下一個步驟。否則，請使用下列步驟將執行個體取代為使用新金鑰對啟動的新執行個體。

   a. 使用 Amazon EC2 主控台來建立新金鑰對。若要使新金鑰對的名稱與已遺失私有金鑰的金鑰對的名稱相同，您必須先刪除現有金鑰對。

   b. 選取要取代的執行個體。請注意執行個體的執行個體類型、VPC、子網路、安全群組和 IAM 角色。

   c. 選擇執行個體後，選擇動作、映像和範本、建立映像。輸入映像的名稱和說明，然後選擇 Create image (建立映像)。

   d. 在導覽窗格中，選擇 AMIs (AMI)。等待映像狀態變更為可用。然後，選擇映像並選擇從 AMI 啟動執行個體。

   e. 填寫欄位以啟動執行個體，確保選擇與要替換的執行個體相同的執行個體類型、VPC、子網路、安全群組和 IAM 角色，然後選擇啟動執行個體。

   f. 出現提示時，請選擇您為新執行個體建立的金鑰對，然後選擇啟動執行個體。

g. (選用) 如果原始執行個體有關聯的 Elastic IP 地址，將其傳輸至新執行個體。如果原始執行個體除了根磁碟區之外還有 EBS 磁碟區，請將它們傳輸至新執行個體。

5. 將根磁碟區從原始執行個體中分離，如下所示：

   a. 選取原始執行個體，然後選擇儲存索引標籤。請注意根裝置名稱下的根裝置名稱。在區塊型儲存設備下尋找具有此裝置名稱的磁碟區，並記下磁碟區 ID。

   b. 在導覽窗格中，選擇 Volumes (磁碟區)。

   c. 在磁碟區清單中，選擇您記為根裝置的磁碟區，然後選擇動作、分離磁碟區。在將磁碟區狀態變更為 available (可用) 後，繼續下一步驟。

6. 如果您建立了新的執行個體來取代原始執行個體，您現在可以終止原始執行個體。不再需要它。對於此程序的其餘部分，原始執行個體的所有參考都會套用至您建立的新執行個體。

步驟 3：將磁碟區連接至暫時執行個體

接下來，啟動暫時執行個體，並將磁碟區連接至此執行個體做為次要磁碟區。這是您用來修改組態檔案的執行個體。

啟動暫時執行個體並連接磁碟區

1. 啟動暫時執行個體，如下所示：

   a. 在導覽窗格中，選擇 Instances (執行個體)，選擇 Launch instances (啟動執行個體)，然後選擇 AMI。

   > **Important**
   >
   > 若要避免磁碟簽章衝突，您必須為不同版本的 Windows 選取 AMI。例如，如果原始執行個體執行 Windows Server 2019，則使用適用於 Windows Server 2016 的基本 AMI 啟動暫時執行個體。

   b. 保留預設執行個體類型，然後選擇 Next: Configure Instance Details (下一步：設定執行個體的詳細資訊)。

   c. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面中，對於 Subnet (子網路)，選取與原始執行個體相同的可用區域，然後選擇 Review and Launch (檢閱和啟動)。

> **Important**
>
> 暫時執行個體與原始執行個體必須位於相同可用區域。如果您的暫時執行個體位於不同的可用區域，則無法將原始執行個體的根磁碟區與它連接。

d. 在 Review Instance Launch (檢閱執行個體啟動) 頁面，選擇 Launch (啟動)。

e. 出現提示時，建立新金鑰對，將其下載到電腦上的安全位置，然後選擇 Launch Instances (啟動執行個體)。

2. 將磁碟區連接到暫時執行個體作為輔助磁碟區，如下所示：

a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從原始執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。

b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入暫時執行個體的名稱或 ID，然後從清單中選取執行個體。

c. 對於 Device (裝置)，輸入 **xvdf** (如果尚未出現在此)，選擇 Attach (連接)。

步驟 4：修改組態檔案

將磁碟區連接至暫時執行個體做為次要磁碟區之後，請在組態檔案中修改 `Ec2SetPassword` 外掛程式。

修改組態檔案

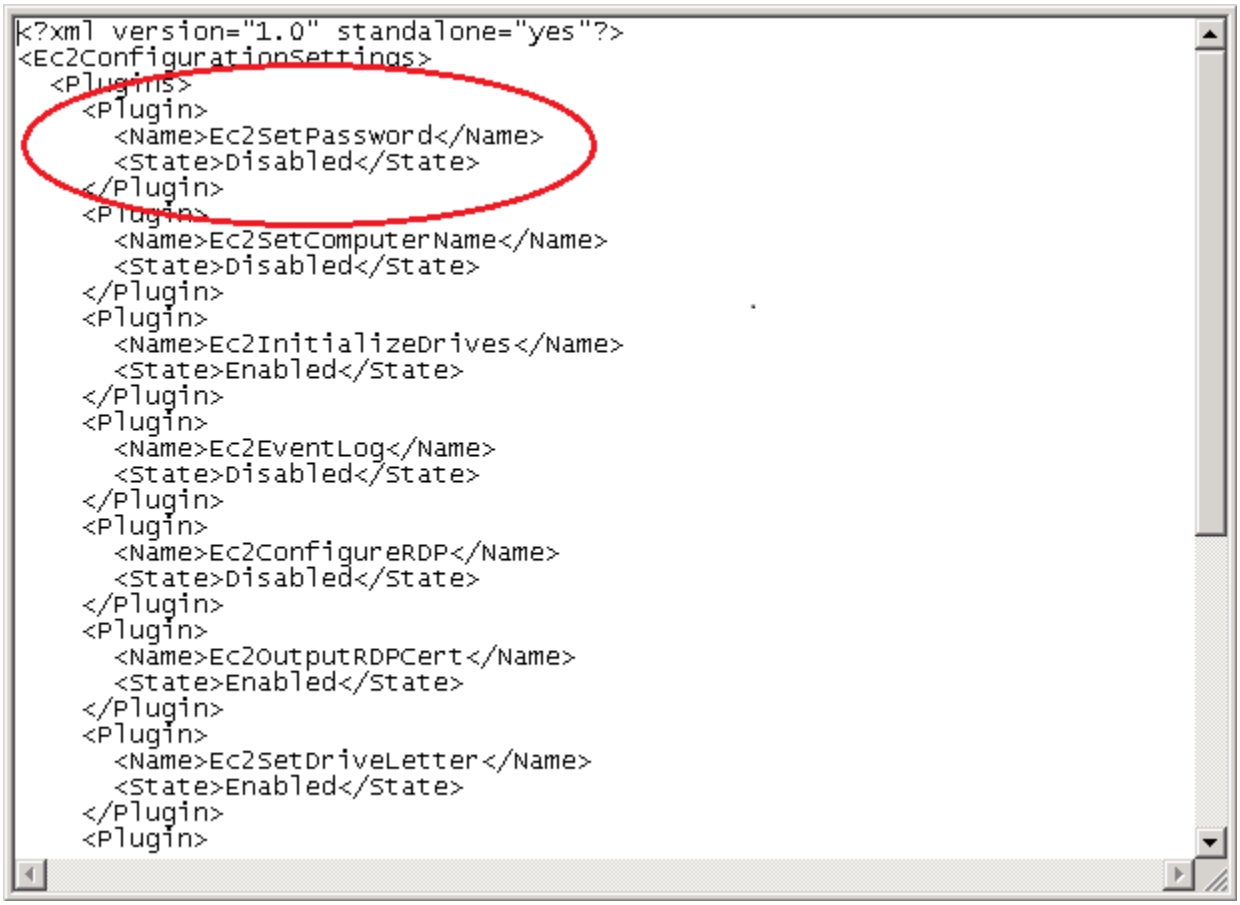
1. 從暫時執行個體，修改次要磁碟區上的組態檔案，如下所示：

a. 啟動並連接至暫時執行個體。

b. 使用下列指示讓磁碟機上線：讓 Amazon EBS 磁碟區可供使用。

c. 導覽至次要磁碟區，然後使用「記事本」等文字編輯器開啟 `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`。

d. 在檔案頂端，找出名稱為 `Ec2SetPassword` 的外掛程式，如螢幕擷取畫面所示。將狀態從 `Disabled` 變更為 `Enabled` 並儲存檔案。



2. 修改組態檔案之後，將次要磁碟區從暫時執行個體中分離，如下所示：

   a. 使用 Disk Management (磁碟管理) 公用程式讓磁碟區離線。

   b. 中斷與暫時執行個體的連線，並返回 Amazon EC2 主控台。

   c. 在導覽窗格中選擇 Volumes (磁碟區)、選取磁碟區，然後選擇 Actions (動作)、Detach Volume (分離磁碟區)。在將磁碟區狀態變更為 available (可用) 後，繼續下一步驟。

步驟 5：重新啟動原始執行個體

修改組態檔案之後，將磁碟區重新連接至原始執行個體做為根磁碟區，然後使用金鑰對來連接至執行個體，以擷取管理員密碼。

1. 將磁碟區重新連接到原始執行個體，如下所示：

   a. 在導覽窗格中，選擇 Volumes (磁碟區)，選取從暫時執行個體分離的磁碟區，然後選擇 Actions (動作)、Attach Volume (連接磁碟區)。

b. 在 Attach Volume (連接磁碟區) 對話方塊中，對於 Instances (執行個體)，開始輸入您的原始執行個體的名稱或 ID，然後選取執行個體。

c. 對於 Device (裝置)，輸入 **/dev/sda1**。

d. 選擇 Attach (連接)。在磁碟區狀態變更為 `in-use` 之後，繼續下一個步驟。

2. 在導覽窗格中，選擇 Instances (執行個體)。選取原始執行個體，然後依序選擇 Instance state (執行個體狀態)、Start instance (啟動執行個體)。在執行個體狀態變更為 `Running` 之後，繼續下一個步驟。

3. 使用新金鑰對的私有金鑰擷取新的 Windows 系統管理員密碼，然後連線到執行個體。如需詳細資訊，請參閱 使用 RDP 連線至您的 Windows 執行個體。

> **Important**
>
> 先停止執行個體，接著再啟動後，該執行個體將得到新的公有 IP 地址。務必使用目前的公有 DNS 名稱連接至執行個體。如需詳細資訊，請參閱 Amazon EC2 執行個體狀態變更。

4. (選用) 如果不再使用暫時執行個體，您可以終止它。選取暫時執行個體，再依序選擇 Instance State (執行個體狀態)、Terminate instance (終止執行個體)。

針對 Amazon EC2 Windows 執行個體的 Sysprep 問題進行疑難排解

如果您在映像準備期間發生問題或收到錯誤訊息，請檢閱下列日誌。日誌位置取決於您使用 Sysprep 執行的是 EC2Config、EC2Launch v1 或 EC2Launch v2。

- `%WINDIR%\Panther\Unattendgc` (EC2Config、EC2Launch v1 和 EC2Launch v2)
- `%WINDIR%\System32\Sysprep\Panther` (EC2Config、EC2Launch v1 和 EC2Launch v2)
- `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt` (僅限 EC2Config)
- `C:\ProgramData\Amazon\Ec2Config\Logs` (僅限 EC2Config)
- `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log` (僅限 EC2Launch v1)
- `%ProgramData%\Amazon\EC2Launch\log\agent.log` (僅限 EC2Launch v2)

如果您在使用 Sysprep 準備映像期間收到錯誤訊息，可能會無法連接 OS。若要檢閱日誌檔案，您必須停止執行個體，並將其根磁碟區連接到另一個運作狀態良好的執行個體做為輔助磁碟區，然後於次要

磁碟區檢閱前述的日誌。如需依名稱排列日誌檔用途的詳細資訊，請參閱 Microsoft 文件中的 Windows 安裝程式相關日誌檔。

如果您在 Unattendgc 日誌檔案中找到錯誤，請使用 Microsoft Error Lookup Tool 取得錯誤的其他詳細資訊。Unattendgc 日誌檔案中所報告的下列問題，通常是因為執行個體上有一或多個毀損的使用者描述檔而導致的結果：

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

有兩個選項可解決此問題：

選項 1

在執行個體上使用 Regedit，搜尋下列機碼。確認沒有已刪除之使用者的描述檔登錄機碼。

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion
\ProfileList\
```

選項 2

1. 編輯相關檔案，如下所示：
   - Windows Server 2012 R2 和舊版 — 編輯 EC2Config 回應檔案 (`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`)。
   - Windows Server 2016 和 2019 — 編輯 unattend.xml 回應檔案 (`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`)。
   - Windows Server 2022 — 編輯 unattend.xml 回應檔案 (`C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml`)。
2. 將 `<CopyProfile>true</CopyProfile>` 變更為 `<CopyProfile>false</CopyProfile>`。
3. 重新執行 Sysprep。請注意，此組態變更將會在 Sysprep 完成之後，刪除內建管理員使用者描述檔。

使用 EC2Rescue 疑難排解受損的 Amazon EC2 Linux 執行個體

適用於 Linux 的 EC2Rescue 是容易使用的開源工具，您可在 Amazon EC2 Linux 執行個體執行它，利用其 100 多個模組的程式庫診斷、疑難排解及修復常見問題。模組是包含 BASH 或 Python 指令碼和必要中繼資料的 YAML 檔案。

適用於 Linux 的 EC2Rescue 執行個體的一些一般使用案例包括：

- 收集 syslog 和套件管理員日誌
- 收集資源使用率資料
- 診斷和修復已知有問題的核心參數和常見 OpenSSH 問題

> **Note**
>
> `AWSSupport-TroubleshootSSH` AWS Systems Manager Automation Runbook 安裝適用於 Linux 的 EC2Rescue，然後使用 工具來檢查或嘗試修正防止 SSH 連線至 Linux 執行個體的常見問題。如需詳細資訊，請參閱 AWSSupport-TroubleshootSSH。

如果您使用的是 Windows 執行個體，請參閱 the section called “適用於 Windows 執行個體的 EC2Rescue”。

主題

- 在 Amazon EC2 Linux 執行個體上安裝 EC2Rescue
- 在 Amazon EC2 Linux 執行個體上執行 EC2Rescue 命令
- 為 Amazon EC2 Linux 執行個體開發 EC2Rescue 模組

在 Amazon EC2 Linux 執行個體上安裝 EC2Rescue

適用於 Linux 的 EC2Rescue 工具可安裝在符合下列要件的 Amazon EC2 Linux 執行個體上。

先決條件

- 支援的作業系統：
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SUSE Linux Enterprise Server 12+
  - RHEL 7+
  - Ubuntu 16.04+
- 軟體要求：
  - Python 2.7.9+ 或 3.2+

安裝 EC2Rescue

AWSSupport-TroubleshootSSH Runbook 會安裝適用於 Linux 的 EC2Rescue，然後使用 工具來檢查或嘗試修正無法透過 SSH 遠端連線至 Linux 機器的常見問題。如需詳細資訊，並執行此自動化，請參閱 支援-TroubleshootSSH。

如果您的系統具有所需的 Python 版本，則您可以安裝標準組建。否則，您可以安裝隨附的組建，其中包括最小的 Python 複本。

安裝標準組建

1. 從工作中的 Linux 執行個體下載 適用於 Linux 的 EC2Rescue 工具：

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (選用) 確認適用於 Linux 安裝檔的 EC2Rescue 簽章。如需詳細資訊，請參閱(選用) 確認 適用於 Linux 的 EC2Rescue 的簽章。
3. 下載 sha256 雜湊檔案：

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. 確認 tarball 的完整性：

```
sha256sum -c ec2rl.tgz.sha256
```

5. 解壓縮 tarball：

```
tar -xzvf ec2rl.tgz
```

6. 列出說明檔確認安裝成功：

```
cd ec2rl-<version_number>
./ec2rl help
```

安裝隨附的組建

如需下載和限制清單的連結，請參閱 github 上的 適用於 Linux 的 EC2Rescue。

(選用) 確認 適用於 Linux 的 EC2Rescue 的簽章

下列是驗證用於以 Linux 為基礎之作業系統的 適用於 Linux 的 EC2Rescue 套件有效性的建議程序。

當您從網際網路下載應用程式時，建議您驗證軟體發佈者的身分，並檢查應用程式在發佈之後並未遭更改或損毀。如此可保護您，避免安裝到包含病毒或其他惡意程式碼的應用程式版本。

如果在執行此主題中的步驟後，您判斷 適用於 Linux 的 EC2Rescue 的軟體已遭更改或損毀，請勿執行安裝檔案。反之，請聯絡 Amazon Web Services。

適用於以 Linux 為基礎之作業系統的 適用於 Linux 的 EC2Rescue 檔案使用 GnuPG 簽章，這是一種用於安全數位簽章的 Pretty Good Privacy (OpenPGP) 標準的開放原始碼實作。GnuPG （ 也稱為 GPG) 透過數位簽章提供身分驗證和完整性檢查。 會 AWS 發佈公有金鑰和簽章，供您用來驗證下載的 EC2Rescue for Linux 套件。如需 PGP 和 GnuPG (GPG) 的詳細資訊，請參閱 https：//https://www.gnupg.org/。

第一步是與軟體發佈者建立信任。下載軟體發佈者的公開金鑰，檢查公開金鑰的擁有者是否為聲稱的擁有者，然後將公開金鑰新增至您的 keyring。您的 keyring 是一組已知的公開金鑰。在您建立公開金鑰的真實性之後，即可用它來驗證應用程式的簽章。

任務

- 驗證和匯入公開金鑰
- 驗證套件的簽章

驗證和匯入公開金鑰

此程序的下一個步驟是驗證 適用於 Linux 的 EC2Rescue 公開金鑰，並將其做為信任的金鑰新增至 GPG keyring。

驗證和匯入 適用於 Linux 的 EC2Rescue 公開金鑰

1. 在命令提示中，使用以下命令以取得我們公開 GPG 建置金鑰的複本：

   ```
   curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
   ```

2. 在您儲存 `ec2rl.key` 的目錄的命令提示字元中，使用下列命令將 適用於 Linux 的 EC2Rescue 公開金鑰匯入至您的 keyring：

   ```
   gpg2 --import ec2rl.key
   ```

此命令會傳回類似以下的結果：

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

> **Tip**
>
> 如果您看到錯誤，指出找不到命令，請使用 `apt-get install gnupg2` (基於 Debian 的 Linux) 或 `yum install gnupg2` (基於 Red Hat 的 Linux) 安裝 GnuPG 公用程式。

驗證套件的簽章

在您安裝 GPG 工具、驗證及匯入 適用於 Linux 的 EC2Rescue 公有金鑰，以及驗證 適用於 Linux 的 EC2Rescue 公有金鑰可信任之後，您就可以驗證 適用於 Linux 的 EC2Rescue 安裝指令碼的簽章。

驗證 適用於 Linux 的 EC2Rescue 安裝指令碼簽章

1. 在命令提示字元上，執行以下命令以下載安裝指令碼的簽章檔案：

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
```

2. 在您儲存 `ec2rl.tgz.sig` 與 適用於 Linux 的 EC2Rescue 安裝檔案的目錄中，在命令提示字元上執行以下命令以驗證簽章。兩個檔案都必須存在。

```
gpg2 --verify ./ec2rl.tgz.sig
```

輸出應類似以下所示：

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C  C36A F780 4843 2FAE 2A1C
     Subkey fingerprint: 966B 0D27 85E9 AEEC 1146  7A9D 8851 1153 6991 ED45
```

如果輸出包含 `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"` 片語，表示簽章已成功驗證，您可以繼續執行 適用於 Linux 的 EC2Rescue 安裝指令碼。

如果輸出包含 `BAD signature` 片語，請檢查您是否已正確執行程序。如果您持續收到此回應，請聯絡 Amazon Web Services 並且不要執行您先前下載的安裝檔。

以下是您可能會看到的警告的詳細資訊：

- WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.這是指您個人對於您所擁有適用於 適用於 Linux 的 EC2Rescue 的真實公有金鑰的信任。在理想世界中，您請會前往 Amazon Web Services 辦公室並獲得該金鑰。不過，通常您會從網站下載。在此情況下，該網站是 Amazon Web Services 網站。
- gpg2: no ultimately trusted keys found. 這表示該特定金鑰未獲得您 (或您信任的其他人) 的「最終信任」。

如需詳細資訊，請參閱 https：//https://www.gnupg.org/。

在 Amazon EC2 Linux 執行個體上執行 EC2Rescue 命令

EC2Rescue 是命令行工具。在 Linux 執行個體上安裝 EC2Rescue 之後，您可以透過執行 `./ec2rl help` 來取得如何使用工具的一般說明。您可以透過執行 `./ec2rl list` 來檢視可用模組，也可以透過執行 `./ec2rl help` *`module_name`* 來取得特定模組的協助。

下列是您可執行以開始使用本工具的常用任務。

任務

- 執行 EC2Rescue 模組
- 上傳 EC2Rescue 模組結果
- 建立 Amazon EC2 Linux 執行個體備份

執行 EC2Rescue 模組

執行所有 EC2Rescue 模組

使用 ./ec2rl run 命令而不指定任何其他參數。部分模組需要根存取權。若您不是根使用者，請在執行命令時使用 sudo。

```
./ec2rl run
```

執行特定 EC2Rescue 模組

使用 ./ec2rl run 命令，若為 `--only-modules`，請指定要執行的模組名稱。有些模組需要引數才能加以使用。

```
./ec2rl run --only-modules=module_name --arguments
```

例如，若要執行 dig 模組以查詢 `amazon.com` 網域，請使用下列命令。

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

檢視 EC2Rescue 模組結果

執行模組，然後檢視 `cat /var/tmp/ec2rl/`*logfile_location* 的日誌檔。例如，可以在以下位置找到 dig 模組的日誌檔：

```
cat /var/tmp/ec2rl/timestamp/mod_out/run/dig.log
```

上傳 EC2Rescue 模組結果

如果 支援 已請求 EC2Rescue 模組的結果，您可以使用 EC2Rescue 工具上傳日誌檔案。您可以將結果上傳到 提供的位置， 支援 或上傳到您擁有的 Amazon S3 儲存貯體。

將結果上傳至 提供的位置 支援

使用 ./ec2rl upload 命令。對於 `--upload-directory`，指定日誌檔位置。對於 `--support-url`，指定由 支援提供的 URL。

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --support-
url="url_provided_by_aws_support"
```

將報告上傳至 Amazon S3 儲存貯體

使用 ./ec2rl upload 命令。對於 `--upload-directory`，指定日誌檔位置。對於 `--presigned-url`，請指定 S3 儲存貯體的預先簽章 URL。如需為 Amazon S3 產生預先簽章的 URL 的詳細資訊，請參閱使用預先簽章的 URL 上傳物件。

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --presigned-
url="presigned_s3_url"
```

建立 Amazon EC2 Linux 執行個體備份

您可以透過建立 AMI 或建立附加磁碟區的快照來使用 EC2Rescue 備份 Linux 執行個體。

若要建立 AMI

使用 `./ec2rl run` 命令，以及針對 `--backup`，指定 `ami`。

```
./ec2rl run --backup=ami
```

若要建立所有附加磁碟區的多磁碟區快照

使用 `./ec2rl run` 命令，以及針對 `--backup`，指定 `allvolumes`。

```
./ec2rl run --backup=allvolumes
```

若要建立指定附加磁碟區的快照

使用 `./ec2rl run` 命令，以及針對 `--backup`，指定要備份的磁碟區 ID。

```
./ec2rl run --backup=vol-01234567890abcdef
```

為 Amazon EC2 Linux 執行個體開發 EC2Rescue 模組

模組是使用 YAML (資料序列化標準) 撰寫。模組的 YAML 檔案是由單一文件組成，代表模組及其屬性。

新增模組屬性

下表列出可用的模組屬性。

屬性	描述
name	模組的名稱。此名稱長度應小於或等於 18 個字元。
version	模組的版本編號。

屬性	描述
title	模組的簡述標題。此值長度應小於或等於 50 個字元。

helptext	模組的延伸描述。各行長度應小於或等於 75 個字元。如果模組使用引數 (無論必要或選用)，請以 helptext 值包含它們。
	
	例如：
	
	`helptext: !!str |`
	`  Collect output from ps for system analysis`
	`  Consumes --times= for number of times to repeat`
	`  Consumes --period= for time period between repetition`

placement	應該執行模組的階段。支援的值為： • prediagnostic • run • postdiagnostic
語言	撰寫模組程式碼的語言。支援的值為： • bash • Python **Note** Python 程式碼必須與 Python 2.7.9+ 和 Python 3.2+ 都相容。

屬性	描述
remediation	指出模組是否支援修補。支援的值為 `True` 或 `False`。 如果無此屬性，模組將預設為 `False`，讓其成為不支援修補之模組的選用屬性。
content	完整的指令碼程式碼。
限制條件	包含限制值的物件名稱。
domain	說明模組如何分組或分類的描述項。包含的模組集使用下列網域： • application • net • os • performance
class	模組執行的任務類型描述項。包含的模組集使用下列類別： • collect (收集程式的輸出) • diagnose (根據條件集通過/失敗) • gather (複製檔案並寫入特定檔案)
distro	此模組支援的 Linux 發行版本清單。包含的模組集使用下列發行版本： • alami (Amazon Linux) • rhel • ubuntu • suse
必要	模組從 CLI 選項使用的必要引數。
選擇性	模組可以使用的選用引數。

屬性	描述
software	在模組中使用的軟體可執行檔。此屬性旨在指定非預設安裝的軟體。適用於 Linux 的 EC2Rescue 邏輯在執行模組前會確保這些程式存在並可執行。
package	可執行檔的來源軟體套件。此屬性旨在提供套件有關軟體的延伸詳細資料，包括下載或取得未來資訊的 URL。
sudo	指出是否一定要有根存取才能執行模組。 您不必在模組指令碼中實作 sudo 檢查。如果此值為 true，則當執行使用者有根存取時，適用於 Linux 的 EC2Rescue 邏輯只會執行模組。
perfimpact	指出模組是否會顯著影響其執行所在環境的效能。如果此值為 true 且無 `--perfimpact=true` 引數，則跳過模組。
parallelexclusive	指定需要互斥的程式。例如，所有指定 "bpf" 的模組都依序執行。

新增環境變數

下表列出可用的環境變數。

環境變數	描述
`EC2RL_CALLPATH`	`ec2rl.py` 的路徑。此路徑可用以尋找 lib 目錄及使用經過廠商處理的 Python 模組。
`EC2RL_WORKDIR`	診斷工具的主要 tmp 目錄。 預設值：`/var/tmp/ec2rl` 。
`EC2RL_RUNDIR`	存放所有輸出的目錄。

環境變數	描述
	預設值：/var/tmp/ec2rl/<date&timestamp> 。
EC2RL_GATHEREDDIR	放置所收集模組資料的根目錄。 預設值：/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/ 。
EC2RL_NET_DRIVER	執行個體上按字母順序第一個非虛擬網路界面使用的驅動程式。 範例: • xen_netfront • ixgbevf • ena
EC2RL_SUDO	如果適用於 Linux 的 EC2Rescue 執行為根，即為 true；否則為 false。
EC2RL_VIRT_TYPE	如執行個體中繼資料所提供的虛擬化類型。 範例： • default-hvm • default-paravirtual
EC2RL_INTERFACES	系統上的界面列舉清單。值是包含名稱的字串，例如 eth1、 eth0等。這是由 產生的functions.bash ，並且僅適用於已取得它的模組。

使用 YAML 語法

建構您的模組 YAML 檔案時，應該注意下列各點：

- 三個連字號 (---) 表示文件的明確開頭。
- 從資料串流建立物件時，!ec2rlcore.module.Module 標籤會告訴 YAML 剖析器要呼叫哪個建構函數。您可於 module.py 檔案中找到建構函數。
- !!str 標籤會告訴 YAML 剖析器不要嘗試判斷資料類型，改將內容解譯為字串常值。
- 管道字元 (|) 會告訴 YAML 剖析器此值為常值型純量。在本例中，剖析器包含所有空格。這對模組而言很重要，因為會保留縮排和換行字元。
- YAML 標準會縮排兩格，如下列範例所示。請確定您的指令碼保持標準縮排 (例如，Python 四格)，然後在模組檔案中將完整內容縮排兩格。

範例模組

範例一 (mod.d/ps.yaml)：

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
  Collect output from ps for system analysis
  Requires --times= for number of times to repeat
  Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
      printf "%0.s=" {1..80}
      echo -e "\nERROR: "$BASH_COMMAND" exited with an error on line ${BASH_LINENO[0]}"
      exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
```

```
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every
 $period seconds."
  for i in $(seq 1 $times); do
      ps auxww
      sleep $period
  done
constraint:
  requires_ec2: !!str False
  domain: !!str performance
  class: !!str collect
  distro: !!str alami ubuntu rhel suse
  required: !!str period times
  optional: !!str
  software: !!str
  sudo: !!str False
  perfimpact: !!str False
  parallelexclusive: !!str
```

使用 EC2Rescue 疑難排解受損的 Amazon EC2 Windows 執行個體

EC2Rescue for Windows Server 是容易使用的工具，您可在 Amazon EC2 Windows Server 執行個體執行它以診斷及故障診斷可能的問題。收集日誌檔案和故障診斷問題以及主動搜尋可能發生問題的區域，極其重要。它甚至可以從其他執行個體檢查 Amazon EBS 根磁碟區，並收集相關日誌以故障診斷使用該磁碟區的 Windows Server 執行個體。以下是 EC2Rescue 可以解決的一些常見問題：

- 防火牆、遠端桌面通訊協定 (RDP) 或網路介面組態所造成的執行個體連線問題
- 由於停止錯誤、開機迴圈或登錄損壞而導致的作業系統開機問題
- 可能需要進階日誌分析和疑難排解的問題

EC2Rescue for Windows Server 有兩個不同模組：

- 資料收集器模組可從所有不同來源收集資料
- 分析器模組可根據一系列預先定義的規則分析收集的資料，以識別問題並提供建議

EC2Rescue for Windows Server 工具只能在執行 Windows Server 2012 和更新版本的 Amazon EC2 執行個體上執行。當此工具啟動時，它會檢查自己是否在 Amazon EC2 執行個體上執行。

> **Note**
>
> `AWSSupport-ExecuteEC2Rescue` AWS Systems Manager Automation Runbook 使用 EC2Rescue 工具進行故障診斷，並盡可能修正與指定 EC2 執行個體的常見連線問題。如需詳細資訊，並執行此自動化，請參閱 >AWSSupport-ExecuteEC2Rescue。

如果使用 Linux 執行個體，請參閱 the section called “適用於 Linux 執行個體的 EC2Rescue”。

主題

- 使用 EC2Rescue GUI 來疑難排解受損的 Windows 執行個體
- 使用 EC2Rescue CLI 來疑難排解受損的 Windows 執行個體
- 使用 EC2Rescue 和 Systems Manager 來疑難排解受損的 Windows 執行個體

使用 EC2Rescue GUI 來疑難排解受損的 Windows 執行個體

EC2Rescue for Windows Server 可對離線執行個體執行下列分析：

選項	描述

診斷與救援	EC2Rescue for Windows Server 可以偵測並解決下列服務設定的問題：
	• 系統時間
	  • RealTimeisUniversal - 偵測是否啟用 `RealTimeisUniversal` 登錄機碼。如果停用，當時區值不是設為 UTC 時，Windows 系統時間會產生差異。
	• Windows 防火牆
	  • Domain networks (網域網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。
	  • Private networks (私有網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。

選項	描述

	• Guest or public networks (訪客或公有網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。
	
	• 遠端桌面
	  • Service Start (服務啟動) - 偵測是否啟用遠端桌面服務。
	  • Remote Desktop Connections (遠端桌面連線) - 偵測是否啟用此服務。
	  • TCP Port (TCP 連接埠) - 偵測遠端桌面服務正在接聽哪一個連接埠。
	
	• EC2Config (Windows Server 2012 R2 和更早的版本)
	  • Installation (安裝) - 偵測安裝的是哪個 EC2Config 版本。
	  • Service Start (服務啟動) - 偵測是否啟用 EC2Config 服務。
	  • Ec2SetPassword - 產生新的管理員密碼。
	  • Ec2HandleUserData - 允許您下次開機執行個體時執行使用者資料指令碼。
	
	• EC2Launch (Windows Server 2016 及更新版本)
	  • Installation (安裝) - 偵測安裝的是哪個 EC2Launch 版本。
	  • Ec2SetPassword - 產生新的管理員密碼。
	
	• 網路界面
	  • DHCP Service Startup (DHCP 服務啟動) - 偵測是否啟用 DHCP 服務。

選項	描述
	• Ethernet detail (乙太網路詳細資訊) - 如果偵測到，顯示網路驅動程式版本的相關資訊。 • DHCP on Ethernet (乙太網路上的 DHCP) - 偵測是否啟用 DHCP。 • 磁碟簽章狀態 • Signature on disk (磁碟上的簽章) 和 Signature on Boot Configuration Database (BCD) (開機組態資料庫 (BCD) 上的簽章)- 偵測磁碟簽章和 BCD 簽章是否相同。如果值不同，EC2Rescue 會嘗試使用 BCD 上的簽章覆寫磁碟簽章。
還原	執行下列動作之一： • Last Known Good Configuration (上一個已知的良好組態) - 嘗試將執行個體開機為上一個已知的可開機狀態。 • Restore registry from backup (從備份還原登錄) - 從 `\Windows\System32\config\RegBack` 還原登錄。
擷取日誌	允許您擷取執行個體的日誌用於分析。

EC2Rescue for Windows Server 可從作用中和離線的執行個體收集下列資料：

項目	描述
事件日誌	收集應用程式、系統和 EC2Config 事件日誌。
登錄	收集 SYSTEM 和 SOFTWARE Hive。
Windows Update 日誌	收集 Windows Update 產生的日誌檔案。

項目	描述
	Note 在 Windows Server 2016 年及更新版本中，日誌會以 Event Tracing for Windows (ETW) 格式收集。
Sysprep 日誌	收集 Windows 系統準備工具產生的日誌檔案。
驅動程式設定日誌	收集 Windows SetupAPI 日誌 (`setupapi.dev.log` 和 `setupapi.setup.log`)。
開機組態	收集 `HKEY_LOCAL_MACHINE\BCD00000000` Hive。
記憶體傾印	收集執行個體上所有的記憶體傾印檔案。
EC2Config 檔案	收集 EC2Config 服務產生的日誌檔案。
EC2Launch 檔案	收集 EC2Launch 指令碼產生的日誌檔案。
SSM Agent 檔案	收集 SSM Agent 和修補程式管理員日誌產生的日誌檔案。
EC2 ElasticGPUs 檔案	收集與 Elastic GPU 相關的事件日誌。
ECS	收集與 Amazon ECS 相關的日誌。
CloudEndure	收集與 CloudEndure 代理程式相關的日誌檔案。
AWS MGN 或 DRS 日誌檔案的複寫代理程式	收集與 AWS Transform MGN 或 相關的日誌檔案 AWS 彈性災難復原。

EC2Rescue for Windows Server 可從作用中執行個體收集下列其他資料：

項目	描述
系統資訊	收集 MSInfo32。
群組政策結果	收集 Group Policy (群組政策) 報告。

分析離線的執行個體

Offline Instance (離線的執行個體) 選項對偵錯 Windows 執行個體的開機問題很有幫助。

在離線的執行個體上執行動作

1. 從工作中的 Windows Server 執行個體，下載 EC2Rescue for Windows Server 工具並解壓縮檔案。

   您可以執行下列 PowerShell 命令來下載 EC2Rescue，而不需要變更 Internet Explorer 增強式安全性設定 (ESC)：

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   此命令會將 EC2Rescue .zip 檔案下載到目前登入使用者的桌面。

   > **Note**
   >
   > 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

2. 如果故障的執行個體尚未停止，請停止它。
3. 從故障的執行個體分離 EBS 根磁碟區，再將此磁碟區連接到已安裝EC2Rescue for Windows Server 的工作中 Windows 執行個體。

4. 在工作中的執行個體上執行 EC2Rescue for Windows Server 工具並選擇 Offline Instance (離線的執行個體)。
5. 選取新掛載磁碟區上的磁碟，然後選擇 Next (下一步)。
6. 確認選取磁碟，然後選擇 Yes (是)。
7. 選擇要執行的離線執行個體選項並選擇 Next (下一步)。

EC2Rescue for Windows Server 工具會掃描磁碟區，並根據選取的日誌檔案收集故障診斷資訊。

收集作用中執行個體的資料

您可收集作用中執行個體的日誌和其他資料。

收集作用中執行個體的資料

1. 連接至 Windows 執行個體。
2. 將 EC2Rescue for Windows Server 工具下載到您的 Windows 執行個體並解壓縮檔案。

   您可以執行下列 PowerShell 命令來下載 EC2Rescue，而不需要變更 Internet Explorer 增強式安全性設定 (ESC)：

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   此命令會將 EC2Rescue .zip 檔案下載到目前登入使用者的桌面。

   > **Note**
   >
   > 如果您在下載檔案時收到錯誤，且您使用 Windows Server 2016 或更早版本，則 PowerShell 終端機可能需要啟用 TLS 1.2。您可以使用下列命令為目前的 PowerShell 工作階段啟用 TLS 1.2，然後再試一次：
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

3. 開啟 EC2Rescue for Windows Server 應用程式並接受授權合約。
4. 選擇 Next (下一步)、Current instance (目前的執行個體)、Capture logs (擷取日誌)。

5. 選取要收集的資料項目並選擇 Collect... (收集)。閱讀警告內容，然後選擇 Yes (是) 繼續。
6. 為 ZIP 檔案選擇檔案名稱及位置，然後選擇 Save (儲存)。
7. 完成 EC2Rescue for Windows Server 之後，選擇 Open Containing Folder (開啟包含的資料夾) 檢視 ZIP 檔案。
8. 選擇 Finish (完成)。

使用 EC2Rescue CLI 來疑難排解受損的 Windows 執行個體

EC2Rescue for Windows Server 命令列界面 (CLI) 讓您透過編寫程式的方式執行 EC2Rescue for Windows Server 外掛程式 (稱為「動作」)。

EC2Rescue for Windows Server 工具有兩種執行模式：

- /online—這可讓您對 EC2Rescue for Windows Server 安裝所在的執行個體採取動作，例如收集日誌檔案。
- /offline:<device_id>—這可讓您對連線到不同 Amazon EC2 Windows 執行個體的離線根磁碟區採取動作，此執行個體已安裝 EC2Rescue for Windows Server。

將 EC2Rescue for Windows Server 工具下載到您的 Windows 執行個體並解壓縮檔案。您可以使用下列命令來檢視說明檔案：

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server 會議主持人可在 Amazon EC2 Windows 執行個體上執行下列動作：

- 收集動作
- 救援動作
- 還原動作

收集動作

> **Note**
>
> 您可收集所有日誌、完整的日誌群組，或群組內的個別日誌。

EC2Rescue for Windows Server 可從作用中和離線的執行個體收集下列資料。

日誌群組	可用日誌	描述
`all`		收集所有可用的日誌。
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	收集應用程式、系統和 EC2Config 事件日誌。
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	收集執行個體上所有的記憶體傾印檔案。
`ec2config`	• `'Log Files'` • `'Configuration Files'`	收集 EC2Config 服務產生的日誌檔案。
`ec2launch`	• `'Logs'` • `'Config'`	收集 EC2Launch 指令碼產生的日誌檔案。
`ssm-agent`	• `'Log Files'` • `'Patch Baseline Logs'` • `'InstanceData'`	收集 SSM Agent 和修補程式管理員日誌產生的日誌檔案。
`sysprep`	`'Log Files'`	收集 Windows 系統準備工具產生的日誌檔案。
`driver-setup`	• `'SetupAPI Log Files'` • `'DPInst Log File'` • `'AWS PV Setup Log File'`	收集 Windows SetupAPI 日誌 (`setupapi.dev.log` 和 `setupapi.setup.log`)。
`registry`	• `'SYSTEM'` • `'SOFTWARE'` • `'BCD'`	收集 `SYSTEM` 和 `SOFTWARE` Hive。

日誌群組	可用日誌	描述
egpu	• 'Event Log' • 'System Files'	收集與 Elastic GPU 相關的事件日誌。
boot-config	'BCDEDIT Output'	收集 HKEY_LOCAL_MACHINE\BCD00000000 Hive。
windows-update	'Log Files'	收集 Windows Update 產生的日誌檔案。 **Note** 在 Windows Server 2016 年及更新版本中，日誌會以 Event Tracing for Windows (ETW) 格式收集。
cloudendure	• 'Migrate Script Logs' • 'Driver Logs' • 'CloudEndure File List'	收集與 CloudEndure 代理程式相關的日誌檔案。

EC2Rescue for Windows Server 可從作用中執行個體收集下列其他資料。

日誌群組	可用日誌	描述
system-info	'MSInfo32 Output'	收集 MSInfo32。
gpresult	'GPResult Output'	收集 Group Policy (群組政策) 報告。

以下為可用的選項：

- /output:<outputFilePath> - 需要目標檔案路徑位置，才能以 zip 格式儲存收集的日誌檔案。

- /no-offline - 離線模式中使用的選用屬性。完成動作後不會將磁碟區設為離線。
- /no-fix-signature - 離線模式中使用的選用屬性。完成動作後不會修正可能的磁碟簽章衝突。

範例

以下是使用EC2Rescue for Windows Server CLI 的範例。

線上模式範例

收集所有可用的日誌：

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

只收集特定的日誌群組：

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

收集日誌群組內的個別日誌：

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI
 Log Files' /output:<outputFilePath>
```

離線模式範例

收集 EBS 磁碟區中所有可用的日誌。磁碟區依 device_id 值指定。

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

只收集特定的日誌群組：

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

救援動作

EC2Rescue for Windows Server 可以偵測並解決下列服務設定的問題：

服務群組	可用動作	描述
all		

服務群組	可用動作	描述
system-time	'RealTimeIsUniversal'	系統時間 • RealTimeisUniversal - 偵測是否啟用 RealTimeisUniversal 登錄機碼。如果停用，當時區值不是設為 UTC 時，Windows 系統時間會產生差異。
firewall	• 'Domain networks' • 'Private networks' • 'Guest or public networks'	Windows 防火牆 • Domain networks (網域網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。 • Private networks (私有網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。 • Guest or public networks (訪客或公有網路) - 偵測此 Windows 防火牆描述檔為啟用或停用。
rdp	• 'Service Start' • 'Remote Desktop Connections' • 'TCP Port'	遠端桌面 • Service Start (服務啟動) - 偵測是否啟用遠端桌面服務。 • Remote Desktop Connections (遠端桌面連線) - 偵測是否啟用此服務。 • TCP Port (TCP 連接埠) - 偵測遠端桌面服務正在接聽哪一個連接埠。

服務群組	可用動作	描述
ec2config	• 'Service Start' • 'Ec2SetPassword' • 'Ec2HandleUserData'	EC2Config • Service Start (服務啟動) - 偵測是否啟用 EC2Config 服務。 • Ec2SetPassword - 產生新的管理員密碼。 • Ec2HandleUserData - 允許您下次開機執行個體時執行使用者資料指令碼。
ec2launch	'Reset Administrator Password'	產生新的 Windows 管理員密碼。
network	'DHCP Service Startup'	網路界面 • DHCP Service Startup (DHCP 服務啟動) - 偵測是否啟用 DHCP 服務。

以下為可用的選項：

- /level:<level> - 動作應該觸發的檢查層級選用屬性。允許的值為：information、warning、error、all。預設會設定為 error。
- /check-only - 會產生報告但不修改離線磁碟區的選用屬性。

> **Note**
>
> 如果 EC2Rescue for Windows Server 偵測到可能的磁碟簽章衝突，它會在離線處理程序預設完成後更正簽章，即使您使用 /check-only 選項也是如此。您必須使用 /no-fix-signature 選項來防止更正。

- /no-offline - 在動作完成後防止將磁碟區設為離線的選用屬性。
- /no-fix-signature - 完成動作後不會修正可能磁碟簽章衝突的選用屬性。

救援範例

以下是使用EC2Rescue for Windows Server CLI 的範例。磁碟區使用 device_id 值指定。

嘗試修正磁碟區中找到的所有問題：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

嘗試修正磁碟區服務群組內的所有問題：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

嘗試修正磁碟區服務群組內的特定項目：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

指定磁碟區中嘗試修正的多個問題：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

還原動作

EC2Rescue for Windows Server 可以偵測並解決下列服務設定的問題：

服務群組	可用的動作	描述
還原上一個已知的良好組態	`lkgc`	Last Known Good Configuration (上一個已知的良好組態) - 嘗試將執行個體開機為上一個已知的可開機狀態。
從最新的備份還原 Windows 登錄	`regback`	Restore registry from backup (從備份還原登錄) - 從 `\Windows\System32\config\RegBack` 還原登錄。

以下為可用的選項：

- /no-offline—在動作完成後防止將磁碟區設為離線的選用屬性。
- /no-fix-signature—完成動作後不會修正可能磁碟簽章衝突的選用屬性。

還原範例

以下是使用EC2Rescue for Windows Server CLI 的範例。磁碟區使用 device_id 值指定。

還原磁碟區中上一個已知的良好組態：

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

在磁碟區上還原上一個 Windows 登錄備份：

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

使用 EC2Rescue 和 Systems Manager 來疑難排解受損的 Windows 執行個體

支援 為您提供 Systems Manager Run Command 文件，以與已啟用 Systems Manager 的執行個體連接，以執行適用於 Windows Server 的 EC2Rescue。執行命令文件稱為 `AWSSupport-RunEC2RescueForWindowsTool`。

此 Systems Manager 執行命令文件會執行下列任務：

- 下載並驗證EC2Rescue for Windows Server。
- 匯入 PowerShell 模組以緩解您與工具的互動。
- 使用提供的命令和參數執行 EC2RescueCmd。

Systems Manager 執行命令文件接受三個參數：

- Command (命令)—EC2Rescue for Windows Server 動作。目前允許的值為：
  - ResetAccess—重設本機管理員的密碼。會重設目前執行個體的本機管理員密碼，隨機產生的密碼則以 `/EC2Rescue/Password/<INSTANCE_ID>` 安全存放在參數存放區中。如果您選取此動作但不提供任何參數，則密碼會使用預設的 KMS 金鑰 自動加密。或者，您可在 Parameters (參數) 中指定 KMS 金鑰 ID，使用您自己的金鑰加密密碼。
  - CollectLogs—執行 EC2Rescue for Windows Server 搭配 `/collect:all` 動作。如果您選取此動作，`Parameters` 必須包含要將日誌上傳到該處的 Amazon S3 儲存貯體名稱。

- FixAll—執行 EC2Rescue for Windows Server 搭配 `/rescue:all` 動作。如果您選取此動作，`Parameters` 必須包含要救援的區塊型儲存設備名稱。
- Parameters (參數)—針對指定命令傳遞的 PowerShell 參數。

要求

要執行 ResetAccess 動作，您的 Amazon EC2 執行個體必須連接政策，以授予將加密密碼寫入參數存放區的許可。將此政策連接到相關 IAM 角色後，請等候數分鐘再嘗試重設執行個體的密碼。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    }
  ]
}
```

若您使用的是自訂 KMS 金鑰，而非預設 KMS 金鑰，請改用此政策。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
```

```
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<kmskeyid>"
      ]
    }
  ]
}
```

檢視文件的 JSON

下列程序說明如何檢視本文件的 JSON。

檢視 Systems Manager 執行命令文件的 JSON

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，展開變更管理工具並選擇文件。
3. 在搜尋列中，輸入 `AWSSupport-RunEC2RescueForWindowsTool`，然後選取 `AWSSupport-RunEC2RescueForWindowsTool` 文件。
4. 選擇 Content (內容) 索引標籤。

範例

以下為如何使用 AWS CLI以 Systems Manager 執行命令文件來執行 EC2Rescue for Windows Server 的範例。如需使用 傳送命令的詳細資訊 AWS CLI，請參閱 send-mand。

範例

- 嘗試修正離線根磁碟區內的所有已識別問題
- 收集目前 Amazon EC2 Windows 執行個體的日誌

- 重設本機管理員密碼

嘗試修正離線根磁碟區內的所有已識別問題

嘗試修正在連線到 Amazon EC2 Windows 執行個體之離線根磁碟區中找到的問題：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=FixAll, Parameters='xvdf'" --output
 text
```

收集目前 Amazon EC2 Windows 執行個體的日誌

收集目前線上 Amazon EC2 Windows 執行個體的所有日誌，然後將它們上傳至 Amazon S3 儲存貯體：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=CollectLogs, Parameters='amzn-s3-
demo-bucket'" --output text
```

重設本機管理員密碼

下列範例示範您可用以重設本機管理員密碼的方法。輸出會提供參數存放區連結，您可在此找到隨機產生的安全密碼，然後以本機管理員身分使用您的 Amazon EC2 Windows 執行個體 RDP。

使用預設的 AWS KMS key 別名/aws/ssm 重設線上執行個體的本機管理員密碼：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess" --output text
```

使用 KMS 金鑰 重設線上執行個體的本機管理員密碼：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess, Parameters=a133dc3c-
a2g4-4fc6-a873-6c0720104bf0" --output text
```

> **Note**
>
> 在此範例中，KMS 金鑰 為 a133dc3c-a2g4-4fc6-a873-6c0720104bf0。

EC2 Serial Console for 執行個體

使用 EC2 序列主控台，您可以存取 Amazon EC2 執行個體的序列埠，您可以使用此連接埠來疑難排解開機、網路設定和其他問題。序列主控台不需要您的執行個體具有任何網路功能。使用序列主控台，您可以為執行個體輸入命令，就好像鍵盤和監視器直接連接到執行個體的序列埠一樣。序列主控台工作階段在執行個體重新開機和停止期間會持續。在重新開機期間，您可以從一開始檢視所有的開機訊息。

依預設，無法存取序列主控台。您的組織必須授與對序列主控台的帳戶存取權，並設定 IAM 政策，以授與使用者對序列主控台的存取權限。可透過使用執行個體 ID、資源標籤和其他 IAM 控制桿，以細微層級控制序列主控台存取。如需詳細資訊，請參閱設定對 EC2 序列主控台的存取。

您可以透過使用 EC2 主控台或 AWS CLI存取序列主控台。

序列主控台可供免費使用。

主題

- EC2 序列主控台的先決條件
- 設定對 EC2 序列主控台的存取
- 連線至 EC2 序列主控台
- 終止與 EC2 序列主控台的連線
- 使用 EC2 序列主控台進行 Amazon EC2 執行個體疑難排解

EC2 序列主控台的先決條件

若要連線 EC2 序列主控台和使用所選工具來進行疑難排解，必須滿足下列必要條件：

- AWS 區域
- Wavelength 區域
- AWS Outpost
- 本機區域
- 執行個體類型
- 授與存取權
- 支援瀏覽器型用戶端
- 執行個體狀態
- Amazon EC2 Systems Manager

- 設定所選的疑難排解工具

AWS 區域

除了 AWS 區域 之外，所有 都支援 AWS European Sovereign Cloud (Germany)。

Wavelength 區域

不支援。

AWS Outpost

支援所有 AWS Outpost。

本機區域

所有本地區域皆支援。

執行個體類型

支援的執行個體類型：

- Linux
  - 在 Nitro 系統上建置的所有虛擬化執行個體。
  - 所有裸機執行個體，除了：
    - 一般用途：a1.metal、mac1.metal、mac2.metal、mac2-m1ultra.metal、mac2-m2.metal、mac2-m2pro.metal、mac-m3ultra.metalmac-m4.metal、、 mac-m4max.metal和 mac-m4pro.metal
    - 加速運算：g5g.metal
    - 記憶體最佳化：u-6tb1.metal、u-9tb1.metal、u-18tb1.metal、 u-12tb1.metal和 u-24tb1.metal
- Windows
  - 在 Nitro 系統上建置的所有虛擬化執行個體。
  - 所有裸機執行個體，除了：
    - 一般用途：a1.metal、mac1.metal、mac2.metal、mac2-m1ultra.metal、mac2-m2.metal、mac2-m2pro.metal、mac-m3ultra.metalmac-m4.metal、、 mac-m4max.metal和 mac-m4pro.metal
    - 加速運算：g5g.metal

- 記憶體最佳化：u-6tb1.metal、u-9tb1.metal、u-18tb1.metal、 u-12tb1.metal和 u-24tb1.metal

授與存取權

您必須完成設定工作，才能向 EC2 序列主控台授與存取權。如需詳細資訊，請參閱設定對 EC2 序列主控台的存取。

支援瀏覽器型用戶端

若要使用以瀏覽器為基礎的用戶端連線到序列主控台，您的瀏覽器必須支援 WebSocket。如果您的瀏覽器不支援 WebSocket，請使用您自己的金鑰和 SSH 用戶端連線到序列主控台。

瀏覽器型用戶端會在連接埠 443 連線至您區域的 EC2 Instance Connect 代理端點。端點格式取決於分割區：

- 商業 AWS 區域和 AWS GovCloud (US) 區域使用端點格式：prod.*region*.oneclickv2-proxy.ec2.aws.dev。
- 中國區域使用端點格式：prod.*region*.oneclickv2-proxy.ec2.a2z.org.cn。

下列範例顯示每個分割區的端點：

- 對於美國東部（維吉尼亞北部）區域，端點為 prod.us-east-1.oneclickv2-proxy.ec2.aws.dev。
- 對於中國（北京）區域，端點為 prod.cn-north-1.oneclickv2-proxy.ec2.a2z.org.cn。

> **Note**
>
> 您的瀏覽器必須能夠到達此端點。如果防火牆或 VPN 封鎖連線，連線會失敗。

執行個體狀態

必須為 running。

若執行個體處於 pending、stopping、stopped、shutting-down，或 terminated 狀態，將無法連線到序列主控台。

如需執行個體狀態的詳細資訊，請參閱 Amazon EC2 執行個體狀態變更。

Amazon EC2 Systems Manager

如果執行個體使用 Amazon EC2 Systems Manager，則必須在執行個體上安裝 SSM Agent 3.0.854.0 版或更新版本。如需有關 SSM Agent 的資訊，請參閱 AWS Systems Manager 使用者指南中的使用 SSM Agent。

設定所選的疑難排解工具

若要使用序列主控台對執行個體進行疑難排解，您可以使用 Linux 執行個體上的GRUB 或 SysRq，以及 Windows 執行個體上的特殊管理主控台 (SAC)。在使用這些工具之前，您必須先對要使用工具所在的每個執行個體執行設定步驟。

使用執行個體作業系統的指示來設定您所選之疑難排解工具。

(Linux 執行個體) 設定 GRUB

若要設定 GRUB，請根據用來啟動執行個體的 AMI 選擇下列其中一個程序。

Amazon Linux 2

在 Amazon Linux 2 執行個體上設定 GRUB

1. 使用 SSH 連線至您的 Linux 執行個體
2. 在 `/etc/default/grub` 中新增或變更下列選項：
   - 設定 `GRUB_TIMEOUT=1`。
   - 新增 `GRUB_TERMINAL="console serial"`。
   - 新增 `GRUB_SERIAL_COMMAND="serial --speed=115200"`。

   以下是 `/etc/default/grub` 的範例。您可能需要根據系統設定來變更組態。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0"
GRUB_TIMEOUT=1
GRUB_DISABLE_RECOVERY="true"
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

3. 執行下列命令來套用更新的組態。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Ubuntu

在 Ubuntu 執行個體上設定 GRUB

1. 連線到您的執行個體。
2. 在 /etc/default/grub.d/50-cloudimg-settings.cfg 中新增或變更下列選項：
   - 設定 GRUB_TIMEOUT=1。
   - 新增 GRUB_TIMEOUT_STYLE=menu。
   - 新增 GRUB_TERMINAL="console serial"。
   - 移除 GRUB_HIDDEN_TIMEOUT。
   - 新增 GRUB_SERIAL_COMMAND="serial --speed=115200"。

   以下是 /etc/default/grub.d/50-cloudimg-settings.cfg 的範例。您可能需要根據系統設定來變更組態。

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=1
GRUB_TIMEOUT_STYLE=menu

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
 nvme_core.io_timeout=4294967295"

# Set the grub console type
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed 115200"
```

3. 執行下列命令來套用更新的組態。

```
[ec2-user ~]$ sudo update-grub
```

RHEL

在 RHEL 執行個體上設定 GRUB

1. 連線到您的執行個體。
2. 在 /etc/default/grub 中新增或變更下列選項：
   - 移除 GRUB_TERMINAL_OUTPUT。
   - 新增 GRUB_TERMINAL="console serial"。
   - 新增 GRUB_SERIAL_COMMAND="serial --speed=115200"。

   以下是 /etc/default/grub 的範例。您可能需要根據系統設定來變更組態。

   ```
   GRUB_TIMEOUT=1
   GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
   GRUB_DISABLE_RECOVERY="true"
   GRUB_ENABLE_BLSCFG=true
   GRUB_TERMINAL="console serial"
   GRUB_SERIAL_COMMAND="serial --speed=115200"
   ```
3. 執行下列命令來套用更新的組態。

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
   ```

   針對 RHEL 9.2 和較早版本，請使用下列命令。

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

CentOS

對於使用 CentOS AMI 啟動的執行個體，預設會針對序列主控台設定 GRUB。

以下是 /etc/default/grub 的範例。您的組態可能會視您的系統設定而有所不同。

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_CMDLINE_LINUX="console=tty0 crashkernel=auto console=ttyS0,115200"
GRUB_DISABLE_RECOVERY="true"
```

(Linux 執行個體) 設定 SysRq

若要設定 SysRq，您可以針對目前的開機週期啟用 SysRq 命令。若要使設定持續性，您也可以啟用 SysRq 命令以供後續開機使用。

為目前開機週期啟用所有 SysRq 命令

1. 連線到您的執行個體。
2. 執行下列命令。

   ```
   [ec2-user ~]$ sudo sysctl -w kernel.sysrq=1
   ```

   此設定會在下次重新開機時清除。

為後續開機啟用所有 SysRq 命令

1. 建立文件 /etc/sysctl.d/99-sysrq.conf 並在您最喜歡的編輯器中開啟。

   ```
   [ec2-user ~]$ sudo vi /etc/sysctl.d/99-sysrq.conf
   ```

2. 新增以下這一行。

   ```
   kernel.sysrq=1
   ```

3. 重新啟動執行個體以套用變更。

   ```
   [ec2-user ~]$ sudo reboot
   ```

4. 出現 login 提示時，輸入您先前設定之密碼型使用者的使用者名稱，然後按 Enter。

5. 出現 Password 提示時，輸入密碼，然後按 Enter。

(Windows 執行個體) 啟用 SAC 和開機功能表

> **Note**
>
> 如果您在執行個體上啟用 SAC，則依賴密碼擷取的 EC2 服務將無法從 Amazon EC2 主控台運作。Amazon EC2 上的 Windows 啟動代理程式 (EC2Config、EC2Launch v1 和 EC2Launch v2) 倚賴序列主控台執行各種任務。在執行個體上啟用 SAC 時，這些任務不會成功執行。如需 Amazon EC2 啟動代理程式的詳細資訊，請參閱 the section called “設定 Windows 執行個體”。如果您啟用 SAC，可以稍後停用它。如需詳細資訊，請參閱停用 SAC 和開機選單。

使用下列其中一種方法來啟用執行個體上的 SAC 和開機功能表。

PowerShell

在 Windows 執行個體上啟用 SAC 和開機功能表

1. 連線到您的執行個體，並從提高權限的 PowerShell 命令列執行下列步驟。
2. 啟用 SAC。

```
bcdedit /ems '{current}' on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 啟用開機功能表。

```
bcdedit /set '{bootmgr}' displaybootmenu yes
bcdedit /set '{bootmgr}' timeout 15
bcdedit /set '{bootmgr}' bootems yes
```

4. 透過重新啟動執行個體來套用更新的組態。

```
shutdown -r -t 0
```

Command prompt

在 Windows 執行個體上啟用 SAC 和開機功能表

1. 連接到您的執行個體，並從命令提示字元執行下列步驟。
2. 啟用 SAC。

```
bcdedit /ems {current} on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 啟用開機功能表。

```
bcdedit /set {bootmgr} displaybootmenu yes
bcdedit /set {bootmgr} timeout 15
bcdedit /set {bootmgr} bootems yes
```

4. 透過重新啟動執行個體來套用更新的組態。

```
shutdown -r -t 0
```

設定對 EC2 序列主控台的存取

若要設定對序列主控台的存取權，您必須在帳戶層級授與序列主控台存取權，然後設定 IAM 政策以授予使用者存取權。針對 Linux 執行個體，您必須在每個執行個體上設定以密碼為基礎的使用者，以便使用者可以使用序列主控台進行疑難排解。

EC2 序列主控台會使用虛擬序列連接埠連線與執行個體互動。此連線與執行個體 VPC 無關，因此即使發生開機失敗或網路組態問題，您也可使用執行個體作業系統並執行疑難排解工具。由於此連線位於 VPC 網路外部，EC2 序列主控台不會使用執行個體安全群組或子網路網路 ACL 來授權執行個體的流量。

開始之前

確認符合先決條件。

目錄

- EC2 序列主控台的存取層級
- 管理 EC2 序列主控台的帳戶存取
- 設定 EC2 序列主控台存取的 IAM 政策

- 在 Linux 執行個體上設定作業系統使用者密碼

EC2 序列主控台的存取層級

依預設，帳戶層級無法存取序列主控台。您需要在帳戶層級明確授予對序列主控台的存取權。如需詳細資訊，請參閱 管理 EC2 序列主控台的帳戶存取。

您可以使用服務控制政策 (SCP) 來允許對組織內序列主控台的存取。然後，您可以使用 IAM 政策控制存取，在使用者層級進行精細的存取控制。透過組合使用 SCP 和 IAM 政策，您擁有對序列主控台不同層級的存取控制。

組織層級

您可以使用服務控制政策 (SCP) 來允許組織內成員帳戶存取序列主控台。如需 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的服務控制政策。

執行個體層級

您可以使用 IAM PrincipalTag 和 ResourceTag 結構，以及依其 ID 指定執行個體來設定序列主控台存取政策。如需詳細資訊，請參閱設定 EC2 序列主控台存取的 IAM 政策。

使用者層級

您可以設定 IAM 政策，以允許或拒絕指定的使用者將 SSH 公開金鑰推送至特定執行個體的序列主控台服務的權限，從而在使用者層級設定存取。如需詳細資訊，請參閱設定 EC2 序列主控台存取的 IAM 政策。

作業系統層級 (僅限 Linux 執行個體)

您可以在客體作業系統層級設定使用者密碼。在某些使用案例下，這可讓您存取序列主控台。不過，若要監控日誌，您不需要以密碼為基礎的使用者。如需詳細資訊，請參閱 在 Linux 執行個體上設定作業系統使用者密碼。

管理 EC2 序列主控台的帳戶存取

依預設，帳戶層級無法存取序列主控台。您需要在帳戶層級明確授予對序列主控台的存取權。

此設定是在帳戶層級配置，可直接在帳戶中設定，或使用宣告式政策設定。它必須在您要授予序列主控台存取權的每個 AWS 區域 中設定。您可使用宣告式政策同時在多個區域及多個帳戶套用設定。使用宣告式政策時，您無法直接在帳戶中修改設定。本主題說明如何直接在帳戶內配置設定。如需使用宣告式政策的相關資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

內容

- 授予使用者管理帳戶存取的許可
- 檢視序列主控台的帳戶存取狀態
- 授與序列主控台的帳戶存取權
- 拒絕帳戶存取序列主控台

授予使用者管理帳戶存取的許可

若要允許使用者管理對 EC2 序列主控台的帳戶存取權，您必須授予他們所需的 IAM 許可。

下列政策會授與檢視帳戶狀態，以及允許和阻止帳戶存取 EC2 序列主控台的權限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:EnableSerialConsoleAccess",
                "ec2:DisableSerialConsoleAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

如需詳細資訊，請參閱「IAM 使用者指南」中的建立 IAM 政策。

檢視序列主控台的帳戶存取狀態

Console

檢視序列主控台的帳戶存取

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡片的設定下，選擇 EC2 序列主控台。
4. 在 EC2 序列主控台索引標籤上，EC2 序列主控台存取的值為允許或禁止。

AWS CLI

檢視序列主控台的帳戶存取

使用 get-serial-console-access-status 命令。

```
aws ec2 get-serial-console-access-status
```

以下為範例輸出。若值為 `true`，表示允許帳戶存取序列主控台。

```
{
    "SerialConsoleAccessEnabled": true,
    "ManagedBy": "account"
}
```

`ManagedBy` 欄位指示配置設定的實體。可能的值為 `account` (直接設定) 或 `declarative-policy`。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

PowerShell

檢視序列主控台的帳戶存取

使用 Get-EC2SerialConsoleAccessStatus cmdlet。

```
Get-EC2SerialConsoleAccessStatus -Select *
```

以下為範例輸出。若值為 `True`，表示允許帳戶存取序列主控台。

```
ManagedBy SerialConsoleAccessEnabled
--------- --------------------------
account   True
```

`ManagedBy` 欄位指示配置設定的實體。可能的值為 `account` (直接設定) 或 `declarative-policy`。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策。

授與序列主控台的帳戶存取權

Console

授與帳戶存取序列主控台

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡片的設定下，選擇 EC2 序列主控台。
4. 選擇管理。
5. 若要允許存取帳戶中所有執行個體的 EC2 序列主控台，請選取運許核取方塊。
6. 選擇更新。

AWS CLI

授與帳戶存取序列主控台

使用 enable-serial-console-access 命令。

```
aws ec2 enable-serial-console-access
```

以下為範例輸出。

```
{
    "SerialConsoleAccessEnabled": true
}
```

PowerShell

授與帳戶存取序列主控台

使用 Enable-EC2SerialConsoleAccess cmdlet。

```
Enable-EC2SerialConsoleAccess
```

以下為範例輸出。

```
True
```

拒絕帳戶存取序列主控台

Console

拒絕帳戶存取序列主控台

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在帳戶屬性卡片的設定下，選擇 EC2 序列主控台。
4. 選擇管理。
5. 若要阻止存取帳戶中所有執行個體的 EC2 序列主控台，請清除允許核取方塊。
6. 選擇更新。

AWS CLI

拒絕帳戶存取序列主控台

使用 disable-serial-console-access 命令。

```
aws ec2 disable-serial-console-access
```

以下為範例輸出。

```
{
    "SerialConsoleAccessEnabled": false
}
```

PowerShell

拒絕帳戶存取序列主控台

使用 Disable-EC2SerialConsoleAccess cmdlet。

```
Disable-EC2SerialConsoleAccess
```

以下為範例輸出。

```
False
```

設定 EC2 序列主控台存取的 IAM 政策

依預設，使用者無法存取序列主控台。您的組織必須設定 IAM 政策，以授予使用者所需的存取權。如需詳細資訊，請參閱「IAM 使用者指南」中的建立 IAM 政策。

若要存取序列主控台，請建立包含 `ec2-instance-connect:SendSerialConsoleSSHPublicKey` 動作的 JSON 政策文件。此動作會授予使用者將公有金鑰推送至序列主控台服務的許可，此服務會啟動序列主控台工作階段。建議限制對特定 EC2 執行個體的存取。否則，擁有此許可的所有使用者都可以連線到所有 EC2 執行個體的序列主控台。

範例 IAM 政策

- 明確允許存取序列主控台
- 明確拒絕存取序列主控台
- 使用資源標籤來控制對序列主控台的存取

明確允許存取序列主控台

依預設，沒有人可以存取序列主控台。若要授與對序列主控台的存取權，您必須設定政策以明確允許存取。建議您設定限制存取特定執行個體的政策。

下列政策允許存取特定執行個體的序列主控台，並依其執行個體 ID 進行識別。

請注意，`DescribeInstances`、`DescribeInstanceTypes`、和 `GetSerialConsoleAccessStatus` 動作不支援資源層級許可，因此為這些動作指定所有資源，以 * (星號) 表示。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
             "Resource": "*"
```

```
        },
        {
            "Sid": "AllowinstanceBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

明確拒絕存取序列主控台

下列 IAM 政策允許存取所有執行個體的序列主控台 (以 * (星號) 表示)，並明確拒絕存取特定執行個體的序列主控台 (依其 ID 進行識別)。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
```

```
        }
    ]
}
```

使用資源標籤來控制對序列主控台的存取

您可以使用資源標籤來控制對執行個體序列主控台的存取。

屬性型存取控制是一種授權策略，可根據可連接到使用者 AWS 和資源的標籤定義許可。例如，下列政策允許使用者啟動執行個體的序列主控台連線，只有當該執行個體的資源標籤和主參與者的標籤具有相同的標籤鍵 SerialConsole 值時，才能啟動執行個體的序列主控台連線。

如需使用標籤控制對 AWS 資源的存取的詳細資訊，請參閱《IAM 使用者指南》中的控制對 AWS 資源的存取。

請注意，DescribeInstances、DescribeInstanceTypes、和 GetSerialConsoleAccessStatus 動作不支援資源層級許可，因此為這些動作指定所有資源，以 * (星號) 表示。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SerialConsole":
 "${aws:PrincipalTag/SerialConsole}"
                }
            }
        }
    ]
}
```

在 Linux 執行個體上設定作業系統使用者密碼

您無需密碼即可連線到序列主控台。不過，若要使用序列主控台進行 Linux 執行個體疑難排解，執行個體必須具有以密碼為基礎的 OS 使用者。

您可以為任何 OS 使用者 (包括根使用者) 設定密碼。請注意，根使用者可以修改所有檔案，而每個 OS 使用者的許可有限制。

您必須為每個要使用序列主控台的執行個體設定使用者密碼。這是每個執行個體都有的一次性要求。

> **Note**
>
> 根據預設， 提供的 AMIs AWS 不會設定密碼型使用者。如果您使用已設定根使用者密碼的 AMI 啟動執行個體，則可以略過這些指示。

在 Linux 執行個體上設定作業系統使用者密碼

1. 連線到您的執行個體。您可以使用任何方法連線到執行個體，EC2 序列主控台連線方法除外。
2. 若要設定使用者的密碼，請使用 passwd 命令。在下列範例中，使用者是 root。

   ```
   [ec2-user ~]$ sudo passwd root
   ```

   下列為範例輸出。

   ```
   Changing password for user root.
   New password:
   ```

3. 出現 New password 提示時，輸入新密碼。

4. 出現提示時，重新輸入密碼。

連線至 EC2 序列主控台

您可以使用 Amazon EC2 主控台或透過 SSH 連線到 EC2 執行個體的序列控制台。連線到序列主控台後，您可以使用它來疑難排解開機、網路設定和其他問題。如需疑難排解的詳細資訊，請參閱 使用 EC2 序列主控台進行 Amazon EC2 執行個體疑難排解。

考量事項

- 每個執行個體僅支援 1 個使用中的序列主控台連線。
- 除非您終止連線，否則序列主控台連線通常會持續 1 個小時。但是，在系統維護期間，Amazon EC2 將終止序列主控台工作階段的連線。

  連線的持續時間並不是由 IAM 憑證的持續時間決定。即使您的 IAM 憑證過期，連線仍會繼續保留，直到達到序列主控台連線的最長持續時間為止。使用 EC2 序列主控台體驗時，若您的 IAM 憑證過期，請關閉瀏覽器頁面來終止連線。
- 中斷與序列主控台的連線後，需要 30 秒才能銷毀工作階段，以允許新的工作階段。
- 支援的序列主控台連接埠：`ttyS0` (Linux 執行個體) 和 `COM1` (Windows 執行個體)
- 當您連線到序列主控台時，可能會發現執行個體的輸送量略有下降。

主題

- 使用以瀏覽器為基礎的用戶端連接
- 使用您自己的金鑰和 SSH 用戶端來進行連接
- EC2 序列主控台端點和指紋

使用以瀏覽器為基礎的用戶端連接

您可以使用以瀏覽器為基礎的用戶端連線到 EC2 執行個體的序列主控台。您可以在 Amazon EC2 主控台中選取執行個體並選擇連線到序列主控台來執行這項操作。以瀏覽器為基礎的用戶端會處理權限，並提供成功的連線。

EC2 序列控制台適用於大多數瀏覽器，並支援鍵盤和滑鼠輸入。

連線之前，請確認您已完成事前準備。

使用以瀏覽器為基礎的用戶端 (Amazon EC2 主控台) 連線到執行個體的序列埠

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 請選取執行個體並選擇 Actions (動作)、Monitor and troubleshoot (監控和疑難排解)、EC2 Serial Console (EC2 序列主控台)、Connect (連線)。

   或者，請選取執行個體並選擇 Connect (連線)、EC2 Serial Console (EC2 序列主控台)、 Connect (連線)。

   瀏覽器內的終端機視窗隨即開啟。
4. 按 Enter。如果傳回登入提示，表示您已連線到序列主控台。

   如果螢幕保持黑色，您可以使用下列資訊來協助解決連線到序列主控台的問題：

   - 檢查您是否已設定序列主控台的存取權。如需詳細資訊，請參閱設定對 EC2 序列主控台的存取。
   - (僅限 Linux 執行個體) 使用 SysRq 連線到序列主控台。SysRq 不需要您使用瀏覽器型用戶端連線。如需詳細資訊，請參閱(Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解。
   - (僅限 Linux 執行個體) 重新啟動 getty。如果您對執行個體有 SSH 存取權限，然後使用 SSH 連接到您的執行個體，並使用以下命令重新啟動 Getty。

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```
   - 將執行個體重新開機。您可以使用 SysRq (Linux 執行個體)、EC2 主控台或 AWS CLI來為執行個體重新開機。如需詳細資訊，請參閱 (Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解 (Linux 執行個體) 或 重新啟動您的 Amazon EC2 執行個體。
5. (僅限Linux 執行個體) 出現 `login` 提示時，輸入您先前設定的密碼型使用者的使用者名稱，然後按 Enter。
6. (僅限Linux 執行個體) 出現 `Password` 提示時，輸入密碼，然後按 Enter。

使用您自己的金鑰和 SSH 用戶端來進行連接

當您使用序列主控台 API 時，可以使用自己的 SSH 金鑰和從您選擇的 SSH 用戶端連接執行個體。這可讓您從發佈公有的金鑰至執行個體的序列主控台功能受益。

將 SSH 金鑰推送至執行個體後，SSH 連線將不受您為授予使用者 EC2 序列主控台存取權而設定的 IAM 政策約束。

開始之前

確認符合先決條件。

使用 SSH 連線到執行個體的序列主控台

1. 將 SSH 公開金鑰推送到執行個體，以啟動序列控制台工作階段

   使用 send-serial-console-ssh-public-key 命令將 SSH 公開金鑰推送至執行個體。這會啟動序列主控台工作階段。

   如果此執行個體已經啟動序列主控台工作階段，則命令會失敗，因為您一次只能開啟一個工作階段。中斷與序列主控台的連線後，需要 30 秒才能銷毀工作階段，以允許新的工作階段。

   ```
   aws ec2-instance-connect send-serial-console-ssh-public-key \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --serial-port 0 \
       --ssh-public-key file://my_key.pub \
       --region us-east-1
   ```

2. 使用私有金鑰連線到序列主控台

   在從序列主控台服務中移除公開金鑰之前，請使用 ssh 命令連線到序列主控台。在將其移除之前，您有 60 秒時間。

   使用對應於公開金鑰的私有金鑰。

   使用者名稱格式為 `instance-id.port0`，其中包含執行個體 ID 和連接埠 0。在下列範例中，使用者名稱為 `i-001234a4bf70dec41EXAMPLE.port0`。

   每個區域的序列主控台服務端點各不相同。請參閱每個區域端點的 EC2 序列主控台端點和指紋 資料表。在下列範例中，序列主控台服務位於 *us-east-1* 區域。

   ```
   ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-
   connect.us-east-1.aws
   ```

   下列範例使用 `timeout 3600` 將 SSH 工作階段設定為 1 小時後終止。工作階段終止之後，工作階段期間啟動的程序可能會在執行個體上繼續執行。

   ```
   timeout 3600 ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-
   instance-connect.us-east-1.aws
   ```

3. (選擇性) 驗證指紋

   當您第一次連線到序列主控台時，系統會提示您確認指紋。您可以將序列主控台指紋與顯示進行驗證的指紋進行比較。如果這些指紋不符合，可能有人嘗試發動中間人攻擊。如果它們匹配，您可以放心地連接到序列主控台。

   以下指紋適用於 us-east-1 區域的序列控制台服務。如需每個區域的指紋，請參閱 EC2 序列主控台端點和指紋。

   ```
   SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
   ```

   指紋只會在您第一次連線到序列主控台時出現。

4. 按 Enter。如果傳回提示，表示您已連線到序列主控台。

   如果螢幕保持黑色，您可以使用下列資訊來協助解決連線到序列主控台的問題：

   - 檢查您是否已設定序列主控台的存取權。如需詳細資訊，請參閱設定對 EC2 序列主控台的存取。
   - (僅限 Linux 執行個體) 使用 SysRq 連線到序列主控台。SysRq 不需要您使用 SSH 進行連線。如需詳細資訊，請參閱(Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解。
   - (僅限 Linux 執行個體) 重新啟動 getty。如果您對執行個體有 SSH 存取權限，然後使用 SSH 連接到您的執行個體，並使用以下命令重新啟動 Getty。

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```

   - 將執行個體重新開機。您可以使用 SysRq (僅限 Linux 執行個體)、EC2 主控台或 AWS CLI來為執行個體重新開機。如需詳細資訊，請參閱 (Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解 (僅限 Linux 執行個體) 或 重新啟動您的 Amazon EC2 執行個體。

5. (僅限Linux 執行個體) 出現 `login` 提示時，輸入您先前設定的密碼型使用者的使用者名稱，然後按 Enter。
6. (僅限Linux 執行個體) 出現 `Password` 提示時，輸入密碼，然後按 Enter。

EC2 序列主控台端點和指紋

下列項目是 EC2 序列主控台的服務端點和指紋。若要以程式設計方式連線至執行個體的序列主控台，您可使用 EC2 序列主控台端點。每個 AWS 區域的 EC2 序列主控台端點和指紋都是獨一無二的。

區域名稱	區域	Endpoint	指紋
美國東部 (俄亥俄)	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
美國東部 (維吉尼亞北部)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
美國西部 (加利佛尼亞北部)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OHldlcMET8u7QLSX3jmRTRAPFHVtqbyoLZBMUCqiH3Y
美國西部 (奧勒岡)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EMCIe23TqKaBI6yGHainqZcMwqNkDhhAVHa1O2JxVUc
Africa (Cape Town)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RMWWZ2fVePeJUqzjO5jL2KIgXsczoHlz21Ed00biiWI
亞太地區 (香港)	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0Q1lpiXxChoZHplnAkjbP7tkm2xXViC9bJFsjYnifk
亞太區域 (台北)	ap-east-2	ec2-serial-console.ap-east-2.api.aws	SHA256：z1rsF5DE8aQqsHwBr/D40tR2GnyMqnScCUa1z1eFwDQ

區域名稱	區域	Endpoint	指紋
亞太地區 (海德拉巴)	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJgPBSwV4/shN+OPITValoewAuYj15DVW845JEhDKRs
亞太地區 (雅加達)	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfns32XITqL/4O0zIfbx4bZgsYFqy3o8mIk
亞太地區 (馬來西亞)	ap-southeast-5	ec2-serial-console.ap-southeast-5.api.aws	SHA256:cQXTHQMRcqRdIjmAGoAMBSExeoRobYyRwT67yTjnEiA
亞太區域 (紐西蘭)	ap-southeast-6	ec2-serial-console.ap-southeast-6.api.aws	SHA256：wNltdH0gVfM5uHJKjrw+ElFuKoJgalcSKqjCS+lLkv4
亞太地區 (墨爾本)	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Avaq27hFgLvjn5gTSShZ0oV7h90p0GG46wfOeT6ZJvM
亞太區域 (孟買)	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oBLXcYmklqHHEbliARxEgH8IsO51rezTPiSM35BsU40
亞太地區 (大阪)	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/jiBKBnBuFnHr9aXsgEV3G8Tu/vVHFXE/3UcyjsQ

區域名稱	區域	Endpoint	指紋
亞太區域 (首爾)	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrqrI
亞太區域 (新加坡)	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
亞太地區 (悉尼)	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
亞太區域 (泰國)	ap-southeast-7	ec2-serial-console.ap-southeast-7.api.aws	SHA256:KCAZiRYrR1Q2lqsg7vTwixWmvc2wmjVT31XRgSdEfDY
亞太區域 (東京)	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k
加拿大 (中部)	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4

區域名稱	區域	Endpoint	指紋
加拿大西部 (卡加利)	ca-west-1	ec2-serial-console.ca-west-1.api.aws	SHA256:s3rc8Il2xhbhr3iedjJNxGAFLPGOLjx7IxxXrGckk6Q
中國 (北京)	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y
中國 (寧夏)	cn-northwest-1	ec2-serial-console.cn-northwest-1.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
歐洲 (法蘭克福)	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c
歐洲 (愛爾蘭)	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
歐洲 (倫敦)	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8

區域名稱	區域	Endpoint	指紋
歐洲 (米蘭)	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:IC 0kOVJnpgF yBVrxn0A7 n99ecLbXS X95cuuS7X7QK30
Europe (Paris)	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pym eNe8BnFVngY3RPAr/ kxswJUzfrlxeEWs
歐洲 (西班牙)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:Go CW2DFRlu6 69QNxqFxE csR6fZUz/4F4n7T45Z cwoEc
Europe (Stockholm)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tk GFFUVUDvo cDiGSS3Cu 8Gdl6w2uI 32EPNpKFKLwX84
歐洲 (蘇黎世)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8P px2mBMf6W dCw0NUlzKfwM4/IfRz 4OaXFutQXWp6mk
以色列 (特拉維夫)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR 6q8v6kNNP i8+QSFQ4d j5dimNmZP TgwgsM1SNvtYyU

區域名稱	區域	Endpoint	指紋
墨西哥 (中部)	mx-central-1	ec2-serial-console.mx-central-1.api.aws	SHA256:BCuVl13iQNk+CcVnt18Ef4p2ZHUrBBAOxlFetB32GS0
Middle East (Bahrain)	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nPjLLKHu2QnLdUq2kVArsoK5xvPJOMRJKCBzCDqC3k8
中東 (阿拉伯聯合大公國)	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ+l0dFwPeyykB4MPBYhl/XzXNeFSDKBvLE
南美洲 (聖保羅)	sa-east-1	serial-console.ec2-instance-connect.sa-east-1.aws	SHA256:rd2+/32Ognjew1yVIemENaQzC+Botbih62OqAPDq1dI
AWS GovCloud (US-East)	us-gov-east-1	serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com	SHA256:tIwe19GWsoyLClrtvu38YEEh+DHIkqnDcZnmtebvF28
AWS GovCloud (US-West)	us-gov-west-1	serial-console.ec2-instance-connect.us-gov-west-1.amazonaws.com	SHA256:kfOFRWLaOZfB+utbd3bRf8OlPf8nGO2YZLqXZiIw5DQ

終止與 EC2 序列主控台的連線

如果您不再需要連線至執行個體的 EC2 序列主控台，就可以終止其連線。當您終止與序列主控台的連線時，執行個體上執行的任何 shell 工作階段都會繼續執行。如果想要結束 shell 工作階段，您必須先將其結束，然後才能終止與序列主控台的連線。

考量事項

- 除非您終止連線，否則序列主控台連線通常會持續 1 個小時。但是，在系統維護期間，Amazon EC2 將終止序列主控台工作階段的連線。
- 中斷與序列主控台的連線後，需要 30 秒才能銷毀工作階段，以允許新的工作階段。

終止與序列主控台連線的方式取決於用戶端。

瀏覽器型用戶端

若要終止與序列主控台的連線，請關閉序列主控台的瀏覽器內終端機視窗。

標準開啟 SSH 用戶端

若要終止與序列主控台的連線，請使用下列命令關閉 SSH 連線。必須在新行之後立即執行此命令。

```
~.
```

根據您使用的 SSH 用戶端，您用來關閉 SSH 連線的命令可能會有所不同。

使用 EC2 序列主控台進行 Amazon EC2 執行個體疑難排解

透過使用 EC2 序列主控台，您可以連接到執行個體的序列埠來疑難排解開機、網路設定和其他問題。

針對執行個體的作業系統和您在執行個體上設定的工具，使用相關指示。

工具

- (Linux 執行個體) 使用 GRUB 對執行個體進行疑難排解
- (Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解
- (Windows 執行個體) 使用 SAC 為您的執行個體進行疑難排解

先決條件

在開始之前，請確定您已完成先決條件，包括設定您所選的疑難排解工具。

(Linux 執行個體) 使用 GRUB 對執行個體進行疑難排解

GNU GRUB (GNU Grand 統一開機載入程式的簡稱，通常稱為 GRUB) 是大多數 Linux 作業系統的預設開機載入程式。從 GRUB 功能表中，您可以選擇要開機的核心，或修改功能表項目以變更核心的開機方式。這在疑難排解失敗的執行個體時非常有用。

GRUB 功能表會在開機過程中顯示。功能表無法透過一般 SSH 存取，但您可以使用 EC2 序列主控台來存取。

您可以開機至單一使用者模式或緊急模式。單一使用者模式會以較低的執行層級啟動核心。例如，它可能掛載檔案系統，但不會啟動網路，讓您有機會執行修復執行個體所需的維護。緊急模式與單一使用者模式類似，不同之處在於核心以最低的執行層級執行。

開機至單一使用者模式

1. 連線到執行個體的序列主控台。
2. 使用下列命令重新啟動執行個體。

```
[ec2-user ~]$ sudo reboot
```

3. 在重新開機期間，當 GRUB 功能表出現時，按任意鍵停止啟動過程。
4. 在 GRUB 功能表中，使用方向鍵選取要開機的核心，然後按鍵盤上的 e。
5. 使用方向鍵將游標定位在包含核心的行上。該行以 `linux` 或 `linux16` 開頭，取決於用於啟動執行個體的 AMI。對於 Ubuntu，兩行以 `linux` 開頭，必須在下一個步驟中修改這兩行。
6. 在該行的末尾，添加單詞 `single`。

   以下是 Amazon Linux 2 的範例。

```
linux /boot/vmlinuz-4.14.193-149.317.amzn2.aarch64 root=UUID=d33f9c9a-\
dadd-4499-938d-ebbf42c3e499 ro  console=tty0 console=ttyS0,115200n8 net.ifname\
s=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.she\
ll=0 single
```

7. 按 Ctrl+X 以開機進入單一使用者模式。
8. 出現 `login` 提示時，輸入您先前設定之密碼型使用者的使用者名稱，然後按 Enter。
9. 出現 `Password` 提示時，輸入密碼，然後按 Enter。

開機進入緊急模式

遵循與單一使用者模式相同的步驟，但在步驟 6 中新增字詞 `emergency`，而非 `single`。

(Linux 執行個體) 使用 SysRq 對執行個體進行疑難排解

系統要求 (SysRq) 金鑰，有時稱為「神奇 SysRq」，可以用來在 shell 之外直接向核心傳送命令，無論核心正在執行什麼操作，都會回應。例如，如果執行個體已停止回應，您可以使用 SysRq 金鑰告訴核心當機或重新開機。如需詳細資訊，請參閱 Wikipedia 中的神奇 SysRq 金鑰。

您可以在以 EC2 序列主控台瀏覽器為基礎的用戶端或 SSH 用戶端中使用 SysRq 命令。每個用戶端傳送中斷請求的命令是不同的。

若要使用 SysRq，請根據您所使用的用戶端選擇下列其中一個程序。

Browser-based client

在序列主控台瀏覽器型用戶端中使用 SysRq

1. 連線到執行個體的序列主控台。
2. 若要傳送中斷請求，請按 CTRL+0 (零)。如果您的鍵盤支援，您也可以使用「暫停」或「Break」鍵來傳送中斷請求。

```
[ec2-user ~]$ CTRL+0
```

3. 若要發出 SysRq 命令，請按鍵盤上對應至所需命令的按鍵。例如，若要顯示 SysRq 命令清單，請按 h。

```
[ec2-user ~]$ h
```

h 命令會輸出類似下列內容。

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

SSH client

在 SSH 用戶端中使用 SysRq

1. 連線到執行個體的序列主控台。

2. 若要傳送中斷請求，請按下 ~B (波狀符號，後面接著大寫 B)。

```
[ec2-user ~]$ ~B
```

3. 若要發出 SysRq 命令，請按鍵盤上對應至所需命令的按鍵。例如，若要顯示 SysRq 命令清單，請按 h。

```
[ec2-user ~]$ h
```

h 命令會輸出類似下列內容。

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

> **Note**
>
> 根據您使用的 SSH 用戶端，您用於傳送中斷請求的命令可能會有所不同。

(Windows 執行個體) 使用 SAC 為您的執行個體進行疑難排解

Windows 的特殊管理主控台 (SAC) 功能提供了一種疑難排解 Windows 執行個體的方法。透過連線到執行個體的序列主控台並使用 SAC，您可以中斷開機程序並以安全模式啟動 Windows。

> **Note**
>
> 如果您在執行個體上啟用 SAC，則依賴密碼擷取的 EC2 服務將無法從 Amazon EC2 主控台運作。Amazon EC2 上的 Windows 啟動代理程式 (EC2Config、EC2Launch v1 和 EC2Launch v2) 倚賴序列主控台執行各種任務。在執行個體上啟用 SAC 時，這些任務不會成功執行。如需 Amazon EC2 啟動代理程式的詳細資訊，請參閱 the section called “設定 Windows 執行個體”。如果您啟用 SAC，可以稍後停用它。如需詳細資訊，請參閱停用 SAC 和開機選單。

任務

- 使用 SAC

- 使用開機功能表
- 停用 SAC 和開機選單

使用 SAC

使用 SAC

1. 連接至序列主控台。

   如果執行個體上已啟用 SAC，序列主控台會顯示 SAC> 提示。

```
Computer is booting, SAC started and initialized.

Use the "ch -?" command for information about using channels.
Use the "?" command for general help.


SAC>?
EVENT: The CMD command is now available.
SAC
```

2. 若要顯示 SAC 命令，請輸入「?」，然後按 Enter。

   預期的輸出結果

```
SAC>?
ch                  Channel management commands.  Use ch -? for more help.
cmd                 Create a Command Prompt channel.
d                   Dump the current kernel log.
f                   Toggle detailed or abbreviated tlist info.
? or help           Display this list.
i                   List all IP network numbers and their IP addresses.
i <#> <ip> <subnet> <gateway> Set IPv4 addr., subnet and gateway.
id                  Display the computer identification information.
k <pid>             Kill the given process.
l <pid>             Lower the priority of a process to the lowest possible.
lock                Lock access to Command Prompt channels.
m <pid> <MB-allow>  Limit the memory usage of a process to <MB-allow>.
p                   Toggle paging the display.
r <pid>             Raise the priority of a process by one.
s                   Display the current time and date (24 hour clock used).
s mm/dd/yyyy hh:mm  Set the current time and date (24 hour clock used).
t                   Tlist.
restart             Restart the system immediately.
shutdown            Shutdown the system immediately.
crashdump           Crash the system. You must have crash dump enabled.
```

3. 若要建立命令提示通道 (例如 cmd0001 或 cmd0002)，請輸入「cmd」，然後按 Enter。
4. 若要檢視命令提示通道，請按 ESC，然後按 TAB。

   預期的輸出結果

```
Name:                 Cmd0001
Description:          Command
Type:                 VT-UTF8
Channel GUID:         ef9f20a0-1287-11eb-82b0-0e4ba51872e5
Application Type GUID: 63d02271-8aa4-11d5-bccf-00b0d014a2d0

Press <esc><tab> for next channel.
Press <esc><tab>0 to return to the SAC channel.
Use any other key to view this channel.
```

5. 若要切換通道，請同時按 ESC+TAB+通道號碼。例如，若要切換至 cmd0002 頻道 (如果已建立)，請按 ESC+TAB+2。
6. 輸入命令提示通道所需的憑證。

```
Please enter login credentials.
Username: Administrator
Domain  : .
Password: *******
```

命令提示字元與您在桌面上取得的完整功能命令 Shell 相同，但是除了它，不允許讀取已經輸出的字元。

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 10.0.17763.1

Copyright (C) Microsoft Corporation.
On computer: EC2AMAZ-ASR4SAI

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online           46 GB    46 GB

DISKPART>
```

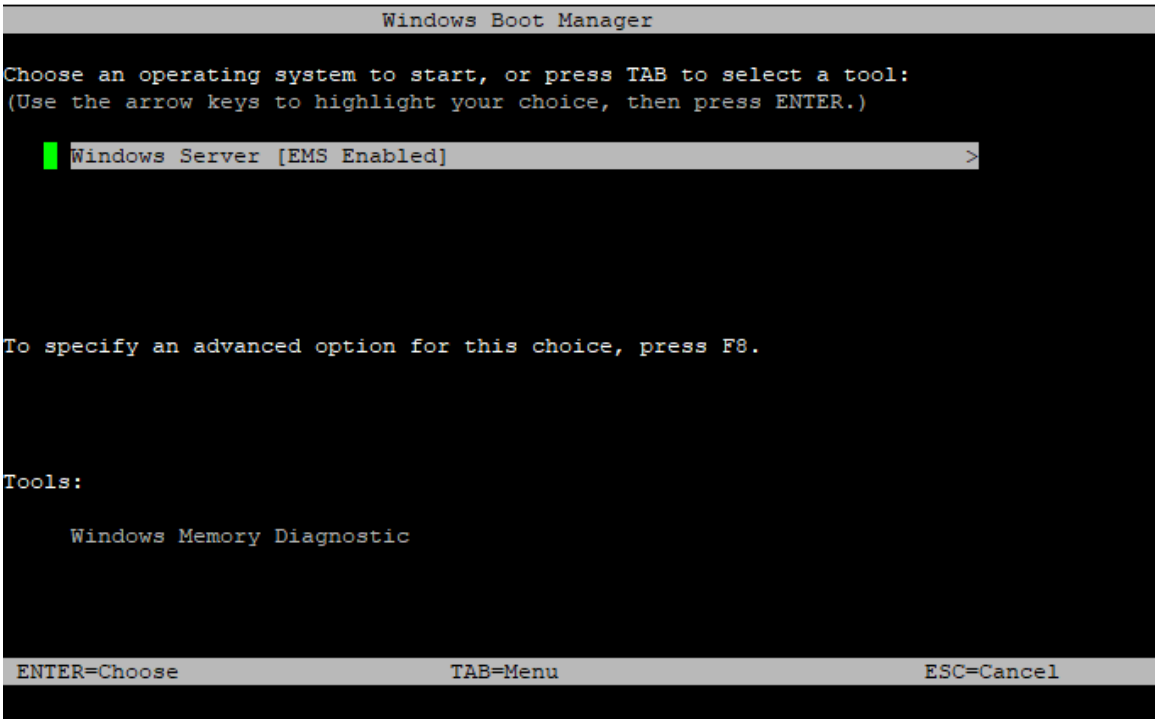
也可以從命令提示字元使用 PowerShell。

請注意，您可能需要將進度偏好設定設定為無訊息模式。

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32>
```

使用開機功能表

如果執行個體啟用了開機功能表，且透過 SSH 連線後重新啟動，您應該會看到開機功能表，如下所示。



開機功能表命令

ENTER

啟動選取的作業系統項目。

TAB

切換至「工具」功能表。

ESC

取消並重新啟動執行個體。

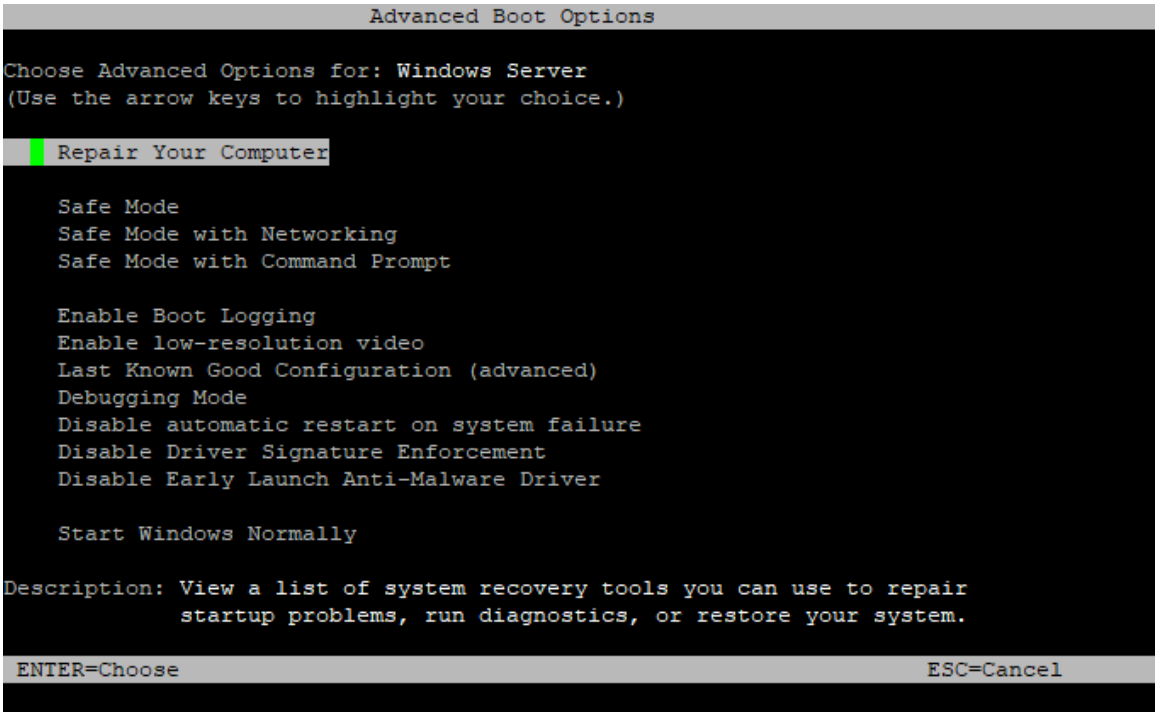
ESC 後跟 8

相當於按下 F8 。顯示所選項目的進階選項。

ESC 鍵 + 向左箭頭

回到初始開始功能表。

單獨的 ESC 鍵不會帶您回到主功能表，因為 Windows 正在等待查看是否正在進行逸出序列。



停用 SAC 和開機選單

如果您啟用 SAC 和開機選單，可以稍後停用這些功能。

使用下列其中一種方法來停用執行個體上的 SAC 和開機功能表。

PowerShell

在 Windows 執行個體上停用 SAC 和開機功能表

1. 連線到您的執行個體，並從提高權限的 PowerShell 命令列執行下列步驟。
2. 首先透過將值變更為 no 來停用開機選單。

```
bcdedit /set '{bootmgr}' displaybootmenu no
```

3. 然後透過將值變更為 off 來停用 SAC。

```
bcdedit /ems '{current}' off
```

4. 透過重新啟動執行個體來套用更新的組態。

```
shutdown -r -t 0
```

Command prompt

在 Windows 執行個體上停用 SAC 和開機功能表

1. 連接到您的執行個體，並從命令提示字元執行下列步驟。
2. 首先透過將值變更為 no 來停用開機選單。

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. 然後透過將值變更為 off 來停用 SAC。

```
bcdedit /ems {current} off
```

4. 透過重新啟動執行個體來套用更新的組態。

```
shutdown -r -t 0
```

傳送診斷中斷以偵錯無法連線的 Amazon EC2 執行個體

> **Warning**
>
> 診斷中斷專供進階使用者使用。不正確的使用可能會對執行個體造成負面影響。將診斷中斷傳送至執行個體可能會觸發執行個體當機並重新開機，這會導致資料遺失。

您可以將診斷中斷傳送至無法連線或沒有回應的執行個體，以手動觸發 Linux 執行個體的核心錯誤，或 Windows 執行個體的停止錯誤 (通常稱為藍色畫面錯誤)。

Linux 執行個體

發生核心錯誤時，Linux 作業系統通常會當機並重新開機。作業系統的具體行為取決於其組態。核心錯誤也可用來促使執行個體的作業系統核心執行任務，例如產生損毀傾印檔案。然後，您可以利用損毀傾印檔案中的資訊來分析根本原因，對執行個體進行除錯。作業系統會代表執行個體本身在本機產生損毀傾印資料。

Windows 執行個體

一般而言，Windows 作業系統在停止錯誤發生時會當機並重新開機，但具體行為取決於其組態。停止錯誤也會導致作業系統將除錯資訊 (例如核心記憶體傾印) 寫入檔案。然後，您可以利用此資訊來分析根本原因，以進行執行個體的除錯。作業系統會代表執行個體本身在本機產生記憶體傾印資料。

將診斷中斷傳送至執行個體之前，建議您先參閱作業系統的說明文件，然後進行必要的組態變更。

內容

- 支援的執行個體類型
- 先決條件
- 傳送診斷中斷

支援的執行個體類型

所有 Nitro 型執行個體類型都支援診斷中斷，但採用 AWS Graviton 處理器的執行個體類型除外。如需詳細資訊，請參閱建置在 AWS Nitro System 和 Graviton 上的執行個體。 AWS

先決條件

使用診斷中斷之前，您必須設定執行個體的作業系統。這可確保在核心錯誤 (Linux 執行個體) 或停止錯誤 (Windows 執行個體) 時，其執行您需要的動作。

Linux 執行個體

設定 Amazon Linux 2 或 Amazon Linux 2023 以在核心錯誤發生時產生損毀傾印

1. 連線到您的執行個體。
2. 安裝 kexec 和 kdump。

   ```
   [ec2-user ~]$ sudo yum install kexec-tools -y
   ```

3. 設定核心保留適當的記憶體數量給次要核心。該保留的記憶體數量取決於執行個體可用的記憶體總數。使用您偏好的文字編輯器開啟 `/etc/default/grub` 檔案，找出開頭為 `GRUB_CMDLINE_LINUX_DEFAULT` 的那一行，然後依下列格式新增 `crashkernel` 參數：`crashkernel=`*`memory_to_reserve`*。例如，若要保留 256MB，請如下修改 `grub` 檔案：

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. 儲存變更並關閉 grub 檔案。
5. 重建 GRUB2 組態檔案。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. 在以 Intel 和 AMD 處理器為架構的執行個體上，send-diagnostic-interrupt 命令會將 unknown non-maskable interrupt (NMI) 傳送至執行個體。您必須將核心設定為收到不明 NMI 時損毀。使用您偏好的文字編輯器開啟 /etc/sysctl.conf 檔案，然後新增下列命令。

```
kernel.unknown_nmi_panic=1
```

7. 重新開機並重新連接您的執行個體。
8. 確認核心已使用正確的 crashkernel 參數開機。

```
$ grep crashkernel /proc/cmdline
```

下列輸出範例指出組態成功。

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0
```

9. 確認 kdump 服務在執行中。

```
[ec2-user ~]$ systemctl status kdump.service
```

下列輸出範例顯示 kdump 服務在執行中的結果。

```
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
 enabled)
   Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
```

```
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
 Main PID: 2503 (code=exited, status=0/SUCCESS)
```

> **Note**
>
> 根據預設，損毀傾印檔案會儲存至 `/var/crash/`。若要變更位置，請使用您偏好的文字編輯器來修改 `/etc/kdump.conf` 檔案。

設定 SUSE Linux Enterprise、Ubuntu 或 Red Hat Enterprise Linux

在以 Intel 和 AMD 處理器為架構的執行個體上，`send-diagnostic-interrupt` 命令會將 unknown non-maskable interrupt (NMI) 傳送至執行個體。您必須透過調整作業系統的組態檔案，將核心設定為在收到未知 NMI 時損毀。如需如何設定以損毀核心的詳細資訊，請參閱作業系統的文件：

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

Windows 執行個體

設定 Windows 在停止錯誤發生時產生記憶體傾印

1. 連線到您的執行個體。
2. 開啟控制台，選擇系統、進階系統設定。
3. 在系統內容對話方塊中，選擇進階索引標籤。
4. 在啟動及修復區段中，選擇設定...。
5. 在系統失敗區段中，視需要進行設定，然後選擇確定。

如需有關設定 Windows 停止錯誤的詳細資訊，請參閱 Windows 記憶體傾印檔案選項的概觀。

傳送診斷中斷

完成必要的組態變更後，您可以使用 AWS CLI 或 Amazon EC2 API 將診斷中斷傳送至執行個體。

AWS CLI

將診斷中斷傳送至執行個體

使用 send-diagnostic-interrupt 命令。

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

PowerShell

將診斷中斷傳送至執行個體

使用 Send-EC2DiagnosticInterrupt cmdlet。

```
Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```

Amazon EC2 使用者指南的文件歷史記錄

下表說明從 2019 年開始對「Amazon EC2 使用者指南」進行的重要補充。本指南也會經常更新，以處理您傳送給我們的意見回饋。

變更	描述	日期
EC2 Fast Launch 的許可檢查	新增指引，說明當您使用啟動範本啟用 EC2 Fast Launch 時，Amazon EC2 如何驗證您的許可，包括使用`$Latest`或`$Default`啟動範本版本的安全考量。	2026 年 8 月 14 日
區域和區域的地圖檢視	AWS 全域檢視中的區域和區域頁面現在包含地圖檢視，可在互動式世界地圖上顯示您的所有區域和區域。您可以在清單檢視和地圖檢視之間切換。您的檢視偏好設定會保留在各個工作階段中。	2026 年 8 月 12 日
將目前的根磁碟區取代為現有的 EBS 磁碟區	您可以將 Amazon EC2 執行個體目前的根磁碟區取代為您已設定的現有 Amazon EBS 磁碟區。	2026 年 7 月 9 日
「即時」EC2 機群的其他啟動範本覆寫	類型的 EC2 機群`instant`現在支援下列額外的啟動範本覆寫參數：`IamInstanceProfile` 、 `KeyName`和`MetadataOptions` 。	2026 年 7 月 9 日
尋找公有 AMI 的 SSM 參數	您可以找到與公有 AMI 相關聯的 AWS Systems Manager 參數，並使用它來自動參考該 AMI 的最新版本。	2026 年 7 月 7 日

專用主機上的 SEV-SNP	專用主機現在支援 AMD SEV-SNP，可讓您在專用實體伺服器上執行機密運算工作負載。	2026 年 7 月 1 日
AWS 全域檢視	EC2 Global View 現在已重新命名為 AWS Global View。您可以使用它，在單一檢視中檢視所有區域的 EC2 資源和其他支援 AWS 的資源。	2026 年 6 月 30 日
精確時間置放群組	您可以使用精確時間置放群組，將執行個體放置在可直接存取 AWS 基礎設施中高精確時間來源的硬體上。	2026 年 6 月 29 日
搭配允許的 AMIs 使用浮水印	您現在可以將 AMI 浮水印與允許 AMIs搭配使用，將執行個體啟動限制為僅具有指定浮水印篩選條件 AMIs。新的允許 AMIs 參數`ImageWatermarks` 具有下列篩選條件：`MaximumDaysSinceSourceImageCreated` 、`SourceImageRegion` 、 `MaximumDaysSinceWatermarkCreated` 和 `WatermarkKey` 。	2026 年 6 月 22 日
AWS 適用於 Windows 的 VMClock 裝置	AWS 提供適用於 Windows 的 null （無函數） 驅動程式套件，可讓作業系統辨識裝置。	2026 年 6 月 17 日

AMI 浮水印	您可以使用 AMI 浮水印來追蹤私有 AMIs 的來源並強制執行控管政策。當您從執行個體建立 AMIs、複製 AMIs 或與其他帳戶共用 AMIs 時，浮水印會持續存在。	2026 年 6 月 10 日
取消容量保留	您可以取消未來日期的容量保留。視取消時間而定，可能會產生費用。	2026 年 6 月 2 日
ENA Express 跨可用區支援	ENA Express 現在支援相同區域內不同可用區域中 Amazon EC2 執行個體之間的流量。	2026 年 5 月 11 日
受管資源可見性	您可以控制 AWS 服務代表您佈建的資源是否出現在 Amazon EC2 主控台檢視和 API 清單操作中。	2026 年 4 月 22 日
開始使用 EFA 和 NIXL	您可以將 Elastic Fabric Adapter (EFA) 與 NIXL 搭配使用，以用於分解的推論工作負載。	2026 年 3 月 17 日
教學課程：使用可中斷的容量保留啟動 EC2 機群	新的教學課程說明如何建立使用可中斷容量預留的 EC2 機群，讓未使用的預留容量可供您的可中斷工作負載使用。	2026 年 3 月 16 日
本機區域中的容量區塊	ML 的容量區塊現在可在 Local Zones 中使用。	2026 年 2 月 25 日
IMDSv2 強制執行	您可以在每個帳戶層級強制使用 IMDSv2 AWS 區域。強制執行時，執行個體只能在設定為需要 IMDSv2 時啟動。	2026 年 2 月 24 日

巢狀虛擬化	您可以在虛擬化 Amazon EC2 執行個體內執行 Hyper-V 和 KVM 等 Hypervisor。	2026 年 2 月 16 日
次要網路	次要網路是專用於特殊聯網使用案例的虛擬網路。您可以在 VPC 和次要網路中啟動選取的 EC2 執行個體。	2026 年 2 月 9 日
容量區塊共用	您可以與 AWS 組織中的其他共用 EC2 AWS 帳戶 容量區塊。	2026 年 2 月 2 日
可中斷的容量保留	可中斷的容量保留可協助您暫時為帳戶中的其他工作負載提供未使用的容量。	2025 年 11 月 21 日
P6-B300 執行個體的執行個體拓撲	您可取得 `p6-b300.48xlarge` 執行個體的執行個體拓撲資訊。	2025 年 11 月 18 日
AMI 原始伺服器	您現在可以透過取得其所有前代 AMI 的清單來追蹤 AMIs原始伺服器。	2025 年 11 月 12 日
容量保留拓撲	啟動執行個體前，您可以使用 DescribeCapacityReservationTopology API，描述預留容量在網路階層中相對於彼此的位置。	2025 年 10 月 28 日
最佳化 CPU 以節省許可成本	您現在可配置作用中 CPU，為包含授權的 Windows 及 SQL Server 執行個體節省許可成本。	2025 年 10 月 15 日

新的 EC2 容量管理器	Amazon EC2 新增了 EC2 容量管理器，讓您可透過單一位置監控、分析及管理跨隨需執行個體、Spot 執行個體與容量保留的容量用量。	2025 年 10 月 15 日
AWSEC2CapacityManagerServiceRolePolicy – 新的受管政策	Amazon EC2 新增了 AWSEC2CapacityManagerServiceRolePolicy 受管政策，可讓您代您管理容量資源並與 AWS Organizations 整合。	2025 年 10 月 15 日
GPUPowerUtilization 指標	加速運算執行個體的新 GPUPowerUtilization 指標。	2025 年 10 月 13 日
EC2 Instance Connect Endpoint IPv6 支援	EC2 Instance Connect Endpoint 目前支援 IPv6 連線。您現在可將 EC2 Instance Connect Endpoints 設定為支援雙堆疊或僅 IPv6，以連接至具有 IPv6 位址的執行個體。	2025 年 10 月 2 日
允許的 AMI - 新增參數	允許 AMIs，這項功能可讓您限制探索和使用 內的 AMIs AWS 帳戶，現在支援四個新參數： AWS Marketplace 產品代碼、AMI 名稱、自 AMI 棄用以來的最長期間，以及 AMIs的最長存留期。	2025 年 9 月 25 日
執行個體儲存體磁碟區的詳細效能統計資料	Amazon EC2 為連接至 Nitro 型執行個體的 NVMe 執行個體儲存體磁碟區提供即時、高解析度的效能統計資料。	2025 年 9 月 16 日

Amazon EC2 Mac 專用主機更新	Amazon EC2 Mac 專用主機支援專用主機自動復原與重新啟動型主機維護。	2025 年 8 月 28 日
Local Zones 的 AMI 複本	您現在可將 Amazon EBS 支援的 AMI 複製到 Local Zones 或從 Local Zones 複製出來。	2025 年 8 月 28 日
AmazonEC2ImageReferencesAccessPolicy – 新的受管政策	Amazon EC2 新增了 AmazonEC2ImageReferencesAccessPolicy 受管政策，以便提供掃描 EC2 DescribeImageReferences API 支援的全部資源類型的許可。	2025 年 8 月 26 日
AMI 用量報告	如果您與其他 共用 Amazon Machine Image AMIs) AWS 帳戶，無論是與特定 AWS 帳戶、組織、組織單位 (OUs) 或公開 ，您都可以透過建立 AMIs用量報告來查看這些 AMI 的使用方式。報告提供在 EC2 執行個體或啟動範本中使用 AWS 帳戶 AMIs的可見性，以及參照每個 AMI 的執行個體或啟動範本數量。	2025 年 8 月 26 日
AMI 參考	您可以識別參考指定 Amazon Machine Image (AMIs) AWS 的資源，無論 AMIs 是公有還是私有，還是擁有者。此可見度有助您確認資源使用最新的合規 AMI。	2025 年 8 月 26 日
修改 EC2 Instance Connect Endpoint	您現在可以使用 AWS CLI 或 SDK 修改現有的 EC2 Instance Connect Endpoint。	2025 年 8 月 14 日

容量保留共用增強功能	您現在可以在叢集置放群組中共用容量保留。	2025 年 8 月 12 日
已為 EC2 Instance Connect Endpoint 更新受管政策	為支援修改現有的 EC2 Instance Connect 端點，Amazon EC2 更新了 `Ec2InstanceConnectEndpoint` 政策，以新增在 EC2 Instance Connect 端點建立的網路介面上指派及取消指派 IPv6 位址，以及修改安全群組的許可。Amazon EC2 亦更新了此政策，以便能使用 `StringLike` 條件運算子來取代 `Null` 條件運算子。	2025 年 7 月 31 日
執行個體拓撲 - 其他區域支援	DescribeInstanceTopology API 現可在亞太區域 (墨爾本) 使用。	2025 年 7 月 31 日
強制終止	您現在可以強制終止執行個體。	2025 年 7 月 31 日
執行個體拓撲 - 其他區域支援	DescribeInstanceTopology API 現在可在 中使用 AWS GovCloud (US-West)。	2025 年 7 月 24 日
終止執行個體時跳過 OS 關機	終止執行個體時，您可跳過作業系統 (OS) 正常關機。	2025 年 7 月 23 日
停止執行個體時跳過 OS 關機	停止執行個體時，您可跳過作業系統 (OS) 正常關機。	2025 年 7 月 23 日
us-east-1 中的 P6-B200 容量區塊	P6-B200 容量區塊現可在美國東部 (維吉尼亞北部) - `us-east-1` 使用。	2025 年 7 月 23 日

更新了 EC2 Fast Launch 服務連結角色的受管政策	為協助防止資源孤立，Amazon EC2 更新了此政策，以新增描述磁碟區、磁碟區屬性及網路介面，以及刪除 EC2 Fast Launch 所建立磁碟區及網路介面的許可。	2025 年 7 月 17 日
EC2 序列主控台 - 其他區域支援	EC2 序列主控台現也可在亞太區域 (泰國) 和墨西哥 (中部) 使用。	2025 年 7 月 15 日
EC2 Instance Connect - 其他區域支援	EC2 Instance Connect 現在也可在亞太區域 (泰國) 和墨西哥 (中部) 使用。	2025 年 7 月 15 日
硬體封包時間戳記	您可在 EC2 執行個體上啟用硬體封包時間戳記功能，為每個傳入的網路封包新增 64 位元奈秒精確度的時間戳記。	2025 年 6 月 23 日
AMI 取消註冊期間刪除快照	取消註冊 AMI 時，您可同時自動刪除其相關聯的快照。	2025 年 5 月 28 日
使用者啟動的重新啟動移轉	當您的執行個體收到 EC2 排程重新啟動事件通知時，預設會在事件之前重新啟動執行個體時啟用執行個體移轉。	2025 年 5 月 20 日
Amazon EC2 Mac 執行個體的 SIP 組態	您可為 x86 Mac 執行個體和 Apple 晶片 Mac 執行個體設定系統完整性保護 (SIP) 設定。	2025 年 5 月 20 日
根磁碟區所有權委派	更新 Apple 晶片 Mac 執行個體的作業系統時，您現在可自動化 Amazon EBS 根磁碟區的所有權委派。	2025 年 5 月 20 日

P6-B200 執行個體的執行個體拓撲	您可取得 p6-b200.48xlarge 執行個體的執行個體拓撲資訊。	2025 年 5 月 15 日
更新了 EC2 Fast Launch 服務連結角色的受管政策	更新了 EC2 Fast Launch 服務連結角色政策，新增建立與維護 Amazon EventBridge 事件規則的存取權，以及其他增強 EC2 Fast Launch 使用體驗的動作。如需有關服務連結角色授予之許可的詳細資訊，請參閱 EC2FastLaunchServiceRolePolicy 授予的許可。	2025 年 5 月 14 日
更新了 EC2 Fast Launch 完整存取政策的受管政策	更新了 EC2 Fast Launch 完整存取政策，新增建立及修改啟動範本作業、建立網路及安全群組資源、授權傳入規則，以及刪除 EC2 Fast Launch 所建立的資源。此政策還可描述及建立 CloudFormation 堆疊，以及刪除 EC2 Fast Launch 建立的堆疊。	2025 年 5 月 14 日
簡化 EC2 Fast Launch 的入門流程	更新了 EC2 Fast Launch 以簡化入門流程。	2025 年 5 月 14 日
AWS Config 與允許的 AMIs整合	使用 ec2-instance-launched-with-allowed-ami AWS Config 規則來檢查執行中或已停止的執行個體是否使用AMIs 條件AMIs啟動。	2025 年 3 月 11 日

已更新 AWSEC2CapacityReservationFleetRolePolicy 受管政策	Amazon EC2 更新了 `AWSEC2CapacityReservationFleetRolePolicy` 受管政策，可使用 `ArnLike` 條件運算子而非 `StringLike` 條件運算子。	2025 年 3 月 3 日
基於時間的 AMI 複本	您現在可為 EBS 支援型 AMI 複製操作請求完成時長，確保 AMI 複製在特定時間範圍內完成。	2025 年 2 月 25 日
更新的 AmazonEC2ReadOnlyAccess 政策	Amazon EC2 已將 `GetSecurityGroupsForVpc` 操作新增至現有 AmazonEC2ReadOnlyAccess 政策。	2024 年 12 月 27 日
受管執行個體	您現可在 EC2 主控台中檢視 Amazon EC2 受管執行個體。	2024 年 12 月 1 日
宣告式政策	您現可使用宣告式政策同時在多個區域和帳戶中套用帳戶層級設定。宣告式政策支援設定 EC2 序列主控台存取、允許 AMI、IMDS 預設值，以及封鎖 VPC、AMI 和快照的公有存取設定。如需詳細資訊，請參閱「AWS Organizations 使用者指南」中的宣告式政策，以及每項支援功能的特定文件。	2024 年 12 月 1 日
允許 AMI	您現可透過指定 AMI 必須符合的條件，在 Amazon EC2 控制 AMI 的探索及使用。	2024 年 12 月 1 日
即時容量區塊	您現可為 ML 保留容量區塊，最快 30 分鐘即可啟動。	2024 年 11 月 21 日

未來日期的容量保留	您現可請求未來日期的容量保留。	2024 年 11 月 21 日
延長容量區塊	您現可延長現有容量區塊的期限。	2024 年 11 月 21 日
6 個月容量區塊	您現可為 ML 保留最多 6 個月 (182 天) 的容量區塊。	2024 年 11 月 21 日
已儲存的篩選條件集	您現可建立自訂的篩選器群組並重複使用，以高效率的方式檢視 EC2 資源。	2024 年 11 月 20 日
效能保護	當針對 EC2 機群或 Spot 機群使用基於屬性的執行個體類型選擇時，您現可啟用效能保護，以確保所選執行個體類型類似於或超過指定效能基準。	2024 年 11 月 20 日
僅限容量保留	您現可指定執行個體只在容量保留或容量保留資源群組中執行。	2024 年 11 月 20 日
識別來源 AMI	您現可識別用於建立 AMI 的來源 AMI。	2024 年 11 月 13 日
分割容量	您可從現有容量保留中分割容量來建立新保留。	2024 年 10 月 30 日
移動容量	您現可將容量從一個容量保留移至另一容量保留。	2024 年 10 月 30 日
新手教學課程	適合新手的兩個新教學課程：啟動我的第一個 EC2 執行個體，以及啟動測試 EC2 執行個體並連線。	2024 年 10 月 21 日
Windows Server 2025 支援	新增對 Windows Server 2025 的支援。	2024 年 10 月 16 日

EC2 全域檢視	EC2 全域檢視現在可讓您檢視帳戶中所有區域的容量保留和容量區塊。	2024 年 10 月 16 日
即時遷移主機維護	Amazon EC2 專用主機現在支援即時遷移主機維護，可自動將支援的執行個體從受損專用主機遷移到替代專用主機，而無需停止或重新啟動。	2024 年 10 月 15 日
共用容量保留的帳單指派	您現在可以將共用容量保留之任何可用容量的帳單指派給屬於相同 AWS 組織的消費者帳戶。	2024 年 10 月 14 日
PTP 硬體時鐘 - 其他區域支援	PTP 硬體時鐘現在也可在美國東部 (俄亥俄) 和亞太地區 (馬來西亞) 使用。	2024 年 9 月 23 日
EC2 Instance Connect 支援 IPv6	您現可使用 EC2 Instance Connect 來連線執行個體的公用 IPv6 地址。	2024 年 9 月 23 日
EC2 Instance Connect 字首清單	您現可在安全性群組中建立規則時，為 IPv4 或 IPv6 地址選取受管字首清單，以允許來自 EC2 Instance Connect 服務的 SSH 流量。	2024 年 9 月 23 日
隨需容量保留管理新功能	您現可分割容量保留、在容量保留之間移動容量，以及修改容量保留的執行個體資格屬性。	2024 年 8 月 14 日

C6g、C6gn、C6gd、C7g、C7gd、M6g、M6gd、M7g、M7gd、R6g 和 R6gd 的休眠支援	讓在 C6g、C6gn、C6gd、C7g、C7gd、M6g、M6gd、M7g、M7gd、R6g 和 R6gd 執行個體類型上執行的新啟動執行個體進入休眠。	2024 年 7 月 30 日
支援 Graviton 執行個體類型的 AMI 的休眠支援	在新啟動執行個體從支援 Graviton 執行個體類型的 Amazon Linux 或 Ubuntu AMI 啟動後，讓其進入休眠。	2024 年 7 月 30 日
Credential Guard 支援的其他執行個體類型	您現可為 C7i、C7-flex、M7i、M7i-flex、R7i、R7i-flex 和 T3 執行個體啟用 Credential Guard。	2024 年 6 月 26 日
EC2 執行個體類型搜尋工具 - 其他參數	EC2 執行個體類型搜尋工具現在提供其他參數，可讓您為工作負載指定更詳細的需求。	2024 年 6 月 5 日
EC2 快速啟動新受管政策	新增 EC2FastLaunchFullAccess 政策，從執行個體執行與 EC2 快速啟動功能相關的 API 動作。	2024 年 5 月 14 日
AMI 取消註冊保護	您可以在 AMI 上開啟註銷保護以防止意外或惡意刪除。	2024 年 4 月 23 日
PTP 硬體時鐘 - 執行個體類型支援	PTP 硬體時鐘現在可在 M7a、M7g、M7i、R7a 和 R7i 執行個體類型上使用。	2024 年 4 月 22 日
新增 Nitro 效能考量事項以增強網路連線	此頁面著重於網路考量事項，以協助針對 Nitro 型 Amazon EC2 執行個體進行效能調整。	2024 年 4 月 4 日

基於 VSS 的 EBS 快照新受管政策	Amazon EC2 VSS 擁有新 IAM 受管政策，您可以將其新增到執行個體設定檔角色中，以確保您的許可保持最新並遵循最佳實務。	2024 年 3 月 28 日
PTP 硬體時鐘 - 美國東部 (維吉尼亞北部)	PTP 硬體時鐘現於美國東部 (維吉尼亞北部) 區域可用。	2024 年 3 月 26 日
設定 IMDSv2 作為帳戶預設值	您可在您的帳戶中將所有新 EC2 執行個體啟動設定為預設使用執行個體中繼資料服務版本 2 (IMDSv2)。	2024 年 3 月 25 日
標記從快照建立的新 Linux AMI	當您從快照建立 Linux AMI 時，您可標記新 AMI。	2024 年 3 月 7 日
複製時標記新 AMI 和快照	當您複製 AMI 時，您可以為新 AMI 和新快照新增相同的標籤，或者可以使用不同的標籤來進行標記。	2024 年 3 月 7 日
移除 AWS Management Pack 頁面	AWS Management Pack 主要與 Windows Server 2012 和舊版搭配使用。不再支援這些舊版作業系統平台版本。若要管理和疑難排解在 AWS 和內部部署上執行的伺服器機群，請參閱AWS Systems Manager 機群管理員。	2024 年 2 月 12 日
MacOS AMI 已預先安裝 EC2 Instance Connect	EC2 Instance Connect 現已預先安裝在 macOS Sonoma 14.2.1 或更新版本、macOS Ventura 13.6.3 或更新版本，以及 macOS Monterey 12.7.2 或更新版本的 AMI。	2024 年 1 月 26 日

適用於 CentOS、macOS 和 RHEL 的 EC2 Instance Connect 支援	您現在可以在受支援的 CentOS、macOS 和 RHEL AMI 上安裝 EC2 Instance Connect。	2023 年 12 月 6 日
C7a、C7i、R7a、R7i 和 R7iz 的休眠支援	對您新啟動且在 C7a、C7i、R7a、R7i 和 R7iz 執行個體類型上執行的執行個體進行休眠。	2023 年 12 月 1 日
Amazon Q EC2 執行個體類型選取器	Amazon Q EC2 執行個體類型選取器會考量您的使用案例、工作負載類型和 CPU 製造商偏好設定，以及排定價格和效能優先順序的方式。然後，它會使用這些資料，針對您的新工作負載，提供指導和建議最適合的 Amazon EC2 執行個體類型。	2023 年 11 月 28 日
EC2 免費方案	您可以從 EC2 儀表板追蹤 EC2 免費方案用量。	2023 年 11 月 26 日
Console-to-Code	Console-to-Code 可協助您開始使用自動化程式碼。Console-to-Code 會記錄您的主控台動作，然後使用生成式 AI，以您偏好的基礎架構即程式碼格式建議程式碼。您可以使用該程式碼作為起點，對其進行自訂以使其針對特定使用案例準備好投入生產。	2023 年 11 月 26 日

可設定的閒置連線追蹤逾時	處於閒置狀態的安全群組連線可能會導致連線追蹤耗盡，並造成不追蹤連線及捨棄封包。您現在可以在彈性網路介面上設定安全群組連線追蹤的逾時 (以秒為單位)。	2023 年 11 月 17 日
PTP 硬體時鐘	支援的執行個體現在具有精確時間協定 (PTP) 硬體時鐘。PTP 硬體時鐘支援 NTP 或直接 PTP 連線。	2023 年 11 月 16 日
為已啟用休眠的執行個體變更執行個體類型	您現在可以為已啟用休眠的執行個體變更執行個體類型 (處於 `stopped` 狀態的情況下)。	2023 年 11 月 16 日
執行個體拓撲	您可以使用 DescribeInstanceTopology API 偵測執行個體的位置，然後使用此資訊來透過在實際位置彼此較近的執行個體上執行 HPC 和 ML 任務以對其進行最佳化。	2023 年 11 月 13 日
EC2 快速啟動共用 AMI 支援	您現可在與您共用的 AMI 啟用 EC2 快速啟動功能。當您為共用 AMI 啟用 EC2 快速啟動功能時，系統會在您的帳戶中建立預先佈建快照。	2023 年 11 月 6 日
ML 的容量區塊	現在您可以為日後預留 GPU 執行個體，以支援短時間的機器學習 (ML) 工作負載。	2023 年 10 月 31 日
Spot 執行個體休眠	現在您可以讓 Spot 執行個體進入休眠，方法與隨需執行個體目前可使用的休眠體驗和執行個體系列相同。	2023 年 10 月 24 日

「封鎖 AMI 公有存取權」的預設設定	預設情況下，所有新帳戶和沒有公開 AMI 的現有帳戶都會啟用「適用於 AMI 的封鎖公開存取」。	2023 年 10 月 20 日
Amazon EC2 全域檢視	Amazon EC2 Global View 支援其他資源類型和可自訂的顯示選項。	2023 年 10 月 18 日
對 Ubuntu 22.04.2 LTS 的休眠支援 (Jammy Jellyfish)	讓從 Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI 啟動的新啟動執行個體休眠。	2023 年 10 月 16 日
停用 AMI	可以停用 AMI 以防止其用於執行個體啟動。	2023 年 10 月 12 日
附接的 EBS 狀態檢查	您可以使用附接的 EBS 狀態檢查來監控附接至執行個體的 Amazon EBS 磁碟區是否可連線。	2023 年 10 月 11 日
對 Red Hat Enterprise Linux 9 的休眠支援	讓從 Red Hat Enterprise Linux 9 AMI 啟動的新啟動執行個體休眠。	2023 年 10 月 2 日
對 Microsoft Windows Server 2022 的休眠支援	讓從 Microsoft Windows Server 2022 AMI 啟動的新啟動執行個體休眠。	2023 年 10 月 2 日
AL2023 支援休眠	讓從 AL2023 AMI 啟動的新啟動執行個體休眠。	2023 年 10 月 2 日
啟動 Spot 機群中的 Spot 執行個體中斷	您可以在 Amazon EC2 主控台中選取 Spot 機群並啟動機群中的 Spot 執行個體中斷，以便您可以測試 Spot 執行個體上的應用程式如何處理中斷。	2023 年 9 月 21 日

封鎖 AMI 的公有存取權	您可以在帳戶層級啟用封鎖 AMI 的公有存取權，以阻止任何公開 AMI 的嘗試。	2023 年 9 月 12 日
M7i 和 M7i-flex 休眠支援	對您新啟動且在 M7i 和 M7i-flex 執行個體類型上執行的執行個體進行休眠。	2023 年 8 月 22 日
EC2-Classic 已遭棄用	搭配 EC2-Classic，EC2 執行個體可在與其他客戶共用的單一平面網路中執行。Amazon VPC 取代 EC2-Classic。使用 Amazon VPC，您的執行個體可在邏輯上與您的 AWS 帳戶隔離的 Virtual Private Cloud (VPC) 中執行。	2023 年 8 月 8 日
專用主機	您可以在 Outpost 上的特定硬體資產中分配專用執行個體。	2023 年 6 月 20 日
EC2 Instance Connect Endpoint	您現在可以透過 SSH 或 RDP 連線到執行個體，而不需要執行個體擁有公有 IPv4 地址。	2023 年 6 月 13 日
IMDS Package Analyzer	您現在可以使用 IMDS Package Analyzer，識別 EC2 執行個體上的 IMDSv1 呼叫來源。	2023 年 6 月 1 日
EC2 序列主控台裸機執行個體	EC2 序列主控台現在支援連線至所選裸機執行個體的序列埠。	2023 年 4 月 11 日
啟動範本配額	您現在在 Service Quotas 主控台，或是使用 Service Quotas CLI，就能檢視啟動範本和啟動範本版本的配額。	2023 年 4 月 3 日

容量保留使用率通知	AWS Health 現在會在您帳戶中容量保留的容量使用率低於 20% 時傳送通知。	2023 年 4 月 3 日
容量保留群組	您現在可以將與您共用的容量保留新增到您擁有的容量保留群組。	2023 年 3 月 30 日
修改執行個體中繼資料選項	您現在可使用 Amazon EC2 主控台修改執行個體中繼資料選項。	2023 年 3 月 20 日
就地 macOS 作業系統更新	現在，您可以在 M1 Mac 執行個體上執行就地 Apple macOS 作業系統更新。	2023 年 3 月 14 日
UEFI preferred	現在，您可以建立支援 Unified Extensible Firmware Interface (UEFI) 和 Legacy BIOS 開機模式的 AMI。	2023 年 3 月 3 日
修改適用於 IMDSv2 的 AMI	修改現有 AMI，以便從 AMI 啟動的執行個體預設需要 IMDSv2。	2023 年 2 月 28 日
基於 Windows 虛擬化的安全性 - Credential Guard	您可以在支援的 Amazon EC2 執行個體上啟用 Credential Guard，這是一個基於虛擬化的安全性 (VBS) 功能。	2023 年 1 月 31 日

EC2 Instance Connect Endpoint 的全新受管政策	新增了 `Ec2InstanceConnectEndpoint` 政策，(該政策已連接至 AWSServiceRoleForEC2InstanceConnect 服務連結角色)，以便您建立 EC2 Instance Connect Endpoint 時，允許 Amazon EC2 代表您執行操作。	2023 年 1 月 24 日
啟動範本中的 AMI 別名	您可以在啟動範本中指定 AWS Systems Manager 參數而非 AMI ID，以避免每次 AMI ID 變更時都必須更新範本。	2023 年 1 月 19 日
C6i、I3en 和 M6i 休眠支援	對您新啟動且在 C6i、I3en 和 M6i 執行個體類型上執行的執行個體進行休眠。	2022 年 12 月 19 日
預防未完成的寫入	利用區塊儲存功能，預防未完成的寫入，提升 I/O 密集型關聯式資料庫工作負載的效能，並減少延遲，而不會對資料彈性產生負面影響。	2022 年 11 月 29 日
ENA Express	使用 ENA Express 增加輸送量並將 EC2 執行個體之間的網路流量尾延遲降至最低。	2022 年 11 月 28 日
複製 AMI 標籤	複製 AMI 時，您可以同時複製使用者定義的 AMI 標籤。	2022 年 11 月 18 日
適用於存放和還原的 AMI 大小	可存放至 Amazon S3 儲存貯體中和從 Amazon S3 儲存貯體還原的 AMI (壓縮之前) 大小現可達到 5,000 GB。	2022 年 11 月 16 日

Spot 執行個體的 priceCapacityOptimized 分配策略	使用 `priceCapacityOptimized` 分配策略的 Spot 機群會考慮價格和容量，來選取最不可能中斷且價格最低的 Spot 執行個體集區。	2022 年 11 月 10 日
Spot 執行個體的 price-capacity-optimized 分配策略	使用 `price-capacity-optimized` 分配策略的 EC2 機群會考慮價格和容量，來選取最不可能中斷且價格最低的 Spot 執行個體集區。	2022 年 11 月 10 日
取消與您的帳戶共用 AMI	如果 AMI 已與 共用， AWS 帳戶 而且您不希望再與您的帳戶共用，您可以從 AMI 的啟動許可中移除您的帳戶。	2022 年 11 月 4 日
轉移彈性 IP 地址	您現在可以 AWS 帳戶 將彈性 IP 地址從一個轉移到另一個。	2022 年 10 月 31 日
取代根磁碟區	您可以使用 AMI 取代執行中執行個體的根 Amazon EBS 磁碟區。	2022 年 10 月 27 日
自動將執行個體連線至資料庫	使用自動連線功能，將一個或多個 EC2 執行個體快速連線至 RDS 資料庫，以允許它們之間的流量。	2022 年 10 月 10 日
AMI 配額	配額現在適用於建立及共用 AMI。	2022 年 10 月 10 日
設定適用於 IMDSv2 的 AMI	設定 AMI，以便從 AMI 啟動的執行個體預設需要 IMDSv2。	2022 年 10 月 3 日

啟動 Spot 執行個體中斷	您可以在 Amazon EC2 主控台中選取 Spot 執行個體並啟動中斷，以便您可以測試 Spot 執行個體上的應用程式如何處理中斷。	2022 年 9 月 26 日
已驗證的 AMI 供應商	在 Amazon EC2 主控台中，Amazon 或已驗證的 Amazon 合作夥伴擁有的公用 AMI 標記為 Verified provider (已驗證的供應商)。	2022 年 7 月 22 日
上的置放群組 AWS Outposts	新增在 Outpost 上放置群組的主機分散策略。	2022 年 6 月 30 日
上的專用主機 AWS Outposts	您可以配置 AWS Outposts上的專用執行個體。	2022 年 5 月 31 日
執行個體停止保護	若要避免意外停止執行個體，您可啟用執行個體的停止保護。	2022 年 5 月 24 日
UEFI 安全開機	UEFI 安全開機依托 Amazon EC2 的長期安全開機程序，提供額外的深度防禦，幫助客戶保護軟體免受重新開機期間持續存在的威脅。	2022 年 5 月 10 日
NitroTPM	Nitro 信任平台模組 (NitroTPM) 是由 AWS Nitro 系統提供的虛擬裝置，符合 TPM 2.0 規格。	2022 年 5 月 10 日
AMI 狀態變更事件	現在當 AMI 狀態變更時，Amazon EC2 會產生一個事件。您可以使用 Amazon EventBridge 來偵測並對這些事件做出反應。	2022 年 5 月 9 日

描述公有金鑰	您可以查詢 Amazon EC2 金鑰對的公有金鑰和建立日期。	2022 年 4 月 28 日
建立金鑰對	您可以在建立新的金鑰對時指定金鑰格式 (PEM 或 PPK)。	2022 年 4 月 28 日
在啟動時掛載 Amazon FSx 檔案系統	您可以使用新的啟動執行個體精靈，在啟動時掛載新的或現有的 Amazon FSx for NetApp ONTAP 或 Amazon FSx for OpenZFS 檔案系統。	2022 年 4 月 12 日
新的啟動執行個體精靈	Amazon EC2 主控台中全新和改善的啟動體驗，提供更快、更輕鬆啟動 EC2 執行個體的方式。	2022 年 4 月 5 日
自動棄用公用 AMI	在預設情況下，所有公用 AMI 的棄用日期都設定為自 AMI 建立日期起兩年。	2022 年 3 月 31 日
執行個體中繼資料類別：autoscaling/target-lifecycle-state	使用 Auto Scaling 群組時，您可以從執行個體中繼資料中存取執行個體的目標生命週期狀態。	2022 年 3 月 24 日
AMI 上次啟動時間	`lastLaunchedTime` 表示您的 AMI 上次用於啟動執行個體的時間。	2022 年 2 月 28 日
ED25519 金鑰	EC2 執行個體 Connect 和 EC2 序列主控台不支援 ED25519 金鑰。	2022 年 1 月 20 日
用於容量保留的額外 RHEL 平台	適用於按需容量保留的其他 Red Hat Enterprise Linux 平台。	2022 年 1 月 11 日

設定 Windows AMI 以實現更快速啟動	使用預先佈建快照設定 Windows AMI，將啟動執行個體的速度提高 65%。	2022 年 1 月 10 日
執行個體中繼資料中的執行個體標籤	您可以從執行個體中繼資料存取執行個體的標籤。	2022 年 1 月 6 日
集群放置群組中的容量保留	可以在集群放置群組中建立容量保留。	2022 年 1 月 6 日
Spot 機群終止前啟動	Spot 機群可以在新的替代 Spot 執行個體啟動之後，終止收到重新平衡通知的 Spot 執行個體。	2021 年 11 月 4 日
EC2 機群終止前啟動	EC2 機群可以在新的替代 Spot 執行個體啟動之後，終止收到重新平衡通知的 Spot 執行個體。	2021 年 11 月 4 日
比較時間戳記	您可以將 Amazon EC2 Linux 執行個體的時間戳記與 ClockBound 進行比較，來判斷事件的真實時間。	2021 年 11 月 2 日
與組織和 OU 共用 AMI	您現在可以與下列 AWS 資源共用 AMIs：組織和組織單位 OUs)。	2021 年 10 月 29 日
Spot 配置分數	根據您的 Spot 容量需求，取得 AWS 區域或可用區域的建議。	2021 年 10 月 27 日
Spot 機群的屬性型執行個體類型選取	指定執行個體必須具有的屬性，然後 Amazon EC2 會識別具有這些屬性的所有執行個體類型。	2021 年 10 月 27 日

EC2 機群的屬性型執行個體類型選取	指定執行個體必須具有的屬性，然後 Amazon EC2 會識別具有這些屬性的所有執行個體類型。	2021 年 10 月 27 日
隨需容量預留機群	您可以使用容量預留機群，啟動一個群組或機群的容量預留。	2021 年 10 月 5 日
Ubuntu 20.04 LTS - Focal 的休眠支援	讓從 Ubuntu 20.04 LTS - Focal AMI 啟動的新啟動執行個體休眠。	2021 年 10 月 4 日
EC2 Fleet 和目標隨需容量預留	EC2 Fleet 可以按照 `targeted` 容量預留啟動隨需執行個體。	2021 年 9 月 22 日
專用主機上的 T3 執行個體	Amazon EC2 專用主機上的 T3 執行個體支援。	2021 年 9 月 14 日
針對 RHEL、Fedora 和 CentOS 的休眠支援	對您新啟動且從 RHEL、Fedora 和 CentOS AMI 啟動的執行個體進行休眠。	2021 年 9 月 9 日
Amazon EC2 全域檢視	Amazon EC2 Global View 可讓您在單一主控台中檢視跨多個 AWS 區域的 VPCs、子網路、執行個體、安全群組和磁碟區。	2021 年 9 月 1 日
C5d、M5d 和 R5d 的休眠支援	對您新啟動且在 C5d、M5d 和 R5d 執行個體類型上執行的執行個體進行休眠。	2021 年 8 月 19 日
Amazon EC2 金鑰對	Amazon EC2 現已支援 Linux 和 Mac 執行個體上的 ED25519 金鑰。	2021 年 8 月 17 日

網路介面字首	您可以自動或手動將私有 IPv4 或 IPv6 CIDR 範圍指派給網絡介面。	2021 年 7 月 22 日
事件時段	您可以為重新啟動、停止或終止 Amazon EC2 執行個體的排程事件定義自訂、每週重複的事件時段。	2021 年 7 月 15 日
支援安全群組規則的資源 ID 和標記	您可以依資源 ID 參考安全群組規則。您也可以為安全群組規則新增標籤。	2021 年 7 月 7 日
取代 AMI	您現在可以指定何時取代 AMI。	2021 年 6 月 11 日
Windows 每秒計費	Amazon EC2 會按秒收取 Windows 和 SQL Server 用量的費用，最低收費為一分鐘。	2021 年 6 月 10 日
上的容量保留 AWS Outposts	您現在可以在 AWS Outposts 上使用容量預留。	2021 年 5 月 24 日
共用容量保留	您現在可以共用在 Local Zones 和 Wavelength 區域中建立的容量預留。	2021 年 5 月 24 日
根磁碟區更換	您現在可以使用根磁碟區更換任務來取代執行中執行個體的根 EBS 磁碟區。	2021 年 4 月 22 日
使用 S3 存放和還原 AMI	將 EBS 後端 AMI 存放在 S3 中，並從 S3 將其還原以啟用跨分割區複製 AMI。	2021 年 4 月 6 日
EC2 序列主控台	透過建立執行個體的序列埠連線，疑難排解開機和網路連線問題。	2021 年 3 月 30 日

開機模式	Amazon EC2 現在支援在選定 AMD 和 Intel EC2 執行個體上進行 UEFI 開機。	2021 年 3 月 22 日
建立反向 DNS 記錄	您現在可以為彈性 IP 地址設定反向 DNS 查詢。	2021 年 2 月 3 日
在 AMI 建立時標記 AMI 和快照	當您建立 AMI 時，可以使用相同的標籤來標記 AMI 和快照，或者可以使用不同的標籤來進行標記。	2020 年 12 月 4 日
使用 Amazon EventBridge 來監控 Spot Fleet 事件	建立 EventBridge 規則，此規則可觸發程式設計動作以回應 Spot Fleet 狀態的變更和錯誤。	2020 年 11 月 20 日
使用 Amazon EventBridge 來監控 EC2 Fleet 事件	建立 EventBridge 規則，此規則可觸發程式設計動作以回應 EC2 Fleet 狀態的變更和錯誤。	2020 年 11 月 20 日
刪除 instant 機群	刪除 類型 `instant` 的 EC2 Fleet，並以單一 API 呼叫來終止機群中的所有執行個體。	2020 年 11 月 18 日
對 T3 和 T3a 的休眠支援	對新啟動且在 T3 和 T3a 執行個體類型上執行的執行個體進行休眠。	2020 年 11 月 17 日
Amazon EFS Quick Create	您可以在啟動時使用 Amazon EFS Quick Create 建立 Amazon EFS 檔案系統並將其掛載到執行個體。	2020 年 11 月 9 日
執行個體中繼資料類別：events/recommendations/rebalance	針對執行個體發出 EC2 執行個體重新平衡建議通知的大約時間 (UTC)。	2020 年 11 月 4 日

EC2 執行個體重新平衡推薦	當 Spot 執行個體的中斷風險提高時，會以訊號來通知您。	2020 年 11 月 4 日
在 Wavelength 區域中的容量保留	容量預留 現在可以在 Wavelength 區域中建立和使用。	2020 年 11 月 4 日
容量重新平衡	設定 Spot Fleet 或 EC2 Fleet，以在 Amazon EC2 發出重新平衡建議時啟動替代 Spot 執行個體。	2020 年 11 月 4 日
對 I3、M5ad 和 R5ad 的休眠支援	對您新啟動且在 I3、M5ad 和 R5ad 執行個體類型上執行的執行個體進行休眠。	2020 年 10 月 21 日
Spot 執行個體 vCPU 限制	Spot 執行個體限制現在是根據執行中的 Spot 執行個體正在使用或將使用的 vCPU 的數量來管理的，直到完成開放請求。	2020 年 10 月 1 日
Local Zones 中的容量保留	容量預留現在可以在 Local Zones 中建立和使用。	2020 年 9 月 30 日
對 M5a 和 R5a 的休眠支援	對您新啟動且在 M5a 和 R5a 執行個體類型上執行的執行個體進行休眠。	2020 年 8 月 28 日
執行個體中繼資料提供執行個體位置和放置資料	`placement` 類別下的新執行個體中繼資料欄位：區域、置放群組名稱、磁碟分割號碼、主機 ID 和可用區域 ID。	2020 年 8 月 24 日
容量保留群組	您可以使用 AWS 資源群組來建立容量保留的邏輯集合，然後將執行個體啟動至這些群組。	2020 年 7 月 29 日

EC2Launch v2	您可以使用 EC2Launch v2 在執行個體啟動期間執行任務，並在執行個體停止後再啟動或重新啟動時執行，也可以依需求執行任務。EC2Launch v2 支援所有版本的 Windows Server，並取代 EC2Launch 和 EC2Config。	2020 年 6 月 30 日
攜帶自有 IPv6 位址	您可以將部分或所有 IPv6 地址範圍從內部部署網路帶到 AWS 您的帳戶。	2020 年 5 月 21 日
使用 Systems Manager 參數啟動執行個體	您可以在啟動執行個體時指定 AWS Systems Manager 參數，而非 AMI。	2020 年 5 月 5 日
自訂排程的事件通知	您可以自訂排程事件通知，以便在電子郵件通知中包含標籤。	2020 年 5 月 4 日
在專用主機上的 Windows Server	您可以使用由 Amazon 提供的 Windows Server AMI 在專用主機上運行最新版本的 Windows Server。	2020 年 4 月 7 日
停止並啟動 Spot 執行個體	停止您由 Amazon EBS 支援的 Spot 執行個體，並隨時啟動它們，而不需要依賴停止中斷行為。	2020 年 1 月 13 日
資源標記	您可以標記輸出限定網際網路閘道、本機閘道、本機閘道路由表、本機閘道虛擬介面、本機閘道虛擬介面群組、本機閘道路由表 VPC 關聯，以及本機閘道路由表虛擬介面群組關聯。	2020 年 1 月 10 日

使用工作階段管理員連線至執行個體	您可以從 Amazon EC2 主控台使用一個執行個體啟動工作階段管理員工作階段。	2019 年 12 月 18 日
專用主機和主機資源群組	專用主機現在可與主機資源群組搭配使用。	2019 年 12 月 2 日
專用主機共用	您現在可以跨 AWS 帳戶共用專用主機。	2019 年 12 月 2 日
帳戶層級的預設額度規格	您可以在每個 AWS 區域的帳戶層級設定每個爆量效能執行個體系列的預設點數規格。	2019 年 11 月 25 日
執行個體類型探索	您可以尋找符合您需求的執行個體類型。	2019 年 11 月 22 日
專用主機	您現在可以將 專用主機 設定為支援執行個體系列中的多個執行個體類型。	2019 年 11 月 21 日
執行個體中繼資料服務第 2 版	您可以使用 執行個體中繼資料服務第 2 版，這種工作階段導向的方法，可讓您請求執行個體中繼資料。	2019 年 11 月 19 日
Elastic Fabric Adapter	Elastic Fabric Adapter 現在可與 Intel MPI 2019 Update 6 搭配使用。	2019 年 11 月 15 日
休眠支援隨需 Windows 執行個體	您可以讓隨需 Windows 執行個體進入休眠。	2019 年 10 月 14 日
已將預留執行個體購買放入佇列	最早可提前三年將 預留執行個體 購買放入佇列。	2019 年 10 月 4 日
診斷中斷	您可以將診斷中斷傳送至無法連線或沒有回應的執行個體，以觸發核心錯誤。	2019 年 8 月 14 日

容量最佳化分配策略	使用 EC2 Fleet 或 Spot Fleet，可從 Spot 集區以適用於所啟動之執行個體數量的最佳容量來啟動 Spot 執行個體。	2019 年 8 月 12 日
隨需容量保留共用	您現在可以跨 AWS 帳戶共用容量保留。	2019 年 7 月 29 日
Elastic Fabric Adapter	EFA 現在支援 Open MPI 3.1.4 和 Intel MPI 2019 Update 4。	2019 年 7 月 26 日
EC2 執行個體連線	EC2 Instance Connect 為使用 Secure Shell (SSH) 連接到執行個體的簡單且安全方式。	2019 年 6 月 27 日
主機復原	當 專用主機 上發生非預期的硬體失敗時，在新主機上自動重新啟動您的執行個體。	2019 年 6 月 5 日
VSS 應用程式一致性快照	使用 AWS Systems Manager Run Command 擷取連接至 Windows 執行個體之所有 Amazon EBS 磁碟區的應用程式一致性快照。	2019 年 5 月 13 日
從 Windows 到 Linux 的 Microsoft SQL Server 資料庫平台重建助理	將現有的 Microsoft SQL Server 工作負載從 Windows 移到 Linux 作業系統。已更新「Amazon EC2 使用者指南」的 Microsoft SQL Server 連結點。	2019 年 5 月 8 日
Windows 自動升級	使用 執行 EC2 Windows 執行個體的自動升級 AWS Systems Manager。	2019 年 5 月 6 日

Elastic Fabric Adapter	您可以將 Elastic Fabric Adapter 連接到執行個體，以加速高效能運算 (HPC) 應用程式。	2019 年 4 月 29 日

如需有關 Amazon EC2 執行個體類型版本的資訊，請參閱「Amazon EC2 執行個體類型指南」中的文件歷程記錄。

2018 年及以前的歷程記錄

下表說明 2018 年和較早年度對「Amazon EC2 使用者指南」的重要補充內容。

功能	API 版本	描述	發行日期
分區置放群組	2016-11-15	分區置放群組會將執行個體分散到邏輯分區，確保一個分區中的執行個體不會與其他分區中的執行個體共用底層硬體。如需詳細資訊，請參閱分區置放群組。	2018 年 12 月 20 日
讓 EC2 Linux 執行個體進入休眠	2016-11-15	當您的 Linux 執行個體已啟用休眠並符合休眠必要條件時，您才能讓執行個體進入休眠。如需詳細資訊，請參閱 讓 Amazon EC2 執行個體進入休眠。	2018 年 11 月 28 日
Amazon Elastic Inference Accelerator	2016-11-15	您可以將 Amazon EI 加速器連接至執行個體，並新增 GPU 加速以降低執行深度學習推論的成本。	2018 年 11 月 28 日
Spot 主控台建議使用眾多執行個體	2016-11-15	Spot 主控台建議您依據 Spot 最佳實務使用眾多執行個體 (執行個體多角化)，以滿足應用程式所需的最低硬體規格 (vCPU、記憶體和儲存)。如需詳細資訊，請參閱 建立 Spot 機群。	2018 年 11 月 20 日
新的 EC2 Fleet 請求類型：instant	2016-11-15	EC2 Fleet 現可支援新的 `instant` 請求類型，您可用來跨執行個體類型同步佈建容量，並購買模型。`instant` 請求會在 API 回應中傳回啟動的執行個體，但不會採取進一步動作，讓您可以	2018 年 11 月 14 日

功能	API 版本	描述	發行日期
		控制是否啟動執行個體以及啟動時機。如需詳細資訊，請參閱 EC2 機群和 Spot 機群請求類型。	
Spot 節省資訊	2016-11-15	您可以檢視針對單一 Spot Fleet 或所有 Spot 執行個體使用 Spot 執行個體所節省的成本。如需詳細資訊，請參閱 購買 競價型執行個體省下的成本。	2018 年 11 月 5 日
主控台支援最佳化 CPU 選項	2016-11-15	當您啟動執行個體時，您可以使用 Amazon EC2 主控台最佳化 CPU 選項以符合特定工作負載或商業需求。如需詳細資訊，請參閱 Amazon EC2 執行個體的 CPU 選項。	2018 年 10 月 31 日
主控台支援從執行個體建立啟動範本	2016-11-15	您可以使用 Amazon EC2 主控台並將執行個體做為新啟動範本的基礎，以建立新的啟動範本。如需詳細資訊，請參閱 建立 Amazon EC2 啟動範本。	2018 年 10 月 30 日
On-Demand Capacity Reservations	2016-11-15	在特定的可用區域中，您可以為 Amazon EC2 執行個體預留任何期限的容量。這可讓您建立和管理容量保留，而不受預留執行個體 (RI) 的帳單折扣影響。如需詳細資訊，請參閱 使用 EC2 隨需容量保留來保留運算容量。	2018 年 10 月 25 日
使用自有 IP 地址 (BYOIP)	2016-11-15	您可以將部分或所有公有 IPv4 地址範圍從內部部署網路帶到 AWS 您的帳戶。將地址範圍帶至 之後 AWS，其會在您的帳戶中顯示為地址集區。您可以從您的地址集區建立彈性 IP 地址，並將它用於您的 AWS 資源。如需詳細資訊，請參閱 將自有 IP 位址 (BYOIP) 用於 Amazon EC2。	2018 年 10 月 23 日
在建立專用主機時套用標籤和主控台支援	2016-11-15	您可以在建立您的專用主機時套用標籤，並使用 Amazon EC2 主控台來管理您的專用主機標籤。如需詳細資訊，請參閱 配置 Amazon EC2 專用主機供您的帳戶使用。	2018 年 10 月 8 日

功能	API 版本	描述	發行日期
對 Spot Fleet 的排程擴展的主控台支援	2016-11-15	根據日期和時間，來增加或減少機群目前的容量。如需詳細資訊，請參閱 排程擴展：依排程擴展 Spot 機群。	2018 年 9 月 20 日
EC2 Fleet 的分配策略	2016-11-15	您可以指定隨需容量是由價格 (最低價格優先) 或優先順序 (最高優先順序優先) 履行。您可以指定要配置目標 Spot 容量的 Spot 集區數量。如需詳細資訊，請參閱 使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量。	2018 年 7 月 26 日
Spot Fleets 的分配策略	2016-11-15	您可以指定隨需容量是由價格 (最低價格優先) 或優先順序 (最高優先順序優先) 履行。您可以指定要配置目標 Spot 容量的 Spot 集區數量。如需詳細資訊，請參閱 使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量。	2018 年 7 月 26 日
自動快照生命週期	2016-11-15	您可以使用 Amazon Data Lifecycle Manager 來自動化建立和刪除 EBS 磁碟區的快照。如需詳細資訊，請參閱 Amazon Data Lifecycle Manager。	2018 年 7 月 12 日
啟動範本 CPU 選項	2016-11-15	當您使用命令列工具建立啟動範本時，您可以最佳化 CPU 選項來符合特定工作負載或商務需求。如需詳細資訊，請參閱 建立 Amazon EC2 啟動範本。	2018 年 7 月 11 日
標記 專用主機	2016-11-15	您可以標記您的 專用主機。	2018 年 7 月 3 日
取得最新主控台輸出	2016-11-15	當您使用 get-console-output AWS CLI 命令時，可以擷取某些執行個體類型的最新主控台輸出。	2018 年 5 月 9 日

功能	API 版本	描述	發行日期
最佳化 CPU 選項	2016-11-15	當您啟動執行個體時，您可以最佳化 CPU 選項來符合特定工作負載或商業需求。如需詳細資訊，請參閱 Amazon EC2 執行個體的 CPU 選項。	2018 年 5 月 8 日
EC2 Fleet	2016-11-15	您可以使用 EC2 Fleet 跨不同 EC2 執行個體類型和可用區域，並且跨隨需執行個體、保留執行個體和 Spot 執行個體購買模型來啟動一組執行個體。如需詳細資訊，請參閱EC2 機群 和 Spot 機群。	2018 年 5 月 2 日
Spot Fleets 中的隨需執行個體	2016-11-15	您可以在 Spot Fleet 請求中，加入對隨需容量的請求，來確保永遠有執行個體容量。如需詳細資訊，請參閱EC2 機群 和 Spot 機群。	2018 年 5 月 2 日
建立時標記 EBS 快照	2016-11-15	您可以在建立期間將標籤套用至快照。	2018 年 4 月 2 日
變更置放群組	2016-11-15	您可以將執行個體移入或移出置放群組，或變更其置放群組。如需詳細資訊，請參閱 變更 EC2 執行個體的放置位置。	2018 年 3 月 1 日
較長的資源 ID	2016-11-15	您可以為更多資源類型啟用較長的 ID 格式。	2018 年 2 月 9 日
網路效能提升	2016-11-15	在其他執行個體或 Amazon S3 之間傳送或接收網路流量時，集群放置群組之外的執行個體現在可享有更高的頻寬。	2018 年 1 月 24 日
標記彈性 IP 地址	2016-11-15	您可以標記彈性 IP 地址。	2017 年 12 月 21 日
Amazon Time Sync Service	2016-11-15	您可以使用 Amazon Time Sync Service 來確保執行個體的時間正確。如需詳細資訊，請參閱EC2 執行個體上的精確時鐘和時間同步。	2017 年 11 月 29 日

功能	API 版本	描述	發行日期
無限制 T2	2016-11-15	T2 效能不受限執行個體可隨時按需要大幅提升效能以超越基準。如需詳細資訊，請參閱 爆量效能執行個體。	2017 年 11 月 29 日
啟動範本	2016-11-15	啟動範本可包含用以啟動執行個體的所有或部分參數，因此您不需要在每次啟動執行個體時指定。如需詳細資訊，請參閱 將執行個體啟動參數存放在 Amazon EC2 啟動範本中。	2017 年 11 月 29 日
分散置放	2016-11-15	若應用程式具有應該分開保存的少量重要執行個體，建議使用分散置放群組。如需詳細資訊，請參閱 分散放置群組。	2017 年 11 月 29 日
Spot 執行個體休眠	2016-11-15	Spot 服務可以在發生中斷時讓 Spot 執行個體休眠。	2017 年 11 月 28 日
Spot Fleet 目標追蹤	2016-11-15	您可以為 Spot Fleet 設定目標追蹤擴展政策。如需詳細資訊，請參閱 目標追蹤擴展：透過為特定指標設定目標值，擴展 Spot 機群。	2017 年 11 月 17 日
Spot Fleet 與 Elastic Load Balancing 整合	2016-11-15	您可以將一或多個負載平衡器連接至 Spot Fleet。	2017 年 11 月 10 日
合併及分割 可轉換預留執行個體	2016-11-15	您可以為新的 可轉換預留執行個體 交換 (合併) 兩個或多個 可轉換預留執行個體。您也可以使用修改程序將可轉換預留執行個體分割為較小的保留。如需詳細資訊，請參閱 交換 可轉換預留執行個體。	2017 年 11 月 6 日
修改 VPC 租用	2016-11-15	您可將 VPC 的執行個體租用屬性從 `dedicated` 變更為 `default`。如需詳細資訊，請參閱 變更 VPC 的執行個體租用。	2017 年 10 月 16 日

功能	API 版本	描述	發行日期
每秒計費	2016-11-15	Amazon EC2 會按秒收取 Linux 用量的費用，最低收費為一分鐘。	2017 年 10 月 2 日
中斷時停止	2016-11-15	您可以指定當 競價型執行個體中斷時，Amazon EC2 應該停止或終止它們。如需詳細資訊，請參閱 Spot 執行個體中斷的行為。	2017 年 9 月 18 日
標記 NAT 閘道	2016-11-15	您可以標記 NAT 閘道。如需詳細資訊，請參閱標記您的 資源。	2017 年 9 月 7 日
安全群組規則說明	2016-11-15	您可以為安全群組規則新增說明。	2017 年 8 月 31 日
Elastic Graphics	2016-11-15	將 Elastic Graphics 加速器連接至您的執行個體，來加速應用程式的圖形效能。	2017 年 8 月 29 日
復原彈性 IP 地址	2016-11-15	如果您發行用於 VPC 的彈性 IP 地址，您可以將它復原。	2017 年 8 月 11 日
標記 Spot Fleet 執行個體	2016-11-15	您可以設定 Spot Fleet 以自動標記其啟動的執行個體。	2017 年 7 月 24 日
在建立期間標記資源	2016-11-15	您可以在建立期間將標籤套用至執行個體和磁碟區。如需詳細資訊，請參閱 標記您的 資源。此外，您也可以使用標籤型資源層級許可，來控制套用的標籤。如需更多詳細資訊，請參閱 在建立期間授予標籤 Amazon EC2 資源的許可。	2017 年 3 月 28 日
在連接的 EBS 磁碟區上執行修改	2016-11-15	有了連接至大多數 EC2 執行個體的大多數 EBS 磁碟區，您可以修改磁碟區大小、類型和 IOPS，無需分離磁碟區或停止執行個體。	2017 年 2 月 13 日

功能	API 版本	描述	發行日期
連接 IAM 角色	2016-11-15	您可以連接、分離或取代現有執行個體的 IAM 角色。如需詳細資訊，請參閱 Amazon EC2 的 IAM 的角色。	2017 年 2 月 9 日
專用的 競價型執行個體	2016-11-15	您可以在 Virtual Private Cloud (VPC) 中的單一租戶硬體上執行 競價型執行個體。如需詳細資訊，請參閱 在單一租戶硬體上啟動。	2017 年 1 月 19 日
IPv6 支援	2016-11-15	您可以將 IPv6 CIDR 與 VPC 和子網建立關聯，並指派 IPv6 地址給 VPC 中的執行個體。如需詳細資訊，請參閱 Amazon EC2 執行個體 IP 定址。	2016 年 12 月 1 日
Spot Fleet 的自動擴展		您現在可以為 Spot Fleet 設定擴展政策。如需詳細資訊，請參閱 了解 Spot 機群的自動擴展。	2016 年 9 月 1 日
彈性網路轉接器 (ENA)	2016-04-01	您現在可以將 ENA 用於增強聯網。如需詳細資訊，請參閱增強 Amazon EC2 執行個體的聯網能力。	2016 年 6 月 28 日
檢視及修改較長 ID 的增強型支援	2016-04-01	您現在可以檢視及修改其他 IAM 使用者、IAM 角色或根使用者的較長 ID 設定。	2016 年 6 月 23 日
在 AWS 帳戶之間複製加密的 Amazon EBS 快照	2016-04-01	您現在可以在 AWS 帳戶之間複製加密的 EBS 快照。	2016 年 6 月 21 日
擷取執行個體主控台的螢幕擷取畫面	2015-10-01	當您對無法連線的執行個體除錯時，您現在可以取得其他資訊。如需詳細資訊，請參閱擷取無法連線執行個體的螢幕擷取畫面。	2016 年 5 月 24 日
兩個新的 EBS 磁碟區類型	2015-10-01	您現在可以建立輸送量最佳化 HDD (st1) 和 Cold HDD (sc1) 磁碟區。	2016 年 4 月 19 日

功能	API 版本	描述	發行日期
新增 Amazon EC2 的 NetworkPacketsIn 和 NetworkPacketsOut 指標		新增 Amazon EC2 的 NetworkPacketsIn 和 NetworkPacketsOut 指標。如需詳細資訊，請參閱 執行個體指標。	2016 年 3 月 23 日
Spot Fleet 的 CloudWatch 指標		您現在可取得 Spot Fleet 的 CloudWatch 指標。如需詳細資訊，請參閱使用 CloudWatch 監控 EC2 機群或 Spot 機群。	2016 年 3 月 21 日
排程執行個體	2015-10-01	排程的預留執行個體 (排程執行個體) 可讓您以指定的開始時間和期間，購買每日、每週或每月的容量保留。	2016 年 1 月 13 日
較長的資源 ID	2015-10-01	Amazon EC2 正在逐步為某些 Amazon EC2 和 Amazon EBS 資源類型引入較長IDs。在選擇期間，您可以為支援的資源類型啟用較長的 ID 格式。	2016 年 1 月 13 日
ClassicLink DNS 支援	2015-10-01	您可以為 VPC 啟用 ClassicLink DNS 支援，以將定址於連結的 EC2-Classic 執行個體與 VPC 中的執行個體之間的 DNS 主機名稱解析為私有 IP 地址，而不是公有 IP 地址。	2016 年 1 月 11 日
專用主機	2015-10-01	Amazon EC2 專用主機是具有專供您使用之執行個體容量的實體伺服器。如需詳細資訊，請參閱 Amazon EC2 專用執行個體。	2015 年 11 月 23 日
Spot 執行個體持續時間	2015-10-01	您現在可以為 競價型執行個體指定持續時間。不支援 Spot 區塊 (2023 年 1 月)。	2015 年 10 月 6 日
Spot Fleet 修改請求	2015-10-01	您現在可以修改 Spot Fleet 請求的目標容量。如需詳細資訊，請參閱 修改 Spot Fleet 請求。	2015 年 9 月 29 日

功能	API 版本	描述	發行日期
Spot Fleet 多樣化分配策略	2015-04-15	您現在可以使用單一 Spot Fleet 請求，將 Spot 執行個體配置到多個 Spot 集區。如需詳細資訊，請參閱使用配置策略來判斷 EC2 機群或 Spot 機群如何達成 Spot 和隨需容量。	2015 年 9 月 15 日
Spot Fleet 執行個體權重	2015-04-15	您現在可以定義每個執行個體類型對應用程式效能所貢獻的容量單位，並據以調整每個 Spot 集區您願意為 競價型執行個體支付的價格。如需詳細資訊，請參閱使用執行個體權重來管理 EC2 機群或 Spot 機群的成本和效能。	2015 年 8 月 31 日
新的重新啟動警示動作和用於警示動作的新 IAM 角色		新增重新啟動警示動作和用於警示動作的新 IAM 角色。如需詳細資訊，請參閱 建立警示以停止、終止、重新啟動或復原執行個體。	2015 年 7 月 23 日
Spot Fleets	2015-04-15	您可以管理 Spot 執行個體集合或機群，而不是管理個別 Spot 執行個體請求。如需詳細資訊，請參閱EC2 機群 和 Spot 機群。	2015 年 5 月 18 日
將彈性 IP 地址遷移至 EC2-Classic	2015-04-15	您無法將配置用於 EC2-Classic 的彈性 IP 地址遷移為在 VPC 中使用。	2015 年 5 月 15 日
將 VM 與多個磁碟匯入為 AMI	2015-03-01	VM Import 程序現在支援將 VM 與多個磁碟匯入為 AMI。如需詳細資訊，請參閱VM Import/Export 使用者指南中的使用 VM 匯入/匯出將 VM 匯入為映像。	2015 年 4 月 23 日
Systems Manager		Systems Manager 可讓您設定及管理 EC2 執行個體。	2015 年 2 月 17 日
Systems Manager for Microsoft SCVMM 1.5		您現在可以使用 Systems Manager for Microsoft SCVMM 來啟動執行個體，並將 VM 從 SCVMM 匯入 Amazon EC2。	2015 年 1 月 21 日

功能	API 版本	描述	發行日期
EC2 執行個體的自動復原功能		您可以建立 Amazon CloudWatch 警示來監控 Amazon EC2 執行個體，並在執行個體因基礎硬體故障或需要 AWS 介入修復的問題而受損時自動復原執行個體。復原的執行個體與原始執行個體相同，包括執行個體 ID、IP 地址及所有執行個體中繼資料。 如需詳細資訊，請參閱 自動執行個體復原功能。	2015 年 1 月 12 日
ClassicLink	2014-10-01	ClassicLink 可讓您將 EC2-Classic 執行個體連結至帳戶中的 VPC。您可以將 VPC 安全群組與 EC2-Classic 執行個體建立關聯，讓 EC2-Classic 執行個體可以使用私有 IP 地址與您 VPC 中的執行個體通訊。	2015 年 1 月 7 日
Spot 執行個體終止通知		防止 Spot 執行個體中斷的最佳方法就是建立容錯的應用程式架構。此外，您也可以使用 Spot 執行個體終止通知，此通知會在 Amazon EC2 必須終止 Spot 執行個體的兩分鐘前發出警告。 如需詳細資訊，請參閱 Spot 執行個體中斷通知。	2015 年 1 月 5 日
Systems Manager for Microsoft SCVMM		Systems Manager for Microsoft SCVMM 提供簡單且easy-to-use界面，用於從 Microsoft SCVMM 管理 AWS 資源，例如 EC2 執行個體。	2014 年 10 月 29 日
`DescribeVolumes` 分頁支援	2014-09-01	`DescribeVolumes` API 呼叫現在支援使用 `MaxResults` 和`NextToken` 參數來分頁結果。如需更多資訊，請參閱Amazon EC2 API Reference中的 DescribeVolumes。	2014 年 10 月 23 日

功能	API 版本	描述	發行日期
新增對 Amazon CloudWatch Logs 的支援		您可以使用 Amazon CloudWatch Logs 來監控、存放和存取您的系統、應用程式，以及來自執行個體或其他來源的自訂日誌檔案。然後，您可以使用 Amazon CloudWatch 主控台、CLI 中的 CloudWatch Logs AWS 命令或 CloudWatch Logs SDK，從 CloudWatch Logs 擷取相關聯的日誌資料。	2014 年 7 月 10 日
新的 EC2 Service Limits (EC2 服務限制) 頁面		使用 Amazon EC2 主控台中的 EC2 Service Limits (EC2 服務限制) 頁面，檢視 Amazon EC2 及 Amazon VPC 為每個區域提供的目前資源限制。	2014 年 6 月 19 日
Amazon EBS 一般用途 SSD 磁碟區	2014-05-01	一般用途 SSD 磁碟區提供經濟實惠的儲存空間，適合絕大多數的工作負載。這些磁碟區的延遲時間不到 10 毫秒，且可在延長的時間內爆量至 3,000 IOPS，且其基礎效能為 3 IOPS/GiB。一般用途 SSD 磁碟區的大小範圍可從 1 GiB 到 1 TiB。	2014 年 6 月 16 日
AWS Management Pack		AWS Management Pack 現在支援 System Center Operations Manager 2012 R2。	2014 年 5 月 22 日
Amazon EBS encryption	2014-05-01	Amazon EBS 加密 可讓您順暢地加密 EBS 資料磁碟區和快照，無需建立及維護安全金鑰管理基礎設施。EBS 加密透過使用 AWS 受管金鑰加密資料，來啟用靜態資料安全。還可以在託管 EC2 執行個體的伺服器上進行加密，當資料在 EC2 執行個體和 EBS 儲存之間移動時，提供資料加密。	2014 年 5 月 21 日
Amazon EC2 用量報告		Amazon EC2 用量報告是一組顯示使用 EC2 的成本和用量資料之報告。	2014 年 1 月 28 日

功能	API 版本	描述	發行日期
匯入 Linux 虛擬機器	2013-10-15	VM Import 程序現在支援匯入 Linux 執行個體。如需詳細資訊，請參閱 VM Import/Export 使用者指南。	2013 年 12 月 16 日
RunInstances 的資源層級許可	2013-10-15	您現在可以在 中建立政策 AWS Identity and Access Management ，以控制 Amazon EC2 RunInstances API 動作的資源層級許可。如需詳細資訊和範例政策，請參閱Amazon EC2 的 Identity and Access Management。	2013 年 11 月 20 日
從 啟動執行個體 AWS Marketplace		您現在可以 AWS Marketplace 使用 Amazon EC2 啟動精靈從 啟動執行個體。如需詳細資訊，請參閱從 AWS Marketplace AMI 啟動 Amazon EC2 執行個體。	2013 年 11 月 11 日
新的 Launch Wizard		新增並重新設計 EC2 Launch Wizard。如需詳細資訊，請參閱使用主控台中的啟動執行個體精靈，啟動 EC2 執行個體。	2013 年 10 月 10 日
修改保留執行個體的執行個體類型	2013-10-01	您現在可以修改相同系列之 Linux 預留執行個體的執行個體類型 (例如 M1、M2、M3、C1)。如需詳細資訊，請參閱 修改 預留執行個體。	2013 年 10 月 9 日
修改 Amazon EC2 預留執行個體	2013-08-15	您現在可以修改區域中的預留執行個體。如需更多詳細資訊，請參閱 修改 預留執行個體。	2013 年 9 月 11 日
指派公有 IP 地址	2013-07-15	您現在可以在啟動 VPC 中的執行個體時，指派公有 IP 地址。如需詳細資訊，請參閱 在啟動時指派公有 IPv4 位址。	2013 年 8 月 20 日
授予資源層級許可	2013-06-15	Amazon EC2 支援新的 Amazon Resource Name (ARN) 和條件金鑰。如需詳細資訊，請參閱 Amazon EC2 身分型政策。	2013 年 7 月 8 日

功能	API 版本	描述	發行日期
增量式快照複本	2013-02-01	您現在可以執行增量式快照複本。	2013 年 6 月 11 日
AWS Management Pack		AWS Management Pack 會連結 Amazon EC2 執行個體，以及在其中執行的 Windows 或 Linux 作業系統。 AWS Management Pack 是 Microsoft System Center Operations Manager 的延伸。	2013 年 5 月 8 日
新的 Tags (標籤) 頁面		Amazon EC2 主控台中新增 Tags (標籤) 頁面。如需詳細資訊，請參閱 標記您的 Amazon EC2 資源。	2013 年 4 月 4 日
將 AMI 從一個區域複製到另一個區域	2013-02-01	您可以將 AMI 從一個區域複製到另一個區域，讓您能夠快速輕鬆地在多個 AWS 區域中啟動一致的執行個體。 如需詳細資訊，請參閱複製 Amazon EC2 AMI。	2013 年 3 月 11 日
將執行個體啟動至預設 VPC 中	2013-02-01	AWS 您的帳戶能夠根據region-by-region EC2-Classic 或 VPC 中啟動執行個體，或僅在 VPC 中啟動執行個體。如果您只能在 VPC 中啟動執行個體，則會為您建立預設 VPC。當您啟動執行個體時，它會在您的預設 VPC 中啟動，除非您建立非預設 VPC 並在啟動執行個體時指定它。	2013 年 3 月 11 日
EBS 快照複本	2012-12-01	您可以使用快照複本，建立資料的備份、建立新的 Amazon EBS 磁碟區，或建立 Amazon Machine Image (AMI)。	2012 年 12 月 17 日
更新Provisioned IOPS SSD 磁碟區之 EBS 指標和狀態檢查	2012-10-01	更新 EBS 指標以包含Provisioned IOPS SSD 磁碟區之兩個新指標。另外新增Provisioned IOPS SSD 磁碟區之狀態檢查。	2012 年 11 月 20 日

功能	API 版本	描述	發行日期
Spot 執行個體請求狀態	2012-10-01	Spot 執行個體請求狀態可讓您輕鬆判斷 Spot 請求的狀態。	2012 年 10 月 14 日
Amazon EC2 預留執行個體 Marketplace	2012-08-15	預留執行個體 Marketplace 會將擁有不再需要之 Amazon EC2 預留執行個體的賣方，與想要購買額外容量的買方進行比對。經由預留執行個體 Marketplace 買賣的預留執行個體運作起來與任何其他預留執行個體相同，不同之處在於可以有低於完整標準的剩餘期限，並能以不同的價格出售。	2012 年 9 月 11 日
Provisioned IOPS SSD 適用於 Amazon EBS	2012-07-20	Provisioned IOPS SSD 磁碟區為依賴一致且快速回應時間的 I/O 密集型工作負載 (例如資料庫應用程式) 提供可預測的高效能。	2012 年 7 月 31 日
Amazon EC2 執行個體上的 IAM 角色	2012-06-01	Amazon EC2 的 IAM 角色提供： • AWS 在 Amazon EC2 執行個體上執行的應用程式存取金鑰。 • 自動輪換 Amazon EC2 執行個體上的 AWS 存取金鑰。 • 在向 AWS 服務提出請求的 Amazon EC2 執行個體上執行之應用程式的精細許可。	2012 年 6 月 11 日

功能	API 版本	描述	發行日期
Spot 執行個體功能可讓您更輕鬆地開始處理可能的中斷情況。		您現在可用下列方式管理 競價型執行個體： • 使用 Auto Scaling 啟動組態指定您願意為 競價型執行個體 支付的金額，並設定排程以指定願意為 競價型執行個體 支付的金額。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 請求 Spot 執行個體。 • 啟動或終止執行個體時取得通知。 • 使用 CloudFormation 範本在具有 AWS 資源的堆疊中啟動 Spot 執行個體。	2012 年 6 月 7 日
Amazon EC2 之 EC2 執行個體匯出及狀態檢查的時間戳記	2012-05-01	新增將您原本匯入 EC2 的 Windows Server 執行個體匯出的支援。 新增對執行個體狀態和系統狀態時間戳記的支援，以指出狀態檢查失敗的日期和時間。	2012 年 5 月 25 日
Amazon VPC 之 EC2 執行個體匯出及執行個體和系統狀態檢查的時間戳記	2012-05-01	新增將 EC2 執行個體匯出至 Citrix Xen、Microsoft Hyper-V 和 VMware vSphere 的支援。 新增對執行個體和系統狀態檢查中時間戳記的支援。	2012 年 5 月 25 日
AWS Marketplace AMIs	2012-04-01	新增對 AWS Marketplace AMIs支援。	2012 年 4 月 19 日
預留執行個體定價方案	2011-12-15	新增說明如何利用預留執行個體定價方案內建的折扣定價之章節。	2012 年 3 月 5 日

功能	API 版本	描述	發行日期
Amazon Virtual Private Cloud 中 EC2 執行個體的彈性網絡介面 (ENI)	2011-12-01	新增 VPC 中 EC2 執行個體之彈性網絡介面 (ENI) 的相關章節。如需詳細資訊，請參閱 彈性網路介面。	2011 年 12 月 21 日
新的 Amazon EC2 預留執行個體方案類型	2011-11-01	您可以從各種預留執行個體方案中選擇因應您預期使用的執行個體。	2011 年 12 月 01 日
Amazon EC2 執行個體狀態	2011-11-01	您可以檢視執行個體狀態的其他詳細資訊，包括規劃的排程事件 AWS ，這些事件可能會影響您的執行個體。這些操作活動包括套用軟體更新或安全性修補程式所需的執行個體重新開機，或發生硬體問題時所需的執行個體淘汰。如需詳細資訊，請參閱 監控您的 Amazon EC2 執行個體狀態。	2011 年 11 月 16 日
Amazon VPC 中的 競價型執行個體	2011-07-15	新增對 Amazon VPC 中 競價型執行個體支援的相關資訊。透過此更新，使用者可以在 Virtual Private Cloud (VPC) 中啟動 競價型執行個體。藉由在 VPC 中啟動 競價型執行個體，競價型執行個體的使用者就可以受益於 Amazon VPC。	2011 年 10 月 11 日
為 CLI 工具的使用者簡化 VM Import 程序	2011-07-15	VM Import 程序透過增強 `ImportInstance` 和 `ImportVolume` 功能而得以簡化，現在會在建立匯入任務之後將映像上傳至 Amazon EC2。此外，藉由引進 `ResumeImport` ，使用者可以在任務停止位置重新啟動未完成的上傳。	2011 年 9 月 15 日
支援匯入 VHD 檔案格式		VM Import 現在可以匯入 VHD 格式的虛擬機器映像檔案。VHD 檔案格式與 Citrix Xen 和 Microsoft Hyper-V 虛擬化平台相容。在此版本中，VM Import 現在支援 RAW、VHD 和 VMDK (VMware ESX 相容) 映像格式。如需詳細資訊，請參閱 VM Import/Export 使用者指南。	2011 年 8 月 24 日

功能	API 版本	描述	發行日期
更新 Amazon EC2 VM Import Connector for VMware vCenter		新增 Amazon EC2 VM Import Connector for VMware vCenter 虛擬應用裝置 (連接器) 1.1 版的相關資訊。此更新包括對網際網路存取權的代理支援、更佳的錯誤處理、更精確的任務進度列，以及數種錯誤修正。	2011 年 6 月 27 日
競價型執行個體可用區域定價變更	2011-05-15	新增 競價型執行個體可用區域定價功能的相關資訊。在此版本中，新的可用區域定價選項已新增為您在查詢 Spot 執行個體請求和 Spot 價格歷史記錄時所傳回資訊的一部分。這些新增功能可讓您更輕鬆地判斷將 Spot 執行個體啟動至特定可用區域中所需的價格。	2011 年 5 月 26 日
AWS Identity and Access Management		新增有關 AWS Identity and Access Management (IAM) 的資訊，這可讓使用者指定使用者可以搭配 Amazon EC2 資源使用哪些 Amazon EC2 動作。如需詳細資訊，請參閱Amazon EC2 的 Identity and Access Management。	2011 年 4 月 26 日
專用執行個體		專用執行個體是在您 Amazon Virtual Private Cloud (Amazon VPC) 內啟動的執行個體，這些執行個體在主機硬體層級上是實體相互隔離的。專用執行個體可讓您利用 Amazon VPC 和 AWS 雲端，包括隨需彈性佈建和僅支付使用量，同時在硬體層級隔離 Amazon EC2 運算執行個體。如需詳細資訊，請參閱Amazon EC2 專用執行個體。	2011 年 3 月 27 日
AWS 管理主控台的預留執行個體更新		AWS 管理主控台的更新可讓使用者更輕鬆地檢視其預留執行個體，並購買其他預留執行個體，包括專用預留執行個體。	2011 年 3 月 27 日

功能	API 版本	描述	發行日期
中繼資料資訊	2011-01-01	新增中繼資料的相關資訊，以反映 2011-01-01 版本中的變更。如需詳細資訊，請參閱 使用執行個體中繼資料管理您的 EC2 執行個體 和 執行個體中繼資料分類。	2011 年 3 月 11 日
Amazon EC2 VM Import Connector for VMware vCenter		新增 Amazon EC2 VM Import Connector for VMware vCenter 虛擬應用裝置 (連接器) 的相關資訊。此連接器是 VMware vCenter 的外掛程式，其與 VMware vSphere Client 整合，並提供可用來將 VMware 虛擬機器匯入 Amazon EC2 的圖形使用者介面。	2011 年 3 月 3 日
強制磁碟區分離		您現在可以使用 AWS 管理主控台 強制分離 Amazon EBS 磁碟區與執行個體。	2011 年 2 月 23 日
執行個體終止保護		您現在可以使用 AWS 管理主控台來防止執行個體終止。如需詳細資訊，請參閱變更執行個體終止保護。	2011 年 2 月 23 日
VM Import	2010-11-15	新增 VM Import 的相關資訊，該工具可讓您將虛擬機器或磁碟區匯入 Amazon EC2。如需詳細資訊，請參閱 VM Import/Export 使用者指南。	2010 年 12 月 15 日
執行個體的基本監控	2010-08-31	新增 EC2 執行個體之基本監控的相關資訊。	2010 年 12 月 12 日
篩選條件和標籤	2010-08-31	新增列出、篩選及建立標籤資源的相關資訊。如需詳細資訊，請參閱 尋找您的 Amazon EC2 資源 和 標記您的 Amazon EC2 資源。	2010 年 9 月 19 日
等冪執行個體啟動	2010-08-31	新增在執行執行個體時確保等冪的相關資訊。	2010 年 9 月 19 日

功能	API 版本	描述	發行日期
AWS Identity and Access Management 適用於 Amazon EC2 的		Amazon EC2 現在已與 AWS Identity and Access Management (IAM) 整合。如需詳細資訊，請參閱 Amazon EC2 的 Identity and Access Management。	2010 年 9 月 2 日
Amazon VPC IP 地址目標	2010-06-15	Amazon VPC 使用者現在可以指定 IP 地址，來指派在 VPC 中啟動的執行個體。	2010 年 7 月 12 日
對 Amazon EBS 磁碟區的 Amazon CloudWatch 監控		現在可自動對 Amazon EBS 磁碟區進行 Amazon CloudWatch 監控。	2010 年 6 月 14 日
Windows 預留執行個體		Amazon EC2 現在支援 Windows 預留執行個體。	2010 年 2 月 22 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。