



最佳化 上的 Microsoft 工作負載成本 AWS

AWS 規範指南



AWS 規範指南: 最佳化 上的 Microsoft 工作負載成本 AWS

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

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

Table of Contents

簡介	1
概觀	1
物件	1
本指南的使用方式	1
目標業務成果	3
成本優化旅程	4
最佳化成本的主要建議	6
概觀	6
熱門推薦	6
AWS 最佳化與授權評估	8
概觀	8
評估選項	8
全面評估	9
範圍工作負	9
收集資料	9
分析資料	10
規劃後續步驟	12
評估影響	13
後續步驟	14
其他資源	14
Amazon 上的 Windows EC2	15
自動化停止和啟動排程	15
概觀	15
案例研究	16
成本最佳化案例	17
成本最佳化建議	18
其他資源	29
大小適中的 Windows 工作負載	29
概觀	29
成本最佳化案例	30
成本最佳化建議	30
建議	38
其他資源	38
為 Windows 工作負載選取正確的執行個體類型	39

概觀	39
成本最佳化建議	39
後續步驟	48
其他資源	49
為 Windows 和SQL伺服器工作負載帶來授權	49
概觀	49
Amazon EC2 專用主機	50
AWS 授權選項	53
攜帶 Windows Server 授權	53
成本最佳化案例	55
成本最佳化建議	60
其他資源	60
最佳化 Windows on Amazon 的支出 EC2	61
概觀	61
了解 Savings Plans	61
成本最佳化案例	66
成本最佳化建議	69
其他資源	71
使用 AWS 工具監控成本	71
概觀	71
成本最佳化建議	72
其他資源	75
SQL 伺服器	76
選擇高可用性和災難復原解決方案	77
概觀	77
SQL Server Always On 可用性群組	78
SQL Server Always On 容錯移轉叢集執行個體	79
SIOS DataKeeper	81
Always On 可用性群組	83
分散式可用性群組	84
日誌傳送	85
AWS Database Migration Service	87
AWS Elastic Disaster Recovery	87
成本比較	88
成本最佳化建議	91
其他資源	92

了解SQL伺服器授權	93
概觀	93
AWS 授權選項	93
取得授權的成本影響	94
授權最佳化	94
成本最佳化建議	91
其他資源	38
為SQL伺服器工作負載選取正確的EC2執行個體	100
概觀	100
成本比較	100
成本最佳化案例	101
成本最佳化建議	102
其他資源	106
合併執行個體	106
概觀	106
成本最佳化案例	107
成本最佳化建議	108
其他資源	108
比較SQL伺服器版本	109
概觀	109
成本影響	109
成本最佳化建議	111
其他資源	115
評估SQL伺服器開發人員版本	116
概觀	116
成本影響	116
其他資源	38
在 Linux 上評估SQL伺服器	119
概觀	119
成本影響	120
成本最佳化建議	121
其他資源	122
最佳化SQL伺服器備份策略	122
概觀	122
使用VSS啟用的快照進行伺服器層級備份	123
SQL 使用的伺服器備份 AWS Backup	124

資料庫層級備份	126
成本最佳化建議	134
其他資源	136
現代化SQL伺服器資料庫	136
概觀	136
資料庫方案	137
Amazon RDS和 Aurora 比較	137
成本最佳化建議	139
其他資源	143
最佳化SQL伺服器儲存	143
概觀	143
SSD Amazon 的儲存類型、效能和成本 EBS	144
Amazon 的一般SSD成本最佳化 EBS	145
其他資源	147
使用 Compute Optimizer 最佳化SQL伺服器授權	147
概觀	147
成本最佳化建議	148
設定 Compute Optimizer	148
其他資源	150
使用 Compute Optimizer 最佳化SQL伺服器大小	150
概觀	150
設定 Compute Optimizer	150
其他資源	151
檢閱SQL伺服器工作負載 Trusted Advisor 的建議	151
概觀	151
成本最佳化建議	152
設定 Trusted Advisor	152
其他資源	153
容器	154
將視窗應用程式移至容器	154
概觀	154
成本效益	155
成本最佳化建議	156
後續步驟	159
其他資源	159
優化 Amazon ECS 上 AWS Fargate 任務的成本	159

概觀	159
成本效益	160
成本最佳化建議	160
後續步驟	165
其他資源	165
掌握您的 Amazon EKS 成本	166
概觀	166
成本效益	166
成本最佳化建議	166
後續步驟	169
其他資源	170
重新平台視窗應用程式與應用程式容器	170
概觀	170
成本效益	171
成本最佳化建議	171
後續步驟	171
其他資源	171
儲存	173
Amazon EBS	173
將 Amazon EBS 卷從 gp2 遷移到 gp3	173
修改 Amazon EBS 快照	177
刪除未連接的 Amazon 卷 EBS	179
Amazon FSx	182
選擇正確的SMB檔案儲存空間	182
在 Amazon 中啟用重複資料刪除功能 FSx	186
瞭解FSx適用於 Windows 檔案伺服器的資料分片	188
了解 Amazon 中的HDD大量使用 FSx	192
使用單一可用區域	194
AWS Storage Gateway	196
Amazon S3 檔案閘道	196
Amazon FSx 檔案閘道	196
成本影響	196
成本最佳化建議	198
其他資源	201
Active Directory	202
Amazon EC2 上的自我管理活動目錄	202

概觀	202
成本影響	202
成本最佳化建議	203
其他資源	207
AWS Managed Microsoft AD	207
概觀	207
成本影響	207
成本最佳化建議	207
其他資源	208
AD Connector	209
概觀	209
成本影響	209
成本最佳化建議	209
其他資源	210
.NET	211
重構現代。NET並移動到	212
概觀	212
成本影響	212
成本最佳化建議	213
其他考量和資源	213
容器化。NET应用	214
概觀	214
成本影響	214
成本最佳化建議	215
其他資源	217
使用重力彈執行個體和容器	218
概觀	218
成本影響	218
成本最佳化建議	220
其他資源	220
Support 靜態的動態縮放。NET框架應用	221
概觀	221
成本影響	225
成本最佳化建議	226
其他資源	227
使用快取來減少資料庫需求	227

概觀	227
成本影響	227
成本最佳化建議	228
其他資源	233
考慮使用無伺服器。 NET	234
概觀	234
成本影響	234
成本最佳化建議	234
其他資源	237
考慮專門建置的資料庫	238
概觀	238
成本影響	240
成本最佳化建議	242
其他資源	244
後續步驟	245
文件歷史紀錄	246
詞彙表	247
#	247
A	247
B	250
C	251
D	254
E	257
F	259
G	260
H	260
I	261
L	263
M	264
O	268
P	270
Q	272
R	272
S	275
T	277
U	279

V	279
W	279
Z	280
.....	cclxxxi

最佳化 Microsoft 工作負載的成本 AWS

比爾·菲佛，大通·林德曼和凱文·索肯，亞 Amazon Web Services () AWS

2024 年 6 月 ([文件歷史記錄](#))

概觀

本指南提供建議、最佳做法和策略，以協助您將 Microsoft 工作負載的成本最佳化 AWS。本指南還包括基礎 AWS 知識、成本最佳化技術和參考架構，可協助您建置並自動化符合業務目標的具成本效益、高效能的工作負載。本指引統稱為 Microsoft 關於 AWS 成本最佳化 (MACO)。MACO 指引由業界專家開發，並以實際情境為基礎。

本指南涵蓋下列 Microsoft 工作負載：

- Amazon Elastic Compute Cloud (Amazon EC2) 上的窗口
- SQL Server
- 容器
- 儲存
- Active Directory
- .NET

物件


本指南適用於建築師、工程師、行政人員、董事、首席技術官、技術決策者和 AWS 合作夥伴。這很有幫助，但沒有必要具備先前的經驗，並且對 AWS 計費、Microsoft 技術和 AWS 系統管理有基本的了解。

本指南的使用方式


您可以使用本指南來規劃和實施 MACO 雲端之旅。我們建議您從頭到尾閱讀本指南，以全面了解最佳化 Microsoft 工作負載成本的選項和方法 AWS。您可以根據組織的需求檢閱下列工作負載區段：

- [Amazon EC2 上的窗口](#)
- [SQL Server](#)

- [容器](#)
- [儲存](#)
- [Active Directory](#)
- [.NET](#)

 Important

本指南中提供的程式碼範例僅用於示範目的。在生產環境中使用它之前，最好先在開發環境中測試所有代碼。在您實作任何程式碼之前，建議您先小批次測試程式碼，然後使用 [AWS Cost Explorer](#)。這可協助您疑難排解邊緣案例，以及稍後可能造成問題的其他問題。

 Important

本指南中的定價範例是根據發佈時的價格而定。價格可能變動。此外，您的成本可能會因您的 AWS 區域、AWS 服務配額以及與雲端環境相關的其他因素而有所不同。

目標業務成果

本指南可協助您和您的組織達成下列業務成果：

- 了解如何使用最佳 AWS 化與授權評估 (AWS OLA)，根據資源使用率、第三方授權和應用程式相依性，評估和最佳化您目前的內部部署和雲端環境。
- 使用適用於 Microsoft 工作負載的 AWS 現代化計算器，開發成本最佳化的商業案例。
- 優化特定 Microsoft 工作負載的成本，包括亞馬遜彈性運算雲端 (Amazon EC2) 上的 Windows 工作負載、SQL 伺服器、容器、儲存、活動目錄和 .NET 的工作負載。

成本優化旅程

雲端移轉旅程的範圍、時間和特定路徑取決於您的業務目標、技術需求和其他因素。本節提供雲端移轉旅程的範例，重點是[雲端財務管理](#)，[AWS並遵守 MACO 建議和最佳做法](#)。您可以使用此範例來瞭解如何為 Microsoft 工作負載設計雲端移轉旅程。

下列高階工作說明組織可採取的方法來實作 MACO 建議和最佳作法：

- 建立標記策略並啟用使用者定義的成本配置標籤。如需詳細資訊，請參閱標記 [AWS 資源的 AWS 白皮書最佳做法](#)。
- 根據應用模組、團隊或部門來定義預算。如需詳細資訊，請參閱 [AWS 帳單與成本管理使用者指南 AWS Budgets 中的使用管理成本](#)。
- 執行 AWS 最佳化與授權評估 (AWS OLA) 以加速節省成本。如需詳細資訊，請參閱 AWS 文件中的 [AWS 最佳化和授權評估](#)。
- 為視窗和 SQL 伺服器工作負載使用您自己的授權 (BYOL)，使用 Amazon Elastic Compute Cloud Dedicated Hosts 如需詳細資訊，請參閱本指南的 [為 Windows 和 SQL 伺服器工作負載帶來授權](#) 一節。
- 最佳化您的 SQL 伺服器授權 AWS。如需詳細資訊，請參閱本指南的 [了解 SQL 伺服器授權](#) 一節。
- 為 Windows 工作負載選取正確的執行個體類型。如需詳細資訊，請參閱本指南的 [為 Windows 工作負載選取正確的執行個體類型](#) 一節。
- 為 SQL 工作負載選取正確的執行個體類型。如需詳細資訊，請參閱本指南的 [為 SQL 伺服器工作負載選取正確的 EC2 執行個體](#) 一節。
- 將 Amazon Elastic Block Store (Amazon EBS) 從 gp2 遷移到 gp3。如需詳細資訊，請參閱本指南的 [將 Amazon EBS 卷從 gp2 遷移到 gp3](#) 一節。
- 開啟 EC2 執行個體排程器控制工作負載 AWS。如需詳細資訊，請參閱本指南的 [自動化停止和啟動排程](#) 一節。
- 使用 SQL Server 開發人員版來移除非生產工作負載的 SQL Server 成本。如需詳細資訊，請參閱本指南的 [評估 SQL 伺服器開發人員版本](#) 一節。
- 使用適用於 Windows 檔案伺服器的 Amazon FSx 單一可用區域來開發和測試工作負載。如需詳細資訊，請參閱本指南的 [使用單一可用區域](#) 一節。
- 使 AWS Compute Optimizer 用正確調整您的 Windows 工作負載大小。如需詳細資訊，請參閱本指南的 [大小適中的 Windows 工作負載](#) 一節。
- 使用 Savings Plans，在 Amazon EC2 上優化 Windows 上的支出。如需詳細資訊，請參閱本指南的 [最佳化 Windows on Amazon 的支出 EC2](#) 一節。

- 在 FSx 上啟用 Windows 檔案伺服器的重複資料刪除功能。如需詳細資訊，請參閱本指南的 [在 Amazon 中啟用重複資料刪除功能 FSx](#) 一節。
- 針對 FSx for Windows File Server 上的檔案系統使用資料分割。如需詳細資訊，請參閱本指南的 [瞭解 FSx 適用於 Windows 檔案伺服器的資料分片](#) 一節。
- 最佳化您的 SQL 伺服器備份策略。如需詳細資訊，請參閱本指南的 [最佳化 SQL 伺服器備份策略](#) 一節。
- 使靜態 .NET 框架應用程式支持動態縮放。如需詳細資訊，請參閱 [Support 靜態的動態縮放。NET 框架應用](#) 本指南中的。
- 使用無伺服器 .NET 微服務。如需詳細資訊，請參閱本指南的 [考慮使用無伺服器。NET](#) 一節。
- 將您的 Windows 應用程式移至容器。如需詳細資訊，請參閱本指南的 [容器化。NET 應用](#) 一節。
- 用 [AWS Compute Optimizer](#) 於 AWS Fargate 適當的大小適用於 Amazon Elastic Container Service (Amazon ECS) 上運行的 Windows 容器。如需詳細資訊，請參閱本指南的 [啟用運算最佳化工具](#) 一節。
- 重構為現代 .NET 並轉移到 Linux。如需詳細資訊，請參閱本指南的 [重構現代。NET 並移動到](#) 一節。
- 運用重力子執行個體和容器。如需詳細資訊，請參閱本指南的 [使用重力彈執行個體和容器](#) 一節。
- 將 SQL 伺服器資料庫現代化。如需詳細資訊，請參閱本指南的 [現代化 SQL 伺服器資料庫](#) 一節。
- 設計活動目錄基礎結構。如需詳細資訊，請參閱本指南的 [Active Directory](#) 一節。

如需有關專注於雲端財務管理的客戶旅程的詳細資訊 AWS，請參閱 AWS 白皮書 [雲端財務管理功能](#)。

最佳化成本的主要建議

概觀

成本最佳化是 [AWS Well-Architected 架構](#) 的其中一個支柱，它在您的雲端移轉計劃中扮演著至關重要的角色。您可以在本指南中找到有關成本最佳化的建議，但本節提出了影響最大的建議。您可以快速實作這些建議，這些建議會對您的組織產生重大影響。這些建議可協助您為整個成本最佳化工作奠定基礎。

熱門推薦

下表列出影響最高的成本最佳化的主要建議。「難以實現」列根據最容易實現的範圍 (1) 到最難實現的範圍 (5) 對每個優化進行評分。「預估節省成本」欄會根據百分比顯示您的組織針對每個建議的最佳化可節省多少成本的預估值。

最佳化	難以實施	估計節省
大小適中的 Windows 工作負載	3	25%
為 Windows 和 SQL 伺服器工作負載帶來授權	3	30%
評估 SQL 伺服器開發人員版本	2	20%
了解 SQL 伺服器授權	2	最高可達 50%
自動化停止和啟動排程	3	最高可達 40%
為 Windows 工作負載選取正確的執行個體類型	1	十 — 30%
重構現代。NET 並移動到	5	十至二成
最佳化 Windows on Amazon 的支出 EC2	3	最高可達二十至四成
將 Amazon EBS 卷從 gp2 遷移到 gp3	4	最高可達 20%

⚠ Important

上表預估節省的費用適用於每個個別的技术領域，而不是帳戶內的整體 AWS 支出。例如，您可以在各種環境類型和大小中實作執行個體排程器，以改變可能節省的成本。這些估算特別適用於 Amazon EC2 執行個體成本，並不意味著其他 AWS 服務執行個體可以節省任何整體成本。這些估計是作為衡量標準提供的，而不是保證。

MACO 專家可以更深入地討論成本優化。若要設定會議以深入了解您的使用案例，請連絡您的客戶團隊或寄送電子郵件至 optimize-microsoft@amazon.com。

AWS 最佳化與授權評估

概觀

最佳 [AWS 化與授權評估 \(AWS OLA\)](#) 可協助您根據資源使用率、第三方授權和應用程式相依性，評估和最佳化目前的內部部署和現有雲端環境。您可以使用 AWS OLA 來協助您的組織建置移轉和授權策略，以便在 AWS 您移轉至 AWS 或評估現有 Microsoft 工作負載時節省成本。O AWS LA 還可以幫助您實現以下目標：

- 瞭解現有的部署、應用程式效能和合約。
- 正確的大小您的資源。
- 制定 AWS 雲端。
- 使用現有投資並僅按您使用的部分付費，以降低或消除成本。

我們建議您將 O AWS LA 設為 [成本優化旅程](#) 的第一步。您可以使用 AWS Partner Network 來完成 O AWS LA。他們將協助您收集評估資料，並為您提供最佳化授權和執行個體成本的建議。

下圖提供評估程序的概觀。



評估選項

您可以從下列兩個 AWS OLA 選項中選擇適用於您的 Microsoft 工作負載 AWS：

- 精簡版 — 在此使用案例中，您的所有工作負載都在 VMware 上。您可以提 AWS 供 [RVT](#) ools 的輸出。然後，AWS 可以提供 1-5 天的周轉時間。此方法使用直接從 VMware vCenter 擷取的 point-in-time 資訊來開發大小建議，並提供隨需定價選項。

- 完整版 — 在此使用案例中，您會在不同的雲端提供者、實體伺服器 and 虛擬伺服器中執行混合環境。AWS 使用作業系統代理程式收集 14 到 30 天的使用情況資料。這可讓您根據 AWS 應用程式的使用模式，做出明智的執行個體大小決策。AWS 使用多種協力廠商工具 (例如 Cloudamize) 來完成分析。AWS 與其合作 AWS Partner Network 以協助提供最終的總體擁有成本 (TCO) 評估，並提供多種定價選項，這些選項包括定價模式和不同架構。

全面評估

完整的 AWS OLA 評估將通過一小時的電話開始。在此通話期間，可 AWS 協助您判斷最佳的 AWS 基礎結構以支援您的移轉、選擇資料收集方法，以及建立完成時間表。在組織中實作探索工具取決於資料收集方法、組織的大小，以及組織用來管理其伺服器叢集的工具。收集使用情況資料通常需要兩週時間。

完整的 AWS OLA 過程需要 30-45 天，包括以下階段：

- 範圍工作負
- 收集資料
- 分析資料
- 規劃後續步驟

範圍工作負

首先，AWS 與您和您的團隊合作確定評估範圍。這通常會依環境類型劃分 (例如，非生產和生產)。範圍包括工作負載的位置。這可能是您要移轉到的工作負載 AWS、已在其上執行的工作負載 AWS (例如，Amazon EC2 的 AWS OLA)，或是在其他雲端供應商中執行的工作負載。

收集資料

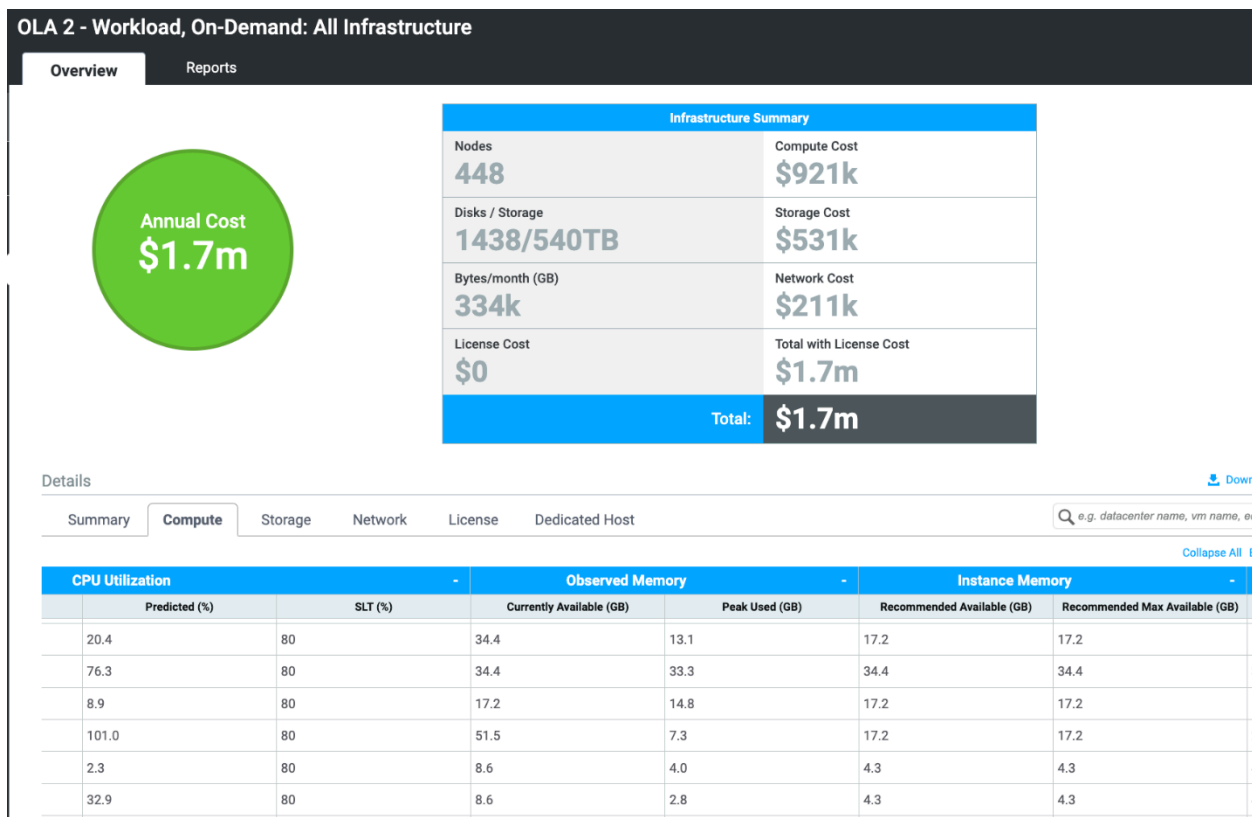
接下來，AWS 部署工具以協助探索資源，並從伺服器收集效能資料。此工具提供四個部署選項：

- 可以查詢虛擬機器管理程序的工具 (僅需要 VMware vCenter 或超 V 認證)
- 可部署在實體或虛擬機器上的代理程式
- 根據您的環境和作業系統，使用 SSH、視窗遠端管理 (WinRM) 或視窗管理檢測 (WMI) 進行無代理程式探索
- 平面檔案資料收集與分析

對於工具部署，您可以混合搭配每個選項並合併結果。確保您選擇的任何選項都不會壓縮 IT 資源是至關重要的。AWS 努力使評估過程盡可能統包。AWS OLA 團隊和 Microsoft 專業解決方案架構師除了簡短的電話通話以外，還將準備總體擁有成本 (TCO) 分析和建議以供檢閱。

分析 CPU 使用率、RAM 使用率、儲存輸送量、IOPS 和網路輸送量時，資料收集通常需要兩到三週的時間。理想情況下，此集合發生在您的業務月份的高峰時段 (例如，在 end-of-month 財務報告期間)。AWS 想要捕捉尖峰使用情況，因為這為適當大小的實例提供了良好的統計示 AWS 例，同時仍然保證性能可以超過內部部署的可用性能。AWS 將使用率指標與不同處理器世代的效能啟發式法合併，以準確鎖定特定工作負載所需的 CPU 和 RAM 數量。這些目標通常小於內部部署分配的目標。這不僅可以降低執行個體大小的運算成本，還可以最佳化授權成本。

下列儀表板檢視顯示可透過評估擷取的基礎架構成本範例。

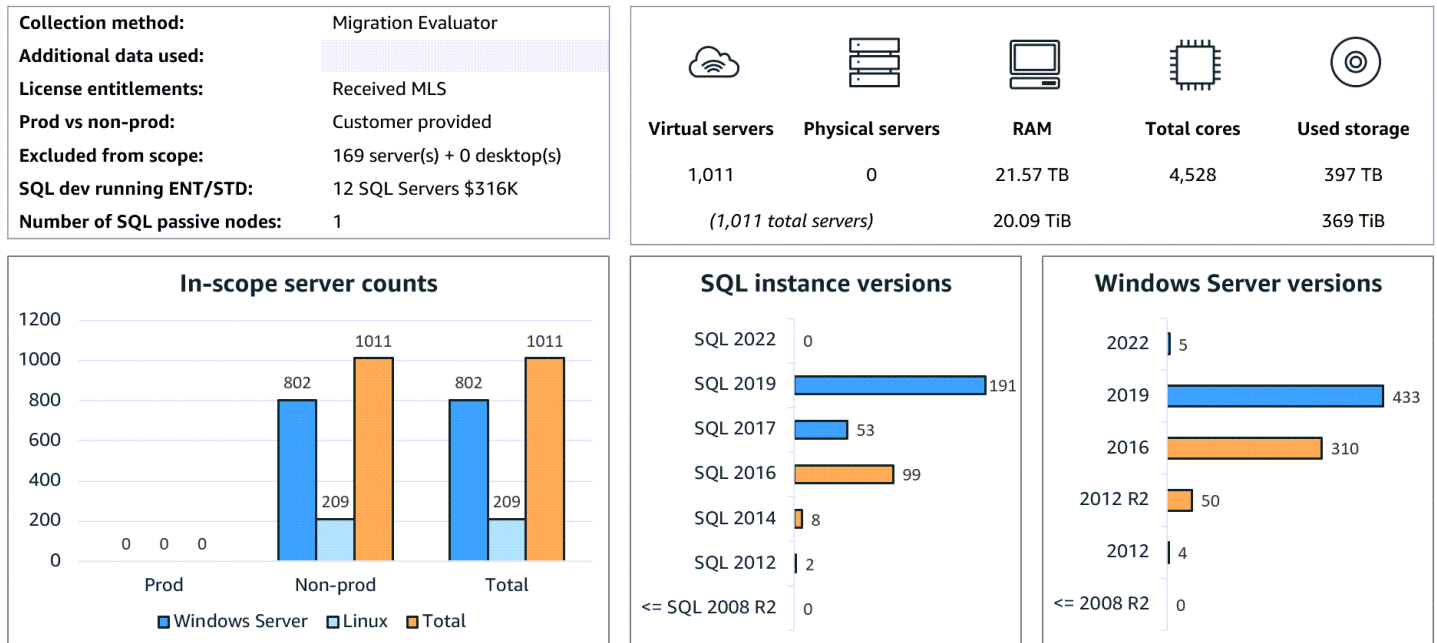


分析資料

AWS 在資料收集完成後提供一份報告簡報。AWS 檢閱資料、摘要發現結果，然後針對內部部署使用和雲端移轉提出建議。您可以透過檢查合併機會、彈性增益 (工作負載可以關閉或季節性調整)、正確 SKU 商機 (例如，SQL Server 企業版正在使用中，但資源需求和功能使用建議 SQL Server 標準版適用) 來降低運算和授權成本。對於核心授權的 SQL Server 等產品，將工作負載放在更昂貴的運算執行

個體中通常具有經濟意義。也就是說，如果 CPU 設定檔和 RAM 與 vCPU 的比率達到淨效果，以減少包含授權和使用自有授權 (BYOL) 使用案例的授權核心數目。

以下顯示根據評估收集的資料進行分析的範例。



常見的最佳化案例包括識別 AWS 資源最佳化機會和協力廠商授權節省。

AWS 資源最佳化機會的範例：

- 避免為尖峰使用量過度佈建。
- 避免過度指定和使用不足的資源。
- 適當調整執行個體大小，並移轉至最新一代的 EC2 執行個體。
- 移至受管理的資料庫，節省營運成本。

第三方授權節省的範例：

- 減少執行相同工作負載所需的核數。
- 擺脫不必要的 SQL 伺服器企業版和附加元件套件。
- 刪除殭屍服務器並替換過時的硬件。
- 使用 BYOL 和包含授權的選項來減少 future 的商業合約。
- 現代化為開放原始碼和雲端原生解決方案。

規劃後續步驟

最後，AWS 使用收集的效能資料來估算特定的工作負載大小和成本。AWS 還可以彙總查看您的範圍環境並提供定量分析。這可協助您判斷最佳選項是內部部署重新整理還是移轉至 AWS。您可以使用 AWS OLA 末尾提供的 TCO 分析摘要 (如以下範例所示)，建立雲端經濟商業案例。

	Option 1: Amazon EC2 shared	Option 1a: Amazon EC2 shared + power management	Option 2: Amazon EC2 mixed	Option 2a: Amazon EC2 mixed + power management
<i>Option details: compute</i>	100% Reserved Instances (RIs)	RIs + on-demand power management	100% RIs	RIs + on-demand power management
<i>Option details: Microsoft licenses</i>	WS LI and SQL BYOL	WS LI and SQL BYOL	WS BYOL or LI+SQL BYOL	WS BYOL or LI+SQL BYOL
Compute costs ¹				
Year 1 compute cost	\$414,546	\$482,623	\$504,019	\$513,941
Year 1 vendor license included cost	\$392,858	\$244,415	\$9,804	\$4,783
	\$807,404	\$727,038	\$513,823	\$518,724
<i>Total compute savings in year 1, compared to Option 1</i>	—	10% (\$80,366)	36% (\$293,581)	36% (\$288,680)
Storage and networking costs ²				
Annual estimated storage cost	\$336,494	\$336,494	\$336,494	\$336,494
Annual estimated networking cost	\$41,455	\$41,455	\$41,455	\$41,455
	\$377,949	\$377,949	\$377,949	\$377,949
Microsoft license costs**				
WS/CIS annual Software Assurance (SA) + current SPLA/Subs cost	\$0	\$0	\$0	\$0
WS/CIS license + SA + SPLA/Subs true-up cost	\$0	\$0	\$0	\$0
SQL annual SA + current SPLA/Subs cost	\$0	\$0	\$0	\$0
SQL license SA + current SPLA/Subs true-up cost	\$0	\$0	\$0	\$0
	\$0	\$0	\$0	\$0
Total estimated costs	\$1,185,353	\$1,104,987	\$891,772	\$896,673
<i>Annual TCO savings in year 1, compared to Option 1</i>	—	7% (\$80,366)	25% (\$293,581)	24% (\$288,680)

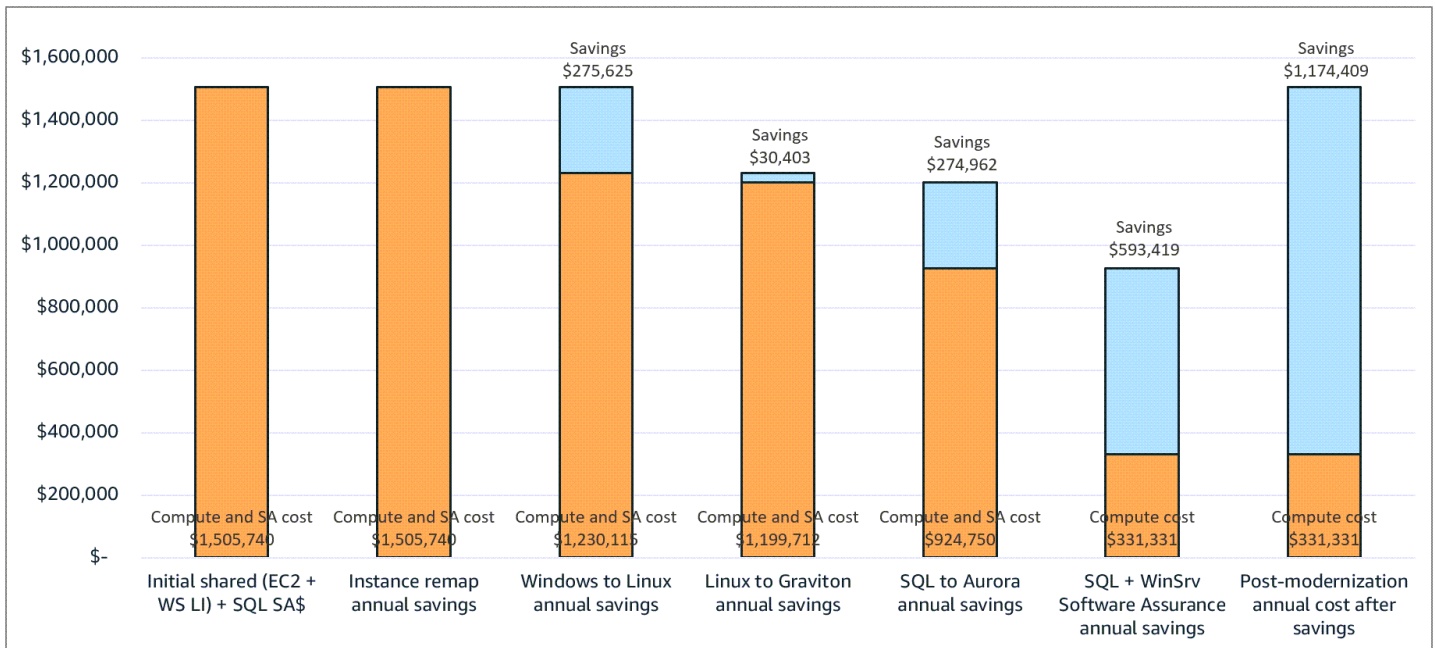
¹ Pricing model used: 3-year, no upfront RI

² Software Assurance and true-up costs provided by Microsoft

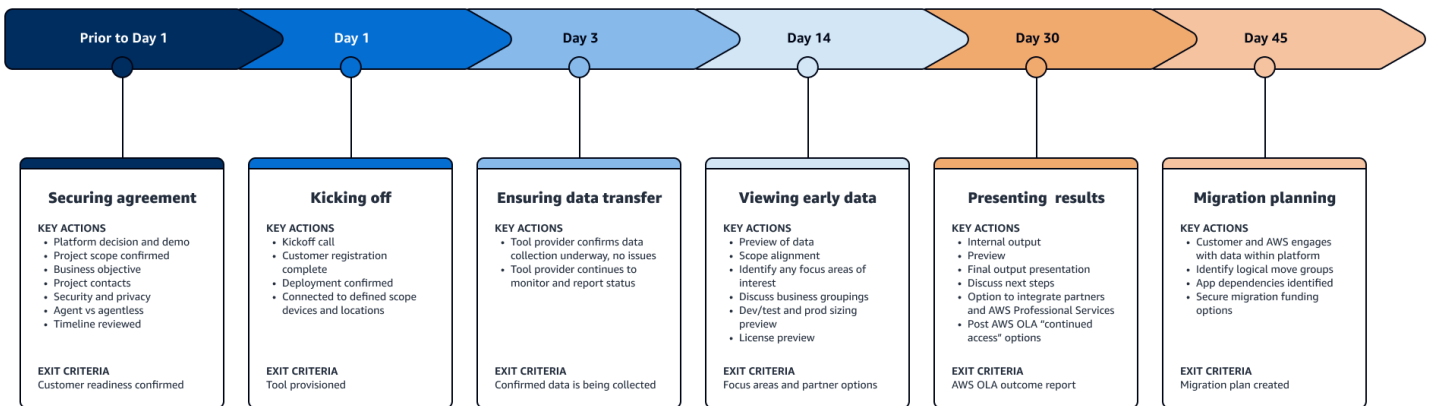
O AWS LA 也提供下列建議，深入瞭解現代化可能對您現有工作負載的影響：

- 移至 Linux 作業系統。
- 新增 ARM 處理器 (AWS 引力子) 的應用程式支援。
- 將 SQL 伺服器工作負載移至 Amazon Aurora。
- 將 Windows 和 SQL Server 工作負載移至開放原始碼技術，以消除軟體保證。

下圖顯示透過現代化技術 (例如從 Windows 移至 Linux 或從 SQL 伺服器移至 Aurora)，可達到的成本節省成本。



從開始到結束，完整的 AWS OLA 過程大約需要 45 天。下圖顯示了一個示例時間表。



如果您擁有純 VMware 環境並且可以提供 RVTtools 的輸出，則可以將此時間表縮短為一個工作週。此外，還 AWS 可以分析包含資產和使用率資料的一般檔案，例如 CPU 平均值、CPU 峰值、RAM 平均值和 RAM 峰值。

評估影響

一般客戶通常會從適當規模的工作中降低 20% 到 30% 的成本。正確的大小會根據使用情況資料，將來源工作負載與大小最佳 AWS 執行個體相符 這些適當的大小調整不僅可以減少每月的 AWS 環境成本，而且通常可以節省組織中其他地方的成本。例如，Windows 或 SQL Server 授權獲得 20-30% 的收益可以減少 Microsoft 的下一正常運作，或釋放其他應用程式的授權。line-of-business SQL Server 工作負載的合併和正確調整大小通常是實現最戲劇性的財務收益的地方。

AWS 可協助您將系統分類為現代化儲存貯體。有些系統是傳統系統，在財務上無法觸及，而其他系統可能會現代化為容器或無伺服器應用程式，在這些系統中節省了最大的成本。與您的 AWS 團隊的對話會從雲端支援的一般主題轉移到更具體的討論，討論如何以及為何應將特定工作負載現代化。AWS 還可以幫助您探索潛在的創新機會。

後續步驟

如果您正在開始針對在內部部署環境中執行的 Microsoft 工作負載進行成本最佳化之旅 AWS，請與您的 AWS 帳戶團隊聯絡並要求 O AWS LA。AWS 團隊成員可以回答您的問題，並幫助您決定 AWS OLA 是否最終是您和您的組織的正確選擇。或者，您也可以[在線申請 AWS OLA](#)。

其他資源

- [AWS 最佳化與授權評估 \(AWS 文件\)](#)
- [AWS RE: 發明 2022-如何在 AWS \(ENT205\) 上節省成本並優化 Microsoft 工作負載 \(\)](#) YouTube

Amazon 上的 Windows EC2

[Amazon Elastic Compute Cloud \(Amazon EC2 \)](#) 是高度靈活且可擴展的雲端運算平台，非常適合執行 Windows 工作負載。您可以使用 Amazon 在安全、可靠、高可用性和可調適的基礎設施上 EC2 部署、管理和擴展 Windows Server 工作負載 AWS 雲端。請考慮在 Amazon 上執行 Windows 工作負載的下列主要優點 EC2：

- 可擴展性 – Amazon EC2 可讓您輕鬆擴展 Windows 工作負載，以適應不斷變化的需求。您可以快速建立新的 EC2 執行個體來處理增加的需求，並在不再需要執行個體時輕鬆終止執行個體。您只需支付實際使用的資源。
- 彈性 – Amazon 上的 Windows EC2 支援各種執行個體類型，這些類型旨在滿足從一般用途執行個體到記憶體或運算最佳化執行個體的各種工作負載需求。此彈性可確保您可以為特定 Windows 型應用程式選擇最佳執行個體類型，將效能最大化並將成本降至最低。
- 安全性 – 為您的 Windows 工作負載 AWS 提供多層安全性，包括網路防火牆、資料加密和安全存取控制。這表示您可以信任您的應用程式和資料受到保護，同時仍可完全控制您的安全設定和組態。
- 成本效率 – pay-as-you 即用定價模型可讓您僅支付使用的資源，無需在硬體和軟體中預先投資。此模型也可讓您最佳化成本、減少資本支出，並提高營運效率。對於各種規模的企業而言，這是理想的定價模式。

本指南的本節涵蓋下列主題：

- [自動化停止和啟動排程](#)
- [大小適中的 Windows 工作負載](#)
- [為 Windows 工作負載選取正確的執行個體類型](#)
- [為 Windows 和 SQL 伺服器工作負載帶來授權](#)
- [最佳化 Windows on Amazon 的支出 EC2](#)
- [使用 AWS 工具監控成本](#)

自動化停止和啟動排程

概觀

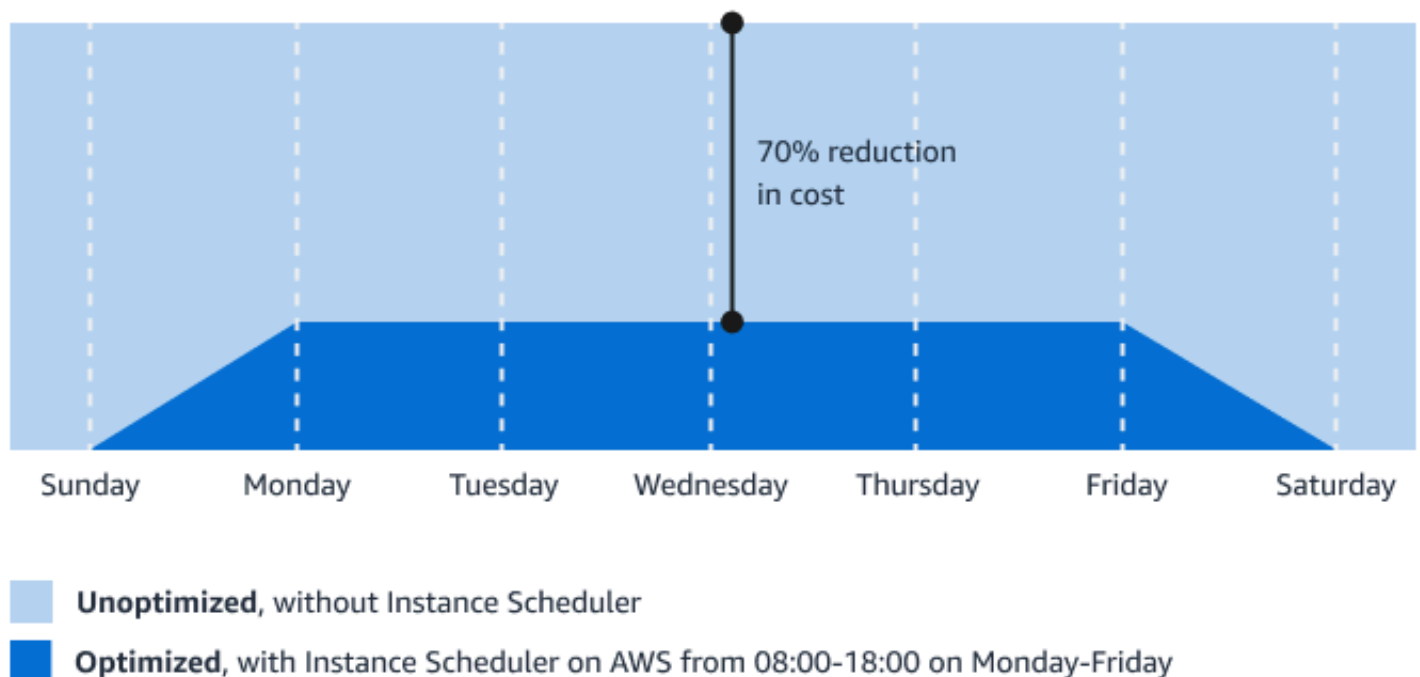
[上的執行個體排程器 AWS](#) 可以透過自動啟動和停止 [Amazon EC2](#) 和 [Amazon Relational Database Service \(AmazonRDS \)](#) 執行個體，協助您降低營運成本。如果您讓所有執行個體持續以完全使用率

執行，最終可能會支付未使用的資源。開啟執行個體排程器 AWS 可讓您在不需要執行個體的時段內關閉執行個體，例如非營業時間、週末或使用量低的其他時段。這可能會導致隨著時間的推移大幅節省成本。

上的執行個體排程器 AWS 也提供跨帳戶執行個體排程、自動標記，以及使用命令列介面或[AWS Systems Manager](#)維護時段來設定排程或期間的功能。這些功能可協助您更有效且準確地在不同專案或團隊中追蹤和分配成本，以協助您管理執行個體。

案例研究

考慮在 AWS 生產環境中使用執行個體排程器的公司範例，以每天在營業時間外自動停止執行個體。如果此公司以完全使用率保留其所有執行的執行個體，則其可以為僅在正常營業時間內必要的執行個體節省高達 70% 的成本。下表顯示每週使用率如何從 168 小時減少為 50 小時。

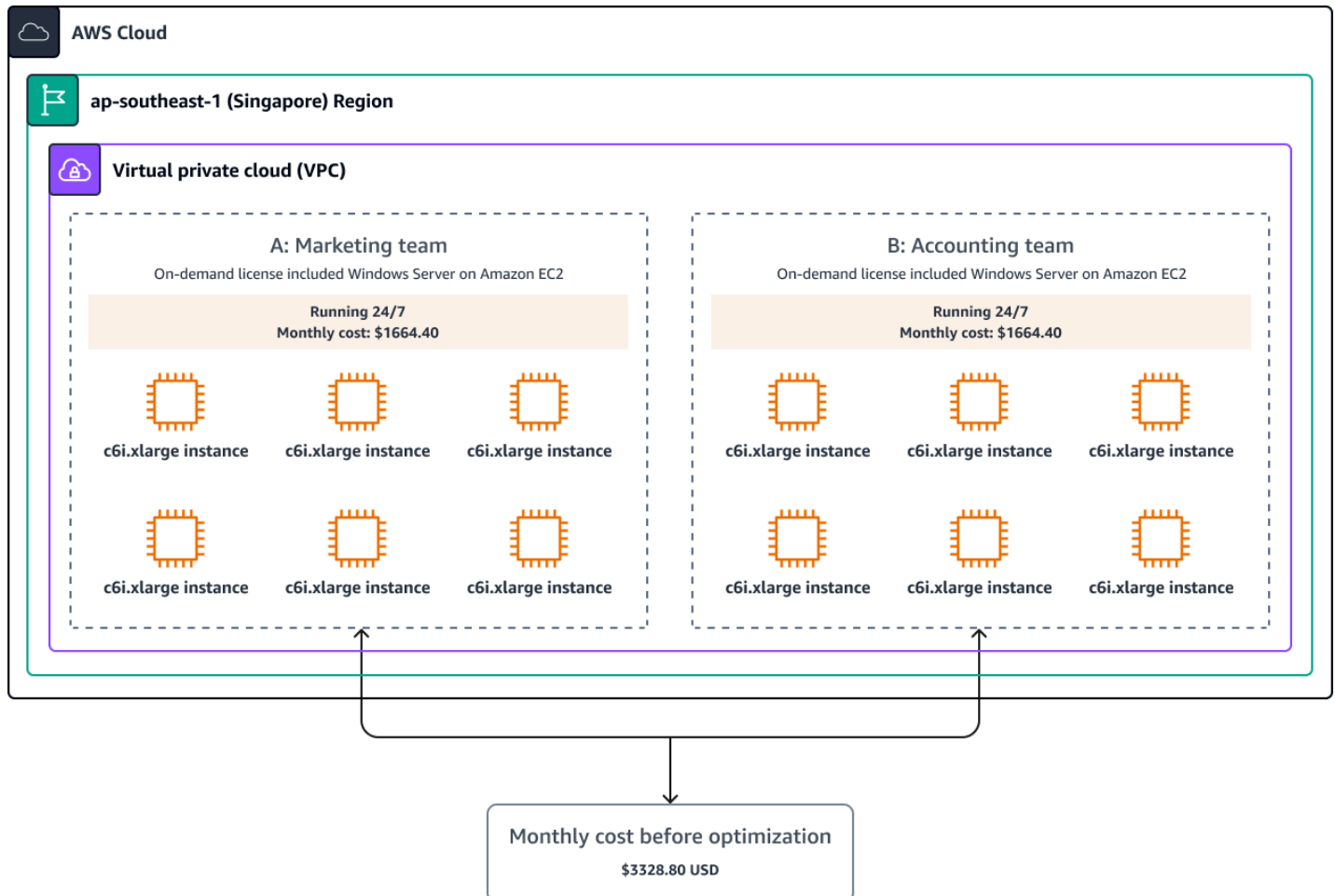


考慮另一個範例。電力公用事業公司 Jamaica Public Service Company Limited (JPS) 將其資料庫遷移至 Amazon RDS。現在， JPS 使用 Amazon EC2 託管 API 服務並執行其他應用程式。對於 JPS，上的執行個體排程器 AWS 成為管理非生產環境的關鍵工具。JPS 使用 上的執行個體排程器 AWS 來降低開發成本，並根據團隊需求和工作排程管理 EC2 執行個體。這有助於將成本 JPS 降低 40%。如需詳細資訊，請參閱 AWS 案例研究 [Jamaica Public Service 有效地遷移至雲端，使用 AWS 執行個體排程器 將成本降低 40%](#)。

成本最佳化案例

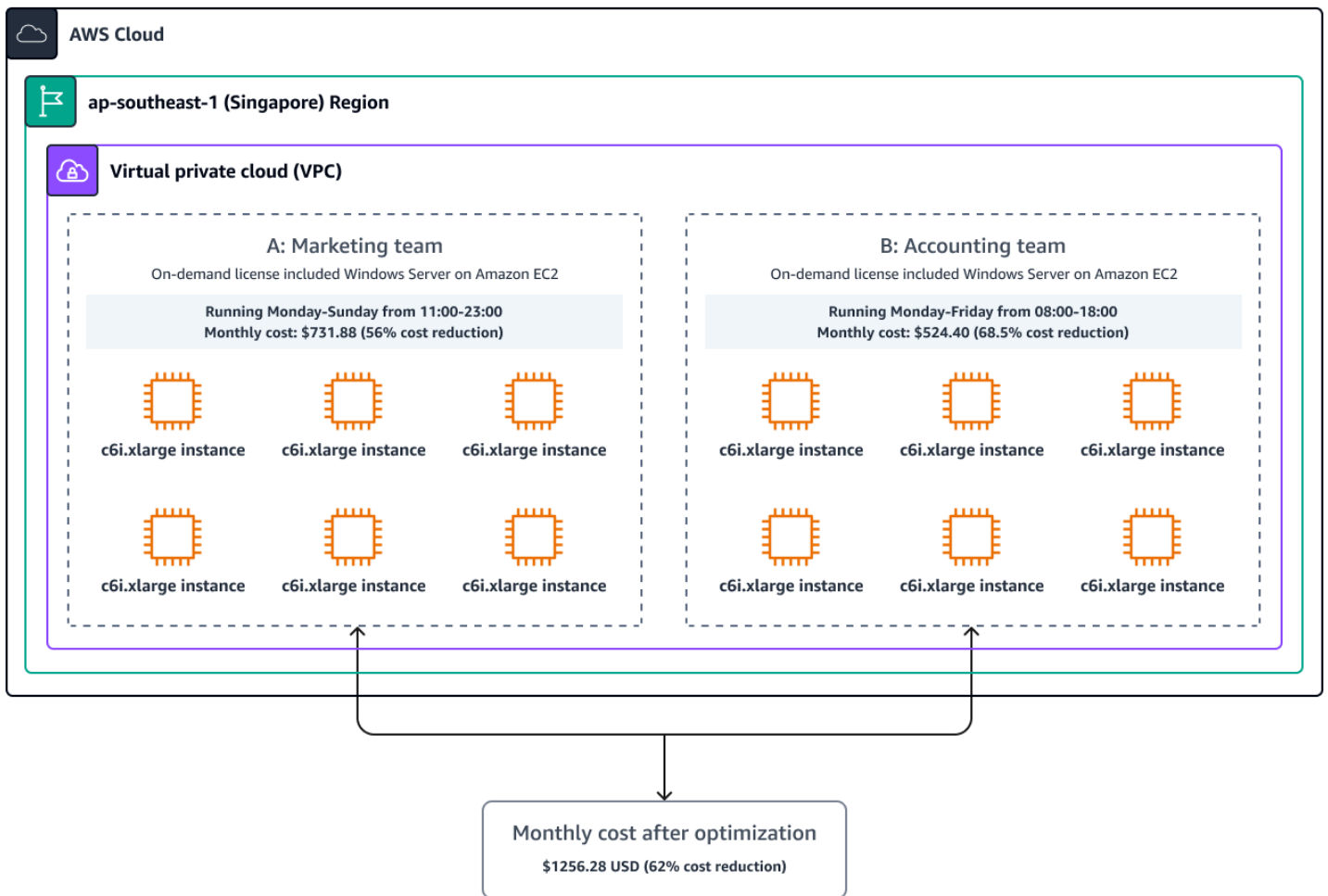
下列範例案例有助於說明在 上使用執行個體排程器的成本優勢 AWS。在此案例中，新加坡一家大型零售公司在 Amazon 上部署兩個 Windows 環境 EC2。行銷團隊會使用稱為工作負載 A 的第一個環境，在商店開啟時分析即時的店內交易。稱為工作負載 B 的第二個環境會保留給會計團隊，該團隊只會在正常營業時間內運作。考慮到目前的使用模式，這兩個環境（24/7）目前的操作排程並不理想，並且需要最佳化以降低公司的操作成本。

下圖顯示最佳化之前的每月成本。



例如，3 月有 31 天，其中 23 天是工作日。如果行銷團隊在 上使用執行個體排程器，AWS 並僅在需要時才操作其執行個體（即每個月 321 小時，而不是每月 730 小時），則他們可能會每月節省 932.52 美元。這相當於降低了 56% 的營運成本。會計團隊也會體驗到顯著的優勢，其執行個體使用時間從每月 730 小時降至 230 小時。這會導致 1,140 美元或 68.5% 的降低。公司每月可節省總計 2,072.52 美元（相當於 62% 的縮減），或每年節省 24,870.24 美元。

下圖顯示最佳化後的每月成本。



Note

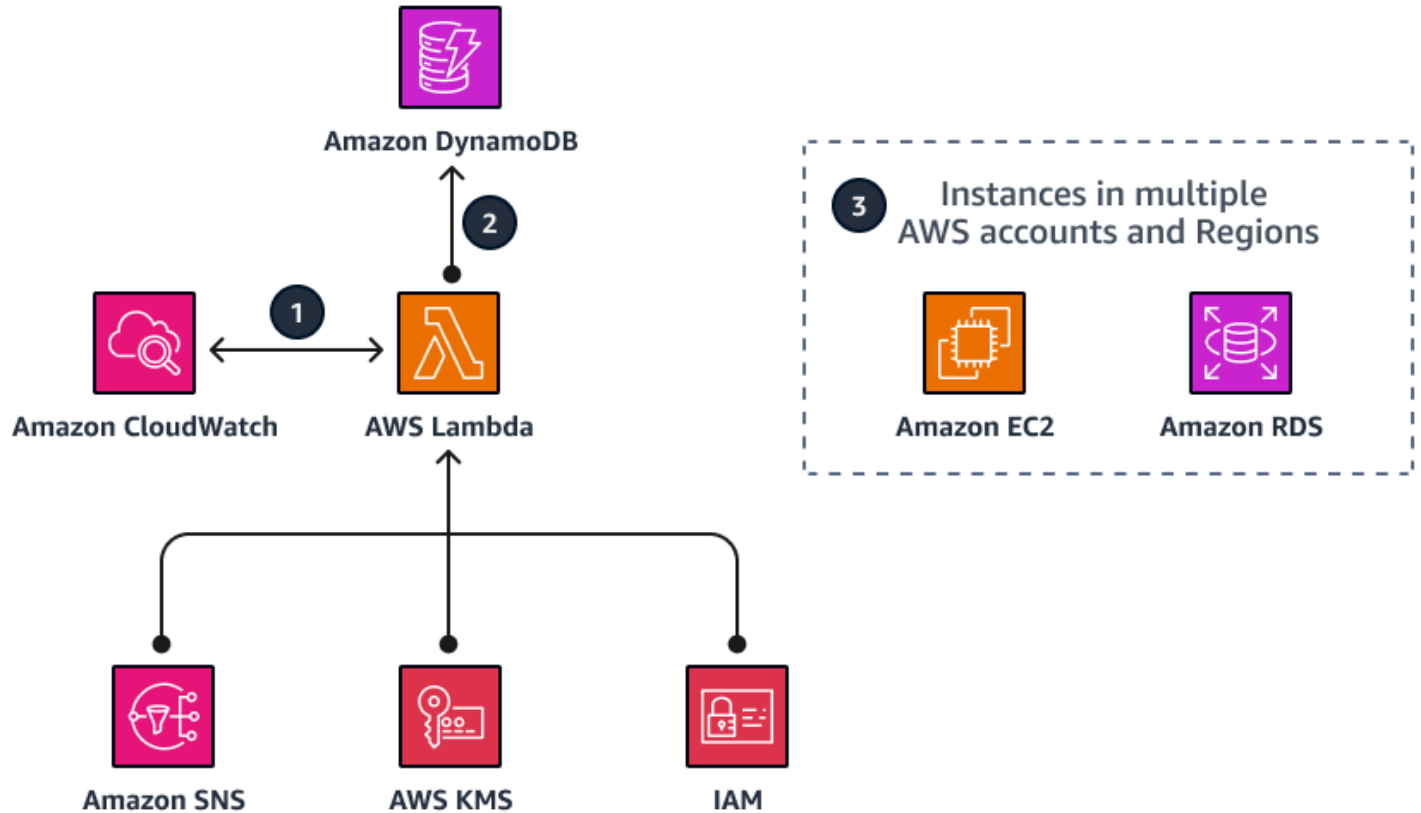
此範例的定價是使用 2023 年 3 月 [AWS Pricing Calculator](#) 的 來決定。

成本最佳化建議

本節說明如何根據 AWS 先前成本最佳化案例一節中的範例案例來部署和設定執行個體排程器。建議您採取下列步驟，使用 上的執行個體排程器來最佳化成本 AWS：

1. 啟動執行個體排程器堆疊
2. 設定期間
3. 設定排程
4. 標記 執行個體

下列架構圖顯示執行個體排程器堆疊 AWS 雲端 在 中建立的內容。



圖表顯示下列工作流程步驟：

1. AWS CloudFormation 範本會以您定義的間隔設定 Amazon CloudWatch 事件。此事件會叫用 AWS Lambda 函數。在組態期間，您可以定義 AWS 區域 和 帳戶。您也可以定義執行個體排程器的自訂標籤，AWS 用於將排程與適用的 Amazon EC2 執行個體、Amazon RDS 執行個體和叢集建立關聯。
2. 排程組態值儲存在 Amazon DynamoDB 中，Lambda 函數會在每次執行時擷取這些值。然後，您可以將自訂標籤套用至適用的執行個體。
3. 在執行個體排程器的初始組態期間，您可以定義標籤金鑰，以識別適用的 Amazon EC2 和 Amazon RDS 執行個體。建立排程時，您指定的名稱會用作標籤值，以識別您要套用至已標記資源的排程。

啟動執行個體排程器堆疊

本節說明如何在上啟動執行個體排程器的 CloudFormation 堆疊 AWS。

Note

在上執行執行個體排程器時，您必須負擔 AWS 服務使用的成本 AWS。截至 2023 年 1 月，在 us-east-1 區域中使用預設設定執行此解決方案的成本約為每月 9.90 美元的 Lambda 費用，或者如果您有 Lambda 免費方案每月用量額度，則費用會更低。如需詳細資訊，請參閱 AWS 解決方案程式庫中 [AWS 實作指南上執行個體排程器](#) 的成本一節。

若要啟動執行個體排程器堆疊，請完成下列步驟。

1. 登入 [AWS Management Console](#) 並選擇 [啟動解決方案](#)（可下載範本）以啟動 instance-scheduler-on-aws.template CloudFormation 範本。

Note

您也可以將 [下載範本](#) 作為自有實作的起點。

2. 根據預設，範本會在美國東部（維吉尼亞北部）區域啟動。若要在不同區域中啟動執行個體排程器，請使用主控台導覽列中的區域選取器。

Note

此範例使用亞太區域（新加坡）區域。

3. 在建立堆疊頁面的先決條件 - 準備範本區段中，確認已選取範本就緒選項。在範本來源區段中，確認已選取 Amazon S3 URL 選項。
4. 確認正確的範本 URL 位於 Amazon S3 URL text 方塊中，然後選擇下一步。
5. 在指定堆疊詳細資訊頁面上，為您的解決方案堆疊指派名稱。如需命名字元限制的相關資訊，請參閱 AWS Identity and Access Management (IAM) 文件中的 [IAM 和 STS 限制](#)。本指南中範例的堆疊名稱稱為 MyInstanceScheduler。

Note

堆疊名稱不能包含超過 28 個字元。

6. 在參數下，檢閱範本的參數，並視需要加以修改。
7. 選擇 Next (下一步)。在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。
8. 在檢閱頁面上，檢閱並確認設定。選取確認範本將建立 IAM 資源的方塊。

9. 選擇建立以部署堆疊。

設定期間

部署 CloudFormation 範本之後，解決方案會建立 DynamoDB 資料表，其中包含範例期間規則和排程，您可以用來做為建立自訂期間規則和排程的參考。如需期間組態範例，請參閱 [文件上執行個體排程器 AWS 中的範例排程](#)。

若要完成此案例的步驟，您必須產生對應於每個工作負載的期間，並滿足其特定需求。例如：

```
Period 1 (Workload A):  
  Name: retail-hours  
  Days: Monday to Sunday  
  Hours: 1100 - 2300  
Period 2 (Workload B):  
  Name: office-hours  
  Days: Monday to Friday  
  Hours: 0800 - 1800
```

若要設定期間，請完成下列步驟：

1. 登入 [DynamoDB 主控台](#)，並確保您位於您在 [上](#) 啟動執行個體排程器 CloudFormation 範本的相同區域 AWS。
2. 在導覽窗格中，選擇資料表，然後選擇名為 `ConfigTable` 的資料表。
3. 選擇探索資料表項目。
4. 若要建立辦公時間的時段，請選取辦公時間項目的時段。
5. 在編輯項目頁面上，將開始時間變更為 0800，結束時間變更為 1800。保留工作日的預設值。

Note

開始時間和結束時間值決定執行個體的啟動和停止時間，而工作日值則決定此排程適用於一週中的哪些天（此範例為週一至週五）。

6. 選擇 Save changes (儲存變更)。
7. 若要複製辦公時間期間並用來建立新的零售時間期間，請選取辦公時間項目的期間。然後，從動作功能表中，選擇複製項目。
8. 修改屬性以符合您的需求。下列屬性用於滿足範例案例的需求：

```
type: period
name: retail-hours
begintime: 11:00
description: Retail hours
endtime: 23:00
weekdays: mon-sun
```

9. 選擇建立項目。

10. 在 DynamoDB 中 ConfigTable，識別您剛在項目清單中建立的兩個期間。

設定排程

在上的執行個體排程器內容中 AWS，排程是指一或多個期間和相關時區的應用程式。然後，這些排程會以標籤的形式指派給您的執行個體。本節說明如何建立兩個排程（如下所示），以適應兩個範例工作負載的不同時間模式，然後將排程與您在上一節中建立的期間建立關聯。

```
Schedule 1:
  Name: singapore-office-hours
  Period: office-hours
  Timezone: Asia/Singapore
Schedule 2:
  Name: singapore-retail-hours
  Period: retail-hours
  Timezone: Asia/Singapore
```

若要建立和設定排程，請完成下列步驟：

1. 登入 [DynamoDB 主控台](#)，並確保您位於您在 上啟動執行個體排程器 CloudFormation 範本的相同區域 AWS。
2. 在導覽窗格中，選擇資料表，然後選擇名為 的資料表 ConfigTable。
3. 選擇探索資料表項目。
4. 若要複製英國辦公時間排程，並使用它來建立新的辦公時間排程（新加坡辦公時間，在此範例中），請選取 uk-office-hours 項目的排程。然後，從動作功能表中，選擇複製項目。
5. 修改屬性以符合您的需求。下列屬性用於滿足範例案例的需求：

```
type: schedule
name: singapore-office-hours
description: Office hours in Singapore
```



```
periods: office-hours  
timezone: Asia/Singapore
```

6. 選擇建立項目。
7. 重複步驟 4–6，使用下列屬性值建立新加坡零售時數的排程：

```
type: schedule  
name: singapore-retail-hours  
description: Retail hours in Singapore  
periods: retail-hours  
timezone: Asia/Singapore
```

8. 在 DynamoDB 中 ConfigTable，識別您建立的兩個排程和兩個期間。

標記 執行個體

建立排程之後，您必須使用標籤將排程配置到要使用的特定執行個體。您可以使用 [中的標籤編輯器](#) [AWS Resource Groups](#) 來產生標籤，並將標籤指派給您的 Amazon EC2 執行個體。

1. 登入 [AWS Management Console](#)，並確保您位於先前啟動 CloudFormation 範本的相同區域。
2. 開啟 [Resource Groups 主控台](#)。在導覽窗格中，展開標記，然後選擇標籤編輯器。
3. 在尋找要標記的資源區段中，針對區域，選擇您的區域。針對資源類型，選擇 Amazon EC2 或 Amazon RDS。此案例著重於工作負載 A 中的 Amazon EC2 執行個體。行銷團隊正在新加坡區域中使用工作負載 A。此工作負載的資源已標記部門金鑰和行銷值。您可以使用此標籤來搜尋執行個體。
4. 選擇搜尋資源。
5. 從搜尋結果清單中選取要包含在排程中的執行個體，然後選擇管理所選資源的標籤。
6. 在編輯所有所選資源的標籤區段中，選擇新增標籤，將執行個體排程器排程標籤新增至 EC2 執行個體。您可以使用符合排程的標籤索引鍵和值（先前在 DynamoDB 中建立）。
7. 對於標籤金鑰，新增排程。對於標籤值，輸入 singapore-retail-hours。
8. 選擇檢閱和套用標籤變更。
9. 若要將標籤套用至您選取的所有 EC2 執行個體，請選擇將變更套用至所有選取的。
10. 針對您要套用的任何其他排程重複步驟 3–9。

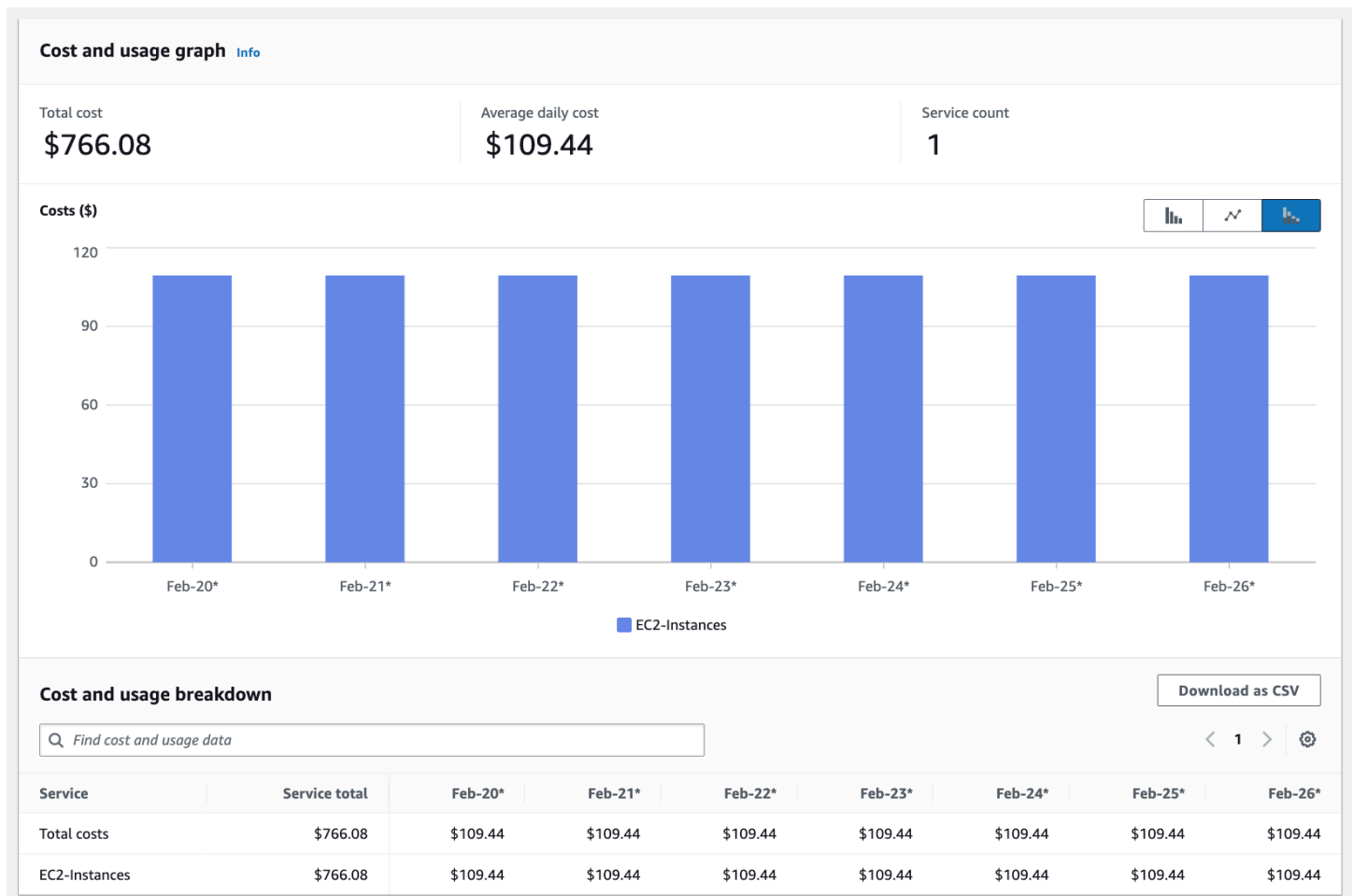
驗證結果

建議您使用 [AWS Cost Explorer](#) 來測量在 上使用執行個體排程器的成本效益 AWS。您可以使用 Cost Explorer 執行下列動作：

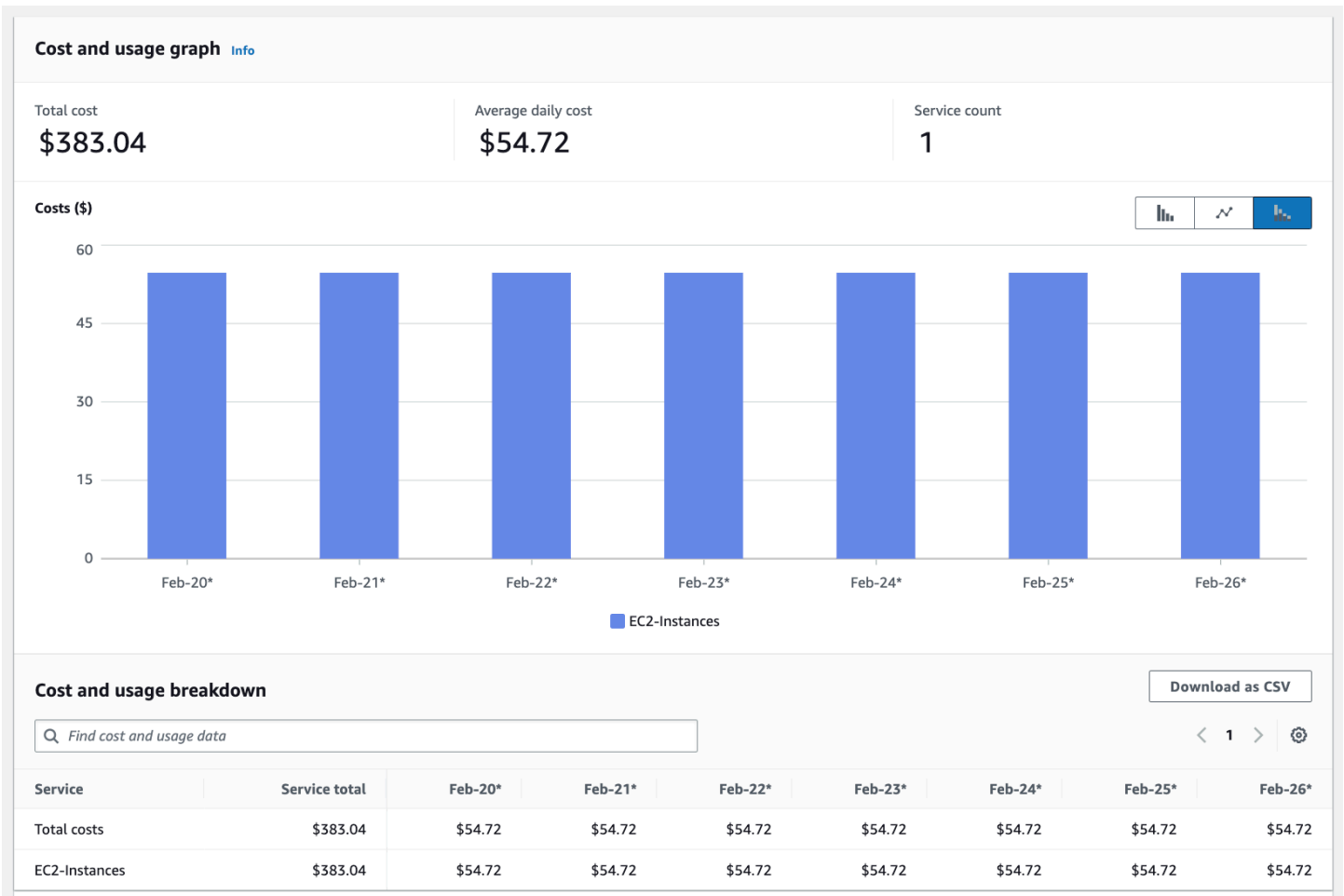
- 檢視和分析與EC2執行個體相關聯的成本，包括執行個體排程器管理的執行個體。
- 依標籤篩選您的 Cost Explorer 檢視，以便您可以專注於特定工作負載，並使用執行個體排程器取得節省成本的精細檢視。
- 深入了解使用 Instance Scheduler 的財務影響。
- 識別進一步成本最佳化的機會，並做出資料驅動的決策，以最佳化您的 AWS 支出。

下列圖表說明在最佳化前七天（週一至週日）內使用 Instance Scheduler 操作工作負載 A 和工作負載 B 的成本。

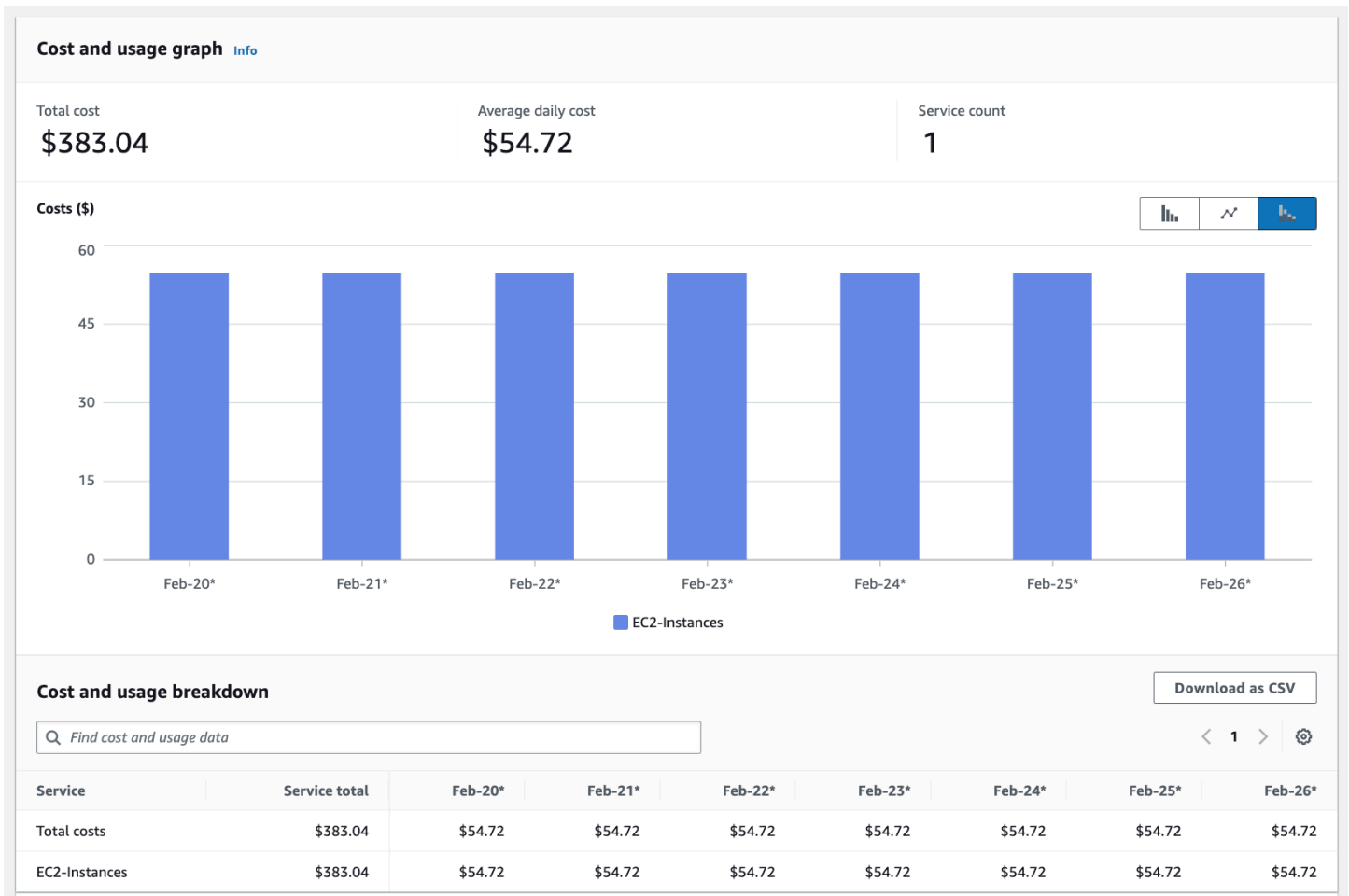
工作負載 A 和 B 的合併總費用



工作負載 A 費用

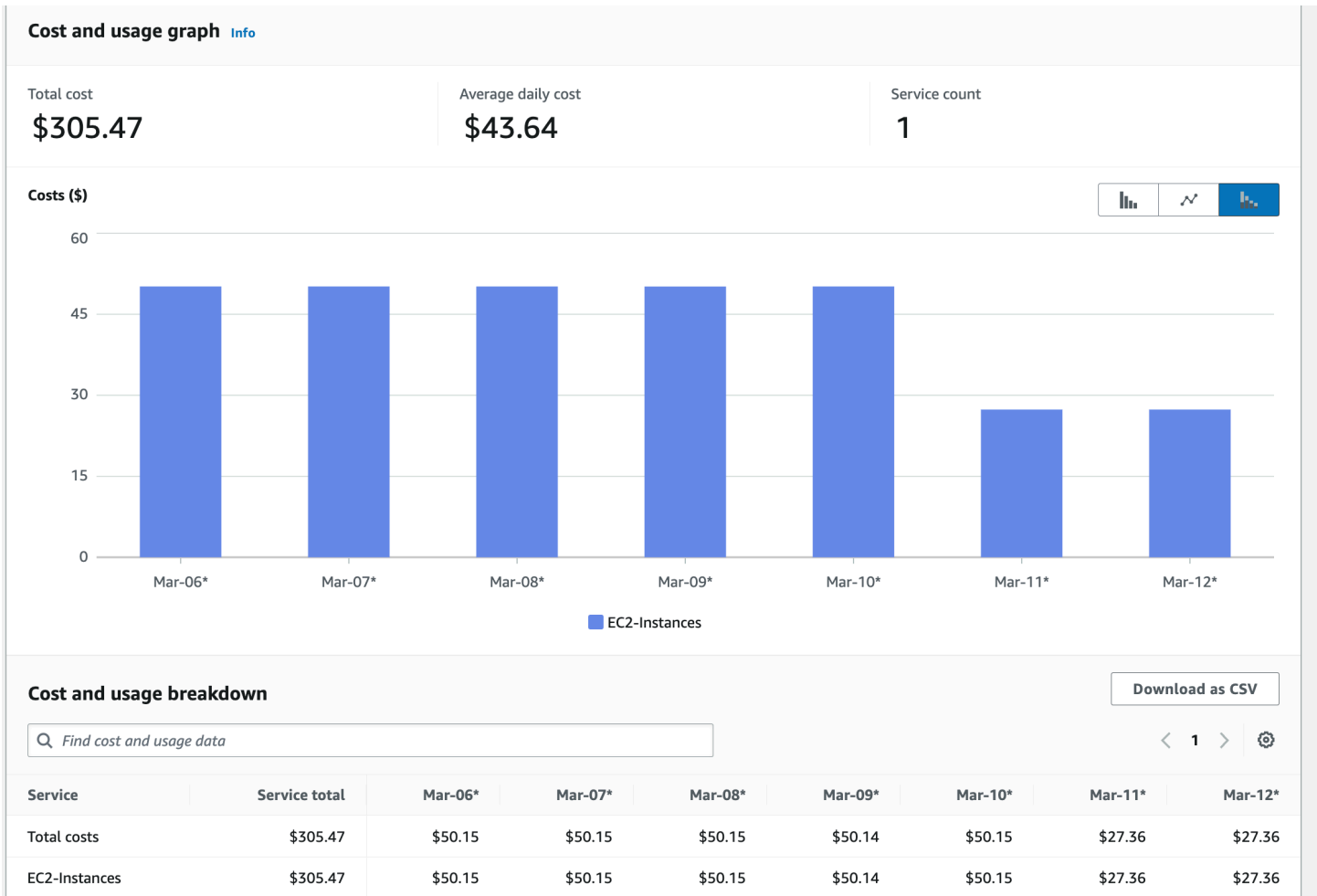


工作負載 B 費用

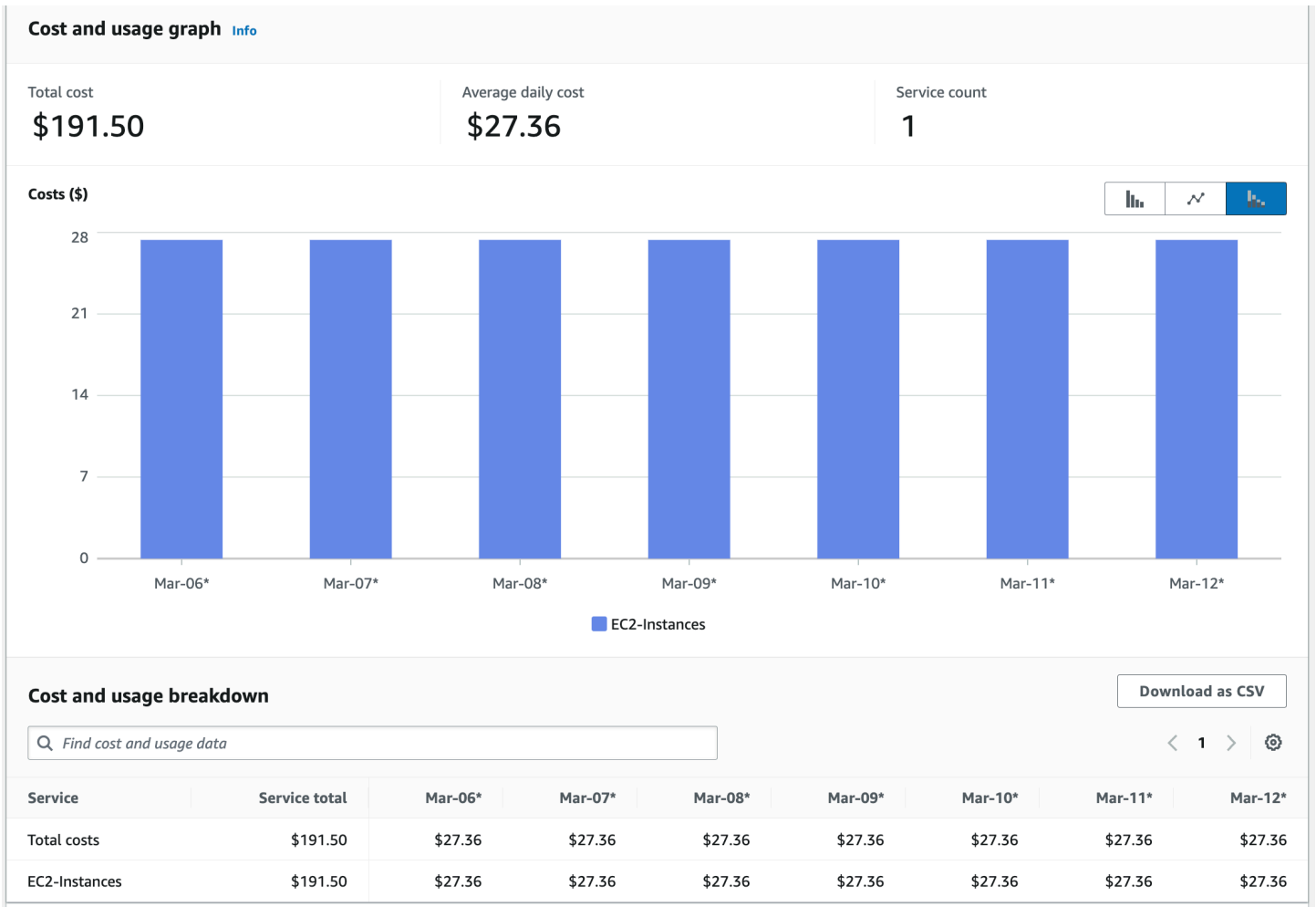


在此案例中，Cost Explorer 會顯示在 實作執行個體排程器所產生的成本降低 AWS。下表描述最佳化後七天（週一至週日）內工作負載 A 和工作負載 B 的操作成本。

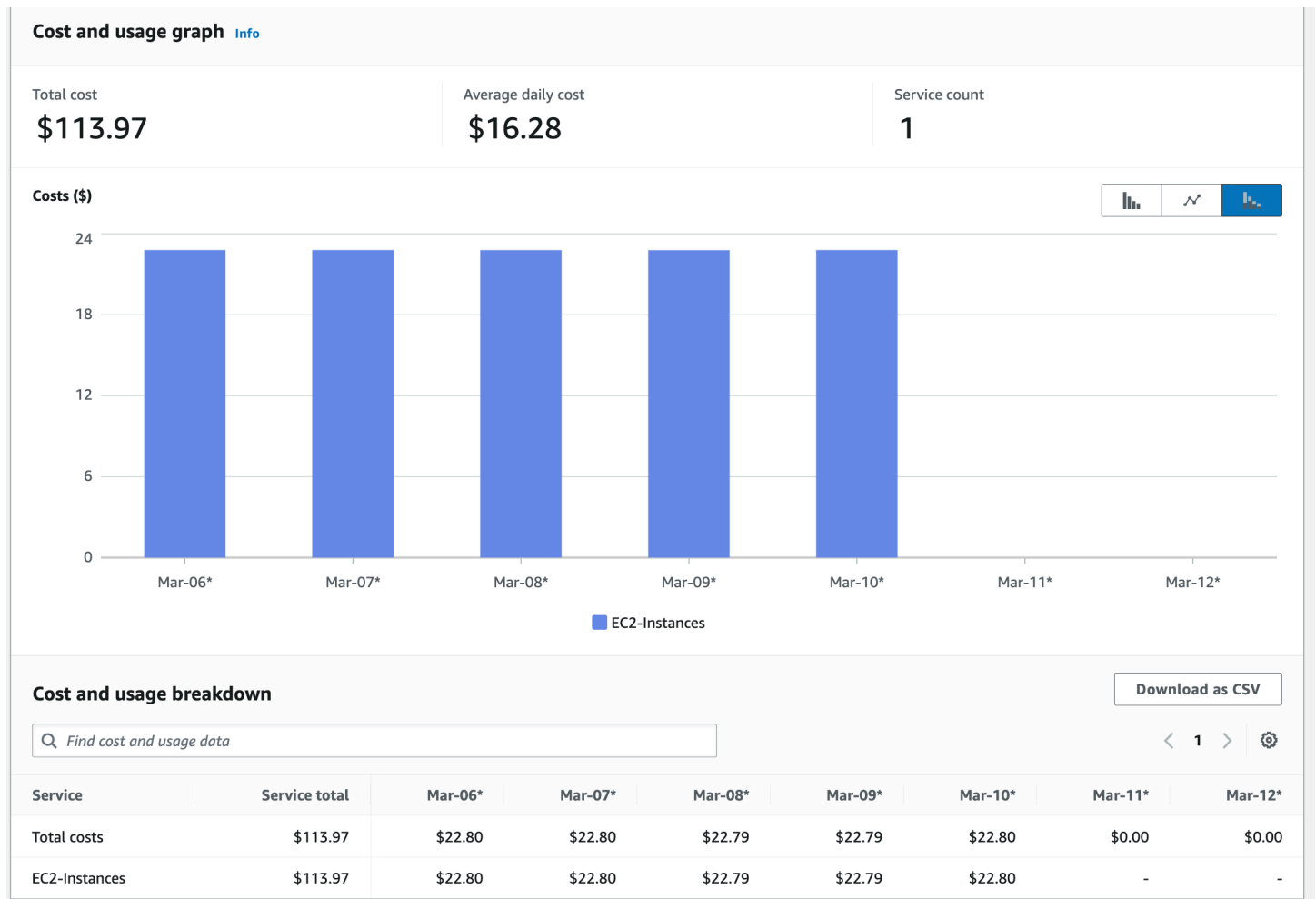
工作負載 A 和 B 的合併總費用



工作負載 A 費用



工作負載 B 費用



其他資源

- [自動化啟動和停止 AWS 執行個體](#) (AWS 文件上的執行個體排程器)
- [返回基本：使用執行個體排程器控制 Amazon EC2和 Amazon RDS Resource Costs](#) (YouTube)
- [標記 AWS 資源](#) (標記 AWS 資源使用者指南)
- [使用 \(文件 \) 分析您的成本 AWS Cost Explorer](#) AWS Billing and Cost Management

大小適中的 Windows 工作負載

概觀

正確調整大小是最有效的節省成本工具之一。AWS 提供各種方法，以收集正確的大小資訊，包括使用 [AWS 最佳化和授權評估 \(AWS OLA \)](#) 檢閱潛在的工作負載，以及使用 檢閱現有的工作負載 [AWS Cost Explorer](#)。

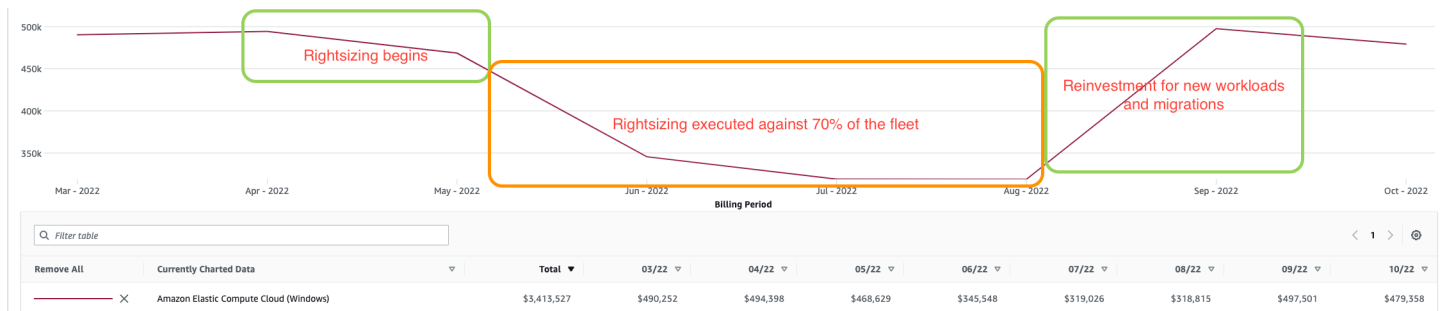
本節說明如何使用 [AWS Compute Optimizer](#) 來識別 Amazon EC2 正確的大小調整機會。Compute Optimizer 有助於防止下列類型的 AWS 資源過度佈建和佈建不足：

- [Amazon Elastic Compute Cloud \(Amazon EC2 \)](#) 執行個體類型
- [Amazon Elastic Block Store \(Amazon EBS \)](#) 磁碟區
- 上的 [Amazon Elastic Container Service \(Amazon ECS \)](#) 服務 AWS Fargate
- [AWS Lambda](#) 以 [Amazon CloudWatch](#) 提供的使用率資料為基礎的函數

成本最佳化案例

測量正確大小調整的有效性可能很困難，因為適當的大小調整工作可以導向特定的應用程式、團隊或整個組織。例如，請考慮將數千個執行個體遷移至的組織 AWS，其中 90% 的機群包含 Windows 工作負載。組織可以使用 Compute Optimizer 來分析其機群，並探索其帳戶和之間的大量過度佈建 AWS 區域。然後，他們可以使用 [AWS Systems Manager 自動化](#)，透過多個維護時段正確調整機群的大小。因此，組織會設法調整大小適中的執行個體類型，使其機群的 70%，並節省 35% 的成本。

下列儀表板說明此範例組織策略性地實作 Compute Optimizer 的正確大小調整建議時，在幾個月內達成的節省。其目標是盡可能有效率地操作其現有工作負載，以便繼續從接近其合約結束的託管資料中心進行停滯遷移。



成本最佳化建議

建議您採取下列步驟，使用 Compute Optimizer 來最佳化成本：

- 啟用運算最佳化工具
- 啟用 Windows 節點的記憶體指標集合
- 使用 Compute Optimizer 建議
- 標記執行個體以正確調整大小
- 啟用成本分配標籤，以使用 AWS 帳單工具

- 使用 AWS Systems Manager Automation 實作正確的大小調整建議
- 考慮替代的調整大小方法
- 在 Cost Explorer 中檢閱成本前後

啟用運算最佳化工具

您可以在組織或 中的單一帳戶層級啟用 [Compute Optimizer](#) AWS Organizations。整個組織的組態會針對所有成員帳戶的整個機群中新的和現有的執行個體提供持續報告。這可讓正確的大小調整成為週期性活動，point-in-time 而不是活動。

組織層級

對於大多數組織而言，使用 Compute Optimizer 最有效率的方式是在組織層級。這可提供多帳戶和多區域組織可見性，並將資料集中到一個來源以供檢閱。若要在組織層級啟用此功能，請執行下列動作：

1. 使用具有**必要許可**的角色登入您的 [Organizations 管理帳戶](#)，並選擇加入此組織中的所有帳戶。您的組織必須[啟用所有功能](#)。
2. 啟用管理帳戶後，您可以登入帳戶、查看所有其他成員帳戶，並瀏覽其建議。

Note

最佳實務是設定 Compute Optimizer [的委派管理員帳戶](#)。這可讓您執行最低權限的原則。如此一來，您就可以將對組織管理帳戶的存取降到最低，同時仍可提供整個組織的服務的存取。

單一帳戶層級

如果您以成本高的帳戶為目標，但無法存取 AWS Organizations，您仍然可以為該帳戶和區域啟用 Compute Optimizer。若要了解選擇加入程序，請參閱 Compute Optimizer 文件中的 [入門 AWS Compute Optimizer](#)。

啟用 Windows 節點的記憶體指標集合

記憶體指標為 Compute Optimizer 提供必要的基本指標，以便在組織中提出資訊充足的正確大小調整建議。這是因為在提供建議之前執行的 CPU、記憶體、網路和儲存體分析。

若要將記憶體指標從 Windows EC2 執行個體傳遞至 Compute Optimizer，您必須啟用 CloudWatch 代理程式，並設定每 60 秒收集一次記憶體指標。搭配使用記憶體指標無需額外費用 CloudWatch。

啟用 CloudWatch 代理程式並設定記憶體指標

下載 [ComputeOptimize.yml](#) 檔案。您可以使用此檔案來啟用帳戶中所有執行個體的記憶體集合。範本檔案會產生下列元件：

- [AWS Systems Manager 參數存放區](#) – 此存放收集記憶體指標所需的 CloudWatch 代理程式組態。
- AWS Identity and Access Management 附加的 (IAM) 受管政策角色 – 這是 Systems Manager Automation 文件。 [AWSAWS Systems Manager](#)
- [AWS Systems Manager 文件](#) – 這會安裝和設定代理程式 CloudWatch (取代任何現有的 CloudWatch 組態)。
- [AWS Systems Manager State Manager 關聯](#) – 這可讓 Systems Manager 文件在您帳戶中的所有執行個體上執行。

Important

執行此範本會覆寫執行個體上的任何現有 CloudWatch 組態。

接著，執行下列動作：

1. 登入 AWS Management Console 並開啟 [CloudFormation 主控台](#)。
2. 在導覽窗格中，選擇 Stacks (堆疊)。
3. 選擇建立堆疊，然後選擇使用現有資源 (匯入資源)。
4. 選擇 Next (下一步)。
5. 針對範本資源，選取上傳範本檔案。
6. 選擇檔案，然後上傳 ComputeOptimize.yml 檔案。
7. 選擇 Next (下一步)。
8. 在指定堆疊詳細資訊頁面上，針對堆疊名稱輸入堆疊的名稱，然後選擇下一步。
9. 在識別資源頁面上，輸入您要匯入之資源的識別符值。
10. 選擇匯入資源。
11. 部署堆疊之後，請選擇輸出索引標籤，以尋找您關聯的金鑰、值和描述。

監控關聯進度

1. 部署 CloudFormation 堆疊完成後，請開啟 [Systems Manager 主控台](#)。

2. 在導覽窗格中的節點管理區段中，選擇 State Manager。
3. 在關聯頁面上，選擇關聯中的關聯 ID。
4. 選擇 Execution history (執行歷程記錄) 標記。
5. 在執行 ID 欄中，選擇您關聯的執行 ID。狀態應為成功。

檢視 中的指標 CloudWatch

建議您等待至少五分鐘，讓指標填入 CloudWatch。

1. 開啟 [CloudWatch 主控台](#)。
2. 在導覽窗格中，展開指標區段，然後選擇所有指標。
3. 確認指標顯示在 CWAgent 命名空間下方。

Note

若要將設定套用至任何新執行個體，請重新執行關聯。

使用 Compute Optimizer 建議


請考慮一個範例，專注於在單一帳戶和單一區域中進行正確的大小調整。在此範例中，運算最佳化工具會在所有帳戶的組織層級啟用。請記住，正確調整大小是一個破壞性的過程，在大多數情況下，應用程式擁有人會在數週的排程維護時段中，以精確度執行。

如果您在組織的管理帳戶中導覽至 Compute Optimizer（如下列步驟所示），您可以選擇要調查的帳戶。在此範例中，在 us-east-1 區域中的單一帳戶中有六個執行的執行個體。所有六個執行個體都過度佈建。目標是根據 Compute Optimizer 的建議調整執行個體的大小。

識別過度佈建的執行個體和匯出建議詳細資訊

1. 登入 AWS Management Console 並開啟 [Compute Optimizer 主控台](#)。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在儀表板頁面上的搜尋方塊中，輸入 Region=US East (N. Virginia)。然後，輸入 調查結果=預留。這些篩選條件可讓您查看 us-east-1 區域中所有佈建過度的執行個體。
4. 若要檢閱超額佈建 EC2 執行個體的詳細建議，請向下捲動至 EC2 執行個體卡，然後選擇檢視建議。
5. 選擇匯出並儲存檔案以供日後使用。

6. 針對 S3 儲存貯體，輸入您要作為匯出檔案目的地的 Amazon S3 儲存貯體名稱。

 Note

若要儲存建議以供未來檢閱，您必須有一個 S3 儲存貯體可供 Compute Optimizer 在每個區域中寫入。如需詳細資訊，請參閱 Compute Optimizer 文件中的[適用於的 Amazon S3 儲存貯體政策 AWS Compute Optimizer](#)。

7. 在匯出篩選條件區段中，選取組織中所有成員帳戶的建議核取方塊。

8. 針對資源類型，選擇 EC2 執行個體。


9. 在要包含的資料欄區段中，選取全選核取方塊。

10. 選擇 Export (匯出)。

根據建議選擇執行個體

執行個體建議是根據 Compute Optimizer 收集和分析的效能指標。請務必注意執行個體上執行的工作負載，以確保您選擇最佳執行個體。此範例假設您可以從最新一代的 Amazon EC2 [R6i](#)、[R5](#) 和 [T3](#) 執行個體中進行選擇。T3 執行個體可爆量，網路頻寬功能較低。R5 和 R6 執行個體每小時的成本相同，幾乎完全相同。不過，R6 執行個體具有更高的網路頻寬容量、具有最新一代的 Intel 處理器，並提供與 R5 相同的運算佔用空間。在此範例中，R6 是調整大小的最佳選擇。

1. 在 [Compute Optimizer 主控台](#) 中，從導覽列選擇 EC2 執行個體的建議。此頁面顯示目前執行個體類型的比較，以及取代它的建議選項。
2. 若要取得您想要正確大小的執行個體 ID，請從 中的管理帳戶開啟 [Amazon S3 主控台](#) AWS Organizations。
3. 在導覽窗格中，選擇儲存貯體，然後選擇您用來儲存匯出結果的儲存貯體。
4. 在物件索引標籤上，從物件清單中選擇匯出檔案，然後選擇下載。
5. 若要從檔案擷取執行個體資訊，您可以使用 Microsoft Excel 中資料索引標籤上的文字對資料欄按鈕。

 Note

執行個體會以 Amazon Resource Names () IDs 表示 ARNs。請務必將分隔符號設定為 "/"，然後擷取執行個體 ID。或者，您可以撰寫指令碼或使用整合式開發環境 (IDE) 來修剪 ARN。

6. 在 Excel 中，篩選調查結果欄，以僅顯示 OVER_PROVISIONED 執行個體。這些是您要針對正確大小設定目標的執行個體。
7. 將執行個體儲存在文字編輯器IDs中，以便日後輕鬆存取。

標記執行個體以正確調整大小

標記工作負載是用於在中組織資源的強大工具 AWS。標籤可讓您詳細了解成本，並協助退款。如需將標籤新增至 AWS 資源的策略和方法的詳細資訊，請參閱 AWS 白皮書[標記 AWS 資源的最佳實務](#)。在此範例中，您可以使用 [AWS 標籤編輯器](#)，在您想要設定目標以在維護時段調整大小的超額佈建執行個體之間進行標記調整。您也可以使用此標籤來檢視變更前後的成本。

1. 登入 AWS Management Console，並開啟包含用於調整大小之執行個體的帳戶 [AWS Resource Groups 主控台](#)。
2. 在導覽列的標記區段中，選擇標籤編輯器。
3. 針對區域，選取您的目標區域。
4. 針對資源類型，選擇 AWS::EC2::Instance。
5. 選擇搜尋資源。
6. 在資源搜尋結果頁面上，選取您要調整大小的所有執行個體，然後選擇管理所選資源的標籤。
7. 選擇 Add tag (新增標籤)。
8. 針對標籤金鑰，輸入 Rightsizing。對於標籤值，輸入啟用的。然後，選擇檢閱並套用標籤變更。

Note

您可以在 Cost Explorer 中包含其他中繼資料，例如團隊或業務單位，以協助稍後進行篩選。

建立使用者定義標籤並將其套用至資源後，標籤最多可能需要 24 小時才會出現在您的成本分配標籤頁面上以啟動。在您選取標籤進行啟用之後，標籤可能需要 24 小時才會變成作用中。

對於進階使用者，您可以在目標帳戶和區域中使用 [AWS CloudShell](#) 來標記多個執行個體。例如：

```
bash
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="type-m5"
```

```
# Get a list of instance IDs
INSTANCE_IDS=$(aws ec2 describe-instances --query
  "Reservations[].Instances[].InstanceId" --output text)
# Loop through each instance ID and add the tag
for INSTANCE_ID in $INSTANCE_IDS; do
  aws ec2 create-tags --resources $INSTANCE_ID --tags Key=$TAG_KEY,Value=$TAG_VALUE
done
```

啟用成本分配標籤以使用 AWS 帳單工具

建議您啟用使用者定義的成本分配標籤。這可讓 AWS 計費工具（例如 Cost Explorer 和 ）識別和篩選 Rightsizing 標籤 AWS Cost and Usage Report。如果您未啟用此功能，則標籤篩選選項和資料將無法使用。如需使用成本分配標籤的資訊，請參閱 文件中的 AWS Billing and Cost Management [啟用使用者定義的成本分配標籤](#)。

1. 登入 AWS Management Console 並開啟 [AWS Billing 主控台](#)。
2. 在導覽窗格中的帳單區段中，選擇成本分配標籤。
3. 在使用者定義的成本分配標籤索引標籤上，輸入授權。
4. 選取授權標籤金鑰，然後選擇啟用。

24 小時後，標籤應該會顯示在 Cost Explorer 中。

使用 Systems Manager Automation 實作正確的大小調整建議

調整大小是需要停止和啟動執行個體的案例。在此案例中，您可能需要在維護時段中處理此中斷，並需要不同的團隊來處理自己的調整大小。在變更執行個體類型之前，請在 Amazon EC2 文件中檢閱 [相容執行個體類型的考量事項](#)。

本節中的範例步驟使用名為 [AWS-ResizeInstance](#) 的 Systems Manager Automation 文件，實作每個帳戶和區域正確的大小調整建議。這種方法對於大多數組織來說都是典型的，因為大多數組織需要不同的執行個體類型來實現不同的目的。您也可以使用相同的 AWS-ResizeInstance 自動化文件來鎖定單一和多帳戶部署。

1. 登入 AWS Management Console 並開啟 [Systems Manager 主控台](#)。
2. 在導覽窗格中的共用資源區段中，選擇文件。
3. 在搜尋列中，輸入 AWS-ResizeInstance，然後從 AWSResizeInstance 搜尋結果中選擇。
4. 選擇 Execute automation (執行自動化)。
5. 在執行自動化 Runbook 頁面上，選擇簡易執行。

6. 在輸入參數區段中，輸入 InstanceId 和 InstanceType。保留其餘的預設值。
7. 選擇執行，然後等待自動化完成變更執行個體類型的步驟。

考慮替代的調整大小方法

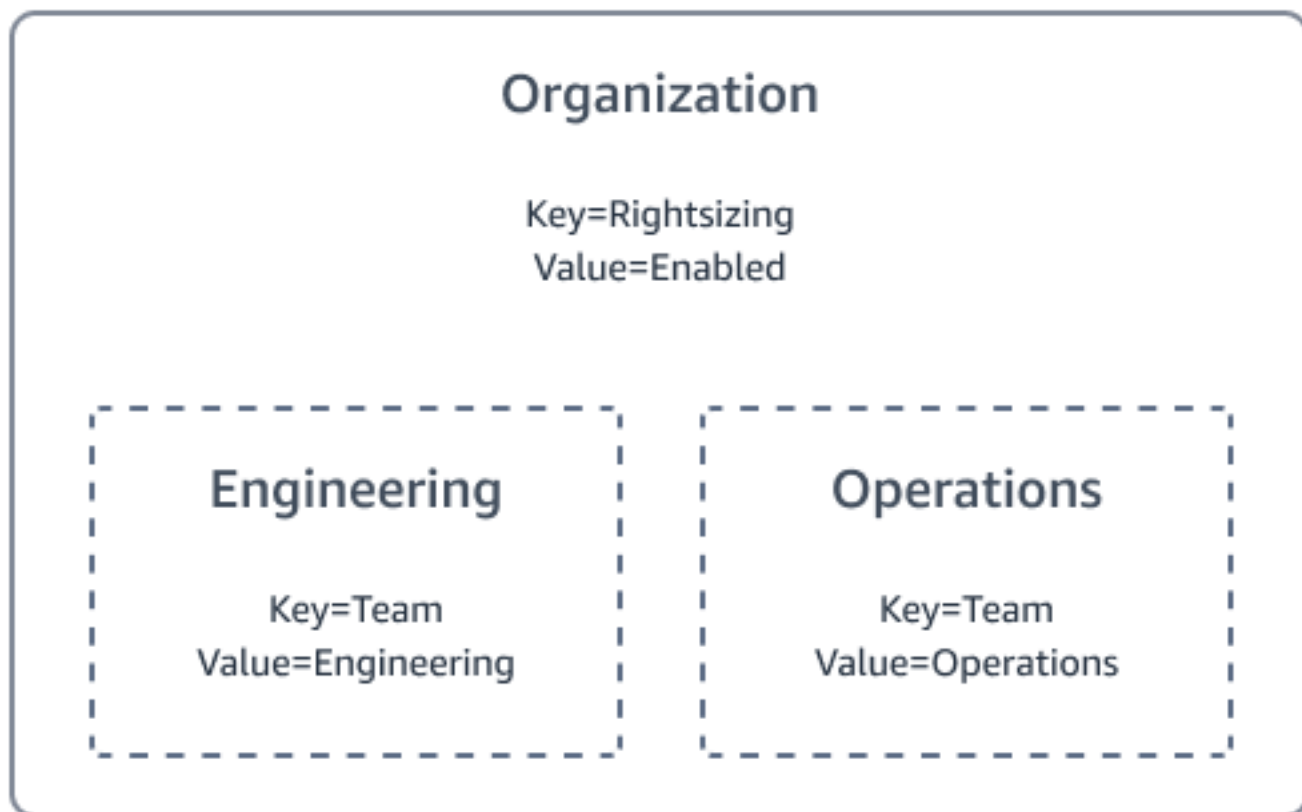
如果您使用啟動範本來部署執行個體，則可以使用適當大小的執行個體類型更新啟動範本，然後執行執行個體重新整理，以使用適當大小的版本取代執行個體。

如果您打算在多個帳戶和區域使用正確的大小調整程序，則必須建立自訂 Systems Manager 自動化文件。本文件可讓您在多個執行個體中以參數和目標執行個體的形式饋送，以移至相同的目的地執行個體類型（例如，所有執行個體轉換為 t3a.medium，無論來源執行個體類型為何）。

在 Cost Explorer 中檢閱費用前後

在您調整資源大小後，您可以使用 Cost Explorer，使用 Rightsizing 標籤在成本之前和之後顯示。請記住，您可以使用 [資源標籤](#) 來追蹤成本。透過使用多層標籤，您可以精細地了解成本。在本指南中介紹的範例中，Rateizing 標籤用於將一般標籤套用至所有目標執行個體。然後，會使用團隊標籤來進一步組織資源。下一步是介紹應用程式標籤，以進一步顯示操作特定應用程式的成本影響。

下圖顯示組織的標籤結構。



考慮正確調整營運團隊所擁有生產 Web 伺服器大小的企業範例。在 Cost Explorer 中，Rightsizing 標籤設定為啟用，而 Team 標籤設定為操作。在此範例中，正確的規模調整工作會將操作成本從每小時 0.89 美分降低到每小時 0.28 美分。假設每月 744 小時，正確調整規模之前的年度成本為 7,945.92 美元。在正確調整規模後，年成本降至 2,499.84 美元。這表示年度工作負載成本降低了 68.5%。想像一下，這在大型組織中的影響。請記住，這是在範例環境中完成的，而且執行個體大部分是閒置的。在生產環境中，您可以看到節省 10–35%。

現在，請考慮正確調整工程團隊擁有的生產基礎主機大小的影響。在 Cost Explorer 中，Rightsizing 標籤設定為已啟用，而 Team 標籤設定為 engineering。在此範例中，正確的規模調整工作會將成本從每小時 0.75 美分降低到每小時 0.44 美分。假設每月 744 小時，正確調整規模之前的年度成本為 6,696.00 美元。在正確調整規模後，年度成本降至 3,928.32 美元。

如果您使用多個標籤，則可以篩選資料以精細化成本詳細資訊。在此範例中，團隊標籤可減少噪音，讓您可以在團隊層級檢視影響。由於已啟用 Rightsizing 標籤，因此您也可以篩選具有已啟用值或不存在的任何標籤執行個體。這可以提供正確大小調整工作的全域檢視，特別是在 Cost Explorer 層級的管理帳戶（付款人）中檢視時。此檢視可讓您查看所有帳戶和執行個體。

請考慮單一帳戶層級的範例，其中 Rightsizing 標籤設定為啟用。營運成本從每小時 1.64 美元降至每小時 0.72 美分。假設每月 744 小時，正確調整規模之前的年度成本為 14,641.92 美元。在正確調整規模後，年度成本降至 6,428.16 美元。這表示此帳戶的運算成本降低了 56%。

在展開正確的規模調整旅程之前，請考慮下列事項：

- AWS 提供許多降低成本的選項。這包括 [AWS OLA](#)，其中會在移至之前 AWS 檢閱您的內部部署執行個體 AWS。AWS OLA 也提供正確的大小調整建議和授權指南。
- 在購買 [Savings Plans](#) 之前，請先完成所有正確的大小調整。這可協助您避免在 Savings Plans 承諾上過度購買。

建議

我們建議採取下列步驟：

1. 檢閱您現有的環境，並考慮將 Amazon EBS gp2 磁碟區轉換為 gp3 磁碟區。
2. 檢閱 [Savings Plans](#)。

其他資源

- [AWS Compute Optimizer](#)（AWS 文件）

- [標記 AWS 資源的最佳實務](#) (AWS 白皮書)
- [如何從您的 \(\) 收集和 AWS Compute Optimizer AWS Trusted Advisor 跨 AWS Organizations 收集資料](#) YouTube
- [最佳化效能並降低授權成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 執行個體](#) (AWS 部落格上的 Microsoft 工作負載)

為 Windows 工作負載選取正確的執行個體類型

概觀

與內部部署環境相比，雲端中操作的工作負載之間存在顯著區別，這是過度佈建的做法。購買內部部署使用的實體硬體時，您預期資本支出會持續一段預定的期間，通常為 3-5 年。為了適應硬體生命週期期間的預期成長，會使用比工作負載目前所需的資源更多的資源來取得硬體。因此，實體硬體通常會過度佈建，遠超過實際工作負載的需求。

虛擬機器 (VM) 技術已成為利用剩餘硬體資源的有效方法。VMs 使用 vCPUs 和過度佈建的管理員 RAM，讓 Hypervisor 透過將未使用的資源配置給每個 VM，來管理忙碌和閒置伺服器之間的實體資源用量。管理時 VMs，分配給每個 VM 的 vCPU RAM 和資源會以資源協調者的角色運作更多，而不是實際用量的指標。VM 資源過度配置可能很容易超過可用運算資源的三倍。

[Amazon Elastic Compute Cloud \(Amazon EC2 \)](#) 不會 VMs 過度佈建基礎硬體，因為這是不必要的。雲端運算是一種營運費用，而不是資本費用，您只需支付使用的費用。如果您的工作負載未來需要更多資源，請在實際需要時佈建這些資源，而不是先佔。

選擇正確的 [Amazon EC2 執行個體類型](#) 有數百種選項。如果您計劃將 Windows 工作負載遷移至雲端，AWS 會提供 [AWS OLA](#)，以協助您更了解目前的工作負載，並提供其在 上效能的範例 AWS。此 AWS OLA 分析旨在將適當的 EC2 執行個體類型和大小與您的實際內部部署使用量配對。

如果您已在 Amazon 上執行工作負載 EC2 並尋求成本最佳化策略，本指南的本節可協助您識別 Amazon EC2 執行個體之間的差異及其對典型 Windows 工作負載的適用性。

成本最佳化建議

若要最佳化 EC2 執行個體類型的成本，建議您執行下列動作：

- 為您的工作負載選擇正確的執行個體系列
- 了解處理器架構之間的價格差異

- 了解跨EC2世代的效能差異價格
- 遷移至較新的執行個體
- 使用爆量執行個體

為您的工作負載選擇正確的執行個體系列

請務必為您的工作負載選擇正確的執行個體系列。

Amazon EC2執行個體分為下列各種群組：

- 一般用途
- 運算最佳化
- 記憶體最佳化
- 加速運算
- 儲存最佳化
- HPC 最佳化

大多數 Windows 工作負載都屬於下列類別：

- 一般用途
- 運算最佳化
- 記憶體最佳化

若要進一步簡化，請考慮每個類別中的基準EC2執行個體：

- 運算最佳化 – C6i
- 一般用途 – M6i
- 記憶體最佳化 – R6i

上一代EC2執行個體在處理器類型中表現出微小差異。例如，C5 運算最佳化執行個體的處理器比 M5 一般用途執行個體或 R5 記憶體最佳化執行個體更快。最新一代的EC2執行個體（C6i、M6i、R6i、C6a、M6a和 R6a都會跨執行個體系列使用相同的處理器。由於處理器在最新一代的執行個體中是一致的，因此執行個體系列之間的價差現在更取決於的數量RAM。RAM 執行個體擁有的越多，成本就越高。

下列範例說明在 us-east-1區域中執行之 Intel 4 vCPU 執行個體的每小時定價。

執行個體	vCPUs	RAM	每小時價格
c6i.xlarge	4	8	0.17 美元
m6i.xlarge	4	16	0.19 美元
r6i.xlarge	4	32	0.25 美元

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

易爆執行個體

雖然關閉未使用的運算資源以避免產生費用是雲端運算的最佳實務，但並非所有工作負載都可以關閉，而且每次需要時都會開啟。有些工作負載會長時間保持閒置，但必須每天 24 小時都能存取。

可爆執行個體（T3）提供一種方法，讓您整天維持線上尖峰或低利用率工作負載，同時仍維持低運算成本。易爆 EC2 執行個體具有執行個體可在短時間內使用的 vCPU 資源數量上限。這些執行個體使用以 [可爆 CPU 點數為基礎的系統](#)。這些點數會在一天的閒置期間累積。可爆式執行個體提供不同的 vCPU-to-RAM 比率，使它們成為在某些情況下運算最佳化執行個體和其他執行個體的一般用途執行個體的替代方案。

下列範例說明在 us-east-1 區域中執行的 T3 執行個體（即爆量執行個體）的每小時定價。

執行個體	vCPUs	RAM (GB)	每小時價格
t3.nano	2	0.5	\$0.0052
t3.micro	2	1	\$0.0104
t3.small	2	2	\$0.0208
t3.medium	2	4	0.0416 美元
t3.large	2	8	\$0.0832
t3.xlarge	4	16	0.1664 美元

執行個體	vCPUs	RAM (GB)	每小時價格
t3.2xlarge	8	32	0.3328 美元

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

了解處理器架構之間的價格差異

[Intel](#) 處理器自成立以來一直是 EC2 執行個體的標準。較早世代的 EC2 執行個體，例如 C5、M5 和 R5，不會指示 Intel 作為處理器架構（因為它是預設值）。較新一代的 EC2 執行個體，例如 C6i、M6i 和 R6i 包含「i」來表示 Intel 處理器的使用。

處理器架構註釋的變更是由於引入了其他處理器選項。與 Intel 最相似的處理器是 [AMD](#)（以「a」表示）。AMD EPYC 處理器使用相同的 x86 架構，並提供類似於 Intel 處理器的效能，但價格較低。如下列定價範例所示，AMD EC2 執行個體提供運算成本的大約 10% 折扣，與其 Intel 對等產品相比。

Intel 執行個體	每小時價格	AMD 執行個體	價格	% 差異
c6i.xlarge	0.17 美元	c6a.xlarge	0.153 美元	10%
m6i.xlarge	0.192 美元	m6a.xlarge	0.1728 美元	10%
r6i.xlarge	0.252 美元	r6a.xlarge	0.2268 美元	10%

Note


定價是以 us-east-1 區域中的隨需每小時定價為基礎。

第三個主要處理器架構選項是 EC2 執行個體上的 [AWS Graviton 處理器](#)（以「g」表示）。Graviton 處理器由設計 AWS，可在 Amazon 上提供最佳價格效能 EC2。目前的 Graviton 處理器不僅比其 Intel 的處理器便宜 20%，而且還提升了 20% 或更高的效能。新一代 Graviton 處理器預計將進一步擴展此效能差異，測試顯示效能額外提高 25%。

Windows Server 無法在以ARM架構為基礎的 Graviton 處理器上執行。事實上，Windows Server 只在 x86 處理器上執行。雖然您無法在 Windows Server 中使用 Graviton 型執行個體來提高 40% 的價格效能，但您仍然可以使用具有特定 Microsoft 工作負載的 Graviton 處理器。例如，[較新的版本。NET 可以在 Linux 上執行](#)。這表示這些工作負載可以使用ARM處理器，並從更快、更經濟實惠的 Graviton EC2執行個體中受益。

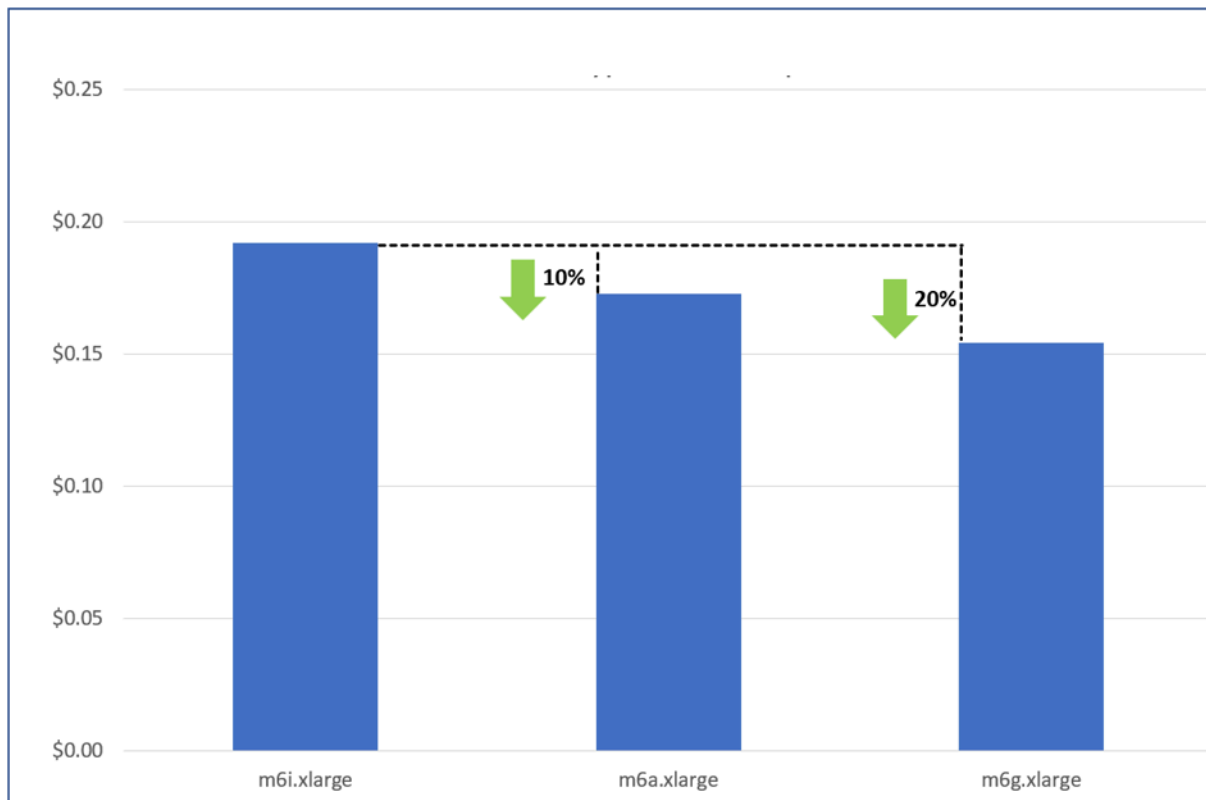
下列範例說明在 us-east-1區域中執行的 Graviton 執行個體的每小時定價。

Intel 執行個體	每小時價格	Graviton 執行個體	每小時價格	% 差異
c6i.xlarge	0.17 美元	c6g.xlarge	0.136 美元	20%
m6i.xlarge	0.192 美元	m6g.xlarge	0.154 美元	20%
r6i.xlarge	0.252 美元	r6g.xlarge	0.2016 美元	20%

 Note

定價是以 us-east-1區域中的隨需每小時定價為基礎。

下圖比較 M 系列執行個體的價格。



了解跨EC2世代的價格效能差異

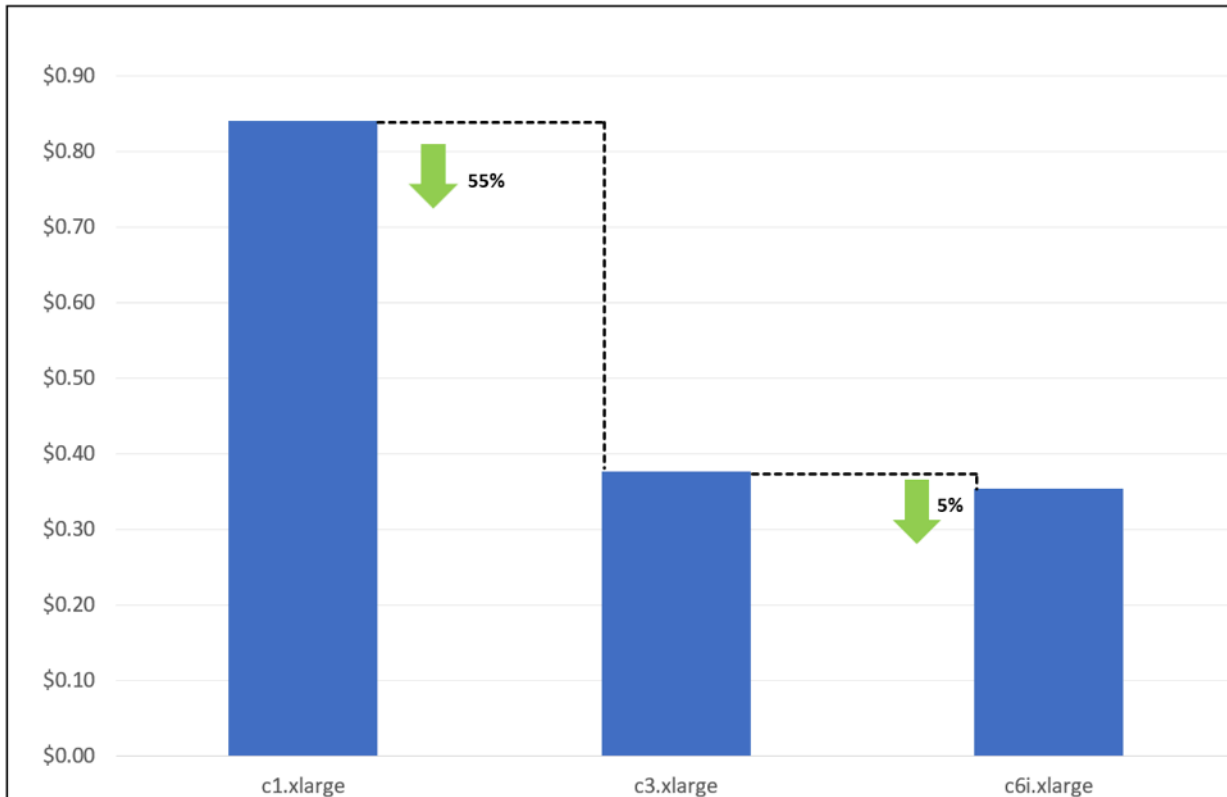
Amazon 最一致的特徵之一EC2是，每個新一代都比上一代產品提供更好的價格效能。如下表所示，較新一代EC2執行個體的價格會隨著後續版本而降低。

運算最佳化執行個體	每小時價格	一般用途執行個體	每小時價格	記憶體最佳化執行個體	每小時價格
C1.xlarge	0.52 美元	M1.xlarge	0.35 美元	r1.xlarge	N/A
C3.xlarge	0.21 美元	M3.xlarge	0.266 美元	r3.xlarge	0.333 美元
C5.xlarge	0.17 美元	M5.xlarge	0.192 美元	r5.xlarge	0.252 美元

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

下列圖表比較不同世代 C 系列執行個體的成本。



不過，如下表所示，第 6 代執行個體的价格與第 5 代相同。

運算最佳化執行個體	每小時價格	一般用途執行個體	每小時價格	記憶體最佳化執行個體	每小時價格
C5.xlarge	0.17 美元	M5.xlarge	0.192 美元	r5.xlarge	0.252 美元
C6i .xlarge	0.17 美元	M6i .xlarge	0.192 美元	r6i.xlarge	0.252 美元

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

儘管成本相同，但由於處理器速度更快、網路輸送量增強，以及 Amazon Elastic Block Store (Amazon EBS) 輸送量增加，新一代提供卓越的價格效能 IOPS。

最重要的價格效能改善之一是 [X2i 執行個體](#) 的增強。此世代的執行個體提供比上一代高出 55% 的價格效能。如下表所示，x2iedn 在每個效能方面都展現出改善（全部價格與上一代相同）。

執行個體	每小時價格	vCPUs	RAM	處理器速度	執行個體儲存體	聯網	Amazon EBS輸送量	EBS IOPS
x1e.2xlarge	1.66 美元	8	244	2.3 GHz	237GB SSD	10 Gbps	125 MB/s	7400
x1iedn.2xlarge	1.66 美元	8	256	3.5 GHz	240GB NVMe SSD	25 Gbps	2500 MB/s	65000

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

範例方案

考慮追蹤交付工具並希望改善其 SQL 伺服器效能的分析公司範例。MACO SME 檢閱公司的效能瓶頸後，公司會從 x1e.2xlarge 執行個體轉換為 x2iedn.xlarge 執行個體。新的執行個體大小較小，但對 x2 執行個體的增強功能可透過使用緩衝集區延伸來提高 SQL 伺服器效能和最佳化。這可讓公司從 SQL Server Enterprise 版本降級為 SQL Server Standard 版本。它還允許公司將其 SQL 伺服器授權從 8 減少 vCPUs 到 4vCPUs。

最佳化之前：

Server	EC2 執行個體	SQL 伺服器版本	每月成本
產品DB1	x1e.2xlarge	Enterprise	3,918.64 美元
產品DB2	x1e.2xlarge	Enterprise	3,918.64 美元
總計			7,837.28 美元

最佳化後：

Server	EC2 執行個體	SQL 伺服器版本	每月成本
產品DB1	x2iedn.xlarge	標準	1,215.00 美元
產品DB2	x2iedn.xlarge	標準	1,215.00 美元
總計			2,430.00 美元

所有合併後，從 x1e.2xlarge 執行個體變更為 x2iedn.xlarge 執行個體可讓公司在範例案例中每月在生產資料庫伺服器上節省 5,407 美元。這可將工作負載的總成本降低 69%。

Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

遷移至較新的執行個體

較舊世代的 Amazon 會在 Xen Hypervisor EC2 上執行，而較新一代的 Amazon 會在 [AWS Nitro System](#) 上執行。Nitro 系統幾乎會將主機硬體的所有運算和記憶體資源提供給執行個體。這可改善整體效能。從 [Xen 遷移至 Nitro 型執行個體](#) 時，有特殊考量。例如，[AWS Windows AMIs](#) 會設定為 Microsoft 安裝媒體使用的預設設定和自訂。自訂包含支援最新一代執行個體類型的驅動程式和組態（在 [Nitro System 上建置的執行個體](#)）。

如果您從自訂 Windows AMIs 或從 Amazon AMIs 提供的 Windows 啟動執行個體，而該執行個體是在 2018 年 8 月之前建立，建議您完成從 [遷移到 Amazon 文件中最新一代執行個體類型的](#) 步驟。EC2

使用爆量執行個體

雖然爆量執行個體是節省運算成本的好方法，但我們建議您在下列情況下避免這些執行個體：

- 具有桌面體驗的 [Windows Server 最低規格](#) 需要 2 GB 的 RAM。避免將 t3.micro 或 t3.nano 執行個體與 Windows Server 搭配使用，因為它們缺少最低數量的 RAM。
- 如果您的工作負載繁多，但沒有足夠的閒置時間來建置爆量額度，則使用正常 EC2 執行個體比使用爆量執行個體更有效率。我們建議您 [監控您的 CPU 點數](#)，以驗證這一點。
- 在大多數情況下，建議您避免將爆量執行個體與 SQL 伺服器搭配使用。SQL Server 的授權是以 vCPUs 指派給執行個體的數目為基礎。如果 SQL 伺服器在一天的大部分時間都處於閒置狀態，則您

將支付未充分利用SQL的授權。在這些情況下，我們建議您將多個SQL伺服器執行個體合併到較大的伺服器。

後續步驟

建議您採取下列步驟，以最佳化 Amazon EC2 Windows 執行個體的成本：

- 使用最新一代的EC2執行個體以獲得最佳價格效能。
- 搭配AMD處理器使用EC2執行個體可降低 10% 的運算成本。
- 選擇符合您工作負載的EC2執行個體類型，將資源使用率最大化。

下表顯示 Windows 工作負載的一般起點範例。還有其他選項，例如執行個體儲存磁碟區，以增強SQL伺服器工作負載或具有更大 vCPU-to-RAM比率的EC2執行個體。我們建議您徹底測試工作負載，並使用監控工具 AWS Compute Optimizer，例如協助進行必要的調整。

工作負載	典型	選用
Active Directory	T3, M6i	R6i
檔案伺服器	T3, M6i	C6i
Web 伺服器	T3, C6i	M6i, R6i
SQL 伺服器	R6i	x2iedn、X2iezn

如果您必須變更EC2執行個體類型，程序通常只需要簡單的伺服器重新啟動。如需詳細資訊，請參閱 [Amazon 文件中的變更執行個體類型](#)。EC2

在您變更執行個體類型之前，建議您考慮下列事項：

- 您必須先停止 Amazon 支援的執行個體，EBS才能變更其執行個體類型。請務必在執行個體停止時規劃停機時間。停止執行個體並變更其執行個體類型可能會花費數分鐘，重新啟動您的執行個體所需要花費的時間則會根據您應用程式的啟動指令碼而有所不同。如需詳細資訊，請參閱 Amazon EC2 文件中的 [停止和啟動執行個體](#)。
- 當您停止和啟動執行個體時，會將執行個體 AWS 移至新的硬體。如果您的執行個體具有公有IPv4地址，會 AWS 釋出地址，並為執行個體提供新的公有IPv4地址。如果您需要未變更的公有IPv4地址，請使用 [彈性 IP 地址](#)。

- 如果在執行個體上啟用[休眠](#)，則無法變更執行個體類型。
- 您無法變更 [Spot 執行個體](#)的執行個體類型。
- 如果您的執行個體位於 Auto Scaling 群組中，Amazon EC2 Auto Scaling 會將停止的執行個體標記為運作狀態不佳，並可能會終止執行個體並啟動替換執行個體。若要避免這種情況，您可以在變更執行個體類型時，暫停群組的擴展程序。如需詳細資訊，請參閱 Amazon [Auto Scaling 文件中的暫停和繼續 Auto Scaling 群組的程序](#)。EC2 Auto Scaling
- 當您使用執行個體存放區磁碟區變更執行個體的NVMe執行個體類型時，更新的執行個體可能會有額外的執行個體存放區磁碟區，因為即使未在 Amazon Machine Image (AMI) 或NVMe執行個體區塊裝置映射中指定執行個體存放區磁碟區，所有執行個體存放區磁碟區都可以使用。否則，更新後的執行個體通常會有與您在啟動原始執行個體時指定之執行個體儲存體磁碟區相同數目的執行個體儲存體磁碟區。

其他資源

- [Amazon EC2執行個體類型](#) (AWS 文件)
- [AWS 最佳化和授權評估](#) (AWS 文件)

為 Windows 和SQL伺服器工作負載帶來授權

概觀

如果您在 Microsoft 工作負載和現有企業授權協議上有大量投資，您可以選擇多個 AWS 選項來支援這些工作負載，包括[包含的授權 \(由提供 AWS\)](#) 和 [自攜授權 \(BYOL\)](#) 選項。您可以使用 [Amazon EC2 Dedicated Hosts](#) 充分利用現有的 Microsoft 授權合約，並將 Windows Server 帶到 AWS。這最多可為您節省 Amazon EC2執行個體成本的 50%。由於 Windows 授權大約佔執行個體成本的一半，因此 AWS 將 Windows Server 帶到專用主機上可大幅節省成本。由於 Windows Server 無法設定為[預設 \(共用\) 租用](#)，因此如果您想要在上使用現有 Windows Server 授權，專用主機是理想的選擇 AWS。

專用主機不僅適用於 Windows Server BYOL執行個體。它們還提供靈活性，以匹配現有SQL伺服器工作負載的內部部署授權。專用主機會公開基礎伺服器的實體核心，並可讓您在實體核心層級授權SQL伺服器。在預設 (共用) 租用中，SQL伺服器授權是根據CPU分配給執行個體的虛擬數量而定。此功能可讓您在 上 AWS 以與您的內部部署授權策略一致的方式授權SQL伺服器工作負載。因此，與預設 (共用) 租用相比，除了節省執行個體成本之外，您還可以使用合格的 Windows 授權，節省高達 50% 的SQL伺服器授權成本。如需此案例的詳細資訊，請參閱本指南的[了解SQL伺服器授權](#)一節。

Amazon EC2 專用主機

Amazon EC2 專用主機基本上是 AWS 用來執行其 EC2 運算方案的相同 EC2 主機。差別在於，這些主機完全專用於單一客戶，並提供對基礎實體基礎設施的獨家存取。您可以使用專用主機在完全專用於您的硬體上執行執行個體，而不是與其他 AWS 客戶共用資源。這可讓您更妥善地控制雲端資源，並可讓您將 Windows Server 和 SQL Server 等自己的軟體授權帶到 來降低成本 AWS。

請謹記以下幾點：

- 專用主機是完全專用於單一客戶的實體伺服器。您可以查看專用主機的通訊端和實體核心，以便解決授權合規要求，例如每個通訊端、每個核心或每個虛擬機器軟體授權合約。
- 可支援相同執行個體系列的多個執行個體大小的專用主機稱為異質專用主機。這些[執行個體系列](#)包括 T3, A1, C5, M5, R5, C5n, R5n 和 M5n 相反地，其他執行個體系列僅支援相同專用主機上的一個執行個體大小。這些稱為同質專用主機。
- 專用主機是以每個主機為基礎計費。這表示無論您在其中執行多少執行個體，每個專用主機都會向您收取費用。專用主機定價會根據選取的執行個體系列、區域和付款選項而有所不同。您可以為工作負載選擇最佳組態，以達到所需的效能和成本結果。

此圖表說明共用租用執行個體與專用主機之間的差異。



同質專用主機

考慮使用 M6i 專用主機的情況。M6i 和 R6i 專用主機有兩個通訊端、64 個實體核心，並支援相同大小的執行個體類型。這些稱為同質專用主機。這表示您可以在單一 M6i 專用主機上啟動的執行個體數量取決於執行個體大小。

例如：

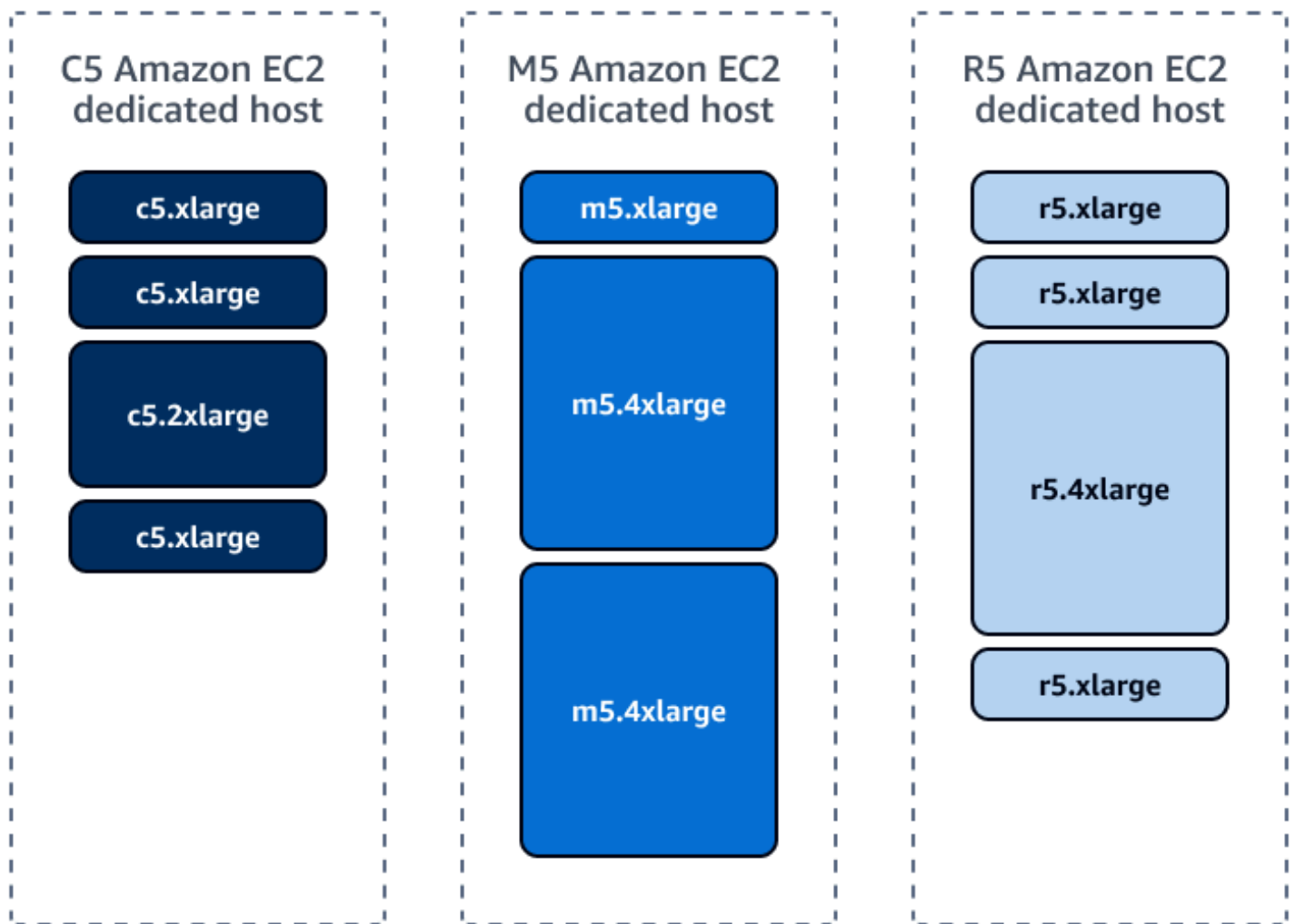
- 對於 xlarge (4vCPUs) ，您最多可以在此專用主機上啟動 32 個 m6i.xlarge 執行個體。
- 若為 8xlarge (32 vCPUs) ，您最多可以在此專用主機上啟動 4 個 m6i.8xlarge 執行個體。
- 如果是金屬 (128 vCPUs) ，您最多可以在此專用主機上啟動 1 m6i.metal 執行個體。

下圖顯示 M6 執行個體的專用主機選項。



異質專用主機

支援相同主機上多個執行個體大小的專用主機稱為異質 Amazon EC2 專用主機。下圖顯示具有各種執行個體大小的 C5, M5 和 R5 專用主機範例，例如 2xlarge、xlarge 和 4xlarge。



專用主機管理

在管理 Amazon EC2 專用主機方面，建議您考慮下列事項：

- 若要充分利用專用主機，您可以在[組織中的多個帳戶之間共用單一主機](#)。主機共用可讓資源最佳化，並可透過使用主機上的每個可用插槽來節省成本。透過在業務單位之間共用專用主機，您可以集中 IT 基礎設施並提高資源利用率，同時仍然保持工作負載之間的分離。如果您是組織的一部分，AWS Organizations 並在您的組織中啟用共用，則組織中的取用者會自動獲得共用專用主機的存取權。否則，消費者會收到加入資源共享的邀請，並且在接受邀請後便能存取共用的專用執行個體。
- 您可以在專用主機上執行 Windows Server 2022，其使用包含授權的模型，因為 Windows Server 2019 是您可以的最新版本 BYOL。如果您想要在專用主機上使用 Windows Server 2022，則必須使用 Windows Server 2022 授權包含的執行個體。
- [AWS License Manager](#) 是全方位解決方案，可管理跨 AWS 和內部部署環境不同廠商的軟體授權。如果您[使用 License Manager](#)，您可以更清楚地了解和控制軟體授權的使用方式，進而節省成本並

改善合規性。您可以使用 License Manager 設定規則來模擬您唯一的授權條款。這可讓您強制執行這些規則，並防止授權濫用。這可以降低不合規的風險，並改善授權管理程序。

- 您可以使用 License Manager，透過使用主機[資源群組](#)來自動化主機的置放、發行和復原。這可以提高生產力並減少管理開銷。License Manager 也根據授權規則，提供跨內部部署環境 AWS 和內部部署環境的授權用量集中檢視，讓您輕鬆管理整個組織的增量授權購買、合規和供應商稽核。此外，License Manager 與 AWS Organizations 和 AWS Resource Access Manager (AWS RAM) 整合，以在帳戶和區域之間共用授權組態。這可讓您根據排程為整個環境建立報告，並在單一集中管理授權規則 AWS 帳戶。最終，這可以改善治理並降低複雜性。
- 為單一區域中的專用主機設計高可用性時，請確定您已為生產關鍵工作負載在至少兩個可用區域中配置至少兩個專用主機。如需詳細資訊，請參閱參考部署[上的 Amazon EC2 Dedicated Hosts for Microsoft Windows AWS](#)。
- 對於每個專用主機執行個體系列，您可以針對每個執行個體大小執行的執行個體數量都有限制。如需詳細資訊，請參閱 Amazon EC2 文件中的[專用主機組態表](#)。

AWS 授權選項

授權分為下列主要類別：

- 包含授權 – 此授權選項可讓您隨需購買和使用授權，只需支付使用的費用。它非常適合您在授權使用上尋求靈活性，並希望避免預付成本的使用案例。您可以從各種 Windows Server、SQL Server 和其他 Microsoft 產品中進行選擇。
- BYOL 具有 License Mobility 的產品 – 如果您已經擁有現有授權，並想要在雲端中使用它們，此授權選項可讓您透過 [Microsoft License Mobility 方案](#) 將自己的授權帶到雲端。具有授權行動性的產品，例如具有軟體保證 (SA) 的 SQL 伺服器，可以提供給共用或專用租用。這可降低 AWS 執行個體成本。
- BYOL 沒有 License Mobility 的產品 – 對於 Windows Server 這類缺少 License Mobility 的 Microsoft 產品，AWS 會提供專用選項，以在雲端使用這些產品。此外，專用主機可在實體核心層級啟用授權，因此可節省 50% 或更多執行工作負載所需的授權。專用主機是大多數時間執行的穩定且可預測工作負載的絕佳選擇。

攜帶 Windows Server 授權

自備 Windows 授權是授權最佳化的最有效策略之一，因為它可讓您利用現有的投資並降低 AWS 您的費用。特定 BYOL 案例不需要 SA 或 License Mobility 利益，但 Amazon EC2 專用基礎設施始終是必要的。若要符合資格，您必須在 2019 年 10 月 1 日之前購買永久授權，或在 2019 年 10 月 1 日之

前根據作用中 Enterprise Enrollment 新增永久授權做為校正。在這些特定BYOL案例中，您只能將授權升級至 2019 年 10 月 1 日之前可用的版本。例如，如果您在 2017 年捨棄 SA，則有權最多僅部署 Windows Server 2016，而不是 2019。不過，2019 是適用於 BYOL 的最後一個版本 AWS。如需詳細資訊，請參閱 AWS 文件中的[授權 – Windows Server](#)。

攜帶授權可能會大幅影響在上執行 Microsoft 工作負載的成本 AWS。當您自帶授權時，您不需要為在雲端中執行的執行個體支付額外的授權成本，這可以節省大量成本。

下表顯示在各種組態上 24/7 全天候執行單一 c5.xlarge 執行個體的隨需每月成本。

組態	每月成本 (USD)
Windows Server + SQL Server Enterprise 版	1,353.00 美元 (LI)
Windows Server + SQL Server Standard 版	609.00 美元 (LI)
僅限 Windows Server	259.00 美元 (LI)
僅限運算 (Linux)	127.00 美元

您可以使用現有的授權來降低授權成本，並節省整體 AWS 帳單的費用。

若要在 Amazon EC2 Dedicated Hosts BYOL上符合的資格，您必須自備軟體授權，例如適用於 Windows Server 和 SQL Server 的。BYOL 可讓您在上述使用現有的授權 AWS，並節省成本。若要自帶授權，您必須擁有軟體供應商的授權許可，並且必須提供軟體的安裝媒體或映像。安裝媒體或映像可用來在專用主機上啟動執行個體。若要進一步了解如何建立 BYOL AMI，請參閱 AWS 部落格 Microsoft 工作負載中的[如何使用 VM Import/Export AMIs 從內部部署建立 Windows Server Bring-Your-Own-License](#)。

Note

設定為 Auto 的授權類型相當於[AWS 包含授權的選項](#)。此選項可能會導致不需要的隨需支出。您需要切換[授權類型](#)。

成本最佳化案例

正確調整和最佳化授權是上成本最佳化的關鍵元件 AWS。如果您實作正確的策略，您可以使用 Amazon EC2 Dedicated Hosts 和 BYOL 選項，降低授權成本、維持合規，並從授權投資中獲得最佳價值。

本節涵蓋下列範例案例：

- 使用 T3 專用主機節省成本
- 將共用租用與專用主機與 SQL 伺服器進行比較 BYOL
- 高度可用的 SQL 伺服器部署

使用 T3 專用主機節省成本

T3 專用主機與傳統上提供固定 CPU 資源的其他 Amazon EC2 專用主機不同。相反地，T3 專用主機支援可爆量執行個體，這些執行個體能夠共用 CPU 資源、提供基準 CPU 效能，以及在需要時爆量。共享 CPU 資源也稱為過度訂閱，這可讓單一 T3 專用主機支援比相當一般用途專用主機多四倍的執行個體。

T3 專用主機提供比任何其他 Amazon EC2 專用主機更高的執行個體密度 TCO，進而降低了。可爆 T3 執行個體可讓您合併更多執行個體，並 low-to-moderate 平均 CPU 利用率比以往任何時候都更少的主機。與其他 Amazon 專用主機相比，T3 EC2 專用主機也在更多 vCPU 和記憶體組合中提供較小的執行個體大小。較小的執行個體大小有助於降低 TCO 並協助交付相當於或大於內部部署主機的合併比率。

T3 專用主機最適合執行具有 low-to-moderate CPU 使用率和符合資格的每個插槽、每個核心或每個虛擬機器軟體授權 BYOL 的軟體，包括 Microsoft Windows 桌面、Windows Server、SQL 伺服器和 Oracle 資料庫。

使用 T3 專用主機來減少 Windows Server Datacenter 授權（每個核心）

在內部部署環境中，您會利用一個事實，即您可以輕鬆地在 VMware 主機 CPUs 上過度訂閱實體，並實現高度整合。

請考量下列範例。您目前在內部部署環境中使用 10x36 核心、384 GB RAM VMware 主機。此外，每個主機都執行 96x2 v CPU、4 GB RAM Windows Server 虛擬機器，平均 CPU 使用率較低。

您現在可以透過將虛擬機器移至 T3 專用主機來實現更高層級的合併，其數量 RAM 是目前內部部署 VMware 主機的兩倍。您可以在 T3 專用主機上執行相同數量的伺服器，主機成本降低 50%。這可協助您將 Windows Server 授權成本降低 33%。下表強調使用 T3 專用主機可節省的成本。

	內部部署VMware主機	T3 專用主機	節省
實體伺服器	10	5	
每個主機的實體核心	36	48	
RAM 每個主機 (GB)	384	768	
2 v CPU , RAMVMs每 個主機 4 GB	96	192	
總數 VMs	960	960	
Windows Server Datacenter 授權總數 (每個核心) = (伺 服器數目 * 實體核心計 數)	10 * 36 = 360	5 * 48 = 240	33%

將共用租用與專用主機與SQL伺服器進行比較 BYOL

考慮實際範例，以示範 Amazon EC2 專用主機的價值。在此案例中，組織在具有 240 個核心的內部部署環境中執行SQL伺服器工作負載，並希望在上以具成本效益的方式部署相同的工作負載 AWS。如果此組織擁有自己的授權（BYOL），他們將繼續為 SA 付費，並減少直接影響其成本的核心數量。

下圖比較 Microsoft 權限和SQL伺服器之間的 AWS 節省。

Microsoft entitlements (Enterprise Agreements)		SQL Server savings with AWS	
	Number of cores	AWS shared vCPUs	AWS BYOL/Dedicated Hosts cores
SQL Server Enterprise edition	208	120	96
SQL Server Standard edition	32	20	-
Total SA cost	\$341,000	\$197,418	\$151,355

透過在 AWS 共用租用上正確調整執行個體大小，您可以將SQL伺服器授權減少至 140 個核心。這會導致 SA 成本為 197,000 美元。

Amazon EC2 專用主機可讓您在實體核心層級授權SQL伺服器。在共用租用中，SQL伺服器授權是以 vCPUs 配置給執行個體的數量為基礎，這是不可能的。因此，透過使用兩個 R5 專用主機，每個都具

有 48 個核心，您只需涵蓋 96 個核心，而不需要共用租用 vCPUs 所需的 140 個核心。透過部署 R5 專用主機並在實體層級授權工作負載，您可以將所需的 SQL Server Enterprise Edition 授權數目縮減至 96 個核心。這表示您可以部署多達 192 個核心（針對超執行緒計算）的 SQL 伺服器工作負載，同時仍然符合授權需求並大幅節省成本。

在此情況下，組織每年支付約 341,000 美元的 SA 成本。在正確調整共用租用大小之後，它們會使用 140 將成本降低至 197,000 美元 vCPUs。Amazon EC2 Dedicated Hosts 進一步將成本降低至 151,000 美元（約降低 56%）。

高度可用的 SQL 伺服器部署

此範例會分析成本如何影響具有 AWS 各種授權考量的 SQL 伺服器部署。假設組織需要在上部署六個 SQL Server Enterprise 伺服器 AWS 以支援三個應用程式。這些伺服器需要高可用性，每個伺服器都有 16 vCPUs 和 256 GB 的容量 RAM。請參閱下列案例詳細資訊：

- 伺服器 – SQL 伺服器
- 作業系統版本 – Windows Server Datacenter 2019
- SQL Server Edition – SQL Server Enterprise 2019
- vCPU – 16
- 記憶體（GB）– 256
- 數量 – 6

若要在不影響效能 AWS 的情況下最佳化的成本，建議您根據 CPU、記憶體、網路和磁碟（IOPS/BW）使用率來調整執行個體大小。正確調整工作負載大小後，請將它們放在 x2iedn.4xlarge 執行個體類型上，該類型提供 16 個 vCPUs。不過，此執行個體類型也包含工作負載所需的兩倍記憶體。仍然可以進一步最佳化。

案例 1

組織使用 Windows 和 SQL Server 的授權包含選項，在 AWS 共用租用上部署六個 Server Enterprise SQL 伺服器。使用此選項，Windows 和 SQL Server 授權的成本會併入執行個體價格。請參閱下列案例詳細資訊：

- 共用租用（執行個體）– x2iedn.4xlarge
- 每小時成本（USD）– \$10.0705
- 每月每單位成本（USD）– \$7,351.47
- 伺服器數量 – 6

- CPU – 16
- 記憶體 – 512
- 6 個伺服器的每月成本 – \$44,108

案例 2

組織在共用租用上具有 SA 和 BYOL for SQL Server。這表示組織使用 Windows 的包含授權選項，但會根據 vCPUs 分配給執行個體的數目提供自己的 SQL 伺服器授權。由於組織有六個 SQL Server Enterprise 伺服器，vCPUs 每個伺服器為 16 個，因此總共需要 96 vCPUs 個伺服器。請參閱下列案例詳細資訊：

- 共用租用（執行個體）– x2iedn.4xlarge
- 每小時成本（USD）– \$4.0705
- 每月每單位成本（USD）– \$2971.47
- 伺服器數量 – 6
- CPU – 16
- 記憶體 – 512
- BYOL 核心 – 96
- 6 個伺服器的每月成本 – \$17,828

透過使用 SA 自有的 SQL 伺服器授權，此案例中的組織相較於使用 SQL Server 的授權包含選項，可以節省成本。精確的成本節省取決於特定授權協議的定價和條款。在此情況下，將 SQL Server Enterprise 授權提供給時，AWS 成本每個月會減少 26,280 美元 AWS。

案例 3

組織 BYOL 對 Amazon EC2 專用主機上的 Windows 和 SQL Server 都有。這表示組織將在實體核心層級指派授權，使其僅授權主機的實體核心。在實體核心層級的授權可讓您部署執行個體數量上限，而不會影響所需的授權。此授權模型通常與 Windows Server Datacenter 和 SQL Server Enterprise Edition 搭配使用。

此案例使用兩個 X2iezn Amazon EC2 專用主機。每個主機都有 24 個實體核心和 48 個 vCPUs。這可為六個 SQL Server Enterprise 伺服器提供足夠的容量，RAM 每個伺服器各有 16 vCPUs 和 256 GB。請參閱下列案例詳細資訊：

- 專用主機數目 – 2

- 執行個體系列 – x2iezn
- 每小時成本 (USD) – \$11.009
- 每月每單位成本 (USD) – \$8 , 036
- 實體核心 – 48
- 可用 vCPU – 96
- 需要 Windows Server 核心授權 – 24
- SQL Server Enterprise 核心所需的授權 – 24
- 每月成本 – 16 , 073

兩個 X2iezn 系列 Amazon EC2 專用主機的總成本為每月 16 , 073 美元。如需定價的詳細資訊，請參閱此案例的 AWS Pricing Calculator [預估](#)。在這種情況下，組織可以攜帶 Windows 授權，每月節省 1 , 755.65 美元。如果他們使用 Amazon EC2 專用主機，也可以減少所需的 SQL 伺服器授權數量。在共用租用中，他們需要 96 個 SQL Server Enterprise 授權，以涵蓋具有 16 vCPUs 個授權的六個 SQL Server Enterprise 伺服器。不過，透過在實體核心層級使用 Amazon EC2 專用主機和授權，他們可以將所需的授權數量減少為 48 個核心。

下列詳細資訊會比較範例 3 的成本，並顯示相較於其他案例，在 Amazon EC2 Dedicated Hosts 上部署工作負載與 BYOL 選項，可節省多少成本。

- 內部部署伺服器 – SQL 伺服器
- vCPU – 16
- 記憶體 – 256
- 伺服器數量 – 6
- 案例 1 的每月成本：Windows (LI) + SQL Server Enterprise (LI) – \$44 , 108
- 案例 2 的每月成本：Windows (LI) + SQL Server Enterprise (BYOL) – \$17 , 828
- 案例 3 的每月成本：Amazon EC2 專用主機上的 Windows (LI) + SQL Server Enterprise (BYOL) – \$16 , 073

Note

成本以隨需定價為基礎。您可以使用 Savings Plans 或專用預留執行個體進一步降低成本。相較於隨需定價，這些選項提供彈性定價模型，可大幅節省成本。透過這些計劃，您可以承諾 1 或 3 年的期限。如需詳細資訊，請參閱本指南的 [Windows on Amazon 最佳化支出 EC2](#) 一節。

請考慮下列 Amazon EC2 專用主機的付款選項：

- [專用主機](#) (Amazon EC2 文件)
- [專用主機預留](#) (Amazon EC2 文件)
- [Savings Plans](#) (Amazon EC2 文件)

現在[AWS Pricing Calculator](#)支援專用主機定價。這可協助您選擇適當的基礎專用主機。

成本最佳化建議

建議您採取下列步驟，使用來最佳化成本 AWS Cost Explorer：

1. [啟用 Cost Explorer](#)。
2. 使用 Cost Explorer [檢視和分析 Amazon Dedicated Host 部署的成本和用量](#)。EC2
3. 驗證您正在執行 BYOL。您可以在 Amazon EC2主控台的執行個體或AMI頁面，或 describe-images或 describe-instances命令傳回的回應中，顯示下列平台詳細資訊和用量操作值。
 - 平台詳細資訊：Windows、用量操作：RunInstances：0002 (包含授權)
 - 平台詳細資訊：Windows BYOL、用量操作：RunInstances：0800

其他資源

- [授權類型轉換的合格授權類型](#) (AWS License Manager 文件)
- [AWS License Manager 和 專用主機研討會](#) (AWS License Manager 研討會)
- [Amazon EC2 專用主機 FAQs](#) (AWS 文件)
- [如何使用 VM Import/Export AMIs 從內部部署建立 Windows Server Bring-Your-Own-License](#) (部落格上的 AWS Microsoft 工作負載)
- [VM 匯入/匯出](#) (AWS 文件)
- [Amazon Web Services 和 Microsoft：常見問題](#) (AWS 文件)
- [License Manager 中的授權類型轉換](#) (AWS License Manager 文件)
- [在 Amazon EC2 專用主機上部署高可用性SQL伺服器](#) (AWS Cloud Operations & Migrations 部落格)

最佳化 Windows on Amazon 的支出 EC2

概觀

將伺服器遷移至的其中一個首要問題 AWS 是基礎設施成本。事實上，雲端的好處之一是按需支付資源，但有一些生產工作負載需要 24 小時全年無休提供。[Savings Plans](#) 旨在節省跨 EC2 執行個體 AWS Lambda 和 的穩定狀態 AWS 用量 AWS Fargate。

Savings Plans 提供彈性的定價模式，可協助您降低 Amazon EC2、Fargate、Lambda 和 Amazon SageMaker 用量的定價，以換取對一致用量（例如每小時 10 美元）的承諾。您承諾在一到三年內維持一致的每小時運算支出量，並換取該用量的折扣。

您可以使用 Savings Plans 來選擇三種不同的付款選項：

- 無預付選項不需要任何預付款項，您的承諾只會每月收取。
- 部分預付選項提供 Savings Plans 的較低價格。您至少需要預先支付一半的承諾費用，剩餘部分則每月收費。
- 所有預付選項提供最低價格，且您的全部承諾會在一次付款中收費。

您可以在 [AWS Cost Explorer](#) 中追蹤您的 Savings Plans 到期和即將排入佇列的 Savings Plans AWS Cost Explorer。您可以使用 Savings Plans 提醒，在計劃過期日期前 1、7、30 或 60 天，或當承諾排入購買佇列時，接收預先電子郵件通知。這些通知也會在過期日期提醒您。您可以傳送通知給最多 10 個電子郵件收件人。

了解 Savings Plans

每種類型的運算用量都有隨需費率和 Savings Plans 費率。如果您承諾達到每小時 10 美元的運算用量，則以 Savings Plans 費率取得所有用量的 Savings Plans 價格，最高可達 10 美元。運算支出承諾以外的任何用量都會按常規隨需費率收費。您可以在 [AWS Cost Explorer](#) 中使用 Cost Explorer 來開始使用 Savings Plans AWS Management Console。

您可以使用 [Cost Explorer](#) 中提供的建議，輕鬆對 Savings Plans 做出承諾，以實現最大節省。建議的每小時承諾取決於您的歷史隨需用量，以及您選擇的計劃類型、期限長度和付款選項。Savings Plans 會先套用至購買該計畫的帳戶，然後與合併帳單系列中的其他帳戶共用。

Note

預設 AWS Organizations 會啟用 中的 Savings Plans 共用選項。您可以在付款人帳戶的 AWS Billing 主控台中拒絕此選項。您可以造訪[建議](#)頁面，查看 AWS 建議協助您節省合格用量的 Savings Plans。您可以隨時重新整理這些建議，以便輕鬆購買最佳 Savings Plans

計算節省計劃

Compute Savings Plans 提供最大的彈性，並協助您降低成本。無論 EC2 執行個體系列、大小、可用區域、區域、作業系統或租用，這些計劃都會自動套用至執行個體用量。它們也適用於 Fargate 或 Lambda 使用。例如，透過 Compute Savings Plans，您可以將工作負載從 C4 變更為 M5 執行個體、從 EU（愛爾蘭）轉移到 EU（倫敦），或隨時 EC2 將工作負載從 Fargate 或 Lambda 移至 Fargate。您會自動繼續支付 Savings Plans 價格。

EC2 執行個體 Savings Plans

EC2 Instance Savings Plans 提供最深層的折扣，以換取對區域中個別執行個體系列用量的承諾（例如，在維吉尼亞北部承諾一致的 M5 用量層級）。這會自動為您提供該區域中所選執行個體系列的隨需價格折扣，無論可用區域、大小、作業系統或租用。EC2 Instance Savings Plans 可讓您變更該區域中系列中執行個體之間的用量。例如，您可以從執行 Windows 的 c5.xlarge 移至執行 Linux 的 c5.2xlarge，並自動受益於 Savings Plans 價格。

運算和 EC2 執行個體 Savings Plans 都適用於屬於 Amazon EMR、Amazon Elastic Kubernetes Service（Amazon EKS）和 Amazon Elastic Container Service（Amazon ECS）叢集的 EC2 執行個體。Savings Plans 不涵蓋 EMR、Amazon EKS、Amazon 和 Amazon ECS 費用，但基礎 EC2 執行個體為。EC2 Instance Savings Plans 會在 Compute Savings Plans 之前套用，因為 Compute Savings Plans 具有更廣泛的適用性。

Note

做出承諾後，您無法輕鬆變更 Savings Plan。建議您在承諾任一 Savings Plans 選項之前，先仔細規劃。Savings Plans 提供比隨需定價更低的價格，以換取承諾，且無法在期間內取消。

每小時承諾範例

如果您購買 Savings Plan，您會對計劃期限做出每小時的貨幣承諾。如果您承諾達到每小時 10 美元的運算用量，則 Savings Plan 定價會自動套用至每小時最多 10 美元的所有用量。超出承諾的任何用

量都會按常規隨需費率收費。您可以使用 Cost Explorer 中的 Savings Plans 購買建議工具，取得可最大限度地節省成本的建議承諾。特定計畫的每小時財務承諾無法在計畫期間內修改。如果您想要在分析用量之後增加承諾，則可以購買額外的 Savings Plan 來涵蓋超額用量。

Savings Plans 的優點

與預留執行個體相比，Savings Plans 提供更靈活的定價模型，可節省您的成本，同時充分利用 Savings Plans 提供的更多運算選項。即使您的運算需求改變，Savings Plans 也提供折扣。這可協助您跟上不斷變化的動態環境，而不會產生任何額外的管理開銷。以下是使用 Savings Plans 的一些其他優點：

- 易於使用 – 獲得自動折扣以換取貨幣承諾。
- 彈性 – 適用於多種使用類型的單一承諾。
- 潛在節省 – 有多種方式可以節省。請考量下列範例：
 - 使用 Compute Savings Plans 節省 60% 的 Windows Server 工作負載 ([d2.8xlarge、3 年、全部預付、視窗、共用租用、us-east-2](#))
 - 使用 EC2 執行個體 Savings Plans ([d2.8xlarge、3 年、全部預付、視窗、共用租用、us-east-2](#)) 節省 73% 的 Windows Server 工作負載
 - 非外部執行個體類型 ([t3 系列、3 年、全部預付、視窗、共用租用、us-east-2](#)) 節省 28–41%
 - Windows Server 平均節省 25–40%

Note

EC2 由於靈活性降低，執行個體 Savings Plans 提供比 Compute Savings Plans 更高的折扣。您承諾以折扣價格使用。

每種類型的運算用量都有 Savings Plan 費率和隨需費率。下表顯示每種作業系統類型的 Savings Plans 和隨需費率。系統會針對承諾用量向您收取 Savings Plans 費率，超過承諾範圍的任何用量都會按常規隨需費率收費。

執行個體名稱	Savings Plans 費率	隨需節省	隨需費率	作業系統	區域	付款選項	術語長度
x2iedn.xlarge	0.32 美元	61%	0.83 美元	Linux	美國東部 (維吉尼亞北部)	不預付	3
x2iedn.xlarge	\$2.01	50%	\$1.02	Windows	美國東部 (維吉尼亞北部)	不預付	3
x2iedn.xlarge	\$1.02	20%	2.52 美元	包含 Windows 授權 + SQL Server Enterprise Edition	美國東部 (維吉尼亞北部)	不預付	3
x2iedn.xlarge	0.32 美元	61%	0.83 美元	BYOL	美國東部 (維吉尼亞北部)	不預付	3

Savings Plans 包含作業系統，且有單獨的折扣BYOL。它們都會在 [Compute Savings Plans 計算器](#) 中細分。

預留執行個體定價模型

AWS 根據稱為預留執行個體的承諾，有另一個定價模型。如果運算在您做出承諾之後發生變更，導致預留執行個體變成未使用，則此模型可能會發生問題。Savings Plans 旨在提供與[標準和可轉換預留執行個體](#)類似的成本降低，但具有更大的靈活性。無論EC2執行個體系列、大小、作業系統、租用或區域為何，Compute Savings Plans 都會提供較低的執行個體使用價格。它們也實現了最大的靈活性。

下表可協助您選擇 Savings Plans 或預留執行個體。

	Reserved Instance	EC2 執行個體 Savings Plans	計算節省計劃
平均 1 年折扣	高達 38%	高達 29%	高達 29%
平均 3 年折扣	高達 58%	高達 73%	高達 60%
執行個體系列	Fixed	Fixed	彈性
執行個體大小	已修正 (非 Linux)	彈性	彈性
地理位置	1 個區域	1 個區域	彈性
作業系統	Fixed	彈性	彈性
服務	Amazon EC2 或 Amazon RDS	Amazon EC2	Amazon EC2、Fargate、Lambda
付款選項	全部、部分、無預付	全部、部分、無預付	全部、部分、無預付
執行個體限制	每個可用區域 20 個	沒有限制	沒有限制

Note

Savings Plans 的運作方式是為您提供以小時貨幣承諾為基礎的折扣。在計劃期間內，無法取消或變更每小時財務承諾，但您可以購買其他 Savings Plans 來支付額外的用量。這可讓您在機群成長時保持一致的每小時承諾。

您可以使用 [AWS Cost Explorer](#) 或 [AWS 雲端 Intelligence Dashboards](#) 等工具來追蹤您的承諾。Cost Explorer 提供涵蓋範圍目標行，可協助您的組織規劃其 Savings Plans 涵蓋範圍策略。如果工作負載的 75% 為穩定狀態，則 75% 是良好目標。根據動態工作負載，這會留下 25% 的隨需/可變支出。如果您需要將提高到 85% 的涵蓋範圍，您可以購買另一個 Savings Plans 承諾來提高每小時貨幣承諾。

Note

我們建議您購買 Savings Plans 而非預留執行個體，但如果您已購買預留執行個體，這兩個承諾模型可以一起運作。

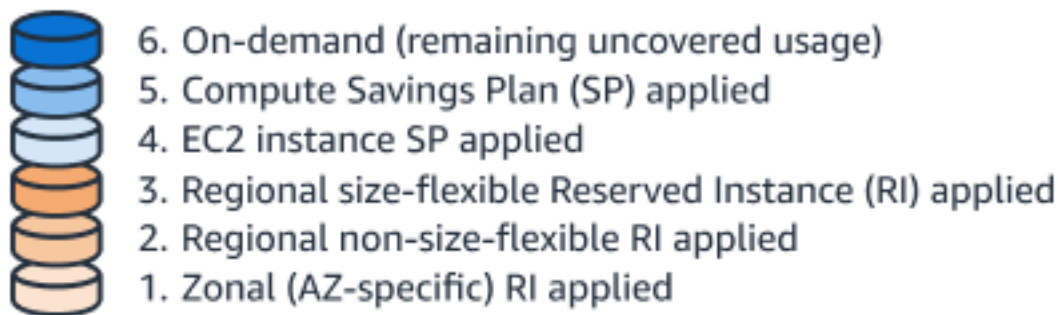
請考慮您購買預留執行個體，但想要開始嘗試 Savings Plans 選項的範例。此組合有邏輯可套用至您的最終帳單。以下是您可以套用至的階層 AWS 帳戶：

1. 區域預留執行個體適用於擁有該執行個體的帳戶。如果預留執行個體還剩小時，則適用於組織的其餘部分。
2. Windows 的非大小彈性區域預留執行個體適用於擁有 Windows 之帳戶的相符用量。任何剩餘的項目都會推展到組織的其他部分。
3. 大小靈活的區域預留執行個體適用於擁有該執行個體的帳戶（先是系列中最小的執行個體，再到較大的執行個體），然後是組織的其他部分。
4. 區域預留執行個體適用於任何未使用的隨需容量保留。
5. EC2 Instance Savings Plans 適用於購買它的帳戶。
6. Compute Savings Plans 適用於購買它的帳戶。

Note

折扣從產生最高折扣的用量開始，然後降至最低折扣。對於最常見的執行個體類型（例如 T3, M6 和 C5），傳統上 Windows 執行個體的折扣潛力低於 Linux。這表示在大多數情況下，Linux 執行個體比 Windows 執行個體受益更多。

下圖顯示將預留執行個體與 Savings Plans 分割後的價格。運算和 EC2 執行個體 Savings Plans 都適用於先執行的執行個體，然後適用於未使用的隨需容量保留。



成本最佳化案例

本節涵蓋使用授權包含計費模型的 Amazon EC2 專用主機和 Amazon EC2 執行個體的成本最佳化案例。

Amazon EC2 專用主機

請考慮您將將內部部署 Windows 工作負載遷移至 的案例 AWS。您的資料中心具有下列伺服器：

- 兩個伺服器，具有 16 vCPU 和 128 GB RAM
- 兩個伺服器，具有 32 vCPU 和 164 GB RAM
- 一個伺服器，具有 8 vCPU 和 64 GB RAM
- 16 個伺服器，具有 vCPU 和 32 GB RAM

此外，假設您可以將自己的授權帶至 ， AWS 因為您有足夠的授權可以帶出。下表顯示您可以在 中使用的伺服器執行個體 AWS。

執行個體類型	CPU	RAM	Amount (數量)
r5.4xlarge	16	128	2
r5.8xlarge	32	256	2
r5.2xlarge	8	64	1
r5.xlarge	4	32	16
			21

一項分析顯示，這 21 個虛擬機器可以使用 R5 執行個體系列主機分散到兩個專用主機。下表顯示這兩個專用主機的成本。

專用主機隨需 案例	預付薪資	1 個月	1 年	3 年	AWS Pricing Calculator
隨需	無	10,123 美元	121,475 美元	364,392 美元	AWS Pricing Calculator 估算
1 年 Savings Plan	無	7,447 美元	89,362 美元	–	AWS Pricing Calculator 估算

專用主機隨需案例	預付薪資	1 個月	1 年	3 年	AWS Pricing Calculator
3 年 Savings Plan	無	5,476 美元	65,712 美元	197,128 美元	AWS Pricing Calculator 估算
3 年 Savings Plan, 含預付款項	84,438 美元	2,755 美元	117,499 美元	183,618 美元	AWS Pricing Calculator 估算

如果您有要遷移至的伺服器 AWS，則 1 年 Savings Plan 的最終價格為 89,362 美元，而非隨需價格為 121,475 美元。這表示一年後可獲得 26.5% 的折扣。如果您正在考慮長時間停留 AWS，則可以選擇 3 年 Savings Plan，以節省更深層的成本。在三年結束時，您需支付 \$197,128，而不是 \$364,392。這可節省三年後總金額的 46%。

包含授權的 Amazon EC2 執行個體

請考慮一個案例，其中您將遷移單一三層應用程式至 AWS，並且您想要使用提供的授權 AWS。此外，假設您的應用程式適用於下列伺服器：

- 兩個 Web 伺服器，具有 2 GB vCPUs 和 4 GB RAM
- 兩個應用程式伺服器，具有 8 GB vCPUs 和 16 GB RAM
- 兩個資料庫伺服器，具有 16 vCPUs 和 64 GB RAM (使用 SQL Server Standard 版)

下表顯示您可以在中使用的伺服器執行個體 AWS。

執行個體類型	CPU	RAM	Amount (數量)
c5.large	2	4	2
c5.2xlarge	8	16	2
r5.2xlarge	8	64	2
			6 個伺服器

下表顯示 中這些伺服器的成本 AWS。

由 包含的授 權 AWS	預付薪資	1 個月	1 年	3 年	AWS Pricing Calculator
隨需	無	3,912 美元	46,950 美元	140,849 美 元	AWS Pricing Calculator 估 算
1 年 Savings Plan	無	3,466 美元	41,952 美元		AWS Pricing Calculator 估 算
3 年 Savings Plan, 無需預 付	無	3,189 美元	38,264 美元	114,804 美 元	AWS Pricing Calculator 估 算
3 年 Savings Plan, 含預付 款項	112,110 美 元	無	無	無	AWS Pricing Calculator 估 算

如果您想要以隨需定價執行這些伺服器用於生產環境（24/7），您每月需支付 3,912 美元的成本。支付此每月成本等於一年後 46,950 美元，三年後總計 140,849 美元。

如果您選擇 1 年無預付款的 Savings Plan，每月費用會降低至 3,466 美元。在第一年結束時，您需支付 \$41,952。這是 11% 的總折扣。如果您選擇 3 年不預付的 Savings Plan，每月成本將降至 3,189 美元。在三年結束時，您需支付 \$114,804。這可讓您節省 18.5% 的費用。

成本最佳化建議

這兩種情況都可協助您在 中規劃和預測工作負載時節省成本 AWS。請務必了解，與第一個案例相比，第二個案例的折扣較少。在第二個案例中，授權價格會包含在雲端伺服器的價格中。AWS 不會提供授權價格的折扣，但您可以始終攜帶授權（在特定情況下），並始終 AWS 保證最佳的運算/執行個體價格。

建議您執行下列動作，以控制運算和執行個體資源的 AWS 支出：

- 存取建議

- 根據您的需求自訂建議
- 檢閱每小時承諾

存取建議

您可以使用 [Amazon EC2主控台](#) 來存取 Savings Plan 的建議。您甚至可以下載建議，以便稍後以 CSV 格式檢閱。如需詳細資訊，請參閱 [Savings Plans 文件中的監控您的 Savings Plans](#)。

根據您的需求自訂建議

開啟 [Amazon EC2主控台](#)，展開執行個體區段，然後選擇 Savings Plans。此頁面會顯示建議前後的執行個體和運算定價。您也可以針對建議調整下列因素：

- 術語 – 例如，1–3 年
- 付款選項 – 例如預付、部分預付或無預付
- 歷史記錄 – 例如，過去 7、30 或 60 天

檢閱每小時承諾

使用相同的範例，假設您有一個 24 小時全年無休執行的執行個體。建議是使用 Savings Plan。根據規模，您的隨需價格為每小時 120 美元。您可以選擇每小時提交 90 美元，但這可能會因區域、執行個體和購買選項而有所不同。在此範例中，相較於隨需成本，您可以節省 25%。如果使用率和涵蓋範圍低於您定義的閾值，您也可以追蹤它們，並在預算即將結束時設定提醒。

檢閱建議

建議您仔細檢閱 Savings Plan 建議。AWS 不會在未經您許可的情況下變更任何內容。這些只是建議，由您決定是否套用。

購買計劃

開啟 [Amazon EC2主控台](#)，展開執行個體區段，然後選擇 Savings Plans。然後，選擇購買 Savings Plans。根據您的需求，您可以選擇下列選項：術語、區域、執行個體系列、每小時承諾、付款選項，甚至是開始日期。您可以從 Compute Savings Plans、EC2Instance Savings Plans 和 SageMaker Savings Plans 中進行選擇。如需詳細資訊，請參閱 [Savings Plans 文件中的購買 Savings Plans](#)。

取得使用率報告

購買 Savings Plan 後，您可以取得使用率報告。此報告可協助您檢查使用率、查看購買的計劃是否足以涵蓋並最大化折扣，以及取消或新增折扣。此報告可以匯出為其他格式，例如 CSV。如需詳細資訊，請參閱 Savings Plans 文件中的[使用率報告](#)。

遵循購買最佳實務

建議您在購買 Savings Plans 之前遵循這些最佳實務：

- 使用 [AWS Trusted Advisor](#) 移除閒置 EC2 資源。
- 在購買 Savings Plans 之前執行任何正確的規模調整。
- 建立您持續保留 30-60 天的每小時費率。
- 購買一項承諾，以涵蓋組織願意的一致每小時費率。考慮需求或季節的波動。
- 選擇每季檢閱 Savings Plans 預算，以維持一致的費率（例如，S Savings Plans 涵蓋範圍的 70% 涵蓋範圍目標）。如果費率低於所需的涵蓋範圍，請購買額外的 Savings Plan 做為校正，以符合您的涵蓋範圍目標。

其他資源

- [Amazon EC2 Reserved Instances 的 Savings Plans](#)（AWS 白皮書）
- [了解 Savings Plans 如何套用至您的 AWS 用量](#)（Savings Plans 文件）
- [宣佈 EC2 Windows Server 和 SQL Server Instances 的每秒計費](#)（AWS 文件）
- [AWS 成本最佳化系列：Savings Plans 影片 | Amazon Web Services](#)（YouTube）

使用 AWS 工具監控成本

概觀

成本可見性是最佳化成本的關鍵因素 AWS。AWS 擁有許多工具，您可以使用這些工具來視覺化成本並建立提醒以因應這些成本。這些工具包括等工具 AWS Budgets，可協助您追蹤和報告支出。本節涵蓋監控 Windows AWS 支出的特定方法，以便您可以追蹤並回應預算需求。這包括將必要的標籤新增至您的 Windows EC2 資源。這些標籤可讓您使用 [來正確監控 Windows EC2 和其他 Microsoft 服務 AWS Budgets](#)。

透過監控支出並使用 AWS 工具建立提醒，您可以更了解目前的支出、預計支出和異常支出。如果您使用 [Savings Plans](#) 協助降低每小時 EC2 執行個體定價，建議您檢視 Savings Plan 的整體使用率和涵蓋範圍。這可協助您確保持續實現節省。您可以使用 AWS Cost Explorer 來檢視 Savings Plan 庫存，並根據先前的用量取得其他 Savings Plans 的建議。您也可以使用 [AWS Budgets](#) 和 [設定](#) 來追蹤特定支出 [AWS Cost Anomaly Detection](#)。

成本最佳化建議

建議您採取下列步驟，使用 AWS Budgets、Cost Explorer 和異常偵測來最佳化成本：

- 標記 Windows EC2 資源
- 使用 [設定提醒](#) AWS Budgets
- 啟用成本異常偵測
- 取得即時支出分析
- 使用 Cost Explorer 檢視 Windows 授權包含的支出

標記 Windows EC2 資源

若要有效監控 AWS 支出，您必須為要監控的工作負載建立 [標記策略](#)。這很重要，以便您可以分類分組資源，並收到特定支出的通知，而不是一般用量支出。您可以使用標記資源，這些資源不僅有助於降低成本，還可以用於其他目的，例如 [AWS Systems Manager 自動化](#)。此外，我們建議您針對 [必要的標籤](#) 實作一些管理。

若要追蹤您在 AWS Budgets、Cost Explorer 和 Cost Anomaly Detection 中的支出，您必須確保有適當的標籤。您可以使用標籤為符合這些標籤的項目設定特定預算，以便在支出增加時收到提醒。

例如，您可以使用簡單的標籤，例如 Key=OS Value=Windows。這會將您的所有 Windows 執行個體整合到一個群組中，您可以追蹤其支出。您也可以將標籤用於其他項目，例如 Systems Manager。建立標籤後，您必須啟用標籤以進行成本追蹤。考慮新增 [AWS Config 規則來監控連接至特定資源的標籤](#)。如果有執行中的資源不包含適當的標籤，AWS Config 可以提醒您，這可讓您準確呈現 Windows EC2 支出。

標籤準備就緒後，您可以在 [中](#) 建立自訂預算 AWS Billing。這可讓您檢視 Windows EC2 支出。您可以設定每日預算或每月預算。

使用 設定提醒 AWS Budgets

在此範例案例中，您會為 Windows 建立每日預算 EC2。這是經常性預算，使用自動調整選項來追蹤您的支出並相應地調整預算。如果您有靜態環境，則可以改用固定預算。請務必選擇基準時間範圍（例如 30 天）。

1. 登入 AWS Management Console 並開啟 [AWS Cost Management 主控台](#)。
2. 在導覽窗格中，選擇 Budgets (預算)。
3. 在頁面頂端，選擇 Create budget (建立預算)。
4. 在預算設定下，選擇自訂（進階）。
5. 在預算類型下，選擇成本預算。然後選擇下一步。
6. Under Details，for Budget 名稱，輸入預算的名稱。例如，Windows EC2 會花費。
7. 在設定預算金額下，針對期間，選擇每日。
8. 對於預算續約類型，選擇預算期間之後重設的預算的經常性預算。
9. 針對開始日期，選擇開始日期或期間，開始追蹤預算金額。
10. For Budgeting 方法，選擇自動調整（新增）。
11. 針對基準時間範圍，選擇自訂範圍，然後輸入 30 天。
12. 選擇 Next (下一步)。
13. 在預算範圍區段中，選取篩選特定 AWS 成本維度。這是使用標籤建立適當維度的地方。AWS Budgets 不支援平台類型作為其篩選條件中的選項。因此，您必須套用作業系統標籤。
14. 選擇新增篩選條件，然後從維度選取標籤選項。
15. 選擇 OS 標籤，然後選擇 Windows 值以建立標籤的預算。
16. 選擇 Next (下一步)。
17. 在設定警示頁面上，選擇新增警示閾值。在這裡，您設定了兩個提醒：一個為 50% 閾值，另一個為 100% 閾值。如果 50% 閾值警示在當月中點之前遭到違反，則會提供警告。如此一來，您就可以檢查支出是否超出預期，並在月底之前做出回應。
18. 對於閾值，輸入 50 並選擇預算金額的 %。
19. 針對觸發程序，選擇實際。
20. 對於電子郵件收件人，輸入電子郵件地址。為閾值 100 新增另一個提醒。

Note

此範例使用警示的簡單電子郵件通知，但您也可以使用 [Amazon Chime 或 Slack](#)。

啟用成本異常偵測

您可以使用成本標籤來設定異常的支出提醒。例如，您可以使用 [AWS Cost Anomaly Detection](#) 來建立支出的監控，並在系統偵測到帳戶中的異常支出時收到提醒。

若要為先前建立的 Key=OS 和 Value=Windows 標籤設定監視器和警示，請執行下列動作：

1. 登入 AWS Management Console 並開啟 [AWS Cost Management 主控台](#)。
2. 在導覽窗格中，選擇 Cost Anomaly Detection (成本異常偵測)。
3. 選擇成本監控索引標籤，然後選擇建立監控。
4. 在步驟 1 中，選擇成本分配標籤作為您的監視器類型。
5. 針對成本分配標籤金鑰，選擇 Windows EC2 支出。
6. 對於成本分配標籤值，選擇 Windows。
7. 針對為您的監視器命名，輸入 Windows EC2 支出。
8. 選擇 Next (下一步)。
9. 若要為警示建立訂閱，請選取建立新的訂閱。如果您有現有的訂閱，請選取 Choose an existing subscription (選擇現有的訂閱)。
10. 對於訂閱名稱，輸入 Windows EC2 支出異常。
11. 針對警示頻率，選擇每日摘要。
12. 針對提醒收件人，輸入您的電子郵件地址。
13. 選擇新增閾值。對於閾值，輸入 10，然後選擇高於預期速度的百分比。
14. 選擇 Create monitor (建立監視器)。

即時查看支出

警示是監控 Windows EC2 支出的實用工具，但如果您想要即時檢視支出，則必須使用 Cost Explorer。觀看此影片，了解 Cost Explorer 如何讓您分析和降低成本 EC2。如需詳細資訊，請參閱 [AWS 支援 | 了解和降低您的 EC2 成本](#) 影片 YouTube。

檢視 Windows 授權包含的支出

您可以使用 Cost Explorer 來檢視帳戶中的 EC2 Windows 支出。若要查看 Windows 的授權包含支出，您必須在 Cost Explorer 中設定下列正確的 [篩選條件](#)：

- 針對平台，選擇 Windows (Amazon VPC)。對於 API 操作，選擇 RunInstance : 0002。這是授權包含 Windows EC2 執行個體的 AWS Billing 程式碼。

- 如果您想要檢視BYOL執行個體支出，請將 RunInstance : 0002 變更為 RunInstance : 0800。這是 Windows EC2 的帳單代碼BYOL。

透過 Cost Explorer 中的此可見性，您可以快速篩選成本，確切地篩選您在 Windows 上的支出EC2。如果您想要更深入地探索 AWS 支出，您可以使用 AWS Cost and Usage Report 篩選個別執行個體層級的支出。您也可以產生可在 Amazon 中視覺化 QuickSight 並建置自訂儀表板的報告。

如需詳細資訊，請參閱 上的[AWS 支援您 - 視覺化您的成本和用量報告影片](#) YouTube。

其他資源

- 使用 (AWS Config 文件) [設定所需的標籤 AWS Config](#)
- [AWS Budgets 教學課程 - 設定提醒 AWS Billing | Amazon Web Services](#) (YouTube)
- [AWS Cost and Usage Report 查詢程式庫](#) (AWS Well-Architected Labs)

SQL 伺服器

客戶在上執行 Microsoft AWS 工作負載已超過 15 年，比任何其他雲端供應商都長。這主要是因為 AWS 具有雲端 Microsoft 應用程式的經驗最多，並在下列領域提供 Windows Server 和 Microsoft SQL Server 的最佳平台：

- 更高的效能和可靠性
- 更好的安全和身分服務
- 更多遷移支援
- 最廣泛且最深層的功能
- 降低總擁有成本 (TCO)
- 彈性的授權選項

AWS 支援建置和執行依賴 SQL 伺服器之 Windows 應用程式所需的一切，包括 Active Directory、.NET、SQL Server、Windows 桌面即服務，以及所有支援的 Windows Server 版本。憑藉久經考驗的專業知識，AWS 可協助您輕鬆提升和轉移、重構甚至現代化 Windows 工作負載。

本指南的本節涵蓋下列主題：

- [選擇高可用性和災難復原解決方案](#)
- [了解 SQL 伺服器授權](#)
- [為 SQL 伺服器工作負載選取正確的 EC2 執行個體](#)
- [合併執行個體](#)
- [比較 SQL 伺服器版本](#)
- [評估 SQL 伺服器開發人員版本](#)
- [在 Linux 上評估 SQL 伺服器](#)
- [最佳化 SQL 伺服器備份策略](#)
- [現代化 SQL 伺服器資料庫](#)
- [最佳化 SQL 伺服器儲存](#)
- [使用 Compute Optimizer 最佳化 SQL 伺服器授權](#)
- [使用 Compute Optimizer 最佳化 SQL 伺服器大小](#)
- [檢閱 SQL 伺服器工作負載 Trusted Advisor 的建議](#)

選擇高可用性和災難復原解決方案

概觀

建議您為上的SQL伺服器部署設計架構 AWS，以符合您的業務需求，同時滿足[災難復原 \(DR\) 目標](#)，包括復原時間目標 (RTO) 和復原點目標 (RPO)。下列解決方案可協助您在 Amazon Elastic Compute Cloud (Amazon EC2) 上為SQL伺服器設計正確的架構，同時最佳化SQL伺服器工作負載的成本。

- SQL Server Always On 可用性群組 – SQL Server Always On 可用性群組為SQL伺服器資料庫提供高可用性和災難復原 (HA/DR) 解決方案。可用性群組由一組一起容錯移轉的使用者資料庫組成。Always On 可用性群組也會在資料庫層級提供備援，但不需要共用儲存體，每個複本都有自己的本機儲存體。您可以將此功能部署為 HA/DR 解決方案。如需詳細資訊，請參閱 Microsoft 文件中的[什麼是 Always On 可用性群組？](#)。
- SQL Server Always On 容錯移轉叢集執行個體 (FCI) – SQL Server Always On FCIs 使用 Windows Server 容錯移轉叢集 (WSFC) 在SQL伺服器執行個體層級提供 HA。FCIs 需要共用儲存才能託管資料庫。您可以使用共用區塊儲存或共用檔案儲存。例如，您可以使用 Amazon FSx for Windows File Server 或 Amazon FSx for NetApp ONTAP 作為具有多個可用區域的共用儲存解決方案。如需詳細資訊，請參閱 Microsoft 文件中的[Always On 容錯移轉叢集執行個體 \(SQL 伺服器\)](#)。
- SIOS DataKeeper – SIOS DataKeeper 可以透過啟用同時跨越可用區域和的SQL伺服器FCI，協助您滿足 HA 和 DR 需求 AWS 區域。SIOS DataKeeper SAN會使用本機 Amazon Elastic Block Store (Amazon EBS) 磁碟區建立叢集虛擬，並使用 HA 可用區域之間的不同步複寫，同時使用區域和之間的非同步複寫進行災難復原。如需詳細資訊，請參閱 SIOS 文件中的[Windows 應用程式的高可用性保護](#)。
- 分散式可用性群組 – 分散式可用性群組是一種特殊類型的可用性群組，跨越兩個不同的 Always On 可用性群組。可用性群組可以位於兩個不同的區域 (例如 us-east-1 和 us-west-1)。您可以將分散式可用性群組視為可用性群組的可用性群組，因為基礎 Always On 可用性群組是在兩個不同的 WSFC 叢集上設定。SQL 部署分散式可用性群組需要 Server Enterprise Edition。如需詳細資訊，請參閱 Microsoft 文件中的[分散式可用性群組](#)。
- 日誌寄件 – 您可以實作日誌寄件，以保護跨多個區域的資料庫，在極少數情況下，區域會受到影響且無法使用。根據交易和日誌寄件頻率，您可以在幾分鐘RTO內實現 RPO和。如需詳細資訊，請參閱 Microsoft 文件中的[關於 Log Shipping \(SQL 伺服器\)](#)。
- AWS Elastic Disaster Recovery – Elastic Disaster Recovery 是一種軟體即服務 (SaaS) 應用程式，可管理伺服器從任何基礎設施複寫到 AWS 的 DR。您也可以使用 Elastic Disaster Recovery 跨區域複寫SQL伺服器。Elastic Disaster Recovery 是以代理程式為基礎的解決方案，可將整個虛擬機

器複寫，包括作業系統、所有安裝的應用程式和所有資料庫複製到預備區。如需詳細資訊，請參閱 [Elastic Disaster Recovery 文件中的什麼是 Elastic Disaster Recovery ?](#)。

- AWS Database Migration Service (AWS DMS) – AWS DMS 支援將資料即時遷移至 和從 遷移 AWS，包括不同的 區域。您可以使用此功能在不同區域中設定個別的SQL伺服器執行個體，以做為災難復原資料庫。如需詳細資訊，請參閱 AWS DMS 文件中的 [什麼是 AWS Database Migration Service ?](#)。

SQL Server Always On 可用性群組

如果您只針對高可用性 [Always On 可用性群組](#) 使用 SQL Server Enterprise 版，則可以利用基本可用性群組降級為 SQL Server Standard 版。您可以使用基本可用性群組，而不是 Always On 可用性群組，將成本從 65% 降低到 75%。

Note

如需不同SQL伺服器版本之間成本差異的詳細資訊，請參閱本指南的 [比較SQL伺服器版本](#) 一節。

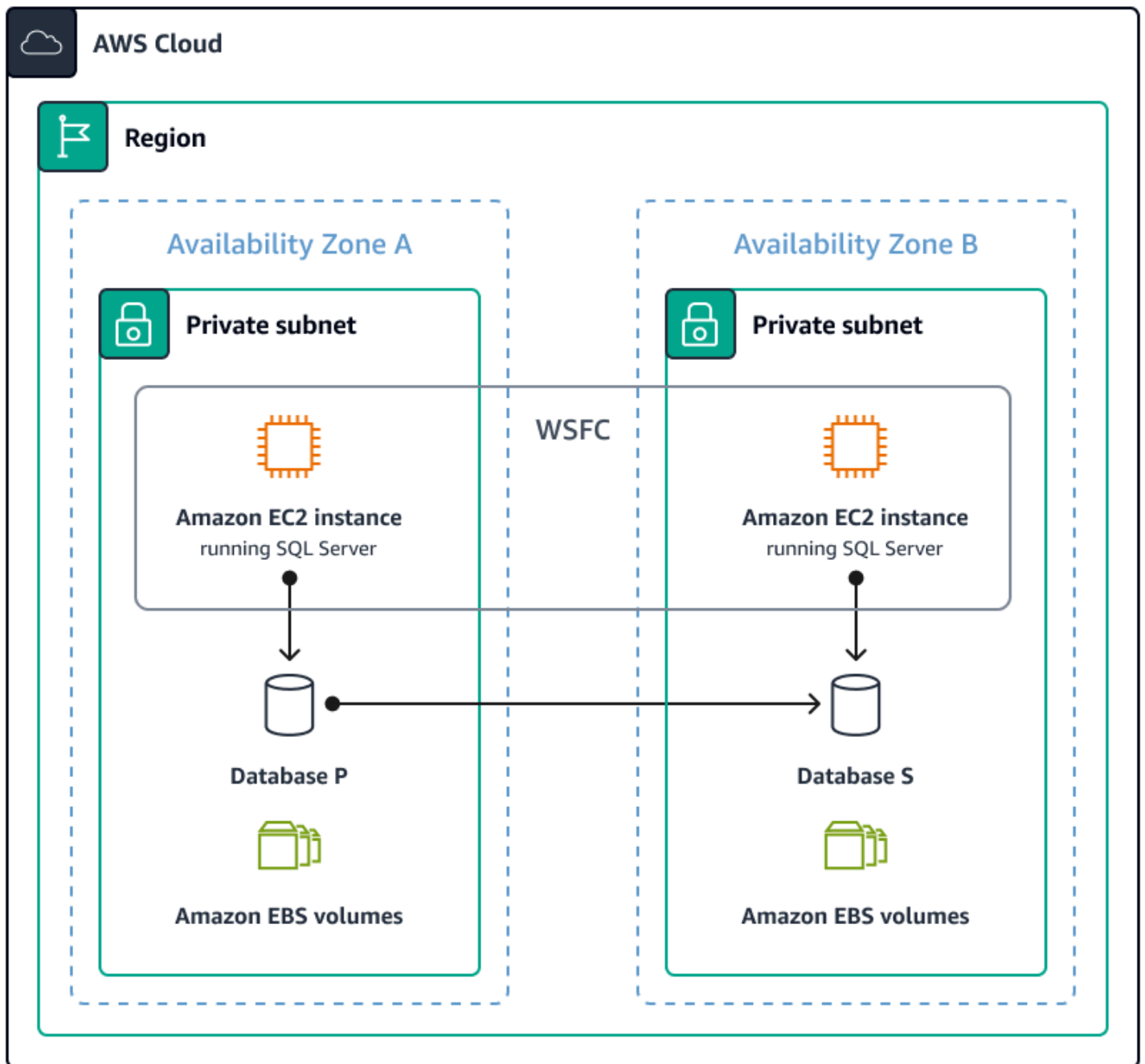
功能

- SQL Server Standard 版中可用
- 兩個複本的限制（主要和次要）
- 次要複本上沒有讀取存取權
- 次要複本上沒有完整性檢查

限制

- 每個可用性群組僅支援一個可用性資料庫
- 基本可用性群組不能是分散式可用性群組的一部分

下圖顯示 Windows Server 容錯移轉叢集解決方案的範例架構。



SQL Server Always On 容錯移轉叢集執行個體

您可以使用容錯移轉叢集執行個體（FCIs）來確保持續的資料庫操作，同時將停機時間降到最低，並降低資料遺失的風險。FCIs 如果您在沒有僅供讀取複本組態的情況下，為SQL伺服器資料庫尋找高可用性，會提供可靠的解決方案。

與可用性群組不同，FCIs可以提供可靠的容錯移轉解決方案，而無需 SQL Server Enterprise 版本。相反地，FCIs只需要 SQL Server Standard 版授權。您可以使用 FCIs 將SQL伺服器授權成本降低 65–75%。

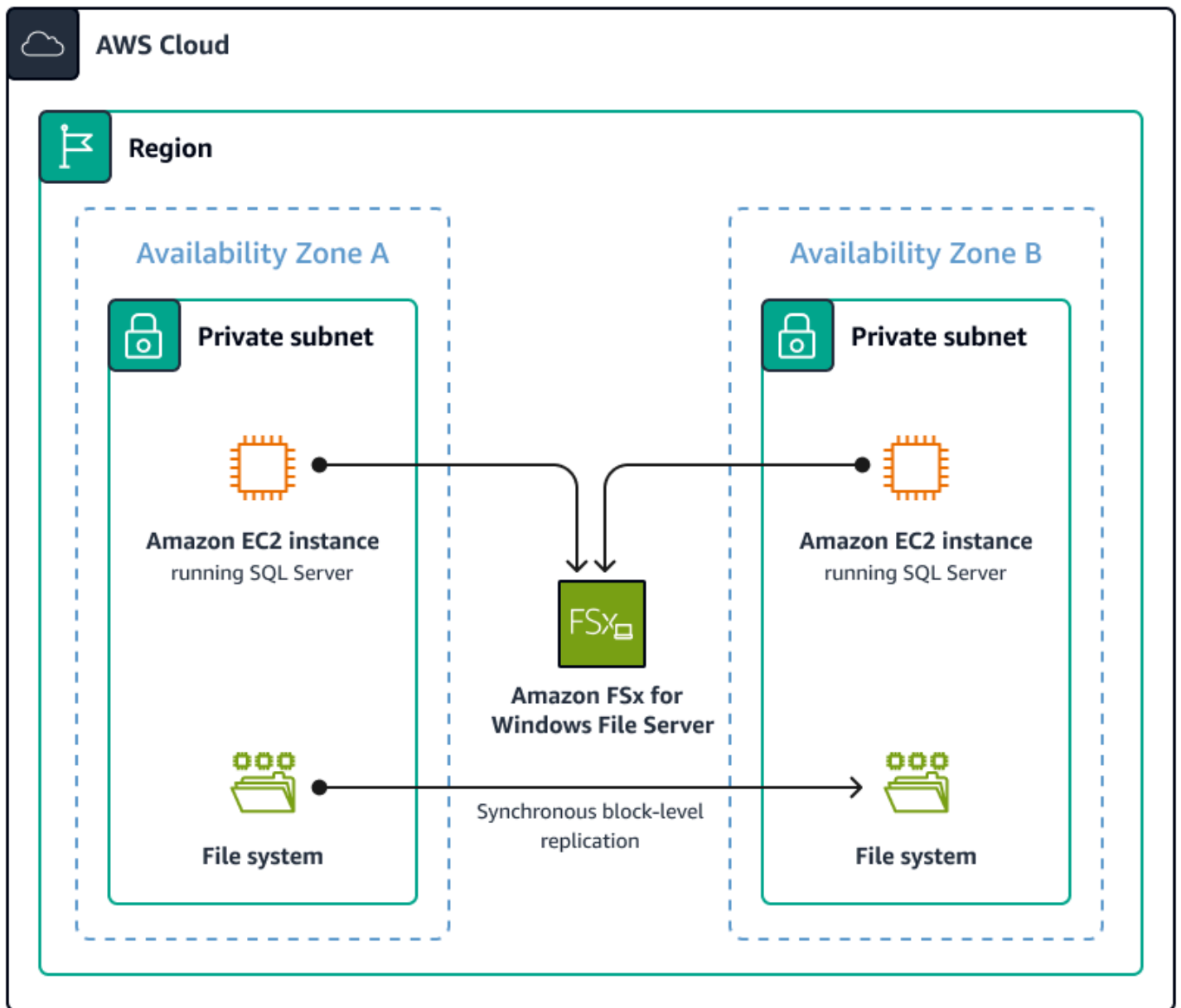
Note

如需有關SQL伺服器版本之間成本差異的詳細資訊，請參閱本指南的[比較SQL伺服器版本](#)一節。

考慮下列各項：

- Amazon FSx for Windows File Server 提供功能強大的解決方案，可滿足您的SQL伺服器FCI共用儲存需求。您可以使用 FSx for Windows File Server 來避免購買儲存體複寫解決方案的授權，並自行管理共用儲存體。這可以大幅節省 30-40% 的成本。如需詳細資訊，請參閱 AWS Storage Blog 上的[使用 Amazon FSx for Windows File Server 文章簡化 Microsoft Server SQL高可用性部署](#)。
- 透過[軟體保證利益摘要](#)（可下載PDF）和自攜授權（BYOL）模型，只要次要伺服器是被動的，您就可以利用被動容錯移轉利益。這可節省SQL授權的成本，因為您不必向叢集的被動節點提供授權。

下圖顯示FCI使用 FSx for Windows File Server 的SQL伺服器架構範例。



SIOS DataKeeper

如果您打算在 FCI 上部署 SQL 伺服器，建議您考慮共用儲存需求 AWS。傳統的內部部署安裝通常使用儲存區域網路（SAN）來滿足共用儲存需求，但這不是上的可行選項 AWS。Amazon FSx for Windows File Server 是 FCI 上建議的 SQL 伺服器儲存解決方案 AWS，但其限制會阻止在不同 中新增叢集伺服器 AWS 區域。

您可以使用 [SIOS DataKeeper](#) 來建立同時涵蓋可用區域和區域的 SQL 伺服器 FCI，同時將成本降低 58–71%。SIOS DataKeeper 可協助您實現的高可用性優勢 FCI。這 SIOS DataKeeper 為組織提供了具成本效益且可靠的解決方案。

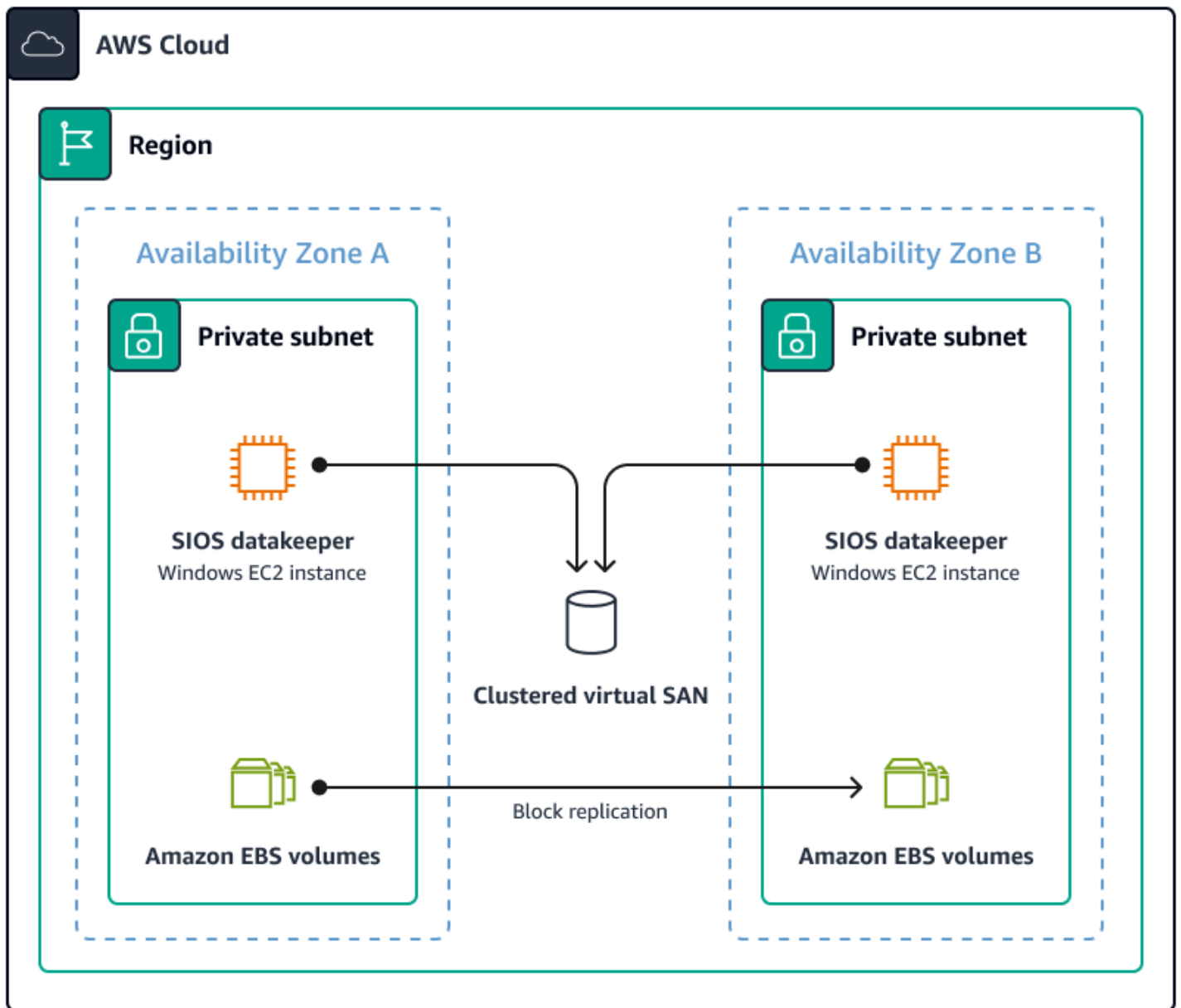
請考慮使用 的下列其他優點SIOS DataKeeper :

- SIOS DataKeeper SAN會使用本機EBS磁碟區建立叢集虛擬 ，並使用可用區域之間的不同步複寫來實現高可用性。對於災難復原，會在區域之間SIOS DataKeeper 使用非同步複寫。
- SIOS DataKeeper 使用 SQL Server Standard 版提供企業級叢集功能。相較於使用 SQL Server Enterprise 版的 SQL Server Always On 可用性群組實作高可用性，這可降低 65–75% 的SQL伺服器授權成本。透過 SIOS DataKeeper，您可以建立高可用性、彈性且符合成本效益的SQL伺服器環境，以符合組織的需求。

Note

如需有關SQL伺服器版本之間成本差異的詳細資訊，請參閱本指南的[比較SQL伺服器版本](#)一節。

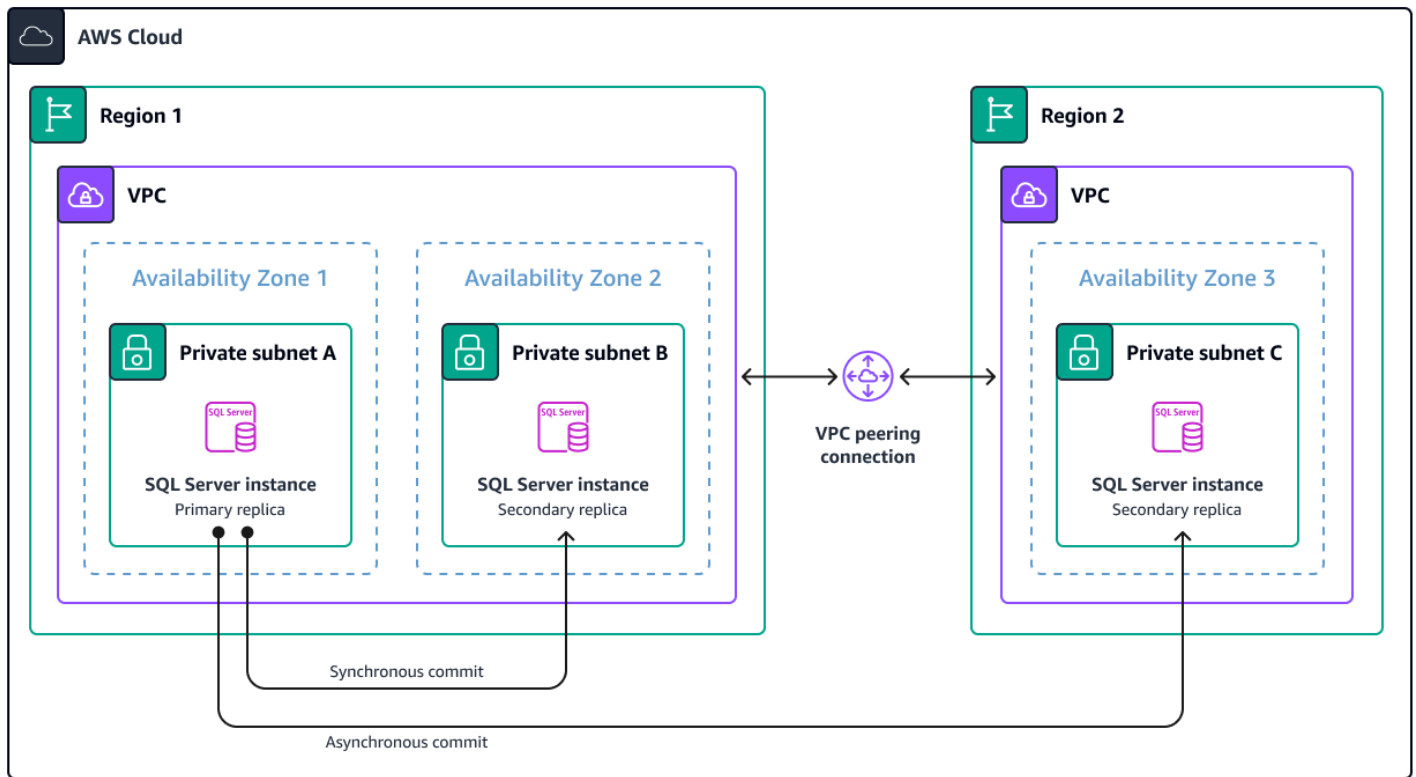
下圖顯示FCI使用叢集虛擬SAN解決方案之SQL伺服器的範例架構。



Always On 可用性群組

您可以使用 Always On 可用性群組來實現高可用性和災難復原。您可以透過在一個區域中的兩個可用區域部署 SQL 伺服器來實現高可用性。您可以透過跨區域擴展可用性群組來實現災難復原。

下圖顯示以 Always On 可用性群組為基礎的解決方案範例架構。圖表區域 1 中的複本使用同步 Commit，可提供可用性群組的自動容錯移轉。區域 2 中的複本正在使用非同步 Commit，這將需要可用性群組的手動容錯移轉。

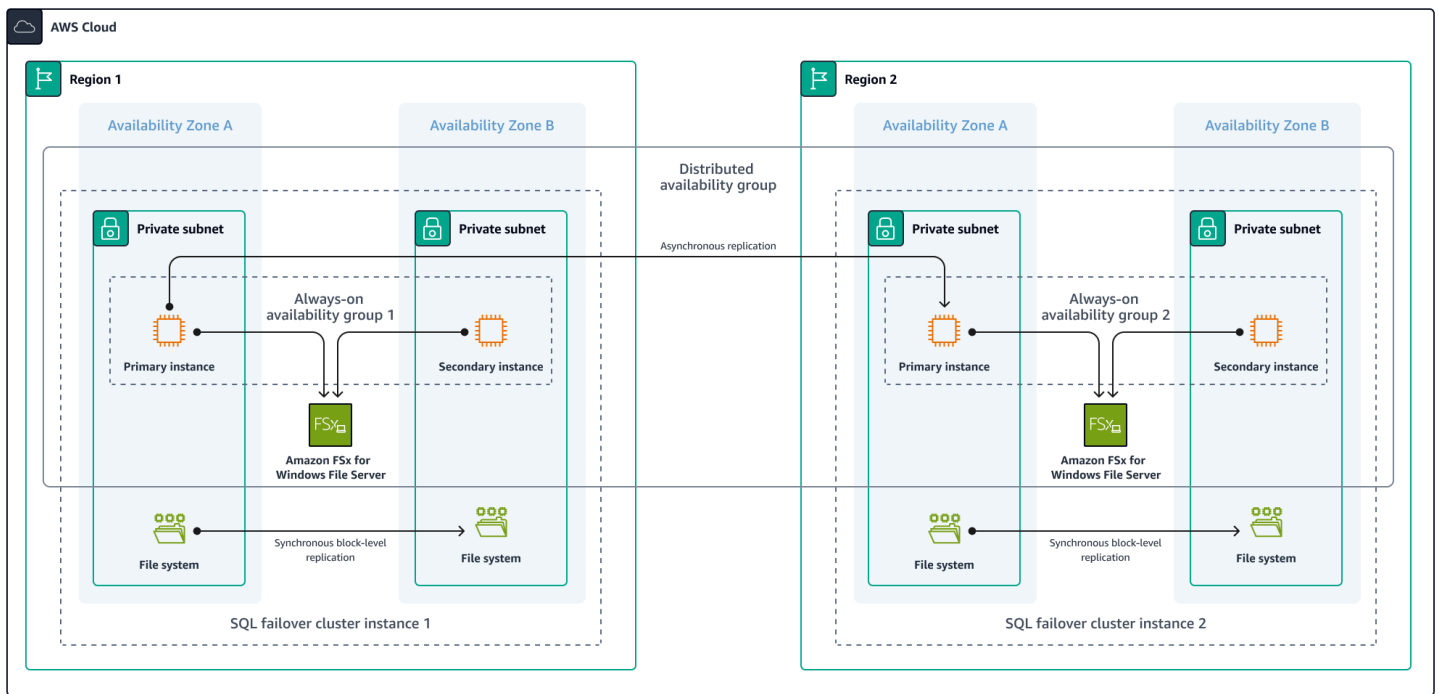


分散式可用性群組

對於您無法在可靠性或災難復原上妥協的任務關鍵SQL型伺服器部署，我們建議採用多區域方法。跨多個區域分發您的可用性群組是維持業務連續性和將停機時間降至最低的最有彈性解決方案。

此架構充分利用 Amazon FSx for Windows File Server 的功能，包括共用儲存、同步區塊層級複寫和 SQL 伺服器 FCIs。這些功能可讓您建立跨越多個可用區域的高可用性 SQL 伺服器環境。透過在另一個區域中複寫此設定，您就能取得完全備援的系統，即使最嚴重的中斷也能處理。此解決方案的區別在於其提供的彈性和安全性層級。分散式可用性群組的網域獨立架構可讓基礎 Windows 叢集伺服器加入不同的 Active Directory 網域，而憑證型身分驗證可確保 SQL 伺服器環境的最大保護，並提供多區域 DR 策略的高 RTORPO 需求。如需建置多區域架構的相關資訊，請參閱 [架構部落格中的欄位備註：使用 FCI 和分散式可用性群組建置 SQL 多區域伺服器 AWS 架構](#)。

下圖顯示使用分散式可用性群組的多區域解決方案的範例架構。



日誌傳送

日誌寄件是一種經過驗證、可靠且符合成本效益的方法，可在發生意外中斷時跨區域保護資料庫。幾十年來，組織一直使用日誌寄件來保護其資料。

如果您在實作日誌寄件 AWS，您可以根據交易頻率RPO和日誌寄件任務，在幾分鐘RTO內完成和。萬一區域變得無法存取，日誌寄件會確保您的資料安全且可復原。

請考慮使用日誌寄件的下列其他優點：

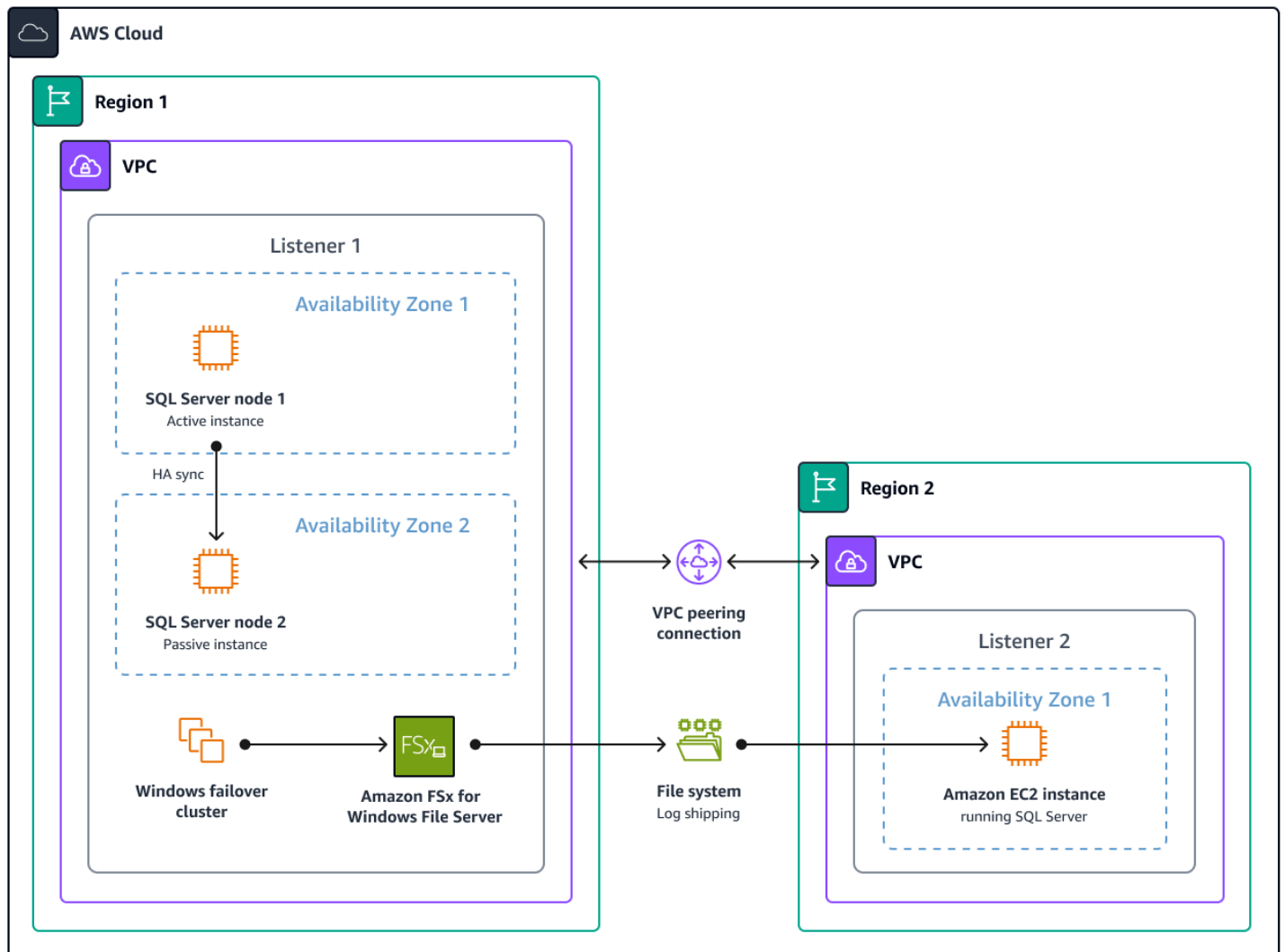
- 使用日誌寄件來降低跨區域的災難復原彈性，進而降低成本並滿足您的業務需求。日誌寄件會減少您的，TCO因為您只需要 SQL Server Standard 版或 SQL Server Web 版授權。
- 使用具有有效**軟體保證**的日誌寄件，從災難復原/被動伺服器移除授權成本。當您將日誌寄件與軟體保證搭配使用時，只需要授權主要/作用中SQL伺服器。
- 減少 65% 到 75% 的SQL伺服器授權成本，無需使用 SQL Server Enterprise 版在區域之間設定分散式可用性群組。您可以使用 SQL Server Standard 版本和 SQL Server FCIs結合日誌寄件來滿足您的災難復原需求。

Note

如需有關SQL伺服器版本之間成本差異的詳細資訊，請參閱本指南的[比較SQL伺服器版本](#)一節。

如需詳細資訊，請參閱 AWS 架構部落格中的[使用 FCI Amazon FSx for Windows 組態的 Server 日誌寄件來擴展SQL伺服器 SQL DR。](#)

下圖顯示日誌寄件解決方案的範例架構。

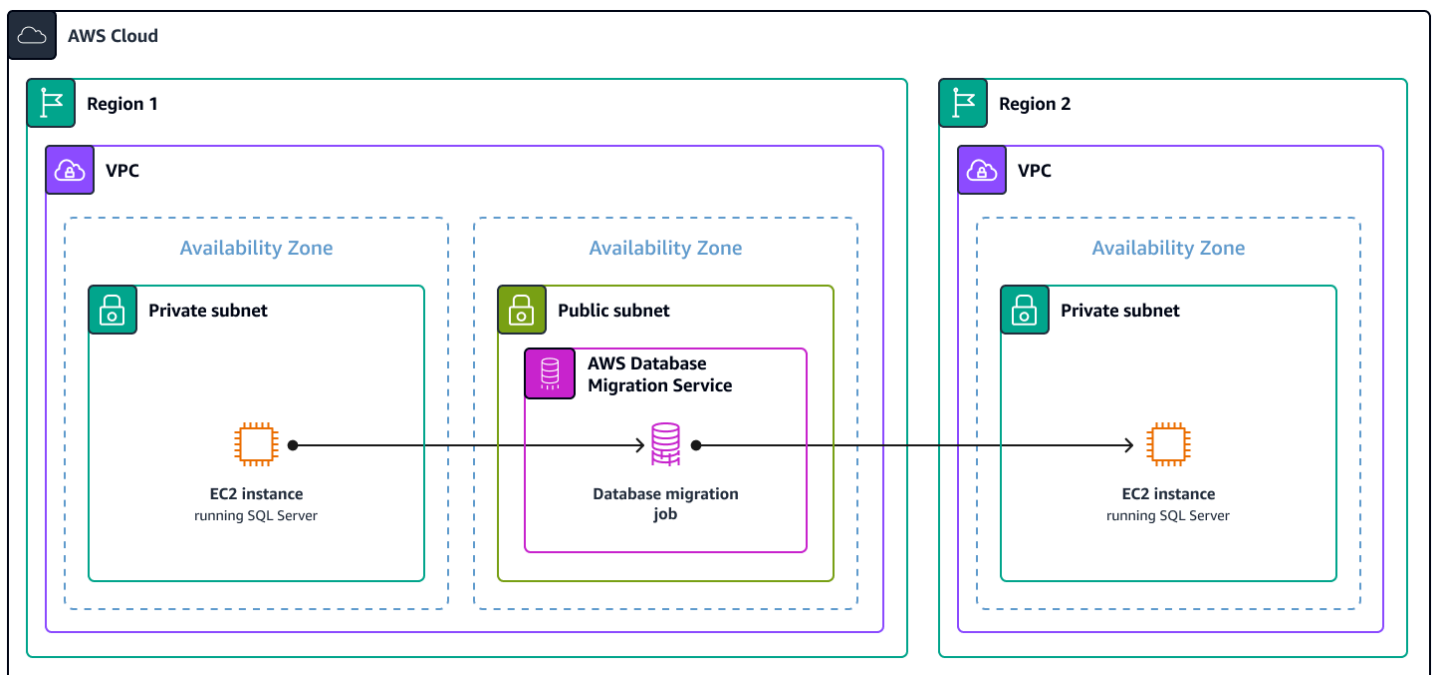


AWS Database Migration Service

您可以使用 AWS Database Migration Service (AWS DMS) 根據您的應用程式需求來設計 HA/DR 解決方案。AWS DMS 可讓您輕鬆將資料複製到相同區域 (HA) 或跨區域 (DR) 的次要SQL伺服器資料庫。這種方法在技術上健全，可讓您在最佳化資源用量的同時，最大化對 AWS 基礎設施的投資。

AWS DMS 是一項符合成本效益的服務。您只需支付傳輸程序和任何其他日誌儲存期間使用CPU的資源的費用。這表示您可以受益於此解決方案，而不會產生重大的額外成本。您可以使用 AWS DMS 來確保資料可用和可存取，同時將與授權和資源用量相關的成本降至最低。

下圖顯示以 為基礎的解決方案的範例架構 AWS DMS。



AWS Elastic Disaster Recovery

有些組織必須確保所有關鍵業務應用程式都已備妥災難復原計劃。過去，許多這些組織大量投資於傳統災難復原解決方案，這需要您預先建置和維護整個重複的基礎設施。這種方法昂貴、耗時且難以擴展。

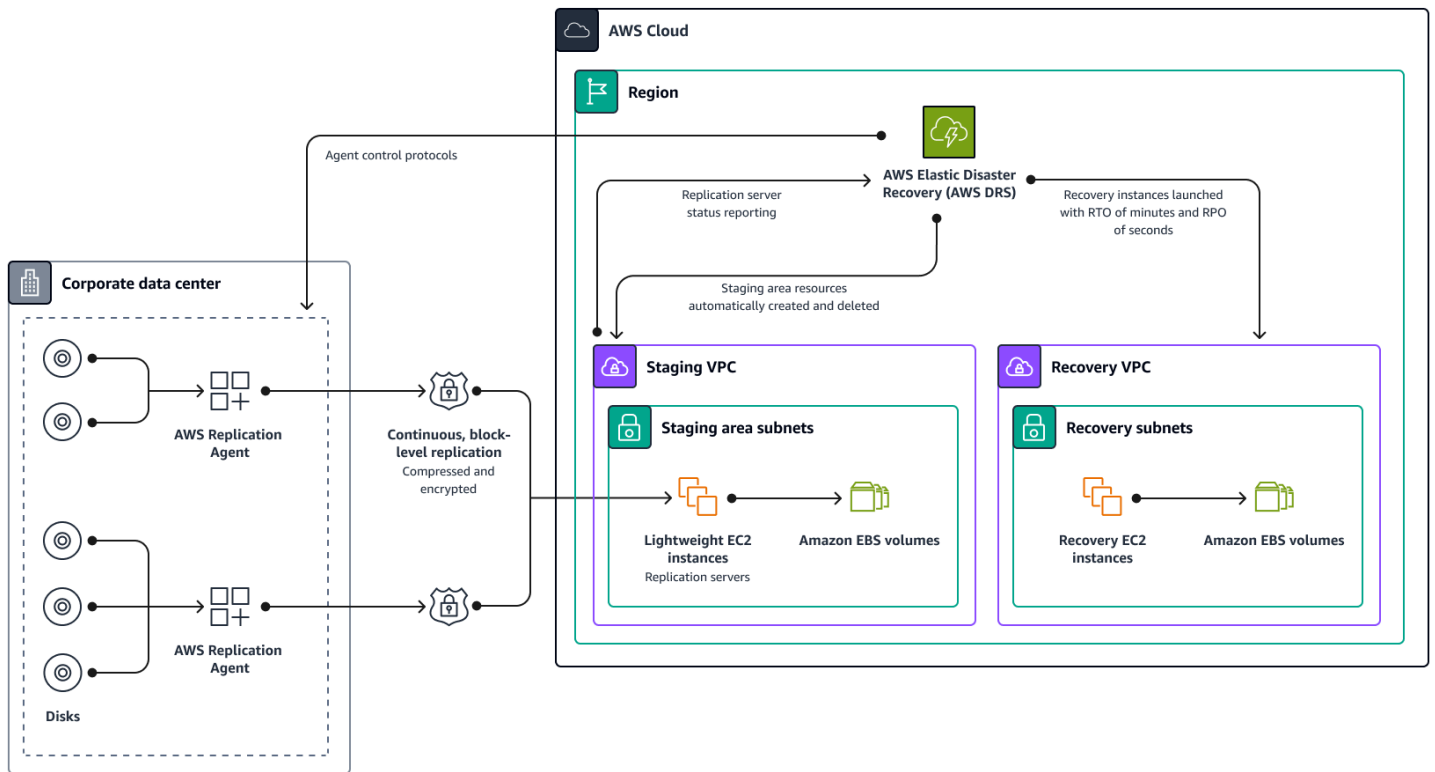
現在，您可以使用 AWS Elastic Disaster Recovery 來消除預先建置災難復原基礎設施的需求。災難復原機器不會在 Elastic Disaster Recovery 中啟動，除非必要，因此您只需為在需要時使用的項目支付費用。這表示您可以大幅降低軟體授權和高效能運算成本。

此外，災難復原解決方案的暫存區域包含低成本的 Amazon Elastic Block Store (Amazon EBS) 磁碟區。EBS 磁碟區進一步降低佈建重複資源的成本。這可讓您降低整體災難復原成本，同時仍能維持

強大且可靠的災難復原解決方案，以符合您的業務需求。您可以使用 Elastic Disaster Recovery 來專注於核心業務活動，同時 AWS 為災難復原解決方案處理基礎基礎設施。

對於SQL伺服器，您可以使用 Elastic Disaster Recovery 作為具成本效益的災難復原選項。如果您使用作用中的軟體保證，則會涵蓋容錯、高可用性SQL伺服器架構中被動節點的授權。不過，您仍然需要為被動伺服器線上支付運算成本。使用 Elastic Disaster Recovery，主要伺服器可以複寫到 DR 環境，而無需維護有效的軟體保證，也無需支付災難復原運算成本。這種節省的組合可降低 50% 以上的SQL伺服器災難復原成本。

下圖顯示以 Elastic Disaster Recovery 為基礎的解決方案的範例架構。



如需詳細資訊，請參閱[如何在部落格上的 Microsoft 工作負載上使用 為還原的 DR 網站設定SQL伺服器高可用性 AWS Elastic Disaster Recovery](#)。AWS

成本比較

下表比較本節中涵蓋的 HA/DR 解決方案成本。基於此比較的目的，會做出下列假設：

- 執行個體類型 – r5d.xlarge
- 授權類型 – 包含 Windows 和SQL伺服器的授權
- 區域 – us-east-1

解決方案	高可用性	災難復原	企業版	標準版本	成本
日誌傳送	否	是	是	是	SQL Server Enterprise 版本：\$32,674.8 (2 個節點) SQL Server Standard 版本：14,804.4 美元 (2 個節點)
Always On 可用性群組	是	是	是	是，但基本可用性群組 (2 個節點)	SQL Server Enterprise 版本：\$32,674.8 (2 個節點) SQL Server Standard 版本：14,804.4 美元 (2 個節點)
永遠開啟 FCIs	是	否	是	是 (2 個節點)	SQL Server Standard 版本：14,804.4 美元
分散式可用性群組	是	是	是	否	SQL Server Enterprise 版本：\$65,349.6 (4 個節點)

解決方案	高可用性	災難復原	企業版	標準版本	成本
彈性災難復原	否	是	是	是	<p>大約。每月 107.48 美元，用於複寫 1 個執行個體和 1 TB 儲存體</p> <p>注意：每個複寫伺服器都會每小時向 Elastic Disaster Recovery 收費。無論磁碟數量、儲存體大小、演習或復原啟動次數，或是您正在複寫的區域，成本都相同。</p>

解決方案	高可用性	災難復原	企業版	標準版本	成本
SIOS Data Keeper	是	是	是	是	具有軟體保證的 Always On 可用性群組 (2 個節點、24 個核心) : \$213,480 搭配 SIOS DataKeeper 和 軟體保證在 SQL Server Standard 版本上執行的 2 節點 SQL 伺服器叢集 : \$61,530 (2 個節點)
AWS DMS	否	是	是	是	r5.xlarge 執行個體和 1 TB 儲存體每月 745.38 美元

成本最佳化建議

建議您採取下列步驟，選擇符合組織需求的 HA/DR 解決方案：

- 檢閱本指南中的 [為 SQL 伺服器工作負載選取正確的 EC2 執行個體](#) 一節。
- 在尖峰工作負載期間執行效能計數器，判斷工作負載的 IOPS 和輸送量需求：
 - $IOPS = \text{磁碟讀取/秒} + \text{磁碟寫入/秒}$
 - $\text{輸送量} = \text{磁碟讀取位元組/秒} + \text{磁碟寫入位元組/秒}$
- 使用下列儲存磁碟區類型可提升效能並節省成本：

- NVMe tempdb和 緩衝集區延伸的執行個體儲存
- 資料庫檔案的 io2 磁碟區
- [AWS Trusted Advisor](#) 使用 針對 Amazon 上SQL伺服器的成本最佳化提出建議EC2。您不需要為 安裝代理程式 Trusted Advisor 以進行SQL伺服器最佳化檢查。 Trusted Advisor 會檢查您的 Amazon EC2 SQL Server 包含授權的執行個體組態，例如虛擬 CPUs (vCPUs)、版本和版本。然後，根據最佳實務 Trusted Advisor 提出建議。
- AWS Compute Optimizer 適用於 Amazon EC2執行個體和 Amazon EBS正確大小調整建議。
- 使用 [AWS Pricing Calculator](#) 設計您的 HA/DR 策略以進行成本估算。
- 若要判斷從 SQL Server Enterprise 版本降級為 SQL Server Standard 版本是否為可能的選項，請使用 [sys dm_db_persisted_sku_features](#) 動態管理檢視來識別目前資料庫中作用中的版本特定功能。

Note

Side-by-side 使用授權包含的EC2執行個體時，SQL伺服器版本變更需要遷移。

- 執行半年或年度災難復原演練，以更好地建構設計，以使用定義的 RTO和 來復原資料庫RPO。這也可以協助您識別任何架構弱點。

其他資源

- [使用 Amazon FSx for Windows File SQL Server 簡化 Microsoft Server 高可用性部署](#) (AWS 儲存部落格)
- [欄位備註：使用 FCI和分散式可用性群組建置多區域SQL伺服器架構](#) (AWS 架構部落格)
- 在 [上為SQL伺服器架構災難復原 AWS：第 1 部分](#) (AWS 資料庫部落格)
- [Amazon FSx for Windows 的 Microsoft SQL高可用性](#) (YouTube)
- [使用 Amazon 最大化 Microsoft SQL Server 效能 EBS](#) (AWS Storage Blog)
- [比較您的內部部署儲存模式與 AWS 儲存服務](#) (AWS 儲存部落格)
- [計劃NAS使用 Amazon FSx File Gateway 取代資料中心](#) (AWS 儲存部落格)
- 在 (AWS 儲存部落格) [上最佳化高可用性SQL伺服器部署的成本 AWS](#)
- [如何使用 \(上的 Microsoft 工作負載 \) 設定 SQL Server Always On 可用群組的災難復原 AWS Elastic Disaster Recovery AWS](#)
- [如何為使用 \(上的 Microsoft 工作負載 \) 還原的 DR 站台上的SQL伺服器設定高可用性 AWS Elastic Disaster Recovery AWS](#)

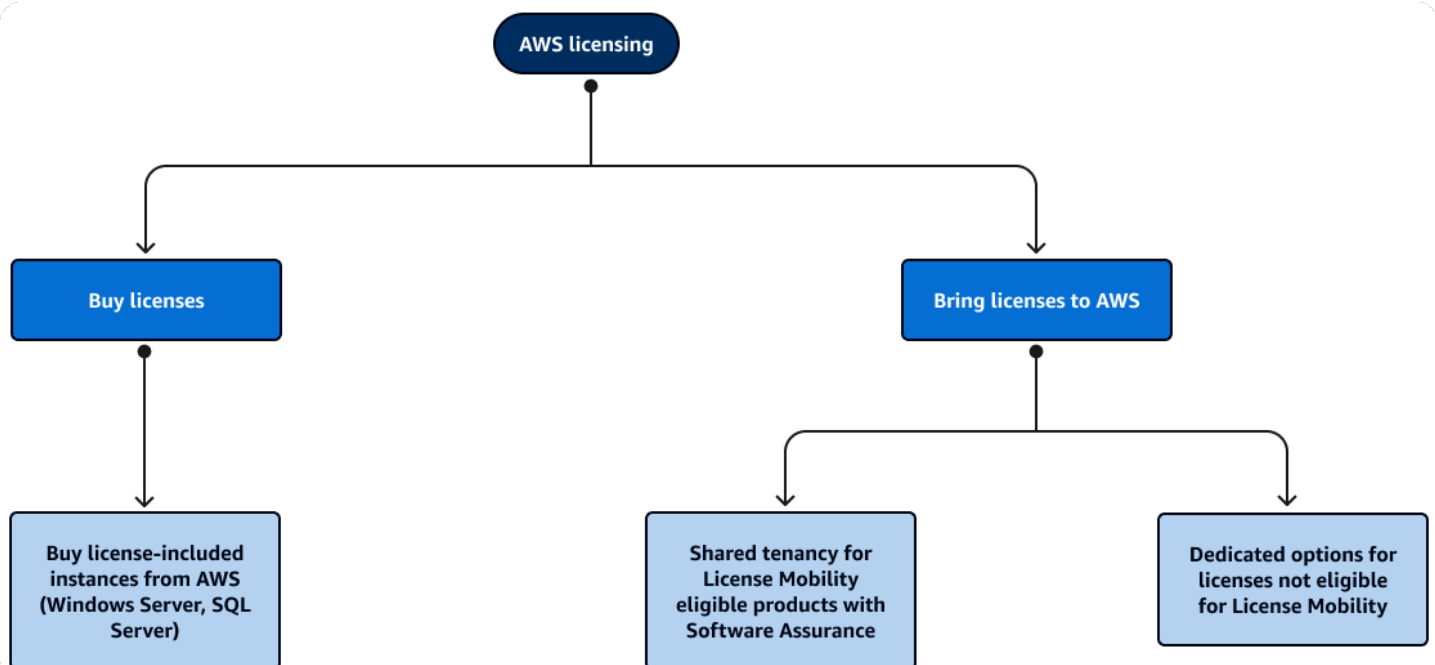
了解SQL伺服器授權

概觀

隨著越來越多的企業將工作負載移至雲端，雲端平台上的成本最佳化已成為首要任務。授權是與在上執行 Microsoft 工作負載相關聯的最重要成本之一 AWS。本節說明如何 AWS 透過最佳化 Microsoft License for SQL Server 來最佳化的成本。

AWS 授權選項

AWS 提供廣泛的彈性成本最佳化選項以供授權。這些授權選項旨在協助您降低成本、維持合規性，並滿足您的業務需求。



AWS 將授權分類為三種主要類型：

1. 包含授權 – 此授權選項可讓您隨需購買和使用授權，只需支付使用的費用。包含授權的選項非常適合您需要彈性使用授權，並希望避免預付成本的案例。您可以從一系列 Windows Server、SQL Server 和其他 Microsoft 產品中進行選擇。
2. Bring Your Own License (BYOL) 產品具有授權行動性 – 此授權選項專為您已擁有現有授權，並希望在雲端使用這些授權的案例而設計。AWS 可讓客戶透過 Microsoft 的 [License Mobility](#) 方案，將自己的授權帶到雲端。您可以將具有授權行動性的產品，例如具有軟體保證 (SA) 的 SQL 伺服器，帶到共用或專用租用，以降低 AWS 執行個體成本。

3. BYOL 沒有授權行動性的 產品 – 對於沒有授權行動性的 Microsoft 產品，例如 Windows Server，AWS 提供專用選項以在雲端使用這些產品。此外，專用主機提供在實體核心層級進行授權的機會。這可為您節省 50% 或更多執行工作負載所需的授權。對於大多數時間執行的穩定且可預測的工作負載，專用主機是很好的選擇。

取得授權的成本影響

攜帶授權可能會對在上執行 Microsoft 工作負載的成本產生重大影響 AWS。如果您自帶授權，則不需要為在雲端中執行的執行個體支付額外的授權成本。這可以大幅節省成本。

下列比較顯示執行單一 c5.xlarge 執行個體 24/7 的隨需每月成本：

- Windows Server + SQL Server Enterprise 版本：每月 1353 美元（包含授權）
- Windows Server + SQL Server Standard 版：\$609/月（包含授權）
- 僅限 Windows Server：\$259/月（包含授權）
- 僅限運算（Linux）：每月 127 美元

最後，自帶授權可能會對在上執行 Microsoft 工作負載的成本產生重大影響 AWS。如果您使用現有的授權，可以降低授權成本，並節省整體 AWS 帳單的費用。

授權最佳化

AWS 最佳化和授權評估（AWS OLA）可以降低運算和授權成本，協助您最佳化授權。AWS OLA 旨在評估您在遷移計劃工作負載上執行的工作負載 AWS 或工作負載的授權需求。AWS OLA 提供最佳化授權使用的建議。

最佳化授權用量的其中一個關鍵策略是[正確調整執行個體大小](#)。正確調整大小涉及根據工作負載的 CPU、記憶體和儲存需求來選擇正確的執行個體類型。透過選擇適當的執行個體大小，您可以確保以具成本效益的方式使用資源。這可以大幅節省成本。

使用 Microsoft 軟體授權時，軟體執行的核心數量是決定授權成本的關鍵因素。例如，Windows Server 和 SQL Server 授權通常以核心數目進行授權。透過正確調整執行個體的大小，您可以降低 Microsoft 軟體執行的核心數量，進而降低執行個體的成本和所需的授權數量。

成本最佳化建議

最佳化授權是上成本最佳化的關鍵元件 AWS。透過實作正確的策略，您可以降低授權成本、維持合規性，並從您的授權投資中獲得最佳價值。本節概述授權最佳化的多種策略。

攜帶您的合格 Windows Server 授權

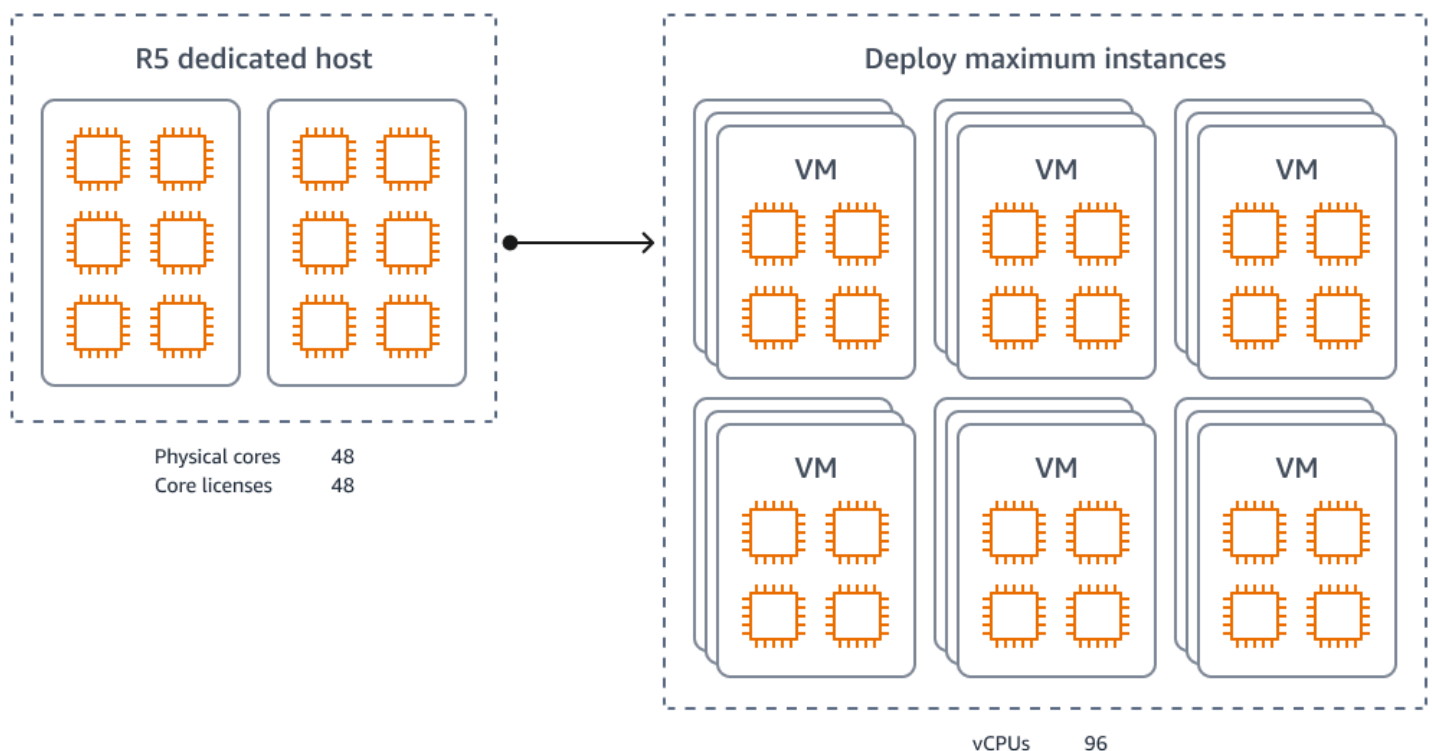
自備 Windows Server 授權是授權最佳化的最有效策略之一。此策略可讓您利用現有的投資來減少 AWS 支出。

例如，如果您在 1/10/2019 之前購買授權，或根據在該日期之前簽署的作用中 Enterprise Agreement 以校正形式購買授權，則可以在 [Amazon EC2 專用主機](#) 上部署 Windows Server 2019 和更舊版本。此規則是基於 Microsoft 在 2019 年針對沒有 License Mobility 的產品，例如 Windows Server，在 [上市提供者](#)（例如 Alibaba AWS 或 Google Cloud）上部署時，對其授權條款和條件所做的變更。根據新條款，您無法將自己的 Windows Server 授權帶到，AWS 但必須使用包含授權的執行個體。不過，如果您在該日期之前購買永久授權，您仍然可以在 Amazon EC2 專用主機上部署這些 Windows Server 授權。

實體層級授權

在實體核心層級授權可讓您僅授權主機的實體核心，因此您可以部署最多數量的執行個體，而不會影響所需的授權數量。這通常透過使用 Windows Server Datacenter 和 SQL Server Enterprise 版本來完成。

例如，請考慮具有 48 個核心的 R5 專用主機，其轉換為 96 個 vCPUs。如果您使用 Windows Server Datacenter 版本，則只需要 48 個授權。這可讓您部署最多 96 個的執行個體組合 vCPUs，如下圖所示。



如果您有足夠的工作負載來最大化可在主機上執行的執行個體數量，則此方法可能特別符合成本效益。透過在實體核心層級進行授權，您可以避免每個執行個體的額外授權成本，並為您的授權投資實現最佳價值。

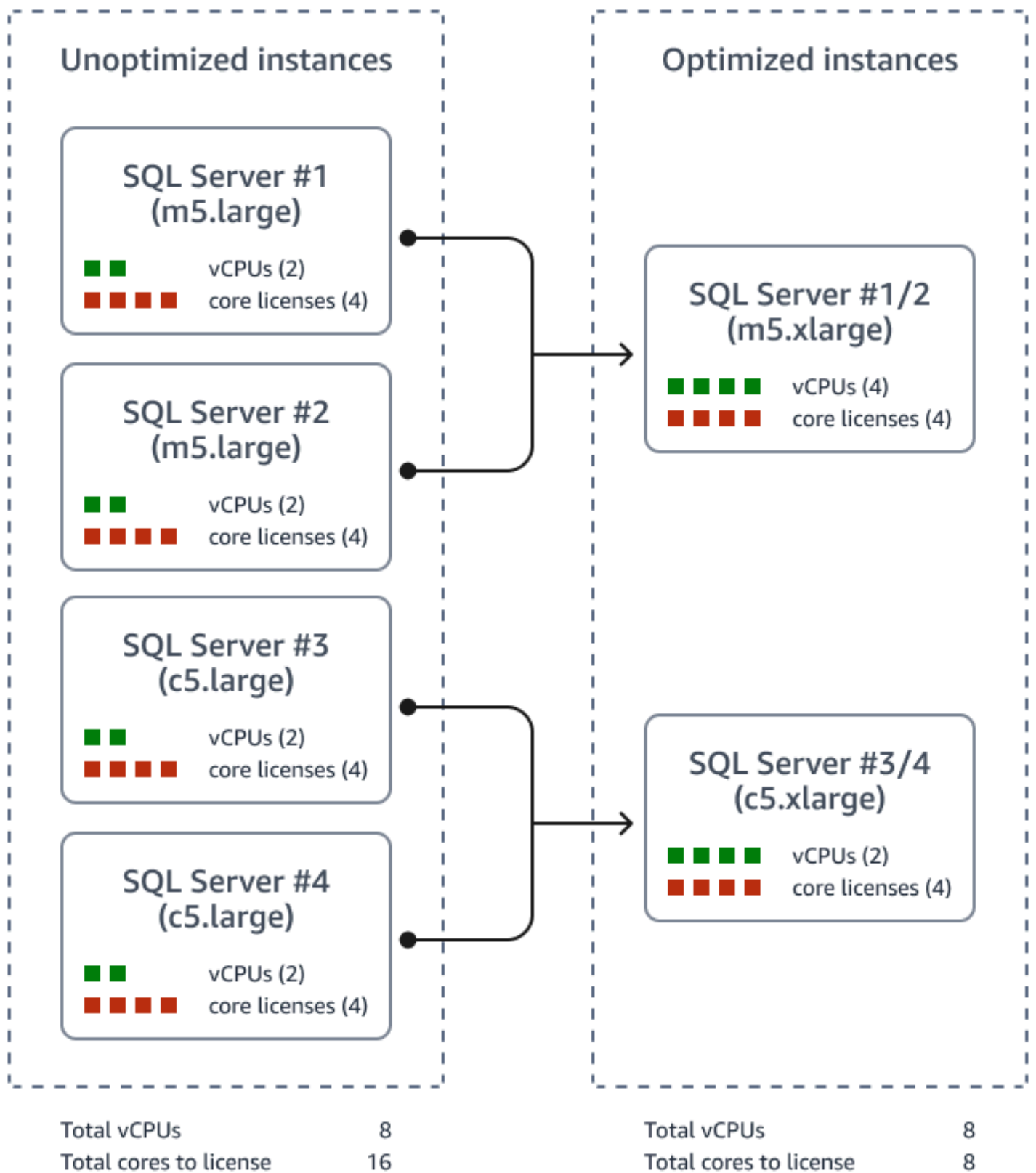
在SQL伺服器實體核心層級的授權

在共用租用中，SQL伺服器授權是以vCPUs配置給執行個體的數目為基礎。相反地，使用專用主機，您可以在實體核心層級或vCPU層級授權SQL Server Enterprise版本。

與先前的R5專用主機範例一樣，如果您在實體核心層級授權SQL Server Enterprise版，則只需要48個SQL Server Enterprise版授權即可授權主機。相反地，在共用租用中，只有v授權的選項CPU，您必須擁有相同工作負載的96個SQL Server Enterprise Edition授權。因此，相較於共用租用，專用主機最多可為您節省50%的SQL伺服器授權成本。這是除了透過攜帶合格的Windows授權節省執行個體成本之外。

合併SQL伺服器執行個體

[SQL 伺服器合併](#)是將多個SQL伺服器執行個體合併到一個伺服器的程序。SQL伺服器要求每個執行個體至少四個核心授權，即使執行個體只有兩個vCPUs。這表示在少於四個核心的伺服器上執行SQL伺服器可能會導致您過度授權這些執行個體，並在必要時使用更多授權。



例如，將兩個執行個體合併為兩個執行個體 vCPUs，每個執行個體合併為四個執行個體，vCPUs 可減少 50% 的授權需求。這是因為只需要四個核心授權，而不是八個。

如需整合的詳細資訊，請參閱本指南的[SQL 伺服器整合](#)一節。

降級 SQL 伺服器版本

[變更 SQL 伺服器版本](#)可以是最佳化授權用量和降低成本的關鍵策略。企業版的 SQL 伺服器比標準版貴得多，因此降級可以大幅節省成本。

透明資料加密 (TDE) 和 Always On 可用性群組是 SQL Server Enterprise 版中的兩個常用功能。不過，如果您不需要 SQL Server Enterprise Edition 的完整功能集，您可以考慮這些功能具有成本效益的替代方案。例如，您可以從 SQL Server 2019 開始 TDE 進入 SQL Server Standard 版本。取代 Always On 可用性群組，您可以在 FSx for Windows File Server 上使用容錯移轉叢集與共用儲存，以獲得 SQL Server Standard 版本的高可用性。

透過從 SQL Server Enterprise 版本降級為 SQL Server Standard 版本，您可以大幅降低授權成本。如需詳細資訊，請參閱 AWS Storage Blog 上文章的[高可用性 SQL 伺服器部署最佳化成本 AWS](#)。

除了降低授權成本之外，降級 SQL 伺服器版本也有助於減少軟體保證支出，並協助您避免未來的校正。如果您將未使用的授權退回到貨架，您可以避免額外的授權成本，並從您的授權投資中獲得最佳價值。

請務必仔細評估您的 SQL 伺服器工作負載，並判斷哪些功能對您的業務需求至關重要。如需詳細資訊，請參閱 AWS 在規範指南中[評估您的環境](#)，並判斷您的 Microsoft SQL Server 資料庫是否使用 SQL Server Enterprise 版本特定的功能。

如果您選擇正確的 SQL 伺服器版本，並使用 SQL Server Enterprise 版本功能的替代方案，您可以節省大量成本，同時保持合規性並滿足您的業務需求。如需降級選項的詳細資訊，請參閱本指南的[比較 SQL 伺服器版本](#)一節。

在非生產環境中使用 SQL Server Developer Edition

在非生產環境中，您可以在內部部署環境中使用 MSDN 訂閱來部署可授權的 SQL 伺服器版本，例如 Enterprise 或 Standard 版本。不過，MSDN 訂閱沒有 License Mobility。因此，如果您遷移至 AWS，則無法將這些授權帶過來。您必須改用 SQL Server Developer Edition。

SQL Server Developer Edition 是全功能版的 SQL Server，免費提供。此版本適用於 SQL Server 2016 版及更新版本。您可以從 Microsoft 網站下載。SQL 伺服器開發人員版本旨在用於所有非生產環境，例如開發、測試和預備，只要它未連接到即時生產資料。

如果您在非生產環境中使用 SQL Server Developer Edition，可以避免額外的授權成本。如需詳細資訊，請參閱本指南的[評估SQL伺服器開發人員版本](#)一節。

針對SQL伺服器工作負載CPU最佳化

在某些情況下，由於其他因素，例如 RAM 或網路限制，您可能需要選擇 CPUs 超過工作負載所需數量的執行個體類型。不過，AWS 提供解決方案，協助您在這些情況下最佳化授權成本。

您可以像大多數擁有 SQL 伺服器核心授權的客戶一樣，停用超執行緒或關閉 EC2 執行個體 CPUs，以限制 CPUs 主機可用的數量。此選項可讓您利用其他執行個體功能，例如 RAM，同時節省購買額外授權的成本。只有在使用時才能使用此選項，而且如果您使用的是 SQL 伺服器或 Windows Server 隨附的授權，BYOL 則無法使用此選項。

例如，如果您部署 r5.4xlarge 執行個體，因為工作負載需要 128 GB 的記憶體，但只需要八個 SQL 伺服器核心，則您可以在執行個體啟動時停用超執行緒，只需要八個作用中 CPUs。如此一來，您就可以在必要的 SQL 伺服器授權上節省 50% 的費用，因為您只需要授權目前正在使用的八個核心。

執行個體類型	總計 vCPUs	具備最佳化 CPUs 功能的 Active vCPU	SQL 伺服器授權節省
r5.4xlarge	16	8	50%
r5.12xlarge	48	8	83%

如果您將最佳化 CPUs 功能與包含授權的 Windows Amazon Machine Images (AMIs) 搭配使用，您仍然必須支付所有核心的 Windows 授權。因此，授權執行個體並定期重新評估執行個體類型非常重要。

如果您授權執行個體，可以確保為工作負載使用最具成本效益的執行個體類型。隨著 AWS 推出新的執行個體類型，評估這些新執行個體是否能夠以較少的核心滿足工作負載需求非常重要。

其他資源

- [Amazon Web Services 和 Microsoft：常見問題](#) (AWS 文件)

為SQL伺服器工作負載選取正確的EC2執行個體

⚠ Important

閱讀本節之前，建議您先閱讀本指南的[了解SQL伺服器授權](#)，並為[Windows 工作負載選取正確的執行個體類型](#)。

概觀

Microsoft SQL Server 已在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上執行超過 15 年。AWS 已利用該經驗協助開發 Amazon EC2執行個體，以適應從最低規格到高效能多區域叢集執行的SQL伺服器工作負載。

選擇正確的SQL伺服器EC2執行個體，主要取決於您的工作負載。了解SQL伺服器如何獲得授權、如何使用記憶體，以及SQL伺服器功能如何與 Amazon EC2產品保持一致，有助於引導您找到應用程式的最佳EC2執行個體。

本節說明各種SQL伺服器工作負載，以及如何與特定EC2執行個體配對，以將授權和運算成本降至最低。

成本比較

Amazon EC2可讓您攜帶自己的授權 (BYOL) 或隨著 Windows Server 和 SQL Server 授權付費。對於 pay-as-you-go 授權，Windows Server 和 SQL Server 授權的授權成本會依EC2執行個體的每小時成本來計算。例如，您可以AMI有不同價格的不同。的價格AMI取決於 AMI 執行的SQL伺服器版本。

Windows Server 和 SQL Server 定價不會逐項列出。您在 [等工具](#)上找不到分項定價[AWS Pricing Calculator](#)。如果您選取不同的授權包含方案組合，則可以降低授權成本，如下表所示。

EC2 執行個體	AMI	計算價格	Windows 授權價格	SQL 授權價格	總價
r5.xlarge	Linux (運算定價)	183.96 美元	-	-	183.96 美元
r5.xlarge	Linux + SQL 開發人員	183.96 美元	0 USD	0 USD	183.96 美元

EC2 執行個體	AMI	計算價格	Windows 授權價格	SQL 授權價格	總價
r5.xlarge	Windows Server (LI)	183.96 美元	134.32 美元	-	318.28 美元
r5.xlarge	Windows + SQL開發人員	183.96 美元	134.32 美元	0 USD	318.28 美元
r5.xlarge	Windows + SQL Web (LI)	183.96 美元	134.32 美元	49.64 美元	367.92 美元
r5.xlarge	Windows + SQL 標準 (LI)	183.96 美元	134.32 美元	350.4 美元	\$668.68
r5.xlarge	Windows + SQL 企業版 (LI)	183.96 美元	134.32 美元	1095 美元	1413.28 美元

Note

上表中的定價是以 us-east-1 區域中的隨需定價為基礎。

執行 SQL Server 最符合成本效益的方法，是保持在較低層級的版本，直到您需要更高層級版本的功能為止。如需詳細資訊，請參閱本指南的[比較SQL伺服器版本](#)一節。從 SQL Server Web Edition 升級至 SQL Server Standard Edition 的費用超過SQL伺服器授權成本的七倍，以及從 Standard Edition 移至 Enterprise Edition 的費用超過三倍。授權成本的差異是需要考慮的主要因素，並在本節的其餘部分中探索。

成本最佳化案例

請考慮分析公司追蹤交付工具正在尋求改善其SQL伺服器效能的範例案例。MACO 專家檢閱公司的效能瓶頸後，公司會從 x1e.2xlarge 執行個體轉換為 x2iedn.xlarge 執行個體。雖然執行個體大小較小，但對 x2 執行個體的增強功能會使用緩衝集區延伸來改善SQL伺服器效能和最佳化。這可讓公司從 SQL

Server Enterprise 版本降級為 SQL Server Standard 版本，並將其 SQL 伺服器授權從 8 降級 vCPUs 為 4vCPUs。

最佳化之前：

Server	EC2 執行個體	SQL 伺服器版本	每月成本
產品DB1	x1e.2xlarge	Enterprise	\$3,918.64
產品DB2	x1e.2xlarge	Enterprise	\$3,918.64
總計			7,837.28 美元

最佳化後：

Server	EC2 執行個體	SQL 伺服器版本	每月成本
產品DB1	x2iedn.xlarge	標準	\$1,215.00
產品DB2	x2iedn.xlarge	標準	\$1,215.00
總計			\$2,430.00

從 x1e.2xlarge 執行個體到 x2iedn.xlarge 執行個體的合併變更，可讓範例客戶每月在其生產資料庫伺服器上節省 5,407 美元。這將工作負載的總成本降低了 69%。

Note

上表中的定價是以 us-east-1 區域中的隨需定價為基礎。

成本最佳化建議

記憶體最佳化執行個體

SQL 伺服器最重要的層面之一是了解其對記憶體的依賴。SQL 伺服器嘗試使用作業系統 RAM 未使用的所有可用（預設安裝最多 2 TB）。它出於效能原因這樣做。使用記憶體中的資料比必須不斷從磁碟提

取資料、進行變更，然後將其寫入磁碟的效能更高。反之，SQL 伺服器會嘗試從連接的資料庫盡可能載入最多的資料，並將該資料保留在中 RAM。對資料所做的變更會在記憶體中發生，並在稍後硬化至磁碟。

Note

如需 SQL 伺服器如何寫入變更的詳細說明，請參閱 Microsoft 文件中的 [寫入頁面](#)。

由於 SQL 伺服器在大量中表現較佳 RAM，因此通常建議從 [Amazon EC2 記憶體最佳化](#) 執行個體類型開始。記憶體最佳化執行個體具有多種用途，並提供各種不同的選項。R 系列具有 1 比 8 vCPU-to-RAM 的比率，並可選擇 Intel 處理器、AMD 處理器、增強型聯網、增強型 EBS 效能、執行個體儲存和增強型處理器速度。對於記憶體繁重的工作負載，也有一個 X 系列結合了許多相同的選項，並將 vCPU-to-RAM 比率擴展到 1 到 32。由於記憶體最佳化執行個體的多樣性，您可以將它們套用至所有形狀和大小的 SQL 伺服器工作負載。

低於最低資源的工作負載（少於 4vCPUs）

雖然某些使用案例與爆量式（T3）執行個體運作良好，但我們建議您一般避免將爆量式執行個體用於 SQL 伺服器工作負載。SQL Server 的授權是以 vCPUs 指派給執行個體的數目為基礎。如果 SQL 伺服器一天的大部分時間都處於閒置狀態，且正在取得爆量額度，則您要為未充分利用的 SQL 使用權付費。此外，SQL 伺服器對於每部伺服器有 4 個核心的最低授權需求。這表示，如果您的 SQL 伺服器工作負載不需要 4 vCPUs 值的運算能力，則需支付您未使用的 SQL 伺服器授權。在這些情況下，最好將 [多個 SQL 伺服器執行個體合併](#) 到較大的伺服器上。

使用最少資源的工作負載（小於 64 GB RAM）

許多 64 GB 以下的 SQL 伺服器工作負載 RAM 不會優先考慮高效能或高可用性。對於這些類型的工作負載，如果應用程式涵蓋在 Microsoft 的授權限制範圍內，SQL 則伺服器 Web 版本可能非常適合。

Important

SQL 根據 Microsoft 的授權條款，伺服器 Web 版本具有受限制的使用案例。SQL 伺服器 Web 版本只能用於支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務。它可能無法用於支援 line-of-business 應用程式（例如，客戶關係管理、企業資源管理和其他類似應用程式）。

SQL 伺服器 Web 版本可擴展至 32 vCPUs 和 64 GB，RAM 比 SQL Server Standard 版便宜 86%。對於低資源工作負載，使用 AMD 記憶體最佳化執行個體，例如 r6a，其運算價格比 Intel 便宜 10%，也是將運算和 SQL 授權成本保持在最低水準的好方法。

具有平均資源的工作負載（小於 128 GB RAM）

SQL Server Standard 版用於大多數高達 128 GB 的 SQL 伺服器工作負載 RAM。SQL Server Standard 版比 SQL Server Enterprise 版便宜 65–75%，且可擴展至 48 vCPUs 和 128 GB RAM。由於 128 GB RAM 限制通常在 48 vCPU 限制之前達到，因此大多數想要避免升級至 SQL Server Enterprise 版本的客戶都以其為重。

SQL 伺服器具有稱為 [緩衝集區延伸](#) 的功能。此功能可讓 SQL 伺服器使用磁碟的一部分作為的延伸 RAM。緩衝區集區擴充功能與超快速儲存搭配使用時運作良好，就像 [Amazon EC2 執行個體儲存體 NVMe SSDs](#) 中使用的一樣。包含執行個體儲存體的 Amazon EC2 執行個體會以執行個體名稱（例如 r5d、r6id 和 x2iedn）中的 "d" 表示。

緩衝區集區延伸無法取代一般 RAM。不過，如果您需要超過 128 GB 的 RAM，則可以將緩衝集區延伸項目與 r6id.4xlarge 和 x2iedn.xlarge 等 EC2 執行個體搭配使用，以延遲升級至 Enterprise Edition 授權。

高效能工作負載（小於 128 GB RAM）

SQL 需要高效能的伺服器工作負載由於依賴許多資源，因此成本最佳化具有挑戰性。不過，了解 EC2 執行個體的差異可能會讓您無法做出錯誤的選擇。

下表顯示各種記憶體最佳化 EC2 執行個體及其效能限制。

	r5b	r6idn	r7iz	x2iedn	x2iezn
處理器	3.1 GHz	3.5 GHz	3.9 GHz	3.5 GHz	4.5 GHz
	第 2 代 Intel Xeon 處理器	第 3 代 Intel Xeon 處理器	第 4 代 Intel Xeon 可擴展處理器	第 3 代 Intel Xeon 處理器	第 2 代 Intel Xeon 處理器
CPU : RAM 比率	1 : 8	1 : 8	1 : 8	1 : 32	1 : 32
最大 vCPU	96	128	128	128	48

	r5b	r6idn	r7iz	x2iedn	x2iezn
最大值 RAM	768 GB	1,024 GB	1,024 GB	4,096 GB	1,536 GB
執行個體儲存體	–	NVMe SSD (4 個 1900 GB)	–	NVMe SSD (2 個 1900 GB)	–
io2 Block Express	支援	支援	支援	支援	–
最大值 EBS IOPS	260,000	350,000	160,000	260,000	80,000
最大EBS輸送量	60 Gbps	80 Gbps	40Gbps	80 Gbps	19 Gbps
網路頻寬上限	25 Gbps	200 Gbps	50 Gbps	100 Gbps	100 Gbps

每個執行個體都用於不同的用途。了解您的SQL伺服器工作負載可協助您選擇最適合您的執行個體類型。

屬性的詳細資訊：

- r5b – r5b 中的「b」屬性表示此執行個體類型專注於EBS高效能。在第五代記憶體最佳化執行個體中，r5b 是首選。這是第一個使用 io2 Block Express 磁碟區並達到 260,000 個儲存上限IOPS的執行個體類型。r5b 執行個體類型仍然是符合成本效益的替代方案，可滿足EBS高效能需求。
- r6idn – 第六代的記憶體最佳化執行個體提供比上一代大幅改善。r5b EBS的效能增強功能會進一步採用 r6idn，將最大值提升IOPS至 350,000。r6idn 也具有 tempdb 和緩衝集區擴充功能的執行個體存放區磁碟區，可進一步提升SQL伺服器效能。
- x2iedn – x2iedn 類似於 r6idn。它提供類似層級的增強型 EBS、增強型聯網和NVMeSSD執行個體儲存，但高記憶體工作負載和低CPU數量（較低的SQL伺服器授權成本）的比率為 1 : 32 vCPU-to-RAM。
- x2iezn – x2iezn 中的 "z" 屬性表示此執行個體類型著重於高處理器效能。Cascade Lake 處理器具高達 4.5 的全核心渦輪頻率。GHz建議您在想要保持低 vCPU 數量的情況下，使用此EC2執行個體，並搭配 1 : 32 vCPU-to-RAM 比例。這反過來可以降低SQL伺服器授權成本。

- r7iz – r7iz 中的 "z" 屬性表示此執行個體類型著重於高處理器效能。Sapphire rapids 處理器具有高達 3.9 的全核心渦輪頻率GHz。如同 x2iezn 執行個體，r7iz 會優先考慮高頻率處理器效能，但 vCPU-to-RAM比例為 1 : 8。

其他資源

- [一般用途 Amazon EC2執行個體](#) (AWS 文件)
- [比較工具](#) (Vantage)
- [授權 – SQL 伺服器](#) (AWS 文件)

合併執行個體

本節著重於將多個SQL伺服器執行個體合併到相同伺服器的成本最佳化技術，以將授權成本降至最低並最大化資源使用率。

概觀

建立執行個體是安裝SQL伺服器資料庫引擎程序的一部分。SQL 伺服器執行個體是完整的安裝，包含自己的伺服器檔案、安全登入和系統資料庫（主機、模型、msdb 和 tempdb）。由於執行個體擁有自己的所有檔案和服務，因此您可以在相同的作業系統上安裝多個SQL伺服器執行個體，而不需要執行個體互相干擾。但是，由於執行個體都安裝在相同的伺服器上，因此它們都共用相同的硬體資源，例如運算、記憶體和聯網。

通常在生產環境中，每個伺服器只能使用單一SQL伺服器執行個體，因此「忙碌」執行個體不會過度使用共用硬體資源。將自己的作業系統與自己的資源提供給每個SQL伺服器執行個體，是比依賴資源管理更好的界限。對於需要大量 RAM和 CPU 資源的高效能SQL伺服器工作負載來說尤其如此。

不過，並非所有SQL伺服器工作負載都會使用大量資源。例如，某些組織會為其每個客戶指派自己的專用SQL伺服器執行個體，用於合規或安全目的。對於通常不會處於作用中狀態的小型用戶端或用戶端，這表示使用最少的資源執行SQL伺服器執行個體。

如 [Microsoft SQL Server 2019：授權指南](#) 中所述，每個執行 SQL Server 的伺服器必須至少有四個 CPU授權。這表示即使您執行只有兩個的伺服器vCPUs，您仍然必須為四個 授權SQL伺服器vCPUs。如果您使用 Server SQL Standard 版，則根據 [Microsoft 的公有SQL伺服器定價](#)，差異為 \$3, 945。對於使用最少資源執行多個SQL伺服器執行個體的組織，必須授權未使用的資源的合併成本可能相當高。

成本最佳化案例

本節探討的範例案例，將執行四個 Windows Server 伺服器之間的差異與 SQL 同時執行多個 SQL 伺服器執行個體的單一大型 Windows Server 伺服器進行比較。

如果每個 SQL 伺服器執行個體只需要 2 GB vCPUs 和 8 GB RAM，則除了每小時運算成本 \$0.096 之外，SQL 伺服器授權的每部伺服器總成本為 \$7,890。

EC2 執行個體	vCPUs	RAM	價格	vCPUs 授權	SQL 伺服器授權總成本
m6i.large	2	8	0.096	4	7,890 美元

將此擴展到四個伺服器，SQL 伺服器授權的總成本為 31,560 美元，每小時運算成本為 0.384 美元。

EC2 執行個體	vCPUs	RAM	價格	vCPUs 授權	SQL 伺服器授權總成本
4 個 m6i.large	2	32	0.384	16	31,560 美元

如果您將所有四個 SQL 伺服器執行個體合併到單一 EC2 執行個體，運算資源和運算的總數會保持不變。不過，透過移除不必要的 SQL 伺服器授權成本，您可以將執行工作負載的總成本降低 15,780 美元。

EC2 執行個體	vCPUs	RAM	價格	vCPUs 授權	SQL 伺服器授權總成本
m6i.2xlarge	8	32	0.384	8	15,780 美元

Note

在上述表格中，運算成本會顯示 us-east-1 區域內執行 Windows Server 的 Amazon EC2 伺服器每小時隨需定價。SQL Server Standard Edition 授權成本是指 [Microsoft 的公有 SQL 伺服器定價](#)。

成本最佳化建議

如果您正在考慮合併SQL伺服器執行個體，最大的問題是您要合併的每個執行個體的資源消耗。請務必取得長期的效能指標，以便更了解每個伺服器上的工作負載模式。一些常見的資源消耗監控工具包括 [Amazon CloudWatch](#)、[Windows Performance Monitor](#) (perfmon) 和 SQL Server 的 [原生監控工具](#)。

我們建議您在分析您的SQL伺服器工作負載是否可以合併以使用相同的伺服器資源，而不會互相干擾時，考慮下列問題：

- 穩定狀態期間會耗用哪些資源 (、CPU記憶體和網路頻寬) ？
- 尖峰期間會耗用哪些資源 (、CPU記憶體和網路頻寬) ？
- 尖峰發生的頻率為何？尖峰是否一致？
- 一個伺服器的資源尖峰是否與另一個伺服器的資源尖峰同時發生？
- SQL 伺服器使用的儲存 [IOPS](#) 體和 [輸送量](#) 為何？

如果您想要繼續執行合併SQL伺服器執行個體的計劃，請參閱 Cloud Operations & Migrations 部落格上的 [在 Amazon 執行個體文章上執行多個SQL伺服器EC2執行個體](#)。AWS 此文章提供如何在SQL伺服器中進行組態變更以新增其他執行個體的指示。在開始之前，請考量在相同伺服器上安裝多個執行個體時的次要差異：

- 預設SQL的伺服器資料庫執行個體會命名MSSQLSERVER並使用連接埠 1433。
- 安裝在相同伺服器上的每個額外執行個體都是「已命名」的資料庫執行個體。
- 每個具名執行個體都有唯一的執行個體名稱和唯一的連接埠。
- [SQL Server Browser](#) 必須執行 以協調至具名執行個體的流量。
- 每個執行個體都可以針對資料庫資料檔案和個別登入使用不同的位置。
- SQL 伺服器 [最大伺服器記憶體設定](#) 必須根據每個執行個體的效能需求進行設定，其合併總數也為基礎作業系統留下足夠的記憶體。
- 您可以使用SQL伺服器 [原生備份和還原](#) 功能，或 [AWS DMS](#) 進行遷移或合併。

其他資源

- [SQL 伺服器授權資料表](#) (AWS 雲端操作與遷移部落格)
- [SQL 伺服器多執行個體設定部落格文章](#) (AWS Cloud Operations & Migrations 部落格)
- [SQL 伺服器最佳實務指南](#) (AWS 規範指南文件)

比較SQL伺服器版本

概觀

Microsoft SQL Server 授權是 Windows 工作負載環境的最大費用之一。SQL Server 的授權成本可以輕鬆延伸到運算成本之外，以執行工作負載。如果您選擇了錯誤的版本，您可以支付不使用或甚至不需要的功能。本節比較下列SQL伺服器版本，包括其功能和相對成本：

- Enterprise – SQL Server Enterprise Edition 提供具有高效能、無限制虛擬化和數個商業智慧 (BI) 工具的資料中心功能。
- 標準 – SQL Server Standard 版為小型組織和部門提供基本的資料管理和商業智慧。
- Web – SQL 伺服器 Web 版本適用於作為 Web 主機或 Web 增值提供者的公司 (VAPs)。此版本提供低總擁有成本，並為小型到大型 Web 屬性提供可擴展性和可管理性功能。

Important

您可以使用 SQL Server Web Edition 僅支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務。您無法使用 SQL Server Web Edition 支援 line-of-business 應用程式 (例如客戶關係管理或企業資源管理應用程式)。

- 開發人員 – SQL 伺服器開發人員版本包含 Enterprise 版本的所有功能，但僅供開發用途使用。
- Express – SQL Server Express 版本是免費的資料庫，可用於學習或建置桌面應用程式。您可以將 Express 版本更新為其他版本。

Note

SQL Server Evaluation 版本提供 180 天的試用期。

成本影響

您可以從 Microsoft 轉銷商購買SQL伺服器授權，並透過 AWS 軟體保證將授權帶到。或者，您可以將SQL伺服器授權與包含授權的 Amazon 的 pay-as-you-go EC2 模型搭配使用AMIs。

如果您從 Microsoft 經銷商購買SQL伺服器授權，核心授權會以兩個一包的形式銷售，且每個伺服器必須至少授權四個核心。下表顯示 Enterprise 和 Standard 版本之間的成本比較。

版本	SQL Server Enterprise 版 (2 個核心套件)	SQL Server Standard 版 (2 個核心套件)	節省
2022	15,123 美元	3,945 美元	74%
2019	13,748 美元	3,586 美元	74%

Note

上表中的定價是以 Microsoft Server [SQL2022](#) 和 [Server SQL2019](#) 的公開定價為基礎。

下列成本比較顯示使用包含授權的 Amazon EC2 託管不同版本的 SQL 伺服器 AMIs。在此比較中，SQL 伺服器託管在 us-east-1 區域中的 r6i.xlarge (4 v CPU) 上。

執行個體	運算成本	Windows 授權成本	SQL 伺服器授權成本	總計
R6i .xlarge (Linux)	183.96 美元	–	–	183.96 美元
R6i .xlarge + Windows	183.96 美元	134.32 美元	–	318.28 美元
R6i .xlarge + SQL Server Web 版本	183.96 美元	134.32 美元	49.35 美元	367.63 美元
R6i .xlarge + SQL Server Standard 版	183.96 美元	134.32 美元	350.4 美元	668.68 美元
R6i .xlarge + SQL Enterprise 版	183.96 美元	134.32 美元	1,095 美元	1,413.28 美元

您可以為您的工作負載選取正確的SQL伺服器版本，以節省高達 95% 的SQL伺服器授權成本。下表比較 r6i.xlarge 執行個體上的SQL伺服器授權成本。

版本	節省 %
與 Enterprise 相比的標準	68%
Web 與標準相比	86%
Web 與 Enterprise 相比	95%

在大多數情況下，組織會從 Enterprise 切換到 Standard Edition，但在某些情況下，可以從 Standard 或 Enterprise Edition 切換到 Web Edition。

成本最佳化建議

您可以根據擴展限制、高可用性、效能和安全性，為工作負載選擇最佳版本。下表顯示SQL伺服器版本支援的功能。這可協助您決定要使用的版本。此比較適用於 [SQL Server 2016 SP1及更新版本](#)。

擴展限制

下表比較不同SQL伺服器版本的擴展限制。

功能	企業版	標準版本	Web 版本	Express 版本
SQL Server Database Engine、SQL Server Analysis Services (SSAS) 或 SQL Server Reporting Services (SSRS) 的單一執行個體使用的最大運算容量	作業系統上限	限制為小於 4 個通訊端或 24 個核心	僅限 4 個通訊端或 16 個核心中較少的通訊端	僅限 4 個通訊端或 4 個核心中較少的通訊端

功能	企業版	標準版本	Web 版本	Express 版本
每個SQL伺服器 資料庫引擎執行 個體的緩衝集區 最大記憶體	作業系統上限	128 GB	64 GB	1410 MB
每個SQL伺服器 資料庫引擎執行 個體的緩衝集區 延伸容量上限	設定的記憶體數 量上限為 32 倍	設定的記憶體上 限為 4 倍	N/A	N/A
關聯式資料庫大 小上限	524 PB	524 PB	524 PB	10 GB
Columnstore 快 取或記憶體最佳 化資料的記憶體 上限	作業系統上限	32 GB	16 GB	352 MB

如果您的應用程式需要少於 16 個核心（32 個vCPUs）和 64 GB 的 RAM，則您可以從SQL伺服器 Web 版本開始評估。如果您的工作負載需要超過 64 GB 的記憶體或其他高可用性選項，則必須升級至 SQL Server Standard 版本。

您可以使用 SQL Server Web Edition 來支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務，但您無法使用 SQL Server Web Edition 來支援業務應用程式。如需SQL伺服器 Web 版本使用案例的詳細資訊，請聯絡 [Microsoft Licensing Support](#) 或您的 Microsoft 經銷商。

您可以針對最多 24 個核心（48 個vCPUs）和 128 GB 記憶體的工作負載使用 SQL Server Standard 版。不過，您可以使用[緩衝集區擴充](#)功能，讓 SQL Server Standard 版能夠使用[本機執行個體儲存體](#)，例如 r6id EC2執行個體中存在的執行個體。這可將記憶體擴展到最大記憶體組態的四倍大小。當記憶體需求開始增加時，這種功能組合可能會延遲伺服器必須升級至 Enterprise 版本。

您可以在緩衝區集區和[頁面預期壽命](#)計數器中找到資料庫頁面，以識別記憶體使用率。頁面預期壽命會告訴您頁面在排清回磁碟之前記憶體中的時間長度。此計數器預設值為 300。如果頁面存放在記憶體中數小時或數天，則有機會減少配置的記憶體。

高可用性

下表比較不同SQL伺服器版本的高可用性功能。

功能	企業版	標準版本	Web 版本	Express 版本
伺服器核心支援 1	是	是	是	是
日誌傳送	是	是	是	否
資料庫鏡像	是	FULL 安全模式	僅作為見證人	僅作為見證人
備份壓縮	是	是	否	否
Always On 容錯 移轉叢集執行個 體	16 個節點	2 個節點	否	否
Always On 可用 性群組	最多 8 個次要複 本，包括 2 個同 步次要複本	否	否	否
基本可用性群組	否	2 個節點	否	否
線上頁面和檔案 還原	是	否	否	否
線上索引	是	否	否	否
線上結構描述變 更	是	否	否	否
快速復原	是	否	否	否
鏡像備份	是	否	否	否
熱新增記憶體和 CPU	是	否	否	否
加密備份	是	是	否	否

功能	企業版	標準版本	Web 版本	Express 版本
混合備份至 Microsoft Azure (備份至 URL)	是	是	否	否
災難復原的容錯移轉伺服器	是	是	否	否
提供高可用性的容錯移轉伺服器	是	是	否	否

其他常見功能

下表比較不同SQL伺服器版本的最常見功能。如需功能的完整清單，請參閱 Microsoft 文件中的 [SQL Server 2019 版本和支援的功能](#)。

功能	企業版	標準版本	Web 版本	Express 版本
(效能) 資源總監	是	否	否	否
(安全) 透明資料庫加密 (TDE)	是	是	是	否
(安全) 可擴展金鑰管理 (EKM)	是	否	否	否
(複寫) Oracle 發佈	是	否	否	否
(複寫) 對等交易複寫	是	否	否	否
變更資料擷取	是	是	否	否

SQL 伺服器開發人員版本

所有非生產工作負載，例如開發、QA、測試、暫存和UAT環境，都可以使用 SQL Server Developer Edition 節省 100% 的SQL伺服器授權成本。[下載SQL伺服器](#) 之後，您可以使用共用租用在EC2執行個體上安裝SQL伺服器開發人員版本。SQL Server Developer Edition 不需要專用基礎設施。如需詳細資訊，請參閱本指南的 [SQL Server Developer Edition](#) 建議。

切換版本

對於現有工作負載，從一個版本切換到另一個版本需要大量測試。檢查在 Enterprise 或 Standard 版本上執行的工作負載是否使用版本特定的功能，以及這些功能是否有任何替代解決方案是最佳實務。例如，如果您想要查看資料庫是否使用任何企業級功能，您可以執行所有資料庫上的[動態管理檢視 \(DMV\)](#)，如下列範例命令所示。

```
SELECT feature_name FROM sys.dm_db_persisted_sku_features; GO
```

有些 Enterprise Edition 功能無法在 T- 中擷取SQL，例如作為SQL維護任務一部分的線上重新索引。這些必須手動驗證。

遷移考量事項

您授權SQL伺服器的方式將決定切換版本的選項。AMIs，包括SQL伺服器 AMIs在內，具有包含在EC2執行個體價格中的授權成本，授權成本受限於AMI。您可以使用[AWS 帳單代碼](#)來驗證中包含的SQL伺服器版本AMI。對於包含 AWS 授權的執行個體，在作業系統中變更 SQL Server 版本不會變更與相關聯的帳單AMI。您必須將資料庫遷移至AMI執行新版SQL伺服器的新EC2執行個體。

如果您攜帶自己的授權，則有更多彈性。通常仍建議遷移至執行新版本的另一個EC2執行個體。如果某件事未按計劃進行，這可讓您輕鬆容錯回復。不過，如果您必須使用現有的伺服器，您仍然 side-by-side可以安裝SQL伺服器，並在執行個體之間遷移資料庫。如需版本降級的詳細資訊 side-by-side，請參閱 MSSQLTips 網站上的[SQL伺服器中的版本升級和降級](#)。

其他資源

- [SQL Server 2022 \(Microsoft Learn \) 的版本和支援的功能](#)
- [sys.dm_db_persisted_sku_features \(Transact-SQL \) \(Microsoft Learn \)](#)
- [您應該使用哪個版本的SQL伺服器？ \(Brent Ozar Unlimited \)](#)
- [AWS Pricing Calculator \(AWS\)](#)

評估SQL伺服器開發人員版本

概觀

[SQL Server Developer Edition](#) 是免費版的 SQL Server，其中包含 Enterprise Edition 的所有功能，可用於任何非生產環境。在無法使用 Microsoft Developer Network (MSDN) 授權的雲端中，SQL 伺服器開發人員版本是節省成本的好方法，而不必為開發和測試工作負載提供授權。對於執行大型開發和測試環境，並尋求降低不必要的成本的團隊來說，這一點尤其如此。

生產環境定義為應用程式（例如網際網路網站）最終使用者存取的環境，其用途不僅在於收集應用程式的意見回饋或接受測試。構成生產環境的其他案例包括：

- 連線至生產資料庫的環境
- 支援生產環境災難復原或備份的環境
- 至少部分時間用於生產的環境，例如在活動尖峰時段輪換至生產的伺服器

如需授權的詳細資訊，請參閱文件中的 AWS [Amazon Web Services](#) 和 Microsoft：[常見問題](#)。

成本影響

如果您將 SQL Server Developer Edition 用於非生產工作負載，則可以為開發和測試環境節省 100% 的目前 SQL 伺服器授權成本。

SQL 伺服器版本	SQL Server Enterprise 版 (2 個核心套件)	SQL Server Standard 版 (2 個核心套件)	SQL 伺服器開發人員版本
2022	15,123 美元	3,945 美元	免費
2019	13,748 美元	3,586 美元	免費

Note

上表中的定價是以 Microsoft Server [SQL2022](#) 和 Server [SQL2019](#) 的公開定價為基礎。

下表比較在 us-east-2 區域中 vCPUs 使用隨需定價與 4 執行的不同 SQL 伺服器版本的成本。這適用於依賴 中包含授權之執行個體的案例 AWS。

EC2 執行個體	AMI	計算價格	Windows 授權價格	SQL 伺服器授權價格	總價
r5.xlarge	Linux (運算定價)	183.96 美元	–	–	183.96 美元
r5.xlarge	Linux + SQL Server Developer 版本	183.96 美元	0 USD	0 USD	183.96 美元
r5.xlarge	Windows Server (LI)	183.96 美元	134.32 美元	–	318.28 美元
r5.xlarge	Windows + SQL Server 開發人員版本	183.96 美元	134.32 美元	0 USD	318.28 美元
r5.xlarge	Windows + SQL Server Web Edition (LI)	183.96 美元	134.32 美元	49.64 美元	367.92 美元
r5.xlarge	Windows + SQL Server Standard 版 (LI)	183.96 美元	134.32 美元	350.4 美元	\$668.68
r5.xlarge	Windows + SQL Server Enterprise Edition (LI)	183.96 美元	134.32 美元	1095 美元	1413.28 美元

成本最佳化案例

資料完整性公司進行新的擷取後，想要將新取得的工作負載從受管託管提供者的目前位置遷移，以與中的其他工作負載合併 AWS 雲端。初始定價顯示，與目前的受管服務供應商 AWS 相比，在上執行的公司的 SQL 伺服器工作負載成本會高出 60%。MACO SME 評估了估算，發現客戶實際上在受管託管提

供者為開發和測試環境支付SQL了伺服器授權。透過在遷移期間將非生產工作負載切換到 SQL Server Developer 版本，公司減少了 40% 的SQL伺服器授權。

SQL Amazon 上包含的伺服器授權 EC2

如果您在使用[包含 授權的AMIs](#)EC2執行個體上有 SQL 伺服器，則無法從 Enterprise Edition 直接轉換為 Developer Edition。包含授權的執行個體的授權成本會綁定至 AMI。即使從作業系統內解除安裝 SQL 伺服器，EC2執行個體仍需支付授權成本。

若要轉換為 Developer Edition，您必須[下載 SQL Server Developer Edition](#)，將其安裝在新的EC2執行個體上，然後遷移資料庫。您可以使用各種方法在EC2執行個體之間遷移SQL伺服器資料庫。如需詳細資訊，請參閱指南中的遷移 Microsoft Server 資料庫中的SQL伺服器資料庫 AWS 雲端[SQL遷移方法](#)。您也可以使用 [Automated SQL Server Developer 解決方案](#)來準備您計劃遷移的新執行個體。

SQL Amazon BYOL上的伺服器 EC2

如果您有使用的SQL伺服器執行個體BYOL，您可以選擇下列就地轉換或 side-by-side降級選項：

- 從 Microsoft 網站下載[SQL伺服器開發人員版本](#)。如需手動或自動安裝指示，請參閱 AWS 部落格上的[自動化SQL伺服器開發人員部署](#)文章。
- 使用[SQL伺服器原生備份和還原](#)來遷移資料庫，或將資料庫從一個SQL執行個體分離/連接到另一個執行個體。
- 使用 [自動化工具](#)進行大量部署。

Note

SQL 伺服器開發人員版本僅適用於非生產環境。

其他資源

- [自動化 SQL Server Developer 部署，以在 \(部落格 \) 上部署 SQL Server Developer Edition EC2AWS](#)
- [SQL 2022 年定價](#) (Microsoft)
- [SQL 2019 年定價](#) (Microsoft)
- [授權選項](#) (SQLAmazon 上的伺服器EC2)
- [AWS Pricing Calculator](#) (SQLAmazon EC2 文件上的伺服器)

- [Microsoft SQL Server 2019 授權指南](#) (從 Microsoft 下載)
- [SQL Server 2022 開發人員版本](#) (從 Microsoft 下載)

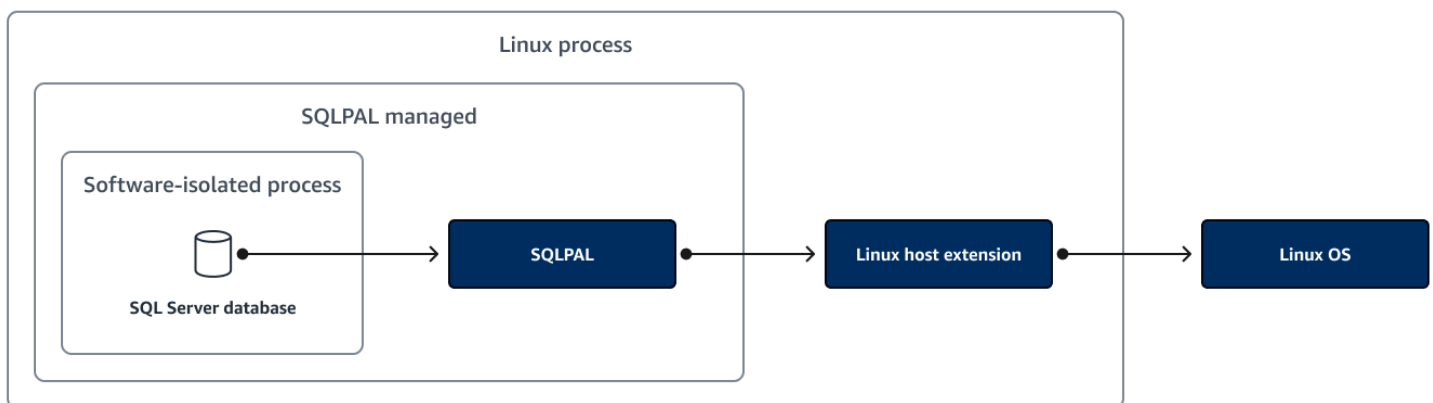
在 Linux 上評估SQL伺服器

概觀

自 SQL Server 2017 以來，可以在 Linux 作業系統上安裝SQL伺服器。SQL Linux 上的伺服器已就緒，可提供彈性、高效能、安全功能、減少 TCO、HA/DR 功能，以及絕佳的使用者體驗。您可以將 Windows Server 上的SQL伺服器切換至 Linux 上的SQL伺服器，以節省 Windows Server 授權成本。

對於 Linux，SQL伺服器可在 Red Hat Enterprise Linux (RHEL)、SUSELinux Enterprise Server (SLES)、Ubuntu 和 Amazon Linux 2 上部署。SQL 伺服器資料庫引擎在 Windows Server 和 Linux 上執行的方式相同，但在使用 Linux 時，某些任務有一些基本變更。在 Linux 和 Windows 上執行 SQL Server Always On 應用程式與容錯移轉叢集之間的一個關鍵區別。如果您在 Windows Server 主機上部署 Always On 可用性群組，則可以利用 [Windows Server 容錯移轉叢集 \(WSFC \)](#) 和 Active Directory 作為支援容錯移轉叢集的內建功能。不過，Active Directory WSFC和 都無法支援 Linux 上的容錯移轉叢集。如果您想要在 Linux 上啟動SQL伺服器的容錯移轉叢集，您可以使用 [ClusterLabs Pacemaker AWS Launch Wizard](#)簡化 Linux 執行個體上的叢集設定和SQL安裝。

SQL Windows 和 Linux 上的伺服器共用共同的程式碼基礎。也就是說，SQL伺服器核心引擎完全沒有變更為在 Linux 上執行。SQL 伺服器引入了 Platform Abstraction Layer (SQLPAL)，如下圖所示。



SQLPAL 負責抽象化SQL伺服器與基礎作業系統之間的呼叫和通訊。主機延伸模組只是原生 Linux 應用程式。低階作業系統函數是原生呼叫，用於最佳化 I/O、記憶體和CPU用量。當主機擴充功能啟動時，它會載入並初始化 SQLPAL，然後調用SQL伺服器。SQLPAL 啟動隔離的軟體程序，為其餘程式碼提供所需的翻譯。將此新層新增至SQL伺服器架構意味著無論作業系統為何，都可以使用在 Windows 上讓SQL伺服器如此強大的相同企業級核心功能和優點。

成本影響

對於 r5.2xlarge 執行個體，在每個案例中，Windows Server 授權成本降低約為 268 美元。與使用價格較低的伺服器版本相比，減少是 SQL 伺服器總成本的較高百分比。下表顯示節省的成本。

執行個體	版本	Windows 上的 SQL 伺服器每月成本	Linux 上的 SQL 伺服器每月成本	節省
r5.2xlarge	Web	735 美元	466 美元	37%
r5.2xlarge	標準	1,337 美元	1,068 美元	20%
r5.2xlarge	Enterprise	2,826 美元	2,558 美元	10%

Note

上表中的價格估算是根據 us-east-1 區域中的隨需定價，並且可以直接在 [中檢視 AWS Pricing Calculator](#)。

考慮一個範例案例，其中客 SMB 群中的 ISV 客戶希望為其開發環境節省成本。他們已在一組 Windows 伺服器上使用 SQL Server Developer Edition。透過從 Windows with SQL Server Developer Edition 切換至 Linux with SQL Server Developer Edition，ISV 客戶可以在開發工作負載上節省 33%。下表顯示此案例的下列預估成本。

估算	每月成本
Windows + SQL 伺服器	9,307.72 美元
Linux + SQL 伺服器	6,218.36 美元
預估節省成本	\$3,089.36 (33%)

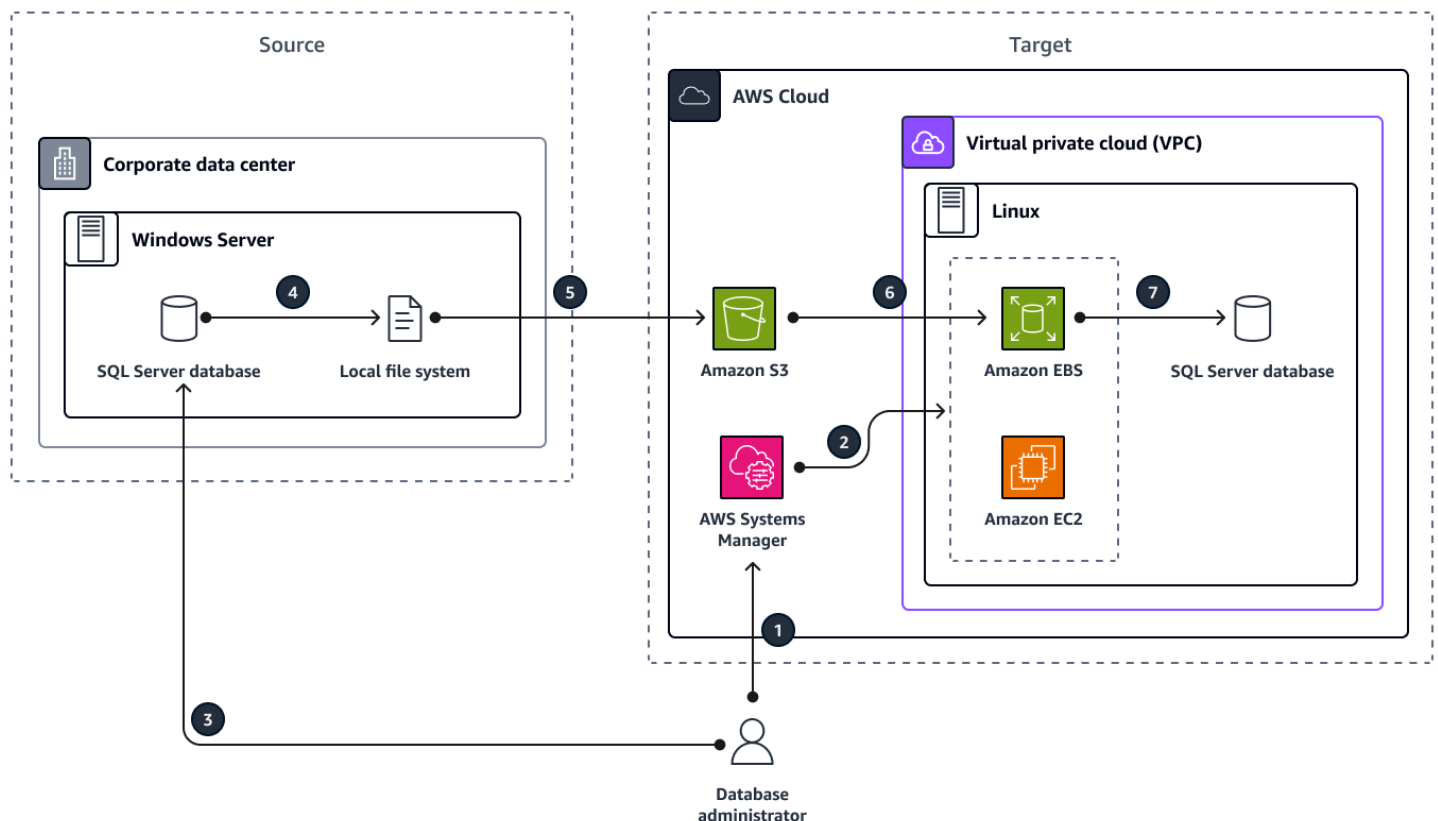
在另一個範例案例中，公司會將授權包含的 SQL 伺服器 EC2 執行個體從 Windows 遷移至 Linux。該公司每年總共節省 300,000 美元的 Windows Server 授權成本，大約是帳單總額 AWS 的 20%。

成本最佳化建議

我們建議您考慮下列事項：

- SQL 從 Server 2017 開始支援 Linux 上的 SQL 伺服器。
- 若要協助進行切換，您可以使用 [Windows 至 Linux 的 Microsoft SQL Server Databases 的 轉譯助理](#)。轉換助理是一種指令碼工具，可以透過檢查是否有常見的不相容、從 Windows 主機匯出資料庫，然後將資料庫匯入在 Ubuntu 16.04 上執行 Microsoft SQL Server 2017 的 EC2 執行個體，協助您將現有 SQL 伺服器工作負載從 Windows 移至 Linux 作業系統。
- 您也可以使用 SQL 伺服器中的 [備份和還原](#) 功能，從 Windows 上的 SQL 伺服器切換到 Linux。
- 您可以使用 輕鬆快速地部署到 Linux 或 Ubuntu 上的 SQL 伺服器 [AWS Launch Wizard](#)。啟動精靈可以根據您的應用程式需求，在獨立和高可用性案例中部署 Linux 或 Ubuntu 上的 SQL 伺服器。如需詳細資訊，請參閱 AWS 部落格 Microsoft 工作負載中的 [部署至 Linux 上的 SQL 伺服器永遠 AWS Launch Wizard](#)。

下圖顯示 解決方案的架構，該解決方案使用 Windows 至 Linux 的 Microsoft SQL Server 資料庫轉換助理。



其他資源

- [Linux SQL 伺服器概觀](#) (Microsoft Learn)
- [Linux 上的SQL伺服器安裝指南](#) (Microsoft Learn)
- [使用 部署至 Linux 上的 SQL Server Always AWS Launch Wizard](#) (AWS 部落格上的 Microsoft 工作負載)
- [Linux 上的高可用性SQL伺服器](#) (AWS 開放原始碼部落格)

最佳化SQL伺服器備份策略

概觀

大多數組織都在尋找正確的解決方案來保護其 [Amazon EC2](#) 上的 SQL Server 上的資料，以滿足其目前對復原點目標 (RPO)、自上次備份以來可接受的時間上限，以及復原時間目標 (RTO)，以及服務中斷和還原服務之間的可接受延遲上限的要求。如果您在EC2執行個體上執行SQL伺服器，您有多個選項可建立資料備份並還原資料。保護 Amazon 上 SQL Server 資料的備份策略EC2包括：

- 使用支援 Windows [Volume Shadow Copy Service](#) (VSS) 的 Amazon Elastic Block Store (AmazonEBS) 快照或的伺服器層級備份 [AWS Backup](#)
- 在SQL伺服器中使用[原生備份和還原的資料庫層級備份](#)

您具有下列[資料庫層級原生備份](#)的儲存選項：

- 具有 [Amazon EBS磁碟區](#)的本機備份
- 使用 [Amazon FSx for Windows File Server](#) 或 Amazon FSx for 的網路檔案系統備份 NetApp ONTAP
- 使用 將網路備份至 Amazon Simple Storage Service (Amazon S3) [AWS Storage Gateway](#)
- 直接備份至 Amazon S3 for SQL Server 2022

本節執行下列動作：

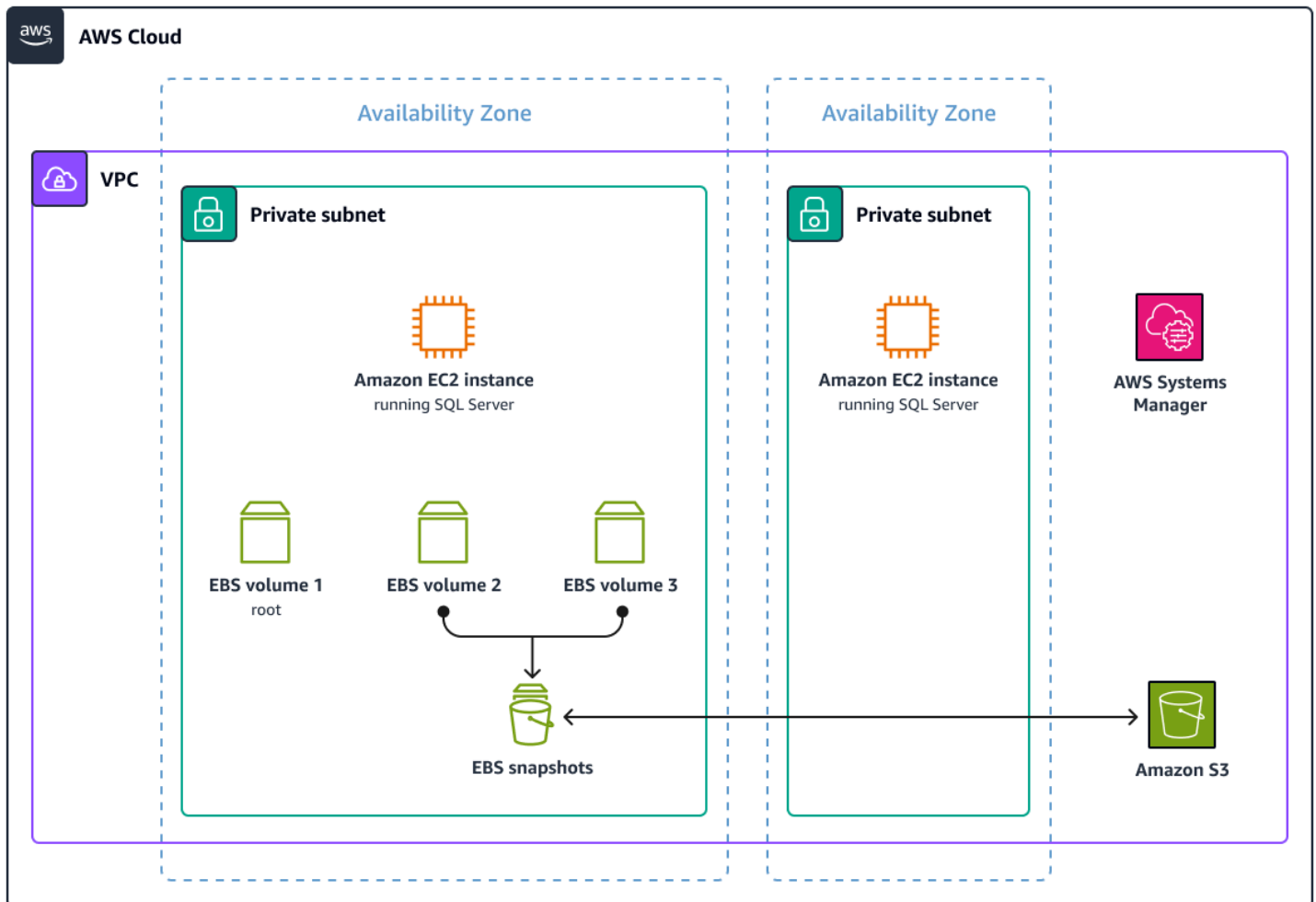
- 反白顯示可協助您節省儲存空間的功能
- 比較不同後端儲存選項之間的成本
- 提供深入文件的連結，以協助實作這些建議

使用VSS啟用的快照進行伺服器層級備份

VSS啟用的快照架構會使用 AWS Systems Manager [執行命令](#) 在您的SQL伺服器執行個體上安裝VSS代理程式。您也可以使用 Run Command 叫用整個工作流程，將作業系統和應用程式緩衝區排清到磁碟、暫停 I/O 操作、拍攝 point-in-time EBS 磁碟區快照，然後繼續 I/O。

此執行命令會建立連接至目標執行個體之所有EBS磁碟區的自動快照。也可以選擇排除根磁碟區，因為使用者資料庫檔案通常儲存在其他磁碟區上。如果您為SQL伺服器檔案分割多個EBS磁碟區以建立單一檔案系統，Amazon EBS也支援使用單一API命令的損毀一致性多磁碟區快照。如需啟用應用程式一致性 [VSSEBS快照](#) 的詳細資訊，請參閱 Amazon EC2 文件中的 [建立VSS應用程式一致性快照](#)。

下圖顯示使用VSS已啟用的快照進行伺服器層級備份的架構。



考慮使用VSS啟用的快照的下列優點：

- 資料庫執行個體的第一個快照會包含完整的資料庫執行個體資料。相同資料庫執行個體的后續快照均採用[遞增](#)處理，這表示系統僅會儲存上次執行快照後有所變更的資料。
- EBS 快照提供 point-in-time 復原。
- 您可以從[快照 還原至新的SQL伺服器EC2執行個體](#)。
- 如果執行個體使用 Amazon 加密，EBS 或者如果執行個體中的資料庫使用 加密TDE，則該執行個體或資料庫會自動以相同的加密還原。
- 您可以複製[跨區域的自動備份](#)。
- 當您從快照還原磁碟區時，應用程式會立即存取該EBS磁碟區。這表示從快照還原一個或多個基礎EBS磁碟區後，您可以立即讓SQL伺服器上線。
- 依預設，還原的磁碟區會在應用程式第一次嘗試讀取時，從 Amazon S3 中擷取基礎區塊。這表示從快照還原EBS磁碟區後，效能可能會延遲。音量最終達到標稱性能。不過，您可以使用[快速快照還原 \(FSR\)](#) 快照來避免延遲。
- 您可以使用[EBS快照 的生命週期管理](#)。

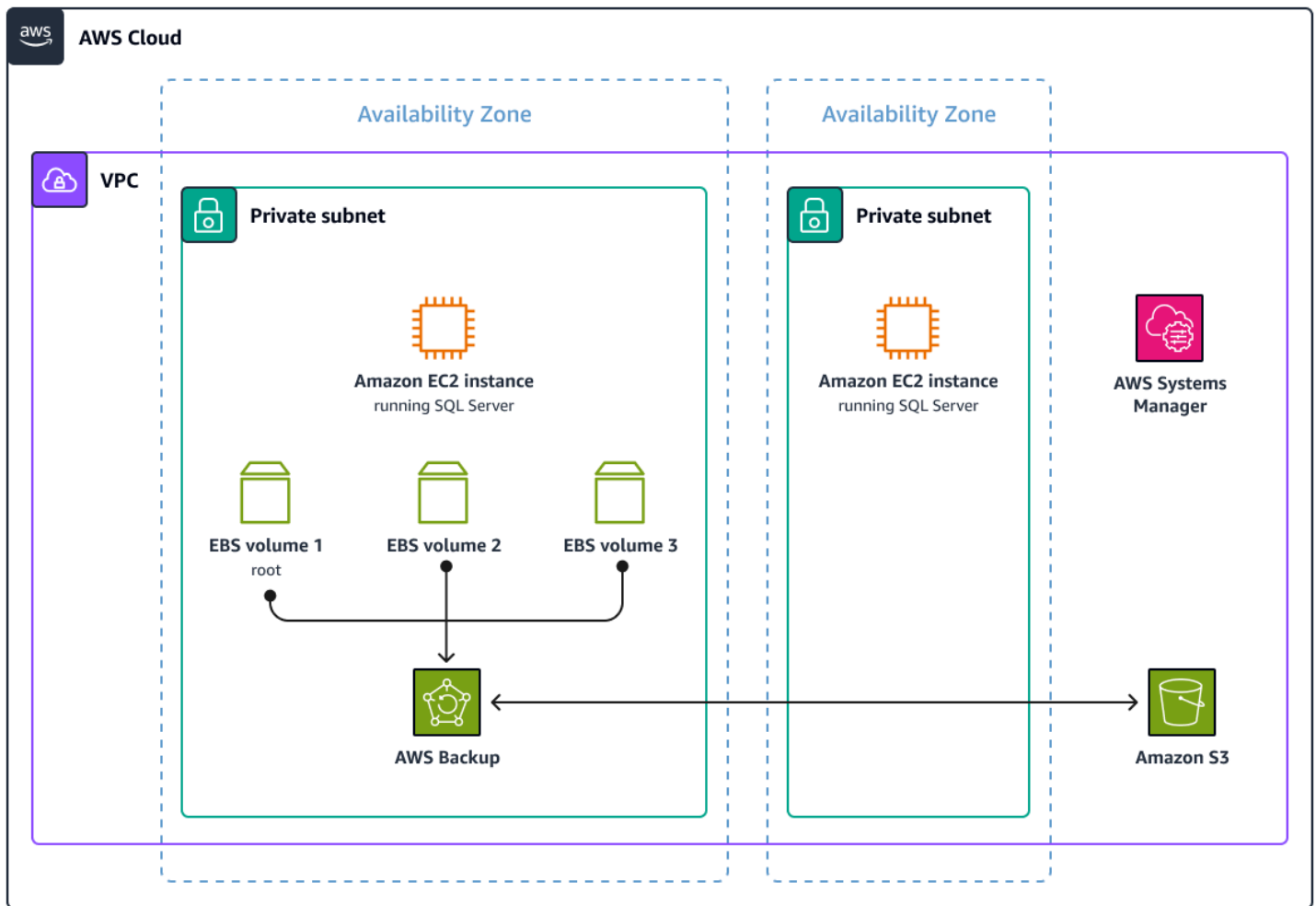
考慮使用VSS啟用的快照的下列限制：

- 您無法使用SQL伺服器執行個體的加密快照執行跨區域 point-in-time 復原。
- 您無法建立未加密執行個體的加密快照。
- 您無法還原個別資料庫，因為快照是在EBS磁碟區層級進行拍攝。
- 您無法將執行個體還原至其本身。
- 資料庫執行個體的快照必須使用與資料庫執行個體相同的 AWS Key Management Service (AWS KMS) 金鑰加密。
- 在快照備份過程中，儲存 I/O 會暫停幾秒鐘 (約 10 毫秒)。

SQL 使用的伺服器備份 AWS Backup

您可以使用 [AWS Backup](#) 來集中和自動化跨的資料保護 AWS 服務。AWS Backup 提供具成本效益、完全受管、以政策為基礎的解決方案，可簡化大規模資料保護。AWS Backup 也可協助您支援法規遵循義務，並實現業務連續性目標。搭配 AWS Organizations，AWS Backup 可讓您集中部署資料保護 (備份) 政策，以設定、管理和管理整個組織 AWS 帳戶 和資源的備份活動。

下圖顯示EC2使用 為 上的SQL伺服器備份和還原解決方案的架構 AWS Backup。



請考慮使用 備份SQL伺服器時具有下列優點 AWS Backup：

- 可自動化備份排程、保留管理和生命週期管理。
- 您可以集中整個組織的備份策略，跨越多個帳戶和 AWS 區域。
- 您可以透過 集中監控備份活動和警示 AWS 服務。
- 可以為災難復原規劃實作跨區域備份。
- 該解決方案可支援跨帳戶備份。
- 可以使用次要備份加密執行安全備份。
- 所有備份都支援使用加密金鑰進行 AWS KMS 加密。
- 解決方案適用於 TDE。
- 可以從 AWS Backup 主控台還原至特定復原點。
- 您可以備份整個SQL伺服器執行個體，其中包含所有SQL伺服器資料庫。

資料庫層級備份

這些方法使用原生 Microsoft SQL Server 備份功能。您可以在 SQL 伺服器執行個體上備份個別資料庫，並還原個別資料庫。

原生 SQL 伺服器備份和還原的每個選項也支援下列各項：

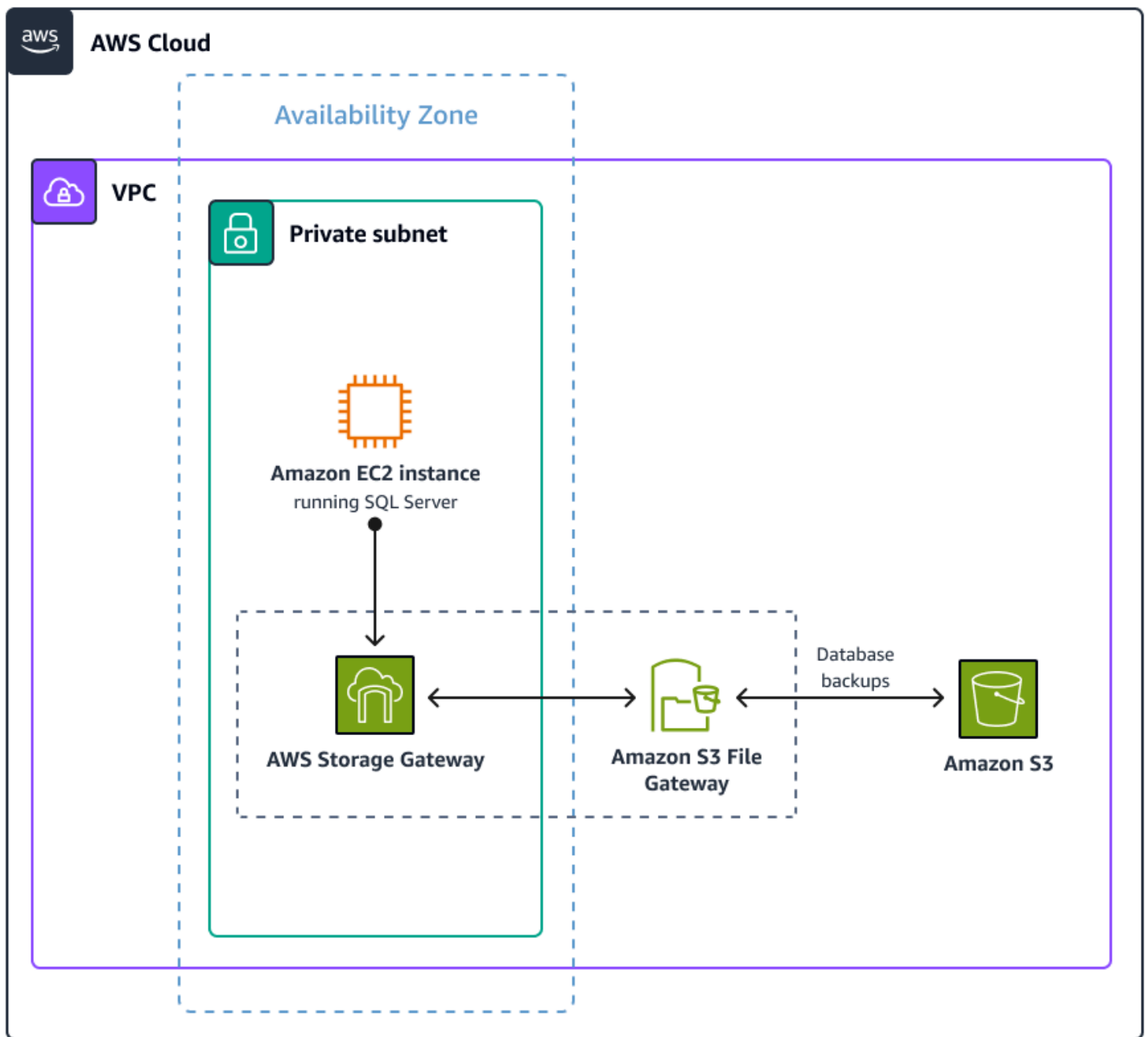
- 壓縮和多檔案備份
- 完整、差異和 T 日誌備份
- TDE-加密的資料庫

SQL 伺服器原生備份和還原至 Amazon S3

SQL Amazon 上的伺服器 EC2 支援 SQL 伺服器資料庫的原生備份和還原。您可以備份您的 SQL 伺服器資料庫，然後將備份檔案還原至現有資料庫或新的 SQL 伺服器 EC2 執行個體、Amazon RDS for SQL Server 或內部部署伺服器。

Storage Gateway 是一種混合雲端儲存服務，可讓內部部署應用程式存取幾乎無限制的雲端儲存。您可以使用 Storage Gateway 將 Microsoft SQL Server 資料庫直接備份到 Amazon S3，減少內部部署儲存體佔用空間，並使用 Amazon S3 進行持久、可擴展且符合成本效益的儲存。

下圖顯示使用 Storage Gateway 和 Amazon S3 的原生備份和還原解決方案的架構。



考慮將原生SQL伺服器備份與 Storage Gateway 搭配使用的下列優點：

- 您可以在EC2執行個體上將儲存閘道對應為伺服器訊息區塊（SMB）檔案共用，並將備份傳送至 Amazon S3。
- 備份會直接傳送至 S3 儲存貯體或透過 Storage Gateway 檔案快取。
- 支援多檔案備份。

考慮使用 Storage Gateway 進行原生備份的下列限制：

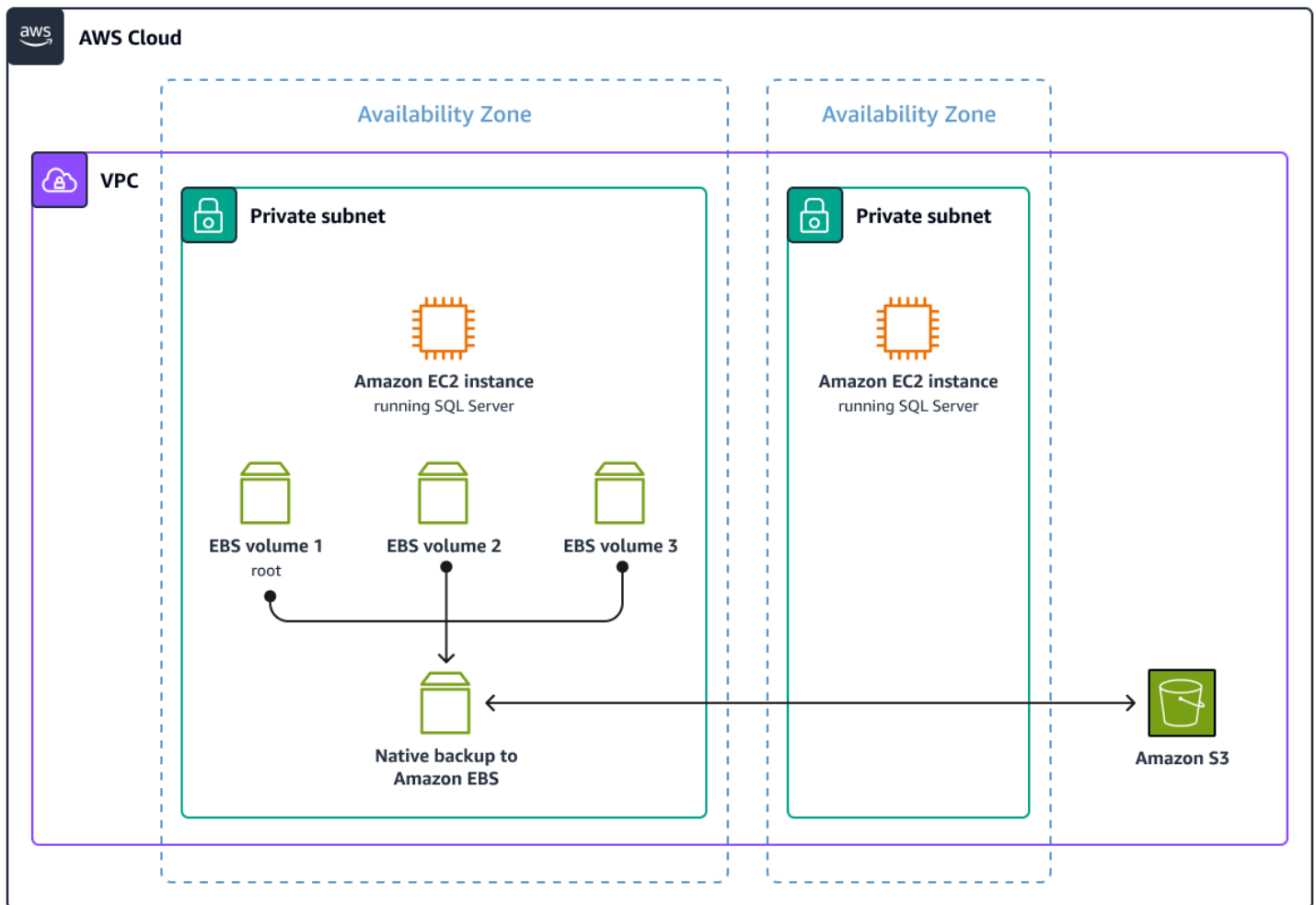
- 必須為每個單獨的資料庫設定備份和還原。
- 必須管理備份檔案的 [Amazon S3 生命週期政策](#)。

如需如何設定 Storage Gateway 的詳細資訊，請參閱 AWS 使用部落格上的文章在 [Amazon S3 中儲存 SQL 伺服器備份 AWS Storage Gateway](#)。

SQL 伺服器原生備份至EBS磁碟區

您可以對SQL伺服器資料庫進行原生備份，並將檔案存放在 Amazon EBS磁碟區中。Amazon EBS 是一種高效能區塊儲存服務。EBS 磁碟區是彈性的，支援加密。它們可以分離並連接到EC2執行個體。您可以在相同EBS磁碟區類型或不同EBS磁碟區類型的EC2執行個體上備份SQL伺服器。備份到不同EBS磁碟區的一個優點是節省成本。

下圖顯示 磁碟EBS區的原生備份架構。



考慮使用SQL伺服器原生備份至EBS磁碟區的下列優點：

- 您可以備份SQL伺服器EC2執行個體上的個別資料庫，並還原個別資料庫，而不必還原完整的執行個體。
- 支援多檔案備份。
- 您可以使用 SQL Server Agent 和 SQL Server 任務引擎來排程備份任務。
- 可以透過硬體選擇獲得效能優勢。例如，可以使用 st1 儲存磁碟區來達到更高的輸送量。

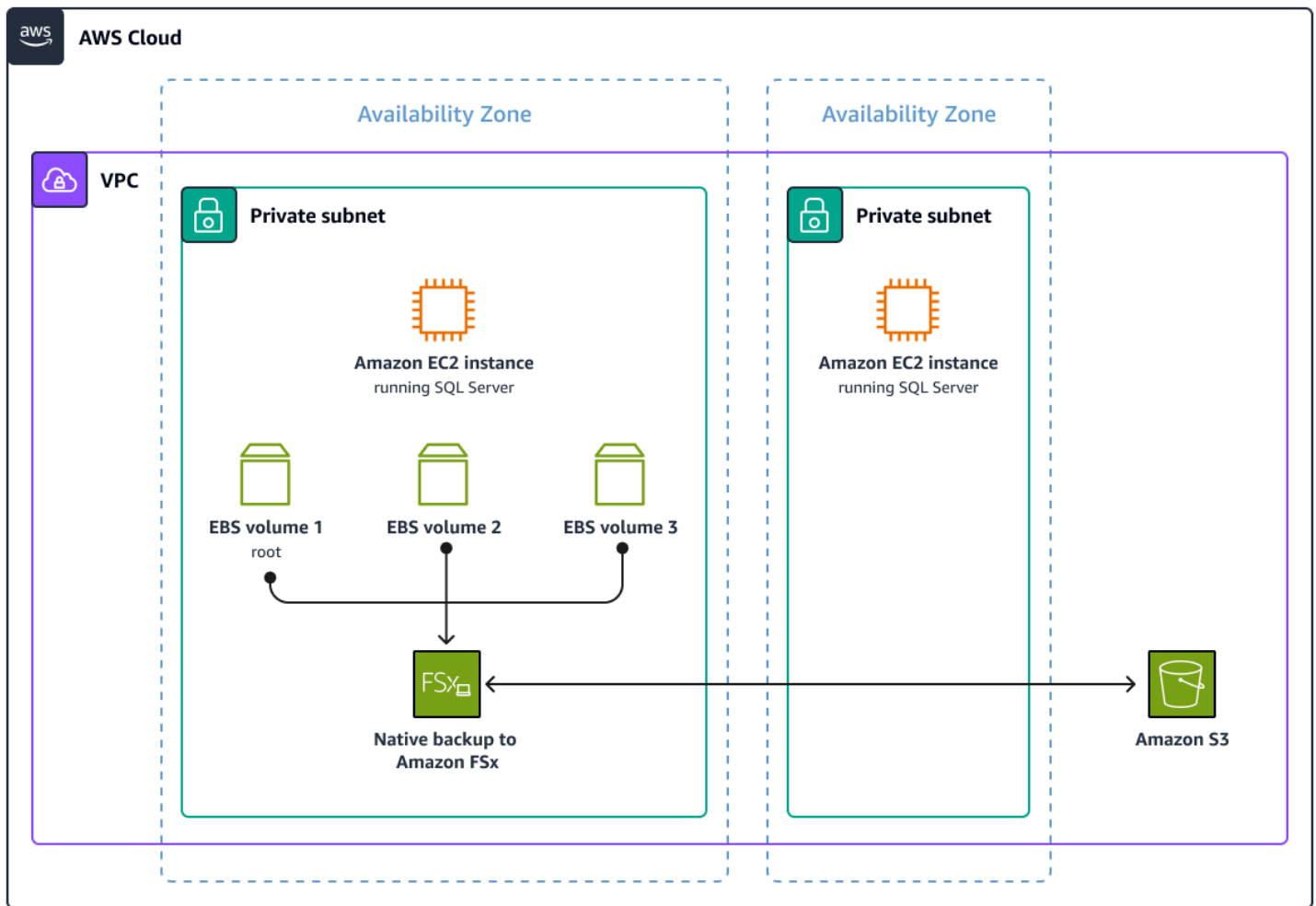
考慮使用原生備份至EBS磁碟區的下列限制：

- 您必須手動將備份從EBS磁碟區移至 Amazon S3。
- 對於大型備份，您必須在 Amazon 上管理磁碟空間EC2。
- 在EC2執行個體上，Amazon EBS輸送量可能會成為瓶頸。
- 在 Amazon 上儲存備份需要額外的儲存體EBS。

SQL Amazon FSx for Windows File Server 的伺服器原生備份

[Amazon FSx for Windows File Server](#) 是完全受管的原生 Windows 檔案系統，可提供高達 64 TB 的儲存空間，旨在提供快速、可預測且一致的效能。針對 FSx Windows File Server 的 上的多可用區域檔案系統部署 AWS，引進原生支援。<https://aws.amazon.com/blogs/aws/amazon-fsx-for-windows-file-server-update-new-enterprise-ready-features/>原生支援可讓您更輕鬆地在 上部署 Windows 檔案儲存，並在多個可用區域之間 AWS 具有高可用性和備援。AWS 也推出[SMB對持續可用（CA）檔案共用的支援](#)。您可以使用 FSx for Windows File Server 作為SQL伺服器資料庫的備份儲存體。

下圖顯示 Windows File Server FSx的 原生SQL伺服器備份架構。



考慮使用 Windows File Server FSx 的 原生SQL伺服器備份的下列優點：

- 您可以將SQL伺服器資料庫備份至 Amazon FSx 檔案共用。
- 您可以備份SQL伺服器執行個體上的個別資料庫，並還原個別資料庫，而不必還原完整的執行個體。
- 支援分段備份。
- 您可以使用 SQL Server Agent 和任務引擎來排程備份任務。
- 相較於 Amazon EBS，執行個體的網路頻寬更高。

考慮使用 Windows File Server FSx 的 原生SQL伺服器備份的下列限制：

- 您必須FSx使用 AWS Backup 或 手動將備份從 Amazon S3 移至 Amazon S3 AWS DataSync。
- 大型備份可能需要額外的額外額外負荷才能在 Amazon 上管理磁碟空間FSx。
- EC2 執行個體網路輸送量可能是瓶頸。
- 在 FSx Windows File Server 的 上存放備份需要額外的儲存體。

SQL FSx的伺服器備份至 Amazon NetApp ONTAP

使用 FSx的快照ONTAP一律會一致當機，但需要您查詢（或暫停資料庫的 I/O），才能建立應用程式一致性快照。您可以使用 NetApp SnapCenter（具有特定應用程式的外掛程式的協調工具，包括SQL伺服器）與 FSxONTAP建立應用程式一致性快照，並保護、複寫和複製資料庫，無需額外費用。

NetApp SnapCenter

NetApp SnapCenter 是應用程式一致性資料保護的統一平台。SnapCenter 稱為快照作為備份。本指南採用相同的命名慣例。SnapCenter 提供單一面板，用於管理應用程式一致性備份、還原和複製。您可以為特定資料庫應用程式新增 SnapCenter 外掛程式，以建立應用程式一致性備份。SnapCenter 適用於 SQL Server 的外掛程式提供下列功能，可簡化資料保護工作流程。

- 完整備份和日誌備份的精細備份和還原選項
- 就地還原和還原至替代位置

如需的詳細資訊 SnapCenter，請參閱使用 [NetApp SnapCenter 搭配 Amazon 保護SQL伺服器工作負載FSx，以在 NetApp ONTAP AWS Storage 部落格上發佈](#)。

備份的成本最佳化

下列選項可協助您降低在上儲存SQL伺服器備份的成本 AWS。

- 在建立備份檔案期間啟用[SQL伺服器壓縮](#)，並將最小的檔案傳送至儲存體。例如，3 : 1 壓縮率表示您在磁碟空間上儲存了約 66%。若要在這些資料欄上查詢，您可以使用下列 Transact-SQL 陳述式：`SELECT backup_size/compressed_backup_size FROM msdb..backupset;`
- 對於前往 S3 儲存貯體的備份，請啟用 [Amazon S3 Intelligent-Tiering](#) 儲存類別，將儲存成本降低 30%。
- 對於前往FSx適用於 Windows File Server 或FSx適用於的備份ONTAP，請使用單一可用區域節省 50% 的成本（與使用多個可用區域相比）。如需定價資訊，請參閱 [Amazon FSx for Windows File Server Pricing](#) 和 [Amazon FSx for NetApp ONTAP Pricing](#)。
- SQL Server 2022 最有效率的選項是直接備份至 Amazon S3。您可以避免 Storage Gateway，以節省額外的成本。

備份基準測試結果

本節根據本指南中涵蓋的備份解決方案的效能基準測試結果，比較範例 1 TB 資料庫的成本和效能觀點的下列選項。

- EC2 執行個體規格 – r5d.8xlarge 搭配 Windows Server 2019 和 SQL Server 2019 開發人員版本
- 資料庫規格 – 1 TB 大小且TDE已停用

測試是以 r5d.8xlarge 執行個體和 1 TB SQL Server 資料庫作為來源來執行。根據最佳實務設定來源系統，來源資料庫包含四個資料檔案（每個 250 GB）和一個日誌檔案（50 GB），分散到不同的 gp3 磁碟區。SQL 伺服器原生BACKUP命令包括寫入 10 個備份檔案，使用壓縮來最佳化備份效能，並減少透過網路傳送並寫入目標的資料量。在所有測試案例中，儲存效能是瓶頸。

對於這些類型的測試，有各種各樣的可能組態。此測試著重於最佳化效能、成本、可擴展性和實際使用案例。下表顯示為備份目標選項擷取的效能指標。

備份選項	Level	執行持續時間 (Appx)	備份速率	USD 每月成本*
本機 EBS st1 的 原生備份HDD， 2 TB	資料庫	00 : 30 : 46 分鐘	554.7 Mbps	92.16 美元
本機 EBS SSD gp3 的原生備 份，2 TB	資料庫	00 : 22 : 00 分鐘	512 Mbps	193.84 美元
原生備份至 FSx for Windows File Server HDD，2 TB @512 Mbps 輸送量	資料庫	00 : 20 : 58 分鐘	814.0 Mbps	1,146 美元
原生備份至 FSx for Windows File Server SSD，2 TB @512 Mbps 輸送量	資料庫	00 : 20 : 00 分鐘	814.0 Mbps	1,326 美元
原生備份至具 有 2 TB gp3 的 S3 File Gateway m6i.4xlarge (16	資料庫	00 : 23 : 20 分鐘	731.5 Mbps	470.42 美元

備份選項	Level	執行持續時間 (Appx)	備份速率	USD 每月成本*
v CPU , 64 GB)				
EBS VSS 快照	EBS 磁碟區	00 : 00 : 02 秒 00 : 00 : 53 秒	不適用快照	<u>51 美元</u>
AWS Backup (AMI 備份)	AMI	00 : 00 : 04 秒 00 : 08 : 00 分鐘	不適用快照	<u>75 美元</u>
直接原生SQL 伺服器備份至 Amazon S3 (SQL Server 2022)	資料庫	00 : 12 : 00 分鐘	731.5 Mbps	<u>前 50 TB/月 , 每 GB \$0.023 , 每 月 \$23.55</u>
FSx 適用於 的原生備份 ONTAP (使用 SnapCenter)	資料庫	—	—	<u>\$440.20</u>

上表假設下列事項：

- 不包括資料傳輸和 Amazon S3 成本。
- 儲存體價格包含在執行個體定價中。
- 成本以 us-east-1 區域為基礎。
- 輸送量和 IOPS 成長 10%，而多個備份的整體變化率為當月 10%。

測試結果顯示最快的選項是 Windows File SQL Server FSx 的原生伺服器資料庫備份。備份至 Storage Gateway 和本機連接磁碟 EBS 區是更具成本效益的選項，但效能較慢。對於伺服器層級備份 (AMI)，我們建議您使用 AWS Backup 以獲得最佳效能、成本和可管理性。

成本最佳化建議

了解在 Amazon 上備份 SQL Server 的可能解決方案，EC2是保護資料、確保您符合備份需求，以及制定從關鍵事件復原的計劃的關鍵。本節中探索的備份和還原SQL伺服器執行個體和資料庫的不同方法，可協助您設計備份和還原策略，以保護資料並滿足組織的需求。

本節涵蓋下列備份選項：

- 壓縮
- Amazon S3 Intelligent-Tiering
- 單一可用區域
- 備份至 URL

每個選項提供的指南都是高階。如果您想要在您的組織中實作任何這些建議，建議您聯絡帳戶團隊。然後，團隊可以與 Microsoft 專家 SA 互動以領導對話。您也可以透過電子郵件聯絡 optimize-microsoft@amazon.com。

總而言之，我們建議下列事項：

- 如果您使用的是 SQL Server 2022，則備份到 Amazon S3 是最符合成本效益的選項。
- 如果您使用的是 SQL Server 2019 和更早的SQL伺服器版本，請考慮將 Amazon S3 支援的 Storage Gateway 備份為最具成本效益的選項。

壓縮

壓縮的目標是減少每個備份耗用的儲存，這有利於各種儲存選項。您必須啟用SQL伺服器[SQL執行個體](#)層級的伺服器備份壓縮。下列範例示範如何使用備份資料庫新增壓縮關鍵字：

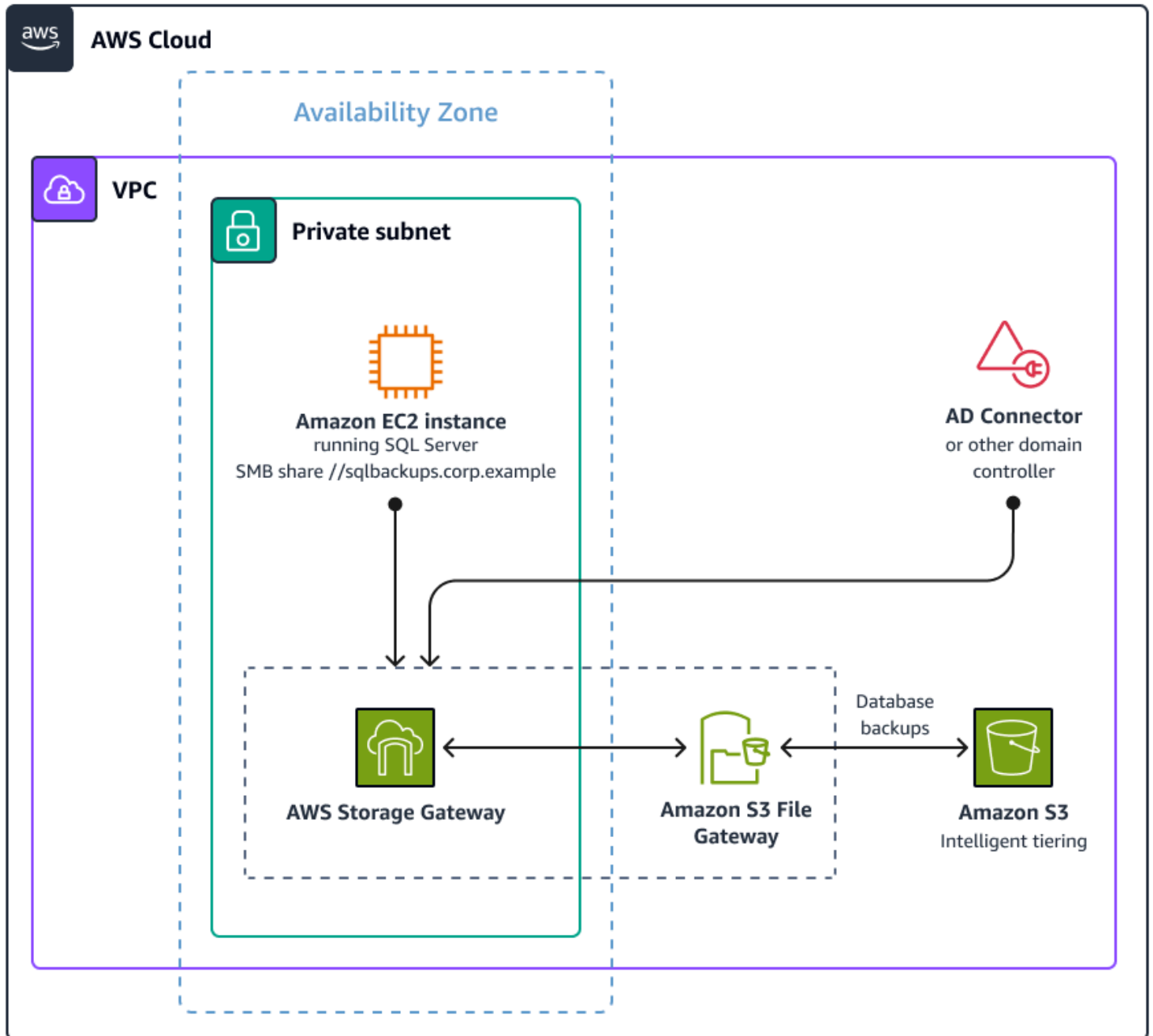
```
BACKUP DATABASE <database_name> TO DISK WITH COMPRESSION (ALGORITHM =
QAT_DEFLATE)
```

Amazon S3 Intelligent-Tiering

對於前往 Amazon S3 儲存貯體的備份，您可以將 [Amazon S3 Intelligent-Tiering](#) 啟用為 Amazon S3 File Gateway [儲存類別](#)。這可以減少高達 30% 的儲存成本。然後，您可以使用可與 [Active Directory 網域](#) 整合SMB的檔案共用，將 S3 File Gateway 掛載到您的SQL伺服器。這為您提供共用的存取控制、利用現有服務帳戶的能力，以及使用常見 Microsoft 集中檔案通訊協定存取 Amazon S3。對於可

能沒有直接連線至網域控制器的帳戶，您可以使用 [Active Directory Connector](#) 來促進與內部部署或雲端中的 Active Directory 通訊。若要在閘道上設定 Active Directory 設定，您必須為網域控制器指定 Active Directory Connector，以將請求代理至 Active Directory。

下圖顯示以 S3 Intelligent-Tiering 為基礎的解決方案架構。



根據預設，寫入 S3 儲存貯體的備份檔案會使用 Standard 層。若要將備份檔案從標準層轉換為 S3 Intelligent-Tiering，您必須 [建立生命週期規則](#)。您也可以使用 [AWS Management Console](#) 來啟用 S3 Intelligent-Tiering。如需詳細資訊，請參閱 文件中的 AWS [開始使用 Amazon S3 Intelligent-Tiering](#)。

單一可用區域

若要建立單一可用區域檔案系統，請在[建立 FSx for Windows File Server 檔案系統](#) 時選擇單一可用區域選項。Amazon FSx 也會每天使用 Windows Volume Shadow Copy Service 來取得檔案系統的高度耐久備份（存放在 Amazon S3 中），並可讓您隨時進行其他備份。請記住使用單一可用區域的一些問題。例如，如果佈建 SMB 檔案系統受影響的可用區域一次中斷數小時，則無法存取檔案共用。如果您需要存取資料，則必須從來源區域中可用可用區域中的備份還原資料。如需詳細資訊，請參閱本指南的[使用單一可用區域](#)一節。

備份至 URL

對於 SQL Server 2022，[備份至 URL](#) 功能允許直接備份至 Amazon S3。這是在中執行 SQL 的 Server 2022 的理想備份方法 AWS，因為您可以在儲存層取得完整的 Amazon S3 功能集，並移除先前版本中所需的設備成本 AWS Storage Gateway，以促進此功能。實作此功能時，需要考量兩個主要成本：資料傳輸成本和所選 S3 儲存體類別。如果您想要 Amazon S3 的原生災難復原功能，則必須考量[該跨區域複寫](#)會產生跨區域[資料輸出成本](#)。若要進一步了解如何設定此選項，請參閱部落格 Microsoft AWS 工作負載上的將[SQL 伺服器資料庫備份至 Amazon S3](#) 文章。

其他資源

- [Amazon 上 SQL 伺服器的備份和還原選項 EC2](#) (AWS 規範指南)
- [Point-in-time RDS 使用 \(儲存部落格 \) 復原和持續備份 Amazon AWS Backup AWS](#)
- [使用 NetApp SnapCenter 搭配 Amazon FSx for \(儲存部落格 \) 保護您的 SQL 伺服器工作負載 NetApp ONTAP AWS](#)
- [開始使用 Amazon S3 Intelligent-Tiering](#) (AWS 入門資源中心)
- [Amazon RDS for SQL Server 的備份和還原策略](#) (AWS 資料庫部落格)
- [將內部部署 Microsoft SQL Server 資料庫遷移至 Amazon EC2](#) (AWS 規範指南)
- [在 Amazon 上部署 Microsoft SQL Server 的最佳實務 EC2](#) (AWS 白皮書)

現代化 SQL 伺服器資料庫

概觀

如果您正在邁向現代化舊式資料庫的旅程，以實現可擴展性、效能和成本最佳化，則伺服器等商業資料庫可能會面臨挑戰 SQL。商業資料庫昂貴、鎖定客戶並提供懲罰性授權條款。本節提供從 SQL 伺服器遷移和現代化到開放原始碼資料庫的選項的高階概觀，以及為工作負載選擇最佳選項的相關資訊。

您可以將SQL伺服器資料庫重構為 Amazon Aurora PostgreSQL 等開放原始碼資料庫，以節省 Windows 和SQL伺服器授權成本。Aurora 等雲端原生現代資料庫結合了開放原始碼資料庫的彈性和低成本，以及商業資料庫的強大企業級功能。如果您有可變工作負載或多租戶工作負載，您也可以遷移至 [Aurora 無伺服器 V2](#)。視工作負載特性而定，這可將成本降低 90%。此外，AWS 還提供 [Babelfish for Aurora Postgre SQL](#)、[AWS Schema Conversion Tool \(AWS SCT \)](#) 等工具以及 [AWS Database Migration Service \(AWS DMS \)](#) 等服務等功能，以簡化上SQL伺服器資料庫的遷移和現代化 AWS。

資料庫方案

從 Windows 上的 SQL Server 遷移至 Amazon Aurora、Amazon RDS for My SQL或 Amazon RDS for PostgreSQL 等開放原始碼資料庫，可以大幅節省成本，而不會影響效能或功能。考慮下列各項：

- 從 Amazon 上的 SQL Server Enterprise 版本切換EC2到 Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL 可以節省高達 80% 的成本。
- 從 Amazon 上的 SQL Server Enterprise 版本切換EC2到 Amazon Aurora Postgre SQL-Compatible Edition 或 Amazon Aurora My SQL-Compatible Edition 可以節省高達 70% 的成本。

對於傳統資料庫工作負載，Amazon RDS for PostgreSQL 和 Amazon RDS for MySQL 地址需求，並為關聯式資料庫提供符合成本效益的解決方案。Aurora 增加了許多可用性和效能功能，先前僅限於昂貴的商業廠商。Aurora 中的彈性功能是額外的成本。不過，與其他商業廠商的類似功能相比，Aurora 的彈性成本仍然比相同類型功能的商業軟體收費便宜。Aurora 架構經過最佳化，相較於標準 MySQL 和 PostgreSQL 部署，可大幅改善效能。

由於 Aurora 與開放原始碼 PostgreSQL 和我的SQL資料庫相容，因此具有可攜性的額外優勢。無論最佳選項是 Amazon RDS for Postgre SQL、Amazon RDS for My SQL或 Aurora，都是為了了解業務需求，並將必要的功能映射到最佳選項。

Amazon RDS和 Aurora 比較

下表摘要說明 Amazon RDS和 Amazon Aurora 之間的主要差異。

類別	Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL	Aurora PostgreSQL 或 Aurora MySQL
效能	良好的效能	效能提升 3 倍
容錯移轉	通常為 60–120 秒*	通常為 30 秒

類別	Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL	Aurora PostgreSQL 或 Aurora MySQL
可擴展性	最多 5 個僅供讀取複本 以秒為單位的延遲	最多 15 個僅供讀取複本 以毫秒為單位的延遲
儲存	高達 64 TB	高達 128 TB
儲存 HA	具有一或兩個待命的多可用區，每個待命區都有資料庫複本	預設 3 個可用區域 6 個資料複本
備份	每日快照和日誌備份	連續、非同步備份至 Amazon S3
Aurora 的創新	NA	100 GB 快速資料庫複製
	自動擴展僅供讀取複本	
	查詢計劃管理	
	Aurora Serverless	
	具有全域資料庫的跨區域複本	
	叢集快取管理**	
	平行查詢	
	資料庫活動串流	

*大型交易可能會增加容錯移轉時間

**在 Aurora Postgre 中提供SQL

下表顯示本節中涵蓋的不同資料庫服務的估計每月成本。

資料庫服務	USD 每月成本*	AWS Pricing Calculator (需要 AWS 帳戶)
Amazon RDS for SQL Server Enterprise 版	3,750 美元	估算
Amazon RDS for SQL Server Standard 版	2,318 美元	估算
SQL Amazon 上的 Server Enterprise 版本 EC2	2,835 美元	估算
SQL Amazon 上的 Server Standard 版本 EC2	1,345 美元	估算
Amazon RDS for PostgreSQL	742 美元	估算
Amazon RDS for MySQL	712 美元	估算
Aurora PostgreSQL	1,032 美元	估算
Aurora MySQL	1,031 美元	估算

* 儲存體價格包含在執行個體定價中。成本以 us-east-1 區域為基礎。輸送量 和 IOPS 是假設。計算適用於 r6i.2xlarge 和 r6g.2xlarge 執行個體。

成本最佳化建議

異質資料庫遷移通常需要將資料庫結構描述從來源轉換為目標資料庫引擎，並將資料從來源遷移至目標資料庫。遷移的第一步是評估 SQL 並將伺服器結構描述和程式碼物件轉換為目標資料庫引擎。

您可以使用 [AWS Schema Conversion Tool \(AWS SCT \)](#) 來評估資料庫與各種目標開放原始碼資料庫選項的相容性 SQL，例如 Amazon RDS for MySQL 或 Amazon RDS for PostgreSQL、Aurora MySQL 和 PostgreSQL。您也可以使用 Babelfish Compass 工具來評估與 Babelfish for Aurora PostgreSQL 的相容性 SQL。這使得 AWS SCT 和 Compass 功能強大的工具在決定遷移策略之前，先了解相關的前期工作。如果您決定繼續，會將結構描述所需的變更 AWS SCT 自動化。Babelfish Compass 背後核心理念是允許 SQL 資料庫移至 Aurora，無需或很少修改。Compass 將評估現有的 SQL 資料庫，以確定是否可以完成此操作。如此一來，在將資料從 SQL 伺服器遷移到 Aurora 之前，就會知道結果。

AWS SCT 會自動轉換和遷移資料庫結構描述和程式碼至目標資料庫引擎。您可以使用 Babelfish for Aurora PostgreSQL 將資料庫和應用程式從 SQL 伺服器遷移至 Aurora PostgreSQL，而不需要或最少的結構描述變更。這可以加速遷移。

遷移結構描述後，您可以使用 AWS DMS 遷移資料。AWS DMS 可以執行完整資料載入並複寫變更，以執行遷移，並盡可能縮短停機時間。

本節會更詳細地探索下列工具：

- AWS Schema Conversion Tool
- Babelfish for Aurora PostgreSQL
- Babelfish Compass
- AWS Database Migration Service

AWS Schema Conversion Tool

您可以使用 AWS SCT 來評估現有的 SQL 伺服器資料庫，並評估與 Amazon RDS 或 Aurora 的相容性。若要簡化遷移程序，您也可以使用 AWS SCT 將結構描述從一個資料庫引擎轉換為異質資料庫遷移中的另一個引擎。您可以使用 AWS SCT 來評估應用程式，並轉換內嵌應用程式程式碼，用於撰寫 C#、C++、Java 和其他語言的應用程式。如需詳細資訊，請參閱文件中的 AWS SCT [SQL 使用 轉換 應用程式 AWS SCT](#)。

AWS SCT 是支援許多資料庫來源的免費 AWS 工具。若要使用 AWS SCT，請將它指向來源資料庫，然後執行評估。然後，[AWS SCT](#) 評估結構描述並產生評估報告。評估報告包括執行摘要、複雜性和遷移工作、適當的目標資料庫引擎，以及轉換建議。若要下載 AWS SCT，請參閱文件中的 [AWS SCT 安裝、驗證和更新 AWS SCT](#)。

下表顯示 AWS SCT 產生的範例執行摘要，以顯示將資料庫變更為不同目標平台所涉及的複雜性。

目標平台	自動或最小變更		複雜動作				
	儲存物件	程式碼物件	轉換動作	儲存物件	程式碼物件		
Amazon RDS for MySQL	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56

Amazon Aurora My SQL- Compatible Edition	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56
Amazon RDS for PostgreSQL	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Amazon Aurora PostgreSQL- 相容版本	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Amazon RDS for MariaDB	60 (98%)	7 (30%)	42	1 (2%)	1	16 (70%)	58
Amazon Redshift	61 (100%)	9 (39%)	124	0 (0%)	0	14 (61%)	25
AWS Glue	0 (0%)	17 (100%)	0	0 (0%)	0	0 (0%)	0
Babelfish	59 (97%)	10 (45%)	20	2 (3%)	2	12 (55%)	30

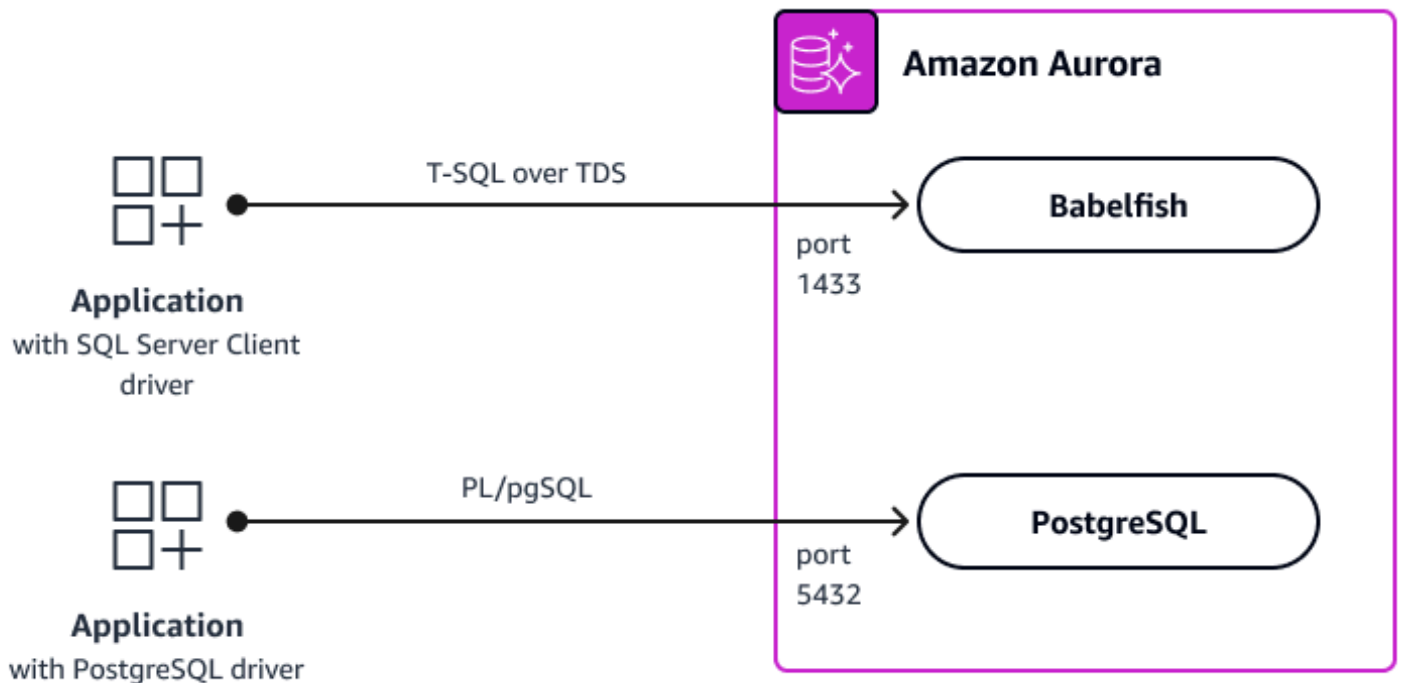
AWS SCT 報告也提供無法自動轉換的結構描述元素詳細資訊。您可以參考[AWS 遷移手冊](#)來關閉 AWS SCT 轉換差距並最佳化目標結構描述。有許多資料庫遷移手冊可協助異質遷移。

Babelfish for Aurora PostgreSQL

Babelfish for Aurora PostgreSQL 延伸了 Aurora PostgreSQL，能夠接受來自 SQL 伺服器用戶端的資料庫連線。Babelfish 可讓最初為 SQL 伺服器建置的應用程式直接與 Aurora PostgreSQL 搭配使用 SQL，

且程式碼幾乎不會變更，而且不會變更資料庫驅動程式。Babelfish 將 Aurora PostgreSQL 轉換為雙語，以便 Aurora PostgreSQL 可以使用 T-SQL 和 PL/pgSQL 語言。Babelfish 會將從 SQL 伺服器遷移到 Aurora Postgre 的努力降至最低 SQL。這可加速遷移、將風險降至最低，並大幅降低遷移成本。您可以繼續使用 T-SQL 後遷移，但也可以[選擇使用 PostgreSQL 原生工具](#)進行開發。

下圖說明使用 T-SQL 的應用程式如何連線至 SQL 伺服器中的預設連接埠 1433，並使用 Babelfish 翻譯程式與 Aurora PostgreSQL 資料庫通訊，而使用 PL/pgSQL 的應用程式可以使用 Aurora PostgreSQL 中的預設連接埠 5432 直接並同時連線至 Aurora Postgre 資料庫 SQL。



Babelfish 不支援某些 SQL Server T-SQL 功能。因此，Amazon 提供評估工具來 line-by-line 分析您的 SQL 陳述式，並判斷 Babelfish 是否不支援其中任何陳述式。

Babelfish 評估有兩個選項。AWS SCT 可以評估您的 SQL 伺服器資料庫與 Babelfish 的相容性。另一個選項是 Babelfish Compass 工具，這是建議的解決方案，因為 Compass 工具會與 Babelfish for Aurora Postgre 的新版本更新 SQL。

Babelfish Compass

[Babelfish Compass](#) 是免費下載的工具，與最新版本的 Babelfish for Aurora Postgre 保持一致 SQL。相反地，AWS SCT 會在一段時間後支援較新的 Babelfish 版本。[Babelfish Compass](#) 會針對 SQL 伺服器資料庫結構描述執行。您也可以使用 SQL Server Management Studio () 等工具來擷取來源 SQL 伺服器資料庫結構描述 SSMS。然後，您可以透過 Babelfish Compass 執行結構描述。這會產生報告，詳

細說明SQL伺服器結構描述與 Babelfish 的相容性，以及在遷移之前是否需要進行任何變更。Babelfish Compass 工具也可以自動化許多這些變更，最終加速遷移。

評估和變更完成後，您可以使用 SSMS或 sqlcmd 等SQL伺服器原生工具，將結構描述遷移至 Aurora PostgreSQL。如需指示，請參閱 AWS 資料庫部落格上的[使用 Babelfish 文章從SQL伺服器遷移至 Amazon Aurora](#)。

AWS Database Migration Service

遷移結構描述後，您可以使用 AWS Database Migration Service (AWS DMS) 將資料遷移至 AWS，並縮短停機時間。AWS DMS 不僅會執行完整的資料載入，還會在來源系統啟動和執行時，將變更從來源複寫至目的地。來源和目標資料庫同步後，切換活動可能會發生，其中應用程式指向完成遷移的目標資料庫。AWS DMS 目前僅針對 Aurora PostgreSQL 目標使用 Babelfish 執行完整資料載入，且不會複寫變更。如需詳細資訊，請參閱 AWS DMS 文件中[使用 Babelfish 作為的目標 AWS Database Migration Service](#)。

AWS DMS 可以同時執行同質（在同一個資料庫引擎之間）和異質（在不同資料庫引擎之間）遷移。AWS DMS 支援許多來源和目的地資料庫引擎。如需詳細資訊，請參閱 AWS 資料庫部落格中的[使用文章將SQL伺服器資料庫遷移至 Amazon RDS for SQL Server AWS DMS](#)。

其他資源

- [再見 Microsoft SQL Server , Hello Babelfish](#) (AWS 新聞部落格)
- [SQL 使用 AWS Schema Conversion Tool 轉換資料庫結構描述和應用程式 CLI](#) (AWS 資料庫部落格)
- [使用從 欄位學習的最佳實務和課程，將SQL伺服器遷移至 Amazon Aurora PostgreSQL](#) (AWS 資料庫部落格)
- [驗證從 Microsoft SQL Server 遷移至 Amazon RDS for PostgreSQL 和 Amazon Aurora Postgre 的資料庫物件SQL](#) (AWS 資料庫部落格)

最佳化SQL伺服器儲存

概觀

本節著重於EC2工作負載中 Amazon Elastic Block Store (Amazon EBS) SQL Server SSD儲存體的成本最佳化。

您可以在 上部署和執行SQL伺服器工作負載時，有多種儲存選項 AWS。選取正確的儲存體應基於用途、架構、耐久性、效能、容量和成本。執行SQL伺服器工作負載 AWS 的客戶通常會使用 Amazon EBS、NVMe、Amazon FSx和 Amazon Simple Storage Service (Amazon S3) 儲存體的組合。

Amazon EBS是連接至EC2運算執行個體的網路連接儲存體，用於儲存和處理一般作業系統、應用程式、資料庫和備份檔案。Amazon EBS 固態硬碟 (SSD) 儲存包含一般用途 SSD (gp2 和 gp3) 和佈建 IOPSSSD (io1、io2 和 io2BX)。考慮下列各項：

- 有些EC2執行個體，例如 r5d，已將本機NVMeSSDs實體連接至主機執行個體。這些磁碟區提供區塊層級儲存，常用於SQL伺服器 tempdb 或緩衝集區延伸。
- Amazon FSx for Windows File Server 是完全受管的檔案儲存服務，而 FSx 的 NetApp ONTAP Amazon 則是建立在 NetApp熱門ONTAP檔案系統的完全受管共用儲存。Amazon FSx 經常用於在高可用性、SQL伺服器容錯移轉叢集執行個體 (FCI) 組態中執行SQL伺服器工作負載。此解決方案託管SQL伺服器資料和日誌檔案，減少EC2執行個體EBS的效能需求。
- Amazon S3 是一種物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。您可以在 Amazon S3 上儲存SQL伺服器原生備份檔案、AMIs、EBS快照、應用程式日誌等。

SSD Amazon 的儲存類型、效能和成本 EBS

SSD Amazon 的儲存成本EBS通常會隨著耐久性和效能的增加而增加。儲存體目前有五種磁碟區類型，每個類型都有[自己的獨特效能指標](#)。如需 SSD後端磁碟區的使用案例和特性摘要，請參閱 Amazon EBS 文件的[固態硬碟 \(SSD \) 磁碟區](#)區段中的表格。

您可以使用 Amazon CloudWatch 來監控SSD效能、擷取趨勢資料，並在達到特定閾值時設定警示。如果您是在 上執行SQL伺服器工作負載 AWS，請考慮啟用[詳細的監控](#)和部署[CloudWatch 自訂指標](#)，以擷取詳細的磁碟延遲、IOPS、輸送量、磁碟佇列長度、使用與可用容量等磁碟效能指標。您可以使用這些 CloudWatch 效能指標來識別佈建不足和佈建過度的儲存，並提供歷史資料點來準確定義儲存需求。

SSD Amazon 的儲存成本EBS也會因配置的容量而有所不同。下表顯示不同磁碟區類型的比較。所有磁碟區類型都有 1 TB 的容量和類似的效能組態。

磁碟區類型	最大值 IOPS (16 KiB I/O)	最大輸送量 (128 KiB I/O)	每 1TB 的價格	節省成本百分比
gp2	3,000	250	102.40 美元	

磁碟區類型	最大值 IOPS (16 KiB I/O)	最大輸送量 (128 KiB I/O)	每 1TB 的價格	節省成本百分比
gp3	3,000	250	86.92 美元	15%
io1	16,000	500	1,168 美元	
io2	16,000	500	1,168 美元	
gp3	16,000	500	146.92 美元	87%
io2bx	16,000	4,000	1,168 美元	
gp3	16,000	1,000	181.92 美元	84%

Note

上表中的效能和成本指標是根據的 [估計值](#)，以每個磁碟區為基準 AWS Pricing Calculator。AWS 帳戶 需要 才能存取 中的預估 AWS Pricing Calculator。

Amazon EBS SSD gp3 磁碟區以低成本提供卓越的效能。對於需要低於 16,000 IOPS 和 500 輸送量的工作負載，如果您在 io1 或 io2 磁碟區上選擇 gp3 MiBps 磁碟區，則最多可節省 87%。

相較於一般 io2 磁碟區，io2 Block Express (io2BX) 磁碟區可提供更高的效能。在 16,000 時 IOPS，io1 或 io2 磁碟區只能傳輸 500 MiBps 個輸送量，而 io2BX 磁碟區最多可設定 4,000 MiBps 個傳輸量。與 io1 和 io2 磁碟區相比，io2BX 磁碟區以完全相同的價格提供介於 16,000 到 64,000 IOPS 之間的輸送量四倍以上。將一般 io2 磁碟區連接至 io2BX io2BX-supported 磁碟區。EC2 如需 io2BX-supported 的 EC2 執行個體清單，請參閱 Amazon EBS 文件中的 [佈建磁碟 IOPS SSD](#) 區。在部署新儲存體之前，您可以使用 [AWS Pricing Calculator](#) 來估算每月成本，並根據持久性、效能和容量之間的權衡，了解對成本的影響。

Amazon 的一般 SSD 成本最佳化 EBS

建議您評估要儲存的內容，並確保您使用正確的儲存類型和類別。例如，Amazon S3 提供很棒的價格點、內建生命週期政策和複寫選項，非常適合 SQL 伺服器備份。SQL Server 2022 可以直接備份至 Amazon S3，而舊版的 SQL 伺服器依賴本機備份。如果您正在執行較舊版本的 SQL 伺服器，請考慮備

份至 Amazon EBSHDD磁碟區，然後將備份複製到 Amazon S3。相較於使用 gp3 磁碟區進行備份，此解決方案可以節省 53% 的費用。

下表顯示 Amazon gp3、Amazon EBS HDD st1 和 Amazon S3 EBS 上 1 TB 儲存體的價格差異。

儲存體類型	容量	價格 pm
EBS gp3 500 MiBps	1 TB	96.92 美元
EBS st1 爆量 500 MiBps		46.08 美元
S3 Standard		23.55 美元
S3 標準 (不常存取)		12.80 美元
S3 Glacier Deep Archive		1.03 美元

Note

上表中的成本指標是以中的[估計值](#)為基礎 AWS Pricing Calculator。AWS 帳戶需要才能存取中的預估 AWS Pricing Calculator。

建議您考慮下列事項：

- 啟用詳細的監控和部署 CloudWatch 自訂指標，以準確擷取其儲存效能需求。
- 將 Amazon EBS儲存體從 gp2 升級到 gp3，以降低成本、提高靈活性並改善效能。
- 將 Amazon EBS儲存體從 io1 升級到 io2，以提高耐用性和效能彈性。
- 盡可能使用 io2BX 而非 io1 或 io2，以提高耐用性和效能。
- 在選擇儲存體時，請考慮採用有助於 mix-and-match降低容量需求和高效能磁碟區成本的方法。例如，您可以將低成本 gp3 磁碟區用於根磁碟區（作業系統）、SQL伺服器安裝、系統資料庫（不包括 tempdb），以及效能較低的使用者資料庫。這有助於降低 io2 磁碟區的容量和成本，而 io2 磁碟區專用於高效能使用者資料庫。
- 如果您是在上託管SQL伺服器資料庫 AWS，我們建議您在每個資料庫使用多個SQL伺服器資料檔案。這可讓有機會在多個磁碟區之間分配讀取/寫入工作負載，進而降低每個磁碟區的效能和容量需求，進而降低成本。

- 即使生產工作負載需要更高效能的儲存，例如 io1 或 io2/io2BX，考慮非生產工作負載的 gp3 磁碟區，以協助降低成本。
- 隨著時間的推移，追蹤和趨勢儲存使用率，以輕鬆識別使用量激增和意外成本。
- [AWS Compute Optimizer](#) 用於根據實際使用率向上或向下擴展EBS磁碟區的建議。
- 使用的彈性 AWS 來調整 Amazon 磁碟SSD區的效能和容量需求EBS。與內部部署環境不同，您不需要為未來的工作負載過度佈建儲存效能和容量。您可以將現有的SQL伺服器工作負載遷移至 AWS，並視需要調整效能或容量，同時保持資料庫上線。

其他資源

- [Amazon EBS磁碟區類型](#) (Amazon EBS 文件)
- [Amazon Elastic Block Store \(Amazon EBS \)](#) (Amazon EBS 文件)
- [佈建磁碟IOPSSSD區](#) (Amazon EBS 文件)
- [SSD 執行個體存放區磁碟區](#) (Amazon EC2 文件)
- [Amazon 的 Amazon CloudWatch 指標 EBS](#) (Amazon EBS 文件)
- [Amazon EC2儲存最佳化執行個體的規格](#) (Amazon EC2 文件)
- [使用 NetApp SnapCenter 搭配 Amazon FSx for \(Storage 部落格 \) 保護您的SQL伺服器工作負載 NetApp ONTAPAWS](#)
- [Amazon EC2FAQ](#) (AWS 產品頁面)

使用 Compute Optimizer 最佳化SQL伺服器授權

如何使用 最佳化SQL伺服器授權的指南 AWS Compute Optimizer。

概觀

[AWS Compute Optimizer](#) 可以在 Amazon Elastic Compute Cloud (Amazon) 上建議 Microsoft SQL Server 工作負載的授權最佳化機會EC2。Compute Optimizer 可以提供自動化建議，以降低授權成本。Compute Optimizer 的建議會列在具有 Microsoft SQL Server 授權的每個EC2執行個體旁。提供的資訊包括建議的儲存機會、EC2執行個體隨需價格，以及每小時自帶授權 (BYOL) 價格。此資訊可協助您決定是否應該降級授權版本。

Compute Optimizer EC2 會根據推斷的工作負載類型自動探索 Amazon 上的SQL伺服器執行個體。若要檢視授權建議，您可以在 Compute Optimizer 中選取SQL伺服器執行個體，然後使用唯讀資料

庫憑證，透過 [Amazon CloudWatch Application Insights](#) 進行身分驗證。Compute Optimizer 會分析您是否使用任何 SQL Server Enterprise 版本功能。如果沒有使用 Enterprise Edition 功能，Compute Optimizer 建議您降級至 Standard Edition，以減少授權成本。

您也可以使用 Compute Optimizer 為執行SQL伺服器工作負載的 Amazon EC2執行個體提出大小調整建議。如需詳細資訊，請參閱本指南中的[使用 Compute Optimizer 最佳化SQL伺服器大小](#)。

成本最佳化建議

Compute Optimizer 中的授權建議可協助您評估您在 Microsoft SQL Server 中使用的功能，並為工作負載選擇最具成本效益的版本。SQL Server Enterprise Edition 比 Standard Edition 更昂貴。如需詳細資訊，請參閱本指南中的[比較SQL伺服器版本](#)，並參閱 Microsoft 網站上的 [SQL Server 2022 定價](#)。投資時間來設定 Compute Optimizer 來評估您的SQL伺服器機群並提供建議，可以大幅降低您的授權成本。

授權詳細資訊頁面提供下列資訊：

- 使用資料表來比較您目前的授權設定（例如版本、模型和執行個體核心數量）與 Compute Optimizer 建議。
- 使用使用率圖表來檢閱在分析期間使用的企業版本功能數目。

如需詳細資訊，請參閱 Compute Optimizer 文件中的[檢視商業軟體授權建議的詳細資訊](#)。

設定 Compute Optimizer

Compute Optimizer 使用 `mssql_enterprise_features_used` 指標來分析商業軟體授權。如需此指標的詳細資訊，請參閱[商業軟體授權的指標](#)。

1. 請確定您有適當的許可，可以選擇加入 Compute Optimizer。如需詳細資訊，請參閱下列內容：
 - [選擇加入 Compute Optimizer 的政策](#)
 - [授予獨立運算最佳化工具存取權的政策 AWS 帳戶](#)
 - [為組織的管理帳戶授予 Compute Optimizer 存取權的政策](#)
2. 連接 CloudWatch Application Insights 所需的執行個體角色和政策。如需指示，請參閱[政策以啟用商業軟體授權建議](#)。
3. 使用您的 Microsoft SQL Server 資料庫憑證啟用 CloudWatch Application Insights。如需指示，請參閱 CloudWatch 文件中的[設定用於監控的應用程式](#)。

Note

若要產生商業軟體授權的建議，至少需要連續 30 小時的 CloudWatch 指標資料。如需詳細資訊，請參閱[CloudWatch 指標需求](#)。

4. 使用下列 SQL 查詢來設定 CloudWatch Application Insights 的最低權限存取權。

```
GRANT VIEW SERVER STATE TO [LOGIN];  
GRANT VIEW ANY DEFINITION TO [LOGIN];
```

這會啟用新的服務 PrometheusSqlExporterSQL。

5. 從目標 AWS 帳戶 或組織管理帳戶中，選擇加入 Compute Optimizer。如需指示，請參閱[選擇加入您的帳戶](#)。

Note

選擇加入後，調查結果和最佳化建議最多可能需要 24 小時才能產生。

6. 在 [Compute Optimizer 主控台](#) 中，選擇導覽窗格中的 授權。
7. 在調查結果欄中，搜尋指標調查結果不足的任何執行個體。如果 Compute Optimizer 偵測到 CloudWatch Application Insights 未啟用或許可不足，則會傳回此調查結果。如需詳細資訊，請參閱[尋找原因](#)。執行下列動作以解決這些調查結果：
 - a. 選擇執行個體。
 - b. 新增秘密。
 - c. 確認已連接執行個體角色和政策。
 - d. 選擇啟用授權建議。
8. 在調查結果欄中，搜尋具有未最佳化調查結果的任何執行個體。如果 Compute Optimizer 偵測到您的 Amazon EC2 基礎設施未使用您支付的任何 Microsoft SQL Server 授權功能，則會傳回此調查結果。如需詳細資訊，請參閱[尋找原因](#)。執行下列動作以解決這些調查結果：
 - a. 選擇執行個體。
 - b. 比較目前的授權版本與建議的版本。
 - c. 檢閱目前的授權使用率圖表。
 - d. 如果您想要降級授權，請選擇實作建議。

- e. 檢閱需求並遵循指示來降級授權。如果您想要自動化程序，請參閱[使用 AWS Systems Manager 文件進行降級SQL伺服器企業版本以降低成本](#)（AWS 部落格）。

其他資源

- [使用（部落格）降低 Microsoft SQL Server 授權成本 AWS Compute Optimizer](#) AWS
- [什麼是 AWS Compute Optimizer？](#)（AWS 文件）
- [檢視商業軟體授權建議](#)（AWS 文件）
- [降級 Microsoft SQL Server 版本](#)（AWS 文件）
- [Microsoft SQL Server on AWS](#)（AWS）
- [Microsoft Licensing on AWS](#)（AWS）
- [Microsoft SQL Server 2019 定價](#)（Microsoft）
- [Microsoft SQL Server 2022 定價](#)（Microsoft）

使用 Compute Optimizer 最佳化SQL伺服器大小

概觀

[AWS Compute Optimizer](#) 協助資料庫管理員（DBAs）在 Amazon Elastic Compute Cloud（Amazon EC2）上探索 Microsoft SQL Server 工作負載，並授權 EC2 執行個體以降低高達 25% 的許可成本。Compute Optimizer 中的[推論工作負載類型](#)功能會使用機器學習（ML），並自動偵測可能在 AWS 資源上執行的應用程式。Compute Optimizer 包含對 SQL 伺服器支援作為推斷工作負載類型。透過使用推斷的工作負載類型功能，您可以根據在 Amazon EC2 執行個體上執行的特定工作負載，找出節省成本的機會。

使用此功能，您可以根據支援的推斷工作負載類型，例如 SQL 伺服器，來分類節省成本的機會。Compute Optimizer 可以自動探索過度佈建的 SQL 伺服器 EC2 執行個體。您可以切換到 EC2 主控台來縮減執行個體的大小，這有助於降低授權和基礎設施成本。

您也可以使用 Compute Optimizer 提出 SQL 伺服器授權建議。如需詳細資訊，請參閱本指南中的[使用 Compute Optimizer 最佳化SQL伺服器授權](#)。

設定 Compute Optimizer

如需將 Compute Optimizer 與 SQL 伺服器推論工作負載搭配使用的指示，請參閱[最佳化效能並降低授權成本：AWS Compute Optimizer 為 Amazon EC2 SQL Server 執行個體（部落格）使用](#)。AWS

您可以選擇加入獨立帳戶、組織成員的帳戶，以及組織管理帳戶。對於獨立帳戶和成員帳戶，選擇加入只會啟用該帳戶的 Compute Optimizer。對於組織管理帳戶，您可以選擇是否僅在該帳戶中啟用 Compute Optimizer，還是針對組織的所有成員帳戶啟用 Compute Optimizer。

Compute Optimizer 選擇加入程序會自動建立 AWS Identity and Access Management (IAM) 服務連結角色。如需詳細資訊，請參閱[使用 AWS Compute Optimizer 的服務連結角色](#)。

Compute Optimizer 會根據 Amazon CloudWatch 指標來分析您的資源，例如 CPU、I/O、網路和 Amazon Elastic Block Store (AmazonEBS) 用量。若要產生建議，在過去 14 天內至少需要連續 30 小時的 CloudWatch 指標資料。如果您啟用增強型基礎設施指標功能，則會將使用率指標延長至 93 天。如需詳細資訊，請參閱 Compute Optimizer 文件中的[CloudWatch 指標需求](#)和[增強型基礎設施指標](#)。

Compute Optimizer 會根據 v CPU、記憶體、儲存體、網路、風險和遷移工作，提供與每個選項相關的選項和節省成本。您可以使用 CloudWatch 指標儀表板來分析用於提出建議的資料。使用此資料，您可以授權正在執行 SQL 伺服器工作負載的 EC2 執行個體。如需如何變更執行個體類型的詳細資訊，請參閱 Amazon EC2 文件中的[變更執行個體類型](#)。

其他資源

- [AWS Compute Optimizer 識別和篩選 Microsoft SQL Server 工作負載](#) (AWS)
- [最佳化效能並降低授權成本：AWS Compute Optimizer 為 Amazon EC2 SQL Server 執行個體使用](#) (AWS 部落格)
- [什麼是 AWS Compute Optimizer ?](#) (AWS 文件)
- [檢視 EC2 執行個體建議](#) (AWS 文件)

檢閱 SQL 伺服器工作負載 Trusted Advisor 的建議

概觀

[AWS Trusted Advisor](#) 提供建議，協助您遵循 AWS 最佳實務。透過分析您的用量、組態和支出，Trusted Advisor 提供可行的建議來降低成本、改善系統可用性和效能，或協助消除安全漏洞。本節著重於可協助您降低中操作 SQL 伺服器工作負載成本的 Trusted Advisor 檢查 AWS 雲端。

成本最佳化建議

Trusted Advisor 提供建議，協助您在 Amazon Elastic Compute Cloud (Amazon) 上最佳化SQL 伺服器工作負載EC2。檢查會檢查SQL伺服器工作負載，並自動列出需要最佳化的執行個體。操作化 Trusted Advisor 建議可以降低成本並改善組織的安全狀態。

以下是著重於 Microsoft SQL Server 的 Trusted Advisor 檢查：

- Microsoft [SQL Server 的 Amazon EC2執行個體過度佈建](#) – 此檢查會分析執行 SQL Server 的 Amazon EC2執行個體，並在執行個體超過 SQL Server 軟體 vCPU 限制時提醒您。例如，具有 SQL Server Standard 版本的執行個體最多可以使用 48 個 vCPUs。具有 SQL Server Web 的執行個體最多可以使用 32 個 vCPUs。

版本	vCPU Min.	vCPU 最大值
Web	4	32
標準	4	48
Enterprise	4	作業系統限制

- Microsoft [SQL Server 的 Amazon EC2執行個體整合](#) – 此檢查會分析您的 Amazon EC2執行個體，並在執行個體少於SQL伺服器授權數目下限時提醒您。您可以合併較小的SQL伺服器執行個體，以協助降低成本。如果您有許多小型、包含授權的SQL伺服器執行個體，請考慮合併。根據 [Microsoft SQL Server 2019 授權指南](#)，SQL伺服器每個執行個體至少需要 4 個 vCPU 授權。如果您合併這些資料庫，則可以節省授權成本。您可以根據執行個體上的資料庫數目、資料庫大小上限和資料庫大小總計來做出決策。Web、Standard 和 Enterprise 版本的SQL伺服器支援合併。如需詳細資訊，請參閱[合併SQL伺服器資料庫](#) (Microsoft 部落格文章)。

AWS 不建議僅將大型生產資料庫放在一個伺服器上。不過，您可以合併用於非生產環境的較小項目，例如用於開發、測試和預備。這取決於您目前的SQL伺服器用量；如果您有低用量的資料庫，您可以合併在一個伺服器上。

設定 Trusted Advisor

執行下列動作，以評估 中的SQL伺服器導向檢查 Trusted Advisor。

1. 登入 AWS Management Console。
2. 開啟 [AWS Trusted Advisor 主控台](#)。

3. 在導覽窗格中的建議下，選擇成本最佳化。
4. 在成本最佳化檢查清單中，檢閱 Microsoft SQL Server 的 Amazon EC2 執行個體合併狀態，以及針對 Microsoft SQL Server 檢查過度佈建的 Amazon EC2 執行個體狀態。
 - 綠色勾號表示您的 Amazon EC2 執行個體已進行最佳設定。
 - 橘色警示符號表示有改進的機會。
5. 選擇檢查以查看其詳細資訊和建議。
6. 請遵循檢查提供的指示，最佳化正在執行 SQL 伺服器工作負載的 Amazon EC2 執行個體。
7. 定期監控執行個體，並定期重新整理檢查。

其他資源

- [Trusted Advisor 檢查參考](#) (AWS 文件)
- [Microsoft SQL Server on AWS](#) (AWS)
- [Microsoft Licensing on AWS](#) (AWS)
- [SQL Server 2019 定價](#) (Microsoft)
- [AWS Launch Wizard for SQL Server](#) (AWS 文件)

容器

現代化是一個轉型的旅程，提供許多選項，包括將單位分解為微服務、使用無伺服器函數 (AWS Lambda) 將應用程式重新架構為事件驅動，以及將資料庫從 SQL Server 重新利用到 Amazon Aurora 或專門建置的受管資料庫。與其他現代化選項相比，將 .NET 架構應用程式重新平台到 Linux 和 Windows 容器所需的現代化途徑更少。容器具有以下優點：

- 加速創新 — 移至容器可讓您更輕鬆地將開發生命週期的各個階段自動化，包括建置、測試和部署應用程式。透過將這些流程自動化，開發和營運團隊有更多時間專注於創新。
- 降低總體擁有成本 (TCO) — 移轉至容器也可以減少您對授權管理和端點保護工具的依賴。由於容器是暫時的運算單位，因此您可以自動化並簡化修補、擴展以及備份和還原等管理工作。這減少了管理和操作容器型工作負載的總體擁有成本。最後，與虛擬機器相比，容器更有效率，因為您可以透過提供更好的隔離來使應用程式的放置最大化。這會增加應用程式基礎結構資源的使用率。
- 提高資源使用率 — 與虛擬機器相比，容器更有效率，因為您可以使用容器將應用程式的放置最大化。這樣可以提供更好的隔離來增加應用程式基礎架構資源的使用率。
- 縮小技能差距 — AWS 提供沉浸式的日子，以提高您的開發團隊對容器技術和 DevOps 實踐的技能。

本節涵蓋下列主題：

- [將視窗應用程式移至容器](#)
- [優化 Amazon ECS 上 AWS Fargate 任務的成本](#)
- [掌握您的 Amazon EKS 成本](#)
- [重新平台視窗應用程式與應用程式容器](#)

如需授權資訊，請參閱 [Amazon Web Services](#) 和 [Microsoft：常見問題集的授權部分](#)，或將您的問題以電子郵件傳送至 microsoft@amazon.com。

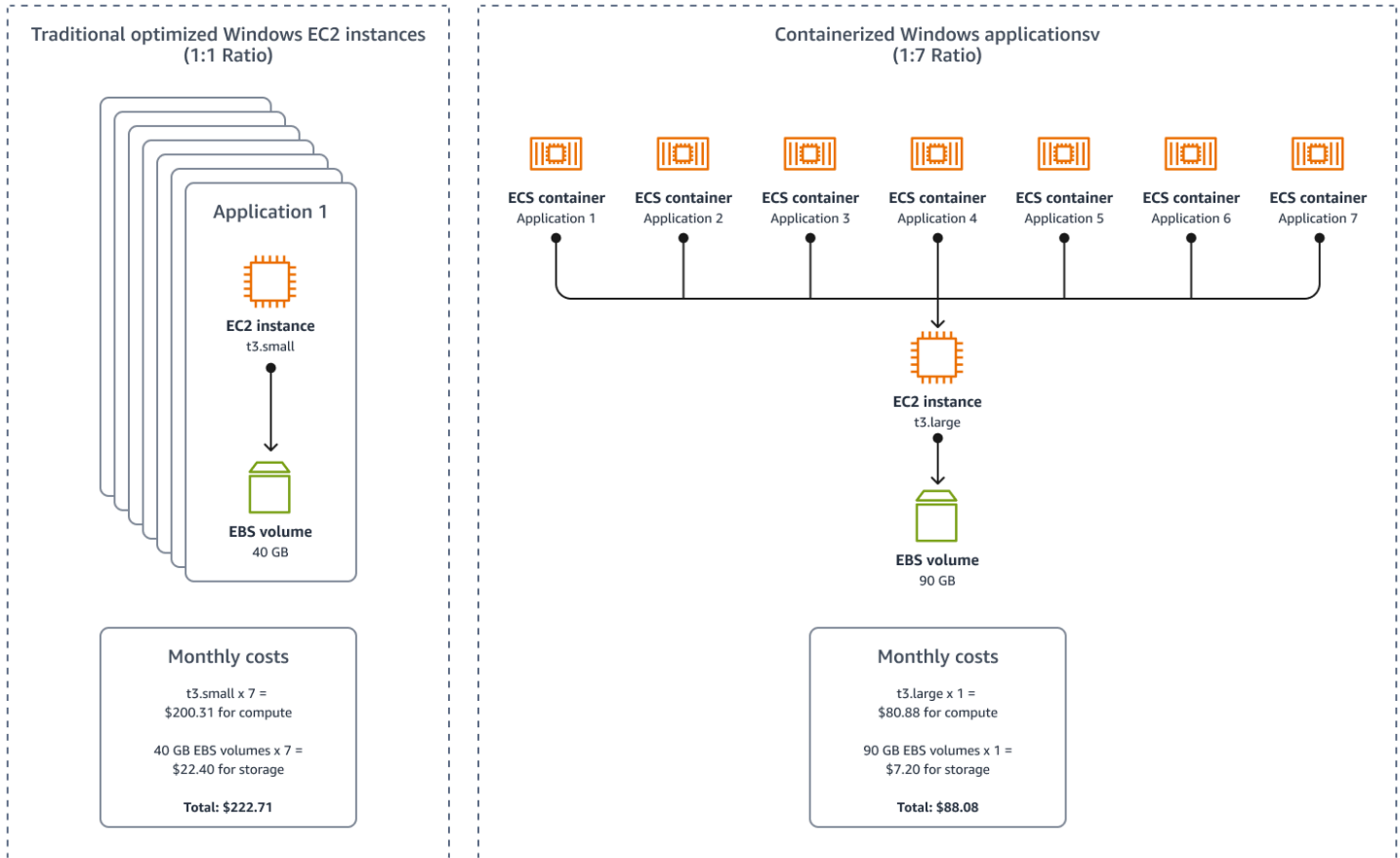
將視窗應用程式移至容器

概觀

根據 [CNCF 2021 年年度調查](#)，96% 的組織正在使用或評估容器來實現基礎架構現代化。這是因為容器可以幫助您的組織降低風險、提高營運效率和速度，並提高靈活性。您也可以使用容器來降低執行應用程式的成本。本節提供跨 AWS 容器服務 (包括 [亞馬遜彈性容器服務 \(Amazon ECS\)](#)、[亞馬遜彈性 Kubernetes 服務 \(Amazon EKS\)](#) 等容器服務) 以符合成本效益的方式執行容器的建議。 [AWS Fargate](#)

成本效益

下列資訊圖表顯示企業根據[AWS 優化和授權評估 \(AWS OLA\)](#) 建議，將 ASP.NET 架構應用程式整合到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體，可節省的成本。下列資訊圖表顯示將應用程式移至 Windows 容器，可達到哪些額外的節省成本。



AWS OLA 建議企業提升並轉移到個別 t3. 小型執行個體。如下列效能使用率分析所示，企業可以在內部部署伺服器上執行七個 ASP.NET 應用程式，以達成這些節省的成本。

Server name	Storage	Operating system	On-premises CPU AVG utilization	On-premises CPU peak utilization	On-premises RAM (GB)	On-premises RAM AVG utilization (GB)	On-premises RAM peak utilization (GB)	Instance size	vCPU	RAM (GB)
1 AppServer01	60	Windows Server 2012	7.00%	17.00%	8	13.50%	17.10%	t3.small	2	2
2 AppServer02	39	Windows Server 2012	20.07%	22.00%	16	7.50%	12.40%	t3.small	2	2
3 AppServer03	39	Windows Server 2012	24.00%	25.50%	16	8.80%	11.90%	t3.small	2	2
4 AppServer04	4	Windows Server 2012	21.40%	24.00%	16	7.80%	10.70%	t3.small	2	2
5 AppServer05	40	Windows Server 2012	21.30%	23.00%	16	8.20%	12.00%	t3.small	2	2
6 AppServer06	39	Windows Server 2012	21.50%	23.50%	16	7.90%	10.90%	t3.small	2	2
7 AppServer07	39	Windows Server 2012	21.60%	22.90%	16	8.40%	11.50%	t3.small	2	2

Annotations above the table:

- Not sharable (points to Storage)
- Nearing end of life (points to Operating system)
- Approximately 15% overhead (points to CPU utilization)
- The OS accounts for 1.2 GB per instance (points to RAM utilization)
- Optimized for EC2 (points to Instance size, vCPU, and RAM)

進一步的分析顯示，企業可以透過在容器上執行工作負載來節省更多成本。容器可減少系統資源 (例如 CPU、RAM 和磁碟使用量) 的作業系統負荷 (在下一節中說明)。在這個案例中，企業可以將所有七個應用程式整合到一個 t3.large 執行個體，而且仍有 3 GB 的 RAM 可供備用。遷移到容器可協助企業透過使用容器而非 Amazon EC2，在運算和儲存間平均節省 64% 的成本。

成本最佳化建議

下一節提供透過合併應用程式和使用容器來最佳化成本的建議。

減少您在 Amazon EC2 上的視窗足跡

Windows 容器可讓您將更多應用程式合併到較少的 EC2 執行個體上，從而減少 Amazon EC2 上的 Windows 佔用空間。例如，假設您有 500 個 ASP.NET 應用程式。如果您在 Amazon EC2 上為每個應用程式運行一個內核，則等於 500 個 Windows 實例 (t3 小)。如果您假設使用 Windows 容器 (使用 t3.large) 的比率為 1:7 (視 EC2 執行個體類型/大小而定)，則您只需要大約 71 個 Windows 執行個體。這表示您在 Amazon EC2 足跡上的視窗減少了 85.8 百分比。

降低視窗授權成本

如果您授權 Windows 執行個體，則不需要授權在該執行個體上執行的容器。因此，使用 Windows 容器整合 ASP.NET 應用程式可以大幅降低您的 Windows 授權成本。

減少儲存空間

每次啟動新 EC2 執行個體時，都需要建立並支付新的 Amazon Elastic Block Store (Amazon EBS) 磁碟區，以容納作業系統。隨著這種擴展，成本隨之擴展。如果您使用容器，則可以降低儲存成本，因為所有容器共用相同的基礎作業系統。此外，容器會使用層的概念，針對以該映像為基礎的所有執行中容器重複使用不可變部分的容器映像檔。在上述範例案例中，所有容器都在執行 .NET Framework，因此所有容器都共用中繼和不可變的 ASP.NET 架構層。

將 end-of-support 伺服器遷移到容器

Support 視窗服務器 2012 和視窗服務器 2012 R2 在十月 10, 2023 結束。您可以移轉在 Windows Server 2012 或舊版上執行的應用程式，方法是將應用程式容器化為在新的作業系統上執行。如此一來，您就可以避免在不合規的作業系統上執行應用程式，同時充分利用容器所提供的成本效益、降低風險、營運效率、速度和敏捷性。

使用這種方法需要注意的是，如果您的應用程序需要與當前使用的操作系統版本相關的特定 API (例如 COM Interop)。在這種情況下，您必須測試將應用程序移動到較新的 Windows 版本。Windows

容器會將其基本容器映像 (例如 Windows 伺服器 2019) 與容器主機的作業系統 (例如 Windows 伺服器 2019) 對齊。測試和移動到容器可以通過更改 Dockerfile 中的基本映像並部署到運行最新版本 Windows 的一組新主機，從而使 future 的操作系統升級更容易。

移除第三方管理工具和授權

管理伺服器機群需要使用數個協力廠商系統作業工具來進行修補和組態管理。這些可能會使基礎結構管理變得複雜，而且您通常會產生第三方授 如果您使用容器 AWS，則不需要在操作系統端管理任何內容。容器執行階段會管理容器。這意味著底層主機是短暫的，並且可以輕鬆替換。您可以執行容器，而不需要直接管理容器主機。此外，您可以使用免費工具，例如輕 AWS Systems Manager Session Manager 鬆訪問主機並對問題進行故障排除。

改善控制和可攜性

相較於 EC2 執行個體，容器可讓您對伺服器資源 (例如 CPU 和 RAM) 進行更精細的控制。對於 EC2 執行個體，您可以透過選取執行個體系列、執行個體類型和 [CPU 選項來控制 CPU](#) 和 RAM。但是，使用容器時，您可以準確定義要分配給 ECS 任務定義中的容器或 [Amazon EKS 中的網繭](#) 的 CPU 或 RAM 數量。事實上，我們建議您 [指定 Windows 容器的容器層級 CPU 和記憶體](#)。這種級別的粒度帶來了成本效益。請考慮下面的示例代碼：

```
json
{
  "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/demo-
service:1",
  "containerDefinitions": [
    {
      "name": "demo-service",
      "image": "mcr.microsoft.com/dotnet/framework/samples:aspnetapp-
windowsservercore-ltsc2019",
      "cpu": 512,
      "memory": 512,
      "links": [],
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 0,
          "protocol": "tcp"
        }
      ],
    }
  ],
}
```

加速創新

移至容器可讓您更輕鬆地將開發生命週期的各個階段自動化，包括建置、測試和部署應用程式。如果您將這些流程自動化，那麼您可以讓您的開發和營運團隊有更多時間專注於創新。

降低總持有成本

移至容器通常會減少對授權管理和端點保護工具的依賴。由於容器是暫時的運算單位，因此您可以自動化並簡化修補、擴展以及備份和還原等管理工作。這可以減少管理和操作容器型工作負載的總體擁有成本。與虛擬機器相比，容器更有效率，因為它們可讓您將應用程式的放置最大化，以便提高應用程式基礎結構資源的使用率。

縮小技能差距

AWS 提供計劃和沉浸式日，以提高客戶開發團隊對容器和 DevOps 技術的技能。這包括動手諮詢和啟用。

重構為 .NET 5 + 並使用 Linux 容器

雖然您可以透過將 .NET Framework 應用程式移至容器來降低成本，但是當您將舊版 .NET 應用程式重構為雲端原生替代方案時，可進一步節省成本。AWS

移除授權成本

將您的應用程式從 Windows 上的 .NET 架構重構為 Linux 上的 .NET 核心，可節省約 45% 的成本。

存取最新的增強功能

將您的應用程式從 Windows 上的 .NET 架構重構到 Linux 上的 .NET 核心，可讓您存取最新的增強功能，例如重力 2。與同類執行個體相比，Graviton2 提供高出 40% 的效能價格。

改善安全性和效能

將您的應用程式從 Windows 上的 .NET 架構重構為 Linux 容器上的 .NET 核心，可提升安全性和效能。這是因為您取得最新的安全性修補程式、受益於容器隔離，並可存取新功能。

使用 Windows 容器，而不是在一個 IIS 執行個體上執行多個應用程式

請考慮使用 Windows 容器，而不是在具有網際網路資訊服務 (IIS) 的 EC2 Windows 執行個體上執行多個應用程式的下列優點：

- 安全性 — 容器提供開箱即用的安全性層級，無法透過 IIS 層級的隔離來達成。如果一個 IIS 網站或應用程式遭到入侵，則所有其他託管的網站都會暴露且易受攻擊。比起透過網頁弱點取得伺服器控制權，容器逸出是罕見且難以利用的弱點。
- 彈性 — 在程序隔離中執行容器並擁有自己的執行個體的能力，可提供更精細的網路選項。容器也在許多 EC2 執行個體之間提供複雜的分發方法。在單一 IIS 執行個體上合併應用程式時，您無法獲得這些好處。
- 管理額外負荷 — 伺服器名稱指示 (SNI) 會產生需要管理和自動化的額外負荷。此外，您必須努力處理典型的作業系統管理操作，例如修補，故障排除 BSOD (如果未到位 auto 擴展)，端點保護等。根據[安全性最佳做法](#)設定 IIS 網站是一項耗時且持續的活動。您甚至可能需要設定[信任層級](#)，這也會增加管理負荷。容器被設計為無狀態和不可變的。最終，如果您改用 Windows 容器，則部署會更快速、更安全且可重複。

後續步驟

投資現代化基礎架構以執行您的舊式工作負載，可為您的組織帶來巨大的好處。AWS 容器服務可讓您更輕鬆地在內部部署或雲端中管理基礎架構，因此您可以專注於創新和業務需求。雲端中將近 80% 的容器都在 AWS 今天執行。AWS 為幾乎所有使用案例提供了一組豐富的容器服務。若要開始使用，請參閱[容器位於 AWS](#)。

其他資源

- [使用 ECS 容量提供者和 EC2 Spot 執行個體 \(AWS 部落格\) 將容器工作負載的成本最佳化](#)
- [Amazon ECS 和 AWS Fargate \(AWS 部落格\) 的成本優化檢查清單](#)
- [Amazon EKS on AWS 引力 2 正式推出：多架構應用程式的考量事項 \(部落格\) AWS](#)
- [上 AWS 的 Kubernetes 的成本最佳化 \(部落格\) AWS](#)
- [利用卡彭特整合優化您的 Kubernetes 運算成本 \(部落格\) AWS](#)

優化 Amazon ECS 上 AWS Fargate 任務的成本

概觀

正確的大小調整 AWS Fargate 任務是成本優化的重要步驟。通常情況下，應用程式是針對 Fargate 任務構建的任意大小，並且永遠不會重新審視。這可能會導致 Fargate 任務過度佈建和不必要的支出。本節說明如[AWS Compute Optimizer](#)何使用提供可行的建議，以便針對在 Fargate 上執行的 Amazon Elastic Container Service (Amazon ECS) 服務優化任務 CPU 和記憶體。Compute Optimizer 也會量

化採用這些建議的成本影響。這可讓您根據節省商機的大小來排定最佳化工作的優先順序。Compute Optimizer 建議提供容器層級 CPU 和記憶體組態，以縮減工作規模。

成本效益

在 Fargate 上正確調整 Amazon ECS 任務的大小可以將長時間執行任務的成本降低 30—70%。在不檢閱應用程式效能指標以適當調整任務大小的情況下，您可以將 EC2 運算執行個體使用的相同思維方式套用至容器大小。這導致了過大的 Fargate 任務，從而增加了閒置資源的成本。您可以使用 Compute Optimizer，以反應方式顯示正確的大小調整機會。理想情況下，應用程式擁有者會檢閱特定的應用程式效能指標，並移除作業系統額外負荷，以確保指定適當的工作大小。如需詳細資訊，請參閱本指南的 < [將 Windows 應用程式移至容器](#) > 一節。

成本最佳化建議

本節提供有關使用 Compute Optimizer 在 Fargate 任務上調整 Amazon ECS 大小的建議。

作為成本優化過程的一部分，我們建議您執行以下操作：

- 啟用 Compute Optimizer
- 使用 Compute Optimizer
- 將任務標記為正確的大小
- 啟用成本分配標籤以使用 AWS 帳單工具
- 實施正確的大小建議
- 在 Cost Explorer 中檢閱成本前後的成本

啟用 Compute Optimizer

您可以[AWS Compute Optimizer](#)在中的組織或單一帳戶層級啟用 AWS Organizations。整個組織的配置為所有成員帳戶的整個叢集中的新執行個體和現有執行個體提供持續報告。這使得正確的調整大小是一個週期性的活動，而不是一個 point-in-time 活動。

組織層級

對於大多數組織而言，使用計算最佳化工具的最有效方法是在組織層級。這提供了組織的多帳戶和多區域可見性，並將資料集中到單一來源以供檢閱。若要在組織層級啟用此功能，請執行下列動作：

1. 使用具有**必要權限**的角色登入您的[AWS Organizations 管理帳戶](#)，然後選擇選擇加入此組織內的所有帳戶。您的組織必須[啟用所有功能](#)。
2. 啟用管理帳戶後，您可以登入帳戶、查看所有其他成員帳戶，以及瀏覽其建議。

Note

最佳做法是為運算最佳化程式設定[委派的系統管理員帳戶](#)。這可讓您行使最低權限原則，將對 AWS Organizations 管理帳戶的存取減至最少，同時仍提供對整個組織服務的存取權限。

單一帳戶級別

如果您以高成本但無法存取的帳戶為目標 AWS Organizations，您仍然可以為該帳戶和區域啟用 Compute Optimizer。若要瞭解選擇加入程序，請參閱[開始使用 AWS Compute Optimizer](#)。

Note

建議會每天重新整理，最多可能需要 12 小時才能產生。請記住，Compute Optimizer 在過去 14 天內需要 24 小時的指標，才能在 Fargate 上針對 Amazon ECS 產生建議。如需詳細資訊，請參閱[Compute Optimizer 文件中的 Fargate 上 Amazon ECS 服務](#)的要求。

Compute Optimizer 會針對您在 Fargate 上的 Amazon ECS 服務自動分析下列 Amazon CloudWatch 和 Amazon ECS 使用率指標：

- CPUUtilization— 服務中使用的 CPU 容量百分比。
- MemoryUtilization— 服務中使用的記憶體百分比。

使用 Compute Optimizer

考慮一個範例，著重於在單一帳戶和單一區域內進行正確的大小變更。在此範例中，所有帳戶的組織層級都會啟用「Compute Optimizer」。請記住，正確的大小調整是一個破壞性的程序，在大多數情況下，應用程式擁有者會在排定的維護時段內精確地執行數週。

如果您從組織的管理帳戶導覽至 Compute Optimizer (如下列步驟所示)，則可以選擇要調查的帳戶。在此範例中，一個工作正在中過度佈建的單一帳戶中 us-east-1 執行。重點是將 Amazon ECS 服務的大小調整為建議的大小。

1. 開啟 [\[計算最佳化處理程式\]](#)
2. 在「儀表板」頁面上，依據搜尋 = 過度佈建進行篩選，以查看 Fargate 上的所有 Amazon ECS 服務。
3. 若要檢閱 Fargate 上過度佈建 ECS 服務的詳細建議，請向下捲動，然後選擇 [\[檢視建議\]](#)。

4. 選擇導出並保存文件以備 future 使用。

Note

若要儲存建議以供 future 檢閱，您必須在每個區域中擁有可用的 S3 儲存貯體供 Compute Optimizer 寫入。如需詳細資訊，請參閱運算優化器文件 [AWS Compute Optimizer 中的 Amazon S3 儲存貯體政策](#)。

若要查看 Compute Optimizer 的建議，請執行下列動作：

1. 在 [計算最佳化處理程式主控台](#) 中，移至 [匯出建議] 頁面
2. 對於 S3 儲存貯體目的地，選擇您的 S3 儲存貯體。
3. 在 [匯出篩選器] 區段中，對於 [資源類型]，選擇 Fargate 上的 ECS 服務。
4. 在 Fargate 的 ECS 服務建議頁面上，鑽研 Fargate 上的其中一個 ECS 服務，然後查看 Compute Optimizer 的 CPU 和記憶體建議。例如，檢閱 [比較目前設定與建議的工作大小] 和 [比較目前設定與建議的容器大小] 區段中的建議。

要獲取 Fargate 的 ECS 服務列表，您需要正確的尺寸，請執行以下操作：

1. 開啟 [Amazon S3 主控台](#)。
2. 在導覽窗格中，選擇「值區」，然後選擇要匯出結果的值區。
3. 在 [物件] 索引標籤上，選取您的物件並選擇 [下載]。
4. 在您下載的結果中，篩選搜尋結果欄，以便在 Fargate 上僅顯示過度佈建的 Amazon ECS 服務。這會顯示您計劃針對正確大小設定的 Amazon ECS 服務。
5. 將工作定義儲存在文字編輯器中，以供日後使用。

正確調整標籤任務的

標記工作負載是組織中資源的強大工具 AWS。您可以使用標籤來獲得成本的精細可見性，並啟用退款功能。有許多方法和策略可以在 AWS 資源中添加標籤以處理退款和自動化。如需詳細資訊，請參閱 [標記 AWS 資源的 AWS 白皮書最佳做法](#)。下列範例用 [AWS CloudShell](#) 來標記屬於目標帳戶和 AWS 區域任何 Amazon ECS 服務一部分的所有任務。

```
#!/bin/bash
# Set variables
```

```
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$( w secs list-clusters -query 'clusterArns' -output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$( w secs list-services -cluster $ClustersArn -query 'serviceArns' -output
  text)
  for ServiceArn in $ServiceArns; do
    TasksArns=$( w secs list-tasks -cluster $ClustersArn -service-name $ServiceArn -query
    'taskArns' -output text)
    for TasksArn in $TasksArns; do
      w secs tag-resource -resource-arn $TasksArn -tags key=$TAG_KEY,value=$TAG_VALUE
    done
  done
done
```

下列程式碼範例顯示如何啟用[標籤傳播](#)至所有 Amazon ECS 服務。

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$(aws ecs list-clusters --query 'clusterArns' --output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$(aws ecs list-services --cluster $ClustersArn --query 'serviceArns' --
  output text)
  for ServiceArn in $ServiceArns; do
    aws ecs update-service --cluster $ClustersArn --service $ServiceArn --propagate-tags
    SERVICE &>/dev/null
    aws ecs tag-resource --resource-arn $ServiceArn --tags key=$TAG_KEY,value=$TAG_VALUE
  done
done
```

啟用成本分配標籤以使用 AWS 帳單工具

建議您啟動使用者定義的成本配置標籤。如此一來，您就可以在 AWS 計費工具中辨識及篩選「正確調整」標籤 (例如，AWS Cost Explorer 和 AWS Cost and Usage Report)。如果未啟用此功能，則無法使用標籤篩選選項和資料。如需有關使用成本配置標籤的資訊，請參閱 AWS Billing and Cost Management 文件中的[啟動使用者定義的成本配置標籤](#)

等待 24 小時後，您可以在 Cost Explorer 中看到標籤，然後再在下一節中實作正確的大小建議。若要這麼做，請在 [Cost Explorer] 中搜尋 [正確調整大小] 標籤。

實施正確的大小建議

Compute Optimizer 會提供工作或容器大小建議。要實現正確的大小建議，請執行以下操作。

1. 開啟 [Amazon ECS 主控台](#)。
2. 從導覽列中選擇包含您任務定義的區域。
3. 在導覽窗格中，選擇 Task Definitions (任務定義)。
4. 在 Task definitions (任務定義) 頁面上，選擇任務，然後選擇 Create new revision (建立新修訂版)。
5. 在 Create new task definition revision (建立新任務定義修訂版) 頁面上進行變更。若要更新容器大小建議，請更新 [ECS](#) 任務定義中的 container Definitions 區塊 memory 下方 cpu 並加以更新。例如：

```
"containerDefinitions": [  
  {  
    "name": "your-container-name",  
    "image": "your-image",  
    "cpu": 1024,  
    "memory": 2048,  
  }  
],
```

6. 驗證資訊，然後選擇 Create (建立)。

若要更新 Amazon ECS 服務，請執行下列動作：

1. 開啟 [Amazon ECS 主控台](#)。
2. 在 Clusters (叢集) 頁面上，選取您的叢集。
3. 在 Cluster overview (叢集概觀) 頁面中，選取服務，然後選擇 Update (更新)。
4. 針對 Task definition (任務定義)，選擇要使用的任務定義系列和修訂。

對於進階操作員，您可以 CloudShell 使用更新 Amazon ECS 服務。例如：

```
bash  
#!/bin/bash  
# Set variables
```

```
ClustersName="workshop-cluster"  
ServiceName="lab7-fargate-service"  
TaskDefinition="lab7-fargate-demo:3"  
# update the service  
aws ecs update-service --cluster $ClustersName --service $ServiceName --task-definition  
$TaskDefinition
```

檢討成本前後

在您有適當的資源大小之後，您可以使用 Cost Explorer 顯示成本之前和之後的成本，方法是使用「正確調整大小」標籤。回想一下，您可以使用[資源標籤](#)來追蹤成本。通過使用多層標籤，您可以對成本進行精細的可見性。在本指南中涵蓋的範例中，RightSize 標籤用於將一般標籤套用至所有目標實例。然後，團隊標籤用於進一步組織資源。下一步是引入應用程式標籤，以進一步顯示操作特定應用程式的成本影響。

考慮一個可以通過對單一帳戶級別使用「正確大小」標籤來實現的成本降低的示例。在此示例中，運營成本從每天 30.26 美元到每天 7.56 美元。假設每月 744 小時，正確尺寸之前的年度成本是 \$11,044.9。在正確的尺寸之後，年度成本降至 2,759.4 美元。這可轉換為此帳戶的運算成本降低 75%。想像一下，這在一個大型組織中的影響。

在開始您的正確尺寸之旅之前，請考慮以下事項：

- AWS 提供許多降低成本的選項。這包括 [AWS OLA](#)，其中在移至之前 AWS 審查您的內部部署執行個體 AWS。AWS OLA 還為您提供適當調整大小的建議和授權指南。
- 在購買 [Savings Plans](#) 之前，請完成所有正確的尺寸。這可以幫助您避免超額購買您的 Savings Plans 承諾。

後續步驟

我們建議您執行以下步驟：

1. 檢閱您現有的環境，並考慮將 Amazon EBS gp2 磁碟區轉換為 gp3 磁碟區。
2. 查看 [Savings Plans](#)。

其他資源

- [開始使用 Compute Optimizer](#) (AWS 文件)
- [標記 AWS 資源的最佳做法](#) (AWS 白皮書)

- [窗戶上的容器 AWS \(AWS 工作室 \)](#)

掌握您的 Amazon EKS 成本

概觀

要有效監控 Kubernetes 部署的成本，必須有完整檢視。唯一的固定和已知成本是 Amazon Elastic Kubernetes Service (Amazon EKS) 控制平面。這包括構成部署的所有其他元件，從運算和儲存到網路，都是根據您的應用程式需求而定的可變數量。

您可以使用 [Kubecost](#) 來分析 Kubernetes 基礎結構的成本，從[命名空間](#)和[服務](#)到個別[網繭](#)，然後在儀表中顯示資料。Kubecost 表示了運算和儲存等叢集內成本，以及 [Amazon 簡單儲存服務 \(Amazon S3\) 儲存貯體](#)和 [Amazon Relational Database Service 服務 \(Amazon RDS\) 執行個體](#)等 out-of-cluster 成本。Kubecost 會根據此資料提出適當大小的建議，並顯示可能影響系統的重要警示。Kubecost 可與[AWS Cost and Usage Report](#)整合，以顯示[運算節省計劃](#)、[預留執行個體](#)和其他 [discount 計劃](#)所節省的成本。

成本效益

Kubecost 提供可視化您 Amazon EKS 部署成本的報告和儀表板。它可讓您從叢集向下鑽研到每個不同元件，例如控制器、服務、節點、網繭和磁碟區。這可讓您全面檢視在 Amazon EKS 環境中執行的應用程式。藉由啟用此可見性，您可以根據 Kubecost 建議採取行動，或檢視每個應用程式的精細層級的成本。適當調整 Amazon EKS 節點群組的大小，可提供與標準 EC2 執行個體相同的潛在節省成本。如果您可以正確調整容器和節點的大小，則可以從執行容器所需的執行個體大小以及 auto 擴展群組中所需的 EC2 執行個體數量中移除運算膨脹。

成本最佳化建議

若要充分利用 Kubecost，我們建議您執行下列動作：

1. 將 Kubecost 部署到您的環境
2. 取得 Windows 應用程式的精細成本明細
3. 正確大小的叢集節點
4. 正確大小的容器請求
5. 管理使用率低的節點
6. 補救放棄的工作
7. 建議採取行動

8. 更新自我管理節點

將 Kubecost 部署到您的環境

[Amazon EKS Finhack 研討會](#) 教您如何部署設定為在自有帳戶中使用 Kubecost 的 Amazon EKS 環境。AWS 這使您可以獲得該技術的實踐經驗。如果您有興趣在組織中舉辦此研討會，請連絡您的客戶團隊。

若要使用 [Helm](#) 將 Kubecost 部署到您的 Amazon EKS 叢集，請參閱與 [Kubecost 協同合作，以便為 EKS 客戶提供成本監控](#)，張貼在部落格上。AWS 或者，您也可以參考官方的 [Kubecost 文件](#)，以取得有關安裝和設定 Kubecost 的說明。如需 Windows 節點 Support 的相關資訊，請參閱 Kubecost 說明文件中的 [Windows 節點支援](#)。

取得 Windows 應用程式的精細成本明細

雖然您可以使用 [Amazon EC2 Spot 執行個體](#) 大幅節省成本，但您也可以從 Windows 工作負載傾向於可設定狀態的事實中受益。Spot 執行個體的使用取決於應用程式，我們建議您確認這些執行個體是否適用於您的使用案例。

若要取得 Windows 應用程式的精細成本明細，請[登入 Kube cost](#)。在導覽頁面中，選擇 [儲蓄]。

正確大小的叢集節點

在 [Kubecost](#) 中，從導覽列選擇 [節省]，然後選擇 [適當調整叢集節點大小]。

假設 Kubecost 報告叢集在 vCPU 和 RAM 方面都是過度佈建的範例。下表顯示了來自 Kubecost 的詳細信息和建議。

	Current	推薦：簡單	推薦：複雜
總計數	美元 (每月)	美元/月	每月 3 美元
節點計數	4	5	4
CPU	74 個 vCPU	10 個 vCPU	8 個 vCPU
RAM	一百二十二	20 GB	十八 GB
實例細分	2 厘米大尺碼 +2 個更多	中等	2 厘米。大尺寸 +1 多

如 Kubecost 部落格文章所述：[尋找 Kubernetes 叢集的最佳節點組合](#)，簡單選項會使用單一節點群組，而複雜的選項則使用多節點群組方法。了解如何採用按鈕可以執行一鍵式集群調整大小。這需要安裝 [Kubecost 叢集](#) 控制器。

如果您使用的是不是 [eksctl](#) 建立的 [自我管理 Windows 節點](#)，請參閱 [更新現有的自我管理節點群組](#)。這些指示說明如何在 [Auto Scaling 群組](#) 使用的 Amazon EC2 啟動範本中變更執行個體類型。

正確大小的容器請求

在 [Kubecost](#) 中，從導覽列選擇 [節省]，然後前往 [要求適當調整大小的建議] 頁面。此頁面顯示網繭的 [效率](#)、適當大小的建議，以及預估的成本節省。您可以使用「自訂」按鈕，依「叢集」、「節點」、「命名空間\控制器」等進行篩選。

舉例來說，假設 Kubecost 已經計算出某些網繭在 CPU 和 RAM (記憶體) 方面已過度佈建。然後，Kubecost 建議您調整為新的 CPU 和 RAM 值，以達到預估的每月節省成本。若要變更 CPU 和 RAM 值，您必須更新 [部署資訊清單](#) 檔案。

管理使用率低的節點

在 [Kubecost](#) 中，從導覽列選擇 [節省]，然後選擇 [管理未充分利用的節點]。

考慮一個例子，其中頁面顯示叢集中的一個節點在 CPU 和 RAM (記憶體) 方面的使用率不足，因此可以耗盡並終止或調整大小。選擇未通過節點和網繭檢查的節點將提供有關為何無法排空的詳細資訊。

補救放棄的工作

在 [Kubecost](#) 中，從導覽列選擇 [節省]，然後選擇 [捨棄的工作負載] 頁面。在此範例中，您可以依名稱為 windows 的命名空間篩選。此頁面顯示未達到流量臨界值且被視為已放棄的網繭。網繭需要在定義的期間內傳送或接收特定數量的網路流量。

在仔細考量放棄一或多個網繭之後，您可以透過縮減複本數目、刪除部署、調整其大小以消耗較少的資源，或通知應用程式擁有者您認為已放棄部署，以節省成本。

建議採取行動

在適當大小的叢集節點區段中，Kubecost 會分析叢集中工作者節點的使用情況，並提出正確調整節點大小以降低成本的建議。[Amazon EKS 可以使用兩種類型的節點群組：自我管理和受管。](#)

更新自我管理節點

如需更新自我管理節點的相關資訊，請參閱 Amazon EKS 文件中的 [自我管理節點更新](#)。它指出使用創建的節點組 `eksctl` 無法更新，並且必須使用新配置遷移到新節點組。

舉例來說，假設您有一個名為的 Windows 節點群組 `ng-windows-m5-2xlarge` (使用 `m5.2xlarge` EC2 執行個體)，而且您想要將網繭遷移到名為的 [新節點群組 `ng-windows-t3-large`](#) (由 `t3.large` EC2 執行個體支援以節省成本)。

若要在使用由部署的節點群組時移轉至新節點群組 `eksctl`，請執行下列動作：

1. 若要尋找網繭目前所在的節點，請執行 `kubectl describe pod <pod_name> -n <namespace>` 命令。
2. 執行 `kubectl describe node <node_name>` 命令。輸出結果會顯示節點正在 `m5.2xlarge` 執行個體上執行。它也符合節點群組名稱 (`ng-windows-m5-2xlarge`)。
3. 若要將部署變更為使用節點群組 `ng-windows-t3-large`，請刪除節點群組 `ng-windows-m5-2xlarge` 並執行 `kubectl describe svc,deploy,pod -n windows`。現在已刪除其節點群組，部署會立即開始重新部署。

Note

刪除節點群組時，服務將會停機。

4. 幾分鐘後再次執行 `kubectl describe svc,deploy,pod -n windows` 命令。輸出顯示網繭再次處於 [執行中] 狀態。
5. 若要顯示網繭現在正在節點群組上執行 `ng-windows-t3-large`，請再次執行 `kubectl describe pod <pod_name> -n <namespace>` 和 `kubectl describe node <node_name>` 命令。

替代的調整大小

此方法適用於任何自我管理或受控節點群組的組合。[將工作負載從 EKS 自我管理節點群組無縫移轉至 EKS 管理的節點群組](#) 部落格文章提供指引，說明如何將工作負載從具有過大型執行個體類型的節點群組遷移到大小適當的節點群組，而且沒有任何停機時間。

後續步驟

Kubecost 使您可以輕鬆地將 Amazon EKS 環境的成本視覺化。Kubecost 與 Kubernetes 和 AWS API 的深度整合，可協助您找到潛在的成本節約。您可以在 Kubecost 的「儲蓄」儀表板中看到這些建議。Kubecost 也可以透過其 [叢集控制器](#) 功能，為您實作其中一些建議。

我們建議您檢閱 [AWS 和 Kubecost 合作中的 step-by-step 部署](#)，以便透過容器部落格為 EKS 客戶提供 [成本監控](#) 部落格文章。AWS

其他資源

- [Amazon EKS 工作坊](#) (Amazon EKS 工作坊)
- [AWS 和 Kubecost 合作，為 EKS 客戶提供成本監控](#) (部落格)AWS
- [Amazon EKS 芬哈克工作AWS 坊 \(工作室 \)](#)
- [窗戶上的容器 AWS](#) (AWS 工作室)

重新平台視窗應用程式與應用程式容器

概觀

[AWS App2Container](#) 是一種用於將 Java 和 .NET Web 應用程序遷移和現代化為容器的命令行工具。App2Container 會分析並建立在裸機、虛擬機器、Amazon 彈性運算雲端 (Amazon EC2) 執行個體或其他雲端供應商中執行的所有應用程式的清單。您選取要容器化的應用程式。App2Container 將應用程式構件和相依性封裝到容器映像中、設定網路連接埠，並產生必要的亞馬遜彈性容器服務 (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 部署成品，這些成品是基礎設施即程式碼 (IaC) 範本。App2Container 會佈建將容器化應用程式部署至生產環境所需的雲端基礎結構和 CI\ CD 管線。[如需詳細資訊，請參閱 App2Container 文件中的應用程式容器的運作方式。](#)

透過 App2Container，您可以移轉至應用程式 AWS 並將其現代化為容器，同時標準化應用程式的部署和作業。您可以使用 App2Container 協助快速建置概念驗證 (PoC)，或加速容器中的生產工作負載部署。

使用 Windows 應用程式時，有幾件事需要記住。應用程序容器支持 ASP.NET 應用程序的容器化部署在 Microsoft 互聯網信息服務 (IIS)，包括在 Windows 服務器 2016，視窗服務器 2019，或視窗服務器核心 2004 上運行的 IIS 託管的視窗通信基礎 (WCF) 應用程序。如需詳細資訊，請參閱 App2Container 文件中的 [Windows 支援的應用程式](#)。App2Container 會使用 Windows 伺服器核心做為其容器構件的基礎映像，並將 Windows 伺服器核心容器版本與您執行容器化命令的伺服器的作業系統 (OS) 版本相符。這種方法將應用程式與基礎作業系統分離，以便您可以在不執行傳統移轉的情況下升級作業系統。

如果您使用工作者電腦來容器化應用程式，容器基礎映像 (例如 Windows Server 2019 長期服務通道 (LTSC)) 會與您的工作者機器作業系統 (例如 Windows 伺服器 2019) 相符。如果您直接在應用程式伺服器上執行容器化，該版本會與您的應用程式伺服器作業系統相符。如果您的應用程式是在 Windows 伺服器 2008 或 2012 R2 上執行，您仍然可以透過設定工作者電腦進行容器化和部署步驟來使用 App2Container。應用程式容器不支援在 Windows 用戶端作業系統上執行的應用程式，例如視窗

7 或視窗 10。App2Container 支持湯姆貓，東米和 JBoss (獨立模式) 框架的 Java 進程。如需詳細資訊，請參閱 [App2Container 相容性](#)。

成本效益

與伺服器部署設計模式相比，容 one-application-to-one 器化和整合應用程式最多可 [節省 60% 的運算](#) 成本。App2Container 可協助加速應用程式容器化程序。以下是使用 App2Container 來滿足您的現代化需求的一些好處：

- App2 容器提供，不收取額外費用。
- App2Container 支持容器映像中的多個應用程序。
- 使用 App2Container 將舊版 .NET 應用程式移至容器，以解決即將終止支援的作業系統。您可以移至較新的作業系統，避免支付延伸支援費用，並降低安全性風險。
- 容器是封裝 .NET 應用程式的有效率且符合成本效益的方法。檢閱 [MACO 建議-移至容器中容器的優點](#)。
- 應用程式整合和容器化可以更有效率地運用您的運算資源，協助您減少運算、儲存和授權需求。
- 移轉至容器可降低營運負擔和基礎架構成本，並提高開發可攜性和部署靈活性。

成本最佳化建議

如需有關如何使用 App2Container 的指示，請參閱 [開始](#) 使用。AWS App2Container 如需有關 App2Container 命令的詳細資訊，請參閱 [App2 Container 命令參考](#)。

後續步驟

App2Container 可以加速應用程式容器化和部署到 Amazon EKS 或 Amazon ECS 的程序。將應用程式部署到容器可降低運算、網路和儲存成本，並降低應用程式操作員的營運成本。

如需 App2Container 的實際操作體驗，請參閱使用工作坊 [進行現代化](#)。AWS App2Container 如果您想獲得深入的學習體驗，請要求您的 AWS 帳戶團隊設置 App2Container 沉浸式日。

其他資源

- 使用 [容器化複雜的多層 Windows 應用程式](#) (部落格文 AWS App2Container 章) AWS
- [使用 AWS App2Container \(AWS 部落格文章\) 將舊版 ASP.NET 應用程式容器化](#)
- [App2Container 支援的應用程式](#) (AWS 文件)
- 工作坊 [現代化 \(AWS App2Container 工作坊\)](#) AWS 工作室)

- [AWS App2Container 常見問題](#) (AWS 網站)

儲存

為您的 Microsoft 工作負載選擇正確的儲存裝置是一項重要的架構決策。在決策過程中，我們建議您制定儲存計劃，並決定應用程式和服務的功能需求。本章提供下列儲存選項的概觀，這些選項可能會納入您的規劃。

章節：

- [Amazon EBS](#)
- [Amazon FSx](#)
- [AWS Storage Gateway](#)

Amazon EBS

Amazon 彈性區塊存放區 (AmazonEBS) 是一種全受管區塊儲存服務，可讓您存放可與 Amazon 彈性運算雲端 (AmazonEC2) 執行個體搭配使用的永久區塊層級儲存磁碟區。您可以利用 Amazon 中的多項功能，針對 EBS 對雲端中的 Windows 工作負載有效管理和最佳化儲存資源。例如，您可以使用 Amazon EBS 佈建工作負載所需的確切數量 IOPS 和輸送量、從各種磁碟區類型中選取以符合工作負載需求的磁碟區類型，以及使用工具來識別和消除浪費的儲存資源。這種對儲存效能和使用量的精細控制可協助您最佳化儲存資源，同時避免不必要的成本。

本節涵蓋下列主題：

- [將 Amazon EBS 卷從 gp2 遷移到 gp3](#)
- [修改 Amazon EBS 快照](#)
- [刪除未連接的 Amazon 卷 EBS](#)

將 Amazon EBS 卷從 gp2 遷移到 gp3

概觀

固態硬碟 (SSD) 是生產和高效能工作負載的標準儲存選項。Amazon EBS 為中高效能工作負載提供 [一般用途 SSD 磁碟區](#)。許多 AWS 服務（包括 AmazonEC2）的標準是 [gp2](#)，這些通用 SSD 磁碟區的第二代。第三代通用目的 SSDs，稱為 [gp3](#)，於 2020 年 12 月發布。

與上一代相比，gp3 產品對性能自定義方面進行了顯著改進。對於 Amazon EBS gp2 磁碟區而言，效能與磁碟區的大小緊密結合。對於每 1 GB 的容量，gp2 磁碟區就會獲得 3 IOPS 個效能。也就是說，

一個 2,000 GB 的 Gp2 磁碟區能夠容納 6,000 個。IOPS 對於 gp3 磁碟區，可以獨立於儲存容量自訂效能。即使是小容量磁碟區也能達到高達 16,000 IOPS 和 1,000 MB/s 的效能能力。

gp3 磁碟區的另一個主要變更是基準 IOPS 效能。GP3 磁碟區的開始時間為 3,000 IOPS。相較之下，gp2 磁碟區的大小必須達到 1 TiB，才能達到相同的效能能力。對於 Windows 服務器，它通常有一個 C：驅動器小於 1 TiB，從 gp2 升級到 gp3 是一個顯著的性能改進。

最後，與 gp2 卷相比，gp3 卷的價格是最大的改進之一。gp3 磁碟區提供所有增強的效能功能，成本是 gp2 磁碟區的 20%。

成本影響

由於能夠獨立於容量擴充效能，因此瞭解新增額外 IOPS 和輸送量的定價層面非常重要。對於 gp2 磁碟區，定價以佈建容量為每月 0.10 USD。對於 gp3 磁碟區，定價類似於高性能 [佈建 IOPSSSD 磁碟區](#)，其容量成本只有一個，而額外 IOPS 和輸送量則需另外支付費用。

如下表所述，gp3 磁碟區的容量價格為每月每 GB 0.08 美元 (比 GP2 便宜 20%)，而個別成本則為每個佈建月 0.005 美元，超過 3,000 美元，而輸送量則超過 125 美元，每個佈建 IOPS 月 0.04 USD。IOPS MiBs MiBs

	gp3	gp2
磁碟區大小	1 GiB – 16 TiB	1 GiB – 16 TiB
基準線 IOPS	3,000	3 IOPS /GiB (最少 100 IOPS) 至最多 16,000 IOPS
		小於 1 TiB 的磁碟區最高可爆炸至 3,000 IOPS
最 IOPS 大/音量	16,000	16,000
基準輸送量	125 MiBs	輸送量限制介於 128 MiBs — 250 之間 MiBs，視磁碟區大小而定
最大輸送量/磁碟區	1,000 MiBs	250 MiBs
價格	每月 0.08 美元	0.10 美元/GiB-月

	gp3	gp2
	每月供應 3,000 元以上的 IOPS 免費費用及 3,000 IOPS 美元 MiBs 免費提供 125 美元，且 每月佈建 MiBs 超過 125 美元 MiBs	

Important

雖然 gp3 磁碟區在容量和效能方面有不同的成本，但如果 gp3 磁碟區設定為相同的效能等級，gp3 磁碟區總是比 gp2 磁碟區便宜。

下表顯示透過各種容量和效能組態將 gp2 轉換為 gp3 磁碟區，可節省成本的範例。

gp2 組態的範例

磁碟區大小 (GiB)	馬克斯 IOPS	吞吐量 (MiBs)	費用 (USD/月)
30	3000	128	3.00
100	3000	128	10.00 美元
500	3000	250	\$50.00
1000	3000	250	\$100.00
2000	6000	250	\$200.00
6000	16000	250	\$600.00

gp3 (基準線) 組態的範例

馬克斯 IOPS	吞吐量 (MiBs)	費用 (USD/月)	降低成本 (與 GP2 相比)
3000	125	\$2.40	20%
3000	125	8.00 美元	20%
3000	125	\$40.00	20%
3000	125	80.00	20%
3000	125	160.00	20%
3000	125	\$480.00	20%

gp3 (gp2 相符) 組態的範例

馬克斯 IOPS	吞吐量 (MiBs)	費用 (USD/月)	降低成本 (與 GP2 相比)
3000	128	2.52 美元	16%
3000	128	8.12 美元	19%
3000	250	\$45.00	10%
3000	250	85.00 美元	15%
6000	250	\$180.00	10%
16000	250	\$550.00	8%

有關成本分析，請參閱 [Amazon 資源中的 EBSgp2 到 gp3 遷移成本節省計算器部分](#)。EBS 您可以下載計算器並使用它來查找通過將 gp2 卷遷移到 gp3 可以節省多少。

成本最佳化建議

如需如何完成遷移程序的指示，請參閱 [將 Amazon EBS 磁碟區從 gp2 遷移到 gp3，並在儲存部落格上節省高達 20% 的 AWS 成本](#)。

其他資源

- [將您的 Amazon EBS 卷從 gp2 遷移到 gp3，並節省高達 20% 的成本 \(AWS 存儲博客 \)](#)
- [建立 AWS Config 自訂規則以優化 Amazon EBS 磁碟區類型 \(AWS 雲端操作和移轉部落格\)](#)
- [透過刪除未使用的 AWS Amazon EBS 磁碟區來控制您的成本 \(AWS 雲端操作和遷移部落格\)](#)
- [Amazon EBS 遷移實用程序 \(GitHub \)](#)
- [從 2020 年尋找節省金錢：發明公告 \(AWS 雲財務管理 \)](#)
- [成本最佳化研討會 \(AWS Well-Architected 的實驗室\)](#)
- [gp2 到 gp3 遷移成本節省計算器 \(下載 \)](#)

修改 Amazon EBS 快照

概觀

刪除 EBS 磁碟區以及管理快照的保留與封存是控制成本的重要方面。您可以透過拍 point-in-time 攝快照，將 EBS 磁碟區上的資料備份到 Amazon 簡單儲存服務 (Amazon S3)。快照是增量備份，因此它們只會儲存在最近快照之後變更的裝置上的區塊。如此無須複製所有資料，可大幅減少建立快照所需的時間，並節省儲存成本。每個快照都包含將資料 (從建立快照開始) 還原到新 EBS 磁碟區所需的所有資訊。

EBS 快照的費用是以十億位元月計算。您需支付快照的大小以及快照保留多久的費用。定價依儲存層而有所不同。對於標準層，只會針對儲存的變更區塊向您收費。對於封存層，您需支付所有儲存的快照區塊的費用。您也需要支付從封存層擷取快照的費用。以下是每個儲存層的範例案例：

- 標準層 — 您有一個儲存 100 GB 資料的磁碟區。您需支付第一個快照 (快照 A) 的完整 100 GB 資料費用。在下一個快照 (快照 B) 時，您有 105 GB 的資料。然後，您只需支付增量快照 B 的額外 5 GB 儲存空間費用。
- 封存層 — 您封存快照 B。然後，快照會移至封存層，而您需支付 105 GB 完整快照區塊的費用。

您可以使用 [Amazon Data Lifecycle Manager](#) 協助您設定生命週期，安排保留和管理快照。

成本影響

EBS 磁碟區和快照的費用會分開管理。EBS 快照的費率會比使用中 EBS 磁碟區低。當執行個體終止時，每個連結 EBS 磁碟區的 [DeleteOnTermination 屬性](#) 值會決定要保留還是刪除磁碟區。依預設，根磁碟區的 DeleteOnTermination 屬性設定 True 為 False。對於所有其他磁碟區類型，它會設定為 True。這會產生操作員打算刪除 EC2 執行個體的情況，但除了根磁碟區之外，還會留下新增至執行個體的磁碟區。如

需檢查不再需要的磁碟區 (及其關聯快照) 的指示，請參閱 Amazon 說明 EBS 文件中的檢視 [Amazon 磁碟區的相關資訊](#)。

根據預設，當您建立快照時，快照會存放在 Amazon EBS 快照標準層 (標準層) 中。存放在標準層中的快照是增量快照。這表示僅儲存最近快照後，磁碟區上已變更的區塊。[Amazon EBS Snapshot Archive](#) 是一種新的儲存層，可用於低成本、長期儲存罕見存取的快照，不需要頻繁或快速擷取。標準與存檔之間的定價差異非常重要，在設定快照策略時，應該是重要的考量因素。對於計劃存放 90 天或更長時間且幾乎不需要存取的快照，Amazon EBS 快照存檔可降低高達 75% 的快照儲存成本。

Amazon EBS 快照儲存	費用
標準	每月 0.05 美元
存檔	每月 5 億美元

在較小的環境中，節省成本可能並不顯著。在大規模的情況下，即使 EBS 磁碟區已被刪除，有多個帳戶和數千個 EC2 執行個體 TBs 的 EBS 快照，節省的成本更為明顯。

下表比較每月僅 50 TB 的標準層和封存層。即使在這個較低的規模下，每年仍然節省了數千美元。

Amazon EBS 快照儲存	每月費用	每年成本
標準型 50 TB	\$312.50	\$3,750
檔案 50 TB	78.13 美元	937.60 美元
	每年儲蓄	\$2,812.40

成本最佳化建議

刪除快照可能不會降低您組織的資料儲存成本。其他快照可能會參考該快照的資料，被參考資料一律予以保留。例如，當您建立包含 10 GiB 資料之磁碟區的第一個快照，其快照大小也是 10 GiB。由於快照是增量式的，建立相同磁碟區的第二個快照時，僅會包含自從建立第一個快照後所有變動的資料區塊。第二個快照也會參考第一個快照中的資料。如果您變更 4 GiB 的資料並拍攝第二個快照，則第二個快照的大小為 4 GiB。此外，第二個快照會參考第一個快照中未變更的 6 GiB。如需詳細資訊，請參閱 [為什麼刪除磁碟區的快照並刪除 EBS 磁碟區本身後，我的儲存成本沒有降低？](#) 在 AWS 知識中心。

考慮下列各項：

- 其他人 AWS 帳戶 擁有並與您的帳戶共用的快照不會向您收取費用。只有當您將共用快照複製到帳戶時，才會向您收費。您也需支付從共用快照建立的EBS磁碟區的費用。
- 如果快照 (快照 A) 被另一個快照 (快照 B) 參考，則刪除快照 B 可能不會降低儲存成本。刪除快照時，只會移除該快照專屬的資料。其他快照所參照的資料會保留下來，而且我們會向您收取這些參考資料的費用。若要刪除增量快照，請參閱 Amazon EBS 文件中的[增量快照刪除](#)。

快照整潔是在中執行工作負載時的標準作業實務 AWS。隨著時間的推移，快照可能會為您不需要的資料增加昂貴的費用。

其他資源

- [透過刪除未使用的 AWS Amazon EBS 磁碟區來控制您的成本](#) (AWS 雲端操作和遷移部落格)
- [刪除 Amazon EBS 快照](#) (Amazon EBS 文檔)
- [成本最佳化研討會](#) (AWS Well-Architected 的實驗室)
- [使用 Amazon 資料生命週期管理員自動存檔 Amazon EBS 快照](#) (AWS 儲存部落格)

刪除未連接的 Amazon 卷 EBS

概觀

未連接 (孤立) EBS 磁碟區可能會導致您的環境中不必要的儲存成本。AWS 必須將定期檢閱和刪除未使用和未使用的EBS磁碟區納入為 AWS 環境衛生的一部分。最佳做法是有適當的程序來持續檢閱EBS磁碟區的使用情況。您可以使用[AWS Compute Optimizer](#)來檢閱未充分利用的執行個體。本節可協助您識別、管理及刪除未附加或未充分利用的EBS磁碟區。

Amazon EBS

[Amazon 彈性區塊存放區 \(AmazonEBS\)](#) 是一種區塊級裝置，可為 Amazon 彈性運算雲端 (AmazonEC2) 執行個體提供儲存磁碟區。EBS提供永久性儲存，可靈活地附加和從EC2執行個體中分離。這表示即使EC2執行個體終止，EBS磁碟區的生命週期仍會持續存在。[DeleteOnTermination](#)屬性是一項功能，可控制執行個體終止時要保留還是刪除連接的EBS磁碟區。根據預設，根磁碟區的屬性設定True為，導致刪除。它會False針對其他磁碟區設定為，進而導致保留。

成本影響

未連接的EBS磁碟區 (也稱為未使用或孤立磁碟區) 會根據佈建的儲存區大小和儲存區類型，產生與連結磁碟區相同的費用。雖然 Amazon EBS 收費的平均成本可能看起來很低，每 GB 每月 0.10 美元，但至關重要的是要認識到，未使用的EBS卷積累可能會隨著時間的推移導致顯著成本。

例如，請考慮保留 50 個未使用的磁碟區 (每個EBS磁碟區都佈建 100 GB 的儲存區大小) 所產生的後果，如下表所示。

儲存磁碟區數量	磁碟區類型	大小	每月總成本
50 個磁碟區	千分之二 (0.10 美元) USD	100 GB	100 GB 的EBS磁碟區數個月 0.10 美元 = USD 50 美元 USD

上表中的案例產生了每月約 500 美元或每年 6,000 美元的成本降低。這是邁向降低成本的有效步驟。請務必將刪除未連接的EBS磁碟區整合為 AWS 環境衛生的常規做法。

成本最佳化建議

您可以使 AWS 用輕鬆自動刪除未連接的EBS磁碟區。例如，您可以使用 AWS Lambda、AWS Config、Amazon 和 AWS Systems Manager 根據年齡 CloudWatch、標籤和其他規格定義刪除未連接磁碟區的條件。您還可以使用它們 AWS 服務 來大規模自動執行清理過程。

為了避免意外後果，我們建議您在刪除未連接EBS的磁碟區之前執行盡職調查。

管理未連接的磁碟EBS區

我們建議您考慮以下最佳做法：

- 符合法規要求 — 確認刪除未連接的EBS磁碟區是否符合您組織的治理和合規要求。
- 設定資料備份和保留政策 — 刪除未連接的EBS磁碟區之前，請先將任何重要資料備份到另一個儲存存庫 (例如 [Amazon S3](#))。對於資料保留，[Amazon EBS 快照](#)是保留資料比EBS磁碟區更具成本效益的方式，並且可以在 future 需要時還原磁碟區。如需有效管理快照的詳細資訊，請參閱本指南的[修改 Amazon EBS 快照](#)一節。
- 檢查相依性 — 檢查未連接EBS磁碟區與其他 AWS 資源之間是否有相依性。您可以使用[AWS Management Console](#) 或[API](#)來收集有關EBS磁碟區的描述性資訊，例如大小、狀態和關聯的資源。這是防止刪除任何暫時未連接的資源的重要步驟。
- 建立保留原則 — 為未連接的EBS磁碟區建立保留期。這可協助您識別適當的時間來刪除未連接的磁碟區，確保您的 AWS 環境保持最佳化狀態。例如，您可以建立 [Amazon EventBridge](#) 規則，按排程啟動 Lambda 函數。Lambda 函數可以使用主動識別任何未連接的EBS磁碟區、套用標記機制以便輕鬆追蹤，並在未連接的磁EBS碟區達到或超過定義的閾值時傳送通知。AWS SDK

- 標記未連接的EBS磁碟區 — [標記EBS磁碟區](#)是一種實用的做法，可協助您根據環境、應用程式或擁有者等屬性來組織和識別磁碟區。這在決定要刪除哪些未連接的磁碟區時特別有用，因為它可讓您根據磁碟區的標籤快速識別不再需要的磁碟區。
- 確保安全刪除 — 檢視上次連接EBS磁碟區的時間可協助您判斷刪除磁碟區是否安全。如需詳細資訊，請參閱[如何使用 AWS CLI 命令列出特定 Amazon EBS 磁碟區的附件或分離歷史記錄？](#) 在 AWS 知識中心。
- 身分未充分利用的EBS磁碟區 — 強烈建議您識別和移除未充分利用的EBS磁碟區，以降低儲存成本並維持最佳化 AWS 的環境。AWS Trusted Advisor 並[AWS Compute Optimizer](#)可協助您識別未充分利用的EBS磁碟區，並提供降低成本並提高效率的建議。例如，請參閱使用 AWS Trusted Advisor(GitHub) [設定自動化EBS磁碟區](#)以最佳化磁碟區、[建立 Trusted Advisor 組織 \(TAO\) 儀表板](#) (AWS 工作坊 Studio) 和[使用的成本最佳化 Amazon EBS 磁碟區 AWS Compute Optimizer](#) (AWS 儲存部落格)。

自動清理未連接的磁碟EBS區

我們建議您考慮下列工具，以協助您自動清除未連接的EBS磁碟區：

- [AWS APIs\(DescribeVolumes\)](#) — 您可以使用AWSSDKs或 () 來篩選和尋找未連接的EBS AWS Command Line Interface 磁碟區。AWS CLI您可以使用指令碼或安排程執行的 [Lambda 函數](#)將此程序自動化，以節省時間和精力。來自的[範例指令碼](#)會 GitHub 示範其運作方式。指令碼使用 Lambda 分析 AWS CloudTrail 記錄並識別未連接的EBS磁碟區。
- [AWS Systems Manager 自動化](#) — 這可讓您自動化基礎架構中的例行維護和補救工作。要開始使用，請[創建一個自動化 runbook](#)，它定義了一系列以特定順序執行的步驟。例如，您可以建立 runbook，先建立未連接EBS磁碟區的快照，然後刪除磁碟區本身。這可以幫助您自動化任務，否則如果手動完成，這些任務將耗時且容易出錯。
- [AWS Config](#)— 這可讓您評估、稽核和追蹤 AWS 資源隨時間變更。透過擷取組態變更，您可以用 AWS Config 來評估環境中的合規性、控管和資源使用率。例如，AWS Config 可以識別[未使用的 EBS磁碟區](#)。此外，您可以將 AWS Systems Manager Automation 與關聯，AWS Config 以自動修復未使用EBS磁碟區的刪除。

其他資源

- 使用[AWS Config 和 \(AWS 規範性指南EBS\) 刪除未使用的 Amazon 彈性區塊存放區 AWS Systems Manager\(Amazon\) 磁碟區](#)
- [透過刪除未使用的 AWS Amazon EBS 磁碟區來控制您的成本](#) (AWS 雲端操作和遷移部落格)
- [AWSConfigRemediation-DeleteUnused EBSVolume](#) (AWS Systems Manager 自動化手冊參考)

Amazon FSx

Amazon Windows 檔案伺服器是針對 FSx 對 Windows 工作負載最佳化的全受管檔案儲存服務。它為您提供簡單且可擴充的解決方案，可執行 Windows 型應用程式和工作負載，而不需要複雜的儲存基礎結構管理。您可以使 FSx 用 Windows 檔案伺服器輕鬆佈建和存取共用檔案儲存空間，以本機方式支援您的 Windows 應用程式，包括 Microsoft SQL 伺服器、SharePoint、Microsoft 和自訂 .NET 應用程式。此外，FSx Windows File Server 可提供彈性的定價選項 (例如 pay-as-you-go 和儲存配額)，以及自動重複資料刪除功能，以減少儲存空間並最佳化效能和成本，協助您管理成本。

本節涵蓋下列主題：

- [選擇正確的 SMB 檔案儲存空間](#)
- [在 Amazon 中啟用重複資料刪除功能 FSx](#)
- [瞭解 FSx 適用於 Windows 檔案伺服器的資料分片](#)
- [了解 Amazon 中的 HDD 大量使用 FSx](#)
- [使用單一可用區域](#)

選擇正確的 SMB 檔案儲存空間

概觀

AWS 提供各種完全受控的儲存服務，讓您擁有領先業界的檔案服務的豐富功能，同時結合最新的 AWS 基礎架構創新與安全性。您可以將 AWS 服務納入基礎架構即程式碼 (IaC) 工作流程中，並將其與 AWS 運算、監控和資料保護服務整合。對於 Windows 工作負載，您可以從兩種完全受管的檔案服務中進行選擇，以符合您的應用程式需求：FSx 適用於 Windows 檔案伺服器和 Amazon FSx for NetApp ONTAP。

FSx 視窗檔案伺服器

Amazon FSx Windows 檔案伺服器提供在 Windows 伺服器上建置的全受管共用儲存體，並提供廣泛的資料存取、資料管理和功能。FSx 對於 Windows 文件伺服器很容易與 Windows 環境集成，因為它是一個 Windows 本機服務。我們建議將 Windows 檔案伺服器用 FSx 於使用者和群組共用、伺服 SQL 器、Windows 應用程式和虛擬桌面基礎結構的永遠在容錯移轉叢集執行個體 (VDI)。FSx Windows 文件伺服器還可以與 Amazon FSx 文件網關，Amazon Kendra，Amazon S3 的審計日誌和 Amazon 數據 Firehose 很好地集成。

適用於 ONTAP 的 FSx

FSxfor 基ONTAP於 NetApp的專有ONTAP文件系統。這需要一定程度的提升，而且主要建議現有的內部部署 NetApp 使用者使用。一般使用案例包括使用者和群組共用、SQL伺服器的永遠在容錯移轉叢集執行個體，以及 Windows 應用程式。FSxONTAP支援多種通訊協定、大於 64 TB 的檔案系統 (沒有 DFS命名空間伺服器的 PB 規模)、複製、複製、快照、壓縮 (儲存效率)，以及資料的智慧型分層。

成本影響

FSx視窗檔案伺服器

FSxWindows 檔案伺服器是針對伺服器部署容錯移轉叢集執行個體 AWS 的第一個共用儲存解決方案。SQL使FSx用 Windows 檔案伺服器時，您可以使用SQL標準版授權啟動容錯移轉叢集執行個體。但是，這會防止您依賴「一律開啟」可用性群組，而這些群組需要SQL伺服器企業版授權。從SQL伺服器企業標準版切換到SQL伺服器標準版，您可以節省 65—75% 的[SQL伺服器](#)授權。

您可以使用FSx於容錯移轉叢集執行個體的 Windows 檔案伺服器，從一般儲存體卸載儲EBS存區 I/O。透過將 Windows 檔案伺服器FSx的 I/O 卸載到，您可以縮減EC2執行個體，這些執行個體仰賴高 Amazon EBS 輸送量IOPS，而且不會影響儲存輸送量。

適用於 ONTAP 的 FSx

您可以使用 FSx for ONTAP 在區塊通訊協定 i 上執行 Microsoft 容錯移轉叢集，SCSI並受益於SQL伺服器即時檔案初始化、使用的跨區域複寫 SnapMirror、防毒支援和複製。如果您建立多個資料庫副本進行測試，複製可能會在空間耗用和建立這些資料庫副本的速度上產生重大差異。此外，您可以使用 for NetApp SnapCenter 來管理SQL伺服器EC2執行個體的備份、還原和複製功能ONTAP。FSxFSxfor ONTAP 還提供從低成本容量儲存池儲存的自動分層，SSD以實現效能和成本效益的混合體。

FSx支ONTAP援 NetApp的檔案系統 (ONTAP)，與FSx支援 Windows 原生檔案系統的 Windows NTFS 檔案伺服器不同。的最小大小FSxONTAP為 1024 GB，而FSx對於 Windows 文件伺服器可以啟動低至 32 GB。

與 Microsoft 分佈式文件系統集成

FSx適用於 Windows 檔案伺服器，並FSxONTAP與微軟的[分散式檔案系統 \(DFS\)](#) 整合，以便順暢整合至現有部署。規劃您的架構時，請記住下列事項：

- FSx適用於 Windows 檔案伺服器，以及FSx兩種部署類型 [DFSN\(多個可用區域和單一可用區域\) 上的DFS命名空間 \(\)](#) 的ONTAP支援。
- 僅FSx適用於 Windows 檔案伺服器支援[DFS複寫 \(DFSR\)](#)，且僅在使用單一可用區域時。

成本最佳化建議

Windows 檔案伺服器 and FSx for ONTAP 的效能與設定相關，其價格也是如此。FSx for Windows 檔案伺服器的定價主要取決於儲存容量和儲存體類型、輸送量容量、備份和傳輸的資料。使用 FSx for ONTAP，您需要支付 SSD 儲存 SSD IOPS、容量集區使用量、輸送量容量和備份的費用。

檔案服務	5 TB 儲存空間的成本	組態	區域
FSx 視窗檔案伺服器	\$982.78	單一可用區域 SSD (一萬五千 IOPS) 32 MBps 5 TB 備份 (不節省重複資料刪除功能)	美國東部 (維吉尼亞北部)
適用於 ONTAP 的 FSx	\$979.28	單一可用區域 100% SSD 15,000 個讀寫容量層級 15,000 SSD IOPS 128 MBps 5 TB 備份 (不節省重複資料刪除功能)	美國東部 (維吉尼亞北部)

請謹記以下幾點：

- 重複資料刪除與壓縮可讓您藉由縮減資料大小，在實體裝置上儲存更多資料，但您需要支付佈建的固態硬碟 (SSD) 或硬碟 (HDD) 儲存體的費用。
- 您可以使用 FSx for ONTAP 來分層您的資料。100% 的數據定期訪問並且需要 SSD 存儲空間非常罕見。您可以將不常存取和不常存取的資料移至容量層，以節省成本。
- 此處提到的價格是以層級的 100% 資料和 SSD 層級 15,000 IOPS 的資料來 SSD 計算。

備份

根據預設，FSx適用於 Windows 檔案伺服器ONTAP和 FSx Windows 檔案伺服器的完全受管備份都會儲存在 Amazon S3 上。但是，FSx對於ONTAP還有一個額外的備份選項 SnapVault，可以將備份配置為駐留在容量層中。備份 SnapVault 是一種自我管理的機制，比預設的全受管備份選項更具成本效益。完全受管理的備份選項為每 GB 每月 0.05 USD。上的 SnapVault 備份 ONTAP (10:1 SSD 到容量儲存池) FSx 為 0.03221 美元 (0.05 美元)。

請謹記以下幾點：

- AWS 受管備份提供一小時的精細度。[SnapVault](#)使您可以低至五分鐘。
- 您可以使用 NetApp的工具 (例如CLI和API) 來設定 SnapVault 關聯性和快照複寫。
- 啟用 SnapVault 磁碟區上的all分層原則，以使用容量層做為備份資料的儲存體。
- SnapVault 目的地可以位於相同 AWS 區域、跨區域或內部部署。這通常是單一可用區域或多個可用區域檔案系統備份目的地。相較之下，AWS Backup 由 Amazon S3 的區域彈性提供支援。

正確的尺寸

您也可以透過適當的調整大小並防止過度佈建，節省成本並充分利用您的檔案系統。

要正確大小，請執行以下操作：

1. 根據資料識別您目前的需求。對於一般 Windows 工作負載，您可以使用內建的作業系統工具，例如[效能監視器](#)。
2. 在效能監視器中，使用下列計數器來評估您目前的效能需求。擷取間隔設定為一秒，記錄檔大小上限為 1,000 MB，並啟用覆寫。

```
Logman.exe create counter PerfLog-Short -o "c:\perflogs\PerfLog-Long.blg" -f bincirc -v mmdhmm -max 1024 -c "\LogicalDisk(*)\*" "\Memory\*" "\.NET CLR Memory(*)\*" "\Cache\*" "\Network Interface(*)\*" "\Paging File(*)\*" "\PhysicalDisk(*)\*" "\Processor(*)\*" "\Processor Information(*)\*" "\Process(*)\*" "\Thread(*)\*" "\Redirector\*" "\Server\*" "\System\*" "\Server Work Queues(*)\*" "\Terminal Services\*" -si 00:00:01
```

3. 若要啟動記錄擷取，請執行logman start PerfLog-Short命令。若要停止記錄擷取，請執行logman stop PerfLog-Short命令。

Note

您可以在 c:\perflogs 中找到執行擷取的伺服器上的效能記錄檔。如需詳細資訊，請參閱 Microsoft 文件中的 [Windows 效能監視器概觀](#)。

4. 確定正確的組態後，請使用 Microsoft 等磁碟 stress 工具在 Amazon FSx 檔案系統上測試估計值是否正確 [DISKSPD](#)。
5. 如果您對效能感到滿意，請切換至檔案共用。

我們建議採用保守的儲存容量方法，因為它只能擴充。輸送量容量可以根據需要擴展和縮減。

其他資源

- [Amazon FSx 的 NetApp ONTAP FAQs](#) (AWS 網站)
- [使用新指標最佳化 FSx 適用於 Windows 檔案伺服器的 Amazon 效能](#) (AWS 儲存部落格)

在 Amazon 中啟用重複資料刪除功能 FSx

概觀

重複資料刪除功能可讓您以更有效率的方式儲存資料，同時降低容量需求。它涉及在不影響其保真性或完整性的情況下查找和刪除數據中的重複。重複資料刪除使用子檔案變數大小的區塊和壓縮，對一般檔案伺服器提供 2:1 的最佳化比率，虛擬化資料最高可達 20:1。重複資料刪除比 NTFS 壓縮更有效。重複資料刪除架構的固有之處在於硬體故障期間的彈性，具有對資料和中繼資料的完整總和檢查碼驗證，包括中繼資料的備援和最多存取的資料區塊。

FSx Windows 文件伺服器完全支持重複數據刪除。使用它可以平均節省 50% — 60% 的一般用途檔案共用。在共用範圍內，使用者文件可節省 30% 至 50%，軟體開發資料集最多可節省 70% 至 80%。請務必瞭解，使用重複資料刪除功能可節省的儲存空間，取決於資料集的性質，包括檔案之間存在多少重複資料。如果存儲的數據本質上是動態的，則重複數據刪除不是一個好選擇。

成本影響

為了因應企業中的資料儲存成長，管理員整合伺服器並制定容量擴充和資料最佳化的關鍵目標。重複資料刪除的預設設定可立即節省成本，或者管理員也可以微調設定以查看其他優勢。例如，您可以將重複資料刪除設定為僅在特定檔案類型上執行，也可以建立自訂作業排程。

在高層級，重複資料刪除有三種類型的工作：最佳化、記憶體回收和清除。請注意，在最佳化之後執行記憶體回收工作之前，空間不會釋放。您可以排程工作，也可以手動執行工作。當您手動啟動工作時，也可以使用排程重複資料刪除工作時所有可用的設定 (排程特定的設定除外)。

即使只能有效節省 25% 的重複資料刪除功能，Windows 檔案伺服器也能大幅節省成本。FSx 這些預計節省的成本是根據 AWS Pricing Calculator。

成本最佳化建議

依預設，不會啟 FSx 用 Windows 檔案伺服器檔案系統的重複資料刪除功能。若要使用 [遠端管理](#) 啟用重複資料刪除功能 PowerShell，您必須執行 `Enable-FSxDedup` 指令，然後使用 `Set-FSxDedupConfiguration` 指令來設定組態。如需詳細資訊，請參閱「Windows [檔案伺服器](#)」說明文件中 FSx 的「[管理檔案系統](#)」。

若要啟用重複資料刪除功能，請執行下列命令：

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -  
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

若要驗證您的重複資料刪除組態，請執行下列命令：

```
Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -ConfigurationName  
FSxRemoteAdmin -ScriptBlock {  
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days  
Mon,Tues,Wed,Sat -Start 09:00 -DurationHours 7  
}
```

藉由執行 PowerShell `Measure-DedupFileMetadata` 指令程式，您可以判斷刪除資料夾群組、單一資料夾或單一檔案時，磁碟區上可回收多少磁碟空間，然後執行資源回收工作。具體來說，該 `DedupDistinctSize` 值告訴您刪除這些文件時，您可以獲得多少空間。檔案通常具有在其他資料夾之間共用的區塊，因此重複資料刪除引擎會計算哪些區塊是唯一的，而且會在記憶體回收工作之後刪除。

預設的 [重複資料刪除工作排程](#) 旨在適用於建議的工作負載，並儘可能不侵入 (不包括針對備份使用類型啟用的優先順序最佳化工作)。如果工作負載的資源需求很大，建議您排定只在閒置時間執行的工作，或是減少或增加重複資料刪除工作所允許使用的系統資源量。

根據預設，重複資料刪除會使用 25% 的可用記憶體。但是，這可以通過使用來增加 `-memory switch`。對於最佳化工作，我們建議您設定 15 到 50 的範圍。對於排程的工作，您可以使用較高的記

記憶體消耗量。例如，使用記憶體回收和清理工作 (通常排程在下班時間內執行)，您可以設定較高的記憶體消耗量 (例如 50)。

如需有關重複資料刪除設定的其他資訊，請參閱 [Windows 檔案伺服器說明文件中的使用重複資料刪除降低儲存成本](#)。FSx

其他資源

- [了解重複資料刪除](#) (Microsoft 文件)
- [利用重複資料刪除功能降低儲存成本](#) (FSx適用於 Windows 檔案伺服器文件)

瞭解FSx適用於 Windows 檔案伺服器的資料分片

概觀

FSx對於 Windows 文件服務器的性能取決於配置。它主要以儲存類型、儲存容量和輸送量組態為基礎。您選取的輸送量容量會決定檔案伺服器可用的效能資源，包括網路 I/O 限制、CPU和記憶體，以及檔案伺服器強加的磁碟 I/O 限制。您選取的儲存容量和儲存區類型會決定儲存磁碟區可用的效能資源，也就是儲存磁碟所施加的磁碟 I/O 限制。除了效能之外，組態選擇也會影響成本。FSxWindows 檔案伺服器的定價主要取決於儲存容量和儲存體類型、輸送量容量、備份和傳輸的資料。

如果您有相對較大的檔案儲存空間和效能需求，您可以從資料分割中獲益。數據分片涉及將[文件數據分割為較小的數據集](#) (碎片)，並將其存儲在不同的文件系統中。從多個執 parallel 個體存取資料的應用程式可以同時讀取和寫入這些碎片，以達到高等級的效能。同時，您仍然可以在通用命名空間下向應用程式呈現統一檢視。此外，它還可以幫助擴展檔案資料儲存空間，超越每個檔案系統支援的大型檔案資料集 (64 TB)，最高可達數百 PB。

成本影響

對於大型資料集而言，為 Windows 檔案伺服器檔案系統部署多個小型資料集通常比較有效，而不是一個大型SSD共FSx用以達到相同等級的效能。使用 Windows 檔案伺服器HDD和SSD儲存類型的組合，可以更好地節省成本，並可讓您將工作負載與最佳的基礎磁碟子系統配FSx對。在下表中，您可以看到單一 17 TB 檔案系統之間的差異，並將其與增加相同容量的多個較小檔案系統進行比較。

具有多重工作負載的大型SSD檔案

伺服器名稱	費用	組態	區域
Amazon FSx 文件服務器	\$5,716 USD	十七 TB SSD 30% 重複資料刪除 每秒 256 兆比特 17 TB 資料備份	美國東部 (維吉尼亞北部)

分割工作負載使 DFSN

伺服器名稱	費用	組態	區域	Share (分享)
Amazon FSx 文件服務器	\$1,024 USD	二 TB SSD 20% 重複資料刪除 每秒 128 兆比特 2 TB 備份 Multi-AZ	美國東部 (維吉尼亞北部)	分享到 1
Amazon FSx 文件服務器	港幣 2,132 元 USD	五 TB SSD 30% 重複資料刪除 每秒 256 兆比特 5 TB 備份 Multi-AZ	美國東部 (維吉尼亞北部)	分享到 2
Amazon FSx 文件服務器	\$1,036 USD	十 TB HDD 40% 重複資料刪除	美國東部 (維吉尼亞北部)	分享到 3

伺服器名稱	費用	組態	區域	Share (分享)
		每秒 128 兆比特		
		10 TB 資料備份		
		Multi-AZ		
DFSN視窗EC2實例	27 美元 USD	t3a.medium 2 vCPUs 4 GiB 記憶體	美國東部 (維吉尼亞北部)	DFSN實例

大型SSD檔案系統的年度成本為 68,592 美元。分割工作負載的年度成本為 50,640 美元。在此範例中，將工作負載與適當的後端儲存比對時，可節省 26% 的成本。如需有關定價估算的詳細資訊，請參閱估算[AWS Pricing Calculator](#)值。

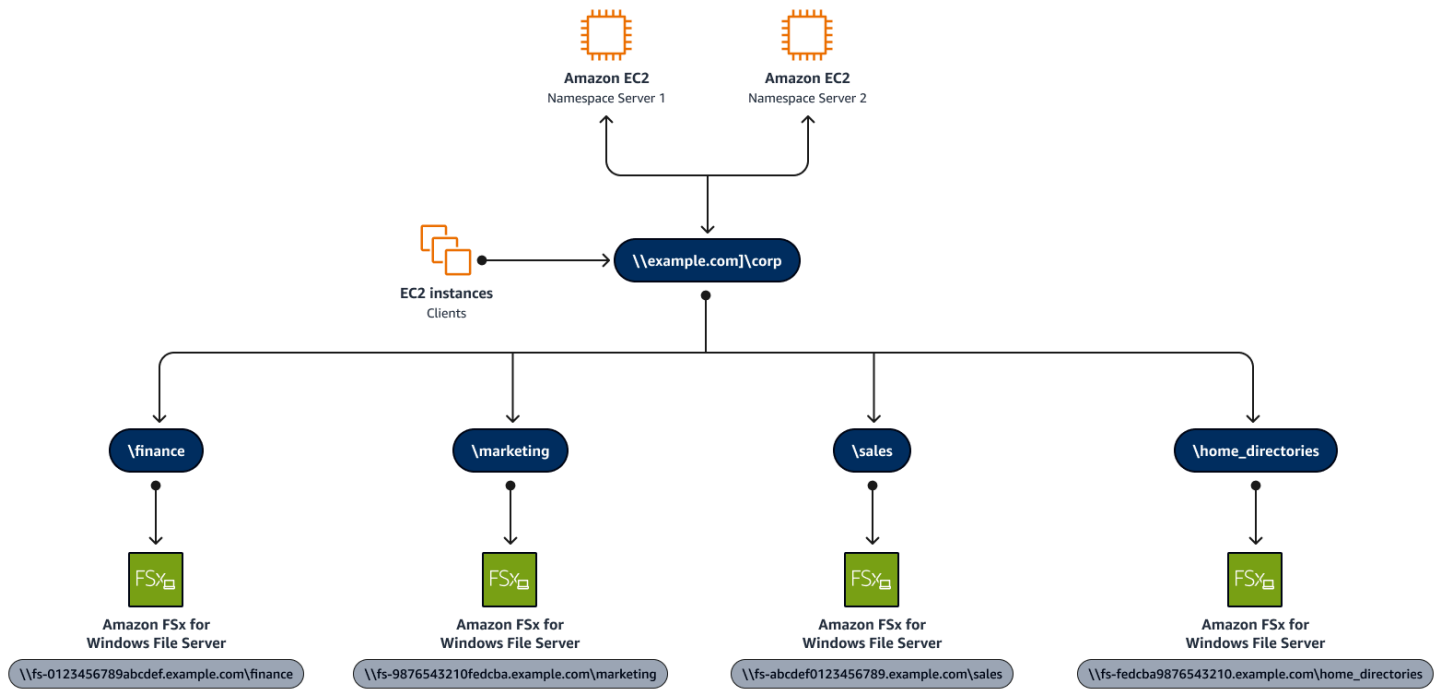
成本最佳化建議

若要部署重複資料刪除解決方案，您必須根據資料類型、I/O 大小和 I/O 存取模式來設定 [Microsoft DFS 命名空間](#)。每個命名空間最多支援 50,000 個檔案共用和數百 PB 儲存容量的彙總。

它最有效地選擇分片慣例，將 I/O 平均分配到您計劃使用的所有檔案系統。監控您的工作負載將有助於進一步優化或降低成本。如果您需要協助衡量 Amazon FSx 檔案系統的效能資訊，請參閱 [FSxWindows 檔案伺服器說明文件中FSx的 Windows 檔案伺服器效能](#)。

選擇分片策略後，您可以使用DFS命名空間將檔案系統分組，以便輕鬆存取共用。如此一來，使用者就能看到一個同質檔案系統，而實際上他們正在使用專門建置的使用案例存取各種不同的檔案系統。請務必使用適當的命名慣例來建立共用，以便您的使用者可以輕鬆解密共用設計的工作負載。標籤生產和非生產共用也很重要，因此最終使用者不會錯誤地將檔案放置在錯誤的檔案系統中。

下圖顯示如何使用單一DFS命名空間做為多個 Amazon FSx 檔案系統的存取點。



請謹記以下幾點：

- 您可以將 Windows 檔案伺服器共FSx用的現有共用新增至DFS樹狀結構。
- Amazon 不FSx能添加到DFS共享路徑的根目錄。您只有一個子資料夾。
- 您必須部署EC2執行個體以提供DFS命名空間組態。

如需有關 DFS-N 組態的詳細資訊，請參閱 Microsoft 說明文件中的[DFS命名空間概觀](#)。如需有關使用 DFS命名空間的詳細資訊，請參閱 < 將[DFS命名空間與 Amazon FSx 搭配 Windows 檔案伺服器使用](#) > 影片 (詳見)。YouTube

其他資源

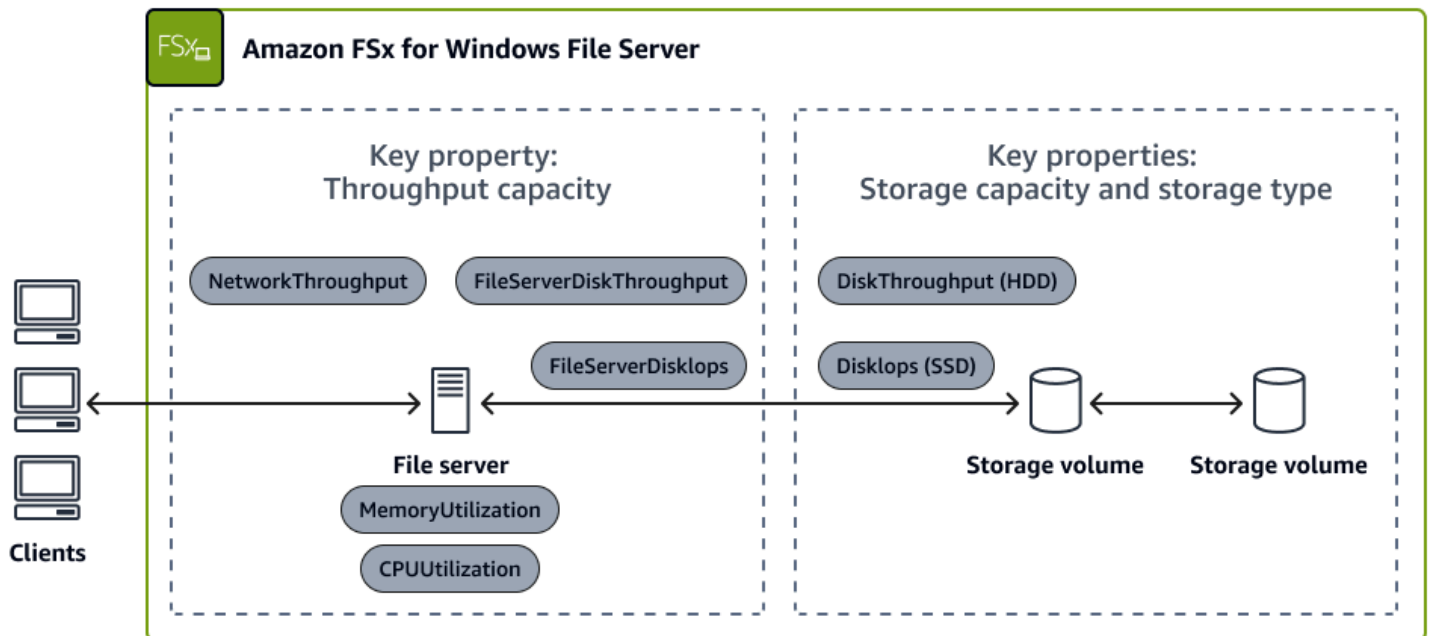
- [使用DFS命名空間將多個檔案系統分組](#) (Amazon FSx 文件)
- [逐步解說 6：使用碎片擴展效能](#) (Amazon FSx 文件)
- [將DFS命名空間與 Amazon FSx 搭配 Windows 檔案伺服器使用](#) (AWS 實驗室)

了解 Amazon 中的HDD大量使用 FSx

概觀

Amazon Windows 檔案伺服器提供獨立FSx於檔案系統容量選擇輸送量的彈性。提供兩種容量設定：HDD(硬碟機) 和 SSD (固態硬碟)。[EBSst1 磁碟機](#)用於中HDD的檔案系統儲存。[EBSio1 驅動器](#)用於SSD。

下圖顯示輸送量與儲存設定之間的關係。



使用基HDD礎儲存，您會收到 12 個IOPS基準，其中包含 80 個突發磁碟 IOPS (IOPs每 TiB 的儲存體)，輸送量為 12 MB /秒基準，每秒 80 個突發 MB (每 TiB 的儲存體)。例如，如果您的共用大小為 50 TB，則輸送量和IOPS。

Amazon FSx 視窗文件伺服器提供 80 突發IOPS。當您的使用率低於基準速率時，系統會自動重新填入突發積分，並在您的使用率高於基準速率時自動消耗。例如，如果您的工作負載在一小時內僅使用 10 IOPS /TB (低於基準速率 2 IOPS TB)，則可以在接下來的一小時內使用 14 IOPS /TB (比基準線 IOPS高出 2 TB)，然後再次耗盡突發積分。

對於檔案操作，Amazon FSx for Windows 檔案伺服器可提供一致的低於一毫秒的儲存延遲，以及低於 10 毫秒的SSD儲存延遲。HDD對於所有檔案系統 (包括具有HDD儲存裝置的檔案系統)，Amazon FSx for Windows File Server 會在檔案伺服器上提供快速 (記憶體內) 快取，因此無論儲存類型為何，您都可以為主動存取的資料獲得高效能和低於一毫秒的延遲。

在適當的情況下，使用HDD儲存裝置有助於降低整體儲存容量的成本，並為您的需求提供可靠的儲存平台。

成本影響

Amazon FSx Windows 檔案伺服器效能取決於三個因素：儲存容量、儲存類型和輸送量。網路 I/O 效能和記憶體內快取大小完全取決於輸送量容量，而磁碟 I/O 效能則由輸送量容量、儲存類型和儲存容量的組合決定。

雖然SSD建議用於 I/O 密集型工作負載，但有各種各樣的工作負載可以滿足HDD效能規格的需求。HDD儲存裝置專為廣泛的工作負載而設計，包括主目錄、使用者和部門共用，以及內容管理系統。例如，如果您的使用者只需要低延遲存取支援目前專案的資料，那麼您儲存的大部分資料不常存取。

您可以使[AWS Pricing Calculator](#)用提供 20 TB SSD 與中HDD檔案系統的比較us-east-1。如下表所示，即使沒有節省重複資料刪除功能，在比較HDD檔案系統與檔案系統時，成本差異仍顯著。SSD

Amazon FSx 文件系統配置	每月成本
20 TB 異地同步備份 SSD () us-east-1	\$4,699.30
20 TB 異地同步備份 HDD () us-east-1	542.88 美元
估計每月儲蓄	\$4,156.42

Note

如需更多FSx節省 Windows 檔案伺服器的費用，請參閱本指南的[啟用 Amazon 中的重複資料刪除—FSx節](#)。

藉由正確識別您的效能需求，您可以選擇適合工作負載的儲存裝置，並降低成本。

成本最佳化建議

如果您決定使用HDD儲存裝置，請測試您的檔案系統，以確保檔案系統符合您的效能需求。HDD相對於儲存裝置，SSD儲存的成本較低，但IOPS每單位儲存體的磁碟輸送量和磁碟層級較低。它可能適用於 I/O 需求較低的一般用途使用者共用和主目錄、不常擷取資料的大型內容管理系統，或具有少量大型檔案的資料集。

無法變更現有檔案系統的儲存區類型。若要轉換 Amazon FSx for Windows 檔案伺服器檔案系統的儲存類型，您必須備份現有的檔案系統，並將其還原到具有所需儲存類型的新檔案系統。如果您想要將現有的SSD檔案系統轉換為HDD檔案系統，請注意HDD，最小容量為 2 TB。

若要使用不同儲存類型還原備份，請執行下列動作：

1. [備份您現有的檔案系統。](#)
2. [使用HDD儲存類型建立新的 Amazon FSx 檔案系統。](#)
3. 將備份還原到具有所需儲存類型的新文件系統。
4. 確認新檔案系統具有正確的儲存類型，且資料完整無缺。

在將變更移至生產環境之前，建議您先分析 Amazon FSx 檔案系統的效能，並確認變更是否可接受。如需詳細指引，請參閱 AWS 儲存部落格上的[文章，使用新指標最佳化適用FSx於 Windows 檔案伺服器的 Amazon 效能。](#)

其他資源

- [使用 Amazon 優化成本 FSx](#) (Amazon FSx 文檔)

使用單一可用區域

概觀

本節說明何時使用適用[FSx於 Windows 檔案伺服器的 Amazon](#) 單一可用區域實作更有益。它涵蓋移至單一可用區域可降低成本，同時仍可讓您使用 Amazon FSx for Windows 檔案伺服器做為受管檔案儲存服務的案例。我們建議您針對生產工作負載為 Amazon FSx 實作單一可用區域。這有助於確保您擁有多個可用區域的備援。

成本影響

與多個可用區域實作相比，單一可用區域檔案系統可降低約 40% 的成本。使用多個可用區域檔案系統時，您每月需要支付每 GB 0.230 美元，SSD而單一可用區域檔案系統則為每月 0.025 USD，HDD而單一可用區域檔案系統則為SSD每月 0.013 USD。HDD您可以查看成本的比較，並使用建立您自己的估算值[AWS Pricing Calculator](#)。

對於 10 TB 檔案系統而言，這可能是針對多個可用區域每月支付約 1,200 美元的費用，或單一可用區域每月支付 680 USD 的差異。此[範例](#)針FSx對 Windows 檔案伺服器檔案系統使用 10 TB SSD。重複資料刪除的估計可節省 50%。整體而言，單一可用區域的入門成本較低，但下一節會涵蓋一些警告。

成本最佳化建議

單一可用區域部署

SLAs為了確保單一可用區域適合，請考慮您自己的內部內部儲存在 Windows 檔案伺服器上FSx的資料。這需要瞭解您是否必須提供SLAs給客戶 (內部和外部)，以及 Amazon FSx 單一可用區域的三九可用性是否仍然允許您滿足這些需求。SLAsFSx對於具有單一可用區域的 Windows 檔案伺服器，仍有 99.9% 的執行時間。SLA適用FSx於多個可用區域的 Amazon 大於 99.99%。對於關鍵任務工作負載，我們建議您在單一可用區域上使用多個可用區域，即使需要額外付費也是如此。

單一可用區域部署非常適合工作負載，例如SQL伺服器資料庫的備份。他們可以通過HDD層提供低成本的存儲，同時仍然為您提供一致的正常運行時間。如果生產工作負載需要更高層級的可用性，例如高可用性SQL伺服器或生產應用程式存取，則單一可用區域並不適合您的工作負載。對於備份、非生產測試和開發環境，Amazon FSx 單一可用區域實作可以降低您的營運成本。

Amazon FSx 單一可用區域檔案系統運作良好的一個使用案例是在使用多個 Amazon FSx 單一可用區域檔案系統的生產環境中，如同使用「永遠開啟」可用性群組的高可用性SQL伺服器叢集中的每個伺服器儲存。如需詳細資訊，請參閱 AWS 儲存部落格[上 AWS張貼的 < 最佳化高可用性SQL伺服器部署的成本 >](#)。

多區域複製

如果您想要利用 Amazon FSx 的多區域複製功能，那麼使用單一可用區域檔案系統 (僅可使用單一可用區域檔案系統的檔案系統) 降低成本的潛在選項。您可以部署支援使用原生 Microsoft DFS-R 的[單一可用區檔案系統](#)。DFS-R 具有跨區域和多個站點自動複製數據的功能。如需有關使用 Amazon 設定 DFS-R 的詳細資訊FSx，請參閱 Amazon FSx 文件中的[使用 Microsoft 分散式檔案系統複製](#)。

另一種節省多區域成本的替代方法是使用 AWS Storage Gateway。這可讓您在另一個區域實作[Amazon FSx 檔案閘道](#)，以進行 Amazon FSx 的多區域存取。如需詳細資訊，請參閱本指南的[AWS Storage Gateway](#) 一節。

如果您跨區域工作，則必須考慮跨區域資料流量的資料傳輸成本。跨區域移動的流量會產生 0.02/Gb 的費用。因此，如果您在大量的情況下有一致的資料變更，這將增加您的整體成本。[例如](#)，1 TB 的資料傳輸量大約等於 20.48 美元。

Maintenance window (維護時段)

如果您在 Amazon 搭配使用單一可用區域，則維護時段是重要的考量因素FSx。在維護期間，由於基礎 Windows 伺服器的例行軟體修補程式，Amazon FSx 檔案系統大約 20 分鐘無法使用。如果您使用檔案系統進行隔夜備份，請相應地調整 Amazon FSx 維護時段，以避免備份期間中斷。您可以在建立 Amazon FSx 檔案系統後調整[維護時段](#)。

其他資源

- [可用性和耐久性：單一可用區和異地同步備份檔案系統](#) (Amazon FSx 文件)
- [Amazon FSx 視窗檔案伺服器定價](#) (AWS 網站)

AWS Storage Gateway

AWS Storage Gateway 是一種混合雲存儲服務，將內部部署環境與 AWS 雲存儲連接起來。它可讓您將現有的內部部署基礎結構與無縫整合 AWS，讓您能夠從雲端儲存和擷取資料，並在混合式環境中執行應用程式。對於 Windows 工作負載，您可以使用 Storage Gateway，使用原生 Windows 通訊協定 (例如SMB和) 來儲存和存取資料NFS。您可以使用 Storage Gateway，AWS 藉由使用內部部署硬體和軟體做為通往雲端的橋樑，降低在上執行 Windows 工作負載的相關成本。這使您可以充分利用的可擴展性和成本效益，AWS 而無需對現有基礎架構進行重大變更。

在 Storage Gateway 的保護下，您可以取得 Amazon S3 檔案閘道、Amazon FSx 檔案閘道、磁帶閘道和磁碟區閘道。S3 檔案閘道和FSx檔案閘道最常用於 Microsoft 工作負載。

Amazon S3 檔案閘道

[Amazon S3 檔案閘道](#)可讓您將檔案存放在 Amazon S3，同時使用傳統SMB共用功能為使用者提供存取權。這提供了熟悉的使用者界面，並透過將資料存放在 Amazon S3 並利用各種 Amazon S3 儲存層級來協助降低成本。您可以使用 S3 智慧型分層實作 Storage Gateway，協助您將生命週期檔案自動移至成本最低的儲存層級，進一步降低成本。建議使用 S3 檔案閘道進行向外延展、唯讀存取、快速重複讀取 (從快取) 和資料庫傾印。通常不建議用於高效能或高可用性寫入、編輯檔案或部門共用。

Amazon FSx 檔案閘道

使用 [Amazon FSx Windows FSx 檔案系統時](#)，[Amazon 檔案閘道](#)也可以節省成本。您可以站立FSx檔案閘道，提供對其他區域中 Amazon FSx 檔案系統的本地化存取，以避免擁有兩個獨立檔案系統的成本。如果您有多個內部部署檔案伺服器，而且想要合併這些伺服器以避免支付多個硬體裝置的費用，這也會很有幫助。

成本影響

Amazon S3 檔案閘道

設定 S3 檔案閘道非常簡單，因為您可以使用 Storage Gateway 的啟動精靈。您可以在幾分鐘內使用 AWS 環境中的EC2執行個體來部署閘道。設定閘道之後，您可以將 Storage Gateway 共用設定為可透

過SMB和通NFS訊協定存取。對於一般的 Windows 工作負載，您也可以使用此設定來利用作用中目錄環境並設定檔案共用的權限。您可以有效地將 Storage Gateway 集成到正常使用中，因為它將作為典型的 Windows 文件共享工作。檔案和資料夾會儲存為物件和存NTFS取控制清單 (ACLs) 做為中繼資料。

下表將 10 TB 儲存空間的成本與三個可用的儲存選項進行比較：

- FSx視窗檔案伺服器
- Amazon S3 檔案閘道
- Amazon 彈性塊商店 (AmazonEBS)

如果您使用 Amazon S3，將 10 TB 儲存存放在相當便宜的價格中，因為您可以將資料分割成不同的用量層。在定價估算中，使用 S3 智慧型分層作為其定價彈性。這包括 S3 標準中的 80%、10% 的不常存取，以及 Amazon S3 冰川的 10%。雖然您可以使用 S3 Glacier，但請務必設定適當的生命週期規則，以確保移至 S3 Glacier 的任何檔案都不需要立即存取。S3 Glacier 純粹用於存檔用途，而不是常規訪問用途。

儲存系統	10 TB 儲存空間的成本	區域
FSx適用於 Windows 檔案伺服器 (假設省下 50% 的重複資料刪除功能)	683.20 元 USD SSD	美國東部 (維吉尼亞北部)
Amazon S3 檔案閘道	智慧型分層 USD	美國東部 (維吉尼亞北部)
Amazon EBS	1,335.69 元 USD GP3	美國東部 (維吉尼亞北部)

考慮下列各項：

- 在 S3 Glacier 中，除非您使用[RestoreObject](#)API將物件還原回 Amazon S3，否則您會收到一般 I/O 錯誤。我們建議您使用 Amazon CloudWatch 事件針對此 I/O 錯誤使用通知。這樣，您的操作團隊就可以對用戶在他們可能需要訪問的文件上得到此錯誤的反應。如需這些錯誤的詳細資訊，請參閱 Amazon S3 檔案閘道文件 `InaccessibleStorageClass` 中的[錯誤](#)。
- 除了 S3 Glacier 的存取限制之外，Storage Gateway 上的[每個物件/資料夾只ACLs允許 10](#) 個。在您決定使用 Storage Gateway 之前，請確定您不需要超過 10 個ACL項目。

Amazon FSx 檔案閘道

檔案閘道與 Amazon S3 檔案閘道類似，可讓您存取可長期保留資料的檔案系統。FSx 在 Amazon S3 檔案閘道中，資料駐留在 Amazon S3 中。對於 FSx 檔案閘道，您的資料位於 FSx 於 Windows 檔案伺服器上。雖然異地同步備份選項適用於 FSx 於 Windows 檔案伺服器，但沒有多區域選項。如果您有全球公司或遠端辦公室，您可能需要提供一個在地理位置上更接近使用者的共用儲存平台，以避免延遲。如果您要部署另一個 Amazon FSx 檔案系統，這會增加全新的 Windows 檔案伺服器檔案系統和必要儲存的成本。FSx 若要避免建立全新的檔案系統並複製成本，您可以在次要區域部署 FSx 檔案閘道。這可為使用者提供對檔案的本地化存取，同時有助於降低整體成本。

儲存系統	10 TB 儲存空間的成本	區域
Amazon FSx 文件服務器	683.20 元 USD SSD	美國東部 (維吉尼亞北部)
Amazon FSx 檔案閘道	單一閘道器	美國東部 (維吉尼亞北部)

Note

上表中的價格是根據 [Storage Gateway 定價](#) 而定。

請謹記以下幾點：

- FSx 檔案閘道可協助您每月節省約 180 美元 (或每年 2100 美元) 的多區域工作負載。
- FSx 檔案閘道的資料傳輸費用要低得多，因為它只需要快取定期存取的檔案，而不是完整的次要副本。
- 雖然您可以在不同區域中 FSx 針對 Windows 檔案伺服器進行兩個部署，並使用 AWS Backup 或保持更新 AWS DataSync，但這兩個選項都不是接近即時的。

成本最佳化建議

Amazon S3 檔案閘道

S3 檔案閘道為儲存檔案提供了低成本的選項，但是在實作和使用檔案系統方面，還有一些問題需要考慮。例如，S3 檔案閘道需要使用虛擬機器才能執行 Storage Gateway 軟體。在中 AWS，依預設，Storage Gateway 是使 EC2 用 m5.xlarge 執行個體在 Amazon 中部署。如果您想要降低內部部署

儲存體成本，可以將 Storage Gateway 部署為虛擬化平台 (例如 VMware Hyper-V) 上的虛擬應用裝置。

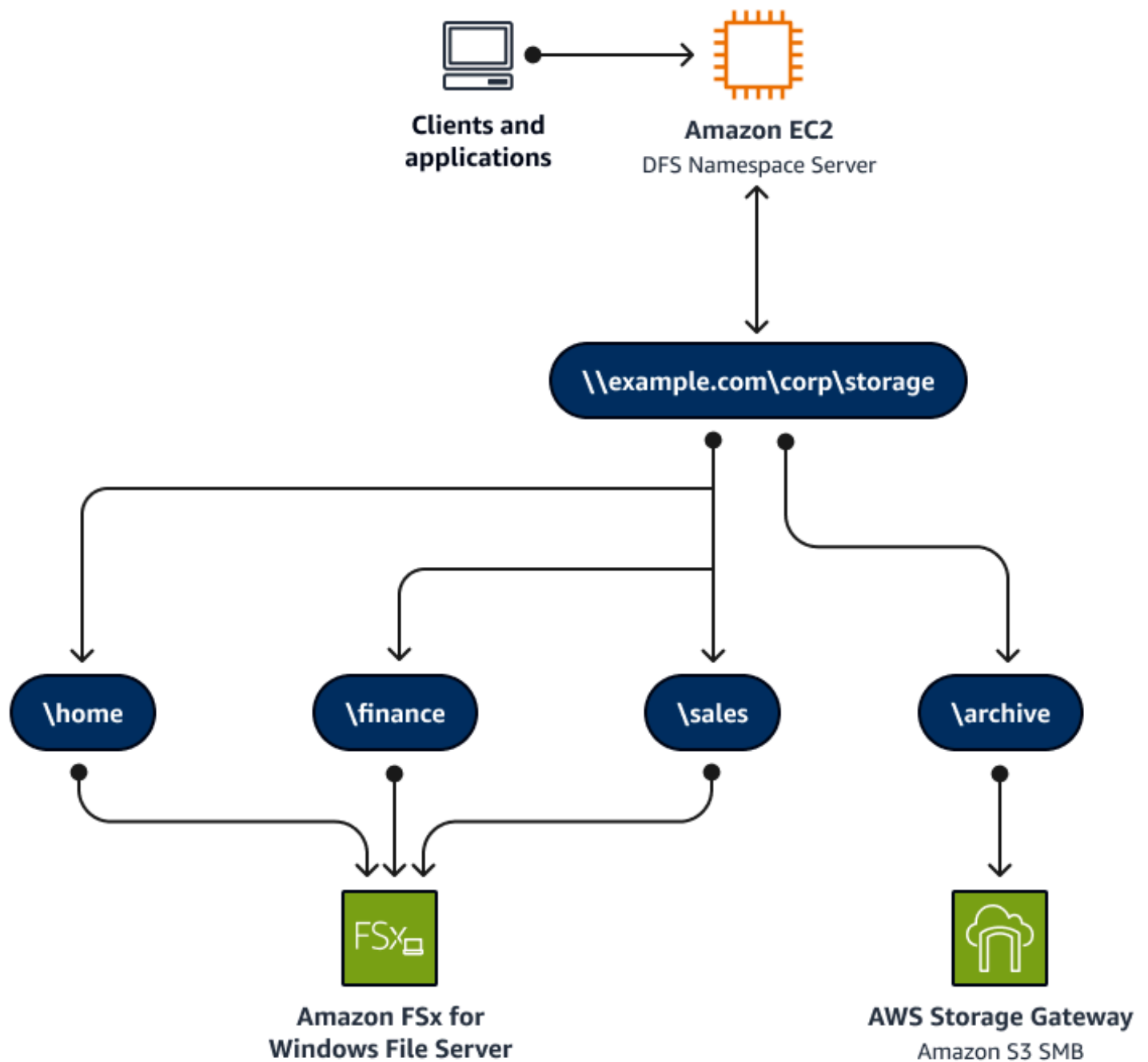
高可用性考量

執行 Storage Gateway 是存取檔案的單一失敗點。為了避免不必要的停機時間，我們建議您實施嚴格的存取控制，讓使用者可以對 Storage Gateway 執行個體進行變更或停止和啟動。此外，對於進行部署 AWS，使用 Amazon Data Lifecycle Manager 建立路由快照以快速復原 Storage Gateway 實作也很有幫助。如果您正在使用內部部署執行 Storage GatewayVMware，則可以將其設定為[高可用性](#)。

執行多個檔案系統

將日常使用的檔案工作負載與封存工作負載分開，有助於避免不必要的儲存成本。Storage Gateway 必須與 Windows 檔案伺服器檔案系統—FSx起部署的能力。透過使用[DFS命名空間](#)，您可以顯示在 Windows 檔案伺服器上FSx執行的主要日常使用儲存，以及在 Amazon S3 中執行的儲存 (可透過 Storage Gateway 存取)。

下圖顯示如何使用單一DFS命名空間做為不同後端儲存選項的前端存取點。



用戶端會導向至資料夾結構，例如 `\\example.com\` 儲存空間。此主目錄包含子目錄。Windows 檔案伺服器檔案系統包含正常存取的檔案共FSx用。您可以將 Storage Gateway 上建立的檔案共用用於封存資料。使用者可以手動將項目封存至封存資料夾，或者您可以建立程序，自動將一些檔案從一般檔案共用移至封存資料夾。

考慮下列各項：

- 檢閱您的儲存需求，並為[快取提供足夠的儲存空間](#)。
- 將閘道新增至您的作用中目錄組態，並使用[標準 Windows ACLs 存取檔案](#)。

FSx檔案閘道

FSx檔案閘道的部署類似於 S3 檔案閘道的部署，但如果您使用啟動精靈，它會更容易。如需詳細指示，請參閱 [Amazon FSx 檔案閘道文件中的步驟 3：建立和啟用](#) Amazon FSx 檔案閘道。在環境中部署FSx檔案閘道後，您可以將其與現有 Amazon FSx 檔案系統建立關聯，並取得檔案的存取權。

儲存是部署FSx檔案閘道時的主要考量因素。默認存儲提供 150 GB，這是用於緩存文件的相當大量空間。針對可用空間不足建立監控警示，有助於在不過度配置的情況下，正確調整儲存

其他資源

- [AWS Storage Gateway 資源](#) (AWS 文件)

Active Directory

執行 Windows 伺服器的 Amazon 彈性運算雲端 (Amazon EC2) 是一種安全、可靠且高效能的環境，可用於部署以 Windows 為基礎的應用程式和工作負載。您可以快速佈建執行個體，並視需要擴展或縮減，而只需按使用量付費。使用中目錄服務作為 Windows 伺服器環境中識別管理的主要來源。

本節涵蓋下列主題：

- [Amazon EC2 上的自我管理活動目錄](#)
- [AWS Managed Microsoft AD](#)
- [AD Connector](#)

Amazon EC2 上的自我管理活動目錄

概觀

本節提供有關降低在 Amazon 彈性運算雲 (亞馬遜 EC2) 上運行活動目錄的成本的建議。主要焦點是確定您可以適當調整 Active Directory 網域控制站的大小，並使用的彈性 AWS 雲端來調整您的環境所需。AWS 可協助您輕鬆停止執行個體並調整其大小以符合您不斷變化的需求；如果擴充速度太快，則可縮小執行個體的大小。選擇正確的執行個體大小和類型可大幅節省成本。

成本影響

下表顯示選擇高載執行個體系列執行個體與一般用途執行個體之間的差異。這種選擇每個月可以為您節省大量資金。適當的規劃和調整執行個體大小可協助您管理成本。

執行個體類型	執行個體的數目	vCPU	記憶體	費用
t3a.medium	2	2	8	每月
m5a.large	2	2	8	每月

如需有關成本的詳細資訊，請參閱 AWS Pricing Calculator [估算值](#)。

每月節省 178.12 美元的網域控制站每年可省下超過 2,000 美元的費用。請記住，在一個帳戶中只有兩個域控制站的佔用空間很小。大規模搭配多個帳戶和額外的網域控制站，這樣的節省可大幅降低成本。

成本最佳化建議

當您部署您的使用中目錄環境時，Microsoft 會提供[容量規劃建議](#)。我們建議您在規劃或擴展 Active Directory 環境時考慮下列主要元件：

- 記憶體
- 網路
- 儲存
- 處理器

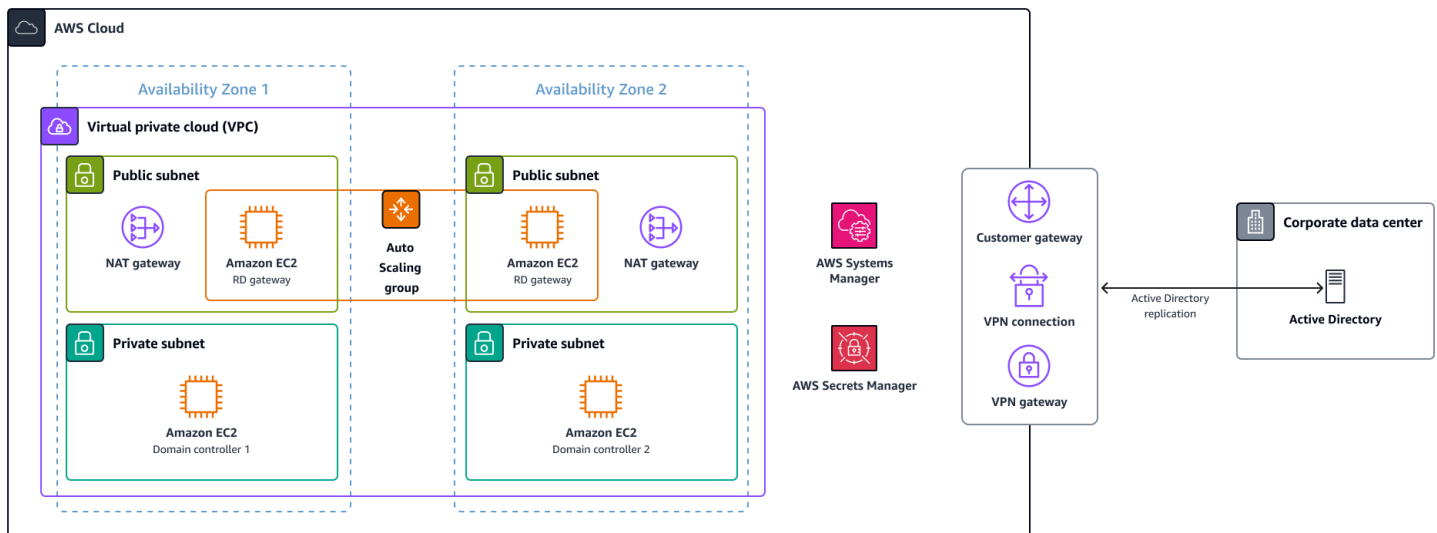
在牢記這些主要元件的同時，您可以選取對 Active Directory 環境有意義的執行個體類型 AWS。本節涵蓋 AWS 部署案例的幾個作用中目錄範例。這些案例明確表示 AWS，如果您不打算處理與內部部署環境中相同數目的使用者和電腦，就不需要在中複寫內部部署環境。

下表針對 AWS 佔用空間重點說明 vCPU、記憶體和磁碟的重要元件。

元件	估計
儲存空間/資料庫大小	每位使用者可使用 40 至 60 KB
RAM	資料庫大小 基本作業系統建議 第三方應用
網路	1 GB
CPU	每個核心可同時使用 1,000 名使用

混合式部署案例

下圖顯示使用中目錄的混合式部署的範例架構。



如圖所示，您通常擁有內部部署的佔用空間，然後將其擴展到 AWS 雲端。在遷移的初始階段，您通常不會將所有使用者和伺服器都部署在 AWS。這就是為什麼最初部署較小尺寸的佔用空間以節省遷移工作的成本很重要的原因。

如果您要維護內部部署使用伺服器和使用者在 AWS 內部部署進行驗證的內部部署佔用空間，則不需要中的網域控制站使用相同的佔用空間。遵循 Active Directory 的最佳做法，您可以實作適當的 [Active Directory 站台和服務](#)，以驗證使用者和電腦到您的內部部署使用量，而只驗證您的 AWS 足跡給中 AWS 的網域控制站。這可讓您 AWS 透過限制只使用 AWS 資源，而不是所有內部部署基礎結構，以避免使用過大的 Active Directory 佔用空間。如需設計混合式安裝程式的指引，請參閱 Microsoft 文件中的 [正確放置網域控制站和網站考量](#)。

透過適當的調整大小來優化 AWS 移轉

如果您要為使用者部署新的 Active Directory 執行個體，或打算針 AWS 對 Active Directory 基礎結構完全移轉至，建議您針對上表中選擇的執行個體，根據 Microsoft 針對 vCPU、記憶體和磁碟空間的建議規劃大小。

如果這是新的佔用空間，您可以從小規模開始，並利用輕鬆 [變更執行個體類型](#) 的功能，以便在環境成長時調整其大小 AWS。本指南的 [Amazon EC2 上的 Windows](#) 一節說明如何監控和檢閱上的 CPU 和記憶體使用率 AWS。這樣，您就知道何時增加 EC2 執行個體的大小。

如果您要將內部部署 Active Directory 環境完全移轉至 AWS，您可以實作相同的大小規模計劃，以確保適當的效能。在複製內部部署中的內容之前 AWS，我們建議您完成 Active Directory 環境的全面檢閱。這可協助您避免過度佈建。請務必使用效能監視器來收集有關您現有網域控制站的流量和使用量的資訊。這可以讓您了解整體使用情況，以便您可以正確規模並最終降低成本。

最佳化作用中目錄 AWS

如果您正在執行 Active Directory AWS，也必須持續監控使用率，並視需要變更執行個體大小以減少支出。您可以使用 AWS Compute Optimizer 來取得您正在執行之資源的相關資訊 AWS。如需使用 Compute Optimizer 調整 Windows 工作負載大小的相關資訊，請參閱本指南中的 [Amazon EC2 上的 Windows](#) 一節。如需更全面的深入探討，您可以使用效能監視器來監視 Active Directory 網域控制站的使用率、評估效能，然後相應地調整大小。

您也可以使用 CloudWatch 來監視網域控制站的效能。若要最佳化您的網域控制站 (擴充或縮減)，您可以使用中提供的指標 CloudWatch 來協助您做出正確的決策。您可以使用 CloudWatch 代理程式來設定要傳送資料收集的自訂效能監視器測量結果。如需指示，請參閱 [如何使用 CloudWatch 代理程式來檢視 Windows 伺服器上效能監視器的度量？](#) 在 AWS 知識中心。

部署 CloudWatch 代理程式之後，您可以在代理程式組態檔案下設定下列測量結果 `metrics_collected`：

指標類別	指標名稱
資料庫對執行個體 (NTDSA)	資料庫快取命中率
I/O 資料庫讀取平均延遲	
I/O 資料庫讀取數/秒	
I/O 記錄寫入平均延遲	
DirectoryServices (新台幣)	連結時間
DRA 擱置中複寫作業	
DRA 擱置複寫同步處理	
DNS	遞查詢/秒
每秒遞迴查詢失敗	
接收的TCP 查詢/每秒	
收到的查詢總數/秒	
每秒傳送回應總數	

指標類別	指標名稱
收到的UDP查詢/每秒	
LogicalDisk	平均磁碟佇列長度
% 可用空間	
記憶體	% 使用中的認可位元組
長期平均待命快取存留期	
網路介面	傳送位元組/秒
接收的位元組/秒	
目前頻寬	
NTDS	ATQ 預估佇列延遲
ATQ 請求延遲	
DS 目錄讀取/秒	
每秒 DS 目錄搜尋	
DS 目錄寫入次數/秒	
LDAP 用戶端工作階	
每秒 LDAP 搜尋	
LDAP 成功繫結 (每秒)	
處理器	% 處理器時間
整個系統的安全統計	凱伯洛斯身份驗證
NTLM 驗證	

其他資源

- [使用中的目錄網域服務 AWS：合作夥伴解決方案部署指南](#) (AWS 文件)
- [作用中目錄網域服務的容量規劃](#) (Microsoft 說明文件)
- [在 EC2 執行個體上執行作用中目錄的設計考量](#) (AWS 白皮書)

AWS Managed Microsoft AD

概觀

AWS Directory Service for Microsoft Active Directory 也稱為 AWS Managed Microsoft AD，由 Windows 伺服器作用中目錄提供動力，並由管理 AWS。您可以使用 AWS Managed Microsoft AD 將廣泛的使用中目錄感知應用程式移轉到 AWS 雲端。AWS Managed Microsoft AD 適用於各種原生的活動目錄應用程式和服務。它也支援 [AWS 受管理的應用程式和服務](#)。雖然 AWS Managed Microsoft AD 由於服務及其計費機制而沒有太多的成本優化槓桿，但有一些設計原則可以幫助您將成本保持在最低限度。

成本影響

由於 AWS Managed Microsoft AD 是以目前 SKU 為基礎的受管理服務，因此規模是相對簡單的程序。目前有兩種規模 SKU 可供選擇：標準版和企業版。其他 SKU 包括目錄共用、新增其他網域控制站 (包括其他區域)，以及跨區域資料傳輸。

成本最佳化建議

AWS Managed Microsoft AD 標準版和 AWS Managed Microsoft AD 企業版之間存在差異。企業版支援多達 500,000 個活動目錄對象，125 個帳戶共享 (軟限制)，並具有多區域支持。標準版最多可支援 30,000 個 Active Directory 物件、五個帳戶共用 (軟限制至大約 30 個上限)，而且不支援多區域。

在選取目錄類型之前，需要考慮的問題如下：

- 是否需要多區域支援？
- 該目錄是否將與 30 多個帳戶共享？
- 活動目錄對象計數是否將超過 30,000？

如果上述任何問題的答案是肯定的，則需要企業版。如果所有問題的答案都是否定的，我們建議您從標準版開始。

Note

您可以將目錄從標準版升級到企業版，但目錄無法降級。部署標準版不會通過單向門。如果您希望將目錄升級到企業版，請聯繫 AWS。

當您在 AWS Managed Microsoft AD 企業版中共用目錄時，每個共用都需要支付費用。這低於在每個帳戶中部署目錄的成本，但請記住，如果未勾選，共用成本可能會增加。我們建議您只與包含 Amazon 關聯式資料庫服務 (Amazon RDS) 和 Windows 檔案伺服器專用 Amazon FSx 的帳戶共用目錄，因為只有這些服務支援此功能。請記住，您可以選擇將適用 FSx for Windows File Server 與自我管理的作用中目錄整合，包括 AWS Managed Microsoft AD。如果其他帳戶只需要 Amazon FSx，則您可以針對自我管理的 Amazon FSx 部署，AWS Managed Microsoft AD 而不需要共用目錄。

決定何時部署其他網域控制站時，請記住，只 AWS Managed Microsoft AD 支援相同 VPC 中不同可用區域中的兩個子網路。新增其他網域控制站不允許您新增其他子網路。若要判斷是否因為效能問題而必須新增其他網域控制站，請檢閱中的[網域控制站效能指標 CloudWatch](#)。這會告訴您是否有一個或所有網域控制站不堪重負。如果您判斷只有一個網域控制站不堪重負，新增其他網域控制站不會減輕負載，而且您需要深入研究應用程式不會在目前可用的網域控制站之間進行負載平衡。如果大量使用所有網域控制站，新增額外的網域控制站可減少現有網域控制站的負載。如需有關如何自動擴展的指示，請參閱 AWS 安全部落格中的[如何根據使用率指標自動 AWS Managed Microsoft AD 調整規模](#)。

如果您將目錄延伸到多個區域，建議您不要使用目錄 NETLOGON 或 SYSVOL 共用來儲存檔案。所有網域控制站會複寫這些共用的內容。不將共用資料用於檔案儲存，可將資料傳輸成本降至最低。

您也可以選擇使用註冊企業合約 AWS。企業合約可讓您選擇量身打造最符合您需求的合約。如需詳細資訊，請參閱[企業客戶](#)。

其他資源

- [AWS Managed Microsoft AD 配額](#) (AWS Directory Service 文件)
- [AWS Directory Service 定價](#) (AWS 網站)
- [使用中目錄網域服務 AWS](#) (AWS 白皮書)

AD Connector

概觀

[AD Connector](#) 是一種代理服務，可讓您輕鬆將現有的現場部署 Microsoft Active Directory 連接到相容的 [AWS 應用程式](#)，例如 Amazon WorkSpaces、Amazon QuickSight，以及亞馬遜彈性運算雲端 (Amazon EC2) 執行個體的無縫網域加入，而不會在雲端中快取任何資訊。您可以使用 AD Connector 將一個服務帳戶新增至您的作用中目錄。AD Connector 不需要目錄同步處理，也不需要裝載同盟基礎結構的成本和複雜性。由於服務的性質及其計費機制，AD Connector 的成本最佳化槓桿並不多，但您可以遵循本節中的設計建議，將成本降至最低。

成本影響

AD Connector 是以預設 SKU 為基礎的受管理服務。這使得調整大小的過程很簡單。有兩種尺寸 SKU 可供選擇：小尺寸和大尺寸。您可以使 [AWS Pricing Calculator](#) 用涉及 AD Connector 的成本估算。

成本最佳化建議

除了後端運算資源之外，小型和大型連接器大小之間沒有區別。

在選取目錄類型之前，需要考慮的問題如下：

- 是否有大量 (10,000 +) 使用與 AD Connector 整合的 AWS 應用程式的作用中使用者？
- 使用者是多個、深層或循環巢狀群組的成員嗎？

如果兩個問題的答案都是否定的，我們建議您從小尺寸開始。如果您對上述任何一個問題的回答是肯定的，那麼大尺寸可能值得考慮。您可以從較小的 AD Connector 開始，如果目錄因效能而受損，您可以要求將目錄升級為較大的大小。

Note

您可以將 AD Connector 從小型升級到大型，但無法降級 AD Connector。

大部分的效能問題與 AD Connector 器無關，但內部部署 Active Directory 網域控制站因為許多使用者是許多、深層或循環巢狀群組的成員而不知所措。

您也可以選擇使用註冊企業合約 AWS。企業合約可讓您選擇量身打造最符合您需求的合約。如需詳細資訊，請參閱 [企業客戶](#)。

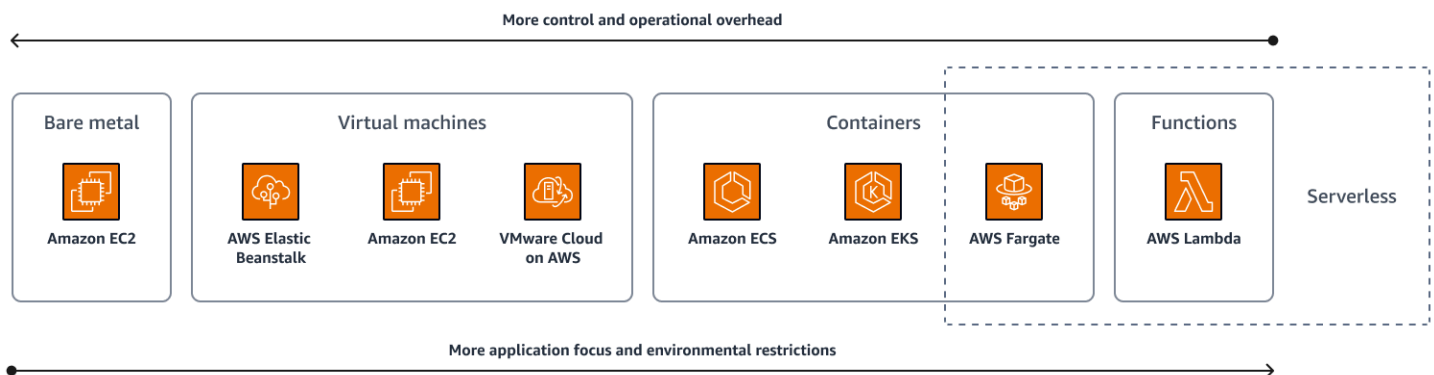
其他資源

- [AD Connector 配額](#) (AWS Directory Service 文件)
- [其他目錄類型定價](#) (AWS 網站)
- [使用中目錄網域服務 AWS](#) (AWS 白皮書)

.NET

開發和部署。 .NET 應用程式是協助您實現雲端運算所提供的規模和敏捷性的重要關鍵。對於許多傳統。 .NET 應用程式是執行應用程式的最合適運算選擇 AWS 是透過 AWS Elastic Beanstalk 或 Amazon 彈性運算雲端 (AmazonEC2) 使用虛擬機器。它也有可能運行。 .NET 視窗和 Linux 容器中的應用程式。

的介紹. .NET 核心使您能夠設計現代。 .NET 充分利用所有雲端優勢的應用程式。現代應用程式可以使用一組傳統的運算選項，也可以針對各種類型的無伺服器環境，包括 AWS Fargate 或 AWS Lambda。 .NET6+ 現在提供高效能的工作負載託管，例如 Gra EC2 viton2 系列的 ARM64 EC2 執行個體。這可讓您存取 Amazon 上可用的最新一代處理器 EC2。這表示您的應用程式可以託管在專用於您工作負載類型的運算上，例如視訊編碼、Web 伺服器和高效能運算 (HPC)。



本節提供了幫助您適應. NET 應用程式可充分利用雲端的優勢，並著重於成本效益。

本節涵蓋下列主題：

- [重構現代。 .NET 並移動到](#)
- [容器化。 .NET 应用](#)
- [使用重力彈執行個體和容器](#)
- [Support 靜態的動態縮放。 .NET 框架應用](#)
- [使用快取來減少資料庫需求](#)
- [考慮使用無伺服器。 .NET](#)
- [考慮專門建置的資料庫](#)

重構現代。NET並移動到

概觀

現代化傳統。NET 框架應用程序可以幫助您提高安全性，性能和可擴展性。一種有效的現代化方法。NET 框架應用程序是將它們遷移到現代。NET 版本 (6 歲以上)。以下是將這些應用程式移至開放原始碼的一些主要優點。NET:

- 透過在 Linux 作業系統上執行，以降低 Windows 授權成本
- 充分利用現代語言的可用性
- 取得最佳化以在 Linux 上執行的效能

許多組織仍在執行舊版的 .NET 框架。這可能會造成安全風險，因為舊版本中的漏洞已不再由 Microsoft 解決。Microsoft 已經結束對最新版本的 .NET 框架 4.5.2、4.6 和 4.6.1。評估繼續執行舊版架構的風險和好處是非常重要的。為了降低風險並降低成本，值得投入時間和精力來重構為現代版本的 .NET。

成本影響

請考慮一般用途 EC2 執行個體類型 (m5)，它提供了運算、記憶體和網路資源之間的平衡。這些實例適用於各種應用程序，例如 Web 服務器，中型數據庫和源代碼存儲庫。

例如，在美國東部 (維吉尼亞北部) 的 Windows 伺服器 (包括授權) 上具有 4 GB vCPUs 和 16 GB 記憶體的隨選 m5.xlarge 執行個體，每月費用為 274.48 美元。Linux 服務器上的相同資源每月花費 140.16 美元。在此範例中，當您從中移轉應用程式時，成本會降低 49%。NET 框架到現代版本的 .NET 並在 Linux 服務器上運行您的應用程序。您的費用可能會根據您在選擇執行個體時選擇的選項 (例如 [EC2 執行個體類型](#)、作業系統、儲存) 而有所不同。您可以使用 [Savings Plans](#) 或 [預留執行個體](#)，進一步優化成本。如需詳細資訊，請使 [AWS Pricing Calculator](#) 用執行成本估算。對於包含 Windows 的執行個體，無論定價模式為何，授權費用均為 [CPU 每小時 0.046 美元/v](#)。

移植這些。NET 框架應用程序到現代。NET 需要開發人員努力。您必須評估應用程式及其相依性，以查看它們是否與目標平台版本相容。[AWS 移植助理](#).NET 是掃描的輔助工具。NET 框架應用程序並生成一個 .NET 相容性評估，協助您將應用程式移植到與 Linux 相容的速度更快。的移植助理 NET 識別與的不相容性。NET，尋找已知的替代物，並產生詳細的相容性評估。移植解決方案之後，您必須手動變更專案，才能使用相依性順利編譯專案。這樣可以減少將應用程式現代化為 Linux 所涉及的手動工作量。如果您的應用程式支援 ARM 處理器，移至 Linux 就能解鎖使用重力子執行個體的功能。這可以幫助您在進一步降低成本方面獲得 20% 的額外成本。如需詳細資訊，請參閱 [電源](#)。[NET5 與 AWS 重力 2：運算部落格中的基準](#)。AWS

還有其他工具，例AWS 如 [.NET 重構](#) 和 [.NET 升級助手](#)，可以幫助您移植舊版。NET 框架應用程序到現代。NET。

成本最佳化建議

若要移轉。NET 框架應用程序，請執行以下操作：

1. 先決條件 — 若要使用 .NET，您必須安裝。NET5+ 在您計劃分析應用程序源代碼的機器上。機器上的資源必須具有至少 1.8 的GHz處理速度、4 GB 的記憶體和 5 Gb 的儲存空間。[如需詳細資訊，請參閱 .NET 文件。](#)
2. 評估 — 下載 .NET 作為 [可執行](#) (下載) 文件。您可以在計算機上下載並安裝該工具，以開始對應用程序進行評估。評估頁面包含移植的專案、套件，且 APIs 與現代版本不相容。NET。基於這個原因，您會在評估之後在解決方案中收到建置錯誤。您可以檢視或下載評估發現項目至 CSV 檔案。[如需詳細資訊，請參閱 移植助理中的移植解決方案。](#) NET 文件。
3. 重構 — 評估應用程式之後，您可以將專案移植到目標架構版本。移植解決方案時，移植助理將修改您的項目文件和某些代碼。您可以檢查日誌以查看源代碼的更改。在大多數情況下，程式碼需要額外的努力才能完成遷移和測試，以使其生產準備就緒。根據應用程序的不同，某些更改可能包括實體框架，身份和身份驗證。[如需詳細資訊，請參閱 移植助理中的移植解決方案。](#) NET 文件。

這是將應用程式現代化為容器的第一步。可能有許多業務和技術驅動程序來實現您的 .NET 架構應用程式到 Linux 容器。其中一個重要的驅動因素是通過從 Windows 操作系統轉移到 Linux 來降低總體擁有成本。如此可降低將應用程式移轉至的跨平台版本時的授權成本。NET 以及容器，以最佳化資源使用率。

將應用程式移植到 Linux 之後，您可以使用將應 [AWS App2Container](#) 應用程式容器化。App2Container 使用 Amazon ECS 或 Amazon EKS 作為端點服務，您可以直接部署到這些服務。App2Container 提供所有必要的基礎結構即程式碼 (IaC) 部署構件，以重複容器化您的應用程式。

其他考量和資源

- 如果您有在 VB 上構建的應用程序。NET (2002 年的遺留框架)，並希望將它們移植到 .NET6、看到 [VB 的傳統港口](#)。[NET 應用程式至 .NET6.0 與移植助理](#)。NET 在 Microsoft 工作負載上的 AWS 博客上發布。
- 如果您在 Windows 通信基礎 (WCF) 上有舊版應用程序，並希望在現代運行它們。NET，您可以採用核心 WCF。[如需詳細資訊，請參閱 WCF 使用移植助理將舊版 WCF 應用程式現代化為核心。](#) [NET](#) 在 Microsoft 工作負載上的 AWS 博客上發布。

- 您可以將移植助理作為擴充功能新增至您的 Visual Studio IDE。這使您可以執行轉換代碼所需的所有任務，而無需您的 IDE 和移植助手之間切換。NET 工具。如需詳細資訊，請參閱[加速。NET 應用程式現代化與移植助理](#)。NETMicrosoft 工作負載 AWS 部落格上的 IDE 擴充功能。
- [AWS 移植助理](#)。NET 現在是開源工具，其中包含評估的源代碼和兼容性分析組件。這可以鼓勵您的開發人員使用和共享。NET 移植知識和最佳實踐。
- 您可以移植。NET 框架應用程式到現代。NET 在 Linux 上，通過使用 AWS 工具包。NET 重構。如需詳細資訊，請參閱[加速。NET 現代化與 AWS 工具包](#)。NETMicrosoft 工作負載部落格上的重構貼文。
AWS
- 您可以[加速。ASP.NET 要使用的核心應 AWS 用程式](#) AWS App2Container。

容器化。NET 应用

概觀

容器是一種輕量且有效率的方式，以一致且可重複的方式封裝及部署應用程式。本節說明如何使用 AWS Fargate 無伺服器容器服務來降低。NET 應用程式同時提供可擴充且可靠的基礎架構

成本影響

影響使用容器節省成本的有效性的一些因素包括應用程式的大小和複雜性、需要部署的應用程式數量，以及應用程式的流量和需求程度。對於小型或簡單的應用程式，與傳統基礎架構方法相比，容器可能無法大幅節省成本，因為管理容器和相關服務的負荷可能實際上會增加成本。但是，對於較大或更複雜的應用程式，使用容器可以透過改善資源使用率並減少所需執行個體的數量來節省成本。

我們建議您在使用容器以節省成本時，請注意以下事項：

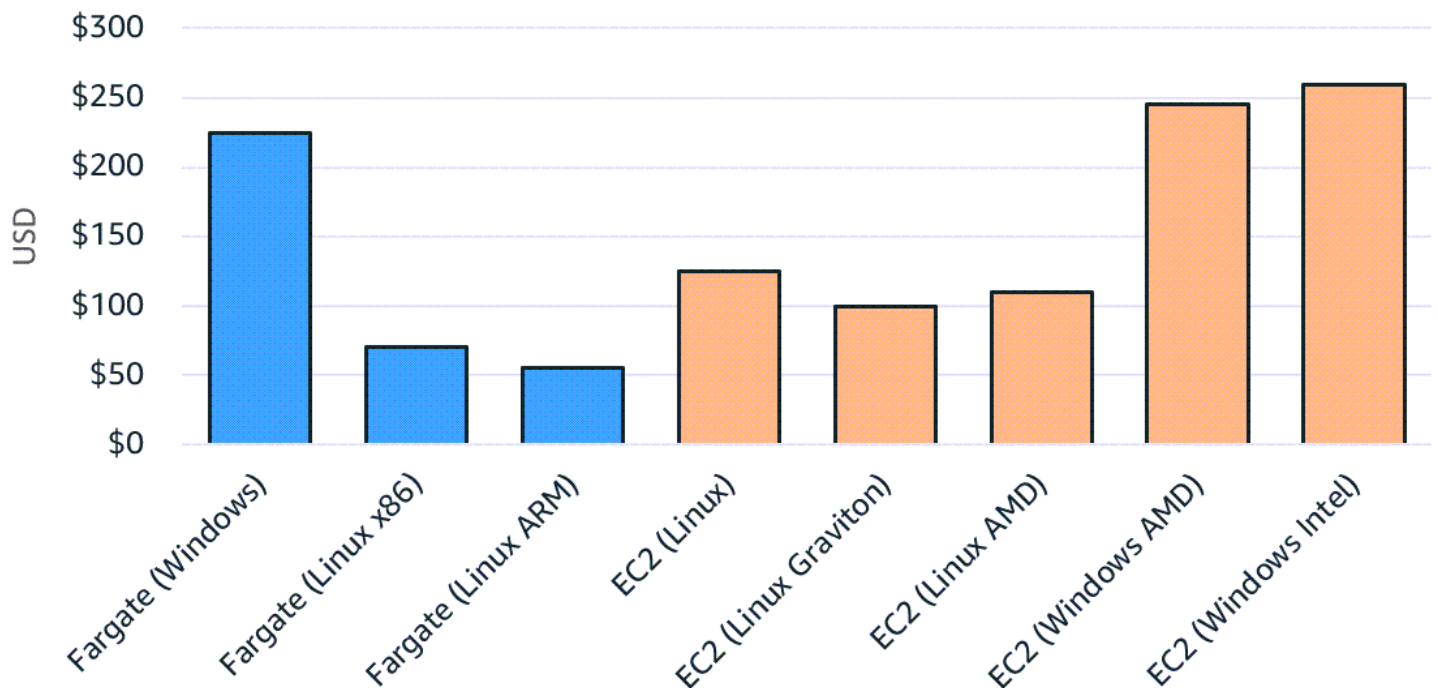
- 應用程式規模與複雜度：較大、更複雜的應用程式更適合容器化，因為它們往往需要更多資源，而且可以從改善的資源使用率中獲益更多。
- 應用程式數量 — 組織必須部署的應用程式越多，透過容器化可節省更多的成本。
- 流量和需求 — 遇到高流量和需求的應用程式可以從容器提供的可擴充性和彈性中獲益。這樣可以節省成本。

不同的架構和作業系統會影響容器成本。如果您使用的是 Windows 容器，則由於授權考量，成本可能不會降低。Linux 容器的授權成本較低或不存在。下表使用美國東部 (俄亥俄) 區域的基本組態，其 AWS Fargate 中包含下列設定：每月 30 個工作每個執行 12 小時，並配置了 4 vCPUs 和 8 GB 的記憶體。

您可以從兩個主要運算平台中進行選擇，以便在其上執行容器 AWS：[EC2基於容器主機和無伺服器](#)或[AWS Fargate](#)。如果您使用 Amazon 彈性容器服務 (AmazonECS) 而非 Fargate，則必須維護執行中的運算 (執行個體)，以允許放置引擎在需要時實例化容器。如果您改用 Fargate，則只會佈建所需的運算容量。

下圖顯示了使用 Fargate 與 Amazon EC2 的等效容器的差異。由於 Fargate 具有靈活性，應用程式的任務每天可以執行 12 小時，在下班時間內使用率為零。但是，對於 AmazonECS，您必須使用 [Auto Scaling EC2 執行個體群組](#) 來控制運算容量。這可能會導致每天 24 小時執行容量，最終可能會增加成本。

Monthly costs of Fargate and Amazon EC2



成本最佳化建議

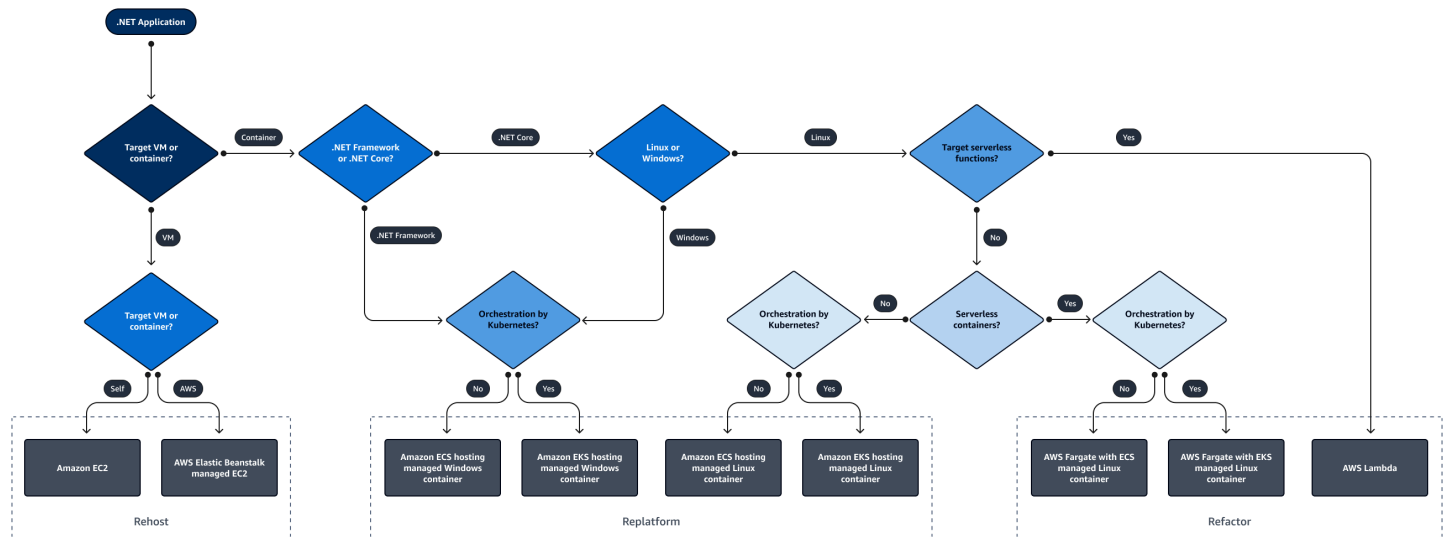
使用 Linux 容器而不是視窗

如果您使用 Linux 容器而不是 Windows 容器，則可以大幅節省成本。例如，如果您執行 .NET 在 Linux 上的核心，而不是執行 .NET 在 Windows 上的框架。如果您使用 ARM 架構 (AWS 重力子) 而不是 x86，則可以額外節省 40% 的費用。

如果您打算針對現有的 .NET 架構應用程式時，您必須將這些應用程式移植到 .NET (例如 .NET 6.0)，以便使用 Linux 容器。一個主要的考慮因素是權衡重構的成本與通過降低 Linux 容器成本所節省的成本相比。如需將應用程式移植到現代軟體的詳細資訊。NET，請參閱 [.NET](#) 在文 AWS 檔中。

轉移到現代的另一個好處。NET (也就是說，遠離 .NET 框架) 是額外的現代化機會可用。例如，您可以考慮將應用程式重新架構為更具擴充性、敏捷性和成本效益的微服務架構。

下圖說明探索現代化機會的決策程序。



善加利用 Savings Plans

容器可以幫助您利用 [Compute Savings Plan](#) s 來降低 Fargate 成本。彈性 discount 模式提供與可轉換預留執行個體相同的折扣。Fargate 定價是根據您開始下載容器映像到 Amazon ECS 任務終止 (四捨五入到最接近的秒數) 開始使用的 v CPU 和記憶體資源計算。[Fargate 的 Savings Plan](#) s 最多可節省 50% 的 Fargate 用量，以換取承諾在一年或三年期間使用特定計算用量 (以每小時美元計算)。您可以使用 [AWS Cost Explorer](#) 來幫助您選擇 Savings Plan。

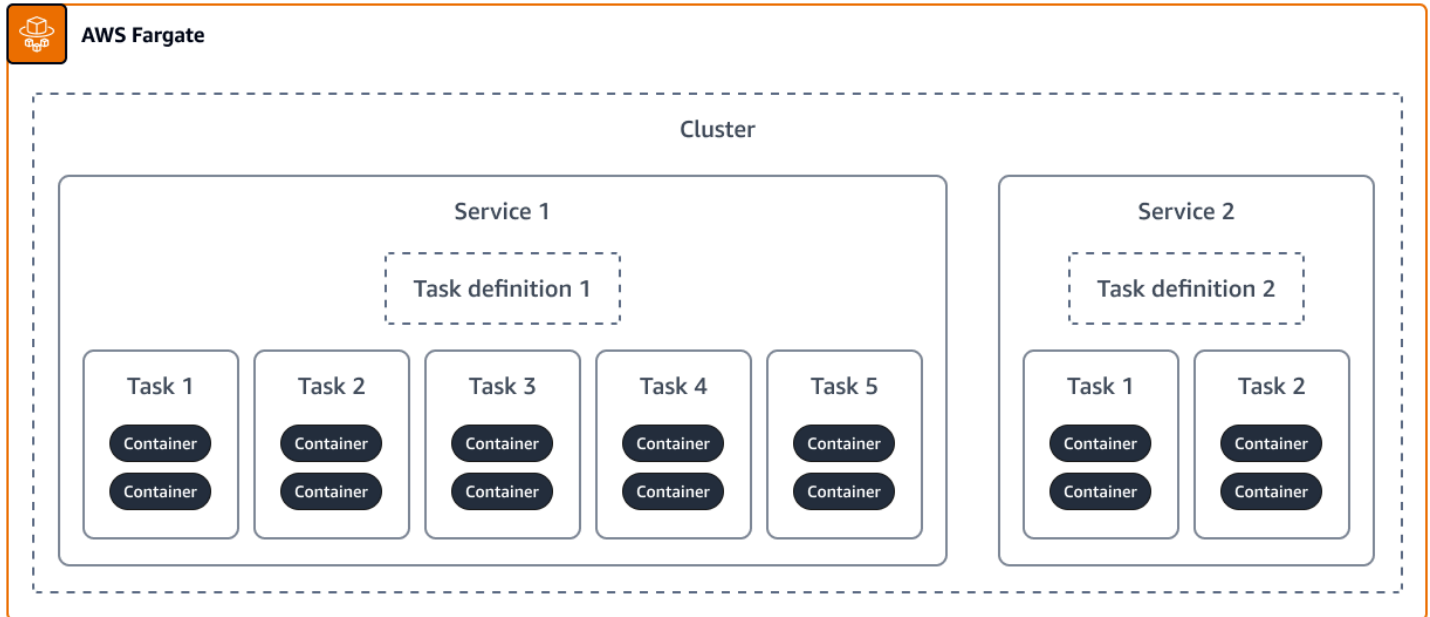
重要的是要了解 Compute Savings Plans 適用於首先為您節省最大的成本的用量。例如，如果您在中執行 t3.medium Linux 執行個體，而 us-east-2 且執行相同的 Windows t3.medium 執行個體，則 Linux 執行個體會先獲得 Savings Plan 權益。這是因為 Linux 執行個體有 50% 的節省潛力，而相同的 Windows 執行個體則有 35% 的節省潛力。如果您的 Savings Plan 中運行了其他符合條件的資源 AWS 帳戶，例如 Amazon EC2 或 Lambda，那麼您的 Savings Plan 不必先應用於 Fargate。有關更多信息，請參閱 [節省計劃文檔中的了解 Savings Plans 如何應 AWS 用於您的用量以及本指南的 EC2 「優化 Windows 在 Amazon 上的支出」](#) 一節。

正確的大小 Fargate 任務

確保 Fargate 任務的大小正確，以實現最大程度的成本優化是非常重要的。通常，開發人員在最初確定其應用程式中使用的 Fargate 任務的配置時，沒有所有必要的使用信息。這可能會導致任務過度佈建，然後導致不必要的支出。為了避免這種情況，我們建議您加載在 Fargate 上運行的測試應用程式，以了

解特定任務配置如何在不同的使用情況下執行。您可以使用負載測試結果，vCPU，任務的內存分配和 auto 擴展策略來找到性能和成本之間的正確平衡。

下圖顯示 Compute Optimizer 工具如何針對最佳工作和容器大小產生建議。



一種方法是使用負載測試工具，例如[分佈式負載測試中描述](#)的工具 AWS，以建立 v CPU 和內存使用率的基準。執行負載測試以模擬一般應用程式負載之後，您可以微調工作的 v CPU 和記憶體組態，直到達到基準使用率為止。

其他資源

- [Amazon ECS 和 AWS Fargate\(AWS 容器部落格文章\) 的成本最佳化檢查清單](#)
- [Amazon ECS 啟動類型的理論成本優化：Fargate vs EC2 \(AWS 容器博客文章 \)](#)
- [移植助理.NET \(AWS 文件 \)](#)
- [分散式負載測試 AWS \(AWS 解決方案程式庫\)](#)
- [AWS Compute Optimizer 啟動對 Amazon ECS 服務的支持 AWS Fargate \(AWS 雲財務管理博客文章 \)](#)

使用重力彈執行個體和容器

概觀

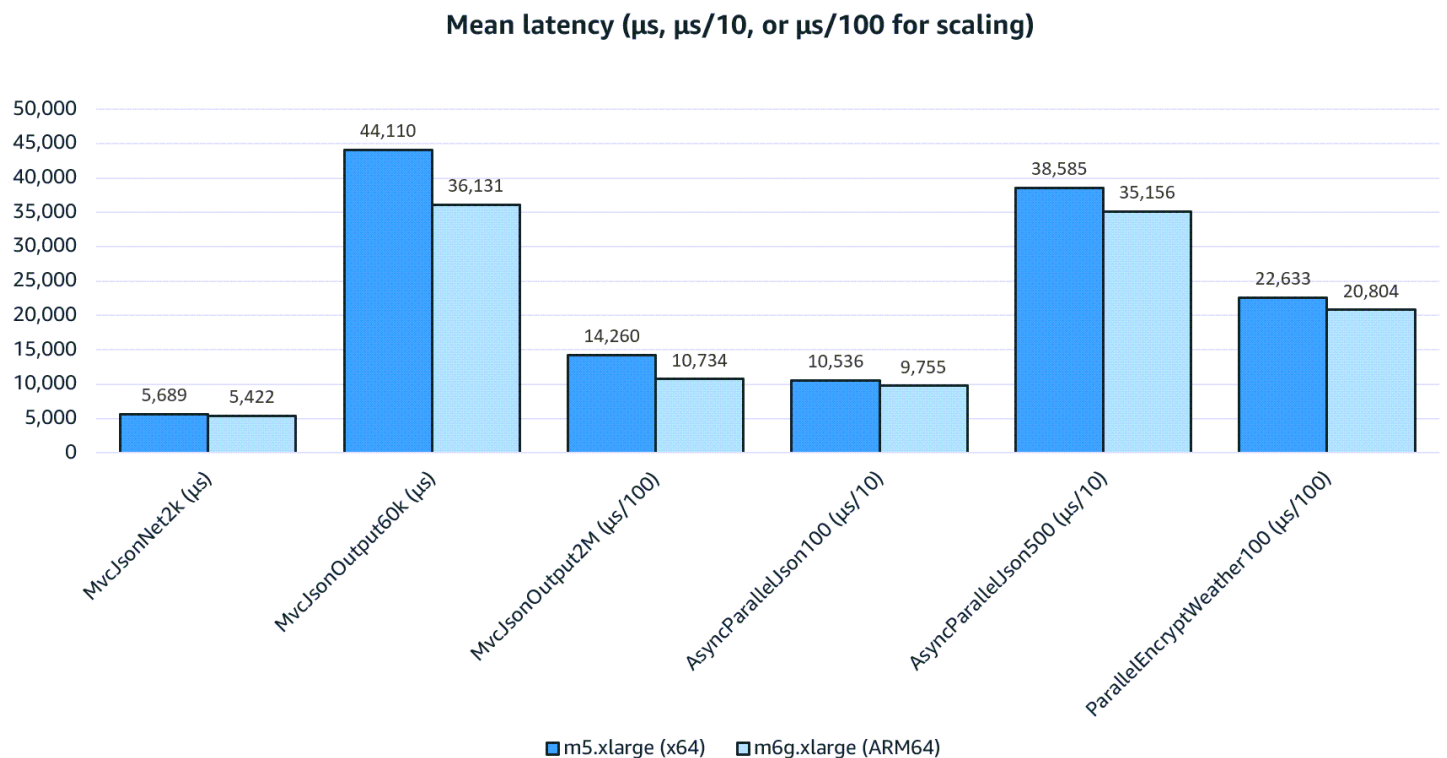
AWS Graviton 執行個體由 AWS 專為在 Amazon 彈性運算雲端 (Amazon) 中執行的雲端工作負載 (包括執行中的容器 EC2) 提供最佳價格效能的 ARM 處理器提供支援。AWS 目前有三代引力子可在 Amazon 上使用。EC2 本指南重點介紹了重力子 2 和 3 的使用。NET 應用程式是因為當您使用最新版本的 Graviton 時，可大幅節省成本。請記住，重力子執行個體只能執行 Linux 作業系統。因此，重力子執行個體是一個強大的產品。NET 在 Linux 上運行，但不是 Windows 操作系統或舊版的選項。NET 框架應用程序。

與同級 EC2 執行個體相比，Graviton 3 的效率提高了 60%，效能提升高達 40%。本指南著重於使用 Graviton 的成本效益，但請務必注意，Graviton 可提供效能改善和改善環境永續性的額外優勢。

成本影響

當您切換到引力子時，您最多可節省 45% 的成本。在您重構任何舊版之後。NET 框架應用程序到現代 .NET 版本中，您可以解鎖使用引力子實例的能力。移動到引力子是一種有效的成本優化技術。NET 開發人員。

下表中的範例顯示了透過移轉至 Graviton 實例可達到效能改善的可能性。



如需上圖中用來建立結果之基準測試方法的完整細分和說明，請參閱[供電。NET5 與 AWS 重力 2：運算部落格中的基準](#)。AWS

提高效率的原因之一是 x86 和引力子CPU之間 v 的含義差異。在 x86 架構中，v CPU 是超執行緒所達成的邏輯核心。在 Graviton 中，v CPU 等同於實體核心，可讓 v 完全投CPU入工作負載。

與同類的 x86/x64 執行個體相比，引力 2 的價格效能提高了 40%。引力蛋 3 提供以下超過重力 2 的功能：

- 提升效能設定檔，效能提升 25%
- 浮點效能提升高達兩倍
- 加密工作負載效能提升高達兩倍
- 機器學習效能提升高達三倍

此外，Graviton3 是雲端中第一個具有記憶體功能的執行個體。DDR5

下表顯示重力型執行個體與同等 x86 型執行個體之間節省成本的差異。

此表格顯示 19.20% 的重力子節省成本。

執行個體類型	架構	v CPU	記憶體 (GB)	每小時成本 (按需求)
t4g.xlarge	ARM	4	16	0.1344
t3.xlarge	x86	4	16	0.1664

此表格顯示 14.99% 的重力子節省成本。

執行個體類型	架構	v CPU	記憶體 (GB)	每小時成本 (按需求)
c7g.4xlarge	ARM	16	32	\$0.5781
c6i.4xlarge	x86	16	32	0.6800

考慮引力子時，請務必測試應用程式的效能設定檔。重力子不是可靠的軟件開發實踐的替代品。您可以使用測試來驗證您是否充分利用基礎運算資源。

成本最佳化建議

有幾種方法可以利用重力子處理器/執行個體。本節將引導您完成從使用 x86 架構機器移至 Graviton () 執行個體所需的變更。ARM

變更 Lambda 中的執行階段設定

建議您在中切換執行階段設定 AWS Lambda。如需詳細資訊，請參閱 Lambda 文件中的[修改執行階段環境](#)。從那時起 NET 是一種編譯語言，您必須遵循構建過程才能使其工作。如需如何執行此作業的範例，請參閱。[NET 在引力子上](#)。GitHub

容器

對於容器化工作負載，請建立多架構容器映像。您可以通過在 Docker 構建命令中指定多個體系結構來做到這一點。例如：

```
docker buildx build -t "myImageName:latest" --platform linux/amd64,linux/arm64 --push .
```

您也可以使用工具，例 AWS Cloud Development Kit (AWS CDK) 如[協助協調組建](#)。如需 Docker 的範例，請參閱 Docker 說明文件中的[使用 Docker 桌面為 Arm 和 x86 建置多重架構映像](#)。

Amazon EC2

若要 ARM 從 x86/x64 遷移至，請在編譯步驟中指定 ARM 架構的目標。在視覺工作室中，您可以創建一個 ARM64 CPU。如需指示，請參閱 [Microsoft 文件中的若要設定以 Arm64 和其他平台為目標的專案](#)。

如果您使用的是 .NET CLI，然後在 ARM 機器上運行構建產生與 Graviton 兼容的構建。若要觀看示範，請觀看「[加速](#)」。NET 使用 Arm64 在重力動 AWS 力 2 開啟時進行 6 次性能。YouTube 依賴性問題將導致編譯時錯誤，然後可以單獨解決。只要有任何依賴關係的 ARM 庫，轉換應該相對簡單。

其他資源

- [如何使用 Amazon 上的引力 ARM 和 Spot 執行個體建置容器並節省成本 ECS \(AWS 部落格\)](#)
- [AWS Lambda 由 AWS Graviton2 處理器驅動的功能 — 在手臂上運行您的功能並獲得高達 34% 的更佳價格性能 \(博客 \) AWS](#)
- [將 AWS Lambda 函數移轉至 ARM 式 AWS 重力 on2 處理器 \(部落格 \) AWS](#)

- [建置和部署。NETWeb 應用程式到 AWS 引ARM力子 2 Amazon ECS 群集使用 AWS CDK \(博客 \) AWS](#)
- [引力子快速啟動 — 協助您將工作負載轉移至 AWS 引力子的新計畫 \(部落格\)AWS](#)
- [供電。NET5 與 AWS 重力 2 : 基準 \(博客 \) AWS](#)

Support 靜態的動態縮放。 NET框架應用

概觀

將雲端用於應用程式的其中一個主要優點是彈性，或是能夠根據需求擴展或擴展運算。這可讓您只為所需的運算容量付費，而不需要針對尖峰使用量進行佈建。網絡星期一，在線零售商可以快速獲得比正常更多的流量（例如，[幾分鐘內數千個百分比](#)），這是一個很好的彈性例子。

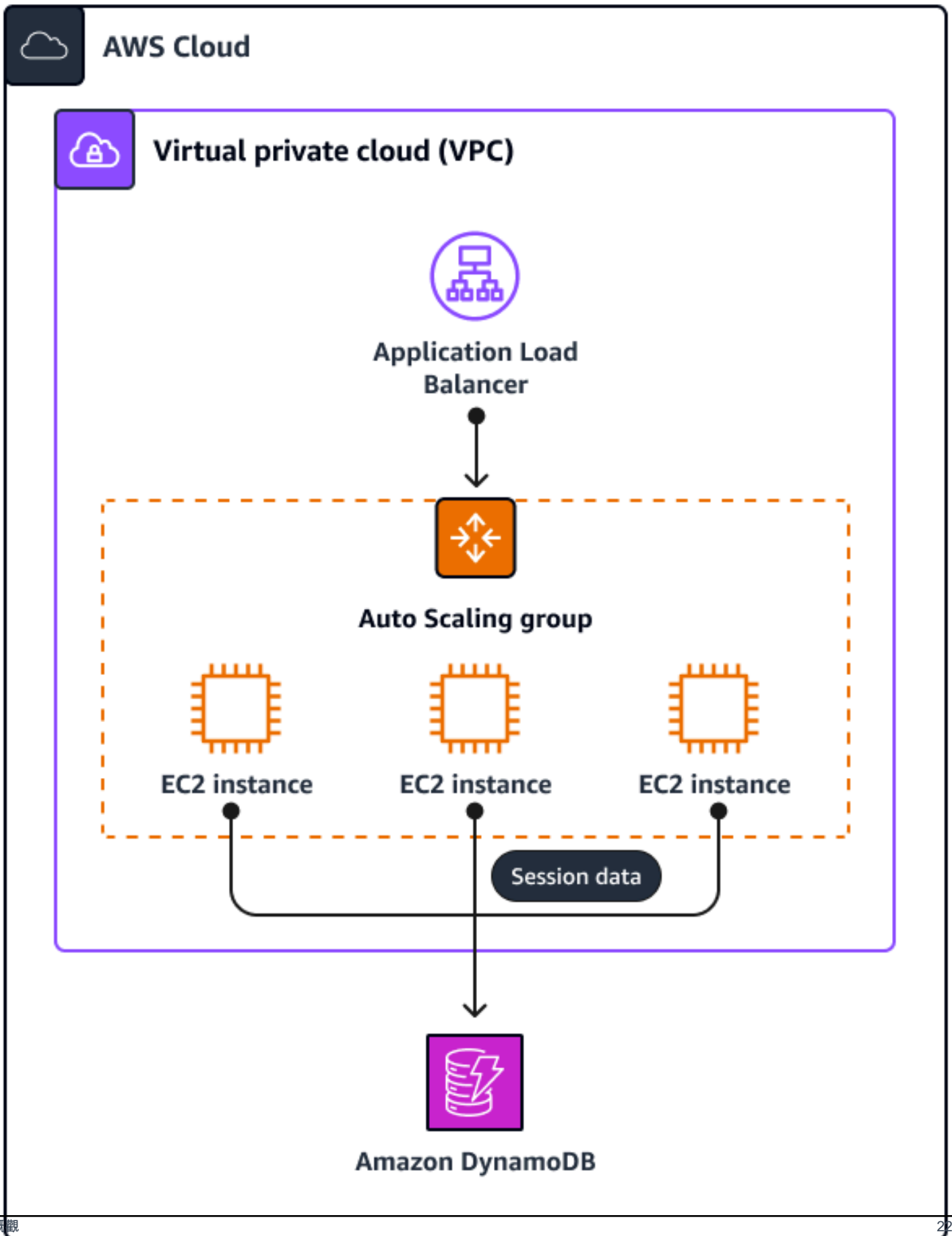
如果你帶來的遺產。NETWeb 應用程式到雲端（例如，ASP.NET在上執行的架構應用程式IIS），由於應用程式的可設定狀態性質，快速擴充負載平衡伺服器陣列的能力可能很困難或不可能。用戶會話數據存儲在應用程式的內存中，通常與[ASP.NET會話狀態](#)或靜態變量保存必須保存的交叉請求數據。使用者工作階段相依性通常是透過負載平衡器黏滯工作階

事實證明，這在操作上具有挑戰性。當需要增加容量時，您必須刻意佈建和新增伺服器。這可能是一個緩慢的過程。如果發生修補或未預期的失敗，將節點停止服務，可能會造成使用者體驗的問題，而導致與受影響節點相關聯的所有使用者失去狀態。充其量，這將需要用戶再次登錄。

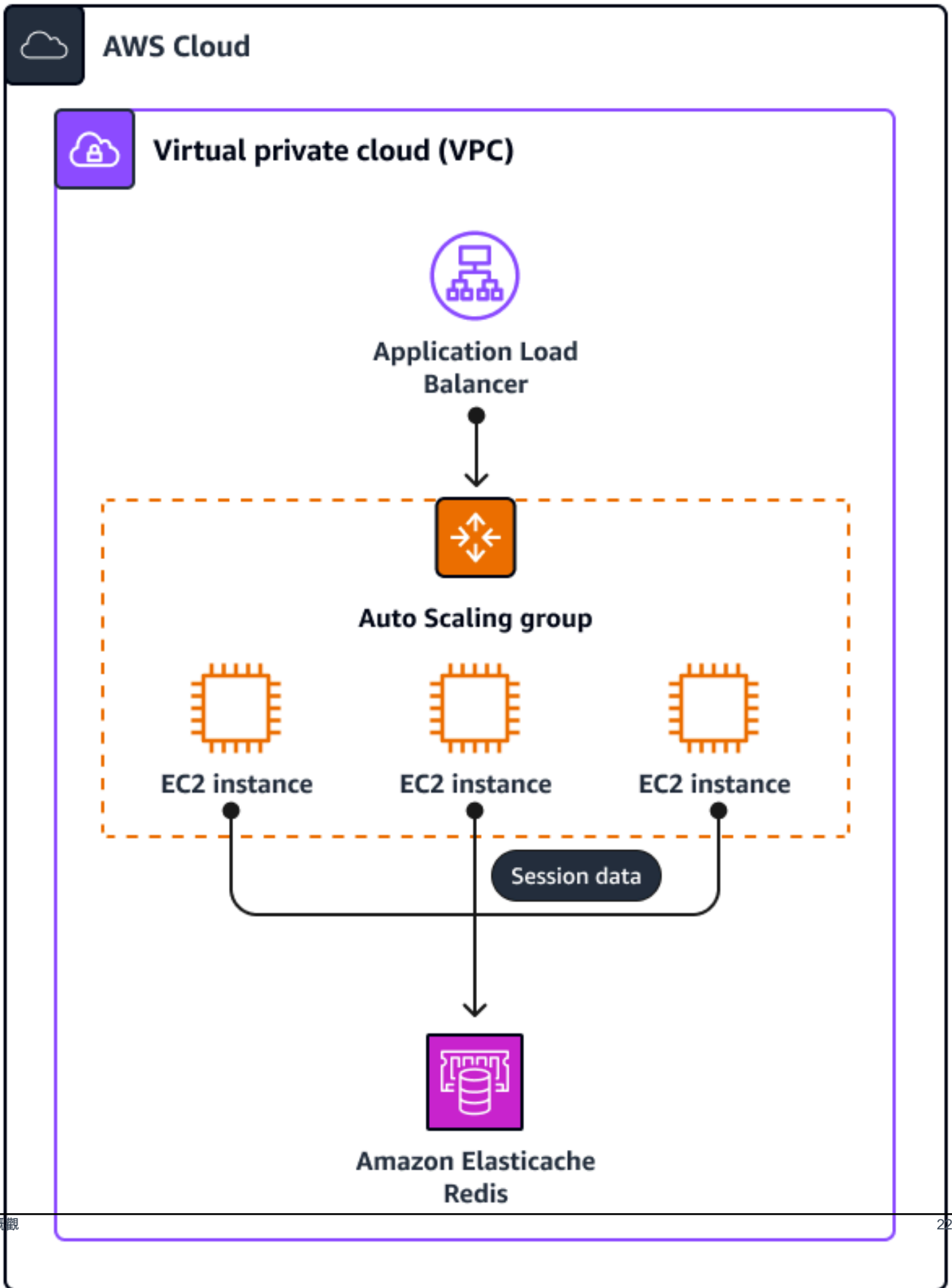
透過集中化的ASP工作階段狀態。NET應用程式並將自動調度資源規則套用至舊版 ASP.NET應用程式，您可以利用雲端的彈性，並可能在執行應用程式時利用節省成本的優勢。例如，您可以透過運算擴展性降低成本，但也可以從可用的不同定價模式中進行選擇，例如減少[預留執行個體使用量](#)和使用[Amazon Spot 執行個體定價](#)。

兩種常見技術包括使用[Amazon DynamoDB 做為工作階段狀態提供者](#)，以及使用[Amazon ElastiCache \(RedisOSS\) 做為 ASP.NET會話存儲](#)。

下圖顯示使用 DynamoDB 做為工作階段狀態提供者的架構。



下圖顯示使用 ElastiCache (RedisOSS) 作為工作階段狀態提供者的架構。



成本影響

若要判斷生產應用程式擴充的優點，建議您設定實際需求的模型。本節提供下列假設來建模範例應用程式：

- 從旋轉中加入和移除的例證是相同的，並且不引入任何例證大小變化。
- 為了維持應用程式的高可用性，伺服器使用率永遠不會低於兩個作用中伺服器。
- 伺服器數量會隨流量線性擴展 (也就是說，流量需要兩倍的運算量)。
- 流量在一個月內以六小時為增量進行建模，當日變化和一個異常的流量峰值 (例如，促銷銷售)，一天的 10 倍流量。週末流量是根據基本使用率建模。
- 夜間流量是根據基本使用率建模，而工作日流量則以 4 倍的使用率建模。
- 預留執行個體定價採用一年的無預付定價。正常的日間定價使用隨需定價，而突發需求則使用競價型執行個體

下圖說明了該模型如何利用彈性的優勢。NET 應用程式而非針對尖峰使用量佈建。這樣可以節省約 68%。

Comparison of cumulative costs for peak provisioning and autoscaling



如果您使用 DynamoDB 做為工作階段狀態儲存機制，請使用下列參數：

```
Storage: 20GB
Session Reads: 40 million
Session Writes: 20 million
```

```
Pricing Model: On demand
```

這項服務的估計每月費用約為每月 35.00 美元。

如果您使用 ElastiCache (RedisOSS) 作為工作階段狀態儲存機制，請使用下列參數：

```
Number of Nodes: 3
Node size: cache.t4g.medium
Pricing Model: 1y reserved
```

這項服務的估計每月費用約為每月 91.00 美元。

成本最佳化建議

第一步是在傳統中實現會話狀態。NET 應用程式。如果您使用 ElastiCache 作為狀態儲存機制，請遵循 AWS SDK for .NET 文件中「[什麼是](#)」[AWS SDK for .NET](#) 中的指引。[如果您使用的是 DynamoDB](#)，請按照 [ElastiCache ASP NET AWS 開發者工具部落格中的工作階段存放區](#)

如果應用程式使用 InProc 工作階段開始，請確定您打算儲存在工作階段中的所有物件都可以序列化。若要這麼做，請使用 `SerializableAttribute` 屬性來裝飾其實例將儲存在工作階段中的類別。例如：

```
[Serializable()]
public class TestSimpleObject {
    public string SessionProperty {get;set;}
}
```

此外，`.NETMachineKey` 所有使用中的伺服器之間必須相同。通常情況下，從一般 Amazon 機器映像檔 (AMI) 建立執行個體時。例如：

```
<machineKey
validationKey="some long hashed value"
decryptionKey="another long hashed value"
validation="SHA1"/>
```

但是，確保基本映像如果更改，則將其配置為相同的非常重要。NET 機器映像（可在 IIS 或伺服器級別配置）。如需詳細資訊，請參閱 [SystemWebSectionGroup. MachineKey Microsoft 文檔中的屬性](#)。

最後，您必須決定將伺服器新增至 Auto Scaling 群組以回應擴展事件的機制。有幾種方法可以完成此操作。我們建議您使用以下方法進行無縫部署。NET「Auto Scaling」群組中 EC2 執行個體的架構應用程式：

- 使用 [EC2Image Builder](#) 來設定包AMI含完全設定的伺服器 and 應用程式的。然後，您可以使用它AMI 來配置 [Auto Scaling 組的啟動模板](#)。
- 用 [AWS CodeDeploy](#) 於部署您的應用程式。CodeDeploy 可直接與 [Amazon EC2 Auto Scaling](#) 合。這提供了為應用程序的每個版本創建一個新AMI的替代方法。

其他資源

- [使用 EC2 Image Builder 創建圖像](#) (EC2Image Builder 文檔)
- [部署。NETWeb 應用程式 AWS CodeDeploy 搭配 Visual Studio 小組服務使用](#) (AWS 開發人員工具 部落格)

使用快取來減少資料庫需求

概觀

您可以使用快取作為有效的策略，以協助您降低.NET應用程式。當應用程式需要頻繁存取資料時，許多應用程式會使用後端資料庫，例如SQL伺服器。維護這些後端服務以應付需求的成本可能很高，但是您可以使用有效的快取策略，藉由減少大小和擴展需求來減少後端資料庫的負載。這可協助您降低成本並改善應用程式的效能。

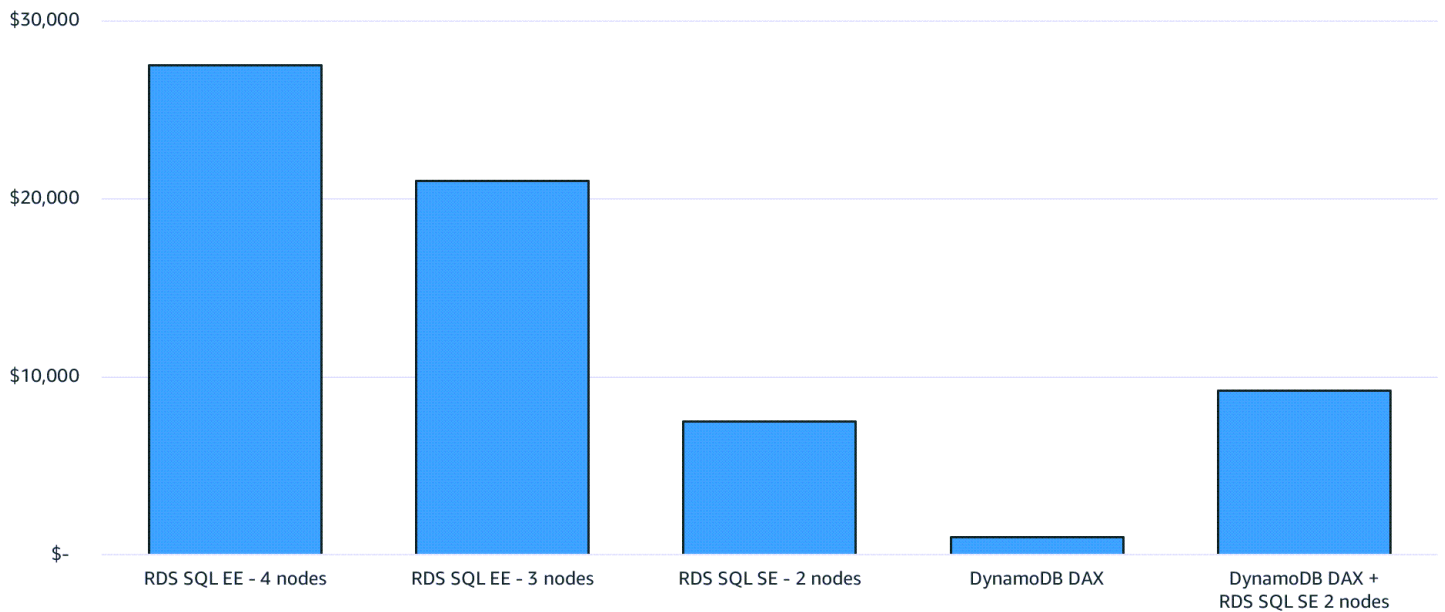
快取是一種非常有用的技術，可以節省與使用較昂貴資源 (例如SQL伺服器) 的讀取繁重工作負載相關的成本。為您的工作負載使用正確的技術非常重要。例如，本地緩存不可擴展，並要求您為應用程序的每個實例維護本地緩存。您應該權衡與潛在成本相比的效能影響，以便較低的基礎資料來源成本抵銷與快取機制相關的任何額外成本。

成本影響

SQL服務器要求您在調整數據庫大小時考慮讀取請求。這可能會影響成本，因為您可能需要引入僅供讀取複本以適應負載。如果您使用僅供讀取複本，請務必瞭解它們僅適用於SQL伺服器企業版。此版本需要比SQL伺服器標準版更昂貴的授權。

下圖旨在幫助您了解緩存有效性。它顯示了 Amazon SQL 服務RDS器與四個 db.m4.2xlarge 節點運行服務器企業版。SQL它部署在異地同步備份配置中，具有一個僅供讀取複本。獨佔讀取流量 (例如，SELECT查詢) 會導向至僅供讀取複本。相較之下，Amazon DynamoDB 使用兩個節點動態 B 加速器 () 叢集。DAX

下圖顯示不需要處理高讀取流量的專用僅供讀取複本的結果。



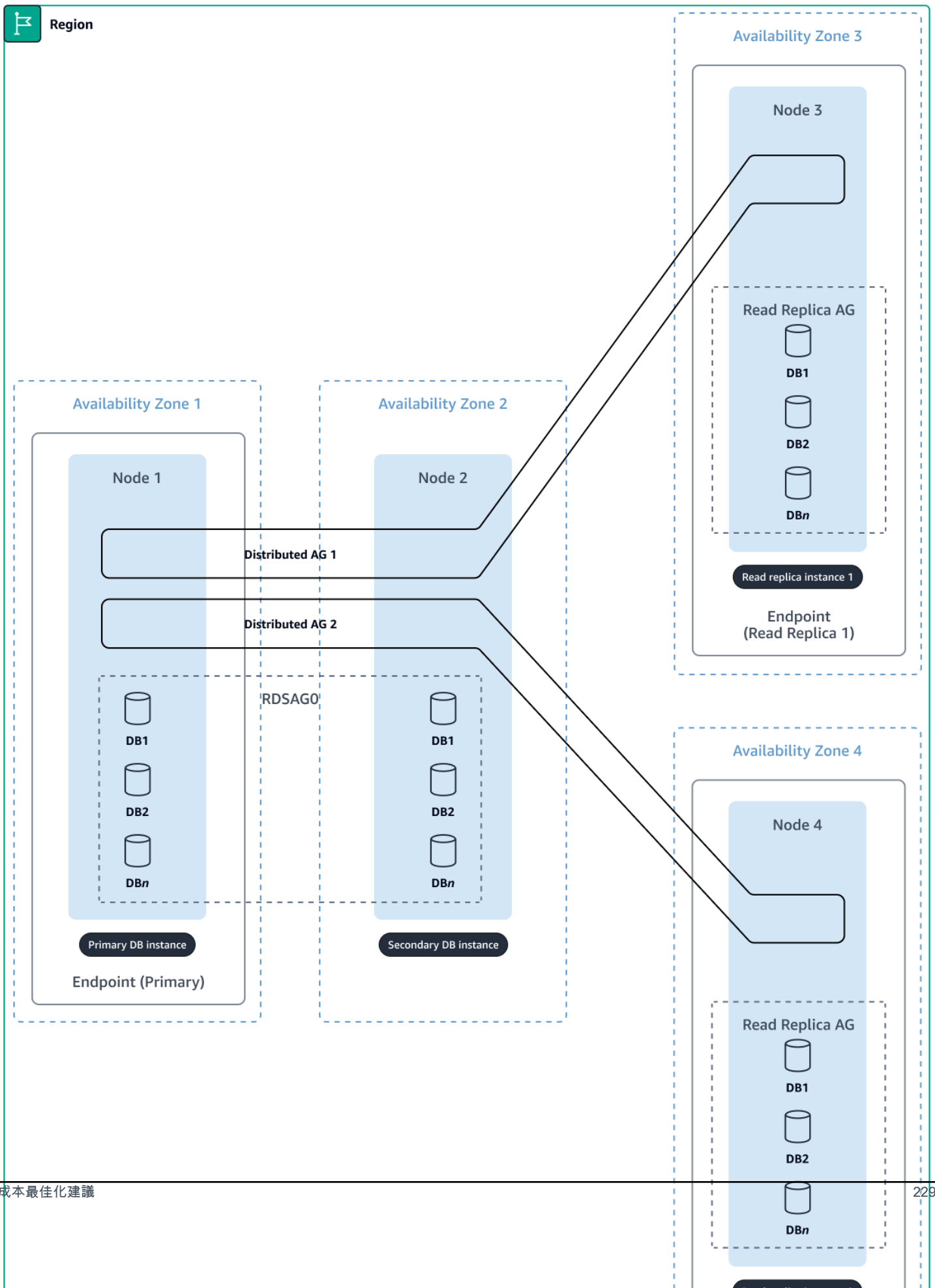
您可以使用無僅供讀取複本的本機快取，或將 Amazon 上的 SQL Server 作為快取層並 DAX 排引入，RDS 以節省大量成本。此層會從 SQL 伺服器卸載，並減少執行資料庫所需的 SQL 伺服器大小。

成本最佳化建議

本機快取

本機快取是快取內部部署環境或雲端中託管之應用程式內容的最常用方法之一。這是因為它相對容易和直觀的實施。本地緩存涉及從資料庫或其他源獲取內容，並在內存或磁盤上本地緩存以便更快地訪問。這種方法雖然易於實現，但對於某些用例來說並不理想。例如，這包括快取內容需要隨時間持續存在的使用案例，例如保留應用程式狀態或使用者狀態。另一個使用案例是需要從其他應用程式執行個體存取快取內容時。

下圖說明具有四個節點和兩個僅供讀取複本的高可用性 SQL 伺服器叢集。



使用本機快取時，您可能需要負載平衡多個EC2執行個體之間的流量。每個執行個體都必須維護自己的本機快取。如果緩存存儲有狀態信息，則需要對數據庫進行定期提交，並且用戶可能需要為每個後續請求（粘性會話）轉發到相同的實例。在嘗試擴展應用程式時，這是一個挑戰，因為某些執行個體可能會過度使用，而有些執行個體則因為流量分配不均勻而未充分利用。

您可以將本機快取（記憶體內或使用本機儲存）用於 .NET 應用程式。為此，您可以添加功能以將對象存儲在磁盤上並在需要時進行檢索，或者從數據庫查詢數據並將其保存在內存中。若要在 C# 中執行本機快取記憶體內和來自 SQL 伺服器的資料本機儲存，例如，您可以使用 MemoryCache 和 LiteDB 程式庫的組合。MemoryCache 提供記憶體內快取，而 LiteDB 是快速且輕量級的嵌入式無 SQL 磁碟型資料庫。

若要執行記憶體內快取，請使用 .NET 圖書館 System.Runtime.MemoryCache。下列程式碼範例會示範如何使用 System.Runtime.Caching.MemoryCache 類別在記憶體中快取資料。這個類提供了一種將數據暫時存儲在應用程序的內存中的方法。這可減少從較昂貴的資源（例如資料庫或）擷取資料的需求，藉此協助改善應用程式的效能 API。

以下是代碼的工作原理：

1. 已建立 MemoryCache named 的私有靜態執行 `_memoryCache` 個體。該緩存被賦予一個名稱（`dataCache`）來標識它。然後，緩存存儲和檢索數據。
2. 該 `GetData` 方法是一個通用方法，它有兩個參數：一個 `string` 鍵和一個調用的 `Func<T>` 委託 `getData`。該密鑰用於標識緩存的數據，而 `getData` 委託表示當數據不存在於緩存中時被執行的數據檢索邏輯。
3. 該方法首先檢查數據是否存在於使用該 `_memoryCache.Contains(key)` 方法的緩存中。如果資料位於快取中，則方法會使用擷取資料，`_memoryCache.Get(key)` 並將其轉換為預期的類型 `T`。
4. 如果數據不在緩存中，則該方法會調用 `getData` 委託以獲取數據。然後，它通過使用將數據添加到緩存中 `_memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10))`。此呼叫指定快取項目應在 10 分鐘後過期，此時資料會自動從快取中移除。
5. 該 `ClearCache` 方法需要一個 `string` 鍵作為參數，並通過使用從緩存中刪除與該鍵相關聯的數據 `_memoryCache.Remove(key)`。

```
using System;
using System.Runtime.Caching;

public class InMemoryCache
{
    private static MemoryCache _memoryCache = new MemoryCache("dataCache");
```



```
public static T GetData<T>(string key, Func<T> getData)
{
    if (_memoryCache.Contains(key))
    {
        return (T)_memoryCache.Get(key);
    }

    T data = getData();
    _memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10));

    return data;
}

public static void ClearCache(string key)
{
    _memoryCache.Remove(key);
}
}
```

您可以使用以下代碼：

```
public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";

        Func<string> getSampleData = () =>
        {
            // Replace this with your data retrieval logic
            return "Sample data";
        };

        string data = InMemoryCache.GetData(cacheKey, getSampleData);
        Console.WriteLine("Data: " + data);
    }
}
```

下面的例子演示了如何使用 [LiteDB](#) 在本地存儲緩存數據。您可以使用 LiteDB 作為記憶體內快取的替代方案或補充。下面的代碼演示了如何使用 LiteDB 庫在本地存儲緩存數據。該 `LocalStorageCache` 類包含用於管理緩存的主要功能。

```
using System;
```

```
using LiteDB;

public class LocalStorageCache
{
    private static string _liteDbPath = @"Filename=LocalCache.db";

    public static T GetData<T>(string key, Func<T> getData)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            var item = collection.FindOne(Query.EQ("_id", key));

            if (item != null)
            {
                return item;
            }
        }

        T data = getData();

        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            collection.Upsert(new BsonValue(key), data);
        }

        return data;
    }

    public static void ClearCache(string key)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection("cache");
            collection.Delete(key);
        }
    }
}

public class Program
{
    public static void Main()
    {
```

```
string cacheKey = "sample_data";

Func<string> getSampleData = () =>
{
    // Replace this with your data retrieval logic
    return "Sample data";
};

string data = LocalStorageCache.GetData(cacheKey, getSampleData);
Console.WriteLine("Data: " + data);
}
}
```

如果您的靜態快取或靜態檔案不會經常變更，您也可以將這些檔案存放在 Amazon Simple Storage Service (Amazon S3) 物件儲存中。應用程式可以在啟動時擷取靜態快取檔案，以便在本機使用。如需有關如 Amazon S3 用 .NET，請參閱 [Amazon S3 文件中的下載物件](#)。

使用緩存 DAX

您可以使用可跨所有應用程式執行個體共用的快取層。[DynamoDB 加速器 \(DAX\)](#) 是適用於 DynamoDB 的全受管、高可用性的記憶體內快取，可提供十倍的效能改善。您可以使 DAX 用減少 DynamoDB 表格中過度佈建讀取容量單位的需求來降低成本。這對於讀取過重且需要重複讀取個別金鑰的工作負載特別有用。

DynamoDB 是按需定價或已佈建容量，因此每月的讀取和寫入數量會造成成本。如果您讀取過重的工作負載，DAX 叢集可減少 DynamoDB 表格上的讀取次數來協助降低成本。如需有關如何設定的指示 DAX，請參閱 DynamoDB 文件中的[使用 DynamoDB 加速器進行記憶體內加速 \(DAX\)](#)。如需相關資訊。NET 應用程式整合，觀看[將 Amazon DynamoDB 整合 DAX 到您的 .ASP NET 應用程式](#)於 YouTube。

其他資源

- [使用 DynamoDB 加速器進行記憶體內加速 \(DAX\)-Amazon DynamoDB 動態 B 文件](#)
- [將 Amazon DynamoDB 整合 DAX 到您的 .ASP NET 應用程式](#) (YouTube)
- [下載物件](#) (Amazon S3 文件)

考慮使用無伺服器。NET

概觀

無伺服器運算已成為建置和部署應用程式的熱門方法。這主要是因為無伺服器方法在建置現代化架構時提供的可擴充性和敏捷性。不過，在某些情況下，請務必考慮無伺服器運算的成本影響。

Lambda 是無伺服器運算平台，可讓開發人員執程式碼，而不需要專用伺服器。Lambda 是一個特別有吸引力的選擇。NET 希望降低基礎架構成本的開發人員 隨著 Lambda，.NET 開發人員可以開發和部署可高度擴展性和潛在成本效益的應用程式 透過使用無伺服器方法，開發人員不再佈建伺服器來處理應用程式要求。相反，開發人員可以創建按需執行的功能。這使得無伺服器方法比執行、管理和調整虛擬機器更具可擴充性、可管理性，且可能更具成本效益。因此，您只需為應用程式使用的資源付費，而不必擔心資源使用率不足或伺服器維護成本。

開發人員可以使用現代的跨平台。NET 建置快速、高效且符合成本效益的無伺服器應用程式的版本。的 .NET 核心和較新版本是一個免費的開源框架，比以前更適合在無服務器平台上運行。NET 框架版本。這可讓開發人員縮短開發時間並提高應用程式效能。現代。NET 還支持一系列編程語言，包括 C# 和 F#。因此，對於希望在雲中構建現代架構的開發人員來說，這是一個有吸引力的選擇。

本節說明如何使用 Lambda 作為無伺服器選項來節省成本。您可以透過微調 Lambda 函數的執行設定檔、使用 [原生調整](#) Lambda 函數的記憶體分配大小，以及移至以重力為基礎的函數 AOT，進一步最佳化成本。

成本影響

您可以降低多少成本取決於幾個因素，包括除了分配的記憶體數量和每個函數的持續時間之外，您的無伺服器函數還將執行多少次執行。AWS Lambda 提供免費方案，其中包括每月一百萬個免費請求和每月 400,000 GB 的運算時間。在這些免費方案限制範圍內或附近的工作負載，您可以大幅降低每月的成本。

使用以 Lambda 函數做為目標的負載平衡器時，也可能會產生額外費用。計算方式為負載平衡器針對 [Lambda 目標](#) 處理的資料量。

成本最佳化建議

正確調整您的 Lambda 函數

正確的大小是在成本優化的基本做法。NET 基於 Lambda 函數。此程序包括識別最佳記憶體組態，以平衡效能與成本效益，而不需要變更程式碼。

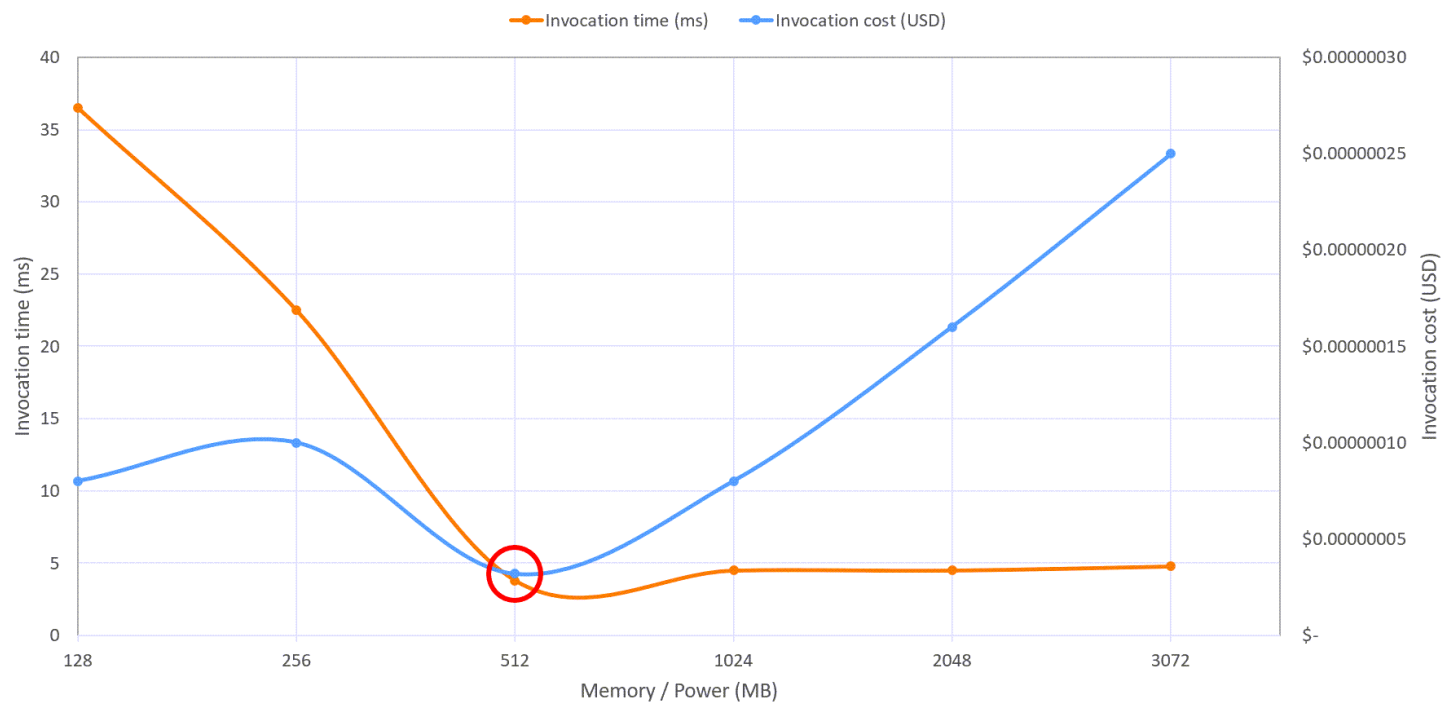
藉由設定 Lambda 函數的記憶體，範圍介於 128 MB 到 10,240 MB 之間，您也可以調整呼叫期間 CPU 可用的 v 數量。這可讓記憶體或 CPU-bound 應用程式在執行期間存取額外的資源，進而可能減少叫用持續時間和整體成本。

但是，請確定您的 .NET 以 Lambda 為基礎的函數可能是手動且耗時的程序，特別是在頻繁變更時。

「[AWS Lambda 電源調整](#)」工具可以根據範例承載分析一組記憶體組態，協助您識別適當的組態。

例如，增加一個 .NET 以 Lambda 為基礎的函數可以改善總叫用時間並降低成本，而不會影響效能。功能的最佳記憶體配置可能會有所不同。AWS Lambda 電源調整工具可以幫助識別每個功能的最具成本效益的配置。

在下列範例圖表中，總叫用時間會隨著此 Lambda 函數的記憶體增加而改善。這會導致在不影響函數的原始性能的情況下降低總執行成本。對於此函數，函數的最佳記憶體配置為 512 MB，因為這是資源使用率對於每次呼叫的總成本而言最有效率的地方。這因函數而異，使用 Lambda 函數上的工具可以識別它們是否從正確的大小中受益。



我們建議您定期完成此練習，作為發行新更新時任何整合測試的一部分。如果不常更新，請定期執行此練習，以確保功能的調整和正確的大小。為 Lambda 函數識別適當的記憶體設定後，您可以為程序新增正確的大小。「AWS Lambda 電源調整」工具會產生程式設計輸出，可供 CI/CD 工作流程在新程式碼發行期間使用。這使您可以自動化內存配置。

您可以免費下載[AWS Lambda 電源調整工具](#)。如需有關如何使用工具的指示，請參閱中的[如何執行狀態機](#) GitHub。

Lambda 也支援原生功能AOT，這使 .NET 要預編譯的應用程式。這可以通過減少執行時間來幫助降低成本。NET 函數。如需建立原生AOT函數的詳細資訊，請參閱 [NET在 Lambda 文件中包含原生AOT編譯的函數](#)。

避免閒置等待時間

Lambda 函數持續時間是用於計算帳單的一個維度。當函數代碼進行阻塞調用時，系統會按照等待接收響應的時間向您收取費用。當 Lambda 函數鏈結在一起，或者函數充當其他函數的協調器時，此等待時間可能會增加。如果您有批次作業或訂單交付系統等工作流程，這會增加管理額外負荷。此外，可能無法在最長 15 分鐘的 Lambda 逾時內完成所有工作流程邏輯和錯誤處理。

而不是在函數程式碼中處理這個邏輯，我們建議您重新架構您的解決方案，以 [AWS Step Functions](#) 作為工作流程的協調器使用。使用標準工作流程時，您需要支付工作流程中的每個狀態轉換費用，而不是工作流程的總持續時間。此外，您可以將重試、等待條件、錯誤工作流程和 [回呼](#) 的支援移至狀態條件，讓 Lambda 函數專注於商業邏輯。如需詳細資訊，請參閱 AWS 運算 [部落格中的最佳化 AWS Lambda 成本第 2 部分](#)。

移至以重力為基礎的函數

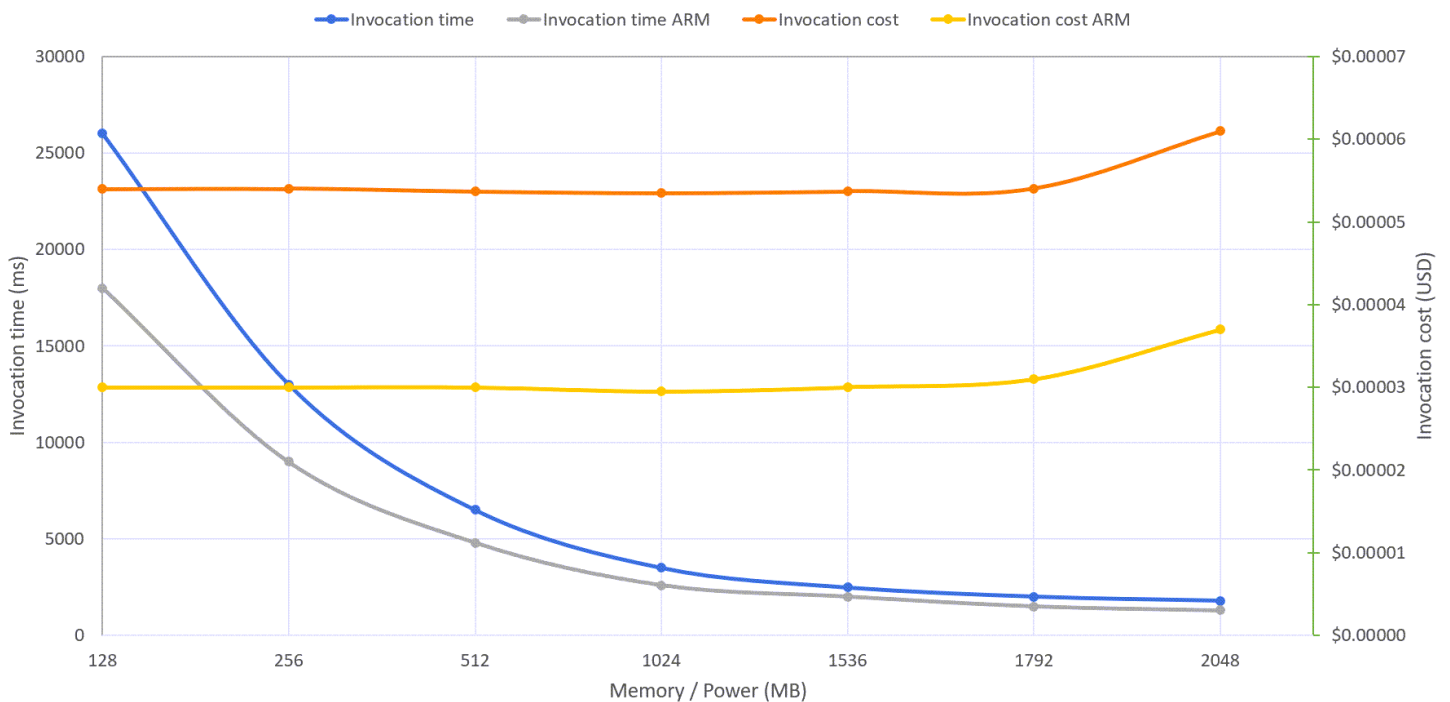
由下一代重力 on2 處理器提供支援的 Lambda 函數現已正式推出。使用處理器架構的 Graviton2 功能旨在為各種無伺服器工作負載降低 20% 的成本提供高達 19% 的效能。ARM 由 Graviton2 處理器驅動的功能具有更低的延遲和更好的效能，是為關鍵任務無伺服器應用程式提供動力的理想選擇。

移轉至以重力為基礎的 Lambda 函數可能是符合成本效益的選擇。NET 希望將 Lambda 成本最佳化的開發人員。以重力為基礎的函數使用基礎的處理器，而不是傳統的 x86 處理器。這樣可以大幅節省成本，而不會犧牲效能。

雖然轉移到以重力為基礎的函數有許多好處，但我們建議您考慮一些挑戰和考量。例如，基於重力的函數需要使用 Amazon Linux 2，這可能與所有功能都不兼容。NET 應用程式。此外，協力廠商 ARM 式庫或相依性可能存在與處理器不相容的相容性問題。

如果您正在執行 .NET 架構應用程式並希望利用 Lambda 的無伺服器優勢，您可以考慮將應用程式移植到現代化 .NET。NET 使用 [移植助理](#)。NET。這可以幫助您加速傳統的移植。NET 應用到現代 .NET，讓應用程式能夠在 Linux 上執行。

下圖比較 x86 和 ARM /重力 on2 架構結果，以獲得計算質數的函數。



該函數使用單個線程。當記憶體設定為 1.8 GB 時，會報告兩種架構的最短持續時間。在此之上，Lambda 函數可以訪問超過 1 vCPU，但在這種情況下，該函數無法使用額外的功率。出於同樣的原因，成本是穩定的，內存高達 1.8 GB。記憶體越多，因為這個工作負載沒有額外的效能優勢，所以成本就會增加。Graviton2 處理器顯然能為這項運算密集型功能提供更好的效能與更低的成本。

若要將您的功能設定為搭配 Graviton 使用和 ARM 基於處理器的功能，請執行下列動作：

1. 登入，然 AWS Management Console 後開啟 [Lambda 主控台](#)。
2. 選擇建立函數。
3. 針對 Function name (函數名稱)，輸入名稱。
4. 針對執行階段，選擇。NET6 (C #/ PowerShell)。
5. 對於「架構」，請選取 arm64。
6. 進行您需要的任何其他設定，然後選擇 [建立功能]。

其他資源

- [作為目標的 Lambda 函數](#) (AWS 文件)
- [使用 AWS Compute Optimizer\(AWS 運算部落格\) 將 AWS Lambda 成本和效能最佳化](#)
- [最佳化您的 AWS Lambda 成本 — 第 1 部分](#) (AWS 計算部落格)
- [最佳化您的 AWS Lambda 成本 — 第 2 部分](#) (AWS 計算部落格)

- [建置無伺服器。NET 應用程式 AWS Lambda 使用 .NET7 \(AWS 計算部落格\)](#)

考慮專門建置的資料庫

概觀

執行以 Microsoft 為基礎的工作負載最昂貴的方面之一來自商業資料庫 (例如伺服器) 的授權。SQL 企業通常在 SQL Server 上進行標準化，作為首選的數據庫平台，並且在組織的發展文化中根深蒂固。無論使用案例為何，開發人員通常都會選擇關聯 SQL 式伺服器型號。原因包括：

- 該企業已經擁有可用的 SQL 伺服器實例和/或授權。
- 團隊通過使用共享庫和業務邏輯習慣了 SQL 編程模型。ORMs
- 管理層是不知道的替代方案。
- 開發人員不知道的替代方案。

依據特定目的建置的資料庫可配合您使用案例的資料存取模式。隨著企業採用更現代化的架構 (例如微服務) 以及個別應用程式的範圍狹窄，因此越來越多地採用這些資料庫。

專門建置的資料庫不會排除關聯式模型，也不需要 No SQL (非關聯式) 模型。事實上，為了回應工作負載的特定需求而選取關聯式資料庫，就會被視為是專門建置的。使用專門構建的數據庫可以幫助團隊降低與他們相關的數據庫成本。NET 應用程式，同時也獲得標準的雲端優勢，例如可擴充性、彈性，以及減少無差別繁重的工作。

下表顯示提供的專用資料庫。AWS

資料庫	Type	特性
Amazon Aurora 郵政 SQL 或 Amazon Aurora 我 SQL	关系	資料具有固定結構的使用案例 關聯式資料庫自然地透過 ACID 交易維護資料
Amazon DynamoDB	鍵值對	沒有使用哈希表數據結構存儲數據的數據 SQL 庫 高效能儲存與擷取非結構化資料

資料庫	Type	特性
		使用案例包括使用者設定檔、工作階段狀態和購物車資料
Amazon ElastiCache	記憶體內	<p>高性能沒有將非結構化數據存儲在存取時間低於一毫秒的內存中的數據SQL庫</p> <p>用於經常存取的暫時資料 (例如使用者工作階段), 以及在其他較慢的資料存放區前面做為快取層</p> <p>包括對兩個 ElastiCache (Redis 的OSS) 和 ElastiCache (內存緩存) 的支持</p>
Amazon 記憶數據庫	耐用的記憶體	Redis 相容的專用資料庫, 具有耐用的儲存裝置
Amazon Timestream	時間序列	<p>專為按暫時順序擷取高輸送量資料而設計的資料庫</p> <p>使用案例包括物聯網 (IoT) 應用程式, 以及儲存指標或遙測資料</p>
Amazon DocumentDB	文件	<p>沒有存儲沒有規定結構的數據或與其他數據強制關係的數據SQL庫</p> <p>通常用於讀取密集型工作負載, 例如產品目錄</p>
Amazon Neptune	圖表	<p>沒有數據SQL庫同時保存數據和數據項之間的連接的表示</p> <p>使用案例包括詐騙偵測、推薦引擎和社交應用程式</p>

資料庫	Type	特性
Amazon Quantum 賬本數據庫 (AmazonQLDB)	分類帳	<p>儲存交易資料，並提供每筆交易的密碼驗證，提供可稽核、不可變的歷史記錄</p> <p>當需要單一可驗證的事實來源時，應用程式通常會使用</p>
Amazon Keyspaces	寬柱	<p>基於阿帕奇卡桑德拉的高性能分佈式數據庫</p> <p>使用案例包括 IoT 應用程式、事件處理和遊戲應用程式</p>

採用專用資料庫的重要驅動因素可歸因於消除商業授權。不過，[DynamoDB](#) (包括[隨需模式](#))、[Aurora](#)、[Amazon Neptune](#) 和 [Amazon Keyspaces](#) 等資料庫的 auto-scaling 功能可讓您針對一般情況佈建容量，而非尖峰使用量。專門建置的資料庫 (例如 Timestream 或 Amazon) 是無伺服器的 QLDB，可自動擴充以符合需求，無需任何預先佈建。

AWS SQL 如果您想要使用專門建置的[開放原始碼相容關聯式資料庫](#)，但無法或不願意對您的應用程式進行重大程式碼變更，則會提供 [Babelfish to Postgre](#)。在某些情況下，Babelfish 允許您使用現有的 SQL 服務器訪問代碼，幾乎沒有任何更改。

為應用程式選擇專門建置的關聯式資料庫時，請務必保留應用程式所需的相同 (或功能等同) 功能。此建議將特定用途建置的資料庫解決為應用程式的主要資料存放區。特定應用程式 (例如快取) 會在其他建議中加以解決。

成本影響

採用專門建置的資料庫。NET 工作負載雖然不太可能直接影響運算消耗/成本，但可能會直接影響 NET 應用程式。事實上，與敏捷性、可擴充性、彈性和資料耐久性等額外優勢相比，節省成本可能是次要目標。

不在本指南的範圍內，說明如何為應用程式選擇專用資料庫，並重新架構資料策略以有效使用這些資料庫的完整程序。如需詳細資訊，請參閱 AWS 教學課程 [目錄中的特定目的建置資料庫](#)。

下表顯示數個範例，說明以特定目的建置的資料庫取代 SQL Server 如何改變應用程式成本。請注意，這些只是粗略的估計。需要對實際工作負載進行基準測試和優化，才能計算確切的生產成本。

這些是一些常用的專用資料庫估計值SSD，其中包括隨需運算和 100 GB 的單一執行個體資料庫。us-east-1授權成本包括SQL伺服器授權和軟體保證。

下表顯示商業資料庫範例的估計成本。

資料庫引擎	授權模式	執行個體類型/規格	AWS 運算 + 儲存成本	授權費用	每月總成本
SQLAmazon 服務器標準版 EC2	包括許可證	大容量 (8/64 GB) CPU RAM	1,345.36 美元	\$0.00	1,345.36 美元
SQLAmazon 服務器企業版 EC2	包括許可證	大容量 (8/64 GB) CPU RAM	\$2,834.56	\$0.00	\$2,834.56
SQLAmazon 服務器標準版 EC2	BYOL	大容量 (8/64 GB) CPU RAM	644.56 美元	456.00 美元	1,100.56 元
SQLAmazon 服務器企業版 EC2	BYOL	大容量 (8/64 GB) CPU RAM	644.56 美元	\$1,750.00	\$2,394.56
SQLAmazon 服務器標準版 RDS		資料庫 .r6i.2 大型 (8/64 GB) CPU RAM	2,318.30 美元	\$0.00	2,318.30 美元
SQLAmazon 服務器企業版 RDS		資料庫 .r6i.2 大型 (8/64 GB) CPU RAM	3,750.56 元	\$0.00	3,750.56 元

下表顯示針對特定目的建置範例的估計成本。

資料庫引擎	執行個體類型/規格	AWS 運算 + 儲存成本	授權費用	每月總成本
Amazon Aurora 郵政 SQL	大容量 (8 /64 GB) CPU RAM	855.87 美元	\$0.00	855.87 美元
DynamoDB	已佈建的WCU基礎 RCU	72.00 美元		72.00 美元
Amazon DocumentDB	資料庫 .r6i.2 大型 (8/64 GB) CPU RAM	778.60 美元		778.60 美元
Amazon QLDB	80 米寫 四百萬閱讀 2 TB 日誌 六百 GB 指數	320.00 美元		320.00 美元

Important

此表格是根據購買前三年內含軟體保證之SQL伺服器的預估授權成本。對於SQL伺服器標準版：\$4,100 美元，2 個核心包，3 年。對於SQL伺服器企業版：15,700 美元，2 個核心包，3 年。

我們建議您在採用專門建置的資料庫之前，先考慮成本影響。例如，更新應用程式以使用專用資料庫的成本與應用程式和來源資料庫的複雜性有關。規劃此架構交換器時，請務必考慮總體擁有成本。這包括重構您的應用程式、提升人員對新技術的技術，以及仔細規劃每個工作負載的預期效能和耗用量。從那裡，你可以做出決定，如果投資是值得的成本節約。在大多數情況下，維護 end-of-support 產品是一種安全性和合規性風險，而修復產品的成本是值得的努力和初始投資。

成本最佳化建議

對於 .NET 正在訪問 SQL 伺服器的應用程式，有針對特定目的構建的關係數據庫的替換庫。您可以在應用程式中實作這些程式庫，以取代類似的 SQL Server 應用程式功能。

下表強調了一些可在許多常見案例中使用的程式庫。

程式庫	資料庫	更換	框架兼容性
Npgsql 實體框架核心提供者	Amazon Aurora 郵政 SQL	實體框架核心SQL服務器提供	現代。NET
Npgsql 實體框架 6 提供者	Amazon Aurora 郵政 SQL	實體框架 6.0 SQL 服務器提供者	。NET 框架
Npgsql (.ADO NET 兼容後SQL庫)	Amazon Aurora 郵政 SQL	ADO.NET	。NET 框架/現代。NET
我的SQL實體框架核心提供者	亞馬遜我的極光 SQL	實體框架核心SQL服務器提供	現代。NET
柚子。EntityFrameworkCore。MySQL	亞馬遜我的極光 SQL	實體框架核心SQL服務器提供	現代。NET

[SQL通過使用巴貝爾魚連接到 Amazon Aurora Postgre](#) 不需要任何特殊的編碼即可連接。但是，所有代碼都應在使用之前進行徹底測試。

其他專門建置的資料庫都有可供存取的程式庫。NET 兼容的庫，使您能夠訪問目的構建的數據庫。範例包括：

- [使用 Amazon DynamoDB 無SQL資料庫](#) (說明文件)AWS SDK for .NET
- [C# 驅動程序](#) (MongoDB 文檔)
- [。NET](#) (時間流文檔)
- [Amazon QLDB 驅動程序.NET](#) (Amazon QLDB 文檔)
- [使用卡桑德拉。NET核心客戶端驅動程序以編程方式訪問 Amazon Keyspaces](#) (Amazon Keyspaces 文檔)
- [使用.NET連接到 Neptune 資料庫執行個體](#) (Neptune 文件)

如果您移轉至目的內建資料庫，您可以使用下列工具 AWS 來協助進行移轉程序：

- [AWS Schema Conversion Tool \(AWS SCT\)](#) 可協助您將SQL伺服器結構描述轉換為 Amazon Aurora 和 Amazon DynamoDB。

- [AWS Database Migration Service \(AWS DMS\)](#) 可協助您一次或持續將資料從SQL伺服器移轉至 Aurora 或 DynamoDB。
- [巴貝爾魚指南針](#)可以幫助您檢查您的SQL服務器數據庫的兼容性，以便與巴貝爾魚 Aurora Postgre 一起使用。SQL

其他資源

- [將SQL伺服器遷移到 Amazon Aurora 郵政指南 SQL \(AWS 資料庫部落格\)](#)
- [。NET現代化工作AWS 坊 \(工作室\)](#)
- [巴貝爾魚APP現代化沉浸日 \(工作坊工作室 \) AWS](#)
- [。NET沉浸式日 \(AWS 工作坊工作室\)](#)
- [開始使用 Amazon Timestream 與 .NET \(GitHub\)](#)
- [Amazon QLDB 驅動程序. NET— 快速入門教程 \(Amazon QLDB 文檔 \)](#)
- [專為現代化設計的專用資料庫。NET應用程式 AWS \(AWS 簡報\)](#)

後續步驟

完成本指南的檢閱後，我們建議您採取下列步驟來實作 MACO：

1. 與 MACO 專家聯繫。MACO 專家可以幫助回答您的問題並解決您的疑慮。如果您已經與 AWS 客戶團隊合作，請聯絡團隊並向 MACO 專家尋求協助。如果您沒有客戶團隊，請[連絡 `optimize-microsoft@amazon.com`](mailto:optimize-microsoft@amazon.com)。
2. 套用建議。應用您在本指南中學到的建議，最佳實踐和策略，並與 MACO 專家交談。
3. 追蹤成本變更。標記您的工作負載，並使用諸如以 AWS Cost Explorer 及 AWS Budgets 進行詳細的成本追蹤、監控和控制等服務。

文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

變更	描述	日期
SQL 伺服器 and 容器更新	我們新增了 [使用運算最佳化處理程式來最佳化 SQL Server 大小] 、 [檢閱 SQL Server 工作負載的 Trusted Advisor 建議] 和 [重新平台 Windows 應用程式與 App2 Container] 區段。	2024年6月29 日
SQL 伺服器授權最佳化	我們新增了 使用計算最佳化處理器最佳化 SQL Server 授權 區段。	2024年5月22 日
初次出版	—	2023 年 12 月 21 日

AWS 規範指南詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式常用詞彙。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

數字

7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的內部部署 Oracle 資料庫遷移至 Amazon Aurora Postgre SQL-Compatible Edition。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將您的內部部署 Oracle 資料庫遷移至 中的 Oracle 的 Amazon Relational Database Service (Amazon RDS) AWS 雲端。
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 (CRM) 系統遷移至 Salesforce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將內部部署 Oracle 資料庫遷移至 中 EC2 執行個體上的 Oracle AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移至相同平台的雲端服務。範例：遷移 Microsoft Hyper-V 應用程式至 AWS。
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

A

ABAC

請參閱 [屬性型存取控制](#)。

抽象服務

請參閱 [受管服務](#)。

ACID

請參閱 [原子、一致性、隔離、持久性](#)。

主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但比 [主動被動遷移](#) 需要更多工作。

主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。目標資料庫在遷移期間不接受任何交易。

彙總函數

在一組資料列上操作並計算該群組單一傳回值的 SQL 函數。彙總函數的範例包括 SUM 和 MAX。

AI

請參閱 [人工智慧](#)。

AIOps

請參閱 [人工智慧操作](#)。

匿名化

在資料集中永久刪除個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

反模式

經常用於重複性問題的解決方案，其解決方案具有反效益、無效或效果不如替代方案。

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體侵害。

應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是 [產品組合探索和分析程序](#) 的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱[什麼是人工智慧？](#)

人工智慧操作 (AIOps)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需如何在遷移策略AIOps中使用 AWS 的詳細資訊，請參閱[操作整合指南](#)。

非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

原子、一致性、隔離、持久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱 AWS Identity and Access Management (IAM) 文件[ABAC AWS](#)中的。

權威性資料來源

您存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、編輯或假名化資料。

可用區域

中與其他可用區域中的故障 AWS 區域 隔離的不同位置，並對相同區域中的其他可用區域提供低成本、低延遲的網路連線。

AWS 雲端採用架構 (AWS CAF)

來自的指導方針和最佳實務架構 AWS，可協助組織制定高效且有效的計劃，以成功地遷移至雲端。AWS CAF 將指導方針整理成六個重點領域：業務、人員、治理、平台、安全和操作。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。為此，AWS CAF 提供人員開發、訓練和通訊的指引，協助組織準備好成功採用雲端。如需詳細資訊，請參閱[AWS CAF網站](#)和[AWS CAF白皮書](#)。

AWS 工作負載資格架構 (AWS WQF)

評估資料庫遷移工作負載、建議遷移策略並提供工作估算的工具。AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

B

錯誤的機器人

旨在中斷或傷害個人或組織的[機器人](#)。

BCP

請參閱[業務連續性規劃](#)。

行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以搭配 Amazon Detective 使用行為圖表來檢查失敗的登入嘗試、可疑API呼叫和類似的動作。如需詳細資訊，請參閱偵測文件中的[行為圖中的資料](#)。

大端序系統

首先儲存最高有效位元組的系統。另請參閱[永久性](#)。

二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 (藍色) 中執行目前的應用程式版本，並在另一個環境 (綠色) 中執行新的應用程式版本。此策略可協助您在影響最小的情況下快速復原。

機器人

軟體應用程式，透過網際網路執行自動化任務，並模擬人類活動或互動。某些機器人很有用或有益，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。某些其他稱為壞機器人的機器人，旨在中斷或傷害個人或組織。

殭屍網路

受到[惡意軟體](#)感染且由單一方控制的[機器人](#)網路，稱為機器人繼承器或機器人運算子。Botnet 是擴展機器人及其影響的最佳已知機制。

分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱[關於分支](#)（GitHub 文件）。

碎片存取

在特殊情況下，以及透過核准的程序，使用者取得其通常無權存取 AWS 帳戶之存取權的快速方法。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS [實作碎片程序](#) 指標。

棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和[綠地](#)策略。

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

業務如何創造價值 (例如，銷售、客戶服務或營銷)。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 [AWS 上執行容器化微服務](#) 白皮書的[圍繞業務能力進行組織](#) 部分。

業務連續性規劃 (BCP)

一種解決破壞性事件 (如大規模遷移) 對營運的潛在影響並使業務能夠快速恢復營運的計畫。

C

CAF

請參閱[AWS 雲端採用架構](#)。

Canary 部署

版本向最終使用者緩慢且增量的版本。當您有信心時，您可以部署新版本並完全取代目前的版本。

CCoE

請參閱 [Cloud Center of Excellence](#)。

CDC

請參閱 [變更資料擷取](#)。

變更資料擷取 (CDC)

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以使用 CDC 進行各種用途，例如稽核或複寫目標系統中的變更，以維持同步。

混亂工程

故意引入故障或破壞性事件，以測試系統的復原能力。您可以使用 [AWS Fault Injection Service \(AWS FIS \)](#) 執行實驗，以強調 AWS 工作負載並評估其回應。

CI/CD

請參閱 [持續整合和持續交付](#)。

分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

用戶端加密

在目標 AWS 服務接收資料之前，在本機加密資料。

Cloud Center of Excellence (CCoE)

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端企業策略部落格上的 [CCoE文章](#)。

雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲端運算通常連接到 [邊緣運算](#) 技術。

雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱 [建置您的雲端操作模型](#)。

採用雲端階段

組織在遷移至時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展您的雲端採用（例如，建立登陸區域、定義 CCoE、建立操作模型）
- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段由 Stephen Orban 在企業 AWS 雲端 策略部落格的 [The Journey Toward Cloud-First 和採用階段](#) 部落格中定義。如需有關它們如何與 AWS 遷移策略關聯的資訊，請參閱 [遷移準備指南](#)。

CMDB

請參閱 [組態管理資料庫](#)。

程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲端儲存庫包括 GitHub 或 AWS CodeCommit。程式碼的每個版本都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

冷資料

很少存取且通常為歷史資料的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且成本較低的儲存層或類別，可以降低成本。

電腦視覺 (CV)

使用機器學習從數位映像和影片等視覺化格式分析和擷取資訊的 [AI](#) 欄位。例如，AWS Panorama 提供將 CV 新增至內部部署攝影機網路的裝置，而 Amazon 則 SageMaker 提供 CV 的影像處理演算法。

組態偏離

對於工作負載，組態會從預期狀態變更。這可能會導致工作負載變得不合規，而且通常是漸進和無意的。

組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常會在遷移 CMDB 的產品組合探索和分析階段使用來自的資料。

一致性套件

您可以組合的 AWS Config 規則和修復動作集合，以自訂合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶和區域中或整個組織中的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的[一致性套件](#)。

持續整合和持續交付 (CI/CD)

自動化軟體發程序的來源、建置、測試、暫存和生產階段的程序。CI/CD 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。如需詳細資訊，請參閱[持續交付的優點](#)。CD 也可表示持續部署。如需詳細資訊，請參閱[持續交付與持續部署](#)。

CV

請參閱[電腦視覺效果](#)。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected Framework 中安全支柱的元件。如需詳細資訊，請參閱[資料分類](#)。

資料漂移

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變化。資料偏離可以降低 ML 模型預測的整體品質、準確性和公平性。

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

架構架構架構，提供分散式、分散式資料擁有權，並具有集中式管理與治理。

資料最小化

僅收集和處理嚴格必要資料的原則。在中實作資料最小化 AWS 雲端可以降低隱私權風險、成本和分析碳足跡。

資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱 [在上建置資料周邊 AWS](#)。

資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

資料來源

追蹤資料整個生命週期的原始伺服器 and 歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

資料主體

正在收集和處理資料的個人。

資料倉儲

支援商業智慧的資料管理系統，例如分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

資料庫操作語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱 [資料庫定義語言](#)。

深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

defense-in-depth

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在 上採用此策略時 AWS，您可以在 AWS

Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，方法 defense-in-depth 可能會結合多重要素驗證、網路分割和加密。

委派的管理員

在中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶，以管理組織的帳戶並管理該服務的許可。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的 [可搭配 AWS Organizations 運作的服務](#)。

部署

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

開發環境

請參閱 [環境](#)。

偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS 上實作安全控制中的 [偵測性控制](#)。

開發值串流映射 (DVSM)

用於識別和排序限制的程式，這些限制會對軟體開發生命週期中的速度和品質產生不利影響。DVSM 延伸了最初為精實生產實務設計的價值串流映射程序。它專注於透過軟體開發程序建立和移動價值所需的步驟和團隊。

數位分身

真實世界系統的虛擬表示法，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

維度資料表

在 [星狀結構描述](#) 中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常為文字欄位或像文字一樣的離散數字。這些屬性通常用於查詢限制、篩選和結果集標籤。

災難

阻止工作負載或系統在其主要部署位置中實現其業務目標的事件。這些事件可能是自然災難、技術故障或人為動作的結果，例如意外錯誤組態或惡意軟體攻擊。

災難復原 (DR)

您用來將 [災難造成的停機時間和資料遺失降至最低的策略和程序](#)。如需詳細資訊，請參閱 AWS Well-Architected Framework [中的雲端中的工作負載災難復原 AWS：復原](#)。

DML

請參閱[資料庫操作語言](#)。

領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何搭配 strangler fig 模式使用網域驅動設計的資訊，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX \) Web 服務](#)。

DR

請參閱[災難復原](#)。

漂移偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源 中的漂移，或者您可以使用 AWS Control Tower 來[偵測您的登陸區域中可能會影響對治理要求合規性的變更](#)。
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html>

DVSM

請參閱[開發值串流映射](#)。

E

EDA

請參閱[探索性資料分析](#)。

邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與[雲端運算](#) 相比，邊緣運算可以減少通訊延遲並縮短回應時間。

加密

將純文字資料轉換為可人工讀取的運算程序。

加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

端點

請參閱[服務端點](#)。

端點服務

您可以在虛擬私有雲端（VPC）中託管以與其他使用者共用的服務。您可以使用 建立端點服務，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management（IAM）主體。這些帳戶或主體可以透過建立介面端點，私下連線至您的VPC端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud（AmazonVPC）文件中的[建立端點服務](#)。

企業資源規劃（ERP）

可自動化和**管理企業關鍵業務流程**（例如會計[MES](#)、和專案管理）的系統。

信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service（AWS KMS）文件中的[信封加密](#)。

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如，AWS CAF安全特徵包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱[計畫實作指南](#)。

ERP

請參閱[企業資源規劃](#)。

探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。EDA 透過計算摘要統計資料和建立資料視覺化來執行。

F

事實資料表

[星狀結構描述](#) 中的中央資料表。它儲存有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含量值的資料，以及包含維度資料表外部索引鍵的資料。

快速失敗

使用頻繁且增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在中 AWS 雲端，邊界，例如可用區域 AWS 區域、控制平面或資料平面，這些邊界會限制失敗的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱[AWS 故障隔離界限](#)。

功能分支

請參閱[分支](#)。

特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

功能重要性

特徵對於模型的預測有多重要。這通常表示為數值分數，可透過各種技術計算，例如 Shapley 累加解釋 (SHAP) 和整合式漸層。如需詳細資訊，請參閱[使用的機器學習模型可解釋性：AWS](#)。

特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

FGAC

請參閱[精細存取控制](#)。

精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。

閃切遷移

一種資料庫遷移方法，透過[變更資料擷取](#)使用連續資料複寫，以盡可能在最短時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

G

地理封鎖

請參閱[地理限制](#)。

地理限制 (地理封鎖)

在 Amazon 中 CloudFront，此選項可防止特定國家/地區的使用者存取內容分發。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的[限制內容的地理分佈](#)。

Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被視為舊版，而以[中繼線為基礎的工作流程](#)是現代的首選方法。

綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為[棕地](#)。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

防護機制

高階規則，可協助管理組織單位 () 之間的資源、政策和合規性OUs。預防性防護機制會強制執行政策，以確保符合合規標準。它們是透過使用服務控制政策和IAM許可界限來實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。其實作方式是使用 AWS Config、AWS Security Hub、Amazon GuardDuty、AWS Trusted Advisor、Amazon Inspector 和自訂 AWS Lambda 檢查。

H

HA

請參閱[高可用性](#)。

異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如, Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分, 而轉換結構描述可能是一項複雜任務。[AWS 提供有助於結構描述轉換的 AWS SCT](#)。

高可用性 (HA)

工作負載在遇到挑戰或災難時持續運作的能力, 無需介入。HA 系統設計為自動容錯移轉、持續提供高品質效能, 以及處理不同的負載和故障, 並將效能影響降至最低。

歷史現代化

一種用於現代化和升級操作技術 (OT) 系統的方法, 以更好地滿足製造業的需求。歷史記錄器是一種資料庫, 用於從工廠的不同來源收集和存放資料。

異質資料庫遷移

將來源資料庫遷移至共用相同資料庫引擎的目標資料庫 (例如 Microsoft SQL Server 至 Amazon RDS for SQL Server)。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

常用資料

經常存取的資料, 例如即時資料或最近的轉譯資料。此資料通常需要高效能儲存層或類別, 才能提供快速的查詢回應。

修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性, 修正程式通常在典型 DevOps 的發行工作流程之外建立。

超級護理期間

在切換後, 遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常, 此期間的長度為 1-4 天。在超級護理期間結束時, 遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

laC

將[基礎設施視為程式碼](#)。

身分型政策

連接至一或多個IAM主體的政策, 其定義其在 AWS 雲端環境中的許可。

閒置應用程式

在 90 天內，平均 CPU 和記憶體用量介於 5% 到 20% 的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

IloT

請參閱 [工業物聯網](#)。

不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有基礎設施。與可變基礎設施相比，不可避免的 [基礎設施](#) 本質上更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的 [使用不可變基礎設施](#) 部署最佳實務。

傳入（傳入）VPC

在 AWS 多帳戶架構中，VPC 接受、檢查和路由來自應用程式外部的網路連線。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

工業 4.0

由 [Klaus Schwab](#) 於 2016 年推出的術語，指透過連線能力、即時資料、自動化、分析和 AI/ML 的進步來現代化製造程序。

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

工業物聯網（IloT）

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱 [建置工業物聯網（IloT）數位轉型策略](#)。

檢查 VPC

在 AWS 多帳戶架構中，集中 VPC 管理 VPCs（在相同或不同的 AWS 區域）、網際網路和內部部署網路之間的網路流量檢查。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱[什麼是 IoT？](#)

可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱[使用的機器學習模型可解釋性 AWS](#)。

IoT

請參閱[物聯網](#)。

IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 提供的基礎 ITSM。

IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需整合雲端操作與 ITSM 工具的相關資訊，請參閱[操作整合指南](#)。

ITIL

請參閱[IT 資訊庫](#)。

ITSM

請參閱[IT 服務管理](#)。

L

標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會被明確指派安全標籤值。使用者安全標籤與資料安全標籤之間的交集決定使用者可以看到哪些資料列和資料欄。

登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱[設定安全且可擴展的多帳戶 AWS 環境](#)。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱[標籤型存取控制](#)。

最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的[套用最低權限許可](#)。

隨即轉移

請參閱[7 Rs](#)。

小端序系統

首先儲存最低有效位元組的系統。另請參閱[端點](#)。

較低的環境

請參閱[環境](#)。

M

機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱[機器學習](#)。

主要分支

請參閱[分支](#)。

惡意軟體

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬程式、間諜軟體和鍵盤側錄程式。

受管服務

AWS 服務可 AWS 操作基礎設施層、作業系統和平台，而且您可以存取端點來存放和擷取資料。Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

製造執行系統 (MES)

用於追蹤、監控、記錄和控制生產程序的軟體系統，可將原物料轉換為工廠的成品。

MAP

請參閱[遷移加速程式](#)。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在運作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[建置機制](#)。

成員帳戶

除了屬於中組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

MES

請參閱[製造執行系統](#)。

訊息佇列遙測傳輸 (MQTT)

根據[發佈/訂閱](#)模式的輕量型 machine-to-machine (M2M) 通訊協定，適用於資源受限的 [IoT](#) 裝置。

微服務

小型、獨立的服務，透過定義明確的方式進行通訊，APIs通常由小型、獨立的團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱[使用無 AWS 伺服器服務整合微服務](#)。

微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務使用輕量型，透過定義明確的介面進行通訊APIs。此架構中的每個微服務都可以進行更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱[在上實作微服務 AWS](#)。

Migration Acceleration Program (MAP)

提供諮詢支援、訓練和服務，以協助組織建立強大的操作基礎以遷移至雲端，並協助抵銷遷移的初始成本的 AWS 計畫。MAP 包含以有條理方式執行舊版遷移的遷移方法，以及一組可自動化和加速常見遷移案例的工具。

大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是 [AWS 遷移策略](#) 的第三階段。

遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。遷移工廠團隊通常包括操作、業務分析師和擁有者、遷移工程師、開發人員和在衝刺中工作的 DevOps 專業人員。20% 至 50% 之間的企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的 [遷移工廠的討論](#) 和 [雲端遷移工廠指南](#)。

遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。遷移中繼資料的範例包括目標子網路、安全群組和 AWS 帳戶。

遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：EC2 使用 AWS Application Migration Service 重新託管遷移至 Amazon。

遷移產品組合評估 (MPA)

線上工具，提供驗證商業案例以遷移至的資訊 AWS 雲端。MPA 提供詳細的產品組合評估（伺服器大小調整、定價、TCO 比較、遷移成本分析）以及遷移規劃（應用程式資料分析和資料收集、應用程式分組、遷移優先順序和波規劃）。[MPA 工具](#)（需要登入）可供所有 AWS 顧問和 APN 合作夥伴顧問免費使用。

遷移就緒狀態評估 (MRA)

使用取得組織雲端就緒狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序 AWS CAF。如需詳細資訊，請參閱 [遷移準備程度指南](#)。MRA 是 [AWS 遷移策略](#) 的第一個階段。

遷移策略

用於將工作負載遷移至的方法 AWS 雲端。如需詳細資訊，請參閱本詞彙表中的 [7 個 Rs](#) 項目，並請參閱將 [組織動員以加速大規模遷移](#)。

機器學習 (ML)

請參閱[機器學習](#)。

現代化

將過時的 (舊版或單一) 應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱[中的應用程式現代化策略 AWS 雲端](#)。

現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱[中的評估應用程式的現代化準備 AWS 雲端](#)程度。

單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱[將單一體系分解為微服務](#)。

MPA

請參閱[遷移產品組合評估](#)。

MQTT

請參閱[訊息佇列遙測傳輸](#)。

多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」或者「這個客戶對哪種產品類別最感興趣？」

可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性，AWS Well-Architected Framework 建議使用[不可變的基礎設施](#)作為最佳實務。

O

OAC

請參閱[原始存取控制](#)。

OAI

請參閱[原始存取身分](#)。

OCM

請參閱[組織變更管理](#)。

離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

OI

請參閱[操作整合](#)。

OLA

請參閱[操作層級協議](#)。

線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

OPC-UA

請參閱[開啟程序通訊 - Unified Architecture](#)。

開放程序通訊 - Unified Architecture (OPC-UA)

工業自動化的 machine-to-machine (M2M) 通訊協定。OPC-UA 提供與資料加密、身分驗證和授權方案的互通性標準。

操作層級協議 (OLA)

闡明哪些功能性 IT 群組承諾相互交付的協議，以支援服務層級協議 (SLA)。

操作預備檢閱 (ORR)

問題及相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[操作整備檢閱 \(ORR \)](#)。

操作技術 (OT)

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造中，整合 OT 和資訊技術 (IT) 系統是 [Industry 4.0](#) 轉型的關鍵重點。

操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱[操作整合指南](#)。

組織追蹤

由建立的追蹤 AWS CloudTrail 會記錄 AWS 帳戶中組織中所有的事件 AWS Organizations。在屬於組織的每個 AWS 帳戶中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱文件中的 CloudTrail[為組織建立追蹤](#)。

組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變革採用、解決轉型問題，以及推動文化和組織變革，協助組織準備和轉換新系統和策略。在 AWS 遷移策略中，由於雲端採用專案所需的變更速度，因此此架構稱為人員加速。如需詳細資訊，請參閱[OCM 指南](#)。

原始存取控制 (OAC)

在中 CloudFront，用於限制存取以保護您的 Amazon Simple Storage Service (Amazon S3) 內容的增強型選項。OAC 支援所有中的所有 S3 儲存貯體 AWS 區域，伺服器端加密使用 AWS KMS (SSE-KMS)，以及對 S3 儲存貯體的動態PUT和DELETE請求。

原始存取身分 (OAI)

在中 CloudFront，此選項用於限制存取，以保護您的 Amazon S3 內容。當您使用時OAI，會 CloudFront 建立 Amazon S3 可以驗證的主體。已驗證的主體只能透過特定 CloudFront 分發存取 S3 儲存貯體中的內容。另請參閱[OAC](#)，它提供更精細和增強的存取控制。

ORR

請參閱[操作預備檢閱](#)。

OT

請參閱[操作技術](#)。

傳出（輸出）VPC

在 AWS 多帳戶架構中，VPC處理從應用程式內啟動之網路連線的。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

P

許可界限

連接至IAM主體的IAM管理政策，用於設定使用者或角色可擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的[許可界限](#)。

個人身分資訊（PII）

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。的範例PII包括名稱、地址和聯絡資訊。

PII

請參閱[個人識別資訊](#)。

手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

PLC

請參閱[可程式邏輯控制器](#)。

PLM

請參閱[產品生命週期管理](#)。

政策

可以定義許可（請參閱[身分型政策](#)）、指定存取條件（請參閱[資源型政策](#)）或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱[服務控制政策](#)）。

混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則

可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱[在微服務中啟用資料持久性](#)。

組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱[評估遷移準備程度](#)。

述詞

傳回 true 或的查詢條件 false，通常位於 WHERE 子句中。

述詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這可減少必須從關聯式資料庫擷取和處理的資料量，並提高查詢效能。

預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[預防性控制](#)。

委託人

中可執行動作和存取資源 AWS 的實體。此實體通常是 AWS 帳戶、IAM 角色或使用者的根使用者。如需詳細資訊，請參閱 IAM 文件中[角色術語和概念](#)中的主體。

設計隱私

一種系統工程方法，在整個工程過程中將隱私權納入考量。

私有託管區域

容器，其中包含您希望 Amazon Route 53 如何回應一個或多個內網域及其子網域的 DNS 查詢的資訊 VPCs。如需詳細資訊，請參閱 Route 53 文件中的[使用私有託管區域](#)。

主動控制

旨在防止部署不合規資源的[安全控制](#)。這些控制項會先掃描資源，然後再佈建。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的[控制項參考指南](#)，並請參閱在上實作安全[控制項中的主動](#)控制項。 AWS

產品生命週期管理 (PLM)

產品整個生命週期的資料和程序管理，從設計、開發和啟動，到成長和成熟，再到拒絕和移除。

生產環境

請參閱[環境](#)。

可程式設計邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

擬匿名化

將資料集中的個人識別碼取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

發佈/訂閱 (pub/sub)

一種模式，可在微服務之間啟用非同步通訊，以提高可擴展性和回應能力。例如，在微服務型中 [MES](#)，微服務可以將事件訊息發佈到其他微服務可以訂閱的頻道。系統可以新增新的微服務，而無需變更發佈服務。

Q

查詢計劃

一系列步驟，如指示，用於存取SQL關聯式資料庫系統中的資料。

查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

R

RACI 矩陣

請參閱 [負責、負責、已諮詢、知情 \(RACI \)](#)。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

RASCI 矩陣

請參閱 [負責、負責、已諮詢、知情 \(RACI \)](#)。

RCAC

請參閱 [資料列和資料欄存取控制](#)。

僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

重新架構師

請參閱 [7 Rs](#)。

復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

復原時間目標 (RTO)

服務中斷和服務還原之間的可接受延遲上限。

重構

請參閱 [7 Rs](#)。

區域

地理區域 AWS 的資源集合。每個 AWS 區域 都獨立於其他，以提供容錯能力、穩定性和彈性。如需詳細資訊，請參閱[指定 AWS 區域 哪些帳戶可以使用](#)。

迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

重新託管

請參閱 [7 Rs](#)。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新定位

請參閱 [7 Rs](#)。

轉譯形式

請參閱 [7 Rs](#)。

回購

請參閱 [7 Rs](#)。

彈性

應用程式抵抗中斷或從中斷中復原的能力。[在中規劃復原能力時，高可用性和災難復原](#)是常見的考量因素 AWS 雲端。如需詳細資訊，請參閱[AWS 雲端 彈性](#)。

資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

負責任、負責、已諮詢、知情（RACI）矩陣

定義所有涉及遷移活動和雲端操作各方的角色和責任的矩陣。矩陣名稱衍生自矩陣中定義的責任類型：負責人（R）、責任（A）、已諮詢（C）和知情（I）。支援（S）類型為選用。如果您包含支援，則矩陣稱為RASCI矩陣，如果您排除該矩陣，則稱為RACI矩陣。

回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[回應性控制](#)。

保留

請參閱 [7 Rs](#)。

淘汰

請參閱 [7 Rs](#)。

輪換

定期更新[秘密](#)的程序，讓攻擊者更難存取憑證。

資料列和資料欄存取控制（RCAC）

使用已定義存取規則的基本靈活SQL表達式。RCAC 包含資料列許可和資料欄遮罩。

RPO

請參閱[復原點目標](#)。

RTO

請參閱[復原時間目標](#)。

執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

S

SAML 2.0

許多身分提供者（IdPs）使用的開放標準。此功能會啟用聯合單一登入（SSO），因此使用者可以登入 AWS Management Console 或呼叫操作，AWS API 而不必 IAM 為您組織中的每個人建立中的使用者。如需 SAML 2.0 型聯合的詳細資訊，請參閱 IAM 文件中的[關於 SAML 2.0 型聯合](#)。

SCADA

請參閱 [監控控制和資料擷取](#)。

SCP

請參閱 [服務控制政策](#)。

秘密

在中 AWS Secrets Manager，以加密形式存放的機密或限制資訊，例如密碼或使用者憑證。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 [Secrets Manager 文件中的 Secrets Manager 秘密中的內容？](#)。

安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全控制有四種主要類型：[預防性](#)、[偵測性](#)、[回應性](#) 和 [主動](#)。

安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

安全資訊和事件管理（SIEM）系統

結合安全資訊管理（SIM）和安全事件管理（SEM）系統的工具和服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生警示。

安全回應自動化

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為[偵測](#)或[回應](#)式安全控制，協助您實作 AWS 安全最佳實務。自動化回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換憑證。

伺服器端加密

由接收資料的 AWS 服務 加密其目的地的資料。

服務控制政策 (SCP)

為 AWS Organizations 中的組織的所有帳戶提供集中控制許可的政策。SCPs 定義管理員可委派給使用者或角色之動作的防護機制或設定限制。您可以使用 SCPs 作為允許清單或拒絕清單，以指定允許或禁止的服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的[服務控制政策](#)。

服務端點

URL 的進入點的 AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 [AWS 服務端點](#)。

服務層級協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

服務層級指標 (SLI)

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標 (SLO)

代表服務運作狀態的目標指標，由[服務層級指標](#)測量。

共同責任模式

描述您共享 AWS 的雲端安全與合規責任的模型。AWS 負責雲端的安全，而您要負責雲端的安全。如需詳細資訊，請參閱[共同責任模式](#)。

SIEM

請參閱[安全資訊和事件管理系統](#)。

單一失敗點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱[服務層級協議](#)。

SLI

請參閱[服務層級指示器](#)。

SLO

請參閱[服務層級目標](#)。

split-and-seed 模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱 [中的階段式應用程式現代化方法 AWS 雲端](#)。

SPOF

請參閱 [單一失敗點](#)。

星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構專為 [資料倉儲](#) 或商業智慧用途而設計。

Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由 [Martin Fowler 引入](#)，作為重寫單一系統時管理風險的方式。如需如何套用此模式的範例，請參閱 [使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX \) Web 服務](#)。

子網

中 IP 地址的範圍 VPC。子網必須位於單一可用區域。

監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 [Amazon CloudWatch Synthetics](#) 來建立這些測試。

T

標籤

作為中繼資料的鍵值對，用於組織您的 AWS 資源。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱 [標記您的 AWS 資源](#)。

目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

測試環境

請參閱[環境](#)。

訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

傳輸閘道

可用來互連 VPCs 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的[什麼是傳輸閘道](#)。

主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

受信任的存取權

將許可授予您指定的服務，以代表您在組織中在其帳戶中 AWS Organizations 執行任務。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱文件中的 AWS Organizations [AWS Organizations 搭配使用其他 AWS 服務](#)。

調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

雙比薩團隊

一個小型 DevOps 團隊，您可以使用兩個披薩來饋送。雙披薩團隊規模可確保軟體開發中的最佳協作。

U

不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱[量化深度學習系統的不確定性](#)指南。

未區分的任務

也稱為繁重型，是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、維護和容量規劃。

較高的環境

請參閱[環境](#)。

V

清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

VPC 對等

兩個之間的連線VPCs，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱[Amazon 文件中的VPC互連內容](#)。VPC

漏洞

損害系統安全性的軟體或硬體缺陷。

W

暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等慢的查詢。

視窗函數

對以某種方式與目前記錄相關聯的資料列群組執行計算的SQL函數。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器 and 應用程式。

WORM

請參閱[寫入一次，讀取許多](#)。

WQF

請參閱[AWS工作負載資格架構](#)。

寫入一次，讀取許多 (WORM)

一次性寫入資料的儲存模型，可防止刪除或修改資料。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為[不可變的](#)。

Z

零時差漏洞

利用[零時差漏洞](#)的攻擊，通常是惡意軟體。

零時差漏洞

生產系統中未緩解的缺陷或漏洞。威脅發動者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

殭屍應用程式

平均CPU和記憶體用量低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。