

AWS Well-Architected Framework

永續性支柱



永續性支柱: AWS Well-Architected Framework

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

摘要與簡介	i
簡介	1
雲端永續性	2
共同責任模式	2
雲端的永續性	3
雲端永續性	3
永續性得利於雲端	4
雲端中永續性的設計原則	4
改進程序	6
範例情節	6
識別改進目標	7
資源	7
評估特定改進	7
代理指標	7
業務指標	8
關鍵績效指標	8
估計改進	9
評估改進	9
確定改進的優先順序和規劃改進	10
測試和驗證改進	10
將變更部署至生產環境	11
測量結果並複製成功	12
永續性作為非功能要求	14
雲端中永續性的最佳實務	15
區域選擇	15
SUS01-BP01 根據您的業務要求和永續性發展目標選擇區域	15
因應需求	17
SUS02-BP01 動態擴展工作負載基礎架構	17
SUS02-BP02 讓 SLA 符合永續性目標	20
SUS02-BP03 停止建立和維護未使用的資產	21
SUS02-BP04 根據使用者的聯網要求優化其工作負載的地理位置	22
SUS02-BP05 為執行的活動優化團隊成員資源	24
SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化	25
軟體和架構	27

SUS03-BP01 最佳化非同步與排程任務的軟體和架構	28
SUS03-BP02 移除或重構使用量低或完全未使用的工作負載元件	30
SUS03-BP03 優化程式碼中耗用最多時間或資源的部分	31
SUS03-BP04 優化對裝置和設備的影響	33
SUS03-BP05 使用最能支援資料存取和儲存模式的軟體模式和架構	34
資料管理	36
SUS04-BP01 實作資料分類政策	36
SUS04-BP02 使用支援資料存取和儲存模式的技術	38
SUS04-BP03 使用政策來管理資料集的生命週期	41
SUS04-BP04 使用彈性和自動化擴充區塊儲存或檔案系統	43
SUS04-BP05 移除不需要或多餘的資料	44
SUS04-BP06 使用共用檔案系統或儲存體存取通用資料	46
SUS04-BP07 盡可能減少跨網路的資料移動	48
SUS04-BP08 僅在難以重新建立時才備份資料	50
硬體和服務	51
SUS05-BP01 使用最低數量的硬體來滿足需求	52
SUS05-BP02 使用影響最小的執行個體類型	53
SUS05-BP03 使用受管服務	56
SUS05-BP04 將硬體型運算加速器的使用方式最佳化	58
程序和文化	59
SUS06-BP01 採用可快速導入永續性改進的方法	60
SUS06-BP02 讓工作負載保持在最新狀態	61
SUS06-BP03 提高建置環境的使用率	63
SUS06-BP04 使用受管 Device Farm 進行測試	64
結論	66
作者群	67
深入閱讀	68
文件修訂	69
聲明	70
AWS Glossary	71

永續性支柱 - AWS Well-Architected Framework

出版日期：2023 年 10 月 3 日 ([文件修訂](#))

本白皮書的重點是 Amazon Web Services (AWS) Well-Architected Framework 的永續性支柱。它提供設計原則、營運指引、最佳實務、潛在的取捨，以及您可以用來達到 AWS 工作負載之永續性目標的改善計劃。

簡介

AWS Well-Architected Framework 可協助您了解在 AWS 上建置工作負載時所做決策的優缺點。透過此架構，您將了解架構的最佳實務，以便在 AWS 雲端設計和操作安全、可靠、有效率、經濟實惠且永續的工作負載。該架構可讓您根據最佳實務一致地量測架構，並找出需要改進的方面。擁有 Well-Architected 工作負載可大幅提升您支援業務成果的能力。

此架構以六大支柱為基礎：

- 卓越營運
- 安全性
- 可靠性
- 效能達成效率
- 成本最佳化
- 永續性

本文件專重於永續性支柱，而且在永續性範圍內，它專重於環境永續性。其適用於擔任技術職務的人員，例如技術長 (CTO)、架構師、開發人員和營運團隊成員。

閱讀本文件之後，您將了解在考慮永續性的情況下，設計雲端架構時應使用的 AWS 目前的建議和策略。藉由採用本文中的實務，您可以建置將效率最大化並減少浪費的架構。

雲端永續性

永續性學科規範業務活動對環境、經濟和社會的長期影響。AWS Well-Architected [聯合國世界環境與發展委員會](#) 將永續發展定義為「滿足現今需求而不犧牲後代滿足其需求之能力的發展」。您的企業或組織可能會對環境產生負面影響，例如直接或間接的碳排放、不可回收的廢棄物，以及對乾淨水源等共享資源的破壞。

在建置雲端工作負載時，永續性實務是關於了解所使用服務的影響、量化整體工作負載生命週期的影響，以及應用設計原則和最佳實務來減少這些影響。本文件專注於環境影響，尤其是能源消耗和效率，因為這方面是架構師採取直接行動來減少資源使用的重要手段。

當專注於環境影響時，您應該了解通常如何說明這些影響，以及對您組織自己的排放會計的後續影響。AWS Well-Architected [溫室氣體協定](#) 將碳排放組織成以下範圍，以及雲端供應商 (例如 AWS) 之每個範圍內的相關排放範例：

- 範圍 1：來自組織活動或在組織控制下的所有直接排放。例如，資料中心備份發電機的燃料燃燒。
- 範圍 2：來自購買和用來驅動資料中心和其他設施之電力的間接排放。例如，來自商業發電的排放。
- 範圍 3：來自組織活動且無法從源頭控制的所有其他間接排放。AWS 範例包括與資料中心建構相關的排放，以及與資料中心所部署 IT 硬體之製造和運輸相關的排放。

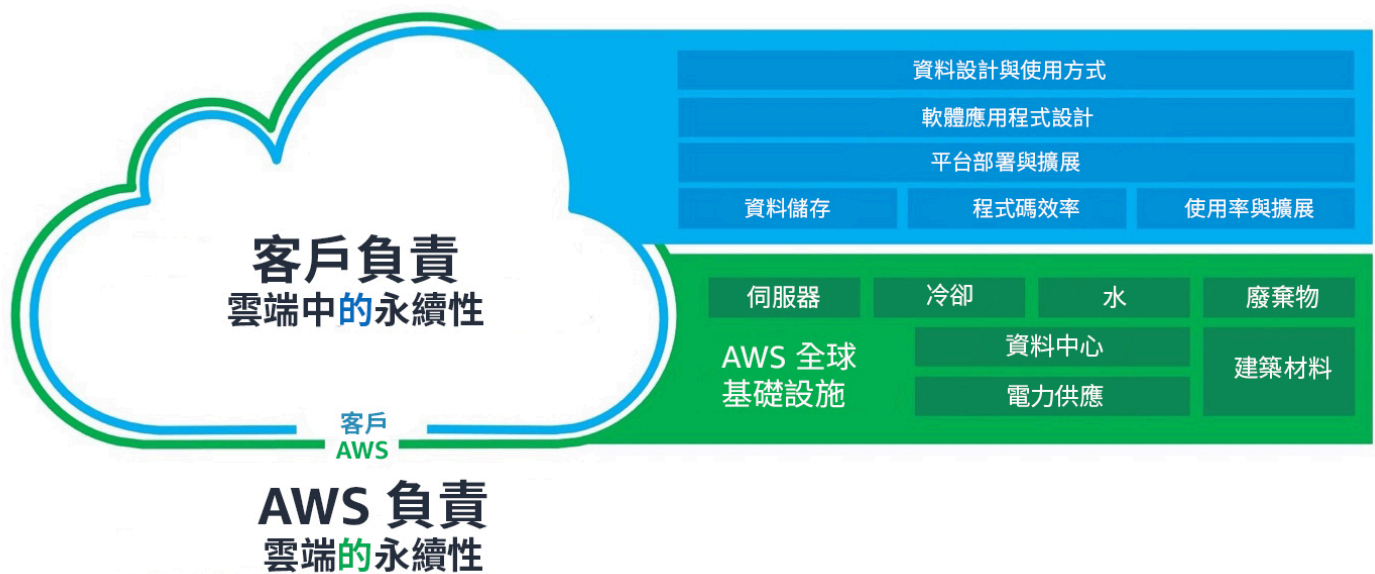
從 AWS 客戶的觀點來看，來自 AWS 上執行之工作量的排放會被視為間接排放，以及範圍 3 排放的一部分。部署的每個工作負載都會產生以前每個範圍之總 AWS 排放量的一小部分。每個工作量的實際數量各不相同，並取決於幾個因素，包括所使用的 AWS 服務、這些服務消耗的能量、為其執行所在 AWS 服務中心提供服務之電網的碳強度，以及 AWS 的可再生能源採購。

本文件首先描述了環境永續性的共同責任模型，然後提供架構最佳實務，因此您可以透過減少您的工作負載在 AWS 資料中心執行所需的總資源，來最大程度地降低這些工作負載的影響。

共同責任模式

環境永續性是客戶與 AWS 之間의 共同責任。

- AWS 負責優化雲端 的 永續性 – 提供高效、共用的基礎設施、水資源管理和採購可再生能源。
- 客戶負責雲端 in 的永續性 – 優化工作負載和資源利用率，並最大程度地減少針對工作負載進行部署所需的總資源。



共同責任模型

雲端的永續性

雲端供應商具有較低的碳足跡，並且比典型的內部部署替代方案更節能，因為他們投資於有效的電力和冷卻技術、操作節能伺服器母體，以及實現高伺服器利用率。雲端工作負載會利用共用的資源 (例如聯網、電力、冷卻和實體設施) 來減少影響。您可以將雲端工作負載遷移到更有效率的技術，並使用雲端服務轉變工作負載，以獲得更好的永續性。

資源

- [移至 Amazon Web Services 的碳減少機會](#)
- [AWS 提供永續性解決方案](#)

雲端永續性

雲端中的永續性是一項持續的工作，主要專注於跨工作負載所有元件的節能和提高效率，方法為透過從佈建的資源中實現最大收益，並最大程度地減少所需的總資源。這項工作的範圍可以從初始選取有效的程式設計語言、採用現代演算法、使用有效的資料儲存技術、部署到正確大小和有效的運算基礎設施，以及最大程度地減少對高耗電最終使用者硬體的要求。

永續性得利於雲端

除了最大程度地減少所部署之工作負載的影響外，您還可以使用 AWS 雲端，來執行其設計旨在支援更廣泛永續性挑戰的工作負載。這些挑戰的範例包括減少碳排放、降低能源消耗、回收水資源，或減少其他業務或組織領域的浪費。

永續性 得利於 雲端是在您使用 AWS 技術，解決更廣泛的永續性挑戰時。例如，您可以使用機器學習服務 (例如 [Amazon Monitron](#))，偵測產業機械中的異常行為。使用此偵測資料，您可以進行預防性維護，以降低由意外設備故障引起之環境事故的風險，並確保機械繼續以最高效率操作。

雲端中永續性的設計原則

在架構雲您的工作負載，以獲取最大永續性並最大程度地減少影響時，套用這些設計原則。

- **了解您的影響：** 衡量您雲端工作負載的影響，並建立工作負載未來影響的模型。包含所有影響來源，包括客戶使用您產品所產生的影響，以及產品最後除役和淘汰所產生的影響。審視每個工作單元所需的資源和排放量，將生產輸出與雲端工作負載的總體影響進行比較。使用這些資料來建立關鍵績效指標 (KPI)、評估提高生產力的同時也降低影響的方法，以及預估建議變更隨時間的影響。
- **建立永續性目標：** 對於每個雲端工作負載，建立長期永續性目標，例如減少每項交易所需的運算和儲存資源。針對改進現有工作負載永續性的投資回報建立模型，並為業主提供投資於永續性目標所需的資源。針對成長著手規劃，以及建構您的工作負載，讓成長導致的每個使用者或每項交易 (根據適當的單位測量) 影響強度降低。這些目標有助於支持貴企業或組織更廣泛的永續性發展目標、識別迴歸，以及排定潛在改進領域的優先順序。
- **最大化使用率：** 調整合適的工作負載規模並實施高效率設計，以確保高使用率，並將底層硬體的能源效率發揮到最大。由於每部主機會有基準耗電量，因此兩部以 30% 使用率執行的主機效率，會低於一部以 60% 執行的主機效率。同時消除或最大程度地減少閒置資源、處理和儲存，以減少為工作負載供電所需的能源總量。
- **預測並採用新的、更有效率的軟硬體產品：** 支持合作夥伴和供應商進行上游改進，以協助您減少雲端工作負載的影響。持續監控和評估更有效率的新軟硬體產品。針對靈活性進行設計，以允許快速採用高效率的新技術。
- **使用受管服務：** 在廣泛的客戶群中共用服務，有助於最大化資源利用，進而減少支援雲端工作負載所需的基礎設施數量。例如，客戶可以將工作負載遷移到 AWS 雲端並採用受管服務 (例如適用於無伺服器容器的 AWS Fargate，AWS 在其中大規模運作並負責高效率操作)，來分攤電力和網路等通用資料中心元件的影響。使用可協助將影響降至最低的受管服務，例如使用 Amazon S3 生命週期組態將不常存取的資料自動移動到冷儲存，或使用 Amazon EC2 Auto Scaling 調整容量以滿足需求。

- 減少雲端工作負載的下游影響：減少使用您的服務所需的能源或資源量。減少或消除客戶需要升級其裝置才能使用您服務的需求。使用 Device Farm 進行測試以了解預期影響，並與客戶進行測試以了解使用您服務的實際影響。

改進程序

架構改進程序包括了解您擁有的項目以及可採取什麼改進措施、選擇改進目標、測試改進、採納成功的改進措施、量化成功，以及分享心得以便在其他地方複製成功，然後重複這個循環。

改進目標可以是：

- 消除浪費、低利用率和閒置或未用資源
- 從您耗用的資源取得最大價值

Note

使用您佈建的所有資源，並盡可能使用的最低資源完成相同的工作。

在最佳化的早期階段，首先消除浪費或使用率低的區域，接著轉向更具目標性的最佳化，以符合您的特定工作負載。

監控資源消耗隨時間的變化。找出累積變化導致無效率或大幅增加資源消耗的地方。決定因應消耗變化所需的改進，並實施優先改進措施。

以下步驟被設計為一個反覆程序，用來評估、優先處理、測試和部署永續性為主的雲端工作負載改進。

1. 識別改進目標：針對本文件中識別之永續性的最佳實務審查您的工作負載，並識別改進目標。
2. 評估特定改進：透過潛在改善、預估成本和業務風險這三方面評估特定變更。
3. 優先處理和規劃改進：優先處理以最低成本和風險提供最大改進的變更，並建立測試和實作計劃。
4. 測試和驗證改進：實作測試環境中的變更，以驗證其改進潛力。
5. 將變更部署至生產環境：跨生產環境實作變更。
6. 測量結果並複製成功：尋找機會來複製跨工作負載的成功，並還原成果不可接受的變更。

範例情節

下列範例情境稍後會在本文件中引用，以說明改進程序的每個步驟。

您的公司有一個工作負載，可在 Amazon EC2 執行個體上執行複雜的映像操作，並存放修改後的檔案和原始檔案以供使用者存取。處理活動會大量使用 CPU，並且輸出檔案非常大。

識別改進目標

了解可以協助您實現永續性目標的最佳實務。您稍後可在本文件中尋找這些 [最佳實務](#) 的詳細描述和改進建議。

審查您的工作負載和使用的資源。識別 熱點 ，例如大型部署和經常使用的資源。評估這些熱點以取得機會，來改善資源的有效利用率，並減少實現業務成果所需的總資源。

針對最佳實務審查您的工作負載，並識別改進的候選項目。

將此步驟套用至 [範例情節](#)，您可將以下最佳實務識別為可能的改進目標：

- 使用最低數量的硬體來滿足需求
- 使用最能支援資料存取和儲存模式的技術

資源

- [優化您的 AWS 永續性架構，第 I 部分：運算](#)
- [優化您的 AWS 永續性架構，第 II 部分：儲存](#)
- [優化您的 AWS 永續性架構，第 III 部分：聯網](#)

評估特定改進

了解您的工作量佈建以完成工作單位的資源。評估潛在的改進，並估計其潛在影響、實作成本，以及相關聯的風險。

若要測量一段時間後的改進，請首先了解您已在 AWS 佈建的內容，以及如何耗用這些資源。

從 AWS 用量的完整概觀開始，然後使用 AWS 成本和用量報告來協助識別熱點。使用此 [AWS 範例程式碼](#)，藉助 Amazon Athena 來協助您審查並分析您的報告。

代理指標

當評估特定的變更時，您也須評估哪些指標最能量化該變更對相關聯資源的影響。這些指標稱為 代理指標。選取代理指標，最能反映您評估的改進類型，以及作為改進目標的資源。這些指標可能會隨著時間而演進。

佈建以支援工作量的資源包括運算、儲存和網路資源。使用您的代理指標評估佈建的資源，以查看如何耗用這些資源。

使用您的代理指標來測量為了實現業務成果而佈建的資源。

資源	範例代理指標	改進目標
運算	vCPU 分鐘	所佈建資源的最大使用率
儲存	佈建的 GB	減少總佈建量
網路	傳輸的 GB 或傳輸的封包	減少總傳輸量和傳輸距離

業務指標

選取業務指標來量化業務成果的實現程度。您的業務指標應反映工作負載提供的值，例如同時作用中使用者的數量、已提供的 API 呼叫或完成的交易數量。這些指標可能會隨著時間而演進。在評估財務型業務指標時要謹慎，因為交易價值的不一致性會使比較無效。

關鍵績效指標

使用以下公式，將佈建的資源除以所實現的業務成果，以確定每工作單位的佈建資源。

$$\text{每個工作單位佈建的資源} = \frac{\text{所佈建資源的代理指標}}{\text{成果的商業指標}}$$

KPI 公式

使用每個工作單位的資源做為 KPI。根據佈建的資源建立基準，做為比較基礎。

資源	範例 KPI	改進目標
運算	每筆交易 vCPU 分鐘	所佈建資源的最大使用率
儲存	每筆交易 GB	減少總佈建量
網路	每筆交易傳輸的 GB 或每筆交易傳輸的封包	減少總傳輸量和傳輸距離

估計改進

將改進同時估計為所佈建資源的定量減少 (如您的代理指標所示)，以及每個工作單位佈建的基準資源的變更百分比。

資源	範例 KPI	改進目標
運算	每筆交易 vCPU 分鐘的 % 減少	最大化使用率
儲存	每筆交易 GB 的 % 減少	減少總佈建量
網路	每筆交易傳輸的 GB 或每筆交易傳輸的封包的 % 減少	減少總傳輸量和傳輸距離

評估改進

針對預期的淨收益評估潛在的改進。評估實作和維護的時間、成本和工作量，以及業務風險，例如非預期的影響。

針對性改進通常代表耗用的資源類型之間的權衡。例如，若要減少運算耗用量，您可以存放結果，或者若要限制傳輸的資料，您可以在將結果傳送至用戶端之前處理資料。但這些 [權衡](#) 會在稍後詳細討論。

在評估工作負載的風險時，包含非功能要求，其中包括安全性、可靠性、效能效率、成本優化，以及改進對您操作工作負載能力的影響。

將此步驟套用至 [範例情節](#)，您可以使用下列結果評估目標改進：

最佳實務	針對性改進	潛在	成本	風險
使用最低數量的硬體來滿足需求	實作預測性擴展來減少低利用率期間	中	低	低
使用最能支援資料存取和儲存模式的技術	實作更有效的壓縮機制，以減少總儲存量和實現它的時間	高	低	低

實作預測性排程可減少未充分利用或未使用之執行個體所耗用的 vCPU 小時，從而對現有的擴展機制提供適中的效益，估計耗用的資源減少了 11%。涉及的成本很低，並包括雲端資源的組態和 Amazon EC2 Auto Scaling 的預測性擴展操作。當反應式執行橫向擴展，以回應超出預測的需求時，風險是受約束的表現。

實作更有效的壓縮將產生重大影響，在所有原始影像和操作的影像中，檔案會大幅地減少，估計生產環境中的儲存需求減少了 25%。實作新的演算法是一種不費力的替代方案，而涉及的風險很低。

確定改進的優先順序和規劃改進

根據成本最低且風險可接受的最大預期影響，確定您所識別之改進的優先順序。

判斷最初要專注哪些改進，並將其包括在您的資源規劃和開發藍圖中。

將此步驟套用至 [範例情節](#)，您可以確定目標改進的優先順序，如下所示：

優先順序	稍微改善	潛在	成本	風險
1	實作更有效的壓縮機制	高	低	低
2	實作預測性擴展	中	低	低

更新檔案壓縮的高潛力、低成本和風險，使其成為貴公司的高價值目標，而優先於實作預測性擴展。您確定實作具有中等潛在影響、低成本和低風險的預測性擴展，應該是在檔案壓縮完成之後的優先改善。

您指派團隊成員，以實作改進的檔案壓縮，並將預測性擴展新增至您的待處理項目。

測試和驗證改進

執行最小投資的小型測試，以降低大規模工作的風險。

在測試環境中實作工作負載的代表性副本，以限制執行測試和驗證的成本和風險。執行一組預先定義的測試交易、測量佈建的資源，並確定每工作單位使用的資源來建立測試基準。

在測試環境中實作目標改進，並在相同條件下使用相同的方法重複測試。然後，透過適當的改進來測量佈建的資源和每工作單位使用的資源。

從每工作單位佈建的資源基準計算百分比變化，並確定預期定量減少生產環境中佈建的資源。針對預期值比較這些值。判斷結果是否為可接受的改進等級。評估其他所耗用資源中的任何權衡是否使改進的淨收益不可接受。

判斷改進是否成功，以及是否應投入資源，實作生產環境中的變更。如果此時更改被評估為不成功，請重定導向您的資源，來測試並驗證您的下一個目標，並繼續您的改進週期。

每個工作單位佈建的資源中的 % 減少	所佈建資源中的定量減少	動作
已符合期望	已符合期望	繼續改進
未符合期望	已符合期望	繼續改進
已符合期望	未符合期望	追求替代改進
未符合期望	未符合期望	追求替代改進

將此步驟套用至 [範例情節](#) 您可以執行測試來驗證成功。

在您對改進的壓縮演算法執行測試之後，每個工作單元所佈建資源 (原始影像和修改後影像所需的儲存空間) 中的減少百分比已符合期望，佈建的儲存空間平均減少 30%，而增加的運算負載可忽略不計。

您確定將改進的壓縮演算法套用至生產環境中現有檔案所需的額外運算資源，相比於所實現的儲存空間減少微不足道。您已確認成功定量減少所需資源 (數 TB 儲存空間)，並且核准改進用於生產環境部署。

將變更部署至生產環境

對生產環境實作已測試、驗證和核准的改進。實作有限部署、確認工作負載的功能、測試在有限部署內所佈建資源和每個工作單元所耗用資源的實際減少情況，並檢查是否有意外的變更結果。在成功測試之後繼續完整部署。

如果測試失敗或您遇到不可接受的意外變更結果，請還原變更。

將此步驟套用至 [範例情節](#)，您可以採取以下動作。

您透過藍綠部署方法使用有限部署，在生產環境中實作變更。針對新部署之執行個體的功能測試成功。對於原始影像檔和操作後的影像檔，您會看到佈建的儲存空間平均減少 26%。您看不到壓縮新文件時運算負載增加的任何證據。

您注意到壓縮影像檔的經過時間出現非預期的減少，並將其歸因於新壓縮演算法的高度優化程式碼。

您繼續新版本的完整部署。

測量結果並複製成功

以下列方式測量結果並複製成功：

- 測量每個工作單位所佈建資源的初始改進，以及所佈建資源中的定量減少。
- 將初始估計和測試結果與您的生產環境測量結果進行比較。識別確可能導致差異的因素，並在適當的情況下更新您的估計和測試方法。
- 確定成功和成功程度，並與利害關係人分享結果。
- 如果您由於測試失敗或變更帶來的意外負面後果而必須還原變更，請識別貢獻因素。若可行則反覆進行，或評估達到變更目標的新方法。
- 吸取你所學到的經驗、建立標準，並將成功的改進套用到其他可以同樣受益的系統。跨團隊和組織擷取和分享您的方法、相關成品和淨收益，以便其他人可以採用您的標準並複製您的成功。
- 監控每個工作單元佈建的資源，並追蹤一段時間後的變化和總體影響。對工作負載的變更，或者您的客戶如何耗用您的工作負載，可能會對您的改進效果產生影響。如果您注意到改進的效果在短期內顯著下降，或者一段時間後效果累積下降，請重新評估改進機會。
- 量化一段時間後改進帶來的淨收益 (包括在可用時套用改進的其他團隊所獲得的收益)，以顯示改進活動的投資報酬率。

將此步驟套用至 [範例情節](#)，您可以測量下列結果。

在部署新的壓縮演算法，並將其套用至現有的影像檔之後，您的工作負載顯示初始改進將儲存需求減少了 23%。

測量值與初始估計值 (25%) 大略一致，與測試 (30%) 相比的顯著差異被確定為測試中使用的影像檔不代表生產環境中存在的影像檔的結果。您修改測試影像集，以更適當地反映生產環境中的影像。

改進被視為完全成功。所佈建儲存的總減少量比估計的 25% 減少了 2%，但 23% 仍然是永續性影響中的巨大改進，並伴隨著對等的成本節約。

變更的唯一意外結果是有利地減少執行壓縮的經過時間，以及對等減少耗用的 vCPU。這些改進歸功於高度優化的程式碼。

您建立一個內部開放原始碼專案，您可以在其中分享您的程式碼、相關聯的成品、有關如何實作變更的指引，以及實作結果。內部開放原始碼專案可讓您的團隊輕鬆地針對其所有持續的檔案儲存使用案例採

用程式碼。您的團隊會採用改進作為標準。內部開放原始碼專案的次要好處是，採用解決方案的每個人都可以從對解決方案的改進中獲益，而且任何人都可以為專案的改進做出貢獻。

您發佈您的成功並在您的整個組織中分享開放原始碼專案。採用解決方案的每個團隊都以最少的投資複製收益，並將其新增至從您的投資中獲得的淨收益。您將此資料發佈為持續成功案例。

您將繼續監控改進在一段時間後的影響，並視需要對內部開放原始碼專案進行變更。

永續性作為非功能要求

將永續性新增至您的業務需求清單，可以產生更符合成本效益的解決方案。專注於從您使用的資源中獲得更多價值，並使用更少的資源，在 AWS 上直接轉換為成本節約，因為您只需為使用的項目付費。

達到永續性目標可能不需要在一個或多個其他傳統指標 (例如運行時間、可用性或回應時間) 中進行對等權衡。您可以在永續性方面獲得顯著的收益，而不會對服務等級產生任何可衡量的影響。在需要進行小幅度權衡的情況下，這些權衡所獲得的永續性改進可能會優於服務品質的變更。

鼓勵您的團隊成員在開發功能要求時持續嘗試永續性改進。團隊還應該在設定目標時嵌入代理指標，以確保他們在開發工作負載時評估資源強度。

以下是可以減少您耗用之雲端資源的範例權衡：

調整結果的品質： 您可以透過近似運算犧牲結果品質 (QoR) 來換取工作負載強度的下降。採用近似運算的實務尋找機會，以利用客戶所需項目與您實際產生的項目之間的差距。例如，您將資料置於設定的資料結構，您可以在 SQL 中捨棄 ORDER BY 運算子，以刪除不必要的處理，進而節省資源，同時仍提供可接受的答案。

調整回應時間： 以較慢的回應時間提供答案可藉由將共用開銷降到最低，來節省碳量。處理隨機操作、暫時性任務可能會產生啟動開銷。將任務分組並以批次處理它們，而不是在每次任務到達時支付開銷費用。批次處理可透過犧牲回應時間，以降低啟動執行個體、下載原始程式碼，以及執行程序所需的共同開銷。

調整可用性： 使用 AWS，您只需通過幾個步驟新增冗餘並達到高可用性目標。您可以藉由佈建一律導致利用率降低的閒置資源，透過靜態穩定性這類技術增加冗餘。在設定目標時評估業務需求。可用性方面相對輕微的權衡可以導致利用率大幅改進。例如，靜態穩定性架構模式涉及佈建閒置的容錯移轉容量，以在元件故障之後立即開始接受負載。放寬可用性要求，可以透過允許留出時間將部署取代資源自動化，來移除閒置線上容量的需求。隨需新增容錯移轉容量可提高整體使用率，在一般營運期間不會影響業務，而且有降低成本的次要好處。

雲端中永續性的最佳實務

最佳化工作負載位置，以及針對需求、軟體、資料、硬體和程序最佳化您的架構，以提高能源效率。每一個領域都代表了採用最佳實務的機會，透過最大化使用率、最小化廢棄物以及為支援工作負載而部署和供電的總資源，來降低雲端工作負載的永續性影響。

主題

- [區域選擇](#)
- [因應需求](#)
- [軟體和架構](#)
- [資料管理](#)
- [硬體和服務](#)
- [程序和文化](#)

區域選擇

您為工作負載選擇的區域會極大程度地影響其 KPI，包括效能、成本和碳足跡。若要有有效改進這些 KPI，請根據您的業務要求和永續性目標，選擇工作負載的區域。

最佳實務

- [SUS01-BP01 根據您的業務要求和永續性發展目標選擇區域](#)

SUS01-BP01 根據您的業務要求和永續性發展目標選擇區域

根據您的業務要求和永續性發展目標選擇工作負載的區域，以將其 KPI 最佳化，包括效能、成本和碳足跡。

常見的反模式：

- 您可以根據自身所在位置選取工作負載的區域。
- 您可以將所有工作負載資源合併到單一地理位置。

建立此最佳實務的優勢：將工作負載放在 Amazon 可再生能源專案附近或所公佈的碳強度較低的區域附近，有助於降低雲端工作負載的碳足跡。

未建立此最佳實務時的風險暴露等級：中

實作指引

AWS 雲端 是會持續擴張的區域和連接點 (POP) 網路，並透過全球網路基礎設施彼此連結起來。您為工作負載選擇的區域會極大程度地影響其 KPI，包括效能、成本和碳足跡。若有效改進這些 KPI，請根據您的業務要求和永續性發展目標，選擇工作負載的區域。

實作步驟

- 請遵循這些步驟，根據您的業務要求 (包括合規、可用功能、成本和延遲) 評估工作負載的可能區域，並將這些區域列入候選清單：
 - 根據必須遵守的當地法規，確認這些區域符合規範。
 - 使用 [AWS 區域服務清單](#) 來檢查區域是否有您執行工作負載時所需的服務和功能。
 - 使用 [AWS Pricing Calculator](#) 計算工作負載在每個區域的成本。
 - 測試終端使用者所在位置和每個 AWS 區域 之間的網路延遲。
- 選擇 Amazon 可再生能源專案附近的區域，以及電網公佈的碳強度低於其他位置 (或區域) 的區域。
 - 識別相關的永續性指導方針，根據[溫室氣體協定](#) (市場型和位置型方法) 來追蹤和比較逐年的碳排放。
 - 根據您用來追蹤碳排放的方法來選擇區域。如需根據永續性指導方針來選擇區域的詳細資訊，請參閱[如何根據永續性目標來選取工作負載的區域](#)。

資源

相關文件：

- [了解您的碳排放預估值](#)
- [全球 Amazon](#)
- [可再生能源方法](#)
- [為工作負載選取區域時應考慮的事項](#)

相關影片：

- [永續建構和降低您的 AWS 碳足跡](#)

因應需求

使用者和應用程式使用工作負載和其他資源的方式，可協助您找到改善的機會，以達成永續性目標。擴展基礎架構以持續符合需求，並確認您僅使用支援使用者所需的最低資源。讓服務層級符合客戶需求。妥善放置資源，以限制使用者和應用程式使用資源所需的網路。移除未使用的資產。為團隊成員提供滿足其需求的裝置，同時將對永續性的影響降至最低。

最佳實務

- [SUS02-BP01 動態擴展工作負載基礎架構](#)
- [SUS02-BP02 讓 SLA 符合永續性目標](#)
- [SUS02-BP03 停止建立和維護未使用的資產](#)
- [SUS02-BP04 根據使用者的聯網要求優化其工作負載的地理位置](#)
- [SUS02-BP05 為執行的活動優化團隊成員資源](#)
- [SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化](#)

SUS02-BP01 動態擴展工作負載基礎架構

利用雲端的彈性動態擴展您的基礎架構，以達到雲端資源的供需平衡，避免工作負載出現過度佈建的容量。

常見的反模式：

- 您不隨著使用者負載擴展基礎架構。
- 您一律手動擴展基礎架構。
- 您在擴展事件之後維持增加容量，而不是縮減規模。

建立此最佳實務的優勢：設定並測試工作負載彈性有助於有效達到雲端資源的供需平衡，並避免過度佈建的容量。您可以利用雲端中的彈性，在需求尖峰期間或之後自動擴展容量，以確保您使用的資源數量正好足以滿足業務所需。

未建立此最佳實務時的風險暴露等級：中

實作指引

雲端提供的彈性可透過各種機制來動態擴展或減少資源，以滿足需求的變化。平衡供需關係可將工作負載受到的影響降到最低。

需求可為固定或可變，需要指標和自動化以確保該項管理不致成為繁重的工作。應用程式可藉由修改執行個體大小進行垂直調整 (縱向擴展或縮減規模)、藉由修改執行個體數目進行水平調整 (縮減或橫向擴展)，或進行兩者的合併調整。

您可以使用多種不同的方法達到資源的供需平衡。

- 目標追蹤法：監控您的擴展指標，並視需要自動增加或減少容量。
- 預測擴展：縮減每日和每週趨勢的預期。
- 排程法：根據可預測的負載變化設定您自己的擴展排程。
- 服務擴展：挑選按設計原本就會擴展的服務 (例如無伺服器)，或提供自動擴展功能。

辨別使用率低或無使用率的時期，並調整資源規模以移除過剩容量、提高效率。

實作步驟

- 彈性會比對您擁有的資源供應與這些資源的需求。執行個體、容器和函數提供了彈性機制，可與自動擴展功能結合使用，或是作為服務功能提供。AWS 提供了多種自動擴展機制，以確保工作負載可在使用者負載較低時迅速輕易地縮減規模。以下是自動擴展機制的幾個範例：

Auto scaling mechanism	Where to use
Amazon EC2 Auto Scaling	用來確認您擁有正確數量的 Amazon EC2 執行個體可處理應用程式的使用者負載。
Application Auto Scaling	用來自動將個別 AWS 服務的資源擴展到 Amazon EC2 以外，例如 Lambda 函數或 Amazon Elastic Container Service (Amazon ECS) 服務。
Kubernetes Cluster Autoscaler	用來自動擴展 AWS 上的 Kubernetes 叢集。

- 擴展常與 Amazon EC2 執行個體或 AWS Lambda 函數等服務一起討論。請考慮設定非運算服務 (例如 [Amazon DynamoDB](#) 讀取和寫入容量單位或 [Amazon Kinesis Data Streams](#) 碎片) 以符合需求。
- 確認會對要部署的工作負載類型驗證擴充或縮減規模的指標。如果您要部署影片轉碼應用程式，則預期為 100% CPU 使用率，且不應做為您的主要指標。您可以將 [自訂指標](#) (例如記憶體使用率) 用於擴展政策 (如有必要)。若要選擇正確的指標，請考量 Amazon EC2 的下列指引：
 - 指標應為有效的使用率指標，並說明執行個體的忙碌程度。

- 指標值必須根據 Auto Scaling 群組中的執行個體數量按比例增加或減少。
- 對於 Auto Scaling 群組請使用[動態擴展](#)，而非[手動擴展](#)。我們也建議您在動態擴展中使用[目標追蹤擴展政策](#)。
- 確認工作負載部署可處理橫向擴展和縮減事件。建立縮減事件的測試案例，以確認工作負載的行為符合預期，且不會對使用者體驗造成影響 (例如失去黏性工作階段)。您可以使用[活動歷史](#)來驗證 Auto Scaling 群組的擴展活動。
- 評估工作負載以取得可預測模式，並在預計發生預測中的變化和隨需規劃變化時主動擴展。透過預測擴展，可以避免過度佈建容量的需求。如需詳細資訊，請參閱[Amazon EC2 Auto Scaling 的預測擴展](#)。

資源

相關文件：

- [Amazon EC2 Auto Scaling 入門](#)
- [EC2 的預測擴展，採用機器學習技術](#)
- [使用 Amazon OpenSearch Service、Amazon Data Firehose 和 Kibana 分析使用者行為](#)
- [什麼是 Amazon CloudWatch？](#)
- [在 Amazon RDS 上使用 Performance Insights 監控資料庫負載](#)
- [介紹對於 Amazon EC2 Auto Scaling 預測擴展的原生支援](#)
- [介紹 Karpenter - 一個開放原始碼的高效能 Kubernetes Cluster Autoscaler](#)
- [深入探討 Amazon ECS 叢集 Auto Scaling](#)

相關影片：

- [打造兼具成本、能源和資源優勢的運算環境](#)
- [更好、更快、更便宜的運算：成本優化 Amazon EC2 \(CMP202-R1\)](#)

相關範例：

- [實驗室：Amazon EC2 Auto Scaling 群組範例](#)
- [實驗室：使用 Karpenter 實作自動擴展](#)

SUS02-BP02 讓 SLA 符合永續性目標

根據您的永續性目標審查並優化工作負載的服務水準協議 (SLA)，例如可用性或資料保留期，以盡可能減少支援工作負載所需的資源，同時繼續滿足商業需求。

常見的反模式：

- 工作負載 SLA 不明或語意不清。
- 您僅針對可用性和效能定義 SLA。
- 您對所有的工作負載使用相同的設計模式 (例如多可用區域架構)。

建立此最佳實務的效益：讓 SLA 符合永續性目標，可達到最佳資源用量並滿足商業需求。

未建立此最佳實務時的風險暴露等級：低

實作指引

SLA 會定義雲端工作負載應有的服務水準，例如回應時間、可用性和資料保留。其影響範圍涵蓋雲端工作負載的架構、資源用量和環境影響。請定期審查 SLA，並做出能大幅降低資源用量的取捨，換取可接受的服務水準降低。

實作步驟

- 定義或重新設計支援永續性目標，同時正好滿足業務要求的 SLA。
- 做出能大幅降低永續性影響的取捨，換取可接受的服務水準降低。
 - 永續性和可靠性：高可用性的工作負載往往會耗用較多資源。
 - 永續性和效能：使用較多資源以提升效能，可能會對環境造成較大的影響。
 - 永續性和安全性：保護過度的工作負載可能會對環境造成較大的影響。
- 使用優先執行業務關鍵功能的設計模式 (例如 [AWS 上的微型服務](#))，對於非關鍵功能允許採用較低的服務水準 (例如回應時間或復原時間目標)。

資源

相關文件：

- [AWS 服務水準協議 \(SLA\)](#)
- [服務水準協議對 SaaS 供應商的重要性](#)

相關影片：

- [提供永續、高效能的架構](#)
- [打造兼具成本、能源和資源優勢的運算環境](#)

SUS02-BP03 停止建立和維護未使用的資產

將您工作負載中未使用的資產除役，以降低支援您個人需求所需的雲端資源數量，並盡可能減少浪費。

常見的反模式：

- 您未分析應用程式是否有冗餘或不再需要的資產。
- 您未移除冗餘或不再需要的資產。

建立此最佳實務的優勢：移除未使用的資產可釋出資源，並改善工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

未使用的資產會耗用儲存空間和運算能力等資源。識別這些資產並將其消除可以釋出這類資源，進而提升雲端架構的效能。定期分析應用程式資產 (例如預先編譯的報告、資料集和靜態影像) 和資產存取模式，找出冗餘、未充分利用和可以除役的目標。移除這類冗餘資產，避免工作負載中的資源浪費。

實作步驟

- 使用監控工具識別不再需要的靜態資產。
- 移除任何資產之前，均應先評估該移除對架構的影響。
- 制定計劃並移除不再需要的資產。
- 合併重疊產生的資產以消除冗餘處理。
- 更新您的應用程式，使其不再產生及儲存不需要的資產。
- 指示第三方停止生產和儲存代表您管理但不再需要的資產。
- 指示第三方合併代表您生產的冗餘資產。
- 定期審查您的工作負載以識別並移除未使用的資產。

資源

相關文件：

- [優化您的 AWS 永續性基礎設施，第 II 部分：儲存](#)
- [如何終止我的 AWS 帳戶 上不再需要的作用中資源？](#)

相關影片：

- [如何檢查我的 AWS 帳戶 上是否有不再需要的作用中資源，並加以移除？](#)

SUS02-BP04 根據使用者的聯網要求優化其工作負載的地理位置

為您的工作負載選取可減少網路流量傳輸距離的區域和服務，並減少支援工作負載所需的整體網路資源。

常見的反模式：

- 您可以根據自身所在位置選取工作負載的區域。
- 您可以將所有工作負載資源合併到單一地理位置。
- 通過現有資料中心的所有流量。

建立此最佳實務的優勢：將工作負載分配到使用者附近的位置，可提供最低的延遲，同時減少網路間的資料移動，並降低環境影響。

未建立此最佳實務時的曝險等級：中

實作指引

AWS 雲端 基礎設施是根據如下的位置選項而建置的：區域、可用區域、放置群組和邊緣節點，例如 [AWS Outposts](#) 和 [AWS Local Zones](#)。這些位置選項負責維護應用程式元件、雲端服務、邊緣網路和內部部署資料中心之間的連線。

分析工作負載中的網路存取模式，以識別如何使用這些雲端位置選項，以及減少網路流量必須輸送的距離。

實作步驟

- 分析您工作負載中的網路存取模式，以識別使用者如何使用您的應用程式。

- 使用監控工具，例如 [Amazon CloudWatch](#) 和 [AWS CloudTrail](#)，收集關於網路活動的資料。
- 分析資料以識別網路存取模式。
- 根據下列關鍵元素，為您的工作負載部署選取區域：
 - 您的永續目標：相關說明請見 [區域選擇](#)。
 - 資料所在位置：對於資料密集型應用程式 (例如大數據和機器學習)，應用程式碼執行時應盡可能接近資料。
 - 使用者所在的位置：對於面向使用者的應用程式，請選擇接近工作負載使用者的一或多個區域。
 - 其他限制：請考量成本和合規性等限制，相關說明請見 [為工作負載選取區域時應考慮的事項](#)。
- 使用本機快取或 [AWS 快取解決方案](#) 取得常用的資產，以提升效能、減少資料移動，以及降低環境影響。

服務	使用時機
Amazon CloudFront	用來快取靜態內容 (例如影像、指令碼和影片) 以及動態內容 (例如 API 回應或 Web 應用程式)。
Amazon ElastiCache	用來快取 Web 應用程式的內容。
DynamoDB Accelerator	用來將記憶體內加速新增至 DynamoDB 資料表。

- 使用可協助您在更接近工作負載使用者的位置執行程式碼的服務：

服務	使用時機
Lambda@Edge	用於在物件未經快取時起始的大量運算作業。
Amazon CloudFront Functions	用於簡單的使用案例，例如可由短期函數起始的 HTTP(s) 請求或回應操作。
AWS IoT Greengrass	用來為連線的裝置執行本機運算、傳訊和資料快取。

- 使用連線共用來支援連線重複使用，減少所需資源。
- 使用不倚賴持續連線和同步更新的分散式資料存放區來實現一致性，以服務區域的人口。

- 以共用動態容量取代預先佈建的靜態網路容量，與其他訂閱者分攤網路容量的永續性影響。

資源

相關文件：

- [優化您的 AWS 永續性基礎架構，第 III 部分：聯網](#)
- [Amazon ElastiCache 文件](#)
- [什麼是 Amazon CloudFront？](#)
- [Amazon CloudFront 主要功能](#)

相關影片：

- [揭密 AWS 上的資料傳輸](#)
- [擴展新一代 Amazon EC2 執行個體的網路效能](#)

相關範例：

- [AWS 聯網研討會](#)
- [永續性架構 - 盡可能減少跨網路的資料移動](#)

SUS02-BP05 為執行的活動優化團隊成員資源

優化提供給團隊成員的資源，以盡量減少對環境永續性的影響，同時支援他們的需求。

常見的反模式：

- 您忽略團隊成員所使用的裝置對雲端應用程式的整體效率產生的影響。
- 您手動管理及更新團隊成員所使用的資源。

建立此最佳實務的優勢：優化團隊成員資源，可為具備雲端功能的應用程式改善整體效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

了解團隊成員用來使用您的服務的資源、其預期生命週期，以及財務和永續性的影響。實作將這些資源優化的策略。例如，在使用率高的可擴展基礎設施上執行複雜的操作 (例如轉譯和編譯)，而不是在使用率低的高功率單一使用者系統上執行。

實作步驟

- 根據使用方式佈建工作站和其他裝置。
- 使用虛擬桌面和應用程式串流來限縮升級與裝置要求。
- 將大量使用處理器或記憶體的任务移至雲端，以運用其彈性。
- 評估程序和系統對裝置生命週期的影響，並選擇在滿足業務需求的同時可將裝置更換需求降至最低的解決方案。
- 為裝置實作遠端管理，以減少必要商務差旅時間。
 - [AWS Systems Manager Fleet Manager](#) 是一種整合式使用者介面 (UI) 體驗，可協助您從遠端管理在 AWS 內部部署環境執行的節點。

資源

相關文件：

- [什麼是 Amazon WorkSpaces？](#)
- [Amazon WorkSpaces 的成本優化器](#)
- [Amazon AppStream 2.0 文件](#)
- [NICE DCV](#)

相關影片：

- [在 AWS 上管理 Amazon WorkSpaces 的成本](#)

SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化

緩衝和調節可讓需求曲線趨於扁平化，並減少您的工作負載所需的已佈建容量。

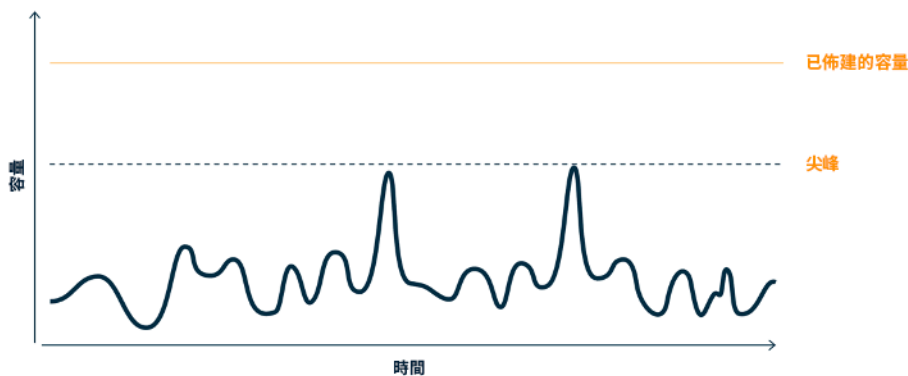
常見的反模式：

- 您非必要地立即處理用戶端要求。
- 您未分析用戶端要求的需求。

建立此最佳實務的優勢：讓需求曲線趨於扁平化，可減少工作負載所需的已佈建容量。減少已佈建的容量意味著較低的能源耗用量和環境影響。

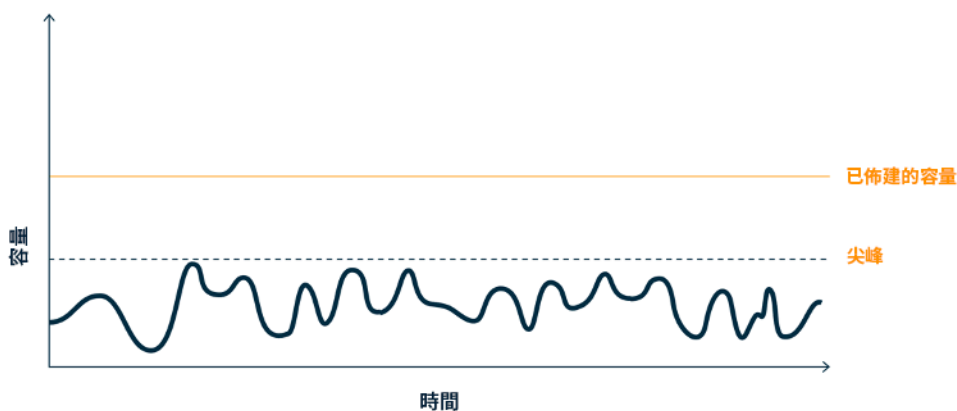
未建立此最佳實務時的風險暴露等級：低

使工作負載需求曲線扁平化，有助於減少工作負載所需的已佈建容量，以及降低對環境造成的影響。假設某个工作負載的需求曲線如下圖所示。此工作負載有兩個尖峰，為了處理這些尖峰，已佈建了資源容量 (以橙色線顯示)。用於此工作負載的資源和能源並非由需求曲線底下的區域表示，而是已佈建的容量底下的區域，因為這兩個尖峰必須用已佈建的容量處理。



需求曲線圖，內含兩個需要大量已佈建容量的相異尖峰。

您可以使用緩衝或調節來修正需求曲線，並使尖峰趨緩，意即減少已佈建的容量和耗用的能源。在用戶端可執行重試時實作調節。實作緩衝機制以儲存要求，並將處理的時間往後延遲。



調節對需求曲線和已佈建容量的影響。

實作步驟

- 分析用戶端要求以判斷如何予以回應。應考量的問題包括：
 - 此要求是否可進行非同步處理？
 - 用戶端是否有重試能力？
- 如果用戶端有重試功能，您可以實作調節，以告知來源若目前無法處理要求，則應稍後再試。
 - 您可以使用 [Amazon API Gateway](#) 來實作調節。
- 針對無法執行重試的用戶端，必須實作緩衝區使需求曲線扁平化。緩衝會延遲要求處理，讓以不同速率執行的應用程式能夠有效地通訊。緩衝為主方法使用佇列或串流來接受生產者傳出的訊息。消費者可讀取訊息並進行處理，允許以符合取用者業務要求的速度運作訊息。
 - [Amazon Simple Queue Service \(Amazon SQS\)](#) 是一個受管服務，可提供佇列，允許單一取用者讀取個別訊息。
 - [Amazon Kinesis](#) 可提供串流，允許許多取用者讀取相同訊息。
- 分析整體需求、變更率及所需的回應時間，以適當調整所需的調節或緩衝區大小。

資源

相關文件：

- [Amazon SQS 入門](#)
- [使用佇列和訊息進行應用程式整合](#)

相關影片：

- [為分散式應用程式選擇適當的傳訊服務](#)

軟體和架構

實施可執行負載順暢並保持已部署資源一致高使用率的模式，將資源消耗降至最低。由於使用者行為隨時間改變，元件可能會因缺乏使用而閒置。修改模式和架構來整合未充分利用的元件，提高整體使用率。淘汰不再需要的元件。了解工作負載元件的效能，並最佳化消耗最多資源的元件。注意客戶用來存取服務的裝置，並實施可最小化裝置升級需求的模式。

最佳實務

- [SUS03-BP01 最佳化非同步與排程任務的軟體和架構](#)
- [SUS03-BP02 移除或重構使用量低或完全未使用的工作負載元件](#)
- [SUS03-BP03 優化程式碼中耗用最多時間或資源的部分](#)
- [SUS03-BP04 優化對裝置和設備的影響](#)
- [SUS03-BP05 使用最能支援資料存取和儲存模式的軟體模式和架構](#)

SUS03-BP01 最佳化非同步與排程任務的軟體和架構

使用有效率的軟體和架構模式 (例如佇列驅動)，讓所部署的資源一直保持高使用率。

常見的反模式：

- 在雲端工作負載中過度佈建資源以滿足未預料到的突增需求。
- 您的架構未透過傳訊元件將非同步訊息的傳送者與接受者分離。

建立此最佳實務的優勢：

- 有效率的軟體和架構模式可盡量減少工作負載中的未使用資源，並改善整體效率。
- 您可以將非同步訊息的處理與接收分開擴展。
- 透過傳訊元件，可用性要求會比較寬鬆，不用太多資源即可滿足。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用有效率的架構模式 (例如[事件驅動架構](#))，以便能平均地使用元件，並盡量避免工作負載過度佈建。使用有效率的架構模式可盡量地讓閒置資源不會因為需求隨時間發生變化而有乏人問津的情形。

了解工作負載元件的要求，並採用能夠提升整體資源使用率的架構模式。淘汰不再需要的元件。

實作步驟

- 分析工作負載需求以判斷如何回應。
- 如果請求或作業不需要同步回應，請使用佇列驅動的架構和 Auto Scaling 工作節點，以將使用率最大化。以下是您可能會考慮使用佇列驅動架構的一些範例：

Queuing mechanism	Description
AWS Batch 作業佇列	AWS Batch 作業會提交至作業佇列並停留其中，直到能夠排定在運算環境中執行為止。
Amazon Simple Queue Service 和 Amazon EC2 Spot 執行個體	搭配使用 Amazon SQS 與 Spot 執行個體即可建置能容錯且有效率的架構。

- 對於可以隨時處理的佇列或作業，請使用排程機制來批次處理作業，以提升效率。以下是 AWS 上排程機制的幾個範例：

Scheduling mechanism	Description
Amazon EventBridge 排程器	這是 Amazon EventBridge 的一項功能，可讓您大規模地建立、執行和管理已排定的任務。
AWS Glue 時間型排程	在 AWS Glue 中定義爬蟲程式和作業的時間型排程。
Amazon Elastic Container Service (Amazon ECS) 排定的任務	Amazon ECS 支援建立排定的任務排定的任務會使用 Amazon EventBridge 規則，依排程執行任務或是在 EventBridge 事件的回應中執行任務。
Instance Scheduler	設定 Amazon EC2 和 Amazon Relational Database Service 執行個體的開始和停止排程。

- 如果您的架構中使用輪詢和 Webhook 機制，請將其更換為事件。使用[事件驅動的架構](#)可建置高效率的工作負載。
- 利用 [AWS 上的無伺服器](#)來消除過度佈建的基礎設施。
- 將架構的個別元件調整為適當大小，避免閒置資源等待輸入。

資源

相關文件：

- [什麼是 Amazon Simple Queue Service ?](#)
- [什麼是 Amazon MQ ?](#)
- [根據 Amazon SQS 擴展](#)
- [什麼是 AWS Step Functions ?](#)
- [什麼是 AWS Lambda ?](#)
- [搭配 Amazon SQS 使用 AWS Lambda](#)
- [什麼是 Amazon EventBridge ?](#)

相關影片：

- [移至事件驅動型架構](#)

SUS03-BP02 移除或重構使用量低或完全未使用的工作負載元件

移除未使用且不再需要的元件，並重構使用率低的元件，以盡量避免工作負載中的浪費。

常見的反模式：

- 您未定期檢查個別工作負載元件的使用率水準。
- 您未查看並分析 AWS 適當調整大小的工具 (例如 [AWS Compute Optimizer](#)) 所提供的建議。

建立此最佳實務的優勢：移除未使用的元件可盡量避免浪費，並改善雲端工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

審查您的工作負載以識別閒置或未使用的元件。有一個迭代改進程序可由需求的變更或新雲端服務的發行來觸發。例如，[AWS Lambda](#) 函數執行時間的大幅下降可能意味著必須降低記憶體大小。此外，隨著 AWS 發行新的服務和功能，工作負載的最佳服務與架構可能會改變。

持續監控工作負載活動，並找機會改善個別元件的使用率水準。藉由移除閒置元件和執行適當調整大小的活動，您將可用最少的雲端資源達到業務要求。

實作步驟

- 監控及擷取工作負載關鍵元件的使用率指標 (例如 [Amazon CloudWatch 指標](#) 中的 CPU 使用率、記憶體使用率或網路輸送量)。

- 對於穩定的工作負載，請定期檢查 AWS 適當調整大小的工具 (例如 [AWS Compute Optimizer](#))，以識別閒置、未使用或未充分利用的元件。
- 對於暫時性工作負載，請評估使用率指標以識別閒置、未使用或未充分利用的元件。
- 淘汰不再需要的元件和相關聯的資產 (例如 Amazon ECR 映像)。
- 重構或整合未充分利用的元件與其他資源，以提高利用效率。例如，您可將多個小資料庫佈建至單一 [Amazon RDS](#) 資料庫執行個體，而不要在未充分利用的個別執行個體上執行資料庫。
- 了解[您的工作量佈建以完成工作單位的資源](#)。

資源

相關文件：

- [AWS Trusted Advisor](#)
- [什麼是 Amazon CloudWatch？](#)
- [自動清理 Amazon ECR 中未使用的映像](#)

相關範例：

- [Well-Architected 實驗室：使用 AWS Compute Optimizer 適當調整大小](#)
- [Well-Architected 實驗室 - 優化硬體模式和觀察永續性 KPI](#)

SUS03-BP03 優化程式碼中耗用最多時間或資源的部分

優化您的架構不同元件中執行的程式碼，將資源使用量降至最低，同時發揮最大效能。

常見的反模式：

- 您略過資源用量的程式碼優化。
- 您通常藉由增加資源來因應效能問題。
- 您的程式碼審查和開發程序未追蹤效能變更。

建立此最佳實務的優勢：使用有效率的程式碼可將資源用量壓到最低，並改善效能。

未建立此最佳實務時的曝險等級：中

實作指引

請務必檢查各個功能領域 (包括雲端架構應用程式的程式碼)，以優化其資源用量和效能。持續監控您的工作負載在建置環境和生產環境中的效能，並找機會改進資源用量特別高的程式碼片段。採用定期審查程序，在您的程式碼內識別低效使用資源的錯誤或反模式。使用簡單有效的演算法為您的使用案例產生相同結果。

實作步驟

- 在擬定工作負載時採用自動化程式碼審查程序，以改善品質並識別錯誤和反模式。
 - [使用 Amazon CodeGuru Reviewer 自動進行程式碼審查](#)
 - [使用 Amazon CodeGuru 偵測並行錯誤](#)
 - [使用 Amazon CodeGuru 提升 Python 應用程式的程式碼品質](#)
- 在您執行工作負載時監控資源，以識別每個工作單元中具有高資源需求的元件，作為程式碼審查目標。
- 在進行程式碼審查時，使用程式碼分析工具來識別程式碼中耗用最多時間或資源的部分，作為優化目標。
 - [透過 Amazon CodeGuru Profiler 降低組織的碳足跡](#)
 - [透過 Amazon CodeGuru Profiler 了解 Java 應用程式中的記憶體用量](#)
 - [透過 Amazon CodeGuru Profiler 改善客戶體驗並降低成本](#)
- 使用針對工作負載最高效率的作業系統和程式設計語言。如需關於高能效程式設計語言 (包括 Rust) 的詳細資料，請參閱 [Rust 的永續性](#)。
- 將需要大量運算資源的演算法取代為會產生相同結果、但更簡單有效率的版本。
- 移除不必要程式碼，例如排序和格式化。

資源

相關文件：

- [什麼是 Amazon CodeGuru Profiler？](#)
- [FPGA 執行個體](#)
- [在 AWS 上建立的工具中的 AWS SDK](#)

相關影片：

- [使用 Amazon CodeGuru Profiler 改善程式碼效率](#)
- [使用 Amazon CodeGuru 自動進程式碼審查和應用程式效能建議](#)

SUS03-BP04 優化對裝置和設備的影響

了解您的架構中使用的裝置和設備，並使用策略降低其用量。這樣可以盡量減輕對雲端工作負載的整體環境影響。

常見的反模式：

- 您忽略了客戶使用的裝置所受到的環境影響。
- 您手動管理及更新客戶所使用的資源。

建立此最佳實務的優勢：實作為客戶裝置優化的軟體模式和功能，可降低雲端工作負載的整體環境影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

實作為客戶裝置優化的軟體模式和功能，可透過數種方式降低環境影響：

- 實作具回溯相容性的新功能，可減少硬體更換的數量。
- 將應用程式優化以有效執行於裝置上，有助於降低其能源耗用量及延長電池使用壽命 (若是由電池供電)。
- 優化裝置的應用程式也可減少網路上的資料傳輸。

了解客戶您的架構中使用的裝置和設備、其預期生命週期，以及更換這些元件的影響。實作適當的軟體模式和功能，以盡可能減少裝置能源耗用量，以及客戶更換裝置和手動加以升級的需求。

實作步驟

- 清查您的架構中使用的裝置。裝置可以是行動裝置、平板裝置、IOT 裝置、智慧電燈，甚至是工廠的智慧裝置。
- 優化在裝置上執行的應用程式：
 - 採用在背景執行任務之類的策略來降低能源耗用量。

- 在建置承載時考慮網路頻寬和延遲，並實施可協助應用程式在低頻寬、高延遲連結上良好運作的功能。
- 將承載和檔案轉換為裝置所需的優化格式。例如，您可以使用 [Amazon Elastic Transcoder](#) 或 [AWS Elemental MediaConvert](#) 將較大的高品質數位媒體檔案轉換為使用者可在行動裝置、平板電腦、Web 瀏覽器和聯網電視上播放的格式。
- 在伺服器端執行需要大量運算的活動 (例如影像渲染)，或使用應用程式串流來改善舊裝置的使用者體驗。
- 對輸出進行分段和分頁，特別是對於互動式工作階段，以管理承載並限制本機儲存要求。
- 使用自動化空中 (OTA) 機制將更新部署至一或多個裝置。
 - 您可以使用 [CI/CD 管道](#) 更新行動應用程式。
 - 您可以使用 [AWS IoT Device Management](#) 從遠端大規模管理連網裝置。
- 若要測試新功能和更新，請使用具有代表性硬體集的受管 Device Farm，並迭代開發以最大化支援的裝置。如需詳細資訊，請參閱 [SUS06-BP04 使用受管 Device Farm 進行測試](#)。

資源

相關文件：

- [什麼是 AWS Device Farm ?](#)
- [Amazon AppStream 2.0 文件](#)
- [NICE DCV](#)
- [在執行 FreeRTOS 的裝置上更新韌體的 OTA 教學](#)

相關影片：

- [AWS Device Farm 簡介](#)

SUS03-BP05 使用最能支援資料存取和儲存模式的軟體模式和架構

了解資料在工作負載中的使用方式、使用者的使用方式、傳輸方式以及儲存方式。使用最能支援資料存取和儲存的軟體模式與架構，以盡可能減少支援工作負載所需的運算、聯網和儲存資源。

常見的反模式：

- 您假設所有工作負載具有類似的資料儲存和存取模式。

- 您只使用一個存儲層 – 假設所有工作負載都適合該層。
- 您假設資料存取模式不會隨著時間改變。
- 您的架構支援潛在的高資料存取爆量，這會導致資源在大部分的時間處於閒置狀態。

建立此最佳實務的優勢：根據資料存取和儲存模式選取及優化您的架構，有助於降低開發複雜性並提升整體使用率。了解何時使用全域表、資料分割和快取將協助您降低營運負擔，並根據您的工作負載需求進行擴展。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用與您的資料特性和存取模式最相符的軟體和架構模式。例如，使用 [AWS 上的現代資料架構](#) (可讓您使用針對個人獨特分析使用案例而優化的專用服務)。這些架構模式可實現高效資料處理以及降低資源用量。

實作步驟

- 分析您的資料特性和存取模式，以識別雲端資源的正確組態。應考量的重要特性包括：
 - 資料類型：結構化、半結構化、非結構化
 - 資料成長：有界限、無界限
 - 資料耐用性：持續性、暫時性、臨時
 - 存取模式：讀取或寫入、更新頻率、尖峰或一致
- 使用最能支援資料存取和儲存模式的架構模式。
 - [開始建構吧！現代資料架構](#)
 - [AWS 上的資料庫：使用正確的工具完成任務](#)
- 利用可原生處理壓縮資料的技術。
- 使用專用[分析服務](#)進行架構中的資料處理。
- 使用最能支援您主導查詢模式的資料庫引擎。管理您的資料庫索引，以確保高效率的查詢執行。如需詳細資訊，請參閱 [AWS 資料庫](#)。
- 選取可減少架構中網路容量消耗的網路通訊協定。

資源

相關文件：

- [Athena 壓縮支援檔案格式](#)
- [使用 Amazon Redshift 從單欄資料格式複製](#)
- [在 Firehose 中轉換您的輸入記錄格式](#)
- [AWS Glue 中 ETL 輸入和輸出的格式選項](#)
- [轉換為單欄格式，提高 Amazon Athena 的查詢效能](#)
- [使用 Amazon Redshift 從 Amazon S3 載入壓縮的資料檔案](#)
- [在 Amazon Aurora 上使用 Performance Insights 監控資料庫負載](#)
- [在 Amazon RDS 上使用 Performance Insights 監控資料庫負載](#)
- [Amazon S3 Intelligent-Tiering 儲存類別](#)

相關影片：

- [在 AWS 上建置現代資料架構](#)

資料管理

實作資料管理實務來減少支援工作負載所需的佈建儲存，以及使用它的必要資源。了解您的資料，並使用最能支援資料業務價值及其使用方式的儲存技術和組態。當需求減少時，將資料循環到效率較高、效能較低的儲存，並刪除不再需要的資料。

最佳實務

- [SUS04-BP01 實作資料分類政策](#)
- [SUS04-BP02 使用支援資料存取和儲存模式的技術](#)
- [SUS04-BP03 使用政策來管理資料集的生命週期](#)
- [SUS04-BP04 使用彈性和自動化擴充區塊儲存或檔案系統](#)
- [SUS04-BP05 移除不需要或多餘的資料](#)
- [SUS04-BP06 使用共用檔案系統或儲存體存取通用資料](#)
- [SUS04-BP07 盡可能減少跨網路的資料移動](#)
- [SUS04-BP08 僅在難以重新建立時才備份資料](#)

SUS04-BP01 實作資料分類政策

將資料分類以了解其對業務成果的關鍵性，並選擇適當的 節能儲存層來儲存資料。

常見的反模式：

- 您未以正在處理或已儲存的類似特性來識別資料資產 (例如敏感性、業務關鍵性或法規要求)。
- 您未實作資料目錄以清查資料資產。

建立此最佳實務的優勢：實作資料分類政策，可讓您確認最節能的資料儲存層。

未建立此最佳實務時的風險暴露等級：中

實作指引

資料分類的工作之一，是識別正在處理的資料類型，以及由組織擁有或操作的資訊系統中儲存的資料類型。此外也須確認資料的關鍵性，以及資料損毀、遺失或誤用可能造成的影響。

若要實作資料分類政策，請從資料的情境使用採取逆向思維，並建立適當的分類機制，將指定資料集的關鍵性程度納入組織操作的考量中。

實作步驟

- 對您工作負載現有的各種資料類型執行清查。
 - 如需關於資料分類類別的詳細資料，請參閱[資料分類白皮書](#)。
- 根據組織面臨的風險，判定資料的關鍵性、機密性、完整性和可用性。使用這些要求，將資料分組為您採用的其中一個資料分類層。
 - 範例請見[分類資料及保護新創公司的四個簡單步驟](#)。
- 定期稽核您的環境是否有未標記和未分類的資料，並對資料進行適當的分類和標記。
 - 範例請見 [AWS Glue 中的資料目錄和編目程式](#)。
- 建立提供稽核和管控能力的資料目錄。
- 確認並記載每個資料類別的處理程序。
- 使用自動化功能持續稽核環境以識別未標記和未分類的資料，並對資料進行適當的分類和標記。

資源

相關文件：

- [使用 AWS 雲端 以支援資料分類](#)
- [標記來自 AWS Organizations 的政策](#)

相關影片：

- [在 AWS 上實現敏捷性與資料管控](#)

SUS04-BP02 使用支援資料存取和儲存模式的技術

使用最能支援您的資料存取和儲存方式的儲存技術，以在支援工作負載的同時，也將佈建的資源降至最低。

常見的反模式：

- 您假設所有工作負載具有類似的資料儲存和存取模式。
- 您只使用一個存儲層 – 假設所有工作負載都適合該層。
- 您假設資料存取模式不會隨著時間改變。

建立此最佳實務的優勢：根據資料存取和儲存模式來選取及優化您的儲存技術，可協助您降低達成商業需求所需的雲端資源，並改善雲端工作負載的整體效率。

未建立此最佳實務時的曝險等級：低

實作指引

選擇最適合您的存取模式的儲存解決方案，或者考慮變更存取模式，以符合儲存解決方案，從而達到最大的效能效率。

- 評估您的資料特性和存取模式，以收集儲存需求的重要特性。應考量的重要特性包括：
 - 資料類型：結構化、半結構化、非結構化
 - 資料成長：有界限、無界限
 - 資料耐用性：持續性、暫時性、臨時
 - 存取模式：讀取或寫入、頻率、尖峰或一致
- 將資料遷移至支援您的資料特性和存取模式的適當儲存技術。以下提供 AWS 儲存技術及其重要特性的一些範例：

類型	技術	重要特性
物件儲存	Amazon S3	一項物件儲存服務，具有不受限的可擴展性、高可用性，

類型	技術	重要特性
		以及多個可存取性選項。對 Amazon S3 輸入和存取物件時，可以使用相關服務 (例如 Transfer Acceleration 或 存取點)，來支援您的位置、安全需求和存取模式。
封存儲存	Amazon S3 Glacier	針對資料封存而建置的 Amazon S3 儲存類別。
共用檔案系統	Amazon Elastic File System (Amazon EFS)	可供多種類型的運算解決方案存取的可掛載檔案系統。Amazon EFS 會自動增長及縮減儲存體，藉以進行效能優化而提供一致的低延遲。
共用檔案系統	Amazon FSx	建置於最新的 AWS 運算解決方案之上，用以支援四個常用的檔案系統：NetApp ONTAP、OpenZFS、Windows File Server 和 Lustre。Amazon FSx 延遲、輸送量和 IOPS 會隨著檔案系統而不同，當您為工作負載需求選取適當的檔案系統時，應予以考量。
區塊儲存	Amazon Elastic Block Store (Amazon EBS)	針對 Amazon Elastic Compute Cloud (Amazon EC2) 而設計的可擴展、高效能區塊儲存服務。Amazon EBS 包含支援 SSD 的儲存系統 (用於交易、IOPS 密集工作負載)，以及支援 HDD 的儲存系統 (用於輸送量密集工作負載)。

類型	技術	重要特性
關聯式資料庫	Amazon Aurora 、 Amazon RDS 、 Amazon Redshift	旨在支援 ACID (單元性、一致性、隔離行為、持續性) 交易，並維護參考完整性和強大的資料一致性。許多傳統應用程式、企業資源規劃 (ERP)、客戶關係管理 (CRM) 和電子商務系統都使用關聯式資料庫來儲存資料。
鍵值資料庫	Amazon DynamoDB	已針對常見的存取模式進行優化，通常用於儲存和擷取大量資料。高流量 Web 應用程式、電子商務系統和遊戲應用程式是鍵值資料庫的典型使用案例。

- 對於固定大小的儲存系統 (例如 Amazon EBS 或 Amazon FSx)，請監控可用儲存空間，並且在接近閾值時自動執行儲存空間分配。您可以利用 Amazon CloudWatch 來收集及分析下列項目的不同指標：[Amazon EBS](#) 和 [Amazon FSx](#)。
- Amazon S3 儲存類別可設定於物件層級，且單一儲存貯體可包含儲存在所有儲存類別間的物件。
- 您也可以使用 Amazon S3 生命週期政策在儲存類別之間自動轉換物件或移除資料，而無須進行任何應用程式變更。在考量這些儲存機制時，您通常需要在資源效率、存取延遲與可靠性之間做出取捨。

資源

相關文件：

- [Amazon EBS 磁碟區類型](#)
- [Amazon EC2 執行個體儲存體](#)
- [Amazon S3 Intelligent-Tiering](#)
- [Amazon EBS I/O 特性](#)
- [使用 Amazon S3 儲存類別](#)
- [什麼是 Amazon S3 Glacier？](#)

相關影片：

- [AWS 上資料湖的架構模式](#)
- [深入探討 Amazon EBS \(STG303-R1\)](#)
- [使用 Amazon S3 優化儲存效能 \(STG343\)](#)
- [在 AWS 上建置現代化資料架構](#)

相關範例：

- [Amazon EFS CSI 驅動程式](#)
- [Amazon EBS CSI 驅動程式](#)
- [Amazon EFS 公程式](#)
- [Amazon EBS 自動擴展](#)
- [Amazon S3 範例](#)

SUS04-BP03 使用政策來管理資料集的生命週期

管理所有資料的生命週期並自動執行刪除，將工作負載所需的儲存總量降至最低。

常見的反模式：

- 您手動刪除資料。
- 您未刪除任何工作負載資料。
- 您未根據資料的保留和存取要求，將資料移至更節能的儲存層。

建立此最佳實務的優勢：使用資料生命週期政策可確保工作負載中的資料會以有效率的方式存取和保留。

未建立此最佳實務時的風險暴露等級：中

實作指引

資料集在其生命週期內，通常會有不同的保留和存取要求。例如，應用程式可能需要在一段時間內頻繁存取某些資料集。這段時間過後，便不會頻繁存取這些資料集。

為了在資料集的完整生命週期內有效率地管理資料集，請設定生命週期政策，也就是定義了資料集處理方式的規則。

有了生命週期組態規則後，便能指示特定儲存服務將資料集轉移至更節能的儲存層、將其封存，或加以刪除。

實作步驟

- [對工作負載內的資料集進行分類。](#)
- 定義每個資料類別的處理程序。Define handling procedures for each data class.
- 設定自動生命週期政策以強制執行生命週期規則。下面幾個範例會說明如何為不同的 AWS 儲存服務設定自動化的生命週期政策：

Storage service	How to set automated lifecycle policies
Amazon S3	您可以使用 Amazon S3 生命週期 在物件的完整生命週期內管理物件。如果存取模式不明、會變化或是無法預測，則可以使用 Amazon S3 Intelligent-Tiering ，讓其監控存取模式，並自動將未存取的物件移至成本較低的存取層。您可以利用 Amazon S3 Storage Lens 指標來識別生命週期管理中的最佳化機會和落差。
Amazon Elastic Block Store	您可以使用 Amazon Data Lifecycle Manager 來自動建立、保留和刪除 Amazon EBS 快照與 Amazon EBS 支援的 AMI。
Amazon Elastic File System	Amazon EFS 生命週期管理 會自動管理檔案系統的檔案儲存。
Amazon Elastic Container Registry	Amazon ECR 生命週期政策 會根據存留時間長短或計數讓映像過期，藉此自動清理容器映像。
AWS Elemental MediaStore	您可以使用 物件生命週期政策 來管控物件儲存在 MediaStore 容器內的時間長度。

- 請刪除已超過保留期間的未使用磁碟區、快照和資料。利用原生服務功能 (例如 Amazon DynamoDB Time To Live 或 Amazon CloudWatch 日誌保留) 來執行刪除作業。
- 根據生命週期規則，在適用的情況下彙總和壓縮資料。

資源

相關文件：

- [使用 Amazon S3 儲存類別分析將 Amazon S3 生命週期規則最佳化](#)
- [使用 AWS Config 規則 評估資源](#)

相關影片：

- [使用 Amazon S3 生命週期來簡化資料生命週期並將儲存成本最佳化](#)
- [使用 Amazon S3 Storage Lens 降低儲存成本](#)

SUS04-BP04 使用彈性和自動化擴充區塊儲存或檔案系統

隨著資料的增長使用彈性和自動化擴充區塊儲存或檔案系統，以盡可能縮小整體的已佈建儲存。

常見的反模式：

- 您為了日後的需求購買大型區塊儲存或檔案系統。
- 您過度佈建了檔案系統的每秒輸入/輸出作業數 (IOPS)。
- 您未監控資料磁碟區的使用率。

建立此最佳實務的優勢：盡可能減少儲存系統的過度佈建可減少閒置資源，並改善工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用適合工作負載的大小分配、輸送量和延遲，建立區塊儲存和檔案系統。隨著資料的增長使用彈性和自動化擴充區塊儲存或檔案系統，而無須過度佈建這些儲存服務。

實作步驟

- 對於固定大小的儲存體 (例如 [Amazon EBS](#))，請確認監控使用的儲存量佔整體儲存大小的比例，並在達到閾值時建立自動化 (如可能) 以增加儲存大小。

- 使用彈性磁碟區和受管區塊資料服務，以在持久性資料增長時自動分配額外的儲存空間。例如，您可以使用 [Amazon EBS 彈性磁碟區](#) 來變更磁碟區大小、磁碟區類型，或調整 Amazon EBS 磁碟區的效能。
- 為您的檔案系統選擇適當的儲存類別、效能模式和輸送量模式以因應商業需求 (勿過量)。
 - [Amazon EFS 效能](#)
 - [Linux 執行個體上的 Amazon EBS 磁碟區效能](#)
- 設定資料磁碟區的目標使用率水準，並調整超出預期範圍的磁碟區大小。
- 根據資料適當調整唯讀磁碟區的大小。
- 將資料遷移到物件存放區，避免從區塊儲存的固定磁碟區大小佈建多餘容量。
- 定期審查彈性磁碟區和檔案系統以終止閒置磁碟區，並縮減過度佈建的資源以符合目前的資料大小。

資源

相關文件：

- [Amazon FSx 文件](#)
- [什麼是 Amazon Elastic File System ?](#)

相關影片：

- [深入探討 Amazon EBS 彈性磁碟區](#)
- [有助於提升效能和節省成本的 Amazon EBS 與快照優化策略](#)
- [使用最佳實務優化 Amazon EFS 的成本與效能](#)

SUS04-BP05 移除不需要或多餘的資料

移除不需要或多餘的資料，以盡量降低儲存資料集時所需的儲存資源。

常見的反模式：

- 您複製可以輕鬆取得或建立的資料。
- 您備份所有資料，而不考慮該資料是否重要。
- 您只會不定期地刪除資料、在發生營運事件時刪除資料，或完全不刪除資料。
- 您重複儲存資料，而不理會儲存服務的耐用性。
- 您在沒有任何商務理由的情況下啟用 Amazon S3 版本控制。

建立此最佳實務的優勢：移除不需要的資料會降低工作負載所需的儲存大小，以及工作負載環境所受到的影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

請勿儲存您不需要的資料。請自動刪除不需要的資料。使用會在檔案層級和區塊層級刪除重複資料的技術。利用服務原生的資料複寫和備援功能。

實作步驟

- 評估您是否可以藉由使用 [AWS Data Exchange](#) 和 [AWS 上的開放資料登錄檔](#) 中現有的公開提供的資料集，以避免儲存資料。
- 使用可在區塊和物件層級刪除重複資料的機制。下面幾個範例會說明如何在 AWS 上刪除重複資料：

Storage service	Deduplication mechanism
Amazon S3	使用 AWS Lake Formation FindMatches ，透過新的 FindMatches ML Transform 來尋找整個資料集內的相符記錄。
Amazon FSx	在適用於 Windows 的 Amazon FSx 上啟用 重複資料刪除 。
Amazon Elastic Block Store 快照	快照是增量備份，這表示只會儲存最近一次快照後裝置上發生變更的區塊。

- 分析資料存取以識別不需要的資料。將生命週期政策自動化。利用原生服務功能 (例如 [Amazon DynamoDB Time To Live](#)、[Amazon S3 Lifecycle](#) 或 [Amazon CloudWatch 日誌保留](#)) 來執行刪除作業。
- 使用 AWS 上的資料虛擬化功能將資料留在其來源上，並避免資料重複。
 - [AWS 上的雲端原生資料虛擬化](#)
 - [實驗室：使用 Amazon Redshift 資料共用來最佳化資料模式](#)
- 使用可以進行增量備份的備份計數。
- 利用 [Amazon S3](#) 的耐久性和 [Amazon EBS 的複寫功能](#) 來滿足耐久性目標，而非利用自我管理的技術 (例如獨立硬碟冗餘陣列 (RAID))。
- 集中日誌和追蹤資料、刪除重複的日誌項目，並建立根據需要微調詳細程度的機制。

- 僅在合理的情況下才預先填入快取。
- 建立快取監控和自動化，據以調整快取大小。
- 推送工作負載新版本時，從物件存放區和邊緣快取移除過時的部署和資產。

資源

相關文件：

- [變更 CloudWatch Logs 中的日誌資料保留](#)
- [Amazon FSx for Windows File Server 上的重複資料刪除](#)
- [Amazon FSx for ONTAP 的功能，包括重複資料刪除](#)
- [使 Amazon CloudFront 上的檔案無效](#)
- [使用 AWS Backup 來備份和還原 Amazon EFS 檔案系統](#)
- [什麼是 Amazon CloudWatch Logs？](#)
- [在 Amazon RDS 上使用備份](#)

相關影片：

- [使用 ML Transforms for AWS Lake Formation 來模糊比對資料和刪除重複資料](#)

相關範例：

- [我要如何使用 Amazon Athena 分析 Amazon S3 伺服器存取日誌？](#)

SUS04-BP06 使用共用檔案系統或儲存體存取通用資料

採用共用檔案系統或儲存體以避免資料重複，並且讓工作負載有更高效率的基礎設施。

常見的反模式：

- 您為每個用戶端佈建儲存體。
- 您未從非作用中用戶端卸離資料磁碟區。
- 您未提供跨平台和系統的儲存體存取。

建立此最佳實務的優勢：使用共用檔案系統或儲存裝置，無須複製資料即可與一或多個取用者共用資料。這有助於減少工作負載所需的儲存資源。

未建立此最佳實務時的風險暴露等級：中

實作指引

如果有多個使用者或應用程式在存取相同的資料集，則務必要使用共用儲存技術，讓您的工作負載有高效的基礎設施。共用儲存技術提供了集中儲存和管理資料的位置，可避免資料重複。此外也可強制執行跨不同系統的資料一致性。再者，共用儲存技術可讓您更有效率地使用運算能力，因為多個運算資源可同時平行存取及處理資料。

請在必要時才從這些共用儲存服務擷取資料，且應卸離未使用的磁碟區以釋出資源。

實作步驟

- 當資料有多個取用者時，將資料遷移到共用儲存體。以下是 AWS 共用儲存技術的一些範例：

Storage option	When to use
Amazon EBS Multi-Attach	Amazon EBS Multi-Attach 可讓您將單一佈建 IOPS SSD (io1 或 io2) 磁碟區連接至位於相同可用區域中的多個執行個體。
Amazon EFS	請參閱 何時應選擇 Amazon EFS 。
Amazon FSx	請參閱 選擇 Amazon FSx 檔案系統 。
Amazon S3	不需要檔案系統結構、且設計為使用物件儲存的應用程式，可使用 Amazon S3 作為可大規模擴展、耐久、低成本的物件儲存解決方案。

- 有需要時才將資料複製到共用檔案系統或從中擷取資料。例如，您可以建立[由 Amazon FSx for Lustre 支援的 Amazon S3 檔案系統](#)，並僅將處理任務所需的資料子集載入至 Amazon FSx。
- 根據您的使用模式適當刪除資料，如[SUS04-BP03 使用政策來管理資料集的生命週期](#)中所述。
- 將磁碟區與未積極使用它們的用戶端分開。

資源

相關文件：

- [將檔案系統連結至 Amazon S3 儲存貯體](#)
- [在無伺服器應用程式中對 AWS Lambda 使用 Amazon EFS](#)
- [Amazon EFS Intelligent-Tiering 優化變更存取模式的工作負載成本](#)
- [將 Amazon FSx 用於內部部署資料儲存庫](#)

相關影片：

- [透過 Amazon EFS 進行儲存成本優化](#)

SUS04-BP07 盡可能減少跨網路的資料移動

使用共用檔案系統或物件儲存體存取通用資料，將支援工作負載資料移動所需的整體網路資源降至最低。

常見的反模式：

- 無論資料使用者位於何處，您都將所有資料儲存在相同的 AWS 區域中。
- 您未優化資料大小和格式，便將其移至網路。

建立此最佳實務的優勢：優化整個網路間的資料移動，可減少工作負載所需的整體網路資源，並降低其環境影響。

未建立此最佳實務時的曝險等級：中

實作指引

要在您的組織移動資料，需要運算、聯網和儲存資源。使用相關技術盡可能減少資料移動，並改善工作負載的整體效率。

實作步驟

- 選取工作負載的區域時，請將區域與資料或使用者的鄰近性視為 [決策因素](#)。
- 對區域性使用的服務進行分區，以便將區域專屬的資料存放在使用它的區域內。
- 使用有效率的檔案格式 (例如 Parquet 或 ORC)，並在透過網路移動資料之前先壓縮資料。
- 請勿移動未使用的資料。一些有助於避免移動未使用資料的範例：
 - 減少 API 回應 (僅回應相關資料)。

- 彙總詳細的資料 (不需要記錄層級資訊)。
- 請參閱 [Well-Architected 實驗室 - 使用 Amazon Redshift 資料共用來優化資料模式](#)。
- 考慮在 [AWS Lake Formation 中使用跨帳戶資料共用](#)。
- 使用可協助您在更接近工作負載使用者的位置執行程式碼的服務。

服務	使用時機
Lambda@Edge	用於在物件未經快取時執行的大量運算作業。
CloudFront 函數	處理簡單的使用案例，例如可由短期函數起始的 HTTP(s) 請求/回應操作。
AWS IoT Greengrass	為連線的裝置執行本機運算、傳訊和資料快取。

資源

相關文件：

- [優化您的 AWS 永續性基礎架構，第 III 部分：聯網](#)
- [AWS 全球基礎設施](#)
- [Amazon CloudFront 主要功能，包括 CloudFront Global Edge Network](#)
- [壓縮 Amazon OpenSearch Service 中的 HTTP 請求](#)
- [使用 Amazon EMR 進行中間資料壓縮](#)
- [將壓縮的資料檔案從 Amazon S3 載入 Amazon Redshift](#)
- [使用 Amazon CloudFront 提供壓縮檔案服務](#)

相關影片：

- [揭密 AWS 上的資料傳輸](#)

相關範例：

- [永續性架構 - 盡可能減少跨網路的資料移動](#)

SUS04-BP08 僅在難以重新建立時才備份資料

避免備份沒有商業價值的資料，以盡可能降低工作負載的儲存資源需求。

常見的反模式：

- 您沒有資料的備份策略。
- 您備份了可輕易重新建立的資料。

建立此最佳實務的優勢：避免備份非關鍵資料可減少工作負載所需的儲存資源，並降低其環境影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

避免備份非必要的資料，有助於降低成本和工作負載所使用的儲存資源。僅備份具有商業價值或需要滿足合規要求的資料即可。檢查備份政策，並在復原案例中排除沒有價值的暫時性儲存。

實作步驟

- 實作如 [SUS04-BP01 實作資料分類政策](#) 所列的資料分類政策。
- 根據您的 [復原時間目標 \(RTO\)](#) 和 [復原點目標 \(RPO\)](#) 使用資料分類的關鍵性，並設計備份策略。避免備份非關鍵資料。
 - 排除可輕易重新建立的資料。
 - 從備份排除暫時性資料。
 - 排除資料的本機副本，除非從共同位置還原資料所需的時間不符合服務水準協議 (SLA) 的要求。
- 使用自動化解決方案或受管服務來備份業務關鍵資料。
 - [AWS Backup](#) 是一種全受管服務，可讓您在雲端和內部部署環境輕鬆集中化和自動化 AWS 服務的資料保護。如需如何使用 AWS Backup 建立自動化備份的實作指引，請參閱 [Well-Architected 實驗室 - 測試備份並還原資料](#)。
 - [使用 AWS Backup 自動進行 Amazon EFS 的備份及優化備份成本](#)。

資源

相關的最佳實務：

- [REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料](#)

- [REL09-BP03 自動執行資料備份](#)
- [REL13-BP02 使用定義的復原策略達到復原目標](#)

相關文件：

- [使用 AWS Backup 來備份和還原 Amazon EFS 檔案系統](#)
- [Amazon EBS 快照](#)
- [在 Amazon Relational Database Service 上使用備份](#)
- [APN 合作夥伴：可以幫助備份的合作夥伴](#)
- [AWS Marketplace：可用於備份的產品](#)
- [備份 Amazon EFS](#)
- [備份 Amazon FSx for Windows File Server](#)
- [Amazon ElastiCache for Redis 的備份和還原](#)

相關影片：

- [AWS re:Invent 2021 - 使用 AWS 進行備份、災難復原和勒索軟體防護](#)
- [AWS Backup 示範：跨帳戶和跨區域備份](#)
- [AWS re:Invent 2019：深入探討 AWS Backup，ft.Rackspace \(STG341\)](#)

相關範例：

- [Well-Architected 實驗室 - 測試備份並還原資料](#)
- [Well-Architected 實驗室 - 透過適用於分析工作負載的容錯回復進行備份和還原](#)
- [Well-Architected 實驗室 - 災難復原 - 備份和還原](#)

硬體和服務

透過變更硬體管理實務，尋求降低工作負載永續性影響的機會。將佈建和部署所需的硬體量降至最低，並為個別工作負載選取最高效率的硬體和服務。

最佳實務

- [SUS05-BP01 使用最低數量的硬體來滿足需求](#)

- [SUS05-BP02 使用影響最小的執行個體類型](#)
- [SUS05-BP03 使用受管服務](#)
- [SUS05-BP04 將硬體型運算加速器的使用方式最佳化](#)

SUS05-BP01 使用最低數量的硬體來滿足需求

使用最低數量的硬體讓您的工作負載有效達成商業需求。

常見的反模式：

- 您未監控資源使用率。
- 您的架構中有資源處於低使用率水準。
- 您未審查靜態硬體的使用率以判斷是否應調整其大小。
- 您未根據業務 KPI 設定運算基礎設施的硬體使用率目標。

建立此最佳實務的優勢：適當調整雲端資源大小有助於降低工作負載的環境影響、節省金錢並維護效能基準。

未建立此最佳實務時的風險暴露等級：中

實作指引

建議選取您的工作負載所需的硬體總數，以改善其整體效率。AWS 雲端 提供的彈性可透過各種機制 (例如 [AWS Auto Scaling](#)) 動態擴充或減少資源數量，以因應需求的變化。此外也提供 [API 和 SDK](#)，讓修改資源變得非常輕鬆。使用這些功能可以頻繁變更工作負載實作。此外，使用 AWS 工具提供的適當調整大小指導方針可讓您有效操作雲端資源，而達到您的商業需求。

實作步驟

- 選擇最符合您的需求的執行個體類型。
 - [如何為工作負載選擇適當的 Amazon EC2 執行個體類型？](#)
 - [Amazon EC2 機群的屬性型執行個體類型選取。](#)
 - [使用屬性型執行個體類型選取建立 Auto Scaling 群組。](#)
- 針對變動工作負載，以較小的增量進行擴展。
- 使用多種運算購買選項，以在執行個體的彈性、可擴展性和成本節省之間取得平衡。
 - [隨需執行個體](#)最適用於新的有狀態尖峰工作負載 (其執行個體類型、位置或時間不具彈性)。

- [Spot 執行個體](#) 很適合用來補強具有容錯能力和彈性的應用程式適用的其他選項。
- 對於狀態穩定、允許隨著您的需求變更保有彈性 (例如 AZ、區域、執行個體系列或執行個體類型) 的工作負載，請使用 [Compute Savings Plans](#)。
- 利用執行個體和可用區域的多樣化達到最大的應用程式可用性，並在情況允許時充分運用過剩容量。
- 使用 AWS 工具提供的適當調整大小建議，調整您的工作負載。
 - [AWS Compute Optimizer](#)
 - [AWS Trusted Advisor](#)
- 協商服務水準協議 (SLA)，允許暫時減少容量，同時自動化部署替換資源。

資源

相關文件：

- [優化您的 AWS 永續性基礎設施，第 I 部分：運算](#)
- [Amazon EC2 機群的 Auto Scaling 屬性型執行個體類型選取](#)
- [AWS Compute Optimizer 文件](#)
- [操作 Lambda：效能優化](#)
- [Auto Scaling 文件](#)

相關影片：

- [打造兼具成本、能源和資源優勢的運算環境](#)

相關範例：

- [Well-Architected 實驗室 - 在 AWS Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小 \(Level 200\)](#)

SUS05-BP02 使用影響最小的執行個體類型

持續監控並使用新的執行個體類型，讓能源效率方面的改進充分發揮效用。

常見的反模式：

- 您僅使用一個執行個體系列。

- 您僅使用 x86 執行個體。
- 您在 Amazon EC2 Auto Scaling 組態中指定了一個執行個體類型。
- 您以不符合設計宗旨的方式使用 AWS 執行個體 (例如，您將運算優化的執行個體用於記憶體密集型工作負載)。
- 您未定期評估新的執行個體類型。
- 您未查看 AWS 適當調整大小的工具 (例如 [AWS Compute Optimizer](#)) 提供的建議。

建立此最佳實務的優勢：藉由使用節能且適當調整大小的執行個體，將可大幅降低環境受到的影響以及工作負載成本。

未建立此最佳實務時的曝險等級：中

實作指引

在雲端工作負載中使用高效執行個體，是降低資源用量和提高成本效益的關鍵。持續關注新執行個體類型的發佈，並運用能源效率改進，包括旨在支援特定工作負載 (例如機器學習訓練和推論以及影片轉碼) 的執行個體類型。

實作步驟

- 了解並探索可降低工作負載環境影響的執行個體類型。
 - 訂閱 [AWS 最新消息](#)，隨時掌握最新的 AWS 技術和執行個體。
 - 了解不同的 AWS 執行個體類型。
 - 觀看下列資源，了解 AWS Graviton 型執行個體如何在 Amazon EC2 中的能源使用提供最佳效能功耗比。 [re:Invent 2020 - 深入探討搭載 AWS Graviton2 處理器的 Amazon EC2 執行個體](#) 和 [深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體](#)。
- 進行相關規劃，將工作負載轉移至影響程度最低的執行個體類型。
 - 定義一個程序來評估工作負載的新功能和執行個體。利用雲端的靈活性快速測試新功能類型對您的工作負載環境永續性有何改善。使用代理指標，測量您需要多少資源才能完成一個工作單位。
 - 如果可行，請修改工作負載，以使用不同數量的 CPU 和不同數量的記憶體，從而最大化您選擇執行個體類型的空間。
 - 考慮將您的工作負載轉移至 Graviton 型執行個體，以改善工作負載的效能效率。
 - [AWS Graviton Fast Start](#)
 - [將工作負載轉移至 AWS Graviton 型 Amazon Elastic Compute Cloud 執行個體時所需考量的事項](#)

- [AWS Graviton2 for ISVs](#)
- 請考慮選取 AWS Graviton 選項 (在您要使用的 [AWS 受管服務中](#))。
- 將工作負載遷移至有執行個體對永續性影響最小，且仍符合業務要求的區域。
- 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 [AWS Trainium](#)、[AWS Inferentia](#)，和 [Amazon EC2 DL1](#)。AWS Inferentia 執行個體 (例如 Inf2 執行個體) 所提供的效能功耗比最多會比同類 Amazon EC2 執行個體高出 50%。
- 使用 [Amazon SageMaker Inference Recommender](#) 適當調整 ML 推論端點的大小。
- 對於激增的工作負載 (不常需要額外容量的工作負載)，請使用 [高載效能執行個體](#)。
- 對於無狀態和容錯工作負載，請使用 [Amazon EC2 Spot 執行個體](#) 提高雲端整體使用率，並減少未使用資源的永續性影響。
- 操作並優化您的工作負載執行個體。
 - 對於暫時性工作負載，請評估 [執行個體 Amazon CloudWatch 指標](#) (例如 CPUUtilization)，以確認執行個體是否閒置或未充分利用。
 - 對於穩定的工作負載，請定期檢查 AWS 適當調整大小的工具 (例如 [AWS Compute Optimizer](#))，以找出對執行個體進行優化和適當調整大小的機會。
 - [Well-Architected 實驗室 - 適當調整大小的建議](#)
 - [Well-Architected 實驗室 - 使用 Compute Optimizer 適當調整大小](#)
 - [Well-Architected 實驗室 - 優化硬體模式和觀察永續性 KPI](#)

資源

相關文件：

- [優化您的 AWS 永續性基礎架構，第 I 部分：運算](#)
- [AWS Graviton](#)
- [Amazon EC2 DL1](#)
- [Amazon EC2 容量保留機群](#)
- [Amazon EC2 Spot 機群](#)
- [函數：Lambda 函數組態](#)
- [Amazon EC2 機群的屬性型執行個體類型選取](#)
- [在 AWS 上建置永續性、高效且成本優化的應用程式](#)
- [Contino 永續性儀表板如何協助客戶優化其碳足跡](#)

相關影片：

- [深入探討搭載 AWS Graviton2 處理器的 Amazon EC2 執行個體](#)
- [深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體](#)
- [打造兼具成本、能源和資源優勢的運算環境](#)

相關範例：

- [解決方案：在 AWS 上優化深度學習工作負載以維持永續性的指引](#)
- [Well-Architected 實驗室 - 適當調整大小的建議](#)
- [Well-Architected 實驗室 - 使用 Compute Optimizer 適當調整大小](#)
- [Well-Architected 實驗室 - 優化硬體模式和觀察永續性 KPI](#)
- [Well-Architected 實驗室 - 將服務遷移至 Graviton](#)

SUS05-BP03 使用受管服務

使用受管服務以提高雲端中的操作效率。

常見的反模式：

- 您使用具有低使用率的 Amazon EC2 執行個體來執行應用程式。
- 您的內部團隊僅管理工作負載，而沒有時間專注於創新或簡化。
- 您為在受管服務上可更高效執行的任務部署及維護技術。

建立此最佳實務的優勢：

- 使用受管服務可將責任轉移給 AWS，他們擁有從數百萬客戶積累而成的洞察，可帶來新的創新和效率動能。
- 基於多租用戶控制平面，受管服務將服務產生的環境影響分散到眾多使用者。

未建立此最佳實務時的風險暴露等級：中

實作指引

受管服務可將維持已部署硬體的高使用率和永續性優化的責任轉移給 AWS。受管服務也免除了維護服務的營運和管理重擔，讓您的團隊有更多時間可專注於創新。

審查您的工作負載，以識別可能被 AWS 受管服務取代的元件。例如，[Amazon RDS](#)、[Amazon Redshift](#) 和 [Amazon ElastiCache](#) 提供受管分析服務。[Amazon Athena](#)、[Amazon EMR](#) 和 [Amazon OpenSearch Service](#) 提供受管分析服務。

實作步驟

1. 清查工作負載中的服務和元件。
2. 評估並識別可能被受管服務取代的元件。以下舉例說明您可能會考慮使用受管服務的時機：

Task	What to use on AWS
託管資料庫	使用受管 Amazon Relational Database Service (Amazon RDS) 執行個體，而非在 Amazon Elastic Compute Cloud (Amazon EC2) 上維護您自己的 Amazon RDS 執行個體。
託管容器工作負載	使用 AWS Fargate ，而非實作您自己的容器基礎設施。
託管 Web 應用程式	使用 AWS Amplify 託管 ，供靜態網站和伺服器端轉譯的 Web 應用程式作為全受管 CI/CD 和託管服務。

3. 識別相依性並建立遷移計劃。據以更新執行手冊和程序手冊。
 - [AWS Application Discovery Service](#) 會自動收集並提供有關應用程式相依性和利用率的詳細資訊，協助您在計劃遷移時做出更明智的決定。
4. 在遷移至受管服務之前先測試服務。
5. 使用遷移計劃將自我託管服務取代為受管服務。
6. 在遷移完成後繼續監控服務，以便在必要時進行調整及優化服務。

資源

相關文件：

- [AWS 雲端 產品](#)
- [AWS 總體擁有成本 \(TCO\) 計算器](#)

- [Amazon DocumentDB](#)
- [Amazon Elastic Kubernetes Service \(EKS\)](#)
- [Amazon Managed Streaming for Apache Kafka \(Amazon MSK\)](#)

相關影片：

- [使用 AWS Managed Services 大規模進行雲端操作](#)

SUS05-BP04 將硬體型運算加速器的使用方式最佳化

將加速運算執行個體的使用方式最佳化，以降低工作負載的實體基礎設施需求。

常見的反模式：

- 未監控 GPU 使用率。
- 針對工作負載使用一般用途的執行個體，但專用執行個體可以提供更高的效能、較低的成本，以及更優異的效能功耗比。
- 您使用硬體型運算加速器來執行任務，但使用 CPU 型運算加速器來執行時會更有效率。

建立此最佳實務的優勢：將硬體型加速器的使用方式優化，可以降低工作負載的實體基礎設施需求。

未建立此最佳實務時的曝險等級：中

實作指引

如果需要高處理能力，使用加速運算執行個體可讓您獲得好處，因為其可讓您存取硬體型運算加速器，例如圖形處理單元 (GPU) 和現場可程式化邏輯閘陣列 (FPGA)。這些硬體加速器在執行某些功能 (例如圖形處理或資料模式比對) 時，會比 CPU 型加速器更有效率。許多加速工作負載 (例如轉譯、轉碼和機器學習) 在資源用量方面極為變化不定。只在需要時執行此硬體，不需要時便將其自動除役，以將資源消耗降至最低。

實作步驟

- 識別哪些 [加速運算執行個體](#) 可以滿足您的要求。
- 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 [AWS Trainium](#)、[AWS Inferentia](#)，和 [Amazon EC2 DL1](#)。AWS Inferentia 執行個體 (例如 Inf2 執行個體) 最多可提供 [比同類 Amazon EC2 執行個體高出 50% 的效能功耗比](#)。

- 請收集加速運算執行個體的用量指標。例如，您可以使用 CloudWatch 代理程式來收集指標，像是 `utilization_gpu` 和 `utilization_memory`，並將其用於您的 GPU，相關說明請見 [使用 Amazon CloudWatch 收集 NVIDIA GPU 指標](#)。
- 優化硬體加速器的程式碼、網路運作和設定，以確保系統會充分利用基礎硬體。
 - [優化 GPU 設定](#)
 - [Deep Learning AMI 中的 GPU 監控和優化](#)
 - [將 I/O 優化以針對 Amazon SageMaker 中的深度學習訓練進行 GPU 效能調校](#)
- 使用最新的高效能程式庫和 GPU 驅動程式。
- 使用自動化來釋出不使用的 GPU 執行個體。

資源

相關文件：

- [加速運算](#)
- [開始建構吧！使用自訂晶片和加速器來進行建構](#)
- [如何為工作負載選擇適當的 Amazon EC2 執行個體類型？](#)
- [Amazon EC2 VT1 執行個體](#)
- [選擇最佳的 AI 加速器和模型編譯以 Amazon SageMaker 推斷電腦視覺](#)

相關影片：

- [如何為深度學習選取 Amazon EC2 GPU 執行個體](#)
- [部署經濟實惠的深度學習推斷](#)

程序和文化

透過變更開發、測試和部署實務來尋找降低永續性影響的機會。

最佳實務

- [SUS06-BP01 採用可快速導入永續性改進的方法](#)
- [SUS06-BP02 讓工作負載保持在最新狀態](#)
- [SUS06-BP03 提高建置環境的使用率](#)

- [SUS06-BP04 使用受管 Device Farm 進行測試](#)

SUS06-BP01 採用可快速導入永續性改進的方法

採用相關方法和程序來驗證潛在改善、盡可能降低測試成本，以及提供小幅改善。

常見的反模式：

- 審查應用程式的永續性是僅需在專案開始時執行一次的任務。
- 您的工作負載已過時，因為發行程序太繁瑣而無法導入資源效率的小幅變更。
- 您沒有改善工作負載以維持永續性的機制。

建立此最佳實務的優勢：建立導入和追蹤永續性改善的程序後，您將可持續採用新的特性和功能、消除問題，並改善工作負載效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

在將潛在永續性改善部署到生產環境之前，先加以測試和驗證。在計算改善所帶來的未來潛在利益時，應考慮測試成本。開發低成本測試方法以提供小幅改善。

實作步驟

- 在開發積存中新增永續性改善的要求。
- 使用迭代[改善程序](#)對這些改善進行識別、評估、優先順序設定、測試及部署。
- 持續改善並簡化您的開發程序。例如，[使用持續整合與持續交付 \(CI/CD\) 管道自動執行軟體交付程序](#)，以測試及部署潛在改善，進而減少工作量和手動程序導致的錯誤。
- 使用最低可行的代表元件開發並測試潛在改善，以降低測試成本。
- 持續評估改善的影響，並視需要進行調整。

資源

相關文件：

- [AWS 提供永續性解決方案](#)

- [以 AWS CodeCommit 為基礎的可擴展敏捷開發實務](#)

相關影片：

- [提供永續、高效能的架構](#)

相關範例：

- [Well-Architected 實驗室 - 將成本和用量報告轉換為效率報告](#)

SUS06-BP02 讓工作負載保持在最新狀態

將工作負載保持在最新狀態，以採用高效功能、去除問題，以及改善工作負載的整體效率。

常見的反模式：

- 您假設目前的架構是靜態的，且不會隨著時間而更新。
- 您沒有任何系統或定期規律可評估更新的軟體與套件是否與您的工作負載相容。

建立此最佳實務的優勢：建立讓工作負載保持在最新狀態的程序後，您將可採用新的特性和功能、解決問題，並改善工作負載效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

最新的作業系統、執行階段、中介軟體、程式庫和應用程式可改善工作負載效率，讓您更輕鬆地採用更有效率的技術。隨著供應商提供符合自身永續性目標的功能，最新軟體也可能包含更準確測量工作負載對永續性影響的功能。定期以最新的功能和版本將工作負載保持在最新狀態。

實作步驟

- 定義相關程序和排程來評估工作負載的新功能和執行個體。利用雲端的靈活性快速測試新功能對您的工作負載有何改善，藉以：
 - 降低永續性的影響。
 - 獲得效能效率。
 - 消除已計劃改善的障礙。

- 提升測量和管理工作永續性影響的能力。
- 清查工作負載軟體和架構，並識別需要更新的元件。
- 您可以使用 [AWS Systems Manager Inventory](#)，從您的 Amazon EC2 執行個體收集作業系統 (OS)、應用程式和執行個體中繼資料，並快速了解哪些執行個體正在執行您的軟體政策所需的軟體與組態，以及哪些執行個體需要更新。
- 了解如何更新工作負載的元件。

Workload component	How to update
機器影像	使用 EC2 Image Builder 管理適用於 Linux 或 Windows Server 映像的 Amazon Machine Image (AMI) 的更新。
容器影像	搭配使用 Amazon Elastic Container Registry (Amazon ECR) 與您現有的管道來 管理 Amazon Elastic Container Service (Amazon ECS) 映像 。
AWS Lambda	AWS Lambda 包含 版本管理功能 。

- 使用更新程序自動化，以減少部署新功能的工作量，並避免手動程序引起的錯誤。
- 您可以使用 [CI/CD](#) 自動更新 AMI、容器映像，以及其他與您的雲端應用程式有關的成品。
- 您可以使用 [AWS Systems Manager Patch Manager](#) 之類的工具自動執行系統更新的程序，並使用 [AWS Systems Manager 維護時段](#) 來排程活動。

資源

相關文件：

- [AWS 架構中心](#)
- [AWS 最新消息](#)
- [AWS 開發人員工具](#)

相關範例：

- [Well-Architected 實驗室 - 清查和修補程式管理](#)

- [實驗室：AWS Systems Manager](#)

SUS06-BP03 提高建置環境的使用率

提高資源的使用率以開發、測試及建置您的工作負載。

常見的反模式：

- 您以手動方式佈建或終止您的建置環境。
- 您讓建置環境在測試、建置或發行活動以外執行 (例如，在開發團隊成員的非上班時間執行環境)。
- 您為建置環境過度佈建資源。

建立此最佳實務的優勢：藉由提高建置環境的使用率，您將可改善雲端工作負載的整體效率，同時為建置人員配置有效開發、測試和建置所需的資源。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用自動化和基礎設施即程式碼，在需要時啟動建置環境，並在不使用時將其關閉。常見的模式是排程可用性時間，使之與開發團隊成員的工作時間一致。您的測試環境應該會與生產組態近似。不過，請找機會使用具有高載容量的執行個體類型、Amazon EC2 Spot 執行個體、自動調整規模資料庫服務、容器和無伺服器技術，以根據使用量調整開發和測試容量。將資料量限定為剛好達到測試要求。如果在測試中使用生產資料，請尋求從生產環境共用資料的可能性，而不要移動資料。

實作步驟

- 使用基礎設施即程式碼佈建您的建置環境。
- 使用自動化來管理開發和測試環境的生命週期，並且讓建置資源發揮最大效益。
- 利用策略讓開發和測試環境達到最大的使用率。
 - 使用最低可行的代表環境來開發和測試潛在改善。
 - 在情況允許時使用無伺服器技術。
 - 使用隨需執行個體補充開發人員裝置。
 - 使用具有高載容量的執行個體類型、Spot 執行個體和其他技術，以根據使用量調整建置容量。
 - 採用原生雲端服務來獲得安全的執行個體 Shell 存取，而非部署堡壘主機機群。
 - 根據您的建置任務自動調整建置資源規模。

資源

相關文件：

- [AWS Systems Manager Session Manager](#)
- [Amazon EC2 高載效能執行個體](#)
- [什麼是 AWS CloudFormation？](#)
- [什麼是 AWS CodeBuild？](#)
- [AWS 上的 Instance Scheduler](#)

相關影片：

- [持續整合最佳實務](#)

SUS06-BP04 使用受管 Device Farm 進行測試

使用受管 Device Farm 有效測試代表性硬體集上的新功能。

常見的反模式：

- 您在個別實體裝置上手動測試及部署應用程式。
- 您未在真正的實體裝置上使用應用程式測試服務來測試及操作應用程式 (例如 Android、iOS 和 Web 應用程式)。

建立此最佳實務的優勢：使用受管 Device Farm 來測試具備雲端功能的應用程式有許多好處：

- 將有更多功能可用來測試各種裝置上的應用程式。
- 無須再以內部基礎設施進行測試。
- 提供多種裝置類型 (包括較舊且較不熱門的硬體)，因而無須再進行不必要的裝置升級。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用受管 Device Farm 有助於簡化對代表性硬體集上的新功能進行測試的程序。受管 Device Farm 提供多種裝置類型 (包括較舊且較不熱門的硬體)，並避免不必要的裝置升級對客戶的永續性造成影響。

實作步驟

- 定義您的測試要求和計劃 (例如測試類型、作業系統和測試排程)。
 - 您可以使用 [Amazon CloudWatch RUM](#) 來收集和分析用戶端資料，並研擬您的測試計劃。
- 選取可支援測試要求的受管 Device Farm。例如，您可以使用 [AWS Device Farm](#) 來測試和了解您的變更對代表性硬體集有何影響。
- 使用持續整合/持續部署 (CI/CD) 排程及執行您的測試。
 - [整合 AWS Device Farm 與您的 CI/CD 管道以執行跨瀏覽器 Selenium 測試](#)
 - [使用 AWS DevOps 和行動服務建置及測試 iOS 和 iPadOS 應用程式](#)
- 持續審查測試結果並進行必要的改進。

資源

相關文件：

- [AWS Device Farm 裝置清單](#)
- [檢視 CloudWatch RUM 儀表板](#)

相關範例：

- [Android 的 AWS Device Farm 範例應用程式](#)
- [iOS 的 AWS Device Farm 範例應用程式](#)
- [AWS Device Farm 的 Appium Web 測試](#)

相關影片：

- [透過最終使用者洞察與 Amazon CloudWatch RUM 優化應用程式](#)

結論

越來越多的組織設定永續性目標，以應對政府法規、競爭優勢，以及客戶、員工和投資者需求的變化。CTO、架構師、開發人員和營運團隊成員正在尋找其可以直接為組織的永續性目標做出貢獻的方式。藉由使用 AWS 服務支援的這些設計原則和最佳實務，您可以做出明智的決策，在安全性、成本、效能、可靠性和卓越營運與您的 AWS 雲端 工作量的永續性成果之間取得平衡。您採取以跨工作負載降低資源用量和提高效率的每個動作，減少環境影響，並達到貴組織更廣泛的永續性目標。

作者群

此文件的作者包括：

- Sam Mokhtari , Amazon Web Services 資深效率主管與解決方案架構師
- Brendan Sisson , Amazon Web Services 永續性首席解決方案架構師
- Margaret O'Toole , Amazon Web Services 永續性技術主管
- Steffen Grunwald , Amazon Web Services 永續性首席解決方案架構師
- Amazon 永續性首席工程師 Ryan Eccles
- Rodney Lester , Amazon Web Services 首席架構師
- Adrian Cockcroft , Amazon Web Services 副總裁與永續性架構師
- Ian Meyers , Amazon Web Services 解決方案架構技術總監

深入閱讀

如需其他資訊，請參閱：

- [AWS Well-Architected](#)
- [AWS 架構中心](#)
- [雲端永續性](#)
- [AWS 提供永續性解決方案](#)
- [氣候承諾](#)
- [聯合國永續發展目標](#)
- [溫室氣體協定](#)

文件修訂

若要收到此白皮書更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
已更新風險等級	對最佳實務風險等級進行了小幅度更新。	October 3, 2023
已更新最佳實務指引	最佳實務已更新，納入了以下方面的新指引： 因應需求 、 軟體和架構 、 資料 和 硬體和服務 。	July 13, 2023
已更新，適用新框架	最佳實務已更新，納入了規範性指引，並增加了新的最佳實務。	April 10, 2023
白皮書已更新	最佳實務更新了新的實作指引。	December 15, 2022
白皮書已更新	已擴充最佳實務並新增了改善計劃。	October 20, 2022
初版	永續性支柱 – AWS Well Architected Framework 已發佈。	December 2, 2021

聲明

客戶應負責對本文件中的資訊自行進行獨立評估。本文件：(a) 僅供參考之用，(b) 代表目前的 AWS 產品供應與實務，如有變更恕不另行通知，以及 (c) 不構成 AWS 及其附屬公司、供應商或授權人的任何承諾或保證。AWS 產品或服務以「現況」提供，不提供任何明示或暗示的擔保、主張或條件。AWS 對其客戶之責任與義務，應受 AWS 協議之約束，且本文件並不屬於 AWS 與其客戶間之任何協議的一部分，亦非上述協議之修改。

© 2021, Amazon Web Services, Inc. 或其合作夥伴。保留所有權利。

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the AWS 詞彙表 Reference.