

AWS Well-Architected Framework

# 效能達成效率支柱



# 效能達成效率支柱: 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 .....                                       | 1  |
| 摘要 .....  | 1  |
| 簡介 .....  | 1  |
| 效能達成效率 .....                                      | 3  |
| 設計原則 .....  | 3  |
| 定義 .....  | 3  |
| 選擇架構 .....  | 5  |
| PERF01-BP01 了解可用的雲端服務和特徵 .....                    | 5  |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務 ..... | 7  |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP03 將成本因素納入架構決策 .....                     | 9  |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP04 評估權衡如何影響客戶和架構效率 .....                 | 11 |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP05 使用政策和參考架構 .....                       | 12 |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP06 使用基準化分析來推動架構決策 .....                  | 14 |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| PERF01-BP07 針對架構選擇使用資料驅動的方法 .....                 | 16 |
| 實作指引 .....  | 6  |
| 資源 .....  | 6  |
| 運算與硬體 .....                                       | 19 |
| PERF02-BP01 選擇最適合您工作負載的運算選項 .....                 | 19 |
| 實作指引 .....  | 6  |
| 實作步驟 .....  | 6  |
| 資源 .....  | 6  |
| PERF02-BP02 了解可用的運算組態和特徵 .....                    | 22 |

|   |    |
|---|----|
| 實作指引 .....                                | 6  |
| 實作步驟 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF02-BP03 收集與運算相關的指標 .....              | 25 |
| 實作指引 .....                                | 6  |
| 實作步驟 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF02-BP04 設定運算資源及適當調整其大小 .....          | 28 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF02-BP05 動態擴展運算資源 .....                | 30 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF02-BP06 使用最佳化的硬體型運算加速器 .....          | 32 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| 資料管理 .....                                | 35 |
| PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區 ..... | 35 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF03-BP02 評估資料存放區可用的組態選項 .....          | 44 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF03-BP03 收集並記錄資料存放區效能指標 .....          | 48 |
| 實作指引 .....                                | 6  |
| 實作步驟 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF03-BP04 實作策略以改善資料存放區中的查詢效能 .....      | 51 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| PERF03-BP05 實作利用快取的資料存取模式 .....           | 53 |
| 實作指引 .....                                | 6  |
| 資源 .....                                  | 6  |
| 網路與內容交付 .....                             | 56 |
| PERF04-BP01 了解聯網如何影響效能 .....              | 56 |
| 實作指引 .....                                | 6  |

|   |    |
|---|----|
| 資源 .....  | 6  |
| PERF04-BP02 評估可用的聯網功能 .....                     | 59 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN .....         | 64 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF04-BP04 使用負載平衡將流量分配到多個資源 .....              | 66 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF04-BP05 選擇網路通訊協定以提高效能 .....                 | 70 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF04-BP06 根據網路需求選擇工作負載的位置 .....               | 72 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF04-BP07 根據指標最佳化網路組態 .....                   | 76 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| 程序和文化 .....                                     | 80 |
| PERF05-BP01 建立關鍵績效指標 (KPI) 以衡量工作負載健康狀態和效能 ..... | 81 |
| 實作指引 .....                                      | 6  |
| 實作步驟 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域 .....         | 83 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF05-BP03 定義提高工作負載效能的程序 .....                 | 86 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF05-BP04 對工作負載執行負載測試 .....                   | 87 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |
| PERF05-BP05 使用自動化主動修復效能相關問題 .....               | 89 |
| 實作指引 .....                                      | 6  |
| 資源 .....  | 6  |

---

|                                  |    |
|----------------------------------|----|
| PERF05-BP06 即時更新工作負載和服務的狀態 ..... | 91 |
| 實作指引 .....                       | 6  |
| 實作步驟 .....                       | 6  |
| 資源 .....                         | 6  |
| PERF05-BP07 定期檢閱指標 .....         | 92 |
| 實作指引 .....                       | 6  |
| 資源 .....                         | 6  |
| 結論 .....                         | 95 |
| 作者群 .....                        | 96 |
| 深入閱讀 .....                       | 97 |
| 文件修訂 .....                       | 98 |

# 效能達成效率支柱 – AWS Well-Architected Framework

出版日期：2024 年 6 月 27 日 ([文件修訂](#))

## 摘要

本白皮書著重於介紹 [AWS Well-Architected Framework](#) 的效能達成效率支柱。本文件的範圍在於提供指引，協助客戶有效率地使用雲端資源以滿足其業務需求，並隨著需求變更與技術發展維持高效率。

## 簡介

此 [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，以便在雲端設計和操作可靠、安全、有效率、經濟實惠且永續的工作負載。此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。我們相信，擁有 Well-Architected 工作負載可大幅提高企業成功的可能性。

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

本白皮書著重於如何將效能達成效率支柱原則應用到工作負載。在傳統的內部部署環境中，實現持久的高效能是一項充滿挑戰性的任務。使用本白皮書中的原則可協助您在 AWS 上建立架構，以長時間有效率地提供持續的效能。本文件中的指引和最佳實務分佈於五個關鍵重點領域，可做為在 AWS 上建置效能達成效率雲端解決方案的指導原則。這些重點領域包括：

- [選擇架構](#)
- [運算與硬體](#)
- [資料管理](#)
- [網路與內容交付](#)
- [程序和文化](#)

本白皮書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。閱讀本白皮書之後，您將了解設計效能雲端架構時要使用的 AWS 最佳實務和策略。



# 效能達成效率

效能達成效率支柱著重在有效率地使用運算資源以滿足需求，以及如何隨著需求變更與技術發展維持高效率。

## 主題

- [設計原則](#)
- [定義](#)

## 設計原則

以下設計原則可協助您在雲端中達成和保持高效率工作負載。

- 讓進階技術變得更普及：將複雜的任務委派給雲端廠商，讓團隊更輕鬆地實作進階技術。與其要求 IT 團隊了解新技術的託管和執行方式，不如考慮使用技術即服務。例如，NoSQL 資料庫、媒體轉碼和機器學習均為需要專業知識的技術。在雲端，這些技術成為團隊可以使用的服務，讓團隊可專注於產品開發，而非資源佈建及管理。
- 在幾分鐘內將業務擴展到全球：在全球多個 AWS 區域部署工作負載，讓您以最低的成本，為客戶提供更低遲、更優質的體驗。
- 使用無伺服器架構：採用無伺服器架構，您便無需執行和維護實體伺服器來完成傳統運算活動。例如，無伺服器儲存服務可以充當靜態網站 (因此無需 Web 伺服器)，而事件服務可以為您託管程式碼。如此一來，即可減輕管理實體伺服器的營運負擔，而且由於這些受管服務是在雲端規模上運行，因此還可以降低交易成本。
- 提高試驗頻率：藉助虛擬及可自動化的資源，您可以使用不同類型的執行個體、儲存設備或組態，迅速完成比較測試。
- 考慮機械共鳴作用：使用最符合您目標的技術方法。例如，為工作負載選取資料庫或儲存時，請考量資料存取模式。

## 定義

專注於下列領域，以便在雲端實現效能達成效率：

- [選擇架構](#)
- [運算與硬體](#)

- [資料管理](#)
- [網路與內容交付](#)
- [程序和文化](#)

採取資料驅動的方法來建置高效能架構。從高階設計到選取和設定資源類型，收集架構各方面的資料。

定期審查您的選擇，確保充分利用不斷演進的 AWS 雲端。監控可確保您能察覺預期效能發生的任何偏差情形。在架構中做出權衡以改進效能，例如使用壓縮或快取，或放寬一致性要求。

# 選擇架構

適用於特定工作負載的最佳解決方案各不相同，而解決方案通常會結合多種方法。Well-Architected 工作負載會使用多種解決方案，並採用不同的功能以提升效能。

AWS 資源有多種類型和組態，可讓您更輕鬆地找到最符合需求的方法。您還可以發現使用內部部署基礎設施不易實現的選項。例如，Amazon DynamoDB 這種受管服務，可提供全受管的 NoSQL 資料庫及任何規模下的十毫秒內延遲時間。

這個重點領域分享了如何選取高效率、高效能雲端資源和架構模式的各種指引和最佳實務。

## 最佳實務

- [PERF01-BP01 了解可用的雲端服務和特徵](#)
- [PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務](#)
- [PERF01-BP03 將成本因素納入架構決策](#)
- [PERF01-BP04 評估權衡如何影響客戶和架構效率](#)
- [PERF01-BP05 使用政策和參考架構](#)
- [PERF01-BP06 使用基準化分析來推動架構決策](#)
- [PERF01-BP07 針對架構選擇使用資料驅動的方法](#)

## PERF01-BP01 了解可用的雲端服務和特徵

持續了解並探索可用的服務和組態，有助您做出更完善的架構決策，並提升工作負載架構的效能效率。

常見的反模式：

- 您可以使用雲端作為並置資料中心。
- 遷移到雲端後，您沒有現代化應用程式。
- 對於需要保留的所有項目，您只使用一種儲存類型。
- 您使用的執行個體類型與目前標準最相符，但大於需求。
- 您會部署和管理可做為受管服務的技術。

建立此最佳實務的優勢：透過考慮新的服務和組態，您可以大幅改善效能、降低成本，並最佳化維護工作負載所需的工作量。這麼做也可幫助您縮短具有雲端功能之產品的價值實現時間。

未建立此最佳實務時的曝險等級：高

## 實作指引

AWS 持續推出可改善效能並降低雲端工作負載成本的新服務和特徵。即時掌握這些新服務和特徵的資訊，對於在雲端中維持效能效用至關重要。現代化工作負載架構也可有助您加速生產力、推動創新，並發掘更多成長機會。

### 實作步驟

- 清查工作負載軟體和架構以存放相關服務。決定要深入了解的產品類別。
- 探索 AWS 供應項目，以識別並了解相關服務和組態選項，這些選項可協助您改善效能，並降低成本和操作複雜性。
  - [Amazon Web Services Cloud](#)
  - [AWS Academy](#)
  - [AWS 有哪些最新消息？](#)
  - [AWS 部落格](#)
  - [AWS Skill Builder](#)
  - [AWS 活動和研討會](#)
  - [AWS 培訓 和認證](#)
  - [AWS Youtube 頻道](#)
  - [AWS 研討會](#)
  - [AWS 社群](#)
- 使用沙盒 (非生產) 環境來學習和試驗新服務，而不會產生額外成本。
- 持續了解新的雲端服務和功能。

## 資源

相關文件：

- [概述 Amazon Web Services](#)
- [Amazon EC2 功能](#)
- [使用合作夥伴學習計劃AWS逐步學習](#)
- [AWS 訓練與認證](#)

- [成為AWS解決方案架構師的學習途徑](#)
- [AWS 架構中心](#)
- [AWS Partner Network](#)
- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)
- [建置 AWS 現代化應用程式](#)

相關影片：

- [AWS re:Invent 2023 - Amazon EC2 最新消息](#)
- [AWS re:Invent 2022 - 使用 Amazon ECS 降低營運和基礎架構成本](#)
- [AWS re:Invent 2023 - 使用 AWS 建置雲端的高效率、靈活性和創新](#)
- [AWS re:Invent 2022 - 部署適用於高效能和低成本推論的機器學習模型](#)
- [This is my Architecture](#)

相關範例：

- [AWS 範例](#)
- [AWS SDK 範例](#)

## PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務

憑藉文件、解決方案架構師、專業服務或適當的合作夥伴等雲端公司資源，來引導您做出架構決策。這些資源可協助您檢閱和改善架構，以實現最佳效能。

常見的反模式：

- 您使用 AWS 做為通用雲端供應商。
- 您使用 AWS 服務的方式與其設計宗旨不符。
- 您遵循所有指引，但未考量自身的業務環境。

建立此最佳實務的優勢：運用雲端供應商或適當合作夥伴的指引，有助您選擇適合工作負載的正確架構，並對自己的決策充滿信心。

未建立此最佳實務時的曝險等級：中

## 實作指引

AWS 提供廣泛的指引、文件和資源，有助您建置和管理有效率的雲端工作負載。AWS 文件提供程式碼範例、教學課程和詳細的服務說明。除了文件外，AWS 提供培訓和認證計畫、解決方案架構師和專業服務，有助客戶探索雲端服務的不同層面，並在 AWS 上實作有效的雲端架構。

利用這些資源，深入了解寶貴的知識和最佳實務、節省時間，並運用 AWS 雲端 取得更好的成果。

### 實作步驟

- 檢閱 AWS 文件和指引，並遵循最佳實務。這些資源可協助您有效選擇和設定服務，並達到更好的效能。
  - [AWS 文件](#) (如使用者指南和白皮書)
  - [AWS 部落格](#)
  - [AWS 培訓 和認證](#)
  - [AWS Youtube 頻道](#)
- 參加 AWS 合作夥伴活動 (例如 AWS 全球高峰會、AWS re:Invent、使用者群組和研討會)，向 AWS 專家學習使用 AWS 服務的最佳實務。
  - [使用合作夥伴學習計劃AWS逐步學習](#)
  - [AWS 活動和研討會](#)
  - [AWS 研討會](#)
  - [AWS 社群](#)
- 聯絡 AWS 尋求協助，滿足您的其他指引或產品資訊需求。AWS 解決方案架構師和 [AWS 專業服務](#) 可提供解決方案實作的相關指導。[AWS 合作夥伴](#) 會提供 AWS 專業知識，協助您提升業務的靈活性和創新性。
- 使用 [AWS Support](#) 如果您需要技術支援才能有效使用服務。[我們的支援計劃](#) 旨在為您提供適當的工具組合和專業知識，以便您在最佳化效能、管理風險並控制成本的同時，透過 AWS 取得成功。

## 資源

相關文件：

- [AWS 架構中心](#)

- [AWS Partner Network](#)
- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)
- [AWS 企業支援](#)

相關影片：

- [This is my Architecture](#)
- [AWS re:Invent 2023 - 搭配 Amazon EventBridge 的進階活動驅動型模式](#)
- [AWS re:Invent 2023 - 在 AWS 上實作分散式設計模式](#)
- [AWS re:Invent 2023 - 應用程式架構即程式碼](#)

相關範例：

- [AWS 範例](#)
- [AWS SDK 範例](#)
- [AWS 分析參考架構](#)

## PERF01-BP03 將成本因素納入架構決策

將成本因素納入架構決策中，以提高雲端工作負載的資源使用率和效能效率。當您意識到雲端工作負載的成本影響時，您就更有可能利用有效的資源並減少浪費的做法。

常見的反模式：

- 您只能使用一個執行個體系列。
- 您沒有根據開放原始碼解決方案，來評估授權解決方案。
- 您沒有定義儲存生命週期政策。
- 您沒有檢閱 AWS 雲端 的新服務和特徵。
- 您只能使用區塊儲存。

建立此最佳實務的優勢：將成本因素納入決策中，可讓您使用更有效率的資源並探索其他投資選擇。

未建立此最佳實務時的曝險等級：中

## 實作指引

針對成本最佳化工作負載可以提高資源利用率，並避免雲端工作負載的浪費。將成本納入架構決策中，通常包括適當調整工作負載元件的大小和啟用彈性，進而提高雲端工作負載效能效率。

### 實作步驟

- 建立成本目標，例如雲端工作負載的預算限制。
- 找出造成工作負載成本增加的關鍵元件 (例如執行個體和儲存)。您可以使用 [AWS Pricing Calculator](#) 和 [AWS Cost Explorer](#) 找出造成工作負載中成本增加的關鍵因素。
- 了解 [在雲端中](#) 的定價模式，例如隨選、保留執行個體和 Savings Plans Spot 執行個體。
- 使用 [AWS Well-Architected 成本最佳化最佳實務](#) 最佳化這些關鍵元件的成本。
- 持續監控和分析成本，以找出工作負載中成本最佳化的機會。
  - 使用 [AWS Budgets](#) 在產生無法接受的成本時收到提醒。
  - 使用 [AWS Compute Optimizer](#) 或者 [AWS Trusted Advisor](#) 取得成本最佳化建議。
  - 使用 [AWS 成本異常偵測](#) 取得自動化的成本異常偵測和根本原因分析。

## 資源

相關文件：

- [什麼是 AWS 帳單和成本管理？](#)
- [使用 AWS 進行成本最佳化](#)
- [選擇 AWS 成本管理策略](#)
- [AWS 成本管理入門指南](#)
- [成本智慧儀表板的詳細概要](#)
- [AWS 架構中心](#)
- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)

相關影片：

- [This is my Architecture](#)
- [AWS re:Invent 2023 - AWS 成本最佳化的最新消息](#)



- [AWSRE:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度](#)
- [AWS re:Invent 2023 - AWS 儲存成本最佳化最佳實務](#)
- [AWS re:Invent 2023 - 最佳化在多帳戶環境中的成本](#)

相關範例：

- [AWS Compute Optimizer 示範程式碼](#)
- [成本最佳化研討會](#)
- [雲端財務管理技術實作教戰手冊](#)
- [新創公司最佳化：調整應用程式效能以達到最大效率](#)
- [無伺服器最佳化研討會 \(效能與成本\)](#)
- [擴展具成本效益的架構](#)

## PERF01-BP04 評估權衡如何影響客戶和架構效率

在評估與效能相關的改善之處時，請判斷哪些選擇將對客戶和工作負載效率產生影響。例如，如果使用鍵值資料存放區可提高系統效能，請務必評估此變更最終一致性本質對客戶的影響。

常見的反模式：

- 即使實作過程中有所取捨，您都假設應實作所有效能增益。
- 您只會在效能問題達到臨界點時才會評估工作負載變更。

建立此最佳實務的優勢：評估與效能相關的潛在改善時，您必須決定進行此變更的優缺點是否符合工作負載需求。在某些情況下，您可能需要實作其他控制來彌補權衡。

未建立此最佳實務時的曝險等級：高

### 實作指引

根據對效能和客戶造成的影響，找出架構中的關鍵領域。確定如何進行改進、這些改進帶來的權衡，以及它們如何影響系統和使用者體驗。例如，實作快取資料有助於大幅提升效能，但需要明確的策略來確定更新或使快取資料失效的方式和時間，以防止不正確的系統行為。

### 實作步驟

- 了解工作負載需求和 SLA。

- 清楚定義評估因素。因素可能涉及工作負載的成本、可靠性、安全性和效能。
- 選擇可滿足您需求的架構和服務。
- 進行試驗和概念驗證 (POC)，以評估權衡因素以及對客戶和架構效率的影響。通常，高可用性、高效能且安全的工作負載會耗用更多雲端資源，但能夠提供更完善的客戶體驗。了解工作負載複雜性、效能和成本之間的權衡。通常，如果優先考慮兩個因素，將會犧牲第三個因素。

## 資源

相關文件：

- [Amazon 建置者資料中心](#)
- [Amazon QuickSight KPI](#)
- [Amazon CloudWatch RUM](#)
- [X-Ray 文件](#)
- [了解能在雲端中快速建構架構的彈性模式和權衡](#)

相關影片：

- [最佳化應用程式透過 Amazon CloudWatch RUM](#)
- [AWS re:Invent 2023 - 容量、可用性、成本效率：聚焦三件事](#)
- [AWS re:Invent 2023 - 適用於鬆散耦合系統的進階整合模式與權衡](#)

相關範例：

- [使用 Amazon CloudWatch Synthetics 測量頁面載入時間](#)
- [Amazon CloudWatch RUM Web 用戶端](#)

## PERF01-BP05 使用政策和參考架構

選擇服務和組態時，使用內部政策和現有的參考架構，以便在設計和實作工作負載時提高效率。

常見的反模式：

- 您可以廣泛使用可能影響公司管理開銷的技術。

建立此最佳實務的優勢： 建立架構、技術和供應商選擇的政策，讓您快速做出決策。

未建立此最佳實務時的曝險等級： 中

## 實作指引

在選擇資源和架構時，制定內部政策可讓您在選擇架構時有可遵循的標準和準則。這些指導方針則可簡化在選擇合適的雲端服務時的決策流程，並提高效能效率。使用政策或參考架構來部署工作負載。將服務整合到雲端部署，然後使用效能測試以確保您可以繼續滿足效能需求。

## 實作步驟

- 清楚了解雲端工作負載的需求。
- 檢閱內部和外部政策，以找出最相關的政策。
- 使用由 AWS 或業界最佳實務提供的適當參考架構。
- 針對常見情況，建立包含政策、標準、參考架構和規範性指引的連續體。這樣做可加快團隊的應變速度。為產業量身打造資產 (如果適用)。
- 針對沙盒環境中的工作負載驗證這些政策和參考架構。
- 隨時掌握產業標準和 AWS 更新，確保政策和參考架構有助最佳化雲端工作負載。

## 資源

相關文件：

- [AWS 架構中心](#)
- [AWS Partner Network](#)
- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)
- [AWS 建築部落格](#)

相關影片：

- [This is my Architecture](#)
- [AWS re:Invent 2022 - 利用 SAP 和 AWS 參考架構提升業務價值](#)

相關範例：

- [AWS 範例](#)
- [AWS SDK 範例](#)

## PERF01-BP06 使用基準化分析來推動架構決策

基準化分析現有工作負載的效能，以了解工作負載在雲端的效能，並根據該資料推動架構決策。

常見的反模式：

- 您依賴的常見基準化分析未能反映工作負載特性。
- 您依賴客戶意見回饋和感受作為唯一的基準化分析。

建立此最佳實務的好處：基準化分析目前的實作，可協助您衡量效能改善之處。

未建立此最佳實務時的風險暴露等級：中

### 實作指引

使用基準化分析搭配綜合測試，以評估工作負載元件執行情況。與負載測試相比，基準化分析通常速度更快；要評估特定元件的技術時，會使用基準化分析。當您缺少執行負載測試的完整解決方案時，通常可在新專案開始時使用基準化分析。

您可以建置自己的自訂基準分析測試，也可以使用產業標準測試 (例如 [TPC-DS](#))，對工作負載進行基準分析。比較環境時，產業基準化分析很有幫助。對於確定您希望在架構中進行的特定營運類型，自訂基準化分析非常實用。

基準化分析時，務必要預熱測試環境，以獲得有效的結果。多次執行相同的基準化分析，以確保您已擷取到隨著時間出現的任何變化。

由於基準化分析的速度通常比負載測試要快，因此可以在部署管道中盡早使用基準化分析，以便能更快提供有關效能偏差的回饋。當您評估元件或服務中的重大變更時，藉助基準化分析，您可以更快速地查看所做的變更是否合理。請務必使用基準化分析搭配負載測試，因為負載測試會告訴您工作負載在生產中的效能。

### 實作步驟

- 規劃和定義：
  - 為您的基準定義目標、基準線、測試案例、指標 (例如 CPU 使用率、延遲或輸送量) 以及 KPI。

- 關注使用者體驗方面的需求，以及回應時間和可存取性等因素。
- 找出適合您工作負載的基準工具。您可以使用 AWS 服務 (例如 [Amazon CloudWatch](#))，或與您的工作負載相容的第三方工具。
- 組態設定和檢測：
  - 設定您的環境並配置您的資源組態。
  - 實作監控和記錄以擷取測試結果。
- 基準化分析及監控：
  - 在測試期間執行基準化分析並監控指標。
- 分析和記錄：
  - 記錄您的基準化分析流程和調查結果。
  - 分析結果以識別瓶頸、趨勢和待改進領域。
  - 使用測試結果做出架構決策，並調整工作負載。這可能包括變更服務或採用新功能。
- 最佳化並重複：
  - 根據您的基準調整資源組態和配置。
  - 調整後重新測試您的工作負載，以驗證您的改善情況。
  - 記錄您的學習過程，並重複該過程以找出其他需要改進的地方。

## 資源

### 相關文件：

- [AWS 架構中心](#)
- [AWS Partner Network](#)
- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)
- [Amazon CloudWatch RUM](#)
- [Amazon CloudWatch Synthetics](#)
- [基因組學工作流程，第 5 部分：自動基準化分析](#)
- [在 Amazon SageMaker JumpStart 中基準化分析及最佳化端點部署](#)

### 相關影片：

- [AWS re:Invent 2023 - 基準化分析 AWS Lambda 冷啟動](#)
- [在雲端中進行狀態化服務的基準化分析](#)
- [This is my Architecture](#)
- [透過 Amazon CloudWatch RUM 最佳化應用程式](#)
- [Amazon CloudWatch Synthetics 的示範](#)

相關範例：

- [AWS 範例](#)
- [AWS SDK 範例](#)
- [分散式負載測試](#)
- [使用 Amazon CloudWatch Synthetics 測量頁面載入時間](#)
- [Amazon CloudWatch RUM Web 用戶端](#)

## PERF01-BP07 針對架構選擇使用資料驅動的方法

為各種架構選擇定義適合的清晰、資料驅動型方法，以確認是否使用正確的雲端服務和組態，滿足您的特定業務需求。

常見的反模式：

- 您認為自己的目前架構是固定不變的，且不應隨著時間而更新。
- 架構選擇是根據猜測和假設。
- 你隨著時間推移而導入了架構變更，卻沒有提供充分的理由。

建立此最佳實務的優勢：透過採用明確定義的方法來做出架構選擇，您使用資料來影響工作負載設計，並隨著時間的推移做出明智的決策。

未建立此最佳實務時的曝險等級：中

### 實作指引

使用內部經驗和雲端知識，或使用外部資源 (例如已發佈的使用案例或白皮書)，以選擇架構中的資源和服務。您應有定義明確的流程，有助試驗和基準化分析可在工作負載中使用的服務。

關鍵工作負載的待辦項目不僅只包括使用者案例 (提供與業務和使用者相關的功能)，還應包括構成工作負載架構跑道的技術案例。這條跑道掌握技術和新服務的新進展，並根據資料和適當的理由而採用這些技術和新服務。這證明架構仍然是與時俱進，不會停滯不前。

## 實作步驟

- 與關鍵利害關係人互動，以定義工作負載需求，包括效能、可用性和成本考量。考慮工作負載的使用者數量和使用模式等因素。
- 建立架構跑道，或根據功能待辦項目優先順序設定的技術待辦項目。
- 評估不同的雲端服務 (如需詳細資訊，請參閱 [PERF01-BP01 了解可用的雲端服務和特徵](#))。
- 探索符合效能需求的不同架構模式，例如微服務或無伺服器 (如需更多詳細資訊，請參閱 [PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務](#))。
- 諮詢其他團隊、架構圖表和資源，例如 AWS 解決方案架構設計師、[AWS 架構中心](#) 和 [AWS Partner Network](#)，以協助您選擇適合工作負載的架構。
- 定義輸送量和回應時間等效能指標，以協助您評估工作負載的效能。
- 試驗並使用定義的指標，來驗證所選架構的效能。
- 視需要持續監控並進行調整，以維持架構的最佳效能。
- 記錄您選擇的架構和決策，作為未來更新和學習的參考。
- 根據經驗、新技術和指出目前方法中需要變更或問題的指標，持續檢閱和更新架構選擇方法。

## 資源

相關文件：

- [AWS 解決方案程式庫](#)
- [AWS 知識中心](#)
- [可在 AWS 上建置端對端資料驅動型應用程式的架構模式](#)

相關影片：

- [This is my Architecture](#)
- [AWS re:Invent 2021 - 資料驅動型企業：從願景轉為價值](#)

- [AWS re:Invent 2022 - 提供永續且高效能的架構](#)
- [AWSRE:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度](#)
- [AWS re:Invent 2022 - AWS 最佳化：讓您立即獲得結果的可採取動作的步驟](#)

相關範例：

- [AWS 範例](#)
- [AWS SDK 範例](#)



# 運算與硬體

特定工作負載的最佳運算選擇會根據應用程式設計、使用模式和組態設定而有所不同。架構會針對不同元件使用不同運算選擇，並採用不同功能以提升效能。若選錯運算資源，可能使架構的效能達成效率降低。

這個重點領域分享了如何識別和最佳化運算選項，以在雲端中實現效能達成效率的各種指引和最佳實務。

## 最佳實務

- [PERF02-BP01 選擇最適合您工作負載的運算選項](#)
- [PERF02-BP02 了解可用的運算組態和特徵](#)
- [PERF02-BP03 收集與運算相關的指標](#)
- [PERF02-BP04 設定運算資源及適當調整其大小](#)
- [PERF02-BP05 動態擴展運算資源](#)
- [PERF02-BP06 使用最佳化的硬體型運算加速器](#)

## PERF02-BP01 選擇最適合您工作負載的運算選項

為工作負載選擇最合適的運算選項，可讓您改善效能、減少不必要的基礎架構成本，並降低維護工作負載所需的作業工作量。

常見的反模式：

- 您使用曾用於內部部署的同一個運算選項。
- 您不了解雲端運算選項、特徵以及解決方案，以及那些解決方案可以如何改善運算效能。
- 您在替代運算選項更精確地符合工作負載特性時，過度佈建現有運算選項以符合擴展或效能需求。

建立此最佳實務的好處：透過找出運算需求並根據可用選項進行評估，您就可以提高工作負載的資源效率。

未建立此最佳實務時的風險暴露等級：高

## 實作指引

為了最佳化雲端工作負載以提高效能效率，請務必根據使用案例和效能需求選擇最合適的運算選項。AWS 提供多種運算選項，以滿足雲端中不同工作負載的需求。例如，您可以使用 [Amazon EC2](#) 來啟動和管理虛擬伺服器、使用 [AWS Lambda](#) 來執行程式碼，而不必佈建或管理伺服器、使用 [Amazon ECS](#) 或 [Amazon EKS](#) 執行和管理容器，或者使用 [AWS Batch](#) 平行處理大量資料。根據擴展和運算需求，您應該根據自己的情況選擇並設定最佳的運算解決方案。您也可以考慮在單一工作負載中使用多種運算解決方案，因為每種運算解決方案都有優缺點。

下列步驟會引導您選擇正確的運算選項，以符合工作負載特性和效能需求。

## 實作步驟

- 了解工作負載運算需求。要考量的關鍵需求包括處理需求、流量模式、資料存取模式、擴展需求，以及延遲需求。
- 了解在 AWS 上適用於工作負載的不同運算選項 (如 [PERF01-BP01 了解可用的雲端服務和特徵](#) 中所述)。以下是一些關鍵的 AWS 運算選項、其特性和常見使用案例：

| AWS service   | Key characteristics   | Common use cases  |
|---|---|---|
| <a href="#">Amazon Elastic Compute Cloud (Amazon EC2)</a>   | Has dedicated option for hardware, license requirements, large selection of different instance families, processor types and compute accelerators | Lift and shift migrations, monolithic application, hybrid environments, enterprise applications |
| <a href="#">Amazon Elastic Container Service (Amazon ECS)</a> ,<br><a href="#">Amazon Elastic Kubernetes Service (Amazon EKS)</a> | Easy deployment, consistent environments, scalable  | Microservices, hybrid environments  |
| <a href="#">AWS Lambda</a>  | <a href="#">無伺服器運算</a> service that runs code in response to events and automatically manages the underlying compute resources.                   | Microservices, event-driven applications  |

| AWS service                      | Key characteristics  | Common use cases                        |
|----------------------------------|--|---|
| <a href="#">AWS Batch</a>        | Efficiently and dynamically provisions and scales <a href="#">Amazon Elastic Container Service (Amazon ECS)</a> , <a href="#">Amazon Elastic Kubernetes Service (Amazon EKS)</a> , and <a href="#">AWS Fargate</a> compute resources, with an option to use On-Demand or Spot Instances based on your job requirements | HPC, train ML models                    |
| <a href="#">Amazon Lightsail</a> | Preconfigured Linux and Windows application for running small workloads  | Simple web applications, custom website |

- 評估與每個運算選項相關聯的成本 (例如每小時費用或資料傳輸) 和管理開銷 (例如修補和調整規模)。
- 在非生產環境中執行試驗和基準化分析，以找出哪個運算選項最能滿足工作負載需求。
- 在您試驗和找出新的運算解決方案，請規劃遷移並驗證效能指標。
- 使用像是 [Amazon CloudWatch](#) 的 AWS 監控工具，和 [AWS Compute Optimizer](#) 之類的最佳化服務，根據實際使用模式持續最佳化運算資源。

## 資源

相關文件：

- [使用 AWS 進行雲端運算](#)
- [Amazon EC2 執行個體類型](#)
- [Amazon EKS 容器：Amazon EKS 工作節點](#)
- [Amazon ECS 容器：Amazon ECS 容器執行個體](#)
- [函數：Lambda 函數組態](#)

- [容器的規範指引](#)
- [無伺服器的規範指引](#)

相關影片：

- [AWS re:Invent 2023 - AWS Graviton：AWS 工作負載的最佳性價比](#)
- [AWS re:Invent 2023 - AMS 中新增 Amazon Elastic Compute Cloud 生成式 AI 功能](#)
- [AWS re:Invent 2023 - Amazon Elastic Compute Cloud 的最新消息](#)
- [AWS re:Invent 2023 - 智慧型節約：Amazon Elastic Compute Cloud 成本最佳化策略](#)
- [AWS re:Invent 2021 - 支援新一代 Amazon Elastic Compute Cloud：深入探討 Nitro System](#)
- [AWS re:Invent 2019 - 最佳化 AWS 運算的效能和成本](#)
- [AWS re:Invent 2019 - Amazon Elastic Compute Cloud 基礎](#)
- [AWS re:Invent 2022 - 部署適用於高效能和低成本推論的機器學習模型](#)
- [AWS re:Invent 2019 - 最佳化 AWS 運算的效能和成本](#)
- [Amazon EC2 基礎](#)
- [部署適用於高效能和低成本推論的機器學習模型](#)

相關範例：

- [遷移 Web 應用程式至容器](#)
- [執行 Serverless Hello World](#)
- [Amazon EKS 研討會](#)
- [Amazon EC2 研討會](#)
- [使用 Amazon Elastic Compute Cloud 自動擴展功能的高效率和彈性的工作負載](#)
- [使用容器服務遷移至 AWS Graviton](#)

## PERF02-BP02 了解可用的運算組態和特徵

了解運算服務的可用組態選項和特徵，有助您佈建適量的資源並提高效能效率。

常見的反模式：

- 您沒有根據工作負載特性，評估運算選項或可用的執行個體系列。

- 您為了滿足尖峰需求而過度佈建運算資源。

建立此最佳實務的優勢：熟悉 AWS 運算特徵和組態，如此您就能使用為符合工作負載特性和需求而經最佳化的運算解決方案。

未建立此最佳實務時的曝險等級：中

## 實作指引

每個運算解決方案都有獨特的組態和特徵，可支援不同的工作負載特性和需求。了解這些選項如何與工作負載互補，以及判斷哪種組態選項最適合您的應用程式。這些選項的範例包括執行個體系列、大小、特徵 (GPU、I/O)、爆量、逾時、函數大小、容器執行個體，以及並行。如果工作負載使用相同運算選項的時間已超過四週，並且您預計特性未來仍將保持不變，您可以使用 [AWS Compute Optimizer](#) 從 CPU 和記憶體的角度來判斷目前的運算選項是否適合此工作負載。

## 實作步驟

1. 了解工作負載需求 (例如 CPU 需求、記憶體和延遲)。
2. 檢閱 AWS 文件和最佳實務，以了解可協助改善運算效能的建議組態選項。以下是一些需要考慮的關鍵組態選項：

| 組態選項   | 範例  |
|--------|---|
| 執行個體類型 | <ul style="list-style-type: none"> <li>• <a href="#">運算最佳化</a> 執行個體非常適合需要高 vCPU 與記憶體比例的工作負載。</li> <li>• <a href="#">記憶體最佳化</a> 執行個體會提供大量記憶體，以支援記憶體密集工作負載。</li> <li>• <a href="#">儲存最佳化</a> 執行個體專為需要對本機儲存進行高循序讀寫存取 (IOPS) 的工作負載而設計。</li> </ul> |
| 定價模式   | <ul style="list-style-type: none"> <li>• <a href="#">隨需執行個體</a> 讓您無需長期承諾，即可按小時或秒使用運算容量。這些執行個體非常適合應對超出基本性能需求的突增負載。</li> <li>• <a href="#">Savings Plans</a> 與隨需執行個體相比，可大幅節省成本，但使用者必須在一年或三年期間使用特定數量的運算能力。</li> </ul>                            |

| 組態選項         | 範例  |
|--------------|---|
|              | <ul style="list-style-type: none"> <li>• <a href="#">Spot 執行個體</a> 可讓您以折扣價利用未使用的執行個體容量，以實現無狀態且具有容錯功能的工作負載。</li> </ul>   |
| Auto Scaling | 使用 <a href="#">Auto Scaling</a> 組態來比對運算資源與流量模式。   |
| 規模調整         | <ul style="list-style-type: none"> <li>• 使用 <a href="#">Compute Optimizer</a> 取得採用機器學習技術的建議，以了解哪個運算組態最符合您的運算特性。</li> <li>• 使用 <a href="#">AWS Lambda 功能調校</a>，為 Lambda 函數選擇最佳組態。</li> </ul>   |
| 硬體型運算加速器     | <ul style="list-style-type: none"> <li>• <a href="#">加速運算執行個體</a> 執行圖形處理或資料模式比對等功能時，會比以 CPU 為基礎的替代方案更有效率。</li> <li>• 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 <a href="#">AWS Trainium</a>、<a href="#">AWS Inferentia</a>，和 <a href="#">Amazon EC2 DL1</a></li> </ul> |

## 資源

相關文件：

- [使用 AWS 進行雲端運算](#)
- [Amazon EC2 執行個體類型](#)
- [Amazon EC2 執行個體的處理器狀態控制](#)
- [Amazon EKS 容器：Amazon EKS 工作節點](#)
- [Amazon ECS 容器：Amazon ECS 容器執行個體](#)
- [函數：Lambda 函數組態](#)

相關影片：

- [AWS re:Invent 2023 – AWS Graviton : AWS 工作負載的最佳性價比](#)
- [AWS re:Invent 2023 – AWS Management Console 中新增 Amazon EC2 生成式 AI 功能](#)
- [AWS re:Invent 2023 – Amazon EC2 最新消息](#)
- [AWS re:Invent 2023 – 智慧優惠組合 : Amazon EC2 成本最佳化策略](#)
- [AWS re:Invent 2021 – 支援下一代 Amazon EC2 : 深入探討 Nitro 系統](#)
- [AWS re:Invent 2019 – Amazon EC2 基礎](#)
- [AWS re:Invent 2022 – https://www.youtube.com/watch?v=5B4-s\\_ivn1o](https://www.youtube.com/watch?v=5B4-s_ivn1o)

相關範例：

- [Compute Optimizer 示範程式碼](#)
- [Amazon EC2 spot 執行個體研討會](#)
- [使用 Amazon EC2 AWS Auto Scaling 達到高效且彈性的工作負載](#)
- [Graviton 開發人員研討會](#)
- [AWS for Microsoft workloads immersion day](#)
- [AWS for Linux workloads immersion day](#)
- [AWS Compute Optimizer 示範程式碼](#)
- [Amazon EKS 研討會](#)

## PERF02-BP03 收集與運算相關的指標

記錄並追蹤與運算相關的指標，進一步了解運算資源的效能，並改善效能及使用率。

常見的反模式：

- 您只使用手動日誌檔案來搜尋指標。
- 您只使用監控軟體記錄的預設指標。
- 您只會在有問題時審查指標。

建立此最佳實務的優勢：收集效能相關指標，可協助您符合應用程式效能與業務需求，確保滿足工作負載需求。這麼做也可以協助您持續改善工作負載中的資源效能和使用率。

未建立此最佳實務時的曝險等級：高

## 實作指引

雲端工作負載可以產生大量資料，例如指標、日誌和事件。在 AWS 雲端中，收集指標是提高安全性、成本效率、效能和可永續發展性的關鍵步驟。AWS 可使用監控服務提供各種效能相關的指標，例如 [Amazon CloudWatch](#) 為您提供寶貴的洞察。CPU 使用率、記憶體使用率、磁碟 I/O 以及網路輸入和輸出等指標，可協助您深入了解使用率層級或效能瓶頸。將這些指標納入資料驅動的方法，以主動調整和優化工作負載的資源。在理想的情況下，您應該在單一平台中收集與運算資源相關的所有指標，並實作保留政策，以支援成本和營運目標。

## 實作步驟

1. 找出與工作負載相關的效能相關指標。您應該收集與資源使用率和雲端工作負載運作方式有關的指標 (例如回應時間和輸送量)。
  - a. [Amazon EC2 預設指標](#)
  - b. [Amazon ECS 預設指標](#)
  - c. [Amazon EKS 預設指標](#)
  - d. [Lambda 預設指標](#)
  - e. [Amazon EC2 記憶體和磁碟指標](#)
2. 為工作負載選擇並設定合適的記錄和監控解決方案。
  - a. [AWS 原生可觀測性](#)
  - b. [適用於 OpenTelemetry 的 AWS Distro](#)
  - c. [Amazon Managed Service for Prometheus](#)
3. 根據工作負載需求，為指標定義必要的篩選條件和彙總。
  - a. [使用 Amazon CloudWatch Logs 和指標篩選條件，量化自訂應用程式指標](#)
  - b. [使用 Amazon CloudWatch 策略標記收集自訂指標](#)
4. 為指標設定資料保留政策，以符合安全性和營運目標。
  - a. [CloudWatch 指標的預設資料保留](#)
  - b. [CloudWatch Logs 的預設資料保留](#)
5. 如有必要，為指標建立警示和通知，以協助您主動回應效能相關問題。
  - a. [使用 Amazon CloudWatch 異常偵測為自訂指標建立警示](#)
  - b. [使用 Amazon CloudWatch RUM 為特定網頁建立指標和警示](#)
6. 使用自動化來部署指標和記錄彙總代理程式。



- a. [AWS Systems Manager 自動化](#)
- b. [OpenTelemetry 收集器](#)

## 資源

相關文件：

- [監控與可觀察性](#)
- [最佳做法：使用 AWS 實作可觀察性](#)
- [Amazon CloudWatch 文件](#)
- [使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌](#)
- [存取 Amazon CloudWatch Logs 進行 AWS Lambda](#)
- [搭配容器執行個體使用 CloudWatch Logs](#)
- [發佈自訂指標](#)
- [AWS Answers：集中式記錄](#)
- [發佈 CloudWatch 指標的 AWS 服務](#)
- [監控 AWS Fargate Amazon EKS](#)

相關影片：

- [AWS re:Invent 2023 – \[LAUNCH\] 現代工作負載的應用程式監控](#)
- [AWS re:Invent 2023 – 實作應用程式可觀察性](#)
- [AWS re:Invent 2023 – 打造有效的可觀察性策略](#)
- [AWS re:Invent 2023 – 使用 AWS Distro for OpenTelemetry 實現無縫可觀察性](#)
- [AWS 上的應用程式效能管理](#)

相關範例：

- [AWS for Linux Workloads Immersion Day- Amazon CloudWatch](#)
- [監控 Amazon ECS 叢集和容器](#)
- [使用 Amazon CloudWatch 儀表板進行監控](#)
- [Amazon EKS 研討會](#)

## PERF02-BP04 設定運算資源及適當調整其大小

設定運算資源及適當調整其大小，以符合工作負載的效能需求，並避免未充分使用資源或過度使用資源的情況。

常見的反模式：

- 您忽略工作負載效能需求，導致過度佈建或佈建不足的運算資源。
- 您只選擇適用於所有工作負載的最大或最小執行個體。
- 為了方便管理，您只使用一個執行個體系列。
- 您忽略來自 AWS Cost Explorer 或 Compute Optimizer 適當調整大小的建議。
- 您沒有重新評估工作負載是否適用於新的執行個體類型。
- 您只驗證組織的少量執行個體組態。

建立此最佳實務的優勢：適當調整運算資源的大小，可避免過度佈建和佈建不足的資源，以確保雲端中的最佳作業。適當調整運算資源的大小，通常可以提高效能和增強客戶體驗，同時降低成本。

未建立此最佳實務時的曝險等級：中

### 實作指引

適當調整大小可讓組織以有效率且符合成本效益的方式操作雲端基礎架構，同時滿足其業務需求。過度佈建雲端資源可能會導致額外成本，而佈建不足可能會導致低落的效能和不佳的客戶體驗。AWS 提供類似 [AWS Compute Optimizer](#) 和 [AWS Trusted Advisor](#) 之類的工具，這類工具會使用歷史資料，來提供適當調整運算資源大小的建議。

### 實作步驟

- 選擇最適合您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。](#)
  - [運用 Karpenter 整合，最佳化 Kubernetes 運算成本](#)
- 分析工作負載的各種效能特性，以及這些特性與記憶體、網路和 CPU 使用量的關係。使用此資料，選擇最適合您工作負載設定檔和效能目標的資源。

- 使用 Amazon CloudWatch 之類的 AWS 監控工具，監控資源使用情況。
- 為運算資源選取適合的組態。
  - 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 和 AWS Trusted Advisor)，以找出對運算資源進行最佳化和適當調整大小的機會。
- 在即時環境中實作之前，先測試非生產環境中的組態變更。
- 持續重新評估新的運算供應項目，並且根據工作負載需求進行比較。

## 資源

### 相關文件：

- [使用 AWS 進行雲端運算](#)
- [Amazon EC2 執行個體類型](#)
- [Amazon ECS 容器：Amazon ECS 容器執行個體](#)
- [Amazon EKS 容器：Amazon EKS 工作節點](#)
- [函數：Lambda 函數組態](#)
- [Amazon EC2 執行個體的處理器狀態控制](#)

### 相關影片：

- [Amazon EC2 基礎](#)
- [AWS re:Invent 2023 – AWS Graviton：AWS 工作負載的最佳性價比](#)
- [AWS re:Invent 2023 – AWS Management Console 中新增 Amazon EC2 生成式 AI 功能](#)
- [AWS re:Invent 2023 – Amazon EC2 最新消息](#)
- [AWS re:Invent 2023 – 智慧優惠組合：Amazon EC2 成本最佳化策略](#)
- [AWS re:Invent 2021 – 支援下一代 Amazon EC2：深入探討 Nitro 系統](#)
- [AWS re:Invent 2019 – Amazon EC2 基礎](#)

### 相關範例：

- [AWS Compute Optimizer 示範程式碼](#)

- [Amazon EKS 研討會](#)
- [適當調整大小的建議](#)

## PERF02-BP05 動態擴展運算資源

為滿足需求，請使用雲端的彈性，來動態擴充或縮減運算資源，並避免為工作負載佈建過多或過少的容量。

常見的反模式：

- 您可以手動增加容量，對警示做出反應。
- 您使用與內部部署相同的大小準則 (通常是靜態基礎結構)。
- 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：設定和測試運算資源的彈性，可協助您節省成本、維持效能基準，並隨著流量變化改善可靠性。

未建立此最佳實務時的曝險等級：高

### 實作指引

AWS 透過各種擴展機制，讓您能夠彈性動態擴充或縮減資源，以因應需求的變化。動態擴展與運算相關指標結合，讓工作負載能夠自動回應變更，並使用最佳的運算資源集來達成其目標。

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：：監控擴展指標，並視需要自動增加或減少容量。
- 預測擴展：根據預測每日和每週趨勢進行擴展。
- 排程法：根據可預測的負載變更來設定您自己的擴展排程。
- 服務擴展：選擇可根據設計自動擴展的服務 (例如無伺服器)。

您必須確保工作負載部署可以同時處理擴展和縮減事件。

### 實作步驟

- 運算執行個體、容器和函數提供了彈性機制，可能是與自動調整規模功能結合使用，或是作為服務功能提供。以下是自動擴展機制的幾個範例：

| 自動擴展  | 應用環境   |
|---|--|
| <a href="#">Amazon EC2 Auto Scaling</a>                 | 確認您擁有正確數量的 <a href="#">Amazon EC2</a> 執行個體可處理應用程式的使用者負載。   |
| <a href="#">Application Auto Scaling</a>                | 自動將個別 AWS 服務的資源擴展到 Amazon EC2 以外，例如 <a href="#">AWS Lambda</a> 函數或 <a href="#">Amazon Elastic Container Service (Amazon ECS)</a> 服務。 |
| <a href="#">Kubernetes Cluster Autoscaler/Karpenter</a> | 自動擴展 Kubernetes 叢集。  |

- 擴展經常會與 Amazon EC2 執行個體或 AWS Lambda 函數等運算服務一併討論。請務必同時考慮非運算服務的組態，例如，[AWS Glue](#)，以符合需求。
- 確認用於擴展的指標符合要部署之工作負載的特性。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。請改用轉碼任務佇列的深度。您可以將 [自訂指標](#) 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。
- 請確定您使用的是 [動態擴展](#) 而非 [手動擴展](#) 處理您的 Auto Scaling 群組。我們也建議您將 [目標追蹤擴展政策](#) 用於您的動態擴展。
- 確認工作負載部署可同時處理擴展事件 (擴充和縮減)。例如，您可以使用 [活動歷史](#) 來確認 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。您可以透過預測擴展來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：

- [使用 AWS 進行雲端運算](#)
- [Amazon EC2 執行個體類型](#)
- [Amazon ECS 容器：Amazon ECS 容器執行個體](#)
- [Amazon EKS 容器：Amazon EKS 工作節點](#)

- [函數：Lambda 函數組態](#)
- [Amazon EC2 執行個體的處理器狀態控制](#)
- [深入探討 Amazon ECS 叢集 Auto Scaling](#)
- [介紹 Karpenter - 一個開放原始碼的高效能 Kubernetes Cluster Autoscaler](#)

相關影片：

- [AWS re:Invent 2023 – AWS Graviton：AWS 工作負載的最佳性價比](#)
- [AWS re:Invent 2023 – AWS 管理主控台中新增 Amazon EC2 生成式 AI 功能](#)
- [AWS re:Invent 2023 – Amazon EC2 最新消息](#)
- [AWS re:Invent 2023 – 智慧優惠組合：Amazon EC2 成本最佳化策略](#)
- [AWS re:Invent 2021 – 支援下一代 Amazon EC2：深入探討 Nitro 系統](#)
- [AWS re:Invent 2019 – Amazon EC2 基礎](#)

相關範例：

- [Amazon EC2 Auto Scaling 群組範例](#)
- [Amazon EKS 研討會](#)
- [擴展 Amazon EKS 工作負載時搭配在 IPv6 上執行](#)

## PERF02-BP06 使用最佳化的硬體型運算加速器

使用硬體加速器執行特定功能，比以 CPU 為基礎的替代方案更有效率。

常見的反模式：

- 在工作負載中，您尚未基準化分析一般用途執行個體和專用執行個體，而專用執行個體可以提供更高的效能並降低成本。
- 您使用硬體型運算加速器來執行任務，比起使用以 CPU 為基礎的替代方案更有效率。
- 未監控 GPU 使用率。

建立此最佳實務的好處：透過使用硬體型加速器，例如圖形處理單元 (GPU) 和現場可程式化閘道陣列 (FPGA)，您就可以更有效率地執行特定處理功能。

未建立此最佳實務時的風險暴露等級：中

## 實作指引

加速運算執行個體可讓您使用硬體型運算加速器，例如 GPU 和 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，不需要時便會自動除役，以改善整體效能效率。

### 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) [所提供的效能功耗比最多會比同類 Amazon EC2 執行個體高出 50%](#)。
- 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 GPU 的 `utilization_gpu` 和 `utilization_memory` 等指標，如[使用 Amazon CloudWatch 收集 NVIDIA GPU 指標](#)中所示。
- 將硬體加速器的程式碼、網路運作和設定最佳化，以確保系統會充分利用基礎硬體。
  - [將 GPU 設定最佳化](#)
  - [Deep Learning AMI 中的 GPU 監控和最佳化](#)
  - [將 I/O 最佳化以針對 Amazon SageMaker 中的深度學習訓練來調校 GPU 效能](#)
- 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：

- [在 Amazon Elastic Container Service 上使用 GPU](#)
- [GPU 執行個體](#)
- [使用 AWS Trainium 的執行個體](#)
- [使用 AWS Inferentia 的執行個體](#)
- [開始建構吧！使用自訂晶片和加速器來進行建構](#)
  
- [加速運算](#)

- [Amazon EC2 VT1 執行個體](#)
- [如何為工作負載選擇適當的 Amazon EC2 執行個體類型？](#)
- [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：

- [AWS re:Invent 2021 - 如何為深度學習選取 Amazon Elastic Compute Cloud GPU 執行個體](#)
- [AWS re:Invent 2022 - \[全新推出！\] 介紹 AWS Inferentia2 型的 Amazon EC2 Inf2 執行個體](#)
- [AWS re:Invent 2022 - 利用 AWS Trainium 加速深度學習和創新](#)
- [AWS re:Invent 2022 - 利用 NVIDIA 進行 AWS 深度學習：從訓練到部署](#)

相關範例：

- [Amazon SageMaker 和 NVIDIA GPU Cloud \(NGC\)](#)
- [SageMaker 與 Trainium 和 Inferentia 一起用於最佳化深度學習訓練和推論工作負載](#)
- [在 Amazon SageMaker 中使用 Amazon Elastic Compute Cloud Inf1 執行個體最佳化 NLP 模型](#)



# 資料管理

特定系統的最佳資料管理解決方案會根據資料類型 (區塊、檔案或物件)、存取模式 (隨機或循序)、所需輸送量、存取頻率 (線上、離線、封存)、更新頻率 (WORM、動態) 及可用性和耐用性限制而有所不同。Well-Architected 工作負載會使用專用資料存放區，這些存放區採用不同的功能以提升效能。

這個重點領域分享了最佳化資料儲存、移動和存取模式，以及資料存放區效能達成效率的各種指引和最佳實務。

## 最佳實務

- [PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區](#)
- [PERF03-BP02 評估資料存放區可用的組態選項](#)
- [PERF03-BP03 收集並記錄資料存放區效能指標](#)
- [PERF03-BP04 實作策略以改善資料存放區中的查詢效能](#)
- [PERF03-BP05 實作利用快取的資料存取模式](#)

## PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區

了解資料特性 (例如可共用、大小、快取大小、存取模式、延遲、輸送量和資料的持續性)，為工作負載選擇適合的專用資料存放區 (儲存或資料庫)。

常見的反模式：

- 由於具備某種特定類型資料庫解決方案的內部經驗和知識，您堅持使用某個資料存取區。
- 您假設所有工作負載都有類似的資料儲存和存取需求。
- 您未實作資料目錄以清查資料資產。

建立此最佳實務的好處：了解資料特性和需求，可協助您判斷能滿足工作負載需求的最有效率且效能最高的儲存技術。

未建立此最佳實務時的風險暴露等級：高

## 實作指引

選取和實作資料儲存時，請確定查詢、擴展和儲存特性能滿足工作負載資料需求。AWS 提供多種資料儲存和資料庫技術，包括區塊儲存、物件儲存、串流儲存、檔案系統、關聯式、鍵值、文件、記憶體

內、圖形、時間序列和帳本資料庫。每個資料管理解決方案都有選項和組態，可供您支援使用案例和資料模型。透過了解資料特性和需求，您就可以擺脫單一儲存技術和一體適用的限制性方法，專注於如何適當管理資料。

## 實作步驟

- 對工作負載現有的各種資料類型執行清查。
- 了解並記錄資料特性和需求，包括：
  - 資料類型 (非結構化、半結構化、關聯式)
  - 資料量與成長
  - 資料耐用性：持續性、暫時性、臨時
  - ACID (單元性、一致性、隔離行為、持續性) 需求
  - 資料存取模式 (大量讀取或大量寫入)
  - 延遲
  - 輸送量
  - IOPS (每秒輸入/輸出操作次數)
  - 資料保留期
- 了解 AWS 上可用於工作負載的不同資料存放區 (儲存和資料庫服務)，這些資料存放區可以滿足資料特性 (詳情請參閱[PERF01-BP01 了解可用的雲端服務和特徵](#))。AWS 儲存技術及其重要特性的一些範例包含：

| 類型                | AWS 服務                            | 重要特性  |
|-------------------|-----------------------------------|---|
| Object storage    | <a href="#">Amazon S3</a>         | Unlimited scalability, high availability, and multiple options for accessibility. Transferring and accessing objects in and out of Amazon S3 can use a service, such as <a href="#">Transfer Acceleration</a> or <a href="#">存取點</a> , to support your location, security needs, and access patterns. |
| Archiving storage | <a href="#">Amazon S3 Glacier</a> | Built for data archiving.   |

| 類型                 | AWS 服務   | 重要特性  |
|--------------------|--|---|
| Streaming storage  | <a href="#">Amazon Kinesis</a><br><a href="#">Amazon Managed Streaming for Apache Kafka (Amazon MSK)</a> | Efficient ingestion and storage of streaming data.  |
| Shared file system | <a href="#">Amazon Elastic File System (Amazon EFS)</a>  | 可供多種類型的運算解決方案存取的可掛載檔案系統。  |
| Shared file system | <a href="#">Amazon FSx</a>   | Built on the latest AWS compute solutions to support four commonly used file systems: NetApp ONTAP, OpenZFS, Windows File Server, and Lustre. Amazon FSx <a href="#">延遲、輸送量和 IOPS</a> vary per file system and should be considered when selecting the right file system for your workload needs. |
| Block storage      | <a href="#">Amazon Elastic Block Store (Amazon EBS)</a>  | Scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS includes SSD-backed storage for transactional, IOPS-intensive workloads and HDD-backed storage for throughput-intensive workloads.  |

| 類型                  | AWS 服務   | 重要特性  |
|---------------------|--|---|
| Relational database | <a href="#">Amazon Aurora</a> , <a href="#">Amazon RDS</a> , <a href="#">Amazon Redshift</a> . | Designed to support ACID (atomicity, consistency, isolation, durability) transactions, and maintain referential integrity and strong data consistency. Many traditional applications, enterprise resource planning (ERP), customer relationship management (CRM), and ecommerce use relational databases to store their data. |
| Key-value database  | <a href="#">Amazon DynamoDB</a>  | Optimized for common access patterns, typically to store and retrieve large volumes of data. High-traffic web apps, ecommerce systems, and gaming applications are typical use-cases for key-value databases.   |
| Document database   | <a href="#">Amazon DocumentDB</a>  | Designed to store semi-structured data as JSON-like documents. These databases help developers build and update applications such as content management, catalogs, and user profiles quickly.   |

| 類型                   | AWS 服務  | 重要特性  |
|----------------------|---|---|
| In-memory database   | <a href="#">Amazon ElastiCache</a> ,<br><a href="#">Amazon MemoryDB for Redis</a> | Used for applications that require real-time access to data, lowest latency and highest throughput. You may use in-memory databases for application caching, session management, gaming leaderboards, low latency ML feature store, microservices messaging system, and a high-throughput streaming mechanism |
| Graph database       | <a href="#">Amazon Neptune</a>  | Used for applications that must navigate and query millions of relationships between highly connected graph datasets with millisecond latency at large scale. Many companies use graph databases for fraud detection , social networking, and recommendation engines.   |
| Time Series database | <a href="#">Amazon Timestream</a>   | Used to efficiently collect, synthesize, and derive insights from data that changes over time. IoT applications, DevOps, and industrial telemetry can utilize time-series databases.  |

| 類型          | AWS 服務   | 重要特性   |
|-------------|--|--|
| Wide column | <a href="#">Amazon Keyspaces (適用於 Apache Cassandra)</a>      | Uses tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table. You typically see a wide column store in high scale industrial apps for equipment maintenance, fleet management, and route optimization. |
| Ledger      | <a href="#">Amazon Quantum Ledger Database (Amazon QLDB)</a> | Provides a centralized and trusted authority to maintain a scalable, immutable, and cryptographically verifiable record of transactions for every application. We see ledger databases used for systems of record, supply chain, registrations, and even banking transactions.             |

- 如果您要建置資料平台，請利用 AWS 上的 [現代資料架構](#)，來整合資料湖、資料倉儲和專用的資料存放區。
- 為工作負載選擇資料存放區時，需要考慮的關鍵問題如下：

| Question                    | Things to consider  |
|-----------------------------|---|
| How is the data structured? | <ul style="list-style-type: none"> <li>• 如果資料是非結構化的，請考慮 <a href="#">Amazon S3</a> 之類的物件存放區，或 <a href="#">Amazon DocumentDB</a> 之類的 NoSQL 資料庫</li> </ul> |

| Question   | Things to consider   |
|--|--|
|  | <ul style="list-style-type: none"> <li>• 若為鍵值資料，請考慮 <a href="#">DynamoDB</a>、<a href="#">Amazon ElastiCache (Redis OSS)</a> 或 <a href="#">Amazon MemoryDB</a></li> </ul>   |
| What level of referential integrity is required?   | <ul style="list-style-type: none"> <li>• 若為外部索引鍵限制，關聯式資料庫 (例如 <a href="#">Amazon RDS</a> 和 <a href="#">Aurora</a>) 可以提供這種層級的完整性。</li> <li>• 通常，在 NoSQL 資料模型內，您會將資料去正規化為單一文件或文件集合，以便在單一請求中擷取，而不是跨文件或資料表聯結。</li> </ul>                     |
| Is ACID (atomicity, consistency, isolation, durability) compliance required?   | <ul style="list-style-type: none"> <li>• 如果需要與關聯式資料庫相關聯的 ACID 屬性，請考慮關聯式資料庫，例如 <a href="#">Amazon RDS</a> 和 <a href="#">Aurora</a>。</li> <li>• 如果 <a href="#">NoSQL 資料庫</a> 需要強大的一致性，您可以使用 <a href="#">DynamoDB</a> 搭配高度一致性讀取。</li> </ul> |
| How will the storage requirements change over time? How does this impact scalability?                                | <ul style="list-style-type: none"> <li>• <a href="#">DynamoDB</a> 和 <a href="#">Amazon Quantum Ledger Database (Amazon QLDB)</a> 等無伺服器資料庫將動態擴展。</li> <li>• 關聯式資料庫在佈建的儲存上具有上限，一旦達到這些限制，通常必須透過碎片化這類機制進行水平分割。</li> </ul>                    |
| What is the proportion of read queries in relation to write queries? Would caching be likely to improve performance? | <ul style="list-style-type: none"> <li>• 包含大量讀取作業的工作負載可從快取層中受益，例如 <a href="#">ElastiCache</a> 或 <a href="#">DAX</a> (如果資料庫是 DynamoDB)。</li> <li>• 讀取也可以卸載至具有關聯式資料庫的讀取複本，例如 <a href="#">Amazon RDS</a>。</li> </ul>                        |

| Question  | Things to consider   |
|---|--|
| Does storage and modification (OLTP - Online Transaction Processing) or retrieval and reporting (OLAP - Online Analytical Processing) have a higher priority? | <ul style="list-style-type: none"> <li>對於高輸送量讀取按原狀交易處理，請考慮使用 NoSQL 資料庫，例如 DynamoDB。</li> <li>對於具有一致性的高輸送量和複雜的讀取模式 (如聯結)，請使用 Amazon RDS。</li> <li>對於分析查詢，請考慮使用單欄式資料庫，例如 <a href="#">Amazon Redshift</a> 或使用 <a href="#">Athena</a> 或 <a href="#">Amazon QuickSight</a> 將資料匯出至 Amazon S3 和執行分析。</li> </ul> |
| What level of durability does the data require?   | <ul style="list-style-type: none"> <li>Aurora 會自動跨區域內的三個可用區域複寫資料，這表示資料可以耐久，因而降低資料遺失的機會。</li> <li>DynamoDB 會自動跨多個可用區域進行複寫，這會提供高可用性和資料耐久性。</li> <li>Amazon S3 提供 11 個 9 的耐用性。許多資料庫服務 (例如 Amazon RDS 和 DynamoDB) 支援將資料匯出至 Amazon S3，進行長期保留和封存。</li> </ul>   |
| Is there a desire to move away from commercial database engines or licensing costs?   | <ul style="list-style-type: none"> <li>考慮開放原始碼引擎，例如 Amazon RDS 或 Aurora 上的 PostgreSQL 和 MySQL。</li> <li>運用 <a href="#">AWS Database Migration Service</a> 和 <a href="#">AWS Schema Conversion Tool</a>，從商務資料庫引擎遷移至開放原始碼</li> </ul>   |
| What is the operational expectation for the database? Is moving to managed services a primary concern?  | <ul style="list-style-type: none"> <li>利用 Amazon RDS (而非 Amazon EC2)，和 DynamoDB 或 Amazon DocumentDB (而非自行託管 NoSQL 資料庫)，可以降低營運負擔。</li> </ul>  |



| Question   | Things to consider  |
|--|---|
| How is the database currently accessed? Is it only application access, or are there business intelligence (BI) users and other connected off-the-shelf applications? | <ul style="list-style-type: none"><li>• 如果您依賴外部工具，則可能必須維護與其所支援資料庫的相容性。Amazon RDS 完全與其支援的不同引擎版本相容，包括 Microsoft SQL Server、Oracle、MySQL 和 PostgreSQL。</li></ul> |

- 在非生產環境中執行試驗和基準化分析，以找出哪個資料存放區能滿足工作負載需求。

## 資源

相關文件：

- [Amazon EBS 磁碟區類型](#)
- [Amazon EC2 儲存](#)
- [Amazon EFS : Amazon EFS 效能](#)
- [Amazon FSx for Lustre 效能](#)
- [Amazon FSx for Windows File Server 效能](#)
- [Amazon S3 Glacier : S3 Glacier 文件](#)
- [Amazon S3 : 請求率和效能考量](#)
- [AWS 的雲端儲存](#)
- [Amazon EBS I/O 特性](#)
- [AWS 的雲端資料庫](#)
- [AWS 資料庫快取](#)
- [DynamoDB Accelerator](#)
- [Amazon Aurora 最佳實務](#)
- [Amazon Redshift 效能](#)
- [Amazon Athena 10 大效能秘訣](#)
- [Amazon Redshift Spectrum 最佳實務](#)
- [Amazon DynamoDB 最佳實務](#)
- [在 Amazon EC2 和 Amazon RDS 之間選擇](#)
- [實作 Amazon ElastiCache 的最佳實務](#)

## 相關影片：

- [AWS re:Invent 2023：提高 Amazon Elastic Block Store 效率並更具成本效益](#)
- [AWS re:Invent 2023：利用 Amazon Simple Storage Service 最佳化儲存價格和效能](#)
- [AWS re:Invent 2023：在 Amazon Simple Storage Service 上建置資料湖並進行最佳化](#)
- [AWS re:Invent 2022：在 AWS 上建置現代化資料架構](#)
- [AWS re:Invent 2022：在 AWS 上建置資料網格架構](#)
- [AWS re:Invent 2023：深入探索 Amazon Aurora 及其創新](#)
- [AWS re:Invent 2023：使用 Amazon DynamoDB 的進階資料建模](#)
- [AWS re:Invent 2022：透過專用資料庫建置現代化應用程式](#)
- [Amazon DynamoDB 深入探討：進階設計模式](#)

## 相關範例：

- [AWS 專用資料庫研討會](#)
- [專為開發人員打造的資料庫](#)
- [AWS 現代資料架構 Immersion Day](#)
- [在 AWS 上建置資料網格](#)
- [Amazon S3 範例](#)
- [使用 Amazon Redshift 資料共用來最佳化資料模式](#)
- [資料庫遷移](#)
- [MS SQL Server - AWS Database Migration Service \(AWS DMS\) 複寫示範](#)
- [資料庫現代化實際操作研討會](#)
- [Amazon Neptune 範例](#)

## PERF03-BP02 評估資料存放區可用的組態選項

了解並評估資料存放區可用的各種特徵和組態選項，以最佳化工作負載的儲存空間和效能。

### 常見的反模式：

- 所有工作負載只能使用一種儲存類型，例如 Amazon EBS。

- 您為所有工作負載使用佈建 IOPS，卻未針對所有儲存層進行實際測試。
- 您未意識到所選資料庫管理解決方案的組態選項。
- 僅依靠增加執行個體大小，而不查看其他可用的組態選項。
- 未測試資料存放區的擴展特性。

建立此最佳實務的好處：藉由探索和試驗資料存放區組態，您能夠降低基礎架構成本、改善效能，以及減少維護工作負載所需的工作量。

未建立此最佳實務時的風險暴露等級：中

## 實作指引

工作負載可以根據資料儲存和存取需求使用一或多個資料存放區。若要最佳化效能效率和成本，您必須評估資料存取模式，以判斷適當的資料存放區組態。在探索資料存放區選項時，請考量各種層面，例如儲存選項、記憶體、運算、讀取複本、一致性需求、連線共用，以及快取選項。試驗這些不同的組態選項來改善效能效率指標。

## 實作步驟

- 了解資料存放區的目前組態 (例如執行個體類型、儲存大小或資料庫引擎版本)。
- 檢閱 AWS 文件和最佳實務，以了解可協助改善資料存放區效能的建議組態選項。要考慮的關鍵資料存放區選項如下：

| Configuration option                              | Examples   |
|---|--|
| Offloading reads (like read replicas and caching) | <ul style="list-style-type: none"> <li>• 若為 DynamoDB 資料表，您可以使用 DAX 卸載讀取，以進行快取。</li> <li>• 您可以建立 Amazon ElastiCache (Redis OSS) 叢集，並將應用程式設定為先從快取中讀取，如果請求的項目不存在，則退回到資料庫。</li> <li>• 關聯式資料庫 (例如 Amazon RDS 和 Aurora) 和已佈建的 NoSQL 資料庫 (例如 Neptune 和 Amazon DocumentDB) 全都支援新增讀取複本，以卸載工作負載的讀取部分。</li> </ul> |

| Configuration option  | Examples   |
|---|--|
| Scaling writes (like partition key sharding or introducing a queue) | <ul style="list-style-type: none"><li>• 無伺服器資料庫 (例如 DynamoDB) 將自動擴展。確定您已佈建足夠的讀取容量單位 (RCU) 來處理工作負載。</li><li>• 對於關聯式資料庫，您可以增加執行個體的大小，以適應增加的工作負載或增加佈建 IOPS，以允許增加基礎儲存的輸送量。</li><li>• 您也可以直接在資料庫前面引進佇列，而不是直接寫入至資料庫。此模式允許您將擷取與資料庫分離並控制流量，因此資料庫不會癱瘓。</li><li>• 批次處理寫入請求而不是建立許多短期交易，有助改善高寫入量關聯式資料庫中的輸送量。</li><li>• 取決於容量模式，DynamoDB 之類的無伺服器資料庫可以自動擴展寫入輸送量，或透過調整已佈建的容量單位 (WCU) 來進行。</li><li>• 當您達到指定分區索引鍵的輸送量限制時，仍可能會遇到熱分區的問題。這可以透過選擇更均勻分佈的分區索引鍵，或對分區索引鍵進行寫入碎片化來緩解。</li></ul> |
| Policies to manage the lifecycle of your datasets                   | <ul style="list-style-type: none"><li>• 您可以使用 <a href="#">Amazon S3 生命週期</a>，以在物件的整個生命週期中管理物件。如果存取模式不明、會變化或是無法預測，則可以使用 <a href="#">Amazon S3 Intelligent-Tiering</a>，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 <a href="#">Amazon S3 Storage Lens</a> 指標，來找出生命週期管理中的最佳化機會和落差。</li><li>• <a href="#">Amazon EFS 生命週期管理</a> 會自動管理檔案系統的檔案儲存。</li></ul>  |

| Configuration option              | Examples   |
|-----------------------------------|--|
| Connection management and pooling | <ul style="list-style-type: none"><li>• Amazon RDS Proxy 可以與 Amazon RDS 和 Aurora 搭配使用，以管理資料庫的連線。</li><li>• 無伺服器資料庫 (例如 DynamoDB) 沒有與其相關聯的連線，但會考慮佈建的容量和自動擴展政策來處理負載中的峰值。</li></ul> |

- 在非生產環境中執行試驗和基準化分析，以找出哪個組態選項能滿足工作負載需求。
- 完成試驗之後，請規劃遷移並確認效能指標。
- 使用 AWS 監控 (例如 [Amazon CloudWatch](#)) 和最佳化 (例如 [Amazon S3 Storage Lens](#)) 工具，以透過實際使用模式持續最佳化資料存放區。

## 資源

相關文件：

- [AWS 的雲端儲存](#)
- [Amazon EBS 磁碟區類型](#)
- [Amazon EC2 儲存](#)
- [Amazon EFS : Amazon EFS 效能](#)
- [Amazon FSx for Lustre 效能](#)
- [Amazon FSx for Windows File Server 效能](#)
- [Amazon S3 Glacier : S3 Glacier 文件](#)
- [Amazon S3 : 請求率和效能考量](#)
- [Amazon EBS I/O 特性](#)
- [AWS 的雲端資料庫](#)
- [AWS 資料庫快取](#)
- [DynamoDB Accelerator](#)
- [Amazon Aurora 最佳實務](#)
- [Amazon Redshift 效能](#)
- [Amazon Athena 10 大效能秘訣](#)

- [Amazon Redshift Spectrum 最佳實務](#)
- [Amazon DynamoDB 最佳實務](#)

相關影片：

- [AWS re:Invent 2023：提高 Amazon Elastic Block Store 效率並更具成本效益](#)
- [AWS re:Invent 2023：利用 Amazon Simple Storage Service 最佳化儲存價格和效能](#)
- [AWS re:Invent 2023：在 Amazon Simple Storage Service 上建置資料湖並進行最佳化](#)
- [AWS re:Invent 2023：AWS 檔案儲存最新消息](#)
- [AWS re:Invent 2023：深入探索 Amazon DynamoDB](#)

相關範例：

- [AWS 專用資料庫研討會](#)
- [專為開發人員打造的資料庫](#)
- [AWS 現代資料架構 Immersion Day](#)
- [Amazon EBS 自動擴展](#)
- [Amazon S3 範例](#)
- [Amazon DynamoDB 範例](#)
- [AWS 資料庫遷移範例](#)
- [資料庫現代化研討會](#)
- [使用 Amazon RDS for Postgress 資料庫上的參數](#)

## PERF03-BP03 收集並記錄資料存放區效能指標

追蹤並記錄資料存放區的相關效能指標，以了解資料管理解決方案的成效。這些指標可協助您最佳化資料存放區、確認是否符合工作負載需求，並提供工作負載執行方式的清晰概觀。

常見的反模式：

- 您只使用手動日誌檔案來搜尋指標。
- 您只會將指標發佈至您團隊所使用的內部工具，而沒有工作負載的全貌。
- 您只會使用所選監控軟體記錄的預設指標。

- 您只會在有問題時審查指標。
- 您只會監控系統層級指標，而不會擷取資料存取或用量指標。

建立此最佳實務的優勢：建立效能基準可協助您了解正常行為和工作負載的要求。可以更快地識別和偵錯異常模式，進而改善資料存放區的效能和可靠性。

未建立此最佳實務時的曝險等級：高

## 實作指引

若要監控資料存放區的效能，您必須記錄一段時間的多個效能指標。這可讓您偵測異常，以及針對業務指標測量效能，以確認是否符合工作負載需求。

指標應該同時包括支援資料存放區的基礎系統和資料庫指標。基礎系統指標可能包括 CPU 使用率、記憶體、可用磁碟儲存、磁碟 I/O、快取命中率和網路傳入和傳出指標，而資料存放區指標可能包括每秒交易數、熱門查詢、平均查詢率、回應時間、索引使用情況、表格鎖定、查詢逾時，以及開啟的連線數目。此資料對於了解工作負載的執行方式，以及資料管理解決方案的使用方式至關重要。將這些指標納入資料驅動的方法，以調整和最佳化工作負載的資源。

使用工具、程式庫和系統來記錄與資料庫效能有關的效能測量值。

## 實作步驟

1. 找出要追蹤的資料存放區關鍵效能指標。
  - a. [Amazon S3 指標和維度](#)
  - b. [監控 Amazon RDS 執行個體中的指標](#)
  - c. [在 Amazon RDS 上使用 Performance Insights 監控資料庫負載](#)
  - d. [增強型監控概觀](#)
  - e. [DynamoDB 指標和維度](#)
  - f. [監控 DynamoDB Accelerator](#)
  - g. [使用 Amazon CloudWatch 進行監控 Amazon MemoryDB](#)
  - h. [我應該監控哪些指標？](#)
  - i. [監控 Amazon Redshift 叢集效能](#)
  - j. [Timestream 指標和維度](#)
  - k. [Amazon Aurora Amazon CloudWatch 指標](#)
  - l. [在 Amazon Keyspaces \(for Apache Cassandra\) 中的記錄和監控](#)

- m. [監控 Amazon Neptune 資源](#)
2. 使用核准的記錄和監控解決方案來收集這些指標。 [Amazon CloudWatch](#) 可以收集架構中各種資源的指標。您還可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方解決方案，來設定可指出何時超過閾值的警示。
3. 檢查資料存放區監控是否能從可偵測效能異常的機器學習解決方案中獲益。
  - a. [Amazon RDS Amazon DevOps Guru](#) 可讓您查看效能問題，並做出更正動作的建議。
4. 在監控和記錄解決方案中設定資料保留，以符合安全性和營運目標。
  - a. [CloudWatch 指標的預設資料保留](#)
  - b. [CloudWatch Logs 的預設資料保留](#)

## 資源

相關文件：

- [AWS 資料庫快取](#)
- [Amazon Athena 10 大效能秘訣](#)
- [Amazon Aurora 最佳實務](#)
- [DynamoDB Accelerator](#)
- [Amazon DynamoDB 最佳實務](#)
- [Amazon Redshift Spectrum 最佳實務](#)
- [Amazon Redshift 效能](#)
- [AWS 的雲端資料庫](#)
- [Amazon RDS Performance Insights](#)

相關影片：

- [AWS re:Invent 2022 - 使用 Amazon RDS 與 Aurora 進行效能監控，提供 Autodesk](#)
- [使用 Amazon DevOps Guru for Amazon RDS 進行資料庫效能監控和調整](#)
- [AWS re:Invent 2023 - AWS 檔案儲存最新消息](#)
- [AWS re:Invent 2023 - 深入探索 Amazon DynamoDB](#)
- [AWS re:Invent 2023 - 建置與最佳化 Amazon S3 上的資料湖](#)
- [AWS re:Invent 2023 - AWS 檔案儲存最新消息](#)



- [AWS re:Invent 2023 - 深入探索 Amazon DynamoDB](#)
- [在 Amazon ElastiCache 上監控 Redis 工作負載的最佳實務](#)

相關範例：

- [AWS 資料集擷取指標收集架構](#)
- [Amazon RDS 監控研討會](#)
- [AWS 專用資料庫研討會](#)

## PERF03-BP04 實作策略以改善資料存放區中的查詢效能

實作策略以最佳化資料並改善資料查詢，以便為工作負載提供更高的可擴展性和更高效的效能。

常見的反模式：

- 您未分割資料存放區中的資料。
- 您在資料存放區中只使用一種檔案格式儲存資料。
- 您未在資料存放區中使用索引。

建立此最佳實務的優勢：最佳化資料和查詢效能可提高效率、降低成本並改善使用者體驗。

未建立此最佳實務時的曝險等級：中

### 實作指引

資料最佳化和查詢調整是資料存放區中效能效率的關鍵層面，因為其會影響整個雲端應用程式工作負載的效能和回應能力。未經過最佳化的查詢可能會使用更多資源和造成更大的瓶頸，進而降低資料存放區的整體效率。

資料最佳化包括數個技術，以確保高效的資料儲存和存取。這也有助於改善資料存放區中的查詢效能。關鍵策略包括資料分割、資料壓縮和資料去常規化，這些有助於資料的儲存和存取達到最佳化。

### 實作步驟

- 了解和分析在資料存放區中執行的重要資料查詢。
- 找出資料存放區中速度緩慢的查詢執行，並使用查詢計畫了解其目前狀態。
  - [分析 Amazon Redshift 中的查詢計畫](#)

- [在 Athena 中使用 EXPLAIN 和 EXPLAIN ANALYZE](#)
- 實作策略以改善查詢效能。有些關鍵策略包括下列情況：
  - 使用 [單欄檔案格式](#) (例如, Parquet 或 ORC)。
  - 壓縮資料存放區中的資料以減少儲存空間和 I/O 作業。
  - 資料分割可將資料拆分為較小的部分, 進而縮短資料掃描時間。
    - [在 Athena 中分割資料](#)
    - [資料分割和資料分發](#)
  - 在查詢中對共同欄進行資料索引編制。
  - 使用具體化視觀表進行頻繁查詢。
    - [了解具體化視觀表](#)
    - [在 Amazon Redshift 中建立具體化視觀表](#)
  - 選擇正確的聯結作業以進行查詢。當您聯結兩張資料表時, 請指定聯結左側為較大資料表, 並指定聯結右側為較小資料表。
  - 分散式快取解決方案可改善延遲並減少資料庫 I/O 操作的次數。
  - 定期維護, 例如執行統計。
- 在非生產環境中試驗和測試策略。

## 資源

相關文件：

- [Amazon Aurora 最佳實務](#)
- [Amazon Redshift 效能](#)
- [Amazon Athena 10 大效能秘訣](#)
- [AWS 資料庫快取](#)
- [實作 Amazon ElastiCache 的最佳實務](#)
- [在 Athena 中分割資料](#)

相關影片：

- [AWS re:Invent 2023 - AWS 儲存成本最佳化最佳實務](#)
- [AWS re:Invent 2022 - 使用 Amazon RDS 與 Aurora 進行效能監控, 提供 Autodesk](#)

- [使用新的查詢分析工具最佳化 Amazon Athena 查詢](#)

相關範例：

- [Amazon S3 Select - 直接查詢資料而不需要執行伺服器或資料庫](#)
- [AWS 專用資料庫研討會](#)

## PERF03-BP05 實作利用快取的資料存取模式

為快速擷取經常存取的資料，實作利用快取的存取模式。

常見的反模式：

- 您快取的資料經常變更。
- 您以為快取的資料能夠持久儲存且永遠可用，因而過於依賴。
- 您未考慮快取資料的一致性。
- 您未監控快取實作的效率。

建立此最佳實務的優勢：將資料儲存在快取中可改善讀取延遲、讀取輸送量、使用者體驗和整體效率，並降低成本。

未建立此最佳實務時的曝險等級：中

### 實作指引

快取是軟體或硬體的元件，其用途在於儲存資料，以便未來請求相同的資料時，能夠更快且更有效率地提供服務。若儲存在快取中的資料遺失，可以透過重複先前的計算或從其他資料存放區提取的方式重新建構該資料。

資料快取可能是改善整體應用程式效能並減輕基礎主要資料來源負擔的最有效策略之一。資料可在應用程式中的多個層級快取，例如在進行遠端呼叫的應用程式內，稱為用戶端快取，或者使用快速的次要服務來儲存資料，稱為遠端快取。

#### 用戶端快取

透過用戶端快取，每個用戶端 (查詢後端資料儲存的應用程式或服務) 都可將特定的查詢結果儲存在本機上，並在指定的時間內保留。這樣就能透過先查看本機用戶端快取，減少整體網路對資料儲存的請求數量。如果結果不存在，應用程式就可以查詢資料儲存，並將這些結果儲存在本機上。此模式可讓每個

用戶端將資料儲存在最靠近的位置 (用戶端本身)，進而將延遲降至最低。用戶端也可在後端資料儲存無法使用時，繼續為部分查詢提供服務，以提高整體系統的可用性。

這種方法的缺點是，若有多個用戶端，則可能會將相同的快取資料儲存在本機上。這樣會導致這些用戶端之間發生重複使用儲存和資料不一致的情況。某一個用戶端可能快取了查詢的結果，而過了一分鐘後，另一個用戶端也可能執行相同的查詢，但得到不同的結果。

## 遠端快取

為了解決用戶端之間資料重複的問題，可使用快速的外部服務 (也稱為 遠端快取) 來儲存查詢的資料。每個用戶端都會在查詢後端資料儲存之前查看遠端快取，而不會查看本機資料存放區。這種策略可讓用戶端之間的回應更趨一致、提高儲存資料的效率，以及增加快取的資料量，因為儲存空間的擴展與用戶端無關。

遠端快取的缺點是，整個系統可能產生較高的延遲，因為需要額外的網路跳轉來查看遠端快取。用戶端快取可與遠端快取一起使用，以提供多層次快取來改善延遲。

## 實作步驟

1. 找出可有效利用快取的資料庫、API 和網路服務。有大量讀取工作負載、高讀寫比例或擴展成本昂貴的服務，都適合使用快取。
  - [資料庫快取](#)
  - [啟用 API 快取以提升回應能力](#)
2. 確定最適合您存取模式的適當快取策略類型。
  - [快取策略](#)
  - [AWS 快取解決方案](#)
3. 遵循適合您的資料存放區的 [快取最佳實務](#)。
4. 為所有資料設定快取失效策略，例如存留時間 (TTL)，以便在資料時效性與減輕後端資料儲存壓力之間取得平衡。
5. 在用戶端中啟用像是自動連線重試、指數退避、用戶端逾時和連線共用等功能 (如果有的話)，因為這些功能可以改善效能和可靠性。
  - [最佳實務：Redis 用戶端和 Amazon ElastiCache \(Redis OSS\)](#)
6. 設定 80% 或更高的目標來監控快取命中率。較低的值可能表示快取大小不足，或存取模式無法有效利用快取。
  - [我應該監控哪些指標？](#)
  - [在 Amazon ElastiCache 上監控 Redis 工作負載的最佳實務](#)

- [使用 Amazon CloudWatch 監控 Amazon ElastiCache \(Redis OSS\) 的最佳實務](#)
7. 實作 [資料複寫](#) 將讀取卸載到多個執行個體，並改善資料讀取效能和可用性。

## 資源

相關文件：

- [使用 Amazon ElastiCache Well-Architected Lens](#)
- [使用 Amazon CloudWatch 監控 Amazon ElastiCache \(Redis OSS\) 的最佳實務](#)
- [我應該監控哪些指標？](#)
- [使用 Amazon ElastiCache 大幅提高效能白皮書](#)
- [快取挑戰和策略](#)

相關影片：

- [Amazon ElastiCache 學習路徑](#)
- [專為使用 Amazon ElastiCache 最佳實務獲得成功而設計](#)
- [AWS re:Invent 2020 - 專為使用 Amazon ElastiCache 最佳實務獲得成功而設計](#)
- [AWS re:Invent 2023 - \[LAUNCH\] 向您介紹 Amazon ElastiCache Serverless](#)
- [AWS re:Invent 2022 - 使用 Redis 重新塑造資料層的 5 種好方法](#)
- [AWS re:Invent 2021 - 深入探討 Amazon ElastiCache \(Redis OSS\)](#)

相關範例：

- [使用 Amazon ElastiCache \(Redis OSS\) 大幅提高 MySQL 資料庫效能](#)

# 網路與內容交付

工作負載的最佳聯網解決方案會根據延遲、輸送量需求、抖動和頻寬而有所不同。實體限制 (例如使用者或內部部署資源) 會決定位置選項。這些限制可能隨著邊緣節點或資源位置而有所差異。

在 AWS 上，聯網以虛擬化方式存在，並提供多種不同的類型和組態。如此就能更容易滿足您的聯網需求。AWS 提供了多種產品功能 (例如，增強型聯網、經 Amazon EC2 聯網最佳化的執行個體、Amazon S3 Transfer Acceleration 和動態 Amazon CloudFront)，可最佳化網路流量。AWS 還提供了聯網功能 (例如，Amazon Route 53 延遲路由、Amazon VPC 端點、AWS Direct Connect 和 AWS Global Accelerator)，可減少網路距離或抖動。

這個重點領域分享了在雲端中設計、設定和操作高效網路和內容交付解決方案的各種指引和最佳實務。

## 最佳實務

- [PERF04-BP01 了解聯網如何影響效能](#)
- [PERF04-BP02 評估可用的聯網功能](#)
- [PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN](#)
- [PERF04-BP04 使用負載平衡將流量分配到多個資源](#)
- [PERF04-BP05 選擇網路通訊協定以提高效能](#)
- [PERF04-BP06 根據網路需求選擇工作負載的位置](#)
- [PERF04-BP07 根據指標最佳化網路組態](#)

## PERF04-BP01 了解聯網如何影響效能

分析並了解網路相關決策如何影響您的工作負載，以提供高效率的效能並改善使用者體驗。

常見的反模式：

- 通過現有資料中心的所有流量。
- 您讓所有流量路由經過中央防火牆，而非使用雲端原生網路安全工具。
- 您佈建 AWS Direct Connect 連線，但不了解實際用量需求。
- 在定義聯網解決方案時，您未考慮工作負載特性和加密負擔。
- 您將內部部署概念和策略用於雲端中的聯網解決方案。

建立此最佳實務的優勢：了解聯網如何影響工作負載效能協助您識別潛在的瓶頸、改善使用者體驗、提高可靠性，以及隨著工作負載的變更降低營運維護成本。

未建立此最佳實務時的曝險等級：高

## 實作指引

網路負責處理應用程式元件、雲端服務、邊緣網路和內部部署資料之間的連線，因此對工作負載效能可能有大幅的影響。除了工作負載效能外，使用者體驗也可能受到網路延遲、頻寬、通訊協定、位置、網路擁塞、抖動、輸送量和路由規則的影響。

具有來自工作負載的聯網要求的記錄清單，包括延遲、封包大小、路由規則、通訊協定和支援的流量模式。檢閱可用的聯網解決方案，並識別哪個服務符合您的工作負載聯網特性。雲端型網路可以快速重建，因此隨著時間演進您的網路架構是改善效能達成效率的必要條件。

實作步驟：

1. 定義並記錄聯網效能需求，包括網路延遲、頻寬、通訊協定、位置、流量模式 (峰值和頻率)、輸送量、加密、檢查，以及路由規則等指標。
2. 了解關鍵 AWS 聯網服務，如 [VPC](#)、[AWS Direct Connect](#)、[Elastic Load Balancing \(ELB\)](#) 和 [Amazon Route 53](#)。
3. 擷取下列主要聯網特性：

| 特性       | 工具和指標  |
|----------|--|
| 基礎聯網特性   | <ul style="list-style-type: none"> <li>• <a href="#">VPC Flow Logs</a></li> <li>• <a href="#">AWS Transit Gateway Flow Logs</a></li> <li>• <a href="#">AWS Transit Gateway 指標</a></li> <li>• <a href="#">AWS PrivateLink 指標</a></li> </ul> |
| 應用程式聯網特性 | <ul style="list-style-type: none"> <li>• <a href="#">Elastic Fabric Adapter</a></li> <li>• <a href="#">AWS App Mesh 指標</a></li> <li>• <a href="#">Amazon API Gateway 指標</a></li> </ul>   |
| 邊緣聯網特性   | <ul style="list-style-type: none"> <li>• <a href="#">Amazon CloudFront 指標</a></li> <li>• <a href="#">Amazon Route 53 指標</a></li> <li>• <a href="#">AWS Global Accelerator 指標</a></li> </ul>  |

| 特性     | 工具和指標  |
|--------|--|
| 混合聯網特性 | <ul style="list-style-type: none"> <li>• <a href="#">AWS Direct Connect 指標</a></li> <li>• <a href="#">AWS Site-to-Site VPN 指標</a></li> <li>• <a href="#">AWS Client VPN 指標</a></li> <li>• <a href="#">AWS 雲端 WAN 指標</a></li> </ul>   |
| 安全聯網特性 | <ul style="list-style-type: none"> <li>• <a href="#">AWS Shield、AWS WAF 和 AWS Network Firewall 指標</a></li> </ul>   |
| 追蹤特性   | <ul style="list-style-type: none"> <li>• <a href="#">AWS X-Ray</a></li> <li>• <a href="#">VPC Reachability Analyzer</a></li> <li>• <a href="#">Network Access Analyzer</a></li> <li>• <a href="#">Amazon Inspector</a></li> <li>• <a href="#">Amazon CloudWatch RUM</a></li> </ul> |

#### 4. 對網路效能進行基準測試：

- a. [基準](#) 網路輸送量，當執行個體位於同一 VPC 時，有些因素可能會影響 Amazon EC2 網路效能。測量同一 VPC 中 Amazon EC2 Linux 執行個體之間的網路頻寬。
- b. 執行 [負載測試](#) 以試驗各種聯網解決方案和選項。

## 資源

### 相關文件：

- [Application Load Balancer](#)
- [Linux 上的 EC2 增強型聯網](#)
- [Windows 上的 EC2 增強型聯網](#)
- [EC2 置放群組](#)
- [在 Linux 執行個體上啟用搭配彈性網路轉接器 \(ENA\) 的增強型聯網](#)
- [Network Load Balancer](#)
- [AWS 的聯網產品](#)
- [Transit Gateway](#)
- [轉換到 Amazon Route 53 中以延遲為基礎的路由](#)



- [VPC 端點](#)

## 相關影片：

- [AWS re:Invent 2023 - AWS 聯網基礎](#)
- [AWS re:Invent 2023 - 聯網對您的應用程式有何助益？](#)
- [AWS re:Invent 2023 - 進階 VPC 設計及新功能](#)
- [AWS re:Invent 2023 - 開發人員的雲端聯網指南](#)
- [AWS re:Invent 2019 - 與 AWS 和混合 AWS 網路架構的連線能力](#)
- [AWS re:Invent 2019 - 最佳化 Amazon EC2 執行個體的網路效能](#)
- [AWS 線上高峰會 - 改善應用程式的全球網路效能](#)
- [AWS re:Invent 2020 - 搭配 Well-Architected Framework 的聯網最佳實務和秘訣](#)
- [AWS re:Invent 2020 - 大規模遷移中的 AWS 聯網最佳實務](#)

## 相關範例：

- [AWS Transit Gateway 和可擴展的安全解決方案](#)
- [AWS 聯網研討會](#)
- [實際操作網路防火牆研討會](#)
- [觀察和診斷在 AWS 上的網路](#)
- [尋找並解決在 AWS 上的網路設定錯誤](#)

## PERF04-BP02 評估可用的聯網功能

評估雲端中可能提升效能的聯網功能。透過測試、指標和分析來測量這些功能的影響。例如，利用可用的網路層級功能來降低延遲、網路距離或抖動。

## 常見的反模式：

- 您只在單一區域中活動，這是因為該區域是您總部的所在區域。
- 您使用防火牆而非安全群組來篩選流量。
- 您透過中斷 TLS 來檢測流量，而非使用安全群組、端點政策和其他雲端原生功能。
- 您只使用子網路來分隔，而非採用安全群組的方式。

建立此最佳實務的優勢：評估所有服務功能和選項可提高工作負載效能、降低基礎架構成本、減少維護工作負載所需的人力，以及提升整體安全狀態。您可以使用全球 AWS 骨幹來為客戶提供最佳的聯網體驗。

未建立此最佳實務時的曝險等級：高

## 實作指引

AWS 提供 [AWS Global Accelerator](#) 和 [Amazon CloudFront](#) 等服務，可協助提高網路效能，而大多數 AWS 服務都有產品功能 (例如 [Amazon S3 Transfer Acceleration](#) 功能) 可用來最佳化網路流量。

檢閱您可以使用哪些網路相關組態選項，及其對工作負載可能有何影響。效能最佳化取決於了解這些選項如何與您的架構互動，以及這些選項對衡量效能與使用者體驗的影響。

## 實作步驟

- 建立工作負載元件清單。
  - 考慮在建置統一的全球網路時，使用 [AWS 雲端 WAN](#) 來建置、管理和監控您組織的網路。
  - 監控您的全球和核心網路，方法是使用 [Amazon CloudWatch Logs 指標](#)。利用 [Amazon CloudWatch RUM](#)，它提供了洞見來幫助您識別、了解和強化使用者的數位體驗。
  - 檢視 AWS 區域與可用區域之間的彙總網路延遲，以及每個可用區域內的網路延遲，使用 [AWS Network Manager](#) 獲得洞見，讓您深入了解應用程式效能與基礎 AWS 網路效能的關係。
  - 使用現有的組態管理資料庫 (CMDB) 工具或 [AWS Config](#) 之類的服務，建立工作負載的庫存及了解其設定方式。
- 如果這是現有的工作負載，請識別並記載效能指標的基準，並將重心放在瓶頸和有待改善的領域上。每個工作負載的效能相關聯網指標，都會隨著業務要求和工作負載特性而不同。一開始對您的工作負載而言，下列指標可能都是必須檢閱的：頻寬、延遲、封包遺失、抖動和重新傳輸。
- 如果這是新的工作負載，請執行 [負載測試](#) 以找出效能瓶頸。
- 對於您找出的效能瓶頸，請檢閱您解決方案的組態選項，以找出改善效能的機會。參考下列主要聯網選項和功能：

### 改善機會

網路路徑或路由

### 解決方案

使用 [Network Access Analyzer](#) 識別路徑或路由。

| 改善機會   | 解決方案   |
|--------|--|
| 網路通訊協定 | 請參閱 <a href="#">PERF04-BP05 選擇網路通訊協定以提高效能</a>  |
| 網路拓撲   | <p>在連接多個帳戶時，評估 <a href="#">VPC 對等互連</a> 和 <a href="#">AWS Transit Gateway</a> 之間在操作和效能上的權衡。AWS Transit Gateway 會簡化所有 VPC 的互連方式，如此便可橫跨數千個 AWS 帳戶並進入內部部署網路。使用下列工具在多個帳戶間共用您的 AWS Transit Gateway：<a href="#">AWS Resource Access Manager</a>。</p> <p>請參閱 <a href="#">PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN</a></p>  |
| 網路服務   | <p><a href="#">AWS Global Accelerator</a> 是一項連網服務，可使用 AWS 全域網路基礎架構將使用者流量效能提升最多達 60%。</p> <p><a href="#">Amazon CloudFront</a> 可以改善全域工作負載內容交付和延遲的效能。</p> <p>使用 <a href="#">Lambda@edge</a> 執行多項功能，以自訂能讓 CloudFront 就近提供給使用者的內容、減少延遲並改善效能。</p> <p>Amazon Route 53 提供 <a href="#">以延遲為基礎的路由</a>、<a href="#">地理位置路由</a>、<a href="#">地理位置臨近性路由</a> 和 <a href="#">以 IP 為基礎的路由</a> 等選項，可協助您提升全球對象的工作負載效能。若您的工作負載分散於全球，請檢閱您的工作負載流量和使用者位置，找出能夠最佳化工作負載效能的路由選項。</p> |

| 改善機會   | 解決方案   |
|--------|--|
| 儲存資源功能 | <p><a href="#">Amazon S3 Transfer Acceleration</a> 功能讓外部使用者得以獲益於 CloudFront 的聯網最佳化，以將資料上傳到 Amazon S3。這樣就可以更輕易地從與 AWS 雲端 沒有專用連線的遠端位置輸送大量資料。</p> <p><a href="#">Amazon S3 多區域存取點</a> 可將內容複寫至多個區域，並提供單一存取點以簡化工作負載。使用多區域存取點時，您可以使用可識別最低延遲儲存貯體的服務，來要求資料或將資料寫入 Amazon S3。</p>  |
| 運算資源功能 | <p><a href="#">彈性網路介面 (ENA)</a> 為 Amazon EC2 執行個體、容器和 Lambda 函數所使用，會受到個別流程的限制。請檢閱您的置放群組以最佳化 <a href="#">EC2 聯網輸送量</a>。若要避免個別流程發生瓶頸，請將應用程式設計為使用多個流程。若要監控及掌握運算相關的聯網指標，請使用 CloudWatch 指標和 <a href="#">ethtool</a>。此 ethtool 命令包含在 ENA 驅動程式中，會將可發佈為 <a href="#">自訂指標</a> 的其他網路相關指標公開至 CloudWatch。</p> <p><a href="#">Amazon 彈性網路介面卡 (ENA)</a> 可提供進一步最佳化，具體方法是為您於叢集置放群組內的執行個體提供更理想的 <a href="#">輸送量</a>。</p> <p><a href="#">Elastic Fabric Adapter (EFA)</a> 是 Amazon EC2 執行個體的網路介面，可讓您在 AWS 上大規模執行需要高階節點間通訊的工作負載。</p> <p><a href="#">經 Amazon EBS 最佳化的執行個體</a> 使用最佳化的組態堆疊，可提供更多專用容量以增加 Amazon EBS I/O。</p> |

## 資源

### 相關文件：

- [Application Load Balancer](#)
- [Linux 上的 EC2 增強型聯網](#)
- [Windows 上的 EC2 增強型聯網](#)
- [EC2 置放群組](#)
- [在 Linux 執行個體上啟用搭配彈性網路轉接器 \(ENA\) 的增強型聯網](#)
- [Network Load Balancer](#)
- [搭配 AWS 的聯網產品](#)
- [轉換到 Amazon Route 53 中的 Latency-Based Routing](#)
- [VPC 端點](#)
- [VPC Flow Logs](#)

### 相關影片：

- [AWS re:Invent 2023 – 準備好踏出下一步了嗎？設計可促進成長與靈活性的網路](#)
- [AWS re:Invent 2023 – 進階 VPC 設計及新功能](#)
- [AWS re:Invent 2023 – 開發人員的雲端聯網指南](#)
- [AWS re:Invent 2022 – 深入了解 AWS 聯網基礎設施](#)
- [AWS re:Invent 2019 – 連接 AWS 和混合 AWS 網路架構的連線能力](#)
- [AWS re:Invent 2018 – 最佳化 Amazon EC2 執行個體的網路效能](#)
- [AWS Global Accelerator](#)

### 相關範例：

- [AWS Transit Gateway 和可擴展的安全解決方案](#)
- [AWS 聯網研討會](#)
- [觀察和診斷網路](#)
- [尋找並解決在 AWS 上的網路設定錯誤](#)

## PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN

需要混合式連線來連接內部部署和雲端資源時，請佈建足夠的頻寬來滿足您的效能需求。預估混合工作負載的頻寬和延遲需求。您的規模調整需求取決於這些數字。

常見的反模式：

- 您只針對網路加密需求評估 VPN 解決方案。
- 您未評估備份或備援連線選項。
- 您無法識別所有工作負載需求 (加密、通訊協定、頻寬和流量需求)。

建立此最佳實務的優勢：選取和設定適當的連線解決方案將提高工作負載的可靠性，並充分利用效能。藉由識別工作負載需求、提前規劃和評估混合解決方案，就能盡可能減少昂貴的實體網路變更和營運負擔，同時實現更高的價值。

未建立此最佳實務時的曝險等級：高

### 實作指引

根據您的頻寬要求開發混合式聯網架構。[AWS Direct Connect](#) 可讓您將內部部署網路與 AWS 進行私密連線。需要高頻寬且低延遲，同時可達到一致效能時，適合這個選項。VPN 連線會建立透過網際網路的安全連線。使用時機包括：只需要臨時連線、須考量成本因素，或是在使用 AWS Direct Connect 的情況下，等待建立彈性實體網路連線時做為緊急應變措施。

如果您的頻寬需求很高，可以考慮多個 AWS Direct Connect 或 VPN 服務。流量可在服務之間達到負載平衡，雖然因為延遲和頻寬的差異，我們不建議在 AWS Direct Connect 和 VPN 之間達到負載平衡。

### 實作步驟

1. 預估現有應用程式的頻寬和延遲需求。
  - a. 針對移至 AWS 的現有工作負載，利用來自您的內部網路監控系統的資料。
  - b. 針對您沒有監控資料的新或現有工作負載，請諮詢產品擁有者以決定適當的效能指標，並且提供良好的使用者體驗。
2. 選取專用連線或 VPN 做為您的連線選項。根據所有工作負載要求 (加密、頻寬和流量需求)，您可以選擇 AWS Direct Connect 或 [AWS VPN](#) (或兩者)。下圖可協助您選擇適當的連線類型。
  - a. [AWS Direct Connect](#) 使用專用連線或託管連線，為 AWS 環境提供速度從 50 Mbps 到 100 Gbps 的專用連線。這可為您提供受管和受控的延遲以及佈建頻寬，因此您的工作負載可以有效率地連

線到其他環境。使用 AWS Direct Connect 合作夥伴，您就可以擁有來自多個環境的端對端連線能力，提供擴充的網路和一致的效能。AWS 提供使用原生 100 Gbps、連結彙總群組 (LAG) 或 BGP 等價多路徑 (ECMP) 的擴展直接連線連線頻寬。

- b. [AWS Site-to-Site VPN](#) 提供受管 VPN 服務，支援網際網路通訊協定安全性 (IPsec)。建立 VPN 連線時，每個 VPN 連線都包含兩個通道以獲得高可用性。
3. 依照 AWS 文件中所述，選擇適當的連線選項：
    - a. 如果您決定使用 AWS Direct Connect，請為您的連線選取適當的頻寬。
    - b. 如果您跨多個位置使用 AWS Site-to-Site VPN 連線至 AWS 區域，請使用 [加速 Site-to-Site VPN 連線](#) 以創造改善網路效能的機會。
    - c. 如果您的網路設計包含 [透過 AWS Direct Connect 的 IPsec VPN 連線](#) 請考慮使用私有 IP VPN 來提高安全性並實現區隔。[AWS Site-to-Site 私有 IP VPN](#) 是以傳輸虛擬介面 (VIF) 為基礎進行部署。
    - d. [AWS Direct Connect SiteLink](#) 可讓您在位於全世界的資料中心之間建立低延遲的冗餘連線，方法是在 [AWS Direct Connect 位置之間利用最快速的路徑傳送資料](#)，並略過 AWS 區域。
  4. 在部署到生產環境之前，先驗證您的連線設定。進行安全性和效能測試，確保其符合您的頻寬、可靠性、延遲和合規需求。
  5. 定期監控連線效能和使用情況，並視需要最佳化。

## 決定性效能流程圖

## 資源

### 相關文件：

- [搭配 AWS 的聯網產品](#)
- [AWS Transit Gateway](#)
- [VPC 端點](#)
- [建置可擴展且安全的多 VPC AWS 網路基礎設施](#)
- [用戶端 VPN](#)

### 相關影片：

- [AWS re:Invent 2023 – 使用 AWS 建立混合網路連線能力](#)
- [AWS re:Invent 2023 – 連接 AWS 的安全遠端連線能力](#)
- [AWS re:Invent 2022 – 最佳化 Amazon CloudFront 效能](#)
- [AWS re:Invent 2019 – 連接 AWS 和混合 AWS 網路架構的連線能力](#)
- [AWS re:Invent 2020 – AWS Transit Gateway Connect](#)

相關範例：

- [AWS Transit Gateway 和可擴展的安全解決方案](#)
- [AWS 聯網研討會](#)

## PERF04-BP04 使用負載平衡將流量分配到多個資源

在多個資源或服務之間分配流量，以讓您的工作負載能夠利用雲端提供的彈性。您也可以使用負載平衡來卸載加密終止，以提升效能、可靠性，以及有效管理和路由流量。

常見的反模式：

- 您在選擇負載平衡器類型時不考慮工作負載要求。
- 您未利用負載平衡器功能來進行效能最佳化。
- 工作負載在不使用負載平衡器的情況下，直接公開到網際網路。
- 您可以透過現有的負載平衡器路由所有網際網路流量。
- 您可以使用一般 TCP 負載平衡，並讓每個運算節點處理 SSL 加密。

建立此最佳實務的優勢：負載平衡器會處理單一可用區域中或跨多個可用區域的應用程式流量之各式負載，並實現高可用性、自動擴展及更充分利用您的工作負載。

未建立此最佳實務時的風險暴露等級：高

### 實作指引

負載平衡器會做為您的工作負載的進入點，從該處將您的流量分散到後端目標，例如運算執行個體或容器，以提高使用率。

選擇正確的負載平衡器類型是最佳化架構的第一步。從列出您的工作負載特性開始，例如通訊協定 (例如 TCP、HTTP、TLS 或 WebSockets)、目標類型 (例如執行個體、容器或無伺服器)、應用程式要求



(例如長時間執行連線、使用者身分驗證或黏性) 和置放 (例如 Region、Local Zone、Outpost 或區域隔離)。

AWS 為您的應用程式提供了多種模型來使用負載平衡。[Application Load Balancer](#) 最適合 HTTP 和 HTTPS 流量的負載平衡，並提供了針對現代應用程式架構 (包括微型服務和容器) 交付的進階請求路由。

[Network Load Balancer](#) 最適合需要極高效能的 TCP 流量的負載平衡。它能夠每秒處理數百萬個請求，同時保持超低延遲性，並且還進行優化，可處理突發的和不穩定的流量模式。

[Elastic Load Balancing](#) 提供整合的憑證管理和 SSL/TLS 解密，讓您能夠靈活地集中管理負載平衡器的 SSL 設定，並從工作負載中卸載 CPU 密集型工作。

選擇正確的負載平衡器之後，您可以開始利用其功能來減少後端為流量提供服務所需投入的工作量。

例如，同時使用 Application Load Balancer (ALB) 和 Network Load Balancer (NLB)，您可以執行 SSL/TLS 加密卸載，這是避免您的目標完成 CPU 密集型 TLS 交握，並且改善憑證管理的機會。

在您的負載平衡器中設定 SSL/TLS 卸載時，它會負責將往返用戶端的流量進行加密，同時將未加密的流量交付給您的後端，釋放您的後端資源並且改善用戶端的回應時間。

Application Load Balancer 也可以為 HTTP/2 流量提供服務，不需要在您的目標上支援它。這個簡單的決策可以改善您的應用程式回應時間，因為 HTTP/2 更有效率地使用 TCP 連線。

定義架構時，應考慮您的工作負載延遲要求。例如，如果您有對延遲敏感的應用程式，您可能會決定使用 Network Load Balancer，以獲得極低的延遲。另外，您可能會決定藉由利用 [AWS Local Zones](#) 中的 Application Load Balancer 或甚至是 [AWS Outposts](#)，讓工作負載更靠近您的客戶。

對延遲敏感的工作負載的另一個考慮是跨區域負載平衡。使用跨區域負載平衡，每個負載平衡器節點會將已註冊目標之間的流量分散到所有允許的可用區域中。

使用與您的負載平衡器整合的 Auto Scaling。效能效率系統的其中一個關鍵層面與適當調整後端資源的規模有關。若要完成此操作，您可以利用後端目標資源的負載平衡器整合。使用與 Auto Scaling 群組整合的負載平衡器，目標會視需要從負載平衡器新增或移除，以因應傳入流量。負載平衡器也可以針對容器化工作負載與 [Amazon ECS](#) 和 [Amazon EKS](#) 整合。

- [Amazon ECS - 服務負載平衡](#)
- [Amazon EKS 上的應用程式負載平衡](#)
- [Amazon EKS 上的網路負載平衡](#)

## 實作步驟

- 定義您的負載平衡需求，包括流量、可用性和應用程式可擴展性。
- 為您的應用程式選擇正確的負載平衡器類型。
  - 針對 HTTP/HTTPS 工作負載使用 Application Load Balancer。
  - 針對在 TCP 或 UDP 上執行的非 HTTP 工作負載使用 Network Load Balancer。
  - 如果您想要利用兩個產品的功能，使用兩者個組合 ([ALB 做為 NLB 的目標](#))。例如，如果您想要搭配使用 NLB 的靜態 IP 與來自 ALB 的 HTTP 標題型路由，或者如果您想要將您的 HTTP 工作負載公開到 [AWS PrivateLink](#)。
  - 如需負載平衡器的完整比較，請參閱 [ELB 產品比較](#)。
- 盡可能使用 SSL/TLS 卸載。
  - 將 HTTPS/TLS 接聽程式設定為讓 [Application Load Balancer](#) 和 [Network Load Balancer](#) 都與 [AWS Certificate Manager](#) 整合。
  - 請注意，基於合規理由，某些工作負載可能需要端對端加密。在此情況下，必須允許在目標進行加密。
  - 如需安全最佳實務，請參閱 [SEC09-BP02 強制執行傳輸中加密](#)。
- 選取正確的路由演算法 (僅 ALB)。
  - 路由演算法可以造成您的後端目標的妥善使用程度和它們影響效能程度的差異。例如，ALB 提供 [兩個路由演算法的選項](#)：
    - 最低未解決請求：針對當您的應用程式的請求因複雜性而異或您的目標因處理功能而異的情況時，用來讓負載更佳地分散到您的後端目標。
    - 輪詢均衡：當請求和目標類似，或是如果您需要在目標之間平均分散請求時使用。
- 考慮跨區域或區域隔離。
  - 針對延遲改善和區域失敗網域使用跨區域關閉 (區域隔離)。在 NLB 中預設為關閉，[在 ALB 中您可以依據各個目標群組加以關閉](#)。
  - 使用跨區域開啟來增加可用性和彈性。根據預設，ALB 的跨區域為開啟，[在 NLB 中您可以依據各個目標群組加以開啟](#)。
- 為您的 HTTP 工作負載開啟 HTTP keep-alives (僅 ALB)。使用這項功能，負載平衡器可以重複使用後端連線，直到 keep-alive 逾時到期，改善您的 HTTP 請求和回應時間，同時減少您的後端目標上的資源使用率。如需如何針對 Apache 和 Nginx 執行此操作的詳細資訊，請參閱 [使用 Apache 或 NGINX 做為 ELB 的後端伺服器的最佳設定是什麼？](#)
- 開啟負載平衡器的監控功能。
  - 為您的 [Application Load Balancer](#) 和 [Network Load Balancer](#) 開啟存取日誌。

- ALB 要考慮的主要欄位是 `request_processing_time`、`request_processing_time` 和 `response_processing_time`。
- NLB 要考慮的主要欄位是 `connection_time` 和 `tls_handshake_time`。
- 請準備好在您需要日誌時進行查詢。您可以使用 Amazon Athena 查詢 [ALB 日誌](#) 和 [NLB 日誌](#)。
- 建立效能相關指標的警示，例如 [ALB 的 TargetResponseTime](#)。

## 資源

相關文件：

- [ELB 產品比較](#)
- [AWS 全球基礎設施](#)
- [使用可用區域親和性改善效能並且降低成本](#)
- [使用 Amazon Athena 逐步執行日誌分析](#)
- [查詢 Application Load Balancer 日誌](#)
- [監控您的 Application Load Balancers](#)
- [監控您的 Network Load Balancer](#)
- [使用 Elastic Load Balancing 在您的 Auto Scaling 群組中的執行個體之間分散流量](#)

相關影片：

- [AWS re:Invent 2023：聯網對您的應用程式有何助益？](#)
- [AWS re:Inforce 2022：如何使用 Elastic Load Balancing 大規模增強您的安全態勢](#)
- [AWS re:Invent 2018：Elastic Load Balancing：深入探討和最佳實務](#)
- [AWS re:Invent 2021 - 如何為您的 AWS 工作負載選擇正確的負載平衡器](#)
- [AWS re:Invent 2019：針對不同工作負載充分發揮 Elastic Load Balancing](#)

相關範例：

- [Gateway Load Balancer](#)
- [使用 Amazon Athena 進行日誌分析的 CDK 和 AWS CloudFormation 範例](#)

## PERF04-BP05 選擇網路通訊協定以提高效能

根據對工作負載效能的影響，做出系統和網路間通訊協定的決策。

實現輸送量的延遲和頻寬之間存在關係。如果您的檔案傳輸使用傳輸控制協定 (TCP)，較高的延遲很可能會降低整體輸送量。有一些方法可以使用 TCP 調校和最佳化的傳輸通訊協定來解決這個問題，但有一個解決方案是透過使用者資料包協定 (UDP)。

常見的反模式：

- 無論效能需求為何，您都可以將 TCP 用於所有工作負載。

建立此最佳實務的優勢：確認針對使用者與工作負載元件之間的通訊使用適當的通訊協定，可協助改善您的應用程式的整體使用者體驗。例如，無連線 UDP 雖然達到高速，但卻失去重新傳輸能力或高可靠性。TCP 是功能完整的通訊協定，但需要更大的額外負荷來處理封包。

未建立此最佳實務時的曝險等級：中

### 實作指引

如果您能夠為應用程式選擇不同的通訊協定，而且您有這方面的專業知識，請使用不同的通訊協定來最佳化您的應用程式和最終使用者體驗。請注意，這種方法伴隨著相當高的難度，只有在您已先利用其他方式最佳化應用程式之後，才應嘗試此方法。

改善您的工作負載效能的主要考慮是了解延遲和輸送量要求，然後選擇可最佳化效能的網路通訊協定。

#### 考慮使用 TCP 的時機

TCP 提供可靠的資料交付，並且可用於可靠性和保證資料交付很重要的工作負載元件間通訊。許多 Web 式應用程式仰賴 TCP 型通訊協定 (例如 HTTP 和 HTTPS) 開啟 TCP 通訊端，以在應用程式元件之間進行通訊。電子郵件和檔案資料傳輸是常見的應用程式，同樣會使用 TCP，因為 TCP 是應用程式元件之間簡單又可靠的傳輸機制。使用 TLS 與 TCP 會對通訊增加一些負擔，導致提高延遲和降低輸送量，但也會帶來安全上的優勢。負擔主要來自交握處理的增加負擔，需要數個往返才能完成。一旦交握完成，加密和解密資料的負擔相對小。

#### 考慮使用 UDP 的時機

UDP 是無連線導向的通訊協定，因此適用於需要快速、有效傳輸的應用程式，例如日誌、監控和 VoIP 資料。此外，如果您有會回應來自大量用戶端之小型查詢的工作負載元件，請考慮使用 UDP，以確保

最佳工作負載效能。資料包傳輸層安全性 (DTLS) 是等同於 Transport Layer Security (TLS) 的 UDP。使用 DTLS 與 UDP 時，負擔是來自加密和解密資料，因為交握處理已簡化。DTLS 也會對 UDP 封包增加少量負擔，因為 DTLS 包含額外欄位，可指出安全參數及偵測竄改。

### 考慮使用 SRD 的時機

Scalable Reliable Datagram (SRD) 是針對高輸送量工作負載最佳化的網路傳輸通訊協定，能夠在多個路徑之間負載平衡流量，並且快速從封包捨棄或連結失敗復原。因此 SRD 最適合用於需要運算節點之間高輸送量和低延遲通訊的高效能運算 (HPC) 工作負載。這可能包含平行處理任務，例如牽涉到在節點之間大量資料傳輸的模擬、建模和資料分析。

## 實作步驟

1. 使用 [AWS Global Accelerator](#) 和 [AWS Transfer Family](#) 服務來改善您的線上檔案傳輸應用程式的輸送量。AWS Global Accelerator 服務可協助您在用戶端裝置與 AWS 上工作負載之間實現較低的延遲。使用 AWS Transfer Family，您可以使用 TCP 型通訊協定，例如 Secure Shell File Transfer Protocol (SFTP) 和 File Transfer Protocol over SSL (FTPS)，安全地擴展和管理與 AWS 儲存服務之間的檔案傳輸。
2. 使用網路延遲來判斷 TCP 是否適合工作負載元件之間的通訊。如果您的用戶端應用程式與伺服器之間的網路延遲高，則 TCP 三向交握會耗費一些時間，因此會影響您的應用程式的回應能力。例如到第一個位元組的時間 (TTFB) 和往返時間 (RTT) 等指標可用來測量網路延遲。如果您的工作負載為使用者提供動態內容的服務，請考慮使用 [Amazon CloudFront](#)，這會建立與每個動態內容來源的持久性連線，移除會拖慢每個用戶端請求速度的連線設定時間。
3. 使用 TLS 與 TCP 或 UDP 會由於加密和解密的影響，導致對您的工作負載增加延遲和減少輸送量。針對此類工作負載，請考慮 [Elastic Load Balancing](#) 上的 SSL/TLS 卸載，藉由允許負載平衡器處理 SSL/TLS 加密和解密程序而不是讓後端執行個體進行處理，來改善工作負載效能。這可協助減少後端執行個體上的 CPU 使用率，可以改善效能和增加容量。
4. 使用 [Network Load Balancer \(NLB\)](#) 來部署依賴 UDP 通訊協定的服務，例如身分驗證和授權、記錄、DNS、IoT 和串流媒體，改善您的工作負載的效能和可靠性。NLB 會在多個目標之間分散傳入 UDP 流量，讓您水平地擴展工作負載、增加容量以及減少單一目標的負擔。
5. 針對您的高效能運算 (HPC) 工作負載，請考慮使用 [彈性網路介面卡 \(ENA\) 快速版](#) 功能，該功能使用 SRD 通訊協定來改善網路效能，方法是針對 EC2 執行個體之間的網路流量，提供較高的單一流量頻寬 (25Gbps) 和較低的尾端延遲 (99.9 百分位數)。
6. 使用 [Application Load Balancer \(ALB\)](#) 來路由和負載平衡工作負載元件之間的 gRPC (遠端程序呼叫) 流量或 gRPC 用戶端與服務之間的流量。gRPC 會使用 TCP 型 HTTP/2 通訊協定進行傳輸，並且提供如較輕網路足跡、壓縮、有效二進位序列化、支援多種語言和雙向串流的優點。

## 資源

相關文件：

- [如何將 UDP 流量路由到 Kubernetes](#)
- [Application Load Balancer](#)
- [Linux 上的 EC2 增強型聯網](#)
- [Windows 上的 EC2 增強型聯網](#)
- [EC2 置放群組](#)
- [在 Linux 執行個體上啟用搭配彈性網路轉接器 \(ENA\) 的增強型聯網](#)
- [Network Load Balancer](#)
- [搭配 AWS 的聯網產品](#)
- [轉換到 Amazon Route 53 中的 Latency-Based Routing](#)
- [VPC 端點](#)

相關影片：

- [AWS re:Invent 2022 – 擴展新一代 Amazon Elastic Compute Cloud 執行個體的網路效能](#)
- [AWS re:Invent 2022 – 應用程式聯網基礎](#)

相關範例：

- [AWS Transit Gateway 和可擴展的安全解決方案](#)
- [AWS 聯網研討會](#)

## PERF04-BP06 根據網路需求選擇工作負載的位置

評估資源置放的選項以減少網路延遲和提高輸送量，藉由減少頁面載入和資料傳輸時間來提供最佳的使用者體驗。

常見的反模式：

- 您可以將所有工作負載資源合併到單一地理位置。
- 您選擇的區域最接近您的位置，但不是最接近工作負載最終使用者。

建立此最佳實務的優勢：使用者體驗因使用者與您的應用程式之間的延遲而大受影響。透過使用適當的 AWS 區域 和 AWS 私有全球網路，就可以減少延遲並為遠程使用者提供更優質的體驗。

未建立此最佳實務時的風險暴露等級：中

## 實作指引

例如 Amazon EC2 執行個體的資源會放到 [AWS 區域](#)、[AWS Local Zones](#)、[AWS Outposts](#) 或 [AWS Wavelength](#) 區域內的可用區域。此位置的選擇會影響來自特定使用者位置的網路延遲和輸送量。[Amazon CloudFront](#) 和 [AWS Global Accelerator](#) 等的邊緣服務也可以用來改善網路效能，做法是在邊緣節點快取內容，或透過 AWS 全球網路為使用者提供工作負載的最佳路徑。

Amazon EC2 提供了適用於聯網的置放群組。置放群組是執行個體的邏輯分組，可降低延遲。使用搭配支援的執行個體類型以及彈性網路轉接器 (ENA) 的置放群組，可讓工作負載加入低延遲、低抖動的 25 Gbps 網路。建議將置放群組用於受益於低網路延遲、高網路輸送量或兩者兼而有之的工作負載。

對延遲敏感的服務是在邊緣節點使用 AWS 全球網路交付，例如 [Amazon CloudFront](#)。這些節點通常可提供如內容交付網路 (CDN) 和網域名稱系統 (DNS) 之類的服務。透過將這些服務置於邊緣，工作負載就可以在低延遲的情況下回應內容或 DNS 解析的請求。這些服務還提供地理服務，例如內容的地理定位 (根據最終使用者的位置提供不同的內容)，或以延遲為基礎的路由，將最終使用者定向到最近區域的 (最小延遲)。

使用邊緣服務來減少延遲及啟用內容快取。為 DNS 和 HTTP/HTTPS 正確設定快取控制，以從這些方法中獲得最大的效益。

## 實作步驟

- 擷取與往返網路介面的 IP 流量有關的資訊。
  - [使用 VPC Flow Logs 記錄 IP 流量](#)
  - [在 AWS Global Accelerator 中保留用戶端 IP 位址的方式](#)
- 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](#)) 收集網路活動的相關資料。
  - 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：
  - 資料的所在位置：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。
  - 使用者的所在位置：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之類的限制，相關說明請見 [為工作負載選取區域時應考量的事項](#)。

- 使用 [AWS Local Zones](#) 執行影片轉譯等工作負載。Local Zones 可讓您因運算和儲存資源更接近最終使用者而獲益。
- [AWS Outposts](#) 適用於需要保持內部部署的工作負載，而您希望該工作負載能夠與 AWS 中的其他工作負載無縫執行。
- 像是高解析度即時影片串流、高保真度音訊和擴增實境或虛擬實境 (AR/VR) 等應用程式，需要 5G 裝置的極低延遲。針對此類應用程式，請考慮 [AWS Wavelength](#)。AWS Wavelength 會將 AWS 運算及儲存服務嵌入 5G 網路，為開發、部署和擴展極低延遲應用程式提供行動裝置邊緣運算基礎設施。
- 使用本機快取或 [AWS 快取解決方案](#) 取得常用資產，以提升效能、減少資料移動，以及降低環境影響。

| Service                              | When to use  |
|--------------------------------------|--|
| <a href="#">Amazon CloudFront</a>    | 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)。 |
| <a href="#">Amazon ElastiCache</a>   | 用來快取 Web 應用程式的內容。                                    |
| <a href="#">DynamoDB Accelerator</a> | 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，如下所示：

| Service                              | When to use                            |
|--------------------------------------|--|
| <a href="#">Lambda@edge</a>          | 用於在物件未經快取時起始的大量運算作業。                   |
| <a href="#">Amazon CloudFront 函數</a> | 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。 |
| <a href="#">AWS IoT Greengrass</a>   | 用來為連線的裝置執行本機運算、傳訊和資料快取。                |

- 某些應用程式需要藉由減少第一個位元組延遲和抖動並且增加輸送量，來獲得固定的進入點或較高的效能。這些應用程式可以從在邊緣節點提供靜態任播 IP 位址和 TCP 終止的網路服務獲益。[AWS Global Accelerator](#) 可以改善您的應用程式效能高達 60%，並且提供多區域架構的快速容錯移轉。AWS Global Accelerator 提供靜態任播 IP 地址，做為一或多個 AWS 區域中託管應用程式的



固定進入點。這些 IP 位址允許流量傳入盡可能靠近您的使用者的 AWS 全球網路。AWS Global Accelerator 會藉由建立用戶端與最接近用戶端之 AWS 邊緣節點之間的 TCP 連線，減少初始連線設定時間。請檢閱 AWS Global Accelerator 的使用情形，以改善您的 TCP/UDP 工作負載的效能，並且提供多區域架構的快速容錯移轉。

## 資源

相關的最佳實務：

- [COST07-BP02 根據成本實作區域](#)
- [COST08-BP03 實作可降低資料傳輸成本的服務](#)
- [REL10-BP01 將工作負載部署至多個位置](#)
- [REL10-BP02 為您的多位置部署選取適當位置](#)
- [SUS01-BP01 根據業務需求和永續性目標選擇區域](#)
- [SUS02-BP04 根據工作負載的聯網需求最佳化其地理位置](#)
- [SUS04-BP07 將跨網路的資料移動降到最少](#)

相關文件：

- [AWS 全球基礎設施](#)
- [AWS Local Zones 和 AWS Outposts，為您的邊緣工作負載選擇正確的技術](#)
- [置放群組](#)
- [AWS Local Zones](#)
- [AWS Outposts](#)
- [AWS Wavelength](#)
- [Amazon CloudFront](#)
- [AWS Global Accelerator](#)
- [AWS Direct Connect](#)
- [AWS Site-to-Site VPN](#)
- [Amazon Route 53](#)

相關影片：

- [AWS Local Zones 解說影片](#)
- [AWS Outposts：概觀和運作方式](#)
- [AWS re:Invent 2023 - 邊緣與內部部署工作負載的遷移策略](#)
- [AWS re:Invent 2021 - AWS Outposts：在內部部署環境帶來 AWS 體驗](#)
- [AWS re:Invent 2020：AWS Wavelength：在 5G 邊緣以極低延遲執行應用程式](#)
- [AWS re:Invent 2022 - AWS Local Zones：為分散的邊緣建置應用程式](#)
- [AWS re:Invent 2021 - 使用 Amazon CloudFront 建置低延遲網站](#)
- [AWS re:Invent 2022 - 使用 AWS Global Accelerator 改善效能與可用性](#)
- [AWS re:Invent 2022 - 使用 AWS 建置您的全球廣域網路](#)
- [AWS re:Invent 2020：使用 Amazon Route 53 進行全球流量管理](#)

相關範例：

- [AWS Global Accelerator 自訂路由研討會](#)
- [使用邊緣功能處理重新撰寫和重新導向](#)

## PERF04-BP07 根據指標最佳化網路組態

使用收集和分析的資料來做出有關最佳化網路組態的明智決策。

常見的反模式：

- 您假設所有效能相關問題都與應用程式有關。
- 您只能從靠近已部署工作負載的位置測試網路效能。
- 您針對所有網路服務使用預設組態。
- 您過度佈建網路資源來提供足夠的容量。

建立此最佳實務的優勢：收集您的 AWS 網路的必要指標和實作網路監控工具，可讓您了解網路效能和最佳化網路組態。

未建立此最佳實務時的曝險等級：低

## 實作指引

監控往返 VPC、子網路或網路界面的流量，對於了解如何利用 AWS 網路資源和如何最佳化網路組態而言非常重要。使用下列 AWS 聯網工具，即可進一步檢查流量用量、網路存取和日誌的相關資訊。

### 實作步驟

- 確定要收集的關鍵績效指標，例如延遲或封包遺失。AWS 提供了幾種工具，可幫助您收集這些指標。使用下列工具，即可進一步檢查流量用量、網路存取和日誌的相關資訊：

| AWS 工具  | 在何處使用   |
|---|---|
| <a href="#">Amazon VPC IP Address Manager</a> | 使用 IPAM 來規劃、追蹤和監控您的 AWS 和內部部署工作負載的 IP 地址。這是最佳化 IP 地址用量和分配的最佳實務。   |
| <a href="#">VPC Flow Logs</a>                 | 使用 VPC Flow Logs 來擷取有關往返您的 VPC 中網路介面流量的詳細資訊。使用 VPC Flow Logs 可診斷過於嚴格或寬鬆的安全群組規則，並且判斷往返網路介面的流量方向。                               |
| <a href="#">AWS Transit Gateway Flow Logs</a> | 使用 AWS Transit Gateway Flow Logs 擷取往返傳輸閘道的 IP 流量相關資訊。   |
| <a href="#">DNS 查詢記錄</a>                      | 記錄 Route 53 所收到的公有或私有 DNS 查詢的相關資訊。使用 DNS 日誌，您可以藉由了解請求的網域或子網域或是回應 DNS 查詢的 Route 53 邊緣節點，最佳化 DNS 組態。                            |
| <a href="#">Reachability Analyzer</a>         | Reachability Analyzer 可協助您分析和偵錯網路連線能力。Reachability Analyzer 是組態分析工具，可讓您執行您的 VPC 中來源資源與目的地資源之間的連線能力測試。此工具可協助您確認您的組態符合您預期的連線能力。 |
| <a href="#">Network Access Analyzer</a>       | Network Access Analyzer 可協助您了解資源的網路存取。您可以使用 Network Access Analyzer 來指定您的網路存取要求，並識別未符合您的指定要求的潛在網路路徑。藉由最佳                      |

| AWS 工具                                | 在何處使用  |
|---------------------------------------|--|
| <a href="#">Amazon CloudWatch</a>     | <p>化您的對應網路組態，您可以了解及確認網路的狀態，並且示範 AWS 上的網路是否符合您的合規要求。</p> <p>使用 <a href="#">Amazon CloudWatch</a> 並且為網路選項開啟適當的指標。請確定為您的工作負載選擇正確的網路指標。例如，您可以開啟 VPC 網路地址用量、VPC NAT 閘道、AWS Transit Gateway、VPN 通道、AWS Network Firewall、Elastic Load Balancing 和 AWS Direct Connect 等指標。持續監控指標是觀察和了解您的網路狀態和用量的良好實務，可協助您根據您的觀察來最佳化網路組態。</p> |
| <a href="#">AWS Network Manager</a>   | <p>您可以使用 AWS Network Manager 監控 <a href="#">AWS 全球網絡</a> 的即時和歷史效能，以因應操作和規劃用途。Network Manager 提供了 AWS 區域與可用區域之間及每個可用區域內的彙總網路延遲，讓您更了解應用程式效能與基礎 AWS 網路效能之間的關係。</p>  |
| <a href="#">Amazon CloudWatch RUM</a> | <p>使用 Amazon CloudWatch RUM 收集指標來為您提供洞見，以協助您識別、了解和改善使用者體驗。</p>   |

- 使用 VPC 和 AWS Transit Gateway Flow Logs 找出最活躍的發言者和應用程式流量模式。
- 評估並最佳化目前的網路架構，包括 VPC、子網路和路由。例如，您可以評估不同的 VPC 對等互連或 AWS Transit Gateway 如何協助您改善架構中的網路。
- 評估網路中的路由路徑，以確認目的地之間一律使用最短路徑。Network Access Analyzer 可幫助您實現這點。

## 資源

相關文件：

- [公有 DNS 查詢記錄](#)

- [什麼是 IPAM ?](#)
- [什麼是 Reachability Analyzer ?](#)
- [什麼是 Network Access Analyzer ?](#)
- [您的 VPC 的 CloudWatch 指標](#)
- [使用 Apache Parquet 格式的 VPC Flow Logs 針對網路分析最佳化效能和降低成本](#)
- [使用 Amazon CloudWatch 指標監控您的全球和核心網路](#)
- [持續監控網路流量和資源](#)

#### 相關影片：

- [AWS re:Invent 2023 – 開發人員的雲端聯網指南](#)
- [AWS re:Invent 2023 – 準備好踏出下一步了嗎？設計可促進成長與靈活性的網路](#)
- [AWS re:Invent 2023 – 進階 VPC 設計及新功能](#)
- [AWS re:Invent 2022 – 深入了解 AWS 聯網基礎設施](#)
- [AWS re:Invent 2020 – 搭配 AWS Well-Architected Framework 的聯網最佳實務和秘訣](#)
- [AWS re:Invent 2020 – 監控網路流量並對其進行疑難排解](#)

#### 相關範例：

- [AWS 聯網研討會](#)
- [AWS 網路監控](#)
- [觀察和診斷在 AWS 上的網路](#)
- [尋找並解決在 AWS 上的網路設定錯誤](#)

# 程序和文化

在架構工作負載時，您可以採取一些原則和實務，來更有效率地執行高效能雲端工作負載。這個重點領域提供了各種最佳實務，以培養高效能雲端工作負載的文化。

打造這類文化時，請考慮以下重要原則：

- **基礎設施即程式碼：** 使用 AWS CloudFormation 範本等方法將您的基礎設施定義為程式碼。使用範本可讓您將基礎設施與應用程式程式碼和組態一起置於原始檔控制中。這可讓您在基礎設施中套用開發軟體時所使用的相同做法，進而快速進行迭代。
- **部署管道：** 使用持續整合/持續部署 (CI/CD) 管道 (例如，原始程式碼儲存庫、建置系統、部署和測試自動化) 來部署您的基礎設施。這樣您就可以在反覆執行的過程中，採用可重複、一致且低成本的方式進行部署。
- **定義明確的指標：** 設定並監控指標以擷取關鍵績效指標 (KPI)。我們建議您同時使用技術和業務指標。對於網站或行動應用程式，關鍵指標是擷取第一個位元組或轉譯的時間。其他一般適用的指標包括執行緒計數、垃圾回收率和等待狀態。業務指標 (例如每個請求的彙總累計成本) 會提示您降低成本的方法。仔細考慮您計劃如何解釋指標。例如，您可以選擇最大值或第 99 個百分位數，而非平均值。
- **自動執行效能測試：** 在部署程序中，成功通過快速執行測試之後，就會自動開始進行效能測試。自動化應建立一個新的環境，設定如測試資料之類的初始條件，然後執行一系列基準測試和負載測試。這些測試的結果應與組建版本綁定，方便您追蹤長時間的效能變化。對於長期執行的測試，您可以讓管道的這個部分與組建版本的其餘部分不同步。或者，您可以使用 Amazon EC2 Spot 執行個體在夜間執行效能測試。
- **負載產生：** 您應建立一系列的測試指令碼來複寫綜合性或預錄的使用者旅程。這些指令碼應該是冪等及非耦合的形式，而且您可能需要納入「預熱型」指令碼才能產生有效的結果。您的測試指令碼應盡可能地複寫生產環境中的使用行為。您可以使用軟體或軟體即服務 (SaaS) 解決方案來產生負載。考慮使用 [AWS Marketplace](#) 解決方案和 [Spot 執行個體](#) — 它們會是產生負載的經濟實惠方式。
- **效能可見度：** 關鍵指標應對您的團隊可見，尤其是針對每個組建版本的指標。這可讓您查看隨時間變化出現的任何顯著的正面或負面趨勢。您也應顯示錯誤或例外狀況數量的指標，以確保您測試的是可運作的系統。
- **視覺化：** 使用視覺化技術可以清楚指出何處出現效能問題、熱點、等待狀態或較低的利用率。在架構圖上重疊效能指標 — 呼叫圖表或程式碼有助於快速識別問題。
- **定期審查程序：** 架構效能不佳通常是效能審查程序不存在或中斷的結果。如果您的架構效能不佳，則實作效能審查程序可讓您不斷反覆進行改善。
- **持續最佳化：** 培養文化以持續優化雲端工作負載效能達成效率。

## 最佳實務

- [PERF05-BP01 建立關鍵績效指標 \(KPI\) 以衡量工作負載健康狀態和效能](#)
- [PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域](#)
- [PERF05-BP03 定義提高工作負載效能的程序](#)
- [PERF05-BP04 對工作負載執行負載測試](#)
- [PERF05-BP05 使用自動化主動修復效能相關問題](#)
- [PERF05-BP06 即時更新工作負載和服務的狀態](#)
- [PERF05-BP07 定期檢閱指標](#)

## PERF05-BP01 建立關鍵績效指標 (KPI) 以衡量工作負載健康狀態和效能

找出定量和定性衡量工作負載效能的 KPI。KPI 可協助您衡量與業務目標相關之工作負載的健康狀態和效能。

常見的反模式：

- 您只監控系統層級指標，以洞悉工作負載，但不了解對這些指標的業務影響。
- 您假設 KPI 已發佈，並作為標準指標資料分享。
- 您未定義可量化且可衡量的 KPI。
- 您沒有將 KPI 與業務目標或策略保持一致。

建立此最佳實務的好處：找出代表工作負載健康狀態和效能的特定 KPI，有助團隊以一致的標準排定優先事項並定義成功的業務成果。與所有部門分享這些指標，可提供對閾值、期望和業務影響的可見性和一致性。

未建立此最佳實務時的風險暴露等級：高

## 實作指引

KPI 可讓業務和工程團隊以一致的標準衡量目標和策略，以及將這些因素結合以產生商業成果的方式。例如，網站工作負載可能使用頁面載入時間，作為整體效能的指示。此指標將是衡量最終使用者體驗的多個資料點之一。除了找出頁面載入時間閾值外，您還應該記錄未符合理想效能時預期的成果或業務風險。較長的頁面載入時間會直接影響最終使用者，降低他們的使用者體驗評級，並可能導致客戶流失。

當您定義 KPI 閾值時，請同時結合業界基準和最終使用者期望。例如，如果目前業界基準是網頁在兩秒內載入，但最終使用者期望網頁在一秒內載入，則您在建立 KPI 時應將這兩個資料點都列入考慮。

團隊必須使用即時精密資料和歷史資料作為參考，來評估工作負載 KPI，並建立儀表板，針對 KPI 資料執行指標數學，以衍生營運和使用率洞察。KPI 應該加以記錄，並包含支援業務目標和策略的 KPI 和閾值，以及對應到受監控的指標。當業務目標、策略或最終使用者要求變更時，應該重新檢視 KPI。

## 實作步驟

- 識別利害關係人：識別並記錄關鍵業務利害關係人，包括開發和營運團隊。
- 定義目標：與利害關係人合作，以定義和記錄工作負載的目標。思考工作負載的關鍵績效，例如輸送量、回應時間和成本，以及業務目標 (如使用者滿意度)。
- 檢閱業界最佳實務：檢閱業界最佳實務以識別符合您工作負載目標的相關 KPI。
- 識別指標：識別與工作負載目標相符的指標，並可協助衡量績效和業務目標的指標。根據這些指標建立 KPI。範例指標是指像平均回應時間或並行使用者人數之類的量測值。
- 定義並記錄 KPI：使用業界最佳實務和工作負載目標，為您的工作負載 KPI 設立目標。使用此資訊，來設定嚴重性或警示層級的 KPI 閾值。找出並記錄未符合 KPI 時的風險和影響。
- 實作監控：使用 [Amazon CloudWatch](#) 或 [AWS Config](#) 之類的監控工具來收集指標並衡量 KPI。
- 以視覺化方式溝通關鍵績效指標：使用 [Amazon QuickSight](#) 等儀表板工具以視覺化方式呈現，並與利害關係人溝通 KPI 議題。
- 分析和最佳化：定期審查和分析 KPI，以確定需要改善的工作負載區域。與利害關係人合作執行這些改進措施。
- 重新檢視和調整：定期檢閱指標和 KPI 以評估其有效性，尤其是當業務目標或工作負載績效發生變化時。

## 資源

相關文件：

- [CloudWatch 文件](#)
- [監控、記錄和效能 AWS Partner](#)
- [AWS 可觀測性工具](#)
- [關鍵績效指標 \(KPI\) 對大規模雲端遷移的重要性](#)
- [如何使用 KPI 儀表板追蹤成本最佳化 KPI](#)



- [X-Ray 文件](#)
- [使用 Amazon CloudWatch 儀表板](#)
- [Amazon QuickSight KPI](#)

相關影片：

- [AWS re:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度](#)
- [AWSre:Invent 2023 - 使用 AWS Health 以大規模管理資源生命週期事件](#)
- [AWS re:Invent 2023 - Pinterest 的效能與效率：最佳化最新執行個體](#)
- [AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果](#)
- [AWS re:Invent 2023 - 打造有效的可觀測性策略](#)
- [AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控](#)
- [AWS re:Invent 2023 - 為前一千萬名使用者擴展 AWS](#)
- [AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能](#)
- [為您的業務建立有效的指標策略 | AWS 活動](#)

相關範例：

- [使用 Amazon QuickSight 建立儀表板](#)

## PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域

了解並找出提高工作負載效能將對效率或客戶體驗產生正面影響的地方。例如，具有大量客戶互動的網站可受益於邊緣服務的使用，因為這樣可以將內容交付移至更接近客戶的地方。

常見的反模式：

- 您假設標準運算指標 (例如 CPU 使用率或記憶體壓力) 足以找出效能問題。
- 您只會使用所選監控軟體記錄的預設指標。
- 您只會在有問題時審查指標。

建立此最佳實務的好處：了解關鍵效能領域可協助工作負載擁有者監控 KPI 並優先處理具有高影響力的待改善之處。

未建立此最佳實務時的風險暴露等級：高

## 實作指引

設置端到端追蹤，以找出流量模式、延遲和關鍵的效能區域。監控資料存取模式是否有緩慢查詢或分段和分區不佳的資料。使用負載測試或監控來找出工作負載受限面向。

透過了解架構、流量模式和資料存取模式，來提高效能效率，並確定延遲和處理時間。找出隨著工作負載的成長，可能會影響客戶體驗的潛在瓶頸。當您已調查這些面向時，請審視自己可以部署哪個解決方案，來消除這些效能疑慮。

### 實作步驟

- 設置端到端監控，來擷取所有工作負載組成部分和指標。以下是 AWS 上的監控解決方案的範例。

| Service   | Where to use   |
|---|--|
| <a href="#">Amazon CloudWatch 實際使用者監控 (RUM)</a>         | To capture application performance metrics from real user client-side and frontend sessions.   |
| <a href="#">AWS X-Ray</a>                               | To trace traffic through the application layers and identify latency between components and dependencies. Use X-Ray service maps to see relationships and latency between workload components. |
| <a href="#">Amazon Relational Database Service 績效詳情</a> | To view database performance metrics and identify performance improvements.  |
| <a href="#">Amazon RDS 增強型監控</a>                        | To view database OS performance metrics.   |
| <a href="#">Amazon DevOps Guru</a>                      | To detect abnormal operating patterns so you can identify operational issues before they impact your customers.  |

- 執行測試，來產生指標、確定流量模式、瓶頸和關鍵效能區域。以下是如何執行測試的幾個範例：
  - 設定 [CloudWatch Synthetic Canaries](#)，以程式設計方式使用 Linux Cron 任務或評分運算式，模擬以瀏覽器為基礎的使用者活動，以產生長期一致的指標。

- 使用 [AWS 分散式負載測試](#) 解決方案，來產生尖峰流量或以預期的成長速率測試工作負載。
- 評估指標和遙測，來找出關鍵的效能領域。與團隊檢視這些領域，討論監控和解決方案，來避免瓶頸。
- 進行效能改善的實驗，並透過資料來衡量這些變更。舉例來說，您可以使用 [CloudWatch Evidently](#)，測試新的改善項目和對工作負載的效能影響。

## 資源

### 相關文件：

- [re:Invent 2023 中 AWS 可觀測性的最新消息](#)
- [Amazon 建置者資料中心](#)
- [X-Ray 文件](#)
- [Amazon CloudWatch RUM](#)
- [Amazon DevOps Guru](#)

### 相關影片：

- [AWS re:Invent 2023 - \[LAUNCH\] 現代工作負載的應用程式監控](#)
- [AWS re:Invent 2023 - 實作應用程式可觀測性](#)
- [AWS re:Invent 2023 - 打造有效的可觀測性策略](#)
- [AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控](#)
- [AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果](#)
- [AWS re:Invent 2022 - Amazon 建置者資料中心：25 年的 Amazon 卓越營運](#)
- [AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能](#)
- [使用 Amazon CloudWatch Synthetics 進行應用程式的視覺監控](#)

### 相關範例：

- [使用 Amazon CloudWatch Synthetics 測量頁面載入時間](#)
- [Amazon CloudWatch RUM Web 用戶端](#)
- [X-Ray SDK for Python](#)

- [AWS 上的分散式負載測試](#)

## PERF05-BP03 定義提高工作負載效能的程序

定義一個程序，以在新的服務、設計模式、資源類型和組態可用時對其進行評估。例如，對新的執行個體方案執行現有的效能測試，以判斷其是否可能改善工作負載。

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。
- 您會隨時間導入架構變更，而且無須指標佐證。

建立此最佳實務的好處：定義進行架構變更的程序後，即可啟用收集的資料，以隨著時間影響工作負載設計。

未建立此最佳實務時的風險暴露等級：中

### 實作指引

工作負載的效能有一些關鍵限制。記錄這些內容，以便您知道哪種創新可以改善工作負載的效能。當新服務或技術可用時，請使用此資訊來找出緩解限制或瓶頸的方法。

找出工作負載的關鍵效能限制。記錄工作負載的效能限制，讓您知道哪些類型的創新可能會改善工作負載的效能。

### 實作步驟

- 識別 KPI：按照[PERF05-BP01 建立關鍵績效指標 \(KPI\) 以衡量工作負載健康狀態和效能](#)中所述，找出工作負載效能 KPI，以設立工作負載基準。
- 實作監控：使用[AWS 可觀測性工具](#)收集效能指標並衡量 KPI。
- 執行分析：執行深入分析，以找出工作負載中效能不佳的區域 (例如組態和應用程式程式碼)，如[PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域](#)所述。使用分析和效能工具，找出效能改善策略。
- 驗證改善項目：使用沙盒或試生產環境來驗證改善策略的有效性。
- 實作變更：實作生產中的變更，並持續監控工作負載的效能。記錄改善項目並與利害關係人溝通這些變更。
- 重新檢視和調整：定期檢閱您的績效改進流程，以確定需要增強的區域。

## 資源

相關文件：

- [AWS 部落格](#)
- [AWS 最新消息](#)
- [AWS Skill Builder](#)

相關影片：

- [AWS re:Invent 2022 - 提供永續且高效能的架構](#)
- [AWS re:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度](#)
- [AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果](#)
- [AWS re:Invent 2022 - 利用最佳實務指引最佳化您的 AWS 工作負載](#)

相關範例：

- [AWS Github](#)

## PERF05-BP04 對工作負載執行負載測試

對工作負載執行負載測試，以確認可以處理生產負載並找出任何效能瓶頸。

常見的反模式：

- 您載入測試工作負載的個別部分，而非整個工作負載。
- 您在與生產環境不同的基礎設施上載入測試。
- 您只對預期的 (而非超標) 負載進行負載測試，以協助預測未來可能發生問題的位置。
- 您可以執行負載測試，不需參閱[Amazon EC2測試政策](#)，也不必提交模擬事件提交表單。這會導致測試無法執行，因為它看起來像拒絕服務事件。

建立此最佳做法的好處：在負載測試下測量效能時，會顯示您將在負載增加到何種程度時受到影響。這可在變更影響工作負載之前，讓您先預測所需的變更。

未建立此最佳實務時的風險暴露等級：低

## 實作指引

雲端中的負載測試是在實際條件下，以預期的使用者負載來衡量雲端工作負載效能的程序。此程序包括佈建類似生產環境的雲端環境、使用負載測試工具產生負載，以及分析指標，以便評估您的工作負載處理實際負載的能力。必須使用生產資料的綜合或處理過的版本 (移除敏感或可識別身分的資訊) 執行負載測試。在交付管道中自動執行負載測試，並將結果與預先定義的 KPI 和閾值進行比較。此程序可幫助您繼續實現所需的效能。

### 實作步驟

- 定義測試目標：識別您要評估的工作負載效能，例如輸送量和回應時間。
- 選取測試工具：選擇並設定適合您工作負載的負載測試工具。
- 設定您的環境：根據您的生產環境設定測試環境。您可以使用 AWS 服務執行生產規模的環境，進而測試架構。
- 實作監控：使用監控工具，例如 Amazon CloudWatch 收集架構中資源之間的指標。您也可以收集和發佈自訂指標。
- 定義情境：定義負載測試方案和參數 (如測試持續時間和使用者數量)。
- 進行負載測試：大規模執行測試情境。利用 AWS 雲端 測試工作負載，以發現無法擴展的地方或是否以非線性方式擴展。例如，使用 Spot 執行個體以低成本產生負載，並在生產中遇到瓶頸之前發現瓶頸。
- 分析測試結果：分析結果以找出效能瓶頸和需要改善的區域。
- 記錄並共享調查結果：記錄和報告調查結果和建議。與利害關係人共享此資訊，協助他們對效能最佳化策略做出明智的決定。
- 不斷反覆執行：負載測試應以規律的節奏執行，尤其是在系統更改更新後。

### 資源

相關文件：

- [Amazon CloudWatch RUM](#)
- [Amazon CloudWatch Synthetics](#)
- [AWS 上的分散式負載測試](#)

相關影片：

- [AWS Summit ANZ 2023：透過 AWS 分散式負載測試讓您放心加快腳步](#)
- [AWS re:Invent 2022 - 為您的前一千萬名使用者擴展 AWS](#)
- [使用 AWS 解決方案解決問題：分散式負載測試](#)
  
- [AWS re:Invent 2021 - 透過最終使用者洞察與 Amazon CloudWatch RUM 優化應用程式](#)
- [Amazon CloudWatch Synthetics 的示範](#)

相關範例：

- [AWS 上的分散式負載測試](#)

## PERF05-BP05 使用自動化主動修復效能相關問題

使用關鍵績效指標 (KPI) 搭配監控和提醒系統，主動處理效能相關的問題。

常見的反模式：

- 您只讓操作人員有能力對工作負載進行操作變更。
- 您讓所有警示篩選到操作團隊，無須主動修復。

建立此最佳實務的好處：主動修復警示動作，可讓支援人員專注在無法自動採取行動的項目上。有助操作人員處理所有警示，不會不堪負荷，而能專注在重要警示。

未建立此最佳實務時的風險暴露等級：低

### 實作指引

使用警示觸發自動化動作，盡可能修復問題。如果無法自動回應，則將警示上報給能夠回應的人員。例如，您可能有一個可以預測關鍵績效指標 (KPI) 預期值，且會在超過特定閾值時發出警示的系統，或是在 KPI 超出預期值時可以自動停止或回復部署的工具。

實作可在工作負載執行時提供效能可見度的程序。建置監控儀表板並建立效能預期的基準規範，以確定工作負載是否以最佳狀態執行。

### 實作步驟

- 識別修復工作流程：識別並了解可自動修復的效能問題。使用 [Amazon CloudWatch](#) 或 AWS X-Ray 等 AWS 監控解決方案，以協助您更完整了解問題的根本原因。

- 定義自動化程序：建立可用於自動修正問題的逐步修復程序。
- 設定啟動事件：設定事件以自動啟動修復程序。例如，您可以定義觸發程式，在執行個體達到特定 CPU 使用率閾值時自動重新啟動執行個體。
- 自動化修復：使用 AWS 服務和技術以自動化修復程序。例如，[AWS Systems Manager 自動化](#) 提供安全且可擴展的方式，來自動化修復程序。如果變更未成功解決問題，請務必使用自我修復邏輯以回復變更。
- 測試工作流程：在試生產環境中測試自動修復程序。
- 實作工作流程：在生產環境中實作自動修復。
- 開發手冊：開發並記錄一本手冊，其中概述修復計劃的步驟，包括啟動事件、修復邏輯和採取的動作。務必培訓利害關係人，協助他們有效回應自動修復事件。
- 審查和調整：定期評估自動修復工作流程的有效性。視需要調整啟動事件和修復邏輯。

## 資源

### 相關文件：

- [CloudWatch 文件](#)
- [監控、記錄和效能 AWS Partner Network 合作夥伴](#)
- [X-Ray 文件](#)
- [使用 CloudWatch 中的警示和警示動作](#)
- [建置雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務](#)
- [透過自動表格最佳化，自動調整 Amazon Redshift 效能](#)

### 相關影片：

- [AWS re:Invent 2023 - 自動擴展、修復和智慧型自我修復策略](#)
- [AWS re:Invent 2023 - \[LAUNCH\] 現代工作負載的應用程式監控](#)
- [AWS re:Invent 2023 - 實作應用程式可觀測性](#)
- [AWS re:Invent 2021 - 以明智的方式自動化雲端操作](#)
- [AWS re:Invent 2022 - 在 AWS 環境中大規模設定控制項](#)
- [AWS re:Invent 2022 - 使用 AWS 以自動化修補程式管理及合規作業](#)
- [AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能](#)
- [AWS re:Invent 2023 - 減輕負擔：使用 Amazon RDS 診斷並解決效能問題](#)



- [AWS re:Invent 2021 - {新啟動} 使用 Amazon DevOps Guru 自動偵測並解決問題](#)
- [AWS re:Invent 2023 - 集中化您的營運](#)

相關範例：

- [CloudWatch Logs 自訂警示](#)

## PERF05-BP06 即時更新工作負載和服務的狀態

掌握最新的雲端服務和特徵，以採用高效功能、解決問題，並改善工作負載的整體效能效率。

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。

建立此最佳實務的好處：透過建立程序掌握新服務和供應項目的最新資訊，您就可以採用新特徵和功能、解決問題並改善工作負載效能。

未建立此最佳實務時的風險暴露等級：低

### 實作指引

當新服務、設計模式和產品特徵推出時，評估提升效能的方法。透過評估、內部討論或外部分析，確定哪些方法可以提高工作負載效能或效率。定義程序來評估與工作負載相關的更新、新功能和服務。例如，建立新技術的使用概念證明或與內部小組協商。嘗試新的想法或服務時，執行效能測試以衡量其對工作負載效能的影響。

### 實作步驟

- 清查工作負載：清查工作負載軟體和架構，並識別需要更新的元件。
- 確定更新來源：找出與工作負載元件相關的新聞和更新來源。舉例來說，您可以訂閱 [AWS 最新消息部落格](#)，以了解與您工作負載元件相符的產品。您可以訂閱 RSS 摘要或管理 [電子郵件訂閱](#)。
- 定義更新排程：定義排程以評估工作負載的新服務和特徵。
  - 您可以使用 [AWS Systems Manager Inventory](#)，從 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軟體政策所需的軟體與組態，以及哪些執行個體需要更新。

- 評估新的更新：了解如何更新工作負載的元件。利用雲端的靈活性快速測試新特徵對工作負載有何改善，藉以提高效能效率。
- 使用自動化：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。
- 記錄程序：記錄您在評估更新和新服務的過程。向擁有者提供所需的時間和空間，來研究、測試、試驗和驗證更新及新服務。回顧所記錄的業務需求和 KPI，來協助排定哪個更新將帶來正面業務影響的優先順序。

## 資源

相關文件：

- [AWS 部落格](#)
- [AWS 最新消息](#)
- [利用自動化 EC2 Image Builder 管道實作最新影像](#)

相關影片：

- [AWS re:Inforce 2022 - 使用 AWS，自動化修補程式管理與合規](#)
- [所有物件修補程式：AWS Systems Manager | AWS 事件](#)

相關範例：

- [庫存與修補程式管理](#)
- [一個觀察工作坊](#)

## PERF05-BP07 定期檢閱指標

作為日常維護或對事件或事故的回應，檢閱所收集的指標。透過這些審查來識別哪些指標是解決問題的關鍵，以及哪些其他指標 (如果被追蹤) 將有助找出、解決或預防問題。

常見的反模式：

- 您讓指標長時間持續處於警示狀態。
- 您建立自動化系統無法採取行動的警示。

建立此最佳實務的好處：持續檢閱正在收集的指標，以確認指標的識別和處理是否正確或防止問題發生。如果讓指標長時間持續處於警示狀態，指標也會變得過時。

未建立此最佳實務時的風險暴露等級：中

## 實作指引

不斷改進指標收集和監控。作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，從而防止事故發生或更快地解決將來的事故。

作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，進而防止事故發生或更快地解決將來的事故。

## 實作步驟

- 定義指標：定義與工作負載目標一致的關鍵績效指標以利於監視，包括回應時間和資源使用率等指標。
- 建立基準：為每個指標設定基準和所需的值。基準應提供參考點以識別偏差或異常狀況。
- 設定節奏：設定節奏 (例如每週或每月一次) 以審查重要指標。
- 識別效能問題：在每次審查期間評估趨勢以及與基準值的偏差。查看是否有任何效能瓶頸或異常情況。對於已確認的問題，請展開深入根本原因分析，以了解問題背後的主要原因。
- 識別糾正動作：使用您的分析來識別糾正動作。這可能包括參數調整、修復錯誤和調整資源規模。
- 記錄調查結果：記錄您的調查結果，包括已識別的問題、根本原因和糾正動作。
- 反覆執行與改善：持續評估並改善指標審查流程。利用先前審查中學到的教訓，以隨時間強化流程。

## 資源

相關文件：

- [CloudWatch 文件](#)
- [使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌](#)
- [使用 CloudWatch Metrics Insights 查詢您的指標](#)

- [監控、記錄和效能 AWS Partner Network 合作夥伴](#)
- [X-Ray 文件](#)

相關影片：

- [AWS re:Invent 2022 - 在 AWS 環境中大規模設定控制項](#)
- [AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能](#)
- [AWS re:Invent 2023 - 打造有效的可觀測性策略](#)
- [AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控](#)
- [AWS re:Invent 2023 - 減輕負擔：使用 Amazon RDS 診斷並解決效能問題](#)

相關範例：

- [使用 Amazon QuickSight 建立儀表板](#)
- [CloudWatch 儀表板](#)

## 結論

為實現和維持效能達成效率，需要採取資料驅動的方法。您應積極考慮存取模式和權衡因素，以便讓您能夠執行優化進而獲得更高效能。使用以基準化分析和負載測試為基礎的審查程序，可讓您選取適當的資源類型和組態。將基礎設施視為程式碼可讓您快速安全地發展架構，同時可使用資料針對架構作出以事實為基礎的決策。結合使用主動監控和被動監控，可確保您架構的效能不會隨著時間的推移而降低。

AWS 致力於協助您建置既能提供商業價值又能有效執行的架構。使用本白皮書中所討論的工具和技術以確保成功。

# 作者群

協力完成本文件的個人與組織如下：

- Sam Mokhtari , Amazon Web Services 資深效率主管與解決方案架構師
- Josh Hart , Amazon Web Services 解決方案架構師
- Richard Trabing , Amazon Web Services 解決方案架構師
- Brett Looney , Amazon Web Services 解決方案首席架構師
- Nina Vogl , Amazon Web Services 解決方案首席架構師
- Eric Pullen , Amazon Web Services 解決方案架構師
- Julien Lépine , Amazon Web Services 專家 SA 經理
- Ronnen Slasky , Amazon Web Services 解決方案架構師

## 深入閱讀

如需其他協助，請參考以下資源：

- [AWS Well-Architected Framework](#)
- [AWS 架構中心](#)

## 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                |
|---------------------------|---|-------------------|
| <a href="#">白皮書已更新</a>    | 最佳實務更新了新的實作指引。  | June 27, 2024     |
| <a href="#">主要更新與重新建構</a> | 支柱經過重新建構，分成五個最佳實務領域 (原為八個)。內容已整合成五個領域並已更新。<br><br>新的最佳實務領域包括 <a href="#">選擇架構</a> 、 <a href="#">運算與硬體</a> 、 <a href="#">資料管理</a> 、 <a href="#">網路與內容交付</a> 和 <a href="#">程序和 文化</a> 。 | October 3, 2023   |
| <a href="#">小幅度更新</a>     | 移除非包容性語言。   | April 13, 2023    |
| <a href="#">新框架的更新</a>    | 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    |
| <a href="#">白皮書已更新</a>    | 最佳實務更新了新的實作指引。  | December 15, 2022 |
| <a href="#">白皮書已更新</a>    | 已擴充最佳實務並新增了改善計劃。  | October 20, 2022  |
| <a href="#">小幅度更新</a>     | 已移除非包容性語言。  | April 22, 2022    |
| <a href="#">小幅度更新</a>     | 已將永續性支柱新增至簡介。   | December 2, 2021  |
| <a href="#">小幅度更新</a>     | 更新了連結。  | March 10, 2021    |
| <a href="#">小幅度更新</a>     | 將 AWS Lambda 逾時變更為 900 秒，並更正 Amazon   | October 5, 2020   |



---

|                        |  |                  |
|------------------------|--|------------------|
|                        | Keyspaces (for Apache Cassandra) 的名稱。          |                  |
| <a href="#">小幅度更新</a>  | 修正了中斷的連結。                                      | July 15, 2020    |
| <a href="#">新框架的更新</a> | 對內容進行了重大審查和更新                                  | July 8, 2020     |
| <a href="#">白皮書已更新</a> | 語法問題的小幅度更新                                     | July 1, 2018     |
| <a href="#">白皮書已更新</a> | 重新整理白皮書以體現 AWS 上的變更                            | November 1, 2017 |
| <a href="#">初版</a>     | 效能達成效率支柱 – AWS Well-Architected Framework 已發佈。 | November 1, 2016 |