



AWS 大型遷移的策略和最佳實務

# AWS 規範指南



# AWS 規範指南: AWS 大型遷移的策略和最佳實務

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

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

# Table of Contents

簡介 .....	1
大型移轉指南 .....	1
範圍、策略、時間表 .....	3
範圍 — 您正在遷移什麼？ .....	3
策略 — 為什麼要移轉？ .....	3
時間軸：您需要在什麼時候完成遷移？ .....	4
最佳實務 .....	5
人員 .....	5
行政支持 .....	5
團隊協作和所有權 .....	6
培訓 .....	7
技術 .....	8
自動化、追蹤與工具整合 .....	8
必要條件和移轉後驗證 .....	10
處理 .....	11
準備大型移轉 .....	11
執行您的大型移轉 .....	15
其他考量 .....	17
結論 .....	20
資源 .....	21
AWS大型遷移 .....	21
相關AWS規範指引資源 .....	21
其他參照 .....	21
影片 .....	21
貢獻者 .....	22
文件歷史紀錄 .....	23
詞彙表 .....	24
# .....	24
A .....	24
B .....	27
C .....	28
D .....	31
E .....	34
F .....	36

---

G .....	37
H .....	37
I .....	38
L .....	40
M .....	41
O .....	45
P .....	47
Q .....	49
R .....	49
S .....	52
T .....	54
U .....	56
V .....	56
W .....	56
Z .....	57
.....	lviii

# AWS 大型移轉的策略和最佳做法

Amazon Web Services ( [貢獻者](#) )

2022 年 5 月 ( [文件歷史記錄](#) )

許多 AWS 客戶希望盡可能快地將大量伺服器 and 應用程式移轉至，但對業務的影響最小。AWS 雲端 您的組織可能會啟動大型移轉專案，因為資料中心租用即將續約或終止，或是因為您的組織正在採取技術轉型的第一步。但是，大規模不會僅由範圍內的伺服器數量進行量化。它還考慮了人員、流程、技術和優先順序等複雜性，從而考慮遷移所產生的組織轉型層級。

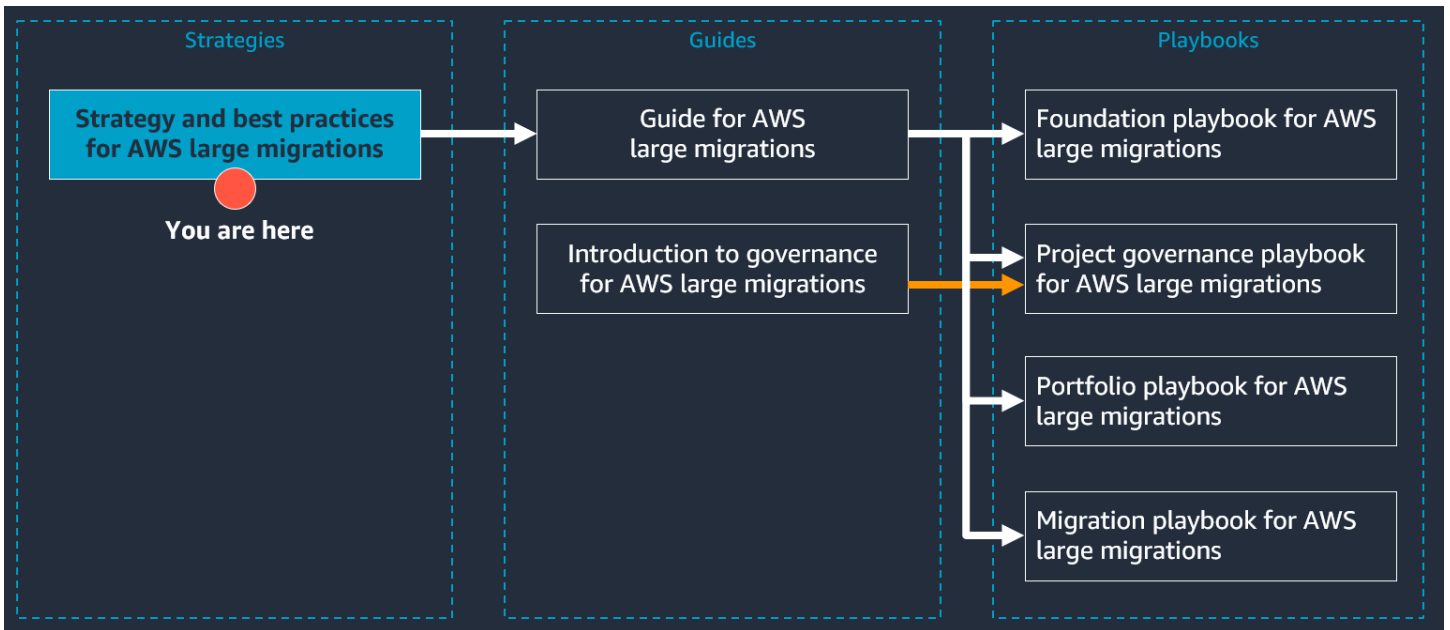
本指南著重於您大規模移動到的能力 AWS。您可以移轉現有的應用程式，幾乎不需要變更，甚至 您可以使用雲端做為啟動點，將這些應用程式帶到雲端原生或無伺服器技術，並且可以將應用程式現代化，以發揮更多的業務優勢。

本指南討論大規模移轉的最佳做法，並提供來自不同領域 (例如金融服務和醫療保健) 客戶的使用案例。它還提供了在客戶遷移到 AWS. 本指南的目的是協助處於大規模移轉初始階段的客戶。不過，在移轉過程的任何階段，本指南中的最佳做法和策略都是有益的。假設您已經擁有 100 級的知識，AWS 服務 並且知道 [AWS 建議的遷移程序](#)。

## 大型移轉指南

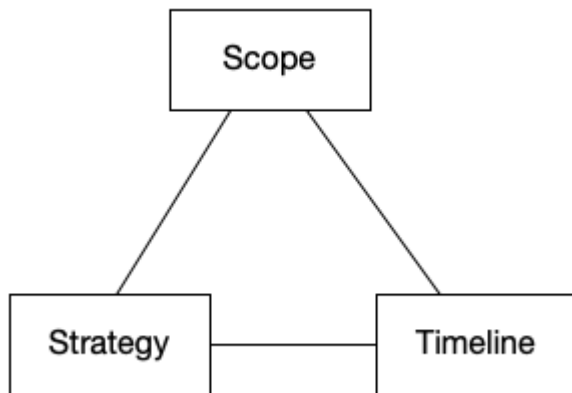
遷移 300 台以上的伺服器被視為是一項大型遷移。對於大多數企業而言，大型移轉專案所面臨的人員、流程和技術挑戰通常都是新手。本文件是有關大型移轉至 AWS . AWS 雲端本系列旨在幫助您從一開始就應用正確的策略和最佳實務，以簡化您的雲端旅程。

下圖顯示了此系列中的其他文件。首先查看策略，然後查看指南，然後繼續閱讀教戰手冊。若要存取完整序列，[請參閱 AWS 雲端](#)。



# 範圍、策略和時間表

三個關鍵要素構成了所有程序的構建組塊及其在大型遷移中的相關性：範圍、策略和時間表。



若要設定移轉歷程的階段，必須從移轉程式開始就對齊並瞭解這些元素。對這些元素之一的任何變更都會影響其他元素。重新調整必須考慮到每一個變化，無論變化看起來多麼基本或明智。

## 範圍 — 您正在遷移什麼？

即使您處於遷移的中途，程序的總範圍也很常見。這是因為各種因素可能要等到後期階段才能解壓縮。例如，在移轉過程中，您可能會發現未記錄在組態管理資料庫 (CMDB) 中的陰影 IT 口袋。或者，規劃可能集中在伺服器檢視上，而不考慮執行這些應用程式所需的支援網路和安全性服務 (例如與AWS合作夥伴的 VPN 連線，以及憑證授權單位簽署憑證)。我們建議投入一些時間來定義範圍，從您的目標業務結果向後工作。您最終可能會使用探索工具來發現資產，這是本指南稍後將討論的最佳做法。

範圍將改變，因為大型遷移帶有未知數。這些未知數可能是系統的形式，這些系統已成為環境考古學的一部分，幾乎不了解它們的相關性，或者導致延遲和轉移到您所制定的計劃的生產事件。關鍵是要靈活，並制定應變計劃以保持計劃向前發展。

## 策略 — 為什麼要移轉？

您可能會因以下一或AWS多個原因而計劃移轉至：

- 您的應用程式團隊想要實作新的 CI/CD 管線、部署最新的應用程式堆疊，或將未支援的舊版平台現代化。
- 您的基礎架構團隊必須在租賃到期之前迅速離開老化的資料中心，並由供應商關閉電源。
- 董事會已決定您需要移轉至雲端作為策略方向，讓企業 future 的快速變化步伐。

無論是什麼原因，所有這些原因和更多原因都將放在您的業務和 IT 組織的心中。了解驅動程序是什麼，溝通它們並優先考慮它們是關鍵。每個額外的驅動程式都可能增加時間、成本、範圍和風險，以至您已經大規模的遷移。充分了解策略對時間表和範圍的影響是關鍵。

定義移轉策略之後，成功的主要關鍵之一就是協調各種利益相關者和團隊的需求。執行移轉需要整個組織中的不同團隊，包括基礎結構、安全性、應用程式和作業。這些團隊將擁有個人優先事項和其他可能已經開始的項目。如果這些團隊正在努力實現不同的時間表和優先順序，則同意並實施遷移計劃會更具挑戰性。遷移團隊和主要利益相關者必須確保所有參與的團隊都朝著單一目標努力，並將其優先順序與單一遷移時間表保持一致。

我們建議您探索如何在各個團隊之間保持一致所需的業務成果。例如，移轉至AWS並使用AWS Key Management Service (AWS KMS) 來加密靜態儲存區，可能會同時滿足移轉和安全性目標。

通常，企業希望將應用程式現代化，這可能會導致基礎架構升級，而基礎架構團隊則希望節儉並將基礎架構變更減至最少。大型遷移的心態應盡可能基本。參與的團隊必須避免嘗試一次做所有事情。

為了實現這一目標，請在項目的早期設置正確的期望。關鍵訊息應該是「先移轉，然後再進行現代化」。這種方法不僅使組織能夠減少技術債務並最終大規模運營，還AWS 雲端可以使用可提供的可擴展性和敏捷性，為不同的現代化方法開闢了途徑。長遠思考將有助於基礎設施團隊簡化基礎設施的部署和管理。因此，企業可以擁有更快的功能發布週期。

## 時間軸：您需要在什麼時候完成遷移？

根據您的業務案例，你必須確保你沒有採取超過可能實現在分配的時間。如果您要移轉的動因是以固定的完成日期為基礎，您必須選擇符合該時間表需求的策略。大多數大型移轉都是以這些時間為基礎的限制，因此遷移策略必須定義固定的時間表和結果，並且沒有擴充或超限的空間。

在這些時間敏感的移轉類型中，我們建議使用「先移轉，然後再進行現代化」方法。這有助於設定期望，並鼓勵團隊確保他們的個別專案計劃和預算符合整體移轉目標。重要的是要儘早找出項目中的任何分歧，快速失敗並解決指導委員會級別上的分歧，並吸引正確的利益相關者以確保對齊到位。

相反地，如果您移轉的主要目標是獲得應用程式現代化的好處，則必須在計畫的早期提出。許多計畫都是以固定截止日期為基礎的初始目標開始，而且不會針對想要解決未解決問題和問題的利益相關者的需求進行規劃。在某些情況下，這些問題在源系統中已經存在了多年，但現在它們已成為移轉的人工阻止劑。

移轉期間的現代化活動可能會影響商務應用程式的功能。即使是被認為是小型升級，例如操作系統版本更改，也可能對程序時間表產生重大影響。這些不應該被認為是微不足道的。



# 大型移轉的最佳做法

根據管理組織運作方式的因素而定，大型移轉可能會變得具有挑戰性。本節介紹了一些關鍵因素，如果在工作的初始階段進行解決並在整個專案中進行追蹤，這些因素可以簡化大型移轉。

下列大型移轉的最佳作法是根據從其他客戶擷取的資料。最佳做法分為三類：

- 人員
- 技術
- Processes

## 人的角度

本節著重於以下人員觀點的關鍵領域：

- 執行支援 — 識別有權做出決策的單執行緒領導者
- 團隊協作和所有權 — 在各個團隊之間進行合作
- 培訓 — 積極培訓團隊進行各種工具

## 行政支持

在本節中：

- [點選單螺紋引線](#)
- [調整高級領導團隊](#)

## 點選單螺紋引線

開始進行大型移轉時，請務必確定一位 100% 致力於專案且負責的單執行緒技術領導者。該領導者能夠通過保持一致的優先級來做出決策，幫助避免孤島並簡化工作流程。

一個大型的全球遷移客戶可以在計劃開始時，每週從一台服務器擴展到第二個月初，每週將一台服務器擴展到 80 多台服務器。作為單執行緒領導者，CIO 的全面支援對於快速擴充遷移的伺服器至關重要。CIO 每週與遷移團隊一起參加移轉切換電話，以確保即時上報和解決問題，進而加速移轉速度。

## 調整高級領導團隊

在各個團隊之間建立與遷移成功標準相關的一致性非常重要。雖然遷移規劃和實作可由小型專屬的團隊完成，但在定義策略和執行周邊活動時會出現挑戰。這些潛在的障礙可能需要採取行動或升級來自 IT 組織的不同領域，包括以下內容：

- 商業
- 應用程式
- 聯網
- 安全
- 基礎設施
- 第三方廠商

應用程式擁有者的直接行動、領導能力、一致性，以及向單執行緒領導者的明確升級變得非常重要。

## 團隊協作和所有權

在本節中：

- [建立跨功能雲端支援團隊](#)
- [事先定義核心遷移團隊以外的團隊和個人的需求](#)
- [驗證移轉工作負載時沒有授權問題](#)

### 建立跨功能雲端支援團隊

大型移轉專案的關鍵第一步是讓組織能夠在雲端工作。若要達成此目的，我們建議您建置[雲端啟用引擎](#) (CEE)。CEE 是一個有權力和負責任的團隊，專注於組織遷移到 AWS。CEE 應該是一個跨職能團隊，其中包括基礎結構、應用程式、作業和安全性的代表性。該團隊負責以下職責：

- 制定政策
- 定義和實施將建立組織雲運營模型的工具，流程和架構
- 繼續促進利益相關者在他們所代表的所有領域中對齊

一位醫療保健客戶並非從 CEE 開始。但是，通過初始試點遷移，確定了差距。在最終移民切換日期之前，團隊實施了移民作戰室，並在嚴格的截止日期之前實施。在遷移戰爭室中，來自基礎架構、安全性、應用程式和業務的利益相關者可以協助解決問題。

## 事先定義核心遷移團隊以外的團隊和個人的需求

識別核心方案外的團隊和個人，並定義他們在移轉規劃階段的參與程度。為了在後期階段促進遷移的動力，請特別注意應用程序團隊的參與。需要他們對應用程序的知識，診斷問題的能力以及在切換時簽署的要求。

雖然移轉將由核心團隊領導，但應用程式團隊可能會參與驗證移轉計劃和在切換期間進行測試。客戶通常會以基礎架構專案的形式來處理雲端移轉，而不是作為應用程式移轉。這可能會導致移轉期間發生問題。

我們建議您在選取移轉策略時考慮應用程式團隊所需的參與。例如，與正在變更更多應用程式環境的重新平台或重構策略相比，重新裝載策略需要較少的應用程式團隊參與。如果應用程式擁有者的可用性有限，請考慮使用重新裝載或重新平台，而不是重構、重新定位或回購策略。

### 驗證移轉工作負載時沒有授權問題

當您將公司現成的產品遷移到雲端時，授權可能會有所變更。您的授權合約可能集中在您的內部部署遺產上。例如，授權可能是由 CPU 或連結至特定 MAC 位址。或者，授權合約可能不包含在公有雲環境中託管的權限。但是，與廠商重新協商授權可能會包含較長的交貨時間，並為移轉提供硬式封鎖程式。

我們建議您在定義移轉範圍後立即與您的採購或供應商管理團隊合作。授權也可能會影響您的目標架構和移轉模式。

## 培訓

在本節中：

- [訓練團隊有關新工具和流程](#)

### 訓練團隊有關新工具和流程

定義移轉策略之後，請投入時間來瞭解移轉和目標作業模型可能需要哪些訓練。在移轉期間，您可能會使用組織的新工具AWS Database Migration Service，例如工具。主動訓練團隊可減少移轉階段所遇到的延遲情況。

我們建議您尋求積極的知識轉移方法，這些方法可以提供一個機會以動手的方式嘗試工具。舉例來說，AWS專業服務為負責大型移轉的三個系統整合商 (SI) 合AWS作夥伴提供了數個雲端移轉工廠訓練課程。這樣可以確保團隊在進入移轉階段時具有基本的熟悉度。它還幫助確定了可以在每個 SI AWS 合作夥伴團隊中擔任第一線升級的主題專家 (SME)。

## 技術角度

技術為加速大型移轉提供了良好的基礎。例如，雲端移轉工廠解決方案著重於如何為移轉提供 end-to-end 自動化。本節將探討使用技術達到所需規模和速度的一些最佳實務，並符合範圍、策略和時間表。

總體原則是盡可能地查看自動化領域。如果您的範圍內有數千部伺服器，手動執行工作可能是一項昂貴且耗時的工作。

若要執行移轉，通常會使用數種工具，例如：

- 探索
- 遷移實施
- 組態管理資料庫 (CMDB)
- 庫存試算表
- 專案管理

這些工具用於遷移的不同階段，從評估到動員到實施。這些工具的選擇是由業務目標和時間表驅動的。

規劃好移轉階段之後，下一個步驟是確保移轉小組具備使用所需工具的技能。如果團隊缺乏技能或經驗，請計劃有針對性的培訓以提高技能。如果可能，請建立事件，讓團隊可以在安全的環境中取得移轉工具的經驗。例如，團隊是否有沙坑或實驗室伺服器可以移轉以體驗工具？或者，初始開發工作負載用於學習目的是否可以接受？

## 自動化、追蹤與工具整合

在本節中：

- [自動化移轉探索以減少所需的時間](#)
- [自動化重複工作](#)
- [自動化追蹤和報告，加快決策速度](#)
- [探索可協助您移轉的工具](#)

### 自動化移轉探索以減少所需的時間

大部分的大型移轉程式都是透過瞭解移轉的範圍 (必須移轉的項目) 和制定策略 (如何移轉) 開始進行。發現是這方面的一個重要方面。會擷取所需的中繼資料點，以形成移轉策略決策樹。若要以步調移轉工作負載，您必須識別必要的移轉中繼資料，並將其匯入實作程序，例如移轉工廠。擷取、轉換、載入 (ETL) 遷移中繼資料的全自動化機制，可大幅縮短探索程序所需的時間和工作量。

一位客戶為遷移工廠開發了完全自動化的資料擷取程序。包含所有移轉中繼資料的遷移波計劃都是在 Microsoft 上的試算表中託管和維護 SharePoint。對來源進行變更後，會啟動一個 AWS Lambda 函數，將資料載入移轉工廠，而無需手動介入。這個自動化的資料擷取程序可協助客戶減少手動工作、減少人為錯誤，並加速他們的速度。他們能夠將超過 1,000 部伺服器遷移到 AWS。

## 自動化重複工作

在移轉實作階段，許多小型程序必須經常重複執行。例如，使用 AWS Application Migration Service (MGN) 時，您必須在移轉範圍內的每部伺服器上安裝代理程式。

建置符合特定業務與技術需求的移轉工廠，是達成成功大型移轉所需效率與速度的最有效方法。移轉工廠提供整合與協調架構，該架構使用標準化資料集來加速移轉。確定所有任務後，花時間自動化所有可以與規範性 Runbook 一起自動化的手動任務。

[雲端移轉工廠](#) 解決方案就是其中一個範例。雲端移轉工廠的設計目的是提供移轉自動化基礎，您可以在此基礎上自動化組織的特定層面。例如，您可能想要更新 CMDB 中的旗標，以反白顯示內部部署伺服器現在可以解除委任。在這個案例中，您可以建立在移轉波結束時執行此工作的自動化。雲端移轉工廠擁有集中式的中繼資料儲存區，其中包含所有波形、應用程式和伺服器中繼。自動化腳本可以連接到雲遷移工廠以獲取該波中的服務器列表，並相應地執行任何操作。雲端移轉工廠支援 [AWS Application Migration Service](#)。

## 自動化追蹤和報告，加快決策速度

我們建議您建立自動化移轉報告儀表板，以追蹤和報告即時資料，包括計畫的關鍵績效指標 (KPI)。移轉專案涉及整個組織的利益相關者，包括下列各項：

- 應用團隊
- 測試人員
- 退役團隊
- 建築師
- 基建團隊
- 領導

為了履行其角色，這些利益相關者需要即時資料。例如，網路團隊必須瞭解即將到來的遷移浪潮，才能瞭解內部部署資源與 AWS。領導團隊想要瞭解遷移的程度已完成多少。擁有可靠、自動化的資料即時摘要可防止錯誤通訊，並提供可以做出決策的基礎。

一位大型醫療保健客戶正在努力邁向資料中心的退出，並且即將到來的期限考慮到規模和複雜性，最初花費了大量時間來跟踪和傳達利益相關者之間的遷移狀態。遷移團隊後來使用 Amazon QuickSight 建立可視化資料的自動化儀表板，大幅簡化追蹤和通訊，同時提高遷移速度。

## 探索可協助您移轉的工具

為您的移轉選擇正確的工具並不容易，尤其是組織中沒有人管理過大型移轉時。

我們建議您花時間選擇合適的工具來支援移轉。這項探索可能涉及許可證成本，但是當您考慮到更廣泛的計劃時，它可以提供成本效益。或者，您可能會發現內嵌在組織中的工具可以提供類似的結果。例如，您可能已經在資產中部署了應用程式效能監視工具，可提供豐富的探索資訊。

由於缺乏熟悉度，技術客戶最初不願在遷移期間執行自動化探索工具。因此，SI AWS 合作夥伴必須為每個應用程式執行 510 小時的會議，才能手動探索資產，包括伺服器名稱、作業系統版本和相依性。據估計，如果已使用探索工具，探索工作可能會減少超過 1,000 小時。

## 必要條件和移轉後驗證

在本節中：

- [在遷移前階段建立 landing zone](#)
- [大綱前提活動](#)
- [實施遷移後檢查以持續改進](#)

### 在遷移前階段建立 landing zone

我們建議您事先建置AWS目標環境或 landing zone 域，而不要在移轉波動期間建置目標虛擬私有雲端 (VPC) 和子網路。建立架構良好的 landing zone 是移轉的先決條件。landing zone 應包括監控，治理，運營和安全控制。

在移轉前建置和驗證 landing zone，可將在新環境中執行工作負載所帶來的不確定性降到最低。有了 landing zone，利益相關者就可以專注於移轉工作負載，而不必擔心帳戶或 VPC 層級管理的層面。

### 大綱前提活動

除了 landing zone 之外，在遷移之前對齊其他技術先決條件非常重要，尤其是交貨時間較長的程序。例如，進行必要的防火牆變更，以允許將資料從內部部署複製到AWS。儘早溝通技術先決條件，有助於準備和分配所需的資源。遷移到停止是很常見的，因為尚未滿足先決條件。這不僅會影響進行中的移轉浪潮，還可能會在解決問題時推回所有 future 移轉的日期。



一家金融服務公司旨在進行大規模遷移AWS，目標是騰出多個數據中心。但AWS是，它們在內部部署之間可用的帶寬並不足以滿足他們預期的速度。不幸的是，增加帶寬需要一個新的連接，並且交貨時間為三個月。這表示移轉速度在前三個月受到限制。

## 實施遷移後檢查以持續改進

最後，請記得實作移轉後驗證，例如作業整合、成本最佳化，以及治理和合規性檢查。移轉後驗證包括評估先前移轉的工作負載，以發掘應該應用於 future 浪潮的技術課程。

此外，這是實施成本控制操作的絕佳機會。例如，在移轉期間，您可能會決定將AWS執行個體大小與內部部署資產相符，以減少效能測試的需求。既然測試已不再在資料中心關閉關鍵路徑上，您可以使用 Amazon CloudWatch 評估執行個體使用率，並判斷較小規模的執行個體是否適合。

為了說明此階段的重要性，大型技術客戶正在執行大型移轉，但最初並未包含移轉後驗證。移轉超過 100 部伺服器之後，他們會發現AWS Systems Manager代理程式 (SSM 代理程式) 未正確設定。所有先前移轉的伺服器都必須進行修復，且移轉停止。客戶還確定執行個體的大小是初始估計值的五倍，因此他們在每個遷移波結束時實施了成本檢查點。

## 流程透視

流程帶來了一致性，但它們也會發展並且容易發生變化，因為每個項目都是獨一無二的。當您重複執行此程序時，您將找出缺口和改進空間，這些差距和改進空間可能會在失敗、學習、採用和反覆運算時帶來巨大的好處。這些變化可能會導致項目和業務將來可以利用的新想法或創新，這為增長提供了帶 future 質量和團隊信心的催化劑。

移轉中的程序可能很複雜，因為它們會跨越先前可能未連結的技術和邊界。此觀點提供有關大型移轉之特定需求的程序和指導。

## 準備大型移轉

以下各節概述了必要的核心原則，以確保您以明確的方向開始遷移之旅，並向利益相關者提供對其成功至關重要的支持。

在本節中：

- [定義業務驅動因素並溝通時間表、範圍和策略](#)
- [定義清晰的升級路徑，以幫助刪除阻止程序](#)
- [將不必要的變更](#)

- [儘早記錄一個 end-to-end 過程](#)
- [記錄標準移轉模式和人工因素](#)
- [為移轉中繼資料和狀態建立單一事實來源](#)

## 定義業務驅動因素並溝通時間表、範圍和策略

接近大型移轉至時AWS，您會很快發現有許多方法可以移轉伺服器。例如，您可以執行下列操作：

- 使用 [AWS Application Migration Service](#).
- 將您的應用程式容器化，並將其託管在 [Amazon 彈性容器服務 \(Amazon ECS\)](#) 或 [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) 受管容器平台上。
- 將您的工作負載重新設計為完全無伺服器應用程式。

若要判斷正確的移轉路徑，請務必向業務驅動程式向後工作。如果您的最終目標是提高業務敏捷性，那麼您可能會喜歡第二個模式，這兩種模式涉及更多層次的轉型。如果您的目標是在年底之前撤離資料中心，則可能會因為重新託管提供的速度而選擇重新託管工作負載。

大型移轉通常涉及廣泛的利害關係人，包括下列各項：

- 應用擁有者
- 網絡團隊
- 資料庫管理員
- 行政贊助

識別移轉的業務驅動因素，並在文件中包含該清單 (例如移轉程式成員可以存取的專案章程) 是關鍵。此外，建立與目標業務成果密切一致的關鍵績效指標 (KPI)。

例如，一位客戶想要在 12 個月內遷移 2,000 部伺服器，以達成騰出資料中心的目標業務成果。但是，他們的安全團隊沒有朝著這個目標保持一致。其結果是幾個月的技術辯論，討論是否要錯過資料中心關閉日期，但是要進一步將應用程式現代化，還是先重新託管以便及時關閉資料中心，然後在應用程式上進行現代化改造。AWS

## 定義清晰的升級路徑，以幫助刪除阻止程序

大型雲端移轉計畫通常涉及廣泛的利益相關者。畢竟，您可能正在變更已在內部部署託管數十年的應用程式。每個利益相關者都有相互矛盾的優先事項是很常見的。



雖然所有優先事項都可能推動價值，但該計劃可能具有有限的預算和定義的目標結果。管理各利益相關者並專注於目標業務成果可能具有挑戰性。當您將其乘以遷移範圍內的數百或數千個應用程式時，這項挑戰就會變得更加複雜。此外，利益相關者可能會向不同的領導團隊報告，這些團隊有其他優先事項考慮到這一點，除了清楚記錄目標業務成果之外，定義明確的升級矩陣以幫助消除阻滯劑非常重要。這可以節省大量時間，並有助於使各個團隊朝著共同目標保持一致。

證明這一點的一個例子是一家金融服務公司，其目標是在 12 個月內撤出其主要數據中心。沒有明確的任務或升級途徑，這導致利益相關者制定了他們想要的遷移路徑，無論時間和預算的限制如何。在向 CIO 升級之後，設定了明確的任務，並提供了要求所需決定的機制。

## 將不必要的變更

改變是好的，但更多的變化意味著更多的風險。核准大型移轉的商業案例後，很可能會出現推動此計劃的目標業務成果，例如在特定日期前撤離資料中心。儘管技術人員通常會想要重新編寫所有內容以充分利用 AWS 服務，但這可能不是您的業務目標。

一位客戶專注於將公司的整個 Web 規模基礎架構遷移到 AWS。他們建立了為期兩週的規則，做為防止應用程式團隊花費數月重新撰寫應用程式的機制。透過使用為期兩週的規則，當數百個應用程式必須在多年期間移動時，客戶能夠以一致的節奏維持長期遷移。如需詳細資訊，請參閱部落格文章 [「兩週規則：在 10 天內重構雲端應用程式」](#)。

我們建議盡量減少任何與業務結果不符的變化。相反，建置機制來管理 future 專案中的這些額外變更。

## 儘早記錄一個 end-to-end 過程

記錄大型移轉程序初期階段的完整移轉程序和擁有權指派。本文件對於教育所有利益相關者瞭解移轉的執行方式及其角色和責任非常重要。此文件也會協助您瞭解可能發生問題的位置，並在您進行移轉時提供程序的更新與反覆項目。

在遷移專案的開發期間，請確定已瞭解任何現有程序，並清楚記錄整合指向和相依性。包括需要與外部處理程序擁有者互動的地方，包括變更請求、服務要求、廠商支援以及網路和防火牆支援。了解該過程後，我們建議將所有權記錄在負責任的，負責的，諮詢，知情的 ( RACI ) 矩陣中，以跟踪不同的活動。若要完成程序，請識別遷移每個步驟所涉及的時間表，以建立倒數計劃。倒數計劃通常會從工作負載移轉切換日期和時間開始向後運作。

對於跨國家用電器公司而言，該公司在不到一年的時間內 AWS 成功遷移並退出了四個數據中心，這種文檔方法非常有效。他們有六個不同的組織團隊和多個第三方參與，這些負擔引入了管理開銷，從而導致 back-and-forth 決策和實施延遲。AWS 專業服務團隊與客戶及其第三方一起識別遷移活動的關鍵流

程，並將其記錄給個別擁有者。所有參與團隊共享並同意產生的 RACI 矩陣。使用 RACI 矩陣和升級矩陣，客戶減輕了造成延遲的阻礙和問題。然後，他們能夠提前退出數據中心。

在使用 RACI 和升級矩陣的另一個例子中，一家保險公司能夠在不到 4 個月的時間內退出數據中心。客戶瞭解並實作共同的責任模型，並遵循詳細的 RACI 矩陣，以追蹤整個移轉期間每個流程和活動的進度。因此，客戶在實作的前 12 週內就能遷移超過 350 部伺服器。

## 記錄標準移轉模式和人工因素

將其視為為實現創建 cookie 切割器。可重複使用的參考，文檔，手冊和模式是擴展的關鍵。這些記錄了 future 遷移專案可以重複使用和避免的經驗、學習、陷阱、問題和解決方案，從而顯著加速移轉。模式和文物也是一項投資，將有助於改善流程並指導 future 的項目。

例如，一位客戶正在執行為期一年的遷移，其中三個不同的 SI AWS 合作夥伴正在遷移應用程式。在早期階段，每個 AWS 合作夥伴都在使用自己的標準、手冊和工件。這給客戶團隊帶來了許多壓力，因為相同的信息可以以不同的方式呈現給他們。在這些早期的困難之後，客戶建立了要用於移轉的所有文件和成品的中央擁有權，並提交建議變更的程序。這些資產包括以下內容：

- 標準移轉程序和檢查清單
- 網狀圖樣式與格式標準
- 基於業務重要性的應用程式架構和安全標準

此外，對任何這些文件和標準的更改都會每週發送給所有團隊，並且要求每個合作夥伴確認收到並遵守任何更改。這大幅改善了遷移專案的通訊和一致性，而當另一個業務單位開始進行大量遷移工作時，該團隊能夠採用現有的流程和文件，大幅加速他們的成功。

## 為移轉中繼資料和狀態建立單一事實來源

規劃大型移轉時，建立事實來源對於讓各個團隊保持一致並實現資料導向的決策非常重要。當您開始此旅程時，您可能會發現可以使用的許多資料來源，例如組態管理資料庫 (CMDB)、應用程式效能監視工具、清單清單等。

或者，您可能會發現資料來源很少，因此您必須建立擷取所需資料的機制。例如，您可能需要使用探索工具來揭露技術資訊，並調查 IT 主管以取得企業資訊。

請務必將各種資料來源彙總到可用於移轉的單一資料集中。然後，您可以在實施期間使用單一事實來源追蹤遷移。例如，您可以追蹤已移轉的伺服器。

想要移轉所有工作負載的金融服務客戶，AWS 專注於使用已提供的資料集規劃移轉。此資料集存在重要差距，例如業務重要性和相依性資訊，因此該計畫開始了探索練習。

在另一個範例中，同一產業的公司根據對其伺服器基礎結構清查的 out-of-date 瞭解，進入了移轉波的實作。他們很快就開始看到遷移數量減少，因為數據不正確。在這種情況下，應用程式所有者不被理解，這意味著他們無法及時找到測試人員。此外，資料與應用程式團隊已完成的解除委任不一致，因此伺服器在執行時不會用於商業目的。

## 執行您的大型移轉

在您建立業務成果並將策略傳達給利益相關者之後，您可以轉到規劃如何將大型移民的範圍開拓為可持續移民事件或浪潮。以下範例提供製定波浪計畫的關鍵指引。

在本節中：

- [提前規劃遷移波，確保穩定的流量](#)
- [將波浪實施和波浪規劃保持為獨立的流程和團隊](#)
- [從小規模開始，取得良好](#)
- [最小化切換視窗的數目](#)
- [快速失敗、套用體驗並重複執行](#)
- [不要忘記回顧](#)

### 提前規劃遷移波，確保穩定的流量

規劃移轉是計畫最重要的階段之一。這句話說：「如果你沒有計畫，你打算失敗。」提前規劃移轉波可讓專案在團隊變得更加主動地處理遷移情況時迅速流動。它可以幫助項目更輕鬆地擴展，並且隨著項目需求的增加和變得複雜，它改善了決策和預測。事先規劃也提高了團隊適應變化的能力。

例如，一位大型金融服務客戶正在進行數據中心退出計畫。最初，客戶以連續的方式規劃遷移波，在開始計畫下一波之前完成一波。這種方法導致更少的準備時間。當利益相關者收到其應用程式移轉至的通知時AWS，在開始移轉之前，他們仍然需要執行幾個步驟。這為程序增加了顯著延遲。客戶意識到這一點之後，他們實作了以未來為中心的整體移轉規劃串流，並提前數個月就規劃了移轉波。這為應用程式團隊提供了足夠的通知，以執行其遷移前活動，例如通知AWS合作夥伴，授權分析等。然後，他們可以從程序的關鍵路徑中刪除這些任務。

### 將波浪實施和波浪規劃保持為獨立的流程和團隊

當波浪規劃和波浪實施團隊是分開的時候，這兩個流程可以 parallel 工作。透過通訊與協調，這可避免拖慢移轉速度，因為伺服器或應用程式沒有足夠的準備好達到預期的速度。例如，移轉小組可能需要每週遷移 30 部伺服器，但目前的浪潮中只有 10 部伺服器已準備就緒。這個挑戰通常是由以下原因造成的：

- 遷移實施團隊沒有參與波浪規劃，並且在波浪規劃階段收集的數據不完整。移轉實作小組必須先收集更多伺服器資料，才能開始浪潮。
- 遷移實施計劃在波浪規劃之後立即開始，之間沒有緩衝區。

提前計劃波浪，並在準備和波實施開始之間創建緩衝區至關重要。同樣重要的是要確保波浪規劃團隊與移轉小組共同合作，以收集正確的資料並避免重做。

## 從小規模開始，取得良好

計劃從小規模開始，並隨後每個波浪提高遷移速度。初始波形應該是單一小型應用程式，少於 10 部伺服器。在後續浪潮中新增其他應用程式和伺服器，以達到完整的移轉速度。優先考慮較不複雜或風險的應用程序，並按計劃提高速度，使團隊有時間適應共同工作並學習流程。此外，該團隊可以識別和實施每個波的過程改進，這可以大大提高後續波的速度。

一位客戶在一年內遷移了 1,300 多台伺服器。移轉團隊從試驗式移轉和一些較小的浪潮開始，就能找出多種改善後續移轉的方法。例如，他們較早識別出新的資料中心網路區段。他們在過程早期與防火牆團隊合作，制定了允許與遷移工具進行通信的防火牆規則。這有助於防止 future 浪潮中不必要的延遲。此外，該團隊還能夠開發腳本，以幫助在每個波形中自動執行更多的發現和切換過程。從小起步幫助團隊專注於早期流程改進，並大大提高了他們的信心。

## 最小化切換視窗的數目

大規模移轉需要有紀律的方法來驅動比例。在某些區域中過於靈活，對於大型遷移而言是一種反模式。透過限制每週切換視窗的數目，用於切換活動的時間具有更高的價值。

例如，如果切換視窗太靈活，您最終可能會得到 20 個切換，每個伺服器各有五個伺服器。相反地，您可以有兩個切換，每個伺服器各有 50 部。由於每個切換的時間和精力都相似，因此切換時間較少、較大的切換可減少排程的作業負擔，並且限制不必要的延遲。

一家大型科技公司試圖在合同到期之前從幾個租賃的數據中心遷移出去。缺少到期將導致昂貴的短期續訂期限。在遷移過程中，應用程式團隊可以指定最後一分鐘的移轉排程，包括基於任何原因選擇退出移轉，只要在切換前幾天。這導致了項目早期階段的許多延遲。通常，客戶必須在最後一刻與其他應用團隊進行協商才能填寫。客戶最終提高了他們的規劃紀律，但這個早期的錯誤導致遷移團隊不斷承受 stress。整體排程延遲導致某些應用程式無法及時離開資料中心。

## 快速失敗、套用體驗並重複執行

每個遷移最初都有陷阱。儘早失敗有助於團隊學習，了解瓶頸，並將學到的經驗教訓應用於更大的浪潮。由於以下原因，預計遷移中的前幾波會很慢：

- 團隊成員正在調整彼此和過程。
- 大型遷移通常涉及許多不同的工具和人員。
- 整合、測試、失敗、學習並持續改善 end-to-end 流程需要時間。

問題是常見的，並且在第一波波的預期。了解並將其傳達給整個組織非常重要，因為有些團隊可能不喜歡嘗試新事物並失敗。失敗可能會阻止團隊，並成為 future 遷移的阻止者。確保每個人都明白初始問題是工作的一部分，並鼓勵每個人嘗試失敗是成功遷移的關鍵。

一家公司計劃在 24—36 個月內遷移超過 10,000 台伺服器。為了實現這一目標，他們需要每月遷移近 300 台伺服器。但是，這並不意味著他們從第一天就遷移了 300 台服務器。第一對夫婦浪潮是學習波浪，以便團隊能夠理解事情是如何工作的，以及誰有權做什麼。他們還確定了可以改善流程的集成，例如與 CMDB 和 CyberArk。他們利用學習浪潮來失敗，改善和再次失敗，完善了過程和自動化。6 個月後，他們每週能夠遷移 120 多台伺服器。

## 不要忘記回顧

這是一個敏捷過程的重要組成部分。這是團隊溝通，調整，學習，同意和前進的地方。在最基本層面的回顧性是回顧，討論發生了什麼，確定什麼進展順利，什麼需要改進。然後可以根據這些討論來建立改進。回顧包含一些形式或過程周圍的教訓學習的想法。回顧很重要，因為要達到大型移轉成功的規模和速度，程序、工具和團隊必須不斷發展和改進。回顧可以在這方面發揮重要作用。

傳統的課程學習課程直到計劃結束才會發生，因此通常這些課程不會在下一次遷移浪潮開始時進行審查。對於大型遷移，學到的經驗應該應用於下一波浪潮，並且應該是波浪規劃過程的關鍵部分。

對於一位客戶，每週舉行回顧，以討論和記錄從切換中汲取的經驗教訓。在這些會議中，他們發現了從流程角度或自動化進行簡化的範圍的領域。這導致實施倒數計時程表，其中包含特定活動、擁有者和自動化腳本，以最大限度地減少手動任務，包括在切換期間驗證第三方工具和 Amazon CloudWatch 代理程式安裝。

在另一家大型科技公司，定期與團隊進行回顧，以識別先前遷移浪潮的問題。這導致了程序、指令碼和自動化的改進，使平均移轉時間在程式過程中減少了 40%。

## 其他考量

許多區域都必須被納入大型移轉程式。以下各節提供了必須考慮的其他項目的想法。

在本節中：

- [隨時清理](#)
- [為任何額外的轉換實施多個階段](#)



## 隨時清理

如果遷移的費用是您預期的 10 倍，而且在關閉並清理用於遷移的資源之前，該項目仍未完成，則移轉不會成功。此清除應該是移轉後活動的一部分。它可確保您不會在環境中留下未使用的資源和服務，這些資源和服務會增加成本。移轉後清除也是防止會暴露您環境的威脅和弱點的良好安全作法。

轉移到的兩個關鍵結果AWS 雲端是節省成本和安全性。留下未使用的資源可能會破壞移至雲端的業務目的。未清除的最常見資源包括：

- 測試數據
- 測試資料庫
- 測試帳戶，包括防火牆規則、安全群組和網路存取控制清單 (網路 ACL) IP 位址
- 為測試佈建的連接埠
- Amazon Elastic Block Store (Amazon EBS) 磁碟區
- 快照
- 複寫 (例如停止將資料從內部部署複寫到AWS)
- 佔用空間的檔案 (例如用於移轉的暫存資料庫備份)
- 託管移轉工具的執行個體

在不良清理作法的一個範例中，SI AWS 合作夥伴不會在成功移轉後移除複寫代理程式。AWS稽核發現，已移轉的複寫伺服器 and EBS 磁碟區每個月的費用為 20,000 美元。為了緩解這個問題，AWS專業服務創建了一個自動審核過程，通知 SI AWS 合作夥伴時，過時的服務器仍在複製。然後，SI AWS 合作夥伴可以對未使用和過時的實例採取行動。

對於 future 的移轉，我們採用了一個程序來定義移轉後 48 小時的超級護理期間，以確保平台採用順利。客戶的基礎結構團隊接著提交內部部署伺服器的解除委任要求。建議在取消委任要求核准後，相應波形的伺服器將從應用程式遷移服務主控台中移除。

## 為任何額外的轉換實施多個階段

進行大型移轉時，請務必將注意力集中在核心目標上，例如資料中心關閉或基礎架構轉型。在較小的移轉中，範圍蠕變可能會產生最小的影響。但是，幾天的額外努力乘以數千台服務器可能會為程序增加大量時間。此外，額外的變更可能還需要更新支援團隊的文件、流程和訓練。

若要克服潛在的範圍蠕變，您可以在移轉中實作多階段方法。例如，如果您的目標是要騰出資料中心，階段 1 可能只包括將工作負載重新裝載至AWS盡可能快的速度。重新託管工作負載後，階段 2 可以實施轉型活動，而不會危及目標業務結果。

例如，一位客戶計劃在 12 個月內退出其資料中心。不過，它們的移轉包含其他轉型活動，例如推出新的應用程式效能監控工具和升級作業系統。移轉範圍內有超過 1,000 部伺服器，因此這些活動增加了移轉的大幅延遲。此外，這種方法還需要在使用新工具方面進行培訓。客戶後來決定實施一種多階段方法，最初專注於重新託管。這提高了他們的遷移速度，並降低了無法達到數據中心關閉日期的風險。

## 結論

與較小的移轉相比，大型移轉會帶來不同的挑戰。這主要是由於規模引入的複雜性。例如，將代理程式安裝到單一伺服器上相當簡單，大約需要 5 分鐘的時間。不過，如果您的移轉範圍內有 5,000 部伺服器，則大約需要 416 個小時，將會面臨下列挑戰：

- 很可能有多個操作系統需要不同的進程。
- 由於先前的合併和收購，可能會有單獨的微軟活動目錄域進行管理。
- 需要有效的程序和工具來協調每個波的代理程式安裝，然後追蹤和報告進度。

此策略概述以AWS專業服務經驗為基礎的大型移轉最佳實務，協助廣大客戶。這包括人員、流程和技術觀點。如果您想開始或正在遷移到AWS，AWS專業服務的顧問將很樂意為您提供幫助。請聯絡您的AWS代表以開始對話。

對於後續步驟，我們建議您檢閱「AWS規範指引」系列，旨在協助您規劃和完成AWS雲端。如需完整系列，請參閱[大型移轉至AWS雲端](#)。



# 資源

## AWS大型遷移

[若要存取大型移轉的完整AWS規範指引系列，請參閱AWS 雲端。](#)

### 相關AWS規範指引資源

- [透過雲端移轉工廠自動化大規模伺服器移轉](#)
- [評估應用程式在移轉至AWS 雲端](#)
- [建立安全且可擴充的多帳戶AWS環境](#)
- [評估遷移準備](#)
- [動員您的組織以加速大規模移轉](#)

### 其他參照

- [AWS雲移轉工廠解決方案](#)
- [免費雲端遷移服務AWS](#)
- [AWS Database Migration Service](#)
- [使用遷移AWS](#)

### 影片

- [執行大規模移轉至AWS \(AWS重新:發明 2020\)](#)
- [CloudEndure 遷移工廠最佳實務 \(AWSre: Invent 2020 年會\)](#)

## 貢獻者

該策略由 AWS 專業服務中的全球大型遷移老虎團隊撰寫。團隊已成功地代表 AWS 客戶將 AWS 數千台伺服器移轉至。本文件的貢獻者包括：

- 克里斯貝克, 首席產品工程師
- 德韋恩·博德隆, 高級雲端應用程式架構師
- 小魯道夫 塞拉達, 高級應用架構師
- 普拉提克, 首席雲端架構師
- 比爾大衛, 首席客戶解決方案經理
- 家偉, 高級顧問
- 呂華禮先生, 首席顧問
- 喬恩·麥迪遜, 首席雲端架構
- 奈阿布舍克, 高級解決方案建築師
- 達米安·倫納, 高級移民專家
- 阿米特·魯德拉茹, 高級雲架構師

## 文件歷史紀錄

下表說明此策略的重大變更。如果您想收到有關 future 更新的通知，您可以訂閱 [RSS 摘要](#)。

變更	描述	日期
<a href="#">CloudEndure 移除遷移服務</a>	我們移除了 CloudEndure 移轉服務的參照。AWS Application Migration Service 是移轉至的建議使用的主要 lift-and-shift 移轉服務 AWS 雲端。	2022 年 5 月 11 日
<a href="#">更新的 AWS 解決方案名稱</a>	我們已將參考 AWS 解決方案的名稱從 CloudEndure 移轉工廠更新至雲端移轉工廠。	2022 年 5 月 2 日
<a href="#">更新的資源</a>	我們使用大型遷移系列中的最新文件更新了「 <a href="#">簡介</a> 」和「 <a href="#">資源</a> 」部分。	2022 年 3 月 8 日
<a href="#">初始出版</a>	—	2021 年 9 月 16 日

# AWS 規範指南詞彙表

以下是 AWS Prescriptive Guidance 所提供策略、指南和模式的常用術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

## 數字

### 7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的內部部署 Oracle 資料庫遷移至 Amazon Aurora Postgre SQL-Compatible Edition。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將您的內部部署 Oracle 資料庫遷移至 中的 Oracle 的 Amazon Relational Database Service ( Amazon RDS ) AWS 雲端。
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 ( CRM ) 系統遷移至 Salesforce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將內部部署 Oracle 資料庫遷移至 中 EC2 執行個體上的 Oracle AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移至相同平台的雲端服務。範例：遷移 Microsoft Hyper-V 應用程式至 AWS。
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

## A

### ABAC

請參閱 [屬性型存取控制](#)。

## 抽象服務

請參閱 [受管服務](#)。

## ACID

請參閱 [原子、一致性、隔離、持久性](#)。

## 主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但比 [主動被動遷移](#) 需要更多工作。

## 主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。目標資料庫在遷移期間不接受任何交易。

## 彙總函數

在一組資料列上操作並計算該群組單一傳回值的 SQL 函數。彙總函數的範例包括 SUM 和 MAX。

## AI

請參閱 [人工智慧](#)。

## AIOps

請參閱 [人工智慧操作](#)。

## 匿名化

在資料集中永久刪除個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

## 反模式

經常性問題的常用解決方案，其解決方案具有反效益、無效或效果不如替代方案。

## 應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體侵害。

## 應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是 [產品組合探索和分析程序](#) 的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

## 人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱[什麼是人工智慧？](#)

## 人工智慧操作 (AIOps)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需遷移策略AIOps中 AWS 如何使用的詳細資訊，請參閱[操作整合指南](#)。

## 非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

## 原子度、一致性、隔離、持久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

## 屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱 AWS Identity and Access Management (IAM) 文件[ABAC AWS](#)中的。

## 權威性資料來源

您存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將來自權威資料來源的資料複製到其他位置，以處理或修改資料，例如匿名、編輯或假名化資料。

## 可用區域

與其他可用區域中的故障 AWS 區域 隔離的不同位置，並對相同區域中的其他可用區域提供低成本、低延遲的網路連線。

## AWS 雲端採用架構 (AWS CAF)

來自的指導方針和最佳實務架構 AWS，可協助組織制定高效且有效的計劃，以成功地遷移至雲端。AWS CAF 將指導方針組織到六個重點領域：業務、人員、治理、平台、安全和操作。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。為此，AWS CAF 提供人員開發、訓練和通訊的指引，協助組織成功採用雲端。如需詳細資訊，請參閱[AWS CAF 網站](#)和[AWS CAF 白皮書](#)。

## AWS 工作負載資格架構 ( AWS WQF )

評估資料庫遷移工作負載、建議遷移策略並提供工作估算的工具。AWS WQF 隨附於 AWS Schema Conversion Tool ( AWS SCT )。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

## B

### 錯誤的機器人

旨在中斷或傷害個人或組織的[機器人](#)。

### BCP

請參閱[業務持續性規劃](#)。

### 行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以搭配 Amazon Detective 使用行為圖表來檢查失敗的登入嘗試、可疑API的呼叫和類似的動作。如需詳細資訊，請參閱偵測文件中的[行為圖中的資料](#)。

### 大端序系統

首先儲存最高有效位元組的系統。另請參閱[端點](#)。

### 二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

### Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

### 藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 (藍色) 中執行目前的應用程式版本，並在另一個環境 (綠色) 中執行新的應用程式版本。此策略可協助您在影響最小的情況下快速復原。

### 機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。某些機器人很有用或有益，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。某些其他稱為壞機器人的機器人，旨在中斷或傷害個人或組織。

## 殭屍網路

受到[惡意軟體](#)感染且由單一方控制的[機器人](#)網路，稱為機器人繼承器或機器人運算子。殭屍網路是擴展機器人及其影響的最佳已知機制。

## 分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱[關於分支](#)（GitHub 文件）。

## 碎片存取

在特殊情況下，以及透過核准的程序，使用者取得其通常無權存取 AWS 帳戶之存取權的快速方法。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS [實作碎片程序](#) 指標。

## 棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和[綠地](#)策略。

## 緩衝快取

儲存最常存取資料的記憶體區域。

## 業務能力

業務如何創造價值 (例如，銷售、客戶服務或營銷)。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 [AWS 上執行容器化微服務](#) 白皮書的[圍繞業務能力進行組織](#) 部分。

## 業務連續性規劃 (BCP)

一種解決破壞性事件 (如大規模遷移) 對營運的潛在影響並使業務能夠快速恢復營運的計畫。

# C

## CAF

請參閱[AWS 雲端採用架構](#)。

## Canary 部署

版本向最終使用者緩慢且增量的版本。當您有信心時，您可以部署新版本並完全取代目前的版本。



## CCoE

請參閱 [Cloud Center of Excellence](#)。

## CDC

請參閱 [變更資料擷取](#)。

### 變更資料擷取 ( CDC )

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以使用 CDC 進行各種用途，例如稽核或複寫目標系統中的變更，以維持同步。

## 混亂工程

故意引入故障或破壞性事件，以測試系統的復原能力。您可以使用 [AWS Fault Injection Service \( AWS FIS \)](#) 執行實驗，以強調 AWS 工作負載並評估其回應。

## CI/CD

請參閱 [持續整合和持續交付](#)。

## 分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

## 用戶端加密

在目標 AWS 服務接收資料之前，在本機加密資料。

## Cloud Center of Excellence ( CCoE )

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端企業策略部落格上的 [CCoE 文章](#)。

## 雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲端運算通常連接到 [邊緣運算](#) 技術。

## 雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱 [建置您的雲端操作模型](#)。

## 採用雲端階段

組織在遷移至時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎：進行基礎投資以擴展雲端採用（例如，建立登陸區域、定義 CCoE、建立操作模型）
- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段由 Stephen Orban 在企業 AWS 雲端 策略部落格的 [The Journey Toward Cloud-First](#) 和 [採用階段](#) 部落格中定義。如需有關它們如何與 AWS 遷移策略關聯的資訊，請參閱 [遷移準備指南](#)。

## CMDB

請參閱 [組態管理資料庫](#)。

## 程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲端儲存庫包括 GitHub 或 Bitbucket Cloud。每個版本的程式碼都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

## 冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

## 冷資料

很少存取且通常為歷史資料的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且成本較低的儲存層或類別，可以降低成本。

## 電腦視覺 (CV)

使用機器學習從數位映像和影片等視覺化格式分析和擷取資訊的 [AI](#) 欄位。例如，AWS Panorama 提供將 CV 新增至內部部署攝影機網路的裝置，而 Amazon 則 SageMaker 提供 CV 的影像處理演算法。

## 組態偏離

對於工作負載，組態會從預期狀態變更。這可能會導致工作負載變得不合規，而且通常是漸進和無意的。

## 組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常會在遷移 CMDB 的產品組合探索和分析階段使用來自的資料。

## 一致性套件

您可以組合的 AWS Config 規則和修復動作集合，以自訂合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶和區域中或整個組織中的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的[一致性套件](#)。

## 持續整合和持續交付 (CI/CD)

自動化軟體發程序的來源、建置、測試、暫存和生產階段的程序。CI/CD is commonly described as a pipeline. CI/CD 可協助您自動化程序、提高生產力、改善程式碼品質，以及更快交付。如需詳細資訊，請參閱[持續交付的優點](#)。CD 也可表示持續部署。如需詳細資訊，請參閱[持續交付與持續部署](#)。

## CV

請參閱[電腦視覺效果](#)。

# D

## 靜態資料

網路中靜止的資料，例如儲存中的資料。

## 資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected Framework 中安全支柱的元件。如需詳細資訊，請參閱[資料分類](#)。

## 資料偏離

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變化。資料偏離可以降低 ML 模型預測的整體品質、準確性和公平性。

## 傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

## 資料網格

架構架構架構，提供分散式、分散式的資料擁有權，並具有集中式管理和治理。

## 資料最小化

僅收集和處理嚴格必要資料的原則。在中實作資料最小化 AWS 雲端可以降低隱私權風險、成本和分析碳足跡。

## 資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有受信任身分才能從預期的網路存取受信任資源。如需詳細資訊，請參閱[在上建立資料周邊 AWS](#)。

## 資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

## 資料來源

追蹤資料整個生命週期的原始伺服器 and 歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

## 資料主體

正在收集和處理資料的個人。

## 資料倉儲

支援商業智慧的資料管理系統，例如分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

## 資料庫定義語言 ( DDL )

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

## 資料庫操作語言 ( DML )

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

## DDL

請參閱[資料庫定義語言](#)。

## 深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

## 深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

## defense-in-depth

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在 上採用此策略時 AWS，您可以在 AWS

Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，方法 defense-in-depth 可能會結合多重要素驗證、網路分割和加密。

## 委派的管理員

在中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶，以管理組織的帳戶並管理該服務的許可。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的 [可搭配 AWS Organizations 運作的服務](#)。

## 部署

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

## 開發環境

請參閱 [環境](#)。

## 偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS 上實作安全控制中的 [偵測性控制](#)。

## 開發值串流映射 ( DVSM )

用於識別和排定限制的流程，這些限制會對軟體開發生命週期中的速度和品質產生不利影響。DVSM 延伸了最初為精實生產實務設計的價值串流映射程序。它專注於透過軟體開發程序建立和移動價值所需的步驟和團隊。

## 數位分身

真實世界系統的虛擬表示法，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

## 維度資料表

在 [星狀結構描述](#) 中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常是文字欄位或離散數字，其行為與文字類似。這些屬性通常用於查詢限制、篩選和結果集標籤。

## 災難

阻止工作負載或系統在其主要部署位置實現其業務目標的事件。這些事件可能是自然災難、技術故障或人為動作的結果，例如意外的組態設定錯誤或惡意軟體攻擊。

## 災難復原 ( DR )

您用來將 [災難造成的停機時間和資料遺失降至最低的策略和程序](#)。如需詳細資訊，請參閱 AWS Well-Architected Framework [中的雲端中工作負載的災難復原 AWS：復原](#)。

## DML

請參閱[資料庫操作語言](#)。

### 領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何搭配 strangler fig 模式使用網域驅動設計的資訊，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \( ASMX \) Web 服務](#)。

## DR

請參閱[災難復原](#)。

### 漂移偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來[偵測系統資源中的漂移](#)，也可以使用 AWS Control Tower 來[偵測登陸區域中可能影響治理要求合規性的變更](#)。

## DVSM

請參閱[開發值串流映射](#)。

## E

### EDA

請參閱[探索性資料分析](#)。

### 邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與[雲端運算](#)相比，邊緣運算可以減少通訊延遲並縮短回應時間。

### 加密

將純文字資料轉換為人類可讀取的運算程序。

### 加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

## 端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

## 端點

請參閱[服務端點](#)。

## 端點服務

您可以在虛擬私有雲端（VPC）中託管以與其他使用者共用的服務。您可以使用 建立端點服務，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management（IAM）主體。這些帳戶或主體可以透過建立介面端點，私下連線至您的VPC端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud（AmazonVPC）文件中的[建立端點服務](#)。

## 企業資源規劃（ERP）

可自動化和**管理企業關鍵業務流程**（例如會計[MES](#)、和專案管理）的系統。

## 信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service（AWS KMS）文件中的[信封加密](#)。

## 環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

## epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如，AWS CAF安全特徵包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱[計畫實作指南](#)。

## ERP

請參閱[企業資源規劃](#)。



## 探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。EDA 透過計算摘要統計資料和建立資料視覺化來執行。

## F

### 事實資料表

[星狀結構描述](#) 中的中央資料表。它存放有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含量值的資料，以及包含維度資料表外部索引鍵的資料。

### 快速失敗

使用頻繁且增量測試來減少開發生命週期的哲學。這是敏捷方法的關鍵部分。

### 故障隔離界限

在中 AWS 雲端，邊界，例如可用區域 AWS 區域、控制平面或資料平面，這些邊界會限制故障的影響，並有助於改善工作負載的復原能力。如需詳細資訊，請參閱[AWS 故障隔離界限](#)。

### 功能分支

請參閱[分支](#)。

### 特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

### 功能重要性

特徵對於模型的預測有多重要。這通常表示為數值分數，可透過各種技術計算，例如 Shapley 累加解釋 (SHAP) 和整合式漸層。如需詳細資訊，請參閱[使用的機器學習模型可解譯性：AWS](#)。

### 特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

## FGAC

請參閱[精細存取控制](#)。

### 精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。



## 閃切遷移

一種資料庫遷移方法，透過[變更資料擷取](#)使用連續資料複寫，以盡可能在最短時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

## G

### 地理封鎖

請參閱[地理限制](#)。

#### 地理限制 (地理封鎖)

在 Amazon 中 CloudFront，此選項可防止特定國家/地區的使用者存取內容分發。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的[限制內容的地理分佈](#)。

### Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被視為舊版，而以[中繼線為基礎的工作流程](#)是現代的首選方法。

### 綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為[棕地](#)。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

### 防護機制

高階規則，可協助管理跨組織單位 ( ) 的資源、政策和合規性OUs。預防性防護機制會強制執行政策，以確保符合合規標準。其實作方式是使用服務控制政策和IAM許可界限。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。其實作方式是使用 AWS Config、AWS Security Hub、Amazon GuardDuty、AWS Trusted Advisor、Amazon Inspector 和自訂 AWS Lambda 檢查。

## H

### HA

請參閱[高可用性](#)。

## 異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如, Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分, 而轉換結構描述可能是一項複雜任務。[AWS 提供有助於結構描述轉換的 AWS SCT](#)。

## 高可用性 ( HA )

工作負載在遇到挑戰或災難時持續運作的能力, 無需介入。HA 系統的設計目的是自動容錯移轉、持續提供高品質效能, 以及處理不同的負載和故障, 並將效能影響降至最低。

## 歷史現代化

一種用於現代化和升級操作技術 ( OT ) 系統的方法, 以更好地滿足製造業的需求。歷史記錄器是一種資料庫類型, 用於從工廠的不同來源收集和存放資料。

## 異質資料庫遷移

將來源資料庫遷移至共用相同資料庫引擎的目標資料庫 ( 例如 Microsoft SQL Server 遷移至 Amazon RDS for SQL Server )。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

## 常用資料

經常存取的資料, 例如即時資料或最近的轉譯資料。此資料通常需要高效能儲存層或類別, 才能提供快速的查詢回應。

## 修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性, 修正程式通常在典型 DevOps 的發行工作流程之外建立。

## 超級護理期間

在切換後, 遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常, 此期間的長度為 1-4 天。在超級護理期間結束時, 遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

## laC

將[基礎設施視為程式碼](#)。

## 身分型政策

連接至一或多個IAM主體的政策, 其定義其在 AWS 雲端 環境中的許可。

## 閒置應用程式

在 90 天內，平均 CPU 和記憶體用量介於 5% 到 20% 的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

## IIoT

請參閱 [工業物聯網](#)。

## 不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有基礎設施。與可變基礎設施相比，不可避免的 [基礎設施](#) 本質上更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的 [使用不可變基礎設施部署](#) 最佳實務。

## 傳入（傳入）VPC

在 AWS 多帳戶架構中，VPC 接受、檢查和路由來自應用程式外部的網路連線。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

## 增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

## 工業 4.0

由 [Klaus Schwab](#) 於 2016 年推出的術語，指透過連線能力、即時資料、自動化、分析和 AI/ML 的進步來現代化製造程序。

## 基礎設施

應用程式環境中包含的所有資源和資產。

## 基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

## 工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱 [建置工業物聯網 \(IIoT\) 數位轉型策略](#)。

## 檢查 VPC

在 AWS 多帳戶架構中，集中 VPC 管理 VPCs（在相同或不同的 AWS 區域）、網際網路和內部部署網路之間的網路流量檢查。[AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

## 物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱[什麼是 IoT？](#)

## 可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱[使用的機器學習模型可解釋性 AWS](#)。

## IoT

請參閱[物聯網](#)。

## IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 提供的基礎 ITSM。

## IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需整合雲端操作與 ITSM 工具的相關資訊，請參閱[操作整合指南](#)。

## ITIL

請參閱[IT 資訊庫](#)。

## ITSM

請參閱[IT 服務管理](#)。

## L

## 標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會被明確指派安全標籤值。使用者安全標籤與資料安全標籤之間的交集決定使用者可以看到哪些資料列和資料欄。

## 登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱[設定安全且可擴展的多帳戶 AWS 環境](#)。

## 大型遷移

遷移 300 部或更多伺服器。

## LBAC

請參閱[標籤型存取控制](#)。

## 最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的[套用最低權限許可](#)。

## 隨即轉移

請參閱[7 Rs](#)。

## 小端序系統

首先儲存最低有效位元組的系統。另請參閱[永久性](#)。

## 較低的環境

請參閱[環境](#)。

# M

## 機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱[機器學習](#)。

## 主要分支

請參閱[分支](#)。

## 惡意軟體

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬程式、間諜軟體和鍵盤側錄程式。

## 受管服務

AWS 服務可 AWS 操作基礎設施層、作業系統和平台，而且您可以存取端點來存放和擷取資料。Amazon Simple Storage Service ( Amazon S3 ) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

## 製造執行系統 ( MES )

用於追蹤、監控、記錄和控制生產程序的軟體系統，可將原物料轉換為工廠的成品。

## MAP

請參閱[遷移加速程式](#)。

## 機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在運作時強化和改善本身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[建置機制](#)。

## 成員帳戶

除了屬於中組織一部分的管理帳戶 AWS 帳戶之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

## MES

請參閱[製造執行系統](#)。

## 訊息佇列遙測傳輸 ( MQTT )

根據[發佈/訂閱](#)模式的輕量型 machine-to-machine ( M2M ) 通訊協定，適用於資源受限的 [IoT](#) 裝置。

## 微服務

小型、獨立的服務，透過定義明確的方式進行通訊，APIs通常由小型、獨立的團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱[使用無 AWS 伺服器服務整合微服務](#)。

## 微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務會使用輕量型，透過定義明確的介面進行通訊APIs。此架構中的每個微服務都可以進行更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱[在上實作微服務 AWS](#)。

## Migration Acceleration Program ( MAP )

提供諮詢支援、訓練和服務，以協助組織建立強大的操作基礎以遷移至雲端，並協助抵銷遷移初始成本的 AWS 計畫。MAP 包含以系統化方式執行舊版遷移的遷移方法，以及一組可自動化和加速常見遷移案例的工具。

### 大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是 [AWS 遷移策略](#) 的第三階段。

### 遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。遷移工廠團隊通常包括操作、業務分析師和擁有者、遷移工程師、開發人員和在衝刺中工作的 DevOps 專業人員。20% 至 50% 之間的企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的 [遷移工廠的討論](#) 和 [雲端遷移工廠指南](#)。

### 遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。遷移中繼資料的範例包括目標子網路、安全群組和 AWS 帳戶。

### 遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：EC2 使用 AWS Application Migration Service 重新託管遷移至 Amazon。

### 遷移產品組合評估 ( MPA )

線上工具，提供驗證商業案例以遷移至的資訊 AWS 雲端。MPA 提供詳細的產品組合評估（伺服器大小調整、定價、TCO 比較、遷移成本分析）以及遷移規劃（應用程式資料分析和資料收集、應用程式分組、遷移優先順序和波規劃）。此 [MPA 工具](#)（需要登入）可供所有 AWS 顧問和 APN 合作夥伴顧問免費使用。

### 遷移就緒狀態評估 ( MRA )

使用取得組織雲端整備狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序 AWS CAF。如需詳細資訊，請參閱 [遷移準備程度指南](#)。MRA 是 [AWS 遷移策略](#) 的第一個階段。

### 遷移策略

用於將工作負載遷移至的方法 AWS 雲端。如需詳細資訊，請參閱本詞彙表中的 [7 個 Rs](#) 項目，並請參閱將 [組織動員以加速大規模遷移](#)。



## 機器學習 (ML)

請參閱[機器學習](#)。

## 現代化

將過時的 (舊版或單一) 應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱[中的應用程式現代化策略 AWS 雲端](#)。

## 現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱[中的評估應用程式的現代化準備 AWS 雲端](#)程度。

## 單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱[將單一體系分解為微服務](#)。

## MPA

請參閱[遷移產品組合評估](#)。

## MQTT

請參閱[訊息佇列遙測傳輸](#)。

## 多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」或者「這個客戶對哪種產品類別最感興趣？」

## 可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性，AWS Well-Architected Framework 建議使用[不可變的基礎設施](#)作為最佳實務。



## O

### OAC

請參閱[原始存取控制](#)。

### OAI

請參閱[原始存取身分](#)。

### OCM

請參閱[組織變更管理](#)。

### 離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

### OI

請參閱[操作整合](#)。

### OLA

請參閱[操作層級協議](#)。

### 線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

### OPC-UA

請參閱[開啟程序通訊 - Unified Architecture](#)。

### 開放程序通訊 - Unified Architecture ( OPC-UA )

工業自動化的 machine-to-machine ( M2M ) 通訊協定。OPC-UA 提供與資料加密、身分驗證和授權方案的互通性標準。

### 操作層級協議 ( OLA )

闡明哪些功能性 IT 群組承諾相互交付的協議，以支援服務層級協議 ( SLA )。

### 操作預備檢閱 ( ORR )

問題及相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[操作整備檢閱 \( ORR \)](#)。

## 操作技術 ( OT )

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造中，整合 OT 和資訊技術 ( IT ) 系統是 [Industry 4.0](#) 轉型的主要重點。

## 操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱[操作整合指南](#)。

## 組織追蹤

由建立的追蹤 AWS CloudTrail 會記錄 AWS 帳戶中組織中所有的事件 AWS Organizations。在屬於組織的每個 AWS 帳戶中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱文件中的 CloudTrail[為組織建立追蹤](#)。

## 組織變更管理 ( OCM )

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變革採用、解決轉型問題，以及推動文化和組織變革，協助組織準備和轉換新系統和策略。在 AWS 遷移策略中，此架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱[OCM指南](#)。

## 原始存取控制 ( OAC )

在中 CloudFront，用於限制存取以保護您的 Amazon Simple Storage Service ( Amazon S3 ) 內容的增強型選項。OAC 支援所有中的所有 S3 儲存貯體 AWS 區域，伺服器端加密搭配 AWS KMS ( SSE-KMS )，以及對 S3 儲存貯體的動態PUT和DELETE請求。

## 原始存取身分 ( OAI )

在中 CloudFront，此選項用於限制存取，以保護您的 Amazon S3 內容。當您使用時OAI，會 CloudFront 建立 Amazon S3 可以驗證的主體。已驗證的主體只能透過特定 CloudFront 分發存取 S3 儲存貯體中的內容。另請參閱[OAC](#)，它提供更精細和增強的存取控制。

## ORR

請參閱[操作預備檢閱](#)。

## OT

請參閱[操作技術](#)。

## 傳出（輸出）VPC

在 AWS 多帳戶架構中，VPC 處理從應用程式內啟動之網路連線的。 [AWS Security Reference Architecture](#) 建議設定具有傳入、傳出和檢查的網路帳戶 VPCs，以保護應用程式與更廣泛的網際網路之間的雙向介面。

## P

### 許可界限

連接至 IAM 主體的 IAM 管理政策，以設定使用者或角色可擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的 [許可界限](#)。

### 個人身分資訊（PII）

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。的範例 PII 包括名稱、地址和聯絡資訊。

### PII

請參閱 [個人識別資訊](#)。

### 手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

### PLC

請參閱 [可程式邏輯控制器](#)。

### PLM

請參閱 [產品生命週期管理](#)。

### 政策

可以定義許可（請參閱 [身分型政策](#)）、指定存取條件（請參閱 [資源型政策](#)）或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱 [服務控制政策](#)）。

### 混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則

可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱[在微服務中啟用資料持久性](#)。

## 組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱[評估遷移準備程度](#)。

## 述詞

傳回 true 或的查詢條件 false，通常位於 WHERE 子句中。

## 述詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這可減少必須從關聯式資料庫擷取和處理的資料量，並改善查詢效能。

## 預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[預防性控制](#)。

## 委託人

中可執行動作和存取資源 AWS 的實體。此實體通常是 AWS 帳戶、IAM 角色或使用者的根使用者。如需詳細資訊，請參閱 IAM 文件中[角色術語和概念](#)中的主體。

## 設計隱私

一種系統工程方法，在整個工程過程中將隱私權納入考量。

## 私有託管區域

容器，其中包含您希望 Amazon Route 53 如何回應一個或多個內網域及其子網域的 DNS 查詢的資訊 VPCs。如需詳細資訊，請參閱 Route 53 文件中的[使用私有託管區域](#)。

## 主動控制

旨在防止部署不合規資源的[安全控制](#)。這些控制項會在佈建資源之前對其進行掃描。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的[控制項參考指南](#)，並請參閱在上實作安全[控制項中的主動](#)控制項。 AWS

## 產品生命週期管理 (PLM)

產品整個生命週期的資料和程序管理，從設計、開發和啟動，到成長和成熟，再到拒絕和移除。

## 生產環境

請參閱[環境](#)。

## 可程式設計邏輯控制器 ( PLC )

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

## 擬匿名化

將資料集中的個人識別碼取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

## publish/subscribe (pub/sub)

一種模式，可在微服務之間啟用非同步通訊，以提高可擴展性和回應能力。例如，在微服務型中[MES](#)，微服務可以將事件訊息發佈到其他微服務可以訂閱的頻道。系統可以新增新的微服務，而無需變更發佈服務。

## Q

### 查詢計劃

一系列步驟，如指示，用於存取SQL關聯式資料庫系統中的資料。

### 查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

## R

### RACI 矩陣

請參閱[負責、負責、已諮詢、知情 \( RACI \)](#)。

### 勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

### RASCI 矩陣

請參閱[負責、負責、已諮詢、知情 \( RACI \)](#)。

### RCAC

請參閱[資料列和資料欄存取控制](#)。

## 僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

## 重新架構師

請參閱 [7 Rs](#)。

## 復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

## 復原時間目標 (RTO)

服務中斷和服務還原之間的可接受延遲上限。

## 重構

請參閱 [7 Rs](#)。

## 區域

地理區域 AWS 的資源集合。每個 AWS 區域 都獨立於其他，以提供容錯能力、穩定性和彈性。如需詳細資訊，請參閱 [指定 AWS 區域 哪些帳戶可以使用](#)。

## 迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

## 重新託管

請參閱 [7 Rs](#)。

## 版本

在部署程序中，它是將變更提升至生產環境的動作。

## 重新定位

請參閱 [7 Rs](#)。

## 轉譯表

請參閱 [7 Rs](#)。

## 回購

請參閱 [7 Rs](#)。

## 彈性

應用程式抵抗中斷或從中斷中復原的能力。[在中規劃恢復能力時，高可用性和災難復原](#)是常見的考量事項 AWS 雲端。如需詳細資訊，請參閱[AWS 雲端 復原能力](#)。

## 資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

## 負責任、負責、已諮詢、知情（RACI）矩陣

定義所有涉及遷移活動和雲端操作之各方的角色和責任的矩陣。矩陣名稱衍生自矩陣中定義的責任類型：負責人（R）、責任（A）、已諮詢（C）和知情（I）。支援（S）類型為選用。如果您包含支援，則矩陣稱為RASCI矩陣，如果您排除它，則稱為RACI矩陣。

## 回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[回應性控制](#)。

## 保留

請參閱 [7 Rs](#)。

## 淘汰

請參閱 [7 Rs](#)。

## 輪換

定期更新[秘密](#)的程序，讓攻擊者更難存取憑證。

## 資料列和資料欄存取控制（RCAC）

使用已定義存取規則的基本靈活SQL表達式。RCAC 包含資料列許可和資料欄遮罩。

## RPO

請參閱[復原點目標](#)。

## RTO

請參閱[復原時間目標](#)。

## 執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

# S

## SAML 2.0

許多身分提供者（IdPs）使用的開放標準。此功能會啟用聯合單一登入（SSO），因此使用者可以登入 AWS Management Console 或呼叫操作，AWS API 而不必 IAM 為您組織中的每個人建立使用者。如需 SAML 2.0 型聯合的詳細資訊，請參閱 IAM 文件中的[關於 SAML 2.0 型聯合](#)。

## SCADA

請參閱 [監控控制和資料擷取](#)。

## SCP

請參閱 [服務控制政策](#)。

## 秘密

在 AWS Secrets Manager 中，以加密形式存放的機密或限制資訊，例如密碼或使用者憑證。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 [Secrets Manager 文件中的 Secrets Manager 秘密中的內容？](#)。

## 安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全控制有四種主要類型：[預防性](#)、[偵測性](#)、[回應性](#) 和 [主動](#)。

## 安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

## 安全資訊和事件管理（SIEM）系統

結合安全資訊管理（SIM）和安全事件管理（SEM）系統的工具和服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生警示。

## 安全回應自動化

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為[偵測](#)或[回應](#)式安全控制，協助您實作 AWS 安全最佳實務。自動化回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換憑證。

## 伺服器端加密

由接收資料的 AWS 服務 加密其目的地的資料。



## 服務控制政策 (SCP)

為 AWS Organizations 中的組織的所有帳戶提供集中控制許可的政策。SCPs 定義管理員可委派給使用者或角色之動作的防護機制或設定限制。您可以使用 SCPs 做為允許清單或拒絕清單，以指定允許或禁止的服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的[服務控制政策](#)。

## 服務端點

URL 的進入點 AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 [AWS 服務端點](#)。

## 服務層級協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

## 服務層級指標 (SLI)

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

## 服務層級目標 (SLO)

代表服務運作狀態的目標指標，由[服務層級指標](#)測量。

## 共同責任模式

描述您與共同 AWS 承擔雲端安全與合規責任的模型。AWS 負責雲端的安全，而負責雲端的安全。如需詳細資訊，請參閱[共同責任模式](#)。

## SIEM

請參閱[安全資訊和事件管理系統](#)。

## 單一失敗點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

## SLA

請參閱[服務層級協議](#)。

## SLI

請參閱[服務層級指示器](#)。

## SLO

請參閱[服務層級目標](#)。

## split-and-seed 模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱[中的階段式應用程式現代化方法 AWS 雲端](#)。

## SPOF

請參閱[單一失敗點](#)。

## 星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構專為[資料倉儲](#)或商業智慧用途而設計。

## Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由 [Martin Fowler 引入](#)，作為重寫單一系統時管理風險的方式。如需如何套用此模式的範例，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \( ASMX \) Web 服務](#)。

## 子網

中 IP 地址的範圍 VPC。子網必須位於單一可用區域。

## 監控控制和資料擷取 ( SCADA )

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

## 對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

## 合成測試

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 [Amazon CloudWatch Synthetics](#) 來建立這些測試。

# T

## 標籤

作為中繼資料的鍵值對，用於組織您的 AWS 資源。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱[標記您的 AWS 資源](#)。

## 目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

## 任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

## 測試環境

請參閱[環境](#)。

## 訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

## 傳輸閘道

可用來互連 VPCs 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的[什麼是傳輸閘道](#)。

## 主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

## 受信任的存取權

將許可授予您指定的服務，以代表您在組織中執行任務 AWS Organizations，並在其帳戶中執行任務。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱文件中的 AWS Organizations [AWS Organizations 搭配使用其他 AWS 服務](#)。

## 調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

## 雙比薩團隊

一個小型 DevOps 團隊，您可以使用兩個披薩來饋送。雙披薩團隊規模可確保軟體開發中的最佳協作。

## U

### 不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱[量化深度學習系統的不確定性](#)指南。

### 未區分的任務

也稱為繁重型，是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、維護和容量規劃。

### 較高的環境

請參閱[環境](#)。

## V

### 清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

### 版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

### VPC 對等

兩個之間的連線VPCs，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱[Amazon 文件中的VPC互連內容](#)。VPC

### 漏洞

損害系統安全性的軟體或硬體缺陷。

## W

### 暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

## 暖資料

不常存取的資料。查詢這類資料時，通常可接受中等慢的查詢。

## 視窗函數

對以某種方式與目前記錄相關聯之資料列群組執行計算的SQL函數。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

## 工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

## 工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器 and 應用程式。

## WORM

請參閱[寫入一次，讀取許多](#)。

## WQF

請參閱[AWS工作負載資格架構](#)。

## 寫入一次，讀取許多 (WORM)

一次性寫入資料的儲存模型，可防止刪除或修改資料。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為[不可變的](#)。

## Z

### 零時差漏洞

利用[零時差漏洞](#)的攻擊，通常是惡意軟體。

### 零時差漏洞

生產系統中未緩解的缺陷或漏洞。威脅發動者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

### 殭屍應用程式

平均CPU和記憶體用量低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。