



選擇用於主機轉換資料庫的遷移工具

# AWS 規範指引



## AWS 規範指引: 選擇用於主機轉換資料庫的遷移工具

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
遷移案例 .....	2
遷移波次、伺服器 and 資料庫 .....	3
使用 AWS Application Migration Service 進行遷移 .....	4
優點 .....	4
缺點 .....	4
最適合的案例 .....	5
具有多個遷移波次的大規模遷移 .....	5
單一應用程式遷移 .....	5
使用原生資料庫工具和 AWS DMS 進行遷移 .....	7
優點 .....	7
缺點 .....	8
後續步驟：決策矩陣 .....	9
資源 .....	10
文件歷史記錄 .....	11
詞彙表 .....	12
# .....	12
A .....	12
B .....	15
C .....	16
D .....	19
E .....	22
F .....	24
G .....	25
H .....	26
I .....	27
L .....	29
M .....	30
O .....	34
P .....	36
Q .....	38
R .....	39
S .....	41
T .....	44

---

U .....	45
V .....	46
W .....	46
Z .....	47
.....	xlviii

# 選擇用於主機轉換資料庫的遷移工具

Mike Kuznetsov 和 Harpreet Virk , Amazon Web Services (AWS)

2022 年 6 月 ([文件歷史記錄](#))

在您規劃將大型工作負載遷移至 AWS 時，我們建議您遵循 [AWS 指引](#) 並將遷移程序分為三個階段：評估、調動和遷移。遷移之旅涉及許多因素，包括範圍（「什麼？」）、策略（「為什麼？」）和時間表（「何時？」），如 [AWS 大型遷移的策略和最佳實務](#) 中討論。您選擇移轉的每個工作負載可能會遵循不同的移轉策略，如七種常見策略 ([7 Rs](#)) 所定義。大多數工作負載都遵循 lift-and-shift 移轉的重新裝載案例。選取策略後，您可以解決「如何？」的問題，此問題至少著重在三個方面 ([人員、技術和程序](#))。

本指南適用於規劃將內部部署工作負載遷移至 AWS 雲端 的任何人，包括 IT 和業務主管、計畫和專案經理、產品擁有者以及營運和基礎設施經理。

本指南重點介紹主機轉換遷移路徑，其中涉及將應用程式遷移至雲端而不進行任何變更以利用雲端功能。例如，在 AWS 雲端 中將內部部署 Microsoft SQL Server 資料庫遷移至 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 SQL Server 是一種主機轉換策略。更具體地說，本指南討論了最適合 lift-and-shift 移轉工作負載 (其範圍包含資料庫) 的工具，以及為移轉選取特定服務時要考慮的因素。它回答了以下問題：哪種服務最支援資料庫遷移？用於非資料庫伺服器的服務是否也可以用於資料庫伺服器，或者應以不同的方式處理資料庫伺服器？如果我的主機轉換遷移變成主機轉換和平台轉換的混合方法怎麼辦？

## 遷移案例

遷移案例可以根據規模 (涉及伺服器或應用程式數量)、遷移時間 (必須以多快的速度完成) 以及切換期間允許的停機時間長度來區分。這些考量事項會影響 AWS 服務的選擇，以最好地支援遷移。

- 大規模遷移 (例如整個資料中心結束) 包括多個業務單位以及許多應用程式及其資料庫。這些通常具有緊迫的時間範圍或固定的截止日期 (例如在資料中心關閉之前完成)，但停機時間限制不太嚴格，從而允許更長的切換時段。
- 業務單位或企業群組的大型但不太複雜的遷移也可能涉及多個應用程式。但是，這些可能具有更靈活的時間表，有時也具有更嚴格的停機時間要求 (更短的切換時段)。
- 單一大型業務關鍵應用程式的遷移通常包括高度密集的資料庫伺服器或叢集，例如處理大量交易的線上交易處理 (OLTP) 資料庫。在切換時段期間，這些遷移通常需要最短或接近零的停機時間。

這些使用案例的遷移要求範圍很廣，從盡可能快的速度 (通常與最長停機時間關聯) 至最大精細程度和對細節的最密切關注 (通常支援遷移期間盡可能短的停機時間)。

## 遷移波次、伺服器 and 資料庫

遷移專案從下列未決問題清單開始，這些問題在[上一節](#)中描述三個案例中很常見：

- 如何建置遷移波次？
- 如何將每個波次內的伺服器分組？
- 資料庫伺服器應在應用程式伺服器之前遷移還是與應用程式伺服器一起遷移？
- 我們應使用哪個工具來進行資料庫遷移？

為了解決這些問題，我們需要釐清一些定義。術語資料庫是否描述執行 DBMS 軟體或邏輯資料庫進入點的伺服器？它是否指相同伺服器或伺服器叢集上的多個資料庫中的一個資料庫？視內容而定，資料庫可能指任一選項。資料庫管理員 (DBA) 通常會考慮邏輯資料庫，而不是實體伺服器。不過，在移轉的環境中，特別是大規模的 lift-and-shift 移轉，資料庫通常會對應於實體伺服器或伺服器叢集。

資料庫的遷移必須與所使用的應用程式保持一致並關聯，因為邏輯資料庫永遠是應用程式及其相依性的一部分。但是，該邏輯資料庫的實體位置可能會有所不同。例如，它可能位於：

- 獨立實體伺服器，不存在任何其他資料庫
- 與其他邏輯資料庫共置的獨立實體伺服器
- 實體伺服器叢集，可以作為單一邏輯資料庫，也可以作為為其他應用程式提供服務的更大資料庫集的一部分

如果應用程式與資料庫伺服器之間沒有明確的相 one-to-one 依性，特別是當來自不同應用程式的多個邏輯資料庫在同一個實體伺服器上並置時，形成移轉波的相依性對應會很複雜。這類資料庫系統需要[重新平台](#)的方法，該方法涉及資料庫的分解或合併，這會使大規模的移轉工作變得複雜。lift-and-shift

這是資料庫遷移工具 (例如資料庫引擎供應商或第三方的原生工具) 可以提供協助的地方。這些工具在邏輯資料庫層級上運作，而不是 AWS Application Migration Service 的區塊層級複寫方法，並且可以在實體伺服器與位置之間的邏輯層級上移動資料或資料庫。

# 使用 AWS Application Migration Service 進行遷移

大多數大型隨即轉移遷移至 AWS 使用 [AWS Application Migration Service](#)。此服務在實體層級上運作，方式是將儲存在任何直接附接的區塊儲存裝置 (例如硬碟或 SAN 磁碟機) 上的資料移至 AWS 上對應的 Amazon Elastic Block Store (Amazon EBS) 儲存裝置。它本質上實作傳統備份/還原程序 (透過複製硬碟)，但也提供近幾秒的復原點目標 (RPO) 和幾分鐘的復原時間點目標 (RTO)，方式是達到來源系統與暫存區域中的目標儲存裝置之間的持續資料保護 (CDP) 同步模式。

## 優點

對於任何規模的隨即轉移遷移，AWS Application Migration Service 有許多好處：

- 複製很容易設定 (不需要陡峭的學習曲線)。
- 透過在來源機器上推出代理程式，可以快速擴展。
- 您可以使用 [雲端遷移工廠](#) 自動執行大多數手動任務、管理多台機器以及協調遷移波次。
- 它支援 [各種 x86 作業系統架構](#)。
- 它支援任何類型的來源環境 (內部部署資料中心、任何其他雲端或其他 AWS 區域)。
- 它與應用程式無關，因此它支援來源伺服器上執行的任何應用程式。
- 無需授權或購買。AWS 提供依用量計費定價，因此您只需在使用此服務時支付費用，而無需長期合約或複雜的授權。AWS Application Migration Service 為每部伺服器提供 90 天的免費期限，這對大多數遷移來說已經足夠。如需詳細資訊，請參閱 AWS 網站上的 [AWS Application Migration Service 定價](#)。

## 缺點

當您使用 AWS Application Migration Service 等區塊層級複寫工具時，可能會遇到下列極端情況和需要考慮的因素：

- 您可以使用相同的工具遷移具有共用區塊儲存的叢集系統，例如 Windows Server 容錯移轉叢集 (WSFC) 或部分 Microsoft SQL Server Always On 可用性群組叢集，但其需要特定的程序和更長的切換時段 (此 [部落格文章](#) 中描述了一些方法)。
- 資料變更率較高的系統 (例如 OLTP 資料庫) 可能具有更高的效能或網路需求，這會讓遷移工作變得複雜。
- 網路頻寬必須足以滿足規劃遷移和切換時段內傳輸的資料量。這是因為 Application Migration Service 不提供離線傳送選項，與 [AWS Snowball](#) 不同。



- 遷移需要較短的停機時段 (RTO 為幾分鐘)。AWS Application Migration Service 使用 EBS 快照作為遷移程序的一部分，從此類快照建立的新 EBS 磁碟最初效能較差。對於部分資料庫讀取模式，這可能會增加有效切換時段，直到遷移的資料庫達到最佳效能。

## 最適合的案例

如前兩部分中所討論，AWS Application Migration Service 完全涵蓋了幾乎所有隨即轉移遷移，幾乎沒有缺點。此工具可以處理大多數極端情況 (例如資料庫叢集)，只要這些系統所需的較長切換時段符合您的遷移要求。

下列各部分更深入地介紹了兩種案例：

- 具有多個遷移波次的大規模遷移
- 需要最短停機時間的單一應用程式遷移

### 具有多個遷移波次的大規模遷移

大規模遷移的範例是資料中心結束，其特點是規模和速度要求，通常涉及特定的時間限制 (例如終止資料中心的租賃合約)。在這種情況下，規模決定方法。遷移波次由應用程式 (包括資料庫) 確定和分組，每個波次都具有規劃的準備、遷移和最終切換階段以及特定的停機時間要求。

在每個遷移波次內，最好使用 AWS Application Migration Service 區塊層級複寫來大規模移動資料庫伺服器，但下列可能需要單獨遷移方法的特殊情況除外：

- 大多數叢集資料庫系統
- 變更率超過可用網路輸送量的資料庫系統 (這可能會導致複寫延遲)

對於每種特殊情況，請考慮您的停機時間要求和使用其他遷移工具所涉及的工作量之間的權衡。在某些情況下，對所有系統使用相同的遷移方法比使用個別工具並為特定系統建立不同的程序要容易得多。如果 AWS Application Migration Service 不支援特定系統的停機時間要求，我們建議您改用 [使用原生資料庫工具和 AWS DMS 進行遷移](#) 部分中所述的方法之一。

### 單一應用程式遷移

單一應用程式案例代表了一種用於遷移單一業務關鍵應用程式的精細方法，此應用程式在遷移期間需要最短或接近零的停機時間，或者是一些此類應用程式。遷移的範圍可能會有所不同，具體取決於業務關鍵性和停機時間要求，而不是先前 (大規模遷移) 案例的速度和規模要求。如果應用程式允許在 AWS

Application Migration Service 的 RTO 內停機，則應採用與前面所述的任何大規模遷移相同的方式進行處理。

但是，在切換時間必須盡可能接近最小值的情況下 (例如，當遷移的資料庫具有需要很長時間才能以最佳效能操作且切換時段必須保持較小的讀取模式時)，您應考慮更詳細、更精細的方法。一般來說，該方法涉及額外的步驟和要求，需要更多的精力和時間。其中一個步驟是將資料庫遷移與應用程式伺服器遷移分開，並使用[下一節](#)中描述的資料庫遷移工具來保持來源資料庫和目標資料庫同步。您可以使用各種方法來實現同步。每種方法都有自己的優點和缺點，並且涉及停機時間量與同步複雜性之間的不同權衡。不過，保持來源環境與目標環境之間的資料庫同步可讓您取消資料庫層與應用程式層的連結。假設應用程式伺服器不在本機儲存持久性資料，而是將所有內容傳遞至資料庫層，同步也會消除應用程式層的停機時間限制。

解耦資料庫和應用程式層可讓您如上一個案例一樣使用 AWS Application Migration Service 來遷移應用程式伺服器。目標應用程式伺服器可以在目標環境中啟動，而來源系統仍在工作，處理資料並將其儲存在資料庫層中。由於資料庫層已經與目標同步，因此切換時間非常短 – 它僅涉及切換網路並將客戶請求從舊來源系統重新導向至目標環境。您可以使用多種方法來執行此操作：遵循[藍/綠部署](#)的指引，通常透過切換 DNS 端點、使用加權 DNS 區域、使用 [AWS Global Accelerator](#) 減少存留時間 (TTL) DNS 傳播延遲以及類似方法。

# 使用原生資料庫工具和 AWS DMS 進行遷移

許多 DBA 熟悉處理資料庫遷移和複寫的各種工具。這些工具通常由資料庫引擎供應商和第三方公司提供，其在特定資料庫引擎的邏輯層級上運作，這與 AWS Application Migration Service 提供的完全與應用程式無關的區塊層級複寫方法不同。

以下是這些工具的清單，範圍從最簡單至更複雜的方法：

- 完整備份/還原對於 IT 人員來說是一個熟悉、知名且易於使用的程序。方法取決於資料庫引擎的類型。此程序通常會移動共置在相同資料庫伺服器上的多個邏輯資料庫，也可用於將資料庫還原到受管服務中，例如 Amazon Relational Database Service (Amazon RDS)。備份/還原是最簡單的方法，但由於備份的大小以及在目標資料庫上建立、複製和還原其所需的時間，與其他選項相比需要更長的切換時段。如需有關此方法的詳細資訊，請參閱 AWS 方案指引網站上的 [原生 SQL Server 備份/還原](#) 和 [Oracle RMAN](#)。
- 邏輯備份或匯出是另一種取得完整或部分邏輯資料庫複本的方法。此原生資料庫引擎工具可讓您分解大型資料庫伺服器以遷移與特定應用程式關聯的選定資料庫。它對遷移內容提供了比完整備份/還原更多的控制，還支援將 Amazon RDS 作為目標。但是，出於與上一方法相同的原因，此選項也需要更長的切換時段。
- 原生資料庫高可用性 (HA) 工具包括 Microsoft SQL Server 和 Oracle 的 Data Guard 複寫中的 Always On 或分散式可用性群組叢集。此方法需要花費大量精力來跨延伸的、跨網站 HA 叢集進行設定，且可能會導致部分效能降級，因為實現完全同步主動/主動部署的延遲較長。但是，此方法在切換期間提供最接近零的停機時間。
- [AWS Database Migration Service](#) (AWS DMS) 和原生資料庫複寫工具 (例如 Oracle GoldenGate、Qlik 和 Talend) 支援變更資料擷取 (CDC) 複寫。您可以使用這些工具複製部分或完整資料庫，並具有接近零停機時間的優勢，因為其會讓目標資料庫與來源資料庫保持同步。您也可以將此方法與 [AWS Schema Conversion Tool](#) (AWS SCT) 和 AWS DMS 搭配使用進行異質遷移，同時遷移和現代化資料庫。
- 如果網路輸送量是資料庫遷移期間的瓶頸，您可以將 AWS DMS 與 [AWS Snowball](#) 結合使用來遷移和現代化非常大的資料庫。如需詳細資訊，請參閱 [新的 AWS DMS 和 AWS Snowball 整合允許大量資料庫遷移和大型資料庫遷移](#) 部落格文章。

## 優點

與區塊層級複寫方法相比，使用資料庫工具進行遷移具有下列優點：

- 部分工具提供了停機時間最少的遷移。其中包括支援原生 HA 叢集或 CDC 複寫的 AWS DMS 和原生工具。
- 您可以使用大多數 DBA 熟悉的工具來遷移叢集資料庫。
- 您可以將資料庫現代化作為遷移工作流程的一部分，並移到 Amazon RDS 或 Amazon Aurora 等受管資料庫服務。
- 從單體式基礎設施移至微服務、分割大型資料庫伺服器或叢集、或將較小的資料庫合併到較大的執行個體或 AWS 服務時，您可以利用合併和分解 (或部分資料庫遷移)。

## 缺點

上一節中討論的大多數好處都不屬於典型隨即轉移遷移方案，而是屬於平台轉換方法。此外，原生資料庫遷移方法在大規模遷移中也存在一些缺點，例如：

- 準備 – 您必須先完全預先佈建和設定目標基礎設施、資料庫伺服器和叢集，然後才能使用任何原生資料庫方法。
- 複雜性 – 某些方法 (例如完整備份或邏輯備份/還原) 必須與其他複寫方法結合使用，才能偵測自建立初始備份以來的所有變更。
- 可擴展性 – 當您大規模遷移時，沒有簡單的自動化架構可以將這些方法推廣到其他資料庫叢集和伺服器。

## 後續步驟：決策矩陣

雖然每次遷移都是唯一的，且具有自己的挑戰、限制和要考慮的多種因素，但您可以使用一些通用標準來識別最適合您的使用案例的遷移策略和服務。識別這些因素並排定其優先順序可協助您縮小選擇範圍。使用下表作為決策樹：從您的使用案例最重要的因素開始，選擇最適合您遷移的工具。

條件	AWS Application Migration Service	資料庫工具 (原生工具或 AWS DMS)
架構	實體 (區塊層級)	邏輯，資料庫引擎層級
擴展	大規模遷移	精細；擴展限制*
速度和複雜性	快速結束案例，降低複雜性	較慢、更複雜的方法，需要更多的規劃和測試
時間表	支持積極的時間表	需要額外的精力和時間
類型	以現況隨即轉移 (僅一對一)	平台轉換或現代化，具有分解和合併選項 (一對多、多對一)
預先佈建	不需要；自動遷移	需要資料庫和基礎設施佈建
停機	需要停機時間，RTO 在幾分鐘內	可能幾乎不停機，但非常昂貴 (透過同步/非同步延伸叢集、CDC 複寫和類似方法)
資料變更率	可能具有聯網或效能限制	更多可用選項
限制	叢集化系統；非 x86 平台	原生資料庫工具支援叢集資料庫和非 x86 平台；AWS DMS 涵蓋了大多數資料庫引擎

\* [AWS Database Migration Accelerator 程式計畫](#) 匯集了遷移工具和專家，可協助您大規模遷移資料庫。

# 資源

## AWS 服務

- [使用 AWS 進行遷移](#)
- [AWS Application Migration Service](#)
- [AWS Database Migration Service](#)

## 遷移策略

- [AWS 大規模遷移策略和最佳實務](#)
- [調動您的組織以加速大規模遷移](#)

## 資料庫遷移、原生資料庫工具

- [關聯式資料庫的遷移策略](#)
- [將 Microsoft SQL Server 資料庫遷移至 AWS 雲端](#)
- [將 Oracle 資料庫遷移至 AWS 雲端](#)
- [資料庫主機轉換模式](#)

## 部落格文章

- [加速遷移至 AWS](#) (AWS 架構部落格)
- [將應用程式遷移至雲端的 6 個策略](#) (AWS 雲端企業策略部落格)
- [資料庫遷移 – 開始之前您需要了解什麼?](#) (AWS 資料庫部落格)

## 文件歷史記錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

變更	描述	日期
<a href="#">初次出版</a>	—	2022 年 6 月 22 日

# AWS 規範指引詞彙表

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：將內部部署 Oracle 資料庫遷移至 Amazon Aurora PostgreSQL 相容版本。
- 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 資料庫遷移至中的 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 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，但需要比 [主動被動遷移](#) 更多的工作。

## 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

在一組資料列上運作的 SQL 函數，會計算群組的單一傳回值。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。匿名資料不再被視為個人資料。

## 反模式

經常用於重複性問題的解決方案，其解決方案具有反效益、無效或效果不如替代方案。

## 應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體侵害。

## 應用程式組合

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

## 人工智慧 (AI)

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

## 人工智慧操作 (AIOps)

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

## 非對稱加密

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

## 原子、一致性、隔離、耐久性 (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 ([ABAC](#)) 文件中的 [Word for AWS](#)。IAM

## 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。您可以將資料從權威資料來源複製到其他位置，以處理或修改資料，例如匿名化、修訂或擬匿名化資料。

## 可用區域

中與其他可用區域中的故障 AWS 區域 隔離的不同位置，並對相同區域中的其他可用區域提供便宜的低延遲網路連線。

## AWS 雲端採用架構 (AWS CAF)

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 將指引整理成六個重點領域：業務、人員、治理、平台、安全和操作。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。為此，AWS CAF 提供人員開發、訓練和通訊的指引，協助組織準備好成功採用雲端。如需詳細資訊，請參閱[AWS CAF 網站](#)和[AWS CAF 白皮書](#)。

## AWS 工作負載資格架構 (AWS WQF)

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

## B

### 錯誤的機器人

旨在中斷或傷害個人或組織的[機器人](#)。

### BCP

請參閱[業務持續性規劃](#)。

### 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、可疑的 API 呼叫和類似的動作。如需詳細資訊，請參閱偵測文件中的[行為圖中的資料](#)。

### 大端序系統

首先儲存最高有效位元組的系統。另請參閱[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。有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。某些其他機器人，稱為不良機器人，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，稱為機器人繼承者或機器人運算子。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，以測試系統的彈性。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 [建置您的雲端營運模型](#)。

## 採用雲端階段

組織在遷移至 時通常會經歷的四個階段 AWS 雲端：

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

這些階段由 Stephen Orban 在部落格文章 [The Journey Toward Cloud-First](#) 和企業策略部落格上的 [採用階段](#) 中定義。AWS 雲端 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。查詢此類資料時，通常可接受慢查詢。將此資料移至效能較低且價格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 SageMaker 則提供 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。您通常會在遷移的產品組合探索和分析階段使用來自 CMDB 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 帳戶和區域中或跨組織的單一實體。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](#)。

## 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，並具有集中式的管理和治理。

## 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的身分才能從預期的網路存取受信任的資源。如需詳細資訊，請參閱 [在上建置資料周邊 AWS](#)。

## 資料預先處理

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

## 資料來源

在整個生命週期中追蹤資料的原始伺服器 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 AWS 上採用此策略時，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，a defense-in-depth 方法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 AWS Organizations 中，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。它著重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呈現，例如建築、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

### 維度資料表

在 [星狀結構描述](#) 中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常為文字欄位或像文字那樣行為的離散數字。這些屬性通常用於查詢限制、篩選和結果集標籤。

### 災難

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

## 災難復原 (DR)

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

## 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 ASMX Gateway 逐步現代化舊版 Microsoft ASP.NET \(API\) 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偏離，也可以使用 AWS Control Tower 來[偵測可能會影響對治理要求合規性的登陸區域中的變更](#)。 <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html>

## DVSM

請參閱[開發值串流映射](#)。

## E

### EDA

請參閱[探索性資料分析](#)。

### EDI

請參閱[電子資料交換](#)。

## 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。

## 電子資料交換 (EDI)

組織之間的商業文件自動交換。如需詳細資訊，請參閱[什麼是電子資料交換](#)。

## 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 VPC 端點，私下連線至您的端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud (Amazon VPC) 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 是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欄。

## 快速失敗

使用頻繁且增量測試來縮短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彈性。如需詳細資訊，請參閱[AWS 故障隔離界限](#)。

## 功能分支

請參閱[分支](#)。

## 特徵

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

## 功能重要性

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

## 特徵轉換

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

## 幾下提示

在請求 [LLM](#) 執行類似任務之前，提供少量示範任務和所需輸出的範例。此技術是內文學習的應用，其中模型會從內嵌在提示中的範例 (快照) 中學習。對於需要特定格式設定、推理或網域知識的任務，少量擷取提示非常有效。另請參閱[零擷取提示](#)。

## 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。

## FM

請參閱[基礎模型](#)。

### 基礎模型 (FM)

大型深度學習神經網路，已針對廣義和未標記資料的大量資料集進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言進行交談。如需詳細資訊，請參閱[什麼是基礎模型](#)。

## G

### 生成式 AI

經過大量資料訓練的 [AI](#) 模型子集，可以使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱[什麼是生成式 AI](#)。

## 地理封鎖

請參閱[地理限制](#)。

### 地理限制 (地理封鎖)

在 Amazon CloudFront 中，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。

### 金色影像

系統或軟體的快照，用作部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

### 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 (OUs) 的資源、政策和合規性。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。其實作方式包括使用 AWS Config AWS Security Hub、Amazon GuardDuty、AWS Trusted Advisor、Amazon Inspector 和自訂 AWS Lambda 檢查。

## H

### HA

請參閱[高可用性](#)。

### 異質資料庫遷移

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

## 高可用性 (HA)

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

## 歷史現代化

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

## 保留資料

從用於訓練機器學習模型的資料集中保留的歷程記錄、已標記的資料的一部分。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

## 異質資料庫遷移

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

## 常用資料

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

## 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的 DevOps 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。

## I

## IaC

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，可定義其在 AWS 雲端環境中的許可。

## 閒置應用程式

在 90 天內的平均 CPU 和記憶體用量介於 5% 到 20% 之間的應用程式。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的部署](#)最佳實務。

## 傳入（輸入）VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 VPC。[AWS 安全參考架構](#)建議設定具有傳入、傳出和檢查 VPCs 的網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向界面。

## 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 年引進的術語，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，集中式 VPC 可管理 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 安全參考架構](#)建議使用傳入、傳出和檢查 VPCs 設定您的網路帳戶，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[AWS 的機器學習模型可解釋性](#)。

## 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 ITSM 的基礎。

## 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，其中每個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](#)。

## 大型語言模型 (LLM)

預先訓練大量資料的深度學習 [AI](#) 模型。LLM 可以執行多個任務，例如回答問題、彙整文件、將文字翻譯為其他語言，以及完成句子。如需詳細資訊，請參閱[什麼是 LLMs](#)。

## 大型遷移

遷移 300 部或更多伺服器。

## LBAC

請參閱[標籤型存取控制](#)。

## 最低權限

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

## 隨即轉移

請參閱 [7 Rs](#)。

## 小端序系統

首先儲存最低有效位元組的系統。另請參閱[端點](#)。

## LLM

請參閱[大型語言模型](#)。

## 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、間諜軟體和鍵盤記錄器。

## 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[遷移加速計畫](#)。

## 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。機制是可在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[建置機制](#)。

## 成員帳戶

除了屬於組織的管理帳戶 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

小型獨立服務，透過定義明確的 APIs 進行通訊，通常由小型獨立團隊擁有。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 APIs 透過定義明確的界面進行通訊。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營運基礎以遷移至雲端，並協助抵銷遷移的初始成本的 AWS 計畫。MAP 包含以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、開發人員和從事衝刺工作的 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：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

## 遷移產品組合評估 (MPA)

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

## 遷移就緒狀態評估 (MRA)

使用 AWS CAF 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。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[遷移您的組織以加速大規模遷移](#)。

## 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 A machine-to-machine (M2M) 通訊協定。OPC-UA 提供具有資料加密、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 IT 群組承諾交付給彼此的協議，以支援服務層級協議 (SLA)。

## 操作準備檢閱 (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[操作就緒審核 \(ORR\)](#)。

## 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。在製造業中，OT 和資訊技術 (IT) 系統的整合是 [Industry 4.0](#) 轉型的關鍵重點。

## 操作整合 (OI)

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

## 組織追蹤

由建立的追蹤 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有事件 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 CloudTrail 文件中的[為組織建立追蹤](#)。

## 組織變更管理 (OCM)

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

## 原始存取控制 (OAC)

In CloudFront 是用於限制存取以保護 Amazon Simple Storage Service (Amazon S3) 內容的增強型選項。OAC 支援所有 S3 儲存貯體 AWS 區域中的所有伺服器端加密 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)

In CloudFront，用於限制存取以保護您的 Amazon S3 內容的選項。當您使用 OAI 時，CloudFront 會建立 Amazon S3 可以驗證的主體。已驗證的主體只能透過特定 CloudFront 分佈存取 S3 儲存貯體中的內容。另請參閱[OAC](#)，其可提供更精細且增強的存取控制。

## ORR

請參閱[操作準備檢閱](#)。

## OT

請參閱[操作技術](#)。

## 傳出（傳出）VPC

在 AWS 多帳戶架構中，處理從應用程式內起始之網路連線的 VPC。[AWS Security Reference Architecture](#) 建議使用傳入、傳出和檢查 VPCs 設定您的網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向界面。

## P

### 許可界限

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

### 個人身分識別資訊 (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。PII 的範例包括名稱、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[可程式設計邏輯控制器](#)。

### PLM

請參閱[產品生命週期管理](#)。

### 政策

可定義許可（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最大許可的物件 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

透過整個開發程序將隱私權納入考量的系統工程方法。

## 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 DNS 內網域及其子網域的 VPCs 查詢的資訊。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](#)，並參閱在實作安全[控制項中的主動](#)控制項。 AWS

## 產品生命週期管理 (PLM)

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

### 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。

### 提示鏈結

使用一個 [LLM](#) 提示的輸出作為下一個提示的輸入，以產生更好的回應。此技術用於將複雜的任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和關聯性，並允許更精細、更個人化的結果。

### 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。假名化資料仍被視為個人資料。

## publish/subscribe (pub/sub)

一種模式，可啟用微服務之間的非同步通訊，以提高可擴展性和回應能力。例如，在微服務型 [MES](#) 中，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。系統可以在不變更發佈服務的情況下新增新的微服務。

## 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、諮詢、知情 \(RACI\)](#)。

## RAG

請參閱[擷取增強型生成](#)。

## 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 您的帳戶可以使用哪些](#)。

## 迴歸

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

## 重新託管

請參閱 [7 Rs](#)。

## 版本

在部署程序中，它是將變更提升至生產環境的動作。

## 重新定位

請參閱 [7 Rs](#)。

## 轉譯形式

請參閱 [7 Rs](#)。

## 回購

請參閱 [7 Rs](#)。

## 彈性

應用程式抵抗中斷或從中斷中復原的能力。[在中規劃彈性時，高可用性和災難復原](#)是常見的考量因素 AWS 雲端。如需詳細資訊，請參閱[AWS 雲端 彈性](#)。

## 資源型政策

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

## 負責、負責、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：責任 (R)、責任 (A)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除該矩陣，則稱為 RACI 矩陣。

## 回應性控制

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

## 保留

請參閱 [7 Rs](#)。

## 淘汰

請參閱 [7 Rs](#)。

## 擷取增強產生 (RAG)

在產生回應之前，[LLM](#) 參考其訓練資料來源外之權威資料來源的[生成 AI](#) 技術。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱[什麼是 RAG](#)。

## 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、彈性 SQL 表達式。RCAC 由資料列許可和資料欄遮罩組成。

## 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)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而不必在 IAM 中為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 [SAML 文件中的關於 Word 2.0 型聯合](#)。IAM

## SCADA

請參閱[監督控制和資料擷取](#)。

## SCP

請參閱[服務控制政策](#)。

## 秘密

您以加密形式存放的 AWS Secrets Manager 機密或限制資訊，例如密碼或使用憑證。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？](#)。

## 設計安全性

透過整個開發程序將安全性納入考量的系統工程方法。

## 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[主動性](#)。

## 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 會定義管理員可委派給使用者或角色之動作的防護機制或設定限制。您可以使用 SCPs 作為允許清單或拒絕清單，以指定允許或禁止的服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的 [服務控制政策](#)。

## 服務端點

的進入點 URL 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。

## 服務層級指示器 (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)

目標指標，代表服務的運作狀態，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發生故障，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。此結構專為在[資料倉儲](#)中使用或用於商業智慧目的而設計。

## 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 ASP Gateway 逐步現代化舊版 Microsoft ASMX.NET \(API\) Web 服務](#)。

## 子網

VPC 中的 IP 地址範圍。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。

## 系統提示

提供內容、指示或指引給 [LLM](#) 以指示其行為的技術。系統提示可協助設定內容，並建立與使用者互動的規則。

# T

## 標籤

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

## 目標變數

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

## 任務清單

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



## 測試環境

請參閱[環境](#)。

## 訓練

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

## 傳輸閘道

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

## 主幹型工作流程

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

## 受信任的存取權

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

## 調校

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

## 雙比薩團隊

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

# U

## 不確定性

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

## 未區分的任務

也稱為繁重型，是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分的任務範例包括採購、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 VPCs 之間的連線，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 Amazon [VPC 文件中的什麼是 Word 對等](#)。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

SQL 函數，對以某種方式與目前記錄相關聯的資料列群組執行計算。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 Workload Qualification Framework](#)。

## 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](#)。

## 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。威脅實施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。

### 零擷取提示

為 [LLM](#) 提供執行任務的指示，但沒有可協助引導任務的範例 (快照)。LLM 必須使用其預先訓練的知識來處理任務。零擷取提示的有效性取決於任務的複雜性和提示的品質。另請參閱 [微拍提示](#)。

### 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。