套用適用於 Amazon Neptune 的 AWS Well-Architected 架構

AWS 方案指引

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 方案指引: 套用適用於 Amazon Neptune 的 AWS Well-Architected 架構

Copyright © 2025 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
目標對象 1
目標 1
卓越營運 3
使用 IaC 方法自動化部署 3
進行頻繁、小型、可逆的變更 3
預期失敗 4
從所有操作失敗中學習 5
使用記錄功能來監控未經授權的或異常的活動 5
安全 6
實作資料安全 6
保護您的網路 7
實作身分驗證和授權 7
可靠性 9
了解 Neptune 服務配額 9
了解 Neptune 部署模式 10
管理和擴展 Neptune 叢集 10
管理備份和容錯移轉事件 11
效能效率 12
了解圖形建模 12
最佳化查詢 13
正確大小的叢集 14
最佳化寫入 15
成本最佳化 16
了解所需的用量模式和服務 16
選取關注成本的資源 17
為您的工作負載選擇最佳的 Neptune 執行個體組態 18
適當大小的資料儲存和傳輸 19
永續性 20
AWS 區域選擇 20
根據使用者行為模式的使用 20
最佳化軟體開發和架構模式 21
資源 22
貢獻者 23

文件歷史紀錄 24
詞彙表 25
..... 25
A 25
B 28
C 29
D 32
E 35
F 37
G 38
H 39
I 40
L 42
M 43
O 47
P 49
Q 51
R 52
S 54
T 57
U 58
V 59
W 59
Z 60
..... lxi

為 Amazon Neptune 套用 AWS Well-Architected 架構

Amazon Web Services（貢獻者）

2023 年 9 月 (文件歷史記錄)

您可以使用 Amazon Amazon Neptune 在 Amazon Web Services (AWS) 上建置圖形型解決方案。本指南提供在規劃 Neptune 部署時套用 AWS Well-Architected Framework 原則的規範性指導。套用適用於 Amazon Neptune Analytics 的 AWS Well-Architected 架構涵蓋 Neptune 圖形分析引擎的相同主題。

AWS Well-Architected Framework 可協助您為各種應用程式和工作負載建置安全、高效能、彈性且高效率的基礎設施。它也提供一致的方法來評估架構並實作可擴展的設計。

AWS Well-Architected Framework 是以下列六個支柱為基礎：

- 卓越營運
- 安全
- 可靠性
- 效能效率
- 成本最佳化
- 永續性

本指南提供 Well-Architected Framework 設計支柱和最佳實務的資訊，以及部署 Neptune 時應謹記的考量事項 AWS。

目標對象

本指南適用於資料工程師、解決方案架構師和資料分析師，他們設計並實作使用圖形的解決方案 AWS。

目標

本指南可協助您和組織執行下列動作：

- 根據您的使用案例和查詢模式，從支援的部署選項和查詢語言中進行選擇。

- 遵循 AWS Well-Architected 設計模式，以協助改善彈性和安全性。
- 設計您的查詢以獲得最佳效能並節省成本。
- 了解如何在生產環境中管理 Neptune 叢集時達到營運效率。

卓越營運支柱

AWS Well-Architected Framework 的卓越營運支柱著重於執行和監控系統，以及持續改善程序和程序。它包含有效支援開發和執行工作負載、深入了解其操作，以及持續改善支援程序和程序以提供商業價值的能力。您可以透過自我修復工作負載來降低操作複雜性，無需人工介入即可偵測和修復大多數問題。您可以遵循本節中所述的最佳實務來實現此目標。當您的工作負載偏離預期行為時，請使用 Amazon Neptune 指標、API 和機制來正確回應。 APIs

此卓越營運支柱的討論著重於下列關鍵領域：

- 基礎設施即程式碼 (IaC)
- 變更管理
- 彈性策略
- 事件管理
- 合規稽核報告
- 日誌記錄和監控

使用 IaC 方法自動化部署

使用 IaC 在 Neptune 上自動化部署的最佳實務包括：

- 盡可能套用基礎設施即程式碼 (IaC) 來部署 Neptune 叢集。若要取得一致的環境組態，請使用 AWS CloudFormation 範本、 AWS Cloud Development Kit (AWS CDK)或 HashiCorp Terraform 為您的叢集建立所有必要的資源。
- 盡可能自動化 Neptune 操作程序，例如調整執行個體大小、新增或移除僅供讀取複本，或在全域資料表上執行手動容錯移轉。
- 從用戶端將連線字串存放在外部。使用擷取、轉換和載入 (ETL) 程序，以促進藍/綠部署策略、災難復原 (DR) 和近乎零的停機時間遷移到新叢集。連線字串可以存放在 AWS Secrets Manager、Amazon DynamoDB 或任何可以動態變更它們的位置。
- 使用標籤將中繼資料新增至 Neptune 資源，並根據標籤追蹤用量。如需詳細資訊，請參閱標記 Amazon Neptune 資源。

進行頻繁、小型、可逆的變更

下列建議著重於小型、可逆的變更，以將複雜性降至最低，並降低工作負載中斷的可能性：

- 在來源控制服務中存放 IaC 範本和指令碼，例如 GitHub 或 GitLab。

  重要：請勿在來源控制中存放 AWS 登入資料。

- 要求 IaC 部署使用持續整合和持續交付 (CI/CD) 服務，例如 AWS CodeDeploy或 AWS CodeBuild。這些服務會在非生產環境中編譯、測試和部署程式碼，其中包含暫時性 Neptune 叢集，然後再影響您的生產 Amazon Neptune 叢集。
- 在較低的環境中測試基礎設施和應用程式查詢，然後再將其部署到生產環境。這將最大限度地降低中斷的可能性，並有助於確保它們能與您的工作負載和規模一起正常運作。

預期失敗

自我修復基礎設施透過預測故障並嘗試在不介入的情況下解決任何問題，來示範卓越營運。下列建議可協助您使用 Neptune 實現該成熟度：

- 建立監控計畫，使用 Amazon CloudWatch 指標來監控資料庫執行個體的 CPU 和記憶體用量，並了解用量模式。建立 CloudWatch 儀表板和警示，以用於在您應用程式日誌中找到的關鍵指標和 Neptune 用戶端回應。如需高或低 CPU 使用率指標的詳細資訊，請參閱 Neptune 文件中的使用 CloudWatch 在 Neptune 中監控資料庫執行個體效能。

  如果您在查詢`FreeableMemory`中經常遇到out-of-memory例外狀況，則請考慮使用 X2 系列的執行個體。

- 設定通知以監控 Neptune 叢集的運作狀態。例如， `BufferCacheHitRatio` 應該持續處於高（大於 99.9%)，而 `MainRequestQueuePendingRequests`應該持續處於低（理想情況下為 0，但取決於您的需求和延遲容差）。
- 請考慮使用僅供讀取複本，以在 Neptune 中實現高可用性。您應該在與寫入器執行個體不同的可用區域中有至少兩個僅供讀取複本，以確保執行個體在容錯移轉事件期間隨時可用以提供僅供讀取查詢。
- 根據使用率指標自動擴展僅供讀取複本。如需詳細資訊，請參閱自動擴展 Amazon Neptune 資料庫叢集中的複本數量。
- 測試資料庫執行個體的容錯移轉，以了解您的使用案例執行此程序需時多長。
- 如果您的應用程式需要完全中斷 AWS 區域 ，請考慮使用全域資料庫做為 DR 計畫的一部分。

從所有操作失敗中學習

自我修復基礎設施是一種長期工作，會在發生罕見問題或回應效果不如預期時，在反覆運算中發展。採用下列實務可推動專注於該目標：

- 透過從所有失敗中學習來推動改進。
- 跨團隊和組織分享學到的知識。如果組織中的多個團隊使用 Neptune，請建立通用聊天室或使用者群組來分享學習和最佳實務。

使用記錄功能來監控未經授權的或異常的活動

若要觀察異常效能和活動模式，請將日誌存放在 Amazon CloudWatch Logs 中。請考慮下列最佳實務：

- 啟用慢查詢記錄。定期檢閱日誌，並診斷為何特定查詢緩慢。使用 Gremlin、SPARQL 或 openCypher 的 Neptune 解釋和描述檔端點，深入了解為何這些查詢速度緩慢。
- 啟用 Neptune 稽核日誌，並定期檢閱日誌是否有未經授權的存取或異常。
- 如果您使用的是慢查詢記錄或稽核記錄，請啟用發佈至 CloudWatch Logs。這可協助您避免執行個體上的磁碟空間用盡。Neptune 執行個體的日誌儲存容量有限，且會在超過日誌空間時覆寫較舊的日誌檔案。CloudWatch Logs 支援長期保留日誌。CloudWatch Logs 中的增強型監控功能將改善您查詢日誌和診斷問題的能力。
- 若要為您的稽核日誌提供更好的分析工具，您可以設定 Neptune 資料庫叢集，將稽核日誌資料發佈至 CloudWatch Logs 中的日誌群組。使用 CloudWatch Logs，您可以執行日誌資料的即時分析、使用 CloudWatch 來建立警示和檢視指標，並使用 CloudWatch Logs 將您的日誌記錄儲存到高耐用性儲存體。如需詳細資訊，請參閱將 Neptune 日誌發佈至 Amazon CloudWatch Logs。
- Neptune 支援使用 記錄控制平面動作 AWS CloudTrail。如需詳細資訊，請參閱使用 記錄 Amazon Neptune API 呼叫 AWS CloudTrail。

安全支柱

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 和 之間的共同責任。共同責任模型將此描述為雲端本身的安全和雲端內部的安全：

- 雲端的安全性 – AWS 負責保護在 AWS 服務 中執行的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。第三方稽核人員會定期測試和驗證安全性的有效性 AWS ，做為AWS 合規計畫的一部分。若要了解適用於 Amazon Neptune 的合規計劃，請參閱合規計劃範圍內的AWS 服務。
- 雲端安全性 – 您的責任取決於您使用 AWS 服務 的 。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。如需有關資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需歐洲資料保護的相關資訊，請參閱AWS 共同責任模型和 GDPR 部落格文章。

安全支柱可協助您了解如何在使用 Neptune 時套用共同責任模型。下列主題說明如何將 Neptune 設定為實現您的安全及合規目標。您也會了解如何使用其他 AWS 服務 來協助您監控和保護 Neptune 資源。

安全支柱包含下列主要重點區域：

- 資料安全
- 網路安全
- 身分驗證和授權

實作資料安全

資料洩漏和違規會讓您的客戶面臨風險，並可能對您的公司造成重大負面影響。下列最佳實務有助於保護您的客戶資料免於意外和惡意暴露：

- 叢集名稱、標籤、參數群組、 AWS Identity and Access Management (IAM) 角色和其他中繼資料不應包含機密或敏感資訊，因為該資料可能會出現在帳單或診斷日誌中。
- URIs或存放在 Neptune 中的外部伺服器連結不應包含登入資料資訊來驗證請求。
- Neptune 加密的執行個體可以協助保護您的資料免於發生未經授權的基礎儲存體存取，為資料提供另一層保護。您可以使用 Neptune 加密來提高部署在雲端之應用程式的資料保護。您也可以進行 Neptune 加密，以滿足靜態資料的合規要求。

若要啟用新 Neptune 資料庫執行個體的加密，請在 Neptune 主控台的啟用加密區段中選擇是（預設勾選），或設定 AWS::Neptune::DBCluster::StorageEncrypted 屬性 AWS CloudFormation。如果啟用加密，Neptune 預設將使用 Amazon Relational Database Service (Amazon RDS) AWS 受管金鑰，或者您可以建立客戶受管金鑰。如需建立 Neptune 資料庫執行個體的相關資訊，請參閱建立新的 Neptune 資料庫叢集。如需詳細資訊，請參閱加密 Neptune 資源靜態。您的自動化和手動快照會使用您為 Neptune 叢集選取的相同加密。

- 使用 SPARQL 和 openCypher 語言時，請練習適當的輸入驗證和參數化技術，以防止 SQL 注入和其他形式的攻擊。避免使用使用者提供的輸入建構使用字串串連的查詢。使用參數化查詢或準備好的陳述式，安全地將輸入參數傳遞至圖形資料庫。如需詳細資訊，請參閱 openCypher 參數化查詢和 SPARQL Injection Defence 的範例。 https://owasp.org/www-pdf-archive/Onofri-NapolitanoOWASPDayItaly2012.pdf
- 對於 Gremlin 語言，請使用 Gremlin 語言變體，而不是直接傳遞以字串為基礎的 Gremlin 指令碼，以避免潛在的注入問題。

保護您的網路

Amazon Neptune 資料庫叢集只能在虛擬私有雲端 (VPC) 上建立 AWS。Neptune 資料庫叢集的端點只能在該 VPC 內存取，通常是從在該 VPC 中執行的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體存取。您可以透過限制存取 Neptune 資料庫叢集所在的 VPC 來保護 Neptune 資料。如需詳細資訊，請參閱連線至 Amazon Neptune 圖表。

為了保護您的傳輸中資料，Neptune 會使用安全通訊協定和密碼，透過 HTTPS 強制 SSL 連線至任何執行個體或叢集端點。Neptune 為您的 Neptune 資料庫執行個體提供 SSL 憑證。Neptune SSL 憑證僅支援叢集端點、讀取器端點和執行個體端點主機名稱。如果您使用負載平衡器或代理伺服器（例如 HAProxy)，則必須使用 SSL 終止，並在代理伺服器上擁有您自己的 SSL 憑證。SSL 傳遞沒有作用，因為提供的 SSL 憑證與代理伺服器主機名稱不符。如需使用 SSL 連線至 Neptune 端點的詳細資訊，請參閱使用 HTTP REST 端點連線至 Neptune 資料庫執行個體。

實作身分驗證和授權

若要控制誰可以在 Neptune 資料庫叢集和資料庫執行個體上執行 Neptune 管理 動作，請使用 IAM 登入資料。當您 AWS 使用 IAM 登入資料連線至 時，您的 IAM 角色必須具有 IAM 政策，授予執行 Neptune 管理操作所需的許可。確保您遵循最低權限原則，僅授予完成任務所需的許可。如需詳細資訊，請參閱使用不同類型的 IAM 政策來控制 Neptune 存取和使用暫時登入資料的 IAM 身分驗證。

若要控制誰可以連線至 Neptune 叢集並查詢資料，您可以使用 IAM 來驗證您的 Neptune 資料庫執行個體或資料庫叢集。如果您在 Neptune 資料庫叢集中啟用 IAM 身分驗證，則必須先驗證存取資料庫叢集的任何人。如需詳細資訊，請參閱在 Neptune 中啟用 IAM 資料庫身分驗證，以取得啟用 IAM 身分驗證的步驟。

當 IAM 資料庫身分驗證啟用時，每個請求都必須使用 AWS Signature 第 4 版進行簽署。若要了解如何在啟用 IAM 身分驗證的情況下將簽署的請求傳送至所有 Neptune 端點，請參閱使用 AWS 簽章版本 4 連接和簽署。許多程式庫和工具，例如 awscurl，已支援 AWS Signature 第 4 版。

為了與其他 互動 AWS 服務，Amazon Neptune 使用 IAM 服務連結角色。服務連結角色是直接連結至 Neptune 的一種獨特 IAM 角色類型。服務連結角色是由 Neptune 預先定義，並包含服務 AWS 服務 代表您呼叫其他 所需的所有許可。如需詳細資訊，請參閱使用 Neptune 的服務連結角色。

可靠性支柱

可靠性支柱包含工作負載在預期情況下正確且一致地執行其預期函數的能力。包括在整個生命週期中執行及測試工作負載。

可靠的工作負載始於對軟體和基礎設施的前期設計決策。您的架構選擇會對所有 Well-Architected 支柱的工作負載行為產生影響。為求可靠性，您必須依循特定模式。

可靠性支柱著重於下列關鍵領域：

- 工作負載架構，包括服務配額和部署模式
- 變更管理
- 故障管理

了解 Neptune 服務配額

除了中國和 GovCloud AWS 區域 之外，所有支援的 Neptune 叢集磁碟區大小上限為 128 TB (TiB)，其中配額為 64 TiB。

128 TiB 配額足以在圖形中存放約 2000 億至 4000 億個物件。在已標記的屬性圖形 (LPG) 中，物件是節點、節點或節點上的屬性。在資源描述架構 (RDF) 圖形中，物件是四元。

對於任何 Neptune Serverless 叢集，您可以同時設定 Neptune 容量單位 (NCUs的最小和最大數量。每個 NCU 包含 2 GB (GiB) 的記憶體和相關聯的 vCPU 和聯網。最小和最大 NCU 值適用於叢集中的任何無伺服器執行個體。您可以設定的最大 NCU 值最高為 128.0 NCU，而最小值最低為 1.0 NCU。觀察 Amazon CloudWatch 指標`ServerlessDatabaseCapacity`，擷取您經常在其中執行的範圍，並關聯該範圍內的不良行為或成本`NCUUtilization`，以最佳化最適合您應用程式的 NCU 範圍。如果您發現您的工作負載擴展速度不夠快，請增加最小 NCUs，以便在擴展時提供足夠的處理來因應初始突增。

您可以建立的資料庫資源數量上，每個區域 AWS 帳戶 都有配額。這些資源包括資料庫執行個體和資料庫叢集。在您到達某項資源的上限時，所發出用來建立該資源的額外呼叫都會伴隨著異常而失敗。有些配額是可依請求增加的軟配額。如需 Amazon Neptune 和 Amazon RDS、Amazon Aurora 和 Amazon DocumentDB （具有 MongoDB 相容性） 之間共用的配額清單，以及可用時請求增加配額的連結，請參閱 Amazon RDS 中的配額。

了解 Neptune 部署模式

在 Neptune 資料庫叢集中，有一個主要資料庫執行個體和最多 15 個 Neptune 複本。主要資料庫執行個體支援讀取和寫入操作，並對叢集磁碟區執行所有資料修改。Neptune 複本會連線至與主要資料庫執行個體相同的儲存磁碟區，且僅支援讀取操作。Neptune 複本可以從主要資料庫執行個體卸載讀取工作負載。

若要實現高可用性，請使用僅供讀取複本。在不同的可用區域中擁有一或多個僅供讀取複本執行個體可提高可用性，因為僅供讀取複本可做為主要執行個體的容錯移轉目標。如果寫入器執行個體失敗，Neptune 會將僅供讀取複本執行個體提升為主要執行個體。發生這種情況時，提升的執行個體重新啟動時會短暫中斷（通常少於 30 秒），在此期間，對主要執行個體發出的讀取和寫入請求會失敗，但有例外。為了取得最高的可靠性，請考慮使用不同可用區域中的兩個僅供讀取複本。如果可用區域 1 中的主要執行個體離線，可用區域 2 中的執行個體會提升為主要執行個體，但發生這種情況時無法處理查詢。因此，在轉換期間需要可用區域 3 中的執行個體來處理讀取查詢。

如果您使用的是 Neptune Serverless，所有可用區域中的讀取器和寫入器執行個體都會根據其資料庫負載，獨立地進行擴展和縮減。您可以將讀取器執行個體的提升層設定為 0 或 1，以便隨著寫入器執行個體的容量向上和向下擴展。這可讓您隨時接管目前的工作負載。

管理和擴展 Neptune 叢集

您可以使用 Neptune 自動擴展自動調整資料庫叢集中的 Neptune 複本數量，以根據 CPU 使用率閾值滿足您的連線和工作負載需求。透過自動擴展，Neptune 資料庫叢集可以處理工作負載突然增加的問題。當工作負載減少時，自動擴展會移除不必要的複本，這樣您就不會支付未使用的容量。請注意，新的執行個體啟動可能需要長達 15 分鐘的時間，因此僅自動擴展並不足以因應快速的需求變化。

您只能將自動擴展與已有一個主要寫入器執行個體和至少一個僅供讀取複本執行個體的 Neptune 資料庫叢集搭配使用（請參閱 Amazon Neptune 資料庫叢集和執行個體)。此外，叢集中的所有僅供讀取複本執行個體都必須處於可用狀態。如果任何僅供讀取複本處於可用以外的狀態，則 Neptune 自動擴展在叢集中的每個僅供讀取複本都可用之前不會執行任何操作。

如果您遇到需求快速變化，請考慮使用無伺服器執行個體。無伺服器執行個體可以在短時間內垂直擴展，而自動擴展會在較長的期間水平擴展。此組態提供最佳可擴展性，因為無伺服器執行個體垂直擴展，而自動擴展會執行個體化新的僅供讀取複本，以處理超過單一無伺服器執行個體最大容量的工作負載。如需 Amazon Neptune Serverless 容量擴展的詳細資訊，請參閱 Neptune Serverless 資料庫叢集中的容量擴展。

如果您的擴展需要在可預測的時間變更，您可以排程對最小執行個體、最大執行個體和閾值的變更，以更好地處理這些轉移需求。請記得至少提前 15 分鐘排程橫向擴展事件，以允許這些執行個體在需要時上線。

您可以使用參數群組中的參數，在 Amazon Neptune 中管理資料庫組態。參數群組扮演引擎組態值的容器，以套用至一或多個資料庫執行個體。修改參數群組中的叢集參數時，請了解靜態和動態參數之間的差異，以及套用它們的方式和時間。使用狀態端點來查看目前套用的組態。

管理備份和容錯移轉事件

Neptune 會自動備份您的叢集磁碟區，並在備份保留期間保留備份的資料。Neptune 備份具有連續性和增量性，因此，您可以快速還原到備份保留期內的任何時間點。您可以在建立或修改資料庫叢集時指定 1–35 天的備份保留期。

若要在備份保留期間之後保留備份，您也可以拍攝叢集磁碟區中資料的快照。儲存快照需要支付 Neptune 的標準儲存費用。

當您建立資料庫叢集的 Amazon Neptune 快照時，Neptune 會建立叢集的儲存磁碟區快照，備份其所有資料，而不只是個別執行個體。您稍後可以從這個資料庫叢集快照進行還原來建立新的資料庫叢集。當您還原資料庫叢集時，請提供要還原的資料庫叢集快照名稱，然後為還原所建立的新資料庫叢集提供名稱。

測試您的系統如何回應容錯移轉事件。使用 Neptune API 強制容錯移轉事件。當您想要模擬資料庫執行個體的故障，以便在容錯移轉發生後將操作還原至原始可用區域時，使用容錯移轉重新啟動會很有幫助。如需詳細資訊，請參閱設定和管理多可用區域部署。當您重新啟動資料庫寫入器執行個體時，它會容錯移轉到待命複本。重新啟動 Neptune 複本不會啟動容錯移轉。

為您的用戶端設計可靠性。在容錯移轉事件期間測試其行為。在用戶端中以指數退避邏輯實作重試邏輯。您可以在 AWS Lambda Amazon Neptune 的函數範例中找到實作此邏輯的程式碼範例。

AWS Backup 如果您有一組通用的備份需求，則考慮在多個資料庫引擎之間套用，請考慮使用 。

效能效率支柱

AWS Well-Architected Framework 的效能效率支柱著重於如何在擷取或查詢資料時最佳化效能。效能最佳化是下列的增量和持續程序：

- 確認業務需求
- 測量工作負載效能
- 識別效能不佳的元件
- 調校元件以符合您的業務需求

效能效率支柱提供使用案例特定的指導方針，可協助識別正確的圖形資料模型和要使用的查詢語言。它還包括從 Amazon Neptune 擷取資料和使用資料的最佳實務。

效能效率支柱著重於下列關鍵領域：

- 圖形建模
- 查詢最佳化
- 叢集大小正確
- 寫入最佳化

了解圖形建模

了解標籤屬性圖形 (LPG) 與資源描述架構 (RDF) 模型之間的差異。在大多數情況下，這是偏好的問題。不過，有幾個使用案例，其中一個模型比另一個模型更適合。如果您需要圖形中連接兩個節點的路徑知識，請選擇 LPG。如果您想要跨 Neptune 叢集或其他圖形三元存放區聯合資料，請選擇 RDF。

如果您要建置軟體即服務 (SaaS) 應用程式或需要多租用戶的應用程式，請考慮將租用戶的邏輯分離併入資料模型，而不是每個叢集有一個租用戶。若要實現該類型的設計，您可以使用名為 SPARQL 的圖形和標籤策略，例如將客戶識別符新增至標籤，或新增代表租用戶識別符的屬性鍵值對。確保您的用戶端層注入這些值，以保持該邏輯分隔。

查詢的效能取決於處理查詢時需要評估的圖形物件（節點、邊緣、屬性）數量。因此，圖形模型可能會對您的應用程式效能產生重大影響。盡可能使用精細的標籤，並僅存放完成路徑判斷或篩選所需的屬性。若要達到更高的效能，請考慮預先計算圖形的部分，例如建立摘要節點或連接常見路徑的更多直接邊緣。

盡量避免在邊緣數量異常偏高且具有相同標籤的節點之間導覽。這類節點通常有數千個邊緣（其中大多數節點的邊緣計數為十）。結果會提高運算和資料複雜性。這些節點在某些查詢模式中可能不會有問題，但建議您以不同的方式建立資料模型以避免這種情況，尤其是如果您將導覽至節點做為中繼步驟時。您可以使用慢查詢日誌來協助識別跨這些節點導覽的查詢。您可能會觀察到比平均查詢模式高出許多的延遲和資料存取指標，特別是如果您使用除錯模式時。

如果您的使用案例支援節點和邊緣，則使用決定性節點 IDs，而不是使用 Neptune 來為 IDs 指派隨機 GUID 值。依 ID 存取節點是最有效率的方法。

最佳化查詢

openCypher 和 Gremlin 語言可以交替用於 LPG 模型。如果效能是首要考量，請考慮互換使用兩種語言，因為對於特定查詢模式，其中一種可能會比另一種更好。

Neptune 正在轉換為其替代查詢引擎 (DFE)。openCypher 僅在 DFE 上執行，但 Gremlin 和 SPARQL 查詢都可以使用查詢註釋選擇性地設定為在 DFE 上執行。請考慮使用已啟用 DFE 測試您的查詢，並在不使用 DFE 時比較查詢模式的效能。

Neptune 已針對從單一節點或一組節點開始的交易類型查詢進行最佳化，並從那裡散出，而不是評估整個圖形的分析查詢。對於分析查詢工作負載，請考慮使用適用於 Pandas 的 AWS 開發套件，或使用 neptune-export 結合 AWS Glue或 Amazon EMR。

若要識別模型和查詢中的效率不佳和瓶頸，請針對每個查詢語言使用 `profile`和 `explain` API，以取得查詢計劃和查詢指標的詳細說明。 APIs 如需詳細資訊，請參閱 Gremlin 設定檔、openCypher 說明和 SPARQL 說明。

了解您的查詢模式。如果圖形中的不同邊緣數量變大，預設 Neptune 存取策略可能會變得效率低下。下列查詢可能會變得非常沒有效率：

- 未提供邊緣標籤時，跨邊緣向後導覽的查詢。
- 在內部使用此相同模式的子句，例如在 Gremlin `.both()`中，或捨棄任何語言節點的子句（需要捨棄傳入邊緣而不了解標籤）。
- 存取屬性值而不指定屬性標籤的查詢。這些查詢可能會變得相當沒有效率。如果這符合您的使用模式，請考慮啟用 OSGP 索引（物件、主旨、圖形、述詞）。

使用慢查詢記錄來識別慢查詢。緩慢的查詢可能是由未最佳化的查詢計劃或不必要地進行大量索引查詢所造成，這可能會增加 I/O 成本。Gremlin、SPARQL 或 openCypher 的 Neptune 解釋和描述檔端點可協助您了解為何這些查詢速度緩慢。原因可能包括下列項目：

- 與圖形中的平均節點相比，邊緣數量異常高的節點（例如，數千個節點與十個節點相比）可能會增加運算複雜性，因此延遲更長且資源消耗更大。判斷這些節點是否已正確建模，或是否可以改善存取模式，以減少必須周遊的邊緣數量。
- 未最佳化的查詢將包含未最佳化特定步驟的警告。重寫這些查詢以使用最佳化步驟可能會改善效能。
- 備援篩選條件可能會導致不必要的索引查詢。同樣地，備援模式可能會導致重複的索引查詢，可透過改善查詢進行最佳化（請參閱 設定檔輸出`Index Operations - Duplication ratio`中的）。
- Gremlin 等某些語言沒有強式輸入的數值，而是改用類型提升。例如，如果值為 55，Neptune 會尋找整數、長數、浮點數和其他相當於 55 的數值類型。這會導致其他操作。如果您事先知道您的類型相符，您可以使用查詢提示來避免這種情況。
- 您的圖形模型可能會大幅影響效能。考慮使用更精細的標籤或預先計算捷徑到多躍點線性路徑，以降低需要評估的物件數量。

如果僅查詢最佳化不允許您達到效能需求，請考慮搭配 Neptune 使用各種快取技術來達成這些需求。

正確大小的叢集

根據您的並行和輸送量需求調整叢集的大小。叢集中每個執行個體可處理的並行查詢數量等於該執行個體上虛擬 CPUs(vCPUs) 數量的兩倍。在佔用所有工作者執行緒時抵達的其他查詢會放入伺服器端佇列。當工作者執行緒可用時，這些查詢會先first-in-first-out(FIFO) 處理。`MainRequestQueuePendingRequests` Amazon CloudWatch 指標會顯示每個執行個體目前的佇列深度。如果此值經常高於零，請考慮選擇具有更多 vCPU 的執行個體。 vCPUs 如果佇列深度超過 8，192，Neptune 將傳回`ThrottlingException`錯誤。

每個執行個體大約 65% 的 RAM 會預留給緩衝區快取。緩衝區快取會保留有效的資料集（而非整個圖形；僅是查詢中的資料）。若要判斷從緩衝區快取而非儲存中擷取的資料百分比，請監控 CloudWatch 指標 `BufferCacheHitRatio`。如果此指標通常低於 99.9%，請考慮嘗試具有更多記憶體的執行個體，以判斷其是否降低您的延遲和 I/O 成本。

僅供讀取複本的大小不必與寫入器執行個體相同。不過，繁重的寫入工作負載可能會導致較小的複本落後，並重新啟動，因為它們無法跟上複寫。因此，我們建議讓複本等於或大於寫入器執行個體。

為僅供讀取複本使用自動擴展時，請記住，最多可能需要 15 分鐘才能將新的僅供讀取複本上線。當用戶端流量快速增加但可預測時，請考慮使用排程擴展來設定較高的僅供讀取複本數目下限，以考量該初始化時間。

無伺服器執行個體支援數個不同的使用案例和工作負載。針對下列案例，考慮透過佈建執行個體進行無伺服器：

- 您的工作負載經常在一天中波動。
- 您已建立新的應用程式，且不確定工作負載大小為何。
- 您正在執行開發和測試。

請務必注意，無伺服器執行個體比同等佈建執行個體更昂貴，以每 GB RAM 為基準，費用為一美元。每個無伺服器執行個體都包含 2 GB 的 RAM，以及相關聯的 vCPU 和聯網。在您的選項之間執行成本分析，以避免意外帳單。一般而言，只有當工作負載每天只有數小時非常繁重，且一天的其餘時間幾乎為零，或工作負載在一天中大幅波動時，您才能透過無伺服器節省成本。

最佳化寫入

若要最佳化寫入，請考慮下列事項：

- Neptune 大量載入器是初始載入資料庫或附加至現有資料的最佳方式。Neptune 載入器不是交易的，無法刪除資料，因此如果這些是您的需求，請勿使用它。
- 您可以使用支援的查詢語言進行交易更新。若要最佳化寫入輸入/輸出操作，請批次寫入資料，每個遞交 50-100 個物件。物件是 LPG 中節點或邊緣的節點、邊緣或屬性，或是 RDF 中的三重儲存或四元儲存。
- 所有 Neptune 寫入操作都是每個連線的單一執行緒。傳送大量資料至 Neptune 時，請考慮具有多個平行連線，每個連線都是寫入資料。當您選擇 Neptune 佈建執行個體時，執行個體大小會與多個 vCPUs相關聯。Neptune 會為執行個體上的每個 vCPU 建立兩個資料庫執行緒，因此在測試最佳平行處理時，從 vCPUs數量的兩倍開始。 無伺服器執行個體會以大約每 4 個 NCUs 一個的速率擴展 vCPUs 數量。
- 在所有寫入程序期間規劃並有效率地處理 ConcurrentModificationExceptions，即使一次只有單一連線正在寫入資料。設計您的用戶端，確保`ConcurrentModificationExceptions`可靠性。
- 如果您想要刪除所有資料，請考慮使用快速重設 API，而不是發出並行刪除查詢。與前者相比，後者需要更長的時間，並產生大量的 I/O 成本。
- 如果您想要刪除大部分的資料，請考慮使用 neptune-export 將資料載入新叢集，以匯出您要保留的資料。然後刪除原始叢集。

成本最佳化支柱

AWS Well-Architected 架構的成本最佳化支柱著重於避免不必要的成本。下列建議可協助您符合 Amazon Neptune 的成本最佳化設計原則和架構最佳實務。

成本最佳化支柱著重於下列關鍵領域：

- 了解一段時間內的支出並控制資金分配
- 選取正確類型和數量的資源
- 擴展以滿足業務需求，而不會過度花費

了解所需的用量模式和服務

如果您的資料模型具有可識別的圖形結構，且您的查詢需要探索關係並周遊多個躍點，Neptune 非常適合您的工作負載。圖形資料庫不適合下列模式：

- 主要是單一跳轉查詢（考慮您的資料是否可更好地表示為物件的屬性）
- 儲存為屬性的 JSON 或 BLOB 資料
- 跨資料集彙總的查詢，例如計算大量節點的數值屬性總和

考慮將多個專用資料庫結合使用，以用於特定存取模式，是否可滿足您的所有需求。例如：

- 需要較不頻繁複雜圖形導覽以及高度並行擷取單一節點屬性的 API，最好使用 Neptune、DynamoDB 或 Amazon DocumentDB 中的一或多個項目來呈現。
- 關聯式資料庫可以與 Neptune 共存，以維護您現有的功能，但 Neptune 僅用於在關聯式資料庫中執行和擴展不佳的多躍點周遊。

了解與 Neptune 互動和補充 Neptune 的服務相關的成本，包括下列項目：

- Amazon Simple Storage Service (Amazon S3) 大量載入 Neptune 的資料檔案儲存成本
- 用於插入或升級查詢、讀取查詢和 Neptune 串流處理的 Lambda 函數
- 建置在 Neptune 上的 API 層，可與 或 中的用戶端應用程式（而非直接連線至資料庫）互動 AWS AppSync
- AWS Glue 用於在 Neptune 之間傳輸資料的任務

- Amazon Kinesis 或 Amazon Managed Streaming for Apache Kafka (Amazon MSK) 執行個體接收串流資料以近乎即時地擷取至 Neptune。
- AWS Database Migration Service 用於將關聯式資料遷移至 Neptune
- Jupyter 筆記本和深度圖形程式庫機器學習模型的 Amazon SageMaker Runtime 成本

選取關注成本的資源

Neptune 定價是根據每小時執行個體成本（或無伺服器使用的 Neptune 運算單位）、資料 I/O 和儲存用量。執行個體平均佔整體成本的 85%，因此適當調整大小可能會造成重大的成本影響。正確調整執行個體大小的最佳方法是在各種執行個體上測試應用程式效能，並比較下列因素：

- `MainRequestQueuePendingRequests` CloudWatch 指標是否保持在接近零的一致低數？
- `BufferCacheHitRatio` CloudWatch 指標大部分時間是否保持在 99.9% 或更高？
- 執行個體成本和相關聯資料 I/O 成本的成本和效能曲線是多少？ 需要頻繁將緩衝區快取交換與儲存體的過小執行個體，可能會大幅增加資料讀取成本。 在這些情況下， `BufferCacheHitRatio` 會經常捨棄。

執行個體成本會隨著相同執行個體系列中的大小線性擴展。`db.r6i.2xlarge` 執行個體的每小時成本是`db.r6i.xlarge`執行個體的兩倍，也有資源分配的兩倍。`db.r6i.24xlarge` 執行個體是`db.r6i.xlarge`執行個體每小時成本的 24 倍。

估計您必須支援的並行查詢數量。您可以擁有介於零到十五個僅供讀取複本，以處理唯讀查詢。如果您的需求因一天、一週或一個月的時間而異，您可以使用多個較小的執行個體來根據排程進行擴展。執行個體上的每個 vCPU 提供兩個執行緒，用於處理並行查詢。三個僅供`db.r6i.xlarge`讀取複本，每個都有 4 個 vCPU，可以處理 24 個並行查詢。

如果您的流量改為以每秒查詢 (QPS) 來測量，您必須實驗來判斷查詢的平均延遲。Neptune 叢集可支援的每秒查詢數量等於 `vCPU × 2 × (1 second/average query latency)`。例如，如果您有 4 個 vCPU 且查詢延遲為 100 毫秒 (0.1 秒），則 `QPS = 4 × 2 × (1s/0.1s) = 80 queries per second`。

佈建的執行個體，對於持續、穩定且可預測的工作負載而言，比無伺服器更便宜。當您的工作負載每天只需要數小時（例如 `db.r6i.4xlarge`) 的高度使用量，然後當天的剩餘時間幾乎沒有流量（例如 1 個 Neptune 運算單位），則無伺服器提供最佳化成本的機會。無伺服器執行個體會擴展數小時，然後再縮減，比起整天使用佈建的`db.r6i.4xlarge`執行個體，成本更低。

為您的工作負載選擇最佳的 Neptune 執行個體組態

如需 Neptune 的入門級實驗，您可以使用 AWS 免費方案。750 個免費小時的 `db.t3.medium`和 `db.t4g.medium` 執行個體用量足以讓您以低規模充分了解 Neptune。您的叢集將在免費試用期結束後保留，但從該時間點開始，您將需要支付使用費。

`db.t3.medium` 和 `db.t4g.medium`執行個體適合低成本的開發環境，但請注意，其 RAM 與 vCPU 的比率 (2：1) 比 R 系列執行個體 (8：1) 或 X 系列執行個體 (16：1) 更小。效能描述檔可能與這些類別不同，特別是在查詢瀏覽圖形大部分`OutOfMemoryExceptions`時。若要判斷後者條件是否可能受到影響，請檢查 `BufferCacheHitRatio` CloudWatch 指標。

我們強烈建議您不要使用 T 系列執行個體進行任何效能或負載測試，因為您可能會遇到不一致的結果，這些結果並不表示生產環境。

當您的工作負載相當穩定且可預測時，佈建的執行個體可為您提供最佳的成本和效能組合。根據所需的請求並行和查詢複雜性選擇執行個體大小。較高的並行需要更多 vCPUs。較高的查詢複雜性需要更多 RAM。使用 `MainRequestQueuePendingRequests` CloudWatch 指標來判斷前者的影響 （大於零表示的並行請求多於可處理的請求）。使用 `BufferCacheHitRatio` CloudWatch 指標來判斷後者的影響。經常低於 99.9% 的比率表示沒有足夠的 RAM 來包含正在評估的圖形工作部分，這會導致更頻繁的快取交換。如果 R 系列執行個體提供足夠的並行但沒有足夠的 RAM，請考慮嘗試 X 系列執行個體。

Neptune 文件說明無伺服器執行個體的理想使用案例。如果您不確定佈建或無伺服器是否最適合您，而成本是您的主要考量，請在無伺服器中測試工作負載，以判斷使用的 NCUs 數量，並將佈建 (`N hours × hourly provisioned cost`) 的成本與無伺服器 () 進行比較`sum of NCUs × hourly cost per NCU`。如果您不確定同等大小的佈建執行個體，一個 NCU 相當於大約 2 GB 的 RAM 和相關聯的 vCPU 和聯網。如果您的佈建執行個體來自 R6i 系列，則比率為每 8 GB RAM 1 個 vCPU，或 4 NCUs，以及相關聯的聯網。

將無伺服器用於主要執行個體和複本執行個體時，請記住，促銷方案 0 和 1 中的僅供讀取複本會根據寫入器執行個體擴展其 NCUs，以便在容錯移轉事件發生時適當擴展。根據您的哪些執行個體 - 寫入器或讀取器 - 接收最多流量，設定這些執行個體的 NCU 限制。

在每週 7 天、每天 24 小時不需要叢集的環境中，請考慮撰寫指令碼，以便在不使用時關閉 Neptune 執行個體，並在使用前再次啟動它們。Neptune 執行個體會每 7 天自動重新啟動一次，以確保套用必要的維護更新。如果您打算長時間關閉執行個體，請使用每週指令碼再次關閉執行個體。

適當大小的資料儲存和傳輸

更有效率的查詢（例如，需要接觸較少節點、邊緣和圖形中屬性的查詢）需要較少的 I/O 傳輸，而且可能可以使用較小的執行個體，因為需要較少的緩衝區快取。使用設定檔或解釋查詢語言的端點來最佳化您的查詢，並考慮最佳化您的圖形模型以滿足您的查詢效能。

Neptune 在大型字串上使用字典編碼，該字典針對效能進行了最佳化，而非效率。如果您有大型 BLOBs、JSON 或經常變更屬性的字串，請考慮將它們存放在 Amazon S3、Amazon DynamoDB 或 Amazon DocumentDB 中的 Neptune 外部，並在 Neptune 節點中僅存放參考。

在某些情況下，選擇較大的執行個體大小可能會比較便宜。如果您的 I/O 成本因為 低而非常高`BufferCacheHitRatio`，則較大的緩衝區快取可能會大幅降低該成本。這是因為所有資料都適合快取中，而不是經常從儲存體交換並產生 I/O 傳輸速率。

Neptune 使用copy-on-write。複製將圖形分割成多個碎片時，最好不要刪除複製叢集上不需要的資料，因為這將涉及建立新資料頁面，進而增加儲存成本。在複製事件之前未變更的資料將存在於兩個叢集共用的單一資料頁面中，並且只會針對該單一複本收費。

請勿啟用 OSGP 索引或使用 R5d 執行個體，除非您已測試確認它們對您的工作負載造成重大影響。這兩者都專為極少發生的案例而設計，而且它們可能會以最少或沒有收益的方式增加您的成本。

永續性支柱

永續性支柱著重於將執行雲端工作負載的環境影響降至最低。關鍵主題包括共同的責任模型，用於永續性、了解影響和最大化使用，以將必要的資源降至最低並減少下游影響。

永續性支柱包含下列主要重點領域：

- 您的影響
- 永續性目標
- 最大化用量
- 預期和採用新的、更有效率的硬體和軟體產品
- 使用 受管服務
- 降低下游影響

本指南著重於您的影響。如需其他永續性設計原則的詳細資訊，請參閱 AWS Well-Architected Framework。

您的選擇和要求會影響環境。如果您選擇 AWS 區域 碳強度較低的 ，而且如果您的需求反映實際的工作負載需求，而不只是最大化運作時間和耐用性，工作負載的永續性就會增加。下一節討論最佳實務和周到考量事項，如果在工作負載設計和持續操作中採用，將對環境產生正面影響。

AWS 區域選擇

有些 AWS 區域 接近 Amazon 可再生能源專案，或位於電網已發佈的碳強度低於其他項目的位置。請考慮對於工作負載可能可行之區域的永續性影響，並將您的清單與可使用 Neptune 的區域交叉參考。

根據使用者行為模式的使用

正確調整用量大小以符合使用者的流量和行為，有助於將服務對環境的影響 AWS 降至最低。設計解決方案時，請考慮下列最佳實務：

- 監控 Amazon CloudWatch 指標，例如 `CPUUtilization`、`MainRequestQueuePendingRequests`和 `TotalRequestsPerSec`，以判斷您的需求何時最高和最低，並確保您的叢集資源在這段期間大小正確。
- 在未使用非生產環境的時段內，自動停止非生產環境。如需詳細資訊，請參閱部落格文章使用資源標籤自動停止和啟動 Amazon Neptune 環境資源。

- 如果您的流量模式經常且無法預測地變化，請考慮使用隨需求擴展和縮減的 Neptune Serverless 執行個體，而不是使用為尖峰流量佈建的執行個體。
- 除了業務連續性目標之外，請考慮將您的服務層級協議與永續性目標保持一致。減輕多區域災難復原、高可用性或長期備份保留等需求，尤其是非生產環境或非任務關鍵工作負載，可以減少實現這些目標所需的資源量。

最佳化軟體開發和架構模式

若要避免浪費，請最佳化模型和查詢，並共用運算資源，以便使用 Neptune 執行個體和叢集中可用的所有資源。特定最佳實務包括：

- 讓開發人員共用 Neptune 執行個體和 Jupyter Notebook 應用程式執行個體，而不是每個建立自己的執行個體。透過使用多租用戶分割策略，在單一 Neptune 叢集中為每個開發人員提供自己的邏輯分割區，並在單一 Jupyter 執行個體上為每個開發人員建立個別的筆記本資料夾。
- 實作模式，將資源的使用率最大化，並將閒置時間降至最低，例如將資料和批次記錄一起載入大型交易的平行執行緒。
- 最佳化您的查詢和圖形模型，將計算結果所需的資源降至最低。
- 對於 Gremlin 查詢結果，請使用結果快取功能，將重新計算分頁或經常重複查詢所花費的資源降至最低。
- 讓您的 Neptune 環境保持最新狀態。最新版本的 Neptune 支援更有效率的最新 EC2 執行個體，例如 Graviton。它們也具有查詢最佳化改進和錯誤修正，可減少計算查詢所需的資源量。

資源

參考

- AWS Well-Architected
- AWS Well-Architected Framework 文件
- 套用 AWS Well-Architected Framework for Amazon Neptune Analytics
- Amazon Neptune 最新更新
- 最佳實務：充分利用 Neptune

部落格文章

- 使用 Apache TinkerPop Gremlin 自動化測試 Amazon Neptune 資料存取
- 使用資源標籤自動停止和啟動 Amazon Neptune 環境資源
- Amazon Neptune 資料平面動作的精細精細存取控制
- 將 RDFox 與 Amazon Neptune 整合，使用語意推理從您的 RDF 圖形推斷新事實 Amazon Neptune
- 使用 Amazon Neptune ML 建置即時詐騙偵測解決方案

免費 AWS Skill Builder 課程

- Amazon Neptune 入門
- 在 Amazon Neptune 上建置應用程式
- Amazon Neptune 的資料建模

貢獻者

本指南的貢獻者包括：

- Brian O'Keefe，首席 Neptune 解決方案架構師， AWS
- Abhishek Mishra | 資深 Neptune 解決方案架構師 AWS
- Ganesh Sawhney | 策略合作夥伴成功解決方案架構師、 AWS
- Michael Havey | 資深 Neptune 解決方案架構師 AWS
- Kevin Phillips，Neptune 解決方案架構師， AWS
- Melissa Kwok，Neptune 解決方案架構師， AWS
- Sakti Mishra，首席解決方案架構師 AWS
- Javed Ali | 資深解決方案架構師 AWS

文件歷史紀錄

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

變更	描述	日期
初次出版	—	2023 年 9 月 27 日

AWS 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的術語。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：將您的現場部署 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：將您的現場部署 Oracle 資料庫遷移至 中的 Amazon Relational Database Service (Amazon RDS) for Oracle 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：將您的現場部署 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到相同平台的雲端服務。範例：將 Microsoft Hyper-V應用程式遷移至 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

請參閱原子性、一致性、隔離性、耐久性。

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權。匿名資料不再被視為個人資料。

反模式

經常用於重複性問題的解決方案，其中解決方案具有反生產力、無效或比替代解決方案更有效。

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攻擊。

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。如需有關如何在 AWS 遷移策略中使用 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。

原子性、一致性、隔離性、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 (IAM) 文件》中的 ABAC for AWS。

授權資料來源

您存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、修訂或假名化資料。

可用區域

中的不同位置 AWS 區域 ，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、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、安全和營運。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。因此， AWS CAF 為人員開發、訓練和通訊提供指引，協助組織做好成功採用雲端的準備。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)

一種工具，可評估資料庫遷移工作負載、建議遷移策略，並提供工作預估值。 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。您可以將行為圖與 Amazon Detective 搭配使用來檢查失敗的登入嘗試、可疑的 API 呼叫和類似動作。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 Endianness。

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。此策略可協助您快速復原，並將影響降至最低。

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。有些其他機器人稱為惡意機器人，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受單一方控制之機器人的網路，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，讓使用者能夠快速存取他們通常無權存取 AWS 帳戶 的 。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS 實作打破玻璃程序指標。

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

版本對最終使用者的緩慢和增量版本。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。您可以將 CDC 用於各種用途，例如稽核或複寫目標系統中的變更以保持同步。

混沌工程

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，為您的 AWS 工作負載帶來壓力，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。

雲端卓越中心 (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、建立營運模型)

- 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 於部落格文章 The Journey Toward Cloud-First 和 Enterprise Strategy 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 遷移策略相關的詳細資訊，請參閱遷移整備指南。

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。程式碼的每個版本都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

冷快取

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

冷資料

很少存取且通常是歷史資料的資料。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影像和影片等視覺化格式分析和擷取資訊的 AI 欄位。例如，Amazon SageMaker AI 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。您通常在遷移的產品組合探索和分析階段使用 CMDB 中的資料。

一致性套件

您可以組合的 AWS Config 規則和修補動作集合，以自訂您的合規和安全檢查。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)

自動化軟體發行程序的來源、建置、測試、暫存和生產階段的程序。CI/CD 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。如需詳細資訊，請參閱持續交付的優點。CD 也可表示持續部署。如需詳細資訊，請參閱持續交付與持續部署。

CV

請參閱電腦視覺。

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更。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，提供分散式、分散式資料擁有權與集中式管理。

資料最小化

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

資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱在 上建置資料周邊 AWS。

資料預先處理

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

資料來源

在整個資料生命週期中追蹤資料的來源和歷史記錄的程序，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。

資料庫處理語言 (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。

深度防禦

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，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 中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶來管理組織的帳戶，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)

一種程序，用於識別並優先考慮對軟體開發生命週期中的速度和品質造成負面影響的限制。DVSM 擴展了最初專為精簡製造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

在星星結構描述中，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，其行為類似於文字。這些屬性通常用於查詢限制、篩選和結果集標記。

災難

防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 上工作負載的災難復原 AWS：雲端中的復原。

DML

請參閱資料庫處理語言。

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。如需有關如何將領域驅動的設計與 strangler fig 模式搭配使用的資訊，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。

偏離偵測

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 來偵測登陸區域中可能影響控管要求合規性的變更。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html

DVSM

請參閱開發值串流映射。

E

EDA

請參閱探索性資料分析。

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改善回應時間。

電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱什麼是電子資料交換。

加密

將人類可讀取的純文字資料轉換為加密文字的運算程序。

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) 委託人。這些帳戶或主體可以透過建立介面 VPC 端點私下連接至您的端點服務。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)

一種系統，可自動化和管理企業的關鍵業務流程（例如會計、MES 和專案管理）。

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 () 文件中的信封加密。 AWS Key Management Service AWS KMS

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 安全概念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。透過計算摘要統計並建立資料可視化來執行 EDA。

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：包含度量的資料，以及包含維度資料表外部索引鍵的資料欄。

快速失敗

一種使用頻繁且增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在 中 AWS 雲端，像是可用區域 AWS 區域、控制平面或資料平面等界限會限制故障的影響，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。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱 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。

少量擷取提示

在要求 LLM 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。少量的提示對於需要特定格式、推理或網域知識的任務來說非常有效。另請參閱零鏡頭提示。

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，以盡可能在最短的時間內遷移資料，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，已在廣義和未標記資料的大量資料集上進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱什麼是基礎模型。

G

生成式 AI

已針對大量資料進行訓練的 AI 模型子集，可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)

Amazon CloudFront 中的選項，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 工作流程會被視為舊版，而以幹線為基礎的工作流程是現代、偏好的方法。

黃金影像

系統或軟體的快照，做為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

有助於跨組織單位 (OU) 來管控資源、政策和合規的高層級規則。預防性防護機制會強制執行政策，以確保符合合規標準。透過使用服務控制政策和 IAM 許可界限來將其實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。它們是透過使用 AWS Config AWS Security Hub、Amazon GuardDuty、、Amazon Inspector AWS Trusted Advisor和自訂 AWS Lambda 檢查來實作。

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)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統的設計目的是自動容錯移轉、持續提供高品質的效能，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

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種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部分歷史標記資料。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

將您的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。

熱資料

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，通常會在典型 DevOps 發行工作流程之外執行修補程式。

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

90 天期間 CPU 和記憶體平均使用率在 5% 至 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。不可變基礎設施本質上比可變基礎設施更一致、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 (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 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

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

Klaus 於 2016 年引進的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進展，指製造程序的現代化。

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，請參閱建立工業物聯網 (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 之間（在相同或不同的 中 AWS 區域)、網際網路和內部部署網路之間的網路流量檢查。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，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 的機器學習模型可解譯性 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 為 ITSM 提供了基礎。

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。如需有關將雲端操作與 ITSM 工具整合的資訊，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獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會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、摘要文件、將文字翻譯成其他語言，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 個 R。

小端序系統

首先儲存最低有效位元組的系統。另請參閱 Endianness。

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 會 AWS 操作基礎設施層、作業系統和平台，而您會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)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

MAP

請參閱遷移加速計劃。

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。

製造執行系統

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

一種小型的獨立服務，它可透過定義明確的 API 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 API，透過明確定義的介面進行通訊。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)

一種 AWS 計畫，提供諮詢支援、訓練和服務，協助組織建立強大的營運基礎，以移至雲端，並協助抵銷遷移的初始成本。MAP 包括用於有條不紊地執行舊式遷移的遷移方法以及一組用於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營運、業務分析師和擁有者、遷移工程師、開發人員以及從事 Sprint 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。

遷移組合評定 (MPA)

線上工具，提供驗證商業案例以遷移至 的資訊 AWS 雲端。MPA 提供詳細的組合評定 (伺服器適當規模、定價、總體擁有成本比較、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，以及波次規劃)。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。

遷移準備程度評定 (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

用來將工作負載遷移至 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，並請參閱動員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《》中的現代化應用程式的策略 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 - 統一架構。

開放程序通訊 - 統一架構 (OPC-UA)

用於工業自動化machine-to-machine(M2M) 通訊協定。OPC-UA 提供資料加密、身分驗證和授權機制的互通性標準。

操作水準協議 (OLA)

一份協議，闡明 IT 職能群組承諾向彼此提供的內容，以支援服務水準協議 (SLA)。

操作整備審查 (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 中的操作準備度審查 (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，整合 OT 和資訊技術 (IT) 系統是工業 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 建立的線索 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有 的所有事件 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 文件中的建立組織追蹤。

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 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。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)

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

原始存取身分 (OAI)

CloudFront 中的一個選項，用於限制存取以保護 Amazon S3 內容。當您使用 OAI 時，CloudFront 會建立一個可供 Amazon S3 進行驗證的主體。經驗證的主體只能透過特定 CloudFront 分發來存取 S3 儲存貯體中的內容。另請參閱 OAC，它可提供更精細且增強的存取控制。

ORR

請參閱操作整備審核。

OT

請參閱操作技術。

傳出 (輸出) VPC

在 AWS 多帳戶架構中，處理從應用程式內啟動之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

P

許可界限

附接至 IAM 主體的 IAM 管理政策，可設定使用者或角色擁有的最大許可。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

請參閱個人身分識別資訊。

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

可定義許可的物件（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。

設計隱私權

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

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 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，並參閱實作安全控制項中的主動控制項。 AWS

產品生命週期管理 (PLM)

管理產品整個生命週期的資料和程序，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 提示的輸出做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相關性，並允許更精細、個人化的結果。

擬匿名化

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

發佈/訂閱 (pub/sub)

一種模式，可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、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

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、諮詢、告知 (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 個 R。

區域

地理區域中的 AWS 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 個 R。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

請參閱 7 Rs。

Replatform

請參閱 7 Rs。

回購

請參閱 7 Rs。

彈性

應用程式抵禦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 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。

負責者、當責者、事先諮詢者和事後告知者 (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 個 R。

檢索增強生成 (RAG)

一種生成式 AI 技術，其中 LLM 會在產生回應之前參考訓練資料來源以外的授權資料來源。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，讓攻擊者更難存取登入資料。

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 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 (IdP) 使用的開放標準。此功能可啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而無需為您組織中的每個人在 IAM 中建立使用者。如需有關以 SAML 2.0 為基礎的聯合詳細資訊，請參閱 IAM 文件中的關於以 SAML 2.0 為基礎的聯合。

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，例如密碼或使用者登入資料。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 秘密中的什麼內容？。

依設計的安全性

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。

安全資訊與事件管理 (SIEM) 系統

結合安全資訊管理 (SIM) 和安全事件管理 (SEM) 系統的工具與服務。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，並產生提醒。

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 執行個體或輪換登入資料。

伺服器端加密

由接收資料的 AWS 服務 在其目的地加密資料。

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。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。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。

服務水準協議 (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。

單一故障點 (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

請參閱服務層級指標。

SLO

請參閱服務層級目標。

先拆分後播種模型

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

請參閱單一故障點。

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。此結構旨在用於資料倉儲或商業智慧用途。

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。如需有關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

您 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

一種向 LLM 提供內容、指示或指導方針以指示其行為的技術。系統提示有助於設定內容，並建立與使用者互動的規則。

T

標籤

做為中繼資料以組織 AWS 資源的鍵值對。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

可以用於互連 VPC 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 AWS Organizations 及其帳戶中執行任務。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 搭配使用 AWS Organizations 與其他 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

兩個比薩就能吃飽的小型 DevOps 團隊。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

也稱為繁重工作，這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 對等互連

兩個 VPC 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 VPC 對等互連。

漏洞

危害系統安全性的軟體或硬體瑕疵。

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，通常可接受中等速度的查詢。

視窗函數

SQL 函數，對與目前記錄在某種程度上相關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，然後再遷移伺服器和應用程式。

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)

儲存模型，可一次性寫入資料，並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

零時差入侵

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瑕疵或漏洞。威脅行為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。

零鏡頭提示

提供 LLM 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。零鏡頭提示的有效性取決於任務的複雜性和提示的品質。另請參閱少量擷取提示。

殭屍應用程式

CPU 和記憶體平均使用率低於 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。