在 上測試無伺服器應用程式 AWS

AWS 方案指引

AWS 方案指引: 在 上測試無伺服器應用程式 AWS

Table of Contents

在 上測試無伺服器應用程式 AWS

Amazon Web Services ???（貢獻者）

2026 年 3 月 (文件歷史記錄)

本指南討論測試無伺服器應用程式的方法論，說明測試期間可能遇到的挑戰，並介紹最佳實務。這些測試技術旨在協助您更快地進行反覆運算並更自信地發佈程式碼。

本指南適用於希望為其無伺服器應用程式建立測試策略的開發人員。您可以使用本指南作為了解測試策略的起點，然後訪問無伺服器測試範例儲存庫，查看遵循本指南中描述的模式和最佳實務的測試範例。本指南說明無伺服器測試方法、說明客戶在測試無伺服器應用程式時遇到的挑戰，並介紹測試無伺服器應用程式的最佳實務。這些技術旨在協助開發人員更快速地迭代，並更有信心地發佈。

概觀

自動化測試是重要的投資，有助於確保應用程式品質和開發速度。測試也能加速開發人員的意見回饋。身為開發人員，您希望能夠在應用程式上進行快速反覆運算，並獲取程式碼品質的意見回饋。許多開發人員習慣於撰寫應用程式，這些應用程式會部署到桌上型電腦的環境中，無論是直接部署到作業系統，或是在容器型環境中。當您在桌上型電腦或容器型環境中工作時，通常會針對完全在桌上型電腦中託管的程式碼撰寫測試。不過，在無伺服器應用程式中，架構元件可能無法部署至桌面環境，而可能只存在於雲端中。雲端架構可能包括持久性層、簡訊系統、安全建構、APIs 和其他元件。當您編寫依賴於這些元件的應用程式碼時，可能很難確定設計和執行測試的最佳方法。

本指南可協助您調整測試策略，以減少摩擦和混淆，並提高程式碼品質。

先決條件

本指南假設您熟悉自動化測試的基礎知識，包括如何使用自動化軟體測試來確保軟體品質。本指南提供無伺服器應用程式測試策略的高階簡介，不需要任何實際操作的撰寫測試經驗。

定義

本指南使用以下術語：

- 單元測試是針對單個架構元件的程式碼單獨執行的測試。
- 整合測試 會針對兩個或多個架構元件執行，通常是在雲端環境中。
- End-to-end測試會驗證整個應用程式或工作流程的行為。

- 模擬器是應用程式（通常由第三方提供），旨在模擬雲端服務，而無需佈建或調用任何雲端資源。
- 模擬 (也稱為仿造) 是測試應用程式中的實作，用該相依性的模擬來取代某個相依性。

目標

本指南中的最佳實務旨在協助您達成兩個主要目標：

- 提高無伺服器應用程式的品質
  - 在架構界限進行測試
  - 在程式碼邊界進行測試
- 縮短實作或變更功能的時間

提高軟體品質

應用程式的品質在很大程度上取決於開發人員測試各種案例以驗證功能的能力。當您未實作自動化測試時，或者更常見的是，如果您的測試無法充分涵蓋所需的案例，則無法判斷或保證應用程式的品質。

在伺服器型架構中，團隊能夠輕鬆定義測試範圍：必須測試在應用程式伺服器上執行的程式碼。呼叫伺服器的其他元件或伺服器呼叫的相依性，通常會被負責伺服器應用程式的團隊認為是外部項目或超出測試範圍。

無伺服器應用程式通常由較小的工作單位組成，例如 AWS Lambda 函數，其在自己的環境中執行。團隊可能會在單一應用程式中負責很多倍這樣的小型單位。部分應用程式功能可完全委派給受管服務 (例如 Amazon Simple Storage Service (Amazon S3) 或 Amazon Simple Queue Service (Amazon SQS))，而無需使用任何內部開發的程式碼。用於軟體測試的傳統基於伺服器的模型可能會將受管服務視為應用程式外部的服務，從而將其排除在外。這可能會導致覆蓋範圍不足，其中關鍵場景可能僅限於手動探索性測試，或是一些結果因環境而異的整合測試案例。因此，採用包含託管服務行為和雲端組態的測試策略可以提高軟體品質。

在架構界限進行測試

隨著無伺服器應用程式的成長，它們自然會分散在多個架構元件中。雖然這使用 AWS 分散式功能，但可能會讓end-to-end行為難以理解。

識別自然界限

依照無伺服器最佳實務設計架構時（一個函數 = 一個任務，解耦），您會注意到子系統周圍的自然界限。這些界限代表應用程式中的邏輯分隔點。

做為測試合約的界限

這些架構界限是測試邊緣的絕佳候選項目。將每個界限視為合約，並驗證其行為符合其定義的規格。將這些界限視為應用程式中的接縫，您可以在其中插入測試驗證。

主要優點

以下是在架構界限測試的主要優點：

- 聚焦測試範圍 – 獨立測試子系統，而無需了解整個應用程式。
- 合約驗證 – 確保每個界限在系統演進時維持其預期的行為
- 雙用途檢測 - 這些相同的界限在生產中產生了絕佳的可觀測性勾點
- 測試繫帶 – 可讓您測試非同步無伺服器系統。它可協助您在事件流經子系統時擷取和驗證事件，以測試事件驅動的架構。

在程式碼邊界進行測試

透過將 Lambda 程式碼等基礎設施程式碼與您的核心業務邏輯分開來定義清晰的程式碼界限。此分離會建立不同的測試範圍，以簡化您的測試策略。

邊界模式

在 Lambda 函數中建立兩個清晰的程式碼邊界：

- 外部界限 (Lambda 處理常式） – 一種薄型轉接器層，可處理 AWS Lambda
- 內部界限 （商業邏輯） – 獨立於 Lambda 執行期的純商業邏輯方法

作為轉接器的處理常式 （外部範圍）

您的 Lambda 函數處理常式應該是一個薄層：

- 從傳入 `event`和 `context` 物件擷取資料
- 驗證擷取的資料
- 僅將相關詳細資訊傳遞至商業邏輯方法
- 以 Lambda 的預期格式傳回結果

商業邏輯 （內部範圍）

您的核心商業邏輯應該：

- 獨立於 Lambda 的特定詳細資訊操作
- 接受簡單且經過驗證的輸入
- 傳回可預測的輸出
- 初始化需要最少的相依性

依範圍測試優點

- 內部界限測試 – 針對商業邏輯進行全面的單元測試，無需 Lambda 複雜性或環境設定
- 外部界限測試 – 驗證轉接器層事件處理和資料擷取的重點整合測試
- 最低測試負荷 – 大多數測試不需要複雜的環境或廣泛的相依性

這種以界限為基礎的方法可讓您將大部分程式碼測試為純函數，同時保持 Lambda 測試的最小和目標性。

縮短實作或變更功能的時間

您可以在反覆開發週期期間發現這些問題，將軟體錯誤和組態問題對成本和排程的影響降至最低。當開發人員無法偵測到這些問題時，更多人必須投入額外的精力來識別問題。

無伺服器架構包括透過 API 呼叫提供關鍵應用程式功能的受管服務。因此，您的開發週期應包含測試，以驗證快樂的路徑（與這些服務的互動如預期般運作）和悲傷的路徑（呼叫失敗、傳回非預期的回應，或跨環境的行為不同）。如果沒有這些測試，您可能會因為本機環境與部署環境之間的差異而遇到問題。發生這種情況時，您必須花費額外的時間來嘗試重現和驗證修正，因為每個反覆運算現在都需要針對與您偏好設定不同的環境驗證變更。

適當的無伺服器測試策略可為包括呼叫其他服務的測試提供準確的結果，從而縮短反覆運算時間。

無伺服器應用程式的測試技巧

針對無伺服器應用程式程式碼執行測試有三種主要方法：模擬測試、模擬測試和雲端測試。您通常會發現，在單個專案內會混合使用這些方法。例如，您可以在本機開發程式碼時使用模擬測試、在部署之前更緊密地複寫服務行為的模擬測試，以及作為夜間持續整合和持續交付 (CI/CD) 程序一部分的雲端測試。

模擬測試

模擬測試是一個策略，可透過它在程式碼中建立替代物件，來模擬雲端服務的行為。例如，您可以撰寫使用 Amazon S3 服務模擬的測試，也可以將該模擬設定為在呼叫`PutObject`方法時傳回特定回應物件。執行測試時，該模擬會將回應傳回，而無需呼叫 Amazon S3 或任何其他服務端點。

這些替代物件通常由模擬框架產生，以減少給定測試必需的實作量。有些模擬架構是通用的，有些則是專門針對 AWS SDKs。

模擬與虛設都屬於更廣泛的仿造類別。模擬物件與模擬器不同 (本節稍後會討論)，模擬通常由開發人員建立或設定，作為測試程式碼的一部分，而模擬器是獨立的應用程式，會以與其所模擬系統的相同方式公開 API (例如 REST API)。

以下是使用模擬物件的優勢：

- 模擬物件可以模擬超出應用程式控制範圍的第三方服務，例如 API 和軟體即服務 (SaaS) 供應商，無需直接存取這類服務。
- 模擬物件在測試失敗條件非常實用，尤其當這種情況 (例如服務中斷) 很難模擬時。
- 如同模擬器，模擬架構可以在設定後提供快速的本機開發。
- 模擬物件可以為幾乎任何類型的物件提供替代行為，因此模擬策略可以為各種比模擬器更廣泛的服務建立涵蓋範圍。
- 當新功能或行為可用時，模擬測試可以使用 （或回復） 通用模擬架構來更快地做出反應，該架構可在更新後的 AWS SDK 可用時立即模擬新功能。

以下是模擬物件測試的缺點：

- 模擬物件通常需要進行相當複雜的設定和組態工作，尤其是在嘗試確定不同服務的傳回值以正確模擬回應的情況。

- 因為模擬由編寫測試的開發人員編寫或設定，所以這是開發人員的額外責任。
- 您可能需要存取雲端，才能了解服務的 API 和傳回值。
- 模擬也可能很難維護，因為每當被模擬的雲端 API 簽章變更時、傳回值結構描述演變時、或者測試的應用程式邏輯擴展為呼叫新 API 時，其都需要更新。這些變更會為開發人員帶來額外的測試開發工作量。
- 模擬測試可能會在本機通過，但在雲端中失敗，因為模擬 - 而不是複寫 - 實際服務的行為，使得未偵測到環境特定的問題。
- 模擬架構，例如模擬器，無法偵測 AWS Identity and Access Management (IAM) 政策違規、服務配額限制或承載大小限制，也不會觸發可能影響部署環境中應用程式行為的輔助服務 AWS CloudTrail，例如 Amazon CloudWatch 或 Amazon GuardDuty。
- 雖然模擬比較適合模擬應用程式在授權失敗或超過配額時將執行的動作，但模擬測試無法判斷程式碼部署到生產環境時實際發生的結果。

仿真測試

模擬測試由本機執行的應用程式啟用，稱為類似 的模擬器 AWS 服務。模擬器具有類似於其雲端對應項目的 API，且會提供類似的傳回值。模擬器也可以模擬由 API 呼叫啟動的狀態變更。例如，呼叫 `PutObject`方法時，Amazon S3 模擬器可能會在本機磁碟上存放物件。當 使用相同的金鑰`GetObject`呼叫 時，模擬器會從磁碟讀取相同的物件並傳回它。

以下是模擬測試的優點：

- 它為習慣在本機環境中專門開發程式碼的開發人員提供最熟悉的環境。例如，如果您熟悉 n 層應用程式的開發，您可能有一個資料庫引擎和 Web 伺服器，類似於在生產環境中執行的，在本機電腦上執行以提供快速、本機、隔離的測試功能。
- 它不需要對雲端基礎設施 （例如開發人員雲端帳戶） 進行任何變更，因此使用現有的測試模式輕鬆實作。模擬測試具有快速的本機開發反覆運算的優點。

但是，模擬器有幾個缺點：

- 它們通常難以設定和複寫，尤其是當它們用作 CI/CD 管道的一部分時。這可能會增加 IT 人員或管理其軟體安裝的開發人員的工作量和維護。
- 被模擬的功能和 API 通常滯後於服務變更，並且可能會阻礙採用新功能，直到新增支援為止。
- 模擬器可能需要支援、更新、錯誤修復或功能對等性增強，這些都是模擬器創作者 (通常是第三方公司) 的責任。

- 依賴模擬器的測試可以在本機提供成功的結果，但由於與 IAM 政策和配額 AWS 等其他方面的互動，通常不會模擬這些測試可能會在雲端中失敗。
- 有些 AWS 服務 沒有可用的模擬器，因此如果您只依賴模擬，您可能沒有應用程式部分令人滿意的測試選項。

在雲端進行測試

雲端中的測試是針對部署到雲端環境而非桌面環境的程式碼執行測試的程序。在雲端進行測試的價值會隨著雲端原生應用程式的開發而增加。例如：

- 您可以針對最新的 APIs 和服務功能在雲端中執行測試，確保您的測試反映最新的傳回值和行為，因為 會 AWS 繼續啟動新的服務和功能。
- 您的測試可以涵蓋 IAM 政策、服務配額、組態以及所有服務。
- 雲端測試環境與生產執行期環境非常相似，因此在雲端執行的測試可能會在環境中進行時獲得最一致的結果。

在雲端中進行測試的缺點包括：

- 部署至雲端環境所花費的時間通常比部署至桌面環境更長。諸如 AWS Serverless Application Model (AWS SAM) Accelerate、AWS Cloud Development Kit (AWS CDK) 監看模式以及 SST 等工具有助於降低雲端部署反覆運算相關的延遲。
- 與使用本機開發環境的本機測試不同，雲端測試可能會產生額外的服務費用。
- 如果您沒有高速網際網路存取，在雲端進行測試可能較不可行。
- 在受管制的產業中，企業安全政策可能會限制開發人員對雲端環境的存取，使得在本機開發工作流程中執行雲端測試變得困難或無法執行。
- 環境界限通常在開發人員環境之共用帳戶的堆疊層級繪製，有時使用命名空間類型策略 (例如使用字首來識別擁有權)。對於生產前環境和生產環境，通常會在帳戶層級設定界限，以便將工作負載隔離出來，使其不受擾鄰問題的影響，支援最低權限安全控制，並保護敏感資料。建立隔離環境的需求可能會對 DevOps 團隊造成額外的負擔，特別是如果他們位於對帳戶和基礎設施具有嚴格控制的企業中。

測試無伺服器應用程式時的挑戰

當您使用模擬器和模擬呼叫在本機電腦上測試無伺服器應用程式時，隨著程式碼在 CI/CD 管道中從一個環境進入到另一個環境，您可能會遇到測試不一致的情況。您在桌面上建立以驗證應用程式商業邏輯的單元測試，可能不會包含或準確代表雲端服務的關鍵層面。無法單獨在本機執行完整的測試。它們需要驗證多個服務之間的許可和組態。

以下各節概述了實作雲端測試策略時可能遇到的挑戰。下列各節概述客戶在嘗試實作雲端測試策略時遇到的挑戰，以及我們實現有效測試涵蓋範圍的最佳實務指引。

範例：建立 Amazon S3 儲存貯體的 Lambda 函數

如果 Lambda 函數的邏輯依賴於建立 Amazon S3 儲存貯體，則完整測試應確認已成功呼叫 Amazon S3 且儲存貯體已成功建立。在模擬物件測試設定中，您可能會模擬成功回應，並可能加入測試案例來應對回應失敗的狀況。在模擬測試案例中，可能會呼叫 `CreateBucket` API，但發出呼叫的身分不會來自擔任角色的 Lambda 服務，而是改用預留位置身分驗證，這通常是您更寬鬆的角色或使用者身分。

如果成功地 (或失敗) 呼叫 Amazon S3，先前討論的模擬和仿真設定很可能會測試 Lambda 函數會做什麼。不過，根據函數的組態，這些測試將無法擷取 Lambda 函數是否能夠成功建立儲存貯體。此組態可能由基礎設施表示為產品和服務的程式碼 (IaC) AWS CloudFormation，例如 AWS SAM， 或 HashiCorp Terraform。一個可能的問題是指派給函數的角色沒有允許 `s3:CreateBucket`動作的連接政策，因此函數在部署到雲端環境時一律會失敗。

範例：處理來自 Amazon SQS 訊息的 Lambda 函數

如果 Amazon Simple Queue Service (Amazon SQS) 佇列是 Lambda 函數的來源，則完整測試應確認訊息放入佇列時已成功叫用 Lambda 函數。模擬測試和模擬測試通常設定為直接執行 Lambda 函數程式碼，並透過傳遞 JSON 事件承載 （或還原序列化物件） 做為函數處理常式的輸入來模擬 Amazon SQS 整合。

模擬 Amazon SQS 整合的本機測試將測試 Amazon SQS 使用指定的承載呼叫 Lambda 函數時，Lambda 函數會執行的動作，但它無法確保 Amazon SQS 會在部署到雲端環境時成功叫用 Lambda 函數。

以下是您在使用 Amazon SQS 和 Lambda 時可能會遇到的一些組態問題範例：

- Amazon SQS 可見性逾時過低，導致原本只要調用一次，變成調用多次。

- Lambda 函數的執行角色不允許從佇列讀取訊息（透過 `sqs:ReceiveMessage`、`sqs:DeleteMessage`或 `sqs:GetQueueAttributes`)。
- 傳遞至 Lambda 函數的範例事件超出 Amazon SQS 訊息大小配額。因此測試無效，因為 Amazon SQS 永遠無法傳送那麼大的訊息。

如上述範例所示，涵蓋商業邏輯但不涵蓋雲端服務之間組態的測試可能會產生不可靠的結果。

用於測試無伺服器應用程式的最佳實務

下列各節概述在測試無伺服器應用程式時實現有效涵蓋範圍的最佳實務。

排定雲端測試的優先順序

對於設計良好的應用程式，您可以使用各種測試技術來滿足各種要求和條件。但是，根據目前的工具，建議您盡可能專注於在雲端中進行測試。雖然在雲端中進行測試可能會造成開發人員延遲、增加成本，有時需要投資額外的 DevOps 控制項，但此技術可提供最可靠、準確且完整的測試涵蓋範圍。

您應該可以存取在其中執行測試的隔離環境。理想情況下，每個開發人員都應擁有專用的 AWS 帳戶 ，以避免在多個使用相同程式碼的開發人員嘗試在具有相同名稱的資源上部署或調用 API 呼叫時，可能發生的任何資源命名問題。這些環境都應設定適當的警示和控制項，以避免出現不必要的支出。例如，您可以限制可建立的資源類型、層級或大小，並在預估成本超過指定閾值時設定電子郵件提醒。

如果您必須 AWS 帳戶 與其他開發人員共用單一 ，自動化測試程序應該為每個開發人員命名唯一的資源。例如，導致 AWS SAM CLI sam 部署或 sam 同步命令的更新指令碼或 TOML 組態檔案，可以自動指定包含本機開發人員使用者名稱的堆疊名稱。 https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/sam-cli-command-reference-sam-sync.html

在雲端進行測試對所有測試階段 (包括單元測試、整合測試和端對端測試) 來說都很有價值。

如有必要，請使用模擬

模擬架構是撰寫快速單元測試的實用工具。當測試需要涵蓋複雜的內部業務邏輯 (例如數學或財務計算或模擬) 時，其尤其有價值。尋找具有大量測試案例或輸入變化的單元測試，其中輸入不會改變對其他雲端服務的模式或呼叫內容。為這些場景建立模擬測試可以改善開發人員反覆運算時間。

使用模擬測試的單元測試所涵蓋的程式碼也應被雲端中的測試所涵蓋。這是必要的，因為模擬仍在開發人員的筆記型電腦或建置機器上執行，而且環境的設定可能與雲端中的環境不同。例如，您的程式碼可能包含使用更多記憶體或花費比 Lambda 設定在特定輸入參數執行時配置更多時間的 AWS Lambda 函數。或者，您的程式碼可能包含未以相同方式 （或完全） 設定的環境變數，而差異可能會導致程式碼的行為不同或失敗。

請勿使用雲端服務的模擬來驗證這些服務整合的適當實作。雖然在測試其他功能時模擬雲端服務是可以接受的，但您應該在雲端測試雲端服務呼叫，以驗證正確的組態和功能實作。

模擬可為單元測試增加價值，尤其是當您經常測試大量案例時。整合測試的這種優勢有所減少，因為實作必要模擬的工作量會隨著連接點的數量而增加。端對端測試不應使用模擬物件，因為這些測試通常會面對無法使用模擬架構輕鬆模擬的狀態和複雜邏輯。

了解模擬測試的權衡

模擬器是特定使用案例的實際選擇。例如，網際網路存取有限、不一致或緩慢的開發團隊可能會發現，模擬測試是在移至雲端環境之前，最可靠的程式碼迭代方式。

對於大多數其他情況，請選擇性地使用模擬器。當您非常依賴模擬器時，在模擬廠商發佈更新以提供功能同位之前，將新的 AWS 服務功能納入您的測試可能會變得困難。模擬器還需要預先和持續投資，才能跨開發系統和建置機器進行設定和組態。此外，許多雲端服務沒有可用的模擬器；選取模擬優先策略可能會排除這些服務的使用，或產生未針對實際服務行為進行良好測試的程式碼和組態。

如果您使用模擬測試， 會盡可能搭配雲端測試來驗證適當的雲端組態，並測試與只能在模擬環境中模擬或模擬之 服務的互動。

模擬測試可以提供單位測試的快速意見回饋，並且根據模擬軟體的功能和行為同位， 可能也支援一些整合和end-to-end測試。

透過自然界限進行範圍測試

隨著無伺服器應用程式跨更多架構元件成長，子系統周圍會出現自然界限，尤其是在遵循單一用途函數和事件驅動解耦等最佳實務時。這些界限可做為有效的測試邊緣，您可以在其中驗證元件之間的合約。

識別架構邊界

在應用程式設計中尋找自然接縫：

- 服務之間，例如將發佈者連線至消費者的 Amazon EventBridge 規則
- 在 API 邊緣，例如前端 Lambda 函數的 Amazon API Gateway 端點
- 圍繞工作流程，例如 AWS Step Functions 協調多個服務
- 在儲存層，例如觸發下游處理的 Amazon DynamoDB 串流

將 Lambda 程式碼與商業邏輯分開

將 Lambda 程式碼與核心商業邏輯隔離，簡化您的測試。您的 Lambda 處理常式應該做為 AWS 執行時間與應用程式邏輯之間的精簡轉接器。它應該擷取並驗證事件資料，然後委派給沒有 Lambda 相依

性的可測試函數。這可讓您的商業邏輯更方便攜帶、更易於推理，且無需模擬 Lambda 物件或設定複雜環境即可直接進行測試。

將界限視為合約

在邊界測試，而不是透過邊界測試。驗證哪些項目跨邊緣，而不需要整個下游系統。這些相同的界限也在生產環境中做為可觀測性勾點。您可以使用 Amazon CloudWatch Logs、 AWS X-Ray traces 和 EventBridge 事件來檢測您測試的架構接縫以進行監控。

使用非同步工作流程的測試機制

無伺服器應用程式通常依賴非同步模式，其中事件觸發處理、訊息流經佇列，以及工作流程跨越多項服務，而無需立即回應。您無法直接叫用函數並檢查傳回值。結果稍後可能會出現在資料庫、日誌串流或其他 服務中。

測試工具正在測試您隨應用程式一起部署的基礎設施，以觀察和驗證此非同步行為。測試機制通常包括：

- 訂閱應用程式產生的相同事件的事件接聽程式
- 可擷取測試結果的儲存機制 （例如 DynamoDB 資料表或 Amazon S3 儲存貯體）
- 測試程式碼中等待預期結果出現的輪詢邏輯

您的測試程式碼會啟動事件、等待工作流程完成，然後查詢測試機制以確認預期的結果發生。

最佳實務如下：

- 為非同步操作定義明確的 SLAs – 建立工作流程應花費的時間，並在測試中將這些工作流程用作輪詢逾時
- 使用唯一識別符進行測試隔離 – 產生每個測試執行的唯一檔案名稱、訊息 IDs 或關聯字符，以防止測試之間的干擾
- 將測試基礎設施與您的應用程式一起部署 – 在您的infrastructure-as-code範本中包含測試設備資源，以便它們在您的應用程式演進時保持同步
- 測試執行後清除測試資料 – 這可防止在雲端環境中累積測試成品

測試機制對於驗證跨多個服務之工作流程的整合測試、驗證完整使用者旅程的end-to-end測試，以及服務透過 EventBridge、Amazon SNS、Amazon SQS 或 Amazon Kinesis 通訊的事件驅動架構來說，是最有價值的。

組織雲端環境以進行開發人員隔離

在雲端進行測試需要彼此隔離的環境。當開發人員共用單一 AWS 帳戶，例如團隊開發帳戶時，請考慮為每個開發人員或功能分支建立個別的應用程式堆疊。這可隔離資源、防止命名衝突，並在測試期間避免配額爭用或雜訊鄰近問題。

使用 AWS Systems Manager 參數存放區或類似工具來管理堆疊特定的組態，例如 API 端點和佇列名稱。為了提高成本效益，在開發人員堆疊之間共用 Amazon Relational Database Service (Amazon RDS) 叢集等昂貴的資源，同時保持每個堆疊隔離輕量型無伺服器資源 （ 例如 Lambda 函數、API Gateway 階段和 DynamoDB 資料表 ）。

在受管制的產業中，企業安全政策可能會限制開發人員對雲端環境的存取，使得在本機開發工作流程中難以執行雲端測試。在這些情況下，模擬測試可以填補本機模擬測試和完整雲端驗證之間的差距，但應該在存取允許時補充雲端測試。

加速回饋迴圈

在雲端進行測試時，請使用工具和技術來加速開發回饋迴圈。例如，使用AWS SAM 加速和 AWS CDK 監看模式可縮短將程式碼修改推送至雲端環境所需的時間。GitHub 無伺服器測試範例儲存庫 中的範例會探究其中一些技術。

我們也建議您在開發期間儘早從本機機器建立和測試雲端資源，而不只是在簽入來源控制之後。這種作法可在制定解決方案時加快探索和實驗的速度。此外，從開發電腦中自動化部署可協助您更快發現雲端組態問題，並減少更新和核准來源控製修改所浪費的人力。

常見問答集

我有一個 Lambda 函數，它可以執行計算並傳回結果，無需呼叫任何其他服務。我真的需要在雲端進行測試嗎？

是。 AWS Lambda 函數具有可能變更測試結果的組態參數。所有 Lambda 函數程式碼都依賴逾時和記憶體設定，如果未正確設定，可能會導致函數失敗。Lambda 政策也會向 Amazon CloudWatch 啟用標準輸出記錄。即使您的程式碼未直接呼叫 CloudWatch，也需要許可才能啟用記錄，而且無法準確模擬或模擬該許可。

雲端中的測試如何協助進行單元測試？ 如果它位於雲端並連接到其他資源， 不是整合測試嗎？

我們將單元測試定義為獨立在架構元件上運行的測試。此定義不一定會排除使用服務呼叫或其他網路通訊。

許多無伺服器應用程式確實都具有可獨立進行測試的架構元件 (即使在雲端也是如此)。基本範例是 Lambda 函數，會接受一些輸入、解譯它，並將訊息傳送至 Amazon Simple Queue Service (Amazon SQS) 佇列。此類函數的單元測試可能會測試輸入值是否會導致某些值出現在佇列訊息之中。考慮使用安排、動作、宣告模式撰寫的測試：

- 排列 – 配置資源 （ 接收訊息的佇列，以及測試中的函數 ）。
- 動作 – 呼叫測試中的 函數。
- Assert – 擷取函數傳送的訊息，並驗證輸出。

模擬物件測試方法包括使用正在進行的模擬物件來模擬佇列，以及建立含有 Lambda 函數程式碼的類別或模組的進行中執行個體。在宣告階段，佇列的訊息會從模擬物件擷取。

在雲端方法中，測試會針對測試目的建立 Amazon SQS 佇列，並透過設定為將隔離的 Amazon SQS 佇列作為輸出目的地使用的環境變數來部署 Lambda 函數。執行 Lambda 函數後，測試會從 Amazon SQS 佇列擷取訊息。

雲端測試會執行相同的程式碼、宣告相同的行為，並驗證應用程式的功能正確性。不過，它還可以驗證 Lambda 函數 AWS Identity and Access Management 的下列設定：(IAM) 角色、IAM 政策，以及函數的逾時和記憶體設定。

後續步驟和資源

使用以下資源進行進一步閱讀和其他具體範例。

實作範例

GitHub 上的 無伺服器測試範例儲存庫 包含遵循本指南所述模式和最佳實務的具體測試範例。儲存庫包含前幾節所述之模擬物件、模擬和雲端測試程序的範例程式碼和引導式逐步解說。使用此儲存庫來取得來自 的最新無伺服器測試指引 AWS。

深入閱讀

請造訪 Serverless Land 以存取最新的部落格、影片和無 AWS 伺服器技術訓練。

參考

- 使用 Accelerate 加速無伺服器開發 AWS SAM(AWS 部落格文章）
- 使用 CDK Watch 提高開發速度 (AWS 部落格文章）
- 模擬服務與 AWS Step Functions Local 整合 (AWS 部落格文章）
- 測試無伺服器應用程式入門 (AWS 部落格文章）
- 使用 VS Code IDE 中的 LocalStack 整合加速無伺服器測試 (AWS 部落格文章）
- 使用 本機偵錯函數 AWS SAM (AWS 文件）

工具

- AWS SAM – 測試和偵錯無伺服器應用程式
- AWS SAM – 與自動化測試整合
- AWS Lambda – 在主控台中測試 Lambda 函數
- LocalStack 雲端模擬器 – 使用 LocalStack 增強無伺服器應用程式的LocalStack測試體驗

貢獻者

編寫

- Dan Fox， AWS
- Leslie Raj， AWS
- Rohan Mehta， AWS
- 羅布希爾 AWS

檢閱

- Brian Krygsman， AWS

技術寫入

- Lilly AbouHarb， 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
更新	我們新增了在架構和程式碼邊界進行測試、非同步工作流程的測試機制，以及開發人員環境隔離的指引。我們也更新了模擬測試建議。	2026 年 3 月 18 日
初次出版	—	2022 年 12 月 9 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：將內部部署 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，但只有來源資料庫會在資料複寫至目標資料庫時處理來自連線應用程式的交易。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

永久刪除資料集中個人資訊的程序。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

經常用於經常性問題的解決方案，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

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、低延遲網路連線能力。

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，例如在網際網路上編製資訊索引的 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，並透過核准的程序，讓使用者快速取得他們通常無權存取 AWS 帳戶 之 的存取權。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) 執行試驗，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

部落格文章中的 Stephen Orban 定義了這些階段： AWS 雲端 企業策略部落格上的邁向雲端優先之旅和採用階段。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

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

資料主體

正在收集和處理其資料的個人。

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，以管理組織的帳戶和管理該服務的許可。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。

deployment

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並排定優先順序。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

一種使用頻繁和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給 LLM。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。對於需要特定格式、推理或網域知識的任務，少量的提示非常有效。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 CSPM、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、修補或修改現有的基礎設施。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

由 Klaus Schwab 於 2016 年推出的術語，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 （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、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。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 建議使用不可變基礎設施做為最佳實務。

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

請參閱開放程序通訊 - 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)

問題和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。在製造中，OT 和資訊技術 (IT) 系統的整合是工業 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)

當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。

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

以預留位置值取代資料集中個人識別符的程序。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 個 R。

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)

服務中斷與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 Rs。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

請參閱 7 個 R。

Replatform

請參閱 7 個 R。

回購

請參閱 7 個 R。

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

矩陣，定義所有參與遷移活動和雲端操作之各方的角色和責任。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 個 R。

淘汰

請參閱 7 Rs。

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。此功能會啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS 管理主控台 或呼叫 AWS API 操作，而不必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。

斯卡達

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 秘密中的內容？ Secrets Manager 文件中的 。

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。

未區分的任務

也稱為繁重工作，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

危及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。