中的效能工程分階段方法 AWS 雲端

AWS 方案指引

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

AWS 方案指引: 中的效能工程分階段方法 AWS 雲端

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
什麼是性能工程？ 1
為什麼要使用效能工程？ 1
效能工程支柱 2
產生測試資料 3
測試資料產生工具 4
測試可觀測性 5
日誌 6
監控 9
追蹤 12
測試自動化 14
測試自動化工具 15
測試報告 16
標準化錄音 16
效能支柱範例 17
資源 19
貢獻者 21
文件歷史紀錄 22
詞彙表 23
..... 23
A 23
B 26
C 27
D 30
E 33
F 35
G 36
H 37
I 38
L 40
M 41
O 45
P 47
Q 49

R 49
S 52
T 55
U 56
V 57
W 57
Z 58
..... lix

性能工程的分階段方法 AWS 雲端

Amazon Web Services（貢獻者）

2024 年 4 月 (文件歷史記錄)

本指南概述了針對在 Amazon Web Services 上執行的應用程式工作負載規劃、建置和啟用效能工程的最佳實務 (AWS)。它為性能工程奠定了四個支柱，並提出了不同的方法來滿足應用程序的性能需求。本指南針對每個支柱列出了用於設置性能測試和測試環境的工具和解決方案。

什麼是性能工程？

效能工程包含系統開發生命週期期間所套用的技術，以確保符合非功能性效能需求 (例如輸送量、延遲或記憶體使用量)。

在性能測試開始之前，您需要設置性能環境。典型的性能環境具有以下幾個支柱：

- 產生測試資料
- 測試可觀測性
- 測試自動化
- 測試報告

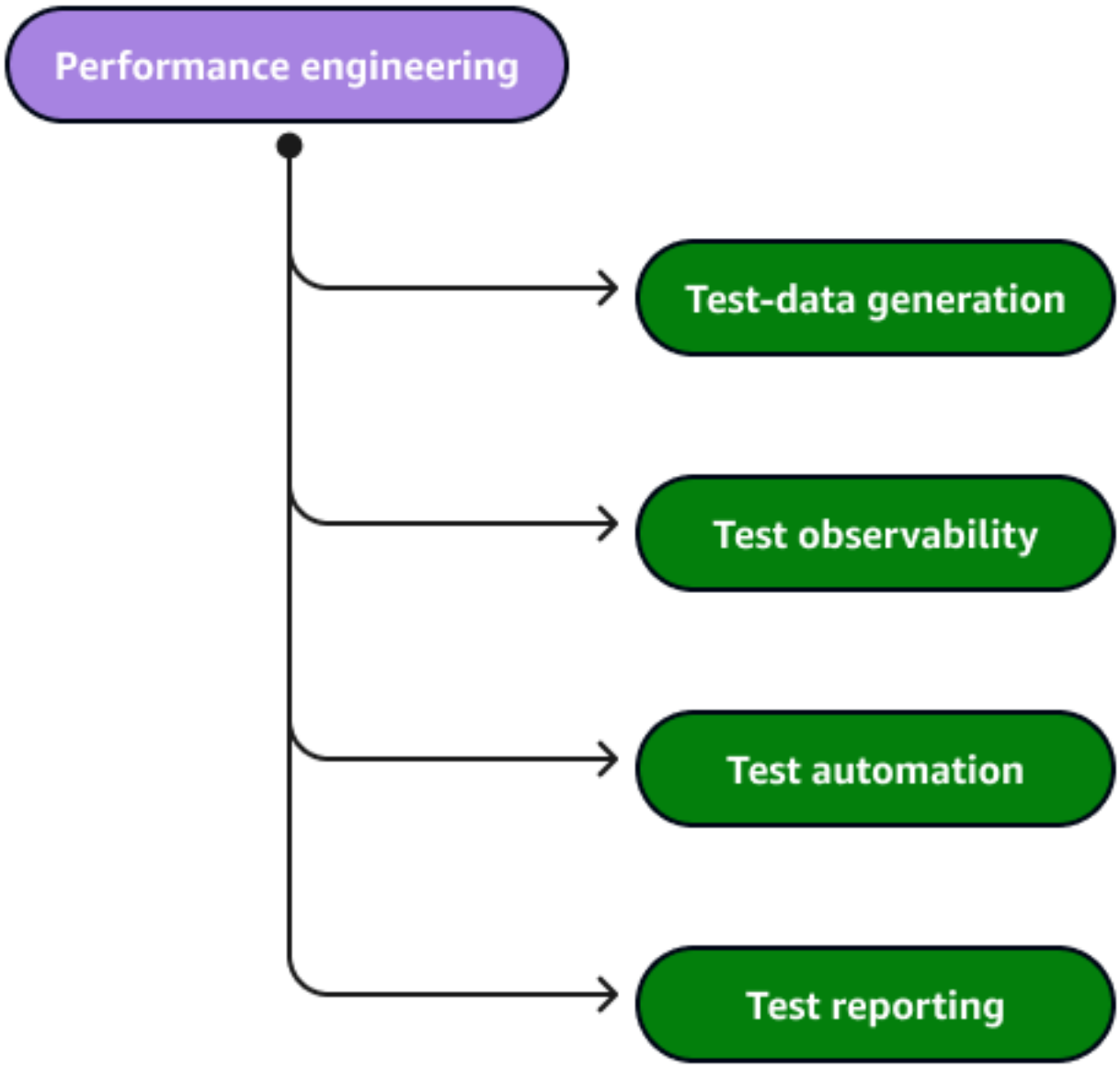
為什麼要使用效能工程？

性能工程是從設計階段開始就持續優化應用程序性能的過程。它通過避免在開發週期的後期階段重工和代碼重構為業務帶來巨大的價值。在設計階段開始效能工程會導致執行效能較佳的應用程式，因為效能可以納入設計。效能工程需要系統架構師、開發人員和品質保證的積極參與。 DevOps

性能工程的支柱

若要實現效能工程思維，在為應用程式設定效能工程的同時，建立堅實的基礎是非常重要的。性能工程需要建立四個主要支柱：

- 測試資料產生 — 效能工程師設定工具來產生測試資料。
- 測試可觀測性 — 效能工程師會設定可觀察性環境，以確保可以記錄和追蹤效能執行，以及監控處理負載的資源。
- 測試自動化 — 效能工程師開發自動化測試，使用 Apache JMeter 或 ghz 等工具來模擬使用者流量和系統負載。
- 測試報告-收集有關每次測試運行的配置以及性能結果的數據。資料可讓組態變更與效能相互關聯，並提供寶貴的見解。



結合這些支柱將鼓勵從設計的初始階段開始實現績效思維。這將有助於避免在開發和測試的後期階段對應用程序或環境進行更改。

產生測試資料

測試數據生成涉及生成和維護用於運行性能測試用例的大量數據。這種生成的數據充當輸入到測試用例，以便應用程序可以在一組不同的數據進行測試。

通常，生成測試數據是一個複雜的過程。不過，使用建立不當的資料集可能會導致生產環境中無法預期的應用程式行為。用於性能測試的測試數據生成與傳統的測試數據生成方法不同。它需要真實案例，而

且大多數客戶都希望使用類似於其實際生產資料的資料來測試其工作負載。生成的測試數據通常還需要在每次測試運行後重置或刷新為其原始狀態，這增加了時間和精力。

產生測試資料包括下列主要考量事項：

- 準確性 — 數據的準確性在測試的各個方面都很重要。不準確的資料會產生不正確的 例如，當產生信用卡交易時，該交易不應該是 future 的日期。
- 有效性 — 資料對於使用案例應該是有效的。例如，在測試信用卡交易時，建議每位使用者每天產生 10,000 筆交易，因為這與有效的使用案例有很大差異。
- 自動化 — 測試資料產生的自動化可以帶來耗時效益。它還導致有效的測試自動化。手動生成測試數據可能會產生相對於質量和時間的要求的後果。

  根據用例，可以採用不同的機制，如下所示：

  - API 驅動 — 在這種情況下，開發人員提供了測試數據生成 API，測試人員可以使用該 API 來生成數據。使用測試工具，如 JMeter 的，測試人員可以使用業務 API 擴展數據生成。例如，如果您有一個 API 來添加用戶，則可以使用相同的 API 來創建數百個具有不同配置文件的用戶。同樣地，您可以透過呼叫刪除 API 作業來刪除使用者。對於複雜的工作流程應用程式，開發人員可以提供可跨不同元件產生資料集的複合 API。使用這種方法，測試人員可以編寫自動化生成和根據自己的要求刪除數據集。

    但是，如果系統很複雜，或者每次調用的 API 響應時間很長，則設置和拆除數據可能需要很長時間。
  - SQL 陳述式驅動 — 另一種方法是使用後端 SQL 陳述式來產生大量的資料。開發人員可以提供以範本為基礎的 SQL 陳述式來產生測試資料。測試人員可以使用語句來填充數據，或者他們可以在這些語句之上創建包裝腳本以自動生成測試數據。使用這種方法，測試人員可以填充和拆除數據非常快，如果數據需要在測試完成後被重置。但是，這種方法需要直接訪問應用程序的數據庫，這在典型的安全環境中可能是不可能的。此外，無效的查詢可能會導致資料填入不正確，進而產生扭曲的結果。開發人員也必須持續更新應用程式程式碼中的 SQL 陳述式，以反映應用程式隨著時間的變更。

測試資料產生工具

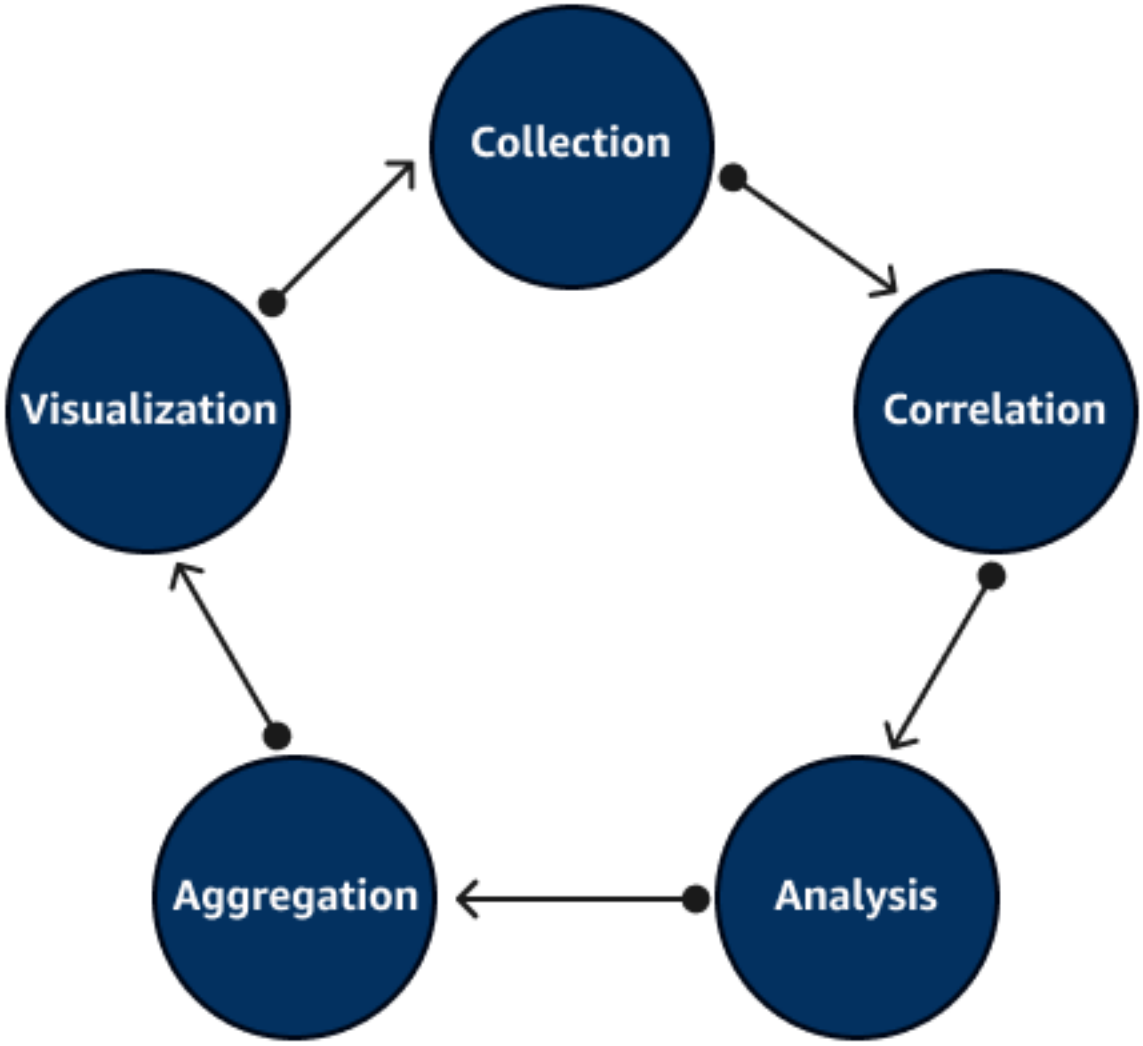
AWS 提供可用於產生測試資料的原生自訂工具：

- 亞馬遜 Kinesis 資料產生器 — Amazon Kinesis 資料產生器 (KDG) 可簡化產生資料並將其傳送至 Amazon Kinesis 的任務。該工具提供了直接在瀏覽器中運行的用戶友好的用戶界面。如需詳細資訊和參考實作，請參閱使用新的 Amazon Kinesis 資料產生器測試您的串流資料解決方案部落格文章。

- AWS Glue 測試資料產生器 — AWS Glue 測試資料產生器為使用 AWS Glue PySpark 無伺服器工作產生測試資料提供可設定的架構。必要的測試資料說明可透過 YAML 設定檔完全設定。有關詳細信息和參考實現，請參閱AWS Glue 測試數據生成器 GitHub 存儲庫。

測試可觀測性

測試可觀測性支援在效能測試執行期間，在網路、基礎結構和應用程式中收集、關聯、彙總和分析遙測。您可以獲得系統行為、效能和健康狀態的完整洞察。這些見解可協助您更快地偵測、調查及修復問題。透過新增人工智慧和機器學習，您可以主動對問題做出反應、預測和預防。



可觀測性依賴於記錄、監視和追蹤。成功實作這些活動的責任涵蓋應用程式和基礎結構團隊。

在設計階段開始時，應用程式團隊應瞭解其可觀察性堆疊的目前狀態，包括記錄、監視和追蹤。然後，他們可以選擇更順利地集成到可觀察性堆棧中的工具。

同樣地，基礎架構團隊負責管理和擴充可觀測性基礎結構。

考慮有關測試可觀察性以下幾個方面：

- 應用程式記錄和追蹤的可用性
- 記錄檔和追蹤的關聯性
- 節點、容器和應用程式指標的可用性
- 根據需求設置和更新可觀察性基礎設施的自動化
- 能夠視覺化遙測
- 擴展可觀測性基礎設施

日誌

記錄是保持有關發生在系統中事件的數據的過程。記錄檔可能包含有關目前作業的問題、錯誤或資訊。記錄檔可分為不同類型，如下所示：

- 事件日誌
- 伺服器記錄
- 系統日誌
- 授權和存取記錄
- 稽核日誌

開發人員可以在日誌中搜索特定的錯誤代碼或模式，根據特定字段對其進行過濾，或將其安全存檔以備 future 分析。日誌可幫助開發人員針對性能問題執行根本原因分析，並在系統組件之間進行關聯。

建立有效的記錄解決方案需要應用程式和基礎架構團隊之間的密切協調 除非有可調整的記錄基礎結構支援使用案例，例如剖析、篩選、緩衝和記錄檔的關聯性，否則應用程式記錄檔並無用處。您可以簡化常見的使用案例，例如產生關聯識別碼、業務關鍵方法的記錄執行時間，以及定義記錄模式。

應用團隊

應用程式開發人員必須確保產生的記錄遵循記錄最佳做法。最佳做法包括：

- 產生關聯 ID 以追蹤唯一要求
- 記錄業務關鍵方法所花費的時間
- 在適當的記錄層級記錄

- 共享一個通用的日誌庫

當您設計與不同微服務互動的應用程式時，請使用這些記錄設計原則來簡化後端的篩選和記錄擷取作業。

產生關聯 ID 以追蹤唯一要求

當應用程式收到要求時，它可以檢查相互關聯 ID 是否已存在於標頭中。如果 ID 不存在，應用程序應該生成一個 ID。例如，應用程式負載平衡器會新增名為的標頭`X-Amzn-Trace-Id`。應用程式可以使用標頭，將負載平衡器的要求與應用程式相關聯。同樣地，`traceId`如果呼叫相依的微服務，應用程式應該注入，以便要求流程中不同元件產生的記錄檔是相互關聯的。

記錄業務關鍵方法所花費的時間

當應用程序接收到一個請求，它與不同的組件進行交互。應用程式應以定義的模式記錄業務關鍵方法所花費的時間。這可以更容易地解析後端的日誌。它還可以幫助您從日誌中生成有用的見解。您可以使用諸如面向方面的編程（AOP）之類的方法來生成這樣的日誌，以便您可以將日誌記錄問題與業務邏輯分開。

在適當的記錄層級記錄

應用程序應該寫入具有有用信息量的日誌。使用記錄層級，依嚴重性對事件進行分類。例如，對需要調查的重要事件使用`WARNING`和`ERROR`級別。用`INFO``DEBUG`於詳細追蹤和大量事件。設定記錄處理常式，以僅擷取生產中所需的層級。在`INFO`層級產生太多記錄是沒有幫助的，它會增加後端基礎結構的壓力。 `DEBUG`日誌記錄可能很有用，但應謹慎使用。使用`DEBUG`記錄檔可能會產生大量資料，因此不建議在效能測試環境中使用。

共享一個通用的日誌庫

應用程序團隊應該使用一個通用的日誌庫，例如 適用於 Java 的 AWS SDK，具有預定義的通用日誌記錄模式，開發人員可以用作其項目中的依賴關係。

基建團隊

DevOps 工程師在後端過濾和擷取日誌時，可以使用下列日誌設計原則來減少工作量。基礎架構小組必須設定並支援下列資源。

日誌代理

日誌代理（日誌寄件人）是一種從一個位置讀取日誌並將其發送到另一個位置的程序。記錄代理程式可用來讀取儲存在電腦上的記錄檔，並將記錄事件上傳至後端以進行集中化。

記錄檔是非結構化資料，必須先結構化，才能從中取得有意義的見解。記錄檔代理程式會使用剖析器來讀取記錄陳述式，並擷取相關欄位，例如時間戳記、記錄層級和服務名稱，並將該資料結構成 JSON 格式。在邊緣具有輕量級日誌代理程式很有用，因為它會導致較少的資源使用率。記錄代理程式可以直接推送至後端，也可以使用將資料推送至後端的中繼記錄轉寄站。使用記錄轉送器會從來源的記錄代理程式卸載工作。

日誌解析器

日誌解析器將非結構化日誌轉換為結構化日誌。記錄代理程式剖析器也會透過新增中繼資料來豐富記錄檔。數據的數據解析可以在源（應用程序端）完成，也可以集中完成。用於存儲日誌的結構描述應該是可擴展的，以便您可以添加新字段。我們建議使用標準日誌格式，例如 JSON。不過，在某些情況下，記錄檔必須轉換成 JSON 格式，才能更好地搜尋。撰寫正確的剖析器運算式可以有效率地轉換

日誌後端

Logs 後端服務會從各種來源收集、擷取和視覺化記錄資料。記錄代理程式可以直接寫入後端或使用中介記錄轉寄站。在性能測試時，請務必存儲日誌，以便日後可以搜索它們。為每個應用程序分別將日誌存儲在後端。例如，針對應用程式使用專用索引，並使用索引模式來搜尋散佈在不同相關應用程式的記錄檔。我們建議您儲存至少 7 天的資料以進行記錄搜尋。但是，將資料儲存較長的時間可能會導致不必要的儲存成本。因為在效能測試期間會產生大量的記錄檔，因此記錄基礎結構要擴展和調整記錄後端的大小相當重要。

日誌可視化

若要從應用程式記錄中取得有意義且可行的洞察，請使用專用的視覺化工具來處理原始記錄資料並將其轉換為圖形表示。圖表、圖形和儀表板等視覺效果有助於發現趨勢、模式和異常情況，這些趨勢、模式和異常情況在查看原始日誌時可能不明顯。

使用視覺化工具的主要優點包括能夠跨多個系統和應用程式關聯資料，以識別相依性和瓶頸。互動式儀表板支援以不同粒度層級向下鑽研資料，以疑難排解問題或發現使用趨勢。專業的資料視覺化平台提供諸如分析、警示和資料共用等功能，可增強監控和分析。

透過在應用程式記錄上運用資料視覺化的強大功能，開發和營運團隊可以深入瞭解系統和應用程式效能。所得出的見解可用於多種用途，包括最佳化效率、改善使用者體驗、增強安全性和容量規劃。最終結果是針對各種利益相關者量身定制的儀表板，提供將日誌數據匯總為可行且有見地的信息的 at-a-glance 視圖。

自動化記錄基礎架構

因為不同的應用程式有不同的需求，所以記錄基礎結構的安裝和作業自動化是非常重要的。使用基礎結構即程式碼 (IaC) 工具來佈建記錄基礎結構的後端。然後，您可以將記錄基礎結構佈建為共用服務，或作為特定應用程式的獨立自訂部署佈建。

我們建議開發人員使用持續交付 (CD) 管道來自動執行下列作業：

- 視需求部署記錄基礎結構，並在不需要時將其拆除。
- 跨不同目標部署記錄代理程式。
- 部署日誌解析器和轉發器配置。
- 部署應用程式儀表板

記錄工具

AWS 提供原生記錄、警示和儀表板服務。以下是流行 AWS 服務 的日誌記錄資源：

- Amazon Ser OpenSearch vice 可協助組織從各種來源收集、擷取和視覺化日誌資料。如需詳細資訊，請參閱集中式記錄 OpenSearch。
- Amazon CloudWatch 代理程式和 AWS Fluent Bit 是最受歡迎的日誌代理程式 AWS。如需將 CloudWatch 代理程式與 Amazon CloudWatch 日誌洞見搭配使用的相關資訊，請參閱部落格文章：利用 Amazon 日誌洞察簡化 Apache 伺服器 CloudWatch 記錄。如 AWS 需 Fluent Bit 參考實作，請參閱部落格文章使用 Fluent Bit 集中式容器記錄。

監控

監控是收集不同指標 (例如 CPU 和記憶體) 的程序，並將它們儲存在時間序列資料庫中，例如 Prometheus 的 Amazon 受管服務。監控系統可以是基於推送或拉取。在以推送為基礎的系統中，來源會定期將量度推送至時間序列資料庫。在基於拉動的系統中，抓取工具會從各種來源抓取指標並將其存儲在時間序列數據庫中。開發人員可以分析指標，過濾指標，並繪製它們隨著時間的推移以可視化性能。成功實作監控可分為兩大領域：應用程式和基礎架構。

對於應用程式開發人員而言，下列指標很重要：

- 延遲 — 接收回應所花費的時間
- 要求傳輸量 — 每秒處理的要求總數
- 要求錯誤率 — 錯誤總數

擷取涉及商業交易之每個資源 (例如應用程式容器、資料庫) 的資源使用率、飽和度和錯誤計數。例如，監視 CPU 使用率時，您可以追蹤效能測試執行期間的平均 CPU 使用率、平均負載和尖峰負載。當資源在 stress 測試期間達到飽和度時，但在效能執行期間可能無法達到飽和度。

指標

應用程式可以使用不同的致動器 (例如彈簧引導致動器) 來監控其應用。這些生產級程式庫通常會公開 REST 端點，用於監視執行中應用程式的相關資訊。這些程式庫可以監視基礎結構、應用程式平台和其他資源。如果任何默認指標不符合要求，則開發人員必須實現自定義指標。自訂指標可協助追蹤無法透過預設實作資料追蹤的業務關鍵績效指標 (KPI)。例如，您可能想要跟踪業務操作，例如第三方 API 集成延遲或完成的交易總數。

基數

基數是指量度的唯一時間序列數目。量度會標示為提供其他資訊。例如，追蹤特定 API 要求計數的 REST 型應用程式表示基數為 1。如果您新增使用者標籤來識別每位使用者的要求計數，則基數會隨使用者數量成比例增加。透過新增建立基數的標籤，您可以依不同群組對量度進行切片和切塊。請務必針對正確的使用案例使用正確的標籤，因為基數會增加後端監視時間序列資料庫中的度量序列數目。

解析度

在典型的監視設置中，監視應用程序配置為定期從應用程序中抓取指標。抓取的週期性定義了監視數據的粒度。由於有更多可用的資料點，因此以較短的間隔收集的指標往往會提供更準確的效能檢視。不過，時間序列資料庫的負載會隨著儲存更多項目而增加。通常 60 秒的粒度是標準解析度，1 秒是高解析度。

DevOps 團隊

應用程式開發人員經常要求 DevOps 工程師設定監視環境，以視覺化基礎架構和應用程式的指標。工程 DevOps 師必須設置一個可擴展的環境，並支持應用程序開發人員使用的數據可視化工具。這涉及從不同來源抓取監控數據，並將數據發送到中央時間序列數據庫，例如 Prometheus 的 Amazon 託管服務。

監控後端

監控後端服務支援指標資料的收集、儲存、查詢和視覺化。它通常是一個時間序列數據庫，例如 Amazon Prometheus 或 InfluxDB 的管理服務。 InfluxData 使用服務探索機制，監視收集器可以從不同來源收集指標並加以儲存。雖然性能測試，它存儲的指標數據，以便它可以在以後的時間被搜索是非常重要的。我們建議您為指標儲存至少 15 天的資料。但是，將指標存儲更長的持續時間並不會增加顯著的好處，並且會導致不必要的存儲成本。因為效能測試可以產生大量的指標，因此在提供快速查詢效能的同時擴展指標基礎結構非常重要。監視後端服務提供查詢語言，可用來檢視指標資料。

視覺效果

提供可顯示應用程式資料的視覺化工具，以提供有意義的見解。工程 DevOps 師和應用程式開發人員應該學習監視後端的查詢語言，並密切合作以產生可重複使用的儀表板範本。在儀表板上，包括延遲和錯誤，同時還顯示基礎結構和應用程式資源的資源使用率和飽和度。

自動化監控基礎架構

與記錄類似，自動化監控基礎結構的安裝和操作非常重要，以便您可以滿足不同應用程式的不同需求。使用 IaC 工具佈建監視基礎結構的後端。然後，您可以將監視基礎結構佈建為共用服務，或作為特定應用程式的獨立自訂部署佈建。

使用 CD 管線自動執行下列作業：

- 根據需求部署監控基礎架構，並在不需要時將其拆除。
- 更新監視組態以篩選或彙總指標。
- 部署應用程式儀表板

監控工具

適用於 Prometheus 的 Amazon 受管服務是一種與 Prometheus 相容的監控服務，適用於容器基礎設施和容器的應用程式指標，可用來安全地大規模監控容器環境。如需詳細資訊，請參閱適用於 Prometheus 的 Amazon 受管服務入門的部落格文章。

Amazon CloudWatch 提供全棧監控。 AWS CloudWatch 同時支援 AWS 原生和開放原始碼解決方案，讓您隨時瞭解技術堆疊中發生的情況。

原生 AWS 工具包括下列項目：

- Amazon CloudWatch 儀表
- CloudWatch 容器洞見
- CloudWatch 度量
- CloudWatch 警報

Amazon CloudWatch 提供專門建置的功能，可解決特定使用案例，例如透過容器洞察進行容 CloudWatch 器監控。這些功能內建於其中， CloudWatch 因此您可以設定記錄、指標收集和監控。

針對您的容器化應用程式和微服務，請使用容器深入解析來收集、彙總和摘要指標和記錄。容器見解適用於 Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service

(Amazon EKS) 以及 Amazon Elastic Compute Cloud (Amazon EC2) 上的 Kubernetes 平台。容器深入解析會以內嵌指標格式收集資料做為效能記錄事件。這些效能記錄事件項目使用支援高基數資料擷取和大規模儲存的結構化 JSON 結構描述。

有關使 AWS 用 Amazon EKS 實作容器見解的資訊，請參閱部落格文章介紹使用發行版的 Amazon EKS Fargate 的 Amazon CloudWatch 容器見解。 OpenTelemetry

追蹤

跟踪涉及專門使用有關程序的進程日誌信息。記錄中的深入分析可協助工程師偵錯個別交易並找出瓶頸。追蹤可以自動啟用，也可以使用手動檢測來啟用。

由於應用程式與不同的服務整合，因此請務必識別應用程式及其基礎服務的執行方式。追蹤可使用軌跡和跨距。跟踪是完整的請求過程，每個跟踪由跨度組成。跨度是一個標記的時間間隔，是一個系統的各個組件或服務中的活動。跟踪提供了對應用程序發出請求時會發生什麼情況的大圖片。

應用團隊

應用程式開發人員會傳送傳入和輸出要求的追蹤資料，以及應用程式內其他事件，以及每個要求的中繼資料，藉此檢測其應用程式 若要產生追蹤，必須檢測應用程式才能產生追蹤。儀器可以是自動或手動的。

自動儀表

您可以使用自動檢測從應用程式收集遙測，而不需要修改原始程式碼。自動檢測代理程式可以產生應用程式或服務的應用程式追蹤。一般而言，您可以使用組態變更來新增代理程式或其他機制。

庫檢測涉及進行最少的應用程序代碼更改以添加預構建的檢測。檢測以特定程式庫或架構為目標，例如 AWS SDK、Apache HTTP 用戶端或 SQL 用戶端。

手動儀器

在這種方法中，應用程式開發人員會將檢測程式碼新增至應用程式，在他們想要收集追蹤資訊的每個位置。例如，使用方面導向程式設計 (AOP) 來收集追蹤資料。 AWS X-Ray 開發人員可以使用 SDK 來檢測其應用程式。

抽樣

追蹤資料通常是在大量產生的。重要的是要有一種機制來確定是否應該導出跟踪數據。採樣是確定應導出哪些數據的過程。這樣做通常是為了節省成本。透過自訂取樣規則，您可以控制記錄的資料量。您也可以在不變更和重新部署程式碼的情況下變更取樣行為。控制取樣率以產生適當的軌跡數量非常重要。

應用程序開發人員可以通過將元數據添加為鍵值對來註釋跟踪。註釋豐富了跟踪，並幫助在後端優化過濾。

DevOps 團隊

DevOps 工程師經常被要求為應用程式開發人員設定追蹤環境，以視覺化基礎架構和應用程式的追蹤。追蹤環境設定包括從不同來源收集追蹤資料，並將其傳送至中央存放區以進行視覺化。

追蹤後端

追蹤後端是一種服務，例如收集 AWS X-Ray 應用程式所提供之要求的相關資料。它提供的工具可讓您用來檢視、篩選和深入瞭解該資料，以識別問題和最佳化的機會。對於應用程式的任何追蹤要求，您可以查看有關要求和回應的詳細資訊，以及應用程式對下游 AWS 資源、微服務、資料庫和 Web API 進行的其他呼叫的詳細資訊。

自動化追蹤

由於不同的應用程式有不同的追蹤需求，因此自動化追蹤基礎結構的組態與作業非常重要。使用 IaC 工具來佈建追蹤基礎結構的後端。

使用 CD 管線自動執行下列作業：

- 視需求部署追蹤基礎結構，並在不需要時拆除它。
- 跨應用程式部署追蹤組態。

追蹤工具

AWS 為追蹤及其相關視覺效果提供下列服務：

- AWS X-Ray 除了來自應用程式使用的 AWS 服務 (已與 X-Ray 整合) 的追蹤外，還會接收來自應用程式的追蹤。有數個 SDK、代理程式和工具可用來檢測您的應用程式以進行 X-Ray 追蹤。如需詳細資訊，請參閱 AWS X-Ray 文件。

  開發人員也可以使用 AWS X-Ray SDK 將追蹤傳送至 X-Ray。 AWS X-Ray 提供Go、Java、Node.js Python、.NET 和Ruby. 每個 X-Ray SDK 提供以下內容：

  - 攔截程式，可新增至您的程式碼以追蹤傳入的 HTTP 請求
  - 用於檢測應用程式用來呼叫其他 AWS 服務的 AWS SDK 用戶端的用戶端處理常式
  - 用於檢測對其他內部和外部 HTTP Web 服務的呼叫的 HTTP 用戶端

X-Ray SDK 也支援檢測對 SQL 資料庫的呼叫、自動 AWS SDK 用戶端檢測和其他功能。SDK 不會將追蹤資料直接傳送至 X-Ray，而是將 JSON 區段文件傳送至偵聽 UDP 流量的精靈程序。X-Ray 守護程序緩衝隊列中的段，並將其批量上傳到 X-Ray。如需有關使用 X-Ray SDK 檢測應用程式的詳細資訊，請參閱 X-Ray 文件。

- Amazon Ser OpenSearch vice 是用於執行和擴展 OpenSearch 叢集的 AWS 受管服務，可用於集中存放日誌、指標和追蹤。可觀測性外掛程式為從常見資料來源收集和監控指標、日誌和跟蹤提供了統一的體驗。集中在一個位置收集和監控資料，可為您的整個基礎架構提供完整堆疊的 end-to-end 可觀察性。如需實作資訊，請參閱OpenSearch 服務文件。
- AWS 適用於 OpenTelemetry (ADOT) 的 AWS 發行版是以雲端原生運算基礎 (CNCF) 專案為基礎的發行版。 OpenTelemetry ADOT 目前包括對 Java 和 Python 的自動檢測支持。此外，ADOT 透過 AD OT 託管 Lambda 層Java，支援使用 Node.js 和執行階段對 AWS Lambda 函數及其下游請求進Python行自動檢測。開發人員可以使用 ADOT 收集器將追蹤傳送到包括 AWS X-Ray Amazon OpenSearch 服務在內的不同後端。

  如需如何使用 ADOT SDK 檢測應用程式的參考範例，請參閱文件。如需如何使用 ADOT 開發套件將資料傳送至 Amazon OpenSearch 服務的參考範例，請參閱OpenSearch 服務文件。

  有關如何檢測在 Amazon EKS 上運行的應用程序的參考示例，請參閱博客文章使用適用於 AWS Distro 的 Amazon EKS 附加組件的指標和跟踪收集。 OpenTelemetry

測試自動化

使用專門架構和工具進行自動化測試，可減少人為干預並最大化品質。自動化性能測試是從自動化測試，如單元測試和集成測試沒有什麼不同。

使用不同階段的 DevOps 管道進行性能測試。

測試自動化管道的五個階段為：

1. 設定 — 針對此階段使用「產生測試資料」一節中所述的測試資料方法。產生逼真的測試資料對於取得有效的測試結果至關重要。您必須仔細建立涵蓋各種使用案例並與即時生產資料密切相符的各種測試資料。在執行全面效能測試之前，您可能需要執行初始試用測試，以驗證測試指令碼、環境和監視工具。

2. 測試工具 — 若要進行效能測試，請選取適當的負載測試工具，例如 JMeter 或 ghz。在模擬真實世界的使用者負載方面，考慮最適合您的業務需求的方法。
3. 測試運行 — 建立測試工具和環境後，在一系列預期的用戶負載和持續時間內運行 end-to-end 性能測試。在整個測試過程中，請密切監控所測試系統的健康狀況。這通常是長時間執行的階段。監控自動測試失效的錯誤率，如果錯誤過多，則停止測試。

   負載測試工具提供資源使用率、回應時間和潛在瓶頸的深入解析。
4. 測試報告 — 收集測試結果以及應用程序和測試配置。自動收集應用程式組態、測試組態和結果，有助於記錄效能測試相關資料並集中儲存。集中維護效能資料有助於提供良好的見解，並支援以程式設計方式為企業定義成功標準。
5. 清理 — 完成效能測試回合後，請重設測試環境和資料，以便為後續執行做好準備。首先，您可以在執行期間還原對測試資料所做的任何變更。您必須將資料庫和其他資料倉庫還原至其原始狀態，以還原測試期間產生的任何新記錄、更新或已刪除的記錄。

您可以重複使用管線來多次重複測試，直到結果反映出您想要的效能為止。您也可以使用管道來驗證程式碼變更不會破壞效能。您可以在非工作時間執行程式碼驗證測試，並使用可用於疑難排解的測試和可觀察性資料。

最佳做法包括：

- 記錄開始和結束時間，並自動生成用於記錄的 URL，這可以幫助您在適當的時間窗口中過濾可觀察性數據。監視和跟踪系統。
- 在調用測試時在標題中注入測試標識符。應用程式開發人員可以在後端使用識別碼做為篩選器，藉此豐富他們的記錄、監視和追蹤資料。
- 將管線限制為一次只執行一次。執行並行測試會產生雜訊，可能會在疑難排解期間造成混 在專用的效能環境中執行測試也很重要。

測試自動化工具

測試工具在任何測試自動化中發揮重要作用。開放原始碼測試工具的熱門選項包括：

- Apache JMeter 是經驗豐富的、功能強大的工具。多年來，Apache JMeter 變得更加可靠並新增了功能。使用圖形介面，您無需了解程式設計語言即可建立複雜的測試。公司，如 BlazeMeter支持阿帕奇 JMeter 的。
- K6 是一種免費工具，可提供支援、負載來源託管以及用於組織、執行和分析負載測試的整合式 Web 介面。

- Vegeta 負載測試遵循不同的概念。您可以定義特定速率，而不是定義並行或向系統施加負載。然後，此工具會建立獨立於系統回應時間的負載。
- H@@ ey 和 ab (Apache HTTP 伺服器工作台標記工具) 是基本工具，您可以使用命令列在單一端點上執行指定的負載。如果您具有伺服器來執行這些工具，這是產生負載的最快方法。即使本機筆記型電腦也能執行，儘管它可能不足以產生高負載。
- ghz 是一個命令行實用程序和 Go 包，用於負載測試和工作台標記 gRPC 服務。

AWS 提供 AWS 解決方案的分佈式負載測試。此解決方案可建立並模擬數以千計的連線使用者，以恆定的速度產生交易記錄，而不需要佈建伺服器。如需詳細資訊，請參閱AWS 解決方案資料庫。

您可以使用 AWS CodePipeline 自動化效能測試管線。如需使用自動化 API 測試的詳細資訊 CodePipeline，請參閱部AWS DevOps 落格和AWS 文件。

測試報告

測試報告是指與系統、應用程式、服務或程序效能相關的資料收集、分析和呈現方式。它涉及到測量各種指標和指標，以評估效率, 響應, 可靠性, 和一個特定的系統或組件的整體有效性.

性能測試報告涉及根據分析的上下文和目標選擇相關指標。常見的效能測量結果包括回應時間、傳輸量、錯誤率、資源使用率 (CPU、記憶體、磁碟) 和網路延遲。

收集效能相關資料之後，就必須將其儲存在中央存放庫中。這些測試結果可能來自不同的環境、應用程式和測試工具。當您在不同環境中執行多個工作負載時，很難收集與效能相關的資料，並在這些資料點之間建立關聯，以得出明智的結論。我們建議定義標準方法，使用中央存放庫進行資料儲存和視覺化收集效能指標資料。

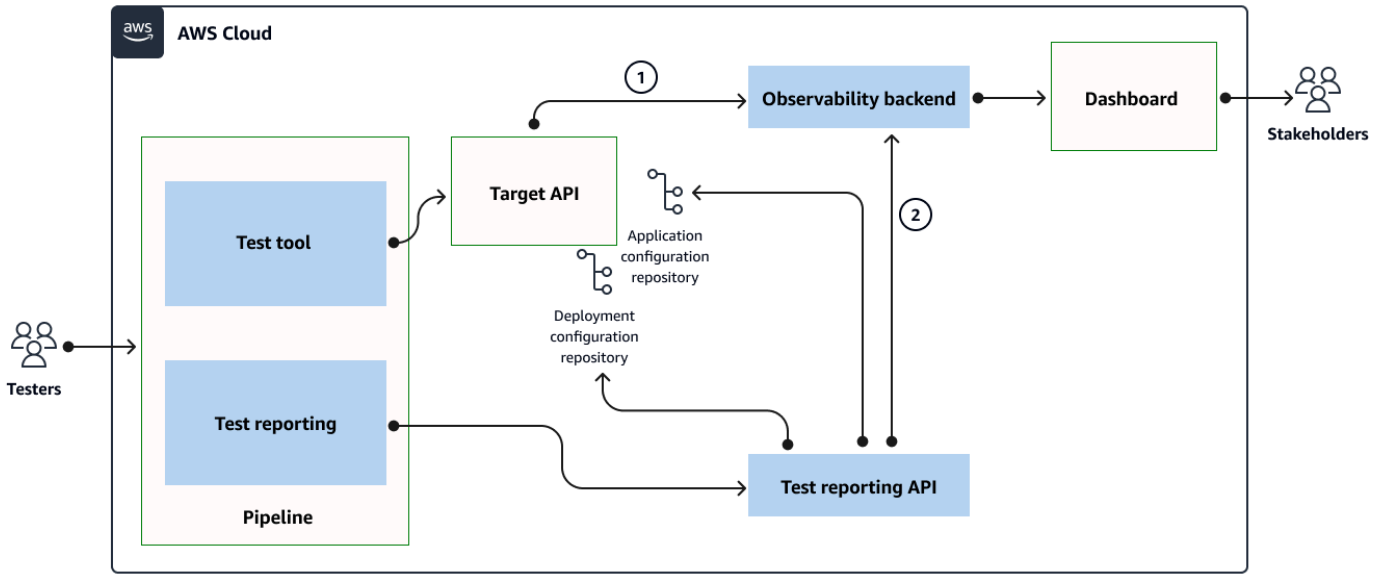
標準化錄音

我們建議將不同利益相關者執行效能測試的方式標準化，並將產生的資料寫入中央儲存庫。例如，這可能採用接受結果並將其存儲到持久存儲解決方案中的 API 的形式。在需要從來源 (例如 Prometheus 的 Amazon 受管服務) 擷 GitOps 取資料的情況下，API 可以根據描述如何從部署規格和 Kubernetes 規格擷取欄位的結構描述檔案，直接從指定來源提取這些詳細資料。架構文件可以使用JSONPath表達式或 Prometheus 查詢語言（PromQL）。如前所述，收集的指標應與績效分析的內容和目標相關。

傳遞至 API 的資料可以包含與應用程式和執行測試的環境相關的詳細資料和標籤。這有助於對性能測試數據進行分析。

效能工程支柱的實際運作

以下參考架構示範了測試特定 API 的效能工程支柱。



1. 記錄、監視和追蹤資料會從目標 API 傳送至後端。
2. 調用時，測試報告 API 將結果和配置信息發送到後端。

核心組件是被測的目標 API 或應用程序。目標 API 會以方式與應用程式組態存放庫和部署組態存放庫同步，以取得最新的應用程式和基礎結構組態。 GitOps這種同步允許自動化測試針對 Git 存儲庫中定義的應用程序的當前所需狀態及其支持的基礎結構運行。

測試自動化管道可自動產生測試資料、執行測試，以及報告目標 API 的測試結果。

目標 API 使用可觀察性最佳實踐生成性能見解（指標，日誌和跟踪），並將指標數據流式傳輸到可觀察性後端。

測試報告 API 收集所有與測試相關的報告數據（配置和測試結果），並將其存儲在可觀察性後端。

效能洞見和報告資料 (組態、測試結果) 的彙總可協助您查詢目標 API 的效能相關資料。例如，您可能會詢問以下內容：

- 前十名最慢的交易是什麼？
- 什麼是 P99，P90，每個測試的平均次數？
- 兩個測試運行的配置如何比較？

在一段時間內將測試案例與結果，配置和指標相關聯有助於識別最佳配置和性能結果。

使用這些測試結果，您可以為 API 做出更精確、資料驅動的決策，並在將 API 帶入生產環境時充滿信心。

資源

AWS 服務

- Amazon CloudWatch
- AWS CodePipeline
- AWS Distro for OpenTelemetry
- Amazon OpenSearch Service
- AWS X-Ray

實作

- amazon-kinesis-data-generator
- AWS Glue 測試資料產生器
- 上的分散式負載測試 AWS

部落格文章

- 使用 Fluent Bit 進行集中式容器記錄
- 使用新的 Amazon Kinesis Data Generator 測試串流資料解決方案
- 使用 AWS Distro for OpenTelemetry 介紹 Amazon EKS Fargate 的 Amazon CloudWatch Container Insights
- 使用 在 Kubernetes 上追蹤應用程式 AWS X-Ray
- 使用 AWS 適用於 Distro for OpenTelemetry 的 Amazon EKS 附加元件的指標和追蹤集合
- Amazon Managed Service for Prometheus 入門

研討會

- 可 AWS 觀測性簡介

AWS 方案指引

- 負載測試應用程式（指南）

第三方應用程式

- Apache JMeter
- K6
- Vegeta
- https://github.com/rakyll/hey 嘿，ab
- ghz

貢獻者

本文件的貢獻者包括：

- 瓦倫·夏爾馬, 高級首席顧問, AWS
- 阿卡什·庫馬爾, 高級首席顧問, AWS
- 阿查納·巴特納加爾, 實踐經理, AWS
- 普拉迪夏爾馬, 專業服務 II, AWS

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
初次出版	—	2024年4月24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

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。

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。某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 在部落格文章 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)

AI 欄位???，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。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，並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

2016 年 Klaus Schwab 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 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、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，請參閱此詞彙表中的 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)

用於工業自動化的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)

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。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

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

請參閱擷取增強產生。

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 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。

重新定位

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 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。