AWS 雲端採用架構：平台視角

AWS 規範指南

AWS 規範指南: AWS 雲端採用架構：平台視角

Table of Contents

歡迎 1
簡介 2
平台架構 5
Start 5
定義多帳戶策略 5
定義預防性控制 5
定義組織單位結構 5
定義網路連線 5
定義 DNS 策略 6
定義標籤標準 6
定義可觀察性策略 7
提前 7
定義主動和偵探控制 7
定義服務入職的標準 7
定義模式和原則 7
Excel 7
定義修復模式 7
溝通和優化政策 8
了解財務管理能力 8
平台工程 9
Start 9
建立 landing zone 並部署護欄 9
建立驗證 9
部署您的網路 10
收集、彙總和保護事件和記錄資料 10
建立控制 10
實作雲端財務管理 10
提前 10
建置自動化基礎 10
提供集中的觀察性服務 11
實施系統管理和 AMI 治理 11
管理認證使用 11
建立安全工具 11
Excel 12

透過自動化來源和散發身分建構 12
針對不同環境的異常模式新增偵測和警示 12
針對威脅進行分析和模型 12
持續收集、檢閱和重新設定權限 12
選取、測量並持續改善您的平台指標 12
資料架構 13
Start 13
定義總體功能 13
組織資料區 13
規劃資料的敏捷性和民主化 14
定義安全的資料傳遞 14
規劃成本效益 14
提前 14
瞭解特徵工程 14
規劃將資料集非規範化 15
設計可攜性與擴充性 15
Excel 15
設計可配置的框架 15
計劃建立統一的分析引擎 15
定義 DataOps 15
資料工程 16
Start 16
部署資料湖 16
開發資料擷取模式 16
加速資料處理 17
提供資料視覺化服務 17
進階 18
實作近乎即時的資料處理 18
驗證資料品質 18
證明資料轉換服務 18
啟用資料民主化 19
Excel 19
提供 UI 型協調 19
整合 DataOps 19
佈建和協調 21
Start 21

部署目 hub-and-spoke 錄模型 21
組織範本以供重複使用 21
套用預設參數以重複使用 21
建立核准程序 22
提前 22
建立自助入口網站 22
啟用私人市集 22
管理權利 22
Excel 22
與採購系統整合 22
與您的 ITSM 工具整合 23
實作生命週期管理與版本發佈系統 23
現代應用開發 24
Start 24
探索現代化方法 24
採用雲端原生運算功能 24
使用容器化 25
使用新式資料庫 25
提前 25
最佳化您的現代建築 25
使用服務網格技術 26
確保可見性和可追溯 26
Excel 26
擁抱微服務 26
持續整合和持續交付 28
Start 28
採用軟體元件管理 28
建立 CI/CD 管道 28
部署自動化測試 29
建立文件 29
使用基礎設施作為程式碼 29
保留和追蹤標準指標 29
進階 30
使用組態管理 30
整合監控和記錄 30
建立合併的節奏 30

擷取部署後的行為 30
Excel 31
整合 AI/ML 技術 31
採用混亂工程實務 32
最佳化效能 32
實作進階可觀測性 32
實作 GitOps 實務 33
結論 34
深入閱讀 35
貢獻者 36
文件歷史紀錄 37
詞彙表 38
..... 38
A 38
B 41
C 42
D 45
E 48
F 50
G 51
H 51
I 52
L 54
M 55
O 59
P 61
Q 63
R 63
S 66
T 68
U 70
V 70
W 70
Z 71
..... lxxii

AWS 雲端採用架構：平台觀點

Amazon Web Services（貢獻者）

2023 年十月 (文件歷史記錄)

數位轉型是主管改善客戶體驗、創新和靈活性的最大推動因素。它使用機器學習 (ML)、人工智慧 (AI)、大數據，以及雲端的速度和規模，以滿足不斷變化的業務條件和不斷變化的客戶需求。

Amazon Web Services（AWS）是世界上最全面和廣泛採用的雲平台。它可以幫助您轉型組織，同時降低業務風險，改善環境，社會和治理（ESG）績效，增加收入並提高營運效率。

AWS 雲端採用架構 (AWS CAF) 使用 AWS 最佳實務來協助您加速業務成果。使用 AWS CAF 來識別轉型機會並排定優先順序、評估和改善雲端準備程度，以及反覆演進您的轉型藍圖。

AWS CAF 將其指導分為六個角度：商業，人員，治理，平台，安全性和運營。每個透視都包含在單獨的指南中。本指南涵蓋平台觀點，其重點是透過企業級、可擴充的混合式雲端環境加速雲端工作負載的交付。

簡介

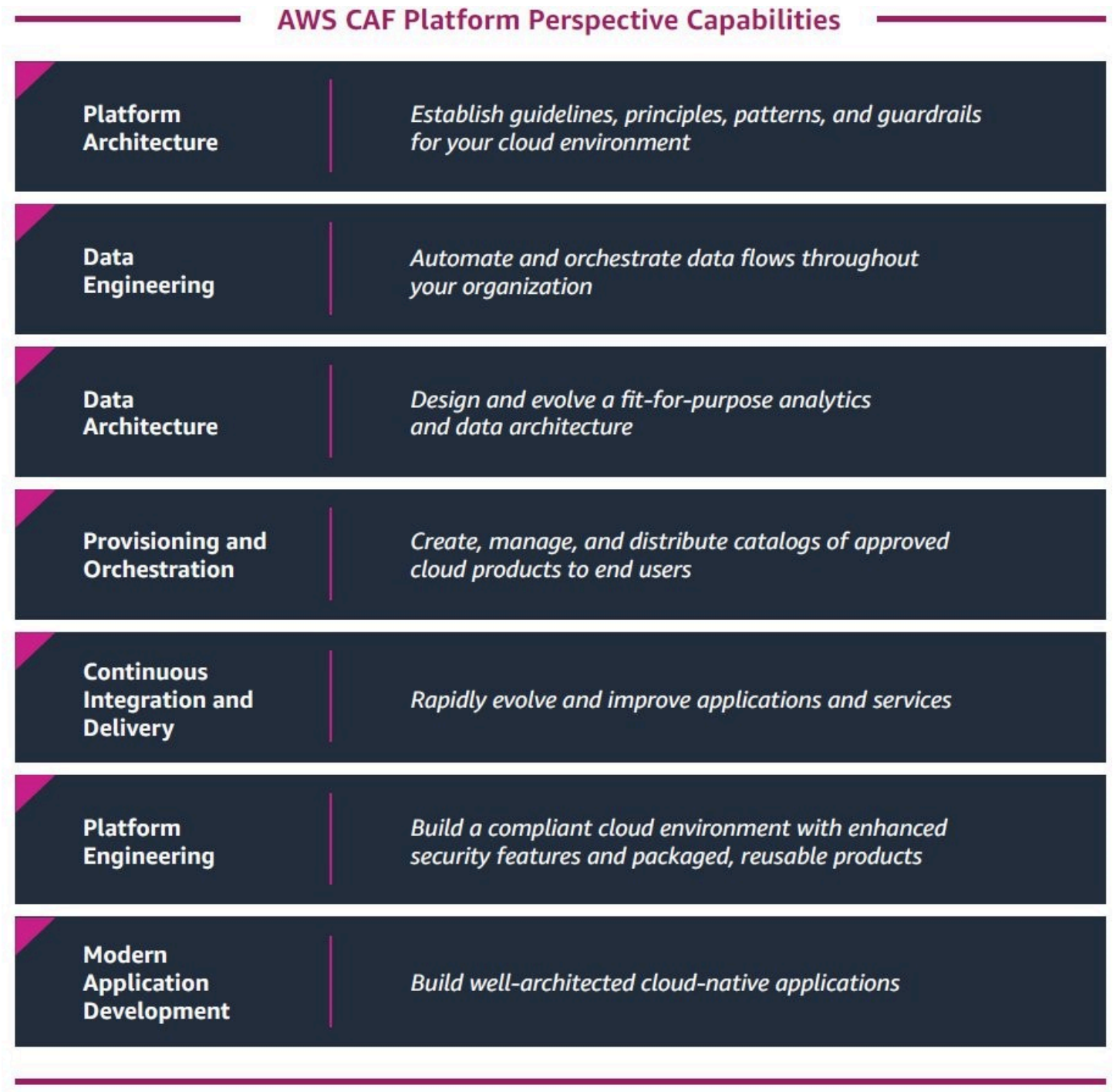
數以百萬計的客戶，包括增長最快的初創公司，最大的企業和領先的政府組織，都在使用。 AWS請參閱 AWS 網站上的客戶成功案例。) 他們可以遷移和現代化舊式工作負載、變得更加資料導向、自動化和最佳化業務程序，以及重塑作業模式。他們能夠通過降低業務風險，改善環境，社會和治理（ESG）績效，增加收入並提高運營效率來改善業務成果。

有效使用雲端進行數位轉型 (組織雲端準備程度) 的組織能力，是由一組基礎功能所強化。能力是一種組織能力，可以使用程序部署資源 (人員、技術以及任何其他有形或無形資產) 來達成特定結果。 AWS CAF 可識別這些功能，並提供規範性指導，讓全球數以千計的組織成功地用來改善雲端準備程度並加速雲端轉型旅程。

AWS CAF 將其能力分為六個角度：

- 業務
- 人物
- 管治
- 平台
- 安全性
- 操作

平台觀點著重於透過企業級、可擴充的混合式雲端環境加速雲端工作負載的交付。這種環境包括如下圖所示七個功能。這些功能由在雲端轉型旅程中與功能相關的利益相關者管理。典型的利益相關者包括技術長 (CTO)、技術領導者、建築師和工程師。



這些功能將在本指南的以下各節中詳細討論。每個部分都提供了有關如何在特定功能中開始，前進和最終擅長的指導方針。

- 平台架構
- 平台工程
- 資料架構

- 數據工程
- 佈建和協調
- 現代應用開發
- 持續整合與持續交付 (CI/CD)

平台的角度是 AWS CAF 的關鍵部分。這是跨所有其他角度做出的決策融合在一起，以提供業務敏捷性和價值的關係。在這裡做出的決定有助於或阻礙您在基礎層面的業務目標。 AWS CAF Platform 的觀點有助於建立企業級、可擴充的雲端環境，為您的組織轉型提供支撐。透過這個角度， AWS CAF 會引導您建立一個強大的平台，以實現您的雲端旅程，最終導致重大的業務轉型和成長。

當您透過平台觀點進行工作時，請考慮與需要開發的業務領導者之間的跨職能聯繫，以及他們為您的團隊和組織帶來的價值。額外關注作業模型變更和團隊拓撲，以確保符合需求。 此外，尋求開發團隊建置平台所需的技能，並在應用程式團隊之間使用平台。 在做出這些決策時，請記住組織的人員、業務、治理、安全性和營運目標，這些決策對於確保平台的採用和努力取得成功至關重要。

AWS 合作AWS 夥伴網路提供工具和服務，例如研討會和訓練，可協助您實作並改善安全性狀態。AWS 專業服務是一個由專家組成的全球團隊，透過一系列 AWS 符合 CAF 的產品，協助您達成與雲端轉型相關的特定成果。

平台架構

為您的雲端環境建立並維護準則、原則、模式和護欄。

架構良好的雲端環境可協助您加速實作、降低風險並推動雲端採用。平台架構功能可在您的組織內為推動雲端採用的企業標準建立共識。您可以定義最佳實務藍圖和護欄，以促進驗證、安全性、網路以及記錄和監控。此外，您還要考慮並規劃因延遲、資料處理或資料駐留需求而需要在內部部署保留的工作負載，並評估混合雲使用案例，例如雲端爆量、備份和災難復原、分散式資料處理和邊緣運算。

Start

定義多帳戶策略

一個良好的多賬戶策略考慮規模和營運能力問題。這意味著將您的工作負載隔離為最符合您營運需求的邏輯模式。我們建議您從一組基礎帳戶開始，以容納企業中的集中式和分散式服務。您可以集中安全性、財務和營運功能，以有效地管理和管理您的分散式和自主團隊和帳戶。您將希望整個組織保持一致，以了解平台和工作負載的分段和管理方式。瞭解此結構可協助您確保安全性原則適用於驗證和授權，同時與平台不斷發展的可接受使用原則保持一致。

定義預防性控制

使用一組內嵌的預設控制項 (護欄) 規劃安全的多帳戶環境。開始瞭解並使用諸如服務控制原則 (SCP) 之類的機制來管理整個組織的服務使用情況，包括可在 AWS 區域 雲端平台內使用的服務使用情況。政策提供了一個集中式機制，用於控制所有帳戶可用的最大權限，並確保它們遵守組織的存取控制準則。

定義組織單位結構

組織單位 (OU) 是根據法規要求和軟體開發生命週期 (SDLC) 環境來管理和分類帳戶的實用方法。透過使用 OU，組織可簡化跨雲端基礎架構申請適當原則和權限的程序。工作負載 OU 是專為支援應用程式基礎結構資源的帳戶所設計，並確保強制執行正確的原則。使用 OU 和 SCP 有助於增強組織的雲端基礎架構的安全性和合規性，同時確保應用程式和服務的順暢運作。這最終導致了更有效和更強大的雲採用過程。

定義網路連線

網路連線是任何雲端基礎架構的關鍵方面，這些基礎架構支援建立安全、可擴充且高可用性的網路，以支援應用程式和工作負載。精心設計的網路可提供一致的高效能，並確保在不同環境中順暢運作。

設計網路架構時，請考慮您是否有因延遲、資料處理或資料存放需求而要在內部部署保留的工作負載。透過評估混合雲使用案例，例如雲端爆發、備份和災難復原到雲端、分散式資料處理以及邊緣運算，您可以識別下列方面的關鍵需求：

- 與網際網路之間的連線。這方面涉及在您的應用程式或工作負載與網際網路之間提供安全可靠的連線。這種連接對於促進對基於 Web 的資源的訪問，實現用戶和應用程序之間的通信以及確保在需要時可以訪問您的服務至關重要。
- 跨雲端環境的連線能力。此區域著重於在雲端基礎架構內的各種元件和服務之間建立穩固的連線。它可確保在不同雲端服務之間輕鬆共用和存取資料和資源，從而促進有效率的協同作業和更順暢的作業。這裡的主要考慮因素是您使用虛擬私有雲 (VPC)。為了讓事情變得簡單，請考慮針對如何建立和追蹤 VPC 建立標準。請考慮以程式設計方式建立這些標準，並規劃使用 IP 位址管理 (IPAM) 解決方案。分配足夠的 IP 空間以實現成長，並設計子網路結構，以便在使用多個可用區域時輕鬆進行疑難排解。設計和實作網路連線時，請務必遵循 VPC 的安全性最佳做法。
- 內部部署網路與雲端環境之間的連線能力。這方面涉及內部部署基礎結構與雲端環境的整合。透過在兩者之間建立安全可靠的連線，組織可以從混合式架構的優勢中獲益。例如，您可以同時使用內部部署資源和雲端服務，以改善效能、延展性和成本最佳化。

透過解決網路連線的這三個關鍵領域，您可以建置強大的雲端基礎架構，以有效地支援您的應用程式和工作負載，因此您可以充分利用雲端採用的優勢。 記下網路需求，並建立簡單的設計，讓您能夠根據您的多帳戶策略進行擴充。

定義 DNS 策略

精心規劃的 DNS 策略可協助您避免隨著雲端環境的成長而複雜。如果您維護內部部署 DNS 功能，建議您針對任何雲端型 DNS 需求，設計使用內部部署 DNS 基礎結構和雲端 DNS 的混合式 DNS 架構。使用解析器端點和轉送規則，將 DNS 解析與內部部署 DNS 環境整合。使用私有託管區域來保存有關您希望 Cloud DNS 如何回應一或多個網路中網域及其子網域的查詢的相關資訊。

定義標籤標準

標記資源是有效管理成本和識別資源擁有權的重要作法。 考慮您的組織將如何進一步允許雲端使用，包括在平台內使用特定服務。定義標記策略，以追蹤哪些團隊正在部署哪些資源。 從 AWS CAF 營運的角度獲取輸入，並使用標籤為您部署的基礎結構自動執行任務。

此外，透過使用相關中繼資料標記資源，您可以根據 AWS CAF 治理觀點中雲端財務管理 (CFM) 功能中指定的組織需求來分組和追蹤支出。識別支援您的會計和財務實務的報告機制，包括違反財務政策時要採取的行動。

定義可觀察性策略

建立可觀察性策略是最佳化和保護雲端架構的關鍵一步。此策略圍繞將雲端服務產生的指標和日誌轉換為可行的洞察，以便進行策略決策。排定監控關鍵績效指標的優先順序，並設定警示，以搶先解決潛在問題。為了防止工具擴散、最佳化成本，並專注於組織最重要的事情，請在您的平台和應用程式中整合此可觀察性策略。有關進一步的指導，請參閱我們關於制定可觀察性策略的演示文稿（AWS Re：Invent 2022）。

提前

定義主動和偵探控制

為了推進，您的組織必須確定是否需要在環境中進行主動和偵測控制（護欄）。建立原則，以確定角色和使用者在組織單位 (OU) 內的帳戶中所擁有的護欄或限制。檢閱平台的任何預設偵探護欄，並選擇要套用的護欄。視需要建立額外的預防性和偵測控制項，並依據 OU 將它們分組，以符合您的多帳戶策略。考慮您需要哪些組織工具和機制來檢查偵測控制項所識別的不相容資源。

定義服務入職的標準

建立平台可接受使用的標準，以及與服務消費相關的模式，以及如何管理這些標準。考慮允許使用哪些初始服務。創建概述這些標準的文檔，並將其發布給平台的用戶和操作員。確保這些標準隨著時間的推移而適應，以滿足組織不斷變化的目標以及不斷發展的雲端運算能力。

定義模式和原則

使用應用程式擁有者的輸入，考慮組織允許使用哪些架構模式，並開始定義標準化的藍圖。標準化可讓您在雲端進行擴充時，提供更大的管理能力並降低管理負擔。使用整合至變更控制程序和 IT 服務管理 (ITSM) 系統的服務目錄，定義將使用基礎結構即程式碼 (IaC) 並規劃簡化部署模式的模式。定義如何使用這些藍圖，以及允許例外狀況的情況。規劃這些例外狀況及其治理，並考量驗證、安全性監控和護欄。

Excel

定義修復模式

考慮如何對偵探護欄檢查進行註釋和優先順序，以便根據您的安全性和合規性框架進行修復。 計劃使用自動化來偵測資源的 out-of-policy 佈建，包括違反預算和標記政策的資源。識別在更新手冊和教戰手

冊時設定和衡量服務等級目標所需的功能。設定這些做法的定期檢閱，並設定意見反應機制，以擷取與平台演進相關的資料。定義相應地建立和更新手冊和教戰手冊的機制。

溝通和優化政策

為所有文件建立集中式內容管理系統，並將其散發給平台的使用者和操作員。建立一個機制來擷取意見反應，以便 future 考慮原則的變更。

了解財務管理能力

當 Organizations 對預算保持透明和全面的了解時，就會蓬勃發展。這使他們能夠做出明智的決策，有效地分配資源並實現其戰略目標。清晰的預算視圖通過促進明智的決策，有效的資源分配，成本控制，績效衡量以及維護責任和合規性，從而幫助組織脫穎而出。這最終導致了一個更有效，財務穩定和繁榮的組織。如果您有成功的標記策略，您可以在中AWS Budgets使用成本篩選器，根據資源標籤篩選費用。這可協助您建立針對特定專案、部門、環境或其他準則量身打造的預算，進一步提升財務管理能力。您可以將成本分配標籤和 Co AWS st Categories 與標籤相關聯，以便在報告成本時提高財務洞察力和透明度。

平台工程

使用封裝且可重複使用的雲端產品，建置安全、合規的多帳戶雲端環境。

為了支援開發團隊來支援創新，該平台需要快速適應，以跟上業務需求。（請參閱 AWS CAF 業務觀點。）它必須這樣做，同時具有足夠的靈活性以適應產品管理需求，並且足夠堅持安全限制，並且速度足以滿足營運需求。此程序需要建置具有增強安全性功能的合規多帳戶雲端環境，以及封裝且可重複使用的雲端產品。

有效的雲端環境可讓您的團隊輕鬆佈建新帳戶，同時確保這些帳戶符合組織政策。一組精心策劃的雲端產品可讓您編纂最佳實務、協助您進行控管，並協助提高雲端部署的速度和一致性。部署您的最佳實踐藍圖，以及偵探和預防性護欄。將您的雲端環境與現有環境整合，以啟用所需的混合雲使用案例。

自動化帳戶佈建工作流程，並使用多個帳戶來支援您的安全性和治理目標。設定內部部署與雲端環境之間以及不同雲端帳戶之間的連線。在您現有的身分識別提供者 (IdP) 與雲端環境之間實作聯合，以便使用者可以使用現有的登入認證進行驗證。集中記錄、建立跨帳戶安全性稽核、建立輸入和輸出 DNS 解析器，以及取得帳戶和防護儀表板的可見度。

根據企業標準和組態管理，評估和認證要使用的雲端服務。Package 並持續改善企業標準，做為自助式部署產品和可消耗性服務。利用基礎結構即程式碼 (IaC)，以宣告式的方式定義組態。建立支援團隊，將平台傳播給開發人員和商業使用者，並允許他們建立整合，以加速整個組織的採用。

若要完成下列各節中討論的工作，您必須建立能力與團隊，以便讓您的組織邁向現代化平台工程。如需技術詳細資訊，請參閱 AWS白皮書中的建立雲端基礎。

Start

建立 landing zone 並部署護欄

當您開始邁向成熟的平台工程之旅時，您必須首先使用平台架構功能中所定義的偵探和預防性護欄來部署您的 landing zone。護欄可確保應用程式擁有者消耗雲端資源時，不會違反組織標準。透過此機制，您可以自動化帳戶佈建工作流程，以使用多個帳戶來支援您的安全性和治理目標。

建立驗證

根據 AWS CAF Security 觀點中規定的標準，在所有環境、系統、工作負載和程序中實作身分識別管理和存取控制。對於員工身分識別，請限制 AWS Identity and Access Management (IAM) 使用者的使用，而是仰賴可讓您在集中式位置管理身分識別的身分識別提供者。這樣可以更輕鬆地跨多個應用程式

和服務管理存取，因為您是從單一位置建立、管理和撤銷存取權。使用現有程序來管理建立、更新和移除存取權，以包含您的 AWS 環境。

部署您的網路

根據您的平台架構設計，建立集中式網路帳戶，以控制進出環境的入站和輸出流量。我們建議您設計網路，以便在內部部署網路與 AWS 環境之間、網際網路以及環 AWS 境之間快速佈建的連線能力。集中化網路管理可讓您部署網路控制，藉由使用預防性和反應式控制來隔離整個環境的網路和連線能力。

收集、彙總和保護事件和記錄資料

使用 Amazon CloudWatch 跨帳戶觀察能力。它提供了一個統一的界面，用於搜索，可視化和分析鏈接帳戶中的指標，日誌和跟踪，並消除帳戶界限。

如果您的組織對集中式記錄控制和安全性有特定的合規性需求，請考慮設定專用的記錄封存帳戶。這提供了專門用於日誌數據的集中式加密存儲庫。定期輪換加密金鑰，以增強此歸檔的安全性。

視需要使用遮罩技術，實作可靠的原則來保護敏感記錄資料。針對符合性、安全性和稽核記錄使用記錄彙總，並確保使用嚴格的保護和身分識別結構，以防止未經授權的變更記錄組態。

建立控制

根據 AWS CAF Security 觀點的定義，部署符合您業務需求的基礎安全性功能。部署其他預防性和偵測控制項，並在需要時以程式設計方式一致地在所有帳戶中佈建這些控制項。依照平台架構功能所定義，將偵測控制項整合至作業工具中，以便可透過作業機制檢閱不符合規定的資源。

實作雲端財務管理

根據 AWS CAF 治理觀點，實施成本分配標籤和 AWS 成 Cost Categories，使您組織的標記策略與雲端消費的財務責任保持一致。 AWS 「Cost Categories」可讓您使用中發佈的工具 (例如AWS Cost Explorer和帳單資料)，向內部成本中心收取或顯示雲端費用AWS Cost and Usage Report。

提前

建置自動化基礎

在繼續之前，請先評估並認證雲端服務以符合您的平台架構使用。然後，封裝並持續改善可部署產品和可消耗性服務的企業標準，並使用基礎結構即程式碼 (IaC) 以宣告式方式定義組態。基礎結構自動化可透過角色型存取控制 (RBAC) 或屬性型存取控制 (ABAC) 來存取每個帳戶中的特定服務，藉此模擬軟體

開發週期。使用 API 部署方法以快速佈建新帳戶，並使其與您的服務和事件管理功能保持一致，或開發自助服務功能。在建立帳戶時自動化網路整合和 IP 分配，以確保合規性和網路安全性。使用設定要使用的原生連接器，將新帳戶與您的 IT 服務管理 (ITSM) 解決方案整合。 AWS視需要更新您的教戰手冊和手冊。

提供集中的觀察性服務

為了實現有效的雲端觀察能力，您的平台應支援本機和集中式記錄資料的即時搜尋和分析。隨著營運擴展，您的平台對日誌、指標和跟踪進行索引、視覺化和解釋的能力是將原始數據轉化為可行洞察的關鍵。

通過關聯日誌，指標和跟踪，您可以提取可行的結論並制定有針對性的，明智的響應。建立規則，以便主動回應記錄檔、指標或追蹤中所識別的安全事件或模式。隨著 AWS 解決方案的擴展，請確保您的監控策略同時擴展，以維護和增強您的觀察能力。

實施系統管理和 AMI 治理

使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的 Organizations 廣泛需要操作工具來大規模管理執行個體。軟體資產管理、端點偵測與回應、庫存管理、漏洞管理和存取管理是許多組織的基礎功能。 這些功能通常是透過安裝在執行個體上的軟體代理程式來提供。開發將代理程式和其他自訂組態封裝到 Amazon 機器映像 (AMI) 的功能，並讓雲端平台的消費者可以使用這些 AMI。使用預防性和偵測控制來管理這些 AMI 的使用。AMI 應該包含可大規模管理長時間執行 EC2 執行個體的工具，特別是對於不定期使用新 AMI 的可變 Amazon EC2 工作負載。您可以大規模使用AWS Systems Manager自動化代理程式升級、收集系統庫存、遠端存取 EC2 執行個體，以及修補作業系統弱點。

管理認證使用

根據 AWS CAF 安全性的觀點，實現角色和臨時憑據。使用工具來管理執行個體或內部部署系統的遠端存取，方法是使用預先安裝的代理程式，而不 減少對長期認證的依賴，並掃描 IaC 範本中的硬式編碼認證。如果您無法使用臨時登入資料，請使用程式設計工具 (例如應用程式 Token 和資料庫密碼) 來自動執行認證輪替和管理。透過使用 IaC 的最低權限原則編碼使用者、群組和角色，並防止使用護欄手動建立身分帳戶。

建立安全工具

安全監控工具應支援跨基礎架構、應用程式和工作負載的精細安全監控，並提供用於模式分析的彙總檢視。與所有其他安全性管理工具一樣，您應該擴充延伸偵測與回應 (XDR) 工具，以根據 AWS CAF Security 觀點 AWS 中定義的需求，提供評估、偵測、回應及修復應用程式、資源和環境安全性的功能。

Excel

透過自動化來源和散發身分建構

使用 IaC 工具編碼和版本標識構造，例如角色，策略和模板。使用政策驗證工具來檢查安全警告、錯誤、一般警告、IAM 政策的建議變更以及其他發現項目。在適當的情況下，部署和移除以自動化方式提供暫時存取環境的身分識別結構，並禁止使用主控台的個人進行部署。

針對不同環境的異常模式新增偵測和警示

主動評估環境中的已知弱點，並針對異常事件和活動模式新增偵測。檢閱發現結果並向平台架構團隊提出建議，以進一步推動效率和創新的變更。

針對威脅進行分析和模型

根據 AWS CAF Security 觀點的要求，針對產業和安全性基準執行持續監控和衡量。當您實作檢測方法時，請判斷哪些類型的事件資料和資訊最適合您的安全性管理功能。此監控包含數種攻擊媒介，包括服務使用情況。您的安全基礎應該包括在多帳戶環境中進行安全記錄和分析的全面功能，其中包括關聯來自多個來源的事件的能力。使用特定的控制項和護欄，防止對此配置進行變更。

持續收集、檢閱和重新設定權限

記錄身分角色和權限的變更，並在偵探護欄偵測到與您預期的設定狀態有所偏差時，實作警示。使用彙總和模式識別工具來檢閱集中式事件集合，並視需要重新設定權限。

選取、測量並持續改善您的平台指標

為了實現成功的平台運營，請建立並定期查看綜合指標。確保它們符合組織目標和利益相關者的需求。追蹤平台效能和改善指標，並使用團隊啟用和工具採用指標，結合作業參數，例如修補程式、備份和合規性。

使用CloudWatch跨帳戶觀察能力進行有效率的指標管理。此服務簡化了資料彙總和視覺化，以實現明智的決策和針對性的增強功能。使用這些指標作為成功的指標和變化的驅動因素，以營造持續改進的環境。

資料架構

設計和發展資 fit-for-purpose 料和分析架構。

精心設計的數據和分析架構對於獲得可行的見解至關重要。透過設計和發展 fit-for-purpose 資料和分析架構，組織可降低複雜性、成本和技術債務，同時從不斷增長的資料量中獲得寶貴的見解。通過與 AWS CAF 原則保持一致，企業可以創建與其現有平台無縫集成的數據架構。這種調整使組織能夠充分利用現代數據處理和分析技術提供的優勢。

資料和分析架構是組織從資料衍生價值的能力的藍圖。它可以幫助組織獲得新的業務洞察力，是業務增長的催化劑。為了支援業務需求，現代化的資料架構應符合短期和長期的業務目標，並且對組織的文化和情境需求是獨一無二的。在當今世界中，成功實施和採用數據和分析架構是基於在正確的時間為正確的消費者提供正確數據的原則。

這是透過規劃和組織組織資料資產的模型 (實體或邏輯) 的方式、如何保護資料，以及這些資料模型如何彼此互動以解決業務問題，以及衍生未知模式並產生深入分析資訊來達成這個目標。

Start

定義總體功能

在目前的商業環境中，對於現代資料分析平台而言，從資料衍生價值以支援組織中的各種領域至關重要。現代資料架構不採用單一資料架構方法，而應該包含針對特定使用案例建置和最佳化的工具集和模式。該架構應該能夠發展並包含基本的構建模塊，例如可擴展的數據湖，專門打造的分析服務，統一的數據訪問和統一控管。

組織資料區

如何組織和存儲數據以便快速輕鬆地訪問是數據架構的關鍵方面。這可以通過在數據湖中設置自定義數據區來實現。資料區的分類如下：

- 從異質來源收集的原始資料
- 精心策劃和轉換的資料，以支援每個網域的分析需求
- 針對報告需求的使用案例或以產品為基礎的資料集市
- 具有安全性和合規控制的外部公開資料

規劃資料的敏捷性和民主化

分析平台的有效性取決於佈建資料的速度，以及將佈建的資料民主化以供使用。資料佈建的靈活性是因為資料架構能夠根據使用案例，以各種方式取得和處理資料，例如即時、近乎即時、批次、微批次或混合式資料。透過定義由資料管理員監控的資料共用和存取控制工作流程來實現資料民主化。實作資料市場是使資料民主化的推動因素之一。

定義安全的資料傳遞

現代資料架構是外界安全性的堡壘，但可讓員工或資料使用者輕鬆存取其工作職能所定義，並遵守法規遵循限制，例如《Health 保險可攜性與責任法案》(HIPAA)、個人識別資訊 (PII)、一般資料保護規範 (GDPR) 等。這是通過基於角色的訪問控制（RBAC）和基於標籤的訪問控制（TBAC）方法實現的。開啟時 AWS，標籤用於控制對資料的存取，以簡化存取控制管理。按照 AWS CAF 安全性觀點中概述的原則來執行此操作。

規劃成本效益

傳統的資料倉儲提供緊密結合的運算與儲存，而且資源使用成本高。現代化架構可將運算與儲存分離，並根據資料生命週期實作階層式儲存。例如，在上 AWS，您可以使用 Amazon Simple Storage Service (Amazon S3) 控制成本，並將資料儲存與運算分離。Amazon S3 儲存類別是專為針對不同存取模式提供成本最低的儲存。此外， AWS 運算工具 (例如 Amazon Athena AWS Glue、Amazon Redshift 和 Amazon SageMaker 執行階段) 都是無伺服器的，因此您不需要管理基礎設施，而且只需按使用量付費。

提前

現代資料架構可進一步強化，以增加資料使用量的廣度，從支援業務和營運功能的標準分析到支援預測和洞察的更複雜功能，並有助於加快決策速度。為了實現這一目標，該架構支持以下各節中描述的功能。

瞭解特徵工程

特徵工程使用機器學習，並涉及設置圖徵商店或功能市場。數據科學團隊為受監督和無監督學習模型創建新功能（衍生屬性），並將其存儲在功能集市中，以簡化轉換並提高數據準確性。企業可以在多個分析模型中重複使用這些功能，進而提高上市速度。

規劃將資料集非規範化

建構非規範化的資料集或資料集市可以大幅簡化商務使用者的資料集，方法是讓所需資料在單一位置隨時可用，並提高分析速度。如果精心設計，一個記錄可支援多種使用模式，並縮短整體開發生命週期。非規範化資料集的有效治理也很重要，原因有兩個。實作非標準化資料可能會產生大量的冗餘資料集，這可能會成為大規模管理的挑戰。此外，如果未正確建模，這些資料集可能會越來越難以重新利用。

設計可攜性與擴充性

大型組織很少將所有應用程式和使用者放在單一資料平台上。他們的應用程式和資料存放區通常分佈在傳統的內部部署和雲端平台上，因此分析團隊難以混合和合併資料。建議您根據網域、地理位置、商業使用案例等特性將資料容器化。這種容器化增加了各種平台和應用程序之間的可移植性，並支持更有效的消費。將資料分割到容器中並透過 API 公開，可協助您更輕鬆地擴展資料架構。它支援混合式 end-to-end 資料流程，並協助內部部署和雲端應用程式順暢運作。

Excel

隨著組織內部的現代分析架構不斷發展，因此通過引入可重複使用的概念來管理這種變化非常重要。這些概念可提高耐用性和採用率，同時控制成本。以下各節將討論一些要考慮的概念。

設計可配置的框架

Organizations 通常會建立多個複雜的模型來滿足其獨特的業務需求。這些模型需要建立多個資料管線和工程特徵。隨著時間的推移，這會產生顯著的冗餘並增加運營成本。建立結合一組參數驅動、可配置基礎模型的框架，可縮短開發時間和營運成本。分析引擎可以實作這些可設定的模型，以提供所需的輸出。

計劃建立統一的分析引擎

業務問題是獨一無二的，通常需要自訂技術來滿足需求，從而導致組織中產生多個分析引擎。設計和開發可支援多種程式設計範例的統一 AI 分析引擎介面，可簡化使用並降低成本。

定義 DataOps

大多數資料專業人員會花費大量時間執行資料作業，例如尋找正確的資料、轉換、建模等。擁有敏捷的數據操作（DataOps）可以通過打破數據工程師，數據科學家，數據所有者和分析師的孤島大大增強數據架構。 DataOps 促進團隊之間更好的溝通、縮短週期時間，並確保高資料品質。隨著時間的推移，數據和分析架構經歷了許多轉型，因為業務需求不斷變化和技術進步。組織必須努力開發、實作和維護資料和分析架構，這些架構會隨著時間的推移而發展並支援其業務。

資料工程

自動化和協調整個組織的資料流程。

使用中繼資料自動化處理原始資料的管道，並產生最佳化輸出。利用跨 AWS CAF平台架構和平台工程功能以及營運角度定義的現有架構防護和安全控制。與平台工程支援團隊合作，為常用模式開發可重複使用的藍圖，以簡化管道部署。

Start

部署資料湖

為結構化和非結構化資料使用適當的儲存解決方案，以建立基礎資料儲存功能。這可讓您從各種來源收集和儲存資料，並使資料可供存取以供進一步處理和分析。資料儲存是資料工程策略的關鍵元件。精心設計的資料儲存架構可讓組織以高效且符合成本效益的方式儲存、管理和存取其資料。 AWS 提供各種資料儲存服務，以滿足特定業務需求。

例如，您可以將 Amazon Simple Storage Service （ Amazon S3 ） 用於物件儲存、Amazon Relational Database Service （ Amazon RDS ） 用於關聯式資料庫，以及 Amazon Redshift 用於資料倉儲，藉此建立基礎資料儲存功能。這些服務可協助您以安全且符合成本效益的方式存放資料，並輕鬆存取資料以進行進一步處理和分析。我們建議您也實作資料儲存最佳實務，例如資料分割和壓縮，以提高效能並降低成本。

開發資料擷取模式

若要自動化和協調資料流程，請建立資料擷取程序，以從各種來源收集資料，包括資料庫、檔案和APIs。您的資料擷取程序應支援業務敏捷性，並將治理控制納入考量。

協調器應能夠執行以雲端為基礎的服務，並提供自動化排程機制。它應該提供任務之間的條件式連結和相依性選項，以及輪詢和錯誤處理功能。此外，它應該與警示和監控系統無縫整合，以確保管道順利執行。

一些常見的協調機制包括：

- 以時間為基礎的協調會在遞迴間隔和定義的頻率啟動工作流程。
- 事件型協調會根據事件的發生啟動工作流程，例如建立檔案或API請求。
- 輪詢會實作機制，任務或工作流程會在其中呼叫 服務 （ 例如，透過 API ），並等待定義的回應，然後再繼續進行下一個步驟。

現代架構設計強調利用可簡化雲端基礎設施管理的受管服務，並減少開發人員和基礎設施團隊的負擔。此方法也適用於資料工程。我們建議您在適用的情況下使用 受管服務來建置資料擷取管道，以加速資料工程程序。這些服務類型的兩個範例是 Amazon Managed Workflows for Apache Airflow （Amazon MWAA）和 AWS Step Functions：

- Apache Airflow 是廣受歡迎的協調工具，用於以程式設計方式撰寫、排程和監控工作流程。 AWS 提供 Amazon Managed Workflows for Apache Airflow （Amazon MWAA） 作為受管服務，可讓開發人員專注於建置，而不是管理協調工具的基礎設施。Amazon 使用 Python 指令碼MWAA輕鬆編寫工作流程。定向非週期性圖形（DAG）代表工作流程，作為任務集合，顯示每個任務的關係和相依性。您可以擁有任意數量DAGs，Apache Airflow 會根據每個任務的關係和相依性執行這些項目。
- AWS Step Functions 協助開發人員建置低程式碼視覺化工作流程，以自動化 IT 和業務流程。您使用 Step Functions 建置的工作流程稱為 狀態機器 ，工作流程的每個步驟稱為 狀態 。您可以使用 Step Functions 來建立內建錯誤處理、參數傳遞、建議安全設定和狀態管理的工作流程。這些會減少您必須寫入和維護的程式碼數量。任務透過與另一個 AWS 服務或您在內部部署或雲端環境中託管的應用程式協調來執行工作。

加速資料處理

資料處理是了解現代組織收集的大量資料的重要步驟。若要開始使用資料處理， AWS 提供 等受管服務AWS Glue，可提供強大的擷取、轉換和載入（ETL）功能。組織可以使用這些服務開始處理和轉換原始資料，包括清理、標準化和彙總資料，以準備進行分析。

資料處理從簡單的技術開始，例如彙總和篩選，以執行初始資料轉換。隨著資料處理需求不斷演進，您可以實作更進階ETL的程序，以便從各種來源擷取資料、將其轉換為滿足您的特定需求，並將其載入集中式資料倉儲或資料庫以進行統一分析。此方法可確保資料準確、完整，並可及時進行分析。

透過使用 AWS 受管服務進行資料處理，組織可以受益於更高層級的自動化、可擴展性和成本效益。這些服務可自動化許多例行資料處理任務，例如結構描述探索、資料分析和資料轉換，並為更具策略性的活動釋放寶貴的資源。此外，這些服務會自動擴展以支援不斷增長的資料磁碟區。

提供資料視覺化服務

尋找方法，讓使用資料視覺化以有意義的快速方式解譯資料的決策者可以使用資料。透過視覺化，您可以解譯模式，並提高各種利害關係人的參與度，無論他們的技術技能如何。良好的平台可讓資料工程團隊佈建資源，快速提供資料視覺化，且幾乎無需額外負荷。您也可以使用工具輕鬆查詢資料存放區，而不需要工程專業知識，以提供自助功能。請考慮使用內建工具，透過資料視覺效果和互動式儀表板提供無伺服器商業智慧，以及使用自然語言查詢後端資料。

進階

實作近乎即時的資料處理

資料處理是任何資料工程管道的重要元件，可讓組織將原始資料轉換為有意義的洞見。除了傳統的批次處理之外，即時資料處理在當今快節奏的商業環境中變得越來越重要。即時資料處理可讓組織在事件發生時回應事件，並改善決策和營運效率。

驗證資料品質

資料品質會直接影響衍生自資料的洞察和決策的準確性和可靠性。實作資料驗證和清除程序對於確保您使用高品質且值得信賴的資料進行分析至關重要。

資料驗證涉及透過比對預先定義的規則和條件來檢查資料的準確性、完整性和一致性。這有助於識別資料中的任何差異或錯誤，並確保其符合用途。資料清除涉及識別和更正資料中的任何不正確、不一致或重複。

透過實作資料品質程序和工具，組織可以提高從資料衍生之洞察的準確性和可靠性，進而產生更好的決策和營運效率。這不僅增強了組織的效能，還提高利益相關者對產生的資料和分析的信心和信任。

證明資料轉換服務

資料轉換會準備進階分析和機器學習模型的資料。它涉及使用資料正規化、擴充和重複資料刪除等技術，以確保資料乾淨、一致且準備好進行分析。

- 資料標準化涉及將資料組織成標準格式、消除冗餘，以及確保資料在不同來源之間保持一致。這可以更輕鬆地分析和比較來自多個來源的資料，並讓組織更全面地了解其操作。
- 資料擴充涉及使用來自外部來源的其他資訊增強現有資料，例如人口統計資料或市場趨勢。這可針對客戶行為或產業趨勢提供寶貴的洞見，而這些趨勢可能單獨從內部資料來源看不見。
- 重複資料刪除涉及識別和移除重複的資料項目，並確保資料準確且無錯誤。這在處理大型資料集時尤其重要，其中即使只有一小部分重複也可能會扭曲分析結果。

透過使用進階資料轉換技術，組織可以確保其資料具有高品質、準確且已準備好進行更複雜的分析。這會導致更好的決策、提高營運效率，以及市場中的競爭優勢。

啟用資料民主化

透過讓資料可供所有員工存取、理解和使用，促進資料普及化文化。資料民主化有助於員工做出資料驅動的決策，並為組織的資料驅動文化做出貢獻。這意味著分解孤島並建立文化，讓所有員工共用和使用資料來推動決策。

整體而言，資料民主化是關於建立一種文化，讓組織中的每個人都能重視、存取和理解資料。透過啟用資料民主化，組織可培養以資料為導向的文化，以推動創新、改善決策，最終帶來業務成功。

Excel

提供 UI 型協調

若要建置敏捷且使用有效方法的組織，請務必規劃現代協調平台，該平台由跨業務單位的開發和操作資源使用。目標是開發、部署和共用資料管道和工作流程，而不依賴單一團隊、技術或支援模型。這可透過 UI 型協調等功能來實現。互動等 drag-and-drop功能可讓技術專業知識不足的使用者建構DAGs和說明機器資料流程。然後，這些元件可以產生可協調資料管道的可執行程式碼。

DataOps 有助於克服資料管理的複雜性，並確保跨組織的無縫資料流程。中繼資料驅動的方法可確保資料品質和合規，以符合組織的命令。投資微服務、容器化和無伺服器函數等工具集可提高可擴展性和靈活性。

依靠資料工程團隊從資料中產生價值，並將基礎設施任務留在 day-to-day自動化中，讓組織能夠在自動化和協調方面實現卓越。資料流程管理任務近乎即時的監控和記錄支援立即的修復動作，並改善資料流程管道的效能和安全性。這些原則有助於實現可擴展性和效能，同時確保安全的資料共用模型，並為組織在未來的成功做好準備。

整合 DataOps

DataOps 是一種現代化的資料工程方法，強調開發和操作程序的整合，以簡化資料管道的建立、測試和部署。為了實作 DataOps 最佳實務，組織會使用基礎設施作為程式碼 IaC ） 和持續整合和持續交付 （ CI/CD ） 工具。這些工具支援自動化管道建立、測試和部署，可大幅提升效率並減少錯誤。DataOps 團隊與平台工程支援團隊合作來建置這些自動化，因此每個團隊都可以專注於他們最擅長的事項。

實作 DataOps 方法有助於為資料工程師、資料科學家和商業使用者建立協作環境，並快速開發、部署和監控資料管道和分析解決方案。這種方法提供跨團隊更順暢的溝通和協作，進而加快創新速度並取得更好的結果。

若要充分利用 的優點 DataOps，請務必簡化資料工程程序。這可透過使用平台工程團隊的最佳實務來實現，包括程式碼檢閱、持續整合和自動化測試。透過實作這些實務，組織可以確保資料管道可靠、可擴展且安全，並同時符合業務和技術利益相關者的需求。

佈建和協調

建立、管理核准的雲端產品目錄，並將其散發給使用者。

隨著組織的成長，以一致、可擴充且可重複的方式佈建基礎架構變得更具挑戰性。簡化的佈建與協調可協助您達成一致的控管並符合合規性需求，同時讓使用者僅部署核准的雲端產品。

在組織中重複使用預先核准的產品，可讓開發人員更快速、更一致地建置應用程式，同時滿足組織的安全性和治理需求。

Start

部署目 hub-and-spoke 錄模型

在服務目錄中管理作為產品組合的軟體資產，會以 hub-and-spoke 模式與一或多個帳戶中的使用者共用。您可以使用私有市場和私人商店來策劃各種各樣的第三方解決方案，並將其與您的基礎架構即代碼（IaC）模板一起分發。

若要讓您的建置人員能夠使用預先核准的產品，請定義要檢閱、核准並將這些產品發佈給使用者的程序。首先設計和實作包含這些預先核准產品的集中管理儲存庫。設計一個系統，當您組織中的使用者需要使用每個產品時，授與此存放庫中的授權和產品的存取權。

允許組織中的建置人員將產品提交給發行機制核准，以便在核准組織中的所有使用者之後，即可使用這些產品供組織中的所有使用者使用。

組織範本以供重複使用

當您編寫解決方案的 IaC 範本並定義您的 hub-and-spoke模型時，您應該為每個支點帳戶定義兩種範本類別：已佈建/強制執行以及可供使用。已佈建/強制執行的範本會直接從管理帳戶佈建至每個成員帳戶，做為基礎功能。可供使用的範本可供建置人員以自助服務的方式瀏覽和佈建。

套用預設參數以重複使用

實現包含構建器可以預先選擇的默認參數的 IaC 模板。如此一來，建置人員就能與控管保持一致，而不必評估每個參數的詳細資訊，並防止他們做出不正確的選擇。這種方法僅公開安裝所需的內容。例如，使用限制功能來AWS Service Catalog實作此方法，該功能可控制套用至特定產品組合中產品的規則。當產生器小組使用範本的自助佈建時，會預先設定此自訂。

建立核准程序

如果使用者有業務理由使用產品，則應該能夠提交存取未獲核准之產品的請求。建置通知系統，在使用者所使用產品的更新可供使用時通知使用者，以便他們遵循最新的安全性更新。

建立工作流程，讓建置人員透過自助服務入口網站提交新產品以供審核。建置者可以使用入口網站來定義產品的對象，並識別應具有產品存取權的使用者群組。對於每次提交，請使用您定義的程序來檢閱、核准產品，並將其發佈至自助入口網站。

提前

建立自助入口網站

建立自助入口網站，以散發、瀏覽和使用核准的雲端產品。組織中的使用者可以使用此入口網站來搜尋建置基礎結構所需的產品，以及將應用程式部署到環境中。為可存取入口網站中產品的使用者建立權限界限，並設定使用者可以使用授權產品的次數限制。在您的每個支點帳戶中設定一組可以直接佈建或以自助服務模式提供的基本資源集，因為帳戶是透過使用自訂的解決方案建立的。 AWS Control Tower

啟用私人市集

私人市場提供已購買產品（軟件，數據和專業服務）的精選目錄，並以一種 hub-and-spoke 模式實施（具有一個管理帳戶和多個成員帳戶），因此對話帳戶只能訂閱已批准的軟件。此產品治理有助於控制軟體成本，並簡化法律和合約審查。在管理帳戶層級建立私人市集，做為主要中樞。

管理權利

啟用僅允許授權使用者和工作負載在廠商定義的限制內使用授權的控制項。這有助於降低昂貴稽核和意外授權調整的風險。

Excel

與採購系統整合

透過將現有的採購流程整合到中來補充它們AWS Marketplace。這是通過將您的採購系統（Coupa 或 SAP Ariba）擴展到私有市場來完成，以便您的用戶可以遵循現有的採購和審批流程來獲得軟件。建立適當的 IAM 管理權限，用 AWS Marketplace 來產生必要的資訊來設定您的採購解決方案，並設定您的

採購解決方案以完成整合。例如，您可以設定沖孔、將採購單附加至 AWS 發票，然後調整採購程序以使用標準佈建解決方案。

使您的建設者能夠通過內部 API 訪問預先批准的產品，以便用戶可以將產品合併到其應用程序中，或為其團隊構建自己的個性化門戶以使用產品。整合提交和發佈程序以建立新產品，並允許使用者透過 API 要求新授權和存取產品。

與您的 ITSM 工具整合

如果適用，請連接 IT 服務管理 (ITSM) 工具，並自動更新您的組態管理資料庫 (CMDB)。建立程序和機制，以評估組織使用的產品。建立機制，通知使用者預先核准的產品，他們需要更新以達到合規性。使用 ITSM 工具來分析您的環境，並在需要重大更新時，將安全性和合規性更新推送到組織內的產品。

實作生命週期管理與版本發佈系統

在整個開發生命週期中，維護 IaC 範本的版本，以及從範本佈建的服務版本。您可以使用為目錄實作的 hub-and-spoke 模型，定義在網輻層級是否需要強制更新 (例如，如果並行版本可用於自助佈建)，以及哪些版本需要標示為過時。使用 hub-and-spoke 目錄也有助於視需要管理新版本的稽核和發佈。

現代應用開發

建置架構良好的雲端原生應用程式。

現代應用程式開發實務對於組織來說，建置架構良好的雲端原生應用程式並保持競爭力至關重要。企業可以使用容器和無伺服器運算等雲端原生技術，建立可擴充且靈活的應用程式，以因應不斷變化的市場需求。這些技術可讓組織最佳化資源使用率、降低成本，並改善其應用程式的效能。

當您設計現代應用程式時，請開發適用於作業和開發的敏捷解決方案。現代化的應用程式會自動回應客戶需求的變化，並且能夠抵禦故障。工程師可以快速開發和部署變更，並監控應用程式效能。現代應用程式的設計是自我修復能力，並且能夠在需要時擴展到大型或小型流量，包括零成本無流量。

建置架構良好的雲端原生應用程式需要深入瞭解基礎技術及其最佳實務。組 Organizations 應採用微服務架構，並將其應用程式設計為模組化且鬆散耦合，以實現獨立部署和可擴充性。這種方法可讓組織將其應用程式分解為更小、更易於管理的元件，這些元件可以快速且獨立地進行開發、測試和部署。

Start

探索現代化方法

從調查容器、無伺服器技術和其他能夠開發微服務的方法開始，這些方法可提高資源效率、協助改善安全性並將基礎架構支出降到最低。選擇現有差異化和企業應用程式現代化，以提高效率並將現有投資的價值最大化。根據價值主導的決策，考量重新平台 (將自我管理的容器、資料庫或訊息代理程式轉換為受管理雲端服務) 和重構 (重新開發應用程式以採用雲端原生架構)。

當您更新現有的基於雲的應用程序時，成功的方法涉及使用扼殺程序圖模式將架構逐步分解為微服務。此程序有助於採用當代的應用方法，因此您可以實現固有的好處，並向大型組織展示其價值。請考慮將您的應用程式建構為獨特的微服務，在適用情況下利用事件驅動架構。確保您的架構將無法變更的服務配額和實體資源納入考量，以避免影響工作負載效能或可靠性。

採用雲端原生運算功能

雲端原生運算功能是現代應用程式開發的關鍵。這種方法要求組織考慮其運算單元的託管方式，並為每個使用案例或服務找出最佳選項。例如，AWS Lambda提供執行應用程式程式碼的無伺服器機制，並在事件驅動架構中扮演關鍵角色。Lambda 函數會視需求啟動，parallel 執行至定義的最大並行運作，因此它們可以擴充以執行各種工作。

使用容器化

在現代軟體開發中，管理應用程式及其相依性已成為一項日益複雜的工作，尤其是當您考慮需要在各種環境中維持一致性時。為了解決這些挑戰，諸如 Docker 之類的容器化技術已成為封裝應用程式及其相依性的有效解決方案。無論應用程式的執行階段環境為何，容器都能確保一致且可重複的部署，因此本機環境中的開發行為與雲端環境中的生產開發相同。這種方法可以減少環境或其配置中的不匹配可能引起的錯誤。

使用新式資料庫

當您使用現代資料庫時，應用程式中的每個微服務都可以使用符合其需求的正確專用資料庫，進而提高靈活性和效能，同時降低成本。例如，一個微服務可能會使用 NoSQL 資料庫在儲存工作階段資料時達到高輸送量，另一個微服務可能使用關聯式資料庫執行複雜的表格聯結，而另一個微服務則可能使用量子總帳資料庫來追蹤區塊鏈的變更。

現代資料庫提供可擴充性和彈性。與傳統資料庫相比，它們還可以提供更好的安全性、合規性和可靠性。它們使組織能夠更有效地存儲和管理其數據，並確保應用程序可以在正確的時間訪問正確的數據，從而獲得更好的性能和用戶體驗。

移轉至現代資料庫是現代應用程式開發的重要組成部分。透過使用正確的資料儲存解決方案，組織可以最佳化其資料管理功能，並提供更有效率且可靠的應用程式。透過讓每個微服務獨立，並為每個微服務選擇正確的技術，組織可以進一步最佳化其資料功能，以達到最高的效率和可擴充性，同時將成本降至最低。

提前

最佳化您的現代建築

若要實現進一步的最佳化，請優化無伺 AWS 服器技術的實作，並開發可以使用 Amazon API Gateway 和.AWS Lambda 使用 Amazon Route 53 實作服務探索，並確AWS Cloud Map保元件之間的無縫通訊。

採用 API 版本控制、快取和速率限制，以維持不同應用程式版本之間的相容性和效能。透過 AWS Identity and Access Management (IAM) 和資源政策增強安全性。這些有助於確保您的基礎結構受到保護，並且僅授予授權實體存取權限。

如果可能，請使用無伺服器服務執行容器，而不必管理基礎結構。這使您能夠專注於開發核心應用程序，並允許更好的資源管理和性能。它還可以幫助您充分利用可擴展性，靈活性和成本效益的優勢。

透過深入探討無伺服器架構的複雜性，並整合這些進階實務，組織可以發掘改進和微調的機會，最終將其雲端原生應用程式的潛力發揮到極致。這種追求有助於採用更複雜的應用程式模式，進一步提升整體使用者體驗。它還使組織能夠在軟件開發過程中變得更加敏捷和高效。

使用服務網格技術

隨著組織越來越多地採用微服務架構來建置和部署應用程式，管理這些服務之間的複雜性、安全性和通訊變得至關重要。Istio、Linkerd 或 Consul 等服務網格技術在協助增強微服務的安全性、可觀察性和可靠性方面發揮著關鍵作用。

確保可見性和可追溯

現代實務在開發過程中提供了更高的可見性和可追溯性，並使其更容易遵守業界標準和最佳實務。 能見度和監控對於現代應用程式開發至關重要。實作監控與記錄解決方案，以提供應用程式效能的寶貴見解，讓組織能夠識別需要改善的領域，並將其應用程式 我們建議您與平台工程團隊合作，以確保提供可用的工具，以提供應用程式錯誤、效能和合規性的可 end-to-end見性和監控，以便快速偵測、診斷和解決問題。

Excel

擁抱微服務

對許多組織而言，現代應用程式開發是企業成功的代名詞。微服務是此轉型的核心，組織可以從擁抱這些強大的架構模式中受益。

微服務提供高度可擴展、彈性和敏捷的應用程式架構。將應用程式分解為可獨立部署的小型服務，組織可以選擇快速重複執行特定元件，而不會影響應用程式的其他部分。進階復原模式 (例如斷路器和隔板) 在確保這些應用程式的高可用性方面扮演至關重要的角色。

斷路器充當一種安全機制，可透過暫時停止或轉移來自不良服務的通訊來防止串聯故障，以便恢復。艙壁隔離資源並限制潛在故障的影響範圍。這些模式共同創造了一個強大的架構，可以承受不可預見的中斷並保持最佳性能。

實作微服務的另一個重要方面是採用領域驅動設計 (DDD) 原則。DDD 側重於創建業務領域的共享理解，並將其轉換成一個結構良好的軟件設計。這種方法導致更具凝聚力和可維護的微服務，並確保應用程序與組織的需求一步發展。

最佳化服務間通訊對於以微服務為基礎的應用程式也很重要。透過實作 gRPC 或 GraphQL 等進階通訊協定，組織可以大幅提升服務之間的通訊效率。這些通訊協定提供類型安全、低延遲和彈性等功能，有助於改善應用程式的整體效能和可維護性。

採用微服務的組織提供了促進創新、敏捷性和協同作業的環境。開發團隊通常圍繞業務能力進行組織，並且非常注重持續集成和持續交付（CI/CD）實踐。他們有能力快速做出決策、實驗和迭代，並擁抱共同責任和責任的文化。

持續整合和持續交付

比使用傳統軟體開發和基礎設施管理程序的組織更快地發展和改善應用程式和服務。

採用具有持續整合和持續交付的DevOps實務（CI/CD) promotes a streamlined, automated, and efficient process for building, testing, and deploying applications. CI/CD 可快速交付軟體、降低部署錯誤的風險，並確保應用程式隨時掌握最新功能和錯誤修正。主要目標是透過使用傳統軟體開發和基礎設施管理程序，以更快的速度發展和改善應用程式和服務。

Start

採用軟體元件管理

軟體元件管理是管理用於建置軟體的所有個別元件的實務，包括程式庫、架構、原始程式碼儲存庫、模組、成品和第三方相依性。我們建議您使用 Git 或 Apache Subversion 等版本控制系統來管理原始程式碼、啟用協同作業，以及維護程式碼變更的歷史記錄。您可以監控儲存庫中的變更和事件，以自動化程序、建立管道、管理程式碼，並視需要將工作流程與其他服務整合。

建立 CI/CD 管道

CI/CD pipelines are sets of automated instructions that are initiated by changes committed to the version control system. They typically include instructions for building the application, running automated tests, and deploying code to a specific environment. You can set up an automated CI/CD 管道，例如使用 AWS CodePipeline、Jenkins GitLab或 CircleCI 等工具。您也可以直接在支援管道產生版本的控制系統中設定這些參數。

從用於持續整合的最低可行管道開始，然後轉換至包含更多動作和階段的持續交付管道。將連續交付組態視為程式碼。您可以為每個分支和團隊使用多個不同的管道，因此請思考您需要設定哪些組態變數，以及如何最好地支援將使用管道的團隊。

考慮部署時段：您要部署程式碼的日期和時間。請考慮您系統的低需求時數，因此如果您必須復原，它對您的客戶的影響最小。其他最佳實務包括避免在星期五部署，以及在高峰值日期或假日之前實作程式碼凍結。當遞交的作者無法使用時（例如休假時），請考慮定義有關部署程式碼的規則。請記住，部署失敗，您可能需要依賴外部協助。評估不同的部署方法，例如就地部署、滾動部署、不可變部署和藍/綠部署。請考慮使用完整受管服務進行持續交付工作流程，以提高可用性和安全性，同時將複雜性和管理降至最低。

部署自動化測試

現代實務建議向左移（將測試移近開發人員和 IDE，以及在生命週期早期），以便在問題投入儲存庫並啟動管道之前偵測和修復問題。此做法涉及與開發人員的快速意見回饋循環，因為在開發人員進行編碼時偵測到錯誤。向左移與成本較低有關，因為測試不需要執行管道，這可能會導致非同步意見回饋和更高的操作費用。

自動化測試會在開發程序的早期發現錯誤，並包含單元測試、整合測試和功能測試。建議您鼓勵開發人員儘早使用工具建立單位測試，並在將程式碼推送至中央儲存庫之前執行測試。此外，請確定您的自動化程序包含靜態程式碼分析、效能基準測試和安全應用程式測試。

建立文件

除了實作CI/CD pipeline to streamline development workflows, you should maintain clear and comprehensive documentation to ensure the pipeline's ongoing effectiveness, maintainability, and scalability. Documentation is a vital aspect of CI/CD管道之外，它還為開發團隊提供了管道設計、元件和程序的清楚了解。建立文件時，請先從管道概觀開始，解釋架構和設計權衡，描述正在使用的工具和技術，指定初始組態和設定，概述安全措施和存取控制，並包含疑難排解和維護資訊。

使用基礎設施作為程式碼

使用 Terraform、Ansible 或 等工具AWS CloudFormation來管理基礎設施，並確保一致且可複製的環境。將基礎設施視為程式碼，確保您追蹤基礎設施的變更，並避免直接在主控台中進行變更。定義所有基礎設施，包括資料庫佈建程式碼，並使用管道部署這些變更。考慮在管道中以程式碼形式執行資料庫整合，其中包含一小部分已消毒生產資料。如果可能，請進行變更並追蹤程式碼中的變更。

與軟體程式碼一樣，請遵循基礎設施程式碼的下列最佳實務：

- 使用版本控制。
- 使用錯誤追蹤和票務系統。
- 在套用變更之前，請同儕檢閱變更。
- 建立基礎設施程式碼模式和設計。
- 測試基礎設施變更。

保留和追蹤標準指標

若要維持高水準的效能，請根據關鍵指標進行開發和追蹤，以了解管道的運作狀態和業務影響，包括：

- 建置頻率。建置數量可讓您深入了解團隊的生產力和變更的複雜性。
- 部署頻率。定期部署表示健康、敏捷的開發程序。
- 變更的前置時間。測量變更達到生產的平均時間，可協助您識別部署流程中的瓶頸。
- 流經管道的平均時間。從初始管道階段到每個後續階段的平均時間有助於最佳化您的工作流程。
- 生產變更磁碟區。追蹤到達生產環境的變更數量，可以提供生產環境穩定性的洞見。
- 建置時間。平均建置時間可以指出程式碼庫或基礎設施中的潛在問題。

進階

使用組態管理

組態管理工具在自動化部署、組態和管理軟體和基礎設施方面扮演著重要角色。它們提供系統化方法，以處理變更，並維護各種環境中基礎設施、軟體和組態的所需狀態。這些工具可讓開發人員使用宣告性或強制性語言來定義系統所需的狀態。然後，組態管理工具會自動將這些組態套用至目標系統的程序，以確保一致性和可重複性。

使用組態管理工具自動化軟體和基礎設施的部署、組態和管理。AWS Systems Manager State Manager 是一種安全且可擴展的組態管理服務，可自動化將受管節點和其他 AWS 資源保持在您定義狀態的程序。

整合監控和記錄

將監控和記錄解決方案整合到 CD 管道中可為開發團隊和整體軟體開發程序帶來許多好處。這些解決方案可以提供應用程式效能的即時洞見、更快速地識別和解決問題，並促進持續改進，以協助確保應用程式在其生命週期中保持可靠、高效能和可擴展性。投資監控和記錄解決方案是維護強大且高效的 CD 管道的關鍵層面，最終有助於高品質軟體的成功交付。

建立合併的節奏

承諾或合併程式碼至少每天變更主線（區塊或主線）分支一次，或者最好是，在每項任務之後，每天變更多次。此節奏會導致多個每日管道調用。拉式分支工作流程模型與此方法保持一致。使用特徵標記、暗啟動 和類似技術來自訂客戶使用的功能。

擷取部署後的行為

部署之後，請使用自動化合成測試擷取生產行為，並將結果與連續交付管道同步，以確保立即執行修正動作。開發人員的首要任務應該是盡快修正管道中發現的錯誤、將程式碼變更遞交至原始程式碼儲存庫，以及驗證管道中的錯誤解決方案。

部署後的最佳做法包括觀察最重要的關鍵效能指標（KPIs），並驗證生產環境中沒有錯誤。自動化錯誤處理和評估部署後KPIs，以量化版本的影響。自動產生速度、安全性和穩定性指標，開發人員可以使用這些指標進行改善。如需詳細資訊，請參閱 上的解決方案DevOps 監控儀表板 AWS。

Excel

採用尖端實務和技術以獲得最佳效能。持續改進 CI/CD 程序可協助您改善軟體品質、縮短上市時間，並提高靈活性。新的技術和工具會不斷出現，這使得您的組織必須隨時掌握最新資訊並加以調整，以維持競爭優勢。

若要保持適應性，請考慮下列事項：

- 將所有內容定義為程式碼，包括您的應用程式、組態、基礎設施、資料、 AWS 帳戶和組織、部署管道、聯網，以及安全和合規控制。
- 為運算映像、共用服務和應用程式建立對應的部署管道。
- 考慮一種 GitOps 模型，其中提取型請求會透過將現有基礎設施狀態與所需狀態進行比較來啟動工作流程來部署變更，如程式碼中所述。
- 考慮使用 CD 管道來部署機器學習（ML）、資料、物聯網（IoT和其他工作負載。
- 以數位方式簽署所有建置成品，並將其存放在安全的儲存庫中。
- 透過自動產生軟體物料清單來追蹤軟體的驗證，該清單會建立部署給客戶的所有版本化和數位簽章成品的記錄。
- 消除軟體交付程序中的所有手動活動後，請移除手動檢閱板。

對於已自動化整個軟體交付程序的應用程式和服務，請考慮持續部署，其中團隊部署變更，將管道中的所有檢查傳遞給生產中的客戶。如需視覺化，請參閱 網站上的什麼是持續交付？ AWS 的第一個圖表。

整合 AI/ML 技術

將人工智慧（AI）和機器學習（ML）技術整合到 CI/CD 管道中具有多種優點，包括下列各項：

- 自動化測試產生
- 智慧測試優先順序
- 偵測問題的預測性分析
- 異常偵測和根本原因分析
- 程式碼檢閱和品質保證

- 部署最佳化

如需詳細資訊，請參閱 AWS 網站上的將情報新增至開發人員操作。

採用混亂工程實務

Chaos 工程涉及刻意將故障注入系統，以測試其承受意外事件並從中復原的能力。透過識別弱點並主動解決它們，組織可以提高其整體系統可靠性，並將潛在問題的影響降至最低。

採用混亂工程實務，使用 Gremlin、Chaos Monkey 或 Litmus 等工具來測試系統的彈性。定期執行受控實驗，以識別漏洞、驗證容錯能力，並確保您的應用程式正常處理意外故障。這種主動方法有助於提高系統可靠性，並有助於更強大的 CI/CD 管道。

最佳化效能

使用分析工具、即時監控和回饋循環，持續最佳化應用程式的效能。套用下列技術，以確保您的應用程式可以處理增加的流量和需求：

- 程式碼最佳化
- 分析
- 即時監控
- 回饋迴圈
- 快取
- 負載平衡
- 可擴展性和效能測試

實作進階可觀測性

提升雲端基礎設施的可觀測性，不僅止於收集、彙總和分析指標、日誌和追蹤的基礎。當使用 Amazon CloudWatch 和 等工具增強可觀測性時AWS X-Ray，它會演變為策略實務，以推動持續交付和創新。

在強大的 CI/CD 管道中，進階的可觀測性可讓您探索洞察，不只是關於應用程式和基礎設施，還包括關於整個系統的效能和運作狀態，包括管道本身。這些洞見可協助您：

- 快速識別、了解和解決潛在問題，以改善應用程式穩定性並減少停機時間
- 簡化您的 CI/CD 程序，以建立更快且更可靠的交付
- 深入了解程式碼變更和部署的影響，以推動明智的決策

- 最佳化資源使用率，以提高營運效率和成本效益

若要提升可觀測性：

- 將可觀測性嵌入應用程式和基礎設施的每個層面，以建立系統效能、行為和運作狀態的全面檢視。
- 使用 Amazon 等工具集中資料收集、儲存和分析 CloudWatch ，統一您的可觀測性資料，以便輕鬆存取和解讀。
- 使用 AWS X-Ray 進行分散式追蹤，以了解您的應用程式及其基礎服務的表現。
- 建立回饋循環以持續改進，並使用可觀測性資料來驅動系統的迭代增強。

採用進階可觀測性不僅在於維護系統，更是實現卓越營運並推動組織中持續創新的策略。

實作 GitOps 實務

實作 GitOps 實務，使用 Git 儲存庫作為單一事實來源來管理基礎設施和應用程式組態。此方法可簡化變更管理、增強可追蹤性，並確保跨環境的一致性。

結論

本指南是成功實作和管理雲端成功採用基礎的教戰手冊。它討論瞭如何：

- 直接解決平台架構中的技術挑戰和複雜性，為您的雲端環境及其中的資料建立可靠的準則和原則。
- 利用強大的佈建和協調流程來建置平台工程。
- 允許使用符合規範的多帳戶雲端環境，以可擴展且可重複的方式管理核准的雲端產品並將其分發給使用者。
- 使用資料工程所需的工具來 Support 資料架構決策，以推動資料導向的決策。
- 將這些功能與現代化的應用程式開發策略和 CI/CD 程序搭配使用，以提升組織內的敏捷性、效率和創新能力。
- 在您自己的決策中建立跨職能關係，並從其他 AWS CAF 角度獲取投入，以確保您的平台及其背後團隊的成功。

深入閱讀

AWS 雲端採用架構 (AWS CAF) 資源：

- 電子書
- 有聲書
- 資訊圖
- AWS CAF 應用於人工智慧、Machine Learning 和生成 AI
- 業務角度
- 人的角度
- 治理角度
- 運營角度
- 安全性觀點

其他資源：

- AWS 建築中心
- AWS 案例研究
- AWS 一般參考
- AWS 詞彙表
- AWS 知識中心
- AWS 規定指引
- AWS 合作夥伴解決方案 (舊稱為「快速
- AWS 安全性文件
- AWS 解決方案庫
- AWS 訓練與認證
- AWS Well-Architected
- AWS 白皮書和指南
- 開始使用 AWS
- Amazon Web Services 概述

貢獻者

本指南的貢獻者包括：

- 托尼·聖地亞哥，高級合夥人解決方案 AWS
- 馬蒂亞斯·文德, 企業技術專家, AWS
- 亞歷克斯托雷斯，高級解決方案架構師 AWS
- 邁克爾·萊恩德格, 高級 DevSecOps 顧問, AWS
- 亞歷克斯利文斯頓，首席解決方案架構師和 CloudOps 專家 AWS
- 布魯斯·庫珀, 主要 SDE, AWS
- 拉文德·托塔, 高級諮詢顧問, AWS
- 高級實踐經理桑桑雅梓 AWS
- 保羅·杜瓦爾, 主任, DevSecOps AWS
- 傑里米·坦南特，首席雲端交付經理， AWS
- 斯尼沙, 主要基礎設施領導, AWS
- 現在, 全球領先, AWS 雲端採用框架, 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 10 月 25 日

AWS 規範指南詞彙表

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

數字

7 R

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

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

A

ABAC

請參閱屬性型存取控制 。

抽象服務

請參閱 受管服務 。

ACID

請參閱原子、一致性、隔離、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工作。

主動-被動式遷移

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

彙總函數

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

AI

請參閱人工智慧。

AIOps

請參閱人工智慧操作 。

匿名化

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

反模式

經常性問題的常用解決方案，其解決方案具有反效益、無效或效果不如替代方案。

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體侵害。

應用程式組合

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

人工智慧 (AI)

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

人工智慧操作（AIOps）

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

非對稱加密

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

原子度、一致性、隔離、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ABAC AWS中的 。

權威性資料來源

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

可用區域

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

AWS 雲端採用架構（AWS CAF）

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

AWS 工作負載資格架構（AWS WQF）

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

B

錯誤的機器人

旨在中斷或傷害個人或組織的機器人。

BCP

請參閱業務持續性規劃 。

行為圖

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

大端序系統

首先儲存最高有效位元組的系統。另請參閱端點 。

二進制分類

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

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

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

機器人

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

殭屍網路

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

分支

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

碎片存取

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

棕地策略

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

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

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

業務連續性規劃（BCP）

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

C

CAF

請參閱AWS 雲端採用架構 。

Canary 部署

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

CCoE

請參閱 Cloud Center of Excellence 。

CDC

請參閱變更資料擷取 。

變更資料擷取（CDC）

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

混亂工程

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

CI/CD

請參閱持續整合和持續交付。

分類

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

用戶端加密

在目標 AWS 服務 接收資料之前，在本機加密資料。

Cloud Center of Excellence（CCoE）

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

雲端運算

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

雲端操作模型

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

採用雲端階段

組織在遷移至 時通常會經歷的四個階段 AWS 雲端：

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

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

CMDB

請參閱組態管理資料庫 。

程式碼儲存庫

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

冷快取

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

冷資料

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

電腦視覺 （CV）

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

組態偏離

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

組態管理資料庫 （CMDB）

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

一致性套件

您可以組合的 AWS Config 規則和修復動作集合，以自訂合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶 和 區域中或整個組織中的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的一致性套件。

持續整合和持續交付 (CI/CD)

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

CV

請參閱電腦視覺 。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

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

資料偏離

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

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

架構架構架構，提供分散式、分散式的資料擁有權，並具有集中式管理和治理。

資料最小化

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

資料周邊

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

資料預先處理

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

資料來源

追蹤資料整個生命週期的原始伺服器和歷史記錄的程序，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。

資料庫操作語言 （DML）

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱資料庫定義語言 。

深度整體

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

深度學習

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

defense-in-depth

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

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

委派的管理員

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

部署

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

開發環境

請參閱環境 。

偵測性控制

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

開發值串流映射 （DVSM）

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

數位分身

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

維度資料表

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

災難

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

災難復原 （DR）

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

DML

請參閱資料庫操作語言 。

領域驅動的設計

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

DR

請參閱災難復原。

漂移偵測

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

邊緣運算

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

加密

將純文字資料轉換為人類可讀取的運算程序。

加密金鑰

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

端序

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

端點

請參閱服務端點 。

端點服務

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

企業資源規劃 （ERP）

可自動化和管理企業關鍵業務流程 （例如會計MES、 和 專案管理 ） 的系統。

信封加密

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

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

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

epic

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

ERP

請參閱企業資源規劃 。

探索性資料分析（EDA）

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

F

事實資料表

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

快速失敗

使用頻繁且增量測試來縮短開發生命週期的哲學。這是靈活方法的關鍵部分。

故障隔離界限

在 中 AWS 雲端，限制失敗效果並協助改善工作負載彈性的邊界 AWS 區域，例如可用區域、控制平面或資料平面。如需詳細資訊，請參閱AWS 故障隔離界限 。

功能分支

請參閱分支 。

特徵

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

功能重要性

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

特徵轉換

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

FGAC

請參閱精細存取控制 。

精細存取控制（FGAC）

使用多個條件來允許或拒絕存取請求。

閃切遷移

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

G

地理封鎖

請參閱地理限制。

地理限制 (地理封鎖)

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

Gitflow 工作流程

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

綠地策略

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

防護機制

高階規則，可協助管理組織單位（）之間的資源、政策和合規性OUs。預防性防護機制會強制執行政策，以確保符合合規標準。它們是透過使用服務控制政策和IAM許可界限來實作。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。其實作方式是使用 AWS Config、 AWS Security Hub、Amazon GuardDuty、 AWS Trusted Advisor、Amazon Inspector 和自訂 AWS Lambda 檢查。

H

HA

請參閱高可用性。

異質資料庫遷移

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

高可用性 （HA）

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

歷史現代化

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

異質資料庫遷移

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

常用資料

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

修補程序

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

超級護理期間

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

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，其定義其在 AWS 雲端 環境中的許可。

閒置應用程式

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

IIoT

請參閱工業物聯網。

不可變的基礎設施

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

傳入（傳入）VPC

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

增量遷移

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

工業 4.0

由 Klaus Schwab 於 2016 年推出的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進步，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，請參閱建置工業物聯網（IIoT）數位轉型策略。

檢查 VPC

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

物聯網 (IoT)

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

可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱使用 的機器學習模型可解譯性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 提供 的基礎ITSM。

IT 服務管理（ITSM）

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

ITIL

請參閱 IT 資訊庫 。

ITSM

請參閱 IT 服務管理。

L

標籤型存取控制（LBAC）

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

登陸區域

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

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱標籤型存取控制 。

最低權限

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

隨即轉移

請參閱 7 Rs 。

小端序系統

首先儲存最低有效位元組的系統。另請參閱永久性 。

較低的環境

請參閱環境 。

M

機器學習 (ML)

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

主要分支

請參閱分支 。

惡意軟體

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

受管服務

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

製造執行系統 （ MES ）

用於追蹤、監控、記錄和控制生產程序的軟體系統，可將原物料轉換為工廠的成品。

MAP

請參閱遷移加速計畫 。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在運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 中組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

MES

請參閱製造執行系統 。

訊息佇列遙測傳輸 （ MQTT ）

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

微服務

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

微服務架構

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

Migration Acceleration Program （MAP）

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

大規模遷移

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

遷移工廠

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

遷移中繼資料

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

遷移模式

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

遷移產品組合評估 （MPA）

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

遷移就緒狀態評估 （MRA）

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

遷移策略

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

機器學習 (ML)

請參閱機器學習。

現代化

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

現代化準備程度評定

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

單一應用程式 (單一)

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

MPA

請參閱遷移產品組合評估 。

MQTT

請參閱訊息佇列遙測傳輸 。

多類別分類

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

可變基礎設施

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

O

OAC

請參閱原始存取控制。

OAI

請參閱原始伺服器存取身分。

OCM

請參閱組織變更管理。

離線遷移

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

OI

請參閱 操作整合。

OLA

請參閱操作層級協議。

線上遷移

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

OPC-UA

請參閱開啟程序通訊 - Unified Architecture。

開放程序通訊 - Unified Architecture（OPC-UA）

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

操作層級協議（OLA）

闡明哪些功能性 IT 群組承諾交付給彼此的協議，以支援服務層級協議（SLA）。

操作預備檢閱（ORR）

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

操作技術（OT）

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

操作整合 (OI)

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

組織追蹤

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

組織變更管理（OCM）

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

原始存取控制（OAC）

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

原始存取身分（OAI）

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

ORR

請參閱操作預備檢閱 。

OT

請參閱操作技術 。

傳出（輸出）VPC

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

P

許可界限

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

個人身分資訊（PII）

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

PII

請參閱個人識別資訊 。

手冊

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

PLC

請參閱可程式邏輯控制器 。

PLM

請參閱產品生命週期管理。

政策

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

混合持久性

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

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

組合評定

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

述詞

傳回 `true`或 的查詢條件`false`，通常位於`WHERE`子句中。

述詞下推

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

預防性控制

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

委託人

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

設計隱私

一種系統工程方法，在整個工程過程中將隱私權納入考量。

私有託管區域

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

主動控制

旨在防止部署不合規資源的安全控制。這些控制項會先掃描資源，然後再佈建。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的控制項參考指南，並請參閱在 上實作安全控制項中的主動控制項。 AWS

產品生命週期管理 （PLM）

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

生產環境

請參閱環境 。

可程式設計邏輯控制器（PLC）

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

擬匿名化

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

publish/subscribe (pub/sub)

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

Q

查詢計劃

一系列步驟，如指示，用於存取SQL關聯式資料庫系統中的資料。

查詢計劃迴歸

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

R

RACI 矩陣

請參閱負責、負責、已諮詢、知情（RACI）。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

RASCI 矩陣

請參閱負責、負責、已諮詢、知情（RACI）。

RCAC

請參閱資料列和資料欄存取控制。

僅供讀取複本

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

重新架構師

請參閱 7 Rs。

復原點目標（RPO）

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

復原時間目標（RTO）

服務中斷和服務還原之間的可接受延遲上限。

重構

請參閱 7 Rs。

區域

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

迴歸

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

重新託管

請參閱 7 Rs。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新定位

請參閱 7 Rs。

轉譯形式

請參閱 7 Rs。

回購

請參閱 7 Rs。

彈性

應用程式抵抗中斷或從中斷中復原的能力。在 中規劃復原能力時，高可用性和災難復原是常見的考量事項 AWS 雲端。如需詳細資訊，請參閱AWS 雲端 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。

負責任、負責、已諮詢、知情（RACI）矩陣

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

回應性控制

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

保留

請參閱 7 Rs 。

淘汰

請參閱 7 Rs 。

輪換

定期更新秘密的程序，讓攻擊者更難存取憑證。

資料列和資料欄存取控制（RCAC）

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

RPO

請參閱復原點目標 。

RTO

請參閱復原時間目標 。

執行手冊

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

S

SAML 2.0

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

SCADA

請參閱 監控控制和資料擷取 。

SCP

請參閱服務控制政策 。

秘密

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

安全控制

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

安全強化

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

安全資訊和事件管理（SIEM）系統

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

安全回應自動化

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

伺服器端加密

由接收資料的 AWS 服務 加密其目的地的資料。

服務控制政策（SCP）

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

服務端點

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

服務層級協議（SLA）

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

服務層級指標（SLI）

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標（SLO）

代表服務運作狀態的目標指標，由服務層級指標 測量。

共同責任模式

描述您與 共同 AWS 承擔雲端安全與合規責任的模型。 AWS 負責雲端的安全，而 負責雲端的安全。如需詳細資訊，請參閱共同責任模式。

SIEM

請參閱安全資訊和事件管理系統 。

單一失敗點（SPOF）

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱服務層級協議 。

SLI

請參閱服務層級指示器 。

SLO

請參閱服務層級目標 。

split-and-seed 模型

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

SPOF

請參閱單一失敗點 。

星狀結構描述

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

Strangler Fig 模式

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

子網

中 IP 地址的範圍VPC。子網必須位於單一可用區域。

監控控制和資料擷取 （SCADA）

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

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

T

標籤

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

目標變數

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

任務清單

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

測試環境

請參閱環境 。

訓練

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

傳輸閘道

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

主幹型工作流程

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

受信任的存取權

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

調校

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

雙比薩團隊

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

U

不確定性

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

未區分的任務

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

較高的環境

請參閱環境 。

V

清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

VPC 對等

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

漏洞

損害系統安全性的軟體或硬體缺陷。

W

暖快取

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

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等慢的查詢。

視窗函數

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

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

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

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）

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

Z

零時差漏洞

利用零時差漏洞的攻擊，通常是惡意軟體。

零時差漏洞

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

殭屍應用程式

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

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。