在 上操作代理式 AI AWS

AWS 方案指引

AWS 方案指引: 在 上操作代理式 AI AWS

Table of Contents

在 上操作代理式 AI AWS

Aaron Sempf、Brad Ryan、Bhargs Srivathsan 和 Akhil Bhaskar，Amazon Web Services

2025 年 8 月 (文件歷史記錄)

代理式 AI 不是功能，而是新的操作範例。投資於有紀律的架構、信任架構和符合業務的部署模型的組織將領導新一代的適應性智慧企業。

代理式 AI 代表自主軟體代理程式和生成式 AI 的收斂。它將客服人員的決策和目標導向行為與大型語言模型 (LLMs) 的語言理解和產生功能融合在一起。這些客服人員可以在動態企業環境中進行推理、行動、調整和協作。若要操作此潛力，企業必須將其思維從模型部署轉移到客服人員基礎設施。

本指南提供組織策略，將代理式 AI 從隔離實驗轉換為企業規模、創造價值的基礎設施。它可協助您在具有控管、可擴展性和業務一致性的工作流程中嵌入智慧型代理程式。

重點領域和建議

本指南在操作代理式 AI 時，著重於下列基本領域。針對每個重點領域提供組織和業務建議：

- 重點區域 1：釐清客服人員意圖和範圍 – 使客服人員符合業務優先事項和認知瓶頸。將客服人員視為數位團隊成員，而不只是工具。
- 重點領域 2：設計可編譯性和協同合作 – 透過任意代理程式接受具有模組化架構、語意通訊協定和動態委派的多代理程式系統。
- 重點區域 3：多租戶和控制的架構師 – 使用共用代理程式服務、集中式控管和角色型存取來建置可擴展的租戶感知基礎設施。
- 重點領域 4：透過身分、護欄和可觀測性建立信任 – 強制執行可追蹤性、執行時間控制和可解釋性，以獲得利益相關者的信任。
- 重點區域 5：管理生命週期 – 建立持續整合和持續部署 (CI/CD) 管道、提示版本控制、遙測和持續重新訓練，以支援代理式 AI 效能和效率。
- 重點區域 6：使客服人員模型與業務模型保持一致 – 透過以用量為基礎的模型、內部 ROI 指標和商業產品來獲利代理程式功能。

您可以使用本指南中的建議，為您的業務大規模代理式 AI 做好準備。其中概述了組織必須如何針對客服人員 AI 進行重組，包括為客服人員 (AgentOps) 團隊建置 DevOps、可互通系統，以及擴展採用率的變更管理策略。它強調決策優先的思維，並與 AWS Well-Architected Framework 保持一致。

目標對象

本指南適用於企業架構師、AI/ML 工程主管，以及正在設計和擴展代理系統、將 AI 嵌入核心業務工作流程，以及在生產環境中操作 LLMs 和自動代理程式的數位轉型策略師。若要了解本指南中的概念和建議，您應該熟悉現代雲端原生架構和分散式系統、大型語言模型、基礎模型功能，以及 AI 控管、DevOps 和平台工程的原則。

目標

透過實作本指南中的建議，您的組織可以實現下列業務成果：

- 透過自動化的目標導向代理程式加速決策和工作流程執行，以減少人力瓶頸和認知負載。
- 透過可重複使用的多租用戶代理程式平台，跨業務單位進行可擴展且符合成本效益的智慧型功能部署。
- AI 系統中的彈性、信任和管理能力更高，可在受監管、關鍵任務或面對客戶的環境中自信地採用。

關於此內容系列

本指南是代理式 AI 相關系列的一部分 AWS。如需詳細資訊和檢視此系列中的其他指南，請參閱 AWS 規範性指導網站上的 客服人員 AI 。

代理式 AI 的策略基礎

代理系統不是新的。軟體代理程式，包括機器人程序自動化 (RPA) 和決策引擎，已存在數十年。但它們簡單且決定性，旨在遵循預先定義的規則和符號邏輯來執行重複的低變異任務。隨著生成式 AI 的增加，遊戲已變更。大型語言模型 (LLMs) 現在可解譯複雜的輸入、動態產生回應，以及快速合成知識。您現在可以擴展機構，無需使用易碎或硬式編碼的邏輯。現在，客服人員可以推理、做出決策、叫用工具、適應內容，以及跨工作流程與其他客服人員協調。他們可以自動朝目標運作、維護記憶體，以及反映結果。

不過，原始功能還不夠。沒有整合的智慧會產生創新，而不是影響。若要從強大的 LLMs釋放價值，企業必須從隔離的實驗轉移到工程生態系統。代理程式必須視為生產級服務，其運作方式與任何企業系統相同。這包括控管、可觀測性、安全身分模型和生命週期管理。它們還必須產生真正的業務成果，而不是投機潛力。這些系統的架構應具有明確的決策和容錯界限。請務必整合自動化復原機制、即時效能監控和可擴展的資源管理。這可協助您處理客服人員互動的動態、非確定性質，同時在整個企業工作流程中維持一致的服務水準。

在基礎層面上，企業必須重新思考如何將智慧嵌入營運結構中。代理程式的設計必須能夠與核心系統整合、遵守企業政策，並提供可衡量的值。他們需要跨部門、網域和使用者內容大規模運作。代理程式 AI 的操作最終與使用有關；部署 AI 來執行隔離任務和部署代理程式以發展您的商業模型之間存在差異。

代理式 AI 代表一種新的營運理念，需要我們如何處理系統、程序和人員以在整個組織中擴展智慧的基本轉移。代理程式會成為可擴增人類能力的策略資產。透過將代理式 AI 整合到其操作中，組織可以釋放可推動商業價值、增強人類功能和最佳化複雜工作流程的洞察。

代理式 AI 的策略重點領域

為了從早期原型轉向生產級和價值產生系統，團隊需要融合架構、程序和產品思維的一致策略。

許多組織仍然以工具優先或以模型為中心的思維模式接近 AI。生成式 AI 已擴增實驗，但通常沒有明確符合業務策略或可衡量的結果。如果沒有定義的策略角色，客服人員可能會成為耗盡資源而非提供可擴展價值的新實驗。若要建立代理式 AI 的策略角色，組織必須從業務優先順序開始。識別認知過載、決策瓶頸或分段工作流程的領域，其中自主性可以提供緩解。使用網域特定的問題陳述式來塑造客服人員的責任。將客服人員視為數位團隊成員，而不是可以推理、委派和調整的工具。

決策科學是結合資料科學、分析和行為建模以改善決策的紀律。它應該在客服人員架構程序的早期整合，使設計與業務成果保持一致。透過識別決策模式、模擬權衡和量化價值影響，決策科學可協助您找出客服人員自主權可以提供最高價值的位置。決策科學可以加速決策、減少錯誤並啟用即時調整。此資料資訊基礎以可衡量的洞見為基礎，可更緊密地整合現有企業技術，例如規則引擎、分析平台和預測模型。

為了協助建立客服人員的策略角色，本節介紹了構成客服人員 AI 營運骨幹的基礎重點領域。每個映射到核心任務，從技術領導者、架構師或產品擁有者的角度來看，他們負責如何構想和設計客服人員。這些重點區域不是循序步驟。每個 都值得在整個系統生命週期中重新檢視，以培養彈性、可擴展且可獲利的代理程式生態系統。

本節包含下列重點領域：

- 重點區域 1：釐清客服人員意圖和範圍
- 重點領域 2：設計可編譯性和協同合作
- 重點區域 3：多租戶和控制的架構師
- 重點領域 4：透過身分、護欄和可觀測性建立信任
- 重點區域 5：管理生命週期
- 重點區域 6：使客服人員模型與業務模型保持一致

重點區域 1：釐清客服人員意圖和範圍

待完成的任務：「協助我確保每個客服人員解決具有明確界限的真實問題，而不只是冷靜的示範。」

代理式 AI 不只是建置功能。這是關於以正確的方式解決正確的問題，以獲得正確的結果。首先完全清楚代理式 AI 解決方案的意圖。

策略

組織通常從模型可以執行的操作開始（例如呼叫 APIs、回答問題或產生摘要），並對其修改使用案例。這會導致範圍模糊、整合不佳，以及技術上令人驚豔但操作上無用的客服人員。相反地，首先透過如下的特定問題來定義客服人員的角色：

- 客服人員負責哪些特定結果？
- 代表誰行事？
- 誰會受益？
- 客服人員的自主性從哪裡開始和結束？
- 失敗時會發生什麼情況？

範圍良好的客服人員具有明確的任務、定義的責任和可衡量的成功條件。請勿將客服人員視為助理或聊天機器人。反之，請為其提供任務標題。將其視為客戶成功代理程式、產品傳回處理常式或合規監控。

吸引利益相關者或客戶時，請強調代理式 AI 系統的可擴展性和適應性。這些客服人員隨著業務發展，透過學習和意見回饋持續改善。為了減少阻力並加速採用，請強調代理工具如何以工作者同情心設計。它們提供建立信任的透明度、控制和選用覆寫機制。客服人員不會取代人員，而是增強人力能力和決策，協助員工保持循環並專注於高價值的任務。

成功實作的關鍵是使代理式 AI 與特定、高影響的業務成果保持一致。鼓勵團隊和合作夥伴從解決可見困擾的重點試驗專案開始。快速獲勝會產生可衡量的投資報酬率 (ROI)、建立內部接受度，並建立動力以廣泛採用。

為了引導採用和成熟度，組織可以沿著演進模型架構代理程式設計。客服人員自主權、複雜性和業務影響會逐漸增加。以下是此模型的階段：

- 觀察者代理程式會呈現來自雜訊的洞見。例如，市場情緒代理程式，可跨數位管道追蹤品牌感知。
- 助理客服人員支援人為決策。一個範例是交易諮詢代理程式，可合成銷售團隊的競爭對手資料和市場條件。
- 自治代理程式會在定義的界限內獨立運作。範例是資源配置代理程式，可根據需求動態調整雲端基礎設施。
- Orchestrator 代理程式會協調多代理程式工作流程。例如，供應鏈最佳化代理程式，可管理庫存、物流和預測代理程式之間的互動。
- 創新者代理程式會產生新的策略可能性。例如，商業模型創新代理程式，可分析市場趨勢並建議新的收入串流。

這些策略成果和成熟度層級的架構代理程式可提高焦點、加速採用，並建立利益相關者的信心。

為了支援在此焦點區域中保持一致 AWS 服務，例如 Amazon Quick， 可以視覺化連結至客服人員驅動成果的關鍵效能指標 (KPIs)。您可以使用 Amazon CloudWatch 近乎即時地監控客服人員行為、效能指標和系統運作狀態。使用操作意見回饋來調整客服人員互動和資源使用。 AWS CloudTrail可以在早期實驗和精簡階段提供客服人員活動和整合模式的可見性。

定義意圖和範圍的商業價值

採用代理式 AI 代表組織如何處理數位轉型和卓越營運的關鍵轉變。這不只是自動化。這是關於啟用智慧型自主權，以加速決策和價值實現。

主要業務驅動因素包括下列項目：

- 競爭優勢 – 早期採用者透過更快的洞察、更好的服務和適應性操作獲得策略優勢。
- 客戶體驗增強 – 客服人員提供即時、個人化的全年無休支援，以提高滿意度和忠誠度。
- 營運效率 – 代理式 AI 透過自動化複雜、重複的決策任務，大幅降低人類認知負載。這可讓員工專注於更高價值的活動，並降低成本。

跨產業的實際使用案例包括下列項目：

- 金融服務 – AI 代理器可以提供個人化的財務建議並偵測詐騙。
- 醫療保健 – 分類和治療計畫代理程式可以改善臨床輸送量。
- 零售 – 客服人員可以擔任智慧型購物助理或即時最佳化庫存。
- 製造 – 客服人員可以執行預測性維護或協調供應鏈。

重點領域 2：設計可編譯性和協同合作

待完成的任務：「讓我像建置服務一樣建置客服人員 - 模組化且可測試，以便視需要進行撰寫和協調。」

許多 AI 工作一開始都是單體、以模型為中心的試驗。它們很有用，但很難跨網域擴展或適應複雜的問題。當這些代理程式設計為互通時的值複合。在技術中，可編譯性是結合模組化元件以建立靈活、可擴展的解決方案，以適應變化。如果沒有可編寫性，智慧在特定工作流程中會遭到鎖定。此外，客服人員協同合作引入了傳統自動化團隊可能無法處理的協同運作、狀態管理和通訊協定溝通複雜性。

策略

接受多代理程式範例。組織部門之類的模型代理程式：模組化、專門和可互通。定義明確的界面、共用內容格式和標準通訊協定，例如模型內容通訊協定 (MCP) 或 Agent2Agent (A2A)。採用多代理程式協同運作模式，例如扭曲、圖形或階層協調。這些模式可協助客服人員根據任務結構和信任層級，以動態方式從彼此探索功能和請求服務，無論是平行、循序或共識驅動的工作流程。

若要提升可擴展且受管的協同合作，請使用 仲裁代理程式。這類代理程式是一種中性授權機構，可根據已知功能和備用策略促進任務委派。雖然 不是集中式控制器，但任意代理程式在信任和合規方面扮演重要角色。它可確保敏感或受管制的任務只會路由至符合身分和政策要求的客服人員。它可做為政策繫結工作流程的 Gatekeeper。它強制執行隔離並啟用可解釋的委派。最重要的是，任意代理程式不是瓶頸；它與以水平peer-to-peer方式操作的自我協調代理程式共存。這些客服人員會委派子任務、共用內容，以及直接解析相依性。

此混合模型支援確定性指派（透過仲裁代理程式）和緊急協同合作。它將結構與彈性混合。在此架構中，客服人員可以分類為下列特殊角色：

- 決策代理程式，例如政策強制執行者、資源分配器和風險評估者
- 知識代理程式，例如內容彙整工具、模式辨識器和異常偵測器
- 執行代理程式，例如任務執行器、品質控制器和整合管理員

若要有效協調，多代理程式系統必須支援強大的互動通訊協定，以進行狀態管理、故障復原和衝突解決。即使代理程式獨立運作，這也會提高穩定性和責任。

建立明確的擴展規則，例如負載型代理程式執行個體化、內容感知資源配置，以及自動化功能探索和註冊。這些措施有助於系統動態成長，以回應需求或複雜性。

將代理程式設計為分散式傳訊底線中ready-to-use模組。例如，您可以使用 Amazon EventBridge 搭配 A2A 或 MCP，而非孤立的服務。採用版本控制、CI/CD 管道和代理程式範本，以支援系統穩定性，同時加速內部採用和生命週期演變。鼓勵程式碼重複使用和標準化，以減少整合摩擦並提升彈性生態系統。

協同合作是一種力乘數。它解鎖了跨多代理程式環境的擴展、專業化和彈性。為了支援此動態協同合作，組織應該建構輕量型控制平面以進行客服人員協調。此控制平面包含下列項目：

- 功能登錄檔，定義每個代理程式可執行的操作，並支援對等探索的版本化中繼資料
- 使用 arbiter 或主管代理程式根據內容、可用性和政策路由任務的任務仲裁邏輯
- 啟用即時決策內容和安全交接的生命週期和狀態追蹤

控制平面可確保多代理程式系統保持可擴展性、政策一致且容錯能力，而不會集中授權或減慢操作速度。

不過，多代理程式環境也會帶來營運挑戰。維護客服人員互動之間的內容、管理共用狀態，以及協調動作都可能提高複雜性和成本。如果您在客服人員間通訊期間使用取用權杖的 LLMs，成本可能會增加。這些成本必須與大規模智慧自主的複合商業優勢權衡。

若要解決這些挑戰，請考慮抽象關鍵問題的代理程式平台，例如：

- 標準化通訊協定和語意格式
- 內建協同運作邏輯和動態路由
- 客服人員之間的共用內容和記憶體管理
- 故障期間的備用處理和正常降級

對於採用多代理程式策略的團隊，最佳方法是從小規模開始並設計規模。從解決實際問題的目標單一代理程式解決方案開始。然後，逐步將這些客服人員組成合作的系統，讓每個客服人員都可以根據共用目標和整個系統的內容來探索、協調和委派。

重要的是，強大的錯誤處理和正常降級必須是主要設計原則。當客服人員無法使用或失敗時，多客服人員系統應該能夠繼續部分工作流程或啟動備份邏輯。這可提升可靠性，無需剛性耦合。

AWS 服務 提供強大的功能來大規模支援此架構。Amazon EventBridge 和 EventBridge 管道為多代理程式訊息提供結構化、事件驅動的骨幹。為了管理模組化行為， AWS AppConfig可啟用跨代理程式執行個體的安全動態組態切換。若要支援共用內容和記憶體管理，請使用 Amazon DynamoDB 進行輕量化的租用戶感知狀態持久性和跨客服人員的快速內容擷取。您可以使用 Amazon Simple Storage Service (Amazon S3) 來儲存結構化提示歷史記錄、共用成品或代理程式產生的輸出。對於需要有狀態協調的更複雜工作流程， AWS Step Functions可以使用檢查點和錯誤復原邏輯來協調長時間執行的程序。這些服務共同協助您建立可組合、彈性和語意連線的多代理程式系統，以滿足企業需求。

多代理程式系統的商業價值

雖然許多組織使用單一代理程式解決方案開始 AI 旅程，但可透過可擴展的多代理程式系統釋放代理程式 AI 的完整潛力。這些系統是解決複雜、分散式問題，以及建立穩固、彈性 AI 生態系統以因應業務需求的關鍵。

多代理程式系統的核心業務優勢包括下列項目：

- 可擴展性 – 任務和工作負載可以分散到專門的代理程式，以提高容量和效能。
- 彈性 – 可以新增、取代或修改代理程式，並將中斷降至最低，從而在動態環境中實現敏捷性。

- 彈性 – 由於備援角色和智慧型容錯移轉，即使個別客服人員失敗，系統穩定性也會保持。
- 專業化 – 專門建置的代理程式會以更高的效率和精確度執行任務。
- 成本效率 – 可重複使用的代理程式元件可加速開發並降低新功能部署的成本。

雖然多代理程式系統需要更預先的規劃，但它們可提供長期的靈活性、速度和創新容量。投資彈性代理程式協同合作架構的企業，可以快速部署新的 AI 功能、適應不斷變化的需求，並在越來越多的代理程式驅動的競爭環境中領導。

重點區域 3：多租戶和控制的架構師

待完成的任務：「協助我跨多個客戶擴展客服人員使用量，而不會失去控制權、責任或可見性。」

早期原型可以獨立提供價值，但大多數企業需要同時支援多個客戶、部門或工作流程。這表示每個代理程式都必須在明確定義的政策、資料和身分界限內操作。如果沒有多租用戶，操作會變得脆弱且成本高昂，而控管會成為修補程式。

策略

遵循軟體即服務 (SaaS) 架構的原則。例如，租戶隔離、政策強制執行和資源控制的設計。架構師代理程式和具有租戶感知記憶體、組態和身分的協同運作平台。若要強制執行界限，請使用標記、角色型存取控制 (RBAC)，以及身分和存取管理範圍。

採用統一的可觀測性層，其中代理程式遙測會依租戶內容彙總。實作集中式政策引擎和組態型功能切換，以強制執行動態行為規則。

建置代理程式部署即服務。讓內部團隊或客戶能夠以可擴展、受管 APIs的形式使用客服人員功能。為這些模式 AWS 提供堅實的基礎。您可以使用 Amazon Cognito 管理使用者和租戶身分，AWS Organizations以及跨帳戶控管的服務控制政策 (SCPs)，以及 AWS Resource Access Manager (AWS RAM) 安全地共用功能。此外， AWS AppConfig可以動態管理租用戶或環境的客服人員行為。這些服務有助於強制執行邊界和政策，同時支援共用基礎設施。

從靜態部署到動態佈建的這種轉換，將代理式 AI 轉換為整個企業的平台。

多租戶代理程式平台的商業價值

多租戶不只是架構上的便利性，而是業務加速器。隨著智慧型客服人員在各部門和團隊之間擴散，組織必須支援成長，而不會複製基礎設施或分割控管。

多租戶系統的主要業務優勢包括：

- 可擴展性 – 多租戶代理程式平台可讓內部團隊、業務單位或用戶端更快速地加入 AI 功能，而不需要自訂環境。
- 成本效率 – 共用基礎設施可將備援部署降至最低、合併營運成本，並簡化跨環境的維護。
- 控管和風險降低 – 集中式政策控制、身分模型和可觀測性可協助客服人員在所有租用戶中更安全且合規地運作。
- 服務可重複使用性 – 為了促進重複使用並減少重複，租戶感知代理程式可以作為內部服務提供，例如用於擴充、合規或摘要。

多租戶系統的範例使用案例包括下列項目：

- 跨子公司部署的合規代理程式會透過租戶特定的組態，將其邏輯調整為符合當地法規。這樣就不需要為每個區域建立單獨的代理程式。
- 內部工作流程自動化代理程式為具有不同資料界限和許可的多個部門提供服務。它會在加速任務履行的同時維持隔離。

透過將代理程式設計為multi-tenant-aware服務，組織可以避免孤立 AI 計畫的額外負荷。相反地，它們會培養統一的智慧平台。此架構可實現可擴展的推展、操作一致性和更好的投資報酬率。它也可讓您更輕鬆地將 AI 採用擴展到整個企業。

重點領域 4：透過身分、護欄和可觀測性建立信任

要完成的任務：「給予我信心，相信客服人員將安全且可預測地採取行動，尤其是在沒有人觀看時。」

自治代理程式挑戰傳統控制模型。如果未正確管理，他們獨立推理和行動的能力會帶來風險。如果沒有明確的擁有權、可稽核性或政策限制，它們可能會偏離其預期的行為。建立組織信任不僅需要技術可靠性。它需要可解釋性、責任和一致性。

策略

建置身分優先控制系統做為受信任自主權的骨幹。每個代理程式都必須使用可驗證的身分、範圍許可和可追蹤的執行歷史記錄來操作。代理程式應內嵌在零信任架構中，其中包括租用戶繫結、內容存取繼承，以及透過護欄和政策引擎強制執行執行期。這可讓您根據組織規則和風險狀態來稽核、反轉或限制客服人員動作。

透過智慧型護欄在執行時間內嵌信任強制執行。這包括根據行為模式或工作負載條件的速率控制和限流、與自動擴展一起強制執行的資源界限，以及評估風險的決策評分。建置觸發條件，以在超過閾值時參與human-in-the-loop工作流程。

每個代理程式也必須透明且可解釋。透過記錄、追蹤和推理摘要嵌入結構化遙測，以公開決策邏輯。支援決策追蹤和影響追蹤。這可協助您將客服人員動作連線至關鍵指標或結果。實作偏離偵測機制，以監控與預期行為或政策的偏差。

引進可持續觀察客服人員行為和系統模式的反射客服人員。它們應該即時標記異常或不一致。這些代理程式有助於控管可啟動功能重新驗證、調整或解除委任的意見回饋迴圈。

建立管控委員會，以檢閱客服人員政策、核准功能變更，以及監督事件回應通訊協定。必須獲得、衡量和持續強化信任。

AWS 為實作此信任架構提供了堅實的基礎：

- AWS Identity and Access Management (IAM) 強制執行以角色為基礎的執行和許可界限
- Amazon CloudWatch 和 AWS X-Ray支援完整可見性和可追蹤性。
- Amazon GuardDuty 並AWS Config偵測安全異常或政策偏離。

這些服務可一起大規模實現身分強制執行、執行期安全和以信任為基礎的控管。它們有助於使自動化系統既強大又可靠。

受信任自主權的商業價值

隨著客服人員變得更自主，信任會成為企業採用、控管和營運效能的關鍵驅動因素。建立身分、可觀測性和護欄的基礎，有助於組織將代理式 AI 擴展到敏感網域，而不會犧牲管控或控制。

主要業務驅動因素包括下列項目：

- 控管保證 – 強大的身分模型、稽核線索和許可界限可降低合規風險並支援法規一致性。
- 營運持續性 – 執行期護欄和異常偵測有助於防止意外行為，並支援從邊緣案例故障中自我復原。
- 利益相關者信心 – 決策可解釋性和遙測與內部利益相關者、風險經理和外部稽核人員建立信任。
- 事件彈性 – 內嵌可觀測性可在問題發生時加速根本原因分析和回應時間。

範例使用案例包括：

- 在金融服務中，詐騙偵測代理程式必須公開其推理、記錄每個具有可追蹤身分的動作，並在嚴格範圍的 IAM 角色下操作。
- 在醫療保健中，自動分類代理程式必須強制執行執行期安全檢查、在達到閾值時上報至人工審核，並提供完整的日誌以進行臨床監督。

透過將信任機制嵌入到代理程式生命週期中，組織可以允許其系統以負責任的方式自動操作。此基礎可降低風險，並使客服人員能夠以透明度和完整性代表業務行事。

最後，信任自主性可為使用者和領導層提供可跨核心操作擴展智慧型客服人員的信心，以加速採用。

重點區域 5：管理生命週期

要完成的任務：「確保我的團隊可以隨著時間改善客服人員，而不會發生混沌或英雄。」

與僅由程式碼塑造的傳統應用程式不同，代理程式行為也會由提示、記憶體、工具和訓練內容塑造。這些因素會隨著時間而漂移。漂移會降低可靠性、增加成本，並使除錯幾乎不可能。如果沒有生命週期控制，客服人員會停止交付價值並開始累積風險。

策略

建立客服人員的 DevOps (AgentOps) 做為實務。整合專為客服人員量身打造的 CI/CD 管道。使用這些管道來測試提示輸出、驗證工具整合，以及描述成本效能行為。維護提示、政策和模型互動的版本歷史記錄。

使用可觀測性資料的回饋迴圈來啟動重新訓練、提示調校或代理程式淘汰。整合全系統反射機制，例如改善登錄，以將學習機制化。

建置效能遙測儀表板，顯示決策準確性、延遲、成本和可靠性。為了使用 AWS 基礎設施簡化和加速生命週期管理， 團隊可以使用客服人員工具組。其中一個範例是 Strands Agents SDK，它為提示版本控制、工具註冊和 CI/CD 整合提供結構化工具 AWS 服務，例如 AWS CodePipeline、 AWS Cloud Development Kit (AWS CDK)和 AWS Lambda。此外，使用 Amazon S3 和 Amazon Elastic File System (Amazon EFS) 來儲存代理程式成品和訓練資料。使用 AWS Step Functions 來自動化複雜的重新訓練或驗證工作流程。當代理器需要自訂模型調校或微調 LLM 協同運作之外的工作流程時，您可以使用 Amazon SageMaker AI。生命週期紀律會將代理程式從實驗轉換為持久且不斷發展的資產。

隨著時間的推移，此生命週期系統形成了創新的骨幹。它可協助您靈活地重新編譯、重新訓練和重新部署功能。這會將代理程式層轉換為活系統，能夠因應意見回饋和機會而演進。

生命週期管理的商業價值

有效的生命週期管理是客服人員效能和成本效益的關鍵驅動因素。它可確保智慧代理器在演進時繼續提供準確、可靠且符合價值的結果。根據預設，代理程式不會保持價值。它們必須與不斷變化的業務需求、工作流程和資料環境同步發展。有紀律的 AgentOps 團隊可協助客服人員隨著時間保持準確、高效，並與企業目標保持一致。

主要業務驅動因素包括下列項目：

- 效能一致性 – 持續測試、提示驗證和重新訓練可協助客服人員在不斷變化的條件和資料集之間維持決策品質。
- 成本最佳化 – 遙測驅動的分析可識別效率不佳的工具、高字符提示或不必要的執行。然後，您可以調校 以降低營運成本。
- 更快速的反覆運算 – 使用 CI/CD 的生命週期自動化可加速開發週期，協助團隊安心實驗、部署和改善客服人員。
- 降低風險 – 提示版本控制、轉返支援和結構化評估機制有助於防止迴歸，並支援安全可靠的變更管理。

範例使用案例包括下列項目：

- 監控客戶支援代理程式的延遲、模型成本和使用者意見回饋。可觀測性顯示成本峰值，提示重新調校其內嵌提示和備用模型邏輯。
- 合約摘要代理程式會根據法律團隊的意見回饋進行更新。版本控制的提示會在生產發行前於沙盒環境中進行測試，以支援安全與品質。

透過結構化生命週期管理，組織可以超越被動式維護，以主動、持續改進。客服人員會成為適應性數位資產，這些資產會根據業務目標進行衡量、改進和重新驗證。此實務將代理程式生態系統轉換為高效能、成本感知和彈性系統，提供持久的價值，同時跟上變化的步伐。

重點區域 6：使客服人員模型與業務模型保持一致

要完成的任務：「顯示影響，以便我能夠證明持續投資的合理性。」

如果客服人員與業務成果無關，即使是技術能力強大的客服人員也會成為負債。代理程式必須提供效率、獲利或策略差異。不過，大多數企業都難以定義客服人員如何符合定價、封裝或使用模式。如果沒有明確符合商業價值，很難證明擴展甚至維持投資的合理性。

策略

採用產品管理實務。將代理程式視為具有可衡量 ROI 的可獲利服務。根據決策、工作階段或結果定義定價策略。然後，將客服人員功能封裝到與客戶客群或內部業務單位一致的分層產品中。

為了提升永續性，組織必須透過客服人員部署來擷取直接價值和成長倍數。請考慮使用下列 ROI 指標來測量立即值：

- 每個決策的成本 – 針對人類對等項目基準客服人員處理成本。
- 時間壓縮 – 量化加速週期的值，例如更快的銷售或核准。
- 減少錯誤 – 測量提高準確性、一致性和合規性的節省。

除了這些立即收益之外，客服人員還可以釋放下列長期成長機會：

- 功能堆疊 – 結合代理程式服務來建立特定網域的垂直解決方案。
- 網路效果 – 透過協調複合公用程式的多代理生態系統增加價值。
- 市場延伸 – 透過外部消耗、啟用代理程式的服務產生新的收入串流。

從業務指標 （例如節省成本、轉換提升或time-to-resolution) 建立意見回饋迴圈，以推動客服人員的持續發展。分析用量遙測和使用者滿意度分數，以精簡您的值一致性和藍圖優先順序。透過將客服人員功能直接連結至商業模型，組織可以自行定位以擷取永續、可複合的價值，而不只是技術成果。

以下提供強大的追蹤和獲利架構來 AWS 服務 支援此一致性：

- AWS Cost Explorer 和 Amazon CloudWatch 可讓您深入了解每個代理程式的成本和營運效率。
- Amazon API Gateway 可啟用代理程式端點的計量存取、速率限制和分層定價。
- AWS Marketplace 提供將客服人員和客服人員解決方案發佈為商業產品的管道。

這些服務可協助您將代理程式功能轉換為可擴展、價值驅動的數位產品，以符合企業成長和獲利策略。

不斷發展代理式 AI 的軟體交付

現代軟體交付已透過簡單假設來塑造，您可以控制您運送的系統。您可以定義需求、撰寫邏輯、針對預期結果進行測試，以及部署可預測的服務。即使敏捷和 DevOps 方法仍然依賴於每個衝刺提供決定性、可驗證且主要在人為監督範圍內的原則。

代理式 AI 會提升該基礎。代理程式系統解譯、推理和調整，而不是遵循指令碼。他們的行為取決於您撰寫的程式碼、他們操作的內容、他們獲得的輸入、他們可以存取的工具，以及他們獲指派的目標。簡言之，他們不遵循訂單；他們追求成果。

這使得交付的控制和對齊更少。您必須塑造其行為，而不是提供指示。這表示傳統的軟體開發生命週期 (SDLC) 不再適用，因為它專為邏輯型、人為控制的系統而設計。

本節包含下列主題：

- 代理式 AI 的意圖區域
- 不斷發展代理式 AI 的交付生命週期
- 為團隊做好客服人員 AI 的準備

代理式 AI 的意圖區域

我們需要一個包含自主性、不確定性和出現的模型，而不是定義、建置、測試和發行等剛性階段。反之，您可以使用意圖區域。意圖的zone定義了邊界空間，其中代理程式可以在限制範圍內以自主性操作。目標是從微管理每個任務轉移到設計環境，讓客服人員可以安全地採取行動、學習和協作。您可以指定什麼（所需結果）、原因（意圖）和護欄（限制條件、政策和信任界限）。鑑於這些界限和此資訊，客服人員會找出方法。

將環境視為空間，而不是組裝線。您可以控制誰可以輸入、他們可以做什麼，以及他們可以前往何處。但一旦進入，他們就可以視需要自由導覽。這就是代理系統在沒有混沌的情況下擴展的方式。

這不只是哲學上的轉移，而是實際的轉移。無法透過單位測試完整測試以代理程式為基礎的系統的非確定性輸出。它不能像靜態二進位檔一樣進行版本控制。代理程式會隨著時間變更、適應新資料，並以無法預測的方式與其他系統互動。嘗試使用傳統模型交付它們會導致脆弱、無法擴展的架構。最壞的情況是，這會導致對無法實際控管的系統產生錯誤的信心。

當團隊接受以意圖為基礎的交付時，他們獲得兩個優勢：

- 控制它最重要的位置 – 它們定義界限而不是輸出。

- 透過委派的可擴展性 – 可讓客服人員處理人類無法硬式編碼的複雜性。

這就是您將隔離原型遷移到實際的生產級代理系統的方式，這些系統可以重複且可靠地交付價值。

不斷發展代理式 AI 的交付生命週期

若要支援智慧型自適應行為，SDLC 必須從決定性控制重新建構為自適應意圖。以下是發展客服人員 AI 傳統 SDLC 所需的變更：

- 規劃成為意圖設計。團隊定義目標、限制條件和預期的客服人員行為。政策和成功條件是以對齊為框架，而不是邏輯。
- 架構會變成堆疊。團隊專注於定義角色、界面、護欄、備用機制和可觀測性，而不是編寫每個決策路徑的指令碼。
- 測試會變成行為評估。團隊不會宣告特定輸出，而是驗證客服人員是否保持在可接受的範圍內，並在不同的輸入下實現意圖。
- 部署會變成持續協同運作。代理程式系統會部署執行時間控制、即時監控和意見回饋管道，以啟用即時調校。
- 反覆運算會成為意見回饋和適應。團隊會觀察客服人員如何進化、在何處成功或何時偏離，而不是傳統的程式碼變更修補程式週期。如有必要，團隊會介入更新的限制、重新訓練，以及新增或修改控制機制。

專注於反覆運算、實驗和快速意見回饋的現有實務位於一半。轉向代理系統並非拒絕敏捷原則。事實上，這是它們的自然演變。敏捷思維強調了剛性計劃的適應性、意見回饋和工作解決方案。這完全符合代理系統的性質，可即時學習、調整和回應內容。如果您已經執行短期週期、快速驗證假設，以及透過持續交付來管理不確定性，您已準備好領導此轉換。

但有一些主要差異。傳統的敏捷方法假設交付的物件是確定性的。其假設在建置之後，物件的行為將一致且可預測，並具有相同輸入的可重複結果。此可重複性可協助您放心地偵錯、測試和重複。代理程式系統破壞了該模型。它們很概率、內容敏感，並且能夠獨立發展。這表示某些敏捷實務變得較不實用，例如根據故事完成的速度追蹤、嚴格的接受條件或確定性衝刺規劃。

傳統 SDLC 的下列層面適用於代理式 AI：

- 反覆開發和交付
- 客戶意見回饋做為主要訊號
- 跨功能協同合作

- 持續整合和部署

代理式 AI 必須發展下列傳統 SDLC 層面：

- 按照意圖重新定義完成。 專注於客服人員的行為是否符合其在定義限制內的預期目標。
- 從接受條件轉移到行為防護機制。
- 展開已完成的定義，以納入執行階段整備，其中包括可觀測性、可解釋性和支援持續學習和信任的意見回饋機制。
- 排定前期規劃的即時意見回饋迴圈和行為追蹤優先順序

好消息是，您不需要擲出 SDLC 手冊。您只需要將其從管理程式碼發展為塑造行為。在代理程式系統中，成功不僅在於軟體是否執行，還在於其行為。

為團隊做好客服人員 AI 的準備

軟體工程不會消失。它正在演變。任務會從撰寫函數轉移到為智慧行為塑造架構和控制機制。在代理式 AI 的世界中，建置不再是困難的部分，管理出現就是。對於大多數工程團隊而言，演變就像是思維轉移，而不是技術飛躍。而不是詢問「系統會做什麼？」 問題變成「我們賦予它追求的能力，以及如何知道它是否繼續？」

對於工程團隊而言，客服人員 AI 的演變需要下列變更：

- 文化轉移 – 團隊必須適應他們無法完全控制的系統中的不確定性和自主性。
- 新角色 – 意圖設計人員、行為測試人員和可觀測性工程師成為交付的核心。
- 共享語言 – 團隊需要明確、共享對目標、護欄和成功訊號的理解，就像他們曾經需要規格和測試案例一樣。

隨著生成式 AI 的成熟，我們將看到更多代理系統與客戶、產品和操作互動。成功的組織不會是具有最佳模型的組織。它將是能夠將客服人員整合到真實工作流程的那些可以放心、控制和速度的工作流程。這表示交付模型和工程團隊必須一起發展。意圖區域可讓您抽象化。它們可協助您操作自主權，而不會放棄責任。他們也提供跨團隊的共用架構，以協助控管無法硬式編碼的系統。

如需準備客服人員 AI 團隊的詳細資訊，請參閱本指南的大規模準備客服人員 AI 業務一節。

為大規模代理式 AI 業務做好準備

正如本指南中所述的重點領域收斂，代理程式 AI 會從隔離的函數轉移到統一的智慧層，而這些層可以理解為功能平台。此平台不只是執行任務。它跨網域演進、調整和協調。代理程式成為模組化、可重複使用且可探索的服務，可加速創新、降低認知負載，並在整個企業中推動可衡量的結果。此平台檢視會設定階段，以在整個操作模型中嵌入可擴展的智慧。

操作化代理程式 AI 需要的不只是部署智慧型代理程式。它要求在業務如何組織團隊、設計流程和管理技術方面進行基本轉型。與轉移至雲端或 DevOps 重新定義的操作模型一樣，代理式 AI 引入了決策自動化、持續學習和自主協調的新紀元。成功取決於使系統、人員和程序與此新的操作理念保持一致。

本節包含下列主題：

- 協調團隊和擁有權模型
- 管理變更和組織整備
- 架構互通性和協同合作
- 將管控建置到代理程式結構
- 採用決策優先的操作思維
- 使用目的和意圖擴展

協調團隊和擁有權模型

成熟度的第一步是跨功能對齊。企業必須建立 AgentOps 團隊，其中包含 AI/ML 從業人員和網域專家，例如分散式系統架構師、軟體工程師、產品擁有者、合規主管和平台架構師。這些團隊共同擁有代理程式的整個生命週期，從設計和部署到重新訓練和監控。

代理程式佈建和發行應遵循雲端原生實務，例如使用 AWS Cloud Development Kit (AWS CDK)和 AWS CodePipeline 做為基礎設施的程式碼和自動化部署。此結構促進共同的責任，並加速反覆運算。就像 DevOps 統一開發和操作一樣，AgentOps 會將智慧與控管和執行連線。

為了提高效率，這些團隊也需要共用語言。業務利益相關者必須了解什麼是客服人員、他們的操作方式，以及他們推動的結果。訓練和內部啟用至關重要。透過解密客服人員並將這個心理模型嵌入日常對話中，組織可以釋放更廣泛的參與和更一致的創新。

為了使用 來加速代理程式的開發和整合 AWS 服務，團隊可以採用 Strands Agents SDK 等架構，該 SDK 提供 CLI 型工具，用於堆疊、設定和封裝代理程式。Strands Agents 旨在順暢地與 AWS 基礎

設施搭配使用，例如 Amazon Bedrock、AWS Lambda、Amazon EventBridge AWS CDK、 和 AWS CodePipeline。它可實現快速原型設計和部署，同時維持生產級標準。

但是，只有結構和工具是不夠的。擴展代理式 AI 需要深思熟慮的文化、教育和領導力準備，以確保採用能根植於整個組織。

管理變更和組織整備

成功擴展代理程式 AI 需要的不只是部署基礎設施或智慧型代理程式。它需要結構化的組織變革方法。這包括文化準備度、技能開發、指標驅動的意見回饋迴圈和執行一致性，以確保採用是有意且永續的。

促進文化發展

- 將客服人員定位為團隊成員，而非替代人員，以減少阻力並建立信任。
- 透明地傳達客服人員的功能和限制，以設定實際的期望。
- 為客服人員應將決策呈報給更高權限或將部分程序委派給人工合作者，建立明確的交接通訊協定。

建立技能開發架構

- 為工程師、產品經理、網域主管和合規主管提供量身打造的角色型訓練。
- 建立卓越中心，以分享最佳實務、工具模式和可重複使用的資產。
- 透過指導計畫將 AI 專家與領域專家配對，以彌補知識差距。

定義指標和意見回饋迴圈

- 將技術和業務 KPIs 錨定為策略價值，以評估影響。價值的範例包括決策延遲、解析度準確性和成本節省。
- 系統化且持續地擷取使用者對表面摩擦點和採用挑戰的意見回饋。
- 執行定期回顧，以評估客服人員效能、用量趨勢和改善機會。

從頂端調整領導

- 將客服人員計畫連結至策略成果和投資報酬率，以取得高階主管的贊助。
- 組成包含技術和業務領導的跨職能控管委員會。
- 量身打造溝通策略，以提高所有組織層級的清晰度和參與度。

這種變更管理的系統性方法可確保技術實作符合組織成熟度。它為信任、採用和長期商業價值奠定了基礎。

架構互通性和協同合作

隔離的代理程式部署可提供本機勝利。但是，當客服人員可以動態探索、叫用和協同合作時，企業價值就會出現。這表示定義客服人員註冊、身分驗證和功能交換的標準。在架構上，這反映了從整體到微服務的轉移，這是可組合、可重複使用和鬆耦合的單位，可一起解決複雜的問題。

A2A 和 MCP 等新興通訊協定是基本的。啟用客服人員、工具和記憶體系統的語意互通性。A2A 支援對等層級互動，可讓客服人員交涉任務擁有權、共用內容和協調工作流程。MCP 透過提供共用結構描述來補充這一點，以便在代理程式及其環境之間交換內容資料。它會標準化如何叫用函數、存取 APIs，以及維護狀態。這些通訊協定共同提升了客服人員生態系統的可擴展性、一致性和長期可維護性。

控管仍然至關重要。控制層，例如任意代理程式，可啟用政策感知委派，而不會引入集中式瓶頸。這些代理程式充當信任代理程式。它們強制執行界限，同時讓其他客服人員自行組織。客服人員協作可協助組織同時靈活和信任地擴展其客服人員 AI 生態系統。

將管控建置到代理程式結構

隨著自主性提高，風險也會提高。管控必須從第一天開始內嵌到代理程式架構中。這包括定義政策界限，以限制允許客服人員執行的動作、強制執行身分模型來判斷他們代表的對象，以及實作可解釋性和可追蹤性。可觀測性系統必須使用 Amazon CloudWatch 和 等服務擷取代理程式行為的遙測AWS X-Ray，這些服務提供跨代理程式工作流程的集中式記錄和分散式追蹤。反射式代理程式可以根據這些遙測饋送持續稽核和評估效能。

控管也必須隨著代理程式生態系統的成熟而發展。隨著客服人員變得更有能力且更自主，監督機制必須變得更適應。政策更新、功能閘道和執行時間行為限制條件需要是動態且可大規模強制執行的。信任不是鎖定功能。它透過架構、行為和程序持續強化。AWS Identity and Access Management (IAM) 和 在強制執行安全身分、執行時間許可界限和環境特定行為時AWS AppConfig扮演關鍵角色，跨客服人員切換。

採用決策優先的操作思維

傳統自動化著重於程序效率，這會更快、更可靠地執行預先定義的指令碼或工作流程。反之，代理式 AI 引進了決策優先自動化。客服人員會即時評估內容、權重選項和調整行為。這種從執行優先轉移到

決策優先的思維需要重新考慮成功指標和結果。代理式 AI 的成功不是只透過任務完成來衡量成功，而是透過決策與意圖、政策和不斷變化的條件的一致性來衡量。

組織必須評估決策品質、time-to-action對變革的回應能力，而不是僅測量任務完成或週期時間。KPIs應包含下列指標：

- 決策品質 – 客服人員對特定使用者或案例的個人化回應有多好？ 它是否做出符合業務目標和使用者內容的細微決策？
- Time-to-action – 客服人員評估情況和回應的速度有多快且多聰明？ 延遲是否低到足以感覺適應性和類似人類？
- 認知卸載 – 代理程式能夠代表人類處理多少手動分析、分類或例行決策？ 它是否減少工作量或只是轉移工作量？

接受以決策為優先思維的企業可以變得更有彈性、更具適應性，並且能夠以新的複雜性操作。

使用目的和意圖擴展

成功擴展代理式 AI 與試驗更多工具無關。這是關於建置耐久的企業智慧層。這需要投資平台基礎設施、營運文化、控管架構和策略一致性。企業必須採用刻意的方法。他們必須將代理程式視為不是實驗，而是其數位操作模型的核心元件。

與 AWS Well-Architected Framework 保持一致有助於您的系統符合可靠性、安全性、效能效率和成本最佳化的企業標準。Strands Agents SDK 等工具可以透過提供結構化提示、工具註冊和 CI/CD 整備度來加速此旅程。這有助於團隊使用熟悉的 AWS 工作流程，從實驗轉移到可擴展的交付。

代理式 AI 不是工具，而是將智慧嵌入 操作的轉變。相應地準備的組織可以自動化更多、更智慧地操作、更快地適應，並在日益複雜的世界中建立持久的優勢。

客服人員 AI 操作的結論

代理式 AI 不只是技術轉移。它代表企業新作業系統的出現。接受此轉型的組織會超越狹窄的自動化使用案例，並將智慧建置為其營運的基礎。此轉移旨在重新設計如何做出決策、系統如何適應，以及如何大規模實現結果。

在日益增加的複雜性、即時需求和資訊超載的時代，指令碼化自動化的傳統模型已達到其限制。成功現在取決於直接將智慧嵌入工作流程的能力，讓系統感知、推理、採取行動和發展。代理式 AI 可以將自主性與目的、決策與控管保持一致，以及適應責任。

此轉換需要從執行優先到決策優先的思維。代理系統不僅僅遵循指示。它們解釋目標、權衡權衡，並在定義的限制範圍內追求結果。在這種情況下，成功不僅取決於任務完成。它還取決於即時所做決策的品質、敏捷性和可解釋性。組織必須重新思考指標、獎勵和系統設計，以支援在不確定的情況下以智慧方式運作的客服人員。

操作化代理式 AI 不是plug-and-play升級。這是架構和文化轉型。它需要跨生命週期管理、信任強制執行、互通性以及符合商業模型的有紀律實務。它也需要不斷發展的交付模型，例如塑造意圖區域、嵌入執行期護欄，以及持續使客服人員行為與策略成果保持一致。團隊必須採用共用語言、共用擁有權，以及共同負責客服人員效能和安全性。

企業整備度可以判斷誰在這個新環境中茁壯成長。組織必須投資於內部啟用、AgentOps 功能，以及擴展和建立長期價值的控管架構。成功的人可以建立更智慧的系統，也可以建立更具適應性、彈性和洞察力的企業。

本指南會鋪設基礎。它將策略連接到執行，並準備組織建立可擴展的智慧型代理程式平台。關於 上的代理式 AI 的更廣泛內容系列 AWS 提供了補充指導。若要檢視此系列中的其他指南，請參閱 AWS 規範指引網站上的客服人員 AI。此內容系列提供路線圖，以紀律和意圖操作自主性。

若要開始使用，請識別高影響的決策空間，客服人員可以在速度、準確性或回應能力方面提供可衡量的改善。然後部署具有檢測、管控和回饋迴圈的聚焦試點代理程式。使用此值來驗證值假設、產生內部動量，並在方法中建立信任。透過學習的動量複合。

代理式 AI 不是目的地；它是隨您的業務一起發展的功能層。它代表長期轉向智慧做為基礎設施。在這個空間中領導的組織可以自動化更多、回應更快、適應更快，並建置能夠以企業規模導覽複雜性的操作模型。

操作代理式 AI 的資源

AWS 服務

下列 AWS 服務 和 功能可協助您在 中建置和操作代理式 AI 系統 AWS 雲端：

- Amazon API Gateway 可以將代理程式功能公開為可擴展，並提供以用量為基礎的定價。
- AWS AppConfig 為租用戶或環境的客服人員提供執行期組態管理和功能切換。
- Amazon Bedrock 是客服人員可用於推理、產生和提示執行的基礎模型服務。
- AWS Cloud Development Kit (AWS CDK) 是一種基礎設施即程式碼服務，可用來部署和管理代理程式堆疊。
- AWS CloudTrail 會記錄事件歷史記錄，以便您可以追蹤客服人員活動、稽核追蹤和整合行為。
- Amazon CloudWatch 可以管理日誌、指標和警示，以監控客服人員效能和多客服人員協同合作行為。
- AWS CodePipeline 提供 CI/CD 自動化，您可以用來測試、驗證和部署代理程式程式碼。
- Amazon Cognito 是一種身分服務，可用於管理多代理系統中的使用者和租戶身分驗證。
- AWS Config 提供代理程式政策和環境組態的合規和偏離偵測。
- AWS Cost Explorer 可以追蹤客服人員層級的用量，並協助調整成本以最大化您的投資報酬率。
- Amazon DynamoDB 是一種儲存服務，可用於代理程式記憶體、改善日誌和內容狀態。
- Amazon Elastic File System (Amazon EFS) 是一種共用檔案系統，可用於跨工作流程的客服人員協作或中繼處理。
- Amazon EventBridge 是核心事件匯流排，可用來在代理程式結構中路由任務和協調通訊。
- Amazon EventBridge 管道可以簡化連線客服人員和服務的事件擷取和路由。
- Amazon GuardDuty 提供可支援安全代理程式執行的威脅偵測和異常監控。
- AWS Identity and Access Management (IAM) 可協助您定義代理程式執行和資料存取的精細許可。
- AWS Lambda 是一種無狀態運算服務，可執行代理程式邏輯和扭曲無人機。
- AWS Marketplace 是一個外部分發平台，您可以用來提供代理程式功能做為商業產品。
- AWS Organizations 是一項跨帳戶管理和政策強制執行服務，可協助您管理多租戶代理程式基礎設施。
- AWS Organizations 服務控制政策可做為在帳戶或組織單位層級控制許可的護欄。
- Amazon Quick 是一種生成式 AI 驅動的商業智慧 (BI) 平台，可協助您分析資料、建立視覺化效果、自動化工作流程，以及與整個組織中的其他人協作。

- AWS Resource Access Manager (AWS RAM) 可協助您在帳戶和客服人員服務之間共用功能。
- Amazon SageMaker AI 是一種服務，可用於基礎模型以外的模型訓練、微調和推論。
- Amazon Simple Storage Service (Amazon S3) 為提示程式庫、模型成品和代理程式產生的資料提供物件儲存。
- AWS Step Functions 是一種工作流程引擎，可協助您協調多代理程式流程和重新訓練管道。
- AWS X-Ray 提供分散式追蹤，您可以用來追蹤客服人員決策流程和服務相依性。

其他 AWS 資源

- 上的代理式 AI 基礎 AWS
- 上的客服人員 AI 模式和工作流程 AWS
- 上的客服人員 AI 架構、通訊協定和工具 AWS
- 在 上建置代理式 AI 的無伺服器架構 AWS
- 在 上建置代理式 AI 的多租戶架構 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	—	2025 年 8 月 12 日

AWS 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

數字

7 R

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

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

A

A2A Agent-to-Agent)

支援任務委派和狀態轉移的agent-to-agent協同合作的狀態通訊協定。

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。

客服人員

一種 AI 系統，可使用工具自動推理、規劃和採取行動來實現目標。

客服人員操作

在生產環境中大規模建置、測試、部署和執行 AI 代理器的操作實務。

彙總函數

在一組資料列上操作並計算群組單一傳回值的 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

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，例如在網際網路上編製資訊索引的 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，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。

公民開發人員

在沒有專業技術技能的情況下，使用無程式碼/低程式碼平台建立 AI 應用程式的商業使用者。

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 遷移策略關聯的資訊，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。

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

一種運算程序，可將人類可讀取的純文字資料轉換為加密文字。

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 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，已針對廣義和未標記資料的大量資料集進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱什麼是基礎模型。

FM 閘道

集中式中介，可控制和標準化對基礎模型的存取。也稱為 LLM 閘道。

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。

護欄 (AI)

可篩選、驗證和限制代理程式輸入和輸出的安全機制，以協助確保負責任且安全的 AI 行為。

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。

human-in-the-loop (HitL)

一種工作流程模式，其中代理程式執行會在關鍵決策點暫停進行人工審核和核准。

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

請參閱基礎設施即程式碼。

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 Rs。

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。

MCP

請參閱模型內容通訊協定。

模型內容通訊協定 (MCP)

用於代理程式對工具通訊的無狀態通訊協定。

MCP 伺服器

透過模型內容通訊協定公開一或多個工具的服務。

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

屬於組織一部分的管理帳戶 AWS 帳戶 以外的所有 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)

此 AWS 計畫提供諮詢支援、訓練和服務，以協助組織建立強大的營運基礎，以移至雲端，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，整合 OT 和資訊技術 (IT) 系統是工業 4.0 轉型的關鍵重點。

操作整合 (OI)

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

組織追蹤

由 建立的線索 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有 的所有事件 AWS Organizations。在屬於組織的每個 AWS 帳戶 中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱 CloudTrail 文件中的建立組織追蹤。

組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。在 AWS 遷移策略中，此架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱 OCM 指南。

原始存取控制 (OAC)

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

原始存取身分 (OAI)

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

ORR

請參閱操作整備審核。

OT

請參閱操作技術。

傳出 (輸出) VPC

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

P

許可界限

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

個人身分識別資訊 (PII)

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

PII

請參閱個人身分識別資訊。

手冊

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

PLC

請參閱可程式設計邏輯控制器。

PLM

請參閱產品生命週期管理。

政策

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

混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則可以更輕鬆地實作並達到更好的效能和可擴展性。

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 個 R。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

請參閱 7 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)、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 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？ 在 Secrets Manager 文件中。

設計安全性

透過整個開發程序將安全性納入考量的系統工程方法。

安全控制

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

安全強化

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

安全資訊與事件管理 (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)

服務效能層面的測量，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。

陰影 AI

在組織內受管管道之外建置或使用的未授權 AI 應用程式。

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。

tool

代理程式可以叫用以在外部系統中執行操作的函數或 API。

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，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，多次讀取。

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缺陷或漏洞。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。