在 上建置代理式 AI 的多租戶架構 AWS

AWS 方案指引

AWS 方案指引: 在 上建置代理式 AI 的多租戶架構 AWS

Table of Contents

在 上建置代理式 AI 的多租戶架構 AWS

Aaron Sempf 和 Tod Golding，Amazon Web Services

2025 年 7 月 (文件歷史記錄)

代理式 AI 代表一種破壞性模式轉移，需要組織重新思考如何建置、交付和操作其系統。代理程式模型讓團隊探索將系統分解為一或多個建立新路徑、可能性和值的代理程式的新方法。

許多客服人員討論中心圍繞用於建置和實作客服人員的工具、架構和模式。我們不僅必須採用良好的工具來建立代理程式，還必須採用新的整合通訊協定、身分驗證策略和探索機制，這些機制可以做為代理程式架構的基礎。

當客服人員工具的數量增加時，團隊也必須考慮其客服人員如何處理更傳統的架構挑戰。規模、嘈雜的鄰里、彈性、成本和營運效率是基本主題，您必須在設計、建置和部署代理程式時進行評估。無論自動代理程式和智慧代理程式可能如何運作，我們也必須確保它們達到符合業務需求的規模、效率和敏捷性經濟。

本指南的目標是探索客服人員足跡的各種維度。這包括檢閱各種代理程式部署和使用模式，並強調建立可處理架構目標的代理程式的不同策略。它還意味著透過引入多租用戶設定中通常需要的內部建構，查看如何在多租用戶環境中使用代理程式。

目標對象

本指南適用於希望建置 AI 驅動型多租戶系統的架構師、開發人員和技術領導者。

目標

本指南可協助您執行以下操作：

- 了解多租用戶代理程式部署、探索孤立和集區模型，以及租用戶內容如何影響代理程式實作
- 探索客服人員管理，包括加入、租用戶隔離，以及跨單一和多供應商環境的資源管理
- 評估多租戶客服人員的各個層面，包括資料擁有權、監控和測試

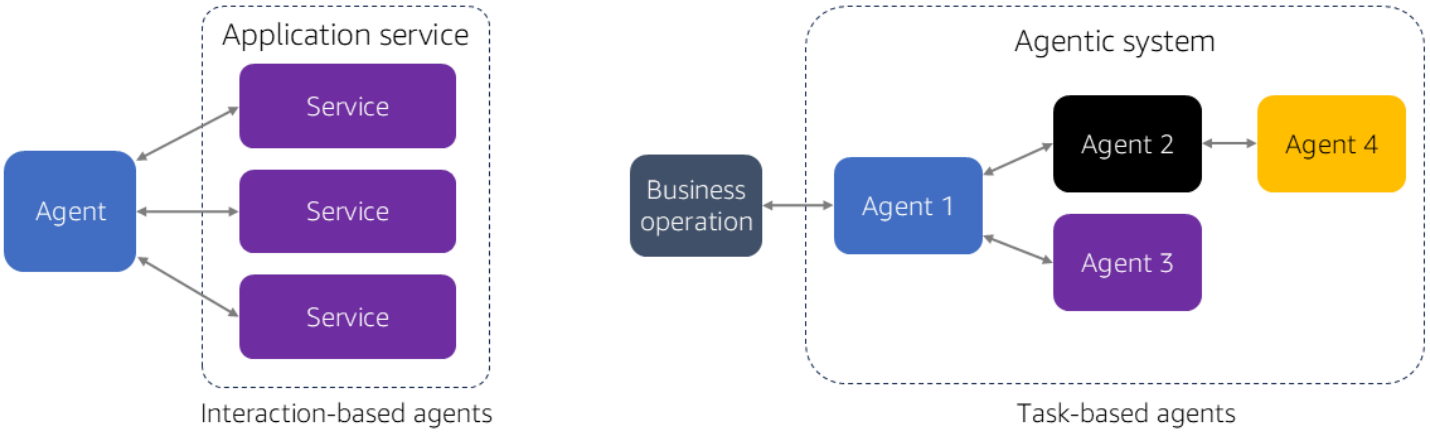
關於此內容系列

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 。

客服人員基本概念

在我們討論架構詳細資訊之前，我們應該概述客服人員扮演的不同角色，因為「客服人員」是一個可套用至許多使用案例的過載術語。讓我們從一些有助於分類的廣泛術語開始。

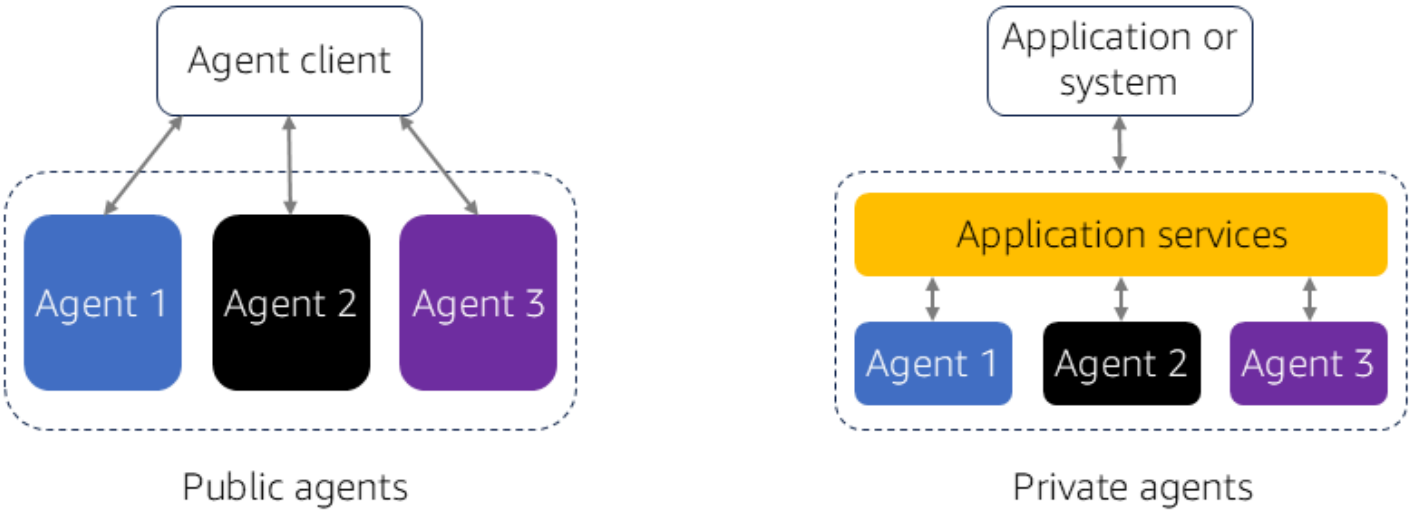
在最外層，我們需要從分類客服人員的角色和性質開始。這具有挑戰性，因為在各種情況下，客服人員可以套用到任意數量的問題。不過，在此討論中，我們專注於將代理程式引入應用程式或系統的意義。在這個模型中，我們強調客服人員如何和何地最能豐富您的系統體驗。您選擇的選項會影響代理程式的建置、整合和套用至不同網域和使用案例的方式。下圖顯示建置器使用的兩種代理程式模式。



圖表左側是互動型客服人員。在此模式中，客服人員會在現有系統中建立檢視，以協調與基礎服務的互動，以實現目標或成果。關鍵是將代理程式新增至系統，做為驅動系統特徵和功能的替代方法。例如，假設獨立軟體廠商 (ISV) 具有 UX 的會計系統，用於執行操作。互動型客服人員可簡化與這些現有功能的互動。了解如何達到鬆散定義的目標，以及提供協調已知路徑的方式。

相反地，圖表右側的任務型系統代表不同的方法。該系統中的客服人員會使用他們的知識和能力來學習完成任務並推動業務成果。您可以爭論這兩種模型都實現業務成果，但任務型模型依賴客服人員本身來確定如何實現成果。這類客服人員不太確定，而是依賴他們學習和發展的能力。相反地，互動型客服人員的設計主要是為了協調一組已知功能。這些差異會影響您如何建置、範圍和整合客服人員以支援您的業務。

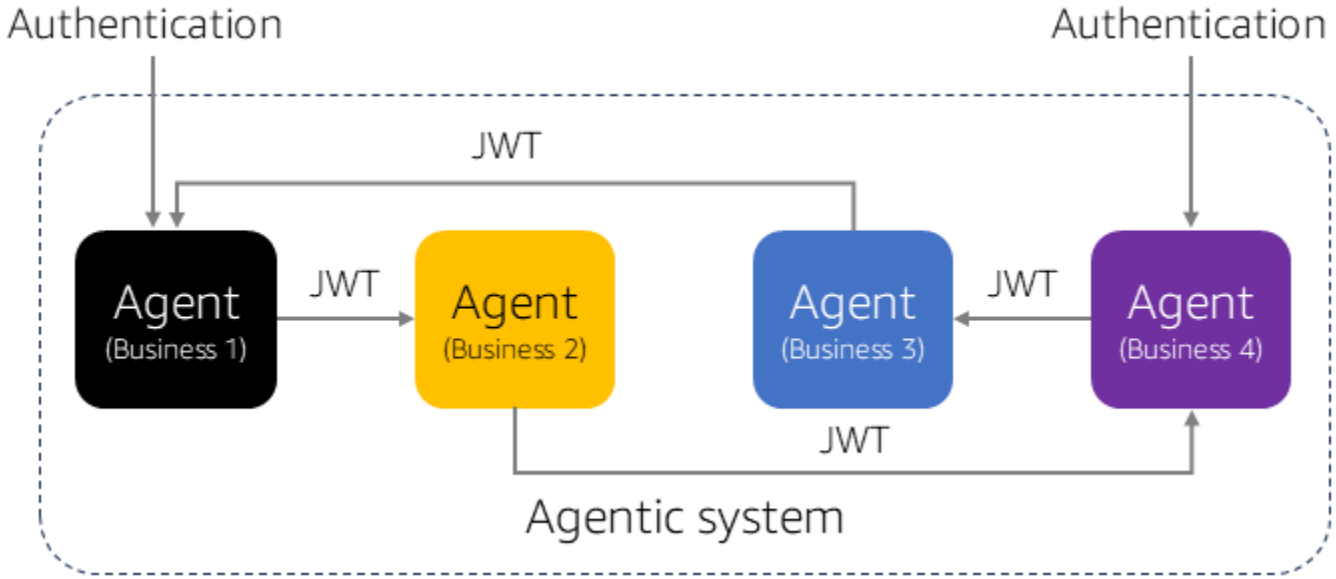
我們也需要描述部署代理程式的方式和位置的術語。代理程式位於系統足跡內的 可能會影響其建置、範圍和保護的方式。下圖概述可套用至客服人員的兩個不同模型。



圖表左側是具有三種不同代理程式的部署系統。代理程式會公開到可能是其他代理程式或應用程式的外部用戶端。在此模型中，代理程式稱為公有代理程式。

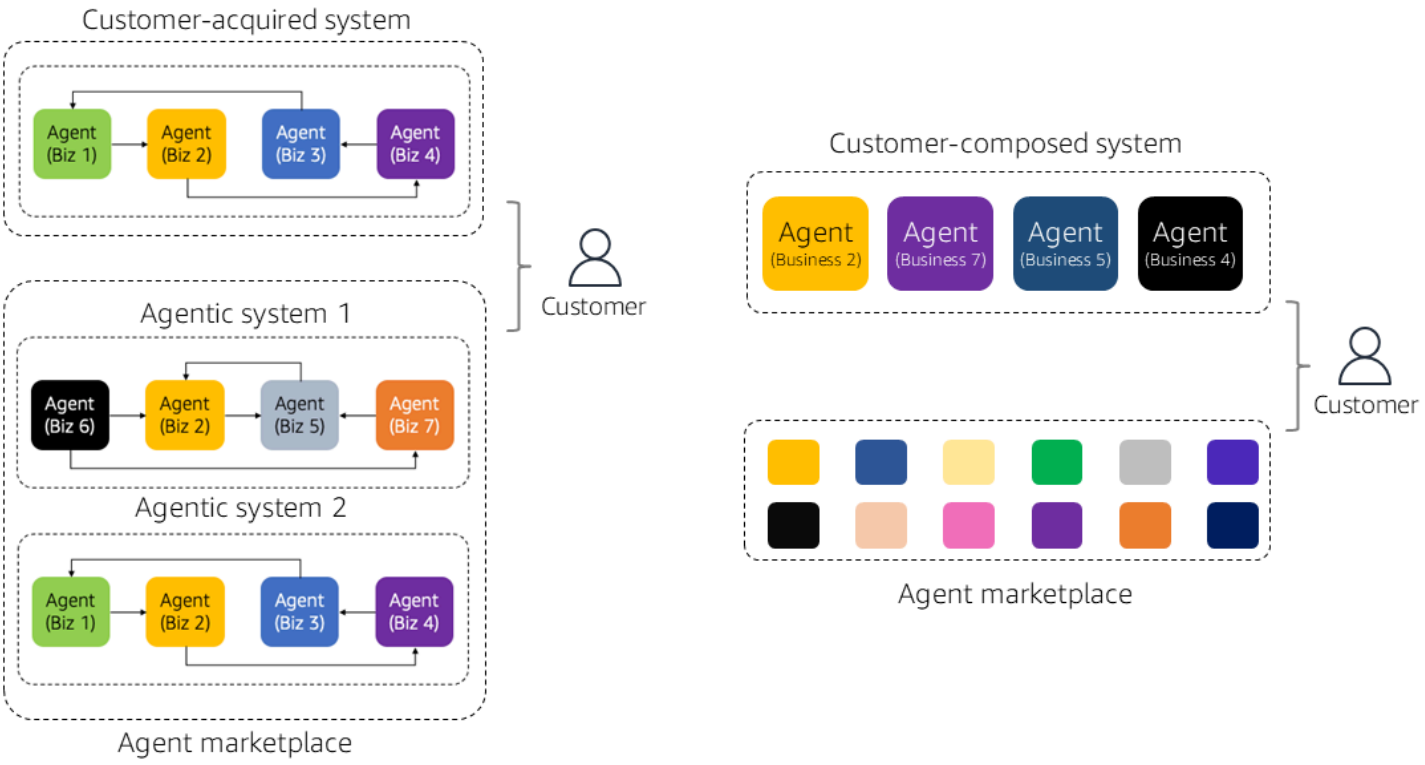
相反地，右側的圖表顯示解決方案實作中的客服人員。在這種情況下，使用者或系統會取用一系列的應用程式服務。這些使用者與應用程式互動，不知道客服人員是體驗的一部分。然後，基礎系統的服務會叫用和協調代理程式。以這種方式部署的代理程式稱為私有代理程式。

代理程式的許多值著重於公有模型，其中供應商可能會發佈其代理程式，目的是將其與其他第三方代理程式整合。然後，代理程式會成為互連服務網格或 Web 的一部分，這些服務可以共同解決許多使用案例。雖然這些代理程式可用於許多網域，business-to-business的使用案例是自然擬合的。下圖提供概念化的檢視，說明組合可解決特定問題的集合代理程式可能是什麼樣子。



圖表顯示四名業務客服人員一起合作，以達成一組目標。以這種方式編寫代理程式時，它們代表代理程式系統，而這類系統有許多口味。它們可以是一組預先封裝的協作代理程式，通常以單一單位形式使用。或者，想要挑選並選擇最符合其需求的客服人員組合的客戶可以動態組合系統。

這兩種方法都為客服人員整合提供可行的途徑。有些客服人員的建置期望他們能整合到特定系統中，藉此將價值、觸角和影響最大化。這種客服人員系統的概念也會引發有關如何取得客服人員的問題，而且可能有許多方法可以解決這個問題。下圖提供如何透過交易體驗建立這些代理程式和系統的範例。

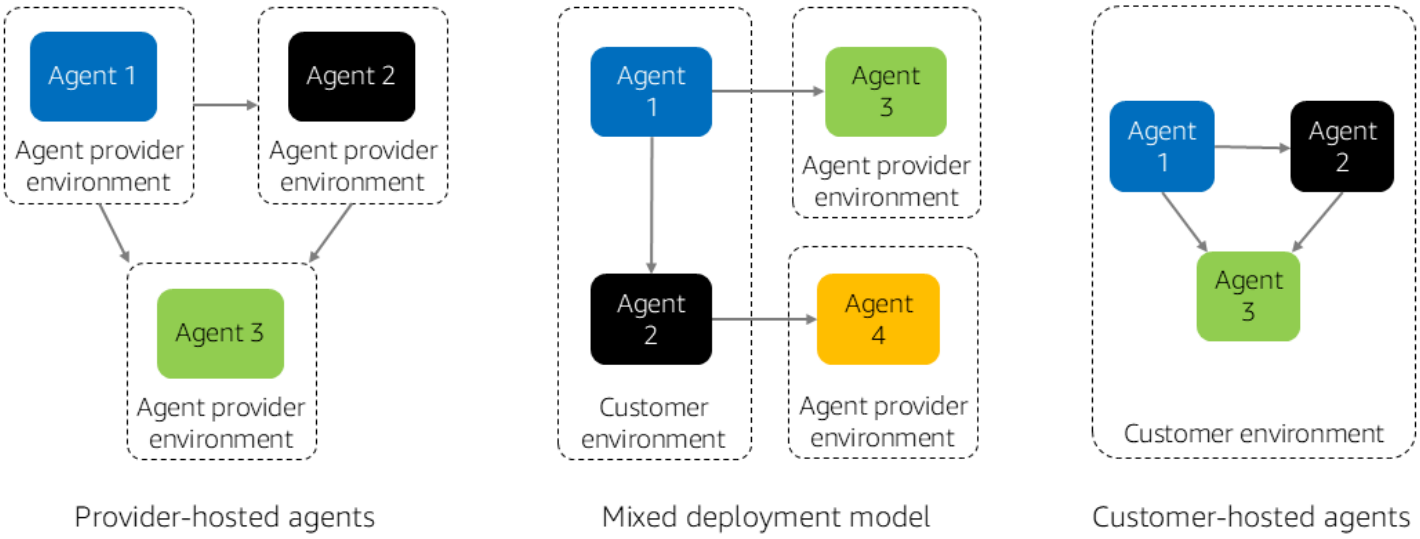


顯示兩個市場體驗範例。在左側，市場用於取得預先封裝的系統。在此案例中，市場會探索並加入系統，以因應需要整合和協調多個客服人員的廣泛目標。

右側的範例顯示一個市場，其中發現客服人員並將其組成為客服人員系統。在此案例中，客戶可以建置任何相容的整合式客服人員系統，以滿足其需求。以這種方式組合客服人員的能力取決於個別客服人員的相容性模型和整合需求。

客服人員託管考量事項

現在您已了解更廣泛的代理程式概念，讓我們來討論託管和執行這些代理程式的意義。我們必須考慮運算的執行方式和位置、擴展方式、操作方式，以及管理方式。同時，我們預期做為客服人員看到的某些模式會更廣泛地套用和採用。下圖顯示可能的排列範例。



這裡顯示三種不同的策略。在圖表左側，您會看到一個模型，在每個客服人員供應商的環境中託管、擴展和管理我們的客服人員。這些代理程式會做為服務發佈和使用，在標示為代理程式即服務 (AaaS) 模型的 中操作。在右側是一個模型，其中供應商的客服人員都託管在專用客戶環境中。

圖表中間是一個混合部署模型，結合這兩種策略，在客戶環境中託管一些客服人員，並與在供應商環境中遠端託管的一些客服人員互動。

第四個選項（未顯示）可能是將客服人員建置為由客服人員基礎設施服務擴展和管理的低程式碼或無程式碼服務。我們不會詳細說明這些內容，因為受管代理程式的架構和託管主要由擁有服務的組織決定。

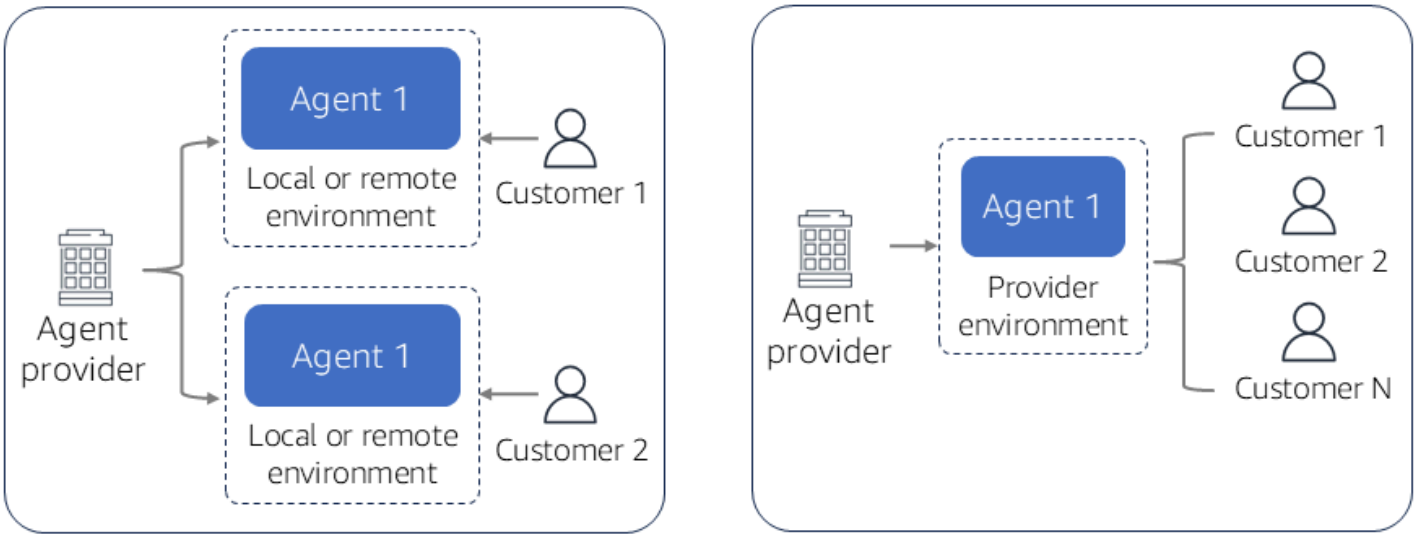
您可以想像可能影響採用其中一個模型的因素範圍。例如，合規、法規和安全限制，可能會將某人推向客戶託管的客服人員。擴展、敏捷性和效率可能會將組織推向更多 AaaS 模型。

這裡的關鍵概念是，客服人員可以透過多種方式部署和託管。您的任務是判斷如何最佳地套用客服人員。足跡、安全性和部署等因素都會大幅影響您建置和操作代理程式的方式。例如，私有和公有代理程式可能有不同的設計和發行生命週期。

代理程式符合多租戶

很容易將代理程式視為建置區塊，其中將代理程式視為一系列的自主元件，這些元件經過組合，以支援特定網域或業務問題的需求。其中更有趣的是，當我們開始考慮供應商如何封裝和使用這些代理程式時。在許多方面，代理程式會成為企業的成本和收入來源。客服人員提供者必須考慮使用其服務的不同角色、角色的耗用設定檔，以及允許客服人員供應商建立定價和分層模型以符合消費者的獲利策略。

客服人員提供者可以支援部署其客服人員的多種模型，以滿足客戶需求。下圖顯示兩個主要代理程式部署模型的概念檢視。

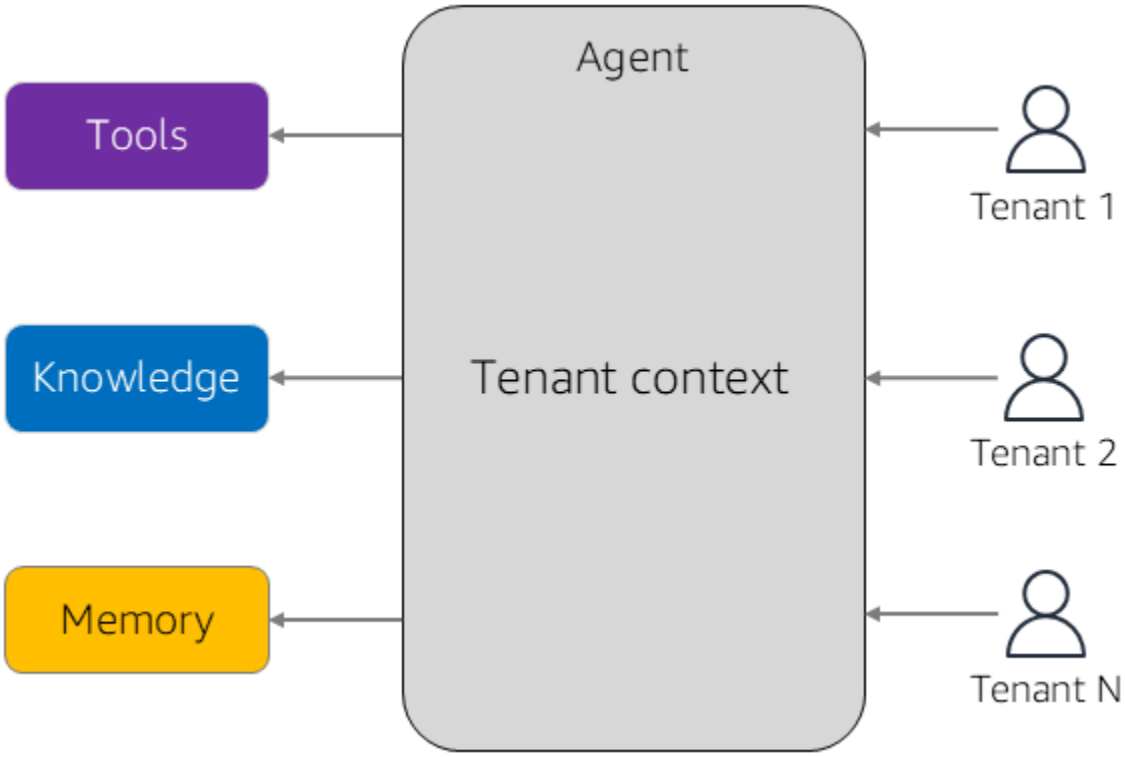


圖表的左側顯示客戶專用客服人員模型。客服人員提供者會透過為每個加入的客戶部署個別的客服人員執行個體來建置客服人員。透過此方法，客服人員的功能及其取得知識的能力將僅限於特定客戶環境的範圍。這最終代表了每位客戶的體驗，繼承了支援專用客戶環境的一些複雜性和優勢。

相反地，圖表右側的圖表具有部署在提供者環境中的單一代理程式。客服人員會根據所有客戶的集體體驗，處理來自多個客戶的請求，不斷發展和學習。新增的每個新客戶只會代表客服人員的另一個有效用戶端。代理程式的運作方式就像代理程式即服務 (AaaS) 模型，使用共用建構來支援用戶端的需求。在這兩個執行個體中，客服人員取用者可以是應用程式、系統，甚至是其他客服人員。

有兩種方式可以查看 AaaS 模型。上述模型為所有客戶提供相同的體驗。這表示代理程式的內部不會包含考慮請求用戶端內容的任何專業層級。一般而言，對於此模式，假設代理程式的範圍、目標和價值本質以一組通用套用至所有用戶端的共用資源、知識和結果為中心。

AaaS 的替代方法是用戶端內容影響客服人員的體驗和實作。下圖提供在此內容中 AaaS 代理程式使用量的概念檢視。

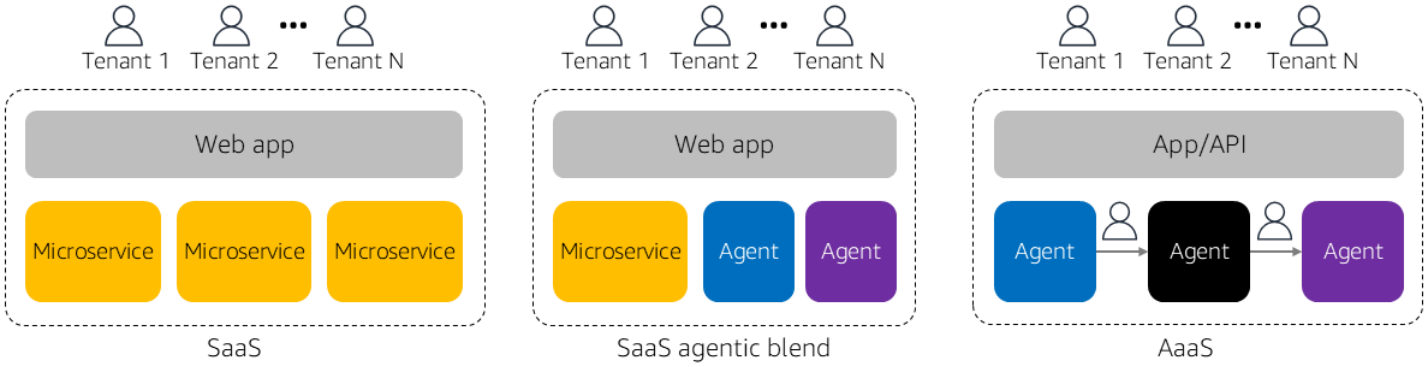


在此 AaaS 檢視中，傳入請求的來源和內容會大幅影響客服人員的足跡。做為代理程式基礎實作一部分的資源、動作和工具，可能會針對每個傳入租用戶請求而有所不同。代理程式的值與其使用租用戶內容來實現受租用戶狀態、知識和其他因素影響的動作和結果的能力相關聯。有些請求可能會產生唯一的租用戶結果，而其他請求則可能會產生更量身打造的每個租用戶結果。這為客服人員的學習能力新增了新的維度，其中可能包括更上下文，以及取得和套用增強目標成果的知識。

對於提供者，AaaS 模型提供許多優點。隨著多個客戶使用單一代理程式，供應商有機會實現規模經濟、提高營運效率、控制成本，並建立統一的管理體驗。這有可能提高客服人員業務的靈活性、創新和成長。

這些品質與推動軟體即服務 (SaaS) 模型採用的相同原則重疊。基本上，AaaS 模型是建置為多租戶服務，可繼承 SaaS 環境中許多相同的擴展、彈性、隔離、加入和操作屬性。在許多方面，AaaS 體驗會大量借用 SaaS 供應商使用的策略和實務，但區分這些條款是合理的。基於我們的目的，重點在於需要多租用戶支援的建置和操作代理程式所帶來的影響。

對於可以公平對待所有使用者且不需要管理持久性、敏感性或客戶特定資料的系統，租用的概念將對他們的客服人員影響最小。對於預期服務多個客戶的系統，同時保留資料隔離、自訂和內容意識，支援多個租戶可能是客服人員設計、策略和目標的重要元素。下圖顯示如何在代理程式環境中使用多租戶。



此圖表左側是傳統多租戶架構。它包含 Web 應用程式和一系列實作商業邏輯的微服務。多個租用戶會使用此環境的共用基礎設施，擴展以滿足租用戶人口不斷變化的工作負載。環境是透過所有租戶的單一玻璃窗格進行操作和管理。

試想這個心理模型如何映射到此圖表右側的客服人員。一個代理程式會執行由一或多個租用戶耗用的 AaaS 模型。代理程式可能來自多個供應商，其中租用戶內容在其之間流動，因為一個代理程式的單一執行個體必須處理來自多個租用戶的請求。

此圖表中間的範例是混合模型，其中客服人員是整體 SaaS 體驗的一部分。系統的某些部分是在較傳統的模型中實作，而系統的其他部分則依賴客服人員。此模式在許多 SaaS 產品中可能很常見，尤其是正在轉換為客服人員體驗的組織。此模型通常會持續存在，因為並非所有系統都以純 AaaS 形式交付。另請注意，多租用戶仍然適用於模型的代理程式。雖然代理程式可能內嵌在系統中，但仍然可以處理來自多個租用戶的請求。

詢問多租戶是否真正重要是很自然的。您可以說客服人員處理請求，因此支援租用的效果可能很小。但是，當我們深入探討多租戶代理的影響時，租用可能會直接影響代理程式如何影響工具、記憶體、資料和其他代理程式組件的存取、部署和設定方式，以支援個別租戶。租用也會影響擴展、限流、定價、分層和其他業務層面如何套用至客服人員的架構。

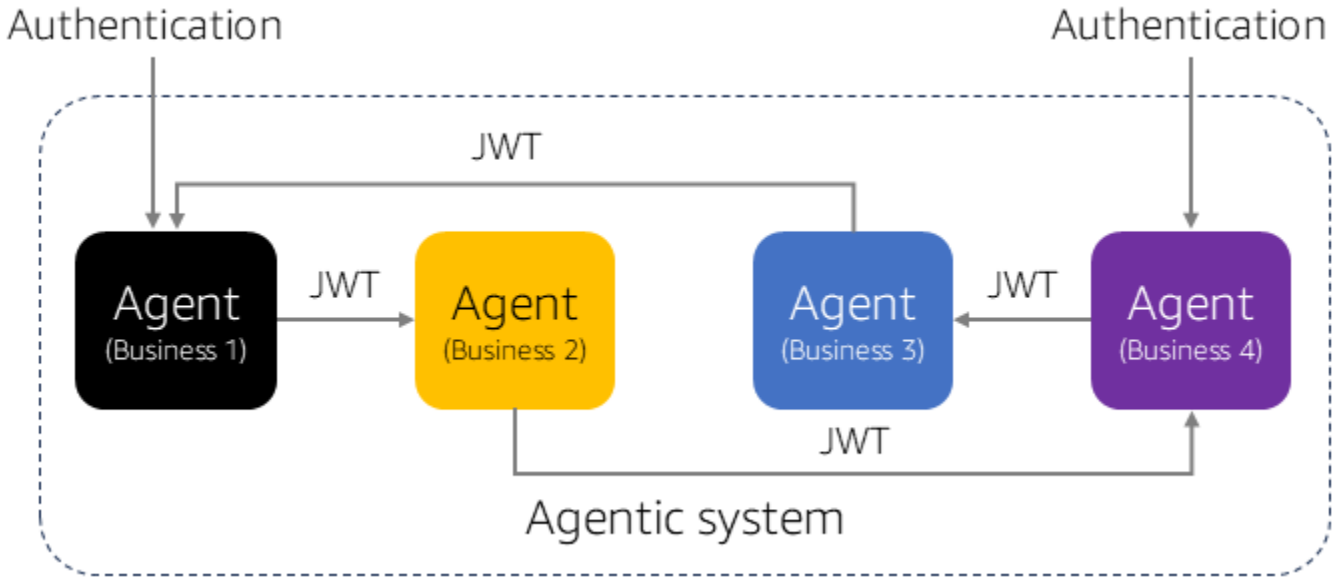
其中一個重點是，有代理使用案例需要多租用戶支援。挑戰是判斷多租戶如何塑造代理體驗的整體設計和架構。對於某些客服人員，多租戶支援代表差異化功能，允許客服人員將租戶特定內容套用至提供目標成果的客服人員。

在後續章節中，您將看到我們為了描述多租用戶 SaaS 架構而建立的術語和設計模式如何有用。AaaS 模型可以透過借用有用的層面來採用這些概念，在需要這些概念的地方引進新的客服人員特定概念。

身分、租戶內容和代理系統

將租戶內容新增至個別客服人員並不特別具有挑戰性。在許多執行個體中，團隊可以依賴將使用者和系統繫結到租用戶的典型機制，並將租用戶感知字符傳遞給客服人員。當我們考慮租戶內容和身分如何支援多個客服人員時，這很重要。在此模型中，租戶必須繫結至跨所有協同合作客服人員的身分。

一般而言，代理網域需要更交叉的身分模型，以符合代理系統目前和新興的需求。代理程式供應商需要支援作業系統隨附之唯一安全性、合規性和授權模型的身分機制。這在由客戶或其他客服人員組成系統的環境中特別具有挑戰性。每個加入的客服人員都必須將其身分和租戶內容連線至客服人員互動。下圖重點介紹agent-to-agent (a2a) 互動中的潛在身分和租戶內容挑戰。



此圖表顯示一系列供應商建置的客服人員，與我們涵蓋的客服人員系統互動。它現在已修改為身分和租戶內容。此案例是支援多個進入點的代理系統範例。我們假設此系統中的每個代理程式都需要自己的身分驗證機制，才能將系統或使用者解析為指定的租用戶。當這些客服人員互動時，租戶內容會傳遞至 JSON Web 字符 (JWT)，該字符將用於授權存取並將租戶內容注入客服人員。

在概念上，此案例的主要差異是客服人員獨立部署和操作，這表示每個客服人員都必須能夠解析其身分並授權存取。關鍵在於其身分必須具備一些分散式能力，才能處理更廣泛的代理系統的需求。客服人員共享租用戶內容的方式也必須保持一致。

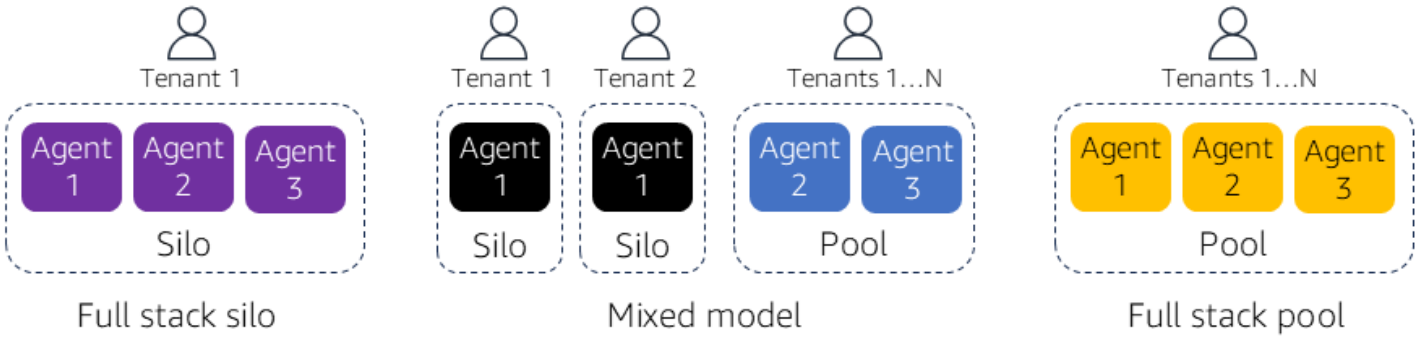
將 SaaS 商業價值套用至 AaaS

一般而言，當我們查看在as-a-service模型中執行任何系統時，我們會考慮體驗的性質，以及其技術和營運足跡如何推動業務成果。例如，採用 SaaS 時，組織會使用規模經濟、營運效率、成本設定檔和敏捷性來推動成長、利潤和創新。

做為 AaaS 交付的代理程式可能以類似的業務成果為目標。透過支援多個租用戶，客服人員可以將資源消耗與租用戶活動保持一致。這會產生傳統 SaaS 環境隨附的規模經濟。AaaS 也允許組織管理、操作和部署代理程式，讓代理程式頻繁發行，並提高代理程式供應商的靈活性。關鍵在於 AaaS 模型不依賴技術。它建立並推動促進成長、簡化採用和簡化操作的業務策略。

代理程式部署模型

在基本 AaaS 體驗中，供應商可能會使用各種模式部署代理程式。有許多因素會影響客服人員的部署方式，以滿足客戶、效能、合規、地理位置和安全需求。不同的部署策略會影響代理程式的設計、實作和使用方式。在這裡，我們可以介紹傳統多租戶術語來標記不同的部署策略。下圖顯示在 AaaS 環境中部署代理程式的不同排列。

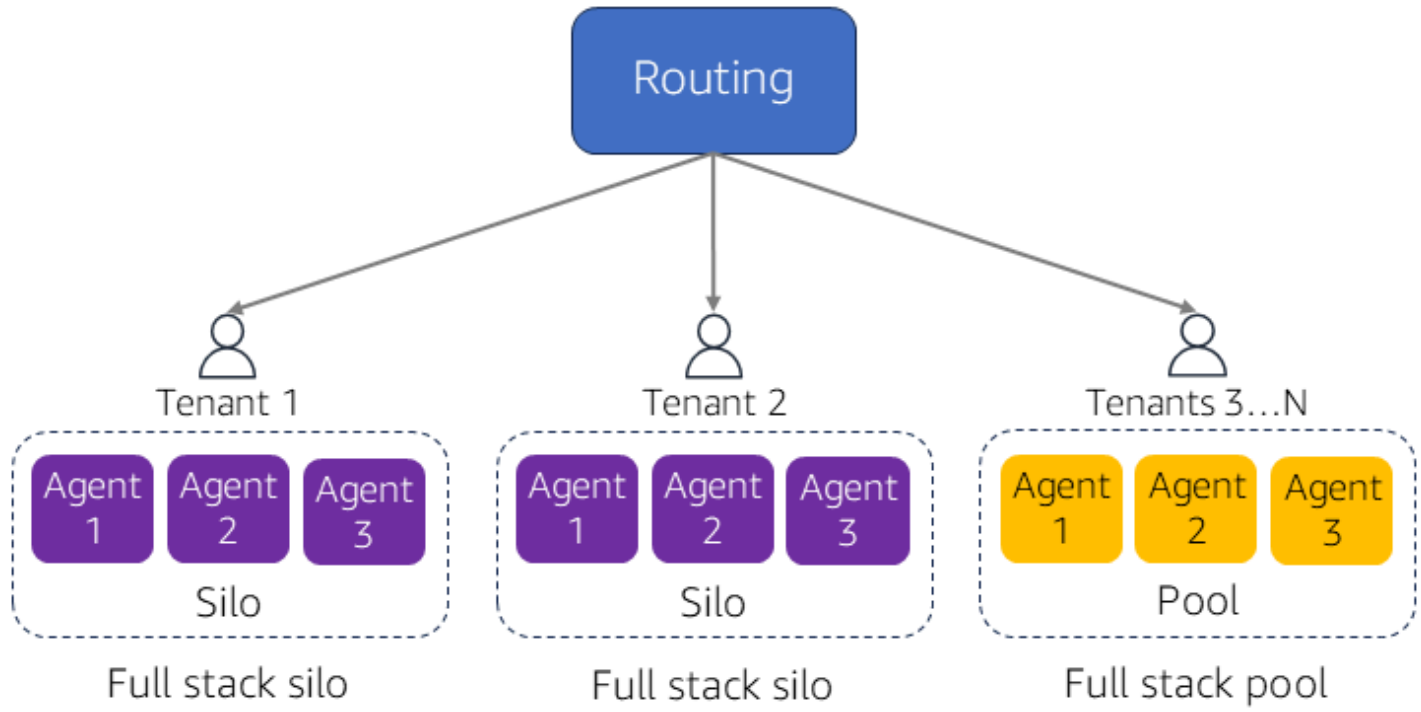


此圖表代表三種客服人員部署模式。左側是一個孤立模型，其中每個租用戶都有一個完全隔離的體驗和一組專用客服人員。在此案例中，客服人員不會跨租用戶共用運算、資源或執行環境。

中間範例說明混合模型，其中租用戶使用孤立和混合客服人員的組合。例如，代理程式 1 以孤立模式部署，而每個租用戶都會收到專用執行個體，而代理程式 2 和 3 則以集區模型運作，跨租用戶共用資源。

右側是全集區模型，其中所有客服人員都會在租用戶之間共用，提供傳統的多租用戶部署。在此案例中，租戶會利用常見的運算、記憶體和服務基礎設施來執行代理程式。

想法是客服人員可以在不同的部署模型中操作，使用專用（孤立）或跨租用戶共用（混合）的運算和相依資源。這些部署策略並非互斥。客服人員服務通常支援各種客戶需求，結合這兩種模型來平衡效能、隔離、成本和可擴展性。下圖顯示支援相同操作環境中多個部署組態的代理系統。

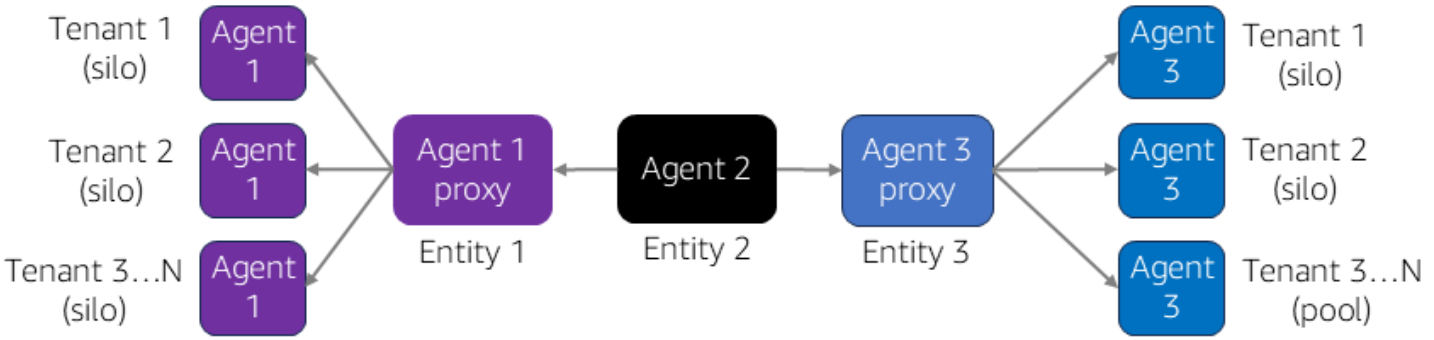


在此圖表中，代理程式供應商有三個透過代理程式即服務 (AaaS) 部署的代理程式。它們支援兩種類型的租戶。在左側，兩個租用戶具有合規和效能要求，可透過全堆疊孤立模型解決這些要求。右側的其餘租用戶會在租用戶共用資源的集區模型中執行。

如果目標是敏捷性和營運效率，請嘗試限制與支援每個租戶部署模型相關的效果。這表示設定路由和其他體驗機制，允許透過單一窗格管理、操作和部署代理程式。

如果您在低程式碼或無程式碼環境中建置代理程式，則不會有孤立或集區代理程式的概念。反之，代理程式可能由另一個代理程式完全管理。孤立和集區模型更適用於組織控制代理程式建構和足跡的環境。在此情況下，團隊應考慮要支援的部署模型。

表面上，這些部署模型不會直接影響代理程式在更廣泛的系統中的運作方式。代理程式可能無法直接了解部署在孤立或集區模型中的其他代理程式。反之，這些部署策略可以實作為環境中路由建構的一部分。下圖顯示如何使用路由策略實作孤立和集區模型的範例。



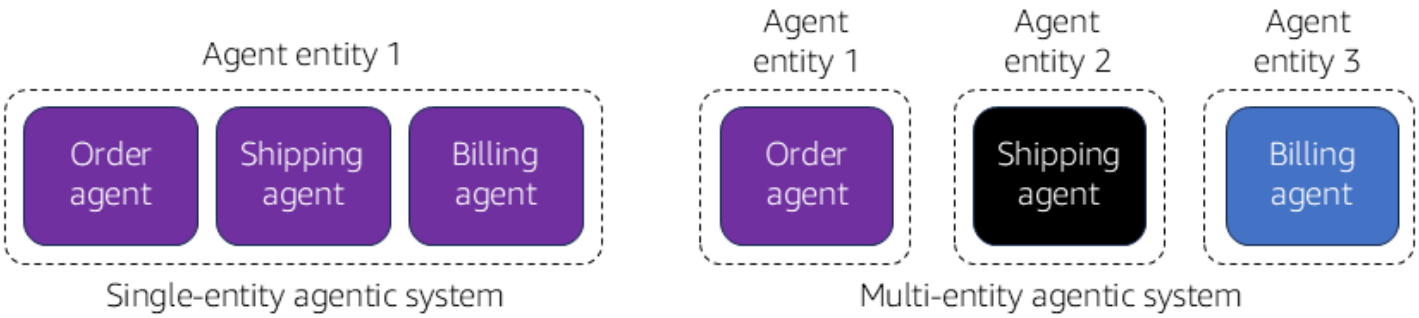
此範例包含來自三個不同供應商的三個客服人員。每個代理程式提供者都可以選擇實作自己的部署策略。例如，代理程式 1 使用代理將傳入請求分發給一組孤立租用戶代理程式。代理程式 2 不需要路由，並透過一個集區代理程式支援所有租戶請求。代理程式 3 是一種混合模型部署，其中一些租用戶處於孤立狀態，而其他租用戶則處於集區狀態。

如果您選擇支援這些部署模型的 和 方式取決於您解決方案的本質。您可能不需要支援任一模型。不過，您可以擁有必須考慮支援此策略的執行個體，例如合規、雜訊鄰近、效能或分層。

介紹和套用租戶內容

如果我們建置支援多租用戶的代理程式，我們首先必須考慮如何設定租用戶內容，這些內容將用於在代理程式實作中套用租用戶特定的政策、策略和機制。

在最基本的層級，您可以透過我們在傳統多租戶架構中使用的常見工具和機制，將租戶內容引入客服人員。這可能是透過 API 金鑰、OAuth 或各種其他驗證機制。此範例著重於將已驗證的系統或使用者解析為保存租用戶內容的 JSON Web 字符 (JWT) 金鑰。然後，JWT 會透過系統傳播。當我們考慮如何編寫代理系統時，這變得更加有趣。下圖顯示兩種代理環境的範例。



在此圖表中，左側的模型代表代理程式系統，其中所有代理程式都由單一實體擁有、管理和託管。當您完全控制整個體驗時，您可以使用典型的策略，透過每個代理程式傳遞租戶。

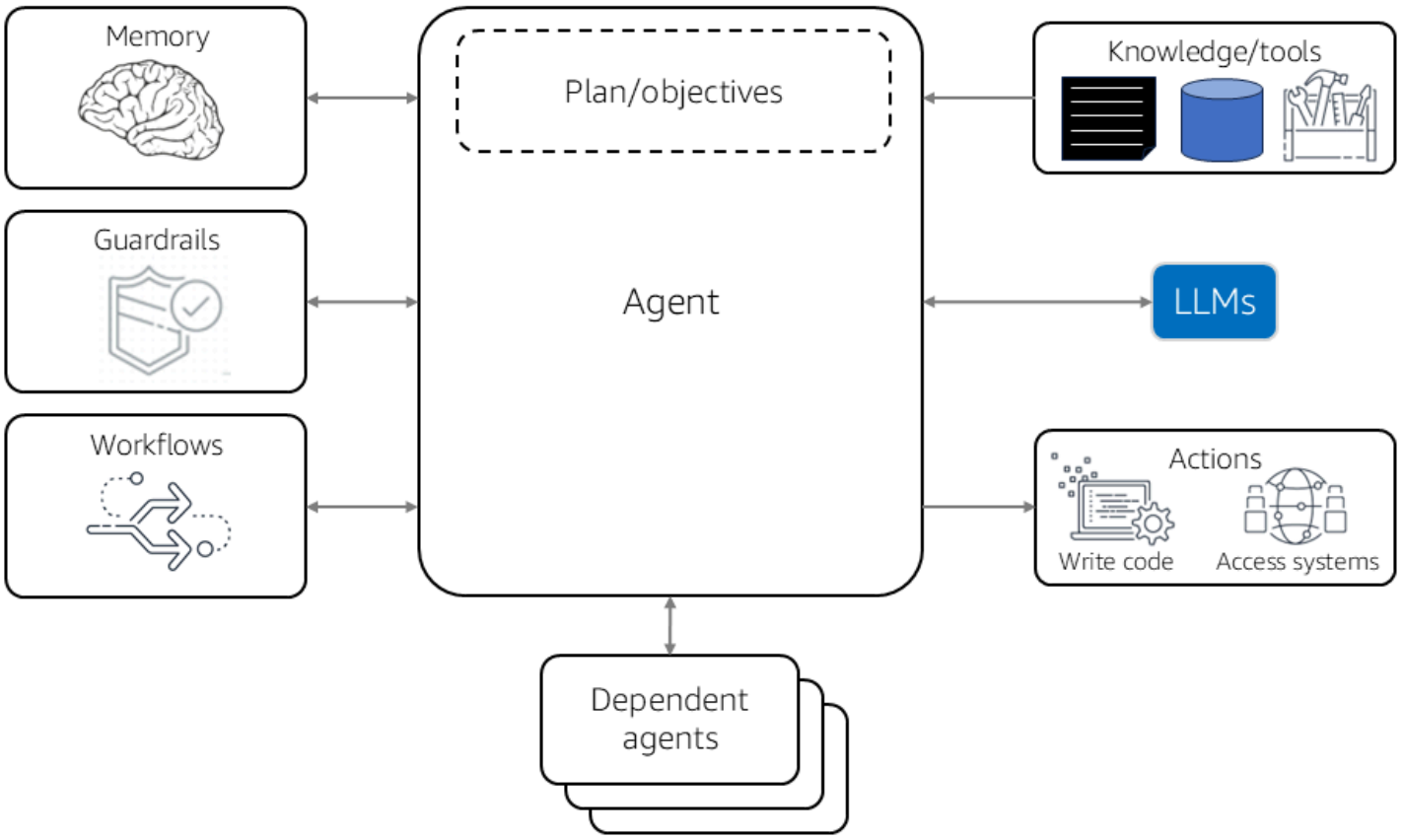
右側的模型可能更常見，代表跨越多個實體的客服人員系統。代理程式是獨立建置、管理和操作，因此每個代理程式都有自己的身分驗證和授權機制。這裡的挑戰是，我們需要一種通用的方式來解析和共用這些客服人員之間的租戶內容。這倚賴更分散的模型，其中每個代理程式都必須能夠驗證系統或使用者，並根據套用的機制將其解析為租戶。

建置租戶感知代理程式

多租戶會影響我們實作個別客服人員的方式。當客服人員處理請求時，請考慮租戶內容如何影響客服人員存取資料、做出決策和叫用動作的方式。為了更好地了解多租用戶如何影響代理程式的描述檔，首先確定建構模組如何成為任何代理程式的一部分。

挑戰是客服人員的範圍、性質和設計只是具體的，因為供應商會自行選擇客服人員體驗的設計。最後，代理程式的重點是，它是一種自主學習服務，可以存取各種工具、資料來源和記憶體，以確定如何最好地解決任務。

確切了解代理程式使用的策略和模式並不重要。在多租用戶模型中，更重要的是識別代理程式的各個部分如何設定、存取和套用。考慮一個潛在的代理程式環境，它依賴一系列的資源和機制來實現其目標。下圖顯示這類代理程式的範例。

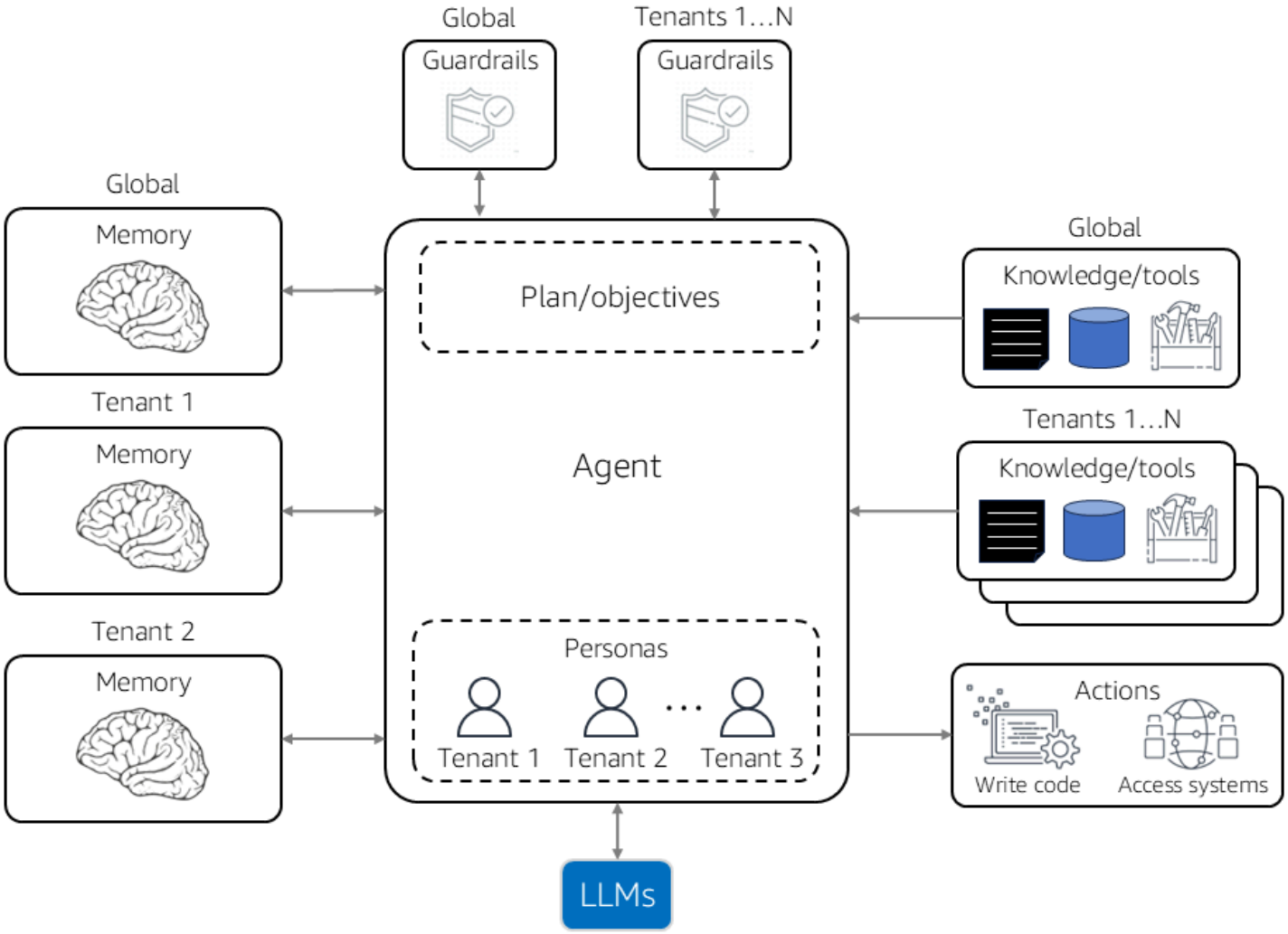


此圖表代表全方位的代理程式可能性，顯示可結合以實現目標的各種工具和機制。在圖表左側，請注意代理程式如何依賴記憶體作為其內容的一部分、定義引導其活動的政策的護欄，以及導向特定任務的工作流程。有些 可能認為不應在此內容中包含工作流程，但在某些情況下，工作流程對於客服人員體驗而言是不可或缺的。

圖表右側顯示知識和工具等輸入如何提供額外的洞見和內容，以增強客服人員的功能。然後，代理程式會輸出動作，例如編寫程式碼或存取系統。圖表底部顯示客服人員如何依賴一個或多個內部或第三方客服人員，這些客服人員可以做為更廣泛的系統的一部分進行協調。

我們現在可以思考引進多租用戶意味著什麼。租用會強制我們考慮代理程式如何以及在何處引進可決定行為和動作的策略和機制。這為我們對客服人員的知識、學習、工具和記憶體的想法提供了另一個維度。

現在，讓我們考慮如何修改此模型以支援多租用戶。下圖顯示多代理程式模型的範例。



在此圖表中，我們介紹了租戶角色，旨在塑造客服人員如何整合租戶內容。例如，在圖表的左側，代理程式記憶體會遭到修改，以支援租戶特定的記憶體。在 代理程式支援租戶特定知識和工具的圖表右側也是如此。相同的支援也會套用至護欄。

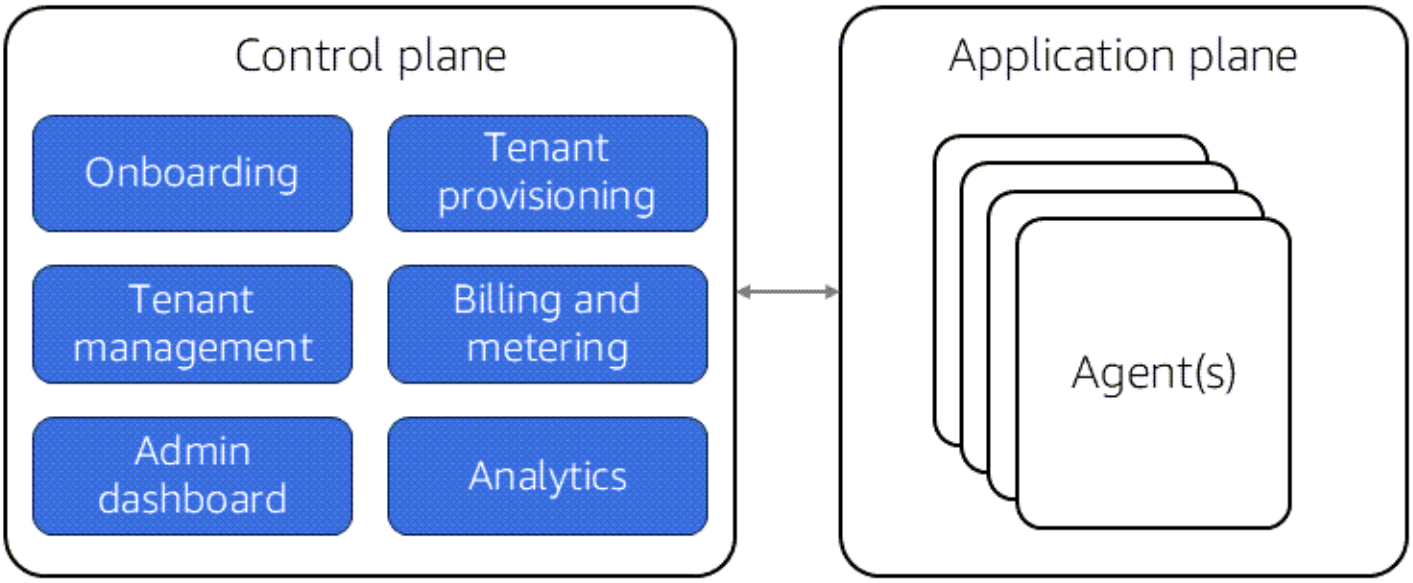
這可能是一個極端的範例，因為並非多租用戶代理程式的所有層面都需要每個租用戶的資源。重點是，您應該考慮為特定租戶量身打造代理程式如何增強其有效性。這種方法可讓您的代理程式提高其影響和價值，在其回應中提供更相關的內容，並開發專用功能。客服人員接著將能夠學習、調整和執行特別適合不同角色的任務。

主要想法是租戶內容會直接影響您建置客服人員的方式。它也可以塑造租戶與外部實體的互動，包括其他客服人員。建置多租戶代理程式會帶來傳統挑戰，例如雜訊鄰里、租戶隔離、分層、限流和成本管理。代理程式的設計和架構必須解決這些基本多租戶概念，我們將在下一節中探討這些概念。

在代理環境中使用控制平面

多租戶最佳實務通常會將實作分成兩個不同的部分：控制平面和應用程式平面。控制平面提供單一窗格，可存取跨越環境租用戶的操作、管理和協同運作機制。應用程式平面是商業邏輯、功能和功能所在的位置。

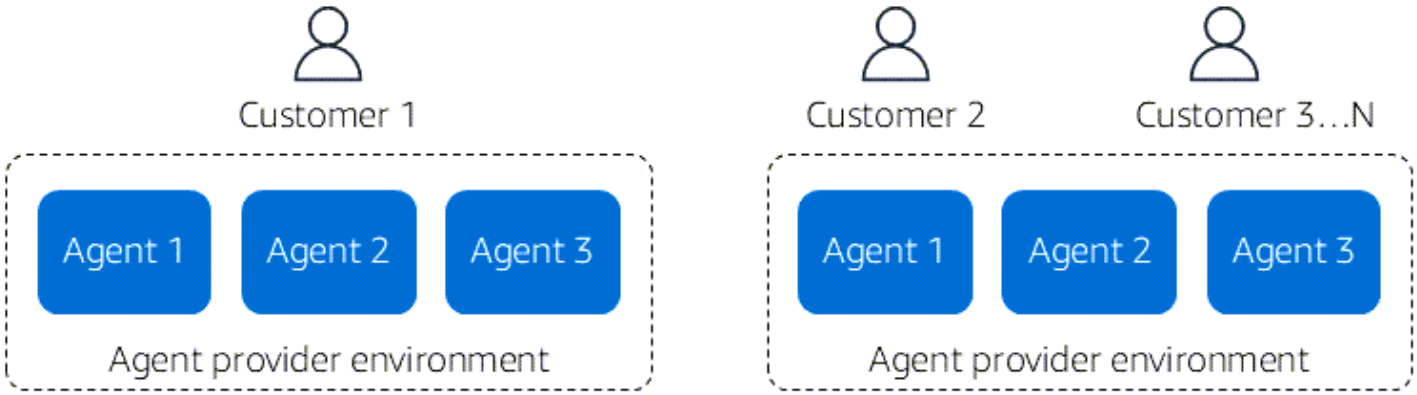
此責任劃分也適用於代理程式模型。多租戶代理程式需要一定程度的集中式管理、操作和洞察，而且透過控制平面持續解決這些需求是合理的。下圖顯示這些平面如何在客服人員即服務 (AaaS) 環境中分割的概念檢視。



此圖顯示傳統的控制和應用程式平面分離。新功能是，控制平面現在管理組成 AaaS 環境的代理程式。控制平面會與所有代理程式互動，因為我們假設代理程式是由一個供應商建置、管理和部署。

此模型引入了額外的複雜性層，特別是在客服人員生命週期和第三方協調中，但保留了問題的基礎分離。控制平面仍然透過協調客服人員的組態、提供租戶和客服人員可觀測性、收集用於計費的消耗和計量資料，以及管理租戶政策，來提供相同的核心功能。

如果您考慮多代理程式系統納入來自各種供應商的代理程式，則此案例會變得更加複雜。下圖顯示這類模型的範例。



此圖表描述來自不同供應商的四個代理程式，這些代理程式屬於多代理程式系統的一部分。第三方供應商仍然會操作和部署每個代理程式，這些代理程式設定為啟用來自一或多個供應商的授權存取。不過，代理程式會保持在提供者的控制下，因此每個代理程式都會維護自己的控制平面。

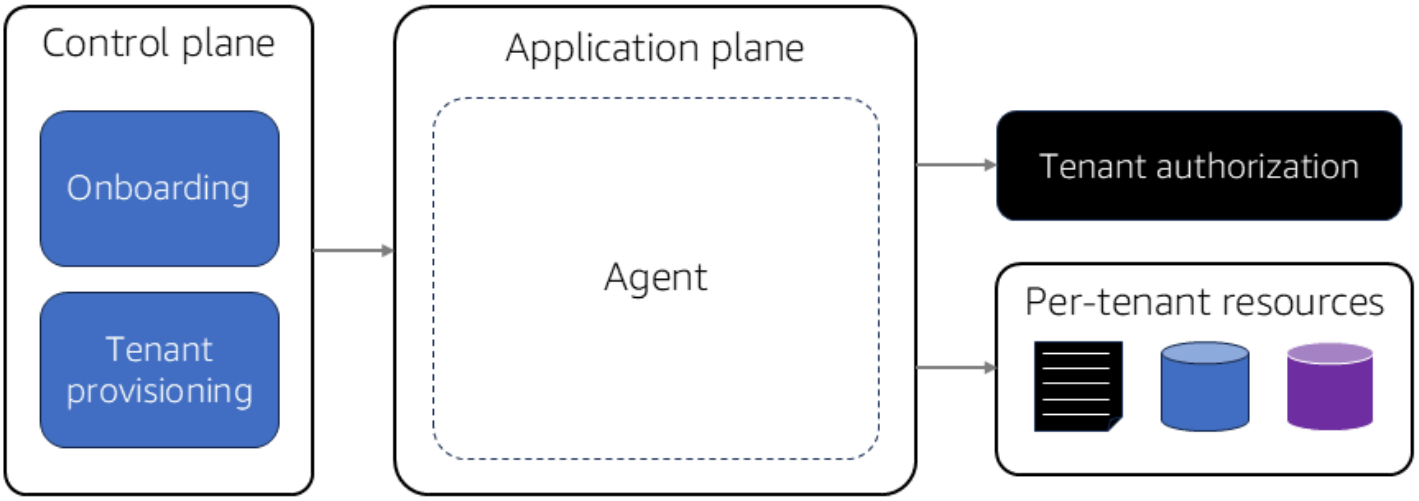
基本上，這些多租戶客服人員的行為就像是與其他客服人員整合的第三方服務。因此，他們必須擁有自己的控制平面，才能提供代理程式功能的集中式操作、組態和管理。

我們假設客服人員是在供應商託管體驗中執行的獨立服務。但是，在客服人員消費者對如何和在何處託管客服人員施加更多限制的情況下，這可能並不明確。

將租戶加入客服人員

加入通常是任何 AaaS 環境的重要部分。如何建立、設定和佈建租用戶通常涉及許多移動組件、整合和工具。客服人員加入體驗可能需要在 AaaS 控制平面中找到的相同服務，其中包括租用戶身分、分層、佈建每個租用戶資源，以及設定租用戶政策。

客服人員加入的方法會受到客服人員環境的足跡和租用模型影響。孤立和集區客服人員各有自己的細微差別，使用單一客服人員或多個客服人員的選擇也會影響加入程序。下圖顯示加入如何影響客服人員組態的概念檢視。



每次您加入代理程式時，控制平面都必須採取必要步驟，讓租戶能夠存取代理程式。如何介紹租用戶會根據客服人員授權模式而有所不同，但假設您將建立租用戶身分，將客服人員請求與個別租用戶建立關聯。此租用戶內容會將代理程式套用到路由、範圍和控制存取，以決定代理程式體驗。

加入也可能需要您設定 代理程式使用的任何每個租用戶資源。控制平面的租用戶佈建服務在這裡會將您的代理程式連線至代理程式諮詢的租用戶特定資料和資源。

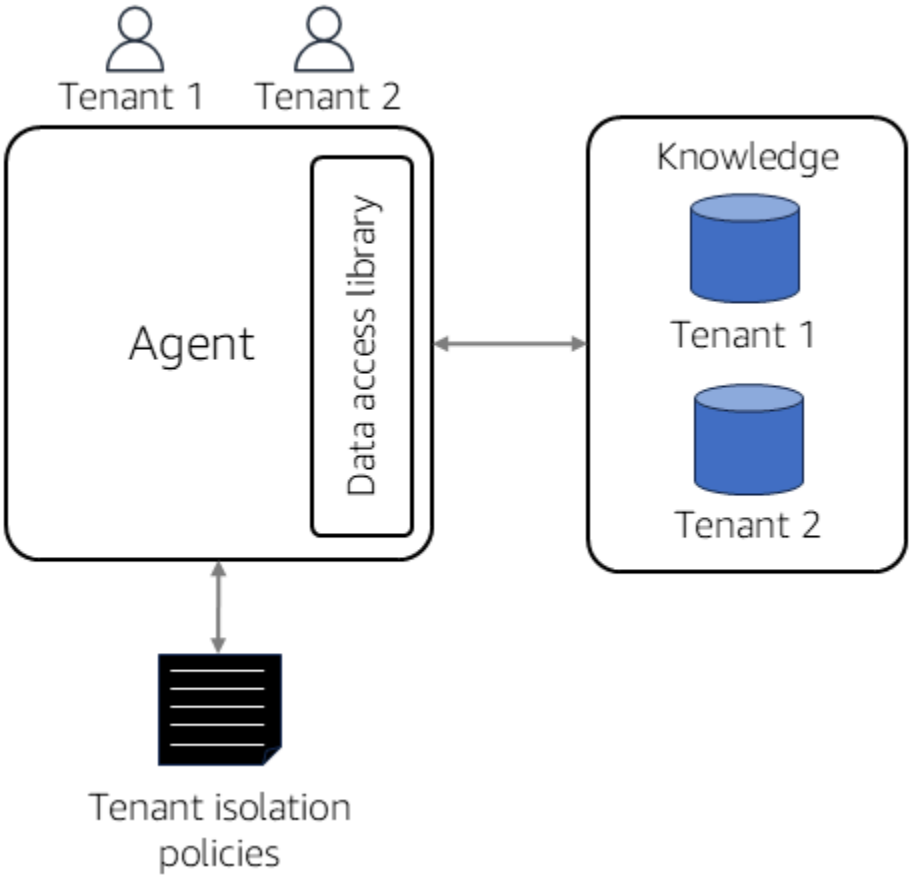
如果您的系統依賴整合第三方客服人員，您還必須在加入過程中解決這些客服人員的需求。這的運作方式取決於授權客服人員之間存取的安全和整合機制。理想情況下，協調和設定agent-to-agent身分驗證和授權的必要步驟是透過自動加入來處理。

強制執行租戶隔離

租戶隔離是適用於所有多租戶設定的概念。這表示您的政策和策略可確保一個租用戶無法存取其他租用戶資源。對於多租用戶代理程式，您可能需要引入有助於強制執行 和 代理程式的租用戶隔離要求的建構和機制。

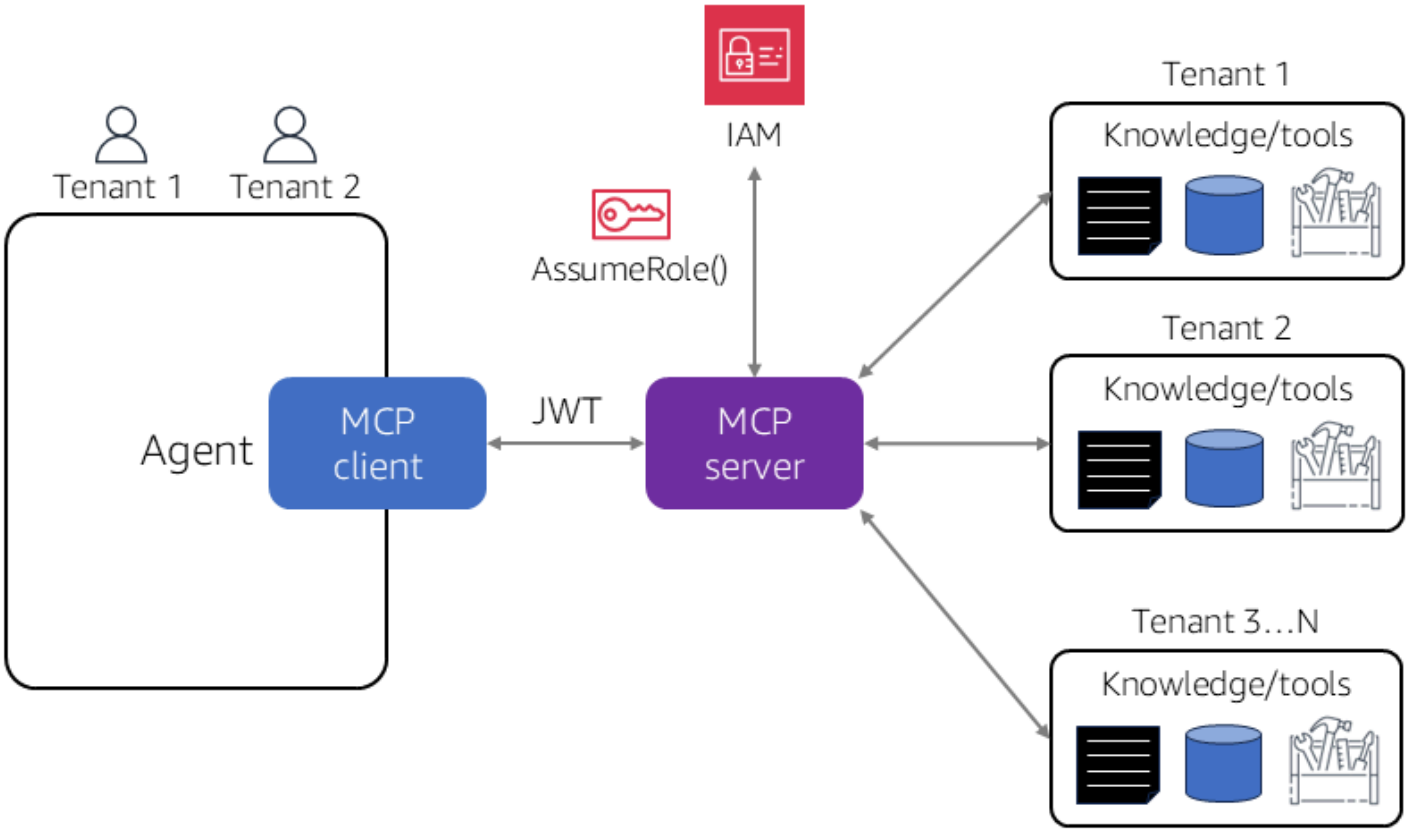
套用租用戶隔離就像使用傳統多租用戶系統的其他策略一樣。一般而言，當您建構 AaaS 架構時，請識別系統中請求或動作可存取資源的任何區域，以判斷請求是否超過任何租用戶界限。例如，微服務可能對每個租戶的專用 Amazon DynamoDB 資料表具有相依性。這需要您引進政策，以確保另一個租用戶無法存取一個租用戶的資料表。

在此執行個體中，請考慮透過代理程式鏡頭的租用戶隔離，以及其與每個租用戶任何資源的互動。下圖顯示客服人員如何套用租用戶隔離政策以控制租用戶資源存取的概念範例。



在此圖表的右側，代理程式具有存放在個別向量資料庫中的每個租用戶知識。當客服人員處理請求時，會檢查提出請求的租戶內容。根據此，代理程式會套用適當的隔離政策，以確保租用戶不會存取其指定界限以外的資料或資源。

如果您的代理程式使用模型內容通訊協定 (MCP)，它也可以實作租用戶隔離模型。下圖顯示如何引入 MCP 和套用隔離政策的範例。



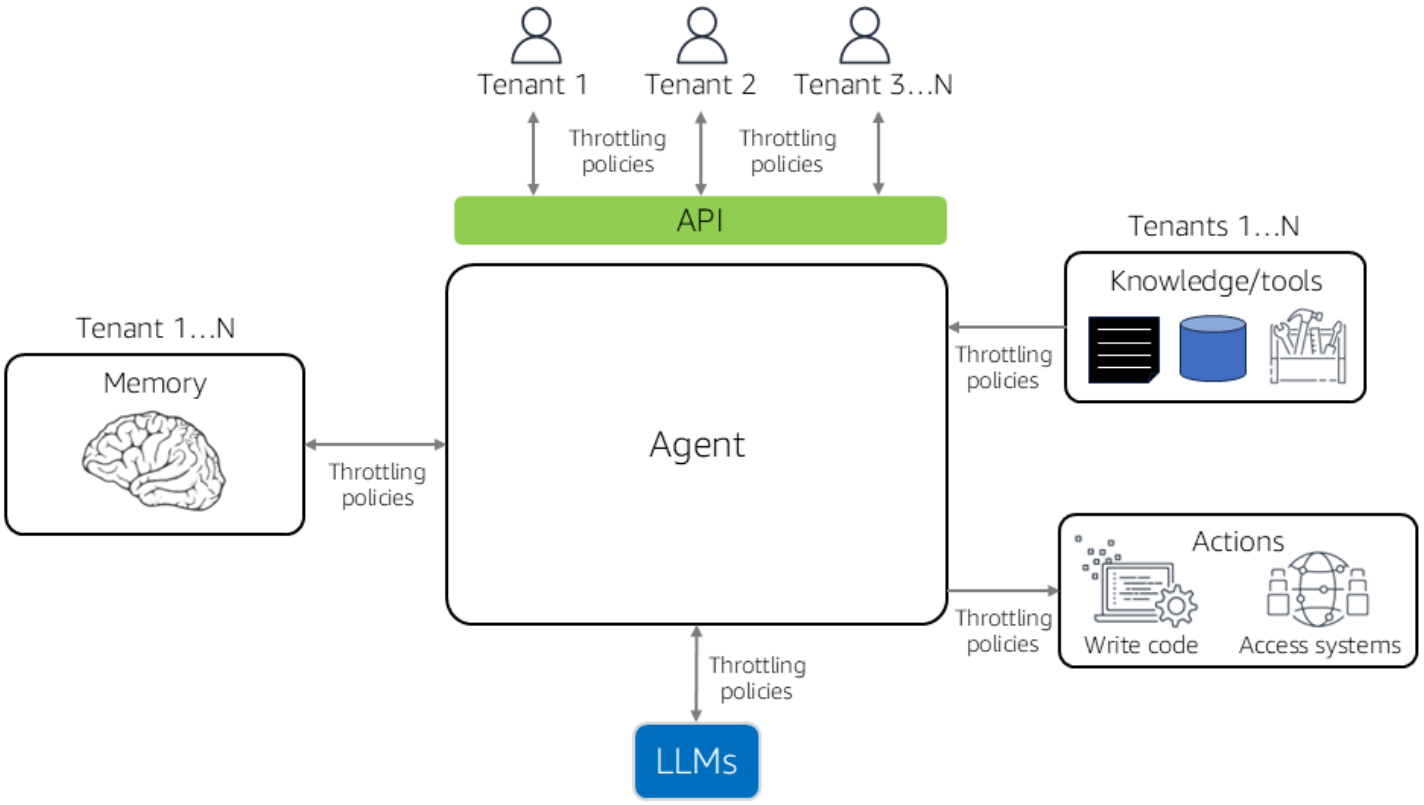
MCP 是一種標準化通訊協定，可讓代理程式用來與任何工具、資料和資源整合。在此範例中，MCP 用戶端和 MCP 伺服器會與圖表右側的租戶特定知識和工具互動。租用戶內容會從用戶端流向伺服器，而伺服器會使用此內容從 AWS Identity and Access Management (IAM) 服務取得租用戶範圍的登入資料。登入資料控制對每個租用戶資源的存取，確保一個租用戶可以存取另一個租用戶的資源。

隨著客服人員整合多租戶，他們必須引入機制，在處理請求時套用租戶隔離政策。在某些情況下，IAM 可協助限制對租戶資源的存取。在其他執行個體中，您可能需要引進其他工具或架構，才能套用租戶隔離政策。

雜訊鄰里和客服人員

在多租用戶 AaaS 環境中，多個租用戶共用代理程式，請思考在何處以及如何引入防止雜訊鄰近條件的政策。政策可以引入適用於所有消耗的一般用途限流，或者您可以擁有根據指定角色套用限流的租用戶或層型政策。與高級租用戶相比，您對基本層租用戶的消耗限制可能會更大。

此限流概念可套用至多個架構點。下圖顯示介紹雜訊鄰近政策的一些可能領域範例。



在我們先前的多代理程式實作審核中，我們檢查了您的代理程式可以使用的不同資源，強調了代理程式中每個租戶資源的可能性。每個接觸點都是引入限流政策的潛在區域，這有助於確保租戶不超過系統或租戶分層政策的消耗限制。

引入雜訊鄰近保護的最佳位置是在租戶共用資源的架構點。如果單一租用戶使用不成比例，這些共用或集區元件，例如運算、記憶體、APIs 和大型語言模型，最容易降低效能。

套用限流的一個自然位置是在客服人員的進入點，有時稱為「外部邊緣」。在這裡，您可以在客服人員開始處理請求之前，引進全域或以tenant-tier-based費率限制。調節也可以在執行路徑中更深入地套用，例如當代理程式呼叫 LLM、存取記憶體或叫用共用工具時。

這些政策可協助您強制執行公平使用、在負載下維持客服人員彈性，以及維持租用戶間的一致體驗。根據您的目標，您可能會專注於一般系統保護（彈性）或精細管理租戶體驗（例如，使用層型權利）。

資料、操作和測試

代理程式和資料擁有權

客服人員實作的檢閱反白顯示了客服人員倚賴指定租戶資料的案例。在此執行個體中，請考慮資料生命週期，更重要的是，考慮其存放位置。這對資料性質影響代理程式存取方式的產業和使用案例尤其重要。

AaaS 供應商必須評估如何解決多租戶環境中的資料問題，這可能會影響客服人員的加入、隔離和操作。適用的細微差別和策略會根據您使用的工具、技術和資料而有所不同。您可以透過許多方式處理此問題，這是您在建立任何 AaaS 產品時需要注意的事項。

多租戶客服人員操作

當您建構代理程式環境時，請考慮如何操作和管理代理程式。身為供應商，您需要指標、資料、洞見和日誌，可讓您監控客服人員的運作狀態、規模和活動。這在多租用戶代理程式環境中更為明顯，您會希望了解個別租用戶如何使用代理程式資源。

當您需要客服人員互動的洞察時，這在多客服人員設定中更為重要。能夠分析和追蹤客服人員之間的活動，對於疑難排解影響系統規模、準確性和效能的問題至關重要。

營運團隊也可以描述 LLM 互動，以更好地了解客服人員在 LLMs 上放置的負載。此資料對於精簡代理程式實作至關重要。它也可以讓營運團隊了解客服人員和租用如何影響系統的整體成本設定檔。

訓練和測試多租戶代理程式

與建置代理程式相關聯的一個挑戰是他們預期會學習和發展。這也表示，在將代理程式移至生產環境之前，我們必須測試代理程式、精簡代理程式，並提高其準確性。您可以在許多領域檢查和評估客服人員是否正確評估和分類意圖，或選擇和叫用適當的工具和動作。變數清單很廣泛，但這最終是為了確保您的代理程式找到可實現目標的結果。

檢查與測試代理程式相關的所有移動組件和原則超出本文件的範圍，但請注意，測試策略會增加多租戶 AaaS 環境的複雜性。例如，如果代理程式具有資料、記憶體和其他關聯式套用至每個租用戶的建構，則每個租用戶資源都可以塑造代理程式的結果。

如果您使用 代理程式來模擬案例，您可能需要針對租戶特定的使用案例擴展模擬。相應地，您必須精簡驗證程序，以允許每個租用戶驗證條件不同的執行個體。

考量和討論

SaaS 適用於何處？

產業專家積極討論客服人員如何影響軟體即服務 (SaaS) 環境。雖然客服人員確實正在為許多系統變更軟體，但建議客服人員讓交付模型過時是一段延伸。有些 SaaS 提供者可能會因為採用代理程式而中斷，有些供應商可能會完全重新思考其價值主張，方法是倚賴代理程式即服務 (AaaS) 模型。其他人可能會選擇性地引進客服人員來解決特定需求，進而達到平衡。

本主題很有趣，因為採用最佳 SaaS 原則可能代表 SaaS 的下一個演變。這可能表示 SaaS 正在進行，或者可能表示 SaaS 的基本原則正在以代理程式為基礎的模型中封裝和實現。決定術語最終到達何處可能較不重要，但概念中的 SaaS 似乎不太可能消失。代理程式更有可能塑造 SaaS 足跡。

最終，我們必須決定哪些策略可以套用到 AaaS，這表示組織能夠採用代理架構和商業策略，以便供應商可以最大限度地提高其代理系統的效率、價值和影響。代理程式不是黑框。客服人員會耗用資源、擴展操作、依賴資料並產生成本 - 提供者必須解決的所有因素。客服人員提供者必須評估多租戶原則如何塑造服務方案並最佳化營運模型。

說明

隨著設計因網域、預期使用案例和目標產業而異，代理程式環境持續演進。此演變的一部分包括進一步完善我們對於架構師在設計和建置代理程式時考慮的策略、模式和權衡的觀點。

完整的客服人員策略必須同時符合商業和技術目標。這包括定義目標市場和角色、建立定價和資源管理策略，以及判斷客服人員如何適應更大的系統。這些考量事項在交付 AaaS 時特別重要，其中擴展、成本效益和創新是主要目標。

操作功能同樣重要。環境必須支援監控客服人員活動、運作狀態指標和用量模式。這在多代理程式系統中變得更加複雜，其中操作必須跨獨立代理程式進行協調。

整體而言，這項客服人員討論只會抓取可能屬於代理系統的各種架構考量的表面。除了選擇適當的工具、架構和 LLMs 之外，成功還取決於建立符合可擴展性、效率、部署和多租用戶業務需求的架構。

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 7 月 14 日

AWS 規範性指引詞彙表

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

數字

7 R

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

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

A

ABAC

請參閱屬性型存取控制。

抽象服務

請參閱 受管服務。

ACID

請參閱原子性、一致性、隔離性、耐久性。

主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但比主動-被動遷移需要更多的工作。

主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫會在資料複寫至目標資料庫時處理來自連線應用程式的交易。目標資料庫在遷移期間不接受任何交易。

彙總函數

在一組資料列上運作的 SQL 函數，會計算群組的單一傳回值。彙總函數的範例包括 SUM和 MAX。

AI

請參閱人工智慧。

AIOps

請參閱人工智慧操作。

匿名化

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

反模式

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

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體攻擊。

應用程式組合

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

人工智慧 (AI)

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

人工智慧操作 (AIOps)

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

非對稱加密

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

原子性、一致性、隔離性、耐久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱《 AWS Identity and Access Management (IAM) 文件》中的 ABAC for AWS。

授權資料來源

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

可用區域

中的不同位置 AWS 區域 ，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、低延遲網路連線能力。

AWS 雲端採用架構 (AWS CAF)

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

AWS 工作負載資格架構 (AWS WQF)

一種工具，可評估資料庫遷移工作負載、建議遷移策略，並提供工作預估值。 AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

B

錯誤的機器人

旨在中斷或傷害個人或組織的機器人。

BCP

請參閱業務持續性規劃。

行為圖

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

大端序系統

首先儲存最高有效位元組的系統。另請參閱 Endianness。

二進制分類

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

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 （藍色） 中執行目前的應用程式版本，並在另一個環境 （綠色） 中執行新的應用程式版本。此策略可協助您快速復原，並將影響降至最低。

機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。有些機器人有用或有益，例如在網際網路上編製資訊索引的 Web 爬蟲程式。有些其他機器人稱為惡意機器人，旨在中斷或傷害個人或組織。

殭屍網路

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

分支

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

碎片存取

在特殊情況下，並透過核准的程序，讓使用者快速取得他們通常無權存取 AWS 帳戶 之 的存取權。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS 實作打破玻璃程序指標。

棕地策略

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

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

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

業務連續性規劃 (BCP)

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

C

CAF

請參閱AWS 雲端採用架構。

Canary 部署

版本對最終使用者的緩慢和增量版本。當您有信心時，您可以部署新版本並完全取代目前的版本。

CCoE

請參閱 Cloud Center of Excellence。

CDC

請參閱變更資料擷取。

變更資料擷取 (CDC)

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

混沌工程

故意引入故障或破壞性事件，以測試系統的彈性。您可以使用 AWS Fault Injection Service (AWS FIS) 執行試驗，為您的 AWS 工作負載帶來壓力，並評估其回應。

CI/CD

請參閱持續整合和持續交付。

分類

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

用戶端加密

在目標 AWS 服務 接收資料之前，在本機加密資料。

雲端卓越中心 (CCoE)

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

雲端運算

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

雲端操作模型

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

採用雲端階段

組織在遷移至 時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展雲端採用 (例如，建立登陸區域、定義 CCoE、建立營運模型)

- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

部落格文章中的 Stephen Orban 定義了這些階段： AWS 雲端 企業策略部落格上的邁向雲端優先之旅和採用階段。如需有關它們如何與 AWS 遷移策略相關的詳細資訊，請參閱遷移整備指南。

CMDB

請參閱組態管理資料庫。

程式碼儲存庫

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

冷快取

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

冷資料

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

電腦視覺 (CV)

使用機器學習從數位影像和影片等視覺化格式分析和擷取資訊的 AI 欄位。例如，Amazon SageMaker AI 提供 CV 的影像處理演算法。

組態偏離

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

組態管理資料庫 (CMDB)

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

一致性套件

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

持續整合和持續交付 (CI/CD)

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

CV

請參閱電腦視覺。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

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

資料偏離

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

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

架構架構，提供分散式、分散式資料擁有權與集中式管理。

資料最小化

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

資料周邊

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

資料預先處理

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

資料來源

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

資料主體

正在收集和處理其資料的個人。

資料倉儲

支援商業智慧的資料管理系統，例如 分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

資料庫處理語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱資料庫定義語言。

深度整體

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

深度學習

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

深度防禦

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在 上採用此策略時 AWS，您可以在 AWS Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，defense-in-depth方法可能會結合多重要素驗證、網路分割和加密。

委派的管理員

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

deployment

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

開發環境

請參閱 環境。

偵測性控制

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

開發值串流映射 (DVSM)

一種程序，用於識別對軟體開發生命週期中的速度和品質造成負面影響的限制並排定優先順序。DVSM 擴展了最初專為精簡製造實務設計的價值串流映射程序。它著重於透過軟體開發程序建立和移動價值所需的步驟和團隊。

數位分身

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

維度資料表

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

災難

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

災難復原 (DR)

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

DML

請參閱資料庫處理語言。

領域驅動的設計

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

DR

請參閱災難復原。

偏離偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源中的偏離，也可以使用 AWS Control Tower 來偵測登陸區域中可能影響控管要求合規性的變更。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html

DVSM

請參閱開發值串流映射。

E

EDA

請參閱探索性資料分析。

EDI

請參閱電子資料交換。

邊緣運算

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

電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱什麼是電子資料交換。

加密

將人類可讀取的純文字資料轉換為加密文字的運算程序。

加密金鑰

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

端序

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

端點

請參閱 服務端點。

端點服務

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

企業資源規劃 (ERP)

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

信封加密

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

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

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

epic

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

ERP

請參閱企業資源規劃。

探索性資料分析 (EDA)

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

F

事實資料表

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

快速失敗

一種使用頻繁和增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在 中 AWS 雲端，像是可用區域 AWS 區域、控制平面或資料平面等邊界會限制故障的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱AWS 故障隔離界限。

功能分支

請參閱分支。

特徵

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

功能重要性

特徵對於模型的預測有多重要。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱 的機器學習模型可解譯性 AWS。

特徵轉換

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

少量擷取提示

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

FGAC

請參閱精細存取控制。

精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。

閃切遷移

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

FM

請參閱基礎模型。

基礎模型 (FM)

大型深度學習神經網路，已在廣義和未標記資料的大量資料集上進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱什麼是基礎模型。

G

生成式 AI

已針對大量資料進行訓練的 AI 模型子集，可使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱什麼是生成式 AI。

地理封鎖

請參閱地理限制。

地理限制 (地理封鎖)

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

Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被視為舊版，而以幹線為基礎的工作流程是現代、偏好的方法。

黃金影像

系統或軟體的快照，做為部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

綠地策略

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

防護機制

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

H

HA

請參閱高可用性。

異質資料庫遷移

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

高可用性 (HA)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統旨在自動容錯移轉、持續提供高品質的效能，以及處理不同的負載和故障，並將效能影響降至最低。

歷史現代化

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種來源收集和存放資料。

保留資料

從用於訓練機器學習模型的資料集中保留的部分歷史標記資料。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

異質資料庫遷移

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

熱資料

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

修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性，通常會在典型 DevOps 發行工作流程之外執行修補程式。

超級護理期間

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

I

IaC

將基礎設施視為程式碼。

身分型政策

連接至一或多個 IAM 主體的政策，可定義其在 AWS 雲端 環境中的許可。

閒置應用程式

90 天期間 CPU 和記憶體平均使用率在 5% 至 20% 之間的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

IIoT

請參閱工業物聯網。

不可變的基礎設施

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

傳入 (輸入) VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

增量遷移

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

工業 4.0

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

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱建立工業物聯網 (IIoT) 數位轉型策略。

檢查 VPC

在 AWS 多帳戶架構中，集中式 VPC，可管理 VPCs 之間 （在相同或不同的 中 AWS 區域)、網際網路和內部部署網路之間的網路流量檢查。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

物聯網 (IoT)

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

可解釋性

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

IoT

請參閱物聯網。

IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 為 ITSM 提供了基礎。

IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需有關將雲端操作與 ITSM 工具整合的資訊，請參閱操作整合指南。

ITIL

請參閱 IT 資訊庫。

ITSM

請參閱 IT 服務管理。

L

標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會決定使用者可以看到哪些資料列和資料欄。

登陸區域

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

大型語言模型 (LLM)

預先訓練大量資料的深度學習 AI 模型。LLM 可以執行多個任務，例如回答問題、摘要文件、將文字翻譯成其他語言，以及完成句子。如需詳細資訊，請參閱什麼是 LLMs。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱標籤型存取控制。

最低權限

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

隨即轉移

請參閱 7 個 R。

小端序系統

首先儲存最低有效位元組的系統。另請參閱 Endianness。

LLM

請參閱大型語言模型。

較低的環境

請參閱 環境。

M

機器學習 (ML)

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

主要分支

請參閱分支。

惡意軟體

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

受管服務

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

製造執行系統 (MES)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

MAP

請參閱遷移加速計劃。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的建置機制。

成員帳戶

除了屬於組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。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 建議使用不可變的基礎設施作為最佳實務。

O

OAC

請參閱原始存取控制。

OAI

請參閱原始存取身分。

OCM

請參閱組織變更管理。

離線遷移

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

OI

請參閱 操作整合。

OLA

請參閱操作層級協議。

線上遷移

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

OPC-UA

請參閱開啟程序通訊 - 統一架構。

開放程序通訊 - 統一架構 (OPC-UA)

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

操作水準協議 (OLA)

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

操作整備審查 (ORR)

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

操作技術 (OT)

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

操作整合 (OI)

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

組織追蹤

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 支援使用 S3 AWS KMS (SSE-KMS) 的所有伺服器端加密中的所有 S3 儲存貯體 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。

依設計的隱私權

透過整個開發程序將隱私權納入考量的系統工程方法。

私有託管區域

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 查詢的資訊。如需詳細資訊，請參閱 Route 53 文件中的使用私有託管區域。

主動控制

旨在防止部署不合規資源的安全控制。這些控制項會在佈建資源之前對其進行掃描。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的控制項參考指南，並參閱實作安全控制項中的主動控制項。 AWS

產品生命週期管理 (PLM)

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

生產環境

請參閱 環境。

可程式設計邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

提示鏈結

使用一個 LLM 提示的輸出做為下一個提示的輸入，以產生更好的回應。此技術用於將複雜任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和相關性，並允許更精細、個人化的結果。

擬匿名化

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

發佈/訂閱 (pub/sub)

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

Q

查詢計劃

一系列步驟，如指示，用於存取 SQL 關聯式資料庫系統中的資料。

查詢計劃迴歸

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

R

RACI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

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)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除它，則稱為 RACI 矩陣。

回應性控制

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

保留

請參閱 7 個 R。

淘汰

請參閱 7 個 R。

檢索增強生成 (RAG)

一種生成式 AI 技術，其中 LLM 會在產生回應之前參考訓練資料來源以外的授權資料來源。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱什麼是 RAG。

輪換

定期更新秘密的程序，讓攻擊者更難存取登入資料。

資料列和資料欄存取控制 (RCAC)

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

RPO

請參閱復原點目標。

RTO

請參閱復原時間目標。

執行手冊

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

S

SAML 2.0

許多身分提供者 (IdP) 使用的開放標準。此功能會啟用聯合單一登入 (SSO)， AWS 管理主控台 讓使用者可以登入 或呼叫 AWS API 操作，而不必為組織中的每個人在 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

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 AWS 服務 接收資料的 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。

SIEM

請參閱安全資訊和事件管理系統。

單一故障點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱服務層級協議。

SLI

請參閱服務層級指標。

SLO

請參閱服務層級目標。

先拆分後播種模型

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

SPOF

請參閱單一故障點。

星狀結構描述

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

Strangler Fig 模式

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

子網

您 VPC 中的 IP 地址範圍。子網必須位於單一可用區域。

監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

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

系統提示

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

T

標籤

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

目標變數

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

任務清單

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

測試環境

請參閱 環境。

訓練

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

傳輸閘道

可以用於互連 VPC 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的什麼是傳輸閘道。

主幹型工作流程

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

受信任的存取權

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

調校

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

雙比薩團隊

兩個比薩就能吃飽的小型 DevOps 團隊。雙披薩團隊規模可確保軟體開發中的最佳協作。

U

不確定性

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

未區分的任務

也稱為繁重工作，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，通常可接受中等緩慢的查詢。

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。