



AWS 使用生成式 AI 在上轉換應用程式開發和維護操作模型

AWS 方案指引



AWS 方案指引: AWS 使用生成式 AI 在上轉換應用程式開發和維護操作模型

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

簡介	1
目標對象	1
目標	1
將生成式 AI 整合到 ADM 的優勢	2
了解 ADM 中的操作模型	3
IT 操作模型	3
ADM 操作模型	5
ADM 操作模型中的商務層元素	7
ADM 操作模型中的服務整合層元素	8
ADM 操作模型中的組織結構層元素	8
ADM 操作模型中的組織功能層元素	10
將生成式 AI 整合到 ADM 實務中	12
商務層	12
服務整合層	14
組織結構層	15
組織結構和角色	16
組織功能層	25
整合挑戰和緩解策略	27
動作區域和建議	28
建置採用 AI 技術的 ADM 目標操作模型	31
策略對齊元件	32
組織結構元件	32
人才和技能元件	33
治理和道德元件	33
效能測量元件	33
合作夥伴生態系統元件	33
技術和工具元件	34
程序元件	34
實作採用 AI 技術的 ADM 目標操作模型	35
實作 AI 驅動 ADM TOM 的藍圖	36
所有實作階段的最佳實務	39
後續步驟	40
Resources	41
附錄 A：架構	43

附錄 B：檢查清單	46
階段 1：基礎設定	46
階段 2：能力建置	48
階段 3：轉換擴展	50
文件歷史紀錄	53
詞彙表	54
#	54
A	54
B	57
C	59
D	61
E	65
F	66
G	68
H	69
I	70
L	72
M	73
O	76
P	79
Q	81
R	81
S	84
T	87
U	88
V	88
W	89
Z	90
.....	xci

AWS 使用生成式 AI 在上轉換應用程式開發和維護操作模型

Dhana Vadivelan , Amazon Web Services (AWS)

2025 年 4 月 ([文件歷史記錄](#))

組織目前在應用程式開發和維護 (ADM) 實務上面臨前所未有的挑戰。生成式 AI 從根本上改變了應用程式的建置、設計、測試、記錄和部署方式，進而改變了整個軟體開發生命週期 (SDLC)。

ADM 包含從業務需求分析到開發和維護的完整應用程式生命週期，代表管理應用程式的完整實務。SDLC 定義結構化方法和階段，以在此更廣泛的 ADM 架構中建置軟體。

為了協助組織進行採用 AI 技術的 ADM 實務轉型，此策略文件提供：

- 全面分析 AI 對 ADM 的影響，包括操作模型和角色特定變更
- 增強組織能力並解決關鍵挑戰的策略
- 用於建置和實作採用 AI 技術的 ADM 操作模型的架構
- 採用 AI 技術的 ADM 操作模型的分階段實作方法，從快速獲勝到完整 AI 整合

目標對象

建議下列對象使用此策略文件：

- IT 主管，例如技術長 (CTOs)、技術主管、技術主管、架構師和計畫經理
- 業務主管，例如資訊長 (CIOs)、資料長 (CDOs)、產品工程副總裁 (VPs) 和業務營運VPs

目標

此策略文件可協助您的組織達成下列目標：

- 檢查您目前的 ADM 操作模型，以轉換為 AI 時代。
- 解決生成式 AI 整合的獨特挑戰。
- 實作分階段轉換策略，將生成式 AI 整合到組織的 ADM。

將生成式 AI 整合到 ADM 的優勢

對於 IT 領導者而言，將生成式 AI 整合到組織的 ADM 可以提供下列優點，以增強組織的功能：

- 透過快速原型設計和回應式軟體開發加速創新週期。
- 自動化架構定義、程式碼產生和測試中的例行任務。
- 提升軟體品質和可靠性，將瑕疵降至最低並降低風險。
- 透過處理增加的複雜性和開發量來改善營運可擴展性。

對於業務領導者而言，生成式 AI 的整合可以提供超越技術改進的優勢，以創造商業價值：

- 更快速地交付以客戶為中心的應用程式，快速適應市場需求。
- 透過 AI 技術提高營運效率，獲得競爭優勢。
- 將您的組織定位為 AI 驅動型開發的領導者，吸引頂尖人才。
- 透過改善生產力和最佳化資源配置來實現成本效益。

各產業的早期採用者正在 ADM 中使用 AWS 生成式 AI 服務的優勢：

- 開發速度 – 使用 Amazon Q Developer [BlackBerry](#)改善 SDLC 敏捷性和品質。
- 程式碼產生 – [BT 群組](#)使用 Amazon CodeWhisperer 自動化 12% 的重複性任務，這會成為 Amazon Q Developer 的一部分。
- 現代化 – [Novacomp](#) 使用 Amazon Q Developer 將 Java 應用程式現代化時間從 3 週縮短為 50 分鐘。
- 文件 – [ADP](#) 使用 Amazon Q Developer 將舊版系統文件時間從幾週縮短到不到一天。
- 生產力 – [澳洲國家銀行](#)使用 Amazon Q Developer 來實現 50% 接受 AI 產生的程式碼建議。
- 應用程式現代化 – [Deloitte](#) 使用 Amazon Q Developer 來加速現代化階段，降低專案複雜性和完成時間。[TCP](#) 使用 Amazon Q Developer 來加速大型主機現代化、快速分析和記錄舊版 COBOL 程式碼。
- 應用程式遷移 – [Cognizant](#)使用 Amazon Q Developer 自動化複雜的遷移程序，提高轉型專案的速度和簡單性。[HCLTech](#) 也使用 Amazon Q Developer，採用 AI 代理程式來加速 VMware、.NET 和大型主機工作負載。
- 應用程式效率 – [IBM 諮詢在](#) 上的 AI 型 SDLC 解決方案 AWS Marketplace 利用 Amazon Bedrock 在整個應用程式生命週期中提高效率和品質。

了解 ADM 中的操作模型

在探索 AI 對 ADM 的轉換影響之前，請務必了解 ADM 環境中操作模型的基礎知識。本節提供典型 IT 操作模型的概觀。然後，檢閱 ADM 操作模型的關鍵元件和層，這會為 AI 驅動的變更設定階段。

在本節中：

- [典型 IT 操作模型概觀](#)
- [ADM 操作模型概觀](#)

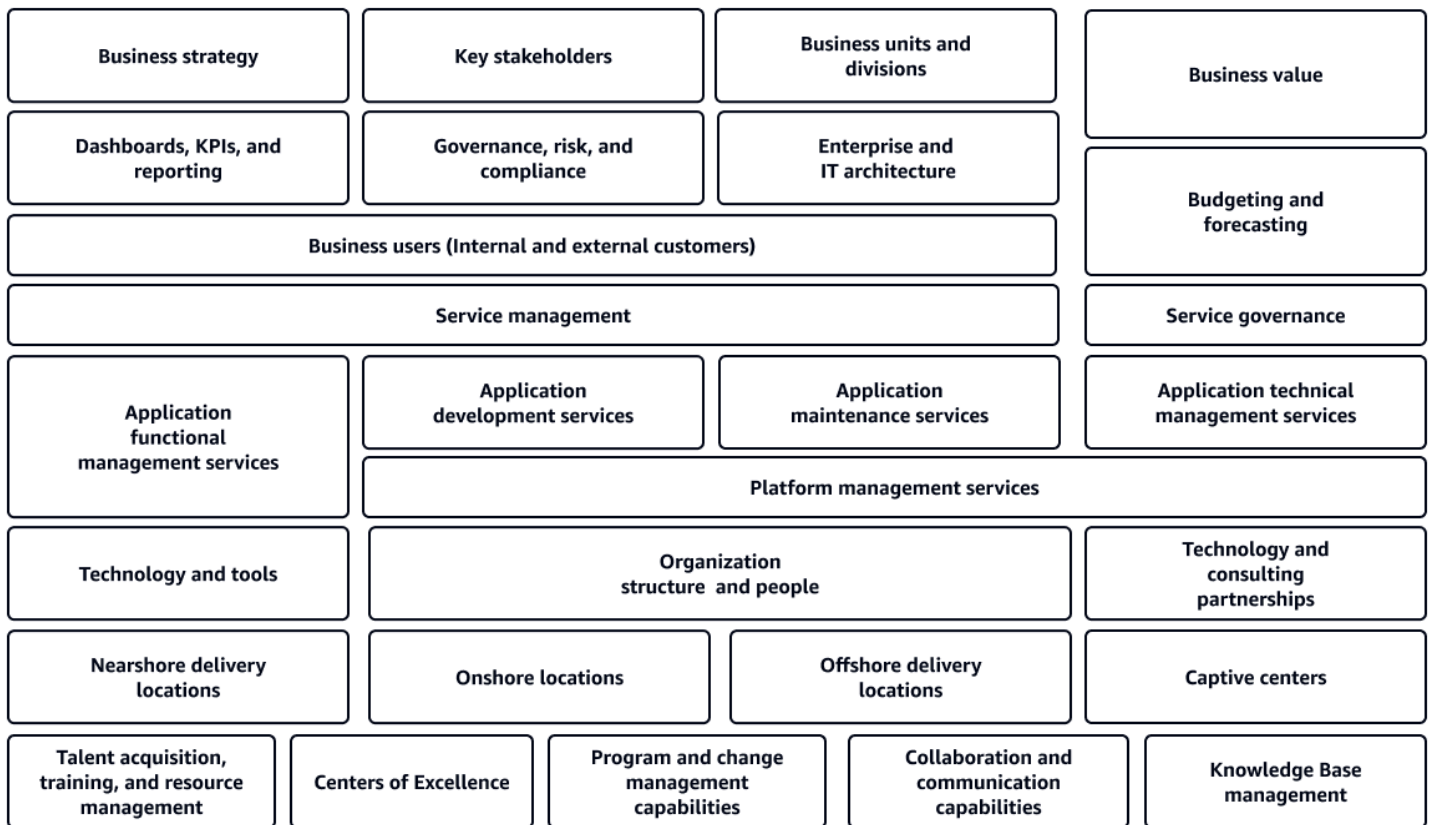
典型 IT 操作模型概觀

操作模型是任何組織中成功交付 IT 服務的基礎。這是定義組織如何透過其操作建立和交付價值的藍圖。營運模型的核心是將人員、程序和各種技術與業務策略保持一致。（如需操作模型的詳細資訊，請參閱 Open Group 網站上的[定義 IT 操作模型](#)。）

如下圖所示，典型的 IT 操作模型包含多個關鍵元件：

- 組織結構和角色
 - 主要利益相關者
 - 業務單位和部門
 - 商業使用者（內部和外部客戶）
 - 人員角色
 - 技術和諮詢合作夥伴關係
- 治理和決策架構
- 企業和 IT 架構
- 核心程序和工作流程
 - 商業策略
 - 商業價值
 - 預算和預測
 - 應用程式功能管理服務
 - 應用程式開發服務
 - 應用程式維護服務
 - 應用程式技術管理服務

- 平台管理服務
- 技術和工具
- 效能指標
 - 儀表板、關鍵績效指標 (KPIs) 和報告
- 組織功能
 - 程式設計和變更管理
 - 協作和通訊
 - 知識庫管理
- 文化和工作方式
 - 人才招聘、培訓和資源管理
 - 卓越中心 (COE)
 - 近岸交付位置
 - 海外據點
 - 海外交付位置
 - 網頁擷取中心



設計良好的操作模型不僅能說明day-to-day操作。這是可提升競爭優勢的策略性資產。操作模型可讓組織快速回應市場變化、有效創新，並提供更高的客戶價值。設計良好的操作模型的關鍵優勢是適應性。您組織的操作模型必須靈活地支援其所選的實務，同時保持一致性和效率。無論您為 ADM 使用傳統瀑布法、敏捷架構或混合方法，此調整能力都適用。

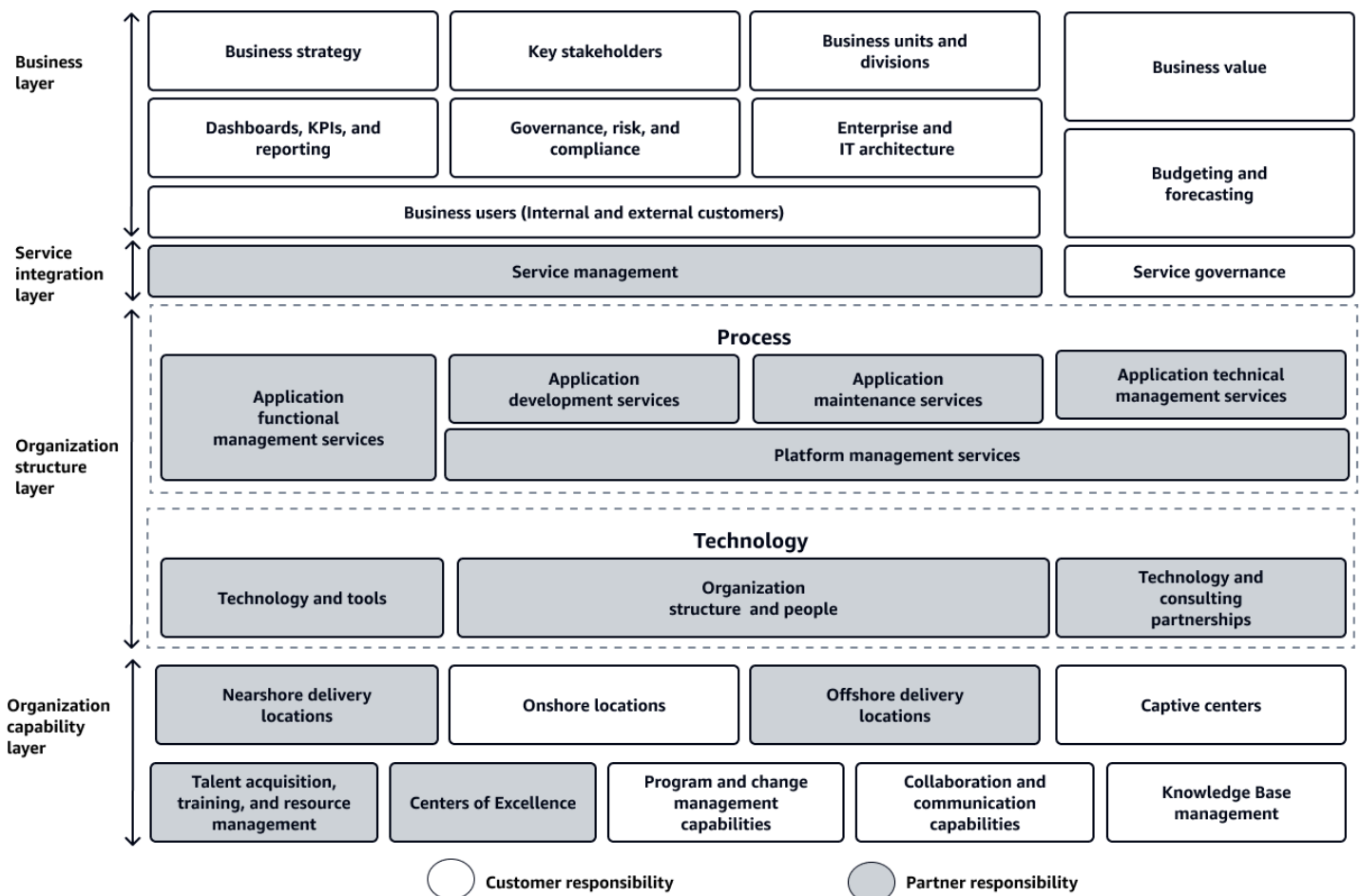
ADM 操作模型概觀

從典型 IT 操作模型概念轉換到 ADM 的特定內容需要了解這些原則如何適用於軟體開發和維護程序。ADM 操作模型提供全方位的架構，可管理從規劃到開發到維護的整個應用程式生命週期。它有助於實現業務目標和 IT 執行之間的成功一致性。

建立 ADM 操作模型通常是諮詢和技術合作夥伴交付的客戶（業務和內部 IT）和/或合作夥伴（應用程式受管服務 (AMS)）之間共同責任。這種協作方法利用多樣化的專業知識，並符合組織的特定需求和技術環境。

如下圖所示，ADM 模型由扮演關鍵角色的互連層組成：

- **業務層** – 此頂層可讓 ADM 活動與組織的策略目標保持一致。在這裡，領導者會定義業務策略、塑造企業架構，並建立控管機制。隨著生成式 AI 整合變得越來越普遍，此層變得越來越動態。它有助於快速且持續地在業務目標和開發活動之間保持一致。
- **服務整合層** – 此操作關係可彌補業務需求和技術實作之間的差距。隨著組織引進生成式 AI，此層協調人類團隊和 AI 系統之間的複雜互動，以提供無縫的服務。
- **組織結構層** – 此層著重於人員、程序和技術，並在 AI 整合期間經歷重大變更。角色將演進，團隊將重新構想程序，而技術堆疊將擴展以包含 AI 工具。此層可推動組織生成式 AI 轉型的實際實作。
- **組織能力層** – 此基礎層涉及全球資源的策略分佈，以及 AI 擴增 ADM 所需的基本技能和專業知識的培養。隨著 AI 整合的進展，此層在開發新能力、建立卓越中心 (COE) 和培養持續學習文化方面扮演重要角色。



當組織準備將生成式 AI 整合至其 ADM 實務時，他們可以視需要重塑此模型的每一層。組織可以重新構想 SDLC 程序、重新定義角色和重新校正技術堆疊，以完全受益於生成式 AI。

ADM 操作模型的真正功能在於其轉換和管理變更的能力。此轉型需要所有利益相關者之間的緊密合作，以協助確保 AI 擴增 ADM 實務的一致性和有效實作。

如需每個 layer 的詳細資訊，請參閱下列各節：

- [ADM 操作模型中的商務層元素](#)
- [ADM 操作模型中的服務整合層元素](#)
- [ADM 操作模型中的組織結構層元素](#)
- [ADM 操作模型中的組織功能層元素](#)

ADM 操作模型中的商務層元素

客戶負責與下列元素相關的活動：

- 商業策略
 - 改善客戶體驗並推動關鍵業務成果
 - 將核心系統現代化，以實現高業務影響
 - 增強敏捷性和創新功能
- 業務單位和支援職能（地理區域和國家/地區）
 - LOBs
 - Marketing
 - 人力資源
 - 採購
 - 法律聲明
 - 資訊技術 (IT)
- 儀表板、KPIs和報告
 - 服務效能報告
 - 服務水準協議 (SLA) 和操作水準協議 (OLA) 監控和報告
 - 業務效能報告
- 控管、風險和合規
 - 指導委員會和每季審查
 - 風險評估和管理
 - 稽核、合規和法規報告

- 企業和 IT 架構
 - 符合業務的 IT 策略
 - 架構和設計原則
 - 技術標準和政策
- 預算和預測
 - 預算規劃和控制
 - 財務效能管理
 - 需求預測和規劃
- 商業價值
 - 改善彈性
 - 提高生產力
 - 提高業務敏捷性
 - 新功能版本

ADM 操作模型中的服務整合層元素

此層包含下列服務管理（諮詢和技術合作夥伴的責任）和服務控管（客戶的責任）的關鍵領域：

- 服務管理包括 IT 服務的交付，包括服務台、事件和問題管理、變更管理和服務層級管理。採用 AI 技術的自動化和智慧型支援功能可增強服務品質和效率。
- 服務控管著重於監督和控制機制，包括服務驗證、可用性管理、容量規劃和組態管理。透過有效的服務控管，服務符合業務目標，同時維持合規和效能標準。

ADM 操作模型中的組織結構層元素

組織層著重於人員、程序和技術。

合作夥伴負責與人員元素相關的活動。在某些情況下，客戶擁有共同來源的參與模型，導致對下列項目負有共同責任：

- 組織結構和人員角色
 - 產品管理 – 專案擁有者和業務分析師
 - 專案管理 – 專案經理、Scrum Master 和 Agile Coach
 - 架構和設計 – 解決方案架構師、技術主管和使用者體驗 (UX) 設計人員

- 開發 – 軟體開發人員和使用者介面 (UI) 設計工具
- 品質保證 – 測試負責人、品質保證 (QA) 測試人員和效能工程師
- 操作 – DevOps 工程師和發行管理員
- 支援和維護 – 支援工程師和技術作者
- 主題專家 (SME) – 安全主題專家 (SMEs)、整合 SME 和領域特定的 SMEs

合作夥伴負責與下列程序元素相關的活動：

- 應用程式功能管理服務
 - 業務流程管理
 - 資訊和資料管理
 - 功能管理
- 應用程式開發服務
 - 專案和需求管理
 - Architecture
 - 設計和開發
 - 測試和品質保證 (QA)
- 應用程式維護服務 (操作)
 - 服務支援管理 (ITSM)
 - 服務請求管理
 - 更新和修補程式管理
 - 服務改進
- 應用程式技術管理服務
 - 應用程式基本知識支援 (第 1 級)
 - 中介軟體管理
 - 資料庫管理
 - 服務改進
- 平台管理服務
 - 受管登陸區域
 - 受管作業系統 (OS)
- 資料庫

- 可觀測性
- 安全
- 網路
- 備份
- 整合
- 雲端財務
- 其他服務

合作夥伴負責下列技術元素的實作和管理相關活動：

- 技術和工具
 - 包括雲端、虛擬化、容器、作業系統、資料庫和其他管理工具
 - 開發人員工具和整合式開發環境 (IDEs)
 - 持續整合和持續開發 (CI/CD) 工具
 - 錯誤修正和 IT 服務管理 (ITSM) 工具
- 技術和諮詢合作夥伴關係
 - Hyperscaler (AWS 及其他)
 - 技術 ISVs
 - IT/服務台供應商
 - 基礎設施委外 (網路、資料中心、安全和工作場所服務)

ADM 操作模型中的組織功能層元素

一般而言，客戶必須負責對下列活動做出關鍵決策：

- 程式設計和變更管理功能
 - 產品組合管理
 - 功能和待處理項目管理
 - 組織變更管理
- 協同合作和通訊功能
 - 生產力工具

- 通訊工具
- 知識庫管理
 - 市場研究
 - 客戶意見回饋和問題解決方案
 - 商業和網域知識
- 境內據點，例如公司辦公室、區域辦公室或遠端站點
- 網頁擷取中心

一或多個諮詢合作夥伴負責實作和管理與下列元素相關的活動：

- 近岸交付位置
- 海外交付位置
- 人才招聘、培訓和認證，以及資源管理
- 卓越中心
 - 創新
 - 技術評估和概念驗證 (POC)
 - POVs、最佳實務、標準和政策

將生成式 AI 整合到 ADM 實務中

生成式 AI 會將 ADM 實務重新塑造為操作模型的所有層。這種轉型技術可以將組織的重點從成本管理轉移到價值建立和創新，從而實現更敏捷和以客戶為中心的方法。

本節概述生成式 AI 如何重塑 ADM 操作模型的下列每個層：

- [商務層](#)
- [服務整合層](#)
- [組織結構層](#)
- [組織功能層](#)

對於每個 layer，其目前狀態和 AI 驅動的未來狀態的概觀提供了生成式 AI 整合的轉型潛力的洞見。此外，下列各節可協助您導覽 ADM 實務中的 AI 驅動演進：

- [整合挑戰和緩解策略](#)
- [動作區域和建議](#)

了解這些變更可協助您有效地利用生成式 AI 來增強組織的軟體開發和維護功能。

ADM 操作模型的業務層

商業層構成 ADM 操作模型的策略基礎。生成式 AI 正在轉型業務策略、利益相關者角色和關鍵領域，例如企業架構、報告、管理和預算。

策略和主要利益相關者

ADM 操作模型包括內部和外部利益相關者，專注於使業務策略和目標與組織操作和成果保持一致。傳統上，這些利益相關者優先考慮應用程式可靠性、發行速度、營運效率、降低成本和應用程式合理化。

從傳統方法轉移到 AI 增強型程序時，利益相關者角色和優先順序會發生下列重要變更：

- 策略重點 – 從成本管理轉移到價值建立和創新。
- 協作決策 - AI 驅動的洞察有助於跨職能策略。
- 敏捷回應能力 – 更快速地適應市場變化和使用者需求。
- 以客戶為中心的方法 – 增強對使用者體驗和滿意度的關注。
- 持續學習 – 強調 AI 讀寫能力和持續技能開發。

這些變更會在業務和服務整合層的各个層面漣漪，影響下列關鍵領域：

- 企業和 IT 架構
- 儀表板和報告
- 控管、風險和合規
- 預算和預測

企業和 IT 架構

下表針對與企業和 IT 架構相關的關鍵問題，提供目前狀態和對應的未來狀態與生成式 AI。

目前狀態	生成式 AI 的未來狀態
手動建立和更新架構文件	自動化架構文件和檢閱
架構變更的靜態影響分析	架構變更的即時影響分析
修正不常更新的藍圖	因應市場變化的自適應藍圖
架構概念的技術術語密集溝通	適用於架構概念的 AI 驅動自然語言界面

儀表板和報告

下表針對與儀表板和報告相關的關鍵問題，提供目前狀態和對應的未來狀態與生成式 AI。

目前狀態	生成式 AI 的未來狀態
具有一般洞見的靜態儀表板	具有使用者特定洞見的即時自適應儀表板
被動問題管理	主動解決問題的預測分析
資料存取的技術查詢語言	非技術利益相關者的自然語言查詢
手動報告產生和金鑰效能指標 (KPI) 追蹤	自動化報告產生和智慧 KPI 建議

控管、風險和合規

下表針對與控管、風險和合規相關的關鍵問題，提供目前狀態和對應的未來狀態與生成式 AI。

目前狀態	生成式 AI 的未來狀態
手動政策檢查和合規稽核	自動化政策檢查和合規監控
根據歷史資料的定期風險評估	具有早期警告和緩解策略的智慧型風險評估
靜態合規文件	動態合規文件產生和更新

預算和預測

下表提供目前狀態和對應的未來狀態，以及與預算和預測相關的關鍵問題生成式 AI。

目前狀態	具有生成式 AI 的未來狀態
歷史資料型手動成本建模	根據歷史資料進行預測性成本建模
定期資源配置調整	即時動態資源配置
由於時間限制，案例規劃有限	預算評估的自動化案例規劃
主觀專案優先順序	符合業務目標的智慧型專案優先順序

ADM 操作模型的服務整合層

服務整合層可做為業務需求和技術執行之間的關鍵橋樑，協調跨 IT 服務的互動。AI 與這一層的整合帶來了服務管理和服務控管方面的變更。

服務管理

下表針對與服務管理相關的關鍵問題，提供目前狀態和對應的未來狀態與生成式 AI。

目前狀態	生成式 AI 的未來狀態
使用內部知識庫搜尋和手動建立的標準操作程序 (SOPs 自助	採用 AI 技術的自助式代理程式，使用企業儲存庫產生動態 SOPs
適用於標準服務請求的自助式工具，例如存取資料和軟體安裝	使用 AI 代理程式工作流程的自動化服務請求

人類客服人員回應使用者查詢	採用 AI 技術的聊天機器人，提供即時的內容感知回應
語言和通訊管道選項有限	跨聊天、語音、SMS 和虛擬助理的多語言、多管道支援
被動問題管理	採用 AI 技術的服務台，可預測常見問題，並在使用者遇到問題之前主動建議解決方案

服務控管

下表針對與服務控管相關的關鍵問題，提供目前狀態和對應的未來狀態與生成式 AI。

目前狀態	生成式 AI 的未來狀態
服務水準協議 (SLA) 管理的被動方法	預測性服務層級管理，以預測潛在的 SLA 違規
手動可用性管理	AI 增強的可用性管理，持續提供服務
靜態容量和效能管理	最佳化資源配置的智慧型容量和效能管理
手動服務驗證和測試	自動化服務驗證和測試
定期組態管理資料庫 (CMDB) 更新	適用於即時 CMDB 更新的 AI 驅動組態管理

先前針對**業務層**和服務整合層使用生成式 AI 的未來狀態範例只是一個開始。隨著 AI 技術的演進，期望出現更創新的解決方案。這些進展有助於增強主動、高效和自動化的 IT 服務管理和管控。

使用這些範例做為組織生成式 AI 轉型方法的起點。請考慮這些範例以及您的 ADM 操作模型變更。持續評估符合您組織需求和目標的新 AI 應用程式。這種前瞻思維方法可協助您保持 IT 服務管理 (ITSM) 創新的前沿。

ADM 操作模型的組織結構層

組織結構層包含人員、程序 and 技術。當組織在 ADM 操作模型中引入生成式 AI 時，就會發生最可見和深入的變更。角色演進、組織重新構想程序，以及技術堆疊擴展以包含生成式 AI 工具。

本節提供在組織 ADM 轉型中實際實作生成式 AI 的洞見，涵蓋組織結構、個別角色和核心程序的變更。透過接受這些策略轉移，您可以讓組織有效地在 ADM 操作模型中整合生成式 AI。此轉型可以改善

開發速度、軟體品質和創新容量，進而提升您的競爭優勢。實際影響會根據組織的特定內容和實作而有所不同。

平台管理服務、技術和工具，以及合作夥伴關係

平台管理服務為應用程式團隊提供一組核心共用功能和標準化服務，包括：

- 編碼參考架構和設計模式
- 部署核准架構和組態的自助式機制
- 標準化的開發、可觀測性和操作工具
- 支援設定環境、持續整合和持續部署 (CI/CD) 管道，以及管理程序
- 集中式控管和安全標準

一般而言，平台工程和雲端營運團隊會管理這些服務、協作以支援應用程式團隊並推動持續改進。

生成式 AI 正在以下列方式轉換平台管理服務：

- 適用於架構建議的 AI 助理會根據專案需求、建議的設計模式和組織標準，建議最佳的參考架構。
- 智慧型自助式佈建使用 AI 來自動化和最佳化部署處理複雜工作流程的資源和服務。
- 採用 AI 技術的可觀測性可提供更深入的洞見，並自動化整個平台的異常偵測。
- AIOps 代理程式會使用核准的標準操作程序 (SOPs) 來處理多個自動化修復工作流程。
- 自動化合規檢查會持續驗證並使用 AI 強制執行控管和安全標準。

這些採用 AI 技術的增強功能可讓基礎設施團隊專注於解決複雜的耗時問題，並改善應用程式的可靠性，進而提升平台管理的效率和有效性。

將生成式 AI 功能整合到受管服務合作夥伴的現有平台產品中。透過此策略，您可以實現下列優點：

- 利用進階 AI 技術，並利用合作夥伴的專業知識和經過驗證的程序。
- 使用整合式 AI 功能增強您的平台工程和雲端操作。
- 在提升 AI 功能的同時，維持已建立的受管服務合作夥伴關係的優勢。

組織結構和角色

生成式 AI 整合需要重新構想 ADM 組織結構。調整組織結構中關鍵角色的責任至關重要。這些 AI 驅動的變更可協助您的團隊更有效率地工作，並提供更高的價值。

組織結構取決於幾個因素：

- 業務開發規模 – 範例包括應用程式的範圍和複雜性，例如交易系統、藥物探索和企業資源規劃 (ERP)。
- 特定客戶需求 – 範例包括支付卡產業資料安全標準 (PCI DSS) 的支付系統合規，以及製藥產業的良好實務 (GxP) 合規。
- 使用的方法 – 範例包括敏捷和瀑布方法。

有些角色會根據專案需求合併或擴展。涉及進階技術或嚴格合規需求的專案通常包括特殊角色，例如資料科學家、機器學習 (ML) 專家、進階商業應用程式程式設計 (ABAP) 開發人員和合規主管。

下列各節重點介紹 ADM 中隨著生成式 AI 整合而演進的常見角色。這些角色正在擴展和調整以使用 AI 功能，這可以增強其在組織內的價值和影響。這種演變代表跨許多角色的技能開發和職業成長的機會。下列層面提供每個角色在與生成式 AI 整合時如何發展的洞見：

- 目前重點 – 角色中的人員目前執行的主要任務
- AI 驅動的轉移 – 生成式 AI 可以併入角色的方式
- 主要優點 – 將生成式 AI 整合到角色中所獲得的優點
- 關鍵考量 – 考量角色的 AI 驅動轉移時的考量
- 關鍵步驟 – 角色中的人員可以採取的主要步驟，以協助他們適應 AI

此全方位檢視可協助您了解目前狀態、變革方向，以及成功導覽每個角色 AI 驅動轉換所需的步驟。您可以深入了解 AI 如何增強現有角色，以及如何為這些進展準備組織結構。

產品擁有者或業務分析師

下表提供產品擁有者或業務分析師角色如何適應以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 需求收集 • 特徵優先順序 • 利益相關者通訊
AI 驅動的輪班	<p>將 AI 用於：</p> <ul style="list-style-type: none"> • 資料驅動型決策流程和加速市場洞察

<p>主要優點</p>	<ul style="list-style-type: none"> • 建立業務需求文件 (BRD)，根據客戶意見回饋和需求排定功能的優先順序 • 更快速的需求收集和分析 • 改善功能與市場需求的一致性 • 更全面的使用者案例和使用案例
<p>關鍵考量</p>	<ul style="list-style-type: none"> • 確保 AI 理解複雜的商業環境 • 維持有意義的利益相關者關係
<p>關鍵步驟</p>	<ul style="list-style-type: none"> • 實作 AI 驅動的市場分析和需求工具。 • 開發快速工程技能，以有效進行 AI 互動。 • 建立利益相關者程序，以驗證 AI 產生的洞見

專案經理

下表提供專案經理角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
<p>目前焦點</p>	<ul style="list-style-type: none"> • 活動規劃 • 資源配置 • 風險管理
<p>AI 驅動的輪班</p>	<ul style="list-style-type: none"> • 採用 AI 進行增強型預測規劃和即時專案智慧。
<p>主要優點</p>	<ul style="list-style-type: none"> • 改善資源配置準確性 • 增強風險識別和緩解 • 即時專案運作狀態監控和預測分析
<p>關鍵考量</p>	<ul style="list-style-type: none"> • 平衡 AI 建議與人類判斷 • 確保團隊採用 AI 驅動的方法
<p>關鍵步驟</p>	<ul style="list-style-type: none"> • 整合 AI 驅動的專案規劃和風險評估工具。 • 制定 AI 人類協作決策的通訊協定。

- 在 AI 擴增的專案管理實務中提升技能團隊。

UI/UX 設計工具

下表提供使用者介面/使用者體驗 (UI/UX) 設計工具角色如何適應以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 建立使用者介面設計和原型 • 執行使用者研究和可用性測試 • 確保跨應用程式的最佳使用者體驗
AI 驅動的輪班	<ul style="list-style-type: none"> • 使用 AI 進行快速設計反覆運算、資料驅動型使用者洞察和自動化可用性測試。
主要優點	<ul style="list-style-type: none"> • 更快速產生 UI 設計替代方案 • 增強的使用者研究分析和角色建立 • 自動化可用性測試和意見回饋分析
關鍵考量	<ul style="list-style-type: none"> • 平衡 AI 產生的設計與品牌指導方針和使用者需求 • 在 AI 輔助設計程序中保持創造力和創新
關鍵步驟	<ul style="list-style-type: none"> • 整合 AI 驅動的專案規劃和風險評估工具。 • 制定 AI 人類協作決策的通訊協定和程序。 • 在 AI 擴增的專案管理實務中提升技能團隊。

全堆疊開發人員

下表提供完整堆疊開發人員角色如何適應以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 建立使用者介面設計和原型 • 執行使用者研究和可用性測試 • 確保跨應用程式的最佳使用者體驗

AI 驅動的輪班	<ul style="list-style-type: none"> • 接受 AI 以獲得全方位的開發協助和最佳化。
主要優點	<ul style="list-style-type: none"> • 加速全堆疊程式碼的產生和最佳化 • AI 驅動的 API 設計和整合 • 跨堆疊的自動化效能調校
關鍵考量	<ul style="list-style-type: none"> • 與 AI 工具一起維持多種技術的熟練度 • 確保 AI 和手動開發元件之間的一致性和整合
關鍵步驟	<ul style="list-style-type: none"> • 在整個堆疊中開發 AI 輔助開發的專業知識。 • 建立整合 AI 產生和手動程式碼的程序和準則。 • 在全堆疊開發中為新興 AI 工具實作持續學習計劃。

解決方案架構師

下表提供解決方案架構師角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 設計全方位的企業解決方案 • 使技術解決方案與業務目標保持一致 • 確保跨系統的整合和互通性 • 建立詳細的設計文件
AI 驅動的輪班	<ul style="list-style-type: none"> • 使用 AI 進行快速解決方案原型、資料驅動型架構決策、自動化整合分析和設計文件產生。
主要優點	<ul style="list-style-type: none"> • 更快速產生和評估解決方案替代方案 • 增強技術解決方案與業務目標的一致性 • 改善系統整合和互通性的評估 • 加速建立完整的設計文件
關鍵考量	<ul style="list-style-type: none"> • 確保 AI 產生的解決方案滿足複雜的業務需求

關鍵步驟

- 在 AI 擴增設計程序中維護企業架構的整體檢視
- 驗證 AI 產生之設計文件的準確性和完整性
- 開發 AI 解決方案設計工具和方法的專業知識。
- 建立程序來驗證 AI 產生的解決方案提案是否符合業務需求。
- 實作 AI 驅動的工具，以進行持續的解決方案最佳化和整合評估。
- 採用 AI 輔助文件工具來建立和維護設計文件。

軟體開發人員

下表提供軟體開發人員角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 程式碼編寫 • 除錯 • Maintenance (維護)
AI 驅動的輪班	<ul style="list-style-type: none"> • 接受 AI 作為編碼配套，以提高生產力和品質。
主要優點	<ul style="list-style-type: none"> • 加速程式碼產生和完成 • 改善程式碼品質和一致性 • 更快速的錯誤偵測和解決
關鍵考量	<ul style="list-style-type: none"> • 在 AI 產生的程式碼中維護程式碼可讀性和效能 • 平衡 AI 工具與核心程式設計技能的依賴性
關鍵步驟	<ul style="list-style-type: none"> • 改善 AI 輔助編碼和配對程式設計技術的使用。

- 建立檢閱和最佳化 AI 產生程式碼的準則。
- 為新興 AI 開發工具實作持續學習計劃。

測試工程師

下表提供測試工程師角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 測試案例設計 • 瑕疵識別 • 品質保證
AI 驅動的輪班	<ul style="list-style-type: none"> • 實作 AI 以實現全方位的自動化測試策略。
主要優點	<ul style="list-style-type: none"> • 提高測試案例產生和執行的自動化 • 改善測試資料品質和涵蓋範圍 • 透過預測瑕疵分析進行早期問題偵測
關鍵考量	<ul style="list-style-type: none"> • 確保 AI 產生的測試案例之外的全面涵蓋 • 使用探索性方法平衡自動化測試
關鍵步驟	<ul style="list-style-type: none"> • 開發 AI 測試策略設計和資料建模的技能。 • 建立持續改進 AI 測試模型的程序。 • 實作 AI 擴增的探索性測試程序和技術。

版本管理員

下表提供發行管理員角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 規劃和協調軟體版本 • 管理發行排程和相依性 • 確保順暢的部署和發行後穩定性

AI 驅動的輪班	<ul style="list-style-type: none"> 採用 AI 進行智慧版本規劃、自動化部署和預測穩定性管理。
主要優點	<ul style="list-style-type: none"> AI 驅動的版本規劃和風險評估 自動化部署和復原策略 預測發佈後監控和問題偵測
關鍵考量	<ul style="list-style-type: none"> 在 AI 建議與業務優先順序和限制之間取得平衡 在自動化部署案例中維持控制和監督
關鍵步驟	<ul style="list-style-type: none"> 開發採用 AI 技術的發行管理工具和預測分析的技能。 建立人工驗證 AI 產生的發行計畫的程序。 實作 AI 驅動的發行後監控和快速回應標準操作程序 (SOP)。

技術主管

下表提供技術領導角色如何調整以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> 監督應用程式開發和操作程序 確保開發團隊與營運需求之間的一致性 管理從開發到生產的應用程式生命週期 推動開發和營運效率的持續改進
AI 驅動的輪班	<ul style="list-style-type: none"> 利用 AI 進行增強型應用程式生命週期管理、自動化操作分析和預測性資源最佳化。
主要優點	<ul style="list-style-type: none"> 改善開發和營運團隊之間的協調 增強型應用程式效能監控和預測性維護 根據操作分析的自動化資源配置和擴展 經常變更的次數

關鍵考量	<ul style="list-style-type: none"> • 加速問題解決並縮短停機時間 • 在關鍵操作中平衡 AI 驅動的自動化與人工監督 • 確保 AI 工具在整個應用程式生命週期的無縫整合 • 管理 AI 擴增 DevOps 實務的文化轉移
關鍵步驟	<ul style="list-style-type: none"> • 開發 AI 驅動的應用程式生命週期管理工具的專業知識。 • 建立將 AI 洞察整合至開發和營運決策的程序。 • 實作 AI 驅動的監控和預測性維護系統。 • 建立訓練計畫，以在 AI 擴增的 DevOps 實務中提升團隊技能。

DevOps 工程師

下表提供 DevOps 工程師角色如何適應以使用生成式 AI 功能的概觀。

角色的層面	Description
目前焦點	<ul style="list-style-type: none"> • 實作和維護持續整合和持續部署 (CI/CD) 管道 • 自動化基礎設施佈建和管理 • 確保開發和操作之間的無縫整合
AI 驅動的輪班	<ul style="list-style-type: none"> • 使用 AI 增強自動化、預測分析和智慧型基礎設施管理。
主要優點	<ul style="list-style-type: none"> • 加速部署週期 • 改善系統可靠性和效能 • 主動問題偵測和解決
關鍵考量	<ul style="list-style-type: none"> • 將 AI 工具與現有的 DevOps 程序整合 • 在自動化與必要的人工監督之間取得平衡

關鍵步驟

- 實作採用 AI 技術的 CI/CD 管道最佳化。
- 採用 AI 輔助基礎設施即程式碼 (IaC) 產生工具。
- 在 AIOps 中開發用於預測性維護和自動化修復的技能。

支援工程師

下表提供支援工程師角色如何調整以使用生成式 AI 功能的概觀。

角色的層面**Description****目前焦點**

- 解決使用者問題和事件
- 維護系統的可靠性
- 為最終 使用者提供技術協助

AI 驅動的輪班

- 採用 AI 進行智慧問題分類、自動化問題解決和預測支援。

主要優點

- 問題解決時間更快
- 改善第一次通話的解析率
- 主動識別潛在的系統問題

關鍵考量

- 確保 AI 系統準確了解並分類複雜的技術問題
- 在客戶互動中保持人工接觸

關鍵步驟

- 實作 AI 支援的知識庫，以更快速解決問題。
- 採用 AI 驅動的票證分類和路由系統。
- 開發與 AI 聊天機器人和虛擬助理合作以獲得客戶支援的程序和技能。

ADM 操作模型的組織功能層

傳統上，組織功能，例如知識管理、溝通和協作，以及程式或變更管理工具，都缺乏 AI 特定的焦點。當您將生成式 AI 整合到您的 ADM 實務中時，您的組織功能必須演進。本節概述轉型和策略的關鍵領

域，以有效利用您的 AMS 合作夥伴。本節也探討 AI 如何推動全球資源分發、培養基本技能、發展新能力、建立 AI CoEs，以及培養持續學習文化。

策略合作夥伴和人才開發 – 若要建立策略合作夥伴關係並開發 AI 整合的人才，請專注於這些關鍵計畫：

- 實作全面的 AI 訓練計畫。
- 建立 AI 卓越中心 (COEs)。
- 使用 AI 改善職業規劃、招聘、訓練和資源最佳化。
- 實作位置特定的 AI 採用變更管理計畫。
- 使用 AI 更有效率地開發最佳實務、標準和觀點 (POVs)。
- 執行與 IT 架構藍圖一致的技术評估和概念驗證 (POCs)。

操作模型重新設計 – AI 的整合需要重新設計操作模型，包括下列變更：

- 重新定義角色以整合 AI 擴增開發。
- 將 AI 驅動的策略任務指派給現場團隊，以與關鍵決策者保持密切的協作。
- 為 AI 產生的程式碼開發新的 QA 程序。

增強協同合作和知識管理 – 考慮透過這些方法來增強協同合作和知識管理：

- 實作 AI 支援的協作工具，以減少時區相依性。
- 利用 AI 更有效地編目和編製企業知識的索引。
- 使用來自客戶意見回饋、問題解決和產業趨勢的 AI 驅動洞察，以加速市場研究和業務需求分析。

控管與合規 – 若要在將 AI 整合到操作模型時協助確保適當的控管與合規，請考慮實作下列措施：

- 建立具有特定位置合規要求的全域 AI 控管架構。
- 解決 AI 產生資產的 IP 擁有權，並降低侵權風險。

基礎設施和工具標準化 – 整個組織的基礎設施和工具標準化的有效 AI 整合涉及以下步驟：

- 投資於可從所有位置存取的雲端 AI 擴增平台。
- 全球標準化 AI 工具和環境。

效能指標和參與模型調整 – 調整 AI 驅動程序的效能指標和參與模型包括下列關鍵動作：

- 開發考慮 AI 貢獻的新 KPIs。
- 實作 AI 輔助專案估算工具。
- 考慮靈活的參與模式，包括以成果為基礎的定價。
- 定義 AI 資產的消耗型定價模型，涵蓋授權、基礎設施和受管服務工作。

計劃和變更管理擴增 – 若要強化計劃和變更管理，請考慮以下策略：

- 使用 AI 增強內部人才、諮詢和 AMS 合作夥伴之間共同來源模型。
- 改善新計畫的知識收集、方法精簡和重複使用體驗。

透過專注於這些領域，您可以有效地將生成式 AI 整合到全球交付位置和組織功能。此方法有助於加速 ADM 操作模型的轉換。它可提高決策速度並增強業務成果的交付，同時平衡每個位置的優勢並解決 AI 整合的挑戰。

整合挑戰和緩解策略

雖然將生成式 AI 整合到 ADM 的優勢很大，但存在挑戰。了解這些障礙對於制定有效的緩解策略至關重要。下表為將生成式 AI 整合到 ADM 時可能受到影響的領域提供關鍵挑戰和對應的緩解措施。

區域圖	主要挑戰	風險緩解策略
資料管理	資料品質和整合挑戰	確保跨各種系統和程序的一致、高品質資料。
治理和道德	AI 控管和道德	建立 AI 使用和決策的明確準則。
人力調整	文化適應	為 AI 擴增角色準備人力。
程序整合	與現有程序整合	將 AI 無縫整合到已建立的工作流程中。
信任、可靠性和人為監督	驗證 AI 產生的洞見和建議，以獲得一致的準確性	保持適當的人工控制，同時利用 AI 自動化。

技術複雜性	缺乏技能和經驗	管理 AI 增強型系統的複雜性增加。
安全性與合規	缺乏資料保護和 IP 擁有權準則	在 AI 驅動的環境中維護資料保護和法規遵循。
組織一致性	AI 建議一致性	確保 AI 建議符合組織政策和最佳實務。
平台複雜性	缺乏技能和變革準備	管理 AI 增強型平台和 IT 支援服務的複雜性。
委外挑戰	外包操作中的功能差距	解決受管服務供應商中的 AI 準備度。

動作區域和建議

若要將生成式 AI 成功整合到您的 ADM 操作模型中，請考慮下列動作領域的建議。這些建議可協助您導覽組織的轉型之旅，並克服常見的挑戰。

控管和策略 – 若要建立有效的 AI 控管，並將其與整體業務策略保持一致，請考慮實作這些關鍵動作：

1. 與 AI 擁護者建立跨職能 AI 指導委員會。
2. 制定明確的 AI 控管政策，包括道德使用準則。
3. 持續將 KPIs 和業務目標與 AI 功能保持一致。
4. 在 AI 驅動的合規程序上與監管機構合作。

AI 卓越中心 – 為了最大限度地提高 AI 卓越中心 (COE) 在 ADM 實務中的影響，請專注於這些計畫：

1. 建立並啟動專用 AI COE，以推動採用、確保最佳實務，並跨 ADM 提供指引。
2. 開發全面的 COE 操作程序和概述 AI 相關服務和支援的服務目錄。
3. 透過進階 AI 研究和策略合作夥伴關係，持續擴展 COE 功能。

教育和文化 – 若要支援 AI 採用和整個組織持續學習的文化，請考慮這些動作：

1. 在整個組織中實作全面的 AI 讀寫能力計畫。

2. 培養實驗、學習和適應的文化。
3. 建立訓練計畫，以在 AI 擴增操作中提升平台團隊技能。

技術和程序– 若要將 AI 有效地整合到您的技術堆疊和程序中，請優先考慮這些計畫：

1. 實作適用於架構建議和資源佈建的 AI 驅動工具。
2. 開發 AI 模型以進行預測容量規劃和效能最佳化。
3. 整合採用 AI 技術的可觀測性和異常偵測系統。
4. 建立 AI 輔助的合規檢查和安全監控程序。
5. 跨專案實作標準化的資料收集架構。
6. 開發可同時適應瀑布和敏捷方法的 AI 模型。

資料和安全性 – 若要支援資料品質和安全性工作，請專注於這些動作：

1. 投資資料整合、品質保證和安全程序。
2. 建立意見回饋機制以持續改善 AI 系統。

變更管理 – 若要促進 AI 技術的順利採用，請使用下列變更管理方法：

1. 重新設計利益相關者通訊管道，以進行 AI 增強的協同合作。
2. 實作變更管理計畫，在 AI 產生的洞見中建立信任。

技能開發 – 若要建置必要的 AI 功能，請支援此技能開發計畫：

- 升級資料科學、AI 解譯和 AI 支援工具的技能團隊。

合作夥伴關係 – 若要利用外部專業知識，請考慮這些合作夥伴關係的想法：

1. 利用應用程式受管服務 (AMS) 合作夥伴進行 AI 實作。
2. 考慮跨平台工程服務進行 AI 整合的基礎設施和/或 CloudOps 受管服務合作夥伴。
3. 使用 IT 服務管理合作夥伴來整合 AI 與服務管理和控管服務。

人為監督 – 若要維持適當的人為控制和責任，請實作下列方法：

- 建立人工監督 AI 產生的建議的通訊協定。

接受這些 AI 驅動的變更並有系統地解決挑戰，可協助您建立更靈活、更有效率且創新的 ADM 操作模型。成功的關鍵在於平衡人類專業知識與 AI 功能，使 IT 服務與組織目標緊密保持一致。這種方法可以推動顯著的商業價值、增強組織的競爭優勢，並使組織處於下一個 ADM 時代的領導位置。

建置採用 AI 技術的 ADM 目標操作模型

當您使用生成式 AI 來考慮 ADM 實務時，設計全面的目標操作模型 (TOM) 非常重要。TOM 說明組織操作模型的所需狀態。您組織的 ADM TOM 應使其人員、程序、技術、組織和治理與其策略願景保持一致。

下表列出 TOM 的八個元件。

TOM 元件	元件元素
策略一致性	<ul style="list-style-type: none">• 值驅動程式• 業務目標一致性• AI 藍圖
組織結構	<ul style="list-style-type: none">• AI 卓越中心• 新的 AI 角色• 跨職能團隊
人才和技能	<ul style="list-style-type: none">• 職業路徑• 持續學習• AI 讀寫能力要求• 技能差距分析
治理和道德	<ul style="list-style-type: none">• 法規合規• 資料隱私權架構• AI 道德政策
效能測量	<ul style="list-style-type: none">• 持續監控• 業務影響報告• 回饋迴圈• AI 特定 KPIs
合作夥伴生態系統	<ul style="list-style-type: none">• 合作夥伴評估指標• 資料共用通訊協定• AI 功能需求

技術和工具

- 協同創新
- 資料基礎設施
- AI 工具生態系統
- AI 平台選擇
- 傳統系統整合

Processes

- AI 增強 SDLC
- AI 模型管理
- 控管工作流程

建置 ADM TOM 是影響組織各方面的轉型程序。請仔細考慮每個 ADM 元件及其相互依存性，為採用 AI 技術的 SDLC 建立穩固的基礎。

實作 ADM TOM 應該根據組織的特定需求和內容量身打造。當您實作此模型時，請根據組織的獨特挑戰和機會持續評估和調整模型。

下列各節提供有關 ADM 操作模型中元件的詳細資訊，包括其互動。

策略對齊元件

策略一致性元件定義 AI 驅動的 ADM 的策略目標，使 AI 計劃與業務目標保持一致。此元件會在 ADM 程序中清楚說明 AI 的值，並設定 AI 整合的成功條件。此元件與其他元件互動，如下所示：

- 值驅動因素會影響效能測量元件中的 AI 特定 KPIs。
- 業務目標一致性會通知在組織結構元件中建立新的 AI 角色。
- AI 藍圖會引導 AI 平台選擇技術和工具元件。

組織結構元件

組織結構元件可處理 ADM 組織的設計，該組織支援具有新角色的 AI 擴增開發。此元件建立 AI 卓越中心 (COE)，並發展 AI 整合的現有角色。

- AI COE 支援在人才和技能元件中持續學習。
- 新的 AI 角色會影響合作夥伴生態系統元件中的新 AI 功能需求。
- 跨功能團隊可在程序元件中靈活整合 AI 增強的 SDLC。

人才和技能元件

人才和技能元件可識別 ADM 角色和人員所需的 AI 技能和能力。此元件定義 AI 讀寫能力需求，並建立以 AI 為重心的職業道路。

- 職業路徑與組織結構元件中的新 AI 角色保持一致。
- AI 讀寫能力要求支援控管和道德元件中的 AI 道德政策。
- 技能差距分析會通知技術和工具元件中的 AI 工具生態系統。

治理和道德元件

治理和道德元件為 ADM 中的 AI 使用建立道德架構，包括政策和審查委員會。此元件定義 AI 驅動 ADM 實務的資料隱私權和安全性需求。

- 法規合規會影響策略一致性元件中的價值驅動因素。
- 資料隱私權架構會影響合作夥伴生態系統元件中的資料共用通訊協定。
- AI 道德政策會在程序元件中引導 AI 模型管理。

效能測量元件

效能測量元件設計了具有 AI 特定 KPIs 的新架構，用於 ADM 效能測量。此元件概述測量、報告和最佳化 ADM 中 AI 影響的方法。

- 業務影響報告會影響合作夥伴生態系統元件中的合作夥伴評估指標。
- 意見回饋迴圈支援在人才和技能元件中持續學習。
- AI 特定的 KPIs 會通知策略一致性元件中的業務目標一致性。

合作夥伴生態系統元件

合作夥伴生態系統元件定義了對 AMS 合作夥伴和協作程序中 AI 功能的期望。此元件會建立合作夥伴互動的資料共用和模型擁有權原則。

- 合作夥伴評估指標會在效能測量元件中通知 AI 特定的 KPIs。
- AI 功能需求會影響人才和技能元件中的技能差距分析。
- 協作創新支援技術和工具元件中的 AI 工具生態系統。

技術和工具元件

技術和工具元件指定 AI 技術和工具，以支援轉換後的 ADM 程序。此元件可識別 AI 驅動 ADM 的整合點和資料需求。

- 資料基礎設施支援效能測量元件中的業務影響報告。
- 舊版系統整合會影響程序元件中的 AI 增強 SDLC。
- AI 平台選擇會影響合作夥伴生態系統元件中的協作創新。

程序元件

程序元件重新設計 SDLC 以整合 AI，透過 AI 功能增強每個階段。此元件開發開發中 AI 模型管理和控管的新程序。

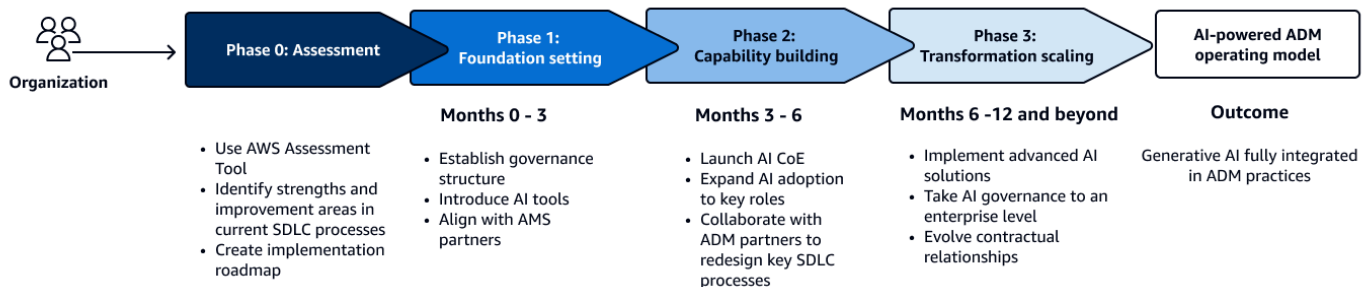
- AI 增強 SDLC 會影響效能測量元件中的持續監控。
- AI 模型管理與技術和工具元件中的資料基礎設施相關。
- 控管工作流程支援控管和道德元件中的資料隱私權架構。

實作採用 AI 技術的 ADM 目標操作模型

使用結構化的分階段方法實作生成式 AI 應用程式開發和維護 (ADM) 目標操作模型 (TOM)。下列方法在快速獲勝與長期轉型變更之間取得平衡，同時將對目前操作的干擾降至最低。每個階段都會處理 TOM 的特定元件，強調它們在整個實作過程中的相互依存性和演變。

如下圖所示，實作策略包含 12 個月內從基本到進階複雜性的階段：

- 階段 1：基礎設定 – 此階段發生在第 1–3 個月。它建立基本的控管結構，並引進必要的 AI 工具，同時實現快速成功。
- 階段 2：能力建置 – 此階段發生在第 3-6 個月。它擴展了 AI 採用，並解決了中等複雜性的程序。啟動 AI COE、將 AI 採用擴展至專案管理和操作角色，並與您的 ADM 合作夥伴合作，使用生成式 AI 重新設計關鍵 SDLC 程序。
- 階段 3：轉換擴展 – 此階段發生在第 6 到 12 個月（及之後）。它實作了進階解決方案，並解決了更高的複雜性挑戰。例如，針對架構設計、全堆疊開發和安全監控實作進階 AI 解決方案。將您的 AI 治理成熟到企業層級，並發展您與 ADM 合作夥伴的合約關係，以反映新的 AI 驅動現實。



Note

開始實作之前，請執行採用 AI 技術的 SDLC 整備度評估，以建立組織目前 SDLC 功能的基準，並識別需要改善的關鍵領域。如需詳細資訊，請參閱[後續步驟](#)。

實際時間表可能會因組織內容、實作方法和實作大小和規模等其他因素而有所不同。有些組織可能會達到較短或較長的時間範圍，取決於其特定情況和成熟程度。

透過進行這些階段，您可以有系統地轉換組織的 ADM 實務，使用 AI 來推動創新、效率和競爭優勢。如需在組織中使用分階段方法的詳細資訊，請參閱[實作 AI 驅動 ADM TOM 的藍圖](#)，以及[所有實作階段的最佳實務](#)。

組織可以透過此轉型之旅來增強其內部功能。此旅程也需要持續調整，並與所有利益相關者進行清晰的溝通。結果是整合的全域 ADM 目標操作模型，可搭配您的諮詢和技術服務供應商進行 AI 驅動的軟體開發和維護。

實作 AI 驅動 ADM TOM 的藍圖

下表提供參考藍圖，使用分階段方法實作 ADM TOM，同時將對目前操作的干擾降至最低。對於每個 ADM 元件，藍圖會描述每個實作階段中發生的相關活動。

ADM 元件	基礎設定：第 1-3 個月	能力建置：第 3-6 個月	轉換擴展：第 6-12 個月及之後
策略一致性	<ul style="list-style-type: none"> 啟用 AI 轉向委員會。 設定具有業務一致性的願景、任務和目標。 開發 AI 技術和工具策略和藍圖。 	<ul style="list-style-type: none"> 持續將 KPIs 和業務目標與 AI 功能保持一致。 在 AI 計畫上保持明確的利益相關者溝通，並產生影響。 檢閱業務成果和投資報酬率。 	<ul style="list-style-type: none"> 持續將 KPIs 和業務目標與 AI 功能保持一致。 在 AI 計畫上保持明確的利益相關者溝通，並產生影響。 檢閱業務成果和投資報酬率。 將 AI 控管與 EA 整合。 與 AMS 合作夥伴建立跨功能 AI 控管。 將全球 AI 工具跨內部和 AMS 合作夥伴團隊進行標準化。
組織結構	<ul style="list-style-type: none"> 識別跨職能 AI 擁護者。 識別 AI 整合的關鍵角色。 	<ul style="list-style-type: none"> 透過專用團隊啟動 AI COE。 	<ul style="list-style-type: none"> 實作 AI 驅動的組織和持續最佳化。

人才和技能

- 實作基本 AI 訓練計畫。
- 針對軟體開發人員和測試工程師等高傾向角色採用 AI 工具。
- 實作進階 AI 訓練計畫。
- 實作角色特定的 AI 訓練計畫。
- 實作角色特定的 AI 訓練計畫。
- 開發以 AI 為重心的職業道路和進展。
- 為境內和海外團隊實作共用訓練計畫。
- 實作角色特定的 AI 訓練計畫。
- 將 AI 採用擴展至產品擁有者、BA、SA 和網域 SMEs。
- 建立 AI 創新激勵計畫。
- 建立機制，在您的組織和 AMS 合作夥伴之間持續分享 AI 知識。

治理和道德

- 制定 AI 道德準則。
- 建立 AI 相關 IP 和資料用量的指導方針。
- 建立風險評估架構。
- 與監管機構合作以符合規範。
- 實作 AI 控管政策和程序。
- 平衡 AI 自動化與人工監督，以確保品質並維持控制。
- 平衡 AI 自動化與人工監督，以確保品質並維持控制。
- 平衡 AI 自動化與人工監督，以確保品質並維持控制。
- 為 AMS 合作夥伴開發 AI 特定的專案和合約範本和 SLAs。
- 持續檢閱和解決 ADM 中 AI 用量部分的資料隱私權和安全問題。

效能測量

- 為 ADM 建立 AI 目標和關鍵成功指標。
- 為大型語言模型 (LLMs) 建立關鍵成功指標。
- 開發 ADM 程序的 AI 特定 KPIs。
- 開發 ADM 合作夥伴效能的 AI 特定 KPIs。
- 實作 AI 成本分配和投資報酬率追蹤。
- 建立 KPIs 並實作 ADM 和 SDLC 效能儀表板。
- 實作 AI 驅動的洞察，以持續改善 ADM 全球交付模型。
- 根據意見回饋和結果持續監控和調整。

合作夥伴生態系統	<ul style="list-style-type: none"> 讓 AMS 合作夥伴參與轉型規劃。 將 AI 整合角色與 AMS 合作夥伴保持一致。 使用 AMS 和 CloudOps 合作夥伴評估 AI 準備程度。 檢閱現有的 AMS 合約以進行 AI 整合。 	<ul style="list-style-type: none"> 與 AMS 和 CloudOps 合作夥伴建立聯合 AI COE。 與 ADM 合作夥伴合作，在 TOM 中整合 AI。 與 AMS 合作夥伴合作，為 ADM 實作進階 AI 解決方案。 	<ul style="list-style-type: none"> 與 AMS 合作夥伴合作，為 ADM 實作進階 AI 解決方案。 使用 AMS 合作夥伴標準化 AI 工具和環境。 定期評估 AI 對 AMS 外包價值主張的影響。 考慮 AI 增強型服務的彈性參與模型和以成果為基礎的定價。
技術和工具	<ul style="list-style-type: none"> 實作 AI 支援的知識庫，以更快速解決問題。 實作採用 AI 技術的協作工具。 採用 AI 輔助編碼和測試工具。 	<ul style="list-style-type: none"> 整合 AI 驅動的專案規劃和風險評估工具。 實作 AI 支援的版本管理和預測性維護。 實作 AI 輔助專案估價器工具。 	<ul style="list-style-type: none"> 實作 AI 驅動的架構決策支援工具。 採用 AI 驅動的全堆疊程式碼產生和最佳化工具。 為所有交付位置實作以雲端為基礎的 AI 擴增平台。
Processes	<ul style="list-style-type: none"> 建立整合 AI 產生和手動程式碼的指導方針。 為採用 AI 的工具建立程序和 SOPs。 建立回饋迴圈以持續改善 LLMs。 	<ul style="list-style-type: none"> 重新設計 ADM 程序以將 AI 納入 TOM。 在陸上、近岸和海上位置之間開發 AI 驅動 SOPs。 	<ul style="list-style-type: none"> 建立 AI 驅動架構決策和產生完整堆疊程式碼的程序。 建立 AI 輔助的合規檢查和安全監控程序。 在採用 AI 技術的 ADM 操作模型上建立程序改進機制。

如需有關包含任務陳述式、目標和策略計劃之 ADM AI 願景架構的資訊，請參閱[附錄 A：適用於 ADM 的 AI 願景範例架構](#)。如需涵蓋三個階段中控管、組織結構、角色、程序和工具的詳細實作檢查清單，請參閱[附錄 B：ADM TOM 的實作檢查清單](#)。

所有實作階段的最佳實務

下列最佳實務對於在所有實作階段都很重要。對於每個最佳實務，會顯示其相關的操作模型元件，指出模型的哪個方面受影響最大：

- 根據意見回饋和結果持續監控和調整方法。（效能測量）
- 與所有利益相關者清楚溝通各種 AI 計畫及其影響。（策略對齊）
- 平衡 AI 自動化與人工監督，以協助確保品質和維護控制。（政府和道德）
- 定期評估 AI 計劃的投資報酬率 (ROI)，並相應地調整策略。（效能測量；策略一致性）
- 解決全球交付模型中 AI 使用量特有的資料隱私權和安全問題。（政府和道德）
- 定期評估 AI 對外包價值主張的影響，並視需要調整參與模型。（合作夥伴生態系統；策略一致性）

後續步驟

此策略文件探討生成式 AI 如何影響應用程式開發和維護 (ADM) 操作模型的每一層。它描述了如何實現潛在利益，例如提高開發速度、減少生產瑕疵並改善客戶滿意度分數。若要開始組織的 AI 驅動型軟體開發生命週期 (SDLC) 旅程，並為新一代 ADM 實作其目標操作模型，請使用下列步驟。

成功的 AI 整合需要平衡 AI 功能與人類專業知識。此平衡可在整個組織的 SDLC 程序和 ADM 實務中推動創新、效率和競爭優勢。遵循這些步驟，您可以將組織置於 AI 擴增軟體開發的最前線。這種方法可推動顯著的商業價值，並增強您在業界的競爭優勢。

步驟 1：執行整備評估

使用 [AWS 評估工具](#) 中的 AI 驅動型軟體開發評估 (AIS DLC - V1.0) 來評估您目前的 SDLC 功能和目前 ADM 操作模型的準備程度。此評估可協助您：

- 識別現有 SDLC 程序和 ADM 實務中的優勢和改進領域。
- 明確指出 AI 可能對您的業務產生最大影響的領域。
- 排定修復活動的優先順序，並建立實作藍圖。

步驟 2：建立基礎功能

若要了解並協助使用生成式 AI 在 SDLC 中建置基礎功能，請參閱 [AWS 使用生成式 AI 在上加速軟體開發生命週期](#)。此策略文件提供 AWS 架構最佳實務，可協助您執行下列任務，以實作您的藍圖：

- 建立 AI 整合的堅實基礎。
- 讓您的程序符合產業最佳實務。
- 讓您的團隊準備好進行 AI 擴增開發。

步驟 3：實作分階段方法

若要實作 ADM 目標操作模型，請參閱 [藍圖](#)，其中包含從初始快速獲勝到完整 AI 整合的所有階段。利用 [範例架構](#) 和 [實作檢查清單](#)。

[早期採用者](#) 的成功案例展示了 AI 在應用程式開發和維護方面的轉型潛力。

Resources

AWS 部落格文章

- [重要的新功能可讓您更輕鬆地使用 Amazon Bedrock 來建置和擴展生成式 AI 應用程式，並實現令人驚豔的結果](#)
- [使用生成式 AI 轉換軟體開發生命週期 \(SDLC\)](#)

AWS 服務 resources

- [Amazon Bedrock 代理程式](#)
- [Amazon Bedrock 代理程式流程](#)
- [Amazon Bedrock 護欄](#)
- [Amazon Bedrock 知識庫](#)
- [Amazon Bedrock 安全與隱私權](#)
- [使用 Amazon Quick Q 回答商業問題](#)
- [什麼是 Amazon Q Business ?](#)
- [什麼是 Amazon Q Developer ?](#)

AWS 解決方案程式庫

- [上的生成式 AI 應用程式建置器 AWS](#)
- [使用 Amazon Q Developer 建立自訂編碼搭配的指南](#)
- [使用 Amazon OpenSearch Service 自訂搜尋企業知識庫的指引](#)

AWS 其他資源

- [在 SDLC 中使用生成式 AI 的實際方法](#)
- [將智慧新增至您的開發人員操作](#)
- [生成式 AI 能力合作夥伴](#)
- [生成式 AI 客戶案例](#)
- [與的合作夥伴成功 AWS](#)
- [什麼是 AIOps ?](#)

- [什麼是 SDLC \(軟體開發生命週期\) ?](#)

其他資源

- [定義 IT 操作模型, 文件編號 W17B \(開放群組, 2017 年 9 月\)](#)

附錄 A：ADM AI 願景的範例架構

組織可以針對應用程式開發和維護 (ADM) 調整 AI 願景的這個範例架構，以明確表達其轉型目標。此範例包含簡介、明確的任務陳述式、可量化的目標，以及符合可衡量成功指標的策略計畫。

簡介

在現今快速發展的數位環境中，組織必須持續創新，才能保持競爭。專案 <您的專案名稱> 計畫代表我們透過生成式 AI 技術的策略整合，改變應用程式開發和維護 (ADM) 實務的遠大願景。

透過利用 AI 的強大功能，<您的公司名稱> 旨在大幅提升我們的開發速度、程式碼品質和營運效率。這種方法培養了前所未有的創新水準。此轉型將簡化我們的程序，讓團隊能夠提供卓越的軟體解決方案。這些解決方案將推動實際的商業價值和成長。

下列文件概述了我們透過生成式 AI 實現 ADM 目標操作模型 (TOM) 的任務、目標和關鍵策略計畫。

任務陳述式

利用生成式 AI 技術來轉換我們的 ADM 實務和軟體開發生命週期 (SDLC) 程序，以實現更快的創新、改善品質和增強的商業價值交付。

目標

1. 透過 AI 輔助程序，將應用程式開發和交付時間表加速 <您的公司價值>%。
2. 使用 AI 技術的分析和最佳化，改善程式碼品質並減少 <您的公司價值>% 的瑕疵。
3. 使用 AI 擴增工具和工作流程，透過 <您的公司價值>% 提高開發人員生產力。
4. 透過智慧型自動化和預測性維護，將營運成本降低 <您的公司價值>%。
5. 透過啟用 <您的公司價值> x 更快對市場變化和客戶需求的回應，提高業務敏捷性。

策略計畫

為了實現我們定義的目標並衡量實現商業價值的成功，我們將專注於符合關鍵績效指標的策略計畫，如下表所示。

策略計畫	關鍵任務	效能指標
1. AI 驅動的開發環境	1.1 實作 AI 輔助程式碼產生和完成工具。	• 新功能和應用程式的Time-to-market

	1.2. 整合 AI 驅動的程式碼檢閱和最佳化程序。	
	1.3. 開發 AI 增強型測試和品質保證工作流程。	
2. 智慧操作和維護	2.1 部署採用 AI 技術的監控和預測性維護系統。	<ul style="list-style-type: none">• 瑕疵率和平均解決時間• 交付應用程式的客戶滿意度分數
	2.2. 實作 AIOps 以進行自動化事件回應和解決。	
	2.3. 利用 AI 進行容量規劃和資源最佳化。	
3. AI 擴增的需求和描述 n	3.1 使用 AI 進行快速原型設計和反覆運算。	<ul style="list-style-type: none">• 開發人員生產力（例如：每天程式碼行、完成故事點）
	3.2. 實作 AI 輔助的市場分析和需求收集。	
	3.3. 開發 AI 驅動的工具，將業務需求轉換為技術規格	
4. 人才和組織轉型	4.1 為 ADM 建立 AI 卓越中心 (COE)。	<ul style="list-style-type: none">• ADM 程序中 AI 實作的投資回報 (ROI)
	4.2. 為所有角色開發全面的 AI 訓練計劃。	
	4.3. 重新定義任務角色和職業路徑，以整合 AI 技能。	

5. 治理和道德架構

5.1 為 ADM 程序中負責的 AI 使用建立政策。 • 符合政策、標準和法規需求

5.2. 建立 AI 道德審查委員會以進行持續監督。

5.3. 在採用 AI 技術的 ADM 中制定資料隱私權和安全性的指導方針。

專注於這些策略計畫，並根據定義的指標衡量進度，將推動 ADM 功能的顯著改善。這種方法透過採用 AI 技術的創新和效率，為業務和客戶提供更大的價值。我們預期達到以下結果：

- <您的公司價值> 百分比- <您的公司價值> 開發速度的百分比增加
- <您的公司價值> 百分比- <您的公司價值> 百分比降低生產瑕疵
- <您的公司價值> 百分比- <您的公司價值> 客戶滿意度分數的百分比改善

附錄 B：ADM TOM 的實作檢查清單

此完整檢查清單提供您實作應用程式開發和維護 (ADM) 目標操作模型 (TOM) 的結構化方法。檢查清單會考量下列每個實作階段的控管、組織結構、人員角色、程序和工具：

- [階段 1：基礎設定](#)
- [階段 2：能力建置](#)
- [階段 3：轉換擴展](#)

每個階段都以上一個階段為基礎，讓組織能夠有系統地擴展其 AI 功能，同時管理風險並確保企業整體採用的永續性。

階段 1：基礎設定

此階段發生在第 1–3 個月。它建立基本的控管結構，並引進必要的 AI 工具，同時實現快速成功。

治理和組織

- 1.1. 建立 AI 控管指導委員會。
- 1.2. 為 ADM 程序制定初始 AI 道德準則。
- 1.3. 建立基準 AI 風險評估架構。
- 1.4. 識別跨 ADM 團隊進行 AI 整合的關鍵角色。
- 1.5. 定義現有團隊中的初始 AI 擁護者角色。
- 1.6. 概述 ADM 中 AI 卓越中心 (COE) 的願景和任務。
- 1.7. 跨 ADM 團隊執行 AI 技能差距分析。
- 1.8. 為所有員工開發基本的 AI 讀寫能力訓練計畫。
- 1.9. 檢閱現有的廠商合約是否有 AI 整合潛力。
- 1.10. 在 ADM 中建立 AI 計畫的初始預算準則。

Roles (角色)

1.11. 軟體開發人員

- 採用 AI 輔助編碼、配對程式設計和程式碼完成工具。
- 建立檢閱和最佳化 AI 產生程式碼的準則。

1.12. 測試工程師

- 採用 AI 驅動的測試案例產生、執行和資料品質改善工具。
- 實作 AI 擴增的探索性測試技術。

1.13. UX 設計工具

- 採用 AI 輔助設計工具和資料驅動型設計技術。

1.14. DevOps 工程師

- 實作採用 AI 技術的 CI/CD 管道最佳化。
- 採用 AI 輔助基礎設施即程式碼 (IaC) 產生工具。

1.15. 支援工程師

- 使用 AI 支援的知識庫，加快問題解決速度。
- 實作 AI 驅動的票證分類和路由系統。

Processes

1.16. 針對複雜問題建立明確的呈報通訊協定。

1.17. 建立整合 AI 產生和手動程式碼的指導方針。

1.18. 為 AI 產生的程式碼開發新的 QA 程序。

1.19. 建立人工監督 AI 產生設計的程序。

1.20. 建立持續精簡 AI 測試模型的程序。

1.21. 改善新計畫的知識收集、方法精簡和重複使用體驗。

工具

- 1.22. 採用 AI 輔助編碼、配對程式設計和程式碼完成工具。
- 1.23. 實作 AI 驅動的程式碼品質、一致性檢查和錯誤偵測系統。
- 1.24. 為設計文件採用 AI 輔助文件工具。
- 1.25. 實作 AI 支援的協作工具，以減少時區相依性。
- 1.26. 採用 AI 驅動的測試案例產生、執行和資料品質改善工具。
- 1.27. 實作 AI 輔助專案估算工具。
- 1.28. 使用 AI 設定預測瑕疵分析。
- 1.29. 採用 AI 輔助設計工具和資料驅動型設計技術。

階段 2：能力建置

此階段發生在第 3-6 個月。它擴展了 AI 採用，並解決了中等複雜性的程序。

治理和組織

- 2.1. 實作 AI 控管政策和程序。
- 2.2. 為 ADM 專案建立 AI 道德審查程序。
- 2.3. 開發 ADM 程序的 AI 特定 KPIs。
- 2.4. 建立新的以 AI 為重心的角色，例如 AI 整合專家。
- 2.5. 重新調整團隊結構以支援 AI 擴增工作流程。
- 2.6. 使用專用團隊啟動 AI COE。
- 2.7. 建立 COE 操作程序和服務目錄。
- 2.8. 實作角色特定的 AI 訓練計劃。
- 2.9. 開發以 AI 為重心的職業道路和發展模型。
- 2.10. 制定 AI 特定的採購指導方針。

2.11. 實作 AI 成本分配和投資報酬率 (ROI) 追蹤機制。

Roles (角色)

2.12. 專案經理

- 整合 AI 驅動的專案規劃、風險評估和資源配置工具。
- 制定 AI 人類協作決策的通訊協定。
- 使用 AI 設定即時專案運作狀態監控和預測分析。

2.13. 版本管理員

- 採用 AI 支援的版本管理、規劃和風險評估工具。
- 使用 AI 實作自動化部署和復原策略。
- 設定預測發行後監控和問題偵測系統。

2.14. 站點可靠性工程師

- 採用 AI 驅動的預測性維護工具。
- 實作採用 AI 技術的異常偵測和自動化修復系統。

2.15. 技術寫入器

- 使用 AI 輔助的文件產生工具。
- 實作 AI 支援的內容最佳化和可讀性分析。

Processes

2.16. 建立意見回饋迴圈，根據專案成果持續改善 AI 模型。

2.17. 實作 AI 支援系統的持續學習機制。

2.18. 實作 AI 預測模型的持續學習機制。

2.19. 建立驗證 AI 產生的解決方案提案的程序。

2.20. 建立人工驗證 AI 產生的發行計畫的程序。

工具

- 2.21. 整合 AI 驅動的專案規劃、風險評估和資源配置工具。
- 2.22. 使用 AI 設定即時專案運作狀態監控和預測分析。
- 2.23. 實作 AI 驅動的工​​具來持續最佳化解決方案。
- 2.24. 實作 AI 驅動的​​使用者研究分析和角色建立系統。
- 2.25. 使用 AI 設定自動化可用性測試和意見回饋分析。
- 2.26. 採用 AI 支援的版本管理、規劃和風險評估工具。
- 2.27. 使用 AI 實作自動化部署和復原策略。
- 2.28. 設定預測發行後監控和問題偵測系統。
- 2.29. 實作 AI 驅動的監控、預測性維護和資源配置系統。
- 2.30. 使用 AI 設定加速問題解決程序。

階段 3：轉換擴展

此階段會​​在第 6–12 個月及之後發生。它實作了進階解決方案，並解決了更高的複雜性挑戰。

治理和組織

- 3.1. 將 AI 管控整合到整體企業管控中。
- 3.2. 實作 AI 政策的持續改進程序。
- 3.3. 建立跨職能 AI 控管委員會。
- 3.4. 在所有 ADM 團隊中完全整合 AI 角色。
- 3.5. 實作 AI 驅動的組織設計最佳化。
- 3.6. 展開 COE 功能以包含進階 AI 研究。
- 3.7. 與外部 AI 研究機構建立合作夥伴關係。

- 3.8. 實作 AI 支援的個人化學習路徑。
- 3.9. 為員工建立 AI 創新激勵計劃。
- 3.10. 開發 AI 特定的合約範本和服務水準協議 (SLAs)。
- 3.11. 實作 ADM 的 AI 驅動財務預測和最佳化。

Roles (角色)

3.12. 產品擁有者或業務分析師

- 實作 AI 驅動的市場分析和需求收集工具。
- 開發快速的工程技能，以進行有效的 AI 互動。

3.13. 解決方案架構師

- 採用 AI 驅動的解決方案設計工具和方法。
- 實作 AI 驅動的工具有來持續最佳化解決方案。

3.14. 全堆疊開發人員

- 採用 AI 驅動的全堆疊程式碼產生和最佳化工具。
- 實作 AI 驅動的 API 設計和整合系統。

3.15. 技術主管

- 採用 AI 驅動的應用程式生命週期管理工具。
- 建立訓練計畫，以在 AI 擴增的 DevOps 實務中提升團隊技能。

3.16. 安全主題專家 (SME) 實作 AI 驅動的威脅偵測和回應系統。

- 採用 AI 輔助的安全政策產生和合規檢查工具。

3.17. 網域特定 SME

- 使用 AI 工具來擷取特定領域的知識和應用程式。
- 實作 AI 輔助網域建模和模擬工具。

Processes

- 3.18. 重新設計企業架構 (EA) 程序，以整合 AI 驅動的洞察和自動化。
- 3.19. 實作 AI 系統的持續學習機制，以跟上不斷變化的法規。
- 3.20. 建立明確的通訊協定，以人工監督 AI 產生的合規建議。
- 3.21. 建立明確的通訊協定，以人工監督 AI 產生的建議。
- 3.22. 實作全面的變更管理策略。

工具

- 3.23. 實作 AI 驅動的架構決策支援系統。
- 3.24. 設定採用 AI 技術的整合和互通性評估系統。
- 3.25. 投資 AI 分析的資料整合和品質保證程序。
- 3.26. 為 AI 驅動的報告建立強大的安全和管理架構。
- 3.27. 實作適用於架構建議和資源佈建的 AI 驅動工具。
- 3.28. 整合 AI 驅動的觀測性和異常偵測系統。
- 3.29. 建立 AI 輔助的合規檢查和安全監控程序。
- 3.30. 實作 AI 驅動的市場分析和需求收集工具。
- 3.31. 採用 AI 技術的解決方案設計工具和方法。
- 3.32. 採用 AI 驅動的全堆疊程式碼產生和最佳化工具。
- 3.33. 實作 AI 驅動的 API 設計和整合系統。
- 3.34. 使用 AI 設定跨堆疊的自動化效能調校。
- 3.35. 採用 AI 驅動的應用程式生命週期管理工具。
- 3.36. 投資於可從所有位置存取的雲端 AI 增強型平台。
- 3.37. 全球標準化 AI 工具和環境。

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 4 月 18 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、持久性](#)。

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[機器學習模型可解釋性 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 闡道

集中式中介，可控制和標準化對[基礎模型](#)的存取。也稱為 LLM 闡道。

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 許可界限來將其實施。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

laC

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釋性 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 and 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。