



將 AWS 雲端中的應用程式現代化的分階段方法

AWS 方案指引



AWS 方案指引: 將 AWS 雲端中的應用程式現代化的分階段方法

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
目標業務成果	1
現代化程序	3
步驟 1. 評估您的應用程式	3
使用現代化診斷程序手冊	3
識別指標	5
步驟 2. 從小型開始並建置動能	6
驗證優先順序驅動程式	6
完成詳細資訊	7
建置基礎平台服務和現代化應用程式	7
步驟 3. 建立可擴展的現代化藍圖	7
現代化準備因素	9
代碼	9
建置和測試	9
發行版本	9
營運	10
最佳化	10
就緒狀態	10
後續步驟	11
Resources	12
相關指南	12
AWS 資源	12
文件歷史紀錄	13
.....	xiv

將 AWS 雲端中的應用程式現代化的分階段方法

Vijay Thumma 和 Ashish Ameta , Amazon Web Services (AWS)

2023 年 5 月 ([文件歷史記錄](#))

現代化需要多維度方法來採用和使用新技術、更快地交付產品組合、應用程式和基礎設施價值，以及以最佳價格定位組織進行擴展。它涉及最佳化、維護應用程式，並在現代化模型中操作，而不會中斷，並要求您簡化業務操作、架構和整體工程實務。

現代化不僅與應用程式有關，還需要提供安全且彈性操作架構的現代化基礎設施。在業務流程品質、可用性和敏捷性方面，應用程式和基礎設施是不可分割的。在不考慮基礎設施的情況下現代化應用程式會導致高整體成本，並對效能和品質造成負面影響。現代應用程式是結合新的架構模式、操作模型和軟體交付程序所建置。它們從零擴展到數百萬使用者，管理 TB (如果不是 PB) 的資料，可在全球使用，並在幾毫秒內回應。當您現代化在 Amazon Web Services (AWS) 雲端中管理的工作負載組合時，您可以使用容器、無伺服器技術、專用資料存放區和軟體自動化來重建、重構或取代這些工作負載，以獲得提供的完整敏捷性和總成本最佳化 (TCO) 優勢 AWS。

本指南適用於應用程式擁有者、企業擁有者、架構師、技術主管和專案經理。它討論了如何為在現代化評估階段中選取的應用程式開發基礎功能，以及使用分階段方法加速現代化工作的方式。

本指南是內容系列的一部分，涵蓋 建議的應用程式現代化方法 AWS。系列也包含：

- [AWS 雲端中應用程式現代化策略](#)
- [評估 AWS 雲端中應用程式的現代化準備程度](#)
- [將整體分解為微服務](#)
- [使用無 AWS 伺服器服務整合微服務](#)

目標業務成果

您應該預期應用程式現代化分階段方法的下列結果：

- 使用build-and-prove方法和雲端原生架構，例如微型服務，以更快的速度創新組織容量和功能。
- 一種變革管理和操作模型，可透過訓練和工具改進來建立組織準備度。
- 一種團隊方法，可協助在短短 12 週內交付初始結果，提供體驗式學習，並實現獨立且持久的客戶成功。
- 以微服務、APIs、可重複使用元件和容器化為基礎的組合式應用程式架構。

- 適用於特定策略應用程式的可擴展現代化藍圖，其中包括在split-and-seed模型中運作的規範性指導。在此模型中，現代化功能和服務會擴展到專注於業務成果的多個工程團隊。當定義新的最低可行產品 (MVPs) 時，初始團隊成員會分割以建立新的產品團隊。

現代化程序

階段式現代化方法的目標是使用現代化維度來提供增量值、將這些值套用至核心、區分業務的應用程式子集，以及加速採用現代技術。分階段方法包含三個步驟：

1. 使用現代化診斷手冊評估應用程式的成熟度。採取全方位的採用方法，並開發符合預期業務成果的流程。
2. 從小開始，準備交付 MVP，透過及早並逐步推動業務成果來獲得動力。此階段通常需要 12 到 16 週。
3. 建立可擴展的現代化藍圖，以在split-and-seed模型中運作。

下列各節會更詳細地討論這些步驟。

主題

- [步驟 1. 評估您的應用程式](#)
- [步驟 2. 從小型開始並建置動能](#)
- [步驟 3. 建立可擴展的現代化藍圖](#)

步驟 1. 評估您的應用程式

此階段的目標是：

- 徹底了解應用程式環境，並為現代化資料平台準備應用程式，因此您可以加速實現價值的時間，而不會影響業務，然後進行現代化、最佳化和擴展。
- 描述您的應用程式環境，以識別與變更相關的優點、風險和成本。
- 提供end-to-end的服務：從策略和規劃；透過部署、遷移和應用程式現代化；到持續支援。
- 建置提供可重複使用實務和工具的政策、建議和控制，以提供持續的商業價值。

在評估階段，應用程式擁有者和架構師會使用現代化診斷程序手冊來驗證其現代化目標和優先順序。

使用現代化診斷程序手冊

現代化診斷程序手冊提供程序，用於判斷企業從目前狀態移至未來狀態的值。這包含現代化涉及的技術變更。

您可以使用診斷程序手冊來判斷應用程式或應用程式套件的優先順序，以進行雲端現代化，並識別現代化期間需要解決的元件。

診斷維度

現代化診斷程序手冊可協助您了解應用程式或一組應用程式目前和目標（遷移後）狀態的下列維度：

- 應用程式分組 – 是否有理由將應用程式分組（例如，依技術或操作模型）以進行現代化？
- 定序 – 根據相依性，應用程式是否應該進行現代化？
- 技術 – 技術類別有哪些（例如，中介軟體、資料庫、簡訊）？
- 相依性 – 應用程式是否對其他系統或中介軟體具有關鍵相依性？
- 環境 – 使用多少個開發、測試和生產環境？
- 儲存 – 什麼是儲存需求（例如測試資料的複本數量）？
- 操作模型 – 應用程式的所有元件是否可以採用持續整合和持續交付 (CI/CD) 管道？
 - 若是如此，應該將哪些基礎設施責任分配給應用程式團隊和對象？
 - 如果沒有，營運團隊應該承擔哪些基礎設施責任（例如修補）？
- 交付模型：
 - 根據應用程式或應用程式群組，您應該修改、重構、重寫或取代？
 - 現代化的哪個部分應該使用雲端原生服務？
- 技能集 – 需要哪些專業知識？例如：
 - 雲端應用程式背景，使用容器和無伺服器技術從頭建置具有模組化架構的應用程式。
 - DevOps 專業知識可開發 CI/CD 程序、基礎設施即程式碼，以及使用開放原始碼和 AWS 工具和服務的自動化或應用程式可觀測性等領域的解決方案。
- 現代化方法 – 考量應用程式目前的狀態、雲端技術選擇、目前的技術負債、CI/CD、監控、技能和操作模型，需要完成哪些技術遷移工作？
- 現代化時機 – 商業產品組合的時機考量或其他可能影響現代化時機的計劃工作考量為何？
- 基礎設施的單位和總成本 – 根據經濟分析 AWS，在內部部署維護工作負載的年度成本是多少？

根據這些維度評估應用程式，可協助您在將現代化推向雲端時，在商業、技術和經濟上保持穩定。

構成要素

當您將應用程式現代化時，您可以將觀察分為三個基本區塊：業務敏捷性、組織敏捷性和工程有效性。

- 業務敏捷性 – 與業務內部將業務需求轉化為需求的有效性相關的實務。交付組織對業務請求的回應能力，以及業務在將功能發佈到生產環境中的控制能力。
- 組織敏捷性 – 定義交付程序的實務。範例包括敏捷的方法和 DevOps 典禮，以及角色指派和清晰度，以及整個組織的整體協作、溝通和啟用。
- 工程有效性 – 與品質保證、測試、CI/CD、組態管理、應用程式設計和原始程式碼管理相關的開發實務。

識別指標

若要了解您是否正在為客戶提供重要的服務，您必須實作可推動改善並加速交付的措施。目標、問題、指標 (GQM) 提供有效的架構，以確保您的指標符合這些條件。請依照下列步驟使用此架構來從目標中恢復：

1. 識別您正在執行的目標或結果。
2. 衍生必須回答的問題，以判斷目標是否達成。
3. 決定應測量或可測量的項目，以充分回答問題。有兩種測量類別：
 - 產品指標，可確保您為客戶提供重要的服務。
 - 操作指標，可確保您改善軟體交付生命週期。

產品指標

產品指標著重於業務成果，應在確定新工作範圍的投資報酬率 (ROI) 時建立。建立產品指標的實用技巧是詢問在實作新工作範圍時，業務會有什麼變化。將此思維正式化為測試形式，以專注於交付現代化功能時的真實情況，會很有幫助。

例如，如果您認為從舊版系統遷移交易將釋放加入用戶端的新機會，有什麼改善？必須建立多少容量才能加入下一個用戶端？如何建構測試以驗證該結果？在此案例中，您的產品指標可能包含下列項目：

- 識別商業價值測試或假設（例如，釋放 x% 的交易容量將加入 y% 的新業務）。
- 建立基準（例如，x 交易目前的容量支援 y 個客戶）。
- 驗證結果（例如，您已將容量提高 x%，因此您現在可以加入 y% 的新業務嗎？）

操作指標

若要判斷您是否正在改善軟體交付生命週期並加速現代化，您必須知道交付軟體的前置時間和實作時間。也就是說，您可以將業務需求轉換為生產功能的速度有多快？

有用的操作指標包括：

- 前置時間 – 從請求到生產，工作範圍需要多少時間？
- 週期時間 – 從頭到尾實作工作範圍需要多長時間？
- 部署頻率 – 您部署變更到生產環境的頻率為何？
- 還原服務的時間 – 從失敗中復原所需的時間（以平均修復時間或 MTTR 計算）？
- 變更失敗率 – 失敗之間的平均時間 (MTBF) 是多少？

步驟 2. 從小型開始並建置動能

此步驟的目標是交付初始的最少可行產品 (MVP)，以獲得動力。這種方法可讓您提早並遞增地推動業務成果。

驗證優先順序驅動程式

開始與應用程式團隊進行現代化工作之前，建議您先驗證先前決定的優先順序驅動因素。請遵循下列步驟：

1. 從診斷手冊編譯您需要的資訊。
 - 從優先順序應用程式清單中收集優先順序驅動因素和可行性評估。
 - 為您的應用程式收集轉換和目標狀態處置。
 - 識別雲端現代化規劃中的應用程式擁有者、架構師和利益相關者。
 - 如果知道，請取得相依性或應用程式套件排序的資訊。
 - 判斷庫存項目與相依性或應用程式套件群組的關聯。應用程式可能具有與其他元件緊密結合或相依的個別元件，而且您可能想要將這些元件現代化在一起。
2. 與步驟 1 的人員安排一小時或兩小時的會議，以驗證優先順序驅動因素。
 - 嘗試依解決方案工程師或架構師將多個（最多三個或四個）應用程式分組，並根據應用程式相依性或應用程式套件資訊在一次會議中進行討論。
 - 決定此近期會議每位團隊成員的角色和期望。
3. 舉行會議。

完成詳細資訊

在您遵循上一節中的程序來驗證優先順序驅動因素之後，您可以收集詳細資訊以判斷現代化方法和時間。

在此階段中，核心團隊與應用程式團隊在短期、兩天的衝刺中並肩合作，為雲端上的應用程式設計未來狀態 AWS。活動包括產品定義、產品探索、故事撰寫、價值串流映射和設計 CI/CD 程序。以下是一些想法：

- 為應用程式的每個個別元件建立模型（例如網路組態、儲存組態、資料庫、伺服器，以及應用程式在伺服器上的部署方式）。
- 使用容器或無伺服器技術等工具，將該模型解構為不同的建置區塊和組態。
- 將應用程式功能與基礎基礎設施上的任何相依性分開。將應用程式的函數抽象到您可以移動的元件中，而無需變更任何原始程式碼。
- 使用 CI/CD 工具和機制與 DevOps 緊密整合。

建置基礎平台服務和現代化應用程式

在此 12 週階段中，核心團隊受到完整堆疊團隊的支援，以提供優先的業務使用案例。此工作由多個雙披薩團隊執行。例如，平台工程團隊成立是為了開發基礎平台服務，而產品團隊成立是為了提供新的業務成果：

- 平台工程團隊會設定、整合和自訂 AWS 支援雲端基礎、開發人員工作流程和資料分析功能的服務。更大且更複雜的企業可能有多個團隊支援這些功能。
- 產品團隊為啟動階段中優先考慮的業務成果開發新的服務和體驗。隨著產品團隊開發新的服務，他們也會將核心業務功能現代化。

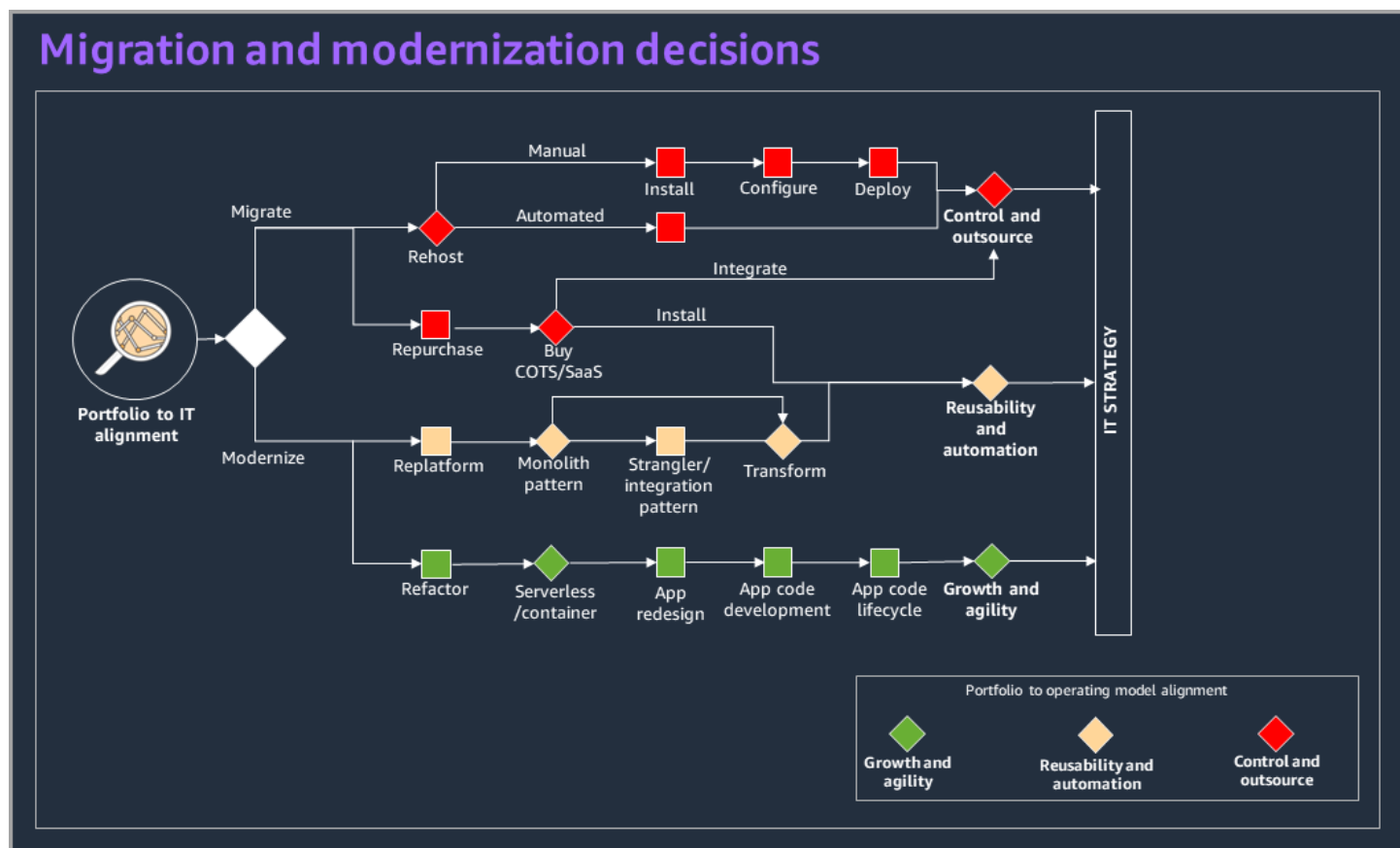
平台工程和產品團隊提供您可以評估的最低可行產品 (MVP)。初始 MVP 成功後，您可以使用 split-and-seed 方法擴展現代化計劃，藉此識別新應用程式並分割初始團隊成員以建立新的產品團隊。

步驟 3. 建立可擴展的現代化藍圖

在優先順序成果和應用程式的初始 MVP 版本之後，我們建議您制定藍圖，以擴展和加速現代化工作、提高開發人員生產力，以及快速創新。核心團隊會分割和植入新團隊，以便在專注於業務成果的多個工程團隊中擴展組織的功能和服務。透過隨著時間採用 split-and-seed 方法，您的組織可以進行更多開發並加速現代化的速度。

現代化藍圖應概述以明確定義模式進行應用程式現代化的實際和持續方法，例如事件驅動型、碰撞型、網域驅動型設計、分解、現代資料庫選項等。

藍圖應包含決策樹矩陣，如下圖所示，以識別應用程式元件並將其移至受管服務（例如資料庫服務），而不變更商業邏輯，或進程式碼層級的變更，以改善效能、可擴展性、可管理性、可靠性和資源使用量。



現代化準備因素

現代化應用程式時，請遵守下列標準和最佳實務。

主題

- [代碼](#)
- [建置和測試](#)
- [發行版本](#)
- [營運](#)
- [最佳化](#)
- [就緒狀態](#)

代碼

- 提供程式碼註解來記錄軟體的功能，並使用它們來產生文件。
- 遵循支援頻繁程式碼檢查和可追蹤功能請求的程式碼管理和部署程序。
- 建置包含單位、功能、效能和關鍵路徑測試的測試套件，具有 100% 的程式碼涵蓋範圍。
- 鼓勵程式碼重複使用，在您的程式碼基礎中提供相同或類似的功能。
- 開發原型，在投資完整的程式碼開發之前，先與使用者驗證功能。

建置和測試

- 根據測試重新定義功能完整性，以改善品質並防止重複發生的問題。
- 自動化接受測試。
- 監控所有自動化測試，並建立處理故障的程序。
- 追蹤生產和非生產環境中的效能、根據實際流量和負載測試定義服務層級目標 (SLOs)，並提供擴展以符合效能需求的能力。
- 從組態檔案中擷取敏感資料，並提供自動化和監控組態的工具。

發行版本

- 支援相依性（例如資料庫版本）、迴歸測試和追蹤，以自動化部署。

- 每次成功建置後，將程式碼逐步發佈至生產環境。
- 有效管理功能旗標（切換）：支援執行期組態、監控用量、在整個開發週期中維護旗標，以及依類別指派擁有者。
- 在建置管道中提供可追蹤性，以追蹤觸發條件、失敗通知和成功完成。
- 執行自動化部署程序，並在持續交付中測試「零接觸」程式碼更新。
- 使用零停機時間、全自動化的藍/綠部署方法。
- 確保您的資料庫結構描述變更在所有開發和生產環境中都一致地實作。

營運

- 建立已與您的通知系統整合的 DevOps 分類 Runbook。
- 請確定您的監控和通知系統符合服務層級目標 (SLOs)，並支援閾值、運作狀態檢查、非標準 HTTP 回應和非預期的結果。
- 建立有效的風險管理和災難復原程序。
- 開發符合您業務和法律要求的日誌輪換和保留策略。
- 開發可追蹤產品效能、衡量新功能是否成功，以及在指標不符合預期時顯示提醒的儀表板。

最佳化

- 根據效能和品質指標定期檢閱和改善程序。
- 實作根本原因分析和預防程序，以防止問題重複發生。
- 提供擷取產品運作狀態的資料驅動指標，並確保所有通知和動作都以這些指標為基礎。

就緒狀態

- 將跨職能團隊（包括業務合作夥伴、開發人員、測試人員和架構師）專用於您的現代化工作。

後續步驟

本指南提供逐步且增量的方法，讓您的應用程式現代化。我們建議您開始使用 技術和交付來範圍化您的工作流程，為期短期、中期和長期。您可以預期在工具、架構和實務上投入更多，以推動跨團隊更有效的交付和工程實務。設定現代化程序的目標、了解您計劃現代化的每個應用程式，並為每個應用程式選擇最佳的現代化方法。監控和最佳化您的進度。

AWS 將與您組織的業務開發團隊互動，以推動技術現代化並管理您的end-to-end需求，包括應用程式架構、設計和開發、行動支援、測試和協作解決方案。這些服務結合了經過驗證的程序、智慧型自動化、資料和模式的使用、開放標準，以及適當的人員，以協助您現代化舊版應用程式。主要目標的目標是：

- 協助開發平台，提供全自動化、以工具為基礎的方法來現代化傳統技術，並將其與現代化的整體知識結合。
- 使用全堆疊團隊建置 MVP，並將現代化工程功能專注於客戶成果和規模。

AWS 專業服務和 AWS 合作夥伴直接與資深技術和業務客戶利益相關者合作，推動企業現代化，同時反覆交付終端客戶價值。

Resources

相關指南

- [AWS 雲端中應用程式現代化策略](#)
- [評估 AWS 雲端中應用程式的現代化準備程度](#)
- [雲端中的現代化操作 AWS](#)
- [將整體分解為微服務](#)
- [使用無 AWS 伺服器服務整合微服務](#)
- [AWS Change Acceleration 6-Point架構和組織變革管理工具組](#)

AWS 資源

- [AWS 文件](#)
- [AWS 一般參考](#)
- [AWS 詞彙表](#)

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
擴展指引	—	2023 年 5 月 25 日
初次出版	—	2020 年 12 月 18 日

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。