使用 z/OS 資料庫的共用 IBM Db2 進行逐步遷移，以複寫您的大型主機應用程式

AWS 方案指引

AWS 方案指引: 使用 z/OS 資料庫的共用 IBM Db2 進行逐步遷移，以複寫您的大型主機應用程式

Table of Contents

簡介 1
業務成果 1
AWS Mainframe Modernization 3
平台轉換 3
自動化重構 3
轉換的優點 4
轉換程序 5
規劃 5
應用程式探索 5
資料相依性 6
容量基準 6
波規劃 8
建置 8
應用程式一致性 8
Architecture 9
執行中 11
兩階段遞交 (2PC) 11
執行期基礎設施 12
測試 14
來源環境 14
目標環境 14
分析 15
在 中測試您的應用程式 AWS Mainframe Modernization 15
切換 17
佈建 17
上線 17
轉返 18
結束 18
Architecture 18
最佳實務 20
網路延遲 20
安全性 20
應用程式控管 21
彈性 21

後續步驟 22
Resources 23
AWS 文件 23
Rocket 軟體參考 23
IBM 參考 23
工具 23
AWS 規範性指引模式和指南 23
文件歷史紀錄 24
詞彙表 25
..... 25
A 25
B 28
C 29
D 32
E 35
F 37
G 38
H 39
I 40
L 42
M 43
O 47
P 49
Q 51
R 51
S 54
T 57
U 58
V 59
W 59
Z 60
..... lxi

使用 z/OS 資料庫的共用 IBM Db2 進行逐步遷移，以複寫您的大型主機應用程式

Luis Gustavo Dantas 和 Andre Botura，Amazon Web Services (AWS)

2025 年 5 月 (文件歷史記錄)

在不斷變化的企業技術環境中，大型主機現代化已成為需要保持競爭性和敏捷性的組織的關鍵需求。這種轉型不只是用新舊系統取代舊系統；它是策略演變，可彌補過去強大、可靠的基礎與未來動態、創新可能性之間的差距。

大型主機曾經是無可爭議的企業運算領導者，現在正處於轉捩點。其無對等處理能力和安全功能幾十年來一直保持其相關性，但現今的企業需要能夠與雲端服務無縫整合的系統、支援行動應用程式，並利用人工智慧和大數據分析的力量。

現代化不一定需要從大型主機完全遷移。有些組織正在選擇利用大型主機和雲端環境優勢的混合式方法。此策略可讓他們在逐漸轉換至更現代化的平台時，維護重要的舊版應用程式。這項技術轉換不僅涉及系統更新，還需要轉型組織文化和技能。隨著公司現代化，他們透過填補世代差距並促進持續學習和創新，同時投資於新技術及其人力資源。

本指南討論逐步遷移策略，在大型主機系統的優點與現代雲端技術的優點之間取得平衡。這種分階段的轉換方法會先遷移應用程式層，同時保持與現有 IBM Db2 for z/OS 資料庫的連線，以簡化轉換程序，並將對關鍵業務操作的中斷降至最低，同時在雲端中採用新功能。

本指南專為參與大型主機現代化計畫的技術決策者和實作團隊而設計。主要受眾包括企業和解決方案架構師、技術專案經理和現代化計畫領導者，他們需要同時了解大型主機轉換的策略和技術層面。內容對實作團隊同樣重要，包括大型主機應用程式開發人員 AWS 或雲端工程師、資料庫管理員和 DevOps 工程師，他們負責執行實作現代化。

業務成果

公司有許多令人信服的理由可以更新其舊版應用程式。此程序會在各產業之間建立緊迫感。當較舊的專家淘汰時，他們會留下顯著的知識差距，這使得在失去此專業知識之前現代化系統至關重要。此外，公司受到降低成本、提高敏捷性以及快速回應快速變化市場條件的需求所驅動。

新興技術和增強客戶體驗的需求進一步增強了數位轉型的推動。這些因素結合與維護複雜系統相關的風險，促使組織在現代化其 IT 基礎設施時迅速採取行動。

特別是大型主機現代化，提供了精細的平衡動作。公司必須保留大型主機知名的穩定性和安全性，同時接受現代架構提供的彈性和可擴展性。此程序涉及有關要遷移哪些應用程式、要重寫哪些應用程式，以及要保留在大型主機上的複雜決策。

現代化的關鍵驅動因素包括敏捷性和降低成本：

- 敏捷性和上市時間。現代系統可加速採購程序，並更快速回應不斷變化的市場需求。採用 DevOps 和 SysOps 實務可大幅提升生產力和部署速度。
- 降低成本。現代化通常會透過下列方式降低基礎設施成本：
  - Pay-as-you-go模型，可讓成本與實際用量保持一致。
  - 降低與舊版系統相關的授權費用。
  - 改善彈性，提供更好的資源配置。
  - 主動-主動、高可用性設定，可增強系統彈性，同時最佳化資源使用率。

根據這些業務驅動因素，COBOL 應用程式轉換被視為現代化的策略方法。您可以使用共用資料庫來遵循漸進遷移路徑，以平衡現代化的需求與維持業務持續性的必要性。此方法可讓您利用現代架構的優點，同時保留 COBOL 應用程式的可靠性。因此，您可以實現敏捷性、成本效益和創新，同時緩解與大規模、突然轉換相關的風險。本指南中描述的共用 Db2 資料庫方法提供傳統系統和現代平台之間的橋樑，並啟用更順暢、更受控的現代化程序。

在本指南中：

- AWS Mainframe Modernization
- 轉換程序
- 最佳實務
- 後續步驟
- 資源
- 文件歷史記錄

AWS Mainframe Modernization

> **Note**
>
> AWS Mainframe Modernization 服務（受管執行期環境體驗）不再開放給新客戶。對於類似 AWS Mainframe Modernization Service（受管執行期環境體驗）的功能，請探索 AWS Mainframe Modernization Service（自我管理體驗）。現有客戶可以繼續正常使用該服務。如需詳細資訊，請參閱AWS Mainframe Modernization 可用性變更。

AWS Mainframe Modernization 此服務可讓您將舊版大型主機應用程式遷移至雲端原生環境、保留現有的商業邏輯和投資、使用自動化工具和受管執行期服務、最佳化應用程式效能，以及降低營運成本。此服務可簡化現代化程序，因此您可以利用雲端的強大功能，同時維持核心大型主機系統的價值。AWS 提供兩種主要主機現代化的重要方法：修改和自動重構。

平台轉換

AWS Mainframe Modernization 使用 Rocket Software（先前稱為 Micro Focus) 的 Replatform 為想要將大型主機應用程式遷移至雲端的企業提供強大的轉換選項，將中斷降至最低。此解決方案可讓您在上重新編譯和執行現有的 COBOL 和 PL/I 應用程式 AWS ，而不需要重大的程式碼變更。

Replatform with Rocket Software AWS 解決方案的主要優點包括：

- 保留現有的商業邏輯和投資
- 降低風險並加快上市時間
- 改善 AWS 基礎設施的可擴展性和效能
- 存取現代開發工具和實務

您可以使用此解決方案來維護熟悉的大型主機程式設計語言，同時利用 的彈性、成本效益和創新 AWS 雲端。

自動化重構

對於比轉換更轉換的方法，您可以使用 AWS Blu Age，其可將大型主機應用程式自動重構為 Java 型雲端原生應用程式。此解決方案可協助您更全面地現代化舊版系統，並將其轉換為可充分利用雲端原生技術的應用程式。

AWS Blu Age 的主要優點包括：

- 將舊版程式碼轉換為可維護的現代 Java 應用程式
- 減少手動工作量和潛在錯誤的自動化轉換
- 建立針對 最佳化的雲端原生應用程式 AWS 服務
- 改善敏捷性，更輕鬆地與現代技術整合

AWS Blu Age 可協助您遷移應用程式並為雲端做好準備，為創新和成長開啟新可能性。如需此方法的詳細資訊，請參閱 文件中的 AWS Mainframe Modernization 使用 AWS Blu Age 自動重構應用程式。

轉換的優點

本指南討論在 上重建大型主機 COBOL 應用程式的方法 AWS。此方法旨在現代化舊版系統，同時暫時保留 z/OS 的 IBM Db2，以簡化轉換程序。透過最初維護現有的資料庫結構，您可以在遷移期間降低複雜性和風險。此分階段方法可協助您受益於 的可擴展性和成本效益， AWS 雲端 同時保留關鍵資料完整性。分階段轉換的優點包括下列項目：

- 加速現代化：與重新構想雲端中的舊版應用程式相比，複寫和重構通常需要的時間和資源更少，因為它們不涉及重寫整個應用程式。此方法也支援更漸進的轉換，讓組織能夠以自己的步調進行現代化，同時立即受益於 的可擴展性和成本效益 AWS 雲端。
- 風險緩解：與重構許多組織相比，複寫提供了幾個優勢。公司可以維護現有的 COBOL 和 PL/I 程式碼庫、保留多年的商業邏輯，並將與大規模程式碼變更相關的風險降至最低。
- 資料持續性和分階段遷移：轉換的顯著優勢是最初將 z/OS 的資料保留在 Db2 中的原始資料格式。此策略可避免立即、複雜且具有潛在風險的資料遷移程序。透過在初始階段期間在其原始環境中維護資料，您可以保留資料完整性、減少停機時間，並將現代化過程中資料遺失或損毀的風險降至最低。第二個步驟是，您可以規劃將受控的分階段資料遷移至雲端原生資料庫，該資料庫涉及徹底的測試和驗證，同時應用程式繼續在已轉換的環境中執行。
- 彈性和面向未來的公司：對於在大型主機技能和應用程式方面有重大投資的公司，重建提供了一種實用的現代化途徑，以平衡創新與持續性。它提供了最初保留關鍵資料結構和存取方法的靈活性，同時還為未來的現代化工作設定舞台，包括最終資料遷移到完全雲端原生解決方案。

組織可以遵循轉換方法，以自己的步調進行現代化，並解決立即需求，同時規劃長期數位轉型目標。這種方法也讓公司有機會在雲端原生服務上培訓員工。

轉換程序

對於想要利用雲端運算優勢，同時保留其寶貴的舊版應用程式的組織而言，大型主機現代化是至關重要的步驟。此轉換會帶來重大挑戰。大型主機應用程式通常高度耦合，並具有複雜的相互依存性，這些相互依存性在數十年的營運過程中不斷演進。這種複雜性需要謹慎且有條不紊地進行現代化。

組織需要導覽下列關鍵階段，才能成功轉換：

- 規劃：此階段涉及對現有系統的全面探索和現代化工作的優先順序。組織會評估其目前的基礎設施、識別關鍵應用程式，並判斷哪些系統需要先進行現代化。
- 建置：在此階段，組織會建立遷移應用程式和開發新系統和基礎設施的程序。這包括設計和實作現代化架構，以及編譯原始程式碼。
- 執行中：此步驟包含建立執行期環境來託管已轉換的應用程式。它涉及設定必要的硬體、軟體和雲端基礎設施，以支援現代化系統，並確保它們可以在新環境中有效率地運作。
- 測試：此階段包括對現代化系統的嚴格驗證，以確認符合所有功能和效能要求。進行廣泛測試，以驗證新環境的資料完整性、系統相容性和整體效能。
- 切換：最後一個階段著重於實作策略，以順利轉換和控制從舊版大型主機到現代化環境的轉移。這包括仔細規劃遷移排程和應變計畫，以將業務營運中斷降至最低。

下列各節詳細討論這些階段：

- 規劃
- 建置
- 執行中
- 測試
- 切換

規劃

為了有效導覽大型主機舊版應用程式的需求，組織通常會從對大型主機環境的全面評估開始。

應用程式探索

在這個初始階段的強大工具是 Rocket Enterprise Analyzer，它提供對大型主機應用程式的結構、相依性和複雜性的深入見解。此工具可協助您判斷現代化工作的範圍、潛在風險和最佳化的機會。

要發現的一個關鍵方面是大型主機系統內的複雜資料相依性 Web。這些相依性通常隱藏在傳統程式碼層下，並可能大幅影響現代化工作。透過映射不同的應用程式和模組如何與各種資料來源互動，您可以更好地了解您計劃實作的任何變更的潛在影響。

資料相依性

徹底評估資料相依性，可以顯示大型主機環境中有關資料流程、資料品質和資料控管的重要資訊。此知識在規劃資料遷移策略、確保現代化期間的資料完整性，以及識別資料最佳化的機會時非常寶貴。透過清楚地了解您的資料，您可以更明智地決定哪些現代化方法對現有操作最有效且最不造成干擾。

由上而下的分析，可依交易或任務控制語言 (JCL) 任務識別資料表的使用，是建立波動規劃和優先順序的關鍵。此方法可釐清大型主機系統不同元件之間的關係，並協助您開發策略分階段的現代化方法。透過識別最常存取的資料表和程序，您可以優先考慮現代化工作：您可以先專注於高影響的領域，並確保更順暢的轉換，並將關鍵業務營運中斷降至最低。

除了使用 Rocket Enterprise Analyzer 來探索資料相依性之外，許多組織也會使用自己的自訂解決方案來深入了解其大型主機環境。這些內部工具通常會利用 IBM Db2 目錄和系統管理設施 (SMF) 記錄中提供的大量資訊。

容量基準

規劃大型主機轉譯專案的一個步驟是收集有關目前工作負載耗用量的詳細資訊。此資料可協助您準確預測和佈建目標雲端環境中的初始所需容量。例如，我們建議您從 IBM Customer Information Control System (CICS) 或 Information Management System (IMS) 和任務控制語言 (JCL) 任務收集線上交易和批次交易的每小時百萬次指令/秒 (MIPS) 耗用資料。

IBM 為大型主機運算中的 MIPS 提供各種定價模型，其中許多模型以尖峰用量為中心。在這些峰值型模型中，最常見的是四小時滾動峰值。

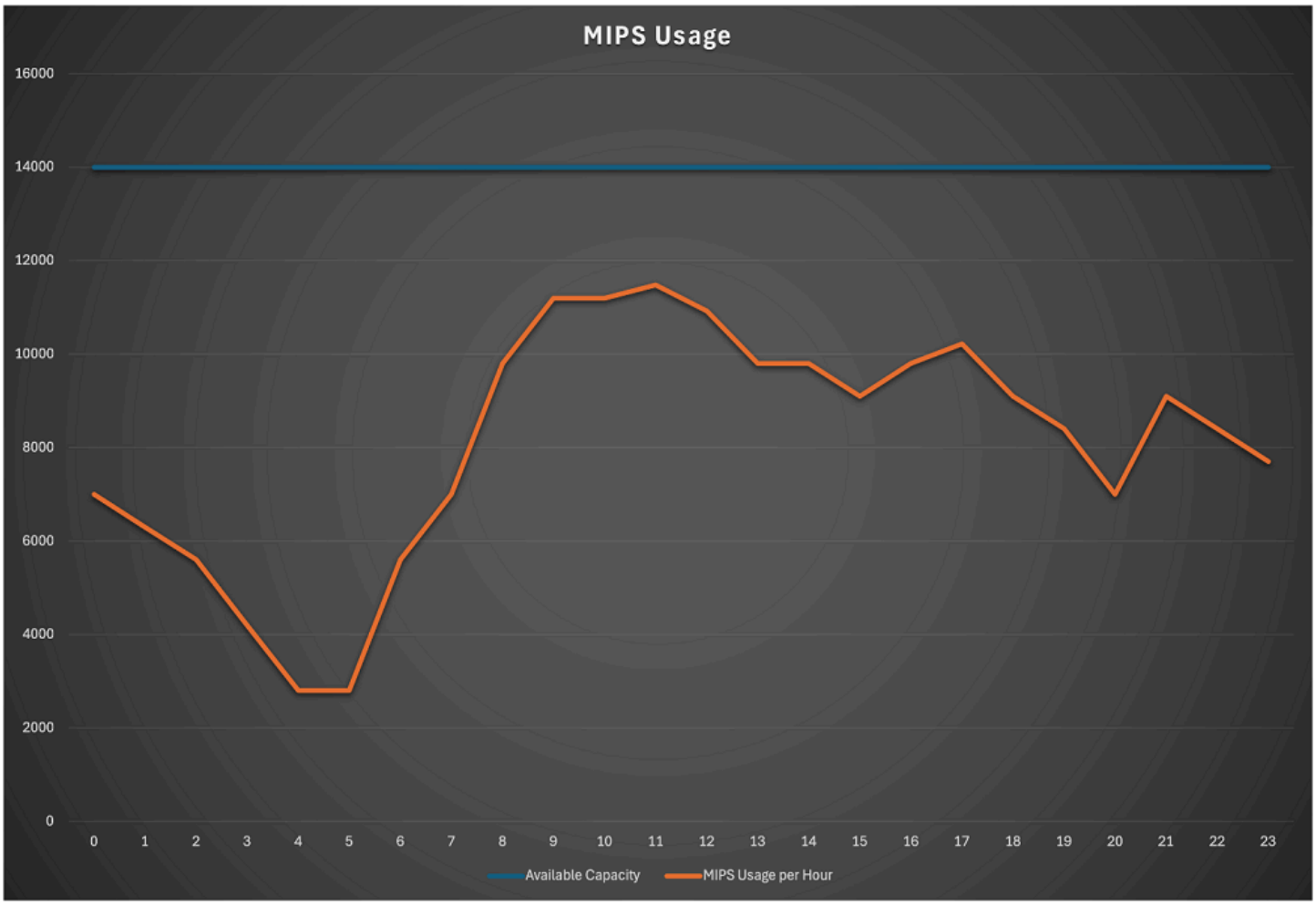
大型主機成本包括影響整體費用的五個關鍵領域：

- 軟體授權通常是主要元件。它涵蓋作業系統、中介軟體、資料庫和各種應用程式，而成本有時與機器容量或用量相關。
- 硬體費用包括首次購買或租用大型主機設備、持續維護和升級。
- 由於受管資料數量龐大，且涉及磁碟系統、磁帶程式庫和相關聯的管理軟體，因此儲存成本可能相當龐大。
- 人事費用涵蓋專業大型主機專業人員的薪資，例如系統程式設計人員和資料庫管理員。
- 災難復原和業務連續性措施，包括備份系統、備援硬體和離站復原設施，在確保高可用性和快速復原方面代表了重大投資。

這五個成本類別結合 MIPS 型費用，構成大多數大型主機預算的核心。不過，根據組織的大小、產業和特定大型主機使用率模式，其相對比例可能會有很大的差異。

每小時 MIPS 資料對於全面了解大型主機工作負載模式和效能至關重要。與每日或每月平均值不同，每小時資料提供精細的洞見，可顯示系統資源使用率一整天中的細微波動。此詳細程度對於在雲端 中準確評估應用程式的效能和容量需求來說非常寶貴。

透過分析每小時 MIPS 資料，您可以識別尖峰使用期間、 spot 趨勢，並找出彙總資料中可能模糊的潛在瓶頸，如下圖所示。這種精細程度允許更精確的容量規劃，有助於最佳化資源配置，並可能導致節省成本並改善系統效率。



每小時 MIPS 資料也做為基本效能基準工具。它會建立系統效能的詳細基準，這在您規劃或評估遷移或升級等系統變更時特別重要。透過比較變更前和變更後每小時 MIPS 資料，您可以準確測量這些修改對系統效能的影響，並確保您的大型主機持續滿足組織的需求。

若要收集每小時 MIPS 資料，您有幾個選項。其中一種方法是直接使用 SMF 記錄。這些記錄提供有關系統活動和資源用量的豐富資訊。或者，您可以使用專門的工具，例如 IBM 子容量報告工具 (SCRT)，這可以簡化收集和分析 MIPS 資料的程序。

無論您選擇哪種方法，收集資料的時間都很重要，最好是幾個月。此延長的收集期間可讓您考慮工作負載中的週期性變化，例如end-of-month處理尖峰或季節性波動。透過擷取這些長期模式，您可以更準確且全面地了解大型主機的效能特性，從而實現更明智的決策和更有效的容量管理。

波規劃

您可以使用所收集的資訊，以策略方式排定大型主機複寫計劃的優先順序。謹慎的方法是從較不重要的工作負載開始，例如非核心商業交易或批次任務，讓團隊以最低的基本操作風險來獲得經驗並精簡程序。此外，將唯讀工作負載視為早期遷移的候選項目可能很有利，因為這些工作負載通常涉及較低的複雜性和較低的資料不一致風險。這種方法可讓您在重建工作中建立信心和動力。

此外，將共用 Db2 資料表以進行寫入或更新操作的工作負載分組可以簡化遷移程序。透過識別這些互連工作負載，您可以規劃具有一致性的遷移波，以維護資料完整性並將對複雜臨時解決方案的需求降至最低。此策略不僅可以降低資料衝突的風險，還可以透過同時解決相關元件來最佳化整體重建時間軸。最後，此資料驅動的優先順序方法可確保對關鍵性、複雜性和相互依存性的平衡考量，並導致更有效率且成功的大型主機現代化程序。

建置

使用共用的 Db2 資料庫可在大型主機和雲端環境中同時執行相同或一致的應用程式。當您在兩個平台上維護相同的應用程式版本時，此方法提供數種優點，並在操作中提供增強的彈性和可靠性。

此策略的一個主要優點是能夠實作有效的復原計劃。如果在遷移或部署期間發生問題，具有相同的應用程式版本可讓 無縫還原至先前的狀態，並將停機時間和潛在的資料不一致降至最低。

應用程式一致性

在轉換過程中，將應用程式元件從分散式來源控制管理員鏡像到大型主機是一種策略方法。此方法支援使用現代原始程式碼管理工具，同時保持與大型主機環境的同步。此鏡像程序是暫時的，只會持續到工作負載在分散式平台上的生產中完全運作為止。

透過將已轉換應用程式的原始碼遷移至分散式變更管理工具，您可以利用現代原始碼管理員提供的多種優點。其中包含：

- 增強的協同合作：分散式工具通常會透過包含提取請求、程式碼檢閱和分支策略等功能，為團隊協同合作提供更好的支援。

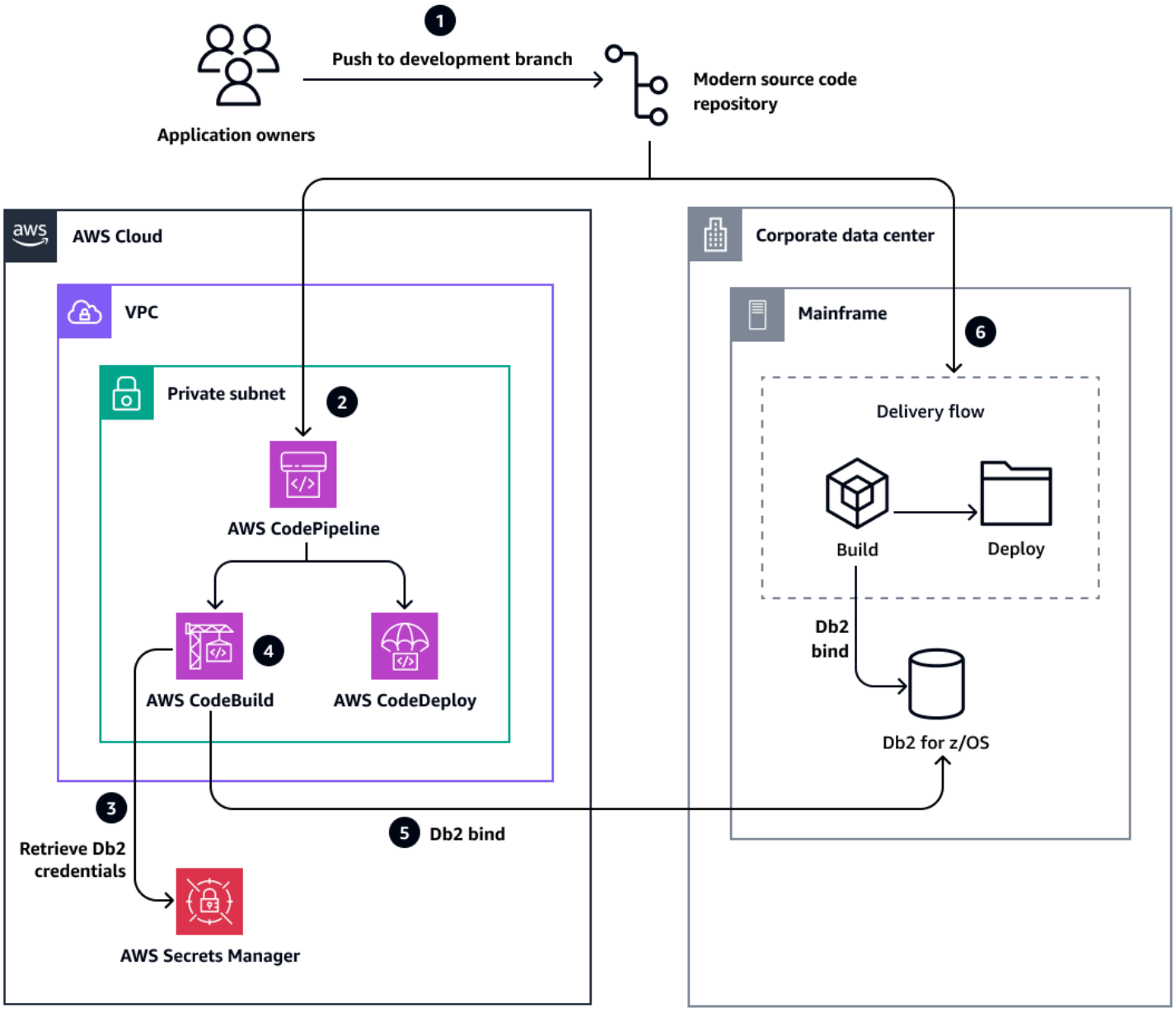
- 改善版本控制：現代系統提供更精細的版本控制，讓您更輕鬆地追蹤變更和管理不同版本的程式碼。
- 與 CI/CD 管道整合：許多分散式工具與持續整合和持續部署 (CI/CD) 管道無縫整合，可簡化開發程序。
- 更好的可見性和可追蹤性：這些工具通常提供卓越的儀表板和報告功能，並提供更深入的開發程序洞察。
- 現代開發實務的支援：分散式系統通常更適合敏捷的方法和 DevOps 實務。

鏡像程序涉及將程式碼從分散式來源控制管理員同步回大型主機。這可確保兩個環境在轉換期間保持一致。不過，您必須實作鏡射做為單向同步，其中 會將流程從分散式系統更新至大型主機，而不是雙向。此方法可維持一致性，並防止兩個環境中同時更新可能產生的潛在衝突。

透過採用此鏡像策略，您可以逐步將開發工作轉移到分散式平台，同時確保大型主機環境保持up-to-date。這可在轉換過程中提供更順暢的轉換和安全網路。當工作負載在分散式生產環境中完全正常運作且穩定時，您可以逐步淘汰鏡像程序，並完成遷移至現代原始程式碼管理系統。

Architecture

下圖顯示分散式原始碼管理系統如何鏡射應用程式元件，並維持 AWS 雲端 和大型主機環境之間的同步。 AWS 雲端 環境使用 CI/CD 服務，例如 AWS CodeBuild、 AWS CodePipeline和 AWS CodeDeploy來建置和部署應用程式。



在此工作流程中：

1. 應用程式擁有者會將新的應用程式版本交付至原始碼儲存庫的開發分支。
2. 新版本會觸發 AWS CodePipeline。
3. AWS CodeBuild 從 擷取 Db2 登入資料AWS Secrets Manager。
4. CodeBuild 會編譯應用程式。
5. CodeBuild 針對 z/OS 使用 Db2 來繫結應用程式。
6. 大型主機交付流程也會建置和部署應用程式。

執行中

為了確保雲端應用程式與內部部署資料庫之間的最佳效能和低延遲，我們建議您實作 AWS Direct Connect。此服務可在 AWS 與組織的資料中心之間提供專用網路連線，相較於以網際網路為基礎的連線，可提供更一致的網路效能並降低延遲。這對於需要快速回應時間的資料庫操作特別重要。

若要為在 上執行的應用程式實現高可用性 (HA) 和彈性 AWS，您可以使用下列元件實作強大的架構：

- Elastic Load Balancing (ELB)：您可以部署負載平衡器，將傳入流量分散到應用程式執行所在的多個 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。這可確保工作負載的均勻分佈，並為用戶端請求提供單一進入點。
- Auto Scaling 群組：託管應用程式的 EC2 執行個體可以組織成 Auto Scaling 群組。這可讓基礎設施根據預先定義的指標自動調整執行個體數量，例如 CPU 使用率或網路流量。在尖峰時間，可以啟動其他執行個體來處理增加的負載，而在更安靜的期間，可以終止不必要的執行個體以最佳化成本。
- EC2 執行個體：應用程式可以部署在 Auto Scaling 群組內的 EC2 執行個體上。這些執行個體應分散到多個可用區域，以增強容錯能力並確保高可用性。
- 異地同步備份部署：透過將應用程式執行個體分散到多個可用區域，系統可以承受單一可用區域的故障，而不會影響整體可用性。

此架構可讓應用程式根據需求無縫擴展，同時維持高可用性。負載平衡器可確保流量平均分佈於運作狀態良好的執行個體，而 Auto Scaling 群組會根據實際工作負載管理執行個體數量。

若要進一步增強可靠性，您可以使用 Amazon CloudWatch 實作強大的監控和警示系統，以協助快速偵測和回應任何效能問題或故障。此外，定期測試自動擴展功能和容錯移轉案例，可確保系統在各種負載條件和潛在故障期間如預期般運作。

透過採用此方法，您可以受益於 的可擴展性和靈活性， AWS 雲端 同時保持與現場部署 Db2 資料庫的安全連線。此混合式設定是實現完整雲端遷移的絕佳途徑，並在整個過程中提供漸進式轉移和風險緩解。

兩階段遞交 (2PC)

AWS Mainframe Modernization 使用 Rocket Software 的 Replatform 透過實作擴充架構 (XA) 來支援兩階段遞交 (2PC) 交易。此功能對於跨分散式系統維護資料完整性至關重要，尤其是在複雜交易通常跨越多個資源的大型環境中。

XA 架構與 Replatform with Rocket Software AWS 整合，可協調資料庫和訊息佇列等各種資源的交易。此整合可確保分散式交易的所有部分一致遞交或復原，以維持整個系統的一致性。

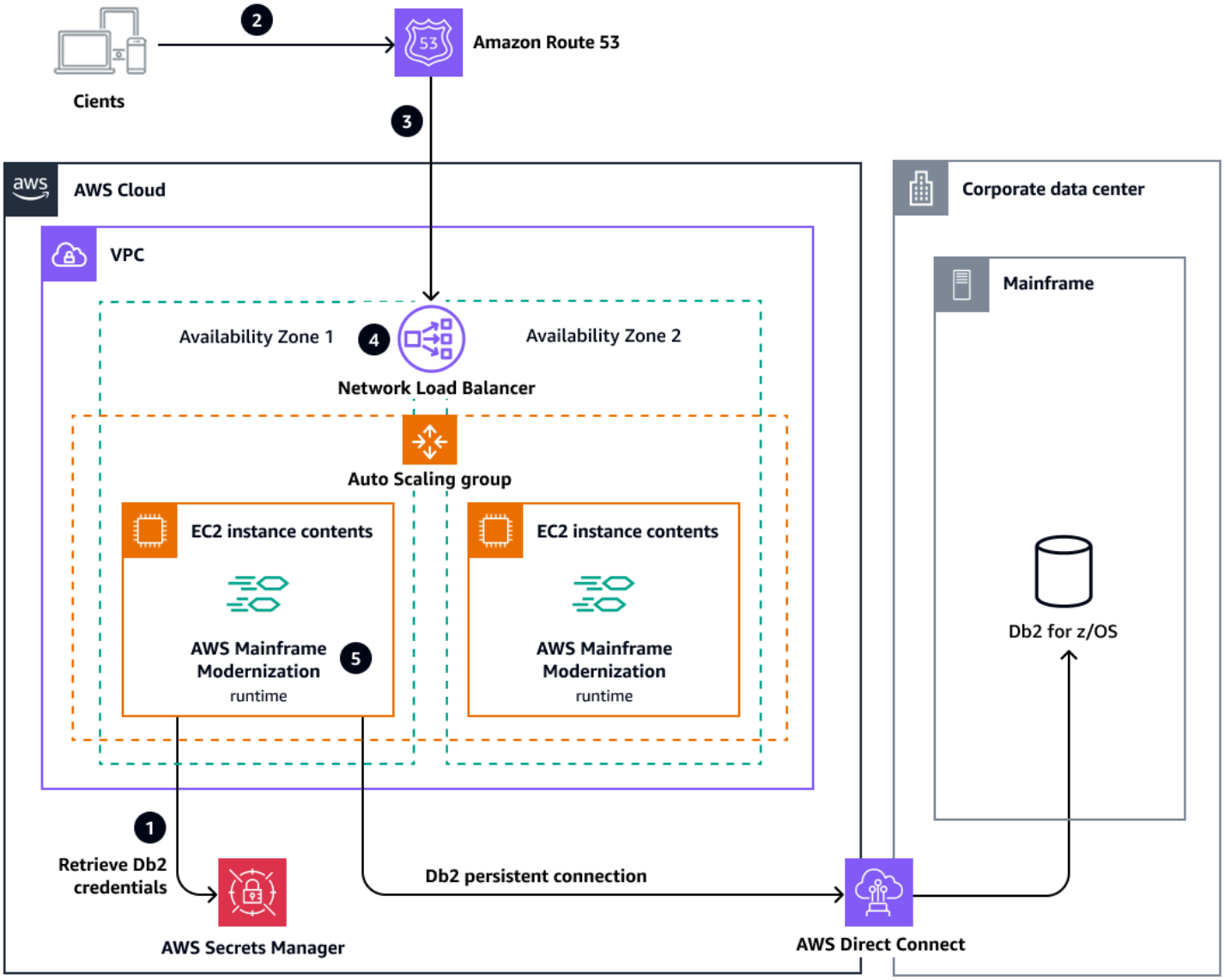
2PC 程序包含兩個階段：

- 準備階段：交易管理員會查詢交易中涉及的所有資源管理員，以確保他們已準備好遞交。
- 遞交階段：如果所有資源管理員都做出正面回應，交易管理員會指示他們遞交變更。如果有任何資源管理員無法遞交，則會指示所有管理員轉返變更。

透過使用 XA， AWS Replatform 與 Rocket Software 提供可靠且可擴展的解決方案，可在現代化大型主機環境中管理複雜的分散式交易。此功能對於想要將大型主機應用程式遷移至雲端而不犧牲交易完整性或效能的組織至關重要。

執行期基礎設施

下圖顯示 中的高可用性和彈性環境 AWS 雲端 ，其中包含兩個可用區域、Auto Scaling 群組中的 EC2 執行個體、Network Load Balancer，以及 AWS 和大型主機環境之間的專用連線 AWS Direct Connect。



在此架構中：

1. 當 AWS Mainframe Modernization 執行時間開始時，它會從 擷取 Db2 登入資料，AWS Secrets Manager並開啟與 Db2 for z/OS 的持久性連線。

> **Note**
>
> AWS Mainframe Modernization 服務（受管執行期環境體驗）不再開放給新客戶。對於類似 AWS Mainframe Modernization Service（受管執行期環境體驗）的功能，請探索 AWS Mainframe Modernization Service（自我管理體驗）。現有客戶可以繼續正常使用該服務。如需詳細資訊，請參閱AWS Mainframe Modernization 可用性變更。

2. 用戶端會在 Amazon Route 53 中繫結 Network Load Balancer 地址。
3. Route 53 會將交易重新導向至 Network Load Balancer。
4. Network Load Balancer 會將交易分散到多個 EC2 執行個體。
5. 在 上執行的工作負載會透過持久性連線與 z/OS 的 Db2 AWS Mainframe Modernization 互動 AWS Direct Connect。

測試

當您將 z/OS 的 Db2 維護為共用資料庫時，修改 COBOL 應用程式時，請務必確保新的系統功能與原始系統相同。此混合環境提供獨特的挑戰和測試機會。下列策略概述功能等效性測試的全方位方法，旨在驗證已轉換應用程式的效能、資料完整性，以及與 z/OS 資料庫現有 Db2 的無縫整合。

首先識別需要在系統之間比較的關鍵業務流程和交易。然後，建立包含特定案例的詳細測試計畫，以有效地評估這些交易的功能等效性。最後，開發涵蓋所有已識別案例的完整測試資料集，並確保這兩個系統都相同，以便進行準確的比較。

來源環境

- 初始快照（第一個快照）：
  - 請確定測試期間其他應用程式未使用資料表，因為這可能會影響同等測試。
  - 在執行任何測試之前，為交易所使用的 z/OS 資料表拍攝 Db2 快照。
- 來源系統測試：
  - 在原始 COBOL 應用程式上執行完整的測試套件。
  - 記錄所有交易、輸入和輸出。
  - 監控系統效能和資源使用率。
- 來源後測試快照（第二個快照）：
  - 完成來源系統測試後，請拍攝 z/OS 資料庫 Db2 的另一個快照。

目標環境

- 資料庫重設：
  - 使用第一個快照將資料庫還原為其初始狀態。
- 目標系統測試（已修改的環境）：
  - 在已轉換的應用程式上執行相同的測試套件。

- 確保所有目標系統測試都使用與來源系統測試相同的輸入。
- 監控系統效能和資源使用率。

- 目標後測試快照（第三個快照）：
  - 完成目標系統測試後，拍攝 z/OS 資料庫 Db2 的最終快照。

分析

- 比較和分析：
  - 比較第二個和第三個快照，以識別資料中的任何差異。
  - 分析測試結果，並比較來源和目標系統的輸出。
  - 評估兩個環境之間的效能指標。
- 整合測試：
  - 執行涉及已轉換應用程式和任何剩餘 COBOL 元件的測試。
  - 驗證兩個環境之間的無縫互動。
- 容錯移轉和復原測試：
  - 測試一個環境失敗且另一個環境接管的案例。
  - 確保容錯移轉情況下的資料一致性和完整性。
- 負載和壓力測試：
  - 執行具有不同負載的測試，以評估混合系統在壓力下的表現。
  - 識別任一環境中的任何瓶頸或效能問題。
- 文件和報告：
  - 記錄所有測試結果、差異和效能指標。
  - 準備可比較來源和目標系統的完整報告。

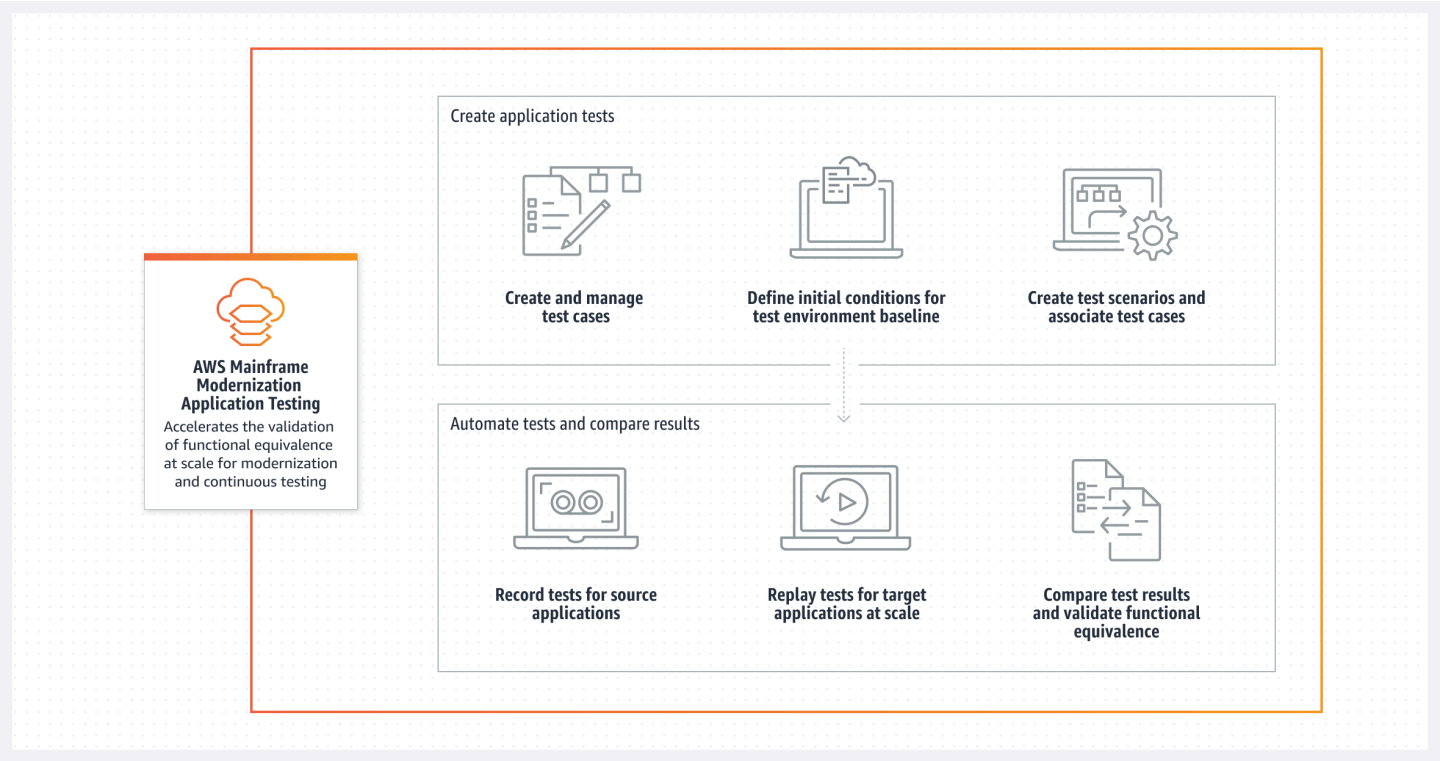
在 中測試您的應用程式 AWS Mainframe Modernization

AWS Mainframe Modernization 應用程式測試 此服務可大規模自動執行應用程式測試。 AWS 應用程式測試 可協助最佳化和降低大型主機應用程式現代化和測試專案成本。

> **Note**
>
> AWS Mainframe Modernization 服務（受管執行期環境體驗）不再開放給新客戶。對於類似 AWS Mainframe Modernization Service（受管執行期環境體驗）的功能，請探索 AWS

Mainframe Modernization Service（自我管理體驗）。現有客戶可以繼續正常使用該服務。如需詳細資訊，請參閱AWS Mainframe Modernization 可用性變更。

下圖顯示 如何在高階 AWS 應用程式測試 運作。



程序包含下列步驟：

1. 建立和管理測試案例，這是測試動作的最小單位。識別最能代表來源和目標系統之間的功能同等性的資料類型。
2. 透過指定 CloudFormation 範本和其他屬性來定義測試環境的組態。
3. 建立測試套件，這是測試案例的集合。
4. 上傳和重播資料集：擷取大型主機上的輸入和輸出資料集，將其上傳到 AWS，然後在目標系統上重播測試案例。
5. 比較來源和目標資料集。 AWS 應用程式測試 會自動比較來源和目標系統的輸出資料集。檢閱並評估這些項目以識別差異。

如需詳細資訊，請參閱AWS Mainframe Modernization文件。

切換

在大型主機現代化中，其中一個最重要的挑戰是在轉換到新平台期間將停機時間和風險降至最低。藍/綠部署策略提供強大且靈活的系統遷移方法。

藍/綠部署是一種技術，可透過執行兩個稱為藍和綠的相同生產環境來減少停機時間和風險。以下是它在大型主機現代化環境中的運作方式：

- 藍色環境：這是目前處理所有生產流量的大型主機系統。
- 綠色環境：這是您 AWS 準備好接管的新現代化平台。

藍/綠切換策略包括以下步驟：佈建、上線、發生問題時復原，以及結束。

佈建

在此階段中，您會 AWS 遵循下列步驟，在 上佈建新的 （綠色） 環境：

1. 轉譯環境：Route 53 託管區域必須包含指向大型主機環境 （藍色） 的 DNS 記錄。
2. 驗證連線：確保您的 AWS 帳戶 和內部部署交易管理員與 z/OS 資料庫的 Db2 之間有適當的連線。
3. 執行煙霧測試：使用 AWS 負載平衡器地址來存取已轉換的環境，並執行全面的煙霧測試來驗證下列項目：
   - 所有預期的工作負載都可用。
   - 3270 筆交易正在正確處理。
   - 與適用於 z/OS 的 Db2 進行的資料互動如預期般運作。

上線

在此階段，您將流量轉移到綠色環境並監控變更。

1. 使用 Route 53 中的流量路由政策來轉移流量：
   - 選項 A：您可以一次轉移所有流量。
   - 選項 B：或者，您可以使用漸進加權分佈。
2. 監控並驗證：
   - 在流量轉移時密切觀察 AWS 環境。
   - 檢查 3270 交易處理。

- 驗證 Db2 以進行 z/OS 通訊。
- 監控效能問題。
- 讓使用者驗證交易結果。

轉返

如果發生問題，您可以快速更新 Route 53，將流量重新導向回內部部署大型主機 （藍色） 環境。

在嘗試另一個切換之前，您應該調查並解決問題。

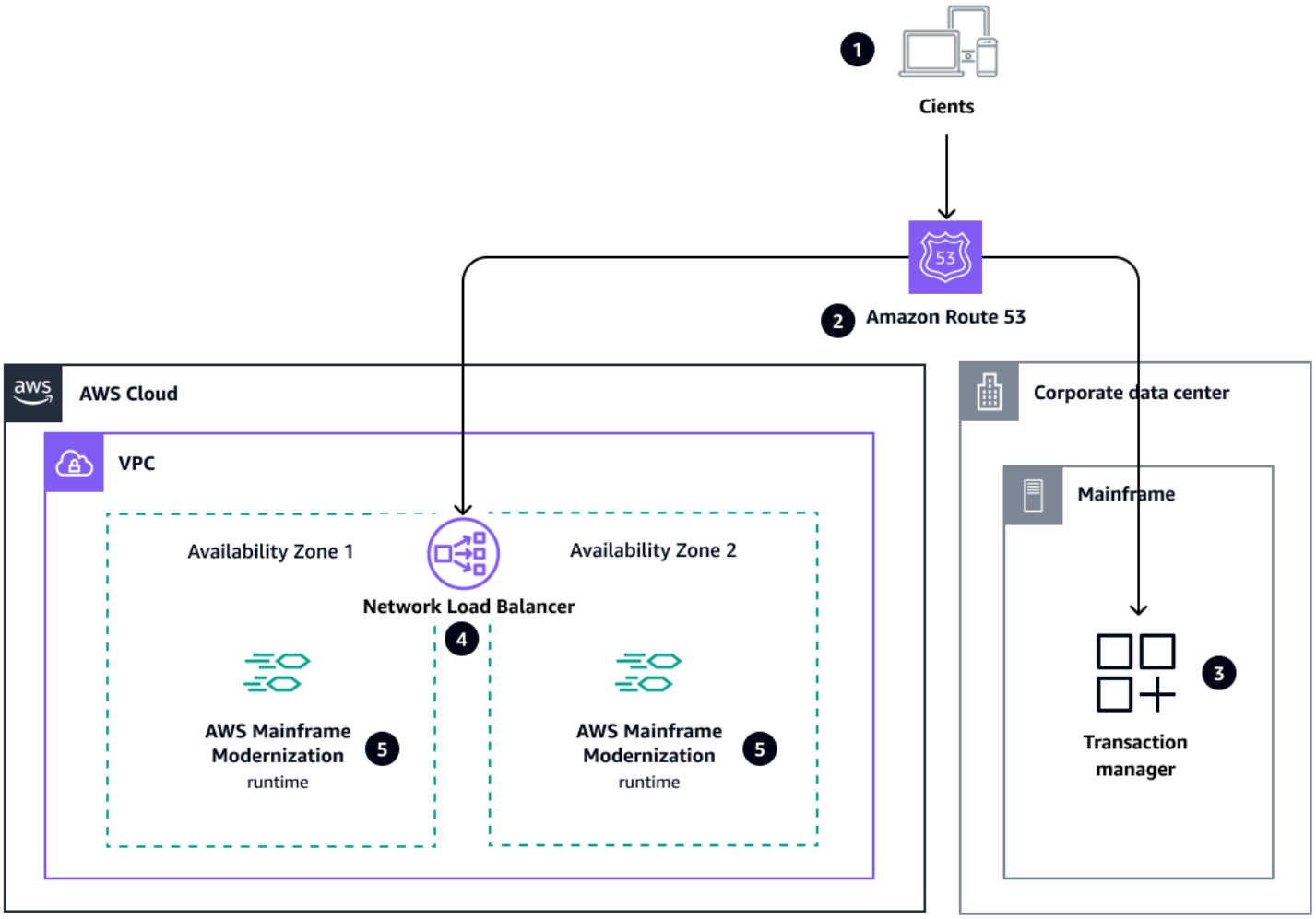
結束

監控流量並驗證綠色環境是否正常運作後，您可以逐步將應用程式流量增加到 AWS。

在穩定期間之後，您可以停用大型主機交易 （藍色） 環境，並將 z/OS 資料庫的 Db2 保留在內部部署。

Architecture

下圖說明切換流程。



切換程序包含下列項目：

1. 前端 (BFFs) 的用戶端應用程式、前端和後端會將交易傳送至 Route 53 網域名稱。
2. Route 53 會根據定義的路由政策，將連線路由至大型主機交易管理員或 Network Load Balancer。
3. 交易管理員會處理傳送至大型主機的交易。
4. Network Load Balancer 會將交易分配到可用的轉換環境以進行處理。
5. AWS Mainframe Modernization 轉換環境會處理請求。

最佳實務

本節概述了一組最佳實務，用於解決將大型主機工作負載轉換為雲端環境時的主要挑戰，同時將資料庫保留在 z/OS 的 Db2 上。

網路延遲

若要準確預測在複寫工作期間將應用程式與 Db2 資料庫分開的延遲影響，我們建議您對交易和批次程序的 Db2 呼叫數量進行徹底的評估。此評估應使用追蹤資料完成，並應包含下列步驟：

- 收集追蹤資料：收集代表性交易和批次任務的詳細追蹤，並確保追蹤擷取所有 Db2 互動，包括項目和結束。
- 分析追蹤資料：計算每個交易和批次任務的 Db2 項目數和結束數，並計算每個交易和批次程序的平均 Db2 互動數。
- 測量目前的回應時間：檢查 Db2 存取是否符合應用程式的服務層級協議 (SLA)。
- 估計網路延遲：判斷已轉譯應用程式與 Db2 資料庫之間的預期網路延遲。考慮實體距離、網路基礎設施和潛在瓶頸等因素。
- 計算潛在影響：針對每個交易和批次程序，將 Db2 項目的數量乘以預估的網路延遲。將此計算時間新增至目前的回應時間，以預測新的總處理時間。
- 評估結果：評估預測的延遲增加是否適合您的業務需求，並識別可能需要最佳化或重新設計以緩解延遲問題的任何交易或程序。
- 考慮緩解策略：探索連線集區、快取或批次資料擷取等選項，以減少個別 Db2 互動的數量。評估將經常存取的資料移近應用程式層的可能性。

透過遵循這些步驟，您將能夠對轉換策略的可行性進行資料驅動型決策，並在影響您的生產環境之前識別任何潛在的效能問題。此方法有助於確保順利轉換，同時維持資料庫相依應用程式的可接受效能等級。

安全性

- 保護您的應用程式建置：使用虛擬私有雲端 (VPC) 中的私有子網路來執行 AWS CodeBuild ，以協助確保隔離和增強的安全性。從 CodeBuild 子網路 CIDR 實作 Db2 信任的內容，以便在建置過程中安全存取資料庫。
- 保護您的執行期環境：使用來自執行期子網路 CIDR 的 Db2 信任內容進行安全資料庫連線。

- 安全地管理資料庫登入資料：實作定期登入資料輪換排程，將未經授權的存取風險降至最低。安全地存放 Db2 登入資料 AWS Secrets Manager。
- 建立網路安全：實作強大的網路分割和防火牆規則，以保護建置和執行期環境。使用正確的 AWS Direct Connect 和 組合 AWS Site-to-Site VPN ，以達到必要的安全層級。
- 強制加密：對應用程式與 z/OS 的 Db2 之間傳輸中的資料強制加密。

應用程式控管

- 建立事實來源：建立新的軟體組態管理 (SCM)，例如 GitHub，做為已遷移應用程式程式碼的單一事實來源。這可確保一致性，並消除轉換期間雲端和大型主機環境之間的版本差異。
- 更新變更管理程序：更新變更管理程序，以考慮在此新的雙環境範式中修改程式碼。此程序應包括：
  - 清除程式碼變更的核准工作流程。
  - 將程式碼合併至主分支之前，必須檢閱程式碼。
  - 同步部署程序，以確保兩個環境同時接收更新。
  - 在任一環境中發生問題時的轉返機制。

彈性

雲端運算的彈性引入了範例轉移，可大幅改變大型主機的成本結構和資源管理。與具有固定容量和尖峰定價模型的傳統大型主機環境不同，雲端平台提供動態可擴展性和pay-as-you-go的方法，可能會導致大幅節省成本並改善營運效率。

在雲端環境中，組織可以根據實際需求即時擴展或縮減運算資源，因此不需要過度佈建來容納尖峰負載。這種彈性允許企業只為他們使用的資源付費，而不是投資昂貴的硬體和軟體授權，來處理偶爾的用量激增。

如需定價運作方式的詳細資訊 AWS，請參閱 AWS 定價。

後續步驟

大型主機現代化是一項複雜且重要的計畫，需要專業知識和進階解決方案。您可以加速現代化程序，並透過策略合作夥伴關係實現更快的業務成果，協助您完成下列任務：

- 評估和排定優先順序：檢閱您的大型主機應用程式，並識別哪些應用程式適合進行轉換，同時將資料庫保留在 z/OS 的 Db2 上。考慮複雜性、業務重要性和潛在投資報酬率 (ROI) 等因素。
- 制定遷移策略：建立詳細的計劃以複寫您選取的應用程式，包括時間表、資源配置和風險緩解策略。
- 評估工具和技術：研究並選取適當的工具和技術，以促進轉換程序，例如應用程式現代化平台或程式碼轉換工具。
- 與專家互動：考慮與大型主機現代化專家或具有轉換專案經驗的諮詢公司合作。
- 概念驗證：從小規模的概念驗證開始，以驗證您的方法，並在擴展到更大的應用程式之前識別潛在的挑戰。
- 測試和驗證：制定全面的測試策略，以確保您已轉換的應用程式正常運作，並維護與現有 Db2 for z/OS 資料庫的資料完整性。
- 訓練和知識轉移：為您的團隊做好準備，為新環境做好準備，方法是為經過修改的應用程式和引進的任何新工具或技術提供訓練。
- 階段式實作：考慮階段式的轉換方法，您可以在其中逐步遷移應用程式，同時監控效能並解決出現的任何問題。
- 持續最佳化：在轉換後，持續監控和最佳化應用程式的效能，以及其與 z/OS 資料庫 Db2 的互動，以確保長期成功。
- 依您的步調進行現代化：現在工作負載正在執行 AWS 並已利用雲端，請開始規劃現代化的重新構想階段。

Resources

如需大型主機遷移和現代化的詳細資訊，請參閱下列資源。

AWS 文件

- 將 Amazon Route 53 設定為 DNS 服務
- 將流量路由到 ELB 負載平衡器
- 加權路由
- 使用 Rocket 軟體來複寫應用程式

Rocket 軟體參考

- Micro Focus 外部通話界面 (ECI)
- CICS Web 服務

IBM 參考

- 信任的內容 （適用於 z/OS 文件的 IBM Db2)

工具

- Rocket Enterprise 伺服器

AWS 規範性指引模式和指南

- 使用 AWS Mainframe Modernization 和 建置 COBOL Db2 程式 AWS CodeBuild
- 的 DevOps AWS Mainframe Modernization
- 大型主機現代化：遷移應用程式程式碼的解耦模式
- 使用信任的內容， AWS 在 上保護和簡化 Db2 聯合資料庫中的使用者存取

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 5 月 7 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)

當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 Rs。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

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 Rs。

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？ Secrets Manager 文件中的 。

依設計的安全性

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)

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標 (SLO)

代表服務運作狀態的目標指標，由服務層級指標測量。

共同責任模式

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

SIEM

請參閱安全資訊和事件管理系統。

單一故障點 (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，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

危及系統安全性的軟體或硬體瑕疵。

W

暖快取

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

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等速度的查詢。

視窗函數

SQL 函數，對與目前記錄在某種程度上相關的資料列群組執行計算。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

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

WORM

請參閱寫入一次，讀取許多。

WQF

請參閱AWS 工作負載資格架構。

寫入一次，讀取許多 (WORM)

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瑕疵或漏洞。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。