aws

投資混沌工程作為策略必要性

# AWS 方案指引

# AWS 方案指引: 投資混沌工程作為策略必要性

# Table of Contents

# 投資混沌工程作為策略必要性

Adrian Hornsby，Amazon Web Services

2025 年 1 月 (文件歷史記錄)

混沌工程實務使用受控中斷來識別系統問題和機會，以防止中斷和其他事件。混沌工程已成為改善彈性系統的必要條件，但普遍採用面臨著誤解、文化阻力、資源以及如何量化商業價值的障礙。設定初始目標有助於啟動混沌工程工作，同時量化投資報酬率 (ROI) 可證明持續投資是合理的，尤其是在經濟壓力下。

此策略文件概述了擷取量化操作改進和定性組織利益的整體方法。最終目標是將混沌工程視為網路安全的策略必要性，而不是持續的成本調整練習。

# 停機時間成本和混沌工程的出現

資訊技術情報諮詢 (ITIC) 估計，90% 的企業面臨每小時停機時間超過 300，000 USD 的成本，其中 41% 超過每小時 1 到 500 萬美元。除了立即損失收入之外，停機時間還可能導致長期問題，包括合規失敗、降低股票價格、大幅緩解成本，甚至是品牌受損。

雖然停機時間通常與產生收入的線上系統相關聯，但負面影響遠遠超出此範圍。所有大型企業和組織，無論其主要收入模式為何，都非常依賴其內部系統的可用性，例如人力資源和薪資。

影響這些核心內部服務的停機時間可能會抑制公司運作的能力，導致嚴重的營運中斷和財務影響。產生的問題可能包括下列項目：

- 支付員工和廠商費用的延遲
- 無法處理客戶訂單或交易
- 洩露安全系統允許的敏感資料洩露
- 生產力和營收機會的損失
- 不合規的法規處罰
- 損害品牌評價

混沌工程刻意引入受控制的中斷。使用混沌工程來了解或驗證系統對損害的回應，已成為改善系統彈性的關鍵實務。混沌工程可讓您的組織主動發現問題、驗證彈性機制，最終降低意外停機時間及其相關成本的風險。混沌工程的優點包括：

- 暴露技術債務
- 運動操作性肌肉
- 建立對系統的信心
- 識別故障點
- 改善監控和可觀測性
- 支援以實驗為基礎的學習
- 提供改善的彈性，以減少停機時間

隨著系統變得越來越複雜且客戶期望增加，混沌工程變得越來越重要。Gartner 建議將混沌工程作為組織減少意外停機時間並改善彈性的關鍵實務。

# 混沌工程的採用挑戰

雖然混沌工程是改善系統彈性越來越重要的實務，但採用混沌工程可能會面臨下列障礙：

- 有關風險的誤解 – 常見的誤解是，混亂工程僅在生產環境中進行，這會導致對過度風險的擔憂。這種感知源於對混沌工程實務的系統性和控制性質缺乏理解。如 AWS Well-Architected Framework 中所述，請先在非生產環境中執行錯誤模擬。
- 更長期的商業價值 – 混沌工程的好處會逐漸累積，因此難以量化商業價值並證明初始投資的合理性。較慢的 ROI 讓組織難以排定優先順序並堅持混沌工程。
- 技能和專業知識差距 – 混沌工程需要一組獨特的技能和專業知識，這些技能和專業知識可能在您的組織中不易取得。建置或取得此專業知識可能是一個重大的障礙，尤其是對剛接觸實務和資源有限的組織而言。

此策略文件的其餘部分將主要著重於第二個挑戰，也就是展現混沌工程的商業價值。

# 混沌工程的累積效果

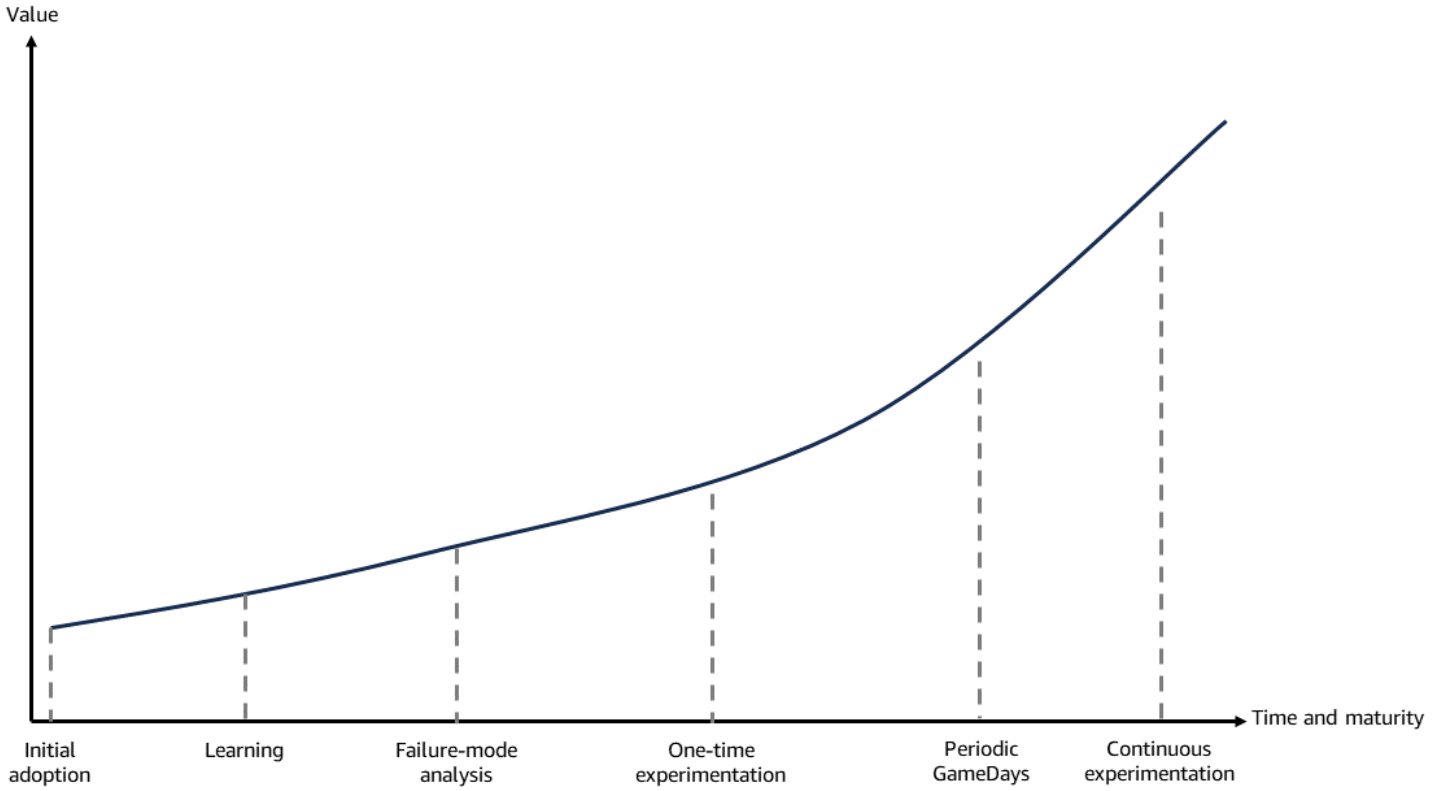
與具有明確定義開始和結束日期的傳統技術專案不同，混沌工程是持續學習和持續改善系統彈性的實務。隨著時間的推移，混沌工程複合的優勢。

隨著系統不斷發展和成長，新的故障模式也會出現。需要更多混沌實驗來識別潛在問題。修正問題可能需要數月的時間，特別是在系統和程序複雜的大型企業中，或當錯誤由外部服務供應商所擁有時。

將失敗作為學習和改進機會的文化轉移會隨著時間的推移而增長，並在組織中變得深厚。在自動化混沌工程實驗和開發支援工具方面的投資，會繼續簡化和增強混沌工程實務。建立這種機構知識並了解系統彈性是隨著時間累積的逐步程序。隨著實務隨著不斷發展的系統而成熟，透過混沌工程開發的知識、程序和工具會提高價值。

下圖顯示當混沌採用進行下列階段時，值如何隨時間增加：

- 初始採用
- 學習
- 失敗模式分析
- 一次性實驗
- 定期 GameDays
- 持續實驗




如圖所示，混沌工程的優點通常會在將任何錯誤注入系統之前開始。規劃和設計混沌實驗本身的過程可提供立即價值。識別潛在的故障案例、單一故障點和系統中的不確定性區域會導致改善。

例如，寫下失敗案例並討論潛在的層疊效果，稱為失敗模式和效果分析 (FMEA) 的程序，有助於發現可能忽略的明顯弱點或差距。您的組織可以主動解決這些問題，即使在系統受到任何故意中斷之前也是如此。如需詳細資訊，請參閱彈性分析架構。

此外，系統可觀測性和監控的更多關注，通常會伴隨混沌的工程計畫立即開始帶來好處。改善對系統行為和故障模式的可見性，有助於團隊進一步了解系統的正常操作條件。更高的可見性也有助於團隊了解操作條件在推送至限制時如何降低、調整和失敗。

與連續實驗模式相比，一次性實驗和定期 GameDay 模式都是更多手動方法。他們需要更親自動手和探索性的程序，讓工程師透過觀察和實驗主動塑造和精簡假設。

另一方面，連續實驗模式本質上是自動化的。此模式著重於以受控制且反覆的方式執行已核准和已驗證的假設。它透過專用混沌管道在開發過程中使用自動化和整合，以協助確保一致且可重複的實驗。

# Grassroots 混沌工程計畫

混沌工程之旅通常從基層開始，其中工程團隊識別需求，並開始獨立實驗混沌工程。

在此基層方法中，團隊會實驗、學習和改善他們的混沌工程實務。混沌工程的值可以透過以下實際結果來示範：

- 減少事件
- 更好的可觀測性
- 更快的復原時間
- 改善和持續的系統彈性

Grassroots 混沌工程計畫通常會在特定組織條件下出現。他們需要具有高度工程自主性的環境，其中團隊可以自由實驗和創新，而不會有過多的官僚障礙。彈性工程或分散式系統的在地專業知識至關重要，因為它為了解和實作混沌實驗提供了技術基礎。最重要的是，這些計畫通常依賴混沌擁護者 - 熱衷於了解混沌工程價值的個人。混沌倡導者願意倡導採用混沌工程、教育他們的同儕，並推動初始實驗。如果沒有組織自由、技術專業知識和積極的擁護者，基層混沌工程工作很少取得根源，無論其潛在優勢。

# 混沌工程採用中目標的角色

初始目標通常從組織中的基層混沌工程工作中有機地出現。由於需要解決自己的重複性問題，這些團隊或群組通常會探索混沌工程實務，而無需明確核准或排定更高層級的優先順序。

團隊可以使用這些結果來建立吸引人的案例，以實現更廣泛的組織採用，有效地成為其他團隊的證明據點。

在基層工作的好處變得太重要而無法忽略之後，這些團隊可以將他們的工作和知識提升到領導力並設定目標。這種提高的可見性可以促進整個組織的彈性目標的採用，並導致對混沌工程實作所需的支援和資源。

目標，特別是那些由領導階層推動並為了回應重大中斷而建立的目標，在促進採用混沌工程實務方面扮演關鍵角色。常見的目標類型包括下列項目：

- 識別和減少單點故障 (SPOF) 的可用性目標
- 改善從中斷或故障中復原的能力的服務復原目標
- 滿足特定服務水準目標 (SLOs) 的使用者體驗目標
- 指標驅動的目標，以追蹤緩解已知可用性風險及實作建議彈性措施的進度
- 展現營運彈性的法規和合規目標

如需某些目標類型以及 Amazon 和其他組織如何在採用混沌工程期間使用目標的詳細資訊，請參閱附錄 A。

這些目標可做為令人信服的理由，並提供目標性、可行的方法，以推動混沌工程採用。一開始，目標可做為傳統 ROI 指標的代理。當可量化彈性 ROI 計算可能難以獲得時，目標提供令人信服的理由。如果採用初期沒有此類目標，混沌工程實務會面臨無法證明其有效性並獲得更廣泛的組織接受的風險。

# 從目標到衡量 ROI 的轉移

隨著實務的成熟和初始目標的實現，重點最終會從設定目標轉向量化混沌工程的有形經濟效益 - 投資報酬率 (ROI)。輪班來自兩個主要原因：

- 經濟考量
- 保留客戶體驗和信任

## 經濟考量

在經濟成長和財務良好的情況下，公司通常不需要廣泛的理由來設定混沌工程策略的特定目標。不過，金融環境的變化已導致許多組織重新評估其投資，而混沌工程實作需要提供量化的 ROI。

這些公司現在的任務是設定明確的傳統 ROI 指標，以展示混沌工程實務的價值和影響。此挑戰會因預防矛盾而更加複雜。當成功預防事件導致難以合理化投資時，就會發生預防矛盾，因為利益相關者往往會低估避免的災難。即使組織具有深厚的卓越營運文化，也會面臨使用 ROI 指標來證明持續採用混沌工程的壓力。

## 保留客戶體驗和信任

長期而言，維持目標驅動的彈性可能具有挑戰性。在達到復原時間目標等初始目標之後，直到下一次重大中斷之前，持續進行混沌工程投資會變得困難。投資的流程和 ebb 會建立被動的鋸齒週期。對於每次新的中斷，投資彈性尖峰，並以新的目標解決根本原因。達到新目標後，投資會下降，直到下一個事件，重新啟動被動迴圈。

驅動這種被動方法的中斷會對客戶產生負面影響。關鍵問題：客戶在捨棄服務供應商之前，會容忍多少次重大中斷，以支持更具彈性的競爭對手？

# 量化混沌工程的 ROI

目前，發佈的資源很少提供全面的方法或真實世界資料，用於量化混沌工程的長期投資報酬率 (ROI)。

在白皮書 The Business Case for Chaos Engineering 中，Netflix 提供寶貴的方程式來計算混沌工程的 ROI。此方程式為開始混沌工程之旅的組織提供了起點。

方程式要求您準確估計下列項目的成本：

- 可預防和不可預防的中斷
- 混沌工程計畫實作成本
- 混沌引發的傷害成本

混沌引發的傷害是指在混沌工程實驗中，刻意將故障或亂流條件注入系統所造成的負面影響或中斷。此方程式需要估算可預防和不可預防中斷的成本、混沌工程計畫實作成本，以及混沌引發的傷害成本。

確定混沌工程計畫可以預防哪些問題是一項困難的任務。它需要一個假設分析，涉及查看問題的根本原因，並推測混沌工程實驗如何有助於識別它們。這種分析具有挑戰性，因為現代系統非常複雜，在各種元件、服務和第三方程式庫之間具有許多相互依存關係和互動。此外，系統中的故障通常是非確定性的，在事後可能很難完全了解導致故障的條件。

雖然 Netflix 建議的方法有一些限制，但對於開始探索混沌工程的組織來說，它是很好的基礎。方程式可以引導您估算成本和潛在利益，這可協助您做出實作此類程式的決策。不過，隨著組織在混沌工程旅程中的進展，擴展 ROI 評估以納入更全面的觀點非常重要。

這種整體方法不僅會擷取減少中斷和工程成本的直接好處，也會強調對組織整體恢復能力的長期轉型效果。它擷取了混沌工程的複合優點和更廣泛的組織效果，以更準確地呈現混沌工程的真實價值和影響。

## ROI 量化的整體方法

整體投資報酬率評估不僅必須考慮 量化指標，還必須考慮定性因素。整體方法需要來自組織的實際資料，這些組織會在較長的時間內大規模執行混沌工程。您可以透過收集的任何方程式方法 ROI 資料，使用從基層專案和目標開始的資料。

量化指標著重於數量或頻率。測量結果是客觀的，而且可以進行統計分析。範例包括問卷、實驗和資料分析。量化措施可以包括下列項目：

- 事件指標

- 成本
- 改善項目
- 合規
- 採用率
- 客戶滿意度

追蹤量化措施可以證明混沌工程的直接操作優勢。

定性措施是描述性的，專注於了解經驗和意見。它們通常是主觀的，而且無法輕鬆地以數字測量。對於混沌工程，定性措施會擷取更廣泛的組織影響。定性措施可以包含下列項目：

- 員工信心
- 文化轉移
- 協作
- 訓練有效性
- 人才保留
- 品牌評價
- 競爭優勢

透過同時考慮量化財務影響和定性組織利益，您可以針對持續的混沌工程投資做出更明智的決策，同時培養彈性文化。

如需這些措施及其相關事件分類架構的詳細資訊，請參閱附錄 B 和附錄 C。

# 將投資報酬率轉換為混沌工程作為策略必要性

雖然監控 ROI 很吸引人，但衡量混沌工程價值的挑戰通常會導致組織優先考慮立即的短期效率，而不是策略彈性投資。這種方法會將混沌工程忽略為彈性的關鍵驅動因素，以及避免中斷的競爭優勢。混沌工程的實際價值是防止未來的故障。Chaos 工程支援長期業務持續性。

不專注於投資報酬率，而是將混沌工程視為網路安全。如 Forbes 文章網路安全作為策略投資中所說明：ROI 最佳化如何導致更安全的未來，網路安全不應被視為組織的成本中心或強制性費用，因為這種思維無法識別強大的網路安全措施可以隨著時間提供的策略價值。相反地，作者認為透過轉移觀點將網路安全視為推動競爭優勢的長期投資，組織可以在其各自市場內釋放創新、營運效率和差異性的新途徑。透過採用這種方法，作者得出的結論是，資訊安全長 (CISOs) 可以更好地保護領導層的參與和資金。然後，他們可以將公司定位在風險越來越高的網路環境中超越競爭對手。這種長期的網路安全策略價值建立與混沌工程實務中固有的持續改善平行。

雖然安全保護組織操作和保護資產的能力，混沌工程有助於確保核心系統和服務的可用性、可靠性和可復原性。為了實現長期價值和競爭優勢，請將混沌工程視為核心功能和策略必要性，而不是需要持續證明的倡議。

下圖顯示混沌工程從草根演變到目標和投資報酬率，成為策略。


在基層，個別團隊通常會根據當地需求獨立實驗。這些實驗由熱衷的工程師擁護，他們透過減少事件並改善可觀測性來展現價值。

當這些工作成功時，團隊可以將其學習提升為領導力。透過這種可見性，努力會轉換為目標驅動的階段。組織設定恢復能力和復原的正式目標，以資源和對更廣泛的實作的支援為後盾。

最後，混沌工程不僅需要持續的 ROI 理由才能被視為策略必要性，與網路安全類似。在這個階段，混沌工程會完全整合到組織程序中。實作著重於長期彈性，而非短期指標。混沌工程被視為維持競爭優勢和客戶信任的重要核心功能。

## 將混沌工程整合至您的組織

若要將混沌工程提升到與安全性相同的重要性層級，請考慮下列建議：

- 將混沌工程建立為不可協商的實務 – 就像網路安全被視為組織的基本需求一樣，將混沌工程視為確保系統彈性和可靠性的必要實務。將混沌工程整合到組織的程序、工具和文化中，而不是將其視為選擇性或選擇性活動。如需詳細資訊，請參閱彈性生命週期架構指南。
- 安全執行層級的接受和支援 – 與安全計畫一樣，混沌工程工作必須獲得執行領導層的接受和主動支援。這包括配置專用資源、預算和人員，以在整個組織中實作和維持混沌工程實務。
- 實作控管和監督 – 與 CISO 和安全控管架構類似，請建立專用的混沌工程團隊或首席應變主管。此團隊或角色負責監督和協調不同團隊和業務單位的混沌工程工作。
- 將混沌工程整合到開發和操作週期中，就像將安全實務整合到軟體開發和部署程序中一樣，讓混沌工程成為軟體開發和交付生命週期的無縫部分。
- 定期進行混沌工程演練和模擬 – 類似於安全漏洞模擬和事件回應演練，定期進行混沌工程實驗，以驗證事件回應功能並主動識別潛在的盲點。
- 使用混沌工程來維護 Runbook – 如同執行安全性審查一樣，使用混沌工程實驗來驗證 Runbook 的有效性和準確性，以回應事件和復原。此外，混沌工程實驗可作為逼真的模擬，讓待命工程師練習執行 Runbook 程序。模擬可協助工程師維護操作的肌肉記憶體，以及處理真實世界事件的準備程度。
- 培養彈性文化 – 如同安全意識訓練一樣，投資於混沌工程教育和知識分享計畫，以培養彈性文化。包括訓練計畫、跨職能合作，以及針對採用混沌工程實務之團隊的獎勵。
- 測量和報告彈性指標 – 定期監控彈性指標，並向利益相關者報告。使用本文件中討論的量化和定性指標作為起點。
- 將彈性視為競爭優勢 – 網路安全措施可提供競爭優勢。同樣地，請將您的混沌工程和彈性功能視為差異化因素，協助您為客戶提供更可靠且值得信賴的服務。

## 取得高階主管的支持

混沌工程通常缺乏 C-suite 傳統責任中的明確擁有者。CEO 關心成長、獲利能力和市場領導力。CFO 專注於財務績效、成本控制和風險管理。CTO 優先考慮技術策略、產品藍圖和卓越工程。CISO 會監督安全與合規。

由於沒有真正擁有彈性的單一主管，通常很難獲得接受和支援。然而，系統故障會影響營收、客戶滿意度和品牌評價，這是 CEO 和 CFO 的考量。CTO 和 CISO 的任務是實作彈性措施，但可能缺乏組織要求。這種模棱兩可的情況可能會阻礙進行策略投資，並使組織符合常見的彈性策略。

這種模棱兩可的情況也使得獲得高管對諸如混沌工程等恢復能力計劃的認可變得具有挑戰性。畢竟，C 級領導者正在處理許多策略優先順序：成長、創新、客戶體驗、合規等。

若要有效地將混沌工程的價值傳達給 C 級主管，請考慮下列方法：

- 判斷高階主管的主要考量和決策驅動因素。

  例如，高階主管是否擔心客戶流失、法規遵循、降低成本或競爭壓力？ 將混沌工程定位為力乘數，以符合公司的獨特挑戰和目標。
- 識別共同目標和策略成果。

  您的混沌工程策略如何支援整個組織的成長策略、客戶體驗、市場機會和營運效率？ 根據目標、業務影響、ROI 和不執行措施的風險來排定措施的優先順序。
- 使用關鍵彈性指標，以可量化的詞彙傳達混沌工程策略的有效性。

  從這四個關鍵彈性指標開始：可用性、偵測時間、回應時間，以及復原時間。將這些直接連結到業務成果，例如營收、節省成本和品牌評價。
- 請勿在技術詳細資訊中遺失。

  專注於整體情緒和可衡量的業務影響。C-suite 關心推動成長、增強客戶信任和促進創新的結果。

# 預防悖論

當錯誤在資訊清單之前成功緩解時，說服利益相關者了解所採取預防措施的價值和必要性會變得具有挑戰性。此現象稱為預防矛盾。預防矛盾是將混沌工程整合為策略必要性的最大障礙，源自於人類認知中的固有偏差。

Y2K 錯誤可做為此悖論的絕佳說明。數年的準備和數十億美元投入於更新全球的電腦系統。不過，許多 對 2000 年的順利轉換被解釋為 Y2K 問題過度膨脹本質的證明。很少發現預防性工作的成功。

這種預防矛盾繼續挑戰現今投資於混沌工程的組織。當透過主動措施成功避免潛在的中斷時，完全沒有災難可能會使用於預防的資源難以合理化。

此現象的根本原因在於我們思維處理資訊的方式。人類認知過程旨在回應和記住實際事件和可見結果。防止災難時，沒有可保留或共用的戲劇性敘述。預防矛盾的另一個層面是後視偏差。在事件發生後，個人傾向於做出沒有發生什麼事的結論，因此這不是真正的問題。無法識別防止實際問題的適當預防措施的可能性。這個心理盲點為組織帶來了永久的挑戰。您越成功預防和恢復能力，在回顧方面，您的工作就越不必要。

為了解決預防矛盾問題，您的組織可以採取特定步驟，讓預防的隱形工作可見、可測量且受到重視。可能的步驟包括下列項目：

- 記錄並模擬在沒有預防措施的情況下可能發生的情況。
- 分享預防性措施避免潛在災難的事件案例。

- 指向未準備且因此遭受後果的對等組織。
- 在潛在影響的情況下呈現預防成本。
- 將預防工作細分為可見的里程碑和成就。
- 針對預防措施的存在原因及其歷史重要性，建立機構記憶體。
- 定期向利益相關者說明彈性和混沌工程實務的價值。

# 結論

混沌工程是組織的策略必要條件。雖然您的採用旅程可能面臨錯誤概念、文化阻力和資源限制等挑戰，但建立明確、以領導為導向的目標可能會引發程序。隨著實務成熟，透過擷取量化營運改善和定性組織利益的整體方法量化投資報酬率。整體方法在經濟壓力期間尤其重要。

若要將此策略必要性轉換為現實，請先評估組織目前的成熟度層級。您的組織是處於基層實驗階段、目標驅動階段還是介於兩者之間？ 根據此評估，建立量身打造的藍圖，以完成下列作業：

- 建立混沌工程控管（例如，指派首席恢復官）。
- 將混沌實務整合至開發工作流程。
- 實作定期訓練計劃。
- 開發全面的彈性指標。

此轉換不會在夜間進行。不過，採取這些具體步驟，同時確保持續的執行支援，將有助於將混沌工程提升到與網路安全相同的策略層級。與網路安全類似，混沌工程可能會成為組織營運 DNA 和程序不可或缺的一部分。

# 資源

- ITIC 2021 全球伺服器硬體、伺服器作業系統可靠性調查結果
- 混沌工程的商業案例
- 作為策略投資的網路安全：ROI 最佳化如何帶來更安全的未來
- I&O 領導者的混沌工程指南
- 如何使用 AWS Resilience Hub 分數
- 使用 AWS Resilience Hub 主控台實作建議的實驗

# 附錄 A – 混沌工程的目標類型

以下目標類型的描述包括 Amazon 和其他組織如何設計混沌工程目標的實際範例。

## 彈性架構目標

採用混沌工程的初始驅動因素之一是識別和減少跨系統和基礎設施的單點故障 (SPOF)。目標設定為驗證關鍵系統和架構的彈性，特別是新服務或應用程式。

彈性架構目標涉及執行混沌實驗，以模擬服務相依性的失敗。實驗會確認逾時、重試、快取行為和斷路器組態是否正常運作。這些實驗有助於發現修復問題，防止影響客戶的事件。如需範例，請參閱使用混沌工程在 Prime Video 中建置彈性服務。

## 服務復原目標

服務復原目標著重於改善從營運中斷或基礎設施故障復原的能力。例如，您的組織可能旨在在發生中斷時，為您的核心服務實現特定的復原時間目標 (RTO)。團隊可以設計混沌實驗，以驗證和最佳化疏散策略、容錯移轉機制和自動化復原程序。最佳化最終會減少服務還原所需的時間。如需範例，請參閱 AWS Lambda：under-the-hood的恢復能力。

## 使用者體驗目標

維持一致且可靠的使用者體驗至關重要，特別是在高流量期間或關鍵事件期間。在這種情況下，設定以特定服務層級目標 (SLOs) 為中心的目標。這種以客戶為中心的方法可確保彈性工作直接與提供卓越的使用者體驗保持一致，即使在遇到故障或條件降低的情況下也是如此。如需範例，請參閱工程彈性：Amazon Search 的混沌工程之旅中的課程。

## 指標驅動的目標

您可以根據量化指標建立目標，例如彈性分數，該分數的計算方式為將點授予採用經驗證的彈性最佳實務的服務。然後，您可以使用特定的混沌實驗來判斷彈性分數。此分數可以做為衡量團隊在緩解已知可用性風險以及實作建議彈性衡量指標方面追蹤進度的衡量標準。不過，請務必謹慎解譯此類分數，並避免過度強調單一指標，而犧牲更廣泛的彈性目標。如需範例，請參閱了解彈性分數。

# 法規合規目標

金融服務業已成為接受混沌工程的先鋒，主要是由要求強大彈性的嚴格法規要求所推動。法規將要求金融機構主動識別、測試和修復其關鍵系統和程序中的漏洞。這些法規包括下列項目：

- 強化美國聯邦機構所發行之營運恢復能力的健全運作實務相關文章
- 歐洲中央銀行的營運彈性指導方針
- 歐盟委員會提出的數位營運彈性法案 (DORA) 提案

如果您的組織是金融機構，請設定透過全面測試和驗證策略展現營運彈性的明確目標，以遵守這些法規。如需範例，請參閱 London Stock Exchange Group 在 上使用混沌工程 AWS 來改善彈性。

# 附錄 B – 量化和定性措施

本節概述量化指標，以追蹤營運改進和定性措施，以評估混沌工程實務中更廣泛的組織結果。

## 量化措施

下列量化措施提供追蹤關鍵指標的架構，可示範透過混沌工程實務實現的直接事件和操作改進：

- 事件：
  - 事件頻率 – 追蹤事件分類架構中的事件數量，並根據其在一段時間內的重要性（關鍵、主要、次要）進行分類。如需事件分類架構的詳細資訊，請參閱附錄 C。
  - 停機時間和降級 – 測量每個事件分類的停機時間或服務降級總持續時間。
  - 事件回應指標 – 若要了解事件、測量偵測時間、識別時間、緩解時間、復原時間、呈報時間，以及每個事件分類的其他相關指標。
  - 影響客戶的事件 – 追蹤影響客戶的事件數量，或在影響客戶之前包含的事件百分比。
  - Runbook 變更 – 追蹤從混沌實驗中獲得的洞見所產生的 Runbook 更新或修訂數量。Runbook 提供執行特定操作或程序以從特定類型的事件復原的詳細說明。
- 成本：
  - 基礎設施成本 – 收集基礎設施成本的資料，包括雲端運算資源以及改善彈性所採取動作所需的備援措施。
  - 客戶影響 – 測量與系統故障或停機時間相關的客戶體驗、流失率和收入損失的影響。
  - 員工生產力 – 追蹤工程和營運團隊在事件回應、消防、撰寫事後事件和其他與系統故障相關的被動性任務所花費的時間。
- 持續的系統改進 – 計算因混沌實驗洞察而實作的程序改進、架構變更或自動化復原機制的數量。
- 合規 – 追蹤成本並努力滿足與營運彈性相關的法規要求或產業標準。
- 採用 – 追蹤整個組織的混沌實務採用率。
- 客戶滿意度 – 測量客戶滿意度指標的變更，以衡量改善的系統可靠性如何影響業務。

## 定性措施

以下定性措施提供一個架構，用於追蹤透過混沌工程實務實現的更廣泛的組織成果：

- 員工信心和準備：

  - 調查團隊會定期測量他們在處理真實世界事件時的可信度，以及他們對隨需輪換的感知準備程度。
  - 追蹤參與混沌實驗的待命工程師百分比，作為培訓的一部分。
- 文化轉移：
  - 評估彈性思維透過問卷、意見回饋工作階段或稽核滲透組織的程度。
  - 追蹤積極擁護和倡導混沌工程實務的團隊數量。
- 跨功能協作和知識分享：
  - 追蹤與混沌工程學習相關的跨團隊知識分享工作階段或研討會的頻率和出席率。
  - 追蹤涉及多個團隊或部門的關節混沌工程計畫數量。
- 訓練有效性：
  - 透過執行訓練後問卷或評估，評估混沌工程訓練計畫的有效性。
  - 追蹤參與混沌工程訓練計畫和閱讀事後文章的工程師人數。
- 人才吸引和保留：
  - 評估混沌工程計畫是否透過減少修復中斷所花費的時間和精力，來協助吸引和保留頂尖工程人才。
- 品牌評價：
  - 追蹤與組織對營運彈性所展現承諾相關的任何品牌感知或評價變化。
- 競爭優勢：
  - 在系統可用性方面追蹤相較於業界同儕的競爭優勢。

# 附錄 C – 事件分類

在分類架構中追蹤事件至關重要，因為該架構提供對影響系統的故障類型和問題的整體檢視。如果您的組織只追蹤單一類別中的事件，例如基礎設施故障，您可能會錯過其他領域的洞見和改進機會。透過追蹤跨多個類別的事件，您可以更好地了解要執行的各種混沌實驗。此觀點有助於識別潛在的盲點，並支援工程範圍的擴展，這會導致更具彈性和容錯能力的系統。

建議的事件分類架構旨在協助根據事件的性質和潛在影響來分類事件。它使用高階分類，將事件分組為八個主要類別：

- 部署問題：
  - 失敗的部署
  - 回復失敗
  - 部署期間的組態問題
- 軟體錯誤和迴歸：
  - 功能錯誤
  - 整合問題
  - 效能問題
  - 配額問題
  - 彈性機制問題（重試、逾時）
  - 資料完整性問題
- 測試問題：
  - 缺少測試
  - 無效測試
  - Flaky 測試
- 基礎設施故障：
  - 硬體故障（伺服器、網路裝置、儲存）
  - 擴展問題
  - 相依性失敗（第三方服務、APIs)
  - 網路連線問題
- 操作問題：
  - 人為錯誤（設定錯誤、意外變更）

  - 監控和提醒失敗
  - 容量規劃問題
  - 備份和還原失敗
- 安全事件：
  - 未經授權的存取嘗試
  - 資料外洩
  - 拒絕服務 (DoS) 攻擊
- 第三方服務中斷：
  - 雲端供應商中斷
  - DNS 失敗
  - 外部 API 和服務中斷
- 環境因素：
  - 自然災難（地震、火災、洪水、停電）
  - 與天氣相關的問題

這是非明確的範例分類架構，您可以量身打造以符合您的特定需求和組織。我們建議您隨著系統演進或新類型的事件出現，定期檢閱和更新分類架構。

# 文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 RSS 摘要。

| 變更 | 描述 | 日期 |
| --- | --- | --- |
| 初次出版 | — | 2025 年 1 月 28 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

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，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。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

在資料集中永久刪除個人資訊的程序。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

經常用於重複性問題的解決方案，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

您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，例如在網際網路上為資訊編製索引的 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，以及透過核准的程序，讓使用者能夠快速存取他們通常無權存取 AWS 帳戶 的 。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。

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

這些階段由 Stephen Orban 於部落格文章 The Journey Toward Cloud-First 和 Enterprise Strategy 部落格上的採用階段中定義。 AWS 雲端 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。

部署

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)

組織之間商業文件的自動交換。如需詳細資訊，請參閱什麼是電子資料交換。

加密

一種運算程序，可將人類可讀取的純文字資料轉換為加密文字。

加密金鑰

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

端序

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

端點

請參閱 服務端點。

端點服務

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

企業資源規劃 (ERP)

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

信封加密

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

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

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

epic

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

ERP

請參閱企業資源規劃。

探索性資料分析 (EDA)

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

# F

事實資料表

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

快速失敗

一種使用頻繁和增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

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

功能分支

請參閱分支。

特徵

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

功能重要性

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

特徵轉換

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

少量擷取提示

在要求 LLM 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。對於需要特定格式、推理或網域知識的任務，少量的提示可以有效。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，以及自然語言的交談。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 工作流程會被視為舊版，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、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。

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部署最佳實務。

傳入 (輸入) VPC

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

增量遷移

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

工業 4.0

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

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

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

工業物聯網 (IIoT)

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

檢查 VPC

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

物聯網 (IoT)

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

可解釋性

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

IoT

請參閱物聯網。

IT 資訊庫 (ITIL)

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

IT 服務管理 (ITSM)

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

ITIL

請參閱 IT 資訊庫。

ITSM

請參閱 IT 服務管理。

# L

標籤型存取控制 (LBAC)

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

登陸區域

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

大型語言模型 (LLM)

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

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱標籤型存取控制。

最低權限

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

隨即轉移

請參閱 7 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、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。一個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)

一種 AWS 計畫，提供諮詢支援、訓練和服務，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，請參閱此詞彙表中的 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

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。如需詳細資訊，請參閱在微服務中啟用資料持久性。

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。

可程式邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

提示鏈結

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

擬匿名化

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

發佈/訂閱 (pub/sub)

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

# Q

查詢計劃

一系列步驟，如指示，用於存取 SQL 關聯式資料庫系統中的資料。

查詢計劃迴歸

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

# R

RACI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

RASCI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RCAC

請參閱資料列和資料欄存取控制。

僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

重新架構師

請參閱 7 Rs。

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)

服務中斷和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 Rs。

Replatform

請參閱 7 Rs。

回購

請參閱 7 Rs。

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

定義所有涉及遷移活動和雲端操作之各方的角色和責任的矩陣。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 Rs。

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。此功能可啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而無需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。

**SCADA**

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 文件中的 Secrets Manager 秘密中的什麼內容？。

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 AWS 服務 接收資料的 在其目的地加密資料。

服務控制政策 (SCP)

為 AWS Organizations中的組織的所有帳戶提供集中控制許可的政策。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的服務控制政策。

服務端點

的進入點 URL AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 AWS 服務 端點。

服務水準協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

服務層級指標 (SLI)

服務效能方面的測量，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。

單點故障 (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。如需詳細資訊，請參閱量化深度學習系統的不確定性指南。

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，通常可接受中等緩慢的查詢。

視窗函數

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

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

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

WORM

請參閱寫入一次，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，並防止資料遭到刪除或修改。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。