Unable to locate subtitle

AWS Well-Architected 架構

AWS Well-Architected 架構: ***Unable to locate subtitle***

Table of Contents

AWS Well-Architected Framework

出版日期： 2022 年 10 月 20 日 (文件修訂)

AWS Well-Architected 架構可協助您了解在 AWS 上建置系統時所做決策的優缺點。透過使用架構，您將了解在雲端設計和操作可靠、安全、有效率且經濟實惠的系統的架構最佳實務。

簡介

AWS Well-Architected 架構可協助您了解在 AWS 上建置系統時所做決策的優缺點。透過此架構，您將了解架構的最佳實務，以便在 AWS 雲端設計和操作安全、可靠、有效率、經濟實惠且永續的工作負載。它可讓您根據最佳實務一致地量測架構，並找出需要改進的方面。審查架構的程序是就架構決策進行的建設性對話，並非一種稽核機制。我們相信，擁有架構良好的系統可大幅提高企業成功的可能性。

AWS 解決方案架構師對於橫跨廣泛的各種垂直業務和使用案例建構解決方案，已累積多年經驗。我們已協助設計及審查數千套客戶在 AWS 上的架構。從這些經驗當中，我們已找出在雲端建構系統的最佳實務和核心策略。

AWS Well-Architected Framework 記錄一組基本問題，能讓您了解特定架構是否妥善符合雲端最佳實務的條件。該架構提供一致的方針，可依照您預計自現代雲端系統可獲得的品質來評估系統，並能得知欲達到此等品質會需要的修補措施。AWS 持續在演進當中，我們也不斷地從與客戶一同工作之中學到更多，因此架構完善的定義會始終精進下去。

本架構適用於擔任技術職務的人員，例如技術長 (CTO)、架構師、開發人員和營運團隊成員。內容說明設計及操作雲端工作負載時運用的 AWS 最佳實務和策略，並提供連結，可取得進一步實作的詳細資訊，和架構模式。如需詳細資訊，請參閱 AWS Well-Architected 首頁。

AWS 也免費提供您審查工作負載的服務。AWS Well-Architected AWS Well-Architected Tool (AWS WA Tool) 是雲端服務，提供您一致的程序以審查和量測使用 AWS Well-Architected Framework 的結構。AWS WA Tool 會給予推薦，使您的工作負載更可靠、更安全、更高效並且更經濟實惠。

為協助您應用最佳實務，我們特別成立 AWS Well-Architected 實驗室，提供程式碼與文件儲存庫，給您實作最佳實務的實際經驗。我們也與精選 AWS 合作夥伴網路 (APN) 合作夥伴組成團隊並肩合作，而這些合作夥伴即 AWS Well-Architected 合作夥伴計劃的成員。。這些 AWS 合作夥伴對 AWS 擁有深入的知識，能協助您審查和改進工作負載。

定義

AWS 的專家每一天都在輔助客戶進行系統架構，善用雲端的最佳實務。當您的設計演進時，有我們一同進行架構上的權衡。您將這些系統部署至即時環境後，我們可得知這些系統的效能狀況，以及這些權衡形成的後果。

我們便是基於得到的專業知識建立起 AWS Well-Architected Framework，其提供一套一致的最佳實務，供客戶和合作夥伴評估架構；並提供一份問題，您可用來評估架構與 AWS 最佳實務的吻合程度。

AWS Well-Architected Framework 以六個支柱為基礎：卓越營運、安全性、可靠性、效能達成效率、成本最佳化和永續性。

表 1.AWS Well-Architected Framework 的支柱

姓名	描述
卓越營運	可有效支援開發和執行工作負載、深入了解其營運狀況，以及持續改善支援流程和程序以產生商業價值的能力。
安全性	安全性支柱說明如何利用雲端技術，以能夠提升安全狀態的方式來保護資料、系統和資產。
可靠性	可靠性支柱包括工作負載如預期般正確、一致地執行其預期功能的能力。包括在整個生命週期中執行及測試工作負載。本白皮書深入說明在 AWS 上實作可靠工作負載的相關事項，提供最佳實務指導。
效能達成效率	有效率地使用運算資源以滿足系統需求，並隨著需求變更與技術發展來保持該效率需求的能力。
成本最佳化	在最低價格之下執行系統以產生商業價值的能力。
永續性	能夠透過獲取所佈建資源的最大效益並將所需的總資源數降至最低，以減少工作負載所有組件的能源消耗並提高其效率，從而持續改善永續性影響。

在 AWS Well-Architected Framework 中，我們會使用下列術語：

- 路由層 代表 針對要求一起交付的程式碼、組態和 AWS 資源。一個元件往往是技術擁有的單元，並自其他元件所解偶。
- 工作負載 是指 一組一起提供業務價值的元件。工作負載通常是商業和技術領導人溝通所談及的最細節的內容。
- 我們心目中的 架構 是指工作負載之中元件一同運作的方式。元件通訊與互動的方式往往成為架構圖的焦點。
- 里程碑 標示架構於產品生命週期之中演進的重要改變 (設計、實作、測試、上線，投入生產)。
- 在組織內， 技術組合 是業務運作所需工作負載的集合。
- AWS Well-Architected 工作量 是將任務針對實作所需的時間、工作和複雜性進行分類。每個組織都需要考慮團隊的大小和專業知識，以及工作負載的複雜性，以取得其他內容，將組織的工作量適當地分類。
  - 高： 工作可能需要數週或數個月。這可以分成多個案例、版本和任務。
  - 中： 工作可能需要數天或數週。這可以分成多個版本和任務。
  - 低： 工作可能需要數小時或數天。這可以分成多個任務。

建立工作負載的架構時，您可依照業務環境，在各支柱之間作出權衡。這些業務決定可主導您工程設計的優先順序。您可以優化以開發環境中的可靠性作為代價改善永續性影響並降低成本，或者針對關鍵任務解決方案，以較高成本和永續性影響達到可靠性的優化。在電子商務解決方案中，效能能影響營收和客戶購買的傾向。安全和卓越營運一般不會為了其他支柱而權衡妥協。

論架構

在內部部署環境中，客戶經常設置集中團隊負責技術架構，疊覆在其他產品或功能團隊上，以確保其遵照最佳實務。技術架構團隊通常包含一組角色，例如技術架構師 (基礎設施)、解決方案架構師 (軟體)、資料架構師、網聯架構師，和安全架構師。這類團隊經常採取 TOGAF 或 Zachman 框架 作為企業架構能力的部分。

在 AWS 上，我們偏好將能力分散至團隊中，不以集中團隊具備該項能力。選擇將決策權分散有其風險存在，確保團隊符合內部標準即為一例。我們以兩種方式降低這類風險。首先，我們 演練 (做事方式、程序、標準，及可接受的規範)，其會專注在使得各個團隊具備該項能力，並且請到專家，確保該團隊提高所需符合標準的標竿。第二，我們實作 機制 實施自動化檢查，以確保符合標準。

Jeff Bezos 說道「立意良好是不夠的，需要以良好的機制才能有所實現」。

這相當於將人為的盡力取代為機制，其能夠檢查是否遵循規則或程序 (經常為自動化形式)。這種分散式的作法 受到 Amazon 領導方針的支持，遍及所有角色培養 一種返向工作從客戶需求出發的 文化。返向工作是我們創新程序的基礎部分。我們從客戶及其期望著手，根據之定義並主導我們的工作方向。以客戶為尊的團隊會因應客戶的需要建置產品。

對架構而言，這表示我們期望每個團隊皆有能力建立架構，並且遵照最佳實務。為協助新團隊獲得這些能力，或使現有團隊提高標竿，我們促成與首席工程師的虛擬社群聯繫，委請審查團隊的設計，並協助團隊了解 AWS 最佳實務有哪些。首席工程設計社群使得最佳實務成為可見並可取得。例如，他們的一種作法是藉由午餐會報，專講將最佳實務套用至實際範例。這些會報經過錄製，可作為新進團隊成員的到任參考資料。

AWS 最佳實務源自我們以網際網路規模執行數千套系統所累積的經驗。我們偏好以資料定義最佳實務，不過也會起用主題專家，例如首席工程師進行訂定。當首席工程師看出有新的最佳實務出現時，會以社群形式工作，確保團隊遵守這些實務。假以時日，這些最佳實務會正式列入我們內部的審查程序，以及成為落實合規的機制。Well-Architected 架構是我們內部審查程序面向客戶的實作版，透過我們遍及領域的角色例如「解決方案架構」和內部工程設計團隊，將首席工程設計思維予以編撰。Well-Architected 架構是可擴展的機制，讓您能夠善用這些學習成果帶來的優勢。

依循這種對於架構的責任採取分散形式的首席工程設計社群作法，我們相信 Well-Architected 企業架構能因應客戶的需要而成形。技術領導者 (例如技術長或開發經理) 遍及您所有工作負載執行 Well-Architected 審查，能讓您更了解技術組合所具的風險。採行此方式之下，您可看出遍及團隊的主題，您的組織能以機制、培訓或午餐會報妥善顧及，如此一來首席工程師可向多個團隊分享對於特定領域的想法。

一般設計原則

Well-Architected 架構會確定一組一般設計原則，以促進在雲端進行良好的設計：

- 停止猜測您的容量需求：如果您在部署系統時做出糟糕的容量決定，可能最後變成坐擁昂貴的閒置資源，或處理容量有限的效能影響。而利用雲端運算，這些問題都會消失。您可依照需要使用大小不拘的容量，自動上下調整。
- 生產規模測試系統：在雲端，您可隨需建立生產規模的測試環境、完成測試，再將資源除役。因為您只為執行中的測試環境付費，所以能以與內部部署測試相較之下相當微小比例的成本來模擬即時環境。

- 自動化讓架構試驗更容易：自動化可讓您用低成本建立並複製工作負載，避免產生人工開支。您可追蹤自動化的變更，稽核其影響，並可視需要還原為先前參數。
- 允許演進的架構：在傳統環境中，架構上的決策往往實作成為靜態的一次性活動，其生命週期當中只有系統的少數主要版本。隨著業務及其環境持續改變，這些初始決定可能妨礙系統，無法符合不斷改變的業務要求。在雲端，按需自動化與測試的能力，可降低因設計變更而形成衝擊的風險。如此可允許系統隨時間演進，因此企業能以標準實務的形式享有創新的優勢。
- 使用資料來驅動架構：在雲端，您可收集架構上的選擇對於工作負載的行為有何影響的資料。如此可讓您為如何提升工作負載，做出以事實為根據的決策。您的雲端基礎設施為程式碼，因此可隨時間利用該資料得知基礎設施的適當選擇及提升。
- 透過演練日進行改進：為測試您的架構與程序的執行情況，可定期排定演練日，以模擬生產中的活動。如此可協助您了解何處有改善空間，並能協助發展組織處理活動的經驗。

架構的支柱

建立軟體系統很像是在興建大樓。若基礎不牢固，結構問題可能會逐漸影響建築物的完整性和功能。建構技術解決方案時，若您忽略卓越營運、安全性、可靠性、效能效率、成本最佳化和永續性這六大支柱，那麼建置滿足您期望與需求的系統將成為一項挑戰。將這些支柱納入您的架構，可協助您產出穩定又高效的系統。如此可允許您聚焦在設計的其他面向，例如功能要求。

支柱

- 卓越營運
- 安全性
- 可靠性
- 效能達成效率
- 成本最佳化
-

卓越營運

卓越營運支柱包括支援開發和執行工作負載、深入了解其營運狀況，以及持續改善支援流程和程序以產生商業價值的能力。

卓越營運支柱概述了設計原則、最佳實務和相關問題。您可以在下列白皮書中找到規範指引：卓越營運支柱白皮書。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端有五大設計原則來幫助實現卓越營運：

- 以程式碼執行營運：在雲端，您可以在整個環境中套用與您應用程式程式碼所用相同的工程原則。您可將整個工作負載 (應用程式、基礎設施) 定義為程式碼，並以程式碼加以更新。您可以程式碼實作

營運程序，並透過觸發這些程式碼來自動化執行，進而回應事件。透過以程式碼執行營運，您可限制人為錯誤並實現對事件的一致回應。

- 進行頻繁、細微和可逆的變更：設計工作負載以允許定期更新元件。進行小增量變更，以便在變更失敗時能撤回變更 (盡可能不影響客戶)。
- 經常完善營運程序：在使用營運程序時，尋找機會予以改善。發展工作負載，同時適當發展程序。設定定期演練日，以審查並驗證所有程序是否有效以及團隊是否熟悉這些程序。
- 預料失敗：執行「事前剖析」演練，以識別潛在的失敗來源，進而排除或減少這些來源。測試您的失敗情境並驗證您對它們的影響的理解。測試您的回應程序，以確保它們確實有效且團隊熟悉程序的執行。設定定期演練日，以測試工作負載和團隊對模擬事件的回應。
- 從所有營運失敗中學習經驗：從所有營運事件和失敗中學習經驗，進而不斷推動改善。跨團隊及在整個組織中分享獲得的經驗。

定義

雲端有四種最佳實務領域可實現卓越營運：

- 組織
- 準備
- 營運
- 演進

組織的領導階層定義業務目標。貴組織必須了解要求和優先順序，並運用這些資訊規劃和進行用以幫助達成業務成果的工作。您的工作負載必須提供支援工作負載所需的資訊。透過自動化重複程序的方式，實作整合、部署及交付工作負載的服務，將可讓生產享有更多有利的變更。

工作負載的操作本質上就可能存在著風險。您必須了解這些風險，並做出明智的決策才能進入生產階段。您的團隊必須能夠支援您的工作負載。從所需業務成果衍生的業務和營運指標，將讓您能夠了解工作負載的運作狀態、營運活動，並回應事件。您的優先事項會隨著業務需求和業務環境的變化而改變。運用這些方面做為回饋迴圈，以持續推動貴組織的改善和工作負載的操作。

最佳實務

主題

- 組織
- 準備

- 營運
- 演進

組織

您的團隊需要對您的整個工作負載，以及團隊成員在其中的作用達成共識，並且擁有共同的業務目標，以便設定能助力業務成功的優先事項。明確定義的優先事項將實現工作的最大收益。評估內部與外部客戶需求，並讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定工作的重點領域。評估客戶需求將確保您對實現業務成果所需的支援有透徹的了解。確保您了解由貴組織管控所定義的、可能要求或強調特定重點的準則或義務以及外部因素，例如法規合規要求和產業標準。確認您是否設有識別內部管控和外部合規要求變更的機制。如果未識別要求，請確保您已對此決定進行盡職調查。定期審查您的優先事項，以便在需求變更時更新優先事項。

評估對業務的威脅 (例如，業務風險和責任、資訊安全威脅)，並將此資訊保存在風險登記表內。評估風險，以及在相互衝突的利益或替代方法之間做出權衡的影響。例如，新功能加速上市可能是成本優化所強調的重點，或您可以為非關聯式資料選擇關聯式資料庫，以便更輕鬆地遷移系統，而非重構。管理收益和風險，以便在確定工作重點時做出明智的決定。某些風險或選擇可能在一段時間內是可以接受的，相關風險可能得以減輕，也可能出現無法接受風險存在的事實，在此情況下，您將需要採取動作來解決風險。

您的團隊必須了解其在達成業務成果中所扮演的角色。團隊需要了解自己在促成其他團隊成功的過程中所扮演的角色、其他團隊在促進其成功的過程中所扮演的角色，以及擁有共同目標。了解責任、擁有權、決策方式，以及誰有權制定決策，將有助於找到工作重點，並充分發揮團隊的優勢。團隊的需求將由其所支援的客戶、組織、團隊組成，以及工作負載的特性形塑而成。合理來說，無法要求單一操作模式支援貴組織中的所有團隊及其工作負載。

確保每個應用程式、工作負載、平台和基礎設施元件都有已識別擁有者，而且每個流程和程序都有負責其定義的已識別擁有者，以及負責其執行的擁有者。

透過了解每個元件、流程和程序的商業價值、為何部署這些資源或為何執行活動，以及該擁有權為何存在，有助於團隊成員採取適當動作。明確定義團隊成員的責任，以便他們能夠適當採取動作，並具備識別責任和擁有權的機制。設立可請求新增、變更和例外情況的機制，就能避免創新受到限制。在團隊之間制定協議，說明團隊如何共同合作以互相支援和協助達成業務成果。

為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。參與的高階領導層應設定期望並衡量成功。他們是採用最佳實務和組織演進的發起者、倡導者和推動者。給予團隊成員充分授權，讓他們可在成果出現風險時採取動作以將影響降到最低，同時鼓勵他們在遇到風險時，向決策者和利害關係人呈報，以便處理問題並避免事件發生。針對已知風險和計劃事件進行及時、明確且可採取動作的溝通，讓團隊成員能夠及時採取適當的動作。

鼓勵試驗以加速學習，讓團隊成員保持興趣並積極參與。團隊必須發展自己的技能集，以採用新技術，並支援需求和責任的變更。提供專門的結構化時間用於學習，以支援並鼓勵這一舉措。確保團隊成員擁有可助力取得成功並進行擴展的資源 (包括工具和團隊成員)，以協助達成您的業務成果。利用跨組織的多樣性，尋求多種獨特的觀點。使用此觀點來增加創新、挑戰假設，並降低確認偏差的風險。在團隊中增加包容性、多樣性和可及性，以獲得有益的觀點。

若有適用於貴組織的外部法規或合規要求， AWS 雲端合規 來協助教育您的團隊，以便他們可以判斷對您的優先事項的影響。Well-Architected 架構強調學習、衡量和改善。它為您提供可評估架構並實作將隨時間擴展之設計的一致方法。AWS 會提供 AWS Well-Architected Tool，以協助您在部署前檢閱方法、在生產前檢閱工作負載狀態，以及檢閱生產中的工作負載狀態。您可以將工作負載與最新的 AWS 架構最佳實務做比較、監控工作負載的整體狀態，以及深入了解潛在風險。AWS Trusted Advisor 是一款可存取核心檢查集的工具，這些檢查提出了優化建議，可能有助您確定優先事項。商業和企業支援客戶可存取針對安全性、可靠性、效能和成本優化的其他檢查，從而進一步協助確定他們的優先事項。

AWS 可以協助您教育您的團隊有關 AWS 及其服務的知識，從而增進他們對自己的選擇會如何影響工作負載的了解。您應使用 AWS Support (AWS 知識中心、AWS 論壇和 AWS Support 中心) 和 AWS 文件中的資源來教育您的團隊。透過 AWS Support 中心與 AWS Support 聯繫，以取得 AWS 問題方面的協助。AWS 也分享了我們透過在 Amazon Builders' Library 中營運 AWS 所學到的最佳實務和模式。您可透過 AWS 部落格和官方 AWS 播客獲得其他各種實用資訊。AWS Training and Certification 透過 AWS 基礎原理自主進度數位課程提供一些免費培訓。您還可以報名參加講師指導下的培訓，以進一步協助開發團隊的 AWS 技能。

您應該使用能集中管控跨帳戶環境的工具或服務，例如 AWS Organizations，以便協助您管理操作模式。AWS Control Tower 等服務會擴大此管理功能，讓您能定義帳戶設定的藍圖 (支援您的操作模式)、使用 AWS Organizations 套用持續管控，以及自動化新帳戶的佈建作業。AWS Managed Services、AWS Managed Services 合作夥伴等受管服務供應商，或 AWS 合作夥伴網路中的受管服務供應商，都會提供實作雲端環境的專業知識，並支援您的安全和合規要求及業務目標。將受管服務加入操作模式後，便可節省時間和資源，讓您的內部團隊精簡並專注於將使您的企業脫穎而出的策略性成果，而非開發新技能和功能。

下列問題著重於卓越營運方面的這些考量。(如需卓越營運問題清單和最佳實務，請參閱 附錄。)

OPS 1：如何決定您的優先事項？

每個人都必須了解自己在實現商業價值過程中的角色。擁有共同目標以設定資源優先順序。這會充分發揮您所做努力的優勢。

OPS 2：如何建構組織以支援業務成果？

您的團隊必須了解其在達成業務成果中所扮演的角色。團隊需要了解自己在促成其他團隊成功的過程中所扮演的角色、其他團隊在促進其成功的過程中所扮演的角色，以及擁有共同目標。了解責任、擁有權、決策方式，以及誰有權制定決策，將有助於找到工作重點，並充分發揮團隊的優勢。

OPS 3：您的組織文化如何支援您的業務成果？

為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。

您可能會發現，您在某個時間點會想要強調一小部分的優先事項。長期利用平衡的方法，以確保開發所需的功能和管理風險。定期審查優先事項，並隨需求的變更進行更新。如果責任和擁有權未定義或未知，則您會面臨風險，不僅無法及時執行必要的動作，在解決這些需求時還會出現冗餘和可能相互衝突的工作。組織文化對團隊成員工作滿意度和留任率有直接影響。讓團隊成員參與其中並習得能力，以便讓業務得以成功。必需要經由試驗才能實現創新，並讓想法轉化為成果。認識到不想要的結果是成功的試驗，因其已識別出不會助力成功的路徑。

準備

要為卓越營運做好準備，您必須了解您的工作負載及其預期行為。然後，您就能將其設計出來，以了解它們的狀態並建置可提供支援的程序。

設計您的工作負載，使其提供必要資訊，讓您了解所有元件的內部狀態 (例如，指標、日誌、事件和追蹤)，以支援可觀測性和調查問題。透過反覆操作，開發監控工作負載運作狀態所需的遙測、識別成果的風險在何時發生，並實現有效回應。在檢測您的工作負載時，擷取大量資訊以實現狀況認知 (例如，狀態變更、使用者活動、權限存取、利用率計數器)，從而知道您可使用篩選條件選擇某段時間內最有用的資訊。

採用的方法需能夠改善變更進入生產環境的流程，並支援重構、快速提供品質意見回饋及修復錯誤。這會加快有助益的變更進入生產環境的速度、限制部署問題，並快速識別和修復部署活動所導致或在您的環境中所發現的問題。

採用可快速提供品質意見回饋，並從成果不盡理想的改變中快速復原的方法。使用這些實務可緩解部署變更所帶來問題的影響。為變更失敗做好規劃，以便在必要時能夠快速回應，同時測試並驗證所做變更。了解環境中的計劃內活動，以便管理會影響計劃內活動的變更風險。強調頻繁、細微、可逆的變更，以限制變更範圍。透過回復變更，可以更輕鬆地進行故障診斷並加快修復速度。這也表示您從有價值變更中受益的頻率會提高。

評估工作負載、流程、程序及人員的營運準備度，以了解與工作負載相關的營運風險。您應使用一致的程序 (包括手動或自動檢查清單) 來獲悉工作負載或變更執行就緒的時間。這樣一來，您也將能尋找任何需要您制定解決方案的領域。具備可記錄例行活動的執行手冊，以及可指引問題解決程序的程序手冊。了解收益和風險，以做出明智決策，讓變更順利進入生產環境。

AWS 讓您能以程式碼檢視您的整個工作負載 (應用程式、基礎設施、原則、管控和營運)。這表示您可以將用於應用程式程式碼的相同工程規則套用到堆疊的每個元素，並在團隊或組織之間分享這些元素，以擴大開發工作的優勢。在雲端以程式碼執行營運，並利用安全進行試驗的能力，開發工作負載、營運程序以及實務失敗案例。使用 AWS CloudFormation，您將能擁有一致的範本化沙盒開發、測試和生產環境，同時還能提高營運控制等級。

下列問題著重於卓越營運方面的這些考量。

OPS 4：您如何設計工作負載以便了解其狀況？

設計工作負載，以便它為您提供了解其內部狀態所需的跨全部元件 (例如指標、日誌和追蹤) 的資訊。這讓您能在適當時機提供有效回應。

OPS 5：您如何減少缺陷、幫助輕鬆修復，以及改善生產流程？

採用改善改變生產流程的方法，藉此重構、快速提供品質意見回饋及修復錯誤。這會加快有助益的改變發揮作用的速度、限制部署問題，並快速識別和修復部署活動造成的問題。

OPS 6：您如何緩解部署風險？

採用可快速提供品質意見回饋，並從成果不盡理想的改變中快速復原的方法。使用這些實務可緩解部署變更所帶來問題的影響。

OPS 7：您如何知道自己準備好支援工作負載？

評估工作負載、流程和程序及人員的營運準備度，了解工作負載相關營運風險。

對以程式碼實作營運活動進行投資，從而最大程度地提高營運人員的生產力，將錯誤率降至最低以及實現自動回應。使用「事前剖析」可預測失敗並適時建立程序。依照一致的標記策略，使用資源標籤和

AWS Resource Groups 來套用中繼資料，以識別您的資源。標記您的資源，以用於組織、成本會計、存取控制，以及將自動執行營運活動設為目標。採用可利用雲端彈性的部署實務，以促進開發活動和系統的預部署，進而加快實作速度。當您變更您用於評估工作負載的檢查清單時，請計劃如何處理不再合規的即時系統。

營運

我們可根據業務和客戶成果的實現情況，衡量是否成功運作工作負載。定義預期成果，確定如何衡量成功，並識別可用於這些計算的指標，以判斷您的工作負載和營運是否成功。營運運作狀態包括工作負載的運作狀態，以及為支援工作負載所執行營運活動 (例如，部署和事件回應) 的運作狀態和成功情況。建立指標基準以便進行改善、調查和介入；收集並分析指標；然後，驗證您對營運成功及其隨著時間的變化情況的理解。使用收集的指標來確定您是否滿足客戶和業務需求，並識別有待改善的領域。

要實現卓越營運，必須有效地管理營運事件。這適用於計劃和非計劃中的營運事件。使用已建立的執行手冊處理已充分了解的事件，並使用程序手冊協助調查和解決問題。根據事件對業務和客戶的影響來確定回應事件的優先順序。確保如因回應事件而發出提醒，則將由明確識別的擁有者執行關聯程序。事先定義解決事件所需的人員，並納入向上呈報觸發條件，以在必要時根據緊迫性和影響力，在其中新增額外的參與人員。識別並邀請具有權限的個人來決定行動方案，該方案將受到先前未解決的事件回應的業務影響。

透過針對目標受眾 (例如，客戶、業務、開發人員、營運) 量身定制的儀表板和通知來傳達工作負載的運行狀態，以便他們能採取適當的動作，進而管理他們的期望並在恢復正常營運時得到通知。

在 AWS 中，您可以產生從工作負載或以原生方式從 AWS 收集的指標的儀表板視圖。您可以利用 CloudWatch 或第三方應用程式，來彙總和顯示營運活動的業務、工作負載和營運等級視圖。AWS 可透過記錄功能 (包括 AWS X-Ray 、CloudWatch、CloudTrail 和 VPC Flow Logs) 提供工作負載洞見，從而能夠識別工作負載問題，以支援根本原因分析和修復。

下列問題著重於卓越營運方面的這些考量。

OPS 8：您如何了解工作負載的運作狀態？

定義、擷取和分析工作負載指標，掌握工作負載事件，以便採取適當行動。

OPS 9：您如何了解營運狀況？

定義、擷取和分析營運指標，掌握營運事件，以便採取適當行動。

OPS 10：您如何管理工作負載和營運事件？

準備和驗證回應事件的程序，大幅降低工作負載中斷情形。

您收集的所有指標都應該符合業務需求及其支援的結果。開發針對已充分了解之事件的指令碼式回應，並自動化其效能以回應事件辨識。

演進

您必須學習、分享和持續改善以維持卓越營運。投入工作週期以持續逐漸改善。針對所有影響客戶的事件執行事件後分析。確定成因和預防措施，限制或防止其再次發生。視情況與受影響的社群溝通成因。定期評估改進機會 (例如，功能請求、問題修復和合規要求) 並確定其優先順序，包括工作負載和營運程序。

在您的程序中納入回饋迴圈，以快速識別有待改善的領域並從營運執行中獲得經驗。

在遊戲日內，可跨團隊分享獲得的經驗，進而分享這些經驗的益處。分析獲得的經驗中的趨勢，並執行營運指標的跨團隊回溯分析，以識別改善機會和方法。實作旨在帶來改善的變更，並評估結果以判斷是否成功。

在 AWS 上，您可以將日誌資料匯出至 Amazon S3 或直接將日誌傳送至 Amazon S3，以便長期儲存。您可以使用 AWS Glue，在 Amazon S3 中探索和準備日誌資料，以進行分析並將關聯的中繼資料儲存在 AWS Glue Data Catalog 中。Amazon Athena，透過與 AWS Glue 的原生整合，可用來分析日誌資料，並使用標準 SQL 進行查詢。您可以使用 Amazon QuickSight 這類商業智慧工具來視覺化、探索並分析資料。探索可能推動改善的感興趣趨勢和事件。

下列問題著重於卓越營運方面的這些考量。

OPS 11：您如何改善營運？

投入時間和資源持續逐漸改善，以加強營運的效果和效率。

成功的營運演進基於：頻繁、細微的改善；提供安全的環境和時間來試驗、開發和測試改善；鼓勵營造從失敗中學習的環境。隨著營運控制等級的提高，對沙盒、開發、測試和生產環境的營運支援可促進開發，並提高將變更部署至生產中後取得成功結果的可預測性。

資源

請參閱下列資源，進一步了解我們卓越營運的最佳實務。

文件

- DevOps 與 AWS

白皮書

- 卓越營運支柱

影片

- Amazon 的 DevOps

安全性

安全支柱包含能夠保護資料、系統和資產，以利用雲端技術來改善安全性。

安全支柱概述了設計原則、最佳實務和相關問題。您可以在下列白皮書中找到規範指引： 安全支柱白皮書。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端安全有七個設計原則：

- 建立強大的身份識別基礎：實作最低授權原則，並對於每個與 AWS 資源的互動強制執行職責與適當的授權分離。集中化身份管理，旨在消除對長期靜態登入資料的倚賴。

- 啟用可追溯性：即時監控、提醒和稽核動作和對您環境的變更。將日誌和指標收集與系統進行整合，以自動調查並採取動作。
- 在所有層套用安全性：使用多個安全控制套用深度防禦方法。套用至所有層級 (例如，網路邊緣、VPC、負載平衡、每個執行個體和運算服務、作業系統、應用程式和程式碼)。
- 自動化安全最佳實務：將基於軟體的安全性機制自動化，可提高您安全、快速和以具成本效益的方式擴展的能力。建立安全架構 (包括實作控制) 在版本控制的範本中作為程式碼定義和管理。
- 保護傳輸中資料和靜態資料：將您的資料分為不同的敏感性等級，並使用適當的機制，例如加密、權杖化及存取控制。
- 讓人員遠離資料：使用機制和工具，來降低或消除對直接存取或手動處理資料的需要。在處理敏感資料時，這降低了處理不當或修改以及人為錯誤的風險。
- 為安全事件做準備：為事故做好萬全準備，建立與您組織的要求吻合的事故管理和調查政策與程序。執行失敗回應模擬和使用工具與自動化，以提高偵測、調查和復原的速度。

定義

雲端安全有六個最佳實務領域：

- 安全性
- Identity and Access Management
- 偵測
- 基礎設施保護
- 資料保護
- 事故回應

在架構任何工作負載之前，您需要採取影響安全性的實務。您會希望控制誰可以做什麼。另外，您需要能夠識別安全事故、保護系統和服務，並透過資料保護維持資料的保密與完整。您應當具備界定完善且經過演練的程序，以因應安全事故。這些工具和技術之所以重要，因為能支援諸多目的，例如防止財務損失或遵循法規義務。

AWS 共同的責任模式讓採用雲端的組織能夠達成安全與合規目標。AWS 能實體上地保護支援本公司雲端服務的基礎設施，好讓作為 AWS 客戶的您專心使用服務以達成目標。AWS 雲端還提供對安全資料更好的存取，並有自動方式可回應安全事件。

最佳實務

主題

安全性

若要安全地操作工作負載，您必須將總體最佳實務套用到每個安全領域。採用您在組織和工作負載層級所定義的卓越營運要求和程序，將這些要求和程序套用到所有領域。

透過 AWS 和產業建議與威脅情報持續取得最新資訊，可協助您發展威脅模型和控制目標。自動化安全程序、測試和驗證可讓您擴展安全操作。

下列問題著重於這些安全方面的考量。(如需安全問題和最佳實務的清單，請參閱 附錄。)

SEC 1：如何安全地操作工作負載？

若要安全地操作工作負載，您必須將總體最佳實務套用到每個安全領域。採用您在組織和工作負載層級所定義的卓越營運要求和程序，將這些要求和程序套用到所有領域。透過 AWS 和產業建議與威脅情報持續取得最新資訊，可協助您發展威脅模型和控制目標。自動化安全程序、測試和驗證可讓您擴展安全操作。

在 AWS 中，建議根據不同的功能和合規或資料敏感性等要求，依帳戶分隔不同的工作負載。

身分和存取管理

Identity and Access Management 是資訊安全計畫的關鍵部分，可確保只有經過授權和身分驗證的使用者和元件，才能以您想要的方式存取您的資源。例如，您應定義主體 (即為可在您的帳戶內執行動作的帳戶、使用者、角色和服務)，建立與這些主體一致的政策，並實作強勢憑證管理。這些權限管理元素構成身份驗證與授權的核心。

在 AWS 中，權限管理主要由 AWS Identity and Access Management (IAM) 服務支援，它讓您可以控制對 AWS 服務和資源的使用者和程式設計存取。您應該套用精細的政策，將權限分配給使用者、群組、角色或資源。您還可以要求使用強式密碼，例如要求複雜性等級、避免重複使用以及強制執行多重因素認證 (MFA)。您可以將聯合身份驗證與現有目錄服務一起使用。對於要求系統有權存取 AWS 的工作負載，IAM 可以透過角色、執行個體描述檔、聯合身份和臨時登入資料來實現安全存取。

下列問題著重於安全方面的這些考量。

SEC 2：如何管理人員和機器的身分？

處理操作安全的 AWS 工作負載時，您需要管理兩種身分類型。了解您需要管理和授予存取權的身分類型，有助於確保正確的身分在適當的條件下存取正確的資源。

人員身分：您的管理員、開發人員、操作員和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些人是組織的成員，或與您協作的外部使用者，以及透過 Web 瀏覽器、用戶端應用程式或互動式命令列工具與 AWS 資源互動的使用者。

機器身分：您的服務應用程式、操作工具和工作負載需要身分，才能向 AWS 服務發出請求，例如讀取資料。這些身分包括在 AWS 環境中執行的機器，例如 Amazon EC2 執行個體或 AWS Lambda 函數。您也可以為需要存取權的外部人員管理機器身分。此外，您可能也有在 AWS 外部，需要存取 AWS 環境的機器。

SEC 3：如何管理人員和機器的許可？

管理許可，以控制對需要存取 AWS 和工作負載的人員和機器身分的存取。許可控制誰可以在何種條件下存取哪些內容。

登入資料不得在任何使用者或系統之間共用。應使用最低權限的方法以及最佳實務 (包括密碼要求和強制執行 MFA) 來授予使用者存取權限。包括對 AWS 服務的 API 呼叫在內的程式設計存取應使用臨時和有限權限的登入資料 (例如由 AWS Security Token Service 發出的登入資料) 執行。

AWS 提供了可以幫助您進行身份和存取管理的資源。為了幫助您學習最佳實務，請探索我們的實作實驗室， 了解管理登入資料和身份驗證、 控制人為存取和 控制程式設計存取。

偵測

您可以使用偵測控制來識別潛在的安全威脅或事故。它們是管控框架的重要組成部分，可用於支援品質流程、法律或合規義務以及用於威脅識別和回應工作。偵測控制有不同的類型。例如，建立資產及其詳細屬性的詳細目錄可促進更有效的決策 (和生命週期控制)，以幫助建立營運基準。您還可以使用內部稽核，即檢查與資訊系統相關的控制，以確保實務符合政策和要求，並確保已根據定義的條件設定正確的自動提醒通知。這些控制是重要的反應式因素，可以幫助您的組織識別和了解異常活動的範圍。

在 AWS 中，您可以透過處理日誌、事件和監控來實作偵測控制，以進行稽核、自動分析和警示。CloudTrail 日誌、AWS API 呼叫和 CloudWatch 監控指標並發出警示，AWS Config 提供組態歷程記錄。Amazon GuardDuty 是受管威脅偵測服務，可持續監控惡意或未經授權的行為，協助您保護 AWS 帳戶和工作負載。也提供服務層級日誌。例如，您可以使用 Amazon Simple Storage Service (Amazon S3) 記錄存取請求。

下列問題著重於這些安全方面的考量。

SEC 4：您如何偵測和調查安全事件？

從日誌和指標中擷取並分析事件以掌握情況。針對安全事件和潛在威脅採取行動，有助於保護工作負載。

日誌管理對 Well-Architected 工作負載至關重要，原因包括安全/鑑識，以及法規或法律要求等。分析日誌並對其進行回應，以便可以識別潛在的安全事故，這一點至關重要。AWS 提供了讓您能夠定義資料保留生命週期或定義將在何處儲存、存檔或最終刪除資料的功能，從而使日誌管理更易於實作。這使得可預測和可靠的資料處理更加簡單，且更具成本效益。

基礎設施保護

基礎設施保護包括符合最佳實務和組織或監管義務所必需的控制方法，例如深度防禦。這些方法的使用對於雲端或內部部署成功持續營運至關重要。

在 AWS 中，您可以透過使用 AWS 原生技術或透過 AWS Marketplace 獲得的合作夥伴產品和服務，來實作有狀態和無狀態封包檢查。您應該使用 Amazon Virtual Private Cloud (Amazon VPC) 建立一個私有、安全且可擴展的環境，您可以在其中定義拓撲，包括閘道、路由表以及公有和私有子網路。

下列問題著重於安全方面的這些考量。

SEC 5：如何保護您的網路資源？

任何具有某種網路連線能力的工作負載，無論是網際網路或私有網路，都需要多層防禦來協助保護其不受外部和內部網路威脅的影響。

SEC 6：您如何保護運算資源？

工作負載中的運算資源需有多層防護，協助防範外部和內部威脅。運算資源包括 EC2 執行個體、容器、AWS Lambda 函數、資料庫服務、IoT 裝置等。

不管是何種類型的環境，建議使用多層防禦。就基礎設施保護而言，許多概念和方法在雲端和內部部署均有效。加強邊界保護、監控入口和出口以及全面的記錄、監控和提醒，對於有效的資訊安全計劃均很重要。

AWS 客戶能夠量身訂製或強化 Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Elastic Container Service (Amazon ECS) 容器或 AWS Elastic Beanstalk 執行個體的組態，並在一個不變的 Amazon Machine Image (AMI) 中持續地長期保留組態。然後，無論是由 Auto Scaling 觸發還是手動啟動，使用此 AMI 啟動的所有新虛擬伺服器 (執行個體) 都將獲得此強化組態。

資料保護

在設計任何系統之前，應建立影響安全性的基礎實務。例如，資料分類可基於敏感層級將組織的資料分類，加密則能對未經授權的存取將資料呈現為無法辨識，以保護資料。這些工具和技術之所以重要，因為能支援諸多目的，例如防止財務損失或遵循法規義務。

在 AWS 中，以下實務有助於保護資料：

- 作為 AWS 客戶，您可完全控管資料。
- AWS 讓您可以更輕鬆地加密資料和管理金鑰，包括常規的金鑰輪換。這些可以透過 AWS 輕鬆地自動化或由您手動維護。
- 提供了包含重要內容 (例如檔案存取和變更) 的詳細記錄。
- AWS 設計的儲存系統具有卓越彈性。例如，Amazon S3 Standard、S3 Standard-IA、S3 One Zone-IA 和 Amazon Glacier 都在給定年份內提供 99.999999999% 的物件耐用性。此耐用性等級相當於 0.000000001% 物件年平均預期損失率。
- 版本控制可以作為更大的資料生命週期管理過程的一部分，可以防止意外的覆寫、刪除和類似損害。

- AWS 永遠不會主動移動區域之間的資料。除非您明確啟用相關功能或利用提供相關功能的服務，否則放置在某個區域中的內容將保留在該區域中。

下列問題著重於安全方面的這些考量。

SEC 7：您如何分類資料？
資料分類可讓您根據關鍵性和敏感度將資料分類，協助判定適當的保護和保留控制。

SEC 8：您如何保護靜態資料？
實作多個控制來保護您的靜態資料，以降低未經授權的存取或不當處理的風險。

SEC 9：您如何保護傳輸中資料？
實作多個控制以保護傳輸中的資料，減少未經授權的存取或遺失的風險。

AWS 提供多種加密靜態資料和傳輸中資料的方法。我們將功能內建到我們的服務中，讓您可以更輕鬆地加密資料。例如，我們為 Amazon S3 實作了伺服器端加密 (SSE)，讓您可以更輕鬆地以加密形式儲存資料。您還可以安排由 Elastic Load Balancing (ELB) 處理整個 HTTPS 加密和解密過程 (通常稱為 SSL 終止)。

事故回應

即使採用了非常成熟的預防和偵測控制，您的組織仍應建立適當的流程，來回應和緩和安全事故的潛在影響。工作負載的架構嚴重影響團隊在事故期間有效執行、隔離或控制系統，以及將營運恢復到已知良好狀態的能力。在發生安全事故之前布置好工具和存取權限，然後在演練日期間例行練習事故回應，將幫助您確保架構可以適應即時調查和復原。

在 AWS 中，以下實務有助於有效地回應事故：

- 提供了包含重要內容的詳細記錄，例如檔案存取和變更。
- 可以自動處理事件並觸發工具，以透過使用 AWS API 來自動執行回應。
- 您可以使用 AWS CloudFormation 預先佈建工具和「潔淨室」。這樣一來，您就可以在安全、隔離的環境中進行鑑識。

下列問題著重於這些安全方面的考量。

SEC 10：您如何預估、回應事件以及從事件中復原？
準備對於及時且有效的調查、回應事件以及從事件中復原至關重要，有助於將對組織的干擾降到最低。

確保您有一種方法可以快速授予安全團隊存取權限，並自動隔離執行個體以及為鑑識收集資料和狀態。

資源

請參閱以下資源，進一步了解我們的安全最佳實務。

文件

- AWS 雲端安全
- AWS 合規
- AWS 安全部落格

白皮書

- 安全支柱
- AWS 安全概觀
- AWS 風險與合規

影片

- 聯盟 AWS 安全狀況
- 共同責任概觀

可靠性

可靠性支柱包括工作負載如預期般正確、一致地執行其預期功能的能力。包括在整個生命週期中執行及測試工作負載。本白皮書深入說明在 AWS 上實作可靠工作負載的相關事項，並提供最佳實務指導。

可靠性支柱概述了設計原則、最佳實務和相關問題。您可以在下列白皮書中找到規範指引：可靠性支柱白皮書。

主題

設計原則

雲端可靠性有五個基本的設計原則：

- 自動從失敗中復原：透過監控工作負載的關鍵績效指標 (KPI)，您可在達到臨界值時觸發自動化。這些 KPI 應為業務價值的衡量指標，而非服務營運的技術方面。如此一來，即可自動通知和追蹤失敗，以及自動化可解決或修復失敗的復原程序。藉助更複雜的自動化功能，您可以在發生失敗前進行預測和修補。
- 測試復原程序：在內部部署環境中，經常執行測試以證明工作負載可在特定情況下正常工作。測試通常不可用於驗證復原策略。在雲端，您可測試工作負載會發生哪些失敗情境，同時可驗證復原程序。您可使用自動化來模擬不同的失敗情境或重新建立會導致之前失敗的情境。此方法會在實際的失敗情境發生前公開您可以測試和修復的失敗路徑，從而降低風險。
- 水平擴展以提高總體工作負載可用性：使用多個小資源取代一個大資源，以降低整體工作負載上發生單一失敗時造成的影響。將請求分散到多個較小的資源，以確保它們不會有共同的失敗點。
- 停止猜測容量：內部部署工作負載失敗的一個常見原因是資源飽和，即當對工作負載的需求超出該工作負載的容量時發生的情況 (這通常為阻斷服務攻擊的目標)。在雲端，您可以監控需求和工作負載利用率，並自動新增或刪除資源，以保持可滿足需求的最佳水平，而不會過度佈建或佈建不足。仍然存在限制，但是某些配額可以控制，而其他限制則可管理 (請參閱管理Service Quotas和限制)。
- 管理自動化變更：應使用自動化來執行對基礎設施的變更。需要管理的變更包括之後可以追蹤和審查的自動化變更。

定義

雲端可靠性有四個最佳實務領域：

- 基礎

- 工作負載架構
- 變更管理
- 失敗管理

若要實現可靠性，您必須先從基礎開始，即服務配額和網路拓撲能適應工作負載的環境。分散式系統的工作負載架構在設計上必須能防止失敗並減輕失敗的影響。工作負載必須處理需求或要求的變更，且在設計上須能偵測失敗並自動進行自我修復。

最佳實務

主題

基礎

基礎要求是其範圍超過單一工作負載或專案的要求。在建立任何系統架構之前，應確立會影響可靠性的基本要求。例如，您必須為資料中心提供足夠的網路頻寬。

藉助 AWS，這些基礎需求中的大多數已予以納入或可以按需要進行處理。設計的雲端近乎無限，因此 AWS 有責任滿足足夠的聯網和運算容量的要求，讓您可以根據需要自由變更資源大小和分配。

下列問題著重於可靠性方面的這些考量。(如需可靠性問題清單和最佳實務，請參閱 附錄。)

REL 1：您如何管理服務配額和限制？
雲端型工作負載架構會有服務配額 (也稱為服務限制)。這些配額旨在用於防止不慎佈建超過您所需的資源，並限制 API 操作上的請求率，以防止服務遭到濫用。此外也會有資源限制，例如，您可將位元壓入光纖電纜的速率或實體磁碟上的儲存量會受到限制。

REL 2：如何規劃您的網路拓撲？

工作負載經常存在於多個環境中。這些環境包括多個 (可公開存取和私有的) 雲端環境，且可能包含您現有的資料中心基礎設施。計畫必須包括系統內和系統間連線、公有 IP 地址管理、私有 IP 地址管理和網域名稱解析等網路考量因素。

雲端型工作負載架構會有服務配額 (也稱為服務限制)。這些配額旨在用於防止不慎佈建超過您所需的資源，並限制 API 操作上的請求率，以防止服務遭到濫用。工作負載經常存在於多個環境中。您必須監控和管理這些適用於所有工作負載環境的配額。這些環境包括多個 (可公開存取和私有的) 雲端環境，且可能包含您現有的資料中心基礎設施。計畫必須包括系統內和系統間連接、公有 IP 地址管理、私有 IP 地址管理和網域名稱解析等網路考量因素。

工作負載架構

可靠的工作負載始於對軟體和基礎設施的前期設計決策。您的架構選擇會對所有 Well-Architected 支柱的工作負載行為產生影響。為求可靠性，您必須依循特定模式。

藉助 AWS，工作負載開發人員可以選擇要使用的語言和技術。AWS 開發套件為 AWS 服務提供特定語言 API，讓編碼不再如此複雜。這些開發套件加上各種語言選項，可讓開發人員實作本文列出的可靠性最佳實務。開發人員也可 在 Amazon Builders' Library 中閱讀和了解 Amazon 如何建置和操作軟體。

下列問題著重於可靠性方面的這些考量。

REL 3：如何設計您的工作負載服務架構？

使用服務導向架構 (SOA) 或微型服務架構，建置擴展性與可靠性高的工作負載。服務導向架構 (SOA) 是透過服務界面讓軟體元件可重複使用的做法。微型服務架構則進一步讓元件變得更小、更簡單。

REL 4：如何在分散式系統中設計防止失敗的互動？

分散式系統倚賴通訊網路來互連元件，例如伺服器或服務。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務可防止失敗，並延長平均失敗間隔時間 (MTBF)。

REL 5：如何設計分散式系統中的互動以緩解或承受故障？

分散式系統倚賴通訊網路來互連元件 (例如，伺服器或服務)。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務讓工作負載能夠承受壓力或故障，更快速地從其中復原，並減輕這類受損的影響。最終縮短平均復原時間 (MTTR)。

變更管理

必須預期並因應工作負載或其環境的變更，才能實現可靠的工作負載操作。變更包括對工作負載強加的變更，例如需求峰值，以及內部的變更，例如功能部署和安全性修補程式。

您可以使用 AWS 監控工作負載的行為，並自動化對 KPI 的回應。例如，隨著工作負載的使用者增加，您的工作負載可能會新增其他伺服器。您可以控制有權作出工作負載變更的人員，並稽核這些變更的歷史紀錄。

下列問題著重於可靠性方面的這些考量。

REL 6：如何監控工作負載資源？

日誌和指標是深入了解工作負載運作狀態的強大工具。您可以設定工作負載以監控日誌和指標，並在超過閾值或發生重大事件時傳送通知。監控可讓您的工作負載識別何時會超過低效能閾值或發生故障，以便自動復原來回應。

REL 7：如何設計工作負載以適應需求變更？

可擴展工作負載提供自動新增或移除資源的彈性，以便隨時盡可能符合目前需求。

REL 8：您如何實作變更？

需有控制變更以部署新功能，並確保工作負載和運作環境執行已知軟體，且能以可預測的方式修補或取代。如果這些變更不受控制，則難以預測這些變更的效果，或是解決肇因於這些變更的問題。

當您建立工作負載架構以根據需求的變更自動新增和刪除資源時，其不僅可以提高可靠性，而且還能確保企業成功不會成為負擔。在適當監控下，當 KPI 偏離預期規範時，您的團隊將會自動收到提醒。自動記錄對環境的變更，讓您可進行稽核並快速識別可能影響可靠性的動作。對變更管理的控制將確保您能執行交付所需可靠性的規則。

失敗管理

在任何合理複雜的系統中，均有可能會發生失敗。為達可靠性要求，您的工作負載應在發生失敗時察覺此情況，並採取行動以免影響可用性。工作負載必須能夠承受失敗並自動修復問題。

藉助 AWS，您可以利用自動化對監控資料作出反應。例如，當特定指標超過臨界值時，您可以觸發可修補問題的自動化動作。此外，您無需嘗試診斷和修正生產環境中的失敗資源，而是可以用新的資源取代它，並對失敗的頻外資源執行分析。由於雲端可讓您以低成本建立整個系統的臨時版本，因此您可以使用自動化測試來驗證完整的復原程序。

下列問題著重於可靠性方面的這些考量。

REL 9：您如何備份資料？

備份資料、應用程式和組態，以符合復原時間目標 (RTO) 和復原點目標 (RPO) 的要求。

REL 10：如何使用故障隔離來保護您的工作負載？

故障隔離界限會在工作負載內將失敗影響限制至有限數量的元件。界限外的元件不受失敗影響。使用多個故障隔離界限時，您可以限制對工作負載的影響。

REL 11：如何設計工作負載以承受元件失敗？

需要高可用性和低平均復原時間 (MTTR) 的工作負載必須建立彈性架構。

REL 12：如何測試可靠性？

在將工作負載設計為可彈性應對生產壓力之後，測試是確保其依設計運作並交付您預期之彈性的唯一方法。

REL 13：您如何規劃災難復原 (DR)？

備妥備份和冗餘工作負載元件是 DR 策略的開始。RTO 和 RPO 是您還原 工作負載的目標。根據業務需求設定這些目標。實作策略以滿足這些目標，考量工作負載資源和資料的位置和功能。發生中斷的可能性和復原成本也是重要因素，可反映為工作負載提供災難復原的商業價值。

定期備份資料並測試備份檔案，從而確保您可以從邏輯和物理錯誤中復原。管理失敗的關鍵是對導致失敗的工作負載頻繁進行自動化測試，然後觀察它們可如何復原。定期執行此操作，並確保在出現重大工作負載變更後也能觸發此類測試。主動追蹤 KPI，以及復原時間目標 (RTO) 和復原點目標 (RPO)，以評估工作負載的彈性 (尤其是在失敗測試情境下)。追蹤 KPI 將能助您識別和減輕單一失敗點。其目標是徹底測試您的工作負載復原程序，以便您確信即使面對持續問題，您也可以復原所有資料並繼續為客戶提供服務。應與執行正常生產程序一樣執行復原程序。

資源

請參閱以下資源，進一步了解我們的可靠性最佳實務。

文件

- AWS 文件
- AWS 全球基礎設施
- AWS Auto Scaling：擴展計畫的運作方式
- 什麼是 AWS Backup？

白皮書

- 可靠性支柱：AWS Well Architected
- 實作 AWS 上的微型服務

效能達成效率

效能達成效率要件包括有效率地使用運算資源以滿足系統需求，並隨著需求變更與技術發展來保持該效率需求的能力。

效能達成效率支柱概述了設計原則、最佳實務和相關問題。您可以在下列白皮書中找到規範指引：效能達成效率支柱白皮書。

主題

設計原則

雲端的效能達成效率有五個設計原則：

- 讓進階技術變得更普及：將複雜的任務委派給雲端廠商，讓團隊更輕鬆地實作進階技術。與其要求 IT 團隊了解新技術的託管和執行方式，不如考慮使用技術即服務。例如，NoSQL 資料庫、媒體轉碼和機器學習均為需要專業知識的技術。在雲端，這些技術成為團隊可以使用的服務，讓團隊可專注於產品開發，而非資源佈建及管理。
- 在幾分鐘內將業務擴展到全球在全球多個 AWS 區域部署工作負載，讓您以最低的成本，為客戶提供較低的延遲和更好的體驗。
- 使用無伺服器架構：採用無伺服器架構，您便無需執行和維護實體伺服器來完成傳統運算活動。例如，無伺服器儲存服務可以充當靜態網站 (因此無需 Web 伺服器)，而事件服務可以為您託管程式碼。如此一來，即可減輕管理實體伺服器的營運負擔，而且由於這些受管服務是在雲端規模上運行，因此還可以降低交易成本。
- 提高試驗頻率：藉助虛擬及可自動化的資源，您可以使用不同類型的執行個體、儲存設備或組態，迅速完成比較測試。
- 考慮機械共鳴作用：了解雲端服務的使用方式，並一律使用最符合工作負載目標的技術方法。例如，在您選擇資料庫或儲存方法時，請考慮資料存取模式。

定義

雲端效能達成效率的最佳實務有四個領域：

- 選擇
- 審查
- 監控
- 權衡

採取資料驅動的方法來建置高效能架構。從高階設計到資源類型的選擇和組態，收集架構各方面的資料。

定期審查您的選擇，確保充分利用不斷演進的 AWS 雲端。監控可確保您能察覺任何與預期效能的偏差。在架構中做出權衡以改進效能，例如使用壓縮或快取，或放寬一致性要求。

最佳實務

主題

- 選擇
- 審查
- 監控
- 權衡

選擇

適用於特定工作負載的最佳解決方案各不相同，而解決方案通常會結合多種方法。Well-Architected 工作負載會使用多重解決方案，並啟用不同功能以提升效能。

AWS 資源有多種類型和組態，可讓您更輕鬆地找到最符合工作負載需求的方法。您還可以發現使用內部部署基礎設施不易實現的選項。例如，Amazon DynamoDB 這種受管服務，可提供全受管的 NoSQL 資料庫及任何規模下的十毫秒內延遲時間。

下列問題著重於效能達成效率方面的這些考量。(如需效能達成效率問題和最佳實務的清單，請參閱 附錄。)

PERF 1：您如何選擇效能最佳的架構？

欲讓工作負載達到最佳效能通常需要採用多種方法。Well-Architected 系統會使用多重解決方案和功能以提升效能。

使用資料驅動的方法，為您的架構選取模式和實作，並達成具有成本效益的解決方案。AWS 解決方案架構師、AWS 參考架構和 AWS 合作夥伴網路 (APN) 合作夥伴，可根據產業知識協助您選取架構，不過必須使用透過基準化分析或負載測試獲得的資料，為架構進行優化。

您的架構可能會結合許多不同的架構方法 (例如，事件驅動、ETL 或管道)。實作架構將使用專屬於您的架構效能優化的 AWS 服務。在以下各節中，我們將討論您應該考慮的四大主要資源類型 (運算、儲存、資料庫和網路)。

運算

選擇符合您要求、效能需求並提供高效率成本和精力的運算資源，讓您能夠使用相同數量的資源來完成更多工作。評估運算選項時，請注意您對工作負載效能和成本的要求，並依據這些要求做出明智的決策。

在 AWS 中，提供了三種運算形式：執行個體、容器和函數。

- 執行個體 是虛擬伺服器，可讓您使用按鈕或 API 呼叫，來變更其功能。由於在雲端中，資源決策不是固定的，您可以使用不同的伺服器類型進行試驗。在 AWS 上，這些虛擬伺服器執行個體具有不同系列和大小，並且可提供眾多不同功能，包括固態硬碟 (SSD) 和圖形處理單元 (GPU)。
- 容器 是將作業系統虛擬化的一種方法，可讓您在隔離資源的程序中執行應用程式及其相依性。AWS Fargate 是容器的無伺服器運算，或者，如果您需要控制運算環境的安裝、組態和管理，則可使用 Amazon EC2。您也可以從多個容器協調平台中選擇：Amazon Elastic Container Service (ECS) 或 Amazon Elastic Kubernetes Service (EKS)。
- 函數 可從您想執行的程式碼中將執行環境抽象化。例如，AWS Lambda 可讓您無需執行執行個體便能執行程式碼。

下列問題著重於效能達成效率方面的這些考量。

PERF 2：您如何選取運算解決方案？

工作負載的最佳運算解決方案會根據應用程式設計、使用模式和組態設定而有所不同。架構可針對不同元件使用不同運算解決方案並啟用不同功能，以提升效能。為架構選錯運算解決方案，可能使效能達成效率降低。

在建立使用運算的架構時，您應利用可用的彈性機制來確保您有足夠的容量，可在需求變更時維持效能。

儲存

雲端儲存是雲端運算中很重要的元件之一，儲存了工作負載所使用的資訊。雲端儲存通常比傳統內部部署的儲存系統更為可靠、可擴展且安全。為您的工作負載選擇物件、區塊和檔案儲存服務，以及雲端資料遷移選項。

在 AWS 中，儲存有三種形式：物件、區塊和檔案：

- 物件儲存 提供可擴展且耐用的平台，以利從任何網際網路位置存取資料，例如使用者產生的內容、作用中存檔、無伺服器運算、大數據儲存或備份與復原。Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，可提供領先業界的可擴展性、資料可用性、安全性和效能。Amazon S3 的設計可提供 99.999999999% (11 個 9) 的耐久性，並為全球公司存放數百萬個應用程式資料。
- 區塊儲存 可為每個虛擬主機提供高可用性、一致性、低延遲的區塊儲存，而且類似於直接連結存放裝置 (DAS) 或存放區域網路 (SAN)。Amazon Elastic Block Store (Amazon EBS) 是專為需要 EC2 執行個體存取持久性儲存的工作負載所設計，可協助您以適當的儲存容量、效能和成本來調整應用程式。
- 檔案儲存 可讓您跨多個系統存取共用檔案系統。像 Amazon Elastic File System (EFS) 這類檔案儲存解決方案非常適合大型內容儲存庫、開發環境、媒體存放區或使用者主目錄等使用案例。Amazon FSx 可讓您以輕鬆且經濟實惠的方式啟動和執行熱門的檔案系統，因此您可以利用廣泛使用的開放原始碼和商業授權檔案系統的豐富功能集和快速效能。

下列問題著重於效能達成效率方面的這些考量。

PERF 3：您如何選取儲存解決方案？

系統的最佳儲存解決方案會根據存取方法類型 (區塊、檔案或物件)、存取模式 (隨機或連續)、所需傳輸量、存取頻率 (線上、離線、封存)、更新頻率 (WORM、動態) 及可用性和耐用性限制而有所不同。Well-Architected 系統使用多重儲存解決方案，並啟用不同功能以提升效能並有效使用資源。

當您要選取儲存解決方案時，務必確保其符合您的存取模式，以達到您想要的效能。

資料庫

雲端提供專門打造的資料庫服務，可解決工作負載呈現的不同問題。您可以從許多專門打造的資料庫引擎中選擇，包括關聯式、鍵值、文件、記憶體內、圖形、時間序列和總帳資料庫。透過挑選最佳資料庫來解決特定問題 (或一組問題)，您可以擺脫限制性的「一體適用」單體資料庫，並專注在建置應用程式以滿足客戶的效能需求。

在 AWS 中，您可以從多個專門打造的資料庫引擎中選擇，包括關聯式、鍵值、文件、記憶體內、圖形、時間序列和總帳資料庫。使用 AWS 資料庫，您無須擔心伺服器佈建、修補、設定、組態、備份或復原等資料庫管理任務。AWS 會持續監控您的叢集，透過自我修復的儲存和自動擴展來保持工作負載正常啟動和執行，讓您能專注於開發價值較高的應用程式。

下列問題著重於效能達成效率方面的這些考量。

PERF 4：您如何選擇資料庫解決方案？

系統的最佳資料庫解決方案可能會依可用性、一致性、分割容錯度、延遲、耐用性、可擴展性及查詢能力的需求而有所不同。許多系統針對不同子系統使用不同資料庫解決方案，並啟用不同功能以提升效能。為系統選錯資料庫解決方案和功能，可能使效能達成效率降低。

工作負載的資料庫方法對效能達成效率有重大影響。它通常是根據組織預設而非透過資料驅動的方法所選擇的區域。與儲存一樣，務必要考慮工作負載的存取模式，同時也務必要考慮其他非資料庫解決方案是否可以更有效地解決問題 (例如使用圖形、時間序列或記憶體中的儲存資料庫)。

網路

由於網路位於所有工作負載元件之間，因此對工作負載效能和行為都有極大的正面和負面影響。也有高度依賴網路效能的工作負載，例如高效能運算 (HPC)，其中深度了解網路對於提升叢集效能非常重要。您必須判斷頻寬、延遲、抖動和輸送量的工作負載需求。

在 AWS 上，聯網是虛擬化的，並提供多種不同的類型和組態。這讓您可以更輕鬆地將聯網方法與需求進行匹配。AWS 提供了多種產品功能 (例如，增強型聯網、經 Amazon EBS 優化的執行個體、Amazon S3 Transfer Acceleration 和動態 Amazon CloudFront)，可對網路流量進行優化。AWS 還提供了聯網功能 (例如，Amazon Route 53 延遲路由、Amazon VPC 端點、AWS Direct Connect 和 AWS Global Accelerator)，可減少網路距離或抖動。

下列問題著重於效能達成效率方面的這些考量。

PERF 5：您如何設定聯網解決方案？

工作負載的最佳網路解決方案會根據延遲、輸送量需求、抖動和頻寬而有所不同。實體限制 (例如使用者或內部部署資源) 會決定位置選項。這些限制條件可以隨著節點或資源置放而位移。

部署網路時，您必須考慮位置。您可以選擇將資源置放在靠近資源用以減少距離的位置。隨著工作負載的演進，使用聯網指標變更聯網組態。透過利用區域、置放群組和邊緣服務，您可以顯著提高效能。雲端型網路可以快速重建或修改，因此隨著時間演進您的網路架構是維持效能達成效率的必要條件。

審查

雲端技術正在快速發展，您必須確保工作負載元件會使用最新技術和方法，來持續改善效能。您必須持續評估並考量工作負載元件的變更，以確保符合其效能和成本目標。機器學習和人工智慧 (AI) 等新技術可讓您重新構思客戶體驗，並跨所有業務工作負載進行創新。

利用受客戶需求驅動的 AWS 持續創新。我們會定期發佈新的區域、節點、服務和功能。上述任何一個版本均可明顯提高架構的效能達成效率。

下列問題著重於效能達成效率方面的這些考量。

PERF 6：您如何發展工作負載，以運用新版本的優勢？

架構工作負載時，可選擇的選項有限。但一段時間後會有可改善工作負載效能的新技術和方法推出。

架構效能不佳通常是效能審查程序不存在或中斷的結果。如果您的架構效能不佳，則實作效能審查程序可讓您套用 Deming 的計畫 – 執行 – 檢查 – 行動 (PDCA) 週期，來推動迭代改善。

監控

實作工作負載後，您必須監控其效能，以便在其影響客戶前修正任何問題。超過閾值時，應使用監控指標來發出警示。

Amazon CloudWatch 是一種監控和可觀測性服務，可為您提供資料和可採取動作的洞見，以監控工作負載、回應整個系統的效能變更、優化資源使用率，以及取得運作狀況的統一檢視。CloudWatch 會從在 AWS 和內部部署伺服器上執行的工作負載中收集監控和作業資料 (形式為記錄、指標和事件)。AWS X-Ray 可協助開發人員分析並偵錯生產、分散式應用程式。透過 AWS X-Ray，您可以收集應用程式表現的洞見，並找出根本原因和效能瓶頸。您可以使用這些分析結果來快速即時做出反應，保持工作負載順暢運作。

下列問題著重於效能達成效率方面的這些考量。

PERF 7：您如何監控資源來確保達成預期效能？

系統效能可能會隨時間降低。監控系統效能以識別效能降低情況，並修復內部或外部因素，如作業系統或應用程式負載。

確保您未看到誤報，這是有效監控解決方案的關鍵。自動觸發可以避免人為錯誤，並可減少解決問題的時間。規劃在生產環境中進行模擬的演練日，以測試您的警示解決方案，並確保其能正確識別問題。

權衡

當您建立架構解決方案時，請考慮權衡，以確保採用了最佳方法。根據您的情況，您可以權衡一致性、耐用性和時間與延遲的空間，進而提高效能。

使用 AWS，您可以在數分鐘內實現全球化，並在全球多個位置部署資源，以更接近最終使用者。您還可以將唯讀複本動態新增至資訊儲存區 (例如資料庫系統)，以減少主要資料庫上的負載。

下列問題著重於效能達成效率方面的這些考量。

PERF 8：您如何採用權衡來增進效能？

架構解決方案時，判斷權衡項目可讓您選擇最佳方法。您通常可以透過權衡一致性、耐用性和時間與延遲的空間來提升效能。

在變更工作負載時，收集並評估指標以確定這些變更的影響。衡量對系統以及最終使用者的影響，以了解您的權衡如何影響您的工作負載。使用系統的方法 (例如負載測試) 來探索權衡是否可以提高效能。

資源

請參閱以下資源，進一步了解我們的效能達成效率最佳實務。

文件

- Amazon S3 效能優化
- Amazon EBS 磁碟區效能

白皮書

- 效能達成效率支柱

影片

- AWS re:Invent 2019：Amazon EC2 基礎 (CMP211-R2)

- AWS re:Invent 2019：領導者會議： 聯盟的儲存狀態 (STG201-L)
- AWS re:Invent 2019：領導者會議：AWS 專用資料庫 (DAT209-L)
- AWS re:Invent 2019：與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- AWS re:Invent 2019：支援下一代 Amazon EC2: 深入探討 Nitro 系統 (CMP303-R2)
- AWS re:Invent 2016：擴充至首個 1,000 萬名使用者 (ARC211-R)

成本最佳化

成本優化支柱包含在最低價格之下執行系統以產生商業價值的能力。

成本優化支柱概述了設計原則、最佳實務和相關問題。您可以在下列白皮書中找到規範指引：成本優化支柱白皮書。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端成本優化有五個設計原則：

- 實作雲端財務管理：為實現財務成功並加速在雲端實現商業價值，您需要投資雲端財務管理/成本優化。您的組織需要投入時間和資源，在這個新的技術與使用管理領域中打造能力。與安全或卓越營運能力類似，您需要透過知識累積、計畫、資源和程序打造能力，以成為具成本效率的組織。
- 採用消費模式：僅為您需要的運算資源付費，依照業務要求增減用量，不必倚賴複雜的預測。例如，開發與測試環境通常僅於一週工作日的一天八小時當中使用。您可在不使用這些資源時加以停止，有潛力可節省 75% 成本 (40 小時相對於 168 小時)。
- 衡量整體效率：測量工作負載的商業輸出和遞送的相關成本。以此測量值可得知您從增加輸出與降低成本獲取的增益。
- 停止將金錢花在繁重的無差別工作上：AWS 會處理資料中心營運的繁重工作，例如架設、堆疊和支援伺服器。通過受管服務，它也免除了管理作業系統和應用程式這些營運負擔。這可讓您專注於客戶和業務專案，而非 IT 基礎設施。

- 分析和歸因支出：採雲端式能更容易準確識別系統的用量和成本，繼而允許將 IT 成本透明化地歸因至個別工作負載擁有者。如此有助於測量投資報酬率 (ROI)，並且讓工作負載擁有者有機會優化資源和降低成本。

定義

雲端成本優化的最佳實務有五個方面：

- 實作雲端財務管理
- 支出和用量感知
- 具有經濟效益的資源
- 管理需求與供應資源
- 隨時間優化

如同 Well-Architected 架構內的其他支柱，有權衡事項需要考量，例如，該針對上市速度還是成本進行優化。在某些情況下，最好是針對速度來優化，例如快速上市、推出新功能，或只是滿足截止日期，而不是投資在預付成本優化。設計決策有時會因倉促而不是資料來引導，因為總是會有「以防萬一」過度補償的趨向，而不是花時間為最經濟實惠的部署做基準化分析測試。這恐怕會導致過度佈建和優化不足的部署。不過，若需要將內部部署環境內的資源「提升和轉移」至雲端，然後再實施優化，這是理性的選擇。前期對成本優化策略進行適當投資，並確保一致奉行最佳實務，避免不必要的過度佈建，可讓您更穩當地體現雲端的經濟效益。以下各節提供初始和持續實作工作負載雲端財務管理和成本優化的技術和最佳實務。

最佳實務

主題

實作雲端財務管理

採用雲端之後，技術團隊因核准、採購和基礎設施部署週期縮短而加快創新速度。實現商業價值和財務成功需要新的雲端財務管理方法。此方法為雲端財務管理，透過在整個組織實作知識建置、計畫、資源和程序，打造整個組織的能力。

許多組織是由許多不同的單位組成，每個單位都具有不同的優先事項。以下能力將協助建立更高效的組織：讓您的組織與一系列約定的財務目標保持一致，並為組織提供達成這些目標所需的機制。有能力的組織將更快速地創新和建立，且面對任何內部或外部因素時更靈活、適應性更強。

在 AWS 中，您可以使用 Cost Explorer 、Amazon Athena (選用) 和 Amazon QuickSight，搭配成本和用量報告 (CUR) 在整個組織中提供成本和用量感知。AWS 預算可針對成本和用量提供主動通知。AWS 部落格提供新服務和功能的相關資訊，確保您能夠隨時掌握最新的服務版本。

下列問題著重於成本優化方面的這些考量。(如需成本優化問題清單和最佳實務的清單，請參閱 附錄。)

COST 1：如何實作雲端財務管理？

透過實作雲端財務管理，組織可以透過優化成本和用量以及在 AWS 上進行規模調整，實現商業價值和財務上的成功。

建立成本優化職能部門時，使用團隊成員，並在團隊中增加 CFM 和成本優化方面的專家。現有的團隊成員將會了解組織目前的運作方式，以及如何快速實作改善。同時也考慮納入具有輔助或專業技能集的人員，例如分析和專案管理方面的人員。

在組織中實作成本感知時，改善現有的計畫和程序或在此基礎是上進行建置。在現有的程序和計畫中新增內容會比建立新的程序和計畫快得多。這會更快實現結果。

支出和用量感知

雲端提供的增強彈性和敏捷性，可促進創新和快節奏開發和部署。它消除了與佈建內部部署基礎設施相關的手動程序和時間，包括識別硬體規格、協商價格報價、管理採購訂單、安排裝運以及部署資源。然而，欲享有易用性和幾乎無限制的隨需容量，對於支柱需要換上新思維。

許多企業是以各種團隊執行多個系統之下所組成。能將資源成本歸因至個別組織或產品擁有者，能帶動高效使用的行為模式，有助於減少浪費。準確的成本歸因可讓您知道哪些產品具有真正的獲利能力，並就預算分配做出更明智的決策。

在 AWS 中，您可以使用 AWS Organizations 或 AWS Control Tower 來建立帳戶結構，如此可實現區隔並協助您分配成本和用量。您也可以對資源使用標記，利用商業和組織資訊確定用量和成本情況。使

用 AWS Cost Explorer 查看您的成本和用量，或使用 Amazon Athena 和 Amazon QuickSight 建立自訂儀表板和分析。透過 AWS 預算的通知，以及使用 AWS Identity and Access Management (IAM) 和 Service Quotas 的控制措施，控制成本和用量。

下列問題著重於成本優化方面的這些考量。

COST 2：您如何管控用量？

建立原則和機制以確保產生的成本合理，同時達成目標。您可以運用相互制衡的方法，在不超支的情況下創新。

COST 3：您如何監控用量和成本？

建立原則和程序以監控並適當分配成本。這可讓您衡量並改善此工作負載的成本效益。

COST 4：如何進行資源除役？

從啟動到結束專案期間，控制變更並管理資源。這可確保您關閉或終止未使用的資源，以減少浪費。

您可使用成本分配標籤為 AWS 用量和成本進行分類和追蹤。當您對 AWS 資源 (例如 EC2 執行個體或 S3 儲存貯體) 加上標籤時，AWS 就能以您的用量和標籤產生成本和使用報告。您可加上代表組織類別 (例如成本中心、工作負載名稱或擁有者) 的標籤，以便跨多項服務安排成本。

確保您在成本與用量報告和監控中使用正確的詳細資訊和精細度層級。如需高層級的洞見和趨勢，請透過 AWS Cost Explorer 使用每日精細度。如需更深入的分析和檢查，請使用 AWS Cost Explorer 中的每小時精細度，或 Amazon Athena 和搭配成本和用量報告 (CUR) 的 Amazon QuickSight 中的每小時精細度。

將加有標籤的資源結合實體生命週期追蹤 (員工、專案)，可找出不再為組織產生價值且應當除役的孤立資源或專案。您可以設定帳單提醒，通知您預測的超支。

具有經濟效益的資源

為您的工作負載使用適當的執行個體和資源，是節約成本的關鍵。例如，假設報告程序在較小的伺服器上執行時要花五小時，但在兩倍昂貴的較大伺服器上執行只需一小時。這兩種伺服器產出的結果相同，但較小的伺服器經過一段時間會形成較高成本。

架構完善的工作負載會用最具成本效益的資源，帶來明顯正面的經濟影響。您並有機會可利用受管服務來降低成本。例如，與其維護伺服器以遞送電子郵件，可使用以訊息為單位收費的服務。

AWS 備有各種具有彈性且經濟的定價選項，讓您以最符合需要的方式獲取 Amazon EC2 和其他服務的執行個體。隨需 執行個體 讓您可以按時數為運算容量付費，無最低承諾的要求。 Savings Plans 和預留執行個體 與隨需定價相較，可節省高達 75% 的成本。使用 Spot 執行個體，您可善用未用的 Amazon EC2 容量，與隨需定價相較可節省高達 90% 的成本。Spot 執行個體 適合用在系統能耐受使用伺服器叢集之處，其中個別伺服器能動態性地來去，例如無狀態 Web 伺服器、批次處理，或使用 HPC 和大型資料時。

選擇適當的服務也能降低用量和成本；例如 CloudFront 能將資料傳輸降至最低，甚至完全消除成本，例如在 RDS 上利用 Amazon Aurora 免於昂貴的資料庫授權成本。

下列問題著重於成本優化方面的這些考量。

COST 5：您選擇服務時如何評估成本？

Amazon EC2、Amazon EBS 和 Amazon S3 是基礎 AWS 服務。Amazon RDS 和 Amazon DynamoDB 等受管服務為更高等級或應用程式等級的 AWS 服務。選擇適當的基礎和受管服務，您便可為成本優化此工作負載。舉例來說，您可以使用受管服務減少或省下大部分管理和營運開銷，讓您從事應用程式或企業相關活動。

COST 6：您選擇資源類型、大小和數量時，如何達成成本目標？

確保您為手上的任務選擇適當的資源大小和資源數量。您透過選擇最具成本效益的類型、大小和數量，最大限度地減少浪費。

COST 7：您如何使用定價模式降低成本？

使用最適合您資源的定價模式，大幅減少支出。

COST 8：您如何規劃資料傳輸費？

務必規劃和監控資料傳輸費，以便做出可大幅減少成本的架構決策。小但有效的架構變更可隨時間大幅減少營運成本。

透過在選擇服務時考慮成本因素，並以 Cost Explorer 和 AWS Trusted Advisor 等工具定期審查 AWS 用量，您可積極監測使用率，並隨之調整部署。

管理需求與供應資源

待您移至雲端後，即可僅為所需付費。您可以在需要時供應資源以符合工作負載需求，避免因過度佈建付出高昂成本和造成浪費。您也可以使用調節、緩衝區或佇列來修改需求，以讓需求變得平緩，並以較少的資源來滿足需求，從而降低成本，或稍後使用批次服務來處理。

在 AWS 中，您可自動佈建資源以符合工作負載需求。Auto Scaling 使用基於需求或時間的方法，讓您可以視需要新增和移除資源。若您能預期需求變更，則可省下更多成本，並確保資源符合工作負載需求。您可以使用 Amazon API Gateway 實作調節，或使用 Amazon SQS 在工作負載中實作佇列。這兩者都可讓您修改工作負載元件的需求。

下列問題著重於成本優化方面的這些考量。

COST 9：如何管理需求和供應資源？

針對支出和效能達到平衡的工作負載，請確保使用購買的每個項目，並避免極少使用執行個體。往任一端傾斜的使用指標，對您組織在營運成本 (因過度使用而降低效能) 或浪費的 AWS 花費 (因過度佈建) 方面會造成負面影響。

在設計修改需求與供給資源時，請主動思考用量模式、佈建新資源所需的時間，以及需求模式的可預測性。管理需求時，請確保您的佇列或緩衝區大小正確，而且在所需的時間內回應工作負載需求。

隨時間優化

隨著 AWS 推出新服務和功能，最佳實務是檢視現有架構決策，以確保持續發揮最大成本效益。隨著您的要求變更，請主動將不再需要的資源、整項服務和系統加以除役。

透過實作新功能或資源類型可逐步優化工作負載，同時盡量減少實作變更所需的工作量。這可隨著時間持續提高效率，並確保您持續使用最新的技術來降低營運成本。您也可以使用新的服務來取代工作負載

中的元件，或將新元件新增至工作負載中。這可以大幅提高效率，因此定期檢閱工作負載並實作新服務和功能至關重要。

下列問題著重於成本優化方面的這些考量。

COST 10：您如何評估新服務？

隨著 AWS 推出新服務和功能，最佳實務是檢視現有架構決策，以確保持續發揮最大成本效益。

在定期審查您的部署時，請評估較新的服務能如何為您節省成本。例如，RDS 上的 Amazon Aurora 能降低關聯式資料庫的成本。使用 Lambda 等無伺服器函數時，無需操作和管理執行個體來執行程式碼。

資源

請參閱以下資源，進一步了解我們成本優化的最佳實務。

文件

- AWS 文件

白皮書

- 成本優化支柱

https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html?ref=wellarchitected-wp

主題

-
-
-
-
-

-
-
-
-
-
-

主題

-
-
-
-
-
-
-

???

- https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html?ref=wellarchitected-wp

- https://www.youtube.com/watch?v=oz9iO0EOpI0&ref=wellarchitected-wp

審查程序

架構審查的執行方式必須一致，採行鼓勵深入探索的無譴責作法。應為輕量程序 (數小時而非數日)，屬於一種對話而非稽核。就架構進行審查的目的是找出可能需要解決的重要問題，或是有改進空間之處。審查的結果是一套行動，應能提升客戶使用工作負載得到的體驗。

如同「論架構」一節所討論，建議由各團隊成員對其架構的品質負起責任。我們建議建置架構的團隊成員使用 Well-Architected 架構以持續審查其架構，而非舉行正式審查會議。採取持續作法可讓您的團隊成員隨著架構演進更新答案，並隨著您遞送功能而提升架構。

AWS Well-Architected Framework 符合 AWS 於內部審查系統與服務的方式。其所根據的前提為能影響架構方針的一套設計原則，並提出問題，確保人員不致於忽略根本原因分析 (RCA) 中經常列為重點的領域。每當內部系統、AWS 服務或客戶有明顯問題，我們都會查看 RCA，了解是否能提升所使用的審查程序。

審查應在產品生命週期的重要里程碑，並於設計階段早期實施，以免成為 單向門戶 難以變更，而且需趕在正式運作日期之前。(許多決定為可逆的雙向門戶。這些決定可採用輕量程序。單向門戶難以、甚至無法逆轉，實施之前需要更多檢查工作。) 進入生產環境之後，您的工作負載可隨著新增功能和變更技術實作而繼續演進。工作負載的架構會隨時間而變化。您需要遵守良好的衛生實務，以阻止您推動演進的同時，其架構上的特性隨之衰退。在您作出重要的架構變更時，應遵照一套衛生程序，包括 Well-Architected 審查。

若您想以審查作為一次性的快照或獨立測量，建議確定在對話中包含所有適當人員。我們經常發現，到審查時團隊才初次真正了解實作了些什麼。審查另一個團隊的工作負載時，一種效果良好的方式是就其架構進行一連串非正式對話，能探詢出大多數問題的答案。接著您即可透過一兩次會議進行追蹤，釐清或深入探索模稜兩可或看出有風險的領域。

開會時的一些建議項目如下：

- 有白板的會議室
- 任何圖或設計備註的列印紙本
- 需要另外研究答案的問題動議清單 (例如「我們有無啟用加密？」)

在您完成審查之後，應列有問題清單，可根據業務環境排列優先順序。也建議考量這些問題對於您的團隊之日常工作有何影響。若您及早解決這些問題，即可空出時間創造商業價值，不必忙於解決重複發生的問題。當您解決問題時，可以更新審查，了解架構改良的情形。

雖然審查完成後，其價值所在自然明朗，但您可能會發現新的團隊起初可能會有所抗拒。經由對團隊教育審查的益處，可解決下列幾項反對說法：

- 「我們太忙了!」(團隊預備進行盛大推出時，往往會這麼說。)
  - 既然預備進行盛大推出，一定希望過程能夠順利。審查可讓您了解可能漏掉的任何問題。
  - 建議您在產品生命週期之中及早實施審查，以發現風險並開發配合功能遞送藍圖的減緩計劃。
- 「我們沒有時間處理結果！」(往往在作為目標的活動無法挪動，例如超級盃時會這麼說。)
  - 這些活動無法挪動。您是否真的想在對於架構所具風險不知情的情況下迎接活動？ 就算無法解決所有的問題，仍然可在發生狀況時握有處理問題的程序手冊。
- 「我們不想讓解決方案實作的秘密外流！」
  - 如果您向團隊指出 Well-Architected Framework 中的疑問，他們就能看出這些疑問完全不會顯露商業或技術專屬資訊。

在您與組織內的團隊實施多重審查之時，可能會找出主題上的問題。例如，可能會發現一群團隊的問題集中在特定支柱或主題上。建議以全面方式審視所有的審查，並找出有助於解決這些主題問題的任何機制、培訓或首席工程設計對談。

結論

AWS Well-Architected Framework 提供了遍及六大支柱的架構最佳實務，用於設計和營運可靠、安全、有效率、經濟實惠且永續發展的雲端系統。該架構提供一套問題，允許您審查現有或提議的架構。其也為各支柱提供一套 AWS 最佳實務。在您的架構中使用該架構可協助您產生穩定且有效率的系統，讓您能夠專注於功能需求。

作者群

協力完成本文件的個人與組織如下：

- Brian Carlson：Amazon Web Services Well-Architected 營運主管
- Ben Potter，Amazon Web Services Well-Architected 安全主管
- Seth Eliot，Amazon Web Services Well-Architected 可靠性主管
- Eric Pullen，Sr.Amazon Web Services 資深解決方案架構師
- Rodney Lester，Amazon Web Services 首席解決方案架構師
- Jon Steele：Amazon Web Services 資深技術客戶經理
- Max Ramsay，Amazon Web Services 首席安全解決方案架構師
- Callum Hughes，Ronnen Slasky，Amazon Web Services 解決方案架構師
- Philip Fitzsimons，Amazon Web Services Well-Architected 內容程式經理

深入閱讀

AWS 架構中心

AWS 雲端合規

AWS Well-Architected 合作夥伴計劃

AWS Well-Architected Tool

AWS Well-Architected 首頁

卓越營運支柱白皮書

安全支柱白皮書

可靠性支柱白皮書

效能達成效率支柱白皮書

成本優化支柱白皮書

永續性支柱白皮書

在 Amazon Builders' Library 中

文件修訂

若要收到此白皮書更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
小幅度更新	已在附錄中新增工作量的定義和更新最佳實務。	October 20, 2022
白皮書已更新	已新增永續性支柱和更新的連結。	December 2, 2021
???		November 20, 2021
小幅度更新	已移除非包容性語言。	April 22, 2021
小幅度更新	已修正數個連結。	March 10, 2021
小幅度更新	整體的小幅度編輯變更。	July 15, 2020
新框架的更新	檢閱和重寫大多數問題和答案。	July 8, 2020
白皮書已更新	新增 AWS Well-Architected Tool，連結至 AWS Well-Architected 實驗室及 AWS Well-Architected 合作夥伴、小處修復以促成架構有多種語言版本。	July 1, 2019
白皮書已更新	審查並重新撰寫大多數的問題和答案，以確保問題一次聚焦在一個主題之上。這使得部分先前的問題分為數個問題。新增定義的共同詞彙 (工作負載、元件等)。變更主要本文中的問題呈現，以含入描述性文字。	November 1, 2018

白皮書已更新	更新以簡化問題文字，將答案標準化，並提升可讀性。	June 1, 2018
白皮書已更新	卓越營運移至支柱前端並重新撰寫，使其成為其他支柱的框架。重新整理其他支柱，以反映 AWS 的演進。	November 1, 2017
白皮書已更新	更新架構以含入卓越營運支柱，並修訂及更新其他支柱以減少重複，並納入與數千客戶一同執行審查之所學。	November 1, 2016
小幅度更新	使用目前的 Amazon CloudWatch Logs 資訊更新了附錄。	November 1, 2015
初版	已發佈 AWS Well-Architected Framework。	October 1, 2015

附錄：問題與最佳實務

主題

卓越營運

主題

組織

問題

OPS 1　如何決定您的優先事項？

每個人都必須了解自己在實現商業價值過程中的角色。擁有共同目標以設定資源優先順序。這會充分發揮您所做努力的優勢。

最佳實務

- OPS01-BP02 評估內部客戶需求
- OPS01-BP03 評估管控要求
- OPS01-BP04 評估合規要求
- OPS01-BP05 評估威脅態勢
- OPS01-BP06 評估權衡
- OPS01-BP07 管理收益和風險

OPS01-BP01 評估外部客戶需求

讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定將工作重點放在哪些外部客戶需求上。這將確保您對實現想要的業務成果所需的營運支援有透徹的了解。

常用的反模式：

- 您已決定不在核心上班時間以外的時間提供客戶支援，但尚未檢閱歷史支援請求資料。您不知道這是否會影響您的客戶。
- 您正在開發新功能，但尚未與客戶互動，以了解是否需要該功能，若需要又應以何種形式提供，而且未進行試驗以驗證交付的需求和方法。

建立此最佳實務的優勢： 需求獲得滿足的客戶更有可能持續回購。評估和了解外部客戶的需求，將讓您了解如何安排工作的優先順序來實現商業價值。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 了解業務需求：只有業務、開發及營運團隊等利害關係人擁有共同的目標並達成共識，方能讓業務取得成功。
  - 審查外部客戶的業務目標、需求和優先事項：與關鍵利害關係人 (包括業務、開發和營運團隊) 進行互動，以討論外部客戶的目標、需求和優先事項。這將確保您對實現業務和客戶成果所需的營運支援有透徹的了解。
  - 建立共識：在以下方面建立共識：工作負載的業務功能、每個團隊在工作負載營運過程中的角色，以及這些因素如何支援內外部客戶的共同業務目標。

資源

相關文件：

- AWS Well-Architected Framework 概念 – 反饋迴圈

OPS01-BP02 評估內部客戶需求

讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定將工作重點放在哪些內部客戶需求上。這將確保您對實現業務成果所需的營運支援有透徹的了解。

利用您制定的優先事項，聚焦於改善作業，因為它們能發揮最大的影響力 (例如，發展團隊技能、改善工作負載效能、降低成本、自動化執行手冊或提升監控力)。根據需求變更更新您的優先順序。

常用的反模式：

- 您已決定在不向產品團隊諮詢的情況下，變更他們的 IP 地址配置，以便更輕鬆地管理網路。您不知道這會對您的產品團隊造成什麼影響。
- 您正在實作新的開發工具，但尚未與內部客戶互動，以了解是否需要該工具或其是否與現有的實務相容。
- 您正在實作新的監控系統，但尚未聯絡內部客戶，以了解他們是否有應該考慮的監控或報告需求。

建立此最佳實務的優勢： 評估和了解內部客戶的需求，將讓您了解如何安排工作的優先順序來實現商業價值。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 了解業務需求：只有業務、開發及營運團隊等利害關係人擁有共同的目標並達成共識，方能讓業務取得成功。
  - 審查內部客戶的業務目標、需求和優先事項：與關鍵利害關係人 (包括業務、開發和營運團隊) 進行互動，以討論內部客戶的目標、需求和優先事項。這將確保您對實現業務和客戶成果所需的營運支援有透徹的了解。
  - 建立共識：在以下方面建立共識：工作負載的業務功能、每個團隊在工作負載營運過程中的角色，以及這些因素如何支援內外部客戶的共同業務目標。

資源

相關文件：

- AWS Well-Architected Framework 概念 – 反饋迴圈

OPS01-BP03 評估管控要求

確保您了解由貴組織所定義的、可能要求或強調特定重點的準則或義務。評估內部因素，例如組織政策、標準和要求。確認您設有確定管控變更的機制。如果未確定管控要求，請確保您已對此決定進行盡職調查。

常用的反模式：

- 您正在接受稽核，並經要求需提供內部管控的合規證明。您從未評估合規要求，因此您不知道自己是否合規。
- 您已遭受導致經濟損失的洩露。您發現，可能涵蓋經濟損失的保險取決於您是否實作了特定安全控制措施，而您的管控要求實作這些控制措施，但其尚未落實到位。
- 您的管理帳戶遭到入侵，導致公司網站遭到破壞，並使得客戶信任受損。您的內部管控要求使用多重因素驗證 (MFA) 來保護管理帳戶的安全。您未使用 MFA 保護管理帳戶的安全，可能受到處罰。

建立此最佳實務的優勢： 評估和了解組織套用到工作負載的管控要求，將可讓您了解如何安排工作的優先順序來實現商業價值。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 了解管控要求：評估內部管控因素，例如，計畫或組織政策、計畫政策、問題或系統特定政策、標準、程序、基準和準則。確認您設有確定管控變更的機制。如果未確定管控要求，請確保您已對此決定進行盡職調查。

資源

相關文件：

- AWS 雲端 合規

OPS01-BP04 評估合規要求

評估外部因素，例如合規要求和產業標準，以確保您了解可能要求或強調特定重點的準則或義務。如果未確定合規要求，請確保對此決定進行盡職調查。

常用的反模式：

- 您正在接受稽核，並經要求需提供產業規範的合規證明。您從未評估合規要求，因此您不知道自己是否合規。
- 您的管理帳戶遭到入侵，導致客戶資料遭到下載，並使客戶信任受損。您的產業最佳實務要求使用 MFA 來保護管理帳戶的安全。您未使用 MFA 保護管理帳戶的安全，可能受到客戶訴訟。

建立此最佳實務的優勢： 評估和了解套用到工作負載的合規要求，可讓您了解如何安排工作的優先順序來實現商業價值。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 了解合規要求：評估外部因素，例如合規要求和產業標準，以確保您了解可能要求或強調特定重點的準則或義務。如果未確定合規要求，請確保已對此決定進行盡職調查。
  - 了解法規合規要求：確定您在法律上有義務滿足的法規合規要求。根據這些要求來找到工作重點。範例包括隱私權和資料保護法規定的義務。
    - AWS 合規
    - AWS 合規計劃
    - AWS 合規最新資訊
  - 了解產業標準和最佳實務：識別適用於工作負載的產業標準和最佳實務要求，例如，支付卡產業資料安全標準 (PCI DSS)。根據這些要求來找到工作重點。
    - AWS 合規計劃
  - 了解內部合規要求：確定您的組織建立的合規要求和最佳實務。根據這些要求來找到工作重點。範例包括資訊安全政策和資料分類標準。

資源

相關文件：

- AWS 雲端 合規
- AWS 合規
- AWS 合規最新資訊
- AWS 合規計劃

OPS01-BP05 評估威脅態勢

評估對業務的威脅 (例如，競爭、業務風險和負債、營運風險和資訊安全威脅)，並將最新的資訊保存在風險登記表內。決定工作重點的領域時，加入風險影響。

AWS Well-Architected 架構 強調 學習、衡量和改善。它為您提供可評估架構並實作將隨時間擴展之設計的一致方法。AWS 提供 AWS Well-Architected Tool ，以協助您在部署前檢閱方法、在生產前檢閱工作負載狀態，以及檢閱生產中的工作負載狀態。您可以將它們與最新的 AWS 架構最佳實務做比較、監控工作負載的整體狀態，以及深入了解潛在風險。

AWS 客戶還有資格獲得對其關鍵任務工作負載的指導式 Well-Architected 審查， 進而依循 AWS 最佳實務 衡量其架構。企業支援客戶有資格進行 營運審查，該審查旨在助其識別在雲端營運的方法的差距。

這些審查的跨團隊參與有助於建立對您的工作負載以及團隊角色可如何助力成功的共識。透過審查識別的需求可以助您確定優先順序。

AWS Trusted Advisor 是一款可存取核心檢查集的工具，這些檢查提出了優化建議，可能有助您確定優先事項。 商業和企業支援客戶 可存取針對安全性、可靠性、效能和成本優化的其他檢查，從而進一步協助確定他們的優先事項。

常用的反模式：

- 您在產品中使用舊版的軟體程式庫。您不知道，程式庫的安全性更新是否存在可能對工作負載產生意外影響的問題。
- 您的競爭對手剛發佈的產品版本，可解決客戶對您產品的許多抱怨。您尚未排定處理這些已知問題之事項的優先順序。
- 監管機構一直在追尋像您這樣不符合法律法規合規要求的公司。您尚未排定處理任何未解決合規要求之事項的優先順序。

建立此最佳實務的優勢： 識別和了解組織和工作負載所面臨的威脅，讓您可以判斷要解決哪些威脅、它們的優先順序，以及執行此作業所需的資源。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 評估威脅態勢：評估對業務的威脅 (例如，競爭、業務風險和負債、營運風險和資訊安全威脅)，以便您在決定工作重點時考量其影響。

- AWS 最新安全公告
- AWS Trusted Advisor
- 維護威脅模型：建立和維護用於識別潛在威脅、已規劃和就地緩解措施及其優先順序的威脅模型。審查顯示為事件的威脅的機率、從這些事件中復原的成本、導致的預期傷害，以及防止這些事件的成本。當威脅模型的內容變更時，修改優先順序。

資源

相關文件：

- AWS 雲端 合規
- AWS 最新安全公告
- AWS Trusted Advisor

OPS01-BP06 評估權衡

評估在相互衝突的利益或替代方法之間做出權衡的影響，以幫助您在確定工作重點或選擇行動方案時做出明智的決定。例如，新功能加速上市可能是成本優化所強調的重點，或您可為非關聯式資料選擇關聯式資料庫，以便更輕鬆地遷移系統，而非遷移至針對您的資料類型優化的資料庫並更新您的應用程式。

AWS 可以協助您教育您的團隊有關 AWS 及其服務的知識，從而增進他們對自己的選擇會如何影響工作負載的了解。您應使用 AWS Support (AWS 知識中心， AWS 開發論壇和 AWS Support中心) 和 AWS 文件 資源來教育您的團隊。透過 AWS Support中心聯絡 AWS Support，以獲取 AWS 相關問題的幫助。

AWS 也分享了我們透過 在 Amazon Builders' Library 中營運 AWS 所學到的最佳實務和模式。您可透過 AWS 部落格 和 官方 AWS 播客。

常用的反模式：

- 您使用關聯式資料庫來管理時間序列和非關聯式資料。有針對支援您使用的資料類型進行最佳化的資料庫選項，但您並不了解其優點，因為您尚未評估解決方案之間的權衡。
- 您的投資者要求您證明支付卡產業資料安全標準 (PCI DSS) 的合規性。您沒有考量滿足要求和繼續您目前開發工作之間的權衡取捨。相反地，您繼續開發工作，而不證明合規性。由於對平台安全性及其投資的擔憂，您的投資者會停止對公司的支援。

建立此最佳實務的優勢： 了解您所選擇的影響和後果，讓您可以排定選項的優先順序。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 評估權衡：評估在相互衝突的利益之間做出權衡的影響，以幫助您在確定工作重點時做出明智的決定。例如，相比成本優化更強調新功能加速上市。
- AWS 可以協助您教育您的團隊有關 AWS 及其服務的知識，從而增進他們對自己的選擇會如何影響工作負載的了解。您應使用 AWS Support (AWS 知識中心、AWS 論壇和 AWS Support 中心) 和 AWS 文件中的資源來教育您的團隊。透過 AWS Support中心聯絡 AWS Support，以獲取 AWS 相關問題的幫助。
- AWS 也分享了我們透過在 Amazon Builders' Library 中營運 AWS 所學到的最佳實務和模式。您可透過 AWS 部落格和官方 AWS 播客獲得其他各種實用資訊。

資源

相關文件：

- AWS 部落格
- AWS 雲端 合規
- AWS 開發論壇
- AWS 文件
- AWS 知識中心
- AWS Support
- AWS Support中心
- 在 Amazon Builders' Library 中
- 官方 AWS 播客

OPS01-BP07 管理收益和風險

管理收益和風險，以便在確定工作重點時做出明智的決定。例如，部署具有未解決問題的工作負載可能有益，以便可以為客戶提供重要的新功能。相關風險可能得以減輕，也可能出現無法接受風險存在的事實，在此情況下，您將需要採取動作來解決風險。

您可能會發現，您在某個時間點會想要強調一小部分的優先事項。長期利用平衡的方法，以確保開發所需的功能和管理風險。根據需求變更更新您的優先順序

常用的反模式：

- 您已決定包含一個程式庫，該程式庫會執行其中一個開發人員在網際網路上找到的您所需的任何項目。您尚未評估從未知來源採用此程式庫的風險，並且不知道它是否包含弱點或惡意程式碼。
- 您已決定開發和部署新功能，而不是修正現有問題。在部署功能之前，您一直未評估將問題置之不理的風險，而且不知道會對客戶造成什麼影響。
- 由於合規團隊的不明疑慮，您決定不部署客戶經常請求的功能。

建立此最佳實務的優勢： 識別您選擇的可用優勢，並了解組織面臨的風險，讓您可以做出明智的決策。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 管理收益和風險：在決策的收益與所涉及的風險之間取得平衡。
  - 確定收益：根據業務目標、需求和優先事項確定收益。範例包括上市時間、安全性、可靠性、效能和成本。
  - 確定風險：根據業務目標、需求和優先事項確定風險。範例包括上市時間、安全性、可靠性、效能和成本。
  - 評估風險與收益並做出明智決定：根據關鍵利害關係人 (包括業務、開發和營運團隊) 的目標、需求和優先事項，確定收益和風險的影響。評價收益的價值時要考慮發生風險的可能性及其代價。例如，強調上市速度優先於可靠性，可能提供競爭優勢。不過，如果發生可靠性問題，則可能會縮短正常執行時間。

OPS 2 如何建構組織以支援業務成果？

您的團隊必須了解其在達成業務成果中所扮演的角色。團隊需要了解自己在促成其他團隊成功的過程中所扮演的角色、其他團隊在促進其成功的過程中所扮演的角色，以及擁有共同目標。了解責任、擁有權、決策方式，以及誰有權制定決策，將有助於找到工作重點，並充分發揮團隊的優勢。

最佳實務

- OPS02-BP01 已為資源識別擁有者
- OPS02-BP02 已為流程和程序識別擁有者
- OPS02-BP03 已為營運活動識別負責其效能的擁有者
- OPS02-BP04 團隊成員知道他們負責的項目
- OPS02-BP05 存在機制用來識別責任和擁有權
- OPS02-BP06 存在用於請求新增、變更和例外狀況的機制

- OPS02-BP07 團隊之間的責任為預先定義或經過協商

OPS02-BP01 已為資源識別擁有者

了解誰擁有各個應用程式、工作負載、平台和基礎設施元件、該元件提供什麼商業價值，以及該擁有權為何存在。透過了解這些個別元件的商業價值，以及其如何支援業務成果，可得知對元件套用的流程和程序。

建立此最佳實務的優勢： 透過了解擁有權，可識別誰可以核准改進項目和/或實作這些改進項目。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 已為資源識別擁有者：定義擁有權對環境中的資源使用案例的意義。指定並記錄資源的擁有者，至少包括名稱、聯絡資訊、組織和團隊。借助使用中繼資料 (例如標籤或資源群組) 的資源來儲存資源擁有權資訊。使用 AWS Organizations 建構帳戶並實作政策，以確保擷取擁有權和聯絡資訊。
  - 定義擁有權形式及其指派方式：擁有權在您具有不同使用案例的組織中可能有多個定義。您可能希望將工作負載擁有者定義為承擔工作負載營運風險和責任的個人，以及最終有權對工作負載做出決策的個人。在將擁有權累計到父組織時，您可能希望根據財務或管理責任來定義擁有權。開發人員可能是其開發環境的擁有者，並負責其操作造成的事件。他們的產品主管可能自行負擔與開發環境營運相關的財務成本。
  - 定義擁有組織、帳戶、資源集合或個別元件的人員：在可適當存取的位置中定義和記錄擁有權，該位置已經過組織，可支援探索。更新變更的定義和擁有權詳細資訊。
  - 擷取資源中繼資料中的擁有權：使用標籤或資源群組等中繼資料擷取資源擁有權，並指定擁有權和聯絡資訊。使用 AWS Organizations 建構帳戶並確保擷取擁有權和聯絡資訊。

OPS02-BP02 已為流程和程序識別擁有者

了解誰具有個別流程和程序的擁有權、為何使用特定流程和程序，以及為何該擁有權存在。了解使用特定流程和程序的原因，能夠幫助發現改進機會。

建立此最佳實務的優勢： 透過了解擁有權，可識別誰可以核准改進項目和/或實作這些改進項目。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 已為流程和程序識別負責其定義的擁有者：擷取環境中使用的流程和程序，以及負責其定義的個人或團隊。

- 識別流程和程序：識別為支援工作負載所執行的營運活動。將這些活動記錄在可探索的位置中。
- 定義擁有流程或程序定義的人員：唯一識別負責活動規格的個人或團隊。他們負責確保具備適當技能的團隊成員能夠成功執行該活動，且該團隊成員具備正確許可、存取權和工具。如果執行該活動時發生問題，執行該活動的團隊成員需負責提供改善活動所需的詳細回饋。
- 擷取活動成品中繼資料中的擁有權：在 AWS Systems Manager 等服務中，透過文件和做為函數的 AWS Lambda 自動化的程序，支援以標籤形式擷取中繼資料資訊。使用標籤或資源群組擷取資源擁有權，並指定擁有權和聯絡資訊。使用 AWS Organizations 建立標記政策，並確保擷取擁有權和聯絡資訊。

OPS02-BP03 已為營運活動識別負責其效能的擁有者

了解誰負責在已定義的工作負載上執行特定活動，以及為什麼該責任存在。透過了解誰負責執行活動，可得知誰將會進行活動、驗證結果，以及提供回饋給活動擁有者。

建立此最佳實務的優勢： 透過了解誰負責執行活動，可得知在需要採取動作時通知誰，以及誰將會執行動作、驗證結果，以及提供回饋給活動擁有者。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 已為營運活動識別負責其效能的擁有者：擷取執行環境中所使用之流程和程序的責任
  - 識別流程和程序：識別為支援工作負載所執行的營運活動。將這些活動記錄在可探索的位置中。
  - 定義負責執行每個活動的人員：識別負責活動的團隊。確保他們擁有活動的詳細資訊，以及執行活動所需的技能和正確的許可、存取權和工具。他們必須了解活動執行條件 (例如，事件或排程)。讓此資訊可供探索，如此組織的成員便能夠識別他們針對特定需求需要聯絡的人員 (團隊或個人)。

OPS02-BP04 團隊成員知道他們負責的項目

透過了解您角色的責任以及您為業務成果做出貢獻的方式，可得知任務的優先順序以及您的角色為何很重要。如此可讓團隊成員辨識需求並適當地回應。

建立此最佳實務的優勢： 透過了解您的責任，可得知您所做的決定、您採取的動作，以及如何將活動交給其適當的擁有者。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 確保團隊成員了解其角色和責任：識別團隊成員的角色和責任，並確保他們了解其角色的期望。讓此資訊可供探索，如此組織的成員便能夠識別他們針對特定需求需要聯絡的人員 (團隊或個人)。

OPS02-BP05 存在機制用來識別責任和擁有權

如果沒有識別個人或團隊，就會有定義的向某人向上呈報的路徑，該人員有權指派擁有權或為需解決的需求進行規劃。

建立此最佳實務的優勢： 透過了解有責任或擁有權的人員，讓您可以聯絡適當的團隊或團隊成員，以提出請求或轉換任務。擁有有權指派責任或擁有權或為解決需求進行規劃的已識別人員，可降低無作為和需求得不到解決的風險。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 存在機制用來識別責任和擁有權：為您的組織成員提供可存取的機制，以探索和識別擁有權和責任。這些機制讓他們可以針對特定需求識別要聯絡的人員 (團隊或個人)。

OPS02-BP06 存在用於請求新增、變更和例外狀況的機制

您可以向流程、程序和資源的擁有者提出請求。評估收益和風險後，若可行並經判斷是合適的行為，則應制定明智的決策以核准請求。

建立此最佳實務的優勢： 機制存在的目的應為請求新增、變更和例外狀況，以支援團隊的活動。如果沒有此選項，目前狀態就會成為創新的限制。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 存在用於請求新增、變更和例外狀況的機制：當標準很嚴格時，創新會受到限制。為您的組織成員提供機制，讓他們可向流程、程序和資源的擁有者提出請求，以支援其業務需求。

OPS02-BP07 團隊之間的責任為預先定義或經過協商

團隊間已定義或協商說明如何相互配合及支援的協議 (例如，回應時間、服務等級目標或服務等級協議)。透過了解團隊工作對於業務成果和其他團隊及組織成果的影響，可得知其任務的優先順序，並讓他們能做出適當的回應。

如果責任和擁有權未定義或未知，則您會面臨風險，不僅無法及時處理必要的活動，在解決這些需求時還會出現冗餘和可能相互衝突的工作。

建立此最佳實務的優勢： 透過建立團隊、目標和溝通需求之方法之間的責任，可簡化請求的流程，並協助確保提供必要的資訊。此舉可減少團隊之間的轉換任務所造成的延遲，並協助支援達成業務成果。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 團隊之間的責任為預先定義或經過協商：指定團隊互動的方法以及彼此支援所需的資訊，有助於將請求反覆審查和釐清時產生的延遲降到最低。擁有定義期望的特定協議 (例如，回應時間或履行時間)，讓團隊可以適當地制定有效的計畫和資源。

OPS 3 您的組織文化如何支援您的業務成果？

為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。

最佳實務

- OPS03-BP01 高層的支持
- OPS03-BP02 授權團隊成員在成果有風險時採取動作
- OPS03-BP03 鼓勵向上呈報
- OPS03-BP04 溝通需及時、清楚且可行
- OPS03-BP05 鼓勵進行試驗
- OPS03-BP06 團隊成員得以並受到鼓勵來維持和發展自己的技能集
- OPS03-BP07 適當地為團隊提供資源
- OPS03-BP08 鼓勵並尋求來自團隊內部和跨團隊的多樣化建議

OPS03-BP01 高層的支持

資深領導階層清楚地設定對組織的期望並評估成功情況。資深領導階層是採用最佳實務和組織演進的發起者、倡導者和推動者

建立此最佳實務的優勢： 積極參與的領導階層、清楚傳達的期望和共同目標，能夠確保團隊成員知道對他們的期望。評估成功情況可識別成功的障礙，因此可藉由發起者、倡導者及其代表的介入來解決這些障礙。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 高層的支持：資深領導階層清楚設定對組織的期望並評估成功情況。資深領導階層是採用最佳實務和組織演進的發起者、倡導者和推動者
  - 設定期望：為您的組織定義和發佈目標，包括衡量目標的方式。
  - 追蹤目標的達成情況：定期衡量目標逐步達成的情況，並分享結果，以便在成果有風險時採取適當的動作。
  - 提供實現目標所需的資源：根據新資訊、目標的變更、責任或您的業務環境等，定期檢閱資源是否仍然適當，或者是否需要其他資源。
  - 倡導您的團隊：保持與團隊的合作，讓您了解團隊的情況，以及是否有外部因素正影響著他們。當您的團隊受到外部因素影響時，請重新評估目標並適當地調整目標。找出阻礙您團隊進度的障礙。代表您的團隊來協助解決障礙並消除不必要的負擔。
  - 成為採用最佳實務的推動者：確認提供量化效益的最佳實務，並認可建立者和採用者。鼓勵進一步採用，以擴大已達成的效益。
  - 成為團隊演變的推動者：建立持續改進的文化。鼓勵人員和組織的成長和發展。提供需要隨時間逐步達成的長期奮鬥目標。調整這個願景，以在需求、業務目標以及業務環境變化時，配合您的需求、業務目標和業務環境。

OPS03-BP02 授權團隊成員在成果有風險時採取動作

工作負載擁有者已定義指引和範圍，授權團隊成員在成果有風險時做出回應。當事件超出定義的範圍時，採取向上呈報機制來取得方向。

建立此最佳實務的優勢： 透過及早測試和驗證變更，您能以最低的成本來解決問題，並限制對客戶的影響。在部署前進行測試，可將引入的錯誤數量降到最低。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 授權團隊成員在成果有風險時採取動作：為您的團隊成員提供許可、工具和機會，以練習有效回應所需的技能。
  - 讓您的團隊成員有機會練習回應所需的技能：提供替代的安全環境，在其中可安全地測試和培訓流程及程序。執行演練日，讓團隊成員可以在模擬的安全環境中獲得回應實際事件的體驗。
  - 定義並認可團隊成員採取動作的權限：透過指派許可和對其支援的工作負載和元件的存取權，明確地定義團隊成員採取動作的權限。認可他們有權在成果有風險時採取動作。

OPS03-BP03 鼓勵向上呈報

如果團隊成員認為成果有風險，則其機制可協助將疑慮向上呈報至決策制定者和利害關係人，而且我們鼓勵這麼做。應該儘早且經常向上呈報，以便識別風險，並防止風險引發事件。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 鼓勵儘早且經常向上呈報：組織認可儘早且經常呈報是最佳實務。組織認可並接受，向上呈報可能經證明是毫無根據的，然而有機會防止事件的發生，則好過於不向上呈報而錯過該機會。
  - 制定向上呈報的機制：制定定義進行向上呈報的時機與方式的記錄程序。記錄擁有越來越多採取動作或核准動作的權限的人員及其聯絡資訊。向上呈報應持續進行，直到團隊成員確信已將風險交給能夠解決問題的人員，或已聯絡承擔工作負載營運風險和責任的人員。該人員最終負責與工作負載相關的所有決策。向上呈報的內容應包括風險的本質、工作負載的關鍵性、受影響者、影響為何，以及緊急性 (也就是預期影響的時間為何)。
  - 保護向上呈報的員工：如果團隊成員圍繞無回應決策制定者或利害關係人向上呈報，則制定保護團隊成員免受報復的政策。制定機制以識別是否發生此情況，並適當地做出回應。

OPS03-BP04 溝通需及時、清楚且可行

存在的機制可用來及時通知團隊成員已知的風險和計劃的事件。提供必要的內容、詳細資訊和時間 (如果可能) 來支援判斷是否需要採取動作、需要什麼動作，並及時採取動作。例如，提供軟體漏洞的通知，以便加快修補的速度，或提供計劃的銷售促銷活動的通知，如此就能實作變更凍結，避免服務中斷的風險。

計劃的事件可以記錄在變更行事曆或維護排程中，讓團隊成員可以確定哪些活動待處理。

在 AWS 上， AWS Systems Manager 變更行事曆 可用來記錄這些詳細資訊。它支援以程式設計方式檢查行事曆狀態，以判斷行事曆在特定時間點是否有活動。可根據特定已核准時段來規劃 營運活動，該特定時段是針對潛在破壞性活動而預留。AWS Systems Manager Maintenance Windows 可讓您針對執行個體和其他 支援的資源 來排定活動，以自動化活動並讓這些活動可供探索。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 溝通需及時、清楚且可行：已設立機制，以清楚且可行的方式提供風險或計劃事件的通知，並提供足夠的通知，以便適當的回應。

- 記錄變更行事曆上計劃的活動，並提供通知：提供可存取的資訊來源，您可以在其中發現計劃的事件。提供來自相同系統之計劃事件的通知。
- 追蹤可能影響工作負載的事件和活動：監控漏洞通知和修補程式資訊，了解外部漏洞以及與工作負載元件相關的潛在風險。提供通知給團隊成員，以便讓他們可以採取動作。

資源

相關文件：

- AWS Systems Manager 變更行事曆
- AWS Systems Manager 維護時段

OPS03-BP05 鼓勵進行試驗

試驗可加速學習，讓團隊成員保持興趣和參與度。不理想的結果是成功的試驗，因其已識別出不會助力成功的路徑。團隊成員不會因取得不理想結果的成功試驗而受懲罰。必需要經由試驗才能實現創新，並讓想法轉化為成果。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 鼓勵進行試驗：鼓勵試驗以支援學習和創新。
  - 試驗各種技術：鼓勵對現在或未來可能具有適用性的技術進行試驗，以實現業務成果。這些知識可能會幫助未來的創新。
  - 帶著目標進行試驗：鼓勵對團隊成員要達成的特定目標進行試驗，或對近期可能具有適用性的技術進行試驗。這些知識可能會幫助您的創新。
  - 提供實驗的結構化時間：指定團隊成員可以免於其正常責任的特定時間，讓他們可以專注於自己的試驗。
  - 提供資源以支援試驗：為進行試驗所需的資源提供資金 (例如，軟體或雲端資源)。
  - 認可成功：識別由試驗所產生的價值。了解含有不理想成果的試驗是成功的，並且這種試驗識別了不會助力成功的路徑。團隊成員不會因試驗而來的不理想成果受到懲罰。

OPS03-BP06 團隊成員得以並受到鼓勵來維持和發展自己的技能集

團隊必須發展自己的技能集，以採用新技術，並支援需求和責任的變更，以支援您的工作負載。新技術的技能成長通常是團隊成員滿意度的來源，並可支援創新。支援團隊成員追求和維持產業認證，以驗證

和認可他們不斷成長的技能。交叉培訓以促進知識轉移，並在失去熟練的、經驗豐富且具備機構知識的成員時，降低重大影響的風險。提供學習專用的結構化時間。

AWS 提供了許多資源，包括 AWS 入門資源中心， AWS 部落格， AWS 線上技術會談， AWS 活動和網路研討會以及 AWS Well-Architected 實驗室，而這些資源均提供了可教育您的團隊的說明、範例和演練。

AWS 也分享了我們透過 在 Amazon Builders' Library 中 操作 AWS 所學到的最佳實務和模式，以及透過 不同途徑獲得的各種教材，例如 AWS 部落格 和 官方 AWS 播客。

您應該利用 AWS 提供的教育資源，例如 Well-Architected 實驗室、 AWS Support (AWS 知識中心， AWS 開發論壇和 AWS Support中心) 和 AWS 文件 資源來教育您的團隊。透過 AWS Support中心聯絡 AWS Support，以獲取 AWS 相關問題的幫助。

AWS 培訓 和認證 透過 AWS 基礎原理自主進度數位課程提供一些免費培訓。您還可以報名參加講師指導下的培訓，以進一步協助開發團隊的 AWS 技能。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 團隊成員得以並受到鼓勵來維持和發展自己的技能集：若要採用新技術、支援創新、需求和責任的變更以支援工作負載，必需進行持續的教育。
  - 為教育提供資源：提供專門的結構化時間、培訓教材和實驗室資源存取權，並支援參與會議和專業組織，這些會議和組織可為教育工作者和同儕提供學習的機會。為資淺團隊成員提供接近資深團隊成員的機會，讓資深團隊成員成為導師，或允許資深團隊成員伴隨資淺團隊成員工作，並展示他們的方法和技能。鼓勵學習與工作不直接相關的內容，以便取得更廣泛的視野。
  - 團隊教育和跨團隊參與：針對團隊成員持續的教育需求進行規劃。為團隊成員提供機會 (暫時或永久地) 加入其他團隊，以分享讓整個組織受益的技能和最佳實務
  - 支援產業認證的追求和維持：支援團隊成員取得與維持可驗證所學知識並認可其成就的產業認證。

資源

相關文件：

- AWS 入門資源中心
- AWS 部落格
- AWS 雲端 合規
- AWS 開發論壇

- AWS 文件
- AWS 線上技術會談
- AWS 活動和網路研討會
- AWS 知識中心
- AWS Support
- AWS 培訓 和認證
- AWS Well-Architected 實驗室，
- 在 Amazon Builders' Library 中
- 官方 AWS 播客。

OPS03-BP07 適當地為團隊提供資源

維持團隊成員能力，並提供工具和資源，以支援您的工作負載需求。為團隊成員指派過多的任務會增加因人為錯誤所造成的事件風險。對工具和資源的投資 (例如，為經常執行的活動提供自動化) 可以提高團隊的有效性，讓他們能夠支援其他的活動。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 適當地為團隊提供資源：確保您了解團隊的成功，以及促成他們成功或不成功的因素。以適當的資源來支援團隊。
  - 了解團隊效能：衡量團隊營運成果的達成情況與資產的開發。追蹤一段時間內輸出和錯誤率的變更。與團隊合作，了解影響他們的工作相關挑戰 (例如，責任增加、技術變更、人員損失或客戶支援增加)。
  - 了解對團隊效能的影響：保持與團隊的合作，讓您了解團隊的情況，以及是否有外部因素正影響著他們。當您的團隊受到外部因素影響時，請重新評估目標並適當地調整目標。找出阻礙您團隊進度的障礙。代表您的團隊來協助解決障礙並消除不必要的負擔。
  - 提供團隊取得成功所需的資源：定期檢閱資源是否仍然適當、或是否需要額外資源，並做出適當的調整以支援團隊。

OPS03-BP08 鼓勵並尋求來自團隊內部和跨團隊的多樣化建議

利用跨組織的多樣性，尋求多種獨特的觀點。使用此觀點來增加創新、挑戰假設，並降低確認偏差的風險。在團隊中增加包容性、多樣性和可及性，以獲得有益的觀點。

組織文化對團隊成員工作滿意度和留任率有直接影響。讓團隊成員參與其中並習得能力，以便讓業務得以成功。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 尋求多樣化的意見和觀點：鼓勵每個人做出貢獻。為代表性不足的群體發聲。在會議中輪換角色和責任。
  - 擴展角色和責任：為團隊成員提供機會，以擔任他們可能不會擔任的角色。他們會透過角色，以及與他們可能不會與之互動的新團隊成員互動，而獲得經驗和觀點。他們會將自己的經驗和觀點帶到新的角色，並帶給和他們互動的團隊成員。隨著觀點增加，可能會出現額外的商機，或者可能識別出新的改進機會。讓團隊內的成員輪流處理其他人通常執行的常見任務，以了解執行這些任務的需求和影響。
  - 提供安全且友善的環境：制定政策與控制措施，保護組織內團隊成員在精神和身體上的安全。團隊成員應該能夠在不擔心報復行為的情況下進行互動。當團隊成員感到安全且受歡迎時，他們才更有可能參與進來並具備生產力。您的組織越多樣化，您就越能了解所支援的人員，包括您的客戶。當您的團隊成員感到安心、可以自在的暢所欲言，而且有信心他們的聲音不會被淹沒，他們才更有可能分享寶貴的洞見 (例如，行銷機會、可及性的需求、尚未有服務的市場區段、環境中未確認的風險)。
  - 讓團隊成員能夠充分參與：提供員工充分參與所有與工作相關的活動所需的資源。面對日常挑戰的團隊成員已發展出解決挑戰的技能。這些以獨特方式發展的技能可為組織提供顯著的效益。為團隊成員提供必要住宿支援，將可提高從他們的貢獻中所獲得的效益。

準備

問題

- OPS 4 您如何設計工作負載以便了解其狀況？
- OPS 5 您如何減少缺陷、幫助輕鬆修復，以及改善生產流程？
- OPS 6 您如何緩解部署風險？
- OPS 7 您如何知道自己準備好支援工作負載？

OPS 4 您如何設計工作負載以便了解其狀況？

設計工作負載，以便它為您提供了解其內部狀態所需的跨全部元件 (例如指標、日誌和追蹤) 的資訊。這讓您能在適當時機提供有效回應。

最佳實務

- OPS04-BP01 實作應用程式遙測
- OPS04-BP02 實作和設定工作負載遙測
- OPS04-BP03 實作使用者活動遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作交易可追溯性

OPS04-BP01 實作應用程式遙測

應用程式遙是工作負載可觀察性的基礎。您的應用程式應該發出遙測，讓您洞悉應用程式的狀態和業務成果的成就。從疑難排解到衡量新功能的影響，應用程式遙測可告知您建置、操作和發展工作負載的方式。

應用程式遙測由指標和日誌組成。指標是診斷資訊，就如您的脈搏或體溫。指標是共同用來描述應用程式的狀態。隨時間收集指標可以用來開發基準和偵測異常。日誌是應用程式傳送的訊息，其中關於內部狀態或發生的事件。錯誤碼、交易識別碼和使用者動作都是所記錄事件的範例。

預期成果：

- 您的應用程式會發出指標和日誌，讓您洞悉其運作狀態和業務成果的成就。
- 所有應用程式的指標和日誌會集中儲存在工作負載中。

常用的反模式：

- 您的應用程式不會發出遙測。當發生錯誤時，您必須仰賴客戶來告知您。
- 客戶已回報，您的應用程式沒有回應。不親自使用應用程式來了解目前的使用者體驗，您便沒有遙測功能，且無法確認問題存在或描述問題特性。

建立此最佳實務的優勢：

- 您可以了解應用程式的運作狀態、使用者體驗，以及業務成果的成就。
- 您可以快速反映應用程式運作狀態中的變更。
- 您可以開發應用程式運作狀態趨勢。
- 您可以做出明智的決策，改善您的應用程式。
- 您可以更快地偵測並解決應用程式問題。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

實作應用程式遙測包含三個步驟：識別儲存遙測的位置、識別描述應用程式狀態的遙測，以及檢測要發出遙測的應用程式。

舉例來說，商務公司具有以微型服務為基礎的架構。作為架構設計程序的一部分，他們識別了應用程序遙測，而其會協助他們了解每個微型服務的狀態。例如，使用者購物車服務已發出遙測，其中包括關於新增到購物車、放棄購物車，以及將商品新增到購物車所需時間長度等事件。所有微型服務都會記錄錯誤、警告和交易資訊。遙測會傳送至 Amazon CloudWatch 進行儲存和分析。

實作步驟

第一步是針對工作負載中的應用程式識別儲存遙測的中心位置。如果您沒有現有的平台， Amazon CloudWatch 會提供遙測收集、儀表板、分析和事件產生功能。

若要識別您需要的遙測，請參考下列問題：

- 我的應用程式的運作狀態是否良好？
- 我的應用程式是否實現了業務成果？

  您的應用程式應該發出日誌和指標，並共同回答這些問題。如果您無法使用現有的應用程式遙測來回答這些問題，請與商務和工程利害關係人合作，以建立可以回答這些問題的遙測清單。當識別並開發新的應用程式遙測時，您可以向 AWS 帳戶 團隊要求專家技術建議。

  一旦識別了其他應用程式遙測，請與您的工程利害關係人合作，以檢測您的應用程式。 適用於 Open Telemetry 的 AWS Distro 提供 API、程式庫和代理程式，收集應用程式遙測。 此範例示範如何使用自訂指標檢測 JavaScript 應用程式。 。

  客戶若想要了解 AWS 提供的可觀察性服務，可以透過自己的 一個觀察工作坊 來運作或向其 AWS 帳戶 團隊要求支援來指引他們。此工作坊會透過 AWS 的可觀察性解決方案指引您，並提供如何使用它們的實際操作範例。

  如需更深入探討應用程式遙測，請閱讀 Amazon Builder's Library 中的 偵測分散式系統，以了解運作狀態 一文。它會說明 Amazon 如何檢測應用程式，以及如何指引您開發自己的檢測指導方針。

實作計劃的工作量： 中

資源

相關的最佳實務：

the section called “OPS04-BP02 實作和設定工作負載遙測” – 應用程式遙測是工作負載遙測的元件。若要了解整體工作負載的運作狀態，您必須了解構成工作負載之個別應用程式的運作狀態。

the section called “OPS04-BP03 實作使用者活動遙測” – 使用者活動遙測通常是應用程式遙測的子集。新增至購物車事件、點擊流或已完成交易等使用者活動，可讓您洞悉使用者體驗。

the section called “OPS04-BP04 實作相依性遙測” – 相依性與應用程式遙測相關，而且可能會配備至您的應用程式。如果您的應用程式依賴 DNS 或資料庫等外部相依性，您的應用程式可以發出有關連線能力、逾時和其他事件的指標和日誌。

the section called “OPS04-BP05 實作交易可追溯性” – 跨工作負載追蹤交易需要每個應用程式發出其如何處理共用事件的相關資訊。個別應用程式處理這些事件的方式是透過其應用程式遙測發出的。

the section called “OPS08-BP02 定義工作負載指標” – 工作負載遙測是工作負載的重要運作狀態指標。重要應用程式指標是工作負載指標的一部分。

相關文件：

- AWS Builders' Library – 偵測分散式系統，以瞭解運作狀態
- 適用於 OpenTelemetry 的 AWS Distro
- AWS Well-Architected 卓越營運支柱白皮書 – 設計遙測
- 使用篩選條件從日誌事件建立指標
- 使用 Amazon CloudWatch 實作記錄和監控
- 使用適用於 OpenTelemetry 的 AWS Distro 監控應用程式運作狀態和效能
- 新增功能 – 如何使用 Amazon CloudWatch Agent 更好地監控自訂應用程式指標
- AWS 的可觀測性
- 案例 – 將指標發佈至 CloudWatch
- 開始建置 – 如何有效率地監控您的應用程式
- 使用 CloudWatch 搭配 AWS SDK

相關影片：

- AWS re:Invent 2021 - 可觀察性開放原始碼方式
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體收集指標和日誌

- 如何輕鬆地為您的 AWS 工作負載設定應用程式監控 - AWS 線上技術會談
- 精通無伺服器應用程式的可觀察性 - AWS 線上技術會談
- 搭配 AWS 的開放原始碼可觀察性 - AWS 虛擬研討會

相關範例：

- AWS 記錄和監控範例資源
- AWS 解決方案：Amazon CloudWatch 監控架構
- AWS 解決方案：集中式記錄
- 一個觀察工作坊

OPS04-BP02 實作和設定工作負載遙測

設計和設定您的工作負載，以發出有關其內部狀態和當前狀況的資訊，例如 API 呼叫量、HTTP 狀態碼和擴展事件。使用此資訊來協助確定何時需要回應。

使用 Amazon CloudWatch 這類服務彙總工作負載元件的日誌和指標 (例如， AWS CloudTrail 的 API 日誌、 AWS Lambda 指標， Amazon VPC 流程日誌和 其他服務) 建立持續整合/持續部署 (CI/CD) 管道。

常用的反模式：

- 您的客戶在抱怨效能不佳。您的應用程式最近無變更，因此您懷疑工作負載元件發生問題。您沒有可分析的遙測資料，以判斷哪個或哪些元件造成效能不佳。
- 您的應用程式無法連線。您缺乏遙測資料來判斷它是否為網路問題。

建立此最佳實務的優勢： 了解工作負載內部發生的情況，讓您可以視需要做出回應。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 實作日誌和指標遙測：檢測您的工作負載，以發出有關其內部狀態、狀況和業務成果實現情況的資訊。使用此資訊來確定何時需要回應。
  - 使用 Amazon CloudWatch 更好地了解您的 VM - AWS 線上技術會談
  - Amazon CloudWatch 的運作方式
  - 什麼是 Amazon CloudWatch？

- 使用 Amazon CloudWatch 指標
- 什麼是 Amazon CloudWatch Logs？
  - 實作和設定工作負載遙測：設計和設定您的工作負載，以發出有關其內部狀態和當前狀況的資訊 (例如 API 呼叫量、HTTP 狀態碼和擴展事件)。
    - Amazon CloudWatch 指標和維度參考
    - AWS CloudTrail
    - 什麼是 AWS CloudTrail？
    - VPC Flow Logs

資源

相關文件：

- AWS CloudTrail
- Amazon CloudWatch 文件
- Amazon CloudWatch 指標和維度參考
- Amazon CloudWatch 的運作方式
- 使用 Amazon CloudWatch 指標
- VPC Flow Logs
- 什麼是 AWS CloudTrail？
- 什麼是 Amazon CloudWatch Logs？
- 什麼是 Amazon CloudWatch？

相關影片：

- AWS 上的應用程式效能管理
- 使用 Amazon CloudWatch 更好地了解您的 VM
- 使用 Amazon CloudWatch 更好地了解您的 VM - AWS 線上技術會談

OPS04-BP03 實作使用者活動遙測

檢測您的應用程式程式碼，以發出有關使用者活動的資訊 (例如，點按流或已開始、已放棄和已完成的交易)。使用此資訊來了解應用程式如何被使用、使用模式以及確定何時需要回應。

常用的反模式：

- 您的開發人員已部署新功能，而不需使用者遙測功能，而且使用率也已提升。您無法判斷提高的使用率是來自新功能的使用，還是新程式碼產生的問題。
- 您的開發人員已部署新功能，而不需使用者遙測功能。若不主動詢問您的客戶，您無法判斷客戶是否正在使用它。

建立此最佳實務的優勢： 了解客戶如何使用您的應用程式來識別使用模式、意外行為，並讓您能夠在必要時做出回應。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 實作使用者活動遙測：設計您的應用程式程式碼，以發出有關使用者活動的資訊 (例如，點按流或已開始、已放棄和已完成的交易)。使用此資訊來了解應用程式如何被使用、使用模式以及確定何時需要回應。

OPS04-BP04 實作相依性遙測

設計和設定您的工作負載，以發出有關其相依資源之狀態 (例如，可達性或回應時間) 的資訊。外部相依性的範例可包含外部資料庫、DNS 和網路連線。使用此資訊來確定何時需要回應。

常用的反模式：

- 若未手動執行檢查來查看您的 DNS 供應商是否正常運作，您將無法判斷應用程式無法存取的原因是否是 DNS 問題。
- 您的購物車應用程式無法完成交易。如果沒有聯絡信用卡處理供應商進行驗證，您就無法判斷是否是供應商的問題。

建立此最佳實務的優勢： 了解依存項目的運作狀態，讓您可以視需要做出回應。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 實作相依性遙測：設計和設定您的工作負載，以發出有關其所依賴系統的狀態和狀況的資訊。此處提供的一些範例包括：外部資料庫、DNS、網路連線和外部信用卡處理服務。

- Amazon CloudWatch Agent 與 AWS Systems Manager 整合 - 適用於 Linux 和 Windows 的統一指標和日誌收集
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌

資源

相關文件：

- Amazon CloudWatch Agent 與 AWS Systems Manager 整合 - 適用於 Linux 和 Windows 的統一指標和日誌收集
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌

相關範例：

- Well-Architected 實驗室 – 相依性監控

OPS04-BP05 實作交易可追溯性

實作您的應用程式程式碼並設定您的工作負載元件，以發出有關整個工作負載交易流的資訊。使用此資訊來確定何時需要回應，並幫助確定問題的根本原因。

在 AWS 上，您可以使用 AWS X-Ray等分散式追蹤服務，在交易通過工作負載時收集和記錄追蹤、產生地圖以查看交易如何在不同的工作負載和服務之間流動、深入了解元件之間的關係，以及即時確定和分析問題。

常用的反模式：

- 您已實作跨多個帳戶的無伺服器微型服務架構。您的客戶遇到了間歇性的效能問題。因為缺少讓您能找出應用程式中存在效能問題及造成問題原因的軌跡，所以您無法找出哪個函數或元件應該負責。
- 您正嘗試判斷工作負載中效能瓶頸的位置，以便在開發工作中解決這些瓶頸。您無法查看應用程式元件和與其互動的服務之間的關係，以判斷瓶頸的位置，原因是您缺少讓自己可深入檢視影響應用程式效能的特定服務和路徑之軌跡。

建立此最佳實務的優勢： 了解工作負載中的交易流程，讓您可以了解工作負載交易的預期行為，以及工作負載中預期行為的變化，讓您能夠在必要時做出回應。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 實作交易可追溯性：設計您的應用程式和工作負載，以發出有關跨系統元件的交易流的資訊，例如交易階段、作用中元件和完成活動的時間。使用此資訊確定正在進行的活動、已經完成的活動以及已完成活動的結果。這可以幫助您確定何時需要回應。例如，某個元件內的交易回應時間比預期的長，可能表示該元件存在問題。
  - AWS X-Ray
  - 什麼是 AWS X-Ray？

資源

相關文件：

- AWS X-Ray
- 什麼是 AWS X-Ray？

OPS 5 您如何減少缺陷、幫助輕鬆修復，以及改善生產流程？

採用改善改變生產流程的方法，藉此重構、快速提供品質意見回饋及修復錯誤。這會加快有助益的改變發揮作用的速度、限制部署問題，並快速識別和修復部署活動造成的問題。

最佳實務

- OPS05-BP01 使用版本控制
- OPS05-BP02 測試並驗證變更
- OPS05-BP03 使用組態管理系統
- OPS05-BP04 使用建置和部署管理系統
- OPS05-BP05 執行修補程式管理
- OPS05-BP06 共用設計標準
- OPS05-BP07 實作用於提高程式碼品質的實務
- OPS05-BP08 使用多個環境
- OPS05-BP09 進行頻繁、細微和可逆的變更
- OPS05-BP10 完全自動化整合和部署

OPS05-BP01 使用版本控制

使用版本控制來追蹤變更和發佈。

許多 AWS 服務都提供版本控制功能。使用修訂版或原始程式碼控制系統 (例如 AWS CodeCommit)，管理程式碼和其他成品，例如基礎架構之版本控制的 AWS CloudFormation 範本。

常用的反模式：

- 您已在工作站上開發和存放程式碼。您在遺失程式碼的工作站上發生無法復原的儲存失敗。
- 在您的變更覆寫現有的程式碼之後，您需重新啟動應用程式，且它將無法再運作。您無法還原為變更版本。
- 您對另一人需要編輯的報告檔案加上了寫入鎖定。他們會與您聯絡，要求您停止處理該檔案，以便他們完成任務。
- 您的研究團隊一直在進行詳細的分析，以塑造您未來的工作。某人意外地將他們的購物清單儲存在最終報告中。您無法還原變更，且必須重新建立報告。

建立此最佳實務的優勢： 透過使用版本控制功能，您可以輕鬆回復為已知的良好狀態、舊版本，並限制資產遺失的風險。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 使用版本控制：在版本控制的儲存庫中維護資產。此舉可實現變更追蹤、新版本部署、對現有版本的變更偵測以及還原到先前的版本 (例如，在發生故障時復原到已知的良好狀態)。將組態管理系統的版本控制功能整合到您的程序中。
  - AWS CodeCommit 簡介
  - 什麼是 AWS CodeCommit？

資源

相關文件：

- 什麼是 AWS CodeCommit？

相關影片：

- AWS CodeCommit 簡介

OPS05-BP02 測試並驗證變更

測試和驗證變更以幫助限制和偵測錯誤。自動化測試以減少由手動程序引起的錯誤，並減少測試工作量。

許多 AWS 服務都提供版本控制功能。使用修訂版或原始程式碼控制系統 (例如 AWS CodeCommit)，管理程式碼和其他成品，例如基礎架構之版本控制的 AWS CloudFormation 範本。

常用的反模式：

- 您將新程式碼部署到生產環境中，然後客戶開始來電，因為您的應用程式不再運作。
- 您可以套用新的安全群組，以增強週邊安全。它在運作時隨附意外後果；您的使用者無法存取您的應用程式。
- 您可以修改新函數所叫用的方法。另一個函數也依賴該方法，且不再運作。問題無法偵測到並進入生產環境。另一函數有一段時間不會被叫用，最後在生產環境中失敗，而無原因的任何關聯。

建立此最佳實務的優勢： 透過及早測試和驗證變更，您能以最低的成本來解決問題，並限制對客戶的影響。在部署前進行測試，可將引入的錯誤數量降到最低。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 測試並驗證變更：應該在生命週期的所有階段 (例如，開發、測試和生產) 測試變更並驗證結果。使用測試結果來確認新功能，並減輕失敗部署的風險和影響。自動執行測試和驗證，以確保檢閱的一致性，減少由手動程序引起的錯誤，並減少工作量。
  - 什麼是 AWS CodeBuild？
  - 對 AWS CodeBuild 本機建置支援

資源

相關文件：

- AWS 開發人員工具
- 對 AWS CodeBuild 本機建置支援

- 什麼是 AWS CodeBuild？

OPS05-BP03 使用組態管理系統

使用組態管理系統進行和追蹤組態變更。這些系統可減少由手動程序引起的錯誤，並減少部署變更的工作量。

靜態組態管理在初始化資源時設定值，這些值預期在資源的整個生命週期內保持一致。部分範例包括在執行個體上設定 Web 或應用程式伺服器的組態，或定義 AWS 服務 (在 AWS Management Console 內) 的組態，或透過下列項目定義該組態： AWS CLI。

動態組態管理在初始化時設定值，這些值可以預期在資源的整個生命週期內保持一致。例如，您可以設定功能切換，透過組態變更啟用程式碼中的功能，或者在事故期間變更日誌詳細資訊等級以擷取更多資料，然後在事故後改回來，這會消除目前不需要的日誌及其相關費用。

如果您在執行個體、容器、無伺服器功能或裝置上執行的應用程式中具有動態組態，則可以使用 AWS AppConfig 跨您的環境管理和部署它們。

在 AWS 上，您可以使用 AWS Config 跨帳戶和區域持續監控 AWS 資源組態。它可讓您追蹤其組態歷史紀錄、了解組態變更如何影響其他資源、以及針對預期或所需的組態稽核它們，方法是使用 AWS Config 規則 和 AWS Config 合規套件。

在 AWS 上，您可以使用 AWS 開發人員工具 等服務 (例如，AWS CodeCommit、 AWS CodeBuild、AWS CodePipeline、 AWS CodeDeploy和 AWS CodeStar) 建立持續整合/持續部署 (CI/CD) 管道。

製作變更行事曆，並追蹤可能受到變更實作影響的計劃的重要業務或營運活動或事件。調整活動以管理這些計畫的風險。 AWS Systems Manager 變更行事曆 AWS Systems Manager 變更行事曆提供一種機制，可將時間區塊記錄為變更開啟或變更關閉， 以及紀錄原因， 並與其他 AWS 帳戶 分享該資訊。AWS Systems Manager Automation 指令碼可設定為遵照變更行事曆狀態。

AWS Systems Manager 維護時段 可用來在指定的時間排程 AWS SSM Run Command 或 Automation 指令碼、AWS Lambda 叫用或 AWS Step Functions 活動的效能。在變更行事曆中標記這些活動，以便將它們納入您的評估中。

常用的反模式：

- 您跨機群手動更新 Web 伺服器組態，而且由於更新錯誤，許多伺服器無法回應。
- 您在數小時內手動更新應用程式伺服器機群。變更期間的組態不一致會導致未預期的行為。
- 某人已更新您的安全群組，無法再存取您的 Web 伺服器。若不知道發生了什麼變更，您需花大量時間來調查問題，從而延長復原的時間。

建立此最佳實務的優勢： 採用組態管理系統可減少進行和追蹤變更的工作量，以及手動程序造成的錯誤頻率。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 使用組態管理系統：使用組態管理系統來追蹤和實作變更，以減少由手動程序引起的錯誤，並減少工作量。
  - 基礎設施組態管理
  - AWS Config
  - 什麼是 AWS Config？
  - AWS CloudFormation 簡介
  - 什麼是 AWS CloudFormation？
  - AWS OpsWorks
  - 什麼是 AWS OpsWorks？
  - AWS Elastic Beanstalk 簡介
  - 什麼是 AWS Elastic Beanstalk？

資源

相關文件：

- AWS AppConfig
- AWS 開發人員工具
- AWS OpsWorks
- AWS Systems Manager 變更行事曆
- AWS Systems Manager 維護時段
- 基礎設施組態管理
- 什麼是 AWS CloudFormation？
- 什麼是 AWS Config？
- 什麼是 AWS Elastic Beanstalk？
- 什麼是 AWS OpsWorks？

相關影片：

- AWS CloudFormation 簡介
- AWS Elastic Beanstalk 簡介

OPS05-BP04 使用建置和部署管理系統

使用建置和部署管理系統。這些系統可減少由手動程序引起的錯誤，並減少部署變更的工作量。

在 AWS 中，您可以使用 AWS 開發人員工具 等服務 (例如，AWS CodeCommit、 AWS CodeBuild、AWS CodePipeline、 AWS CodeDeploy和 AWS CodeStar) 建立持續整合/持續部署 (CI/CD) 管道。

常用的反模式：

- 在開發系統中編譯程式碼之後，您將可執行檔複製到生產系統中，然後其無法啟動。本機日誌檔案指出其因缺少相依性而失敗。
- 您在開發環境中使用新功能成功建置應用程式，並將程式碼提供給品質保證 (QA)。它的 QA 失敗，原因是它缺少靜態資產。
- 在花費大量精力之後的星期五，您已在開發環境中成功手動建置應用程式，包括您新編碼的功能。在星期一，您無法重複讓您成功建置應用程式的步驟。
- 您執行為新版本建立的測試。然後，您會在下週設定測試環境，並執行所有現有的整合測試，接著執行效能測試。新的程式碼具有無法接受的效能影響，必須重新開發，然後重新測試。

建立此最佳實務的優勢： 透過提供用於管理建置和部署活動的機制，您可以減少執行重複性任務的工作量，讓團隊成員專注於高價值的創意任務，並限制手動程序引入錯誤。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用建置和部署管理系統：使用建置和部署管理系統來追蹤和實作變更，以減少由手動流程引起的錯誤，並減少工作量。從程式碼簽入到建置、測試、部署和驗證，完全自動化整合和部署管道。此舉可減少前置時間，增加變更頻率，並降低工作量。
  - 什麼是 AWS CodeBuild？
  - 軟體開發持續整合最佳實務
  - Slalom：AWS 上適用於無伺服器應用程式的 CI/CD
  - AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署

- 什麼是 AWS CodeDeploy？

資源

相關文件：

- AWS 開發人員工具
- 什麼是 AWS CodeBuild？
- 什麼是 AWS CodeDeploy？

相關影片：

- 軟體開發持續整合最佳實務
- AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
- Slalom：AWS 上適用於無伺服器應用程式的 CI/CD

OPS05-BP05 執行修補程式管理

執行修補程式管理以獲取功能，解決問題並保持遵循管控。自動化修補程式管理，以減少由手動程序引起的錯誤，並減少修補工作量。

修補程式和漏洞管理屬於您利益和風險管理活動的一部分。最好擁有不可變的基礎設施，並在已驗證的已知良好狀態下部署工作負載。如果這不可行，可選擇剩餘的方法，即實施修補程式。

更新機器映像、容器映像或 Lambda 或 Lambda 自訂執行階段和其他程式庫 以移除漏洞，屬於修補程式管理的一部分。您 應該 使用 EC2 Image Builder，管理適用於 Linux 或 Windows Server 映像的 Amazon Machine Image (AMIs) 更新。您可以使用 Amazon Elastic Container Registry 搭配現有管道來 管理 Amazon ECS 映像 和 管理 Amazon EKS 映像。AWS Lambda 包含 版本 管理功能。

若未首先在安全環境中進行測試，就不應在生產系統上執行修補程式。只有在修補程式能夠支援營運或業務成果時，才應套用修補程式。在 AWS 上，您可以使用 AWS Systems Manager Patch Manager 自動化受管系統的修補程序，以及 使用 AWS Systems Manager 維護時段。

常用的反模式：

- 您必須在兩小時內套用所有新的安全性修補程式，結果導致應用程式與修補程式不相容而致使多次停機。
- 未修補的程式庫會導致意外後果，因為未知方利用其中的漏洞來存取您的工作負載。

- 您自動修補開發人員環境，而未通知開發人員。您收到來自開發人員的多個投訴，開發人員表示其環境如預期停止運作。
- 您尚未在持久性執行個體上修補商用現成軟體。當您有軟體問題並聯絡廠商時，他們會通知您該版本不受支援，您必須修補至特定程度才能獲得任何協助。
- 您使用的加密軟體的最新修補程式可大幅改善效能。未修補的系統因未修補仍存在效能問題。

建立此最佳實務的優勢： 透過建立修補程式管理程序 (包括修補的條件和跨環境分佈的方法)，您將能夠實現其優勢並控制其影響。這樣一來，便能採用所需的功能和特性、消除問題，並持續遵循管控要求。實作修補程式管理系統和自動化，以減少部署修補程式的工作量，並限制手動程序引起的錯誤。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 修補程式管理：修補系統以補救問題，獲得所需的功能或特性，並保持符合管控政策和供應商支援要求。在不可變系統中，部署適當的修補程式集以實現所需的結果。自動化修補程式管理機制，以縮短修補時間，減少由手動程序引起的錯誤並降低修補工作量。
  - AWS Systems Manager Patch Manager

資源

相關文件：

- AWS 開發人員工具
- AWS Systems Manager Patch Manager

相關影片：

- AWS 上適用於無伺服器應用程式的 CI/CD
- 設計時考量 OPS

相關範例：

- Well-Architected 實驗室 – 庫存和修補程式管理

OPS05-BP06 共用設計標準

在團隊之間共用最佳實務，以提高認識並最大化開發工作的效益。

在 AWS 上，可使用程式碼方法來定義和管理應用程式、運算、基礎設施和營運。如此一來，您可輕鬆進行發佈、分享及採用。

許多 AWS 服務和資源旨在跨帳戶進行分享，從而讓您可跨團隊分享已建立的資產和經驗。例如，您可將 CodeCommit 儲存庫、 Lambda 函數、 Amazon S3 儲存貯體和 AMI 分享給特定帳戶。

發佈新資源或更新時，請使用 Amazon SNS 提供 發佈通知。訂閱者可以使用 Lambda 獲得新版本。

如果您的組織中強制執行共用標準，則必須存在用於請求標準新增、變更及例外狀況的機制，以支援團隊的活動。如果沒有此選項，標準就會限制創新。

常用的反模式：

- 您已建立自己的使用者身份驗證機制，且組織中的每一個其他開發團隊也是一樣。您的使用者必須針對要存取的系統的每一部分，維護一組單獨的登入資料。
- 您已建立自己的使用者身份驗證機制，且組織中的每一個其他開發團隊也是一樣。必須滿足的新合規要求已加諸於您的組織。每個開發團隊現在都必須投資資源來實作新要求。
- 您已建立自己的畫面版面配置，且組織中的每一個其他開發團隊也是一樣。您的使用者抱怨很難導覽不一致的介面。

建立此最佳實務的優勢： 使用共用標準來支援最佳實務之採用，並在標準滿足多個應用程式或組織的要求時，將開發工作的效益發揮到最大。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 共用設計標準：在團隊之間分享現有的最佳實務、設計標準、檢查清單、操作程序以及指導和管控要求，以降低複雜性並最大化開發工作的收益。確保存在用於請求對設計標准進行變更、新增和例外的程序，以支援持續的改進和創新。確保團隊了解發佈的內容，以便他們可以利用內容，限制重新作業以及工作上徒勞無功。
  - 委託存取您的 AWS 環境
  - 共用 AWS CodeCommit 儲存庫
  - 輕鬆授權 AWS Lambda 函數
  - 與特定 AWS 帳戶共用 AMI
  - 使用 AWS CloudFormation Designer URL 加速範本共用
  - 搭配 Amazon SNS 使用 AWS Lambda

資源

相關文件：

- 輕鬆授權 AWS Lambda 函數
- 共用 AWS CodeCommit 儲存庫
- 與特定 AWS 帳戶共用 AMI
- 使用 AWS CloudFormation Designer URL 加速範本共用
- 搭配 Amazon SNS 使用 AWS Lambda

相關影片：

- 委託存取您的 AWS 環境

OPS05-BP07 實作用於提高程式碼品質的實務

實作實務以提高程式碼品質並將缺陷降至最少。部分範例包括測試驅動的開發、程式碼檢閱和標準採用。

在 AWS 上，您可以整合 Amazon CodeGuru 這類服務與管道，以自動 識別潛在程式碼和安全問題，方法為使用程式分析和機器分學習。CodeGuru 提供如何實作 AWS 最佳實務來解決這些問題的建議。

常用的反模式：

- 為了能夠更快測試您的功能，您已決定不整合您的標準輸入清理程式庫。測試之後，您忘記要完成程式庫合併，就遞交程式碼。
- 對於正在處理的資料集，您的經驗不多，且不知道資料集中可能存在一連串邊緣案例。這些邊緣案例與您實作的程式碼不相容。

建立此最佳實務的優勢： 透過採用提高程式碼品質的實務，您可以協助將引入生產中的問題降至最低。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 實作用於提高程式碼品質的實務：實作實務以提高程式碼品質，將缺陷和缺陷被部署的風險降至最低。例如，測試驅動的開發、結對程式設計、程式碼審查和標准採用。

- Amazon CodeGuru

資源

相關文件：

- Amazon CodeGuru

OPS05-BP08 使用多個環境

使用多個環境進行實驗、開發和測試您的工作負載。當環境接近生產環境時使用更高的控制等級，以確保您的工作負載在部署後將按預期執行。

常用的反模式：

- 您在共享開發環境中進行開發，而另一名開發人員覆寫您的程式碼變更。
- 對共享開發環境的限制性安全控制，讓您無法試驗新服務和功能。
- 您對生產系統執行負載測試，並給使用者造成停機。
- 在生產環境中發生導致資料遺失的嚴重錯誤。在您的生產環境中，您嘗試重新建立導致資料遺失的條件，以便您能夠識別其發生情況並防止再次發生。為防止更多資料在測試期間遺失，您必須讓使用者無法使用應用程式。
- 您正在操作多租用戶服務，且無法支援客戶對專用環境的要求。
- 您可能不一定總是進行測試，但當在生產環境中時，請務必測試。
- 您認為簡單的單一環境會覆寫環境內變更的影響範圍。

建立此最佳實務的優勢： 透過部署多個環境，您可以支援多個同時開發、測試和生產環境，而不會在開發人員或使用者社群之間產生衝突。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用多個環境：為開發人員沙盒環境施加最少的控制，以推動實驗。提供多個單獨的開發環境，以使並行工作成為可能，從而提高開發敏捷性。在接近生產的環境中實作更嚴格的控制，以允許開發人員創新。使用基礎設施即程式碼和組態管理系統，以部署設定了與生產環境一致控制的環境，從而確保系統在部署時能夠按預期執行。當不使用環境時，關閉環境以避免與空閒資源相關的成本 (例如，在夜間和周末關閉開發系統)。進行負載測試時，部署與生產環境等效的環境，以獲得有效結果。

- 什麼是 AWS CloudFormation？
- 如何使用 AWS Lambda 定期停止和啟動 Amazon EC2 執行個體？

資源

相關文件：

- 如何使用 AWS Lambda 定期停止和啟動 Amazon EC2 執行個體？
- 什麼是 AWS CloudFormation？

OPS05-BP09 進行頻繁、細微和可逆的變更

頻繁、細微和可逆的變更會縮小變更的範圍和影響。這樣可以簡化故障診斷，實現更快的修復，並提供回復變更的選項。

常用的反模式：

- 您在每一季都部署應用程式的新版本。
- 您經常變更資料庫結構描述。
- 您執行手動就地更新，並覆寫現有的安裝和組態。

建立此最佳實務的優勢： 您透過經常部署小的變更，更快認識到開發工作帶來的效益。當變更很小時，可更容易識別它們是否會產生意外的後果。如果變更可逆，由於復原得到簡化，實作變更的風險會更小。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 進行頻繁、細微、可逆的變更：頻繁、細微和可逆的變更可縮小變更範圍和變更影響。這樣可以簡化故障診斷，實現更快的修復，並提供回復變更的選項。它還可以提高您為企業帶來價值的速度。

OPS05-BP10 完全自動化整合和部署

自動化工作負載的建置、部署和測試。此舉可減少由手動程序引起的錯誤，以及部署變更的工作量。

依照一致的標記策略，使用 資源標籤 和 AWS Resource Groups 來套用 中繼資料， 以識別您的資源。標記您的資源，以用於組織、成本會計、存取控制，以及將自動執行營運活動設為目標。

常用的反模式：

- 週五，您完成了為功能分支編寫新程式碼。週一，在執行您的程式碼品質測試指令碼和每個單位測試指令碼之後，您將針對下一排程版本來檢查程式碼。
- 系統會指派您編寫修正程式碼，以解決影響生產環境中大量客戶的重大問題。測試修正後，您遞交程式碼和電子郵件變更管理內容，以請求核准將其部署到生產環境中。

建立此最佳實務的優勢： 透過實作自動化建置和部署管理系統，您可以減少手動程序引起的錯誤，以及部署變更的工作量，讓您的團隊成員能夠專注於提供商業價值。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 使用建置和部署管理系統：使用建置和部署管理系統來追蹤和實作變更，以減少由手動流程引起的錯誤，並減少工作量。從程式碼簽入到建置、測試、部署和驗證，完全自動化整合和部署管道。此舉可減少前置時間，增加變更頻率，並降低工作量。
  - 什麼是 AWS CodeBuild？
  - 軟體開發持續整合最佳實務
  - Slalom：AWS 上適用於無伺服器應用程式的 CI/CD
  - AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
  - 什麼是 AWS CodeDeploy？

資源

相關文件：

- 什麼是 AWS CodeBuild？
- 什麼是 AWS CodeDeploy？

相關影片：

- 軟體開發持續整合最佳實務
- AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
- Slalom：AWS 上適用於無伺服器應用程式的 CI/CD

OPS 6 您如何緩解部署風險？

採用可快速提供品質意見回饋，並從成果不盡理想的改變中快速復原的方法。使用這些實務可緩解部署變更所帶來問題的影響。

最佳實務

- OPS06-BP01 為失敗變更進行規劃
- OPS06-BP02 測試並驗證變更
- OPS06-BP03 使用部署管理系統
- OPS06-BP04 使用有限的部署進行測試
- OPS06-BP05 使用平行環境進行部署
- OPS06-BP06 部署頻繁、細微和可逆的變更
- OPS06-BP07 完全自動化整合和部署
- OPS06-BP08 自動化測試和復原

OPS06-BP01 為失敗變更進行規劃

計劃在變更未達到理想成果時，恢復到已知的良好狀態，或者在生產環境中進行補救。透過這樣準備可加快回應速度，以縮短復原時間。

常用的反模式：

- 您執行了部署，而您的應用程式變得不穩定，但系統中似乎有作用中使用者。您必須決定是否要復原變更並影響作用中使用者，或在知道使用者無論如何都會受到影響的情況下，等待復原變更。
- 在進行路由變更後，您可以存取新的環境，但其中一個子網路變成無法連線。您必須決定是否要復原所有項目，或嘗試修正無法存取的子網路。當您做出該決定時，子網路仍無法連線。

建立此最佳實務的優勢： 具有適當的計劃可減少從不成功變更中復原的平均時間 (MTTR)，從而減少對最終使用者的影響。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 為失敗變更進行規劃：計劃在變更未達到理想成果時，恢復到已知的良好狀態 (即回復變更)，或者在生產環境中進行補救 (即向前回復變更)。當您確定無法回復的變更時，在提交之前應進行盡職調查。

OPS06-BP02 測試並驗證變更

在生命週期所有階段測試變更並驗證結果，以確認新功能，並將失敗部署的風險和影響降至最低。

在 AWS 上，您可以建立臨時平行環境，以降低試驗和測試的風險、工作量及成本。使用 AWS CloudFormation 自動化這些環境的部署，以確保臨時環境的一致實作。

常用的反模式：

- 您為應用程式部署一個很酷的新功能。但它無法運作。您不知道。
- 您更新憑證。您不小心將憑證安裝到錯誤的元件。您不知道。

建立此最佳實務的優勢： 透過在部署之後測試和驗證變更，您可以及早識別問題，藉此降低對客戶造成的影響。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 測試並驗證變更：在生命週期所有階段 (例如，開發、測試和生產) 測試變更並驗證結果，以確認新功能，並將失敗部署的風險和影響降至最低。
  - AWS Cloud9
  - 什麼是 AWS Cloud9？
  - 在交付程式碼之前，如何在本機測試和偵錯 AWS CodeDeploy

資源

相關文件：

- AWS Cloud9
- AWS 開發人員工具
- 在交付程式碼之前，如何在本機測試和偵錯 AWS CodeDeploy
- 什麼是 AWS Cloud9？

OPS06-BP03 使用部署管理系統

使用部署管理系統來追蹤和實作變更。此舉可減少由手動程序引起的錯誤，以及部署變更的工作量。

在 AWS 中，您可以使用 AWS 開發人員工具 等服務 (例如，AWS CodeCommit、 AWS CodeBuild、AWS CodePipeline、 AWS CodeDeploy和 AWS CodeStar) 建立持續整合/持續部署 (CI/CD) 管道。

常用的反模式：

- 您手動將更新部署到跨整個機群的應用程式伺服器，而由於更新錯誤，許多伺服器無法回應。
- 您在數小時內手動部署到應用程式伺服器機群。變更期間的版本不一致會造成未預期的行為。

建立此最佳實務的優勢： 採用部署管理系統可減少部署變更的工作量，以及手動程序造成的錯誤頻率。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 使用部署管理系統：使用部署管理系統來追蹤並實作變更。這可減少由手動流程引起的錯誤，並減少部署變更的工作量。從程式碼簽入到測試、部署和驗證，自動化整合和部署管道。此舉可減少前置時間，增加變更頻率，並進一步降低工作量。
  - AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
  - 什麼是 AWS CodeDeploy？
  - 什麼是 AWS Elastic Beanstalk？
  - 什麼是 Amazon API Gateway？

資源

相關文件：

- AWS CodeDeploy 使用者指南
- AWS 開發人員工具
- 在 AWS CodeDeploy 中嘗試範例藍/綠部署
- 什麼是 AWS CodeDeploy？
- 什麼是 AWS Elastic Beanstalk？
- 什麼是 Amazon API Gateway？

相關影片：

- 深入了解使用 AWS 的進階持續交付技術
- AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署

OPS06-BP04 使用有限的部署進行測試

繼續現有系統現有系統的同時，在有限的部署中進行測試，以在大規模部署之前確認理想成果達成與否。例如，使用部署 Canary 測試或一體式部署。

常用的反模式：

- 您一次性將失敗的變更部署至所有生產環境。您不知道。

建立此最佳實務的優勢： 透過在受限部署之後測試和驗證變更，您可以及早識別出問題，並將對客戶的影響降至最低，從而有機會進一步緩解對客戶的影響。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 使用有限的部署進行測試：使用有限的部署搭配現有系統進行測試，以在大規模部署之前確認理想成果達成與否。例如，使用部署 Canary 測試或一體式部署。
  - AWS CodeDeploy 使用者指南
  - 使用 AWS Elastic Beanstalk 進行藍/綠部署
  - 設定 API Gateway Canary 版本部署
  - 在 AWS CodeDeploy 中嘗試範例藍/綠部署
  - 在 AWS CodeDeploy 中使用部署組態

資源

相關文件：

- AWS CodeDeploy 使用者指南
- 使用 AWS Elastic Beanstalk 進行藍/綠部署
- 設定 API Gateway Canary 版本部署
- 在 AWS CodeDeploy 中嘗試範例藍/綠部署
- 在 AWS CodeDeploy 中使用部署組態

OPS06-BP05 使用平行環境進行部署

在平行環境中實作變更，然後轉換到新環境。維護先前的環境，直到確認已成功部署為止。此舉可透過還原到先前的環境來將還原時間減至最少。

常用的反模式：

- 您透過修改現有系統來執行可變部署。發現變更失敗之後，您必須再次修改系統以還原延長復原時間的舊版本。
- 在維護時段期間，您停用舊環境，然後開始建置新的環境。執行程序多個小時之後，您發現部署無法復原的問題。雖然非常疲倦，但您仍被迫找到先前的部署程序，並開始重建舊環境。

建立此最佳實務的優勢： 透過使用平行環境，您可以預先部署新的環境，並在需要時轉換至這些環境。如果新環境不成功，您可以轉換回原始環境來快速復原。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 使用平行環境進行部署：在平行環境上實作變更，然後轉換到切換新環境。維護先前的環境，直到確認已成功部署為止。此舉可透過還原到先前的環境來將還原時間減至最少。例如，將不可變的基礎架構用於藍/綠部署。
  - 在 AWS CodeDeploy 中使用部署組態
  - 使用 AWS Elastic Beanstalk 進行藍/綠部署
  - 設定 API Gateway Canary 版本部署
  - 在 AWS CodeDeploy 中嘗試範例藍/綠部署

資源

相關文件：

- AWS CodeDeploy 使用者指南
- 使用 AWS Elastic Beanstalk 進行藍/綠部署
- 設定 API Gateway Canary 版本部署
- 在 AWS CodeDeploy 中嘗試範例藍/綠部署
- 在 AWS CodeDeploy 中使用部署組態

相關影片：

- 深入了解使用 AWS 的進階持續交付技術

OPS06-BP06 部署頻繁、細微和可逆的變更

透過頻繁、細微和可逆的變更來縮小變更範圍。透過回復變更，可以更輕鬆地進行故障診斷並加快修復速度。

常用的反模式：

- 您在每一季都部署應用程式的新版本。
- 您經常變更資料庫結構描述。
- 您執行手動就地更新，並覆寫現有的安裝和組態。

建立此最佳實務的優勢： 您透過經常部署小的變更，更快認識到開發工作帶來的效益。當變更很小時，可更容易識別它們是否會產生意外的後果。如果變更可逆，由於復原得到簡化，實作變更的風險會更小。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 部署頻繁、細微、可逆的變更：透過頻繁、細微和可逆的變更來縮小變更範圍。透過回復變更，可以更輕鬆地進行故障診斷並加快修復速度。

OPS06-BP07 完全自動化整合和部署

自動化工作負載的建置、部署和測試。此舉可減少由手動程序引起的錯誤，以及部署變更的工作量。

依照一致的標記策略，使用 資源標籤 和 AWS Resource Groups 來套用 中繼資料， 以識別您的資源。標記您的資源，以用於組織、成本會計、存取控制，以及將自動執行營運活動設為目標。

常用的反模式：

- 週五，您完成了為功能分支編寫新程式碼。週一，在執行您的程式碼品質測試指令碼和每個單位測試指令碼之後，您將針對下一排程版本來檢查程式碼。
- 系統會指派您編寫修正程式碼，以解決影響生產環境中大量客戶的重大問題。測試修正後，您遞交程式碼和電子郵件變更管理內容，以請求核准將其部署到生產環境中。

建立此最佳實務的優勢： 透過實作自動化建置和部署管理系統，您可以減少手動程序引起的錯誤，以及部署變更的工作量，讓您的團隊成員能夠專注於提供商業價值。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 使用建置和部署管理系統：使用建置和部署管理系統來追蹤和實作變更，以減少由手動流程引起的錯誤，並減少工作量。從程式碼簽入到建置、測試、部署和驗證，完全自動化整合和部署管道。此舉可減少前置時間，增加變更頻率，並降低工作量。
  - 什麼是 AWS CodeBuild？
  - 軟體開發持續整合最佳實務
  - Slalom：AWS 上適用於無伺服器應用程式的 CI/CD
  - AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
  - 什麼是 AWS CodeDeploy？
  - 深入了解使用 AWS 的進階持續交付技術

資源

相關文件：

- 在 AWS CodeDeploy 中嘗試範例藍/綠部署
- 什麼是 AWS CodeBuild？
- 什麼是 AWS CodeDeploy？

相關影片：

- 軟體開發持續整合最佳實務
- 深入了解使用 AWS 的進階持續交付技術
- AWS CodeDeploy 簡介 - 使用 Amazon Web Services 進行自動化軟體部署
- Slalom：AWS 上適用於無伺服器應用程式的 CI/CD

OPS06-BP08 自動化測試和復原

自動測試部署的環境，以確認理想成果達成與否。當無法實現結果時，自動還原到先前的良好狀態，以最大限度縮短還原時間，並減少由手動程序引起的錯誤。

常用的反模式：

- 您將變更部署至工作負載。在看到變更完成之後，您開始部署後測試。在您看到它們完成之後，您會發現工作負載無法運作，且客戶中斷連線。然後您開始復原到之前的版本。經過長時間偵測問題後，手動重新部署會延長復原時間。

建立此最佳實務的優勢： 透過在部署之後測試和驗證變更，您可以立即識別出問題。透過自動復原至舊版本，將對客戶的影響降至最低。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 自動化測試和還原：自動測試部署的環境，以確認理想成果達成與否。當無法實現結果時，自動還原到先前的良好狀態，以最大限度縮短還原時間，並減少由手動程序引起的錯誤。例如，在部署後執行詳細的綜合使用者事務，驗證結果，並在失敗時復原。
  - 使用 AWS CodeDeploy 重新部署和回復部署

資源

相關文件：

- 使用 AWS CodeDeploy 重新部署和回復部署

OPS 7 您如何知道自己準備好支援工作負載？

評估工作負載、流程和程序及人員的營運準備度，了解工作負載相關營運風險。

最佳實務

- OPS07-BP01 確保人員能力
- OPS07-BP02 確保對營運準備度進行一致的審查
- OPS07-BP03 使用執行手冊執行程序
- OPS07-BP04 使用程序手冊來調查問題
- OPS07-BP05 做出部署系統和變更的明智決策

OPS07-BP01 確保人員能力

建立一種機制，用於驗證您有適當數量受過培訓的人員來為營運需求提供支援。培訓人員並根據需要調整人員能力，以保持有效的支援。

您需要擁有足夠的團隊成員，以妥善應對所有活動 (包括隨時待命)。確保您的團隊具備取得成功所需的必要技能，並進行工作負載、操作工具和 AWS 的培訓。

AWS 提供了許多資源，包括 AWS 入門資源中心， AWS 部落格， AWS 線上技術會談， AWS 活動和網路研討會以及 AWS Well-Architected 實驗室，而這些資源均提供了可教育您的團隊的說明、範例和演練。此外， AWS 培訓 和認證 透過 AWS 基礎原理自主進度數位課程提供一些免費培訓。您還可以報名參加講師指導下的培訓，以進一步協助開發團隊的 AWS 技能。

常用的反模式：

- 在沒有能夠支援使用中平台和服務的熟練團隊成員之情況下，部署工作負載。
- 在預期支援時數內無團隊成員可用的情況下部署工作負載。
- 在團隊成員休假或生病請假而無足夠團隊成員提供支援的情況下，部署工作負載。
- 在未檢閱對支援該工作負載和其他工作負載之團隊成員的額外影響情況下，部署額外工作負載。

建立此最佳實務的優勢： 擁有熟練的團隊成員可有效支援您的工作負載。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 人員能力：驗證人員是否已經過充分培訓，可有效支援工作負載。
  - 團隊規模：確保擁有足夠且訓練有素的團隊成員，以妥善應對營運活動，包括隨時待命。
  - 團隊技能：確保您的團隊成員就 AWS、工作負載和營運工具獲得足夠培訓，可履行其職責。
    - AWS 活動和網路研討會
    - 歡迎來到 AWS 培訓 培訓與認證
  - 審查能力：隨著營運條件和工作負載變化，審查團隊的規模和技能，以確保有足夠能力維持卓越營運。進行調整以確保團隊規模和技能與團隊支援的工作負載的營運要求相匹配。

資源

相關文件：

- AWS 部落格
- AWS 活動和網路研討會
- AWS 入門資源中心
- AWS 線上技術會談
- 歡迎來到 AWS 培訓 培訓與認證

相關範例：

- Well-Architected 實驗室

OPS07-BP02 確保對營運準備度進行一致的審查

使用營運準備度審查 (ORR)，來確認您可以運行工作負載。 ORR 是在 Amazon 開發的機制，可確認團隊是否可放心地運行工作負載。ORR 是使用需求檢查清單的審查和檢查程序。ORR 是一種自助服務體驗，團隊會透過此體驗來進行工作負載的認證。ORR 包含的最佳實務皆汲取我們多年來建置軟體所獲得的經驗。

ORR 檢查清單包含架構建議、營運程序、事件管理和發行品質。錯誤糾正 (CoE) 程序是這些項目的主要驅動要素。您專屬的事件後分析應有助於專屬 ORR 的發展。ORR 不只是遵循最佳實務，還能防止先前發生過的事件再發。最後，ORR 中也能夠包含安全性、管控和合規需求。

在工作負載啟動以全面供應前，並在整個軟體開發生命週期執行 ORR。在啟動前執行 ORR 可改善安全運行工作負載的能力。定期針對工作負載重新執行 ORR 可捕捉最佳實務中的任何偏移。您可以為新服務的推出制定 ORR 檢查清單，並為定期審查制定 ORR。此可協助您掌握新出現的最佳實務最新狀態，並採納從事件後分析獲得的經驗。隨著您可以更熟練地使用雲端後，您就可以在架構中建置 ORR 需求作為預設值。

預期成果： 您制定 ORR 檢查清單，內含組織的最佳實務。ORR 會在工作負載啟動前執行。ORR 會在工作負載生命週期的過程中定期執行。

常見的反模式：

- 您啟動工作負載，但不知道自己是否能夠運行工作負載。
- 啟動工作負載的認證中未納入管控和安全性需求。
- 不會定期重新評估工作負載。
- 工作負載啟動，但不需設置必要的程序。

- 您可以在多個工作負載中看到重複出現的相同根本原因失敗。

建立此最佳實務的優勢：

- 工作負載包含架構、程序和管理最佳實務。
- 經驗已納入 ORR 程序中。
- 工作負載啟動時，已設置必要的程序。
- ORR 會在工作負載的整個軟體生命週期執行。

若未建立此最佳實務的風險等級： 高

實作指引

ORR 有兩個部分：程序和檢查清單。貴組織應採用 ORR 程序，並由執行主辦人支援此程序。至少，必須在工作負載啟動以全面供應前執行 ORR。在整個軟體開發生命週期執行 ORR，使其與最佳實務或新需求保持同步。ORR 檢查清單應包含組態項目、安全性和管控需求，以及來自貴組織的最佳實務。在經過一段時間後，您可以使用服務，例如 AWS Config、 AWS Security Hub，和 AWS Control Tower 防護機制，來將 ORR 中的最佳實務建置在防護機制中，以便自動偵測最佳實務。

客戶範例

在發生數個生產事件後，AnyCompany Retail 決定實作 ORR 程序。他們建立了一份檢查清單，其中由最佳實務、管控和合規需求，以及從中斷中汲取的經驗教訓所組成。在工作負載啟動前，新的工作負載會執行 ORR。每個工作負載每年都會使用一部分的最佳實務來執行 ORR，以便納入在 ORR 檢查清單中新增的最佳實務和需求。經過一段時間後，AnyCompany Retail 使用 AWS Config 來偵測最佳實務，進而縮短 ORR 程序的時間。

實作步驟

若要進一步了解 ORR，請閱讀 「營運準備度審查 (ORR)」白皮書。其中提供詳細的資訊，說明 ORR 程序的歷史、如何建立您專屬的 ORR 實務，以及如何制定 ORR 檢查清單。以下步驟是該文件的精簡版本。如需深入了解 ORR 是什麼，以及如何建立您專屬的 ORR，我們建議閱讀該白皮書。

1. 召集關鍵利害關係人，包含安全性、營運和開發等團隊的代表人員。
2. 請每位利害關係人提供至少一個需求。對於第一次的反覆測試，請嘗試將項目數限制在三十個以下。
   - 附錄 B：來自「營運準備度審查 (ORR)」白皮書的 ORR 問題範例包含您可以開始使用的範例問題。

3. 將需求集中放在試算表中。
    - 您可以使用 在 AWS Well-Architected Tool 中 使用自訂聚焦 來制定 ORR 並在帳戶和 AWS 組織之間進行共用。
4. 找出要在其中執行 ORR 的一個工作負載。啟動前的工作負載或內部工作負載是理想的選擇。
5. 演練 ORR 檢查清單，並記下任何所探索的項目。如果採取緩解措施，那就可能無法進行探索。對於缺少緩解措施的任何探索，請將那些探索新增至項目的待辦清單中，然後在啟動前加以實作。
6. 隨著時間持續在 ORR 檢查清單中新增最佳實務和需求。

使用 Enterprise Support 的 AWS Support 客戶可請求 「營運準備度審查」研討會 (透過其技術客戶經理)。研討會是互動式的 逆向思維 課程，可讓您制定自己的 ORR 檢查清單。

實作計劃的工作量： 高。在組織中採用 ORR 實務需要高層和利害關係人的支持。使用貴組織提供的各方意見，來建立和更新檢查清單。

資源

相關的最佳實務：

- OPS01-BP03 評估管控要求 – ORR 檢查清單原本就很適合用來管控需求。
- OPS01-BP04 評估合規要求 – ORR 檢查清單中有時會包含合規需求。有些時候，它們會是獨立的程序。
- OPS03-BP07 適當地為團隊提供資源 – 團隊能力是 ORR 需求的絕佳候選項。
- OPS06-BP01 為失敗變更進行規劃 – 啟動工作負載前，必須先建立回復或向前回復計劃。
- OPS07-BP01 確保人員能力 – 若要支援工作負載，您必須具備所需的人員。
- SEC01-BP03 識別和驗證控制目標 – 安全性控制目標是絕佳的 ORR 需求。
- REL13-BP01 定義停機和資料遺失的復原目標 – 災難復原計劃是絕佳的 ORR 需求。
- COST02-BP01 根據貴組織的需求制定政策 – 將成本管理政策納入 ORR 檢查清單是很棒的做法。

相關文件：

- AWS Control Tower - AWS Control Tower 中的防護機制
- AWS Well-Architected Tool - 自訂聚焦
- Adrian Hornsby 提供的營運準備度審查範本
- 「營運準備度審查 (ORR)」白皮書

相關影片：

- AWS Support 為您提供支援 | 建立有效的營運準備度審查 (ORR)

相關範例：

- 營運準備度審查 (ORR) 聚焦範例

相關服務：

- AWS Config
- AWS Control Tower
- AWS Security Hub
- 使用自訂聚焦

OPS07-BP03 使用執行手冊執行程序

路由層 執行手冊 是為了實現特定結果而記錄的程序。執行手冊由一系列可供遵循以完成某項工作的步驟組成。早在航空器製造初期，操作過程中就會使用執行手冊。在雲端操作中，我們使用執行手冊來降低風險及達到預期成果。簡言之，執行手冊就是完成一項工作的檢查清單。

執行手冊是工作負載的運作不可或缺的部分。從新團隊成員的上線到部署主要版本，執行手冊無論由誰使用，都是可提供一致結果的編碼程序。執行手冊應在集中發佈，並隨著程序的演進而更新，因為更新執行手冊是變更管理程序的重要環節。其中也應包含關於問題發生時的錯誤處理、工具、許可、例外狀況和呈報的指引。

隨著組織的成熟，您可以開始將執行手冊自動化。請從簡短且常用的執行手冊開始著手。使用指令碼語言自動執行步驟，或使步驟較容易執行。前幾個執行手冊完成自動化後，您會專注於將較複雜的執行手冊自動化。經過一段時間後，您大多數的執行手冊應該都已做了某種程度的自動化。

預期成果： 您的團隊有一系列執行工作負載任務的逐步指南。執行手冊中包含預期成果、必要的工具和許可，以及錯誤處理指示。這些執行手冊會集中存放，並且經常更新。

常見的反模式：

- 憑藉記憶完成程序中的每個步驟。
- 手動部署變更而不使用檢查清單。
- 不同的團隊成員執行相同程序，但使用的步驟不同，或結果不同。

- 執行手冊失去與系統變更和自動化的同步。

建立此最佳實務的優勢：

- 降低手動工作的錯誤率。
- 以一致的方式執行操作：
- 新的團隊成員可更快開始執行工作。
- 可將執行手冊自動化以節省人力。

未建立此最佳實務時的曝險等級： 中

實作指引

根據組織的成熟度，執行手冊採取數種形式。其中至少應包含逐步說明文字文件。預期成果應明確指出。明確記載必要的特殊許可或工具。提供詳細指引，說明在發生狀況時應如何處理錯誤及呈報。列出執行手冊擁有者，並將其集中發佈。執行手冊列入文件後，應請團隊的其他成員加以執行，以進行驗證。隨著程序的演進，請根據您的變更管理程序更新執行手冊。

隨著組織逐漸成熟，您的文字執行手冊應該要自動化。使用諸如 AWS Systems Manager 自動化的服務，您可以將一般文字轉換為可對工作負載執行的自動化。這些自動化可作為事件的應變動作來執行，以降低您維持工作負載的操作負擔。

客戶範例

AnyCompany Retail 必須在軟體部署期間執行資料庫結構描述更新。雲端維運團隊與資料庫管理團隊共同建置用來手動部署這些變更的執行手冊。執行手冊以檢查清單格式列出了程序中的每個步驟。其中包含相關發生狀況時進行錯誤處理的章節。他們將執行手冊發佈於內部 Wiki，與其他執行手冊放在一起。雲端維運團隊規劃要在未來的衝刺期間將執行手冊自動化。

實作步驟

如果您沒有現有的文件儲存庫，版本控制儲存庫將是您開始建置執行手冊程式庫的絕佳選擇。您可以使用 Markdown 來建置執行手冊。我們提供了範例執行手冊範本，讓您用來開始建置執行手冊。

```
# ###### ## ###### | #### ID | ## | ##### | #### | ###### | ###### | ## POC |
 |-------|-------|-------|-------|-------|-------|-------| | RUN001 | ########### #####
## | ## | ## | #### | 2022-09-21 | #### | ## ## 1.### 2.###
```

1. 如果您沒有現有的文件儲存庫或 Wiki，請在您的版本控制系統中建立新的版本控制儲存庫。

2. 識別沒有執行手冊的程序。經常執行、步驟數較少，且失敗的影響程度不高的程序，就是理想的程序。
3. 在您的文件儲存庫中，使用範本建立新的草稿 Markdown 文件。填入 ######## 和必要欄位 (在 ## ########)。
4. 從第一個步驟開始，填入執行手冊的 #### 部分。
5. 將執行手冊提供給團隊成員。讓他們使用執行手冊來驗證步驟。如有任何事項缺漏或需要釐清，請更新執行手冊。
6. 將執行手冊發佈至您的內部文件存放區。發佈後，請告知團隊和其他利害關係人。
7. 一段時間後，您會建置執行手冊程式庫。隨著該程式庫的擴增，您應開始設法將執行手冊自動化。

實作計劃的工作量： 低。執行手冊的最低標準是逐步文字指南。將執行手冊自動化可能會增加實作工作量。

資源

相關的最佳實務：

- OPS02-BP02 已為流程和程序識別擁有者：執行手冊應有擁有者負責加以維護。
- OPS07-BP04 使用程序手冊來調查問題：執行手冊和程序手冊兩者相類似，但有一項顯著差異：執行手冊有預期成果。在許多情況下，當程序手冊識別出根本原因時，就會觸發執行手冊。
- OPS10-BP01 使用程序進行事件、事故和問題管理：執行手冊是良好事件、事故和問題管理實務的一部分。
- OPS10-BP02 每個提醒建立一個程序：執行手冊和程序手冊應用來回應提醒。一段時間後，這些因應動作應該要自動化。
- OPS11-BP04 知識管理：維護執行手冊是知識管理的重要環節。

相關文件：

- 使用自動化的執行手冊和程序手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- AWS 大型遷移的遷移程序手冊 - 任務 4：改進您的遷移執行手冊
- 使用 AWS Systems Manager 自動化執行手冊完成營運任務

相關影片：

- AWS re:Invent 2019：執行手冊、事故報告和事故應變的 DIY 指南 (SEC318-R1)
- 如何使用 AWS | Amazon Web Services 將 IT 營運自動化
- 將指令碼整合到 AWS Systems Manager 中

相關範例：

- AWS Systems Manager：自動化演練
- AWS Systems Manager：從最新的快照執行手冊還原根磁碟區
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事故應變執行手冊
- Gitlab - 執行手冊
- Rubix - 用來在 Jupyter 筆記本中建置執行手冊的 Python 程式庫
- 使用 Document Builder 建立自訂執行手冊
- Well-Architected 實驗室：使用程序手冊和執行手冊將操作自動化

相關服務：

- AWS Systems Manager

OPS07-BP04 使用程序手冊來調查問題

程序手冊是用來調查事件的逐步指南。事件發生時，我們會使用程序手冊來調查、確認影響範圍和找出根本原因。程序手冊可用於各種情境，從部署失敗到安全性事件皆涵蓋在內。在許多案例中，程序手冊可釐清根本原因，而執行手冊則用來緩解該根本原因。程序手冊是組織事件應變計劃的關鍵要素。

優良的程序手冊有幾個重要的特點。它會透過探索的過程來逐步引導使用者。請試著從各種角度思考，我們應遵循哪些步驟來診斷事件？ 透過程序手冊明確定義，在程序手冊中是否需要特殊工具或提高權限。制定溝通計劃，向利害關係人告知調查的最新狀態是關鍵要素。在無法釐清根本原因的狀況下，程序手冊應具備呈報計劃。如果已確定根本原因，程序手冊應指向執行手冊，後者會描述如何解決該根本原因。程序手冊應集中存放並定期維護。如果您使用程序手冊來發出特定警示，請為團隊提供警示中該程序手冊的指標。

隨著組織逐漸成熟，將程序手冊自動化。從涵蓋低風險事件的程序手冊開始。使用指令碼來自動化探索步驟。確保您有配套執行手冊來緩解常見的根本原因。

預期成果： 您的組織具備常見事件的程序手冊。該程序手冊存放在集中的位置，可供團隊成員使用。程序手冊會頻繁更新。對於任何已知的根本原因，都已建立配套執行手冊。

常見的反模式：

- 調查事件並沒有標準的方法。
- 團隊成員依賴肌肉記憶或機構知識，來針對失敗的部署進行疑難排解。
- 新團隊成員會學習如何透過試錯來調查問題。
- 各個團隊間並未共用調查問題的最佳實務。

建立此最佳實務的優勢：

- 程序手冊可為您省下緩解事件所需的心力。
- 不同的團隊成員可以使用相同的程序手冊，以一致的方式找出根本原因。
- 您可以為已知的根本原因制定執行手冊，進而縮短復原時間。
- 程序手冊可協助團隊成員更快開始做出貢獻。
- 團隊可以透過可重複的程序手冊擴展其程序。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

您如何根據組織的成熟度來建立和使用程序手冊。如果您剛接觸雲端，請在中央文件儲存庫中建立文字形式的程序手冊。隨著組織逐漸成熟，您就可以透過 Python 之類的指令碼語言將程序手冊半自動化。您可以在 Jupyter 筆記本中執行這些指令碼來加快探索速度。先進的組織具有全自動化的程序手冊，這些手冊適用於透過執行手冊自動修復的常見問題。

透過列出在您工作負載中發生的常見事件，來開始建立程序手冊。為低風險以及根本原因的範圍已縮減至幾個問題的事件選擇程序手冊，然後開始。在您為較簡單情境建立程序手冊後，請接著嘗試風險較高或尚未確定根本原因的情境。

隨著組織逐漸成熟，應將您的文字程序手冊自動化。使用諸如 AWS Systems Manager Automations 的服務，可以將一般文字轉換為自動化。您可以針對工作負載執行這些自動化來加快調查速度。您可以啟動這些自動化來回應事件、縮短事件探索和解決的平均時間。

客戶可使用 AWS Systems Manager Incident Manager 來回應事件。此服務提供單一介面，來分類事件、在探索和緩解期間通知利害關係人，並在整個事件期間進行合作。其使用 AWS Systems Manager Automations 來加快偵測和復原速度。

客戶範例

生產事件會影響 AnyCompany Retail。待命的工程師使用程序手冊來調查問題。隨著透過步驟取得進展時，該工程師會確保程序手冊中識別的重要利害關係人都能了解最新進展。他發現根本原因是後端服務中的一項競賽條件。該工程師使用執行手冊，重新啟動服務，使 AnyCompany Retail 重新上線。

實作步驟

如果您沒有現有的文件儲存庫，我們建議為程序手冊程式庫建立版本控制儲存庫。您可以使用 Markdown 建立程序手冊，Markdown 與多數程序手冊自動化系統都相容。如果您是從頭開始建立，請使用以下範例程序手冊範本。

```
# Playbook Title ## Playbook Info | Playbook ID | Description
 | Tools Used | Special Permissions | Playbook Author | Last
 Updated | Escalation POC | Stakeholders | Communication Plan |
 |-------|-------|-------|-------|-------|-------|-------|-------|-------| | RUN001
 | What is this playbook for? What incident is it used for? | Tools | Permissions |
 Your Name | 2022-09-21 | Escalation Name | Stakeholder Name | How will updates be
 communicated during the investigation? | ## Steps 1.Step one 2.Step two
```

1. 如果您沒有現有的文件儲存庫或 Wiki，請在版本控制系統中為程序手冊建立新的版本控制儲存庫。
2. 找出需要調查的常見問題。應存在根本原因僅限於幾個問題的情境，解決方案的風險很低。
3. 使用 Markdown 範本，填寫 ###### 區段以及程序手冊資訊 ####。
4. 填寫疑難排解步驟。盡可能清楚說明要執行哪些動作或應調查哪些地方。
5. 將程序手冊提供給團隊成員，讓成員透過該手冊來進行驗證。如果缺少任何資訊或內容不清楚，請更新程序手冊。
6. 在文件儲存庫中發佈程序手冊，並通知團隊和任何利害關係人。
7. 此程序手冊程式庫會隨著您新增更多程序手冊而成長。在您有數本程序手冊後，請開始使用 AWS Systems Manager Automations 之類的工具來進行自動化，進而確保自動化和程序手冊都能保持同步。

實作計劃的工作量： 低。程序手冊應為集中存放的文字文件。越來越多發展成熟的組織會開始自動化程序手冊。

資源

相關的最佳實務：

- OPS02-BP02 已為流程和程序識別擁有者：程序手冊應有擁有者，擁有者會負責維護這類手冊。

- OPS07-BP03 使用執行手冊執行程序：執行手冊和程序手冊兩者相類似，但有一項顯著差異：執行手冊有預期成果。在許多情況下，當程序手冊找出根本原因時，就會使用執行手冊。
- OPS10-BP01 使用程序進行事件、事故和問題管理：程序手冊是正常事件、事故和問題管理實務的一部分。
- OPS10-BP02 每個提醒建立一個程序：執行手冊和程序手冊應用來回應警示。一段時間後，應將這些因應措施自動化。
- OPS11-BP04 知識管理：維護程序手冊是知識管理的重要環節。

相關文件：

- 使用自動化的執行手冊和程序手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- 使用 AWS Systems Manager Automation 執行手冊解決營運任務

相關影片：

- AWS re:Invent 2019：執行手冊、事件報告和事件應變的 DIY 指南 (SEC318-R1)
- AWS Systems Manager Incident Manager - AWS 虛擬研討會
- 將指令碼整合到 AWS Systems Manager 中

相關範例：

- AWS 客戶程序手冊架構
- AWS Systems Manager：自動化演練
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事件應變執行手冊
- Rubix - 用來在 Jupyter 筆記本中建置執行手冊的 Python 程式庫
- 使用 Document Builder 建立自訂執行手冊
- Well-Architected 實驗室：使用程序手冊和執行手冊將操作自動化
- Well-Architected 實驗室：使用 Jupyter 事件應變程序手冊

相關服務：

- AWS Systems Manager Automation
- AWS Systems Manager Incident Manager

OPS07-BP05 做出部署系統和變更的明智決策

評估團隊支援工作負載的能力以及工作負載對管控的遵從性。在確定是否轉換系統或將系統投入生產時，比照這些評估部署的收益。了解收益和風險，以做出明智決策。

事前剖析是一種演練，其中團隊會模擬失敗以開發緩解策略。使用事前剖析可預測失敗並適時建立程序。當您變更您用於評估工作負載的檢查清單時，請計劃如何處理不再合規的即時系統。

常用的反模式：

- 在不了解工作負載中存在安全性風險的情況下，決定部署工作負載。
- 在不了解工作負載是否符合您的管控和標準的情況下，決定部署工作負載。
- 在不了解您的團隊是否能支援此工作負載的情況下，決定部署工作負載。
- 在不了解工作負載對組織有何好處的情況下，決定部署工作負載。

建立此最佳實務的優勢： 擁有熟練的團隊成員可有效支援您的工作負載。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 做出部署工作負載和變更的明智決策：評估團隊支援工作負載的能力以及工作負載對管控的合規性。在確定是否轉換系統或將系統投入生產時，比照這些評估部署的收益。了解收益和風險，並做出明智的決策。

營運

問題

- OPS 8 您如何了解工作負載的運作狀態？
- OPS 9 您如何了解營運狀況？
- OPS 10 您如何管理工作負載和營運事件？

OPS 8 您如何了解工作負載的運作狀態？

定義、擷取和分析工作負載指標，掌握工作負載事件，以便採取適當行動。

最佳實務

OPS08-BP01 識別關鍵績效指標

根據所需的業務成果 (例如，訂單率、客戶保留率以及獲利與營運支出的對比) 與客戶成果 (例如，客戶滿意度)，識別關鍵績效指標 (KPI)。評估 KPI 以確定工作負載是否成功。

常用的反模式：

- 企業領導階層會詢問您工作負載如何成功滿足業務需求，但您卻沒有可判斷成功與否的參考框架。
- 您無法判斷為組織營運的商用現成應用程式是否具成本效益。

建立此最佳實務的優勢： 藉由識別關鍵績效指標，您可以實現業務成果，做為對工作負載運作狀態和成功的測試。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 識別關鍵績效指標：根據所需的業務和客戶成果識別關鍵績效指標 (KPI)。評估 KPI 以確定工作負載是否成功。

OPS08-BP02 定義工作負載指標

定義工作負載指標以衡量 KPI 的實現情況 (例如，捨棄的購物車、下單的訂單、成本、價格和分配的工作負載支出)。定義工作負載指標以衡量工作負載的運作狀態 (例如，界面回應時間、錯誤率、提出的請求、完成的請求和使用率)。評估指標以判斷工作負載是否取得了預期的成果，並了解工作負載的運作狀態。

您應該將日誌資料傳送到 CloudWatch Logs 這類服務，並從必要日誌內容的觀察中產生指標。

CloudWatch 擁有專業功能，例如 適用於 .NET 和 SQL Server 的 Amazon CloudWatch 和 Container Insights， 這些功能可協助您在各個特定支援應用程式資源和技術堆疊之間識別並設定關鍵指標、日誌和警示。

常用的反模式：

- 您已定義與任何 KPI 無關或針對任何工作負載量身打造的標準指標。
- 您的指標計算中有錯誤，這會產生無效的結果。
- 您尚未為工作負載定義任何指標。
- 您只衡量了可用性。

建立此最佳實務的優勢： 透過定義和評估工作負載指標，您可以判斷工作負載的運作狀態，並衡量業務成果的實現情況。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 定義工作負載指標：定義工作負載指標以衡量 KPI 的實現情況。定義工作負載指標，以衡量工作負載及其各個元件的運作狀態。評估指標以確定工作負載是否取得了預期的成果，並了解工作負載的運作狀態。
  - 發佈自訂指標
  - 搜尋和篩選日誌資料
  - Amazon CloudWatch 指標和維度參考

資源

相關文件：

- Amazon CloudWatch 指標和維度參考
- 發佈自訂指標
- 搜尋和篩選日誌資料

OPS08-BP03 收集和分析工作負載指標

定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。

您應該將應用程式、工作負載元件、服務和 API 呼叫的日誌資料彙總至像是 CloudWatch Logs 等服務中。從必要日誌內容的觀察中產生指標，以深入了解營運活動的效能。

在 AWS 上，您可以分析工作負載指標並識別操作問題，方法為使用 Amazon DevOps Guru機器學習功能。AWS DevOps Guru 會提供操作問題的通知，隨附 針對性和主動 建議，以解決問題並維護應用程式運作狀態。

在 AWS 共同責任模式中，監控部分可透過下列項目提供給您： AWS Health Dashboard。此儀表板會在 AWS 發生可能影響您的事件時，提供提醒與修補指導。商業和企業支援訂閱客戶還可存取 AWS Health API，進而可以整合至其事件管理系統。

在 AWS 上，您可以 將日誌資料匯出至 Amazon S3 或者 直接傳送日誌 至 Amazon S3 以進行長期儲存。您可以使用 AWS Glue，在 Amazon S3 中探索和準備日誌資料，以進行分析並將關聯的中繼資料儲存在 AWSAWS Glue Data Catalog。 Amazon AthenaAmazon Athena，透過與 AWS Glue 的原生整合，可用來分析日誌資料，並使用標準 SQL 進行查詢。使用 Amazon QuickSight 這類商業智慧工具來視覺化、探索和分析您的資料。

另一個 解決方案 是使用 Amazon OpenSearch Service 和 OpenSearch 儀表板 跨多個帳戶和 AWS 區域 收集、分析和顯示 AWS 上的日誌。

常用的反模式：

- 網路設計團隊會詢問您目前的網路頻寬使用率。您提供目前的指標，網路使用率為 35%。由於時間點測量並未反映使用率趨勢，因此它們降低電路容量，以作為節省成本的措施，這導致廣泛的連線問題。
- 您的路由器失敗。它一直在記錄非關鍵的記憶體錯誤，隨著頻率越來越大，直到完全失敗。您未偵測到此趨勢，因此在路由器造成服務中斷之前，並未取代出錯的記憶體。

建立此最佳實務的優勢： 透過收集和分析工作負載指標，您可以了解工作負載的運作狀態，並深入了解可能影響工作負載或達成業務成果的趨勢。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 收集和分析工作負載指標：定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。
  - 使用 Amazon CloudWatch 指標
  - Amazon CloudWatch 指標和維度參考

- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌

資源

相關文件：

- Amazon Athena
- Amazon CloudWatch 指標和維度參考
- Amazon DevOps Guru
- AWS Glue
- AWSAWS Glue Data Catalog
- Amazon OpenSearch Service
- AWS Health Dashboard
- Amazon QuickSight
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- 使用 Amazon CloudWatch 指標

OPS08-BP04 建立工作負載指標基準

為指標建立基準，以提供期望值，做為比較和識別效能欠佳和過剩的元件的基礎。識別用於改善、調查和介入的閾值。

常用的反模式：

- 伺服器在 95% 的 CPU 使用率下執行，系統會詢問您情況是好或壞。該伺服器的 CPU 使用率尚未進行基準比較，因此您不知道這是好或壞。

建立此最佳實務的優勢： 透過定義基準指標值，您可以評估目前的指標值和指標趨勢，以判斷是否需要採取動作。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 為工作負載指標建立基準：為工作負載指標建立基準，以提供期望值做為比較的基礎。

- 建立 Amazon CloudWatch 警示

資源

相關文件：

- 建立 Amazon CloudWatch 警示

OPS08-BP05 了解工作負載的預期活動模式

建立工作負載活動模式以識別異常行為，以便您可以在需要時做出適當回應。

CloudWatch 透過 CloudWatch 異常偵測 功能套用統計和機器學習演算法，以產生代表一般指標行的預期值範圍。

Amazon DevOps Guru 可以用來透過事件關聯性、日誌分析和套用機器學習分析您的工作負載遙測來識別異常行為。當偵測到非預期行為時，它會提供 相關的指標和事件， 隨附處理行為的建議。

常用的反模式：

- 您正在檢閱網路使用率日誌，並看到網路使用率在上午 11:30 到下午 1:30 之間增加，然後在下午 4:30 到下午 6:00 之間再次增加。您不知道這是否應該視為正常。
- 您的 Web 伺服器每天凌晨 3:00 重新開機。您不知道這是否是預期的行為。

建立此最佳實務的優勢： 透過學習行為模式，您可以識別意外行為，並在必要時採取動作。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 了解工作負載的預期活動模式：建立工作負載活動模式，以確定行為何時超出預期值，以便您在需要時做出適當的回應。

資源

相關文件：

- Amazon DevOps Guru
- CloudWatch 異常偵測

OPS08-BP06 在工作負載結果有風險時發出提醒

當工作負載結果有風險時發出提醒，以便您可以在必要時做出適當的回應。

理想情況下，您先前已識別可對其發出警示的指標閾值，或已識別可用來觸發自動回應的事件。

在 AWS 上，您可以使用 Amazon CloudWatch Synthetics 建立 Canary 指令碼，來監控您的端點和 API，方法為執行與客戶相同的動作。產生的遙測和 取得的洞見 可讓您在客戶受到影響之前識別問題。

您也可以使用 CloudWatch Logs Insights ，透過專門打造的查詢語言，以互動的方式搜尋並分析日誌資料。CloudWatch Logs Insights 會自動 探索 AWS 服務日誌中的欄位， 以及 JSON 中的自訂日誌事件。它可以隨日誌量和查詢複雜性進行擴展，並在幾秒鐘內提供答案，以協助您搜尋事件的影響因素。

常用的反模式：

- 您沒有網路連線能力。沒有人注意到。沒有人嘗試找出原因，或採取動作來恢復連線。
- 套用修補程式之後，您的持久性執行個體不可用，從而中斷使用者。您的使用者已開啟支援案例。沒有人收到通知。沒有人採取動作。

建立此最佳實務的優勢： 透過識別業務成果已產生風險，並提醒採取動作，您就有機會防止或緩解事件的影響。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 在工作負載結果有風險時發出提醒：當工作負載結果有風險時發出提醒，以便您在需要時做出適當的回應。
  - 什麼是 Amazon CloudWatch Events？
  - 建立 Amazon CloudWatch 警示
  - 使用 Amazon SNS 通知呼叫 Lambda 函數

資源

相關文件：

- Amazon CloudWatch Synthetics
- CloudWatch Logs Insights

- 建立 Amazon CloudWatch 警示
- 使用 Amazon SNS 通知呼叫 Lambda 函數
- 什麼是 Amazon CloudWatch Events？

OPS08-BP07 在偵測到工作負載異常時發出提醒

當偵測到工作負載異常時發出提醒，以便您可以在必要時做出適當的回應。

透過長時間分析工作負載指標能夠建立可充分量化的行為模式，以定義事件或發出警示來回應。

經過訓練後， CloudWatch 異常偵測 功能可用於對偵測到的異常發出 警示 ，或提供重疊的預期值至指標資料 圖形 上，以便進行持續比較。

常用的反模式：

- 您的零售網站銷售額突然大幅增加。沒有人注意到。沒有人試圖找出導致此激增的原因。沒有人採取動作，以確保額外負載下的優質客戶體驗。
- 應用程式套用修補程式之後，您的持久性伺服器會經常重新啟動，從而中斷使用者。您的伺服器通常重新開機最多三次，但不會更多次。沒有人注意到。沒有人試圖找出發生此問題的原因。

建立此最佳實務的優勢： 透過了解工作負載行為的模式，您可以識別意外行為，並在必要時採取動作。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 在偵測到工作負載異常時發出提醒：當偵測到工作負載異常時發出提醒，以便您在需要時做出適當的回應。
  - 什麼是 Amazon CloudWatch Events？
  - 建立 Amazon CloudWatch 警示
  - 使用 Amazon SNS 通知呼叫 Lambda 函數

資源

相關文件：

- 建立 Amazon CloudWatch 警示

- CloudWatch 異常偵測
- 使用 Amazon SNS 通知呼叫 Lambda 函數
- 什麼是 Amazon CloudWatch Events？

OPS08-BP08 驗證結果的實現以及 KPI 和指標的有效性

建立工作負載營運的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。

AWS 還可透過 AWS 服務 API 和 SDK (例如 Grafana、Kibana 和 Logstash) 支援第三方日誌分析系統和商業智慧工具。

常用的反模式：

- 頁面回應時間從來未被視為有助於提高客戶滿意度的因素。您從未建立頁面回應時間的指標或閾值。您的客戶對於緩慢問題抱怨不已。
- 您尚未達到最低回應時間目標。為改善回應時間，您已擴展應用程式伺服器。您現在已大幅超出回應時間目標，而且已付費容量中還有大量未使用。

建立此最佳實務的優勢： 透過審查和修訂 KPI 和指標，您可以了解工作負載如何支援業務成果的達成，並找出達成業務目標需要改善的地方。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 驗證結果的實現以及 KPI 和指標的有效性：建立工作負載營運的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。
  - 使用 Amazon CloudWatch 儀表板
  - 什麼是日誌分析？

資源

相關文件：

- 使用 Amazon CloudWatch 儀表板

- 什麼是日誌分析？

OPS 9 您如何了解營運狀況？

定義、擷取和分析營運指標，掌握營運事件，以便採取適當行動。

最佳實務

- OPS09-BP01 識別關鍵績效指標
- OPS09-BP02 定義營運指標
- OPS09-BP03 收集和分析營運指標
- OPS09-BP04 建立營運指標基準
- OPS09-BP05 了解營運活動的預期模式
- OPS09-BP06 在營運成果有風險時發出警示
- OPS09-BP07 在偵測到營運異常時發出提醒
- OPS09-BP08 驗證結果的實現以及 KPI 和指標的有效性

OPS09-BP01 識別關鍵績效指標

根據所需的業務成果 (例如，交付的新功能) 和客戶成果 (例如，客戶支援案例)，識別關鍵績效指標 (KPI)。評估 KPI 以確定營運是否成功。

常用的反模式：

- 企業領導階層會詢問您是否成功完成業務目標，但您卻沒有可判斷成功與否的參考框架。
- 您無法判斷您的維護時段是否會影響業務成果。

建立此最佳實務的優勢： 藉由識別關鍵績效指標，您可以實現業務成果，做為對營運運作狀態和成功的測試。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 識別關鍵績效指標：根據所需的業務和客戶成果識別關鍵績效指標 (KPI)。評估 KPI 以確定營運是否成功。

OPS09-BP02 定義營運指標

定義營運指標以衡量 KPI 的實現情況 (例如，成功部署和失敗部署)。定義營運指標以衡量營運活動的運作狀態 (例如，偵測事件所需的平均時間 (MTTD)，以及從事件中復原所需的平均時間 (MTTR))。評估指標以判斷營運是否取得理想成果，並了解您的營運活動的運作狀態。

常用的反模式：

- 您的運營指標是以團隊認為合理的內容為基礎。
- 您的指標計算中有錯誤，這會產生不正確的結果。
- 您尚未為營運活動定義任何指標。

建立此最佳實務的優勢： 透過定義和評估營運指標，您可以判斷營運活動的運作狀態，並衡量業務成果的實現情況。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 定義營運指標：定義營運指標以衡量 KPI 的實現情況。定義營運指標以衡量營運及其活動的狀況。評估指標以確定營運是否取得理想成果，並了解營運狀況。
  - 發佈自訂指標
  - 搜尋和篩選日誌資料
  - Amazon CloudWatch 指標和維度參考

資源

相關文件：

- AWS Answers：集中式記錄
- Amazon CloudWatch 指標和維度參考
- 使用 Amazon CloudWatch Events 偵測管道狀態中的變更並做出反應
- 發佈自訂指標
- 搜尋和篩選日誌資料

相關影片：

- 制定監控計劃

OPS09-BP03 收集和分析營運指標

定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。

您應該將執行營運活動和操作 API 呼叫的日誌資料彙總至 CloudWatch Logs 這類服務中。從必要日誌內容的觀察中產生指標，以深入了解營運活動的效能。

在 AWS 上，您可以 將日誌資料匯出至 Amazon S3 或者 直接傳送日誌 至 Amazon S3 以進行長期儲存。您可以使用 AWS Glue，在 Amazon S3 中探索和準備日誌資料，以進行分析並將關聯的中繼資料儲存在 AWSAWS Glue Data Catalog。 Amazon AthenaAmazon Athena，透過與 AWS Glue 的原生整合，可用來分析日誌資料，並使用標準 SQL 進行查詢。使用 Amazon QuickSight 這類商業智慧工具來視覺化、探索和分析您的資料。

常用的反模式：

- 我們將新功能的一致交付視為關鍵績效指標。您無法測量部署發生的頻率。
- 您記錄部署、復原的部署、修補程式和復原的修補程式，以追蹤您的營運活動，但沒有人審查指標。
- 您的復原時間目標為可在 15 分鐘內還原遺失的資料庫，該目標設定於系統已部署且沒有使用者時。您現在有一萬名使用者，並已營運兩年。最近的還原時間花費超過兩小時。未記錄此項目，也沒有人知道。

建立此最佳實務的優勢： 透過收集和分析營運指標，您可以了解營運的運作狀態，並深入了解可能影響營運或達成業務成果的趨勢。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 收集和分析營運指標：定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。
  - 使用 Amazon CloudWatch 指標
  - Amazon CloudWatch 指標和維度參考
  - 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌

資源

相關文件：

- Amazon Athena
- Amazon CloudWatch 指標和維度參考
- Amazon QuickSight
- AWS Glue
- AWSAWS Glue Data Catalog
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- 使用 Amazon CloudWatch 指標

OPS09-BP04 建立營運指標基準

為指標建立基準，以提供期望值，做為比較和識別效能欠佳和過剩的營運活動的基礎。

常用的反模式：

- 您需告知預計部署的時間為何。您尚未測量部署所需的時間，也無法判斷預期的時間。
- 您需告知從應用程式伺服器問題中復原需要多長時間。對於從第一次聯絡客戶起計算的所需復原時間，您沒有相關資訊。對於從監控得知的第一次識別問題起計算的所需復原時間，您沒有相關資訊。
- 您需告知您在週末需要多少名支援人員。您不知道週末有多少個典型支援案例，且無法提供預估值。
- 您的復原時間目標為可在 15 分鐘內還原遺失的資料庫，該目標設定於系統已部署且沒有使用者時。您現在有一萬個使用者，並已營運兩年。對於資料庫還原時間為什麼變更的原因，您沒有相關資訊。

建立此最佳實務的優勢： 透過定義基準指標值，您可以評估目前的指標值和指標趨勢，以判斷是否需要採取動作。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 了解營運活動的預期模式：建立營運活動模式，以確定行為何時超出預期值，以便您可以在需要時做出適當的回應。

OPS09-BP05 了解營運活動的預期模式

建立營運活動模式以識別異常活動，以便您可以在必要時做出適當的回應。

常用的反模式：

- 最近，您的部署失敗率大幅增加。您獨立解決每次失敗。您不知道失敗源於不熟悉部署管理系統的新員工執行的部署。

建立此最佳實務的優勢： 透過學習行為模式，您可以識別意外行為，並在必要時採取動作。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 了解營運活動的預期模式：建立營運活動模式，以確定行為何時超出預期值，以便您可以在需要時做出適當的回應。

OPS09-BP06 在營運成果有風險時發出警示

每當營運成果有風險時，就必須發出警示並據以行動。營運成果是可支援生產中工作負載的任何活動。其中包含從部署新版應用程式到從中斷復原的所有作業。您必須以與業務成果一樣的重要性來看待營運成果。

軟體團隊應找出關鍵的營運指標和活動，並為其建立警示。警示必須及時且可據以採取行動。發出警示時，應包含相應執行手冊或程序手冊的參考。發出警示，但未提供相應的動作可能會導致警示疲勞。

預期成果： 當營運活動有風險時，就會傳送警示來促進行動。警示包含為何發出警示的背景資訊，並指向要調查的程序手冊和要採取緩解措施的執行手冊。盡可能自動化執行手冊並傳送通知。

常見的反模式：

- 您正在調查事件，以及正在將支援案例歸檔。支援案例違反服務水準協議 (SLA)，但未發出任何警示。
- 由於最後一刻的程式碼變更，預定於午夜進行的生產部署遭到延遲。未發出任何警示，而部署發生懸置。
- 發生生產中斷，但未傳送任何警示。
- 您的部署時間一直落後於預估值。未採取任何調查動作。

建立此最佳實務的優勢：

- 當營運成果有風險時，發出警示可以協助您透過預先發現問題來支援工作負載。
- 營運成果的運作狀態良好，業務成果因而獲得改善。

- 營運問題的偵測和修復也獲得改善。
- 整體營運運作狀態也有所改善。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

必須先定義營運成果，才能針對這些成果發出警示。透過定義哪些營運活動對貴組織最重要來開始。是否要在兩小時內將其部署至生產，或是在固定的時間內回應支援案例？ 貴組織必須定義關鍵營運活動，以及如何衡量這些活動，如此才能夠監控、改善這些活動，並據以發出警示。您需要一個中心位置，來存放和分析工作負載及營運遙測。相同的機制應能夠在營運成果有風險時發出警示。

客戶範例

CloudWatch 警示會在 AnyCompany Retail 的例行部署期間觸發。超過部署的前置時間。Amazon EventBridge 已在 AWS Systems Manager OpsCenter 中建立 OpsItem。雲端營運團隊使用程序手冊來調查問題，並發現結構描述的變更花費的時間比預期更長。他們向待命的開發人員發出警示，並持續監控部署。在部署完成後，雲端營運團隊就會解析 OpsItem。該團隊會在事後分析事件。

實作步驟

1. 如果您還沒有確定營運 KPI、指標和活動，請著手實作先前所述的此問題的最佳實務 (OPS09-BP01 至 OPS09-BP05)。
   - 使用 企業支援的 AWS Support 客戶 可以要求 營運 KPI 研討會 (透過其技術客戶經理)。此協作研討會可協助您定義與業務目標一致的營運 KPI 和指標，而不需額外費用。聯絡技術客戶經理來進一步了解。
2. 在您建立營運活動、KPI 和指標後，請在可觀察性平台設定警示。警示應具備與其關聯的動作，例如程序手冊或執行手冊。應避免發出不含動作的警示。
3. 經過一段時間後，您應能評估營運指標、KPI 和活動來找出待改善的地方。擷取執行手冊和程序手冊中來自操作人員的回饋，找出在回應警示時待改善的地方。
4. 警示應包含將待改善地方標示為誤判的機制。這會導致對指標閾值的審查。

實作計劃的工作量： 中。在實作此最佳實務前，必須實作幾個最佳實務。在確定營運活動與建立營運 KPI 後，也應建立警示。

資源

相關的最佳實務：

- OPS02-BP03 已為營運活動識別負責其效能的擁有者：每個營運活動和成果都應有確定的負責擁有者。當成果有風險時，該擁有者就應收到警示。
- OPS03-BP02 授權團隊成員在成果有風險時採取動作：發出警示時，團隊中應有專員採取行動來修復此問題。
- OPS09-BP01 識別關鍵績效指標：針對營運成果發出警示，從確定營運 KPI 開始。
- OPS09-BP02 定義營運指標：先建立此最佳實務，再開始產生警示。
- OPS09-BP03 收集和分析營運指標：您必須集中收集營運指標，才能建立警示。
- OPS09-BP04 建立營運指標基準：營運指標基準讓您能夠調整警示並避免警示疲勞。
- OPS09-BP05 了解營運活動的預期模式：您可以透過了解營運事件的活動模式，來改善警示的準確性。
- OPS09-BP08 驗證結果的實現以及 KPI 和指標的有效性：評估營運成果的達成情形，來確保 KPI 和指標是有效的。
- OPS10-BP02 每個提醒建立一個程序：每個警示應具備相關的執行手冊或程序手冊，並為收到警示的人員提供背景資訊。
- OPS11-BP02 執行事故後分析：在收到警示後執行事件後分析，來找出待改善的地方。

相關文件：

- AWS 部署管道參考架構：應用程式管道架構
- GitLab：開始使用敏捷 / DevOps 指標

相關影片：

- 使用 AWS Systems Manager OpsCenter 彙總和解決營運問題
- 將 AWS Systems Manager OpsCenter 與 Amazon CloudWatch 警示整合
- 使用 Amazon EventBridge 將資料來源整合至 AWS Systems Manager OpsCenter

相關範例：

- 使用 Amazon EC2 Systems Manager Automation 和 AWS Health 自動化 Amazon EC2 通知及其他方面的修復動作
- AWS 管理與管控工具研討會 - 2022 年營運
- 在 AWS 上使用 DevOps 監控儀表板來擷取、分析和視覺化指標

相關服務：

- Amazon EventBridge
- AWS Support 主動服務 - 營運 KPI 研討會
- AWS Systems Manager OpsCenter，
- CloudWatch 事件

OPS09-BP07 在偵測到營運異常時發出提醒

在偵測到營運異常時發出提醒，以便您可以在必要時做出適當的回應。

透過長時間分析營運指標能夠建立可充分量化的行為模式，以定義事件或發出警示來回應。

經過訓練後， CloudWatch 異常偵測 功能可用於對偵測到的異常發出 警示 ，或提供重疊的預期值至指標資料 圖形 上，以便進行持續比較。

Amazon DevOps Guru 可以用來透過事件關聯性、日誌分析和套用機器學習分析您的工作負載遙測來識別異常行為。AWS Well-Architected 洞見 會呈現出來，隨附相關資訊和建議。

常用的反模式：

- 您正將修補程式套用到您的執行個體機群。您已在測試環境中成功測試修補程式。對於機群中的大部分執行個體，修補程式失敗。您不採取任何動作。
- 您注意到，有部署動作從週五結束日開始。您的組織已預先定義星期二和星期四的維護時段。您不採取任何動作。

建立此最佳實務的優勢： 透過了解營運行為的模式，您可以識別意外行為，並在必要時採取動作。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 在偵測到營運異常時發出提醒：在偵測到營運異常時發出提醒，以便您可以在需要時做出適當的回應。
  - 什麼是 Amazon CloudWatch Events？
  - 建立 Amazon CloudWatch 警示
  - 使用 Amazon SNS 通知呼叫 Lambda 函數

資源

相關文件：

- Amazon DevOps Guru
- CloudWatch 異常偵測
- 建立 Amazon CloudWatch 警示
- 使用 Amazon CloudWatch Events 偵測管道狀態中的變更並做出反應
- 使用 Amazon SNS 通知呼叫 Lambda 函數
- 什麼是 Amazon CloudWatch Events？

OPS09-BP08 驗證結果的實現以及 KPI 和指標的有效性

建立營運活動的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。

AWS 還可透過 AWS 服務 API 和 SDK (例如 Grafana、Kibana 和 Logstash) 支援第三方日誌分析系統和商業智慧工具。

常用的反模式：

- 部署頻率已隨著開發團隊數量的成長而增加。您定義的預期部署數量為每週一次。您一直每天定期部署。當您的部署系統有問題，而無法部署時，數天都不會偵測到該問題。
- 當您的公司先前只在週一至週五的核心上班時間提供支援時。您已針對事件建立下個工作日回應時間目標。您最近開始提供全年無休支援涵蓋範圍，並隨附兩小時回應時間的目標。您的夜班員工不堪重負，客戶也不滿意。沒有事件回應時間發生問題的跡象，原因是您的通報違背下個工作日目標。

建立此最佳實務的優勢： 透過審查和修訂 KPI 和指標，您可以了解工作負載如何支援業務成果的達成，並找出達成業務目標需要改善的地方。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 驗證結果的實現以及 KPI 和指標的有效性：建立營運活動的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。

- 使用 Amazon CloudWatch 儀表板
- 什麼是日誌分析？

資源

相關文件：

- 使用 Amazon CloudWatch 儀表板
- 什麼是日誌分析？

OPS 10 您如何管理工作負載和營運事件？

準備和驗證回應事件的程序，大幅降低工作負載中斷情形。

最佳實務

- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP02 每個提醒建立一個程序
- OPS10-BP03 根據業務影響確定營運事件的優先順序
- OPS10-BP04 定義向上呈報路徑
- OPS10-BP05 啟用推送通知
- OPS10-BP06 透過儀表板傳達狀態
- OPS10-BP07 自動回應事件

OPS10-BP01 使用程序進行事件、事故和問題管理

您的組織具有處理事件、事故和問題的程序。事件 是發生於工作負載、但可能無需由您介入的事項。事故 是需要介入的事件。 問題 是重複發生而需要介入或無法解決的事件。您需要相關程序來減輕這些事件對業務的影響，並確保您能夠適當因應。

當工作負載發生事故和問題時，您需要有相關程序來加以處理。您如何讓利害關係人得知事件的狀態？應變由誰監控？ 您使用哪些工具來減輕事件的影響? 在此舉例說明一些您為了獲得可靠的應變程序而有待解答的問題。

程序必須集中記載，並且提供給涉及工作負載的每個人使用。如果您沒有集中的 Wiki 或文件存放區，可以使用版本控制儲存庫。您將隨著程序的演進而保有最新計劃。

問題是可以自動化的。這些事件佔據的時間會影響到您的創新能力。請開始建置可重複的程序，以減輕問題。經過一段時間後，您將著重於緩解措施的自動化或修正基礎問題。如此您即有時間投入於工作負載的改進。

預期成果： 您的組織具有處理事件、事故和問題的程序。這些程序會集中記載並存放。這些文件會隨著程序的變更而更新。

常見的反模式：

- 週末發生了事故，而值班工程師不知該如何處理。
- 客戶傳送電子郵件給您，指出應用程式已關閉。您將伺服器重新開機，試著修正問題。此狀況頻繁地發生。
- 有一項事故讓多個團隊各自獨立試著加以解決。
- 您的工作負載中發生了部署，但並未記錄。

建立此最佳實務的優勢：

- 您的工作負載中有事件的稽核軌跡。
- 您的事故中復原的時間減少了。
- 團隊成員可用一致的方式解決事故和問題。
- 調查事故的人力會更加整合。

未建立此最佳實務時的曝險等級： 高

實作指引

實作此最佳實務，意味著您會追蹤工作負載事件。您具有處理事故和問題的程序。這些程序會經常記載、共用及更新。問題經識別後會定出優先順序，然後獲得修正。

客戶範例

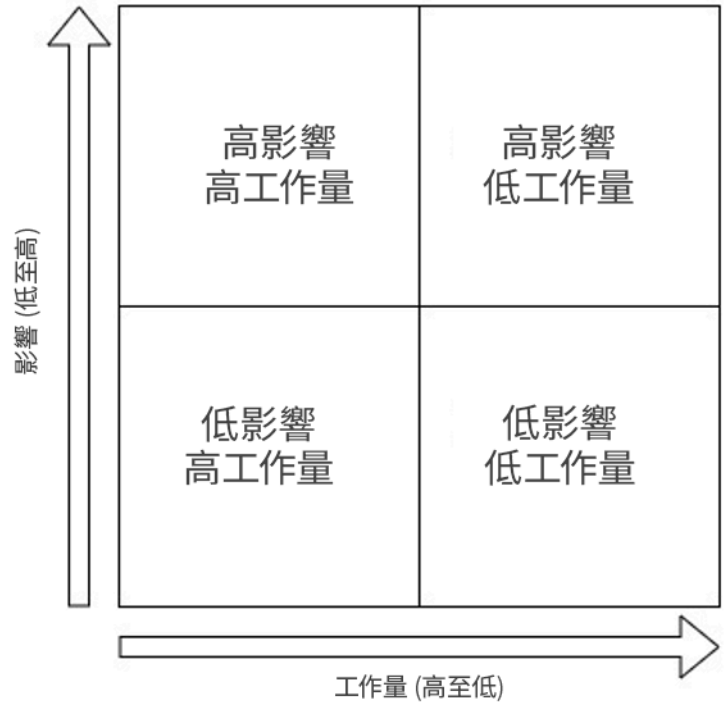
AnyCompany Retail 有某部分的內部 Wiki 專門用來處理事件、事故和問題管理。所有事件都會傳送至 Amazon EventBridge。問題會在 AWS Systems Manager OpsCenter 中識別為 OpsItems，並定出修正的優先順序，以減少無特殊專長人力。程序變更後，會隨即在其內部 Wiki 中更新。他們使用 AWS Systems Manager Incident Manager 來管理事故及協調緩解工作。

實作步驟

1. 事件

- 追蹤發生在工作負載中的事件，即使無需人為介入亦然。
- 與工作負載利害關係人共同擬定應追蹤的事件清單。範例包括已完成的部署或成功的修補。
- 您可以使用諸如 Amazon EventBridge 或者 Amazon Simple Notification Service 等服務來產生要追蹤的自訂事件。

2. 事故
   - 首先請定義事故的溝通計劃。哪些利害關係人必須獲得通知？ 您如何維繫其參與度? 協調工作由誰監控？ 我們建議建立內部交談管道，以利溝通和協調。
   - 為支援工作負載的團隊定義呈報路徑，尤其是團隊未設置當班輪值時。根據您的支援等級，您也可以向 AWS Support 申請立案。
   - 建立用來調查事件的程序手冊。其中應包含溝通計劃和詳細的調查步驟。在您的調查中納入對 AWS Health Dashboard 的檢查。
   - 記載您的事故應變計劃。傳達事故管理計劃，讓內部與外部客戶都了解互動的規則及其應有的預期。對您的團隊成員進行其使用訓練。
   - 客戶可使用 Incident Manager 來設定及管理其事故應變計劃。
   - Enterprise Support 客戶可以要求 事件管理研討會 (透過其技術客戶經理)。這個指導研討會將測試您現有的事故應變計劃，並協助您識別改善的領域。
3. 問題
   - 問題必須在 ITSM 系統中受到識別及追蹤。
   - 識別所有已知問題，並按照修正的工作量以及對工作負載的影響定出優先順序。



- 先解決高影響、低工作量的問題。這些問題解決後，再接著解決位於低影響、低工作量象限的問題。
- 您可以使用 Systems Manager OpsCenter 來識別這些問題、將執行手冊連結至問題，並加以追蹤。

實作計劃的工作量： 中。必須同時具備程序和工具，才能實作此最佳實務。記載您的程序，並且讓工作負載的任何相關人員都可加以使用。經常加以更新。您具有管理問題和加以緩解或修正的程序。

資源

相關的最佳實務：

- OPS07-BP03 使用執行手冊執行程序：已知問題需要相關聯的執行手冊，讓緩解工作保有一致性。
- OPS07-BP04 使用程序手冊來調查問題：事件需使用程序手冊來調查。
- OPS11-BP02 執行事故後分析：從事故復原後務必要執行事後檢討。

相關文件：

- Atlassian - DevOps 時代的事故管理
- AWS 安全事故應變指南
- DevOps 和 SRE 時代的事故管理

- PagerDuty - 什麼是事故管理？

相關影片：

- AWS re:Invent 2020：分散式組織中的事故管理
- AWS re:Invent 2021 - 使用事件驅動架構建置新一代的應用程式
- AWS 為您提供支援 | 探索事故管理桌上模擬演練
- AWS Systems Manager Incident Manager - AWS 虛擬研討會
- AWS 下一步 ft. Incident Manager | AWS 事件

相關範例：

- AWS 管理與管控工具研討會 - OpsCenter
- AWS 主動服務 – 事故管理研討會
- 使用 Amazon EventBridge 建置事件驅動應用程式
- 在 AWS 上建置事件驅動架構

相關服務：

- Amazon EventBridge
- Amazon SNS
- AWS Health Dashboard
- AWS Systems Manager Incident Manager
- AWS Systems Manager OpsCenter

OPS10-BP02 每個提醒建立一個程序

對於引發提醒的任何事件，建立明確定義的回應 (執行手冊或程序手冊)，並指明。此舉可確保對營運事件的有效而迅速的回應，並防止需採取動作的事件被無價值的通知所淹沒。

常用的反模式：

- 您的監控系統會將核准的連線串流以及其他訊息一起提供給您。訊息數量如此龐大，以至於您錯過需要您介入的定期錯誤訊息。

- 您收到提醒，指出網站運作中斷。發生這種情況時沒有已定義的程序。您必須採取臨機操作方法來診斷和解決問題。隨需開發此程序會延長復原時間。

建立此最佳實務的優勢： 只有在需要採取動作時才發出提醒，可防止低值提醒隱藏高值提醒。透過讓每個可採取動作的提醒都具有一個程序，您可針對環境中的事件實現一致且迅速的回應。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 每個提醒建立一個程序：對於引發提醒的任何事件，都應建立明確定義的回應 (執行手冊或程序手冊)，並指明負責人 (例如，個人、團隊或角色) 來對成功完成的程序負責。回應的執行可以自動化，也可以由另一個團隊完成，但負責人要對確保流程交付預期結果負責。透過建立這些程序，您可以確保對營運事件做出迅速有效的回應，並防止需採取行動的事件被無價值的通知所淹沒。例如，自動調整規模功能可能應用於調整 Web 前端規模，但營運團隊可能需負責確保自動調整規模規則和限制符合工作負載需求。

資源

相關文件：

- Amazon CloudWatch 功能
- 什麼是 Amazon CloudWatch Events？

相關影片：

- 制定監控計劃

OPS10-BP03 根據業務影響確定營運事件的優先順序

確保在有多個事件需要介入時，首先解決對業務最重要的事件。影響可能包括人員傷亡、經濟損失或聲譽或信用受損。

常用的反模式：

- 您收到為使用者新增印表機組態的支援請求。處理此問題時，您收到支援請求，而其指出您的零售網站運作中斷。為使用者完成印表機組態後，您便開始處理網站問題。
- 您收到零售網站和薪資系統運作中斷的通知。您不知道應該優先處理哪一個。

建立此最佳實務的優勢： 將對業務影響最大的事件回應排定優先順序，讓您能夠管理該影響。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 根據業務影響，排定操作事件的優先順序：確保在有多個事件需要介入時，首先解決對業務最重要的事件。影響可能包括人員傷亡、經濟損失、違反法規或聲譽或信用受損。

OPS10-BP04 定義向上呈報路徑

在您的執行手冊和程序手冊中定義向上呈報路徑，包括觸發向上呈報的條件以及向上呈報的程序。明確確定每個動作的擁有者，以確保對營運事件做出迅速有效的回應。

在採取行動之前，確定何時需要人為決策。與決策者合作，事先做出該決策，並預先核准行動，如此就不會延長 MTTR 等待回應的時間。

常用的反模式：

- 您的零售網站已運作中斷。您不了解用於恢復網站的執行手冊。您開始打電話給同事，希望有人能夠幫助您。
- 您收到應用程式無法連線的支援案例。您沒有管理系統的許可。您不知道誰有這個許可。您嘗試聯絡開立此案例的系統擁有者，但沒有回應。您沒有此系統的聯絡人，而且您的同事對此不熟悉。

建立此最佳實務的優勢： 透過定義向上呈報、向上呈報觸發條件和向上呈報程序，您可以針對影響以適當的速率將資源系統性地新增到事件中。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 定義向上呈報路徑：在您的執行手冊和程序手冊中定義向上呈報路徑，包括觸發向上呈報的條件以及向上呈報的程序。例如，當執行手冊無法解決問題或經過預定時間，將問題從支援工程師向上呈報給資深支援工程師。適當的向上呈報途徑還有，當程序手冊無法確定工作負載的補救途徑或經過預定時間，從高級支援工程師向上呈報給開發團隊。明確確定每個動作的擁有者，以確保對營運事件做出迅速有效的回應。向上呈報可以包括第三方。例如，網路連接提供商或軟體供應商。向上呈報可以包括受影響系統的指定授權決策者。

OPS10-BP05 啟用推送通知

就您的使用者所用之服務受到影響以及服務再次恢復正常，直接與使用者溝通 (例如，透過電子郵件或簡訊)，以便使用者能夠採取適當動作。

常用的反模式：

- 您的應用程式正遭遇分散式阻斷服務事故，且已數天沒有回應。沒有錯誤訊息。您尚未傳送通知電子郵件。您尚未傳送文字通知。您尚未在社交媒體上分享資訊。您的客戶感到沮喪，正在尋找能夠支援他們的其他廠商。
- 週一，您的應用程式在進行修補之後發生問題，並中斷運作了幾個小時。週二，您的應用程式在程式碼部署之後發生問題，且效能不可靠的情況持續數小時。週三，您的應用程式在進行程式碼部署 (以減輕與失敗修補相關之安全性弱點的影響) 後出現問題，且無法使用的情況持續數小時。週四，沮喪的客戶開始尋找可以支援他們的其他廠商。
- 這個週末您的應用程式將會中斷運作以進行維護。您沒有通知客戶。您的部分客戶已排定會用到您應用程式的活動。他們在發現您的應用程式無法使用時感到非常沮喪。

建立此最佳實務的優勢： 透過定義通知、通知觸發條件和通知程序，讓您的客戶可以在工作負載的問題影響他們時，收到通知和回應。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 啟用推送通知：就服務受到影響以及服務恢復正常，直接與您的使用者溝通 (例如，透過電子郵件或 SMS)，以便使用者能夠採取適當措施。
  - Amazon SES 功能
  - 什麼是 Amazon SES？
  - 設定 Amazon SNS 通知

資源

相關文件：

- Amazon SES 功能
- 設定 Amazon SNS 通知
- 什麼是 Amazon SES？

OPS10-BP06 透過儀表板傳達狀態

提供針對其目標受眾 (例如，內部技術團隊、領導和客戶) 量身定制的儀表板，以傳達業務的當前營運狀態，並提供感興趣的指標。

您可以使用 Amazon CloudWatch 儀表板 建立儀表板，它位於 CloudWatch 主控台中的自訂首頁上。您可以使用 Amazon QuickSight 這類商業智慧服務，建立和發佈工作負載和營運運作狀態 (例如，下單率、連線的使用者和交易時間) 的互動式儀表板。建立儀表板，以顯示指標的系統和業務等級檢視。

常用的反模式：

- 根據要求，您執行應用程式目前使用率的報告來進行管理。
- 在事故期間，相關系統擁有者每 20 分鐘就會聯絡您一次，想知道問題是否已修正。

建立此最佳實務的優勢： 透過建立儀表板，您可以自助存取資訊，讓您的客戶能夠自行獲得相關資訊並自行判斷是否需要採取動作。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 透過儀表板溝通狀態：提供針對其目標受眾 (例如，內部技術團隊、領導階層和客戶) 量身定制的儀表板，以傳達企業的當前營運狀況，並提供感興趣的指標。提供自助獲取狀態資訊選項，減少因回應營運團隊狀態請求而造成的干擾。範例包括 Amazon CloudWatch 儀表板和 AWS Health Dashboard。
  - CloudWatch 儀表板建立和使用自訂指標檢視

資源

相關文件：

- Amazon QuickSight
- CloudWatch 儀表板建立和使用自訂指標檢視

OPS10-BP07 自動回應事件

自動對事件進行回應，以減少由手動程序引起的錯誤，並確保快速一致的回應。

有多種方式可以在 AWS 上將執行手冊和程序手冊動作自動化。若要回應來自 AWS 資源狀態變更的事件，或您自己的自訂事件，您應建立 CloudWatch Events 規則 透過 CloudWatch 目標觸發回應 (例

如，Lambda 函數、Amazon Simple Notification Service (Amazon SNS) 主題、Amazon ECS 任務，以及 AWS Systems Manager Automation)。

要回應超過資源臨界值的指標 (例如，等待時間)，您應使用建立 CloudWatch 警示， 來執行一個或多個動作，方法為使用 Amazon EC2 動作、Auto Scaling 動作，或將通知傳送至 Amazon SNS 主題。如果您需要執行自訂動作來回應警示，則請透過 Amazon SNS 通知叫用 Lambda。使用 Amazon SNS 發佈事件通知和向上呈報訊息，以使人們了解情況。

AWS 還可透過 AWS 服務 API 和 SDK 支援第三方系統。AWS 合作夥伴和第三方提供了許多監控工具，可用於監控、通知和回應。其中一些工具包含 New Relic、Splunk、Loggly、SumoLogic 和 Datadog。

當自動化程序失敗時，您應保留重要的手動程序以供使用

常用的反模式：

- 開發人員檢查其程式碼。此事件原本可能用於啟動建置，然後執行測試，不過沒有發生任何情況。
- 您的應用程式會在停止運作之前記錄特定錯誤。您應非常了解重新啟動應用程式的程序，且可以編寫此程序的指令碼。您可以使用日誌事件來叫用指令碼，並重新啟動應用程式。相反地，當星期日凌晨 3 點發生錯誤時，您做為負責修正系統的待命資源將被喚醒。

建立此最佳實務的優勢： 透過對事件使用自動回應，您可以縮短回應時間，並限制手動活動引入錯誤。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 將事件的回應自動化：自動對事件進行回應，以減少由手動流程引起的錯誤，並確保快速、一致的回應。
  - 什麼是 Amazon CloudWatch Events？
  - 建立隨事件觸發的 CloudWatch Events 規則
  - 使用 AWS CloudTrail 建立隨 AWS API 呼叫觸發的 CloudWatch Events 規則
  - 來自所支援服務的 CloudWatch Events 事件範例

資源

相關文件：

- Amazon CloudWatch 功能
- 來自所支援服務的 CloudWatch Events 事件範例
- 使用 AWS CloudTrail 建立隨 AWS API 呼叫觸發的 CloudWatch Events 規則
- 建立隨事件觸發的 CloudWatch Events 規則
- 什麼是 Amazon CloudWatch Events？

相關影片：

- 制定監控計劃

相關範例：

演進

問題

- OPS 11 您如何改善營運？

OPS 11 您如何改善營運？

投入時間和資源持續逐漸改善，以加強營運的效果和效率。

最佳實務

- OPS11-BP01 建立持續改進程序
- OPS11-BP02 執行事故後分析
- OPS11-BP03 實作回饋迴圈
- OPS11-BP04 知識管理
- OPS11-BP05 定義改進驅動因素
- OPS11-BP06 驗證洞見
- OPS11-BP07 執行營運指標審查
- OPS11-BP08 記錄和分享獲得的經驗
- OPS11-BP09 分配改進時間

OPS11-BP01 建立持續改進程序

定期評估改進機會並排定其優先順序，以專注於它們可在其中提供最大效益的工作。

常用的反模式：

- 您已記錄建立開發或測試環境所需的程序。您可以使用 CloudFormation 將該程序自動化，但您卻從主控台手動執行程序。
- 您的測試顯示，應用程式內絕大多數的 CPU 使用率都屬於一小組缺乏效率的功能。您可以專注於改進這些功能並降低成本，但您的任務是建立新的可用性功能。

建立此最佳實務的優勢： 透過持續改進提供一種機制，以定期評估改進機會、排定機會的優先順序，並專注於它們可在其中提供最大效益的工作。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 定義持續改進程序：定期評估改進機會並確定優先級，以將精力集中在可以帶來最大收益的機會上。實作變更以改善結果，並進行評估以確定成功與否。如果結果未能達到目標，並且改進仍然是優先事項，則使用其他行動方案重複進行。您的營運流程應設立專門的時間和資源，用於持續逐漸改善。

OPS11-BP02 執行事故後分析

審查影響客戶的事件，並識別造成問題的因素和預防性措施。使用此資訊來開發緩解措施，以限制或防止事件再次發生。制定可快速有效回應的程序。適當地傳達成因和為目標受眾量身打造的糾正措施。

常用的反模式：

- 您管理應用程式伺服器。大約每 23 小時 55 分鐘，所有作用中工作階段都會終止。您已嘗試識別應用程式伺服器上發生了什麼問題。您懷疑這反而可能是網路問題，但無法與網路團隊合作，因為他們太忙而無法為您提供支援。您缺少可遵循的預先定義程序來取得支援與收集必要資訊，以判斷發生的情況。
- 您的工作負載內發生資料遺失問題。這是第一次發生，原因尚不確定。您確定它並不重要，因為您可以重新建立資料。資料遺失以影響客戶的較高頻率開始發生。當您還原遺失的資料時，這也會為您帶來額外的操作負擔。

建立此最佳實務的優勢： 透過預先定義的程序來判斷造成事件的元件、條件、動作和事件，讓您能夠找出改進機會。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 使用程序判斷成因：審查所有影響客戶的事故。建立程序來識別和記錄事件的成因，以便您可以制定緩解措施來限制或防止事件再次發生。另外，您還可以制定快速有效地做出回應的程序。根據目標受眾的不同以適當的方式告知根本原因。

OPS11-BP03 實作回饋迴圈

回饋迴圈提供可推動決策的可行洞察。在程序和工作負載中建立回饋迴圈。此可協助您找出問題和需要改善的地方。回饋迴圈也會驗證在改善中所做的投資。這些回饋迴圈是持續改善工作負載的基礎。

回饋迴圈分為兩種： 即時回饋 和 追溯性分析。透過審查營運活動的績效和成果來收集即時的回饋。此回饋來自團隊成員、客戶或活動的自動化輸出。接收 A/B 測試和交付新功能等方面的即時回饋，對於快速檢錯非常重要。

定期進行追溯性分析，以從對營運成果和指標的審查中獲取回饋。這些追溯性分析會在衝刺結束，按規律或在主要版本或事件後發生。這類回饋迴圈會驗證對營運或工作負載所做的投資。其可協助您衡量成功並驗證策略。

預期成果： 您使用即時回饋和追溯性分析來推動改善。存在可擷取使用者和團隊成員回饋的機制。追溯性分析會用來找出可推動改善的趨勢。

常見的反模式：

- 您推出新功能，但沒有辦法收到客戶對該功能的回饋。
- 針對營運改善投入資源和時間後，您無法執行追溯性分析來進行驗證。
- 您收集客戶的回饋，但未能定期審查回饋。
- 回饋迴圈讓我們得以提議行動項目，但軟體開發程序中未納入這些項目。
- 客戶沒有收到他們提議之改善的回饋。

建立此最佳實務的優勢：

- 您可以反過來與客戶合作來推動新功能。
- 您的組織文化可以更快速地應對變化。
- 趨勢會用來找出改善的機會。
- 追溯性分析可驗證對工作負載和營運所做的投資。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

實作此最佳實務表示您同時使用即時回饋和追溯性分析。這些回饋迴圈可推動改善。有許多機制可用來處理即時回饋，包含調查、客戶投票和回饋表單。組織也會使用追溯性分析來找出改善的機會並驗證計劃。

客戶範例

AnyCompany Retail 建立網頁表單，客戶可在其中提供回饋或回報問題。在每週 Scrum 期間，軟體開發團隊會評估使用者回饋。該團隊會定期使用回饋來為其平台的發展釐清方向。他們會在每次衝刺結束時執行追溯性分析，來找出他們想要改善的項目。

實作步驟

1. 即時回饋
   - 您需要制定機制來接收來自客戶和團隊成員的回饋。您也可以設定營運活動來提供自動化的回饋。
   - 組織需要制定程序來審查此回饋、判斷需要改善的項目，並安排改善項目。
   - 您必須將回饋新增至軟體開發程序。
   - 在您著手改善後，請與回饋提交者追蹤後續進展。
     - 您可以使用 AWS Systems Manager OpsCenter， 以 OpsItems 的形式 建立和追蹤這些改善。
2. 追溯性分析
   - 在開發週期結束時，以固定的規律或在主要版本之後，執行追溯性分析。
   - 召集工作負載中參與的利害關係人，進行回顧會議。
   - 在白板或試算表建立三個欄位：停止、開始和持續。
     - 停止 是您希望團隊停止做的任何事。
     - 開始 是您希望開始執行的想法。
     - 持續 是您希望持續執行的項目。
   - 詢問在場人士的想法，收集利害關係人的回饋。
   - 排列回饋的優先順序。將動作和利害關係人指派至任何「開始」或「持續」項目。
   - 將動作新增至軟體開發程序中，並在您執行改善項目時向利害關係人告知最新的狀態。

實作計劃的工作量： 中。若要實作此最佳實務，您需要找到方法來擷取即時回饋並進行分析。此外，您需要建立追溯性分析程序。

資源

相關的最佳實務：

- OPS01-BP01 評估外部客戶需求：回饋迴圈是一種機制，可收集外部客戶的需求。
- OPS01-BP02 評估內部客戶需求：內部利害關係人可以使用回饋迴圈來表達需要和需求。
- OPS11-BP02 執行事故後分析：事件後分析是在事件後執行的追溯性分析的一種重要形式。
- OPS11-BP07 執行營運指標審查：營運指標審查會找出趨勢和待改善的地方。

相關文件：

- 建置 CCOE 時應避開的 7 大陷阱
- Atlassian 團隊程序手冊 - 追溯性
- 電子郵件定義：回饋迴圈
- 根據 AWS Well-Architected Framework 審查建立回饋迴圈
- IBM Garage Methodology - 進行回顧
- Investopedia – PDCS 週期
- 最大化開發人員的效能 (作者：Tim Cochran)
- 營運準備度審查 (ORR) 白皮書 - 反覆執行
- TIL CSI - 持續服務改善
- 當 Toyota 遇見電子商務：Amazon 的精實原則

相關影片：

- 建立有效的客戶回饋迴圈

相關範例：

- Astuto - 開放原始碼客戶回饋工具
- AWS 解決方案 - AWS 上的 QnABot
- Fider - 整理客戶回饋的平台

相關服務：

- AWS Systems Manager OpsCenter

OPS11-BP04 知識管理

存在的機制讓您的團隊成員可以及時探索他們所需的資訊、存取資訊，並識別其是否為最新且完整的資訊。存在的機制是用來識別所需的內容、需要重新整理的內容，以及應存檔的內容，以便該內容不再供其他人參考。

常用的反模式：

- 一個感到沮喪的客戶開立了新產品功能請求的支援案例，以處理感知的問題。它會新增到優先改進項目的清單中。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 知識管理：確保存在的機制讓您的團隊成員可以及時探索他們所需的資訊、存取資訊，並識別其是否為最新且完整的資訊。維護機制，以識別所需的內容、需要重新整理的內容，以及應存檔的內容，以便該內容不再供其他人參考。

OPS11-BP05 定義改進驅動因素

確定改進驅動因素，以幫助您評估改進機會並排定其優先順序。

在 AWS 上，您可以彙整所有營運活動、工作負載和基礎設施的日誌，以建立詳細的活動歷史記錄。然後，您可以使用 AWS 工具，分析某段時間內的營運和工作負載運作狀態 (例如，識別趨勢、將事件和活動與成果關聯，以及在環境間和跨系統進行比較和對比)，根據驅動因素來發現改善機會。

您應使用 CloudTrail 追蹤 API 活動 (透過 AWS Management Console、CLI、SDK 和 API)，以了解整個帳戶中發生的情況。使用 CloudTrail 和 CloudWatch 追蹤您的 AWS 開發人員工具部署活動。這樣會將部署及其成果的詳細活動歷史記錄新增至 CloudWatch Logs 日誌資料中。

將日誌資料匯出至 Amazon S3 以進行長期儲存。您可以使用 AWS Glue，您可以探索和準備 Amazon S3 中的日誌資料以進行分析。使用 Amazon Athena，透過與 AWS Glue 原生整合來分析日誌資料。使用 Amazon QuickSight 這類商業智慧工具來視覺化、探索和分析您的資料

常用的反模式：

- 您有一個可運作但不巧妙的指令碼。您投入時間來重新撰寫它。現在該指令碼相當出色。

- 您的新創公司正嘗試從創投家獲得另一批資金。他們希望您證明 PCI DSS 的合規性。您想要讓客戶滿意，因此您以文件記錄合規情況，但錯過提供給客戶的交付日期，而失去該客戶。做這件事沒有錯，但現在您不知道這是否是對的。

建立此最佳實務的優勢： 藉由決定您想要用於改進的條件，您可以將事件型動機或情緒投資的影響降到最低。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 了解改進驅動因素：僅在支援理想結果時才對系統進行變更。
  - 所需能力：在評估改進機會時，評估所需的功能和能力。
    - AWS 最新消息
  - 不可接受的問題：在評估改進機會時，評估不可接受的問題、錯誤和弱點。
    - AWS 最新安全公告
    - AWS Trusted Advisor
  - 合規要求：在審查改進機會時，評估保持法規、政策的遵從性或保持受到第三方支援所需的更新和變更。
    - AWS 合規
    - AWS 合規計劃
    - AWS 合規最新資訊

資源

相關文件：

- Amazon Athena
- Amazon QuickSight
- AWS 合規
- AWS 合規最新資訊
- AWS 合規計劃
- AWS Glue
- AWS 最新安全公告
- AWS Trusted Advisor

- 將日誌資料匯出至 Amazon S3
- AWS 最新消息

OPS11-BP06 驗證洞見

與跨職能團隊和企業擁有者一起審查您的分析結果和回應。透過這些審查建立共識，確定其他影響並確定行動方案。適當調整回應。

常用的反模式：

- 您看到系統上的 CPU 使用率為 95%，並優先找出降低系統負載的方法。您確定最佳行動方案是擴充。該系統是轉碼器，系統會擴展到一直以 95% 的 CPU 使用率執行。系統擁有者可能向您解釋情況，並讓您聯絡他們。您的時間被浪費了。
- 系統擁有者堅稱系統是任務關鍵性系統。系統未放置在高安全性的環境中。為改善安全性，您實作任務關鍵性系統所需的額外偵測和預防性控制措施。您通知系統擁有者工作已完成，且其需為其他資源支付相應費用。在此通知之後的討論中，系統擁有者了解了其系統不符合的任務關鍵系統的正式定義。

建立此最佳實務的優勢： 透過與企業擁有者和領域專家驗證洞見，您可以建立共識並更有效地引導改進。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 驗證洞見：與企業擁有者和領域專家互動，確保您收集資料的意義得到眾人理解和同意。識別其他疑慮、潛在影響，並確定行動方案。

OPS11-BP07 執行營運指標審查

與來自不同業務領域的跨團隊參與者定期進行營運指標的追溯性分析。透過這些審查確定改進機會、可能的行動方案並分享獲得的經驗。

尋找所有環境 (例如開發、測試和生產) 中的改善機會。

常用的反模式：

- 您的維護時段中斷了重要的零售促銷。如果還有其他影響企業的事件，企業仍然不知道是否有可能會延遲的標準維護時段。

- 您使用組織中常用的錯誤程式庫，因此您經歷了長時間的中斷。之後您已遷移到可靠的程式庫。組織中的其他團隊不知道他們正面臨風險。如果你們定期會面並審查此事故，他們就會注意風險。
- 轉碼器的效能一直在穩定地下降，並影響媒體團隊。這還不是很令人震驚。除非該情況嚴重到足以造成事故，否則您將無法查明該情況。如果您與媒體團隊審查營運指標，則有機會識別指標及其體驗的變更並解決問題。
- 您沒有審查對客戶 SLA 的滿意度。您有不符合客戶 SLA 的趨勢。不符合客戶 SLA，會產生相關的財務處罰。如果你們定期會面，並審查這些 SLA 的指標，您將有機會識別並解決問題。

建立此最佳實務的優勢： 透過定期會議以審查營運指標、事件和事故，您可以在團隊間維持共識、分享獲得的經驗，並可以排定改進項目的優先順序並鎖定改進目標。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 營運指標審查：與來自不同業務領域的跨團隊參與者定期進行營運指標的追溯性分析。與包括業務、開發和營運團隊在內的利害關係人進行互動，以驗證您從即時回饋和追溯性分析獲得的發現，並分享經驗教訓。利用這些洞見確定改進機會和可能的行動方案。
  - Amazon CloudWatch
  - 使用 Amazon CloudWatch 指標
  - 發佈自訂指標
  - Amazon CloudWatch 指標和維度參考

資源

相關文件：

- Amazon CloudWatch
- Amazon CloudWatch 指標和維度參考
- 發佈自訂指標
- 使用 Amazon CloudWatch 指標

OPS11-BP08 記錄和分享獲得的經驗

記錄並分享從營運活動中獲得的經驗，以便您可以在內部以及跨團隊使用它們。

您應分享您的團隊獲得的經驗，以提高整個組織的效益。您希望分享資訊和資源，以防止可避免的錯誤並簡化開發工作。如此可讓您聚焦於提供所需的功能。

使用 AWS Identity and Access Management (IAM) 定義權限，從而實現對您希望在帳戶內及帳戶間分享的資源的受控存取。然後，您使用版本控制的 AWS CodeCommit 儲存器來分享應用程式程式庫、執行指令碼的程序、程序文件及其他系統文件。透過分享對 AMI 的存取以及授權跨帳戶使用 Lambda 函數，進而分享您的運算標準。您還應將基礎設施標準作為 AWS CloudFormation 範本進行分享。

透過 AWS API 和 SDK，您可以整合外部和第三方工具及儲存器 (例如 GitHub、BitBucket 和 SourceForge)。分享您獲得的經驗和開發的知識時，請小心建構權限，以確保分享的儲存器的完整性。

常用的反模式：

- 您使用組織中常用的錯誤程式庫，因此您經歷了長時間的中斷。之後您已遷移到可靠的程式庫。組織中的其他團隊不知道他們正面臨風險。如果您在此程式庫中記錄和分享您的經驗，他們會注意風險。
- 您已在內部共用的微型服務中找出導致工作階段終止的邊緣案例。您已更新對服務的呼叫，以避免此邊緣案例。組織中的其他團隊不知道他們正面臨風險。如果您在此程式庫中記錄和分享您的經驗，他們會注意風險。
- 您已找到一個方法，可大幅降低其中一個微型服務所需的 CPU 使用率。您不知道是否有任何其他團隊可以利用此技術。如果您在此程式庫中記錄和分享您的經驗，其他團隊將有機會這樣做。

建立此最佳實務的優勢： 分享獲得的經驗以協助改進並將經驗的好處發揮到最大。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 記錄和分享獲得的經驗：制定程序來記錄從執行營運活動和追溯性分析中學到的經驗教訓，以便其他團隊可以使用。
  - 分享經驗：制定程序來在團隊之間分享經驗教訓和相關成品。例如，透過可存取的 Wiki 分享更新的程序、指引、管控和最佳實務。透過公共儲存庫共用指令碼、程式碼和程式庫。
    - 委託存取您的 AWS 環境
    - 共用 AWS CodeCommit 儲存庫
    - 輕鬆授權 AWS Lambda 函數
    - 與特定 AWS 帳戶共用 AMI
    - 使用 AWS CloudFormation Designer URL 加速範本共用

- 搭配 Amazon SNS 使用 AWS Lambda

資源

相關文件：

- 輕鬆授權 AWS Lambda 函數
- 共用 AWS CodeCommit 儲存庫
- 與特定 AWS 帳戶共用 AMI
- 使用 AWS CloudFormation Designer URL 加速範本共用
- 搭配 Amazon SNS 使用 AWS Lambda

相關影片：

- 委託存取您的 AWS 環境

OPS11-BP09 分配改進時間

在流程中投入時間和資源，以持續逐漸改善。

在 AWS 上，您可以建立臨時環境複本，從而降低試驗和測試的風險、工作量及成本。這些重複的環境可用於測試從您的分析、試驗和開發得出的結論，以及測試計劃的改善。

常用的反模式：

- 您的應用程式伺服器存在已知的效能問題。它會新增到每個計劃功能實作的待辦項目中。如果計劃功能的新增速率保持不變，則效能問題永遠不會解決。
- 為協助持續改進，您核准管理員和開發人員使用他們額外的時間來選取和實作改進項目。進改永遠不會有完成的一天。

建立此最佳實務的優勢： 透過在程序中投入時間和資源，您可以實現持續逐漸改善。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 分配改進時間：在流程中投入時間和資源，以持續逐漸改善。實作變更以改進和評估結果，從而確定成功與否。如果結果未能達到目標，並且改進仍然是優先事項，則應採取替代行動方案。

安全性

主題

安全基礎

問題

SEC 1 如何安全地操作工作負載？

若要安全地操作工作負載，您必須將總體最佳實務套用到每個安全領域。採用您在組織和工作負載層級所定義的卓越營運要求和程序，將這些要求和程序套用到所有領域。透過 AWS 和產業建議與威脅情報持續取得最新資訊，可協助您發展威脅模型和控制目標。自動化安全程序、測試和驗證可讓您擴展安全操作。

最佳實務

SEC01-BP01 使用帳戶區隔工作負載

先從安全與基礎設施開始，讓您的組織隨著工作負載的成長設定常見的防護機制。該方法提供工作負載之間的邊界和控制。強烈建議進行帳戶層級的區隔，以便將生產工作負載與開發和測試工作負載隔離，或在依照外部合規要求 (例如 PCI-DSS 或 HIPAA) 所定義之不同敏感性等級處理資料的工作負載，與未如此做的工作負載之間提供強大的邏輯邊界。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 使用 AWS Organizations：使用 AWS Organizations 為多個 AWS 帳戶集中強制執行策略型管理。
  - AWS Organizations 入門
  - 如何使用服務控制政策在 AWS Organizations 中的各個帳戶之間設定許可防護機制
- 考慮 AWS Control Tower：AWS Control Tower 提供一種便利方式，根據最佳實務設定和管控全新、安全的多帳戶 AWS 環境。
  - AWS Control Tower

資源

相關文件：

- IAM 最佳實務
- 安全公告
- AWS 安全稽核指導方針

相關影片：

- 使用 AWS Organizations 管理多帳戶 AWS 環境
- 以 Well-Architected 方式提供安全最佳實務
- 使用 AWS Control Tower 管控多帳戶 AWS 環境

SEC01-BP02 保護 AWS 帳戶

在多個層面保護 AWS 帳戶，包括保護 (而非使用 根使用者)，以及使您的聯絡資訊保持在最新狀態。您可以使用 AWS Organizations ，在 AWS 中的工作負載增長和擴展時，集中管理和管控您的帳戶。AWS Organizations 可協助您跨帳戶管理帳戶、設定控制及設定服務。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 使用 AWS Organizations：使用 AWS Organizations 為多個 AWS 帳戶集中強制執行策略型管理。
  - AWS Organizations 入門
  - 如何使用服務控制政策在 AWS Organizations 中的各個帳戶之間設定許可防護機制
- 限制 AWS 根使用者的使用：僅使用根使用者來執行特別需要的任務。
  - 需要 AWS 帳戶根使用者憑證的 AWS 任務
- 針對根使用者啟用多重因素認證 (MFA)：如果 AWS Organizations 未為您管理根使用者，請在 AWS 帳戶 根使用者上啟用 MFA。
  - 根使用者
- 定期變更根使用者密碼：變更根使用者密碼可降低使用已儲存密碼的風險。如果您沒有使用 AWS Organizations，而且任何人都有實體存取權，則尤為重要。
  - 變更 AWS 帳戶根使用者密碼
- 使用 AWS 帳戶根使用者時發出通知：收到通知會自動降低風險。
  - 如何在使用 AWS 帳戶的根存取金鑰時接收通知
- 限制對新增區域的存取：對於新的 AWS 區域，IAM 資源 (例如使用者和角色) 只會傳播到您啟用的區域。
  - 設定權限以為即將推出的 AWS 區域啟用帳戶
- 考慮 AWS CloudFormation StackSets：CloudFormation StackSets 可用於將資源 (包括 IAM 政策、角色和群組) 從核可的範本部署到不同的 AWS 帳戶和區域中。
  - 使用 CloudFormation StackSets

資源

相關文件：

- AWS Control Tower
- AWS 安全稽核指導方針
- IAM 最佳實務
- 安全公告

相關影片：

- 透過自動化和管控大規模採用 AWS
- 以 Well-Architected 方式提供安全最佳實務

相關範例：

- 實驗室：AWS 帳戶 和根使用者

SEC01-BP03 識別和驗證控制目標

根據合規需求以及從威脅模型識別的風險，衍生並驗證您需要套用到工作負載的控制目標和控制。對控制目標與控制持續進行驗證，可協助您測量風險降低的有效性。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 識別合規要求：發現您的工作負載務必遵守的組織、法律和合規要求。
- 識別 AWS 合規資源：識別 AWS 可用於協助您達成合規的資源。
  - https://aws.amazon.com/compliance/
  - https://aws.amazon.com/artifact/

資源

相關文件：

- AWS 安全稽核指導方針
- 安全公告

相關影片：

- AWS Security Hub：管理安全提醒與使合規自動化
- 以 Well-Architected 方式提供安全最佳實務

SEC01-BP04 掌握安全威脅的最新資訊

透過隨時得知最新安全威脅來辨識攻擊向量，以協助您定義並實作適當的控制。取用 AWS Managed Services 可讓您更輕鬆地接收 AWS 帳戶中非預期或異常行為的通知。使用 AWS 合作夥伴工具或第三

方威脅資訊摘要，做為安全資訊流程的一部分進行調查。AWS Well-Architected 通用漏洞披露 (CVE) 清單 此清單包含公開揭露的網路安全漏洞，您可以使用這些漏洞來保持最新狀態。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 訂閱威脅情報來源：定期從多個來源檢閱與工作負載中使用的技術相關的威脅情報。
  - 通用漏洞披露清單
- 考慮 AWS Shield Advanced 服務：如果您的工作負載可透過網際網路存取，它可提供近乎即時的情報來源可見性。

資源

相關文件：

- AWS 安全稽核指導方針
- AWS Shield
- 安全公告

相關影片：

- 以 Well-Architected 方式提供安全最佳實務

SEC01-BP05 及時了解安全建議的最新資訊

隨時取得 AWS 和產業安全建議的最新資訊，以發展工作負載的安全狀態。AWS 安全公告 包含有關安全和隱私權通知的重要資訊。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 關注 AWS 更新：訂閱或定期查看新的建議、秘訣和技巧。
  - AWS Well-Architected 實驗室
  - AWS 安全部落格
  - AWS 服務文件
- 訂閱產業新聞：定期在多個來源檢閱與工作負載中使用技術相關的新聞摘要。

- 範例：通用漏洞披露清單

資源

相關文件：

- 安全公告

相關影片：

- 以 Well-Architected 方式提供安全最佳實務

SEC01-BP06 將管道中安全控制的測試和驗證自動化

為安全機制建立安全基準和範本，在您的建置、管道和程序中接受測試和驗證。使用工具和自動化，持續測試和驗證所有安全控制。例如，掃描機器圖像和基礎設施即程式碼範本，檢查是否有安全漏洞、異常和偏離各階段既定基準。AWS CloudFormation Guard 可以協助您驗證 CloudFormation 範本是否安全、為您節省時間，以及減少組態錯誤的風險。

減少引入生產環境中的錯誤安全組態的數量至關重要；因此，在建置過程中最好能夠執行更多品質控制，並儘可能減少缺陷。應設計持續整合和持續部署 (CI/CD) 管道，在可能的情況下檢測安全問題。CI/CD 管道提供為建置和交付之每個階段增強安全的機會。CI/CD 安全工具也必須持續更新，以緩解不斷演變的威脅。

追蹤對您工作負載組態的變更，以協助進行合規稽核、變更管理，以及可能適用於您的調查。您可以使用 AWS Config，來記錄並評估 AWS 和第三方資源。它可讓您使用規則和一致性套件持續稽核和評估整體合規，這些規則和一致性套件是具有修復動作的規則集合。

變更追蹤應該包括規劃的變更，這是組織變更控制程序的一部分 (有時稱為 MACD—移動、新增、變更、刪除)，也包括非規劃的變更，以及非預期的變更，例如事故。基礎設施上可能會發生變更，但它們也可能與其他類別相關，例如程式碼儲存庫中的變更、機器映像和應用程式庫存變更、程序和政策變更，或文件變更。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 自動化組態管理：透過使用組態管理服務或工具，來自動執行和驗證安全組態。
  - AWS Systems Manager

- AWS CloudFormation
- 在 AWS 上設定 CI/CD 管道

資源

相關文件：

- 如何使用服務控制政策在 AWS Organizations 中的各個帳戶之間設定許可防護機制

相關影片：

- 使用 AWS Organizations 管理多帳戶 AWS 環境
- 以 Well-Architected 方式提供安全最佳實務

SEC01-BP07 使用威脅模型定義風險並確定優先順序

使用威脅模型來識別和維護潛在威脅的最新登錄資料。排定威脅的優先順序並調整安全控制，以防止、偵測和回應威脅。在不斷演變的安全形勢下，重新審視和維護此事項。

威脅建模提供了一種系統化的方法，以協助在設計過程中及早發現並解決安全問題。越早越好，因為與生命週期後期相比，緩解的成本更低。

威脅建模程序的典型核心步驟是：

1. 識別資產、執行者、進入點、元件、使用案例，以及信任等級，並將這些項目包含在設計圖中。
2. 識別威脅清單。
3. 對於每個威脅，識別緩解措施，其中可能包括安全控制實作。
4. 建立並審查風險矩陣以判斷威脅是否得到充分緩解。

威脅建模在工作負載 (或工作負載功能) 層級完成時最有效，可確保所有內容都可用於評估。隨著您的安全態勢演進，重新檢視並維護此矩陣。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 建立威脅模型：威脅模型可協助您識別和解決潛在的安全威脅。
  - NIST：以資料為中心的系統威脅建模指南

資源

相關文件：

- AWS 安全稽核指導方針
- 安全公告

相關影片：

- 以 Well-Architected 方式提供安全最佳實務

SEC01-BP08 定期評估和實作新的安全服務和功能

評估和實作 AWS 和 AWS 合作夥伴提供的安全服務和功能，讓您發展工作負載的安全狀態。AWS 安全部落格強調新的 AWS 服務和功能、實作指南和一般安全指引。AWS 最新消息？ 是及時了解所有新的 AWS 功能、服務和公告的最佳方式。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 規劃定期審查：建立一個審查活動行事曆，其中包含合規要求、對新的 AWS 安全功能和服務的評估以及最新的產業新聞。
- 探索 AWS 服務與功能：探索可用於您正在使用的服務的安全功能，並在新功能發佈時審查這些功能。
  - AWS 安全部落格
  - AWS 安全公告
  - AWS 服務文件
- 定義 AWS 服務的採用程序：定義採用新 AWS 服務的程序。包含您如何評估新 AWS 服務的功能，以及工作負載的合規要求。
- 測試新的服務和功能：在緊密複寫生產服務的非生產環境中發佈新服務和功能時，請對其進行測試。
- 實作其他防禦機制：實作自動化機制以保護您的工作負載，探索可用的選項。
  - 依 AWS Config 規則 修補不合規的 AWS 資源

資源

相關影片：

- 以 Well-Architected 方式提供安全最佳實務

身分和存取管理

問題

- SEC 2 如何管理人員和機器的身份驗證？
- SEC 3 如何管理人員和機器的許可？

SEC 2 如何管理人員和機器的身份驗證？

處理操作安全的 AWS 工作負載時，您需要管理兩種身分類型。了解您需要管理和授予存取權的身分類型，有助於確保正確的身分在適當的條件下存取正確的資源。

人員身分：您的管理員、開發人員、操作員和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些人是組織的成員，或與您協作的外部使用者，以及透過 Web 瀏覽器、用戶端應用程式或互動式命令列工具與 AWS 資源互動的使用者。

機器身分：您的服務應用程式、操作工具和工作負載需要身分，才能向 AWS 服務發出請求，例如讀取資料。這些身份包括在 AWS 環境中執行的機器，例如 Amazon EC2 執行個體或 AWS Lambda 函數。您也可以為需要存取權的外部人員管理機器身分。此外，您可能也有在 AWS 外部，需要存取 AWS 環境的機器。

最佳實務

- SEC02-BP01 使用強式登入機制
- SEC02-BP02 使用臨時登入資料
- SEC02-BP03 安全地存放和使用機密
- SEC02-BP04 倚賴集中化的身分提供者
- SEC02-BP05 定期稽核和輪換登入資料
- SEC02-BP06 利用使用者群組和屬性

SEC02-BP01 使用強式登入機制

強制執行密碼長度下限，並教育使用者避免使用常見密碼或重複使用密碼。透過軟體或硬體機制強制使用 Multi-Factor Authentication (MFA)，以提供額外的驗證保護層。例如，使用 IAM Identity Center 作為身份來源時，請進行 MFA 的「內容感知」或「永遠啟用」設定，並允許使用者註冊自己的 MFA 裝置以加速採用。使用外部身份提供者 (IdP) 時，設定您的 MFA 的 IdP。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 建立 Identify and Access Management (IAM) 政策來強制執行 MFA 登入：建立一個客戶管理的 IAM 政策，禁止所有 IAM 動作，除了允許使用者在下列頁面上假設角色、變更自己的登入資料，以及管理 MFA 裝置： My Security Credentials (我的安全登入資料)。
- 在您的身分供應商中啟用 MFA：在您使用的身分供應商或單一登入服務中啟用 MFA ，例如 AWS IAM Identity Center。
- 設定強式密碼政策：將強式 密碼政策 設定於 IAM 和聯合身分系統中，以協助防範暴力密碼破解攻擊。

- 定期輪換登入資料：確保工作負載的管理員會定期變更其密碼和存取金鑰 (若使用)。

資源

相關文件：

- AWS Secrets Manager 入門
- IAM 最佳實務
- 身份提供者與聯合
- AWS 帳戶根使用者
- AWS Secrets Manager 入門
- 暫時安全登入資料
- 安全合作夥伴解決方案：存取與存取控制
- 暫時安全登入資料
- AWS 帳戶根使用者

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 IAM Identity Center 大規模管理使用者許可
- 在每一層都能掌握身份

SEC02-BP02 使用臨時登入資料

需要身份才能動態取得 臨時登入資料。若是人力身份，請使用 AWS IAM Identity Center 或與 AWS Identity and Access Management (IAM) 角色聯合來存取 AWS 帳戶。若是機器身份，例如 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或 AWS Lambda 函數，需要使用 IAM 角色，而不是具有長期存取金鑰的 IAM 使用者。

若是使用 AWS Management Console的人類身份，需要使用者取得臨時登入資料並聯合至 AWS。您可以使用 AWS IAM Identity Center 使用者入口網站來執行此動作。針對需要 CLI 存取權的使用者，請確定他們使用 AWS CLI v2，其支援與 IAM Identity Center 的直接整合。使用者可以建立連結至 IAM Identity Center 帳戶和角色的 CLI 設定檔。CLI 會自動從 IAM Identity Center 擷取 AWS 登入資料，並代您重新整理。這樣就無需從 IAM Identity Center 主控台複製並貼上臨時 AWS 登入資料。針對 SDK，使用者應倚賴 AWS Security Token Service (AWS STS) 來擔任角色，以接收臨時登入資料。在某些情況下，臨時登入資料可能並不實用。您應注意存放存取金鑰的風險，經常輪換這些金鑰，並盡可能要求多重要素驗證 (MFA) 作為條件。使用上次存取的資訊來決定何時輪換或移除存取金鑰。

若您需要授予取用者存取 AWS 資源，請使用 Amazon Cognito 身份集區，並為其指派一組臨時、有限權限的登入資料來存取您的 AWS 資源。每個使用者的許可都是透過您建立的 IAM 角色 來控制。您可以定義規則，根據使用者 ID 字符中的宣告，為每個使用者選擇角色。您可以對已驗證使用者定義預設角色。您還可以對未驗證訪客使用者定義具有限制許可的 IAM 角色。

若是機器身份，您應倚賴 IAM 角色來授予 AWS 存取權。若是 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，您可以使用 Amazon EC2 的角色。。您可以將 IAM 角色連接至 Amazon EC2 執行個體，讓在 Amazon EC2 上執行的應用程式能夠使用 AWS 透過執行個體中繼資料服務 (IMDS) 自動建立、分發和輪換的臨時安全登入資料。AWS Well-Architected 最新版本 的 IMDS 可協助防範暴露臨時登入資訊的漏洞，並應實作。若要使用金鑰或密碼存取 Amazon EC2 執行個體，AWS Systems Manager 是一種更安全的方式，可使用預先安裝的代理程式存取和管理執行個體，而無須使用存放的密碼。此外，AWS Lambda 等其他 AWS 服務可讓您設定 IAM 服務角色，授予使用臨時登入資料執行 AWS 動作的服務許可。在您無法使用臨時登入資料的情況下，請使用程式設計工具，例如 AWS Secrets Manager，來自動輪換和管理登入資料。

定期稽核和輪換登入資料： 定期驗證 (最好是透過自動化工具) 是確認強制執行正確的控制項的必要項目。若是人類身份，您應要求使用者定期變更密碼，並使用臨時登入資料淘汰存取金鑰。當您從 IAM 使用者移至集中式身份時，可以 產生登入資料報告 來稽核您的 IAM 使用者。我們也建議您在身份供應商中強制執行 MFA 設定。您可以設定 AWS Config 規則 來監控這些設定。若是機器身份，您應倚賴使用 IAM 角色的臨時登入資料。在無法執行此操作的情況下，需要頻繁稽核和輪換存取金鑰。

安全地存放和使用機密： 針對與 IAM 無關且無法利用臨時登入資料的登入資料，例如資料庫登入，請使用專為處理機密管理而設計的服務，例如 Secrets Manager。Secrets Manager 讓您能夠使用 支援

的服務。為了稽核目的，存取機密的叫用會記錄在 AWS CloudTrail 中，而 IAM 許可能夠授予對這些機密的最低存取權。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 實作最低權限政策：將具有最低權限的存取政策指派給 IAM 群組和角色，以反映您已定義的使用者角色或職能。
  - 授予最低權限
- 移除不需要的權限：透過移除不必要的許可來實作最低權限。
  - 透過查看使用者活動來縮小政策範圍
  - 檢視角色存取

- 考慮使用許可界限：許可界限是使用受管政策的進階功能，可設定以身分為基礎的政策可授與 IAM 實體的最大許可。實體的許可界限只允許執行其以身分為基礎的政策和許可界限同時允許的動作。
  - 實驗室：IAM 許可邊界委派角色建立
- 考慮使用資源標籤的許可：您可以使用標籤來控制對支援標記之 AWS 資源的存取。您也可以標記 IAM 使用者和角色，以控制他們可以存取的內容。
  - 實驗室：EC2 的 IAM 標籤型存取控制
  - 屬性型存取控制 (ABAC)

資源

相關文件：

- AWS Secrets Manager 入門
- IAM 最佳實務
- 身份提供者與聯合
- 安全合作夥伴解決方案：存取與存取控制
- 暫時安全登入資料
- AWS 帳戶根使用者

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 AWS IAM Identity Center 大規模管理使用者許可
- 在每一層都能掌握身份

SEC02-BP03 安全地存放和使用機密

對於需要第三方應用程式密碼等機密的人力和機器身分，請在專業服務中使用最新的產業標準，以自動輪換的方式存放機密，例如對於與 IAM 無關且無法利用臨時登入資料的登入資料，例如資料庫登入，請使用專為處理機密管理而設計的服務，例如 AWS Secrets Manager。Secrets Manager 讓您能夠使用支援的服務輕鬆管理、輪換和安全地存放加密機密。為了稽核目的，存取機密的叫用會記錄在 AWS CloudTrail 中，而 IAM 許可能夠授予對這些機密的最低存取權。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 使用 AWS Secrets Manager： AWS Secrets Manager 是一項 AWS 服務，可讓您更輕鬆地管理機密。機密可以是資料庫登入資料、密碼、第三方 API 金鑰，甚至是任意文字。

資源

相關文件：

- AWS Secrets Manager 入門
- 身份提供者與聯合

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務

SEC02-BP04 倚賴集中化的身分提供者

若是人力身份，請倚賴可讓您在集中位置管理身份的身份供應商。因為您從單一位置建立、管理和撤銷存取權，因此這可讓您更輕鬆地管理多個應用程式和服務之間的存取。例如，若有人離開您的組織，您可從一個位置撤銷所有應用程式和服務 (包括 AWS) 的存取權。這可減少多個登入資料的需求，並提供與現有人力資源 (HR) 程序整合的機會。

針對與個別 AWS 帳戶的聯合，您可以透過 SAML 2.0 供應商使用 AWS 的集中化身份，並搭配 AWS Identity and Access Management。您可以使用與下列通訊協定相容的任何供應商，無論是由您在 AWS 中、AWS 外部託管，還是由 AWS Partner 提供： SAML 2.0 通訊協定。您可以使用 AWS 帳戶與所選供應商之間的聯合，透過使用 SAML 聲明取得臨時安全登入資料，授予使用者或應用程式叫用 AWS API 操作的存取權。此外還支援 Web 型單一登入，讓使用者能夠從您的登入網站登入 AWS Management Console。

針對與 AWS Organizations 中多個帳戶的聯合，您可以在 AWS IAM Identity Center (IAM Identity Center)中設定您的身份來源，並指定使用者和群組的存放位置。設定好之後，您的身份供應商即真實來源，而資訊可以使用跨網域身份管理系統 (SCIM) v2.0 通訊協定來 同步 。然後，您可以查詢使用者或群組，並授予他們對 AWS 帳戶、雲端應用程式或兩者的IAM Identity Center存取權。

IAM Identity Center 與 AWS Organizations 整合，讓您只需設定身份供應商一次，然後即可將 存取權授予組織中管理的現有及新帳戶 。IAM Identity Center 為您提供預設存放區，這可用於管理使用者和群組。若您選擇使用 IAM Identity Center 存放區，則建立使用者和群組，並將其存取層級指派給您的 AWS 帳戶和應用程式，並記住最低權限的最佳實務。或者，您可以選擇使用 SAML 2.0 連線至您的外部身份供應商 ， 或使用 AWS Directory Service 連線至您的 Microsoft AD 目錄。設定好之後，您可以透過中央身分提供者進行身分驗證，登入 AWS Management Console 或 AWS 行動應用程式。

若要管理工作負載的最終使用者或取用者，例如行動應用程式，您可以使用 Amazon Cognito。它可為您的 Web 和行動應用程式提供身份驗證、授權和使用者管理。您的使用者可以憑使用者名稱和密碼直接登入，或透過第三方 (例如 Amazon、Apple、Facebook 或 Google) 登入。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 集中管理存取：建立 Identity and Access Management (IAM) 身分提供者實體，以在您的 AWS 帳戶與您的身分提供者 (IdP) 之間建立信任的關係。IAM 支援與 OpenID Connect (OIDC) 或 SAML 2.0 (安全聲明標記語言 2.0) 相容的 IdP。
  - 身份提供者與聯合
- 集中應用程式存取：考慮使用 Amazon Cognito 集中存取應用程式。它可讓您快速輕鬆地將使用者註冊、登入和存取控制新增到 Web 和行動應用程式。 Amazon Cognito 可擴展到數百萬使用者，並支援透過 SAML 2.0 使用社交身分提供者 (例如 Facebook、Google 和 Amazon) 以及企業身分提供者進行登入。

- 移除舊的 IAM 使用者與群組：在您開始使用身分提供者 (IdP) 後，請移除不再需要的 IAM 使用者與群組。

- 尋找未使用的登入資料
- 刪除 IAM 群組

資源

相關文件：

- IAM 最佳實務
- 安全合作夥伴解決方案：存取與存取控制
- 暫時安全登入資料
- AWS 帳戶根使用者

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 AWS IAM Identity Center 大規模管理使用者許可
- 在每一層都能掌握身份

SEC02-BP05 定期稽核和輪換登入資料

當您無法倚賴臨時登入資料且需要長期登入資料時，請稽核登入資料以確保定義的控制 (例如 多重要素驗證 (MFA)) 會定期強制執行、輪換，且具有適當的存取層級。定期驗證 (最好是透過自動化工具) 是確認強制執行正確的控制項的必要項目。若是人類身份，您應要求使用者定期變更密碼，並使用臨時登入資料淘汰存取金鑰。當您從 AWS Identity and Access Management (IAM) 使用者移至集中式身份時，可以 產生登入資料報告 來稽核您的 IAM 使用者。我們也建議您在身份供應商中強制執行 MFA 設定。您可以設定 AWS Config 規則 來監控這些設定。若是機器身份，您應倚賴使用 IAM 角色的臨時登入資料。在無法執行此操作的情況下，需要頻繁稽核和輪換存取金鑰。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 定期稽核登入資料：使用登入資料報告和 Identify and Access Management (IAM) Access Analyzer，來稽核 IAM 登入資料和許可。
  - IAM Access Analyzer
  - 獲取登入資料報告

  - 實驗室：自動化 IAM 使用者清除
- 使用存取層級來檢閱 IAM 許可：為了改善 AWS 帳戶的安全性，請定期檢閱和監控每個 IAM 政策。請確定您的政策授予僅執行必要動作所需的最低權限。
  - 使用存取層級來檢閱 IAM 許可
- 考慮自動化 IAM 資源建立和更新：AWS CloudFormation 可以用來自動化 IAM 資源 (包括角色和政策) 的部署，以減少人為錯誤，因為範本可以進行驗證和進行版本控制。
  - 實驗室：IAM 群組和角色的自動部署

資源

相關文件：

- AWS Secrets Manager 入門
- IAM 最佳實務
- 身份提供者與聯合
- 安全合作夥伴解決方案：存取與存取控制
- 暫時安全登入資料

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 AWS IAM Identity Center 大規模管理使用者許可
- 在每一層都能掌握身份

SEC02-BP06 利用使用者群組和屬性

隨著您管理的使用者人數增加，您需要確定整理使用者的方式，以便大規模管理使用者。將具有共同安全需求的使用者放在身分供應商定義的群組中，並設置機制，以確保可用於存取控制的使用者屬性 (例如，部門或位置) 正確並已更新。使用這些群組和屬性 (而非個別使用者) 來控制存取權。這可讓您透過許可集合一次變更使用者的群組成員資格或屬性集中管理存取權，而不是在使用者存取需求變更時更新許多個別政策。您可以使用 AWS IAM Identity Center (IAM Identity Center) 來管理使用者群組和屬性。IAM Identity Center 支援最常用的屬性，無論是在使用者建立期間手動輸入，還是使用同步引擎自動佈建，例如 System for Cross-Domain Identity Management (SCIM) 規格中所定義。

將具有共同安全需求的使用者放在身分供應商定義的群組中，並設置機制，以確保可用於存取控制的使用者屬性 (例如，部門或位置) 正確並已更新。使用這些群組和屬性 (而非個別使用者) 來控制存取情形。這可讓您透過一次變更使用者的群組成員資格或屬性集中管理存取權，而不是在使用者存取需求變更時更新許多個別政策。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 如果您是使用 AWS IAM Identity Center (IAM Identity Center)，請設定群組：IAM Identity Center 可讓您設定使用者群組，並將所需的許可層級指派給群組。
  - AWS 單一登入 - 管理身分
- 了解屬性型存取控制 (ABAC)：ABAC 是一種授權策略，可根據屬性定義許可。
  - 什麼是 ABAC for AWS？
  - 實驗室：EC2 的 IAM 標籤型存取控制

資源

相關文件：

- AWS Secrets Manager 入門
- IAM 最佳實務
- 身份提供者與聯合
- AWS 帳戶根使用者

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 AWS IAM Identity Center 大規模管理使用者許可
- 在每一層都能掌握身份

相關範例：

- 實驗室：EC2 的 IAM 標籤型存取控制

SEC 3 如何管理人員和機器的許可？

管理許可，以控制對需要存取 AWS 和工作負載的人員和機器身分的存取。許可控制誰可以在何種條件下存取哪些內容。

最佳實務

- SEC03-BP01 定義存取需求
- SEC03-BP02 授予最低權限存取權
- SEC03-BP03 建立緊急存取程序
- SEC03-BP04 持續減少許可
- SEC03-BP05 為您的組織定義許可防護機制
- SEC03-BP06 根據生命週期管理存取
- SEC03-BP07 分析公有和跨帳戶存取權
- SEC03-BP08 安全地共用資源

SEC03-BP01 定義存取需求

工作負載的每個組成部分或資源都需要由管理員、最終使用者或其他組成部分存取。您需要清楚定義哪些人或哪些項目應該可以存取每個組成部分，然後選擇適當的身分類型與身分驗證和授權方法。

常見的反模式：

- 將機密硬式編碼或存放在應用程式中。
- 為每名使用者授予自訂許可。
- 使用長期憑證。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

工作負載的每個組成部分或資源都需要由管理員、最終使用者或其他組成部分存取。您需要清楚定義哪些人或哪些項目應該可以存取每個組成部分，然後選擇適當的身分類型與身分驗證和授權方法。

應提供組織中對 AWS 帳戶 的定期存取 (使用 聯合存取 或集中式的身分提供者)。 您應集中進行身分管理，並確保有既定的實務，可將 AWS 存取整合至員工的存取生命週期。例如，當員工改為擔任具有不同存取層級的任務角色時，其群組成員資格也應變更，以反映新的存取需求。

為非人類身分定義存取需求時，請判斷哪些應用程式和組成部分需要存取權，以及如何授予許可。使用透過最低權限存取模型建置的 IAM 角色是建議的方法。AWS 受管政策 提供預先定義的 IAM 政策，其中涵蓋最常見的使用案例。

AWS 服務，例如 AWS Secrets Manager 和 AWS Systems Manager 參數存放區，可以協助在無法使用 IAM 角色的情況下，將機密從應用程式或工作負載中安全地分離。在 Secrets Manager 中，您可以為憑證建立自動輪換。您可以使用 Systems Manager 來參考指令碼、命令、SSM 文件、組態和自動化工作流程中的參數，方法是使用您在建立參數時指定的唯一名稱。

您可以使用 AWS Identity and Access Management Roles Anywhere 來取得 IAM 中的臨時安全憑證，該憑證適用於在 AWS 以外執行的工作負載。您的工作負載可以使用相同的 IAM 政策 和 IAM 角色，您可以將這些政策和角色與 AWS 應用程式搭配使用，來存取 AWS 資源。

可能的話，請選擇短期暫時憑證，而不是長期靜態憑證。對於您希望 IAM 使用者具備程式設計存取權和長期憑證的情況，請使用 存取金鑰前次使用的資訊 來輪換和移除存取金鑰。

資源

相關文件：

- 屬性型存取控制 (ABAC)
- AWS IAM Identity Center
- IAM Roles Anywhere
- IAM Identity Center 的 AWS 受管政策
- AWSIAM 政策條件
- IAM 使用案例
- 移除不需要的憑證
- 制定政策
- 如何根據 AWS 帳戶、OU 或組織控制對 AWS 資源的存取權
- 使用 AWS Secrets Manager 中增強的搜尋功能來輕鬆識別、安排和管理機密

相關影片：

- 在 60 分鐘內精通 IAM 政策
- 責任區隔、最低權限、委派和 CI/CD

- 簡化身分和存取管理，以促進創新

SEC03-BP02 授予最低權限存取權

透過允許在特定情況下對特定 AWS 資源執行特定動作，僅授予身分所需的存取權。倚賴群組和身分屬性，大規模動態設定許可，而不是定義個別使用者的許可。例如，您可以允許一組開發人員的存取權，以只管理其專案的資源。如此一來，將開發人員從群組中移除時，在使用該群組來進行存取控制的任何地方都會撤銷該開發人員的存取權，完全不需要變更存取政策。

常見的反模式：

- 預設授予使用者管理員許可。
- 使用根帳戶來處理每日活動。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

建立 最低權限 原則可確保在平衡可用性和效率的同時，僅允許身份執行完成特定任務所需的最少量功能。依此原則操作會限制非預期存取，並協助確保您可以稽核誰有權存取哪些資源。在 AWS 中，除了根使用者以外，身分依預設沒有任何許可。根使用者的憑證應受到嚴格的控制，使用範圍僅限於幾個 特定任務。

您可以使用政策明確授予與 IAM 或資源實體連接的許可，例如聯合身分或機器或資源 (例如 S3 儲存貯體) 使用的 IAM 角色。建立和連接政策時，您可以指定必須為 true 的服務動作、資源和條件，以便 AWS 允許存取權。AWS 支援各種條件，以協助您縮減存取權範圍。例如，使用 `PrincipalOrgID` 條件金鑰，AWS Organizations 的識別符經過驗證，以便在 AWS 組織內授予存取權。

此外，您還可以控制 AWS 服務代您發出的請求，例如 AWS CloudFormation 使用 `CalledVia` 條件金鑰建立 AWS Lambda 函數。您應將不同政策類型分層，以便有效地將整體許可限制在某個帳戶中。例如，您可以允許應用程式團隊建立其專屬的 IAM 政策，但使用 許可界限 來限制他們可以授予的許可上限。

有幾個 AWS 功能，可協助您擴展許可管理並遵循最低權限原則。屬性型存取控制 可讓您根據資源的標籤 來限制許可，以便根據套用至資源和呼叫 IAM 主體的標籤來做出授權決定。此可讓您結合標記和許可政策，以達成更精細的資源存取，而不需自訂許多政策。

另一個加快最低權限政策建立的方式，是在活動執行後，將 CloudTrail 許可作為政策基礎。IAM Access Analyzer 可根據活動自動產生 IAM 政策。您可以在組織或個別帳戶層級中使用 IAM Access Advisor， 來追蹤特定政策的最新存取資訊。

建立審查這些詳細資料和移除不需要許可的規律。您應在 AWS 組織中建立許可防護機制，來控制任何成員帳戶中的許可上限。例如 AWS Control Tower 之類的服務有規範性的受管預防性控制項 並可讓您定義自己的控制項。

資源

相關文件：

- IAM 實體的許可界限
- 寫入最低權限 IAM 政策的技巧
- IAM Access Analyzer 透過根據存取活動產生 IAM 政策，來輕鬆實作最低權限許可
- 使用上次存取的資訊以精簡許可
- IAM 政策類型以及何時使用這些政策
- 使用 IAM 政策模擬器測試 IAM 政策
- AWS Control Tower 中的防護機制
- 零信任架構：AWS 觀點
- 如何使用 CloudFormation StackSets 實作最低權限原則

相關影片：

- 下一代許可管理
- 零信任：AWS 觀點
- 我如何使用許可界限，來限制 IAM 使用者和角色避免權限升級？

相關範例：

- 實驗室：IAM 許可界限委派角色建立

SEC03-BP03 建立緊急存取程序

在極少數的狀況下，自動化程序或管道發生問題時，允許緊急存取工作負載的程序。這可協助您倚賴最低權限的存取權，但確保使用者可在需要時取得適當的存取層級。例如，建立一個程序，讓管理員驗證和核准使用者的請求，例如，用於存取的緊急 AWS 跨帳戶角色，或管理員需遵循以驗證和核准緊急請求的特定程序。

常見的反模式：

- 未有可用的緊急程序，而無法從您現有身分組態的中斷中復原。
- 授予長期提升的許可，以供疑難排解或復原之用。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

緊急存取的建立可能有數種形式，您應做好相關的準備。第一個就是主要身分提供者的失敗。在此情況下，您應倚賴具有必要復原許可的第二種存取方法。此方法可能是備份身分提供者或 IAM 使用者。此第二個方法應 受到嚴格的控制、監控，並在 使用時發出通知。緊急存取身分應來自專門用於此用途的帳戶，並僅具備相關許可，以擔任專為復原而設計的角色。

您也應為需要臨時提升管理存取權的緊急存取情況做好準備。常見的情境是將突變的許可限制在用於部署變更的自動化程序中。在此程序發生問題時，使用者可能需要請求提升許可來還原功能。在此情況下，建立以下流程，即使用者可以請求提升存取權，而管理員可以進行驗證和核准。我們會在以下位置提供實作計劃，此計劃詳細說明預先佈建存取權和為緊急情況、角色做準備的最佳實務指引 SEC10-BP05 預先佈建存取權。

資源

相關文件：

- 在 AWS 上監控和通知
- 管理臨時提升的存取權

相關影片：

- 在 60 分鐘內精通 IAM 政策

SEC03-BP04 持續減少許可

當團隊和工作負載決定他們需要的存取時，請移除他們不再使用的許可，並建立檢閱程序以達到最低權限的許可。持續監控和減少未使用的身分和許可。

有時，當團隊和專案剛開始時，您可以選擇授予廣泛存取權 (在開發或測試環境中) 來激發創新和靈活性。我們建議您持續評估存取權，而且尤其在生產環境中，將存取權限制為僅必要許可及達到最低權限。AWS 提供存取權分析功能，以協助您識別未使用的存取權。為了協助您識別未使用的使用者、角色和登入資料，AWS 會分析存取活動，並提供存取金鑰和角色上次使用的資訊。您可以使用 上次存取

的時間戳記 來 識別未使用的使用者和角色，並將其移除。此外，您可以檢閱服務和動作上次存取的資訊，以識別和 加強特定使用者和角色的許可。例如，您可以使用上次存取的資訊來識別您的應用程式角色所需的特定 Amazon Simple Storage Service (Amazon S3) 動作，並限制只能存取這些動作。這些功能可在AWS Management Console中透過程式設計方式提供，讓您能夠將這些功能併入基礎設施工作流程和自動化工具中。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 設定 AWS Identify and Access Management (IAM) Access Analyzer：AWS IAM Access Analyzer 可協助您識別組織和帳戶中與外部實體共用的資源，例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體或 IAM 角色。
  - AWS IAM Access Analyzer

資源

相關文件：

- 屬性型存取控制 (ABAC)
- 授予最低權限
- 移除不需要的登入資料
- 制定原則

相關影片：

- 在 60 分鐘內精通 IAM 政策
- 責任區隔、最低權限、委派和 CI/CD

SEC03-BP05 為您的組織定義許可防護機制

建立通用控制項，限制對組織中所有身分的存取權。比方說，您可以限制對特定 AWS 區域 的存取權，或防止操作人員刪除常見資源，例如用於中央安全團隊的 IAM 角色。

常見的反模式：

- 在組織管理員帳戶中執行工作負載。

- 在相同帳戶中執行生產和非生產工作負載。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

隨著工作負載在 AWS 中成長而需要加以管理，應該使用帳戶區隔這些工作負載，並使用 AWS Organizations 管理那些帳戶。我們建議您建立常見的許可防護機制，限制對組織中所有身分的存取權。比方說，您可以限制對特定 AWS 區域 的存取權，或防止團隊刪除常見資源，例如中央安全團隊使用的 IAM 角色。

您可以透過實作範例服務控制政策來開始使用，例如防止使用者停用重要服務。SCP 使用 IAM 政策語言，並讓您能夠建立所有 IAM 主體 (使用者和角色) 都遵循的控制項。您可以根據特定條件限制對特定服務動作、資源的存取，以滿足您組織的存取控制需求。如有必要，您可以為防護機制定義例外狀況。例如，您可以限制帳戶中所有 IAM 實體的服務動作，但特定管理員角色除外。

我們建議您在管理帳戶中避免執行工作負載。應使用管理帳戶來管控和部署會對成員帳戶造成影響的安全防護機制。某些 AWS 服務支援使用委派的管理員帳戶。當委派的帳戶可用時，您應使用該帳戶，而不是管理帳戶。您應嚴格限制組織管理員帳戶的存取權。

使用多帳戶策略，可讓您在將防護機制套用到工作負載時享有更大的彈性。AWS 安全性參考架構提供規範性指引，其中說明如何設計帳戶結構。AWS Control Tower 之類的 AWS 服務提供的功能，可同時集中管理組織中的預防性和偵測性控制項。清楚定義每個帳戶或組織中 OU 的用途，並根據該用途限制控制項。

資源

相關文件：

- AWS Organizations
- 服務控制政策 (SCP)
- 在多帳戶環境中充分利用服務控制政策
- AWS 安全性參考架構 (AWS SRA)

相關影片：

- 使用服務控制政策來強制執行預防性防護機制
- 使用 AWS Control Tower 大規模建立管控
- 深入探討 AWS 身分和存取管理

SEC03-BP06 根據生命週期管理存取

將存取控制與操作員和應用程式之生命週期以及集中化的聯合身分供應商相整合。例如，在使用者離職或變動職務時移除其存取權。

當您使用個別帳戶管理工作負載時，有時您需要在這些帳戶之間共用資源。建議您使用 AWS Resource Access Manager (AWS RAM)。此服務可讓您輕鬆、安全地在 AWS Organizations和組織單位內共用 AWS 資源。使用 AWS RAM，當帳戶移入和移出共用它們的組織或組織單位時，會自動授予或撤銷共用資源的存取權。這可協助您確保資源只與您的預期帳戶共用。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

使用者存取生命週期：為加入的新使用者、工作職能變更和離開的使用者，實作使用者存取生命週期的政策，以便只有目前的使用者擁有存取權。

資源

相關文件：

- 屬性型存取控制 (ABAC)
- 授予最低權限
- IAM Access Analyzer
- 移除不需要的登入資料
- 制定原則

相關影片：

- 在 60 分鐘內精通 IAM 政策
- 責任區隔、最低權限、委派和 CI/CD

SEC03-BP07 分析公有和跨帳戶存取權

持續監控強調公有和跨帳戶存取權的問題清單。減少僅對需要此類存取之資源的公有存取權和跨帳戶存取權。

常見的反模式：

- 未遵循程序來管控跨帳戶的存取權以及資源的公有存取權。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在 AWS 中，您可以授予另一個帳戶中資源的存取權。您授予直接跨帳戶存取權，方法是使用附加至資源的政策 (例如，Amazon Simple Storage Service (Amazon S3) 儲存貯體政策) 或透過允許某個身分在另一個帳戶中擔任 IAM 角色。使用資源政策時，確認將存取權授予貴組織中的身分，且您預計公開資源。定義程序，來核准所有需要公開提供的資源。

IAM Access Analyzer 使用 可證明的安全 ，來找出從其帳戶外部存取資源的所有路徑。其會持續審查資源政策，並報告公有和跨帳戶存取權的問題清單，讓您輕鬆分析可能的各種存取。考量使用 AWS Organizations 設定 IAM Access Analyzer，來確認您對所有帳戶的能見度。IAM Access Analyzer 也讓您能夠 預覽 Access Analyzer 問題清單，然後再部署資源許可。這可讓您驗證政策變更是否僅授予對您資源的預期公有和跨帳戶存取權。預計授予多帳戶存取權時，您可以使用 信任政策來控制在什麼情況下可以擔任角色。例如，您可以將角色擔任限制在特定來源 IP 範圍。

您也可以使用 AWS Config，透過 AWS Config 政策檢查，報告和修復資源的任何意外公有存取組態。諸如 AWS Control Tower 和 AWS Security Hub 之類的服務可簡化 AWS Organizations 之間的部署檢查和防護機制，來找出和修復公開暴露的資源。例如，AWS Control Tower 具備受管的防護機制，可偵測是否 所有 AWS 帳戶都可復原所有 Amazon EBS 快照。

資源

相關文件：

- 使用 AWS Identity and Access Management Access Analyzer
- AWS Control Tower 中的防護機制
- AWS 基礎安全最佳實務標準
- AWS Config 受管規則
- AWS Trusted Advisor 檢查參考

相關影片：

- 保護多帳戶環境的最佳實務
- 深入了解 IAM Access Analyzer

SEC03-BP08 安全地共用資源

管控跨帳戶或 AWS Organizations 內共用資源的使用量。監控共用資源並審查共用資源的存取。

常見的反模式：

- 在授予第三方跨帳戶存取權時使用預設 IAM 信任政策。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在使用多個 AWS 帳戶管理工作負載時，您可能需要在帳戶之間共用資源。這可能通常會是 AWS Organizations 中的跨帳戶共用。數種 AWS 服務，例如 AWS Security Hub、Amazon GuardDuty和 AWS Backup 都有與 Organizations 整合的跨帳戶功能。您可以使用 AWS Resource Access Manager 來共用其他常見的資源，例如 VPC 子網路或傳輸閘道連接，AWS Network Firewall，或 Amazon SageMaker Runtime 管道。如果您想要確保帳戶的資源共用範圍僅限於 Organizations，我們建議使用服務控制政策 (SCP) 來避免對外部主體的存取。

共用資源時，您應採取措施來防止意外的存取。我們建議結合以身分為基礎的控制項和網路控制項，來為貴組織建立資料周邊。這些控制項應嚴格限制可以共用哪些資源，並防止共用或公開遭禁止的資源。例如，在資料周邊中，您可以使用 VPC 端點政策和 `aws:PrincipalOrgId` 條件，來確保 Amazon S3 儲存貯體的存取身分是貴組織的一員。

在某些案例中，您可能想要允許共用 Organizations 外部的資源或授予第三方存取帳戶。例如，合作夥伴可能會提供監控解決方案，該解決方案需要存取您帳戶中的資源。在那些案例中，您應僅使用第三方需要的權限，來建立 IAM 跨帳戶角色。您也應使用外部 ID 條件 來建立信任政策。使用外部 ID 時，您應為每個第三方產生唯一的 ID。唯一 ID 的提供者或控制者不得是第三方。如果第三方不再需要存取您的環境，您應將角色移除。在所有案例中，您也應避免為第三方提供長期的 IAM 憑證。掌握對其他原生支援共用的 AWS 服務的狀態。例如，AWS Well-Architected Tool 允許 在其他 AWS 帳戶中 共用工作負載。

使用 Amazon S3 之類的服務時，建議您 停用 Amazon S3 儲存貯體的 ACL， 並使用 IAM 政策來定義存取控制。如需限制 從 Amazon CloudFront對 Amazon S3 原點的存取，請從原始存取身分 (OAI) 遷移至原始存取控制 (OAC)，後者支援額外的功能，包含使用下列項目的伺服器端加密功能： AWS KMS。

資源

相關文件：

- 儲存貯體擁有者將跨帳戶許可授予非其擁有的物件
- 如何使用信任政策搭配 IAM

- 在 AWS 上建置資料周邊
- 向第三方授予對 AWS 資源的存取權限時如何使用外部 ID

相關影片：

- 使用 AWS Resource Access Manager 進行精密的存取
- 使用 VPC 端點確保資料周邊的安全
- 在 AWS 上建立資料周邊

偵測

問題

- SEC 4 您如何偵測和調查安全事件？

SEC 4 您如何偵測和調查安全事件？

從日誌和指標中擷取並分析事件以掌握情況。針對安全事件和潛在威脅採取行動，有助於保護工作負載。

最佳實務

- SEC04-BP01 設定服務和應用程式記錄
- SEC04-BP02 集中分析日誌、問題清單和指標
- SEC04-BP03 自動回應事件
- SEC04-BP04 實作可採取行動的安全事件

SEC04-BP01 設定服務和應用程式記錄

設定整個工作負載中的記錄，包括應用程式日誌、資源日誌和 AWS 服務日誌。例如：確定組織內的所有帳戶已啟用 AWS CloudTrail、Amazon CloudWatch Logs、Amazon GuardDuty 和 AWS Security Hub。

基本實務是在帳戶層級建立一套偵測機制。這組基本機制旨在記錄和偵測對您帳戶中所有資源的各種動作。可讓您建立完整的偵測功能，其中包含自動修復的選項，以及新增功能的合作夥伴整合。

在 AWS 中，此基本套組中的服務包括：

- AWS CloudTrail 提供 AWS 帳戶活動的事件歷史記錄，包括透過 AWS Management Console、AWS 開發套件、命令列工具及其他 AWS 服務所採取的動作。
- AWS Config 可監控和記錄 AWS 資源組態，並讓您根據所需的組態自動評估和修復。
- Amazon GuardDuty 是威脅偵測服務，可持續監控惡意活動和未經授權的行為，以保護 AWS 帳戶和工作負載。
- AWS Security Hub 提供以單一位置從多個 AWS 服務和選用的第三方產品將安全提醒或發現結果加以彙總、組織和排列優先順序，為您提供安全提醒和合規狀態的全面檢視。

建立在帳戶層級的基礎上，許多核心 AWS 服務 (例如 Amazon Virtual Private Cloud Console (Amazon VPC)提供服務層級的記錄功能。Amazon VPC 流程日誌 可讓您擷在網路介面取傳入和傳出之 IP 流量的相關資訊，可提供關於連線歷史記錄的寶貴洞見，並能根據異常行為觸發自動動作。

對於不是源自 AWS 服務的Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和應用程式日誌記錄，可以使用 Amazon CloudWatch Logs存放和分析日誌。代理 程式 會從作業系統和執行中的應用程式收集日誌，並且自動存放。日誌在 CloudWatch Logs 中可用之後，您可以 即時處理，或使用 CloudWatch Logs Insights深入分析。

對於收集和彙總日誌而言，同等重要的是從複雜架構產生的大量日誌和事件資料中，提取有意義見解的能力。如需更多詳細資訊，請參閱可靠性支柱白皮書的 監控 經濟實惠的 可靠性支柱白皮書 。日誌本身可能包含視為敏感的資料，無論是當應用程式資料錯誤地進入 CloudWatch Logs 代理程式擷取的日誌檔時，或是為了日誌彙總設定跨區域記錄時，而且在於跨邊界運送特定類型的資訊有法令方面的考量。

其中一種方法是使用 AWS Lambda 函數 (在交付日誌時應事件而觸發)，在轉送到集中記錄位置 (例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體) 之前篩選和修訂日誌資料。未修訂的日誌可以保留在本機儲存貯體中，直到合理時間 (由法規和法律團隊決定) 過去，屆時 Amazon S3 生命週期規則即能自動刪除。使用 Amazon S3 Object Lock可進一步保護 Amazon S3 中的日誌，您可以使用單寫多讀 (WORM) 模型存放物件。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 啟用 AWS 服務的記錄：啟用 AWS 服務的記錄以符合您的需求。記錄功能如下：Amazon VPC 流程日誌、Elastic Load Balancing (ELB) 日誌、Amazon S3 儲存貯體日誌、CloudFront 存取日誌、Amazon Route 53 查詢日誌和 Amazon Relational Database Service (Amazon RDS) 日誌。
  - AWS Answers：原生 AWS 安全記錄功能

- 評估並啟用作業系統和應用程式專屬的記錄，以偵測可疑行為。
  - CloudWatch Logs 入門
  - 開發人員工具和日誌分析
- 將適當控制套用至日誌：日誌可能包含敏感資訊，因此只有授權使用者可以存取。考慮限制對 Amazon S3 儲存貯體和 CloudWatch Logs 日誌群組的許可。
  - Amazon CloudWatch 的身分驗證與存取控制
  - Amazon S3 中的身分和存取管理
- 設定 Amazon GuardDuty：GuardDuty 是威脅偵測服務，可持續尋找惡意活動和未經授權的行為，以保護 AWS 帳戶和工作負載。使用實驗室啟用 GuardDuty 並設定電子郵件的自動提醒。
- 在 CloudTrail 中設定自訂追蹤：設定軌跡可讓您儲存日誌的時間長於預設時間，以便之後分析這些日誌。
- 啟用 AWS Config：AWS Config 提供了您的 AWS 帳戶中 AWS 資源組態的詳細檢視。檢視內容包括資源之間的關係，以及它們過去的組態，以便您了解組態和關係如何隨時間變更。
- 啟用 AWS Security Hub：Security Hub 提供 AWS 安全狀態的全面檢視，並協助您檢查是否符合安全產業標準和最佳實務。Security Hub 會收集 AWS 帳戶、服務和支援的第三方合作夥伴產品的安全資料，協助您分析安全趨勢並找出優先順序最高的安全問題。

資源

相關文件：

- Amazon CloudWatch
- Amazon EventBridge
- 入門：Amazon CloudWatch Logs
- 安全合作夥伴解決方案：記錄與監控

相關影片：

- 集中監控資源組態與合規
- 修補 Amazon GuardDuty 和 AWS Security Hub 發現結果
- 雲端中的威脅管理：Amazon GuardDuty 和 AWS Security Hub

相關範例：

- 實驗室：偵測控制的自動部署

SEC04-BP02 集中分析日誌、問題清單和指標

安全營運團隊倚賴日誌的收集和使用搜尋工具來探索感興趣的潛在事件，這類事件可能是未經授權的活動或無意間的變更。但是，僅分析收集的資料並手動處理資訊，不足以應付繁複架構之中流通的資訊量。單靠分析和報告並不能幫助分配合適的資源，以及時處理事件。

建立成熟的安全營運團隊的最佳實務是，將安全事件和結果流深入整合到通知和工作流程系統中，例如票證系統、錯誤或問題系統或其他安全資訊和事件管理 (SIEM) 系統。如此能使工作流程擺脫電子郵件和靜態報告，讓您能夠路由、向上呈報和管理事件或結果。許多組織也正在將安全提醒整合到其聊天或協作和開發人員生產力平台中。對於開始自動化的組織，以 API 驅動的低延遲票證系統在規劃要先自動化什麼時能夠提供相當大的彈性。

此最佳實務不僅適用於從描述使用者活動或網路事件的日誌訊息所產生的安全事件，也適用於從基礎設施本身偵測到的變更所產生的安全事件。能夠偵測變更、判斷變更是否適當，然後將該資訊路由到正確的修復工作流程，這對於維護和驗證安全架構至關重要；在變更的環境中，其不甚理想的性質足夠細微，以致目前無法透過 AWS Identity and Access Management (IAM) 和 AWS Organizations 組態的組合來阻止執行。

Amazon GuardDuty 和 AWS Security Hub 針對也會透過其他 AWS 服務提供給您的日誌記錄提供彙總、重複資料刪除和分析機制。GuardDuty 會從 AWS CloudTrail 管理和資料事件、VPC DNS 日誌和 VPC 流程日誌等來源擷取、彙總和分析資訊。Security Hub 可從 GuardDuty、AWS Config、Amazon Inspector、Amazon Macie、AWS Firewall Manager，以及可在 AWS Marketplace 取得的眾多第三方安全產品擷取、彙總和分析輸出，而且如果照著建置，也會從您自己的程式碼這樣做。GuardDuty 和 Security Hub 都有管理員-成員模型，可跨多個帳戶彙總發現結果和洞見，其中使用 Security Hub 的客戶通常以內部部署的 SIEM 做為 AWS 端日誌，並有提醒預處理器和彙總器，藉以經由 AWS Lambda 處理器和轉寄站導入 Amazon EventBridge。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 評估日誌處理能力：評估可用於處理日誌的選項。
  - 使用 Amazon OpenSearch Service 記錄和監控 (幾乎) 一切
  - 尋找一個專門從事記錄和監控解決方案的合作夥伴
- 若要開始分析 CloudTrail 日誌，請測試 Amazon Athena。
  - 設定 Athena 來分析 CloudTrail 日誌

- 在 AWS 中實作集中記錄：請參閱下列 AWS 範例解決方案，來集中多個來源記錄。
  - 將記錄解決方案集中化
- 透過合作夥伴實作集中記錄：APN 合作夥伴提供的解決方案可協助您集中分析日誌。
  - 記錄和監控

資源

相關文件：

- AWS Answers：集中式記錄
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- 入門：Amazon CloudWatch Logs
- 安全合作夥伴解決方案：記錄與監控

相關影片：

- 集中監控資源組態與合規
- 修補 Amazon GuardDuty 和 AWS Security Hub 發現結果
- 雲端中的威脅管理：Amazon GuardDuty 和 AWS Security Hub

SEC04-BP03 自動回應事件

使用自動化來調查和修復事件可減少人工作業和人為錯誤，還可讓您擴展調查功能。定期檢閱將協助您調整自動化工具並持續反覆運算。

在 AWS 中，您可以使用 Amazon EventBridge 來調查感興趣的事件，以及自動化工作流程中潛在意外變更的相關資訊。該服務能提供可擴展的規則引擎，旨在代理原生 AWS 事件格式 (例如 AWS CloudTrail 事件) 以及您可從應用程式產生的自訂事件。Amazon GuardDuty 也可讓您將事件路由到建立事件回應系統的工作流程系統 (AWS Step Functions)，或路由到中央安全帳戶，或路由到儲存貯體供進一步分析。

也可以偵測變更，並將此資訊路由到正確的工作流程，方法為使用 AWS Config 規則 和 合規套件。AWS Config 偵測範圍內服務的變更 (但延遲比 EventBridge 高)，並產生可使用 AWS Config 規則

剖析的事件，用於還原、執行合規政策以及將資訊轉發到系統 (例如變更管理平台和營運票證系統)。除了編寫自己的 Lambda 函數來回應 AWS Config 事件，您還可以利用 AWS Config 規則 開發套件和開放原始碼庫 AWS Config 規則。合規套件是 AWS Config 規則 和修補動作的集合，其會部署為以 YAML 範本撰寫的單一實體。路由層 範例合規套件範本 適用於 Well-Architected 安全支柱。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用 GuardDuty 實作自動提醒：GuardDuty 是一種威脅偵測服務，可持續監控惡意活動和未經授權的行為，以保護 AWS 帳戶和工作負載。啟用 GuardDuty 並設定自動提醒。
- 自動化調查程序：開發可調查事件並向管理員報告相關資訊的自動化程序，以節省時間。
  - 實驗室：Amazon GuardDuty 實作

資源

相關文件：

- AWS Answers：集中式記錄
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- 入門：Amazon CloudWatch Logs
- 安全合作夥伴解決方案：記錄與監控
- 設定 Amazon GuardDuty

相關影片：

- 集中監控資源組態與合規
- 修補 Amazon GuardDuty 和 AWS Security Hub 發現結果
- 雲端中的威脅管理：Amazon GuardDuty 和 AWS Security Hub

相關範例：

- 實驗室：偵測控制的自動部署

SEC04-BP04 實作可採取行動的安全事件

建立傳送給團隊並能讓團隊據此採取行動的提醒。確保提醒包含讓團隊採取動作的相關資訊。對於您擁有的每個偵測機制，您也應該設置一套程序 (採取 執行手冊 或 程序手冊的形式) 以進行調查。例如，當您啟用 Amazon GuardDuty時，就會產生不同的 發現結果。每種發現結果類型都應該在 Runbook 有一個輸入項目，例如，如果發現 木馬程式 ，您的執行手冊會有簡單的指示，指示某人進行調查和修復。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 探索 AWS 服務可用的指標：針對您所使用的服務，探索透過 Amazon CloudWatch 提供的指標。
  - AWS 服務文件
  - 使用 Amazon CloudWatch 指標
- 設定 Amazon CloudWatch 警示。
  - 使用 Amazon CloudWatch 警示

資源

相關文件：

- Amazon CloudWatch
- Amazon EventBridge
- 安全合作夥伴解決方案：記錄與監控

相關影片：

- 集中監控資源組態與合規
- 修補 Amazon GuardDuty 和 AWS Security Hub 發現結果
- 雲端中的威脅管理：Amazon GuardDuty 和 AWS Security Hub

基礎設施保護

問題

- SEC 5 如何保護您的網路資源？

- SEC 6 您如何保護運算資源？

SEC 5 如何保護您的網路資源？

任何具有某種網路連線能力的工作負載，無論是網際網路或私有網路，都需要多層防禦來協助保護其不受外部和內部網路威脅的影響。

最佳實務

- SEC05-BP01 建立網路層
- SEC05-BP02 控制所有層級的流量
- SEC05-BP03 自動化網路保護
- SEC05-BP04 實作檢查和保護

SEC05-BP01 建立網路層

將共用連線能力需求的元件分組成許多層。例如：不需存取網際網路的虛擬私有雲端 (VPC) 中的資料庫叢集，應放置在沒有往返網際網路路由的子網路中。在沒有 VPC 的情況下操作的無伺服器工作負載中，與微型服務類似的分層和區隔可以達到相同的目標。

Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、Amazon Relational Database Service (Amazon RDS) 資料庫叢集等元件，以及共用連線能力要求的 AWS Lambda 函數可分成子網路所組成的層級。例如：不需存取網際網路的 VPC 中的 Amazon RDS 資料庫叢集，應放置在沒有往返網際網路路由的子網路中。此控制項的分層方法可減輕單一層組態錯誤所造成的影響，這可能會允許意外存取。對於 Lambda，您可以在 VPC 中執行函數，以利用 VPC 型控制。

對於可以包含數千個 VPC、AWS 帳戶和內部部署網路的網路連線，您應該使用 AWS Transit Gateway。它可作為中樞，控制流量在所有連線網路之間路由的方式，其作用就像輪輻。Amazon Virtual Private Cloud 和 AWS Transit Gateway 之間的流量仍保存在 AWS 私有網路，可減少外部威脅向量，例如分散式拒絕服務 (DDoS) 攻擊和常見入侵程式，例如 SQL 插入、跨網站指令碼、跨網站偽造請求或濫用不完整的身份驗證碼。AWS Transit Gateway 區域間對等也可為區域間的流量加密，不會有單點故障或頻寬瓶頸。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 在 VPC 中建立子網路：為每一層中建立子網路 (在包含多個可用區域的群組中)，並建立路由表的關聯以控制路由。

- VPC 和子網路
- 路由表

資源

相關文件：

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 安全性
- AWS WAF 入門

相關影片：

- 適用於許多 VPC 的 AWS Transit Gateway 參考架構
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 的應用程式加速和保護

相關範例：

- 實驗室：VPC 的自動部署

SEC05-BP02 控制所有層級的流量

建構網路拓撲時，您應該檢查每個元件的連線需求。例如，如果元件需要網際網路可存取性 (傳入和傳出)、連線至 VPC、邊緣服務和外部資料中心。

VPC 可讓您定義橫跨 AWS 區域的網路拓撲，使用您所設定的私有 IPv4 位址範圍，或 AWS 選取的 IPv6 位址範圍。您應該對傳入和傳出流量採用深度防禦方法的多個控制，包括使用安全群組 (狀態檢測防火牆)、網路 ACL、子網路和路由表。在 VPC 內，您可以在可用區域中建立子網路。每個子網都有一個關聯的路由表，定義路由規則，以管理子網路內流量所經過的路徑。您可以透過讓路由前往連接到 VPC 的網際網路或 NAT 閘道，或透過另一個 VPC 來定義網際網路可路由子網路。

當執行個體、Amazon Relational Database Service (Amazon RDS) 資料庫或其他服務在 VPC 內啟動時，每個網路介面都有自己的安全群組。此防火牆位於作業系統層之外，可用來定義允許傳入和傳出流量的規則。您還可以定義安全群組之間的關係。例如，資料庫層安全群組內的執行個體，藉由參照套用至所涉及執行個體的安全群組，僅接受來自應用程式層內執行個體的流量。除非您使用非 TCP 通訊

協定，否則應該不需要從網際網路直接存取 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體 (即使連接埠受安全群組限制)，無須使用負載平衡器或 CloudFront。這有助於防止透過作業系統或應用程式問題意外受到存取。子網路也可以連接著網路 ACL，做為無狀態的防火牆。您應該設定網路 ACL 以縮小層級之間允許的流量範圍，並請注意，需要同時定義傳入和傳出規則。

有些 AWS 服務會要求元件存取網際網路以進行 API 呼叫，其中 AWS API 端點 所在位置。其他 AWS 服務會使用 VPC 端點 (在您的 Amazon VPC 內)。許多 AWS 服務 (包括 Amazon S3 和 Amazon DynamoDB) 都支援 VPC 端點，而這項技術已廣泛應用於 AWS PrivateLink。我們建議您使用此方法安全地存取 AWS 服務、第三方服務，以及您在其他 VPC 中託管的專屬服務。AWS PrivateLink 上的所有網路流量都停留在全球 AWS 骨幹網路上，而且永遠不會周遊網際網路。連線只能由服務的消費者啟動，而不能由服務的供應商啟動。使用 AWS PrivateLink 進行外部服務存取可讓您建立沒有網際網路存取的氣隙 VPC，並協助保護您的 VPC 免於外部威脅向量的攻擊。第三方服務可以使用 AWS PrivateLink，允許其客戶透過私有 IP 地址從其 VPC 連線到服務。對於需要對網際網路進行對外連線的 VPC 資產，這些資產可以透過 AWS 受管 NAT 閘道、僅對外網際網路閘道或您建立和管理的 Web 代理進行僅對外 (單向)。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 控制 VPC 中的網路流量：實作 VPC 最佳實務來控制流量。
  - Amazon VPC 安全性
  - VPC 端點
  - Amazon VPC 安全群組
  - 網路 ACL
- 控制邊緣的流量：實作 Amazon CloudFront 等邊緣服務，以提供多一層的保護和其他功能。
  - Amazon CloudFront 使用案例
  - AWS Global Accelerator
  - AWS Web 應用程式防火牆 (AWS WAF)
  - Amazon Route 53
  - Amazon VPC 輸入路由
- 控制私有網路流量：實作可保護工作負載私有流量的服務。
  - Amazon VPC 對等互連
  - Amazon VPC 端點服務 (AWS PrivateLink)
  - Amazon VPC Transit Gateway

- AWS Direct Connect
- AWS 站點對站點 VPN
- AWS Client VPN
- Amazon S3 存取點

資源

相關文件：

- AWS Firewall Manager
- Amazon Inspector
- AWS WAF 入門

相關影片：

- 適用於許多 VPC 的 AWS Transit Gateway 參考架構
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 的應用程式加速和保護

相關範例：

- 實驗室：VPC 的自動部署

SEC05-BP03 自動化網路保護

自動化保護機制，以借助威脅情報和異常偵測提供自衛網路。例如，可以適應目前威脅並降低其影響的入侵偵測和預防工具。Web 應用程式防火牆即為可將網路保護自動化的範例；例如，使用 AWS WAF Security Automations 解決方案 (https://github.com/awslabs/aws-waf-security-automations) 以自動封鎖來自已知威脅者相關 IP 位址的請求。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 針對以網頁為基礎的流量進行自動保護：AWS 提供的解決方案使用 AWS CloudFormation 自動部署一組 AWS WAF 規則，專門用於篩選常見網頁式攻擊。使用者可從預先設定的保護功能中進行選擇，以定義 AWS WAF Web 存取控制清單 (web ACL) 中包含的規則。

  - AWS WAF 安全自動化
- 考慮 AWS Partner 解決方案：AWS 合作夥伴提供數百種領先業界的產品，這些產品與您內部部署環境中的現有控制項相當、相同或互相整合。這些產品可補充現有的 AWS 服務，讓您能夠在雲端和內部部署環境中部署全方位的安全架構，以及擁有更流暢的體驗。
  - 基礎設施安全

資源

相關文件：

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 安全性
- AWS WAF 入門

相關影片：

- 適用於許多 VPC 的 AWS Transit Gateway 參考架構
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 的應用程式加速和保護

相關範例：

- 實驗室：VPC 的自動部署

SEC05-BP04 實作檢查和保護

檢查和篩選每個層級的流量。為了檢查您的 VPC 組態並找出潛在的意外存取，您可以使用 VPC Network Access Analyzer。您可以指定您的網路存取要求，並識別未符合它們的潛在網路路徑。對於透過 HTTP 通訊協定進行交易的元件，Web 應用程式防火牆可協助防止常見的攻擊。 AWS WAF 是一種 Web 應用程式防火牆，可讓您監控和封鎖符合可設定規則的 HTTP 請求；這些請求會轉送到 Amazon API Gateway API、Amazon CloudFront 或 Application Load Balancer。若要開始使用 AWS WAF，您可以將 AWS 受管規則 結合自己的規則，或使用現有的 合作夥伴整合。

若要跨 AWS Organizations 管理 AWS WAF、AWS Shield Advanced 保護和 Amazon VPC 安全群組，您可以使用 AWS Firewall Manager。它可讓您集中設定和管理所有帳戶和應用程式的防火牆規則，使得共同規則的擴展強制執行更為輕鬆。它也可讓您使用 AWS Shield Advanced或可自動封鎖對

Web 應用程式不必要請求的 解決方案 ，快速地回應攻擊。Firewall Manager 也會使用 AWS Network Firewall。AWS Network Firewall 是一種託管服務，其會使用規則引擎，讓您精細控制有狀態和無狀態網路流量。它支援 Suricata 相容的 開放原始碼入侵防禦系統 (IPS) 規格，供規則用來協助保護您的工作負載。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 設定 Amazon GuardDuty：GuardDuty 是威脅偵測服務，可持續監控惡意活動和未經授權的行為，以保護 AWS 帳戶和工作負載。啟用 GuardDuty 並設定自動提醒。
  - Amazon GuardDuty
  - 實驗室：偵測控制的自動部署
- 設定虛擬私有雲端 (VPC) 流程日誌：VPC 流程日誌讓您可以擷取有關往返 VPC 網路界面 IP 流量的資訊。流程日誌資料可以發佈至 Amazon CloudWatch Logs 和 Amazon Simple Storage Service (Amazon S3)。建立流程日誌後，您可以在選定的目標位置擷取和檢視其資料。
- 考慮 VPC 流量鏡像化：流量鏡像化是一項 Amazon VPC 功能，可用來從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的彈性網路界面複製網路流量，然後將流量傳送到頻外安全與監控設備，以進行內容檢查、威脅監控和故障排除。
  - VPC 流量鏡像化

資源

相關文件：

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 安全性
- AWS WAF 入門

相關影片：

- 適用於許多 VPC 的 AWS Transit Gateway 參考架構
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 的應用程式加速和保護

相關範例：

- 實驗室：VPC 的自動部署

SEC 6 您如何保護運算資源？

工作負載中的運算資源需有多層防護，協助防範外部和內部威脅。運算資源包括 EC2 執行個體、容器、AWS Lambda 函數、資料庫服務、IoT 裝置等。

最佳實務

- SEC06-BP01 執行漏洞管理
- SEC06-BP02 減少受攻擊面
- SEC06-BP03 實作受管服務
- SEC06-BP04 自動化運算保護
- SEC06-BP05 讓人員能夠遠距離執行動作
- SEC06-BP06 驗證軟體完整性

SEC06-BP01 執行漏洞管理

經常掃描和修補程式碼、相依性和基礎設施中的漏洞，以協助防禦新的威脅。

從設定運算基礎設施開始，您可以使用 AWS CloudFormation，將建立和更新資源自動化。CloudFormation 可讓您使用 AWS 範例或撰寫自己的範例，建立以 YAML 或 JSON 撰寫的範本。這可讓您建立預設安全的基礎設施範本，而您可以使用 CloudFormation Guard驗證這些範本，以節省時間並減少組態錯誤的風險。例如，您可以建置基礎設施並部署應用程式，方法為使用持續交付搭配 AWS CodePipeline，以將建置、測試和發行自動化。

您負責對您的 AWS 資源進行修補程式管理，包括 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、Amazon Machine Image (AMI) 和許多其他運算資源。對於 Amazon EC2 執行個體，AWS Systems Manager 修補程式管理員可將管理執行個體在安全相關與其他類型方面的更新修補過程自動化。您可以使用修補程式管理員為作業系統和應用程式套用修補程式。(在 Windows Server 上，應用程式支援僅限於 Microsoft 應用程式的更新。) 您可以使用修補程式管理員，在 Windows 執行個體上安裝 Service Pack，並在 Linux 執行個體上執行次要版本升級。您可以按作業系統類型修補 Amazon EC2 執行個體機群或內部部署伺服器和虛擬機器 (VM)。這包括支援的 Windows Server、Amazon Linux、Amazon Linux 2、CentOS、Debian Server、Oracle Linux、Red Hat Enterprise Linux (RHEL)、SUSE Linux Enterprise Server (SLES) 和 Ubuntu Server 版本。您可以掃描執行個體以僅查看修補程式缺失報告，也可以掃描並自動安裝所有缺失的修補程式。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 設定 Amazon Inspector：Amazon Inspector 會測試 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的網路存取性，以及在這些執行個體上執行之應用程式的安全狀態。Amazon Inspector 會評估應用程式的暴露情況、漏洞和偏離最佳實務的程度。
  - 什麼是 Amazon Inspector？
- 掃描原始程式碼：掃描程式庫和相依性中的漏洞。
  - Amazon CodeGuru
  - OWASP：原始程式碼分析工具

資源

相關文件：

- AWS Systems Manager
- 將堡壘主機取代為 Amazon EC2 Systems Manager
- AWS Lambda 的安全概觀

相關影片：

- 在 Amazon EKS 上執行高安全的工作負載
- 保護無伺服器和容器服務的安全
- Amazon EC2 執行個體中繼資料服務的安全最佳實務

相關範例：

- 實驗室：Web 應用程式防火牆的自動部署

SEC06-BP02 減少受攻擊面

透過強化作業系統以及盡量減少使用中的元件、程式庫和外部消耗性服務，來減少意外存取。首先減少未使用的元件，無論它們是作業系統套件或應用程式 (適用於 Amazon Elastic Compute Cloud (Amazon EC2) 型工作負載) 或程式碼中的外部軟體模組 (適用於所有工作負載)。對於常見的作業系統和伺服器軟體，您可以找到許多強化和安全組態指南。例如，您可以從 Center for Internet Security 開始並反覆。

在 Amazon EC2 中，您可以建立自己的 Amazon Machine Image (AMI)，並已對其進行修補和強化，以協助您符合組織的特定安全要求。您在 AMI 上套用的修補程式和其他安全控制，在建立它們的時間點有效—它們不是動態的，除非您在啟動後進行修改，例如，使用 AWS Systems Manager 進行此修改。

您可以使用 EC2 Image Builder 簡化建置安全 AMI 的程序。EC2 Image Builder 會大幅地減少建立和維護黃金映像所需的工作量，而不會編寫和維護自動化。當軟體更新可用時，Image Builder 會自動產生新映像，而無需用戶手動啟動映像構置。EC2 Image Builder 可讓您在生產環境中使用您的映像，搭配 AWS 提供的測試和您自己的測試之前，輕鬆地驗證這些映像的功能和安全性。您也可以套用 AWS 提供的安全設定，以進一步保護您的映像，來符合內部安全準則。例，您可以使用 AWS 提供的範本，產生符合安全技術實作指南 (STIG) 標準的映像。

使用第三方靜態程式碼分析工具，您可以識別常見的安全問題，例如未檢查的函數輸入界限，以及適用的常見漏洞和披露 (CVE)。您可以使用 Amazon CodeGuru 取得支援的語言。相依性檢查工具也可以用來判斷您的程式碼連結的程式庫是否為最新版本、本身是否不使用 CVE，以及是否具有符合您軟體政策要求的授權條件。

使用 Amazon Inspector，您可以對已知 CVE 的執行個體執行組態評定、根據安全基準進行評估，和將缺陷通知自動化。Amazon Inspector 在生產執行個體上或建置管道中執行，並在結果出現時通知開發人員和工程師。您可以透過程式設計方式存取結果，並將您的團隊引導至待辦項目和錯誤追蹤系統。 EC2 Image Builder 可透過自動修補、AWS 提供的安全政策強制執行以及其他自訂項目來維護伺服器映像 (AMI)。使用容器時，會在您的建置管道中定期對照映像儲存庫執行 ECR 影像掃描 ，以在容器中尋找 CVE。

雖然 Amazon Inspector 和其他工具可以有效地識別現有的組態和任何 CVE，但還需要其他方法來測試應用程式層級的工作負載。 Fuzzing 是一種利用自動化尋找錯誤的知名方法，能將格式不正確的資料注入輸入欄位和應用程式的其他區域。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 強化作業系統：設定作業系統以符合最佳實務。
  - 保護 Amazon Linux
  - 保護 Microsoft Windows Server
- 強化容器化資源：設定容器化資源以符合安全最佳實務。
- 實作 AWS Lambda 最佳實務。
  - AWS Lambda 最佳實務

資源

相關文件：

- AWS Systems Manager
- 將堡壘主機取代為 Amazon EC2 Systems Manager
- AWS Lambda 的安全概觀

相關影片：

- 在 Amazon EKS 上執行高安全的工作負載
- 保護無伺服器和容器服務的安全
- Amazon EC2 執行個體中繼資料服務的安全最佳實務

相關範例：

- 實驗室：Web 應用程式防火牆的自動部署

SEC06-BP03 實作受管服務

實作管理資源的服務 (例如 Amazon Relational Database Service (Amazon RDS)、AWS Lambda 和 Amazon Elastic Container Service (Amazon ECS))，能為您減少共同責任模式中的安全維護任務。例如，Amazon RDS 可協助您設定、操作和擴展關聯式資料庫，並使諸如硬體佈建、資料庫設定、修補和備份等管理任務自動化。這表示您有更多空閒時間可以專心用 AWS Well-Architected Framework 中所述的其他方式來保護應用程式。Lambda 可讓您執行程式碼時無須佈建或管理伺服器，您可專注在程式碼層級的連線、叫用和安全等事項，無須擔心基礎設施或作業系統。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 探索可用的服務：探索、測試和實作可管理資源的服務，例如 Amazon RDS、AWS Lambda 和 Amazon ECS。

資源

相關文件：

- AWS 網站
- AWS Systems Manager
- 將堡壘主機取代為 Amazon EC2 Systems Manager
- AWS Lambda 的安全概觀

相關影片：

- 在 Amazon EKS 上執行高安全的工作負載
- 保護無伺服器和容器服務的安全
- Amazon EC2 執行個體中繼資料服務的安全最佳實務

相關範例：

- 實驗室：AWS Certificate Manager 請求公有憑證

SEC06-BP04 自動化運算保護

將保護性運算機制自動化，包括漏洞管理、減少攻擊面和資源管理。自動化可協助您將時間花在保護工作負載的其他層面，並降低人為錯誤的風險。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 自動化組態管理：透過使用組態管理服務或工具，來自動執行和驗證安全組態。
  - AWS Systems Manager
  - AWS CloudFormation
  - 實驗室：VPC 的自動部署
  - 實驗室：EC2 Web 應用程式的自動部署

- 自動修補 Amazon Elastic Compute Cloud (Amazon EC2)：AWS Systems Manager 修補程式管理員可將管理執行個體在安全相關與其他類型方面的更新修補過程自動化。您可以使用修補程式管理員為作業系統和應用程式套用修補程式。
  - AWS Systems Manager Patch Manager
  - 使用 AWS Systems Manager 自動化進行集中式多帳戶和多區域修補

- 實作入侵偵測和預防：實作入侵偵測和預防工具，以監控和阻止執行個體上的惡意活動。
- 考慮 AWS Partner 解決方案：AWS 合作夥伴提供數百種領先業界的產品，這些產品與您內部部署環境中的現有控制項相當、相同或互相整合。這些產品可補充現有的 AWS 服務，讓您能夠在雲端和內部部署環境中部署全方位的安全架構，以及擁有更流暢的體驗。
  - 基礎設施安全

資源

相關文件：

- AWS CloudFormation
- AWS Systems Manager
- AWS Systems Manager Patch Manager
- 使用 AWS Systems Manager 自動化進行集中式多帳戶和多區域修補
- 基礎設施安全
- 將堡壘主機取代為 Amazon EC2 Systems Manager
- AWS Lambda 的安全概觀

相關影片：

- 在 Amazon EKS 上執行高安全的工作負載
- 保護無伺服器和容器服務的安全
- Amazon EC2 執行個體中繼資料服務的安全最佳實務

相關範例：

- 實驗室：Web 應用程式防火牆的自動部署
- 實驗室：EC2 Web 應用程式的自動部署

SEC06-BP05 讓人員能夠遠距離執行動作

移除互動式存取功能可降低人為錯誤的風險，並降低手動設定或管理的可能性。例如，使用變更管理工作流程，以利用基礎設施即程式碼來部署 Amazon Elastic Compute Cloud (Amazon EC2)，然後 AWS Systems Manager 這類工具來管理 Amazon EC2 執行個體，而不允許直接存取或透過堡壘主機

存取。AWS Systems Manager 可以使用 自動化 工作流程，文件 (程序手冊) 和 執行命令等功能，自動化各種維護和部署任務。AWS CloudFormation 堆疊會從管道建立，而且可為您將基礎設施的部署和管理任務自動化，無須直接使用 AWS Management Console或 API。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 取代主控台存取：以 AWS Systems Manager Run Command 取代執行個體的主控台存取 (SSH 或 RDP)，以自動化管理任務。

- AWS Systems Manager Run Command

資源

相關文件：

- AWS Systems Manager
- AWS Systems Manager Run Command
- 將堡壘主機取代為 Amazon EC2 Systems Manager
- AWS Lambda 的安全概觀

相關影片：

- 在 Amazon EKS 上執行高安全的工作負載
- 保護無伺服器和容器服務的安全
- Amazon EC2 執行個體中繼資料服務的安全最佳實務

相關範例：

- 實驗室：Web 應用程式防火牆的自動部署

SEC06-BP06 驗證軟體完整性

實作機制 (例如程式碼簽署) 以驗證工作負載中使用的軟體、程式碼和程式庫，確保它們來自信任的來源且未遭到篡改。例如，您應該驗證二進位程式碼和指令碼的程式碼簽署憑證，以確認作者，並確保自

作者建立後並未遭到篡改。AWS Signer 可以集中管理程式碼簽署生命週期 (包括簽署認證和公有和私有金鑰) 來協助確保程式碼的信任和完整性。您可以了解如何使用進階模式和最佳實務，搭配下列項目進行程式碼簽署： AWS Lambda。此外，相較於供應商的檢查總和，您所下載軟體的檢查總和有助於確保該軟體並未遭到竄改。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 調查機制：程式碼簽署是用來驗證軟體完整性的一種機制。
  - NIST：程式碼簽署的安全考量

資源

相關文件：

- AWS Signer
- 新增功能 – 程式碼簽署，AWS Lambda 的信任和完整性控制

資料保護

問題

- SEC 7 您如何分類資料？
- SEC 8 您如何保護靜態資料？
- SEC 9 您如何保護傳輸中資料？

SEC 7 您如何分類資料？

資料分類可讓您根據關鍵性和敏感度將資料分類，協助判定適當的保護和保留控制。

最佳實務

- SEC07-BP01 識別工作負載內的資料
- SEC07-BP02 定義資料保護控制
- SEC07-BP03 自動識別和分類
- SEC07-BP04 定義資料生命週期管理

SEC07-BP01 識別工作負載內的資料

您需要了解工作負載所處理的資料類型和類別、相關的商業程序、資料擁有者、適用的法律和合規要求、存放的位置，以及因而需要強制執行的控制項。這可能包括分類以指出資料是否打算供公開取得；資料是否僅供內部使用，例如客戶的個人識別資訊 (PII)；資料是否用於更受限制的存取，例如智慧財產、依法特權或標示為敏感資料等等。透過仔細管理適當的資料分類系統，並搭配每個工作負載的保護等級要求，您可以規劃適合資料的控制項和存取或保護等級。例如，公開的內容可供任何人存取，然而重要內容則以受保護的方式進行加密和儲存，需要授權取得金鑰才能將內容解密。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 考慮使用 Amazon Macie 探索資料：Macie 可辨識個人識別資訊 (PII) 或智慧財產等敏感資料。
  - Amazon Macie

資源

相關文件：

- Amazon Macie
- 資料分類白皮書
- Amazon Macie 入門

相關影片：

- 介紹新的 Amazon Macie

SEC07-BP02 定義資料保護控制

根據資料的分類層級保護資料。例如：使用相關建議來保護歸類為公有的資料，同時實作額外的控制以保護敏感資料。

使用資源標籤、根據敏感度 (也可能適用於警告、群體或關注的社群)、IAM 政策、AWS Organizations SCP、AWS Key Management Service (AWS KMS) 和 AWS CloudHSM 將 AWS 帳戶做出區隔，您可以定義和實作資料分類和加密保護的政策。例如，假設您有一個專案用到存放高度關鍵資料的 S3 儲存貯體，或處理機密資料的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，則可以使用 `Project=ABC` 標籤進行標記。只有您的直屬團隊知道專案代號的意義，同時也可作為使用屬性型存取控制的方式。您可以透過金鑰政策和授權，定義對 AWS KMS 加密金鑰的存取等級，以確保

僅限相應的服務能透過安全機制存取敏感內容。如果要根據標籤做出授權決策，您應該確保在 AWS Organizations 中使用標籤政策，正確地定義標籤上的許可。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 定義您的資料識別和分類結構描述：對資料進行識別和分類，評估您儲存的資料的潛在影響和類型，以及誰可以存取它。
  - AWS 文件
- 探索可用的 AWS 控制：針對您目前正在使用或計劃使用的 AWS 服務探索安全控制。許多服務在其文件中皆有安全性區段。
  - AWS 文件
- 識別 AWS 合規資源：識別 AWS 可用於協助的資源。
  - https://aws.amazon.com/compliance/

資源

相關文件：

- AWS 文件
- 資料分類白皮書
- Amazon Macie 入門
- 缺少文字

相關影片：

- 介紹新的 Amazon Macie

SEC07-BP03 自動識別和分類

將資料的識別和分類自動化，可協助您實作正確的控制方法。針對此目的使用自動化而非由人員直接存取，可降低人為錯誤和外洩的風險。您應該使用 Amazon Macie這類工具進行評估，該工具會使用機器學習自動探索、分類和保護 AWS 中的敏感資料。Amazon Macie 可辨識個人識別資訊 (PII) 或智慧財產權等敏感資料，並提供儀表板和提醒，讓您深入了解資料的存取或移動方式。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用 Amazon Simple Storage Service (Amazon S3) 庫存：Amazon S3 庫存是可用來稽核和報告物件複寫與加密狀態的其中一項工具。
  - Amazon S3 庫存
- 考慮 Amazon Macie：Amazon Macie 使用機器學習來自動發現和分類存放在 Amazon S3 中的資料。
  - Amazon Macie

資源

相關文件：

- Amazon Macie
- Amazon S3 庫存
- 資料分類白皮書
- Amazon Macie 入門

相關影片：

- 介紹新的 Amazon Macie

SEC07-BP04 定義資料生命週期管理

您已定義的生命週期策略應以敏感性等級以及法律和組織的要求作為依據。您應考慮保留資料的期間、資料銷毀程序、資料存取管理、資料轉換和資料共享等方面。在選擇資料分類方法時，應在使用性與存取之間取得平衡。也應採納存取權的多個等級，並且耐心依照各等級分別實作安全又兼顧使用方便的方法。一律使用深度防禦方法，並減少為了轉換、刪除或複製資料，對資料與機制的手動存取。例如，要求使用者對應用程式進行強式驗證，並給予應用程式 (而非使用者) 必要的存取許可，以執行遠距離動作。此外，確保使用者來自受信任的網路路徑，並要求存取解密金鑰。應使用儀表板或自動報告之類的工具為使用者提供來自資料的資訊，而不是讓使用者直接存取資料。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 識別資料類型：識別您在工作負載中存放或處理的資料類型。該資料可以是文字、影像、二進位資料庫等。

資源

相關文件：

- 資料分類白皮書
- Amazon Macie 入門

相關影片：

- 介紹新的 Amazon Macie

SEC 8 您如何保護靜態資料？

實作多個控制來保護您的靜態資料，以降低未經授權的存取或不當處理的風險。

最佳實務

- SEC08-BP01 實作安全金鑰管理
- SEC08-BP02 強制靜態加密
- SEC08-BP03 將靜態資料保護自動化
- SEC08-BP04 強制執行存取控制
- SEC08-BP05 使用限制人員存取資料的機制

SEC08-BP01 實作安全金鑰管理

透過定義加密方法 (包含金鑰的儲存、輪換和存取控制)，可以協助保護您的內容，免於未經授權的使用者和不必要的向授權使用者透露。AWS Key Management Service (AWS KMS) 可協助您管理加密金鑰， 並與許多 AWS 服務整合。此服務為您的 AWS KMS 金鑰提供耐用、安全和冗餘的儲存。您可以定義金鑰別名以及金鑰層級的政策。這些政策可幫助您定義金鑰管理員和金鑰使用者。此外，AWS CloudHSM 是雲端硬體安全模組 (HSM)，讓您能夠在 AWS 雲端上輕鬆產生和使用自己的加密金鑰。透過使用 FIPS 140-2 3 級驗證的 HSM，它可以幫助您滿足公司、合同和法規對資料安全的合規要求。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 實作 AWS KMS：AWS KMS 可讓您輕鬆建立和管理金鑰，並控制多種 AWS 服務和應用程式中的加密使用。AWS KMS 是一種安全且具彈性的服務，使用經過 FIPS 140-2 驗證的硬體安全模組來保護您的金鑰。
  - 入門：AWS Key Management Service (AWS KMS)
- 考慮 AWS 加密開發套件：當您的應用程式需要在用戶端對資料進行加密時，可使用整合 AWS KMS 的 AWS 加密開發套件。
  - AWS 加密開發套件

資源

相關文件：

- AWS Key Management Service
- AWS 加密服務和工具
- 入門：AWS Key Management Service (AWS KMS)
- 使用加密保護 Amazon S3 資料

相關影片：

- 在 AWS 中加密如何運作
- 保護 AWS 上的區塊儲存安全

SEC08-BP02 強制靜態加密

您應該確保存放資料的唯一方法是使用加密。AWS Key Management Service (AWS KMS) 與許多 AWS 服務無縫整合，讓您更輕鬆地為所有靜態資料加密。例如，在 Amazon Simple Storage Service (Amazon S3) 中，您可以在儲存貯體上設定 預設加密， 以便將所有新物件自動加密。此外，Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon S3 透過設定預設加密來支援強制加密。您可以使用 AWS Config 規則 ，自動檢查您是否正在將加密用於，例如， Amazon Elastic Block Store (Amazon EBS) 磁碟區， Amazon Relational Database Service (Amazon RDS) 執行個體和 Amazon S3 儲存貯體。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 強制對 Amazon Simple Storage Service (Amazon S3) 執行靜態加密：實作 Amazon S3 儲存貯體預設加密。
  - 如何針對 S3 儲存貯體啟用預設加密？
- 使用 AWS Secrets Manager：Secrets Manager 是一項 AWS 服務，可讓您更輕鬆地管理機密。機密可以是資料庫登入資料、密碼、第三方 API 金鑰，甚至是任意文字。
  - AWS Secrets Manager
- 設定新 EBS 磁碟區的預設加密：使用 AWS 提供的預設金鑰或您自行建立的金鑰，指定您希望以加密形式建立所有新的 EBS 磁碟區。
  - EBS 磁碟區的預設加密
- 設定加密的 Amazon Machine Images (AMI)：複製已啟用加密的現有 AMI 會自動加密根磁碟區和快照。
  - 帶有加密快照的 AMI
- 設定 Amazon Relational Database Service (Amazon RDS)：透過啟用加密選項，為您的 Amazon RDS 資料庫叢集和靜態快照設定啟用加密。
  - 加密 Amazon RDS 資源
- 在其他 AWS 服務中設定加密：對於您使用的 AWS 服務，請決定加密功能。
  - AWS 文件

資源

相關文件：

- 帶有加密快照的 AMI
- AWS 加密工具
- AWS 文件
- AWS 加密開發套件
- AWS KMS 加密詳細資訊白皮書
- AWS Key Management Service
- AWS Secrets Manager
- AWS 加密服務和工具
- Amazon EBS 加密

- EBS 磁碟區的預設加密
- 加密 Amazon RDS 資源
- 如何針對 S3 儲存貯體啟用預設加密？
- 使用加密保護 Amazon S3 資料

相關影片：

- 在 AWS 中加密如何運作
- 保護 AWS 上的區塊儲存安全

SEC08-BP03 將靜態資料保護自動化

使用自動化工具以持續驗證並強制執行靜態資料控制，例如，驗證以確認只有加密的儲存資源。您 可以 使用 AWS Config 規則資料目錄中。AWS Security Hub 也可以透過符合安全標準的自動化檢查來驗證數種不同的控制。此外，您的 AWS Config 規則 可以自動 修復不合規的資源。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

靜態資料 代表您在工作負載中的任何期間，保留在非揮發性儲存體中的任何資料。其中包括：長期存放資料的區塊儲存體、物件儲存體、資料庫、封存、IoT 裝置和任何其他儲存媒介。實作加密和適當的存取控制，保護靜態資料能將未經授權存取的風險降低。

強制靜態加密：您應該確保存放資料的唯一方法是使用加密。AWS KMS 與許多 AWS 服務無縫整合，讓您更輕鬆地為所有靜態資料加密。例如，在 Amazon Simple Storage Service (Amazon S3) 中，您可以在儲存貯體上設定 預設加密， 以便將所有新物件自動加密。此外，Amazon EC2 和 Amazon S3 透過設定預設加密來支援強制加密。您可以使用 AWS 受管 Config 規則 ，自動檢查您是否正在將加密用於，例如， EBS 磁碟區，Amazon Relational Database Service (Amazon RDS) 執行個體和 Amazon S3 儲存貯體。

資源

相關文件：

- AWS 加密工具
- AWS 加密開發套件

相關影片：

- 在 AWS 中加密如何運作
- 保護 AWS 上的區塊儲存安全

SEC08-BP04 強制執行存取控制

強制執行最低權限存取控制和機制 (包括備份、隔離和版本控制)，以保護靜態資料。防止操作員授予您資料的公有存取權。

不同的控制包括存取 (使用最低權限)、備份 (請參閱 可靠性白皮書)、隔離，以及和版本控制，全都有助於保護您的靜態資料。對資料的存取應使用本白皮書稍早涵蓋的偵測機制進行稽核，包括 CloudTrail 和服務層級日誌 (例如 Amazon Simple Storage Service (Amazon S3) 存取日誌)。您應該清查哪些資料可公開存取，並規劃如何隨著時間減少可用的資料量。Amazon S3 Glacier 文件庫鎖定和 Amazon S3 物件鎖定提供強制存取控制的功能，一旦文件庫政策被合規選項鎖定，在鎖定過期之前，就連根使用者也無法變更。此機制符合 SEC、CFTC 和 FINRA 的書籍和記錄管理要求。如需詳細資訊，請參閱 此白皮書。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 強制執行存取控制：強制執行最低權限存取控制，包括對加密金鑰的存取。
  - 管理對 Amazon S3 資源的存取許可的簡介
- 根據不同的分類層級分離資料：針對由 AWS Organizations 管理的資料分類層級，使用不同的 AWS 帳戶。
  - AWS Organizations
- 審查 AWS KMS 政策：審查 AWS KMS 政策中授予的存取層級。
  - 管理對您的 AWS KMS 資源的存取概觀
- 審查 Amazon S3 儲存貯體和物件權限：定期審查 Amazon S3 儲存貯體政策中授予的存取層級。最佳實務是不要設定公開可讀取或可寫入的儲存貯體。考慮使用 AWS Config 偵測公開可用的儲存貯體，以及使用 Amazon CloudFront 從 Amazon S3 提供內容。
  - AWS Config 規則
  - Amazon S3 + Amazon CloudFront：雲端中進行的比對
- 啟用 Amazon S3 版本控制和物件鎖定。
  - 使用版本控制

  - 使用 Amazon S3 物件鎖定來鎖定物件
- 使用 Amazon S3 庫存：Amazon S3 庫存是可用來稽核和報告物件複寫與加密狀態的其中一項工具。
  - Amazon S3 庫存
- 審查 Amazon EBS 和 AMI 共用許可：共用許可可以允許將映像和磁碟區分享到工作負載外部的 AWS 帳戶。
  - 共用 Amazon EBS 快照
  - 分享 AMI

資源

相關文件：

- AWS KMS 加密詳細資訊白皮書

相關影片：

- 保護 AWS 上的區塊儲存安全

SEC08-BP05 使用限制人員存取資料的機制

在正常運作情況下，讓所有使用者遠離直接存取敏感資料和系統的權限。例如，使用變更管理工作流程來使用工具管理 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，而不允許直接存取或堡壘主機存取。這可使用 AWS Systems Manager Automation來達成，其使用 自動化文件 ，內有您用來執行任務的步驟。這些文件可以存放在原始檔控制中、在執行前接受對等審查，並經過徹底測試，以盡量降低與 shell 存取相比的風險。商業使用者可以擁有儀表板，而不是直接存取資料存放區來執行查詢。未使用 CI/CD 管道時，請判斷需要哪些控制和程序，才能充分提供一般停用時的緊急存取機制。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 實作限制人員存取資料的機制：這些機制包括使用 Amazon QuickSight 等儀表板向使用者顯示資料，而不是直接查詢。
  - Amazon QuickSight
- 自動化組態管理：透過使用組態管理服務或工具，在遠距離執行動作，並自動執行和驗證安全組態。避免使用堡壘主機或直接存取 EC2 執行個體。

- AWS Systems Manager
- AWS CloudFormation
- AWS 上 AWS CloudFormation 範本的 CI/CD 管道

資源

相關文件：

- AWS KMS 加密詳細資訊白皮書

相關影片：

- 在 AWS 中加密如何運作
- 保護 AWS 上的區塊儲存安全

SEC 9 您如何保護傳輸中資料？

實作多個控制以保護傳輸中的資料，減少未經授權的存取或遺失的風險。

最佳實務

- SEC09-BP01 實作安全金鑰和憑證管理
- SEC09-BP02 強制執行傳輸中加密
- SEC09-BP03 自動偵測意外的資料存取
- SEC09-BP04 驗證網路通訊

SEC09-BP01 實作安全金鑰和憑證管理

安全地存放加密金鑰和憑證，並依適當的時間間隔，以嚴格的存取控制進行輪換。達成此目標的最佳方式是使用受管服務，例如 AWS Certificate Manager (ACM)。它可讓您輕鬆佈建、管理和部署可與 AWS 服務和您的內部連線資源搭配使用的公有和私有 Transport Layer Security (TLS) 憑證。TLS 憑證可用來保護網路通訊，和建立網際網路中的網站和私有網路中資源的身份。ACM 與 AWS 資源整合，例如 Elastic Load Balancer (ELB)、AWS 分發以及 API Gateway 上的 API，也會處理自動憑證更新。如果您使用 ACM 部署私有根 CA，則它可以提供憑證和私有金鑰兩者，用於 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、容器內等。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 實作安全金鑰和憑證管理：實作您定義的安全金鑰和憑證管理解決方案。
  - AWS Certificate Manager
  - 如何在 AWS 中託管和管理整個私有憑證基礎架構
- 實作安全協定：使用提供身份驗證和機密性的安全協定 (例如 Transport Layer Security (TLS) 或 IPsec) 來減少資料竄改或遺失的風險。請查看 AWS 文件，了解與您使用的服務相關的協定和安全性。

資源

相關文件：

- AWS 文件

SEC09-BP02 強制執行傳輸中加密

根據適當的標準和建議強制執行已定義的加密要求，協助您符合組織、法律和合規上的要求。AWS 服務提供使用 TLS 進行通訊的 HTTPS 端點，從而在與 API 通訊時提供傳輸中加密。不安全的通訊協定 (例如 HTTP) 可以在 VPC 中透過使用安全群組加以稽核和封鎖。HTTP 請求也可以 自動重新導向至 HTTPS 在 Amazon CloudFront 或 Application Load Balancer 中。您可以完全控制您的運算資源，以在各個服務中實作傳輸中加密。此外，還可以從外部網路使用 VPN 連線功能進入 VPC，實現流量加密。如果您有特殊需求，AWS Marketplace 備有第三方解決方案。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 強制執行傳輸中加密：您定義的加密要求應符合最新標準和最佳實務，並僅允許採用安全協定。例如，將安全群組設定為僅允許 HTTPS 協定到 Application Load Balancer 或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。
- 在邊緣服務中設定安全協定：使用 Amazon CloudFront 和必要的加密法設定 HTTPS。
  - 搭配 CloudFront 使用 HTTPS
- 使用 VPN 進行外部連線：考慮使用 IPsec VPN 虛擬私有網路 (VPN)，保護點對點或網路對網路連線，以提供資料隱私和完整性。
  - VPN 連線
- 在負載平衡器中設定安全協定：啟用 HTTPS 接聽程式以保護與負載平衡器的連線。

  - Application Load Balancer 的 HTTPS 接聽程式
- 為執行個體設定安全協定：考慮在執行個體上設定 HTTPS 加密。
  - 教學：在 Amazon Linux 2 上設定 Apache Web 伺服器以使用 SSL/TLS
- 在 Amazon Relational Database Service (Amazon RDS) 中設定安全協定：使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來加密與資料庫執行個體的連線。
  - 使用 SSL 加密與資料庫執行個體的連線
- 在 Amazon Redshift 中設定安全協定：將您的叢集設定為要求 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 連線。
  - 設定連線的安全性選項
- 在其他 AWS 服務中設定安全協定：對於您使用的 AWS 服務，請決定傳輸中加密功能。

資源

相關文件：

- AWS 文件

SEC09-BP03 自動偵測意外的資料存取

使用 Amazon GuardDuty 這類工具，自動偵測可疑活動或將資料移到所定義邊界之外的嘗試。例如，GuardDuty 可以偵測異常的 Amazon Simple Storage Service (Amazon S3) 讀取活動，發現結果為 Exfiltration:S3/AnomalousBehavior。除了 GuardDuty，擷取網路流量資訊的 Amazon VPC 流程日誌還可與 Amazon EventBridge 搭配使用，以觸發異常連線偵測，其中成功和拒絕兩者皆包含在內。Amazon S3 Access Analyzer 可協助評估您的 Amazon S3 儲存貯體中誰可以存取哪些資料。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 自動偵測意外的資料存取：使用工具或偵測機制自動偵測將資料移出定義邊界的嘗試；例如，偵測將資料複製到無法識別之主機的資料庫系統。
  - VPC Flow Logs
- 考慮 Amazon Macie：Amazon Macie 是一項全受管的資料安全和資料隱私服務，它使用機器學習和模式比對來探索和保護 AWS 中的敏感資料。
  - Amazon Macie

資源

相關文件：

- VPC Flow Logs
- Amazon Macie

SEC09-BP04 驗證網路通訊

使用支援身份驗證的通訊協定 (Transport Layer Security (TLS) 或 IPsec) 來驗證通訊的身分。

使用支援身份驗證的網路通訊協定，可讓雙方之間建立信任。此法可增加通訊協定中使用的加密，降低通訊遭到更改或攔截的風險。實作身份驗證的常見通訊協定包括許多 AWS 服務中使用的 Transport Layer Security (TLS)，以及在 AWS Virtual Private Network (AWS VPN) 中使用的 IPsec。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 實作安全協定：使用提供身份驗證和機密性的安全協定 (例如 T TLS 或 IPsec) 來減少資料竄改或遺失的風險。請參閱 AWS 文件 ，了解與您使用的服務相關的協定和安全性。

資源

相關文件：

- AWS 文件

事故回應

問題

- SEC 10 您如何預估、回應事件以及從事件中復原？

SEC 10 您如何預估、回應事件以及從事件中復原？

準備對於及時且有效的調查、回應事件以及從事件中復原至關重要，有助於將對組織的干擾降到最低。

最佳實務

- SEC10-BP01 確定關鍵人員和外部資源

- SEC10-BP02 制定事件管理計劃
- SEC10-BP03 準備鑑識功能
- SEC10-BP04 自動化遏制能力
- SEC10-BP05 預先佈建存取權
- SEC10-BP06 預先部署工具
- SEC10-BP07 執行演練日

SEC10-BP01 確定關鍵人員和外部資源

確定可以幫助您的組織回應事件的內部和外部人員、資源及法律義務。

當您在雲端定義回應事件的方法時，與其他團隊 (例如您的法律顧問、領導階層、業務利害關係人、AWS Support Services 等等) 共同合作時，您必須識別關鍵人員、利害關係人和相關聯絡人。為了減少相依性並縮短回應時間，請確保您的團隊、專業安全團隊和回應人員受過您所使用服務的教育訓練，並有機會實際操作。

我們鼓勵您找出可提供外部專業知識和不同觀點，為您增強回應能力的外部 AWS 安全合作夥伴。您信任的安全合作夥伴可協助您識別您可能不熟悉的潛在風險或威脅。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 識別組織中的關鍵人員：維護人員聯絡清單，將組織中參與事故回應和復原的人員納入其中。
- 識別外部模式：如有必要，可雇用外部合作夥伴，以幫助您應對事件並從中復原。

資源

相關文件：

- AWS 事故回應指南

相關影片：

- 準備和回應 AWS 環境中的安全事故

相關範例：

SEC10-BP02 制定事件管理計劃

建立計劃以協助您回應、在事件期間進行溝通，以及從事件中復原。例如您可以從工作負載和組織最可能發生的情境，開始建立事件回應計畫。包括您在內部和外部進行溝通和向上呈報的流程。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

事件管理計劃對於回應、減輕安全事件所造成潛在影響並從中復原而言至關重要。事件管理計劃是結構清晰的程序，可及時找出、修復和回應安全事件。

雲端有許多在內部部署環境中所見的相同營運角色和需求。建立事件管理計劃時，您必須將與業務成果和合規需求最相符的應變及復原策略納入考量。例如，如果您在 AWS 中運作的工作負載符合美國的 FedRAMP，那麼遵循 NIST SP 800-61 電腦安全處理指南便很有用。同樣地，當運行帶有歐洲 PII (個人身分識別資訊) 資料的工作負載時，請考量以下情境，例如如何保護和回應與 歐盟一般資料保護規範 (GDPR) 中所要求之資料落地的相關問題。

為在 AWS 中運行的工作負載建立事件管理計劃時，請從 AWS 共同的責任模型開始，以便建立事件應變的深度防禦方法。在此模型中，AWS 會管理雲端的安全，但維護雲端的安全是您的責任。此表示您保有控制權，並對您選擇實作的安全控制項負責。AWS AWS 安全事件應變指南 詳細說明在建立以雲端為中心的事件管理計劃時的重要概念和基礎指引。

有效的事件管理計劃必須經過持續的反覆測試，以與您的雲端營運目標保持同步。在您建立和制定事件管理計劃時，請考慮使用以下詳述的實作計劃。

- 針對事件應變提供指導和訓練： 當發生偏差，偏離您定義的基準時 (例如，錯誤的部署或錯誤的設定)，您可能需要回應和調查。為了順利回應和調查，您必須了解在 AWS 環境中可以使用哪些控制項和功能，來回應安全事件，以及需要考量哪些程序，以便為參與事件應變的雲端團隊提供指導和訓練，並確保他們做好準備。
  - 程序手冊 和 執行手冊 都是有效的機制，可讓您以一致的方式，來訓練事件的回應方式。從建立初步清單開始，此清單會列出在事件應變期間頻繁執行的程序，並隨著您學習或使用新的程序持續反覆測試。
  - 透過排定的演練日傳播 程序手冊和執行手冊。在演練日期間，在受控的環境中模擬事件應變，讓團隊可以回想如何回應，並確認事件應變的參與團隊是否都熟知工作流程。審查模擬事件的成果，來找出待改善的地方，並判斷是否需要進一步的培訓或額外的工具。
  - 應將安全性視為每個人的職責。透過將平常運行工作負載的所有人員都納入，來建立對事件管理程序的集體知識。這包括您業務的各個方面；營運、測試、開發、安全性、業務營運和業務領導者。

- 記錄事件管理計劃： 記錄要記錄、據以採取行動和溝通進度的工具及程序，並提供與作用中事件有關的通知。事件管理計劃的目標是確認平常的營運都能盡快恢復，將對業務造成的影響降到最低，以及所有相關的人員都能了解情形。事件的例子包含 (但不限於) 網路連線的遺失或效能降低、無回應的程序或 API、未執行的排定任務 (例如，修補失敗)、應用程式資料或服務不可用、因為安全性事件而造成的意外服務中斷、憑證洩漏或設定錯誤。
  - 找出負責解決事件的主要擁有者，例如工作負載的擁有者。清楚地指導誰將執行事件以及如何處理溝通。當事件解決程序中參與的當事人不只一位時，例如外部廠商，請考慮建立 責任 (RACI) 矩陣，此矩陣會詳述事件解決所需的各種團隊或人員的角色和責任。

    RACI 矩陣詳述以下內容：

    - R： 負責任的 當事人會負責完成任務。
    - A： 專責的 當事人或利害關係人有最終的權力，可決定特定任務是否成功完成。
    - C： 徵求意見的 被諮詢當事人，其身分通常是各領域的專家。
    - I： 收到進度通知的 當事人，通常只在任務完成或交付成果時才會接獲通知。
- 將事件分類： 根據嚴重性和影響分數定義及分類事件，可為分類和解決事件提供有條理的方法。下列建議描述的是 對解決方案造成影響的緊急矩陣， 可將事件造成的影響量化。例如，影響輕微、較不緊急的事件會被視為低嚴重性事件。
  - 高 (H)： 您的業務受到嚴重的影響。與 AWS 資源相關的應用程式重要功能無法使用。為對生產系統造成影響的最關鍵事件而保留。事件帶來的影響會隨著修復措施的時效性而快速提高。
  - 中 (M)： 與 AWS 資源相關的業務服務或應用程式受到中度的影響並在降級的狀態下運作。有助於服務水準目標 (SLO) 的應用程式在服務水準協議 (SLA) 限制中受到影響。系統可以在能力下降的情況下執行，而不會產生太大的財務或聲譽影響。
  - 低 (L)： 與 AWS 資源相關的業務服務或應用程式的非關鍵功能受到影響。系統可以在能力下降的情況下執行，並產生最輕微的財務或聲譽影響。
- 將安全控制項標準化： 將安全控制項標準化的目標是為了實現營運成果的一致性、可追蹤性和可重複性。推動對事件應變至關重要之活動的標準化，例如：
  - 身分和存取管理： 建立機制，來控制資料的存取和同時管理人類與機器身分的權限。將您專屬的身分和存取管理擴展至雲端，使用聯合身分安全搭配單一登入和以角色為基礎的權限，來優化存取管理作業。如需將存取管理作業標準化的最佳實務建議和改善計劃，請參閱 安全支柱白皮書 的身分和存取管理一節。
  - 漏洞管理： 建立機制，來找出在 AWS 環境中很可能遭攻擊者用來入侵和濫用系統的漏洞。實作預防性和偵測性控制作為安全機制，以回應並減輕安全事件的潛在影響。在基礎設施建立和應用程式交付生命週期中，將威脅建模等程序標準化。

  - 組態管理： 定義標準組態和自動化程序，以便在 AWS 雲端 中部署資源。同時將基礎設施和資源佈建標準化，有助於減輕因錯誤部署所造成的錯誤設定或意外人為設定錯誤的風險。請參閱 卓越營運支柱白皮書 設計原則一節，來了解實作此控制項的指引和改善計劃。
  - 稽核控制項的記錄和監控： 實作機制，來監控資源的故障、效能降低和安全性問題。將這些控制項標準化也能提供系統中所發生之活動的稽核軌跡，進而協助及時分類和修復問題。SEC04 下的最佳實務 (「您如何偵測和調查安全事件？」) 提供實作此控制項的指引。
- 使用自動化： 自動化可讓您大規模及時解決事件。AWS 提供數種服務來在事件應變策略的情境中進行自動化。專注於在自動化和人工介入之間尋找適當的平衡。當您在程序手冊和執行手冊中建立事件應變時，請將可重複的步驟自動化。使用 AWS Systems Manager Incident Manager 之類的 AWS 服務來 快速解決 IT 事件。使用 開發人員工具 來提供版本控制和自動化 Amazon Machine Images (AMI) 與基礎設施即程式碼 (IaC) 部署，而不需人工介入。在適用的情況下，使用 Amazon GuardDuty、Amazon Inspector、AWS Security Hub、AWS Config 和 Amazon Macie 之類的受管服務，將偵測和合規評估自動化。使用 Amazon DevOps Guru 之類的機器學習優化偵測功能，在異常營運模式問題發生前加以偵測。
- 執行根本原因分析以及汲取經驗教訓： 實作機制以便於事件後應變審查中汲取經驗教訓。當事件的根本原因揭露更大的缺陷、設計缺點、設定錯誤或再發的可能性時，就會將該事件分類為問題。在這類案例中，分析和解決問題來將對正常營運的中斷降到最低。

資源

相關文件：

- AWS 安全事件應變指南
- NIST：電腦安全事件處理指南

相關影片：

- 將 AWS 中的事件應變和鑑識自動化
- 執行手冊、事件報告和事件應變的 DIY 指南
- 準備和回應 AWS 環境中的安全事件

相關範例：

- 實驗室：使用 Jupyter 的事件應變程序手冊 - AWS IAM
- 實驗室：使用 AWS 主控台和 CLI 來應變事件

SEC10-BP03 準備鑑識功能

了解鑑識調查何時以及如何適合您的回應計劃，對於您的事故回應者而言很重要。您的組織應定義收集的證據以及過程中使用的工具。識別和準備適合的鑑識調查功能，包括外部專家、工具和自動化。您應該預先做出的關鍵決定是您是否將從即時系統中收集資料。如果系統關閉電源或重新啟動，某些資料 (例如易消逝性記憶體的內容或作用中的網路連線) 將會遺失。

您的回應團隊可以結合 AWS Systems Manager Amazon EventBridge 和 AWS Lambda 等工具，在作業系統和 VPC 流量鏡像內自動運行鑑識工具，以取得網路封包擷取，來收集非持久性證據。使用自訂的鑑識工作站和可供回應者存取的工具，在專用安全帳戶中進行其他活動，例如日誌分析或分析磁碟映像。

定期將相關日誌傳送到提供高耐久性和完整性的資料存放區。回應者應該可以存取這些日誌。AWS 會提供數種工具，讓日誌調查更容昉進行，例如 Amazon Athena、Amazon OpenSearch Service (OpenSearch Service) 和 Amazon CloudWatch Logs Insights。此外，還會使用 Amazon Simple Storage Service (Amazon S3) 物件鎖定，安全地保留證據。此服務遵循 WORM (一次寫入-多次讀取) 模型，並防止物件在定義的期間遭到刪除或覆寫。由於鑑識調查技術需要專業培訓，您可能需要聘請外部專家。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 識別鑑識能力：研究組織的鑑識調查能力、可用的工具以及外部專家。
- 自動化事故回應和鑑識

資源

相關文件：

- 如何在 AWS 中將鑑識磁碟收集自動化

SEC10-BP04 自動化遏制能力

將事件範圍侷限與復原自動化，以縮短回應時間與對組織的影響。

一旦您依照程序手冊建立和操演程序和工具，就可以將邏輯解構成為以程式碼為基礎的解決方案，許多回應人員可將其做為工具使用，以做到自動回應，並免除回應人員面對的變通或猜測。如此可以加速回應的生命週期。下一個目標是透過提醒或事件本身 (而不是由回應人員) 叫用，讓此程式碼能夠完全自

動化，以建立事件驅動的回應。這些程序也應該自動將相關資料新增到您的安全系統。例如，涉及來自不需要 IP 地址的流量的事故可以自動填入 AWS WAF 封鎖清單或網路防火牆規則群組，以防止進一步的活動。

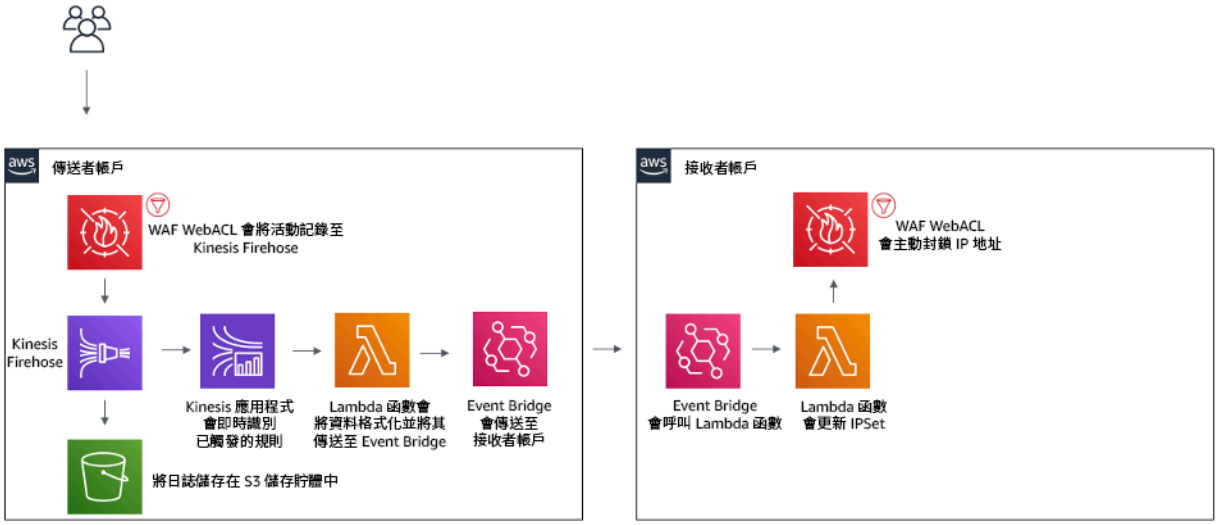


圖 3：自動封鎖已知惡意 IP 地址的 AWS WAF。

使用事件驅動的回應系統，偵測機制會觸發回應機制，以自動修復事件。您可以使用事件驅動的回應功能，減少偵測機制與回應機制之間體現價值的時間。若要建立此事件驅動架構，您可以使用 AWS Lambda；這是一種無伺服器運算服務，可執行程式碼以回應事件，並自動為您管理基礎運算資源。例如，假設您有一個已啟用 AWS CloudTrail 服務的 AWS 帳戶。如果 AWS CloudTrail 發生停用 (透過 `cloudtrail:StopLogging` API 呼叫)，您可以使用 Amazon EventBridge 監控特定的 `cloudtrail:StopLogging` 事件，並叫用 AWS Lambda 函數以呼叫 `cloudtrail:StartLogging` 以重新啟動記錄。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

自動化遏制能力。

資源

相關文件：

- AWS 事故回應指南

相關影片：

- 準備和回應 AWS 環境中的安全事故

SEC10-BP05 預先佈建存取權

確認事件回應者具有在 AWS 中預先佈建的正確存取權限，以縮短調查直至復原所需的時間。

常見的反模式：

- 使用事件應變的根帳戶。
- 更改現有的使用者帳戶。
- 當提供即時權限提升時直接操控 IAM 許可。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

AWS 建議盡可能降低或避免對長期憑證的依賴，而是採用臨時憑證和 即時 權限提升機制。長期憑證容易發生安全性風險並會增加營運負擔。對於大多數管理任務，以及事件應變任務，我們建議您實作 聯合身分 以及 適用於管理存取權的臨時權限提升。在此模型中，使用者會請求提升至較高層級的權限 (例如事件應變角色)；如果使用者符合提升的資格，則會將請求傳送至核准者。如果請求獲得核准，使用者就會收到一組臨時 AWS 憑證 ，使用者可使用此憑證來完成其任務。在這些憑證過期後，使用者就必須提交新的提升權限請求。

我們建議在大多數事件應變情境中，使用臨時權限提升。正確的做法是使用 AWS Security Token Service 和 工作階段政策 來界定存取權的範圍。

當發生聯合身分不可用的情況，例如

- 與遭盜用身分提供者 (IdP) 相關的中斷。
- 設定錯誤或人為錯誤會導致聯合存取管理系統遭到破壞。
- 分散式阻斷服務 (DDoS) 事件或使系統無法使用之類的惡意活動。

在前述的案例中，應會有已設定的 緊急 存取權，可協助調查和及時修復事件。我們建議您使用 具有適當許可的 IAM 使用者， 來執行任務和存取 AWS 資源。僅將根憑證用於 需要根使用者存取權的任務。若要確認事件回應者是否具有 AWS 和其他相關系統的正確存取權，我們建議預先佈建專屬的使用者帳戶。該類使用者帳戶需要提升的存取權，且必須受到嚴格的控制和監控。必須以執行必要任務所需的最低權限來建置這些帳戶，而存取權層級應以事件管理計劃中建立的程序手冊為基礎。

使用專用和專屬的使用者及角色作為最佳實務。透過新增 IAM 政策而臨時提升權限的使用者和角色存取權，會同時使得使用者在事件發生期間的存取權不明確，又有無法將提升的權限撤銷的風險。

您必須盡可能移除相依性，來確認可在各種可能的失敗情境下獲得存取權。為了做到這一點，建立程序手冊來確認事件應變使用者的建立身分是專屬安全性帳戶中的 AWS Identity and Access Management 使用者，且不會透過任何現有的聯合或單一登入 (SSO) 解決方案來管理事件應變使用者。每個個別回應者必須具備其專屬的指定帳戶。帳戶組態必須強制執行 強式密碼政策 和多重要素驗證 (MFA)。如果事件應變程序手冊僅需要 AWS Management Console 的存取權，使用者就不應設定存取金鑰，且應明確禁止使用者建立存取金鑰。您可以使用 IAM 政策或服務控制政策 (SCP) 進行設定，如同 AWS Organizations SCP 的 AWS 安全性 最佳實務中所述。除了在其他帳戶中擔任事件應變角色的能力外，使用者不應具備任何權限。

在事件期間，必須將存取權授予其他內部或外部人員，來協助調查、修復和復原活動。在此案例中，使用先前提到的程序手冊機制，而且必須制定程序，以確認在事件完成後，立即將任何其他存取權撤回。

若要確認事件應變角色的使用是否受到適當的監控和稽核，則必須確保未在人員之間共用為此目的建立的 IAM 使用者帳戶，且除非特定任務所需，否則不得使用 AWS 帳戶 根使用者。如果需要根使用者 (例如，特定帳戶的 IAM 存取權不可用時)，請使用獨立的程序，其中有可用的程序手冊，來確認根使用者密碼和 MFA 權杖是否可用。

若要為事件應變角色設定 IAM 政策，請考慮使用 IAM Access Analyzer 來根據 AWS CloudTrail 日誌產生政策。若要這麼做，請向管理員授予在非生產帳戶上事件應變角色的存取權，並透過程序手冊加以執行。完成後，您就可以建立政策來僅允許所採取的動作。接著就可以將此政策套用至所有帳戶中的所有事件應變角色。您可能希望為每個程序手冊建立個別 IAM 政策，來讓管理和稽核作業更輕鬆。範例程序手冊可能包含勒索軟體、資料洩漏、生產存取權遺失和其他情境的應變計劃。

使用事件應變使用者帳戶來擔任 在其他 AWS 帳戶 中專屬事件應變 IAM 角色。必須將這些角色設定為僅供安全性帳戶中的使用者擔任，而信任關係必須要求呼叫主體使用 MFA 進行驗證。這些角色必須使用嚴格控制範圍的 IAM 政策來控制存取權。確保所有對這些角色的 `AssumeRole` 請求都記錄在 CloudTrail 中並據以發出警示，而使用這些角色採取的任何動作都會記錄下來。

強烈建議必須清楚地命名 IAM 使用者帳戶和 IAM 角色，因此您可以輕鬆地在 CloudTrail 日誌中找到這些帳戶和角色。這類範例便是將 IAM 帳戶命名為 *<USER_ID>*-BREAK-GLASS 以及將 IAM 角色命名為 BREAK-GLASS-ROLE。

CloudTrail 會用來在 AWS 帳戶中記錄 API 活動，且應用來 設定對事件應變角色使用情形的警示。請參閱部落格貼文，其中會說明使用根金鑰如何設定警示。您可以修改說明，以便針對以下事件設定 Amazon CloudWatch 指標篩選條件至篩選條件： `AssumeRole` 事件，該事件與事件應變 IAM 角色相關：

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

由於事件應變角色可能具備很高的存取權限，因此必須將這些警示傳送給多個群組，並據此快速採取行動。

在事件期間，回應者可能需要存取未受 IAM 直接保護的系統。其中可能包含 Amazon Elastic Compute Cloud 執行個體、Amazon Relational Database Service 資料庫或軟體即服務 (SaaS) 平台。強烈建議使用此方法，而不是使用 SSH 或 RDP 等原生通訊協定，AWS Systems Manager Session Manager 會用於對 Amazon EC2 執行個體的所有管理存取權。您可以使用安全且受稽核的 IAM 來控制此存取權。 您也可以使用 AWS Systems Manager Run Command 文件 來自動化部分程序手冊，如此可減少使用者錯誤並縮短復原時間。如需資料庫和第三方工具的存取權，我們建議將存取憑證存放在 AWS Secrets Manager 中，並將存取權授予事件回應者角色。

最後，應將事件應變 IAM 使用者帳戶的管理作業新增至 加入者、異動者和離職者程序中， 並定期審查和測試此管理作業，以確認僅允許預期的存取。

資源

相關文件：

- 管理對 AWS 環境的臨時提升存取權
- AWS 安全事件應變指南
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- 為 IAM 使用者設定帳戶密碼政策
- 在 AWS 中使用多重要素驗證 (MFA)
- 使用 MFA 設定跨帳戶存取權
- 使用 IAM Access Analyzer 來產生 IAM 政策
- 在多帳戶環境中 AWS Organizations 服務控制政策的最佳實務
- 如何在使用 AWS 帳戶的根存取金鑰時接收通知
- 使用 IAM 受管政策來建立精細的工作階段許可

相關影片：

- 將 AWS 中的事件應變和鑑識自動化
- 執行手冊、事件報告和事件應變的 DIY 指南
- 準備和回應 AWS 環境中的安全事件

相關範例：

- 實驗室：AWS 帳戶設定和根使用者
- 實驗室：使用 AWS 主控台和 CLI 來應變事件

SEC10-BP06 預先部署工具

確保安全人員具有預先部署到 AWS 中的適當工具，以縮短調查直至復原的時間。

若要將安全工程和操作功能自動化，您可以使用 AWS 提供的完整 API 和工具集。您可以將身份管理、網路安全、資料保護和監控功能完全自動化，並使用現有的熱門軟體開發方法遞送這些功能。建置安全自動化時，您的系統可以監控、檢閱和啟動回應，而不是讓人員監控您的安全地位並手動回應事件。自動跨 AWS 服務將可搜尋和相關日誌資料提供給事故回應者的有效方法，就是啟用 Amazon Detective。

若您的事件回應團隊持續以相同方式回應警示，可能會形成警示疲勞的風險。隨著時間的推移，團隊可能會變得對收到提醒不敏感，而且在處理一般情況時可能會犯錯，或是錯過不尋常的警示。自動化使用能夠處理重複和一般提醒的功能，讓人員處理敏感和獨特的事件，有助於避免發生提醒疲倦的情形。整合異常偵測系統 (例如 Amazon GuardDuty、AWS CloudTrail Insights 和 Amazon CloudWatch 異常檢測) 可以減輕常見閾值型提醒的負擔。

您可以透過程式設計方式將程序中的步驟自動化，以改善手動程序。定義事件的補救模式之後，您可以將該模式分解為可行的邏輯，並撰寫程式碼來執行該邏輯。回應人員接著可以執行該程式碼來修正問題。隨著時間的推移，您可以將越來越多的步驟自動化，最終自動處理整個類別的常見事件。

對於在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的作業系統內執行的工具，您應該使用 AWS Systems Manager Run Command 進行評估，這可讓您使用在 Amazon EC2 執行個體作業系統上安裝的代理程式，從遠端安全地管理執行個體。這需要 Systems Manager Agent (SSM 代理程式)，此代理程式預設安裝在許多 Amazon Machine Image (AMI) 上。不過請注意，執行個體一旦遭侵害，對於在其上執行的工具或代理程式發出的回應都不應視為可信任。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 預先部署工具：確保安全人員具有預先部署到 AWS 中的適當工具，以便可以對事件做出適當的回應。
  - 實驗室：使用 AWS Management Console和 CLI 處理事故回應
  - 使用 Jupyter 的事故回應手冊 - AWS IAM
  - AWS 安全自動化
- 實作資源標記：使用資訊標記資源 (例如調查中資源的程式碼)，以便您可以在事件期間識別資源。
  - AWS 標記策略

資源

相關文件：

- AWS 事故回應指南

相關影片：

- 執行手冊、事件報告和事件回應的 DIY 指南

SEC10-BP07 執行演練日

演練日也稱為模擬或練習，是內部舉辦的活動，提供有條理的機會，供您在寫實情境下演練事件管理計劃和程序。這些事件應該使用在真實世界情境中使用的相同工具和技術來鍛煉回應者 - 甚至模仿真實世界環境。演練日基本上就是用來準備和反覆改善您的回應能力。對於進行演練日的活動，您可能會發現有價值的原因包括：

- 驗證整備
- 培養信心 – 從模擬和培訓員工得到學習
- 遵守合規或合約義務
- 產生用於認證的成品
- 靈活 – 增量改進
- 加速並改善工具
- 精簡溝通和上報
- 培養安然面對罕見和意外情形的能力

基於上述原因，參與模擬活動所衍生的價值，會在壓力事件期間提高組織發揮的效用。開發既實際又有益的模擬活動可能是艱鉅的作業。雖然測試處理熟知事件的程序或自動化具有某些優勢，但參與創造性的 安全事故回應模擬 (SIRS) 活動，以考驗自己面對意外和持續改善的能力，同樣也有價值。

建立針對您的環境、團隊和工具量身打造的自訂模擬。找出問題並根據它設計您的模擬。這可能是洩露的憑證、與不需要的系統通訊的伺服器，或導致未經授權暴露的錯誤組態等項目。識別熟悉貴組織的工程師來建立情境，以及另一個要參與的群組。情境應該是真實的，且具有足夠的挑戰性來彰現價值。它應該包括實作記錄、通知、呈報和執行執行手冊或自動化的機會。在模擬期間，您的回應者應該鍛煉其技術和組織技能，而且領導者應該參與以建立其事故管理技能。模擬結束時，讚揚團隊的努力，並尋找反覆、重複和擴充到進一步模擬的方法。

AWS 已建立事故回應執行手冊範本 ，您可以不僅將其用來準備您的回應工作，還可以將其做為模擬的基礎。規劃時，模擬可以分成五個階段。

收集證據： 在這個階段，團隊將透過各種方式獲得提醒，例如內部票證系統、來自監控工具的提醒、匿名提示，甚至是公共新聞。然後，團隊開始審查基礎設施和應用程式日誌，以判定入侵的來源。此步驟還應涉及內部呈報和事故領導地位。一旦識別，團隊就會繼續遏制事故

遏制事故： 團隊將判定發生了事故並確定入侵的來源。團隊現在應該採取行動來遏制它，例如，停用遭入侵的憑證、隔離運算資源或撤銷角色的許可。

杜絕事故： 既然他們已遏制事故，團隊就會努力緩解應用程式或基礎設施組態中易受入侵的任何漏洞。這可能包括輪換用於工作負載的所有憑證、修改存取控制清單 (ACL) 或變更網路組態。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 執行 在生產環境中：針對涉及關鍵人員和管理層的各種威脅執行模擬的 事故 回應 活動 (演練日) 。
- 汲取經驗教訓：從 在生產環境中 中獲得的經驗教訓應該成為改善程序的回饋意見的一部分。

資源

相關文件：

- AWS 事故回應指南
- AWS 彈性災難復原

相關影片：

- 執行手冊、事件報告和事件回應的 DIY 指南

可靠性

主題

- 基礎
- 工作負載架構
- 變更管理
- 失敗管理

基礎

問題

- REL 1 您如何管理服務配額和限制？
- REL 2 如何規劃您的網路拓撲？

REL 1 您如何管理服務配額和限制？

雲端型工作負載架構會有服務配額 (也稱為服務限制)。這些配額旨在用於防止不慎佈建超過您所需的資源，並限制 API 操作上的請求率，以防止服務遭到濫用。此外也會有資源限制，例如，您可將位元壓入光纖電纜的速率或實體磁碟上的儲存量會受到限制。

最佳實務

- REL01-BP01 了解服務配額和限制
- REL01-BP02 管理跨帳戶和區域的服務配額
- REL01-BP03 透過架構適應固定服務配額和限制
- REL01-BP04 監控和管理配額
- REL01-BP05 自動配額管理
- REL01-BP06 確保目前配額與最大使用量之間存在足夠差距以適應容錯移轉

REL01-BP01 了解服務配額和限制

您需了解工作負載架構的預設配額和配額增加要求。您也需知道哪些資源限制 (例如，磁碟或網路) 具有潛在影響。

Service Quotas 是一項 AWS 服務，有助於您從單一位置管理 100 多種 AWS 服務的配額。除了查閱配額值外，您也可以從 Service Quotas 主控台或透過 AWS 開發套件請求和追蹤配額增長。AWS Trusted Advisor 提供服務配額檢查功能，會顯示部分服務某些層面的用量和配額。每項服務的預設服務配額也根據各自服務列於 AWS 文件中，例如，請參閱 Amazon VPC 配額。若要在 API Gateway 中設定用於調節 API 的速率限制，請設定用量計畫。在其各自服務上設為組態的其他限制包括佈建 IOPS、分配的 RDS 儲存體，以及 EBS 磁碟區分配。Amazon Elastic Compute Cloud (Amazon EC2) 具有專門的 Service Limits 儀表板，有助於您管理執行個體、Amazon Elastic Block Store (Amazon EBS) 和彈性 IP 地址限制。如果在您的使用案例中，服務配額會影響您的應用程式效能且無法根據您的需求調整，請聯絡 AWS Support 以查看是否有緩解措施。

常用的反模式：

- 部署工作負載，但不考慮所使用 AWS 服務上的服務配額。
- 設計工作負載，但不調查和適應 AWS 服務的設計限制。
- 部署具有重要用途的工作負載取代已知的現有工作負載，但事先未設定必要的配額或聯絡 AWS Support。
- 規劃事件以將流量導引至您的工作負載，但事先未設定必要的配額或聯絡 AWS Support。

建立此最佳實務的優勢： 了解服務配額、API 調節限制和設計限制，可讓您在設計、實作和操作工作負載的過程中納入這些考量。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 在發佈的文件和 Service Quotas 中審查 AWS 服務配額。
  - AWS Service Quotas (先前稱為 Limits)
- 透過查看部署程式碼來確定工作負載所需的所有服務。
- 使用 AWS Config 尋找在 AWS 帳戶 中使用的所有 AWS 資源。
  - AWS Config 支援的 AWS 資源類型和資源關係
- 您也可以使用 AWS CloudFormation 來確定所使用的 AWS 資源。查看在 AWS Management Console或透過 list-stack-resources CLI 命令建立的資源。您也可以查看設為自行在範本中部署的資源。
  - 在 AWS Management Console 上檢視 AWS CloudFormation 堆疊資源和資料
  - AWS CLI for CloudFormation：list-stack-resources
- 決定適用的服務配額。透過 Trusted Advisor 和 Service Quotas 使用可以程式設計方式存取的資訊。

資源

相關文件：

- AWS Marketplace：可追蹤限制的 CMDB 產品
- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 (請參閱 Service Limits 一節)
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 Service Limits
- 什麼是 Service Quotas？

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL01-BP02 管理跨帳戶和區域的服務配額

如果您使用多個 AWS 帳戶或 AWS 區域，確保在生產工作負載執行的所有環境中都要求合適的配額。

系統會針對每個帳戶追蹤服務配額。除非另有說明，否則每個配額都是 AWS 區域特有。除生產環境之外，也會在所有適用的非生產環境中管理配額，因此不會阻礙測試和開發。

常用的反模式：

- 允許一個隔離區域內的資源利用率增長，但無維持其他隔離區域中容量的機制。
- 獨自手動設定隔離區域中的所有配額。
- 未確保隔離區域的部署在部署遺失時調整規模，以適應來自其他區域的流量增加。

建立此最佳實務的優勢： 確保您可以在隔離區域無法使用時可以處理目前的負載，這可協助減少在容錯移轉期間發生的錯誤，而不會對客戶造成阻斷服務狀況。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 根據您的服務要求、延遲、法規和災難復原 (DR) 要求，選擇相關的帳戶和區域。
- 確定所有相關帳戶、區域和可用區域中的服務配額。限制範圍受限於帳戶和區域。
- 什麼是 Service Quotas？

資源

相關文件：

- AWS Marketplace：可追蹤限制的 CMDB 產品
- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 (請參閱 Service Limits 一節)
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 Service Limits
- 什麼是 Service Quotas？

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL01-BP03 透過架構適應固定服務配額和限制

瞭解不可變更的服務配額和實體資源及架構，以防止這些因素影響可靠性。

範例包括網路頻寬、AWS Lambda 承載大小、API Gateway 調節爆量速率，以及 Amazon Redshift 叢集的使用者同時連線數目。

常用的反模式：

- 執行基準化分析的時間過短，利用高載限制，但預期服務會以該容量持續執行一段期間。
- 選擇每位使用者或客戶使用一項服務的一項資源的設計，但未注意到擴展時會導致此項設計失效的設計限制。

建立此最佳實務的優勢： 追蹤 AWS 服務中的固定配額和工作負載其他部分中的限制 (例如連線能力限制、IP 地址限制和第三方服務的限制)，能夠讓您察覺何時趨向於配額限制，並有能力在超過配額前處理配額。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 注意固定服務配額：請注意固定的服務配額和限制，並根據這些因素建立架構。
  - AWS Service Quotas

資源

相關文件：

- AWS Marketplace：可追蹤限制的 CMDB 產品
- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 (請參閱 Service Limits 一節)
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 Service Limits
- 什麼是 Service Quotas？

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL01-BP04 監控和管理配額

評估潛在用量並適當地增加配額，以允許使用量按計劃增長。

對於支援的服務，您可以設定 CloudWatch 警示來監控用量並提醒您已接近配額限制，從而管理配額。這些警示可以從 Service Quotas 或 Trusted Advisor 觸發。您也可以使用 CloudWatch Logs 上的指標篩選條件，搜尋與擷取日誌中的模式，以判斷用量是否正在接近配額閾值。

常用的反模式：

- 設定了正在接近 Service Quotas 的警示，但無如何回應提醒的程序。
- 只設定 Service Quotas 支援的服務警示，但未監控其他服務。

建立此最佳實務的優勢： 自動追蹤 AWS 服務配額並根據這些配額監控您的使用量，可讓您查看何時會接近配額限制。您也可以使用此監控資料來評估何時可能降低配額以節省成本。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 監控和管理您的配額：評估您在 AWS 上的潛在使用量，適當提高區域服務配額，並允許使用量按計劃增長。

- 擷取當前資源消耗 (例如，儲存貯體、執行個體)。使用服務 API 操作 (例如 Amazon EC2 DescribeInstances API) 來收集當前資源消耗。
- 擷取您的目前配額：使用 AWS Service Quotas、AWS Trusted Advisor 和 AWS 文件。
  - 使用 AWS Service Quotas，這是一項 AWS 服務，有助於您從單一位置管理 100 多種 AWS 服務的配額。
  - 使用 Trusted Advisor 服務限制來確定您當前的服務限制。
  - 使用服務 API 操作來確定支援的當前服務配額。
  - 記錄請求增加的配額及其狀態：核准配額增加後，請確保您更新記錄，以反映配額的變更。

資源

相關文件：

- AWS Marketplace：可追蹤限制的 CMDB 產品
- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查，適用於 Service Limits
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 Service Limits
- 什麼是 Service Quotas？
- 使用 Amazon CloudWatch 警示監控 Service Quotas

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL01-BP05 自動配額管理

實作工具以在接近閾值時獲得提醒。您可以使用 AWS Service Quotas API，自動化配額增加請求。您可以自動化配額增加請求。

如果您將組態管理資料庫 (CMDB) 或票務系統與 Service Quotas 整合，則可以自動追蹤配額增加請求和目前的配額。除了 AWS 開發套件外，Service Quotas 也會使用 AWS Command Line Interface (AWS CLI) 提供自動化。

常用的反模式：

- 以試算表追蹤配額和使用量。
- 每日、每週或每月執行使用量報告，然後比較使用量與配額。

建立此最佳實務的優勢： 自動追蹤 AWS 服務配額並根據該配額監控您的使用量，可讓您查看何時會接近配額限制。您可以設定自動化，協助您在需要時請求增加配額。當您的使用量與實現風險降低 (登入資料遭危害時) 和成本節省的優勢背道而馳時，您可能會考慮降低部分配額。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 設定自動監控：使用開發套件實作工具，以在接近閾值時獲得提醒。
  - 使用 Service Quotas，並以如 AWS Limit Monitor 或 AWS Marketplace 中的產品等自動配額監控解決方案擴大此項服務。
    - 什麼是 Service Quotas？
    - AWS 上的配額監視器 - AWS 解決方案
  - 使用 Amazon SNS 和 AWS Service Quotas API 設定由配額閾值觸發的回應。
  - 測試自動化。
    - 設定限制閾值。
    - 與來自 AWS Config、部署管道、Amazon EventBridge 或第三方的變更事件整合。
    - 人工設定較低配額閾值以測試回應。
    - 設定觸發程序以在收到通知時採取適當的措施，以及在必要時讓人員聯絡 AWS Support。
    - 手動觸發變更事件。
    - 執行演練日以測試配額增長變更程序。

資源

相關文件：

- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- AWS Marketplace：可追蹤限制的 CMDB 產品
- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 (請參閱 Service Limits 一節)
- AWS 上的配額監視器 - AWS 解決方案

- Amazon EC2 Service Limits
- 什麼是 Service Quotas？

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL01-BP06 確保目前配額與最大使用量之間存在足夠差距以適應容錯移轉

當資源失敗時，在成功終止之前，其可能仍會被計入限額。在終止失敗的資源之前，確保您的配額涵蓋所有失敗的資源與替換資源的重疊部分。計算此差距時，應考慮可用區域失敗。

常用的反模式：

- 根據目前的需求設定服務配額，而不考量容錯移轉案例。

建立此最佳實務的優勢： 當事件可能影響可用性時，雲端可讓您實作策略來減輕影響或從這些事件中復原。這類策略通常包括建立額外資源以取代失敗的資源。您的配額策略必須容納這些額外的資源。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 確保您的服務配額和最大用量之間存在足夠的差距以適應容錯移轉。
  - 確定服務限制，並在此過程中考慮您的部署模式、可用性要求和使用量增長。
  - 視需要請求增加配額。規劃必要的時間來滿足增加配額的請求。
    - 確定您的可靠性方案 (也稱為「幾個 9」)。
    - 建立故障案例 (例如，元件、可用區域或區域遺失)。
    - 建立您的部署方法 (例如，Canary、藍/綠、紅/黑或滾動)。
    - 為當前限制新增適當的緩衝 (例如 15%)。
    - 為使用量增長制定計畫 (例如，監控使用量趨勢)。

資源

相關文件：

- AWS Marketplace：可追蹤限制的 CMDB 產品

- AWSService Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 (請參閱 Service Limits 一節)
- Amazon EC2 Service Limits
- 什麼是 Service Quotas？

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas

REL 2 如何規劃您的網路拓撲？

工作負載經常存在於多個環境中。這些環境包括多個 (可公開存取和私有的) 雲端環境，且可能包含您現有的資料中心基礎設施。計畫必須包括系統內和系統間連線、公有 IP 地址管理、私有 IP 地址管理和網域名稱解析等網路考量因素。

最佳實務

- REL02-BP01 針對工作負載公有端點使用高可用性網路連線
- REL02-BP02 在雲端中的私有網路與內部部署環境之間佈建冗餘連線能力
- REL02-BP03 確保 IP 子網路分配帳戶具有擴展性和可用性
- REL02-BP04 偏好軸幅式拓撲而非多對多網狀拓撲
- REL02-BP05 在連線的所有私有地址空間中強制使用不重疊的私有 IP 地址範圍

REL02-BP01 針對工作負載公有端點使用高可用性網路連線

這些端點及其路由必須具備高可用性。為達成此目的，請使用高度可用的 DNS、內容交付網路 (CDN)、API Gateway、負載平衡或反向代理。

Amazon Route 53、AWS Global Accelerator、Amazon CloudFront、Amazon API Gateway 和 Elastic Load Balancing (ELB) 全都提供高可用性的公開端點。您也可以選擇評估用於負載平衡和代理的 AWS Marketplace 軟體設備。

您的工作負載提供的服務取用者 (無論是最終使用者還是其他服務) 會請求這些服務端點。有多種 AWS 資源可供您提供高可用性端點。

Elastic Load Balancing 提供跨可用區域的負載平衡，執行 Layer 4 (TCP) 或 Layer 7 (http/https) 路由，與 AWS WAF 整合，以及與 AWS Auto Scaling 整合，以便協助建立自我修復基礎設施並吸收增加的流量，同時在流量減少時釋放資源。

Amazon Route 53 是可擴展的高可用性網域名稱系統 (DNS) 服務，能將使用者請求連接到 AWS 中執行的基礎設施，例如 Amazon EC2 執行個體、Elastic Load Balancing 負載平衡器或 Amazon S3 儲存貯體，還可用於將使用者路由到 AWS 外部的基礎設施。

AWS Global Accelerator 是網路層服務，可供您透過 AWS 全球網路將流量引導至最佳端點。

分散式阻斷服務 (DDoS) 攻擊所存在的風險會將合法流量阻擋在外，並減少使用者的可用性。AWS Shield 為您工作負載上的 AWS 服務端點，提供抵擋這些攻擊的自動防護，無需額外費用。您可以使用 APN 合作夥伴和 AWS Marketplace 提供的虛擬設備來增強這些功能，以滿足您的需求。

常用的反模式：

- 在執行個體或容器上使用公有網際網路地址，並透過 DNS 管理其連線。
- 使用網際網路通訊協定地址，而非網域名稱來定位服務。
- 提供內容 (網頁、靜態資產、媒體檔案) 到大型地理區域，而不使用內容交付網路。

建立此最佳實務的優勢： 透過在工作負載中實作高可用性服務，即可知道負載可供使用者使用。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

確保為工作負載使用者提供高可用性連線：Amazon Route 53、AWS Global Accelerator、Amazon CloudFront、Amazon API Gateway 和 Elastic Load Balancing (ELB) 全都提供高可用性的公開端點。您也可能選擇評估用於負載平衡和代理的 AWS Marketplace 軟體設備。

- 確保您與使用者的連線高度可用。
- 確保您使用的是高度可用的 DNS 來管理應用程式端點的網域名稱。
  - 如果使用者透過網際網路存取您的應用程式，請使用服務 API 操作來確認正確使用網際網路閘道。同時確認託管應用程式端點之子網路的路由表項目正確。
    - DescribeInternetGateways
    - DescribeRouteTables
- 確保在應用程式前面使用高可用性的反向代理或負載平衡器。
  - 如果使用者透過您的內部部署環境存取您的應用程式，應確保 AWS 與您的內部部署環境之間的連線高度可用。
  - 使用 Route 53 來管理您的網域名稱。
    - 什麼是 Amazon Route 53？

- 使用符合您要求的第三方 DNS 供應商。
- 使用 Elastic Load Balancing。
  - 什麼是 Elastic Load Balancing？
- 使用符合您要求的 AWS Marketplace 設備。

資源

相關文件：

- APN 合作夥伴：可以幫助您規劃聯網的合作夥伴
- AWS Direct Connect 恢復能力建議
- 適用於網路基礎設施的 AWS Marketplace
- Amazon Virtual Private Cloud 連線能力選項白皮書
- 多個資料中心 HA 網路連線
- 以 Direct Connect 彈性工具組開始使用
- VPC 端點和 VPC 端點服務 (AWS PrivateLink)
- 什麼是 AWS Global Accelerator？
- 什麼是 Amazon VPC？
- 什麼是 Transit Gateway？
- 什麼是 Amazon CloudFront？
- 什麼是 Amazon Route 53？
- 什麼是 Elastic Load Balancing？
- 使用 Direct Connect Gateway

相關影片：

- AWS re:Invent 2018：進階 VPC 設計及 Amazon VPC 的新功能 (NET303)
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構 (NET406-R1)

REL02-BP02 在雲端中的私有網路與內部部署環境之間佈建冗餘連線能力

在單獨部署的私有網路之間使用多個 AWS Direct Connect 連線和多個 VPN 通道。使用多個 Direct Connect 位置以實現高可用性。如果使用多個 AWS 區域，請至少在其中兩個區域中確保冗餘。您可能

需要評估終止 VPN 的 AWS Marketplace 設備。如果您使用 AWS Marketplace 設備，可在不同的可用區域中部署冗餘執行個體以實現高可用性。

AWS Direct Connect 是一項雲端服務，可讓您輕鬆地建立一個連接內部部署環境和 AWS 的專用網路連線。您的內部部署資料中心可以使用 Direct Connect Gateway，連線至橫跨多個 AWS 區域的多個 AWS VPC。

此冗餘可以解決會影響連線彈性的可能故障：

- 您可如何彈性回應拓樸故障？
- 如果您錯誤設定並刪除連線會怎樣？
- 您是否能夠處理服務流量或使用方面的意外增加情況？
- 您是否能夠應對企圖的分散式阻斷服務 (DDoS) 攻擊？

透過 VPN 將 VPC 連線至內部部署資料中心時，您應在選擇需要在其上執行設備的供應商和執行個體大小時，考慮所需的彈性和頻寬要求。如果您使用的 VPN 設備在實作上沒有彈性，則您應透過第二個設備進行冗餘連線。對於所有這些情況，您需要定義可接受的復原時間並進行測試以確保能滿足這些要求。

如果您選擇使用 Direct Connect 連線將 VPC 連線到資料中心，且您需要此連線具有高可用性，請從各資料中心獲取冗餘 Direct Connect 連線。冗餘連線應使用不同於第一個位置的第二個 Direct Connect 連線。如果您擁有多個資料中心，請確保在不同的位置終止連線。使用 Direct Connect 彈性工具組 有助於您進行此項設定。

如果您選擇使用 AWS VPN 透過網際網路容錯移轉到 VPN，請務必了解其支援至多每個 VPN 通道 1.25 Gbps 的輸送量，但是若在同一 VGW 上終止多個 AWS Managed VPN 通道，則不支援等價多路徑 (ECMP) 傳出流量。除非您可以忍受容錯移轉時低於 1 Gbps 的速度，否則我們不建議您將 AWS Managed VPN 做為 Direct Connect 連線的備份。

您也可以使用 VPC 端點，將 VPC 私密連線至支援的 AWS 服務和採用 AWS PrivateLink 技術的 VPC 端點服務，而不需周遊公有網際網路。端點是虛擬裝置。它們是水平擴展、冗餘和高可用性的 VPC 元件。它們允許 VPC 中的執行個體與服務之間進行通訊，而不會對網路流量帶來可用性風險或頻寬限制。

常用的反模式：

- 您的現場網路與 AWS 之間只有一個連線供應商。
- 使用 AWS Direct Connect 連線的連線功能，但只有一個連線。

- 您的 VPN 連線只有一個路徑。

建立此最佳實務的優勢： 透過在您的雲端環境與您的公司或內部部署環境之間實作冗餘連線，即可確保兩個環境之間的相依服務能夠可靠地進行通訊。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 確保 AWS 與內部部署環境之間具有高度可用的連線。在單獨部署的私有網路之間使用多個 AWS Direct Connect 連線和多個 VPN 通道。使用多個 Direct Connect 位置以實現高可用性。如果使用多個 AWS 區域，請至少在其中兩個區域中確保冗餘。您可能需要評估終止 VPN 的 AWS Marketplace 設備。如果您使用 AWS Marketplace 設備，可在不同的可用區域中部署冗餘執行個體以實現高可用性。
  - 確保與內部部署環境之間存在冗餘連線。您可能需要與多個 AWS 區域的冗餘連線才能滿足可用性需求。
    - AWS Direct Connect 恢復能力建議
    - 使用冗餘站點對站點 VPN 連接提供容錯移轉
      - 使用服務 API 操作來識別對 Direct Connect 線路的正確使用。
        - DescribeConnections
        - DescribeConnectionsOnInterconnect
        - DescribeDirectConnectGatewayAssociations
        - DescribeDirectConnectGatewayAttachments
        - DescribeDirectConnectGateways
        - DescribeHostedConnections
        - DescribeInterconnects
      - 如果僅存在一個 Direct Connect 連線，或者一個都沒有，則設定到虛擬私有閘道的冗餘 VPN 通道。
        - 什麼是 AWS 站點對站點 VPN？
- 擷取您當前的連線 (例如，直接連線、虛擬私有閘道、AWS Marketplace 設備)。
  - 使用服務 API 操作來查詢 Direct Connect 連線的組態。
    - DescribeConnections
    - DescribeConnectionsOnInterconnect
    - DescribeDirectConnectGatewayAssociations

    - DescribeDirectConnectGatewayAttachments
    - DescribeDirectConnectGateways
    - DescribeHostedConnections
    - DescribeInterconnects
  - 使用服務 API 操作來收集路由表使用的虛擬私有閘道。
    - DescribeVpnGateways
    - DescribeRouteTables
  - 使用服務 API 操作收集路由表使用的 AWS Marketplace 應用程式。
    - DescribeRouteTables

資源

相關文件：

- APN 合作夥伴：可以幫助您規劃聯網的合作夥伴
- AWS Direct Connect 恢復能力建議
- 適用於網路基礎設施的 AWS Marketplace
- Amazon Virtual Private Cloud 連線能力選項白皮書
- 多個資料中心 HA 網路連線
- 使用冗餘站點對站點 VPN 連接提供容錯移轉
- 以 Direct Connect 彈性工具組開始使用
- VPC 端點和 VPC 端點服務 (AWS PrivateLink)
- 什麼是 Amazon VPC？
- 什麼是 Transit Gateway？
- 什麼是 AWS 站點對站點 VPN？
- 使用 Direct Connect Gateway

相關影片：

- AWS re:Invent 2018：進階 VPC 設計及 Amazon VPC 的新功能 (NET303)
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構 (NET406-R1)

REL02-BP03 確保 IP 子網路分配帳戶具有擴展性和可用性

Amazon VPC IP 地址範圍必須足夠大，以適應工作負載的要求，包括考慮將來擴展 IP 地址以及跨可用區域將 IP 地址分配給子網路。這包括負載平衡器、EC2 執行個體和容器型應用程式。

規劃您的網路拓樸時，首先要定義 IP 地址空間。私有 IP 地址範圍 (依循 RFC 1918 指導方針) 應分配給各 VPC。在此流程中請滿足下列要求：

- 允許每個區域為多於一個 VPC 準備 IP 地址空間。
- 在 VPC 內，允許多個子網路的空間，並可跨越多個可用區域。
- 經常在 VPC 內留下未用 CIDR 區塊空間，以供未來擴展。
- 確保有 IP 地址空間可以滿足您可能會用到之 EC2 執行個體的任何臨時機群需求，例如，用於機器學習的 Spot Fleets、Amazon EMR 叢集或 Amazon Redshift 叢集。
- 請注意，在各子網路 CIDR 區塊中，前四個 IP 地址和最後一個 IP 地址均預留起來，無法供您使用。
- 您應計劃部署大型 VPC CIDR 區塊。請注意，雖然無法變更或刪除分配給 VPC 的最初 VPC CIDR 區塊，但您可以將不重疊的其他 CIDR 區塊新增至 VPC。無法變更子網路 IPv4 CIDR，但可以變更 IPv6 CIDR。請牢記，部署最大的 VPC 可能 (/16) 會導致超過 65,000 個 IP 地址。單單在基底 10.x.x.x IP 地址空間中，您即可佈建 255 個此類 VPC。因此，您應該選擇過大而非過小，以便更容易管理 VPC。

常用的反模式：

- 建立小型 VPC。
- 建立小型子網路，然後必須隨著增長將子網路新增至組態。
- 錯誤預估 Elastic Load Balancer 可以使用的 IP 地址數量。
- 在相同的子網路中部署許多高流量負載平衡器。

建立此最佳實務的優勢： 如此可確保您可以適應工作負載的增長，並在向上擴展時繼續提供可用性。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 規劃網路以適應增長、法規要求以及與其他網路整合。增長可能會被低估，合規要求可能會發生變化，並且如果沒有適當的規劃，採購或私有網路連線可能會難以實作。
  - 根據您的服務要求、延遲、法規和災難復原 (DR) 要求，選擇相關的 AWS 帳戶和區域。

- 確定您對區域 VPC 部署的需求。
- 確定 VPC 的大小。
  - 確定是否要部署多 VPC 連線。
    - 什麼是 Transit Gateway？
    - 單區域多 VPC 連線
  - 確定您是否需要區隔聯網以滿足法規要求。
  - 讓 VPC 盡可能保持較大規模。雖然無法變更或刪除分配給 VPC 的最初 VPC CIDR 區塊，但您可以將不重疊的其他 CIDR 區塊新增至 VPC。但這可能會導致地址範圍片段化。
  - 讓 VPC 盡可能保持較大規模。雖然無法變更或刪除分配給 VPC 的最初 VPC CIDR 區塊，但您可以將不重疊的其他 CIDR 區塊新增至 VPC。但這可能會導致地址範圍片段化。

資源

相關文件：

- APN 合作夥伴：可以幫助您規劃聯網的合作夥伴
- 適用於網路基礎設施的 AWS Marketplace
- Amazon Virtual Private Cloud 連線能力選項白皮書
- 多個資料中心 HA 網路連線
- 單區域多 VPC 連線
- 什麼是 Amazon VPC？

相關影片：

- AWS re:Invent 2018：進階 VPC 設計及 Amazon VPC 的新功能 (NET303)
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構 (NET406-R1)

REL02-BP04 偏好軸幅式拓撲而非多對多網狀拓撲

如果兩個以上的網路地址空間 (例如，VPC 和內部部署網路) 透過 VPC 對等互連 AWS Direct Connect 或 VPN 連線，則使用軸幅式模型，例如 AWS Transit Gateway 提供的此類模型。

如果您只有這兩種網路，僅需相互連線，但隨著網路數成長，此類網狀連線的複雜性將變得難以處理。AWS Transit Gateway 提供容易維護的軸幅式模型，以便跨多條網路路由流量。

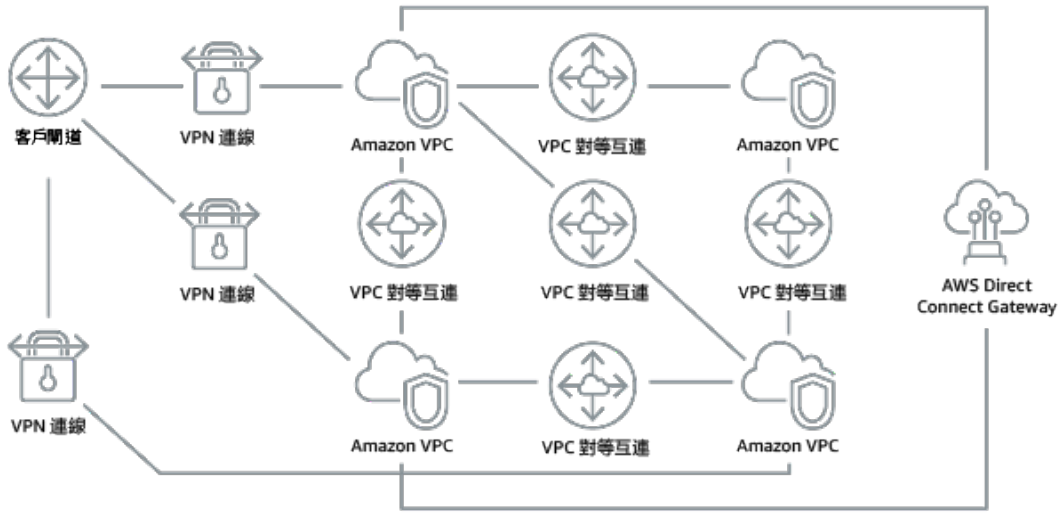


圖 1：未使用 AWS Transit Gateway：您需要將各 Amazon VPC 對等互連並使用 VPN 連接與各現場位置互連，因此隨著規模擴展，此做法會變得複雜。

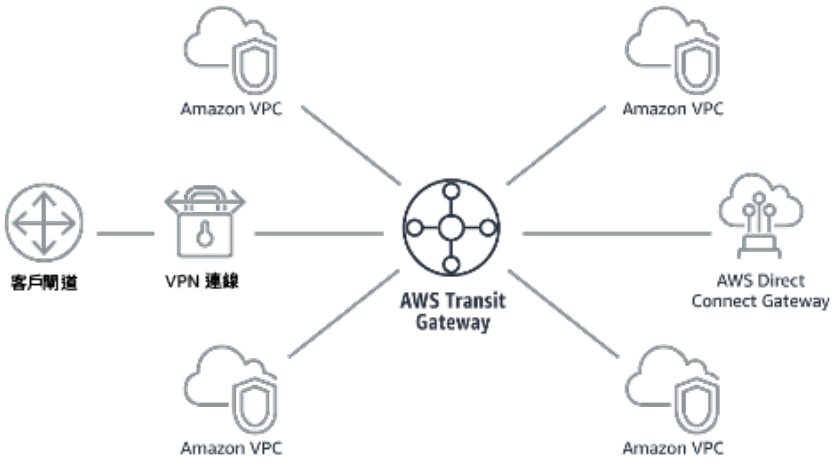


圖 2：使用 AWS Transit Gateway：您只需將各 Amazon VPC 或 VPN 連線到 AWS Transit Gateway，然後它就會路由往返各 VPC 或 VPN 的流量。

常用的反模式：

- 使用 VPC 對等互連來連接兩個以上的 VPC。
- 為每個 VPC 建立多個 BGP 工作階段，以建立橫跨多個 AWS 區域間多個 Virtual Private Clouds (VPC) 的連線。

建立此最佳實務的優勢： 隨著網路數量增加，此類網狀連線的複雜性將變得難以處理。AWS Transit Gateway 提供容易維護的軸幅式模型，以便在多條網路之間路由流量。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 偏好軸幅式拓撲而非多對多網狀拓撲。如果兩個以上的網路地址空間 (VPC、內部部署網路) 透過 VPC 對等互連、AWS Direct Connect 或 VPN 連線，則使用軸幅式模型，例如 AWS Transit Gateway 提供的此類模型。
  - 如果只有這兩種網路，則僅需將這兩種網路相互連線，但隨著網路數量增加，此類網狀連線的複雜性將變得難以處理。AWS Transit Gateway 提供容易維護的軸幅式模型，以便跨多條網路路由流量。
    - 什麼是 Transit Gateway？

資源

相關文件：

- APN 合作夥伴：可以幫助您規劃聯網的合作夥伴
- 適用於網路基礎設施的 AWS Marketplace
- 多個資料中心 HA 網路連線
- VPC 端點和 VPC 端點服務 (AWS PrivateLink)
- 什麼是 Amazon VPC？
- 什麼是 Transit Gateway？

相關影片：

- AWS re:Invent 2018：進階 VPC 設計及 Amazon VPC 的新功能 (NET303)
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構 (NET406-R1)

REL02-BP05 在連線的所有私有地址空間中強制使用不重疊的私有 IP 地址範圍

如果透過 VPN 對等互連或連線，則每個 VPC 的 IP 地址範圍不得重疊。同樣地，您必須避免 VPC 與內部部署環境或您所使用之其他雲端供應商之間出現 IP 地址衝突。您也須有一種在需要時分配私有 IP 地址範圍的方法。

IP 地址管理 (IPAM) 系統可以協助解決此問題。AWS Marketplace 提供多套 IPAM 系統。

常用的反模式：

- 在 VPC 中使用與內部部署或公司網路相同的 IP 範圍。
- 不追蹤用來部署工作負載之 VPC 的 IP 範圍。

建立此最佳實務的優勢： 主動規劃網路可確保在互連網路中不會出現多個相同的 IP 地址。這可防止在使用不同應用程式的工作負載部分中發生路由問題。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 監控和管理您的 CIDR 使用。評估您在 AWS 上的潛在使用情況，將 CIDR 範圍新增到現有 VPC，並建立 VPC 以允許計劃的用量增長。
  - 擷取當前的 CIDR 消耗 (例如，VPC、子網路)
    - 使用服務 API 操作來收集當前的 CIDR 消耗。
  - 記錄您當前的子網路用量。
    - 使用服務 API 操作來收集每個區域中每個 VPC 的子網路。
      - DescribeSubnets
    - 記錄目前用量。
    - 確定是否建立了任何重疊的 IP 範圍。
    - 計算備用容量。
    - 識別重疊的 IP 範圍。如果需要連線重疊範圍，則可以遷移至新的地址範圍，也可以使用 AWS Marketplace 的網路和連接埠轉換 (NAT) 設備。

資源

相關文件：

- APN 合作夥伴：可以幫助您規劃聯網的合作夥伴
- 適用於網路基礎設施的 AWS Marketplace
- Amazon Virtual Private Cloud 連線能力選項白皮書
- 多個資料中心 HA 網路連線
- 什麼是 Amazon VPC？

- 什麼是 IPAM？

相關影片：

- AWS re:Invent 2018：進階 VPC 設計及 Amazon VPC 的新功能 (NET303)
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構 (NET406-R1)

工作負載架構

問題

- REL 3 如何設計您的工作負載服務架構？
- REL 4 如何在分散式系統中設計防止失敗的互動？
- REL 5 如何設計分散式系統中的互動以緩解或承受故障？

REL 3 如何設計您的工作負載服務架構？

使用服務導向架構 (SOA) 或微型服務架構，建置擴展性與可靠性高的工作負載。服務導向架構 (SOA) 是透過服務界面讓軟體元件可重複使用的做法。微型服務架構則進一步讓元件變得更小、更簡單。

最佳實務

- REL03-BP01 選擇如何劃分工作負載
- REL03-BP02 建置專注於特定業務領域和功能的服務
- REL03-BP03 每個 API 都提供服務合約

REL03-BP01 選擇如何劃分工作負載

在確認應用程式的彈性要求時，工作負載劃分是很重要的。應盡可能避免整合型架構。您應審慎考量哪些應用程式元件可分解為微型服務。根據您的應用程式要求，這最終會盡可能由服務導向架構 (SOA) 與微型服務組合而成。可以無狀態的工作負載較有能力部署為微型服務。

預期成果： 工作負載應可受支援、可擴展，並且盡可能地鬆散耦合。

在選擇如何劃分工作負載時，請在效益與複雜性之間取得平衡。讓新產品能率先推出的正確做法，不同於打造可從最初需求擴展的工作負載的做法。重構現有的整合型時，您必須考量應用程式如何能支援以無狀態為方向的解構。將服務細分為較小的服務，可讓明確定義的小型團隊加以開發及管理。但較小的服務可能會帶來複雜性，包括延遲可能增加、偵錯更複雜，以及運作負擔增加。

常見的反模式：

- AWS Well-Architected 微型服務 Death Star 是一種特定情況：基本元件變得高度互相依賴，以致於只要有其中之一失敗，就會引發更加巨大的失敗，而導致元件像整合型一樣僵固且脆弱。

建立此實務準則的優勢：

- 更明確的劃分可造就更高的靈活性、組織彈性及可擴展性。
- 降低服務中斷的影響。
- 應用程式元件可能會有不同的可用性要求，這一點可藉由更細微的劃分來支應。
- 為支援工作負載的團隊明確定義責任。

未建立此最佳實務時的曝險等級： 高

實作指引

根據劃分工作負載的方式，選擇您的架構類型。選擇 SOA 或微型服務架構 (或在少數情況下選擇整合型架構)。即使您選擇從整合型架構開始，仍須確保該架構為模組化，且隨著使用者採用，產品擴展時，該架構最終可以演進成 SOA 或微型服務。SOA 和微型服務各自提供較小的劃分，這些劃分同時也是偏好使用的現代可擴展且可靠的架構；但在部署微型服務架構時，特別要考慮做一些取捨。

主要取捨之一，就是您現在擁有一種分散式運算架構，而其可能會增加您滿足使用者延遲要求的難度，並且在偵測和追蹤使用者互動方面還存在額外的複雜性。您可以利用 AWS X-Ray 來解決此問題。要考慮的另一個影響是，隨著您管理的應用程式數量增加，營運複雜性也隨之增加，因而需要部署多個獨立元件。

整合型、服務導向與微型服務架構

實作步驟

- 決定適當的架構以重構或建置您的應用程式。SOA 和微型服務各自提供較小的分隔，而這是偏好使用的現代可擴展和可靠架構。SOA 會是達成較小分隔的良好折衷方案，同時能避免微型服務的部分複雜性。如需詳細資訊，請參閱 微型服務權衡。
- 如果您的工作負載適用於此類型，且您的組織可以提供支援，則應使用微型服務架構達成最佳的靈活性和可靠性。如需詳細資訊，請參閱 實作 AWS 上的微型服務。
- 考慮遵循 Strangler Fig 模式， 將整合型重構為較小的元件。為此，必須逐步將特定的應用程式元件取代為新的應用程式和服務。AWS Migration Hub Refactor Spaces 可作為增量重構的起點。如需詳細資訊，請參閱 「使用扼制模式順暢地遷移內部部署的工作負載」。
- 實作微型服務時可能需要服務探索機制，讓這些分散式服務能夠彼此通訊。AWS App Mesh 可以搭配服務導向架構使用，以提供可靠的服務探索和存取。 AWS Cloud Map 也可用於動態、使用 DNS 的服務探索。
- 如果您要從整合型遷移至 SOA，Amazon MQ 可在您於雲端重新設計舊版應用程式時，以服務匯流排的形式消弭差距。
- 對於具有單一共用資料庫的現有整合型，請選擇如何將資料重新組織為較小的區段。此時可以按業務單位、存取模式或資料結構來劃分。在重構程序的這個時間點，您應選擇以關聯式或非關聯式 (NoSQL) 類型的資料庫繼續操作。如需詳細資訊，請參閱 「從 SQL 到 NoSQL」。

實作計劃的工作量： 高

資源

相關的最佳實務：

- REL03-BP02 建置專注於特定業務領域和功能的服務

相關文件：

- Amazon API Gateway：使用 OpenAPI 設定 REST API
- 什麼是服務導向架構？
- 有界限的環境 (領域驅動設計的集中模式)
- 實作 AWS 上的微型服務
- 微型服務權衡

- 微型服務 - 此新架構術語的定義
- AWS 上的微型服務
- 什麼是 AWS App Mesh？

相關範例：

- 迭代應用程式現代化研討會

相關影片：

- 透過 AWS 上的微型服務提供卓越品質

REL03-BP02 建置專注於特定業務領域和功能的服務

服務導向架構 (SOA) 會建置具有依業務需求定義之明確描述功能的服務。微型服務運用領域模型和有界限的環境來對此項業務進一步限縮，因此各服務僅做一件事。專注於特定功能讓您能夠區別不同服務的可靠性要求，並更集中瞄準投資目標。簡要的業務問題和與各服務相關的小型團隊，也更容易讓組織擴展。

在設計微型服務架構的過程中，運用領域驅動設計 (DDD) 有助於使用實體建立業務問題模型。例如，對於 Amazon.com 網站而言，實體可能包括包裝、交付、時間表、價格、折扣和貨幣。然後運用 有界限的環境，將此模型進一步劃分成更小的模型，其中具有相似功能和特性的實體歸類成一組。因此，以 Amazon.com 為例，包裝、交付及時間表會是出貨環境的一環，而價格、折扣及貨幣則是定價環境的一環。隨著此模型劃分成多個環境，如何界定微型服務界限的範本便會浮現。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 根據您的業務領域及其各自功能設計工作負載。專注於特定功能讓您能夠區別不同服務的可靠性要求，並更集中瞄準投資目標。簡要的業務問題和與各服務相關的小型團隊，也更容易讓組織擴展。

- 進行領域分析，以便對應工作負載的領域驅動設計 (DDD)。然後您可以選擇符合工作負載需求的架構類型。
  - 如何整合型服務分成微型服務
  - 遭到舊式系統包圍時開始使用 DDD
  - Eric Evans「領域驅動設計：解決軟件核心的複雜性」
  - 實作 AWS 上的微型服務
- 將您的服務分解為最小的元件。您可以使用微型服務架構，將工作負載劃分具有最小的功能的多個元件，以實現組織擴展和敏捷性。
  - 定義工作負載的 API 及其設計目標、限制和其他使用考量。
    - 定義 API。
      - API 定義應考慮增長和其他參數。
    - 定義設計的可用性。
      - 您的 API 可能針對不同功能具有多個設計目標。
    - 建立限制
      - 使用測試來定義工作負載功能的限制。

資源

相關文件：

- Amazon API Gateway：使用 OpenAPI 設定 REST API
- 有界限的環境 (領域驅動設計的集中模式)
- Eric Evans「領域驅動設計：解決軟件核心的複雜性」
- 遭到舊式系統包圍時開始使用 DDD
- 如何整合型服務分成微型服務
- 實作 AWS 上的微型服務
- 微型服務權衡
- 微型服務 - 此新架構術語的定義
- AWS 上的微型服務

REL03-BP03 每個 API 都提供服務合約

服務合約為服務整合團隊間的明訂記載的協議，並包括電腦可讀取的 API 定義、速率限制和效能期望。版本控制策略可讓您的用戶端繼續使用現有 API，並在準備好時將應用程式遷移至更新的 API。只要不違反合約，隨時都可進行部署。服務供應商團隊可以使用自己選擇的技術堆疊，以滿足 API 合約要求。同樣地，服務取用者可以使用自有的技術。

微型服務採用此服務導向架構 (SOA) 的概念，進而建立擁有最少功能組的服務。每個服務均會發佈一個 API 及使用該服務的設計目標、限制和其他考量事項。這可與 呼叫應用程式 建立合約。這樣即可實現三個主要優勢：

- 該服務包含一個待處理的簡明業務問題，以及一個存在業務問題的小團隊。這樣一來，將能更好地進行組織擴展。
- 只要符合 API 和合約要求，團隊能隨時進行部署。
- 只要符合 API 和合約要求，團隊能隨時使用任何他們想要的技術堆疊。

Amazon API Gateway 是一項全受管的服務，可讓開發人員輕鬆地建立、發佈、維護、監控和保護任何規模的 API。涉及接受和處理多達數十萬個並行 API 呼叫 (包括流量管理、授權與存取控制、監控和 API 版本管理) 的所有任務均由 API Gateway 處理。您可以使用 OpenAPI Specification (OAS) (先前稱為 Swagger Specification)，定義 API 合約並將其匯入至 API Gateway。您之後可以使用 API Gateway 進行 API 的版本控制和部署作業。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 每個 API 都提供服務合約：服務合約為服務整合團隊間的明訂記載的協議，並包括電腦可讀取的 API 定義、速率限制和效能期望。
  - Amazon API Gateway：使用 OpenAPI 設定 REST API
    - 版本控制策略可讓用戶端繼續使用現有 API，並在準備好時將應用程式遷移至更新的 API。
    - Amazon API Gateway 是一種全受管的服務，可讓開發人員輕鬆地建立任何規模的 API。您可以使用 OpenAPI Specification (OAS) (先前稱為 Swagger Specification)，定義 API 合約並將其匯入至 API Gateway。您之後可以使用 API Gateway 進行 API 的版本控制和部署作業。

資源

相關文件：

- Amazon API Gateway：使用 OpenAPI 設定 REST API
- 有界限的環境 (領域驅動設計的集中模式)
- 實作 AWS 上的微型服務
- 微型服務權衡
- 微型服務 - 此新架構術語的定義
- AWS 上的微型服務

REL 4 如何在分散式系統中設計防止失敗的互動？

分散式系統倚賴通訊網路來互連元件，例如伺服器或服務。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務可防止失敗，並延長平均失敗間隔時間 (MTBF)。

最佳實務

- REL04-BP01 確定需要哪種分散式系統
- REL04-BP02 實作鬆散耦合相依性
- REL04-BP03 持續執行工作
- REL04-BP04 將所有回應設為等冪

REL04-BP01 確定需要哪種分散式系統

硬式即時分散式系統需要同步、快速給予回應，而軟式即時系統則可以在更長的時段 (分鐘) 內來回應。離線系統會透過批次或非同步處理來處理回應。硬式即時分散式系統具有最嚴格的可靠性要求。

對於硬式即時分散式系統而言， 分散式系統最困難的挑戰 也稱為請求/回覆服務。較困難的是，無法預測請求何時抵達，且必須快速回應 (例如，客戶正在主動等待回應)。範例包括前端 Web 伺服器、訂單管道、信用卡交易、每個 AWS API 和電話語音。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 識別需要哪種分散式系統。分散式系統的挑戰包含延遲、擴展、了解聯網 API、資料編組和解編，以及 Paxos 等演算法的複雜性。隨著系統擴大並且益加分散，理論上的極端案例也變成經常發生的案例。
  - Amazon Builders' Library：分散式系統的挑戰

- 硬式即時分散式系統需要同步、快速給予回應。
- 軟式即時系統則可以在更長的時段 (分鐘) 內來回應。
- 離線系統會透過批次或非同步處理來處理回應。
- 硬式即時分散式系統具有最嚴格的可靠性要求。

資源

相關文件：

- Amazon EC2：確保等冪性
- Amazon Builders' Library：分散式系統的挑戰
- Amazon Builders' Library：可靠性、持續工作，以及咖啡時刻
- 什麼是 Amazon EventBridge？
- 什麼是 Amazon Simple Queue Service？

相關影片：

- 2019 年 AWS 紐約高峰會：事件驅動架構和 Amazon EventBridge 簡介 (MAD205)
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括鬆耦合、持續工作、靜態穩定性)
- AWS re:Invent 2019：移至事件驅動架構 (SVS308)

REL04-BP02 實作鬆散耦合相依性

佇列系統、串流系統、工作流程和負載平衡器之間具有鬆散耦合的相依性。鬆耦合有助於將某個元件的行為與依賴它的其他元件隔離，進而提高彈性和敏捷性。

如果一個元件的變更迫使依賴它的其他元件也變更，則屬於 緊密 耦合。 鬆 耦合會破壞此相依性，因此相依元件只需要知道受版本控制的和已發佈的界面。在相依性之間實作鬆耦合，可避免一個元件中的故障影響另一個元件。

鬆耦合可讓您將其他程式碼或功能新增至某個元件，同時將依賴該元件的其他元件的風險降至最低。此外，您可以向外擴展甚或變更相依性的基礎實作，因此可擴展性也會得到提升。

若要透過鬆耦合進一步改善彈性，請盡可能讓元件採用非同步互動。此模型適用於不需要立即回應的任何互動，以及確認已註冊請求便以足夠的狀況。它涉及產生事件的一個元件和取用事件的另一個元件。

這兩個元件不會透過點對點直接互動來整合，但通常會透過中繼耐用儲存層來整合，例如 SQS 佇列，或如 Amazon Kinesis 或 AWS Step Functions 等串流資料平台。

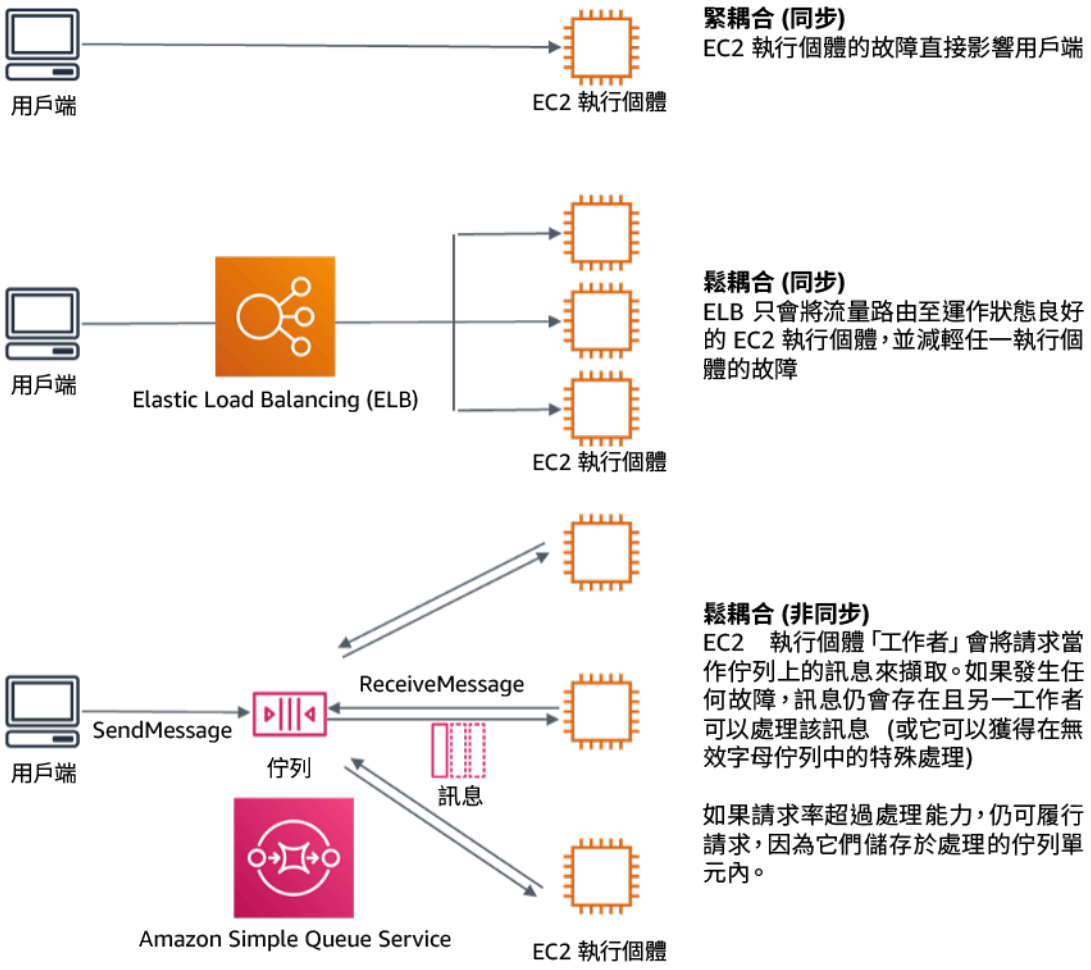


圖 4：佇列系統和負載平衡器之間具有鬆散耦合的相依性。

Amazon SQS 佇列和 Elastic Load Balancer 只是為鬆耦合新增中繼層的兩種方式。事件驅動型架構也可以使用 Amazon EventBridge 在 AWS 雲端 建置。其可從用戶端依賴的服務 (事件消費者) 中抽取用戶端 (事件生產者)。當您需要高輸送量、推送架構的多對多傳訊時，Amazon Simple Notification Service (Amazon SNS) 是有效的解決方案。使用 Amazon SNS 主題，您的發佈者系統可以將訊息散發給大量訂閱者端點，以進行平行處理。

雖然佇列提供多項優勢，但在大多數硬式即時系統中，超過閾值時間 (通常為秒) 的請求應視為過時 (用戶端已放棄且不再等待回應) 且未處理。這樣才可以處理較新的 (且可能仍有效的) 請求。

常用的反模式：

- 將單例部署為工作負載的一部分。

- 在工作負載層之間直接叫用 API，沒有容錯移轉或非同步處理請求的功能。

建立此最佳實務的優勢： 鬆耦合有助於將某個元件的行為與依賴它的其他元件隔離，進而提高彈性和敏捷性。避免一個元件中的失敗影響其他元件。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 實作鬆散耦合相依性。佇列系統、串流系統、工作流程和負載平衡器之間具有鬆散耦合的相依性。鬆耦合有助於將某個元件的行為與依賴它的其他元件隔離，進而提高彈性和敏捷性。
  - AWS re:Invent 2019：移至事件驅動架構 (SVS308)
  - 什麼是 Amazon EventBridge？
  - 什麼是 Amazon Simple Queue Service？
    - Amazon EventBridge 可讓您建置鬆散耦合和分散式的事件驅動型架構。
      - 2019 年 AWS 紐約高峰會：事件驅動架構和 Amazon EventBridge 簡介 (MAD205)
    - 如果一個元件的變更迫使依賴它的其他元件也變更，則屬於緊密耦合。鬆耦合會破壞此相依性，因此相依元件只需要知道受版本控制的和已發佈的界面。
    - 盡可能讓元件採用非同步互動。此模型適用於不需要立即回應的任何互動，以及確認已註冊請求便以足夠的狀況。
      - AWS re:Invent 2019：使用 Amazon SQS 和 Lambda 的可擴展無伺服器事件驅動應用程式 (API304)

資源

相關文件：

- AWS re:Invent 2019：移至事件驅動架構 (SVS308)
- Amazon EC2：確保等冪性
- Amazon Builders' Library：分散式系統的挑戰
- Amazon Builders' Library：可靠性、持續工作，以及咖啡時刻
- 什麼是 Amazon EventBridge？
- 什麼是 Amazon Simple Queue Service？

相關影片：

- 2019 年 AWS 紐約高峰會：事件驅動架構和 Amazon EventBridge 簡介 (MAD205)
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括鬆耦合、持續工作、靜態穩定性)
- AWS re:Invent 2019：移至事件驅動架構 (SVS308)
- AWS re:Invent 2019：使用 Amazon SQS 和 Lambda 的可擴展無伺服器事件驅動應用程式 (API304)

REL04-BP03 持續執行工作

負載大幅快速變更時，系統可能會發生故障。例如，如果您的工作負載正在執行運作狀態檢查，監控數千部伺服器的運作狀態，應該每次傳送相同大小的承載 (目前狀態的完整快照)。無論伺服器全無故障或全部出現故障，運作狀態檢查系統都會持續執行工作，而無大幅快速變更。

例如，如果運作狀態檢查系統正在監控 100,000 部伺服器，則在一般輕型伺服器失敗率下，其負載為額定值。不過，如果重大事件讓一半的伺服器運作狀況不良，則運作狀態檢查系統會因嘗試更新通知系統並向其用戶端溝通狀態，而承受不住負載。因此，運作狀態檢查系統應每次都傳送目前狀態的完整快照。100,000 個伺服器運作狀態 (每個以一位元表示) 只是 12.5 KB 的承載。無論沒有伺服器發生故障，還是全部發生故障，運作狀態檢查系統都會持續執行工作，而大型的快速變更也不會對系統穩定性造成威脅。這實際上是 Amazon Route 53 處理端點 (例如 IP 地址) 的運作狀態檢查，以判斷最終使用者如何路由到其中的方式。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 執行持續工作，以便負載大量快速變更時，系統不會失敗。
- 實作鬆散耦合相依性。佇列系統、串流系統、工作流程和負載平衡器之間具有鬆散耦合的相依性。鬆耦合有助於將某個元件的行為與依賴它的其他元件隔離，進而提高彈性和敏捷性。
  - Amazon Builders' Library：可靠性、持續工作，以及咖啡時刻
  - AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括持續工作)
    - 針對運作狀態檢查系統監控 100,000 部伺服器的範例，將工作負載設計為無論成功或失敗的數量為何，承載大小都保持不變。

資源

相關文件：

- Amazon EC2：確保等冪性

- Amazon Builders' Library：分散式系統的挑戰
- Amazon Builders' Library：可靠性、持續工作，以及咖啡時刻

相關影片：

- 2019 年 AWS 紐約高峰會：事件驅動架構和 Amazon EventBridge 簡介 (MAD205)
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括持續工作)
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括鬆耦合、持續工作、靜態穩定性)
- AWS re:Invent 2019：移至事件驅動架構 (SVS308)

REL04-BP04 將所有回應設為等冪

等冪服務承諾每個請求只完成一次，使得發出多個相同請求與發出單一請求具有相同的效果。等冪服務可讓用戶端更輕鬆地實作重試，而不用擔心錯誤地多次處理請求。為此，用戶端可以使用等冪權杖發出 API 請求，即每次重複請求時，都會使用相同的權杖。等冪服務 API 會使用權杖來傳回與第一次完成請求時傳回之回應相同的回應。

在分散式系統中，執行最多一次動作 (用戶端只發出一個請求) 或至少一次動作 (持續發出請求，直到用戶端確認成功) 很容易。但很難保證動作是等冪的，這表示它 只執行一次， 使得發出多個相同的請求與發出單一請求具有相同效果。透過在 API 中使用等冪性權杖，服務可以收到一次或多次變異請求，而不會產生重複的記錄或副作用。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 將所有回應設為等冪。等冪服務承諾每個請求只完成一次，使得發出多個相同請求與發出單一請求具有相同的效果。
  - 用戶端可以使用等冪權杖發出 API 請求，即每次重複請求時，都會使用相同的權杖。等冪服務 API 會使用權杖來傳回與第一次完成請求時傳回之回應相同的回應。
    - Amazon EC2：確保等冪性

資源

相關文件：

- Amazon EC2：確保等冪性
- Amazon Builders' Library：分散式系統的挑戰
- Amazon Builders' Library：可靠性、持續工作，以及咖啡時刻

相關影片：

- 2019 年 AWS 紐約高峰會：事件驅動架構和 Amazon EventBridge 簡介 (MAD205)
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權 (ARC337) (包括鬆耦合、持續工作、靜態穩定性)
- AWS re:Invent 2019：移至事件驅動架構 (SVS308)

REL 5 如何設計分散式系統中的互動以緩解或承受故障？

分散式系統倚賴通訊網路來互連元件 (例如，伺服器或服務)。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務讓工作負載能夠承受壓力或故障，更快速地從其中復原，並減輕這類受損的影響。最終縮短平均復原時間 (MTTR)。

最佳實務

- REL05-BP01 實作適度降級以將適用的硬相依性轉換為軟相依性
- REL05-BP02 調節請求
- REL05-BP03 控制和限制重試呼叫
- REL05-BP04 快速失敗和限制佇列
- REL05-BP05 設定用戶端逾時
- REL05-BP06 盡可能讓服務無狀態
- REL05-BP07 實作緊急控制桿

REL05-BP01 實作適度降級以將適用的硬相依性轉換為軟相依性

當元件的相依性狀況不良，元件本身仍可運作，但以降級的方式運作。例如，當相依性呼叫失敗時，容錯移轉為預先決定的靜態回應。

考慮由服務 A 呼叫的服務 B，並接著呼叫服務 C。

圖 5：從服務 B 呼叫服務 C 時失敗。服務 B 傳回降級回應給服務 A。

當服務 B 呼叫服務 C 時，其會從服務 C 收到錯誤或逾時。缺少來自服務 C 回應 (及其包含的資料) 的服務 B，會傳回其所能執行的回應。這可能是上次快取的良好值，或者服務 B 可使用預先決定的靜態回應，取代原可能從服務 C 收到的內容。然後它可以將降級的回應傳回給發起人，即服務 A。如果沒有此靜態回應，則服務 C 中的故障會透過服務 B 串聯到服務 A，導致失去可用性。

根據硬相依性可用性方程式中的乘法因數 (請參閱 透過硬性相依性計算可用性)，C 可用性的任何下降都會嚴重影響 B 的有效可用性。透過傳回靜態回應，服務 B 可減輕 C 的故障；而且雖然降級，但使服務 C 的可用性看起來像 100% (假設它在錯誤情況下可靠地傳回靜態回應)。請注意，靜態回應是傳回錯誤的簡單替代方法，並不會嘗試以不同方式重新計算回應。這種以完全不同的機制來嘗試達到相同結果的嘗試稱為備用行為，並且是一種可避免的反模式。

適度降級的另一個範例是 斷路器模式。當故障是暫時性時，應該使用重試策略。如果情況並非如此，且操作可能會失敗，則斷路器模式會阻止用戶端執行可能失敗的請求。在正常處理請求時，斷路器合閘，請求流過。當遠端系統開始傳回錯誤或出現高延遲時，斷路器會開啟，並忽略相依性，或是將結果取代為更輕鬆取得但不完整的回應 (可能只是回應快取)。系統會定期嘗試呼叫該相依性，以確定其是否已復原。發生這種情況時，斷路器將閉合。

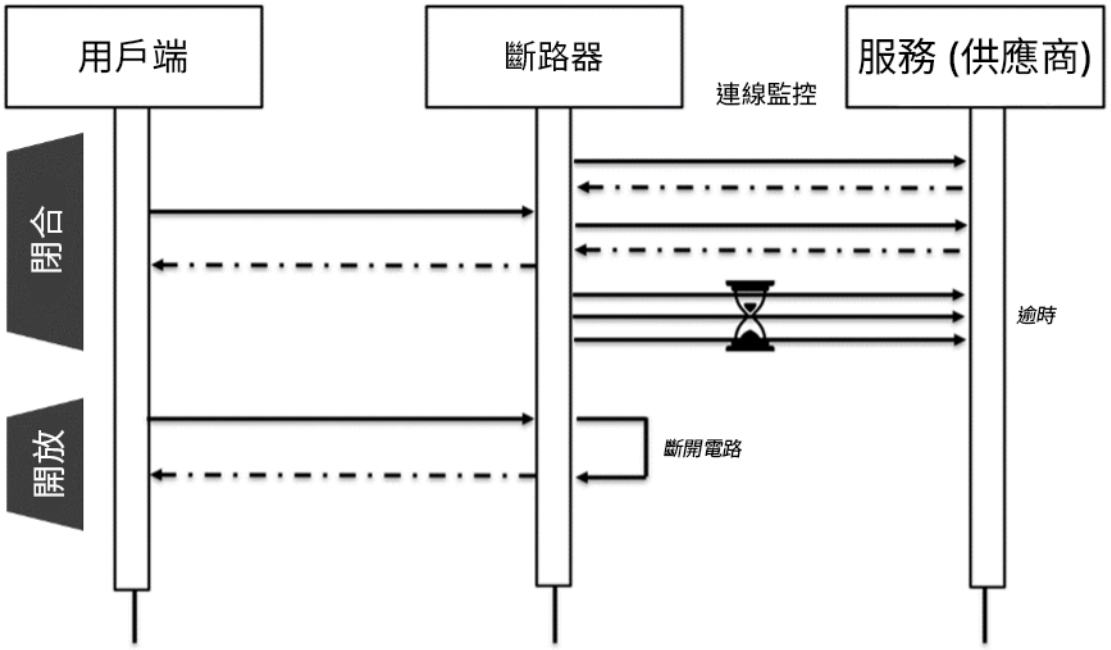


圖 6：顯示關閉和開啟狀態的斷路器。

除圖中所示的關閉和開啟狀態外，在可設定的期間後，斷路器可在開啟狀態下轉換為半開啟狀態。在此狀態下，它會定期嘗試以比平常低得多的速率來呼叫服務。此探查用於檢查服務的運作狀態。在半開啟狀態下成功數次之後，斷路器會轉換為關閉，並恢復正常請求。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 實作適度降級以將適用的硬相依性轉換為軟相依性。當元件的相依性狀況不良，元件本身仍可運作，但以降級的方式運作。例如，當相依性呼叫失敗時，容錯移轉為預先決定的靜態回應。
  - 透過傳回靜態回應，您的工作負載可減輕在其相依性中發生的故障。
    - Well-Architected 實驗室：第 300 級：實作運作狀態檢查和管理相依性以提升可靠性
  - 偵測重試操作何時可能失敗，並防止用戶端以斷路器模式進行失敗的呼叫。
    - CircuitBreaker

資源

相關文件：

- Amazon API Gateway：調節 API 請求以獲得更佳的輸送量

- CircuitBreaker (摘要說明「Release It!」書籍中的斷路器)
- AWS 中的錯誤重試和指數退避
- Michael Nygard「Release It! 設計和部署生產就緒型軟體」
- Amazon Builders' Library：避免分散式系統的備用
- Amazon Builders' Library：避免無法逾越的佇列待辦項目
- Amazon Builders' Library：快取挑戰和策略
- Amazon Builders' Library：逾時、重試、退避與抖動

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

相關範例：

- Well-Architected 實驗室：第 300 級：實作運作狀態檢查和管理相依性以提升可靠性

REL05-BP02 調節請求

調節請求是一種緩解模式，用於回應意外增加的需求。有些請求會接受，但超過定義限制的請求會遭到拒絕，並傳回訊息，指出它們已受到調節。預期用戶端會退避並放棄請求，或以較慢的速率再試一次。

您的服務應設計為處理每個節點或儲存格可以處理的已知請求容量。此容量可以透過負載測試來建立。然後，您需要追蹤請求的到達率，如果臨時到達率超過此限制，則適當的回應是發出訊號，指出已對其進行調節。其讓使用者可以進行重試，而重試可能具有可用容量的其他節點或儲存格。Amazon API Gateway 提供了調節請求的方法。Amazon SQS 和 Amazon Kinesis 可以緩衝請求、平滑請求率，以及減少對可非同步處理的請求進行調節的需求。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 調節請求。這是一種緩解模式，用於回應意外增加的需求。有些請求會接受，但超過定義限制的請求會遭到拒絕，並傳回訊息，指出它們已受到調節。預期用戶端會退避並放棄請求，或以較慢的速率再試一次。
  - 使用 Amazon API Gateway
    - 調節 API 請求以獲得更佳的輸送量

資源

相關文件：

- Amazon API Gateway：調節 API 請求以獲得更佳的輸送量
- AWS 中的錯誤重試和指數退避
- Amazon Builders' Library：避免分散式系統的備用
- Amazon Builders' Library：避免無法逾越的佇列待辦項目
- Amazon Builders' Library：逾時、重試、退避與抖動
- 調節 API 請求以獲得更佳的輸送量

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

REL05-BP03 控制和限制重試呼叫

使用指數退避以在逐漸延長間隔後重試。引進抖動來隨機化這些重試間隔，並限制重試次數上限。

分散式軟體系統中的典型元件包括伺服器、負載平衡器、資料庫和 DNS 伺服器。在操作中，且操作可能失敗時，任何這些項目都可能開始產生錯誤。處理錯誤的預設技術是在用戶端實作重試。此技術可提高應用程式的可靠性和可用性。不過，如果用戶端在發生錯誤時立即嘗試重試失敗的操作，則網路很快就會因為新的和重試的請求而大規模地變得飽和，且每個請求都會爭奪網路頻寬。這可能會導致 重試暴風， 而該情況會降低服務的可用性。此模式可能會持續進行，直到整個系統出現故障為止。

若要避免這類情境，應該使用一般指數退避等 退避演算法 。指數退避演算法會逐漸降低執行重試的速率，從而避免網路擁塞。

許多開發套件和軟體程式庫 (包括來自 AWS 的程式庫) 都會實作這些演算法的一個版本。不過， 絕對不要假設退避演算法存在，請一律測試並驗證是否如此。

單靠簡單退避是不夠的，因為在分散式系統中，所有用戶端都可能會同時退避，從而建立重試呼叫的叢集。Marc Brooker 在其部落格文章 指數退避和抖動，說明了如何修改指數退避中的 wait () 函數，以防止重試呼叫的叢集。解決 方法 是在 wait() 函數中新增抖動。為避免重試太久，實作時應該將退避上限設為最大值。

最後，請務必設定 最大重試次數 或經過時間，之後重試就會失敗。AWS 開發套件預設情況下可實作此功能，並可以對其進行設定。對於堆疊中較低的服務，最大重試限制為零或一時可限制風險，但仍然有效，因為重試會委派給堆疊中較高的服務。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 控制和限制重試呼叫。使用指數退避以在逐漸延長間隔後重試。引進抖動來隨機化這些重試間隔，並限制重試次數上限。
  - AWS 中的錯誤重試和指數退避
    - Amazon SDK 預設會實作重試和指數退避。呼叫自己的相依服務時，在相依性層中實作類似的邏輯。根據您的使用案例確定逾時時間以及何時停止重試。

資源

相關文件：

- Amazon API Gateway：調節 API 請求以獲得更佳的輸送量
- AWS 中的錯誤重試和指數退避
- Amazon Builders' Library：避免分散式系統的備用
- Amazon Builders' Library：避免無法逾越的佇列待辦項目
- Amazon Builders' Library：快取挑戰和策略
- Amazon Builders' Library：逾時、重試、退避與抖動

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

REL05-BP04 快速失敗和限制佇列

如果工作負載無法成功回應請求，則快速失敗。如此將可釋放與請求關聯的資源，並且使服務在資源用盡時復原。如果工作負載能成功回應，但請求率太高，則改為使用佇列來緩衝請求。不過，請勿允許可能導致處理用戶端已放棄的過時請求之長佇列。

此最佳實務適用於該請求的伺服器端或接收者。

請注意，佇列可以在系統的多個層級建立，而且可能會嚴重阻礙快速復原的能力，因為較舊的過時請求 (不再需要回應) 在較新的請求之前處理。請注意佇列存在的位置。它們通常隱藏在記錄至資料庫的工作流程或工作中。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 快速失敗和限制佇列。如果工作負載無法成功回應請求，則快速失敗。如此將可釋放與請求關聯的資源，並且使服務在資源用盡時復原。如果工作負載能成功回應，但請求率太高，則改為使用佇列來緩衝請求。不過，請勿允許可能導致處理用戶端已放棄的過時請求之長佇列。
  - 服務受壓時實作快速失敗。
    - 快速失敗
  - 限制佇列：在佇列式系統中，當處理停止但訊息持續送達時，待處理訊息可能大量積存，使得處理時間增加。工作可能太晚完成而無效，基本上會導致佇列要防範的可用性問題。
    - Amazon Builders' Library：避免無法逾越的佇列待辦項目

資源

相關文件：

- AWS 中的錯誤重試和指數退避
- 快速失敗
- Amazon Builders' Library：避免分散式系統的備用
- Amazon Builders' Library：避免無法逾越的佇列待辦項目
- Amazon Builders' Library：快取挑戰和策略
- Amazon Builders' Library：逾時、重試、退避與抖動

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

REL05-BP05 設定用戶端逾時

適當設定逾時、系統性對其進行驗證，並且不要依賴預設值，因為它們通常設定得太高。

此最佳實務適用於請求的用戶端或寄件者。

針對任何遠端呼叫 (通常為跨程序的任何呼叫) 同時設定連線逾時和請求逾時。許多框架都提供內建的逾時功能，但請注意，許多框架都有無限或過高的預設值。太高的值會降低逾時的實用性，因為當用戶端等待逾時發生時，資源會持續耗用。太低的值可能會增加後端流量和延遲，原因是重試的請求過多。在某些情況下，這可能導致完全停機，原因是正在重試所有請求。

若要進一步了解 Amazon 如何透過抖動使用逾時、重試和退避功能，請參閱 Builder's Library：逾時、重試、退避與抖動。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 針對任何遠端呼叫 (通常為跨程序的任何呼叫) 同時設定連線逾時和請求逾時。許多框架都提供內建的逾時功能，但請注意，許多框架都有無限或過高的預設值。太高的值會降低逾時的實用性，因為當用戶端等待逾時發生時，資源會持續耗用。太低的值可能會增加後端流量和延遲，原因是重試的請求過多。在某些情況下，這可能導致完全停機，原因是正在重試所有請求。
  - AWS SDK：重試與逾時

資源

相關文件：

- AWS SDK：重試與逾時
- Amazon API Gateway：調節 API 請求以獲得更佳的輸送量
- AWS 中的錯誤重試和指數退避
- Amazon Builders' Library：逾時、重試、退避與抖動

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

REL05-BP06 盡可能讓服務無狀態

服務不應要求狀態，或應該卸載狀態，以便在不同的用戶端請求之間，不依賴磁碟上和記憶體中本機儲存的資料。這讓伺服器能夠任意置換，而不會對可用性造成影響。Amazon ElastiCache 或 Amazon DynamoDB 是卸載狀態的適當目的地。

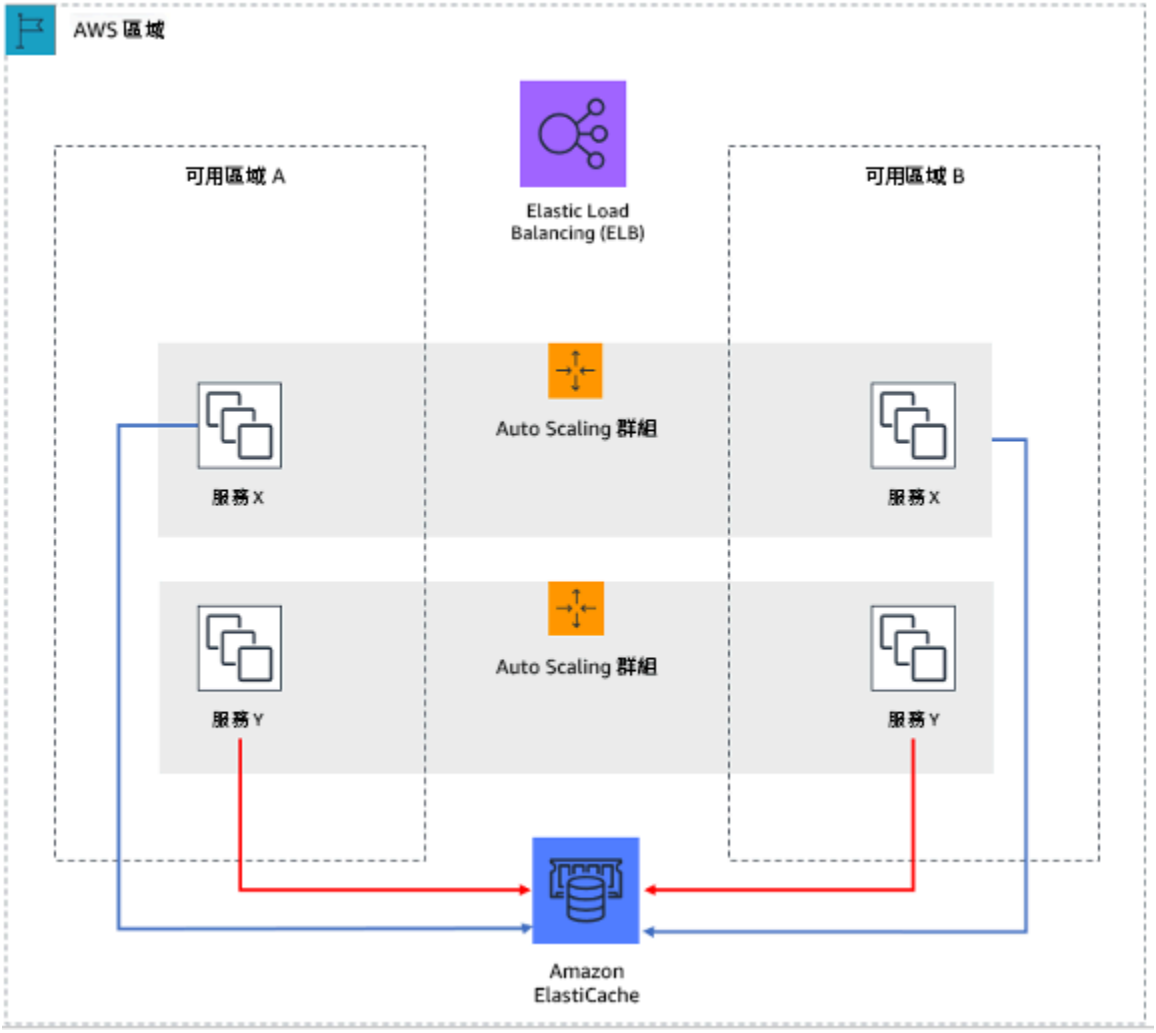


圖 7：在這個無狀態的 Web 應用程式中，工作階段狀態會卸載至 Amazon ElastiCache。

當使用者或服務與應用程式互動時，他們通常會執行形成工作階段的一系列互動。工作階段是使用者在使用應用程式時，在不同請求之間持續存在的唯一資料。無狀態應用程式是一種不需要了解先前互動，也不會儲存工作階段資訊的應用程式。

一旦設計為無狀態，您就可以使用 AWS Lambda 或 AWS Fargate 等無伺服器運算服務。

除了伺服器替換，無狀態應用程式的另一個好處是它們可以水平擴展，因為任何可用的運算資源 (例如，EC2 執行個體和 AWS Lambda 函數) 都可以處理所有請求。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 讓您的應用程式無狀態。無狀態應用程式支援水平擴展，並且可以容忍單個節點的故障。

- 移除實際上可以儲存在請求參數中的狀態。
- 在檢查了是否需要狀態之後，將狀態追蹤移至彈性多可用區域資料儲存，例如 Amazon ElastiCache、Amazon RDS、Amazon DynamoDB 或第三方分散式資料解決方案。儲存無法移動到彈性資料儲存的狀態。
  - 某些資料 (如 Cookie) 可以在標頭或查詢參數中傳遞。
  - 重構以移除可以在請求中快速傳遞的狀態。
  - 某個請求可能實際上並不需要某些資料，這些資料可以隨需擷取。
  - 移除可以非同步擷取的資料。
  - 確定滿足所需狀態要求的資料儲存。
  - 考慮將 NoSQL 資料庫用於非關聯式資料。

資源

相關文件：

- Amazon Builders' Library：避免分散式系統的備用
- Amazon Builders' Library：避免無法逾越的佇列待辦項目
- Amazon Builders' Library：快取挑戰和策略

REL05-BP07 實作緊急控制桿

緊急控制桿是可緩解對工作負載的可用性影響的快速程序。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 實作緊急控制桿。這是可緩解對工作負載的可用性影響的快速程序。它們可以在沒有根本原因的情況下操作。理想的緊急控制桿會提供完全決定性啟用和停用準則，將解析器的認知負擔降至零。控制桿通常是手動的，但也可以自動化
  - 範例控制桿包括
    - 封鎖所有機器人流量
    - 提供靜態頁面而非動態頁面
    - 減少對相依性的呼叫頻率
    - 調節來自相依性的呼叫

- 實作和使用緊急控制桿的秘訣
  - 當啟用控制桿時，請少做，而非多做
  - 保持簡單，避免雙模式行為
  - 定期測試您的控制桿
- 以下是非緊急控制桿動作的範例
  - 新增容量
  - 呼叫依賴您服務的用戶端服務擁有者，並要求他們減少呼叫
  - 變更程式碼並將其釋出

變更管理

問題

- REL 6 如何監控工作負載資源？
- REL 7 如何設計工作負載以適應需求變更？
- REL 8 您如何實作變更？

REL 6 如何監控工作負載資源？

日誌和指標是深入了解工作負載運作狀態的強大工具。您可以設定工作負載以監控日誌和指標，並在超過閾值或發生重大事件時傳送通知。監控可讓您的工作負載識別何時會超過低效能閾值或發生故障，以便自動復原來回應。

最佳實務

- REL06-BP01 監控工作負載的所有元件 (產生)
- REL06-BP02 定義和計算指標 (彙總)
- REL06-BP03 傳送通知 (即時處理和警示)
- REL06-BP04 自動化回應 (即時處理和警示)
- REL06-BP05 分析
- REL06-BP06 定期進行審查
- REL06-BP07 透過您的系統監控請求的端對端追蹤

REL06-BP01 監控工作負載的所有元件 (產生)

使用 Amazon CloudWatch 或第三方工具監控工作負載的元件。使用 AWS Health 儀表板監控 AWS 服務。

工作負載的所有元件都應該受到監控，包括前端、商業邏輯和儲存層。定義關鍵指標，描述如何從日誌擷取指標 (如果需要)，以及設定觸發對應警示事件的閾值。確保指標與工作負載的關鍵績效指標 (KPI) 相關，並使用指標和日誌來識別服務降級的早期預警訊號。例如，與業務成果相關的指標 (例如每分鐘成功處理的訂單數目) 可以比 CPU 使用率這類的技術指標更快地指出工作負載問題。使用 AWS Health 儀表板可針對 AWS 資源下 AWS 服務的效能和可用性，取得個人化檢視。

雲端監控提供新機遇。大部分雲端供應商都開發了可自訂的掛鉤，並且可以提供洞察力來協助您監控多層的工作負載。AWS 服務 (例如 Amazon CloudWatch) 會套用統計和機器學習演算法，以持續分析系統和應用程式的指標、決定正常基準，以及顯現使用者介入最少的異常。異常偵測演算法會考慮指標的季節性和趨勢變更。

AWS 提供大量可用於消費的監控和日誌資訊，這些資訊可以用來定義工作負載特有的指標、按需變更流程，以及採用機器學習技術，而不管 ML 專業知識為何。

此外，監控所有外部端點，以確保它們獨立於基本實作。此主動監控可透過綜合交易 (有時稱為 使用者 Canary，但請別與 Canary 部署混淆) 加以完成，後者會定期執行應用程式消費者執行的一些常見任務。在持續時間中讓這些任務保持簡單扼要，並確定在測試期間不會讓工作負載超載。Amazon CloudWatch Synthetics 讓您能夠 建立綜合 Canary 以監控您的端點和 API。您也可以將綜合性 Canary 用戶端節點與 AWS X-Ray 主控台結合，以指出綜合性 Canary 在所選時段內發生錯誤、故障或調節率等問題。

預期成果：

收集和使用來自工作負載所有元件的關鍵指標，以確保工作負載可靠性和最佳使用者體驗。偵測到工作負載未實現業務成果可讓您快速宣佈災難並從事故中復原。

常用的反模式：

- 僅監控工作負載的外部界面。
- 不產生任何工作負載特有的指標，而且僅依賴工作負載使用的 AWS 服務提供給您的指標。
- 僅在工作負載中使用技術指標，而且不監控與工作負載貢獻的非技術 KPI 相關的任何指標。
- 依賴生產流量和簡單的運作狀態檢查來監控和評估工作負載狀態。

建立此最佳實務的優勢： 工作負載中的所有層級監控，可讓您更快速地預測和解決構成工作負載之元件中的問題。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

1. 在可用的地方啟用記錄。 應該從工作負載的所有元件中取得監控資料。開啟額外記錄 (例如 S3 存取日誌)，並讓您的工作負載可以記錄工作負載特定資料。從 Amazon ECS、Amazon EKS、Amazon EC2、Elastic Load Balancing、AWS Auto Scaling 和 Amazon EMR 等服務中收集 CPU、網路 I/O 和磁碟 I/O 平均值的指標。請參閱 發佈 CloudWatch 指標的 AWS 服務 取得將指標發佈至 CloudWatch 的 AWS 服務清單。
2. 審查所有預設指標並探索任何資料收集差距。 每個服務都會產生預設指標。收集預設指標可讓您更好地了解工作負載元件之間的相依性，以及元件可靠性和效能如何影響工作負載。您也可以建立 自己的指標並將其 發佈至 CloudWatch，方法為使用 AWS CLI 或 API。此
3. 評估所有指標，以判斷哪些指標要對工作負載中的每個 AWS 發出提醒。 您可以選擇要選取對工作負載可靠性有重大影響的指標子集。專注於關鍵指標和閾值可讓您微調 提醒 數目，並可以協助將誤判的情形減至最少。
4. 定義提醒以及在觸發提醒之後工作負載的復原流程。 定義提醒可讓您快速通知、呈報並遵循必要的步驟，從事故中復原並符合您指定的復原時間點目標 (RTO)。您可以使用 Amazon CloudWatch 警示，叫用自動化工作流程，並根據定義的閾值啟動復原程序。
5. 探索如何使用綜合交易來收集有關工作負載狀態的相關資料。 綜合監控會遵循相同的路由並執行與客戶相同的動作，這可讓您持續驗證您的客戶體驗，即使您的工作負載上沒有任何客戶流量也一樣。使用 綜合交易，您可以在客戶探索問題之前先行探索。

資源

相關的最佳實務：

- REL11-BP03 將所有分層的修復自動化

相關文件：

- AWS Health 儀表板入門 – 您的帳戶運作狀態
- 發佈 CloudWatch 指標的 AWS 服務
- Network Load Balancer 的存取日誌
- Application Load Balancer 的存取日誌
- 存取 Amazon CloudWatch Logs 的 AWS Lambda
- Amazon S3 伺服器存取記錄

- 啟用 Classic Load Balancer 的存取日誌
- 將日誌資料匯出至 Amazon S3
- 在 Amazon EC2 執行個體上安裝 CloudWatch 代理程式
- 發布自訂指標
- 使用 Amazon CloudWatch 儀表板
- 使用 Amazon CloudWatch 指標
- 使用 Canary (Amazon CloudWatch Synthetics)
- 什麼是 Amazon CloudWatch Logs？

使用者指南：

- 建立軌跡
- 監控 Amazon EC2 Linux 執行個體的記憶體和磁碟指標
- 搭配容器執行個體使用 CloudWatch Logs
- VPC Flow Logs
- 什麼是 Amazon DevOps Guru？
- 什麼是 AWS X-Ray？

相關部落格：

- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯

相關範例和研討會：

- AWS Well-Architected 實驗室：卓越營運 - 相依性監控
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態
- 可觀測性研討會

REL06-BP02 定義和計算指標 (彙總)

視需要儲存日誌資料並套用篩選條件以計算指標，例如特定日誌事件的計數，或是從日誌事件時間戳記計算的延遲。

Amazon CloudWatch 和 Amazon S3 可作為主要的彙總和儲存層。對於某些服務 (例如 AWS Auto Scaling 和 Elastic Load Balancing)，預設會為跨叢集或執行個體的 CPU 負載或平均請求延遲提供預設

指標。對於 VPC Flow Logs 及 AWS CloudTrail 等串流服務，事件資料將轉寄到 CloudWatch Logs，且您需要定義和套用指標篩選條件以從事件資料中擷取指標。這為您提供時間序列資料，而此資料可作為您定義用於觸發提醒之 CloudWatch 警示的輸入。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 定義和計算指標 (彙總)。視需要儲存日誌資料並套用篩選條件以計算指標，例如特定日誌事件的計數，或是從日誌事件時間戳記計算的延遲
  - 指標篩選條件會定義術語與模式，以在傳送到 CloudWatch Logs 的日誌資料中尋找資料。CloudWatch Logs 使用這些指標篩選條件，將日誌資料轉成數值 CloudWatch 指標，讓您可以對其繪製圖表或設定警示。
    - 搜尋和篩選日誌資料
  - 使用受信任的第三方來彙總日誌。
    - 請遵循第三方的指示。大部分第三方產品可與 CloudWatch 和 Amazon S3 整合。
  - 有些 AWS 服務可以直接將日誌發佈到 Amazon S3。如果您的日誌主要需求是儲存在 Amazon S3 中，則可以輕鬆讓產生日誌的服務直接將它們傳送到 Amazon S3，無須設定其他基礎設施。
    - 直接將日誌傳送至 Amazon S3

資源

相關文件：

- Amazon CloudWatch Logs Insights 範例查詢
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯
- 一個觀察工作坊
- 搜尋和篩選日誌資料
- 直接將日誌傳送至 Amazon S3
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態

REL06-BP03 傳送通知 (即時處理和警示)

當重大事件發生時，需要知道的組織會收到通知。

提醒可以傳送到 Amazon Simple Notification Service (Amazon SNS) 主題，然後推送給任何數量的訂閱者。例如，Amazon SNS 可以將提醒轉寄到電子郵件別名，以便技術人員可以回應。

常用的反模式：

- 設定閾值太低的警示，導致傳送太多通知。
- 不封存警示以供未來探索。

建立此最佳實務的優勢： 事件通知 (甚至是可回應和自動解決的通知) 可讓您擁有事件紀錄，並在未來可能以不同方式處理事件。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 執行即時處理和警示。當重大事件發生時，需要知道的組織會收到通知
  - Amazon CloudWatch 儀表板是 CloudWatch 主控台中的可自訂首頁，您可用來在單一檢視中監控資源，甚至是監控分散在不同區域的資源。
    - 使用 Amazon CloudWatch 儀表板
  - 當指標超過限制時建立警示。
    - 使用 Amazon CloudWatch 警示

資源

相關文件：

- 一個觀察工作坊
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態
- 使用 Amazon CloudWatch 警示
- 使用 Amazon CloudWatch 儀表板
- 使用 Amazon CloudWatch 指標

REL06-BP04 自動化回應 (即時處理和警示)

偵測到事件時，使用自動化以採取動作，例如取代故障的元件。

提醒可以觸發 AWS Auto Scaling 事件，因此叢集可根據需求便能作出反應。提醒可傳送至 Amazon Simple Queue Service (Amazon SQS)，該服務可以用作第三方票證系統的整合點。AWS Lambda 還可以訂閱提醒，為使用者提供非同步無伺服器模型，以動態回應變更。AWS Config 會持續監控和記錄您的 AWS 資源組態，並可觸發 AWS Systems Manager Automation 以修復問題。

Amazon DevOps Guru 可以自動監控應用程式資源，以偵測異常行為並提供目標建議，以縮短問題識別和矯正時間。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用 Amazon DevOps Guru 來執行自動化動作。Amazon DevOps Guru 可以自動監控應用程式資源，以偵測異常行為並提供目標建議，以縮短問題識別和矯正時間。
  - 什麼是 Amazon DevOps Guru？
- 使用 AWS Systems Manager 來執行自動化動作。AWS Config 會持續監控和記錄您的 AWS 資源組態，並可觸發 AWS Systems Manager 以修復問題。
  - AWS Systems Manager Automation
    - 建立和使用 Systems Manager Automation 文件。當自動化流程執行時，這些會定義 Systems Manager 在受管執行個體和其他 AWS 資源上執行的動作。
    - 與自動化文件搭配使用 (程序手冊)
- Amazon CloudWatch 會將警示狀態變更事件傳送到 Amazon EventBridge。建立 EventBridge 規則以自動化回應。
  - 建立 EventBridge 規則，以透過 AWS 資源觸發事件
- 建立和執行計畫以自動化回應。
  - 清查您的所有提醒回應程序。您必須先規劃提醒回應，再將任務排名。
  - 以必須採取的特定動作清查所有任務。這些動作大多記錄在執行手冊中。您也必須擁有適用於意外事件提醒的程序手冊。
  - 檢查所有適用於自動化動作的執行手冊和程序手冊。一般而言，如果某個動作可以受到定義，則很可能可以進行自動化。
  - 將容易出錯或耗時的活動排在第一位。移除錯誤來源並縮短解決時間是最有益的。
  - 建立完成自動化的計畫。維護作用中的計畫以自動化和更新自動化。
  - 檢查自動化機會的手動需求。挑戰您的手動程序，以找出自動化的機會。

資源

相關文件：

- AWS Systems Manager Automation
- 建立 EventBridge 規則，以透過 AWS 資源觸發事件

- 一個觀察工作坊
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態
- 什麼是 Amazon DevOps Guru？
- 與自動化文件搭配使用 (程序手冊)

REL06-BP05 分析

收集日誌檔和指標歷史記錄，並分析這些檔案和歷史記錄，以了解更廣泛的趨勢和工作負載洞見。

Amazon CloudWatch Logs Insights 支援 簡單但功能強大的查詢語言， 您可使用此語言來分析日誌資料。Amazon CloudWatch Logs 還支援訂閱，而這些訂閱允許資料無縫流至 Amazon S3，您可使用 Amazon S3 或 Amazon Athena 來查詢資料。其也支援對大量格式的查詢。請參閱 請參閱 (位於 Amazon Athena 使用者指南中)，以取得詳細資訊。若要分析大型日誌檔集，您可以執行 Amazon EMR 叢集來執行 PB 級分析。

AWS 合作夥伴和第三方提供了許多工具，可用於彙總、處理、儲存和分析。這些工具包含 New Relic、Splunk、Loggly、Logstash、CloudHealth 和 Nagios。但是，系統和應用程式日誌之外的產生對於每個雲端提供者都是唯一的，並且通常對於每個服務也都是唯一的。

資料管理是監控程序中常常被忽略的部分。您需要確定監控資料的保留要求，然後相應地套用生命週期政策。Amazon S3 可支援 S3 儲存貯體層級的生命週期管理。該生命週期管理能以不同方式套用至儲存貯體中的不同路徑。在生命週期即將結束時，您可以將資料傳輸到 Amazon S3 Glacier 進行長期儲存，然後在保留期結束後到期。S3 智慧型分層儲存類別旨在透過自動將資料移至最經濟實惠的存取層來優化成本，而不會影響效能或營運開銷。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- CloudWatch Logs Insights 可讓您以互動方式在 Amazon CloudWatch Logs 中搜尋和分析日誌資料。
  - 使用 CloudWatch Logs Insights 分析日誌資料
  - Amazon CloudWatch Logs Insights 範例查詢
- 使用 Amazon CloudWatch Logs 將日誌傳送至您可以在其中使用的 Amazon S3，或使用 Amazon Athena 來查詢資料。
  - 我要如何使用 Athena 分析 Amazon S3 伺服器存取日誌？
    - 為您的伺服器存取日誌儲存貯體建立 S3 生命週期政策。設定生命週期政策以定期移除日誌檔案。這樣做可減少 Athena 針對每個查詢所分析的資料量。

- 我要如何為 S3 儲存貯體建立生命週期政策？

資源

相關文件：

- Amazon CloudWatch Logs Insights 範例查詢
- 使用 CloudWatch Logs Insights 分析日誌資料
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯
- 我要如何為 S3 儲存貯體建立生命週期政策？
- 我要如何使用 Athena 分析 Amazon S3 伺服器存取日誌？
- 一個觀察工作坊
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態

REL06-BP06 定期進行審查

經常審查工作負載監控的實作方式，並根據重大事件和變更進行更新。

有效的監控是由關鍵業務指標推動。當業務優先事項變更時，確保您的工作負載中會包含這些指標。

稽核您的監控有助於您知道應用程式何時達到其可用性目標。根本原因分析需要能夠發現發生故障時的具體情況。AWS 提供的服務可讓您在事件發生時追蹤服務狀態：

- Amazon CloudWatch Logs： 您可以將日誌儲存在此服務中並檢查其內容。
- Amazon CloudWatch Logs Insights：是一項全受管服務，讓您可以在數秒內分析大量日誌。其可為您提供快速且互動式的查詢和視覺化。
- AWS Config： 您可以查看在不同時間點使用的 AWS 基礎設施。
- AWS CloudTrail： 您可以查看在什麼時間及透過什麼主體叫用了哪些 AWS API。

在 AWS，我們每週舉行一次會議， 以審查營運效能 及在團隊之間分享經驗。由於 AWS 旗下有太多團隊，我們建立了 The Wheel 以隨機挑選要審查的工作負載。建立定期執行營運效能審查和知識共享的機制，可增強您從營運團隊獲得更高效能的能力。

常用的反模式：

- 僅收集預設指標。
- 設定監控策略，但絕不檢閱。

- 部署重大變更時不討論監控。

建立此最佳實務的優勢： 定期檢閱監控可預期潛在問題，而不是在預期問題實際發生時對通知作出反應。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 為工作負載建立多個儀表板。您必須擁有最上層儀表板，其中包含關鍵業務指標，以及經您確認與工作負載預估運作狀態最相關的 (因為用量不同) 技術指標。您也應該有可以檢查各種應用程式層和相依性的儀表板。
  - 使用 Amazon CloudWatch 儀表板
- 排程及定期檢閱工作負載儀表板。定期執行儀表板檢查。您對於檢查深度可能有不同規律。
  - 檢查指標中的趨勢。比較指標值與歷史值，以查看是否有可能指出某項需要調查的趨勢。這些範例包括：增加延遲、減少主要業務功能，以及增加失敗回應。
  - 檢查指標中的異常值/異常。平均值或中位數可以遮罩異常值。查看時間範圍內的最高和最低值，並調查極端分數的原因。隨著您持續消除這些原因，降低極端的定義可讓您持續改善工作負載效能的一致性。
  - 尋找行為中的急劇變化。指標的數量或方向立即變更，可能表示應用程式有所變更，或您可能需要新增其他指標以追蹤的外部因素。

資源

相關文件：

- Amazon CloudWatch Logs Insights 範例查詢
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯
- 一個觀察工作坊
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態
- 使用 Amazon CloudWatch 儀表板

REL06-BP07 透過您的系統監控請求的端對端追蹤

使用 AWS X-Ray 或第三方工具，讓開發人員能夠更輕鬆地分析和偵錯分散式系統，以了解其應用程式及其基礎服務的執行成效。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 透過您的系統監控請求的端對端追蹤。AWS X-Ray 是一種服務，可收集應用程式處理請求的相關資料，並提供可用於檢視、篩選和取得資料洞見的工具，以識別問題和優化機會。對於任何受追蹤的應用程式請求，您不僅可以查看關於請求和回應的詳細資訊，還可以查看應用程式對下游 AWS 資源、微型服務、資料庫和 Web API 發出的呼叫的詳細資訊。
  - 什麼是 AWS X-Ray？
  - 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯

資源

相關文件：

- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯
- 一個觀察工作坊
- Amazon Builders' Library：偵測分散式系統，以瞭解運作狀態
- 使用 Canary (Amazon CloudWatch Synthetics)
- 什麼是 AWS X-Ray？

REL 7 如何設計工作負載以適應需求變更？

可擴展工作負載提供自動新增或移除資源的彈性，以便隨時盡可能符合目前需求。

最佳實務

- REL07-BP01 取得或擴展資源時使用自動化
- REL07-BP02 在偵測到工作負載受損時取得資源
- REL07-BP03 偵測到工作負載需要更多資源時取得資源
- REL07-BP04 對工作負載執行負載測試

REL07-BP01 取得或擴展資源時使用自動化

替換受損的資源或擴展工作負載時，請使用 Amazon S3 和 AWS Auto Scaling 等受管的 AWS 服務進行自動化程序。您還可以使用第三方工具和 AWS 開發套件來自動調整規模。

受管 AWS 服務包括 Amazon S3、Amazon CloudFront、AWS Auto Scaling、AWS Lambda、Amazon DynamoDB、AWS Fargate 和 Amazon Route 53。

AWS Auto Scaling 讓您可以偵測和取代受損的執行個體。這也讓您可以為資源建立擴展計畫，包括 Amazon EC2 執行個體和 Spot 機群叢集、 Amazon ECS 任務、 Amazon DynamoDB 資料表和索引，以及 Amazon Aurora 複本。

擴展 EC2 執行個體時，請確保您使用多個可用區域 (最好至少有三個) 並新增或移除容量，以便在這些可用區域之間維持平衡。ECS 任務或 Kubernetes Pod (使用 Amazon Elastic Kubernetes Service 時) 也應該分散到多個可用區域。

使用 AWS Lambda 時，執行個體會自動擴展。每次收到函數的事件通知時，AWS Lambda 會在其運算叢集內快速找到可用容量，然後執行您的程式碼，直到達到配置的並行為止。您需要確保已在特定 Lambda 和 Service Quotas 中設定必要的並行。

Amazon S3 會自動調整規模以處理高請求率。例如，您的應用程式可以在儲存貯體的每個字首達到每秒至少 3,500 個 PUT/COPY/POST/DELETE 或 5,500 個 GET/HEAD 請求。儲存貯體中的字首數量沒有限制。您可以透過平行化讀取來提升讀取或寫入效能。例如，如果您在 Amazon S3 儲存貯體中建立 10 個字首來平行讀取，則可以將讀取效能擴展為每秒 55,000 個讀取請求。

設定和使用 Amazon CloudFront 或受信任的內容交付網路 (CDN)。CDN 可以提供更快的最終使用者回應時間，而且可以為快取中的內容請求提供服務，因此可減少擴展工作負載的需求。

常用的反模式：

- 實作 Auto Scaling 群組以進行自動修復，但不實作彈性。
- 使用自動調整規模來回應大幅增加的流量。
- 部署高度狀態應用程式，免除彈性選項。

建立此最佳實務的優勢： 自動化會移除在部署和除役資源時可能出現的手動錯誤。自動化可免除因部署或除役需求回應緩慢而造成成本超支和拒絕服務的風險。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 設定和使用 AWS Auto Scaling。這會監控您的應用程式並自動調整容量，以盡可能低的成本維持穩定、可預測的效能。您可以使用 AWS Auto Scaling 為多個服務的多個資源設定應用程式擴展。
  - 什麼是 AWS Auto Scaling？

    - 在 Amazon EC2 執行個體和 Spot 機群、Amazon ECS 任務、Amazon DynamoDB 表格和索引、Amazon Aurora 複本和 AWS Marketplace 設備上設定 Auto Scaling (如適用)。
      - 使用 DynamoDB Auto Scaling 自動管理輸送量
        - 使用服務 API 操作來指定警示、擴展原則、準備時間和冷卻時間。
- 使用 Elastic Load Balancing。負載平衡器可以按路徑或網路連線來分配負載。
  - 什麼是 Elastic Load Balancing？
    - Application Load Balancers 可以按路徑分配負載。
      - 什麼是 Application Load Balancer？
        - 設定 Application Load Balancer，以根據網域名稱下的路徑將流量分配到不同的工作負載。
        - Application Load Balancers 可用於以與 AWS Auto Scaling 整合的方式分配負載，以管理需求。
          - 搭配 Auto Scaling 群組使用負載平衡器
    - Network Load Balancer 可以透過連線分配負載。
      - 什麼是 Network Load Balancer？
        - 設定 Network Load Balancer，以使用 TCP 將流量分配到不同的工作負載，或為您的工作負載分配固定的 IP 地址集。
        - Network Load Balancer 可用於以與 AWS Auto Scaling 整合的方式分配負載，以管理需求。
- 使用高度可用的 DNS 供應商。DNS 名稱讓您的使用者可以輸入名稱 (而不是 IP 地址) 來存取您的工作負載，並將此資訊分發到已定義的範圍 (通常是工作負載的所有使用者)。
  - 使用 Amazon Route 53 或信任的 DNS 供應商。
    - 什麼是 Amazon Route 53？
  - 使用 Route 53 來管理您的 CloudFront 分發和負載平衡器。
    - 確定要管理的網域和子網域。
    - 使用 ALIAS 或 CNAME 紀錄建立適當的紀錄集。
      - 處理記錄
- 使用 AWS 全球網路，優化從使用者到應用程式的路徑。AWS Global Accelerator 可持續監控應用程式端點的運作狀態，並在 30 秒內將流量重新導向到運作狀態良好的端點。
  - AWS Global Accelerator 是一種可改善具備當地或全球使用的應用程式可用性和效能的服務。它提供靜態 IP 地址，做為單一或多個 AWS 區域 (例如 Application Load Balancers、Network Load Balancers 或 Amazon EC2 執行個體) 應用程式端點的固定進入點。
    - 什麼是 AWS Global Accelerator？

- 設定和使用 Amazon CloudFront 或受信任的內容交付網路 (CDN)。內容交付網路可以提供更快的最終使用者回應時間，並且可以處理可能導致不必要的工作負載擴展的內容請求。
  - 什麼是 Amazon CloudFront？
    - 為您的工作負載設定 Amazon CloudFront 分發，或使用第三方 CDN。
      - 您可以限制對工作負載的存取，使其只能透過在端點安全群組或存取政策中使用 CloudFront 的 IP 範圍從 CloudFront 存取。

資源

相關文件：

- APN 合作夥伴：可以幫助您建立自動化運算解決方案的合作夥伴
- AWS Auto Scaling：擴展計畫的運作方式
- AWS Marketplace：可與 Auto Scaling 結合使用的產品
- 使用 DynamoDB Auto Scaling 自動管理輸送量
- 搭配 Auto Scaling 群組使用負載平衡器
- 什麼是 AWS Global Accelerator？
- 什麼是 Amazon EC2 Auto Scaling？
- 什麼是 AWS Auto Scaling？
- 什麼是 Amazon CloudFront？
- 什麼是 Amazon Route 53？
- 什麼是 Elastic Load Balancing？
- 什麼是 Network Load Balancer？
- 什麼是 Application Load Balancer？
- 處理記錄

REL07-BP02 在偵測到工作負載受損時取得資源

在可用性受到影響時視需要主動擴展資源，以還原工作負載可用性。

您必須先設定運作狀態檢查和這些檢查的條件，以指出可用性因資源不足而受到影響的時間。然後，通知適當的人員手動擴展資源，或觸發自動化以自動調整資源規模。

您可以針對工作負載手動調整規模，例如，透過AWS Management Console或 AWS CLI 變更 Auto Scaling 群組中的 EC2 執行個體數量，或修改 DynamoDB 資料表的輸送量。但是，應該盡可能使用自動化 (請參閱 取得或擴展資源時使用自動化) 建立持續整合/持續部署 (CI/CD) 管道。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 在偵測到工作負載受損時取得資源。在可用性受到影響時視需要主動擴展資源，以還原工作負載可用性。
  - 使用擴展計劃 (這是 AWS Auto Scaling 的核心元件) 來設定一組擴展資源的指示。如果您使用 AWS CloudFormation 或將標籤新增至 AWS 資源，則可以針對每個應用程式的不同資源集設定擴展計畫。AWS Auto Scaling 為針對每個資源自訂擴展的策略提供建議。建立擴展計畫之後，AWS Auto Scaling 會將動態擴展和預測擴展方法結合在一起，以支援您的擴展策略。
    - AWS Auto Scaling：擴展計畫的運作方式
  - Amazon EC2 Auto Scaling 可協助您確保您擁有正確的 Amazon EC2 執行個體數量來處理應用程式的負載。您可以建立稱為 Auto Scaling 群組的 EC2 執行個體集合。您可以在每個 Auto Scaling 群組中指定執行個體的最小數量，而 Amazon EC2 Auto Scaling 可確保您的群組大小永遠不會低於此值。您可以在每個 Auto Scaling 群組中指定執行個體的最大數量，而 Amazon EC2 Auto Scaling 可確保您的群組大小永遠不會高於此大小。
    - 什麼是 Amazon EC2 Auto Scaling？
  - Amazon DynamoDB Auto Scaling 使用 AWS Application Auto Scaling 服務代替您動態調整佈建的輸送容量，以回應實際的流量模式。這可讓資料表或全域次要索引增加其佈建的讀取與寫入容量，以在不需調節的情況下處理突然增加的流量。
    - 使用 DynamoDB Auto Scaling 自動管理輸送量

資源

相關文件：

- APN 合作夥伴：可以幫助您建立自動化運算解決方案的合作夥伴
- AWS Auto Scaling：擴展計畫的運作方式
- AWS Marketplace：可與 Auto Scaling 結合使用的產品
- 使用 DynamoDB Auto Scaling 自動管理輸送量
- 什麼是 Amazon EC2 Auto Scaling？

REL07-BP03 偵測到工作負載需要更多資源時取得資源

主動擴展資源以滿足需求並避免可用性影響。

許多 AWS 服務會自動調整規模以滿足需求。如果使用 Amazon EC2 執行個體或 Amazon ECS 叢集，您可以將這些叢集的自動調整規模功能設定為根據與工作負載需求對應之用量指標來執行。對於 Amazon EC2，平均 CPU 使用率、負載平衡器請求計數或網路頻寬可用於擴展 (或縮減) EC2 執行個體。對於 Amazon ECS，平均 CPU 使用率、負載平衡器請求計數和記憶體使用率可用於橫向擴展 (或縮減) ECS 任務。透過在 AWS 上使用 Target Auto Scaling，自動調整規模裝置的作用就像家用恆溫器一樣，可新增或移除資源以維持您指定的目標值 (例如，70% 的 CPU 使用率)。

AWS Auto Scaling 也可以執行 Predictive Auto Scaling，其會使用機器學習分析每個資源的歷史工作負載，並定期預測未來兩天的未來負載。

「利特爾法則」有助於計算您需要的運算執行個體 (EC2 執行個體、並行 Lambda 函數等) 的數量。

$L = \lambda W$

L = 執行個體數量 (或系統中的平均並行)

λ = 請求到達時的平均速率 (請求/秒)

W = 每個請求在系統中花費的平均時間 (秒)

例如，在 100 rps 時，如果每個請求需要 0.5 秒才能處理，您就需要 50 個執行個體才能因應需求。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 偵測到工作負載需要更多資源時取得資源。主動擴展資源以滿足需求並避免可用性影響。
  - 計算處理指定請求率所需的運算資源 (運算並行)。
    - 說說「利特爾法則」的故事
  - 當您有使用的歷史模式時，請設定 Amazon EC2 Auto Scaling 的排程擴展。
    - Amazon EC2 Auto Scaling 的排程擴展
  - 使用 AWS 預測擴展。
    - EC2 的預測擴展，採用機器學習技術

資源

相關文件：

- AWS Auto Scaling：擴展計畫的運作方式
- AWS Marketplace：可與 Auto Scaling 結合使用的產品
- 使用 DynamoDB Auto Scaling 自動管理輸送量
- EC2 的預測擴展，採用機器學習技術
- Amazon EC2 Auto Scaling 的排程擴展
- 說說「利特爾法則」的故事
- 什麼是 Amazon EC2 Auto Scaling？

REL07-BP04 對工作負載執行負載測試

採用負載測試方法來衡量擴展活動是否滿足工作負載要求。

重要的是執行持續的負載測試。負載測試應探索中斷點並和測試工作負載的效能。AWS 讓您可以輕鬆設定臨時測試環境，以塑造生產工作負載的規模。在雲端，您可隨需建立生產規模的測試環境、完成測試，再將資源除役。因為您只為執行中的測試環境付費，所以能以與內部部署測試相較之下相當微小比例的成本來模擬即時環境。

在生產系統承受壓力的演練日，以及客戶使用量較低的時間內，您也應考慮在生產中進行負載測試，並且讓可用的所有人員解釋結果並解決所發生的任何問題。

常用的反模式：

- 在與生產組態不同的部署上執行負載測試。
- 只對工作負載的個別部分而非整個工作負載執行負載測試。
- 使用請求的子集而非代表的實際請求集合來執行負載測試。
- 對高於預期負載的小型安全係數執行負載測試。

建立此最佳實務的優勢： 您會知道架構中的哪些元件在負載時失敗，並能夠識別要監看哪些指標，指出您正在及時處理該負載來解決問題，避免受到該故障的影響。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 執行負載測試，以識別工作負載的哪些層面指出您必須新增或移除容量。負載測試的代表性流量應該與您在生產環境中收到的流量相似。在觀看您已檢測的指標時增加負載，以判斷哪些指標指出何時必須新增或移除資源。

- AWS 上的分散式負載測試：模擬數千名連線的使用者
  - 識別請求混合。您可能會有不同的請求混合，因此您應該在識別流量混合時查看各種時間範圍。
  - 實作載入驅動程式。您可以使用自訂程式碼、開放原始碼或商業軟體實作載入驅動程式。
  - 最初使用小容量的負載測試。您將負載驅動到較小容量 (可能和單一執行個體或容器一樣小)，看到一些立即的影響。
  - 針對較大容量的負載測試。在分散式負載上的效果會有所不同，因此您必須盡可能在接近產品環境的條件下進行測試。

資源

相關文件：

- AWS 上的分散式負載測試：模擬數千名連線的使用者

REL 8 您如何實作變更？

需有控制變更以部署新功能，並確保工作負載和運作環境執行已知軟體，且能以可預測的方式修補或取代。如果這些變更不受控制，則難以預測這些變更的效果，或是解決肇因於這些變更的問題。

最佳實務

- REL08-BP01 將執行手冊用於部署等標準活動
- REL08-BP02 將功能測試整合為部署的一部分
- REL08-BP03 將彈性測試整合為部署的一部分
- REL08-BP04 使用不可變基礎設施進行部署
- REL08-BP05 使用自動化部署變更

REL08-BP01 將執行手冊用於部署等標準活動

執行手冊是實現特定成果的預定義程序。使用執行手冊執行手動或自動進行的標準活動。範例包括部署工作負載、修補工作負載或進行 DNS 修改。

例如，實施程序 以確保部署期間的回復安全性。確保您可以回復部署，且不會對客戶造成任何中斷，這對於打造可靠的服務而言至為關鍵。

對於執行手冊程序，從有效的手動流程開始，以程式碼實作並在適當時將其觸發為自動執行。

即使是高度自動化的複雜工作負載， 執行手冊仍然適用於執行演練日 或滿足嚴格的報告和稽核要求。

請注意，程序手冊用於回應特定事件，而執行手冊用於實現特定成果。執行手冊通常用於例行活動，而程序手冊則用於回應非例行事件。

常用的反模式：

- 在生產環境中對組態執行非計畫中的變更。
- 為了更快速地部署而略過計畫中的步驟，會導致部署失敗。
- 在不測試變更反轉的情況下進行變更。

建立此最佳實務的優勢： 有效的變更規劃可提高您成功執行變更的能力，因為您知道所有受影響的系統。在測試環境中驗證變更可提高您的可信度。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 透過在執行手冊中記錄程序，對熟知的事件做出一致且迅速的回應。
  - AWS Well-Architected Framework：概念：執行手冊
- 使用基礎設施即程式碼的原則來定義您的基礎設施。透過使用 AWS CloudFormation (或受信任的第三方) 來定義您的基礎設施，您可以使用版本控制軟體對變更進行版本控制和追蹤。
  - 使用 AWS CloudFormation (或受信任的第三方供應商) 來定義您的基礎設施。
    - 什麼是 AWS CloudFormation？
  - 使用良好的軟體設計原則，建立單一、解偶的範本。
    - 確定實作的許可、範本和負責方。
      - 使用 AWS Identity and Access Management 控制存取
    - 使用原始檔控制 (例如 AWS CodeCommit 或受信任的第三方工具) 進行版本控制。
      - 什麼是 AWS CodeCommit？

資源

相關文件：

- APN 合作夥伴：可以幫助您建立自動化部署解決方案的合作夥伴
- AWS Marketplace：可用於自動化部署的產品
- AWS Well-Architected Framework：概念：執行手冊
- 什麼是 AWS CloudFormation？

- 什麼是 AWS CodeCommit？

相關範例：

- 使用程序手冊和執行手冊將操作自動化

REL08-BP02 將功能測試整合為部署的一部分

功能測試會作為自動化部署的一部分執行。如果未符合成功條件，則會終止或回復管道。

這些測試會在生產前環境中執行，而且會在生產前暫存於管道中。理想情況下，這是做為部署管道的一部分來完成。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 將功能測試整合為部署的一部分。功能測試會作為自動化部署的一部分執行。如果未符合成功條件，則會終止或回復管道。
  - 在 AWS CodePipeline 中建立模型之軟體發行管道的「測試動作」期間叫用 AWS CodeBuild。此功能可讓您輕鬆針對程式碼執行各種測試，例如單元測試、靜態程式碼分析和整合測試。
    - AWS CodePipeline 新增對於單位的支援，以及透過 AWS CodeBuild 自訂整合測試
  - 使用 AWS Marketplace 解決方案，在軟體交付管道中執行自動化測試。
    - 軟體和測試自動化

資源

相關文件：

- AWS CodePipeline 新增對於單位的支援，以及透過 AWS CodeBuild 自訂整合測試
- 軟體和測試自動化
- 什麼是 AWS CodePipeline？

REL08-BP03 將彈性測試整合為部署的一部分

彈性測試 (使用 混沌工程的原則) 會在生產前環境中做為自動化部署管道的一部分執行。

這些測試會在生產前環境暫存於管道中並在其中執行。這些測試也應該在生產環境中執行，以 在生產環境中。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 將彈性測試整合為部署的一部分。使用混沌工程，這是對工作負載進行實驗的專業領域，以建立可承受生產環境中的動盪條件能力的可信度。
  - 彈性測試會注入故障或資源降級，以評估工作負載是否回應其設計的彈性。
    - Well-Architected 實驗室：第 300 級：測試 EC2 RDS 和 S3 的彈性
  - 這些測試可以定期在自動化部署管道的生產前環境中執行。
  - 這些測試也應該在生產環境中執行，但是以演練日的一部分執行。
  - 使用混沌工程原則，提出各種損害下工作負載表現方式的假設，然後使用彈性測試來測試您的假設。
    - 混沌工程的原則

資源

相關文件：

- 混沌工程的原則
- 什麼是 AWS Fault Injection Simulator？

相關範例：

- Well-Architected 實驗室：第 300 級：測試 EC2 RDS 和 S3 的彈性

REL08-BP04 使用不可變基礎設施進行部署

不可變基礎設施會強制規定生產工作負載上不得就地進行更新、安全性修補程式或組態方面的變更。需要進行變更時，會在新的基礎設施上建置架構並部署到生產環境。

不可變基礎設施範例最常見的實作是 不可變的伺服器。這表示如果伺服器需要更新或修正，則會部署新的伺服器，而非更新已在使用中的伺服器。因此，應用程式中的每個變更都會從軟體推送至程式碼儲存庫 (例如 git push) 開始，而不會透過 SSH 登入伺服器並更新軟體版本。不可變的基礎設施不允許進行變更，因此您可以確定已部署系統的狀態。不可變的基礎設施在本質上更一致、更可靠且更可預測，而且它們可簡化軟體開發和操作的許多方面。

在不可變的基礎設施中部署應用程式時，請使用 Canary 或藍/綠部署。

Canary 部署 是將少量客戶導向至新版本的實務，通常會在單一服務執行個體 (Canary) 上執行。之後，您可以仔細檢查所產生的任何行為變更或錯誤。如果遇到嚴重問題，可以從 Canary 中刪除流量，然後將使用者傳送回以前的版本。如果部署成功，則您可以繼續以期望的速度進行部署，同時監控變更是否有錯誤，直到完全部署為止。AWS CodeDeploy 可以使用將支援 Canary 部署的部署組態來設定 AWS CodeDeploy。

藍/綠部署 與 Canary 部署類似，不同之處在於整個應用程式須並行部署。您可在兩個堆疊 (藍色和綠色) 之間交替部署。再次強調，您可以將流量傳送到新版本，且如果發現部署問題，則可以回復到舊版本。通常會一次切換所有流量，但您也可以將一小部分的流量用於每個版本，以使用 Amazon Route 53 的加權 DNS 路由功能，提高新版本的採用率。可以使用將支援藍/綠部署的部署組態來設定 AWS CodeDeploy 及 AWS Elastic Beanstalk。

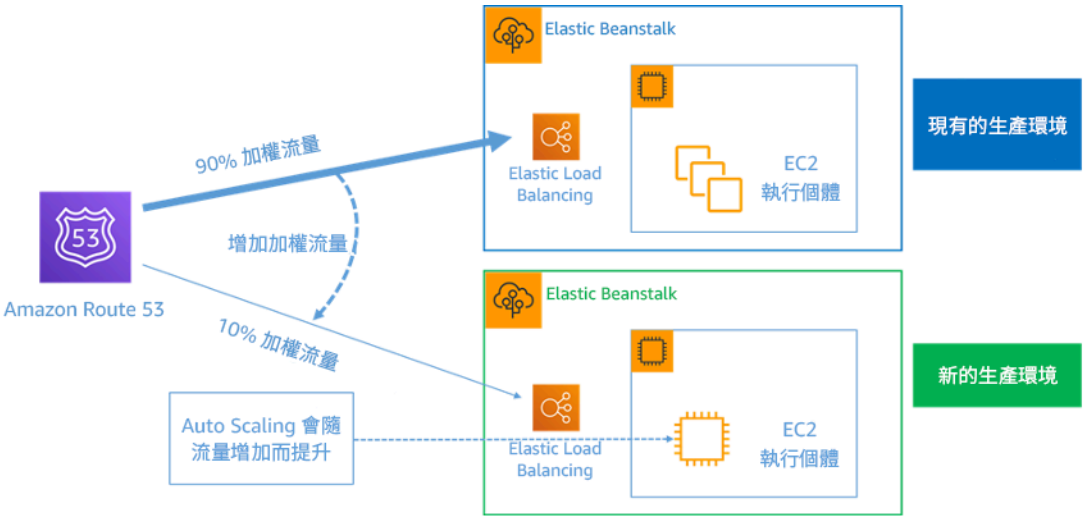


圖 8：使用 AWS Elastic Beanstalk 和 Amazon Route 53 進行藍/綠部署

不可變基礎設施的優勢：

- 降低組態偏移： 透過經常從基本、已知和版本控制的組態取代伺服器，可將基礎設施 重設為已知狀態，並避免組態偏移。 重設為已知狀態，並避免組態偏移。
- 簡化部署：部署不需要支援升級，因此會得到簡化。升級只是新的部署。
- 不可部分完成的可靠部署： 部署成功完成，或是未進行任何變更。其可為部署程序賦予更多信任。
- 利用快速的回復及復原程序打造更安全的部署： 前一個運作版本並未變更，因此部署變得更加安全。如果偵測到錯誤，您可以回復至該版本。
- 一致的測試和偵錯環境： 所有伺服器都會使用相同的映像，因此環境之間沒有差異。一個組建會部署到多個環境中。它還能預防不一致的環境並簡化測試與偵錯。
- 提高可擴展性： 伺服器使用基礎映像，具有一致性和可重複性，因此自動調整規模相當簡單。

- 簡化工具鏈：您可以擺脫管理生產軟體升級的組態管理工具，因此工具鏈得到簡化。不會在伺服器上安裝額外的工具或代理程式。會對基礎映像進行變更、並對變更進行測試然後推出。
- 提高安全性： 藉由拒絕對伺服器進行的所有變更，您可以停用執行個體上的 SSH 並移除金鑰。這可減少攻擊向量，從而改善組織的安全狀態。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用不可變基礎設施進行部署。不可變基礎設施是一種模型，其中不會進行更新、安全性修補程式或組態方面的變更。 就地 進行更新、安全性修補程式或組態方面的變更。需要進行變更時，系統會建置新版本的架構並部署到生產環境。
  - 藍/綠部署概觀
  - 逐步部署無伺服器應用程式
  - 不可變基礎設施：透過不可變實現可靠性、一致性和可信度
  - CanaryRelease

資源

相關文件：

- CanaryRelease
- 逐步部署無伺服器應用程式
- 不可變基礎設施：透過不可變實現可靠性、一致性和可信度
- 藍/綠部署概觀
- Amazon Builders' Library：確保部署期間的回復安全

REL08-BP05 使用自動化部署變更

部署和修補經過自動化以消除負面影響。

改變生產系統是許多組織的最大風險領域之一。我們認為，相較於軟體要解決的業務問題，部署才是我們要解決的首要問題。如今，這表示在營運中實際可行的地方使用自動化，包括測試和部署變更，新增或刪除容量以及移轉資料。AWS CodePipeline 讓您可以管理釋出工作負載所需的步驟。這包含使用 AWS CodeDeploy 的部署狀態，以自動將應用程式程式碼部署到 Amazon EC2 執行個體、內部部署執行個體、無伺服器 Lambda 函數或 Amazon ECS 服務。

建議

儘管傳統觀點建議您將業內人員安排在營運程序中最困難的部分，但是出於這個原因，我們建議您能自動化最困難的程序。

常用的反模式：

- 手動執行變更。
- 在緊急工作流程中略過自動化步驟。
- 不遵循您的計畫。

建立此最佳實務的優勢： 使用自動化部署所有變更可免除引進人為錯誤的可能性，並可在變更生產前進行測試，以確保您的計畫順利完成。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 自動化您的部署管道。部署管道讓您可以叫用自動測試、偵測異常，或者在生產部署之前的某個步驟中停止管道，或者自動回復變更。
  - Amazon Builders' Library：確保部署期間的回復安全
  - Amazon Builders' Library：使用持續交付加快腳步
    - 使用 AWS CodePipeline (或受信任的第三方產品) 來定義和執行管道。
      - 設定管道以在對程式碼儲存器提交變更時啟動。
        - 什麼是 AWS CodePipeline？
      - 使用 Amazon Simple Notification Service (Amazon SNS) 和 Amazon Simple Email Service (Amazon SES) 傳送有關管道中問題的通知，或與團隊聊天工具 (如 Amazon Chime) 整合。
        - 什麼是 Amazon Simple Notification Service？
        - 什麼是 Amazon SES？
        - 什麼是 Amazon Chime？
        - 使用 Webhook 自動化聊天訊息。

資源

相關文件：

- APN 合作夥伴：可以幫助您建立自動化部署解決方案的合作夥伴
- AWS Marketplace：可用於自動化部署的產品
- 使用 Webhook 自動化聊天訊息。
- Amazon Builders' Library：確保部署期間的回復安全
- Amazon Builders' Library：使用持續交付加快腳步
- 什麼是 AWS CodePipeline？
- 什麼是 CodeDeploy？
- AWS Systems Manager Patch Manager
- 什麼是 Amazon SES？
- 什麼是 Amazon Simple Notification Service？

相關影片：

- 2019 年 AWS 高峰會：AWS 上的 CI/CD

失敗管理

問題

- REL 9 您如何備份資料？
- REL 10 如何使用故障隔離來保護您的工作負載？
- REL 11 如何設計工作負載以承受元件失敗？
- REL 12 如何測試可靠性？
- REL 13 您如何規劃災難復原 (DR)？

REL 9 您如何備份資料？

備份資料、應用程式和組態，以符合復原時間目標 (RTO) 和復原點目標 (RPO) 的要求。

最佳實務

- REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料
- REL09-BP02 保護和加密備份
- REL09-BP03 自動執行資料備份

- REL09-BP04 定期執行資料復原以驗證備份的完整性和程序

REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料

所有 AWS 資料存放區都會提供備份功能。Amazon RDS 和 Amazon DynamoDB 等服務會額外地支援啟用時間點復原 (PITR) 的自動備份，這可讓您將備份還原到目前時間之前最多五分鐘或更短的任何時間。許多 AWS 服務提供將備份複製到另一個 AWS 區域 的能力。AWS Backup 是一種工具，可讓您跨 AWS 服務集中化和自動化資料保護。

Amazon S3 可以用作自行受管和 AWS 受管資料來源的備份目的地。Amazon EBS、Amazon RDS 和 Amazon DynamoDB 等 AWS 服務具有內建功能來建立備份。也可以使用第三方備份軟體。

內部部署資料可以備份至 AWS 雲端，方法為使用 AWS Storage Gateway 或 AWS Datasync。Amazon S3 儲存貯體可以用來將此資料儲存在 AWS 上。Amazon S3 提供多個儲存層，例如 Amazon S3 Glacier 或 S3 Glacier Deep Archive 來減少資料儲存的成本。

您能夠從其他資源重現資料來符合資料復原需求。例如， Amazon Elasticache 複本節點 或 RDS 讀取複本 可以用來重現資料，如果主要節點遺失的話。如果像這樣的來源可以用來符合您的 復原時間目標 (RTO) 和復原點目標 (RPO)，您可能不需要備份。另一個範例，如果使用 Amazon EMR，可能不需要備份 HDFS 資料存放區，只要您可以 從 S3 將資料重現至 EMR。 。

選取備份策略時，請考慮復原資料所需的時間。復原資料所需的時間取決於備份的類型 (若有備份策略)，或資料重現機制的複雜性。此時間應該落在工作負載的 RTO 內。

預期成果：

已根據關鍵性識別和分類資料來源。然後，根據 RPO 建立資料復原的策略。此策略涉及備份這些資料來源，或具有從其他來源重現資料的能力。若遺失資料，實作的策略可讓您在定義的 RPO 和 RTO 內復原或重現資料。

雲端成熟度階段： 基礎級

常用的反模式：

- 未注意工作負載的所有資料來源及其關鍵性。
- 未備份關鍵資料來源。
- 只備份某些資料來源，而未使用關鍵性做為準則。
- 沒有已定義的 RPO，或備份頻率無法符合 RPO。
- 未評估是否需要備份，或是否可從其他來源重現資料。

建立此最佳實務的優勢： 識別需要備份的位置並實作機制來建立備份，或者能夠從外部源重現資料，可以改善在中斷期間還原和復原資料的能力。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解和使用工作負載所使用的 AWS 服務和資源的備份功能。大部分 AWS 服務都會提供備份工作負載資料的功能。

實作步驟：

1. 識別工作負載的資料來源。資料可以儲存在多個來源上，例如 資料庫， 磁碟區， 檔案系統， 記錄系統和 物件儲存。如需有關 Web 應用程式後端的建議，請參閱 資源 一節來尋找 相關文件 ，其中關於儲存資料的不同 AWS 服務，以及這些服務提供的備份功能。
2. 根據關鍵性將資料來源分類。不同的資料集對工作負載具有不同的關鍵性等級，因此對彈性具有不同的要求。例如，有些資料可能至關重要，且需要接近零的 RPO，而其他資料可能不太重要，且可以容忍更高的 RPO 和一些資料遺失。同樣地，不同的資料集也可能具有不同的 RTO 要求。
3. 使用 AWS 或第三方服務來建立資料的備份。 AWS Backup 是受管服務，可讓您在 AWS 上建立各種資料來源的備份。其中大部分服務也具有建立備份的原生功能。AWS Marketplace 具有許多也提供這些功能的解決方案。如需有關 Web 應用程式後端的建議，請參閱 資源 ，以取得如何從各種 AWS 服務建立資料備份的相關資訊。
4. 對於未備份的資料，請建立資料重現機制。您可能基於各種原因選擇不備份可從其他來源重現的資料。可能有一種情況，即在需要時從來源重現資料比建立備份更便宜，因為可能有與儲存備份相關聯的成本。另一個範例是從備份中還原比從來源重現資料需要更長的時間，因而導致 RTO 中出現缺口。在這類情況下，考慮取捨並建立一個妥善定義的流程，其中指出在需要資料復原時如何從這些來源重現資料。例如，如果您已將資料從 Amazon S3 載入至資料倉儲 (如 Amazon Redshift) 或 MapReduce 叢集 (如 Amazon EMR)，對該資料執行分析，則這可能是可從其他來源重現的資料範例。只要這些分析的結果存放在某處或可複製，您就不會因為資料倉儲或 MapReduce 叢集故障而遺失資料。其他可從來源複製的範例包括快取 (如 Amazon ElastiCache) 或 RDS 的僅供讀取複本。
5. 建立備份資料的規律。。建立資料來源的備份是一種定期流程，而且頻率應取決於 RPO。

實作計劃的工作量： 中

資源

相關的最佳實務：

REL13-BP01 定義停機和資料遺失的復原目標

REL13-BP02 使用定義的復原策略來滿足復原目標

相關文件：

- 什麼是 AWS Backup？
- 什麼是 AWS DataSync？
- 什麼是磁碟區閘道？
- APN 合作夥伴：可以幫助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- Amazon EBS 快照
- 備份 Amazon EFS
- 備份 Amazon FSx for Windows File Server
- ElastiCache for Redis 備份與還原
- 在 Neptune 中建立資料庫叢集快照
- 建立資料庫快照
- 建立依照排程觸發的 EventBridge 規則
- 跨區域複寫 搭配 Amazon S3
- EFS-to-EFS AWS Backup
- 將日誌資料匯出至 Amazon S3
- 物件生命週期管理
- DynamoDB 的隨需備份和還原
- DynamoDB 的時間點復原
- 使用 Amazon OpenSearch Service 索引快照

相關影片：

- AWS re:Invent 2021 - 使用 AWS 進行備份、災難復原和勒索軟體防護
- AWS Backup 示範：跨帳戶和跨區域備份
- AWS re:Invent 2019：深入探討 AWS Backup，ft.Rackspace (STG341)

相關範例：

- Well-Architected 實驗室：實作 Amazon S3 雙向跨區域複寫 (CRR)

- Well-Architected 實驗室：測試備份並還原資料
- Well-Architected 實驗室：透過適用於分析工作負載的容錯回復進行備份和還原
- Well-Architected 實驗室：災難復原 - 備份和還原

REL09-BP02 保護和加密備份

使用身分驗證和授權 (例如AWS IAM) 控制並偵測對備份的存取。使用加密來防止並檢測是否危及備份的資料完整性。

Amazon S3 支援多種靜態資料的加密方法。使用伺服器端加密時，Amazon S3 會以未加密資料的形式接受物件，然後在儲存這些物件之前將其加密。使用用戶端加密時，您的工作負載應用程式需負責加密資料，然後將資料傳送至 Amazon S3。這兩種方法都可讓您使用 AWS Key Management Service (AWS KMS) 來建立和存放資料金鑰，或者您也可以提供自己的金鑰，之後由您對其負責。使用 AWS KMS 時，您可以透過 IAM 設定政策，設定誰可以和誰無法存取您的資料金鑰和解密資料。

對於 Amazon RDS，如果您已選擇加密資料庫，則備份也會加密。DynamoDB 備份一律加密。

常用的反模式：

- 讓備份和還原自動化的存取權與資料的存取權相同。
- 不加密您的備份。

建立此最佳實務的優勢： 保護您的備份可防止資料遭到竄改，加密資料可防止意外暴露時存取該資料。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 在每個資料存放區使用加密。如果來源資料已加密，則備份也會加密。
  - 在 RDS 中啟用加密。您可以在建立 RDS 執行個體時，使用 AWS Key Management Service 設定靜態加密。
    - 加密 Amazon RDS 資源
  - 在 EBS 磁碟區上啟用加密。您可以在建立磁碟區時設定預設加密或指定唯一金鑰。
    - Amazon EBS 加密
  - 使用必要的 Amazon DynamoDB 加密。DynamoDB 會加密所有靜態資料。您可以使用 AWS 自有的 AWS KMS 金鑰或 AWS 受管 KMS 金鑰，指定帳戶中儲存的金鑰。

    - DynamoDB 靜態加密
    - 管理加密表格
  - 加密存放在 Amazon EFS 中的資料。在建立檔案系統時設定加密。
    - 在 EFS 中加密資料和中繼資料
  - 在來源和目的地區域設定加密。您可以使用 KMS 中存放的金鑰來設定 Amazon S3 中的靜態加密，但金鑰受到區域限定。您可以在設定複寫時指定目的地金鑰。
    - CRR 其餘組態：複寫使用 AWS KMS 中存放的加密金鑰，透過伺服器端加密 (SSE) 所建立的物件。
- 實作存取備份的最低許可。遵循最佳實務，以根據安全最佳實務限制對備份、快照和複本的存取。
  - 安全支柱：AWS Well-Architected

資源

相關文件：

- AWS Marketplace：可用於備份的產品
- Amazon EBS 加密
- Amazon S3：利用加密保護資料
- CRR 其餘組態：複寫使用 AWS KMS 中存放的加密金鑰，透過伺服器端加密 (SSE) 所建立的物件。
- DynamoDB 靜態加密
- 加密 Amazon RDS 資源
- 在 EFS 中加密資料和中繼資料
- AWS 中的備份加密
- 管理加密表格
- 安全支柱：AWS Well-Architected

相關範例：

- Well-Architected 實驗室：實作 Amazon S3 雙向跨區域複寫 (CRR)

REL09-BP03 自動執行資料備份

設定備份以根據復原點目標 (RPO) 所通知的定期排程或資料集中的變更自動執行。資料遺失要求低的關鍵資料集需要經常自動備份，而可以接受一些遺失的不太重要資料可以較不頻繁地備份。

AWS Backup 可以用來建立各種 AWS 資料來源的自動資料備份。Amazon RDS 執行個體幾乎可以持續每五分鐘備份一次，而且 Amazon S3 物件幾乎可以持續每十五分鐘備份一次，同時將時間點復原 (PITR) 提供至備份歷史記錄內的特定時間點。針對其他 AWS 資料來源，例如 Amazon EBS 磁碟區、Amazon DynamoDB 資料表或 Amazon FSx 檔案系統，AWS Backup 可以頻繁地每小時執行自動備份。這些服務也會提供原生備份功能。提供自動備份與時間點復原的 AWS 服務包括 Amazon DynamoDB、 Amazon RDS和 Amazon Keyspaces (適用於 Apache Cassandra) – 這些可以還原至備份歷史記錄內的特定時間點。大部分其他 AWS 資料儲存服務都會提供定期備份排程的能力，頻率為每小時備份一次。

Amazon RDS 和 Amazon DynamoDB 會提供連續備份與時間點復原。一旦啟用了 Amazon S3 版本控制，就會自動執行。Amazon Data Lifecycle Manager 可以用來自動建立、複製和刪除 Amazon EBS 快照。其也可以自動建立、複製、棄用和取消註冊 Amazon EBS 支援的 Amazon Machine Image (AMI) 及其基礎 Amazon EBS 快照。

為了集中檢視備份自動化和歷史記錄，AWS Backup 提供全受管的、基於政策的備份解決方案。它使用 AWS Storage Gateway 在雲端和內部部署中跨多個 AWS 服務，自動集中進行資料備份。

除版本控制之外，Amazon S3 還具有複寫功能。整個 S3 儲存貯體可自動複寫至相同或不同 AWS 區域中的另一個儲存貯體。

預期成果：

以建立的規律建立資料來源備份的自動化流程。

常用的反模式：

- 手動執行備份。
- 使用具有備份功能的資源，但不包含您的自動化中的備份。

建立此最佳實務的優勢： 自動化備份可確保它們根據您的 RPO 定期進行備份，如果未進行備份則會提醒您。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

1. 識別目前正在 手動備份的資料來源。請參閱 REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料 以取得此動作的指引。
2. 判斷工作負載的 PRO。請參閱 REL13-BP01 定義停機和資料遺失的復原目標 以取得此動作的指引。

3. 使用自動化備份解決方案或受管服務。AWS Backup 是全受管服務，可讓您 輕鬆地在雲端和內部部署跨 AWS 服務集中和自動保護資料。。備份計劃是 AWS Backup 的功能，可讓您建立規則，定義要備份的資源，以及應以何種頻率建立這些備份。此頻率應由步驟 2 中建立的 RPO 通知。 此 WA 實驗室 提供如何使用 AWS Backup 建立自動備份的實作指引。大多數存放資料的 AWS 服務都會提供原生備份功能。例如，可以利用 RDS 搭配時間點復原 (PITR) 進行自動備份。
4. 針對自動化備份解決方案或 受管服務不支援的資料來源 (例如內部部署資料來源或訊息佇列)，請考慮使用信任的第三方解決方案建立自動化備份。或者，您可以使用 AWS CLI 或 SDK 建立自動化來執行此動作。您可以使用 AWS Lambda Functions 或 AWS Step Functions，定義涉及建立資料備份的邏輯，以及使用 Amazon EventBridge，以基於步驟 2 中所建立 RPO 的頻率執行它。

實作計劃的工作量： 低

資源

相關文件：

- APN 合作夥伴：可以幫助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- 建立依照排程觸發的 EventBridge 規則
- 什麼是 AWS Backup？
- 什麼是 AWS Step Functions？

相關影片：

- AWS re:Invent 2019：深入探討 AWS Backup，ft.Rackspace (STG341)

相關範例：

- Well-Architected 實驗室：測試備份並還原資料

REL09-BP04 定期執行資料復原以驗證備份的完整性和程序

透過執行復原測試，驗證您的備份程序實作是否符合復原時間目標 (RTO) 和復原點目標 (RPO)。

使用 AWS 時，您可以建立一個測試環境，還原備份來評估 RTO 和 RPO 功能，並針對資料內容和完整性執行測試。

此外，Amazon RDS 和 Amazon DynamoDB 允許時間點復原 (PITR)。使用持續備份時，您可以將資料集還原到指定日期和時間當時的狀態。

預期成果： 使用妥善定義的機制定期復原來自備份的資料，以確保可在工作負載的既定復原時間點目標 (RTO) 內復原。驗證從備份中還原是否會導致資源包含原始資料 (而其中沒有任何損壞或無法存取)，但在復原點目標 (RPO) 內發生資料遺失。

常用的反模式：

- 還原備份，但不查詢或擷取任何資料，以確保還原可用。
- 假設備份存在。
- 假設系統的備份可以完全運作，而且可以從中復原資料。
- 假設從備份中還原或復原資料的時間落在工作負載的 RTO 內。
- 假設備份上包含的資料落在工作負載的 RPO 內。
- 在不使用執行手冊的情況下，或在建立的自動化程序外部，還原特定資料。

建立此最佳實務的優勢： 測試備份的復原確保可在需要時還原資料，而不必擔心資料可能丟失或損壞，也可確保還原和復原可在工作負載的 RTO 內進行，而且任何資料遺失都會落在工作負載的 RPO 內。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

測試備份和還原功能可以提高能夠在中斷期間執行這些動作的信心。定期將備份還原至新位置，並執行測試以驗證資料的完整性。某些應該執行的常用測試正在檢查

所有資料是否可用、未損壞、可存取，並且任何資料遺失都落在工作負載的 RPO 內。此類測試也可以協助確定，復原機制是否足夠快到適應工作負載的 RTO。

1. 識別目前正在 備份的資料來源，以及這些備份的存放位置。請參閱 REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料 以取得如何實作此動作的指引。
2. 建立每個資料來源的 資料驗證準則。不同類型的資料將具有不同的屬性，可能需要不同的驗證機制。在您自信可於生產環境中使用此資料之前，請考慮如何驗證它。一些驗證資料的常用方法是使用資料和備份屬性，例如資料類型、格式、檢查總和、大小，或這些屬性與自訂驗證邏輯的組合。例如，這可能是建立備份時所還原資源與資料來源之間的檢查總和值比較。
3. 建立 RTO 和 RPO， 根據資料關鍵性還原資料。請參閱 REL13-BP01 定義停機和資料遺失的復原目標 以取得如何實作此動作的指引。

4. 存取您的復原功能。審查您的備份和還原策略，以了解它是否可以符合您的 RTO 和 RPO，並視需要調整策略。您可以使用 AWS Resilience Hub，執行工作負載的評定。此評定會針對彈性政策評估您的應用程式組態，並報告您的 RTO 和 RPO 目標是否可以實現。

5. 使用目前建立且用於 生產環境進行資料還原的程序執行測試還原。這些程序取決於原始資料來源的備份方式、備份本身的格式和儲存位置，或是否已從其他源重現資料。例如，如果您是使用類似 AWS Backup 的受管服務，這可能與將備份還原至新資源一樣簡單。如果已使用 AWS 彈性災難復原，您可以 啟動復原練習。。

6. 根據您先前 在步驟 2 中針對資料驗證建立的準則，驗證從已還原的資源 (來自上一個步驟) 進行的資料復原。還原和復原的資料是否包含備份時最新的記錄/項目？ 此資料是否落在工作負載的 RPO 內？

7. 測量還原和復原 所需的時間，並將其與步驟 3 中建立的 RTO 進行比較。此程序是否落在工作負載的 RTO 內？ 例如，比較從還原程序開始到復原驗證完成的時間戳記，以計算此程序需要多長時間。所有 AWS API 都會加上時間戳記，而且此資訊可用於 AWS CloudTrail。儘管此資訊可以提供有關還原程序何時開始的詳細資訊，但驗證完成時的結束時間戳記應由驗證邏輯記錄。如果使用自動程序，則 Amazon DynamoDB 之類服務可以用來存放此資訊。此外，許多 AWS 服務會提供事件歷史記錄，其中提供特定動作何時發生的時間戳記資訊。在 AWS Backup 內，備份和還原動作都稱為 工作，而且這些工作包含時間戳記資訊做為其中繼資料的一部分，而此中繼資料可以用來測量還原和復原所需的時間。

8. 通知利害關係人 如果資料驗證失敗，或如果還原和復原所需的時間超出針對工作負載建立的 RTO。實作自動化來執行此動作 (例如在此實驗室中)時，Amazon Simple Notification Service (Amazon SNS) 之類服務可以用來將電子郵件或 SMS 等通知推送至利害關係人。 這些訊息也可以推送至傳訊應用程式，例如 Amazon Chime、Slack 或 Microsoft Teams， 或用來 使用 AWS Systems Manager OpsCenter 建立任務做為 OpsItems。

9. 將此程序自動化為定期執行。例如，服務 (例如 AWS Lambda 或 AWS Step Functions 中的狀態機器) 可以用來將還原和復原程序自動化，而且 Amazon EventBridge 可以用來定期觸發此自動化工作流程，如下面架構圖所示。了解如何 使用 AWS Backup 將資料復原驗證自動化。此外， 這個 Well-Architected 實驗室 會提供實作體驗，有關在這裡為數個步驟執行自動化的方式。

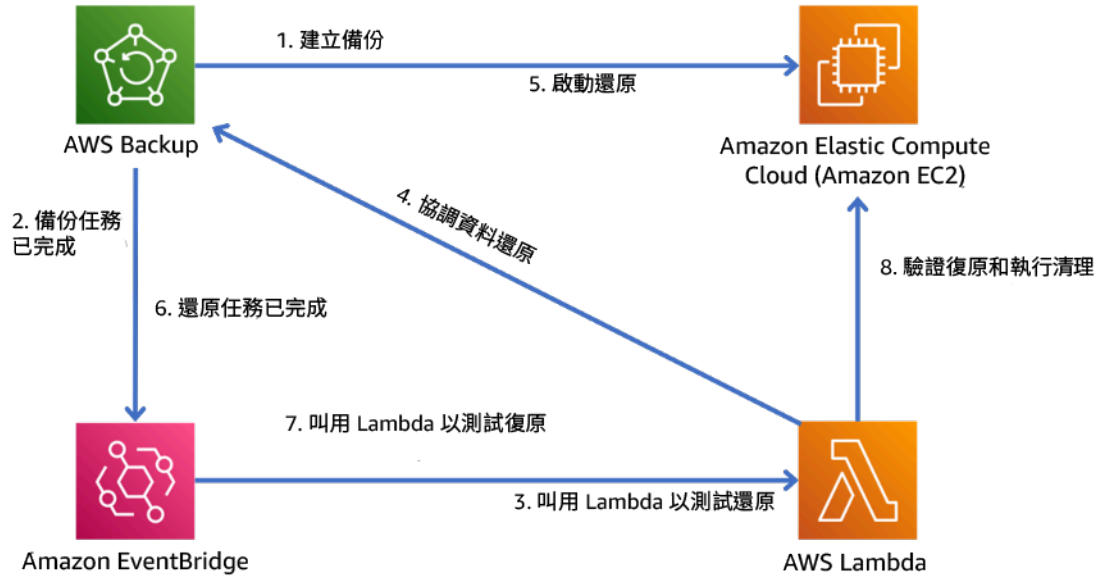


圖 9.自動的備份和還原程序

實作計劃的工作量： 中到高，取決於驗證準則的複雜性。

資源

相關文件：

- 使用 AWS Backup 將資料復原驗證自動化
- APN 合作夥伴：可以幫助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- 建立依照排程觸發的 EventBridge 規則
- DynamoDB 的隨需備份和還原
- 什麼是 AWS Backup？
- 什麼是 AWS Step Functions？
- 什麼是 AWS 彈性災難復原
- AWS 彈性災難復原

相關範例：

- Well-Architected 實驗室：測試備份並還原資料

REL 10 如何使用故障隔離來保護您的工作負載？

故障隔離界限會在工作負載內將失敗影響限制至有限數量的元件。界限外的元件不受失敗影響。使用多個故障隔離界限時，您可以限制對工作負載的影響。

最佳實務

- REL10-BP01 將工作負載部署至多個位置
- REL10-BP02 為您的多位置部署選取適當位置
- REL10-BP03 針對限制在單一位置的元件將復原自動化
- REL10-BP04 使用隔板架構限制影響範圍

REL10-BP01 將工作負載部署至多個位置

跨多個可用區域或視需要跨 AWS 區域，分配工作負載資料和資源。這些位置可以根據需要多樣化。

AWS 服務設計的基本原則之一是避免底層實體基礎設施中出現單點故障。這樣一來，我們將能建置可使用多個可用區域且能應對單一區故障的軟體和系統。同樣地，可將系統建置為能應對單一運算節點、單一儲存磁碟區或資料庫的單一執行個體的故障。建置依賴冗餘元件的系統時，務必要確保元件能獨立運行，而對於 AWS 區域而言，應能自主運行。具有冗餘元件的理論可用性計算，其優點只有在符合此條件時才有效。

可用區域 (AZ)

AWS 區域由多個可用區域組成，它們設計為彼此獨立作業。每個可用區域與其他可用區域是以有意義的實體距離隔開，從而可避免因火災、洪水和龍捲風等環境危害導致相關的失敗情境。每個可用區域也都具有獨立的實體基礎設施：可用區域內部和外部的公用電源專用連接、獨立的備用電源、獨立的機械服務以及獨立的網路連線。這種設計會將任何這些系統中的錯誤僅限制在受影響的可用區域。儘管在地理位置上是分開的，但可用區域位於啟用高輸送量、低延遲聯網的同一區域。整個 AWS 區域 (跨所有可用區域，由多個實體上獨立的資料中心組成) 可以視為工作負載的單一邏輯部署目標，包括同步複寫資料的能力 (例如，在資料庫之間)。這可讓您在主動/主動或主動/待命組態中使用可用區域。

可用區域是各自獨立的，因此當工作負載架構為使用多個區域時，工作負載的可用性也會隨之提高。一些 AWS 服務 (包括 Amazon EC2 執行個體資料平面) 會部署為嚴格的區域服務，其中它們與其所在的可用區域共享命運。不過，其他 AZ 中的 Amazon EC2 執行個體將不受影響並繼續運作。同樣地，如果可用區域中的失敗導致 Amazon Aurora 資料庫失敗，則未受影響 AZ 中的讀取副本 Aurora 執行個體可以自動提升為主要執行個體。另一方面，區域 AWS 服務 (例如 Amazon DynamoDB) 可內部使用主動/主動組態中的多個可用區域，以實現該服務的可用性設計目標，無需您設定 AZ 置放。

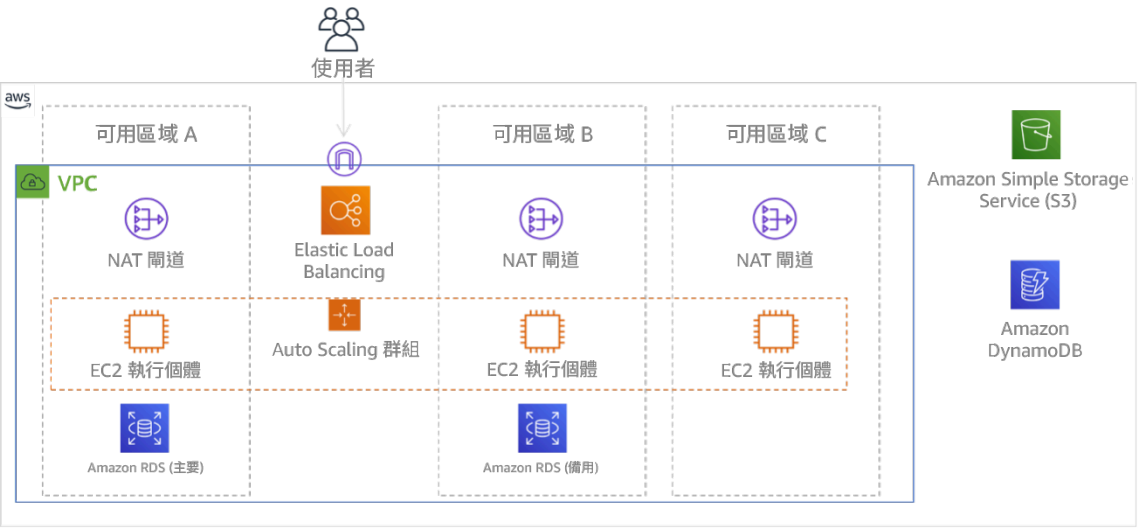


圖 9：跨三個可用區域部署的多層架構。請注意，Amazon S3 和 Amazon DynamoDB 一律自動採用異地同步備份策略。ELB 也會部署至全部三個區域。

儘管 AWS 控制平面通常有能力管理整個區域 (多個可用區域) 內的資源，但是某些控制平面 (包括 Amazon EC2 和 Amazon EBS) 能夠將結果篩選至單一可用區域。完成此操作後，僅在指定的可用區域中處理該請求，從而減少其他可用區域中的中斷風險。此 AWS CLI 範例說明僅從 us-east-2c 可用區域取得 Amazon EC2 執行個體資訊：

```
AWS ec2 describe-instances --filters Name=availability-zone,Values=us-east-2c
```

AWS Local Zones

AWS Local Zones 的作用與各自 AWS 區域內的可用區域類似，它們可在其中被選取為區域 AWS 資源 (如子網路和 EC2 執行個體) 的置放位置。特別之處在於它們不是位於相關聯的 AWS 區域，而是鄰近目前沒有 AWS 區域的大型人口、產業和 IT 中心。然而，它們仍可在本機區域的本機工作負載與在 AWS 區域中執行的本機工作負載之間保持高頻寬、安全的連線。您應該使用 AWS Local Zones，針對低延遲要求部署離使用者更近的工作負載。

Amazon Global Edge Network

Amazon Global Edge Network 由分布在全球各城市的節點組成。Amazon CloudFront 使用此網路以較低的延遲將內容交付給最終使用者。AWS Global Accelerator 讓您可以在這些節點建立工作負載端點，以便在靠近使用者的 AWS 全球網路提供引導服務。Amazon API Gateway 使用 CloudFront 分配啟用邊緣最佳化的 API 端點，以透過最接近的節點加快用戶端存取。

AWS 區域

AWS 區域都設計為自主的，因此，若要使用多區域方法，您要部署專用的服務副本至每個區域。

多區域方法常用於 災難復原 策略，以在一次性大規模事件發生時符合復原目標。請參閱 災難復原 (DR) 計畫 以取得這些策略的詳細資訊。然而在此，我們反而專注於 可用性，尋求隨時間交付平均運行時間目標。對於高可用性目標，多區域架構通常會設計為主動/主動，其中每個服務副本 (在其各自的區域中) 都是主動的 (服務請求)。

> **建議**
>
> 您可以在單一 AWS 區域 內使用異地同步備份策略，滿足大部分工作負載的可靠性目標。僅在工作負載具有極端的可用性要求或其他需要多區域架構的業務目標時，才考慮多區域架構。

AWS 可讓您跨區域操作服務。例如，AWS 使用 Amazon Simple Storage Service (Amazon S3) 複寫、Amazon RDS 讀取複本 (包括 Aurora 讀取複本) 和 Amazon DynamoDB 全域表提供資料的連續、非同步資料複寫。透過持續複寫，您的資料版本幾乎可以立即在您的每個作用中區域中使用。

使用 AWS CloudFormation，您可以定義基礎設施，並以一致方式跨 AWS 帳戶 和跨 AWS 區域 進行部署。為了擴充此功能，AWS CloudFormation StackSets 會讓您可以使用單一作業跨多個帳戶和區域建立、更新或刪除 AWS CloudFormation 堆疊。對於 Amazon EC2 部署執行個體，AMI (Amazon Machine Image) 用來提供資訊，例如硬體組態和安裝的軟體。您可以實作 Amazon EC2 Image Builder 管道，建立您需要的 AMI，並將這些 AMI 複製到作用中區域。這可確保這些 黃金 AMI 具備您在每個新區域中部署和橫向擴展工作負載所需的一切。

若要路由流量，Amazon Route 53 和 AWS Global Accelerator 會啟用政策的定義，而這些政策可決定哪些使用者前往哪個作用中區域端點。透過 Global Accelerator，您可以設定流量刻度盤，來控制導向到每個應用程式端點的流量百分比。Route 53 支援這種百分比方法，也支援多種其他可用政策，包括地理位置臨近性和延遲型政策。Global Accelerator 自動利用廣泛的 AWS 邊緣伺服器網路，盡快將流量上線至 AWS 網路主幹，這會導致降低請求延遲。

所有這些功能都會運作，以保留每個區域的自主權。這種方法幾乎不存在例外情況，包括我們可提供全域交付的服務 (例如 Amazon CloudFront 和 Amazon Route 53) 以及 AWS Identity and Access Management (IAM) 服務的控制平面。大部分服務完全在單一區域內運行。

內部部署資料中心

對於在內部部署資料中心執行的工作負載，請盡可能架構混合式體驗。AWS Direct Connect 提供從內部設施連接至 AWS 的專用網路連線，讓您可以在兩種環境中執行。

另一個選項是使用 AWS Outposts 在內部設施執行 AWS 基礎設施和服務。AWS Outposts 是一種全受管服務，可將 AWS 基礎設施、AWS 服務、API 和工具延伸到您的資料中心。AWS 雲端中使用的硬體基礎設施與資料中心安裝的硬體基礎設施相同。AWS Outposts 會接著連接至最近的 AWS 區域 區域。然後，您可以使用 AWS Outposts 來支援低延遲或有本機資料處理要求的工作負載。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 使用多個可用區域和 AWS 區域。跨多個可用區域或視需要跨 AWS 區域，分配工作負載資料和資源。這些位置可以根據需要多樣化。
  - 區域服務固有地跨可用區域部署。
    - 這包括 Amazon S3、Amazon DynamoDB 和 AWS Lambda (未連線至 VPC 時)
  - 將容器、執行個體和函數中的工作負載部署到多個可用區域中。使用多區域資料存放區，包括快取。使用 EC2 Auto Scaling 的功能、ECS 任務放置、AWS Lambda 函數組態 (在 VPC 中執行時) 和 ElastiCache 叢集。
    - 部署 Auto Scaling 群組時，使用單獨的可用區域中的子網路。
      - 範例：將執行個體分散到多個可用區域
      - Amazon ECS 任務置放策略
      - 設定 AWS Lambda 函數以存取 Amazon VPC 中的資源
      - 選擇區域和可用區域
    - 部署 Auto Scaling 群組時，使用單獨的可用區域中的子網路。
      - 範例：將執行個體分散到多個可用區域
    - 使用 ECS 任務置放參數，指定資料庫子網路群組。
      - Amazon ECS 任務置放策略
    - 將函數設定為在 VPC 中執行時，在多個可用區域中使用子網路。
      - 設定 AWS Lambda 函數以存取 Amazon VPC 中的資源
    - 將多個可用區域與 ElastiCache 叢集一起使用。
      - 選擇區域和可用區域
- 如果您的工作負載必須部署至多個區域，請選擇多區域策略。大多數的可靠性需求都可透過多個可用區域策略，在單一 AWS 區域內滿足。視需要使用多區域策略，以符合您的業務需求。
  - AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
    - 在另一個 AWS 區域的備份可以進一步確保資料在需要時可用。
    - 有些工作負載會有法規要求，規定要使用多區域策略。

- 針對您的工作負載評估 AWS Outposts。如果您的工作負載需要內部部署資料中心達到低延遲要求，或有本機資料處理要求。然後使用 AWS Outposts 在內部部署執行 AWS 基礎設施和服務
  - 什麼是 AWS Outposts？
- 判斷 AWS Local Zones 是否協助您為使用者提供服務。如果您有低延遲要求，請查看 AWS Local Zones 是否靠近您的使用者。如果是如此，則使用它來部署更靠近這些使用者的工作負載。
  - AWS Local Zones 常見問答集

資源

相關文件：

- AWS 全球基礎設施
- AWS Local Zones 常見問答集
- Amazon ECS 任務置放策略
- 選擇區域和可用區域
- 範例：將執行個體分散到多個可用區域
- 全域資料表：使用 DynamoDB 進行多區域複寫
- 使用 Amazon Aurora 全球資料庫
- 使用 AWS Services 部落格系列建立多區域應用程式
- 什麼是 AWS Outposts？

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
- AWS re:Invent 2019：AWS 全球網路基礎設施的創新和營運 (NET339)

REL10-BP02 為您的多位置部署選取適當位置

預期成果

如需高可用性，請一律 (如果可能) 將工作負載元件部署到多個可用區域 (AZ)，如圖 10 所示。對於具有極端彈性要求的工作負載，請仔細評估多區域架構的選項。

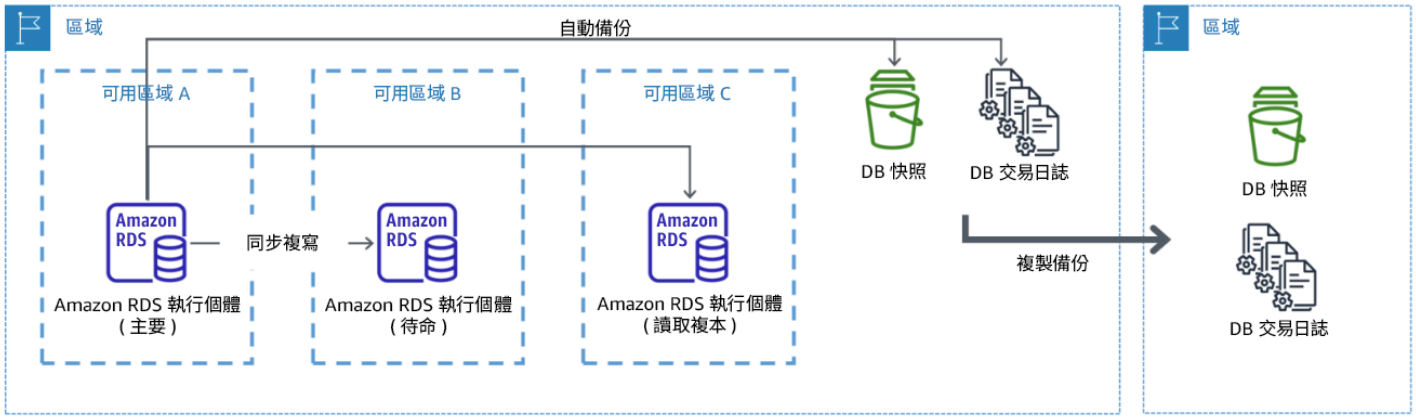


圖 10：備份至另一個 AWS 區域的彈性異地同步備份資料庫部署

常用的反模式

- 當異地同步備份架構滿足要求時，選擇設計多區域架構。
- 如果這些元件之間的彈性和多位置要求不同，則不考慮應用程式元件之間的相依性。

建立此最佳實務的優勢

對於彈性，您應該使用建置防禦層的方法。一層透過使用多個 AZ 建置高度可用的架構來防範更小、更常見的中斷。另一防御層旨在防範發生罕見事件，例如廣泛的自然災害和區域級中斷。這第二層涉及架構您的應用程序以跨越多個 AWS 區域。

- 99.5% 可用性和 99.99% 可用性之間的差異每月超過 3.5 小時。如果工作負載位於多個可用區域中，則工作負載的預期可用性只能達到「四個九」。
- 透過在多個可用區域中執行您的工作負載，您可以隔離電源、冷卻和聯網中的故障，以及火災和洪水等大多數自然災害。
- 針對您的工作負載實作多區域策略有助於其防範影響國家一大片地理區域的廣泛自然災害，或整個區域範圍的技術失敗。請注意，實作多區域架構可能相當複雜，並且通常對於大多數工作負載而言不是必要的。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

若是基於一個可用區域之中斷或局部損失的災難事件，在單一 AWS 區域內的多個可用區域中實作高可用工作負載，可緩解自然發生的災難和技術性災難。每個 AWS 區域都是由多個可用區域構成，每個可用區域都會與其他區域中的錯誤隔離開來，而且會隔開有意義的距離。不過，災難事件若包括失去多個

可用區域元件的風險，而這些元件彼此相距甚遠，您應該實作災難復原選項，以緩解整個區域範圍的失敗。對於需要極端彈性的工作負載 (關鍵基礎設施、健康相關應用程式、金融系統基礎設施等)，可能需要多區域策略。

實作步驟

1. 評估您的工作負載並判斷異地同步備份方法 (單一 AWS 區域) 是否可以滿足彈性需求，或者它們是否需要多區域方法。實作多區域架構來滿足這些要求將引進額外的複雜性，因此請仔細考慮您的使用案例及其要求。使用單一 AWS 區域，幾乎可以一律符合彈性要求。在判斷是否需要使用多個區域時，請考慮以下可能的要求：
   a. 災難復原 (DR)：若是基於一個可用區域之中斷或局部損失的災難事件，在單一 AWS 區域內的多個可用區域中實作高可用工作負載，可緩解自然發生的災難和技術性災難。災難事件若包括失去多個可用區域元件的風險，而這些元件彼此相距甚遠，您應該跨多個區域實作災難復原，以緩解整個區域範圍的自然災難或技術失敗。
   b. 高可用性 (HA)：多區域架構 (在每個區域中使用多個可用區域) 可以用來實現大於四個 9 (> 99.99%) 的可用性。
   c. 堆疊本地化：將工作負載部署到全球對象時，您可以在不同的 AWS 區域 中部署本地化的堆疊，為這些區域中的對象提供服務。本地化可以包括語言、貨幣及存放的資料類型。
   d. 接近使用者： 將工作負載部署到全球對象時，您可以在接近最終使用者所在位置的 AWS 區域中部署堆疊來減少延遲。
   e. 資料落地：某些工作負載受制於資料落地要求，其中來自特定使用者的資料必須保留在特定國家/地區的邊界內。根據討論中的法規，您可以選擇將整個堆疊或只將資料部署到這些邊界內的 AWS 區域。
2. 以下是 AWS 服務提供的異地同步備份功能的一些範例：
   a. 若要使用 EC2 或 ECS 保護工作負載，請在運算資源前面部署 Elastic Load Balancer。然後，Elastic Load Balancing 會提供解決方案，以偵測運作狀態不佳區域中的執行個體，並將流量路由至運作良好的區域。
      i. Application Load Balancers 入門
      ii. Network Load Balancer 入門
   b. 如果執行商務現成軟體的 EC2 執行個體不支援負載平衡，您可以透過實作異地同步備份災難復原方法來實現某種形式的容錯。
      i. the section called “REL13-BP02 使用定義的復原策略來滿足復原目標”
   c. 對於 Amazon ECS 任務，將您的服務平均地部署在三個可用區域之中，以實現可用性與成本的平衡。
      i. Amazon ECS 可用性最佳實務 | 容器

d. 對於非 Aurora Amazon RDS，您可以選擇異地同步備份做為組態選項。在主資料庫執行個體失敗時，Amazon RDS 會自動提升備用資料庫，以接收另一個可用區域中的流量。也可以建立多區域讀取複本來改善彈性。

   i. Amazon RDS 異地同步備份部署

   ii. 在不同的 AWS 區域 中建立讀取複本

3. 以下是 AWS 服務提供的多區域功能的一些範例：

a. 對於服務自動提供異地同步備份可用性的 Amazon S3 工作負載，如果需要多區域部署，請考慮使用多區域存取點。

   i. Amazon S3 中的多區域存取點

b. 對於服務自動提供異地同步備份可用性的 DynamoDB 資料表，您可以輕鬆地將現有的資料表轉換為全域表，以利用多個區域。

   i. 將您的單一區域 Amazon DynamoDB 資料表轉換為全域表

c. 如果您的工作負載面臨 Application Load Balancers 或 Network Load Balancer，請使用 AWS Global Accelerator，透過將流量導向到多個包含運作狀態良好之端點的區域，來改善應用程式的可用性。

   i. AWS Global Accelerator - AWS Global Accelerator 中標準加速器的端點 (amazon.com)

d. 對於利用 AWS EventBridge 的應用程式，請考慮跨區域匯流排，將事件轉送到您選取的其他區域。

   i. 在 AWS 區域 之間傳送和接收 Amazon EventBridge 事件

e. 對於 Amazon Aurora 資料庫，請考慮跨越多個 AWS 區域的 Aurora 全球資料庫。您也可以修改現有的叢集來新增區域。

   i. Amazon Aurora 全球資料庫入門

f. 如果您的工作負載包括 AWS Key Management Service (AWS KMS) 加密金鑰，請考慮多區域金鑰是否適合您的應用程式。

   i. AWS KMS 中的多區域金鑰

g. 如需其他 AWS 服務功能，請在下列一文參閱此部落格系列： 使用 AWS Services 系列建立多區域應用程式

實作計劃的工作量： 中到高

資源

相關文件：

- 使用 AWS Services 系列建立多區域應用程式
- AWS 上的災難復原 (DR) 架構，第 IV 部分：多站點主動/主動
- AWS 全球基礎設施
- AWS Local Zones 常見問答集
- AWS 上的災難復原 (DR) 架構，第 I 部分：在雲端中復原的策略
- 災難復原在雲端中有所不同
- 全域資料表：使用 DynamoDB 進行多區域複寫

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
- Auth0：多區域高可用架構，可擴展至 1.5B+搭配自動容錯移轉的一個月登入

相關範例：

- AWS 上的災難復原 (DR) 架構，第 I 部分：在雲端中復原的策略
- DTCC 所達成的彈性程度遠超乎其在內部部署所能達到的
- Expedia Group 使用多區域、多可用區域架構，搭配專有的 DNS 服務，為應用程式提高彈性
- 使用者：多區域 Kafka 的災難復原
- Netflix：多區域彈性的主動-主動
- 我們如何為 Atlassian Cloud 建置資料彈性
- Intuit TurboTax 在兩個區域上執行

REL10-BP03 針對限制在單一位置的元件將復原自動化

如果工作負載的元件只能在單一可用區域或內部部署資料中心執行，您必須在定義的復原目標內實作完整重建工作負載的功能。

如果因為技術限制而無法實作將工作負載部署至多個位置的最佳實務，您必須實作彈性的替代路徑。您必須將以下能力自動化：重新建立必要基礎設施、重新部署應用程式，以及針對這些案例重新建立必要資料。

例如，Amazon EMR 會在相同可用區域中啟動指定叢集的所有節點，因為在相同區域執行叢集可以提供更高的資料存取速率，從而能提高任務流程的效能。如果為實現工作負載彈性而需要此元件，您必須要有方法重新部署叢集及其資料。此外，對於 Amazon EMR，您還應以異地同步備份以外的方式佈建

冗餘。您可以佈建 多個節點。使用 EMR 檔案系統 (EMRFS)時，EMR 中的資料可存放在 Amazon S3 中，然後可複寫至多個可用區域或 AWS 區域。

同樣地，對於 Amazon Redshift，它預設會將叢集佈建在您所選 AWS 區域內隨機選取的可用區域中。所有叢集節點都佈建在相同區域中。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 實作自我修復。盡可能使用 Automatic Scaling 來部署執行個體或容器。如果無法使用 Automatic Scaling，則對 EC2 執行個體使用自動復原，或者根據 Amazon EC2 或 ECS 容器生命週期事件實作自我修復自動化。
  - 對於不需要單個執行個體 IP 地址、私有 IP 地址、彈性 IP 地址和執行個體中繼資料的執行個體和容器工作負載，使用 Auto Scaling 群組。
    - 什麼是 EC2 自動擴展？
    - 服務自動擴展
      - 啟動組態使用者資料可用於實作自動自我修復大多數工作負載。
  - 對於需要單個執行個體 IP 地址、私有 IP 地址、彈性 IP 地址和執行個體中繼資料的工作負載，使用 EC2 執行個體的自動復原。
    - 復原您的執行個體
      - 在偵測到執行個體失敗時，自動復原會將提醒傳送到 SNS 主題。
  - 在無法使用 Auto Scaling 或 EC2 復原的情況下，使用 EC2 執行個體生命週期事件或 ECS 事件自動執行自我修復。
    - EC2 Auto Scaling 生命週期掛鉤
    - Amazon ECS 事件
      - 使用事件來叫用自動化，以根據您所需的過程邏輯來修復您的元件。

資源

相關文件：

- Amazon ECS 事件
- EC2 Auto Scaling 生命週期掛鉤
- 復原您的執行個體
- 服務自動擴展

- 什麼是 EC2 自動擴展？

REL10-BP04 使用隔板架構限制影響範圍

如同船舶上的隔板一樣，此模式可確保失敗限制在一小部分的請求或用戶端中，如此一來才能限制受損的請求數量，讓大部分的請求可以繼續運行，而不會發生錯誤。資料的隔板通常稱為分割區，而服務的隔板稱為儲存格。

在 以儲存格為基礎的架構中，每個儲存格都是完整的獨立服務執行個體，且具有固定的大小上限。隨著負載增加，工作負載會增加更多儲存格。分割區索引鍵用於傳入流量，以判斷使用哪個儲存格處理請求。任何失敗都會限制在它發生的單一儲存格之中，因此受損的請求數量會受到限制，而其他儲存格會繼續運行，且不會發生錯誤。務必要識別適當的分割區索引鍵，以最大程度地減少跨儲存格互動，並避免在每個請求中都加入複雜的對應服務。需要複雜對應的服務最終僅僅是將問題移轉到了對應服務，而需要跨儲存格互動的服務則會在儲存格之間建立相依性 (因此減低了假定的可用性改善)。

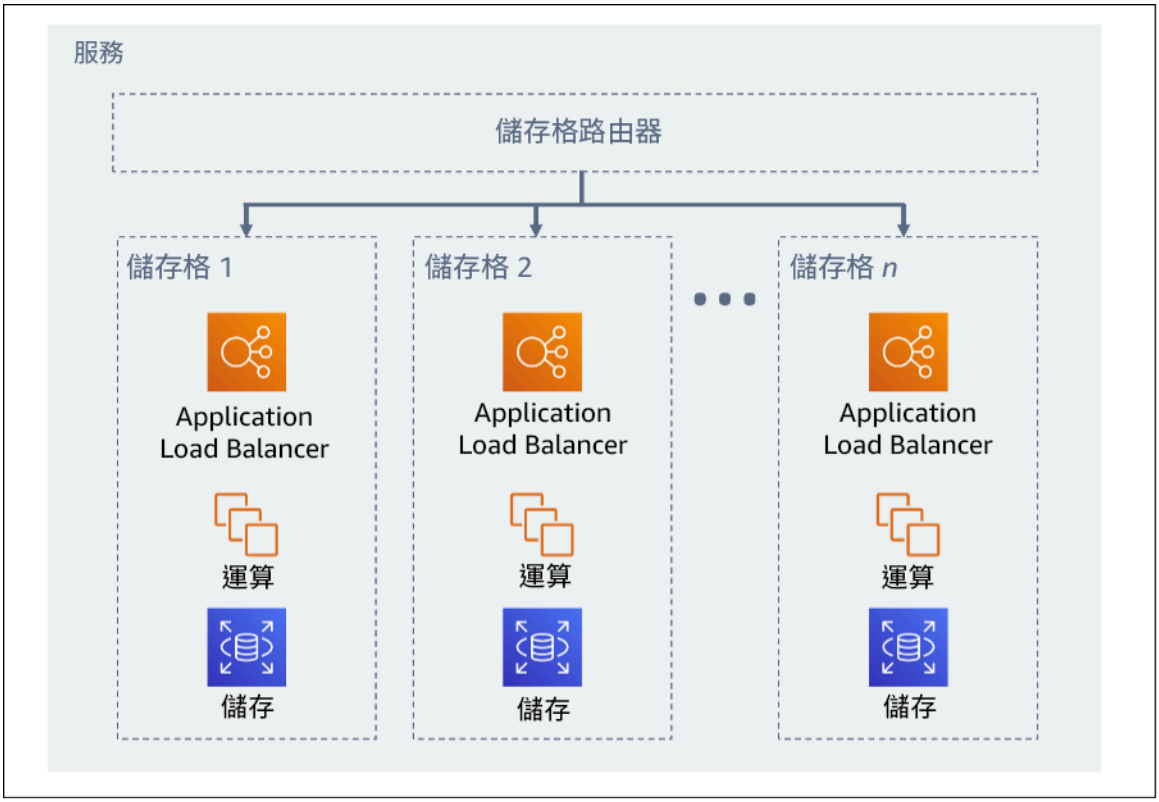


圖 11：儲存格型架構

Colm MacCarthaigh 在他的 AWS 部落格文章中說明 Amazon Route 53 如何使用 隨機切換分區的概念，將客戶請求隔離為分區。此案例中的分區包含兩個或更多個儲存格。根據分割區索引鍵，來自客戶 (或資源，或任何您想要隔離的項目) 的流量會路由至其指派的分區。如果有八個儲存格，每個分區為兩個儲存格，客戶會分割成四個分區，有 25% 的客戶會在發生問題時受到影響。

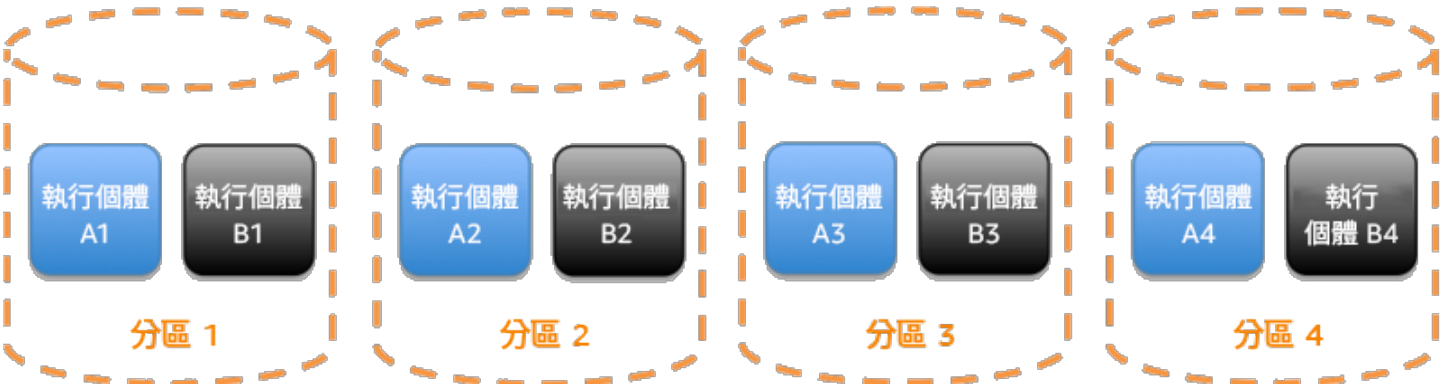


圖 12：服務分為四個傳統分區，每個分區包含兩個儲存格

透過隨機切換分區，您可以建立每個都含有兩個儲存格的虛擬分區，並將您的客戶指派至其中一個虛擬分區。發生問題時，您仍可能會失去整個服務的四分之一，但透過此方式來指派客戶或資源，隨機切換分區的影響範圍遠小於 25%。若有八個儲存格，則會有 28 個含有兩個儲存格的獨特組合，這表示可能會有 28 個隨機切換分區 (虛擬分區)。如果您有數百或數千位客戶，並將每位客戶指派至一個隨機切換分區，則問題造成的影響範圍只有 1/28。這比一般分區好七倍。

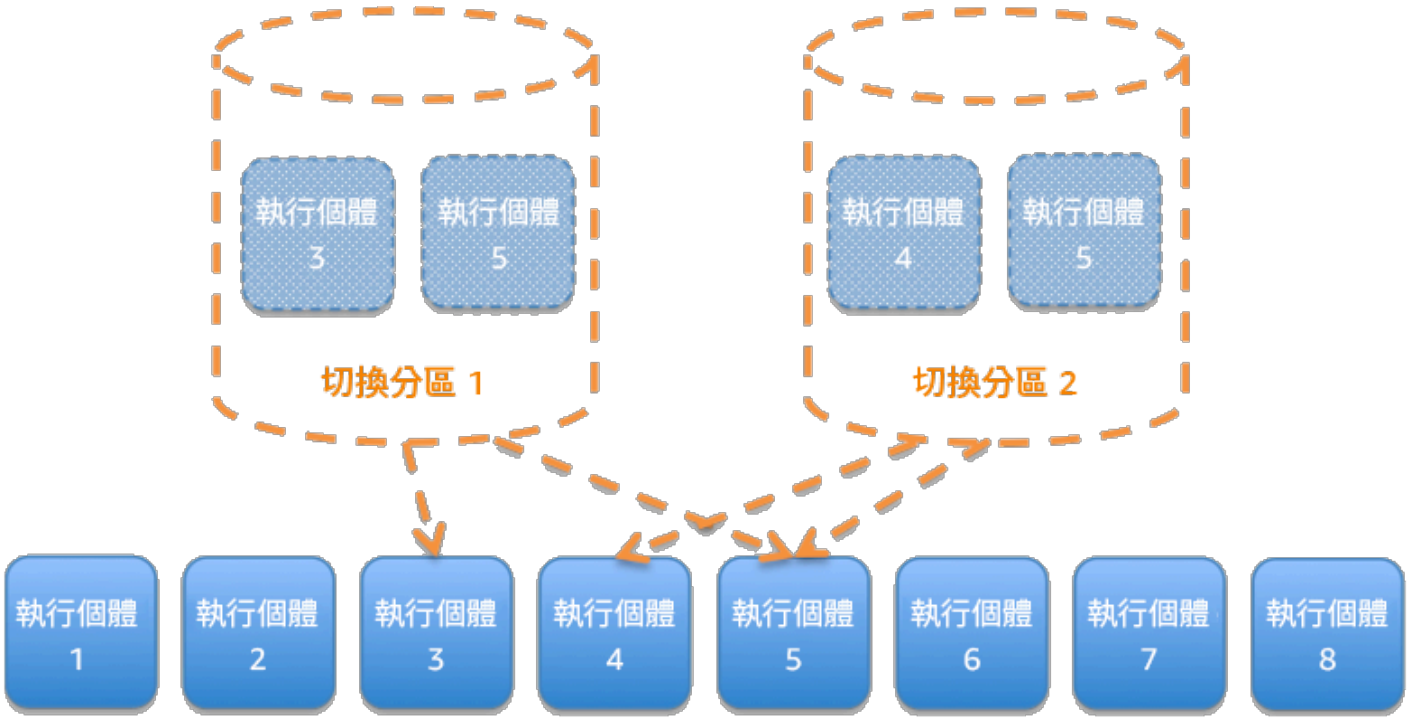


圖 13：將服務分為每個分區擁有兩個儲存格的 28 個隨機切換分區 (虛擬共用) (僅顯示 28 個可能的隨機切換分區中的兩個)

除儲存格之外，分區還可用於伺服器、佇列或其他資源。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 使用隔板架構。如同船舶上的隔板一樣，此模式可確保失敗限制在一小部分的請求/使用者中，如此一來才能限制受損的請求數量，讓大部分的請求可以繼續運行，而不會發生錯誤。資料的隔板通常稱為分割區，而服務的隔板稱為儲存格。
  - Well-Architected 實驗室：搭配隨機分片的故障隔離
  - 隨機切換分區：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)
  - AWS re:Invent 2018：AWS 如何最大程度地減小故障的影響範圍 (ARC338)
- 評估小組型架構是否適用於您的工作負載。在以儲存格為基礎的架構中，每個儲存格都是完整的獨立服務執行個體，且具有固定的大小上限。隨著負載增加，工作負載會增加更多儲存格。分割區索引鍵用於傳入流量，以判斷使用哪個儲存格處理請求。任何失敗都會限制在它發生的單一儲存格之中，因此受損的請求數量會受到限制，而其他儲存格會繼續運行，且不會發生錯誤。務必要識別適當的分割區索引鍵，以最大程度地減少跨儲存格互動，並避免在每個請求中都加入複雜的對應服務。需要複雜對應的服務最終僅僅是將問題移轉到了對應服務，而需要跨儲存格互動的服務則會降低儲存格的自主性 (因此提高了假定的可用性)。
  - Colm MacCarthaigh 在他的 AWS 部落格文章中說明 Amazon Route 53 如何使用隨機切換分區的概念，將客戶請求隔離為分區
    - 隨機分片：神奇的大型故障隔離

資源

相關文件：

- 隨機分片：神奇的大型故障隔離
- Amazon Builders' Library：使用隨機切換分區隔離工作負載

相關影片：

- AWS re:Invent 2018：AWS 如何最大程度地減小故障的影響範圍 (ARC338)
- 隨機切換分區：AWS re:Invent 2019：Amazon Builders' Library 簡介 (DOP328)

相關範例：

- Well-Architected 實驗室：搭配隨機分片的故障隔離

REL 11 如何設計工作負載以承受元件失敗？

需要高可用性和低平均復原時間 (MTTR) 的工作負載必須建立彈性架構。

最佳實務

- REL11-BP01 監控工作負載的所有元件以偵測失敗
- REL11-BP02 容錯移轉至運作良好的資源
- REL11-BP03 將所有分層的修復自動化
- REL11-BP04 復原期間需使用資料平面，而非控制平面
- REL11-BP05 使用靜態穩定性來防止雙模態行為
- REL11-BP06 當事件影響可用性時傳送通知

REL11-BP01 監控工作負載的所有元件以偵測失敗

持續監控工作負載的運作狀態，讓您和自動化系統在發生效能降低或失敗時能夠察覺。根據商業價值監控關鍵績效指標 (KPI)。

所有復原和修復機制首先都必須能夠快速偵測問題。應該先偵測技術故障，以便解決問題。不過，可用性取決於工作負載提供商業價值的能力，因此測量此需求的關鍵績效指標 (KPI) 必須成為偵測和修復策略的一部分。

常用的反模式：

- 未設定任何警示，因此會在未發出通知的情況下發生停機。
- 警示存在，但在此臨界值下無法提供足夠的回應時間。
- 收集的指標經常不足以符合復原時間目標 (RTO)。
- 只會主動監控面對客戶的工作負載層。
- 只會收集技術指標，不收集業務功能指標。
- 無測量工作負載的使用者體驗的指標。

建立此最佳實務的優勢： 在各層級內進行適當的監控，可讓您減少偵測時間，進而減少復原時間。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 根據您的復原目標決定元件的收集間隔。

  - 您的監控間隔取決於復原必須多快完成。您的復原時間取決於所需的復原時間，因此您必須考量此時間和復原時間目標 (RTO)，藉以決定收集頻率。
- 設定元件的詳細監控。
  - 判斷 EC2 執行個體和 Auto Scaling 是否需要詳細監控。詳細監控提供 1 分鐘的間隔指標，預設監控則提供 5 分鐘的間隔指標。
    - 為執行個體啟用或停用詳細監控
    - 使用 Amazon CloudWatch 監控 Auto Scaling 群組和執行個體
  - 判斷 RDS 是否需要增強型監控。增強型監控使用 RDS 執行個體上的代理程式，以取得 RDS 執行個體上不同處理程序或執行緒的實用資訊。
    - 增強監控
- 建立自訂指標來測量業務關鍵績效指標 (KPI)。工作負載會實作關鍵業務功能。這些功能應做為 KPI，以協助確定何時發生間接問題。
  - 發布自訂指標
- 以使用者 Canary 監控使用者的故障體驗。可執行和模擬客戶行為的綜合交易測試 (也稱為 Canary 測試，但請別與 Canary 部署混淆)，是最重要的測試程序之一。針對來自不同遠端位置的工作負載端點持續執行這些測試。
  - Amazon CloudWatch Synthetics 可讓您建立使用者 Canary
- 建立追蹤使用者體驗的自訂指標。如果您可以檢測客戶的體驗，則可以判斷消費者體驗何時變差。
  - 發布自訂指標
- 設定警示以偵測工作負載的任何部分何時未正常運作，並指示何時自動擴展資源。警示會在儀表板上以視覺化方式顯示、透過 Amazon SNS 或電子郵件傳送提醒，以及使用 Auto Scaling 向上或向下擴展工作負載的資源。
  - 使用 Amazon CloudWatch 警示
- 建立儀表板以視覺化指標。儀表板可以讓您以視覺化方式查看趨勢、極端值和其他潛在問題的指標，或提供您可能想要調查之問題的指示。
  - 使用 CloudWatch 儀表板

資源

相關文件：

- Amazon CloudWatch Synthetics 可讓您建立使用者 Canary
- 為執行個體啟用或停用詳細監控

- 增強監控
- 使用 Amazon CloudWatch 監控 Auto Scaling 群組和執行個體
- 發布自訂指標
- 使用 Amazon CloudWatch 警示
- 使用 CloudWatch 儀表板

相關範例：

- Well-Architected 實驗室：第 300 級：實作運作狀態檢查和管理相依性以提升可靠性

REL11-BP02 容錯移轉至運作良好的資源

可確保如果發生資源故障，運作良好的資源可以繼續為請求提供服務。對於位置故障 (例如可用區域或 AWS 區域)，請確保您的系統已就位，可容錯移轉至未受影響位置中運作良好的資源。

AWS 服務 (例如 Elastic Load Balancing 和 AWS Auto Scaling) 會協助跨資源和可用區域分配負載。因此，可以透過將流量轉移到剩餘運作狀態良好的資源來緩解個別資源 (例如 EC2 執行個體) 的失敗或可用區域的損害。對於多區域工作負載，這會更複雜。例如，跨區域僅供讀取複本讓您可以將資料部署至多個 AWS 區域，但您仍須將僅供讀取複本升階為主節點，並在發生容錯移轉時將流量指向其中。Amazon Route 53 和 AWS Global Accelerator 可以協助跨 AWS 區域 路由流量。

如果您的工作負載使用 Amazon S3 或 Amazon DynamoDB 等 AWS 服務，則它們會自動部署至多個可用區域。如果發生失敗，AWS 控制平面會自動為您路由流量至運作良好的位置。資料以冗餘方式存放在多個可用區域中，並且仍然可用。對於 Amazon RDS，您必須選擇異地同步備份做為組態選項，然後在發生失敗時，AWS 會自動將流量導向至運作良好的執行個體。對於 Amazon EC2 執行個體、Amazon ECS 任務或 Amazon EKS Pod，您可以選擇要部署的可用區域。然後，Elastic Load Balancing 會提供解決方案，以偵測運作狀態不佳區域中的執行個體，並將流量路由至運作良好的區域。Elastic Load Balancing 甚至可將流量路由至內部部署資料中心內的元件。

對於多區域方法 (可能也包含內部部署資料中心)，Amazon Route 53 提供一種定義網際網路網域的方法，並指派包含運作狀態檢查的路由政策，以確保流量路由至運作良好的區域。或者，AWS Global Accelerator 提供靜態 IP 地址，做為應用程式的固定進入點，然後使用 AWS 全球網路 (而不是網際網路) 路由至您所選 AWS 區域中的端點，以獲得更好的效能和可靠性。

AWS 在設計服務時會考慮到故障復原。我們設計服務以最大程度地減少從故障復原的時間以及對資料的影響。我們的服務主要使用僅在將請求持久儲存於區域內的多個複本中之後才確認請求的資料存放區。這些資源和服務包括 Amazon Aurora、Amazon Relational Database Service (Amazon RDS) 異地同步備份資料庫執行個體、Amazon S3、Amazon DynamoDB、Amazon Simple Queue Service

(Amazon SQS) 和 Amazon Elastic File System (Amazon EFS)。它們經建構為使用基於儲存格的隔離以及可用區域提供的故障隔離。我們在營運程序中廣泛使用自動化。我們還對我們的取代-重啟功能進行優化，以期從中斷中快速復原。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 容錯移轉至運作良好的資源。可確保如果發生資源故障，運作良好的資源可以繼續為請求提供服務。對於位置故障 (例如可用區域或 AWS 區域)，請確保您的系統已就位，可容錯移轉至未受影響位置中運作良好的資源。
  - 如果您的工作負載使用 Amazon S3 或 Amazon DynamoDB 等 AWS 服務，則它們會自動部署至多個可用區域。如果發生失敗，AWS 控制平面會自動為您路由流量至運作良好的位置。
  - 對於 Amazon RDS，您必須選擇異地同步備份做為組態選項，然後在發生失敗時，AWS 會自動將流量導向至運作良好的執行個體。
    - Amazon RDS 的高可用性 (多可用區域)
  - 對於 Amazon EC2 執行個體或 Amazon ECS 任務，您可以選擇要部署的可用區域。然後，Elastic Load Balancing 會提供解決方案，以偵測運作狀態不佳區域中的執行個體，並將流量路由至運作良好的區域。Elastic Load Balancing 甚至可將流量路由至內部部署資料中心內的元件。
  - 如果採用多區域方法 (可能也包含內部部署資料中心)，確保來自運作狀態良好之位置的資料和資源可以繼續為請求提供服務
    - 例如，跨區域僅供讀取複本讓您可以將資料部署至多個 AWS 區域，但您仍須將僅供讀取複本升階為主節點，並在主要位置發生失敗時將流量指向該主節點。
      - Amazon RDS 讀取複本的概觀
    - Amazon Route 53 提供一種方法，可定義網際網路網域和指派路由政策 (可能包含運作狀態檢查)，以確保流量路由到運作狀態良好的區域。或者，AWSGlobal Accelerator 提供靜態 IP 地址，做為應用程式的固定進入點，然後使用 AWS 全球網路 (而不是公用網際網路) 路由至您所選 AWS 區域中的端點，以獲得更好的效能和可靠性。
      - Amazon Route 53：選擇路由政策
      - 什麼是 AWS Global Accelerator？

資源

相關文件：

- APN 合作夥伴：可以幫助您實現容錯自動化的合作夥伴

- AWS Marketplace：可用於容錯的產品
- AWS OpsWorks：使用自動修復來替換故障的執行個體
- Amazon Route 53：選擇路由政策
- Amazon RDS 的高可用性 (多可用區域)
- Amazon RDS 讀取複本的概觀
- Amazon ECS 任務置放策略
- 為多個可用區域建立 Kubernetes Auto Scaling 群組
- 什麼是 AWS Global Accelerator？

相關範例：

- Well-Architected 實驗室：第 300 級：實作運作狀態檢查和管理相依性以提升可靠性

REL11-BP03 將所有分層的修復自動化

偵測到失敗時，使用自動化功能執行動作來進行修復。

重新啟動的能力 是修復故障的重要工具。如同先前針對分散式系統的討論一樣，最佳實務是盡可能讓服務無狀態。這可防止重新啟動時遺失資料或可用性。在雲端，您可以在重新啟動時 (且通常應該) 取代整個資源 (例如，EC2 執行個體或 Lambda 函數)。重新啟動本身是從故障中復原的一個簡單、可靠方法。工作負載中會發生許多不同類型的故障。硬體、軟體、通訊和營運可能會發生故障。與其建構新穎的機制來捕獲、識別並修正每個不同類型的故障，不如將許多不同類型的故障映射至相同的復原策略。執行個體可能因為硬體故障、作業系統錯誤、記憶體洩漏或其他原因而發生故障。不要為每個情況建置自訂修復，而是將任何情況都視為執行個體故障。終止執行個體，並允許 AWS Auto Scaling 取代之。之後，請對故障的頻外資源執行分析。

另一個範例是能夠重新啟動網路請求。對網路逾時和相依系統故障 (其中相依系統會返回錯誤) 套用相同的復原方法。這兩個事件對系統具有類似的影響，因此，不要嘗試讓任何一個事件成為「特殊情況」，而是藉由指數退避和抖動來採用類似的限制重試策略。

重新啟動的能力 是復原導向運算和高可用性叢集架構中的一種復原機制。

Amazon EventBridge 可用來監控並篩選事件，例如 CloudWatch 警示或其他 AWS 服務的狀態變更。根據事件資訊，它可以觸發 AWS Lambda、AWS Systems Manager Automation 或其他目標，在您的工作負載上執行自訂修復邏輯。

Amazon EC2 Auto Scaling 可設定為檢查 EC2 執行個體的運作狀態。如果執行個體處於執行中以外的任何狀態，或系統狀態為受損，Amazon EC2 Auto Scaling 會將執行個體視為運作狀態不佳，並啟動

替代執行個體。如果使用 AWS OpsWorks，您可以在 OpsWorks 層級中設定 EC2 執行個體的自動修復功能。

對於大規模替換 (例如遺失整個可用區域)，靜態穩定性是高可用性的首選，而不是一次嘗試取得多個新資源。

常用的反模式：

- 個別部署執行個體或容器中的應用程式。
- 部署不透過自動復原就無法部署到多個位置的應用程式。
- 手動復原自動擴展和自動復原無法修復的應用程式。

建立此最佳實務的優勢： 即使工作負載一次只能部署到一個位置，自動修復也會減少平均復原時間，並確保工作負載的可用性。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 使用 Auto Scaling 群組在工作負載中部署分層。Auto Scaling 可以對無狀態應用程式進行自我修復，並新增和移除容量。
  - AWS Auto Scaling 的運作方式
- 對已部署無法在多個位置中部署之應用程式，且可以容忍失敗後重新開機的 EC2 執行個體，實作自動復原。在應用程式無法部署於多個位置時，自動復原可以用來取代失敗硬體並重新啟動執行個體。執行個體中繼資料和相關聯的 IP 地址，以及 Amazon EBS 磁碟區和 Lustre 及 Windows 的彈性檔案系統或檔案系統的掛載點皆會保留。
  - Amazon EC2 自動復原
  - Amazon Elastic Block Store (Amazon EBS)
  - Amazon Elastic File System (Amazon EFS)
  - 什麼是 Amazon FSx for Lustre？
  - 什麼是 Amazon FSx for Windows File Server？
    - 您可以使用 AWS OpsWorks，在層級中設定 EC2 執行個體的自動修復功能。
      - AWS OpsWorks：使用自動修復來替換故障的執行個體
- 當您無法使用自動擴展或自動復原，或自動復原失敗時，則使用 AWS Step Functions 和 AWS Lambda 實作自動復原。當您無法使用自動擴展，且無法使用自動復原或自動復原失敗時，則可以使用 AWS Step Functions 和 AWS Lambda 將修復作業自動化。

- 什麼是 AWS Step Functions？
- 什麼是 AWS Lambda？
  - Amazon EventBridge 可用來監控並篩選事件，例如 CloudWatch 警示或其他 AWS 服務的狀態變更。根據事件資訊，它接著可以觸發 AWS Lambda (或其他目標)，在您的工作負載上執行自訂修復邏輯。
    - 什麼是 Amazon EventBridge？
    - 使用 Amazon CloudWatch 警示

資源

相關文件：

- APN 合作夥伴：可以幫助您實現容錯自動化的合作夥伴
- AWS Marketplace：可用於容錯的產品
- AWS OpsWorks：使用自動修復來替換故障的執行個體
- Amazon EC2 自動復原
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic File System (Amazon EFS)
- AWS Auto Scaling 的運作方式
- 使用 Amazon CloudWatch 警示
- 什麼是 Amazon EventBridge？
- 什麼是 AWS Lambda？
- AWS Systems Manager Automation
- 什麼是 AWS Step Functions？
- 什麼是 Amazon FSx for Lustre？
- 什麼是 Amazon FSx for Windows File Server？

相關影片：

- AWS 中的靜態穩定性：AWS re:Invent 2019：The Amazon Builders' Library 簡介 (DOP328)

相關範例：

- Well-Architected 實驗室：第 300 級：實作運作狀態檢查和管理相依性以提升可靠性

REL11-BP04 復原期間需使用資料平面，而非控制平面

控制平面是用來配置資源，資料平面用來提供服務。一般而言，資料平面的可用性設計目標會高於控制平面，且較為簡單。針對有可能影響彈性的事件實作復原或緩解回應時，使用控制平面作業可能會降低架構的整體彈性。例如，您可以使用 Amazon Route 53 資料平面，根據運作狀態檢查可靠地路由 DNS 查詢，但更新 Route 53 路由政策需使用控制平面，所以不要將控制平面用於復原。

Route 53 資料平面回答 DNS 佇列，以及執行並評估運作狀態檢查。它們遍布全球，而且針對 100% 可用性服務水準協議 (SLA) 設計的。 您可在其中建立、更新和刪除 Route 53 資源的 Route 53 管理 API 和主控台，在控制平面上執行，這些控制平面的設計旨在優先考慮您在管理 DNS 時所需的強式一致性和耐久性。為了實現此目標，控制平面位於單一區域 US East (N. Virginia) 中。儘管將這兩個系統建置為非常可靠，但控制平面未包含在 SLA 中。在極少數情況下，資料平面的彈性設計允許它保持可用性，而控制平面則不允許。對於災難復原和容錯移轉機制，使用資料平面功能提供可能最好的可靠性。

如需資料平面、控制平面，以及 AWS 如何建置服務以符合高可用性目標的詳細資計，請參閱 使用可用區域實現靜態穩定性 文件和 Amazon 建置者資料中心。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 將 Amazon Route 53 用於災難復原時，請使用資料平面，而非控制平面。Route 53 應用程式復原控制器可協助您使用準備度檢查和路由控制，來管理和協調容錯移轉。這些功能會持續監控應用程式從失敗中復原的功能，可讓您在多個 AWS 區域、可用區域和內部部署上控管應用程式復原。
  - 什麼是 Route 53 應用程式復原控制器？
  - 使用 Amazon Route 53 建立災難復原機制
  - 使用 Amazon Route 53 應用程式復原控制器建置高彈性應用程式 (第 1 部分)：單一區域堆疊
  - 使用 Amazon Route 53 應用程式復原控制器建置高彈性應用程式，第 2 部分：單一區域堆疊
- 了解哪些作業位於資料平面，哪些位於控制平面。
  - Amazon Builders' Library：控管較小服務，避免分散式系統過載
  - Amazon DynamoDB API (控制平面和資料平面)
  - AWS Lambda 執行 (分割成控制平面和資料平面)
  - AWS Lambda 執行 (分割成控制平面和資料平面)

資源

相關文件：

- APN 合作夥伴：可以幫助您實現容錯自動化的合作夥伴
- AWS Marketplace：可用於容錯的產品
- Amazon Builders' Library：控管較小服務，避免分散式系統過載
- Amazon DynamoDB API (控制平面和資料平面)
- AWS Lambda 執行 (分割成控制平面和資料平面)
- AWS 資料平面
- 使用 Amazon Route 53 應用程式復原控制器建置高彈性應用程式 (第 1 部分)：單一區域堆疊
- 使用 Amazon Route 53 應用程式復原控制器建置高彈性應用程式，第 2 部分：單一區域堆疊
- 使用 Amazon Route 53 建立災難復原機制
- 什麼是 Route 53 應用程式復原控制器？

相關範例：

- 簡介 Amazon Route 53 應用程式復原控制器

REL11-BP05 使用靜態穩定性來防止雙模態行為

雙模態行為是您的工作負載在正常和失敗模式下展現出不同的行為，例如，當可用區域失敗時，仰賴啟動新的執行個體。您應改為建置靜態穩定且僅以一種模式操作的工作負載。在這種情況下，如果移除一個可用區域，則在每個可用區域佈建足夠的執行個體來處理工作負載負載，然後使用 Elastic Load Balancing 或 Amazon Route 53 運作狀態檢查，將負載從受損的執行個體移出。

運算部署 (例如 EC2 執行個體或容器) 的靜態穩定性可提供最高的可靠性。這必須與成本考量進行權衡。佈建較少的運算容量，而且在發生故障時仰賴啟動新的執行個體，所需的成本會更低。但對於大規模故障 (例如可用區域故障)，此方法效率較低，因為它仰賴在發生故障時對受損情況做出回應，而不是在發生之前為這些受損情況做好準備。您的解決方案應該權衡可靠性與工作負載的成本需求。透過使用更多可用區域，靜態穩定性所需的額外運算量會減少。

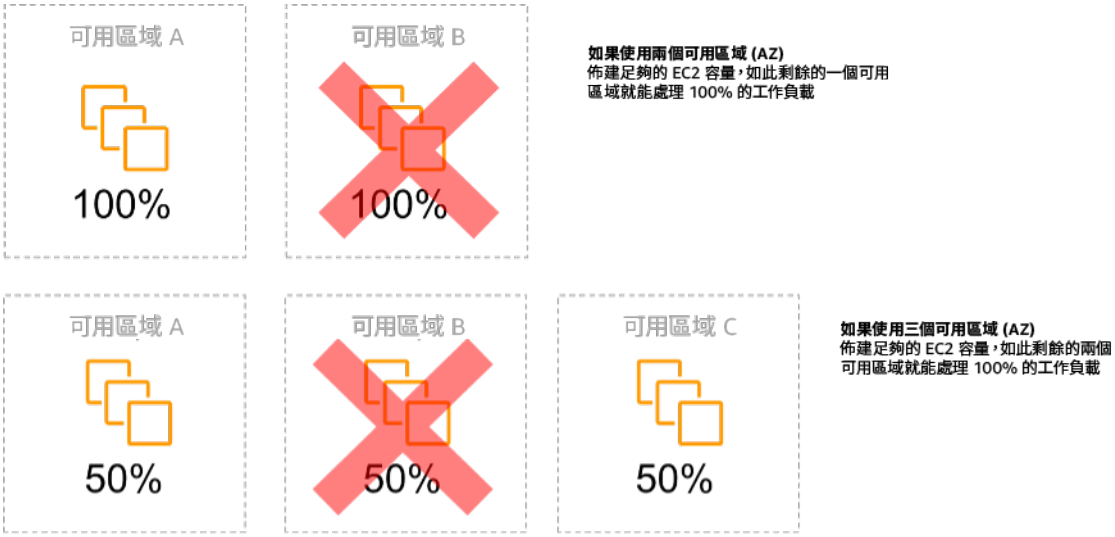


圖 14：跨可用區域之 EC2 執行個體的靜態穩定性

流量轉移後，使用 AWS Auto Scaling 以非同步方式取代故障區域的執行個體，並在運作良好的區域中啟動這些執行個體。

另一個雙模態行為範例是網路逾時，網路逾時可能導致系統嘗試重新整理整個系統的組態狀態。這樣一來，即會給另一個元件新增意外負載，且可能導致其發生故障，從而引發其他意外後果。這種負面意見回饋迴圈會影響工作負載的可用性。反之，您應該建置靜態穩定且僅以一種模式操作的系統。靜態穩定的設計是執行持續工作，並始終以固定的規律重新整理組態狀態。叫用失敗時，工作負載會使用先前的快取數值，並觸發警示。

另一個雙模態行為範例是允許用戶端在發生失敗時繞過您的工作負載快取。這看起來可能是滿足用戶端需求的解決方案，但不應得到允許，因為這會大幅變更工作負載的需求，且可能導致故障。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 使用靜態穩定性來防止雙模態行為。雙模態行為是您的工作負載在正常和失敗模式下展現出不同的行為，例如，當可用區域失敗時，仰賴啟動新的執行個體。
  - 在災難復原計畫中盡可能減少相依關係
  - Amazon Builders' Library：使用可用區域實現靜態穩定性
  - AWS 中的靜態穩定性：AWS re:Invent 2019：The Amazon Builders' Library 簡介 (DOP328)
    - 您應改為建置靜態穩定且僅以一種模式操作的系統。在這種情況下，如果移除一個 AZ，則在每個區域佈建足夠的執行個體來處理工作負載負載，然後使用 Elastic Load Balancing 或 Amazon Route 53 運作狀態檢查，將負載從受損的執行個體移出。

- 另一個雙模態行為範例是允許用戶端在發生失敗時繞過您的工作負載快取。這看起來可滿足用戶端需求的解決方案，但不應允許，因為這會大幅變更工作負載的需求，且可能導致失敗。

資源

相關文件：

- 在災難復原計畫中盡可能減少相依關係
- Amazon Builders' Library：使用可用區域實現靜態穩定性

相關影片：

- AWS 中的靜態穩定性：AWS re:Invent 2019：The Amazon Builders' Library 簡介 (DOP328)

REL11-BP06 當事件影響可用性時傳送通知

當偵測到重大事件時傳送通知，即使事件造成的問題已自動解決。

自動修復功能可讓您的工作負載變得可靠。不過，它也可能會遮蔽需要解決的潛在問題。實作適當的監控和事件，讓您能夠偵測到問題模式 (包括自動修復功能處理的問題模式)，以便解決根本原因問題。Amazon CloudWatch 警示可根據發生的故障來觸發，也可以根據執行的自動修復動作來觸發。CloudWatch 警示可設定為傳送電子郵件，或使用 Amazon SNS 整合在第三方事件追蹤系統中記錄事件。

常用的反模式：

- 傳送無人對其採取行動的警示。
- 進行自動修復自動化，但不通知需要修復。

建立此最佳實務的優勢： 復原事件的通知可確保您不會忽略不常發生的問題。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 當業務關鍵績效指標超過臨界值下限時，發出該指標的警示：對業務 KPI 制定臨界值下限警示，有助於您知道何時無法使用工作負載或工作負載無法運作。
  - 根據靜態臨界值建立 CloudWatch 警示

- 叫用修復自動化的事件警示：您可以直接叫用 SNS API，以透過您建立的任何自動化來傳送通知。
  - 什麼是 Amazon Simple Notification Service？

資源

相關文件：

- 根據靜態臨界值建立 CloudWatch 警示
- 什麼是 Amazon EventBridge？
- 什麼是 Amazon Simple Notification Service？

REL 12 如何測試可靠性？

在將工作負載設計為可彈性應對生產壓力之後，測試是確保其依設計運作並交付您預期之彈性的唯一方法。

最佳實務

- REL12-BP01 使用程序手冊調查失敗
- REL12-BP02 執行事件後分析
- REL12-BP03 測試功能要求
- REL12-BP04 測試擴展和效能需求
- REL12-BP05 使用混沌工程測試彈性
- REL12-BP06 定期執行演練日

REL12-BP01 使用程序手冊調查失敗

透過在程序手冊中記錄調查程序，實現對無法充分理解的失敗情境進行快速一致的回應。程序手冊是為識別造成失敗情境的因素所執行的預先定義步驟。在確定或向上呈報問題之前，任何程序步驟的結果都用於確定要採取的後續步驟。

程序手冊是您必須進行的主動規劃，然後才能有效地採取回應動作。在生產環境中遇到程序手冊未涵蓋的故障情境時，請先解決問題 (解決燃眉之急)。然後返回並查看您為解決問題所採取的步驟，並使用這些步驟在程序手冊中新增新的項目。

請注意，程序手冊用於回應特定事件，而執行手冊則用於實現特定成果。執行手冊通常用於例行活動，而程序手冊則用於回應非例行事件。

常用的反模式：

- 在不知道診斷問題或回應事件的程序之情況下，規劃部署工作負載。
- 調查事件時，未規劃即決定要向哪些系統收集日誌和指標。
- 指標和事件的保留時間過短，無法用以擷取資料。

建立此最佳實務的優勢： 擷取程序手冊可確保一致地遵循程序。有系統地編纂您的程序手冊可限制手動活動引入錯誤。程序手冊自動化可免除團隊成員介入的需要，或在介入開始時提供其他資訊，從而縮短事件回應時間。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 使用程序手冊識別出問題。程序手冊是調查問題的書面程序。透過在程序手冊中記錄程序，對失敗情境做出一致且迅速的回應。程序手冊包含的資訊和指南必須能夠讓技能嫻熟的人員得以收集適用資訊、識別潛在的失敗來源、隔離故障，以及判斷成因 (執行事件後分析)。
  - 將程序手冊實作為程式碼。透過編寫程序手冊指令碼，以程式碼形式執行操作，確保一致性並限制和減少手動程序引起的錯誤。程序手冊可由多個指令碼組成，這些指令碼代表識別成因時可能需要的不同步驟。執行手冊活動可以作為程序手冊活動的一部分被觸發或執行，或者在程序手冊中提示執行，以回應已識別的事件。
    - 透過 AWS Systems Manager 自動化您的操作程序手冊
    - AWS Systems Manager Run Command
    - AWS Systems Manager Automation
    - 什麼是 AWS Lambda？
    - 什麼是 Amazon EventBridge？
    - 使用 Amazon CloudWatch 警示

資源

相關文件：

- AWS Systems Manager Automation
- AWS Systems Manager Run Command
- 透過 AWS Systems Manager 自動化您的操作程序手冊
- 使用 Amazon CloudWatch 警示

- 使用 Canary (Amazon CloudWatch Synthetics)
- 什麼是 Amazon EventBridge？
- 什麼是 AWS Lambda？

相關範例：

- 使用程序手冊和執行手冊將操作自動化

REL12-BP02 執行事件後分析

審查影響客戶的事件，並識別成因和預防性行動項目。使用此資訊來開發緩解措施，以限制或防止事件再次發生。制定可快速有效回應的程序。適當地傳達成因和為目標受眾量身打造的糾正措施。建立一種可以根據需要將這些原因傳達給其他人的方法。

評估現有測試找不到問題的原因。如果測試尚未存在，請為此案例新增測試。

常用的反模式：

- 尋找成因，但未繼續深入尋找其他潛在問題和減輕方法。
- 僅確定人為錯誤原因，不未嘗試可防止人為錯誤發生的任何培訓或或自動化。

建立此最佳實務的優勢： 進行事件後分析並分享結果，以讓其他實作了相同成因的工作負載減輕風險，並讓工作負載能夠在事件發生前實作減輕措施或自動復原。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 建立事件後分析標準。良好的事件後分析提供了機會，為系統中其他地方使用的架構模式問題提出通用解決方案。
  - 確保成因真實且不責備相關人員。
  - 如果您不記錄問題，則無法糾正它們。
    - 確保事件後分析不會讓相關人員受到責備，這樣您就可以平心靜氣看待建議的糾正措施，並促進應用程式團隊誠實地自我評估與合作。
- 使用程序判斷成因。建立程序來識別和記錄事件的成因，以便您可以制定緩解措施來限制或防止事件再次發生。另外，您還可以制定快速有效地做出回應的程序。適當地傳達成因和為目標受眾量身打造的糾正措施。

- 什麼是日誌分析？

資源

相關文件：

- 什麼是日誌分析？
- 為什麼您應該開發錯誤糾正 (COE)

REL12-BP03 測試功能要求

使用驗證所需功能的單位測試和整合測試等技術。

當這些測試做為建置和部署動作的一部分自動執行時，您會獲得最佳成果。例如，使用 AWS CodePipeline 時，開發人員會將變更遞交至來源儲存庫，而 CodePipeline 會在該儲存庫中自動偵測變更。系統會建置這些變更，並執行測試。測試完成後，會將內建的程式碼部署至預備伺服器以進行測試。CodePipeline 會從預備伺服器執行更多測試，例如整合或負載測試。成功完成這些測試後，CodePipeline 會將已測試及已核准的程式碼部署至生產執行個體。

此外，經驗顯示可執行和模擬客戶行為的綜合交易測試 (也稱為 Canary 測試，但請別與 Canary 部署混淆)，是最重要的測試程序之一。針對來自不同遠端位置的工作負載端點持續執行這些測試。Amazon CloudWatch Synthetics 讓您能夠 建立 Canary， 以監控您的端點和 API。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 測試功能要求。這包括驗證所需功能的單位測試和整合測試。
  - 搭配使用 CodePipeline 與 AWS CodeBuild 以測試程式碼和執行建置
  - AWS CodePipeline 新增對於單位的支援，以及透過 AWS CodeBuild 自訂整合測試
  - 持續交付與持續整合
  - 使用 Canary (Amazon CloudWatch Synthetics)
  - 軟體和測試自動化

資源

相關文件：

- APN 合作夥伴：可以幫助實作持續整合管道的合作夥伴
- AWS CodePipeline 新增對於單位的支援，以及透過 AWS CodeBuild 自訂整合測試
- AWS Marketplace：可用於持續整合的產品
- 持續交付與持續整合
- 軟體和測試自動化
- 搭配使用 CodePipeline 與 AWS CodeBuild 以測試程式碼和執行建置
- 使用 Canary (Amazon CloudWatch Synthetics)

REL12-BP04 測試擴展和效能需求

使用負載測試等技術，以驗證工作負載是否滿足擴展和效能需求。

在雲端，您可以隨需建立工作負載的生產規模測試環境。如果您在縮減的基礎設施上執行這些測試，您必須將觀察到的結果擴展到您認為在生產環境中會發生的情況。如果您很謹慎，力求不影響實際使用者，也可以在生產環境中執行負載和效能測試，並將您的測試資料加上標籤，以免與實際使用者資料混淆並損毀使用統計資料或生產報告。

透過測試，確保您的基本資源、擴展設定、服務配額和彈性設計能夠在負載下如預期運作。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

- 測試擴展和效能需求。進行負載測試，以驗證工作負載是否滿足擴展和效能需求。
  - AWS 上的分散式負載測試：模擬數千名連線的使用者
  - Apache JMeter
    - 在與生產環境相同的環境中部署應用程式並執行負載測試。
      - 使用基礎設施即程式碼概念來建立與您的生產環境盡可能相似的環境。

資源

相關文件：

- AWS 上的分散式負載測試：模擬數千名連線的使用者
- Apache JMeter

REL12-BP05 使用混沌工程測試彈性

定期在位於或盡可能鄰近生產環境的環境中執行混沌試驗，以了解您的系統因應不良狀況的能力。

預期成果：

除了以彈性測試驗證您的工作負載在某事件期間的已知預期行為以外，還可以藉由在錯誤注入試驗中套用混沌工程或注入非預期的負載，來定期驗證工作負載的彈性。結合混沌工程與彈性測試，您將可確信工作負載在經歷元件失敗後仍可存留，並且可在 (幾乎) 不受影響的情況下從非預期的中斷復原。

常見的反模式：

- 針對彈性進行設計，但未確認工作負載在錯誤發生時的整體運作情形。
- 未曾在真實的情況和預期的負載下試驗。
- 未將試驗視為程式碼或透過開發週期加以維護。
- 未在 CI/CD 管道中與部署以外執行混沌試驗。
- 在決定要以哪些錯誤進行試驗時，未使用過去的事故後分析。

建立此最佳實務的優勢： 注入錯誤以驗證工作負載的彈性，可讓您確信在發生真正的錯誤時，彈性設計的復原程序將可發揮作用。

未建立此最佳實務時的曝險等級： 中

實作指引

混沌工程可讓您的團隊有能力以受控的方式，持續在服務供應商、基礎架構、工作負載和元件層級注入真實的中斷 (模擬)，且對客戶 (幾乎) 不會造成影響。它可讓您的團隊從錯誤中學習，並且觀察、測量及改善工作負載的彈性，以及驗證在事件發生時會引發提醒，且團隊會收到通知。

持續執行時，混沌工程可能會凸顯您工作負載中的缺陷，且若未解決，可能會對可用性與操作產生負面影響。

> **Note**
>
> 混沌工程是在系統中進行試驗的專業領域，旨在建立對系統承受生產環境中紊亂情況的能力的信心。 混沌工程的原則

如果系統能夠承受這些中斷，則應將混沌試驗視為自動化迴歸測試來維護。此時，您應在系統開發生命週期 (SDLC) 和 CI/CD 管道中執行混沌試驗。

若要確定您的工作負載可以承受元件失敗，請在試驗中注入真實事件。例如，進行遺失 Amazon EC2 執行個體或容錯移轉主要 Amazon RDS 資料庫執行個體的試驗，並驗證您的工作負載未受影響 (或僅受到些微影響)。使用元件錯誤的組合，模擬可用區域的中斷可能導致的事件。

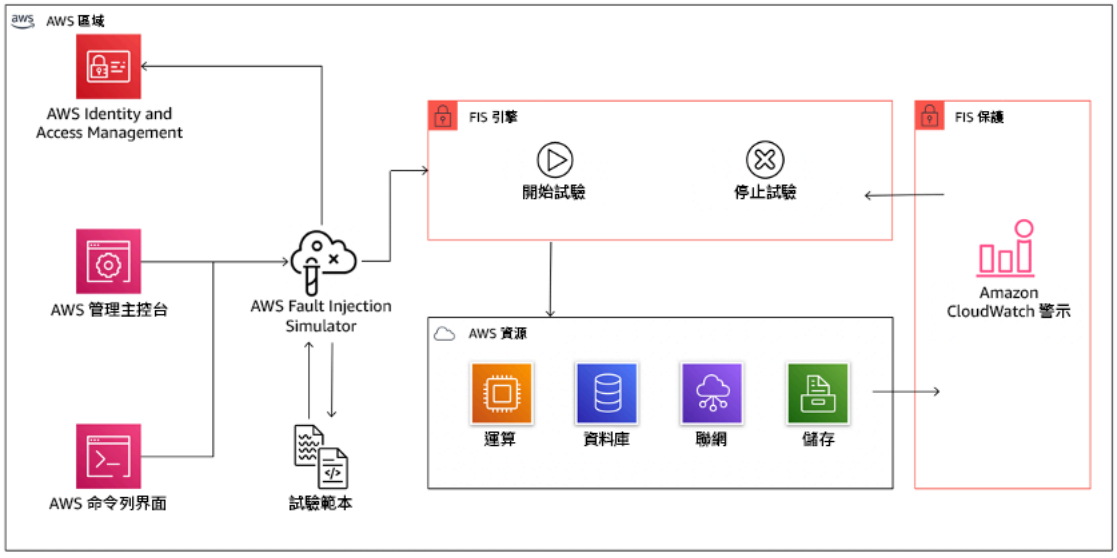
對於應用程式層級的錯誤 (例如當機)，您可以從記憶體和 CPU 用盡之類的壓力源開始著手。

若要對因間歇性網路中斷而產生的外部相依性驗證其 備用或容錯移轉機制 ，您的元件應藉由封鎖對第三方供應商的存取達指定的持續期間 (可延續數秒到數小時)，來模擬這類事件。

其他降級模式可能會導致功能降低和回應速度緩慢，而往往會導致服務中斷。這種降級的常見原因是關鍵服務的延遲增加和不可靠的網路通訊 (丟包)。以這些錯誤 (包括延遲、已捨棄訊息和 DNS 失敗等聯網影響) 進行的試驗，可包含無法解析名稱、無法連線到 DNS 服務，或無法建立相依服務的連線等情境。

混沌工程工具：

AWS Fault Injection Service (AWS FIS) 是用來執行錯誤注入試驗的全受管服務，這些試驗可作為 CD 管道的一部分，或未於管道以外。AWS FIS 很適合在混沌工程演練日期間使用。它支援同時在不同類型的資源間導入錯誤，包括 Amazon EC2、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 和 Amazon RDS。這些錯誤包括終止資源、強制執行容錯移轉、施壓於 CPU 或記憶體、限流，以及封包遺失。由於它與 Amazon CloudWatch 警示整合，因此您可以將停止條件設定為防護機制，以在試驗導致非預期的影響時將其回復。



AWS Fault Injection Service 與 AWS 資源整合，讓您可對工作負載執行錯誤注入試驗。

錯誤注入試驗也有數個第三方選項。其中包括開放原始碼工具，例如 Chaos Toolkit，Chaos Mesh和 Litmus Chaos，以及 Gremlin 之類的商業選項。為了擴展可在 AWS 上注入的錯誤範圍，AWS FIS 與 Chaos Mesh 和 Litmus Chaos 整合，讓您能夠在多項工具間協調錯誤注入工作流程。例如，您可以在使用 AWS FIS 錯誤動作終止隨機選定百分比的叢集節點時，使用 Chaos Mesh 或 Litmus 錯誤對 Pod 的 CPU 執行壓力測試。

實作步驟

- 決定要將哪些錯誤用於試驗。

  評估您的工作負載設計是否有彈性。這類設計 (使用 Well-Architected Framework的最佳實務建立的) 可根據重大相依性、過去的事件、已知問題和合規要求來衡量風險。列出要用來維護彈性的每個設計元素，及其依設計要減輕的錯誤。如需關於建立這類清單的詳細資訊，請參閱 「營運準備度審查」白皮書， 此文件會說明如何建立相關程序來防止過去的事故再次發生。Failure Modes and Effects Analysis (FMEA) 程序提供了相關架構，讓您執行失敗的元件層級分析，並說明失敗對於工作負載有何影響。Adrian Cockcroft 在 Failure Modes and Continuous Resilience 中提供了 FMEA 的詳細說明。

- 指派每個錯誤的優先順序。

  請從粗略的分類開始著手，例如高、中或低。若要評估優先順序，請考量錯誤的頻率，以及失敗對整體工作負載的影響。

  考量特定錯誤的頻率時，請分析此工作負載過去的資料 (如果可用)。如果無法使用，請使用在類似環境中執行的其他工作負載所包含的資料。

  考量特定錯誤的影響時，錯誤的範圍愈大，通常影響就愈大。另請考量工作負載的設計和用途。例如，對執行資料轉換和分析的工作負載而言，存取來源資料存放區的能力至關重要。在此情況下，您應優先執行存取錯誤以及限流存取和延遲注入的試驗。

  事故後分析是您了解失敗模式的頻率與影響的理想資料來源。

  請使用指派的優先順序，決定要先以哪些錯誤進行試驗，以及要以何種順序開發新的錯誤注入試驗。

- 對於您所執行的每個試驗，均應依循混沌工程和連續彈性飛輪操作。

混沌工程和連續彈性飛輪，採用 Adrian Hornsby 的科學方法。

- 將穩定狀態定義為顯示出正常行為之工作負載的某種可測量輸出。

  工作負載的運作若可靠且符合預期，就會呈現穩定狀態。因此，在定義穩定狀態前，請先驗證工作負載的運作狀態良好。穩定狀態不一定表示在錯誤發生時完全不會影響到工作負載，因為有特定百分比的錯誤可能會在可接受的限制內。穩定狀態是您在試驗期間將觀察到的基準，如果您在下一步定義的假設未符合預期，就會凸顯異常。

  例如，支付系統的穩定狀態可定義為 300 TPS、成功率 99%、且來回時間為 500 毫秒的處理。

- 形成關於工作負載將如何回應錯誤的假設。

  良好的假設奠基於工作負載應如何減輕錯誤以維護穩定狀態。假設指出，在發生特定類型的錯誤時，系統或工作負載將持續保有穩定狀態，因為工作負載設有特定緩解機制。特定類型的錯誤和緩解機制應指定於假設中。

  以下是可用於假設的範本 (但也接受其他措辭)：

> **Note**
>
> 若 #### 發生， ###### 工作負載將 ###### 以維護 #########。

例如：

- 若 Amazon EKS 節點群組中有 20% 的節點遭到關閉，Transaction Create API 會在 100 毫秒以內繼續提供第 99 個百分位數的請求 (穩定狀態)。Amazon EKS 節點將在五分鐘內復原，而 Pod 將在試驗起始後的八分鐘內進入排程並處理流量。提醒將在三分鐘內引發。
- 單一 Amazon EC2 執行個體失敗發生時，訂單系統的 Elastic Load Balancing 運作狀態檢查將使 Elastic Load Balancing 僅將請求傳送至其餘運作狀態良好的執行個體，而 Amazon EC2 Auto Scaling 會取代失敗的執行個體，將伺服器端 (5xx) 錯誤的增量維持在 0.01% 以內 (穩定狀態)。
- 主要 Amazon RDS 資料庫執行個體失敗時，供應鏈資料收集工作負載將進行容錯移轉並連線至待命 Amazon RDS 資料庫執行個體，以維持不到 1 分鐘的資料庫讀取或寫入錯誤 (穩定狀態)。

- 藉由注入錯誤來執行試驗。

  試驗依預設應處於安全模式，並獲得工作負載的容許。如果您確知工作負載將失敗，請不要執行試驗。混沌工程應該用來尋找已知的未知或未知的未知。已知的未知 是指您知道，但未能完全了解的事物， 未知的未知 則是指您不知道也未能完全了解的事物。對您確知已失效的工作負載執行試驗，將不會為您帶來新的見解。試驗應經過審慎規劃、具有明確的影響範圍，並且提供在非預期的錯亂發生時可供套用的回復機制。如果您的盡職調查顯示工作負載應可承受試驗，請繼續執行試驗。有數種選項可用來注入錯誤。對於 AWS 上的工作負載，AWS FIS 會提供許多預先定義的錯誤模擬，名為 動作。您也可以定義在 AWS FIS 中執行的自訂動作 (使用 AWS Systems Manager 文件)。

  我們不鼓勵使用自訂指令碼來執行混沌試驗，除非指令碼有能力理解工作負載目前的狀態、能夠發出日誌，並且提供回復機制和停止條件 (若情況允許)。

  支援混沌工程的有效架構或工具集，應追蹤試驗目前的狀態、發出日誌，並提供回復機制以支援受控制的試驗執行。請先從 AWS FIS 這類已建立的服務著手，以便能以明確定義的範圍執行試驗，並且有安全機制可在試驗導入非預期的錯亂時回復試驗。若要進一步了解使用 AWS FIS 的各種試驗，另請參閱 「將彈性和 Well-Architected 應用程式用於混沌工程」實驗室。此外， AWS Resilience Hub 也會分析您的工作負載，並建立可供您選擇在 AWS FIS 中實作並執行的試驗。

> **Note**
>
> 對於每一項試驗，您都應明確了解其範圍與影響。我們建議，錯誤應先在非生產環境中模擬，再於生產環境中執行。

試驗應在真實的負載下執行於生產環境中，且應使用 金絲雀部署 同時推動控制和試驗系統部署 (在情況允許時)。在非尖峰時段執行試驗是很好的做法，可以減少首次在生產環境中試驗時的潛在影響。此外，如果使用實際的客戶流量會伴隨太高的風險，您可以對控制和試驗部署使用生產基礎架構上的綜合流量，來執行試驗。無法使用生產環境時，請在盡可能接近生產環境的生產前環境中執行試驗。

您必須建立防護機制並加以監控，以確定試驗不會超出可接受的限制而影響到生產流量或其他系統。請建立停止條件，以在試驗達到您定義的防護機制指標閾值時，將試驗停止。其中應包括工作負載的穩定狀態指標，以及您對其注入錯誤的元件所適用的指標。路由層 綜合監控 也稱為使用者金絲雀，是您在一般情況下應納入作為使用者代理的指標之一。AWS FIS 的停止條件 被視為試驗範本的一部分受到支援，每個範本最多可啟用五個停止條件。

混沌的準則之一，是盡可能縮小試驗的範圍與影響：

儘管容許某些短期負面影響是必要的，但混沌工程師有責任和義務將試驗的副作用控制在最低限度。

驗證範圍和潛在影響的方法之一，是先在非生產環境中執行試驗，驗證停止條件的閾值在試驗期間會依預期啟動，且有可觀測性會捕捉例外狀況，而不是直接在生產環境中試驗。

執行錯誤注入試驗時，請驗證所有的責任方都會及時獲得通知。請與營運團隊、服務可靠性團隊和客戶支援等適當的團隊通訊，讓他們知道試驗將於何時執行，且預期會有何情況。請為這些團隊提供通訊工具，以便他們在試驗執行期間發現任何不利影響時發出通知。

您必須將工作負載及其基礎系統還原為原始的已知良好狀態。工作負載的彈性設計通常具有自癒能力。但某些錯誤設計或失敗的試驗可能會使您的工作負載處於非預期的失敗狀態。試驗結束時，您必須察覺到這一點，並還原工作負載和系統。透過 AWS FIS，您可以在動作參數內設定回復組態 (也稱為後置動作)。後置動作會將目標回復為動作執行前原有的狀態。無論是自動 (例如使用 AWS FIS) 還是手動，這些後續動作皆應為程序手冊的一部分，以說明如何偵測及處理失敗。

- 驗證假設。

  混沌工程的原則 提供了下列關於如何驗證工作負載穩定狀態的指引：

著重於可測量的系統輸出，而不是系統的內部屬性。這類輸出在一段時間內的測量，會構成系統穩定狀態的代理。整體系統的輸送量、錯誤率和延遲百分位數，全都可能成為呈現穩定狀態行為的相關指標。著重於試驗期間的系統行為模式，混沌工程會驗證系統是否可運作，而非試著驗證其運作情形。

在先前的兩個範例中，我們納入了伺服器端 (5xx) 錯誤的增量低於 0.01% 的穩定狀態指標，以及資料庫讀取和寫入錯誤不到一分鐘的穩定狀態指標。

5xx 錯誤是工作負載的用戶端在失敗模式下將直接經歷的結果，因此可說是良好的指標。資料庫錯誤測量是錯誤的直接產物，因此有其效用，但應同時輔以用戶端影響測量，例如失敗的客戶請求或用戶端遇到的錯誤。此外，請在工作負載的用戶端直接存取的任何 API 或 URI 納入綜合監控 (也稱為使用者金絲雀)。

- 改善工作負載設計的彈性。

  如果未維持穩定狀態，請調查工作負載設計可經由哪些改進來減輕錯誤，運用 AWS Well Architected 可靠性支柱的最佳實務。如需其他指引和資源，請前往 AWS Builder's Library，內含相關文章說明如何 改善您的運作狀態檢查 或 在應用程式碼中使用退避重試，以及其他主題。

  這些變更實作完成後，請再次執行試驗 (在混沌工程飛輪中以虛線表示) 以判斷其有效性。若驗證步驟指出假設成立，則工作負載將處於穩定狀態，且週期會繼續。

- 請定期執行試驗。

  混沌試驗是一個週期，而試驗應被視為混沌工程的一部分定期執行。當工作負載符合試驗的假設後，即應將試驗自動化，以將其視為 CI/CD 管道的迴歸部分持續執行。若要了解其執行方式，請參閱此部落格： 如何使用 AWS CodePipeline 執行 AWS FIS 試驗。此實驗室涉及 CI/CD 管道中的 AWS FIS 試驗， 可讓您進行實際操作。

  錯誤注入試驗也是演練日的一部分 (請參閱 REL12-BP06 定期執行演練日) 建立持續整合/持續部署 (CI/CD) 管道。演練日會模擬失敗或事件，以驗證系統、程序和團隊的應變。目的是實際執行在異常事件發生時團隊將要執行的動作。

- 擷取並儲存試驗結果。

  錯誤注入試驗的結果必須擷取並保存。請納入所有必要資料 (例如時間、工作負載和條件)，以便後續能分析試驗結果和趨勢。舉例來說，結果可包括儀表板的螢幕擷取畫面、指標的資料庫產生的 CSV 傾印，或是試驗中的事件與觀察的手寫記錄。AWS FIS 的試驗記錄 可作為此資料擷取的一部分。

資源

相關的最佳實務：

- REL08-BP03 將彈性測試整合為部署的一部分
- REL13-BP03 測試災難復原實作以驗證實作

相關文件：

- 什麼是 AWS Fault Injection Service？
- 什麼是 AWS Resilience Hub？
- 混沌工程的原則
- 混沌工程：規劃您的第一個試驗
- 彈性工程：學習接受故障
- 混沌工程案例
- 避免分散式系統的備用
- 混沌試驗的金絲雀部署

相關影片：

- AWS re:Invent 2020：使用混沌工程測試彈性 (ARC316)
- AWS re:Invent 2019：透過混沌工程提升彈性 (DOP309-R1)
- AWS re:Invent 2019：在無伺服器環境中執行混沌工程 (CMY301)

相關範例：

- Well-Architected 實驗室：第 300 級：測試 Amazon EC2、Amazon RDS 和 Amazon S3 的彈性
- 「AWS 上的混沌工程」實驗室
- 「將彈性和 Well-Architected 應用程式用於混沌工程」實驗室
- 「無伺服器混沌」實驗室
- 「使用 AWS Resilience Hub 測量及改善您的應用程式彈性」實驗室

相關工具：

- AWS Fault Injection Service
- AWS Marketplace： Gremlin 混沌工程平台
- Chaos Toolkit
- Chaos Mesh
- Litmus

REL12-BP06 定期執行演練日

使用演練日定期執行回應事件和失敗的程序，盡可能接近生產環境 (包括在生產環境中)，並與實際參與失敗情境的人員共同演練。在演練日當天強制執行措施，以確保生產事件不會影響使用者。

演練日模擬失敗或事件，以測試系統、流程和團隊的回應。目的是實際執行在異常事件發生時團隊將要執行的動作。如此可協助您了解何處有改善空間，並能協助發展組織處理活動的經驗。這些應該定期進行，以便您的團隊建置 有關如何回應的 肌肉記憶。

在彈性設計就緒，並已在非生產環境中進行測試之後，演練日就是確保生產中的一切按照計畫運作。演練日，特別是第一個演練日，是一個「全員參與」活動，工程師和操作人員會被告知何時發生，以及會發生什麼情況。執行手冊已就緒。以規定的方式在生產系統中執行模擬事件 (包括可能的失敗事件)，並評估影響。如果所有系統都如設計運作，偵測和自我修復將幾乎不會產生影響。不過，如果觀察到負面影響，測試會回復並視需要手動修復工作負載問題 (使用執行手冊)。由於演練日通常會在生產環境中進行，因此應採取所有預防措施，以確保不會對客戶的可用性造成影響。

常用的反模式：

- 記載您的程序，但絕不練習程序。
- 未在測試練習中納入業務決策者。

建立此最佳實務的優勢： 定期進行演練日可確保所有員工在發生實際事件時遵守政策和程序，並驗證這些政策和程序是否適當。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 安排演練日以定期練習您的執行手冊和程序手冊。演練日應納入生產事件發生時參與的每個人：企業擁有者、開發人員、營運人員和事件反應團隊。
  - 執行負載或效能測試，然後執行錯誤注入。
  - 尋找執行手冊上的異常情況，並尋找練習程序手冊的機會。

- 如果您偏離了執行手冊，應優化執行手冊或更正該行為。如果您練習程序手冊，確定應使用的執行手冊，或建立一個新的執行手冊。

資源

相關文件：

- 什麼是 AWS GameDay？

相關影片：

- AWS re:Invent 2019：透過混沌工程提升彈性 (DOP309-R1)

相關範例：

- AWS Well-Architected 實驗室 - 測試彈性

REL 13 您如何規劃災難復原 (DR)？

備妥備份和冗餘工作負載元件是 DR 策略的開始。RTO 和 RPO 是您還原 工作負載的目標。根據業務需求設定這些目標。實作策略以滿足這些目標，考量工作負載資源和資料的位置和功能。發生中斷的可能性和復原成本也是重要因素，可反映為工作負載提供災難復原的商業價值。

最佳實務

- REL13-BP01 定義停機和資料遺失的復原目標
- REL13-BP02 使用定義的復原策略來滿足復原目標
- REL13-BP03 測試災難復原實作以驗證實作
- REL13-BP04 管理 DR 站點或區域的組態偏移
- REL13-BP05 自動化復原

REL13-BP01 定義停機和資料遺失的復原目標

工作負載具有復原時間目標 (RTO) 和復原點目標 (RPO)。

復原時間目標 (RTO) 是服務中斷與恢復服務之間的最大可接受延遲。這會決定可接受的服務無法使用之時間長度。

復原點目標 (RPO) 是自上次資料復原點之後的最大可接受時間長度。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。

在為您的工作負載選取適用的災難復原 (DR) 策略時，RTO 和 RPO 值是重要的考慮因素。這些目標是由企業決定，然後由技術團隊用來選取和實作 DR 策略。

預期成果：

每個工作負載都獲指派一個 RTO 和 RPO，其是根據業務影響來定義的。工作負載會指派給預先定義的層級，定義服務可用性和可接受的資料遺失，以及相關聯的 RTO 和 RPO。如果這類分層不可行，則可以根據工作負載以定制方式指派此分層，旨在稍後建立層級。RTO 和 RPO 會在選取工作負載的災難復原策略實作時的主要考量之一。挑選 DR 策略的其他考量是成本限制、工作負載相依性和營運要求。

對於 RTO，了解基於中斷持續時間的影響。它是線性的，還是有非線性的影響？(例如，四個小時後，您關閉了一條生產線，直到下一個輪班開始)。

如下的災難復原方法可以協助您了解工作負載關鍵性與復原目標之間的關係。(請注意，X 軸和 Y 軸的實際值應根據您的組織需求加以自訂)。

災難復原方法						
		復原點目標				
		< 1 分鐘	< 1 小時	< 6 小時	< 1 天	+ 1 天
復原時間目標	< 10 分鐘	嚴重	嚴重	高	中	中
	< 2 小時	嚴重	高	中	中	低
	< 8 小時	高	中	中	低	低
	< 24 小時	中	中	低	低	低
	24 + 小時	中	低	低	低	低

圖 16：災難復原方法

常用的反模式：

- 沒有已定義的復原目標。
- 選擇任意復原目標。
- 選擇過於寬鬆且不符合業務目標的復原目標。
- 不了解關機時間和資料遺失的影響。

- 選取不切實際的復原目標，例如零時間復原和零資料遺失，這對於您的工作負載組態可能無法實現。
- 選擇比實際業務目標更嚴格的復原目標。這會被迫進行比工作負載所需更昂貴和更複雜的 DR 實作。
- 選取的復原目標與工作負載的復原目標不相容。
- 您的復原目標未考慮法規合規性要求。
- 已定義工作負載的 RTO 和 RPO，但從未進行測試。

建立此最佳實務的優勢： 需以時間和資料損失的復原目標來引導 DR 實作。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

對於給定的工作負載，您必須了解停機時間和資料遺失對您業務的影響。隨著停機時間或資料遺失的增加，影響會大幅地增長，但這種增長形式可能會根據工作負載類型而有所不同。例如，您可以容忍長達一小時的停機時間而影響不大，但在此之後影響會迅速加大。對業務的影響會以多種形式顯現，包括貨幣成本 (例如收益損失)、客戶信任 (以及對信譽的影響)、營運問題 (例如發不出薪資或生產力下降)，以及監管風險。使用下列步驟來了解這些影響，並為您的工作負載設定 RTO 和 RPO。

實作步驟

1. 確定此工作負載的業務利害關係人，並與他們一起實作這些步驟。工作負載的復原目標是業務決策。然後，技術團隊與業務利害關係人合作，使用這些目標來選取 DR 策略。

   > **Note**
   >
   > 針對步驟 2 和 3，您可以使用 the section called “實作工作表”。

2. 收集必要資訊，藉由回答下列問題來做出決策。
3. 對於組織中的工作負載影響，您是否具有關鍵性的類別或層級？
   a. 若是，請將此工作負載指派給類別。
   b. 若否，請建立這些類別。建立五個或更少的類別，然後縮小每個類別的復原時間目標範圍。範例類別包括：重大、高、中、低。若要了解工作負載如何對應至類別，請考慮工作負載是任務為主、業務為主，還是非業務推動。
   c. 根據類別設定工作負載 RTO 和 RPO。一律選擇比進入此步驟所計算的原始值更嚴格的類別 (更低的 RTO 和 RPO)。如果這導致值發生不適當的大變更，則考慮建立一個新類別。
4. 根據這些答案，將 RTO 和 RPO 指派給工作負載。這可以直接完成，也可以透過將工作負載指派給預先定義的服務層來完成。

5. 在工作負載團隊和利害關係人可存取的位置記錄此工作負載的 災難復原計劃 (DRP)，這是貴組織業務持續性計劃 (BCP) 的一部分。
   a. 記錄 RTO 和 RPO，以及用來決定這些值的資訊。包括用於評估對業務之工作負載影響的策略。
   b. 記錄除 RTO 和 RPO 之外的其他指標，您是否正在追蹤或規劃追蹤災難復原目標。
   c. 建立 DR 策略和執行手冊的詳細資訊時，會將這些資訊新增至此計劃。
6. 藉由在如圖 15 所示的矩陣中查看工作負載的關鍵性，您可以開始建立針對組織定義的預先定義服務層。
7. 在您根據 實作了 DR 策略 (或 DR 策略的概念證明) 之後the section called “REL13-BP02 使用定義的復原策略來滿足復原目標”，請測試此策略以判定工作負載的實際 RTC (復原時間能力) 和 RPC (復原點能力)。如果這些不符合目標復原目標，則可與您的業務利害關係人合作，一起調整這些目標，或可對 DR 策略進行變更以符合目標。

主要問題

1. 在對業務產生嚴重影響之前，工作負載可以關閉的時間上限
   a. 如果工作負載中斷，請判定每分鐘對業務造成的貨幣成本 (直接財務影響)。
   b. 考慮到影響並非總是線性的。一開始影響可能會受到限制，然後在超過關鍵時間點後迅速增加。
2. 在對業務產生嚴重影響之前，可以遺失的資料量上限
   a. 考慮將此值用於您最關鍵的資料存放區。識別其他資料存放區的各自關鍵性。
   b. 如果遺失工作負載資料，可以重建嗎？ 如果在操作上這樣做比備份和還原更容易，則根據用來重建工作負載資料之來源資料的關鍵性來選擇 RPO。
3. 依賴下游游相依性的工作負載或依賴上游相依性的工作負載，其復原目標和可用性期望是什麼？
   a. 選擇可讓此工作負載符合上游相依性要求的復原目標
   b. 鑑於下游相依性的復原能力，選擇可實現的復原目標。可以執行非關鍵的下游相依性 (您可以「解決」的相依性)。或者，使用關鍵的下游相依性，在必要時改善其復原能力。

其他問題

考慮這些問題，以及它們如何套用至這個工作負載：

4. 您是否有不同的 RTO 和 RPO，取決於中斷的類型 (區域與可用區域等)？
5. 您的 RTO/RPO 是否會在特定時間 (季節性、銷售活動、產品發佈) 發生變化？ 若是，有什麼不同的測量和時間界限？
6. 如果工作負載中斷，有多少客戶會受到影響？

7. 如果工作負載中斷，對信譽有何影響？
8. 如果工作負載中斷，可能會發生哪些其他營運影響？ 例如，如果電子郵件系統無法使用，或如果薪資系統無法提交交易，則會影響員工的生產力。
9. 工作負載 RTO 和 RPO 如何與業務線和組織 DR 策略保持一致？
10. 是否有提供服務的內部合約義務？ 未符合它們時會受到處罰嗎？
11. 資料的法規或合規限制是什麼？

實作工作表

您可以將此工作表用於實作步驟 2 和 3。您可以調整此工作表以滿足您的特定需求，例如新增其他問題。

步驟 2：主要問題	是否適用於工作負載？	工作負載 RTO	工作負載 RPO	RTO 調整。	RPO 調整。	簡介
[1] 工作負載可以關閉的最長時間						以開始中斷到復原的時間進行測量
[2] 可以遺失的資料數量上限						以自從上次已知良好的可還原資料集後的時間進行測量
[3a] 上游相依性						輸入最嚴格的上游復原目標
[3b] 下游相依性						輸入最不嚴格的下游復原目標
[3a] 達成一致的上游相依性						如果上游值小於目前值，而下游值更大，則使用相依性來達成一致，
[3b] 達成一致的下游相依性						並在這裡輸入達成一致的值
[3] 相依性						請降低值以符合上游相依性，或根據下游相依性功能提高這些值
步驟 2：其他問題						指出問題是否適用。如果不適用，則略過它
基底 RTO/RPO						將上面的 RTO 和 RPO 值帶至這裡
[4] 中斷類型	[ ]Y / [ ]N					為具有最嚴格需求的事件類型輸入復原目標
[5] 特定時間型目標	[ ]Y / [ ]N					為具有最嚴格需求的時間輸入復原目標
[6] 顛覆客戶	[ ]Y / [ ]N					透過圖表以停機時間或資料遺失的函數表示受影響的客戶。 使用該函數，根據客戶影響輸入最大允許的 RTO 和 RPO
[7] 信譽影響	[ ]Y / [ ]N					與企業合作，根據對信譽的影響決定最大 RTO 和 RPO
[8] 營運影響	[ ]Y / [ ]N					根據營運影響輸入最大 RTO 和 RPO
[9] 組織遵循	[ ]Y / [ ]N					根據 LOB 和組織需求輸入此類型的最大工作負載 RTO 和 RPO
[10] 合約義務	[ ]Y / [ ]N					根據合約義務輸入最大 RTO 和 RPO
[11] 法規合規	[ ]Y / [ ]N					根據適用的法規合規輸入最大 RTO 和 RPO
以其他問題為基礎的目標						從 Q’s 4-11 取得並在這裡輸入最小值（更嚴格的值）
調整後的目標						如果無法滿足上行的目標，請與利害關係人合作放寬限制，並在這裡輸入新的最小值
調整後的 RTO/RPO						輸入基底 RPO/RTO 值或調整後的目標，以較低者為準
步驟 3						
對應至預先定義的類別或層級						向下調整這兩個值（更嚴格）以與最接近的定義層級一致

工作表

實作計劃的工作量： 低

資源

相關的最佳實務：

- the section called “REL09-BP04 定期執行資料復原以驗證備份的完整性和程序”
- the section called “REL13-BP02 使用定義的復原策略來滿足復原目標”
- the section called “REL13-BP03 測試災難復原實作以驗證實作”

相關文件：

- AWS 架構部落格：災難復原系列
- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)
- 使用 AWS Resilience Hub 管理彈性政策
- APN 合作夥伴：可以幫助災難復原的合作夥伴
- AWS Marketplace：可用於災難復原的產品

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
- AWS 上工作負載的災難復原

REL13-BP02 使用定義的復原策略來滿足復原目標

定義一個符合工作負載復原目標的災難復原 (DR) 策略。選擇如下策略：備份與還原；待命 (主動/被動)；或是主動/主動。

如果您的主要位置變成無法執行工作負載，則災難復原策略會依賴在復原站點中支持您工作負載的能力。最常見的復原目標為 RTO 和 RPO，其討論在 REL13-BP01 定義停機和資料遺失的復原目標。

單一 AWS 區域 內跨多個可用區域 (AZ) 的 DR 策略可以緩解火災、洪水和重大停電等災難事件。如果需要實作保護，以防範阻止您的工作負載在給定 AWS 區域 中執行且不太可能發生的事件，您可以使用一個使用多個區域的 DR 策略。

在跨多個區域架構 DR 策略時，您應該選擇下列其中一個策略。這些策略按成本和複雜度遞增的順序列出，以及按 RTO 和 RPO 的遞減順序列出。 復原區域 稱為 AWS 區域，而不是用於工作負載的主要區域。

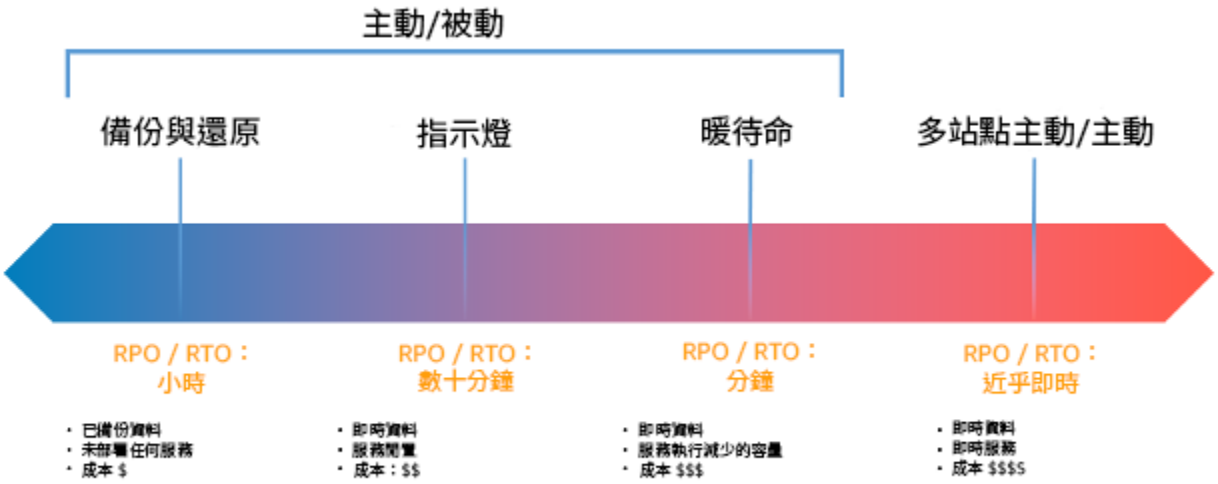


圖 17：災難復原 (DR) 策略

- 備份與恢復 (RPO 幾小時，RTO 24 小時以內)：將您的資料和應用程式備份至復原區域。使用自動或連續備份將啟用時間點復原，在某些情況下可以將 RPO 降低至 5 分鐘。如果發生災難，您將部署您的基礎設施 (使用基礎設施架構即程式碼來減少 RTO)、部署您的程式碼，並還原備份的資料以從復原區域中的災難中復原。
- 指示燈 (RPO 幾分鐘，RTO 幾十分鐘)：在復原區域中佈建核心工作負載基礎設施的副本。將您的資料複寫到復原區域並在該處建立其備份。支援資料複寫和備份所需的資源 (例如資料庫和物件儲存) 始終處於開啟狀態。其他元素 (例如應用程式伺服器或無伺服器運算) 未部署，但可在需要時使用必要的組態和應用程式碼建立。
- 暖待命 (RPO 幾秒鐘，RTO 幾分鐘)：維持工作負載的縮減但完整功能版本，該工作負載始終在復原區域中執行。業務關鍵系統會完全複製且持續開啟，但叢集會縮小。資料會被複寫並存在於復原區域中。當需要復原時，系統會迅速擴展以處理生產負載。熱待命的縱向擴增越多，對 RTO 和控制平面的依賴就越低。完全擴展時，稱之為 熱待命。
- 多區域 (多站點) 主動-主動 (RPO 近乎零，RTO 可能為零) 您的工作負載會部署至多個 AWS 區域，並主動處理來自多個 AWS 區域 的流量。此策略需要您跨區域同步資料。必須避免或處理在兩個不同區域複本中寫入同一記錄所引起的可能衝突，這可能很複雜。資料複寫對於資料同步很有用，而且可以保護您防範某些類型的災難，但它不能保護您防範資料損毀或破壞，除非您的解決方案也包括時間點復原的選項。

> **Note**
>
> 指示燈和暖待命之間的差異有時可能很難理解。這兩者都在您的復原區域中包含一個環境，其中具有主要區域資產的副本。區別在於，若未先採取額外動作，指示燈無法處理請求，而暖待命可以立即處理流量 (容量層級降低)。指示燈將需要您開啟伺服器，可能會部署額外 (非核心) 基礎設施並縱向擴展，而暖待命只需要您縱向擴展 (一切都已部署並執行中)。根據您的 RTO 和 RPO 需求在這兩者之間進行選擇。

預期成果：

對於每個工作負載，都有一個已定義和實作的 DR 策略，讓該工作負載能夠實現 DR 目標。工作負載之間的 DR 策略會利用可重複使用模式 (例如上述策略)。

常用的反模式：

- 針對具有類似 DR 目標的工作負載實作不一致的復原程序。
- 災難發生時臨時實作 DR 策略。
- 沒有 DR 計劃。
- 復原期間依賴控制平面操作。

建立此最佳實務的優勢：

- 使用定義的復原策略可讓您使用常用的工具和測試程序。
- 使用定義的復原策略可讓您更有效地在團隊之間分享知識，並更輕鬆地在他們擁有的工作負載上實作 DR。

若未建立此最佳實務，暴露的風險等級為： 高

- 若沒有事先規劃、實作和測試災難復原策略，您就不可能在發生災難時實現復原目標。

實作指引

對於其中每一個步驟，請參閱下列詳細資訊。

1. 確定將滿足此工作負載之復原要求的 DR 策略。
2. 審查如何實作所選 DR 策略的模式。

3. 評估工作負載的資源，以及在容錯移轉之前 (在正常操作期間) 其哪個組態將位於復原區域中。
4. 確定並實作如何在需要時 (在災難事件期間) 使您的復原區域為容錯移轉做好準備。
5. 確定並實作如何在需要時 (在災難事件期間) 將流量路由至容錯移轉。
6. 設計您的工作負載將如何復原的計劃。

實作步驟

1. 確定將滿足此工作負載之復原要求的 DR 策略。

選擇 DR 策略是在減少停機時間和資料遺失 (RTO 和 RPO) 與實作策略的成本和復雜性之間進行取捨。您應該避免實作比其所需更嚴格的策略，因為這會產生不必要的成本。

例如，在下圖中，企業已確定其最大允許的 RTO 以及其可以在服務還原策略上花費的限制。鑑於業務目標，DR 策略指示燈或暖待命將同時滿足 RTO 和成本準則。

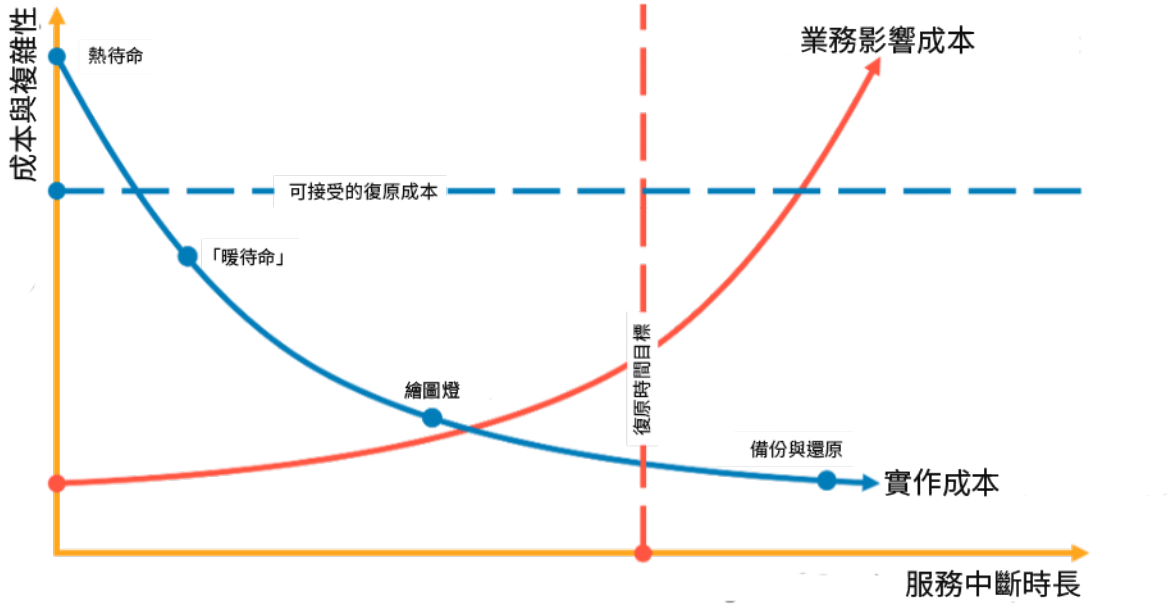


圖 18：根據 RTO 和成本選擇 DR 策略

若要深入了解，請參閱 業務持續性計劃 (BCP)。

2. 審查如何實作所選 DR 策略的模式。

此步驟在於了解您將如何實作所選策略。使用 AWS 區域 做為主要站點和復原站點來解釋這些策略。不過，您也可以選擇使用單一區域內的可用區域，做為您的 DR 策略，這會利用其中多個策略的元素。

在此步驟之後的後續步驟中，您會將此策略套用至您的特定工作負載。

備份與恢復

備份與恢復 是最不複雜的實作策略，但需要更多時間和精力來還原工作負載，從而導致更高的 RTO 和 RPO。始終對資料進行備份並將其複製到另一個站點 (例如另一個 AWS 區域) 是一種很好的做法。

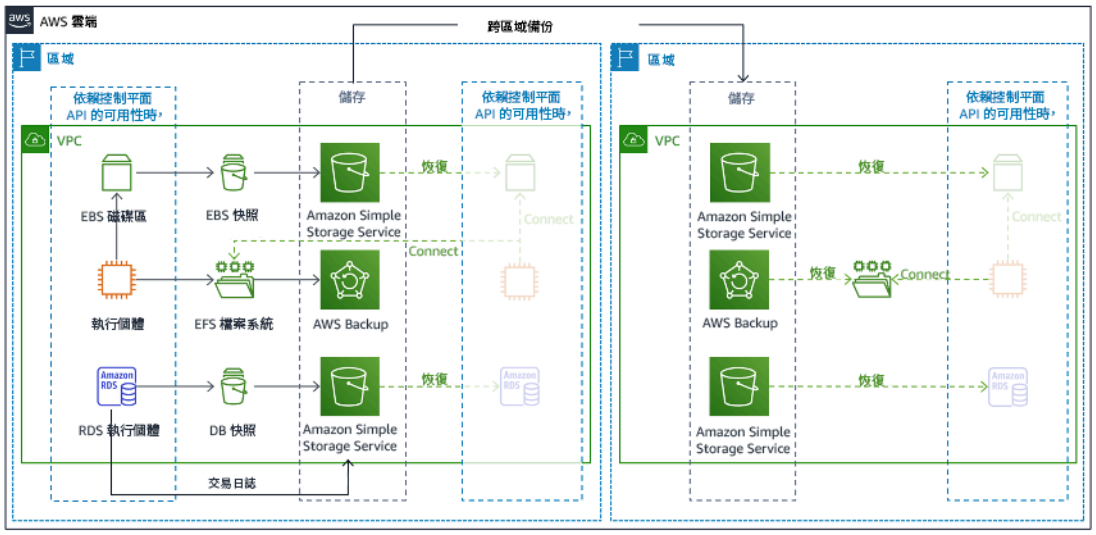


圖 19：備份和還原架構

如需此策略的詳細資訊，請參閱 AWS 上的災難復原 (DR) 架構，第 II 部分：具有快速復原的備份和還原。

指示燈

使用 指示燈 方法，您可以將資料從主要區域複寫至復原區域。用於工作負載基礎設施的核心資源會部署在復原區域中，不過，仍需要額外的資源和任何相依性，才能使其成為功能堆疊。例如，在圖 20 中，未部署任何運算執行個體。

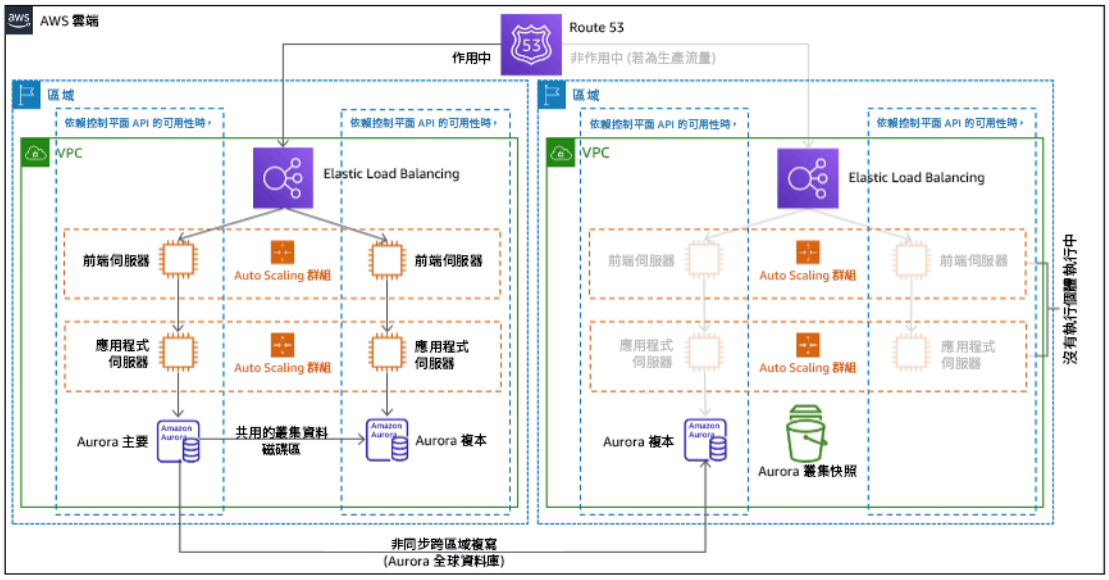


圖 20：指示燈架構

如需此策略的詳細資訊，請參閱 AWS 上的災難復原 (DR) 架構，第 III 部分：指示燈和暖待命。

暖待命

AWS Well-Architected 基礎架構 方法涉及確保在另一個區域中有一個縮減規模，但功能完全的生產環境副本。這種方法擴充了指示燈概念並減少了復原時間，因為您的工作負載始終在另一個區域中開啟。如果部署完整容量的復原區域，這稱為 熱待命。

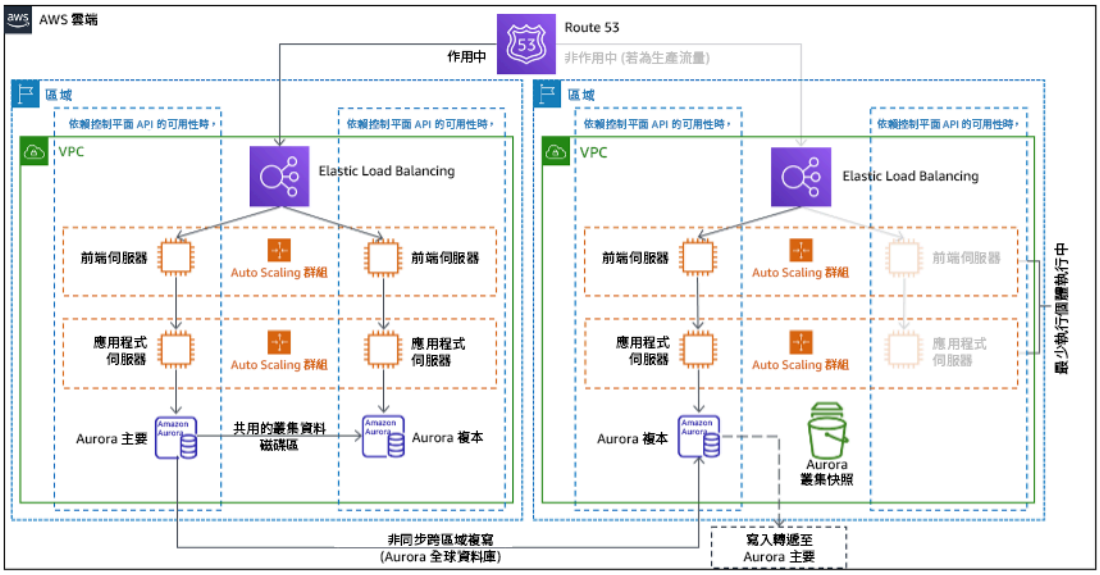


圖 21：暖待命架構

使用暖待命或指示燈需要縱向擴展復原區域中的資源。若要確保需要時容量可用，請考慮針對 EC2 執行個體 使用容量保留。如果使用 AWS Lambda，則 佈建的並行 可以確保執行環境，以便它們準備好立即回應您的函數叫用。

如需此策略的詳細資訊，請參閱 AWS 上的災難復原 (DR) 架構，第 III 部分：指示燈和暖待命。

多站點主動/主動

您可以同時在多個區域中執行工作負載，做為 多站點主動/主動 策略。多站點主動/主動會為來自其部署至的所有區域的流量提供服務。基於 DR 以外的理由，客戶可能會選取此策略。它可以用來提高可用性，或在將工作負載部署至全球對象 (使端點更靠近使用者和/或將本地化的堆疊部署到該區域的對象) 時使用它。作為 DR 策略，如果工作負載無法在其部署至的其中一個 AWS 區域中得到支援，則會撤離該區域，而其餘區域則會用來維護可用性。多站點主動/主動是災難復原策略中操作最複雜的策略，因此只有在業務要求有此需要時才應選取它。

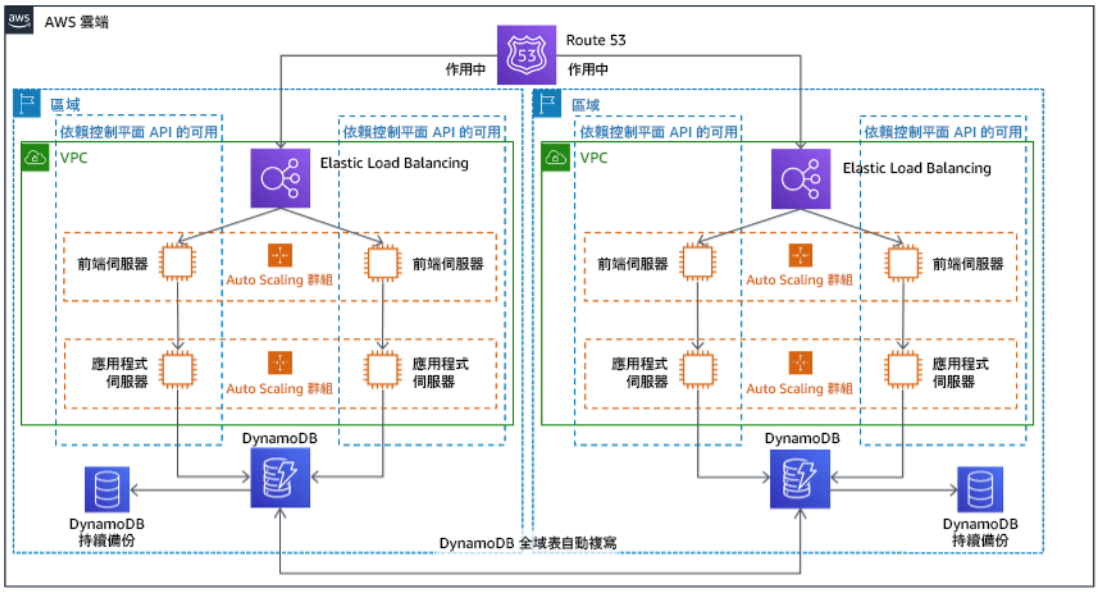


圖 22：多站點主動/主動架構

如需此策略的詳細資訊，請參閱 AWS 上的災難復原 (DR) 架構，第 IV 部分：多站點主動/主動。

其他保護資料的做法

使用所有策略時，您還必須緩解資料災難。持續資料複寫可以保護您防範某些類型的災難，但它不能保護您防範資料損毀或破壞，除非您的策略也包括所存放資料的版本控制，或時間點復原的選項。除了複本之外，您還必須備份復原站點中的複寫資料，以建立時間點備份。

在單一 AWS 區域 內使用多個可用區域 (AZ)

在單一區域內使用多個 AZ 時，您的 DR 實作會使用上述策略的多個元素。首先，您必須建立高可用性 (HA) 架構，使用多個 AZ，如圖 23 所示。此架構會利用多站點主動/主動方法，因為 Amazon EC2 執行個體 和 Elastic Load Balancer 已在多個 AZ 中部署資源，主動處理請求。架構也會示範熱待命，其中如果主要 Amazon RDS 執行個體失敗 (或 AZ 本身失敗)，則待命執行個體會提升至主要執行個體。

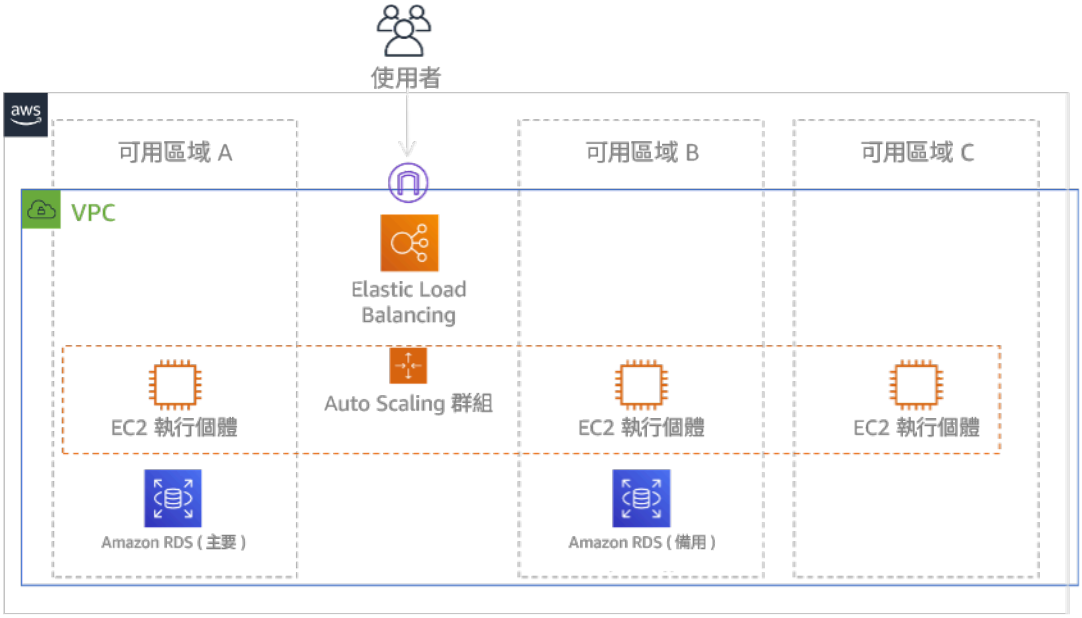


圖 23：異地同步備份架構

除了這種 HA 架構之外，您還需要新增執行工作負載所需之所有資料的備份。這對於受制於單一區域的資料尤其重要，例如 Amazon EBS 磁碟區 或 Amazon Redshift 叢集。如果 AZ 失敗，您需要將此資料還原至另一個 AZ。可能的話，您也應該將資料備份複製到另一個 AWS 區域，做為額外的保護層。

下列部落格文章中描述了一種不太常見的單一區域替代方法 (異地同步備份 DR)： 使用 Amazon Route 53 應用程式復原控制器建置高彈性應用程式，第 1 部分：單一區域堆疊。在這裡，策略是盡可能地在 AZ 之間保持隔離，就像區域的操作方式一樣。使用這種替代策略，您可以選擇主動/主動或主動/被動方法。

注意：某些工作負載具有法規資料落地要求。如果在目前只有一個 AWS 區域的區域性中，這適用於您的工作負載，則多區域將不適合您的業務需求。異地同步備份策略提供良好的保護，可防範大部分災難。

3. 評估工作負載的資源，以及在容錯移轉之前 (在正常操作期間) 其哪個組態將位於復原區域中。

對於基礎設施和 AWS 資源，使用基礎設施即程式碼，例如 AWS CloudFormation ，或使用第三方工具，例如 Hashicorp Terraform。若要使用單一作業跨多個帳戶和區域進行部署，您可以使用 AWS CloudFormation StackSets。對於多站點主動/主動和熱待命策略，您的復原區域中部署的基礎設施具

有與您主要區域相同的資源。對於指示燈和暖待命策略，部署的基礎設施將需要額外的動作，才能為生產做好準備。使用 CloudFormation 參數 和 條件式邏輯，您可以使用單一範本控制部署的堆疊是主動還是待命。這類 CloudFormation 範本的範例包含在 這篇部落格文章。

所有 DR 策略都要求在 AWS 區域內備份資料來源，然後將這些備份複製到復原區域。AWS Backup 提供了一個集中檢視，您可以在其中設定、排定和監控這些資源的備份。對於指示燈、暖待命和多站點主動/主動，您還應該將資料從主要區域複寫到復原區域中的資料資源，例如 Amazon Relational Database Service (Amazon RDS) 資料庫執行個體或 Amazon DynamoDB 資料表。因此，這些資料資源是即時的，而且可以為復原區域中的請求提供服務。

若要深入了解 AWS 服務如何跨區域操作，請參閱此部落格系列，其位於 使用 AWS Services 建立多區域應用程式。

4. 確定並實作如何在需要時 (在災難事件期間) 使您的復原區域為容錯移轉做好準備。

對於多站點主動/主動，容錯移轉意味著撤離一個區域，並依賴剩餘的主 動區域。通常，這些區域已準備好接受流量。對於指示燈和暖待命策略，您的復原動作將需要部署遺漏的資源，例如圖 20 中的 EC2 執行個體，以及任何其他遺漏的資源。

對於上述所有策略，您可能需要提升資料庫的唯讀執行個體，以變成主要讀取/寫入執行個體。

對於備份和還原，從備份還原資料會為該資料建立資源，例如 EBS 磁碟區、RDS 資料庫執行個體和 DynamoDB 資料表。您也需要還原基礎設施和部署程式碼。您可以使用 AWS Backup，還原復原區域中的資料。請參閱 REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料 以取得詳細資訊。重建基礎設施包括建立 EC2 執行個體之類的資源，還有 Amazon Virtual Private Cloud (Amazon VPC))、子網路，以及所需的安全群組。您可以將大部分還原程序自動化。若要了解做法，請參閱 這篇部落格文章。

5. 確定並實作如何在需要時 (在災難事件期間) 將流量路由至容錯移轉。

此容錯移轉作業可以自動或手動啟動。應謹慎使用根據運作狀態檢查或警示自動啟動的容錯移轉，因為不必要的容錯移轉 (誤報) 會產生非可用性和資料遺失等成本。因此通常使用手動啟動的容錯移轉。在此情況下，您仍應將容錯移轉的步驟自動化，讓手動啟動就像按下按鈕一樣簡易。

使用 AWS 服務時，有數個流量管理選項需要考慮。其中一個選項是使用 Amazon Route 53。使用 Amazon Route 53，您可以將一個或多個 AWS 區域中的 IP 端節與一個 Route 53 網域名稱建立關聯。若要實作手動啟動的容錯移轉，您可以使用 Amazon Route 53 應用程式復原控制器，其會提供一個高度可用的資料平面 API，將流量重新路由到復原區域。實作容錯移轉時，使用資料平面作業並避免控制平面作業，其描述在 REL11-BP04 復原期間需使用資料平面，而非控制平面。

若要深入了解這個和其他選項，請參閱 災難復原白皮書的這一節。

6. 設計您的工作負載將如何復原的計劃。

復原是指在災難事件減弱後將工作負載操作回復到主要區域。將基礎設施和程式碼佈建到主要區域通常遵循最初使用的相同步驟，依賴基礎設施即程式碼和程式碼部署管道。復原的挑戰是還原資料存放區，並確保它們與操作中的復原區域保持一致。

在容錯移轉狀態下，復原區域中的資料庫是即時的並具有最新資料。後續目標是從復原區域重新同步到主要區域，確保它是最新的。

有些 AWS 服務將會自動執行此動作。如果使用 Amazon DynamoDB 全域資料表，即使主要區域中的資料表變成無法可用，則當它重新上線時，DynamoDB 仍會繼續傳播任何擱置的寫入。如果使用 Amazon Aurora 全球資料庫 和使用 受管規劃容錯移轉，則會維護 Aurora 全球資料庫的現有複寫拓撲。因此，主要區域中先前的讀取/寫入執行個體將成為複本，並從復原區域中接收更新。

如果這不是自動的，您將需要在主要區域中重建資料庫，做為復原區域中資料庫的複本。在許多情況下，這將涉及刪除舊的主要資料庫並建立新的複本。例如，如需如何在假設非計劃容錯移轉的情況下使用 Amazon Aurora 全球資料庫 執行此動作的指示，請參閱此實驗室： 復原全球資料庫。

容錯移轉後，如果您可以繼續在復原區域中執行，請考慮使其成為新的主要區域。您仍會執行上述所有步驟，使先前的主要區域成為復原區域。有些組織會執行排程輪換，定期 (例如每三個月) 交換其主要區域和復原區域。

容錯移轉和復原所需的所有步驟都應保持在可供所有團隊成員使用的程序手冊中，並定期進行審查。

實作計劃的工作量：高

資源

相關的最佳實務：

- the section called “REL09-BP01 識別並備份所有需要備份的資料，或從來源複製資料”
- the section called “REL11-BP04 復原期間需使用資料平面，而非控制平面”
- the section called “REL13-BP01 定義停機和資料遺失的復原目標”

相關文件：

- AWS 架構部落格：災難復原系列

- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)
- 雲端中的災難復原選項
- 一小時建置無伺服器的多區域、主動-主動後端解決方案
- 多區域無伺服器後端 - 重新載入
- RDS：跨區域複寫僅供讀取複本
- Route 53：設定 DNS 備援
- S3：跨區域複寫
- 什麼是 AWS Backup？
- 什麼是 Route 53 應用程式復原控制器？
- AWS 彈性災難復原
- HashiCorp Terraform：入門 - AWS
- APN 合作夥伴：可以幫助災難復原的合作夥伴
- AWS Marketplace：可用於災難復原的產品

相關影片：

- AWS 上工作負載的災難復原
- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
- AWS 彈性災難復原入門 | Amazon Web Services

相關範例：

- AWS Well-Architected 實驗室 - 災難復原 - 說明 DR 系列的研討會系列

REL13-BP03 測試災難復原實作以驗證實作

定期測試容錯移轉到您的復原站點以確保正常操作，並符合 RTO 和 RPO。

要避免的模式是：開發鮮少執行的復原路徑。例如，您可能有一個次要資料存放區，只供唯讀查詢之用。當您寫入資料存放區而主資料存放區發生故障時，您可能需要容錯移轉到次要資料存放區。如果您不經常測試此容錯移轉，則可能會發現您對次要資料存放區的功能的假設不正確。次要資料存放區的容量 (在您上次測試時可能已經足夠) 在這種情況下可能無法再容忍負載。我們的經驗顯示，唯一能發揮功用的錯誤復原，是您經常測試的路徑。因此，最好擁有少量的復原路徑。您可建立復原模式，並定期

進行測試。若擁有複雜或關鍵復原路徑，您還是需要定期在生產環境中執行該故障，說服自己該復原路徑能發揮功用。在我們剛剛討論的範例中，無論是否需要，您都應定期容錯移轉到備用資料庫。

常用的反模式：

- 切勿在生產環境中執行容錯移轉。

建立此最佳實務的優勢： 定期測試您的災難復原計畫，可確保該計畫能在需要時運作，也能讓您的團隊知道如何執行策略。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

- 為復原設計您的工作負載。定期測試您的復原路徑：復原導向運算 (ROC) 可識別系統中能增強復原能力的特性。這些特性包括：隔離和冗餘，系統範圍內的回復變更能力，監控和確定運行狀態的能力，提供診斷、自動復原和模組化設計的能力，以及重新啟動的能力。練習復原路徑，以確保您可以在指定時間內完成復原到指定狀態。在復原過程中使用您的執行手冊，以記錄問題並在下一次測試前找出其解決方案。
  - 柏克萊加州大學/史丹佛大學復原導向的運算專案
- 使用 CloudEndure Disaster Recovery 實作和測試您的 DR 策略。
  - 透過 CloudEndure 測試災難復原解決方案
  - CloudEndure Disaster Recovery
  - AWS 的 CloudEndure Disaster Recovery

資源

相關文件：

- APN 合作夥伴：可以幫助災難復原的合作夥伴
- AWS 架構部落格：災難復原系列
- AWS Marketplace：可用於災難復原的產品
- CloudEndure Disaster Recovery
- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)
- 透過 CloudEndure 測試災難復原解決方案
- 柏克萊加州大學/史丹佛大學復原導向的運算專案

- 什麼是 AWS Fault Injection Simulator？

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)
- AWS re:Invent 2019：使用 AWS 的備份與還原和災難復原解決方案 (STG208)

相關範例：

- AWS Well-Architected 實驗室 - 測試彈性

REL13-BP04 管理 DR 站點或區域的組態偏移

確保根據需要在 DR 站點或區域提供基礎設施、資料和組態。例如，檢查 AMI 和服務配額是否為最新版本。

AWS Config 會持續監控和記錄 AWS 資源組態。它可以偵測偏移，並觸發 AWS Systems Manager Automation 修正並引發警示。AWS CloudFormation 可額外偵測您已部署之堆疊中的偏移。

常用的反模式：

- 當您在主要位置進行組態或基礎設施變更時，無法在復原位置進行更新。
- 未考量主要和復原位置中潛在的限制 (例如服務差異)。

建立此最佳實務的優勢： 確保 DR 環境與現有環境一致，便可保證完整復原。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 確保您的交付管道同時交付到主要站點和備份站點。用於將應用程式部署到生產中的交付管道，應分發到所有指定的災難復原策略位置，包括開發和測試環境。
- 啟用 AWS Config 追蹤潛在的偏移位置。使用 AWS Config 規則建立系統，以執行災難復原策略，並在發現偏移時產生提醒。
  - 依 AWS Config 規則 修補不合規的 AWS 資源
  - AWS Systems Manager Automation
- 使用 AWS CloudFormation 偵測您的基礎設施。AWS CloudFormation 可以偵測 CloudFormation 範本指定項目與實際部署項目之間的偏移。

- AWS CloudFormation：在整個 CloudFormation 堆疊上偵測偏移

資源

相關文件：

- APN 合作夥伴：可以幫助災難復原的合作夥伴
- AWS 架構部落格：災難復原系列
- AWS CloudFormation：在整個 CloudFormation 堆疊上偵測偏移
- AWS Marketplace：可用於災難復原的產品
- AWS Systems Manager Automation
- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)
- 如何在 AWS 上實作基礎設施組態管理解決方案？
- 依 AWS Config 規則 修補不合規的 AWS 資源

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)

REL13-BP05 自動化復原

使用 AWS 或第三方工具自動化系統復原，並將流量路由到 DR 站點或區域。

根據設定的運作狀態檢查，Elastic Load Balancing 和 AWS Auto Scaling 等 AWS 服務可將負載分散到運作狀態良好的可用區域，而 Amazon Route 5、AWS 和 Global Accelerator 等服務則可將負載路由到運作狀態良好的 AWS 區域。Amazon Route 53 應用程式復原控制器可協助您使用準備度檢查和路由控制功能，來管理和協調容錯移轉。這些功能會持續監控應用程式從失敗中復原的功能，以便您跨多個 AWS 區域、可用區域和內部部署來控制應用程式復原。

對於現有實體或虛擬資料中心或私有雲端上的工作負載， AWS 彈性災難復原(可透過 AWS Marketplace 取得) 可讓組織設定 AWS 的自動化災難復原策略。CloudEndure 也支援 AWS 中的跨區域/跨可用區域災難復原。

常用的反模式：

- 實作相同的自動化容錯移轉和容錯回復會在失敗發生時導致翻動。

建立此最佳實務的優勢： 自動化復原可以消除手動錯誤的機會，減少您的復原時間。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

- 自動化復原路徑。若復原時間較短，則人為判斷和行動無法用於可用性高的方案。系統應在每種情況下都能自動復原。
  - 使用 CloudEndure Disaster Recovery 進行自動化容錯移轉和容錯回復：CloudEndure Disaster Recovery 會持續將您的機器 (包括作業系統、系統狀態組態、資料庫、應用程式和檔案) 複寫至您的目標 AWS 帳戶和慣用區域中的低成本階段區域。發生災難時，您可以指示 CloudEndure Disaster Recovery 在數分鐘內自動啟動處於完全佈建狀態的數千部機器。
    - 執行災難復原容錯移轉和退回
    - CloudEndure Disaster Recovery

資源

相關文件：

- APN 合作夥伴：可以幫助災難復原的合作夥伴
- AWS 架構部落格：災難復原系列
- AWS Marketplace：可用於災難復原的產品
- AWS Systems Manager Automation
- AWS 的 CloudEndure Disaster Recovery
- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)

相關影片：

- AWS re:Invent 2018：適用於多區域主動-主動應用程式的架構模式 (ARC209-R2)

效能達成效率

主題

- 選擇
- 檢閱

- 監控
- 權衡

選擇

問題

- PERF 1 您如何選擇效能最佳的架構？
- PERF 2 您如何選取運算解決方案？
- PERF 3 您如何選取儲存解決方案？
- PERF 4 您如何選擇資料庫解決方案？
- PERF 5 您如何設定聯網解決方案？

PERF 1 您如何選擇效能最佳的架構？

欲讓工作負載達到最佳效能通常需要採用多種方法。Well-Architected 系統會使用多重解決方案和功能以提升效能。

最佳實務

- PERF01-BP01 了解可用的服務和資源
- PERF01-BP02 定義架構選擇程序
- PERF01-BP03 將成本需求因素納入決策
- PERF01-BP04 使用政策或參考架構
- PERF01-BP05 使用雲端供應商或適當的合作夥伴提供的指導
- PERF01-BP06 對現有工作負載進行基準化分析
- PERF01-BP07 對工作負載執行負載測試

PERF01-BP01 了解可用的服務和資源

了解並熟悉雲端中可用的廣泛服務和資源。確定與工作負載相關的服務和組態選項，並了解如何獲得最佳效能。

如果您要評估現有的工作負載，則必須針對其使用的各種服務資源產生監視清單。您的監視清單可協助您評估哪些元件可以被受管服務和較新的技術取代。

常用的反模式：

- 您可以使用雲端做為並置資料中心。
- 您可以使用共用儲存來處理所有需要持久性儲存的物件。
- 您不使用自動調整規模功能。
- 您使用的執行個體類型與目前標準最相符，但大於需求。
- 您會部署和管理可做為受管服務的技術。

建立此最佳實務的優勢： 考量採用可能不熟悉的服務，可以大幅降低基礎設施成本，以及維護服務所需的工作量。您可以透過部署新的服務和功能來加速交期。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

清查工作負載軟體和架構以存放相關服務：收集工作負載的庫存，並決定要進一步了解哪些產品類別。識別可被受管服務替換的工作負載元件，以提高效能並降低操作複雜性。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP02 定義架構選擇程序

使用內部經驗和雲端知識，或外部資源 (例如已發佈的使用案例、相關文件或白皮書)，定義選擇資源和服務的程序。您定義的程序應該鼓勵對可在工作負載中使用的服務進行實驗和基準化分析。

在為架構編寫關鍵使用者案例時，應包括效能要求，例如指定每個關鍵案例應執行的速度。對於這些關鍵案例，您應實作額外執行指令碼的使用者旅程，以確保您可以直觀地了解這些案例會如何根據您的要求予以執行。

常用的反模式：

- 您假設您目前的架構將變成靜態，且一段時間不會更新。
- 您會隨時間導入架構變更，而且無需理由佐證。

建立此最佳實務的優勢： 建立進行架構變更的定義程序後，即可啟用收集的資料，以隨著時間影響工作負載。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

選擇一種架構方法：確定可以滿足效能需求的架構類型。識別限制，例如交付媒體 (桌面、Web、移動、IoT)、遺留需求和整合。識別重複使用的機會，包括重構。諮詢其他團隊、架構圖解和資源，例如 AWS 解決方案架構師、AWS 參考架構和 AWS 合作夥伴，以幫助您選擇架構。

定義效能需求：使用客戶體驗來確定最重要的指標。對於每個指標，確定目標、測量方法和優先級。定義客戶體驗。記錄客戶所需的效能體驗，包括客戶如何評價工作負載的效能。優先考慮關鍵使用者案例的體驗問題。納入效能需求並實作執行指令碼的使用者之旅，以確保您了解這些案例會如何根據您的要求予以執行。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP03 將成本需求因素納入決策

工作負載通常具有營運的成本需求。使用內部成本控制，根據預測的資源需求選取資源類型和大小。

判斷哪些工作負載元件可以被全受管服務 (例如受管資料庫、記憶體內快取和 ETL 服務) 取代。降低營運工作負載可讓您將資源投注在業務成果上。

如需成本需求最佳實務，請參閱 成本優化支柱白皮書中 經濟實惠的 資源章節。

常用的反模式：

- 您只能使用一個執行個體系列。
- 您不會評估授權解決方案與開放原始碼解決方案
- 您只能使用區塊儲存。
- 您可以在 EC2 執行個體和可做為受管服務使用的 Amazon EBS 或暫時性磁碟區上部署常見的軟體。

建立此最佳實務的優勢： 做出選擇時考慮成本，可讓您促成其他投資。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

將工作負載元件最佳化以降低成本：調整工作負載元件的大小並啟用彈性功能，以降低成本並最大化元件效率。判斷哪些工作負載元件可在適當時由受管服務 (例如受管資料庫、記憶體內快取和反向代理) 取代。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心
- AWS Compute Optimizer

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture
- 優化 AWS 運算的效能和成本 (CMP323-R1)

相關範例：

- AWS 範例
- AWS SDK 範例
- 在 Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小
- AWS Compute Optimizer 示範程式碼

PERF01-BP04 使用政策或參考架構

透過評估內部政策和現有參考架構，並使用分析來選取工作負載的服務和組態，來將效能和效率提升至最大。

常用的反模式：

- 您可以廣泛使用可能影響公司管理開銷的技術選擇。

建立此最佳實務的優勢： 建立架構、技術和供應商選擇的政策可讓您快速做出決策。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

使用現有的政策或參考架構來部署工作負載：將服務整合到您的雲端部署，然後使用效能測試以確保您可以繼續滿足效能需求。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP05 使用雲端供應商或適當的合作夥伴提供的指導

使用解決方案架構師、專業服務或適當的合作夥伴等雲端公司資源，來引導您做出決策。這些資源可協助檢閱和改善架構，以實現最佳效能。

當您需要其他指導或產品資訊時，請聯絡 AWS 尋求協助。AWS 解決方案架構師和 AWS 專業服務 會為實作解決方案提供指導 AWS 合作夥伴 會提供 AWS 專業知識，協助您提升業務的靈活性和創新性。

常用的反模式：

- 您使用 AWS 做為通用資料中心供應商。
- 您以非設計宗旨的方式使用 AWS 服務。

建立此最佳實務的優勢： 與您的供應商或合作夥伴協商，可讓您對自己的決策更有信心。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

聯絡 AWS 資源以獲得協助：AWS 解決方案架構師和專業服務會為實作解決方案提供指導。APN 合作夥伴提供 AWS 專業知識，協助您提升業務的靈活性和創新性。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP06 對現有工作負載進行基準化分析

對現有工作負載的效能進行基準化分析，以了解工作負載在雲端的效能。使用從基準化分析中收集的資料，來推動架構決策。

使用基準化分析搭配綜合測試和實際使用者監控，以產生與工作負載元件效能相關的資料。與負載測試相比，基準化分析通常速度更快；要評估特定元件的技術時，會使用基準化分析。當您缺少執行負載測試的完整解決方案時，通常可在新專案開始時使用基準化分析。

您可以建置自己的自訂基準化分析測試，也可以使用產業標準測試，例如 TPC-DS ，來對資料倉儲工作負載進行基準化分析。比較環境時，產業基準化分析很有幫助。對於確定您希望在架構中進行的特定營運類型，自訂基準化分析非常實用。

基準化分析時，務必要預熱測試環境，以確保獲得有效的結果。多次執行相同的基準化分析，以確保您已擷取到隨時間推移出現的任何變化。

由於基準化分析的速度通常比負載測試要快，因此可以在部署管道中盡早使用基準化分析，以便能更快提供有關效能偏差的回饋。當您評估元件或服務中的重大變更時，藉助基準化分析，您可以更快速地查看所做的變更是否合理。請務必使用基準化分析搭配負載測試，因為負載測試將告訴您工作負載在生產中的效能。

常用的反模式：

- 您倚賴不表現工作負載特性的常見基準測試。
- 您依賴客戶意見回饋和感受做為唯一的基準測試。

建立此最佳實務的優勢： 對目前的實作進行基準化分析，可讓您測量效能改善。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

在開發過程中監控效能：實作可隨著工作負載的演進而提供效能可見度的程序。

整合到交付管道：在您的交付管道中自動執行負載測試。將測試結果與預先定義的關鍵績效指標 (KPI) 和閾值進行比較，以確保您能持續符合效能需求。

測試使用者之旅：使用生產資料的綜合或處理過的版本 (移除敏感或身份資訊) 進行負載測試。透過在整個應用程式中使用重新執行或預先程式化的使用者之旅來測試整個架構。

實際使用者監控：使用 CloudWatch RUM 來協助您收集和檢視有關應用程式效能的用戶端資料。使用此資料可協助建立您的實際使用者效能基準測試。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is my Architecture
- 透過 Amazon CloudWatch RUM 優化應用程式
- Amazon CloudWatch Synthetics 的示範

相關範例：

- AWS 範例
- AWS SDK 範例
- 分散式負載測試
- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端

PERF01-BP07 對工作負載執行負載測試

使用不同類型和大小的資源，在雲端部署最新的工作負載架構。監控部署，以擷取可識別瓶頸或過多容量的效能指標。使用此效能資訊，來設計或改善您的架構和資源選擇。

負載測試會使用 實際 工作負載，以便您查看解決方案在生產環境中的效能。必須使用生產資料的綜合或處理過的版本 (刪除敏感或可識別身分的資訊) 執行負載測試。在工作負載中大規模使用重播或預先程式化的使用者旅程，以遍歷整個架構。在交付管道中自動執行負載測試，並將結果與預先定義的 KPI 和閾值進行比較。這可確保您持續達到所需的效能。

常用的反模式：

- 您載入測試工作負載的個別部分，而非整個工作負載。
- 您在與生產環境不同的基礎設施上載入測試。
- 您只對預期的 (而非超標) 負載進行負載測試，以協助預測未來可能發生問題的位置。
- 在不通知 AWS Support 的情況下執行負載測試，並以類似拒絕服務事件的形式來擊敗測試。

建立此最佳實務的優勢： 在負載測試下測量效能時，會顯示負載增加時會受到影響的位置。這可在變更影響工作負載之前，讓您先預測所需的變更。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

透過負載測試驗證方法：載入測試概念驗證，以確認是否符合效能需求。您可以使用 AWS 服務執行生產規模的環境，進而測試您的架構。由於僅在需要時才為測試環境付費，因此您只需花費使用內部部署環境的一小部分成本，就可以執行全面測試。

監控指標：Amazon CloudWatch 可以收集架構中各種資源的指標。您還可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方解決方案來設定可指出何時超過閾值的警示。

大規模測試：負載測試會使用實際工作負載，因此您可以查看解決方案在生產環境中的效能。您可以使用 AWS 服務執行生產規模的環境，進而測試您的架構。由於僅在需要時為測試環境付費，因此與使用內部部署環境相比，可以更低的成本執行全面測試。利用 AWS 雲端測試您的工作負載，以發現無法擴展的地方或是否以非線性方式擴展。例如，使用 Spot 執行個體以低成本產生負載，並在生產中遇到瓶頸之前發現瓶頸。

資源

相關文件：

- AWS CloudFormation
- 使用 CloudFormer 建置 AWS CloudFormation 範本
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分散式負載測試

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- 透過 Amazon CloudWatch RUM 優化應用程式
- Amazon CloudWatch Synthetics 的示範

相關範例：

- AWS 上的分散式負載測試

PERF 2 您如何選取運算解決方案？

工作負載的最佳運算解決方案會根據應用程式設計、使用模式和組態設定而有所不同。架構可針對不同元件使用不同運算解決方案並啟用不同功能，以提升效能。為架構選錯運算解決方案，可能使效能達成效率降低。

最佳實務

- PERF02-BP01 評估可用的運算選項
- PERF02-BP02 了解可用的運算組態選項
- PERF02-BP03 收集與運算相關的指標
- PERF02-BP04 透過適當調整大小來確定所需的組態
- PERF02-BP05 利用資源的可用彈性
- PERF02-BP06 根據指標重新評估運算需求

PERF02-BP01 評估可用的運算選項

了解您的工作負載如何受益於使用不同的運算選項，例如執行個體、容器和函數。

預期成果： 藉由了解所有可用的運算選項，您將注意到提高效能、減少不必要的基礎設施成本，以及降低減少維護工作負載所需操作工作量的機會。如果部署新服務和功能，也可以加速交期。

常用的反模式：

- 在遷移後工作負載中，使用正用於內部部署的同一運算解決方案。
- 缺乏對雲端運算解決方案以及這些解決方案如何提高您運算效能的感認。
- 當替代運算解決方案更精確地符合您的工作負載特性時，現有運算解決方案會變得更大以符合擴展或效能要求。

建立此最佳實務的優勢： 藉由識別運算要求和評估可用的運算解決方案，業務利害關係人和工程團隊將了解使用所選運算解決方案的優勢和限制。選取的運算解決方案應該符合工作負載效能準則。重要準則包括處理需求、流量模式、資料存取模式、擴展需求，以及延遲要求。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解可以使工作負載受益並滿足效能要求的虛擬化、容器化和管理解決方案。一個工作負載可以包含多種類型的運算解決方案。每個運算解決方案都有不同的特性。根據工作負載規模和運算要求，您可以選

取一個運算解決方案，並將其設定為符合您的需求。雲端架構師應該了解執行個體、容器和函數的優缺點。下列步驟將協助您如何選取運算解決方案，以符合工作負載特性和效能要求。

類型	伺服器	容器	函數
AWS 服務	Amazon Elastic Compute Cloud (Amazon EC2)	Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS)	AWS Lambda
重要特性	具有硬體授權要求的專用選項、置放選項，以及根據運算指標大量選取不同的執行個體系列	易於部署、一致環境、在 EC2 執行個體之上執行、可擴展	執行時間短 (15 分鐘或更少)、最大記憶體和 CPU 不如其他服務高、受管硬體層、可擴展到數百萬個並行要求
常見的使用案例	隨即轉移遷移、整合型應用程式、混合環境、企業應用程式	微型服務、混合環境、	微型服務、事件驅動應用程式

實作步驟：

1. 透過評估選取運算解決方案所在位置 the section called “PERF05-BP06 根據網路要求選擇工作負載的位置”。此位置將限制可供您使用的運算解決方案類型。
2. 識別處理位置要求和應用程式要求的運算解決方案類型
   a. Amazon Elastic Compute Cloud (Amazon EC2) 虛擬伺服器執行個體提供廣泛不同的系列和大小。它們可提供眾多不同功能，包括固態硬碟 (SSD) 和圖形處理單元 (GPU)。EC2 執行個體可在選擇執行個體時提供最大彈性。啟動 EC2 執行個體時，您指定的執行個體類型會決定執行個體的硬體。每種執行個體類型均提供了不同的運算、記憶體和儲存功能。我們會根據這些功能，將執行個體類型分組為不同的執行個體系列。典型使用案例包括：執行企業應用程式、高效能運算 (HPC)、訓練和部署機器學習應用程式，以及執行雲端原生應用程式。

b. Amazon Elastic Container Service (Amazon ECS) 是全受管的容器協調服務，可讓您使用 AWS Fargate，在 EC2 執行個體或無伺服器執行個體的叢集上 自動執行和管理容器。您可以使用 Amazon ECS，搭配 Amazon Route 53、Secrets Manager、AWS Identity and Access Management (IAM) 和 Amazon CloudWatch 等服務。如果您的應用程式已容器化，而且您的工程團隊偏好 Docker 容器，則建議使用 Amazon ECS。

c. Amazon Elastic Kubernetes Service (Amazon EKS) 是全受管的 Kubernetes 服務。您可以選擇使用 AWS Fargate 執行 EKS 叢集，無須佈建和管理伺服器。管理 Amazon EKS 已簡化，因為與 AWS 服務整合，例如 Amazon CloudWatch、Auto Scaling 群組、AWS Identity and Access Management (IAM) 和 Amazon Virtual Private Cloud (VPC)。使用容器時，您必須使用運算指標，為工作負載選取最佳類型，類似於使用運算指標選取 EC2 或 AWS Fargate 執行個體類型的方式。如果您的應用程式已容器化，並且您的工程團隊偏好 Kubernetes 而不是 Docker 容器，則建議使用 Amazon EKS。

d. 您可以使用 AWS Lambda 執行程式碼，支援允許的執行時間、記憶體和 CPU 選項。只要上傳程式碼，AWS Lambda 就會管理程式碼執行和擴展所需的所有項目。您可以將程式碼設定為從其他 AWS 服務自動觸發或直接呼叫它。若是針對雲端開發且短期執行的微型服務架構，建議使用 Lambda。

3. 在您已使用新的運算解決方案進行了試驗之後，請規劃您的遷移並驗證您的效能指標。這是持續程序，請參閱 the section called “PERF02-BP04 透過適當調整大小來確定所需的組態”。

實作計劃的工作量： 如果工作負載從某個運算解決方案移至另一個運算解決方案，則有一個 中等 工作量，其中涉及重構應用程式。

資源

相關文件：

- 使用 AWS 進行雲端運算
- EC2 執行個體類型
- EC2 執行個體的處理器狀態控制
- EKS 容器：EKS 工作節點
- Amazon ECS 容器：Amazon ECS 容器執行個體
- 函數：Lambda 函數組態
- 容器的規範指引
- 無伺服器的規範指引

相關影片：

- 如何為新創公司選擇運算選項
- 優化 AWS 運算的效能和成本 (CMP323-R1)
- Amazon EC2 基礎 (CMP211-R2)
- 支援下一代 Amazon EC2：深入探討 Nitro 系統
- 使用 AWS Inferentia (CMP324-R1) 提供高效能 ML 推斷
- 更好、更快、更便宜的運算：成本優化 Amazon EC2 (CMP202-R1)

相關範例：

- 將 Web 應用程式遷移至容器
- 執行 Serverless Hello World

PERF02-BP02 了解可用的運算組態選項

每個運算解決方案都有選項和組態，可供您支援工作負載特性。了解各種選項如何與您的工作負載互補，以及哪種組態選項最適合您的應用程式。這些選項的範例包括執行個體系列、大小、功能 (GPU、I/O)、爆量、逾時、函數大小、容器執行個體，以及並行。

預期成果： 包括 CPU、記憶體、網路輸送量、GPU、IOPS、流量模式和資料存取模式的工作負載特性會加以記錄，並用來設定運算解決方案以符合工作負載特性。其中每一個指標以及工作負載特有的自訂指標都會加以記錄、監控，然後用來優化運算組態以最好地符合要求。

常用的反模式：

- 使用正用於內部部署的同一運算解決方案。
- 未審查運算選項或執行個體系統以符合工作負載特性。
- 使運算變得更大以確保爆量功能。
- 您針對相同的工作負載使用多個運算管理平台。

建立此最佳實務的優勢： 熟悉 AWS 運算方案，才能為每個工作負載確定合適的解決方案。為工作負載選取運算方案之後，您可以快速試驗這些運算方案，以判斷其符合工作負載需求的程度。為符合工作負載特性而優化的運算解決方案將提高效能、降低成本並提升可靠性。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

如果您的工作負載已使用相同的運算選項超過四週，並且您預計特性未來仍將保持不變，您可以使用 AWS Compute Optimizer 根據運算特性提供一個建議。如果 AWS Compute Optimizer 不是選項，起因於缺少指標、 非支援的執行個體類型 或特性中可預測的變更，則您必須根據負載測試和試驗預測指標。

實作步驟：

1. 您是否正在具有 EC2 啟動類型的 EC2 執行個體或容器上執行？
   a. 您的工作負載是否可以使用 GPU 提升效能？
      i. 加速運算 執行個體是 GPU 型執行個體，可為機器學習培訓、推斷和高效能運算提供最高效能。
   b. 您的工作負載是否執行機器學習推斷應用程式？
      i. AWS Inferentia (Inf1) — Inf1 執行個體專為支援機器學習推斷應用程式所建置。客戶可使用 Inf1，執行大型規模的機器學習推斷應用程式，例如影像辨識、語音辨識、自然語言處理、個人化以及詐騙偵測。您可以在 TensorFlow、PyTorch 或 MXNet 等其中一個常用機器學習架構中建立模型，並使用 GPU 執行個體來訓練模型。將機器學習模型訓練到符合需求之後，您可以使用 AWS Neuron在 Inf1 執行個體上部署模型，AWS Neuron 是一種專業的軟體開發套件 (SDK)，其中包含編譯器、執行時間和分析工具，可將 Inferentia 晶片的機器學習推斷效能優化。
   c. 您的工作負載是否與低階硬體整合以改進效能？
      i. 可現場程式化閘道陣列 (FPGA) — 使用 FPGA，您可以為要求最高的工作負載自訂硬體加速執行，進而優化您的工作負載。您可以利用支援的通用程式設計語言 (例如 C 或 Go)，或以硬體為導向的語言 (例如 Verilog 或 VHDL) 來定義演算法。
   d. 您是否有至少四週的指標，並且可以預測您的流量模式和指標在未來仍大致相同？
      i. 使用 Compute Optimizer 取得哪個運算組態最符合運算特性的機器學習建議。
   e. 您的工作負載效能是否受到 CPU 指標限制？
      i. 運算優化 執行個體非常適合於需要高效處理器的工作負載。
   f. 您的工作負載效能是否受到記憶體指標限制？
      i. 記憶體優化 執行個體會提供大量記憶體，以支援記憶體密集工作負載。
   g. 您的工作負載效能是否受到 IOPS 限制？
      i. 儲存優化 執行個體專為需要對本機儲存進行高循序讀寫存取 (IOPS) 的工作負載而設計。
   h. 您的工作負載特性是否代表所有指標之間的平衡需求？
      i. 您的工作負載 CPU 是否需要爆量，以處理流量中的峰值？

A. 高載效能 執行個體類似於運算優化執行個體，不同之處在於它們可讓您突破運算優化執行個體中所識別的固定 CPU 基準。

ii. 一般用途 執行個體提供所有特性的平衡，以支援各種工作負載。

i. 您的運算執行個體是否正在 Linux 上執行，並受到網路介面卡上的網路輸送量限制？

i. 檢閱 效能問題 5、最佳實務 2：評估可用的聯網功能 來找出正確的執行個體類型和系統，以符合您的效能需求。

j. 在您承諾使用一年的特定可用區域中，您的工作負載是否需要一致且可預測的執行個體？

i. 預留執行個體 確認特定可用區域中的容量保留。預留執行個體非常適合於特定可用區域中所需的運算能力。

k. 您的工作負載是否具有需要專用硬體的授權？

i. 專用主機 支援現有的軟體授權，並協助您滿足合規需求。

l. 您的運算解決方案是否會爆量且需要同步處理？

i. 隨需執行個體 可讓您按小時或秒使用運算容量，無需長期承諾。這些執行個體適合於超過效能基準需求的爆量。

m. 您的運算解決方案為無狀態、容錯和非同步嗎？

i. Spot 執行個體 可讓您針對無狀態、容錯工作負載利用未用的執行個體容量。

2. 您是否正在將容器執行於 Fargate？

a. 您的任務效能是否受到記憶體或 CPU 限制？

i. 使用 任務大小 調整您的記憶體或 CPU。

b. 您的效能是否正受到流量模式爆量限制？

i. 使用 Auto Scaling 組態來比對您的流量模式。

3. 您的運算解決方案是否位於 Lambda？

a. 您是否有至少四週的指標，並且可以預測您的流量模式和指標在未來仍大致相同？

i. 使用 Compute Optimizer 取得哪個運算組態最符合運算特性的機器學習建議。

b. 您沒有足夠的指標來使用 AWS Compute Optimizer 嗎？

i. 如果您沒有可供 Compute Optimizer 使用的指標，請使用 AWS Lambda Power Tuning 協助選取最佳組態。

c. 您的函數效能是否受到記憶體或 CPU 限制？

i. 設定您的 Lambda 記憶體 來符合您的效能需求指標。

d. 您的函數在執行時是否逾時？

i. 變更 逾時設定

e. 您的函數效能是否受到活動和並行爆量限制？

   i. 設定 並行設定 來符合您的效能要求。

f. 您的函數是否以非同步方式執行，並在重試時失敗？

   i. 在 非同步組態 設定中設定最長事件保留期限和最多重試次數限制。

實作計劃的工作量：

若要建立此最佳實務，您必須注意目前運算特性和指標。收集這些指標、建立基準，然後使用這些指標來識別理想的運算選項為 低 至 中等 工作量。這最好由負載測試和試驗進行驗證。

資源

相關文件：

- 使用 AWS 進行雲端運算
- AWS Compute Optimizer
- EC2 執行個體類型
- EC2 執行個體的處理器狀態控制
- EKS 容器：EKS 工作節點
- Amazon ECS 容器：Amazon ECS 容器執行個體
- 函數：Lambda 函數組態

相關影片：

- Amazon EC2 基礎 (CMP211-R2)
- 支援下一代 Amazon EC2：深入探討 Nitro 系統
- 優化 AWS 運算的效能和成本 (CMP323-R1)

相關範例：

- 在 Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小
- AWS Compute Optimizer 示範程式碼

PERF02-BP03 收集與運算相關的指標

若要了解運算資源的執行方式，您必須記錄和追蹤各種系統的使用率。此資料可用來更準確地判斷資源需求。

工作負載可以產生大量資料，例如指標、日誌和事件。判斷您現有的儲存、監控和可觀測性服務是否可以管理產生的資料。識別哪些指標反映資源使用率，並可在整個單一平台上進行收集、彙總和相互關聯。這些指標應該代表您的所有工作負載資源、應用程式和服務，以便您可以輕鬆地取得全系統可見性，並快速識別效能改進機會和問題。

預期成果： 與運算相關資源相關的所有指標都會在單一平台上進行識別、收集、彙總和相互關聯，並實作保留以支援成本和營運目標。

常用的反模式：

- 您只使用手動日誌檔案來搜尋指標。
- 您只將指標發佈到內部工具。
- 您只會使用所選監控軟體記錄的預設指標。
- 您只會在有問題時檢閱指標。

建立此最佳實務的優勢： 若要監控工作負載的效能，您必須記錄一段時間的多個效能指標。這些指標可讓您偵測效能中的異常。它們也會協助針對業務指標衡量效能，以確保您符合工作負載需求。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

識別、收集、彙總運算相關指標，並使其相互關聯。使用 Amazon CloudWatch 這類服務可讓實作更快且更輕鬆維護。除了記錄的預設指標外，還會在您的工作負載內識別和追蹤其他系統等級指標。記錄 CPU 利用率、記憶體、磁碟 I/O，以及網路傳入和傳出指標等資料，以洞悉使用率水平或瓶頸。此資料對於了解工作負載的執行方式，以及運算解決方案的使用方式至關重要。將這些指標納入資料驅動的方法，以主動調整和優化工作負載的資源。

實作步驟：

1. 哪些運算解決方案指標務必要追蹤？
    a. EC2 預設指標
    b. Amazon ECS 預設指標
    c. EKS 預設指標

   d. Lambda 預設指標
   e. EC2 記憶體和磁碟指標
2. 我目前是否具有核准的記錄和監控解決方案？
   a. Amazon CloudWatch
   b. 適用於 OpenTelemetry 的 AWS Distro
   c. 適用於 Prometheus 的 Amazon 受管服務
3. 我是否已識別並設定我的資料保留策略，以符合我的安全和營運目標？
   a. CloudWatch 指標的預設資料保留
   b. CloudWatch Logs 的預設資料保留
4. 如何部署您的指標和記錄彙總代理程式？
   a. AWS Systems Manager 自動化
   b. OpenTelemetry 收集器

實作計劃的工作量： 有一個 中 工作量，用來從所有運算資源識別、追蹤、收集、彙總指標，並使其相互關聯。

資源

相關文件：

- Amazon CloudWatch 文件
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- 存取 Amazon CloudWatch Logs 的 AWS Lambda
- 搭配容器執行個體使用 CloudWatch Logs
- 發佈自訂指標
- AWS Answers：集中式記錄
- 發佈 CloudWatch 指標的 AWS 服務
- 在 AWS Fargate 上監控 Amazon EKS

相關影片：

- AWS 上的應用程式效能管理
- 制定監控計劃

相關範例：

- Level 100：使用 CloudWatch 儀表板進行監控
- Level 100：使用 CloudWatch 儀表板監控 Windows EC2 執行個體
- Level 100：使用 CloudWatch 儀表板監控 Amazon Linux EC2 執行個體

PERF02-BP04 透過適當調整大小來確定所需的組態

分析工作負載的各種效能特性，以及這些特性與記憶體、網路和 CPU 使用量的關係。使用此資料，可以選擇最適合您工作負載描述檔的資源。例如，執行個體的 R 系列可以為記憶體密集型工作負載 (例如資料庫) 提供最佳服務。不過，高載工作負載從彈性容器系統中獲益的程度更高。

常用的反模式：

- 您可以選擇可用於所有工作負載的最大型執行個體。
- 您可以將所有執行個體類型標準化為一種類型，以方便管理。

建立此最佳實務的優勢： 如果熟悉 AWS 運算方案，您可以為各種工作負載確定合適的解決方案。為工作負載選擇各種運算方案後，您就可以靈活快速地試驗這些運算方案，以判斷哪些方案符合工作負載需求。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

透過適當調整大小來修改工作負載組態：若要同時最佳化效能和整體效率，請判斷工作負載所需的資源。對於比起 CPU 需要更多記憶體的系統，選擇記憶體優化的執行個體；或者為那些不耗用太多記憶體的資料處理元件，選擇運算優化的執行個體。適當調整大小可以讓您的工作負載在發揮出色效能的同時僅使用所需資源

資源

相關文件：

- AWS Compute Optimizer
- 使用 AWS 進行雲端運算
- EC2 執行個體類型

- ECS 容器：Amazon ECS 容器執行個體
- EKS 容器：EKS 工作節點
- 函數：Lambda 函數組態
- EC2 執行個體的處理器狀態控制

相關影片：

- Amazon EC2 基礎 (CMP211-R2)
- 更好、更快、更便宜的運算：成本優化 Amazon EC2 (CMP202-R1)
- 使用 AWS Inferentia (CMP324-R1) 提供高效能 ML 推斷
- 優化 AWS 運算的效能和成本 (CMP323-R1)
- 支援下一代 Amazon EC2：深入探討 Nitro 系統
- 如何為新創公司選擇運算選項
- 優化 AWS 運算的效能和成本 (CMP323-R1)

相關範例：

- 在 Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小
- AWS Compute Optimizer 示範程式碼

PERF02-BP05 利用資源的可用彈性

雲端提供的彈性可透過各種機制來動態擴展或減少資源，以滿足需求的變化。結合與運算相關的指標，工作負載可以自動回應變更，並使用最佳資源集來實現其目標。

讓供需達到最佳相符，將工作負載的成本降到最低，但您還需規劃充分的供應，為佈建時間和個別資源失敗而預留。需求可為固定或可變，需要指標和自動化，以確保該項管理不致構成明顯的大量比例失衡成本。

搭配 AWS，您可以使用各種不同的方法使供需相符。成本優化支柱白皮書說明如何使用下列方法來計算成本：

- 需求為主方法
- 緩衝為主方法
- 時間為主方法

您必須確保工作負載部署可以同時處理擴展和縮減事件。建立縮減事件的測試案例，以確保工作負載如預期般運作。

常用的反模式：

- 您可以手動增加容量，對警示做出反應。
- 您在擴展事件之後維持增加容量，而不是縮減規模。

建立此最佳實務的優勢： 設定和測試工作負載彈性將有助於節省金錢、維護性能基準，以及在流量變更時提高可靠性。大多數非生產執行個體在未使用時應該予以停止。雖然可以手動關閉未使用的執行個體，但這種做法在較大規模下不切實際。您也可以利用以磁碟區為基礎的彈性，透過在需求高峰期間自動增加運算執行個體數量，並在需求減少時減少容量，藉此最佳化效能和成本。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

利用彈性：彈性會比對您擁有的資源供應與這些資源的需求。執行個體、容器和函數提供了彈性機制，可能是與自動調整規模功能結合使用，或是作為服務功能提供。利用架構中的彈性，以確保您有足夠的容量滿足所有使用規模的效能需求。確保針對要部署的工作負載類型來驗證擴展或縮減彈性資源的指標。如果您要部署影片轉碼應用程式，則預期為 100% CPU 使用率，且不應做為您的主要指標。或者，您可以針對等待擴展執行個體類型的轉碼任務的佇列深度，來進行測量。確定針對要部署的工作負載類型驗證擴展或縮減彈性資源的指標。安全地縮減工作負載元件，與在必要時擴展資源一樣重要。建立縮減事件的測試案例，以確保工作負載如預期般運作。

資源

相關文件：

- 使用 AWS 進行雲端運算
- EC2 執行個體類型
- ECS 容器：Amazon ECS 容器執行個體
- EKS 容器：EKS 工作節點
- 函數：Lambda 函數組態
- EC2 執行個體的處理器狀態控制

相關影片：

- Amazon EC2 基礎 (CMP211-R2)
- 更好、更快、更便宜的運算：成本優化 Amazon EC2 (CMP202-R1)
- 使用 AWS Inferentia (CMP324-R1) 提供高效能 ML 推斷
- 優化 AWS 運算的效能和成本 (CMP323-R1)
- 支援下一代 Amazon EC2：深入探討 Nitro 系統

相關範例：

- Amazon EC2 Auto Scaling 群組範例
- Amazon EFS 教學課程

PERF02-BP06 根據指標重新評估運算需求

使用系統層級指標來確定工作負載隨時間的行為和要求。透過將可用資源與這些需求進行比較來評估您的工作負載需求，並對運算環境進行變更，以達到最適合工作負載描述檔的狀態。例如，隨著時間的流逝，系統可能會比最初想像的要消耗更多的記憶體，因此轉換到不同的執行個體系列或大小，可以同時提高效能和效率。

常用的反模式：

- 您只需監控系統層級指標，即可深入了解工作負載。
- 您會架構運算需求，以滿足尖峰工作負載要求。
- 當移至新的運算解決方案將符合您的工作負載特性時，您可以使運算解決方案變得更大以符合擴展或效能要求。

建立此最佳實務的優勢： 若要最佳化效能和資源利用率，您需要取得整合操作檢視、即時精細資料和歷史參考。您可以建立自動化儀表板來視覺化此資料，並執行指標運算來獲得操作和使用率的洞見。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

使用資料驅動的方法來優化資源：為了獲得最佳效能和效率，使用從工作負載中收集一段時間的資料來調整和優化資源。查看工作負載中當前資源的使用趨勢，並確定可以在何處進行變更以更好地滿足工作負載的需求。當資源過量使用時，系統效能會降低，而利用率不足則會導致資源使用效率低下和成本增加。

資源

相關文件：

- 使用 AWS 進行雲端運算
- AWS Compute Optimizer
- 使用 AWS 進行雲端運算
- EC2 執行個體類型
- ECS 容器：Amazon ECS 容器執行個體
- EKS 容器：EKS 工作節點
- 函數：Lambda 函數組態
- EC2 執行個體的處理器狀態控制

相關影片：

- Amazon EC2 基礎 (CMP211-R2)
- 更好、更快、更便宜的運算：成本優化 Amazon EC2 (CMP202-R1)
- 使用 AWS Inferentia (CMP324-R1) 提供高效能 ML 推斷
- 優化 AWS 運算的效能和成本 (CMP323-R1)
- 支援下一代 Amazon EC2：深入探討 Nitro 系統

相關範例：

- 在 Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小
- AWS Compute Optimizer 示範程式碼

PERF 3 您如何選取儲存解決方案？

系統的最佳儲存解決方案會根據存取方法類型 (區塊、檔案或物件)、存取模式 (隨機或連續)、所需傳輸量、存取頻率 (線上、離線、封存)、更新頻率 (WORM、動態) 及可用性和耐用性限制而有所不同。Well-Architected 系統使用多重儲存解決方案，並啟用不同功能以提升效能並有效使用資源。

最佳實務

- PERF03-BP01 了解儲存特性和要求

- PERF03-BP02 評估可用的組態選項
- PERF03-BP03 根據存取模式和指標制定決策

PERF03-BP01 了解儲存特性和要求

識別並記載工作負載儲存需求，並定義每個位置的儲存特性。儲存特性的範例包括：可共用的存取、檔案大小、成長率、輸送量、IOPS、延遲、存取模式，以及資料的持久性。請使用這些特性，評估區塊、檔案、物件或執行個體儲存服務對您的儲存需求而言是否為最有效的解決方案。

預期成果： 識別和記載儲存要求，並根據各項儲存要求評估可用的儲存解決方案。根據重要儲存特性，團隊將了解選取的儲存服務對於您的工作負載效能有何助益。重要準則包括資料存取模式、成長率、擴展需求，以及延遲要求。

常見的反模式：

- 所有工作負載只能使用一種儲存類型，例如 Amazon Elastic Block Store (Amazon EBS)。
- 您假設所有工作負載都有類似的儲存存取效能需求。

建立此最佳實務的優勢： 根據已識別和所需的特性選取儲存解決方案，可協助您改善工作負載效能、降低成本，以及減少維護工作負載所需的營運工作量。您的工作負載效能將因儲存服務的解決方案、組態和位置而提高。

未建立此最佳實務時的曝險等級： 高

實作指引

確定對工作負載最重要的儲存效能指標，並使用基準化分析或負載測試將改善做為資料驅動方法的一部分實作。使用這些資料來確定您的儲存解決方案受限的地方，並檢查可以改善此解決方案的組態選項。判斷工作負載的預期增長率，並選擇符合這些增長率的儲存解決方案。研究 AWS 儲存產品，以判斷適用於各種工作負載需求的正確儲存解決方案。在 AWS 中佈建儲存解決方案可增加測試儲存產品的機會，以判斷產品是否適合您的工作負載需求。

AWS 服務	重要特性	常用案例
Amazon S3	99.999999999% 的耐久性、不受限的增長、可從任一處存取、基於存取和彈性有數種成本模型	雲端原生應用程式資料、資料封存、備份、分析、資料湖、靜態網站託管、IoT 資料

AWS 服務	重要特性	常用案例
Amazon S3 Glacier	數秒到數小時的延遲、不受限的增長、最低的成本、長期儲存	資料封存、媒體封存、長期備份保留。
Amazon EBS	儲存大小需要管理和監控、低延遲、持久性儲存、99.8% 到 99.9% 的耐久性，大部分的磁碟區類型只能從一個 EC2 執行個體來存取。	COTS 應用程式、I/O 密集型應用程式、關聯式和 NoSQL 資料庫、備份與復原
EC2 執行個體存放區	預先決定的儲存大小、最低的延遲、不保存、只能從一個 EC2 執行個體存取	COTS 應用程式、I/O 密集型應用程式、記憶體內資料存放區
Amazon EFS	99.999999999% 的耐久性、不受限的增長、可供多個運算服務存取	在多個運算服務間共用檔案的現代化應用程式、用來擴展內容管理系統的檔案儲存
Amazon FSx	支援四種檔案系統 (NetApp、OpenZFS、Windows File Server 和 Amazon FSx for Lustre)，每個檔案系統適用的儲存體各不相同，可供多個運算服務存取	雲端原生工作負載、私有雲端爆量、需要特定檔案系統的已遷移工作負載、VMC、ERP 系統、內部部署檔案儲存與備份
Snow Family	可攜式裝置、256 位元加密、NFS 端點、內建運算、TB 級的儲存容量	將資料遷移至雲端儲存，並且在極端內部部署的條件、災難復原、遠端資料收集中運算
AWS Storage Gateway	對採用雲端技術的儲存體、全受管的內部部署快取提供低延遲的內部部署存取	將內部部署資料遷移至雲端、從內部部署來源填入雲端資料湖、現代化檔案共用。

實作步驟：

1. 使用基準化分析或負載測試收集您儲存需求的重要特性。重要特性包括：

a. 可共用 (哪些元件會存取此儲存體)

b. 成長率

c. 輸送量

d. 延遲

e. I/O 大小

f. 耐久性

g. 存取模式 (讀取/寫入、頻率、激增或一致)

2. 識別哪種類型的儲存解決方案支援您的儲存特性。

a. Amazon S3 是一項物件儲存服務，具有不受限的可擴展性、高可用性，以及多個可存取性選項。對 Amazon S3 輸入和存取物件時，可以使用服務 (例如 Transfer Acceleration 或 存取點) 來支援您的位置、安全需求和存取模式。使用 Amazon S3 效能指導方針 協助您優化 Amazon S3 組態，以符合您的工作負載效能需求。

b. Amazon S3 Glacier 是針對資料封存而建置的 Amazon S3 儲存類別。您有三種封存解決方案可供選擇，存取速度從數毫秒到 5-12 小時都有，且有不同的成本與安全選項。Amazon S3 Glacier 可讓您實作支援個人商業需要和資料特性的資料生命週期，以利符合效能要求。

c. Amazon Elastic Block Store (Amazon EBS) 是專為 Amazon Elastic Compute Cloud (Amazon EC2) 而設計的高效能區塊儲存服務。您可以選擇 SSD 或 HDD 型 解決方案，兩者各有不同特性，分別側重於 IOPS 或 輸送量。EBS 磁碟區非常適用於高效能工作負載，是只能存取連結階段系統的檔案系統、資料庫或應用程式的主要儲存。

d. Amazon EC2 執行個體存放區 類似於 Amazon EBS，同樣也連結至一個 Amazon EC2 執行個體，但執行個體存放區只是暫時性的儲存體，理想情況下應作為緩衝區、快取或其他暫時性內容使用。您無法將執行個體存放區中斷連結，且若執行個體關閉，所有資料都將遺失。執行個體存放區可用於資料無需持續保存的高 I/O 效能與低延遲使用案例。

e. Amazon Elastic File System (Amazon EFS) 是一個可掛載的檔案系統，可供多種類型的運算解決方案存取。Amazon EFS 會自動增長及縮減儲存體，解已進行效能優化而提供一致的低延遲。EFS 有 兩種效能組態模式：一般用途和最大 I/O。「一般用途」的讀取延遲低於 1 毫秒，寫入延遲則低於 10 毫秒。「最大 I/O」功能可支援數千個需要共用檔案系統的運算執行個體。Amazon EFS 支援 兩種輸送量模式：高載和佈建。經歷激增存取模式的工作負載將可獲益於高載輸送量模式，而持續偏高的工作負載則可在佈建輸送量模式下保有高效能。

f. Amazon FSx 建置於最新的 AWS 運算解決方案之上，用以支援四個常用的檔案系統：NetApp ONTAP、OpenZFS、Windows File Server 和 Lustre。Amazon FSx 延遲、輸送量和 IOPS 會隨著檔案系統而不同，當您為工作負載需求選取適當的檔案系統時，應予以考量。

g. AWS Snow Family 是一種儲存和運算裝置，支援以雲端為目標的線上和離線資料遷移，以及內部部署的資料儲存與運算。AWS Snow 裝置支援大量收集內部部署資料、處理該資料，以及將該資料移至雲端。在談到檔案數量、檔案大小和壓縮時， 有幾項列載於文件中的效能最佳實務 可供參考。

h. AWS Storage Gateway 提供內部部署應用程式對雲端架構儲存體的存取。AWS Storage Gateway 支援多個雲端儲存服務，包括 Amazon S3、Amazon S3 Glacier、Amazon FSx 和 Amazon EBS。此外也支援多種通訊協定，例如 iSCSI、SMB 和 NFS。它會藉由在內部快取經常存取的資料來提供低延遲的效能，並且只會將有所變更和已壓縮的資料傳送至 AWS。

3. 在您體驗過新的儲存解決方案並且找出最佳組態之後，請規劃您的遷移並驗證您的效能指標。這是連續性的程序，且在重要特性變更時，或是可用的服務或選項變更時，應重新評估。

實作計劃的工作量： 如果工作負載從某個儲存解決方案移至另一個解決方案，可能會有 中等 工作量涉及重構應用程式。

資源

相關文件：

- Amazon EBS 磁碟區類型
- Amazon EC2 儲存
- Amazon EFS：Amazon EFS 效能
- Amazon FSx for Lustre 效能
- Amazon FSx for Windows File Server 效能
- Amazon FSx for NetApp ONTAP 效能
- Amazon FSx for OpenZFS 效能
- Amazon S3 Glacier：Amazon S3 Glacier 文件
- Amazon S3：請求率和效能考量
- AWS 的雲端儲存
- AWS Snow Family
- EBS I/O 特性

相關影片：

- 深入探討 Amazon EBS (STG303-R1)

- 使用 Amazon S3 優化儲存效能 (STG343)

相關範例：

- Amazon EFS CSI 驅動程式
- Amazon EBS CSI 驅動程式
- Amazon EFS 公用程式
- Amazon EBS 自動擴展
- Amazon S3 範例
- Amazon FSx for Lustre Container Storage Interface (CSI) 驅動程式

PERF03-BP02 評估可用的組態選項

評估各種特性和組態選項，以及它們與儲存的關係。了解如何在何處使用佈建 IOPS、SSD、磁帶儲存、物件儲存、存檔儲存或暫時性儲存，以優化工作負載的儲存空間和效能。

Amazon EBS 提供了各種選項，可讓您優化工作負載的儲存效能和成本。這些選項分為兩大類：適用於交易工作負載 (例如資料庫和開機磁碟區) 且支援 SSD 的儲存 (效能主要取決於 IOPS)；以及適用於輸送量密集型工作負載 (例如 MapReduce 和日誌處理) 且支援 HDD 的儲存 (效能主要取決於 MB/s)。

支援 SSD 的磁碟區包括適用於延遲敏感的交易工作負載的最高效能佈建 IOPS SSD 和可使各種交易資料的價格和效能之間達到平衡的一般用途 SSD。

Amazon S3 Transfer Acceleration 可以在用戶端和 S3 儲存貯體之間快速進行遠距離檔案傳輸。Transfer Acceleration 利用了 Amazon CloudFront 分散於全球的節點，以將資料路由至優化的網路路徑。對於 S3 儲存貯體中具有大量 GET 請求的工作負載，請將 Amazon S3 與 CloudFront 搭配使用。上傳大型檔案時，請使用分段上傳，同時上傳多個組件以協助最大化網路輸送量。

Amazon Elastic File System (Amazon EFS) 提供簡單、可擴展、全受管的彈性 NFS 檔案系統，可與 AWS 雲端服務和內部部署資源搭配使用。為了支援各式各樣的雲端儲存工作負載，Amazon EFS 提供兩種效能模式：一般用途效能模式和最高 I/O 效能模式。針對檔案系統，提供兩種傳輸量模式供您選擇：爆量傳輸量和佈建傳輸量。若要判斷工作負載適用的設定，請參閱 Amazon EFS 使用者指南。

Amazon FSx 提供四種檔案系統供您選擇： Amazon FSx for Windows File Server (適用於企業工作負載)、 Amazon FSx for Lustre (適用於高效能工作負載)、 Amazon FSx for NetApp ONTAP (適用於 NetApps 熱門 ONTAP 檔案系統)，以及 Amazon FSx for OpenZFS (適用於 Linux 型檔案伺服器)。FSx 支援 SSD，旨在提供快速、可預測、可擴展和一致的效能。Amazon FSx 檔案系統提供持續的高讀寫速度，以及一致的低延遲資料存取。您可以選擇所需的輸送量等級，以符合工作負載的需求。

常用的反模式：

- 所有工作負載只能使用一種儲存類型，例如 Amazon EBS。
- 您為所有工作負載使用已佈建的 IOPS，卻未針對所有儲存層進行實際測試。
- 您假設所有工作負載都有類似的儲存存取效能需求。

建立此最佳實務的優勢： 評估所有儲存服務選項可降低基礎設施成本和維護工作負載所需的工作量。這可能會加速部署新服務和功能的交期。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

判斷儲存特性：評估儲存解決方案時，判斷您需要的儲存特性，例如共用能力、檔案大小、快取大小、延遲、輸送量和資料的持久性。然後，將您的需求比對最符合您需求的 AWS 服務。

資源

相關文件：

- AWS 的雲端儲存
- Amazon EBS 磁碟區類型
- Amazon EC2 儲存
- Amazon EFS：Amazon EFS 效能
- Amazon FSx for Lustre 效能
- Amazon FSx for Windows File Server 效能
- Amazon Glacier：Amazon Glacier 文件
- Amazon S3：請求率和效能考量
- AWS 的雲端儲存
- AWS 的雲端儲存
- EBS I/O 特性

相關影片：

- 深入探討 Amazon EBS (STG303-R1)

- 使用 Amazon S3 優化儲存效能 (STG343)

相關範例：

- Amazon EFS CSI 驅動程式
- Amazon EBS CSI 驅動程式
- Amazon EFS 公用事業
- Amazon EBS 自動擴展
- Amazon S3 範例

PERF03-BP03 根據存取模式和指標制定決策

根據工作負載的存取模式選擇儲存系統，並透過決定工作負載存取資料的方式來設定儲存系統。選擇物件儲存而不是區塊儲存，以提高儲存效率。設定您選擇的儲存選項以匹配資料存取模式。

您存取資料的方式會影響儲存解決方案的執行方式。選擇最適合您的存取模式的儲存解決方案，或者考慮變更存取模式，以符合儲存解決方案，從而最大化效能。

建立 RAID 0 陣列可讓您實現比在單一磁碟區上佈建的檔案系統更高的效能。當 I/O 效能比容錯能力更重要時，請考慮使用 RAID 0。例如，您可以將其與頻繁使用的資料庫 (其資料複寫已獨立設定) 搭配使用。

針對工作負載使用的所有儲存選項，為您的工作負載選取適當的儲存指標。當使用的檔案系統使用爆量額度時，請建立警示，以便在即將接近額度限制時告知您。您必須建立儲存儀表板，以顯示整體工作負載儲存體運作狀態。

對於固定大小的儲存系統 (例如 Amazon EBS 或 Amazon FSx)，請確保監控使用的儲存量佔整體儲存大小的比例，並在達到閾值時建立自動化 (如可能) 以增加儲存大小

常用的反模式：

- 您假設在客戶未投訴的情況下，儲存效能即已足敷使用。
- 您只使用一個存儲層 – 假設所有工作負載都適合該層。

建立此最佳實務的優勢： 您需要取得整合操作檢視、即時精細資料和歷史參考，以優化效能和資源使用率。您可以建立 1 秒精細度的自動儀表板和資料，以對資料執行指標運算，並獲得儲存需求的操作和使用率洞見。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

優化您的儲存使用和存取方式：根據工作負載的存取模式和可用儲存選項的特性選擇儲存系統。確定儲存資料的最佳位置，這將讓您能夠滿足您的要求，同時減少開銷。在設定和與資料互動時，根據儲存的特性利用效能優化和存取模式 (例如，對儲存磁碟區進行條帶化或對資料進行分割)。

為儲存選項選取適當的指標：確保為工作負載選取適當的儲存指標。每個儲存選項提供各種指標來追蹤工作負載在一段時間內的執行狀況。確保針對任何儲存高載指標進行測量 (例如監控 Amazon EFS 的爆量額度)。對於大小固定的儲存系統，例如 Amazon Elastic Block Store 或 Amazon FSx，請確保您監控的是使用儲存量與整體儲存大小的比較情況。儘可能建立自動化以在達到閾值時增加儲存大小。

監控指標：Amazon CloudWatch 可以收集架構中各種資源的指標。您還可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方解決方案來設定可指出何時超過閾值的警示。

資源

相關文件：

- Amazon EBS 磁碟區類型
- Amazon EC2 儲存
- Amazon EFS：Amazon EFS 效能
- Amazon FSx for Lustre 效能
- Amazon FSx for Windows File Server 效能
- Amazon Glacier：Amazon Glacier 文件
- Amazon S3：請求率和效能考量
- AWS 的雲端儲存
- EBS I/O 特性
- 使用 Amazon CloudWatch 監控和了解 Amazon EBS 效能

相關影片：

- 深入探討 Amazon EBS (STG303-R1)
- 使用 Amazon S3 優化儲存效能 (STG343)

相關範例：

- Amazon EFS CSI 驅動程式
- Amazon EBS CSI 驅動程式
- Amazon EFS 公用事業
- Amazon EBS 自動擴展
- Amazon S3 範例

PERF 4 您如何選擇資料庫解決方案？

系統的最佳資料庫解決方案可能會依可用性、一致性、分割容錯度、延遲、耐用性、可擴展性及查詢能力的需求而有所不同。許多系統針對不同子系統使用不同資料庫解決方案，並啟用不同功能以提升效能。為系統選錯資料庫解決方案和功能，可能使效能達成效率降低。

最佳實務

- PERF04-BP01 了解資料特性
- PERF04-BP02 評估可用選項
- PERF04-BP03 收集並記錄資料庫效能指標
- PERF04-BP04 根據存取模式選擇資料儲存
- PERF04-BP05 根據存取模式和指標優化資料儲存

PERF04-BP01 了解資料特性

選擇您的資料管理解決方案，以最佳方式符合工作負載資料集的特性、存取模式和要求。在選取和實作資料管理解決方案時，您必須確保查詢、擴展和儲存特性支援工作負載資料要求。了解各種資料庫選項如何符合您的資料模型，以及哪些組態選項最適合您的使用案例。

AWS 提供眾多資料庫引擎，包括關聯式、鍵值、文件、記憶體內、圖形、時間序列和總帳資料庫。每個資料管理解決方案都有選項和組態，可供您支援使用案例和資料模型。您的工作負載能夠根據資料特性使用數個不同的資料庫解決方案。藉由選取特定問題的最佳資料庫解決方案，您可以擺脫整合型資料庫，這些資料庫採用一體適用的方法，但此方法有限制性，並專注於管理資料以符合客戶的需求。

預期成果： 工作負載資料特性會加以記錄，並提供足夠的詳細資訊，以利於選取和設定支援的資料庫解決方案，而且這些特性可讓您洞悉潛在的替代方案。

常用的反模式：

- 未考慮將大型資料集分割成具有類似特性之較小資料集合的方法，這會導致錯失使用更多專用資料庫，更好地符合資料和成長特性的機會。

- 未預先識別資料存取模式，這會導致稍後進行昂貴且複雜的修改。
- 使用不會視需要快速擴展的資料儲存策略來限制成長
- 為所有工作負載選擇一個資料庫類型和廠商。
- 堅持使用某個資料庫解決方案，因為對某種特定類型的資料庫解決方案具有內部經驗和知識。
- 保留資料庫解決方案，因為它在內部部署環境中運作良好。

建立此最佳實務的優勢： 熟悉所有 AWS 資料庫解決方案，以便您可以判斷適合各種工作負載的正確資料庫解決方案。為工作負載選取適當的資料庫解決方案之後，您可以快速試驗每個資料庫產品，以判斷它們是否能繼續滿足您的工作負載需求。

若未建立此最佳實務，暴露的風險等級： 高

- 可能未識別潛在的節省成本。
- 可能無法將資料保護到所需的等級。
- 資料存取和儲存效能可能不是最佳的。

實作指引

定義工作負載的資料特性和存取模式。檢閱所有可用的資料庫解決方案來識別哪個解決方案支援您的資料要求。在特定工作負載內，可能選取多個資料庫。評估每個服務或服務群組，並個別存取它們。如果針對部分或全部資料識別了潛在的替代資料管理解決方案，請嘗試可能解開成本、安全性、效能和可靠性優勢的替代實作。如果採用新的資料管理方式，請更新現有文件。

類型	AWS 服務	重要特性	常見的使用案例
關聯式	Amazon RDS、Amazon Aurora	參考完整性、ACID 交易、寫入時的結構描述	ERP、CRM、商務現成軟體
鍵值	Amazon DynamoDB	高輸送量、低延遲、近乎無限可擴展性	購物車 (商務)、產品型錄、聊天應用程式
文件	Amazon DocumentDB	儲存 JSON 文件並查詢任何屬性	內容管理 (CMS)、客戶設定檔、行動應用程式

類型	AWS 服務	重要特性	常見的使用案例
記憶體內	Amazon ElastiCache、Amazon MemoryDB	微秒延遲	快取、遊戲排行榜
圖形	Amazon Neptune	高度關聯式資料，其中資料之間的關係有意義	社交網路、個人化引擎、詐騙偵測
時間序列	Amazon Timestream	其中主要維度為時間的資料	DevOps、IoT、監控
寬欄	Amazon Keyspaces	Cassandra 工作負載。	產業設備維護、路由優化
總帳	Amazon QLDB	不可變且可加密驗證的變更總帳	記錄、醫療保健、供應鏈、金融機構的系統

實作步驟

1. 如何建構資料？(例如，非結構化、鍵值、半結構化、關聯式)
   a. 如果資料是非結構化，請考慮物件存放區，例如 Amazon S3 或 NoSQL 資料庫，例如 Amazon DocumentDB。
   b. 若為鍵值資料，請考慮 DynamoDB， ElastiCache for Redis 或 MemoryDB。
   c. 如果資料具有關聯式結構，需要哪個層級的參考完整性。
      i. 針對外部索引鍵限制，關聯式資料庫 (例如 Amazon RDS 和 Aurora) 可以提供此層級的完整性。
      ii. 通常，在 NoSQL 資料模型內，您會將資料去正規化為單一文件或文件集合，以便可在單一請求中擷取，而不是跨文件或資料表聯結。
2. 需要 ACID (單元性、一致性、隔離行為、持續性) 合規嗎？
   a. 如果需要與關聯式資料庫相關聯的 ACID 屬性，請考慮關聯式資料庫，例如 Amazon RDS 和 Aurora。
3. 需要哪個一致性模式？

   a. 如果您的應用程式可以容忍最終一致性，請考慮 NoSQL 實作。檢閱其他特性以協助選擇哪個 NoSQL 資料庫 最適用。
   b. 如果需要高度一致性，您可以使用高度一致性讀取，搭配 DynamoDB 或關聯式資料庫，例如 Amazon RDS。
4. 必須支援哪些查詢和結果格式？(例如，SQL、CSV、Parque、Avro、JSON 等)
5. 存在哪些資料類型、欄位大小和整體數量？(例如，文字、數字、空間、已計算的時間序列、二進位或 Blob 文件)
6. 儲存要求如何隨時間變更？ 這如何影響可擴展性？
   a. 無伺服器資料庫，例如 DynamoDB 和 Amazon Quantum Ledger Database ，將自動擴增至近乎無限的儲存。
   b. 關聯式資料庫在佈建的儲存上具有上限，一旦達到這些限制，通常必須透過碎片化這類機制進行水平分割。
7. 讀取查詢與寫入查詢的比例是多少？ 快取可能改善效能嗎？
   a. 包含大量讀取作業的工作負載可以從快取層中受益，這可以是 ElastiCache 或 DAX 如果資料庫是 DynamoDB。
   b. 讀取也可以卸載至具有關聯式資料庫的讀取複本，例如 Amazon RDS。
8. 儲存和修改 (OLTP - 線上交易處理) 或擷取和報告 (OLAP - 線上分析處理) 是否具有更高的優先順序？
   a. 如需高輸送量交易處理，請考慮 NoSQL 資料庫，例如 DynamoDB 或 Amazon DocumentDB。
   b. 如需分析查詢，請考慮單欄式資料庫，例如 Amazon Redshift 或將資料匯出至 Amazon S3 並執行分析，方法為使用 Athena 或 QuickSight。
9. 此資料的敏感程度，以及其需要哪個等級的保護和加密？
   a. 所有 Amazon RDS 和 Aurora 都支援使用 AWS KMS 進行靜態資料加密。Microsoft SQL Server 和 Oracle 也會在使用 Amazon RDS 時支援原生透明資料加密 (TDE)。
   b. 針對 DynamoDB，您可以使用更精細的存取控制，搭配 IAM 控制誰可以存取金鑰等級的哪些資料。
10. 資料需要哪個等級的耐久性？
   a. Aurora 會自動跨區域內的三個可用區域複寫您的資料，這表示您的資料可以耐久，因而資料遺失的機會降低。
   b. DynamoDB 會自動跨多個可用區域進行複寫，這會提供高可用性和資料耐久性。
   c. Amazon S3 提供 99.999999999% 耐久性。許多資料庫服務 (例如 Amazon RDS 和 DynamoDB) 支援將資料匯出至 Amazon S3，進行長期保留和封存。

11.復原時間目標 (RTO) 或復原點目標 (RPO) 要求是否影響解決方案？

a. Amazon RDS、Aurora、DynamoDB、Amazon DocumentDB 和 Neptune 全都支援時間點復原，以及隨需備份和還原。

b. 針對高可用性要求，可以全域複寫 DynamoDB 資料格，方法為使用 全域資料表 功能，而且可以使用全域資料庫功能。跨多個區域複寫 Aurora 叢集。此外，還可以使用跨區域複寫，跨 AWS 區域 複寫 S3 儲存貯體。

12.是否希望擺脫商務資料庫引擎/授權成本？

a. 考慮開放原始碼引擎，例如 Amazon RDS 或 Aurora 上的 PostgreSQL 和 MySQL

b. 利用 AWS DMS 和 AWS SCT 從商務資料庫引擎遷移至開放原始碼

13.對資料庫的操作期望是什麼？ 移至受管服務是否為主要問題？

a. 利用 Amazon RDS 而非 Amazon EC2 和 DynamoDB，或利用 Amazon DocumentDB 而非自行託管 NoSQL 資料庫，可以降低營運負擔。

14.目前如何存取資料庫？ 它是否只是應用程式存取，或是否有商業智能 (BI) 使用者和其他連網的現成應用程式？

a. 如果您依賴外部工具，則可能必須維護與其所支援資料庫的相容性。Amazon RDS 完全與其支援的不同引擎版本相容，包括 Microsoft SQL Server、Oracle、MySQL 和 PostgreSQL。

15.下列是潛在資料管理服務的清單，以及最能在哪裡使用這些服務：

a. 關聯式資料庫會使用預先定義的結構描述和它們之間的關係來儲存資料。這些資料庫旨在支援 ACID (單元性、一致性、隔離行為、持續性) 交易，並維護參考完整性和強大的資料一致性。許多傳統應用程式、企業資源規劃 (ERP)、客戶關係管理 (CRM) 和電子商務都使用關聯式資料庫來儲存資料。您可以在 Amazon EC2 上執行許多資料庫引擎，或選擇其中一種 AWS 受管 資料庫服務： Amazon Aurora、 Amazon RDS和 Amazon Redshift。

b. 鍵值資料庫已針對常見的存取模式進行優化，通常用於儲存和擷取大量資料。即使有極大量的並行請求，這些資料庫也能提供快速回應。高流量 Web 應用程式、電子商務系統和遊戲應用程式是鍵值資料庫的典型使用案例。在 AWS 中，您可以利用 Amazon DynamoDB，這是一個全受管、多區域、多主機、耐用的資料庫，針對網際網路規模的應用程式提供內建安全性、備份和還原以及記憶體內快取。

c. 記憶體內資料庫適用於需要即時存取資料、最低延遲和最高輸送量的應用程式。透過將資料直接儲存在記憶體中，這些資料庫可為應用程式提供微秒延遲，因為毫秒延遲不足以因應其需求。您可以將記憶體內資料庫用於應用程式快取、工作階段管理、遊戲排行榜和地理空間應用程式。 Amazon ElastiCache 是全受管的記憶體內資料存放區，與 Redis 或 Memcached。若應用程式也需要更高的耐久性要求， Amazon MemoryDB for Redis 會將其結合為耐久的記憶體內資料庫服務，以取得超快效能。

d. 文件資料庫旨在將半結構化資料儲存為 JSON 類文件。這些資料庫可協助開發人員快速建置和更新應用程式，例如內容管理、目錄和使用者設定檔。 Amazon DocumentDB 是快速、可擴展、高度可用且全受管的文件資料庫服務，可支援 MongoDB 工作負載。

e. 寬欄存放區是一種 NoSQL 資料庫類型。它使用表格、列和欄，但與關聯式資料庫不同，在同一個表格中，欄的名稱和格式會因列而異。您通常會在大規模工業應用程式中看到寬欄存放區，用於設備維護、叢集管理和路由優化。 Amazon Keyspaces (適用於 Apache Cassandra) 是寬欄的可擴展、高度可用且受管的 Apache Cassandra 相容資料庫服務。

f. 圖形資料庫適用此類應用程式：必須在高度連線圖形資料集之間，大規模導覽和查詢數百萬個關係，並且只有毫秒延遲。許多公司使用圖形資料庫進行詐騙偵測、社交聯網和推薦引擎。 Amazon Neptune 是快速、可靠、全受管的圖形資料庫服務，可讓您輕鬆建立和執行搭配高度連線資料集使用的應用程式。

g. 時間序列資料庫可有效率地從隨時間變化的資料收集、合成和衍生洞見。IoT 應用程式、DevOps 和工業遙測可以利用時間序列資料庫。 Amazon Timestream 是適用於 IoT 和操作應用程式的快速、可擴展、全受管時間序列資料庫服務，每天可輕鬆儲存和分析數兆個事件。

h. 總帳資料庫提供集中化且受信任的機構，為每個應用程式維護可擴展、不可變且以密碼編譯方式驗證的交易記錄。我們會看到用於記錄、供應鏈、註冊甚至銀行交易系統的總帳資料庫。 Amazon Quantum Ledger Database (Amazon QLDB) 是全受管總帳資料庫，提供透明、不可變且可加密驗證的交易日誌，由集中式信任的授權單位所擁有。Amazon QLDB 會追蹤每個應用程式資料變更，並維護一段時間內完整且可驗證的變更歷史記錄。

實作計劃的工作量： 如果工作負載從某個資料庫解決方案移至另一個資料庫解決方案，則有一個 高 工作量，其中涉及重構資料和應用程式。

資源

相關文件：

- AWS 的雲端資料庫
- AWS 資料庫快取
- Amazon DynamoDB Accelerator
- Amazon Aurora 最佳實務
- Amazon Redshift 效能
- Amazon Athena 10 大效能秘訣
- Amazon Redshift Spectrum 最佳實務

- Amazon DynamoDB 最佳實務
- 在 EC2 與 Amazon RDS 之間選擇
- 實作 Amazon ElastiCache 的最佳實務

相關影片：

- AWS 專用資料庫 (DAT209-L)
- 探究 Amazon Aurora 儲存的奧秘：運作方式 (DAT309-R)
- 深入探討 Amazon DynamoDB：進階設計模式 (DAT403-R1)

相關範例：

- 使用 Amazon Redshift 資料共用來優化資料模式
- 資料庫遷移
- MS SQL Server - AWS Database Migration Service (DMS) 複寫示範
- 資料庫現代化實際操作研討會
- Amazon Neptune 範例

PERF04-BP02 評估可用選項

了解可用的資料庫選項，以及這些選項如何在您選取資料管理解決方案之前優化您的效能。使用負載測試，識別對您的工作負載而言很重要的資料庫指標。在探索資料庫選項時，請考慮各種層面，例如參數群組、儲存選項、記憶體、運算、讀取複本、最終一致性、連線集區，以及快取選項。嘗試使用這些不同的組態選項來改善指標。

預期成果： 一個工作負載可以具有一或多個根據資料類型使用的資料庫解決方案。資料庫功能和優勢完美符合資料特性、存取模式和工作負載要求。要優化資料庫效能和成本，您必須評估資料存取模式，以判斷適當的資料庫選項。評估可接受的查詢時間，以確保選取的資料庫選項可以符合要求。

常用的反模式：

- 未識別資料存取模式。
- 未意識到所選資料庫管理解決方案的組態選項。
- 僅依靠增加執行個體大小，而不查看其他可用的組態選項。
- 未測試所選解決方案的擴展特性。

建立此最佳實務的優勢： 藉由探索和嘗試使用資料庫選項，您能夠降低基礎設施成本、改善效能和可擴展性，以及減少維護工作負載所需的工作量。

若未建立此最佳實務，暴露的風險等級： 高

- 必須優化 一體適用 資料庫，表示做出不必要的妥協。
- 由於未設定資料庫解決方案以符合流量模式，因此成本更高。
- 擴展問題可能會出現操作問題。
- 可能無法將資料保護到所需的等級。

實作指引

了解您的工作負載資料特性，以便您可以設定資料庫選項。執行負載測試來識別您的重要效能指標和瓶頸。使用這些特性和指標，來評估資料庫選項並嘗試使用不同的組態。

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
擴展運算	增加執行個體大小，Aurora無伺服器執行個體會自動擴展，以回應負載中的變更	隨需容量模式下的自動讀取/寫入擴展，或所佈建容量模式下所佈建讀取/寫入容量的自動擴展	增加執行個體大小	增加執行個體大小、將節點新增至叢集	增加執行個體大小	自動擴展以調整容量	隨需容量模式下的自動讀取/寫入擴展，或所佈建容量模式下所佈建讀取/寫入容量的自動擴展	自動擴展以調整容量
橫向擴展讀取	所有引擎都	增加已佈建的	讀取複本	讀取複本	讀取複本。支	自動擴展	增加已佈建的	自動縱向擴展

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
	支援讀取複本。Aurora 支援自動擴展讀取複本執行個體。	讀取容量單位			援自動擴展讀取複本執行個體		讀取容量單位	至記錄的並行限制
橫向擴展寫入	增加執行個體大小、在應用程式中批次寫入、在資料庫前面新增佇列。透過跨多個執行個體的應用程式層級共用來水平擴展	增加已佈建的寫入容量單位。確定最佳的分區索引鍵以防止分割層級寫入限流	增加主要執行個體大小	在叢集模式下使用 Redis 跨碎片散發寫入	增加執行個體大小	寫入要求可能會在擴展時進行限流。如果遇到限流異常，請繼續以相同 (或更高) 的輸送量傳送資料以自動擴展。批次寫入以減少並行寫入要求	增加已佈建的寫入容量單位。確定最佳的分區索引鍵以防止分割層級寫入限流	自動縱向擴展至記錄的並行限制

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
引擎組態	參數群組	不適用	參數群組	參數群組	參數群組	不適用	不適用	不適用
快取	記憶體內快取，可透過參數群組設定。與 ElastiCache for Redis 等專用快取配對，以卸載經常存取項目的請求	DAX (DAX) 可用的完全受管快取	記憶體內快取。或者，與 ElastiCache for Redis 等專用快取配對，以卸載經常存取項目的請求	主要功能為快取	使用查詢結果快取來快取唯讀查詢的結果	Timestream 有兩個儲存層；其中一個是高效能記憶體內層	部署個別的專用快取 (例如 ElastiCache for Redis)，以卸載經常存取項目的請求	不適用

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoD	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspace	Amazon QLDB
高可用性/災難復原	生產工作負載的建議組態是在第二個可用區域中執行待命執行個體，以在區域內提供彈性。 若為跨區域的彈性，可以使用 Aurora 全域資料庫	區域內的高可用性可以使用 DynanoDB 全域資料表跨區域複寫資料表	如需可用性，跨可用區域建立多個執行個體。可以跨區域共用快照，也可以使用 DMS 複寫叢集，以提供跨區域複寫/災難復原	生產叢集的建議組態是在次要可用區域中建立至少一個節點。ElastiCache Global Datastore 可以用來跨區域複寫叢集。	其他可用區域中的讀取複本會充當容錯移轉目標。可以跨區域共用快照，也可以使用 Neptune 串流複寫叢集，以在兩個不同區域的兩個叢集之間複寫資料。	可在區域內高度使用。跨區域複寫需要使用 Timestream SDK 進行自訂應用程式開發	區域內的高可用性 跨區域複寫需要自訂應用程式邏輯或第三方工具	區域內的高可用性 若要跨區域複寫，請將 Amazon QLDB 日誌的內容匯出到 S3 儲存貯體，並設定儲存貯體進行跨區域複寫。

實作步驟

1. 哪些組態選項適用於選取的資料庫？
    a. Amazon RDS 和 Aurora 的參數群組可讓您調整常用的資料庫引擎層級設定，例如針對快取分配的記憶體或調整資料庫的時區

b. 針對佈建的資料庫服務，例如 Amazon RDS、Aurora、Neptune、Amazon DocumentDB，以及在 Amazon EC2 上部署的服務，您可以變更執行個體類型、佈建的儲存，以及新增讀取複本。

c. DynamoDB 可讓您指定兩個容量模式：隨需和已佈建。若要處理不同的工作負載，您可以隨時在這些模式之間進行變更，以及在已佈建模式下增加分配的容量。

2. 工作負載是大量讀取或大量寫入？

a. 哪些解決方案適用於卸載讀取 (讀取複本、快取等)？

i. 若為 DynamoDB 資料表，您可以使用 DAX 卸載讀取，以進行快取。

ii. 對於關聯式資料庫，您可以建立一個 ElastiCache for Redis 叢集，並將您的應用程式設定為首先從快取中讀取，如果請求的項目不存在，則退回到資料庫。

iii. 關聯式資料庫 (例如 Amazon RDS 和 Aurora) 和已佈建的 NoSQL 資料庫 (例如 Neptune 和 Amazon DocumentDB) 全都支援新增讀取複本，以卸載工作負載的讀取部分。

iv. 無伺服器資料庫 (例如 DynamoDB) 將自動擴展。確定您已佈建足夠的讀取容量單位 (RCU) 來處理工作負載。

b. 哪些解決方案適用於擴增寫入 (分區索引鍵碎片、引進佇列等)？

i. 對於關聯式資料庫，您可以增加執行個體的大小，以適應增加的工作負載或增加已佈建的 IOPS，以允許增加基礎儲存的輸送量。

- 您也可以在資料庫前面引進佇列，而不是直接寫入至資料庫。此模式允許您將擷取與資料庫分離並控制流量，因此資料庫不會癱瘓。
- 批次處理寫入請求而不是建立許多短期交易，有助於改善高寫入量關聯式資料庫中的輸送量。

ii. DynamoDB 之類的無伺服器資料庫可以自動擴展寫入輸送量，或透過調整已佈建的容量單位 (WCU) 來進行，取決於容量模式。

- 您仍會遇到 常用 分區的問題，但這時您達到特定分區索引鍵的輸送量限制。這可以透過選擇更均勻分佈的分區索引鍵，或對分區索引鍵進行寫入碎片化來緩解。

3. 目前或預期的每秒尖峰交易 (TPS) 有多少？ 使用此流量和此容量 +X% 進行測試，以了解擴展特性。

a. 原生工具 (例如 pg_bench for PostgreSQL) 可以用來對資料庫進行壓力測試，並了解瓶頸和擴展特性。

b. 應該擷取生產類流量，以便可以將其重播，來模擬除了綜合工作負載之外的真實條件。

4. 如果使用無伺服器或彈性可擴展運算，請測試在資料庫上進行此擴展的影響。若適用，請引進連線管理或集區，以降低對資料庫的影響。

a. RDS Proxy 可以搭配 Amazon RDS 和 Aurora 使用，以管理資料庫的連線。

   b. 無伺服器資料庫 (例如 DynamoDB) 沒有與其相關聯的連線，但會考慮佈建的容量和自動擴展政策來處理負載中的峰值。
5. 負載是否可預測、負載中是否有峰值，以及是否有閒置期間？
   a. 若有閒置期間，請考慮在這些時段縮減佈建的容量或執行個體大小。Aurora Serverless V2 將根據負載自動擴增和縮減。
   b. 針對非生產執行個體，請考慮在這些非運作時刻暫停或停止這些執行個體。
6. 您是否需要根據存取模式和資料特性分割和分解資料模型？
   a. 考慮使用 AWS DMS 或 AWS SCT，將您的資料移至其他服務。

實作計劃的工作量：

若要建立此最佳實務，您必須注意目前資料特性和指標。收集這些指標、建立基準，然後使用這些指標來識別理想的資料庫組態選項為 低 至 中 工作量。這最好由負載測試和試驗進行驗證。

資源

相關文件：

- AWS 的雲端資料庫
- AWS 資料庫快取
- Amazon DynamoDB Accelerator
- Amazon Aurora 最佳實務
- Amazon Redshift 效能
- Amazon Athena 10 大效能秘訣
- Amazon Redshift Spectrum 最佳實務
- Amazon DynamoDB 最佳實務

相關影片：

- AWS 專用資料庫 (DAT209-L)
- 探究 Amazon Aurora 儲存的奧秘：運作方式 (DAT309-R)
- 深入探討 Amazon DynamoDB：進階設計模式 (DAT403-R1)

相關範例：

- Amazon DynamoDB 範例
- AWS 資料庫遷移範例
- 資料庫現代化研討會
- 使用 Amazon RDS for PostgreSQL 資料庫上的參數

PERF04-BP03 收集並記錄資料庫效能指標

若要了解您的資料管理系統如何執行，請務必追蹤相關指標。這些指標將協助您優化資料管理資源，以確保符合您的工作負載要求，並確保您對工作負載如何執行有著清楚的概觀。使用工具、程式庫和系統來記錄與資料庫效能有關的效能測量值。

有一些與資料庫託管所在系統相關的指標 (例如，CPU、儲存體、記憶體、IOPS)，也有一些用於存取資料本身的指標 (例如，每秒交易數、查詢率、回應時間、錯誤)。任何支援或操作人員都應該可以立即存取這些指標，並且有足夠的歷史記錄能夠識別趨勢、異常和瓶頸。

預期成果： 若要監控資料庫工作負載的效能，您必須記錄一段時間的多個效能指標。這可讓您偵測異常，以及針對業務指標測量效能，以確保符合您的工作負載需求。

常用的反模式：

- 您只使用手動日誌檔案來搜尋指標。
- 您只會將指標發佈至您團隊所使用的內部工具，而且沒有工作負載的全貌。
- 您只會使用所選監控軟體記錄的預設指標。
- 您只會在有問題時檢閱指標。
- 您只會監控系統層級指標，而不會擷取資料存取或用量指標。

建立此最佳實務的優勢： 建立效能基準可協助您了解正常行為和工作負載的要求。異常模式可以更快地識別和偵錯，進而改善資料庫的效能和可靠性。可以設定資料庫容量，以確保最佳成本，而不會犧牲效能。

若未建立此最佳實務，暴露的風險等級： 高

- 無法區分異常與正常效能等級將讓您難以識別問題和做出決策。
- 可能未識別潛在的節省成本。
- 將不會識別可能導致可靠性或效能下降的成長模式。

實作指引

識別、收集、彙總資料庫相關指標，並使其相互關聯。指標應該同時包括支援資料庫的基礎系統和資料庫指標。基礎系統指標可能包括 CPU 使用率、記憶體、可用磁碟儲存、磁碟 I/O 和網路傳入和傳出指標，而資料庫指標可能包括每秒交易數、熱門查詢、平均查詢率、回應時間、索引使用情況、表格鎖定、查詢逾時，以及開啟的連線數目。此資料對於了解工作負載的執行方式，以及資料庫解決方案的使用方式至關重要。將這些指標納入資料驅動的方法，以調整和優化工作負載的資源。

實作步驟：

1. 務必要追蹤哪些資料庫指標？
    a. 監控 Amazon RDS 的指標
    b. 使用 Performance Insights 進行監控
    c. 增強型監控
    d. DynamoDB 指標
    e. 監控 DynamoDB DAX
    f. 監控 MemoryDB
    g. 監控 Amazon Redshift
    h. Timeseries 指標和維度參考
    i. Aurora 的叢集層級指標
    j. 監控 Amazon Keyspaces
    k. 監控 Amazon Neptune
2. 資料庫監控是否會受益於偵測操作異常效能問題的機器學習解決方案？
    a. Amazon DevOps Guru for Amazon RDS 可讓您查看效能問題，並做出更正動作的建議。
3. 您是否需要有關 SQL 使用情況的應用程式層級詳細資訊？
    a. AWS X-Ray 可以檢測至應用程式，以獲得見解並封裝所有資料點以進行單一查詢。
4. 您目前是否具有核准的記錄和監控解決方案？
    a. Amazon CloudWatch 可以收集架構中各種資源的指標。您還可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方解決方案來設定可指出何時超過閾值的警示。
5. 您是否已識別並設定資料保留策略，以符合安全和營運目標？
    a. CloudWatch 指標的預設資料保留
    b. CloudWatch Logs 的預設資料保留

實作計劃的工作量： 有一個 中 工作量，用來從所有資料庫資源識別、追蹤、收集、彙總指標，並使其相互關聯。

資源

相關文件：

- AWS 資料庫快取
- Amazon Athena 10 大效能秘訣
- Amazon Aurora 最佳實務
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳實務
- Amazon Redshift Spectrum 最佳實務
- Amazon Redshift 效能
- AWS 的雲端資料庫
- Amazon RDS Performance Insights

相關影片：

- AWS 專用資料庫 (DAT209-L)
- 探究 Amazon Aurora 儲存的奧秘：運作方式 (DAT309-R)
- 深入探討 Amazon DynamoDB：進階設計模式 (DAT403-R1)

相關範例：

- Level 100：使用 CloudWatch 儀表板進行監控
- AWS 資料集擷取指標收集架構
- Amazon RDS 監控研討會

PERF04-BP04 根據存取模式選擇資料儲存

根據工作負載的存取模式確定要使用的服務和技術。除了效能和規模等非功能性要求外，存取模式還嚴重影響資料庫和儲存解決方案的選擇。第一個維度是對交易、ACID 合規和一致性讀取的需求。並非每個資料庫都支持這些項目，並且大部分 NoSQL 資料庫都會提供最終一致性模型。第二個重要維度是寫

入和讀取隨時間和空間的分佈。全球分散式應用程式需要考慮流量模式、延遲和存取要求，以便識別最佳的儲存解決方案。選擇的第三個關鍵層面是查詢模式靈活性、隨機存取模式，以及一次性查詢。還必須斟酌圍繞文字和自然語言處理、時間系列和圖形的高度專業化查詢功能的考量。

預期成果： 已根據識別和記錄的資料存取模式選取資料儲存。這可能包括最常見的讀取、寫入和刪除查詢、對特定計算和彙總的需求、資料的複雜性、資料相依性，以及必要的一致性需求。

常用的反模式：

- 您只選取一個資料庫廠商來簡化操作管理。
- 您假設資料存取模式不會隨著時間改變。
- 您在應用程式中實作複雜的交易、回復和一致性邏輯。
- 資料庫設定為支援潛在的高流量爆增，這會導致資料庫資源大部分時間保持閒置狀態。
- 使用共用的資料庫進行交易和分析用途。

建立此最佳實務的優勢： 根據存取模式選取和優化您的資料儲存將有助於降低開發複雜性，並優化您的效能機會。了解何時使用讀取複本、全域表、資料分割和快取將協助您降低營運負擔，並根據您的工作負載需求進行擴展。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

識別並評估您的資料存取模式，以選取正確的儲存組態。每個資料庫解決方案都有設定和優化儲存解決方案的選項。使用收集的指標和記錄，並嘗試使用選項來找出最佳組態。使用下表根據每個資料庫服務檢閱儲存選項。

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
擴展儲存	可用來自動擴展所佈建儲存 IOPS	自動擴展。資料表在大小方	儲存自動擴展選項可用來擴	儲存是記憶體內，受制於執行個體	儲存自動擴展選項可用來自動擴展	設定記憶體內和磁性層的保留期間	自動擴增和縮減表格儲存	自動擴展。資料表在大小方

AWS 服務	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
	的儲存自動擴展選項也可以在利用佈建的 IOPS 儲存類型時，獨立於佈建的儲存進行擴展	面不受限制。	展佈建的儲存	類型或計數	佈建的儲存	(以天為單位)		面不受限制。

實作步驟：

1. 識別並記錄資料和流量的預期成長。
   a. Amazon RDS 和 Aurora 支援儲存自動擴增至記錄的限制。除此之外，考慮將較舊資料轉換到 Amazon S3 進行封存、彙總歷史資料進行分析，或透過碎片水平擴展。
   b. DynamoDB 和 Amazon S3 將自動擴展至近乎無限制的儲存容量。
   c. 可以手動調整在 EC2 上執行的 Amazon RDS 執行個體和資料庫大小，並且 EC2 執行個體可以在後來新增 EBS 容量以取得額外的儲存。
   d. 可以根據活動中的變更來變更執行個體類型。例如，您可以在測試時從較小的執行個體開始，然後在您開始接收服務的生產流量時擴展執行個體。Aurora Serverless V2 會自動擴展以回應負載中的變更。

1. 記錄有關正常和尖峰效能 (每秒交易數 TPS 和每秒查詢數 QPS) 和一致性 (ACID 和最終一致性) 的要求。
2. 記錄解決方案部署層面和資料庫存取要求 (全域、多可用區域、讀取複寫、多個寫入節點)

實作計劃的工作量： 如果您的資料管理解決方案沒有日誌或指標，則需要在識別和記錄資料存取模式之前完成該操作。一旦了解您的資料存取模式，選取並設定您的資料儲存是 低 工作量。

資源

相關文件：

- AWS 資料庫快取
- Amazon Athena 10 大效能秘訣
- Amazon Aurora 最佳實務
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳實務
- Amazon Redshift Spectrum 最佳實務
- Amazon Redshift 效能
- AWS 的雲端資料庫
- Amazon RDS 儲存類型

相關影片：

- AWS 專用資料庫 (DAT209-L)
- 探究 Amazon Aurora 儲存的奧秘：運作方式 (DAT309-R)
- 深入探討 Amazon DynamoDB：進階設計模式 (DAT403-R1)

相關範例：

- 在 AWS 上使用分散式負載測試進行試驗和測試

PERF04-BP05 根據存取模式和指標優化資料儲存

使用效能特性和存取模式來優化資料儲存或查詢，以實現最佳效能。測量索引編制、鍵值分佈、資料倉儲設計或快取策略此類的優化，會對系統效能或整體效率造成何種影響。

常用的反模式：

- 您只使用手動日誌檔案來搜尋指標。

- 您只將指標發佈到內部工具。

建立此最佳實務的優勢： 為了確保您符合工作負載所需的指標，您必須監控與讀取和寫入相關的資料庫效能指標。您可以使用這些資料，針對資料儲存層讀取和寫入來新增新的優化。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

根據指標和模式優化資料儲存：使用報告的指標來識別工作負載中任何效能不佳的區域，並優化您的資料庫元件。每個資料庫系統都有不同的效能特性要評估，例如建立索引的方式、快取的方式或在多個系統之間分發的方式。測量優化的影響。

資源

相關文件：

- AWS 資料庫快取
- Amazon Athena 10 大效能秘訣
- Amazon Aurora 最佳實務
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳實務
- Amazon Redshift Spectrum 最佳實務
- Amazon Redshift 效能
- AWS 的雲端資料庫
- 使用 DevOps Guru for RDS 分析效能異常
- DynamoDB 的讀取/寫入容量

相關影片：

- AWS 專用資料庫 (DAT209-L)
- 探究 Amazon Aurora 儲存的奧秘：運作方式 (DAT309-R)
- 深入探討 Amazon DynamoDB：進階設計模式 (DAT403-R1)

相關範例：

- Amazon DynamoDB 的實作實驗室

PERF 5 您如何設定聯網解決方案？

工作負載的最佳網路解決方案會根據延遲、輸送量需求、抖動和頻寬而有所不同。實體限制 (例如使用者或內部部署資源) 會決定位置選項。這些限制條件可以隨著節點或資源置放而位移。

最佳實務

- PERF05-BP01 了解聯網如何影響效能
- PERF05-BP02 評估可用的聯網功能
- PERF05-BP03 為混合式工作負載選擇適當大小的專用連線或 VPN
- PERF05-BP04 利用負載平衡和加密卸載
- PERF05-BP05 選擇網路通訊協定以提高效能
- PERF05-BP06 根據網路要求選擇工作負載的位置
- PERF05-BP07 根據指標優化網路組態

PERF05-BP01 了解聯網如何影響效能

分析並了解與網路相關的決策如何影響工作負載效能。網路為應用程式元件、雲端服務、邊緣網絡和內部部署資料之間的連線負責，因此可以高度地影響工作負載效能。除了工作負載效能外，使用者體驗也會受到網路延遲、頻寬、通訊協定、位置、網路擁塞、抖動、輸送量和路由規則的影響。

預期成果： 具有來自工作負載的聯網要求的記錄清單，包括延遲、封包大小、路由規則、通訊協定和支援的流量模式。檢閱可用的聯網解決方案，並識別哪個服務符合您的工作負載聯網特性。雲端型網路可以快速重建，因此隨著時間演進您的網路架構是改善效能達成效率的必要條件。

常用的反模式：

- 通過現有資料中心的所有流量流程。
- 您尚未了解實際用量要求，就建置過多的 Direct Connect 工作階段。
- 在定義聯網解決方案時，您未考慮工作負載特性和加密負擔。
- 您將內部部署概念和策略用於雲端中的聯網解決方案。

建立此最佳實務的優勢： 了解聯網如何影響工作負載效能協助您識別潛在的瓶頸、改善使用者體驗、提高可靠性，以及隨著工作負載的變更降低營運維護成本。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

識別工作負載的重要網路效能指標，並擷取其聯網特性。使用基準化分析或負載測試，定義並記錄屬於資料驅動方法的要求。使用這些資料來確定您的網路解決方案受限的地方，並檢查可以改善此工作負載的組態選項。了解雲端原生聯網功能和可用選項，以及它們如何根據要求影響您的工作負載效能。每個聯網功能都有優缺點，並且可以設定為符合您的工作負載特性，並根據您的需求進行擴展。

實作步驟：

1. 定義並記錄聯網效能需求：
   a. 包括網路延遲、頻寬、通訊協定、位置、流量模式 (峰值和頻率)、輸送量、加密、檢查，以及路由規則等指標
2. 擷取您的基礎聯網特性：
   a. VPC Flow Logs
   b. AWS Transit Gateway 指標
   c. AWS PrivateLink 指標
3. 擷取您的應用程式聯網特性：
   a. 彈性網路配接卡
   b. AWS App Mesh 指標
   c. Amazon API Gateway 指標
4. 擷取您的邊緣聯網特性：
   a. Amazon CloudFront 指標
   b. Amazon Route 53 指標
   c. AWS Global Accelerator 指標
5. 擷取您的混合聯網特性：
   a. Direct Connect 指標
   b. AWS 站點對站點 VPN 指標
   c. AWS Client VPN 指標
   d. AWS 雲端 WAN 指標
6. 擷取您的安全聯網特性：
   a. AWS Shield、WAF 和 Network Firewall 指標
7. 使用追蹤工具擷取端對端效能指標：

a. AWS X-Ray

b. Amazon CloudWatch RUM

8. 對網路效能進行基準測試：

a. 基準 網路輸送量：當執行個體位於同一 VPC 時，可能會影響 EC2 網路效能的一些因素。測量同一 VPC 中 EC2 Linux 執行個體之間的網路頻寬。

b. 執行 負載測試 嘗試使用聯網解決方案和選項

實作計劃的工作量： 有一個 中等 工作量，用來記錄工作負載聯網要求、選項和可用的解決方案。

資源

相關文件：

- Application Load Balancer
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway
- 轉換到 Amazon Route 53 中的 Latency Based Routing
- VPC 端點
- VPC Flow Logs

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)
- 改善應用程式的全球網路效能
- EC2 執行個體和效能優化最佳實務
- 優化 Amazon EC2 執行個體的網路效能
- 搭配 Well-Architected Framework 的聯網最佳實務和秘訣

- 大規模遷移中的 AWS 聯網最佳實務

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP02 評估可用的聯網功能

評估雲端中可能提升效能的聯網功能。透過測試、指標和分析來測量這些功能的影響。例如，利用可用的網路層級功能來降低延遲、封包遺失或抖動。

許多服務的功用是要改善效能，有些則通常是為了提供優化網路效能的功能。AWS Global Accelerator 和 Amazon CloudFront 之類的服務旨在改善效能，而其他大部分的服務則具有優化網路流量的產品功能。檢閱服務功能 (例如，EC2 執行個體網路功能、增強型聯網執行個體類型、經 Amazon EBS 優化的執行個體、Amazon S3 Transfer Acceleration 和 CloudFront)，以改善您的工作負載效能。

預期成果： 您已記載工作負載內的元件清查，並識別個別元件有哪些聯網組態有助於您達到效能要求。在評估聯網功能之後，您試驗並測量了效能指標，以識別如何使用您可用的功能。

常見的反模式：

- 您將所有的工作負載放在離總部最近的 AWS 區域 中，而不是使用者附近的 AWS 區域。
- 未能對您的工作負載效能進行基準測試，並根據其結果持續評估工作負載效能。
- 您未檢閱服務組態以找出效能改進選項。

建立此最佳實務的優勢： 評估所有服務功能和選項可提高工作負載效能、降低基礎架構成本、減少維護工作負載所需的人力，以及提升整體安全狀態。您可以使用全球 AWS 骨幹，以確保能為客戶提供最佳的聯網體驗。

未建立此最佳實務時的曝險等級： 高

實作指引

檢閱您可以使用哪些網路相關組態選項，及其對工作負載可能有何影響。了解這些選項如何與您的架構互動，以及這些選項對衡量效能與使用者感知效能的影響，對於效能最佳化至關重要。

實作步驟：

1. 建立工作負載元件清單。
   a. 使用下列工具來建置、管理及監控您的組織網路： AWS 雲端 WAN。
   b. 使用下列工具檢視您的網路： Network Manager。使用現有的組態管理資料庫 (CMDB) 工具或 AWS Config 之類的工具，建立工作負載的清查及其設定方式。
2. 如果這是現有的工作負載，請識別並記載效能指標的基準，並將重心放在瓶頸和有待改善的領域上。每個工作負載的效能相關聯網指標，都會隨著業務要求和工作負載特性而不同。一開始對您的工作負載而言，下列指標可能都是必須檢閱的：頻寬、延遲、封包遺失、抖動和重新傳輸。
3. 如果這是新的工作負載，請執行 負載測試 以找出效能瓶頸。
4. 對於您找出的效能瓶頸，請檢閱您解決方案的組態選項，以找出改善效能的機會。
5. 如果您不知道網路路徑或路由，請使用 Network Access Analyzer 加以識別。
6. 檢閱您的網路通訊協定，以進一步降低延遲。
   - PERF05-BP05 選擇網路通訊協定以提高效能
7. 如果您跨多個位置使用 AWS Site-to-Site VPN 連線至 AWS 區域，請檢閱 加速 Site-to-Site VPN 連線 以找出改善聯網效能的機會。
8. 如果您的工作負載流量分散於多個帳戶，請評估網路拓撲和服務以降低延遲。
   - 在連接多個帳戶時，評估 VPC 對等互連 和 AWS Transit Gateway 之間在操作和效能上的權衡。AWS Transit Gateway 支援 AWS Site-to-Site VPN 輸送量擴展至單一 IPsec 最大限制 以上 (使用多重路徑)。Amazon VPC 與 AWS Transit Gateway 之間的流量仍會在私有 AWS 網路上，且不會對網際網路公開。AWS Transit Gateway 會簡化所有 VPC 的互連方式，如此便可跨數千個 AWS 帳戶 並進入內部部署網路。使用下列工具在多個帳戶間共用您的 AWS Transit Gateway：Resource Access Manager。若要取得檢視全域網路流量的能力，請使用 Network Manager 以集中檢視您的網路指標。
9. 檢閱您的使用者位置，並儘可能拉近使用者與工作負載之間的距離。
   a. AWS Global Accelerator 是一項連網服務，可使用 Amazon Web Services 全域網路基礎架構將使用者流量效能提升最多達 60%。網際網路壅塞時，AWS Global Accelerator 會找出最佳的應用程式路徑，以持續保持低封包遺失率、低抖動和低延遲。此外也提供靜態 IP 位址，以方便在可用區域或 AWS 區域 之間移動端點，而無須更新 DNS 組態或變更面向用戶端的應用程式。
   b. Amazon CloudFront 可讓您改善全域的工作負載內容傳遞效能和延遲。CloudFront 具有超過 410 個散佈於全球的連接點，可快取您的內容並降低使用者經歷的延遲。
   c. Amazon Route 53 提供 以延遲為基礎的路由，地理位置路由，地理位置臨近性路由和 以 IP 為基礎的路由 等選項，可協助您提升全球對象的工作負載效能。請檢閱您的工作負載流量和使用者位置，找出能夠優化工作負載效能的路由選項。
10. 評估用來改善儲存 IOPS 的其他 Amazon S3 功能。

a. Amazon S3 Transfer Acceleration 功能讓外部使用者得以獲益於 CloudFront 的聯網優化，以將資料上傳到 Amazon S3。這樣就可以更輕易地從與 AWS 雲端 沒有專用連線的遠端位置輸送大量資料。

b. Amazon S3 多區域存取點 可將內容複寫至多個區域，並提供單一存取點以簡化工作負載。使用多區域存取點時，您可以使用可識別最低延遲儲存貯體的服務，來要求資料或將資料寫入 Amazon S3。

11請檢閱您的運算資源網路頻寬。

a. EC2 執行個體、容器和 Lambda 函數所使用的彈性網絡介面 (ENA) 會按個別流程受到限制。請檢閱您的置放群組以優化 EC2 聯網輸送量。若要避免發生個別流程的瓶頸，請將應用程式設計為使用多個流程。若要監控及檢視您的運算相關聯網指標，請使用 CloudWatch 指標 和 `ethtool`。`ethtool` 包含在 ENA 驅動程式中，會將可發佈為 自訂指標 的其他網路相關指標公開至 CloudWatch。

b. 較新的 EC2 執行個體可以利用增強型聯網。N 系列 EC2 執行個體(例如 `M5n` 和 `M5dn`) 利用第四代自訂 Nitro 卡，為單一執行個體提供高達 100 Gbps 的網路輸送量。相較於基礎的 `M5` 執行個體，這些執行個體提供 4 倍之多的網路頻寬和封包程序，非常適合網路密集型應用程式。

c. Amazon Elastic Network Adapters (ENA) 可提供進一步的優化，具體方法是為您在叢集置放群組內的執行個體提供更理想的 輸送量。

d. Elastic Fabric Adapter (EFA) 是 Amazon EC2 執行個體的網路介面，讓您能夠在 AWS 上大規模執行需要高階節點間通訊的工作負載。透過 EFA，使用訊息傳遞界面 (MPI) 的高效能運算 (HPC) 應用程式，以及使用 NVIDIA 集體通訊函式庫 (NCCL) 的機器學習 (ML) 應用程式可擴展到數千個 CPU 或 GPU。

e. 經 Amazon EBS 優化的 執行個體使用優化的設定堆疊，可提供更多專用容量以增加 Amazon EBS I/O。這項優化能盡量減少 Amazon EBS I/O 與傳出執行個體的其他流量之間的競用，使得 EBS 磁碟區具有最佳效能。

實作計劃的工作量：

若要建立此最佳實務，您必須了解目前有哪些工作負載元件選項會影響網路效能。收集元件、評估網路改進選項、試驗、實作和記載這些改進，是 低 至 中等 工作量。

資源

相關文件：

- Amazon EBS - 優化的執行個體
- Application Load Balancer

- Amazon EC2 執行個體網路頻寬
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- AWS Transit Gateway
- 轉換到 Amazon Route 53 中以延遲為基礎的路由
- VPC 端點
- VPC Flow Logs
- 建置雲端 CMDB
- 使用 AWS Transit Gateway 擴展 VPN 輸送量

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)
- AWS Global Accelerator

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP03 為混合式工作負載選擇適當大小的專用連線或 VPN

當需要公共網路連接 AWS 中的內部部署和雲端資源時，請確保您有足夠的頻寬來符合您的效能要求。預估混合工作負載的頻寬和延遲需求。這些數字將促進 AWS Direct Connect 或 VPN 端點的調整大小需求。

預期成果： 部署需要混合式網路連線能力的工作負載時，您會有多種連線組態選項可用，例如受管和非受管 VPN 或 Direct Connect。為每個工作負載選取適用的連線類型，同時確保在您的位置與雲端之間您有足夠的頻寬和加密要求。

常用的反模式：

- 您只針對網路加密要求評估 VPN 解決方案。
- 您未評估備份或並行連線能力選項。
- 您將預設組態用於路由器、通道和 BGP 工作階段。
- 您無法了解或識別所有工作負載要求 (加密、通訊協定、頻寬和流量需求)。

建立此最佳實務的優勢： 選取和設定適當大小的混合網路解決方案將提高工作負載的可靠性，並充分利用效能機會。藉由識別工作負載要求、提前規劃和評估混合解決方案，您將最大限度地減少昂貴的實體網路變更和營運負擔，同時延長上市時間。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

根據您的頻寬要求開發混合式聯網架構：預估混合式應用程式的頻寬和延遲要求。根據頻寬要求，單一 VPN 連線或 Direct Connect 連線可能不足，而且您必須架構混合式設定，以便啟用跨多個連線的流量負載平衡。由於其私用網路連線，可能需要 Direct Connect，提供更可預測和一致的效能。非常適合需要一致延遲且幾乎無抖動的生產工作負載。

AWS Direct Connect 提供與 AWS 環境的專用連線，速度範圍為 50 Mbps 到最高 10 Gbps。這可為您提供受管和受控的延遲以及佈建頻寬，因此您的工作負載可以輕鬆、高效地連線到其他環境。使用 AWS Direct Connect 合作夥伴之一，您可以從多個環境獲得端到端連線能力，從而為擴展的網路提供一致的效能。

AWS 站點對站點 VPN 是 VPC 的受管 VPN 服務。建立 VPN 連接時，AWS 會提供通道給兩個不同的 VPN 端點。透過 AWS Transit Gateway，您可以簡化多個 VPC 之間的連線，也可以使用單一 VPN 連線 AWS Transit Gateway 連接的任何 VPC。AWS Transit Gateway 還能透過多個 VPN 通道啟用同等成本的多路徑 (ECMP) 路由支援，讓您擴展到超過 1.25 Gbps IPsec VPN 的輸送量限制。

實作計劃的工作量： 有一個 高 工作量，用來評估混合網路的工作負載需求，以及實作混合聯網解決方案。

資源

相關文件：

- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway

- 轉換到 Amazon Route 53 中的 Latency Based Routing
- VPC 端點
- VPC Flow Logs
- 站點對站點 VPN
- 建置可擴展且安全的多 VPC AWS 網路基礎設施
- Direct Connect
- Client VPN

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)
- AWS Global Accelerator
- Direct Connect
- Transit Gateway Connect
- VPN 解決方案
- 使用 VPN 解決方案保護安全

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP04 利用負載平衡和加密卸載

在多個資源或服務之間分配流量，以讓您的工作負載能夠利用雲端提供的彈性。您也可以使用負載平衡來卸載加密終止，以提升效能及有效管理和路由流量。

在實作向外擴展架構時，如果想要使用多個執行個體來獲取服務內容，您可以利用 Amazon VPC 內部的負載平衡器。AWS 為您 ELB 服務中的應用程式提供了多種模型。Application Load Balancer 最適合 HTTP 和 HTTPS 流量的負載平衡，並提供了針對現代應用程式架構 (包括微型服務和容器) 交付的進階請求路由。

Network Load Balancer 最適合需要極高效能的 TCP 流量的負載平衡。它能夠每秒處理數百萬個請求，同時保持超低延遲性，並且還進行優化，可處理突發的和不穩定的流量模式。

Elastic Load Balancing 提供整合的憑證管理和 SSL/TLS 解密，讓您能夠靈活地集中管理負載平衡器的 SSL 設定，並從工作負載中卸載 CPU 密集型工作。

常用的反模式：

- 您可以透過現有的負載平衡器路由所有網際網路流量。
- 您可以使用一般 TCP 負載平衡，並讓每個運算節點處理 SSL 加密。

建立此最佳實務的優勢： 負載平衡器會處理單一可用區域中或橫跨多個可用區域的應用程式流量的不同負載。負載平衡器具備高可用性、自動調整規模，以及讓應用程式具備容錯能力所需的強大安全性。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

將適當的負載平衡器用於您的工作負載：為工作負載選取適當的負載平衡器。如果您必須平衡 HTTP 請求的負載，我們建議使用 Application Load Balancer。針對網路和傳輸通訊協定 (第 4 層 – TCP, UDP) 負載平衡，以及針對極端效能和低延遲應用程式，我們建議使用 Network Load Balancer。Application Load Balancers 支援 HTTPS，Network Load Balancer 支援 TLS 加密卸載。

啟用 HTTPS 或 TLS 加密卸載：Elastic Load Balancing 包括整合式憑證管理、使用者身份驗證和 SSL/TLS 解密。它提供集中管理 TLS 設定並從應用程式卸載 CPU 密集型工作負載的彈性。將所有 HTTPS 流量作為負載平衡器部署的一部分進行加密。

資源

相關文件：

- Amazon EBS - 優化的執行個體
- Application Load Balancer
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway
- 轉換到 Amazon Route 53 中的 Latency Based Routing

- VPC 端點
- VPC Flow Logs

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP05 選擇網路通訊協定以提高效能

根據對工作負載效能的影響，做出系統和網路間通訊協定的決策。

實現輸送量的延遲和頻寬之間存在關係。如果您的檔案傳輸使用 TCP，較高的延遲會降低整體傳輸量。有一些方法可以使用 TCP 調校和優化的傳輸通訊協定來解決這個問題，有些方法則會使用 UDP。

常用的反模式：

- 無論效能需求為何，您都可以將 TCP 用於所有工作負載。

建立此最佳實務的優勢： 為工作負載元件之間的通訊選擇適當的通訊協定，便可確保達到該工作負載的最佳效能。無連線 UDP 雖然達到高速，但卻失去重新傳輸能力或高可靠性。TCP 是功能完整的通訊協定，但需要更大的額外負荷來處理封包。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

優化網路流量：選擇適當的通訊協定，以最佳化工作負載的效能。實現輸送量的延遲和頻寬之間存在關係。如果您的檔案傳輸使用 TCP，較高的延遲會降低整體輸送量。有一些方法可以使用 TCP 調校和優化的傳輸通訊協定來解決延遲，有些方法則會使用 UDP。

資源

相關文件：

- Amazon EBS - 優化的執行個體
- Application Load Balancer
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway
- 轉換到 Amazon Route 53 中的 Latency Based Routing
- VPC 端點
- VPC Flow Logs

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP06 根據網路要求選擇工作負載的位置

使用可用的雲端位置選項來降低網路延遲或提高輸送量。使用 AWS 區域、可用區域、置放群組和邊緣節點 (例如 AWS Outposts、AWS Local Zones 和 AWS Wavelength) 來降低網路延遲或改善輸送量。

AWS 雲端基礎架構是以區域與可用區域為中心所建置。區域是世界上有多個可用區域的實體位置。

可用區域由一或多個分散的資料中心所組成，每個都有備援電源、聯網和連線能力，且置放在不同的機構。這些可用區域讓您能夠運作生產應用程式和資料庫，它們比單一資料中心具有更高的可用性、容錯力和可擴展性

根據下列關鍵元素，為您的部署選擇適當的一個或多個區域：

- 使用者所在的位置：選擇靠近工作負載使用者的區域可確保降低其使用工作負載時的延遲。
- 資料所在位置：對於資料密集型應用程式，資料傳輸是延遲的主要瓶頸。應用程式程式碼應盡可能接近資料予以執行。
- 其他限制：請考慮安全性和合規性等限制。

Amazon EC2 提供了適用於聯網的置放群組。置放群組是執行個體的邏輯分組，可降低延遲或提高可靠性。使用搭配支援的執行個體類型以及彈性網路轉接器 (ENA) 的置放群組，可讓工作負載參與低延遲 25 Gbps 的網路。建議將置放群組用於受益於低網路延遲、高網路輸送量或兩者兼而有之的工作負載。使用置放群組的益處是可以降低網路通訊中的抖動。

使用邊緣位置的全球網路可在邊緣交付對延遲敏感的服務。這些節點通常可提供諸如內容交付網路 (CDN) 和網域名稱系統 (DNS) 之類的服務。透過將這些服務置於邊緣，工作負載可以低延遲回應內容或 DNS 解決的請求。這些服務還提供地理服務，例如內容的地理定位 (根據最終使用者的位置提供不同的內容)，或 Latency-Based Routing，將最終使用者定向到最近區域的 (最小延遲)。

Amazon CloudFront 是全域 CDN，可用於加速靜態內容 (例如影響、指令碼和影片) 以及動態內容 (例如 API 或 Web 應用程式)。它依賴於節點的全球網路，該網路會快取內容並為我們的使用者提供高效能的網路連線能力。CloudFront 還可加快許多其他功能的速度，例如內容上傳和動態應用程式，使其成為所有透過網際網路提供流量的應用程式的效能補充。 Lambda@Edge 是 Amazon CloudFront 的一項功能，可讓您更接近工作負載的使用者執行程式碼，藉此提升效能並減少延遲。

Amazon Route 53 是一項高可用性、可擴展的雲端 DNS Web 服務。該服務旨在為開發人員和企業提供一種非常可靠且經濟實惠的方式，將名稱 (如 www.example.com) 轉換為電腦用來互相連線的數字 IP 地址 (如 192.168.2.1)，將最終使用者路由到網際網路應用程式。Route 53 可與 IPv6 完全相容。

AWS Outposts 是專為因延遲需求而需要保持內部部署的工作負載所設計，而您希望該工作負載能夠與 AWS 中的其他工作負載無縫執行。AWS Outposts 是利用 AWS 設計的硬體所建置的全受管和可設定的運算和儲存機架，讓您能夠在內部部署執行運算和儲存，同時無縫連線到雲端中的 AWS 各種服務。

AWS Local Zones 專為執行需要低於十毫秒延遲的工作負載 (例如影片轉譯和圖形密集型虛擬桌面應用程式) 所設計。Local Zones 可讓您取得所有優勢，讓運算和儲存資源更接近最終使用者。

AWS Wavelength 的設計目的是透過將 AWS 基礎設施、服務、API 和工具擴展到 5G 網路，將超低延遲應用程式交付到 5G 裝置。Wavelength 將儲存和運算嵌入到電信供應商 5G 網路內，可在您的 5G 工作負載需要低於十毫秒的延遲時協助您，例如 IoT 裝置、遊戲串流、自動駕駛車輛和即時媒體製作。

使用邊緣服務來減少延遲及啟用內容快取。確保您已為 DNS 和 HTTP/HTTPS 正確設定了快取控制，才能從這些方法中獲得最大的收益。

常用的反模式：

- 您可以將所有工作負載資源合併到單一地理位置。
- 您選擇的區域最接近您的位置，但不是最接近工作負載最終使用者。

建立此最佳實務的優勢： 無論您想要觸及哪一處的客戶，您必須確保網路都能使用。使用 AWS 的私有全球網路將工作負載部署到離客戶最近的位置，即可確保客戶享有最低的延遲體驗。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

選取正確的位置來減少延遲：識別使用者和資料的位置。利用 AWS 區域、可用區域、置放群組和節點來減少延遲。

資源

相關文件：

- Amazon EBS - 優化的執行個體
- Application Load Balancer
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway
- 轉換到 Amazon Route 53 中的 Latency Based Routing
- VPC 端點
- VPC Flow Logs

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF05-BP07 根據指標優化網路組態

使用收集和分析的資料來做出有關優化網路組態的明智決策。測量這些變更的影響，並利用這些測量結果來做出未來的決策。

為工作負載使用的所有 VPC 網路啟用 VPC Flow Logs。VPC Flow Logs 功能可以擷取有關往返 VPC 網路接口 IP 流量的資訊。VPC Flow Logs 可協助您執行多項任務，例如針對特定流量無法到達執行個體的問題進行疑難排解，進而協助您診斷過於嚴格的安全群組規則。您可以使用 Flow Logs 做為安全工具，來監控到達執行個體的流量、分析網路流量，以及尋找異常的流量行為。

隨著工作負載的演進，使用聯網指標變更聯網組態。雲端型網路可以快速重建，因此隨著時間演進您的網路架構是維持效能達成效率的必要條件。

常用的反模式：

- 您假設所有效能相關問題都與應用程式有關。
- 您只能從靠近已部署工作負載的位置測試網路效能。

建立此最佳實務的優勢：若要確保符合工作負載所需的指標，您必須監控網路效能指標。您可以擷取往返 VPC 中網路界面的 IP 流量資訊，並使用此資料新增最佳化項目，或將工作負載部署到新的地理區域。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

啟用 VPC Flow Logs：VPC Flow Logs 讓您可以擷取有關往返 VPC 網路界面 IP 流量的資訊。VPC Flow Logs 可協助您執行多項任務，例如針對特定流量無法到達執行個體的問題進行疑難排解，進而協助您診斷過於嚴格的安全群組規則。您可以使用 Flow Logs 做為安全工具，來監控到達執行個體的流量、分析網路流量，以及尋找異常的流量行為。

為網路選項啟用適當的指標：確保您為工作負載選擇適當的網路指標。您可以為 VPC NAT 閘道、傳輸閘道和 VPN 通道啟用指標。

資源

相關文件：

- Amazon EBS - 優化的執行個體
- Application Load Balancer
- Linux 上的 EC2 增強型聯網
- Windows 上的 EC2 增強型聯網
- EC2 置放群組
- 在 Linux 執行個體上啟用搭配彈性網路轉接器 (ENA) 的增強型聯網
- Network Load Balancer
- AWS 的聯網產品
- Transit Gateway
- 轉換到 Amazon Route 53 中的 Latency Based Routing
- VPC 端點
- VPC Flow Logs
- 使用 Amazon Cloudwatch 指標監控您的全球和核心網路
- 持續監控網路流量和資源

相關影片：

- 與 AWS 和混合 AWS 網路架構的連線 (NET317-R1)
- 優化 Amazon EC2 執行個體的網路效能 (CMP308-R1)
- 監控網路流量並對其進行疑難排解
- 使用 Amazon VPC Traffic Mirroring 簡化流量監控和可見性

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會
- AWS 網路監控

檢閱

問題

- PERF 6 您如何發展工作負載，以運用新版本的優勢？

PERF 6 您如何發展工作負載，以運用新版本的優勢？

架構工作負載時，可選擇的選項有限。但一段時間後會有可改善工作負載效能的新技術和方法推出。

最佳實務

- PERF06-BP01 掌握最新的資源和服務
- PERF06-BP02 定義提高工作負載效能的程序
- PERF06-BP03 隨時間提升工作負載效能

PERF06-BP01 掌握最新的資源和服務

當新服務、設計模式和產品供應項目推出時，評估提升效能的方法。透過評估、內部討論或外部分析，確定哪些方法可以提高工作負載效能或效率。

定義程序來評估與工作負載相關的更新、新功能和服務。例如，建立使用新技術的概念證明或與內部小組協商。嘗試新的想法或服務時，執行效能測試以衡量其對工作負載效能的影響。使用基礎設施即程式碼 (IaC) 和 DevOps 文化，運用能力，以最少的成本和風險來頻繁測試新想法或技術。

預期成果： 您已記錄組成部分、設計模式和工作負載特性的清單。您使用該文件來建立訂閱清單，以便向團隊通知服務的最新狀態、功能和新產品。您已確定組成部分的利害關係人，他們將會評估新版本並針對業務影響和優先要務提供建議。

常見的反模式：

- 當工作負載不符合效能要求時，只審查新的選項和服務。
- 您假設所有新產品供應項目對於工作負載都沒有幫助。
- 改善工作負載時，您總是選擇建置而不是購買。

建立此最佳實務的優勢： 透過考量新服務或產品供應項目，您可以改善工作負載的效能和效率、降低基礎設施的成本，以及減少維護服務所需的工作量。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

定義程序來評估來自 AWS 的更新、新功能和服務。例如，建立使用新技術的概念證明。嘗試新的想法或服務時，執行效能測試以衡量對工作負載效率或效能的影響。利用您在 AWS 中具備的靈活性，以最少的成本或風險頻繁測試新的想法或技術。

實作步驟

1. 記錄工作負載解決方案。使用組態管理資料庫 (CMDB) 解決方案，來記錄清單並分類服務和相依性。使用 AWS Config 之類的工具，來取得在工作負載使用中 AWS 中的所有服務清單。
2. 使用 標記策略 來記錄每個工作負載組成部分和類別的擁有者。例如，如果您目前使用 Amazon RDS 作為資料庫解決方案，請將資料庫管理員 (DBA) 指派為評估和研究新服務與更新的擁有者，並加以記錄。
3. 找出與工作負載組成部分相關的新聞和更新來源。在先前所提及的 Amazon RDS 範例中，類別擁有者應訂閱 AWS 部落格的最新消息 來了解與其工作負載組成部分相符的產品。您可以訂閱 RSS 摘要或管理 電子郵件訂閱。監控對您使用的 Amazon RDS 資料庫所做的升級、導入的功能、發行的執行個體，以及 Amazon Aurora Serverless 之類新產品。掌握產業部落格、產品和生產組成部分的廠商動態。
4. 記錄您在評估更新和新服務的過程。向類別擁有者提供所需的時間和空間，來研究、測試、實驗和驗證更新及新服務。回顧所記錄的業務需求和 KPI，來協助排定哪個更新將帶來正面業務影響的優先順序。

實作計劃的工作量： 若要建立此最佳實務，您必須了解目前的工作負載組成部分、找出類別擁有者以及找出服務更新的來源。此作業所需的工作量不會很多，但會是一個持續的過程，可能會隨著時間發展和改善。

資源

相關文件：

- AWS 部落格
- AWS 最新消息

相關影片：

- AWS 活動 YouTube 頻道
- AWS 線上技術會談 YouTube 頻道

- Amazon Web Services YouTube 頻道

相關範例：

- AWS Github
- AWS Skill Builder

PERF06-BP02 定義提高工作負載效能的程序

定義一個程序，以在新的服務、設計模式、資源類型和組態可用時對其進行評估。例如，對新的執行個體方案執行現有的效能測試，以判斷其是否可能改善工作負載。

工作負載的效能有一些關鍵限制。記錄這些內容，以便您知道哪種創新可以改善工作負載的效能。當新服務或技術可用時，請使用此資訊來找出緩解限制或瓶頸的方法。

常用的反模式：

- 您假設您目前的架構將變成靜態，而且永遠不會隨著時間更新。
- 您會隨時間導入架構變更，而且無須指標佐證。

建立此最佳實務的優勢： 定義進行架構變更的程序後，即可啟用收集的資料，以隨著時間影響工作負載。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

識別工作負載的關鍵效能限制：記錄工作負載的效能限制，讓您知道哪些類型的創新可能會改善工作負載的效能。

資源

相關文件：

- AWS 部落格
- AWS 最新消息

相關影片：

- AWS 活動 YouTube 頻道
- AWS 線上技術會談 YouTube 頻道
- Amazon Web Services YouTube 頻道

相關範例：

- AWS Github
- AWS Skill Builder

PERF06-BP03 隨時間提升工作負載效能

作為一個組織，使用評估過程中收集的資訊，在新服務或資源可用時主動推動採用。

在評估新的服務或技術以推動變更時，好好利用您收集的資訊。隨著業務或工作負載變化，效能需求也隨之變化。利用從工作負載指標收集的資料，來評估可獲得最大效率或效能效益的區域，並主動採用新的服務和技術來滿足需求。

常用的反模式：

- 您假設您目前的架構將變成靜態，而且永遠不會隨著時間更新。
- 您會隨時間導入架構變更，而且無須指標佐證。
- 您會變更架構的原因就只是業界內的每個人都是採用此做法。

建立此最佳實務的優勢： 若要最佳化工作負載效能和成本，您必須評估所有可用的軟體和服務，以判斷適合工作負載的軟體和服務。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

隨時間發展您的工作負載：在評估新的服務或技術以推動變更時，好好利用您收集的資訊。隨著業務或工作負載變化，效能需求也隨之變化。利用從工作負載指標收集的資料，來評估可達成最大效率或效能效益的區域，並主動採用新的服務和技術來滿足需求。

資源

相關文件：

- AWS 部落格

- AWS 最新消息

相關影片：

- AWS 活動 YouTube 頻道
- AWS 線上技術會談 YouTube 頻道
- Amazon Web Services YouTube 頻道

相關範例：

- AWS Github
- AWS Skill Builder

監控

問題

- PERF 7 您如何監控資源來確保達成預期效能？

PERF 7 您如何監控資源來確保達成預期效能？

系統效能可能會隨時間降低。監控系統效能以識別效能降低情況，並修復內部或外部因素，如作業系統或應用程式負載。

最佳實務

- PERF07-BP01 記錄效能相關指標
- PERF07-BP02 分析事件或事故發生時的指標
- PERF07-BP03 建立用於測量工作負載效能的關鍵績效指標 (KPI)
- PERF07-BP04 使用監控來產生警示型通知
- PERF07-BP05 定期審查指標
- PERF07-BP06 主動監控和警示

PERF07-BP01 記錄效能相關指標

使用監控和可觀察性服務來記錄效能相關指標。指標範例包括記錄資料庫交易、慢速查詢、I/O 延遲、HTTP 請求輸送量、服務延遲或其他關鍵資料。

確定對您的工作負載重要的效能指標並進行記錄。此資料是能夠識別哪些元件會影響整體效能或工作負載效率的重要部分。

從客戶體驗出發，確定重要指標。對於每個指標，確定目標、測量方法和優先級。使用它們來建置警示和通知，以主動解決與效能相關的問題。

常用的反模式：

- 您只需監控作業系統層級指標，以深入了解工作負載。
- 您會架構運算需求，以滿足尖峰工作負載要求。

建立此最佳實務的優勢： 若要最佳化效能和資源利用率，您需要取得關鍵績效指標的整合操作檢視。您可以建立儀表板，並對資料進行指標計算，以獲得操作和使用率的洞見。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

確定與您的工作負載相關的效能指標並進行記錄。此資料有助於識別哪些元件會影響整體效能或工作負載效率。

識別效能指標：使用客戶體驗來識別最重要的指標。對於每個指標，確定目標、測量方法和優先級。使用這些資料點來建置警示和通知，以主動解決與效能相關的問題。

資源

相關文件：

- CloudWatch 文件
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- 發佈自訂指標
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件
- Amazon CloudWatch RUM

相關影片：

- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)

- AWS 上的應用程式效能管理
- 制定監控計劃

相關範例：

- Level 100：使用 CloudWatch 儀表板進行監控
- Level 100：使用 CloudWatch 儀表板監控 Windows EC2 執行個體
- Level 100：使用 CloudWatch 儀表板監控 Amazon Linux EC2 執行個體

PERF07-BP02 分析事件或事故發生時的指標

為回應事件或事故 (或在事件或事故期間)，使用監控儀表板或報告來了解和診斷影響。這些檢視可讓您深入了解工作負載的哪些部分未如預期執行。

在為架構編寫關鍵使用者案例時，應包括效能需求，例如指定每個關鍵案例應執行的速度。對於這些關鍵案例，實作額外執行指令碼的使用者旅程，以確保您了解這些案例會如何根據您的要求予以執行。

常用的反模式：

- 您假設效能事件是一次性問題，而且只與異常有關。
- 只有在回應效能事件時，才會評估現有效能指標。

建立此最佳實務的優勢： 在判斷工作負載是否如預期效能運作時，您必須收集其他指標資料來分析，以回應效能事件。此資料用於了解效能事件的影響，並建議提升工作負載效能的變更。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

優先考慮關鍵使用者案例的體驗問題：在為架構編寫關鍵使用者案例時，應包括效能需求，例如指定每個關鍵案例應執行的速度。對於這些關鍵案例，實作額外執行指令碼的使用者之旅，以確保您了解這些使用者案例會如何根據您的要求予以執行。

資源

相關文件：

- CloudWatch 文件

- Amazon CloudWatch Synthetics
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件

相關影片：

- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)
- 透過 Amazon CloudWatch RUM 優化應用程式
- Amazon CloudWatch Synthetics 的示範

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端

PERF07-BP03 建立用於測量工作負載效能的關鍵績效指標 (KPI)

識別定量和定性衡量工作負載效能的 KPI。KPI 有助於測量工作負載的運作狀態，因為其與業務目標相關。KPI 允許業務和工程團隊在衡量目標和策略時，以及如何將其結合以產生業務成果方面保持一致。當業務目標、策略或最終使用者要求變更時，應該重新檢視 KPI。

例如，網站工作負載可能使用頁面載入時間，做為整體效能的指示。此指標將是衡量最終使用者體驗的多個資料點之一。除了識別頁面載入時間閾值外，您還應該記錄未符合效能時預期的成果或業務風險。頁面載入時間若很長，將會直接影響您的最終使用者、降低其使用者體驗評分，並可能導致客戶流失。當您定義 KPI 閾值時，請同時結合業界基準和最終使用者期望。例如，如果目前業界基準是網頁在兩秒內載入，但您的最終使用者期望網頁在一秒內載入，則您在建立 KPI 時應將這兩個資料點列入考慮。另一個 KPI 範例可能專注於符合內部效能需求。可能根據在產生了生產資料後的一個工作日內產生銷售報告來建立 KPI 閾值。這些報告可能直接影響每日決策和業務成果。

預期成果： 建立 KPI 涉及不同的部門和利害關係人。您的團隊必須使用即時精密資料和歷史資料來評估您的工作負載 KPI，以供參考，並建立儀表板，針對您的 KPI 資料執行指標數學，以衍生營運和使用率見解。KPI 應該加以記錄，說明已同意支援業務目標和策略的 KPI 和閾值，以及對應到受監控的指標。KPI 會識別效能要求，刻意進行審查，以及經常與所有團隊分享並使其理解。清楚地識別風險和取捨，並了解未符合 KPI 閾值時業務會受到何種影響。

常用的反模式：

- 您只監控系統層級指標，以洞悉工作負載，但不了解對這些指標的業務影響。
- 您假設 KPI 已發佈，並做為標準指標資料分享。
- 定義 KPI 但未與所有團隊分享它們。
- 未定義一個量化的可衡量 KPI。
- 未使 KPI 與業務目標或策略保持一致。

建立此最佳實務的優勢： 識別代表工作負載運作狀態的特定指標有助於使團隊在其優先事項上保持一致，並定義成功的業務成果。與所有部門分享這些指標可對閾值、期望和業務影響提供可見性和一致性。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

受工作負載運成狀態影響的所有部門和業務團隊都應為定義 KPI 做出貢獻。單一人員應該推動協同合作、時間軸、文件，以及與組織 KPI 相關的資訊。這個單一執行緒擁有者通常會分享業務目標和策略，並指派利害關係人任務，在其各自部門中建立 KPI。一旦定義了 KPI，營運團隊通常就會協助定義將支援和告知不同 KPI 成功的指標。僅當支援工作負載的所有團隊成員都意識到 KPI 時，KPI 才有效。

實作步驟

1. 識別並記錄利害關係人。
2. 識別公司目標和策略。
3. 審查符合貴公司目標和策略的常見業界 KPI。
4. 審查工作負載的最終使用者期望。
5. 定義並記錄支援公司目標和策略的 KPI。
6. 識別並記錄核准的取捨策略以符合 KPI。
7. 識別並記錄將告知 KPI 的指標。
8. 識別並記錄嚴重性或警示等級的 KPI 閾值。
9. 識別並記錄未符合 KPI 時的風險和影響。
10識別每個 KPI 的審查頻率。
11與支援工作負載的所有團隊交流 KPI 文件。

實作指引的工作量： 定義和交流 KPI 是 低 工作量。這通常可以透過在幾週內與業務利害關係人會面、審查目標、策略和工作負載指標來完成。

資源

相關文件：

- CloudWatch 文件
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件
- 使用 Amazon CloudWatch 儀表板
- Amazon QuickSight KPI

相關影片：

- AWS re:Invent 2019：擴充至首個 1,000 萬名使用者 (ARC211-R)
- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)
- 制定監控計劃

相關範例：

- 使用 Amazon QuickSight 建立儀表板

PERF07-BP04 使用監控來產生警示型通知

使用監控系統和您定義的效能相關關鍵績效指標 (KPI)，當這些測量結果超出預期範圍時自動產生警示。

Amazon CloudWatch 可以收集架構中各種資源的指標。您還可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方監控服務來設定警示，以在超過閾值時進行指示 — 指標超出預期邊界時發出警示。

常用的反模式：

- 您需要靠員工監控指標，並在他們發現問題時做出反應。
- 當可以觸發無伺服器工作流程來完成相同的任務時，您僅倚賴操作執行手冊作業。

建立此最佳實務的優勢： 您可以根據預先定義的閾值或機器學習演算法，來設定提醒並自動化動作，確定指標中的異常行為。這些警示也可以觸發無伺服器的工作流程，藉以修改工作負載的效能特性 (例如，增加運算容量、修改資料庫組態)。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

監控指標：Amazon CloudWatch 可以收集架構中各種資源的指標。您可以收集和發佈自訂指標以顯示業務或衍生指標。使用 CloudWatch 或第三方監控服務設定警示，藉以指出何時超出閾值。

資源

相關文件：

- CloudWatch 文件
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件
- 使用 CloudWatch 中的警示和警示動作

相關影片：

- AWS re:Invent 2019：擴充至首個 1,000 萬名使用者 (ARC211-R)
- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)
- 制定監控計劃
- 搭配 Amazon CloudWatch Events 使用 AWS Lambda

相關範例：

- Cloudwatch Logs 自訂警示

PERF07-BP05 定期審查指標

作為日常維護或對事件或事故的回應，審查收集了哪些指標。透過這些審查來確定哪些指標是解決問題的關鍵，以及哪些其他指標 (如果被追蹤) 將有助於識別、解決或預防問題。

作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，從而可以防止事故發生或更快地解決將來的事故。

常用的反模式：

- 您讓指標長時間持續處於警示狀態。
- 您建立自動化系統無法採取行動的警示。

建立此最佳實務的優勢： 持續審查正在收集的指標，以確保指標正確識別、處理或防止問題發生。如果讓指標長時間持續處於警示狀態，指標也會變得過時。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

不斷改進指標收集和監控：作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，從而可以防止事故發生或更快地解決將來的事故。

資源

相關文件：

- CloudWatch 文件
- 使用 CloudWatch Agent 從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件

相關影片：

- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)
- AWS 上的應用程式效能管理
- 制定監控計劃

相關範例：

- 使用 Amazon QuickSight 建立儀表板
- Level 100：使用 CloudWatch 儀表板進行監控

PERF07-BP06 主動監控和警示

使用關鍵績效指標 (KPI) 搭配監控和提醒系統，主動處理效能相關的問題。使用警示觸發自動化動作，盡可能修復問題。如果無法自動回應，則將警示上報給能夠回應的人員。例如，您可能有一個可以預測關鍵績效指標 (KPI) 預期值並在超過特定閾值時發出警示的系統，或者在 KPI 超出預期值時可以自動停止或回復部署的工具。

實作可在工作負載執行時提供效能可見度的程序。建置監控儀表板並建立效能預期的基準規範，以確定工作負載是否以最佳狀態執行。

常用的反模式：

- 您只讓操作人員有能力對工作負載進行操作變更。
- 您讓所有警示篩選到操作團隊，無須主動修復。

建立此最佳實務的優勢： 主動修復警示動作能夠讓支援人員專注在無法自動採取行動的項目上。如此可確保操作人員不會負荷所有警報，而僅專注於關鍵警報。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在營運過程中監控效能：實作可在工作負載執行時提供效能可見度的程序。建立監控儀表板，並建立效能期望的基準。

資源

相關文件：

- CloudWatch 文件
- 監控、記錄和效能 APN 合作夥伴
- X-Ray 文件
- 使用 CloudWatch 中的警示和警示動作

相關影片：

- 突破混沌的難題：掌握運作相關的情況和洞見 (MGT301-R1)
- AWS 上的應用程式效能管理
- 制定監控計劃

- 搭配 Amazon CloudWatch Events 使用 AWS Lambda

相關範例：

- Cloudwatch Logs 自訂警示

權衡

問題

- PERF 8 您如何採用權衡來增進效能？

PERF 8 您如何採用權衡來增進效能？

架構解決方案時，判斷權衡項目可讓您選擇最佳方法。您通常可以透過權衡一致性、耐用性和時間與延遲的空間來提升效能。

最佳實務

- PERF08-BP01 了解效能至關重要的領域
- PERF08-BP02 了解設計模式和服務
- PERF08-BP03 確定權衡如何影響客戶和效率：
- PERF08-BP04 衡量效能改進的影響
- PERF08-BP05 使用各種與效能相關的策略

PERF08-BP01 了解效能至關重要的領域

了解並找出提高工作負載效能將對效率或客戶體驗產生正面影響的地方。例如，具有大量客戶互動的網站可受益於邊緣服務的使用，因為這樣可以將內容交付移至更接近客戶的地方。

預期成果： 透過了解架構、流量模式和資料存取模式，來提高效能效率，並確定延遲和處理時間。找出隨著工作負載的成長，可能會影響客戶體驗的潛在瓶頸。當您已確定這些面向時，請審視自己可以部署哪個解決方案，來消除這些效能疑慮。

常見的反模式：

- 您假設標準運算指標 (例如， `CPUUtilization` 或記憶體壓力) 足以揪出效能問題。
- 您只會使用所選監控軟體記錄的預設指標。

- 您只會在有問題時審查指標。

建立此最佳實務的優勢： 了解效能的關鍵領域，有助於工作負載擁有者監控 KPI 和優先處理具有高影響力的待改善之處。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

設置端到端追蹤，以找出流量模式、延遲和關鍵的效能區域。監控資料存取模式是否有緩慢查詢或分段和分區不佳的資料。使用負載測試或監控來找出工作負載受限面向。

實作步驟

1. 設置端到端監控，來擷取所有工作負載組成部分和指標。
   - 使用 Amazon CloudWatch 實際使用者監控 (RUM) 來擷取來自實際使用者用戶端和前端工作階段的應用程式效能指標。
   - 設置 AWS X-Ray 來透過應用程式層追蹤流量，並找出組成部分和相依性之間的延遲。使用 X-Ray 服務地圖，來查看工作負載組成部分之間的關係和延遲。
   - 使用 Amazon Relational Database Service 績效詳情 來檢視資料庫效能指標並找出效能待改善之處。
   - 使用 Amazon RDS 增強型監控 來檢視資料庫 OS 效能指標。
   - 收集 每個工作負載組成部分和服務的 CloudWatch 指標， 並找出哪些指標會影響效能效率。
   - 設置 Amazon DevOps Guru 以取得其他績效詳情和建議
2. 執行測試，來產生指標、確定流量模式、瓶頸和關鍵效能區域。
   - 設置 CloudWatch Synthetic Canaries 使用 cron 任務或速率表達式，以程式設計的方式 ########## 以產生長期一致的指標。
   - 使用 AWS 分散式負載測試 解決方案，來產生尖峰流量或以預期成長速率測試工作負載。
3. 評估指標和遙測，來找出關鍵的效能領域。與團隊檢視這些領域，討論監控和解決方案，來避免瓶頸。
4. 進行效能改善的實驗，並透過資料來衡量這些變更。
   - 使用 CloudWatch Evidently 來測試新的改善之處以及對工作負載的效能影響。

實作計劃的工作量： 若要建立此最佳實務，您必須檢視端到端指標，並了解目前工作負載的效能。您需要投入適當的心力，來設置端到端監控並找出關鍵的效能領域。

資源

相關文件：

- Amazon 建置者資料中心
- X-Ray 文件
- Amazon CloudWatch RUM
- Amazon DevOps Guru
- CloudWatch RUM 和 X-Ray

相關影片：

- Amazon 建置者資料中心簡介 (DOP328)
- Amazon CloudWatch Synthetics 的示範

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端
- X-Ray SDK for Node.js
- X-Ray SDK for Python
- X-Ray SDK for Java
- X-Ray SDK for .Net
- X-Ray SDK for Ruby
- X-Ray 常駐程式
- AWS 上的分散式負載測試

PERF08-BP02 了解設計模式和服務

研究並了解有助於提高工作負載效能的各種設計模式和服務。作為分析的一部分，確定您為了實現更高效能而可能付出的代價。例如，使用快取服務可以協助減少資料庫系統所承擔的負載。但快取可能帶來最終一致性，這必須投入工程方面的努力，以期在業務要求與客戶期望內實現。

預期成果： 研究設計模式有助於您選擇可支援最佳效能系統的架構設計。了解您可以使用哪些效能組態選項，以及它們如何影響工作負載。優化工作負載的效能取決於了解這些選項如何與您的架構互動，以及這些選項對衡量效能與使用者感知效能的影響。

常見的反模式：

- 您假設所有傳統 IT 工作負載效能策略皆最適合雲端工作負載。
- 您會建置並管理快取解決方案，而非使用受管服務。
- 您對所有的工作負載使用相同的設計模式，而未評估何種模式可改善工作負載效能。

建立此最佳實務的優勢： 為您的工作負載選取正確的設計模式和服務，將可優化效能，進而帶動卓越營運並提升可靠性。正確的設計模式將符合您目前的工作負載特性，並協助您就未來的成長或變化進行擴展。

未建立此最佳實務時的曝險等級： 高

實作指引

了解您可以使用哪些效能組態選項，以及它們如何影響工作負載。最佳化工作負載的效能取決於了解這些選項如何與您的架構互動，以及這些選項對衡量效能與使用者感知效能的影響。

實作步驟：

1. 評估及檢閱將可改善工作負載效能的設計模式。
    a. AWS Well-Architected Amazon Builders' Library 為您提供 Amazon 如何建置和操作技術的詳細說明。這些文章由 Amazon 的資深工程師撰寫，涵蓋了架構、軟體交付和操作等主題。
    b. AWS 解決方案程式庫 是可供部署的解決方案集合，其中結合了服務、程式碼和組態。這些解決方案由 AWS 和 AWS 合作夥伴所建立，其基礎為常見的使用案例，以及按產業或工作負載類型分組的設計模式。例如，您可以為工作負載設定 分散式負載測試解決方案 。
    c. AWS 架構中心 提供按設計模式、內容類型與技術分組的參考架構圖。
    d. AWS 範例 是包含各種實際操作範例的 GitHub 儲存庫，可協助您瀏覽常見的架構模式、解決方案和服務。此項目會以最新的服務和範例經常更新。
2. 改善您的工作負載，為選取的設計模式建立模型，並使用服務和服務組態選項改善您的工作負載效能。
    a. 使用下列位置的資源訓練您的內部團隊： AWS Skills Guild。
    b. 使用 AWS Partner Network 快速提供專業知識，並擴展您的改進能力。

實作計劃的工作量： 若要建立此最佳實務，您必須了解有哪些設計模式和服務可協助您改善工作負載效能。評估設計模式後，實作設計模式將是 高 工作量。

資源

相關文件：

- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心
- Amazon Builders' Library
- 利用卸載避免超載
- 快取挑戰和策略

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- This is My Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF08-BP03 確定權衡如何影響客戶和效率：

在評估與效能相關的改進時，判斷哪些選擇將如何影響客戶和工作負載效率。例如，如果使用鍵值資料存放區提高系統效能，請務必評估其最終一致性本質對客戶的影響。

透過指標和監控來確定系統中效能不佳的部分。確定如何進行改進、這些改進帶來的權衡，以及它們如何影響系統和使用者體驗。例如，實作快取資料有助於大幅提升效能，但需要明確的策略來確定更新或使快取資料失效的方式和時間，以防止不正確的系統行為。

常用的反模式：

- 您假設應該實作所有效能增益，即使實作有如最終一致性的權衡。

- 您只會在效能問題達到臨界點時才會評估工作負載變更。

建立此最佳實務的優勢： 評估潛在的效能相關改善項目時，必須判斷技術變更的權衡是否與工作負載要求一致。在某些情況下，您可能需要實作其他控制來彌補權衡。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

識別取捨：使用指標和監控來識別系統中效能不佳的部分。判斷如何進行改善，以及權衡對於系統和使用者體驗的影響。例如，實作快取資料有助於大幅提升效能，但需要明確的策略來確定更新或使快取資料失效的方式和時間，以防止不正確的系統行為。

資源

相關文件：

- Amazon Builders' Library
- Amazon QuickSight KPI
- Amazon CloudWatch RUM
- X-Ray 文件

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- 制定監控計劃
- 透過 Amazon CloudWatch RUM 優化應用程式
- Amazon CloudWatch Synthetics 的示範

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端

PERF08-BP04 衡量效能改進的影響

在進行變更以提高效能時，請評估所收集的指標和資料。使用此資訊來判斷效能提升對工作負載、工作負載元件和客戶所造成的影響。此測量可協助您了解權衡所帶來的改善，並協助您判斷是否產生任何負面影響。

Well-Architected 系統利用一組效能策略。確定哪種策略將對給定的熱點或瓶頸產生最大的積極影響。例如，跨多個關聯式資料庫系統將資料分區可以提高整體輸送量，同時保留對交易的支援，並且在每個分區中，快取可以幫助減少負載。

常用的反模式：

- 您會手動部署和管理可做為受管服務的技術。
- 當可以使用多個元件來提高工作負載的效能時，您只需專注於如聯網等單一元件。
- 您依賴客戶意見回饋和感受做為唯一的基準測試。

建立此最佳實務的優勢： 若要實作效能策略，您必須選擇多項服務和功能。當合併採用這些服務和功能時，您將能符合工作負載的效能需求。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

Well-Architected 系統使用效能相關策略的組合。確定哪種策略將對給定的熱點或瓶頸產生最大的積極影響。例如，跨多個關聯式資料庫系統將資料分區可以提高整體輸送量，同時保留對交易的支援，並且在每個分區中，快取可以幫助減少負載。

資源

相關文件：

- Amazon Builders' Library
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分散式負載測試

相關影片：

- Amazon Builders' Library 簡介 (DOP328)

- 透過 Amazon CloudWatch RUM 優化應用程式
- Amazon CloudWatch Synthetics 的示範

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端
- AWS 上的分散式負載測試

PERF08-BP05 使用各種與效能相關的策略

在適用的情況下，使用多種策略來提升效能。這些策略包括：快取資料以防止過多的網路或資料庫呼叫、使用資料庫引擎的唯讀複本來提高讀取速率、在可能的情況下對資料進行分區或壓縮以減少資料量，以及對結果進行緩衝和串流以避免阻塞。

在變更工作負載時，收集並評估指標以確定這些變更的影響。衡量對系統以及最終使用者的影響，以了解您的權衡如何影響您的工作負載。使用系統的方法 (例如負載測試) 來探索權衡是否可以提高效能。

常用的反模式：

- 您假設在客戶未投訴的情況下，工作負載效能即已足敷使用。
- 您只會在已進行效能相關變更後才收集效能資料。

建立此最佳實務的優勢： 若要最佳化效能和資源利用率，您需要取得整合操作檢視、即時精細資料和歷史參考。您可以建立儀表板，並對資料進行指標計算，以在工作負載隨著時間變更時，獲得工作負載的操作和使用率洞見。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

使用資料驅動的方法來發展您的架構：在變更工作負載時，收集並評估指標以確定這些變更的影響。衡量對系統以及最終使用者的影響，以了解您的權衡如何影響您的工作負載。使用系統的方法 (例如負載測試) 來探索權衡是否可以提高效能。

資源

相關文件：

- Amazon Builders' Library
- 實作 Amazon ElastiCache 的最佳實務
- AWS 資料庫快取
- Amazon CloudWatch RUM
- AWS 上的分散式負載測試

相關影片：

- Amazon Builders' Library 簡介 (DOP328)
- AWS 專用資料庫 (DAT209-L)
- 透過 Amazon CloudWatch RUM 優化應用程式

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端
- AWS 上的分散式負載測試

成本最佳化

主題

- 實作雲端財務管理
- 支出和用量感知
- 具有經濟效益的資源
- 管理需求與供應資源
- 隨時間優化

實作雲端財務管理

問題

- COST 1. 如何實作雲端財務管理？

COST 1 如何實作雲端財務管理？

透過實作雲端財務管理，組織可以透過優化成本和用量以及在 AWS 上進行規模調整，實現商業價值和財務上的成功。

最佳實務

- COST01-BP01 建立成本優化職能部門
- COST01-BP02 在財務與技術之間建立合作夥伴關係
- COST01-BP03 建立雲端預算和預測
- COST01-BP04 在組織程序中實作成本感知
- COST01-BP05 報告並通知成本優化
- COST01-BP06 主動監控成本
- COST01-BP07 及時了解新的服務版本

COST01-BP01 建立成本優化職能部門

建立一支團隊 (雲端商業辦公室或雲端卓越中心)，負責建立並維護整個組織的成本感知。該團隊需要在組織中擔任財務、技術和業務角色的人員。

未建立此最佳實務時的曝險等級： 高

實作指引

建立雲端商業辦公室 (CBO) 或雲端卓越中心 (CCOE) 團隊，負責建立並維護雲端運算的成本感知文化。它可以是現有個人、組織內的團隊，或是由組織內主要財務、技術和組織利害關係人組成的新團隊。

此職能部門 (個人或團隊) 會優先進行成本管理和成本優化活動，並在這上面花費所需時間。相較於大型企業的全職職能部門，小型組織的此職能部門可能會花費較少百分比的時間。

此職能部門需要採行跨領域合作的方法，具備專案管理、資料科學、財務分析和軟體或基礎架構開發等能力。此職能部門可在三種不同的所有權下執行成本優化，以改善工作負載效率：

- 集中式： 透過財務營運、成本優化、CBO 或 CCOE 等指定團隊，客戶得以設計和實作管控機制，並推動整家公司的最佳實務。
- 分散式： 協助技術團隊執行優化。

- 混合: 同時結合集中式與分散式，團隊將可互相合作執行成本優化。

可以根據成本優化目標 (例如工作負載效率指標) 來衡量此職能部門的執行和交付能力。

您必須設法讓高層支持此職能部門做出改變，這是成功的關鍵因素。高層保證人被視為是具成本效率雲端消耗的擁護者，並為此職能部門提供向上呈報支援，以確保其成本優化活動獲得組織定義的優先級。若非如此，指引將被忽略，且節省成本將不會列為優先要務。高層保證人和此職能部門可共同確保您的組織會高效使用雲端，並持續提供商業價值。

如果您有商業計劃、Enterprise-On-Ramp 或 Enterprise Support 計劃，且需要建立此團隊或職能部門的相關協助，請透過客戶團隊洽詢雲端財務管理 (CFM) 專家。

實作步驟

- 定義關鍵成員： 您需要確保組織的所有相關部分都做出貢獻，並與成本管理息息相關。組織內的常見團隊通常包括：財務、應用程式或產品擁有者、管理和技術團隊 (DevOps)。有些團隊是全職參與 (財務、技術)，有些團隊則可視需要定期參與。執行 CFM 的個人或團隊通常須具備下列技能：
  - 軟體開發技能 - 如果正在建構指令碼和自動化。
  - 基礎架構工程技能 - 用以部署指令碼或自動化，或理解服務或資源的佈建方式。
  - 操作敏銳度 - CFM 關乎雲端的高效運作，具體做法包括評估、監控、修改、規劃及擴展雲端的有效使用。
- 定義目標和指標： 該職能部門需要以不同的方式提供價值給組織。定義的目標會隨著組織的發展而不斷演變。常見的活動包括：建立和執行整個組織的成本優化教育計畫，制定整個組織的標準 (例如成本優化的監控和報告)，以及設定工作負載優化目標。此職能部門也需要定期向組織報告組織成本優化的能力。

  您可以定義以價值衡量的關鍵績效指標 (KPI)。KPI 可以是以成本衡量或以價值衡量的。在定義 KPI 時，您可以從效率的角度來計算預期成本，並計算預期的商業成果。以價值衡量的 KPI 會將成本與用量指標繫結至商業價值驅動力，協助我們釐清變更 AWS 支出的合理性。要導出以價值衡量的 KPI，首重整個組織的相互合作，以期能共同擬定出一組標準 KPI。
- 建立定期規律： 各群組 (財務、技術和業務團隊) 應定期會談，並審查其目標和指標。一般的規律包括審查組織的狀態、審查目前執行的任何計畫，以及審查整體財務和優化指標。然後，會更詳細地報告關鍵工作負載。

  在這類定期會議中，您可以審查工作負載效率 (成本) 和商業成果。例如，工作負載成本上升 20% 與增加的客戶用量，是相對應的。在此案例中，這 20% 的成本上升可被視為投資。這些定期會議可協助團隊找出以價值衡量、為整個組織提供實質意義的 KPI。

資源

相關文件：

- AWS CCOE 部落格
- 建立雲端商業辦公室
- CCOE - 雲端卓越中心

相關影片：

- Vanguard CCOE 成功案例

相關範例：

- 使用雲端卓越中心 (CCOE) 推動整體企業轉型
- 建置 CCOE 以推動整體企業轉型
- 建置 CCOE 時應避開的 7 大陷阱

COST01-BP02 在財務與技術之間建立合作夥伴關係

讓財務和技術團隊參與討論雲端之旅各個階段的成本和用量。各團隊定期碰面並討論相關主題，例如，組織總目標和具體目標、成本和用量的目前狀態，以及財務和會計實務。

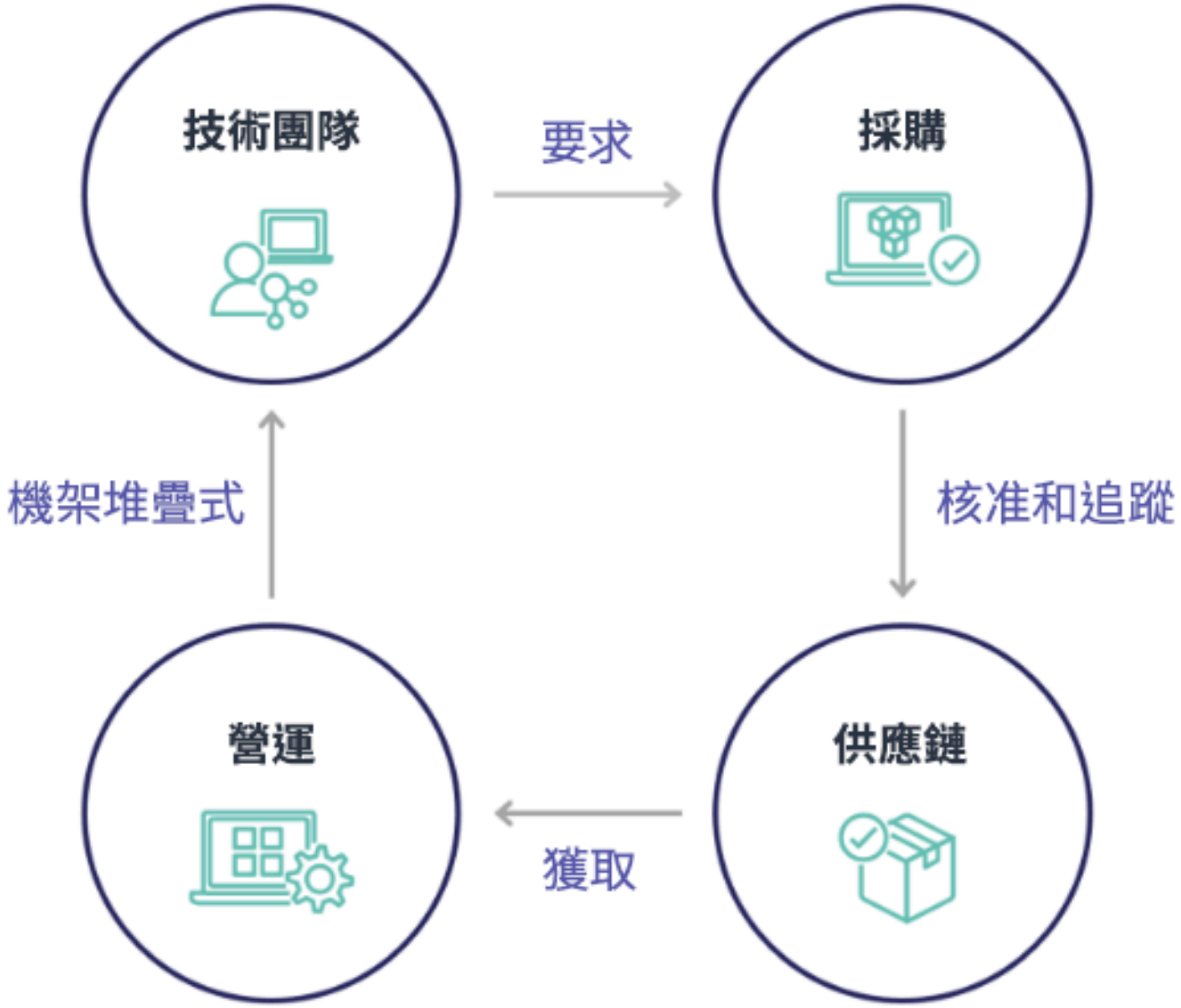
未建立此最佳實務時的曝險等級： 高

實作指引

由於核准、採購和基礎設施部署週期縮短，技術團隊可在雲端提高創新速度。對於之前習慣於執行耗時且資源密集型程序，以便採購資料中心和內部部署環境，並且只在核准專案時才分配成本的財務組織來說，這是一項調整。

就金融與採購組織的觀點而言，資本預算、資金要求、核准、採購和安裝實體基礎架構的流程，在過去數十年來早已廣為人知並標準化：

- 工程或 IT 團隊通常是要求者
- 核准者和採購者由不同的財務團隊擔任
- 營運團隊負責建構、堆疊及交付現成可用的基礎架構



採用雲端後，基礎架構的採購和取用不再受制於一連串的相依性。在雲端模型中，技術及產品小組不再只是建置者，而是產品的操作人員和擁有者，負責處理在過去與財務與營運團隊相關聯的多數活動，包括採購和部署。

要佈建雲端資源，所需的其實就是使用者帳戶，和正確的一組許可。IT 和財務風險也因而降低；這意味著，團隊只需按幾下滑鼠或執行 API 呼叫，即可終止閒置或非必要的雲端資源。這也讓技術團隊得以加速創新 – 基於建立和推翻試驗的靈活性與能力。儘管使用雲端的本質是多變的，就資本預算和預測的角度而言可能會影響到可預測性，但雲端仍讓組織得以降低過度佈建的成本，以及降低因保守的佈建不足而伴隨的機會成本。

在關鍵財務和技術利害關係人之間建立合作夥伴關係，以形成對組織目標的共識，並建立可在雲端運算可變支出模型中取得財務成功的機制。組織內的相關團隊必須參與雲端之旅各個階段的成本和用量討論，包括：

- 財務主管： 財務長、財務總監、財務規劃師、商業分析師、採購和應付帳款必須了解雲端消費模式、購買選項和每月發票開立程序。財務部門需要與技術團隊合作，來建立 IT 價值故事並加以傳播，以協助業務團隊了解技術支出與業務成果之間的連結。這樣，技術支出就不再被視為成本，而是投資。由於雲端與內部部署營運存在基本差異 (例如，用量改變速率、按使用付費定價、分級定價、定價模式以及詳細帳單和用量資訊)，財務組織必須了解雲端用量如何影響商業層面，包括採購程序、激勵追蹤、成本分配和財務報表。
- 技術主管： 技術主管 (包括產品和應用程式擁有者) 必須了解財務需求 (例如，預算限制) 以及業務需求 (例如，服務水準協議)。如此可允許實作工作負載，達成組織希望的目標。

財務與科技的合作夥伴關係可帶來下列好處：

- 財務和技術團隊可近乎即時地檢視成本和用量。
- 財務和技術團隊建立標準操作程序來處理雲端支出變化。
- 在資本如何用於購買承諾折扣 (例如，預留執行個體或 AWS Savings Plans)，以及如何使用雲端來發展組織方面，財務利害關係人會擔任策略顧問。
- 現有的應付帳款和採購程序會與雲端搭配使用。
- 財務和技術團隊共同預測未來的 AWS 成本和用量，以評估並擬定組織預算。
- 透過共同的語言以及對財務概念的一致理解，促進跨組織溝通。

組織內應參與成本和用量討論的其他利害關係人包括：

- 業務單位擁有者： 業務單位擁有者必須了解雲端業務模式，以便對業務單位和全公司提供指引。當有需要預測成長和工作負載用量，以及需要評估長期購買選項，例如預留執行個體或 Savings Plans 時，此項雲端知識相當重要。
- 工程團隊： 在財務與技術團隊之間建立合作夥伴關係至關重要，這是培養成本感知文化，鼓勵工程師對雲端財務管理 (CFM) 採取行動，所不可或缺的。CFM 或財務營運從業人員與財務團隊的常見問題之一，是不易讓工程師了解雲端業務的全貌、遵循最佳實務，以及執行建議的動作。
- 第三方： 如果您的組織使用第三方 (例如，顧問或工具)，請確保他們符合您的財務目標，並能透過其參與模式和投資報酬率 (ROI) 證實符合。通常第三方會報告和分析其管理的一切工作負載，並且提供所設計一切工作負載的成本分析。

要實作 CFM 並取得成功，需要財務、技術和業務團隊之間進行協作，並且需要轉變整個組織傳達和評估雲端支出的方式。請納入工程團隊，使他們在各階段都能加入這些成本與用量的討論中，並鼓勵他們遵循最佳實務，並據以執行已達成共識的動作。

實作步驟

- 定義關鍵成員： 確認您的財務和技術團隊中的所有相關成員都參與此合作夥伴關係。相關財務成員會處理雲端帳單。涉及人員通常包括財務總監、財務控制者、財務規劃師、商業分析師、採購和採購專員。技術成員通常是產品與應用程式擁有者、技術經理以及在雲端進行建置的所有團隊的代表。其他成員可能包括業務單位擁有者，例如，顧問等會影響產品用量的行銷單位，以及實現與目標和機制保持一致並協助報告的第三方人員。
- 定義討論主題： 確定團隊中常見的主題，或需要有共識的主題。從建立時開始追蹤成本，直到帳單支付為止。請記下所有參與的成員，以及需要應用的組織程序。了解採用的每個步驟或程序及相關資訊，例如可用的定價模式、分級定價、折扣模式、預算編列和財務要求。

- 建立定期規律： 若要建立財務與技術的合作夥伴關係，請建立定期通訊規律，以樹立並維持一致性。該群組需要針對他們的目標和指標定期聚會進行討論。一般的規律包括審查組織的狀態、審查目前執行的任何計畫，以及審查整體財務和優化指標。然後，會更詳細地報告關鍵工作負載。

資源

相關文件：

- AWS 新聞部落格

COST01-BP03 建立雲端預算和預測

調整現有的組織預算編列和預測程序，使其與本質會高度變動的雲端成本和用量相容。程序必須是動態的，並使用以趨勢為基礎和/或以業務驅動因素為基礎的演算法。

未建立此最佳實務時的曝險等級： 高

實作指引

客戶使用雲端以提高效率、速度和靈活性，這會產生高度變動的成本和用量。成本會隨著工作負載效率的增加或部署新的工作負載和功能而降低。成本是有可能隨著工作負載效率的提升，或部署新的工作負載和功能而增加的。或者，工作負載會擴展以服務更多客戶，進而提高雲端用量和成本。資源現在的立即存取性比以往更高。借助於雲端的彈性，成本和預測也更具彈性。必須修改現有的組織預算編列程序，以納入這樣的變動。

使用以趨勢為基礎的演算法 (使用歷史成本做為輸入) 或使用以業務驅動因素為基礎的演算法 (例如，新產品推出或區域擴展)，或結合兩者，調整現有的預算和預測程序，使其變得更加機動。

使用 AWS Budgets 藉由指定時段、重複週期或金額 (固定或可變)，並新增篩選條件 (例如服務、AWS 區域和標籤)，來設定精細的自訂預算。若要及時了解現有預算的執行情況，您可以建立和排程 AWS Budgets 報告 ，以定期透過電子郵件將報告傳送給您和利害關係人。您也可以建立 AWS Budgets 提醒 (根據本質上是被動的實際成本，或根據預測成本)，以便有足夠的時間可對潛在的成本超支實作緩解措施。當您的成本或用量超出或預計超出預算額度時，系統會提醒您。

AWS 提供建立動態預測和預算編列流程的靈活性，因此您可以隨時了解成本是否符合或超過預算限制。

使用 AWS Cost Explorer 根據您過去的支出預測已定義的未來時間範圍內的成本。AWS Cost Explorer 預測引擎會根據收費類型 (例如，預留執行個體) 對您的歷史資料進行細分，並結合使用機器學習和基

於規則的模型，來分別預測所有收費類型的支出。使用 AWS Cost Explorer 根據對歷史成本套用的機器學習演算法 (以趨勢為基礎)，預測每日 (最多三個月) 或每月 (最多 12 個月) 的雲端成本。

使用 Cost Explorer 判斷以趨勢為基礎的預測後，請使用 AWS Pricing Calculator 根據預期的用量 (流量、每秒請求數、必要的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體等等) 來估計您的 AWS 使用案例與未來的成本。您可以據此規劃支出、尋找節省成本的機會，以及在使用 AWS 時做出明智的決定。

使用 AWS Cost Anomaly Detection 防止或避免成本超出預期，並強化控制且不影響創新速度。AWS Cost Anomaly Detection 採用進階機器學習技術來識別異常支出與根本原因，以便您迅速做出因應。只需簡單的三個步驟，您即可建立自己的情境化監視器，並且在偵測到任何異常支出時收到提醒。讓建置器代勞建置工作，讓 AWS Cost Anomaly Detection 監控您的支出，並降低帳單超出預期的風險。

如 Well-Architected 成本優化支柱的「財務與技術的合作夥伴關係」一節所述，IT 部門、財務部門與其他利害關係人之間必須建立合作夥伴關係和規律，以確保所有人都會使用相同的工具或程序，而保有一致性。如果預算可能需要改變，提高接觸頻率可能有助於提升對這些變化的因應速度。

實作步驟

- 更新現有的預算和預測程序： 在預算編列和預測程序中，實作以趨勢為基礎和/或以業務驅動因素為基礎的演算法。
- 設定提醒和通知： 使用 AWS Budgets 提醒和 Cost Anomaly Detection。
- 會同主要利害關係人執行定期審查： 例如，會同 IT、財務、平台和其他業務部門的利害關係人，商討如何因應經營方向與用量的變化。

資源

相關文件：

- AWS Cost Explorer
- AWS Budgets
- AWS Pricing Calculator
- AWS Cost Anomaly Detection
- AWS License Manager

相關範例：

- 推出：AWS Cost Explorer 現已提供以用量為基礎的預測

- AWS Well-Architected 實驗室 - 成本與用量管控

COST01-BP04 在組織程序中實作成本感知

在會影響用量的全新或現有程序中實作成本感知、建立成本的透明度與權責劃分，並利用現有程序落實成本感知。在員工培訓中實作成本感知。

未建立此最佳實務時的曝險等級： 高

實作指引

必須在新的和現有的組織程序中實作成本感知。對於其他最佳實務而言，這是基本的必備能力之一。建議盡可能重複使用和修改現有程序，這樣可將對靈活性和速度的影響降到最低。向技術團隊以及業務與財務團隊的決策者報告雲端成本，不僅可增強成本感知，也可為財務與業務利害關係人建立效率的關鍵績效指標 (KPI)。下列建議有助於在您的工作負載中實作成本感知：

- 確認變更管理包含成本測量，以量化變更所帶來的財務影響。這有助於主動解決成本相關疑慮，並提供成本節省資訊。
- 確認成本優化是您營運能力的核心部分。例如，您可以利用現有的事故管理程序，調查並找出成本和用量異常或成本超支的根本原因。
- 透過自動化或工具加速節省成本和實現商業價值。考慮實作的成本時，將投資報酬率 (ROI) 部分納入對話中，以證明投入時間或金錢的合理性。
- 藉由實作雲端支出的回報 (showback) 或計費 (chargeback) 來分配雲端成本 (包括以承諾為基礎的購買選項、共用服務和市場購買的支出)，以實現最具成本感知力的雲端使用。
- 擴展現有的培訓和發展計畫，納入整個組織的成本感知培訓。建議包含持續培訓和認證。這將建立一個能夠自我管理成本和用量的組織。
- 充分利用免費的 AWS 原生工具，例如 AWS Cost Anomaly Detection、AWS Budgets和 AWS Budgets 報告。

如果組織一貫採行 雲端財務管理 (CFM) 實務準則，這些行為將深植於工作與決策制定的機制中。結果會產生更注重成本的文化，無論是開發人員設計新的雲端原生應用程式，還是財務經理分析這些新的雲端投資的投資報酬率，皆注重成本。

實作步驟

- 識別相關的組織程序： 每個組織單位審查其程序，並識別影響成本和用量的程序。任何導致資源建立或終止的程序都需要納入審查。尋找能夠在業務上支援成本感知的程序，例如事故管理和培訓。

- 建立自主的成本感知文化： 確定所有相關的利害關係人都認同成本的改變原因和影響，因此都了解雲端成本。這將可讓您的組織針對創新建立自主的成本感知文化。
- 以成本感知更新程序： 每項程序都會修改為可感知成本。程序可能需要額外的預先檢查，例如評估成本的影響，或進行後置檢查以驗證成本和用量預期的變更是否發生。可以擴展培訓和事件管理等支援程序，以包含成本和用量的項目。

如需協助，請透過客戶團隊洽詢 CFM 專家，或瀏覽下方的資源和相關文件。

資源

相關文件：

- AWS 雲端財務管理

相關範例：

- 高效雲端成本管理的策略
- 成本控制部落格系列 3：如何處理成本衝擊
- AWS Cost Management 入門指南

COST01-BP05 報告並通知成本優化

設定 AWS Budgets 和 AWS Cost Anomaly Detection 以針對目標提供有關成本和用量的通知。定期召開會議以分析您工作負載的成本效率並推廣成本感知文化。

未建立此最佳實務時的曝險等級： 低

實作指引

您必須定期在組織內報告成本和用量優化。您可以提交專門的成本優化報告，或在工作負載的定期營運報告週期中包含成本優化部分。使用服務和工具來識別節省成本的機會，並付諸實行。AWS Cost Explorer 提供儀表板和報告。您可以 AWS Budgets 報告。

使用 AWS Budgets 設定自訂預算以追蹤成本與用量，並且在超出閾值時快速回應電子郵件或 Amazon Simple Notification Service (Amazon SNS) 通知所傳來的提醒。將您偏好的預算 期間設定為每日、每月、每季或每年，並建立特定預算限制，以持續掌握實際或預測的成本與用量逐漸接近預算閾值的情形。您也可以設定 提醒 和 動作 ，使其自動執行以因應這些提醒，或是在超出預算目標時透過核准程序執行。

實作成本和用量通知，以確保能夠快速處理非預期的成本和用量變化。AWS Cost Anomaly Detection 可避免成本超出預期，並強化控制且不影響創新速度。AWS Cost Anomaly Detection 可識別異常支出與根本原因，有助於降低帳單超出預期的風險。只需簡單的三個步驟，您即可建立自己的情境化監視器，並且在偵測到任何異常支出時收到提醒。

您也可以使用 Amazon QuickSight 搭配 AWS Cost and Usage Report (CUR) 資料，用更精細的資料提供高度自訂的報告。Amazon QuickSight 可讓您排程報告及接收定期成本報告電子郵件，以了解歷史成本與用量或節省成本的機會。

使用 AWS Trusted Advisor作為指引，以確認已佈建的資源是否符合 AWS 的成本優化最佳實務。

定期建立相關報告，納入 Savings Plans、預留執行個體和 Amazon Elastic Compute Cloud (Amazon EC2) 適當調整大小的建議 (來自 AWS Cost Explorer)，以開始降低與穩定狀態的工作負載、閒置和低利用率資源相關聯的成本。識別並收回與已部署資源的雲端浪費相關聯的支出。若建立了大小不當的資源，或是發現非預期的不同用量模式時，就會發生雲端浪費。請遵循 AWS 最佳實務以減少浪費，並優化和節省 您的雲端成本。

定期產生報告以找出更好的資源採購選項，進而降低工作負載的單位成本。Savings Plans、預留執行個體或 Amazon EC2 Spot 執行個體等購買選項可為容錯工作負載提供最大的成本節省效益，並且可讓利害關係人 (企業擁有者、財務和技術團隊) 參與這些承諾討論。

將報告分享給相關各方，使其了解可能有助於降低雲端總體擁有成本 (TCO) 的機會或新版本公告。採用新的服務、區域、功能、解決方案或新方法來實現進一步的成本降低。

實作步驟

- 設定 AWS Budgets： 針對您的工作負載在所有帳戶上設定 AWS Budgets。使用標籤來設定整體帳戶支出的預算，以及工作負載的預算。
  - Well-Architected 實驗室：成本與管控用量
- 成本優化報告： 設置定期週期來討論和分析工作負載的效率。使用建立的指標來報告所達到的指標和達到指標所需的成本。識別並修正任何負面趨勢，並識別您可以在整個組織中推廣的正面趨勢。報告應讓應用程式團隊和擁有者、財務和管理層的代表參與。
  - Well-Architected 實驗室：視覺化

資源

相關文件：

- AWS Cost Explorer

- AWS Trusted Advisor
- AWS Budgets
- AWS Budgets 最佳實務
- Amazon CloudWatch
- AWS CloudTrail
- Amazon S3 分析
- AWS Cost and Usage Report

相關範例：

- Well-Architected 實驗室：成本與管控用量
- Well-Architected 實驗室：視覺化
- 開始優化 AWS 雲端成本的關鍵方法

COST01-BP06 主動監控成本

實作工具和儀表板來主動監控工作負載的成本。定期使用已設定的工具或現成可用的工具來審查成本。不要只在收到通知時才查看成本和類別。主動監控和分析成本有助於識別正面趨勢，並讓您在整個組織中推廣。

未建立此最佳實務時的曝險等級： 低

實作指引

建議監控組織內的成本與用量，而不只是在發生例外狀況或異常狀況時。在所有辦公室或工作環境中均可以使用高度可見的儀表板，確保了關鍵人員可存取所需的資訊，並且這些儀表板指出組織專注於成本優化的程度。可見的儀表板可讓您主動推廣成功的成果，並在整個組織中加以實作。

建立日常工作或常規以使用 AWS Cost Explorer 或任何其他儀表板 (例如 Amazon QuickSight)，以查看成本並主動分析。在 AWS 帳戶層級、工作負載層級或特定 AWS 服務層級使用群組和篩選來分析 AWS 服務用量與成本，並驗證是否符合預期。使用每小時和資源層級精細度與標籤，來篩選及識別最高排名資源所產生的成本。您也可以使用 成本智慧儀表板自行建置報告，這是一個 Amazon QuickSight 解決方案，由 AWS 解決方案架構師所建置，會比較您的預算和實際的成本與用量。

實作步驟

- 成本優化報告： 設置定期週期來討論和分析工作負載的效率。使用建立的指標來報告所達到的指標和達到指標所需的成本。識別並修正任何負面趨勢，並識別要在整個組織中推廣的正面趨勢。報告應讓應用程式團隊和擁有者、財務和管理層的代表參與。
- 針對成本與用量建立並啟用每日精細度 AWS Budgets ，以及時採取相關措施來防止任何潛在的成本超支： AWS Budgets 可讓您設定提醒通知，以隨時獲知是否有任何預算類型超出預先設定的閾值。AWS Budgets 的最佳運用方式是將您預期的成本與用量設為限制，如此即可將任何超過預算的部分視為超支。
- 建立 AWS Cost Anomaly Detection 作為成本監視器： AWS Cost Anomaly Detection 會使用進階機器學習技術來識別異常支出與根本原因，以便您迅速做出因應。它可讓您設定成本監視器以定義您要評估的支出區段 (例如個別 AWS 服務、成員帳戶、成本分配標籤和成本類別)，並且可讓您設定接收提醒通知的時間、位置和方式。每個監視器可以為企業擁有者和技術團隊連結多個提醒訂閱，包括每個訂閱的名稱、成本影響閾值和提醒頻率 (個別提醒、每日摘要、每週摘要)。
- 使用 AWS Cost Explorer，或整合您的 AWS Cost and Usage Report (CUR) 資料與 Amazon QuickSight 儀表板，將組織的成本視覺化： AWS Cost Explorer 有簡單易用的介面可讓您視覺化、了解和管理您在一段時間內的 AWS 成本和用量。AWS Well-Architected 成本智慧儀表板 是一個可自訂且可供存取的儀表板，可協助您建立自身成本管理和優化工具的基礎。

資源

相關文件：

- AWS Budgets
- AWS Cost Explorer
- 每日成本與用量預算
- AWS Cost Anomaly Detection

相關範例：

- Well-Architected 實驗室：視覺化
- Well-Architected 實驗室：進階視覺化
- Well-Architected 實驗室：雲端智慧儀表板
- Well-Architected 實驗室：成本視覺化
- Slack 的 AWS Cost Anomaly Detection 提醒

COST01-BP07 及時了解新的服務版本

定期諮詢專家或 AWS 合作夥伴，以了解哪些服務和功能可以降低成本。檢閱 AWS 部落格和其他資訊來源。

未建立此最佳實務時的曝險等級： 低

實作指引

AWS 持續加入新功能，讓您能夠利用最新技術加快試驗及創新速度。您可以實作新的 AWS 服務和功能，以提升工作負載的成本效益。定期檢閱 AWS 成本管理、 AWS 新聞部落格、 AWS 成本管理部落格和 AWS 最新消息 以取得新的服務和功能版本的相關資訊。最新消息貼文會在所有 AWS 服務、功能和區域擴充公告發行時提供其簡短概要。

實作步驟

- 訂閱部落格： 前往 AWS 部落格頁面，訂閱最新消息部落格和其他相關部落格。您可以用電子郵件地址 在通訊偏好設定 頁面進行註冊。
- 訂閱 AWS 新聞： 定期檢閱 AWS 新聞部落格 和 AWS 最新消息 以取得新的服務和功能版本的相關資訊。訂閱 RSS 摘要，或透過您的電子郵件關注公告和版本。
- 關注 AWS 降價： 我們所有服務的定期價降，已成為 AWS 將來自於規模的經濟效益傳遞給客戶的標準機制。截至 2022 年 4 月為止，AWS 自 2006 年推出後已降價 115 次。如果您有任何商業決策因價格考量而未定，您可以在降價和新的服務整合之後再次加以審查。您可以了解先前執行過的降價，包括 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，請前往 AWS 新聞部落格的降價類別。
- AWS 活動和會議： 參加您當地的 AWS 高峰會，以及與您當地區域的其他組織一同參加當地會議。如果您無法親自與會，請試著參加線上活動，聽聽 AWS 專家和其他客戶的商業案例。
- 與您的客戶團隊會面： 與您的客戶團隊排定一個定期規律，與他們會面並討論產業趨勢和 AWS 服務。與您的客戶經理、解決方案架構師和支援團隊進行討論。

資源

相關文件：

- AWS 成本管理
- AWS 最新消息
- AWS 新聞部落格

相關範例：

- Amazon EC2 – 15 年的 IT 成本優化和節省
- AWS 新聞部落格 - 降價

支出和用量感知

問題

- COST 2 您如何管控用量？
- COST 3 您如何監控用量和成本？
- COST 4 如何進行資源除役？

COST 2 您如何管控用量？

建立原則和機制以確保產生的成本合理，同時達成目標。您可以運用相互制衡的方法，在不超支的情況下創新。

最佳實務

- COST02-BP01 根據您的組織要求制定政策
- COST02-BP02 實作總目標和具體目標
- COST02-BP03 實作帳戶結構
- COST02-BP04 實作群組和角色
- COST02-BP05 實作成本控制措施
- COST02-BP06 追蹤專案生命週期

COST02-BP01 根據您的組織要求制定政策

制定定義組織如何管理資源的政策。政策應涵蓋資源和工作負載的成本方面，包括在資源生命週期中資源的建立、修改和除役。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解組織的成本和動因對於有效管理成本和用量，以及識別降低成本機會至關重要。組織通常會營運由多個團隊執行的多個工作負載。這些團隊可能分屬不同組織單位，各有本身的收入流。能將資源成本歸

因至工作負載、個別組織或產品擁有者，能帶動高效使用的行為模式，有助於減少浪費。準確的成本和用量監控可讓您了解組織單位和產品的獲利能力，並允許您對於將資源分配到組織內的何處做出更知情的決策。讓組織內所有層級建立用量意識，是推動變革的關鍵，因為用量的變革會帶來成本變革。請考慮採行多面向的方法以了解您的用量和開支。

執行管控的第一步是使用組織的要求來制定雲端使用政策。這些政策定義您的組織如何使用雲端以及如何管理資源。政策應涵蓋資源和工作負載的成本或用量的各方面，包括在資源生命週期中資源的建立、修改和除役。

政策應該簡單易懂，以便有效地在整個組織中實作。從廣泛的高階政策開始，例如允許使用哪個地理區域，或一天中應該執行資源的時間。逐步為各組織單位和工作負載優化政策。常用政策包括可以使用哪些服務和功能 (例如，測試或開發環境中的效能較低儲存體)，以及不同群組可以使用哪些類型的資源 (例如，開發帳戶中最大的資源大小是中型)。

實作步驟

- 與團隊成員會面： 若要制定政策，請讓組織中所有團隊成員指定其要求，並據此加以記錄。採取反復的方法，從廣泛討論開始，然後在每個步驟持續細化至最小的單位。團隊成員包括對工作負載有直接關係的人員，例如組織單位或應用程式擁有者，以及支援群組，例如安全和財務團隊。
- 定義工作負載的位置： 定義工作負載營運的位置，包括國家和國家中的區域。此資訊用來映射至 AWS 區域和可用區域。
- 定義並分組服務和資源： 定義工作負載所需的服務。針對每項服務，指定所需的類型、大小和資源數量。依職能定義資源群組，例如應用程式伺服器或資料庫儲存體。資源可屬於多個群組。
- 依職能定義並分組使用者： 定義與工作負載互動的使用者，專注於使用者執行的操作以及他們如何使用工作負載，而不是專注於他們的身分或他們在組織中的位置。將類似的使用者或職能分組在一起。您可以使用 AWS 受管政策做為指南。
- 定義動作： 使用先前識別的位置、資源和使用者，定義每個項目在其生命週期內 (開發、營運和除役) 達成工作負載結果所需的動作。根據每個位置中的群組 (不是群組中的個別元素) 來識別動作。從廣泛地讀取或寫入開始，然後縮小精細至每項服務的特定動作。
- 定義審查期間： 工作負載和組織要求會隨著時間變更。定義工作負載審查排程，以確保其與組織優先事項保持一致。
- 記錄政策： 確保組織可視需要存取已定義的政策。這些政策用於實作、維護和稽核環境的存取權。

資源

相關文件：

- 適用於各工作職能的 AWS 受管政策
- AWS 多帳戶帳單策略
- AWS 服務的動作、資源和條件金鑰
- 雲端產品
- 使用 IAM 政策控制對 AWS 區域 的存取
- 全球基礎設施區域和可用區域

COST02-BP02 實作總目標和具體目標

為您的工作負載實作成本和用量目標。總目標可為您的組織提供成本和用量的方向，而具體目標可為您的工作負載提供可測量的結果。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

為您的組織制定成本與用量總目標和具體目標。總目標可為您的組織提供預期結果的指導和方向。具體目標是要實現的具體可衡量成果。總目標範例：平台用量應該大幅增加，而成本只稍微增加 (非線性)。具體目標範例：平台用量增加 20%，成本增加少於 5%。另一個常見的總目標是工作負載每 6 個月必須更有效率。相關的具體目標可能是，工作負載的每個輸出成本需要每 6 個月減少 5%。

雲端工作負載的常見總目標是提高工作負載效率，也就是隨著時間降低工作負載每個業務成果的成本。建議為所有工作負載實作此總目標，並設定一個具體目標，例如每 6-12 個月效率增加 5%。這可透過建立成本優化的能力以及發佈新的服務和功能，在雲端中達成。

實作步驟

- 定義預期的用量等級： 從著重於用量等級開始。與應用程式擁有者、行銷團隊和更大的業務團隊互動，以了解工作負載的預期用量等級。客戶需求如何隨著時間而變更，以及如何因季節性增加或行銷活動而變更。
- 定義工作負載資源與成本： 定義用量等級後，量化達成這些用量等級所需的工作負載資源變更。您可能需要為工作負載元件增加資源的大小或數量、增加資料傳輸，或將工作負載元件變更為特定等級的不同服務。指定這些要點的成本為何，以及當用量發生變化時，成本會有什麼變化。
- 定義業務目標： 從預期用量和成本變更中取得輸出，將此項目與預期的技術變更或任何您正在執行的計畫結合，並制定工作負載的目標。目標必須涵蓋用量、成本和兩者之間的關係。如果預期有成本變更但用量不變，請確認制定有組織計畫，例如培訓和教育等能力打造計畫。
- 定義目標： 對定義的每個總目標，指定可測量的具體目標。如果總目標是要提高工作負載的效率，具體目標會是量化改善量，這一般是花費每一美元所獲得的業務輸出，以及交付時間。

資源

相關文件：

- 適用於各工作職能的 AWS 受管政策
- AWS 多帳戶帳單策略
- 使用 IAM 政策控制對 AWS 區域 的存取

COST02-BP03 實作帳戶結構

實作與您的組織對應的帳戶結構。這有助於在整個組織中分配和管理成本。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

AWS 為一父多子的帳戶結構，通常稱為管理帳戶 (父帳戶，先前稱為付款人) 帳戶成員 (子帳戶，先前稱為連結的帳戶)。無論您的組織規模或用量為何，最佳實務一律至少有一個管理帳戶和一個成員帳戶。所有工作負載資源應僅位於成員帳戶內。

對於您應擁有幾個 AWS 帳戶，並沒有適合所有人的答案。評估目前和未來的運作與成本模式，確保 AWS 帳戶的結構呼應組織的總目標。有些公司基於業務原因建立多個 AWS 帳戶，例如：

- 組織單位、成本中心或特定工作負載之間需要行政管理及/或會計年度和帳單上的區隔。
- AWS 服務限制是依照特定工作負載區分所設定。
- 工作負載和資源之間需要區隔和隔離。

在 AWS Organizations中，合併帳單 建立一或多個成員帳戶與管理帳戶之間的結構。成員帳戶可讓您依群組隔離和區分成本和用量。常見實務是各組織單位分別有成員帳戶 (例如財務、行銷和銷售)，或是各個環境生命週期分立 (例如開發、測試和生產)，或是各工作負載分立 (工作負載 a、b 和 c)，再使用合併帳單彙總這些連結帳戶。

合併帳單可讓您將多個 AWS 帳戶的款項合併至單一管理帳戶之下，同時仍為各連結帳戶的活動提供可見度。由於成本和用量的在管理帳戶中彙總，這可讓您獲得最大的服務容量折扣以及最大的使用承諾折扣 (Savings Plans 和預留執行個體)，以享受最高折扣。

AWS Control Tower 可以快速建立和設定多個 AWS 帳戶，確保管控符合組織的要求。

實作步驟

- 定義分隔要求： 分隔要求是多個因素的組合，包括安全性、可靠性和財務結構。依序處理每個因素，並指定工作負載或工作負載環境是否應與其他工作負載分開。安全性可確保遵守存取和資料要求。可靠性可確保限制受到管理，讓環境和工作負載不會影響其他項目。財務結構可確保有嚴格的財務分隔和責任。常見的分隔範例是生產和測試工作負載會在不同的帳戶開始執行，或使用單獨的帳戶，以便將發票和帳單資料提供給第三方組織。
- 定義分組要求： 分組要求不會覆寫分隔要求，而是用來協助管理。將不需要分隔的類似環境或工作負載分成同一組。例如，將來自一或多個工作負載的多個測試或開發環境分組在一起。
- 定義帳戶結構： 使用這些分隔和群組，為每個群組指定一個帳戶，並確保分隔要求得到維護。這些帳戶是您的成員帳戶或連結帳戶。透過將這些成員帳戶分組到單一管理帳戶或付款人帳戶下，您可以結合用量，以讓所有帳戶獲得更多數量折扣，並為所有帳戶提供單一帳單。您可以分隔帳單資料，以便在每個成員帳戶中檢視單獨的帳單資料。如果不允許透過任何其他帳戶查看某個成員帳戶中的用量或帳單資料，或是需要與 AWS 分開的帳單，請定義多個管理帳戶或付款人帳戶。在這種情況下，每個成員帳戶都有自己的管理帳戶或付款人帳戶。資源應一律放在成員或連結帳戶中。管理帳戶或付款人帳戶只能用於管理。

資源

相關文件：

- 適用於各工作職能的 AWS 受管政策
- AWS 多帳戶帳單策略
- 使用 IAM 政策控制對 AWS 區域 的存取
- AWS Control Tower
- AWS Organizations
- 合併帳單

相關範例：

- 分割 CUR 和共用存取

COST02-BP04 實作群組和角色

實作符合您政策的群組和角色，並控制哪些人員可以建立、修改或除役每個群組中的執行個體和資源。例如，實作開發、測試和生產群組。這適用於 AWS 服務和第三方解決方案。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在制定政策之後，您可以建立組織內的使用者邏輯群組和角色。這可讓您指派許可和控制用量。從簡要的人員分組開始。通常這與組織單位和工作角色 (例如 IT 部門的系統管理員或財務控制者) 相符。這些群組會匯聚執行類似任務且需要類似存取權限的人員。角色定義群組必須執行的工作。例如，IT 的系統管理員需要建立所有資源的存取權限，但分析團隊成員只需要建立分析資源的權限。

實作步驟

- 實作群組： 使用組織政策中定義的使用者群組，視需要實作對應的群組。如需使用者、群組和身份驗證的最佳實務，請參閱安全性支柱。
- 實作角色和政策： 使用組織政策中定義的動作，建立所需角色和存取政策。如需角色和政策的最佳實務，請參閱安全性支柱。

資源

相關文件：

- 適用於各工作職能的 AWS 受管政策
- AWS 多帳戶帳單策略
- 使用 IAM 政策控制對 AWS 區域 的存取
- Well-Architected 安全支柱白皮書

相關範例：

- Well-Architected 實驗室基本身分和存取

COST02-BP05 實作成本控制措施

根據組織政策以及定義的群組和角色實作控制措施。這些控制措施可證明成本的發生始終符合組織要求：例如，使用 AWS Identity and Access Management (IAM) 政策控制對區域或資源類型的存取。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

實作成本控制常見的第一步設定在發生偏離組織政策的成本或用量事件時發出通知。這可讓您快速採取動作，並驗證是否需要採取糾正措施，而不會限制或對工作負載或新的活動造成負面影響。在您了解工作負載和環境限制之後，即可施行管控。在 AWS 中，通知會透過 AWS Budgets執行，您可以在其中

定義每月的 AWS 成本預算、用量和承諾折扣 (Savings Plans 和預留執行個體)。您可以在彙總成本層級 (例如，所有成本) 或更精細的層級建立預算，其中只包含特定維度，例如連結的帳戶、服務、標籤或可用區域。

第二步，您可以透過 AWS Identity and Access Management (IAM) 和 AWS Organizations Service Control Policies (SCP)，在 AWS 中執行管控政策。IAM 可讓您安全地管理對 AWS 服務和資源的存取。使用 IAM，您可以控制誰可以建立和管理 AWS 資源、可建立的資源類型以及建立資源的位置。這可將非必要資源的建立降到最低。使用先前建立的角色和群組，並指派 IAM 政策 以執行正確的用量。SCP 可讓您集中控制組織中所有帳戶的最大可用許可，確保您的帳戶符合您的存取控制指導方針。SCP 只能在啟用所有功能的組織中使用，而且您可以設定 SCP， 為成員帳戶設定預設拒絕或允許的動作。如需實作存取權限管理的詳細資訊，請參閱 Well-Architected 安全支柱白皮書 。

亦可透過管理 Service Quotas 來實作管控。藉由確保服務配額設定為冗餘最低並且正確維護，可盡量避免建立超出組織要求的資源。為達成此目的，您必須了解要求的變更速度能有多快、了解進行中的專案 (包括資源的建立與除役兩者) 並將變更配額的實作速度能有多快列入作為考量因素。Service Quotas 可在需要時用來增加您的配額。

實作步驟

- 實作支出通知： 使用您定義的組織政策，建立 AWS 預算以在支出超出您的政策要求時發出通知。設定多個成本預算 (每個帳戶一個)，各帳戶會通知您整體帳戶支出。然後，針對帳戶中的較小單位，為每個帳戶設定額外的成本預算。這些單位會根據您的帳戶結構而有所不同。一些常見的範例是 AWS 區域、工作負載 (使用標籤) 或 AWS 服務。請確保您將電子郵件分發清單設定為通知收件人，而非個人的電子郵件帳戶。您可以設定超過數量時的實際預算，或使用預測預算來通知預測用量。
- 實作用量控制措施： 使用您定義的組織政策，實作 IAM 政策和角色來指定使用者可以執行的動作，以及他們無法執行的動作。一項 AWS 政策中可包含多項組織政策。使用與您定義政策相同的方式，一開始廣泛定義，然後在每個步驟中套用更精細的控制措施。服務限制也能有效控制用量。在您所有帳戶中實作正確的服務限制。

資源

相關文件：

- 適用於各工作職能的 AWS 受管政策
- AWS 多帳戶帳單策略
- 使用 IAM 政策控制對 AWS 區域 的存取

相關範例：

- Well-Architected 實驗室：成本與用量管控
- Well-Architected 實驗室：成本與用量管控

COST02-BP06 追蹤專案生命週期

追蹤、測量和稽核專案、團隊和環境的生命週期，以避免使用不必要的資源並節省成本。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

確保您追蹤工作負載的整個生命週期。這可確保不再需要工作負載或工作負載元件時，可以停用或修改它們。當您發佈新的服務或功能時，這特別有用。現有的工作負載和元件可能顯示為使用中，但應停用以便將客戶重新導向至新的服務。請注意先前的工作負載階段 – 工作負載進入生產環境後，之前的環境可能會停用或大幅降低容量，直到再次需要這些環境為止。

AWS 提供多種管理和管控服務，您可以用於實體生命週期追蹤。您可以使用 AWS Config 或者 AWS Systems Manager 提供 AWS 資源和組態的詳細目錄。建議與您現行的專案或資產管理系統整合，與持續追蹤您的組織進行中的專案和產品。將您目前的系統與 AWS 提供的一組豐富的活動與指標合併，就能供您檢視重要的生命週期活動，並積極管理資源，以降低不必要的成本。

如需有關 Web 應用程式後端的建議，請參閱 Well-Architected 卓越營運支柱白皮書 。

實作步驟

- 執行工作負載審查： 根據您組織的政策定義，稽核您現有的專案。在稽核上付出的工作量應與組織的大致風險、價值或成本成正比。要納入稽核的關鍵領域包括事件或中斷給組織帶來的風險、對組織的價值或貢獻 (以收入或品牌聲譽來衡量)、工作負載成本 (以資源總成本和營運成本來衡量)，以及工作負載用量 (以每單位時間的組織結果數量來衡量)。如果這些領域在生命週期內發生變化，則需要調整工作負載，例如完整或部分除役。

資源

相關文件：

- AWS Config
- AWS Systems Manager
- 適用於各工作職能的 AWS 受管政策

- AWS 多帳戶帳單策略
- 使用 IAM 政策控制對 AWS 區域 的存取

COST 3 您如何監控用量和成本？

建立原則和程序以監控並適當分配成本。這可讓您衡量並改善此工作負載的成本效益。

最佳實務

- COST03-BP01 設定詳細資訊來源
- COST03-BP02 識別成本歸因類別
- COST03-BP03 建立組織指標
- COST03-BP04 設定帳單和成本管理工具
- COST03-BP05 將組織資訊新增至成本與用量
- COST03-BP06 根據工作負載指標分配成本

COST03-BP01 設定詳細資訊來源

設定 AWS 成本和用量報告，以及 Cost Explorer 每小時精細度，以提供詳細的成本和用量資訊。設定您的工作負載，使其具有每個交付業務成果的日誌項目。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

在 AWS Cost Explorer 中啟用每小時精細程度，並建立 AWS Cost and Usage Report (CUR)。這些資料來源提供整個組織最準確的成本和用量的檢視。CUR 為所有收費的 AWS 服務提供每日或每小時用量精細程度、費率、成本和用量屬性。CUR 中包含所有可能的維度，包括：標記、位置、資源屬性和帳戶 ID。

使用以下自訂項目設定您的 CUR：

- 包含資源 ID
- 自動重新整理 CUR
- 每小時精細度
- 版本控制： 覆寫現有的報告

- 資料整合： Amazon Athena (Parquet 格式和壓縮)

使用 AWS Glue 準備資料用於分析，並使用 Amazon Athena 執行資料分析，使用 SQL 查詢資料。您也可以使用 Amazon QuickSight 建立自訂且複雜的視覺化，並將它們分發到整個組織。

實作步驟

- 設定成本和用量報告： 使用帳單主控台，設定至少一個成本和用量報告。用含所有識別符與資源 ID 的每小時精細度設定報告。您也可以使用不同的精細度建立其他報告，以提供較高層級的摘要資訊。
- 在 Cost Explorer 中設定每小時精細度： 使用帳單主控台，啟用每小時和資源層級資料。

> **Note**
>
> 啟用此功能會產生相關費用。如需詳細資訊，請參閱定價。

- 設定應用程式記錄： 確認您的應用程式會記錄其交付的每個業務成果，以便追蹤和衡量相應成果。確保此資料的精細度至少為每小時，以便與成本和用量資料相符。如需有關 Web 應用程式後端的建議，請參閱 Well-Architected 卓越營運支柱， 以取得有關記錄和監控的詳細資訊。

資源

相關文件：

- AWS 帳戶設定
- AWS Cost and Usage Report (CUR)
- AWS Glue
- Amazon QuickSight
- AWS 成本管理定價
- 標記 AWS 資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告
- Well-Architected 卓越營運支柱，

相關範例：

- AWS 帳戶設定

COST03-BP02 識別成本歸因類別

識別可用於在組織內分配成本的組織類別。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

與您的財務團隊和其他相關利害關係人合作，以了解如何在組織內分配成本的要求。工作負載成本必須在整個生命週期中分配，包括開發、測試、生產和除役。了解組織內學習、員工發展和創意成本的狀況。這有助於將用於此目的的帳戶正確分配到培訓和開發預算，而不是一般 IT 成本預算。

實作步驟

- 定義您的組織類別： 與利害關係人會面，定義可反映組織的結構和要求的類別。這些項目會直接對應至現有財務類別的結構，例如業務單位、預算、成本中心或部門。查看雲端為您的業務交付的成果，例如培訓或教育，因為這些也是組織類別。您可以將多個類別指派給一個資源，而資源可以位於多個不同的類別中，因此請視需要定義任意數量的類別。
- 定義您的功能類別： 與利害關係人會面，定義可反映您業務內具有之功能的類別。這可能是工作負載或應用程式名稱，以及環境類型，例如生產、測試或開發。您可以將多個類別指派給一個資源，而資源可以位於多個不同的類別中，因此請視需要定義任意數量的類別。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告

COST03-BP03 建立組織指標

建立此工作負載所需的組織指標。工作負載的指標範例包括產生的客戶報告或向客戶提供的網頁。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解工作負載的輸出是否算得上業務成功。每個工作負載通常都有少數幾個能夠指出效能的主要輸出。如果您有包含許多元件的複雜工作負載，則可以排定清單的優先順序，或定義和追蹤每個元件的指標。與您的團隊合作，以了解要使用哪些指標。此單位將用於了解工作負載的效率，或每個業務輸出的成本。

實作步驟

- 定義工作負載成果： 與業務中的利害關係人會面，並定義工作負載的成果。這些是客戶用量的主要衡量方式，並且必須是業務指標而非技術指標。每個工作負載應該有少量的高層級指標 (少於五個)。如果工作負載為不同的使用案例產生多個成果，請將它們分組為單一指標。
- 定義工作負載元件成果： 或者，如果您有大型且複雜的工作負載，或者可以用明確定義的輸入和輸出，輕鬆地將工作負載分成元件 (例如微型服務)，則請為每個元件定義指標。工作應該反映元件的價值和成本。從最大的元件開始，並向較小的元件運行。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告

COST03-BP04 設定帳單和成本管理工具

根據組織政策設定 AWS Cost Explorer 和 AWS Budgets。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

若要修改用量和調整成本，組織中的每個人員都必須能夠存取其成本和用量資訊。建議所有工作負載和團隊在使用雲端時設定下列工具：

- 報告： 彙總所有成本和用量資訊
- 通知： 當成本或用量超出定義的限制時，提供通知。

- 目前狀態： 設定儀表板，顯示目前的成本和用量。儀表板應該在工作環境中顯眼的位置提供 (類似營運儀表板)。
- 趨勢： 提供以所需的精細度顯示所需期間內成本與用量變化的功能。
- 預測： 提供顯示未來預估成本的功能。
- 追蹤： 顯示相對於設定的總目標或具體目標目前成本和用量的狀況。
- 分析： 讓團隊成員能夠以所有可能的維度執行每小時精細度的自訂和深度分析。

您可以使用 AWS 原生工具，例如 (AWS Cost Explorer、 AWS Budgets和 Amazon Athena) 搭配 Amazon QuickSight 實現此功能。您也可以使用第三方工具，不過，您必須確保此工具的成本能夠為您的組織帶來相應的價值。

實作步驟

- 建立成本優化群組： 設定您的帳戶，並建立可存取所需成本和用量報告的群組。此群組必須包含擁有或管理應用程式之所有團隊的代表。這可證明每個團隊都能存取他們的成本和用量資訊。
- 設定 AWS Budgets： 針對您的工作負載在所有帳戶上設定 AWS Budgets。使用標籤來設定整體帳戶支出的預算，以及工作負載的預算。
- 設定 AWS Cost Explorer： 為您的工作負載和帳戶設定 AWS Cost Explorer。為追蹤整體支出的工作負載建立儀表板，並建立工作負載的關鍵用量指標。
- 設定進階工具： 或者，您可以為組織建立自訂工具，以提供額外的詳細資訊和精細度。您可以實作進階分析功能，方法為使用 Amazon Athena，以及實作儀表板，方法為使用 Amazon QuickSight。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告

相關範例：

- Well-Architected 實驗室：AWS 帳戶設定
- Well-Architected 實驗室：帳單視覺化

- Well-Architected 實驗室：成本與管控用量
- Well-Architected 實驗室：成本與用量分析
- Well-Architected 實驗室：成本與用量視覺化

COST03-BP05 將組織資訊新增至成本與用量

根據組織、工作負載屬性和成本分配類別來定義標記結構描述。在所有資源上實作標記。使用 Cost Categories 以根據組織屬性將成本與用量分組。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在 AWS 中實作標記 將組織資訊新增到您的資源，然後將這些資訊新增至您的成本與用量資訊。標籤是鍵/值對；鍵已定義，且在組織中必須是唯一的，而值對於一組資源是唯一的。鍵值對的範例：鍵是 Environment (環境)，其值為 Production (生產)。生產環境中的所有資源都會有此鍵/值對。標記可讓您使用有意義且相關的組織資訊，來分類和追蹤成本。您可以套用代表組織類別 (例如成本中心、應用程式名稱、專案或擁有者) 的標籤，並識別工作負載及其特性 (例如，測試或生產)，以在整個組織中劃分成本和用量歸屬。

當您套用標籤至 AWS 資源 (例如 Amazon Elastic Compute Cloud 執行個體或 Amazon Simple Storage Service 儲存貯體) 並啟用標籤時，AWS 會將此資訊加入至成本和用量報告。您可以對已標記和未標記的資源執行報告和分析，以便更符合內部成本管理政策，並確保準確劃分歸屬。

遍及組織的帳戶建立和實作 AWS 標記標準，能讓您以一致統一的方式管理和控管 AWS 環境。使用 用 AWS Organizations 中的標籤政策，定義如何在 AWS Organizations 中將標籤用於帳戶中 AWS 資源的規則。標籤政策可讓您輕鬆採用標準化方法來標記 AWS 資源。

AWS Tag Editor 讓您可以新增、刪除和管理多個資源的標籤。

AWS Cost Categories 讓您可以為成本指派組織的意義，無須在資源上加上標籤。您可以將成本和用量資訊對應到唯一的內部組織結構。您可以定義類別規則，使用帳單維度 (例如帳戶和標籤) 來映射和分類成本。除了標記，這可提供另一個層級的管理功能。您也可以將特定帳戶和標籤對應到多個專案。

實作步驟

- 定義標記結構描述： 聚集整個業務的所有利害關係人來定義結構描述。這通常包括技術、財務和管理角色中的人員。定義所有資源必須具備的標籤清單，以及資源應該具備的標籤清單。確認標籤名稱和值在整個組織中保持一致。

- 標記資源： 使用您定義的成本屬性類別，並根據類別，在工作負載中的所有資源上放置標籤。使用 CLI、Tag Editor 或 Systems Manager 等工具來提高效率。
- 實作 Cost Categories： 您可以建立 Cost Categories 而不實作標記。Cost Categories 會使用現有的成本和用量維度。從您的結構描述建立類別規則，並將其實作至 Cost Categories。
- 自動化標記： 若要確認您在所有資源中保持高層級標記，請自動化標記，以便在建立資源時自動對其進行標記。使用服務內的功能或 AWS CloudFormation 等服務，確保在建立資源時加上標籤。您也可以建立自訂微型服務，該服務會定期掃描工作負載，並移除任何未標記的資源，這非常適合用於測試和開發環境。
- 監控和報告標記： 若要確認您在整個組織中保持高層級標記，請報告並監控工作負載間的標籤。您可以使用 AWS Cost Explorer 檢視已標記和未標記資源的成本，或使用 Tag Editor 等服務。定期檢閱未標記資源的數量，並採取措施來新增標籤，直至達到所需的標記層級。

資源

相關文件：

- AWS CloudFormation 資源標籤
- AWS Cost Categories
- 標記 AWS 資源
- Amazon EC2 和 Amazon EBS 支援您在建立資源時標記資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告

COST03-BP06 根據工作負載指標分配成本

按指標或業務成果分配工作負載的成本，以衡量工作負載的成本效率。實作程序以分析 AWS 成本和用量報告，方法為使用 Amazon Athena，從而獲得洞見和退款功能。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

成本優化以最低的價格提供業務成果，只有依工作負載指標 (依工作負載效率測量) 來分配工作負載成本才能達成。透過記錄檔或其他應用程式監控，監控已定義的工作負載指標。結合此資料與工作負載

成本，您可以透過查看具有特定標籤值或帳戶 ID 的成本來取得成本資料。建議您每小時執行一次此分析。如果您有一些靜態成本元件 (例如，全年無休執行的後端資料庫) 且請求率不同 (例如，上午 9 點到下午 5 點為用量尖峰，夜間請求數量很少)，您的效率通常會有所改變。了解靜態成本與可變成本之間的關係，將協助您確定優化活動的重點。

實作步驟

- 將成本分配到工作負載指標： 使用定義的指標和設定的標記，建立結合工作負載輸出和工作負載成本的指標。使用諸如 Amazon Athena 和 Amazon QuickSight 等分析服務，為整體工作負載和任何元件建立效率儀表板。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS 預算分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和用量報告

COST 4 如何進行資源除役？

從啟動到結束專案期間，控制變更並管理資源。這可確保您關閉或終止未使用的資源，以減少浪費。

最佳實務

- COST04-BP01 在資源生命週期內追蹤資源
- COST04-BP02 實作除役程序
- COST04-BP03 除役資源
- COST04-BP04 自動除役資源

COST04-BP01 在資源生命週期內追蹤資源

定義並實作一種方法，在資源的生命週期內追蹤資源及其與系統的關聯。您可以使用標記來識別資源的工作負載或功能。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

除役不再需要的工作負載資源。一個常見的範例是用於測試的資源，測試完成後，可以移除該資源。使用標籤追蹤資源 (以及執行這些標籤的報告) 可協助您識別要除役的資產。使用標籤是追蹤資源的有效方法，方法是使用資源的功能標記資源，或標記除役日期。然後，即可對這些標籤執行報告。功能標記的值可以是 ## X ### 用來識別資源在工作負載生命週期的用途。

實作步驟

- 實作標記結構描述： 實作識別資源所屬工作負載的標記結構描述，確認工作負載內的所有資源都已相應地加上標籤。
- 實作工作負載輸送量或輸出監控： 實作工作負載輸送量監控或警示，並在輸入請求或輸出完成時觸發。將其設定為在工作負載請求或輸出降至零時提供通知，指示不再使用工作負載資源。如果工作負載在正常條件下定期下降到零，則併入時間因素。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- 標記 AWS 資源
- 發布自訂指標

COST04-BP02 實作除役程序

實作識別和除役孤立資源的程序。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

在您的組織中實作標準化程序，以識別並移除未使用的資源。此程序應該定義執行搜尋的頻率，以及移除資源的程序，以確保符合組織的所有要求。

實作步驟

- 建立並實作除役程序： 與工作負載開發人員和擁有者合作，為工作負載及其資源建置除役程序。此程序應該涵蓋用於驗證工作負載是否正在使用的方法，以及用於驗證每個工作負載資源是否正在使用

的方法。此程序也應涵蓋必要步驟以用於除役資源，並將它們從服務中移除，同時確保符合任何法規要求。此外亦涵蓋任何關聯的資源，例如授權或連接的儲存。此程序應向工作負載擁有者提供除役程序已執行的通知。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor

COST04-BP03 除役資源

除役由諸如定期稽核或用量變更等事件觸發的資源。除役通常會定期執行，而且是手動或自動化的。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

搜尋未使用資源的頻率和努力應該反映潛在節省的成本，因此較低成本的帳戶的分析頻率應該比較大成本帳戶低。搜尋和除役事件可由工作負載的狀態變更觸發，例如產品壽命結束或被取代。搜尋和除役事件也可由外部事件觸發，例如市場條件變化或產品終止。

實作步驟

- 除役資源： 使用除役程序，除役已識別為孤立的每個資源。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor

COST04-BP04 自動除役資源

設計工作負載，在識別和除役非關鍵資源、不需要的資源或低利用率資源時，妥善處理資源終止。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

使用自動化來降低或消除除役程序的相關成本。將工作負載設計為執行自動除役，可降低工作負載生命週期內的整體成本。您可以使用 AWS Auto Scaling 執行除役程序。您也可以使用 API 或開發套件 實作自訂程式碼，自動除役工作負載資源。

實作步驟

- 實作 AWS Auto Scaling： 針對支援的資源，以 AWS Auto Scaling 設定這些資源。
- 設定 CloudWatch 以終止執行個體 執行個體可設定為使用 CloudWatch 警示終止。使用來自於除役程序的指標，以 Amazon Elastic Compute Cloud (Amazon EC2) 動作實作警示。推出之前，確認非生產環境中的操作。
- 在工作負載內實作程式碼： 您可以使用 AWS SDK 或 AWS CLI 來除役工作負載資源。在整合 AWS 的應用程式內實作程式碼，並終止或移除不再使用的資源。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- 建立警示以停止、終止、重新啟動或復原執行個體
- Amazon EC2 Auto Scaling 入門

具有經濟效益的資源

問題

- COST 5 您選擇服務時如何評估成本？
- COST 6 您選擇資源類型、大小和數量時，如何達成成本目標？
- COST 7 您如何使用定價模式降低成本？
- COST 8 您如何規劃資料傳輸費？

COST 5 您選擇服務時如何評估成本？

Amazon EC2、Amazon EBS 和 Amazon S3 是基礎 AWS 服務。Amazon RDS 和 Amazon DynamoDB 等受管服務為更高等級或應用程式等級的 AWS 服務。選擇適當的基礎和受管服務，您便

可為成本優化此工作負載。舉例來說，您可以使用受管服務減少或省下大部分管理和營運開銷，讓您從事應用程式或企業相關活動。

最佳實務

- COST05-BP01 確定組織的成本要求
- COST05-BP02 分析此工作負載的所有元件
- COST05-BP03 對每個元件執行徹底的分析
- COST05-BP04 選取具成本效益授權的軟體
- COST05-BP05 選取此工作負載的元件，以按照組織優先事項來優化成本
- COST05-BP06 對一段時間內的不同用量進行成本分析

COST05-BP01 確定組織的成本要求

與團隊成員一起為此工作負載定義成本最佳化與其他支柱 (例如效能和可靠性) 之間的平衡。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

為工作負載選取服務時，關鍵是了解組織的優先事項。確保您在成本和其他 Well-Architected 支柱之間取得平衡，例如效能和可靠性。完全成本優化的工作負載是最符合您組織需求的解決方案，不一定是成本最低的解決方案。與組織內的所有團隊會面以收集資訊，例如產品、業務、技術和財務團隊。

實作步驟

- 確定組織的成本要求： 與您組織的團隊成員會面，這些成員包括產品管理人員、應用程式擁有者、開發和營運團隊、管理和財務角色。排定此工作負載及其元件的 Well-Architected 支柱優先順序，輸出為依序列出支柱的清單。您也可以為每個支柱新增加權，這可以指出相應支柱有多少個額外焦點，或兩個支柱之間的焦點有多相似。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

COST05-BP02 分析此工作負載的所有元件

確認分析每個工作負載元件，無論當前大小或當前成本如何。審查工作應反映潛在的效益，例如當前和預計的成本。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

對工作負載中的所有元件執行徹底的分析。確保在工作負載生命週期內取得分析成本與潛在節省之間的平衡。您必須找出元件的目前影響和未來潛在影響。例如，如果建議資源的成本是一個月 10 USD，而低於預測的負載不會超過一個月 15 USD，則花一天努力減少 50% (每月 5 USD) 可能會超過系統生命週期內的潛在利益。使用更快速且更有效率的資料型估算，將為此元件建立最佳整體結果。

工作負載可能會隨時間改變，而且如果工作負載架構或用量變化，適當的服務組合可能並非最佳。選擇服務的分析必須納入目前和未來的工作負載狀態以及用量水平。為未來的工作負載狀態或用量實作服務，可減少或消除未來變更所需的工作量，藉此降低整體成本。

AWS Cost Explorer 和 AWS Cost and Usage Report (CUR) 可分析概念驗證 (PoC) 或執行環境的成本。您也可以使用 AWS Pricing Calculator 來估算工作負載成本。

實作步驟

- 列出工作負載元件： 建置所有工作負載元件的清單。這會做為檢查每個元件是否已經過分析的確認清單。所做的工作應反映貴組織優先事項所定義之工作負載的關鍵性。如果有多個資料庫，在功能上將資源分組在一起可提高效率，例如，生產資料庫儲存。
- 排定元件清單的優先順序： 取得元件清單，並依工作順序排定其優先順序。這通常是依最昂貴到最便宜的元件成本排序，或依貴組織優先事項所定義的關鍵性排序。
- 執行分析： 對於清單上的每個元件，檢閱可用的選項和服務並選擇最適合您組織優先事項的選項。

資源

相關文件：

- AWS Pricing Calculator
- AWS Cost Explorer
- Amazon S3 儲存類別
- 雲端產品

COST05-BP03 對每個元件執行徹底的分析

查看每個元件的組織整體成本。透過考慮營運和管理成本來查看整體擁有成本，尤其是在使用受管服務時。審查工作應反映潛在的效益；例如，用於分析的時間與元件成本成正比。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

考慮所節省的時間，讓您的團隊能夠專注於淘汰技術負債、創新和增值功能。例如，您可能需要將內部部署的環境盡快提升和轉移至雲端，稍後進行優化。使用受管服務以去除或降低授權成本所體現的節省也值得探討。受管服務免除了維護服務的營運和管理重擔，讓您專注於創新。此外，因為受管服務以雲端規模運作，可使每次交易或服務的成本較低。

通常受管服務具有屬性，您可設定以確保備充足容量。您必須設定並監控這些屬性，使得額外的容量保持最低程度，並且獲得最大效能。您可使用 AWS Management Console 或 AWS API 和 SDK 來修改 AWS Managed Services 的屬性，以使資源上的需要配合經常變動的需求。例如，可將 Amazon EMR 叢集 (或 Amazon Redshift 叢集上) 節點的數量增加或減少，以便擴展或者縮減。

亦可將多重執行個體裝填到一項 AWS 資源上，進行密度更高的使用。例如，可將多個小資料庫佈建至單一 Amazon Relational Database Service (Amazon RDS) 資料庫執行個體。隨著用量增長，可使用快照和恢復程序，將其中一個資料庫搬遷至專用 Amazon RDS 資料庫執行個體。

將工作負載佈建至受管服務上時，您必須了解調整服務容量的要求。這些要求通常是時間、心力和對一般工作負載運作的影響。佈建的資源必須允許發生任何變更，佈建必要的額外開銷來實現。為了修改服務所需持續投注的心力，利用與系統和監控工具例如 Amazon CloudWatch 相整合的 API 和 SDK，可降低為幾乎是零。

Amazon RDS、Amazon Redshift和 Amazon ElastiCache 提供受管資料庫服務。Amazon Athena、Amazon EMR和 Amazon OpenSearch Service 提供受管分析服務。

AMS 是代表企業客戶和合作夥伴營運 AWS 基礎設施的服務。它提供安全且合規的環境，您可以將工作負載部署至其中。AMS 使用企業雲端營運模型與自動化，讓您符合組織需求、更快速地遷移至雲端，以及降低持續管理成本。

實作步驟

- 執行徹底的分析： 使用元件清單，從最高優先到最低優先順序處理每個元件。對於優先順序更高且成本更高的元件，請執行額外的分析並評估所有可用選項及其長期影響。對於優先順序較低的元件，評估用量的變更是否會變更元件的優先順序，然後執行適當的工作分析。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

COST05-BP04 選取具成本效益授權的軟體

開放原始碼軟體會剔除對工作負載增加大量成本的軟體授權費用。請在需要授權軟體時，避免繫結至任意屬性 (例如 CPU) 的授權，尋找繫結至輸出或成果的授權。這些授權的成本會更接近其提供的效益。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

使用開放原始碼軟體可免除軟體授權的成本。隨著工作負載的大小擴展，這可能會對工作負載成本產生重大影響。測量授權軟體的效益與總成本，以確保您擁有優化的工作負載。模擬授權的任何變更以及這些變更對工作負載成本的影響。如果廠商變更資料庫授權的成本，調查這會如何影響工作負載的整體效率。考慮廠商的歷史定價公告，以了解其產品授權變更趨勢。授權成本也可能獨立於輸送量或用量，例如依硬體擴展的授權 (CPU 綁定授權)。應該避免這些授權，因為成本可能會快速增加，而不會帶來相應結果。

實作步驟

- 分析授權選項： 檢閱可用軟體的授權條款。尋找具備所需功能的開放原始碼版本，以及授權軟體的效益是否超過成本。有利條款將使軟體成本符合其提供的效益。
- 分析軟體供應商： 檢閱來自於廠商的任何歷史定價或授權變更。尋找不符合成果的任何變更，例如，在特定廠商硬體或平台上執行的懲罰性條款。此外，尋找他們執行可能施加的稽核和懲罰的方式。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

COST05-BP05 選取此工作負載的元件，以按照組織優先事項來優化成本

選取所有元件時需考慮成本因素。這包括使用應用程式層級和受管服務，例如 Amazon Relational Database Service (Amazon RDS)、 Amazon DynamoDB、Amazon Simple Notification Service (Amazon SNS) 和 Amazon Simple Email Service (Amazon SES)，以降低整體組織成本。使用無伺服器和容器執行運算，例如 AWS Lambda、用於靜態網站的 Amazon Simple Storage Service (Amazon S3) 和 Amazon Elastic Container Service (Amazon ECS) 建立持續整合/持續部署 (CI/CD) 管道。使用開放原始碼軟體或無授權費用的軟體，將授權成本降到最低：例如，用於運算工作負載的 Amazon Linux，或將資料庫遷移到 Amazon Aurora。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

您可以使用無伺服器或應用程式層級服務，例如 AWS Lambda、Amazon Simple Queue Service (Amazon SQS)， Amazon SNS和 Amazon SES。這些服務讓您無須管理資源，並提供程式碼執行、佇列服務和訊息傳遞功能。另一個好處是，它們可隨用量擴展效能和成本，因此能夠有效率地分配成本和劃分歸屬。

如需無伺服器的詳細資訊，請參閱 Well-Architected 無伺服器應用程式聚焦白皮書。

實作步驟

- 選取每個服務以最佳化成本： 使用您的優先順序清單和分析，選取最符合您組織優先事項的每個選項。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

COST05-BP06 對一段時間內的不同用量進行成本分析

工作負載可能隨時間變更。某些服務或功能在不同的用量層級上更具成本效益。按預計用量隨時間對每個元件執行分析，此工作負載在其整個生命週期內保持成本效益。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

隨著 AWS 發佈新的服務和功能，工作負載的最佳服務可能會改變。所需的努力應與潛在效益相符。工作負載檢閱頻率取決於您的組織需求。如果成本高昂，則更快實作新的服務可節省最多成本，因此更頻繁的檢閱是有利的。觸發檢閱的另一個因素是使用模式變化。用量的重大變更可能表示替代服務更理想。例如，如需較高的資料傳輸速率，直接連線服務可能比 VPN 更便宜，並提供所需的連線能力。預測服務變更的潛在影響，讓您可以監控這些用量等級觸發條件，並更快實作最經濟實惠的服務。

實作步驟

- 定義預測用量模式： 與您的組織 (例如行銷和產品擁有者) 合作，記錄工作負載的預期和預測用量模式。
- 根據預測用量執行成本分析： 使用定義的使用模式，在每個點執行分析。分析工作應該反映潛在成果。例如，如果用量變更很大，則應執行徹底的分析來驗證任何成本和變更。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

COST 6 您選擇資源類型、大小和數量時，如何達成成本目標？

確保您為手上的任務選擇適當的資源大小和資源數量。您透過選擇最具成本效益的類型、大小和數量，最大限度地減少浪費。

最佳實務

- COST06-BP01 執行成本建模
- COST06-BP02 根據資料選取資源類型、大小及數目
- COST06-BP03 根據指標自動選取資源類型、大小和數目

COST06-BP01 執行成本建模

確定組織要求並對工作負載及其每個元件執行成本建模。在不同預測負載下對工作負載執行基準測試活動，並比較成本。建模工作應反映潛在效益。例如，花費的時間與元件成本成正比。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

為您的工作負載及其每個元件執行成本建模，以了解資源之間的平衡，並根據特定效能等級，找出工作負載中每個資源的合適大小。在不同預測負載下對工作負載執行基準測試活動，並比較成本。建模工作應反映潛在效益；例如，花費的時間與元件成本或預測的節省成正比。如需最佳實務，請參閱 檢閱 經濟實惠的 效能達成效率支柱白皮書。

AWS Compute Optimizer 可協助對執行中工作負載進行成本建模。它根據歷史用量，提供運算資源的合適大小建議。這是運算資源的理想資料來源，因為它是免費服務，並利用機器學習根據風險等級提出多個建議。您也可以使用 Amazon CloudWatch 和 Amazon CloudWatch Logs 搭配自訂日誌作為資料來源，以便精簡化其他服務和工作負載元件。

以下是成本建模資料和指標的建議：

- 監控必須準確反映最終使用者的體驗。為時段選擇正確的精細度，並悉心選擇最大或 99%，而非平均值。
- 為分析的時段選擇涵蓋任何工作負載週期所需的正確精細度。例如，假設所執行的是為期兩週的分析，您可能會忽略高利用率的每月週期，導致佈建不足。

實作步驟

- 執行成本建模： 將工作負載或概念驗證部署到具有要測試之特定資源類型和大小的獨立帳戶。使用測試資料執行工作負載，並記錄輸出結果以及測試執行時的成本資料。然後重新部署工作負載或變更資源類型和大小，並重新執行測試。

資源

相關文件：

- AWS Auto Scaling
- Amazon CloudWatch 功能
- 成本優化：Amazon EC2 調整大小
- AWS Compute Optimizer

COST06-BP02 根據資料選取資源類型、大小及數目

根據有關工作負載和資源特性的資料來選擇資源大小或類型。例如，運算、記憶體、輸送量或寫入密集。通常使用工作負載的先前 (內部部署) 版本、文件或其他有關工作負載的資訊來源來進行此選擇。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

根據工作負載和資源特性選擇資源大小或類型，例如，運算、記憶體、輸送量或寫入密集。通常使用成本建模、工作負載的先前版本 (例如內部部署版本)、文件或其他有關工作負載的資訊來源 (白皮書、已發佈的解決方案) 來進行此選擇。

實作步驟

- 根據資料選擇資源： 使用成本建模資料，選擇預期的工作負載用量等級，然後選擇指定的資源類型和大小。

資源

相關文件：

- AWS Auto Scaling
- Amazon CloudWatch 功能
- 成本優化：EC2 調整大小

COST06-BP03 根據指標自動選取資源類型、大小和數目

使用目前執行的工作負載中的指標來選擇正確的大小和類型，以最佳化成本。針對 Amazon Elastic Compute Cloud (Amazon EC2)、Amazon DynamoDB、Amazon Elastic Block Store (Amazon EBS) (PIOPS)、Amazon Relational Database Service (Amazon RDS)、Amazon EMR 和聯網等服務適當地佈建輸送量、大小和儲存。這可透過回饋迴圈 (例如自動調整規模) 或工作負載中的自訂程式碼來完成。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

在工作負載中建立意見回饋迴圈，使用執行中工作負載的作用中指標來變更該工作負載。您可以使用受管服務，例如 AWS Auto Scaling，將其設定為為您執行精簡化操作。AWS 也會提供 API、SDK和

功能，讓修改資源變得非常輕鬆。您可以設定工作負載來停止和啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，以允許變更執行個體大小或執行個體類型。這不僅帶來精簡化的效益，同時消除變更所需的幾乎所有營運成本。

有些 AWS 服務內建自動類型或大小選擇，例如 Amazon Simple Storage Service (Amazon S3) 智慧型分層。Amazon S3 智慧型分層會根據您的使用模式，自動在兩個存取層 (經常存取和不常存取) 之間移動您的資料。

實作步驟

- 設定工作負載指標： 確保擷取工作負載的關鍵指標。這些指標提供客戶體驗 (例如工作負載輸出) 的指示，並符合資源類型和大小 (例如 CPU 和記憶體用量) 之間的差異。
- 檢視適當調整大小的建議： 使用 AWS Compute Optimizer 中的適當調整大小建議來調整您的工作負載。
- 根據指標自動選擇資源類型和大小： 使用工作負載指標，手動或自動選擇您的工作負載資源。在應用程式內設定 AWS Auto Scaling 或實作程式碼，可在需要頻繁變更時減少所需的工作量，而且它可能比手動程序更快地實作變更。

資源

相關文件：

- AWS Auto Scaling
- AWS Compute Optimizer
- Amazon CloudWatch 功能
- CloudWatch 設定
- CloudWatch 發布自訂指標
- 成本優化：Amazon EC2 調整大小
- Amazon EC2 Auto Scaling 入門
- Amazon S3 智慧型分層
- 使用 SDK 啟動 EC2 執行個體

COST 7 您如何使用定價模式降低成本？

使用最適合您資源的定價模式，大幅減少支出。

最佳實務

- COST07-BP01 執行定價模式分析
- COST07-BP02 根據成本實作區域
- COST07-BP03 選取具成本效益條款的第三方協議
- COST07-BP04 針對此工作負載的所有元件實作定價模式
- COST07-BP05 在主要帳戶層級執行定價模式分析

COST07-BP01 執行定價模式分析

分析工作負載的每個元件。判斷元件與資源是否會執行較長期間 (針對承諾折扣)，或動態與短期執行 (針對 Spot 或隨需執行個體)。使用 AWS Cost Explorer 中的建議功能對工作負載執行分析。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

AWS 提供多種 定價模式 ，可讓您以最經濟實惠的方式為資源付費，以符合組織需求。

實作步驟

- 執行承諾折扣分析： 在您的帳戶中使用 Cost Explorer，檢閱 Savings Plans 和預留執行個體建議。若要確認您在承擔相應風險的同時以所需折扣實作正確的建議，請遵循 Well-Architected 實驗室的指示進行。
- 分析工作負載彈性： 使用 Cost Explorer 中的每小時精細度，或自訂儀表板。分析工作負載彈性。尋找正在執行的執行個體數量的定期變更。短期執行個體是 Spot 執行個體或 Spot 叢集的候選項目。
  - Well-Architected 實驗室：Cost Explorer
  - Well-Architected 實驗室：成本視覺化

資源

相關文件：

- 存取預留執行個體的推薦
- 執行個體購買選項

相關影片：

- 節省高達 90% 的成本並在 Spot 執行生產工作負載

相關範例：

- Well-Architected 實驗室：Cost Explorer
- Well-Architected 實驗室：成本視覺化
- Well-Architected 實驗室：定價模式

COST07-BP02 根據成本實作區域

每個區域的資源定價可能不同。考慮區域成本，可協助確保您為此工作負載支付最低的總價。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

當您建構解決方案時，一項最佳實務是盡量將運算資源置於接近使用者之處，以提供較低延遲和強大的資料主權。對於全球受眾，應使用多重位置以滿足這類需要。應選擇能使成本最低的地理位置。

AWS 雲端基礎設施的建置基礎為 區域和可用區域。區域是世界上有多個可用區域的實體位置。可用區域由一或多個分散的資料中心所組成，每個都有備援電源、聯網和連線能力，且置放在不同的機構。

每個 AWS 區域各在當地市場條件之下運作，各區域的資源定價不同。您可以選擇特定區域以操作解決方案的元件或全部，以便以最低價格於全球執行。您可以使用 AWS Pricing Calculator 估算各個區域的工作負載成本。

實作步驟

- 審查區域定價： 分析目前區域的工作負載成本。依服務和用量類型，從最高成本開始，計算其他可用區域的成本。如果預測儲存超過移動元件或工作負載的成本，請遷移至新區域。

資源

相關文件：

- 存取預留執行個體的推薦
- Amazon EC2 定價
- 執行個體購買選項

- 區域表

相關影片：

- 節省高達 90% 的成本並在 Spot 執行生產工作負載

COST07-BP03 選取具成本效益條款的第三方協議

具成本效益的協議和條款可確保這些服務的成本隨其提供的優勢而擴展。選擇可在為您的組織提供額外優勢時擴展的協議和定價。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

當您在雲端使用第三方解決方案或服務時，定價結構必須符合成本優化的成果。定價應根據其提供的結果和價值進行擴展。例如，軟體從節省的成本中提取一定比例，節省得 (成果) 越多，收費就越高。依帳單擴展的協議通常不符合成本優化，除非它們為特定帳單每個部分帶來成果。例如，對於提供 Amazon Elastic Compute Cloud (Amazon EC2) 建議且收費整個帳單一定百分比作為費用的解決方案，如果您使用該解決方案無法提供優勢的其他服務，則成本會上升。另一個範例是受管服務，依受管資源成本百分比計費。較大的執行個體大小不一定需要更多的管理工作，但收費更高。確保這些服務定價安排在其服務中包含成本優化計劃或功能，以提升效率。

實作步驟

- 分析第三方協議和條款： 審查第三方協議中的定價。針對不同的用量等級執行建模，並將新成本納入考量，例如新服務用量，或因工作負載成長而產生的目前服務增加量。決定額外成本是否為您的企業提供所需的優勢。

資源

相關文件：

- 存取預留執行個體的推薦
- 執行個體購買選項

相關影片：

- 節省高達 90% 的成本並在 Spot 執行生產工作負載

COST07-BP04 針對此工作負載的所有元件實作定價模式

永久執行的資源應使用預留容量，例如 Savings Plans 或預留執行個體。設定短期容量以使用 Spot 執行個體或 Spot 機群。隨需執行個體僅用於無法中斷且執行時間不夠長，以及不適合使用預留容量的短期工作負載 (介於 25% 到 75% 之間的時間，視資源類型而定)。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

考慮工作負載元件的需求，並了解潛在的定價模式。定義元件的可用性需求。判斷是否有多個獨立資源在工作負載中執行相同功能，以及隨時間工作負載需求的變化。比較使用預設隨需定價模式和其他適用的模式的資源成本。考量資源或工作負載元件的任何潛在變更。

實作步驟

- 實作定價模式： 使用分析結果購買 Savings Plans (SP)、預留執行個體 (RI) 或實作 Spot 執行個體。如果是第一次購買 RI，請選擇清單中的前 5 或 10 項建議，然後監控和分析下個月或未來兩個月的結果。定期購買少量承諾折扣，例如每兩週或每月。針對可能中斷或無狀態的工作負載，實作 Spot 執行個體。
- 工作負載審查週期： 實作工作負載的審查週期，特別分析定價模型涵蓋範圍。一旦工作負載達到所需的涵蓋範圍，請每二至四週購買額外的承諾折扣，或隨著組織用量變更進行購買。

資源

相關文件：

- 存取預留執行個體的推薦
- EC2 Fleet
- 如何購買預留執行個體
- 執行個體購買選項
- Spot 執行個體

相關影片：

- 節省高達 90% 的成本並在 Spot 執行生產工作負載

COST07-BP05 在主要帳戶層級執行定價模式分析

使用 Cost Explorer Savings Plans 和預留執行個體建議，在管理帳戶層級針對承諾折扣執行定期分析。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

執行定期成本建模，可確保跨多個工作負載進行優化。例如，如果多個工作負載使用隨需執行個體，則在彙總層級變更的風險會更低，而且實作以承諾為基礎的折扣可獲得更低的整體成本。建議以兩週到一個月的頻率定期執行分析。這可讓您進行小幅的調整，因此定價模式的涵蓋範圍會隨著不斷變化的工作負載及其元件不斷演變。

使用 AWS Cost Explorer 建議工具，尋找承諾折扣的機會。

若要尋找 Spot 工作負載的機會，可使用整體用量的每小時檢視，並尋找定期出現用量或彈性變化的時段。

實作步驟

- 執行承諾折扣分析： 在您的帳戶中使用 Cost Explorer，檢閱 Savings Plans 和預留執行個體建議。若要確認在承擔相應風險的同時以所需折扣實作正確的建議，請遵循 Well-Architected 實驗室的指示進行。

資源

相關文件：

- 存取預留執行個體的推薦
- 執行個體購買選項

相關影片：

- 節省高達 90% 的成本並在 Spot 執行生產工作負載

相關範例：

- Well-Architected 實驗室：定價模式

COST 8 您如何規劃資料傳輸費？

務必規劃和監控資料傳輸費，以便做出可大幅減少成本的架構決策。小但有效的架構變更可隨時間大幅減少營運成本。

最佳實務

- COST08-BP01 執行資料傳輸建模
- COST08-BP02 選擇元件以優化資料傳輸成本
- COST08-BP03 實作可降低資料傳輸成本的服務

COST08-BP01 執行資料傳輸建模

收集組織要求並執行工作負載及其每個元件的資料傳輸建模。這可確定其目前資料傳輸要求的最低成本點。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解資料傳輸在工作負載中的位置、傳輸成本及其相關效益。這可讓您做出明智的決策，以修改或接受架構決策。例如，您可能有一個多個可用區域組態，您在可用區域之間複寫資料。您要建立結構成本模型，並決定這是實現所需可靠性和彈性可接受的成本 (類似於在兩個可用區域中支付運算和儲存費用)。

針對不同用量等級建立成本模型。工作負載用量會隨時間改變，在不同等級，不同的服務可能更經濟實惠。

使用 AWS Cost Explorer 或 AWS Cost and Usage Report (CUR) 以了解資料傳輸成本並建模。設定概念驗證 (PoC) 或測試工作負載，並以逼真的模擬負載執行測試。您可以根據不同的工作負載需求建立成本模型。

實作步驟

- 計算資料傳輸成本： 使用 AWS 定價頁面 計算工作負載的資料傳輸成本。針對工作負載用量的增加和減少，計算不同用量等級的資料傳輸成本。如果工作負載架構具有多個選項，請計算每個選項的成本進行比較。
- 將成本與結果連結： 對於產生的每筆資料傳輸成本，請指定工作負載達到的結果。如果在元件之間傳輸，可能是用於解耦，如果在可用區域之間傳輸，則可能是用於冗餘。

資源

相關文件：

- AWS 快取解決方案
- AWS 定價
- Amazon EC2 定價
- Amazon VPC 定價
- 使用 Amazon CloudFront 更快地交付內容

COST08-BP02 選擇元件以優化資料傳輸成本

選擇所有元件，並設計架構以降低資料傳輸成本。這包括使用廣域網路 (WAN) 優化和多可用區域 (AZ) 組態等元件

若未建立此最佳實務，暴露的風險等級： 低

實作指引

為資料傳輸建構，可確保您將資料傳輸成本降至最低。這可能涉及使用內容交付網路以將資料靠近使用者放置，或從您內部至 AWS 使用專用網路連結。您也可以使用 WAN 優化和應用程式優化，來減少元件之間傳輸的資料量。

實作步驟

- 選擇用於資料傳輸的元件： 使用資料傳輸模型，專注於資料傳輸成本最高的位置或工作負載用量變更時資料傳輸成本最高的位置。尋找替代架構或其他元件，以消除或降低資料傳輸需求或降低成本。

資源

相關文件：

- AWS 快取解決方案
- 使用 Amazon CloudFront 更快地交付內容

COST08-BP03 實作可降低資料傳輸成本的服務

實作服務以降低資料傳輸。例如，使用 Amazon CloudFront 之類的內容交付網路 (CDN) 向最終使用者交付內容，使用 Amazon ElastiCache 快取層，或者使用 AWS Direct Connect 代替 VPN 連線到 AWS。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

Amazon CloudFront 是一個全球內容交付網路，在低延遲和高傳輸速度之下遞送資料。其快取位於全球節點的資料，能減輕您的資源所受的負載。藉由 CloudFront，在最低延遲之下交付內容給全球大量使用者方面，您可減少管理所費的心力。

AWS Direct Connect 服務可讓您建立連接至 AWS 的專用網路連線。如此可降低網路成本，增加頻寬，並且比網際網路連線提供更一致的網路體驗。

AWS VPN 可讓您在私有網路和 AWS 全球網路之間建立安全且私有的連線。它非常適合小型辦公室或商業合作夥伴，因為它提供快速且容易使用的連線，而且是全受管的彈性服務。

VPC 端點 允許透過私有網路連接各 AWS 服務，可用於降低公有網路的資料傳輸量和 NAT 閘道的成本。閘道 VPC 端點 不收取小時費用，且支援 Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB。界面 VPC 端點 由 AWS PrivateLink 提供，收取小時費用和每 GB 使用費。

實作步驟

- 實作服務： 使用資料傳輸模型，查看成本最高和磁碟區流量最大的位置。檢閱 AWS 服務，並評估是否有可減少或移除傳輸的服務，特別是聯網和內容交付方面。另請尋找可重複存取資料或大量資料的快取服務。

資源

相關文件：

- AWS Direct Connect
- AWS 探索我們的產品
- AWS 快取解決方案
- Amazon CloudFront
- 使用 Amazon CloudFront 更快地交付內容

管理需求與供應資源

問題

- COST 9 如何管理需求和供應資源？

COST 9 如何管理需求和供應資源？

針對支出和效能達到平衡的工作負載，請確保使用購買的每個項目，並避免極少使用執行個體。往任一端傾斜的使用指標，對您組織在營運成本 (因過度使用而降低效能) 或浪費的 AWS 花費 (因過度佈建) 方面會造成負面影響。

最佳實務

- COST09-BP01 對工作負載需求進行分析
- COST09-BP02 實作緩衝或調節機制來管理需求
- COST09-BP03 動態提供資源

COST09-BP01 對工作負載需求進行分析

分析工作負載隨時間的需求。確認分析涵蓋季節性趨勢，並準確反映整個工作負載生命週期內的運作狀況。分析工作應反映潛在效益：例如，花費的時間與工作負載成本成正比。

若未建立此最佳實務，暴露的風險等級： 高

實作指引

了解工作負載的需求。組織要求應指出請求的工作負載回應時間。回應時間可用來判斷需求是否已得到滿足，或是資源供應是否需要改變以符合需求。

分析應包含需求的可預測性和重複性、需求的變化速率，以及需求的變化量。確保分析針對足夠長的時間執行，以納入任何季節變化，例如月底處理或節假日尖峰。

確保分析工作與實作擴展的潛在效益相符。查看元件的預期總成本，以及工作負載生命週期內用量和成本的任何增加或減少。

您可以使用 AWS Cost Explorer 或者 Amazon QuickSight 搭配 AWS Cost and Usage Report (CUR) 或應用程式日誌，以視覺化的方式分析工作負載需求。

實作步驟

- 分析現有的工作負載資料： 分析現有工作負載、舊版工作負載或預測使用模式中的資料。利用日誌檔和監控資料來深入了解客戶使用工作負載的方式。典型指標包括實際需求 (每秒請求數)、需求率變更或處於不同等級的次數，以及需求變更率。請務必分析工作負載的完整週期，藉此確保收集到所有季節性變更的資料，例如月末或年末事件。分析中所反映的工作應反映工作負載特性。應將工作重點放在需求變更最大的高價值工作負載上。針對需求變更最少的低價值工作負載，應將投入的工作量降到最低。常見價值指標包括風險、品牌知名度、收入或工作負載成本。
- 預測外部影響： 與整個組織中的團隊成員面談，這些成員可能會影響或變更工作負載的需求。常見的團隊是銷售團隊、行銷團隊或業務開發團隊。與這些團隊合作以了解其作業週期，以及是否有任何事件會改變工作負載需求。利用此資料來預測工作負載需求。

資源

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon SQS 入門
- AWS Cost Explorer
- Amazon QuickSight

COST09-BP02 實作緩衝或調節機制來管理需求

緩衝和調節機制會修改工作負載的需求，以消除任何尖峰時段。在用戶端執行重試時實作調節機制。實作緩衝機制以儲存請求，並將處理的時間往後延遲。確認調節和緩衝區經過設計，以便讓用戶端在所需時間內收到回應。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

調節： 如果需求來源具有重試功能，則您可以實作調節。調節會告知來源，如果目前無法服務請求，則應稍後再試。來源將等待一段時間，然後重新嘗試請求。實作調節的優點是限制最大資源量和工作負載成本。在 AWS 中，您可以使用 Amazon API Gateway 實作調節。如需實作調節的詳細資訊，請參閱 Well-Architected 可靠性支柱白皮書 。

緩衝為主： 與調節類似，緩衝會延遲請求處理，讓以不同速率執行的應用程式能夠有效地通訊。緩衝為主方法使用佇列來接受生產者傳出的訊息 (工作單位)。消費者可讀取訊息並進行處理，允許以符合消

費者業務要求的速度運作訊息。不必擔心生產者必須應付調節問題，例如資料耐用性和回壓 (由於消費者運作緩慢而導致生產者慢下來)。

在 AWS 中，有多重服務可供選擇以實作緩衝方法。Amazon Simple Queue Service (Amazon SQS) 是一個受管服務，可提供佇列，允許單一消費者讀取個別訊息。Amazon Kinesis 可提供串流，允許許多消費者讀取相同訊息。

使用緩衝為主方法架構時，確保您的工作負載可在所需的時間內為請求提供服務，並且能夠處理重複的工作請求。

實作步驟

- 分析用戶端要求： 分析用戶端請求，以判斷其是否能夠執行重試。針對無法執行重試的用戶端，則需要實作緩衝區。分析整體需求、變更率及所需的回應時間，以便判斷所需的調節或緩衝區大小。
- 實作緩衝區或調節： 在工作負載中實作緩衝區或調節。Amazon Simple Queue Service (Amazon SQS) 這類佇列可為工作負載元件提供緩衝區。Amazon API Gateway 可為您的工作負載元件提供調節機制。

資源

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon API Gateway
- Amazon Simple Queue Service
- Amazon SQS 入門
- Amazon Kinesis

COST09-BP03 動態提供資源

資源是按計畫進行佈建。這可以是以需求為基礎 (例如，透過自動調整規模)，或是以時間為基礎，其中需求可預測，並且根據時間提供資源。這些方法可儘量減少過度佈建或佈建不足的數量。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

您可以使用 AWS Auto Scaling，或使用 AWS API 或 SDK。透過消除手動變更環境所需的營運成本，這可讓您降低整體工作負載成本，而且執行速度更快。這可確保工作負載資源隨時都能最符合需求。

需求為主的供應： 利用雲端的彈性來供應資源，以滿足不斷變化的需求。藉由取得 API 或服務功能帶來的優勢，以程式設計方式動態性地更動架構中的雲端資源量。如此可允許您增減架構中元件的規模，需求激增時自動增加資源數量以維持效能，待需求消退時減少容量以降低成本。

AWS Auto Scaling 可協助您調整容量，以盡可能最低的成本維持穩定、可預測的效能。它是免費的全受管服務，與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和 Spot 叢集、Amazon Elastic Container Service (Amazon ECS)、Amazon DynamoDB 和 Amazon Aurora 整合。

Auto Scaling 提供自動資源探索，以協助尋找工作負載中可設定的資源，它具有內建的擴展策略以優化效能、成本或兩者之間的平衡，並提供預測擴展以協助處理定期發生的尖峰。

Auto Scaling 可以實作手動、排程或以需求為基礎的擴展。您也可以使用 Amazon CloudWatch 的指標和警示，來觸發工作負載的擴展事件。典型的指標可以是標準 Amazon EC2 指標，例如 CPU 使用率、網路輸送量和 Elastic Load Balancing (ELB) 觀察到的請求/回應延遲。可能時，您應該使用可指示客戶體驗的指標，通常此指標為自訂指標，可能源自您工作負載內的應用程式程式碼。

以需求為主的方法進行建構時，請牢記兩大考量要點。第一，了解必須多迅速地佈建起新的資源。第二，了解供應與需求之間差距的大小會改變。您必須隨時因應需求的改變速度，並為資源失敗做好準備。

ELB 可跨多個資源分配需求以協助您進行擴展。當您實作更多資源時，您可將其新增至負載平衡器以接受需求。Elastic Load Balancing 具有 Amazon EC2 執行個體、容器、IP 地址和 AWS Lambda 函數的支援。

時間為主的供應： 時間為主方法能使資源容量符合可預測或依照時間定義完善的需求。這種方法通常不依存於資源的利用率。時間為主方法能確保需要資源的特定時間有資源可用，並且因為啟動程序和系統或一致性檢查的緣故，能在毫無延遲之下提供。採用時間為主方法，您可在忙碌期提供更多資源或增加容量。

您可以使用排程的 Auto Scaling 實作時間為主方法。可排定工作負載於定義的時間縮放規模 (例如在營業時段開始時) 如此可確保使用者或者需求到來時有資源可用。

您也可善用 AWS API 和 SDK 和 AWS CloudFormation 以視需要自動佈建整個環境以及除役。這種方法十分適合僅在定義的營業時段或時期執行的開發或測試環境。

您可使用 API 縮放環境之內的資源大小 (垂直縮放)。例如，可變更執行個體的大小或類別，以擴展生產工作負載。作法是將執行個體停止再啟動，選擇不同的執行個體大小或類別。此技法亦可套用至其他

資源，例如 Amazon Elastic Block Store (Amazon EBS) Elastic Volumes，在使用中時經過修改可增加大小、調整效能 (IOPS) 或變更磁碟區類型。

以時間為主的方法進行建構時，請牢記兩大考量要點。首先，用量模式的一致性有多高？ 第二，若是模式改變會有何影響？ 您可藉由監控工作負載和使用商業智慧來提高預測的準確性。若看出用量模式有明顯變化，可調整時間以確保涵蓋。

實作步驟

- 設定以時間為基礎的排程： 針對可預測的需求變更，以時間為基礎的擴展機制可以及時提供正確的資源數目。此外，當資源建立和設定的速度不夠快，不足以回應隨需變更時，此機制也能派上用場。透過 AWS Auto Scaling，使用工作負載分析來設定排定的擴展。
- 設定 Auto Scaling： 若要根據作用中的工作負載指標來設定擴展，請使用 Amazon Auto Scaling。使用分析和設定 Auto Scaling 以便在正確的資源層級上觸發，並確保工作負載在所需時間內擴展。

資源

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon EC2 Auto Scaling 入門
- Amazon SQS 入門
- Amazon EC2 Auto Scaling 的排程擴展

隨時間優化

問題

- COST 10 您如何評估新服務？

COST 10 您如何評估新服務？

隨著 AWS 推出新服務和功能，最佳實務是檢視現有架構決策，以確保持續發揮最大成本效益。

最佳實務

- COST10-BP01 制定工作負載審查程序

- COST10-BP02 定期審查和分析此工作負載

COST10-BP01 制定工作負載審查程序

制定一個程序，用於定義工作負載審查的標準和程序。審查工作應反映潛在的效益。例如，核心工作負載或價值超過賬單 10% 的工作負載每季度進行審查，而低於 10% 的工作負載則每年進行審查。

若未建立此最佳實務，暴露的風險等級為： 高

實作指引

若要確認永遠擁有最經濟實惠的工作負載，您必須定期審查工作負載，以了解是否有機會實作新的服務、功能和元件。為了確保您實現低整體成本，程序必須與可能的節省金額成正比。例如，相較於佔整體支出 5% 的工作負載，應該更頻繁且更徹底地檢閱佔整體支出 50% 的工作負載。考量任何外部因素或波動性。如果工作負載服務特定的地理或市場區隔，並且預測該區域會發生改變，則更頻繁的檢閱可能會帶來成本節省。需要檢閱的另一個因素是實作變更的工作量。如果測試與驗證變更需要付出大量成本，則應降低檢閱頻率。

考量維護過時和舊版元件和資源的長期成本，以及無法在其中實作新的功能。目前的測試和驗證成本可能會超過提議的效益。不過，隨著時間推移，工作負載與目前技術之間的差距增大，從而變更的成本可能會大幅增加，進而產生更高的成本。例如，移至新的程式設計語言目前看來可能並非具有成本效益之舉。不過，在五年後，該語言熟練人員的成本可能會增加，而且由於工作負載的成長，您會將更大的工作負載轉移到新的語言，此時需要付出的努力會比以前更多。

將您的工作負載細分成多個元件，指派元件的成本 (估算值就足夠)，然後在每個元件旁列出因素 (例如，工作量和外部市場)。使用這些指標來決定每個工作負載的檢閱頻率。例如，您可能會將 Web 伺服器視為高成本、變更所需工作量低和受外部因素影響高，因此檢閱頻率高。中央資料庫可能是中等成本、變更所需工作量高，以及受外部因素影響低，因此檢閱頻率中等。

實作步驟

- 定義審查頻率： 定義工作負載及其元件的審查頻率。這結合了許多因素，可能因組織內的工作負載而異，也可能因工作負載中的元件而異。常見的因素包括：在收入或品牌方面對組織的重要性、執行工作負載的總成本 (包括營運和資源成本)、工作負載的複雜性、實作變更的簡易性、任何軟體授權合約，以及因懲罰性授權，變更會導致授權成本大幅增加。元件可在功能或技術上進行定義，例如 Web 伺服器和資料庫，或運算和儲存資源。相應平衡這些因素，並為工作負載及其元件制定一個期間。您可以決定每 18 個月審查一次完整工作負載、每 6 個月審查一次 Web 伺服器、每 12 個月審查一次資料庫、每 6 個月審查一次運算和短期儲存，以及每 12 個月審查一次長期儲存。
- 定義審查完整性： 定義審查工作負載或工作負載元件所需的工作量。與審查頻率類似，這需在多個因素之間取得平衡。您可以決定在資料庫元件上花費一週進行分析，以及花費四小時進行儲存審查。

資源

相關文件：

- AWS 新聞部落格
- 雲端運算的類型
- AWS 最新消息

COST10-BP02 定期審查和分析此工作負載

根據每個定義的程序定期審查現有的工作負載。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

若要取得新的 AWS 服務和功能帶來的效益，您必須對工作負載執行檢閱程序，並視需要實作新的服務和功能。例如，您可以檢閱工作負載，並使用 Amazon Simple Email Service (Amazon SES) 取代簡訊元件。這消除了營運和維護執行個體叢集的成本，同時以較低的成本提供所有功能。

實作步驟

- 定期審查工作負載： 使用您定義的程序，以指定的頻率執行審查。確認您在每個元件上付出正確的工作量。此程序與您選取服務來進行成本優化的初始設計程序類似。分析服務以及服務會帶來的效益，此時需考慮變更成本，而不僅僅是長期效益。
- 實作新的服務： 如果分析結果是要實作變更，請先執行工作負載的基準，以了解每個輸出的目前成本。實作變更，然後執行分析以確認每個輸出的新成本。

資源

相關文件：

- AWS 新聞部落格
- 雲端運算的類型
- AWS 最新消息

永續性

主題

- 區域選擇
- 使用者行為模式
- 軟體和架構模式
- 資料模式
- 硬體模式
- 開發與部署程序

區域選擇

問題

- SUS 1 如何選擇區域來支持您的永續性發展目標？

SUS 1 如何選擇區域來支持您的永續性發展目標？

根據您的業務需求和永續性發展目標，選擇要在其中實作工作負載的區域。

最佳實務：

SUS01-BP01 選擇 Amazon 可再生能源專案附近的區域，以及電網公佈的碳強度低於其他位置 (或區域) 的區域

選擇 Amazon 可再生能源專案附近的區域，以及電網公佈的碳強度低於其他位置 (或區域) 的區域。

若未建立此最佳實務，暴露的風險等級為： 中

實作指引

選擇 Amazon 可再生能源專案附近的區域，以及電網公佈的碳強度低於其他位置 (或區域) 的區域。

資源

相關文件：

- 全球 Amazon
- 可再生能源方法
- 為工作負載選取區域時應考慮的事項

使用者行為模式

問題

- SUS 2 如何利用使用者行為模式來支持您的永續性發展目標？

SUS 2 如何利用使用者行為模式來支持您的永續性發展目標？

使用者使用工作負載和其他資源的方式，可協助您找到改善的機會，以達成永續性目標。擴展基礎設施以持續符合使用者負載，同時確保僅部署支援使用者所需的最低資源。讓服務層級符合客戶需求。妥善放置資源，以限制使用者使用資源所需的網路。移除現存未使用的資產。識別未使用的已建立資產並停止產生這些資產。為團隊成員提供滿足其需求的裝置，同時將對永續性的影響降至最低。

最佳實務：

SUS02-BP01 隨使用者負載擴展基礎架構

識別使用率低或無使用率的期間，並縮減資源規模以消除過剩容量、提高效率。

常見的反模式：

- 您不隨著使用者負載擴展基礎架構。
- 您一律手動擴展基礎架構。
- 您在擴展事件之後維持增加容量，而不是縮減規模。

建立此最佳實務的優勢： 設定和測試工作負載彈性，將有助於降低工作負載環境受到的影響、節省金錢，以及維護效能基準。您可以利用雲端中的彈性，在使用者負載尖峰期間或之後自動擴展容量，以確保您使用的資源數量正好足以滿足客戶需求，不會超過。

未建立此最佳實務時的曝險等級： 中

實作指引

- 彈性會比對您擁有的資源供應與這些資源的需求。執行個體、容器和函數提供了彈性機制，可與自動擴展功能結合使用，或是作為服務功能提供。利用架構中的彈性，確保工作負載可在使用者負載較低的時段輕易地迅速縮減規模：
  - 使用 Amazon EC2 Auto Scaling 確認您擁有正確數量的 Amazon EC2 執行個體可處理應用程式的使用者負載。

  - 使用 Application Auto Scaling 自動將個別 AWS 服務的資源擴展到 Amazon EC2 以外，例如 Lambda 函數或 Amazon Elastic Container Service (Amazon ECS) 服務。
  - 使用 Kubernetes Cluster Autoscaler 自動擴展 AWS 上的 Kubernetes 叢集。
- 確認會對要部署的工作負載類型驗證擴充或縮減規模的指標。如果您要部署影片轉碼應用程式，則預期為 100% CPU 使用率，且不應做為您的主要指標。您可以將 自訂指標 (例如記憶體使用率) 用於擴展政策 (如有必要)。若要選擇正確的指標，請考量 Amazon EC2 的下列指引：
  - 指標應為有效的使用率指標，並說明執行個體的忙碌程度。
  - 指標值必須根據 Auto Scaling 群組中的執行個體數量按比例增加或減少。
- 使用 動態擴展 而非 手動擴展 處理您的 Auto Scaling 群組。我們也建議您將 目標追蹤擴展政策 用於動態擴展中。
- 確認工作負載部署可同時處理擴充規模和縮減規模事件。建立縮減事件的測試案例，以確保工作負載如預期般運作。您可以使用 活動歷史 測試並驗證 Auto Scaling 群組的擴展活動。
- 評估工作負載以取得可預測模式，並在預計發生預測中的變化和隨需規劃變化時主動擴展。使用 Amazon EC2 Auto Scaling 的預測擴展 消除過度改進容量的需求。

資源

相關文件：

- Amazon EC2 Auto Scaling 入門
- EC2 的預測擴展，採用機器學習技術
- 使用 Amazon OpenSearch Service、Amazon Data Firehose 和 Kibana 分析使用者行為
- 什麼是 Amazon CloudWatch？
- 什麼是 AWS X-Ray？
- VPC Flow Logs
- 在 Amazon RDS 上使用 Performance Insights 監控資料庫負載
- 介紹對於 Amazon EC2 Auto Scaling 預測擴展的原生支援
- 如何根據記憶體使用率指標建立 Amazon EC2 Auto Scaling 政策 (Linux)
- 介紹 Karpenter - 一個開放原始碼的高效能 Kubernetes Cluster Autoscaler

相關影片：

- 更好、更快、更便宜的運算：成本優化 Amazon EC2 (CMP202-R1)

相關範例：

- 實驗室：Amazon EC2 Auto Scaling 群組範例
- 實驗室：使用 Karpenter 實作自動擴展

SUS02-BP02 讓 SLA 符合永續性目標

定義和更新服務水準協議 (SLA)，例如可用性或資料保留期，以將支援工作負載所需的資源數量降至最低，同時繼續滿足業務需求。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 定義支援永續性目標，同時滿足您業務需求的 SLA。
- 重新定義 SLA 以符合業務需求，但不超過它們。
- 做出能大幅降低永續性影響的取捨，換取可接受的服務水準降低。
- 使用優先執行業務關鍵功能的和設計模式，對於非關鍵功能允許採用較低的服務層級 (例如回應時間或復原時間目標)。

資源

相關文件：

- AWS 服務水準協議 (SLA)
- 服務水準協議對 SaaS 供應商的重要性

相關影片：

- 在 AWS 上建立永續性

SUS02-BP03 停止建立和維護不使用的資產

分析應用程式資產 (例如預先編譯的報告、資料集和靜態影像) 和資產存取模式，找出冗餘、未充分利用和可以除役的目標。合併具有冗餘內容的產生資產 (例如，具有重疊或通用資料集與輸出的每月報告)，以避免重複輸出時消耗資源。將未使用的資產除役 (例如不再販售產品的影像) 以釋放消耗的資源，並減少用於支援工作負載的資源數量。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 管理靜態資產並移除不再需要的資產。
- 管理產生的資產，以及停止產生並移除不再需要的資產。
- 合併重疊產生的資產以消除冗餘處理。
- 指示第三方停止生產和儲存代表您管理但不再需要的資產。
- 指示第三方合併代表您生產的冗餘資產。

資源

相關文件：

- 優化您的 AWS 永續性架構，第 II 部分：儲存

相關影片：

- 在 AWS 上建立永續性

SUS02-BP04 針對使用者位置最佳化工作負載的地理位置

分析網路存取模式，識別客戶的地理連接位置。選擇可減少網路流量傳輸距離的區域和服務，以減少支援工作負載所需的總網路資源。

常見的反模式：

- 您可以根據自身所在位置選取工作負載的區域。

建立此最佳實務的優勢： 將工作負載分配到客戶附近的位置，可提供最低的延遲，同時減少網路間的資料移動，並降低環境影響。

未建立此最佳實務時的曝險等級： 中

實作指引

- 根據下列關鍵元素，為您的工作負載部署選取區域：
  - 您的永續目標： 相關說明請見 區域選擇。

  - 資料所在位置： 對於資料密集型應用程式 (例如大數據和機器學習)，應用程式碼執行時應盡可能接近資料。
  - 使用者所在的位置： 對於面向使用者的應用程式，請選擇接近工作負載客群的區域。
  - 其他限制： 請考量安全性和合規性等限制，相關說明請見 為工作負載選取區域時應考慮的事項。
- 使用 AWS Local Zones 執行諸如影片轉譯和圖形密集型虛擬桌面應用程式之類的工作負載。Local Zones 可讓您因運算和儲存資源更接近最終使用者而獲益。
- 使用本機快取或 AWS 快取解決方案 取得常用的資源，以提升效能、減少資料移動，以及降低環境影響。
  - 使用 Amazon CloudFront 快取靜態內容 (例如影像、指令碼和影片) 以及動態內容 (例如 API 回應或 Web 應用程式)。
  - 使用 Amazon ElastiCache 快取 Web 應用程式的內容。
  - 使用 DynamoDB Accelerator 將記憶體內加速新增至 DynamoDB 資料表。
- 使用可協助您在更接近工作負載使用者的位置執行程式碼的服務：
  - 使用 Lambda@Edge 處理在物件未經快取時所執行的大量運算作業。
  - 使用 Amazon CloudFront 函數 處理簡單的使用案例，例如可由短期函數執行的 HTTP(s) 請求或回應操作。
  - 使用 AWS IoT Greengrass 為連線的裝置執行本機運算、簡訊和資料快取。
- 使用連線共用來啟用連線重複使用，減少所需資源。
- 使用不倚賴持續連線和同步更新的分散式資料存放區來實現一致性，以服務區域的人口。
- 以共用動態容量取代預先佈建的靜態網路容量，與其他訂閱者分攤網路容量的永續性影響。

資源

相關文件：

- 優化您的 AWS 永續性基礎架構，第 III 部分：聯網
- Amazon ElastiCache 文件
- 什麼是 Amazon CloudFront？
- Amazon CloudFront 主要功能
- Lambda@Edge
- CloudFront 函數
- AWS IoT Greengrass

相關影片：

- 在 AWS 上建立永續性

相關範例：

- AWS 聯網研討會

SUS02-BP05 為執行的活動最佳化團隊成員資源

最佳化提供給團隊成員的資源，以盡量減少對永續性的影響，同時支援他們的需求。例如，在使用率高的共用雲端桌面上執行複雜的操作 (例如渲染和編譯)，而不是在使用率低的高功率單一使用者系統上執行。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 根據使用方式佈建工作站和其他裝置。
- 使用虛擬桌面和應用程式串流來限縮升級與裝置要求。
- 將大量使用處理器或記憶體的任務移至雲端。
- 評估程序和系統對裝置生命週期的影響，並選擇在滿足業務需求的同時可將裝置更換需求降至最低的解決方案。
- 為裝置實作遠端管理，以減少必要商務差旅時間。

資源

相關文件：

- 什麼是 Amazon WorkSpaces？
- Amazon AppStream 2.0 文件
- NICE DCV
- AWS Systems Manager Fleet Manager

相關影片：

- 在 AWS 上建立永續性

軟體和架構模式

問題

- SUS 3 如何利用軟體和架構模式，來支持您的永續性發展目標？

SUS 3 如何利用軟體和架構模式，來支持您的永續性發展目標？

實施可執行負載順暢並保持已部署資源一致高使用率的模式，將資源消耗降至最低。由於使用者行為隨時間改變，元件可能會因缺乏使用而閒置。修改模式和架構來整合未充分利用的元件，提高整體使用率。淘汰不再需要的元件。了解工作負載元件的效能，並最佳化消耗最多資源的元件。注意客戶用來存取服務的裝置，並實施可最小化裝置升級需求的模式。

最佳實務：

SUS03-BP01 最佳化非同步與排程任務的軟體和架構

使用高效率的軟體設計和架構，將每個工作單元所需的平均資源降至最低。實作可平均利用元件的機制，減少任務之間的閒置資源，並將負載尖峰的影響降至最低。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 將不需要立即處理的請求放入佇列。
- 增加序列化，讓使用率在不同管道中變得平均。
- 修改個別元件的容量，避免閒置資源等待輸入。
- 建立緩衝區，並建立速率限制，讓外部服務的消耗變得順暢。
- 使用最有效的可用硬體來進行軟體最佳化。
- 使用佇列驅動的架構、管道管理和隨需執行個體工作者，最大化批次處理的使用率。
- 妥善安排任務，避免同時執行的負載尖峰和資源爭用。
- 在一天電力碳強度最低的時段安排工作。

資源

相關文件：

- 什麼是 Amazon Simple Queue Service？
- 什麼是 Amazon MQ？

- 根據 Amazon SQS 擴展
- 什麼是 AWS Step Functions？
- 什麼是 AWS Lambda？
- 搭配 Amazon SQS 使用 AWS Lambda
- 什麼是 Amazon EventBridge？

相關影片：

- 在 AWS 上建立永續性
- 移至事件驅動型架構

SUS03-BP02 移除或重構使用量低或完全不使用工作負載元件

監控工作負載活動，識別各元件使用率隨時間的變化。移除未使用且不再需要的元件，並重構使用率低的元件，減少資源浪費。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 分析功能元件上的負載 (使用交易流程和 API 呼叫等指標)，以識別未使用和未充分利用的元件。
- 淘汰不再需要的元件。
- 重構未充分利用的元件。
- 將未充分利用的元件與其他資源整合，以提高利用效率。

資源

相關文件：

- 什麼是 AWS X-Ray？
- 什麼是 Amazon CloudWatch？
- 使用 ServiceLens 監控應用程式的運作狀態
- 自動清理 Amazon ECR 中未使用的映像

相關影片：

- 在 AWS 上建立永續性

SUS03-BP03 最佳化程式碼中消耗最多時間或資源的區域部分

監控工作負載活動，識別消耗最多資源的應用程式元件。最佳化這些元件中執行的程式碼，將資源使用量降至最低，同時將效能發揮至最大。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 根據資源使用情況監控效能，找出每個工作單元中資源需求高的元件，做為最佳化目標。
- 使用程式碼分析工具來識別程式碼中使用最多時間或資源的部分，作為最佳化目標。
- 將演算法取代為產生相同結果但更有效率的版本。
- 使用硬體加速來改善執行時間較長程式碼區塊的效率。
- 使用針對工作負載最高效率的作業系統和程式設計語言。
- 移除不必要的排序和格式化。
- 使用可根據資料變更頻率和使用方式，將使用的資源降至最低的資料傳輸模式。例如，將狀態變更資訊推送到用戶端，而不是讓它耗用資源來輪詢和接收沒用的「無變更」訊息。

資源

相關文件：

- 什麼是 Amazon CloudWatch？
- 什麼是 Amazon CodeGuru Profiler？
- FPGA 執行個體
- 在 AWS 上建立的工具中的 AWS 開發套件

相關影片：

- 在 AWS 上建立永續性

SUS03-BP04 最佳化對客戶裝置和設備的影響

了解客戶用來使用您服務的裝置和設備、其預期生命週期，以及更換這些元件對財務和永續性的影響。實施軟體模式和架構，將客戶更換裝置和升級設備的需求降至最低。例如，採用與舊版硬體和作業系統版本相容程式碼的新功能，或管理承載的大小，不讓其超過目標裝置的儲存容量。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 清查客戶使用的裝置。
- 使用具有代表性硬體集的受管 Device Farm 進行測試，以了解變更的影響，並迭代開發以最大化支援的裝置。
- 在建置承載時考慮網路頻寬和延遲，並實施可協助應用程式在低頻寬、高延遲連結上良好運作的功能。
- 預先處理資料承載，減少本機處理需求並限縮必要的資料傳輸。
- 在伺服器端執行需要大量運算的活動 (例如影像渲染)，或使用應用程式串流來改善舊裝置的使用者體驗。
- 對輸出進行分段和分頁，特別是對於互動式工作階段，以管理承載並限制本機儲存要求。

資源

相關文件：

- 什麼是 AWS Device Farm？
- Amazon AppStream 2.0 文件
- NICE DCV
- Amazon Elastic Transcoder 文件

相關影片：

- 在 AWS 上建立永續性

SUS03-BP05 使用最能支援資料存取和儲存模式的軟體模式和架構

了解資料在工作負載中的使用方式、使用者的使用方式、傳輸方式以及儲存方式。選擇可將資料處理和儲存要求降至最低的技術。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 分析您的資料存取和儲存模式。
- 以有效率的檔案格式 (例如 Parquet) 存放資料檔案，以避免進行不必要的處理 (例如執行分析時) 並降低佈建的總儲存量。
- 利用可原生處理壓縮資料的技術。
- 使用最能支援您主導查詢模式的資料庫引擎。
- 管理您的資料庫索引，確保索引設計支援高效率的查詢執行。
- 選取可減少網路容量消耗的網路通訊協定。

資源

相關文件：

- Athena 壓縮支援檔案格式
- 使用 Amazon Redshift 從單欄資料格式複製
- 在 Firehose 中轉換您的輸入記錄格式
- AWS Glue 中 ETL 輸入和輸出的格式選項
- 轉換為單欄格式，提高 Amazon Athena 的查詢效能
- 使用 Amazon Redshift 從 Amazon S3 載入壓縮的資料檔案
- 在 Amazon Aurora 上使用績效詳情監控資料庫負載
- 在 Amazon RDS 上使用績效詳情監控資料庫負載
- AWS IoT FleetWise

相關影片：

- 在 AWS 上建立永續性

資料模式

問題

- SUS 4 如何利用資料存取和使用模式，來支持您的永續性發展目標？

SUS 4 如何利用資料存取和使用模式，來支持您的永續性發展目標？

實作資料管理實務來減少支援工作負載所需的佈建儲存，以及使用它的必要資源。了解您的資料，並使用最能支援資料業務價值及其使用方式的儲存技術和組態。當需求減少時，將資料循環到效率較高、效能較低的儲存，並刪除不再需要的資料。

最佳實務：

SUS04-BP01 實作資料分類政策

將資料分類，以了解其對業務成果的重要性。使用此資訊來判斷何時可將資料移動到更節能的儲存，或是可以安全刪除它。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 判斷資料的分佈、保留和刪除需求。
- 對磁碟區和物件使用標記，來記錄用於判斷其管理方式的中繼資料，包括資料分類。
- 定期稽核您的環境是否有未標記和未分類的資料，並對資料進行適當的分類和標記。

資源

相關文件：

- 資料分類程序
- 使用 AWS 雲端 以支援資料分類
- 標記來自 AWS Organizations 的政策

SUS04-BP02 使用支援資料存取和儲存模式的技術

使用最能支援您的資料存取和儲存方式的儲存類型，以在支援工作負載的同時，也將佈建的資源降至最低。例如，固態裝置 (SSD) 比磁性磁碟機更耗能，只應用於活躍資料使用案例。針對不常存取的資料，使用節能的存檔類別儲存。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 監控您的資料存取模式。

- 根據存取模式將資料遷移至適當的技術。
- 將存檔資料遷移到專為該用途設計的儲存。

資源

相關文件：

- Amazon EBS 磁碟區類型
- Amazon EC2 執行個體存放區
- Amazon S3 智慧型分層
- 使用 Amazon S3 儲存類別
- 什麼是 Amazon CloudWatch？
- 什麼是 Amazon S3 Glacier？

相關影片：

- AWS 上資料湖的架構模式

SUS04-BP03 使用生命週期政策來刪除不需要的資料

管理所有資料的生命週期並自動執行刪除時間表，將工作負載的總儲存需求降至最低。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 為所有資料分類類型定義生命週期政策。
- 設定自動生命週期政策以強制執行生命週期規則。
- 刪除不使用的磁碟區和快照。
- 根據生命週期規則，在適用的情況下彙總資料。

資源

相關文件：

- Amazon ECR 生命週期政策

- Amazon EFS 生命週期管理
- Amazon S3 智慧型分層
- 使用 AWS Config 規則 評估資源
- 在 Amazon S3 上管理儲存生命週期
- AWS Elemental MediaStore 中的物件生命週期政策

相關影片：

- Amazon S3 生命週期

SUS04-BP04 將區塊儲存中的過度佈建降至最低

若要最小化總佈建儲存，請建立具有適合工作負載之大小分配的區塊儲存。使用彈性磁碟區，隨著資料成長擴展儲存，無需調整連接到運算資源的儲存大小。定期檢閱彈性磁碟區，並縮減過度佈建的磁碟區以符合目前的資料大小。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 監控資料磁碟區的使用率。
- 使用彈性磁碟區和受管區塊資料服務，以在持久性資料增長時自動分配額外的儲存空間。
- 設定資料磁碟區的目標使用率水準，並調整超出預期範圍的磁碟區大小。
- 根據資料調整唯讀磁碟區的大小。
- 將資料遷移到物件存放區，避免從區塊儲存的固定磁碟區大小佈建多餘容量。

資源

相關文件：

- Amazon EBS 彈性磁碟區
- Amazon FSx 文件
- 什麼是 Amazon CloudWatch？
- 什麼是 Amazon Elastic File System？

SUS04-BP05 移除不需要或多餘的資料

必要時才複製資料，將消耗的總儲存空間降至最低。使用在檔案和區塊層級刪除重複資料的備份技術。限制獨立磁碟冗餘陣列 (RAID) 組態的使用，除非需要滿足服務水準協議 (SLA)。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 使用可在區塊和物件層級刪除重複資料的機制。
- 使用這樣的備份技術：可在區塊、檔案和物件層級進行增量備份和刪除重複資料。
- 只有在需要滿足 SLA 時，才使用 RAID。
- 集中日誌和追蹤資料、刪除重複的日誌項目，並建立根據需要微調詳細程度的機制。
- 僅在合理的情況下才預先填入快取。
- 建立快取監控和自動化，據以調整快取大小。
- 推送工作負載新版本時，從物件存放區和邊緣快取移除過時的部署和資產。

資源

相關文件：

- Amazon EBS 快照
- 變更 CloudWatch Logs 中的日誌資料保留
- Amazon FSx for Windows File Server 上的重複資料刪除
- Amazon FSx for ONTAP 的功能，包括重複資料刪除
- 使 Amazon CloudFront 上的檔案無效
- 使用 AWS Backup 來備份和還原 Amazon EFS 檔案系統
- 什麼是 Amazon CloudWatch Logs？
- 在 Amazon RDS 上使用備份

相關範例：

- 實驗室：使用 Amazon Redshift 資料共用來優化資料模式

SUS04-BP06 使用共用檔案系統或物件儲存體存取通用資料

採用共用儲存和單一真實來源，避免資料重複並降低工作負載的總儲存需求。需要時才從共用儲存體擷取資料。分離未使用的磁碟區以使更多資源可用。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 當資料有多個取用者時，將資料遷移到共用儲存體。
- 需要時才從共用儲存體擷取資料。
- 根據您的使用模式刪除資料，並實施存留時間 (TTL) 功能來管理快取資料。
- 將磁碟區與未積極使用它們的用戶端分開。

資源

相關文件：

- Amazon FSx
- 快取策略
- 什麼是 Amazon Elastic File System？
- 什麼是 Amazon S3？

SUS04-BP07 將跨網路的資料移動降到最少

使用共用儲存，並從區域資料存放區存取資料，將支援工作負載資料移動所需的總網路資源降至最低。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 盡可能將資料存放至接近消費者的位置。
- 對區域性使用的服務進行分區，以便將區域專屬的資料存放在使用它的區域內。
- 透過網路複製變更時，在區塊層級進行複製，而不要在檔案或物件層級進行。
- 在透過網路移動資料之前先壓縮資料。

資源

相關文件：

- 優化您的 AWS 永續性架構，第 III 部分：聯網
- AWS 全球基礎設施
- Amazon CloudFront 主要功能，包括 CloudFront Global Edge Network
- 壓縮 Amazon OpenSearch Service 中的 HTTP 請求
- 使用 Amazon EMR 進行中間資料壓縮
- 將壓縮的資料檔案從 Amazon S3 載入至 Amazon Redshift
- 使用 Amazon CloudFront 提供壓縮檔案服務

SUS04-BP08 僅在難以重新建立時才備份資料

為了將儲存消耗降至最低，僅備份具有業務價值或需要滿足合規要求的資料。檢查備份政策，並在復原案例中排除沒有價值的暫時性儲存。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 使用資料分類來確定需要備份的資料。
- 排除可以輕鬆重建的資料。
- 從備份排除暫時性資料。
- 排除資料的本機副本，除非從共同位置還原資料所需的時間不符合服務水準協議 (SLA) 的要求。

資源

相關文件：

- 使用 AWS Backup 來備份和還原 Amazon EFS 檔案系統
- Amazon EBS 快照
- 在 Amazon Relational Database Service 上使用備份

硬體模式

問題

- SUS 5 您的硬體管理和使用實務如何支持您的永續性發展目標？

SUS 5 您的硬體管理和使用實務如何支持您的永續性發展目標？

透過變更硬體管理實務，尋求降低工作負載永續性影響的機會。將佈建和部署所需的硬體量降至最低，並為個別工作負載選擇最高效率的硬體。

最佳實務：

SUS05-BP01 使用最低數量的硬體來滿足需求

使用雲端功能，您可以頻繁變更工作負載實作。隨著需求變更，更新已部署的元件。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 啟用水平擴展，並使用自動化在負載增加時擴展、在負載減少時縮減。
- 針對變動工作負載，以較小的增量進行擴展。
- 當負載隨著天、週、月或年而變化，讓擴展程度符合週期性使用模式 (例如，薪資系統每雙週會有密集的處理活動)。
- 協商服務水準協議 (SLA)，允許暫時減少容量，同時自動化部署更換資源。

資源

相關文件：

- AWS Compute Optimizer 文件
- 操作 Lambda：效能優化
- Auto Scaling 文件

SUS05-BP02 使用影響最小的執行個體類型

持續關注新執行個體類型的發佈，並運用能源效率改進，包括旨在支援特定工作負載 (例如機器學習訓練、推論和影片轉碼) 的執行個體類型。

常見的反模式：

- 您僅使用一個執行個體系列。

- 您僅使用 x86 執行個體。
- 您在 Amazon EC2 Auto Scaling 組態中指定了一個執行個體類型。
- 您以不符合設計宗旨的方式使用 AWS 執行個體 (例如，您將運算優化的執行個體用於記憶體密集型工作負載)。
- 您未定期評估新的執行個體類型。
- 您未查看 AWS 適當調整大小的工具 (例如 AWS Compute Optimizer) 提供的建議。

建立此最佳實務的優勢： 藉由使用節能且適當調整大小的執行個體，將可大幅降低環境受到的影響以及工作負載成本。

未建立此最佳實務時的曝險等級： 低

實作指引

- 了解並探索可降低工作負載環境影響的執行個體類型。
  - 訂閱 AWS 最新消息 ，隨時掌握最新的 AWS 技術和執行個體。
  - 了解不同的 AWS 執行個體類型。
  - 觀看下列資源，了解 AWS Graviton 型執行個體如何在 Amazon EC2 中的能源使用提供最佳效能功耗比。 re:Invent 2020 - 深入探討搭載 AWS Graviton2 處理器的 Amazon EC2 執行個體 和 深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體。
- 進行相關規劃，將工作負載轉移至影響程度最低的執行個體類型。
  - 定義一個程序來評估工作負載的新功能和執行個體。利用雲端的靈活性快速測試新功能類型對您的工作負載環境永續性有何改善。使用代理指標，測量您需要多少資源才能完成一個工作單位。
  - 如果可行，請修改工作負載，以使用不同數量的 CPU 和不同數量的記憶體，從而最大化您選擇執行個體類型的空間。
  - 考慮將您的工作負載轉移至 Graviton 型執行個體，以改善工作負載的效能效率 (請參閱 AWS Graviton Fast Start 和 AWS Graviton2 for ISVs) 建立持續整合/持續部署 (CI/CD) 管道。請留意 將工作負載轉移至 AWS Graviton 型 Amazon Elastic Compute Cloud 執行個體時所需考量的事項。
  - 請考慮選取 AWS Graviton 選項 (在您要使用的 AWS 受管服務中)。
  - 將工作負載遷移至有執行個體對永續性影響最小，且仍符合業務要求的區域。
  - 對於機器學習工作負載，請使用採用自訂 Amazon Machine Learning 晶片的 Amazon EC2 執行個體，例如 AWS Trainium， AWS Inferentia和 Amazon EC2 DL1。
  - 使用 Amazon SageMaker Inference Recommender 適當調整 ML 推論端點的大小。
  - 對於具有即時視訊轉碼的工作負載，請使用 Amazon EC2 VT1 執行個體。

  - 對於激增的工作負載 (不常需要額外容量的工作負載)，請使用 高載效能執行個體。
  - 對於無狀態和容錯工作負載，請使用 Amazon EC2 Spot 執行個體 提高雲端整體使用率，並減少未使用資源的永續性影響。
- 操作並優化您的工作負載執行個體。
  - 對於暫時性工作負載，請評估 執行個體 Amazon CloudWatch 指標 (例如 `CPUUtilization`)，以確認執行個體是否閒置或未充分利用。
  - 對於穩定的工作負載，請定期檢查 AWS 適當調整大小的工具 (例如 AWS Compute Optimizer)，以找出對執行個體進行優化和適當調整大小的機會。

資源

相關文件：

- 優化您的 AWS 永續性基礎架構，第 I 部分：運算
- AWS Graviton 處理器
- AWS Inferentia
- AWS Trainium
- Amazon EC2 DL1
- Amazon EC2 高載效能執行個體
- Amazon EC2 容量保留機群
- Amazon EC2 Spot 機群
- Amazon EC2 Spot 執行個體
- Amazon EC2 VT1 執行個體
- Amazon EC2 執行個體類型
- AWS Compute Optimizer
- 函數：Lambda 函數組態

相關影片：

- 深入探討搭載 AWS Graviton2 處理器的 Amazon EC2 執行個體
- 深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體

相關範例：

- 實驗室：適當調整大小的建議
- 實驗室：使用 Compute Optimizer 適當調整大小
- 實驗室：優化硬體模式和觀察續性 KPI

SUS05-BP03 使用受管服務

受管服務可將維持高平均使用率和已部署硬體的永續性最佳化責任轉移給 AWS。使用受管服務，將服務的永續性影響分散給服務的所有租用戶，降低您的個人佔比。

若未建立此最佳實務，暴露的風險等級為： 低

實作指引

- 將自我託管服務遷移到受管服務。例如，使用受管 Amazon Relational Database Service (Amazon RDS) ，而非在 Amazon Elastic Compute Cloud (Amazon EC2)上維護您自己的 Amazon RDS 執行個體，或使用受管容器服務，例如 AWS Fargate，而非實作您自己的容器基礎設施。

資源

相關文件：

- AWS Fargate
- Amazon DocumentDB
- Amazon Elastic Kubernetes Service (EKS)
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)
- Amazon Redshift
- Amazon Relational Database Service (RDS)

SUS05-BP04 優化 GPU 的使用

圖形處理器 (GPU) 可能是高功耗的來源，許多 GPU 工作負載會高度變動，例如渲染、轉碼以及機器學習訓練和建模。只在需要時執行 GPU 執行個體，並在不需要時自動除役，以將資源消耗降至最低。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 只把 GPU 用於比 CPU 型替代方案更具效率的任務。

- 使用自動化來釋出不使用的 GPU 執行個體。
- 使用靈活的圖形加速，而不是專用的 GPU 執行個體。
- 運用專屬於您工作負載的自訂用途硬體。

資源

相關文件：

- 加速運算
- AWS Inferentia
- AWS Trainium
- EC2 執行個體的加速運算
- Amazon EC2 VT1 執行個體
- Amazon Elastic Graphics

開發與部署程序

問題

- SUS 6 您的開發和部署程序如何支持您的永續性發展目標？

SUS 6 您的開發和部署程序如何支持您的永續性發展目標？

透過變更開發、測試和部署實務來尋找降低永續性影響的機會。

最佳實務：

SUS06-BP01 採用可以快速引入永續性改進的方法

在將潛在改善部署到生產環境之前，先對其進行測試和驗證。在計算改善所帶來的未來潛在利益時，應考慮測試成本。開發低成本測試方法，以交付小幅改善。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

- 在開發程序中新增永續性要求。
- 允許資源平行運作，以開發、測試和部署永續性改進。

- 在部署到生產環境之前，測試並驗證潛在永續性影響改善。
- 使用最低可行的代表元件測試潛在改善。
- 在將經過測試的永續性改善可用時將其部署至生產環境。

資源

相關文件：

- AWS 提供永續性解決方案

相關範例：

- 實驗室：將 成本和用量報告轉換為效率報告

SUS06-BP02 讓工作負載保持在最新狀態

最新的作業系統、程式庫和應用程式可改進工作負載效率，讓您更輕鬆地採用更有效率的技術。隨著供應商提供符合自身永續性目標的功能，最新軟體也可能包含更準確測量工作負載對永續性影響的功能。

常見的反模式：

- 您假設您目前的架構將變成靜態，且不會隨著時間而更新。
- 您沒有任何系統或定期規律可評估更新的軟體與套件是否與您的工作負載相容。
- 您會隨時間導入架構變更，而且無需理由佐證。

建立此最佳實務的優勢： 藉由建立讓工作負載保持在最新狀態的程序，您將可採用新的特性和功能、解決問題，並且改善工作負載效率。

未建立此最佳實務時的曝險等級： 低

實作指引

- 定義相關程序和排程來評估工作負載的新功能和執行個體。利用雲端的靈活性快速測試新功能對您的工作負載有何改善，藉以：
  - 降低永續性的影響。
  - 獲得效能效率。
  - 消除已計劃改善的障礙。

    - 提升測量和管理永續性影響的能力。
- 清查工作負載軟體和架構，並識別需要更新的元件。您可以使用 AWS Systems Manager 清查 ，從您的 Amazon EC2 執行個體收集作業系統 (OS)、應用程式和執行個體中繼資料，並快速了解哪些執行個體正在執行您的軟體政策所需的軟體與組態，以及哪些執行個體需要更新。
- 了解如何更新工作負載的元件。
    - 使用下列工具管理 Amazon Machine Images (AMI) for Linux/Windows 伺服器映像的更新： EC2 Image Builder。
    - 您應使用 Amazon Elastic Container Registry (Amazon ECR) 搭配現有管道來 管理 Amazon Elastic Container Service (Amazon ECS) 映像 和 管理 Amazon Elastic Kubernetes Service 映像。
    - AWS Lambda 包含 版本管理功能。
- 使用更新程序自動化，以減少部署新功能的工作量，並避免手動程序引起的錯誤。使用 AWS Systems Manager Patch Manager 之類的工具自動執行系統更新的程序，並使用 AWS Systems Manager 維護時段來排程活動。

資源

相關文件：

- AWS 架構中心
- AWS 最新消息
- AWS 開發人員工具
- AWS Systems Manager Patch Manager

相關範例：

- Well-Architected 實驗室：清查和修補程式管理
- 實驗室：AWS Systems Manager

SUS06-BP03 提高建置環境的使用率

使用自動化和基礎設施即程式碼，在需要時啟動生產前環境，並在不使用時將其關閉。常見的模式是排程可用性時間，使之與開發團隊成員的工作時間一致。休眠是一種有用的工具，可保留狀態，並在需要時快速讓執行個體上線。使用具有高載容量的執行個體類型、Spot 執行個體、彈性資料庫服務、容器和其他技術，以根據使用量調整開發和測試容量。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

- 利用自動化來最大化開發和測試環境的使用率。
- 使用自動化來管理開發和測試環境的生命週期。
- 使用最低可行的代表環境來開發和測試潛在改善。
- 使用隨需執行個體補充開發人員裝置。
- 使用自動化來最大化建置資源的效率。
- 使用具有高載容量的執行個體類型、Spot 執行個體和其他技術，以根據使用量調整建置容量。
- 採用原生雲端服務來獲得安全的執行個體 Shell 存取，而非部署堡壘主機機群。

資源

相關文件：

- AWS Systems Manager Session Manager
- Amazon EC2 高載效能執行個體
- 什麼是 AWS CloudFormation？

SUS06-BP04 使用受管 Device Farm 進行測試

受管 Device Farm 可將硬體製造和資源使用的永續性影響分散給多個租用戶。受管 Device Farm 提供多種裝置類型，因此您可以支援較舊且較不熱門的硬體，並避免不必要的裝置升級對客戶的永續性造成影響。

若未建立此最佳實務，暴露的風險等級： 低

實作指引

使用具有代表性硬體集的受管 Device Farm 進行測試，以了解變更的影響，並迭代開發以最大化支援的裝置。

資源

相關文件：

- 什麼是 AWS Device Farm？

聲明

客戶應負責對本文件中的資訊自行進行獨立評估。本文件：(a) 僅供參考之用，(b) 代表目前的 AWS 產品供應與實務，如有變更恕不另行通知，以及 (c) 不構成 AWS 及其附屬公司、供應商或授權人的任何承諾或保證。AWS 產品或服務以「現況」提供，不提供任何明示或暗示的擔保、主張或條件。AWS 對其客戶之責任與義務，應受 AWS 協議之約束，且本文件並不屬於 AWS 與其客戶間之任何協議的一部分，亦非上述協議之修改。