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： 2024 年 6 月 27 日 (文件修訂)

AWS Well-Architected Framework 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、經濟實惠且永續的系統的架構最佳實務。

簡介

AWS Well-Architected Framework 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，有助於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) 是一種雲端服務，可以提供一致的程序，讓您審查和量測使用 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，而減少工作負載所有組件的能源消耗、提高效率，最終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 是應一項要求所一同遞送的程式碼、設定和 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，客戶經常具備負責技術架構的集中團隊 (而技術架構將疊覆在其他產品或功能團隊上) 以確認其過程遵照最佳實務。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，例如為了確認團隊符合內部標準之際。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)，目的是讓各個團隊具備該項能力，並且聘用專家，確認該團隊提高所需符合標準的標竿。第二，我們實作 機制 來實施自動化檢查，確認其符合標準。

> 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。反向作用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，或讓現有團隊提高標竿，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，並協助團隊了解有哪些 AWS 最佳實務。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。首席工程師會在看見新的最佳實務出現時，採取社群方式進行工作，以確認團隊會遵守這些實務。假以時日，這些最佳實務會正式列入我們內部的審查程序，同時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，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。

- 考量架構試驗的自動化：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。
- 考量演進的架構：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。如此可讓系統隨時間演進，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。您可以在卓越營運支柱白皮書中找到實作的指引。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

下列設計原則有助於實現雲端中的卓越營運：

- 以業務成果為中心來組織團隊：團隊可以從領導願景、有效的營運和業務一致的營運模式，獲取業務成果。領導階層應充分投入並致力於 CloudOps 轉型，採用合適的雲端營運模式，激勵團隊以最有效

率的方式運作，並獲得業務成果。正確的營運模式會利用人員、流程和技術能力，並且擴展、最佳化生產力，並於敏捷性、回應能力和適應性脫穎而出。組織的長期願景會轉化為橫跨整個企業的目標，並傳達給利益相關者和雲端服務消費者。目標和營運 KPI 於所有層級皆一致。這種做法可以維持實作下列設計原則所獲得的長期價值。

- 實作可觀測性以獲得可採取行動的見解：全面了解工作負載的行為、效能、可靠性、成本和運作狀態。建立關鍵績效指標 (KPI) 並利用可觀測性遙測，以做出明智決策並在業務成果面臨風險時立即採取行動。根據可採取行動的可觀測性資料，主動改善效能、可靠性和成本。
- 視情況盡可能自動化：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的相同工程原則。您可以將整個工作負載及其營運 (應用程式、基礎架構、組態和程序) 定義為程式碼，並將其更新。接著，便可自動化工作負載營運，在回應事件時啟動這些作業。在雲端中，自動化安全可以透過設定防護機制完成部署，包括錯誤率控制、錯誤臨界值和核准。透過有效的自動化，您就可以達到一致的事件回應、限制人為錯誤，並減少操作人員疲累情況。
- 進行頻繁、細微和可逆的變更：設計可擴展且鬆散耦合的工作負載以允許定期更新元件。自動化部署技術加上較細微的增量變更可縮減影響範圍，並在發生故障時更快地反轉情況。這可讓您更有信心您能為工作負載帶來有益的變更，同時維持品質並迅速適應市場情況的變化。
- 經常完善營運程序：隨著您工作負載的演進，您的營運也應該適當演進。在使用營運程序時，尋找機會予以改善。保持定期檢閱，並驗證所有程序是否有效以及團隊是否熟悉這些程序。如果發現漏洞，請相應地更新程序。向所有利害關係人和團隊傳達程序更新消息。將營運遊戲化以分享最佳實務並教導團隊。
- 預期失敗：盡量提升營運的成果，透過失敗案例了解工作負載的風險概況，及其對於業務成果的影響。測試程序的有效性，以及團隊面對這些模擬失敗的反應。做出明智的決策，並管理經由測試所識別的開放風險。
- 從所有營運事件和指標中學習：從所有營運事件和失敗中學習經驗，進而不斷改善。跨團隊及在整個組織中分享獲得的經驗。學習內容應強調有關營運如何有助於業務成果的資料與軼事。
- 使用受管服務：盡可能地使用 AWS 受管服務以降低營運負擔。圍繞與這些服務的互動建置營運程序。

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服務以實現整合、部署及交付工作負載，將使得更多有益的變更發揮作用。

工作負載的操作本質上就可能存在著風險。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，將讓您了解工作負載的運作狀態、營運活動，並回應事故。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

> **Note**
>
> 所有卓越營運相關問題都會加上 OPS 前置詞做為要點縮寫。

主題

- 組織
- 準備
- 營運
- 演進

組織

您的團隊必須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，以便設定能達到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。評估客戶需求將確定您對實現業務成果所需的支援有透徹的了解。確認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，請確認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。團隊必須了解本身在促成其他團隊成功的過程中所扮演的角色、其他團隊在獲致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。

確認每個應用程式、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。資深領導階層應是採用最佳實務和組織演進的發起者、倡導者和推動者。讓團隊成員能夠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。確認團隊成員擁有可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，則您應使用 AWS 雲端合規提供的資源來協助教育您的團隊，以便他們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。它提供了一致的方法，讓您用於評估架構並實作將隨時間擴展的設計。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，這些檢查提出了最佳化建議，可協助您確定優先事項。商業和企業支援客戶可存取針對安全性、可靠性、效能、成本最佳化和永續性的其他檢查，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，從而讓他們更加了解自己的選擇會如何影響工作負載。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 建置者資料中心營運 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 培訓與認證透過 AWS 基礎原理自主進度數位課程提供一些培訓。您還可以報名參加講師指導下的培訓，以進一步協助發展團隊的 AWS 技能。

使用可讓您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，讓您能夠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，都會提供有關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？

每個人都必須了解其在達成業務成果中所扮演的角色。擁有共同目標，並設定資源優先順序。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。團隊必須了解本身在促成其他團隊成功的過程中所扮演的角色、其他團隊在獲致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，以確認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，以推動業務成功。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。可觀測性不僅是單純的監控，還可根據系統的外部輸出全面了解系統的內部運作狀況。以指標、日誌和追蹤為根基，可觀測性提供了系統行換動態的洞見。有效的可觀測性能夠讓團隊辨別模式、異常情況和趨勢，以便主動解決潛在問題並維持最佳的系統運作狀態。確定關鍵績效指標 (KPI) 至關重要，可確保監控活動與業務目標保持一致。這種一致性可確保團隊使用真正重要的指標來做出資料驅動的決策，進而最佳化系統效能和業務成果。此外，可觀測性使得企業能夠化被動為主動。團隊能夠了解系統內的因果關係，預測並預防問題，而不只是被動回應問題。隨著工作負載的演進，務必重新檢視並改進可觀測性策略，以保持相關性和有效性。

採用的方法需能夠改善變更發揮作用的流程，並實現重構、快速提供品質意見回饋，以及修復錯誤。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，並且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，可以更快進行疑難排解並加快修復速度。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。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。這樣也有助於尋找您必須制定計畫以解決問題的任何領域。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 可讓您以程式碼的形式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 可讓您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：如何在工作負載中實作可觀測性？

在工作負載中實作可觀測性，以便了解其狀態，並根據業務需求做出資料驅動的決策。

OPS 5：您如何減少缺陷、有效輕鬆修復，以及改善生產流程？

採用改善改變生產流程的方法，藉此推動重構、快速提供品質意見回饋及修復錯誤。這些方法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。

對以程式碼形式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

可觀測性讓您能夠專注於有意義的資料，並了解工作負載的互動和結果。透過專注於基本洞見並消除不必要的資料，您就能持續使用簡單直接的方式來了解工作負載效能。重點不只是收集資料，還要正確解譯資料。定義清楚的基準、設定適當的警示閾值，並主動監控任何偏差情況。一旦關鍵指標稍有變化，尤其是與其他資料相關時，就能精確指出特定問題所在。有了可觀測性，您就具備更優異的預測能力，並且能應付潛在的挑戰，進而確保工作負載順利運行並滿足業務需求。

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。確認若因回應事件而發出警示，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，並納入向上呈報程序，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，您可以產生儀表板視圖，用以顯示從工作負載或以原生方式從 AWS 收集的指標。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，讓您從中識別工作負載問題，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：如何在組織中利用工作負載可觀測性？

利用可觀測性確保最佳的工作負載運作狀況。利用相關指標、日誌和追蹤，全面掌握工作負載效能並有效解決問題。

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

學習、分享和持續改善以維持卓越營運。投入工作週期以近乎持續的方式逐漸改善。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執行營運中獲得經驗。

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 中，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 與 AWS Glue 進行原生整合後，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，盡量持續逐漸改善，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。

SEC 2：如何管理人員和機器的身分？

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
- 定義
- 最佳實務
- 資源

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
- 工作負載架構

- 變更管理
- 失敗管理

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。您可以在效能達成效率支柱白皮書中找到實作的指引。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端有五個設計原則來維持效能達成效率：

- 讓進階技術變得更普及：將複雜的任務委派給雲端廠商，讓團隊更順暢地實作進階技術。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，讓團隊能夠專注於產品開發，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，為客戶提供更低延遲、更優質的體驗。
- 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。
- 考慮機械同感作用：了解雲端服務的使用方式，並一律使用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

維持雲端效能達成效率的最佳實務有五個領域：

- 選擇架構
- 運算與硬體
- 資料管理
- 網路與內容交付
- 程序和文化

採取資料驅動的方法來建置高效能架構。從高階設計到選取和設定資源類型，收集架構各方面的資料。

定期審查您的選擇，以確實充分利用不斷演進的 AWS 雲端。監控可確保您能察覺預期效能發生的任何偏差情形。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

選擇架構

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 工作負載會使用多種解決方案，並採用不同的功能以提升效能。

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 Appendix。)

PERF 1: How do you select appropriate cloud resources and architecture patterns for your workload?

Often, multiple approaches are required for more effective performance across a workload. Well-Architected systems use multiple solutions and features to improve performance.

運算與硬體

特定工作負載的最佳運算選擇會根據應用程式設計、使用模式和組態設定而有所不同。架構會針對不同元件使用不同運算選擇，並採用不同功能以提升效能。若選錯運算資源，可能使架構的效能達成效率降低。

在 AWS 中，提供了三種運算形式：執行個體、容器和函數。

- 執行個體是虛擬化伺服器，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)。
- 函數可從您想套用的程式碼中將執行環境抽象化。例如，AWS Lambda 可讓您不需執行執行個體就能執行程式碼。

下列問題著重於效能達成效率方面的這些考量。

PERF 2: How do you select and use compute resources in your workload?

The more efficient compute solution for a workload varies based on application design, usage patterns, and configuration settings. Architectures can use different compute solutions for various

PERF 2: How do you select and use compute resources in your workload?

components and turn on different features to improve performance. Selecting the wrong compute solution for an architecture can lead to lower performance efficiency.

資料管理

特定系統的最佳資料管理解決方案會根據資料類型 (區塊、檔案或物件)、存取模式 (隨機或循序)、所需輸送量、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 工作負載會使用專用資料存放區，這些存放區採用不同的功能以提升效能。

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，以便使用者從任何網際網路位置存取資料，用於使用者產生的內容、作用中封存、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。如 Amazon Elastic File System (Amazon 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 可讓您以高效率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: How do you store, manage, and access data in your workload?

The more efficient storage solution for a system varies based on the kind of access operation (block, file, or object), patterns of access (random or sequential), required throughput, frequency of access (online, offline, archival), frequency of update (WORM, dynamic), and availability and durability constraints. Well-architected systems use multiple storage solutions and turn on different features to improve performance and use resources efficiently.

網路與內容交付

工作負載的最佳聯網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。這些限制可能隨著邊緣節點或資源位置而有所差異。

在 AWS 上，聯網以虛擬化方式存在，並提供多種不同的類型和組態。如此就能更容易滿足您的聯網需求。AWS 提供了多種產品功能 (例如，增強型聯網、經 Amazon EC2 聯網最佳化的執行個體、Amazon S3 Transfer Acceleration 和動態 Amazon CloudFront)，可最佳化網路流量。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 4: How do you select and configure networking resources in your workload?

This question includes guidance and best practices to design, configure, and operate efficient networking and content delivery solutions in the cloud.

程序和文化

在架構工作負載時，您可以採取一些原則和實務，來更有效率地執行高效能雲端工作負載。為了培養高效能雲端工作負載的文化，請考慮下列重要原則和實務。

打造這類文化時，請考慮以下重要原則：

- 基礎設施即程式碼：使用 AWS CloudFormation 範本等方法將您的基礎設施定義為程式碼。使用範本可讓您將基礎設施與應用程式程式碼和組態一起置於原始檔控制中。這可讓您在基礎設施中套用開發軟體時所使用的相同做法，進而快速進行迭代。
- 部署管道：使用持續整合/持續部署 (CI/CD) 管道 (例如，原始程式碼儲存庫、建置系統、部署和測試自動化) 來部署您的基礎設施。這樣您就可以在反覆執行的過程中，採用可重複、一致且低成本的方式進行部署。
- 定義明確的指標：設定並監控指標以擷取關鍵績效指標 (KPI)。我們建議您同時使用技術和業務指標。對於網站或行動應用程式，關鍵指標是擷取第一個位元組或轉譯的時間。其他一般適用的指標包括執行緒計數、垃圾回收率和等待狀態。業務指標 (例如每個請求的彙總累計成本) 會提示您降低成本的方法。仔細考慮您計劃如何解釋指標。例如，您可以選擇最大值或第 99 個百分位數，而非平均值。
- 自動執行效能測試：在部署程序中，成功通過快速執行測試之後，就會自動開始進行效能測試。自動化應建立一個新的環境，設定如測試資料之類的初始條件，然後執行一系列基準測試和負載測試。這

些測試的結果應與組建版本綁定，方便您追蹤長時間的效能變化。對於長期執行的測試，您可以讓管道的這個部分與組建版本的其餘部分不同步。或者，您可以使用 Amazon EC2 Spot 執行個體在夜間執行效能測試。

- 負載產生：您應建立一系列的測試指令碼來複寫綜合性或預錄的使用者旅程。這些指令碼應該以冪等及非耦合的形式呈現，而且您可能需要納入預熱型指令碼才能產生有效的結果。您的測試指令碼應盡可能地複寫生產環境中的使用行為。您可以使用軟體或軟體即服務 (SaaS) 解決方案來產生負載。您可以考慮使用 AWS Marketplace 解決方案和 Spot 執行個體 — 它們會是負載產生的經濟實惠方式。
- 效能可見度：關鍵指標應對您的團隊可見，尤其是針對每個組建版本的指標。這可讓您查看隨時間變化出現的任何顯著的正面或負面趨勢。您也應顯示錯誤或例外狀況數量的指標，以確保您測試的是可運作的系統。
- 視覺化：使用視覺化技術可以清楚指出何處出現效能問題、熱點、等待狀態或較低的利用率。在架構圖上重疊效能指標 — 呼叫圖表或程式碼有助於快速識別問題。
- 定期審查程序：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不斷反覆進行改善。
- 持續最佳化：培養文化以持續最佳化雲端工作負載效能達成效率。

下列問題著重於效能達成效率方面的這些考量。

PERF 5: What process do you use to support more performance efficiency for your workload?

When architecting workloads, there are principles and practices that you can adopt to help you better run efficient high-performing cloud workloads. To adopt a culture that fosters performance efficiency of cloud workloads, consider these key principles and practices.

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 效能最佳化
- 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 2019：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

- 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

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
白皮書已更新	最佳實務更新了新的實作指引。	June 27, 2024
主要更新	主要 效能支柱 經過重新建構，將最佳實務分成五個領域。對安全支柱中事故回應 (SEC 10) 的最佳實務和指引 進行大幅度更新。以下卓越營運領域的主要內容變更和整合： OPS 04、05、06、08 和 09。對於 成本最佳化 和 可靠性支柱 進行整體指引更新。小幅度更新 永續性支柱 風險等級。	October 3, 2023
新框架的更新	最佳實務已更新，納入了規範性指引，並增加了新的最佳實務。安全性和成本最佳化支柱加入了新問題。	April 10, 2023
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了連結。	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

本附錄總結了所有與 AWS Well-Architected Framework 相關的問題與最佳實務。

支柱

- 卓越營運
- 安全性
- 可靠性
- 效能達成效率
- 成本最佳化
- 永續性

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。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。

最佳實務領域

- 組織
- 準備
- 營運
- 演進

組織

問題

- OPS 1. 如何決定您的優先事項？
- OPS 2.如何建構組織以支援業務成果？
- OPS 3.您的組織文化如何支援您的業務成果？

OPS 1. 如何決定您的優先事項？

每個人都應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-BP01 評估客戶需求
- 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 在管理效益和風險的同時評估權衡

OPS01-BP01 評估客戶需求

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。這可確認您對實現想要的商業成果所需的營運支援有透徹的了解。

期望的結果：

- 您可以從客戶成果進行反向操作。
- 您了解您的營運實務如何支援業務成果和目標。
- 您與所有相關方交流。
- 您有掌握客戶需求的機制。

常見的反模式：

- 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：需求被滿足的客戶更有可能成為忠實客戶。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。

未建立此最佳實務時的風險暴露等級：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，才能造就企業的成功。

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：工作負載的業務職能、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

相關的最佳實務：

- OPS11-BP03 實作回饋迴圈

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。

期望的結果：

- 您使用既定的優先事項，將改善工作集中在能發揮最大的影響力的地方 (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)。
- 隨著需求的變化來更新您的優先順序。

常見的反模式：

- 您已決定在不向產品團隊諮詢的情況下，變更他們的 IP 位址配置，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。

未建立此最佳實務時的風險暴露等級：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，才能造就企業的成功。
- 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

相關的最佳實務：

- OPS11-BP03 實作回饋迴圈

OPS01-BP03 評估管控要求

管控是政策、規則或架構的集合，供公司用來達成其商業目標。管控要求產生自您的組織內部。這些要求可能會影響到您所選擇的技術類型，或是您操作工作負載的方式。將組織管控要求納入您的工作負載中。合規是指展現您已實作管控要求的能力。

預期成果：

- 管控要求會併入工作負載的架構設計和操作中。
- 您可以提供您已遵循管控要求的證明。
- 定期審查並更新管控要求。

常見的反模式：

- 您的組織規定根帳戶需進行多重要素驗證。您未能實行此要求，根帳戶遭到損害。
- 在設計工作負載期間，您選擇了未經 IT 部門核准的執行個體類型。您無法啟動工作負載，而必須執行重新設計。
- 您必須有災難復原計劃。您未建立該計劃，且工作負載遭逢長時間的中斷。
- 您的團隊想要使用新的執行個體，但您的管理要求尚未更新予以允許。

建立此最佳實務的優勢：

- 遵循管控要求，可讓您的工作負載符合較大組織的政策。
- 管控要求會反映組織的產業標準和最佳實務。

未建立此最佳實務時的風險暴露等級：高

實作指引

與利害關係人和管控組織共同識別管控要求。將管控要求納入您的工作負載中。能夠證明您已遵循管控要求。

客戶範例

在 AnyCompany Retail，雲端營運團隊與組織內的利害關係人共同制定管控要求。例如，他們禁止對 Amazon EC2 執行個體進行 SSH 存取。如果團隊需進行系統存取，他們必須使用 AWS Systems Manager Session Manager。雲端營運團隊會在新服務推出時定期更新管控要求。

實作步驟

1. 識別工作負載的利害關係人，包括任何集中團隊。
2. 與利害關係人共同識別管控要求。
3. 產生清單後，請排定改善項目的優先順序，並開始在您的工作負載中加以實作。
   a. 使用 AWS Config 之類的服務建立「管控即程式碼」，並驗證確實遵循了管控要求。
   b. 如果您使用 AWS Organizations，您可以利用服務控制政策來實作管控要求。
4. 提供驗證實作情形的文件。

實作計劃的工作量：中。實作遺漏的管控要求可能會導致工作負載重新作業。

資源

相關的最佳實務：

- OPS01-BP04 評估合規要求 - 合規與管控類似，但來自組織外部。

相關文件：

- AWS 管理與管控雲端環境指南
- 多帳戶環境中的 AWS Organizations 服務控制政策的最佳實務

- AWS 雲端 中的管控：敏捷和安全之間的正確平衡
- 什麼是管控、風險和合規 (GRC)？

相關影片：

- AWS 管理與管控：組態、合規和稽核 - AWS 線上技術會談
- AWS re:Inforce 2019：雲端時代的管控 (DEM12-R1)
- AWS re:Invent 2020：使用 AWS Config 實現合規即程式碼
- AWS re:Invent 2020：AWS GovCloud (US) 上的敏捷管控

相關範例：

- AWS Config 合規套件範例

相關服務：

- AWS Config
- AWS Organizations - 服務控制政策

OPS01-BP04 評估合規要求

法規、產業和內部合規要求是定義組織優先順序的重要因子。您的合規架構可能會禁止使用特定技術或地理位置。若未識別出外部合規架構，請運用盡職調查。產生驗證合規性的稽核或報告。

如果聲明您的產品符合特定的合規標準，您必須有內部程序來確保持續的合規性。合規標準的例子包括 PCI DSS、FedRAMP 和 HIPAA。適用的合規標準取決於各種因素，例如解決方案存放或傳輸的資料類型，以及解決方案支援的地理區域。

預期成果：

- 將法規、產業和內部合規要求併入架構選擇中。
- 您可以驗證合規性並產生稽核報告。

常見的反模式：

- 您的工作負載有部分屬於支付卡產業資料安全標準 (PCI-DSS) 架構下，但您的工作負載儲存信用卡資料時並未予以加密。

- 您的軟體開發人員和架構師不知道您的組織必須遵循的合規架構。
- 年度 Systems and Organizations Control (SOC2) Type II 稽核即將到來，但您無法驗證已設置控制。

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。
- 您可以選擇與合規架構相符的適當位置和技術。
- 針對可稽核性設計工作負載，可讓您證明您確實遵循合規架構。

未建立此最佳實務時的風險暴露等級：高

實作指引

若實作此最佳實務，即表示您會在架構設計程序中併入合規要求。您的團隊成員將得知必要的合規架構。您會驗證合規性符合架構。

客戶範例

AnyCompany Retail 儲存客戶的信用卡資訊。卡片儲存團隊的開發人員了解他們必須遵從 PCI-DSS 架構。他們執行了相關步驟，驗證信用卡資訊以安全方式儲存和存取，並遵從 PCI-DSS 架構。他們每年都會與安全團隊共同驗證合規性。

實作步驟

1. 與安全和管控團隊合作，確認您的工作負載必須遵循哪些產業、法規或內部合規架構。在您的工作負載中併入合規架構。
   a. 使用 AWS Compute Optimizer 和 AWS Security Hub 之類的服務驗證 AWS 資源的持續合規性。
2. 讓團隊成員了解合規要求，使其能據以操作及設計工作負載。合規要求應包含在架構和技術選擇中。
3. 根據合規架構，您可能必須產生稽核或合規報告。請與組織合作，盡可能將此程序自動化。
   a. 使用 AWS Audit Manager 之類的服務驗證合規性並產生稽核報告。
   b. 您可以透過 AWS Artifact 下載 AWS 安全與合規文件。

實作計劃的工作量：中。實作合規架構可能並不容易。產生稽核報告或合規文件，會增添額外的複雜性。

資源

相關的最佳實務：

- SEC01-BP03 識別和驗證控制目標 - 安全控制目標是整體合規性的重要環節。
- SEC01-BP06 將管道中安全控制的測試和驗證自動化 - 在您的管道中驗證安全控制。您也可以產生新變更的合規文件。
- SEC07-BP02 定義資料保護控制 - 許多合規架構都以資料處理和儲存政策為基礎。
- SEC10-BP03 準備鑑識功能 - 鑑識功能有時可用來稽核合規性。

相關文件：

- AWS 合規中心
- AWS 合規資源
- AWS 風險與合規白皮書
- AWS 共同責任模式
- 範圍內的 AWS 服務 (依合規計劃)

相關影片：

- AWS re:Invent 2020：使用 AWS Compute Optimizer 實現合規即程式碼
- AWS re:Invent 2021 - 雲端合規、保證和稽核
- AWS Summit ATL 2022 - 在 AWS 上實作合規、保證和稽核 (COP202)

相關範例：

- AWS 上的 PCI DSS 和 AWS 基礎安全最佳實務

相關服務：

- AWS Artifact
- AWS Audit Manager
- AWS Compute Optimizer
- AWS Security Hub

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。

Well-Architected Framework 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，協助您在部署前檢閱方法、在生產前檢閱工作負載狀態，以及檢閱生產階段中的工作負載狀態。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獲得營運審查，該審查旨在助其識別在雲端營運的方法中的差距。

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 是一款可讓您存取核心檢查集的工具，這些檢查能夠提出優化建議，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。

期望的結果：

- 您定期審查並根據 Well-Architected 和 Trusted Advisor 輸出而行動
- 您已知道服務的最新修補程式狀態
- 您了解已知威脅的風險和影響，並據以採取行動
- 您會視需要實作緩解措施
- 您會傳達動作和前後關聯

常見的反模式：

- 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：您可以識別和了解組織和工作負載所面臨的威脅，協助您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。

未建立此最佳實務時的風險暴露等級：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 安全佈告欄
  - 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

相關的最佳實務：

- SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施

相關文件：

- AWS 雲端 合規
- AWS 安全佈告欄
- AWS Trusted Advisor

相關影片：

- AWSre:Inforce 2023 - 協助改善威脅建模的工具

OPS01-BP06 在管理效益和風險的同時評估權衡

來自多方的競爭性利益，可能會使確定工作的優先順序、建置能力以及提供符合業務策略的成果，變得具有挑戰性。例如，您可能會被要求加快新功能上市速度，而不是優化 IT 基礎設施成本。這可能會致使兩方利害關係人相互衝突。在這種情況下，需將決策帶到更高的上級機關以解決衝突。在決策過程中，需要用資料進行判斷，以免受依附情緒左右。

同樣的挑戰可能發生在戰術層面。例如，在使用關聯式或非關聯式資料庫技術之間的選擇，可能會對應用程式的執行產生重大影響。了解各種選擇的可預測結果至關重要。

AWS 可以協助您教育您的團隊有關 AWS 及其服務的知識，從而讓他們更加了解自己的選擇會如何影響工作負載。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。如有更多問題，請聯絡 AWS Support。

AWS 也在 Amazon 建置者資料中心分享營運最佳實務和模式。透過 AWS 部落格和官方AWS播客獲得其他各種實用資訊。

期望的結果：您擁有明確定義的決策管控架構，以便在雲端交付組織中的每個層級進行重要決策。此架構包括風險登錄表、授權做出決策的定義角色，以及可以做出的每個決策層級的定義模型等功能。此架構會預先定義如何解決衝突、需要呈現哪些資料，以及如何優先設定選項，因此當一旦做出決定，您就可以立即提交。決策架構包括一種標準化的方法，可用於審查及衡量每個決策的利益和風險，以了解相關的權衡。這可能包括外部因素，例如遵守法規合規要求。

常見的反模式：

- 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，您在不證明合規性的情況下，繼續開發工作。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。
- 您決定採用您的一位開發人員在網際網路上找到的一個程式庫。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。
- 需要遷移的最初商務理由是基於 60% 的應用程式工作負載的現代化。但是，由於有技術上的困難，最後決定只進行 20% 的現代化，因而降低了計畫的長期效益，使基礎設施團隊以人工方式支援舊式系統的操作員勞動工作增加，導致組織更加依賴未針對此變更做準備的基礎設施團隊開發新技能。

建立此最佳實務的優勢：完全附和並支持董事會層級的業務優先事項、了解獲致成功的風險、做出明智的決策，並能在出現阻礙成功的風險時迅速採取適當行動。了解決策的影響和後果有助於優先考慮選項，並讓領導者更快達成協議，進而改善業務成果。識別您的選擇的優勢，並了解組織面臨的風險，如此才能協助您做出資料導向的決策，而不是按圖說故事。

未建立此最佳實務時的風險暴露等級：中

實作指引

管理效益和風險應由治理機構定義，而該機構可以驅動關鍵決策要求。您希望根據對組織有利的方式來做決策和訂立優先順序，並了解其中可能牽涉到的風險。準確的資訊對於做出組織決策至關重要。這應

該要以紮實的評估測量為基礎，並須由常見的產業成本效益分析實務定義。若要做出這些類型的決策，請在集中化和分散權限之間取得平衡。權衡是無可避免的，而了解每項選擇對於定義的策略和想要達成的業務成果的影響，是非常重要的一環。

實作步驟

1. 在整體雲端管控架構中，使優勢評估做法變成一項例行作業。
   a. 在某些決策執行的集中控制與分散權限之間取得平衡。
   b. 了解加諸在每一項決策上的重擔及其決策流程，都會拖慢您的腳步。
   c. 將外部因素納入您的決策過程中 (如合規要求)。
2. 為不同層級的決策建立有共議的決策架構，其中包括誰需要解鎖受利益衝突影響的決策。
   a. 集中於可能無法逆轉的單向門決策。
   b. 允許較低職階的組織領導者做出雙向門決策。
3. 了解和管理效益和風險。在決策的收益與所涉及的風險之間取得平衡。
   a. 識別效益：根據業務目標、需求和優先事項識別效益。範例包括業務案例影響、上市時間、安全性、可靠性、效能和成本。
   b. 識別風險：根據業務目標、需求和優先事項識別風險。範例包括上市時間、安全性、可靠性、效能和成本。
   c. 評估風險與效益並做出明智決策：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，確定效益和風險的影響。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。
4. 以程式設計方式強制執行關鍵決策，讓您自動遵守合規要求。
5. 利用已知的產業架構和功能 (例如價值流分析和 LEAN)，以目前狀態績效和業務指標為基準，並定義改進這些指標的進度迭代。

實作計畫的工作量：中高

資源

相關的最佳實務：

- OPS01-BP05 評估威脅態勢

相關文件：

- Amazon 首日文化要素 | 做出高品量的高速決策
- 雲端治理
- 管理與治理雲端環境
- 雲端治理和數位時代治理：第一和第二部分

相關影片：

- 播客 | Jeff Bezos | 關於如何做決策

相關範例：

- 利用資料做出明智決策 (DevOps 傳奇)
- 使用開發價值流圖識別 DevOps 結果的限制

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。團隊應了解本身在促成其他團隊成功的過程中所扮演的角色、其他團隊在獲致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 資源已確認擁有者
- OPS02-BP02 流程和程序已確認擁有者
- OPS02-BP03 已為營運活動識別負責其效能的擁有者
- OPS02-BP04 存在管理責任和擁有權的機制
- OPS02-BP05 存在用於要求新增、變更和例外狀況的機制
- OPS02-BP06 團隊之間的責任是預先定義或經過協商的

OPS02-BP01 資源已確認擁有者

工作負載的資源必須已識別變更控制、疑難排解和其他功能的擁有者。系統會為工作負載、帳戶、基礎設施、平台和應用程式指派擁有者。擁有權會使用集中註冊或連接至資源的中繼資料等工具來記錄。元件的商業價值會透露其適用的流程和程序。

期望的結果：

- 資源已使用中繼資料或集中註冊識別擁有者。
- 團隊成員可識別誰擁有資源。
- 帳戶會盡可能擁有單一擁有者。

常見的反模式：

- AWS 帳戶 的替代聯絡人未填入。
- 資源缺少用來識別哪些團隊是其擁有者的標籤。
- 您有不具備電子郵件對應的 ITSM 佇列。
- 兩個團隊對於基礎設施的關鍵部件有重疊的擁有權。

建立此最佳實務的優勢：

- 有了指派的擁有權，資源的變更控制將是簡單明瞭的。
- 對問題進行疑難排解時，您將可接洽正確的擁有者。

未建立此最佳實務時的風險暴露等級：高

實作指引

定義擁有權對環境中的資源使用案例的意義。擁有權可表示誰負責監督資源的變更、誰支援疑難排解期間的資源，或財務責任由誰承擔。指定並記錄資源的擁有者，包括名稱、聯絡資訊、組織和團隊。

客戶範例

AnyCompany Retail 將擁有權定義為擁有資源變更和支援的團隊或個人。他們使用 AWS Organizations 來管理其 AWS 帳戶。替代帳戶聯絡人使用群組收件匣進行設定。每個 ITSM 佇列分別對應至一個電子郵件別名。標籤會指出誰擁有 AWS 資源。對於其他平台和基礎設施，他們有 Wiki 頁面會指出擁有權和聯絡資訊。

實作步驟

1. 首先為您的組織定義擁有權。擁有權可暗示資源的風險由誰承擔、誰擁有資源的變更，或誰支援疑難排解期間的資源。擁有權也可暗示資源的財務或管理擁有權。
2. 使用 AWS Organizations 管理帳戶。您可以集中管理帳戶的替代聯絡人。
   a. 只要使用公司擁有的電子郵件地址和電話號碼作為聯絡資訊，即使聯絡資訊所屬的個人已離職，您仍可存取這些資訊。例如，為帳單、營運和安全建立各別的電子郵件分發清單，在每個作用中

的 AWS 帳戶 中將這些設定為帳戶、安全和營運聯絡人。即使某人休假、職務變動或離職，仍有多人會收到 AWS 通知並且能有所回應。

b. 如果帳戶未由 AWS Organizations 管理，替代帳戶聯絡人可協助 AWS 在必要時聯繫到適當人員。設定帳戶的替代聯絡人以將其指向團體而非個人。

3. 使用標籤來識別 AWS 資源的擁有者。您可以用個別的標籤指定擁有者及其聯絡資訊。
   a. 您可以使用 AWS Config 規則強制資源要有必要的擁有權標籤。
   b. 如需如何為組織建置標記策略的深入指引，請參閱 AWS 標記最佳實務白皮書。
4. 您可以使用 Amazon Q Business 這款運用生成式 AI 技術的對話式助理來提高員工生產力、回答問題，並根據企業系統中的資訊完成任務。
   a. 將 Amazon Q Business 連線到公司的資料來源。Amazon Q Business 提供 40 多個受支援的資料來源的預先建置連接器，包括 Amazon Simple Storage Service (Amazon S3)、Microsoft SharePoint、Salesforce 和 Atlassian Confluence。如需詳細資訊，請參閱 Amazon Q Business 連接器。
5. 對於其他資源、平台和基礎設施，請建立識別擁有權的文件。此文件應開放給所有團隊成員存取。

實作計畫的工作量：低。利用帳戶聯絡資訊和標籤指派 AWS 資源的擁有權。對於其他資源，您可以使用 Wiki 表格這類簡單的工具來記錄擁有權與聯絡資訊，或使用 ITSM 工具來對應擁有權。

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者
- OPS02-BP04 存在管理責任和擁有權的機制

相關文件：

- AWS 帳戶管理 - 更新聯絡資訊
- AWS Organizations - 更新組織中的替代聯絡人
- AWS 標記安全最佳實務白皮書
- 使用 Amazon Q 企業版和 AWS IAM Identity Center 建置私有且安全的企業生成式 AI 應用程式
- 現在普遍可用的 Amazon Q Business，透過生成式 AI 幫助提高員工生產力
- AWS 雲端 作業與遷移部落格 - 使用 AWS Config 和 AWS Organizations 實作自動化和集中化的標記控制
- AWS 安全性部落格 - 使用 AWS CloudFormation Guard 擴展您的預先提交勾點

- AWS DevOps 部落格 - 將 AWS CloudFormation Guard 整合到 CI/CD 管道中

相關研討會：

- AWS 研討會 - 標記

相關範例：

- AWS Config 規則 - Amazon EC2 具有需要的標籤和有效值

相關服務：

- AWS Config 規則 - required-tags
- AWS Organizations

OPS02-BP02 流程和程序已確認擁有者

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，有助於找出改進機會。

期望的結果：您的組織擁有一組完善定義和受維護的作業流程和程序。流程和程序儲存在集中的位置，可供您的團隊成員使用。流程和程序經常依據指派的擁有權進行更新。在可能的情況下，指令碼、範本和自動化文件的實作方式即程式碼。

常見的反模式：

- 程序未記錄。隔離的操作員工作站可能存在片段指令碼。
- 關於如何使用指令碼的知識由少數個人持有，或為非正式的團隊知識。
- 舊版程序即將進行更新，但更新的擁有權不清楚，且原始作者已經離職。
- 程序和指令碼無法進行探索，因此無法適時提供使用 (例如，要回應事故時)。

建立此最佳實務的優勢：

- 流程和程序可以大幅提高工作負載的工作量。
- 新的團隊成員工作更快、效率更高。
- 縮短緩解事件的時間。

- 不同的團隊成員 (和團隊) 可以採用一致方式，使用相同的流程和程序。
- 團隊可用可重複的程序來擴展其程序。
- 標準化的流程和程序有助於減輕團隊之間轉移工作負載職務的影響。

未建立此最佳實務時的風險暴露等級：高

實作指引

- 流程和程序已確認擁有者負責其定義。
  - 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。
  - 唯一識別負責活動規格的個人或團隊。他們負責確認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。
  - 透過 AWSSystems Manager 等服務、文件與 AWS Lambda，擷取活動成品中繼資料中的擁有權。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。
- 長期下來，這些程序應該會獲得改善，而達到可依程式碼執行，減少人工介入的必要性。
  - 例如，考慮 AWS Lambda 函數、CloudFormation 範本或 AWSSystems Manager 自動化文件。
  - 在適當的儲存庫中執行版本控制。
  - 包含合適的資源標籤，以便識別擁有者和文件。

客戶範例

AnyCompany Retail 將擁有權定義為負責應用程式或應用程式群組 (而群組中的應用程式共用常見的架構實務和技術) 之程序的團隊或個人。一開始，流程和程序會作為逐步指南記錄在文件管理系統中，供團隊在託管應用程式的 AWS 帳戶 以及在帳戶下特定資源群組上，使用標籤進行探索。他們使用 AWS Organizations 來管理其 AWS 帳戶。長期下來，這些程序會轉換為程式碼，而資源會以基礎設施為程式碼 (例如 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本) 來定義。作業程序會成為在 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，可以啟動做為排程的任務，以便回應 AWS CloudWatch 警報或 AWS EventBridge 事件，或是因應 IT 服務管理 (ITSM) 平台內的請求而啟動。所有程序都有可識別其擁有權的標籤。自動化和程序的文件會從該程序程式碼儲存庫生成的 wiki 頁面中獲得維護。

實作步驟

1. 記錄現有的流程和程序。

a. 檢閱並保持為最新狀態。

b. 識別每個流程或程序的擁有者。

c. 實施版本控制。

d. 在可能情況下，於共用架構設計的工作負載和環境之間共用流程和程序。

2. 建立意見回饋和改善的機制。

a. 定義程序應多久檢閱一次的政策。

b. 定義檢閱者與核准者適用的程序。

c. 實作問題或票務佇列，以便提供與追蹤意見回饋。

d. 在可能的情況下，流程和程序應由變更核准委員會 (CAB) 進行預先核准和風險分類。

3. 確認執行流程和程序可以供需要執行的人存取和探索。

a. 使用標籤，指出可供工作負載存取流程和程序的所在位置。

b. 使用有意義的錯誤和事件訊息，指出可解決問題的適當流程或程序。

c. 使用 Wiki 和文件管理，讓這些流程和程序一致可供整個組織的人搜尋。

4. 適當時機進行自動化。

a. 當服務和技術提供 API 時，應該開發自動化。

b. 充分教育有關程序的相關知識。發展那些流程的用戶故事和自動化需求。

c. 成功評估流程和程序的使用情況，並提出支援反覆改進的問題。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS02-BP01 資源已確認擁有者
- OPS02-BP04 存在管理責任和擁有權的機制
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 - AWS 上的 DevOps 簡介
- AWS 白皮書 - 標記 AWS 資源的最佳實務
- AWS 白皮書 - 使用多個帳戶整理您的 AWS 環境

- AWS 雲端 作業與遷移部落格 - 建置雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- AWS 雲端 作業與遷移部落格 - 使用 AWS Config 和 AWS Organizations 實作自動化和集中化的標記控制
- AWS 安全性部落格 - 使用 AWS CloudFormation Guard 擴展您的預先提交勾點
- AWS DevOps 部落格 - 將 AWS CloudFormation Guard 整合到 CI/CD 管道中

相關研討會：

- AWS Well-Architected 卓越營運研討會
- AWS 研討會 - 標記

相關影片：

- 如何在 AWS 上將 IT 作業自動化
- AWS re:Invent 2020：使用 AWS Systems Manager 將任何作業自動化
- AWS re:Inforce 2022 - 使用 AWS (NIS306)，自動化修補程式管理與合規
- AWS Support 您 - 深入探討 AWS Systems Manager

相關服務：

- AWS Systems Manager - 自動化
- AWS Service Management Connector

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。

期望的結果：

您的組織明確定義職責，以對定義的工作負載執行特定活動，並回應工作負載產生的事件。組織會記錄流程和履行的擁有權，並讓此資訊可被搜尋到。當組織發生變更時，您可以檢閱和更新職責內容，團隊成員便可據以追蹤及衡量缺失及效率不彰的身分識別活動績效。您可以實作回饋機制來追蹤缺失和改進之處，並支援反覆改進。

常見的反模式：

- 您不記錄職責。
- 隔離的操作員工作站存在片段指令碼。只有少數個人知道如何使用這些指令碼，或將其非正式稱為團隊知識。
- 舊版處理序的更新日期已到期，但沒有人知道該處理序的擁有者，而原始作者已離開組織。
- 處理序和指令碼均無法找到，因此無法適時供人使用 (例如，要回應事故時)。

建立此最佳實務的優勢：

- 您了解誰負責執行活動，也知道在需要採取動作時通知誰，以及誰會執行動作、驗證結果，以及提供回饋給活動擁有者。
- 流程和程序可以大幅提高工作負載的工作量。
- 新的團隊成員工作更快、效率更高。
- 您可以減少緩解事件所需的時間。
- 不同的團隊採用一致方式，使用相同的流程和程序執行任務。
- 團隊可用可重複的程序來擴展其程序。
- 標準化的流程和程序有助於減輕團隊之間轉移工作負載職務的影響。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要開始定義職責，請從現有的文件開始，例如責任矩陣、流程和程序、角色和職責，以及工具和自動化。檢閱並主持有關已記錄流程的責任討論。與團隊一起檢閱以識別文件職責和流程之間的不一致之處。與該團隊的內部客戶一起探討提供的服務，以識別各團隊之間的期望差距。

分析並解決差異。找出改進的機會，並尋找經常受請求的資源密集型活動，這些活動通常是最需要改進的對象。探索最佳實務、模式和規範指引，以簡化和標準化改進事宜。記錄改進機會，並追蹤需完成的改進情況。

長期下來，這些程序應該會演變為以程式碼形式執行，以減少人工介入的必要性。例如，程序可以起始為 AWS Lambda 函數、AWS CloudFormation 範本或 AWS Systems Manager 自動化文件。確認這些程序是否在適當的儲存庫中執行版本控制，而且包含適用的資源標記，以方便各團隊輕鬆識別擁有者和文件。記錄執行活動的責任，然後監控自動化是否成功啟動和操作，以及期望結果的表現。

客戶範例

AnyCompany Retail 將擁有權定義為負責應用程式或應用程式群組 (而群組中的應用程式共用常見的架構實務和技術) 之程序的團隊或個人。起初公司將流程和程序記錄為文件管理系統中的逐步指南。公司使用託管應用程式的 AWS 帳戶 和帳戶內特定資源組上的標籤來探索程序，並使用 AWS Organizations 來管理其 AWS 帳戶。過了一段時間，AnyCompany Retail 將這些流程轉換為程式碼，並使用基礎設施即程式碼 (透過 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本等服務) 定義資源。作業流程會成為在 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，可以啟動為排程任務，以回應 Amazon CloudWatch 警示或 Amazon EventBridge 事件等事件，或是因應 IT 服務管理 (ITSM) 平台內的請求而啟動。所有流程都有標籤用來識別擁有者。團隊會在流程程式碼儲存庫產生的 Wiki 頁面中管理自動化和流程的文件。

實作步驟

1. 記錄現有的流程和程序。
    a. 檢閱並確認文件是否為最新版本。
    b. 確認每個流程或程序都有各自擁有者。
    c. 將程序置於版本控制下。
    d. 在可能情況下，於共用架構設計的工作負載和環境之間共用流程和程序。
2. 建立意見回饋和改善的機制。
    a. 定義程序應多久檢閱一次的政策。
    b. 定義檢閱者與核准者適用的程序。
    c. 處理問題或實作票務佇列以提供和追蹤回饋。
    d. 在可能的情況下，由變更核准委員會 (CAB) 為流程和程序提供預先核准和風險分類。
3. 讓執行流程和程序可供需要執行的人存取和探索。
    a. 使用標籤，指出可供工作負載存取流程和程序的所在位置。
    b. 使用有意義的錯誤和事件訊息，指出可解決問題的適當流程或程序。
    c. 使用 Wiki 或文件管理，讓這些流程和程序一致可供整個組織的人搜尋。
4. 在適當的情況下實行自動化。
    a. 只要服務和技術提供 API，就可以開發自動化程序。
    b. 確認流程是否已獲充分了解，並發展用戶故事和需求，以自動化這些流程。
    c. 衡量流程和程序的成功使用情況，並利用問題追蹤來支援反覆改進。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS02-BP01 資源已確認擁有者
- OPS02-BP02 流程和程序已確認擁有者
- OPS02-BP04 存在管理責任和擁有權的機制
- OPS02-BP05 存在用來識別責任和擁有權的機制
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 | AWS 上的 DevOps 簡介
- AWS 白皮書 | 標記 AWS 資源的最佳實務
- AWS 白皮書 - 使用多個帳戶整理您的 AWS 環境
- AWS 雲端 作業與遷移部落格 | 建置雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- AWS 研討會 - 標記
- AWS 服務管理連接器

相關影片：

- AWS 知識中心直播 | 標記 AWS 資源
- AWS re:Invent 2020 | 使用 AWS Systems Manager 將任何作業自動化
- AWS re:Inforce 2022 | 使用 AWS (NIS306)，自動化修補程式管理與合規
- AWS Support 您 - 深入探討 AWS Systems Manager

相關範例：

- AWS Well-Architected 卓越營運研討會

OPS02-BP04 存在管理責任和擁有權的機制

了解您角色的責任以及為商業成果做出貢獻的方式，如此即可得知任務的優先順序以及您的角色為何很重要。這有助於讓團隊成員辨識需求，並適當地回應。當團隊成員認識到自己的角色時，就會建立擁有權、找出改進機會，並了解如何影響或做出適當的轉變。

偶而，某責任可能沒有明確的歸屬者。在這些情況下，設計一個機制可彌補這個落差。為有權指派擁有權或計劃要解決需求的人建立定義明確的向上呈報路徑。

期望的結果：組織內的團隊具有明確定義的職責，其中包括與資源、要執行的動作、流程和程序的關係。這些責任與團隊的責任和目標以及其他團隊的責任是一致的。您可以用一致且可探索的方式記錄向上呈報路徑，並將這些決策輸入文件成品，例如責任矩陣、團隊定義或 Wiki 頁面。

常見的反模式：

- 團隊的職責模棱兩可或定義不清。
- 團隊未將角色與職責劃上等號。
- 團隊未使其整體目標和具體目標與職責保持一致，因此難以衡量成功。
- 團隊成員的職責未與團隊和整體組織劃上等號。
- 您的團隊未將其職責更新到最新狀態，這使得其與團隊執行的任務不一致。
- 用於確定職責的向上呈報路徑尚未定義或不明確。
- 向上呈報路徑沒有單一執行團隊擁有者，以確保及時回應。
- 角色、職責和向上呈報路徑均無法找到，且在需要時無法隨時派上用場 (例如，在回應事件時)。

建立此最佳實務的優勢：

- 當您了解誰有責任或誰是擁有者時，您可以聯絡適當的團隊或團隊成員以提出請求或轉移任務。
- 為了降低不採取行動和未解決需求的風險，您已確定一個有權指派職責或擁有權的人選。
- 當您清楚定義責任範圍時，您的團隊成員就能獲得自主權和擁有權。
- 從您的職責了解您做的決定、採取的動作，以及如何將活動交給適當的擁有者。
- 識別放棄的責任並不難，因為您明確了解哪些事不在團隊職責範圍內，這有助於向上呈報並澄清。
- 團隊得以避免混亂和緊張，並且更充分地管理其工作負載和資源。

未建立此最佳實務時的風險暴露等級：高

實作指引

識別團隊成員的角色和責任，並確定他們了解加注在其角色的期望。讓此資訊可供探索，如此一來，組織的成員便能夠確定有特定需求時該聯絡誰：團隊或個人。隨著組織尋求利用在 AWS 上遷移和現代化的機會，角色和職責也可能發生變化。讓您的團隊及其成員了解其責任，並在此變化期間訓練他們適度執行其任務。

確定應接受呈報的角色或團隊，以識別責任和擁有權。此團隊可以和各種利害關係人互動，以做出最後決定。但是，他們應該擁有決策過程的管理權。

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。這些機制會教導他們有特定需求時應聯絡誰。

客戶範例

AnyCompany Retail 最近以平移方法完成工作負載從內部部署環境到 AWS 登陸區域的遷移。他們進行了作業審查，反思如何完成常見的作業任務，並驗證其現有的責任矩陣是否反映了新環境的運作。當他們從內部部署遷移到 AWS 時，他們就減少了與硬體和實體基礎設施相關的基礎設施團隊責任。這個運作也揭露了為其工作負載改進和發展作業模式的新機會。

雖然他們識別、解決和記錄大部分職責，但他們也會為任何錯過的或隨著營運上的實務演化而需要改變的任何職責定義向上呈報路徑。若要探索整個工作負載標準化和提高效率的新機會，請提供營運工具 (例如 AWS 和 Systems Manager) 以及安全工具 (例如 AWS Security Hub 和 Amazon GuardDuty) 的存取權。AnyCompany Retail 根據他們希望先解決的改進問題，對職責和策略進行審查。隨著公司採用新的工作方式和技術模式，他們更新自己的責任矩陣以與之相符。

實作步驟

1. 從現有文件開始。一些典型的來源文件可能包括：
   a. 責任或負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣
   b. 團隊定義或 Wiki 頁面
   c. 服務定義和供應項目
   d. 角色或工作描述
2. 審查並主持有關記錄的責任討論：
   a. 與團隊一起審查，識別團隊通常執行的記錄責任和責任之間的不一致性。
   b. 討論內部客戶提供的潛在服務，以識別團隊之間的期望落差。
3. 分析並解決差異。
4. 識別改進機會。

a. 識別經常提出的資源密集型請求，這些請求通常是最需要改進的對象。
b. 尋找最佳實務、模式和規範指引，透過本指引簡化和標準化改進事宜。
c. 記錄改進機會，並追蹤直至完成。

5. 如果團隊尚未進行管理和追蹤責任分派，請確定團隊中擔負此職責的人。
6. 定義團隊請求解釋責任的流程。
   a. 檢閱該流程，並確認流程是否夠清晰且易於使用。
   b. 確保有人扛責並追蹤呈報至得出結論。
   c. 建立營運指標以衡量效用。
   d. 建立回饋機制，確認團隊可以突顯改進機會。
   e. 實施定期檢討的機制。
7. 以可探索且可存取的位置記錄文件。
   a. Wiki 或文件入口網站是共同的選擇。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS01-BP06 評估權衡
- 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-BP07 適當地為團隊提供資源
- OPS09-BP01 使用指標衡量營運目標與 KPI
- OPS09-BP03 檢閱營運指標並優先改進
- OPS11-BP01 建立持續改進程序

相關文件：

- AWS 白皮書 - AWS 上的 DevOps 簡介
- AWS 白皮書 - AWS 雲端 採用架構：營運觀點
- AWS Well-Architected Framework 卓越營運 - 工作負載層級作業模式拓撲
- AWS 規範性指引 - 建置您的雲端作業模式

- AWS 規範性指引 - 為雲端作業模式建立 RACI 或 RASCI 矩陣
- AWS 雲端 營運與遷移部落格 - 透過雲端平台團隊提供商業價值
- AWS 雲端 營運與遷移部落格 - 為什麼要使用雲端作業模式？
- AWS DevOps 部落格 - 組織如何將雲端作業現代化

相關影片：

- AWS 線上峰會 - 加速轉型的雲端作業模式
- AWS re:Invent 2023 - 不過時的雲端安全防護：全新作業模式

OPS02-BP05 存在用於要求新增、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要求。要求包含新增、變更和例外狀況。這些要求會經歷變更管理程序。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要求。

期望的結果：

- 您可以根據指派的擁有權提出變更流程、程序和資源的要求。
- 權衡利益與風險，審慎進行變更。

常見的反模式：

- 您必須更新您部署應用程式的方式，但無法透過營運團隊要求變更部署程序。
- 災難復原計畫必須更新，但沒有已識別的擁有者可接受變更的要求。

建立此最佳實務的優勢：

- 流程、程序和資源可能隨著要求的變更而演變。
- 擁有者可做出關於何時應變更的明智決策。
- 審慎進行變更。

未建立此最佳實務時的風險暴露等級：中

實作指引

若要實作此最佳實務，您必須能夠要求變更流程、程序和資源。變更管理程序可以精簡。記錄變更管理程序。

客戶範例

AnyCompany Retail 使用責任指派 (RACI) 矩陣來識別誰擁有流程、程序和資源的變更。他們記錄了精簡且容易遵循的變更管理程序。使用 RACI 矩陣和程序，任何人都能提交變更要求。

實作步驟

1. 識別您工作負載的流程、程序和資源，及其各自的擁有者。在您的知識管理系統中加以記錄。
   a. 如果您尚未實作 OPS02-BP01 資源已確認擁有者、OPS02-BP02 流程和程序已確認擁有者 或 OPS02-BP03 已為營運活動識別負責其效能的擁有者，請先予以實作。
2. 與組織中的利害關係人合作制定變更管理程序。此程序應涵蓋資源、流程和程序的新增、變更與例外狀況。
   a. 您可以使用 AWS Systems Manager Change Manager 作為工作負載資源的變更管理平台。
3. 在您的知識管理系統中記錄變更管理程序。

實作計畫的工作量：中。制定變更管理程序時，必須在整個組織的多個利害關係人之間取得共識。

資源

相關的最佳實務：

- OPS02-BP01 資源已確認擁有者 - 在您建置變更管理程序之前，資源必須要有已識別的擁有者。
- OPS02-BP02 流程和程序已確認擁有者 - 在您建置變更管理程序之前，程序必須要有已識別的擁有者。
- OPS02-BP03 已為營運活動識別負責其效能的擁有者 - 在您建置變更管理程序之前，營運活動必須要有已識別的擁有者。

相關文件：

- AWS 方案指引 - AWS 大型遷移的基礎程序手冊：建立 RACI 矩陣
- 雲端中的變更管理白皮書

相關服務：

- AWS Systems Manager Change Manager

OPS02-BP06 團隊之間的責任是預先定義或經過協商的

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、服務水準目標或服務水準協議)。團隊間的溝通管道記錄於文件中。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，並協助他們適當地回應。

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。

期望的結果：

- 團隊間的工作或支援協議經過議定並記錄於文件中。
- 相互支援或合作的團隊定義了溝通管道和回應預期。

常見的反模式：

- 生產過程發生問題，兩個不同的團隊各自起始了疑難排解。其各自為政的工作使中斷更為嚴重。
- 營運團隊需要開發團隊的協助，但雙方並未就回應時間達成協議。要求卡在積存中。

建立此最佳實務的優勢：

- 團隊知道如何彼此互動與支援。
- 眾人對回應能力有相同的預期。
- 溝通管道明確定義。

未建立此最佳實務時的風險暴露等級：低

實作指引

實作此最佳實務意味著，團隊間對於彼此的合作方式不會有歧義。正式協議明訂了團隊相互合作或支援的方式。團隊間的溝通管道記錄於文件中。

客戶範例

AnyCompany Retail 的 SRE 團隊與其開發團隊間有一份服務水準協議。無論開發團隊是否是在其票證系統提出要求的，應該都能在十五分鐘內獲得回應。如果發生站點中斷，SRE 團隊將主導調查，並由開發團隊提供支援。

實作步驟

1. 與組織中的利害關係人合作，根據流程和程序制定團隊之間的協議。
   a. 如果兩個團隊之間共用流程或程序，請制定關於團隊應如何共事的執行手冊。
   b. 如果團隊之間相互依賴，請協議要求的回應 SLA。
2. 在您的知識管理系統中記錄責任。

實作計畫的工作量：中。如果團隊之間目前沒有任何協議，與組織中的利害關係人達成協議可能會頗費周章。

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者 - 必須在設定團隊之間的協議之前識別程序擁有權。
- OPS02-BP03 已為營運活動識別負責其效能的擁有者 - 必須在設定團隊之間的協議之前識別營運活動擁有權。

相關文件：

- AWS Executive Insights - 透過雙披薩團隊增添創新動能
- DevOps on AWS 簡介 - 雙披薩團隊

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 提供高層的支持
- 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 團隊成員受到鼓勵來維持和培養自己的技能集
- OPS03-BP07 適當地為團隊提供資源

OPS03-BP01 提供高層的支持

最高階層的資深領導者，也是執行任務發起者，為組織的成果明確訂立期望值和方向，包括評估組織的成功。發起者倡導並推動最佳實務的採用和組織進化。

期望的結果：致力於採用、轉型和最佳化雲端作業的組織，建立了明確的領導和問責制，以實現期望的結果。組織了解其實現新成果所需的每項能力，並將擁有權分派給職能團隊以進行發展。領導者積極定義這個方向、指派擁有權、承擔責任，並定義工作。因此，整個組織的個人可以調動、感受啟發，並積極努力實現想要達到的目標。

常見的反模式：

- 在沒有明確的發起者和雲端作業計畫的情況下，工作負載擁有者可以將工作負載遷移到 AWS。這會造成團隊無法自覺地協同作業來改善和磨練其營運能力。缺乏營運的最佳實務標準會讓團隊不堪重負 (例如操作人員過勞、隨叫隨到和技術負債)，而使創新能力受限。
- 全組織設定了一個新的目標，即採用新興技術，卻不指派領導階層發起者和提供策略。團隊對目標的詮釋不同，這會導致對於努力的重點、其重要的原因以及如何衡量所受到的衝擊感到困惑。因此，組織在採用該技術方面失去了動力。

建立這種最佳實務的優勢：當高層發起者明確溝通並分享願景、方向和目標時，團隊成員就會明白對方對他們抱有什麼期望。當領導者積極參與時，個人和團隊便會開始將努力目標集中在同一個方向，以利於達成所定義的具體目標。如此一來，組織便能將獲致成功的能力最大化。當您評估成功時，更能夠識別出成功的障礙，然後透過高層發起者的介入來移除這些障礙。

未建立此最佳實務時的風險暴露等級：高

實作指引

- 若要在雲端旅程的每個階段 (遷移、採用或最佳化) 中取得成功，最高層領導就必須透過指定的執行任務發起者主動參與過程。執行任務發起者會根據定義的策略調整團隊的思維、技能和工作方式。
  - 解釋原因：闡明並解釋願景和策略背後的原因。
  - 設定期望：為您的組織定義和發佈目標，包括衡量進度和成功的方式。
  - 追蹤目標的達成情況：定期評估目標的遞增實現情況 (不僅是完成任務)。分享結果，以便在結果有風險時，可以採取適當的行動。
  - 提供實現目標所需的資源：將人員和團隊聚集在一起，共同合作建立合適的解決方案，以達成所設定的結果。如此便可減少或消除組織摩擦。

- 支持您的團隊：與團隊保持合作，讓您了解團隊的表現，以及是否有外部因素正影響著他們。找出阻礙您團隊前進的障礙。代表您的團隊來協助解決障礙並消除不必要的負擔。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。
- 推動採用最佳實務：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。
- 鼓勵團隊的進化：建立持續改進的文化，並主動從進步和失敗中學習。鼓勵人員和組織的成長和發展。利用資料和故事發展願景和策略。

客戶範例

AnyCompany Retail 正在歷經企業轉型過程，除了快速重塑客戶體驗、提高生產力，也在運用生成式 AI 加速業務成長。

實作步驟

1. 建立單一執行團隊，並指派主要執行任務發起者來領導和推動轉型。
2. 定義轉型的明確業務成果，並指派擁有權和責任。賦予主要管理高層領導和做出重要決策的權利。
3. 確認您的轉型策略非常清晰，並且由執行任務發起者廣泛傳達給組織的每個層級。
    a. 為 IT 和雲端計畫建立明確定義的業務目標。
    b. 記錄關鍵業務指標，以推動 IT 和雲端轉型。
    c. 一致地將願景傳達給負責部分策略的所有團隊和個人。
4. 發展通訊計畫矩陣，指定需要傳遞給指定的領導者、經理和個別貢獻者的訊息。指定應傳遞此訊息的人員或團隊。
    a. 一致且可靠地完成通訊計畫。
    b. 定期透過面對面活動設定和管理期望。
    c. 接受有關通訊有效性的回饋，並據以調整通訊和計畫。
    d. 安排通訊事件，以主動了解團隊的挑戰，並建立一致的回饋迴路，允許在必要時做出修正。
5. 從領導角度積極參與每個計畫，以確定所有受影響的團隊都了解他們負責達成的成果。
6. 在每次狀態會議上，執行任務發起者都應尋找障礙，檢查團隊的建立指標、故事或回饋，並評估實現目標的進度。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS03-BP04 溝通需及時、清楚且可行
- OP11-BP01 建立持續改進程序
- OPS11-BP07 執行營運指標審查

相關文件：

- 清理組織內的阻礙：高度一致
- 現實中的轉型：務實地接近變革
- 成為具前瞻性的企業
- 建置 CCOE 時應避開的 7 大陷阱
- 瀏覽雲端：成功的關鍵績效指標

相關影片：

- AWS re:Invent 2023：生成式 AI 領導者指南：利用歷史塑造未來 (SEG204)

相關範例：

- Prosci：主要發起者的角色和重要性

OPS03-BP02 授權團隊成員在成果有風險時採取動作

由領導階層注入的文化擁有權行為，會鼓勵任何員工覺得自己該代表整個公司挺身而出，而不受其定義的角色和問責所約束。員工可以在風險出現時主動識別風險並採取適當的行動。這種文化使員工能夠依當下情況判斷而做出高價值的決策。

例如，Amazon 以領導原則為最高準則，並據以推動員工的預期行為，亦即面對情況時挺身而出、解決問題、處理衝突並採取行動。

期望的結果：領導者已帶起一種新文化，允許個人和團隊在緊要關頭做出決定，即使他們在組織內的職階較低 (因為長決策是透過可審核的權限和安全機制定義的)。失敗沒什麼大不了，團隊會反覆學習改善決策和反應，以因應未來類似的情況。如果某人的改進行為可以使其他團隊受益，那麼他們就會主動分享這些行為的相關知識。領導者會衡量營運改善進度，並激勵個人和組織採用這類模式。

常見的反模式：

- 組織中沒有明確的指引或機制來說明確定風險時該怎麼做。例如，當員工注意到網路釣魚攻擊時，因他們未及時向資安防護團隊通報，導致組織內大部分成員都受到攻擊。這會導致資料洩露。
- 您的客戶抱怨服務無法使用，這主要是由於部署失敗所致。您的 SRE 團隊負責部署工具，其長期藍圖中包含部署作業自動復原。在最近推出的一款應用程式中，其中一名工程師設計了一種解決方案，可自動將其應用程式恢復到舊版本。儘管他們的解決方案可以變成 SRE 團隊的模式，但由於沒有流程可追蹤此類改進，其他團隊並未採用。該組織繼續受到部署失敗的困擾，影響客戶並導致進一步的負面情緒。
- 為了保持合規性，您的 infosec 團隊會監督一個長期的流程，代表連接到其 Amazon EC2 Linux 執行個體的操作員定期輪換共享 SSH 金鑰。infosec 團隊需要幾天才能完成輪換金鑰，而且您無法連線到這些執行個體。infosec 內部或外部沒有人建議使用 AWS 上的其他選項來達到相同的結果。

建立這種最佳實務的優勢：透過下放決策權並授權您的團隊做出關鍵決策，您可以更快速解決問題，並提高成功率。此外，團隊開始意識到擁有感，並且失敗是可以接受的。實驗成為文化支柱。高層主管和總監並不覺得他們在工作的每個方面都受到微觀管理。

未建立此最佳實務時的風險暴露等級：中

實作指引

1. 發展一個能夠接受失敗發生之可能性的文化。
2. 為組織內各種職能領域定義明確的擁有權和責任。
3. 向每個人傳達擁有權和責任，以便他們知道誰可以幫助他們推動分散式決策。
4. 定義您的單向和雙向門決策，以幫助個人知道何時需要向上呈報。
5. 建立組織意識，讓所有員工在結果面臨風險時，都有能力在不同層面採取行動。為您的團隊成員提供文件管控、許可權、工具和機會，以讓其練習有效回應所需的技能。
6. 讓您的團隊成員有機會練習回應各種決策所需的技能。定義決策層級後，請執行「演練日」以驗證所有個別貢獻者是否了解並能展示流程。
   a. 提供替代的安全環境，讓員工在其中可測試和訓練流程及程序。
   b. 確認並建立認知，讓團隊成員明白自己在結果出現預先定義的風險等級時有權採取行動。
   c. 透過指派許可和對其支援的工作負載和元件的存取權，定義團隊成員採取動作的許可權限。
7. 提供團隊分享學習心得的能力 (營運成功和失敗)。
8. 讓團隊有能力挑戰現狀，並提供機制來追蹤和衡量改進，以及其對組織的影響。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS01-BP06 在管理效益和風險的同時評估權衡
- OPS02-BP05 存在用來識別責任和擁有權的機制

相關文件：

- AWS 部落格文章 | 敏捷式企業
- AWS 部落格文章 | 衡量成功：諍論和計畫
- AWS 部落格文章 | 放手：啟用團隊自主性
- 集中化或分散式？

相關影片：

- re:Invent 2023 | 如何不使您的轉型遭到破壞 (SEG201)
- re:Invent 2021 | Amazon 建置者資料中心：Amazon 的卓越營運
- 集中化與 分散式

相關範例：

- 使用架構式決策記錄來簡化軟體開發專案的技術決策

OPS03-BP03 鼓勵向上呈報

如果團隊成員認為期望的結果存在風險，而且不符合預期的標準，則領導者會鼓勵團隊成員向高層決策者和利害關係人呈報所擔憂的問題。這是組織文化的一個特色，無論職級高或低，所有人都會這麼做。呈報時間越早越好，且次數越多越好，以便識別風險，並防止風險引發成事件。領導階層不會因向上呈報問題而懲戒個人。

期望的結果：組織內所有的人都很樂意將問題向上呈報至其直屬主管和更高層領導者。領導階層已經特意為團隊成員打好預防針，讓他們能安心呈報任何問題。在組織裡，有一個機制可以讓每個職級的人向上呈報問題。當員工想要呈報給他們的主管時，會共同判定問題的影響程度，以及是否應向上呈報。為

了開始向上呈報的流程，員工需要同時提交一份包含解決問題建議的工作計畫。如果直屬主管沒有及時採取行動，且如果員工認為組織面臨的風險很高，則員工應將問題向上呈報給最高職級的領導者。

常見的反模式：

- 在雲端轉型計畫近況討論會議上，高層執行領導不會提出足夠的探查性問題來尋找發生問題和阻礙之處。只有好消息才會納入近況報告。CIO 明確表示，她只喜歡聽好消息，因為任何提出的挑戰都會使 CEO 認為該計畫不成功。
- 您是一名雲端營運工程師，您注意到應用團隊並未廣泛採用新知識管理系統。公司投入了一年時間和數百萬美元來實作這個新知識管理系統，但員工仍在本機編寫他們的執行手冊，並分享到組織雲端共享區上，因此其他人很難找到與支援工作負載相關的知識。您嘗試讓領導階層注意這個問題，因為唯有一致地使用這個系統才能提高營運效率。當您將這個問題呈報給領導知識管理系統實作的總監時，她會問責於您，因為她認為您的呈報使公司對這項投資產生質疑。
- 負責加強運算資源的 infosec 團隊已決定制定一個流程，就是在運算團隊釋放資源供組織使用之前，需要執行必要的掃描以確保 EC2 執行個體完全受保護。這導致部署資源的時間延遲一週，進而破壞了他們的 SLA。運算團隊害怕將這個問題呈報給雲端副總裁，因為這會讓資安副總裁難堪。

建立此最佳實務的優勢：

在業務受到影響之前便將複雜或關鍵問題解決。減少時間浪費。風險降至最低。團隊在解決問題時變得更主動，並更專注於結果。

未建立此最佳實務時的風險暴露等級：高

實作指引

在組織中每個職級都願意且能夠自如地向上呈報任何問題，這是整體組織的文化根基，而這個根基仰賴有意識的發展，亦即持續強調訓練、與領導階層溝通、設立期望值，以及在整個組織各個層級部署機制。

實作步驟

1. 為您的組織定義政策、標準和期望。
    1. 確保廣泛採用和了解政策、期望和標準。
2. 鼓勵、培訓並賦予員工能力，以在不符合標準時及早向上呈報。
3. 組織認可儘早且經常呈報是最佳實務。接受呈報經證明可能是無根據的，然而有機會防止事件的發生，總好過於不呈報而錯過該機會。
    a. 建立一個向上呈報機制 (如安燈線系統)。

b. 制定定義進行向上呈報的時機與方式的記錄程序。

c. 定義一序列擁有逐漸升高的採取或核准動作權限的人員，以及每個利害關係人的聯絡資訊。

4. 當向上呈報後應持續追蹤，直到團隊成員確定在領導階層採取行動後已使風險降低。

   a. 向上呈報應包括：

      i. 情況及風險性質描述

      ii. 情況的嚴重性

      iii. 誰或什麼會受影響

      iv. 影響程度有多大

      v. 發生衝擊時的緊急情況

      vi. 建議的補救措施和緩解計畫

   b. 保護向上呈報的員工如果團隊成員越過無回應的決策制定者或利害關係人進行越級呈報，則組織應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。

5. 鼓勵將持續改進回饋的文化注入組織內產生的一切事務中，並且不斷地反覆循環下去。回饋迴圈對負責方而言算是微級呈報程序，即使不需要向上呈報，也能找到改進的機會。持續改進文化會迫使每個人更積極主動。

6. 領導者應定期重新強調政策、標準、機制，以及對開放、不畏報復的向上呈報和持續回饋循環機制的渴望。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS02-BP05 存在用於要求新增、變更和例外狀況的機制

相關文件：

- 您如何培養持續改進和從安燈及向上呈報系統中學習的文化？
- 安燈線 (IT 革命)
- AWS DevOps 指南 | 建立明確的向上呈報路徑，並鼓勵具建設性的意見分歧

相關影片：

- Jeff Bezos 談論如何做出決策 (和加快速度)
- 豐田產品系統：停止生產、按鈕和安燈系統電動板
- LEAN 製造中的安燈線

相關範例：

- 在事件管理員中處理向上呈報計畫

OPS03-BP04 溝通需及時、清楚且可行

領導階層負責建立強大有效的溝通，尤其是當組織採用新的策略、技術或工作方式時。領導階層應該為所有員工設立期望以達成公司目標。設計溝通機制，在負責運作由領導階層資助和贊助的計畫的團隊中，建立並維持這樣的認知。利用跨組織的多樣性，仔細聆聽多種獨特的觀點。使用此觀點來增加創新、挑戰假設，並降低確認偏差的風險。在團隊中培養包容性、多樣性和可及性，以獲得有益的觀點。

期望的結果：您的組織設計溝通策略，以解決變革為組織帶來的影響。各團隊保持資訊暢通且積極，以持續彼此合作，而不是彼此對立。個人了解自己的角色對於實現指定的目標有多重要。明白電子郵件只是用於通訊的被動式機制，並據以使用。管理階層與個人貢獻者共度時間，幫助他們了解自身的責任、應完成的任務，以及如何在工作上為整體使命做出貢獻。必要時，領導者直接在較小的場與員工互動，以傳達訊息，並確認這些訊息是否已有效傳達出去。由於溝通策略良好，組織的表現達到或高於領導階層的期望。領導者鼓勵並尋求團隊內部和跨團隊的不同意見。

常見的反模式：

- 您的組織有個五年計畫，旨在將所有工作負載遷移至 AWS。雲端的業務案例包括將所有工作負載的 25% 現代化，並運用無伺服器技術。CIO 傳達此策略以直接報告，並希望每位領導者將此簡報分享給經理、總監和個人貢獻者，無需面對面溝通。CIO 退居幕後，期望由他的組織執行新策略。
- 領導階層沒有提供或使用回饋機制，致使期望差距加劇，進而導致專案停滯不前。
- 組織要求您對安全群組做出變更，但不通知您需要做哪些變更、變更可能對所有工作負載有何影響，以及何時應該行動等詳細資訊。經理轉發來自資安副總裁的電子郵件，並新增該訊息 "Make this happen."
- 遷移策略遭到變更，將規劃的現代化目標從 25% 降低至 10%。這對下游的營運組織產生影響。他們未被告知這個策略改變，因此來不及準備足夠的專業產能來支援平移到 AWS 的更大量工作負載。

建立此最佳實務的優勢：

- 您的組織對新策略或已變更的策略有充分了解，並在強烈動機下據以採取行動，協助彼此實現領導階層設定的整體目標和指標。
- 存在的機制可用來及時通知團隊成員已知的風險和規劃的事件。
- 組織能更有效地採用新工作方式 (包括對人員或組織、流程或技術的變更) 以及所需的技能，而且能更快地獲取商業利益。
- 團隊成員對接收之通訊內容中的背景有一定了解，因此可以更有效率地工作。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要實作此最佳實務，您必須與組織中的利害關係人共同達成溝通標準的協議。在組織內將這些標準公告週知。對於任何重大的 IT 轉型，已建立的規劃團隊可以比忽略此做法的組織更成功地管理變革對其員工的影響。大型組織在管理變革時可能會遇到更多挑戰，因為讓所有個人貢獻者強力支持新策略是至關重要的一環。如果少了這種過渡型規劃團隊，領導階層就得承擔有效溝通的 100% 責任。建立過渡型規劃團隊時，請指派團隊成員與所有組織領導者合作，以定義和管理每個層級的有效溝通。

客戶範例

AnyCompany Retail 已註冊了 AWS Enterprise Support，並依賴其他第三方提供商進行雲端營運。該公司使用聊天和 ChatOps 作為營運活動的主要通訊媒介。特定管道中會填入提醒和其他資訊。當有人必須採取行動時，他們會清楚地說明期望的結果，並且在許多情況下，他們會收到可供使用的執行手冊或程序手冊。他們使用變更行事曆來排程生產系統的重大變更。

實作步驟

1. 在組織內建立一個須為工作結果負責的核心團隊，他們會針對組織內多個層級發生的變化建立和啟動溝通計畫。
2. 建立單一執行團隊擁有權以利於監督。提供個別團隊獨立創新的能力，並平衡一致性機制的使用情況，以在正確的層級上進行檢查，並確定正確的展望方向。
3. 與組織中的利害關係人共同達成溝通標準、實務和計畫方面的協議。
4. 確認核心溝通團隊是否與組織和計畫領導者合作，並代表領導者向適當的員工傳達訊息。
5. 建立策略性溝通機制，以透過公告、共享行事曆、全公司會議，以及親自或一對一的方法來管理變革，以便團隊成員對他們應該採取的行動有適當的期望。
6. 提供必要的背景內容、詳細資訊和時間 (如果可能)，以判斷是否需要採取行動。當需要採取行動時，提供所需的行動及其影響力。

7. 實作可促進策略性溝通的工具，例如內部聊天、電子郵件和知識管理。
8. 實作機制來衡量和驗證所有通訊是否都導致預期的結果。
9. 建立一個回饋迴圈，用以衡量所有通訊的有效性，尤其是當整個組織中的變革引起了反對聲時。
10. 對於所有 AWS 帳戶，請針對帳單、安全性和營運建立替代聯絡人。在理想的情況中，每個聯絡人都應該列入電子郵件分發名單，而不是當成特定的個別聯絡人。
11. 建立向上呈報及其反向的通訊計畫，以用來與您的內部和外部團隊 (包括 AWS 支援和其他第三方供應商) 互動。
12. 在每個轉型計畫的生命週期中，始終如一地啟動和執行溝通策略。
13. 在可行情況下，優先處理可重複的動作，以大規模安全地執行自動化。
14. 當人員在自動化操作過程中需要通訊時，通訊的目的應該是通知團隊、進行稽核或變更管理流程的一部分。
15. 分析來自警示系統的通訊，以尋找不斷建立的誤報或警示。移除或變更這些警示，以便在需要人工干預時啟動。如果起始一項提醒，請提供執行手冊或程序手冊。
    a. 您可以使用 AWS Systems Manager Documents 來建置提示用的程序手冊和執行手冊。
16. 已設立機制，以清楚且可行的方式提供風險或計畫事件的通知，並提供足夠的通知，以便適當的回應。使用電子郵件清單或聊天管道，在計畫性事件發生之前傳送通知。
    a. AWS 聊天機器人可在您的組織傳訊平台內用來傳送提醒及回應事件。
17. 提供可存取的資訊來源，您可以在其中發現計畫的事件。提供來自相同系統之計畫事件的通知。
    a. AWS Systems Manager 變更行事曆可用來建立可進行變更的變更時段。這可為團隊成員提供有關於何時可安全進行變更的通知。
18. 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，讓他們可以採取動作。
    a. 您可以訂閱 AWS 安全公告，以接收 AWS 相關漏洞的通知。
19. 尋求多樣化的意見和觀點：鼓勵每個人做出貢獻。為代表性不足的團體提供溝通機會。在會議中輪換角色和責任。
    a. 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。他們可以透過角色，以及與他們可能不會與之交涉的新團隊成員互動，從中獲得經驗和新觀點。他們也可以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。隨著觀點的增加，找出新興商機或新的改進機會。在團隊內的成員之間輪換其他人通常會執行的常見任務，以讓全員了解執行這些任務的需求和影響。
    b. 提供安全且友善的環境：建立政策與控制措施，保護組織內團隊成員身心上的安全。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

積極參與並具備生產力。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，以及環境中未確認的風險)。

c. 鼓勵團隊成員充分參與：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可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便利性支援，以便從他們的貢獻中獲得更高的效益。

資源

相關的最佳實務：

- OPS03-BP01 提供高層的支持
- 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

相關文件：

- AWS 部落格文章 | 責任和賦權是高績效且敏捷度強的組織的關鍵
- AWS管理階層洞察 | 學習擴展創新，而不是複雜性 | 單一執行團隊領導者
- AWS 安全公告
- Open CVE
- AWS SupportSlack 中的應用程式可管理支援案例
- 以 AWS Chatbot 方式管理 Slack 頻道中的 AWS 資源

相關範例：

- Well-Architected 實驗室：清查和修補程式管理 (Level 100)

相關服務：

- AWS Chatbot
- AWS Systems Manager 變更行事曆
- AWS Systems Manager Documents

OPS03-BP05 鼓勵進行試驗

試驗是將新構想轉化為產品和功能的觸媒。試驗可加速學習，讓團隊成員保持興趣和參與度。我們鼓勵團隊成員經常進行試驗以推動創新。即便結果不如預期仍有其價值，至少我們了解到什麼是不該做的。團隊成員不會因取得不理想結果的成功試驗而受懲罰。

預期成果：

- 您的組織鼓勵試驗以促進創新。
- 試驗被視為一種學習機會。

常見的反模式：

- 您想要執行 A/B 測試，但沒有相關機制可執行試驗。您在沒有測試能力的情況下部署了 UI 變更。其結果導致了負面客戶體驗。
- 您的公司只有模擬和生產環境。沒有沙盒環境可用來試驗新功能或產品，因此您必須在生產環境內試驗。

建立此最佳實務的優勢：

- 試驗可帶動創新。
- 透過試驗，您可以更快回應使用者的意見反映。
- 組織可培養學習文化。

未建立此最佳實務時的風險暴露等級：中

實作指引

試驗應以安全的方式執行。利用多種環境進行試驗，而不會損害生產資源。使用 A/B 測試和功能旗標來測試試驗。為團隊成員提供在沙盒環境中執行試驗的能力。

客戶範例

AnyCompany Retail 鼓勵試驗。團隊成員可將其 20% 的工時投入於試驗或學習新技術。他們有沙盒環境可供創新之用。他們可對新功能進行 A/B 測試，用實際使用者的意見反映加以驗證。

實作步驟

1. 與組織中的領導階層共同推行試驗風氣。應鼓勵團隊成員以安全的方式執行試驗。

2. 為團隊成員提供可安全進行試驗的環境。他們必須能夠存取類似生產環境的環境。
    a. 您可以使用個別的 AWS 帳戶 建立沙盒環境，以供試驗之用。AWS Control Tower 可用來佈建這些帳戶。
3. 使用功能旗標和 A/B 測試安全地進行試驗，並收集使用者的意見反映。
    a. AWS AppConfig Feature Flags 提供建立功能旗標的能力。
    b. Amazon CloudWatch Evidently 可用來對受限部署執行 A/B 測試。
    c. 您可以使用 AWS Lambda 版本部署用於 Beta 測試的新版功能。

實作計劃的工作量：高。為團隊成員提供可安全執行試驗的環境，可能需要可觀的投資。為了使用功能旗標或支援 A/B 測試，您也可能需要修改應用程式程式碼。

資源

相關的最佳實務：

- OPS11-BP02 執行事件後分析 - 從事件中學習與試驗同樣為推動創新的重要因子。
- OPS11-BP03 實作回饋迴圈 - 回饋迴圈是試驗的重要環節。

相關文件：

- 深入了解 Amazon 文化：試驗、失敗、客戶至上
- 在 AWS 中建立和管理沙盒帳戶的最佳實務
- 樹立雲端造就的試驗文化
- 在 SulAmérica Seguros 透過雲端實行試驗和創新
- 試驗愈多次，就愈可能成功
- 使用多個帳戶整理您的 AWS 環境 - 沙盒 OU
- 使用 AWS AppConfig Feature Flags

相關影片：

- AWS On Air ft.Amazon CloudWatch Evidently | AWS Events
- AWS On Air San Fran Summit 2022 ft.AWS AppConfig Feature Flags 與 Jira 整合
- AWS re:Invent 2022 - 部署並非發行：使用功能旗標控制您的推出的項目 (BOA305-R)
- 透過 AWS Control Tower 以程式設計方式建立 AWS 帳戶

- 設定會使用 AWS Organizations 最佳實務的多帳戶 AWS 環境

相關範例：

- AWS 創新沙盒
- End-to-end Personalization 101 for E-Commerce

相關服務：

- Amazon CloudWatch Evidently
- AWS AppConfig
- AWS Control Tower

OPS03-BP06 團隊成員受到鼓勵來維持和培養自己的技能集

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 活動和網路研討會和 AWSWell-Architected Labs，而這些資源提供了可教育您團隊的說明、範例和詳細演練。

如 AWS Support、(AWS re:Post、AWS Support Center) 和AWS Documentation 等資源可協助移除技術障礙，並改善營運。透過 AWS Support Center 聯絡 AWS Support，以獲取相關問題的解答。

AWS 也分享我們透過在 Amazon Builders' Library 中操作 AWS 所學到的最佳實務和模式，以及 AWS 部落格和官方 AWS 播客提供的各種其他有用的教育材料。

AWS 培訓 and Certification 包含透過自主進度數位課程進行的免費培訓，以及依照角色或領域劃分的學習計畫。您還可以報名參加由講師指導的培訓，以進一步協助發展團隊的 AWS 技能。

期望的結果：您的組織不斷評估技能差距，並利用結構化的預算和投資來彌補這些差距。團隊透過技能提升活動 (例如獲取領先業界認證) 鼓勵和激勵其成員。團隊可以利用專屬交叉分享知識計畫，例如午餐和學習、Immersion Days、駭客松和遊戲日。您的組織將其知識系統保持最新狀態且相關，以便對團隊成員進行交叉訓練，其中包括新進員工入職訓練。

常見的反模式：

- 在沒有結構化的培訓計畫和預算情況下，團隊會在嘗試跟上技術發展時遇到不確定性，進而導致耗損率增加。
- 做為遷移至 AWS 的一部分，您的組織展示了團隊之間的技能差距，和不同的雲端流暢性。若沒有努力提升技能，團隊會發現自己在傳統和效率低的雲端環境管理下應付過多任務，這容易致使操作人員過勞。這種蠟燭兩頭燒的情況，會使員工不滿意度激增。

建立這種最佳實務的優勢：當您的組織刻意投資於能提高團隊技能的項目時，該投資還能幫助加速和擴展雲端採用和最佳化。針對性的學習計畫可推動創新，並為團隊培養隨時準備處理事件的營運能力。團隊刻意投資於最佳實務的實作和演進。團隊士氣很高，團隊成員重視他們對業務的貢獻。

未建立此最佳實務時的風險暴露等級：中

實作指引

為了採用新技術、促進創新，並跟上需求和責任的變化以支援您的工作負載，組織必須不斷投資團隊的專業成長。

實作步驟

1. 使用結構化的雲端宣傳計畫： AWS Skills Guild 提供顧問式培訓，以提高人員在雲端技能方面的信心，並激發持續學習文化。
2. 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，並支持員工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，或允許資淺團隊成員伴隨資深團隊成員工作，藉以學習他們的做法和技能。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。
3. 鼓勵使用專家技術資源：利用 AWSre:Post 等資源來存取精心挑選的知識，並加入充滿活力的社群。
4. 建立和維護最新的知識庫：使用知識共享平台，例如 Wiki 和執行手冊。使用 AWS re:Post Private 建立您自己可重複使用的專家知識來源，以簡化協作、提高生產力，並加快員工上線速度。
5. 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。為團隊成員提供加入其他團隊機會 (暫時或永久地)，以分享讓整個組織受益的技能和最佳實務。
6. 支援對產業認證的追求和維持：支援團隊成員取得與維護可驗證所學知識並認可其成就的產業認證。

實作計畫的工作量：高

資源

相關的最佳實務：

- OPS03-BP01 提供高層的支持
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 | 雲端採用架構：人員觀點
- 投資於持續學習以發展組織的未來
- AWS Skills Guild
- AWS 培訓 and Certification
- AWS Support
- AWS re:Post
- AWS 開始使用資源中心
- AWS 部落格
- AWS 雲端 合規
- AWS 文件
- 官方 AWS播客。
- AWS 線上技術會談
- AWS 活動和研討會
- AWS Well-Architected 實驗室
- 在 Amazon Builders' Library 中

相關影片：

- AWS re:Invent 2023 | 以雲端的速度重新培訓：將員工變成企業家
- WS re:Invent 2023 | 採用遊戲化方式建立好奇心文化

OPS03-BP07 適當地為團隊提供資源

提供適當數量的專業老練團隊成員，以及工具和資源來支援您的工作負載需求。團隊成員工作負擔過重時會增加人為錯誤的風險。投資工具和資源 (例如自動化) 可擴展您的團隊效率，並幫助他們無需額外的能力，即可支援更多工作負載。

期望的結果：

- 您已為團隊雇用適當的員工，進而添加了符合您的遷移計畫在 AWS 中操作工作所需的技能。隨著您的團隊在遷移專案執行過程中不斷擴大規模，他們已經熟悉了業務計畫在遷移或現代化其應用程式時使用的核心 AWS 技術。
- 您仔細調整您的人員編制計畫，以利用自動化技術和工作流程有效率地利用資源。較小的團隊現在可以代表應用程式開發團隊管理更多基礎設施。
- 隨著作業優先順序變化，主動識別任何資源人員配置限制，以確保業務計畫的成功。
- 審查報告工作勞累 (例如待命工作導致的疲勞或傳呼過於頻繁) 的營運指標，以確認員工沒有過勞情況。

常見的反模式：

- 當您即將開始多年雲端遷移計畫時，您的員工並沒有提升 AWS 技能，這可能在支援工作負載時產生風險，並且降低員工士氣。
- 您的整個 IT 組織正在轉向敏捷的工作方式。企業將產品組合優先定位，並針對需要首先開發哪些功能設定指標。您的敏捷流程不需要團隊為其工作計畫指派故事點。因此，無法得知下一次工作量所需的產能水平，或者您是否具有工作所需的合適技能。
- 您要求 AWS 合作夥伴遷移您的工作負載，而在合作夥伴完成遷移專案後，您卻沒有為團隊準備支援轉移計畫。您的團隊在有效率且有效地支援工作負載方面遇到了困難。

建立此最佳實務的優勢：您的組織中有具備適當技能的團隊成員能支援工作負載。資源配置可以在不影響工作績效的情況下，適應改變的優先順序。結果是團隊不僅能熟練地支援工作負載，同時還能將盡可能多的時間專注在為客戶創新，提高員工滿意度。

未建立此最佳實務時的風險暴露等級：中

實作指引

雲端遷移的資源規劃應該源自與遷移計畫一致的組織層級，而且組織也必須實作所需的作業模式來支援您的新雲端環境。這應包含了解為業務和應用程式開發團隊部署哪些雲端技術。基礎設施和營運領導應該為領導雲端採用作業的工程師規劃技能差距分析、培訓和角色定義。

實作步驟

1. 使用相關的營運指標，例如員工生產力 (例如支援工作負載的成本或操作人員在事件發生期間所耗用的時間)，定義團隊成功準則。
2. 定義資源產能規劃和檢查機制，以確認在需要時可以提供適當平衡的合格產能，並且可以隨時間調整。
3. 建立機制 (例如，每月向團隊傳送問卷調查)，以了解影響團隊的工作相關挑戰 (例如職責加重、技術變更、人員流失或支援的客戶增加)。
4. 使用這些機制與團隊互動，並發現可能遇到導致員工生產力瓶頸的趨勢。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。找出阻礙您團隊前進的障礙。
5. 定期檢閱目前佈建的資源是否仍然足夠，或是否需要額外資源，並做出適當的調整以支援團隊。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS03-BP06 團隊成員受到鼓勵來維持和培養自己的技能集
- OPS09-BP03 檢閱營運指標並優先改進
- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP07 自動回應事件

相關文件：

- AWS 雲端 採用架構：人員觀點
- 成為具前瞻性的企業
- 優先考慮員工技能以推動業務成長
- 高績效組織 - Amazon 雙披薩團隊
- 具備成熟雲端技術的企業如何獲致成功

準備

問題

- OPS 4.如何在工作負載中實作可觀測性？

- OPS 5.如何減少缺陷、幫助輕鬆修復，以及改善生產流程？
- OPS 6.如何緩解部署風險？
- OPS 7.如何知道自己準備好支援工作負載？

OPS 4.如何在工作負載中實作可觀測性？

在工作負載中實作可觀測性，以便了解其狀態，並根據業務需求做出資料驅動的決策。

最佳實務

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- OPS04-BP03 實作使用者體驗遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作分散式追蹤

OPS04-BP01 識別關鍵績效指標

想在工作負載中實作可觀測性，要先了解工作負載狀態，並根據業務需求做出資料驅動的決策。確保監控活動與業務目標保持一致的最有效方式之一，就是定義和監控關鍵績效指標 (KPI)。

預期成果： 有效率的可觀測性實作會與業務目標密切保持一致，確保監控工作始終能夠帶來實際的業務成果。

常見的反模式：

- 未定義 KPI：在沒有明確 KPI 的情況下工作，可能會導致監控過度或不足，而錯過重要訊號。
- 靜態 KPI：未隨著工作負載或業務目標發展而重新檢視或改進 KPI。
- 未能保持一致：專注於與業務成果沒有直接關係的技術指標，或難與實際問題相關聯的技術指標。

建立此最佳實務的優勢：

- 容易識別問題：業務 KPI 通常比技術指標更能清楚呈現問題所在。比起從眾多技術指標中苦苦尋找，業務 KPI 下降的現象，更能有效地指出問題所在。
- 業務一致性：確保監控活動可直接支援業務目標。
- 效率：優先監控資源並關注重要指標。

- 主動積極：找出並解決問題，不讓問題擴大影響業務。

未建立此最佳實務時的曝險等級： 高

實作指引

若要有效地定義工作負載 KPI：

1. 從業務成果開始著手： 在深入研究指標之前，請先了解所需的業務成果。想要增加銷售量、提高使用者參與度，還是加快回應時間？
2. 讓技術指標與業務目標相互關聯： 並非所有技術指標都會直接影響業務成果。找出有直接影響的技術指標，不過，通常更直接的方式是使用業務 KPI 找出問題。
3. 使用 Amazon CloudWatch： 採用 CloudWatch 定義和監控代表您的 KPI 的指標。
4. 定期檢閱和更新 KPI： 隨著工作負載和業務發展，保持 KPI 的相關性。
5. 讓利害關係人參與： 讓技術和業務團隊一起參與定義和檢閱 KPI 的過程。

實作計劃的工作量： 中

資源

相關的最佳實務：

- the section called “OPS04-BP02 實作應用程式遙測”
- the section called “OPS04-BP03 實作使用者體驗遙測”
- the section called “OPS04-BP04 實作相依性遙測”
- the section called “OPS04-BP05 實作分散式追蹤”

相關文件：

- AWS 可觀測性最佳實務
- CloudWatch 使用者指南
- AWS 可觀測性 Skill Builder 課程

相關影片：

- 研擬可觀測性策略

相關範例：

- One Observability 研討會

OPS04-BP02 實作應用程式遙測

應用程式遙測是工作負載可觀測性的基礎。提供遙測相當重要，因為能讓您獲得可付諸行動的洞見，深入了解應用程式的狀態以及實現的技術與業務成果。從疑難排解到衡量新功能的影響，或確保與業務關鍵績效指標 (KPI) 保持一致，應用程式遙測都能為您指出建置、操作和發展工作負載的方式。

指標，日誌和追蹤是構成可觀測性的三大要素。這些要素可做為診斷工具來描述應用程式的狀態。經過一段時間後，這些要素可協助建立基準和識別異常狀況。然而，為了確保監控活動與業務目標保持一致，就必須定義並監控 KPI。與單獨的技術指標相比，業務 KPI 通常更容易找出問題所在。

其他遙測類型 (例如實際使用者監控 (RUM) 和綜合交易) 可與這些主要資料來源相輔相成。RUM 提供即時使用者互動的洞見，而綜合交易則模擬可能的使用者行為，有助於在實際使用者遇到瓶頸之前便偵測到瓶頸。

期望的結果：獲得有關工作負載效能的可付諸行動洞見。這些洞見可讓您做出有關效能最佳化的主動決策、提高工作負載穩定性、使 CI/CD 程序更順暢，並且有效利用資源。

常見的反模式：

- 不完整的可觀測性：忽略在工作負載的每一層納入可觀測性，導致出現可能遮蔽重要系統效能和行為洞見的盲點。
- 分散的資料檢視：當資料分散在多個工具和系統中時，便難以提供涵蓋工作負載運作狀況和效能的全面概覽。
- 使用者回報的問題：這種現象表示未能透過遙測和業務 KPI 監視主動偵測問題。

建立此最佳實務的優勢：

- 明智的決策：透過遙測和業務 KPI 獲得洞見，就能做出資料驅動的決策。
- 改善運作效率：以資料驅動方式善用資源可帶來成本效益。
- 提高工作負載穩定性：更快偵測並解決問題，進而改善正常運作。
- 更順暢的 CI/CD 程序：從遙測資料獲得的洞見，有助於改進程序並交付可靠的程式碼。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要為您的工作負載實作應用程式遙測，請使用 Amazon CloudWatch 和 AWS X-Ray 等 AWS 服務。Amazon CloudWatch 提供全面的監控工具套件，如此一來您就可在 AWS 和內部部署環境中觀察資源和應用程式。還會收集、追蹤和分析指標、合併和監控日誌資料，並且回應資源的變更，以增進您對工作負載運作方式的了解。同時，AWS X-Ray 可讓您追蹤、分析和偵錯應用程式，藉此深入了解工作負載的行為。透過像是服務圖、延遲分佈情形和追蹤時間軸等功能，AWS X-Ray 提供了洞見，讓您深入了解工作負載的效能及影響它的瓶頸。

實作步驟

1. 確定要收集的資料：確定可提供工作負載運作狀況、效能和行為實質洞見的重要指標、日誌和追蹤。
2. 部署 CloudWatch 代理程式： CloudWatch 代理程式的作用在於，方便您從工作負載及其基礎設施中取得系統和應用程式指標和日誌。CloudWatch 代理程式也可用來收集 OpenTelemetry 或 X-Ray 追蹤，並傳送至 X-Ray。
3. 為日誌和指標實作異常偵測：使用 CloudWatch Logs 異常偵測和 CloudWatch 指標異常偵測，自動識別應用程式操作中的異常活動。這些工具使用機器學習演算法來偵測並針對異常狀況發出提醒，進而提高您的監控功能，並加快對潛在的中斷或安全威脅的回應時間。設定這些功能以主動管理應用程式運作狀態和安全性。
4. 保護敏感日誌資料：使用 Amazon CloudWatch Logs 資料保護來隱藏日誌檔中的敏感資訊。此功能在敏感資料經存取前自動偵測和遮罩，進而協助維護隱私權及合規性。實作資料遮罩，以安全地處理和保護敏感詳細資訊，例如個人身分識別資訊 (PII)。
5. 定義和監控業務 KPI：建立與業務成果相符的自訂指標。
6. 使用 AWS X-Ray 檢測您的應用程式：除了部署 CloudWatch 代理程式之外，檢測您的應用程式以發出追蹤資料也至關重要。此程序可提供工作負載行為和效能的進一步洞見。
7. 標準化整個應用程式的資料收集：標準化整個應用程式的資料收集實務。採取一致的方式有助於找出資料關聯並進行分析，進而提供應用程式行為的全面概覽。
8. 實作跨帳戶可觀測性：透過 Amazon CloudWatch 跨帳戶可觀測性提高跨多個 AWS 帳戶 的監控效率。使用此功能時，您可以將來自不同帳戶的指標、日誌檔和警示合併到單一檢視中，進而簡化管理並改善針對組織 AWS 環境中已確認之問題的回應時間。
9. 分析資料並採取行動：資料收集和正規化完成後，可將 Amazon CloudWatch 用於指標和日誌分析，以及將 AWS X-Ray 用於追蹤分析。這類分析可產生有關工作負載運作狀況、效能和行為的洞見，進而引導您進行決策。

實作計畫的工作量：高

資源

相關的最佳實務：

- OPS04-BP01 定義工作負載 KPI
- 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

相關文件：

- AWS 可觀測性最佳實務
- CloudWatch 使用者指南
- AWS X-Ray 開發人員指南
- 檢測分散式系統，以了解運作狀態
- AWS 可觀測性 Skill Builder 課程
- Amazon CloudWatch 最新消息
- AWS X-Ray 最新消息

相關影片：

- AWS re:Invent 2022 - Amazon 的可觀測性最佳實務
- AWS re:Invent 2022 - 研擬可觀測性策略

相關範例：

- One Observability 研討會
- AWS 解決方案程式庫：使用 Amazon CloudWatch 進行應用程式監控

OPS04-BP03 實作使用者體驗遙測

深入了解客戶體驗以及與應用程式的互動情形非常重要。實際使用者監控 (RUM) 和綜合交易正是合適的強大工具。從 RUM 提供的實際使用者互動相關資料，能夠獲悉真實的使用者滿意度，而綜合交易則會模擬使用者互動，有助於偵測潛在問題，提早防範問題影響實際使用者。

預期成果： 提供使用者體驗、主動偵測問題及最佳化使用者互動的整體概觀，從而獲得順暢的數位體驗。

常見的反模式：

- 沒有實際使用者監控 (RUM) 的應用程式：
  - 延遲偵測到問題：如果沒有 RUM，您可能直到收到使用者投訴，才察覺到效能瓶頸或問題。這種被動回應的方式可能導致客戶不滿意。
  - 缺乏使用者體驗洞見：未使用 RUM 代表您無法獲得使用者與應用程式實際互動情形的重要資料，因此也限制了您最佳化使用者體驗的能力。
- 沒有綜合交易的應用程式：
  - 缺少邊緣案例：綜合交易可協助您測試一般使用者可能不常使用，但對於某些業務功能來說相當關鍵的路徑和功能。缺少的話，這些路徑可能無法正常運作並遭到忽視。
  - 在應用程式未使用的情況下檢查問題：定期綜合測試可模擬實際使用者未積極與您的應用程式互動的情況，進而確保系統隨時正常運作。

建立此最佳實務的優勢：

- 主動偵測問題：找出並解決潛在問題，避免進一步影響實際使用者。
- 最佳化使用者體驗：RUM 提供持續的意見回饋，有助於改進並強化整體使用者體驗。
- 裝置和瀏覽器效能的相關洞見：了解您的應用程式在不同裝置和瀏覽器上的效能表現，以便進一步最佳化。
- 經驗證的業務工作流程：定期綜合交易可確保核心功能和重要路徑維持正常且高效率的運作。
- 增強應用程式效能：利用收集自實際使用者資料的洞見來改善應用程式回應能力和可靠性。

未建立此最佳實務時的曝險等級： 高

實作指引

為了利用 RUM 和綜合交易進行使用者活動遙測，AWS 提供了類似以下的服務： Amazon CloudWatch RUM 和 Amazon CloudWatch Synthetics。指標、日誌和追蹤搭配使用者活動資料，可提供深入應用程式運作狀態和使用者體驗的全方位檢視。

實作步驟

1. 部署 Amazon CloudWatch RUM： 將您的應用程式與 CloudWatch RUM 整合，以收集、分析和呈現實際使用者資料。

   a. 使用 CloudWatch RUM JavaScript 程式庫 將 RUM 與您的應用程式整合。
   b. 設定儀表板以視覺化和監控實際使用者資料。
2. 設定 CloudWatch Synthetics： 建立 Canary 或指令碼編寫的常式，以模擬使用者與應用程式的互動。
   a. 定義關鍵應用程式工作流程和路徑。
   b. 使用 CloudWatch Synthetics 指令碼 設計 Canary 以模擬這些路徑的使用者互動。
   c. 排定依指定間隔執行 Canary 並進行監控，確保一致的效能檢查。
3. 分析資料並採取行動： 利用來自 RUM 和綜合交易的資料獲得洞見，並於偵測到異常時採取修正措施。使用 CloudWatch 儀表板和警報隨時掌握情況。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作分散式追蹤

相關文件：

- Amazon CloudWatch RUM 指南
- Amazon CloudWatch Synthetics 指南

相關影片：

- 透過最終使用者洞察與 Amazon CloudWatch RUM 最佳化應用程式
- AWS on Air ft.Amazon CloudWatch 的實際使用者監控

相關範例：

- One Observability 研討會
- Amazon CloudWatch RUM Web 用戶端的 Git 儲存庫

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間

OPS04-BP04 實作相依性遙測

對於監控工作負載所依賴的外部服務和元件運作狀況與效能，相依項遙測至關重要，可提供連線能力、逾時，以及像是 DNS、資料庫或第三方 API 等其他與相依項相關重要事件的寶貴洞見。當檢測應用程式以產生有關這些相依項的指標、日誌和追蹤時，可更清楚了解可能影響工作負載的潛在瓶頸、效能問題或故障。

期望的結果：確保工作負載所依賴的相依項如預期般正常運作，讓您能夠主動解決問題並確保最佳的工作負載效能。

常見的反模式：

- 忽略外部相依項：僅關注內部應用程式指標，而忽略與外部相依項相關的指標。
- 缺乏主動監控：等待問題出現，而非持續監控相依項的運作狀況與效能。
- 單獨運作的監控：使用多種分散的監控工具，如此可能導致僅掌握相依項的部分運作狀況且獲得不一致的資訊。

建立此最佳實務的優勢：

- 改善工作負載可靠性：確保外部相依項穩定運作並保持最佳效能。
- 更快偵測並解決問題：主動找出並解決相依項相關問題，不讓問題影響工作負載。
- 全方位視角：獲得全方位視角，有效掌握影響工作負載運作狀況的內部和外部元件。
- 增強工作負載可擴展性：了解外部相依項的可擴展性限制與效能特性。

未建立此最佳實務時的風險暴露等級：高

實作指引

從識別您的工作負載所依賴的服務、基礎設施和程序開始，實作相依項遙測。將相依項正常運作時的良好條件量化，然後判斷衡量時所需的資料。有了這些資訊，您就可以打造儀表板並設定警示，以便為營運團隊提供這些相依項狀態的洞見。相依項無法按需求運作時，使用 AWS 工具探索並量化其影響。不斷重新檢視您的策略，以考量優先順序、目標和所獲得洞見的變化。

實作步驟

若要有效實作相依項遙測：

1. 識別外部相依項：與利害關係人協作，共同找出工作負載所依賴的外部相依項。外部相依項可能包含各種服務，像是外部資料庫、第三方 API、前往其他環境的網路連線能力路由，以及 DNS 服務。實現有效相依項遙測的第一步，就是徹底了解這些相依項。
2. 擬訂監控策略：清楚了解外部相依項之後，就可以為其量身打造監控策略。這包括了解每一項相依項的重要性、預期行為，以及任何相關的服務層級協議或目標 (SLA 或 SLT)。設定主動警示，以便在發生狀態變更或效能偏差時通知您。
3. 使用網路監控：使用網際網路監視器和網路監視器，提供全球網際網路和網路狀況的全方位洞見。這些工具可協助您了解並回應影響外部相依項的中斷、干擾或效能降低。
4. 使用 AWS Health Dashboard 隨時掌握資訊：它會在 AWS 遇到可能影響服務的事件時，發出警示並提供修復指引。
    a. 監控使用 Amazon EventBridge 規則的 AWS Health 事件，或以程式設計方式與 AWS Health API 整合，以在您收到 AWS Health 事件時自動執行動作。這些可能是一般動作 (例如將所有計畫的生命週期事件訊息傳送到聊天介面) 或特定動作 (例如在 IT 服務管理工具中啟動工作流程)。
    b. 如果您使用 AWS Organizations，請跨帳戶彙總 AWS Health 事件。
5. 使用 AWS X-Ray 檢測您的應用程式：AWS X-Ray 提供關於應用程式及其基礎相依項如何運作的洞察。透過從頭到尾追蹤請求，您就可以找出應用程式所依賴的外部服務或元件的瓶頸或故障。
6. 使用 Amazon DevOps Guru：這項機器學習驅動的服務可識別操作問題，預測重大問題可能在什麼時候發生，並且建議可採取的特定行動。對於獲得相依項洞見並確保它們不是造成操作問題的根源來說，這項服務非常寶貴。
7. 定期監控：持續監控與外部相依項相關的指標和日誌。針對非預期的行為或效能降低的情況設定警示。
8. 變更後驗證：每當有任何外部相依項更新或變更，便驗證其效能並檢查是否符合您的應用程式需求。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS04-BP01 定義工作負載 KPI
- OPS04-BP02 實作應用程式遙測
- OPS04-BP03 實作使用者活動遙測
- OPS04-BP05 實作交易可追溯性
- OP08-BP04 建立可付諸行動的警示

相關文件：

- Amazon Personal AWS Health Dashboard 使用者指南
- AWS 網路監視器使用者指南
- AWS X-Ray 開發人員指南
- AWS DevOps Guru 使用者指南

相關影片：

- 深入了解影響應用程式效能的網際網路問題
- Amazon DevOps Guru 簡介
- 使用 AWS Health 以大規模管理資源生命週期事件

相關範例：

- 使用 Amazon DevOps Guru 獲得 AIOps 的營運洞見
- AWS Health Aware
- 使用標籤式篩選功能大規模管理 AWS Health 監控和警示

OPS04-BP05 實作分散式追蹤

分散式追蹤可讓您監控和以視覺化的方式了解，在分散式系統中各種來回移動元件的請求。透過從多個來源擷取追蹤資料並在統一的檢視中進行分析，團隊就能更了解請求的流程、瓶頸出現的位置，以及最佳化工作應著重的地方。

預期成果： 提供分散式系統請求流程的全面概覽，實現精確偵錯、最佳化效能，並改善使用者體驗。

常見的反模式：

- 不一致的檢測：並非所有分散式系統中的服務都經過檢測可進行追蹤。
- 忽略延遲：僅專注於錯誤，而未考慮延遲或效能逐漸降低的現象。

建立此最佳實務的優勢：

- 全方位的系統概觀：從進入到退出，徹底視覺化整個請求路徑。
- 強化偵錯：快速識別失敗或效能問題發生的位置。

- 改善使用者體驗：根據實際使用者資料進行監控與最佳化，確保系統符合實際需求。

未建立此最佳實務時的曝險等級： 高

實作指引

首先，識別工作負載中需要檢測的所有元素。將所有元件列入考量之後，就可以利用像是 AWS X-Ray 和 OpenTelemetry 等工具來收集追蹤資料，以便使用 X-Ray 和 Amazon CloudWatch ServiceLens Map 等工具進行分析。與開發人員一起進行定期檢閱，並在討論過程中利用 Amazon DevOps Guru、X-Ray Analytics 和 X-Ray Insights 等工具進行補充，以協助發掘更深入的調查結果。從追蹤資料建立警示，以便在工作負載監視計畫中定義的結果存在風險時發出通知。

實作步驟

若要有效實作分散式追蹤：

1. 採用 AWS X-Ray： 將 X-Ray 整合到您的應用程式中，以獲得深入其行為的洞見、了解效能，並且找出瓶頸的確切位置。利用 X-Ray Insights 進行自動化追蹤分析。
2. 檢測您的服務： 確認每一項服務 (從 AWS Lambda 函數到 EC2 執行個體)都會傳送追蹤資料。檢測的越多項服務，端對端檢視就越清楚。
3. 納入 CloudWatch 實際使用者監控 和 綜合監控： 將實際使用者監控 (RUM) 和綜合監控與 X-Ray 整合在一起。這樣就能擷取實際使用者體驗並模擬使用者互動，以從中找出潛在問題。
4. 使用 CloudWatch 代理程式： 代理程式可從 X-Ray 或 OpenTelemetry 傳送追蹤，進而獲得更深入的洞見。
5. 使用 Amazon DevOps Guru： DevOps Guru 使用來自 X-Ray、CloudWatch、AWS Config 和 AWS CloudTrail 的資料提供可付諸行動的建議。
6. 分析追蹤： 定期檢閱追蹤資料，以找出可能影響應用程式效能的模式、異常或瓶頸。
7. 設定警示： 在 CloudWatch 中設定警報來通報不尋常的模式或過久的延遲，以主動解決問題。
8. 持續改善： 隨著服務增加或修改重新檢視您的追蹤策略，以擷取所有相關資料點。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測

- OPS04-BP03 實作使用者體驗遙測
- OPS04-BP04 實作相依性遙測

相關文件：

- AWS X-Ray 開發人員指南
- Amazon CloudWatch 代理程式使用者指南
- Amazon DevOps Guru 使用者指南

相關影片：

- 使用 AWS X-Ray Insights
- AWS on Air ft.可觀測性：Amazon CloudWatch 和 AWS X-Ray

相關範例：

- 使用 AWS X-Ray 檢測您的應用程式

OPS 5.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，藉此推動重構、快速提供品質意見回饋及修復錯誤。這些方法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。

預期成果： 您的團隊共同撰寫程式碼。合併後，程式碼會是一致的，且變更不會遺失。透過正確的版本控制就能輕鬆復原錯誤。

常見的反模式：

- 您已在工作站上開發和儲存程式碼。您的工作站發生無法復原的儲存錯誤，造成程式碼遺失。
- 變更覆寫現有的程式碼之後，您重新啟動應用程式卻無法運作。您無法還原變更。
- 您對其他人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。某人意外地將自己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和舊版本，並有效降低資產遺失的風險。

未建立此最佳實務時的曝險等級： 高

實作指引

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。

資源

相關的最佳實務：

- OPS05-BP04 使用建置和部署管理系統

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

所部署的每項變更都必須經過測試，以避免在生產環境中發生錯誤。此一最佳實務著重於各種變更 (從版本控制到成品組建) 的測試。除了應用程式的程式碼變更以外，測試也應包含基礎設施、組態、安全控制和操作程序。測試採取多種形式，從單元測試到軟體元件分析 (SCA) 都包括在內。將測試進一步納入軟體整合和交付程序中，可進一步確保成品的品質。

您的組織必須制定所有軟體成品的測試標準。自動化測試可節省人力並避免手動測試錯誤。在某些情況下可能需進行手動測試。開發人員必須有權存取自動化測試結果，以建立可改善軟體品質的回饋迴圈。

期望的結果：您的軟體變更在交付前都經過測試。開發人員有權存取測試結果和驗證。您的組織具有適用於所有軟體變更的測試標準。

常見的反模式：

- 您在部署新軟體變更實未進行任何測試。軟體在生產環境中無法執行，因而導致中斷。
- 新的安全群組使用 AWS CloudFormation 進行部署，而未在生產前環境中測試。安全群組使您的客戶無法連線到應用程式。
- 方法已經過修改，但沒有單元測試。軟體部署至生產環境時失敗。

建立此最佳實務的優勢：降低軟體部署變更失敗率。軟體品質獲得改善。開發人員對於程式碼的可行性感知能力提高。可以安心推出安全政策，以支援組織的合規性。基礎設施變更 (例如自動化擴展政策更新) 會事先經過測試，以符合流量需求。

未建立此最佳實務時的風險暴露等級：高

實作指引

在持續整合的實務過程中，會對所有變更執行測試，從應用程式碼到基礎設施都包含在內。會發佈測試結果，讓開發人員迅速獲得反饋。您的組織具有所有變更都必須通過的測試標準。

搭配 Amazon Q Developer 使用生成式 AI 的力量來提高開發人員的生產力和程式碼品質。Amazon Q Developer 包含產生程式碼建議 (以大型語言模型為基礎)、生產單元測試 (包含邊界條件)，以及透過偵測和修復安全漏洞增強程式碼安全性功能。

客戶範例

在其持續整合管道中，AnyCompany Retail 對所有軟體成品執行了數種類型的測試。他們實行了測試驅動的開發，因此所有軟體都有測試單元。在成品建置後，他們執行了端對端測試。這個第一輪測試完成後，他們執行了靜態應用程式安全掃描，以尋找已知漏洞。開發人員在每個測試門檻通過後均收到訊息。所有測試都完成後，軟體成品即儲存在成品儲存庫中。

實作步驟

1. 與組織中的利害關係人合作制定軟體成品的測試標準。所有成品均應通過的標準測試為何？ 是否有必須納入測試涵蓋範圍內的合規或管控要求？ 您是否需要執行程式碼品質測試？ 測試完成時，誰需要得知？
    1. AWS 開發管道參考架構包含可在整合管道中對軟體成品執行之測試類型的授權清單。
2. 根據您的軟體測試標準，以必要的測試檢測您的應用程式。每一組測試均應在十分鐘內完成。測試應執行為整合管道的一部分。
    a. 使用 Amazon Q Developer 這款生成式 AI 工具，有助於建立單元測試案例 (包括邊界條件)、使用程式碼和註解產生函數，以及實作已知的演算法。
    b. 使用 Amazon CodeGuru Reviewer 測試您的應用程式碼是否有缺陷。
    c. 您可以使用 AWS CodeBuild 對軟體成品執行測試。
    d. AWS CodePipeline 可將您的軟體測試安排到管道中。

資源

相關的最佳實務：

- OPS05-BP01 使用版本控制
- 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-BP10 完全自動化整合和部署

相關文件：

- 採用測試驅動的開發方法
- 使用 Amazon Q 加速您的軟體開發生命週期
- 現在普遍可用的 Amazon Q Developer 包括新功能的預覽，可用於重塑開發人員體驗
- 在您的 IDE 中使用 Amazon Q Developer 的終極速查表
- 左移工作負載，利用 AI 建立測試

- Amazon Q Developer 中心
- 使用 Amazon CodeWhisperer 的 10 種更快速組建應用程式的方法
- 使用 Amazon CodeWhisperer 超越程式碼覆蓋範圍
- 使用 Amazon CodeWhisperer 的提示詞工程最佳實務
- 使用 TaskCat 和 CodePipeline 的自動化 AWS CloudFormation 測試管道
- 使用開放原始碼 SCA、SAST 和 DAST 工具建置端對端 AWS DevSecOps CI/CD 管道
- 開始測試無伺服器應用程式
- CI/CD 管道是我的發行隊長
- 在 AWS 上實行持續整合和持續交付白皮書

相關影片：

- 使用適用於軟體開發的 Amazon Q Developer 代理程式實作 API
- 透過 JetBrains IDE 安裝、設定和使用 Amazon Q Developer (操作方法)
- 熟悉 Amazon CodeWhisperer 的藝術 - YouTube 播放清單
- AWS re:Invent 2020：可測試的基礎設施：對 AWS 的整合測試
- AWS Summit ANZ 2021 - 透過 CDK 和測試驅動的開發施行測試優先策略
- 使用 AWS CDK 測試基礎設施即程式碼

相關資源：

- 使用生成式 AI 和 Amazon CodeWhisperer 組建應用程式
- Amazon CodeWhisperer 研討會
- AWS 部署管道參考架構 - 應用程式
- AWS Kubernetes DevSecOps 管道
- 政策即程式碼研討會 – 測試驅動的開發
- 使用 AWS CodeBuild 為 GitHub 中的 Node.js 應用程式執行單元測試
- 使用 Serverspec 進行基礎設施程式碼的測試驅動開發

相關服務：

- Amazon Q Developer

- Amazon CodeGuru Reviewer
- AWS CodeBuild
- AWS CodePipeline

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。

靜態組態管理會在初始化資源時設定值，這些值預期會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 服務的組態 (在 AWS Management Console 內) 或透過 AWS CLI。

動態組態管理會在初始化時設定值，這些值可能或是預期會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，透過組態變更啟動程式碼中的功能，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，藉此消除目前不需要的日誌及相關費用。

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。它可協助您追蹤其組態歷史記錄、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進行稽核，方法是使用 AWS Config 規則 和 AWS Config 合規套件。

如果您在 Amazon EC2 執行個體、AWS Lambda、容器、行動應用程式或 IoT 裝置上執行的應用程式中具有動態組態，則可以使用 AWS AppConfig 在您的環境中設定、驗證、部署和監控這些組態。

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

預期成果： 您會在持續整合、持續交付 (CI/CD) 管道中進行設定、驗證和部署。您會進行監控，以確認組態正確無誤。這會將終端使用者和客戶受到的任何負面影響降到最低。

常見的反模式：

- 您手動更新整個機群的 Web 伺服器組態，但由於更新錯誤，導致多部伺服器無法回應。
- 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。若不知道進行了哪些變更，您就需要花大量時間來調查問題，復原時間也會跟著拉長。
- 您可以透過 CI/CD 將生產前組態推送到生產環境中，而不需進行驗證。您讓使用者和客戶面臨使用不正確的資料和服務。

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。組態管理系統提供了管控、合規和法規需求方面的保證。

未建立此最佳實務時的曝險等級： 中

實作指引

組態管理系統可用來追蹤和實作應用程式與環境組態的變更。組態管理系統也可用來減少手動程序所造成的錯誤、讓組態變更可重複且可稽核，以及減少工作量。

實作步驟

1. 確定組態擁有者。
   a. 讓組態擁有者得知任何合規、管控或法規需求。
2. 確定組態項目與交付成果。
   a. 組態項目是指受到 CI/CD 管線內部署影響的所有應用程式和環境組態。
   b. 交付成果包括成功條件、驗證及監控對象。
3. 請根據您的業務需求和交付管道選取工具來進行組態管理。
4. 請考慮針對重大組態變更進行加權部署 (例如金絲雀部署)，以盡量減少錯誤組態造成的影響。
5. 將組態管理整合到 CI/CD 管道中。
6. 驗證所有推送的變更。

資源

相關的最佳實務：

- OPS06-BP01 為失敗變更進行規劃
- OPS06-BP02 測試部署
- OPS06-BP03 採用安全的部署策略
- OPS06-BP04 自動化測試和復原

相關文件：

- AWS Control Tower
- AWS 登陸區域加速器
- AWS Config

- 什麼是 AWS Config？
- AWS AppConfig
- 什麼是 AWS CloudFormation？
- AWS 開發人員工具

相關影片：

- AWS re:Invent 2022 - AWS 工作負載的主動管控與合規
- AWS re:Invent 2020：使用 AWS Config 實現合規即程式碼
- 使用 AWS AppConfig 管理和部署應用程式組態

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) 來建置持續整合/持續部署 (CI/CD) 管道。

預期成果： 您的建置和部署管理系統可支援組織的持續整合持續交付 (CI/CD) 系統，提供了使用正確組態自動化安全推展的功能。

常見的反模式：

- 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，卻無法啟動。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，並提供程式碼以進行品質保證 (QA)。它未通過 QA，因為缺少靜態資產。
- 週五，在經過一番努力之後，您成功在開發環境中手動建置應用程式，包括新編碼的功能。到了週一，您卻無法重複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，必須重新開發及測試。

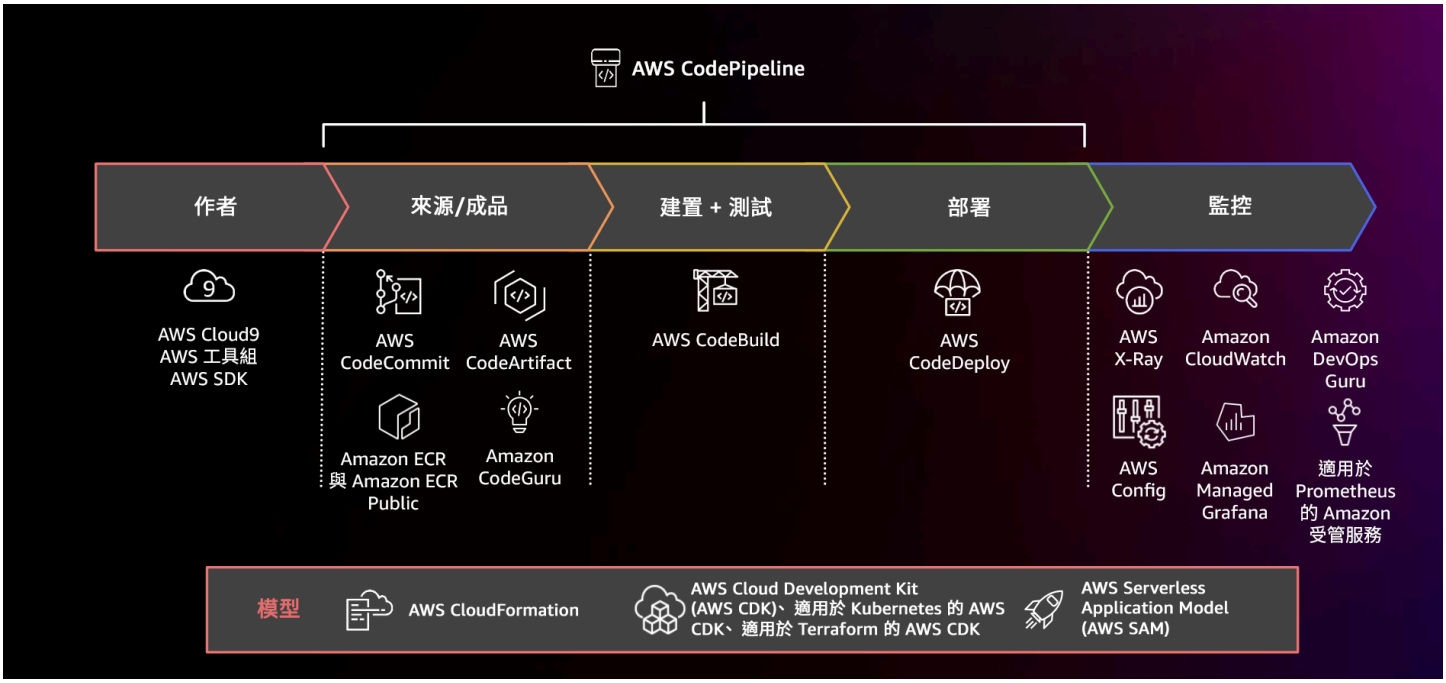
建立此最佳實務的優勢： 透過提供用於管理建置和部署活動的機制，您可以減少執行重複性任務的工作量，讓團隊成員專注於高價值的創意任務，並減少手動程序導致的錯誤。

未建立此最佳實務時的曝險等級： 中

實作指引

建置和部署管理系統可用來追蹤和實作變更、減少手動程序導致的錯誤，以及減少安全部署所需的工作量。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。此舉可縮短前置時間、降低成本、促進增加變更頻率、減少工作量，並且增進協作。

實作步驟



圖中顯示使用 AWS CodePipeline 和相關服務的 CI/CD 管道

1. 使用 AWS CodeCommit 進行版本控制、儲存及管理資產 (例如文件、原始程式碼和二進位檔案)。
2. 使用 CodeBuild 編譯原始程式碼、執行單元測試，以及產生立即可部署的成品。
3. 使用 CodeDeploy 做為部署服務，將應用程式自動部署至 Amazon EC2 執行個體、內部部署執行個體、無伺服器 AWS Lambda 函數或 Amazon ECS。
4. 監控您的部署。

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

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

相關影片：

- AWS re:Invent 2022 - 適用 AWS 上 DevOps 的 AWS Well-Architected 最佳實務

OPS05-BP05 執行修補程式管理

執行修補程式管理以取得功能、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、進行擴展，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。如果這種方法不可行，剩下的方法就是進行修補。

Amazon EC2 Image Builder 提供更新機器映像的管道。在修補程式管理的過程中，請考慮 Amazon Machine Image (AMI) (使用 AMI 影像管道)， 或容器映像 (使用 Docker 映像管道)，同時 AWS Lambda 會提供模式 讓自訂執行階段和其他程式庫 移除漏洞。

您應使用下列工具管理 Amazon Machine Image for Linux 或 Windows 伺服器映像的更新： Amazon EC2 Image Builder。您可以使用 Amazon Elastic Container Registry (Amazon ECR) 搭配現有的管道來管理 Amazon ECS 映像和管理 Amazon EKS 映像。Lambda 包括 版本管理功能。

若未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，以及使用下列工具來排程活動： Systems Manager 維護時段。

預期成果： 您的 AMI 和容器映像已完成修補、處於最新狀態，並準備好啟動。您可以追蹤所有已部署映像的狀態，並了解修補程式的合規狀況。您可以通報目前狀態，並設立程序來滿足合規需求。

常見的反模式：

- 您必須在兩小時內套用所有新的安全修補程式，結果導致應用程式與修補程式不相容而發生多次停機。
- 未修補的程式庫導致意外後果發生，因為有不明對象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次投訴，表示其環境如預期停止運作。
- 您尚未在持續執行的執行個體上修補商用現成軟體。當軟體發生問題而您聯絡廠商時，他們會通知您不支援該版本，您必須修補至特定程度才能獲得協助。
- 您使用的加密軟體近期發佈了修補程式，使效能獲得大幅改善。未修補的系統因未修補仍存在效能問題。
- 收到發生零時差漏洞的通知時，需緊急修正並手動修補所有環境。

建立此最佳實務的優勢： 透過建立修補程式管理程序 (包括修補準則和在各環境中散佈的方法)，您就能擴展和報告修補程度。這樣可保證修補過程安全無虞，並確保能清楚看見已知修正的狀態。如此可促進採用所需的功能、迅速消除問題，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。

未建立此最佳實務時的曝險等級： 中

實作指引

修補系統以補救問題，獲得所需的功能，並保持符合管控政策和廠商支援需求。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、避免手動程序引起的錯誤，並減少修補工作量。

實作步驟

針對 Amazon EC2 Image Builder：

1. 使用 Amazon EC2 Image Builder 指定管道詳細資訊：
    a. 建立映像管道並命名
    b. 定義管道排程和時區
    c. 設定任何相依性
2. 選擇配方：
    a. 選取現有配方或建立新配方
    b. 選取映像類型
    c. 提供配方的名稱和版本
    d. 選取基礎映像
    e. 新增組建元件並新增至目標登錄檔
3. 選用 - 定義您的基礎設施組態。

4. 選用 - 定義組態設定。
5. 檢閱設定。
6. 定期維護配方乾淨度。

針對 Systems Manager Patch Manager：

1. 建立修補基準。
2. 選取路徑操作方法。
3. 啟用合規報告和掃描。

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

- 什麼是 Amazon EC2 Image Builder
- 使用 Amazon EC2 Image Builder 建立映像管道
- 建立容器映像管道
- AWS Systems Manager Patch Manager
- 使用 Patch Manager
- 使用修補程式合規報告
- AWS 開發人員工具

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
- AWS Systems Manager Patch Manager 教學課程

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。記載它們並且隨著您的架構演進讓它們保持在最新狀態。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。如果沒有此選項，標準就會限制創新。

預期成果： 設計標準在貴組織的團隊之間共用。系統會記錄標準並且隨著最佳實務演進保持最新狀態。

常見的反模式：

- 兩個開發團隊各自建立了使用者身分驗證服務。您的使用者必須針對要存取的系統的每一部分，維護一組單獨的憑證。
- 每個團隊管理他們自己的基礎設施。新的合規要求會強制變更您的基礎設施，每個團隊會以不同的方式實作。

建立此最佳實務的優勢： 以共用的標準支援來實踐最佳實務，讓開發工作量發揮最大效益。記錄並且更新設計標準，讓貴組織的最佳實務和安全與合規要求保持在最新狀態。

未建立此最佳實務時的曝險等級： 中

實作指引

在團隊之間共用現有的最佳實務、設計標準、檢查清單、操作程序以及指導和管控要求。對於請求對設計標準進行變更、新增和例外設立程序，以支援改進和創新。讓團隊得知發佈的內容。設立機制讓設計標準隨著最佳實務發展而保持在最新狀態。

客戶範例

AnyCompany Retail 有跨部門架構團隊，該團隊會建立軟體架構模式。這個團隊會建置具有內建合規和管控的架構。採用這些共用標準的團隊會獲得具有內建合規和管控的優點。他們可以快速地在設計標準的基礎上建置。架構團隊每季開會一次，評估架構模式並且視需要更新。

實作步驟

1. 識別擁有開發和更新設計標準的跨部門團隊。這個團隊應與整個組織的利害關係人合作，共同開發設計標準、操作程序、檢查清單、指引和管控需求。記錄設計標準並且在組織內共用。
   a. AWS Service Catalog 可以用來建立套裝服務，代表使用基礎設施即程式碼的設計標準。您可以與所有帳戶共用套裝服務。
2. 設立機制讓設計標準隨著新的最佳實務出現而保持在最新狀態。

3. 如果設計標準是集中強制執行，設立程序來請求變更、更新和豁免。

實作計劃的工作量： 中。開發程序來建立和共用設計標準，即可與整個組織的利害關係人協調和合作。

資源

相關的最佳實務：

- OPS01-BP03 評估管控要求 - 管控需求會影響設計標準。
- OPS01-BP04 評估合規要求 - 合規是建立設計標準中的重要輸入。
- OPS07-BP02 確保對營運準備度進行一致的審查 - 營運準備度檢查清單是在設計您的工作負載時實作設計標準的機制。
- OPS11-BP01 建立持續改進程序 - 更新設計標準是持續改善的一部分。
- OPS11-BP04 執行知識管理 - 在您的知識管理實務中，記錄和共用設計標準。

相關文件：

- 使用 AWS Service Catalog 自動化 AWS Backup
- AWS Service Catalog Account Factory 增強
- Expedia Group 如何使用 AWS Service Catalog 建置資料庫即服務 (DBaaS) 方案
- 維護使用雲端架構模式的可見性
- 簡化在 AWS Organizations 設定中共用您的 AWS Service Catalog 套裝服務

相關影片：

- AWS Service Catalog – 入門
- AWS re:Invent 2020：像專家一樣管理您的 AWS Service Catalog 套裝服務

相關範例：

- AWS Service Catalog 參考架構
- AWS Service Catalog 研討會

相關服務：

- AWS Service Catalog

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、標準採用和配對程式設計。將這些實務併入您的持續整合和交付程序。

期望的結果：貴組織使用例如程式碼檢閱或配對程式設計的最佳實務來改善程式碼品質。開發人員和操作人員在軟體開發生命週期過程中採用程式碼品質最佳實務。

常見的反模式：

- 您將程式碼遞交至應用程式的主要分支，而未進行程式碼檢閱。變更會自動部署到生產並且造成中斷。
- 新的應用程式在沒有任何單位、端對端或整合測試的情況下進行開發。無法在部署之前測試應用程式。
- 您的團隊在生產中進行手動變更以解決缺陷。變更不會經過測試或程式碼檢閱，而且不會在持續整合或交付程序中擷取或記錄。

建立此最佳實務的優勢：透過採用實務來提高程式碼品質，就能協助盡量減少生產環境中引發的問題。程式碼品質有助於最佳實務的使用，例如配對程式設計、程式碼審查，以及 AI 生產力工具的實作。

未建立此最佳實務時的風險暴露等級：中

實作指引

實作實務以提高程式碼品質，在程式碼部署之前將缺陷降至最低。使用像是測試驅動的開發、程式碼檢閱和配對程式設計等實務來提高開發的品質。

搭配 Amazon Q Developer 使用生成式 AI 的力量來提高開發人員的生產力和程式碼品質。Amazon Q Developer 包含產生程式碼建議 (以大型語言模型為基礎)、生產單元測試 (包含邊界條件)，以及透過偵測和修復安全漏洞增強程式碼安全性功能。

客戶範例

AnyCompany Retail 採用數個實務來改善程式碼品質。他們已採用測試驅動開發做為撰寫應用程式的標準。對於某些新功能，他們會讓開發人員在衝刺期間一起進行配對程式設計。每個提取請求都會先經過資深開發人員的程式碼檢閱，然後再整合和部署。

實作步驟

1. 在您的持續整合和交付程序中，採用像是測試驅動的開發、程式碼檢閱和配對程式設計等程式碼品質實務。使用這些技術來改善軟體品質。
   a. 使用 Amazon Q Developer 這一款生成式 AI 工具，可協助建立單元測試案例 (包括邊界條件)、使用程式碼和註釋產生函數、實作已知的演算法、偵測程式碼中的安全政策違規和漏洞、偵測機密、掃描基礎設施即程式碼 (IaC)、文件程式碼，以及更快速學習第三方程式碼庫。
   b. Amazon CodeGuru Reviewer 可以提供讓 Java 和 Python 程式碼使用機器學習的程式設計建議。
   c. 您可以使用 AWS Cloud9 來建立共用開發環境，在其中合作開發程式碼。

實作計畫的工作量：中。有許多方式可以實作此最佳實務，但是組織採用可能會是一項挑戰。

資源

相關的最佳實務：

- OPS05-BP02 測試並驗證變更
- OPS05-BP06 共用設計標準

相關文件：

- 採用測試驅動的開發方法
- 使用 Amazon Q 加速您的軟體開發生命週期
- 現在普遍可用的 Amazon Q Developer 包括新功能的預覽，可用於重塑開發人員體驗
- 在您的 IDE 中使用 Amazon Q Developer 的終極速查表
- 左移工作負載，利用 AI 建立測試
- Amazon Q Developer 中心
- 使用 Amazon CodeWhisperer 的 10 種更快速組建應用程式的方法
- 使用 Amazon CodeWhisperer 超越程式碼覆蓋範圍
- 使用 Amazon CodeWhisperer 的提示詞工程最佳實務
- Agile 軟體指南
- CI/CD 管道是我的發行隊長
- 使用 Amazon CodeGuru Reviewer 自動化程式碼審查

- 採用測試驅動的開發方法
- DevFactory 如何使用 Amazon CodeGuru 組建更好的應用程式
- 關於配對程式設計
- RENGA Inc. 使用 Amazon CodeGuru 自動進行程式碼審查
- 敏捷開發的藝術：測試驅動的開發
- 程式碼檢閱為何重要 (而且確實可節省時間！)

相關影片：

- 使用適用於軟體開發的 Amazon Q Developer 代理程式實作 API
- 透過 JetBrains IDE 安裝、設定和使用 Amazon Q Developer (操作方法)
- 熟悉 Amazon CodeWhisperer 的藝術 - YouTube 播放清單
- AWS re:Invent 2020：使用 Amazon CodeGuru 持續改善程式碼品質
- AWS Summit ANZ 2021 - 透過 CDK 和測試驅動的開發施行測試優先策略

相關服務：

- Amazon Q Developer
- Amazon CodeGuru Reviewer
- Amazon CodeGuru Profiler
- AWS Cloud9

OPS05-BP08 使用多個環境

使用多個環境來試驗、開發和測試您的工作負載。當環境接近生產環境時提高控制層級，以確保您的工作負載在部署後依預期執行。

預期成果： 您有多個環境可反映您的合規和管控需求。您透過環境測試並推廣程式碼，以逐步實現生產環境。

常見的反模式：

- 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，您試圖重建導致資料遺失的條件，以便了解此情況如何發生，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。
- 您不一定會進行測試，但要測試時，您會在生產環境中進行。
- 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： 您可以支援多個同時開發、測試和生產的環境，而不會在開發人員或使用者社群之間產生衝突。

未建立此最佳實務時的曝險等級： 中

實作指引

使用多個環境，並且對開發人員沙盒環境實施最低限度的控制，以協助實驗。提供多個單獨的開發環境，以協助實現並行工作，進而提高開發敏捷性。在環境逐漸達到生產環境的條件時，實施更嚴格的控制，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來部署所設定控制條件與生產環境一致的環境，以確保系統在部署後依預期執行。當不使用環境時，關閉環境以避免產生與閒置資源相關的成本 (例如，在夜間和週末關閉開發系統)。進行負載測試時，部署與生產環境同等的環境，以改善有效的結果。

資源

相關文件：

- AWS 上的 Instance Scheduler
- 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。與變更管理系統、組態管理系統以及建置與交付系統搭配使用時，頻繁、細微和可逆的變更可縮小變更的範圍和影響。透過回復變更，可以更有效地進行疑難排解並加快修復速度。

常見的反模式：

- 您每季部署應用程式的新版本，這表示在這段變更期間，核心服務為關閉狀態。
- 您經常對資料庫結構描述進行變更，但未在您的管理系統中追蹤變更。
- 您執行手動就地更新，並覆寫現有的安裝和組態，但沒有明確的回復計畫。

建立此最佳實務的優勢： 透過經常部署小幅度的變更，加快了開發工作的速度。若變更幅度很小，就更容易了解變更是否會產生意外的後果，也更容易回復。如果變更可逆，由於復原過程較單純，因此實作變更的風險也會降低。變更程序的風險降低，而且變更失敗的影響也會降低。

未建立此最佳實務時的曝險等級： 低

實作指引

透過頻繁、細微和可逆的變更來縮小變更的範圍和影響。這樣可以簡化疑難排解，有助於加速修復，並提供回復變更的選項。另外還可以提高您為企業帶來價值的速度。

資源

相關的最佳實務：

- OPS05-BP03 使用組態管理系統
- OPS05-BP04 使用建置和部署管理系統
- OPS06-BP04 自動化測試和復原

相關文件：

- 實作 AWS 上的微型服務
- 微型服務 - 可觀測性

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， 以協助識別您的資源。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。

預期成果： 開發人員使用工具交付程式碼並推廣至生產環境。開發人員不必登入 AWS Management Console 就可以交付更新。有完整的變更與組態稽核記錄，可滿足管控和合規的需求。程序可在各團隊重複執行並且標準化。開發人員可全心專注於開發和程式碼推送，從而提高生產力。

常見的反模式：

- 週五，您完成了為功能分支編寫新程式碼。週一，執行程式碼品質測試指令碼和每個單位測試指令碼之後，請為下一排程版本檢查程式碼。

- 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。
- 您以開發人員身分登入 AWS Management Console，以使用非標準方法和系統來建立新的開發環境。

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，協助您的團隊成員專注於提供商業價值。您在推廣至生產環境的同時，加快了交付速度。

未建立此最佳實務時的曝險等級： 低

實作指引

您使用建置和部署管理系統來追蹤和實作變更，以減少由手動程序引起的錯誤，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。此舉可縮短前置時間、促進增加變更頻率、減少工作量、加快上市速度、提高生產力，並且在您推廣到生產環境時提高程式碼的安全性。

資源

相關的最佳實務：

- OPS05-BP03 使用組態管理系統
- OPS05-BP04 使用建置和部署管理系統

相關文件：

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

相關影片：

- AWS re\:Invent 2022 - 適用 AWS 上 DevOps 的 AWS Well-Architected 最佳實務

OPS 6.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 測試部署
- OPS06-BP03 採用安全的部署策略
- OPS06-BP04 自動化測試和復原

OPS06-BP01 為失敗變更進行規劃

計劃在部署造成非預期成果時恢復到已知的良好狀態，或者在生產環境中進行修復。擁有制定這類計畫的政策可以協助所有團隊訂立政策，從失敗變更中恢復。一些範例策略包括部署和回復步驟、變更政策、功能旗標、流量隔離和流量轉移。單一版本可能包含多個相關元件變更。策略要能提供您承受或從任何失敗元件變更中恢復的能力。

預期成果： 您已經為失敗變更準備了周延的恢復計畫。此外，您也縮減了發行版本的大小，如此一來，對其他工作負載元件的潛在影響將降到最低。因此，您可以縮短因變更失敗而造成的可能停機時間，並提高回復時間的彈性和效率，進而降低對業務的影響。

常見的反模式：

- 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。
- 您的系統架構並不允許以較小版本進行更新。因此，在部署失敗期間，您無法回復這些大量變更。
- 您未使用基礎架構即程式碼 (IaC)，並且您以手動方式更新了基礎架構，而造成不理想的組態。您無法有效追蹤和還原手動變更。
- 由於您尚未測量部署的增加頻率，您的團隊不會想降低變更規模和改善每次變更的回復計畫，進而造成更高的風險和失敗率。
- 請不要測量因變更失敗而導致中斷的總持續時間。您的團隊無法排定優先順序並改善其部署程序和回復計畫的效能。

建立此最佳實務的優勢： 若制定失敗變更的回復計畫，可將平均復原時間 (MTTR) 降至最低，並減少業務影響。

未建立此最佳實務時的曝險等級： 高

實作指引

發行團隊採用的一致記錄政策和實務可讓組織規劃發生失敗變更時應採取的動作。該政策應允許在特定情況下向前修正。在任何情況下，向前修正或回復計畫都應該在部署到現場生產之前先經過詳細記錄和測試，以將回復變更所需的時間降到最低。

實作步驟

1. 記錄要求團隊有效計劃在特定期間內還原變更的政策。
   a. 政策應指明允許向前修正的情況。
   b. 要求所有參與者皆能存取記錄完善的回復計畫。
   c. 指定回復需求 (例如：發現部署未經授權的變更時)。
2. 分析工作負載每個元件相關所有變更的影響程度。
   a. 如果可重複的變更遵循強制執行變更政策的一致工作流程，則允許這些變更進行標準化、範本化和預先授權。
   b. 縮小變更規模以減少任何變更的潛在影響，進而降低回復時間和對業務的影響。
   c. 確保回復程序會將程式碼回復到已知的良好狀態，避免可能發生的意外。
3. 整合工具和工作流程，以程式設計方式執行政策。
4. 讓其他工作負載擁有者可以看到變更相關資料，以改善任何無法回復失敗變更的診斷速度。
   a. 使用可見的變更資料來衡量這項實務的成效，並識別出反覆改進方式。
5. 使用監控工具來驗證部署成敗，以加速回復的決策過程。
6. 測量失敗變更期間的中斷時間，以持續修正回復計畫。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

- AWS Builders Library | 確保部署期間的回復安全
- AWS 白皮書 | 雲端中的變更管理

相關影片：

- re:Invent 2019 | Amazon 的高可用部署方式

OPS06-BP02 測試部署

使用與生產環境相同的部署組態、安全控制、步驟和程序，在生產前測試發行程序。驗證所有部署的步驟均按照預期完成，例如檢查檔案、組態和服務。透過功能、整合和負載測試以及任何監控 (例如運作狀態檢查) 進一步測試所有變更。透過這些測試，您可以及早發現部署問題，有機會在生產前進行規劃和問題緩解。

您可以建立暫時的平行環境來測試每項變更。使用基礎設施即程式碼 (IaC) 來自動化測試環境的部署，協助減少涉及的工作量，並確保穩定性、一致性和更快的功能交付。

預期成果： 您的組織採用測試驅動的開發文化，其中包含測試部署。如此一來，便能確保團隊專注於交付商業價值，而非管理發行版本。團隊會及早找出部署風險，並訂定適當的緩解方案。

常見的反模式：

- 使用生產版本期間，因為未經測試的部署經常會導致問題，而需要疑難排解或升級處理。
- 您的版本包含更新現有資源的基礎設施即程式碼 (IaC)。您不確定 IaC 是否會成功執行，或對資源造成影響。
- 您為應用程式部署一個新功能。該功能無法按照您的預期運作，且在受影響的使用者回報之前無法預見問題。
- 您更新憑證。您不小心將憑證安裝到錯誤的元件，這些元件未被偵測並因為無法建立與網站的安全連線，而影響了網站訪客。

建立此最佳實務的優勢： 針對部署程序的生產前階段及其帶來的變更進行廣泛測試，將部署步驟對生產的潛在負面影響降到最低。這麼做能增加產品發行期間的信心，並盡可能減少操作支援，同時不影響交付變更的速度。

未建立此最佳實務時的曝險等級： 高

實作指引

測試部署程序與測試部署所產生的變更同樣重要。您可以在生產前環境中測試部署步驟，盡可能準確反映生產環境。諸如不完整或錯誤部署步驟，或者配置錯誤等常見問題都能在生產環境之前偵測。此外，您也可以測試回復步驟。

客戶範例

作為持續整合與持續交付 (CI/CD) 管道的一部分，AnyCompany Retail 在一個類似生產環境中，執行為客戶發行基礎設施和軟體更新所需的定義步驟。流程包含許多預先檢查程序，可以在部署之前偵測到資源偏移 (偵測 IaC 以外所執行的資源變更)，以及驗證 IaC 啟動時所採取的動作。這個程序會驗證部署步驟，例如確認特定檔案和組態已準備就緒，或服務處於執行狀態，並在向負載平衡器重新註冊之前，正確回應本機上的運作狀態檢查。此外，所有變更都標記了許多自動化測試，例如功能、安全性、迴歸、整合和負載測試。

實作步驟

1. 執行安裝前檢查，將生產前環境反映到生產環境。
   a. 使用 偏移偵測 來偵測 AWS CloudFormation 外部資源的變更時間。
   b. 使用 變更集 來確認堆疊變更的目的是否符合啟動變更集時 AWS CloudFormation 所採取的動作。
2. 這會觸發手動核准步驟 (AWS CodePipeline)，以授權生產前環境的部署。
3. 使用部署組態 (例如 AWS CodeDeploy AppSpec 檔案) 來定義部署和驗證步驟。
4. 在適用情況下，會整合 AWS CodeDeploy 和其他 AWS 服務 或 會整合 AWS CodeDeploy 和合作夥伴產品與服務。
5. 監控部署 時會使用 Amazon CloudWatch、AWS CloudTrail,和 Amazon SNS 事件通知。
6. 執行部署後自動化測試，包括功能、安全性、迴歸、整合和負載測試。
7. 故障排除 部署問題。
8. 成功驗證前述步驟後，應該會起始化手動核准工作流程，以授權部署到生產環境。

實作計劃的工作量： 高

資源

相關的最佳實務：

- OPS05-BP02 測試並驗證變更

相關文件：

- AWS 建置者資料中心 | 自動化安全、無人為介入的部署 | 測試部署
- AWS 白皮書 | 在 AWS 上實行持續整合和持續交付
- Apollo 的故事 - Amazon 部署引擎

- 在交付程式碼之前，如何在本機測試和偵錯 AWS CodeDeploy
- 整合網路連線能力測試和基礎設施部署

相關影片：

- re:Invent 2020 | 在 Amazon 測試軟體和系統

相關範例：

- 教學 | 具驗證測試的部署和 Amazon ECS 服務

OPS06-BP03 採用安全的部署策略

安全的生產上市可以控制正面變更的流程，目的是將這些變更對客戶造成的任何負面影響降到最低。安全控制提供檢查機制，以驗證預期結果，並限制變更所帶來的任何負面影響或部署失敗造成的影響範圍。安全上市可能包括諸如功能旗標、一體式、滾動式 (Canary 版本)、不可變、流量拆分和藍/綠部署等策略。

預期成果： 您的組織使用持續整合與持續交付 CI/CD 系統，提供自動化安全上市功能的能力。團隊必須使用適當的安全上市策略。

常見的反模式：

- 您一次性將失敗的變更部署至所有生產環境。因此，所有客戶會同時受影響。
- 同時部署到所有系統的負面影響必須以緊急版本因應。需要數天時間為所有客戶進行錯誤修正。
- 管理生產發行需要多個團隊的規畫和參與。這會限制您為客戶積極提供更新功能的能力。
- 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舊版本，這會延長回復時間。

建立此最佳實務的優勢： 自動化部署持續為客戶在上市速度與交付正面變更之間取得平衡。限制影響範圍可以預防損失慘重的部署失敗，並盡可能提高團隊有效回應故障的能力。

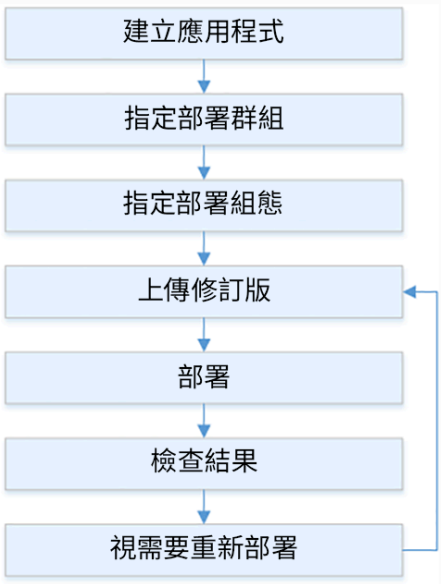
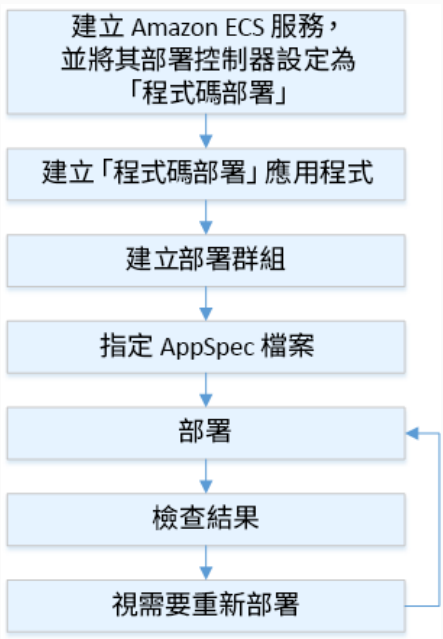
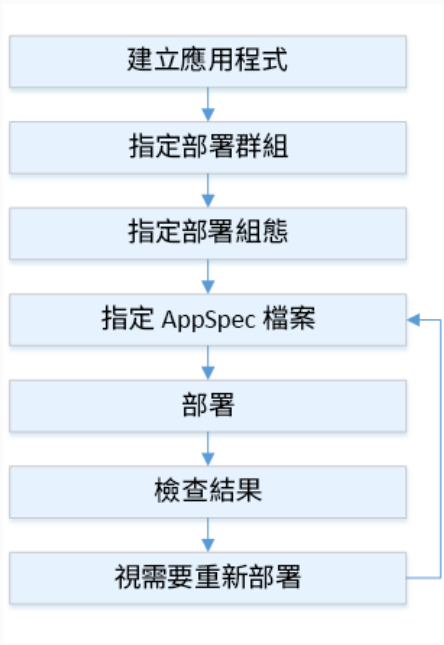
未建立此最佳實務時的曝險等級： 中

實作指引

持續交付的失敗可能導致服務可用性降低和糟糕的客戶體驗。為了盡可能提高部署成功率，請在端對端發行程序中實施安全控制，盡量減少部署錯誤，而目標則是實現零部署失敗。

客戶範例

AnyCompany Retail 的使命是實現最小至零停機時間的部署，這表示部署期間沒有對使用者產生任何可感知的負面影響。為了達成此目標，該企業已建立部署模式 (請參閱下方工作流程圖)，例如滾動部署和藍/綠部署。所有團隊都在其 CI/CD 管道中採用一個或多個模式。

Amazon EC2 的 CodeDeploy 工作流程	Amazon ECS 的 CodeDeploy 工作流程	Lambda 的 CodeDeploy 工作流程
		

實作步驟

1. 使用升級至生產環境的核准工作流程，觸發生產上市步驟的一系列動作。
2. 使用自動化部署系統，例如 AWS CodeDeploy。AWS CodeDeploy 部署選項 包含 EC2/內部部署的就地部署和藍/綠部署、AWS Lambda 以及 Amazon ECS (請參閱下方工作流程表)。
   a. 在適用情況下，會整合 AWS CodeDeploy 和其他 AWS 服務 或 會整合 AWS CodeDeploy 和合作夥伴產品與服務。
3. 針對資料庫使用藍/綠部署，例如 Amazon Aurora 和 Amazon RDS。
4. 監控部署 時會使用 Amazon CloudWatch、AWS CloudTrail 和 Amazon Simple Notification Service (Amazon SNS) 事件通知。
5. 執行部署後自動化測試，包括功能、安全性、迴歸、整合和任何負載測試。
6. 故障排除 部署問題。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS05-BP02 測試並驗證變更
- 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

相關文件：

- AWS 建置者資料中心 | 自動化安全、無人為介入的部署 | 生產部署
- AWS 建置者資料中心 | CI/CD 管道是我的 release captain | 安全、自動化生產版本
- AWS 白皮書 | 在 AWS 上實行持續整合和持續交付 | 部署方式
- AWS CodeDeploy 使用者指南
- 在 AWS CodeDeploy 中使用部署組態
- 設定 API Gateway 金絲雀版本部署
- Amazon ECS 部署類型
- Amazon Aurora 和 Amazon RDS 中的全受管藍/綠部署
- 使用 AWS Elastic Beanstalk 的藍/綠部署

相關影片：

- re:Invent 2020 | 無人為介入：在 Amazon 的持續交付管道
- re:Invent 2019 | Amazon 的高可用部署方式

相關範例：

- 在 AWS CodeDeploy 中嘗試範例藍/綠部署
- Worlshop | 使用 AWS CDK 為 Lambda 金絲雀部署建置 CI/CD 管道
- Workshop | EKS 和 ECS 的藍/綠和 Canary 部署
- Workshop | 建置跨帳戶 CI/CD 管道

OPS06-BP04 自動化測試和復原

為了提高部署程序的速度和可靠性，請在生產前和生產環境中制定自動化測試和回復功能的策略。在部署到生產環境時自動化測試，以模擬人類與系統的互動，驗證部署的變更。自動回復以快速回復到之前已知的良好狀態。回復應該在預先定義的條件下自動啟動，例如當未達到預期成果或自動化測試失敗時。將這兩項活動的自動化可以提高部署成功率，盡可能縮短回復時間，並減少對業務的潛在影響。

預期成果： 您的自動化測試和回復策略將整合至持續整合與持續交付 (CI/CD) 管道。您的監控能夠根據您的成功條件進行驗證，並在失敗時啟動自動回復。這會將終端使用者和客戶受到的任何負面影響降到最低。例如，當所有測試結果都達到標準時，您可以利用相同的測試案例，將程式碼提升至啟動自動迴歸測試的生產環境。若迴歸測試結果不符預期，則會在管線工作流程中啟動自動回復。

常見的反模式：

- 您的系統架構並不允許以較小版本進行更新。因此，在部署失敗期間，您無法回復這些大量變更。
- 您的部署程序包含一系列手動步驟。將變更部署到工作負載之後，即可開始部署後測試。測試之後，您會發現工作負載無法運作，且客戶中斷連線。然後您開始回復到之前的版本。所有這些手動步驟都會延遲整體系統回復，並對客戶造成長期影響。
- 您花時間為應用程式中不常使用的功能開發自動化測試案例，因而大幅降低了自動化測試功能的投資報酬率。
- 您的版本包含彼此獨立的應用程式、基礎設施、修補程式和組態更新。但是，您有一個 CI/CD 管道可以一次交付所有變更。一個元件故障會強迫您還原所有變更，進而使回復過程變得複雜且效率低下。
- 您的團隊在第一個衝刺階段完成編碼，並開始衝刺兩項工作，但直到第三個衝刺階段，計畫中都不包括測試。最終，自動化測試找出第一個衝刺階段的缺漏，必須在測試第二個衝刺階段前解決，才能啟動交付項目，因此整個版本延遲，進而降低您的自動化測試效率。
- 生產版本的自動迴歸測試案例已經完成，但您並未監控工作負載的運作狀況。由於無法查看是否已重啟服務，您不確定是否需要回復或已啟動回復。

建立此最佳實務的優勢： 自動化測試可以提高測試流程的透明度，以及您在更短時間內顧及更多功能的能力。在生產環境中測試和驗證變更，可以立即識別出問題。改善自動化測試工具的一致性可以更精確地偵測問題。透過自動回復至舊版本，將對客戶的影響降至最低。自動化回復最終可減少業務影響，讓您對部署功能更有信心。整體而言，這些功能可縮短交付時間，同時確保品質。

未建立此最佳實務時的曝險等級： 中

實作指引

自動測試已部署的環境，更快確認是否達到預期成果。當無法達成預先定義的結果時，自動還原到先前的良好狀態，以盡量縮短還原時間，並減少由手動程序引起的錯誤。將測試工具與管道工作流程整合，持續進行測試並減少手動輸入。優先處理自動化測試案例，例如減緩最高風險且需要在每次變更時經常測試的案例。此外，還可以根據測試計畫中預先定義的特定條件進行自動回復。

實作步驟

1. 為您的開發生命週期建立測試生命週期，定義需求規劃到測試案例開發、工具配置、自動化測試和測試案例結案等每個測試程序階段。
   a. 根據您的整體測試策略建立針對特定工作負載的測試方式。
   b. 在整個開發生命週期中，考慮適當的連續測試策略。
2. 根據您的業務需求和管道投資，選擇用於測試和回復的自動化工具。
3. 決定您應該分別自動化和手動執行哪些測試案例。這些內容皆可以根據受測功能的業務價值優先順序來決定。使所有團隊成員隨時接收計畫最新資訊，並確認執行手動測試的權責分配。
   a. 將自動化測試功能應用於對自動化有意義的特定測試案例，例如可重複或經常執行的案例、需要重複作業的案例，或跨多個組態所需的案例。
   b. 在自動化工具中定義測試自動化指令碼和成功條件，如此一來，當特定案例失敗時，可以啟動持續的工作流程自動化。
   c. 定義自動回復的特定失敗條件。
4. 測試案例其中複雜度和人工互動具較高的失敗風險，因此必須排定測試自動化的優先順序，透過詳盡的測試案例開發來產生一致的結果。
5. 將您的自動化測試和回復工具整合到 CI/CD 管道。
   a. 為變更制定明確的成功條件。
   b. 監控觀察以偵測這些條件，並在符合特定回復條件時自動回復變更。
6. 執行不同類型的自動化生產測試，例如：
   a. A/B 測試，以顯示結果比較兩個使用者測試組之間的當前版本。
   b. 金絲雀測試讓您能在將變更發佈給所有使用者之前，先將其發佈給一部分使用者。
   c. 功能旗標測試允許您從應用程序外部標記新版本的功能 (每次僅限一個)，進而使每個新功能皆能逐一進行驗證。
   d. 迴歸測試，以現有的關聯元件驗證新功能。
7. 透過其他應用程式和元件，監控應用程式、交易和互動的操作面向。開發報告，以便按照部銅工作負載顯示變更成功率，讓您得以識別出能夠進一步最佳化的自動化和工作流程部分。

a. 開發測試結果報告，協助您快速決定是否應該調用回復程序。

b. 實施策略，允許根據一個或多個測試方法導出的預定義失敗條件進行自動回復。

8. 開發自動化測試用例，以便在未來可重複的變更中重複使用。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS06-BP01 為失敗變更進行規劃
- OPS06-BP02 測試部署

相關文件：

- AWS Builders Library | 確保部署期間的回復安全
- 使用 AWS CodeDeploy 重新部署和回復部署
- 使用 AWS CloudFormation 自動化部署時的 8 個最佳實務

相關範例：

- 使用 Selenium、AWS Lambda、AWS Fargate (Fargate) 和 AWS 開發人員工具進行無伺服器 UI 測試

相關影片：

- re:Invent 2020 | 無人為介入：在 Amazon 的持續交付管道
- re:Invent 2019 | Amazon 的高可用部署方式

OPS 7.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-BP06 啟用生產工作負載的支援計劃

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支援工作負載。他們必須受過組成您的工作負載的平台和服務的培訓。為他們提供操作工作負載所需的知識。您必須擁有足夠受過培訓的人員，才能支援工作負載的一般操作，並且針對會發生的任何事件進行疑難排解。擁有足夠的人員，以便您可以輪替待命和休假的人員，避免倦怠。

預期成果：

- 有足夠受過培訓的人員可以在工作負載可用時支援工作負載。
- 您為人員提供組成您的工作負載的軟體和服務的培訓。

常見的反模式：

- 在沒有受過培訓操作使用中平台和服務的團隊成員之情況下，部署工作負載。
- 沒有足夠的人員可以支援待命輪替或人員休假。

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。
- 具有足夠的團隊成員，您可以支援工作負載和待命輪替，同時降低倦怠風險。

未建立此最佳實務時的風險暴露等級：高

實作指引

驗證人員是否已經過充分培訓，可支援工作負載。確認擁有足夠且訓練有素的團隊成員，以妥善應對一般營運活動，包括待命輪替。

客戶範例

AnyCompany Retail 確保支援工作負載的團隊有適當的配備人員且訓練有素。他們有足夠的工程師可以支援待命輪替。人員會獲得工作負載建置基礎的軟體和平台的培訓，並且鼓勵他們考取認證。有足夠的人員讓員工可以休假，同時仍然支援工作負載和待命輪替。

實作步驟

1. 指派適當數量的人員來操作和支援您的工作負載，包括隨時待命。
2. 為您的人員提供組成您的工作負載的軟體和平台的培訓。
   a. AWS 培訓與認證有 AWS 的相關課程庫。它們提供免費和付費課程，線上或面授。
   b. AWS 主持活動和研討會，您可以向 AWS 專家學習。
3. 定期隨著操作條件和工作負載變更，評估團隊大小和技能。調整團隊大小和技能以符合操作要求。

實作計劃的工作量：高。招聘和培訓團隊來支援工作負載需要大量的努力，但是會有重大的長期優點。

資源

相關的最佳實務：

- OPS11-BP04 執行知識管理 - 團隊成員必須擁有操作和支援工作負載所需的資訊。知識管理是提供這項能力的關鍵。

相關文件：

- AWS 活動和研討會
- AWS 培訓與認證

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

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期望的結果。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。

期望的結果：您的團隊有一系列執行工作負載任務的逐步指南。執行手冊中包含期望的結果、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 (版本控制系統)，並且經常更新。例如，您的執行手冊讓團隊具備可在應用程式警示、運作問題和規劃生命週期事件期間監控、傳達和回應重要帳戶之 AWS Health 事件的能力。

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風險暴露等級：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。期望的結果應明確指出。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。您可以使用 AWS Systems Manager 自動化等服務，將平面文字轉換為可以根據工作負載執行的自動化程序。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。AWS Systems Manager 自動化還提供低程式碼的 視覺設計體驗，以更輕鬆地建立自動化執行手冊。

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。

```
# Runbook Title
## Runbook Info
| Runbook ID | Description | Tools Used | Special Permissions | Runbook Author | Last
 Updated | Escalation POC |
|-------|-------|-------|-------|-------|-------|-------|
| RUN001 | What is this runbook for? What is the desired outcome? | Tools | Permissions
 | Your Name | 2022-09-21 | Escalation Name |
## 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。填寫「執行手冊標題」和「執行手冊資訊」下的必要欄位。
4. 從第一個步驟開始，填寫執行手冊的「步驟」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。

實作計畫的工作量：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 流程和程序已確認擁有者
- OPS07-BP04 使用程序手冊來調查問題
- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP02 每個提醒建立一個流程
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected 架構：概念：執行手冊研製
- 使用自動化的程序手冊和執行手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- 用於 AWS 大型遷移的遷移執行手冊 - 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
- 如何在 AWS 上將 IT 作業自動化 | Amazon Web Services
- 將指令碼整合到 AWS Systems Manager 中

相關範例：

- Well-Architected 實驗室：使用程序手冊和執行手冊將操作自動化
- AWS 部落格文章：建立雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- 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

相關服務：

- AWS Systems Manager 自動化

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畫的關鍵要素。

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。制定溝通計畫，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畫。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。

期望的結果：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做出貢獻。
- 團隊可以透過可重複的程序手冊擴展其程序。

未建立此最佳實務時的風險暴露等級：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 AWS Systems Manager 自動化等服務時，可以將平面文字轉換為自動化。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。

客戶可以使用 AWS Systems Manager Incident Manager 回應事件。此服務提供單一介面分類事件、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。此服務使用 AWS Systems Manager 自動化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「程序手冊名稱」部分和「程序手冊資訊」下的欄位。
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 自動化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。

實作計畫的工作量：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 流程和程序已確認擁有者
- OPS07-BP03 使用執行手冊執行程序
- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP02 每個提醒建立一個流程
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected 架構：概念：程序手冊開發研製

- 使用自動化的程序手冊和執行手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- 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)
- 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 事故應變執行手冊
- 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 自動化
- AWS Systems Manager Incident Manager

OPS07-BP05 做出部署系統和變更的明智決策

為成功和失敗變更工作負載建立程序。事前剖析是一種演練，團隊可藉此模擬失敗，開發緩解策略。使用事前剖析可預測失敗並適時建立程序。評估將變更部署到您的工作負載的優點和風險。確認所有變更都符合管控。

預期成果：

- 您在將變更部署到您的工作負載時做出明智決策。
- 變更符合管控。

常見的反模式：

- 將變更部署到我們的工作負載，而沒有處理失敗部署的程序。
- 對不符合管控要求的生產環境進行變更。
- 部署新版本的工作負載，而未建立資源使用率的基準。

建立此最佳實務的優勢：

- 您對工作負載的失敗變更已做好準備。
- 變更您的工作負載符合管控政策。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用事前剖析來開發失敗變更的程序。記載失敗變更的程序。確定所有變更都符合管控。評估將變更部署到您的工作負載的優點和風險。

客戶範例

AnyCompany Retail 定期執行事前剖析來驗證他們失敗變更的程序。他們在共用 Wiki 中記載程序並且頻繁更新。所有變更都符合管控要求。

實作步驟

1. 在將變更部署到您的工作負載時做出明智決策。建立及檢閱成功部署的準則。開發會觸發變更回復的情境或準則。權衡部署變更的優點與失敗變更的風險。
2. 確認所有變更都符合管控政策。
3. 使用事前剖析為失敗變更進行規劃並且記載緩解策略。執行桌上模擬演練來建立失敗變更的模型，並且驗證回復程序。

實作計劃的工作量：中。實作事前剖析的實務需要貴組織利害關係人的協調和努力

資源

相關的最佳實務：

- OPS01-BP03 評估管控要求 - 管控要求是判斷是否部署變更的關鍵因素。
- OPS06-BP01 為失敗變更進行規劃 - 建立計劃來緩解失敗的部署並且使用事前剖析來驗證它們。

- OPS06-BP02 測試部署 - 每個軟體變更都應該在部署之前先適當的進行測試，以便在生產中減少缺陷。
- OPS07-BP01 確保人員能力 - 擁有支援工作負載的足夠受過培訓的人員，對於為部署系統變更做出明智決策相當重要。

相關文件：

- Amazon Web Services：風險與合規
- AWS 共同責任模式
- AWS 雲端 中的管控：敏捷和安全之間的正確平衡。

OPS07-BP06 啟用生產工作負載的支援計劃

針對您的生產工作負載所依賴的任何軟體和服務啟用支援。根據您的生產服務層級需求選取適當的支援等級。必須要有這些相依性的支援計劃，以備發生服務中斷或軟體問題時使用。記錄支援計劃，以及如何要求所有服務和軟體供應商的支援。實作相關機制以確認支援的聯絡窗口是最新的。

預期成果：

- 為生產工作負載所依賴的軟體和服務實作支援計劃。
- 根據服務層級需求選擇適當的支援計劃。
- 記錄支援計劃、支援等級，以及如何要求支援。

常見的反模式：

- 您沒有主要軟體供應商的支援計劃。您的工作負載因此受到影響，且您無法加速進行修正，或及時獲得供應商提供的更新。
- 擔任軟體供應商主要聯絡窗口的開發人員已離開公司。您無法直接聯繫供應商支援人員。您必須花時間研究及瀏覽通用聯絡系統，因此必要時的回應將更為耗時。
- 軟體供應商發生生產中斷。目前沒有文件說明如何提出支援案例。

建立此最佳實務的優勢：

- 透過適當的支援等級，您將可在必要的時間範圍內獲得回應以滿足服務層級需求。
- 受支援的客戶可在遇到生產問題時加以呈報。
- 軟體和服務供應商可在事件發生期間協助進行疑難排解。

未建立此最佳實務時的風險暴露等級：低

實作指引

針對您的生產工作負載所依賴的任何軟體和服務供應商啟用支援計劃。設定適當的支援計劃以滿足服務層級需求。對 AWS 客戶而言，這意味著在任何有生產工作負載的帳戶上啟用 AWS Business Support (或更高等級)。定期與支援供應商聯繫，取得關於支援優惠、程序和聯絡人的更新。記錄如何要求軟體和服務供應商的支援，包括如何在中斷發生時加以呈報。實作相關機制以保有最新的支援聯絡資料。

客戶範例

在 AnyCompany Retail，所有商業軟體和服務相依性都有支援計劃。例如，他們在所有具有生產工作負載的帳戶上啟用了 AWS Enterprise Support。任何開發人員都可在問題發生時提出支援案例。有 Wiki 頁面提供了相關資訊說明如何要求支援、應通知誰，以及加速處理案例的最佳實務為何。

實作步驟

1. 與組織中的利害關係人合作，識別您的工作負載所依賴的軟體和服務供應商。記錄這些相依性。
2. 確認工作負載的服務層級需求。選取相對應的支援計劃。
3. 針對商業軟體和服務，與供應商共同建立支援計劃。
    a. 為所有生產帳戶訂閱 AWS Business Support 或更高等級可獲得 AWS Support 更快的回應時間，極力建議這麼做。如果您沒有付費支援，則必須有處理問題的行動計劃，而這需要 AWS Support 的協助。AWS Support 提供了多種工具和技術、人員和方案，旨在主動協助您優化效能、降低成本和加快創新速度。AWS Business Support 提供了額外權益，包括能夠存取 AWS Trusted Advisor 和 AWS Personal Health Dashboard，以及更快速的回應時間。
4. 在您的知識管理工具中記錄支援計劃。納入如何要求支援、在提出支援案例時應通知誰，以及在事件發生時如何加以呈報等資訊。任何人在得知支援程序或聯絡資料有所變更時，都可以利用 Wiki 這項機制對文件進行必要的更新。

實作計劃的工作量：低。大部分的軟體和服務供應商都提供選擇加入支援計劃。在您的知識管理系統上記錄並分享支援最佳實務，可確保您的團隊知道在生產問題發生時應如何因應。

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者

相關文件：

- AWS Support Plans

相關服務：

- AWS Business Support
- AWS Enterprise Support

營運

問題

- OPS 8.如何在組織中利用工作負載可觀測性？
- OPS 9.如何了解營運狀況？
- OPS 10.如何管理工作負載和營運事件？

OPS 8.如何在組織中利用工作負載可觀測性？

利用可觀測性確保最佳的工作負載運作狀況。利用相關指標、日誌和追蹤，全面掌握工作負載效能並有效解決問題。

最佳實務

- OPS08-BP01 分析工作負載指標
- OPS08-BP02 分析工作負載日誌
- OPS08-BP03 分析工作負載追蹤
- OPS08-BP04 建立可付諸行動的警示
- OPS08-BP05 建立儀表板

OPS08-BP01 分析工作負載指標

實作應用程式遙測之後，請定期分析收集到的指標。雖然延遲、請求、錯誤和容量 (或配額) 可提供深入了解系統效能的洞見，但務必將檢閱業務成果指標視為優先事項。這樣做可確保您所做的資料驅動決策符合您的業務目標。

預期成果： 獲得深入工作負載效能的精確洞見，有助於做出資料驅動的決策，確保與業務目標保持一致。

常見的反模式：

- 單獨分析指標，未能考慮到其對業務目標的影響。
- 過度依賴技術指標，而輕忽業務指標。
- 未能時常檢閱指標，而錯失即時決策的機會。

建立此最佳實務的優勢：

- 增進對於技術表現與業務成果之間相互關聯的了解。
- 透過即時資料改善了決策過程。
- 主動識別並緩解問題，不讓問題影響業務成果。

未建立此最佳實務時的曝險等級： 中

實作指引

利用像是 Amazon CloudWatch 等工具進行指標分析。AWS 服務 (如 AWS Cost Anomaly Detection 和 Amazon DevOps Guru) 可用來偵測異常狀況，特別是在靜態閾值未知，或行為模式更適合異常偵測的情況下。

實作步驟

1. 分析與檢閱： 定期檢閱和解讀您的工作負載指標。
   a. 將業務成果指標視為優先於純粹技術指標的事項。
   b. 了解資料中峰值、下降或模式的重要性。
2. 利用 Amazon CloudWatch： 使用 Amazon CloudWatch 集中檢視並進行深入分析。
   a. 設定 CloudWatch 儀表板以視覺化您的指標，並長時間進行比較。
   b. 在 CloudWatch 中使用百分位數 以清楚了解指標的分佈情形，這有助於定義 SLA 和了解極端值。
   c. 設定 AWS Cost Anomaly Detection 以識別不尋常的模式，而不依賴靜態閾值。
   d. 實作 CloudWatch 跨帳戶可觀測性 以監控跨區域內多個帳戶的應用程式並進行疑難排解。
   e. 使用 CloudWatch Metric Insights 查詢和分析跨帳戶和區域的指標資料，以找出趨勢和異常狀況。
   f. 套用 CloudWatch Metric Math 來轉換、彙總或對您的指標執行計算，以獲得更深入的洞見。
3. 採用 Amazon DevOps Guru： 納入 Amazon DevOps Guru 以利用其機器學習強化的異常偵測功能，識別無伺服器應用程式操作問題的早期跡象，並矯正問題以免影響客戶。
4. 根據洞見最佳化： 根據您的指標分析做出明智的決策，以調整和改善您的工作負載。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測

相關文件：

- The Wheel 部落格 - 強調持續檢閱指標的重要性
- 百分位數很重要
- 使用 AWS Cost Anomaly Detection
- CloudWatch 跨帳戶可觀測性
- 使用 CloudWatch Metrics Insights 查詢您的指標

相關影片：

- 在 Amazon CloudWatch 中啟用跨帳戶可觀測性
- Amazon DevOps Guru 簡介
- 使用 AWS Cost Anomaly Detection 持續分析指標

相關範例：

- One Observability 研討會
- 使用 Amazon DevOps Guru 獲得 AIOps 的運作洞見

OPS08-BP02 分析工作負載日誌

定期分析工作負載日誌相當重要，藉此能夠深入了解應用程式的各個操作層面。藉由有效率地篩選、視覺化和解讀日誌資料，可持續最佳化應用程式效能和安全。

期望的結果：從徹底的日誌分析中獲得深入應用程式行為和操作的豐富洞見，以確保主動偵測和緩解問題。

常見的反模式：

- 忽略日誌分析，直到出現嚴重問題。
- 未使用一套完整的工具進行日誌分析，而錯過了關鍵的洞見。
- 只倚賴手動檢閱日誌，而未利用自動化和查詢功能。

建立此最佳實務的優勢：

- 主動找出操作瓶頸、安全威脅及其他潛在問題。
- 有效利用日誌資料，以持續最佳化應用程式。
- 加強對應用程式行為的理解，幫助偵錯和疑難排解。

未建立此最佳實務時的風險暴露等級：中

實作指引

Amazon CloudWatch Logs 是強大的日誌分析工具。像是 CloudWatch Logs Insights 和 Contributor Insights 這類整合式功能，可提供簡單直接且有效率的方式從日誌中產生有意義的資訊。

實作步驟

1. 設定 CloudWatch Logs：應用程式和服務以將日誌傳送至 CloudWatch Logs。
2. 使用日誌異常偵測：利用 Amazon CloudWatch Logs 異常偵測自動識別不尋常日誌模式並予以警示。此工具可協助您主動管理日誌檔中的異常狀況，並及早偵測潛在的問題。
3. 設定 CloudWatch Logs Insights：使用 CloudWatch Logs Insights，以互動方式搜尋和分析日誌資料。
   a. 製作查詢以找出模式、視覺化日誌資料，並產生可付諸行動的洞見。
   b. 使用 CloudWatch Logs Insights 模式分析來分析和視覺化頻繁的日誌模式。此功能可協助您了解日誌資料中常見的運作趨勢和潛在異常值。
   c. 使用 CloudWatch Logs 比較 (diff) 在不同時間週期之間，或在不同日誌群組之間執行差異分析。使用此功能精確找出變更，並評估其對系統效能或行為的影響。
4. 使用 Live Tail 即時監控日誌：使用 Amazon CloudWatch Logs Live Tail 即時查看日誌資料。您可以主動監控應用程式的運作活動，從中了解系統效能和潛在問題。
5. 利用 Contributor Insights：使用 CloudWatch Contributor Insights 識別 IP 位址或使用者代理程式等高基數維度中的最高用量者。
6. 實作 CloudWatch Logs 指標篩選條件：設定 CloudWatch Logs 指標篩選條件，將日誌資料轉換為可操作的指標。如此您就能設定警報或進一步分析模式。

7. 實作 CloudWatch 跨帳戶可觀測性：監控和疑難排解區域內跨多個帳戶的應用程式。
8. 定期檢閱和改進：定期檢閱您的日誌分析策略，以擷取所有相關資訊並持續最佳化應用程式效能。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- OPS08-BP01 分析工作負載指標

相關文件：

- 使用 CloudWatch Logs Insights 分析日誌資料
- 使用 CloudWatch Contributor Insights
- 建立和管理 CloudWatch 日誌指標篩選條件

相關影片：

- 使用 CloudWatch Logs Insights 分析日誌資料
- 使用 CloudWatch Contributor Insights 分析高基數資料

相關範例：

- CloudWatch Logs 範例查詢
- One Observability 研討會

OPS08-BP03 分析工作負載追蹤

對於獲得應用程式操作之旅全面性的總覽來說，分析追蹤資料是相當重要的一環。透過視覺化和了解各種不同元件之間的互動，就能微調效能、找出瓶頸，並且增強使用者體驗。

期望的結果：清楚掌握應用程式的分散式操作，就能更快解決問題並增強使用者體驗。

常見的反模式：

- 忽略追蹤資料，只依賴日誌和指標。
- 未將追蹤資料與相關的日誌建立關聯。
- 忽略從追蹤產生的指標，如延遲和故障率。

建立此最佳實務的優勢：

- 改善疑難排解並縮短平均解決時間 (MTTR)。
- 獲得深入相依性及其影響的洞見。
- 快速找出並糾正效能問題。
- 利用追蹤產生的指標做出明智的決策。
- 透過最佳化元件互動改善使用者體驗。

未建立此最佳實務時的風險暴露等級：中

實作指引

AWS X-Ray 提供了全方位的追蹤資料分析套件，能讓您深入了解服務互動的各個層面、監控使用者活動，以及偵測效能問題。像是 ServiceLens、X-Ray Insights、X-Ray Analytics 及 Amazon DevOps Guru 等功能可從追蹤資料產生更深入且可付諸行動的洞見。

實作步驟

下列步驟提供了結構化的方法，以使用 AWS 服務有效實作追蹤資料分析：

1. 整合 AWS X-Ray：確保 X-Ray 與您的應用程式整合以擷取追蹤資料。
2. 分析 X-Ray 指標：使用服務地圖監控應用程式健全狀態，深入了解從 X-Ray 追蹤衍生的指標，例如延遲、請求率、故障率和回應時間分佈。
3. 使用 ServiceLens：利用服務地圖來增強服務和應用程式的可觀測性。如此就能將追蹤、指標、日誌、警報和其他運作狀況資訊整合在一起檢視。
4. 啟用 X-Ray Insights：
   a. 開啟 X-Ray Insights 以在追蹤中自動偵測異常狀況。
   b. 檢查洞見以找出明確的模式並確定根本原因，例如故障率或延遲增加。
   c. 請參考洞察時間軸，依時間順序查看所偵測到問題的分析。
5. 使用 X-Ray Analytics：X-Ray Analytics 可讓您徹底探索追蹤資料、找出明確的模式，以及擷取洞見。

6. 使用 X-Ray 中的群組：在 X-Ray 中建立群組，即可根據如高延遲等條件篩選追蹤，以進行更針對性的分析。
7. 併入 Amazon DevOps Guru：參與 Amazon DevOps Guru 以便利用機器學習模型的優勢，從追蹤中找出明確的操作異常狀況。
8. 使用 CloudWatch Synthetics：使用 CloudWatch Synthetics建立可持續監控端點和工作流程的 Canary。這些 Canary 可與 X-Ray 整合，以提供追蹤資料，用來對要測試的應用程式進行深入分析。
9. 使用真實使用者監控 (RUM)：使用 AWS X-Ray 和 CloudWatch RUM，您可以透過下游受 AWS 管理服務，從應用程式的最終使用者開始分析和除錯請求路徑。這樣做有助於找出影響最終使用者的延遲趨勢和錯誤。
10. 與日誌檔關聯：將 追蹤資料與 X-Ray 追蹤檢視中的相關日誌相關聯，以獲得應用程式行為的精細視角。如此可讓您檢視與追蹤的交易直接相關的日誌事件。
11. 實作 CloudWatch 跨帳戶可觀測性：監控和疑難排解區域內跨多個帳戶的應用程式。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS08-BP01 分析工作負載指標
- OPS08-BP02 分析工作負載日誌

相關文件：

- 使用 ServiceLens 監控應用程式運作狀況
- 使用 X-Ray Analytics 探索追蹤資料
- 使用 X-Ray Insights 偵測追蹤中的異常狀況
- 使用 CloudWatch Synthetics 持續監控

相關影片：

- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray分析應用程式並進行偵錯
- 使用 AWS X-Ray Insights

相關範例：

- One Observability 研討會
- 使用 AWS Lambda 實作 X-Ray
- CloudWatch Synthetics Canary 範本

OPS08-BP04 建立可付諸行動的警示

及時偵測並回應應用程式行為偏差的情況，是相當重要的一環。尤其重要的是，能夠辨識以關鍵績效指標 (KPI) 為基礎的成果何時存在風險，或何時出現非預期的異常狀況。以 KPI 做為警示的基礎，可確保您收到的訊號與業務或營運影響直接相關。這種可付諸行動的警示可推動主動回應，且有助於維持系統效能和可靠性。

期望的結果：接收及時、相關且可付諸行動的警示，以便迅速找出並緩解潛在問題，尤其是 KPI 成果存在風險時。

常見的反模式：

- 設定太多非嚴重警示，導致警示疲勞。
- 未根據 KPI 排定警示的優先順序，因此難以了解問題對業務造成的影響。
- 忽略解決根本原因，導致一再出現相同問題的警示。

建立此最佳實務的優勢：

- 專注於可付諸行動且相關的警示，以減少警示疲勞的情況。
- 透過主動偵測和緩解問題，改善系統運作時間和可靠性。
- 透過整合熱門的警示和通訊工具，強化團隊協作並加快問題解決速度。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要建立有效的警示機制，則務必使用指標、日誌和追蹤資料，因為這些資料會在 KPI 為基礎的成果存在風險或偵測到異常時發出訊號。

實作步驟

1. 確定關鍵績效指標 (KPI)：識別應用程式的 KPI。警示應與這些 KPI 密切相關，才能準確反映業務影響。
2. 實作異常偵測：
   - 使用 Amazon CloudWatch 異常偵測：設定 Amazon CloudWatch 異常偵測以自動偵測不尋常模式，協助您只產生真正的異常警示。
   - 使用 AWS X-Ray Insights：
     a. 設定 X-Ray Insights 以偵測追蹤資料中的異常情況。
     b. 設定 X-Ray Insights 的通知，以便在偵測到問題時發出警示。
   - 與 Amazon DevOps Guru 整合：
     a. 利用 Amazon DevOps Guru 的機器學習功能來偵測現有資料中的操作異常狀況。
     b. 瀏覽至 DevOps Guru 中的通知設定以設定異常警示。
3. 實作可行的警示：設計提供足夠資訊的警示，以便於立即採取行動。
   1. 監控使用 Amazon EventBridge 規則的 AWS Health 事件，或以程式設計方式與 AWS Health API 整合，以在收到 AWS Health 事件時自動執行動作。這些可能是一般動作 (例如將所有計畫的生命週期事件訊息傳送到聊天介面) 或特定動作 (例如在 IT 服務管理工具中啟動工作流程)。
4. 減少警示疲勞：將非重要警示減到最少。若產生大量不重要的警示使團隊疲於奔命，團隊會疏忽嚴重的問題，而降低警示機制的整體效用。
5. 設定複合警示：使用 Amazon CloudWatch 複合警示可合併多個警示。
6. 與提醒工具整合：結合 Ops Genie 和 PagerDuty 等工具。
7. 與 AWS Chatbot「接合」：整合 AWS Chatbot 以將警示轉送到 Amazon Chime、Microsoft Teams 和 Slack。
8. 基於日誌的提醒：使用 CloudWatch 中的日誌指標篩選條件，根據特定日誌事件建立警示。
9. 檢閱和迭代：定期重新檢視並改善警示組態。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測

- OPS04-BP03 實作使用者體驗遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作分散式追蹤
- OPS08-BP01 分析工作負載指標
- OPS08-BP02 分析工作負載日誌
- OPS08-BP03 分析工作負載追蹤

相關文件：

- 使用 Amazon CloudWatch 警示
- 建立複合警示
- 根據異常偵測建立 CloudWatch 警示
- DevOps Guru 通知
- X-ray Insights 通知
- 透過互動式 ChatOps 進行監控、操作和疑難排解您的 AWS 資源
- Amazon CloudWatch 整合指南 | PagerDuty
- 將 Opsgenie 與 Amazon CloudWatch 整合

相關影片：

- 在 Amazon CloudWatch 中建立複合警示
- AWS Chatbot 概觀
- AWS On Air ft.AWS Chatbot 中的變異命令

相關範例：

- 雲端中使用 Amazon CloudWatch 的警示、事件管理和修復功能
- 教學課程：建立 Amazon EventBridge 規則以將通知傳送至 AWS Chatbot
- One Observability 研討會

OPS08-BP05 建立儀表板

儀表板提供人性化的檢視方式，讓您深入了解工作負載的遙測資料。雖然儀表板是重要的視覺介面，但不應取代警示機制，而是相輔相成。經過精心打造後，儀表板不僅能提供快速了解系統運作狀況和效能的洞見，還能對利害關係人呈現有關業務成果和問題影響層面的即時資訊。

期望的結果：

使用視覺呈現的方式，提供清楚、深入系統與業務運作狀況且可付諸行動的洞見。

常見的反模式：

- 包含太多指標、過於複雜的儀表板。
- 依賴儀表板，卻沒有異常偵測警示。
- 儀表板未隨著工作負載發展而更新。

本最佳實務的優勢：

- 立即掌握關鍵系統指標和 KPI。
- 強化利害關係人的溝通與理解。
- 快速深入洞察操作問題的影響層面。

未建立此最佳實務時的風險等級：中

實作指引

以業務為中心的儀表板

專為業務 KPI 量身打造的儀表板，可與更廣泛的利害關係人進行互動。儘管這些人可能對系統指標不感興趣，但他們會急於了解這些數字對業務的影響。以業務為中心的儀表板可確保所有受監控且經過分析的技術和操作指標，都與總體業務目標保持同步。這種一致性確保每個人清楚了解目標，且對於重要性有共同的認知。此外，強調業務 KPI 的儀表板往往更能付諸行動。利害關係人能夠迅速了解營運狀況、需要關注的環節，以及可能對業務成果造成的影響。

了解這點之後，在建立儀表板時，請務必在技術指標與業務 KPI 之間取得平衡。兩者都至關重要，但要滿足的對象不同。在理想情況下，您應有能夠提供全方位視角儀表板，以便深入掌握系統運作狀況與效能，同時也要強調關鍵業務成果及其影響。

Amazon CloudWatch 儀表板是 CloudWatch 主控台中可自訂的首頁，可用來在單一檢視中監控您的資源，甚至能監控分散到不同 AWS 區域 和帳戶中的資源。

實作步驟

1. 建立基本儀表板： 在 CloudWatch 中建立新的儀表板，並提供描述性名稱。
2. 使用 Markdown 小工具：在深入了解指標之前，請使用 Markdown 小工具在儀表板頂部新增文字內容。此內容應說明儀表板涵蓋的內容、所呈現指標的重要性，還可以包含其他儀表板和疑難排解工具的連結。
3. 建立儀表板變數：在適當的情況下合併儀表板變數，以呈現動態靈活的儀表板檢視畫面。
4. 建立指標小工具：新增指標小工具以便將應用程式產生的各種不同指標視覺化，並調整這些小工具以便有效呈現系統運作狀況和業務成果。
5. Log Insights 查詢：利用 CloudWatch Log Insights 從日誌中產生可行的指標，並且在儀表板上顯示這些洞見。
6. 設定警示：將 CloudWatch 警示整合到儀表板中，以快速檢視任何違反其閾值的任何指標。
7. 使用 Contributor Insights：併入 CloudWatch Contributor Insights 以分析高基數欄位，並且更清楚了解資源的首要貢獻者。
8. 設計自訂小工具：對於未能透過標準小工具滿足的特定需求，可考慮建立自訂小工具。這些小工具可從各種資料來源中提取資料，或以獨特的方式呈現資料。
9. 使用 AWS Health Dashboard：利用 AWS Health Dashboard 深入了解您的帳戶運作狀態、事件，以及可能影響服務和資源且即將進行的變更。您也可以在您的 AWS Organizations 中取得運作狀態事件的集中檢視，或建置您自己的自訂儀表板 (如需詳細資訊，請參閱相關範例)。
10. 反覆執行並改進：隨著應用程式發展，請定期重新檢視您的儀表板，以確保其相關性。

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS08-BP01 分析工作負載指標
- OPS08-BP02 分析工作負載日誌
- OPS08-BP03 分析工作負載追蹤
- OPS08-BP04 建立可付諸行動的警示

相關文件：

- 建置用於檢視營運狀況的儀表板

- 使用 Amazon CloudWatch 儀表板

相關影片：

- 建立跨帳戶和跨區域 CloudWatch 儀表板
- AWS re:Invent 2021 - 透過 AWS 雲端 營運儀表板獲得企業能見度)

相關範例：

- One Observability 研討會
- 使用 Amazon CloudWatch 進行應用程式監控
- AWS Health 事件情報儀表板和洞察
- 使用 Amazon Managed Grafana 視覺化 AWS Health 事件

OPS 9.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 使用指標衡量營運目標與 KPI
- OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況
- OPS09-BP03 檢閱營運指標並優先改進

OPS09-BP01 使用指標衡量營運目標與 KPI

從您的組織取得定義營運成功的目標和 KPI，並決定反映這些目標的指標。設定基準做為參考點，並定期重新評估。制定機制以便從團隊收集這些指標以進行評估。

預期成果：

- 已發佈並共用組織運營團隊的目標和 KPI。
- 已建立反映這些 KPI 的指標。範例包括：
  - 票證佇列深度或票證平均存留時間
  - 依問題類型分組的票證計數
  - 處理問題所花的時間，無論是否有標準作業程序 (SOP)

- 從失敗的程式碼推送復原所花的時間長度
- 通話量

常見的反模式：

- 錯過部署期限，因為開發人員須分心處理疑難排解工作。開發團隊要求更多人力，但無法提出確切需要的人力數量，因為無法衡量被佔用的時間。
- 設立了 1 級服務台來處理使用者通話。經過一段時間後，加入了更多工作負載，但並沒有分配更多人員給 1 級服務台。客戶滿意度受到通話時間增加及問題未解決的時間拉長影響而下降，但管理層看不到這些現象的指標，未能採取任何行動。
- 有問題的工作負載已交由另一個營運團隊進行維護。與其他工作負載不同的是，並未針對這個新工作負載提供適當的文件和執行手冊。因此，團隊花費更長的時間進行疑難排解和解決失敗情況。然而，沒有任何指標記載此情況，因此無法明確究責。

建立此最佳實務的優勢： 只要工作負載監控顯示我們應用程式和服務的狀態，監控營運團隊就可讓擁有者深入了解這些工作負載取用者之間的變化，例如業務需求轉變。藉由建立能夠反映營運狀態的指標來衡量這些團隊的效用，並依據業務目標進行評估。指標可突顯支援問題，或識別何時發生偏離服務層級目標的情形。

未建立此最佳實務時的曝險等級： 中

實作指引

安排時間與企業領導者和利害關係人一起確定服務的整體目標。確定各個不同營運團隊應負責的任務，以及能夠應對哪些挑戰。使用這些來集思廣益，找出能夠反映這些營運目標的關鍵績效指標 (KPI)。這些可能包括客戶滿意度、從形成功能概念到部署的時間、平均問題解決時間及其他方面。

從 KPI 中找出最能反映這些目標的資料指標和來源。客戶滿意度可能由各種不同的指標組合而成，例如通話等待或回應時間、滿意度分數，以及提出的問題類型。部署時間可能是測試和部署，加上任何需要新增的部署後修正所需時間的總和。顯示不同類型的問題所花費時間 (或是這些問題的計數) 的統計資料，可提供一個切入視角，以了解需要針對性處理的地方。

資源

相關文件：

- Amazon QuickSight - 使用 KPI
- Amazon CloudWatch - 使用指標

- 建置儀表板
- 如何使用 KPI 儀表板追蹤成本最佳化 KPI

OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況

您須了解營運狀態及趨勢方向，以確定成果何時可能存在風險、是否可支援新增的工作，或是變更對您的團隊造成的影響。營運事件發生時，有提供使用者和營運團隊參考資訊的狀態頁面，就能減輕溝通管道的壓力，並有效傳播資訊。

預期成果：

- 主管對團隊處理的通話量類型和正在進行的工作 (例如部署)可以一目瞭然。
- 發生影響擴及正常營運的情況時，利害關係人和使用者社群就會收到警示。
- 組織領導階層和利害關係人可查看狀態頁面以便回應警示或影響，並且獲得有關營運事件的資訊，例如聯絡窗口、票證資訊及預估的復原時間。
- 領導階層和其他利害關係人會收到報告，報告中會顯示營運統計資料，例如某一段時間內的通話量、使用者滿意度分數、待處理票證數量及其存留時間。

常見的反模式：

- 工作負載停擺，造成服務無法使用。通話量暴增，因為使用者要求得知發生什麼情況。主管也要求得知誰在處理問題，因而增加了通話量。不同的營運團隊重複投入嘗試調查的工作。
- 由於需要新功能，因而轉派數名人員進行工程工作。但未回補空缺，使得解決問題的時間大幅拉長。領導階層並未獲得這些資訊，而是在經過數週且收到使用者不滿意的意見回饋後才察覺此問題。

建立此最佳實務的優勢： 在業務受到影響的營運事件中，各個團隊可能會浪費大量時間和精力來查詢資訊，以試圖了解情況。透過建立廣泛傳播的狀態頁面和儀表板，利害關係人就能迅速獲得資訊，例如是否偵測到問題、誰負責處理問題，或是預計何時恢復正常營運。如此一來，團隊成員就不需花太多時間與其他人溝通狀態，因而有更多時間解決問題。

未建立此最佳實務時的曝險等級： 中

實作指引

建置儀表板，以顯示營運團隊目前的關鍵指標，並且讓營運主管和管理層隨時可存取這些資訊。

建置可快速更新的狀態頁面，以顯示事故或事件何時發生、負責人是誰，以及誰負責協調回應。在此頁面上分享使用者應考量的任何步驟或因應措施，並廣泛傳播位置。鼓勵使用者遇到未知的問題時，先查看此位置。

收集並提供報告，以顯示長時間的營運狀況，並將此資訊傳達給主管和決策者，以說明運營工作及挑戰和需求。

在團隊之間共用這些最能反映目標和 KPI 的指標和報告，以及這些資訊在推動變革方面的影響力。花時間進行這些活動，以在團隊內部和團隊之間提高營運的重要性。

資源

相關文件：

- 測量進度
- 建置用於檢視營運狀況的儀表板

相關解決方案：

- 資料操作

OPS09-BP03 檢閱營運指標並優先改進

預留檢閱營運狀態的專屬時間和資源，以確保依舊優先處理日常業務線所需的服務。召集營運主管和利害關係人定期檢閱指標、重新確認或修改各項目標，並優先改進。

預期成果：

- 營運主管和員工定期開會，以檢閱一段特定報告期間的指標。說明挑戰、一同慶祝成就，並分享學到的經驗。
- 利害關係人與企業領導者會定期收到營運狀態的簡報，並徵求有關目標、KPI 和未來計畫的意見。討論服務交付、營運和維護之間的權衡，並納入相關環境中。

常見的反模式：

- 新產品已推出，但 1 級和 2 級營運團隊未接受足夠的培訓來提供支援，或未配置額外的人員。領導者未看見指出支援單解決次數減少且事故量增加的指標。訂閱數量隨著不滿的使用者離開平台而開始減少，但數週後才採取行動。

- 長久以來一直採用手動程序來執行工作負載維護工作。雖然渴望自動化，但由於系統的重要性較低，因此優先順序較低。然而經過一段時間後，系統的重要性已提高，而現在這些手動程序佔用了大多數營運時間。未安排資源來提供更多營運工具，導致員工隨著工作負載增加而倦怠。等到有員工離職並加入其他競爭對手，領導階層才察覺到此情況。

建立此最佳實務的優勢： 在某些組織中，將相同的時間和注意力分配給服務交付和新產品或方案可能會是一項挑戰。發生這種情況時，業務線可能會因為預期的服務層級逐漸惡化而受到影響。這是因為營運未隨著業務成長而改變和發展，並且可能很快就會落後。假如未定期檢閱營運收集的洞見，那麼察覺到業務風險時，便可能為時已晚。透過分配時間與營運員工和領導階層一起檢閱指標和程序，就能持續掌握營運所扮演的重要角色，並且能夠在風險達到嚴重等級之前發現。營運團隊能夠更深入洞察即將發生的業務變化與計畫，進而採取積極的行動。領導階層對於營運指標的掌握程度，展現了這些團隊在內部和外部的客戶滿意度方面所扮演的角色，並讓他們在各種選擇當中權衡出更適當的優先順序，或確保營運團隊有時間和資源能夠隨著新的業務和工作負載計畫做出改變與發展。

未建立此最佳實務時的曝險等級： 中

實作指引

花時間與利害關係人和營運團隊一起檢閱營運指標，並檢閱報告資料。將這些報告與組織的目標相互比對，以確定是否符合這些目標。找出目標不明確，或者要求與付出之間存在衝突的模糊地帶來源。

找出時間、人員和工具能夠協助實現營運成果的地方。確定哪些 KPI 會受到影響，以及哪些應是成功的目標。定期重新檢視，以確保營運資源充足，可支援業務線。

資源

相關文件：

- Amazon Athena
- Amazon CloudWatch 指標和維度參考
- Amazon QuickSight
- AWS Glue
- AWS Glue Data Catalog
- 使用 Amazon 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

OPS 10.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 定義處理影響服務事件的客戶通訊計畫
- 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

培養能快速管理事件、事故和問題的能力是維持工作負載執行狀況和效能的關鍵。在制定有效的回應和解決策略時，認識並了解這些元素之間的差異是其中關鍵。建立並遵循每個方面的完善定義程序，可協助您的團隊快速有效地處理出現的任何營運挑戰。

預期成果： 您的組織能透過完善記錄文件與集中儲存程序，有效地管理營運事件、事故和問題。這些程序會持續更新以反映變更，簡化處理並維持高服務可靠性和工作負載效能。

常見的反模式：

- 您採用回應方式，而非主動回應事件。
- 採取不一致的方法來處理不同類型的事件或事故。
- 您的組織未分析事件並從中獲得經驗，防止未來再次發生。

建立此最佳實務的優勢：

- 已簡化和標準化的回應程序。
- 減少事件對服務和客戶的影響。
- 快速解決問題。
- 持續改善營運程序。

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。

了解事件、事故和問題

- 事件： 一個 事件 是對動作、狀況或狀態變化的觀察。事件可以是計劃也可能是意外，其發生源頭可能來自工作負載的內部或外部。
- 事故： 事故 是需要回應的事件，例如非計畫的服務中斷或服務品質下降。它們代表需要立即注意，才能恢復正常工作負載作業的中斷。
- 問題： 問題 是一個或多個事故的基本原因。識別和解決問題時必須深入研究事故，預防未來再次發生。

實作步驟

事件

1. 監控事件：
   - 實作可觀測性 和 利用工作負載可觀察性。
   - 由使用者、角色或 AWS 服務所採取的監控動作會記錄成 AWS CloudTrail。
   - 即時回應應用程式中的營運相關變更時可搭配 Amazon EventBridge。
   - 持續評估、監控和記錄資源組態變更與 AWS Config。
2. 建立程序：
   - 制定程序，以便評估哪些事件特別重要而需要加以監控。這當中包含為正常活動和異常活動設定相關臨界值和參數。
   - 確定事件將向上呈報的條件。這可能是根據嚴重程度、對使用者的影響或與預期行為的偏差而定。
   - 定期檢閱事件監控和回應程序。這包括分析過去的事件、調整臨界值以及改進提醒機制。

事故

1. 回應事故：
   - 利用可觀測性工具的洞察，快速識別事故並進行回應。
   - 實作 AWS Systems Manager 操作中心 以彙總、組織及排序營運項目和事件的優先順序。
   - 使用多項服務，例如 Amazon CloudWatch 和 AWS X-Ray 進行更深入的分析和疑難排解。

- 考慮 AWS Managed Services (AMS) 利用其主動、預防和偵測功能，進而增強事件管理。AMS 會擴展營運支援，過程中透過監控、事件偵測和回應以及安全管理等服務。
- Enterprise Support 客戶可以要求 AWS 事件偵測與回應，為生產工作負載提供持續主動監控和事件管理。

2. 建立事件管理程序：
   - 建立結構化事件管理程序，包括清晰的角色、通訊協定和解決步驟。
   - 整合事件管理與多項工具，像是 AWS Chatbot 達到快速的回應和協調。
   - 根據嚴重性進行事件分類，且預先定義 每個類別適用的 事件回應計畫。
3. 學習和改善：
   - 舉行 事件後分析 了解根本原因和解決有效成果。
   - 根據檢閱和不斷變化的實務，持續更新和改善回應計畫。
   - 記錄並分享跨團隊所獲得的經驗，以增強營運韌性。
   - 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。

問題

1. 識別問題：
   - 使用先前事件的資料，識別可能指出更深入系統問題的重複模式。
   - 利用多種工具，像是 AWS CloudTrail 和 Amazon CloudWatch 來分析趨勢，並發現潛在問題。
   - 參與跨職能團隊，包括營運、開發和業務單位，找出根本原因的各種觀點。
2. 建立問題管理程序：
   - 制定結構化的問題管理程序，專注於長期解決方案，而非快速修復。
   - 納入根本原因分析 (RCA) 技術，以調查並了解事件的基本原因。
   - 根據調查結果，更新營運政策、程序和基礎架構，防止重複發生。
3. 持續改善：
   - 培養持續學習和改進的文化，鼓勵團隊主動識別並解決潛在問題。
   - 定期檢閱和修訂問題管理程序和工具，以因應不斷變化的業務和技術環境。
   - 在整個組織中分享洞察和最佳實務，以建立更具韌性和效率的營運環境。
4. 參與 AWS Support：
   - 使用 AWS 支援資源，例如 AWS Trusted Advisor，用於主動指導和最佳化建議。

- 企業支援客戶可以存取專門的程式，例如 AWS 倒數 提供關鍵事件期間的支援。
-

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- 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
- OPS08-BP01 分析工作負載指標
- OPS11-BP02 執行事件後分析

相關文件：

- AWS 安全事件應變指南
- AWS 事件偵測與回應
- AWS 雲端採用架構：營運觀點 - 事件與問題管理
- DevOps 和 SRE 時代的事故管理
- PagerDuty - 什麼是事故管理？

相關影片：

- 常用事件回應提示來自 AWS
- AWS re:Invent 2022 - Amazon 建置者資料中心：25 年的 Amazon 卓越營運
- AWS re:Invent 2022 - AWS 事件偵測和回應 (SUP201)
- 向您介紹 AWS Systems Manager 推出的 Incident Manager

相關範例：

- AWS 主動服務 – 事件管理研討會

- 如何使用 PagerDuty 和 AWS Systems Manager Incident Manager 自動化事件回應
- 使用 AWS Systems Manager Incident Manager 中的呼叫時間表與事件回應人員聯絡
- 改善在 AWS Systems Manager Incident Manager 中處理事件期間的能見度和協作
- 在 AMS 中的事件報告和服務請求

相關服務：

- Amazon EventBridge

OPS10-BP02 每個提醒建立一個程序

為系統中的每個提醒建立清晰且完整定義的程序是達到有效快速事件管理的關鍵。這個實務可確保每個提醒都會導致特定、可採取動作的回應，進而提高營運的可靠性和回應能力。

預期成果： 每個提醒都會啟動特定、定義明確的回應計畫。在可能的情況下，回應會自動化執行，具備清晰的擁有權和清楚定義的呈報路徑。提醒則連結至最新知識庫，因此任何操作人員都可以一致且有效地進行回應。回應會快速且一致地傳達至整個管理層，提高營運效率和可靠性。

常見的反模式：

- 提醒無預定義的回應程序，導致臨時湊合與延遲的解決方案。
- 提醒過載，導致重要提醒遭到忽略。
- 提醒處理不一致，因為缺乏明確的擁有權和責任。

建立此最佳實務的優勢：

- 減少提醒疲勞，透過僅提出可採取動作的提醒。
- 縮短營運相關問題的平均解決時間 (MTTR)。
- 縮短平均調查時間 (MTTI)，利於降低 MTTR。
- 增強可擴充營運相關回應的能力。
- 改善處理營運相關事件的一致性和可靠性。

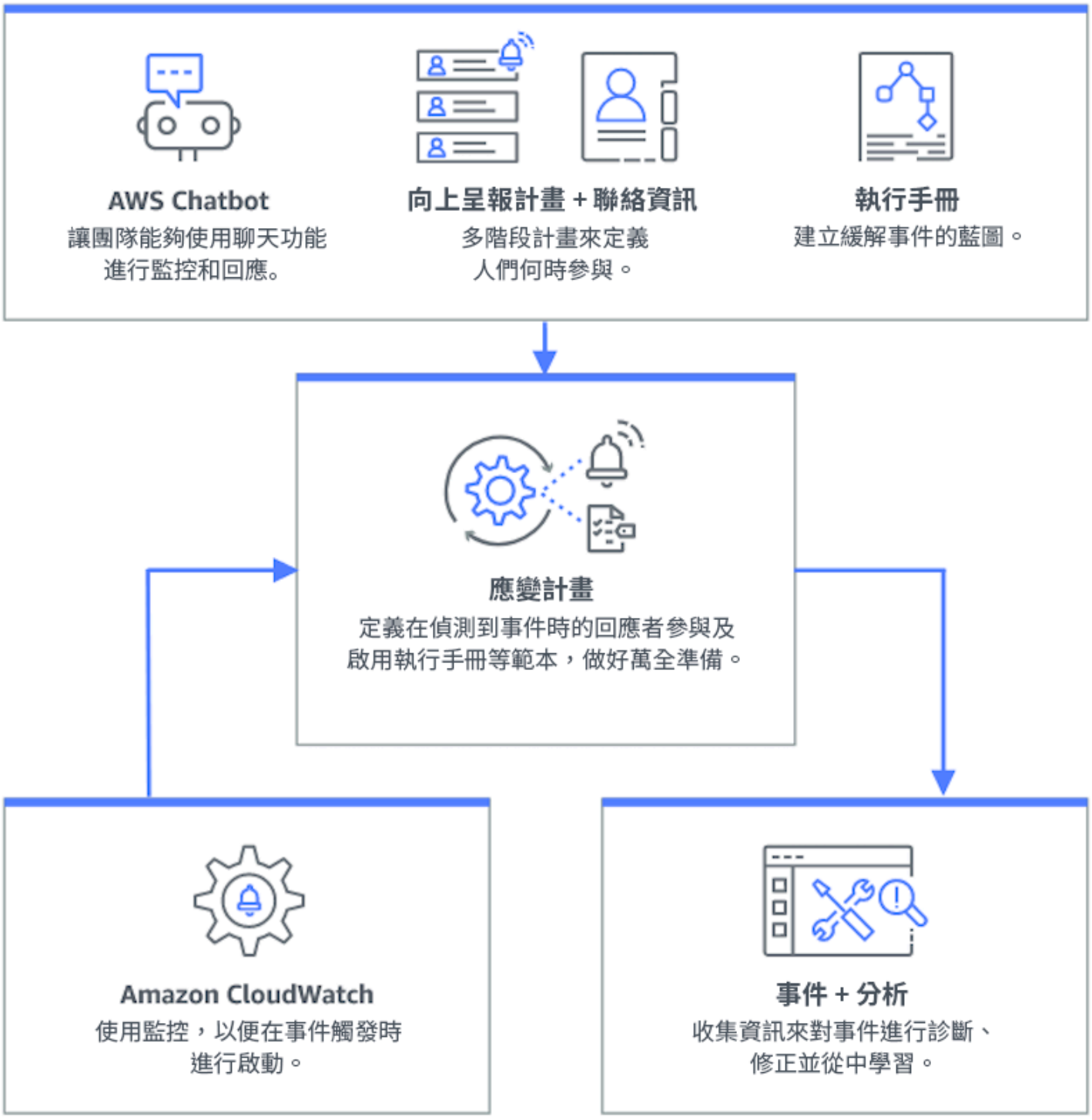
未建立此最佳實務時的曝險等級： 高

實作指引

每個提醒設定一個程序，需要為每個提醒建立清晰的回應計畫，在可能的情況下自動化執行回應，並根據營運意見反映和不斷變化的需求，持續完善這些程序。

實作步驟

下圖說明事件管理工作流程於 AWS Systems Manager Incident Manager。它可快速回應營運相關問題，期間透過自動建立事件，以便回應來自 Amazon CloudWatch 或者 Amazon EventBridge。建立事件之後，Incident Manager 可能以自動或手動方式，集中管理事件、組織相關 AWS 資源資訊，並啟動預定義的回應計畫。這包括執行 Systems Manager 自動化執行手冊以立即採取行動，同時在 OpsCenter 中建立上層營運工作項目，以利追蹤相關的工作和分析。這個簡化程序加快並協調在您整體 AWS 環境中的所有事件回應。



1. 使用複合警報： 建立 複合警報 於 CloudWatch，以便分組相關警報，減少雜訊，並做出更有意義的回應。
2. 整合 Amazon CloudWatch 警報與 Incident Manager 設定 CloudWatch 警報以自動建立事件於 AWS Systems Manager Incident Manager。

3. 整合 Amazon EventBridge 與 Incident Manager： 建立 EventBridge 個規則 以回應事件，並建立使用定義回應計畫的事件。
4. 為 Incident Manager 中的事件做準備：
   - 建立詳細的 回應計畫 於 Incident Manager 中，以因應各種提醒類型。
   - 建立聊天管道經由 AWS Chatbot 已連接到 Incident Manager 中的回應計畫，促進在 Slack、Microsoft Teams 和 Amazon Chime 等平台之間發生事件的即時通訊。
   - 納入 Systems Manager 自動化執行手冊 於 Incident Manager 以推動自動化事件回應。

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS08-BP04 建立可付諸行動的警示

相關文件：

- AWS 雲端採用架構：營運觀點 - 事件與問題管理
- 使用 Amazon CloudWatch 警報
- 設定 AWS Systems Manager Incident Manager
- 為 Incident Manager 中的事件做準備

相關影片：

- 常用事件回應提示來自 AWS

相關範例：

- AWS 研討會 - AWS Systems Manager Incident Manager - 自動化安全性事件的事件回應

OPS10-BP03 根據業務影響確定營運事件的優先順序

是否能迅速回應營運事件很重要，但並非每件事件都同樣重要。當您根據業務影響排定優先順序時，您也會排定解決可能會導致重大後果的事件，例如安全、財務損失、違反法規或損害聲譽。

預期成果： 營運事件回應的順序排定根據是業務營運和目標可能受到的影響程度。這樣會提高回應的效率和有效性。

常見的反模式：

- 每個事件都以相同的緊急程度處理，導致解決重要問題時造成混亂和延遲。
- 您未區分高影響力和低影響力的事件，導致資源分配錯誤。
- 您的組織缺乏清晰的優先順序架構，所以無法依照一致方式來回應營運事件。
- 事件優先順序是根據報告時間順序排列，而不是根據業務成果所受到的影響。

建立此最佳實務的優勢：

- 確保關鍵業務功能可優先獲得關注，最大限度降低潛在損害。
- 改善在多重並行事件期間的資源分配。
- 增強組織維持信任和遵循法規要求的能力。

未建立此最佳實務時的曝險等級： 中

實作指引

面對多起營運事件時，一定要根據所造成的影響和緊急性，採用經過結構化安排的優先事項進行處理。這種方法可協助您做出明智決策，直接有效處理最需要的地方，並緩解業務持續性的風險。

實作步驟

1. 評估影響： 制定分類系統，以便根據事件對企業營運和目標的潛在影響來評估事件的嚴重性。。下列範例顯示影響類別：

影響層級	描述
高	影響許多員工或客戶，高度財務影響、高度聲譽受損或傷害。
中	影響特定群組的員工或客戶，中度財務影響或中度聲譽受損。
低	影響個別員工或客戶，低度財務影響或低度聲譽受損。

2. 評估緊急程度： 定義不同緊急程度需要多快回應事件，期間考慮像是安全性、財務影響和服務層級協議 (SLA) 等因素。下列範例示範緊急情況類別：

緊急程度	描述
高	損害程度大增、受影響的時效性工作、即將升級呈報，或受到影響的 VIP 使用者或群組。
中	損害隨著時間而增加，或受影響的單一 VIP 使用者或群組。
低	邊緣損害隨著時間而增加，或受影響的非時效性工作。

3. 建立優先順序矩陣：
   - 使用矩陣來交互參考影響和緊急性，將不同優先順序指定給不同的組合。
   - 製作能讓負責營運事件回應的所有團隊成員都能存取和理解的表格矩陣。
   - 下列範例矩陣會根據緊急程度和影響力顯示不同嚴重性的事件：

緊急性和影響	高	中	低
高	嚴重	緊急	高
中	緊急	高	正常
低	高	正常	低

4. 訓練和溝通： 訓練回應團隊認識優先順序矩陣，以及在事件發生期間遵循矩陣的重要性。向所有利益相關者傳達優先順序，確定明確的期望。
5. 整合事件回應：
   - 將優先順序矩陣納入您的事件回應計畫和工具。
   - 在可能情況下，自動化事件的分類和優先順序，加快回應時間。
   - 企業支援客戶可以利用 AWS 事件偵測與回應，為生產工作負載提供全年無休 24x7 的主動監控和事件管理。
6. 檢閱和適應： 定期檢閱優先順序程序的有效性，並根據業務環境的意見回應和變化進行調整。

資源

相關的最佳實務：

- OPS03-BP03 鼓勵向上呈報
- OPS08-BP04 建立可付諸行動的警示
- OPS09-BP01 使用指標衡量營運目標與 KPI

相關文件：

- Atlassian - 了解事件嚴重層級
- IT 流程圖 - 檢查清單事件優先順序

OPS10-BP04 定義向上呈報路徑

在您的事件回應協定中建立清晰的向上呈報路徑，以促進及時有效的動作。這包括指定向上呈報命令、詳載向上呈報程序，以及預先核准動作，以加快決策，並縮短平均解決時間 (MTTR)。

預期成果： 結構化且有效率的程序，可將事件向上呈報給適當人員，將回應時間和影響降到最低。

常見的反模式：

- 復原程序缺乏清晰度，導致在嚴重事件期間進行臨時回應。
- 缺少定義的權限和擁有權，導致在需要緊急動作時發生延遲。
- 利益相關者和客戶未依預期收到通知。
- 重要決定發生延遲。

建立此最佳實務的優勢：

- 簡化的事件回應會透過預先定義的向上呈報程序。
- 縮短停機時間，期間搭配預先核准的動作和清晰的擁有權。
- 根據事件嚴重性，改善資源分配和支援層級調整。
- 改善與利益相關者和客戶的溝通。

未建立此最佳實務時的曝險等級： 中

實作指引

快速事件回應的關鍵是定義正確的向上呈報路徑。AWS Systems Manager Incident Manager 支援設定結構化向上呈報計畫和呼叫時間表，這些計畫會提醒適當的人員，以便他們隨時能在發生事件時採取行動。

實作步驟

1. 設定向上呈報命令： 設定 CloudWatch 警報 建立事件於 AWS Systems Manager Incident Manager。
2. 設定呼叫時間表： 建立 呼叫時間表 於 Incident Manager，且時間表與您的向上呈報路徑一致。為當值人員提供必要的權限和工具，以迅速採取行動。
3. 詳細的向上呈報程序：
   - 確定事件應在哪些特定條件下進行呈報。
   - 建立 向上呈報計畫 於 Incident Manager。
   - 向上呈報通道應由聯絡人或呼叫時間表組成。
   - 定義每個向上呈報層級團隊的角色和責任。
4. 預先核准緩解措施： 與決策者合作，預先核准預期情境的動作。使用 Systems Manager 自動化執行手冊 已整合 Incident Manager，而能加快事件解決速度。
5. 指定擁有權： 清楚地識別向上呈報路徑每步驟的內部擁有者。
6. 詳細第三方向上呈報：
   - 記錄第三方服務層級協議 (SLA)，並且根據內部目標進行調整一致。
   - 設定在事件期間，與供應商通訊的清晰協定。
   - 將供應商聯絡人整合到事件管理工具中，以供直接存取。
   - 舉行定期鑽研，包括第三方回應情境。
   - 維持供應商向上呈報資訊妥善記錄，並且易於存取。
7. 訓練和排練向上呈報計畫： 訓練您的團隊進行向上呈報程序，並定期舉辦事件回應鑽研或演練日。Enterprise Support 客戶可以要求 事件管理研討會。
8. 持續改善： 定期檢閱向上呈報路徑的有效性。根據事件事後分析獲得的經驗與持續意見反映，更新您的程序。

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS08-BP04 建立可付諸行動的警示
- OPS10-BP02 每個提醒建立一個程序
- OPS11-BP02 執行事件後分析

相關文件：

- AWS Systems Manager Incident Manager 向上呈報計畫
- 在 Incident Manager 中處理呼叫時間表
- 建立和管理執行手冊
- 使用 AWS IAM Identity Center 管理臨時提升的存取權
- Atlassian - 提供有效事件管理的向上呈報政策

OPS10-BP05 定義處理影響服務事件的客戶通訊計畫

維持客戶的信任感和透明度的關鍵是在影響服務事件期間進行有效通訊。完善定義的通訊計畫可協助組織在事件發生期間快速並清楚地分享內部和外部資訊。

預期成果：

- 強大的通訊計畫，可在影響服務事件期間有效通知客戶和利益相關者。
- 通訊過程維持透明，可以建立信任感，並減少客戶焦慮。
- 將服務影響事件對客戶體驗和業務營運的影響降到最低。

常見的反模式：

- 通訊不足或延遲會導致客戶混淆和心情不悅。
- 訊息過於技術性或模糊，無法傳達使用者受到的實際影響。
- 沒有預先定義的通訊策略，導致訊息不一致和反應性不佳。

建立此最佳實務的優勢：

- 透過主動和清晰的通訊，增強客戶信任感和滿意度。

- 透過預先解決客戶的疑慮，減少支援團隊的負擔。
- 提高可有效管理事件和從中復原的能力。

未建立此最佳實務時的曝險等級： 中

實作指引

建立適合影響服務事件的全面性通訊計畫涉及多方面，從選擇合適的管道到製作訊息和語音。計畫應具適應性、可擴展性，並可滿足不同的服務中斷情境。

實作步驟

1. 定義角色和責任：
    - 指派主要事件經理人員來監督事件回應活動。
    - 指定負責協調所有外部和內部通訊的通訊經理。
    - 包括可經由支援票證進行一致通訊的支援經理人員。
2. 識別通訊管道： 選擇像是工作場所聊天、電子郵件、SMS 簡訊、社群媒體，應用程式內通知和狀態頁面等管道。這些管道應具足夠的韌性，在服務受影響的事件發生的期間能夠獨立運行。
3. 與客戶進行快速、清晰、定期的通訊：
    - 開發適合各種服務障礙情況使用的範本，內容強調簡潔和基本細節。包括服務損毀、預期解決時間和影響的相關資訊。
    - 使用 Amazon Pinpoint，利用推送通知、應用程式內通知、電子郵件、簡訊、語音訊息和經由自訂通道訊息來提醒客戶。
    - 使用 Amazon Simple Notification Service (Amazon SNS) 以程式設計方式，或透過電子郵件、行動推送通知和簡訊提醒訂閱者。
    - 透過公開分享 Amazon CloudWatch 儀表板，透過儀表板傳達狀態。
    - 鼓勵社群媒體互動：
        - 主動監控社群媒體以了解客戶情緒。
        - 在社群媒體平台上發布以進行公開更新和社群互動。
        - 準備能進行一致且清晰的社群媒體通訊的範本。
4. 協調內部通訊： 實作內部協定，使用像是 AWS Chatbot 等工具進行團隊協調和通訊。使用 CloudWatch 儀表板以傳達狀態。
5. 協調通訊，期間搭配專用工具和服務：

- 使用 AWS Systems Manager Incident Manager 搭配 AWS Chatbot，並設定專用聊天通道，以便在事件期間即時進行內部通訊和協調。
- 使用 AWS Systems Manager Incident Manager 執行手冊，在事件期間，自動透過 Amazon Pinpoint、Amazon SNS，或社群媒體平台等第三方工具處理客戶通知。
- 在執行手冊中納入核准工作流程，即可選擇性地檢閱，並在授權所有外部通訊之後，再進行傳送。

6. 實務和改善：
   - 舉行有關使用通訊工具和策略的訓練。增強團隊能力，使其能夠在事件期間及時做出決定。
   - 透過定期鑽研或演練日測試通訊計畫。使用這些測試來完善訊息，並評估通道的有效性。
   - 實作意見反映機制，以便評估事件期間的通訊效果。持續根據意見反映和不斷變化的需求來改善通訊計畫。

實作計劃的工作量： 高

資源

相關的最佳實務：

- OPS07-BP03 使用執行手冊執行程序
- OPS10-BP06 透過儀表板傳達狀態
- OPS11-BP02 執行事件後分析

相關文件：

- Atlassian - 事件通訊最佳實務
- Atlassian - 如何編寫良好狀態更新
- PageDuty - 事件通訊指南

相關影片：

- Atlassian - 建立您自己的事件通訊計畫：事件範本

相關範例：

- AWS Health 儀表板

- 範例 AWS 狀態更新

OPS10-BP06 透過儀表板傳達狀態

使用儀表板做為策略性工具，即時將營運狀態和關鍵指標傳達給不同對象，包括內部技術團隊、領導層和客戶。這些儀表板會以視覺化方式，集中提供系統執行狀況和業務效能，提高資料透明度和決策效率。

預期成果：

- 您的儀表板提供與不同利益相關者相關的系統和業務指標的全面檢視。
- 利益相關者可以主動存取營運資訊，進而降低所需要的頻繁狀態要求。
- 正常操作和事件期間的即時決策獲得增強。

常見的反模式：

- 加入事件管理呼叫的工程師需要狀態更新才能加快速度。
- 依靠手動報告進行管理，將會導致延遲和潛在的誤差。
- 營運團隊經常因事件發生期間的狀態更新而遇到服務中斷。

建立此最佳實務的優勢：

- 讓利益相關者能夠立即存取關鍵資訊，推動明智決策。
- 降低營運效率不彰的方法是將手動報告和經常狀態查詢次數降至最低。
- 提高透明度和信任的方法是能夠即時掌握系統效能和業務指標。

未建立此最佳實務時的曝險等級： 中

實作指引

儀表板能有效傳達系統狀態和業務指標，並可量身打造能滿足不同對象群組需求的內容。包括 Amazon CloudWatch 儀表板與 Amazon QuickSight 等工具能協助您建立互動式的即時儀表板，用於進行系統監控和商業智慧。

實作步驟

1. 確定利益相關者需求： 確定不同對象群組的特定資訊需求，例如技術團隊、領導層和客戶。

2. 選擇適當的工具： 選擇適當的工具，例如 Amazon CloudWatch 儀表板 用於系統監控和 Amazon QuickSight 用於互動式商業智慧。
3. 設計有效的儀表板：
   - 設計儀表板，使其能清楚呈現相關指標和 KPI，確保這類資訊易於理解和可採取動作。
   - 視需要合併系統層級和業務層級視觀圖。
   - 同時包括高層 (範圍較廣概觀) 和低層 (用於詳細分析) 儀表板。
   - 整合在儀表板中的多種自動警報，強調關鍵問題。
   - 註釋儀表板時搭配重要的指標臨界值及目標，可以隨時呈現指標資料。
4. 整合資料來源：
   - 使用 Amazon CloudWatch 彙總和顯示來自各種 AWS 服務的指標、 查詢來自其他資料來源的指標，最後建立系統執行狀況和業務指標的整合檢視。
   - 使用功能，例如 CloudWatch Logs Insights 查詢和視覺化來自不同應用程式和服務的日誌資料。
5. 提供自助服務存取：
   - 與相關利益相關者共用 CloudWatch 儀表板，讓自助資訊能從 儀表板共用功能進型存取。
   - 確保儀表板易於存取，並提供即時最新資訊。
6. 定期更新和完善：
   - 持續更新和完善儀表板，以符合不斷變化的業務需求和利益相關者意見回饋。
   - 定期檢閱儀表板，以確保與必要資訊的相關性，以及達到有效傳輸。

資源

相關的最佳實務：

- OPS08-BP05 建立儀表板

相關文件：

- 建置用於檢視營運能見度的儀表板
- 使用 Amazon CloudWatch 儀表板
- 使用儀表板變數建立靈活的儀表板
- 共用 CloudWatch 儀表板
- 查詢來自其他資料來源的指標
- 新增自訂小工具至 CloudWatch 儀表板

相關範例：

- One Observability 研討會 - 儀表板

OPS10-BP07 自動回應事件

達到快速、一致且無錯誤營運處理的關鍵是自動化事件回應。建立簡化程序，並使用工具來自動管理和回應事件，並將手動介入降到最低，並提高營運效率。

預期成果：

- 透過自動化，減少人為錯誤並縮短解決時間。
- 一致且可靠的營運事件處理。
- 提升營運效率和系統可靠性。

常見的反模式：

- 手動事件處理會導致延遲和錯誤。
- 自動化在重複關鍵任務中遭到忽略。
- 重複的手動工作會導致提醒疲勞並遺失重要問題。

建立此最佳實務的優勢：

- 加速事件回應，減少系統停機時間。
- 透過自動且一致的事件處理，達到可靠營運。

未建立此最佳實務時的曝險等級： 中

實作指引

納入自動化以建立有效營運工作流程，並將手動介入降到最低。

實作步驟

1. 識別自動化機會： 判斷重複的自動化工作，例如問題修復、票證增強功能、容量管理、擴展、部署和測試。
2. 識別自動化命令：
   - 評估並定義可啟動自動回應的特定條件或指標，過程中搭配 Amazon CloudWatch 警報動作。

- 使用 Amazon EventBridge 以回應 AWS 服務、自訂工作負載和 SaaS 應用程式中的事件。
- 考慮啟動事件，例如 特定日誌項目、效能指標臨界值，或 狀態變化 (關於 AWS 資源)。

3. 實作事件驅動型自動化：
   - 使用 AWS Systems Manager 自動化執行手冊，簡化維護、部署和修復工作。
   - 建立事件於 Incident Manager 自動收集，並將相關 AWS 資源的詳細資訊新增到事件中。
   - 主動監控配額，透過使用 AWS 的配額監視器。
   - 自動調整容量，搭配 AWS Auto Scaling 以維持可用性和效能。
   - 自動化開發管道，搭配 Amazon CodeCatalyst。
   - 煙霧測試，或持續監控端點和 API， 過程中使用綜合監控。
4. 透過自動化執行風險緩解：
   - 實作 自動化安全性回應 迅速解決風險。
   - 使用 AWS Systems Manager 狀態管理員 以減少組態偏離。
   - 使用 AWS Config 規則 修復不合規的資源。

實作計劃的工作量： 高

資源

相關的最佳實務：

- OPS08-BP04 建立可付諸行動的警示
- OPS10-BP02 每個提醒建立一個程序

相關文件：

- 使用系統管理員自動化執行手冊搭配 Incident Manager
- 在 Incident Manager 中建立事件
- AWS Service Quotas
- 監控資源用量，並在接近配額時傳送通知
- AWS Auto Scaling
- 什麼是 Amazon CodeCatalyst？
- 使用 Amazon CloudWatch 警報
- 使用 Amazon CloudWatch 警報動作

- 依 AWS Config 規則 修補不合規的資源
- 使用篩選條件從日誌事件建立指標
- AWS Systems Manager 狀態管理員

相關影片：

- 使用 AWS Systems Manager 建立自動化執行手冊
- 如何自動化 AWS 上的 IT 營運
- AWS Security Hub 自動化規則
- 利用 Amazon CodeCatalyst 藍圖快速開始您的軟體專案

相關範例：

- Amazon CodeCatalyst 教學：使用現代三層 Web 應用程式藍圖建立專案
- One Observability 研討會
- 使用 Incident Manager 回應事件

演進

問題

- OPS 11.如何改善營運？

OPS 11.如何改善營運？

投入時間和資源，盡量持續逐漸改善，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 執行事件後分析
- OPS11-BP03 實作回饋迴圈
- OPS11-BP04 執行知識管理
- 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

根據內部和外部架構最佳實務評估您的工作負載。經常特意進行工作負載審查。根據您的軟體開發步調制定改進機會的優先順序。

期望的結果：

- 您經常根據架構最佳實務分析工作負載。
- 在軟體開發過程中，您給予與功能同等優先順序的改善機會。

常見的反模式：

- 您在數年前部署工作負載後，即未對其執行過架構審查。
- 您降低改善機會的優先順序。與新功能相比，這些機會仍然保留在待辦項目中。
- 沒有對組織的最佳實務實作修改的標準。

建立此最佳實務的優勢：

- 您的工作負載依據架構最佳實務保持在最新狀態。
- 您特意逐步調整工作負載。
- 您可以利用組織最佳實務來改進所有工作負載。
- 您獲得的邊際收益會產生累積影響，進而提高效率。

未建立此最佳實務時的風險暴露等級：高

實作指引

經常對工作負載執行架構審查。使用內部和外部最佳實務，評估您的工作負載並識別改進機會。根據您的軟體開發步調制定改進機會的優先順序。

實作步驟

1. 以商定的頻率對生產工作負載進行定期架構審查。使用包含 AWS 特定最佳實務的已記載架構標準。
   a. 將您內部定義的標準用在這些審查上。如果您沒有內部標準，請使用 AWS Well-Architected Framework。
   b. 使用 AWS Well-Architected Tool 來建立內部最佳實務的自訂聚焦，並執行架構審查。
   c. 聯絡您的 AWS 解決方案架構師或技術客戶經理，對您的工作負載執行引導式的 Well-Architected Framework 審查。
2. 在您的軟體開發程序中，為在審查期間找出的改進機會制定優先順序。

實作計畫的工作量：低。您可以使用 AWS Well-Architected Framework 執行年度架構審查。

資源

相關的最佳實務：

- OPS11-BP02 執行事件後分析
- OPS11-BP08 記錄和分享獲得的經驗
- OPS04 實作可觀測性

相關文件：

- AWS Well-Architected Tool - 自訂聚焦
- AWS Well-Architected 白皮書 - 審查程序
- 使用自訂聚焦和 AWS Well-Architected Tool 自訂 Well-Architected 審查
- 在您的組織中實作 AWS Well-Architected Custom Lens 生命週期

相關影片：

- Well-Architected 實驗室 - Level 100：AWS Well-Architected Tool 上的自訂聚焦
- AWS re:Invent 2023 - 在整個組織中擴展 AWS Well-Architected 最佳實務

相關範例：

- AWS Well-Architected Tool

OPS11-BP02 執行事件後分析

檢閱致使客戶受到影響的事件，並識別問題成因和預防性措施。使用此資訊來制定可限制或防止事件再次發生的緩解措施。制定可快速有效回應的程序。適當傳達事件的根本原因與修正措施，內容針對目標對象量身打造。

期望的結果：

- 您已建立包括事件後分析的事件管理程序。
- 您已實施可觀測性計畫以收集事件相關資料。
- 有了這些資料，您可以了解並收集能支援事件後分析程序的指標。
- 您會從事件中學習經驗，進一步改善未來成果。

常見的反模式：

- 您會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。這也會在您還原遺失資料時帶來額外的操作負擔。

建立此最佳實務的優勢：

- 您所預先定義的程序可以判斷造成事件發生的元件、條件、措施和事件，協助您找出改進機會。
- 您可以使用事件後分析的資料來進行改善。

未建立此最佳實務時的風險暴露等級：高

實作指引

使用程序來判斷事件成因。檢閱所有致使客戶受到影響的事件。建立程序來識別和記錄事件的成因，這樣您就能制定緩解措施，達到限制或防止事件再次發生，而且您可以制定能快速有效回應的程序。適當傳達事件的根本原因，並針對目標對象量身打造通訊內容。在您的組織內公開分享學習成果。

實作步驟

1. 收集像是部署變更、設定變更、事件開始時間、警報時間、人員介入時間、緩解開始時間和事件解決時間等指標。
2. 描述在時間軸上的關鍵時間點，以便目標對象了解事件發生的相關活動。
3. 提出下列問題：
   a. 您是否能改善偵測時間？
   b. 指標和警報是否有任何更新而能盡快偵測出事件？
   c. 您是否能改善診斷時間？
   d. 您的回應計畫或升級計畫是否有更新而能讓適當回應人員盡快介入？
   e. 您是否能改善緩解時間？
   f. 是否有任何執行手冊或教戰手冊步驟可讓您新增或加以改進？
   g. 您是否能防止未來事件發生？
4. 建立檢查清單和措施。追蹤與交付所有措施。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS11-BP01 建立持續改進程序
- OPS 4 - 實作可觀測性

相關文件：

- 在 Incident Manager 中執行事件後分析
- 營運準備度檢閱

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 執行事件後分析：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 執行知識管理

知識管理可協助團隊成員尋找資訊以執行其作業。在學習組織中，資訊是任意共用的，助個人一臂之力。資訊可以探索和搜尋。資訊是準確且最新的。存在機制以建立新資訊、更新現有資訊，以及封存過時資訊。最常見的知識管理平台範例是內容管理系統，例如 Wiki。

預期成果：

- 團隊成員可以存取及時、準確的資訊。

- 資訊是可搜尋的。
- 存在機制以新增、更新和封存資訊。

常見的反模式：

- 沒有集中式知識儲存。團隊成員會在他們的本機電腦上管理他們自己的備註。
- 您有自我託管的 Wiki，但是沒有管理資訊的機制，導致資訊過時。
- 某人識別遺漏的資訊，但是沒有要求在團隊 Wiki 中新增它的程序。他們自行新增，但是遺漏關鍵步驟，導致中斷。

建立此最佳實務的優勢：

- 因為資訊任意共用，所以團隊成員握有能力。
- 因為文件是最新的且可搜尋，所以新的團隊成員可以更快上線。
- 資訊是及時、準確且可行的。

未建立此最佳實務時的風險暴露等級：高

實作指引

知識管理是學習組織的重要面向。若要開始，您需要集中儲存庫來存放您的知識 (常見的範例是自我託管的 Wiki)。您必須開發新增、更新和封存知識的程序。開發應該記載哪些項目的標準，並且讓所有人做出貢獻。

客戶範例

AnyCompany Retail 託管內部 Wiki，在其中存放所有知識。團隊成員受到鼓勵在他們執行每日職責時新增至知識庫。跨功能團隊每季會評估哪些頁面最少更新，並且判斷它們是否應該封存或更新。

實作步驟

1. 從識別存放知識所在的內容管理系統開始。跨組織取得利害關係人的協議。
   a. 如果您沒有現有內容管理系統，請考慮執行自我託管 Wiki 或使用版本控制儲存庫做為起點。
2. 開發新增、更新和封存資訊的執行手冊。向您的團隊教育這些程序。
3. 識別哪些知識應該存放在內容管理系統中。從團隊成員執行的每日活動 (執行手冊和程序手冊) 開始。與利害關係人合作來排列新增知識的優先順序。
4. 定期與利害關係人合作來識別過時資訊並且將它封存或更新。

實作計劃的工作量：中。如果您沒有現有內容管理系統，您可以設定自我託管 Wiki 或版本控制文件儲存庫。

資源

相關的最佳實務：

- OPS11-BP08 記錄和分享獲得的經驗 - 知識管理可促進所學習課程的資訊共用。

相關文件：

- Atlassian - 知識管理

相關範例：

- DokuWiki
- Gollum
- MediaWiki
- Wiki.js

OPS11-BP05 定義改進驅動因素

確定改進驅動因素，以幫助您根據資料和回饋迴圈評估改進機會，並排定優先順序。探索系統和流程中的改進機會，並在適當時機進行自動化。

期望的結果：

- 您可以追蹤整個環境的資料。
- 您可以將事件和活動與業務成果相關聯。
- 您可以在環境和系統之間作比較和對比。
- 您可以維護部署和結果的詳細活動歷史記錄。
- 您收集資料以支援您的安全狀態。

常見的反模式：

- 您從整體環境收集資料，但不將事件和活動相關聯。

- 您從全部資產收集詳細資料，因而提高 Amazon CloudWatch 和 AWS CloudTrail 活動和成本。但是，您不會以有意義的方式使用這些資料。
- 在定義改進的驅動因素時，您未考慮業務成果。
- 您未衡量新功能的效果。

建立此最佳實務的優勢：

- 藉由確定改進的條件，您將事件型動機或情緒式投資的影響降到最低。
- 您回應商業事件，而不僅是技術事件。
- 您衡量環境以確定需要改進的地方。

未建立此最佳實務時的風險暴露等級：中

實作指引

- 了解改進驅動因素：僅在理想結果受支援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、錯誤和漏洞。追蹤調整大小的選項，並尋找最佳化機會。
    - AWS 安全佈告欄
    - AWS Trusted Advisor
    - Cloud Intelligence Dashboards
  - 合規要求：在審查改進機會時，評估保持對法規、政策的合規性或保持受到第三方支援所需的更新和變更。
    - AWS 合規
    - AWS 合規計畫
    - AWS 合規最新消息

資源

相關的最佳實務：

- OPS01 組織優先事項

- OPS02 關係和擁有權
- OPS04-BP01 識別關鍵績效指標
- OPS08 利用工作負載可觀測性
- OPS09 了解運作狀態
- OPS11-BP03 實作回饋迴圈

相關文件：

- Amazon Athena
- Amazon QuickSight
- AWS 合規
- AWS 合規最新消息
- AWS 合規計畫
- AWS Glue
- AWS 安全佈告欄
- AWS Trusted Advisor
- 將日誌資料匯出至 Amazon S3
- AWS 最新消息
- 客戶至上的創新必要性
- 數位轉型：炒作或策略性需要？

相關影片：

- AWS re:Invent 2023 - 透過 AWS Support (SUP310) 提高營運效率和恢復能力

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。

期望的結果：

- 您可以定期與企業主審查洞見。企業主為新獲得的洞見提供額外的背景資訊。
- 您可以審查洞見並要求技術同仁提供意見回饋，然後在團隊之間分享您的學習心得。

- 您可以發布資料和洞見，供其他技術和業務團隊檢閱。您在策劃其他部門的新做法時，將自己的學習心得納入考量。
- 與資深領導者一起總結並審查新洞見。資深領導者利用新的洞見來擬定策略。

常見的反模式：

- 您發布了一項新功能。此功能會改變某些客戶行為。您的可觀測性不會考慮這些變更。您不會量化這些變更的好處。
- 您推送新的更新並疏於重新整理 CDN。CDN 快取不再與最新版本相容。您可以衡量含錯誤請求的百分比。您的所有使用者都在與後端伺服器通訊時報告 HTTP 400 錯誤。您調查用戶端錯誤，發現由於測錯維度，而造成時間的浪費。
- 您的服務層級協議規定 99.9% 的正常運作時間，而您的復原點目標是四小時。服務擁有者堅決表示系統達到了零停機時間。您實作昂貴且複雜的複寫解決方案，浪費時間和金錢。

建立此最佳實務的優勢：

- 當與企業擁有者和領域專家驗證洞見時，您可以建立共識並更有效地引導改進。
- 您發現隱藏的問題，並將這些問題納入對未來決策的考量。
- 您將焦點從技術成果轉移到業務成果。

未建立此最佳實務時的風險暴露等級：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。

資源

相關的最佳實務：

- OPS01-BP06 在管理效益和風險的同時評估權衡
- OPS02-BP06 團隊之間的責任是預先定義或經過協商的
- OPS11-BP03 實作回饋迴圈

相關文件：

- 設計雲端卓越中心 (CCOE)

相關影片：

- 建置可觀測性以強化恢復能力

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。

期望的結果：

- 您經常審查影響業務的指標
- 您可以透過可觀測性能力偵測和審查異常狀況
- 您使用資料來支援業務成果和目標

常見的反模式：

- 您的維護時段造成重要的零售促銷活動中斷。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。
- 由於您在組織中常用過時的資源庫，因此長期遭受停機之苦。之後您便遷移到受支援的資源庫。組織中的其他團隊不知道他們正面臨風險。
- 您不會定期審查客戶 SLA 達成率。您有不符合客戶 SLA 的傾向。若不符合客戶 SLA，會產生相關的財務罰責。

建立此最佳實務的優勢：

- 當您定期召開會議以審查作業指標、事件和事故時，您可以在團隊之間保持共識。
- 您的團隊會定期開會，審查指標和事件，所以您能夠針對風險適時採取行動，並識別客戶 SLA。
- 您可以分享學到的教訓，為業務成果的優先順序和有目標性的改進提供所需的資料。

未建立此最佳實務時的風險暴露等級：中

實作指引

- 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。

資源

相關的最佳實務：

- OPS08-BP05 建立儀表板
- OPS09-BP03 檢閱營運指標並優先改進
- OPS10-BP01 使用程序進行事件、事故和問題管理

相關文件：

- Amazon CloudWatch
- Amazon CloudWatch 指標和維度參考
- 發布自訂指標
- 使用 Amazon CloudWatch 指標
- CloudWatch 儀表板和視覺化功能

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，以便您在內部和跨團隊加以使用。您應分享您的團隊獲得的經驗，以提高整個組織的效益。分享資訊和資源，以防止可避免的錯誤和簡化開發工作，並專注於交付所需的功能。

使用 AWS Identity and Access Management (IAM) 定義許可權，允許以受控方式存取您希望在帳戶內和帳戶間共享的資源。

期望的結果：

- 您可以使用版本控制的儲存庫來分享應用程式庫、執行指令碼的程序、程序文件及其他系統文件。
- 您還將基礎設施標準作為版本控制的 AWS CloudFormation 範本分享。
- 您可以回顧跨團隊學到的教訓。

常見的反模式：

- 由於您的組織通常使用錯誤資源庫，致使您的組織遭受長期停機之苦。之後您便遷移到可靠的資源庫。組織中的其他團隊不知道他們正面臨風險。沒有人記錄和分享使用這個資源庫的經驗，而且他們不知道風險的存在。
- 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。
- 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。

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。

未建立此最佳實務時的風險暴露等級：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

資源

相關的最佳實務：

- OPS02-BP06 團隊之間的責任是預先定義或經過協商的
- OPS05-BP01 使用版本控制
- OPS05-BP06 共用設計標準
- OPS11-BP03 實作回饋迴圈
- OPS11-BP07 執行營運指標審查

相關文件：

- 使用文件即程式碼解決方案減少專案延遲

相關影片：

- 委託存取您的 AWS 環境
- AWS Support 為您提供支援 | 探索事故管理桌上模擬演練

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。

期望的結果：

- 您可以建立臨時環境複本，進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，以及測試計畫的改善。
- 您執行「演練日」，並使用故障注入服務 (FIS) 提供團隊在類似生產環境中執行實驗所需的控制和防護機制。

常見的反模式：

- 您的應用程式伺服器存在已知的效能問題。將問題加入到每個規劃功能實作的待辦項目中。如果規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。改進永遠不會有完成的一天。
- 作業驗收已完成，所以您未再進行測試。

建立此最佳實務的優勢：透過在程序中投入時間和資源，您可以持續、逐漸地做出改善。

未建立此最佳實務時的風險暴露等級：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、逐漸地做出改善。
- 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，且改進仍然是優先事項，則應尋求替代行動方案。
- 在演練日期間模擬生產工作負載，並將從這些模擬學得的經驗用於改進。

資源

相關的最佳實務：

- OPS05-BP08 使用多個環境

相關影片：

- AWS re:Invent 2023 - 利用 AWS 故障注入服務提高應用程序恢復能力

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。您可以在下列白皮書中找到規範指引： 安全支柱白皮書。

最佳實務領域

- 安全基礎
- 身分和存取管理
- 偵測
- 基礎設施保護
- 資料保護
- 事故回應
- 應用程式安全

安全基礎

問題

- SEC 1.如何安全地操作工作負載？

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，讓您能夠擴展安全操作。

最佳實務

- SEC01-BP01 使用帳戶區隔工作負載

- SEC01-BP02 保護帳戶根使用者和屬性
- SEC01-BP03 識別和驗證控制目標
- SEC01-BP04 隨時掌握安全威脅和建議的最新資訊
- SEC01-BP05 縮小安全管理範圍
- SEC01-BP06 自動部署標準安全控制措施
- SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施
- SEC01-BP08 定期評估和實作新的安全服務和功能

SEC01-BP01 使用帳戶區隔工作負載

透過多帳戶策略在環境 (例如生產、開發和測試) 與工作負載之間建立共通的防護機制和隔離。強烈建議帳戶層級的區隔，因為這在安全性、帳單和存取方面提供了有力的隔離界限。

預期成果：將雲端作業、不相關的工作負載和環境隔離成不同帳戶的帳戶結構，以提高雲端基礎設施間的安全性。

常見的反模式：

- 將多個具有不同資料敏感度等級且不相關的工作負載置於相同的帳戶中。
- 定義不良的組織單位 (OU) 結構。

建立此最佳實務的優勢：

- 若工作負載遭到意外存取，縮小影響範圍。
- 集中管控對 AWS 服務、資源和區域的存取。
- 利用政策以及集中管理安全服務，維護雲端基礎設施的安全性。
- 自動化帳戶建立和維護流程。
- 集中稽核您的基礎設施以滿足合規性和法規需求。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 帳戶 在以不同的敏感度等級操作的工作負載或資源之間提供安全隔離界限。AWS 提供工具透過多帳戶策略大規模管理您的雲端工作負載，以利用此隔離界限。如需有關 AWS 上多帳戶策略的概念、模式和實作的指引，請參閱使用多個帳戶管理您的 AWS 環境。

當您集中管理多個 AWS 帳戶 時，應該將您的帳戶組織成由組織單位 (OU) 層定義的階層。接著可以組織安全控制並套用至 OU 和成員帳戶，在組織內的成員帳戶上建立一致的預防性控制。安全控制是繼承的，讓您能夠篩選位於 OU 階層較低層級的成員帳戶可用的許可。良好的設計可利用此繼承關係來降低必要的安全政策數目和複雜度，達成每個成員帳戶預期的安全控制。

您可以使用 AWS Organizations 和 AWS Control Tower 這兩個服務來實作和管理在 AWS 環境中的多帳戶結構。AWS Organizations 可讓您將帳戶組織成由一或多個 OU 層所定義的階層，各個 OU 包含數個成員帳戶。服務控制政策 (SCP) 可讓組織管理員於成員帳戶建立細微的預防性控制，而 AWS Config 可用來於成員帳戶建立主動式和偵測控制。許多 AWS 服務皆與 AWS Organizations 整合以提供委派的管理控制，並跨組織內的所有成員帳戶執行服務特定的工作。

位於 AWS Organizations 分層之上的 AWS Control Tower 透過登陸區域為多帳戶 AWS 環境提供了一鍵式最佳實務設定。該登陸區域是通往由 Control Tower 所建立之多帳戶環境的進入點。Control Tower 提供數項優於 AWS Organizations 的優點。提供改進的帳戶管控的三個優點是：

- 整合式強制性安全防護機制，會自動套用至獲准加入組織的帳戶。
- 選擇性防護機制，可針對指定 OU 集合開啟或關閉。
- AWS Control Tower Account Factory 提供帳戶的自動化部署，當中包含組織內部預先核准的基準和設定選項。

實作步驟

1. 設計組織單位結構：設計妥善的組織單位結構可減輕建立和維護服務控制政策及其他安全控制所需的管理負擔。您的組織單位結構應該與您的業務需求、資料敏感度和工作負載結構協調一致。
2. 為您的多帳戶環境建立登陸區域：登陸區域提供一致的安全和基礎設施，您的組織可以從該基礎迅速開發、啟動和部署工作負載。您可以使用定製的登陸區域或 AWS Control Tower 來協調您的環境。
3. 建立防護機制：透過您的登陸區域為您的環境實作一致的安全性防護機制。AWS Control Tower 提供可部署的強制性和選擇性控制清單。實作 Control Tower 時會自動部署強制性控制。檢閱強烈建議和選擇性控制清單，並實作符合您需求的控制。
4. 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) 只會傳播到您指定的區域。當使用 Control Tower 時可以透過主控台，或透過調整 AWS Organizations 中的 IAM 許可政策執行此動作。
5. 考慮 AWS CloudFormation StackSets：StackSets 可幫助您將資源 (包括 IAM 政策、角色和群組) 從核准的範本部署到不同的 AWS 帳戶 和區域中。

資源

相關的最佳實務：

- SEC02-BP04 利用集中式身分提供者

相關文件：

- AWS Control Tower
- AWS 安全稽核指導方針
- IAM 最佳實務
- 使用 CloudFormation StackSets 跨多個 AWS 帳戶 和區域佈建資源
- 組織常見問答集
- AWS Organizations 術語和概念
- 在 AWS Organizations 多帳戶環境中服務控制政策的最佳實務
- AWS 帳戶管理參考指南
- 使用多個帳戶整理您的 AWS 環境

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
- 使用 AWS Control Tower 建立和管控多個帳戶
- 為現有組織啟用 Control Tower

相關研討會：

- Control Tower Immersion Day

SEC01-BP02 保護帳戶根使用者和屬性

根使用者是 AWS 帳戶 中最具特權的使用者，對帳戶內的所有資源具備完整的管理存取權，並且在某些情況下，不受安全政策的限制。停用對根使用者的程式設計存取，為根使用者建立適當的控制，以及避免例行使用根使用者，可降低意外暴露根憑證及後續危及雲端環境的風險。

預期成果：保護根使用者有助於降低因誤用根使用者憑證而可能發生的意外或有意傷害的可能性。建立偵測控制也能在當使用根使用者採取動作時警告適當的人員。

常見的反模式：

- 將根使用者用於需要根使用者憑證以外的工作。
- 疏於定期測試緊急應變計劃以確認重大基礎設施、程序和人員在緊急情況下的運作情形。
- 僅考慮一般帳戶登入流程而疏於考慮或測試替代帳戶復原方法。
- 未將 DNS、電子郵件伺服器和電話提供者作為重要安全周邊的一部分來處理，因為其會用於帳戶復原流程。

建立此最佳實務的優勢：保護對根使用者的存取可建立信心，讓您知道帳戶中的動作受到控制和稽核。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 提供眾多工具來協助保護您的帳戶。然而，由於不會啟用當中部分的措施預設，您必須採取直接的行動加以實作。考慮將這些建議作為保護 AWS 帳戶 的基本步驟。實作這些步驟時，務必建立程序以持續評估和監視安全控制。

首次建立 AWS 帳戶 時，您是從一個對帳戶中所有 AWS 服務和資源具有完全存取權的身分開始。此身分就是所謂的 AWS 帳戶 根使用者。您可以使用您建立該帳戶所用的電子郵件地址和密碼，以根使用者的身分登入。由於 AWS 根使用者獲得的已提升存取權，您必須將 AWS 根使用者限用於執行特別需要它的工作。根使用者登入憑證必須受嚴密防護，並且您應該一律為 AWS 帳戶 根使用者啟用多重要素驗證 (MFA)。

除了一般驗證流程 (使用使用者名稱、密碼和多重要素驗證 (MFA) 裝置登入根使用者) 之外，還有帳戶復原流程會登入 AWS 帳戶 根使用者，而其能夠存取與您的帳戶相關聯的電子郵件地址和電話號碼。因此，保護傳送復原電子郵件的根使用者電子郵件帳戶以及與帳戶相關聯的電話號碼同樣也很重要。另外，對於與根使用者相關聯的電子郵件地址託管在相同 AWS 帳戶的電子郵件伺服器或網域名稱服務 (DNS) 資源上的情況，也要考慮可能的循環相依性。

使用 AWS Organizations 時，會有多個 AWS 帳戶，各自都有根使用者。將一個帳戶指定為管理帳戶，接著可以在該管理帳戶之下新增數層成員帳戶。優先保護您的管理帳戶根使用者後，再來處理成員帳戶根使用者。保護管理帳戶根使用者的策略可不同於成員帳戶根使用者，而且您可以對成員帳戶根使用者設立預防性安全控制。

實作步驟

以下是為根使用者建立控制的建議實作步驟。適用時，可交互參考 CIS AWS Foundations 基準版本 1.4.0 建議。除了這些步驟之外，請諮詢 AWS 最佳實務指導方針來保護您的 AWS 帳戶 和資源。

預防性控制

1. 為帳戶設定準確的聯絡資訊。
   a. 此資訊會用於遺失密碼復原流程、遺失 MFA 裝置帳戶復原流程，以及與您的團隊進行重大安全相關通訊。
   b. 使用由您的企業網域所託管的電子郵件地址 (最好是使用分發清單) 作為根使用者的電子郵件地址。使用分發清單而不是個人的電子郵件帳戶可對長期存取根帳戶提供額外的備援和持續性。
   c. 聯絡資訊上所列的電話號碼應該是針對此用途的專用安全電話。不應公布或與他人共用電話號碼。
2. 請勿為根使用者建立存取金鑰。若存在存取金鑰，請將其移除 (CIS 1.4)。
   a. 去除根使用者任何長期存留的程式設計憑證 (存取和秘密金鑰)。
   b. 若根使用者存取金鑰已存在，您應該將使用這些金鑰的程序轉換為從 AWS Identity and Access Management (IAM) 角色使用暫時存取金鑰，然後刪除根使用者存取金鑰。
3. 確定您是否需要儲存根使用者的憑證。
   a. 如果您使用 AWS Organizations 建立新成員帳戶，則成員帳戶上的根使用者的初始密碼會設為隨機值，並且不會向您公開。必要時，考慮使用 AWS 組織管理帳戶的密碼重設程序獲取對成員帳戶的存取權。
   b. 對於獨立 AWS 帳戶 或管理 AWS 組織帳戶，請考慮建立根使用者的憑證並安全存放。為根使用者啟用 MFA。
4. 在 AWS 多帳戶環境中為成員帳戶根使用者啟用預防性控制。
   a. 考慮為成員帳戶啟用不允許建立根使用者的根存取金鑰預防性防護機制。
   b. 考慮為成員帳戶啟用不允許根使用者的動作預防性防護機制。
5. 如果您需要根使用者的憑證：
   a. 使用複雜密碼。
   b. 為根使用者啟用多重要素驗證 (MFA)，尤其是 AWS Organizations 管理 (付款人) 帳戶 (CIS 1.5)。
   c. 考慮硬體 MFA 裝置以獲得彈性和安全性，因為一次性裝置可減少包含 MFA 代碼的裝置重複用於其他用途的可能性。確認定期更換使用電池的硬體 MFA 裝置。(CIS 1.6)
      - 若要為根使用者設定 MFA，請遵循啟用虛擬 MFA 或硬體 MFA 裝置的指示。

d. 考慮註冊多個 MFA 裝置以備用。 每個帳戶最多允許 8 個 MFA 裝置。
   - 請注意，為根使用者註冊一個以上的 MFA 裝置會自動停用 MFA 裝置遺失時復原帳戶的流程。

e. 請將密碼妥善保管，如果以電子方式儲存密碼，請考慮循環相依性。儲存密碼時，請勿以需要存取相同的 AWS 帳戶 來取得密碼的方式儲存。

6. 選擇性：考慮為根使用者建立定期密碼輪流排程。
   - 憑證管理最佳實務取決於您法規和政策需求。受 MFA 保護的根使用者不依賴把密碼當作單一驗證要素。
   - 定期變更根使用者密碼可降低意外洩露的密碼可能遭到誤用的可能性。

偵測控制

- 建立警示以偵測根憑證的使用 (CIS 1.7)。啟用 Amazon GuardDuty 將透過 RootCredentialUsage 發現結果監控並發出關於根使用者 API 憑證使用的通知。
- 評估並實作適用於 AWS Config 的 AWS Well-Architected 安全支柱合規套件中包含的偵測控制，或者若是使用 AWS Control Tower，Control Tower 內有提供強烈建議的控制。

操作指導

- 確定組織內誰應該存取根使用者憑證。
  - 使用雙人規則，如此沒有單獨一人可以存取所有必要的憑證和 MFA 來取得根使用者存取權。
  - 確認組織而不是單一個人持有對與帳戶相關聯的電話號碼和電子郵件別名 (用於密碼重設和 MFA 重設程序) 的控制權。
- 只在特殊情況下使用根使用者 (CIS 1.7)。
  - AWS 根使用者不可用於日常任務，即使管理任務也一樣。僅以根使用者身分登入執行需要根使用者的 AWS 任務。所有其他動作都應該由其他擔任適當角色的使用者執行。
- 定期檢查根使用者的存取權操作正常，以便在發生需要使用根使用者憑證的緊急情況之前，測試相關程序。
- 定期檢查與帳戶相關聯的電子郵件地址，以及替代聯絡人下所列的電子郵件地址有效。監控這些電子郵件收件匣，查看您可能接收的安全通知 <abuse@amazon.com>。另外確保與帳戶相關聯的任何電話號碼都有效。
- 準備事件回應程序以回應根帳戶誤用的情況。請參考 AWS 安全事件應變指南和安全支柱白皮書的事件應變一節中的最佳實務，取得更多有關為您的 AWS 帳戶 建立事件應變策略的資訊。

資源

相關的最佳實務：

- SEC01-BP01 使用帳戶區隔工作負載
- SEC02-BP01 使用強式登入機制
- SEC03-BP02 授予最低權限存取權
- SEC03-BP03 建立緊急存取程序
- SEC10-BP05 預先佈建存取權

相關文件：

- AWS Control Tower
- AWS 安全稽核指導方針
- IAM 最佳實務
- Amazon GuardDuty – 根憑證使用警示
- 透過 CloudTrail 監控根憑證使用的逐步指引
- 經核准可與 AWS 搭配使用的 MFA 權杖
- 在 AWS 上實作緊急存取
- 改進 AWS 帳戶中 10 大安全性項目
- 如果我發現 AWS 帳戶 中有未授權的活動該怎麼辦？

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
- 限制使用 AWS 根憑證，取自 AWS re:inforce 2022 – 使用 AWS 的安全最佳實務 IAM

相關範例和實驗室：

- 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。

預期成果：您的企業擁有明確定義的安全控制目標，且這些目標與您的合規要求一致。控制是透過自動化和政策實作和強制執行，並且針對其能否有效實現您的目標這點，持續接受評估。稽核人員會定期收到某個時間點和某段時間的有效性實證報告。

常見的反模式：

- 貴企業未充分了解可保證安全的法規要求、市場期望和產業標準
- 您的網路安全架構和控制目標與您的企業需求不符
- 控制的實施未能透過可衡量的方式與您的控制目標完全相符
- 您未使用自動化方式來報告控制的有效性

未建立此最佳實務時的風險暴露等級：高

實作指引

有許多常見的網路安全架構可作為您訂立安全控制目標的基礎。考慮貴企業的法規要求、市場期望和產業標準，以確定哪些架構最合乎所需。範例包括 AICPA SOC 2、HITRUST、PCI-DSS、ISO 27001 和 NIST SP 800-53。

針對已確定的控制目標，務必了解您使用的 AWS 服務如何幫助您實現這些目標。使用 AWS Artifact 來尋找與合乎您目標架構的文件和報告 (其中描述 AWS 所涵蓋的責任範圍，以及您其他責任範圍的指引)。如需進一步的服務特定指引，以了解其符合各種不同架構控制聲明的資訊，請參閱AWS 客戶合規指南。

定義達成目標的控制措施時，請使用預防性控制措施來編制實施，並使用偵測控制來自動執行緩解措施。使用服務控制政策 (SCP) 協助防止您的 AWS Organizations 中不合規的資源組態和動作。在 AWS Config 中實作規則，以監控並報告不合規的資源，然後在規則行為趨於穩定可信後，將其切換為強制執行模式。若要部署合乎您網路安全架構的預先定義和受管規則集，請考慮優先使用 AWS Security Hub 標準。建議您從 AWS 基礎服務最佳實務 (FSBP) 標準和 CIS AWS Foundations Benchmark 開始著手，以實施符合多種標準架構共同擁有的多個目標的控制措施。雖然 Security Hub 本質上沒有所需的控制偵測功能，但可以藉由使用 AWS Config 合規套件補足這方面的能力。

使用 AWS Global Security and Compliance Acceleration (GSCA) 團隊推薦的 APN 合作夥伴組合，即可在需要時獲得安全顧問、諮詢機構、證據收集和通報系統、稽核人員以及其他輔助服務的協助。

實作步驟

1. 評估常見的網路安全架構，並調整您的控制目標，以使其符合您選擇的架構。
2. 取得相關文件，了解您的架構使用 AWS Artifact 的指引和責任。了解合規的哪些部分需由 AWS 共同責任模式承擔，以及哪些部分屬於您的責任。
3. 使用 SCP、資源政策、角色信任政策及其他防護機制，以防止不合規的資源組態和動作。
4. 評估部署合乎您控制目標的 Security Hub 標準和 AWS Config 合規套件。

資源

相關的最佳實務：

- SEC03-BP01 定義存取需求
- SEC04-BP01 設定服務和應用程式記錄
- SEC07-BP01 了解您的資料分類機制
- OPS01-BP03 評估管控要求
- OPS01-BP04 評估合規要求
- PERF01-BP05 使用政策和參考架構
- COST02-BP01 根據貴組織的需求制定政策

相關文件：

- AWS 客戶合規指南

相關工具：

- AWS Artifact

SEC01-BP04 隨時掌握安全威脅和建議的最新資訊

透過監控業界威脅情報刊物和資料摘要以獲得新知，以便隨時掌握最新的威脅和緩解措施。評估根據最新威脅資料自動更新的受管服務產品。

預期成果：隨著產業刊物更新，隨時掌握最新威脅和建議的資訊。 在發現新威脅時，使用自動化技術偵測潛在的弱點和漏洞。您可以採取緩解措施對抗這些威脅。 您採用 AWS 服務，以自動掌握最新威脅情報的資訊。

常見的反模式：

- 沒有可靠且可反覆執行的機制，因而無法隨時掌握新威脅情報。
- 手動維護技術產品組合、工作負載和相依項的清單，而這些都需要人員審查才能得知是否有潛在的弱點和漏洞。
- 沒有既定的機制能夠將工作負載和相依項更新到可用的最新版本，因而無法取得已知的威脅緩解措施。

建立此最佳實務的優勢：利用威脅情報來源隨時掌握新資訊，即可盡量避免錯過可能影響業務的威脅態勢中發生的重大變化。 採用自動化的方式取代手動，以掃描、偵測和修復工作負載及其相依項中存在的潛在弱點或漏洞，如此就能幫助您事先預測並快速降低風險。 這有助於控制與修補漏洞相關的時間和成本。

未建立此最佳實務時的風險暴露等級：高

實作指引

檢閱值得信賴的威脅情報刊物，以掌握威脅態勢。 請參閱 MITRE ATT&CK 知識庫中的文件，了解已知的對手策略、技巧和程序 (TTP)。檢閱 MITRE 的通用漏洞披露 (CVE) 清單，以得知您採用的產品中有哪些已知的漏洞。透過 Open Worldwide Application Security Project (OWASP) 熱門的 OWASP 前 10 大專案，了解 Web 應用程式的重大風險。

透過 CVE 的 AWS 安全公告，隨時掌握 AWS 安全事件和建議的修復步驟。

為了減輕您隨時掌握新知的整體投入與負擔，請考慮使用 AWS 服務，這些服務會隨著時間自動納入新的威脅情報。 例如，Amazon GuardDuty 會隨時提供業界威脅情報的最新資訊，可用來偵測您帳戶內的異常行為和威脅特徵。Amazon Inspector 會自動將其用於連續掃描功能的 CVE 資料庫保持在最新狀態。 AWS WAF 和 AWS Shield Advanced 兩者都提供受管規則群組，這些群組會隨著新的威脅出現自動更新。

檢閱 Well-Architected 卓越營運支柱，了解自動機群管理和修補的資訊。

實作步驟

- 訂閱與您的業務和產業相關的威脅情報刊物更新。訂閱 AWS 安全公告。
- 考慮採用自動納入新威脅情報的服務，例如 Amazon GuardDuty 和 Amazon Inspector。
- 部署符合 Well-Architected 卓越營運支柱最佳實務的機群管理和修補策略。

資源

相關的最佳實務：

- SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施
- OPS01-BP05 評估威脅態勢
- OPS11-BP01 建立持續改進程序

SEC01-BP05 縮小安全管理範圍

判斷您是否可以使用 AWS 服務將某些控制措施的管理轉移至 AWS (受管服務)，藉此縮小安全範圍。這些服務有助於減少安全維護工作，例如基礎設施佈建、軟體設定、修補或備份。

預期成果：您為工作負載選取 AWS 服務時，會考慮安全管理的範圍。除了其他 Well-Architected 考量外，管理開銷和維護工作的成本 (總體擁有成本，亦即 TCO) 會根據您所選取服務的成本加以權衡。您會將 AWS 控制和合規文件納入您的控制評估和驗證程序中。

常見的反模式：

- 部署工作負載時，並未徹底了解您所選取服務的共用責任模式。
- 在虛擬機器上託管資料庫和其他技術時，未先行評估對等的受管服務。
- 比較受管服務選項時，未將安全管理工作納入虛擬機器上託管技術的總體擁有成本中。

建立此最佳實務的優勢：使用受管服務能夠減輕您管理營運安全控制措施的整體負擔，進而降低安全風險和總體擁有成本。若非如此，時間可能會花費在某些安全工作上，而無法轉投入其他為企業創造更多價值的工作。受管服務也可以藉由將某些控制要求轉移到 AWS，以縮小合規要求的範圍。

未建立此最佳實務時的風險暴露等級：中

實作指引

您可以透過多種方式在 AWS 上整合工作負載的元件。您在 Amazon EC2 執行個體上安裝和執行技術時，通常需要承擔最大部分的整體安全責任。為了協助您減輕操作特定控制的負擔，請找出可縮小您的共同責任模式範圍的 AWS 受管服務，並了解如何在現有架構中使用這些服務。範例包括使用 Amazon Relational Database Service (Amazon RDS) 部署資料庫、使用 Amazon Elastic Kubernetes Service (Amazon EKS) 或 Amazon Elastic Container Service (Amazon ECS) 協調容器，或使用無伺服器選項。建置新的應用程式時，請仔細考量哪些服務有助於減少實作和管理安全控制措施方面的時間和成本。

合規要求也是選取服務時的考量因素。受管服務可將某些合規要求轉移到 AWS。請與合規團隊討論，了解他們在稽核您操作和管理的服務方面，以及接受相關 AWS 稽核報告中控制聲明時面對的難度。您可以將 AWS Artifact 中找到的稽核成品提供給稽核人員或監管機構，作為 AWS 安全控制的證據。您也可以使用某些 AWS 稽核成品所提供的責任指南來設計您的架構，以及參考《AWS 客戶合規指南》。本指引可協助您確定應採取哪些額外的安全控制措施，以便支援系統的特定使用案例。

使用受管服務時，務必熟悉將其資源更新為較新版本的流程 (例如，更新 Amazon RDS 所管理資料庫的版本，或 AWS Lambda 函數的程式設計語言執行時期)。雖然受管服務可能會自動執行此操作，但設定更新的節奏並了解對操作的影響仍然是您的責任。AWS Health 這類工具可幫助您在整個環境追蹤和管理這些更新。

實作步驟

1. 評估可取代為受管服務的工作負載元件。
   a. 如果您要將工作負載移轉至 AWS，可在評估是否要主機轉換、重構、平台轉換、重新建置或取代您的工作負載時，考慮減少管理 (時間和費用) 和降低風險。有時候，在一開始遷移時的額外投資，長遠來看可能帶來大幅的節省。
2. 考慮實作受管服務，如 Amazon RDS，而非安裝和管理您自己的技術部署。
3. 使用 AWS Artifact 中的責任指引來協助您確定應針對工作負載採取的安全控制措施。
4. 將使用中的資源記錄起來，並隨時掌握新的服務和方法，以找出縮小範圍的新機會。

資源

相關的最佳實務：

- PERF02-BP01 選擇最適合您工作負載的運算選項
- PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區
- SUS05-BP03 使用受管服務

相關文件：

- AWS Health 的計劃性生命週期事件

相關工具：

- AWS Health
- AWS Artifact

- AWS 客戶合規指南

相關影片：

- 如何使用 AWS DMS 遷移到 Amazon RDS 或 Aurora MySQL 資料庫執行個體？
- AWS re:Invent 2023 - 使用 AWS Health 以大規模管理資源生命週期事件

SEC01-BP06 自動部署標準安全控制措施

在開發和部署整個 AWS 環境的標準安全控制措施時，採用現代 DevOps 實務。 使用基礎設施即程式碼 (IaC) 範本定義標準安全控制措施和組態、擷取版本控制系統中的變更、在 CI/CD 管道中測試變更，並將變更自動部署至您的 AWS 環境。

預期成果：IaC 範本會擷取標準化的安全控制措施，並將其送交至版本控制系統。 CI/CD 管道準備好偵測變更，並自動測試和部署您的 AWS 環境。 防護機制準備好在進行部署之前，先偵測範本中的組態錯誤並發出警示。 工作負載會部署到已採取標準控制措施的環境中。 團隊有權透過自助服務機制部署經核准的服務組態。 已制定安全的備份和復原策略來控制組態、指令碼和相關資料。

常見的反模式：

- 透過 Web 主控台或命令列介面，手動變更標準安全控制措施。
- 依賴個別工作負載團隊手動實作中央團隊定義的控制措施。
- 依賴中央安全團隊應工作負載團隊的要求部署工作負載層級的控制措施。
- 允許相同的個人或團隊開發、測試和部署安全控制自動化指令碼，而未能妥善區分職責，或適當地對其加以制衡。

建立此最佳實務的優勢：使用範本定義標準安全控制措施，可讓您使用版本控制系統追蹤和比較一段時間的變化。 使用自動化方式測試和部署變更可建立標準化和可預測性，提高成功部署的機會，並減少手動重複工作。 為工作負載團隊提供自助服務機制來部署經核准的服務和組態，可降低組態錯誤和濫用的風險。這也有助於讓團隊及早在開發過程中納入控制措施。

未建立此最佳實務時的風險暴露等級：中

實作指引

遵循 SEC01-BP01 使用帳戶區隔工作負載中所述的實務，您最終會有多個 AWS 帳戶，可用於您使用 AWS Organizations 管理的不同環境。 雖然這些環境和工作負載需要的安全控制措施可能各有不同，

但您可以將整個組織的某些安全措施標準化。 範例包括整合集中式身分提供者、定義網路和防火牆，以及設定用於儲存和分析日誌的標準位置。 同樣地，您可以使用基礎設施即程式碼 (IaC) 將同樣嚴謹的應用程式程式碼開發程序應用到基礎設施佈建，也可以使用 IaC 定義和部署標準安全控制措施。

務必盡可能以宣告的方式定義安全控制措施，例如在 AWS CloudFormation 中，並將其儲存在來源控制系統中。 使用 DevOps 實務來自動部署控制措施，以便更容易預測發行版本、使用 AWS CloudFormation Guard 等工具進行自動測試，以及偵測已部署控制措施與所需組態之間的偏差。 您可以使用 AWS CodePipeline、AWS CodeBuild 和 AWS CodeDeploy 等服務來建構 CI/CD 管道。請參考使用多個帳戶整理您的 AWS 環境中的指引，以便在自己的帳戶中設定這些服務，把其與其他部署管道分開來。

您也可以定義範本來將定義和部署 AWS 帳戶、服務和組態的程序標準化。 此技術可讓中央安全團隊管理這些定義，並透過自助服務方式將其提供給工作負載團隊。 實現這一點的方法之一是使用 Service Catalog，您可以在其中將範本發佈為產品，讓工作負載團隊能夠將這些產品納入自己的管道部署中。 如果您使用 AWS Control Tower，有些範本和控制措施可以作為您著手的起點。 Control Tower 還提供 Account Factory 功能，可讓工作負載團隊使用您定義的標準建立新的 AWS 帳戶。 此功能可讓工作負載團隊在需要時，不再依賴中央團隊核准和建立新帳戶。 您可能需要這些帳戶，以便根據如提供的功能、所處理資料的敏感性或其行為等原因，區隔不同的工作負載元件。

實作步驟

1. 決定如何在版本控制系統中儲存和維護範本。
2. 建立 CI/CD 管道以測試和部署您的範本。 定義測試以檢查是否有組態錯誤，以及範本是否符合您公司的標準。
3. 建立標準化範本目錄，供工作負載團隊根據您的需求部署 AWS 帳戶 和服務。
4. 針對控制組態、指令碼和相關資料實施安全的備份和復原策略。

資源

相關的最佳實務：

- OPS05-BP01 使用版本控制
- OPS05-BP04 使用建置和部署管理系統
- REL08-BP05 使用自動化部署變更
- SUS06-BP01 採用可快速導入永續性改進的方法

相關文件：

- 使用多個帳戶整理您的 AWS 環境

相關範例：

- 使用 Service Catalog、AWS Organizations 和 AWS Lambda 自動建立帳戶和佈建資源
- 使用 AWS Secrets Manager、AWS KMS 和 AWS Certificate Manager 加強 DevOps 管道並保護資料

相關工具：

- AWS CloudFormation Guard
- AWS 登陸區域加速器

SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施

This best practice was updated with new guidance on December 6, 2023.

執行威脅建模，為您的工作負載識別並保有潛在威脅及相關緩解措施的最新記錄。排定威脅的優先順序並調整安全控制緩解措施，以防止、偵測和回應威脅。就您的工作負載的情況，以及不斷演變的安全形勢，重新檢視和維護此工作。

未建立此最佳實務時的風險暴露等級：高

實作指引

什麼是威脅建模？

「威脅建模以保護有價值物為目標，識別、溝通和了解威脅及緩解措施。」– 開放 Web 應用程式安全專案 (OWASP) 應用程式威脅建模

為何使用威脅模型？

系統本身錯綜複雜，並且隨時間變得更形複雜且更具能力，而實現更大的商業價值及更高的客戶滿意度和參與度。這表示 IT 設計決策需要考慮不斷增加的使用案例數量。這種複雜性和使用案例數量的排列通常使得非結構化方法無法有效尋找和緩解威脅。反之，您需要一套系統化方法來列舉對系統的潛在威脅，以及策畫緩解措施，並以這些緩解措施為優先來確保組織的有限資源能在改善系統整體安全形勢上發揮最大的影響力。

威脅建模旨在提供這套系統化方法，目的是要在設計過程中及早尋找和解決問題，此時進行緩解的成本和精力與生命週期稍後相比要來得低。此方法與往前移安全性的業界原則相一致。威脅建模最終會與組織的風險管理程序整合，透過使用威脅驅動的方法，協助推動要實作哪些控制措施的決策。

何時執行威脅建模？

在工作負載的生命週期中及早開始威脅建模，可給予您更大的彈性來決定要如何處理所識別的威脅。就跟軟體錯誤一樣，越早識別威脅，就能以越具成本效益的方式加以解決。威脅模型是不斷更新的文件，並且應該持續隨著工作負載的變更而演進。隨時間重新檢視您的威脅模型，包括當有重大變更、威脅形勢有變化，或是採用新功能或服務時。

實作步驟

我們能如何執行威脅建模？

執行威脅建模的方式有很多種。就跟程式設計語言一樣，各有優缺點，而您應該選擇最適合您的方式。其中一個方法是從 Shostack 針對威脅建模的 4 個問題框架開始著手，當中提出自由回答的問題會為您的威脅建模練習提供結構：

1. 目前正在做什麼？

   此問題的目的是幫助您了解正在建置的系統並對之取得一致的意見，以及該系統與安全相關的細節。建立模型或圖表是回答此問題最受歡迎的方法，因為這可幫助您將正在建置的東西視覺化，例如使用資料流程圖。寫下關於您的系統的假設和重要細節也有助您定義涵蓋的範圍。這使得所有參與威脅模型的人能夠專注於相同的事物，並避免偏離至與主題無關的話題 (包括過時的系統版本) 而耗費時間。舉例來說，如果您正在建置 Web 應用程式，可能不值得花時間為瀏覽器用戶端建立作業系統信任開機順序的模型，因為您無法透過您的設計對此產生影響。

2. 什麼可能出錯？

   這是您識別對系統的威脅之處。威脅是意外或有意的動作或事件，會帶來不必要的衝擊，並且可能影響系統安全。對可能出錯之處沒有清楚的了解，便無法對症下藥。

   對於什麼可能出錯，您並沒有標準的清單可循。建立此清單需要團隊內的每個人與涉及的相關角色在威脅建模練習中集思廣益和共同協作。您可以使用識別威脅的模型來協助集思廣益，例如 STRIDE，這會建議不同的類別以進行評估：詐騙、竄改、否認性、資訊洩露、拒絕服務和提升權限。此外，您可能會想要檢閱現有的清單並研究以獲得靈感來協助集思廣益，包括 OWASP 前十大、HiTrust 威脅目錄，以及您組織本身的威脅目錄。

3. 我們要做何處理？

就跟前一個問題一樣，對於所有可能的緩解措施並沒有標準的清單可循。此步驟的輸入是前一步驟識別的威脅、動作和改進之處。

安全與合規是您與 AWS 之間的共同責任。了解當您提出「我們要做何處理？」時，也是在問「誰要對其負責？」，這一點很重要。了解您與 AWS 之間的責任制衡有助您將威脅建模練習的範圍定在您控制之下的緩解措施，這通常是 AWS 服務組態選項與您自身的系統特定緩解措施的組合。

對於共同責任的 AWS 部分，您將發現 AWS 服務在許多合規計畫的範圍之內。這些計畫會幫助您了解 AWS 在維護雲端安全和合規方面設立的強大控制措施。來自這些計畫的稽核報告可供 AWS 客戶從 AWS Artifact 下載。

無論您使用何種 AWS 服務，其始終涉及客戶責任，而您的威脅模型中應該包含與這些責任一致的緩解措施。對於 AWS 服務本身的安全控制緩解措施，您應該考慮跨領域實作安全控制，包括身分和存取管理 (驗證和授權)、資料保護 (靜態和傳輸中)、基礎結構安全、記錄和監控等領域。每個 AWS 服務的文件都有專屬的安全章節，提供將安全控制視為緩解措施的指引。重要的是，考慮您正在編寫的程式碼及其程式碼相依性，並思考您可以設立以解決該些威脅的控制措施。這些控制措施可以是輸入驗證、工作階段處理和界限處理等事項。大多數漏洞通常是在自訂程式碼中引入，因此請專注於此區域。

4. 我們處理得當嗎？

目標是讓您的團隊與組織改進威脅模型的品質以及隨時間執行威脅建模的速度。這些改進出自練習、學習、教導和評量的組合。若要更加深入並實際操作，建議您與您的團隊完成建置人員建立威脅模型的正確方式訓練課程或研討會。此外，如果您正在尋找有關如何將威脅建模整合至您組織的應用程式開發生命週期，請參閱 AWS 安全部落格上的如何進行威脅建模。

威脅編寫器

為了協助並指導您執行威脅建模，請考慮使用威脅編寫器工具，該工具旨在縮短威脅建模實現價值的時間。該工具可幫助您執行以下操作：

- 撰寫與威脅文法相符、可在自然非線性工作流程中使用的有用威脅陳述式
- 產生人類可讀的威脅模型
- 產生機器可讀的威脅模型，以便您能將威脅模型視為程式碼
- 使用洞察儀表板協助您快速識別品質和涵蓋範圍有所改進的領域

如需進一步的參考，請造訪「威脅編寫器」，並切換到系統定義的範例工作區。

資源

相關的最佳實務：

- SEC01-BP03 識別和驗證控制目標
- SEC01-BP04 隨時掌握安全威脅和建議的最新資訊
- SEC01-BP05 縮小安全管理範圍
- SEC01-BP08 定期評估和實作新的安全服務和功能

相關文件：

- 如何進行威脅建模 (AWS 安全部落格)
- NIST：以資料為中心的系統威脅建模指南

相關影片：

- AWS Summit ANZ 2021 - 如何進行威脅建模
- AWS Summit ANZ 2022 - 擴展安全性 – 針對快速和安全交付進行最佳化

相關訓練：

- 建置人員建立威脅模型的正確方式 – AWS Skill Builder 虛擬自訂進度訓練課程
- 建置人員建立威脅模型的正確方式 – AWS 研討會

相關工具：

- 威脅編寫器

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，幫助您推進工作負載的安全狀態。

預期成果：您擁有一套標準實務，可通知您 AWS 和 AWS 合作夥伴發行的新功能和服務。您會評估這些新功能對於環境和工作負載目前和新控制措施的設計有何影響。

常見的反模式：

- 您未訂閱 AWS 部落格和 RSS 摘要，因此未能迅速得知相關的新功能和服務
- 您依靠第二手來源得知安全服務和功能的最新消息和更新
- 您沒有鼓勵組織中的 AWS 使用者隨時掌握最新更新

建立此最佳實務的優勢：若您能隨時掌握新的安全服務和功能，就能在雲端環境和工作負載中實施控制措施方面做出明智的決策。這些來源有助於提高對於不斷變化的安全態勢的意識，以及說明如何使用 AWS 服務來防範新出現的威脅。

未建立此最佳實務時的風險暴露等級：低

實作指引

AWS 會透過幾種管道通知客戶新的安全服務和功能：

- AWS 最新消息
- AWS 新聞部落格
- AWS 安全部落格
- AWS 安全公告
- AWS 文件概觀

您可以使用 Amazon Simple Notification Service (Amazon SNS) 訂閱 AWS 每日功能更新主題，以獲得完整的每日更新摘要。某些安全服務 (例如 Amazon GuardDuty 和 AWS Security Hub) 會提供自己的 SNS 主題，以持續掌握與這些特定服務有關的新標準、調查結果和其他更新。

每年全球各地也會舉行多場會議、活動和網路研討會，於會中宣布並詳細描述新服務和功能。其中特別值得關注的是年度 AWS re:Inforce 安全會議，以及較為常態的 AWS re:Invent 會議。先前提到的 AWS 新聞頻道會分享這些有關安全和其他服務的會議公告，您可以在 YouTube 上的 AWS 活動頻道線上觀看更深入的教育性分組討論環節。

您也可以向您的 AWS 帳戶 團隊詢問有關最新安全服務更新和建議的資訊。如果沒有直接聯絡資訊，您可以透過銷售人員支援表單聯繫您的團隊。同樣地，如果您訂閱 AWS 企業支援，將會收到來自技術客戶經理 (TAM) 的每週更新，您可以與他們安排定期審查會議。

實作步驟

1. 使用您偏好的 RSS 閱讀器訂閱各種不同的部落格和公告，以及訂閱每日功能更新 SNS 主題。
2. 評估要參加哪些 AWS 活動，以獲得有關新功能和服務的第一手資訊。

3. 與您的 AWS 帳戶 團隊排定會議，以討論有關更新安全服務和功能的任何疑問。
4. 考慮訂閱企業支援，即可定期向技術客戶經理 (TAM) 諮詢。

資源

相關的最佳實務：

- PERF01-BP01 了解可用的雲端服務和特徵
- COST01-BP07 及時了解新的服務版本

身分和存取管理

問題

- SEC 2.如何管理人員和機器的身分驗證？
- 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分驗證？

處理操作安全的 AWS 工作負載時，您必須管理兩種身分類型。了解您必須管理和授予存取權的身分類型，有助於確認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，您可能也有設定在 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 使用臨時憑證
- SEC02-BP03 安全地存放和使用機密
- SEC02-BP04 利用集中式身分提供者
- SEC02-BP05 定期稽核和輪換憑證

- SEC02-BP06 採用使用者群組和屬性

SEC02-BP01 使用強式登入機制

登入 (使用登入憑證進行驗證) 可預防當沒有使用多重要素驗證 (MFA) 等機制時的風險，尤其是在登入憑證遭到意外洩露或輕易被猜出的情況下。使用強式登入機制透過要求 MFA 和強式密碼政策來降低這些風險。

預期成果：透過對 AWS Identity and Access Management (IAM) 使用者、AWS 帳戶根使用者、AWS IAM Identity Center (AWS 單一登入的後繼者) 和協力廠商身分提供者使用強式登入機制，來降低 AWS 中意外存取憑證的風險。這意味著要求使用 MFA，強制強式密碼政策，以及偵測異常的登入行為。

常見的反模式：

- 沒有為您的身分強制強式密碼政策，包括複雜密碼和 MFA。
- 在不同使用者之間共用相同的憑證。
- 沒有針對可疑的登入使用偵測控制。

未建立此最佳實務時的風險暴露等級：高

實作指引

人類身分登入 AWS 的方法有很多。AWS 最佳實務是仰賴使用聯合 (直接聯合或使用 AWS IAM Identity Center) 的集中式身分提供者向 AWS 進行驗證。在這種情況下，您應該以您的身分提供者或 Microsoft Active Directory 建立安全的登入程序。

當您第一次開啟 AWS 帳戶 時，是從 AWS 帳戶 根使用者開始。您應該只使用帳戶根使用者來設定使用者的存取權 (以及需要根使用者的任務)。請務必在開啟 AWS 帳戶 後使用 AWS 最佳實務指南立即為帳戶根使用者啟用 MFA。

如果您在 AWS IAM Identity Center 中建立使用者，請保護該服務中的登入程序。對於消費者身分，您可以使用 Amazon Cognito user pools 並保護該服務中的登入程序，或使用 Amazon Cognito user pools 支援的其中一個身分提供者。

如果您使用的是 AWS Identity and Access Management (IAM) 使用者，請使用 IAM 保護登入程序。

無論登入方法為何，強制強式登入政策必不可少。

實作步驟

以下是一般的強式登入建議。您設定的實際設定應由貴公司政策來規定，或使用如 NIST 800-63 的標準。

- 要求 MFA。IAM 最佳實務是對人類身分和工作負載要求 MFA。啟用 MFA 可提供多一層安全防護，要求使用者提供登入憑證和一次性密碼 (OTP)，或是從硬體裝置以密碼編譯方式驗證和產生的字串。
- 強制密碼長度下限，此為密碼強度的要素。
- 強制密碼複雜性，使密碼更難猜測。
- 允許使用者變更自己的密碼。
- 建立個別身分，而不是共用憑證。透過建立個別身分，您可以為每個使用者提供一組獨一無二的安全憑證。個別使用者可讓您稽核每個使用者的活動。

IAM Identity Center 建議：

- 當使用預設目錄來建立密碼長度、複雜性和重複使用需求時，IAM Identity Center 提供預先定義的密碼政策。
- 當身分來源為預設目錄、AWS Managed Microsoft AD 或 AD Connector 時，啟用 MFA 並為 MFA 設定內容感知或永遠開啟設定。
- 允許使用者註冊自己的 MFA 裝置。

Amazon Cognito user pools 目錄建議：

- 設定密碼強度設定。
- 對使用者要求 MFA。
- 針對調適性驗證 (這可封鎖可疑登入) 等功能使用 Amazon Cognito user pools 進階安全性設定。

IAM 使用者建議：

- 在理想情況下，您使用 IAM Identity Center 或直接聯合。然而，您可能需要 IAM 使用者。在這種情況下，請為 IAM 使用者設定密碼政策。您可以使用密碼政策來定義需求，例如最短長度或是密碼是否需要非字母字元等。
- 建立 IAM 政策以強制 MFA 登入，允許使用者管理自己的密碼和 MFA 裝置。

資源

相關的最佳實務：

- SEC02-BP03 安全地存放和使用機密
- SEC02-BP04 利用集中式身分提供者
- SEC03-BP08 在組織內安全地共用資源

相關文件：

- AWS IAM Identity Center (AWS 單一登入的後繼者) 密碼政策
- IAM 使用者密碼政策
- 設定 AWS 帳戶根使用者密碼
- Amazon Cognito 密碼政策
- AWS 憑證
- IAM 安全最佳實務

相關影片：

- 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分

SEC02-BP02 使用臨時憑證

當進行任何類型的驗證時，最好是使用臨時憑證，而不是長期憑證，以降低或消除風險，例如憑證遭到意外洩露、共用或遭竊。

預期成果：為了降低長期憑證的風險，對於人員和機器身分，請盡可能使用臨時憑證。長期憑證會產生許多風險，例如，可能在程式碼中將它們上傳至公有 GitHub 儲存庫。透過使用臨時憑證，您可大幅降低憑證遭到入侵的可能性。

常見的反模式：

- 開發人員使用取自 IAM users 的長期存取金鑰，而不是使用聯合從 CLI 取得臨時憑證。
- 開發人員將長期存取金鑰內嵌在程式碼中，並將該程式碼上傳到公有 Git 儲存庫。
- 開發人員將長期存取金鑰內嵌在行動應用程式中，之後在應用程式商店中提供該行動應用程式。
- 使用者與其他使用者共用長期存取金鑰，或是擁有長期存取金鑰的離職員工仍持有金鑰。
- 對機器身分可以使用臨時憑證時，卻使用長期存取金鑰。

未建立此最佳實務時的風險暴露等級：高

實作指引

對所有 AWS API 和 CLI 請求使用臨時安全憑證，而不是長期憑證。幾乎在任何情況下，對 AWS 服務的 API 和 CLI 請求都必須使用 AWS 存取金鑰簽署。您可以使用臨時或長期憑證簽署這些請求。您唯一應該使用長期憑證 (又稱為長期存取金鑰) 的時候是在使用 IAM 使用者或 AWS 帳戶 根使用者時。當您聯合至 AWS 或透過其他方法擔任 IAM 角色時，系統會產生臨時憑證。每當您使用登入憑證存取 AWS Management Console 時，系統會為您產生臨時憑證以呼叫 AWS 服務。在幾種情況下，您將需要長期憑證，並能夠使用臨時憑證完成幾乎所有任務。

避免使用長期憑證而改用臨時憑證，同時實行減少使用 IAM 使用者並支持聯合和 IAM 角色的策略。雖然對人類和機器身分過去以來都是使用 IAM 使用者，我們現在建議不要使用它們以避免使用長期存取金鑰的風險。

實作步驟

對於人類身分，例如員工、管理員、開發人員、操作員和客戶：

- 您應該仰賴集中式身分提供者並要求人類使用者以聯合搭配身分提供者，使用臨時憑證存取 AWS。您可以直接聯合至各個 AWS 帳戶 或使用 AWS IAM Identity Center (AWS IAM Identity Center 的後繼者) 和自選的身分提供者，為您的使用者進行聯合。與使用 IAM 使用者相比，聯合除了可消除長期憑證外，還提供一些優勢。您的使用者也可以從命令列進行直接聯合，或使用 IAM Identity Center 請求臨時憑證。這表示有少數使用案例會需要 IAM 使用者，或使用者需要長期憑證。
- 當授權讓第三方 (例如軟體即服務 (SaaS) 提供者) 存取 AWS 帳戶中的資源時，您可以使用跨帳戶角色和以資源為基礎的政策。
- 如果您需要授權應用程式供消費者或客戶存取您的 AWS 資源，您可以使用 Amazon Cognito 身分集區或 Amazon Cognito user pools 來提供臨時憑證。憑證的許可透過 IAM 角色設定。 您還可以對未驗證的訪客使用者另外定義一個具有限制許可的 IAM 角色。

對於機器身分，您可能需要使用長期憑證。在這些情況下，您應該要求工作負載使用臨時憑證，並以 IAM 角色存取 AWS。

- 對於 Amazon Elastic Compute Cloud (Amazon EC2)，您可以使用適用於 Amazon EC2 的角色。

- AWS Lambda 可讓您設定 Lambda 執行角色，授予服務許可以使用臨時憑證執行 AWS 動作。有許多其他類似的模型可供 AWS 服務使用 IAM 角色授予臨時憑證。
- 對於 IoT 裝置，您可以使用 AWS IoT Core 憑證提供者來請求臨時憑證。

- 對於內部部署系統或是在 AWS 之外執行並需要存取 AWS 資源的系統，您可以使用 IAM Roles Anywhere。

有些情況無法使用臨時憑證，而您可能需要使用長期憑證。在這些情況下，定期稽核和輪換憑證並針對需要長期憑證的使用案例定期輪換存取金鑰。有些可能需要長期憑證的例子包括 WordPress 外掛程式和第三方 AWS 用戶端。在您必須使用長期憑證的情況下，或是對於 AWS 存取金鑰以外的憑證，例如資料庫登入，您可以使用專為管理機密而設計的服務，例如 AWS Secrets Manager。Secrets Manager 方便您使用支援的服務管理、輪換和安全地儲存加密的機密。如需有關輪換長期憑證的詳細資訊，請參閱輪換存取金鑰。

資源

相關的最佳實務：

- SEC02-BP03 安全地存放和使用機密
- SEC02-BP04 利用集中式身分提供者
- SEC03-BP08 在組織內安全地共用資源

相關文件：

- 臨時安全憑證
- AWS 憑證
- IAM 安全最佳實務
- IAM 角色
- IAM Identity Center
- 身分提供者與聯合
- 輪換存取金鑰
- 安全合作夥伴解決方案：存取與存取控制
- AWS 帳戶根使用者

相關影片：

- 使用 AWS IAM Identity Center (AWS IAM Identity Center 的後繼者) 大規模管理使用者許可
- 在每一層都能掌握身分

SEC02-BP03 安全地存放和使用機密

工作負載需要能夠自動向資料庫、資源和第三方資源證明其身分。這需使用私密存取憑證來完成，例如 API 存取金鑰、密碼和 OAuth 權杖。使用專用服務來儲存、管理和輪換這些憑證有助於降低該些憑證遭到入侵的可能性。

預期成果：實施一種安全管理應用程式憑證的機制並達到以下目標：

- 識別工作負載需要何種機密。
- 盡可能以短期憑證取代長期憑證，來減少所需的長期憑證數目。
- 建立安全的存放區並自動輪換其餘的長期憑證。
- 稽核對存在於工作負載中的機密的存取。
- 持續監控以確認原始程式碼在開發過程中沒有內嵌機密。
- 降低憑證遭意外洩露的可能性。

常見的反模式：

- 沒有輪換憑證。
- 將長期憑證存放在原始程式碼或設定檔中。
- 未加密儲存靜態憑證。

建立此最佳實務的優勢：

- 已加密儲存靜態和傳輸中的機密。
- 透過 API 限制憑證的存取 (把這想成是憑證自動販賣機)。
- 稽核並記錄對憑證的存取 (包括讀寫)。
- 區隔顧慮：由不同的元件執行憑證輪換，而該元件可與其餘的架構分離。
- 自動將機密隨需散發到軟體元件並集中進行輪換。
- 可以精細的方式控制對憑證的存取。

未建立此最佳實務時的風險暴露等級：高

實作指引

以往，憑證用於向資料庫進行驗證，而第三方 API、權杖和其他機密可能內嵌在原始程式碼或環境檔案中。AWS 提供數種機制以安全存放這些憑證，自動輪換並稽核它們的使用情況。

著手機密管理的最佳方法是遵循移除、取代和輪換的指引。最安全的憑證是您不用存放、管理或處理的憑證。有些憑證對於工作負載的運作不再是必要的，故能夠安全移除。

對於工作負載適當運作仍舊是必要的憑證，可能有機會以臨時或短期憑證取代長期憑證。例如，與其對 AWS 私密存取金鑰進行硬式編碼，考慮使用 IAM 角色以臨時憑證取代長期憑證。

部分長期存留的機密可能無法移除或取代。可將這些機密存放在 AWS Secrets Manager 之類的服務中，進行集中存放、管理和定期輪換。

對工作負載的原始程式碼和設定檔的稽核，可能顯現多種類型的憑證。下表概述處理常見憑證類型的策略：

Credential type	Description	Suggested strategy
IAM access keys	AWS IAM access and secret keys used to assume IAM roles inside of a workload	Replace: Use IAM 角色 assigned to the compute instances (such as Amazon EC2 or AWS Lambda) instead. For interoperability with third parties that require access to resources in your AWS 帳戶, ask if they support AWS 跨帳戶存取權. For mobile apps, consider using temporary credentials through Amazon Cognito 身分集區 (聯合身分). For workloads running outside of AWS, consider IAM Roles Anywhere or AWS Systems Manager 混合式啟用.
SSH keys	Secure Shell private keys used to log into Linux EC2 instances, manually or as part of an automated process	Replace: Use AWS Systems Manager or EC2 執行個體連線 to provide programmatic and human access to EC2 instances using IAM roles.
Application and database credentials	Passwords – plain text string	Rotate: Store credentials in AWS Secrets Manager and

Credential type	Description	Suggested strategy
		establish automated rotation if possible.
Amazon RDS and Aurora Admin Database credentials	Passwords – plain text string	Replace: Use the Secrets Manager 與 Amazon RDS 整合 or Amazon Aurora. In addition, some RDS database types can use IAM roles instead of passwords for some use cases (for more detail, see IAM 資料庫身分驗證).
OAuth tokens	Secret tokens – plain text string	Rotate: Store tokens in AWS Secrets Manager and configure automated rotation.
API tokens and keys	Secret tokens – plain text string	Rotate: Store in AWS Secrets Manager and establish automated rotation if possible.

常見的反模式是將 IAM 存取金鑰內嵌在原始程式碼、設定檔或行動應用程式內。當需要 IAM 存取金鑰與 AWS 服務通訊時，請使用臨時 (短期) 安全憑證。這些短期憑證可以透過 IAM 角色 (用於 EC2 執行個體)、執行角色 (用於 Lambda 函數)、Cognito IAM 角色 (用於行動使用者存取)，以及 IoT Core 政策 (用於 IoT 裝置) 提供。當與第三方互動時，偏好委派 IAM 角色的存取權，包含對帳戶資源的必要存取權，而不是設定 IAM 使用者並將其的私密存取金鑰傳送給該第三方。

在很多情況下，工作負載需要儲存機密才能與其他服務和資源相互操作。AWS Secrets Manager 是專為安全管理這些憑證所打造的，可儲存和輪換 API 權杖、密碼和其他憑證。

AWS Secrets Manager 提供五項重要功能以確保敏感憑證的安全存放和處理：靜態加密、傳輸中加密、 全面性稽核、精細存取控制，以及可擴充的憑證輪換。來自 AWS 合作夥伴的其他機密管理服務，或本機開發並提供類似功能和保證的解決方案也可接受。

實作步驟

1. 使用 Amazon CodeGuru 等自動工具識別包含硬式編碼憑證的程式碼路徑。

- 使用 Amazon CodeGuru 掃描您的程式碼儲存庫。審閱完成後，在 CodeGuru 中篩選 `Type=Secrets` 以尋找有問題的程式碼行。

2. 識別可移除或取代的憑證。
   a. 識別不再需要的憑證並標示以進行移除。
   b. 對於內嵌在原始程式碼中的 AWS 機密金鑰，請使用與必要資源相關聯的 IAM 角色加以取代。如果您部分的工作負載位於 AWS 之外但需要 IAM 憑證存取 AWS 資源，請考慮 IAM Roles Anywhere 或 AWS Systems Manager 混合式啟用。
3. 對於其他第三方長期存留且需要使用輪換策略的機密，將 Secrets Manager 整合至程式碼中以在執行時間擷取第三方機密。
   a. CodeGuru 主控台可以使用已探索的憑證自動在 Secrets Manager 中建立機密。
   b. 將 Secrets Manager 的機密擷取整合至您的應用程式程式碼中。
      - 無伺服器 Lambda 函數可以使用語言中立的 Lambda 延伸。
      - 對於 EC2 執行個體或容器，AWS 提供範例用戶端程式碼，可以數種熱門的程式設計語言從 Secrets Manager 擷取機密。
4. 定期審閱您的程式碼基底並重新掃描，以確認程式碼中未加入新的機密。
   - 考慮使用 git-secrets 之類的工具以防將新機密認可到您的原始程式碼儲存庫。
5. 監控 Secrets Manager 活動以尋找非預期使用、不當私密存取或嘗試刪除機密的跡象。
6. 減少對憑證的人員接觸。將讀寫和修改憑證的存取權限於專門用於此用途的 IAM 角色，並且只將擔任該角色的存取權提供給一小組可操作的使用者子集。

資源

相關的最佳實務：

- SEC02-BP02 使用臨時憑證
- SEC02-BP05 定期稽核和輪換憑證

相關文件：

- AWS Secrets Manager 入門
- 身分提供者與聯合
- Amazon CodeGuru 推出機密偵測器
- AWS Secrets Manager 如何使用 AWS Key Management Service

- Secrets Manager 中的機密加密和解密
- Secrets Manager 部落格文章
- Amazon RDS 宣布與 AWS Secrets Manager 整合

相關影片：

- 大規模管理、擷取和輪換密碼的最佳實務
- 使用 Amazon CodeGuru 機密偵測器
- 使用 AWS Secrets Manager 保護混合式工作負載的機密

相關研討會：

- 在 AWS Secrets Manager 中儲存、擷取和管理敏感憑證
- AWS Systems Manager 混合式啟用

SEC02-BP04 利用集中式身分提供者

人力身分 (員工和承包商) 可利用身分供應商來集中管理身分。由於您是從單一位置建立、指派、管理、撤銷和稽核存取權，因此這樣一來可以更好管理多個應用程式和系統中的存取權。

預期成果： 擁有集中式身分提供者，可集中管理員工使用者、身分驗證政策 (例如，要求多重要素驗證 (MFA))，以及對系統和應用程式進行授權 (例如，根據使用者的群組成員資格或屬性指派存取權)。您的員工使用者登入集中身分提供者並聯合 (單一登入) 至內部和外部應用程式，如此一來，使用者就不需記住多個憑證。您的身分提供者與您的人力資源 (HR) 系統整合，因此人事變更會自動同步至您的身分提供者。例如，若有人離開您的組織，您可以自動撤銷聯合應用程式和系統 (包括 AWS) 的存取權。您已在身分提供者中啟用詳細稽核記錄，並監控這些日誌以找出不尋常的使用者行為。

常見的反模式：

- 您未使用聯合和單一登入。您的員工使用者在多個應用程式和系統中建立了不同的使用者帳戶和憑證。
- 您尚未將員工使用者的身分生命週期自動化，例如透過整合身分提供者與您的 HR 系統。使用者離開您的組織或變更職務時，您採取手動程序在多個應用程式和系統中刪除或更新記錄。

建立此最佳實務的優勢： 透過使用集中式身分提供者，您就可以從單一位置管理員工使用者身分和政策，而且能夠將應用程式存取權指派給使用者和群組，並監控使用者登入活動。透過與您的人力資源

(HR) 系統整合，使用者變更職務時，這些變更就會同步至身分提供者，並自動更新指派的應用程式和許可。使用者離開您的組織時，系統會自動停用他們在身分提供者中的身分，並撤銷他們對聯合應用程式和系統的存取權。

未建立此最佳實務時的曝險等級：高

實作指引

員工使用者存取 AWS 的指引

員工使用者 (例如組織中的員工和承包商) 可能需要使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 存取 AWS 來執行其工作職能。您可以透過從集中式身分提供者，在兩個層級聯合至 AWS 的方式，將 AWS 存取權授與員工使用者：直接聯合至各個 AWS 帳戶，或聯合至您的 AWS 組織中的多個帳戶。

- 若要將您的員工使用者直接與各個 AWS 帳戶 聯合，您可以使用集中式身分提供者來聯合至該帳戶中的 AWS Identity and Access Management 。IAM 的彈性可讓您啟用單獨的 SAML 2.0 或 Open ID Connect (OIDC) 身分提供者用於各個 AWS 帳戶，並使用聯合身分使用者屬性進行存取控制。您的員工使用者將透過提供憑證 (例如密碼和 MFA 權杖代碼) 的方式，使用自己的 Web 瀏覽器登入身分提供者。身分提供者會向其瀏覽器發出 SAML 判斷提示，並提交至 AWS Management Console 登入 URL，以允許使用者藉由 承擔 IAM 角色對 AWS Management Console 進行單一登入。您的使用者也可以取得臨時 AWS API 憑證，以便在 AWS CLI 或 AWS SDK (從 AWS STS) 中使用，方法是 使用來自身分提供者的 SAML 判斷提示承擔 IAM 角色 。
- 若要將您的員工使用者與 AWS 組織中的多個帳戶聯合，您可以使用 AWS IAM Identity Center 集中管理員工使用者對 AWS 帳戶 和應用程式的存取權。您可以為組織啟用 Identity Center，並設定您的身分來源。IAM Identity Center 提供了預設身分來源目錄，您可以使用此目錄來管理您的使用者和群組。或者，您可以選擇外部身分來源，方法是 使用 SAML 2.0 連線至您的 外部身分提供者，並 使用 SCIM 自動佈建 使用者和群組， 或是 使用 AWS Directory Service連線至您的 Microsoft AD 目錄。身分來源設定完成後，您就可以透過在您的許可集中定義最低許可政策的方式，指派使用者和群組對 AWS 帳戶 的 存取權。您的員工使用者可以進行身分驗證的方式包括：透過您的集中身分提供者登入 AWS 存取入口網站 以及對 AWS 帳戶 和指派給他們的雲端應用程式進行單一登入。您的使用者可以設定 AWS CLI v2 以透過 Identity Center 進行身分驗證，並取得憑證來執行 AWS CLI 命令。Identity Center 也允許透過單一登入方式存取 AWS 應用程式，例如 Amazon SageMaker Studio 和 AWS IoT Sitewise Monitor 入口網站。

依照上述指引進行後，您的員工使用者在 AWS 上管理工作負載時，將不再需要使用 IAM users 和群組，可直接正常操作。您的使用者和群組會改為在 AWS 外部進行管理，而且使用者能夠以聯合身分存取 AWS 資源。聯合身分會使用您的集中式身分提供者所定義的群組。您應該找出並移除您的 AWS 帳

戶 中不再需要的 IAM 群組、IAM users 和長期存在的使用者憑證 (密碼和存取金鑰)。您可以 藉由 使用 IAM 憑證報告找到未使用的憑證， 刪除相應的 IAM users 和 刪除 IAM 群組。 您可以對組織套用 服務控制政策 (SCP) 以協助防止建立新的 IAM users 和群組，並強制透過聯合身分存取 AWS。

應用程式使用者的指引

您可以使用 Amazon Cognito 做為您的集中式身分提供者來管理應用程式 (例如行動應用程式) 使用者的身分。Amazon Cognito 可為您的 Web 和行動應用程式啟用身分驗證、授權和使用者管理功能。Amazon Cognito 提供了可擴展到數百萬使用者的身分存放區、可支援社交與企業聯合身分，並且提供進階安全功能來協助保護您的使用者和業務。您可以將自訂 Web 或行動應用程式與 Amazon Cognito 整合，在幾分鐘內就能在應用程式中加入使用者身分驗證和存取控制。Amazon Cognito 是以 SAML 和 Open ID Connect (OIDC) 等開放身分標準為基礎所建置，可支援各種不同的合規法規，並與前端和後端開發資源整合。

實作步驟

員工使用者存取 AWS 的步驟

- 使用下列其中一種方法，透過集中式身分提供者將您的員工使用者聯合至 AWS：
  - 使用 IAM Identity Center 透過與您的身分提供者聯合，對您的 AWS 組織中的多個 AWS 帳戶 啟用單一登入。
  - 使用 IAM 將您的身分提供者直接連接到各個 AWS 帳戶，以實現聯合的精細存取。
- 找出並移除已由聯合身分取代的 IAM users 和群組。

應用程式使用者的步驟

- 使用 Amazon Cognito 做為應用程式的集中式身分提供者。
- 使用 OpenID Connect 和 OAuth 將您的自訂應用程式與 Amazon Cognito 整合。您可以使用 Amplify 程式庫來開發自訂應用程式，這些程式庫提供了簡單的介面，可與各種不同的 AWS 服務進行整合，例如用於身分驗證的 Amazon Cognito。

資源

相關 Well-Architected 的最佳實務：

- SEC02-BP06 採用使用者群組和屬性
- SEC03-BP02 授予最低權限存取權
- SEC03-BP06 根據生命週期管理存取

相關文件：

- AWS 中的聯合身分
- IAM 中的安全最佳實務
- AWS Identity and Access Management 最佳實務
- 開始使用 IAM Identity Center 委派管理
- 如何在 IAM Identity Center 中針對進階使用案例使用客戶管理的政策
- AWS CLI v2：IAM Identity Center 憑證提供者

相關影片：

- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) 深入剖析
- AWS re:Invent 2022 - 使用 IAM Identity Center 簡化現有的員工存取權
- AWS re:Invent 2018：在每一層都能掌握身分

相關範例：

- 研討會：使用 AWS IAM Identity Center 實現強大的身分管理
- 研討會：無伺服器身分

相關工具：

- AWS 安全能力合作夥伴：身分和存取管理
- saml2aws

SEC02-BP05 定期稽核和輪換憑證

定期稽核和輪換憑證以限制憑證可用來存取資源的時限。長期憑證會產生許多風險，而透過定期輪換長期憑證可以降低這些風險。

預期成果：實施憑證輪換以幫助降低與使用長期憑證相關聯的風險。定期稽核和修正不符合憑證輪換政策的情況。

常見的反模式：

- 沒有稽核憑證的使用。

- 不必要地使用長期憑證。
- 使用長期憑證並且未定期輪換。

未建立此最佳實務時的風險暴露等級：中

實作指引

當您無法倚賴臨時憑證且需要長期憑證時，請稽核憑證以確保已強制定義的控制 (例如多重要素驗證 (MFA))，定期輪換並且具備適當的存取層級。

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。對於人類身分，您應要求使用者定期變更密碼，並淘汰存取金鑰而改用臨時憑證。隨著您從 AWS Identity and Access Management (IAM) 使用者移向集中式身分，您可以產生憑證報告以稽核您的使用者。

我們也建議您在身分提供者中強制和監控 MFA。您可以設定 AWS Config 規則 或使用 AWS Security Hub 安全標準來監視使用者是否已啟用 MFA。請考慮使用 IAM Roles Anywhere 為機器身分提供臨時憑證。在無法使用 IAM 角色和臨時憑證的情況下，必須經常稽核和輪換存取金鑰。

實作步驟

- 定期稽核憑證：稽核身分提供者和 IAM 中設定的身分有助於確保只有已授權的身分能存取您的工作負載。此類身分可能包括但不限於 IAM 使用者、AWS IAM Identity Center 使用者、Active Directory 使用者，或不同上游身分提供者中的使用者。例如，移除離職的人員和移除不再需要的跨帳戶角色。設立程序以定期稽核由 IAM 實體存取之服務的許可。這有助您識別需要修改的政策，以移除任何不使用的許可。使用憑證報告和 AWS Identity and Access Management Access Analyzer 來稽核 IAM 憑證和許可。您可以使用 Amazon CloudWatch 來設定對特定 API 呼叫 (在 AWS 環境中呼叫) 的警示。Amazon GuardDuty 也可以向您通知未預期的活動，這可指出對 IAM 憑證過於寬鬆的存取或意外存取。
- 定期輪換憑證：當您無法使用臨時憑證時，定期輪換長期 IAM 存取金鑰 (最長每 90 天)。如果在您不知情的情況下意外洩漏了存取金鑰，這可限制憑證可用來存取資源的時限。如需有關輪換 IAM 使用者的存取金鑰的詳細資訊，請參閱輪換存取金鑰。
- 檢閱 IAM 許可：為了改善 AWS 帳戶 的安全，請定期檢閱和監控每個 IAM 政策。確認政策遵守最低權限的原則。
- 考慮自動化 IAM 資源建立和更新： IAM Identity Center 會自動執行許多 IAM 任務，例如角色和政策管理。或者，可以使用 AWS CloudFormation 自動化 IAM 資源 (包括角色和政策) 的部署，以減少人為錯誤，因為可針對範本進行驗證和版本控制。
- 對於機器身分，使用 IAM Roles Anywhere 取代 IAM 使用者： IAM Roles Anywhere 可讓您在傳統上無法使用角色的區域中 (例如內部部署伺服器) 使用角色。IAM Roles Anywhere 使用可信的 X.509

憑證向 AWS 進行驗證及接收臨時憑證。使用 IAM Roles Anywhere 讓您無需輪換這些憑證，因為長期憑證不再儲存於內部部署環境中。請注意，您將需要監視 X.509 憑證，並在快到期時輪換。

資源

相關的最佳實務：

- SEC02-BP02 使用臨時憑證
- SEC02-BP03 安全地存放和使用機密

相關文件：

- AWS Secrets Manager 入門
- IAM 最佳實務
- 身分提供者與聯合
- 安全合作夥伴解決方案：存取與存取控制
- 臨時安全憑證
- 取得 AWS 帳戶 的憑證報告

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分

相關範例：

- Well-Architected 實驗室 - 自動化 IAM 使用者清理
- Well-Architected 實驗室 - 自動化 IAM 群組和角色的部署

SEC02-BP06 採用使用者群組和屬性

根據使用者群組和屬性定義許可，有助於減少政策的數量並降低複雜性，因而更容易實現最低權限原則。 您可以利用使用者群組，根據人員在組織中的職務，從單一位置管理多人的許可。 像是部門或位置等屬性則可在人員職務相似的情況下，針對不同的資源子集提供另一層許可範圍。

預期成果：您可以根據職務，針對執行該職務的所有使用者套用許可變更。 群組成員資格和屬性會控管使用者許可，進而減少管理個別使用者層級許可的需求。 您在身分提供者 (IdP) 中定義的群組和屬性會自動傳播到您的 AWS 環境。

常見的反模式：

- 管理個別使用者的許可，並且針對許多使用者重複此操作。
- 為群組定義的層級過高，授予的許可範圍過廣。
- 為群組定義的層級過於精細，致使成員資格發生重複和混淆的情形。
- 在可以使用屬性替代的情況下，仍使用在資源子集中具有重複許可的群組。
- 未透過整合在您 AWS 環境中的標準化身分提供者管理群組、屬性和成員資格。

未建立此最佳實務時的風險暴露等級：中

實作指引

AWS 許可於與主體 (例如使用者、群組、角色或資源) 相關聯且稱為政策的文件中定義。 針對您的員工，這可讓您根據使用者為組織執行的職務，而不是所存取的資源來定義群組。例如，`WebAppDeveloper` 群組可能附帶可在開發帳戶內用來設定服務 (例如 Amazon CloudFront) 的政策。 `AutomationDeveloper` 群組可能與 `WebAppDeveloper` 群組擁有一些共同的 CloudFront 許可。您可在另一個政策中擷取這些許可，並讓這些許可同時與這兩個群組相關聯，而不是讓這兩種職務的使用者屬於 `CloudFront Access` 群組。

除了群組之外，您還可以使用屬性來進一步設定存取範圍。例如，您的 `WebAppDeveloper` 群組中的使用者可能有 `Project` 屬性，可用來將其專案特定的資源設定至存取範圍內。 若使用此技術，則不需要在所擁有許可相同的情況下，針對處理不同專案的應用程式開發人員建立不同的群組。 您在許可政策中參照屬性的方式是根據其來源，無論其定義為聯合通訊協定的一部分 (例如 SAML、OIDC 或 SCIM)、為自訂 SAML 判斷提示，還是在 IAM Identity Center 內部設定。

實作步驟

1. 確定您將定義群組和屬性的位置。
   a. 依照SEC02-BP04 利用集中式身分提供者中的指引，您可以決定要在身分提供者內、IAM Identity Center 內，或在特定帳戶中使用 IAM user 群組定義群組和屬性。
2. 定義群組。
   a. 根據職務和所需的存取範圍決定您的群組。
   b. 如果是在 IAM Identity Center 內定義，請建立群組，並使用許可集建立所需存取層級的關聯。

c. 如果是在外部身分提供者內定義，請確定提供者是否支援 SCIM 通訊協定，並考慮在 IAM Identity Center 內啟用自動佈建。 此功能可在您的提供者與 IAM Identity Center 之間同步群組的建立、成員資格和刪除。

3. 定義屬性。

   a. 如果使用外部身分提供者，SCIM 和 SAML 2.0 通訊協定預設都會提供特定屬性。 您可以使用 `https://aws.amazon.com/SAML/Attributes/PrincipalTag` 屬性名稱，利用 SAML 判斷提示來定義和傳遞其他屬性。

   b. 如果是在 IAM Identity Center 內定義，請啟用屬性型存取控制 (ABAC) 功能並視需要定義屬性。

4. 根據群組和屬性設定許可的範圍。

   a. 您可考慮在許可政策中加入條件，用來比較您主體的屬性與所存取資源的屬性。 例如，您可以定義一項條件，規定僅在 `PrincipalTag` 條件索引鍵的值與相同名稱的 `ResourceTag` 索引鍵的值相符時，才允許對相關資源的存取。

資源

相關的最佳實務：

- SEC02-BP04 利用集中式身分提供者
- SEC03-BP02 授予最低權限存取權
- COST02-BP04 實作群組和角色

相關文件：

- IAM 最佳實務
- 管理 IAM Identity Center 中的身分
- 什麼是 ABAC for AWS？
- IAM Identity Center 中的 ABAC

相關影片：

- 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分

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 在組織內安全地共用資源
- SEC03-BP09 安全地與第三方共用資源

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

最佳實務是僅授與身分在特定情況下對特定資源執行特定動作所需的存取權。使用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，當開發人員退出專案時，其存取權將自動被撤銷，而無須變更基礎存取政策。

預期成果：使用者應該只擁有完成其工作所需的許可。使用者只應獲得在有限時間內執行特定任務的生產環境存取權，且任務完成後，存取權就應該被撤銷。許可不再需要時就應撤銷，包括當使用者移至不同的專案或工作性質。管理員特權只應授予給一小部分受信任的管理員。並應定期檢查許可，避免許可滲透的問題。電腦或系統帳戶應被授予完成其任務所需的最小許可集。

常見的反模式：

- 預設授予使用者管理員許可。
- 使用根使用者來處理每日活動。
- 建立過於寬鬆的政策，但不具完整的管理員權限。
- 不檢閱許可，無法確定是否符合最低權限存取權。

未建立此最佳實務時的風險暴露等級：高

實作指引

最低權限原則指出，只應允許身分執行完成特定任務所需的最小動作集。這平衡了可用性、效率和安全性。根據此原則運作有助於限制意外存取，也有助於追蹤誰有權存取哪些資源。IAM 使用者和角色在預設情況下沒有任何許可。根使用者預設擁有完整存取權，應該受到嚴格監控，並僅用於需要根存取權的任務。

IAM 政策用於明確授予許可給 IAM 角色或特定資源。例如，以身分為基礎的政策可以連接到 IAM 群組，而 S3 儲存貯體可由以資源為基礎的政策控制。

建立 IAM 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 允許或拒絕存取。AWS 支援各種條件，以協助您縮減存取權範圍。例如，透過使用 `PrincipalOrgID` 條件鍵，如果請求者不屬於您的 AWS 組織，您可以拒絕動作。

此外，您還可以使用 `CalledVia` 條件金鑰控制 AWS 服務代您發出的請求，例如建立 AWS Lambda 函數的 AWS CloudFormation。您應該將不同的政策類型分層，以便建立深度防禦並限制使用者的整體許可。您還可以限制在什麼條件下，可以授予哪些許可。例如，您可以允許應用程式團隊為他們建置的系統建立自己的 IAM 政策，但也必須同時套用許可界限來限制系統可以接受的最大許可。

實作步驟

- 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。
  - API 使用方式的基本政策：決定所需許可的一個方法是檢查 AWS CloudTrail 日誌。這種檢查可讓您根據使用者在 AWS 中實際執行的動作，建立適合的許可。IAM Access Analyzer 可根據活動自動產生 IAM 政策。您可以在組織或帳戶層級使用 IAM Access Advisor 來追蹤特定政策的最後存取資訊。
- 考慮使用適用於各工作性質的 AWS 受管政策。 開始建立精細的許可政策時，可能不知道從何處開始。AWS 提供常見職務的受管政策，例如帳單、資料庫管理員和資料科學家。這些政策可協助縮小使用者的存取權，同時決定如何實施最低權限政策。
- 移除不需要的許可：移除不需要的許可，並削減過於寬鬆的政策。IAM Access Analyzer 政策產生可協助微調許可政策。
- 確保使用者對生產環境具有有限的存取權：使用者應該只能存取具有有效使用案例的生產環境。在使用者執行完需要生產存取權的特定任務後，就應撤銷存取權。限制對生產環境的存取，有助於防止發生會影響生產的意外事件，也能降低意外存取的影響範圍。
- 考慮使用許可界限：許可界限是使用受管政策的功能，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。
- 考慮許可的資源標籤：使用資源標籤的屬性型存取控制模型，可讓您根據資源用途、擁有者、環境或其他條件來授予存取許可。例如，您可以使用資源標籤來區分開發環境和生產環境。使用這些標籤，您可以將開發人員限制在開發環境中。結合標記和許可政策，您可以實現精細的資源存取，無需為每個工作性質定義複雜的自訂政策。
- 使用 AWS Organizations 的服務控制政策。 服務控制政策可集中控制組織中成員帳戶的最大可用許可。重要的是，服務控制政策還能讓您限制成員帳戶中的根使用者許可。此外，還可以考慮使用 AWS Control Tower，它提供規範性受管控制，可以豐富 AWS Organizations。您也可以在 Control Tower 中定義自己的控制項。
- 為您的組織制定使用者生命週期政策：使用者生命週期政策定義了當使用者上線至 AWS、變更職務或範圍或不再需要存取 AWS 時要執行的任務。應在使用者生命週期的每個步驟中進行許可審查，以驗證許可是否受到適當限制並避免許可滲透的問題。
- 建立定期檢查許可的排程，並移除任何不需要的許可：您應該定期檢查使用者存取權，確認使用者沒有過度寬鬆的存取權。AWS Config 和 IAM Access Analyzer 可以在稽核使用者許可時提供幫助。
- 建立職務矩陣：職務矩陣會以視覺化的方式顯示您 AWS 據點內所需的各種角色和存取層級。使用職務矩陣，您可以根據組織內的使用者職責定義和區分許可。使用群組，而不是將許可直接套用至個別使用者或角色。

資源

相關文件：

- 授予最低權限
- 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 IAM 許可界限，將許可管理委託給開發人員
- 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
- 屬性型存取控制 (ABAC)
- 透過查看使用者活動來縮小政策範圍
- 檢視角色存取
- 使用標記來組織環境並提高責任心
- AWS 標記策略
- 標記 AWS 資源

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
- 實驗室：EC2 的 IAM 標籤型存取控制

SEC03-BP03 建立緊急存取程序

建立一項程序，在集中式身分提供者發生問題時，緊急存取您的工作負載。

您必須針對可能導致緊急事件發生的不同故障模式設計不同的程序。例如，正常情況下，您的員工使用者會使用集中式身分提供者 (SEC02-BP04) 聯合至雲端，以管理其工作負載。不過，如果您的集中式身分提供者發生錯誤，或是雲端中聯合的組態經過修改，則您的員工使用者可能無法連至雲端。緊急存取程序可讓授權的管理員透過替代方式 (例如聯合或直接使用者存取的替代形式) 存取您的雲端資源，以修正聯合組態或工作負載的問題。緊急存取程序會持續使用，直到恢復正常聯合機制為止。

預期成果：

- 您已定義並記載可視為緊急情況的故障模式：請考慮正常情況以及使用者用來管理工作負載的系統。考慮這些相依性如何發生錯誤並導致緊急情況。您可以在 可靠性支柱 中找到問題與最佳實務，有助於識別故障模式並架構更具彈性的系統，以盡量降低故障的可能性。
- 您已記載確認故障為緊急情況須遵循的步驟。例如，您可以要求身分管理員檢查主要和待命身分提供者的狀態，如果兩者都無法使用，則發布身分提供者發生錯誤緊急事件。
- 您已針對每一種緊急或故障模式類型定義了緊急存取程序。明確定義可減少部分使用者過度使用一般程序，來處理所有類型的緊急情況。您的緊急存取程序描述了各個程序應在何種情況下使用，以及不應在哪些情況下使用，並指出可能適用的替代程序。
- 您的程序完整記載了詳細指示和程序手冊，可快速有效地遵循。請記住，緊急事件對使用者來說可能會非常緊張，他們可能面對極大的時間壓力，因此程序的設計應盡可能簡單。

常見的反模式：

- 您沒有詳細記載且經過充分測試的緊急存取程序。您的使用者未準備好面對緊急情況，而在緊急事件發生時只能隨機應變。
- 您的緊急存取程序與正常存取機制依賴相同的系統 (例如集中式身分提供者)。這表示，一旦這類系統發生故障，就可能同時影響您的正常和緊急存取機制，並損及您從故障復原的能力。
- 您的緊急存取程序用在非緊急情況。例如，您的使用者經常濫用緊急存取程序，因為他們發現直接進行變更透過管道提交變更容易。
- 您的緊急存取程序未產生足夠的日誌來稽核程序，或是日誌未受監控，無法在發生可能濫用程序的情況時發出警示。

建立此最佳實務的優勢：

- 只要有詳細記載且經充分測試的緊急存取程序，就能縮短使用者回應和解決緊急事件所花的時間。這樣就能進一步減少停機時間，並為客戶帶來更高的服務可用性。
- 您可以追蹤每一項緊急存取請求，以及偵測未經授權的人士試圖濫用程序來處理非緊急事件的情況，並發出警示。

未建立此最佳實務時的曝險等級：中

實作指引

本節提供建立緊急存取程序的指引，用於處理與 AWS 上部署的工作負載相關的數種故障模式，一開始先介紹適用於所有故障模式的通用指引，接著再根據故障模式類型說明特定指引。

適用所有故障模式的通用指引

為故障模式設計緊急存取程序時，請考慮下列事項：

- 記載程序的前提和假設：應該和不應該使用程序的時機。這樣做有助於詳細說明故障模式並記載假設，例如其他相關系統的狀態。舉例來說，故障模式 2 的程序假設身分提供者可以使用，但 AWS 上的組態已經過修改或已過期。
- 預先建立緊急存取程序所需的資源 (SEC10-BP05)。例如，在所有工作負載帳戶中預先建立具有 IAM users 和角色的緊急存取 AWS 帳戶，以及跨帳戶 IAM 角色。這樣就可確定這些資源在緊急事件發生時立即可用。透過預先建立資源，您就不必依賴 AWS 控制平面 API (用來建立和修改 AWS 資源)，因為它們在緊急情況下可能無法使用。此外，預先建立 IAM 資源就不需考慮 因最終一致性而可能發生的延遲。
- 請將緊急存取程序納入您的事件管理計畫當中 (SEC10-BP02)。記載緊急事件的追蹤方式，並傳達給組織中的其他人，例如同儕團隊、您的領導階層，以及適時向外傳達給您的客戶和業務合作夥伴。
- 在您現有的服務請求工作流程系統 (若有的話) 中定義緊急存取請求程序。一般來說，這類工作流程系統可讓您建立接收表單來收集有關請求的資訊、在工作流程的每個階段追蹤請求，以及新增自動和手動核准步驟。將每一個請求與事件管理系統中追蹤的對應緊急事件建立關聯。採用統一的緊急存取系統，可讓您在單一系統中追蹤這些請求、分析使用趨勢並改善程序。
- 確認您的緊急存取程序只能由經授權的使用者啟動，並且視情況要求使用者同儕或管理層的核准。核准程序在營業時間內外都要能夠有效運作。定義在主要核准者沒有空的情況下，如何由次要核准者核准請求，以及如何在您的管理鏈中向上呈報，直到請求獲得核准。
- 確認程序會同時針對成功和失敗的嘗試產生詳細的稽核日誌和事件，以便取得緊急存取權。同時監控請求程序和緊急存取機制，以偵測濫用或未經授權存取的情況。將活動與事件管理系統中正在發生的緊急事件相互關聯，並且在動作於預期時間之外發生時發出警示。例如，您應該監控緊急存取 AWS 帳戶 中的活動並發出警示，因為這不應該用於正常操作。

- 定期測試緊急存取程序，以確認步驟是否清楚，並且快速有效地授予正確的存取層級。您的緊急存取程序應在事故回應模擬的過程中 (SEC10-BP07) 和災難恢復測試中 (REL13-BP03) 進行測試。

故障模式 1：用於聯合至 AWS 的身分提供者無法使用

如 SEC02-BP04 利用集中式身分提供者中所述，我們建議您利用集中式身分提供者來聯合您的員工使用者，以授予 AWS 帳戶 的存取權。您可以使用 IAM Identity Center 聯合至您 AWS 組織中的多個 AWS 帳戶，或是使用 IAM 聯合至個別 AWS 帳戶。在這兩種情況下，員工使用者都會先透過集中式身分提供者進行身分驗證，然後才重新導向至 AWS 登入端點進行單一登入。

萬一您的集中式身分提供者無法使用，您的員工使用者就無法聯合至 AWS 帳戶 或管理其工作負載。在此緊急事件中，您可以提供緊急存取程序讓一小群管理員存取 AWS 帳戶，以便執行無法等到集中式身分提供者恢復連線後才處理的重要工作。例如，您的身分提供者停擺了 4 小時，而在此期間，您需要修改生產帳戶中 Amazon EC2 Auto Scaling 群組的上限，以處理客戶流量意外暴增的情況。您的緊急管理員應遵循緊急存取程序，才能獲得特定生產 AWS 帳戶 的存取權並進行必要的變更。

緊急存取程序依賴預先建立的緊急存取 AWS 帳戶，該帳戶單純用於緊急存取，並擁有 AWS 資源 (例如 IAM 角色和 IAM users) 可支援緊急存取程序。在正常操作期間，任何人都不應該存取緊急存取帳戶，而且您必須監控濫用此帳戶的情況並發出警示 (如需詳細資訊，請參閱前一節「通用指引」)。

緊急存取帳戶具有緊急存取 IAM 角色，有權在需要緊急存取的 AWS 帳戶 中擔任跨帳戶角色。這些 IAM 角色會預先建立並設定信任政策，以便信任緊急帳戶的 IAM 角色。

緊急存取程序可以使用下列其中一種方法：

- 您可以預先建立一組 IAM users 並包含相關的強式密碼和 MFA 權杖，以供緊急存取帳戶中的緊急管理員使用。這些 IAM users 有權承擔 IAM 角色，且後續可在需要緊急存取時跨帳戶存取 AWS 帳戶。我們建議這類使用者的數量越少越好，並且將每一位使用者指派給單一緊急管理員。在緊急情況下，緊急管理員使用者會使用其密碼和 MFA 權杖代碼登入緊急存取帳戶，切換到緊急帳戶中的緊急存取 IAM 角色，最後再切換到工作負載帳戶中的緊急存取 IAM 角色，以執行緊急變更動作。這種方法的優點是，每個 IAM user 都會指派給一名緊急管理員，而且您可以透過檢閱 CloudTrail 事件得知登入的使用者。缺點是，您必須維護多個 IAM users 及其相關聯的長期存在密碼和 MFA 權杖。
- 您可以使用緊急存取 AWS 帳戶 根使用者 來登入緊急存取帳戶、擔任緊急存取的 IAM 角色，並且在工作負載帳戶中擔任跨帳戶角色。我們建議您為根使用者設定強式密碼和多個 MFA 權杖。同時也建議您，將密碼和 MFA 權杖儲存在強制執行強式身分驗證和授權的安全企業憑證保存庫中。您應確保密碼和 MFA 權杖重設要素的安全性：將帳戶的電子郵件地址設定為受到您的雲端安全管理員監控的電子郵件分發清單，並將帳戶的電話號碼設定為同樣受到安全管理員監控的共用電話號碼。這種方法的優點是，只需管理一組根使用者憑證。缺點是，由於這是共用使用者，因此有多個管理員能夠以根使用者的身分登入。您必須稽核企業保存庫日誌事件，以確定哪個管理員簽出了根使用者密碼。

故障模式 2：AWS 上的身分提供者組態已經過修改或已過期

若要讓您的員工使用者聯合至 AWS 帳戶，您可以使用外部身分提供者設定 IAM Identity Center，或建立 IAM 身分提供者 (SEC02-BP04)。一般來說，您可以匯入身分提供者提供的 SAML 中繼資料 XML 文件來進行這些設定。中繼資料 XML 文件包含一個 X.509 憑證，對應於身分提供者用來簽署其 SAML 判斷提示的私密金鑰。

AWS 端的這些組態可能遭到管理員誤改或誤刪。另一種情況是，匯入 AWS 中的 X.509 憑證可能過期，而具有新憑證的新中繼資料 XML 尚未匯入 AWS 中。這兩種情況都可能使員工使用者的 AWS 聯合中斷，導致緊急情況發生。

在這類緊急事件中，您可以提供 AWS 的存取權給身分管理員，以修正聯合問題。例如，您的身分管理員使用緊急存取程序登入緊急存取 AWS 帳戶、切換為 Identity Center 管理員帳戶中的角色，並透過從您的身分提供者匯入最新的 SAML 中繼資料 XML 文件來更新外部身分提供者組態，以重新啟用聯合。聯合修復後，您的員工使用者繼續使用正常操作程序來聯合至其工作負載帳戶。

您可以依照先前「故障模式 1」中詳述的方法來建立緊急存取程序。您可以將最低權限許可授予身分管理員，以限制他們只能存取 Identity Center 管理員帳戶以及在該帳戶中對 Identity Center 執行動作。

故障模式 3：Identity Center 中斷

萬一發生 IAM Identity Center 或 AWS 區域 中斷的情況，建議您設定一個可用來臨時存取 AWS Management Console 的組態。

緊急存取程序會在緊急帳戶中，使用您的身分提供者對 IAM 的直接聯合。如需有關程序和設計考量的詳細資訊，請參閱 設定 AWS Management Console 的緊急存取。

實作步驟

適用所有故障模式的通用步驟

- 建立緊急存取程序專用的 AWS 帳戶。在帳戶中預先建立所需的 IAM 資源，例如 IAM 角色或 IAM users，也可以選擇建立 IAM 身分提供者。此外，在工作負載 AWS 帳戶 中預先建立跨帳戶 IAM 角色，並與緊急存取帳戶中對應的 IAM 角色建立信任關係。您可以使用 AWS CloudFormation StackSets 搭配 AWS Organizations 在組織的成員帳戶中建立此類資源。

- 建立 AWS Organizations 服務控制政策 (SCP) 以拒絕刪除和修改成員 AWS 帳戶 中的跨帳戶 IAM 角色。
- 為緊急存取 AWS 帳戶 啟用 CloudTrail，並將軌跡事件傳送到日誌收集 AWS 帳戶 中的中央 S3 儲存貯體。如果您使用 AWS Control Tower 來設定和管控您的 AWS 多帳戶環境，則您使用 AWS

Control Tower 建立或在 AWS Control Tower 中註冊的每個帳戶都會預設啟用 CloudTrail，並傳送至專用日誌封存 AWS 帳戶 中的 S3 儲存貯體。

- 透過建立在主控台登入時比對的 EventBridge 規則來監控活動的緊急存取帳戶，以及透過緊急 IAM 角色監控 API 活動。當活動於事件管理系統中追蹤的持續緊急事件之外發生時，傳送通知給您的安全營運中心。

適用「故障模式 1：用於聯合至 AWS 的身分提供者無法使用」及「故障模式 2：AWS 上的身分提供者組態已經過修改或已過期」的其他步驟

- 根據您選擇的緊急存取機制預先建立資源：
  - 使用 IAM users： 預先建立 IAM users 並設定強式密碼和相關聯的 MFA 裝置。
  - 使用緊急帳戶根使用者： 設定根使用者使用強式密碼，並將密碼儲存在您的企業憑證保存庫中。將多個實體 MFA 裝置與根使用者建立關聯，並將裝置儲存在您的緊急管理員小組成員可快速存取的位置。

適用「故障模式 3：Identity Center 中斷」的其他步驟

- 如 設定 AWS Management Console 的緊急存取中所述，在緊急存取 AWS 帳戶 中，建立 IAM 身分提供者，以從您的身分提供者啟用直接 SAML 聯合。
- 在 IdP 中建立緊急操作群組，但不新增任何成員。
- 在緊急存取帳戶中建立對應於緊急操作群組的 IAM 角色。

資源

相關 Well-Architected 的最佳實務：

- SEC02-BP04 利用集中式身分提供者
- SEC03-BP02 授予最低權限存取權
- SEC10-BP02 制定事件管理計畫
- SEC10-BP07 執行演練日

相關文件：

- 設定 AWS Management Console 的緊急存取
- 讓 SAML 2.0 聯合身分使用者存取 AWS Management Console

- 緊急存取

相關影片：

- AWS re:Invent 2022 - 使用 IAM Identity Center 簡化現有的員工存取權
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) 深入剖析

相關範例：

- AWS 緊急存取角色
- AWS 客戶程序手冊架構
- AWS 事故應變程序手冊範例

SEC03-BP04 持續減少許可

在團隊確定所需的存取權時，請移除不需要的許可，並建立檢閱程序以達最低權限許可。持續監控人類和機器存取權，並移除不使用的身分和許可。

預期成果：許可政策應該遵守最低權限原則。隨著工作職責和角色的定義變得更具體，您需要檢閱許可政策以移除不必要的許可。若憑證遭到意外洩露或以其他方式在未經授權下遭存取，此方法可縮小影響範圍。

常見的反模式：

- 預設為使用者授予管理員許可。
- 建立過於寬鬆的政策，但不具完整的管理員權限。
- 保留不再需要的許可政策。

未建立此最佳實務時的風險暴露等級：中

實作指引

在團隊和專案剛開始時，可使用寬鬆的許可政策來激發創新和敏捷性。例如，在開發或測試環境中，可以讓開發人員存取廣泛的 AWS 服務。我們建議您持續評估存取權，並將存取權限於完成目前工作所需的該些服務和服務動作。我們建議對人類和機器身分進行此項評估。機器身分有時候稱為系統或服務帳戶，是提供 AWS 存取權給應用程式或伺服器的身分。此存取權在生產環境中尤為重要，因為過於寬鬆的許可可能影響廣大而且可能暴露客戶資料。

AWS 提供多種方法可幫助識別未使用的使用者、角色、許可和憑證。AWS 也有助於分析 IAM 使用者和角色的存取活動，包括相關聯的存取金鑰，以及對 AWS 資源的存取，例如 Amazon S3 儲存貯體中的物件。AWS Identity and Access Management Access Analyzer 政策產生可協助您根據某主體進行互動的實際服務和動作來建立限制性許可。屬性型存取控制 (ABAC) 可以幫助簡化許可管理，因為您可以使用使用者的屬性提供許可給他們，而不是將許可證測直接附加到每個使用者。

實作步驟

- 使用 AWS Identity and Access Management Access Analyzer：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。
- 使用 IAM Access Analyzer 政策產生： IAM Access Analyzer 政策產生可協助您根據 IAM 使用者或角色的存取活動建立精細的許可政策。
- 為 IAM 使用者和角色確定可接受的時間範圍和使用政策：使用上次存取的時間戳記以識別未使用的使用者和角色並將其移除。檢閱服務和動作上次存取的資訊，以識別和設定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 S3 動作，並將該角色的存取權僅限於該些動作。AWS Management Console 中有提供「上次存取的資訊」功能，並且您可透過程式設計的方式將這些功能併入基礎設施工作流程和自動化工具中。
- 考慮在 AWS CloudTrail 中記錄資料事件：在預設情況下，CloudTrail 不會記錄資料事件，例如 Amazon S3 物件層級活動 (如 `GetObject` 和 `DeleteObject`) 或 Amazon DynamoDB 資料表活動 (如 `PutItem` 和 `DeleteItem`)。考慮為這些事件啟用記錄功能以確定哪些使用者和角色需要存取特定 Amazon S3 物件或 DynamoDB 資料表項目。

資源

相關文件：

- 授予最低權限
- 移除不需要的憑證
- 什麼是 AWS CloudTrail？
- 制定政策
- 記錄和監控 DynamoDB
- 為 Amazon S3 儲存貯體和物件啟用 CloudTrail 事件記錄
- 取得 AWS 帳戶的憑證報告

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
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) 深入剖析

SEC03-BP05 為您的組織定義許可防護機制

使用許可防護機制縮小可授予主體的可用許可範圍。許可證測評估鏈包括您的防護機制，可在做授權決策時確定主體的有效許可。 您可以採用分層方式定義防護機制。對整個組織廣泛套用一些防護機制，另外對臨時存取工作階段套用一些精細的防護機制。

預期成果：您可以使用個別 AWS 帳戶 清楚隔離環境。 服務控制政策 (SCP) 用於定義整個組織的許可防護機制。較寬鬆的防護機制設定於最靠近組織根目錄的階層層級，較嚴謹的防護機制則設定於較靠近個別帳戶的層級。 在受支援的情況下，資源政策會定義主體必須滿足才能取得資源存取權的條件。資源政策也會適時縮小允許的動作範圍。 許可界限會設置在管理工作負載許可的主體上，以將許可管理工作委派給個別工作負載擁有者。

常見的反模式：

- 在 AWS 組織內建立成員 AWS 帳戶，但未使用 SCP 來限制其根憑證適用的用途和許可。
- 根據最低權限指派許可，但未對可授予的許可集上限設置防護機制。
- 依賴 AWS IAM 的隱含拒絕基礎來限制許可，相信政策不會授予不需要的明確允許許可。
- 在相同 AWS 帳戶 中執行多個工作負載環境，然後依賴 VPC、標籤或資源政策等機制來強制執行許可界限。

建立此最佳實務的優勢：許可防護機制有助於建立信心，確保不會有不需要的許可授予情況，即使許可政策嘗試這樣做也不必擔心。 此最佳實務可透過縮小需考量的許可範圍上限來簡化定義和管理許可。

未建立此最佳實務時的風險暴露等級：中

實作指引

建議您採用分層方式為您的組織定義許可防護機制。此方式能夠隨著套用額外的分層，有系統地減少可能的許可集上限。這種方式可幫助您根據最低權限原則授予存取權，降低了因政策組態錯誤導致意外存取的風險。

設置許可防護機制的第一步，是將您的工作負載和環境隔離到個別 AWS 帳戶 中。 在沒有明確許可的情況下，某一帳戶中的主體無法存取另一帳戶中的資源，即使兩個帳戶在相同 AWS 組織中或在相同組織單位 (OU) 下亦是如此。 您可以使用 OU 將您要管理的帳戶分組為一個單位。

下一步是減少您可授予組織的成員帳戶內主體的許可集上限。 您可以使用服務控制政策 (SCP) 達到此目的，這些政策可套用至 OU 或帳戶。 SCP 可強制執行通用的存取控制，例如限制對特定 AWS 區域的存取、協助防止資源遭到刪除，或停用有潛在風險的服務動作。 您套用至組織根目錄的 SCP 只會影響其成員帳戶，而不會影響管理帳戶。 SCP 只會控管組織內的主體。您的 SCP 不會控管組織外部存取您資源的主體。

再下一步是使用 IAM 資源政策來設定您可對其控管的資源執行的可用動作範圍，以及設定執行動作的主體必須符合的任何條件。 這個範圍可以很廣泛，例如只要主體屬於組織的一部分就允許所有動作 (使用 PrincipalOrgId 條件金鑰)，也可以很精細，例如只允許特定 IAM 角色執行特定動作。 您可以在 IAM 角色信任政策中採取類似方法，並附帶條件。 如果資源或角色信任政策明確指名相同帳戶中的某個主體作為其控管的角色或資源，則該主體不需要有授予相同許可的附加 IAM 政策。 如果主體位於與資源不同的帳戶中，則該主體確實需要有授予這些許可的附加 IAM 政策。

通常工作負載團隊會希望管理其工作負載需要的許可。 這樣一來，他們便需要建立新的 IAM 角色和許可政策。 您可以擷取允許團隊在 IAM 許可界限中授予的許可範圍上限，並將此文件與 IAM 角色建立關聯，之後團隊就可使用該角色來管理其 IAM 角色和許可。 這種方法可讓他們自由完成其工作，同時降低擁有 IAM 管理存取權的風險。

更詳細的步驟是實作特殊權限存取管理 (PAM) 和臨時提升存取管理 (TEAM) 技術。 PAM 的範例是，要求主體在採取特殊權限動作之前執行多重要素驗證。 如需詳細資訊，請參閱設定受 MFA 保護的 API 存取。TEAM 需要使用解決方案來管理允許主體擁有已提升存取權的核准和時間範圍。 其中一種方法是暫時將主體新增至具有已提升存取權之 IAM 角色的角色信任政策中。 另一種方法是在正常操作情況下，使用工作階段政策縮小 IAM 角色授予主體的許可範圍，然後在核准的期間內暫時解除此限制。若要進一步了解已經過 AWS 和精選合作夥伴驗證的解決方案，請參閱臨時提升的存取權。

實作步驟

1. 將您的工作負載和環境隔離到個別 AWS 帳戶 中。
2. 使用 SCP 減少可授予組織的成員帳戶內主體的許可集上限。
   a. 建議您採用允許清單方法來編寫 SCP，此方法會拒絕所有動作，除了您允許的動作以及符合特定條件的動作。首先定義您要控制的資源，然後將「效果」設定為「拒絕」。使用 NotAction 元素拒絕您所指定的動作以外的所有動作。結合此元素與 NotLike 條件，可定義允許這些動作的時機 (如適用)，例如 StringNotLike 和 ArnNotLike。
   b. 請參閱服務控制政策範例。
3. 使用 IAM 資源政策可縮小資源上許可動作的範圍並指定條件。 在 IAM 角色信任政策中使用條件來建立承擔角色的限制。
4. 將 IAM 許可界限指派至 IAM 角色，之後工作負載團隊可使用這些角色來管理自己的工作負載 IAM 角色和許可。

5. 根據您的需求評估 PAM 和 TEAM 解決方案。

資源

相關文件：

- AWS 上的資料周邊
- 使用資料周邊設置許可防護機制
- 政策評估邏輯

相關範例：

- 服務控制政策範例

相關工具：

- AWS 解決方案：臨時提升的存取管理
- 適用 TEAM 的已驗證安全合作夥伴解決方案

SEC03-BP06 根據生命週期管理存取

監控並調整授予您的主體 (使用者、角色和群組) 在組織內其整個生命週期的許可。隨著使用者變更角色調整群組成員資格，並在使用者離開組織時移除存取權。

預期成果：您會在組織內監控並調整主體整個生命週期的許可，進而降低不必要權限帶來的風險。您會在建立使用者時授予適當的存取權。您可以隨著使用者的職責變更修改存取權，並且在使用者不再為作用中狀態或離開組織時移除存取權。您可以集中管理使用者、角色和群組的變更。您使用自動化的方式將變更傳播到您的 AWS 環境。

常見的反模式：

- 您事先授予身分過多或過廣的存取權限，超過最初所需的範圍。
- 您未隨著身分的角色和職責經過一段時間發生變更，而審查並調整存取權限。
- 您未移除失效或已終止身分的有效存取權限。此舉會增加未經授權存取的風險。
- 您未自動化身分生命週期管理。

未建立此最佳實務時的風險暴露等級：中

實作指引

在身分的整個生命週期中，仔細管理和調整您授予身分 (例如使用者、角色、群組) 的存取權限。此生命週期涵蓋初始入職階段、後續角色和職責變更，以及最終離職或終止。根據生命週期階段主動管理存取權，以維護適當的存取層級。遵守最低權限原則，以降低過度或不必要存取權限帶來的風險。

您可以直接在 AWS 帳戶 內管理 IAM users 的生命週期，或透過從員工身分提供者至 AWS IAM Identity Center 的聯合來進行管理。針對 IAM users，您可以在 AWS 帳戶 內建立、修改和刪除使用者及其相關聯的許可。針對聯合使用者，您可以使用 IAM Identity Center 管理其生命週期，方法是透過跨網域身分管理系統 (SCIM) 通訊協定，同步源自您組織身分提供者的使用者和群組資訊。

SCIM 是開放標準通訊協定，可在不同系統中自動佈建和解除佈建使用者身分。透過將身分提供者與使用 SCIM 的 IAM Identity Center 整合，您就可以自動同步使用者和群組資訊，協助確認組織是否根據其授權身分來源的變更授予、修改或撤銷存取權限。

隨著組織內員工的角色和職責改變，調整他們的存取權限。您可以使用 IAM Identity Center 的許可集來定義不同的工作角色或職責，並將其與適當的 IAM 政策和許可建立關聯。當員工的角色變更時，您可以更新其指派的許可集，以反映他們的新職責。確認他們具有必要的存取權，同時遵守最低權限原則。

實作步驟

1. 定義並記錄存取管理生命週期流程，包括授與初始存取權、定期審查和離職的程序。
2. 實作 IAM 角色、群組和許可界線，以共同管理存取權，並強制執行受允許的最高存取層級。
3. 使用 IAM Identity Center 與聯合身分提供者 (例如 Microsoft Active Directory、Okta、Ping Identity) 整合，使其作為使用者和群組資訊的授權來源。
4. 使用 SCIM 通訊協定可將來自身分提供者的使用者和群組資訊同步到 IAM Identity Center 的身分存放區中。
5. 在 IAM Identity Center 中建立許可集，以代表組織內不同的工作角色或職責。為每個許可集定義適當的 IAM 政策和許可。
6. 實施定期存取權審查、提示撤銷存取權，以及持續改進存取權管理生命週期流程。
7. 為員工提供有關存取權管理最佳實務的培訓和認知。

資源

相關的最佳實務：

- SEC02-BP04 利用集中式身分提供者

相關文件：

- 管理您的身分來源
- 管理 IAM Identity Center 中的身分
- 使用 AWS Identity and Access Management Access Analyzer
- IAM Access Analyzer 政策產生

相關影片：

- AWS re:Inforce 2023 - 使用 AWS IAM Identity Center 管理臨時提升的存取權
- AWS re:Invent 2022 - 使用 IAM Identity Center 簡化現有的員工存取權
- AWS re:Invent 2022 - 使用 Access Analyzer 掌握 IAM 政策之力並駕馭許可

SEC03-BP07 分析公有和跨帳戶存取權

持續監控突顯公有和跨帳戶存取權的發現結果。減少公有存取權和跨帳戶存取權，僅限於需要此類存取的特定資源。

預期成果：知道您共用了哪些 AWS 資源以及共用的對象。持續監控和稽核您共用的資源以確認這些資源僅與已授權主體共用。

常見的反模式：

- 沒有維持共用資源的詳細目錄。
- 未遵循程序來核准跨帳戶或資源的公有存取權。

未建立此最佳實務時的風險暴露等級：低

實作指引

如果您的帳戶位於 AWS Organizations 中，您可以將資源的存取權授予整個組織、特定組織單位或個別帳戶。如果您的帳戶不是組織的成員，您可以與個別帳戶共用資源。您可以使用以資源為基礎的政策 (例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體政策) 來授予直接跨帳戶存取權，或允許另一個帳戶中的主體擔任您帳戶中的 IAM 角色。當使用資源政策時，確認僅將該存取權授予已授權的主體。定義程序，來核准所有需要公開提供的資源。

AWS Identity and Access Management Access Analyzer 採用可證明的安全性來找出從其帳戶外部存取資源的所有路徑。它會持續審查資源政策，並報告公有和跨帳戶存取權的發現結果，讓您輕鬆分析潛

在的各種存取。考慮使用 AWS Organizations 設定 IAM Access Analyzer，以確認您對所有帳戶具有能見度。IAM Access Analyzer 也允許您在部署資源許可之前預覽發現結果。這可讓您驗證政策變更僅授予您資源預期的公有和跨帳戶存取權。設計多帳戶存取權時，您可以使用信任政策來控制可以擔任角色的情況。例如，您可以使用 `PrincipalOrgId` 條件金鑰來拒絕嘗試從 AWS Organizations 外部擔任角色的動作。

AWS Config 可以報告設定不當的資源，並可透過 AWS Config 政策檢查來偵測已設定公開存取的資源。諸如 AWS Control Tower 和 AWS Security Hub 的服務可簡化在 AWS Organizations 間部署控制和防護機制的作業，以識別和修正公開暴露的資源。例如，AWS Control Tower 具備受管的防護機制，可偵測是否有任何可由 AWS 帳戶 還原的 Amazon EBS 快照。

實作步驟

- 考慮為 AWS Organizations 啟用 AWS Config：AWS Config 可讓您將 AWS Organizations 內來自多個帳戶的發現結果彙總到一個委派的管理員帳戶。這提供了全面性檢視並讓您在帳戶間部署 AWS Config 規則 以偵測公開可存取的資源。
- 設定 AWS Identity and Access Management Access Analyzer 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 S3 儲存貯體或 IAM 角色。
- 使用 AWS Config 中的自動矯正以回應 Amazon S3 儲存貯體的公開存取設定中的變更：您可以自動重新啟用 Amazon S3 儲存貯體的封鎖公開存取設定。
- 實作監控和警示以識別 Amazon S3 儲存貯體是否已變為公有：您必須設立監控與警示以識別何時停用 Amazon S3 封鎖公開存取，以及 Amazon S3 儲存貯體是否變為公有。此外，如果您正在使用 AWS Organizations，可以建立服務控制政策來防止對 Amazon S3 公開存取政策進行變更。AWS Trusted Advisor 會檢查具有公開存取許可的 Amazon S3 儲存貯體。將上傳或刪除存取權授予每個人的儲存貯體許可，可讓任何人在儲存貯體中新增、修改或移除項目，進而產生潛在的安全問題。Trusted Advisor 檢查會分析明確的儲存貯體許可，以及可能覆寫儲存貯體許可的相關儲存貯體政策。您也可以使用 AWS Config 來監控 Amazon S3 儲存貯體的公開存取。如需詳細資訊，請參閱如何使用 AWS Config 來監控及回應允許公開存取的 Amazon S3 儲存貯體。檢閱存取權時，請務必考慮 Amazon S3 儲存貯體中包含何種類型的資料。Amazon Macie 有助於探索和保護敏感資料，例如 PII、PHI 憑證 (如私有或 AWS 金鑰)。

資源

相關文件：

- 使用 AWS Identity and Access Management Access Analyzer
- AWS Control Tower 控制程式庫

- AWS 基礎安全最佳實務標準
- AWS Config 受管規則
- AWS Trusted Advisor 檢查參考
- 使用 Amazon EventBridge 監控 AWS Trusted Advisor 檢查結果
- 管理組織內所有帳戶間的 AWS Config 規則
- AWS Config 與 AWS Organizations

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 在組織內安全地共用資源

隨著工作負載數量增加，您可能需要在這些工作負載內共用資源的存取權，或在多個帳戶間多次佈建資源。您可能具備劃分環境 (例如擁有開發、測試和生產環境) 的建構模組。然而，擁有分隔建構模組並不會限制您安全共用的能力。透過共用重疊的元件，您可以降低營運負擔並允許一致的體驗，而不用猜測在多次建立相同的資源時可能錯過了什麼。

預期成果：使用安全方法在組織內共用資源，藉此充分減少意外存取，並協助您的資料外洩防護計畫。減輕與管理個別元件相較下的營運負擔，減少多次手動建立相同元件的錯誤，以及增加工作負載的可擴展性。您可以從多點失敗案例中更短的解決時間獲益，並更有信心確定何時不再需要某元件。如需有關分析外部共用的資源的規範指引，請參閱SEC03-BP07 分析公有和跨帳戶存取權。

常見的反模式：

- 缺乏可持續監控和自動發出意外外部共用通知的程序。
- 對於應該和不應該共用的內容缺乏基準。
- 預設採用廣泛的開放政策而不是在必要時明確共用。
- 必要時手動建立重疊的基礎資源。

未建立此最佳實務時的風險暴露等級：中

實作指引

建構您的存取控制和模式來管控安全地取用共用資源並只與信任的實體共用。監控共用資源並持續審查共用資源存取，在不當或意外共用時獲得警示。檢閱分析公開和跨帳戶存取權協助您確立管控能力以減少外部存取，而僅限於需要存取的資源，以及確立程序持續監控並自動提供警示。

在 AWS Organizations 內跨帳戶共用受到數個 AWS 服務的支援，例如 AWS Security Hub、Amazon GuardDuty 和 AWS Backup。這些服務允許將資料共用到中央帳戶，從中央帳戶存取，或從中央帳戶管理資源和資料。例如，AWS Security Hub 可以將發現結果從個別帳戶轉移到中央帳戶，讓您能夠檢視所有發現結果。AWS Backup 可以對資源進行備份並在帳戶之間共用。您可以使用 AWS Resource Access Manager (AWS RAM) 來共用其他常見的資源，例如 VPC 子網路和 Transit Gateway 附件、AWS Network Firewall 或 Amazon SageMaker 管道。

若要將您的帳戶限制為僅共用組織內的資源，請使用服務控制政策 (SCP) 防止存取外部主體。當共用資源時，結合身分型控制和網路控制為您的組織建立資料周邊，以幫助預防意外存取。資料周邊是一組預防性防護機制，可協助確認只有可信的身分從預期的網路存取可信的資源。這些控制項應適當限制可以共用哪些資源，並防止共用或公開不應該允許的資源。例如，做為資料周邊的一部分，您可以使用 VPC 端點政策和 `AWS:PrincipalOrgId` 條件來確保存取 Amazon S3 儲存貯體的身分屬於您的組織。需要注意的是，SCP 不適用於連結服務的角色 (LSR) 或 AWS 服務主體。

當使用 Amazon S3 時，請停用 Amazon S3 儲存貯體的 ACL 並使用 IAM 政策來定義存取控制。若要限制從 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，包括使用 AWS Key Management Service 的伺服器端加密。

在某些情況下，您可能會想要允許在組織外部共用資源或將資源的存取權授予第三方。如需有關管理許可以在外部共用資源的規範指引，請參閱許可管理。

實作步驟

1. 使用 AWS Organizations。

   AWS Organizations 是一項帳戶管理服務，可讓您將多個 AWS 帳戶 合併至您建立且集中管理的組織中。您可以將帳戶編組成組織單位 (OU) 並將不同的政策附加到各個 OU，以協助滿足您的預算、安全和合規需求。您也可以控制 AWS 人工智慧 (AI) 和機器學習 (ML) 服務收集和儲存資料的方式，並使用與 Organizations 整合的 AWS 服務的多帳戶管理功能。
2. 整合 AWS Organizations 與 AWS 服務。

   當您啟用 AWS 服務代表您在組織的成員帳戶中執行任務時，AWS Organizations 會在每個成員帳戶中為該服務建立一個連結 IAM 服務的角色。您應該使用 AWS Management Console、AWS API

或 AWS CLI 來管理可信存取。如需有關啟用可信存取的規範指引，請參閱使用 AWS Organizations 與其他 AWS 服務以及您可以搭配 Organizations 使用的 AWS 服務。

3. 建立資料周邊。

   AWS 周邊一般表示為由 AWS Organizations 管理的組織。許多人將存取 AWS 資源與內部部署網路和系統一同視為「我的 AWS」的周邊。周邊的目標是要確認若身分可信、資源可信且是預期的網路，則允許存取。

   a. 定義並實作周邊。

      遵循《在 AWS 上建置資料周邊》白皮書的周邊實作中所述的步驟，了解各個授權條件。如需有關保護網路層的規範指引，請參閱保護網路。

   b. 持續監控與警示。

      AWS Identity and Access Management Access Analyzer 可協助您識別組織中與外部實體共用的資源。您可以將 IAM Access Analyzer 與 AWS Security Hub 整合，並將資源的發現結果從 IAM Access Analyzer 傳送並彙總到 Security Hub，以協助分析您環境的安全態勢。若要啟用整合，請在每個帳戶的每個區域中同時啟用 IAM Access Analyzer 和 Security Hub。您還可以使用 AWS Config 規則 來稽核設定，並使用 AWS Chatbot 與 AWS Security Hub 警告適當的一方。您接著可以使用 AWS Systems Manager 自動化文件來修復不合規的資源。

   c. 如需有關持續監控與警示外部共用的資源的規範指引，請參閱分析公開和跨帳戶存取權。

4. 使用 AWS 服務中的資源共用並適當限制。

   許多 AWS 服務都允許您與另一個帳戶共用資源，或鎖定另一個帳戶中的資源，例如 Amazon Machine Images (AMI) and AWS Resource Access Manager (AWS RAM)。限制 `ModifyImageAttribute` API 以指定可信帳戶來共用 AMI。當使用 AWS RAM 來限制僅共用至您的組織時，指定 `ram:RequestedAllowsExternalPrincipals` 條件來協助防止不受信任的身分的存取。相關規範指引和考量，請參閱資源共用和外部目標。

5. 使用 AWS RAM 在帳戶中或與其他 AWS 帳戶 安全地共用。

   AWS RAM 可幫助您安全地將您使用帳戶中的角色和使用者所建立的資源與 AWS 帳戶 共用。在多帳戶環境中，AWS RAM 可讓您建立資源一次並與其他帳戶共用。這個方法有助於降低營運負擔，同時透過與 Amazon CloudWatch 和 AWS CloudTrail 的整合提供一致性、能見度和可稽核性，這是在使用跨帳戶存取權時所沒有的。

   如果您擁有之前使用以資源為基礎的政策共用的資源，可以使用 `PromoteResourceShareCreatedFromPolicy API` 或同等項目將資源共用升級到完整 AWS RAM 資源共用。

在某些情況下，您可能需要採取額外步驟來共用資源。例如，要共用加密快照，您需要共用 AWS KMS 金鑰。

資源

相關的最佳實務：

- SEC03-BP07 分析公有和跨帳戶存取權
- SEC03-BP09 安全地與第三方共用資源
- SEC05-BP01 建立網路層

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
- 您可以搭配 AWS Organizations 使用的 AWS 服務
- 在 AWS 上建立資料周邊：僅允許可信身分存取公司資料

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

相關工具：

- 資料周邊政策範例

SEC03-BP09 安全地與第三方共用資源

您雲端環境的安全並不止於您的組織。您的組織可能仰賴第三方來管理您的部分資料。針對第三方管理的系統的許可管理應該遵循即時存取的做法，採用最低權限的原則搭配臨時憑證。透過與第三方密切合作，您可以同時減少影響範圍以及意外存取的風險。

預期成果：只要憑證有效且作用中，任何人都可以使用與使用者相關聯的長期 AWS Identity and Access Management (IAM) 憑證、IAM 存取金鑰和機密金鑰。使用 IAM 角色和臨時憑證可透過減輕維護長期憑證的工作 (包括管理這些敏感詳細資料的營運負擔)，協助改善您的整體安全態勢。透過在 IAM 信任政策中針對外部 ID 使用通用唯一識別符，以及控制附加到 IAM 角色的 IAM 政策，您可以稽核並確認授予第三方的存取權未過於寬鬆。如需有關分析外部共用的資源的規範指引，請參閱SEC03-BP07 分析公有和跨帳戶存取權。

常見的反模式：

- 無條件地使用預設 IAM 信任政策。
- 使用 IAM 憑證和存取金鑰。
- 重複使用外部 ID。

未建立此最佳實務時的風險暴露等級：中

實作指引

您可能會想要允許在 AWS Organizations 之外共用資源或將帳戶存取權授予第三方。例如，第三方可能提供監控解決方案，而該解決方案需要存取您帳戶中的資源。在該些情況下，僅以第三方需要的權限來建立 IAM 跨帳戶角色。此外，請使用外部 ID 條件定義信任政策。當使用外部 ID 時，您或第三方可以為每個客戶、第三方或租用戶產生唯一 ID。在建立唯一 ID 後，其不應該受除了您之外的任何人控制。第三方必須實作程序，以安全、可稽核且可重新產生的方式將外部 ID 與客戶關聯。

您還可以使用 IAM Roles Anywhere 為 AWS 之外使用 AWS API 的應用程式管理 IAM 角色。

如果第三方不再需要存取您的環境，請移除該角色。避免為第三方提供長期憑證。掌握對其他支援共用的 AWS 服務的狀態。例如，AWS Well-Architected Tool 允許與其他 AWS 帳戶 共用工作負載，而 AWS Resource Access Manager 有助您安全地與其他帳戶共用您擁有的 AWS 資源。

實作步驟

1. 使用跨帳戶角色提供存取權給外部帳戶。

   跨帳戶角色可減少外部帳戶和第三方為了服務客戶所儲存的敏感資訊量。跨帳戶角色允許您在帳戶中將 AWS 資源的存取權安全地授予第三方，例如 AWS Partner 或組織內的其他帳戶，同時維持管理和稽核該存取權的能力。

   第三方可能從混合式基礎設施為您提供服務，或將資料提取至異地。IAM Roles Anywhere 可幫助您啟用第三方工作負載，以安全地與您的 AWS 工作負載進行互動，並進一步減少使用長期憑證的需要。

您不應該使用與使用者相關聯的長期憑證或存取金鑰來提供外部帳戶存取權。反而應該使用跨帳戶角色來提供跨帳戶存取權。

2. 對第三方使用外部 ID。

   使用外部 ID 可讓您指定誰可以擔任在 IAM 信任政策中的角色。信任政策可以要求擔任該角色的使用者聲明他們操作的條件和目標，還提供方法讓客戶擁有者允許只能在特定情況下擔任角色。外部 ID 的主要功能是解決和防止混淆代理人問題。

   如果您是 AWS 帳戶 擁有者並且已為存取您的帳戶以及其他 AWS 帳戶 的第三方設定角色，或是當您代表不同客戶擔任角色時，請使用外部 ID。與您的第三方或 AWS Partner 合作建立要納入 IAM 信任政策中的外部 ID 條件。

3. 使用通用唯一外部 ID。

   實作為外部 ID 產生隨機唯一值的程序，例如通用唯一識別符 (UUID)。在不同客戶間重複使用外部 ID 的第三方並不會解決混淆代理人的問題，因為客戶 A 可能能夠使用客戶 B 的角色 ARN 搭配重複的外部 ID 來檢視客戶 B 的資料。在多租用戶環境中，第三方支援使用不同 AWS 帳戶 的多個客戶，因此第三方必須為各個 AWS 帳戶 使用不同的唯一 ID 作為外部 ID。第三方負責偵測重複的外部 ID 並安全地將各個客戶對應到其個別的外部 ID。第三方應該測試以確認他們只能在指定外部 ID 時擔任該角色。在要求使用外部 ID 之前，第三方不應該儲存客戶角色 ARN 和外部 ID。

   外部 ID 不會被視為機密，但外部 ID 不能是容易猜測的值，例如電話號碼、名稱或帳戶 ID。將外部 ID 設為唯讀欄位，而使外部 ID 不能為了冒充設定的目的而遭到變更。

   您或第三方可以產生外部 ID。定義程序以決定由誰負責產生 ID。無論建立外部 ID 的實體為何，第三方都要在客戶間一致地強制唯一性和格式。

4. 棄用客戶提供的長期憑證。

   棄用長期憑證並使用跨帳戶角色或 IAM Roles Anywhere。如果您必須使用長期憑證，請制定計畫以遷移至角色型存取。如需有關管理金鑰的詳細資訊，請參閱身分管理。另外也與您的 AWS 帳戶 團隊和第三方合作建立風險緩解執行手冊。如需有關回應和緩解潛在安全事件的衝擊的規範指引，請參閱事件回應。

5. 確認該設定具有規範指引且已自動化。

   您的帳戶中為跨帳戶存取權建立的政策必須遵循最低權限原則。第三方必須提供角色政策文件，或使用 AWS CloudFormation 範本或對您來說同等的自動設定機制。這可減少發生與手動政策建立相關聯之錯誤的機率，並提供可稽核的記錄。如需有關使用 AWS CloudFormation 範本來建立跨帳戶角色的詳細資訊，請參閱跨帳戶角色。

第三方應該提供自動化、可稽核的設定機制。然而，透過使用概述所需存取權的角色政策文件，您應該可自動設定角色。使用 AWS CloudFormation 範本或同等項目，您應該透過偏移偵測來監控變更，以作為稽核實務的一部份。

6. 將變更列入考量。

   您的帳戶結構、對第三方的需求或他們提供的服務方案可能發生變更。您應該預期變更和失敗，並透過合適的人員、程序和技術相應進行規劃。定期稽核您提供的存取層級，並實作偵測方法以在發生意外變更時通知您。監控和稽核角色和外部 ID 資料儲存的使用。您應該準備好在發生意外變更或存取模式時撤銷第三方存取權，無論是暫時或永久撤銷。另外，衡量撤銷作業的衝擊，包括執行所花的時間、牽涉的人員、成本，以及對其他資源的衝擊。

   如需有關偵測方法的規範指引，請參閱偵測最佳實務。

資源

相關的最佳實務：

- SEC02-BP02 使用臨時憑證
- 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
- SEC04 偵測

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 角色
- 使用 IAM 角色在 AWS 帳戶間委派存取權
- 如何使用 IAM 存取另一個 AWS 帳戶中的資源？
- IAM 中的安全最佳實務
- 跨帳戶政策評估邏輯
- 如何在向第三方授予對您的 AWS 資源的存取權時使用外部 ID
- 從在外部帳戶中使用自訂資源建立的 AWS CloudFormation 資源收集資訊
- 安全地使用外部 ID 存取其他人擁有的 AWS 帳戶

- 使用 IAM Roles Anywhere 將 IAM 角色擴展到 IAM 之外的工作負載

相關影片：

- 如何允許不同 AWS 帳戶中的使用者或角色存取我的 AWS 帳戶？
- AWS re:Invent 2018：在 60 分鐘內精通 IAM 政策
- AWS 知識中心直播：IAM 最佳實務和設計決策

相關範例：

- Well-Architected 實驗室 - Lambda 跨帳戶 IAM 角色擔任 (Level 300)
- 設定 Amazon DynamoDB 跨帳户存取權
- AWS STS 網路查詢工具

偵測

問題

- SEC 4.如何偵測和調查安全事件？

SEC 4.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 在標準化的位置擷取日誌、調查結果和指標
- SEC04-BP03 建立安全警示的相互關聯並增添其豐富性
- SEC04-BP04 針對不合規資源實施補救措施

SEC04-BP01 設定服務和應用程式記錄

保留服務和應用程式的安全事件日誌這是稽核、調查和操作使用案例的基礎原則，以及由管控、風險和合規 (GRC) 標準、政策和程序所推動的常見安全需求。

預期成果：組織應該能夠在需要執行內部程序或義務時，例如安全事件回應，以可靠且一致的方式及時從 AWS 服務和應用程式擷取安全事件日誌。考慮集中日誌以達到更佳的營運成果。

常見的反模式：

- 日誌存放太久或太早刪除。
- 每個人都能存取日誌。
- 日誌的管控和使用完全仰賴手動程序。
- 儲存每一種日誌以備不時之需。
- 只在必要時檢查日誌完整性。

建立此最佳實務的優勢：對安全事件和證據來源實作根本原因分析 (RCA) 機制，以履行管控、風險和合規義務。

未建立此最佳實務時的風險暴露等級：高

實作指引

根據您的需求進行安全調查或其他使用案例期間，您需要能夠審查相關日誌以記錄和了解該事件的全部範圍和時間表。產生警示也需要日誌，以指出特定關注的動作已發生。選擇、啟用、儲存和設定查詢與擷取機制和警示至關重要。

實作步驟

- 選擇和啟用日誌來源。 在安全調查之前，您需要擷取相關日誌以追溯的方式重新建構 AWS 帳戶 中的活動。選擇並啟用與您的工作負載相關的日誌來源。

  日誌來源選擇條件應該根據您的業務所需的使用案例。使用 AWS CloudTrail 或 AWS Organizations 線索為每個 AWS 帳戶 建立線索，以及為其設定 Amazon S3 儲存貯體。

  AWS CloudTrail 是一種記錄服務，會追蹤對 AWS 帳戶 的 API 呼叫以擷取 AWS 服務活動。預設啟用時，此服務會保留 90 天的管理活動，而其能以 AWS Management Console、AWS CLI 或 AWS SDK 透過 CloudTrail 事件歷史記錄擷取。如需較長的保留期間和資料事件的能見度，可建立 CloudTrail 線索並將其與 Amazon S3 儲存貯體建立關聯，也可以選擇與 Amazon CloudWatch 日誌群組相關聯。或者，您可以建立 CloudTrail Lake，這會將 CloudTrail 日誌保留長達七年，並提供以 SQL 為基礎的查詢設施。

  AWS 建議使用 VPC 的客戶分別使用 VPC Flow Logs 和 Amazon Route 53 解析器查詢日誌來啟用網路流量和 DNS 日誌，並將它們串流處理到 Amazon S3 儲存貯體或 CloudWatch 日誌群組。您可

以為 VPC、子網路或網路介面建立 VPC 流程日誌。對於 VPC Flow Logs，您可以選擇何時何地使用 Flow Logs 來降低成本。

AWS CloudTrail 日誌、VPC Flow Logs 和 Route 53 解析器查詢日誌是在 AWS 中支援安全調查的基本記錄來源。您還可以使用 Amazon Security Lake 以 Apache Parquet 格式和 Open Cybersecurity Schema Framework (OCSF) 收集、正規化並儲存此日誌資料，此種格式隨時可供查詢。Security Lake 還支援其他 AWS 日誌以及來自第三方來源的日誌。

AWS 服務可產生基本日誌來源未擷取的日誌，例如 Elastic Load Balancing 日誌、AWS WAF 日誌、AWS Config 記錄器日誌、Amazon GuardDuty 發現結果、 Amazon Elastic Kubernetes Service (Amazon EKS) 稽核日誌和 Amazon EC2 執行個體作業系統及應用程式日誌。如需記錄和監控選項的完整清單，請參閱《AWS 安全事件回應指南》的附錄 A：雲端功能定義 – 記錄和事件。

- 每個 AWS 服務和應用程式的研究記錄功能： 每個 AWS 服務和應用程式都為您提供日誌儲存的選項，而其各自有自己的保留和生命週期功能。兩個最常見的日誌儲存服務是 Amazon Simple Storage Service (Amazon S3) 和 Amazon CloudWatch。如需長期保留，建議使用具成本效益和彈性生命週期功能的 Amazon S3。若主要的記錄選項是 Amazon CloudWatch 日誌，您應該考慮將較不常存取的日誌封存到 Amazon S3，作為一種選項。
- 選擇日誌儲存：日誌儲存的選擇通常與您使用的查詢工具、保留功能、熟悉度和成本有關。日誌儲存的主要選項是 Amazon S3 儲存貯體或 CloudWatch 日誌群組。

  Amazon S3 儲存貯體提供符合成本效益、耐用的儲存方式，並且具備可選擇的生命週期政策。存放在 Amazon S3 儲存貯體的日誌可使用 Amazon Athena 之類的服務進行查詢。

  CloudWatch 日誌群組透過 CloudWatch Logs Insights 提供耐用的儲存方式和內建查詢設施。
- 識別適當的日誌保留時間 ：當您使用 Amazon S3 儲存貯體或 CloudWatch 日誌群組來存放日誌時，您必須為每個日誌來源建立充分的生命週期，以最佳化儲存和擷取成本。客戶一般擁有三個月到一年的時間使日誌隨時可供查詢，並且最長可保留七年。可用性和保留時間的選擇應該配合您的安全需求與各種法令、法規和業務規定。
- 依照適當的保留和生命週期政策為每個 AWS 服務和應用程式啟用記錄功能：對於組織內的每個 AWS 服務或應用程式，尋找特定的記錄設定指引：
  - 設定 AWS CloudTrail 線索
  - 設定 VPC Flow Logs
  - 設定 Amazon GuardDuty 發現結果匯出
  - 設定 AWS Config 記錄
  - 設定 AWS WAF Web ACL 流量
  - 設定 AWS Network Firewall 網路流量日誌

  - 設定 Elastic Load Balancing 存取日誌
  - 設定 Amazon Route 53 解析器查詢日誌
  - 設定 Amazon RDS 日誌
  - 設定 Amazon EKS 控制平面日誌
  - 為 Amazon EC2 執行個體和內部部署伺服器設定 Amazon CloudWatch 代理程式
- 為日誌選擇並實作查詢機制：對於日誌查詢，您可以使用 CloudWatch Logs Insights (適用於存放在 CloudWatch 日誌群組中的資料) 以及 Amazon Athena 和 Amazon OpenSearch Service (適用於存放在 Amazon S3 中的資料)。您還可以使用第三方查詢工具，例如安全資訊和事件管理 (SIEM) 服務。

  選擇日誌查詢工具的過程中應該考慮安全營運的人員、程序和技術層面。選擇符合營運、業務和安全需求的工具，並且可供存取和長期維護。請記住，將要掃描的日誌數目維持在日誌查詢工具限制之內，以便以最佳狀態運作。因為成本或技術限制的關係，擁有多個查詢工具十分常見。

  例如，您可能使用第三方安全資訊和事件管理 (SIEM) 工具對過去 90 天的資料執行查詢，但基於 SIME 的日誌擷取成本，而使用 Athena 來執行超過 90 天的查詢。無論實作方式為何，請確認您的方法將所需的工具數量最小化以最大化營運效率，尤其是在安全事件調查期間。
- 使用日誌提供警示：AWS 透過數種安全服務提供警示功能：
  - 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，會持續監控惡意活動和未授權行為以保護 AWS 帳戶 和工作負載。GuardDuty 會擷取、彙總和分析來自如 AWS CloudTrail 管理和資料事件、DNS 日誌、VPC Flow Logs 和 Amazon EKS 稽核日誌等來源的資訊。GuardDuty 會直接從 CloudTrail、VPC Flow Logs、DNS 查詢日誌和 Amazon EKS 提取獨立資料串流。您不需要管理 Amazon S3 儲存貯體或修改您收集和儲存日誌的方式。仍舊建議您保留這些日誌，供自身調查和合規用途。
  - 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警示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。

  您還可以使用自訂警示產生引擎，取得這些服務未涵蓋的安全警示或與您的環境相關的特定警示。如需有關建立這些警示和偵測的資訊，請參閱《AWS 安全事件回應指南》中的偵測。

資源

相關的最佳實務：

- SEC04-BP02 在標準化的位置擷取日誌、調查結果和指標

- SEC07-BP04 定義可擴展的資料生命週期管理
- SEC10-BP06 預先部署工具

相關文件：

- AWS 安全事件應變指南
- Amazon Security Lake 入門
- 入門：Amazon CloudWatch Logs
- 安全合作夥伴解決方案：記錄與監控

相關影片：

- AWS re:Invent 2022 - Amazon Security Lake 簡介

相關範例：

- Assisted Log Enabler for AWS
- AWS Security Hub 發現結果歷史匯出

相關工具：

- Snowflake for Cybersecurity

SEC04-BP02 在標準化的位置擷取日誌、調查結果和指標

安全團隊依賴日誌和調查結果來分析可能代表未經授權活動或意外變更的事件。為了簡化此分析，您可在標準化的位置擷取安全日誌和調查結果。 這樣就能提供關注的資料點來建立相互關聯，並且可簡化工具整合。

預期成果：您採用標準化的方式來收集、分析和視覺化日誌資料、調查結果和指標。安全團隊能夠有效率地跨分散的系統建立相互關聯、分析和視覺化安全資料，藉此發現可能發生的安全事件並識別異常。安全資訊和事件管理 (SIEM) 系統或其他機制經整合後，即可查詢和分析日誌資料，以便及時回應、追蹤和向上呈報安全事件。

常見的反模式：

- 多個團隊各自擁有並管理記錄和指標收集工作，而他們的工作方式卻與組織的記錄策略不一致。

- 團隊沒有適當的存取控制可用來限制所收集資料的可見性和更改。
- 團隊未將控管安全日誌、調查結果和指標納入其資料分類政策中。
- 團隊在設定資料收集時，忽略了資料主權和本地化需求。

建立此最佳實務的優勢：擁有標準化的記錄解決方案可用來收集和查詢日誌資料和事件，如此就能改善從內含的資訊中產生的洞察。為收集的日誌資料設定自動化生命週期，可降低日誌儲存所伴隨的成本。您可以根據團隊所需的資料敏感度和存取模式，為收集的日誌資訊建置精細的存取控制。您可以整合工具來建立資料的相互關聯、視覺化資料，以及從資料中產生洞察。

未建立此最佳實務時的風險暴露等級：中

實作指引

隨著組織內 AWS 的使用增加，分散的工作負載和環境數量也會增加。由於這些工作負載和環境會各自產生其內部活動的相關資料，因此在本地擷取和儲存這些資料會為安全營運方面帶來挑戰。安全團隊會使用安全資訊和事件管理 (SIEM) 系統等工具從分散的來源收集資料，並進行相互關聯、分析和回應工作流程。這個過程需要管理一組複雜的許可來存取各種資料來源，還會在操作擷取、轉換和載入 (ETL) 程序上帶來額外的負擔。

為了克服這些挑戰，可考慮依照使用多個帳戶整理您的 AWS 環境所述，將所有相關的安全日誌資料來源彙總到日誌封存帳戶中。這包括來自您的工作負載和 AWS 服務所產生日誌的所有安全相關資料，例如 AWS CloudTrail、AWS WAF、Elastic Load Balancing 和 Amazon Route 53。在標準化的位置透過具有適當跨帳戶許可的個別 AWS 帳戶 擷取此資料有幾個好處。這種做法有助於防止受害的工作負載和環境內的日誌遭到竄改、可為其他工具提供單一整合點，並提供更簡化的模式來設定資料保留和生命週期。 評估資料主權、合規範圍和其他法規的影響，以判斷是否需要多個安全資料儲存位置和保留期。

為了輕鬆擷取和標準化日誌和調查結果，請在您的日誌封存帳戶中評估 Amazon Security Lake。您可以將 Security Lake 設定為自動從常見的來源擷取資料，例如 CloudTrail、Route 53、Amazon EKS 和 VPC Flow Logs。您也可以將 AWS Security Hub 設定為 Security Lake 中的資料來源，如此就能將其他 AWS 服務 (例如 Amazon GuardDuty 和 Amazon Inspector) 的調查結果與您的日誌資料相互關聯。 您也可以使用第三方資料來源整合，或設定自訂資料來源。所有整合都會將您的資料標準化為 Open Cybersecurity Schema Framework (OCSF) 格式，並以 Parquet 檔案形式儲存在 Amazon S3 儲存貯體中，因此不需要進行 ETL 處理。

將安全資料儲存在標準化的位置可提供進階分析功能。AWS 建議您將在 AWS 環境中操作的安全分析工具部署到安全工具帳戶中，與您的日誌封存帳戶加以區隔。 這種方法可讓您深入實作控制措施，以保護日誌和日誌管理程序的完整性和可用性，有別於用於存取日誌的工具。 考慮使用 Amazon

Athena 等服務來執行與多個資料來源相互關聯的隨需查詢。您也可以整合視覺化工具，例如 Amazon QuickSight。採用 AI 技術的解決方案越來越普遍可得，並且能夠執行許多功能，例如將調查結果轉譯成人類可讀的摘要以及自然語言互動等。 擁有標準化的資料儲存位置可供查詢，通常會更容易整合這些解決方案。

實作步驟

1. 建立日誌封存和安全工具帳戶
   a. 使用 AWS Organizations，在安全組織單位下建立日誌封存和安全工具帳戶。如果您使用 AWS Control Tower 來管理組織，則會自動為您建立日誌封存和安全工具帳戶。視需要設定存取和管理這些帳戶的角色與許可。
2. 設定標準化的安全資料位置
   a. 確定您用來建立標準化安全資料位置的策略。 您可以透過像是通用資料湖架構方法、第三方資料產品或 Amazon Security Lake 等選項達成此目的。AWS 建議您從為帳戶設為選擇加入的 AWS 區域 擷取安全資料 (即使沒有在作用中)。
3. 設定將資料來源發佈到標準化位置
   a. 識別安全資料的來源，並將它們設定為發佈到您的標準化位置。 評估以所需格式自動匯出資料的選項，而不是需要開發 ETL 程序的選項。 有了 Amazon Security Lake，您可以從受支援的 AWS 來源和經過整合的第三方系統收集資料。
4. 設定工具以存取標準化位置
   a. 設定 Amazon Athena、Amazon QuickSight 或第三方解決方案等工具，使其具備存取您的標準化位置所需的權限。 設定這些工具，以適時透過對日誌封存帳戶的跨帳戶讀取存取權在安全工具帳戶之外操作。在 Amazon Security Lake 中建立訂閱者，以便為這些工具提供對您資料的存取權。

資源

相關的最佳實務：

- SEC01-BP01 使用帳戶區隔工作負載
- SEC07-BP04 定義可擴展資料生命週期管理
- SEC08-BP04 強制存取控制
- OPS08-BP02 分析工作負載日誌

相關文件：

- AWS 白皮書：使用多個帳戶整理您的 AWS 環境

- AWS 規範性指引：AWS 安全性參考架構 (AWS SRA)
- AWS 規範性指引：應用程式擁有者的記錄和監控指南

相關範例：

- 使用 Amazon Athena 和 Amazon QuickSight 彙總、搜尋和視覺化來自分散來源的日誌資料
- 如何使用 Amazon QuickSight 視覺化 Amazon Security Lake 調查結果
- 使用 Amazon SageMaker Studio 和 Amazon Bedrock 為 Amazon Security Lake 產生 AI 技術支援的洞察
- 使用 Amazon SageMaker 識別 Amazon Security Lake 資料中的網路安全異常
- 接收和轉換 Amazon Security Lake 發佈的事件並傳遞至 Amazon OpenSearch Service
- 如何針對 SIEM 使用 AWS Security Hub 和 Amazon OpenSearch Service

相關工具：

- Amazon Security Lake
- Amazon Security Lake 合作夥伴整合
- Open Cybersecurity Schema Framework (OCSF)
- Amazon Athena
- Amazon QuickSight
- Amazon Bedrock

SEC04-BP03 建立安全警示的相互關聯並增添其豐富性

非預期的活動可能會導致不同來源產生多個安全警示，因此需要進一步在建立這些來源之間的相互關聯並增添豐富性，才能了解完整的內容。 實作安全警示的自動化相互關聯並增添其豐富性，有助於更準確地識別和回應事件。

預期成果：當活動在您的工作負載和環境內產生不同的警示時，自動化機制會建立資料的相互關聯，並使用其他資訊增添該資料的豐富性。此預先處理程序呈現了對事件更詳細的了解，進而幫助調查人員判斷事件的關鍵性，以及它是否構成需要正式回應的事件。此程序可減輕監控和調查團隊的負擔。

常見的反模式：

- 不同組合的人員對不同系統產生的調查結果和警示進行調查 (除非是在因職責區分需求而另有規定的情況下)。

- 您的組織將所有安全調查結果和警示資料收集到標準位置，但要求調查人員手動建立相互關聯和添加資訊。
- 您只依賴威脅偵測系統的情報來回報調查結果和確定關鍵性。

建立此最佳實務的優勢：自動建立警示的相互關聯和增添其豐富性，有助於減輕調查人員的整體認知負擔和手動準備資料的負荷。這種做法可縮短判斷事件是否為「事故」及正式回應所需的時間。額外的內容還可幫助您準確評估事件的嚴重性，因為實際的嚴重性可能高於或低於任何警示表明的嚴重性。

未建立此最佳實務時的風險暴露等級：低

實作指引

安全警示可能來自 AWS 內多個不同的來源，包括：

- Amazon GuardDuty、AWS Security Hub、Amazon Macie、Amazon Inspector、AWS Config、AWS Identity and Access Management Access Analyzer 和 Network Access Analyzer 等服務
- 來自 AWS 服務、基礎設施和應用程式日誌的自動化分析的警示，例如來自 Security Analytics for Amazon OpenSearch Service。
- 回應帳單活動變更的警報，來自下列來源：Amazon CloudWatch、Amazon EventBridge 或 AWS Budgets。
- 第三方來源，例如來自 AWS Partner Network 的威脅情報摘要和安全合作夥伴解決方案
- 來自 AWS 信任與安全或其他來源的聯絡資訊，例如客戶或內部員工。

警示基本上包含有關誰 (主體或身分)、做了什麼 (採取的行動)，以及對象是誰 (受影響的資源) 的資訊。對於每個來源，請確定是否有能夠在這些身分、動作和資源的識別符之間建立映射的方式，以作為建立相互關聯的基礎。可能的形式包括整合警示來源與安全資訊和事件管理 (SIEM) 工具，以便為您自動建立相互關聯、建置您自己的資料管道和處理流程，或結合上述兩者。

可為您建立相互關聯的服務範例為 Amazon Detective。偵測會持續接收來自各種 AWS 和第三方來源的警示，並使用不同形式的情報來構成視覺化圖形以呈現其關係，進而協助調查。

雖然警示的初始關鍵性可協助判斷優先順序，但警示發生的環境則決定了其真實的關鍵性。舉例來說，Amazon GuardDuty 可能針對您工作負載內的 Amazon EC2 執行個體正在查詢非預期的網域名稱而發出警示。GuardDuty 可能會自行對此警示指派「低關鍵性」標簽。然而，若在警示發出之時與其他活動自動建立相互關聯，您就可能會發現有數百個 EC2 執行個體是由相同身分部署的，這種情況會

增加整體營運成本。在此事件下，GuardDuty 可能會將此相互關聯事件內容發佈為新的安全警示，並將關鍵性調整為高，進而加速採取進一步行動。

實作步驟

1. 識別安全警示資訊的來源。了解來自這些系統的警示如何表示身分、動作和資源，以判斷可能的相互關聯。
2. 制定一個機制來擷取不同來源的警示。考慮用於這類用途的服務，例如 Security Hub、EventBridge 和 CloudWatch。
3. 識別資料相互關聯和增添豐富性的來源。範例來源包括 CloudTrail、VPC Flow Logs、Amazon Security Lake，以及基礎設施和應用程式日誌。
4. 將警示與資料相互關聯和增添豐富性的來源整合在一起，以創造更詳細的安全事件內容並構成關鍵性。
   a. Amazon Detective、SIEM 工具或其他第三方解決方案可以自動執行特定層級的擷取、相互關聯和豐富性。
   b. 您也可以使用 AWS 服務來自行建置。例如，您可以調用 AWS Lambda 函數來對 AWS CloudTrail 或 Amazon Security Lake 執行 Amazon Athena 查詢，並將結果發佈至 EventBridge。

資源

相關的最佳實務：

- SEC10-BP03 準備鑑識功能
- OPS08-BP04 建立可付諸行動的警示
- REL06-BP03 傳送通知 (即時處理和警示)

相關文件：

- AWS 安全事件應變指南

相關範例：

- 如何利用帳戶中繼資料增添 AWS Security Hub 調查結果的豐富性
- 如何針對 SIEM 使用 AWS Security Hub 和 Amazon OpenSearch Service

相關工具：

- Amazon Detective
- Amazon EventBridge
- AWS Lambda
- Amazon Athena

SEC04-BP04 針對不合規資源實施補救措施

您的偵測控制措施可能針對不符合您組態需求的資源發出警示。您可以手動或自動實施以程式設計方式定義的補救措施，以修正這些資源並協助盡可能將影響降到最低。當您以程式設計方式定義補救措施時，您可以採取快速且一致的行動。

雖然自動化可以增強安全操作，但您應謹慎實作和管理自動化程序。 設置適當的監督和控制機制，以確認自動化回應是否有效、準確，且合乎組織政策和風險偏好。

預期成果：您會定義資源組態標準，以及在偵測到資源不合規時的修復步驟。在可能的情況下，您已透過程式設計方式定義補救措施，以供人員手動或透過自動化方式啟動。已設置偵測系統，其可識別不合規的資源，並在由安全人員監控的集中式工具中發佈警示。這些工具支援手動或自動執行程式設計的補救措施。自動化補救措施設有適當的監督和控制機制來控管理其使用。

常見的反模式：

- 您實作自動化，但未徹底測試和驗證補救動作。這可能會導致意外的後果，例如中斷正當的業務營運或導致系統不穩定。
- 您透過自動化改善回應時間和程序，但未設置適當的監控與機制，無法在需要時允許人為介入和判斷。
- 您只依賴補救措施，而不是將補救措施視為更廣泛的事件回應和復原計劃的一部分。

建立此最佳實務的優勢：自動化補救措施能夠比手動流程更快回應組態錯誤，進而有助於將可能對業務造成的影響降至最低，並且減少意外使用的機會。 當您以程式設計方式定義補救措施時，就能一致套用這些措施，進而降低人為錯誤的風險。自動化還可同時處理更大量的警示，這點對於大規模操作的環境來說尤其重要。

未建立此最佳實務時的風險暴露等級：中

實作指引

如 SEC01-BP03 識別和驗證控制目標 中所述，像是 AWS Config 等服務可幫助您監控帳戶中資源的組態，以確保符合您的需求。 在偵測到不合規資源時，建議您設定傳送警示至雲端安全狀態管理 (CSPM) 解決方案，例如 AWS Security Hub，以協助修復。這些解決方案為您的安全調查人員提供了一個集中的位置，方便監控問題並採取矯正行動。

有些不合規資源的情況獨特，需要人為判斷來進行修復，有些情況則有標準回應，您可透過程式設計方式定義這類回應。例如，對於設定錯誤的 VPC 安全群組，其標準回應可能是移除不允許的規則並通知擁有者。您可以在 AWS Lambda 函數中、AWSSystems Manager Automation 文件中，或透過您慣用的其他程式碼環境來定義回應。務必確定環境能夠使用具有採取矯正行動所需之最低許可權的 IAM 角色來對 AWS 進行身分驗證。

定義所需的補救措施後，您就可以決定您偏好的補救措施實施方法。AWS Config 可以為您實施補救措施。如果您使用 Security Hub，則可透過自訂動作來執行此操作，該動作會將調查結果資訊發佈至 Amazon EventBridge。然後 EventBridge 規則就能實施您的補救措施。您可以在 Security Hub 中設定自動或手動執行自訂動作。

對於程式化的補救措施，建議您留存所執行動作的完整日誌和稽核，以及其結果。 檢閱並分析這些日誌，以評估自動化流程的有效性，並找出改進之處。擷取 Amazon CloudWatch Logs 中的日誌，以及 Security Hub 中作為調查結果備註的修復結果。

一開始可參考 AWS 上的自動化安全性回應，其中包含預先建置的補救措施，可用來解決常見的安全錯誤組態。

實作步驟

1. 分析警示並排定優先順序。
    a. 將來自各種不同 AWS 服務的安全警示合併到 Security Hub 中，以提供集中查看、排定優先順序和修復的方式。
2. 制定補救措施。
    a. 使用 Systems Manager 和 AWS Lambda 等服務來執行程式化的補救措施。
3. 設定實施補救措施的方式。
    a. 使用 Systems Manager 定義將調查結果發佈到 EventBridge 的自訂動作。設定手動或自動啟動這些動作。
    b. 您也可以使用 Amazon Simple Notification Service (SNS) 傳送通知和警示給相關的利害關係人 (例如，安全團隊或事件應變團隊)，以便在必要時進行人為介入或向上呈報。
4. 檢閱並分析補救措施日誌，以了解有效性和改進之處。

a. 將日誌輸出傳送至 CloudWatch Logs。擷取 Security Hub 中作為調查結果備註的結果。

資源

相關的最佳實務：

- SEC06-BP03 減少手動管理和互動式存取

相關文件：

- AWS 安全事件應變指南 - 偵測

相關範例：

- AWS 上的自動化安全性回應
- 使用 AWS Config 監控 EC2 執行個體金鑰對
- 使用 AWS CloudFormation Guard 政策建立 AWS Config 自訂規則
- 自動修復未加密的 Amazon RDS 資料庫執行個體和叢集

相關工具：

- AWS Systems Manager Automation
- AWS 上的自動化安全性回應

基礎設施保護

問題

- SEC 5.如何保護您的網路資源？
- SEC 6.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 控制網路層內的流量流程
- SEC05-BP03 實作檢測型防護措施
- SEC05-BP04 自動化網路保護

SEC05-BP01 建立網路層

以工作負載元件的邏輯群組為基礎，根據其資料敏感性和存取需求將您的網路拓樸區分成不同層。將需要從網際網路進行傳入存取的元件 (例如公用 Web 端點) 和只需要內部存取的元件 (例如資料庫) 加以區分。

預期成果：您的網路層是整體安全深度防禦方法的一部分，可與工作負載的身分驗證和授權策略相輔相成。根據資料敏感性和存取需求妥善區分的網路層，且具有適當的流量流程和控制機制。

常見的反模式：

- 您在單一 VPC 或子網路中建立所有資源。
- 您建構網路層時未考量資料敏感性需求、元件行為或功能。
- 您將 VPC 和子網路作為所有網路層考量的預設選項，而且未考慮 AWS 受管服務對拓樸的影響。

建立此最佳實務的優勢：建立網路層是透過網路限制非必要路徑的第一步，尤其是前往關鍵系統和資料的路徑。這可讓未經授權的行為者更難存取您的網路並瀏覽至網路內的其他資源。分散的網路層有利於縮小檢測系統的分析範圍，例如針對入侵偵測或防範惡意軟體。這樣可減少誤報和不必要的處理負擔。

未建立此最佳實務時的風險暴露等級：高

實作指引

在設計工作負載架構時，根據元件的責任將其劃分至不同層是常見的做法。例如，Web 應用程式可擁有呈現層、應用程式層和資料層。您可以在設計網路拓樸時採用類似的方法。基礎網路控制措施可協助強制執行工作負載的資料存取需求。例如，在擁有三層的 Web 應用程式架構中，您可以將靜態呈現層檔案儲存在 Amazon S3 上，並從內容交付網路 (CDN) (例如 Amazon CloudFront) 提供它們。應用程式層可以包含 Application Load Balancer (ALB) 在 Amazon VPC 公有子網路 (類似非軍事區，DMZ) 中提供的公有端點，且加上部署到私有子網路中的後端服務。託管資料庫和共用檔案系統等資源的資料層，可位於與應用程式層的資源所在位置不同的私有子網路。在這些層的邊界 (CDN、公有子網路、私有子網路)，您可以部署控制措施，藉此僅允許授權的流量跨越這些邊界。

如同根據工作負載元件的功能用途建立網路層模型，請一併考量要處理的資料敏感性。以 Web 應用程式為例，雖然您所有的工作負載服務可能都位於應用程式層內，但不同的服務可能會處理不同敏感程度的資料。在這種情況下，根據您的控制需求，針對每一種程度的資料敏感性使用多個私有子網路、相同 AWS 帳戶 中的不同 VPC，甚至是不同 AWS 帳戶 中的不同 VPC 來區分應用程式層較為適當。

網路層的進一步考量是工作負載元件的行為一致性。繼續以上述範例說明，在應用程式層中，您可能有接受來自最終使用者或外部系統整合輸入的服務，而導向這些服務的輸入在本質上比對其他服務的輸入帶有更高風險。範例包括檔案上傳、要執行的程式碼指令碼、電子郵件掃描等。將這些服務放置在其自己的網路層中，有助於在其周圍建立更強大的隔離邊界，並可防止其獨特行為造成檢測系統中的誤報警示。

在設計過程中，請考慮使用 AWS 受管服務對網路拓樸的影響。探索像是 Amazon VPC Lattice 等服務如何讓您更輕鬆地跨網路層實現工作負載元件的互通性。使用 AWS Lambda 時，請在您的 VPC 子網路中部署，除非受到特定原因限制而無法這樣做。 判斷 VPC 端點和 AWS PrivateLink 可在哪些地方簡化遵循限制網際網路閘道存取的安全政策。

實作步驟

1. 檢閱您的工作負載架構。根據元件和服務提供的功能、處理的資料敏感性以及其行為，將其邏輯分組。
2. 對於回應網際網路請求的元件，請考慮使用負載平衡器或其他 Proxy 來提供公有端點。探索如何使用像是 CloudFront、Amazon API Gateway、Elastic Load Balancing 和 AWS Amplify 等受管服務轉移安全控制措施，以託管公有端點。
3. 對於在運算環境中執行的元件 (例如 Amazon EC2 執行個體、AWS Fargate 容器或 Lambda 函數)，一開始即根據您的群組將這些元件部署到私有子網路中。
4. 對於全受管 AWS 服務，例如 Amazon DynamoDB、Amazon Kinesis 或 Amazon SQS，請考慮使用 VPC 端點作為透過私有 IP 位址存取的預設方式。

資源

相關的最佳實務：

- REL02 規劃您的網路拓撲
- PERF04-BP01 了解聯網如何影響效能

相關影片：

- AWS re:Invent 2023 - AWS 聯網基礎

相關範例：

- VPC 範例
- 使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer 在 Amazon ECS 上以私有方式存取容器應用程式
- 使用 Amazon CloudFront 在 Amazon S3 儲存貯體中透過 VPC 提供靜態內容

SEC05-BP02 控制網路層內的流量流程

在網路層內，使用進一步的分隔方式，將流量限於每個工作負載所需的流程。首先，專注於控制網際網路或其他外部系統到工作負載與您的環境之間的流量 (南北流量)。接著查看不同元件和系統之間的流量 (東西流量)。

預期成果：您只允許工作負載的元件所需的網路流程互相通訊，以及與其用戶端和其相依的任何其他服務進行通訊。您的設計考量到公有與私有輸入和輸出之間的比較、資料分類、區域法規以及通訊協定要求等因素。在設計最低權限原則的過程中，盡可能採用點對點流量，而非網路對等。

常見的反模式：

- 您採用以周邊為基礎的網路安全方法，只控制網路層邊界處的流量。
- 您假設網路層內的所有流量都經過驗證和授權。
- 您只對輸入流量或輸出流量實施控制措施，而不是對兩者都實施。
- 您只依賴網路元件和網路控制來驗證和授權流量。

建立此最佳實務的優勢：這種做法有助於降低網路內發生未經授權行動的風險，並且為您的工作負載增加一層額外的授權。藉由實施流量流程控制，您就可以限制安全事件的影響範圍，並加快偵測和回應速度。

未建立此最佳實務時的風險暴露等級：高

實作指引

雖然網路層有助於在具有類似功能、資料敏感性層級和行為的工作負載元件周圍建立邊界，但您可以利用一些技術進一步區隔這些層內的元件，藉此建立遵循最低權限原則且更精細的流量控制層級。 在 AWS 內，網路層主要是根據 Amazon VPC 內的 IP 位址範圍，使用子網路定義。您也可以使用不同的 VPC 定義網路層，其用途包括根據業務領域將微服務環境分組等。使用多個 VPC 時，務必使用 AWS Transit Gateway 調解路由。 雖然這可在第 4 層 (IP 位址和連接埠範圍) 使用安全群組和路由表提供流

量控制，但您可以使用 AWS PrivateLink、Amazon Route 53 Resolver DNS 防火牆、AWS Network Firewall 和 AWS WAF 等其他服務進一步控制流量。

針對連線啟動方、連接埠、通訊協定和網路層等方面，了解並清查工作負載的資料流程和通訊需求。 評估可用於建立連線和傳輸資料的通訊協定，以選取符合您的防護需求的通訊協定 (例如 HTTPS 而非 HTTP)。在網路邊界和每一層內擷取這些需求。確定這些需求後，探索僅允許必要的流量流經每個連線點的選項。一開始在 VPC 內使用安全群組會是合適的選擇，因為安全群組可以附加至使用彈性網路介面 (ENI) 的資源，例如 Amazon EC2 執行個體、Amazon ECS 任務、Amazon EKS Pod 或 Amazon RDS 資料庫。與 Layer 4 防火牆不同的是，安全群組可以設置一項規則來依識別符允許來自另一個安全群組的流量，藉此盡量減少群組內的資源隨時間改變所需的更新。您也可以使用安全群組的輸入和輸出規則來篩選流量。

當流量在 VPC 之間移動時，針對簡便路由使用 VPC 對等或針對複雜路由使用 AWS Transit Gateway 的情況很常見。使用這些方法可讓來源和目的地網路的 IP 位址範圍之間的流量更順暢。然而，如果您的工作負載只需要讓流量在不同 VPC 中的特定元件之間流動，則可考慮使用 AWS PrivateLink 的點對點連線。若要這樣做，請確定哪些服務應作為生產者，哪些服務應作為取用者。為生產者部署相容的負載平衡器，對應地開啟 PrivateLink，然後接受取用者的連線請求。 接著從取用者的 VPC 指派私有 IP 位址給生產者服務，取用者可使用該位址提出後續請求。這種方法減少了對等網路的需求。在評估 PrivateLink 的過程中，請納入資料處理和負載平衡的成本。

安全群組和 PrivateLink 有助於控制工作負載的元件之間的流量，而另一項重要的考量則是如何控制允許您的資源存取哪些 DNS 網域 (如有的話)。 根據您 VPC 的 DHCP 組態而定，您可以考慮兩種不同的 AWS 服務來達成此目的。 大多數客戶會使用在 CIDR 範圍的 +2 位址供 VPC 使用的預設 Route 53 Resolver DNS 服務 (也稱為 Amazon DNS 伺服器或 AmazonProvidedDNS)。使用這種方法可建立 DNS 防火牆規則，並將它們與 VPC 建立關聯，以決定要對您提供的網域清單執行哪些動作。

如果您不要使用 Route 53 Resolver，或是想要用網域篩選以外更深入的檢測和流程控制功能來輔助 Resolver，請考慮部署 AWS Network Firewall。此服務會使用無狀態或有狀態規則來檢測個別封包，以判斷是否拒絕或允許流量。您可以採用類似的方法，使用 AWS WAF 篩選前往公有端點的輸入 Web 流量。如需有關這些服務的進一步指引，請參閱 SEC05-BP03 實作檢測型防護措施。

實作步驟

1. 確定工作負載元件之間的必要資料流程。
2. 藉由對輸入和輸出流量採用深度防禦方法，套用多項控制措施，包括使用安全群組和路由表。
3. 使用防火牆對 VPC 上輸入、輸出和 VPC 之間的網路流量定義精細的控制措施，例如 Route 53 Resolver DNS 防火牆、AWS Network Firewall 和 AWS WAF。考慮使用 AWS Firewall Manager 集中設定和管理整個組織的防火牆規則。

資源

相關的最佳實務：

- REL03-BP01 選擇如何劃分工作負載
- SEC09-BP02 強制執行傳輸中加密

相關文件：

- VPC 的安全最佳實務
- AWS 網路最佳化要訣
- AWS 上的網路安全指引
- 在 AWS 雲端 中保護您 VPC 的輸出網路流量

相關工具：

- AWS Firewall Manager

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
- AWS re:Inforce 2023：防火牆和設置位置

相關範例：

- 實驗室：適用於 Web 應用程式的 CloudFront

SEC05-BP03 實作檢測型防護措施

在網路層之間設定流量檢測點，以確保傳輸中的資料符合預期的類別和模式。 分析流量流程、中繼資料和模式，以協助更有效地識別、偵測及回應事件。

預期成果：在網路層之間穿梭的流量會經過檢測和授權。 允許和拒絕的決定取決於明確的規則、威脅情報以及偏離基準行為的程度。 流量越接近敏感資料，防護就會越嚴格。

常見的反模式：

- 僅依賴以連接埠和通訊協定為準的防火牆規則。未善加利用情報系統。
- 根據可能隨時變更的特定現行威脅模式制訂防火牆規則。
- 僅檢測從私有子網路傳輸到公有子網路的流量，或從公有子網路傳輸到網際網路的流量。
- 沒有網路流量基準點可比較，以找出行為異常。

建立此最佳實務的優勢：檢測系統可讓您制訂智慧型規則，例如，只有在流量資料內存在特定條件時，才允許或拒絕流量。隨著威脅態勢經過一段時間而改變，根據最新的威脅情報受益於來自 AWS 和合作夥伴的受管規則集。 這可減輕維護規則和研究入侵指標的工作負擔，進而降低誤報的可能性。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用 AWS Network Firewall，或 AWS Marketplace 上可部署於 (GWLB) 後方的其他防火牆和入侵預防系統 (IPS)，對您的有狀態和無狀態網路流量實施精細的控制。AWS Network Firewall 支援使用與 Suricata 相容的開放原始碼 IPS 規格，來協助您保護工作負載。

使用 GWLB 的 AWS Network Firewall 和廠商解決方案都支援不同的內嵌檢測部署模型。 例如，您可以對每個 VPC 執行檢測、集中於某一檢測 VPC，或在東西流量會流經檢測 VPC 且每個 VPC 都會進行網際網路傳入檢測的混合模式中部署。 另一項考量是，解決方案是否支援解除 Transport Layer Security (TLS) 包裝，以便對任一方向的流量進行深度封包檢測。如需這些組態的相關資訊和深入資訊，請參閱 AWS Network Firewall 最佳實務指南。

如果您使用的解決方案會執行頻外檢測，例如，對來自在混雜模式下運作之網路介面的封包資料進行 pcap 分析，則您可以設定 VPC 流量鏡像。鏡像流量會計入介面的可用頻寬，並且您需支付與非鏡像流量相同的資料傳輸費用。您可以查看 AWS Marketplace 上是否有這些設備的虛擬版本可用，其可能支援 GWLB 後方的內嵌部署。

對於透過 HTTP 型通訊協定交易的元件，請使用 Web 應用程式防火牆 (WAF) 保護您的應用程式不受常見威脅的侵害。AWS WAF 是一種 Web 應用程式防火牆，可讓您在將 HTTP(S) 請求傳送至 Amazon API Gateway、Amazon CloudFront、AWS AppSync 或 Application Load Balancer 之前監控請求，並且在符合您的可設定規則時封鎖請求。在您評估部署 Web 應用程式防火牆時，請考慮深度封包檢測，因為有些防火牆可能會要求您在流量檢測之前先終止 TLS。若要開始使用 AWS WAF，您可以將 AWS 受管規則 與自己的規則結合，或使用現有的合作夥伴整合。

您可以使用 AWS Firewall Manager 集中管理整個 AWS 組織內的 AWS WAF、AWS Shield Advanced、AWS Network Firewall 和 Amazon VPC 安全群組。

實作步驟

1. 判斷您是否可以設定廣泛的檢測規則範圍，例如透過檢測 VPC，或是需要更精細、因個別 VPC 而異的方法。
2. 對於內嵌檢測解決方案：
   a. 如果使用 AWS Network Firewall，請建立規則、防火牆政策及防火牆本身。上述這些設定完成後，您可以將流量路由至防火牆端點以啟用檢測。
   b. 如果使用具有 Gateway Load Balancer (GWLB) 的第三方設備，請在一或多個可用區域中部署並設定您的設備。然後建立您的 GWLB、端點服務、端點，並設定流量的路由。
3. 對於頻外檢測解決方案：
   1. 在應鏡像處理輸入和輸出流量的介面上開啟 VPC 流量鏡像。您可以使用 Amazon EventBridge 規則來調用 AWS Lambda 函數，以便在建立新資源時開啟介面上的流量鏡像。將流量鏡像工作階段指向處理流量的設備前方的 Network Load Balancer。
4. 對於輸入 Web 流量解決方案：
   a. 若要設定 AWS WAF，首先請設定 Web 存取控制清單 (Web ACL)。Web ACL 是一個規則集合，當中包含循序處理的預設動作 (ALLOW 或 DENY)，此動作會定義 WAF 處理流量的方式。您可以在 Web ACL 中建立自己的規則和群組，或使用 AWS 受管規則群組。
   b. Web ACL 設定完成後，請將 Web ACL 與 AWS 資源 (例如 Application Load Balancer、API Gateway REST API 或 CloudFront 分佈) 建立關聯，以開始保護 Web 流量。

資源

相關文件：

- 什麼是流量鏡像？
- 使用第三方安全設備實作內嵌流量檢測
- AWS Network Firewall 範例架構與路由
- 具有 AWS Gateway Load Balancer 和 AWS Transit Gateway 的集中式檢測架構

相關範例：

- 部署 Gateway Load Balancer 的最佳實務
- 適用於加密傳出流量和 AWS Network Firewall 的 TLS 檢測組態

相關工具：

- AWS Marketplace IDS/IPS

SEC05-BP04 自動化網路保護

使用 DevOps 實務 (例如基礎設施即程式碼 (IaC)) 自動部署網路防護措施。 這些實務可協助您透過版本控制系統追蹤網路防護措施中的變更、縮短部署變更所需的時間，並協助您偵測網路防護措施是否偏離所需的組態。

預期成果：您會使用範本定義網路防護措施，並將其送交至版本控制系統中。 當做出新的變更時，自動化管道會因此而啟動，以協調其測試和部署。 設置了政策檢查和其他靜態測試，可在部署前驗證變更。 您可以將變更部署到模擬環境中，以驗證控制措施是否如預期運作。 控制措施得到核准後，也會自動將其部署到實際執行環境中。

常見的反模式：

- 倚賴個別工作負載團隊各自定義自己的整套網路堆疊、防護措施和自動化程序。 未集中發佈網路堆疊和防護措施的標準層面，供工作負載團隊取用。
- 倚賴中央網路團隊來定義網路、防護措施和自動化的所有層面。 未將網路堆疊和防護措施的工作負載特定層面委派給該工作負載的團隊。
- 集中和委派的情況在網路團隊與工作負載團隊之間達到適當的平衡，但未對 IaC 範本和 CI/CD 管道整體實施一致的測試和部署標準。 未在工具中擷取所需的組態，以致無法檢查範本是否遵循規範。

建立此最佳實務的優勢：使用範本定義網路防護措施，可讓您使用版本控制系統追蹤和比較一段時間的變化。 使用自動化方式測試和部署變更可建立標準化和可預測性，提高成功部署的機會，並減少重複的手動組態設定。

未建立此最佳實務時的風險暴露等級：中

實作指引

SEC05-BP02 控制網路層內的流量流程和 SEC05-BP03 實作檢測型防護措施中所述的許多網路保護控制措施都附有受管規則系統，可根據最新威脅情報自動更新。 保護 Web 端點的範例包括 AWS WAF 受管規則和 AWS Shield Advanced 自動化應用程式層 DDoS 緩解措施。使用 AWS Network Firewall 受管規則群組隨時掌握有關信譽不良網域清單和威脅特徵的最新資訊。

除了受管規則之外，建議您使用 DevOps 實務來自動部署您的網路資源、防護措施和您指定的規則。 您可以在 AWS CloudFormation 或您選擇的其他基礎設施即程式碼 (IaC) 工具中擷取這些定義，將它們送交至版本控制系統，並使用 CI/CD 管道部署它們。 使用這種方法可受益於透過 DevOps 管理

網路控制措施的固有優勢，例如，更容易預測的發行版本、使用 AWS CloudFormation Guard 等工具的自動化測試，以及偵測部署的環境與所需組態之間的偏離情形。

根據您在 SEC05-BP01 建立網路層的過程中所做的決定，您可能採用集中管理方法來建立傳入、傳出和檢測流程專用的 VPC。 如 AWS Security Reference Architecture (AWS SRA) 中所述，您可以在專用的網路基礎設施帳戶中定義這些 VPC。 您可以使用類似的技術集中定義工作負載在其他帳戶、其安全群組、AWS Network Firewall 部署、Route 53 Resolver 規則和 DNS 防火牆組態及其他網路資源中使用的 VPC。 您可以透過 AWS Resource Access Manager 將這些資源與其他帳戶共用。 使用這種方法可簡化對網路控制措施的自動測試並將該控制措施部署到網路帳戶的程序，同時只需管理一個目的地。 您可以在混合模式中，透過集中部署和共用特定控制措施，並將其他控制措施委派給個別工作負載團隊及其各自的帳戶，來執行上述操作。

實作步驟

1. 建立擁有權來規範要集中定義網路和防護措施的哪些方面，以及您的工作負載團隊可以維護哪些方面。
2. 建立環境來測試變更，並將變更部署至您的網路及其防護措施。 例如，使用網路測試帳戶和網路實際執行帳戶。
3. 決定您要在版本控制系統中儲存和維護範本的方式。 將儲存中央範本的儲存庫與工作負載儲存庫分開，而工作負載範本可以儲存在專屬於該工作負載的儲存庫中。
4. 建立 CI/CD 管道以測試和部署範本。 定義測試來檢查組態錯誤，以及範本是否遵循您的公司標準。

資源

相關的最佳實務：

- SEC01-BP06 自動部署標準安全控制措施

相關文件：

- AWS Security Reference Architecture - 網路帳戶

相關範例：

- AWS 部署管道參考架構
- 透過 NetDevSecOps 將 AWS 聯網部署現代化
- 整合 AWS CloudFormation 安全測試與 AWS Security Hub 和 AWS CodeBuild 報告

相關工具：

- AWS CloudFormation
- AWS CloudFormation Guard
- cfn_nag

SEC 6.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 從強化的映像佈建運算
- SEC06-BP03 減少手動管理和互動式存取
- SEC06-BP04 驗證軟體完整性
- SEC06-BP05 自動化運算保護

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。

預期成果：建立和維護漏洞管理計畫。定期掃描和修補資源，例如 Amazon EC2 執行個體、Amazon Elastic Container Service (Amazon ECS) 容器和 Amazon Elastic Kubernetes Service (Amazon EKS) 工作負載。設定 AWS 受管資源的維護時段，例如 Amazon Relational Database Service (Amazon RDS) 資料庫。使用靜態程式碼掃描來檢查應用程式原始程式碼的常見問題。如果您的組織具有必備技能或是可以雇用外部協助，請考慮 Web 應用程式滲透測試。

常見的反模式：

- 沒有漏洞管理計畫。
- 執行系統修補而不考慮嚴重性或避免風險。
- 使用已過廠商提供的結束生命週期日期的軟體。
- 在分析程式碼的安全問題之前將其部署至生產環境。

建立此最佳實務的優勢：

未建立此最佳實務時的風險暴露等級：高

實作指引

漏洞管理計畫包括安全評定、識別問題、排定優先順序，以及執行修補作業做為解決問題的一部分。持續掃描工作負載，以發現問題和意外網路暴露並執行修正，自動化是關鍵。自動建立和更新資源可節省時間並降低組態錯誤造成進一步問題的風險。設計良好的漏洞管理計畫也應該考慮在軟體生命週期的開發和部署階段進行漏洞測試。在開發和部署期間實作漏洞管理有助於降低漏洞能夠滲入生產環境的可能性。

實作漏洞管理計畫需要對 AWS 共同責任模式有良好的了解，以及它如何與特定工作負載相關。在共同責任模式下，AWS 負責保護 AWS 雲端 的基礎設施。此基礎設施是由硬體、軟體、網路以及執行 AWS 雲端 服務的設施所組成。您負責雲端中的安全，例如實際的資料、安全組態和 Amazon EC2 執行個體的管理工作，以及確認您的 Amazon S3 物件已適當分類和設定。您著手漏洞管理的方法也可能視取用的服務而異。例如，AWS 會管理修補我們受管的關聯式資料庫服務 Amazon RDS，但是您須負責修補自我託管的資料庫。

AWS 擁有可協助您漏洞管理計畫的各種服務。Amazon Inspector 會持續掃描 AWS 工作負載以發現軟體問題和意外的網路存取。AWS Systems Manager Patch Manager 可協助管理 Amazon EC2 執行個體間的修補工作。Amazon Inspector 和 Systems Manager 可供在 AWS Security Hub 中檢視，其是一項雲端安全狀態管理服務，可協助自動化 AWS 安全檢查並集中化安全警示。

Amazon CodeGuru 可協助使用靜態程式碼分析來識別 Java 和 Python 應用程式中的潛在問題。

實作步驟

- 設定 Amazon Inspector：Amazon Inspector 會自動偵測新啟動的 Amazon EC2 執行個體、Lambda 函數和推送到 Amazon ECR 的合格容器映像，並即刻掃描軟體以發現問題、潛在瑕疵和意外的網路暴露。
- 掃描原始碼：掃描程式庫和相依性的問題和瑕疵。Amazon CodeGuru 可以掃描並提供建議來修正 Java 和 Python 應用程式的常見安全問題。OWASP Foundation 發佈了一份原始程式碼分析工具 (也稱為 SAST 工具) 的清單。
- 實作機制以掃描和修補現有環境，並將掃描實作為 CI/CD 管道建置過程的一部分：實作機制來掃描和修補相依性和作業系統中的問題，以協助抵禦新威脅。定期執行該機制。了解您需要在何處套用修補或解決軟體問題，軟體漏洞管理必不可少。透過儘早將漏洞評定嵌入持續整合/持續交付 (CI/CD) 管道，優先修正潛在的安全問題。您的方法可能視您取用的 AWS 服務而異。要檢查在 Amazon EC2 執行個體中執行的軟體的潛在問題，請將 Amazon Inspector 新增到您的管道，在偵測到問題或潛在瑕疵時通知您並停止建置程序。Amazon Inspector 會持續監控資源。您也可以使用開放原始碼產

品，例如 OWASP Dependency-Check、Snyk、OpenVAS、封裝管理員和 AWS Partner 工具來進行漏洞管理。

- 使用 AWS Systems Manager：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) 和其他運算資源。AWS Systems Manager Patch Manager 可自動化透過安全相關及其他更新來修補受管執行個體的流程。您可以使用 Patch Manager 在 Amazon EC2 執行個體上針對作業系統和應用程式套用修補程式，包括 Microsoft 應用程式、Windows Service Pack 和 Linux 型執行個體的次要版本更新。除了 Amazon EC2 之外，Patch Manager 也可以用來修補內部部署伺服器。

  如需支援的作業系統清單，請參閱《Systems Manager 使用者指南》中的受支援作業系統。您可以掃描執行個體而只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。
- 使用 AWS Security Hub：Security Hub 為您在 AWS 中的安全狀態提供全方位檢視。它會收集多個 AWS 服務間的安全資料，並以標準格式提供該些發現結果，讓您能夠跨 AWS 服務排定安全發現結果的優先順序。
- 使用 AWS CloudFormation：AWS CloudFormation 是基礎設施即程式碼 (IaC) 服務，可透過在多個帳戶和環境間自動化資源部署和標準化資源架構，來協助漏洞管理。

資源

相關文件：

- AWS Systems Manager
- AWS Lambda 的安全概觀
- Amazon CodeGuru
- 透過全新的 Amazon Inspector 改進、自動化雲端工作負載的漏洞管理
- 使用 Amazon Inspector 和 AWS Systems Manager 自動化 AWS 中的漏洞管理和矯正 – 第 1 部分

相關影片：

- 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

SEC06-BP02 從強化的映像佈建運算

透過從強化的映像部署，就可減少意外存取執行時期環境的機會。僅從受信任的登錄檔取得執行時期相依項 (例如容器映像和應用程式庫)，並驗證其簽章。建立自己的私有登錄檔來儲存受信任的映像和程式庫，以供您的建置和部署程序使用。

預期成果：您的運算資源是從強化的基準映像佈建。您只會從受信任的登錄檔擷取外部相依項 (例如容器映像和應用程式庫)，並驗證其簽章。這些都會儲存在私有登錄檔中，以供您的建置和部署程序參考。您會定期掃描和更新映像與相依項，以協助防禦任何新發現的漏洞。

常見的反模式：

- 從受信任的登錄檔取得映像和程式庫，但未先驗證其簽章或執行漏洞掃描，即逕行使用。
- 強化映像，但未定期測試映像以確認是否有新的漏洞或更新到最新版本。
- 安裝或未移除在預期的映像生命週期內不需要的軟體套件。
- 僅依賴修補來讓實際執行運算資源保持最新狀態。單單是修補就仍有可能導致運算資源在經過一段時間後，偏離強化的標準。修補也可能無法移除威脅行為者在安全事件期間安裝的惡意軟體。

建立此最佳實務的優勢：強化映像有助於減少您的執行時期環境中可能成為未經授權使用者或服務意外存取路徑的數目。此外還能在發生任何意外存取的情況時，縮小影響的範圍。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要強化您的系統，請從作業系統、容器映像和應用程式庫的最新版本開始。套用已知問題的修補程式。移除任何不需要的應用程式、服務、裝置驅動程式、預設使用者和其他憑證，藉此盡可能縮減系統規模。採取任何其他必要的行動，例如，停用連接埠以建立只有工作負載所需資源和功能的環境。以此為基準，您就可以安裝用於監控工作負載或管理漏洞等操作所需的軟體、代理程式或其他程序。

您可以使用如 Center for Internet Security (CIS) 和 Defense Information Systems Agency (DISA) 安全技術實作指南 (STIG) 等受信任來源提供的指引來減輕強化系統的負擔。建議您從 AWS 或 APN 合作夥伴發佈的 Amazon Machine Image (AMI) 開始著手，然後以適當的 CIS 和 STIG 控制措施組合為依據，使用 AWS EC2 Image Builder 自動化組態設定。

雖然有許多採用 CIS 或 DISA STIG 建議的強化映像和 EC2 Image Builder 配方可用，但您可能會發現其組態讓您的軟體無法成功執行。在這種情況下，您可以從非強化的基底映像開始著手，安裝您的軟體，然後逐步實施 CIS 控制措施來測試其影響。對於任何讓您的軟體無法順利執行的 CIS 控制措施，請改為測試您是否可在 DISA 中實施更精細的強化建議。持續追蹤您能夠成功實施的不同 CIS 控制措施和 DISA STIG 組態。這些可讓您在 EC2 Image Builder 中用來定義對應的映像強化配方。

對於容器化工作負載，Amazon Elastic Container Registry (ECR) 公有儲存庫上提供了來自 Docker 的強化映像。您可以使用 EC2 Image Builder 搭配 AMI 來強化容器映像。

就如同作業系統和容器映像，您可以透過 pip、npm、Maven 和 NuGet 等工具，從公有儲存庫取得程式碼套件 (或程式庫)。我們建議您藉由整合私有儲存庫 (例如在 AWS CodeArtifact 內) 與受信任的公有儲存庫來管理程式碼套件。此整合可為您處理擷取和儲存套件，以及將套件保持在最新狀態。如此一來，您的應用程式建置程序就能使用像是軟體組成分析 (SCA)、靜態應用程式安全測試 (SAST) 和動態應用程式安全測試 (DAST) 等技術，來取得並測試這些套件的最新版本以及應用程式。

對於使用 AWS Lambda 的無伺服器工作負載，可使用 Lambda 層簡化管理套件相依項的工作。 使用 Lambda 層設定一組跨不同函數共用的標準相依項，並放入獨立的封存中。您可以透過這些層本身的建置程序建立和維護它們，藉由集中處理的方式讓函數保持最新狀態。

實作步驟

- 強化作業系統。將來自受信任來源的基底映像作為建置強化 AMI 的基礎。使用 EC2 Image Builder 協助您自訂映像上安裝的軟體。
- 強化容器化資源。設定容器化資源以符合安全最佳實務。使用容器時，在您的建置管道中定期對照映像儲存庫執行 ECR 映像掃描，以在容器中尋找 CVE。
- 搭配 AWS Lambda 使用無伺服器實作時，請使用 Lambda 層來隔離應用程式函數程式碼和共用的相依程式庫。為 Lambda 設定程式碼簽署，確保只有受信任的程式碼能夠在您的 Lambda 函數中執行。

資源

相關的最佳實務：

- OPS05-BP05 執行修補程式管理

相關影片：

- 深入探索 AWS Lambda 安全

相關範例：

- 使用 EC2 Image Builder 快速建置符合 STIG 規範的 AMI
- 建置更好的容器映像
- 使用 Lambda 層簡化開發流程

- 使用無伺服器架構開發和部署 AWS Lambda 層
- 使用開放原始碼 SCA、SAST 和 DAST 工具建置端對端 AWS DevSecOps CI/CD 管道

SEC06-BP03 減少手動管理和互動式存取

盡可能使用自動化方式來執行部署、組態、維護和調查任務。在發生緊急程序的情況下或在安全 (沙盒) 環境中無法啟用自動化時，請考慮手動存取運算資源。

預期成果：程式化的指令碼和自動化文件 (執行手冊) 會擷取運算資源上獲得授權的動作。這些執行手冊會自動啟動、透過變更偵測系統啟動，或是在需要人為判斷時手動啟動。只有在無法啟用自動化的緊急情況下，才能直接存取運算資源。所有手動活動都會加以記錄並納入審查程序中，以持續改善您的自動化功能。

常見的反模式：

- 透過 SSH 或 RDP 等通訊協定互動式存取 Amazon EC2 執行個體。
- 維護個別使用者登入，例如 `/etc/passwd` 或 Windows 本機使用者。
- 在多個使用者之間共用密碼或私有金鑰以存取執行個體。
- 手動安裝軟體和建立或更新組態檔案。
- 手動更新或修補軟體。
- 登入執行個體以解決問題。

建立此最佳實務的優勢：透過自動化方式執行步驟，有助於降低意外變更和組態錯誤伴隨的操作風險。不再使用 Secure Shell (SSH) 和遠端桌面通訊協定 (RDP) 進行互動式存取，因此縮小了運算資源的存取範圍。這樣也消除了常見的未經授權動作路徑。在自動化文件和程式化指令碼中寫入運算資源管理任務，提供了以更精細的細節程度定義和稽核完整的授權活動範圍的機制。

未建立此最佳實務時的風險暴露等級：中

實作指引

登入執行個體是系統管理的傳統方法。安裝伺服器作業系統後，使用者通常會手動登入以設定系統並安裝所需的軟體。在伺服器的生命週期內，使用者可能會登入以執行軟體更新、套用修補程式、變更組態及排解疑難。

但是，手動存取伴隨著許多風險。它需要伺服器監聽請求，例如 SSH 或 RDP 服務，而這些服務可能成為未經授權的存取路徑。此外，它也會增加執行手動步驟時發生人為錯誤的風險。這些都可能導致工

作負載事故、資料損壞或銷毀，或其他安全問題。人為存取也需要設置防護措施來防止憑證共用行為，因而產生額外的管理負擔。

為了降低這些風險，您可以實作代理程式型的遠端存取解決方案，例如 AWS Systems Manager。AWS Systems Manager Agent (SSM Agent) 會啟動加密通道，因此不需依賴偵聽外部發出的請求。請考慮設定 SSM Agent 以透過 VPC 端點建立此通道。

Systems Manager 可讓您精細控制與受管執行個體互動的方式。您可以定義要執行的自動化程序、誰可以執行它們，以及何時可以執行。Systems Manager 不需互動式存取執行個體，即可套用修補程式、安裝軟體及進行組態變更。Systems Manager 還可提供對遠端 Shell 的存取權，並將工作階段期間調用的每個命令及其輸出記錄到日誌和 Amazon S3。AWS CloudTrail 會記錄 Systems Manager API 的調用以供檢測。

實作步驟

1. 在 Amazon EC2 執行個體上安裝 AWS Systems Manager Agent (SSM Agent)。查看是否包含 SSM Agent，且它會作為基底 AMI 組態的一部分自動啟動。
2. 確認與您的 EC2 執行個體設定檔相關聯的 IAM 角色是否包含 `AmazonSSMManagedInstanceCore` 受管 IAM 政策。
3. 停用執行個體上執行的 SSH、RDP 和其他遠端存取服務。您可以藉由執行啟動範本的使用者資料區段中設定的指令碼，或使用 EC2 Image Builder 等工具建置自訂 AMI，以執行此操作。
4. 確認適用於 EC2 執行個體的安全群組輸入規則不允許在連接埠 22/tcp (SSH) 或連接埠 3389/tcp (RDP) 上的存取。使用 AWS Config 等服務實作偵測，並對設定錯誤的安全群組發出警示。
5. 定義適當的自動化、執行手冊，並在 Systems Manager 中執行命令。使用 IAM 政策定義誰可以執行這些動作，以及允許執行這些動作的條件。在非實際執行環境中完整測試這些自動化程序。在必要時調用這些自動化程序，而非以互動方式存取執行個體。
6. 在必要時，使用 AWS Systems Manager Session Manager 提供對執行個體的互動式存取。開啟工作階段活動記錄，以在 Amazon CloudWatch Logs 或 Amazon S3 中維護稽核記錄。

資源

相關的最佳實務：

- REL08-BP04 使用不可變基礎設施進行部署

相關範例：

- 使用 AWS Systems Manager 取代 SSH 存取以減輕管理和安全負擔

相關工具：

- AWS Systems Manager

相關影片：

- 在 AWS Systems Manager Session Manager 中控制使用者工作階段對執行個體的存取權

SEC06-BP04 驗證軟體完整性

使用加密驗證來驗證工作負載所使用之軟體成品 (包括映像) 的完整性。 以加密方式簽署您的軟體，以防範未經授權的變更在您的運算環境內執行。

預期成果：所有成品都是從受信任的來源取得。廠商網站憑證經過驗證。 下載的成品已藉由其簽章以加密方式驗證。自有軟體會由您的運算環境以加密方式簽署和驗證。

常見的反模式：

- 信任信譽良好的廠商網站來取得軟體成品，但忽略憑證到期通知。 未先確認憑證是否有效，即逕行下載。
- 驗證廠商網站憑證，但未以加密方式驗證從這些網站下載的成品。
- 僅依賴摘要或雜湊值來驗證軟體完整性。 雜湊值確定成品的原版未經修改，但未驗證其來源。
- 即使僅在您自己的部署中使用，也未簽署自有軟體、程式碼或程式庫。

建立此最佳實務的優勢：驗證您的工作負載所依賴之成品的完整性，有助於防止惡意軟體進入您的運算環境。 簽署您的軟體有助於防範運算環境中發生未經授權執行的情況。 藉由簽署和驗證程式碼來保護您的軟體供應鏈。

未建立此最佳實務時的風險暴露等級：中

實作指引

作業系統映像、容器映像和程式碼成品通常在散佈時會提供完整性檢查，例如透過摘要或雜湊值。 這些檢查可讓用戶端運算自有的承載雜湊值並確認其與發佈的雜湊值相同，藉此驗證完整性。 雖然這些檢查有助於驗證承載未遭到竄改，但不會驗證承載來自原始出處 (其來源)。 驗證來源需要使用受信任的授權機構發出的憑證來數位簽署成品。

如果您在工作負載中使用下載的軟體或成品，請確認提供者是否提供了用於驗證數位簽章的公有金鑰。 以下範例說明 AWS 如何提供公有金鑰，以及如何驗證我們發佈的軟體：

- EC2 Image Builder：驗證 AWSTOE 安裝下載的簽章
- AWS Systems Manager：驗證 SSM Agent 的簽章
- Amazon CloudWatch：驗證 CloudWatch 代理程式套件的簽章

請將數位簽章驗證納入您用來取得和強化映像的程序中，如 SEC06-BP02 從強化的映像佈建運算中所述。

您可以使用 AWS Signer 協助您管理簽章驗證，以及您自有軟體和成品的程式碼簽署生命週期。 AWS Lambda 和 Amazon Elastic Container Registry 兩者都提供與 Signer 的整合，可用來驗證程式碼和映像的簽章。 您可以使用「資源」區段中的範例，將 Signer 納入您的持續整合與持續交付 (CI/CD) 管道中，以便自動驗證簽章及自動簽署自有程式碼和映像。

資源

相關文件：

- 容器的加密簽署
- 使用 AWS Signer 協助保護容器映像建置管道的最佳實務
- 宣布使用 AWS Signer 和 Amazon EKS 簽署容器映像
- 設定 AWS Lambda 的程式碼簽署
- Lambda 程式碼簽署的最佳實務和進階模式
- 使用 AWS Certificate Manager Private CA 和 AWS Key Management Service 非對稱金鑰的程式碼簽署

相關範例：

- 使用 Amazon CodeCatalyst 和 AWS Signer 自動化 Lambda 程式碼簽署
- 使用 AWS Signer 簽署和驗證 OCI 成品

相關工具：

- AWS Lambda
- AWS Signer
- AWS Certificate Manager
- AWS Key Management Service

- AWS CodeArtifact

SEC06-BP05 自動化運算保護

自動化運算保護操作以減少人工介入的需求。使用自動化掃描偵測運算資源內的潛在問題，並透過自動化的程式化回應或機群管理操作進行修復。 將自動化納入 CI/CD 程序，以部署具有最新相依項且值得信賴的工作負載。

預期成果：自動化系統會執行運算資源的所有掃描和修補工作。您會使用自動化的驗證方式確認軟體映像和相依項來自受信任的來源，且未遭到竄改。 透過自動化方式檢查工作負載是否有最新的相依項，並且簽署工作負載以在 AWS 運算環境中建立可靠性。 偵測到不合規資源時，系統會啟動自動補救措施。

常見的反模式：

- 遵循不可變的基礎設施實務，但未備妥解決方案來因應緊急修補或替換實際執行系統。
- 使用自動化方式修復設定錯誤的資源，但未設置手動覆寫機制。 可能會發生需要調整需求的情況，且您可能需要暫停自動化程序，直到完成這些變更為止。

建立此最佳實務的優勢：自動化可降低未經授權存取和使用您的運算資源的風險。 此方式有助於防止錯誤的組態進入實際執行環境，並且在發生組態錯誤時偵測到該錯誤並加以修復。 自動化還可協助偵測未經授權存取和使用運算資源的情況，進而縮短您回應的時間。 如此還能進一步縮小問題的整體影響範圍。

未建立此最佳實務時的風險暴露等級：中

實作指引

您可以套用「安全支柱」實務中所述的自動化方式，以保護您的運算資源。SEC06-BP01 執行漏洞管理說明如何在您的 CI/CD 管道中使用 Amazon Inspector，以及如何運用它持續掃描您的執行時期環境，以找出已知的通用漏洞披露 (CVE) 項目。 您可以透過自動化執行手冊，使用 AWS Systems Manager 套用修補程式或從全新映像重新部署，讓您的運算機群隨時擁有最新的軟體和程式庫。 使用這些技術可減少對手動程序和互動式存取運算資源的需求。 請參閱 SEC06-BP03 減少手動管理和互動式存取了解更多資訊。

自動化在部署值得信賴的工作負載方面，也發揮了舉足輕重的作用，如 SEC06-BP02 從強化的映像佈建運算和 SEC06-BP04 驗證軟體完整性中所述。 您可以使用 EC2 Image Builder、AWS Signer、AWS CodeArtifact 和 Amazon Elastic Container Registry (ECR) 等服務來下載、驗證、建構

和儲存強化且經核准的映像和程式碼相依項。 除了 Inspector 之外，這些都可在 CI/CD 程序中發揮作用，因此，只有在確認工作負載的相依項為最新狀態且來自受信任的來源時，工作負載才會進入實際執行環境。 您的工作負載也會經過簽署，如此一來，像是 AWS Lambda 和 Amazon Elastic Kubernetes Service (EKS) 等 AWS 運算環境就能在確認其未遭到竄改後，再允許其執行。

除了這些預防性控制措施之外，您還可以在偵測控制措施中針對運算資源使用自動化。 舉例來說，AWS Security Hub 提供 NIST 800-53 Rev. 5 標準，其中包括如下述的檢查：[EC2.8] EC2 執行個體應使用執行個體中繼資料服務第 2 版 (IMDSv2)。 IMDSv2 會使用工作階段驗證技術，封鎖包含 X-Forwarded-For HTTP 標頭以及網路 TTL 1 的請求，藉此阻止來自外部來源的流量擷取有關 EC2 執行個體的資訊。Security Hub 中的這項檢查可偵測 EC2 執行個體何時使用 IMDSv1，並實施自動補救措施。請參閱 SEC04-BP04 針對不合規資源實施補救措施，進一步了解自動化偵測和補救措施。

實作步驟

1. 使用 EC2 Image Builder 自動建立安全、合規且強化的 AMI。 您可以產生映像，並於當中納入來自 Center for Internet Security (CIS) Benchmarks 的控制措施，或來自基底 AWS 和 APN 合作夥伴映像的安全技術實作指南 (STIG) 標準。
2. 自動化組態管理藉由使用組態管理服務或工具，在您的運算資源中自動強制執行和驗證安全組態。
    a. 使用 AWS Config 自動化組態管理
    b. 使用 AWS Security Hub 自動化安全與合規狀態管理
3. 自動修補或取代 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。AWSSystems Manager Patch Manager 可自動執行透過安全相關及其他更新來修補受管執行個體的流程。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。
    a. AWS Systems Manager Patch Manager
4. 自動掃描運算資源以找出通用漏洞披露 (CVE) 項目，並將安全掃描解決方案內嵌於您的組建管道中。
    a. Amazon Inspector
    b. ECR 映像掃描
5. 考慮使用 Amazon GuardDuty 執行自動化惡意軟體和威脅偵測，以保護運算資源。GuardDuty 還可在 AWS Lambda 函數於您的 AWS 環境中調用時，識別潛在問題。
    a. Amazon GuardDuty
6. 考慮 AWS 合作夥伴解決方案。AWS 合作夥伴提供多種業界領先的產品，這些產品與您內部部署環境中的現有控制措施相當、相同或互相整合。這些產品可與現有的 AWS 服務相輔相成，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。
    a. 基礎設施安全

資源

相關的最佳實務：

- SEC01-BP06 自動部署標準安全控制措施

相關文件：

- 在您的 AWS 基礎設施內充分利用 IMDSv2 的優勢並停用 IMDSv1

相關影片：

- Amazon EC2 執行個體中繼資料服務的安全最佳實務

資料保護

問題

- SEC 7.如何分類資料？
- SEC 8.如何保護靜態資料？
- SEC 9.如何保護傳輸中資料？

SEC 7.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 了解您的資料分類機制
- SEC07-BP02 根據資料敏感性實施資料保護控制措施
- SEC07-BP03 自動識別和分類
- SEC07-BP04 定義可擴展的資料生命週期管理

SEC07-BP01 了解您的資料分類機制

了解您的工作負載要處理的資料分類、其處理需求、相關聯的業務流程、資料儲存在何處，以及誰是資料負責人。 您的資料分類和處理機制應考慮工作負載的適用法律和合規要求，以及需要何種資料控制措施。了解資料是資料分類之旅的第一步。

預期成果：您的工作負載中存在的資料類型已得到充分了解並加以記錄。 設置了適當的控制措施，可根據資料分類來保護敏感資料。 這些控制措施左右著下列考量：允許誰存取資料及存取的目的為何、資料儲存在何處、該資料的加密政策及如何管理加密金鑰、資料的生命週期及其保留要求、適當的銷毀程序、設置了哪些備份和復原程序，以及存取權稽核。

常見的反模式：

- 未制定正式的資料分類政策來定義資料敏感程度及其處理需求
- 未充分了解工作負載內資料的敏感程度，也未在架構和營運文件中擷取這些資訊
- 未能根據您的資料分類和處理政策中規範的資料敏感程度和需求，對資料實施適當的控制措施
- 未能向政策負責人提供有關資料分類和處理需求的意見回饋。

建立此最佳實務的優勢：此實務能消除有關適當處理工作負載內資料的不確定性。 實施正式政策來定義組織中資料的敏感程度及其所需防護措施，有助於符合法律規範和其他網路安全鑑定與認證。 工作負載負責人清楚知道敏感資料儲存在何處以及設置了哪些保護控制措施，因而能夠放心。 將這些資訊納入文件中，可協助新的團隊成員更充分了解並在任職期間及早採取控制措施。這些實務還可針對每一種資料類型實施適當的控制措施，進而有助於降低成本。

未建立此最佳實務時的風險暴露等級：高

實作指引

在設計工作負載時，您可能會考量採取直接了當的方式保護敏感資料。 例如，在多租用戶應用程式中，直接將每一個租用戶的資料視為敏感資料並採取防護措施，讓租用戶無法存取其他租用戶的資料。 同樣地，您可能會直接設計存取控制，只讓管理員修改資料，而其他使用者只擁有讀取層級存取權，或完全無存取權。

藉由在政策中定義並擷取這些資料敏感程度，以及其資料保護需求，您就能正式確定哪些資料要放在您的工作負載中。接著您就可確定是否設置了正確的控制措施、是否可稽核控制措施，以及在發現資料遭不當處理的情況時，要採取何種適當的回應。

為協助分類敏感資料在工作負載內的位置，請考慮使用資源標籤 (如可用)。 例如，您可以對受保護醫療資訊 (PHI) 套用標籤索引鍵為##且標籤值為 PHI 的標籤，以及另一個標籤索引鍵為###且標籤值為#的標籤。 接著您可以使用 AWS Config 這類服務來監控這些資源是否發生變更，並且在發生修改後導致資源不符合保護需求 (例如變更加密設定) 的情況時發出警示。 您可以使用標籤政策 (此為 AWS Organizations 的功能) 擷取標籤索引鍵的標準定義和可接受的值。不建議在標籤索引鍵或值中包含私人或敏感資料。

實作步驟

1. 了解組織的資料分類機制和保護需求。
2. 確定工作負載處理的敏感資料類型。
3. 確認敏感資料根據您的政策儲存在工作負載內並受到保護。 使用自動化測試等技術來稽核控制措施的有效性。
4. 考慮使用資源和資料層級標記 (如可用) 來標記資料的敏感程度和其他操作中繼資料，以協助監控和回應事件。
   a. AWS Organizations 標籤政策可用來強制實施標記標準。

資源

相關的最佳實務：

- SUS04-BP01 實作資料分類政策

相關文件：

- 資料分類白皮書
- 標記 AWS 資源的最佳實務

相關範例：

- AWS Organizations 標籤政策語法和範例

相關工具：

- AWS Tag Editor

SEC07-BP02 根據資料敏感性實施資料保護控制措施

實施資料保護控制措施，為您分類政策中定義的每一個資料類別提供適當的控制層級。 這種做法可讓您保護敏感資料防止遭到未經授權的存取和使用，同時讓資料保持可用且實用。

預期成果：您設置了分類政策，在組織中定義不同程度的資料敏感性。 您針對每一種敏感程度發佈了清楚的指導方針，以界定核准的儲存和處理服務與位置，以及其所需的組態。 您根據所需的保護層級及其相關成本，針對每一種敏感程度實施控制措施。 您設置了監控和警示，以偵測資料是否出現在未

經授權的位置、在未經授權的環境中經過處理、遭到未經授權的人員存取，或是相關服務的組態是否變得不合規。

常見的反模式：

- 對所有資料實施相同層級的保護控制措施。這樣可能會導致對低敏感性資料過度佈建安全控制措施，或對高敏感性資料的保護不足。
- 在定義資料保護控制措施時，未邀集安全、合規和業務團隊的利害關係人參與此過程。
- 忽略實施和維護資料保護控制措施伴隨的營運支出和成本。
- 未定期審查資料保護控制措施，而未能持續遵循分類政策。

建立此最佳實務的優勢：藉由依照資料分類層級實施您的控制措施，您的組織就能在需要時投入更高層級的控制。這可能包括增加保障安全、監控、衡量、修復和報告方面的資源。 在適度採行較少控制措施的情況下，您就可以改善員工、客戶或成員使用的資料存取性和完整性。 這種方法為您的組織帶來了最大的資料使用彈性，同時遵守資料保護要求。

未建立此最佳實務時的風險暴露等級：高

實作指引

根據資料敏感程度實施資料保護控制措施的方式包含幾個重要的步驟。首先，確定工作負載架構內不同的資料敏感程度 (例如公開、內部、機密和受限)，並評估您儲存和處理這些資料的位置。接著根據資料敏感程度定義其隔離界限。建議您使用服務控制政策 (SCP) 將資料分隔到不同的 AWS 帳戶 中，以限制每一種資料敏感程度允許的服務和動作。這樣一來，您就可以建立強大的隔離界限，並強制執行最低權限原則。

定義隔離界限之後，根據資料敏感程度實施適當的保護控制措施。請參閱保護靜態資料和保護傳輸中的資料的最佳實務，以實施如加密、存取控制和稽核等相關控制措施。考慮採用如記號化或匿名化等技術來降低資料的敏感程度。採用集中式系統進行記號化和去記號化，以簡化對整個企業套用一致的資料政策的程序。

持續監控和測試所實施控制措施的有效性。隨著組織的資料態勢和威脅發展，定期審查和更新資料分類機制、風險評估和保護控制措施。實施的資料保護控制措施務必遵循相關產業法規、標準和法律要求。此外，提供安全意識和培訓，幫助員工了解資料分類機制及他們在處理和保護敏感資料方面的責任。

實作步驟

1. 確定工作負載內資料的分類和敏感程度。

2. 為每一種敏感程度定義隔離界限，並確定執行策略。
3. 評估您定義的控制措施是否確實有效控管您的資料分類政策規定的存取、加密、稽核、保留和其他方面。
4. 評估能適時降低資料敏感程度的選項，例如使用記號化或匿名化。
5. 使用自動測試和監控所設定資源的方式來驗證您的控制措施。

資源

相關的最佳實務：

- PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區
- COST04-BP05 強制執行資料保留政策

相關文件：

- 資料分類白皮書
- 安全、識別及合規最佳實務
- AWS KMS 最佳實務
- AWS 服務的加密最佳實務和功能

相關範例：

- 建置無伺服器記號化解決方案為敏感資料提供遮罩
- 如何使用記號化的方式來提高資料安全並縮小稽核範圍

相關工具：

- AWS Key Management Service (AWS KMS)
- AWS CloudHSM
- AWS Organizations

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。使用自動化增強手動判斷，可降低人為錯誤和暴露的風險。

預期成果：您可根據您的分類和處理政策來確認是否有適當的控制措施。自動化工具和服務可協助您識別和分類資料的敏感程度。 自動化還可協助您持續監控環境，以偵測並警示未經授權的資料儲存或處理行為，以便快速採取矯正行動。

常見的反模式：

- 僅依賴手動程序來識別和分類資料，這個過程可能容易出錯且相當耗時。 這樣做可能導致資料分類效率不彰且不一致，尤其隨著資料量增加會每況愈下。
- 未設置追蹤和管理整個組織中資料資產的機制。
- 即使資料在組織內移動和發展，組織仍然忽略持續監控和分類資料的需要。

建立此最佳實務的優勢：採取自動識別和分類資料的方式，能夠更一致且準確地實施資料保護控制措施，進而降低人為錯誤的風險。 自動化還可讓您深入洞悉敏感資料存取和移動的情形，進而協助您偵測未經授權的處理，並採取矯正行動。

未建立此最佳實務時的風險暴露等級：中

實作指引

在工作負載的初始設計階段常會採用人為判斷來分類資料，盡管如此，仍請考慮設置系統來自動識別和分類測試資料，以此作為預防性控制措施。例如，您可提供工具或服務讓開發人員用來掃描代表性的資料，以確定其敏感性。 在 AWS 內，您可以將資料集上傳到 Amazon S3，並使用 Amazon Macie、Amazon Comprehend 或 Amazon Comprehend Medical 掃描這些資料集。 同樣地，請考慮在單元和整合測試的過程中掃描資料，以偵測不該出現敏感資料的位置。在此階段發出有關敏感資料的警示，就能在部署到實際執行環境之前，讓防護措施的落差浮現。其他像是 AWS Glue、Amazon SNS 和 Amazon CloudWatch 中的敏感資料偵測等功能，也可用來偵測 PII 和採取緩解措施。對於任何自動化工具或服務，務必了解其如何定義敏感資料，並利用其他人為或自動化解決方案加強它，以視需要消除任何落差。

持續監控您的環境，以其作為偵測控制措施，藉以偵測敏感資料是否以不合規的方式儲存。 這樣做有助於偵測出在未適當去識別化或修訂的情況下，將敏感資料發送到日誌檔或複製到資料分析環境中的情形。 可使用 Amazon Macie 持續監控儲存在 Amazon S3 中的資料，以偵測其中是否存在敏感資料。

實作步驟

1. 對您的環境執行初步掃描，以進行自動識別和分類。
    a. 初步完整掃描資料有助於全面了解敏感資料在您環境中的位置。若一開始不需要或因成本考量而無法事先完成完整掃描，請評估資料取樣技術是否適合用來實現您的成果。例如，您可設定 Amazon Macie 跨 S3 儲存貯體執行廣泛的自動化敏感資料探索操作。 此功能使用的取樣技術會

以符合成本效益的方式初步分析敏感資料的所在位置。 後續可使用敏感資料探索工作來深入分析 S3 儲存貯體。您也可以將其他資料存放區匯出到 S3，以便讓 Macie 進行掃描。

2. 設定環境的持續掃描。
   a. Macie 的自動化敏感資料探索功能可用來持續掃描您的環境。 若有任何經授權儲存敏感資料的已知 S3 儲存貯體，則可使用 Macie 中的允許清單將其排除在外。
3. 將識別和分類納入您的建置和測試程序中。
   a. 識別開發人員可在工作負載開發過程中用來掃描資料以判斷敏感性的工具。 在整合測試的過程中使用這些工具，以便在敏感資料意外出現時發出警示，並防止進一步部署。
4. 在未經授權的位置發現敏感資料時，實作系統或執行手冊來採取行動。

資源

相關文件：

- AWS Glue：偵測和處理敏感資料
- 在 Amazon SNS 中使用受管資料識別符
- Amazon CloudWatch Logs：使用遮罩協助保護敏感日誌資料

相關範例：

- 使用 Macie 為 Amazon RDS 資料庫啟用資料分類
- 使用 Macie 偵測 DynamoDB 中的敏感資料

相關工具：

- Amazon Macie
- Amazon Comprehend
- Amazon Comprehend Medical
- AWS Glue

SEC07-BP04 定義可擴展的資料生命週期管理

了解您的資料生命週期需求，因為這些需求與您不同層級的資料分類和處理相關。 這可能包括資料一開始進入環境時的處理方式、資料轉換的方式，以及銷毀資料的規則。 請將保留期、存取、稽核和追蹤來源等因素納入考量。

預期成果：您的資料分類會盡可能接近擷取點和時間。當資料分類需要遮罩、記號化或其他降低敏感程度的處理時，您會在盡可能最接近擷取點和時間的條件下執行這些動作。

當資料不再適合保存時，您會遵循政策根據資料的分類將其刪除。

常見的反模式：

- 實作一體適用的方法來管理資料生命週期，而未考量不同的敏感程度和存取需求。
- 僅從資料為可用資料或備份資料的角度來考量生命週期管理，而非兩者均考量。
- 假設已輸入工作負載的資料有效，但未確定其價值或來源。
- 依賴資料耐久性來替代資料備份和保護。
- 保留資料的時間超過其實用性和所需的保留期。

建立此最佳實務的優勢：定義明確且可擴展的資料生命週期管理策略有助於保持合規、提高資料安全性、最佳化儲存成本，以及在維持適當控制之下實現有效率的資料存取和共用。

未建立此最佳實務時的風險暴露等級：高

實作指引

工作負載內的資料通常是動態的。 資料進入工作負載環境時採取的形式，可能與資料儲存或使用在商業邏輯、報告、分析或機器學習上的形式有所不同。 此外，資料的價值可能隨時間而改變。有些資料本質上是暫時性的，會隨著時間失去其價值。 請考量在您的資料分類機制與相關控制措施下，這些資料變更對評估的影響。 可能的話，盡量使用自動化生命週期機制 (如 Amazon S3 生命週期政策和 Amazon Data Lifecycle Manager) 來設定資料保留、封存和到期程序。

區分可供使用的資料與儲存為備份的資料。 考慮使用 AWS Backup 自動備份 AWS 服務中的資料。Amazon EBS 快照提供了複製 EBS 磁碟區並使用 S3 功能來儲存它的方式，包括生命週期、資料保護和存取保護機制。其中兩種機制為 S3 Object Lock 和 AWS Backup Vault Lock，皆可提高您備份的安全性，並且讓您更有效地掌控備份。進行分明的職責和備份存取權劃分管理。在帳戶層級隔離備份，以便在事件發生期間與受影響的環境保持分離。

生命週期管理的另一方面，是記錄資料在工作負載中進度的歷史記錄，稱為資料來源追蹤。如此您就能確信自己知道資料來自何處、執行的任何轉換、哪些擁有者或處理程序做出這些變更，以及時間點。 這份歷史記錄有助於在可能發生安全事件的期間進行問題的疑難排解和調查。 例如，您可以在 Amazon DynamoDB 表中記錄有關轉換的中繼資料。 在資料湖內，您可以針對每一個資料管道階段，將轉換後資料的副本保留在不同的 S3 儲存貯體中。將結構描述和時間戳記資訊儲存在 AWS Glue Data Catalog 中。 無論您採用何種解決方案，請務必考量最終使用者的需求，以確定報告資料來源所需的適當工具。 這樣做將幫助您確定追蹤來源的最佳方式。

實作步驟

1. 分析工作負載的資料類型、敏感程度及存取需求，以分類資料並定義適當的生命週期管理策略。
2. 設計並實施符合法律、法規和組織要求的資料保留政策及自動銷毀程序。
3. 建立流程和自動化功能，以隨著工作負載需求和法規發展，持續監控、稽核和調整資料生命週期管理策略、控制措施及政策。

資源

相關的最佳實務：

- COST04-BP05 強制執行資料保留政策
- SUS04-BP03 使用政策來管理資料集的生命週期

相關文件：

- 資料分類白皮書
- AWS 勒索軟體防禦藍圖
- DevOps 指引：利用資料來源追蹤改善可追溯性

相關範例：

- 如何在 AWS 中於整個生命週期保護敏感資料
- 使用 AWS Glue、Amazon Neptune 和 Spline 為資料湖建置資料歷程

相關工具：

- AWS Backup
- Amazon Data Lifecycle Manager
- AWS Identity and Access Management Access Analyzer

SEC 8.如何保護靜態資料？

實作多項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 自動化靜態資料保護
- SEC08-BP04 強制存取控制

SEC08-BP01 實作安全金鑰管理

安全金鑰管理包括儲存、輪替、存取控制及監控保護工作負載的靜態資料所需的金鑰資料。

預期成果： 可擴展、可重複且自動化的金鑰管理機制。此機制應提供對金鑰資料強制執行最低權限存取的能力，並且在金鑰可用性、機密性和完整性之間提供正確的平衡。金鑰存取權應受到監控，而金鑰資料應透過自動化程序輪替。金鑰資料絕不可供真人身分存取。

常見的反模式：

- 真人存取未加密的金鑰資料。
- 建立自訂的加密演算法。
- 存取金鑰資料的許可過於廣泛。

建立此最佳實務的優勢： 透過為工作負載建立安全的金鑰管理機制，就可以協助保護您的內容，防止未經授權的存取。此外，您可能需要依法加密您的資料。有效的金鑰管理解決方案能夠提供符合這些法規的技術機制，以保護金鑰資料。

未建立此最佳實務時的曝險等級： 高

實作指引

許多法規需求和最佳實務都納入了靜態資料加密做為基本的安全控制。為了符合此控制，您的工作負載須採取某種機制，以安全存放和管理用於加密靜態資料的金鑰資料。

AWS 提供 AWS Key Management Service (AWS KMS) 來為 AWS KMS 金鑰提供耐用、安全和冗餘的儲存。 許多 AWS 服務會與 AWS KMS 整合， 以支援資料加密。AWS KMS 使用 FIPS 140-2 3 級驗證的硬體安全模組來保護您的金鑰。沒有任何機制可將 AWS KMS 金鑰匯出為純文字。

使用多帳戶策略部署工作負載時，會採取的 最佳實務 是將 AWS KMS 金鑰與使用金鑰的工作負載保留在相同的帳戶中。在這個分散式模型中，管理 AWS KMS 金鑰的責任會落在應用程式團隊身上。在其他使用案例中，組織可能會選擇將 AWS KMS 金鑰儲存到集中式帳戶中。此集中式結構須實施其他政策來實現跨帳戶存取權，才能讓工作負載帳戶存取儲存在集中式帳戶中的金鑰，但此結構可能較適合跨多個 AWS 帳戶 共用單一金鑰的使用案例。

無論金鑰資料存放在何處，金鑰的存取權都應透過使用 金鑰政策 和 IAM 政策進行嚴格控管。金鑰政策是控制 AWS KMS 金鑰存取權的主要方式。此外，AWS KMS 金鑰授權可提供 AWS 服務的存取權，以代表您加密和解密資料。請安排時間來檢閱 AWS KMS 金鑰存取控制的最佳實務。

最佳實務是監控加密金鑰的使用情況，以偵測不尋常的存取模式。使用存放在 AWS KMS 中 AWS 管理的金鑰和客戶管理的金鑰執行的操作可記錄在 AWS CloudTrail 中，並且應定期檢閱。應特別注意監控金鑰銷毀事件。為了減少意外或惡意銷毀金鑰資料的情況，金鑰銷毀事件並不會立即刪除金鑰資料。嘗試刪除 AWS KMS 中的金鑰會受到 等待期的約束 (預設為 30 天)，讓管理員有時間檢閱這些動作，並在必要時撤回請求。

大多數 AWS 服務會以顯而易見的方式使用 AWS KMS，您唯一要做的就是決定要使用 AWS 管理或客戶管理的金鑰。如果您的工作負載要求直接使用 AWS KMS 來加密或解密資料，則最佳實務是使用 封套加密 來保護您的資料。此 AWS Encryption SDK 可為您的應用程式提供用戶端加密基本類型，以實作封套加密並與 AWS KMS 整合。

實作步驟

1. 確定適當的 金鑰管理選項 (AWS 管理或客戶管理的金鑰)。
   - 為了方便使用，AWS 為大多數服務提供了 AWS 擁有和 AWS 管理的金鑰，其提供靜態加密功能，而不需要管理金鑰資料或金鑰政策。
   - 使用客戶管理的金鑰時，請考慮使用預設金鑰存放區，以便在敏捷性、安全性、資料主權與可用性之間達到最佳平衡。其他使用案例可能會要求使用自訂金鑰存放區搭配 AWS CloudHSM 或 外部金鑰存放區。
2. 檢閱您用於工作負載的服務清單，以了解 AWS KMS 與服務整合的方式。例如，EC2 執行個體可以使用加密的 EBS 磁碟區，因此要確認從這些磁碟區建立的 Amazon EBS 快照同樣是使用客戶管理的金鑰加密，並減少意外洩漏未加密的快照資料。
   - AWS 服務如何使用 AWS KMS
   - 如需有關 AWS 服務所提供加密選項的詳細資訊，請參閱使用者指南中的「靜態加密」主題或服務的開發人員指南。
3. 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方式。
   - 入門：AWS Key Management Service (AWS KMS)
   - 檢閱 AWS KMS 金鑰存取控制的最佳實務。
4. 考慮 AWS Encryption SDK：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 Encryption SDK。
   - AWS Encryption SDK

5. 啟用 IAM Access Analyzer 以自動檢閱並在發現有過度廣泛的 AWS KMS 金鑰政策時發出通知。
6. 啟用 Security Hub 以在金鑰政策設定錯誤、有排定要刪除的金鑰，或有未啟用自動輪替的金鑰時收到通知。
7. 確定適合 AWS KMS 金鑰的記錄層級。由於 AWS KMS 的呼叫 (包括唯讀事件) 會加以記錄，因此與 AWS KMS 相關聯的 CloudTrail 日誌可能會變得很龐大。
   - 有些組織偏好將 AWS KMS 記錄活動分隔為單獨的軌跡記錄。如需詳細資訊，請參閱「使用 CloudTrail 記錄 AWS KMS API 呼叫」一節 (《AWS KMS 開發人員指南》中)。

資源

相關文件：

- AWS Key Management Service
- AWS 加密服務和工具
- 使用加密保護 Amazon S3 資料
- 封套加密
- 數位主權承諾
- 揭密 AWS KMS 金鑰操作、攜帶自有金鑰、自訂金鑰存放區，以及密文可攜性
- AWS Key Management Service 加密詳細資訊

相關影片：

- 在 AWS 中加密如何運作
- 保護 AWS 上的區塊儲存安全
- AWS 資料保護：使用鎖定、金鑰、簽章和憑證

相關範例：

- 使用 AWS KMS 實作進階存取控制機制

SEC08-BP02 強制靜態加密

您應該對靜態資料強制使用加密。在發生未授權存取或意外洩露的情況時，加密可保持敏感資料的機密性。

預期成果：私有資料應該預設在處於靜態時加密。加密有助於維持資料的機密性，並提供多一層保護以防有意或不慎的資料暴露或外洩。加密的資料必須先解密後才能讀取或存取。任何在未加密下儲存的資料都應該進行清查並加以控制。

常見的反模式：

- 未使用預設加密組態。
- 對解密金鑰提供過於寬鬆的存取權。
- 未監控加密和解密金鑰的使用。
- 在未加密的情況下儲存資料。
- 對所有資料使用相同的加密金鑰，無論資料使用方式、類型和分類。

未建立此最佳實務時的風險暴露等級：高

實作指引

在工作負載中將加密金鑰對應到資料分類。當對資料使用單一或極少數的加密金鑰時，此方法有助於防止過於寬鬆的存取權 (請參閱 SEC07-BP01 了解您的資料分類機制)。

AWS Key Management Service (AWS KMS) 與許多 AWS 服務整合，更方便您加密靜態資料。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。當使用 AWS KMS 時，考慮需要嚴格限制資料的程度。AWS 會代表您管理及使用預設和服務控制的 AWS KMS 金鑰。對於需要對基礎加密金鑰的精細存取權之敏感資料，可考慮客戶自管金鑰 (CMK)。您可全權控制 CMK，包括透過使用金鑰政策進行輪換和存取管理。

此外，Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon S3 可透過設定預設加密來支援強制加密。您可以使用 AWS Config 規則 自動檢查您是否正對如 Amazon Elastic Block Store (Amazon EBS) 磁碟區、Amazon Relational Database Service (Amazon RDS) 執行個體和 Amazon S3 儲存貯體等使用加密。

AWS 也提供用戶端加密選項，允許您在上傳到雲端之前加密資料。AWS Encryption SDK 提供使用封套加密來加密資料的方法。您提供包裝金鑰，而 AWS Encryption SDK 會為它加密的每個資料物件產生唯一的資料金鑰。如果您需要受管的單一租用戶硬體安全模組 (HSM)，可考慮 AWS CloudHSM。AWS CloudHSM 可讓您在 FIPS 140-2 3 級驗證的 HSM 上產生、匯入和管理加密金鑰。AWS CloudHSM 的一些使用案例包括保護用於核發憑證認證機構 (CA) 的私有金鑰，以及為 Oracle 資料庫啟用透明資料加密 (TDE)。AWS CloudHSM 用戶端 SDK 提供軟體，可讓您在將資料上傳到 AWS 之前，使用儲存在 AWS CloudHSM 內的金鑰加密資料用戶端。Amazon DynamoDB Encryption Client 還允許您在上傳到 DynamoDB 資料表之前，加密和簽署項目。

實作步驟

- 強制對 Amazon S3 執行靜態加密：實作 Amazon S3 儲存貯體預設加密。

  為新的 Amazon EBS 磁碟區設定預設加密：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 Amazon EBS 磁碟區。

  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。

  設定 Amazon RDS 加密：透過使用加密選項，為您的 Amazon RDS 資料庫叢集和靜態快照設定啟用加密。

  使用政策限制對適當主體的存取，為每個資料分類建立和設定 AWS KMS 金鑰：例如，建立一個 AWS KMS 金鑰用於加密生產資料，另一個金鑰用於加密開發或測試資料。您還可以提供金鑰來存取其他 AWS 帳戶。考慮針對開發和生產環境擁有不同的帳戶。如果您的生產環境需要解密開發帳戶中的成品，您可以編輯用來加密開發成品的 CMK 金鑰，使生產帳戶能夠解密這些成品。生產環境接著可以擷取解密的資料以用於生產。

  在其他 AWS 服務中設定加密：對於您使用的其他 AWS 服務，請檢閱該服務的安全文件，以確定該服務的加密選項。

資源

相關文件：

- AWS 加密工具
- AWS 文件
- AWS Encryption SDK
- AWS KMS 加密詳細資訊白皮書
- AWS Key Management Service
- AWS 加密服務和工具
- Amazon EBS 加密
- Amazon EBS 磁碟區的預設加密
- 加密 Amazon RDS 資源
- 如何針對 Amazon 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：

- AWS 中加密的運作方式
- 保護 AWS 上的區塊儲存安全

SEC08-BP03 自動化靜態資料保護

使用自動化來驗證和強制執行靜態資料控制措施。 使用自動掃描來偵測資料儲存解決方案的錯誤組態，並盡可能透過自動化的程式化回應執行補救措施。 將自動化納入 CI/CD 程序中，在部署到實際執行環境之前先偵測是否有資料儲存組態錯誤的情形。

預期成果：自動化系統會掃描和監控資料儲存位置，找出是否有控制措施組態錯誤、未經授權存取及意外使用的情況。 偵測到設定錯誤的儲存位置就會啟動自動化補救措施。 自動化程序會建立資料備份，並將不可變的副本儲存在原始環境之外。

常見的反模式：

- 未考慮在受支援的情況下，啟用預設加密設定的選項。
- 制定自動備份和復原策略時，未考慮安全事件還有操作事件。
- 未強制執行儲存服務的公開存取設定。
- 未監控和稽核保護靜態資料的控制措施。

建立此最佳實務的優勢：自動化有助於防止發生資料儲存位置設定錯誤的風險。此方式有助於防止錯誤組態進入您的實際執行環境。此最佳實務也有助於偵測並修正錯誤組態 (如發生)。

未建立此最佳實務時的風險暴露等級：中

實作指引

自動化是保護靜態資料的整體實務中反覆出現的「主題」。SEC01-BP06 自動部署標準安全控制措施說明了如何使用基礎設施即程式碼 (IaC) 範本 (例如使用 AWS CloudFormation) 擷取資源的組態。 這些範本已送交至版本控制系統中，且用於在 AWS 上透過 CI/CD 管道部署資源。 這些技術同樣適用於自動化資料儲存解決方案的組態，例如 Amazon S3 儲存貯體上的加密設定。

您可以在 CI/CD 管道中使用 AWS CloudFormation Guard 內的規則檢查您在 IaC 範本中定義的設定是否有組態錯誤。 您可以透過 AWS Config 監控尚未在 CloudFormation 或其他 IaC 工具中提供的設定，以檢查是否有組態錯誤。 Config 針對錯誤組態所產生的警示可以自動修復，如 SEC04-BP04 針對不合規資源實施補救措施中所述。

將自動化納入您的許可管理策略中，也是整體自動化資料防護措施的一環。SEC03-BP02 授予最低權限存取權和 SEC03-BP04 持續減少許可中說明了如何設定最低權限存取權政策，這些政策會受到 AWS Identity and Access Management Access Analyzer 的持續監控，以在能夠減少許可時產生調查結果。 除了自動化監控許可之外，您還可以設定 Amazon GuardDuty 來監看 EBS 磁碟區 (藉由 EC2 執行個體)、S3 儲存貯體和支援的 Amazon Relational Database Service 資料庫中是否存在異常資料存取行為。

自動化也會在偵測到敏感資料儲存於未經授權的位置時，發揮重要的作用。SEC07-BP03 自動識別和分類說明了 Amazon Macie 如何監控您的 S3 儲存貯體是否有非預期的敏感資料，並產生可啟動自動化回應的警示。

遵循 REL09 備份資料中的實務，制定自動化資料備份和復原策略。資料備份和復原對於操作事件，以及從安全事件中復原來說都相當重要。

實作步驟

1. 在 IaC 範本中擷取資料儲存組態。 使用自動化檢查在 CI/CD 管道中偵測組態錯誤。
   a. 您可以將 <ulink type="marketing" url="cloudformation">&CFN;</ulink> 用於 IaC 範本，並且將 CloudFormation Guard 用於檢查範本是否有組態錯誤。
   b. 使用 AWS Config 採取主動評估模式，執行規則。在建立資源之前，使用此設定作為 CI/CD 管道中的步驟來檢查資源是否合規。
2. 監控資源是否有資料儲存組態錯誤。
   a. 設定 AWS Config 來監控資料儲存資源是否有控制組態方面的變更，並在偵測到組態錯誤時產生警示，以調用修復動作。
   b. 如需有關自動化補救措施的詳細指引，請參閱 SEC04-BP04 針對不合規資源實施補救措施。
3. 透過自動化持續監控並減少資料存取許可。
   a. IAM Access Analyzer 可持續執行，並在有可能減少許可時產生警示。
4. 監控並警示異常資料存取行為。
   a. GuardDuty 會監看 EBS 磁碟區、S3 儲存貯體及 RDS 資料庫等資料儲存資源中，是否存在已知的威脅特徵和偏離基準存取行為。
5. 監控並在敏感資料儲存於非預期位置時發出警示。
   a. 使用 Amazon Macie 持續掃描您的 S3 儲存貯體是否有敏感資料。
6. 自動保護和加密資料備份。
   a. AWS Backup 是受管服務，會在 AWS 上建立各種資料來源的加密和安全備份。 彈性災難復原可讓您透過以秒為單位測量的復原點目標 (RPO)，複製完整的伺服器工作負載並維持持續的資料保

護。 您可以設定讓兩種服務搭配運作，以自動建立資料備份並將其複製到容錯移轉位置。 這樣做有助於在受到操作或安全事件影響時，保持資料的可用性。

資源

相關的最佳實務：

- SEC01-BP06 自動部署標準安全控制措施
- SEC03-BP02 授予最低權限存取權
- SEC03-BP04 持續減少許可
- SEC04-BP04 針對不合規資源實施補救措施
- SEC07-BP03 自動識別和分類
- REL09-BP02 保護和加密備份
- REL09-BP03 自動執行資料備份

相關文件：

- AWS 方案指引：自動加密現有和新的 Amazon EBS 磁碟區
- 在 AWS 上使用 NIST 網路安全架構 (CSF) 進行勒索軟體風險管理

相關範例：

- 如何使用 AWS Config 主動式規則和 AWS CloudFormation 勾點來防止建立不合規的雲端資源
- 使用 AWS Backup 自動化並集中管理 Amazon S3 的資料保護
- AWS re:Invent 2023 - 使用 Amazon EBS 快照實施主動式資料保護
- AWS re:Invent 2022 - 利用現代化資料保護建置並自動化以強化韌性

相關工具：

- AWS CloudFormation Guard
- AWS CloudFormation Guard 規則登錄檔
- IAM Access Analyzer
- Amazon Macie
- AWS Backup

- 彈性災難復原

SEC08-BP04 強制存取控制

若要協助保護您的靜態資料，使用隔離和版本控制等機制來強制存取控制，並套用最低權限原則。防止授予對您資料的公開存取權。

預期成果：確認只有授權使用者能夠在必要時存取資料。透過定期備份和版本控制來保護您的資料以防有意或不慎修改或刪除資料。將重要資料與其他資料分離，以保護其機密性和資料完整性。

常見的反模式：

- 將具有不同敏感度需求或分類的資料存放在一起。
- 對解密金鑰使用過於寬鬆的許可。
- 資料分類不當。
- 未保留重要資料的詳細備份。
- 對生產資料提供持續存取權。
- 未稽核資料存取或定期審查許可。

未建立此最佳實務時的風險暴露等級：低

實作指引

多項控制可協助您保護靜態資料，包括存取 (使用最低權限)、隔離和版本控制。對資料的存取應使用偵測機制進行稽核，例如 AWS 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，並建立計畫以隨著時間減少可用的資料量。

Amazon S3 Glacier Vault Lock 和 Amazon S3 物件鎖定為 Amazon S3 中的物件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。

實作步驟

- 強制存取控制：強制最低權限存取控制，包括對加密金鑰的存取。
- 根據不同的分類層級分離資料：針對資料分類層級使用不同的 AWS 帳戶，並使用 AWS Organizations 來管理這些帳戶。
- 審查 AWS Key Management Service (AWS KMS) 政策：審查 AWS KMS 政策中授予的存取層級。

- 審查 Amazon S3 儲存貯體和物件許可：定期審查 S3 儲存貯體政策中授予的存取層級。最佳實務是避免使用可公開讀取或寫入的儲存貯體。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。確認不允許公開存取的儲存貯體已正確設定為禁止公開存取。依照預設，所有 S3 儲存貯體皆為私有，只有明確獲得存取權的使用者得以存取。
- 啟用 AWS IAM Access Analyzer： IAM Access Analyzer 會分析 Amazon S3 儲存貯體並在 S3 政策將存取權授予外部實體時產生發現結果。
- 適當時，啟用 Amazon S3 版本控制和物件鎖定。
- 使用 Amazon S3 庫存：Amazon S3 庫存可用來稽核和報告 S3 物件的複寫和加密狀態。
- 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與工作負載外部的 AWS 帳戶共用。
- 審查 AWS Resource Access Manager 定期共用以確定是否應該持續共用資源。 Resource Access Manager 可讓您共用 Amazon VPC 內的資源，例如 AWS 網路防火牆政策、Amazon Route 53 解析器規則和子網路。定期稽核共用的資源並停止共用不再需要共用的資源。

資源

相關的最佳實務：

- SEC03-BP01 定義存取需求
- SEC03-BP02 授予最低權限存取權

相關文件：

- AWS KMS 加密詳細資訊白皮書
- 管理對 Amazon S3 資源的存取許可的簡介
- 管理對您 AWS KMS 資源的存取概觀
- AWS Config 規則
- Amazon S3 + Amazon CloudFront：雲端的最佳拍檔
- 使用版本控制
- 使用 Amazon S3 物件鎖定來鎖定物件
- 共用 Amazon EBS 快照
- 共用的 AMI
- 在 Amazon S3 上託管單頁應用程式

相關影片：

- 保護 AWS 上的區塊儲存安全

SEC 9.如何保護傳輸中資料？

實作多項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 強制傳輸中加密
- SEC09-BP03 驗證網路通訊

SEC09-BP01 實作安全金鑰和憑證管理

Transport Layer Security (TLS) 憑證可用來保護網路通訊，和建立網際網路跟私有網路中的網站、資源和工作負載的身份。

預期成果： 能夠在公開金鑰基礎設施 (PKI) 佈建、部署、儲存和更新憑證的安全憑證管理系統。安全金鑰與憑證管理機制可以防止憑證私有金鑰資料外洩，也能定期自動更新憑證。它也能與其他服務整合，為工作負載內的機器資源提供安全的網路通訊和身分識別。金鑰資料絕不可供真人身分存取。

常見的反模式：

- 在憑證部署或更新程序期間執行手動步驟。
- 設計私有 CA 時，請忽略憑證授權單位 (CA) 階層。
- 針對公用資源使用自我簽署憑證。

建立此最佳實務的優勢：

- 透過自動化部署和更新來簡化憑證管理
- 鼓勵使用 TLS 憑證加密傳輸中的資料
- 增加憑證授權單位所採取憑證動作的安全性和可稽核性
- 在 CA 階層中不同層次的管理責任組織

未建立此最佳實務時的曝險等級： 高

實作指引

現代工作負載可透過利用 TLS 等 PKI 通訊協定，來廣泛使用加密網路通訊。PKI 憑證管理可能很複雜，但自動化憑證佈建、部署和更新可以減少憑證管理相關障礙。

AWS 提供兩種管理一般用途 PKI 憑證的服務： AWS Certificate Manager 和 AWS Private Certificate Authority (AWS Private CA)ACM 是客戶在公有和私有 AWS 工作負載中，用來佈建、管理和佈數憑證的主要服務。ACM 則使用 AWS Private CA 的公有憑證授權單位來發行憑證，並 整合 至許多其他 AWS 受管服務，以提供安全的工作負載 TLS 憑證。

AWS Private CA 可讓您建立自己的根憑證授權單位或下層憑證授權單位，並透過 API 發行 TLS 憑證。在您控制和管理 TLS 連線用戶端信任鏈時，您可以使用這些憑證類型。除了 TLS 使用案例之外，AWS Private CA 可以用來發行憑證給 Kubernetes pods、重要裝置產品證明、程式碼簽署，以及其他使用案例，過程中搭配 自訂範本。您也可以使用 IAM Roles Anywhere 提供臨時 IAM 憑證給具有您私有 CA 簽發 X.509 憑證的內部部署工作負載。

除了 ACM 和 AWS Private CA 之外， AWS IoT Core 也為物聯網裝置提供特別支援，用來佈建、管理和部署 PKI 憑證。AWS IoT Core 提供特別機制給 上線物聯網裝置。 大規模提供特別機制至您的公有金鑰基礎設施。

建立私有 CA 階層時的考量

您要建立私有 CA 時，請務必特別留意，預先正確設計 CA 階層。建立私有 CA 階層時，最佳做法是將 CA 階層的每個層級部署到個別 AWS 帳戶。這個刻意的步驟會減少 CA 階層中每個層級的界面面積，讓您更容易察覺 CloudTrail 日誌資料的異常狀況，並在出現未經授權的帳戶存取動作時降低存取或影響範圍。根 CA 應位於自己的個別帳戶中，且只能用來發行一個或多個中繼 CA 憑證。

接著，請在不同於根 CA 帳戶的其他帳戶中建立一個或多個中繼 CA，為終端使用者、裝置或其他工作負載發行憑證。最後，請將憑證從根 CA 發行至中繼 CA，這個動作會將憑證發行給您的終端使用者或裝置。如需深入了解 CA 部署規畫和 CA 階層設計，包括恢復能力、跨區域複寫、在組織中共用 CA 等規畫，請參閱 規劃您的 AWS Private CA 部署。

實作步驟

1. 決定使用案例所需的相關 AWS 服務：
   - 許多使用案例都可以利用 AWS 的現有公有金鑰基礎設施搭配，過程中使用 AWS Certificate Manager。ACM 可用於為 Web 伺服器、負載平衡器或其他用途部署 TLS 憑證。
   - 考慮 AWS Private CA 何時需要建立自己的私有憑證授權單位階層，或需要存取可匯出憑證的權限。ACM 可以接著用於發行 許多類型的終端實體憑證 過程中會使用 AWS Private CA。

- 針對必須為嵌入式物聯網 (IoT) 裝置大規模佈建憑證的使用案例，請考慮 AWS IoT Core。

2. 盡可能實施自動憑證續約：
   - 使用 使用 ACM 的受管更新 搭配 ACM 發行的憑證和 AWS 受管服務。
3. 建立日誌和稽核軌跡：
   - 啟用 CloudTrail 日誌 以追蹤對持有憑證授權單位之帳戶的存取權。請考慮在 CloudTrail 中設定日誌檔完整性驗證，以驗證日誌資料的真實性。
   - 定期產出 稽核報告 ，其中列出您的私有 CA 發行或撤銷的憑證。這些報告可以匯出到 S3 儲存貯體。
   - 部署私有 CA 時，您也需要建立 S3 儲存貯體來儲存憑證撤銷清單 (CRL)。如需詳細了解如何根據工作負載需求設定此 S3 儲存貯體，請參閱 規劃憑證撤銷清單 (CRL)。

資源

相關的最佳實務：

- SEC02-BP02 使用臨時憑證
- SEC08-BP01 實作安全金鑰管理
- SEC09-BP03 驗證網路通訊

相關文件：

- 如何在 AWS 中託管和管理整個私有憑證基礎設施
- 如何鞏固用於汽車和製造領域的企業規模 ACM 私有 CA 階層
- 私有 CA 最佳實務
- 如何使用 AWS RAM 共用您的 ACM 私有 CA 跨帳戶

相關影片：

- 啟動 AWS Certificate Manager 私有 CA (研討會)

相關範例：

- 私人 CA 研討會
- IOT Device Management 研討會 (包括裝置佈建)

相關工具：

- 要使用 AWS Private CA 的 Kubernetes cert-manager 外掛程式

SEC09-BP02 強制傳輸中加密

根據您組織的政策、法規義務和標準強制已定義的加密需求，協助滿足組織、法律和合規上的要求。只有在虛擬私有雲端 (VPC) 以外傳輸敏感資料時才使用加密通訊協定。加密有助於保持資料完整性，甚至當資料傳輸於不受信任的網路時。

預期成果：所有資料都應該在傳輸時使用安全的 TLS 通訊協定和密碼套件加密。您的資源與網際網路之間的網路流量必須經過加密以緩解對資料的未授權存取。完全位於您內部 AWS 環境的網路流量應該盡可能使用 TLS 加密。AWS 內部網路會經預設加密，而且 VPC 內的網路流量無法受詐騙或嗅探，除非未授權方獲得對產生流量的資源 (例如 Amazon EC2 執行個體和 Amazon ECS 容器) 的存取權。考慮使用 IPsec 虛擬私有網路 (VPN) 保護網路對網路流量。

常見的反模式：

- 使用 SSL、TLS 和其他套件元件已棄用的版本 (例如，SSL v3.0、1024 位元 RSA 金鑰和 RC4 密碼)。
- 允許未加密的 (HTTP) 流量來往面向公眾的資源。
- 未監控 X.509 憑證並在到期前更換。
- 對 TLS 使用自我簽署的 X.509 憑證。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 服務提供使用 TLS 的 HTTPS 端點以進行通訊，在與 AWS API 通訊時提供傳輸中加密。不安全的通訊協定 (如 HTTP) 可以在 VPC 中透過使用安全群組加以稽核和封鎖。HTTP 請求也可以在 Amazon CloudFront 中或 Application Load Balancer 上自動重新導向至 HTTPS 。您可以全權控制您的運算資源，以在各個服務中實作傳輸中加密。此外，還可以從外部網路或 AWS Direct Connect 使用 VPN 連線功能進入 VPC，加速流量加密。確認您的用戶端至少使用 TLS 1.2 對 AWS API 進行呼叫，因為 AWS 將於 2023 年 6 月棄用 TLS 1.0 和 1.1。如果您有特殊需求，AWS Marketplace 備有第三方解決方案。

實作步驟

- 強制傳輸中加密：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，設定安全群組，僅允許 HTTPS 協定連至 Application Load Balancer 或 Amazon EC2 執行個體。
- 在邊緣服務中設定安全通訊協定： 使用 Amazon CloudFront 設定 HTTPS並使用適用於您的安全狀態和使用案例的安全設定檔。
- 使用 VPN 進行外部連線：考慮使用 IPsec VPN，保護點對點或網路對網路連線，以協助提供資料隱私和完整性。
- 在負載平衡器中設定安全的通訊協定：選擇安全政策，以提供要連接到接聽程式的用戶端所支援的最強固的密碼套件。為您的 Application Load Balancer 建立 HTTPS 接聽程式.
- 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。
- 設定安全通訊協定：檢閱 AWS 服務文件以確定傳輸中加密功能。
- 設定上傳至 Amazon S3 儲存貯體時的安全存取：使用 Amazon S3 儲存貯體政策控制對資料強制安全存取。
- 考慮使用 AWS Certificate Manager：ACM 可讓您佈建、管理和部署 TLS 憑證與 AWS 服務搭配使用。
- 考慮使用 AWS Private Certificate Authority 滿足私有 PKI 需求：AWS Private CA 可讓您建立私有憑證認證機構 (CA) 階層來核發可用於建立已加密 TLS 管道的終端實體 X.509 憑證。

資源

相關文件：

- AWS 文件
- 搭配 CloudFront 使用 HTTPS
- 使用 AWS Virtual Private Network 將您的 VPC 連接到遠端網路
- 為您的 Application Load Balancer 建立 HTTPS 接聽程式 .
- 教學課程：在 Amazon Linux 2 上設定 SSL/TLS
- 使用 SSL/TLS 加密與資料庫執行個體的連線
- 設定連線的安全性選項

SEC09-BP03 驗證網路通訊

This best practice was updated with new guidance on December 6, 2023.

使用支援身分驗證的通訊協定 (Transport Layer Security (TLS) 或 IPsec) 來驗證通訊的身分。

設計工作負載，以在每當服務、應用程式或使用者之間進行通訊時，使用安全、經驗證的網路通訊協定。使用支援驗證和授權的網路通訊協定可提供更強大的網路流量控制能力，並減少未經授權存取所造成的影響。

預期成果：設計出工作負載，讓其有明確定義的服務間資料平面和控制平面流量。在技術允許的情況下，流量要使用經過驗證和加密的網路通訊協定。

常見的反模式：

- 工作負載內有未經加密或驗證的流量。
- 在多個使用者或實體之間重複使用驗證憑證。
- 僅依賴網路控制作為存取控制機制。
- 建立自訂驗證機制，而非依賴業界標準的驗證機制。
- 服務元件或 VPC 中的其他資源之間有過於寬鬆的流量。

建立此最佳實務的優勢：

- 將未經授權存取所造成的影響範圍限制在工作負載的某個部分。
- 提供只會由已驗證實體執行動作的更高層級保證。
- 透過清楚地定義並強制執行預期的資料傳輸介面來改善服務的去耦。
- 透過請求歸因和明確定義的通訊介面，增強監控、記錄和事件應變。
- 結合網路控制與驗證和授權控制，為您的工作負載提供深度防禦。

未建立此最佳實務時的風險暴露等級：低

實作指引

您工作負載的網路流量模式可分為兩個類別：

- 東西流量代表構成工作負載的服務之間的流量。
- 南北流量代表工作負載和取用者之間的流量。

加密南北流量是常見的做法，使用經過驗證的通訊協定來保護東西流量則較不常見。現代安全實務的建議是，單靠網路設計並無法讓兩個實體之間建立信任的關係。當兩個服務可能位於一個共通的網路邊界內時，最佳做法仍是對這些服務之間的通訊進行加密、驗證和授權。

舉例來說，無論請求來自哪個網路，AWS 服務 API 都會使用 AWS 第 4 版簽署程序 (SigV4) 簽署通訊協定來驗證呼叫者。此驗證可確保 AWS API 可以驗證發出動作請求的身分，該身分接著可與政策結合來作出授權決策，決定是否應允許該動作。

Amazon VPC Lattice 和 Amazon API Gateway 等服務可讓您使用相同的 SigV4 簽署通訊協定，為自己的工作負載中的東西流量新增驗證和授權功能。如果 AWS 環境以外的資源需要與要求進行 SigV4 型驗證和授權的服務進行通訊，您可以在非 AWS 資源上使用 AWS Identity and Access Management (IAM) Roles Anywhere 來取得臨時的 AWS 憑證。使用這些憑證，便可透過 SigV4 簽署服務請求以授權存取。

用於驗證東西流量的另一種常見機制是 TLS 相互驗證 (mTLS)。許多物聯網 (IoT)、企業對企業應用程式和微服務都使用 mTLS，透過使用用戶端和伺服器端 X.509 憑證來驗證 TLS 通訊兩端的身分。這些憑證可由 AWS Private Certificate Authority (AWS Private CA) 核發。您可以使用 Amazon API Gateway 和 AWS App Mesh 等服務，為工作負載之間或工作負載內部的通訊提供 mTLS 驗證。mTLS 會為 TLS 通訊的兩端提供驗證資訊，但不提供授權機制。

最後，OAuth 2.0 和 OpenID Connect (OIDC) 是兩種常用於控制使用者對服務存取行為的通訊協定，但現在也變成服務對服務流量的熱門通訊協定。API Gateway 會提供 JSON Web 權杖 (JWT) 授權器，可讓工作負載使用 OIDC 或 OAuth 2.0 身分提供者所核發的 JWT 來限制 API 路由的存取。OAuth2 的範圍可作為基本授權決策的來源，但仍需要在應用程式層實作授權檢查，而且單靠 OAuth2 範圍並無法支援更複雜的授權需求。

實作步驟

- 定義並記錄您的工作負載網路流量：實作深度防禦策略的第一步是定義工作負載的流量。
  - 建立可清楚定義構成工作負載的不同服務間資料傳輸方式的資料流程圖。此圖是透過已驗證的網路通道強制執行這些流程的第一步。
  - 在開發和測試階段檢測您的工作負載，以驗證資料流程圖是否準確反映工作負載在執行期的行為。
  - 資料流程圖在執行威脅建模練習時也很有用，如 SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施中所述。
- 建立網路控制：考慮用來建立與資料流程一致的網路控制的 AWS 功能。網路邊界不應成為唯一的安全控制，但其可在深度防禦策略中提供一個保護層，以保護您的工作負載。
  - 使用安全群組建立定義和限制資源之間的資料流程。
  - 考慮使用 AWS PrivateLink 與支援 AWS PrivateLink 的 AWS 和第三方服務進行通訊。透過 AWS PrivateLink 介面端點傳送的資料會保留在 AWS 網路骨幹內，不會周遊公用網際網路。
- 在工作負載中跨服務實作驗證和授權：選擇最適合用來在您工作負載中提供經驗證加密流量的 AWS 服務集。

  - 考慮用來保護服務對服務通訊的 Amazon VPC Lattice。VPC Lattice 可以使用 SigV4 驗證結合驗證政策來控制服務對服務的存取。
  - 對於使用 mTLS 的服務對服務通訊，請考慮 API Gateway 或 App Mesh。AWS Private CA 可用來建立能夠核發憑證以與 mTLS 搭配使用的私有 CA 階層。
  - 與使用 OAuth 2.0 或 OIDC 的服務進行整合時，請考慮使用 JWT 授權器的 API Gateway。
  - 對於工作負載和 IoT 裝置之間的通訊，請考慮 AWS IoT Core，它提供了幾種網路流量加密和驗證選項。
- 監控未經授權的存取：持續監控是否有意外的通訊管道、嘗試存取受保護資源的未經授權主體，以及其他不當的存取模式。
  - 如果使用 VPC Lattice 來管理服務的存取，請考慮啟用和監控 VPC Lattice 存取日誌。這些存取日誌包括請求方實體的資訊、包括來源和目的地 VPC 在內的網路資訊，以及請求中繼資料。
  - 考慮啟用 VPC Flow Logs 來擷取網路流量上的中繼資料，並定期檢閱是否有異常狀況。
  - 如需更多有關規劃、模擬和應對安全事件的指引，請參閱 AWS 安全事件應變指南和 AWS Well Architected Framework 安全支柱的事件應變章節。

資源

相關的最佳實務：

- SEC03-BP07 分析公有和跨帳戶存取權
- SEC02-BP02 使用臨時憑證
- SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施

相關文件：

- 評估用來保護 Amazon API Gateway API 的存取控制方法
- 為 REST API 設定相互 TLS 驗證
- 如何使用 JWT 授權器保護 API Gateway HTTP 端點
- 使用 AWS IoT Core 憑證提供者來授權 AWS 服務的直接呼叫
- AWS 安全事件應變指南

相關影片：

- AWS re:invent 2022：向您介紹 VPC Lattice

- AWS re:invent 2020：針對 AWS 上 HTTP API 的無伺服器 API 驗證

相關範例：

- Amazon VPC Lattice 研討會
- 零信任第 1 集 - Phantom Service Perimeter 研討會

事故回應

問題

- SEC 10.如何預測、回應事故以及從事故中復原？

SEC 10.如何預測、回應事故以及從事故中復原？

即使採用了成熟的預防和偵測控制，您的組織仍應實作機制，以回應並緩和安全事故的潛在影響。您的準備工作會大大地影響團隊在事故發生時能否有效運作，以隔離、遏制問題並進行鑑識，以及將營運恢復到已知的良好狀態。在安全事故發生前先備妥工具和存取權，然後在演練日期間定期練習事故回應，有助於確保您能夠復原，同時盡量減少業務中斷。

最佳實務

- SEC10-BP01 確定關鍵人員和外部資源
- SEC10-BP02 制定事件管理計畫
- SEC10-BP03 準備鑑識功能
- SEC10-BP04 開發和測試安全性事故應變程序手冊
- SEC10-BP05 預先佈建存取權
- SEC10-BP06 預先部署工具
- SEC10-BP07 執行模擬
- SEC10-BP08 建立從事故中學習的架構

SEC10-BP01 確定關鍵人員和外部資源

確定可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。

預期成果：您備有一份關鍵人員名單，包含其聯絡資訊，以及他們在回應安全事件時扮演的角色。您可定期檢閱並更新此資訊，以在內部和外部工具中反映人員變更。您在記錄此資訊時考量所有第三方

服務供應商和廠商，包括安全合作夥伴、雲端供應商，以及軟體即服務 (SaaS) 應用程式。在安全事件期間，人員會負起適當層級的責任、得知適當的關聯內容，並擁有適當的存取權能夠做出回應和進行復原。

常見的反模式：

- 回應安全事件時，未隨時備妥包含聯絡資訊、角色和職責的最新關鍵人員名單。
- 假設每個人都了解回應事件和從事件復原時的人員、相依關係、基礎設施和解決方案。
- 沒有能夠呈現主要基礎設施或應用程式設計的文件或知識儲存庫。
- 未制定適當的新員工上任流程，導致他們無法有效地參與安全事件回應工作，例如進行事件模擬。
- 在安全事件期間，當關鍵人員暫時聯絡不到或無法回應時，沒有向上呈報的管道。

建立此最佳實務的優勢：此實務可減少發生事件時，花在識別正確人員和其角色的權責劃分和回應時間。隨時備妥一份最新的關鍵人員及其角色的名單，您就能找到正確的人員進行權責劃分並從事件中復原，藉此在發生事件時盡量減少時間浪費。

未建立此最佳實務時的風險暴露等級：高

實作指引

識別組織中的關鍵人員：隨時備妥組織內應對特定事件所需的人員聯絡名單。發生人員變動 (例如組織變動、晉升和團隊變動) 時，定期檢閱並更新此資訊。這對於事件管理者、事件應變人員和通訊負責人等關鍵角色尤其重要。

- 事件管理者：事件管理者在事件回應期間具有整體權限。
- 事件應變人員：事件應變人員負責調查和補救活動。這些人員可能根據事件類型而有所不同，但通常是負責受影響應用程式的開發人員和營運團隊。
- 通訊負責人：通訊負責人負責內部和外部溝通，特別是與公家機關、監管機構和客戶之間的溝通。
- 主題專家 (SME)：如果是分散式和自主團隊，我們建議您組成 SME 團隊來負責關鍵任務工作負載。他們負責提供對事件所涉及關鍵工作負載的操作和資料分類的深入洞悉。

請考慮使用 AWS Systems Manager Incident Manager 功能來擷取主要聯絡人、制定回應計劃、自動排定待命時間表，以及制定向上呈報計劃。自動排定待命時間表並輪替所有人員，藉此將工作負載的責任分散給其負責人。這樣有助於建立良好的實務，例如，發出相關的指標和日誌，以及定義對工作負載至關重要的警報閾值。

確定外部合作夥伴：企業使用由獨立軟體開發廠商 (ISV)、合作夥伴和子承包商建置的工具，為其客戶打造出獨特的解決方案。邀集上述多方的關鍵人員參與，他們可以協助回應事件並從中復原。我們建議您註冊適當層級的 AWS Support，以便透過支援案例迅速與 AWS 主題專家取得聯繫。請考慮針對工作負載的所有關鍵解決方案提供者做出類似的安排。有些安全事件會促使公開上市公司通知相關的公家機關和監管機構有關事件的情形和影響。隨時備妥並更新相關部門和負責人的聯絡資訊。

實作步驟

1. 設定事件管理解決方案。
   a. 考慮在您的安全工具帳戶中部署 Incident Manager。
2. 在事件管理解決方案中定義聯絡人。
   a. 針對每個聯絡人至少定義兩種類型的聯絡管道 (例如簡訊、電話或電子郵件)，以確保在發生事件時能夠與他們取得聯繫。
3. 定義回應計劃。
   a. 確定發生事件時最合適參與的聯絡人。定義符合要參與之人員角色 (而非個別聯絡人) 的向上呈報計劃。考慮納入可負責通知外部實體的聯絡人，即使他們並未直接參與事件解決工作。

資源

相關的最佳實務：

- OPS02-BP03 確定負責營運活動績效的負責人

相關文件：

- AWS 安全事件應變指南

相關範例：

- AWS 客戶程序手冊架構
- 準備和回應 AWS 環境中的安全事件

相關工具：

- AWS Systems Manager Incident Manager

相關影片：

- Amazon 在開發期間採取的安全方法

SEC10-BP02 制定事件管理計畫

為事件應變制定的第一份文件是事件應變計畫。事件應變計畫應是您事件應變計畫和策略的基礎。

建立此最佳實務的優勢： 開發全面且明確定義的事件應變程序，是成功且可擴展的事件應變計畫的關鍵。當安全事件發生時，明確的步驟和工作流程可協助您及時因應。您可能已具備現有的事件應變程序。無論您目前的狀態為何，都必須定期更新、重複執行和測試事件應變程序。

未建立此最佳實務時的曝險等級： 高

實作指引

事件管理計畫對於回應、減輕安全事件所造成潛在影響並從中復原而言至關重要。事件管理計畫是結構清晰的程序，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。建立事件管理計畫時，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，在操作含有歐洲個人身分識別資訊 (PII) 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 中的工作負載建立事件管理計畫時，請從 AWS 共同的責任模型 開始建置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。此 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畫時的重要概念和基礎指引。

有效的事件管理計畫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畫時，請考慮使用以下詳述的實作計畫。

實作步驟

定義角色和責任

處理安全事件時，需要跨組織的紀律和採取行動的傾向。在事件發生期間，您的組織結構中應該有不同的人員在事件期間負責、當責、備詢及保持通訊，例如人力資源部 (HR)、行政團隊和法務部的代表。請考量這些角色和責任，以及是否必須涉及任何第三方。請注意，許多地區都有當地法律會管理合法和

不合法的事務。儘管為您的安全應變計畫建置負責、當責、備詢及通訊 (RACI) 圖表似乎很形式化，但這麼做可以促進快速直接的溝通，並清楚地概述活動不同階段的領導層。

在事件發生時，納入受影響的應用程式和資源的擁有者及開發人員是非常重要的，因為他們是主題專家 (SME)，可以提供資訊和背景資訊以協助衡量影響性。在您仰賴開發人員和應用程式擁有者的專業知識進行事件應變之前，請務必先與他們建立關係。應用程式擁有者或 SME (例如您的雲端管理員或工程師) 可能需要在環境不同於前或複雜，或是應變人員無法存取的情況下採取行動。

最後，值得信賴的合作夥伴可能會參與調查或回應，因為他們可以提供額外的專業知識和有價值的審視。若您自己的團隊沒有這些技能，您可能需要對外招聘以尋求幫助。

了解 AWS 應變團隊和支援

- AWS Support
  - AWS Support 提供一系列的計畫，可讓您存取工具和專業知識，以支援 AWS 解決方案的成功和運作狀態。如果您需要技術支援和更多資源以利規劃、部署和優化 AWS 環境，您可以選取最符合您 AWS 使用案例的支援計畫。
  - 考慮以 支援中心 (位於 AWS Management Console，需要登入) 作為中心聯絡窗口，以取得影響 AWS 資源的問題所需的支援。對 AWS Support 的存取由 AWS Identity and Access Management 所控制。如需取得 AWS Support 功能存取權的詳細資訊，請參閱 AWS Support 入門。
- AWS 客戶事件應變團隊 (CIRT)
  - AWS 客戶事件應變團隊 (CIRT) 是一個專門的全天候全球 AWS 團隊，在客戶端的有效安全事件期間為客戶提供支援 - AWS 共同的責任模型。
  - 當 AWS CIRT 支援您時，他們會為 AWS 上的有效安全事件提供分類和復原方面的協助。他們可透過使用 AWS 服務日誌協助進行根本原因分析，並為您提供復原的建議。他們也可提供安全建議和最佳實務，以協助您避免事後發生安全事件。
  - AWS 客戶可透過以下途徑洽詢 AWS CIRT： AWS Support 案例。
- DDoS 應變支援
  - AWS 提供 AWS Shield，其中包含受管的分散式拒絕服務 (DDoS) 保護服務，可為執行於 AWS 的 Web 應用程式提供保護。Shield 提供一律開啟的偵測和自動內嵌緩解措施，可將應用程式停機時間和延遲降至最低，讓您無須聯絡 AWS Support 即可享有 DDoS 保護。Shield 有兩個層級：AWS Shield Standard 和 AWS Shield Advanced。若要了解這兩個層級的差異，請參閱 Shield 功能文件。
- AWS Managed Services (AMS)
  - AWS Managed Services (AMS) 可讓您持續管理 AWS 基礎設施，讓您專注在自己的應用程式上。實作最佳實務以維護您的基礎設施，AMS 有助於降低營運開銷和風險。AMS 會自動執行常見

的活動，例如，變更請求、監控、修補程式管理、安全性和備份服務，而且提供佈建、執行和支援基礎設施的完整生命週期服務。

- AMS 負責部署安全偵測控制套件，並提供全年無休的第一線提醒應變措施。提醒啟動時，AMS 會依照一組標準的自動化和手動程序手冊來驗證回應的一致性。這些程序手冊會在上線期間與 AMS 客戶共享，讓他們能夠透過 AMS 來制定和協調應變措施。

制定事件應變計畫

事件應變計畫應是您事件應變計畫和策略的基礎。事件應變計畫應納入正式文件中。事件應變計畫通常包含下列章節：

- 事件應變團隊概觀： 概述事件應變團隊的目標和職能。
- 角色和責任： 列出事件應變利害關係人，並詳細說明他們在事件發生時的角色。
- 通訊計畫： 詳細說明聯絡資訊，以及您在事件期間要如何進行通訊。
- 備份通訊方式： 最佳實務是將頻外通訊作為事件通訊的備用方法。舉例來說，AWS Wickr 就是提供安全頻外通訊通道的應用程式。
- 事件應變的階段和應採取的行動： 列舉事件應變的階段 (例如偵測、分析、消除、抑制及復原)，包括要在這些階段中採取的高階動作。
- 事件嚴重性和優先順序定義： 詳細說明如何分類事件的嚴重性、如何排定事件的優先順序，以及嚴重性定義對於呈報程序有何影響。

儘管不同規模和產業的公司都會有這些章節，但每個組織的事件應變計畫都是獨一無二的。您必須建立最適合貴組織的事件應變計畫。

資源

相關的最佳實務：

- SEC04 (您如何偵測和調查安全事件？)

相關文件：

- AWS 安全事件應變指南
- NIST：電腦安全事件處理指南

SEC10-BP03 準備鑑識功能

在安全事故發生之前，請將開發鑑識功能納入考量，以協助安全事件調查。

未建立此最佳實務時的曝險等級： 中

傳統內部部署鑑識的概念適用於 AWS。如需開始在 AWS 雲端 中建置鑑識功能的重要資訊，請參閱 AWS 雲端 中的鑑識調查環境策略。

設定鑑識的環境和 AWS 帳戶 結構後，請定義在四個階段有效執行合理鑑識方法所需的技術：

- 收集: 收集相關 AWS 日誌，例如 AWS CloudTrail、AWS Config、VPC 流程日誌和主機層級日誌。收集受影響 AWS 資源的快照、備份和記憶體傾印 (如果有的話)。
- 測驗： 檢視透過擷取和評估相關資訊所收集的資料。
- 分析： 分析收集的資料，以了解事故並從中得出結論。
- 報告： 呈現分析階段所產生的資訊。

實作步驟

準備鑑識環境

AWS Organizations 可協助您隨著 AWS 資源的成長和擴展，集中管理和控管 AWS 環境。AWS 組織會合併 AWS 帳戶，以便您可以將其當作一個單位進行管理。您可以使用組織單位 (OU) 將帳戶分組，以作為一個單位進行管理。

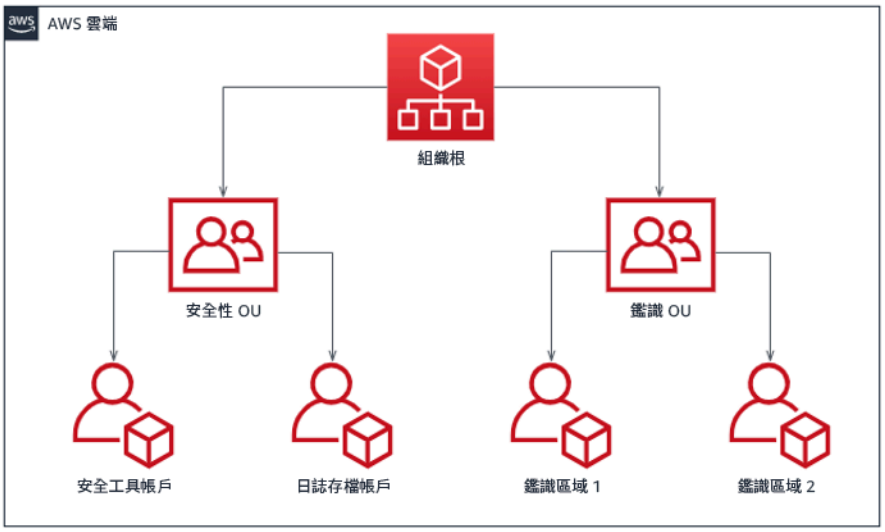
如需事故應變，建議您建立可支援事故應變功能的 AWS 帳戶 結構，其中包括 安全性 OU 和 鑑識 OU。在安全性 OU 中，您應該擁有下列項目的帳戶：

- 日誌存檔： 使用有限的許可彙總日誌存檔 AWS 帳戶 中的日誌。
- 安全性工具： 將安全性服務集中在安全工具 AWS 帳戶 中。此帳戶會以安全性服務的委派系統管理員身分運作。

在鑑識 OU 中，您可以選擇為營運所在的每個區域實作一或多個鑑識帳戶，具體視哪個區域最適合您業務和營運模式而定。如果您為每個區域建立鑑識帳戶，則可以防止該區域以外的 AWS 資源建立，並降低將資源複製到非預期區域的風險。例如，如果您只在 US East (N. Virginia) Region (#### (us-east-1) 和 US West (Oregon) 美國西部 (us-west-2)) 進行營運，則鑑識 OU 中會有兩個帳戶：一個用於 #### (us-east-1 另一個用於 us-west-2)。

您可以為多個區域建立鑑識 AWS 帳戶。您在將 AWS 資源複製到該帳戶時應小心，以確認是否符合資料主權要求。佈建新帳戶需要一些時間，因此必須在事故之前建立和檢測鑑識帳戶，以便回應人員能夠有效地使用這些帳戶進行回應。

下圖顯示範例帳戶結構，包括具有每個區域鑑識帳戶的鑑識 OU：



針對事故應變的每個區域帳戶結構

擷取備份和快照

設定重要系統和資料庫的備份，對於從安全性事故中復原和鑑識用途非常重要。備份就緒後，您可以將系統還原到先前的安全狀態。您可以在 AWS 上拍攝各種資源的快照。快照可為您提供那些資源的時間點備份。有許多 AWS 服務，可以在備份和復原方面為您提供支援。如需這些備份與復原之服務和方法的詳細資訊，請參閱 備份與復原規範指引 和 使用備份從安全性事故中復原。

尤其是當涉及勒索軟體等情況時，務必確保備份是否有充足的保護。如需保護備份的指引，請參閱 在 AWS 中保護備份的 10 大安全性最佳實務。除了確保備份的安全之外，您還應該定期測試備份和還原程序，以確認您現有的技術和程序是否如預期般運作。

自動化鑑識

在安全事件期間，事故應變團隊必須能夠快速收集和分析證據，同時維持事件周圍期間的準確性 (例如擷取與特定事件或資源相關的日誌，或收集 Amazon EC2 執行個體的記憶體傾印)。事故應變團隊手動收集相關證據既具挑戰性又耗時，尤其是範圍遍及大量執行個體和帳戶時。此外，手動收集可能容易出現人為錯誤。基於這些原因，您應盡可能開發和實作鑑識的自動化。

AWS 為鑑識提供許多自動化資源，內容列於以下的資源區段。這些資源是我們已開發和客戶已實作的鑑識模式範例。雖然這些範例在一開始可能是有用的參考架構，但請根據環境、需求、工具和鑑識程序，考慮是否加以修改或建立新的鑑識自動化模式。

資源

相關文件：

- AWS 安全事故應變指南 - 開發鑑識功能
- AWS 安全事故應變指南 - 鑑識資源
- AWS 雲端 中的鑑識調查環境策略
- 如何在 AWS 中自動化鑑識磁碟收集
- AWS 規範性指引 - 自動化事故應變和鑑識

相關影片：

- 自動化事故回應和鑑識

相關範例：

- 自動化事故應變與鑑識架構
- Amazon EC2 的自動化鑑識協調器

SEC10-BP04 開發和測試安全性事故應變程序手冊

準備事故應變流程的關鍵部分是制定程序手冊。事故應變程序手冊提供一系列規範性指引和安全性事件發生時應遵循的步驟。提供清晰的結構和步驟簡化了回應的複雜度並減少人為錯誤的可能性。

未建立此最佳實務時的曝險等級： 中

實作指引

應針對事故案例建立程序手冊，例如：

- 預期事故：應針對您預期的事故建立程序手冊。這包括拒絕服務 (DoS)、勒索軟體和憑證入侵等威脅。
- 已知的安全調查結果或提醒：應針對已知的安全性調查結果和警示 (例如 GuardDuty 調查結果) 建立程序手冊。您可能會收到 GuardDuty 調查結果並思考：「現在該怎麼辦？」 為了防止處理不當或忽

略 GuardDuty 調查結果，請為每個潛在的 GuardDuty 調查結果建立程序手冊。部分修復詳細資料和指引可尋自 GuardDuty 文件。值得注意的是，在預設情況下 GuardDuty 是未啟用的狀態，並且會產生費用。如需 GuardDuty 的相關詳細資訊，請參閱 附錄 A：雲端功能定義 - 可見性與提醒。

程序手冊應包含安全分析師應完成的技術步驟，以便充分調查和應對潛在的安全事故。

實作步驟

要納入程序手冊的項目包括：

- 程序手冊概觀：這份程序手冊可處理哪些風險或事故？ 程序手冊的目標是什麼？
- 先決條件：此事故案例需要哪些日誌、偵測機制和自動化工具？ 預期的通知是什麼？
- 溝通和向上呈報資訊：誰參與其中，其聯絡資訊為何？ 每個利害關係人的責任是什麼？
- 回應步驟：在事故應變的各個階段，應採取哪些戰術步驟？ 分析師應該執行哪些查詢？ 應該執行哪些程式碼以達到預期的成果？
  - 偵測：事故的偵測方式為何？
  - 分析：判斷影響範圍的方式為何？
  - 包含：隔離事故以限制範圍的方式為何？
  - 根除：將威脅從環境中移除的方式為何？
  - 復原：受影響的系統或資源重新投入生產環境的方式為何？
- 預期成果：執行查詢和程式碼後，程序手冊的預期結果是什麼？

資源

相關 Well-Architected 的最佳實務：

- SEC10-BP02 - 制定事故管理計畫

相關文件：

- 事故應變程序手冊的架構
- 制定您自己的事故應變程序手冊
- 事故應變程序手冊範例
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事件應變執行手冊

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

確認安全人員具有預先部署的適當工具，以縮短調查直至復原的時間。

未建立此最佳實務時的曝險等級： 中

實作指引

若要自動化安全回應和操作功能，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。

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復問題。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。

在安全調查期間，您需要能夠檢閱相關日誌以記錄和了解該事故的完整範圍和時間表。產生提醒也需要日誌，以指出特定關注的動作已發生。選擇、啟用、儲存和設定查詢與擷取機制和設定警示至關重要。此外，提供搜尋日誌資料之工具的有效方法是 Amazon Detective。

AWS 擁有 200 多種雲端服務和數千種特徵。我們建議您檢閱可支援並簡化事故應變策略的服務。

除了記錄之外，您還應該開發和實作 中繼資料，。標記可以幫助提供與 AWS 資源用途有關的上下文。標記也可用於自動化。

實作步驟

選取並設定日誌以進行分析和提醒

請參閱下列有關設定事故應變記錄的文件：

- 安全事故應變的記錄策略
- SEC04-BP01 設定服務和應用程式記錄

啟用安全服務以支援偵測和回應

AWS 提供原生偵測、預防性和回應式功能，而其他服務可用於建立自訂安全性解決方案的架構。如需安全事故應變最相關的服務清單，請參閱 雲端功能定義。

制定和實作標記策略

取得有關業務使用案例和圍繞 AWS 資源的相關內部利害關係人的上下文資訊可能很困難。執行此操作的一種方法是使用標籤的形式，此形式會將中繼資料指派給 AWS 資源，並包含使用者定義的鍵值組。您可以建立標籤，依目的、擁有者、環境、處理的資料類型以及您選擇的其他條件來分類資源。

擁有一致的標記策略可讓您快速找出和辨別與 AWS 資源有關的情境資訊，從而加快回應時間並盡可能減少用在組織情境的時間。標籤也可以作為啟動回應自動化的機制。如需要標記哪些內容的詳細資訊，請參閱 標記您的 AWS 資源。您需要先定義要在整個組織中實作的標籤。之後，您將實作並強制執行此標記策略。如需實作和強制執行的詳細資訊，請參閱 使用 AWS 標籤政策和服務控制政策 (SCP) 實作 AWS 資源標記策略。。

資源

相關 Well-Architected 的最佳實務：

- SEC04-BP01 設定服務和應用程式記錄
- SEC04-BP02 在標準化的位置擷取日誌、調查結果和指標

相關文件：

- 安全事故應變的記錄策略
- 事故應變雲端功能定義

相關範例：

- 使用 Amazon GuardDuty 和 Amazon Detective 進行威脅偵測與回應
- 安全中心工作坊
- 使用 Amazon Inspector 管理漏洞

SEC10-BP07 執行模擬

在組織隨著時間成長和發展時，威脅態勢也會跟著演變，因此持續審查事件應變能力是很重要的。執行模擬 (也稱為比賽日) 是可用於執行此評估的一種方法。模擬會使用真實世界的安全事件案例，這些案例旨在模擬威脅參與者的策略、技術和程序 (TTP)，並且讓組織可藉由回應這些可能發生在現實中的模擬網路事件，來運用和評估其事件應變能力。

建立此最佳實務的好處：模擬具有多種好處：

- 驗證網路整備程度和培養事件應變人員的信心。
- 測試工具和工作流程的正確性及效率。
- 根據您的事件應變計畫，精進溝通和呈報方法。
- 提供回應罕見媒介的機會。

未建立此最佳實務時的風險暴露等級：中

實作指引

主要的模擬類型有三種：

- 桌上模擬演練：桌上模擬方法是基於討論的會議，涉及各種事故應變利害關係人的角色和責任練習，並使用已建立的溝通工具和程序手冊。模擬演練促進通常可在虛擬場地、實體場地或兩者的組合於一整天內完成。桌上模擬演練以討論為主軸，因此側重於程序、人員和協作。技術在討論中是不可或缺的一部分，但事件應變工具或腳本的實際使用通常不是桌上模擬演練的一部分。
- 紫隊模擬演練：紫隊模擬演練提高了事故應變人員 (藍隊) 和模擬威脅參與者 (紅隊) 之間的協作層級。藍隊由安全營運中心 (SOC) 的成員組成，但也可以包含在實際網路事件期間涉入的其他利害關係人。紅隊由滲透測試團隊或受過攻擊性安全培訓的主要利害關係人組成。紅隊在設計場景時會與模擬演練協調員合作，使場景精確且可行。在紫隊模擬演練期間，主要重點是偵測機制、工具和支援事件應變工作的標準操作程序 (SOP)。
- 紅隊模擬演練：在紅隊模擬演練期間，攻方 (紅隊) 會進行模擬，以在預定範圍內達到某個目標或一組目標。守方 (藍隊) 不一定知道模擬演練的範圍和持續時間，這對他們應對實際事件的能力可呈現出更真實的評估。由於紅隊模擬演練可能是侵入性測試，請謹慎行事並施加控制，以確認該模擬演練不會對您的環境造成實際傷害。

考慮定期推行網路模擬。每種模擬演練類型都可以為參與者和整個組織提供特有的好處，因此您可以選擇從較不複雜的模擬類型 (例如桌上模擬演練) 開始著手，然後再進入更複雜的模擬類型 (紅隊模擬演練)。您應根據自身的安全成熟度、資源和所需的結果來選擇模擬類型。由於複雜性和成本較高，有些客戶可能不會選擇執行紅隊模擬演練。

實作步驟

無論您選擇的模擬類型為何，模擬通常會執行下列實作步驟：

1. 定義核心演練元素：定義模擬案例和模擬的目標。這兩者都應獲得領導階層的允許。
2. 找出關鍵利害關係人：模擬演練至少需要模擬演練協調員和參與者。根據情境，可能會涉及法律、通訊或主管領導階層等其他利害關係人。
3. 建立和測試情境：如果特定元素不可行，則可能需要在情境建置期間加以重新定義。預計最終的情境會成為此階段的輸出。
4. 促進模擬：模擬的類型將決定使用的促進形式 (編撰的場景對比於高度技術性的模擬場景)。協調員應使其促進策略與模擬演練目標相對應，他們應盡可能吸引所有模擬演練參與者，以提供最大的效益。
5. 撰寫事後報告 (AAR)：找出進展順利的領域、可以改進的領域，以及潛在的差距。AAR 應衡量模擬的有效性以及團隊對於模擬事件的應變能力，以便在未來的模擬追蹤進展幅度。

資源

相關文件：

- AWS 事故應變指南

相關影片：

- AWS GameDay - Security Edition

SEC10-BP08 建立從事故中學習的架構

實作 經驗教訓 的架構和根本原因分析能力，不僅有助改善事故應變能力，還有助防止事故重複發生。透過學習每個事故，您可以協助避免重複相同的錯誤、披露或錯誤設定，不僅能夠改善安全狀態，還可以盡可能縮短因可預防情況而損失的時間。

未建立此最佳實務時的曝險等級： 中

實作指引

實作 經驗教訓 是非常重要的，其可在高層級實現以下幾點：

- 什麼時候開設經驗教訓課程？
- 經驗教訓課程中包含哪些內容？
- 經驗教訓課程的進行方式？
- 這個課程的參與者以及參與方式？
- 如何找出待改善之處？
- 您將如何確保有效地追蹤和實作待改善之處？

此架構不應該針對或責怪個人，而應該專注於改善工具和流程。

實作步驟

除了前述所列的高層級結果之外，確保您提出正確問題以從流程中獲得最大價值 (即協助您找到可行改善之處的資訊) 非常重要。考慮這些問題，有助您發起經驗教訓的討論：

- 事故是什麼？
- 第一次識別事故的時間？
- 事故的識別方式？
- 哪些系統對活動發出提醒？
- 涉及哪些系統、服務和資料？
- 具體發生的事故？
- 哪些方面做得很好？
- 哪些方面做得不好？
- 哪個流程或程序失敗或未能擴展以回應事故？
- 在以下幾個領域有哪些可以改善之處：
  - 人員
    - 需要聯絡的對象實際上是否有空，並且聯絡人清單是最新的嗎？
    - 人們是否缺少有效回應和調查事故所需的培訓或能力？
    - 適當的資源是否已準備就緒且可供使用？
  - 流程
    - 是否遵循流程和程序？

  - 是否已記錄並提供這類事故的流程和程序？
  - 是否缺少必要的流程和程序？
  - 回應人員是否能夠即時存取所需的資訊以回應問題？
- 技術
  - 現有的提醒系統是否能有效地識別活動，並據以發出提醒？
  - 我們如何將偵測時間縮短 50%？
  - 是否需要改善現有提醒，或是需要針對此類事故建立新的提醒？
  - 現有的工具是否允許對事故進行有效的調查 (搜尋/分析)？
  - 可以做什麼來協助加快這類事故的識別速度？
  - 可以做什麼來協助避免這類事故再次發生？
  - 負責改善計畫的人是誰，您將如何測試是否已實作此計畫？
  - 實作和測試其他監控或預防性控制措施和流程的時間表為何？

這份清單並不詳盡，但可作為起點，幫助您識別組織和企業的需求，以及如何分析這些需求，以便最有效地從事故中學習並持續改善安全狀態。最重要的是透過將經驗教訓納入事故應變流程，文件和利害關係人期望的標準部分。

資源

相關文件：

- AWS 安全事故應變指南 - 建立從事故中學習的架構
- NCSC CAF 指南 - 經驗教訓

應用程式安全

問題

- SEC 11.如何在橫跨應用程式設計、開發和部署的整個生命週期內融入安全屬性並進行驗證？

SEC 11.如何在橫跨應用程式設計、開發和部署的整個生命週期內融入安全屬性並進行驗證？

人員培訓、使用自動化測試、了解相依性，以及驗證工具和應用程式的安全屬性，有助於減少生產工作負載中發生安全問題的機率。

最佳實務

- SEC11-BP01 應用程式安全訓練
- SEC11-BP02 自動化在整個開發和發佈生命週期的測試
- SEC11-BP03 定期進行滲透測試
- SEC11-BP04 手動程式碼檢閱
- SEC11-BP05 集中化套件和相依性的服務
- SEC11-BP06 以程式設計方式部署軟體
- SEC11-BP07 定期評估管道的安全屬性
- SEC11-BP08 打造在工作負載團隊中納入安全所有權的計劃

SEC11-BP01 應用程式安全訓練

提供可讓組織內建置人員接受安全開發和操作應用程式等常見實務的訓練。採用著重安全的開發方法，有助於減少只能在安全審查階段偵測到問題的可能性。

預期成果： 軟體的設計與建置應考慮安全層面。組織中的建置人員如果接受過從威脅模型開始的安全開發方法訓練，其所生產軟體的整體品質與安全都能獲得改善。這種方法能縮短遞送軟體或功能所花費的時間，因為其必須在安全審查階段之後重新作業的機率較低。

就這項最佳實務的目的而言，安全開發與所編寫的軟體，以及支援軟體開發生命週期 (SDLC) 的工具或系統相關。

常見的反模式：

- 一直等到安全審查階段，才開始考慮系統的安全屬性。
- 將所有的安全性決定工作全部留給安全團隊。
- 未在 SDLC 溝通如何做出與整體安全期待或組織政策相關的決定。
- 太晚參與安全審查程序。

建立此最佳實務的優勢：

- 可在開發生命週期初期更清楚了解組織對於安全的要求。
- 可以更快識別、修復安全問題，進而加快功能交付速度。
- 改善軟體和系統的品質。

未建立此最佳實務時的風險暴露等級：中

實作指引

提供組織內建置人員的訓練。一開始上威脅模型相關課程，有助於奠定安全訓練的良好基礎。理想狀況下，建置人員應該能夠自助存取與其各自工作負載相關的資訊。這種存取能協助人員做出有關建置中系統安全屬性的明智決策，而不需要詢問其他團隊。參與安全團隊進行審查的程序應該清楚定義，並能輕鬆實施。在審查程序中的步驟則應納入安全訓練當中。如果有已知的實作模式或範本，則其應可輕鬆找出，且連結至整體安全需求。考慮使用 AWS CloudFormation、 AWS Cloud Development Kit (AWS CDK) 建構模組、Service Catalog，或者其他的範本工具，以便降低自訂組態的需求。

實作步驟

- 一開始安排建置人員上威脅模型相關課程，奠定良好基礎，並有助於進行考量安全層面的訓練。
- 提供存取 AWS 培訓 和認證、產業，或 AWS 合作夥伴訓練的權限。
- 提供有關組織安全審查程序的培訓，明確劃分安全團隊、工作負載團隊和其他相關人員之間的責任分配。
- 發布關於如何達到您的安全要求的自助式指南，包含程式碼片段和範本（如有提供）。
- 定期取得建置人員團隊安全審查程序與訓練體驗方面的意見回饋，並使用該意見回饋進行改善。
- 使用演練日或錯誤修復日活動，協助減少問題數量，並提升建置人員的技能水平。

資源

相關的最佳實務：

- SEC11-BP08 打造在工作負載團隊中納入安全所有權的計劃

相關文件：

- AWS 培訓 和認證
- 如何思考雲端安全管控
- 如何建立威脅模型
- 加速訓練 – AWS 技能培養

相關影片：

- 預防性安全：考量與方法

相關範例：

- 威脅模型相關的研討會
- 開發人員的產業認知

相關服務：

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK) (AWS CDK) 建構模組
- Service Catalog
- AWS 錯誤大集合

SEC11-BP02 自動化在整個開發和發佈生命週期的測試

自動化在整個開發和發佈生命週期的安全署性測試。自動化可以讓軟體在發佈之前更容易一致且重複地找出潛在問題，因此能降低將供應軟體的安全問題風險。

預期成果： 自動化測試的目標是提供以程式設計方式，及早偵測潛在問題，而且常常是遍及整個開發生命週期。啟用自動化迴歸測試時，您可以對變更之後的軟體重新執行功能性與非功能性的測試，確認先前測試過的軟體運作仍如預期。如果定義安全單元測試來檢查常見的錯誤組態，例如損壞或缺失的驗證，您就能夠在開發過程中及早發現和修正這些問題。

測試自動化會使用專用測試個案來進行應用程式驗證，測試期間以應用程式需求和所需功能為基礎。自動化測試會將產生的測試輸出與其個別預期輸出進行比較，得到最後結果，進而加快整體的測試生命週期。包括像是迴歸測試與單位測試套組的測試方法最適合自動化應用。自動化安全屬性測試，建置人員就能接收自動化意見回饋，而不需要等待舉行安全檢閱。採用靜態或靜態程式碼分析的自動化測試，可以提高程式碼品質，並協助及早在開發生命週期中偵測出潛在的軟體問題。

常見的反模式：

- 未傳達測試個案與自動化測試的測試結果。
- 僅在即將發佈前執行自動化測試。
- 自動化有經常改變需求的測試個案。
- 無法提供如何解決安全測試結果的指引。

建立此最佳實務的優勢：

- 降低人員評估系統安全署性的依賴性。
- 可在跨多個工作串流之間找到一致結果，進而提高一致性。
- 降低造成安全問題被導入產品線上軟體的可能性。
- 因提早捕捉到軟體問題，而縮短偵測與矯正之間的範圍時段。
- 提高跨多個工作串流之系統或重複行為的能見度，其可用來促進整體組織改進。

未建立此最佳實務時的風險暴露等級：中

實作指引

隨著軟體逐漸建置，採用各種不同機制來測試軟體，確保您正根據應用程式的業務邏輯為主的功能性需求，以及著重應用程式可靠性、效能和安全性的非功能性需求，進行應用程式的測試作業。

靜態應用程式安全測試 (SAST) 分析原始程式碼是否有異常的安全模式，並提供可能存在缺陷程式碼的提示。SAST 依賴靜態輸入來測試某個範圍的已知安全問題，這些輸入包括文件 (需求規格、設計文件，以及設計規格4) 和應用程式原始程式碼。靜態程式碼分析器可協助加快大量程式碼的分析作業。NIST 品質群組則提供原始程式碼安全分析器的比較，其中包括能用於位元組程式碼掃描器和二進位程式碼掃描器的開放原始碼工具。

應用動態分析安全測試 (DAST) 方法以補充您的靜態測試，這個方法會在應用程式執行期間進行測試，找出潛在的意外行為。動態測試可用來偵測出靜態分析無法偵測出的潛在問題。在程式碼儲存、建置和管道等階段進行測試，您就可以檢查進入程式碼當中的各種不同潛在問題類型。Amazon CodeWhisperer 提供程式碼建議，包括在建置人員的 IDE 中執行安全掃描。Amazon CodeGuru Reviewer 可以找出關鍵問題、安全問題，以及在應用程式開發期間難以發現的錯誤，並提供改善程式碼品質的建議。

開發人員安全研討會使用 AWS 開發人員工具，像是 AWS CodeBuild、AWS CodeCommit 和 AWS CodePipeline，執行包括 SAST 和 DAST 測試方法的發佈管道自動化。

隨著 SDLC 繼續進行，建立包括安全團隊定期進行應用程式檢閱的反覆程序。收集自這些安全檢閱的意見回饋應加以解決，並在發佈準備度檢閱時加以驗證。這些檢閱作業會建立堅實強大的應用程式安全狀態，並提供建置人員可解決潛在問題的可行動意見回饋。

實作步驟

- 實作一致的 IDE、程式碼檢閱，以及包括安全測試的 CI/CD 工具。
- 考慮 SDLC 中的哪個位置適合封鎖管道，而不只是通知建置人員出現需要矯正的問題。

- 開發人員安全研討會提供在發佈管道中整合靜態與動態測試的範例。
- 使用自動化工具執行測試或程式碼分析，這些工具包括像是已與開發人員 IDE 完成整合的 Amazon CodeWhisperer，以及可在遞交認可時掃描程式碼的 Amazon CodeGuru Reviewer，協助建置人員在正確時間得到意見回饋。
- 使用 AWS Lambda 進行建置時，您可以使用 Amazon Inspector 來掃描多個功能的應用程式程式碼。
- AWS CI/CD 研討會提供在 AWS 上建置 CI/CD 管道的起點。
- 如果將自動化測試納入 CI/CD 管道，您應該使用票證系統來追蹤通知，以及軟體問題的矯正。
- 如果是可能會產生發現結果的安全測試，連結矯正的指引將有助於建置人員改善程式碼品質。
- 定期分析自動化工具所找到的發現結果，以便找出下次自動化、建置人員訓練或認知行銷活動的優先順序。

資源

相關文件：

- 持續交付與持續部署

- AWS DevOps 能力合作夥伴
- AWS 安全能力合作夥伴 (應用程式安全)
- 選擇 Well-Architected CI/CD 方法
- 使用 Amazon EventBridge 和 Amazon CloudWatch Events 監控 CodeCommit 事件
- Amazon CodeGuru 檢閱中的機密偵測
- 配合有效管控，加速在 AWS 的部署
- AWS 如何達到自動化安全、無人為介入的部署

相關影片：

- 無人為介入：Amazon 的自動化持續交付管道
- 自動化跨帳戶 CI/CD 管道

相關範例：

- 開發人員的產業認知

- AWS CodePipeline 管控 (GitHub)
- 開發人員安全研討會
- AWS CI/CD 研討會

SEC11-BP03 定期進行滲透測試

定期對您的軟體進行滲透測試。這項機制有助於找出無法在自動化測試或手動程式碼審查時偵測到的潛在軟體問題。同時也有助於您了解偵測控制措施的效用。滲透測試應嘗試判斷軟體是否會透過非預期的方式執行，例如暴露原本應受保護的資料，或是授與超乎預期的較廣泛權限。

預期成果： 滲透測試可用來為您的應用程式安全屬性進行偵測、修復和驗證。軟體開發生命週期 (SDLC) 期間應該進行定期與排程型滲透測試。從滲透測試找到的發現結果應事先解決，才能安排軟體發行。您應該分析從滲透測試得到的發現結果，並找出是否有任何問題可使用自動化找出。實施包括主動意見回饋機制的定期和可重複滲透測試程序，可協助建置人員得知指引，並改善軟體品質。

常見的反模式：

- 只對已知或普遍存在的安全問題進行滲透測試。
- 滲透測試應用程式 (不含相依第三方工具和程式庫)。
- 只對套件安全問題進行滲透測試，且不評估已實作的商業邏輯。

建立此最佳實務的優勢：

- 提高軟體在發行前的安全屬性信心。
- 可找出偏好應用程式模式，並藉以提高軟體品質的機會。
- 在開發生命週期初期進行的意見回饋循環流程，當中的自動化或額外訓練可以改善軟體的安全屬性。

未建立此最佳實務時的風險暴露等級：高

實作指引

滲透測試是一種結構化的安全測試練習，過程當中，您會執行計劃的安全性缺口情境，對安全控制進行偵測、修復與驗證。滲透測試從偵察活動開始，過程中會根據目前的應用程式設計與其相依性收集資料。已經建置並執行精選的安全特定測試情境清單。這些測試的主要目的在於找出您的應用程式中的安全問題，這些問題可能會被利用來非預期地存取環境，或未經授權存取資料。當您推出新功能，或是每當應用程式遭遇重大的功能變更或進行技術實作，您就應該進行滲透測試。

您應該識別開發生命週期中最適合進行滲透測試的階段。這項測試的執行時間應該盡量延到系統功能接近預定發行階段之時，而且要保留足夠修復任何問題的時間。

實作步驟

- 建立處理滲透測試範圍限制方式的結構化程序，前提是這個關於威脅模型的程序是維持內容的好方法。
- 識別開發週期中最適合進行滲透測試的時機。進行測試時應該是預期應用程式進行最少變更，而且有足夠時間進行修復。
- 訓練建置人員學會從滲透測試發現結果預期哪些內容，以及如何取得關於修復的資訊。
- 使用工具，透過自動化共通或可重複測試，加速滲透測試程序。
- 分析滲透測試發現結果來找出系統性安全問題，並使用這份資料，得知其他的自動化測試與持續進行的建置人員教育。

資源

相關的最佳實務：

- SEC11-BP01 應用程式安全訓練
- SEC11-BP02 自動化在整個開發和發佈生命週期的測試

相關文件：

- AWS 滲透測試會提供在 AWS 上進行滲透測試的詳細指引
- 配合有效管控，加速在 AWS 的部署
- AWS 安全能力合作夥伴
- 現代化您在 AWS Fargate 上的滲透測試架構
- AWS Fault injection Simulator

相關範例：

- 自動化配合 AWS CodePipeline 的 API 測試 (GitHub)
- 自動化安全協助程式 (GitHub)

SEC11-BP04 手動程式碼檢閱

對您製作的軟體進行手動程式碼檢閱。此程序有助於確認編寫程式碼的人員並非檢查程式碼品質的唯一人員。

預期成果： 在開發期間納入手動程式碼檢閱步驟可提高所編寫軟體的品質，因此有助於提升技能較差團隊成員的程度，而且有機會找出適合實施自動化的位置。手動程式碼檢閱可獲自動化工具和測試支援。

常見的反模式：

- 未在部署前先執行程式碼檢閱。
- 編寫程式碼和檢閱程式碼是相同人員。
- 未使用自動化來協助或協調程式碼檢閱。
- 建置人員在開始檢閱程式碼之前未先經過應用程式安全的訓練。

建立此最佳實務的優勢：

- 程式碼品質更高。
- 經由重複使用常用方法而使程式碼開發更具一致性。
- 減少在滲透測試與後期階段找出問題的數量。
- 團隊內部的知識轉移效能更高。

未建立此最佳實務時的風險暴露等級：中

實作指引

檢閱步驟應該是在整體程式碼管理流程中的實作部分。具體步驟依據分支、提取請求與合併所使用的不同方法而定。您可能使用 AWS CodeCommit 或第三方解決方案，例如，GitHub、GitLab 或 Bitbucket。無論使用哪種方法，您都一定要確認這些程序必須經過檢閱程式碼，才能部署至生產環境。使用 Amazon CodeGuru Reviewer 等工具可以讓協調程式碼檢閱過程變得更簡單。

實作步驟

- 在程式碼管理流程中實作手動檢閱步驟，並先執行這項檢閱之後，再繼續執行。
- 考慮 Amazon CodeGuru Reviewer 用於程式碼檢閱的管理與協助。
- 實作的核准流程必須先完成程式碼檢閱，程式碼才能進入下一個階段。

- 確認已經安排程序，可以識別將在手動程式碼檢閱期間找到，並可自動偵測出的問題。
- 採用符合您的程式碼開發實務之方法，整合手動程式碼檢閱步驟。

資源

相關的最佳實務：

- SEC11-BP02 自動化在整個開發和發佈生命週期的測試

相關文件：

- 使用 AWS CodeCommit 儲存庫中的提取請求
- 使用 AWS CodeCommit 中的核准規則範本
- 關於使用 GitHub 中的提取請求
- 使用 Amazon CodeGuru Reviewer 自動進行程式碼檢閱
- 使用 Amazon CodeGuru Reviewer CLI，自動化偵測 CI/CD 管道中的安全性漏洞與錯誤

相關影片：

- 使用 Amazon CodeGuru 持續改善程式碼品質

相關範例：

- 開發人員安全研討會

SEC11-BP05 集中化套件和相依性的服務

提供可讓建置人員團隊取得軟體套件和其他相依性的集中化服務。這樣套件就能先接受驗證，再納入編寫的軟體，並提供在您的組織中被使用的軟體分析的資料來源。

預期成果： 軟體由一組其他軟體套件，加上原先所寫程式碼共同組成。因此取用重複使用的實作功能變得很簡單，例如 JSON 剖析器或加密程式庫。依照邏輯方式集中這些套件與相依性的來源，可以為安全團隊提供先驗證過套件再提供使用的機制。這個方法也能減少由於現有套件變更或直接由建置人員團隊從網際網路納入任意套件，而引發未預期的風險問題。使用這個方法再加上手動與自動測試流程，就能提高對於開發中軟體品質的信心。

常見的反模式：

- 從網際網路的任意儲存庫中取出套件。
- 新套件未經測試就提供給建置人員。

建立此最佳實務的優勢：

- 更清楚了解哪些套件將用於建置中的軟體。
- 可以在了解過實際使用情況而需要更新套件時通知工作負載團隊。
- 降低在軟體中納入有問題套件的風險。

未建立此最佳實務時的風險暴露等級：中

實作指引

提供可讓建置人員輕鬆取得的套件和其他相依性集中化服務。集中化服務可依照邏輯方式進行集中，而非實作成單一龐大的系統。這個方法可讓您用符合建置人員需求的方式提供服務。您應該實作一種能在發生更新或新需求萌生時，快速在儲存庫新增套件的方法。AWS 服務，例如 AWS CodeArtifact 或類似的 AWS 合作夥伴解決方案就能提供發揮這種能力的方法。

實作步驟：

- 實作依照邏輯方式集中，而且各種軟體開發所在環境均可使用的儲存庫服務。
- 將儲存庫的存取作業納入 AWS 帳戶 銷售程序。
- 建置自動化測試流程，在將套件發行至儲存庫之前先進行測試。
- 維護最常使用的套件、語言，以及變更程度最高團隊的規格表。
- 提供可讓建置人員團隊自動要求新套件與提供意見回饋的機制。
- 定期掃描儲存庫中的套件，識別最近所找到問題的潛在影響。

資源

相關的最佳實務：

- SEC11-BP02 自動化在整個開發和發佈生命週期的測試

相關文件：

- 配合有效管控，加速在 AWS 的部署
- 使用 CodeArtifact 套件來源控制工具組加強您的套件安全

- 偵測使用 Amazon CodeGuru Reviewer 記錄日誌中的安全問題
- 軟體成品的供應鏈層級 (SLSA)

相關影片：

- 預防性安全：考量與方法
- 安全的 AWS 原則 (re:Invent 2017)
- 當安全性、安全和緊迫性都很重要時：Handling Log4Shell

相關範例：

- 多重區域套件發行管道 (GitHub)
- 使用 AWS CodePipeline 在 AWS CodeArtifact 上發行 Node.js 模組 (GitHub)
- AWS CDK Java CodeArtifact 管道範例 (GitHub)
- 使用 AWS CodeArtifact 分發私有.NET NuGet 套件 (GitHub)

SEC11-BP06 以程式設計方式部署軟體

盡可能以程式設計方式進行軟體部署。此方法可減少部署失敗或因人為疏失而發生非預期問題的機率。

預期成果：讓人員遠離資料是在 AWS 雲端 中安全建置的重要原則。這項原則包括軟體的部署方式。

不仰賴人員的軟體部署具備更高的可信度，因為測試結果就是部署結果，而且每次部署都會一致。軟體應該不需要變更就能在不同環境中運作。使用十二因素應用程式開發的原則時，特別是指組態外部化，可以將相同的程式碼部署到多個環境，而不需要任何變更。密碼編譯型簽署的軟體套件是用來確認環境之間未發生任何變更的好方法。這個方法的最終成果是降低變更程序中的風險，並且改善軟體發佈一致性。

常見的反模式：

- 手動部署軟體至生產環境。
- 手動執行因應不同環境需求的軟體變更。

建立此最佳實務的優勢：

- 提高軟體發佈程序的可信度。
- 降低變更失敗影響到業務功能的風險。

- 因變更風險降低而增加發佈規律。
- 部署其間意外事件的自動回復能力。
- 可以密碼編譯方式證明所測試的軟體就是實際部署的軟體。

未建立此最佳實務時的風險暴露等級：高

實作指引

建置 AWS 帳戶 結構，以便排除環境的持續人員存取，並使用 CI/CD 工具來執行部署。建立應用程式的架構，使其能從外部來源取得環境特定組態資料，例如 AWS Systems Manager 參數存放區。簽署通過測試的套件，並在部署期間驗證這些簽章。設定 CI/CD 管道，以便推送應用程式程式碼，並可使用Canary 來確認部署成功。使用像是 AWS CloudFormation 或 AWS CDK 等工具來定義基礎架構，接著使用 AWS CodeBuild 和 AWS CodePipeline 來執行 CI/CD 操作。

實作步驟

- 建置定義明確的 CI/CD 管道，以便簡化部署程序。
- 使用 AWS CodeBuild 和 AWS 程式碼管道提供 CI/CD 功能時，可以讓您輕鬆地將安全測試整合至管道中。
- 遵循使用多個帳戶整理您的 AWS 環境白皮書中的環境區隔相關指引。
- 確認已在執行生產工作負載的環境中無持續人員存取。
- 建立應用程式的架構，使其支援組態資料的外部化。
- 考慮使用藍/綠部署模型進行部署。
- 實作 Canary 來驗證軟體部署成功。
- 使用像是 AWS Signer 或 AWS Key Management Service (AWS KMS) 等密碼編譯工具來簽署與驗證將要部署的軟體套件。

資源

相關的最佳實務：

- SEC11-BP02 自動化在整個開發和發佈生命週期的測試

相關文件：

- AWS CI/CD 研討會

- 配合有效管控，加速在 AWS 的部署
- 自動化安全、無人為介入的部署
- 使用 AWS Certificate Manager Private CA 和 AWS Key Management Service 非對稱金鑰的程式碼簽署
- 程式碼簽署，AWS Lambda 的信任和完整性控制

相關影片：

- 無人為介入：自動化在 Amazon 的持續交付管道

相關範例：

- 使用 AWS Fargate 的藍/綠部署

SEC11-BP07 定期評估管道的安全屬性

採用 Well-Architected 安全原則保護您的流程，特別注意權限的區隔。定期評估管道基礎設施的安全屬性。有效管理管道的安全，就能讓您的軟體通過管道中重重的安全性考驗。

預期成果：用於建置與部署軟體的管道應該遵循針對環境中任何其他工作負載所建議的相同實務。管道中所實作的測試不應由使用測試的建置人員進行編輯。這些管道應該只具備其預計部署所需要的權限，並且應實作保護措施，防止部署至錯誤的環境。管道不應只依賴長期憑證資訊，並且應設定成能夠發出狀態資訊，以驗證建置環境的完整性。

常見的反模式：

- 安全測試可能遭建置人員避開。
- 部署管道的權限過於廣泛。
- 管道未設定進行輸入驗證。
- 未定期檢閱與 CI/CD 基礎設施相關的權限。
- 使用長期有效或硬式編碼的登入資料。

建立此最佳實務的優勢：

- 經由此類管道完成建置與部署的軟體完整性具備更高的可信度。
- 可在發現可疑活動時停止部署作業。

未建立此最佳實務時的風險暴露等級：高

實作指引

從支援 IAM 角色的受管 CI/CD 服務開始，可以減少登入資料外洩情況。套用這些安全支柱原則至您的 CI/CD 管道基礎設施，有助於您判斷哪些地方可以改善安全性。遵循 AWS 部署管道參考架構是建置 CI/CD 環境的好起點。定期檢閱管道實作及分析意外行為日誌，有助於了解用於部署軟體之管道的用量模式。

實作步驟

- 從 AWS 部署管道參考架構開始行動。
- 考慮使用 AWS IAM Access Analyzer，以程式設計方式產生管道的最低權限 IAM原則。
- 整合您的管道與監控與警示，以便您可在 AWS 受管服務 Amazon EventBridge 發生意外或異常活動時收到通知，這樣您就可以將資料路由至像是 AWS Lambda 或 Amazon Simple Notification Service (Amazon SNS) 等目的地。

資源

相關文件：

- AWS 部署管道參考架構
- 監控 AWS CodePipeline
- AWS CodePipeline 安全最佳實務

相關範例：

- DevOps 監控儀表板 (GitHub)

SEC11-BP08 打造在工作負載團隊中納入安全所有權的計劃

打造一項計劃或一種機制，賦予建置人員團隊能對其本身所建軟體做出安全決策的能力。您的安全團隊仍需在審查過程中確認這些決策，但是在建置人員團隊中納入安全所有權的做法，可以建置更快且更安全的工作負載。這項機制也能推動所有權文化，積極影響您所建置系統的運作。

預期成果：若要賦予建置人員團隊安全所有權和決策能力，您可以訓練建置人員對於安全的觀念，或者配合在建置人員團隊中納入或連結安全部門人員，增強人員的訓練。每種方法都有效用，而且可讓團隊

在開發週期前期階段就做出品質更好的安全決策。這個所有權模式是以訓練應用程式安全為基礎。從處理特定工作負載的威脅模型開始，有助於讓設計專注在適當環境內容。成立著重建置人員的安全社群，或是指派與建置人員團隊合作的安全部門工程師的另一項好處，在於您可以更深入了解軟體的編寫方式。這項了解有助於判斷出下一個可以用自動化達到改善的區域。

常見的反模式：

- 將所有的安全決策全部留給安全團隊。
- 未及早在開發程序初期解決安全需求。
- 未諮詢建置人員與安全部門人員在計劃運作方面的意見回饋。

建立此最佳實務的優勢：

- 縮短完成安全檢閱的時間。
- 減少必須在安全檢閱階段中偵測出的安全問題。
- 改善所編寫軟體的整體品質。
- 有機會找出並了解具備高度改善價值的系統性問題或區域。
- 減少因安全檢閱發現結果而必須進行的重新作業量。.
- 改善對於安全功能的感覺。

未建立此最佳實務時的風險暴露等級：低

實作指引

從 SEC11-BP01 應用程式安全訓練 的指引開始。接著找出您認為最適合組織的計劃操作模式。其中兩種主要模式分別是訓練建置人員，以及在建置人員團隊當中納入安全部門人員。在您決定初步方法之後，您應該透過單一或小組型工作負載團隊進行先行試驗，證明該模式適合您的組織。建置人員與組織的安全部門所提供的領導支援，有助於計劃達成與成功實施。隨著這項計劃不斷建置，您一定要選擇可以用來顯示計劃價值的矩陣。了解 AWS 如何解決這個問題可以學到相當多知識。這項最佳實務非常強調組織層面變更與文化。您所使用的工具應該能支援建置人員與安全社群之間的協作。

實作步驟

- 從訓練建置人員處理應用程式安全開始。
- 建立專為教育建置人員的社群和上線計劃。
- 挑選計劃名稱。守門人、擁護者或倡導者是常見手法。

- 找出要應用的模式：訓練建置人員、納入安全部門工程師，或是安排親和性安全角色。
- 從安全部門、建置人員和可能的其他相關小組當中，找出專案贊助者。
- 計劃當中所涉多人的追蹤矩陣，檢閱所花時間，以及建置人員與安全團隊人員的意見回饋。使用這些矩陣來達成改善。

資源

相關的最佳實務：

- SEC11-BP01 應用程式安全訓練
- SEC11-BP02 自動化在整個開發和發佈生命週期的測試

相關文件：

- 如何達成威脅建模
- 如何思考雲端安全管控

相關影片：

- 預防性安全：考慮與方法

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。您可以在下列白皮書中找到規範指引： 可靠性支柱白皮書。

最佳實務領域

- 基礎
- 工作負載架構
- 變更管理
- 失敗管理

基礎

問題

REL 1.如何管理 Service Quotas 和限制？

雲端型工作負載架構會有 Service Quotas (也稱為服務限制)。這些配額的用意在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

了解工作負載架構的預設配額和管理配額增加要求。知道哪些雲端資源限制 (例如，磁碟或網路) 具有潛在影響。

預期成果：客戶可實作適當的指導方針來監控關鍵指標、基礎設施審查和自動矯正步驟，確認未達到可能導致服務降級或中斷的服務配額與限制，藉以防止其 AWS 帳戶 中的服務降級或中斷。

常見的反模式：

- 在部署工作負載時，未了解軟、硬體配額及其對所用服務的限制。
- 部署替換工作負載時，未事先分析及重新設定必要的配額或聯絡支援人員。
- 假設雲端服務沒有限制，且在使用服務時無須考量費率、限制、計數和數量。
- 假設配額會自動增加。
- 不知道配額要求的程序和時間表。
- 假設每個服務在不同區域間的預設雲端服務配額都是相同的。
- 假設可以違反服務限制，且系統會將限制自動擴展或增加到資源限制以上
- 未以尖峰流量測試應用程式，以製造資源使用率的壓力。

- 佈建資源時未分析必要的資源大小。
- 選擇遠超出實際需求或預期尖峰的資源類型，而過度佈建容量。
- 未在新的客戶事件或部署新技術之前事先評估新流量層級的容量要求。

建立此最佳實務的效益：監控及自動管理服務配額和資源限制，可主動減少失敗的狀況。若未遵循最佳實務，客戶服務的流量模式變更即可能導致中斷或降級。藉由在所有區域和所有帳戶間監控並管理這些值，應用程式在遇到不良或非計劃性事件時將會有更高的彈性。

未建立此最佳實務時的風險暴露等級：高

實作指引

Service Quotas 是一項 AWS 服務，可協助您從單一位置管理超過 250 個 AWS 服務的配額。除了查閱配額值外，您也可以從 Service Quotas 主控台或使用 AWS SDK 要求和追蹤配額增長。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 (如需範例，請參閱 Amazon VPC 配額)。

某些服務限制 (例如用於調節 API 的速率限制) 可藉由設定用量計畫在 Amazon API Gateway 中設定。在其各自服務上設為組態的某些限制包括佈建 IOPS、分配的 Amazon RDS 儲存體，以及 Amazon EBS 磁碟區分配。Amazon Elastic Compute Cloud 具有專門的 Service Limits 儀表板，有助於您管理執行個體、Amazon Elastic Block Store 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。

服務配額可能隨著區域而不同，或本質上是通用的。使用達到配額的 AWS 服務，將無法作為預期的正常用法，且可能導致服務中斷或降級。例如，服務配額會限制在區域中使用的 DL Amazon EC2 數目，而該限制可能會在使用 Auto Scaling 群組 (ASG) 的流量擴展事件期間達到。

每個帳戶的服務配額均應定期受到用量評估，以確認該帳戶的適當服務限制為何。這些服務配額可作為操作上的防護機制，以防止不慎佈建超過您所需的資源。此外也可用來限制 API 操作的要求率，以保護服務免於濫用。

服務限制與服務配額不同。服務限制代表特定資源的類型為該資源定義的限制。這有可能是儲存容量 (例如，gp2 的大小限制為 1 GB - 16 TB) 或磁碟輸送量 (10,0000 iops)。制定資源類型的限制，並持續評估用量是否可能超出限制，是很重要的。若意外超出限制，帳戶的應用程式或服務可能會降級或中斷。

如果在某個使用案例中，服務配額會影響到應用程式的效能，且無法根據需求進行調整，請聯絡 AWS Support 以查看是否有緩解措施。如需關於調整固定配額的詳細資料，請參閱 REL01-BP03 透過架構適應固定服務配額和限制。

有許多 AWS 服務和工具可協助您監控及管理 Service Quotas。您應利用這些服務和工具，以提供配額層級的自動或手動檢查。

- 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。這有助於識別接近配額的服務。
- AWS Management Console 提供了相關方法，用以顯示服務配額值、管理及要求新配額、監控配額要求的狀態，以及顯示配額的歷史。
- AWS CLI 和 CDK 提供了程式化方法，可自動管理及監控服務配額層級與用量。

實作步驟

針對 Service Quotas：

- 審查 AWS Service Quotas。
- 若要得知現有的服務配額，請確認使用的服務為何 (例如 IAM Access Analyzer)。約有 250 個 AWS 服務受到服務配額控制。然後，確認每個帳戶和區域內可能使用的特定服務配額名稱。每個區域約有 3000 個服務配額名稱。
- 使用 AWS Config 擴大此配額分析，以尋找在您的 AWS 帳戶 中使用的所有 AWS 資源。
- 使用 AWS CloudFormation 資料來確認您使用的 AWS 資源。查看在 AWS Management Console 中或透過 `list-stack-resources` AWS 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。
- 透過查看部署程式碼來確定工作負載所需的所有服務。
- 決定適用的服務配額。從 Trusted Advisor 和 Service Quotas 使用可以程式設計方式存取的資訊。
- 建立自動監控方法 (請參閱 REL01-BP02 管理跨帳戶和區域的服務配額 和 REL01-BP04 監控和管理配額)，以在服務配額接近或已達限制時發出提醒和通知。
- 建立自動化和程式化方法，以檢查服務配額是否在相同帳戶中的某個區域有所變更，但在其他區域並未變更 (請參閱 REL01-BP02 管理跨帳戶和區域的服務配額 和 REL01-BP04 監控和管理配額)。
- 自動執行掃描應用程式日誌和指標，以確認是否有任何配額或服務限制錯誤。若有這類錯誤存在，請傳送提醒到監控系統。
- 建立工程程序，以在發現特定服務需要更大的配額時計算配額的必要變更 (請參閱 REL01-BP05 自動配額管理)。
- 建立佈建和核准工作流程，以要求變更服務配額。其中應包含要求遭拒絕或部分核准時的例外狀況工作流程。
- 建立工程方法以在佈建之前審查服務配額，並在推行至生產環境之前使用新的 AWS 服務。(例如，載入測試帳戶)。

針對服務限制：

- 建立監控和指標方法，針對接近資源限制的資源讀數發出提醒。適當利用 CloudWatch 進行指標或日誌監控。
- 為每個具有有效應用程式或系統限制的資源建立提醒閾值。
- 建立工作流程和基礎設施管理程序，以在限制接近使用率時變更資源類型。此工作流程應將負載測試納入作為最佳實務，以確認新類型是具有新限制的正確資源類型。
- 使用現有的程序和流程，將已識別的資源遷移至建議的新資源類型。

資源

相關的最佳實務：

- 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
- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性要素：可用性
- AWS Service 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？
- 如何要求增加配額

- 服務端點和配額
- Service Quotas 使用者指南
- AWS 的配額監視器
- AWS 故障隔離界限
- 可用性與備援性
- AWS for Data
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- 在 AWS 上管理每租用戶一帳戶 SaaS 環境中的帳戶生命週期
- 管理和監控工作負載中的 API 調節
- 使用 AWS Organizations 大規模檢視 AWS Trusted Advisor 建議
- 使用 AWS Control Tower 自動執行服務限制增加和企業支援

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas
- 使用 Service Quotas 檢視和管理 AWS 服務的配額
- AWS IAM 配額示範

相關工具：

- Amazon CodeGuru Reviewer
- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOps Guru
- AWS Config
- AWS Trusted Advisor
- AWS CDK

- AWS Systems Manager
- AWS Marketplace

REL01-BP02 管理跨帳戶和區域的服務配額

如果您使用多個帳戶或區域，請在生產工作負載執行的所有環境中都要求合適的配額。

預期成果：在跨帳戶或區域的組態，或有彈性設計使用了區域或帳戶容錯移轉的組態中，服務和應用程式應該不受服務配額用盡的影響。

常見的反模式：

- 允許一個隔離區域內的資源用量增長，但無維持其他隔離區域中容量的機制。
- 在隔離區域中單獨手動設定所有配額。
- 未考量彈性架構 (例如主動或被動) 日後在非主要區域降級期間對配額需求產生的影響。
- 未定期評估配額，並在工作負載執行所在的每個區域和帳戶中進行必要的變更。
- 未利用配額要求範本在多個區域和帳戶間要求增加配額。
- 因誤認為增加配額會產生成本上的影響 (例如運算保留要求) 而未更新服務配額。

建立此最佳實務的效益：確認在區域服務無法使用時，您可以在次要區域或帳戶中處理目前的負載。這有助於降低區域中斷期間發生的錯誤數量或降級程度。

未建立此最佳實務時的風險暴露等級：高

實作指引

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 特有的。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。要維持高水準的彈性，必須持續評估服務配額 (無論自動還是手動)。

由於實作使用主動/主動、主動/被動 – 熱、主動/被動 - 冷和主動/被動 - 指示燈等方法的設計，產生了更多跨區域的工作負載，請務必了解所有區域和帳戶的配額層級。過去的流量模式不一定可明確指出服務配額是否正確設定。

同樣重要的是，每個區域的服務配額名稱限制不一定相同。在某個區域中，該值可能是五，而另一個區域中的值可能是十。這些配額的管理必須跨所有的相同服務、帳戶和區域，以在負載下提供一致的彈性。

在不同區域 (主動區域或被動區域) 間協調所有服務配額差異，並建立持續協調這類差異的程序。被動區域容錯移轉的測試計劃鮮少擴展至尖峰主動容量，意即演練日或桌面演練可能找不到區域之間的服務配額差異，因而無法維持正確的限制。

服務配額漂移，這是指某個指定配額的服務配額限制在某個區域中已變更，但未在所有區域變更的情況，對於追蹤和評估而言非常重要。您應考慮在具有流量甚或雲端承載流量的區域中變更配額。

- 根據您的服務要求、延遲、法規和災難復原 (DR) 要求，選取相關的帳戶和區域。
- 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。您應比較這些值的差異。

實作步驟

- 審查可能超出使用風險等級的 Service Quotas 值。超出 80% 和 90% 閾值時，AWS Trusted Advisor 會提供提醒。
- 審查任何被動區域 (主動/被動設計中) 的服務配額值。確認在主要區域失敗時，負載將可在次要區域中成功執行。
- 自動評估相同帳戶中的區域之間是否發生了任何服務配額漂移，並採取因應措施以變更限制。
- 如果客戶的組織單位 (OU) 是以支援的方式建構的，則應更新服務配額範本，以反映應套用至多個區域和帳戶的任何配額中的變更。
  - 建立範本，並將區域關聯至配額變更。
  - 審查所有現有的服務配額範本，確認是否有任何必要的變更 (區域、限制和帳戶)。

資源

相關的最佳實務：

- REL01-BP01 了解服務配額和限制
- 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
- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置

- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性要素：可用性
- AWS Service 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？
- 如何要求增加配額
- 服務端點和配額
- Service Quotas 使用者指南
- AWS 的配額監視器
- AWS 故障隔離界限
- 可用性與備援性
- AWS for Data
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- 在 AWS 上管理每租用戶一帳戶 SaaS 環境中的帳戶生命週期
- 管理和監控工作負載中的 API 調節
- 使用 AWS Organizations 大規模檢視 AWS Trusted Advisor 建議
- 使用 AWS Control Tower 自動執行服務限制增加和企業支援

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas
- 使用 Service Quotas 檢視和管理 AWS 服務的配額

- AWS IAM 配額示範

相關服務：

- Amazon CodeGuru Reviewer
- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOps Guru
- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

REL01-BP03 透過架構適應固定服務配額和限制

請注意不可變更的服務配額、服務限制和實際資源限制。設計應用程式和服務的架構以防止這些限制影響可靠性。

範例包括 API 閘道的網路頻寬、無伺服器函數叫用承載大小、調節爆量速率，以及資料庫的使用者同時連線數目。

預期成果：應用程式或服務會在正常或高流量條件之下如預期般執行。它們已設計為在該資源的固定限制或服務配額內運作。

常見的反模式：

- 選擇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。
- 執行不切實際的基準並且在測試期間達到服務固定配額。例如，以爆量限制執行測試，但是進行擴充的時間量。
- 選擇若超過固定服務配額時無法擴展或修改的設計。例如，SQS 承載大小為 256KB。
- 未設計可觀測性並且實作以監控和提醒在高流量活動期間可能有風險之服務配額的臨界值

建立此最佳實務的優勢：確認應用程式會在所有預估服務載入層級之下運作，沒有中斷或降級。

未建立此最佳實務時的風險暴露等級：中

實作指引

不同於以較高容量單位取代的軟性服務配額，AWS 服務的固定配額無法變更。這表示這裡所有類型的 AWS 服務都必須在使用於應用程式設計中時評估其潛在硬性容量限制。

硬性限制會顯示在 Service Quotas 主控台中。如果欄顯示 ### = #服務有硬式限制。硬性限制也會顯示在一些資源組態頁面中。例如，Lambda 有無法調整的特定硬性限制。

例如，設計 Python 應用程式在 Lambda 函數中執行時，應用程式應該評估以判斷 Lambda 是否有機會執行超過 15 分鐘。如果程式碼可能執行超過此服務配額限制，則必須考慮替代技術或設計。如果在生產部署後達到此限制，應用程式會遭受降級和中斷直到可以矯正為止。與軟性配額不同，沒有任何方法可以變更這些限制，即使是在緊急嚴重性 1 活動下。

一旦應用程式部署到測試環境，應該使用策略來尋找是否達到任何硬性限制。壓力測試、負載測試和混亂測試應該是引入測試計劃的一部分。

實作步驟

- 檢閱可用於應用程式設計階段的 AWS 服務完整清單。
- 檢閱這些服務的軟性配額限制和硬性配額限制。並非所有限制都會顯示在 Service Quotas 主控台中。一些服務在替代位置中說明這些限制。
- 隨著您設計您的應用程式，檢閱您的工作負載的業務和技術驅動來源，例如業務成果、使用案例、相依系統、可用性目標和災難復原物件。讓您的業務和技術驅動來源引導程序以識別適合您的工作負載的分散式系統。
- 分析區域和帳戶之間的服務負載。許多硬性限制對於服務是區域型的。不過，某些限制是帳戶型。
- 分析區域 (Zonal) 失敗和區域 (Regional) 失敗期間資源用量的彈性架構。在使用主動/主動、主動/被動 – 熱、主動/被動 - 冷和主動/被動 - 指示燈方法的多區預設定進度中，這些失敗案例會導致較高的用量。這會建立達到硬性限制的潛在使用案例。

資源

相關的最佳實務：

- 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-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
- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性要素：可用性
- AWS Service 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？
- 如何要求增加配額
- 服務端點和配額
- Service Quotas 使用者指南
- AWS 的配額監視器
- AWS 故障隔離界限
- 可用性與備援性
- AWS for Data
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- 在 AWS 上管理每租用戶一帳戶 SaaS 環境中的帳戶生命週期
- 管理和監控工作負載中的 API 調節

- 使用 AWS Organizations 大規模檢視 AWS Trusted Advisor 建議
- 使用 AWS Control Tower 自動執行服務限制增加和企業支援
- Service Quotas 的動作、資源和條件金鑰

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas
- 使用 Service Quotas 檢視和管理 AWS 服務的配額
- AWS IAM 配額示範
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權

相關工具：

- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOps Guru
- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

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。

預期成果：已部署管理和監控的主動和自動化系統。這些操作解決方案可確保幾乎達到配額用量臨界值。這些會由請求配額變更主動矯正。

常見的反模式：

- 未設定監控來檢查服務配額臨界值

- 未設定監控硬性限制，即使這些值無法變更。
- 假設請求和保護軟性配額變更所需的時間量是立即或短期間。
- 設定了正在接近服務配額的警示，但無如何回應提醒的程序。
- 只設定 AWS Service Quotas 支援的服務警示，但未監控其他 AWS 服務。
- 未考慮多個區域彈性設計的配額管理，例如主動/主動、主動/被動 – 熱、主動/被動 - 冷和主動/被動 - 指示燈方法。
- 未評估區域之間的配額差異。
- 未評估每個區域特定配額增加請求的需求。
- 未利用多區域配額管理的範本。

建立此最佳實務的優勢：自動追蹤 AWS Service Quotas 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協助限制由於配額耗盡造成的任何降級。

未建立此最佳實務時的風險暴露等級：中

實作指引

針對支援的服務，您可以藉由設定可評估然後傳送提醒或警示的各種不同服務，來監控您的配額。這可協助監控用量並且可以在您接近配額時提醒您。這些警示可以從 AWS Config、Lambda 函數、Amazon CloudWatch 或從 AWS Trusted Advisor 觸發。您也可以使用 CloudWatch 日誌上的指標篩選條件，搜尋與擷取日誌中的模式，以判斷用量是否正在接近配額臨界值。

實作步驟

針對監控：

- 擷取當前資源消耗 (例如，儲存貯體或執行個體)。使用服務 API 操作，例如 Amazon EC2 `DescribeInstances` API，用以收集目前的資源消耗。
- 使用以下項目，擷取您目前基本且適用於服務的配額：
  - AWS Service Quotas
  - AWS Trusted Advisor
  - AWS 文件
  - AWS 服務特定頁面
  - AWS Command Line Interface (AWS CLI)
  - AWS Cloud Development Kit (AWS CDK)

- 使用 AWS Service Quotas，這是一項 AWS 服務，有助於您從單一位置管理超過 250 種 AWS 服務的配額。
- 使用 Trusted Advisor 服務限制以不同臨界值來監控您目前的服務限制。
- 使用服務配額歷史 (主控台或 AWS CLI) 來檢查區域增加。
- 比較每個區域和每個帳戶中的服務配額變更，視需要建立等值。

針對管理：

- 自動化：設定 AWS Config 自訂規則來掃描區域之間的服務配額，並且比較是否有差異。
- 自動化：設定排定 Lambda 函數來掃描區域之間的服務配額，並且比較是否有差異。
- 手動：透過 AWS CLI、API 或 AWS 主控台掃描服務配額，掃描區域之間的服務配額，並且比較是否有差異。報告差異。
- 如果區域之間識別出配額的差異，請視需要請求配額變更。
- 檢閱所有請求的結果。

資源

相關的最佳實務：

- 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-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
- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性要素：可用性

- AWS Service 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？
- 如何要求增加配額
- 服務端點和配額
- Service Quotas 使用者指南
- AWS 的配額監視器
- AWS 故障隔離界限
- 可用性與備援性
- AWS for Data
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- 在 AWS 上管理每租用戶一帳戶 SaaS 環境中的帳戶生命週期
- 管理和監控工作負載中的 API 調節
- 使用 AWS Organizations 大規模檢視 AWS Trusted Advisor 建議
- 使用 AWS Control Tower 自動執行服務限制增加和企業支援
- Service Quotas 的動作、資源和條件金鑰

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas
- 使用 Service Quotas 檢視和管理 AWS 服務的配額
- AWS IAM 配額示範
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權

相關工具：

- AWS CodeDeploy

- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOps Guru
- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

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

資源失敗或無法存取時，在該資源成功終止之前，可能仍會被計入配額。確認您的配額涵蓋失敗或無法存取資源及其替換項目的重疊。計算此差距時，您應該考慮使用像是網路失敗、可用網路失敗或區域失敗的使用案例。

預期成果：資源或資源可存取性中的小型或大型失敗可以涵蓋在目前的服務臨界值內。已在資源規劃中考慮區域 (Zone) 失敗、網路失敗或甚至是區域 (Regional) 失敗。

常見的反模式：

- 根據目前的需求設定服務配額，而不考慮容錯移轉案例。
- 計算服務的尖峰配額時，未考慮靜態穩定性的主體。
- 計算每個區域所需的配額總計時，未考慮可能有無法存取的資源。
- 未針對某些服務及其潛在異常用量模式考慮 AWS 服務故障隔離界限。

建立此最佳實務的優勢：服務中斷事件影響應用程式可用性時，雲端可讓您實作策略來緩解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或無法存取的資源。您的配額策略適用於這些容錯移轉條件，不會由於服務限制耗盡而導致額外降級。

未建立此最佳實務時的風險暴露等級：中

實作指引

評估配額限制時，請考慮由於某些降級而可能發生的容錯移轉案例。應該考慮下列類型的容錯移轉案例：

- 中斷或無法存取的 VPC。
- 無法存取的子網路。
- 可用區域的降級程度已足夠影響許多資源的可存取性。
- 各個網路路由或輸入和輸出點遭到封鎖或變更。
- 區域的降級程度已足夠影響許多資源的可存取性。
- 有多個資源，但是並非所有資源都受到區域或可用區域中的失敗影響。

如上所列的失敗會觸發以啟動容錯移轉事件。對每個情境和客戶進行容錯移轉的決策都是唯一的，因為業務影響差距甚大。不過，在操作方面決定容錯移轉應用程式或服務時，容錯移轉位置中資源的容量規劃及其相關配額都必須在事件之前解決。

檢閱每個服務的服務配額，考慮高於可能發生的正常尖峰。由於網路或許可，這些尖峰可能與可以連線的資源相關，但是仍然是作用中。未終止的作用中資源仍然會計入服務配額限制。

實作步驟

- 確認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若遺失可存取性。
- 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%)。
- 適當時包含靜態穩定性的計算 (區域 (Zonal) 和區域 (Regional))。
- 為使用量增長制定計畫 (例如，監控使用量趨勢)。
- 考慮您最關鍵工作負載的靜態穩定性影響。評估符合所有區域和可用區域中靜態穩定系統的資源。
- 考慮使用隨需容量保留，在任何容錯移轉之前排程容量。這在最關鍵業務排程期間是有用的策略，降低在容錯移轉期間取得正確數量和資源類型的潛在風險。

資源

相關的最佳實務：

- 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
- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性要素：可用性
- AWS Service 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？
- 如何要求增加配額
- 服務端點和配額
- Service Quotas 使用者指南
- AWS 的配額監視器
- AWS 故障隔離界限
- 可用性與備援性
- AWS for Data
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可以幫助進行組態管理的合作夥伴
- 在 AWS 上管理每租用戶一帳戶 SaaS 環境中的帳戶生命週期
- 管理和監控工作負載中的 API 調節
- 使用 AWS Organizations 大規模檢視 AWS Trusted Advisor 建議
- 使用 AWS Control Tower 自動執行服務限制增加和企業支援
- Service Quotas 的動作、資源和條件金鑰

相關影片：

- AWS Live re:Inforce 2019 - Service Quotas
- 使用 Service Quotas 檢視和管理 AWS 服務的配額
- AWS IAM 配額示範
- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權

相關工具：

- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch

- Amazon EventBridge
- Amazon DevOps Guru
- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

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 在雲端和內部部署環境中的私人網路之間佈建備援連線
- REL02-BP03 確保 IP 子網路分配帳戶具有擴展性和可用性
- REL02-BP04 偏好軸輻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 位址範圍

REL02-BP01 針對工作負載公有端點使用高可用性網路連線

建置與您的工作負載公有端點的高度可用網路連線，可協助您減少由於遺失連線的停機時間，並且改善您的工作負載的可用性和 SLA。為達成此目的，請使用高度可用的 DNS、內容交付網路 (CDN)、API 閘道、負載平衡或反向代理。

預期成果：為您的公有端點規劃、建置和操作高度可用網路連線相當重要。如果您的工作負載由於遺失連線而無法連線，即使您的工作負載正在執行且可用，您的客戶還是會看到您的系統是停機。藉由結合工作負載公有端點高度可用和具彈性的網路連線與工作負載本身的彈性架構，為您的客戶提供可行的最佳可用性和服務水準。

AWS Global Accelerator、Amazon CloudFront、Amazon API Gateway、AWS Lambda 函數 URL、AWS AppSync API 和 Elastic Load Balancing (ELB) 都提供高度可用公有端點。Amazon Route 53 針對網域名稱解析提供高度可用 DNS 服務，確認可以解析您的公有端點地址。

您也可以評估用於負載平衡和代理的 AWS Marketplace 軟體設備。

常見的反模式：

- 設計高度可用的工作負載，而未規劃 DNS 和網路連線以取得高可用性。
- 在個別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 IP 地址，而非網域名稱來定位服務。
- 未測試您的公有端點已遺失連線的情境。
- 未分析網路輸送量需求和分發模式。
- 未測試和規劃您的工作負載公有端點的網際網路網路連線可能遭到中斷的情境。
- 提供內容 (例如網頁、靜態資產或媒體檔案) 到大型地理區域，而不使用內容交付網路。
- 未針對分散式阻斷服務 (DDoS) 攻擊加以規劃。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。

建立此最佳實務的優勢：設計高度可用且具彈性的網路連線，可確保您的工作負載可存取並且可供您的使用者使用。

未建立此最佳實務時的風險暴露等級：高

實作指引

建置與您的公有端點的高度可用網路連線的核心是流量的路由。若要確認您的流量可以連線到端點，DNS 必須能夠將網域名稱解析為它們的對應 IP 地址。使用高度可用和可擴展網域名稱系統 (DNS)，例如 Amazon Route 53，來管理您的網域的 DNS 記錄。您也可以使用 Amazon Route 53 提供的運作狀態檢查。運作狀態檢查會確認您的應用程式可連線、可用並且可運作，它們可以透過模仿您的使用者行為的方式進行設定，例如請求網頁或特定 URL。發生失敗時，Amazon Route 53 會回應 DNS 解析請求，並且僅將流量導向到健康的端點。您也可以考慮使用 Amazon Route 53 提供的 Geo DNS 和以延遲為基礎的路由功能。

若要確認您的工作負載本身是高度可用，請使用 Elastic Load Balancing (ELB)。Amazon Route 53 可以用來將流量目標設定為 ELB，這會將流量分發到目標運算執行個體。您也可以使用 Amazon API Gateway 以及 AWS Lambda 做為無伺服器解決方案。客戶也可以在多個 AWS 區域 中執行工作負載。使用多站點主動/主動模式，工作負載可以為來自多個區域的流量提供服務。使用多站點主動/被動模式，工作負載可以為來自主動區域的流量提供服務，而資料會複寫到次要區域並且在主要區域失敗的事件中變成作用中。Route 53 運作狀態檢查接著可以用來控制 DNS 備援從主要區域中的任何端點容錯移轉到次要區域中的端點，確認您的工作負載可連線且可供您的使用者使用。

Amazon CloudFront 藉由使用全世界邊緣節點的網路為請求提供服務，以低延遲和高資料傳輸率提供簡易 API 來分發內容。內容交付網路 (CDN) 藉由在靠近使用者的位置提供放置或快取的內容來服務客戶。這也會改善您的應用程式的可用性，因為內容的負載從您的伺服器轉移到 CloudFront 的邊緣節點。邊緣節點和區域邊緣快取會將您的內容快取複本保存在靠近您觀眾的位置，以便快速擷取並且增加您的工作負載的連線能力和可用性。

針對具有分散各地使用者的工作負載，AWS Global Accelerator 可協助您改善應用程式的可用性和效能。AWS Global Accelerator 提供任播靜態 IP 地址，可做為一或多個 AWS 區域 中託管之應用程式的固定進入點。這可讓流量盡可能輸入到使用者附近的 AWS 全球網路，改善您的工作負載的連線能力和可用性。AWS Global Accelerator 也會使用 TCP、HTTP 和 HTTPS 運作狀態檢查來監控您的應用程式端點的運作狀態。您的端點的運作狀態或組態的任何變更都會觸發將使用者流量重新導向到健康的端點，為您的使用者交付最佳效能和可用性。此外，AWS Global Accelerator 有故障隔離設計，使用由獨立網路區域提供服務的兩個靜態 IPv4 地址，增加您的應用程式的可用性。

為了協助客戶防範 DDoS 攻擊，AWS 提供 AWS Shield Standard。Shield Standard 會自動啟用並且防禦常見基礎設施 (第 3 層和第 4 層) 攻擊，例如 SYN/UDP 泛洪和反射攻擊，在 AWS 上支援您的應用程式的高可用性。針對更複雜和更大型攻擊 (例如 UDP 泛洪)、狀態耗盡攻擊 (例如 TCP SYN 泛洪) 的額外保護，並且協助保護您的應用程式在 Amazon Elastic Compute Cloud (Amazon EC2)、Elastic Load Balancing (ELB)、Amazon CloudFront、AWS Global Accelerator 和 Route 53 上執行，您可以考慮使用 AWS Shield Advanced。針對應用程式層攻擊的保護，例如 HTTP POST 或 GET 泛洪，請使用 AWS WAF。AWS WAF 可以使用 IP 地址、HTTP 標題、HTTP 本文、URI 字串、SQL 隱碼攻擊和跨網站指令碼條件來判斷應該封鎖或允許請求。

實作步驟

1. 設定高度可用 DNS：Amazon Route 53 是可度可用且可擴展網域名稱系統 (DNS) Web 服務。Route 53 會將使用者請求連線到 AWS 上或內部部署執行的網際網路應用程式。如需詳細資訊，請參閱將 Amazon Route 53 設定為您的 DNS 服務。
2. 設定運作狀態檢查：當使用 Route 53 時，請確認只有運作狀態良好的目標是可解析的。從建立 Route 53 運作狀態檢查和設定 DNS 備援開始。以下是設定運作狀態檢查時要考慮的重要層面：
   a. Amazon Route 53 如何判斷運作狀態檢查是運作狀態良好
   b. 建立、更新和刪除運作狀態檢查
   c. 監控運作狀態檢查狀態和取得通知
   d. Amazon Route 53 DNS 的最佳實務
3. 將您的 DNS 服務連線到您的端點。
   a. 當使用 Elastic Load Balancing 做為您流量的目標時，使用指向您負載平衡器區域端點的Amazon Route 53 建立別名記錄。建立別名記錄期間，將 [評估目標運作狀態] 選項設定為 [是]。

b. 針對使用 API Gateway 時的無伺服器工作負載或私有 API，使用 Route 53 將流量指向 API Gateway。

4. 決定內容交付網路。

a. 針對使用更靠近使用者的邊緣節點交付內容，從了解 CloudFront 如何交付內容開始。

b. 從簡易 CloudFront 分發開始。CloudFront 接著會知道您想要從哪裡交付內容，以及如何追蹤和管理內容交付的詳細資料。以下是設定 CloudFront 分發時要了解和考慮的重要層面：

i. 快取如何與 CloudFront 邊緣節點搭配運作

ii. 增加直接從 CloudFront 快取 (快取命中率) 提供服務的請求比例

iii. 使用 Amazon CloudFront Origin Shield

iv. 使用 CloudFront 來源容錯移轉最佳化高可用性

5. 設定應用程式層保護：AWS WAF 可協助您保護免受常見 Web 漏洞和機器人的攻擊，這些攻擊會影響可用性、危及安全性或導致消耗過多資源。若要更深入了解，請參閱 AWS WAF 如何運作，當您準備好實作應用程式層 HTTP POST AND GET 泛洪的保護，請參閱開始使用 AWS WAF。您也可以搭配使用 AWS WAF 與 CloudFront，請參閱 AWS WAF 如何使用 Amazon CloudFront 功能上的文件。

6. 設定額外 DDoS 保護：根據預設，所有 AWS 客戶都能透過 AWS Shield Standard 獲得以您的網站或應用程式為目標，對於常見、最常發生網路和傳輸層 DDoS 攻擊的保護，不需額外費用。針對在 Amazon EC2、Elastic Load Balancing、Amazon CloudFront、AWS Global Accelerator 和 Amazon Route 53 上執行，面向網際網路應用程式的額外保護，您可以考慮 AWS Shield Advanced 和檢閱 DDoS 彈性架構的範例。若要保護您的工作負載和公有端點免於 DDoS 攻擊，請參閱開始使用 AWS Shield Advanced。

資源

相關的最佳實務：

- 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
- REL11-BP04 復原期間需使用資料平面，而非控制平面
- REL11-BP06 當事件影響可用性時傳送通知

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
- 什麼是 AWS Global Accelerator？
- 什麼是 Amazon CloudFront？
- 什麼是 Amazon Route 53？
- 什麼是 Elastic Load Balancing？
- 網路連線功能 - 建立您的雲端基礎
- 什麼是 Amazon API Gateway？
- 什麼是 AWS WAF、AWS Shield 和 AWS Firewall Manager？
- 什麼是 Amazon Route 53 應用程式復原控制器？
- 為 DNS 備援設定自訂運作狀態檢查

相關影片：

- AWS re:Invent 2022 - 使用 AWS Global Accelerator 改善效能與可用性
- AWS re:Invent 2020：使用 Amazon Route 53 進行全球流量管理
- AWS re:Invent 2022 - 操作高可用性多可用區域應用程式
- AWS re:Invent 2022 - 深入了解 AWS 網路基礎設施
- AWS re:Invent 2022 - 建置彈性網路

相關範例：

- 使用 Amazon Route 53 應用程式復原控制器 (ARC) 進行災難復原
- 可靠性研討會
- AWS Global Accelerator 研討會

REL02-BP02 在雲端和內部部署環境中的私人網路之間佈建備援連線

在雲端和內部部署環境中的私人網路之間執行備援連線，以強化連線恢復能力。這目標可以藉由部署兩個或多個連結和流量路徑來實現，並可在網路故障時保持連線通暢。

常見的反模式：

- 您只依賴一種網路連線，這麼做會產生單一故障點。

- 您只使用一個 VPN 通道，或多個以相同可用區域結束的通道。
- 您依賴一個 ISP 進行 VPN 連線，如此可能導致 ISP 中斷期間全面性故障。
- 未實作 BGP 這類的動態路由通訊協定，而該協定對於網路中斷期間重新路由流量卻極為重要。
- 您忽略 VPN 通道的頻寬限制，且高估了該通道的備份功能。

建立此最佳實務的好處：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備援連線，即可確保兩個環境之間的相依服務能夠可靠地進行通訊。

未建立此最佳實務時的風險暴露等級：高

實作指引

使用 AWS Direct Connect 將內部部署網路連線至 AWS 時，您可以使用在一個以上的內部部署位置，以及在一個以上的 AWS Direct Connect 位置中的不同裝置結束的獨立連線，以藉此達到最大的網路恢復能力 (SLA 99.99%)。此拓樸提供了針對裝置故障、連線問題和完整位置中斷的恢復能力。或者，您也可以透過使用兩個個別連線到多個位置 (每個內部部署位置都連接單一 Direct Connect 位置) 獲得強大的恢復能力 (SLA 99.9%)。這種方法可防止由光纖切斷或裝置故障引起的連線中斷，並有助於減輕全面性的位置故障。AWS Direct Connect 恢復能力工具組可協助您設計 AWS Direct Connect 拓樸。

您也可以考慮將 AWS Site-to-Site VPN 在 AWS Transit Gateway 結束當成主要 AWS Direct Connect 連線的具成本效益備份方式。此設定可跨多個 VPN 通道啟用同等成本多重路徑 (ECMP) 路由，即使每個 VPN 通道的上限為 1.25 Gbps，也能達到最高 50Gbps 的輸送量。然而，請務必留意 AWS Direct Connect 仍然是最有效的選擇，因為它可以將網絡中斷機率降到最低，並提供穩定的連線能力。

透過網際網路使用 VPN 將雲端環境連接到內部部署資料中心時，請將兩個 VPN 通道設定為單一站點對站點 VPN 連線的一部分。每個通道應在不同的可用區域終止，如此才能獲得高可用性，並使用備援硬體以防止內部部署裝置故障。此外，請考慮從內部部署位置的不同網際網路服務供應商 (ISP) 提供多個網際網路連線，以避免因單一 ISP 中斷而導致 VPN 連線完全中斷。選擇具有多樣性路由和基礎設施的 ISP，尤其是具有獨立實體路徑到 AWS 端點的 ISP，這樣即能獲得高連線可用性。

除了具有多個 AWS Direct Connect 連線和多個 VPN 通道 (或兩者的組合) 的實體備援外，實作邊界閘道協定 (BGP) 動態路由也很重要。動態 BGP 會根據即時網路條件和設定的政策，從一個路徑的流量自動重新路由到另一個路徑。在發生連結或網路故障事件時，這種動態行為對於維持網路可用性和服務連續性特別有幫助。此動態行為能快速選擇替代路徑，提高網路的恢復能力和可靠性。

實作步驟

- 在 AWS 和您的內部部署環境之間，獲得高度可用的連線。

  - 在單獨部署的私有網路之間使用多個 AWS Direct Connect 連線或 VPN 通道。
  - 使用多個 AWS 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立備援。
- 在可能的情況下使用 AWS Transit Gateway 終止您的 VPN 連線。
- 評估 AWS Marketplace 設備以結束 VPN，或將您的 SD-WAN 擴展至 AWS。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。
- 在您的內部部署環境提供備援連線。
  - 您可能需要與多個 AWS 區域 進行備援連線才能滿足可用性需求。
  - 使用AWS Direct Connect 恢復能力工具組以開始使用。

資源

相關文件：

- AWS Direct Connect 恢復能力建議
- 使用備援 Site-to-Site VPN 連線以提供容錯移轉
- 路由政策和 BGP 社群
- AWS Direct Connect 中的主動/主動和主動/被動組態
- 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
- 建置可擴展且安全的多 VPC AWS 網路基礎設施
- 使用備援 Site-to-Site VPN 連線以提供容錯移轉
- 使用 AWS Direct Connect 恢復能力工具組以開始使用
- 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 Site-to-Site VPN？
- 使用 Direct Connect 閘道

相關影片：

- AWS re:Invent 2018：進階 VPC 設計和 Amazon VPC 的新功能
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構

REL02-BP03 確保 IP 子網路分配帳戶具有擴展性和可用性

Amazon VPC IP 位址範圍必須足夠大，以適應工作負載的要求，包括考慮將來擴展 IP 位址以及跨可用區域將 IP 位址分配給子網路。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 位址空間。
- 在 VPC 內，允許多個子網路的空間，讓您可以跨越多個可用區域。
- 在 VPC 內留下未用 CIDR 區塊空間，以供未來擴展。
- 確保有 IP 位址空間可以滿足您可能會用到之 Amazon 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。Kubernetes 叢集 (例如 Amazon Elastic Kubernetes Service (Amazon EKS)) 應該考慮類似條件，因為每個 Kubernetes Pod 都會依照預設獲得 VPC CIDR 區塊指派的可路由位址。
- 請注意，在各子網路 CIDR 區塊中，前四個 IP 位址和最後一個 IP 位址均預留起來，無法供您使用。
- 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。
- 最大可能的 VPC CIDR 區塊是 /16，最小的是 /28。
- 考慮其他連線的網路 (VPC、內部部署或其他雲端供應商)，並確保 IP 位址空間未重疊。如需詳細資訊，請參閱 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 位址範圍。

預期成果： 您可以使用可擴展的 IP 子網路來適應未來成長趨勢，避免不必要的浪費。

常見的反模式：

- 無法考慮未來成長，導致 CIDR 區塊太小且需要重新配置，最後可能導致停機。
- 錯誤預估 Elastic Load Balancer 可以使用的 IP 位址數量。
- 部署過多高流量負載平衡器於相同的子網路中。
- 使用自動擴展機制，同時無法監控 IP 位址取用。
- 定義的 CIDR 範圍過大，並遠超出未來成長預期，可能導致難以與其他位址範圍發生重疊的網路進行對等互連。

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。

未建立此最佳實務時的曝險等級： 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
  - 製作包含大小適當 CIDR 區塊的 VPC，以適應目前和未來的需求。
    - 如果成長預測不明，您可能希望發生錯誤的是 CIDR 區塊過大，減少未來要重新配置的情況
  - 考慮使用 IPv6 定址 做為雙堆疊 VPC 當中的子網路。非常適合使用 IPv6 的私有子網路中可能包含暫時性執行個體的機群或另外需要大量 IPv4 位址的容器。

資源

相關 Well-Architected 的最佳實務：

- 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 位址範圍

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？

- AWS 上的 IPv6
- 參考架構上的 IPv6
- Amazon Elastic Kubernetes Service 會啟動 IPv6 支援

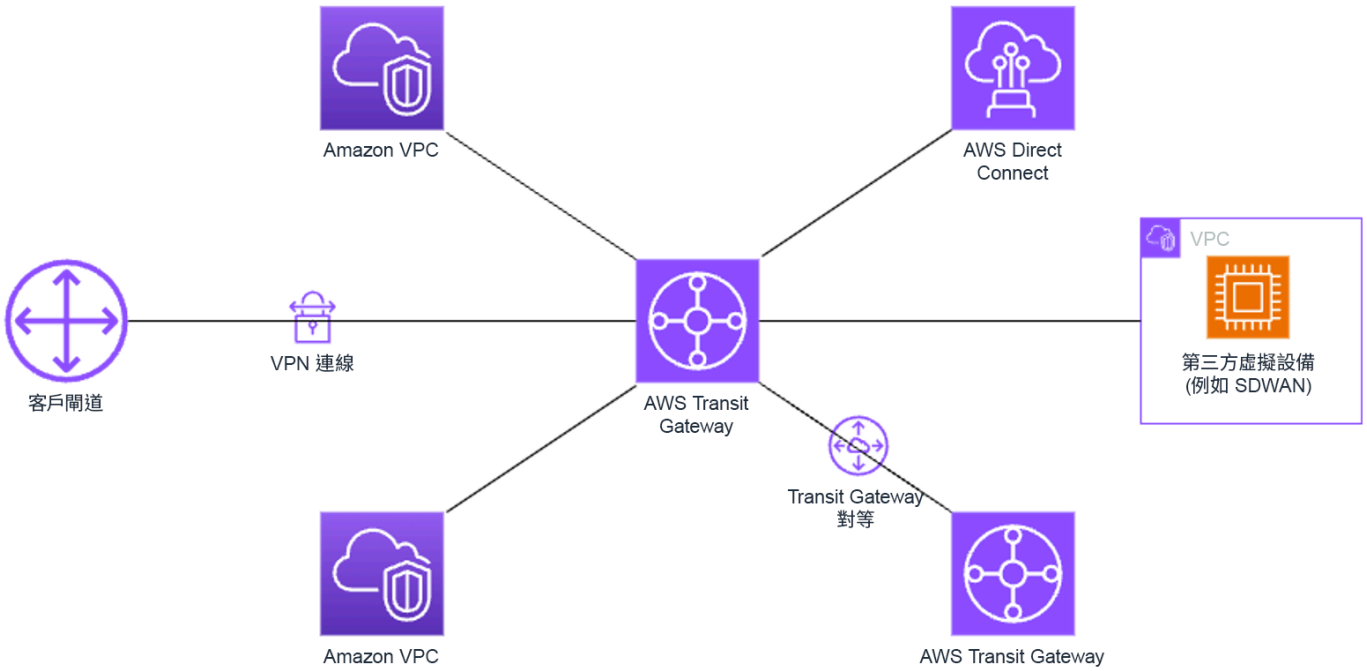
相關影片：

- AWS re:Invent 2018：進階 VPC 設計和 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)
- AWS re:Invent 2023：AWS 準備好踏出下一步了嗎？ 設計可促進成長與靈活性的網路 (NET310)

REL02-BP04 偏好軸輻式拓撲而非多對多網狀拓撲

連接多個私有網路 (例如虛擬私有雲端 (VPC)) 和內部部署網路時，請選擇軸輻式拓樸而不是網狀拓樸。在網狀拓撲中的每個網路都直接連線到其他網路，因而增加複雜性和管理開銷，但軸輻式架構與之不同，是透過單一集線器集中連線的。這種集中連線可簡化網路結構，並增強其可操作性、可擴展性和控制性。

AWS Transit Gateway 是一種受管、可擴展且高度可用的服務，專為建構在 AWS 上的軸輻式網路而設計。此服務可做為網路的中央樞紐，提供網路分割、集中路由，以及簡化對雲端和內部部署環境的連線。下圖說明如何使用 AWS Transit Gateway 建置軸輻式拓樸。



常見的反模式：

- 您的軸輻式架構中的路由政策過於複雜，致使網路效率降低，並使故障排除和主動管理變得複雜。
- 集線器內以路由為基礎的分段不足，可能會形成漏洞，進而使網路遭受未經授權的存取。
- 如果沒有經過仔細的最佳化處理，通過集線器路由的流量可能會導致更高的資料傳輸成本，特別是對於跨可用區域和區域的流量。有效的交通管理策略對控制費用至關重要。

建立此最佳實務的優勢：隨著連線網路的數量增加，網格連線的管理和擴展變得越來越具挑戰性。AWS Transit Gateway 提供可擴展且可靠的受管集線器，用於建構和操作軸輻式拓撲。使用 AWS Transit Gateway 時，您可以建立連線並集中跨多個網路的流量路由。

未建立此最佳實務時的風險暴露等級：中

實作指引

- 規劃您的網路。
- 建立您的 AWS Transit Gateway。
- 連接您的 VPC。
- 如有需要，建立 VPN 連線或 Direct Connect 閘道，並將它們與 Transit Gateway 相連接。
- 透過組態您的 Transit Gateway 路由表，定義如何在連接的 VPC 和其他連線之間轉送流量。
- 視需要使用 Amazon CloudWatch 監控和調整組態，以實現效能和成本最佳化。

資源

相關文件：

- 什麼是 Transit Gateway？
- 建置可擴展且安全的多 VPC AWS 網路基礎設施
- 使用 AWS Transit Gateway 區域間對等建置全球網路
- Amazon Virtual Private Cloud 連線選項
- 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

相關影片：

- AWS re:Invent 2023 - AWS 聯網基礎

- AWS re:Invent 2023 - 進階 VPC 設計及新功能

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 位址範圍

在對等、透過傳輸閘道連接或透過 VPN 連線時，每個 VPC 的 IP 位址範圍不得重疊。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 位址衝突。您也須有一種在需要時分配私有 IP 位址範圍的方法。IP 位址管理 (IPAM) 系統可以協助實現此自動化。

預期成果：

- VPC、內部部署環境或其他雲端供應商之間沒有 IP 位址範圍衝突。
- 適當的 IP 位址管理方便更輕鬆地擴展網路基礎設施，以適應網路需求的增長和變化。

常見的反模式：

- 在 VPC 中使用與內部部署、您的企業網路或其他雲端供應商相同的 IP 範圍
- 不追蹤用來部署工作負載之 VPC 的 IP 範圍。
- 依靠手動 IP 位址管理流程，例如試算表。
- CIDR 區塊大小過大或過小，會導致 IP 位址浪費或位址空間不足以供您的工作負載使用。

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 位址。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。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用 IPAM (例如 Amazon VPC IP Address Manager) 監控和管理您對 CIDR 的使用。AWS Marketplace 也提供數套 IPAM。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 以便用量依規劃增長。

實作步驟

- 擷取目前的 CIDR 消耗 (例如 VPC 和子網路)。
  - 使用服務 API 作業收集目前的 CIDR 消耗。
  - 使用 Amazon VPC IP Address Manager 探索資源。
- 記錄您目前的子網路用量。
  - 使用服務 API 作業收集每個區域中每個 VPC 的子網路。

  - 使用 Amazon VPC IP Address Manager 探索資源。
- 記錄目前用量。
- 判斷您是否已建立任何重疊的 IP 範圍。
- 計算備用容量。
- 識別重疊的 IP 範圍。如果需要連接重疊範圍，您可以遷移到新的位址範圍，或考慮使用私有 NAT 閘道或 AWS PrivateLink 等技術。

資源

相關的最佳實務：

- 保護網路

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
- 連接具有重疊 IP 範圍的網路
- 什麼是 Amazon VPC？
- 什麼是 IPAM？

相關影片：

- AWS re:Invent 2023 - 進階 VPC 設計及新功能
- AWS re:Invent 2019：適用於許多 VPC 的 AWS Transit Gateway 參考架構
- AWS re:Invent 2023：準備好進行下一步了嗎？ 設計可促進成長與靈活性的網路
- AWS re:Invent 2021 - {新推出} 在 AWS 上大規模管理您的 IP 位址

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) 會定義服務，具有依商業需求定義的明訂功能。微型服務使用領域模型和有界限的環境，沿著業務環境界限繪製服務界限。專注於業務領域和功能，有助於團隊為其服務定義獨立的可靠性要求。有界限的環境可隔離和封裝商業邏輯，讓團隊更適切地推論如何處理失敗。

預期成果： 工程師和業務利害關係人共同定義有界限的環境，並將其用來設計系統，作為滿足特定業務功能的服務。這些團隊使用既定的做法 (如事件風暴) 來定義要求。新的應用程式設計為服務妥善定義的界限和鬆散耦合。現有的整合型服務分解為 有界限的環境 ，系統設計改採 SOA 或微型服務架構。整合型服務重構時，會套用已建立的方法 (如 Bubble 環境) 和整合型分解模式。

領域導向服務會以一或多個不共用狀態的程序執行。它們會單獨回應需求的波動，並根據領域的特定要求來處理錯誤情境。

常見的反模式：

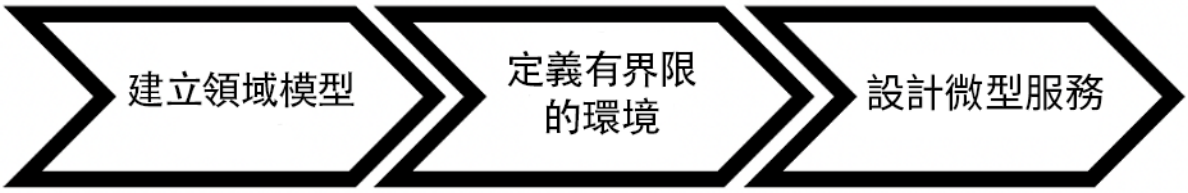
- 團隊是依據特定技術領域 (例如 UI 和 UX、中介軟體或資料庫) 組成的，而不是特定的業務領域。
- 應用程式跨多個領域責任。跨有界限環境的服務可能更難以維護，需要較大量的測試工作，且需要多個領域團隊參與軟體更新。
- 領域相依性 (例如領域實體程式庫) 會跨服務共用，因此一個服務領域出現變更時，需要變更其他服務領域
- 服務合約和商業邏輯無法以通用且一致的領域語言來表達實體，因此會導致翻譯層級使系統複雜化，並增加偵錯工作。

建立此最佳實務的優勢： 應用程式設計為獨立的服務，受到業務領域限制，並使用共同的商務語言。服務可以單獨測試和部署。服務符合實作領域的特定恢復能力要求。

未建立此最佳實務時的曝險等級： 高

實作指引

領域驅動決策 (DDD) 是依據業務領域設計和建置軟體的基礎方法。在建置專注於業務領域的服務時，使用現有架構將有所幫助。使用現有的整合型應用程式時，您可以利用分解模式提供已建立的技術，將應用程式現代化為服務。



領域驅動的決策

實作步驟

- 團隊可舉辦 事件風暴 研討會，以便箋格式快速識別事件、命令、彙總和領域。
- 在領域環境中形成領域實體和函數之後，您可以使用 有界限的環境將領域分成服務，其中具有相似功能和特性的實體會歸類成一組。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。
  - 例如，Amazon.com 網站實體可能包括包裝、交付、排程、價格、折扣和貨幣。
  - 包裝、交付和排程會分組到出貨環境中，而價格、折扣和貨幣則分組到訂價環境中。
- 將整合型服務分解為微型服務 概述了重構微型服務的模式。按業務功能、子領域或交易使用分解的模式，會與領域驅動的方法保持一致。
- 戰術 (如 Bubble 環境) 可讓您在現有或舊版應用程式中導入 DDD，而無須預先重寫和對 DDD 完整承諾。在 Bubble 環境方法中，使用服務對應和協調來建立小型的有界限環境 (或 抗損毀層)，以保護新定義的領域模型免受外部影響。

在團隊執行領域分析並定義實體和服務合約之後，他們可以利用 AWS 服務將其領域導向設計實作為雲端架構服務。

- 藉由定義執行領域商務規則的測試來起始您的開發。測試驅動的開發 (TDD) 和行為驅動的開發 (BDD) 可協助團隊將服務著重於解決業務問題上。
- 選取 AWS 服務 (最符合您的業務領域要求和 微型服務架構)：
  - AWS 無伺服器 讓您的團隊專注於特定的領域邏輯，而不是管理伺服器和基礎設施。
  - AWS 上的容器 簡化基礎設施的管理，讓您得以專注在您的領域要求上。
  - 專用資料庫 協助您根據領域要求找出最適合的資料庫類型。
- 在 AWS 上建置六邊形架構 概述了一個架構，用以將業務邏輯建置到從業務領域回溯運作的服務中，以滿足功能要求，然後附加整合適配器。使用 AWS 服務將介面詳細資訊與商業邏輯分開的模式，可協助團隊專注於領域功能及改善軟體品質。

資源

相關的最佳實務：

- REL03-BP01 選擇如何劃分工作負載
- REL03-BP03 每個 API 都提供服務合約

相關文件：

- AWS 微型服務
- 實作 AWS 上的微型服務
- 如何將整合型服務分成微型服務
- 在置身於舊式系統時開始使用 DDD
- 領域驅動設計：解決軟體核心的複雜性
- 在 AWS 上建置六邊形架構
- 將整合型服務分解為微型服務
- 事件風暴
- 有界限的環境之間的訊息
- 微型服務
- 測試驅動的開發
- 行為驅動的開發

相關範例：

- 企業雲端原生研討會
- 在 AWS 上設計雲端原生微型服務 (從 DDD/EventStormingWorkshop)

相關工具：

- AWS 雲端 資料庫
- AWS 上的無伺服器
- AWS 上的容器

REL03-BP03 每個 API 都提供服務合約

服務合約是 API 生產者與取用者之間的記錄協議，載明於機器可讀取的 API 定義中。合約版本控制策略可讓取用者繼續使用現有的 API，並在準備好時將應用程式遷移至更新的 API。只要遵守合約，就隨時可執行生產者部署。服務團隊可以使用自己選擇的技術堆疊，以滿足 API 合約要求。

預期成果：

常見的反模式： 以服務導向或微型服務架構建置的應用程式能夠獨立運作，同時具有整合的執行期相依性。當雙方遵循共同的 API 合約時，部署至 API 取用者或生產者的變更不會中斷整體系統的穩定性。透過服務 API 進行通訊的元件可以執行獨立運作的版本、升級至執行期相依性，或容錯移轉至災難復原 (DR) 站台，且彼此幾乎沒有影響或完全沒有影響。此外，離散服務能夠獨立擴展而滿足資源需求，不需要其他服務一起擴展。

- 建立不具強型別結構描述的服務 API。這會產生無法用來產生 API 繫結的 API，以及無法以程式設計方式驗證的承載。
- 未採用版本控制策略，會迫使 API 取用者更新和發行，或在服務合約發展時失敗。
- 錯誤訊息會透露基礎服務實作的詳細資訊，而不是說明領域環境和語言中的整合失敗。
- 未使用 API 合約來開發測試案例和模擬 API 實作，以允許單獨測試服務元件。

建立此最佳實務的優勢： 由透過 API 服務合約進行通訊的元件組成的分散式系統可提升可靠性。開發人員可在開發過程中及早發現潛在問題，並在編譯期間進行類型檢查，以確認請求和回應遵循 API 合約，且必要欄位存在。API 合約為 API 提供了清晰的自我記錄介面，並在不同的系統和程式設計語言之間提供了更好的互通性。

未建立此最佳實務時的曝險等級： 中

實作指引

在識別商業領域並確認工作負載區隔後，即可開發服務 API。首先，請定義機器可讀取的 API 服務合約，然後實作 API 版本控制策略。準備好透過 REST、GraphQL 等一般通訊協定或非同步事件來整合服務後，您可以將 AWS 服務併入您的架構中，以便將元件與強型別 API 合約整合。

服務 API 合約的 AWS 服務

將 AWS 服務 (包括 Amazon API Gateway、AWS AppSync和 Amazon EventBridge) 併入您的架構中，以在您的應用程式中使用 API 服務合約。Amazon API Gateway 可協助您直接與原生 AWS 服務和其他 Web 服務整合。API Gateway 支援 OpenAPI 規格 和版本控制。AWS AppSync 是一個

受管 GraphQL 端點，可藉由定義 GraphQL 結構描述來設定，用以定義查詢、變動和訂閱的服務介面。Amazon EventBridge 使用事件結構描述來定義事件及產生事件的程式碼繫結。

實作步驟

- 首先，為您的 API 定義合約。合約會說明 API 的功能，並定義強型別的資料物件和欄位，以用於 API 輸入和輸出。
- 在 API Gateway 中設定 API 時，您可以為端點匯入和匯出 OpenAPI 規格。
  - 匯入 OpenAPI 定義 可簡化 API 的建立，並且可以與 AWS 基礎設施即程式碼工具整合，例如 AWS Serverless Application Model 和 AWS Cloud Development Kit (AWS CDK)。
  - 匯出 API 定義 可簡化與 API 測試工具的整合，並為服務取用者提供整合規格。
- 您可以透過 AWS AppSync 來定義和管理 GraphQL API，方法是 定義 GraphQL 結構描述 檔案以產生合約介面，並簡化與複雜 REST 模型、多個資料庫資料表或舊版服務的互動。
- AWS Amplify 專案 (與 AWS AppSync 整合) 會產生強型別 JavaScript 查詢檔案以供您的應用程式使用，並產生 AWS AppSync GraphQL 用戶端程式庫用於 Amazon DynamoDB 資料表。
- 當您從 Amazon EventBridge 中取用服務事件時，事件會遵循結構描述登錄中已存在的結構描述，或您使用 OpenAPI 規格定義的結構描述。使用登錄中定義的結構描述時，您也可以從結構描述合約產生用戶端繫結，以將程式碼與事件整合。
- 擴充您的 API 或對其進行版本控制。在新增可使用選用欄位或必要欄位的預設值來設定的欄位時，擴充 API 是較簡單的選項。
  - REST 和 GraphQL 等通訊協定的 JSON 型合約可能非常適合進行合約展延。
  - SOAP 等通訊協定的 XML 型合約應進行服務取用者的測試，以確認合約展延的可行性。
- 在對 API 進行版本控制時，請考慮實作 Proxy 版本控制，使用 Facade 來支援版本，以便在單一程式碼基底中維護邏輯。
  - 透過 API Gateway，您可以使用 請求和回應對應 建立 Facade 以提供新欄位的預設值，或從請求或回應中去除已移除的欄位，以簡化因應合約變更的工作。透過此方法，基礎服務可以維護單一程式碼基底。

資源

相關的最佳實務：

- 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
- REL04-BP02 實作鬆耦合相依性

- REL05-BP03 控制和限制重試呼叫
- REL05-BP05 設定用戶端逾時

相關文件：

- 什麼是 API (應用程式設計介面)？
- 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
- 使用 OpenAPI 的 API Gateway 延伸
- OpenAPI 規格
- GraphQL：結構描述和類型
- Amazon EventBridge 程式碼繫結

相關範例：

- Amazon API Gateway：使用 OpenAPI 設定 REST API
- 使用 OpenAPI 設定 Amazon DynamoDB CRUD 應用程式的 Amazon API Gateway
- 無伺服器時代的現代化應用程式整合模式：API Gateway 服務整合
- 使用 Amazon CloudFront 實作標題型 API Gateway 版本控制
- AWS AppSync：建置用戶端應用程式

相關影片：

- 在 AWS SAM 中使用 OpenAPI 管理 API Gateway

相關工具：

- Amazon API Gateway
- AWS AppSync
- Amazon EventBridge

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 識別您依賴的分散式系統類型
- REL04-BP02 實作鬆耦合相依性
- REL04-BP03 持續執行工作
- REL04-BP04 將所有回應設為等冪

REL04-BP01 識別您依賴的分散式系統類型

分散式系統可以是同步、非同步或批次。同步系統必須盡快處理請求，並透過使用 HTTP/S、REST 或遠端程序呼叫 (RPC) 通訊協定進行同步請求和回應呼叫來互相通訊。非同步系統透過中介服務以非同步方式交換資料相互通訊，而無需結合個別系統。批次系統接收大量的輸入資料，在無人為干預的情況下執行自動化資料處理，並產生輸出資料。

期望的結果：設計與同步、非同步和批次相依項有效互動的工作負載。

常見的反模式：

- 工作負載會無限期等待其相依項的回應，這可能導致工作負載用戶端逾時，卻不知道對方是否已收到它們的請求。
- 工作負載使用相互同步呼叫的相依系統鏈。這需要每個系統都可用，並在整個鏈成功之前成功處理請求，因而導致潛在的脆弱行為和整體可用性。
- 工作負載以非同步方式與其相依項進行通訊，並依賴保證訊息一次性傳送的概念，而通常仍有可能接收到重複的訊息。
- 工作負載不使用適當的批次排程工具，並允許同時執行相同的批次工作。

建立此最佳實務的優勢：特定的工作負載通常會在同步、非同步和批次模式之間選擇實作一或多種通訊模式。這個最佳實務可協助您識別與每種通訊模式相關聯的不同權衡，讓您的工作負載能夠承受其任何相依項遭遇中斷的情況。

未建立此最佳實務時的風險暴露等級：高

實作指引

以下各節包含針對每種相依項類型的一般和特定實作指引。

一般指引

- 確保相依項提供的效能和可靠性服務層級目標 (SLO) 符合工作負載的效能和可靠性要求。
- 使用 AWS 可觀測性服務監控回應時間和錯誤率，確保您的相依項能在工作負載所需的層級提供服務。
- 識別工作負載與其相依項通訊時可能會面臨的潛在挑戰。分散式系統面臨的各種挑戰，可能會增加架構複雜性、營運負擔和成本。常見的挑戰包括延遲、網路中斷、資料遺失、擴展和資料複寫延遲。
- 實作強大的錯誤處理和記錄流程，以在您的相依項遇到問題時協助進行疑難排解。

同步相依

在同步通訊中，您的工作負載會傳送要求給其相依項，並封鎖等待回應的運作。當其相依項收到請求時，會嘗試盡快處理，並將回應傳回至您的工作負載。同步通訊的最大挑戰是它會導致時間耦合，這需要您的工作負載和其相依項同時可用。當您的工作負載需要與其相依項同步通訊時，請考量以下指引：

- 您的工作負載不應依賴多個同步相依項來執行單一函數。這個相依鏈增加了整體脆性，因為路徑中的所有相依項都必須可用才能成功完成請求。
- 當相依項不健全或無法使用時，請判斷如何處理錯誤和重試策略。避免使用雙模態行為。雙模態行為是指工作負載在正常和故障模式下呈現不同行為的情況。如需有關雙模態行為的詳細資訊，請參閱 REL11-BP05 使用靜態穩定性來預防雙模態行為。
- 請記住，快速檢錯比讓工作負載等待更好。例如，AWS Lambda 開發人員指南說明如何在調用函數時處理重試和失敗。Lambda
- 在工作負載呼叫其相依項時設定逾時。這種技術可避免等待太久或無限期等待回應的現象。如需有關此問題的實用討論，請參閱為延遲感知型 Amazon DynamoDB 應用程式調整 AWS Java SDK HTTP 請求設定。
- 將工作負載僅為滿足單一請求而呼叫其相依項的次數減到最少。它們兩者之間的冗長通話會增加耦合及延遲。

非同步相依項

若要暫時為您的工作負載與其相依項解耦，它們應該以非同步方式通訊。使用非同步方法，您的工作負載可以繼續進行其他任何處理，而無需等待其相依項或相依項鏈傳送回應。

當您的工作負載需要與其相依項非同步通訊時，請考慮下列指引：

- 根據您的使用案例和需求，決定是否使用訊息傳遞或事件串流。傳訊允許您的工作負載與其相依項通訊時，透過訊息代理程式傳送和接收郵件。事件串流允許您的工作負載及其相依項使用串流服務發佈和訂閱事件，而這些作為連續資料串流來傳遞的事件需要盡快受處理。

- 傳訊和事件串流以不同方式處理訊息，因此您需要根據以下方式做出取捨：
  - 訊息優先順序：訊息代理程式可以在處理一般訊息之前處理高優先順序的訊息。在事件串流中，所有訊息都具有相同的優先順序。
  - 訊息取用：訊息代理程式確保取用者接收到訊息。活動串流取用者必須追蹤他們最後一次閱讀的訊息。
  - 訊息排序：傳訊時，除非您採用先進先出 (FIFO) 方式，否則無法保證按照傳送的確切順序接收訊息。事件串流始終會保留資料的產生順序。
  - 訊息刪除：傳訊時，取用者必須在處理訊息後刪除該訊息。事件串流服務會將訊息附加到串流中，並保留在該串流中，直到訊息的保留期限到期為止。此刪除政策使事件串流適合重播訊息。
- 定義您的工作負載如何知道其相依項何時完成其工作。例如，當您的工作負載以非同步方式調用 Lambda 函數時，Lambda 會將事件放置在佇列中，並傳回不含其他資訊的成功回應。處理完後，Lambda 函數可以將結果傳送到目的地，根據成功或失敗進行設定。
- 建置您的工作負載時，利用冪等性處理重複的訊息。冪等性的意思是即使為相同訊息產生您的工作負載一次以上，工作負載的結果也不會改變。重要的是，如果發生網路失敗或未收到確認，傳訊或串流服務將會重新傳遞訊息。
- 如果您的工作負載沒有收到其相依項的回應，則需要重新提交請求。請考慮限制重試次數，以保留工作負載的 CPU、記憶體和網路資源以處理其他請求。AWS Lambda 文件顯示如何處理非同步調用的錯誤。
- 利用適合的可觀測性、除錯和追蹤工具來管理和操作工作負載與其相依項的非同步通訊。您可以使用 Amazon CloudWatch 監視傳訊和事件串流服務。您也可以利用 AWS X-Ray 檢測工作負載，以快速針對疑難排解問題取得洞見。

批次相依項

批次系統接收輸入資料，啟動一系列工作來處理該資料，並生產一些輸出資料，完全無需手動介入。視資料大小而定，工作的執行可能需要幾分鐘 (在某些情況下) 到幾天時間。當您的工作負載與其批次相依項通訊時，請考慮下列指引：

- 定義工作負載應執行批次工作的時段。您的工作負載可以設定重複模式來調用批次系統，例如每小時或每個月尾。
- 確定資料輸入和受處理資料輸出的位置。選擇一種儲存服務，例如 Amazon Simple Storage Services (Amazon S3)、Amazon Elastic File System (Amazon EFS) 和 Amazon FSx for Lustre，讓您的工作負載能夠大規模讀取和寫入檔案。
- 如果您的工作負載需要呼叫多個批次工作，您可以利用 AWS Step Functions 簡化在 AWS 或內部部署執行的批次工作的協同運作。此範例專案示範了使用 Step 函數、AWS Batch 和 Lambda 的批次工作協同運作。
- 監控批次工作以尋找異常情況，例如完成工作需要比預期更長的時間。您可以使用 CloudWatch Container Insights 等工具來監控 AWS Batch 環境和工作。在這種情況下，您的工作負載會從頭開始停止下一個工作，並通知相關員工有關例外情況。

資源

相關文件：

- AWS 雲端 Operations：監控和可觀測性
- Amazon 建置者資料中心：分散式系統的挑戰
- REL11-BP05 使用靜態穩定性來防止雙模態行為
- AWS Lambda 開發人員指南：AWS Lambda 中的錯誤處理和自動重試
- 為延遲感知型 Amazon DynamoDB 應用程式調整 AWS Java SDK HTTP 請求設定。
- AWS 傳訊
- 什麼是串流資料？
- AWS Lambda 開發人員指南：非同步調用
- Amazon Simple Queue Service 常見問答集：FIFO 佇列
- Amazon Kinesis Data Streams 開發人員指南：處理重複記錄
- Amazon Simple Queue Service 開發人員指南：Amazon SQS 可用的 CloudWatch 指標
- Amazon Kinesis Data Streams 開發人員指南：利用 Amazon CloudWatch 監控 Amazon Kinesis Data Streams 服務
- AWS X-Ray 開發人員指南：AWS X-Ray 概念
- AWS GitHub 上的範例：AWS Step Functions 複雜的協調器應用程式
- AWS Batch 使用者指南：AWS Batch CloudWatch Container Insights

相關影片：

- AWS Summit SF 2022 - 使用 AWS 的全堆疊可觀測性和應用程式監控 (COP310)

相關工具：

- Amazon CloudWatch
- Amazon CloudWatch Logs
- AWS X-Ray
- Amazon Simple Storage Services (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx for Lustre
- AWS Step Functions
- AWS Batch

REL04-BP02 實作鬆耦合相依性

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。

在緊耦合的系統中，對某個元件進行變更時，可能必須變更其他依賴此元件的元件，從而導致所有元件的效能降低。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。

鬆耦合可讓您修改程式碼或新增功能至某個元件，同時將依賴該元件的其他元件的風險降至最低。其還能讓您在元件層級提供細微的恢復能力，您可以橫向擴展，甚至是變更相依性的基礎實作。

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 Amazon SQS 佇列，或如 Amazon Kinesis 或 AWS Step Functions 等串流資料平台。

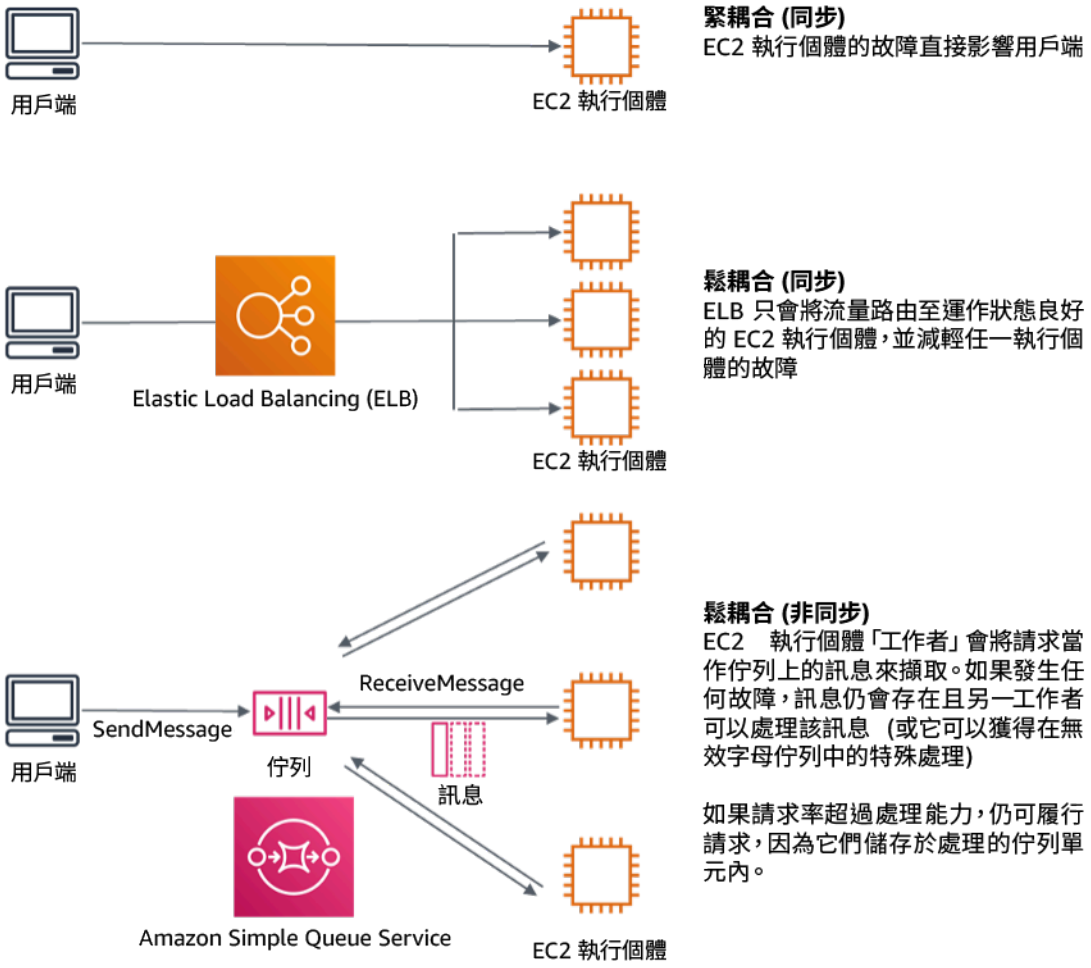


圖 4：佇列系統和負載平衡器之間具有鬆散耦合的相依性。

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 (事件取用者) 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。

預期成果：實作鬆耦合的相依性可讓您將失敗的影響範圍最小化到元件層級，從而有助於診斷和解決問題。它還能簡化開發週期，讓團隊在模組化層級實作變更，而不會影響依賴此元件之其他元件的效能。這種方法可讓您根據資源需求，以及對成本效益有所貢獻之元件的使用情況，在元件層級進行橫向擴展。

常見的反模式：

- 部署整合型工作負載。
- 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。
- 使用共用資料的緊耦合。鬆耦合系統應避免透過共用資料庫或其他形式的緊耦合資料儲存共用資料，這可能會重新引入緊耦合並阻礙可擴展性。
- 忽略反壓。當元件無法以相同的速率處理傳入的資料時，工作負載應該要有能力減緩或停止傳入的資料。

建立此最佳實務的好處：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，進而提高彈性和靈活性。避免一個元件中的失敗影響其他元件。

未建立此最佳實務時的風險暴露等級：高

實作指引

實作鬆耦合相依性。有各種解決方案可讓您建置鬆耦合的應用程式。這些解決方案包括用於實作全受管佇列的服務、自動化工作流程、對事件的回應以及 API 等，其有助於將元件的行為與其他元件隔離，從而提高彈性和靈活性。

- 建置事件驅動架構：Amazon EventBridge 可協助您建置鬆耦合和分散式的事件驅動架構。
- 在分散式系統中實作佇列：您可以使用 Amazon Simple Queue Service (Amazon SQS) 來整合和解耦分散式系統。
- 將元件容器化為微型服務：微型服務可讓團隊建置由小型獨立元件組成的應用程式，這些元件會通過明確定義的 API 進行通訊。Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您更快地開始使用容器。
- 使用 Step Functions 管理工作流程：Step Functions 可協助您將多個 AWS 服務協調為彈性工作流程。
- 利用發布-訂閱 (pub/sub) 傳訊架構：Amazon Simple Notification Service (Amazon SNS) 可提供從發布者到訂閱用戶 (也稱為生產者和取用者) 的訊息傳遞功能。

實作步驟

- 事件驅動架構中的元件會由事件啟動。事件是系統中發生的動作，例如使用者將某個商品新增至購物車。動作成功時會產生可啟動系統下一個元件的事件。
  - 使用 Amazon EventBridge 建置事件驅動應用程式
  - AWS re:Invent 2022 - 使用 Amazon EventBridge 設計事件驅動整合

- 分散式傳訊系統有三個需要針對佇列型架構來實作的主要部分。這些部分包括分散式系統的元件、用於解耦的佇列 (分散在 Amazon SQS 伺服器上)，以及佇列中的訊息。典型的系統中有負責將訊息啟動至佇列的生產者，以及從佇列接收訊息的取用者。為了備援，佇列會在多個 Amazon SQS 伺服器儲存訊息。
  - 基本的 Amazon SQS 架構
  - 使用 Amazon Simple Queue Service 在分散式應用程式之間傳送訊息
- 充分利用的微型服務會增強可維護性並提高可擴展性，因為鬆耦合元件由獨立團隊管理。其還能夠在發生變更時隔離單一元件的行為。
  - 在 AWS 上實作微型服務
  - 開始建構吧！ 使用容器建構微型服務
- AWS Step Functions 可讓您建置分散式應用程式、將程序自動化、協調微型服務等。將多個元件協同運作到自動化工作流程中可讓您解耦應用程式中的相依性。
  - 使用 AWS Step Functions 和 AWS Lambda 建立無伺服器工作流程
  - AWS Step Functions 入門

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？
- 與整合型分手
- 使用 AWS Step Functions 和 Amazon SQS 協調佇列型微型服務
- 基本的 Amazon SQS 架構
- 佇列式架構

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)
- 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
- AWS re:Invent 2022 - 使用 Amazon EventBridge 設計事件驅動整合
- AWS re:Invent 2017：Elastic Load Balancing 深入剖析與最佳實務

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 限流請求
- REL05-BP03 控制和限制重試呼叫
- REL05-BP04 快速檢錯和限制佇列
- REL05-BP05 設定用戶端逾時
- REL05-BP06 盡可能讓系統變成無狀態
- 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

即使相依性變得不可用，應用程式元件仍應繼續執行其核心功能。它們有可能提供稍微陳舊的資料、備用資料，甚至未提供任何資料。這可確保整體系統運作在本地化失敗時只會受到最低限度的阻礙，同時提供核心商業價值。

預期成果： 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。元件的失敗模式應被視為正常運作。工作流程應適當設計，使此類失敗不會導致完全失敗，或至少會進入可預測和可復原的狀態。

常見的反模式：

- 未識別所需的核心業務功能。即使在相依性失敗期間，也不測試元件是否正常運作。
- 發生錯誤時，或只有多個相依性的其中之一無法使用，且仍可傳回部分結果時，就不提供資料。
- 當交易部分失敗時建立不一致的狀態。
- 沒有替代方法可存取中央參數存放區。
- 因重新整理失敗而使本機狀態失效或清空，而未考量這麼做的後果。

建立此最佳實務的優勢： 按正常程序降級可改善系統整體的可用性，並且讓最重要的功能保持運作，即使在失敗期間亦然。

未建立此最佳實務時的曝險等級： 高

實作指引

按正常程序實作降級，有助於將相依性失敗對元件功能的影響降到最低。理想情況下，元件會偵測相依性失敗，並以對其他元件或客戶造成最小影響的方式解決這些問題。

按正常程序降級的架構，意味著在相依性設計期間會考量潛在的失敗模式。對於每種失敗模式，都有一種方法可至少將元件最關鍵的功能提供給呼叫者或客戶。這些考量可能會成為可供測試和驗證的其他要求。理想情況下，即使有一或多個相依性失敗，元件仍然能夠以可接受的方式執行其核心功能。

這在商業上和技術上都同樣值得討論。所有業務要求都很重要，都應盡可能地滿足。然而，若無法滿足各項要求將會如何，仍是值得提出的問題。一個系統可以設計成可用且一致的，但在必須放棄一項要求的情況下，何者較重要？ 對於付款處理，可能應選擇一致性。對於即時應用程式，可能應選擇可用性。對於面向客戶的網站，答案可能取決於客戶的期望。

這意味著什麼，取決於元件的要求，以及應將哪些內容視為其核心功能。例如：

- 電子商務網站可能會顯示來自多個不同系統的資料，例如個人化推薦、排名最高的產品，以及客戶訂單在登陸網頁上的狀態。當一個上游系統失敗時，顯示其他所有內容，而不是向客戶顯示錯誤頁面，仍然是合理的。
- 如果個別作業之一失敗，執行批次寫入的元件仍然可以繼續處理批次。實作重試機制應該要很簡單。為此，您可以向呼叫者傳回關於哪些操作成功、哪些操作失敗及其為何失敗的資訊，或將失敗的請求放入無效字母佇列以實作非同步重試。失敗操作的相關資訊也應記錄下來。
- 處理交易的系統必須確認是否執行了所有更新，或完全未執行更新。對於分佈式交易，可使用 Saga 模式在相同交易的後續操作失敗的情況下回復先前的操作。在此，核心功能保有一致性。

- 具時間性的系統應該能夠處理未及時回應的相依性。在這類情況下，可以使用斷路器模式。若來自相依性的回應開始逾時，系統可以切換到不會進行其他呼叫的關閉狀態。
- 應用程式可從參數存放區讀取參數。使用一組預設的參數建立容器映像，並在參數存放區無法使用時使用這些參數，會很有效用。

請注意，在元件失敗的情況下採取的路徑需進行測試，且應遠比主要途徑簡單。一般來說，應避免使用備用策略。

實作步驟

識別外部和內部相依性。請考量其中可能會發生什麼樣的失敗。思考在這類失敗期間，將上游和下游系統以及客戶受到的負面影響降到最低的方法。

以下列出相依性，並說明如何在其失敗時按正常程序降級：

1. 相依性的部分失敗： 一個元件可能會向下游系統提出多個請求，可以是對一個系統的多個請求，或者對多個系統各提出一個請求。視業務環境而定，對此可能會有不同的適當處理方式 (如需詳細資訊，請參閱實作指引中的先前範例)。
2. 下游系統因高負載而無法處理請求： 如果對下游系統的請求持續失敗，繼續重試是沒有意義的。這樣可能會對已過載的系統產生額外的負載，並使復原變得更加困難。此時可以使用斷路器模式，以監控對下游系統的失敗呼叫。若有大量呼叫失敗，將會停止向下游系統傳送更多請求，且偶爾才會讓呼叫通過，以測試下游系統是否已恢復可用性。
3. 參數存放區無法使用： 若要轉換參數存放區，可以使用容器或機器映像中包含的軟相依性快取或有效的預設值。請注意，這些預設值需要保持最新狀態，並包含在測試套件中。
4. 監控服務或其他非功能性相依性無法使用： 如果元件間歇性地無法傳送記錄、指標或追蹤給中央監控服務，最好還是照常執行業務功能。一般而言，長時間不記錄或推送指標且未顯示任何訊息，是不可接受的。此外，某些使用案例可能需要完整的稽核項目才能滿足合規要求。
5. 關聯式資料庫的主要執行個體可能無法使用： Amazon Relational Database Service 就像絕大多數的關聯式資料庫一樣，只能有一個主要寫入器執行個體。這會對寫入工作負載造成單一失敗點，並使擴展變得更加困難。透過使用多可用區域組態以獲得高可用性，或使用 Amazon Aurora 無伺服器以獲得更好的擴展性，可以減輕部分問題。對於非常高的可用性要求，完全不依賴主要寫入器是有效用的。對於唯讀的查詢可以使用讀取複本，以提供備援和橫向擴展的能力，而不僅僅是縱向擴展。寫入可以緩衝處理 (例如，在 Amazon Simple Queue Service 佇列中)，如此，即使主要寫入器暫時無法使用，仍然可以接受客戶的寫入請求。

資源

相關文件：

- Amazon API Gateway：限流 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 建置者資料中心：避免分散式系統的備用
- Amazon 建置者資料中心：避免無法逾越的佇列待辦項目
- Amazon 建置者資料中心：快取挑戰和策略
- Amazon 建置者資料中心：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 建置者資料中心簡介 (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 限流請求

限流請求以減輕因需求非預期地增加而耗盡資源。系統會處理低於限流速率的請求，並拒絕超過定義限制的請求，且傳回一則訊息，指出請求已遭到限流。

預期成果： 由於客戶流量突然增加、洪水攻擊或重試風暴而造成的大量尖峰，可透過請求限流來緩解，讓工作負載能夠繼續正常處理支援的請求量。

常見的反模式：

- API 端點限流未實作或保留為預設值，而未考量預期的數量。
- API 端點未經過負載測試，或未測試限流限制。
- 限流請求率，而不考量請求大小或複雜性。
- 測試請求率上限或請求大小上限，但不同時測試兩者。

- 資源不會佈建在測試時建立的相同限制。
- 未針對應用程式對應用程式 (A2A) API 取用者設定或考量使用計畫。
- 水平擴展的佇列取用者未進行最大並行設定。
- 未實作個別 IP 地址的速率限制。

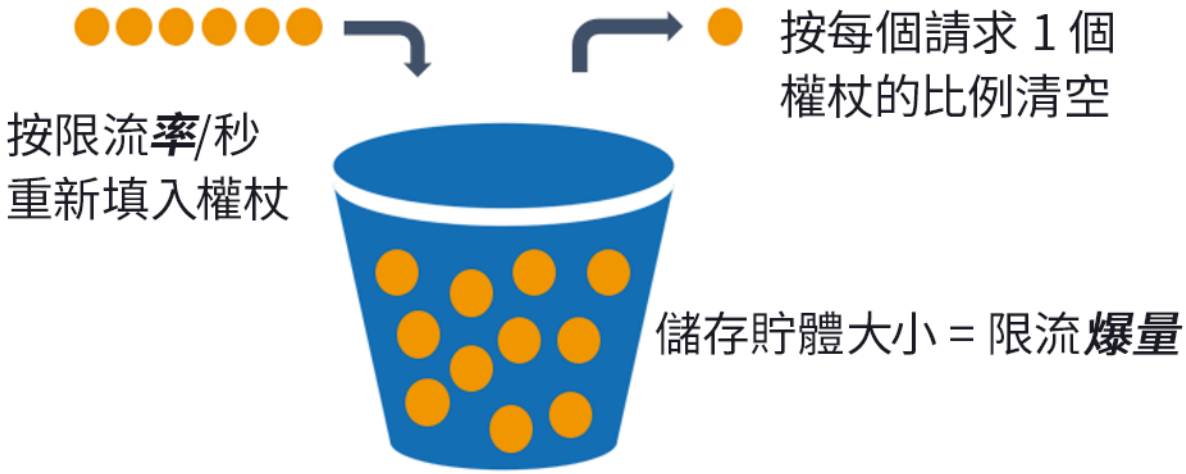
建立此最佳實務的優勢： 設定限流限制的工作負載能夠在非預期的數量尖峰情況下正常運作，並成功處理已接受的請求負載。API 和佇列的請求若突然或持續激增將會受到限制，且不會耗盡請求處理資源。速率限制會限流個別請求者，使來自單一 IP 地址或 API 取用者的大量流量不會耗盡資源而影響到其他取用者。

未建立此最佳實務時的曝險等級： 高

實作指引

服務應設計為處理已知的請求容量；此容量可透過負載測試來建立。如果請求到達率超過限制，會有適當的回應訊號指出請求已受到限流。這可讓取用者處理錯誤並於稍後重試。

當您的服務需要限流實作時，請考慮實作權杖儲存貯體演算法 (在此演算法中，權杖對於請求具重要性)。權杖會按每秒的限流率重新填入，並依照每個請求一個權杖的比例非同步清空。



權杖儲存貯體演算法。

Amazon API Gateway 會根據帳戶和區域限制實作權杖儲存貯體演算法，並且可根據使用計畫對個別用戶端進行設定。此外，Amazon Simple Queue Service (Amazon SQS) 和 Amazon Kinesis 可以緩衝處理請求以緩解請求率，並對於可處理的請求允許提高限流率。最後，您可以透過 AWS WAF 實作速率限制，以限流會產生異常高負載的特定 API 取用者。

實作步驟

您可以為 API 設定具有限流限制的 API Gateway，並在超出限制時傳回 429 #### 錯誤。您可以將 AWS WAF 用於 AWS AppSync 和 API Gateway 端點，以就個別 IP 地址啟用速率限制。此外，如果您的系統可容忍非同步處理，您可以將訊息放入佇列或串流中，以加快對服務用戶端的回應速度，進而提升到更高的限流率。

若使用非同步處理，當您將 Amazon SQS 設定為 AWS Lambda 的事件來源時，您可以 設定並行上限，以避免高事件速率耗用您的工作負載或帳戶中其他服務所需的可用帳戶並行執行配額。

雖然 API Gateway 提供了權杖儲存貯體的受管實作，在您無法使用 API Gateway 的情況下，您可以對服務使用權杖儲存貯體的語言特定開放原始碼實作 (請參閱「資源」中的相關範例)。

- 了解並設定 API Gateway 限流限制 (在個別區域的帳戶層級、個別階段的 API，以及個別使用計畫層級的 API 金鑰)。
- 套用 AWS WAF 速率限制規則 於 API Gateway 和 AWS AppSync 端點，以防止洪水及阻止惡意 IP。此外也可在 A2A 取用者的 AWS AppSync API 金鑰上設定速率限制規則。
- 考慮您是否需要比 AWS AppSync API 的速率限制更高的限流控制，如果需要，請在 AWS AppSync 端點前面設定 API Gateway。
- 將 Amazon SQS 佇列設定為 Lambda 佇列取用者的觸發器時，請將 並行上限 設定為適當值，正好足以符合您的服務水準目標，但不會耗用並行限制而影響到其他 Lambda 函數。當您透過 Lambda 使用佇列時，請考慮在相同帳戶和區域中的其他 Lambda 函數上設定預留並行。
- 使用 API Gateway 進行 Amazon SQS 或 Kinesis 的原生服務整合以緩衝處理請求。
- 如果您無法使用 API Gateway，請查看語言特定程式庫，為您的工作負載實作權杖儲存貯體演算法。查看範例區段並自行研究，以尋找合適的程式庫。
- 測試您預計要設定的限制，或您打算允許增加的限制，並記錄已測試的限制。
- 請勿將限制提高到您在測試時建立的範圍外。增加限制時，請先確認佈建的資源已等同於或大於測試情境中的資源，然後再套用增加。

資源

相關的最佳實務：

- REL04-BP03 持續執行工作
- REL05-BP03 控制和限制重試呼叫

相關文件：

- Amazon API Gateway：限流 API 請求以獲得更佳的輸送量
- AWS WAF：以速率為基礎的規則陳述式
- 導入使用 Amazon SQS 作為事件來源時的 AWS Lambda 並行上限
- AWS Lambda：並行上限

相關範例：

- 三項最重要的 AWS WAF 速率型規則
- Java Bucket4j
- Python 權杖儲存貯體
- 節點權杖儲存貯體
- .NET 系統執行緒速率限制

相關影片：

- 使用 AWS AppSync 實作 GraphQL API 安全最佳實務

相關工具：

- Amazon API Gateway
- AWS AppSync
- Amazon SQS
- Amazon Kinesis
- AWS WAF

REL05-BP03 控制和限制重試呼叫

使用指數退避，在每次重試之間的間隔逐漸拉長後重試請求。在重試之間導入抖動以隨機化重試間隔。限制重試次數上限。

預期成果： 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。在正常操作期間，這些元件可以回應具有暫時性或有限錯誤的請求，以及無論是否重試都將持續存在的錯誤。當用戶端向服務發出請求時，請求會取用資源，包括記憶體、執行緒、連線、連接埠，或任何其他有限的資源。控制和限制重試是釋出資源並將資源耗用量降到最低的策略，可讓處於壓力下的系統元件不致不堪負荷。

當用戶端請求逾時或收到錯誤回應時，他們應該判斷是否要重試。如果執行重試，他們會使用具有抖動和最大重試值的指數退避來執行此作業。因此，後端服務和程序從負載中得到緩解並獲得自我修復的時間，進而更快速地復原和提供成功的請求服務。

常見的反模式：

- 在未新增指數退避、抖動和最大重試值的情況下實作重試。退避和抖動有助於避免因為在共用間隔內無意間進行協調重試而產生人為流量尖峰。
- 在不測試影響的情況下實作重試，或者假設重試已直接內建到 SDK 中，而未測試重試情境。
- 未能了解從相依性發佈的錯誤代碼，因而重試了所有錯誤，包括有明確原因指出缺少權限的錯誤、組態錯誤，或其他預期必須要手動干預才能解決的狀況。
- 未解決可觀測性實務，包括對重複的服務失敗進行監控和提醒，使基礎問題廣為人知並且可以解決。
- 在內建或第三方重試功能堪用時，開發自訂重試機制。
- 以複合重試的方式在應用程式堆疊的多個層級重試，會嘗試在重試風暴中進一步耗用資源。請務必了解這些錯誤如何影響您所依賴的應用程式，然後僅在一個層級實作重試。
- 重試不是等冪的服務呼叫，導致非預期的副作用，例如重複的結果。

建立此最佳實務的優勢： 重試可協助用戶端在請求失敗時獲得所需的結果，但也會耗用更多伺服器的時間來取得他們想要的成功回應。若失敗是罕見或暫時性的，重試可以有效運作。若失敗是由資源超載引起的，重試可能會使情況變得更糟。在用戶端重試中新增具有抖動的指數退避，可讓伺服器在資源超載導致失敗時進行復原。抖動可避免將請求對應到尖峰，而退避會減少將重試新增至正常請求負載所造成的負載上升。最後，請務必設定最大重試次數或經過時間，以避免建立會產生亞穩態失敗的積存。

未建立此最佳實務時的曝險等級： 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。引進抖動來隨機化重試間隔，並限制重試次數上限。

有些 AWS SDK 依預設會實作重試和指數退避。若情況允許，在工作負載中使用這些內建 AWS 實作。在呼叫等冪的服務時，以及重試可改善用戶端可用性時，在您的工作負載中實作類似的邏輯。根據您的使用案例確定逾時時間以及何時停止重試。為那些重試使用案例建置和模擬演練測試情境。

實作步驟

- 確認應用程式堆疊中的最佳層級，以針對您應用程式所依賴的服務實作重試。

- 請注意現有的 SDK 是否會透過指數退避和抖動為您選擇的語言實作經過驗證的重試策略，如果會請優先予以採用，而不是撰寫自己的重試實作。
- 請確認 服務是等冪的 ，然後再實作重試。實作重試後，請務必在生產環境中加以測試和定期模擬演練。
- 在呼叫 AWS 服務 API 時，使用 AWS SDK 和 AWS CLI ，並了解重試組態選項。確認預設值是否適用於您的使用案例，並視需要進行測試和調整。

資源

相關的最佳實務：

- REL04-BP04 將所有回應設為等冪
- REL05-BP02 限流請求
- REL05-BP04 快速檢錯和限制佇列
- REL05-BP05 設定用戶端逾時
- REL11-BP01 監控工作負載的所有元件以偵測故障

相關文件：

- AWS 中的錯誤重試和指數退避
- Amazon 建置者資料中心：逾時、重試、退避與抖動
- 指數退避和抖動
- 使用等冪 API 確保重試安全性

相關範例：

- Spring 重試
- Resilience4j 重試

相關影片：

- 重試、退避和抖動：AWS re:Invent 2019：Amazon 建置者資料中心簡介 (DOP328)

相關工具：

- AWS SDK 和工具：重試行為
- AWS Command Line Interface：AWS CLI 重試

REL05-BP04 快速檢錯和限制佇列

在服務無法成功回應請求時快速檢錯。如此將可釋出與請求關聯的資源，並且使服務可在資源用盡時復原。快速檢錯是一種完善的軟體設計模式，可用來在雲端中建置高度可靠的工作負載。佇列也是一種完善的企業整合模式，可以平滑負載，並且讓用戶端在可容忍非同步處理時釋出資源。如果服務在正常情況下能夠成功回應，但在請求速率太高時會失敗，請使用佇列來緩衝請求。不過，請勿允許建置長佇列積存，這可能導致用戶端已放棄的過時請求受到處理。

預期成果： 當系統遇到資源爭用、逾時、例外狀況或灰色失敗而使服務水準目標無法達成時，快速檢錯的策略可加快系統復原速度。必須吸納流量尖峰且能支應非同步處理的系統，可讓用戶端使用佇列緩衝處理後端服務的請求以快速釋出請求，藉此提升可靠性。緩衝處理要排入佇列的請求時，會實作佇列管理策略，以避免發生無法克服的積存。

常見的反模式：

- 在 DLQ 磁碟區上實作訊息佇列，但不設定無效字母佇列 (DLQ) 或警示，以偵測系統失敗。
- 未測量訊息在佇列中的存留期，這是一種延遲測量，用以了解佇列取用者何時進度落後或發生錯誤導致重試。
- 當處理這些訊息沒有任何價值，且業務需求已不存在時，未清除佇列中已積存的訊息。
- 在後進先出 (LIFO) 佇列可更妥善地滿足用戶端需求時設定了先進先出 (FIFO) 佇列，例如，當不需要嚴格排序，以及積存處理延遲了所有新的和時間敏感的請求，導致所有用戶端都經歷違反服務水準的狀況。
- 將內部佇列公開給用戶端，而不是公開管理工作接受以及將請求放入內部佇列的 API。
- 藉由將資源需求分攤到不同請求型態，將過多的工作請求類型合併到可能加劇積存條件的單一佇列中。
- 儘管需要不同的監控、逾時和資源分配，仍在同一佇列中處理複雜而簡單的請求。
- 不驗證輸入或使用判斷提示在軟體中實作快速檢錯的機制，以對可用正常程序處理錯誤的較高層級元件快顯例外狀況。
- 不會從請求路由中移除錯誤的資源，特別是在失敗處於灰色地帶，因損毀和重新啟動、間歇性相依性失敗、容量減少或網路封包遺失而導致成功與失敗並存。

建立此最佳實務的優勢： 快速檢錯的系統更容易偵錯和修正，並且在發佈至生產環境之前，常會出現編碼和組態方面的問題。納入有效佇列策略的系統，可針對流量尖峰和間歇性系統失敗狀況提供更高的恢復能力和可靠性。

未建立此最佳實務時的曝險等級： 高

實作指引

快速檢錯的策略可以編碼為軟體解決方案，並設定到基礎設施中。除了快速檢錯以外，佇列也是一種簡單而強大的架構技術，可將系統元件平滑負載分離。Amazon CloudWatch 提供監控失敗和發出相關警示的功能。已知系統失敗時，可以叫用緩解策略，包括背離受損的資源。當系統實作佇列與 Amazon SQS 和其他佇列技術來平滑負載，必須考量如何管理佇列積存，以及訊息取用失敗。

實作步驟

- 在軟體中實作程式化判斷提示或特定指標，並使用它們來明確提醒系統問題。Amazon CloudWatch 可協助您根據應用程式日誌模式和 SDK 檢測來建立指標及提醒。
- 使用 CloudWatch 指標和警示背離受損的資源，這些資源會增加處理的延遲，或持續無法處理請求。
- 藉由設計 API 使用非同步處理來接受請求，並使用 Amazon SQS 將請求附加到內部佇列，然後透過成功訊息回應產生訊息的用戶端，讓用戶端可以釋出資源並繼續進行其他工作，同時讓後端佇列取用者處理請求。
- 藉由比較現在時間與訊息時間戳記，在每次從佇列中取出訊息時產生 CloudWatch 指標，來測量和監控佇列處理延遲。
- 因失敗而無法成功處理訊息，或無法在服務水準協議內處理磁碟區中的流量尖峰時，請將較舊或過多的流量排除至溢滿佇列。這可讓您優先處理新工作，並且等到有可用的容量時再處理較舊的工作。這種技術類似於 LIFO 處理，方便對所有新工作進行正常的系統處理。
- 使用無效字母或重新驅動佇列，將無法處理的訊息從積存移到可稍後再研究和解析的位置
- 進行重試，或在可接受的情況下，藉由比較目前時間與訊息時間戳記，捨棄與請求用戶端不再相關的訊息，將舊訊息捨棄。

資源

相關的最佳實務：

- REL04-BP02 實作鬆耦合相依性
- REL05-BP02 限流請求

- REL05-BP03 控制和限制重試呼叫
- REL06-BP02 定義和計算指標 (彙總)
- REL06-BP07 透過您的系統監控請求的端對端追蹤

相關文件：

- 避免無法處理的佇列積存
- 快速檢錯
- 如何防止 Amazon SQS 佇列中不斷增加的訊息積存？
- Elastic Load Balancing：區域轉移
- Amazon Route 53 應用程式復原控制器：流量容錯移轉的路由控制

相關範例：

- 企業整合模式：無效字母通道

相關影片：

- AWS re:Invent 2022 - 操作高可用性多可用區域應用程式

相關工具：

- Amazon SQS
- Amazon MQ
- AWS IoT Core
- Amazon CloudWatch

REL05-BP05 設定用戶端逾時

在連線和請求上妥善設定逾時、有系統地對其進行驗證，並且不要依賴預設值，因為它們不知道工作負載具體細節。

預期成果： 用戶端逾時應考量與等待需要花費異常時間才能完成的請求相關的用戶端、伺服器和工作負載成本。由於無法知道任何逾時的確切原因，用戶端必須使用服務知識來找出對可能原因和適當逾時的期望

用戶端連線根據設定的值逾時。經歷逾時後，用戶端決定退回並重試，或開啟 斷路器。這些模式可避免發出可能使基礎錯誤情況惡化的請求。

常見的反模式：

- 不知道系統逾時或預設逾時。
- 不知道正常的請求完成時間。
- 不知道完成請求異常耗時的可能原因，或是與等待這些作業完成相關聯的用戶端、服務或工作負載效能成本。
- 不知道受損的網路只有在達到逾時後才會造成請求失敗的可能性，以及未採用較短逾時的用戶端和工作負載效能的成本。
- 不測試連線和請求的逾時情境。
- 將逾時設定得太高，這可能會導致較長的等待時間，並增加資源使用率。
- 將逾時設定得太低，導致人為失敗。
- 忽略模式以處理遠端呼叫 (例如斷路器和重試) 的逾時錯誤。
- 不考慮監控服務呼叫錯誤率、延遲的服務水準目標，以及延遲離群值。這些指標可提供對積極或寬鬆逾時的洞見

建立此最佳實務的優勢： 遠端呼叫逾時已設定，且系統設計為按正常程序處理逾時，以便在遠端呼叫回應異常緩慢，而逾時錯誤由服務用戶端正常處理時，可以保留資源。

未建立此最佳實務時的曝險等級： 高

實作指引

針對任何服務相依性呼叫和任何跨程序的呼叫，同時設定連線逾時和請求逾時。許多架構都提供內建的逾時功能，但請注意，對您的服務目標而言，有些架構具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。

決定逾時策略時，請考量下列事項：

- 由於請求的內容、目標服務受損或網路分割失敗，處理請求的時間可能會比平常更長。
- 內容異常昂貴的請求可能會耗用不必要的伺服器和用戶端資源。在此情況下，讓這些請求逾時而不重試，可以保留資源。服務也應透過限流和伺服器端逾時，來保護自己免受異常昂貴的內容影響。

- 因服務受損而異常耗時的請求可能會逾時並重試。應考量請求和重試的服務成本，但如果原因是當地語系化的損害，則重試應該不會很昂貴，而且將可降低用戶端資源耗用量。逾時也可能會根據損害的性質釋出伺服器資源。
- 因網路傳遞請求或回應失敗而需要長時間才能完成的請求，可能會逾時並重試。由於請求或回應未傳遞，因此無論逾時長度為何，結果都是失敗。在此情況下，逾時不會釋出伺服器資源，但會釋出用戶端資源並改善工作負載效能。

利用完善的設計模式 (例如重試和斷路器)，按正常程序處理逾時並支援快速檢錯方法。AWS SDK 和 AWS CLI 允許設定連線和請求逾時，以及具有指數退避和抖動的重試。AWS Lambda 函數支援設定逾時，且透過 AWS Step Functions，您可以建置低程式碼斷路器，以利用預先建置的 AWS 服務和 SDK 整合。AWS App Mesh Envoy 提供逾時和斷路器功能。

實作步驟

- 設定遠端服務呼叫的逾時，並利用內建的語言逾時功能或開放原始碼逾時程式庫。
- 當您的工作負載使用 AWS SDK 進行呼叫時，請檢閱文件以了解語言特定的逾時組態。
  - Python
  - PHP
  - .NET
  - Ruby
  - Java
  - Go
  - Node.js
  - C++
- 在工作負載中使用 AWS SDK 或 AWS CLI 命令時，請設定預設逾時值，方法是設定 AWS 組態預設值 (針對 `connectTimeoutInMillis` 和 `tlsNegotiationTimeoutInMillis`)。
- 套用 命令列選項 `cli-connect-timeout` 和 `cli-read-timeout` 以控制 AWS 服務的一次性 AWS CLI 命令。
- 監控遠端服務呼叫是否有逾時，並對持續性錯誤設定警示，以便您可以主動處理錯誤案例。
- 實作 CloudWatch 指標 和 CloudWatch 異常偵測 於呼叫錯誤率、延遲的服務水準目標，以及延遲離群值，讓您能夠深入了解如何管理過於積極或寬鬆的逾時。
- 設定逾時於 Lambda 函數。
- API Gateway 用戶端在處理逾時期間必須實作本身的重試。API Gateway 支援 50 毫秒至 29 秒的整合逾時 (用於下游整合)，且整合請求若逾時將不會重試。

- 實作 斷路器 模式，以避免在逾時發生時進行遠端呼叫。開啟線路以避免呼叫失敗，並在呼叫正常回應時關閉線路。
- 對於容器型工作負載，請檢閱 App Mesh Envoy 功能，以利用內建的逾時和斷路器。
- 使用 AWS Step Functions 為遠端服務呼叫建置低程式碼斷路器，尤其是在呼叫 AWS 原生 SDK 和支援的 Step Functions 整合以簡化工作負載的情況下。

資源

相關的最佳實務：

- REL05-BP03 控制和限制重試呼叫
- REL05-BP04 快速檢錯和限制佇列
- REL06-BP07 透過您的系統監控請求的端對端追蹤

相關文件：

- AWS SDK：重試與逾時
- Amazon 建置者資料中心：逾時、重試、退避與抖動
- Amazon API Gateway 配額和重要注意事項
- AWS Command Line Interface：命令列選項
- AWS SDK for Java 2.x：設定 API 逾時
- 使用組態物件和組態參考的 AWS Botocore
- AWS SDK for .NET：重試與逾時
- AWS Lambda：設定 Lambda 函數選項

相關範例：

- 使用斷路器模式搭配 AWS Step Functions 和 Amazon DynamoDB
- Martin Fowler：CircuitBreaker

相關工具：

- AWS SDK
- AWS Lambda

- Amazon SQS
- AWS Step Functions
- AWS Command Line Interface

REL05-BP06 盡可能讓系統變成無狀態

系統不應要求狀態，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。這可讓伺服器任意置換，而不會對可用性造成影響。

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。

建立此最佳實務的優勢：設計為無狀態的系統更適應水平擴展，因此可以根據波動的流量和需求增加或移除容量。此系統本質上也具有抵禦失敗的能力，並在應用程式開發中提供靈活性和敏捷性。

未建立此最佳實務時的風險暴露等級：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，並且可以容忍個別節點的故障。分析並了解應用程式中維持架構內狀態的元件。這可協助您評估轉換至無狀態設計的潛在影響。無狀態架構會解除配對使用者資料，並卸載工作階段資料。這提供了獨立擴展每個元件的彈性，以滿足不同的工作負載需求並優化資源使用率。

實作步驟

- 識別並了解應用程式中的有狀態元件。
- 透過將使用者資料與核心應用程式邏輯分隔及管理來解除配對資料。
  - Amazon Cognito 可以使用身分池、使用者集區和 Amazon Cognito Sync 等功能，將使用者資料與應用程式的程式碼解除配對。
  - 您可以使用 AWS Secrets Manager 將機密儲存在安全、集中的位置來解偶使用者資料。這意味著應用程式的程式碼不需要儲存密碼，因此本身更加安全。

  - 考慮使用 Amazon S3 來儲存大型非結構化資料，例如影像和文件。您的應用程式可以在需要時檢索這些資料，進而不需要將其儲存在記憶體中。
  - 使用 Amazon DynamoDB 儲存使用者設定檔等資訊。您的應用程式可以近即時查詢此資訊。
- 將工作階段資料卸載到資料庫、快取或外部檔案。
  - Amazon ElastiCache、Amazon DynamoDB、Amazon Elastic File System (Amazon EFS) 以及 Amazon MemoryDB for Redis 是可以用來卸載工作階段資料的 AWS 服務範例。
- 在確定需要使用所選儲存解決方案保留哪些狀態和使用者資料後，設計無狀態架構。

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：

- Amazon 建置者資料中心：避免分散式系統的備用
- Amazon 建置者資料中心：避免無法逾越的佇列待辦項目
- Amazon 建置者資料中心：快取挑戰和策略
- 在 AWS 上的無狀態 Web 層的最佳實務

REL05-BP07 實作緊急控制桿

緊急控制桿是可緩解工作負載所受之可用性影響的快速程序。

緊急控制桿的運作方法是使用已知且經過測試的機制來停用、限流或變更元件或相依性的行為。這可以減輕因需求意外增加導致資源耗盡所造成的工作負載受損，並降低工作負載內非關鍵元件的故障影響。

預期成果：透過實作緊急控制桿，您可以建立已知良好的程序，以維持工作負載中關鍵元件的可用性。在啟用緊急控制桿期間，工作負載應該會適度降級，並繼續執行其業務關鍵功能。如需適度降級的詳細資訊，請參閱 REL05-BP01 實作適度降級，以將適用的硬相依性轉換為軟相依性。

常見的反模式：

- 非關鍵相依性失敗會影響核心工作負載的可用性。
- 未在非關鍵元件受損期間測試或驗證關鍵元件的行為。

- 沒有為啟用或停用緊急控制桿定義明確且決定性的準則。

建立此最佳實務的優勢：實作緊急控制桿可以藉由為解析器提供已建立的程序來回應意外的需求突增或非關鍵相依性的失敗，以改善工作負載中關鍵元件的可用性。

未建立此最佳實務時的風險暴露等級：中

實作指引

- 識別工作負載中的關鍵元件。
- 設計和建構工作負載中的關鍵元件，以承受非關鍵元件的故障。
- 進行測試以驗證非關鍵元件失敗期間您關鍵元件的行為。
- 定義和監控相關指標或觸發器以啟動緊急控制桿程序。
- 定義構成緊急控制桿的程序 (手動或自動)。

實作步驟

- 識別工作負載中的業務關鍵元件。
  - 工作負載中的每個技術元件應對應到其相關業務功能，並將其排名為關鍵或非關鍵。如需 Amazon 的關鍵和非關鍵功能範例，請參閱任何一天都可以是 Prime Day：Amazon.com 搜尋如何使用混沌工程處理每秒超過 84000 個請求。
  - 這同時是技術和業務方面的決策，並且會因組織和工作負載而異。
- 設計和建構工作負載中的關鍵元件，以承受非關鍵元件的故障。
  - 在相依性分析期間，請考慮所有潛在的故障模式，並驗證您的緊急控制桿機制能為下游元件提供關鍵功能。
- 進行測試以驗證緊急控制桿啟動期間您關鍵元件的行為。
  - 避免雙模態行為。如需詳細資訊，請參閱 REL11-BP05 使用靜態穩定性來防止雙模態行為。
- 定義、監控和警示相關指標，以啟動緊急控制桿程序。
  - 尋找適合監控的指標取決於您的工作負載。一些範例指標是延遲或失敗的相依性請求次數。
- 定義構成緊急控制桿的程序 (手動或自動)。
  - 這可能包括卸載、限流請求或實作適度降級。

資源

相關的最佳實務：

- 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 限流請求
- REL11-BP05 使用靜態穩定性來防止雙模態行為

相關文件：

- 自動化安全、無人為介入的部署
- 任何一天都可以是 Prime Day：Amazon.com 搜尋如何使用混沌工程處理每秒超過 84,000 個請求

相關影片：

- AWS re:Invent 2020：透過不可變實現可靠性、一致性和可信度

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.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)

當組織偵測到潛在問題時，他們會將即時通知和警示傳送給適當的人員和系統，以便快速有效地應對這些問題。

預期成果： 根據服務和應用程式指標設定相關警示，就可以快速回應操作事件。違反警示閾值時，系統會通知適當的人員和系統，以便解決潛在問題。

常見的反模式：

- 將警示的閾值設得過高，會導致無法傳送重要通知。
- 將警示的閾值設得太低，導致使用者因通知過多的干擾而無法針對重要提醒採取行動。
- 當使用情況改變時，未更新警示及其閾值。
- 針對透過自動化動作解決的最佳警示，將通知傳送給人員而未引發自動化動作，會導致傳送過多的通知。

建立此最佳實務的優勢： 將即時通知和警示傳送給適當的人員和系統，以便及早發現問題並快速回應操作事故。

未建立此最佳實務時的曝險等級： 高

實作指引

工作負載應具備即時處理和警示功能，以改善可能影響應用程式可用性問題的可偵測性，並作為自動化回應的觸發程式。組織可以透過使用已定義的指標建立警示來執行即時處理和警示，以便在發生重大事件或指標超過閾值時收到通知。

Amazon CloudWatch 可讓您建立 指標 和複合警示，過程中根據靜態閾值、異常偵測和其他條件使用 CloudWatch 警示。如需有關可使用 CloudWatch 設定的警示類型詳細資訊，請參閱 CloudWatch 文件的警示一節。

您可以為團隊建構 AWS 資源的指標和警示自訂檢視，過程中使用 CloudWatch 儀表板。您可以透過 CloudWatch 主控台中的可自訂首頁，在單一檢視中監控多個區域的資源。

警示可以執行一或多個動作，例如將通知傳送給 或向 Amazon SNS 主題、執行 Amazon EC2 動作或 Amazon EC2 Auto Scaling 動作，或 建立 OpsItem 或 事故 (AWS Systems Manager)。

Amazon CloudWatch 使用 Amazon SNS ，於警示變更狀態時傳送通知，將訊息傳遞從發布者 (生產者) 提供給訂閱用戶 (消費者)。如需設定 Amazon SNS 通知的詳細資訊，請參閱 設定 Amazon SNS。

CloudWatch 傳送 EventBridge 的安全 事件，尤其是每當在 CloudWatch 警示進行建立、更新、刪除，或是狀態變更時。您可以使用 EventBridge 搭配這些事件來建立執行動作的規則，例如，當警示狀態變更時就通知您，或自動觸發在您帳戶中的事件，過程是使用 Systems Manager 自動化功能。

何時應該使用 EventBridge 和 Amazon SNS？

EventBridge 和 Amazon SNS 可用於開發事件驅動的應用程式，將視具體需求來做出選擇。

在建置應用程式以回應您自己應用程式、SaaS 應用程式和 AWS 服務中的事件時，建議使用 Amazon EventBridge。EventBridge 是唯一直接與第三方 SaaS 合作夥伴整合的事件型服務。EventBridge 還會自動從 200 多個 AWS 服務中擷取事件，而不需開發人員在帳戶中建立任何資源。

EventBridge 會將已定義的 JSON 架構用在事件，並協助您建立在整個事件內文套用的規則，以選取要轉寄至 目標的事件。EventBridge目前支援 20 多項 AWS 服務做為目標，包括 AWS Lambda、Amazon SQS、Amazon SNS、Amazon Kinesis Data Streams和 Amazon Data Firehose。

針對需要高散發的應用程式 (數千或數百萬個端點)，建議使用 Amazon SNS。我們看到的常見模式是客戶使用 Amazon SNS 做為規則的目標，以篩選所需的事件並散發到多個端點。

訊息是非結構化的，且可以是任何格式。Amazon SNS 支援將訊息轉寄至六種不同的目標，包括 Lambda、Amazon SQS、HTTP/S 端點、SMS、行動推送和電子郵件。Amazon SNS 一般的延遲時間短於 30 毫秒。透過將服務設定為傳送 AWS 訊息，各種 Amazon SNS 服務就能做到這一點 (超過 30 個，包括 Amazon EC2、Amazon S3和 Amazon RDS)。

實作步驟

1. 建立警示，過程中使用 Amazon CloudWatch 警示。
   a. 指標警示會監控單一 CloudWatch 指標或與 CloudWatch 指標相依的表達式。與超過一段時間間隔的閾值相比，警示會根據指標或表達式的值起始一或多個動作。此動作可能包括將通知傳送給或向 Amazon SNS 主題、執行 Amazon EC2 動作或 Amazon EC2 Auto Scaling 動作，或 建立 OpsItem 或 事故 (AWS Systems Manager)。
   b. 複合警示由規則表達式組成，該規則表達式會將您已建立的其他警示條件納入考量。只有在符合所有規則條件時，複合警示才會進入警示狀態。在複合警示規則表達式中指定的警示可能會包括指標警示和其他複合警示。複合警示可在狀態變更時傳送 Amazon SNS 通知，並可建立 Systems Manager 建立和追蹤這些改善 或 事故 (在進入警示狀態時)，但其無法執行 Amazon EC2 或 Auto Scaling 動作。
2. 設定 Amazon SNS 通知。您可以在建立 CloudWatch 警示時，包含在警示變更狀態時傳送通知的 Amazon SNS 主題。
3. 在 EventBridge 中建立規則 且此規則會比對指定的 CloudWatch 警示。每個規則都支援包括 Lambda 函數的多個目標。例如，您可以定義在可用磁碟空間不足時啟動的警示，該警示會透過 EventBridge 規則觸發 Lambda 函數以清理空間。如需 EventBridge 目標的詳細資訊，請參閱 EventBridge 目標。

資源

相關 Well-Architected 的最佳實務：

- 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)
- REL12-BP01 使用程序手冊調查失敗

相關文件：

- Amazon CloudWatch
- CloudWatch Logs 洞見
- 使用 Amazon CloudWatch 警示
- 使用 Amazon CloudWatch 儀表板
- 使用 Amazon CloudWatch 指標
- 設定 Amazon SNS 通知
- CloudWatch 異常偵測
- CloudWatch Logs 資料保護
- Amazon EventBridge
- Amazon Simple Notification Service

相關影片：

- reinvent 2022 可觀測性影片
- AWS re:Invent 2022 - Amazon 的可觀測性最佳實務

相關範例：

- One Observability 研討會
- Amazon EventBridge 到 AWS Lambda，由 Amazon CloudWatch 警示進行回饋控制

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。

實作警示的自動即時處理，以便系統可以採取快速的糾正措施，並嘗試在觸發警示時防止故障或服務降級。對警示的自動回應可能包括替換故障元件、調整運算容量、將流量重新導向到運作狀態良好的主機、可用區域或其他區域，以及操作人員通知。

預期成果：識別即時警示，並設定警示的自動處理，以調用適當動作，採取這些動作以維持服務水準目標和服務水準協議 (SLA)。自動化的範圍從單一元件的自我修復活動到全站點的容錯移轉。

常見的反模式：

- 針對關鍵的即時警示沒有清晰的清單或目錄。
- 對關鍵警示沒有自動回應 (例如，當運算資源即將耗盡時，發生自動擴展)。
- 矛盾的警示回應動作。
- 操作人員在收到提醒通知時沒有可以遵循的標準作業程序 (SOP)。
- 未監控組態變更，因為未偵測到的組態變更可能會導致工作負載停機。
- 沒有復原意外組態變更的策略。

建立此最佳實務的優勢：自動化警示處理可以提高系統復原能力。系統會自動採取糾正措施，減少人為介入時容易出錯的手動活動。工作負載運作符合可用性目標，並減少服務中斷。

未建立此最佳實務時的風險暴露等級：中

實作指引

為了有效管理提醒並自動化其回應，請根據提醒的重要性和影響來進行分類，記錄回應程序，並在為任務排名前規劃好回應。

識別需要特定動作的任務 (通常會在執行手冊中詳細說明)，並檢查所有執行手冊和程序手冊以確定哪些任務可以自動化。可以定義的動作通常也可以自動化。如果動作無法自動化，請在 SOP 中記錄手動步驟，並對操作人員進行相關培訓。持續挑戰手動程序以尋求自動化機會，以便您可以建立和維護用來自動化提醒回應的計畫。

實作步驟

1. 建立警示清單：若要取得所有警示的清單，您可以使用 AWS CLI (使用 Amazon CloudWatch 命令 `describe-alarms`)。根據您設定的警示數量而定，您可能需要使用分頁來擷取每個呼叫的警示子集，或者，您也可以使用 AWS SDK，使用 API 呼叫取得警示。
2. 記錄所有警示動作：不論是手動還是自動的，請更新執行手冊與所有警示及其動作。AWS Systems Manager 會提供預先定義的執行手冊。如需執行手冊的詳細資訊，請參閱使用執行手冊。如需如何檢視執行手冊內容的詳細資訊，請參閱檢視執行手冊內容。

3. 設定和管理警示動作：對於任何需要動作的警示，請指定使用 CloudWatch SDK 的自動化動作。例如，您可以建立和啟用警示的動作，或停用警示的動作，以根據 CloudWatch 警示自動變更 Amazon EC2 執行個體的狀態。

   您也可以使用 Amazon EventBridge 來自動回應系統事件，例如應用程式可用性問題或資源變更。您可以建立規則以指出您感興趣的事件，以及當事件符合規則時要採取的動作。可以自動啟動的動作包括調用 AWS Lambda 函數、叫用 Amazon EC2 Run Command、將事件轉送至 Amazon Kinesis Data Streams，以及查看使用 EventBridge 自動化 Amazon EC2。

4. 標準作業程序 (SOP)：根據您的應用程式元件，AWS Resilience Hub 建議使用多個 SOP 範本。您可以使用這些 SOP 記錄在發出提醒時操作員應遵循的所有程序。您也可以根據 Resilience Hub 建議來建構 SOP，但您需要具有相關彈性政策的 Resilience Hub 應用程式，以及對該應用程式進行歷史彈性評估。SOP 的建議會由彈性評估產生。

   Resilience Hub 會與 Systems Manager 搭配運作，透過提供一些可以作為這些 SOP 基礎的 SSM 文件來自動執行 SOP 的步驟。例如，Resilience Hub 可能會建議使用 SOP 來根據現有的 SSM 自動化文件新增磁碟空間。

5. 使用 Amazon DevOps Guru 執行自動化動作：您可以使用 Amazon DevOps Guru 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。使用 DevOps Guru 時，您可以近乎即時地監控多個來源的營運資料串流 (包括 Amazon CloudWatch 指標、AWS Config、AWS CloudFormation 和 AWS X-Ray)。您也可以使用 DevOps Guru 來自動地在 OpsCenter 中建立 OpsItems，並將事件傳送至 EventBridge 以進行其他自動化。

資源

相關的最佳實務：

- 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)
- REL08-BP01 將執行手冊用於部署等標準活動

相關文件：

- AWS Systems Manager 自動化
- 建立 EventBridge 規則，以透過 AWS 資源觸發事件
- One Observability 研討會

- 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)

相關影片：

- AWS re:Invent 2022 - Amazon 的可觀測性最佳實務
- AWS re:Invent 2020：使用 AWS Systems Manager 將任何作業自動化
- AWS Resilience Hub 簡介
- 為 Amazon DevOps Guru 通知建立自訂票證系統
- 使用 Amazon DevOps Guru 啟用多帳戶洞見彙總功能

相關範例：

- 可靠性研討會
- Amazon CloudWatch 和 Systems Manager 研討會

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

在透過服務元件處理請求時追蹤請求，讓產品團隊可以更輕鬆地分析和偵錯問題，並改善效能。

預期成果： 對所有元件進行全面追蹤的工作負載可輕易進行偵錯，藉由簡化根本原因探索來改善錯誤和延遲的 平均解決時間 (MTTR)。端對端追蹤可縮短探索受影響的元件時間，並詳細剖析錯誤或延遲的根本原因。

常見的反模式：

- 追蹤可用於某些元件，但不適用於所有元件。例如，在未追蹤 AWS Lambda 的情況下，團隊可能無法清楚了解尖峰工作負載中的冷啟動所造成的延遲。
- 未對追蹤設定綜合金絲雀或實際使用者監控 (RUM)。若沒有金絲雀或 RUM，追蹤分析就會省略用戶端互動遙測，而產生不完整的效能設定檔。
- 混合式工作負載同時包含雲端原生和第三方追蹤工具，但未採取相關步驟來選擇及完全整合單一追蹤解決方案。根據選擇的追蹤解決方案，應使用雲端原生追蹤 SDK 來檢測不是雲端原生的元件，或使用第三方工具來擷取雲端原生追蹤遙測。

建立此最佳實務的優勢： 當開發團隊收到問題的提醒時，他們可以看到系統元件互動的全貌，包括個別元件與記錄、效能和失敗的關聯性。由於追蹤可讓您輕鬆地以視覺化方式識別根本原因，調查根本原因的所需時間將可縮短。詳細了解元件互動的團隊，可在解決問題時做出更明智、更快速的決策。諸如何時應叫用災難復原 (DR) 容錯移轉，或何處最適合實作自我修復策略之類的決策，可藉由分析系統追蹤來改善，最終提升客戶對服務的滿意度。

未建立此最佳實務時的曝險等級： 中

實作指引

操作分散式應用程式的團隊，可使用追蹤工具來建立關聯性識別碼、收集請求追蹤，以及建置連網元件的服務圖。所有應用程式元件均應包含在請求追蹤中，包括服務用戶端、中介軟體閘道和事件匯流排、運算元件和儲存體 (包括鍵值存放區和資料庫)。在端對端追蹤組態中包含綜合金絲雀和實際使用者監控，以測量遠端用戶端互動和延遲，以便您根據服務水準協議和目標正確評估系統效能。

您可以使用 AWS X-Ray 和 Amazon CloudWatch 應用程式監控 檢測服務，在請求通過您的應用程式時提供請求的完整檢視。X-Ray 會收集應用程式遙測，並可讓您在承載、函數、追蹤、服務、API 間將其視覺化和加以篩選，並且可針對無程式碼或低程式碼的系統元件開啟。CloudWatch 應用程式監控包含 ServiceLens，可將您的追蹤與指標、日誌和警示整合在一起。CloudWatch 應用程式監控也包含用來監控端點和 API 的綜合功能，以及用來檢測 Web 應用程式用戶端的實際使用者監控。

實作步驟

- 對所有受支援的原生服務使用 AWS X-Ray，例如 Amazon S3、AWS Lambda 和 Amazon API Gateway。這些 AWS 服務可使用基礎設施即程式碼、AWS SDK 或 AWS Management Console 來啟用具有組態切換的 X-Ray。
- 檢測應用 適用於 Open Telemetry 的 AWS Distro 和 X-Ray 或第三方收集代理程式。
- 檢閱 AWS X-Ray 開發人員指南， 以了解程式設計語言特定的實作方式。這些文件章節會詳細說明如何檢測 HTTP 請求、SQL 查詢，以及應用程式設計語言特有的其他程序。
- 將 X-Ray 追蹤用於 Amazon CloudWatch 綜合金絲雀 和 Amazon CloudWatch RUM 以分析最終使用者用戶端通過您下游 AWS 基礎設施的請求路徑。
- 根據資源運作狀態和金絲雀遙測來設定 CloudWatch 指標和提醒，以便團隊快速收到問題的提醒，然後可使用 ServiceLens 深入探討追蹤和服務圖。
- 啟用第三方追蹤工具的 X-Ray 整合，例如 Datadog、New Relic或 Dynatrace (如果您將第三方工具用於主要追蹤解決方案)。

資源

相關的最佳實務：

- REL06-BP01 監控工作負載的所有元件 (產生)
- REL11-BP01 監控工作負載的所有元件以偵測故障

相關文件：

- 什麼是 AWS X-Ray？
- Amazon CloudWatch：應用程式監控
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 偵錯
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態
- 整合 AWS X-Ray 與其他 AWS 服務
- 適用於 OpenTelemetry 的 AWS Distro 和 AWS X-Ray
- Amazon CloudWatch：使用綜合監控
- Amazon CloudWatch：使用 CloudWatch RUM
- 設定 Amazon CloudWatch 綜合金絲雀和 Amazon CloudWatch 警示
- 可用性和超越各種可能：了解和改善 AWS 上分散式系統的恢復能力

相關範例：

- One Observability 工作坊

相關影片：

- AWS re:Invent 2022 - 如何跨多個帳戶監控應用程式
- 如何監控您的 AWS 應用程式

相關工具：

- AWS X-Ray
- Amazon CloudWatch
- Amazon Route 53

REL 7.如何設計工作負載以適應需求變更？

可擴展的工作負載提供了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，或啟動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，變更 Auto Scaling 群組中的 EC2 執行個體數量，或透過 AWS Management Console 或 AWS CLI 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)。

預期成果：在偵測到故障或客戶體驗降級時，會啟動擴展活動 (自動或手動)，以恢復可用性。

未建立此最佳實務時的風險暴露等級：中

實作指引

在工作負載中的所有元件實作可觀測性和監控，以監控客戶體驗並偵測故障。定義會擴展所需資源的手動或自動程序。如需詳細資訊，請參閱 REL11-BP01 監控工作負載的所有元件以偵測故障。

實作步驟

- 定義會擴展所需資源的手動或自動程序。
  - 擴展程序取決於工作負載內不同元件的設計方式。
  - 擴展程序也會根據所使用的基礎技術而有所不同。
    - 使用 AWS Auto Scaling 的元件可以使用擴展計劃來設定用於擴展資源的一組指示。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。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，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。如需詳細資訊，請參閱擴展計畫的運作方式。
    - Amazon EC2 Auto Scaling 可確認您擁有正確數量的 Amazon EC2 執行個體可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和最大數量，而 Amazon EC2 Auto Scaling 可確保您的群組大小永遠不會低於或高於這些限制。如需詳細資訊，請參閱什麼是 Amazon EC2 Auto Scaling？
    - Amazon DynamoDB 自動擴展使用 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，以在不需限流的情況下處理突然增加的流量。如需詳細資訊，請參閱使用 DynamoDB 自動擴展自動管理輸送容量。

資源

相關的最佳實務：

- REL07-BP01 取得或擴展資源時使用自動化
- REL11-BP01 監控工作負載的所有元件以偵測故障

相關文件：

- AWS Auto Scaling：擴展計畫的運作方式
- 使用 DynamoDB 自動擴展自動管理輸送容量
- 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，衡量擴展活動是否能達到工作負載要求。

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。在雲端中，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段，應將生產中的負載測試納入考慮，並動員所有在場人員共同分析結果並處理可能出現的問題。

常見的反模式：

- 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。
- 依據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效，並能夠識別要監看哪些指標，以便及時識別接近該負載的跡象，從而解決問題並避免由此失效造成的影響。

未建立此最佳實務時的曝險等級： 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。
  - 在 AWS 上執行分散式負載測試：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。
    - 實作負載驅動程式。您可以使用自訂程式碼、開放原始碼或商業軟體實作負載驅動程式。
    - 一開始用小容量執行負載測試。您在負載驅動到較小容量 (可能和單一執行個體或容器一樣小) 之後立刻發現一些影響。
    - 對較大容量執行負載測試。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：

- 在 AWS 上執行分散式負載測試：模擬數千名連線的使用者
- 對應用程式執行負載測試

相關影片：

- AWS Summit ANZ 2023：透過 AWS 分散式負載測試讓您放心加快腳步

REL 8.如何實作變更？

變更須在受控的情況下，才能部署新功能，並確認工作負載和運作環境執行已知的軟體，且能夠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 將恢復能力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。

期望的結果：您可以使用自動化來執行功能測試，而相關的測試資料可縮短測試持續時間和費用，並提高測試結果的準確性。您可以將功能測試整合為部署流程的一部分，這可協助您自動執行發布管道，以實現快速且可靠的應用程式和基礎設施更新。

常見的反模式：

- 您在部署管道以外手動執行測試。
- 您利用手動緊急工作流程跳過自動化作業中的測試步驟。
- 您為了加快時間進程，而沒有遵循既定的計畫和流程。

建立此最佳實務的優勢：功能測試會驗證系統是否根據指定的要求運作。該測試用於一致地驗證元件的預期工作順序，例如使用者介面、API、資料庫和原始程式碼。當您檢查系統的這些元件時，功能測試會驗證每個功能是否符合預期的行為，進而保護使用者期望和軟體的完整性。將功能測試整合成定期部署的一部分，並使用自動化部署所有變更，進而降低導入人為錯誤的可能性。

未建立此最佳實務時的風險暴露等級：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。如果不符合成功條件，則管道會停止或復原。AWS CodePipeline 會為自動化測試提供連續交付管道，允許測試者自動化整個測試和部署流程。它會與 AWS CodeBuild 和 AWS CodeDeploy 等 AWS 服務整合，以將軟體開發生命週期的組建、測試和部署階段自動化。

實作步驟

- 設定管道：使用 AWS CodePipeline 主控台或 AWS Command Line Interface (CLI) 設定來源、建置、測試和部署階段。
  - 定義來源：您可以使用 AWS CodePipeline，自動從 GitHub、AWS CodeCommit 或 Bitbucket 等版本控制系統中檢索原始程式碼，該系統會確認始終使用最新程式碼進行測試。
  - 自動化組建和測試：AWS CodeBuild 可以自動建置和測試您的程式碼，並產生測試報告。該服務支援受歡迎的測試架構，如 JUnit、NUnit 和 TestNG。
  - 部署程式碼：建置並測試程式碼後，AWS CodeDeploy 可以將其部署到您的測試環境，包括 Amazon EC2 執行個體、AWS Lambda 函數或內部部署伺服器。
  - 監控管道：AWS CodePipeline 可以追蹤管道的進度和每個階段的狀態。您也可以根據測試執行狀態使用品質檢查機制來封鎖管道。您也可以接收任何關於管道階段失敗或管道完成的通知。

資源

相關文件：

- 搭配 AWS CodeBuild 使用 AWS CodePipeline 測試程式碼和執行組建版本

- 在 AWS CodeBuild 中記錄和監控
- 功能測試的指標

REL08-BP03 將恢復能力測試整合為部署的一部分

透過特意導入系統故障來整合恢復能力測試，以衡量其在發生中斷情況時的能力。恢復能力測試與通常在部署週期中整合的單元和功能測試不同，恢復測試僅專注於識別系統中非預期的故障。在試生產期間安全地開始使用恢復能力測試整合的同時，設定目標並在生產中實作這些測試，以做為演練日作業的一部分。

期望的結果：恢復能力測試有助於建立對系統承受生產降級能力的信心。實驗能識別可能導致故障的弱點，進而幫助您改善系統，以自動且有效率地減輕故障和降級的衝擊。

常見的反模式：

- 部署流程中缺乏可觀測性和監控性
- 依賴人力解決系統故障
- 品質分析機制不佳
- 專注於系統中已知問題，以及缺乏識別任何未知問題的實驗
- 找到故障點，但沒有將其解決
- 沒有調查結果和執行手冊的文件

建立最佳實務的優勢：整合在您的部署中的恢復能力測試有助於識別系統中的未知問題，否則這些問題會被忽視，進而導致生產中斷。在系統中識別這些未知問題可幫助您記錄調查結果、將測試整合到 CI/CD 流程中，以及製作執行手冊，藉由高效率的可重複機制來簡化緩解措施。

未建立此最佳實務時的風險暴露等級：中

實作指引

可在系統部署中整合的最常見的恢復能力測試形式，是災難復原和混沌工程。

- 在任何重大部署中包括災難復原計畫和標準作業程序 (SOP) 的更新。
- 將可靠性測試整合到您的自動化部署管道中。如 AWS Resilience Hub 等的這類服務可以整合到 CI/CD 管道中，以建立持續的恢復能力評估，這些評估作業會自動作為每個部署的一部分。
- 在 AWS Resilience Hub 中定義您的應用程式。恢復能力評估會產生程式碼片段，可協助您建立復原程序做為應用程式的 AWSSystems Manager 文件，並提供建議的 Amazon CloudWatch 監視器和警示清單。

- 一旦更新了您的 DR 計畫和 SOP，請完成災難復原測試，以確認它們是否有效。災難復原測試可協助您判斷是否可以在事件後還原系統，並恢復正常運作。您可以模擬各種災難復原策略，並識別計畫是否足以滿足您的正常運作時間需求。常見的災難復原策略包括備份和還原、指示燈、冷待機、熱待機、熱待命和主動-主動，而且這些策略在成本和複雜性方面均不相同。在災難復原測試之前，建議您定義復原時間點目標 (RTO) 和復原點目標 (RPO)，以簡化模擬策略的選擇。AWS 提供災難復原工具 (例如 AWS Elastic Disaster Recovery)，協助您開始規劃和測試。
- 混沌工程實驗會導致系統中斷，例如網路中斷和服務故障。透過模擬受控故障，可以發現系統的漏洞，同時遏止注入故障的影響。像其他策略一樣，在非生產環境中使用服務 (例如 AWS Fault Injection Service) 執行受控故障模擬，可在部署至生產之前先得到十足信心。

資源

相關文件：

- 使用恢復能力測試實驗失敗以先做好復原準備
- 利用 AWS Resilience Hub 和 AWS CodePipeline 持續評估應用程式恢復能力
- AWS 上的災難復原 (DR) 架構，第 1 部分：在雲端中復原的策略
- 使用混沌工程驗證工作負載的恢復能力
- 混沌工程的原則
- 混沌工程研討會

相關影片：

- AWS re:Invent 2020：使用混沌工程測試恢復能力
- 利用 AWS 故障注入服務提高應用程式恢復能力
- 利用 AWS Resilience Hub 準備並保護您的應用程式免受中斷

REL08-BP04 使用不可變基礎設施進行部署

不可變基礎設施是一種模式，要求在生產工作負載上不進行現場的更新、安全性修補或組態變更。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。

請遵循不可變基礎設施的部署策略，以提高工作負載部署中的可靠性、一致性和可重複性。

預期成果：使用不可變基礎設施時，就地修改不得在工作負載內執行基礎設施資源。相反地，在需要變更時，會以平行方式與現有資源一起部署新的、包含所有必要變更的更新後基礎設施資源集。此部署會自動進行驗證，如果成功，流量會逐漸轉移到新的資源集。

此部署策略適用於軟體更新、安全修補程式、基礎設施變更、組態更新和應用程式更新等。

常見的反模式：

- 對執行中的基礎設施資源實作就地變更。

建立此最佳實務的優勢：

- 提高跨環境的一致性：由於不同環境的基礎設施資源沒有差異，因此可以提高一致性並簡化測試。
- 降低組態偏移：透過將基礎設施資源更換為已知且具有版本控制的組態，可將基礎設施設定為已知、經過測試且可信的狀態，以避免組態偏移。
- 不可部分完成的可靠部署：部署不是會成功完成，就是完全沒有變更，以提高部署程序的一致性和可靠性。
- 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。
- 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。
- 增強的安全狀態：透過不允許對基礎設施進行變更，可以停用遠端存取機制 (例如 SSH)。這可減少攻擊媒介，從而改善組織的安全狀態。

未建立此最佳實務時的風險暴露等級：中

實作指引

自動化

在定義不可變基礎設施的部署策略時，建議您盡可能使用自動化，以提高可重複性並將人為錯誤的可能性降至最低。如需詳細資訊，請參閱 REL08-BP05 使用自動化部署變更和自動化安全、無人為介入的部署。

使用基礎設施即程式碼 (IaC) 時，基礎設施佈建、協同運作和部署步驟會以程式化、描述性和宣告式的方式加以定義，並儲存在原始程式碼控制系統中。利用基礎設施即程式碼可讓您更輕鬆地自動部署基礎設施，並協助您實現基礎設施不可變性。

部署模式

需要變更工作負載時，不可變基礎設施的部署策略會要求您部署一組新的基礎設施資源，包括所有必要的變更。這組新資源必須遵循推出模式，以最大程度地降低使用者所受到的影響。此部署有兩個主要策略：

金絲雀部署：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 (金絲雀) 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。您可以使用允許進行金絲雀部署的部署組態來設定 AWS CodeDeploy。

藍/綠部署：與金絲雀部署類似，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。您可以使用允許進行藍/綠部署的部署組態來設定 AWS CodeDeploy 及 AWS Elastic Beanstalk。

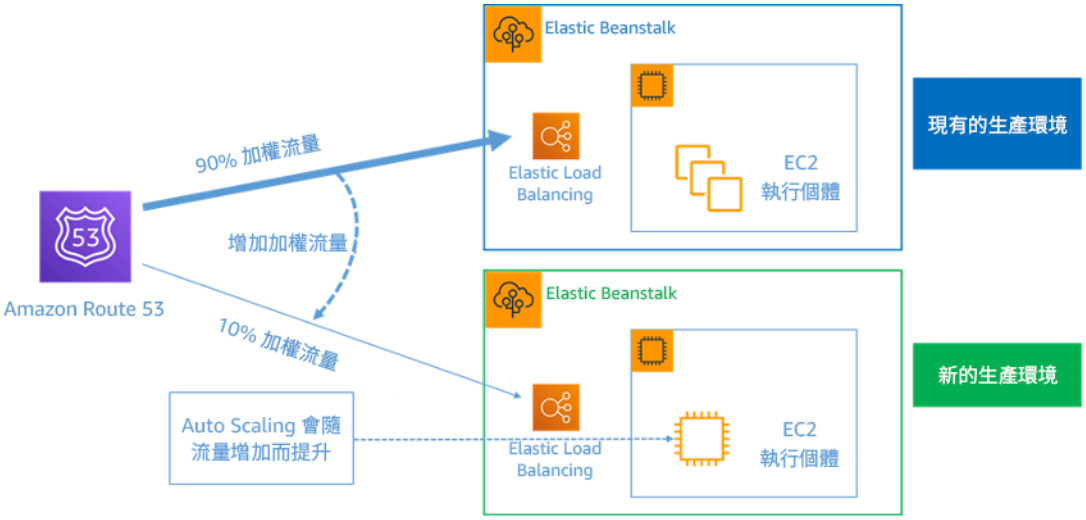


圖 8：使用 AWS Elastic Beanstalk 和 Amazon Route 53 進行藍/綠部署

偏移偵測

偏移是指會導致基礎設施資源具有與預期不同的狀態或組態的任何變更。任何類型的未受管組態變更都違反了不可變基礎設施的概念，應加以偵測並修正，以便能成功實作不可變基礎設施。

實作步驟

- 禁止就地修改執行中的基礎設施資源。
  - 您可以使用 AWS Identity and Access Management (IAM) 指定什麼人或項目可以在 AWS 中存取服務和資源、集中管理精細許可，以及分析存取動作以完善 AWS 內的許可。
- 自動部署基礎設施資源以提高可重複性，並最大限度地減少發生人為錯誤的可能性。

- 如 DevOps on AWS 簡介白皮書所述，自動化是 AWS 服務的基石，且所有服務、功能和產品內部都支援自動化。
- 預先封裝 Amazon Machine Image (AMI) 可以加快其啟動時間。EC2 Image Builder 是全受管的 AWS 服務，可協助您自動建立、維護、驗證、共用和部署自訂、安全且最新的 Linux 或 Windows 自訂 AMI。
- 支援自動化的一些服務包括：
  - AWS Elastic Beanstalk 服務可在熟悉的伺服器 (例如 Apache、NGINX、Passenger 和 IIS) 上，快速部署和擴展使用 Java、.NET、PHP、Node.js、Python、Ruby、Go 和 Docker 所開發的 Web 應用程式。
  - AWS Proton 可協助平台團隊連接並協調開發團隊為了進行基礎設施佈建、程式碼部署、監控和更新所需的所有不同工具。AWS Proton 可讓您以基礎設施即程式碼的方式，自動佈建和部署無伺服器和容器型的應用程式。
- 利用基礎設施即程式碼可讓您輕鬆地自動部署基礎設施，並協助實現基礎設施的不可變性。AWS 會提供能以程式化、描述性和宣告式的方式建立、部署和維護基礎設施的服務。
  - AWS CloudFormation 可協助開發人員以有序且可預測的方式建立 AWS 資源。資源會使用 JSON 或 YAML 格式以文字檔撰寫。範本需要特定語法和結構，而這取決於所建立和管理的資源類型。您可以使用任何程式碼編輯器 (例如 AWS Cloud9) 以 JSON 或 YAML 撰寫資源、將其簽入版本控制系統，然後 CloudFormation 便會以安全、可重複的方式建置指定的服務。
  - AWS Serverless Application Model (AWS SAM) 是開放原始碼架構，您可以用它在 AWS 上建置無伺服器應用程式。AWS SAM 會與其他 AWS 服務整合，並且是 AWS CloudFormation 的延伸。
  - AWS Cloud Development Kit (AWS CDK) 是開放原始碼的軟體開發架構，可讓您使用熟悉的程式設計語言來建模和佈建雲端應用程式資源。您可以使用 AWS CDK 透過 TypeScript、Python、Java 和 .NET 來建模應用程式基礎設施。AWS CDK 會在背景中使用 AWS CloudFormation 以透過安全、可重複的方式佈建資源。
  - AWS Cloud Control API 引入了一組常見的建立、讀取、更新、刪除和列出 (CRUDL) API，以協助開發人員以簡單且一致的方式管理其雲端基礎設施。Cloud Control API 通用 API 可讓開發人員統一管理 AWS 和第三方服務的生命週期。
- 實作可將使用者所受到的影響降到最低的部署模式。
  - 金絲雀部署：
    - 設定 API Gateway 金絲雀版本部署
    - 使用 AWS App Mesh 為 Amazon ECS 建立具有金絲雀部署的管道
  - 藍/綠部署：AWS 上的藍/綠部署白皮書會描述用來實作藍/綠部署策略的範例技術。

- 偵測組態或狀態的偏移。如需詳細資訊，請參閱偵測堆疊和資源的未受管組態變更。

資源

相關的最佳實務：

- REL08-BP05 使用自動化部署變更

相關文件：

- 自動化安全、無人為介入的部署
- 利用 AWS CloudFormation 在 Nubank 建立不可變基礎設施
- 基礎設施即程式碼
- 實作警示以自動偵測 AWS CloudFormation 堆疊中的偏移

相關影片：

- AWS re:Invent 2020：透過不可變實現可靠性、一致性和可信度

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。

期望的結果：您可以透過廣泛的試生產測試、自動回復和分期生產部署，將自動化部署安全性建置在發布流程中。此自動化作業可將部署失敗造成對生產的潛在影響降到最低，而開發人員不再需要主動監視生產的部署。

常見的反模式：

- 您執行手動變更。
- 您利用手動緊急工作流程跳過自動化作業的步驟。
- 您為了加快時間進程，而沒有遵循既定的計畫和流程。
- 您快速執行後續部署，卻不允留封裝時間。

建立此最佳實務的優勢：您使用自動化作業部署所有變更時，可以免除導入人為錯誤的可能性，並提供在變更生產前進行測試的能力。在生產推送之前執行此流程，確認您的計畫是否已完成。此外，自動回復到您的發布流程有利於找出生產問題，並將工作負載回復到先前工作的操作狀態。

未建立此最佳實務時的風險暴露等級：中

實作指引

自動化您的部署管道。部署管道讓您可以調用自動測試、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。採用持續整合及持續交付/部署 (CI/CD) 的文化，其中提交或程式碼變更會經過各種自動化階段關卡 (從建置和測試階段到生產環境部署)。

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，但是同樣出於這個原因，我們建議您將最困難的程序自動化。

實作步驟

您可以依照下列步驟自動執行部署以移除手動作業：

- 設定程式碼儲存庫以安全地儲存您的程式碼：使用 AWS CodeCommit，建立安全的 Git 型儲存庫。
- 設定持續整合服務以編譯原始程式碼、執行測試以及建立部署成品：若要為此目的設定建置專案，請參閱使用主控台開始使用 AWS CodeBuild。
- 設定部署服務以自動執行應用程式部署，並處理應用程式更新的複雜度，而不需依賴容易出錯的手動部署：AWS CodeDeploy 將軟體自動部署到各種運算服務，例如 Amazon EC2、AWS Fargate、AWS Lambda 和您的內部部署伺服器。若要設定這些步驟，請參閱開始使用 CodeDeploy。
- 設定持續交付服務，將您發布的管道自動化，以實現更快、更可靠的應用程式和基礎設施更新：考慮使用 AWS CodePipeline 來協助您自動發布管道。如需詳細資訊，請參閱CodePipeline 教學課程。

資源

相關的最佳實務：

- OPS05-BP04 使用建置和部署管理系統
- OPS05-BP10 完全自動化整合和部署
- OPS06-BP02 測試部署
- OPS06-BP04 自動化測試和復原

相關文件：

- 使用 AWS CodePipeline 連續交付巢狀 AWS CloudFormation 堆疊
- 利用 AWS CodeCommit、AWS CodeBuild、AWS CodeDeploy 和 AWS CodePipeline 完成 CI/CD
- 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 建置者資料中心：確保部署期間的回復安全
- Amazon 建置者資料中心：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：

- AWS Summit 2019：AWS 上的 CI/CD

失敗管理

問題

- REL 9.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.如何規劃災難復原 (DR)？

REL 9.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) 的需求。

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

了解和使用工作負載所使用的資料服務和資源的備份功能。大部分服務都會提供備份工作負載資料的功能。

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

常見的反模式：

- 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。

未建立此最佳實務時的風險暴露等級：高

實作指引

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。AWS Elastic Disaster Recovery 可讓您從內部部署、跨可用區域或跨區域複製完整伺服器工作負載並且維護持續資料保護，使用以秒數測量的復原點目標 (RPO)。

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。

內部部署資料可以使用 AWS Storage Gateway 或 AWS DataSync 備份到 AWS 雲端。Amazon S3 儲存貯體可以用來在 AWS 上存放此資料。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 Amazon 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 Amazon S3 將資料重現到 Amazon 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。

實作步驟

1. 識別工作負載的所有資料來源。資料可以存放在多個資源上，例如資料庫、磁碟區、檔案系統、記錄系統和物件儲存。請參閱資源區段以尋找存放資料所在之不同 AWS 服務的相關文件，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，可以在 AWS 上建立各種資料來源的備份。AWS Elastic Disaster Recovery 會處理對 AWS 區域 的自動化次秒級資料複寫。大部分 AWS 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。請參閱以下所列的資源，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，使用 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
- 什麼是 AWS Elastic Disaster Recovery？

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恢復進行備份和還原
- 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。

常見的反模式：

- 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。

未建立此最佳實務時的風險暴露等級：高

實作指引

使用身分驗證和授權控制並偵測對備份的存取，例如 AWS Identity and Access Management (IAM)。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。使用 AWS Elastic Disaster Recovery 時，所有傳輸中的資料和靜態資料都會加密。使用 Elastic Disaster Recovery，靜態資料可以使用預設 Amazon EBS 加密磁碟區加密金鑰或自訂客戶受管金鑰進行加密。

實作步驟

1. 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 Amazon RDS 中使用加密。您可以在建立 RDS 執行個體時，使用 AWS Key Management Service 設定靜態加密。
   - 在 Amazon EBS 磁碟區上使用加密。您可以在建立磁碟區時設定預設加密或指定唯一金鑰。
   - 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。
   - 加密存放在 Amazon EFS 中的資料。在建立檔案系統時設定加密。
   - 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。
   - 選擇要使用預設或自訂適用於 Elastic Disaster Recovery 的 Amazon EBS 加密。此選項會在模擬區域子網路磁碟和複寫磁碟上加密您的複寫靜態資料。
2. 實作存取備份的最低權限。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。

資源

相關文件：

- AWS Marketplace：可用於備份的產品
- Amazon EBS 加密
- Amazon S3：使用加密保護資料
- 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 Amazon 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 Framework
- 什麼是 AWS Elastic Disaster Recovery？

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。

預期成果：以建立的規律建立資料來源備份的自動化流程。

常見的反模式：

- 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。

未建立此最佳實務時的風險暴露等級：中

實作指引

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 可用於自動化建立、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。

AWS Elastic Disaster Recovery 提供從來源環境 (內部部署或 AWS) 到目標復原區域的持續區塊層級複寫。時間點 Amazon EBS 快照會由服務自動建立及管理。

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。

實作步驟

1. 識別資料來源，這是目前手動備份的資料來源。如需詳細資訊，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。
2. 針對工作負載判斷 RPO。如需詳細資訊，請參閱 REL13-BP01 定義停機和資料遺失的復原目標。
3. 使用自動化備份解決方案或受管服務。AWS Backup 是一種全受管服務，可讓您在雲端和內部部署環境輕鬆集中化和自動化 AWS 服務的資料保護。使用 AWS Backup 中的備份計劃建立規則，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。如需如何使用 AWS Backup 建立自動化備份的實作指引，請參閱測試備份並還原資料。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，以根據 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？
- 什麼是 AWS Elastic Disaster Recovery？

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)。

預期成果：使用妥善定義的機制定期復原來自備份的資料，以確認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。

常見的反模式：

- 還原備份，但不查詢或擷取任何資料，以檢查還原可用。
- 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，視需要還原。

建立此最佳實務的優勢：測試備份的復原確認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。

未建立此最佳實務時的風險暴露等級：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會檢查所有資料是否可用、未損毀、可存取，且任何資料遺失落在工作負載的 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。

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。

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。

AWS Elastic Disaster Recovery 提供 Amazon EBS 磁碟區的持續時間點復原快照。隨著來源伺服器進行複寫，時間點狀態會根據設定的政策隨著時間進行編製。Elastic Disaster Recovery 可藉由針對測試和練習目的啟動執行個體，而不重新導向流量，協助您確認這些快照的完整性。

實作步驟

1. 識別資料來源，這些資料來源目前正在備份，以及這些備份的存放位置。如需實作指引，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。
2. 針對每個資料來源建立資料驗證準則。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。如需實作指引，請參閱 REL13-BP01 定義停機和資料遺失的復原目標。
4. 評估您的復原功能。檢閱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。使用 AWS Resilience Hub，您可以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. 執行測試還原，使用在生產環境進行資料還原的目前建立程序。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，如果您是使用像是 AWS Backup 的受管服務，這可能就像是將備份還原到新資源一樣簡單。如果您使用 AWS Elastic Disaster Recovery，您可以啟動復原練習。
6. 從還原的資源驗證資料復原，根據您先前為資料驗證建立的準則。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. 測量還原和復原以還原和復原，並且與您的已建立 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 之類的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時，例如在此實驗室中，像是 Amazon Simple Notification Service (Amazon SNS) 之類的服務可以用來將推送通知 (例如電子郵件或簡訊) 傳送給利害關係人。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 建立例如 OpsItems 的任務。

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。進一步了解如何使用 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。

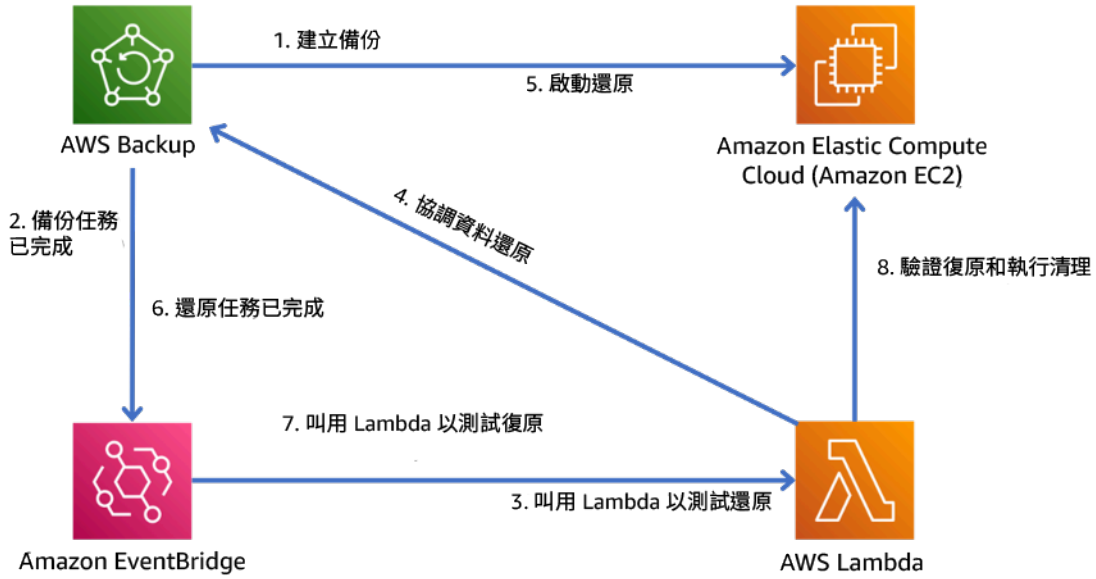


圖 9.自動的備份和還原程序

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 Elastic Disaster Recovery
- AWS Elastic Disaster Recovery

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。

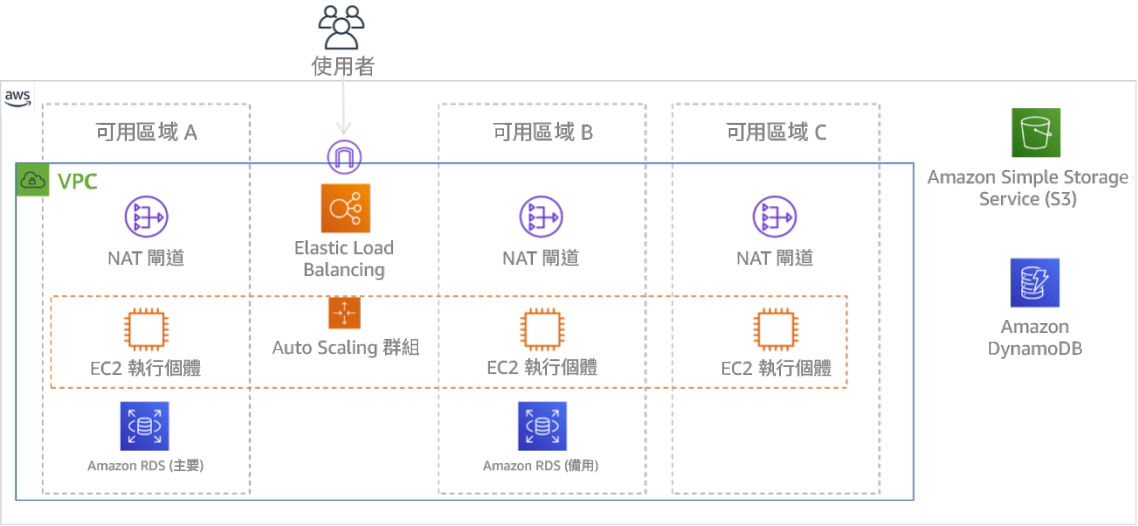


圖 9：跨三個可用區域部署的多層架構。請注意，Amazon S3 和 Amazon DynamoDB 一律自動採用異地同步備份策略。ELB 也會部署至全部三個區域。

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。

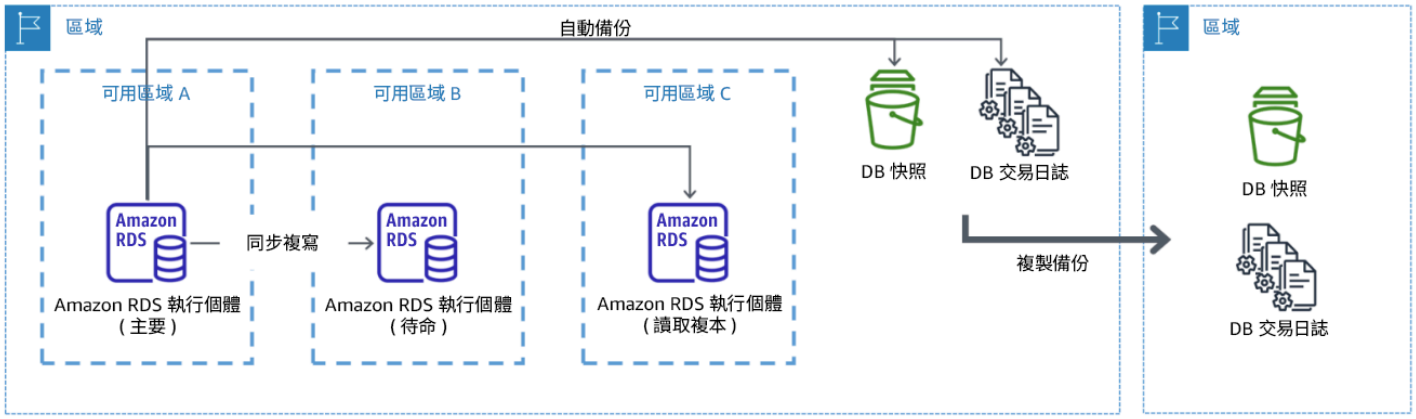


圖 10：備份至另一個 AWS 區域的彈性異地同步備份資料庫部署

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，在定義的復原目標內實作完整重建工作負載的功能。

未建立此最佳實務時的風險暴露等級：中

實作指引

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。

針對部署到內部部署資料中心的有狀態的伺服器型工作負載，您可以使用 AWS Elastic Disaster Recovery 在 AWS 中保護您的工作負載。如果您已在 AWS 中託管，您可以使用 Elastic Disaster Recovery 將工作負載保護到替代可用區域或區域。Elastic Disaster Recovery 會使用持續區塊層級複寫到輕量型模擬區域，提供內部部署和雲端式應用程式的快速、可靠復原。

實作步驟

1. 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 Amazon EC2 Auto Scaling 群組。
     - 啟動範本使用者資料可用於實現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 Amazon EC2 執行個體的自動復原。
     - 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 Amazon EC2 執行個體生命週期事件或 Amazon ECS 事件自動執行自我修復。
     - 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。
   - 保護使用 AWS Elastic Disaster Recovery 限制為單一位置的有狀態的工作負載。

資源

相關文件：

- Amazon ECS 事件
- Amazon EC2 Auto Scaling lifecycle hook
- 復原您的執行個體

- 服務自動擴展
- 什麼是 Amazon EC2 Auto Scaling？
- AWS Elastic Disaster Recovery

REL10-BP04 使用隔板架構限制影響範圍

實作隔板架構 (也稱為小組型架構) 將工作負載內的失敗效應限制為有限數量的元件。

預期成果：小組型架構會使用工作負載的隔離執行個體，其中每個執行個體稱為小組。每個小組都是獨立的，不會與其他小組共用狀態，並且處理整體工作負載請求的子集。這會對個別小組和它處理的請求降低失敗的潛在影響，例如不良的軟體更新。如果工作負載使用 10 個小組為 100 個請求提供服務，發生失敗時，整體請求中 90% 不會受到失敗影響。

常見的反模式：

- 允許小組成長，沒有界限。
- 將程式碼更新或部署同時套用到所有小組。
- 在小組之間共用狀態或元件 (路由器層例外)。
- 將複雜商業或路由邏輯新增至路由器層。
- 不將跨小組互動降至最低。

建立此最佳實務的優勢：使用小組型架構，許多常見類型的失敗會包含在小組本身，提供額外的故障隔離。這些故障界限可以提供對於難以包含之失敗類型的彈性，例如失敗的程式碼部署或已損毀或觸發特定失敗模式的請求 (也稱為毒藥請求)。

實作指引

在船上，隔板可確保船體破口包含在船體的其中一個區段內。在複雜的系統中，通常會複寫這個模式以啟用故障隔離。故障隔離界限會在工作負載內將失敗影響限制為有限數量的元件。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。在 AWS 上，客戶可以使用多個可用區域或區域來提供故障隔離，但是故障隔離的概念也可以延伸為您的工作負載的架構。

整體工作負載是依分割區索引鍵的分割區小組。這個索引鍵需要與服務的精細度保持一致，否則服務的工作負載會自然地透過最小跨小組互動進行細分。分割區索引鍵的範例為客戶 ID、資源 ID 或可在大部分 API 呼叫中輕易存取的其他任何參數。小組路由層會根據分割區索引鍵將請求分散到個別小組，並且對用戶端呈現單一端點。

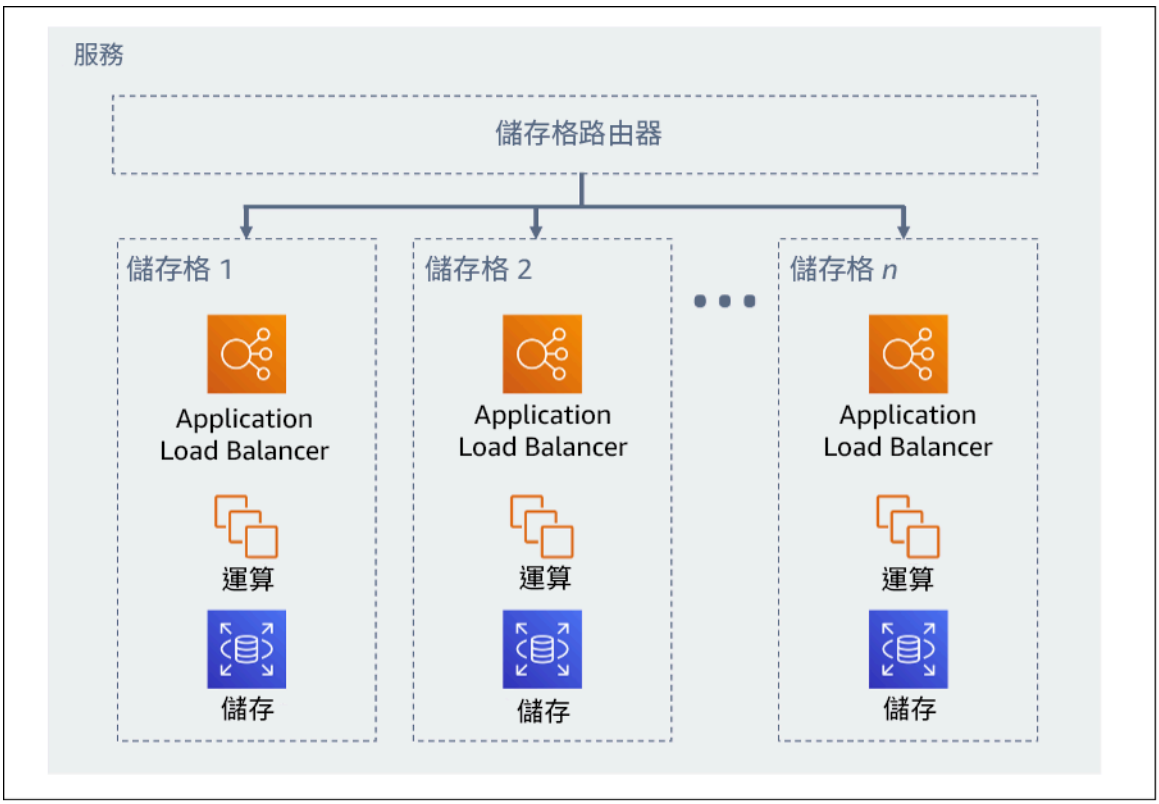


圖 11：小組型架構

實作步驟

設計小組型架構時，有數個設計考慮要考慮：

1. 分割區索引鍵：選擇分割區索引鍵時應該考慮的特殊考慮。
   - 應該與服務的精細度保持一致，否則服務的工作負載會自然地透過最小跨小組互動進行細分。範例為 ## ID 或 ## ID。
   - 分割區索引鍵必須在所有請求中都可供使用，無論是直接或由其他參數確定性地推斷。
2. 持續性小組對應：上游服務在其資源的生命週期中應該只與單一小組互動。
   - 依據工作負載而定，可能需要小組遷移策略，以便從其中一個小組將資料遷移到另一個小組。可能需要小組遷移的可能情境是，如果您的工作負載中特定使用者或資源變得太大並且要求它具備專有小組。
   - 小組不應該在小組之間共用狀態或元件。
   - 因此，應該避免跨小組互動或保持在最低程度，因為這些互動會建立小組之間的相依性，因而消滅故障隔離改善。
3. 路由器層：路由器層會在小組之間共用元件，因此無法遵循與小組相同的區隔策略。

- 建議路由器層以有效率運算的方式使用分割區對應演算法將請求分發到個別小組，例如結合加密雜湊函數和模組化算術以將分割區索引鍵對應至小組。
- 若要避免多小組影響，路由層必須保持簡單並且盡可能水平擴展，如此才能避免此層級內的複雜商業邏輯。這樣有增加的優點，隨時都容易了解其預期行為，以獲得徹底的可測試性。如同 Colm MacCárthaigh 在可靠性、持續工作，以及咖啡時刻中所說明，簡單設計和持續工作模式可產生可靠的系統並且降低抗脆弱性。

4. 小組大小：小組應該有最大大小，而且不應該允許成長超出此大小。
   - 最大大小應該藉由執行徹底測試來識別，直到觸及中斷點並且建立安全的操作邊距。如需如何實作測試實務的詳細資訊，請參閱 REL07-BP04 對工作負載執行負載測試
   - 整體工作負載應該透過新增額外小組來成長，讓工作負載隨著需求的增加而擴展。
5. 多可用區域或多區域策略：應該利用彈性的多個層次來保護不同的失敗網域。
   - 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。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。如需詳細資訊，請參閱 REL10-BP02 為您的多位置部署選取適當位置。
6. 程式碼部署：應該偏向交錯程式碼部署策略，而不是將程式碼變更同時部署到所有小組。
   - 這樣可協助將多個小組由於不良部署或人為錯誤的潛在失敗降至最低。如需詳細資訊，請參閱自動化安全、無人為介入的部署。

未建立此最佳實務時的風險暴露等級：高

資源

相關的最佳實務：

- REL07-BP04 對工作負載執行負載測試
- REL10-BP02 為您的多位置部署選取適當位置

相關文件：

- 可靠性、持續工作，以及咖啡時刻
- AWS 和區隔
- 使用隨機切換分區隔離工作負載

- 自動化安全、無人為介入的部署

相關影片：

- AWS re:Invent 2018：閉環與開放思維：如何取得大小型系統的控制權
- 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)
- AWS Summit ANZ 2021 - 所有事情都一直失敗：設計彈性

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？

須架構具高可用性和低平均復原時間 (MTTR) 需求的工作負載以實現彈性。

最佳實務

- REL11-BP01 監控工作負載的所有元件以偵測故障
- 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-BP07 建立您的產品架構以符合可用性目標和運行時間服務水準協議 (SLA)

REL11-BP01 監控工作負載的所有元件以偵測故障

持續監控工作負載的運作狀態，讓您和自動化系統在發生故障或效能降低時能夠察覺。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。

預期成果： 工作負載的基本元件會單獨監控，以偵測故障發生的時機和位置並發出警示。

常見的反模式：

- 未設定任何警報，因此會在未發出通知的情況下發生中斷。
- 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)。
- 只主動監控面對客戶的工作負載介面。
- 只收集技術指標，未收集業務功能指標。
- 無測量工作負載使用者體驗的指標。
- 建立了太多監控。

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。

未建立此最佳實務時的曝險等級： 高

實作指引

確定將要檢閱以進行監控的所有工作負載。確定需要監控的所有工作負載元件之後，您現在需要確定監控間隔。根據偵測故障所需的時間而定，監控間隔會直接影響復原的速度。平均偵測時間 (MTTD) 是指從發生故障到開始修復作業經過的時間。服務清單應盡可能廣泛且完整。

監控必須涵蓋應用程式堆疊的所有層級，包括應用程式、平台、基礎設施和網路。

您的監控策略應考慮 微小故障的影響。如需微小故障的詳細資訊，請參閱 微小故障 (於《進階多可用區域彈性模式》白皮書中)。

實作步驟

- 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。
- 設定元件和受管服務的詳細監控。
  - 確定是否需要對 EC2 執行個體 和 Auto Scaling 進行詳細監控。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。
  - 確定對 RDS 的 增強型監控 是否必要。增強型監控使用 RDS 執行個體上的代理程式，以取得不同處理程序或執行緒的實用資訊。
  - 判斷以下各項的關鍵無伺服器元件的監控需求： Lambda、 API Gateway、 Amazon EKS、 Amazon ECS，以及所有類型的 負載平衡器。

  - 判斷以下各項的儲存元件的監控需求： Amazon S3、 Amazon FSx、 Amazon EFS和 Amazon EBS。
- 建立 自訂指標 來測量業務關鍵績效指標 (KPI)。工作負載會實作重要的業務功能，這些功能應做為 KPI，以利確定間接問題發生的時間。
- 以使用者 Canary 監控使用者的故障體驗 綜合交易測試 (也稱為 Canary 測試，但請別與金絲雀部署混淆)，可執行和模擬客戶行為，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。
- 建立 自訂指標 來追蹤使用者體驗。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。
- 設定警報 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。警報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，以及搭配使用 Auto Scaling 向上擴展或縮減工作負載資源。
- 建立 儀表板 以視覺化呈現您的指標。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，或指出您可能想要調查的問題。
- 建立 分散式追蹤監控 來監控您的服務。透過分散式監控，您可以了解應用程式及其基礎服務的執行方式，以確定和疑難排解效能問題與錯誤的根本原因。
- 建立監控系統 (使用 CloudWatch 或者 X-Ray) 儀表板，以及在個別區域和帳戶中進行資料收集。
- 建立 Amazon Health Aware 監控的整合，以監控可能降級的 AWS 資源。針對商務基本工作負載，此解決方案可讓您存取 AWS 服務的主動式即時警示。

資源

相關的最佳實務：

- 可用性定義
- REL11-BP06 當事件影響可用性時傳送通知

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
- 增強型監控
- 使用 Amazon CloudWatch 監控您的 Auto Scaling 群組和執行個體
- 發佈自訂指標
- 使用 Amazon CloudWatch 警報

- 使用 CloudWatch 儀表板
- 使用跨區域跨帳戶 CloudWatch 儀表板
- 使用跨區域跨帳戶 X-Ray 追蹤
- 了解可用性
- 實作 Amazon Health Aware (AHA)

相關影片：

- 減少微小故障

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
- One Observability 研討會：探索 X-Ray

相關工具：

- CloudWatch
- CloudWatch X-Ray

REL11-BP02 容錯移轉至運作良好的資源

如果發生資源失敗，運作良好的資源應繼續處理請求。對於位置受損 (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。

設計服務時，請將負載分散到各個資源、可用區域或區域。因此，可以透過將流量轉移到剩餘運作狀態良好的資源來減輕個別資源故障或損害的影響。請考慮發生故障時，如何找到服務及其路由。

設計服務時，務必考慮故障復原。在 AWS，我們設計服務以盡可能減少從故障復原的時間並減輕對資料的影響。我們的服務主要使用的資料存放區，會在請求持久儲存於區域內的多個複本中之後，才確認請求。經過建構後，它們會使用以儲存格為基礎的隔離，以及使用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。我們還將取代-重啟功能最佳化，以期從中斷快速復原。

允許容錯移轉的模式和設計會隨著各 AWS 平台服務而有所不同。許多 AWS 原生受管服務本身就是多個可用區域 (例如 Lambda 或 API Gateway)。其他 AWS 服務 (例如 EC2 和 EKS) 需要特定的最佳實務設計，以支援在 AZ 的各資源或資料儲存容錯移轉。

監控應設定為確認容錯移轉資源是否正常運作、追蹤資源容錯移轉的進度，以及監控業務程序復原。

預期成果： 系統能夠自動或手動使用新資源，以從降級恢復。

常見的反模式：

- 故障計畫不是規劃和設計階段的一部分。
- 未建立 RTO 和 RPO。
- 監控不足，無法偵測出失敗的資源。
- 正確隔離故障網域。
- 未考慮多區域容錯移轉。
- 決定進行容錯移轉時，失敗偵測太過敏感或積極。
- 未測試或驗證容錯移轉設計。
- 進行自動修復自動化，但未通知需要修復。
- 缺少緩衝期，以避免過早容錯恢復。

建立此最佳實務的優勢： 您可以建置更具彈性的系統，在發生故障時透過適當降級並快速復原來維持可靠性。

未建立此最佳實務時的曝險等級： 高

實作指引

AWS 服務，例如 Elastic Load Balancing 和 Amazon EC2 Auto Scaling，會協助各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，設計會更複雜。例如，跨區域僅供讀取複本可讓您將資料部署到多個 AWS 區域。不過仍需要容錯移轉，才能將僅供讀取複本提升為主要複本，然後將流量指向新端點。Amazon Route 53、Route 53 Route 53 ARC、CloudFront 和 AWS Global Accelerator 可協助路由 AWS 區域的各流量。

AWS 服務 (例如 Amazon S3、Lambda、API Gateway、Amazon SQS、Amazon SNS、Amazon SES、Amazon Pinpoint、Amazon ECR、AWS Certificate Manager、EventBridge 或 Amazon DynamoDB) 會由 AWS 自動部署到多個可用區域。如果發生故障，這些 AWS 服務會自動將流量路由到運作良好的位置。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、Amazon Aurora、Amazon Redshift、Amazon EKS 或 Amazon ECS，多可用區域是組態選項。如果啟動容錯移轉，AWS 可將流量導向運作良好的執行個體。此容錯移轉動作可由 AWS 執行，或依客戶要求執行

對於 Amazon EC2 執行個體、Amazon Redshift、Amazon ECS 任務或 Amazon EKS Pod，您可以選擇要部署到哪個可用區域。對於某些設計，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 也可將流量路由至內部部署資料中心內的元件。

對於多區域流量容錯移轉，重新路由可利用 Amazon Route 53、Route 53 ARC、AWS Global Accelerator、Route 53 Private DNS for VPCs 或 CloudFront 來提供定義網際網路網域和指派路由政策 (包括運作狀態檢查) 的方法，以便將流量路由到運作狀態良好的區域。AWS Global Accelerator 提供靜態 IP 地址，做為應用程式端點的固定進入點，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。

實作步驟

- 為所有適當的應用程式和服務建立容錯移轉設計。隔離每個架構元件，並為每個元件建立符合 RTO 和 RPO 的容錯移轉設計。
- 設定較低的環境 (例如開發或測試)，且其中所有服務都需要有容錯移轉計畫。使用基礎設施即程式碼 (IaC) 來部署解決方案，以確保可重複性。
- 設定復原站台 (例如第二個區域)，以實作和測試容錯移轉設計。如有必要，可以臨時設定測試的資源，以限制額外的成本。
- 判斷哪些容錯移轉計畫是由 AWS 自動執行、哪些可由 DevOps 程序自動執行，以及哪些可能要手動執行。記錄並測量每一項服務的 RTO 和 RPO。
- 建立容錯移轉程序手冊，並包括容錯移轉每個資源、應用程式和服務的所有步驟。
- 建立容錯恢復程序手冊，並包括容錯恢復 (含時程) 每個資源、應用程式和服務的所有步驟
- 制定計畫來啟動和演練程序手冊。使用模擬和混亂測試來測試程序手冊的步驟和自動化。
- 對於位置受損 (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。在容錯移轉測試之前，檢查配額、自動擴展層級和執行的資源。

資源

相關 Well-Architected 的最佳實務：

- REL13- 災難復原 (DR) 計畫

- REL10 - 使用故障隔離來保護您的工作負載

相關文件：

- 設定 RTO 和 RPO 目標
- 使用應用程式負載平衡器設定 Route 53 ARC
- 使用 Route 53 加權路由進行容錯移轉
- 透過 Route 53 ARC 進行災難復原
- 具有自動擴展的 EC2
- EC2 部署 - 多可用區域
- ECS 部署 - 多可用區域
- 使用 Route 53 ARC 切換流量
- 具有 Application Load Balancer 和容錯移轉的 Lambda
- ACM 複寫和容錯移轉
- 參數存放區複寫和容錯移轉
- ECR 跨區域複寫和容錯移轉
- Secrets Manager 跨區域複寫組態
- 啟用跨區域複寫以進行 EFS 和容錯移轉
- EFS 跨區域複寫和容錯移轉
- 聯網容錯移轉
- 使用 MRAP 的 S3 端點容錯移轉
- 為 S3 建立跨區域複寫
- 容錯移轉區域 API Gateway 與 Route 53 ARC
- 使用多區域 Global Accelerator 進行容錯移轉
- 透過 DRS 進行容錯移轉
- 使用 Amazon Route 53 建立災難復原機制

相關範例：

- AWS 上的災難復原
- AWS 上的彈性災難復原

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。降級可能透過內部服務機制自動修復，或需要透過矯正動作重新啟動或移除資源。

對於自我管理的應用程式和跨區域修復，復原的設計和自動修復程序可從 現有的最佳實務取得。

重新啟動或移除資源是修復故障的重要工具。最佳實務是盡可能讓服務無狀態。這可防止資源重新啟動時遺失資料或可用性。在雲端，您可以 (且通常應該) 在重新啟動時取代整個資源 (例如，運算執行個體或無伺服器函數)。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。

重新啟動或重試也適用於網路請求。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。

預期成果： 自動執行動作來矯正錯誤偵測。

常見的反模式：

- 佈建資源，但無自動擴展。
- 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。
- 未自動化資料庫容錯移轉。
- 缺乏自動化方法可將流量重新路由至新端點。
- 沒有儲存複寫。

建立此最佳實務的優勢： 自動修復可減少您的平均復原時間，並提高可用性。

未建立此最佳實務時的曝險等級： 高

實作指引

Amazon EKS 或其他 Kubernetes 服務的設計應包括最小和最大複本或有狀態的集合，以及最小叢集和節點群組規模調整。這些機制提供了最少量的連續可用處理資源，同時會使用 Kubernetes 控制平面自動修復任何失敗。

透過使用運算叢集的負載平衡器存取的設計模式應利用 Auto Scaling 群組。Elastic Load Balancing (ELB) 會自動將傳入的應用程式流量分配到一或多個可用區域 (AZ) 中的多個目標和虛擬設備。

未使用負載平衡的叢集式運算設計，其大小設計應考量至少遺失一個節點。這可讓服務在復原新節點的同時，維持在可能減少的容量中自行執行。服務範例包括 Mongo、DynamoDB Accelerator、Amazon Redshift、Amazon EMR、Cassandra、Kafka、MSK-EC2、Couchbase、ELK 和 Amazon OpenSearch Service。其中許多服務都可以設計為納入額外的自動修復功能。某些叢集技術必須在節點遺失時產生警示，才能觸發自動或手動工作流程來重新建立新節點。此工作流程可以使用 AWS Systems Manager 自動化，以快速修復問題。

Amazon EventBridge 可用來監控並篩選事件，例如 CloudWatch 警報或其他 AWS 服務的狀態變更。根據事件資訊，它可以接著調用 AWS Lambda、Systems Manager 自動化或其他目標，以便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。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。

實作步驟

- 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，並新增或移除容量。
- 對於先前提及的運算執行個體，使用 負載平衡 並選擇適當的負載平衡器類型。
- 考慮 Amazon RDS 的修復。使用待命執行個體，設定 自動容錯移轉 至待命執行個體。對於 Amazon RDS 僅供讀取複本，須有自動化工作流程才能將僅供讀取複本設為主要。
- 對於 如果無法將 EC2 執行個體上 已部署的應用程式部署到多個位置，且可以容忍失敗後重新開機，則進行自動復原。無法將應用程式部署到多個位置時，自動復原可以用來取代失敗的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 EBS 磁碟區 和下列掛載點皆會保留：Amazon Elastic File System 或 Lustre 和 Windows的檔案系統。使用 AWS OpsWorks可在層級中設定 EC2 執行個體的自動修復功能。
- 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 進行自動復原。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。
- Amazon EventBridge 可用來監控並篩選事件，例如 CloudWatch 警報 或其他 AWS 服務的狀態變更。根據事件資訊，它接著可以調用 AWS Lambda (或其他目標)，在您的工作負載上執行自訂修復邏輯。

資源

相關的最佳實務：

- 可用性定義
- REL11-BP01 監控工作負載的所有元件以偵測故障

相關文件：

- AWS Auto Scaling 的運作方式
- 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？
- AWS OpsWorks：使用自動修復來替換故障的執行個體
- 什麼是 AWS Step Functions？
- 什麼是 AWS Lambda？
- 什麼是 Amazon EventBridge？
- 使用 Amazon CloudWatch 警報
- Amazon RDS 容錯移轉
- SSM - Systems Manager 自動化
- 彈性架構最佳實務

相關影片：

- 自動佈建和擴展 OpenSearch Service
- 自動 Amazon RDS 容錯移轉

相關範例：

- Auto Scaling 研討會
- Amazon RDS 容錯移轉研討會

相關工具：

- CloudWatch
- CloudWatch X-Ray

REL11-BP04 復原期間需使用資料平面，而非控制平面

控制平面提供的管理 API 適用於建立、讀取和描述、更新、刪除和列出 (CRUDL) 資源，而資料平面則處理日常服務流量。對可能影響彈性的事件實作復原或緩解回應時，請盡量使用最少數量的控制平面操作來復原、重新擴展、還原、修復或容錯移轉服務。資料平面動作應取代這些降級事件期間的任何活動。

例如，以下全都是控制平面動作：啟動新的運算執行個體、建立區塊儲存，以及說明佇列服務。啟動運算執行個體時，控制平面必須執行多項工作，例如尋找具有容量的實體主機、配置網路介面、準備本機區塊儲存磁碟區、產生憑證，以及新增安全規則。控制平面往往是複雜的協同運作。

預期成果： 當資源進入受損狀態時，系統能夠將流量從受損資源轉移到健康狀況良好的資源，來自動或手動復原。

常見的反模式：

- 依賴變更 DNS 記錄來重新路由流量。
- 依賴控制平面擴展操作來取代因佈建資源不足而受損的元件。
- 依靠大量、多服務、多 API 的控制平面動作來修復任何類別的損害。

建立此最佳實務的優勢： 提高自動化修復的成功率可減少平均復原時間，並改善工作負載的可用性。

未建立此最佳實務時的曝險等級： 中：對於某些類型的服務降級，控制平原會受到影響。若倚賴大量使用控制平面來進行修復，可能會增加復原時間 (RTO) 和平均復原時間 (MTTR)。

實作指引

若要限制資料平面動作，請評估每一項服務還原時所需的動作。

利用 Amazon Route 53 Application Recovery Controller 轉移 DNS 流量。這些功能會持續監控應用程式從失敗中復原的功能，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。

Route 53 路由政策使用控制平面，因此不要依賴它進行復原。Route 53 資料平面會答覆 DNS 查詢，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) 所設計。

您在其中建立、更新和刪除 Route 53 資源的 Route 53 管理 API 和主控台，是在控制平面上執行，這些控制平面的設計旨在優先考慮您在管理 DNS 時所需的強大一致性和耐久性。為了實現此目標，控制平面位於單一區域中：美國東部 (維吉尼亞北部)。儘管這兩個系統都建置得非常可靠，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。

對於 Amazon EC2，請使用靜態穩定性設計來限制控制平面動作。控制平面動作包括個別或使用 Auto Scaling 群組 (ASG) 縱向擴展資源。為獲得最高層級的彈性，請在用於容錯移轉的叢集中佈建足夠的容量。如果必須限制此容量閾值，請對整體端對端系統設定節流，以安全地限制總流量達到所限制的資源集。

對於像是 Amazon DynamoDB、Amazon API Gateway、負載平衡 和 AWS Lambda 無伺服器等服務，使用這些服務會利用資料平面。不過，建立新功能、負載平衡器、API 閘道或 DynamoDB 資料表是控制平面動作，應在降級前完成，以準備進行事件和容錯移轉動作的演練。對於 Amazon RDS，資料平面動作允許存取資料。

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訊，請參閱 使用可用區域實現靜態穩定性。

了解哪些作業位於資料平面，哪些位於控制平面。

實作步驟

針對需要在降級事件之後還原的每個工作負載，評估容錯移轉執行手冊、高可用性設計、自動修復設計，或 HA 資源還原計畫。找出可能視為控制平面動作的每個動作。

考慮將控制動作變更為資料平面動作：

- Auto Scaling (控制平面) 與預先擴展 Amazon EC2 資源 (資料平面) 的比較
- 遷移至 Lambda 及其擴展方法 (資料平面) 或 Amazon EC2 ASG (控制平面)
- 使用 Kubernetes 評估任何設計，以及控制平面動作的性質。新增 Pod 是 Kubernetes 中的資料平面動作。動作應限於新增 Pod 而不是新增節點。使用 過度佈建的節點 是限制控制平面動作的慣用方法

請考慮可讓資料平面動作影響相同修復措施的替代方法。

- Route 53 記錄變更 (控制平面) 或 Route 53 ARC (資料平面)
- Route 53 運作狀態檢查以進行更多自動化更新

如果服務具任務關鍵性，請考慮次要區域中的某些服務，以便在未受影響的區域中執行更多控制平面和資料平面動作。

- 主要區域中的 Amazon EC2 Auto Scaling 或 Amazon EKS 與次要區域中的 Amazon EC2 Auto Scaling 或 Amazon EKS 比較，並將流量路由到次要區域 (控制平面動作)
- 將僅供讀取複本設為主要，或在主要區域中嘗試相同的動作 (控制平面動作)

資源

相關的最佳實務：

- 可用性定義
- REL11-BP01 監控工作負載的所有元件以偵測故障

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 Elemental MediaStore 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
- Kubernetes 控制平面和資料平面

相關影片：

- 回歸基礎 - 使用靜態穩定性
- 使用 AWS 全球服務建置彈性的多站點工作負載

相關範例：

- 簡介 Amazon Route 53 應用程式復原控制器
- Amazon Builders' Library：控管較小服務，避免分散式系統過載
- 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
- 使用可用區域實現靜態穩定性

相關工具：

- Amazon CloudWatch
- AWS X-Ray

REL11-BP05 使用靜態穩定性來防止雙模態行為

工作負載應該是靜態穩定的，且只在單一正常模式下運作。雙模態行為是指工作負載在正常和故障模式下呈現不同行為的情況。

例如，您可能在不同的可用區域中啟動新的執行個體，嘗試回復可用區域故障。這可能會導致在故障模式期間產生雙模態回應。您應改為建置靜態穩定且僅以一種模式操作的工作負載。在此範例中，這些執行個體應該在發生故障之前已佈建在第二個可用區域。此靜態穩定設計可以確保工作負載僅在單一模式下運作。

預期成果： 工作負載不會在正常和故障模式出現雙模態行為。

常見的反模式：

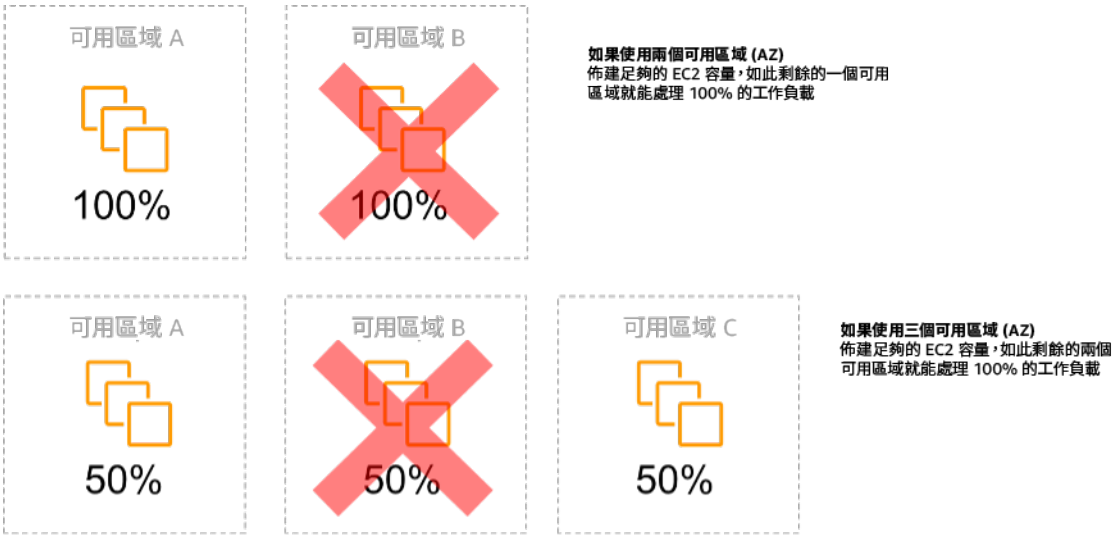
- 假設無論故障範圍，一律可以佈建資源。
- 嘗試在故障期間動態取得資源。
- 在發生故障之前，請勿在多個區域佈建適度的資源。
- 僅考慮運算資源的靜態穩定設計。

建立此最佳實務的優勢： 使用靜態穩定設計執行的工作負載，能夠在正常和故障事件發生時產生可預測的結果。

未建立此最佳實務時的曝險等級： 中

實作指引

雙模態行為是指您的工作負載在正常和故障模式下展現出不同的行為，例如，當可用區域故障時，仰賴啟動新的執行個體。例如在雙模態行為下，如果移除一個可用區域，則當靜態 Amazon EC2 設計在每個可用區域佈建足夠的執行個體來處理工作負載時，系統會執行 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。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。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。



在多個可用區域之 EC2 執行個體的靜態穩定性

這必須在所有彈性情況下，與此模型的成本以及維護工作負載的商業價值互相衡量。佈建較少運算容量並在故障時啟動新執行個體的成本較低，但是對於大規模故障 (例如可用區域損壞)，這種方法的效率較低，因為它同時仰賴作業平面，以及未受影響區域中的足夠資源。

您的解決方案也應該權衡可靠性與工作負載的成本需求。靜態穩定架構適用於多種架構，包括在多個可用區域的運算執行個體、資料庫僅供讀取複本設計、Kubernetes (Amazon EKS) 叢集設計，以及多區域容錯移轉架構。

若在每個區域使用更多資源，也可以實施更靜態的穩定設計。透過新增更多區域，您可以降低靜態穩定性所需的額外運算量。

雙模態行為範例之一是網路逾時，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，您可以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，並啟動警示。

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，但會大幅變更工作負載的需求，且可能導致故障。

評估關鍵工作負載，決定哪些工作負載需要此類彈性設計。針對關鍵工作負載，必須檢視每個應用程式元件。需要靜態穩定性評估的服務類型範例如下：

- 運算：Amazon EC2、EKS-EC2、ECS-EC2、EMR-EC2
- 資料庫：Amazon Redshift、Amazon RDS、Amazon Aurora
- 儲存：Amazon S3 (單一區域)、Amazon EFS (掛載)、Amazon FSx (掛載)
- 負載平衡器： 特定設計之下

實作步驟

- 建置靜態穩定且僅以一種模式操作的系統。在此情況下，請在每個可用區域佈建足夠的執行個體，以處理移除一個可用區域時的工作負載容量。許多服務皆可用於路由到運作狀態良好的資源，例如：
  - 跨區域 DNS 路由
  - MRAP Amazon S3 多區域路由
  - AWS Global Accelerator
  - Amazon Route 53 Application Recovery Controller
- 設定 資料庫讀取複本 以說明遺失單一主要執行個體或僅供讀取複本。若僅供讀取複本為流量提供服務，則每個可用區域中的數量應等同於區域故障時的整體需求。
- 在 Amazon S3 儲存中設定重要資料，以便可用區域故障時，能針對所儲存的資料保持靜態穩定。若 Amazon S3 單區域 – IA 儲存類別，則不應將其視為靜態穩定，因為該區域的遺失會最小化此儲存資料的存取權。
- 負載平衡器 有時會設定錯誤，或本來就設定為供特定可用區域使用。在這種情況下，靜態穩定設計可能是在更複雜的設計中將工作負載分散到多個可用區域。出於安全性、延遲或成本考量，可以使用原始設計來減少區域間流量。

資源

相關 Well-Architected 的最佳實務：

- 可用性定義
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP04 復原期間需使用資料平面，而非控制平面

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
- 故障隔離界線
- 使用可用區域實現靜態穩定性
- 多區域 RDS
- 在災難復原計畫中盡可能減少相依關係
- 跨區域 DNS 路由
- MRAP Amazon S3 多區域路由
- AWS Global Accelerator
- Route 53 ARC
- 單一區域 Amazon S3
- 跨區域負載平衡

相關影片：

- AWS 中的靜態穩定性：AWS re:Invent 2019：Amazon 建置者資料中心簡介 (DOP328)

相關範例：

- Amazon Builders' Library：使用可用區域實現靜態穩定性

REL11-BP06 當事件影響可用性時傳送通知

當偵測到閥值超標時傳送通知，即使問題造成的事件已自動解決。

自動修復功能可讓您的工作負載變得可靠。不過，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)，以解決根本原因問題。

具有韌性的系統可將降級事件立即傳達給權責團隊。這些通知應該透過一個或多個通訊管道傳送。

預期成果： 當超過閾值 (例如錯誤率、延遲或其他關鍵績效指標 (KPI)) 時，營運團隊會立即收到警示，以盡快解決問題，避免或將使用者負面影響降至最低。

常見的反模式：

- 傳送太多警示。
- 傳送不可採取行動的警示。
- 警示閾值設置太高 (太敏感) 或太低 (太遲鈍)。
- 不傳送外部相依性的警示。
- 不考慮 微小故障的影響 (在設計監控和警示時)。
- 進行修復自動化，但不通知權責團隊需要修復。

建立此最佳實務的優勢： 回復通知可讓營運和業務團隊注意到服務降級，讓他們可以立即反應，將平均偵測時間 (MTTD) 和平均復原時間 (MTTR) 降至最低。回復事件的通知也會確認您不會忽略不常發生的問題。

未建立此最佳實務時的曝險等級： 中。若無法實作適當的監控和事件通知機制，您可能就無法偵測到問題模式 (包括自動修復功能處理的問題模式)。只有當使用者聯絡客服或偶然情況下，團隊才會注意到系統降級。

實作指引

定義監控策略時，觸發警示是常見的事件。此事件可能包含警示的識別碼、警示狀態 (例如 ### 或 ##)，以及觸發警示的詳細資訊。在許多情況下，系統應檢測到警示事件並傳送電子郵件通知。這是警示動作範例。警示通知對於可觀測性至關重要，因為它會通知權責人員有問題發生。然而，當可觀測性解決方案對事件的回應措施夠熟練後，便可以自動修復問題，無需人為介入。

建立 KPI 監控警示後，閾值超過時就應會向權責團隊傳送警示。這些警示也可用於觸發嘗試修復降級的自動化程序。

針對更複雜的閾值監控，則應考慮使用複合警示。複合警示會使用數個 KPI 監控警示，根據作業商務邏輯建立警示。CloudWatch 警示可設定為傳送電子郵件，或使用 Amazon SNS 整合或 Amazon EventBridge 在第三方事件追蹤系統中記錄事件。

實作步驟

根據監控工作負載的方式建立各種警示類型，例如：

- 應用程式警示可用來偵測工作負載任何無法正常運作的部分。
- 基礎架構警示 指示何時擴展資源。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，以及搭配使用 Auto Scaling 水平擴展或縮減工作負載資源。
- 簡單 靜態警示 經過建立之後，將會監控指標在指定評估期間內超過靜態閾值的時間。

- 複合警示 可以涵蓋來自多個來源的複雜警示。
- 建立警示後，請建立適當的通知事件。您可以直接調用 Amazon SNS API 來傳送通知，並連結任何自動化程序以進行補救或通訊。
- 整合 Amazon Health Aware 監控的整合，以監控可能降級的 AWS 資源。針對商務基本工作負載，此解決方案可讓您存取 AWS 服務的主動式即時警示。

資源

相關 Well-Architected 的最佳實務：

- 可用性定義

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？
- 發佈自訂指標
- 使用 Amazon CloudWatch 警報
- Amazon Health Aware (AHA)
- 設定 CloudWatch 複合警示
- re:Invent 2022 中 AWS 可觀測性的最新消息

相關工具：

- CloudWatch
- CloudWatch X-Ray

REL11-BP07 建立您的產品架構以符合可用性目標和運行時間服務水準協議 (SLA)

建立您的產品架構以符合可用性目標和運行時間服務水準協議 (SLA)。如果您發佈或私下同意可用性目標或運行時間 SLA，請確認您的架構和操作程序的設計可以支援。

預期成果：每個應用程式都有針對可用性的已定義目標和針對效能指標的 SLA，可加以監控和維護以符合業務成果。

常見的反模式：

- 設計和部署工作負載，而未設定任何 SLA。
- SLA 指標設定為高，而沒有合理或業務要求。
- 設定 SLA 但未考慮相依性及其基礎 SLA。
- 建立應用程式設計而未考慮彈性的共同責任模型。

建立此最佳實務的優勢：根據關鍵彈性目標設計應用程式，可協助您符合業務目標和客戶期望。這些目標可協助推動應用程式設計程序，評估不同的技術和考慮各種權衡。

若未建立此最佳實務，暴露的風險等級： 中

實作指引

應用程式設計必須將多元的要求納入考慮，這些要求是從業務、營運和財務目標衍生而來。在營運要求內，工作負載必須有特定彈性指標目標，才能適當地監控和支援。彈性指標不應該在部署工作負載之後設定或衍生。它們應該在設計階段期間定義，協助引導各種決策和權衡。

- 每個工作負載都應該有自己的一組彈性指標。這些指標可能與其他業務應用程式不同。
- 降低相依性對可用性有正面影響。每個工作負載都應該考慮其相依性及其 SLA。一般而言，選取可用性目標等於或大於工作負載目標的相依性。
- 請考慮鬆散耦合設計，讓您的工作負載在可行時不論是否有相依性受損，都可以正確操作。
- 減少控制平面相依性，特別是復原或降級期間。評估針對任務關鍵性工作負載靜態穩定的設計。使用資源節省來增加工作負載中這些相依性的可用性。
- 可觀測性和檢測對於透過降低平均偵測時間 (MTTD) 和平均修復時間 (MTTR) 來達成 SLA 相當關鍵。
- 低頻率失敗 (MTBF 較長)、較短的失敗偵測時間 (較短 MTTD) 和較短的修復時間 (較短 MTTR)，是用來在分散式系統中改善可用性的三個因素。
- 建立和符合工作負載的彈性指標，是任何有效設計的基礎。這些設計必須考慮到設計複雜性、服務相依性、效能、擴展和成本的權衡。

實作步驟

- 請考慮下列問題，檢閱和記載工作負載設計：
  - 控制平面用於工作負載的哪個地方？

  - 工作負載如何實作容錯能力？
  - 擴展、自動擴展、備援和高可用性元件的設計模式是什麼？
  - 資料一致性和可用性的要求是什麼？
  - 資源節省或資源靜態穩定性是否有任何考慮？
  - 服務相依性是什麼？
- 與利害關係人合作時根據工作負載架構定義 SLA 指標。請考慮工作負載所使用所有相依性的 SLA。
- 一旦設定 SLA 目標，最佳化架構以符合 SLA。
- 一旦設定可符合 SLA 的設計，實作營運變更、處理自動化以及也會著重在降低 MTTD 和 MTTR 的執行手冊。
- 一旦部署，監控和報告 SLA。

資源

相關的最佳實務：

- REL03-BP01 選擇如何劃分工作負載
- REL10-BP01 將工作負載部署至多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 將所有分層的修復自動化
- REL12-BP05 使用混沌工程測試彈性
- REL13-BP01 定義停機和資料遺失的復原目標
- 了解工作負載運作狀態

相關文件：

- 可用性與備援性
- 可靠性支柱 - 可用性
- 測量可用性
- AWS 故障隔離界限
- 彈性的共同責任模型
- 使用可用區域實現靜態穩定性
- AWS 服務水準協議 (SLA)

- AWS 上小組型架構的指引
- AWS 基礎設施
- 進階多可用區域彈性模式白皮書

相關服務：

- Amazon CloudWatch
- AWS Config
- AWS Trusted Advisor

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，進行測試是確認其依設計運作並提供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。

預期成果：您的團隊擁有一致且商定的方法來處理事件後分析。其中一個機制是錯誤糾正 (COE) 程序。COE 程序可幫助您的團隊識別、了解和解決事件的根本原因，同時還能建置機制和防護機制，以限制相同事件再次發生的可能性。

常見的反模式：

- 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。
- 專注於追究責任，而不是了解根本原因，造成恐懼文化並阻礙開放的溝通
- 無法分享見解，只讓一小群人知道事件分析調查結果，讓其他人無法從學到的教訓中受益
- 沒有機制可擷取機構知識而失去寶貴的見解，因為組織不會以更新過的最佳實務形式保存所學到的教訓，並導致重複發生相同或類似根本原因的事件

建立此最佳實務的優勢：進行事件後分析並分享結果可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。

未建立此最佳實務時的風險暴露等級：高

實作指引

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。

COE 程序的基石是記錄和解決問題。建議您定義標準化方式來記錄關鍵的根本原因，並確保加以檢視和解決。為事件後分析程序指派明確的擁有權。指定負責監督事件調查和後續跟進的團隊或個人。

鼓勵專注於學習和改進的文化，而不是追究責任的文化。強調目標是預防未來的事件，而不是懲罰個人。

開發用於進行事件後分析的明確定義程序。這些程序應概述要採取的步驟、要收集的資訊，以及要在分析期間解決的關鍵問題。徹底調查事件，跳脫出直接原因以找出根本原因和成因。使用五個為什麼之類的技術深入了解基礎問題。

維護事件分析所學教訓的儲存庫。此機構知識可以作為未來事件和預防工作的參考。分享事件後分析的調查結果和見解，並考慮舉行公開邀請的事件後檢討會議，以討論學到的教訓。

實作步驟

- 在進行事件後分析時，請確保事件後分析不會讓相關人員受到責備。這可讓事件中的相關人員平心靜氣看待建議的糾正措施，並促進誠實地自我評估與跨團隊合作。
- 定義標準化方式來記錄重要問題。這類文件的範例結構如下：
  - 發生了什麼？
  - 對客戶和您的業務有什麼影響？
  - 根本原因是什麼？
  - 您擁有什麼可以提供支援的資料？
    - 例如，指標和圖表
  - 對關鍵支柱的影響有哪些 (尤其是安全性)？
    - 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。您可以選擇在開發環境中以可靠性作為代價最佳化成本，或者針對關鍵任務解決方案，以較高成本達到可靠性的最佳化。安全始終是首要工作，因為您必須保護客戶。
  - 您獲得了什麼教訓？
  - 您正在採取什麼糾正措施？
    - 動作項目
    - 相關項目
- 建立用於進行事件後分析的明確定義標準作業程序。
- 設定標準化的事件報告程序。全面記錄所有事件，包括初始事件報告、日誌、通訊，以及事件期間採取的行動。
- 請記住，發生事件時不見得會有中斷情形。事件也可能是幾乎錯過的情況，或是系統雖以意想不到的方式執行，卻仍可履行其業務功能。

- 請根據意見回饋和學到的教訓，持續改善事件後分析程序。
- 擷取知識管理系統中的關鍵調查結果，並考慮任何應新增至開發人員指南或部署前檢查清單的模式。

資源

相關文件：

- 為什麼您應該開發錯誤糾正 (COE)

相關影片：

- Amazon 對於成功故障的方法
- AWS re:Invent 2021 - Amazon 建置者資料中心：在 Amazon 卓越營運

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。

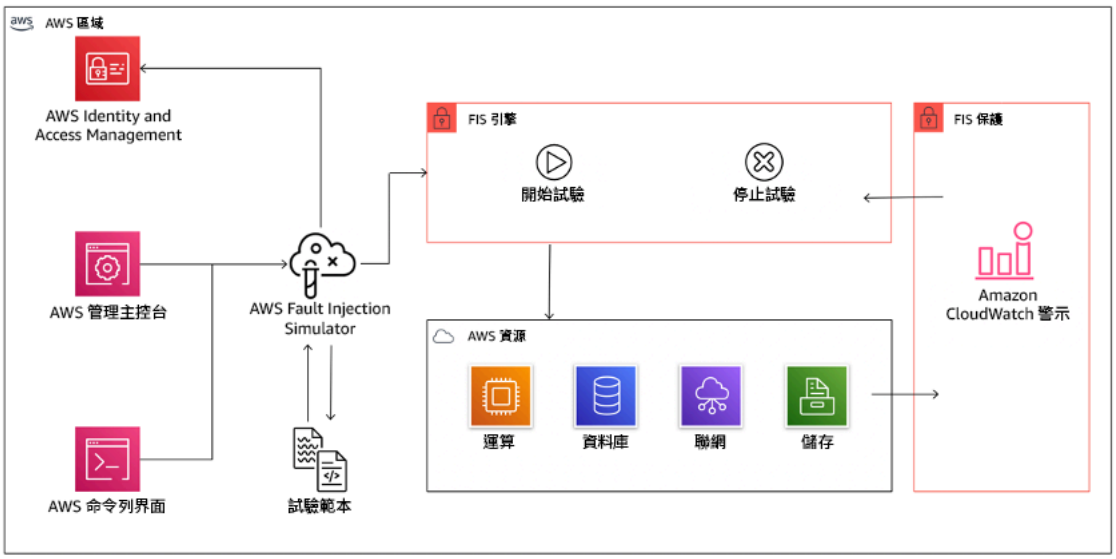
對於應用程式層級的錯誤 (例如當機)，您可以從記憶體和 CPU 用盡之類的壓力源開始著手。

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 將恢復能力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.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。

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)。

常見的反模式：

- 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。
- 沒有災難復原的計劃。
- 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改善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。

未建立此最佳實務時的風險暴露等級：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

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。

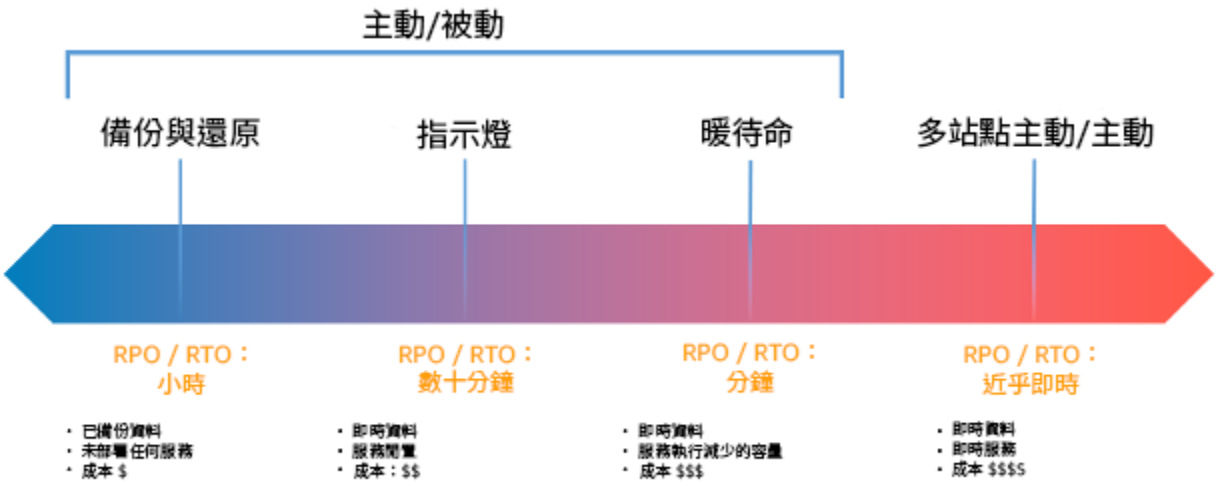


圖 17：災難復原 (DR) 策略

- 備份與還原 (RPO 以小時為單位，24 小時以內的 RTO)：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。暖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。
>
> 當成本是一大顧慮時，且想要達到與暖待命策略所定義類似的 RPO 和 RTO 目標，您可以考慮雲端原生解決方案，例如 AWS Elastic Disaster Recovery，它會採取指示燈方法並且提供改善的 RPO 和 RTO 目標。

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。

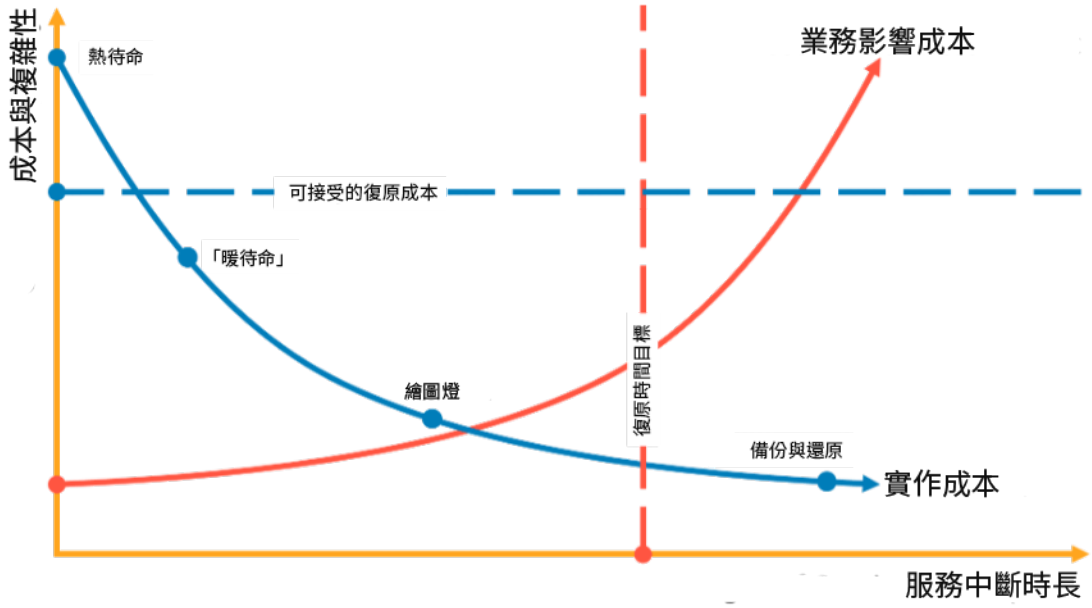


圖 18：根據 RTO 和成本選擇 DR 策略

若要進一步了解，請參閱業務持續性計劃 (BCP)。

2. 檢閱如何實作所選 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。

在下列步驟中，您可以將策略套用到您的特定工作負載。

備份與恢復

備份與還原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。

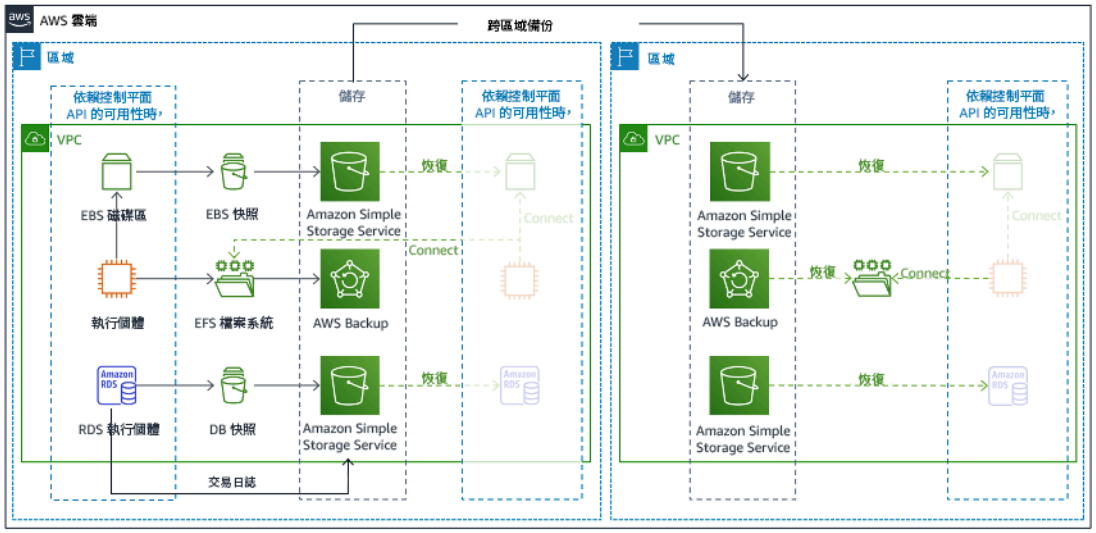


圖 19：備份和還原架構

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。

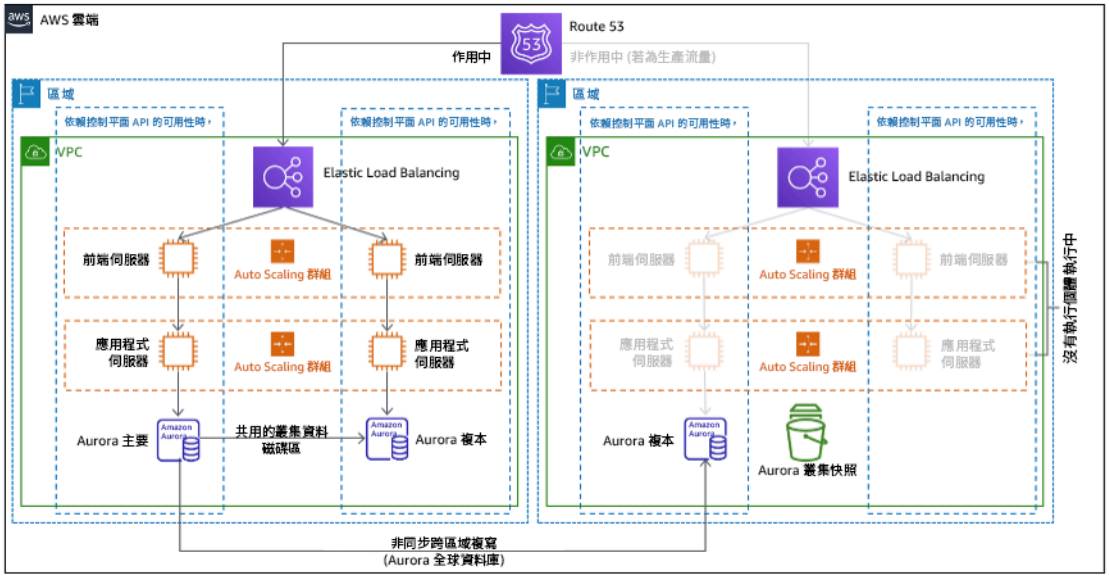


圖 20：指示燈架構

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

暖待命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。

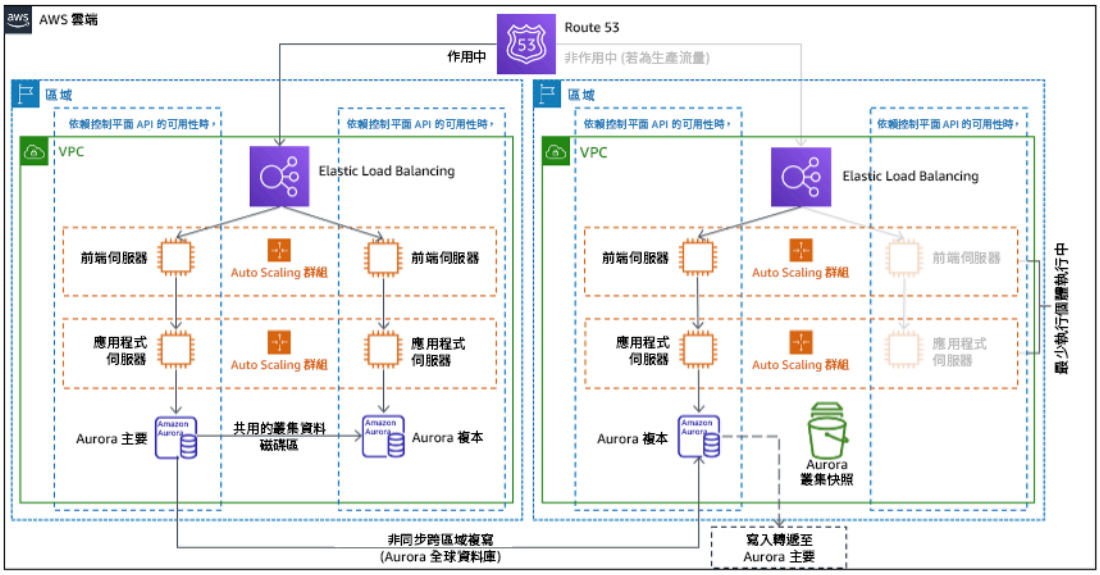


圖 21：暖待命架構

使用暖待命或指示燈需要縱向擴展復原區域中的資源。若要在需要時確認容量可用，請考慮針對 EC2 執行個體使用容量保留。如果使用 AWS Lambda，則佈建的並行可以提供執行環境，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。

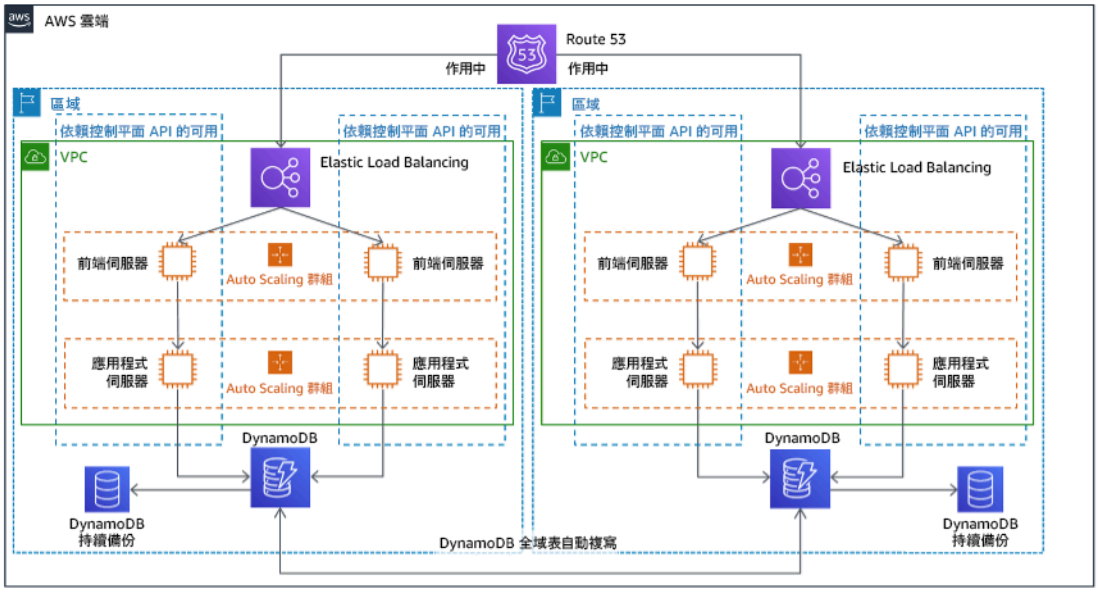


圖 22：多站點主動/主動架構

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。

AWS Elastic Disaster Recovery

如果您針對災難復原考慮指示燈或暖待命策略，AWS Elastic Disaster Recovery 可以提供具有改進優點的替代方法。Elastic Disaster Recovery 可以提供類似於暖待命的 RPO 和 RTO 目標，但是維持指示燈的低成本方法。Elastic Disaster Recovery 會將您的資料從主要區域複寫到您的復原區域，使用持續資料保護來達成以秒數測量的 RPO 和可以分鐘數測量的 RTO。只有複寫資料所需的資源會在復原區域中部署，保持低成本，類似於指示燈策略。使用 Elastic Disaster Recovery 時，服務會在容錯移轉或練習過程中啟動時進行協調。

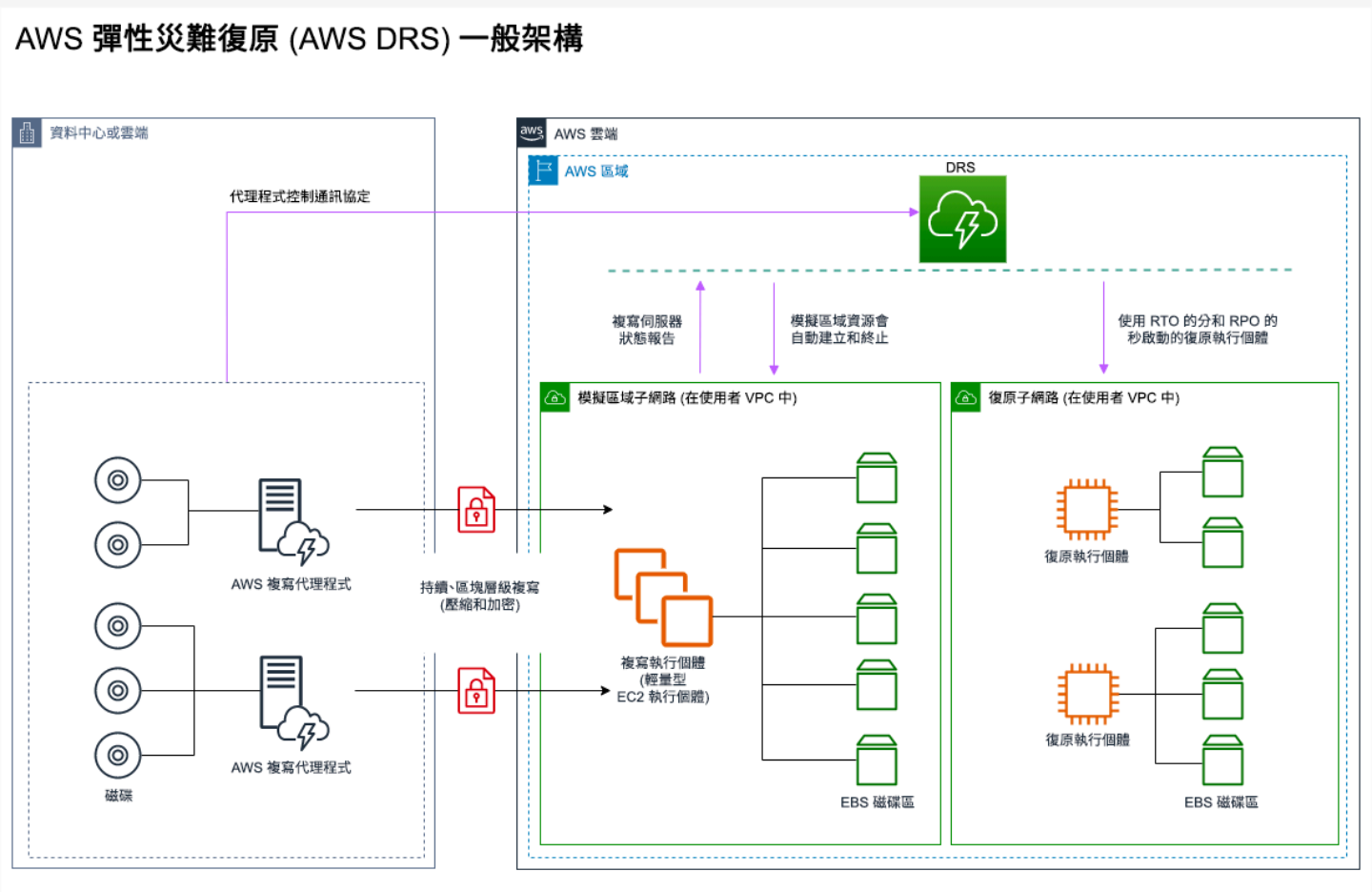


圖 23：AWS Elastic Disaster Recovery 架構

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。

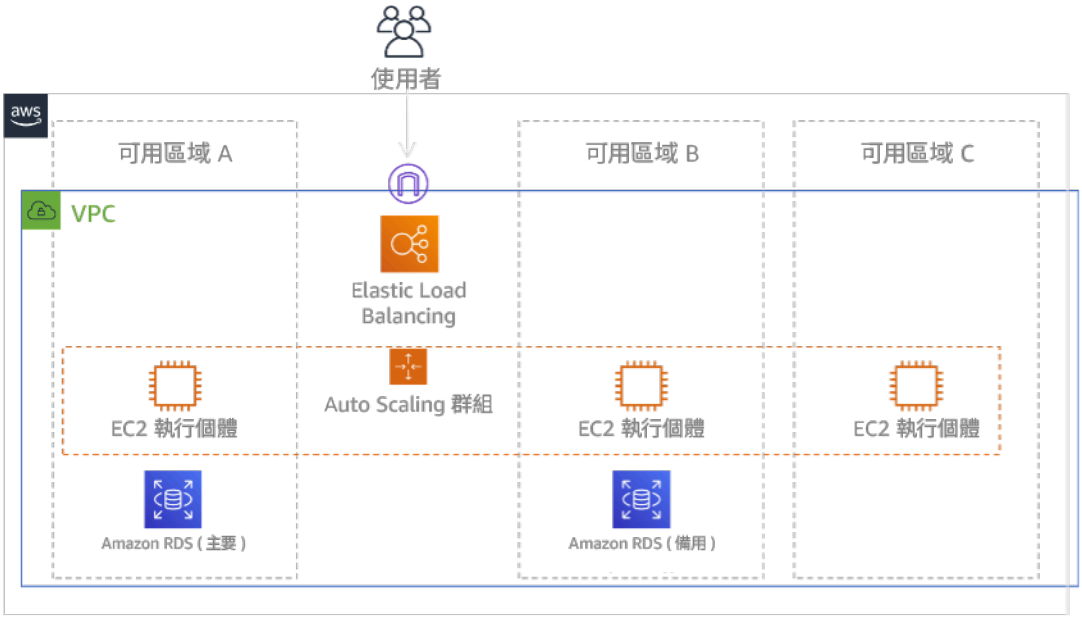


圖 24：多可用區域架構

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 (多可用區域 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。

> **Note**
>
> 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。

針對基礎設施和 AWS 資源使用基礎設施即程式碼，例如 AWS CloudFormation 像是 Hashicorp Terraform 的第三方工具。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已部署堆疊是主動或待命。使用 Elastic Disaster Recovery 時，服務會複寫和協調應用程式組態和運算資源的還原。

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) DB 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，請參閱使用 AWS 服務建立多區域應用程式這個部落格系列。

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。

容錯恢復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。容錯恢復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，則會維持 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 完成此操作，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。

使用 Elastic Disaster Recovery 時，服務會協助協調和自動化容錯恢復程序。如需詳細資訊，請參閱執行容錯恢復。

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：開始使用 - 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)
- 開始使用 AWS 彈性災難復原 | Amazon Web Services

相關範例：

- Well-Architected 實驗室 - 災難復原 - 說明 DR 策略的研討會系列

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認它正常操作，並符合 RTO 和 RPO。

常見的反模式：

- 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劃，可確保該計劃能在需要時運作，也能讓您的團隊知道如何執行策略。

未建立此最佳實務時的風險暴露等級：高

實作指引

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。

實作步驟

1. 為復原設計您的工作負載。定期測試您的復原路徑。復原導向運算可識別系統中能增強復原能力的特性：隔離和備援，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，以確認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。
2. 針對 Amazon EC2 型工作負載，使用 AWS Elastic Disaster Recovery 來實作和啟動您的 DR 策略的練習執行個體。AWS Elastic Disaster Recovery 提供有效率地執行練習的能力，可協助您為容錯移轉事件做準備。您也可以針對測試和練習目的使用 Elastic Disaster Recovery 頻繁地啟動您的執行個體，不需要重新導向流量。

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 Elastic Disaster Recovery
- AWS 上工作負載的災難復原：在雲端中復原 (AWS 白皮書)
- AWS Elastic Disaster Recovery 準備容錯移轉
- 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
- AWS re:Invent 2019：使用 AWS 的備份與還原和災難復原解決方案

相關範例：

- 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

效能達成效率支柱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。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。

最佳實務領域

選擇架構

問題

- PERF 1.如何為工作負載選取合適的雲端資源和架構？

PERF 1.如何為工作負載選取合適的雲端資源和架構？

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 工作負載會使用多種解決方案，並採用不同的功能以提升效能。

最佳實務

- PERF01-BP01 了解可用的雲端服務和特徵
- PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務
- PERF01-BP03 將成本因素納入架構決策
- PERF01-BP04 評估權衡如何影響客戶和架構效率
- PERF01-BP05 使用政策和參考架構
- PERF01-BP06 使用基準化分析來推動架構決策
- PERF01-BP07 針對架構選擇使用資料驅動的方法

PERF01-BP01 了解可用的雲端服務和特徵

持續了解並探索可用的服務和組態，有助您做出更完善的架構決策，並提升工作負載架構的效能效率。

常見的反模式：

- 您可以使用雲端作為並置資料中心。
- 遷移到雲端後，您沒有現代化應用程式。
- 對於需要保留的所有項目，您只使用一種儲存類型。

- 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： 透過考慮新的服務和組態，您可以大幅改善效能、降低成本，並最佳化維護工作負載所需的工作量。這麼做也可幫助您縮短具有雲端功能之產品的價值實現時間。

未建立此最佳實務時的曝險等級： 高

實作指引

AWS 持續推出可改善效能並降低雲端工作負載成本的新服務和特徵。即時掌握這些新服務和特徵的資訊，對於在雲端中維持效能效用至關重要。現代化工作負載架構也可有助您加速生產力、推動創新，並發掘更多成長機會。

實作步驟

- 清查工作負載軟體和架構以存放相關服務。決定要深入了解的產品類別。
- 探索 AWS 供應項目，以識別並了解相關服務和組態選項，這些選項可協助您改善效能，並降低成本和操作複雜性。
  - Amazon Web Services Cloud
  - AWS Academy
  - AWS 有哪些最新消息？
  - AWS 部落格
  - AWS Skill Builder
  - AWS 活動和研討會
  - AWS 培訓 和認證
  - AWS Youtube 頻道
  - AWS 研討會
  - AWS 社群
- 使用沙盒 (非生產) 環境來學習和試驗新服務，而不會產生額外成本。
- 持續了解新的雲端服務和功能。

資源

相關文件：

- 概述 Amazon Web Services
- Amazon EC2 功能
- 使用合作夥伴學習計劃AWS逐步學習
- AWS 訓練與認證
- 成為AWS解決方案架構師的學習途徑
- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心
- 建置 AWS 現代化應用程式

相關影片：

- AWS re:Invent 2023 - Amazon EC2 最新消息
- AWS re:Invent 2022 - 使用 Amazon ECS 降低營運和基礎架構成本
- AWS re:Invent 2023 - 使用 AWS 建置雲端的高效率、靈活性和創新
- AWS re:Invent 2022 - 部署適用於高效能和低成本推論的機器學習模型
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務

憑藉文件、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，來引導您做出架構決策。這些資源可協助您檢閱和改善架構，以實現最佳效能。

常見的反模式：

- 您使用 AWS 做為通用雲端供應商。
- 您使用 AWS 服務的方式與其設計宗旨不符。
- 您遵循所有指引，但未考量自身的業務環境。

建立此最佳實務的優勢： 運用雲端供應商或適當合作夥伴的指引，有助您選擇適合工作負載的正確架構，並對自己的決策充滿信心。

未建立此最佳實務時的曝險等級： 中

實作指引

AWS 提供廣泛的指引、文件和資源，有助您建置和管理有效率的雲端工作負載。AWS 文件提供程式碼範例、教學課程和詳細的服務說明。除了文件外，AWS 提供培訓和認證計畫、解決方案架構師和專業服務，有助客戶探索雲端服務的不同層面，並在 AWS 上實作有效的雲端架構。

利用這些資源，深入了解寶貴的知識和最佳實務、節省時間，並運用 AWS 雲端 取得更好的成果。

實作步驟

- 檢閱 AWS 文件和指引，並遵循最佳實務。這些資源可協助您有效選擇和設定服務，並達到更好的效能。
  - AWS 文件 (如使用者指南和白皮書)
  - AWS 部落格
  - AWS 培訓 和認證
  - AWS Youtube 頻道
- 參加 AWS 合作夥伴活動 (例如 AWS 全球高峰會、AWS re:Invent、使用者群組和研討會)，向 AWS 專家學習使用 AWS 服務的最佳實務。
  - 使用合作夥伴學習計劃AWS逐步學習
  - AWS 活動和研討會
  - AWS 研討會
  - AWS 社群
- 聯絡 AWS 尋求協助，滿足您的其他指引或產品資訊需求。AWS 解決方案架構師和 AWS 專業服務 可提供解決方案實作的相關指導。 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。
- 使用 AWS Support 如果您需要技術支援才能有效使用服務。 我們的支援計劃 旨在為您提供適當的工具組合和專業知識，以便您在最佳化效能、管理風險並控制成本的同時，透過 AWS 取得成功。

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 企業支援

相關影片：

- This is my Architecture
- AWS re:Invent 2023 - 搭配 Amazon EventBridge 的進階活動驅動型模式
- AWS re:Invent 2023 - 在 AWS 上實作分散式設計模式
- AWS re:Invent 2023 - 應用程式架構即程式碼

相關範例：

- AWS 範例
- AWS SDK 範例
- AWS 分析參考架構

PERF01-BP03 將成本因素納入架構決策

將成本因素納入架構決策中，以提高雲端工作負載的資源使用率和效能效率。當您意識到雲端工作負載的成本影響時，您就更有可能利用有效的資源並減少浪費的做法。

常見的反模式：

- 您只能使用一個執行個體系列。
- 您沒有根據開放原始碼解決方案，來評估授權解決方案。
- 您沒有定義儲存生命週期政策。
- 您沒有檢閱 AWS 雲端 的新服務和特徵。
- 您只能使用區塊儲存。

建立此最佳實務的優勢： 將成本因素納入決策中，可讓您使用更有效率的資源並探索其他投資選擇。

未建立此最佳實務時的曝險等級： 中

實作指引

針對成本最佳化工作負載可以提高資源利用率，並避免雲端工作負載的浪費。將成本納入架構決策中，通常包括適當調整工作負載元件的大小和啟用彈性，進而提高雲端工作負載效能效率。

實作步驟

- 建立成本目標，例如雲端工作負載的預算限制。
- 找出造成工作負載成本增加的關鍵元件 (例如執行個體和儲存)。您可以使用 AWS Pricing Calculator 和 AWS Cost Explorer 找出造成工作負載中成本增加的關鍵因素。
- 了解 在雲端中 的定價模式，例如隨選、保留執行個體和 Savings Plans Spot 執行個體。
- 使用 AWS Well-Architected 成本最佳化最佳實務 最佳化這些關鍵元件的成本。
- 持續監控和分析成本，以找出工作負載中成本最佳化的機會。
  - 使用 AWS Budgets 在產生無法接受的成本時收到提醒。
  - 使用 AWS Compute Optimizer 或者 AWS Trusted Advisor 取得成本最佳化建議。
  - 使用 AWS 成本異常偵測 取得自動化的成本異常偵測和根本原因分析。

資源

相關文件：

- 什麼是 AWS 帳單和成本管理？
- 使用 AWS 進行成本最佳化
- 選擇 AWS 成本管理策略
- AWS 成本管理入門指南
- 成本智慧儀表板的詳細概要
- AWS 架構中心
- AWS 解決方案程式庫
- AWS 知識中心

相關影片：

- This is my Architecture
- AWS re:Invent 2023 - AWS 成本最佳化的最新消息

- AWSRE:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度
- AWS re:Invent 2023 - AWS 儲存成本最佳化最佳實務
- AWS re:Invent 2023 - 最佳化在多帳戶環境中的成本

相關範例：

- AWS Compute Optimizer 示範程式碼
- 成本最佳化研討會
- 雲端財務管理技術實作教戰手冊
- 新創公司最佳化：調整應用程式效能以達到最大效率
- 無伺服器最佳化研討會 (效能與成本)
- 擴展具成本效益的架構

PERF01-BP04 評估權衡如何影響客戶和架構效率

在評估與效能相關的改善之處時，請判斷哪些選擇將對客戶和工作負載效率產生影響。例如，如果使用鍵值資料存放區可提高系統效能，請務必評估此變更最終一致性本質對客戶的影響。

常見的反模式：

- 即使實作過程中有所取捨，您都假設應實作所有效能增益。
- 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： 評估與效能相關的潛在改善時，您必須決定進行此變更的優缺點是否符合工作負載需求。在某些情況下，您可能需要實作其他控制來彌補權衡。

未建立此最佳實務時的曝險等級： 高

實作指引

根據對效能和客戶造成的影響，找出架構中的關鍵領域。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。

實作步驟

- 了解工作負載需求和 SLA。

- 清楚定義評估因素。因素可能涉及工作負載的成本、可靠性、安全性和效能。
- 選擇可滿足您需求的架構和服務。
- 進行試驗和概念驗證 (POC)，以評估權衡因素以及對客戶和架構效率的影響。通常，高可用性、高效能且安全的工作負載會耗用更多雲端資源，但能夠提供更完善的客戶體驗。了解工作負載複雜性、效能和成本之間的權衡。通常，如果優先考慮兩個因素，將會犧牲第三個因素。

資源

相關文件：

- Amazon 建置者資料中心
- Amazon QuickSight KPI
- Amazon CloudWatch RUM
- X-Ray 文件
- 了解能在雲端中快速建構架構的彈性模式和權衡

相關影片：

- 最佳化應用程式透過 Amazon CloudWatch RUM
- AWS re:Invent 2023 - 容量、可用性、成本效率：聚焦三件事
- AWS re:Invent 2023 - 適用於鬆散耦合系統的進階整合模式與權衡

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web 用戶端

PERF01-BP05 使用政策和參考架構

選擇服務和組態時，使用內部政策和現有的參考架構，以便在設計和實作工作負載時提高效率。

常見的反模式：

- 您可以廣泛使用可能影響公司管理開銷的技術。

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，讓您快速做出決策。

未建立此最佳實務時的曝險等級： 中

實作指引

在選擇資源和架構時，制定內部政策可讓您在選擇架構時有可遵循的標準和準則。這些指導方針則可簡化在選擇合適的雲端服務時的決策流程，並提高效能效率。使用政策或參考架構來部署工作負載。將服務整合到雲端部署，然後使用效能測試以確保您可以繼續滿足效能需求。

實作步驟

- 清楚了解雲端工作負載的需求。
- 檢閱內部和外部政策，以找出最相關的政策。
- 使用由 AWS 或業界最佳實務提供的適當參考架構。
- 針對常見情況，建立包含政策、標準、參考架構和規範性指引的連續體。這樣做可加快團隊的應變速度。為產業量身打造資產 (如果適用)。
- 針對沙盒環境中的工作負載驗證這些政策和參考架構。
- 隨時掌握產業標準和 AWS 更新，確保政策和參考架構有助最佳化雲端工作負載。

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 建築部落格

相關影片：

- This is my Architecture
- AWS re:Invent 2022 - 利用 SAP 和 AWS 參考架構提升業務價值

相關範例：

- AWS 範例

- AWS SDK 範例

PERF01-BP06 使用基準化分析來推動架構決策

基準化分析現有工作負載的效能，以了解工作負載在雲端的效能，並根據該資料推動架構決策。

常見的反模式：

- 您依賴的常見基準化分析未能反映工作負載特性。
- 您依賴客戶意見回饋和感受作為唯一的基準化分析。

建立此最佳實務的好處：基準化分析目前的實作，可協助您衡量效能改善之處。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用基準化分析搭配綜合測試，以評估工作負載元件執行情況。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分析測試，也可以使用產業標準測試 (例如 TPC-DS)，對工作負載進行基準分析。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著時間出現的任何變化。

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，因為負載測試會告訴您工作負載在生產中的效能。

實作步驟

- 規劃和定義：
  - 為您的基準定義目標、基準線、測試案例、指標 (例如 CPU 使用率、延遲或輸送量) 以及 KPI。
  - 關注使用者體驗方面的需求，以及回應時間和可存取性等因素。
  - 找出適合您工作負載的基準工具。您可以使用 AWS 服務 (例如 Amazon CloudWatch)，或與您的工作負載相容的第三方工具。

- 組態設定和檢測：
  - 設定您的環境並配置您的資源組態。
  - 實作監控和記錄以擷取測試結果。
- 基準化分析及監控：
  - 在測試期間執行基準化分析並監控指標。
- 分析和記錄：
  - 記錄您的基準化分析流程和調查結果。
  - 分析結果以識別瓶頸、趨勢和待改進領域。
  - 使用測試結果做出架構決策，並調整工作負載。這可能包括變更服務或採用新功能。
- 最佳化並重複：
  - 根據您的基準調整資源組態和配置。
  - 調整後重新測試您的工作負載，以驗證您的改善情況。
  - 記錄您的學習過程，並重複該過程以找出其他需要改進的地方。

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
- 基因組學工作流程，第 5 部分：自動基準化分析
- 在 Amazon SageMaker JumpStart 中基準化分析及最佳化端點部署

相關影片：

- AWS re:Invent 2023 - 基準化分析 AWS Lambda 冷啟動
- 在雲端中進行狀態化服務的基準化分析
- This is my Architecture
- 透過 Amazon CloudWatch RUM 最佳化應用程式

- 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 針對架構選擇使用資料驅動的方法

為各種架構選擇定義適合的清晰、資料驅動型方法，以確認是否使用正確的雲端服務和組態，滿足您的特定業務需求。

常見的反模式：

- 您認為自己的目前架構是固定不變的，且不應隨著時間而更新。
- 架構選擇是根據猜測和假設。
- 你隨著時間推移而導入了架構變更，卻沒有提供充分的理由。

建立此最佳實務的優勢： 透過採用明確定義的方法來做出架構選擇，您使用資料來影響工作負載設計，並隨著時間的推移做出明智的決策。

未建立此最佳實務時的曝險等級： 中

實作指引

使用內部經驗和雲端知識，或使用外部資源 (例如已發佈的使用案例或白皮書)，以選擇架構中的資源和服務。您應有定義明確的流程，有助試驗和基準化分析可在工作負載中使用的服務。

關鍵工作負載的待辦項目不僅只包括使用者案例 (提供與業務和使用者相關的功能)，還應包括構成工作負載架構跑道的技術案例。這條跑道掌握技術和新服務的新進展，並根據資料和適當的理由而採用這些技術和新服務。這證明架構仍然是與時俱進，不會停滯不前。

實作步驟

- 與關鍵利害關係人互動，以定義工作負載需求，包括效能、可用性和成本考量。考慮工作負載的使用者數量和使用模式等因素。

- 建立架構跑道，或根據功能待辦項目優先順序設定的技術待辦項目。
- 評估不同的雲端服務 (如需詳細資訊，請參閱 PERF01-BP01 了解可用的雲端服務和特徵)。
- 探索符合效能需求的不同架構模式，例如微服務或無伺服器 (如需更多詳細資訊，請參閱 PERF01-BP02 使用雲端供應商或適當合作夥伴提供的指引，了解架構模式和最佳實務)。

- 諮詢其他團隊、架構圖表和資源，例如 AWS 解決方案架構設計師、 AWS 架構中心和 AWS Partner Network，以協助您選擇適合工作負載的架構。

- 定義輸送量和回應時間等效能指標，以協助您評估工作負載的效能。
- 試驗並使用定義的指標，來驗證所選架構的效能。
- 視需要持續監控並進行調整，以維持架構的最佳效能。
- 記錄您選擇的架構和決策，作為未來更新和學習的參考。
- 根據經驗、新技術和指出目前方法中需要變更或問題的指標，持續檢閱和更新架構選擇方法。

資源

相關文件：

- AWS 解決方案程式庫
- AWS 知識中心
- 可在 AWS 上建置端對端資料驅動型應用程式的架構模式

相關影片：

- This is my Architecture
- AWS re:Invent 2021 - 資料驅動型企業：從願景轉為價值
- AWS re:Invent 2022 - 提供永續且高效能的架構
- AWSRE:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度
- AWS re:Invent 2022 - AWS 最佳化：讓您立即獲得結果的可採取動作的步驟

相關範例：

- AWS 範例
- AWS SDK 範例

運算與硬體

PERF 2.如何在工作負載中選取和使用運算資源？

特定工作負載的最佳運算選擇會根據應用程式設計、使用模式和組態設定而有所不同。架構會針對不同元件使用不同運算選擇，並採用不同功能以提升效能。若選錯運算資源，可能使架構的效能達成效率降低。

最佳實務

- PERF02-BP01 選擇最適合您工作負載的運算選項
- PERF02-BP02 了解可用的運算組態和特徵
- PERF02-BP03 收集與運算相關的指標
- PERF02-BP04 設定運算資源及適當調整其大小
- PERF02-BP05 動態擴展運算資源
- PERF02-BP06 使用最佳化的硬體型運算加速器

PERF02-BP01 選擇最適合您工作負載的運算選項

為工作負載選擇最合適的運算選項，可讓您改善效能、減少不必要的基礎架構成本，並降低維護工作負載所需的作業工作量。

常見的反模式：

- 您使用曾用於內部部署的同一個運算選項。
- 您不了解雲端運算選項、特徵以及解決方案，以及那些解決方案可以如何改善運算效能。
- 您在替代運算選項更精確地符合工作負載特性時，過度佈建現有運算選項以符合擴展或效能需求。

建立此最佳實務的好處：透過找出運算需求並根據可用選項進行評估，您就可以提高工作負載的資源效率。

未建立此最佳實務時的風險暴露等級：高

實作指引

為了最佳化雲端工作負載以提高效能效率，請務必根據使用案例和效能需求選擇最合適的運算選項。AWS 提供多種運算選項，以滿足雲端中不同工作負載的需求。例如，您可以使用 Amazon EC2 來啟動和管理虛擬伺服器、使用 AWS Lambda 來執行程式碼，而不必佈建或管理伺服器、使用 Amazon

ECS 或 Amazon EKS 執行和管理容器，或者使用 AWS Batch 平行處理大量資料。根據擴展和運算需求，您應該根據自己的情況選擇並設定最佳的運算解決方案。您也可以考慮在單一工作負載中使用多種運算解決方案，因為每種運算解決方案都有優缺點。

下列步驟會引導您選擇正確的運算選項，以符合工作負載特性和效能需求。

實作步驟

- 了解工作負載運算需求。要考量的關鍵需求包括處理需求、流量模式、資料存取模式、擴展需求，以及延遲需求。
- 了解在 AWS 上適用於工作負載的不同運算選項 (如 PERF01-BP01 了解可用的雲端服務和特徵 中所述)。以下是一些關鍵的 AWS 運算選項、其特性和常見使用案例：

AWS service	Key characteristics	Common use cases
Amazon Elastic Compute Cloud (Amazon EC2)	Has dedicated option for hardware, license requirements, large selection of different instance families, processor types and compute accelerators	Lift and shift migrations, monolithic application, hybrid environments, enterprise applications
Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS)	Easy deployment, consistent environments, scalable	Microservices, hybrid environments
AWS Lambda	無伺服器運算 service that runs code in response to events and automatically manages the underlying compute resources.	Microservices, event-driven applications
AWS Batch	Efficiently and dynamically provisions and scales Amazon Elastic Container Service (Amazon ECS), Amazon Elastic	HPC, train ML models

AWS service	Key characteristics	Common use cases
	Kubernetes Service (Amazon EKS), and AWS Fargate compute resources , with an option to use On-Demand or Spot Instances based on your job requirements	
Amazon Lightsail	Preconfigured Linux and Windows application for running small workloads	Simple web applications, custom website

- 評估與每個運算選項相關聯的成本 (例如每小時費用或資料傳輸) 和管理開銷 (例如修補和調整規模)。
- 在非生產環境中執行試驗和基準化分析，以找出哪個運算選項最能滿足工作負載需求。
- 在您試驗和找出新的運算解決方案，請規劃遷移並驗證效能指標。
- 使用像是 Amazon CloudWatch 的 AWS 監控工具，和 AWS Compute Optimizer 之類的最佳化服務，根據實際使用模式持續最佳化運算資源。

資源

相關文件：

- 使用 AWS 進行雲端運算
- Amazon EC2 執行個體類型
- Amazon EKS 容器：Amazon 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：

- AWS re:Invent 2023 - AWS Graviton：AWS 工作負載的最佳性價比

- AWS re:Invent 2023 - AMS 中新增 Amazon Elastic Compute Cloud 生成式 AI 功能
- AWS re:Invent 2023 - Amazon Elastic Compute Cloud 的最新消息
- AWS re:Invent 2023 - 智慧型節約：Amazon Elastic Compute Cloud 成本最佳化策略
- AWS re:Invent 2021 - 支援新一代 Amazon Elastic Compute Cloud：深入探討 Nitro System
- AWS re:Invent 2019 - 最佳化 AWS 運算的效能和成本
- AWS re:Invent 2019 - Amazon Elastic Compute Cloud 基礎
- AWS re:Invent 2022 - 部署適用於高效能和低成本推論的機器學習模型
- AWS re:Invent 2019 - 最佳化 AWS 運算的效能和成本
- Amazon EC2 基礎
- 部署適用於高效能和低成本推論的機器學習模型

相關範例：

- 遷移 Web 應用程式至容器
- 執行 Serverless Hello World
- Amazon EKS 研討會
- Amazon EC2 研討會
- 使用 Amazon Elastic Compute Cloud 自動擴展功能的高效率和彈性的工作負載
- 使用容器服務遷移至 AWS Graviton

PERF02-BP02 了解可用的運算組態和特徵

了解運算服務的可用組態選項和特徵，有助您佈建適量的資源並提高效能效率。

常見的反模式：

- 您沒有根據工作負載特性，評估運算選項或可用的執行個體系列。
- 您為了滿足尖峰需求而過度佈建運算資源。

建立此最佳實務的優勢： 熟悉 AWS 運算特徵和組態，如此您就能使用為符合工作負載特性和需求而經最佳化的運算解決方案。

未建立此最佳實務時的曝險等級： 中

實作指引

每個運算解決方案都有獨特的組態和特徵，可支援不同的工作負載特性和需求。了解這些選項如何與工作負載互補，以及判斷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、特徵 (GPU、I/O)、爆量、逾時、函數大小、容器執行個體，以及並行。如果工作負載使用相同運算選項的時間已超過四週，並且您預計特性未來仍將保持不變，您可以使用 AWS Compute Optimizer 從 CPU 和記憶體的角度來判斷目前的運算選項是否適合此工作負載。

實作步驟

1. 了解工作負載需求 (例如 CPU 需求、記憶體和延遲)。
2. 檢閱 AWS 文件和最佳實務，以了解可協助改善運算效能的建議組態選項。以下是一些需要考慮的關鍵組態選項：

組態選項	範例
執行個體類型	• 運算最佳化 執行個體非常適合需要高 vCPU 與記憶體比例的工作負載。 • 記憶體最佳化 執行個體會提供大量記憶體，以支援記憶體密集工作負載。 • 儲存最佳化 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。
定價模式	• 隨需執行個體 讓您無需長期承諾，即可按小時或秒使用運算容量。這些執行個體非常適合應對超出基本性能需求的突增負載。 • Savings Plans 與隨需執行個體相比，可大幅節省成本，但使用者必須在一年或三年期間使用特定數量的運算能力。 • Spot 執行個體 可讓您以折扣價利用未使用的執行個體容量，以實現無狀態且具有容錯功能的工作負載。
Auto Scaling	使用 Auto Scaling 組態來比對運算資源與流量模式。

組態選項	範例
規模調整	• 使用 Compute Optimizer 取得採用機器學習技術的建議，以了解哪個運算組態最符合您的運算特性。 • 使用 AWS Lambda 功能調校 ，為 Lambda 函數選擇最佳組態。
硬體型運算加速器	• 加速運算執行個體 執行圖形處理或資料模式比對等功能時，會比以 CPU 為基礎的替代方案更有效率。 • 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 AWS Trainium、 AWS Inferentia，和 Amazon EC2 DL1

資源

相關文件：

- 使用 AWS 進行雲端運算
- Amazon EC2 執行個體類型
- Amazon EC2 執行個體的處理器狀態控制
- Amazon EKS 容器：Amazon 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：

- AWS re:Invent 2023 – AWS Graviton：AWS 工作負載的最佳性價比
- AWS re:Invent 2023 – AWS Management Console 中新增 Amazon EC2 生成式 AI 功能
- AWS re:Invent 2023 – Amazon EC2 最新消息
- AWS re:Invent 2023 – 智慧優惠組合：Amazon EC2 成本最佳化策略
- AWS re:Invent 2021 – 支援下一代 Amazon EC2：深入探討 Nitro 系統

- AWS re:Invent 2019 – Amazon EC2 基礎
- AWS re:Invent 2022 – https://www.youtube.com/watch?v=5B4-s_ivn1o

相關範例：

- Compute Optimizer 示範程式碼
- Amazon EC2 spot 執行個體研討會
- 使用 Amazon EC2 AWS Auto Scaling 達到高效且彈性的工作負載
- Graviton 開發人員研討會
- AWS for Microsoft workloads immersion day
- AWS for Linux workloads immersion day
- AWS Compute Optimizer 示範程式碼
- Amazon EKS 研討會

PERF02-BP03 收集與運算相關的指標

記錄並追蹤與運算相關的指標，進一步了解運算資源的效能，並改善效能及使用率。

常見的反模式：

- 您只使用手動日誌檔案來搜尋指標。
- 您只使用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： 收集效能相關指標，可協助您符合應用程式效能與業務需求，確保滿足工作負載需求。這麼做也可以協助您持續改善工作負載中的資源效能和使用率。

未建立此最佳實務時的曝險等級： 高

實作指引

雲端工作負載可以產生大量資料，例如指標、日誌和事件。在 AWS 雲端 中，收集指標是提高安全性、成本效率、效能和可永續發展性的關鍵步驟。AWS 可使用監控服務提供各種效能相關的指標，例如 Amazon CloudWatch 為您提供寶貴的洞察。CPU 使用率、記憶體使用率、磁碟 I/O 以及網路輸入

和輸出等指標，可協助您深入了解使用率層級或效能瓶頸。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。 在理想的情況下，您應該在單一平台中收集與運算資源相關的所有指標，並實作保留政策，以支援成本和營運目標。

實作步驟

1. 找出與工作負載相關的效能相關指標。您應該收集與資源使用率和雲端工作負載運作方式有關的指標 (例如回應時間和輸送量)。
   a. Amazon EC2 預設指標
   b. Amazon ECS 預設指標
   c. Amazon EKS 預設指標
   d. Lambda 預設指標
   e. Amazon EC2 記憶體和磁碟指標
2. 為工作負載選擇並設定合適的記錄和監控解決方案。
   a. AWS 原生可觀測性
   b. 適用於 OpenTelemetry 的 AWS Distro
   c. Amazon Managed Service for Prometheus
3. 根據工作負載需求，為指標定義必要的篩選條件和彙總。
   a. 使用 Amazon CloudWatch Logs 和指標篩選條件，量化自訂應用程式指標
   b. 使用 Amazon CloudWatch 策略標記收集自訂指標
4. 為指標設定資料保留政策，以符合安全性和營運目標。
   a. CloudWatch 指標的預設資料保留
   b. CloudWatch Logs 的預設資料保留
5. 如有必要，為指標建立警示和通知，以協助您主動回應效能相關問題。
   a. 使用 Amazon CloudWatch 異常偵測為自訂指標建立警示
   b. 使用 Amazon CloudWatch RUM 為特定網頁建立指標和警示
6. 使用自動化來部署指標和記錄彙總代理程式。
   a. AWS Systems Manager 自動化
   b. OpenTelemetry 收集器

資源

相關文件：

- 監控與可觀察性
- 最佳做法：使用 AWS 實作可觀測性
- 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 進行 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
- 監控 AWS Fargate Amazon EKS

相關影片：

- AWS re:Invent 2023 – [LAUNCH] 現代工作負載的應用程式監控
- AWS re:Invent 2023 – 實作應用程式可觀測性
- AWS re:Invent 2023 – 打造有效的可觀測性策略
- AWS re:Invent 2023 – 使用 AWS Distro for OpenTelemetry 實現無縫可觀測性
- AWS 上的應用程式效能管理

相關範例：

- AWS for Linux Workloads Immersion Day- Amazon CloudWatch
- 監控 Amazon ECS 叢集和容器
- 使用 Amazon CloudWatch 儀表板進行監控
- Amazon EKS 研討會

PERF02-BP04 設定運算資源及適當調整其大小

設定運算資源及適當調整其大小，以符合工作負載的效能需求，並避免未充分使用資源或過度使用資源的情況。

常見的反模式：

- 您忽略工作負載效能需求，導致過度佈建或佈建不足的運算資源。
- 您只選擇適用於所有工作負載的最大或最小執行個體。
- 為了方便管理，您只使用一個執行個體系列。
- 您忽略來自 AWS Cost Explorer 或 Compute Optimizer 適當調整大小的建議。
- 您沒有重新評估工作負載是否適用於新的執行個體類型。
- 您只驗證組織的少量執行個體組態。

建立此最佳實務的優勢： 適當調整運算資源的大小，可避免過度佈建和佈建不足的資源，以確保雲端中的最佳作業。適當調整運算資源的大小，通常可以提高效能和增強客戶體驗，同時降低成本。

未建立此最佳實務時的曝險等級： 中

實作指引

適當調整大小可讓組織以有效率且符合成本效益的方式操作雲端基礎架構，同時滿足其業務需求。過度佈建雲端資源可能會導致額外成本，而佈建不足可能會導致低落的效能和不佳的客戶體驗。AWS 提供類似 AWS Compute Optimizer 和 AWS Trusted Advisor 之類的工具，這類工具會使用歷史資料，來提供適當調整運算資源大小的建議。

實作步驟

- 選擇最適合您需求的執行個體類型：
  - 如何為工作負載選擇適當的 Amazon EC2 執行個體類型？
  - Amazon EC2 機群的屬性型執行個體類型選取
  - 使用屬性型執行個體類型選取建立 Auto Scaling 群組。
  - 運用 Karpenter 整合，最佳化 Kubernetes 運算成本
- 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，選擇最適合您工作負載設定檔和效能目標的資源。
- 使用 Amazon CloudWatch 之類的 AWS 監控工具，監控資源使用情況。
- 為運算資源選取適合的組態。
  - 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 和 AWS Trusted Advisor)，以找出對運算資源進行最佳化和適當調整大小的機會。
- 在即時環境中實作之前，先測試非生產環境中的組態變更。

- 持續重新評估新的運算供應項目，並且根據工作負載需求進行比較。

資源

相關文件：

- 使用 AWS 進行雲端運算
- Amazon EC2 執行個體類型
- Amazon ECS 容器：Amazon ECS 容器執行個體
- Amazon EKS 容器：Amazon EKS 工作節點
- 函數：Lambda 函數組態
- Amazon EC2 執行個體的處理器狀態控制

相關影片：

- Amazon EC2 基礎
- AWS re:Invent 2023 – AWS Graviton：AWS 工作負載的最佳性價比
- AWS re:Invent 2023 – AWS Management Console 中新增 Amazon EC2 生成式 AI 功能
- AWS re:Invent 2023 – Amazon EC2 最新消息
- AWS re:Invent 2023 – 智慧優惠組合：Amazon EC2 成本最佳化策略
- AWS re:Invent 2021 – 支援下一代 Amazon EC2：深入探討 Nitro 系統
- AWS re:Invent 2019 – Amazon EC2 基礎

相關範例：

- AWS Compute Optimizer 示範程式碼
- Amazon EKS 研討會
- 適當調整大小的建議

PERF02-BP05 動態擴展運算資源

為滿足需求，請使用雲端的彈性，來動態擴充或縮減運算資源，並避免為工作負載佈建過多或過少的容量。

常見的反模式：

- 您可以手動增加容量，對警示做出反應。
- 您使用與內部部署相同的大小準則 (通常是靜態基礎結構)。
- 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： 設定和測試運算資源的彈性，可協助您節省成本、維持效能基準，並隨著流量變化改善可靠性。

未建立此最佳實務時的曝險等級： 高

實作指引

AWS 透過各種擴展機制，讓您能夠彈性動態擴充或縮減資源，以因應需求的變化。動態擴展與運算相關指標結合，讓工作負載能夠自動回應變更，並使用最佳的運算資源集來達成其目標。

您可以使用多種不同的方法達到資源的供需平衡。

- 目標追蹤法： ：監控擴展指標，並視需要自動增加或減少容量。
- 預測擴展：根據預測每日和每週趨勢進行擴展。
- 排程法：根據可預測的負載變更來設定您自己的擴展排程。
- 服務擴展：選擇可根據設計自動擴展的服務 (例如無伺服器)。

您必須確保工作負載部署可以同時處理擴展和縮減事件。

實作步驟

- 運算執行個體、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。以下是自動擴展機制的幾個範例：

自動擴展	應用環境
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。
Application Auto Scaling	自動將個別 AWS 服務的資源擴展到 Amazon EC2 以外，例如 AWS Lambda 函數或 Amazon Elastic Container Service (Amazon ECS) 服務。

自動擴展	應用環境
Kubernetes Cluster Autoscaler/Karpenter	自動擴展 Kubernetes 叢集。

- 擴展經常會與 Amazon EC2 執行個體或 AWS Lambda 函數等運算服務一併討論。請務必同時考慮非運算服務的組態，例如， AWS Glue ，以符合需求。
- 確認用於擴展的指標符合要部署之工作負載的特性。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。請改用轉碼任務佇列的深度。您可以將 自訂指標 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。
- 請確定您使用的是 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 用於您的動態擴展。
- 確認工作負載部署可同時處理擴展事件 (擴充和縮減)。例如，您可以使用 活動歷史 來確認 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。您可以透過預測擴展來避免過度佈建容量的需求。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 的預測擴展。

資源

相關文件：

- 使用 AWS 進行雲端運算
- Amazon EC2 執行個體類型
- Amazon ECS 容器：Amazon ECS 容器執行個體
- Amazon EKS 容器：Amazon EKS 工作節點
- 函數：Lambda 函數組態
- Amazon EC2 執行個體的處理器狀態控制
- 深入探討 Amazon ECS 叢集 Auto Scaling
- 介紹 Karpenter - 一個開放原始碼的高效能 Kubernetes Cluster Autoscaler

相關影片：

- AWS re:Invent 2023 – AWS Graviton：AWS 工作負載的最佳性價比

- AWS re:Invent 2023 – AWS 管理主控台中新增 Amazon EC2 生成式 AI 功能
- AWS re:Invent 2023 – Amazon EC2 最新消息
- AWS re:Invent 2023 – 智慧優惠組合：Amazon EC2 成本最佳化策略
- AWS re:Invent 2021 – 支援下一代 Amazon EC2：深入探討 Nitro 系統
- AWS re:Invent 2019 – Amazon EC2 基礎

相關範例：

- Amazon EC2 Auto Scaling 群組範例
- Amazon EKS 研討會
- 擴展 Amazon EKS 工作負載時搭配在 IPv6 上執行

PERF02-BP06 使用最佳化的硬體型運算加速器

使用硬體加速器執行特定功能，比以 CPU 為基礎的替代方案更有效率。

常見的反模式：

- 在工作負載中，您尚未基準化分析一般用途執行個體和專用執行個體，而專用執行個體可以提供更高的效能並降低成本。
- 您使用硬體型運算加速器來執行任務，比起使用以 CPU 為基礎的替代方案更有效率。
- 未監控 GPU 使用率。

建立此最佳實務的好處：透過使用硬體型加速器，例如圖形處理單元 (GPU) 和現場可程式化閘道陣列 (FPGA)，您就可以更有效率地執行特定處理功能。

未建立此最佳實務時的風險暴露等級：中

實作指引

加速運算執行個體可讓您使用硬體型運算加速器，例如 GPU 和 FPGA。這些硬體加速器在執行某些功能 (例如圖形處理或資料模式比對) 時，會比 CPU 型加速器更有效率。許多加速工作負載 (例如轉譯、轉碼和機器學習) 在資源用量方面極為變化不定。只在需要時執行此硬體，不需要時便會自動除役，以改善整體效能效率。

實作步驟

- 識別哪些加速運算執行個體可以滿足您的要求。

- 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。AWS Inferentia 執行個體 (例如 Inf2 執行個體) 所提供的效能功耗比最多會比同類 Amazon EC2 執行個體高出 50%。
- 收集加速運算執行個體的用量指標。例如，您可以使用 CloudWatch 代理程式，來收集 GPU 的 `utilization_gpu` 和 `utilization_memory` 等指標，如使用 Amazon CloudWatch 收集 NVIDIA GPU 指標中所示。
- 將硬體加速器的程式碼、網路運作和設定最佳化，以確保系統會充分利用基礎硬體。
  - 將 GPU 設定最佳化
  - Deep Learning AMI 中的 GPU 監控和最佳化
  - 將 I/O 最佳化以針對 Amazon SageMaker 中的深度學習訓練來調校 GPU 效能
- 使用最新的高效能程式庫和 GPU 驅動程式。
- 使用自動化來釋出不使用的 GPU 執行個體。

資源

相關文件：

- 在 Amazon Elastic Container Service 上使用 GPU
- GPU 執行個體
- 使用 AWS Trainium 的執行個體
- 使用 AWS Inferentia 的執行個體
- 開始建構吧！ 使用自訂晶片和加速器來進行建構

- 加速運算
- Amazon EC2 VT1 執行個體
- 如何為工作負載選擇適當的 Amazon EC2 執行個體類型？
- 選擇最佳的 AI 加速器和模型編譯以 Amazon SageMaker 推斷電腦視覺

相關影片：

- AWS re:Invent 2021 - 如何為深度學習選取 Amazon Elastic Compute Cloud GPU 執行個體
- AWS re:Invent 2022 - [全新推出！] 介紹 AWS Inferentia2 型的 Amazon EC2 Inf2 執行個體
- AWS re:Invent 2022 - 利用 AWS Trainium 加速深度學習和創新

- AWS re:Invent 2022 - 利用 NVIDIA 進行 AWS 深度學習：從訓練到部署

相關範例：

- Amazon SageMaker 和 NVIDIA GPU Cloud (NGC)
- SageMaker 與 Trainium 和 Inferentia 一起用於最佳化深度學習訓練和推論工作負載
- 在 Amazon SageMaker 中使用 Amazon Elastic Compute Cloud Inf1 執行個體最佳化 NLP 模型

資料管理

PERF 3.如何在工作負載中儲存、管理和存取資料？

特定系統的最佳資料管理解決方案會根據資料類型 (區塊、檔案或物件)、存取模式 (隨機或循序)、所需輸送量、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 工作負載會使用專用資料存放區，這些存放區採用不同的功能以提升效能。

最佳實務

- PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區
- PERF03-BP02 評估資料存放區可用的組態選項
- PERF03-BP03 收集並記錄資料存放區效能指標
- PERF03-BP04 實作策略以改善資料存放區中的查詢效能
- PERF03-BP05 實作利用快取的資料存取模式

PERF03-BP01 使用最能滿足資料存取和儲存需求的專用資料存放區

了解資料特性 (例如可共用、大小、快取大小、存取模式、延遲、輸送量和資料的持續性)，為工作負載選擇適合的專用資料存放區 (儲存或資料庫)。

常見的反模式：

- 由於具備某種特定類型資料庫解決方案的內部經驗和知識，您堅持使用某個資料存取區。
- 您假設所有工作負載都有類似的資料儲存和存取需求。
- 您未實作資料目錄以清查資料資產。

建立此最佳實務的好處：了解資料特性和需求，可協助您判斷能滿足工作負載需求的最有效率且效能最高的儲存技術。

未建立此最佳實務時的風險暴露等級：高

實作指引

選取和實作資料儲存時，請確定查詢、擴展和儲存特性能滿足工作負載資料需求。AWS 提供多種資料儲存和資料庫技術，包括區塊儲存、物件儲存、串流儲存、檔案系統、關聯式、鍵值、文件、記憶體內、圖形、時間序列和帳本資料庫。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。透過了解資料特性和需求，您就可以擺脫單一儲存技術和一體適用的限制性方法，專注於如何適當管理資料。

實作步驟

- 對工作負載現有的各種資料類型執行清查。
- 了解並記錄資料特性和需求，包括：
  - 資料類型 (非結構化、半結構化、關聯式)
  - 資料量與成長
  - 資料耐用性：持續性、暫時性、臨時
  - ACID (單元性、一致性、隔離行為、持續性) 需求
  - 資料存取模式 (大量讀取或大量寫入)
  - 延遲
  - 輸送量
  - IOPS (每秒輸入/輸出操作次數)
  - 資料保留期
- 了解 AWS 上可用於工作負載的不同資料存放區 (儲存和資料庫服務)，這些資料存放區可以滿足資料特性 (詳情請參閱PERF01-BP01 了解可用的雲端服務和特徵)。AWS 儲存技術及其重要特性的一些範例包含：

類型	AWS 服務	重要特性
Object storage	Amazon S3	Unlimited scalability, high availability, and multiple options for accessibility. Transferring and accessing objects in and out of Amazon S3 can use a service, such as Transfer Acceleration or 存

類型	AWS 服務	重要特性
		取點, to support your location, security needs, and access patterns.
Archiving storage	Amazon S3 Glacier	Built for data archiving.
Streaming storage	Amazon Kinesis Amazon Managed Streaming for Apache Kafka (Amazon MSK)	Efficient ingestion and storage of streaming data.
Shared file system	Amazon Elastic File System (Amazon EFS)	可供多種類型的運算解決方案存取的可掛載檔案系統。
Shared file system	Amazon FSx	Built on the latest AWS compute solutions to support four commonly used file systems: NetApp ONTAP, OpenZFS, Windows File Server, and Lustre. Amazon FSx 延遲、輸送量和 IOPS vary per file system and should be considered when selecting the right file system for your workload needs.

類型	AWS 服務	重要特性
Block storage	Amazon Elastic Block Store (Amazon EBS)	Scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS includes SSD-backed storage for transactional, IOPS-intensive workloads and HDD-backed storage for throughput-intensive workloads.
Relational database	Amazon Aurora, Amazon RDS, Amazon Redshift.	Designed to support ACID (atomicity, consistency, isolation, durability) transactions, and maintain referential integrity and strong data consistency. Many traditional applications, enterprise resource planning (ERP), customer relationship management (CRM), and ecommerce use relational databases to store their data.
Key-value database	Amazon DynamoDB	Optimized for common access patterns, typically to store and retrieve large volumes of data. High-traffic web apps, ecommerce systems, and gaming applications are typical use-cases for key-value databases.

類型	AWS 服務	重要特性
Document database	Amazon DocumentDB	Designed to store semi-structured data as JSON-like documents. These databases help developers build and update applications such as content management, catalogs, and user profiles quickly.
In-memory database	Amazon ElastiCache , Amazon MemoryDB for Redis	Used for applications that require real-time access to data, lowest latency and highest throughput. You may use in-memory databases for application caching, session management, gaming leaderboards, low latency ML feature store, microservices messaging system, and a high-throughput streaming mechanism
Graph database	Amazon Neptune	Used for applications that must navigate and query millions of relationships between highly connected graph datasets with millisecond latency at large scale. Many companies use graph databases for fraud detection , social networking, and recommendation engines.

類型	AWS 服務	重要特性
Time Series database	Amazon Timestream	Used to efficiently collect, synthesize, and derive insights from data that changes over time. IoT applications, DevOps, and industrial telemetry can utilize time-series databases.
Wide column	Amazon Keyspaces (適用於 Apache Cassandra)	Uses tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table. You typically see a wide column store in high scale industrial apps for equipment maintenance, fleet management, and route optimization.
Ledger	Amazon Quantum Ledger Database (Amazon QLDB)	Provides a centralized and trusted authority to maintain a scalable, immutable, and cryptographically verifiabl e record of transactions for every application. We see ledger databases used for systems of record, supply chain, registrations, and even banking transactions.

- 如果您要建置資料平台，請利用 AWS 上的現代資料架構，來整合資料湖、資料倉儲和專用的資料存放區。
- 為工作負載選擇資料存放區時，需要考慮的關鍵問題如下：

Question	Things to consider
How is the data structured?	• 如果資料是非結構化的，請考慮 Amazon S3 之類的物件存放區，或 Amazon DocumentDB 之類的 NoSQL 資料庫 • 若為鍵值資料，請考慮 DynamoDB、Amazon ElastiCache for Redis 或 Amazon MemoryDB for Redis
What level of referential integrity is required?	• 若為外部索引鍵限制，關聯式資料庫 (例如 Amazon RDS 和 Aurora) 可以提供這種層級的完整性。 • 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，以便在單一請求中擷取，而不是跨文件或資料表聯結。
Is ACID (atomicity, consistency, isolation, durability) compliance required?	• 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。 • 如果 NoSQL 資料庫需要強大的一致性，您可以使用 DynamoDB 搭配高度一致性讀取。
How will the storage requirements change over time? How does this impact scalability?	• DynamoDB 和 Amazon Quantum Ledger Database (Amazon QLDB) 等無伺服器資料庫將動態擴展。 • 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。
What is the proportion of read queries in relation to write queries? Would caching be likely to improve performance?	• 包含大量讀取作業的工作負載可從快取層中受益，例如 ElastiCache 或 DAX (如果資料庫是 DynamoDB)。 • 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。

Question	Things to consider
Does storage and modification (OLTP - Online Transaction Processing) or retrieval and reporting (OLAP - Online Analytical Processing) have a higher priority?	• 對於高輸送量讀取按原狀交易處理，請考慮使用 NoSQL 資料庫，例如 DynamoDB。 • 對於具有一致性的高輸送量和複雜的讀取模式 (如聯結)，請使用 Amazon RDS。 • 對於分析查詢，請考慮使用單欄式資料庫，例如 Amazon Redshift 或使用 Athena 或 Amazon QuickSight 將資料匯出至 Amazon S3 和執行分析。
What level of durability does the data require?	• Aurora 會自動跨區域內的三個可用區域複寫資料，這表示資料可以耐久，因而降低資料遺失的機會。 • 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。 • Amazon S3 提供 11 個 9 的耐用性。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。
Is there a desire to move away from commercial database engines or licensing costs?	• 考慮開放原始碼引擎，例如 Amazon RDS 或 Aurora 上的 PostgreSQL 和 MySQL。 • 運用 AWS Database Migration Service 和 AWS Schema Conversion Tool，從商務資料庫引擎遷移至開放原始碼
What is the operational expectation for the database? Is moving to managed services a primary concern?	• 利用 Amazon RDS (而非 Amazon EC2)，和 DynamoDB 或 Amazon DocumentDB (而非自行託管 NoSQL 資料庫)，可以降低營運負擔。

Question	Things to consider
How is the database currently accessed? Is it only application access, or are there business intelligence (BI) users and other connected off-the-shelf applications?	• 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。

- 在非生產環境中執行試驗和基準化分析，以找出哪個資料存放區能滿足工作負載需求。

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 S3 Glacier：S3 Glacier 文件
- Amazon S3：請求率和效能考量
- AWS 的雲端儲存
- Amazon EBS I/O 特性
- AWS 的雲端資料庫
- AWS 資料庫快取
- 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 Amazon EC2 和 Amazon RDS 之間選擇
- 實作 Amazon ElastiCache 的最佳實務

相關影片：

- AWS re:Invent 2023：提高 Amazon Elastic Block Store 效率並更具成本效益
- AWS re:Invent 2023：利用 Amazon Simple Storage Service 最佳化儲存價格和效能
- AWS re:Invent 2023：在 Amazon Simple Storage Service 上建置資料湖並進行最佳化
- AWS re:Invent 2022：在 AWS 上建置現代化資料架構
- AWS re:Invent 2022：在 AWS 上建置資料網格架構
- AWS re:Invent 2023：深入探索 Amazon Aurora 及其創新
- AWS re:Invent 2023：使用 Amazon DynamoDB 的進階資料建模
- AWS re:Invent 2022：透過專用資料庫建置現代化應用程式
- Amazon DynamoDB 深入探討：進階設計模式

相關範例：

- AWS 專用資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構 Immersion Day
- 在 AWS 上建置資料網格
- Amazon S3 範例
- 使用 Amazon Redshift 資料共用來最佳化資料模式
- 資料庫遷移
- MS SQL Server - AWS Database Migration Service (AWS DMS) 複寫示範
- 資料庫現代化實際操作研討會
- Amazon Neptune 範例

PERF03-BP02 評估資料存放區可用的組態選項

了解並評估資料存放區可用的各種特徵和組態選項，以最佳化工作負載的儲存空間和效能。

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 EBS。
- 您為所有工作負載使用佈建 IOPS，卻未針對所有儲存層進行實際測試。
- 您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。
- 未測試資料存放區的擴展特性。

建立此最佳實務的好處：藉由探索和試驗資料存放區組態，您能夠降低基礎架構成本、改善效能，以及減少維護工作負載所需的工作量。

未建立此最佳實務時的風險暴露等級：中

實作指引

工作負載可以根據資料儲存和存取需求使用一或多個資料存放區。若要最佳化效能效率和成本，您必須評估資料存取模式，以判斷適當的資料存放區組態。在探索資料存放區選項時，請考量各種層面，例如儲存選項、記憶體、運算、讀取複本、一致性需求、連線共用，以及快取選項。試驗這些不同的組態選項來改善效能效率指標。

實作步驟

- 了解資料存放區的目前組態 (例如執行個體類型、儲存大小或資料庫引擎版本)。
- 檢閱 AWS 文件和最佳實務，以了解可協助改善資料存放區效能的建議組態選項。要考慮的關鍵資料存放區選項如下：

Configuration option	Examples
Offloading reads (like read replicas and caching)	• 若為 DynamoDB 資料表，您可以使用 DAX 卸載讀取，以進行快取。 • 您可以建立 Amazon ElastiCache for Redis 叢集，並將應用程式設定為先從快取中讀取，如果請求的項目不存在，則退回到資料庫。 • 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。 • 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。

Configuration option	Examples
Scaling writes (like partition key sharding or introducing a queue)	• 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佈建 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，有助改善高寫入量關聯式資料庫中的輸送量。 • 取決於容量模式，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。 • 當您達到指定分區索引鍵的輸送量限制時，仍可能會遇到熱分區的問題。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。
Policies to manage the lifecycle of your datasets	• 您可以使用 Amazon S3 生命週期，以在物件的整個生命週期中管理物件。如果存取模式不明、會變化或是無法預測，則可以使用 Amazon S3 Intelligent-Tiering，讓其監控存取模式，並自動將未存取的物件移至成本較低的存取層。您可以利用 Amazon S3 Storage Lens 指標，來找出生命週期管理中的最佳化機會和落差。 • Amazon EFS 生命週期管理會自動管理檔案系統的檔案儲存。

Configuration option	Examples
Connection management and pooling	• Amazon RDS Proxy 可以與 Amazon RDS 和 Aurora 搭配使用，以管理資料庫的連線。 • 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。

- 在非生產環境中執行試驗和基準化分析，以找出哪個組態選項能滿足工作負載需求。
- 完成試驗之後，請規劃遷移並確認效能指標。
- 使用 AWS 監控 (例如 Amazon CloudWatch) 和最佳化 (例如 Amazon S3 Storage Lens) 工具，以透過實際使用模式持續最佳化資料存放區。

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 S3 Glacier：S3 Glacier 文件
- Amazon S3：請求率和效能考量
- Amazon EBS I/O 特性
- AWS 的雲端資料庫
- AWS 資料庫快取
- 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 re:Invent 2023：提高 Amazon Elastic Block Store 效率並更具成本效益
- AWS re:Invent 2023：利用 Amazon Simple Storage Service 最佳化儲存價格和效能
- AWS re:Invent 2023：在 Amazon Simple Storage Service 上建置資料湖並進行最佳化
- AWS re:Invent 2023：AWS 檔案儲存最新消息
- AWS re:Invent 2023： 深入探索 Amazon DynamoDB

相關範例：

- AWS 專用資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構 Immersion Day
- Amazon EBS 自動擴展
- Amazon S3 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 Postgress 資料庫上的參數

PERF03-BP03 收集並記錄資料存放區效能指標

追蹤並記錄資料存放區的相關效能指標，以了解資料管理解決方案的成效。這些指標可協助您最佳化資料存放區、確認是否符合工作負載需求，並提供工作負載執行方式的清晰概觀。

常見的反模式：

- 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。
- 您只會在有問題時審查指標。
- 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。可以更快地識別和偵錯異常模式，進而改善資料存放區的效能和可靠性。

未建立此最佳實務時的曝險等級： 高

實作指引

若要監控資料存放區的效能，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，以確認是否符合工作負載需求。

指標應該同時包括支援資料存放區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、快取命中率和網路傳入和傳出指標，而資料存放區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，以及資料管理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，以調整和最佳化工作負載的資源。

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。

實作步驟

1. 找出要追蹤的資料存放區關鍵效能指標。
   a. Amazon S3 指標和維度
   b. 監控 Amazon RDS 執行個體中的指標
   c. 在 Amazon RDS 上使用 Performance Insights 監控資料庫負載
   d. 增強型監控概觀
   e. DynamoDB 指標和維度
   f. 監控 DynamoDB Accelerator
   g. 使用 Amazon CloudWatch 進行監控 Amazon MemoryDB for Redis
   h. 我應該監控哪些指標？
   i. 監控 Amazon Redshift 叢集效能
   j. Timestream 指標和維度
   k. Amazon Aurora Amazon CloudWatch 指標
   l. 在 Amazon Keyspaces (for Apache Cassandra) 中的記錄和監控
   m. 監控 Amazon Neptune 資源
2. 使用核准的記錄和監控解決方案來收集這些指標。 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。

3. 檢查資料存放區監控是否能從可偵測效能異常的機器學習解決方案中獲益。
    a. Amazon RDS Amazon DevOps Guru 可讓您查看效能問題，並做出更正動作的建議。
4. 在監控和記錄解決方案中設定資料保留，以符合安全性和營運目標。
    a. CloudWatch 指標的預設資料保留
    b. CloudWatch Logs 的預設資料保留

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
- 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 re:Invent 2022 - 使用 Amazon RDS 與 Aurora 進行效能監控，提供 Autodesk
- 使用 Amazon DevOps Guru for Amazon RDS 進行資料庫效能監控和調整
- AWS re:Invent 2023 - AWS 檔案儲存最新消息
- AWS re:Invent 2023 - 深入探索 Amazon DynamoDB
- AWS re:Invent 2023 - 建置與最佳化 Amazon S3 上的資料湖
- AWS re:Invent 2023 - AWS 檔案儲存最新消息
- AWS re:Invent 2023 - 深入探索 Amazon DynamoDB
- 在 Amazon ElastiCache 上監控 Redis 工作負載的最佳實務

相關範例：

- AWS 資料集擷取指標收集架構
- Amazon RDS 監控研討會
- AWS 專用資料庫研討會

PERF03-BP04 實作策略以改善資料存放區中的查詢效能

實作策略以最佳化資料並改善資料查詢，以便為工作負載提供更高的可擴展性和更高效的效能。

常見的反模式：

- 您未分割資料存放區中的資料。
- 您在資料存放區中只使用一種檔案格式儲存資料。
- 您未在資料存放區中使用索引。

建立此最佳實務的優勢： 最佳化資料和查詢效能可提高效率、降低成本並改善使用者體驗。

未建立此最佳實務時的曝險等級： 中

實作指引

資料最佳化和查詢調整是資料存放區中效能效率的關鍵層面，因為其會影響整個雲端應用程式工作負載的效能和回應能力。未經過最佳化的查詢可能會使用更多資源和造成更大的瓶頸，進而降低資料存放區的整體效率。

資料最佳化包括數個技術，以確保高效的資料儲存和存取。這也有助於改善資料存放區中的查詢效能。關鍵策略包括資料分割、資料壓縮和資料去常規化，這些有助於資料的儲存和存取達到最佳化。

實作步驟

- 了解和分析在資料存放區中執行的重要資料查詢。
- 找出資料存放區中速度緩慢的查詢執行，並使用查詢計畫了解其目前狀態。
  - 分析 Amazon Redshift 中的查詢計畫
  - 在 Athena 中使用 EXPLAIN 和 EXPLAIN ANALYZE
- 實作策略以改善查詢效能。有些關鍵策略包括下列情況：
  - 使用 單欄檔案格式 (例如，Parquet 或 ORC)。
  - 壓縮資料存放區中的資料以減少儲存空間和 I/O 作業。
  - 資料分割可將資料拆分為較小的部分，進而縮短資料掃描時間。

    - 在 Athena 中分割資料
    - 資料分割和資料分發
  - 在查詢中對共同欄進行資料索引編制。
  - 使用具體化視觀表進行頻繁查詢。
    - 了解具體化視觀表
    - 在 Amazon Redshift 中建立具體化視觀表
  - 選擇正確的聯結作業以進行查詢。當您聯結兩張資料表時，請指定聯結左側為較大資料表，並指定聯結右側為較小資料表。
  - 分散式快取解決方案可改善延遲並減少資料庫 I/O 操作的次數。
  - 定期維護，例如執行統計。
- 在非生產環境中試驗和測試策略。

資源

相關文件：

- Amazon Aurora 最佳實務
- Amazon Redshift 效能
- Amazon Athena 10 大效能秘訣
- AWS 資料庫快取
- 實作 Amazon ElastiCache 的最佳實務
- 在 Athena 中分割資料

相關影片：

- AWS re:Invent 2023 - AWS 儲存成本最佳化最佳實務
- AWS re:Invent 2022 - 使用 Amazon RDS 與 Aurora 進行效能監控，提供 Autodesk
- 使用新的查詢分析工具最佳化 Amazon Athena 查詢

相關範例：

- Amazon S3 Select - 直接查詢資料而不需要執行伺服器或資料庫
- AWS 專用資料庫研討會

PERF03-BP05 實作利用快取的資料存取模式

為快速擷取經常存取的資料，實作利用快取的存取模式。

常見的反模式：

- 您快取的資料經常變更。
- 您以為快取的資料能夠持久儲存且永遠可用，因而過於依賴。
- 您未考慮快取資料的一致性。
- 您未監控快取實作的效率。

建立此最佳實務的優勢： 將資料儲存在快取中可改善讀取延遲、讀取輸送量、使用者體驗和整體效率，並降低成本。

未建立此最佳實務時的曝險等級：中

實作指引

快取是軟體或硬體的元件，其用途在於儲存資料，以便未來請求相同的資料時，能夠更快且更有效率地提供服務。若儲存在快取中的資料遺失，可以透過重複先前的計算或從其他資料存放區提取的方式重新建構該資料。

資料快取可能是改善整體應用程式效能並減輕基礎主要資料來源負擔的最有效策略之一。資料可在應用程式中的多個層級快取，例如在進行遠端呼叫的應用程式內，稱為 用戶端快取，或者使用快速的次要服務來儲存資料，稱為 遠端快取。

用戶端快取

透過用戶端快取，每個用戶端 (查詢後端資料儲存的應用程式或服務) 都可將特定的查詢結果儲存在本機上，並在指定的時間內保留。這樣就能透過先查看本機用戶端快取，減少整體網路對資料儲存的請求數量。如果結果不存在，應用程式就可以查詢資料儲存，並將這些結果儲存在本機上。此模式可讓每個用戶端將資料儲存在最靠近的位置 (用戶端本身)，進而將延遲降至最低。用戶端也可在後端資料儲存無法使用時，繼續為部分查詢提供服務，以提高整體系統的可用性。

這種方法的缺點是，若有多個用戶端，則可能會將相同的快取資料儲存在本機上。這樣會導致這些用戶端之間發生重複使用儲存和資料不一致的情況。某一個用戶端可能快取了查詢的結果，而過了一分鐘後，另一個用戶端也可能執行相同的查詢，但得到不同的結果。

遠端快取

為了解決用戶端之間資料重複的問題，可使用快速的外部服務 (也稱為 遠端快取)來儲存查詢的資料。每個用戶端都會在查詢後端資料儲存之前查看遠端快取，而不會查看本機資料存放區。這種策略可讓用戶端之間的回應更趨一致、提高儲存資料的效率，以及增加快取的資料量，因為儲存空間的擴展與用戶端無關。

遠端快取的缺點是，整個系統可能產生較高的延遲，因為需要額外的網路跳轉來查看遠端快取。用戶端快取可與遠端快取一起使用，以提供多層次快取來改善延遲。

實作步驟

1. 找出可有效利用快取的資料庫、API 和網路服務。有大量讀取工作負載、高讀寫比例或擴展成本昂貴的服務，都適合使用快取。
   - 資料庫快取
   - 啟用 API 快取以提升回應能力
2. 確定最適合您存取模式的適當快取策略類型。
   - 快取策略
   - AWS 快取解決方案
3. 遵循適合您的資料存放區的 快取最佳實務 。
4. 為所有資料設定快取失效策略，例如存留時間 (TTL)，以便在資料時效性與減輕後端資料儲存壓力之間取得平衡。
5. 在用戶端中啟用像是自動連線重試、指數退避、用戶端逾時和連線共用等功能 (如果有的話)，因為這些功能可以改善效能和可靠性。
   - 最佳實務：Redis 用戶端和 Amazon ElastiCache for Redis
6. 設定 80% 或更高的目標來監控快取命中率。較低的值可能表示快取大小不足，或存取模式無法有效利用快取。
   - 我應該監控哪些指標？
   - 在 Amazon ElastiCache 上監控 Redis 工作負載的最佳實務
   - 使用 Amazon CloudWatch 監控 Amazon ElastiCache for Redis 的最佳實務
7. 實作 資料複寫 將讀取卸載到多個執行個體，並改善資料讀取效能和可用性。

資源

相關文件：

- 使用 Amazon ElastiCache Well-Architected Lens

- 使用 Amazon CloudWatch 監控 Amazon ElastiCache for Redis 的最佳實務
- 我應該監控哪些指標？
- 使用 Amazon ElastiCache 大幅提高效能白皮書
- 快取挑戰和策略

相關影片：

- Amazon ElastiCache 學習路徑
- 專為使用 Amazon ElastiCache 最佳實務獲得成功而設計
- AWS re:Invent 2020 - 專為使用 Amazon ElastiCache 最佳實務獲得成功而設計
- AWS re:Invent 2023 - [LAUNCH] 向您介紹 Amazon ElastiCache Serverless
- AWS re:Invent 2022 - 使用 Redis 重新塑造資料層的 5 種好方法
- AWS re:Invent 2021 - 深入探討 Amazon ElastiCache for Redis

相關範例：

- 使用 Amazon ElastiCache for Redis 大幅提高 MySQL 資料庫效能

網路與內容交付

PERF 4.如何在工作負載中選取和設定網路資源？

系統的最有效資料庫解決方案會根據可用性、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。許多系統針對不同的子系統使用不同的資料庫解決方案，並啟用不同功能以提升效能。若選錯資料庫解決方案和功能，可能使系統的效能達成效率降低。

最佳實務

- PERF04-BP01 了解聯網如何影響效能
- PERF04-BP02 評估可用的聯網功能
- PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN
- PERF04-BP04 使用負載平衡將流量分配到多個資源
- PERF04-BP05 選擇網路通訊協定以提高效能
- PERF04-BP06 根據網路需求選擇工作負載的位置
- PERF04-BP07 根據指標最佳化網路組態

PERF04-BP01 了解聯網如何影響效能

分析並了解網路相關決策如何影響您的工作負載，以提供高效率的效能並改善使用者體驗。

常見的反模式：

- 通過現有資料中心的所有流量。
- 您讓所有流量路由經過中央防火牆，而非使用雲端原生網路安全工具。
- 您佈建 AWS Direct Connect 連線，但不了解實際用量需求。
- 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。

未建立此最佳實務時的曝險等級： 高

實作指引

網路負責處理應用程式元件、雲端服務、邊緣網絡和內部部署資料之間的連線，因此對工作負載效能可能有大幅的影響。除了工作負載效能外，使用者體驗也可能受到網路延遲、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。

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。

實作步驟：

1. 定義並記錄聯網效能需求，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. 了解關鍵 AWS 聯網服務，如 VPC、AWS Direct Connect、Elastic Load Balancing (ELB)和 Amazon Route 53。
3. 擷取下列主要聯網特性：

特性	工具和指標
基礎聯網特性	• VPC Flow Logs

特性	工具和指標
	• AWS Transit Gateway Flow Logs • AWS Transit Gateway 指標 • AWS PrivateLink 指標
應用程式聯網特性	• Elastic Fabric Adapter • AWS App Mesh 指標 • Amazon API Gateway 指標
邊緣聯網特性	• Amazon CloudFront 指標 • Amazon Route 53 指標 • AWS Global Accelerator 指標
混合聯網特性	• AWS Direct Connect 指標 • AWS Site-to-Site VPN 指標 • AWS Client VPN 指標 • AWS 雲端 WAN 指標
安全聯網特性	• AWS Shield、AWS WAF 和 AWS Network Firewall 指標
追蹤特性	• AWS X-Ray • VPC Reachability Analyzer • Network Access Analyzer • Amazon Inspector • Amazon CloudWatch RUM

4. 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，有些因素可能會影響 Amazon EC2 網路效能。測量同一 VPC 中 Amazon EC2 Linux 執行個體之間的網路頻寬。
   b. 執行 負載測試 以試驗各種聯網解決方案和選項。

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 中以延遲為基礎的路由
- VPC 端點

相關影片：

- AWS re:Invent 2023 - AWS 聯網基礎
- AWS re:Invent 2023 - 聯網對您的應用程式有何助益？
- AWS re:Invent 2023 - 進階 VPC 設計及新功能
- AWS re:Invent 2023 - 開發人員的雲端聯網指南
- AWS re:Invent 2019 - 與 AWS 和混合 AWS 網路架構的連線能力
- AWS re:Invent 2019 - 最佳化 Amazon EC2 執行個體的網路效能
- AWS 線上高峰會 - 改善應用程式的全球網路效能
- AWS re:Invent 2020 - 搭配 Well-Architected Framework 的聯網最佳實務和秘訣
- AWS re:Invent 2020 - 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
- 實際操作網路防火牆研討會
- 觀察和診斷在 AWS 上的網路
- 尋找並解決在 AWS 上的網路設定錯誤

PERF04-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、網路距離或抖動。

常見的反模式：

- 您只在單一區域中活動，這是因為該區域是您總部的所在區域。
- 您使用防火牆而非安全群組來篩選流量。
- 您透過中斷 TLS 來檢測流量，而非使用安全群組、端點政策和其他雲端原生功能。
- 您只使用子網路來分隔，而非採用安全群組的方式。

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 骨幹來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

AWS 提供 AWS Global Accelerator 和 Amazon CloudFront 等服務，可協助提高網路效能，而大多數 AWS 服務都有產品功能 (例如 Amazon S3 Transfer Acceleration 功能) 可用來最佳化網路流量。

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。效能最佳化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體驗的影響。

實作步驟

- 建立工作負載元件清單。
  - 考慮在建置統一的全球網路時，使用 AWS 雲端 WAN 來建置、管理和監控您組織的網路。
  - 監控您的全球和核心網路，方法是使用 Amazon CloudWatch Logs 指標。利用 Amazon CloudWatch RUM，它提供了洞見來幫助您識別、了解和強化使用者的數位體驗。
  - 檢視 AWS 區域 與可用區域之間的彙總網路延遲，以及每個可用區域內的網路延遲，使用 AWS Network Manager 獲得洞見，讓您深入了解應用程式效能與基礎 AWS 網路效能的關係。
  - 使用現有的組態管理資料庫 (CMDB) 工具或 AWS Config 之類的服務，建立工作負載的庫存及了解其設定方式。

- 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。
- 如果這是新的工作負載，請執行 負載測試 以找出效能瓶頸。
- 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。參考下列主要聯網選項和功能：

改善機會	解決方案
網路路徑或路由	使用 Network Access Analyzer 識別路徑或路由。
網路通訊協定	請參閱 PERF04-BP05 選擇網路通訊協定以提高效能
網路拓撲	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 會簡化所有 VPC 的互連方式，如此便可橫跨數千個 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： AWS Resource Access Manager。 請參閱 PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN
網路服務	AWS Global Accelerator 是一項連網服務，可使用 AWS 全域網路基礎架構將使用者流量效能提升最多達 60%。 Amazon CloudFront 可以改善全域工作負載內容交付和延遲的效能。 使用 Lambda@edge 執行多項功能，以自訂能讓 CloudFront 就近提供給使用者的內容、減少延遲並改善效能。

改善機會	解決方案
	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。若您的工作負載分散於全球，請檢閱您的工作負載流量和使用者位置，找出能夠最佳化工作負載效能的路由選項。
儲存資源功能	Amazon S3 Transfer Acceleration 功能讓外部使用者得以獲益於 CloudFront 的聯網最佳化，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。 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。

改善機會	解決方案
運算資源功能	彈性網路介面 (ENA) 為 Amazon EC2 執行個體、容器和 Lambda 函數所使用，會受到個別流程的限制。請檢閱您的置放群組以最佳化 EC2 聯網輸送量。若要避免個別流程發生瓶頸，請將應用程式設計為使用多個流程。若要監控及掌握運算相關的聯網指標，請使用 CloudWatch 指標和 ethtool。此 `ethtool` 命令包含在 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。 Amazon 彈性網路介面卡 (ENA) 可提供進一步最佳化，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。 Elastic Fabric Adapter (EFA) 是 Amazon EC2 執行個體的網路介面，可讓您在 AWS 上大規模執行需要高階節點間通訊的工作負載。 經 Amazon EBS 最佳化的執行個體 使用最佳化的組態堆疊，可提供更多專用容量以增加 Amazon EBS I/O。

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
- 搭配 AWS 的聯網產品

- 轉換到 Amazon Route 53 中的 Latency-Based Routing
- VPC 端點
- VPC Flow Logs

相關影片：

- AWS re:Invent 2023 – 準備好踏出下一步了嗎？ 設計可促進成長與靈活性的網路
- AWS re:Invent 2023 – 進階 VPC 設計及新功能
- AWS re:Invent 2023 – 開發人員的雲端聯網指南
- AWS re:Invent 2022 – - 深入了解 AWS 聯網基礎設施
- AWS re:Invent 2019 – 連接 AWS 和混合 AWS 網路架構的連線能力
- AWS re:Invent 2018 – 最佳化 Amazon EC2 執行個體的網路效能
- 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
- 觀察和診斷網路
- 尋找並解決在 AWS 上的網路設定錯誤

PERF04-BP03 為您的工作負載選擇適當的專用連線或 VPN

需要混合式連線來連接內部部署和雲端資源時，請佈建足夠的頻寬來滿足您的效能需求。預估混合工作負載的頻寬和延遲需求。您的規模調整需求取決於這些數字。

常見的反模式：

- 您只針對網路加密需求評估 VPN 解決方案。
- 您未評估備份或備援連線選項。
- 您無法識別所有工作負載需求 (加密、通訊協定、頻寬和流量需求)。

建立此最佳實務的優勢： 選取和設定適當的連線解決方案將提高工作負載的可靠性，並充分利用效能。藉由識別工作負載需求、提前規劃和評估混合解決方案，就能盡可能減少昂貴的實體網路變更和營運負擔，同時實現更高的價值。

未建立此最佳實務時的曝險等級： 高

實作指引

根據您的頻寬要求開發混合式聯網架構。AWS Direct Connect 可讓您將內部部署網路與 AWS 進行私密連線。需要高頻寬且低延遲，同時可達到一致效能時，適合這個選項。VPN 連線會建立透過網際網路的安全連線。使用時機包括：只需要臨時連線、須考量成本因素，或是在使用 AWS Direct Connect 的情況下，等待建立彈性實體網路連線時做為緊急應變措施。

如果您的頻寬需求很高，可以考慮多個 AWS Direct Connect 或 VPN 服務。流量可在服務之間達到負載平衡，雖然因為延遲和頻寬的差異，我們不建議在 AWS Direct Connect 和 VPN 之間達到負載平衡。

實作步驟

1. 預估現有應用程式的頻寬和延遲需求。
   a. 針對移至 AWS 的現有工作負載，利用來自您的內部網路監控系統的資料。
   b. 針對您沒有監控資料的新或現有工作負載，請諮詢產品擁有者以決定適當的效能指標，並且提供良好的使用者體驗。
2. 選取專用連線或 VPN 做為您的連線選項。根據所有工作負載要求 (加密、頻寬和流量需求)，您可以選擇 AWS Direct Connect 或 AWS VPN (或兩者)。下圖可協助您選擇適當的連線類型。
   a. AWS Direct Connect 使用專用連線或託管連線，為 AWS 環境提供速度從 50 Mbps 到 100 Gbps 的專用連線。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有效率地連線到其他環境。使用 AWS Direct Connect 合作夥伴，您就可以擁有來自多個環境的端對端連線能力，提供擴充的網路和一致的效能。AWS 提供使用原生 100 Gbps、連結彙總群組 (LAG) 或 BGP 等價多路徑 (ECMP) 的擴展直接連線連線頻寬。
   b. AWS Site-to-Site VPN 提供受管 VPN 服務，支援網際網路通訊協定安全性 (IPsec)。建立 VPN 連線時，每個 VPN 連線都包含兩個通道以獲得高可用性。
3. 依照 AWS 文件中所述，選擇適當的連線選項：
   a. 如果您決定使用 AWS Direct Connect，請為您的連線選取適當的頻寬。
   b. 如果您跨多個位置使用 AWS Site-to-Site VPN 連線至 AWS 區域，請使用 加速 Site-to-Site VPN 連線 以創造改善網路效能的機會。

c. 如果您的網路設計包含 透過 AWS Direct Connect 的 IPsec VPN 連線請考慮使用私有 IP VPN 來提高安全性並實現區隔。 AWS Site-to-Site 私有 IP VPN 是以傳輸虛擬介面 (VIF) 為基礎進行部署。

d. AWS Direct Connect SiteLink 可讓您在位於全世界的資料中心之間建立低延遲的冗餘連線，方法是在 AWS Direct Connect 位置之間利用最快速的路徑傳送資料，並略過 AWS 區域。

4. 在部署到生產環境之前，先驗證您的連線設定。進行安全性和效能測試，確保其符合您的頻寬、可靠性、延遲和合規需求。
5. 定期監控連線效能和使用情況，並視需要最佳化。

決定性效能流程圖

資源

相關文件：

- 搭配 AWS 的聯網產品
- AWS Transit Gateway
- VPC 端點
- 建置可擴展且安全的多 VPC AWS 網路基礎設施
- 用戶端 VPN

相關影片：

- AWS re:Invent 2023 – 使用 AWS 建立混合網路連線能力
- AWS re:Invent 2023 – 連接 AWS 的安全遠端連線能力
- AWS re:Invent 2022 – 最佳化 Amazon CloudFront 效能
- AWS re:Invent 2019 – 連接 AWS 和混合 AWS 網路架構的連線能力
- AWS re:Invent 2020 – AWS Transit Gateway Connect

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案

- AWS 聯網研討會

PERF04-BP04 使用負載平衡將流量分配到多個資源

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、可靠性，以及有效管理和路由流量。

常見的反模式：

- 您在選擇負載平衡器類型時不考慮工作負載要求。
- 您未利用負載平衡器功能來進行效能最佳化。
- 工作負載在不使用負載平衡器的情況下，直接公開到網際網路。
- 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跨多個可用區域的應用程式流量之各式負載，並實現高可用性、自動擴展及更充分利用您的工作負載。

未建立此最佳實務時的風險暴露等級：高

實作指引

負載平衡器會做為您的工作負載的進入點，從該處將您的流量分散到後端目標，例如運算執行個體或容器，以提高使用率。

選擇正確的負載平衡器類型是最佳化架構的第一步。從列出您的工作負載特性開始，例如通訊協定 (例如 TCP、HTTP、TLS 或 WebSockets)、目標類型 (例如執行個體、容器或無伺服器)、應用程式要求 (例如長時間執行連線、使用者身分驗證或黏性) 和置放 (例如 Region、Local Zone、Outpost 或區域隔離)。

AWS 為您的應用程式提供了多種模型來使用負載平衡。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。

選擇正確的負載平衡器之後，您可以開始利用其功能來減少後端為流量提供服務所需投入的工作量。

例如，同時使用 Application Load Balancer (ALB) 和 Network Load Balancer (NLB)，您可以執行 SSL/TLS 加密卸載，這是避免您的目標完成 CPU 密集型 TLS 交握，並且改善憑證管理的機會。

在您的負載平衡器中設定 SSL/TLS 卸載時，它會負責將往返用戶端的流量進行加密，同時將未加密的流量交付給您的後端，釋放您的後端資源並且改善用戶端的回應時間。

Application Load Balancer 也可以為 HTTP/2 流量提供服務，不需要在您的目標上支援它。這個簡單的決策可以改善您的應用程式回應時間，因為 HTTP/2 更有效率地使用 TCP 連線。

定義架構時，應考慮您的工作負載延遲要求。例如，如果您有對延遲敏感的應用程式，您可能會決定使用 Network Load Balancer，以獲得極低的延遲。另外，您可能會決定藉由利用 AWS Local Zones 中的 Application Load Balancer 或甚至是 AWS Outposts，讓工作負載更靠近您的客戶。

對延遲敏感的工作負載的另一個考慮是跨區域負載平衡。使用跨區域負載平衡，每個負載平衡器節點會將已註冊目標之間的流量分散到所有允許的可用區域中。

使用與您的負載平衡器整合的 Auto Scaling。效能效率系統的其中一個關鍵層面與適當調整後端資源的規模有關。若要完成此操作，您可以利用後端目標資源的負載平衡器整合。使用與 Auto Scaling 群組整合的負載平衡器，目標會視需要從負載平衡器新增或移除，以因應傳入流量。負載平衡器也可以針對容器化工作負載與 Amazon ECS 和 Amazon EKS 整合。

- Amazon ECS - 服務負載平衡
- Amazon EKS 上的應用程式負載平衡
- Amazon EKS 上的網路負載平衡

實作步驟

- 定義您的負載平衡需求，包括流量、可用性和應用程式可擴展性。
- 為您的應用程式選擇正確的負載平衡器類型。
  - 針對 HTTP/HTTPS 工作負載使用 Application Load Balancer。
  - 針對在 TCP 或 UDP 上執行的非 HTTP 工作負載使用 Network Load Balancer。
  - 如果您想要利用兩個產品的功能，使用兩者個組合 (ALB 做為 NLB 的目標)。例如，如果您想要搭配使用 NLB 的靜態 IP 與來自 ALB 的 HTTP 標題型路由，或者如果您想要將您的 HTTP 工作負載公開到 AWS PrivateLink。
  - 如需負載平衡器的完整比較，請參閱 ELB 產品比較。
- 盡可能使用 SSL/TLS 卸載。

  - 將 HTTPS/TLS 接聽程式設定為讓 Application Load Balancer 和 Network Load Balancer 都與 AWS Certificate Manager 整合。
  - 請注意，基於合規理由，某些工作負載可能需要端對端加密。在此情況下，必須允許在目標進行加密。
  - 如需安全最佳實務，請參閱 SEC09-BP02 強制執行傳輸中加密。
- 選取正確的路由演算法 (僅 ALB)。
  - 路由演算法可以造成您的後端目標的妥善使用程度和它們影響效能程度的差異。例如，ALB 提供兩個路由演算法的選項：
  - 最低未解決請求：針對當您的應用程式的請求因複雜性而異或您的目標因處理功能而異的情況時，用來讓負載更佳地分散到您的後端目標。
  - 輪詢均衡：當請求和目標類似，或是如果您需要在目標之間平均分散請求時使用。
- 考慮跨區域或區域隔離。
  - 針對延遲改善和區域失敗網域使用跨區域關閉 (區域隔離)。在 NLB 中預設為關閉，在 ALB 中您可以依據各個目標群組加以關閉。
  - 使用跨區域開啟來增加可用性和彈性。根據預設，ALB 的跨區域為開啟，在 NLB 中您可以依據各個目標群組加以開啟。
- 為您的 HTTP 工作負載開啟 HTTP keep-alives (僅 ALB)。使用這項功能，負載平衡器可以重複使用後端連線，直到 keep-alive 逾時到期，改善您的 HTTP 請求和回應時間，同時減少您的後端目標上的資源使用率。如需如何針對 Apache 和 Nginx 執行此操作的詳細資訊，請參閱使用 Apache 或 NGINX 做為 ELB 的後端伺服器的最佳設定是什麼？
- 開啟負載平衡器的監控功能。
  - 為您的 Application Load Balancer 和 Network Load Balancer 開啟存取日誌。
  - ALB 要考慮的主要欄位是 `request_processing_time`、`request_processing_time` 和 `response_processing_time`。
  - NLB 要考慮的主要欄位是 `connection_time` 和 `tls_handshake_time`。
  - 請準備好在您需要日誌時進行查詢。您可以使用 Amazon Athena 查詢 ALB 日誌和 NLB 日誌。
  - 建立效能相關指標的警示，例如 ALB 的 `TargetResponseTime`。

資源

相關文件：

- ELB 產品比較

- AWS 全球基礎設施
- 使用可用區域親和性改善效能並且降低成本
- 使用 Amazon Athena 逐步執行日誌分析
- 查詢 Application Load Balancer 日誌
- 監控您的 Application Load Balancers
- 監控您的 Network Load Balancer
- 使用 Elastic Load Balancing 在您的 Auto Scaling 群組中的執行個體之間分散流量

相關影片：

- AWS re:Invent 2023：聯網對您的應用程式有何助益？
- AWS re:Inforce 2022：如何使用 Elastic Load Balancing 大規模增強您的安全態勢
- AWS re:Invent 2018：Elastic Load Balancing：深入探討和最佳實務
- AWS re:Invent 2021 - 如何為您的 AWS 工作負載選擇正確的負載平衡器
- AWS re:Invent 2019：針對不同工作負載充分發揮 Elastic Load Balancing

相關範例：

- Gateway Load Balancer
- 使用 Amazon Athena 進行日誌分析的 CDK 和 AWS CloudFormation 範例

PERF04-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傳輸控制協定 (TCP)，較高的延遲很可能會降低整體輸送量。有一些方法可以使用 TCP 調校和最佳化的傳輸通訊協定來解決這個問題，但有一個解決方案是透過使用者資料包協定 (UDP)。

常見的反模式：

- 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： 確認針對使用者與工作負載元件之間的通訊使用適當的通訊協定，可協助改善您的應用程式的整體使用者體驗。例如，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。

未建立此最佳實務時的曝險等級： 中

實作指引

如果您能夠為應用程式選擇不同的通訊協定，而且您有這方面的專業知識，請使用不同的通訊協定來最佳化您的應用程式和最終使用者體驗。請注意，這種方法伴隨著相當高的難度，只有在您已先利用其他方式最佳化應用程式之後，才應嘗試此方法。

改善您的工作負載效能的主要考慮是了解延遲和輸送量要求，然後選擇可最佳化效能的網路通訊協定。

考慮使用 TCP 的時機

TCP 提供可靠的資料交付，並且可用於可靠性和保證資料交付很重要的工作負載元件間通訊。許多 Web 式應用程式仰賴 TCP 型通訊協定 (例如 HTTP 和 HTTPS) 開啟 TCP 通訊端，以在應用程式元件之間進行通訊。電子郵件和檔案資料傳輸是常見的應用程式，同樣會使用 TCP，因為 TCP 是應用程式元件之間簡單又可靠的傳輸機制。使用 TLS 與 TCP 會對通訊增加一些負擔，導致提高延遲和降低輸送量，但也會帶來安全上的優勢。負擔主要來自交握處理的增加負擔，需要數個往返才能完成。一旦交握完成，加密和解密資料的負擔相對小。

考慮使用 UDP 的時機

UDP 是無連線導向的通訊協定，因此適用於需要快速、有效傳輸的應用程式，例如日誌、監控和 VoIP 資料。此外，如果您有會回應來自大量用戶端之小型查詢的工作負載元件，請考慮使用 UDP，以確保最佳工作負載效能。資料包傳輸層安全性 (DTLS) 是等同於 Transport Layer Security (TLS) 的 UDP。使用 DTLS 與 UDP 時，負擔是來自加密和解密資料，因為交握處理已簡化。DTLS 也會對 UDP 封包增加小量負擔，因為 DTLS 包含額外欄位，可指出安全參數及偵測竄改。

考慮使用 SRD 的時機

Scalable Reliable Datagram (SRD) 是針對高輸送量工作負載最佳化的網路傳輸通訊協定，能夠在多個路徑之間負載平衡流量，並且快速從封包捨棄或連結失敗復原。因此 SRD 最適合用於需要運算節點之間高輸送量和低延遲通訊的高效能運算 (HPC) 工作負載。這可能包含平行處理任務，例如牽涉到在節點之間大量資料傳輸的模擬、建模和資料分析。

實作步驟

1. 使用 AWS Global Accelerator 和 AWS Transfer Family 服務來改善您的線上檔案傳輸應用程式的輸送量。AWS Global Accelerator 服務可協助您在用戶端裝置與 AWS 上工作負載之間實現較低的

延遲。使用 AWS Transfer Family，您可以使用 TCP 型通訊協定，例如 Secure Shell File Transfer Protocol (SFTP) 和 File Transfer Protocol over SSL (FTPS)，安全地擴展和管理與 AWS 儲存服務之間的檔案傳輸。

2. 使用網路延遲來判斷 TCP 是否適合工作負載元件之間的通訊。如果您的用戶端應用程式與伺服器之間的網路延遲高，則 TCP 三向交握會耗費一些時間，因此會影響您的應用程式的回應能力。例如到第一個位元組的時間 (TTFB) 和往返時間 (RTT) 等指標可用來測量網路延遲。如果您的工作負載為使用者提供動態內容的服務，請考慮使用 Amazon CloudFront，這會建立與每個動態內容來源的持久性連線，移除會拖慢每個用戶端請求速度的連線設定時間。
3. 使用 TLS 與 TCP 或 UDP 會由於加密和解密的影響，導致對您的工作負載增加延遲和減少輸送量。針對此類工作負載，請考慮 Elastic Load Balancing 上的 SSL/TLS 卸載，藉由允許負載平衡器處理 SSL/TLS 加密和解密程序而不是讓後端執行個體進行處理，來改善工作負載效能。這可協助減少後端執行個體上的 CPU 使用率，可以改善效能和增加容量。
4. 使用 Network Load Balancer (NLB) 來部署依賴 UDP 通訊協定的服務，例如身分驗證和授權、記錄、DNS、IoT 和串流媒體，改善您的工作負載的效能和可靠性。NLB 會在多個目標之間分散傳入 UDP 流量，讓您水平地擴展工作負載、增加容量以及減少單一目標的負擔。
5. 針對您的高效能運算 (HPC) 工作負載，請考慮使用 彈性網路介面卡 (ENA) 快速版 功能，該功能使用 SRD 通訊協定來改善網路效能，方法是針對 EC2 執行個體之間的網路流量，提供較高的單一流量頻寬 (25Gbps) 和較低的尾端延遲 (99.9 百分位數)。
6. 使用 Application Load Balancer (ALB) 來路由和負載平衡工作負載元件之間的 gRPC (遠端程序呼叫) 流量或 gRPC 用戶端與服務之間的流量。gRPC 會使用 TCP 型 HTTP/2 通訊協定進行傳輸，並且提供如較輕網路足跡、壓縮、有效二進位序列化、支援多種語言和雙向串流的優點。

資源

相關文件：

- 如何將 UDP 流量路由到 Kubernetes
- 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
- 搭配 AWS 的聯網產品

- 轉換到 Amazon Route 53 中的 Latency-Based Routing
- VPC 端點

相關影片：

- AWS re:Invent 2022 – 擴展新一代 Amazon Elastic Compute Cloud 執行個體的網路效能
- AWS re:Invent 2022 – 應用程式聯網基礎

相關範例：

- AWS Transit Gateway 和可擴展的安全解決方案
- AWS 聯網研討會

PERF04-BP06 根據網路需求選擇工作負載的位置

評估資源置放的選項以減少網路延遲和提高輸送量，藉由減少頁面載入和資料傳輸時間來提供最佳的使用者體驗。

常見的反模式：

- 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：使用者體驗因使用者與您的應用程式之間的延遲而大受影響。透過使用適當的 AWS 區域 和 AWS 私有全球網路，就可以減少延遲並為遠程使用者提供更優質的體驗。

未建立此最佳實務時的風險暴露等級：中

實作指引

例如 Amazon EC2 執行個體的資源會放到 AWS 區域、AWS Local Zones、AWS Outposts 或 AWS Wavelength 區域內的可用區域。此位置的選擇會影響來自特定使用者位置的網路延遲和輸送量。Amazon CloudFront 和 AWS Global Accelerator 等的邊緣服務也可以用來改善網路效能，做法是在邊緣節點快取內容，或透過 AWS 全球網路為使用者提供工作負載的最佳路徑。

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。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，可讓工作負載加入低延遲、低抖動的 25 Gbps 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。

對延遲敏感的服務是在邊緣節點使用 AWS 全球網路交付，例如 Amazon CloudFront。這些節點通常可提供如內容交付網路 (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，工作負載就可以在低延遲的情況下回應內容或 DNS 解析的請求。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)，或以延遲為基礎的路由，將最終使用者定向到最近區域的 (最小延遲)。

使用邊緣服務來減少延遲及啟用內容快取。為 DNS 和 HTTP/HTTPS 正確設定快取控制，以從這些方法中獲得最大的效益。

實作步驟

- 擷取與往返網路介面的 IP 流量有關的資訊。
  - 使用 VPC Flow Logs 記錄 IP 流量
  - 在 AWS Global Accelerator 中保留用戶端 IP 位址的方式
- 分析您工作負載中的網路存取模式，以識別使用者如何使用您的應用程式。
  - 使用監控工具 (例如 Amazon CloudWatch 和 AWS CloudTrail) 收集網路活動的相關資料。
  - 分析資料以識別網路存取模式。
- 根據下列關鍵元素，為您的工作負載部署選取區域：
  - 資料的所在位置：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。
  - 使用者的所在位置：對於面向使用者的應用程式，請選擇接近工作負載使用者的一或多個區域。
  - 其他限制：請考量成本和合規性之類的限制，相關說明請見為工作負載選取區域時應考量的事項。
- 使用 AWS Local Zones 執行影片轉譯等工作負載。Local Zones 可讓您因運算和儲存資源更接近最終使用者而獲益。
- AWS Outposts 適用於需要保持內部部署的工作負載，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。
- 像是高解析度即時影片串流、高保真度音訊和擴增實境或虛擬實境 (AR/VR) 等應用程式，需要 5G 裝置的極低延遲。針對此類應用程式，請考慮 AWS Wavelength。AWS Wavelength 會將 AWS 運算及儲存服務嵌入 5G 網路，為開發、部署和擴展極低延遲應用程式提供行動裝置邊緣運算基礎設施。
- 使用本機快取或 AWS 快取解決方案取得常用資產，以提升效能、減少資料移動，以及降低環境影響。

Service	When to use
Amazon CloudFront	用來快取靜態內容 (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)。
Amazon ElastiCache	用來快取 Web 應用程式的內容。
DynamoDB Accelerator	用來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，如下所示：

Service	When to use
Lambda@edge	用於在物件未經快取時起始的大量運算作業。
Amazon CloudFront 函數	用於簡單的使用案例，例如可由短期函數起始的 HTTP(s) 請求或回應操作。
AWS IoT Greengrass	用來為連線的裝置執行本機運算、傳訊和資料快取。

- 某些應用程式需要藉由減少第一個位元組延遲和抖動並且增加輸送量，來獲得固定的進入點或較高的效能。這些應用程式可以從在邊緣節點提供靜態任播 IP 位址和 TCP 終止的網路服務獲益。AWS Global Accelerator 可以改善您的應用程式效能高達 60%，並且提供多區域架構的快速容錯移轉。AWS Global Accelerator 提供靜態任播 IP 地址，做為一或多個 AWS 區域 中託管應用程式的固定進入點。這些 IP 位址允許流量傳入盡可能靠近您的使用者的 AWS 全球網路。AWS Global Accelerator 會藉由建立用戶端與最接近用戶端之 AWS 邊緣節點之間的 TCP 連線，減少初始連線設定時間。請檢閱 AWS Global Accelerator 的使用情形，以改善您的 TCP/UDP 工作負載的效能，並且提供多區域架構的快速容錯移轉。

資源

相關的最佳實務：

- COST07-BP02 根據成本實作區域
- COST08-BP03 實作可降低資料傳輸成本的服務

- 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
- SUS01-BP01 根據業務需求和永續性目標選擇區域
- SUS02-BP04 根據工作負載的聯網需求最佳化其地理位置
- SUS04-BP07 將跨網路的資料移動降到最少

相關文件：

- AWS 全球基礎設施
- AWS Local Zones 和 AWS Outposts，為您的邊緣工作負載選擇正確的技術
- 置放群組
- AWS Local Zones
- AWS Outposts
- AWS Wavelength
- Amazon CloudFront
- AWS Global Accelerator
- AWS Direct Connect
- AWS Site-to-Site VPN
- Amazon Route 53

相關影片：

- AWS Local Zones 解說影片
- AWS Outposts：概觀和運作方式
- AWS re:Invent 2023 - 邊緣與內部部署工作負載的遷移策略
- AWS re:Invent 2021 - AWS Outposts：在內部部署環境帶來 AWS 體驗
- AWS re:Invent 2020：AWS Wavelength：在 5G 邊緣以極低延遲執行應用程式
- AWS re:Invent 2022 - AWS Local Zones：為分散的邊緣建置應用程式
- AWS re:Invent 2021 - 使用 Amazon CloudFront 建置低延遲網站
- AWS re:Invent 2022 - 使用 AWS Global Accelerator 改善效能與可用性
- AWS re:Invent 2022 - 使用 AWS 建置您的全球廣域網路
- AWS re:Invent 2020：使用 Amazon Route 53 進行全球流量管理

相關範例：

- AWS Global Accelerator 自訂路由研討會
- 使用邊緣功能處理重新撰寫和重新導向

PERF04-BP07 根據指標最佳化網路組態

使用收集和分析的資料來做出有關最佳化網路組態的明智決策。

常見的反模式：

- 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。
- 您針對所有網路服務使用預設組態。
- 您過度佈建網路資源來提供足夠的容量。

建立此最佳實務的優勢： 收集您的 AWS 網路的必要指標和實作網路監控工具，可讓您了解網路效能和最佳化網路組態。

未建立此最佳實務時的曝險等級： 低

實作指引

監控往返 VPC、子網路或網路界面的流量，對於了解如何利用 AWS 網路資源和如何最佳化網路組態而言非常重要。使用下列 AWS 聯網工具，即可進一步檢查流量用量、網路存取和日誌的相關資訊。

實作步驟

- 確定要收集的關鍵績效指標，例如延遲或封包遺失。AWS 提供了幾種工具，可幫助您收集這些指標。使用下列工具，即可進一步檢查流量用量、網路存取和日誌的相關資訊：

AWS 工具	在何處使用
Amazon VPC IP Address Manager。	使用 IPAM 來規劃、追蹤和監控您的 AWS 和內部部署工作負載的 IP 地址。這是最佳化 IP 地址用量和分配的最佳實務。
VPC Flow Logs	使用 VPC Flow Logs 來擷取有關往返您的 VPC 中網路介面流量的詳細資訊。使用 VPC

AWS 工具	在何處使用
	Flow Logs 可診斷過於嚴格或寬鬆的安全群組規則，並且判斷往返網路介面的流量方向。
AWS Transit Gateway Flow Logs	使用 AWS Transit Gateway Flow Logs 擷取往返傳輸閘道的 IP 流量相關資訊。
DNS 查詢記錄	記錄 Route 53 所收到的公有或私有 DNS 查詢的相關資訊。使用 DNS 日誌，您可以藉由了解請求的網域或子網域或是回應 DNS 查詢的 Route 53 邊緣節點，最佳化 DNS 組態。
Reachability Analyzer	Reachability Analyzer 可協助您分析和偵錯網路連線能力。Reachability Analyzer 是組態分析工具，可讓您執行您的 VPC 中來源資源與目的地資源之間的連線能力測試。此工具可協助您確認您的組態符合您預期的連線能力。
Network Access Analyzer	Network Access Analyzer 可協助您了解資源的網路存取。您可以使用 Network Access Analyzer 來指定您的網路存取要求，並識別未符合您的指定要求的潛在網路路徑。藉由最佳化您的對應網路組態，您可以了解及確認網路的狀態，並且示範 AWS 上的網路是否符合您的合規要求。
Amazon CloudWatch	使用 Amazon CloudWatch 並且為網路選項開啟適當的指標。請確定為您的工作負載選擇正確的網路指標。例如，您可以開啟 VPC 網路地址用量、VPC NAT 閘道、AWS Transit Gateway、VPN 通道、AWS Network Firewall、Elastic Load Balancing 和 AWS Direct Connect 等指標。持續監控指標是觀察和了解您的網路狀態和用量的良好實務，可協助您根據您的觀察來最佳化網路組態。

AWS 工具	在何處使用
AWS Network Manager	您可以使用 AWS Network Manager 監控 AWS 全球網絡 的即時和歷史效能，以因應操作和規劃用途。Network Manager 提供了 AWS 區域與可用區域之間及每個可用區域內的彙總網路延遲，讓您更了解應用程式效能與基礎 AWS 網路效能之間的關係。
Amazon CloudWatch RUM	使用 Amazon CloudWatch RUM 收集指標來為您提供洞見，以協助您識別、了解和改善使用者體驗。

- 使用 VPC 和 AWS Transit Gateway Flow Logs 找出最活躍的發言者和應用程式流量模式。
- 評估並最佳化目前的網路架構，包括 VPC、子網路和路由。例如，您可以評估不同的 VPC 對等互連或 AWS Transit Gateway 如何協助您改善架構中的網路。
- 評估網路中的路由路徑，以確認目的地之間一律使用最短路徑。Network Access Analyzer 可幫助您實現這點。

資源

相關文件：

- 公有 DNS 查詢記錄
- 什麼是 IPAM？
- 什麼是 Reachability Analyzer？
- 什麼是 Network Access Analyzer？
- 您的 VPC 的 CloudWatch 指標
- 使用 Apache Parquet 格式的 VPC Flow Logs 針對網路分析最佳化效能和降低成本
- 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：

- AWS re:Invent 2023 – 開發人員的雲端聯網指南
- AWS re:Invent 2023 – 準備好踏出下一步了嗎？ 設計可促進成長與靈活性的網路

- AWS re:Invent 2023 – 進階 VPC 設計及新功能
- AWS re:Invent 2022 – - 深入了解 AWS 聯網基礎設施
- AWS re:Invent 2020 – 搭配 AWS Well-Architected Framework 的聯網最佳實務和秘訣
- AWS re:Invent 2020 – 監控網路流量並對其進行疑難排解

相關範例：

- AWS 聯網研討會
- AWS 網路監控
- 觀察和診斷在 AWS 上的網路
- 尋找並解決在 AWS 上的網路設定錯誤

程序和文化

PERF 5.您的組織實務和文化如何促進工作負載的效能達成效率？

在架構工作負載時，您可以採取一些原則和實務來協助您更有效率地執行高效能雲端工作負載。若要培養文化來促進雲端工作負載的效能達成效率，請考慮下列重要原則和實務：

最佳實務

- PERF05-BP01 建立關鍵績效指標 (KPI) 以衡量工作負載健康狀態和效能
- PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域
- PERF05-BP03 定義提高工作負載效能的程序
- PERF05-BP04 對工作負載執行負載測試
- PERF05-BP05 使用自動化主動修復效能相關問題
- PERF05-BP06 即時更新工作負載和服務的狀態
- PERF05-BP07 定期檢閱指標

PERF05-BP01 建立關鍵績效指標 (KPI) 以衡量工作負載健康狀態和效能

找出定量和定性衡量工作負載效能的 KPI。KPI 可協助您衡量與業務目標相關之工作負載的健康狀態和效能。

常見的反模式：

- 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，並作為標準指標資料分享。
- 您未定義可量化且可衡量的 KPI。
- 您沒有將 KPI 與業務目標或策略保持一致。

建立此最佳實務的好處：找出代表工作負載健康狀態和效能的特定 KPI，有助團隊以一致的標準排定優先事項並定義成功的業務成果。與所有部門分享這些指標，可提供對閾值、期望和業務影響的可見性和一致性。

未建立此最佳實務時的風險暴露等級：高

實作指引

KPI 可讓業務和工程團隊以一致的標準衡量目標和策略，以及將這些因素結合以產生商業成果的方式。例如，網站工作負載可能使用頁面載入時間，作為整體效能的指示。此指標將是衡量最終使用者體驗的多個資料點之一。除了找出頁面載入時間閾值外，您還應該記錄未符合理想效能時預期的成果或業務風險。較長的頁面載入時間會直接影響最終使用者，降低他們的使用者體驗評級，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都列入考慮。

團隊必須使用即時精密資料和歷史資料作為參考，來評估工作負載 KPI，並建立儀表板，針對 KPI 資料執行指標數學，以衍生營運和使用率洞察。KPI 應該加以記錄，並包含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。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。

實作步驟

- 識別利害關係人：識別並記錄關鍵業務利害關係人，包括開發和營運團隊。
- 定義目標：與利害關係人合作，以定義和記錄工作負載的目標。思考工作負載的關鍵績效，例如輸送量、回應時間和成本，以及業務目標 (如使用者滿意度)。
- 檢閱業界最佳實務：檢閱業界最佳實務以識別符合您工作負載目標的相關 KPI。
- 識別指標：識別與工作負載目標相符的指標，並可協助衡量績效和業務目標的指標。根據這些指標建立 KPI。範例指標是指像平均回應時間或並行使用者人數之類的量測值。
- 定義並記錄 KPI：使用業界最佳實務和工作負載目標，為您的工作負載 KPI 設立目標。使用此資訊，來設定嚴重性或警示層級的 KPI 閾值。找出並記錄未符合 KPI 時的風險和影響。
- 實作監控：使用 Amazon CloudWatch 或 AWS Config 之類的監控工具來收集指標並衡量 KPI。

- 以視覺化方式溝通關鍵績效指標：使用 Amazon QuickSight 等儀表板工具以視覺化方式呈現，並與利害關係人溝通 KPI 議題。
- 分析和最佳化：定期審查和分析 KPI，以確定需要改善的工作負載區域。與利害關係人合作執行這些改進措施。
- 重新檢視和調整：定期檢閱指標和 KPI 以評估其有效性，尤其是當業務目標或工作負載績效發生變化時。

資源

相關文件：

- CloudWatch 文件
- 監控、記錄和效能 AWS Partner
- AWS 可觀測性工具
- 關鍵績效指標 (KPI) 對大規模雲端遷移的重要性
- 如何使用 KPI 儀表板追蹤成本最佳化 KPI
- X-Ray 文件
- 使用 Amazon CloudWatch 儀表板
- Amazon QuickSight KPI

相關影片：

- AWS re:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度
- AWSre:Invent 2023 - 使用 AWS Health 以大規模管理資源生命週期事件
- AWS re:Invent 2023 - Pinterest 的效能與效率：最佳化最新執行個體
- AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果
- AWS re:Invent 2023 - 打造有效的可觀測性策略
- AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控
- AWS re:Invent 2023 - 為前一千萬名使用者擴展 AWS
- AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能
- 為您的業務建立有效的指標策略 | AWS 活動

相關範例：

- 使用 Amazon QuickSight 建立儀表板

PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域

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。

常見的反模式：

- 您假設標準運算指標 (例如 CPU 使用率或記憶體壓力) 足以找出效能問題。
- 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。

建立此最佳實務的好處：了解關鍵效能領域可協助工作負載擁有者監控 KPI 並優先處理具有高影響力的待改善之處。

未建立此最佳實務時的風險暴露等級：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。

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。當您已調查這些面向時，請審視自己可以部署哪個解決方案，來消除這些效能疑慮。

實作步驟

- 設置端到端監控，來擷取所有工作負載組成部分和指標。以下是 AWS 上的監控解決方案的範例。

Service	Where to use
Amazon CloudWatch 實際使用者監控 (RUM)	To capture application performance metrics from real user client-side and frontend sessions.
AWS X-Ray	To trace traffic through the application layers and identify latency between components

Service	Where to use
	and dependencies. Use X-Ray service maps to see relationships and latency between workload components.
Amazon Relational Database Service 績效詳情	To view database performance metrics and identify performance improvements.
Amazon RDS 增強型監控	To view database OS performance metrics.
Amazon DevOps Guru	To detect abnormal operating patterns so you can identify operational issues before they impact your customers.

- 執行測試，來產生指標、確定流量模式、瓶頸和關鍵效能區域。以下是如何執行測試的幾個範例：
  - 設定 CloudWatch Synthetic Canaries，以程式設計方式使用 Linux Cron 任務或評分運算式，模擬以瀏覽器為基礎的使用者活動，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的成長速率測試工作負載。
- 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。
- 進行效能改善的實驗，並透過資料來衡量這些變更。舉例來說，您可以使用 CloudWatch Evidently，測試新的改善項目和對工作負載的效能影響。

資源

相關文件：

- re:Invent 2023 中 AWS 可觀測性的最新消息
- Amazon 建置者資料中心
- X-Ray 文件
- Amazon CloudWatch RUM
- Amazon DevOps Guru

相關影片：

- AWS re:Invent 2023 - [LAUNCH] 現代工作負載的應用程式監控

- AWS re:Invent 2023 - 實作應用程式可觀測性
- AWS re:Invent 2023 - 打造有效的可觀測性策略
- AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控
- AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果
- AWS re:Invent 2022 - Amazon 建置者資料中心：25 年的 Amazon 卓越營運
- AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能
- 使用 Amazon CloudWatch Synthetics 進行應用程式的視覺監控

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 Python
- AWS 上的分散式負載測試

PERF05-BP03 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。

常見的反模式：

- 您假設目前的架構是靜態的，且不會隨著時間而更新。
- 您會隨時間導入架構變更，而且無須指標佐證。

建立此最佳實務的好處：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設計。

未建立此最佳實務時的風險暴露等級：中

實作指引

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。

找出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。

實作步驟

- 識別 KPI： 按照PERF05-BP01 建立關鍵績效指標 (KPI) 以衡量工作負載健康狀態和效能中所述，找出工作負載效能 KPI，以設立工作負載基準。
- 實作監控：使用AWS 可觀測性工具收集效能指標並衡量 KPI。
- 執行分析：執行深入分析，以找出工作負載中效能不佳的區域 (例如組態和應用程式程式碼)，如 PERF05-BP02 使用監控解決方案了解效能扮演關鍵作用的領域 所述。使用分析和效能工具，找出效能改善策略。
- 驗證改善項目：使用沙盒或試生產環境來驗證改善策略的有效性。
- 實作變更：實作生產中的變更，並持續監控工作負載的效能。記錄改善項目並與利害關係人溝通這些變更。
- 重新檢視和調整：定期檢閱您的績效改進流程，以確定需要增強的區域。

資源

相關文件：

- AWS 部落格
- AWS 最新消息
- AWS Skill Builder

相關影片：

- AWS re:Invent 2022 - 提供永續且高效能的架構
- AWS re:Invent 2023 - 最佳化成本和效能，並追蹤成功緩解的進度
- AWS re:Invent 2022 - AWS 最佳化：可採取動作的步驟，以便立即獲得成果
- AWS re:Invent 2022 - 利用最佳實務指引最佳化您的 AWS 工作負載

相關範例：

- AWS Github

PERF05-BP04 對工作負載執行負載測試

對工作負載執行負載測試，以確認可以處理生產負載並找出任何效能瓶頸。

常見的反模式：

- 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。
- 您可以執行負載測試，不需參閱Amazon EC2測試政策，也不必提交模擬事件提交表單。這會導致測試無法執行，因為它看起來像拒絕服務事件。

建立此最佳做法的好處：在負載測試下測量效能時，會顯示您將在負載增加到何種程度時受到影響。這可在變更影響工作負載之前，讓您先預測所需的變更。

未建立此最佳實務時的風險暴露等級：低

實作指引

雲端中的負載測試是在實際條件下，以預期的使用者負載來衡量雲端工作負載效能的程序。此程序包括佈建類似生產環境的雲端環境、使用負載測試工具產生負載，以及分析指標，以便評估您的工作負載處理實際負載的能力。必須使用生產資料的綜合或處理過的版本 (移除敏感或可識別身分的資訊) 執行負載測試。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。此程序可幫助您繼續實現所需的效能。

實作步驟

- 定義測試目標：識別您要評估的工作負載效能，例如輸送量和回應時間。
- 選取測試工具：選擇並設定適合您工作負載的負載測試工具。
- 設定您的環境：根據您的生產環境設定測試環境。您可以使用 AWS 服務執行生產規模的環境，進而測試架構。
- 實作監控：使用監控工具，例如 Amazon CloudWatch 收集架構中資源之間的指標。您也可以收集和發佈自訂指標。
- 定義情境：定義負載測試方案和參數 (如測試持續時間和使用者數量)。
- 進行負載測試：大規模執行測試情境。利用 AWS 雲端 測試工作負載，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。
- 分析測試結果：分析結果以找出效能瓶頸和需要改善的區域。
- 記錄並共享調查結果：記錄和報告調查結果和建議。與利害關係人共享此資訊，協助他們對效能最佳化策略做出明智的決定。

- 不斷反覆執行：負載測試應以規律的節奏執行，尤其是在系統更改更新後。

資源

相關文件：

- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分散式負載測試

相關影片：

- AWS Summit ANZ 2023：透過 AWS 分散式負載測試讓您放心加快腳步
- AWS re:Invent 2022 - 為您的前一千萬名使用者擴展 AWS
- 使用 AWS 解決方案解決問題：分散式負載測試

- AWS re:Invent 2021 - 透過最終使用者洞察與 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

PERF05-BP05 使用自動化主動修復效能相關問題

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。

常見的反模式：

- 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。

建立此最佳實務的好處：主動修復警示動作，可讓支援人員專注在無法自動採取行動的項目上。有助操作人員處理所有警示，不會不堪負荷，而能專注在重要警示。

未建立此最佳實務時的風險暴露等級：低

實作指引

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，您可能有可以預測關鍵績效指標 (KPI) 預期值，且會在超過特定閾值時發出警示的系統，或是在 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。

實作步驟

- 識別修復工作流程：識別並了解可自動修復的效能問題。使用 Amazon CloudWatch 或 AWS X-Ray 等 AWS 監控解決方案，以協助您更完整了解問題的根本原因。
- 定義自動化程序：建立可用於自動修正問題的逐步修復程序。
- 設定啟動事件：設定事件以自動啟動修復程序。例如，您可以定義觸發程式，在執行個體達到特定 CPU 使用率閾值時自動重新啟動執行個體。
- 自動化修復：使用 AWS 服務和技術以自動化修復程序。例如，AWS Systems Manager 自動化提供安全且可擴展的方式，來自動化修復程序。如果變更未成功解決問題，請務必使用自我修復邏輯以回復變更。
- 測試工作流程：在試生產環境中測試自動修復程序。
- 實作工作流程：在生產環境中實作自動修復。
- 開發手冊：開發並記錄一本手冊，其中概述修復計劃的步驟，包括啟動事件、修復邏輯和採取的動作。務必培訓利害關係人，協助他們有效回應自動修復事件。
- 審查和調整：定期評估自動修復工作流程的有效性。視需要調整啟動事件和修復邏輯。

資源

相關文件：

- CloudWatch 文件
- 監控、記錄和效能 AWS Partner Network 合作夥伴
- X-Ray 文件
- 使用 CloudWatch 中的警示和警示動作
- 建置雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- 透過自動表格最佳化，自動調整 Amazon Redshift 效能

相關影片：

- AWS re:Invent 2023 - 自動擴展、修復和智慧型自我修復策略
- AWS re:Invent 2023 - [LAUNCH] 現代工作負載的應用程式監控
- AWS re:Invent 2023 - 實作應用程式可觀測性
- AWS re:Invent 2021 - 以明智的方式自動化雲端操作
- AWS re:Invent 2022 - 在 AWS 環境中大規模設定控制項
- AWS re:Invent 2022 - 使用 AWS 以自動化修補程式管理及合規作業
- AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能
- AWS re:Invent 2023 - 減輕負擔：使用 Amazon RDS 診斷並解決效能問題
- AWS re:Invent 2021 - {新啟動} 使用 Amazon DevOps Guru 自動偵測並解決問題
- AWS re:Invent 2023 - 集中化您的營運

相關範例：

- CloudWatch Logs 自訂警示

PERF05-BP06 即時更新工作負載和服務的狀態

掌握最新的雲端服務和特徵，以採用高效功能、解決問題，並改善工作負載的整體效能效率。

常見的反模式：

- 您假設目前的架構是靜態的，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。

建立此最佳實務的好處：透過建立程序掌握新服務和供應項目的最新資訊，您就可以採用新特徵和功能、解決問題並改善工作負載效能。

未建立此最佳實務時的風險暴露等級：低

實作指引

當新服務、設計模式和產品特徵推出時，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，建立新技術的使用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。

實作步驟

- 清查工作負載：清查工作負載軟體和架構，並識別需要更新的元件。
- 確定更新來源：找出與工作負載元件相關的新聞和更新來源。舉例來說，您可以訂閱 AWS 最新消息部落格，以了解與您工作負載元件相符的產品。您可以訂閱 RSS 摘要或管理電子郵件訂閱。
- 定義更新排程：定義排程以評估工作負載的新服務和特徵。
  - 您可以使用 AWS Systems Manager Inventory，從 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軟體政策所需的軟體與組態，以及哪些執行個體需要更新。
- 評估新的更新：了解如何更新工作負載的元件。利用雲端的靈活性快速測試新特徵對工作負載有何改善，藉以提高效能效率。
- 使用自動化：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。
  - 您可以使用 CI/CD 自動更新 AMI、容器映像，以及其他與雲端應用程式有關的成品。
  - 您可以使用 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。
- 記錄程序：記錄您在評估更新和新服務的過程。向擁有者提供所需的時間和空間，來研究、測試、試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。

資源

相關文件：

- AWS 部落格
- AWS 最新消息
- 利用自動化 EC2 Image Builder 管道實作最新影像

相關影片：

- AWS re:Inforce 2022 - 使用 AWS，自動化修補程式管理與合規
- 所有物件修補程式：AWS Systems Manager| AWS 事件

相關範例：

- 庫存與修補程式管理

- 一個觀察工作坊

PERF05-BP07 定期檢閱指標

作為日常維護或對事件或事故的回應，檢閱所收集的指標。透過這些審查來識別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) 將有助找出、解決或預防問題。

常見的反模式：

- 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。

建立此最佳實務的好處：持續檢閱正在收集的指標，以確認指標的識別和處理是否正確或防止問題發生。如果讓指標長時間持續處於警示狀態，指標也會變得過時。

未建立此最佳實務時的風險暴露等級：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，從而防止事故發生或更快地解決將來的事故。

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，進而防止事故發生或更快地解決將來的事故。

實作步驟

- 定義指標：定義與工作負載目標一致的關鍵績效指標以利於監視，包括回應時間和資源使用率等指標。
- 建立基準：為每個指標設定基準和所需的值。基準應提供參考點以識別偏差或異常狀況。
- 設定節奏：設定節奏 (例如每週或每月一次) 以審查重要指標。
- 識別效能問題：在每次審查期間評估趨勢以及與基準值的偏差。查看是否有任何效能瓶頸或異常情況。對於已確認的問題，請展開深入根本原因分析，以了解問題背後的主要原因。
- 識別糾正動作：使用您的分析來識別糾正動作。這可能包括參數調整、修復錯誤和調整資源規模。
- 記錄調查結果：記錄您的調查結果，包括已識別的問題、根本原因和糾正動作。
- 反覆執行與改善：持續評估並改善指標審查流程。利用先前審查中學到的教訓，以隨時間強化流程。

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
- 使用 CloudWatch Metrics Insights 查詢您的指標
- 監控、記錄和效能 AWS Partner Network 合作夥伴
- X-Ray 文件

相關影片：

- AWS re:Invent 2022 - 在 AWS 環境中大規模設定控制項
- AWS re:Invent 2022 - Amazon 如何使用更完善的指標來改善網站效能
- AWS re:Invent 2023 - 打造有效的可觀測性策略
- AWS Summit SF 2022 - 使用 AWS 的完整堆疊可觀測性和應用程式監控
- AWS re:Invent 2023 - 減輕負擔：使用 Amazon RDS 診斷並解決效能問題

相關範例：

- 使用 Amazon QuickSight 建立儀表板
- CloudWatch 儀表板

成本最佳化

成本最佳化支柱包括以最低價格執行系統來產生商業價值的能力。您可以在下列白皮書中找到規範指引： 成本最佳化支柱白皮書。

最佳實務領域

- 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，就可幫助組織最佳化成本和用量，並且在 AWS 上進行規模調整，進而實現商業價值和財務上的成功。

最佳實務

- COST01-BP01 建立優化成本的負責團隊
- 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-BP08 建立成本感知文化
- COST01-BP09 量化成本最佳化所帶來的商業價值

COST01-BP01 建立優化成本的負責團隊

建立負責建立並維護整個組織的成本感知的團隊 (雲端商業辦公室、雲端卓越中心或 FinOps 團隊)。成本最佳化的負責人可以是個人或是團隊，條件是必須是來自財務、技術或業務團隊，且了解整個組織和雲端財務的人員。

未建立此最佳實務時的曝險等級： 高

實作指引

這份說明會介紹雲端商業辦公室 (CBO) 或雲端卓越中心 (CCOE) 的職能，或是負責建立並維護雲端運算成本感知文化的團隊。這個職能可以是現有個人、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) 會優先並花費一定比例的時間，進行成本管理和成本最佳化活動。相較於大型企業的全職職能部門，小型組織的此職能部門花費的時間比例可能較少。

此職能部門 (個人或團隊) 會優先並花費一定比例的時間，進行成本管理和成本最佳化活動。相較於大型企業的全職職能部門，小型組織的此職能部門在成本管理、及最佳化活動方面花費的時間比例可能較少。

此職能部門必須採行跨領域合作的方法，並要具備專案管理、資料科學、財務分析和軟體或基礎架構開發等能力。藉此，在三種不同的所有權下執行成本最佳化，以改善工作負載效率：

- 集中式： 透過 FinOps 團隊、雲端財務管理 (CFM) 團隊、雲端商業辦公室 (CBO) 或雲端卓越中心 (CCoE) 等特設團隊，客戶可以設計和實作管控機制，並在全公司推動最佳實務。
- 分散式： 影響技術團隊，進行成本最佳化。
- 混合: 結合集中式與去中心化方法，讓團隊互相合作，進行成本最佳化。

可以根據成本最佳化目標 (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，這是成功的關鍵因素。高層支持者會成為運用雲端服務節省成本的推動者，並替團隊提報支援，確保成本最佳化活動獲組織認定為優先要務。否則，相關的方針可能不會受到重視，且節省成本將不會被列為優先要務。高層支持者和這個團隊共同協助您的組織，讓其得以聰明高效地使用雲端，並提供商業價值。

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： 貴組織的所有相關人員都必須貢獻己力，進一步了解成本管理。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)。有些團隊必須全職參與 (財務或技術)，有些團隊則可視需要定期參與。執行 CFM 的個人或團隊需要具備下列技能：
  - 軟體開發： 如果您正在建構指令碼和自動化程序。
  - 基礎架構工程： 用以部署指令碼、自動化程序，並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成本最佳化的教育計畫，制定整個組織的標準 (例如成本最佳化的監控和報告)，以及設定工作負載最佳化目標。此職能部門也需要定期向組織報告其成本最佳化的能力。

您可以定義以價值或成本衡量的關鍵績效指標 (KPI)。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連結商業價值驅動力，協助釐清變更 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、整體財務和最佳化指標。然後，再更詳細地報告關鍵工作負載。

  在這類定期會談中，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為整個組織提供實質意義的價值 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。

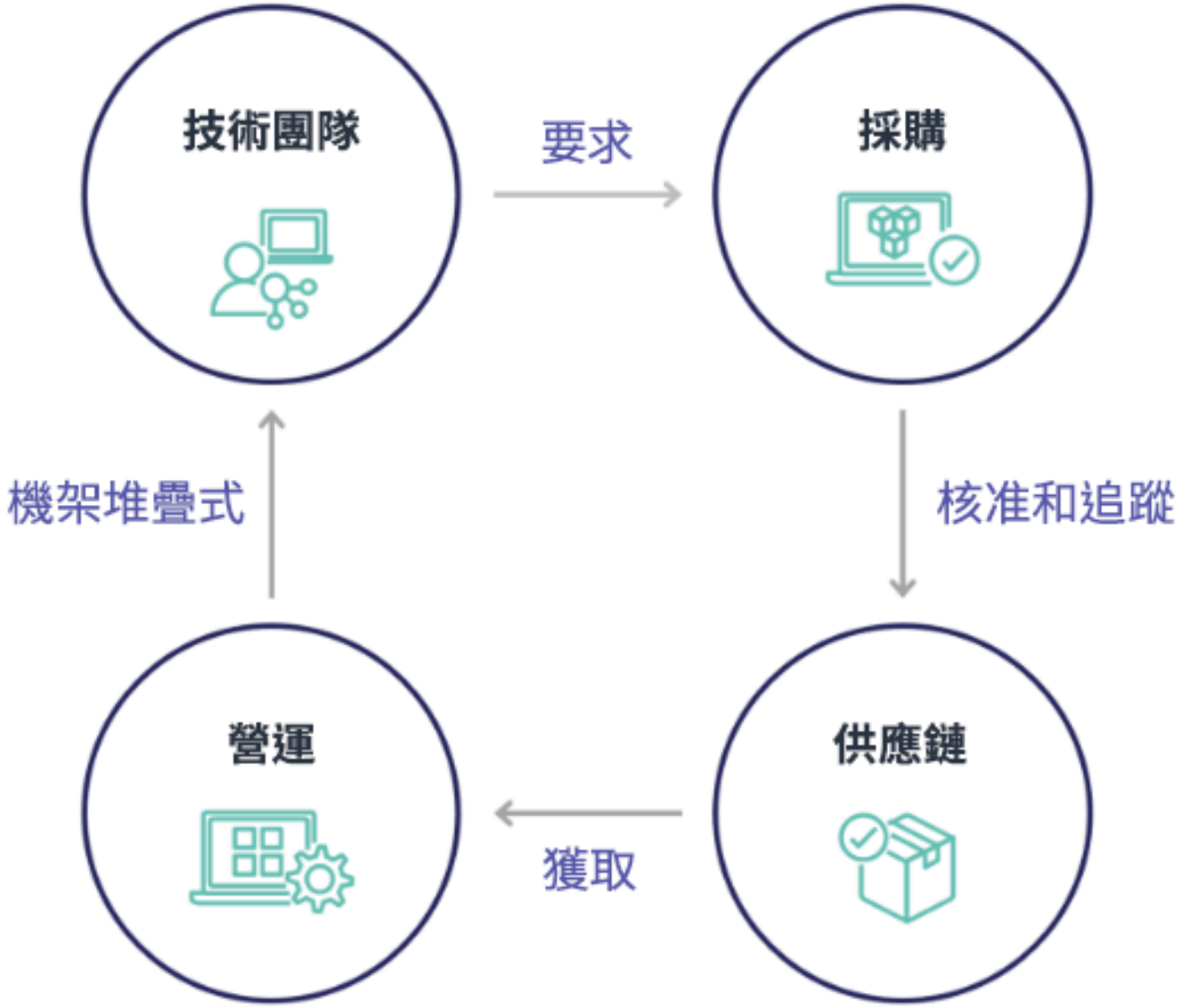
未建立此最佳實務時的曝險等級： 高

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風險暴露等級：高

實作指引

在傳統的內部部署 IT 設定中，客戶通常會面臨固定成本規劃方面的挑戰，而這些成本通常只會在隨著購買新的 IT 硬體和服務以滿足尖峰期需求時偶爾變動一下。相反的，AWS 雲端 會採用不同的方法，就是讓客戶僅依照實際層面的 IT 和業務需求而取用的資源數來支付費用。在雲端環境中，需求可能會按每月、每日甚至每小時波動。

使用雲端的好處包括效率高、速度快且靈活，隨之產生之成本和使用模式的變化度也頗高。為因應更高的工作負載效率或部署新工作負載和功能，成本可能時而降低，時而增加。隨著工作負載擴展以服務不斷擴大的客戶群，由於資源的可存取率變高，雲端用量和成本也相應提升。雲端服務的這種靈活性延伸到成本面和預測面，進而創造出一定程度的彈性。

亦步亦趨地緊隨這些不斷變化的業務需要和需求驅動因素，並以達到最精準規劃為最高目標，這些都是必不可少的要件。傳統組織預算處理程序需要調整才能適應這種變化。

在預測新工作負載的成本時，不妨考慮執行成本建模。成本建模可建立預期雲端成本的基本理解，協助您執行總體擁有成本 (TCO)、投資報酬率 (ROI) 和其他財務分析、與利害關係人進行目標和期望設定，以及找出成本優化的機會。

您的組織應該了解成本定義和已接受的分組。您預測的詳細等級可能會因組織的結構和內部工作流程而有所不同。選取適合您特定需求和組織設定的精細度。了解在哪個層級執行預測至關重要，詳情如下：

- 管理帳戶或 AWS Organizations 層級：管理帳戶是您用於建立 AWS Organizations 的帳戶。依預設，Organizations 會有一個管理帳戶。

- 連結帳戶或成員帳戶：Organizations 中的帳戶是一個標準 AWS 帳戶，其中包含您的 AWS 資源和可以存取那些資源的身分。
- 環境：環境是執行應用程式版本的 AWS 資源集。可以使用多個連結或成員帳戶建立環境。
- 專案：專案是要在固定期間內完成的一組目標或任務。考量在預測期間的專案生命週期是很重要的一件事。
- AWS 服務：群組或類別 (例如運算或儲存服務)，您可以在其中將用於預測的 AWS 服務分組。
- 自訂分組：您可以根據組織的需求建立自訂群組，例如業務部門、成本中心、團隊、成本分配標籤、成本類別、連結帳戶或這些項目的組合。

找出會影響使用成本的業務驅動因素，並分別預測每個因素，以預先計算好預期用量。部分驅動因素可能與組織內的 IT 和產品團隊有所關聯。而您的銷售、行銷和業務主管已經熟知行銷活動、促銷、擴展地理面積和併購等其他業務驅動因素，團隊必須合作和重視所有需求驅動因素。

根據您過去的支出而定義的未來時間範圍內，您可以針對以趨勢為基礎預測使用 AWS Cost Explorer。AWS Cost Explorer 預測引擎會根據收費類型 (例如：預留執行個體) 細分您的歷史資料，並結合使用機器學習和基於規則的模型，分別預測所有收費類型的支出。

一旦建立了預測流程並建置好模型，您就可以使用 AWS Budgets，透過指定時段、重複週期或金額 (固定或可變)，以及篩選條件 (例如服務、AWS 區域 和標籤) 來精細地設定自訂預算。預算通常以一年為期，且保持固定不變，所有參與者必須嚴格遵守預算計畫。相較之下，預測可以更加靈活，而且可以全年隨時調整，並提供一年、兩年或三年的動態預測。當您在技術和商業利益關係人之間建立財務期望時，預算編列和預測都相當重要。準確的預測和實作，不僅讓直接負責佈建成本的利益關係人更能掌握狀況，還能夠提高整體成本感知。

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您的利害關係人。您還可以根據實際成本 (為反應性) 或預測成本，建立 AWS Budgets 警示，從而提供了採取措施緩解潛在成本超支的時間。當您的成本或用量實際超出一定額度，或預測超出預算金額時，系統會提醒您。

使用趨勢型演算法 (輸入歷史成本) 和驅動因素型演算法 (例如：推出新產品、區域性擴展或新工作負載環境)，調整現有預算編列和預測流程，使其變得更加動態靈活，因為這些演算法都非常適合動態和可變的支出環境。一旦使用 Cost Explorer 或任何其他工具確立趨勢型預測後，請使用 AWS Pricing Calculator，根據預期用量 (流量、每秒請求數或必要的 Amazon EC2 執行個體) 估計您的 AWS 使用案例和未來成本。

追蹤預測準確度，因為您可以根據這些預測計算結果來編列預算。監控整合式雲端成本預測的準確性和有效性。定期檢閱與預測相比的實際支出，並根據需要進行調整以提高預測準確性。追蹤預測差異，並針對報告的差異執行根本原因分析，然後根據分析結果來採取行動和調整預測。

如 COST01-BP02 在財務與技術之間建立合作夥伴關係 中所述，務必在 IT、財務和其他利害關係人之間經營好合作關係和規律，才能確認所有人以一致的方式使用相同的工具或程序。如果預算可能需要改變，請提高接觸頻率以協助提升對這些變化的因應速度。

實作步驟

- 定義組織內的成本語言：在組織內建立具有多個維度和群組的通用 AWS 成本語言。確保利害關係人了解預測精細度、定價模型以及成本預測層級。
- 分析趨勢型預測：使用趨勢型預測工具，例如 AWS Cost Explorer 和 Amazon Forecast。從服務、帳戶、標籤和成本類別等多種層面分析您的使用成本。如果需要進階預測，請將您的 AWS 成本和用量 (CUR) 資料匯入 Amazon Forecast，其將線性迴歸當作一種機器學習形式來用於預測。
- 分析以驅動因素為基礎的預測：識別出業務驅動因素對雲端使用情況的影響，並分別預測每個因素，預先計算預期使用成本。與業務單位主管和利害關係人密切合作，了解對新驅動因素的影響，並計算預期成本變動，以準確編列預算。
- 更新現有預測與預算處理程序：根據您所採用的預測方法 (例如趨勢型、業務驅動因素型或結合兩種預測方法)，定義您的預測和預算編列流程。預算應經過計算、切合實際，並以您的預測為基準。
- 設定提醒和通知：使用 AWS Budgets 警示和成本異常偵測，以接收警示和通知。
- 與關鍵的利害關係人一起定期審查：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 Cost and Usage Report
- 使用 Cost Explorer 進行預測
- Amazon QuickSight 預測
- Amazon Forecast
- AWS Budgets

相關影片：

- 如何使用 AWS Budgets 追蹤我的支出和用量
- AWS 成本優化系列：AWS Budgets

相關範例：

- 了解並建置以驅動因素為基礎的預測
- 如何建立和推動預測文化
- 如何改善雲端成本預測
- 使用正確的工具進行雲端成本預測

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

設定雲端預算及相關機制，偵測使用期間的異常情況。針對預先定義的目標設定成本和用量警示的相關工具，並於用量超過目標時接收通知。舉辦定期會議，分析工作負載的成本效益並提升成本感知。

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最佳化。您可以舉辦專門的會議討論成本效益，或在工作負載的定期營運報告週期中包含優化成本的內容。使用服務和工具定期監控您的成本效益，並實施能夠節省成本的措施。

透過多種篩選條件和精細度來檢視成本和用量時，可使用 AWS Cost Explorer，這項功能會提供儀表板和報告，例如依服務或帳戶分類的成本、每日成本或市場成本。根據設定的預算追蹤成本使用和用量狀況時可使用 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。歡迎查看我們的 成本智慧儀表板 (CID) 解決方案 (建於 Amazon QuickSight 上)，可以更清楚檢視資料。

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。

根據您的成本細項和用量，透過視覺化圖表查看您的 Savings Plans 建議。每小時呈現的圖表顯示隨需支出以及建議的 Savings Plans 承諾，提供估計成本節省、Savings Plans 涵蓋範圍和 Savings Plans 使用率的深入分析。這些資訊能協助組織了解 Savings Plans 如何在無需投入時間資源建立模型來分析支出的條件下，應用於每小時的支出。

定期建立相關報告，納入 Savings Plans 重點、預留執行個體和 Amazon EC2 適當調整大小的建議 (來自 AWS Cost Explorer)，開始降低與穩定工作負載、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。遵循 AWS 最佳實務來減少浪費，或請您的客戶團隊和合作夥伴協助您 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。報告參與者應包含應用程式團隊和擁有者、財務和雲端成本相關重要決策者的代表。

資源

相關文件：

- AWS Cost Explorer
- AWS Trusted Advisor
- AWS Budgets
- AWS Cost and Usage Report
- AWS Budgets 最佳實務
- Amazon S3 分析

相關範例：

- Well-Architected 實驗室：成本與管控用量
- 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： 中

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： 中

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 以得知新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 以得知新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降價，已成為 AWS 在客戶從我們的營運規模所獲利益當中提高經濟效益的標準機制。截至 2024 年，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

COST01-BP08 建立成本感知文化

在您的組織中實作變更或計畫，以建立成本感知文化。建議從較小的計劃開始，然後隨著能力的增強和使用雲端的增加，再實作更大和更廣泛的計劃。

未建立此最佳實務時的曝險等級： 低

實作指引

成本感知文化可讓您透過在組織中以系統和分散的方式執行最佳實務，擴展成本優化和雲端財務管理 (財務營運、智慧雲端中心、雲端維運團隊等等)。相較於嚴格的由上而下、集中式方法，成本感知可讓您輕鬆地在整個組織建立高水準的能力。

建立雲端運算的成本感知 (尤其是對於雲端運算的主要成本動因)，可讓團隊了解成本方面的任何變更預期會產生的結果。存取雲端環境的團隊應了解定價模型，以及傳統內部部署資料中心與雲端運算之間的差異。

成本感知文化的主要優點是，技術團隊可主動且持續地優化成本 (例如，在建構新的工作負載，或對現有的工作負載進行變更時，會將其視為非功能性需求)，而不是等到必要時才被動執行成本優化。

文化中的小幅變化可以對目前和未來工作負載的效率產生很大的影響。這些範例包括：

- 在工程團隊中提供可見性和建立感知以了解其工作性質，及其對成本方面有何影響。
- 在您的組織中對成本和用量進行遊戲化。這可以透過公開可見的儀表板，或比較各團隊的標準化成本和用量報告來實現 (例如，每一工作負載的成本和每一交易的成本)。
- 認識成本效益。公開或私下獎勵自願或未經要求完成的成本優化成就，並從錯誤中學習，以避免重蹈覆轍。
- 建立由上而下的組織要求，讓工作負載依預先定義的預算執行。
- 探究企業的變更需求，以及要求的變更對於基礎架構或工作負載組態的成本影響，以確保您只須就需要的部分付費。
- 確定變更的規劃師了解預期的變更有何成本影響，且已經過利害關係人的確認，應以符合成本效益的方式提供商業成果。

實作步驟

- 向技術團隊報告雲端成本： 增強成本感知，並且為財務與業務利害關係人建立效率 KPI。
- 通知利害關係人或團隊成員有已規劃的變更： 在每週變更會議期間建立議程項目來討論已規劃的變更，以及對於工作負載的成本效益影響。
- 與您的客戶團隊會面： 安排與客戶團隊的定期會面，與他們討論產業趨勢和 AWS 服務。與您的客戶經理、架構師和支援團隊進行討論。
- 分享成功案例： 分享關於任何工作負載、AWS 帳戶 或組織降低成本的成功案例，以建立成本優化方面的正面態度與鼓勵。
- 培訓： 確定技術團隊或其成員已受過 AWS 雲端 資源成本感知的相關培訓。
- AWS 活動和會議： 參加當地的 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。
- 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，以關注 AWS 所分享的新版本、實作、範例和變更。

資源

相關文件：

- AWS 部落格
- AWS 成本管理
- AWS 新聞部落格

相關範例：

- AWS 雲端財務管理
- AWS Well-Architected 實驗室：雲端財務管理

COST01-BP09 量化成本最佳化所帶來的商業價值

量化成本優化所帶來的商業價值，可讓您了解給組織提供的全部效益。由於成本優化是一項必要的投資，因此量化商業價值可讓您向利害關係人解釋投資報酬率。量化商業價值有助於您在未來就成本優化投資獲得利害關係人更多的支持，並提供一個框架來衡量組織成本優化活動的成果。

未建立此最佳實務時的風險暴露等級：中

實作指引

量化商業價值意味著衡量企業從採取的行動和決策中所獲得的好處。商業價值可以是有形的 (例如費用降低或利潤增加)，也可以是無形的 (例如品牌信譽提升或客戶滿意度變高)。

量化成本最佳化所帶來的商業價值意味著判斷您在更有效率地支出成本上所做的努力，可以讓您獲得多少價值或收益。例如，如果公司支出 100,000 美元在 AWS 上部署工作負載，然後將其最佳化，而新成本變成只有 80,000 美元，且並未犧牲品質或輸出。在這種情況下，成本最佳化所帶來的量化商業價值會是節省了 20,000 美元。不過，除了節省成本外，公司還可以從更快的交貨時間、提高的客戶滿意度或成本最佳化努力所產生的其他指標等方面來量化價值。利害關係人需要就成本最佳化的潛在價值、工作負載的最佳化成本和回報價值做出決策。

除了報告成本優化所帶來的節省之外，建議您量化提供的額外價值。成本優化效益通常根據每個業務成果所較低的成本進行量化。例如，您可以在購買 Savings Plans (可降低成本和維持工作負載輸出水準) 時量化 Amazon Elastic Compute Cloud (Amazon EC2) 所節省的成本。您可以在閒置的 Amazon EC2 執行個體遭到移除或未連接的 Amazon Elastic Block Store (Amazon EBS) 磁碟區遭到刪除時，量化所降低的 AWS 支出成本。

不過，成本優化的消費絕非僅限於成本降低或避免。考慮擷取額外資料，以測量效率改善和商業價值。

實作步驟

- 評估商業效益：這是分析和調整 AWS 雲端 成本的程序，這個程序所採用的方法會將每一美元支出所能獲得的效益最大化。請不要不顧商業價值，一昧地降低成本，而是要考慮成本最佳化所帶來的商業效益和投資回報，這樣才有可能從支出的成本中獲得更多價值。重點在於聰明地支出，以及在能產生最佳回報的領域進行投資和支出。
- 分析預測的 AWS 成本：預測可讓財務利害關係人與其他內部和外部組織利害關係人設定期望，並可改善組織的財務可預測性。AWS Cost Explorer 可以用來預測您的成本和用量。

資源

相關文件：

- AWS 雲端 成本
- AWS 部落格
- AWS 成本管理
- AWS 新聞部落格
- Well-Architected 可靠性支柱白皮書
- AWS Cost Explorer

相關影片：

- 在 AWS 上充分發揮 Windows 的商業價值

相關範例：

- 測量並最大化 Customer 360 的業務價值
- 採用 Amazon Web Services 受管資料庫所帶來的商業價值
- 獨立軟體開發廠商的 Amazon Web Services 商業價值
- 雲端現代化的商業價值
- 遷移到 Amazon Web Services 的商業價值

支出和用量感知

問題

- COST 2.如何管控用量？
- COST 3.如何監控您的成本和用量？
- COST 4.如何進行資源除役？

COST 2.如何管控用量？

制訂政策和機制以確認產生合理的成本，同時達成目標。您可以運用相互制衡的方法，在不超支的情況下進行創新。

最佳實務

- COST02-BP01 根據貴組織的需求制定政策
- 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 根據貴組織的需求制定政策

制定定義組織如何管理資源的政策，並定期加以檢查。政策應涵蓋資源和工作負載的成本面向，包括資源生命週期中的建立、修改和除役。

未建立此最佳實務時的曝險等級： 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，以及識別降低成本的機會至關重要。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。將資源成本歸因至工作負載、個別組織或產品擁有者的能力，能夠帶動高效使用的行為模式，並且有助於減少浪費。精確的成本和用量監控可協助您了解工作負載的優化程度，以及組織單位和產品的獲利程度。這項知識可讓您更明智地決定應將資源分配到組織內的何處。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面向，包括在資源生命週期中資源的建立、修改和除役。確認已遵循政策和程序，並已實作雲端環境中的任何變更。在 IT 變更管理會議中提出問題，以釐清計畫性變更對成本的影響 (無論是增加還是減少)、商務理由和預期成果。

政策應該簡單易懂，以便有效地在整個組織中實作。政策還需要易於遵循和解釋 (以方便使用) 並且明確 (團隊間不會產生誤解)。此外，必須定期加以檢查 (如我們的機制)，並隨著客戶業務狀況或優先權的變化 (政策會因而過時) 進行更新。

從廣泛的高階政策開始，例如應使用哪個地理區域，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，測試和開發環境中較低效能的儲存體)、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)，以及這些資源的使用期間長短 (暫時、短期還是一段特定期間)。

政策範例

以下是範例政策，可供您檢閱以建立自己的雲端管控政策，其重點為成本優化。確實根據組織的要求和利害關係人的請求來調整政策。

- 政策名稱： 定義明確的政策名稱，例如「資源優化」和「成本降低」政策。
- 目的： 解釋為何應使用此政策，以及預期的結果為何。此政策的目標是要確認部署和執行所需的工作負載以符合業務需求時的最低成本。
- 範圍： 明確定義誰應使用此政策，以及何時應使用此政策，例如 DevOps X Team 在 X 環境 (生產或非生產) 將此政策用於美國東部客戶。

政策聲明

1. 根據工作負載的環境和業務要求 (開發、使用者接受度測試、生產前或生產)，選取美國東部 1 或多個美國東部區域。
2. 將 Amazon EC2 和 Amazon RDS 執行個體排程在早上六點到晚上八點之間執行 (東部標準時間 (EST))。
3. 在八小時後停止所有未使用的 Amazon EC2 執行個體，並在閒置 24 小時後停止未使用的 Amazon RDS 執行個體
4. 在非生產環境中閒置 24 小時後，終止所有未使用的 Amazon EC2 執行個體。提醒 Amazon EC2 執行個體擁有者 (根據標籤) 檢閱其生產環境中已停止的 Amazon EC2 執行個體，並通知他們如果 Amazon EC2 執行個體未使用，將在 72 小時內終止。
5. 使用一般執行個體系列和大小 (例如 m5.large)，然後根據 CPU 和記憶體使用率，使用 AWS Compute Optimizer 調整執行個體大小。
6. 使用自動擴展根據流量動態調整執行中的執行個體數量，以訂定優先順序。
7. 對非關鍵工作負載使用 Spot 執行個體。
8. 檢閱容量要求，以認可可預測工作負載的 Savings Plans 或預留執行個體，並通知雲端財務管理團隊。
9. 使用 Amazon S3 生命週期政策將不常存取的資料移至成本較低的儲存層。若未定義保留政策，請使用 Amazon S3 Intelligent Tiering 將物件自動移至封存層。
10. 使用 Amazon CloudWatch 監控資源使用率並設定警示以觸發擴展事件。
11. 針對每個 AWS 帳戶，使用 AWS Budgets 根據成本中心和業務單位設定帳戶的成本及用量預算。
12. 使用 AWS Budgets 設定帳戶的成本和用量預算，可協助您掌握支出並避免出現非預期的帳單，進而讓您更有效地控制成本。

程序： 提供實作此政策的詳細程序，或參閱說明如何實作每項政策聲明的其他文件。本節應提供執行政策要求的逐步指示。

若要實作此政策，您可以使用各種第三方工具或 AWS Config 規則來檢查是否符合政策聲明，並使用 AWS Lambda 函數觸發自動修復動作。您也可以使用 AWS Organizations 來強制執行政策。此外，您應定期檢閱資源用量，並視需要調整政策，以確認政策持續符合您的商業需求。

實作步驟

- 與利害關係人會面： 若要制定政策，請要求組織內的利害關係人 (雲端業務辦公室、工程師或執行政策的功能決策者) 指定其要求，並將其記錄下來。採取反覆的方法，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。
- 獲取確認： 確定團隊成員均同意誰可對 AWS 雲端 進行存取及部署的政策。請確定成員遵循組織的政策，並確認其資源建立符合議定的政策和程序。
- 建立上線培訓課程： 要求新進的組織成員完成上線培訓課程，以建立對成本的掌握度和組織要求。他們可以根據自身過往的經驗採行不同的政策，也可以完全不列入考量。
- 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。這項資訊用來對應至 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： 確認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：

- 雲端中的變更管理

- 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
- AWS 管理與管控
- 使用 IAM 政策控制對 AWS 區域 的存取
- 全球基礎架構區域和可用區域

相關影片：

- 大規模的 AWS 管理與管控

相關範例：

- VMware - 什麼是雲端政策？

COST02-BP02 實作總目標和具體目標

為您的工作負載實作成本與用量的總目標和具體目標。總目標可為您的組織提供期望的結果方向，具體目標則可提供要為您的工作負載達成的特定可測量成果。

未建立此最佳實務時的風險暴露等級：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。當組織在 AWS 方面不斷成長時，設定和追蹤成本優化的總目標是非常重要的。這些總目標或 關鍵績效指標 (KPI) 可包含隨需支出的百分比，或特定優化服務 (例如 AWS Graviton 執行個體或 gp3 EBS 磁碟區類型等) 的採用。設定可衡量和可實現的目標有助於衡量效率的改善情況，這對於企業營運非常重要。總目標可為您的組織提供預期結果的指引和方向，

具體目標則是要達成的具體可衡量成果。簡言之，總目標是努力的方向，具體目標則表示該方向有多遠，以及多久可以達成總目標；其方針是具體 (Specific)、可測量 (Measurable)、可指派 (Assignable)、切合實際 (Realistic) 與及時 (Timely)，簡稱 SMART。舉例來說，平台用量大幅增加，而成本僅稍微增加 (非線性)，即為總目標。另外，平台用量增加 20%，成本增加少於 5%，則是具體目標之一。另一個常見的總目標，是工作負載的效率每隔六個月就必須有所成長。相關的具體目標是每個業務指標的成本每六個月需要減少百分之五。使用正確的測量結果，並為您的組織設定計算過的 KPI。您可以從基本的 KPI 開始，之後再根據業務需求逐步發展。

成本優化的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的每個業務成果成本。針對所有工作負載實施此總目標，並設定一個具體目標，例如每六個月至一年提高百分之五的效率。在雲端中，建立成本優化能力以及新的服務和功能版本即可實現此一目標。

具體目標是您希望達到以實現總目標的可量化基準，而基準則是將您的實際結果與具體目標做比較。針對運算服務 (例如 Spot 採用、Graviton 採用、最新執行個體類型和隨需範圍)、儲存服務 (例如 EBS GP3 採用、過時的 EBS 快照和 Amazon S3 標準儲存) 或資料庫服務用量 (例如 RDS 開放原始碼引擎、Graviton 採用和隨需範圍) 的每單位成本，建立 KPI 基準。這些基準和 KPI 可協助您確認您是否以最具成本效益的方式使用 AWS 服務。

下表提供標準 AWS 指標清單以供參考。每個組織可以擁有這些 KPI 的不同目標值。

Category	KPI (%)	Description
Compute	EC2 usage Coverage	EC2 instances (in cost or hours) using SP+RI+Spot compared to total (in cost or hours) of EC2 instances
Compute	Compute SP/RI utilization	Utilized SP or RI hours compared to total available SP or RI hours
Compute	EC2/Hour cost	EC2 cost divided by the number of EC2 instances running in that hour
Compute	vCPU cost	Cost per vCPU for all instances
Compute	Latest Instance Generation	Percentage of instances on Graviton (or other modern generation instance types)
Database	RDS coverage	RDS instances (in cost or hours) using RI compared to total (in cost or hours) of RDS instances

Category	KPI (%)	Description
Database	RDS utilization	Utilized RI hours compared to total available RI hours
Database	RDS uptime	RDS cost divided by the number of RDS instances running in that hour
Database	Latest Instance Generation	Percentage of instances on Graviton (or other modern instance types)
Storage	Storage utilization	Optimized storage cost (for example Glacier, deep archive, or Infrequent Access) divided by total storage cost
Tagging	Untagged resources	Cost Explorer: 1.過濾掉抵用金、折扣、稅款、退款、市場，並複製最新的每月成本 2.在 Cost Explorer 中選取僅顯示未標記的資源 3.將未標記資源中的金額除以每月成本。

使用此資料表，其中應納入具體目標值或基準值，並根據您組織的總目標計算這些值。您需要衡量企業的某些指標，並了解該工作負載的業務成果，藉以定義準確且切實的 KPI。當您評估組織內的績效指標時，請劃分服務不同用途的不同指標類型。這些指標主要衡量技術基礎設施的效能和效率，而不是直接衡量整體業務影響。例如，這些指標可能會追蹤伺服器回應時間、網路延遲或系統正常執行時間。這些指標對於評估基礎設施支援組織的技術運作方式至關重要。但是，這些指標不會對提供針對更廣泛的營運目標的直接洞見，例如客戶滿意度、收入增長或市場份額。若要全盤了解業務績效，請使用與業務成果直接相關的策略性業務指標補充這些效率指標。

近乎即時地檢視 KPI 和相關節約機會，並追蹤一段時間內的進度。若要開始定義和追蹤 KPI 目標，建議您使用雲端智慧儀表板 (CID) 的 KPI 儀表板。KPI 儀表板會根據來自成本和用量報告 (CUR) 的資料，提供一系列建議的成本優化 KPI，讓您能夠設定自訂總目標，以及追蹤一段時間內的進度。

如果您有其他解決方案可以設定和追蹤 KPI 總目標，請確定組織中的所有雲端財務管理利害關係人都採用這些方法。

實作步驟

- 定義預期的用量等級：請先將重點放在用量等級上。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產生變化，以及會因季節性增加或行銷活動而做何改變？
- 定義工作負載資源與成本：定義用量層級後，請量化達成那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。指定每個要點的成本，並在用量發生變化時預測成本的變化。
- 定義業務總目標：使用預期的用量和成本變更的輸出，與預期的技術變更或任何您正在執行的計畫結合，並制定工作負載的總目標。總目標必須涵蓋用量、成本和兩者之間的關係。總目標必須簡單具體，以協助大家了解企業預期的成果 (例如，確定將未使用的資源控制在特定成本水位以下)。您無需為每個未使用的資源類型定義總目標，或定義造成總目標和具體目標中損失的成本。如果預期有成本變更但用量不變，請確認已有組織計畫 (例如培訓和教育等能力打造計畫)。
- 定義具體目標：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，具體目標應該量化改善的程度 (通常是所有經費所獲得的業務輸出)，及其應達成時間。例如，您可以設定將過度佈建所造成的浪費降至最低的總目標。訂下這個總目標後，您的具體目標可以是第一層生產工作負載中因運算過度佈建而造成的浪費，不應超出運算成本的百分之十。此外，第二個具體目標可以是第二層生產工作負載中因運算過度佈建而造成的浪費，不應超出運算成本的百分之五。

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
- S.M.A.R.T.總目標
- 如何使用 CID KPI 儀表板追蹤成本優化 KPI

相關影片：

- Well-Architected 實驗室：總目標和具體目標 (Level 100)

相關範例：

- 什麼是單位指標？
- 選取支援業務的單位指標
- 實務中的單位指標 — 經驗傳承
- 單位指標如何協助在業務職能之間建立一致性
- Well-Architected 實驗室：除役資源 (總目標和具體目標)
- Well-Architected 實驗室：資源類型、大小和數目 (總目標和具體目標)

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。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS Organizations 可讓您建立多個 AWS 帳戶，當您在 AWS 上擴展工作負載時，這可協助您集中管控環境。您可以採用組織單位 (OU) 結構來為 AWS 帳戶 進行分組，再於每個組織單位下建立多個 AWS 帳戶，藉此塑造組織階層的模型。若要建立帳戶結構，您必須先決定要以哪個 AWS 帳戶 作為管理帳戶。之後，便能建立新的 AWS 帳戶，或根據您指定的帳戶結構將現有帳戶選為成員帳戶，相關做法請遵循管理帳戶最佳實務和成員帳戶最佳實務。

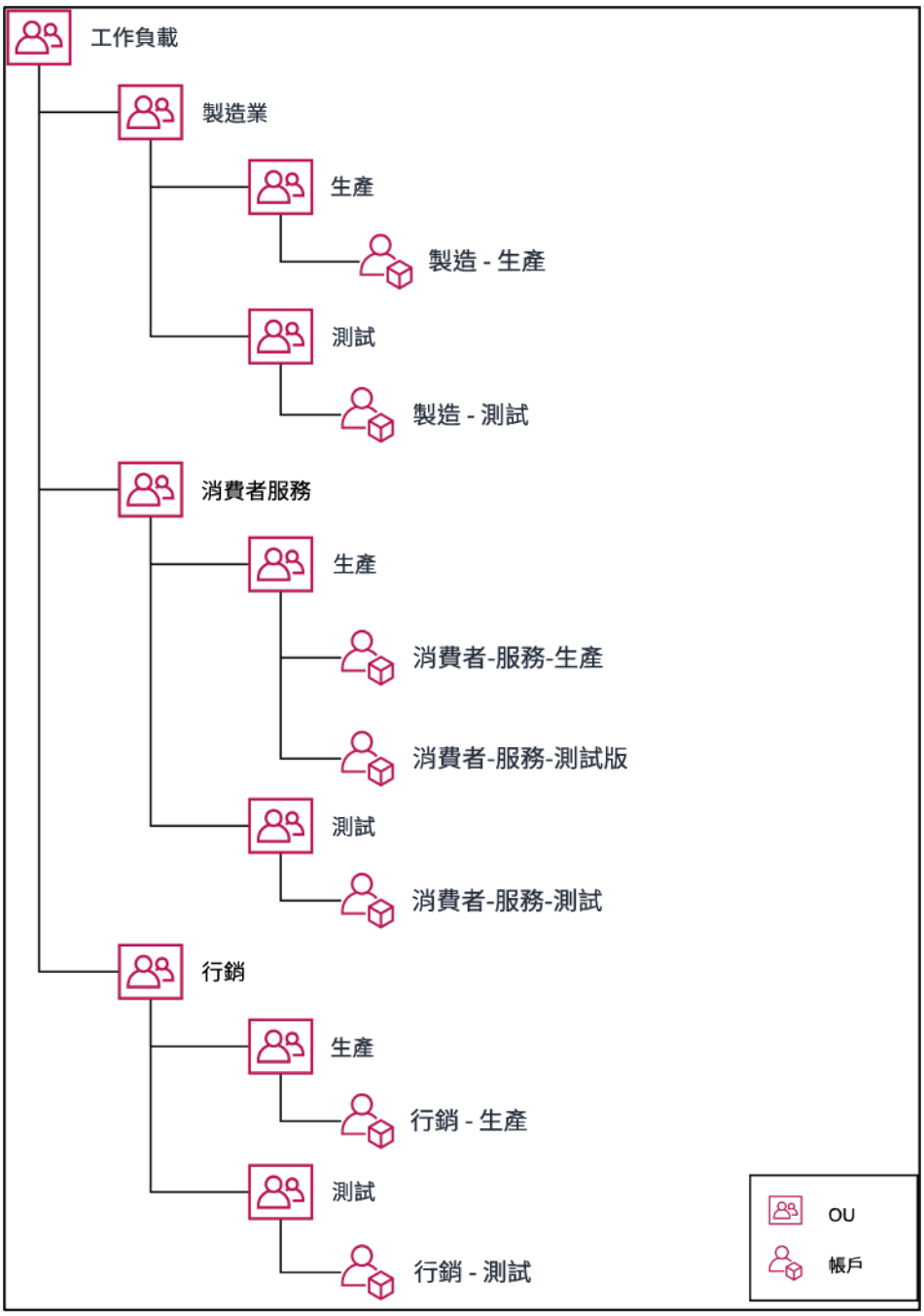
無論您的組織規模或用量為何，都建議您一律要有至少一個管理帳戶，以及一個與管理帳戶連結的成員帳戶。所有工作負載資源都只應位於成員帳戶內，請勿在管理帳戶內建立任何資源。關於應該擁有多少個 AWS 帳戶，並沒有一體適用的答案。請評估您目前和未來的運作與成本模式，確保 AWS 帳戶 的結構呼應組織的目標。有些公司基於業務原因建立多個 AWS 帳戶，例如：

- 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 內，合併帳單會在一或多個成員帳戶與管理帳戶之間建立結構。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。

下圖顯示如何使用 AWS Organizations 與組織單位 (OU) 來將多個帳戶分組，並將多個 AWS 帳戶 放到每個 OU 底下。建議您使用 OU 來處理各種使用案例和工作負載，以便提供用於整理帳戶的模式。



將多個 AWS 帳戶 分組到組織單位底下的範例。

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。為了安全，我們必須遵守存取和資料要求。為求可靠，我們必須有所限制，以免環境和工作負載影響其他資源。請定期檢閱 Well-

Architected 架構的安全性和可靠性支柱，並遵循其中所提供的最佳實務。財務結構會建立嚴格的財務分隔 (不同的成本中心、工作負載擁有權和責任)。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組織內的個別業務單位或部門，或是擁有帳戶的利害關係人。

- 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，並維持分隔要求。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，而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：

- 使用成本分配標籤
- 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
- 管理帳戶和成員帳戶的最佳實務
- 使用多個帳戶整理您的 AWS 環境
- 開啟共用的預留執行個體和 Savings Plans 折扣
- 合併帳單
- 合併帳單

相關範例：

- 分割 CUR 和共用存取

相關影片：

- 向您介紹 AWS Organizations
- 設定會使用 AWS Organizations 最佳實務的多帳戶 AWS 環境

相關範例：

- Well-Architected 實驗室：建立 AWS 組織 (Level 100)
- 分割 AWS Cost and Usage Report 和共用存取
- 為電信公司定義 AWS 多帳戶策略
- 將 AWS 帳戶 最佳化的最佳實務
- AWS Organizations 的組織單位最佳實務

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。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用者角色和群組是設計和實作安全高效系統的基礎建置組塊。角色和群組可協助組織在控制需求與靈活性和生產力的要求兩方面取得平衡，從而最終能支援組織目標和使用者需求。如 AWS Well Architected Framework 安全支柱的身分和存取管理一節所建議，您需要有強大的身分管理和許可，以在合適條件下為合適的人員提供合適資源的存取權。使用者只會獲得要完成其任務所需的存取權。這可將未經授權存取或濫用的相關風險降至最低。

在制定政策後，您可以在組織內建立邏輯群組和使用者角色。這可讓您指派許可、控制使用情況，並協助實作強大的存取控制機制，防止有人未經授權存取敏感資訊。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、財務控制者或商業分析師) 相符。這些群組會將執行類似任務且需要類似存取權限的人員進行分類。角色定義群組必須執行的工作。管理群組和角色的許可會比管理個別使用者的許可容易。角色和群組能以一致且有系統的方式為所有使用者指派許可，以避免錯誤和不一致。

當使用者的角色變更時，管理員可以調整角色或群組層級的存取權，而不是重新設定個別使用者帳戶。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。如需關於使用者、群組和驗證的最佳實務，請參閱 AWS Well-Architected Framework 的安全支柱。
- 實作角色和政策：使用組織政策中定義的動作，建立所需的角色和存取政策。如需關於角色和政策的最佳實務，請參閱 AWS Well-Architected Framework 的安全支柱。

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
- AWS Well-Architected Framework 安全支柱
- AWS Identity and Access Management (IAM)
- AWS Identity and Access Management 政策

相關影片：

- 為什麼要使用身分和存取管理

相關範例：

- Well-Architected 實驗室基本身分和存取
- 使用 IAM 政策控制對 AWS 區域 的存取
- 開始您的雲端財務管理之旅：雲端成本營運

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，控制對區域或資源類型的存取。

未建立此最佳實務時的風險暴露等級：中

實作指引

實作成本控制常見的第一步設定在發生偏離政策的成本或用量事件時發出通知。您可以快速採取動作，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。AWS Budgets 可讓您針對 AWS 的成本、用量和承諾折扣 (Savings Plans 和預留執行個體) 來設定通知和定義每月預算。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。

在透過 AWS Budgets 設定預算限制後，請使用 AWS Cost Anomaly Detection 來降低非預期的成本。AWS Cost Anomaly Detection 是成本管理服務，可使用機器學習來持續監控成本和用量，以偵測不尋常的支出。其可協助您識別異常支出與根本原因，以便您迅速因應。請先在 AWS Cost Anomaly Detection 中建立成本監視器，然後設定美元閾值以選擇提醒偏好 (例如，針對影響金額大於 1,000 美元的異常設定提醒)。收到提醒後，便能分析異常背後的根本原因，以及其對成本的影響。您也可以在 AWS Cost Explorer 中監控和執行您自己的異常分析。

透過 AWS Identity and Access Management 和 AWS Organizations 服務控制政策 (SCP) 在 AWS 中施行管控。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或管理 AWS 資源、可建立的資源類型以及建立資源的位置。這可以最大程度地降低在所定義的政策外建立資源的可能性。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 可集中控制組織中所有帳戶的最大可用許可，讓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 安全性支柱白皮書。

亦可透過管理 AWS 服務配額來實作管控。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。服務配額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 Budgets 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。請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。您也可以預先設定 AWS 預算操作，以實施特定的 IAM 或 SCP 政策，或停止目標 Amazon EC2 或 Amazon RDS 執行個體。預算操作可以自動執行，也可以要求工作流程核准。
- 實作異常支出通知：使用 AWS Cost Anomaly Detection 可降低組織中的意外成本，並分析潛在異常支出的根本原因。在建立成本監視器以識別指定精細度的不尋常支出，並在 AWS Cost Anomaly

Detection 中設定通知後，其便會在偵測到不尋常支出時向您發出提醒。這可讓您分析異常背後的根本原因，並了解其對成本的影響。在設定 AWS Cost Anomaly Detection 時使用 AWS Cost Categories，可識別哪個專案團隊或業務單位團隊能夠分析非預期成本的根本原因，並及時採取必要動作。

- 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
- AWS Budgets
- AWS Cost Anomaly Detection
- 控制您的 AWS 成本

相關影片：

- 如何使用 AWS Budgets 追蹤我的支出和用量

相關範例：

- 範例 IAM 存取管理政策
- 範例服務控制政策
- AWS 預算操作
- 建立 IAM 政策以使用標籤控制 Amazon EC2 資源的存取
- 限制只有特定 Amazon EC2 資源能夠存取 IAM 身分
- 建立 IAM 政策以便依系列限制 Amazon EC2 用量
- Well-Architected 實驗室：成本與用量管控 (Level 100)
- Well-Architected 實驗室：成本與用量管控 (Level 200)

- 使用 AWS Chatbot 來針對 Cost Anomaly Detection 進行 Slack 整合

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。

未建立此最佳實務時的風險暴露等級：低

實作指引

透過有效追蹤專案生命週期，組織可利用增強規劃、管理和資源優化來達成更出色的成本控制。透過追蹤所獲得的見解十分寶貴，可讓您做出有助於專案成本效益和整體成功率的明智決策。

追蹤工作負載的整個生命週期可協助您了解何時不再需要工作負載或工作負載元件。現有的工作負載和元件可能看似有在使用，但當 AWS 發行新的服務或功能時，其可能會除役，也可能會獲得採用。請檢查之前的工作負載階段。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。

您可以使用時間範圍或提醒功能來標記資源，以固定審查工作負載的時間。例如，如果開發環境上次審查時間是幾個月前，那麼現在可能是重新審查的好時機，以了解是否可以採用新服務，或環境是否在使用中。您可以在 AWS 上利用 myApplications 將應用程式分組並做標記，以管理和追蹤中繼資料，例如重要性、環境、上次審核和成本中心。您可以追蹤工作負載的生命週期，也可以監控和管理應用程式的成本、健康狀態、安全狀態和效能。

AWS 提供多種管理和管控服務，可以用於實體生命週期追蹤。您可以使用 AWS Config 或 AWS Systems Manager 來提供 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。

與應用程式生命週期管理 (ALM) 類似，追蹤專案生命週期應該會涉及多個流程、工具和團隊共同合作，例如設計與開發、測試、生產、支援和工作負載備援。

透過仔細監控專案生命週期的每個階段，組織可以獲得重要的見解和增強的控制力，從而能成功地規劃、實作和完成專案。這種仔細的監督會驗證專案不僅符合品質標準，而且會準時地在預算內交付，從而提高整體成本效率。

如需實作實體生命週期追蹤的詳細資訊，請參閱 AWSWell-Architected 卓越營運支柱白皮書。

實作步驟

- 建立專案生命週期監控流程：雲端卓越中心團隊必須建立專案生命週期監控流程。建立結構化與系統化的方法來監控工作負載，以改善專案的控制、可見性和效能。讓監控流程透明、協作並專注於持續改進，以最大程度地提高其有效性和價值。
- 執行工作負載審查：根據組織政策的定義，設定定期規律以稽核現有專案並執行工作負載審查。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：

- 在 AWS 上做標記的指引
- 什麼是 ALM (應用程式生命週期管理)？
- 適用於各工作職能的 AWS 受管政策

相關範例：

- 使用 IAM 政策控制對 AWS 區域 的存取

相關工具

- AWS Config
- AWS Systems Manager
- AWS Budgets
- AWS Organizations
- AWS CloudFormation

COST 3.如何監控您的成本和用量？

制訂政策和程序以監控並適當分配成本。這樣能夠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

設定成本管理和報告工具，以增強成本與使用資料的分析和透明度。設定您的工作負載以建立日誌項目，並利用該項目追蹤和分隔成本和用量。

未建立此最佳實務時的風險暴露等級：高

實作指引

像是精細程度為每小時的成本管理工具，可以提供詳細的帳單資訊，讓組織可以追蹤耗用量，協助找出一些成本增加的原因。這些資料來源提供整個組織最準確的成本和用量的檢視。

您可以使用 AWS 資料匯出 建立 AWS Cost and Usage Report (CUR) 2.0 的匯出檔。這是從 AWS 中接收詳細成本和用量資料的新推薦方法。為所有收費的 AWS 服務 (與 CUR 相同的資訊) 以及一些改善項目，提供每日或每小時用量精細度、費率、成本和使用屬性。CUR 中包含所有可能的維度，例如標記、位置、資源屬性和帳戶 ID。

根據您要建立的匯出類型，共有三種匯出類型：標準資料匯出、匯出至具有 Amazon QuickSight 整合的成本和用量儀表板，或舊版資料匯出。

- 標準資料匯出：定期將資料表匯出到 Amazon S3 的自訂匯出檔。
- 成本和用量儀表板：匯出並整合到 Amazon QuickSight 以部署預先建置的成本和用量儀表板。
- 舊版資料匯出：舊版資料 AWS Cost and Usage Report (CUR) 的匯出。

您可以使用下列自訂建立資料匯出檔：

- 包含資源 ID
- 分割成本分配資料
- 每小時的精細程度

- 版本控制
- 壓縮類型和檔案格式

對於在 Amazon ECS 或 Amazon EKS 上執行容器的工作負載，請啟用分割成本分配資料，如此才能根據容器工作負載消耗共用運算和記憶體資源的方式，將容器成本分配給個別業務單位和團隊。分割成本分配資料會將新容器層級資源的成本和用量資料導入 AWS Cost and Usage Report。分割成本分配資料是透過運算在叢集上執行的個別 ECS 服務和任務的成本所計算。

成本和用量儀表板會定期將成本和用量儀表板匯出到 S3 儲存貯體，並將預先建置的成本和用量儀表板部署到 Amazon QuickSight。如果您想在沒有自訂能力的情況下快速部署成本和用量資料的儀表板，請使用此選項。

如有需要，仍然可用舊式模式匯出 CUR，其中可以整合其他處理服務 (例如AWS Glue) 準備供分析的資料，並使用 SQL 查詢資料，透過 Amazon Athena 執行資料分析。

實作步驟

- 建立資料匯出檔：使用您想要的資料建立自訂的匯出檔，並控制匯出的結構描述。使用基本 SQL 建立帳單與成本管理資料匯出檔，並透過整合 Amazon QuickSight 以視覺化帳單和成本管理資料。您也可以用標準模式匯出資料，進而使用像 Amazon Athena 之類的其他處理工具來分析您的資料。
- 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：若要在過去 14 天存取具有每小時精細度的成本和用量資料，請考慮在帳單主控台啟用每小時資源層級資料。
- 設定應用程式記錄：確認您的應用程式會記錄交付的每個業務成果，以便追蹤和衡量相應成果。確保此資料的精細程度至少為每小時，以符合成本和用量資料。如需更多關於記錄和監控的詳細資訊，請參閱 Well-Architected 卓越營運支柱。

資源

相關文件：

- AWS 資料匯出
- AWS Glue
- Amazon QuickSight
- AWS 成本管理定價

- 標記 AWS 資源
- 使用 Cost Explorer 分析成本
- 管理 AWS Cost and Usage Report
- Well-Architected 卓越營運支柱

相關範例：

- AWS 帳戶設定
- 用於 AWS 帳單與成本管理的資料匯出
- AWS Cost Explorer 常用案例

COST03-BP02 將組織資訊新增至成本與用量

根據您的組織、工作負載屬性和成本分配類別來定義標記結構描述，以便您在成本管理工具中篩選及搜尋資源，或監控成本與用量。情況允許時，依據目的、團隊、環境，或其他與您的業務有關的條件，在所有資源間實作一致的標記。

未建立此最佳實務時的風險暴露等級：中

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。

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。

在組織的各帳戶建立和實作 AWS 標記標準，能讓您以一致統一的方式管理和管控 AWS 環境。使用 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 可讓您新增、刪除和管理多個資源的標籤。透過 Tag Editor，您可以搜尋您要標記的資源，然後在搜尋結果中管理資源的標籤。

AWS Cost Categories 可讓您為成本指派組織的意義，而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：聚集整個企業的所有利害關係人以定義結構描述。這通常包括屬於技術、財務和管理角色的人員。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 AWS Systems Manager 等工具來提高效率。
- 實作 AWS 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 AWS CloudFormation 之類的服務，確認資源在建立時會加上標籤。您也可以使用 Lambda 函數建立自動標記的自訂解決方案，或使用自訂微型服務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：

- 標記最佳實務
- AWS CloudFormation 資源標籤
- AWS Cost Categories
- 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

相關影片：

- 如何標記 AWS 資源以依據成本中心或專案分割我的帳單

- 標記 AWS 資源

相關範例：

- 根據身分或角色自動標記新的 AWS 資源

COST03-BP03 識別成本歸因類別

識別組織分類 (例如業務單位、部門或專案)，這些分類可以將組織內的成本分配給內部取用實體。使用這些分類來強制執行支出權責劃分、建立成本感知並推動有效的取用行為。

未建立此最佳實務時的曝險等級： 高

實作指引

成本分類的程序對預算、會計、財務報告、決策制定、基準和專案管理至關重要。透過對費用進行分類，團隊可更加了解他們在整個雲端之旅中將產生的成本類型，從而做出明智的決策並有效管理預算。

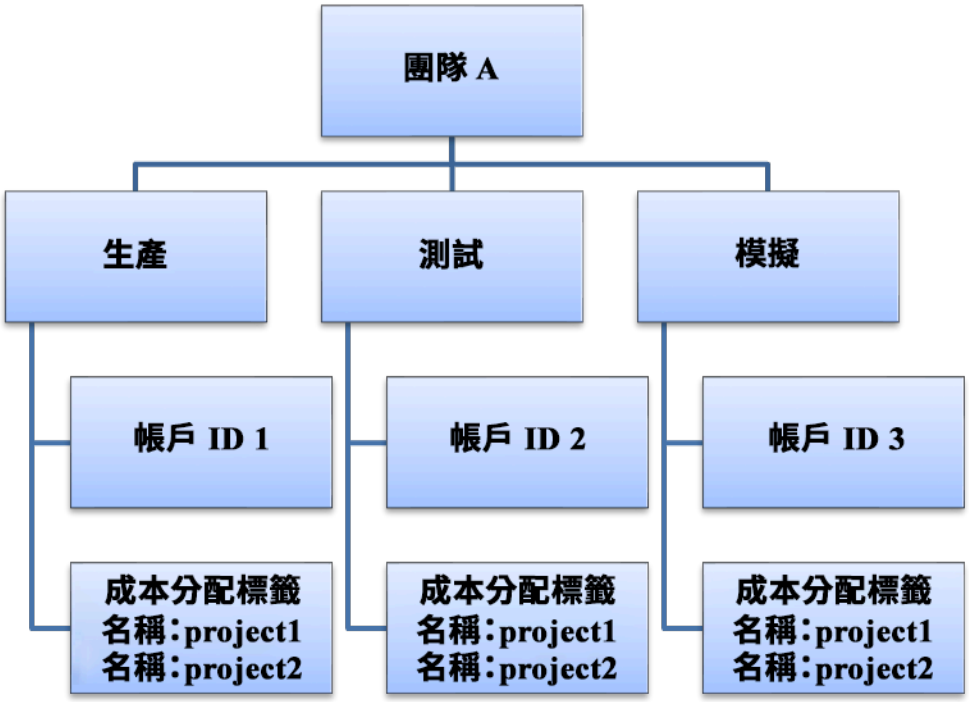
雲端支出權責劃分為有紀律的需求和成本管理建立了有力的誘因。對於將大部分雲端支出分配給取用業務單位或團隊的組織，這樣可以大幅節省雲端成本。此外，分配雲端支出有助於組織採用更多集中式雲端控管的最佳實務。

在定期會議中與財務團隊和其他相關利害關係人合作，了解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，而不是籠統的 IT 成本預算。

與您的組織中的利害關係人共同定義成本歸因類別後，請使用 AWS Cost Categories 將您的成本與用量資訊在 AWS 雲端 中分組成有意義的類別，例如特定專案的成本，或是部門或業務單位的 AWS 帳戶。您可以建立自訂類別，並使用各種不同的維度 (例如帳戶、標籤、服務或費用類型)，根據您定義的規則將成本與用量資訊對應至這些類別中。設定成本類別後，您就能依據這些類別檢視成本與用量資訊，進而讓組織能制定更好的策略與購買決策。這些類別也會顯示在 AWS Cost Explorer、AWS Budgets 和 AWS Cost and Usage Report 中。

例如，假設您為業務單位 (DevOps 團隊) 建立成本類別，並根據您所定義的群組，在每個類別下建立多個規則 (每個子類別的規則)，分別具有多個維度 (AWS 帳戶、成本分配標籤、服務或收費類型)。透過成本類別，您可以使用規則型引擎來組織成本。您設定的規則會將成本組織到類別中。在這些規則中，您可以對每個類別使用多個維度以進行篩選，例如特定 AWS 帳戶、AWS 服務，或費用類型。然後，您可以跨多個產品使用這些類別 (在 AWS Billing and Cost Management 和成本管理 主控台)。其中包括 AWS Cost Explorer、AWS Budgets、AWS Cost and Usage Report 和 AWS Cost Anomaly Detection。

例如，下圖顯示您可以有多個團隊 (成本類別)、多個環境 (規則)，且每個環境有多個資源或資產 (維度)，進而分組您組織中的成本與用量資訊。



成本與用量組織圖表

您也可以使用成本類別建立成本的群組。在您建立成本類別後 (您的用量記錄可在成本類別建立後的 24 小時內更新為新值)，這些類別會出現在 AWS Cost Explorer、 AWS Budgets、 AWS Cost and Usage Report和 AWS Cost Anomaly Detection。在 AWS Cost Explorer 和 AWS Budgets 中，成本類別會顯示為額外的帳單維度。您可以使用此維度來篩選特定成本類別值，或依成本類別分組。

實作步驟

- 定義您的組織類別： 與內部利害關係人和業務單位會談，定義可反映組織的結構和要求的類別。這些類別應該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。查看雲端服務為您的業務帶來的成果，例如培訓或教育，因為這些也是屬於組織類別。
- 定義您的功能類別： 與內部利害關係人和業務單位會談，定義可反映您在企業內具有之職能的類別。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。
- 定義 AWS Cost Categories： 建立成本類別以組織成本和用量資訊，過程中使用 AWS Cost Categories 並將 AWS 成本和用量對應至 適當的類別。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，以便 管理您的成本 (使用 AWS Cost Categories 在分類結構內管理)。

資源

相關文件：

- 標記 AWS 資源
- 使用成本分配標籤
- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS Cost and Usage Report
- AWS Cost Categories
- 使用 AWS Cost Categories 管理您的成本
- 建立成本類別
- 標記成本類別
- 將費用分割到成本類別內
- AWS Cost Categories 功能

相關範例：

- 使用 AWS Cost Categories 組織您的成本與用量資料
- 使用 AWS Cost Categories 管理您的成本
- Well-Architected 實驗室：成本與用量視覺化
- Well-Architected 實驗室：Cost Categories

COST03-BP04 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。

未建立此最佳實務時的風險暴露等級：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：與業務的利害關係人會談，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易將工作負載分成元件 (例如微型服務)，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-BP05 設定帳單和成本管理工具

設定符合組織政策的成本管理工具，以管理及最佳化雲端支出。其中包括以服務、工具和資源來組織及追蹤成本與用量資料、透過整合的帳單和存取許可加強控制、透過預算制定與預測提升規劃效能、接收通知或提醒，以及藉由資源與定價最佳化來降低成本。

未建立此最佳實務時的風險暴露等級：高

實作指引

若要建立健全的權責劃分，可考慮先將您的帳戶策略視為成本分配策略的一部分。正確做到這一點，應該就夠了。否則，後續會發生意料之外和棘手的問題。

若要促進雲端支出的權責劃分，請授予使用者對工具的存取權，以利用工具檢視其成本和用量。AWS 建議您針對下列目的設定所有工作負載和團隊：

- 組織：使用您自己的標記策略和分類法，來建立成本分配與管控基準。使用 AWS Control Tower 或 AWS 組織等工具建立多個 AWS 帳戶。標記受支援的 AWS 資源，並根據您的組織結構 (業務單位、部門或專案) 進行有意義的分類。標記特定成本中心的帳戶名稱，並與 AWS Cost Categories 對應，以將其成本中心的業務單位帳戶分組，讓業務單位擁有者可在一個位置查看多個帳戶的耗用量。
- 存取：在合併帳單中追蹤整個組織的帳單資訊。確認適當的利害關係人和企業擁有者具有存取權。

- 控制：使用適當防護機制建立有效控管機制，以避免在使用服務控制政策 (SCP)、標記政策、IAM 政策和預算警示時發生意外狀況。例如，您只能使用有效的控制機制，允許團隊在偏好的區域建立特定資源，並防止在沒有特定標籤的情況下建立資源 (例如成本中心)。
- 目前狀態：設定顯示目前成本和用量的儀表板。儀表板應置放在工作環境中顯眼的位置，就像營運儀表板那樣。您可以從 AWS 成本最佳化中心或任何受支援的產品匯出資料，並使用成本和用量儀表板來建立此可見性。您可能需要為不同角色建立不同的儀表板。例如，管理者儀表板可能與工程師儀表板不同。
- 通知：當成本或用量超出定義的限制，以及發生 AWS 預算或 AWS 成本異常偵測時，提供通知。
- 報告：總結所有成本和用量資訊。以詳細的可分配成本資料強化對雲端支出的認知與權責劃分。建立與取用報告的團隊相關且內含建議的報告。
- 追蹤：根據設定的總目標或具體目標顯示目前的成本和用量。
- 分析：允許團隊成員使用不同的篩選條件 (資源、帳戶、標籤等) 以每小時、每日或每月的精細度進行自訂及深入分析。
- 檢查：隨時掌握最新的資源部署和成本優化機會。使用 Amazon CloudWatch、Amazon SNS 或 Amazon SES 在組織層級部署資源時取得通知。使用 AWS Trusted Advisor 或 AWS Compute Optimizer 檢閱成本最佳化建議。
- 趨勢報告：以所需的精細度顯示所需時間範圍內的成本與用量變化。
- 預測：使用您所建立的預測儀表板顯示預估的未來成本，以及預估您的資源用量和支出。

您可以利用 AWS成本最佳化中心來了解從集中位置合併的潛在節省成本機會，並建立資料匯出以與 Amazon Athena 整合。您也可以使用 AWS 成本最佳化中心來部署成本和用量儀表板，該儀表板可利用 Amazon QuickSight 進行互動式成本分析，和安全的成本洞見分享。

如果您的組織中沒有基本技能或頻寬，您可以使用 AWS ProServ、AWS Managed Services (AMS) 或 AWS 合作夥伴。您也可以使用第三方工具，但請在那之前驗證價值主張。

實作步驟

- 允許以團隊為基礎的工具存取權：設定您的帳戶，並建立有權存取所需的成本與用量報告以了解取用情形的群組，使用 AWS Identity and Access Management 對 AWS Cost Explorer 之類的工具進行存取控制。這些群組必須包含擁有或管理應用程式的所有團隊中的代表。這可證明每個團隊都能存取其成本和用量資訊以追蹤取用情形。
- 組織成本標籤和類別：系統化跨團隊、業務單位、應用程式、環境和專案的整體成本。使用資源標籤按成本配置標籤來系統化成本。根據各種維度建立成本類別，並使用標籤、帳戶、服務等來對照成本。

- 設定 AWS 預算：針對您的工作負載在所有帳戶上設定 AWS Budgets。使用標籤和成本類別，設定所有帳戶支出的預算，以及工作負載的預算。在 AWS Budgets 中設定通知，以接收超出預算金額時或預估成本超出預算時的提醒。
- 設定AWS 成本異常偵測：將 AWS 成本異常偵測用於您的帳戶、核心服務或您所建立的成本類別，以監控成本與用量並，檢測異常支出狀況。您可以在彙總報告中個別接收提醒，也可以透過電子郵件或 Amazon SNS 主題接收提醒，以便分析和判斷發生異常的根本原因，並找出導致成本上升的因素。
- 使用成本分析工具：為您的工作負載和帳戶設定 AWS Cost Explorer，將成本資料視覺化以供進一步分析。根據歷史成本資料建立工作負載的儀表板，以追蹤整體支出、工作負載的關鍵用量指標，以及未來成本的預測。
- 使用節省成本分析工具：使用 AWS 成本最佳化中心，透過量身打造的建議來識別節約先機，包括刪除未曾使用的資源、調整規模大小、Savings Plans、預訂和運算最佳化工具建議。
- 設定進階工具：您可以選擇性建立視覺效果，以促進互動式分析和共享成本洞見。使用 AWS 成本最佳化中心上的資料匯出功能，可以為您的組織建立由 Amazon QuickSight 支援的成本和用量儀表板，此儀表板會提供額外的詳細資訊和精細度。您也可以透過使用 Amazon Athena 中的匯出資料進行進階查詢，以實施進階分析功能，並在 Amazon QuickSight 上建立儀表板。使用 AWS合作夥伴服務，採用雲端管理解決方案以整合雲端帳單監控和最佳化功能。

資源

相關文件：

- 什麼是 AWS Billing and Cost Management 和成本管理？
- 建立您的最佳實務 AWS 環境
- 標記 AWS 資源的最佳實務
- 標記您的 AWS 資源
- AWS Cost Categories
- 使用 AWS Budgets 分析成本
- 使用 AWS Cost Explorer 分析成本
- 什麼是 AWS 資料匯出？

相關影片：

- 部署雲端智慧儀表板
- 取得任何 FinOps 或成本最佳化指標或 KPI 的提醒

相關範例：

- 成本和用量儀表板由 Amazon QuickSight 提供支援
- AWS 成本和用量管控研討會

COST03-BP06 根據工作負載指標分配成本

根據用量指標或商業成果分配工作負載的成本，以衡量工作負載的成本效率。實作程序以透過分析服務 (可提供洞見和退款功能) 來分析成本和用量資料。

未建立此最佳實務時的風險暴露等級：低

實作指引

成本最佳化的意思是以最低的價格提供業務成果，這只有根據工作負載指標 (依照工作負載效率測量) 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。每小時執行一次此分析。如果您具有靜態成本元件 (例如，持續執行的後端資料庫) 且請求率不同 (例如，用量尖峰在早上九到晚上五點，晚上只有少量請求)，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，有助於讓您聚焦在最佳化活動上。

與 Amazon Elastic Container Service (Amazon ECS) 和 Amazon API Gateway 上的容器化應用程式等資源相比，為共用資源建立工作負載指標可能具有挑戰性。但是，您可以透過某些方式對用量進行分類，以及追蹤成本。如果您需要追蹤 Amazon ECS 和 AWS Batch 共用資源，可以在 AWS Cost Explorer 中啟用分割成本分配資料。使用分割成本分配資料，您可以了解並優化您的容器化應用程式的成本和用量，並根據共用運算和記憶體資源的使用情形，將應用程式成本分配給個別業務實體。

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籤，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

相關範例：

- 使用 AWS 分割成本分配資料提高 Amazon ECS 和 AWS Batch 的成本可見性

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。這樣做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-BP05 強制執行資料保留政策

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。

未建立此最佳實務時的風險暴露等級：高

實作指引

除役不再需要的工作負載資源。常見的範例是用於測試的資源：測試完成後，便可移除資源。使用標籤來追蹤資源 (並針對這些標籤執行報告) 可協助您識別要除役的資產 (不會再使用這些資產，或是其授權將到期時)。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。功能標記的範例值是## X ##，可識別資源在工作負載生命週期的用途。另一個範例是為資源 (例如，待刪除的標籤金鑰名稱和值) 使用 LifeSpan 或 TTL，以定義要除役的時段或特定時間。

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。標記可協助您依用途、團隊、環境或其他與您業務相關的準則，來將資源分類。如需標記使用案例、策略和技術的詳細資訊，請參閱 AWS 標記最佳實務。
- 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。如需未使用的資源或未充分利用的資源的詳細資訊，請參閱 AWS Trusted Advisor 成本最佳化檢查。
- 將 AWS 資源分組：為 AWS 資源建立群組。您可以使用 AWS Resource Groups 來組織和管理位於相同 AWS 區域 的 AWS 資源。您可以針對大多數的資源新增標籤，以便識別和排序組織內的資源。使用 Tag Editor 可大量地對受支援的資源新增標籤。請考慮使用 AWS Service Catalog 來建立、管理和分配已核准的產品組合給終端使用者，以及管理產品的生命週期。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- AWS Trusted Advisor 成本最佳化檢查
- 標記 AWS 資源
- 發布自訂指標

相關影片：

- 如何使用 AWS Trusted Advisor 將成本最佳化

相關範例：

- 組織 AWS 資源
- 使用 AWS Trusted Advisor 將成本最佳化

COST04-BP02 實作除役程序

實作識別和除役未使用資源的程序。

未建立此最佳實務時的風險暴露等級：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。此程序應該要定義執行搜尋的頻率，以及移除資源的程序，以便驗證是否有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。詳述除役資源的必要步驟，將它們從服務中移除，同時確保符合任何的法規要求。應包含任何關聯的資源，例如授權或連接的儲存。發出通知讓工作負載擁有者知道除役程序已經執行。

  使用下列除役步驟來引導您了解過程中應檢查的事項：

  - 識別要除役的資源：識別 AWS 雲端 中符合除役資格的資源。記錄所有必要資訊，並排定除役時間。在規劃時間表時，請務必考慮到過程中可能會發生沒預期到的問題。
  - 協調和溝通：與工作負載擁有者合作，確認要除役的資源
  - 記錄中繼資料並建立備份：如果是生產環境資源的必要項目或如果是重要資源，請記錄中繼資料 (例如公用 IP、區域、AZ、VPC、子網路和安全群組) 並建立備份 (例如 Amazon Elastic Block Store 快照或擷取 AMI、金鑰匯出和憑證匯出)。
  - 驗證基礎設施即程式碼：確定資源的部署工具是 AWS CloudFormation、Terraform、AWS Cloud Development Kit (AWS CDK) 或任何其他基礎設施即程式碼部署工具，以便能在必要時加以重新部署。
  - 防止存取：套用限制性控制措施一段時間，以便在您確定資源必要性時，防止有人使用資源。確認資源環境可在必要時恢復為原始狀態。
  - 遵循內部除役程序：遵循組織的管理任務和除役程序，例如移除組織網域內的資源、移除 DNS 記錄，以及移除組態管理工具、監控工具、自動化工具和安全工具內的資源。

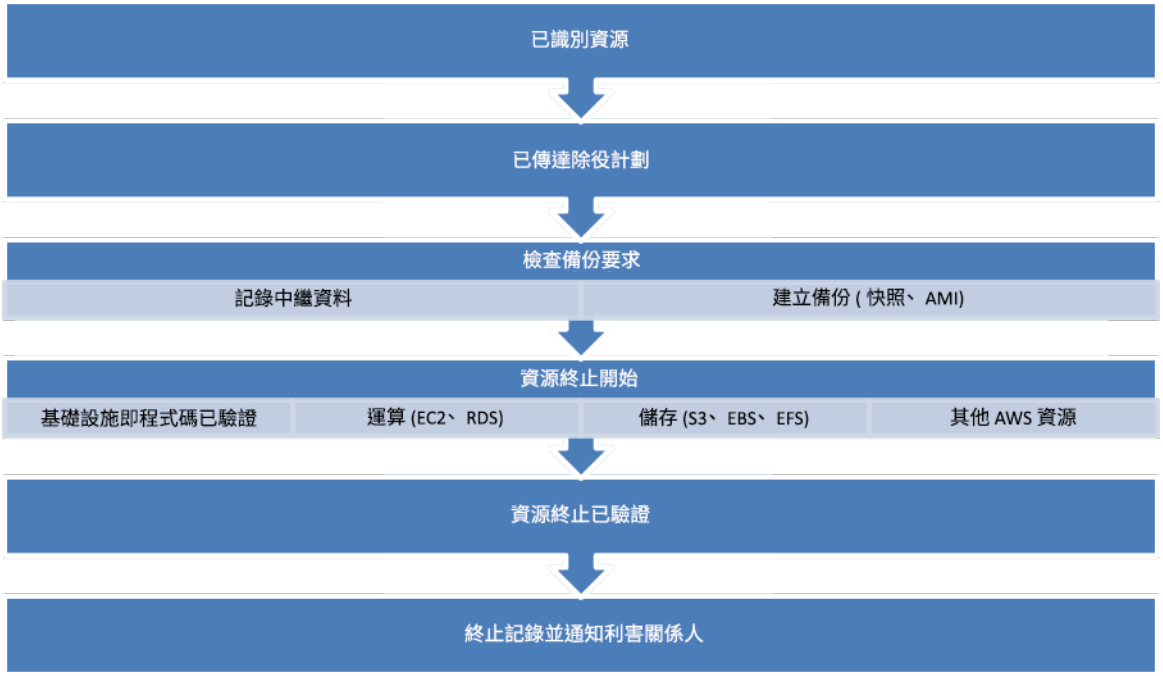
  如果有 Amazon EC2 執行個體，請參考下列清單。如需詳細資訊，請參閱「如何刪除或終止 Amazon EC2 資源？」

  - 停止或終止所有 Amazon EC2 執行個體和負載平衡器。Amazon EC2 執行個體終止後，仍會在主控台中短暫顯示。您不需要為任何未處於執行中狀態的執行個體付費
  - 刪除您的 Auto Scaling 基礎設施。
  - 釋放所有專用執行個體。
  - 刪除所有 Amazon EBS 磁碟區和 Amazon EBS 快照。
  - 釋放所有彈性 IP 地址。
  - 取消註冊所有 Amazon Machine Image (AMI)。

- 終止所有 AWS Elastic Beanstalk 環境。

如果資源是 Amazon S3 Glacier 儲存中的物件，而且在封存未達最低儲存持續時間之前就將其刪除，則會按比例向您收取過早刪除費 (Amazon S3 Glacier 的最低儲存持續時間取決於所使用的儲存類別)。如需每個儲存類別的最低儲存持續時間摘要，請參閱各種 Amazon S3 儲存類別的效能。如需過早刪除費的計算方式，請參閱 Amazon S3 定價。

下面的簡單除役程序流程圖會概述除役步驟。在將資源除役之前，請先確認組織已不再使用這些識別為要除役的資源。



資源除役流程。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- AWS CloudTrail

相關影片：

- 刪除 CloudFormation 堆疊但保留某些資源

- 了解是哪位使用者啟動了 Amazon EC2 執行個體

相關範例：

- 刪除或終止 Amazon EC2 資源
- 了解是哪位使用者啟動了 Amazon EC2 執行個體

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，其執行方式可以手動，也可以自動。

未建立此最佳實務時的風險暴露等級：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高成本帳戶低。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：對於不再需要或是授權合約結束的 AWS 資源來說，這是折舊階段。請先完成所有最終檢查，再移至處置階段並將資源除役，以防止發生任何不需要的中斷，例如擷取快照或備份。使用除役程序，除役已識別為未使用的每個資源。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor

相關範例：

- Well-Architected 實驗室：除役資源 (Level 100)

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。

未建立此最佳實務時的風險暴露等級：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 來執行除役程序。您也可以使用 API 或 SDK 實作自訂程式碼，自動除役工作負載資源。

現代化應用程式會優先以無伺服器方式來建置，這個策略會優先採用無伺服器服務。AWS 針對下列三個堆疊層都開發了無伺服器服務：運算、整合和資料存放區。使用無伺服器架構可讓您透過自動縱向擴展和縮減規模，在低流量期間節省成本。

實作步驟

- 實作 AWS Auto Scaling：對於受支援的資源，請使用 AWS Auto Scaling 來為其進行設定。AWS Auto Scaling 可以協助您在取用 AWS 服務時，獲得最佳的使用率和成本效率。當需求下降時，AWS Auto Scaling 會自動移除超額的資源容量，以免您超支。
- 設定 CloudWatch 來終止執行個體：執行個體可以設定為使用 CloudWatch 警示加以終止。使用來自於除役程序的指標，以 Amazon Elastic Compute Cloud 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 將工作負載資源除役。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。
- 使用無伺服器服務：在 AWS 上優先建置無伺服器架構和事件驅動架構，以建置和執行應用程式。AWS 提供了多個無伺服器技術服務，這些服務本身就會提供自動最佳化的資源使用率和自動化的除役 (縮減和橫向擴展)。在使用無伺服器應用程式時，系統會自動為您提供最佳化的資源使用率，您永遠不會因為過度佈建而支付費用。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- AWS 上的無伺服器

- 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
- 在 Amazon CloudWatch 警示中新增終止動作

相關範例：

- 排程自動刪除 AWS CloudFormation 堆疊
- Well-Architected 實驗室：自動除役資源 (Level 100)
- Servian AWS 自動清理

COST04-BP05 強制執行資料保留政策

對支援的資源定義資料保留政策，以根據組織的要求處理物件刪除。識別並刪除不再需要的非必要或孤立資源與物件。

未建立此最佳實務時的風險暴露等級：中

使用資料保留政策和生命週期政策，降低已識別資源的除役程序相關成本和儲存成本。定義資料保留政策和生命週期政策以執行自動化儲存類別遷移和刪除，可降低生命週期內的整體儲存成本。您可以使用 Amazon Data Lifecycle Manager 自動建立和刪除 Amazon Elastic Block Store 快照與 Amazon EBS 支援的 Amazon Machine Image (AMI)，並且可使用 Amazon S3 Intelligent-Tiering 或 Amazon S3 生命週期組態來管理 Amazon S3 物件的生命週期。您也可以使用 API 或 SDK，針對要自動刪除的物件建立生命週期政策和政策規則。

實作步驟

- 使用 Amazon Data Lifecycle Manager：對 Amazon Data Lifecycle Manager 使用生命週期政策，以自動刪除 Amazon EBS 快照和 Amazon EBS 支援的 AMI。
- 設定儲存貯體的生命週期組態：對儲存貯體使用 Amazon S3 生命週期組態，定義要讓 Amazon S3 在物件的生命週期內執行的動作，以及根據業務要求在物件的生命週期結束時進行的刪除。

資源

相關文件：

- AWS Trusted Advisor
- Amazon Data Lifecycle Manager

- 如何設定 Amazon S3 儲存貯體的生命週期組態

相關影片：

- 使用 Amazon Data Lifecycle Manager 自動執行 Amazon EBS 快照
- 使用生命週期組態規則清空 Amazon S3 儲存貯體

相關範例：

- 使用生命週期組態規則清空 Amazon S3 儲存貯體
- Well-Architected 實驗室：自動除役資源 (Level 100)

具有經濟效益的資源

問題

- COST 5.如何在選取服務時評估成本？
- COST 6.如何在選取資源類型、大小和數量時達成成本目標？
- COST 7.如何使用定價模式降低成本？
- COST 8.如何規劃資料傳輸費？

COST 5.如何在選取服務時評估成本？

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。選取適當的基礎和受管服務，就可最佳化此工作負載的成本。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 對不同用量執行一段時間內的成本分析

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。

未建立此最佳實務時的風險暴露等級：高

實作指引

大多數組織的資訊技術 (IT) 部門會由多個小型團隊組成，每個團隊都有自己的議程和重點領域，而這會反映出其團隊成員的專業和技能。您需要了解組織的整體目標、優先順序、目標，以及每個部門或專案如何為這些目標做出貢獻。對於實現組織目標和全面預算規劃來說，將所有重要資源進行分類至關重要，這些資源包括人員、設備、技術、材料和外部服務。採用這種系統化方法來識別和了解成本，是為組織建立實際、可靠成本計畫的基礎。

為工作負載選取服務時，關鍵是了解組織的優先事項。在成本最佳化和其他 AWS Well-Architected Framework 支柱之間取得平衡，例如效能和可靠性。此流程應有系統且定期地進行，以反映組織目標、市場條件和營運動態的變化。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。與組織中的所有團隊 (例如產品、業務、技術和財務團隊) 會面以收集資訊。評估在相互衝突的利益或替代方法之間做出權衡的影響，以協助您在確認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。

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。輸出應為依序列出的支柱清單。您也可以為每個支柱新增加權，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。
- 解決技術負債並加以記錄：在工作負載審查期間，請解決技術負債。記錄積存項目以在將來重新檢視工作負載，目標是重構或重新架構以將工作負載進一步最佳化。向其他利害關係人清楚傳達所做出的權衡至關重要。

資源

相關的最佳實務：

- REL11-BP07 建立您的產品架構以符合可用性目標和運行時間服務水準協議 (SLA)

- OPS01-BP06 評估權衡

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 分析工作負載的所有元件

確認會分析每個工作負載元件，無論目前大小或目前成本為何。審查工作應反映潛在的效益，例如當前和預計的成本。

未建立此最佳實務時的風險暴露等級：高

實作指引

旨在為組織提供商業價值的工作負載元件，可能涵蓋多種服務。您可以針對每個元件選擇特定的 AWS 雲端 服務來滿足業務需求。這個選擇可能會受到熟悉與否或之前使用這些服務的經驗等因素所影響。

在確定組織的要求 (如 COST05-BP01 確定組織的成本要求) 後，請對工作負載中的所有元件執行徹底的分析。考慮當前和預測的成本與大小來分析每個元件。針對工作負載生命週期中的任何潛在工作負載節約來考量分析的成本。在分析此工作負載的所有元件上所付出的努力，應與最佳化該特定元件所預期的潛在節約或改進相當。例如，如果所提議資源的成本是每月 10 美元，而低於預測的負載不會超過每月 15 美元，則努力一天以減少 50% 成本 (每月 5 美元) 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預估，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有所改變，那麼適當的服務組合不一定會是最佳組合。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。例如，使用 EMR Serverless 最初可能是合適的選擇。但是，隨著該服務的取用量增加，轉換到 EC2 上的 EMR 可以降低工作負載中該元件的成本。

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。

撰寫一個工作流程供技術團隊遵循，用以審核其工作負載。讓這個工作流程保持簡單，但也涵蓋所有必要的步驟，以確保各團隊了解每個工作負載元件及其定價。然後，您的組織可以根據每個團隊的特定需求，遵循並自訂此工作流程。

1. 列出工作負載使用的每個服務：這是一個很棒的起點。識別目前使用中的所有服務以及成本來源。
2. 了解這些服務的定價方式：了解每項服務的定價模式。根據用量、資料傳輸和特定功能定價等因素，不同的 AWS 服務具有不同的定價模式。
3. 專注於具有非預期工作負載成本，且與您預期的用量和業務成果不符的服務：識別成本與使用 AWS Cost Explorer 或 AWS Cost and Usage Report 的價值或用量不成比例的異常值或服務。務必將成本與業務成果相關聯，如此才能確定最佳化作業的優先順序。
4. AWS Cost Explorer、CloudWatch Logs、VPC Flow Logs 和 Amazon S3 Storage Lens，以了解這些高成本的根本原因：這些工具有助於「診斷」高成本。每項服務都提供不同的角度來檢視分析用量與成本。例如，Cost Explorer 會協助判斷整體成本趨勢、CloudWatch Logs 會提供營運洞見、VPC Flow Logs 會顯示 IP 流量，而 Amazon S3Storage Lens 對於儲存分析非常有用。
5. 將 AWS Budgets 用於設定服務或帳戶的特定金額預算：設定預算是主動式的管理成本方式。將 AWS Budgets 用於設定自訂預算閾值，並在成本超過這些閾值時接收提醒。
6. 設定 Amazon CloudWatch 警示以傳送帳單和用量提醒：設定成本和用量指標的監控和提醒。當達到某些閾值時，CloudWatch 警示會通知您，進而改善介入回應時間。

透過策略性審查所有工作負載元件，無論其目前屬性為何，都會隨著時間產生顯著的增強和節省更多成本。在這個審查流程中所投入的努力應經過深思熟慮，並仔細考慮可能實現的潛在優勢。

實作步驟

- 列出工作負載元件：建置工作負載元件的清單。使用此清單來驗證是否已分析每個元件。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。將功能性相同的資源分在同一組，有助於提高效率 (如果有多個資料庫，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，並依照工作付出程度安排優先順序。這通常是依最昂貴到最便宜的元件成本，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
- AWS 雲端 產品

相關影片：

- AWS 成本最佳化系列：CloudWatch

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。考量營運和管理成本以計算總體擁有成本，尤其是在使用雲端供應商的受管服務時。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)。

未建立此最佳實務時的風險暴露等級：高

實作指引

考量如何節省時間，讓您的團隊能夠專注於淘汰技術負債、創新和附加價值功能，以及創造企業與眾不同之處。例如，您可能需要將內部部署環境中的資料庫盡快「隨即轉移」至雲端 (也稱為主機轉換)，然後進行優化。能否使用 AWS 上的受管服務以去除或降低授權成本，進而獲得節省的效益，是值得探討的。AWS 上的受管服務免除了維護服務的營運和管理重擔 (例如修補或升級作業系統)，讓您得以專注於創新和業務。

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。您可以進行可能的優化以獲得實際的好處，且無須變更應用程式的核心架構。例如，您可能會想要藉由遷移至資料庫即服務平台 (例如 Amazon Relational Database Service (Amazon RDS)) 或將應用程式遷移至全受管平台 (例如 AWS Elastic Beanstalk)，來縮短管理資料庫執行個體所花費的時間。

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，使資源需求與持續變動的需求保持一致。例如，您可將 Amazon EMR 叢集 (或 Amazon Redshift 叢集) 上的節點數量增加或減少，以進行橫向擴展或縮減。

您也可將多個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，將其中一個資料庫遷移至專用 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 提供受管分析服務。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。
- 比較受管和未受管資源：針對您所管理的資源考量營運成本，並將其與 AWS 受管資源比較。例如，審查您在 Amazon EC2 執行個體上執行的資料庫，並且與 Amazon RDS 選項 (AWS 受管服務) 比較，或將 Amazon EMR 相較於在 Amazon EC2 上執行 Apache Spark。從自我管理工作負載移轉至 AWS 全受管工作負載時，請仔細研究您的選項。應考量的三大因素，是您要使用的受管服務類型、您將用來遷移資料的程序，以及了解 AWS 共同責任模式。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- AWS 雲端 產品
- AWS 共同責任模式

相關影片：

- 為何要移轉至受管資料庫？
- 什麼是 Amazon EMR？如何用它來處理資料？

相關範例：

- 為何要移轉至受管資料庫
- 使用 AWS DMS 將相同 SQL Server 資料庫中的資料合併到單一 Amazon RDS for SQL Server 資料庫中
- 大規模將資料遞送至 Amazon Managed Streaming for Apache Kafka (Amazon MSK)
- 將 ASP.NET Web 應用程式遷移至 AWS Elastic Beanstalk

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。

未建立此最佳實務時的風險暴露等級：低

實作指引

開放原始碼源於軟體開發的背景，以指出該軟體符合某些免費發行條件。開放原始碼軟體會由任何人都可以檢查、修改和增強的原始程式碼組成。根據業務要求、工程師的技能、預測用量或其他技術相依性，組織可以考慮使用 AWS 上的開放原始碼軟體，以最大程度地降低其授權成本。換句話說，使用開放原始碼軟體可降低軟體授權成本。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。

請根據總成本來測量授權軟體的效益，以將工作負載最佳化。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。

例如，相較於執行另一個在 Windows 上執行的 Amazon EC2 執行個體，使用 Linux 作業系統在 us-east-1 中操作 Amazon EC2 執行個體可讓您削減大約 45% 的成本。

AWS Pricing Calculator 提供了全面性的方法讓您比較各種資源與不同授權選項的成本 (例如 Amazon RDS 執行個體和不同的資料庫引擎)。此外，AWS Cost Explorer 還為現有工作負載的成本提供了寶貴的觀點，尤其是具有不同授權的工作負載的成本。對於授權管理，AWS License Manager 提供了簡化的方法讓您監督和處理軟體授權。客戶可以在 AWS 雲端 中部署和操作自己喜歡的開放原始碼軟體。

實作步驟

- 分析授權選項：檢閱可用軟體的授權條款。尋找具有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。有利條款會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，尋找他們執行稽核和可能施加的懲罰的方式。

資源

相關文件：

- AWS 上的開放原始碼

- AWS 總體擁有成本 (TCO) 計算器
- Amazon S3 儲存類別
- 雲端產品

相關範例：

- 開放原始碼部落格
- AWS 開放原始碼部落格
- 最佳化和授權評定

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

選取工作負載的所有元件時均應考量成本。這包括使用應用程式層級和受管服務或無伺服器、容器或事件驅動架構，以降低整體成本。使用開放原始碼軟體、無需授權費用的軟體或替代方案，藉以將授權成本降至最低。

未建立此最佳實務時的曝險等級： 中

實作指引

選取所有元件時均應考量服務和選項的成本。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 和 Amazon Simple Storage Service (Amazon S3) 靜態網路適用。在情況允許時將應用程式容器化，並使用 AWS 受管容器服務，例如 Amazon Elastic Container Service (Amazon ECS) 或 Amazon Elastic Kubernetes Service (Amazon EKS)。

使用開放原始碼軟體或沒有授權費用的軟體，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)。

您可以使用無伺服器或應用程式層級服務，例如 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，這些服務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。

使用 事件驅動架構 也可以搭配無伺服器服務。事件驅動架構是推送架構，因此一切都會在事件呈現於路由器時隨需進行。如此，您就無須付費持續進行輪詢以檢查事件。這表示網路頻寬耗用量、CPU 使用率、閒置機群容量和 SSL/TLS 交握都可降低。

如需有關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。與其增加容量以符合需求，您應考慮使用其他選項，以較低的成本獲得更好的效能。例如，您應審查資料庫在 AWS 上的預期流量，並考慮增加執行個體大小，或使用 Amazon ElastiCache 服務 (Redis 或 Memcached) 為資料庫提供快取的機制。
- 評估事件驅動架構： 使用無伺服器架構也可讓您為分散式微型服務應用程式建置事件驅動架構，以利設計可擴展、彈性、敏捷且符合成本效益的解決方案。

資源

相關文件：

- AWS 總體擁有成本 (TCO) 計算器
- AWS 無伺服器
- 什麼是事件驅動架構
- Amazon S3 儲存類別
- 雲端產品
- Amazon ElastiCache for Redis

相關範例：

- 事件驅動架構入門
- 事件驅動架構
- 如何利用 Amazon ElastiCache for Redis 提高成本效益達 100 倍以上
- 使用 AWS Lambda 函數的最佳實務

COST05-BP06 對不同用量執行一段時間內的成本分析

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。按預計用量對每個元件執行一段時間內的分析，讓工作負載在其生命週期內保持成本效益。

未建立此最佳實務時的風險暴露等級：中

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。

如果需要將資料移至 AWS 雲端 中，您可以選取 AWS 所提供的各種服務以及合作夥伴工具，以便遷移您的資料集，無論是檔案、資料庫、機器映像、區塊磁碟區甚或磁帶備份均可。例如，若要對 AWS 移入或移出大量資料，或是在邊緣處理資料，您可以使用其中一項 AWS 專用裝置，以符合成本效益的方式離線移動數以 PB 計的資料。另一個範例是，在資料傳輸速率較高時，直接連線服務可能會比 VPN 更便宜，為您的企業提供所需的連線能力。

根據對不同用量在一段時間內的成本分析，審查您的擴展活動。分析結果，確認是否可以調整擴展政策，以使用多個執行個體類型和購買選項新增執行個體。審查您的設定，確認是否可以降低最小值，以較小的機群大小處理使用者要求，以及新增更多資源以符合預期的高需求。

與組織內的利害關係人討論，並使用 AWS Cost Explorer 的預測功能對服務變更的潛在影響進行預測，藉以對不同用量執行一段時間內的成本分析。使用 AWS Budgets、CloudWatch 帳單警示和 AWS Cost Anomaly Detection 監控用量等級觸發程序，以快速識別及實作最符合成本效益的服務。

實作步驟

- 定義預測使用模式：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使用模式。與業務利害關係人討論關於歷史和預測成本與用量增加的議題，並確定這類增加符合業務要求。識別您預期會有較多使用者使用 AWS 資源的日曆日、週或月，這表示您應增加現有資源的容量或採用其他服務，以降低成本並提升效能。
- 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。分析工作應反映潛在成果。例如，如果用量變化很大，則應執行徹底的分析以驗證任何成本和變化。換句話說，當成本增加時，企業的用量也應增加。

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

- Amazon EC2 Auto Scaling
- 雲端資料遷移
- AWS Snow Family

相關影片：

- AWS OpsHub for Snow Family

COST 6.如何在選取資源類型、大小和數量時達成成本目標？

確認您為手邊的任務選取適當的資源大小和數量。選取最具成本效益的類型、大小和數量，就能盡量減少浪費。

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-BP04 考慮使用共用資源

COST06-BP01 執行成本建模

識別組織要求 (例如商業需求和現有承諾)，並對工作負載及其每個元件執行成本建模 (整體成本)。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。

未建立此最佳實務時的風險暴露等級：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。了解成本考量，可在評估計劃性工作負載部署的價值實現成果時，傳達組織的商業案例和決策程序。

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 AWS Well-Architected Framework 的效能效率要素的審查一節。

例如，若要為包含運算資源的工作負載建立成本模型，AWS Compute Optimizer 將有助於對執行中的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。請確定 CloudWatch Agent 已部署至 Amazon EC2 執行個體以收集記憶體指標，可在 AWS Compute Optimizer 內為您提供更精確的建議。這是運算資源的理想資料來源，因為它是免費服務，並使用機器學習根據風險等級提出多個建議。

有多個服務可與自訂日誌搭配使用，作為其他服務和工作負載元件 (例如 AWS Trusted Advisor、Amazon CloudWatch 和 Amazon CloudWatch Logs) 適當調整大小操作的資料來源。AWS Trusted Advisor 會檢查資源，並標示使用率偏低的資源，以協助您為資源適當調整大小以及建立成本模型。

以下是成本建模資料和指標的建議：

- 監控必須精確反映使用者體驗。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。
- 考量您現有的承諾、為其他工作負載選取的定價模式，以及加速創新和專注於核心業務價值的能力，藉此為您的計劃性工作負載選擇正確的 AWS 服務。

實作步驟

- 執行資源的成本建模：將工作負載或概念驗證部署到有特定資源類型和大小要測試的獨立帳戶。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，並再次執行測試。納入可能用於這些資源之任何產品的授權費用，以及在建立成本模型時部署和管理這些資源的預估營運 (勞工或工程師) 成本。考慮建立一段時間 (每小時、每日、每月、每月或三年) 的成本模型。

資源

相關文件：

- AWS Auto Scaling
- 識別適當調整大小的機會
- Amazon CloudWatch 功能
- 成本優化：Amazon EC2 適當調整大小
- AWS Compute Optimizer

- AWS 定價計算器

相關範例：

- 執行資料驅動型成本建模
- 預估計劃性 AWS 資源組態的成本
- 選擇適當的 AWS 工具

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。

未建立此最佳實務時的風險暴露等級：中

實作指引

Amazon EC2 提供各種執行個體類型，其各自具有不同等級的 CPU、記憶體、儲存和網路容量，適合不同的使用案例。這些執行個體類型具有 CPU、記憶體、儲存和網路功能的不同組合，可讓您在選取適合專案的資源組合時獲得多樣選擇。每個執行個體類型都有多種大小，因此您可以根據工作負載的需求調整資源。若要判斷您需要的執行個體類型，請收集有關您計劃在執行個體上執行之應用程式或軟體系統要求的詳細資訊。這些詳細資訊應包括以下內容：

- 作業系統
- CPU 核心數量
- GPU 核心
- 系統記憶體 (RAM) 數量
- 儲存類型和空間
- 網路頻寬要求

確定運算要求的目的以及需要的執行個體，然後探索各種 Amazon EC2 執行個體系列。Amazon 提供下列執行個體類型系列：

- 一般用途
- 運算最佳化
- 記憶體最佳化

- 儲存最佳化
- 加速運算
- HPC 最佳化

若要深入了解特定 Amazon EC2 執行個體系列可以滿足的具體目的和使用案例，請參閱 AWS 執行個體類型。

收集系統要求對於您選取最適合需求的特定執行個體系列和執行個體類型來說非常重要。執行個體類型的名稱由系列名稱和執行個體大小組成。例如，t2.micro 執行個體來自 T2 系列，並且是微型大小。

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 (白皮書或已發佈的解決方案) 來進行此選擇。使用 AWS 定價計算器或成本管理工具可協助您對執行個體類型、大小和組態做出明智的決策。

實作步驟

- 根據資料選取資源：使用成本建模資料來選取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。依據成本建模資料，決定虛擬 CPU 數目、總記憶體 (GiB)、本機執行個體儲存體磁碟區 (GB)、Amazon EBS 磁碟區和網路效能等級，並將執行個體所需的資料傳輸速率納入考量。一律根據詳細分析和準確的資料進行選取，以最佳化效能，同時有效地管理成本。

資源

相關文件：

- AWS 執行個體類型
- AWS Auto Scaling
- Amazon CloudWatch 功能
- 成本優化：EC2 調整大小

相關影片：

- 為您的工作負載選取合適的 Amazon EC2 執行個體
- 為服務調整適當大小

相關範例：

- 探索和比較 Amazon EC2 執行個體類型變得更容易

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。為運算、儲存、資料和聯網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。

未建立此最佳實務時的風險暴露等級：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 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 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

- 設定工作負載指標來提升可觀察性：擷取工作負載的重要指標。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。針對運算資源，請分析效能資料以將 Amazon EC2 執行個體調整到適當大小。識別閒置的執行個體，以及未充分使用的執行個體。要尋找的重要指標是 CPU 使用率和記憶體使用率 (例如，90% 的時間有 40% CPU 使用率，如在 AWS Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小)。識別在四週期間內，CPU 使用率達到最大且記憶體使用率小於 40% 的執行個體。這些便是需要適當調整大小以降低成本的執行個體。針對 Amazon S3 之類的儲存資源，您可以使用 Amazon S3 Storage Lens，以便在儀表板中查看儲存貯體層級中各種類別的 28 項指標，以及 14 天的歷史資料 (預設值)。您可以依摘要和成本最佳化或事件來篩選 Amazon S3 Storage Lens 儀表板，以分析特定指標。
- 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以及成本管理主控台中的 Amazon EC2 適當調整大小工具，或檢閱 AWS Trusted Advisor 的適當調整資源大小以在工作負載上進行調整。在為不同資源調整適當大小時，無論其為 Amazon EC2 執行個體、AWS 儲存類別或 Amazon RDS 執行個體類型，都請務必使用合適的工具，並遵循適當調整大小指導方針。針對儲存資源，您可以使用 Amazon S3 Storage Lens，以便能夠檢視物件儲存用量、活動趨勢並提出可行建議，以將成本最佳化並套用資料保護最佳實務。使用 Amazon S3 Storage Lens 透過分析組織內的指標而得出的適合情境的建議，您可以立即採取行動來將儲存最佳化。

- 根據指標自動選取資源類型和大小：使用工作負載指標，手動或自動選取您的工作負載資源。針對運算資源，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。您可以啟動並自動擴展單一 Auto Scaling 群組內的一組隨需執行個體和 Spot 執行個體。除了獲得使用 Spot 執行個體的折扣外，您還可以使用預留執行個體或 Savings Plan 來獲得常規隨需執行個體定價的折扣費率。這些因素合在一起可協助您將 Amazon EC2 執行個體所能節省的成本最佳化，並確定應用程式所需的規模和效能。您也可以在 Auto Scaling Groups (ASG) 中使用屬性型執行個體類型選取 (ABS) 策略，以透過一組屬性 (例如 vCPU、記憶體和儲存) 來表達您的執行個體要求。您可以自動使用新發行的較新一代執行個體類型，並使用 Amazon EC2 Spot 執行個體來存取更大範圍的容量。Amazon EC2 機群和 Amazon EC2 Auto Scaling 會選取和啟動符合指定屬性的執行個體，您不必再手動挑選執行個體類型。針對儲存資源，您可以使用 Amazon S3 Intelligent Tiering 和 Amazon EFS Infrequent Access 功能，以便能夠在資料存取模式發生改變時，自動選取可自動節省儲存成本的儲存類別，卻又不會影響效能或造成營運負擔。

資源

相關文件：

- AWS Auto Scaling
- AWS 適當調整大小
- AWS Compute Optimizer
- Amazon CloudWatch 功能
- CloudWatch 設定
- CloudWatch 發布自訂指標
- Amazon EC2 Auto Scaling 入門
- Amazon S3 Storage Lens
- Amazon S3 Intelligent-Tiering
- Amazon EFS Infrequent Access
- 使用 SDK 啟動 Amazon EC2 執行個體

相關影片：

- 為服務調整適當大小

相關範例：

- Amazon EC2 機群的 Auto Scaling 屬性型執行個體類型選取
- 使用排定的擴展來將 Amazon Elastic Container Service 的成本最佳化
- Amazon EC2 Auto Scaling 的預測擴展
- 使用 Amazon S3 Storage Lens 將成本最佳化並獲得用量檢視能力
- Well-Architected 實驗室：適當調整大小的建議 (Level 100)
- Well-Architected 實驗室：在 AWS Compute Optimizer 和記憶體使用率已啟用的情況下適當調整大小 (Level 200)

COST06-BP04 考慮使用共用資源

對於已在組織層級部署多個業務單位的服務，請考慮使用共用資源來提高使用率，並降低總體擁有成本 (TCO)。使用共用資源時，可以利用現有解決方案、共用元件或兩者，以同時集中管理和處理成本，進而達到發揮成本效益之目的。管理通用功能，例如監控、備份和連線能力，無論是在帳戶邊界或在專用帳戶中。您也可以透過實施標準化、減少重複和降低複雜度來降低成本。

未建立此最佳實務時的風險暴露等級：中

實作指引

如果多個工作負載都發生相同的功能，請使用現有的解決方案和共用元件來改善管理並最佳化成本。考慮使用現有資源 (特別是共用資源，例如非生產資料庫伺服器或目錄服務)，以遵循安全性最佳做法和組織法規來降低雲端成本。為了發揮最佳價值和效率，將成本分配 (透過用量文件和撤銷付款) 回到促進消費的業務相關領域至關重要。

用量文件是指將雲端成本細分為可歸因類別的報表，例如消費者、業務單位、總帳帳戶或其他負責實體。用量文件的目標是向團隊、業務單位或個人展示他們耗用的雲端資源成本。

撤銷付款意指根據適合特定財務管理流程的策略，將中央服務支出分配給成本單位。對客戶而言，撤銷付款是將一個共用服務帳戶產生的成本，計入適合客戶報告流程的不同財務成本類別。透過建立撤銷付款機制，您可以報告不同業務單位、產品和團隊所產生的成本。

工作負載可以分類為關鍵和非關鍵。根據此分類，使用一般組態的共用資源，以減少關鍵工作負載。為了進一步最佳化成本，請僅為關鍵工作負載保留專用伺服器。在多個帳戶之間共用資源或佈建資源，以有效管理資源。即使在不同的開發、測試和生產環境下，安全共用也是可行做法，而且不會影響組織結構。

若要進一步了解並最佳化容器化應用程式的成本和使用情況，請使用分割成本分配資料，這些資料可協助您根據應用程式使用共用運算和記憶體資源的方式，將成本分配給個別業務實體。分割成本分配資料

能協助您在 Amazon Elastic Container Service (Amazon ECS) 或 Amazon Elastic Kubernetes Service (Amazon EKS) 上執行的容器工作負載中，實現任務層級的用量文件和撤銷付款。

針對分散式架構，建立共用服務 VPC，而該服務可提供集中存取每個 VPC 中工作負載所需的共用服務。這些共用服務可包括目錄服務或 VPC 端點等資源。若要降低管理開銷和成本，請從中央位置共用資源，而不是在每個 VPC 中建立資源。

使用共用資源可以幫您節省營運成本、將資源使用率提升到最高，並改善一致性。在多帳戶設計中，您可以集中託管某些 AWS 服務，並使用中心的數個應用程式和帳戶存取服務，以節省成本。您可以使用 AWS Resource Access Manager (AWS RAM) 來共用其他常見的資源，例如 VPC 子網路和 AWS Transit Gateway 附件、AWS Network Firewall 或 Amazon SageMaker 管道。在多帳戶環境中，使用 AWS RAM 可讓您建立一次資源，並與其他帳戶共享資源。

各組織應有效地標記共用成本，並驗證其成本的大部分並沒有未標記或未分配的情況。如果您未有效分配共用成本，也沒有人負責管理共用成本，則共用雲端成本可能會呈螺旋式上升。您應該知道在資源、工作負載、團隊或組織層級上哪些地方產生了成本，因為與所達到的業務成果相比時，您就會對在哪個層級交付何種價值有更深刻的了解。最後，各組織會因共用雲端基礎設施所節省下來的成本而受益。鼓勵分配共用雲端資源的成本，以最佳化雲端支出。

實作步驟

- 評估現有資源：檢閱針對您的工作負載使用類似服務的現有工作負載。視工作負載的元件而定，請在商業邏輯或技術需求許可下，考慮採用現有平台。
- 在 AWS RAM 中使用共用資源並據以施加限制：使用 AWS RAM 以與組織內的其他 AWS 帳戶共用資源。共用資源時，不需要複製多個帳戶中的資源，如此可將資源維護的作業負擔減到最低。此流程也會幫助您安全地將您使用帳戶中的角色和使用者所建立的資源與其他 AWS 帳戶 共用。
- 標記資源：標記成本報告的候選資源，並在成本類別中為這些資源分類。啟用這些成本相關資源標籤來分配成本，以視覺化方式呈現 AWS 資源用量。專注在成本和用量可見度方面建立適當的精細度，並利用成本分配報告和 KPI 追蹤影響雲端消費行為。

資源

相關的最佳實務：

- SEC03-BP08 在組織內安全地共用資源

相關文件：

- 什麼是 AWS Resource Access Manager？

- 您可以搭配 AWS Organizations 使用的 AWS 服務
- 可共用的 AWS 資源
- AWS 成本與用量 (CUR) 查詢

相關影片：

- AWS Resource Access Manager - 具有受管許可的精細存取控制
- 如何設計 AWS 成本分配策略
- AWS Cost Categories

相關範例：

- 如何撤銷付款共用服務：AWS Transit Gateway 範例
- 如何使用 CUR 為 Savings Plans 建立撤銷付款/用量文件模式
- 針對符合成本效益的多帳戶微型服務架構使用 VPC 共享
- 使用 AWS 分割成本分配資料提高 Amazon EKS 的成本可見性
- 使用 AWS 分割成本分配資料提高 Amazon ECS 和 AWS Batch 的成本可見性

COST 7.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，就能盡量減少支出。

最佳實務

- COST07-BP01 執行定價模式分析
- COST07-BP02 根據成本選擇區域
- 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 在管理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。判斷元件與資源會執行較長期間 (針對承諾折扣)，還是動態短期執行 (針對 Spot 或隨需)。使用成本管理工具中的建議對工作負載執行分析，並且對這些建議套用商業規則，以達到高報酬。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 有多種定價模式，可讓您根據組織的需求和產品，以最經濟實惠的方式為資源付費。請與您的團隊合作，確認最適當的定價模式。定價模式常會包含多種選項的組合，這取決於您的可用性

隨需執行個體可讓您根據您所執行的執行個體按小時或秒 (最低 60 秒) 支付運算或資料庫容量的費用，而無需長期承諾或預付款。

Savings Plans 是一種彈性定價模式，您只需承諾一年或三年期的穩定使用量 (以每小時的金額計算)，即可以低價使用 Amazon EC2、Lambda 和 AWS Fargate (Fargate)。

Spot 執行個體是一種 Amazon EC2 定價機制，可讓您以折扣的每小時費率要求備用運算容量 (最多可折扣隨需價格的 90%)，且無需前期承諾。

預留執行個體可讓您藉由預付容量費用而享有最高 75% 的折扣。如需詳細資訊，請參閱透過保留達到最佳成本。

您可能會選擇為生產、品質和開發環境的相關資源納入 Savings Plan。或者，由於沙盒資源在需要時才會開啟，您可能會為該環境中的資源選擇隨需模式。請使用 Amazon Spot 執行個體降低 Amazon EC2 成本，或使用 Compute Savings Plans 降低 Amazon EC2、Fargate 和 Lambda 成本。AWS Cost Explorer 建議工具提供透過 Savings Plans 獲得承諾折扣的機會。

如果您過去曾購買 Amazon EC2 的預留執行個體，或已在您的組織內建立成本分配準則，您將可繼續使用 Amazon EC2 預留執行個體。但我們建議應擬定相關策略，在未來使用 Savings Plans 作為更具彈性的節省成本機制。您可以隨時重新整理 AWS Cost Management 中的 Savings Plans (SP) 建議，以重新產生新的 Savings Plans 建議。使用預留執行個體 (RI) 降低 Amazon RDS、Amazon Redshift、Amazon ElastiCache 和 Amazon OpenSearch Service 成本。有三個選項提供 Saving Plans 和預留執行個體：全額預付款、部分預付款和無預付款。使用 AWS Cost Explorer RI 和 SP 購買建議中提供的建議。

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。您可以將 Spot 執行個體用於各種不同的容錯和彈性應用程式。範例包括無狀態 Web 伺服器、API 端點、大數據和分析應用程式、容器化工作負載、CI/CD 與其他彈性工作負載。

分析您的 Amazon EC2 和 Amazon RDS 執行個體是否可在未使用時 (下班時間和週末) 關閉。相較於全年無休地使用，此方法可讓您降低成本達 70% 甚或更高。如果您有僅需在特定時間啟用的 Amazon Redshift 叢集，您可以暫停叢集，等稍後再繼續執行。Amazon Redshift 叢集或 Amazon EC2 和 Amazon RDS 執行個體停止時，運算計費也會隨之停止，而只會計算儲存費用。

請注意，隨需容量保留 (ODCR) 並非定價折扣。容量保留會依同等的隨需費率計費，無論您是否以預留容量執行執行個體。若需要為預計要執行的資源提供足夠的容量，就必須考量這些因素。ODCR 無須綁定長期承諾，您不再需要時即可取消，但也可適用 Savings Plans 或預留執行個體所提供的折扣。

實作步驟

- 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，分析您工作負載的彈性。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
- 審查現有的定價合約：審查目前基於長期需求的合約或承諾。分析您目前擁有的項目，以及有多少承諾正在使用中。運用既有的合約折扣或企業協議。企業協議可為客戶提供根據其需求自訂最適切合約的選項。針對長期承諾，請考慮將預留定價折扣、預留執行個體或 Savings Plans 用於特定執行個體類型、執行個體系列、AWS 區域 和可用區域。
- 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
- AWS 企業

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 根據成本選擇區域

此最佳實務已於 2023 年 7 月 13 日隨新版指引更新。

每個區域的資源定價可能不同。識別區域成本差異，並僅部署於具有較高成本的區域，以符合延遲、資料落地和資料主權要求。考量區域成本，有助於您為此工作負載支付最低的總價。

未建立此最佳實務時的曝險等級： 中

實作指引

此 AWS 雲端 基礎設施 是全球性的，託管於 全球多個地點，其建置基礎為 AWS 區域、可用區域、Local Zones、AWS Outposts 和 Wavelength Zones。區域是世界上的實體位置，每個區域各有一個地理區域，而 AWS 在其中有多個可用區域。可用區域是每個區域內的多個隔離位置，由一或多個分散的資料中心組成，各自有其備援電力、網路和連線能力。

每個 AWS 區域 各在當地市場條件之下運作，且各區域的資源定價因土地、光纖設施、電力和稅賦等因素而有所差異。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。使用 AWS 計算器 按位置類型 (區域、Wavelength Zone 和 Local Zone) 和區域搜尋服務，以預估您的工作負載在不同區域中的成本。

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。根據您的業務、資料隱私權、效能和安全要求，選取適當的地理位置。對於全球各地都有使用者的應用程式，請使用多個位置。

如果您在資料隱私權、安全和業務要求方面不受約束，請使用提供較低 AWS 服務價格的區域來部署工作負載。例如，如果您的預設區域是 ap-southeast-2 (雪梨)，且沒有使用其他區域方面的限制 (例如資料隱私權、安全)，則將非關鍵性 (開發和測試) Amazon EC2 執行個體部署在 north-east-1 (維吉尼亞北部) 區域，將可降低成本。

	合規	延遲	成本	服務/功能
區域 1	✓	15 毫秒	$$	✓
區域 2	✓	20 毫秒	$$$	X
區域 3	✓	80 毫秒	$	✓
區域 4	✓	15 毫秒	$$	✓
區域 5	✓	20 毫秒	$$$	X
區域 6	✓	15 毫秒	$	✓
區域 7	✓	80 毫秒	$	✓
區域 8	✓	15 毫秒	$	X

區域功能矩陣表

上方的矩陣表顯示區域 4 是這種情況下的最佳選擇，因為與其他區域相比，其延遲很低、服務可供使用，並且是成本最低的區域。

實作步驟

- 檢閱 AWS 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。
- 檢閱多區域部署的要求： 分析您的業務要求和義務 (資料隱私權、安全或效能)，確認是否有任何限制使您無法使用多個區域。如果沒有使用單一區域的限制，請使用多個區域。
- 分析所需的資料傳輸： 選取區域時請考量資料傳輸成本。將資料存放在接近客戶與資源之處。選取資料流動成本較低、且資料傳輸最少的 AWS 區域。根據資料傳輸的商業需求，您可以使用 Amazon CloudFront、AWS PrivateLink、 AWS Direct Connect和 AWS Virtual Private Network 降低網路成本、提升效能並增強安全性。

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

相關範例：

- 常見架構的資料傳輸成本概觀
- 全球部署的成本考量
- 為工作負載選取區域時應考慮的事項
- Well-Architected 實驗室：按區域限制服務用量 (Level 200)

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。

未建立此最佳實務時的風險暴露等級：中

實作指引

市場上有多種產品可以幫助您管理雲端環境的成本。它們在功能方面可能會有一些差異，而這取決於客戶要求，例如有些客戶專注於成本管控或成本可見性，其他客戶則專注於成本最佳化。有效成本最佳化和管控的一個關鍵因素是使用具有必要功能和合適定價模式的合適工具。這些產品具有不同的定價模式。有些產品會向您收取每月賬單的一定百分比，有些產品則收取所實現節省金額的百分比。理想情況下，請只為您需要的功能付費。

當您在雲端中使用第三方解決方案或服務時，定價結構務必要符合您想要的成果。定價應根據其提供的結果和價值進行擴展。例如，在會從節省的成本中提取一定比例的軟體中，節省的成本 (成果) 越多，收費就越高。會隨著開支增加而要支付更多費用的授權協議可能不會永遠對您的成本最佳化目標有利。但是，如果供應商能為您帳單的所有部分提供明確的效益，則此擴展費用可能是合理的。

例如，如果您使用其他無效益的服務，則會提供 Amazon EC2 建議並收取整個帳單一定比例的解決方案可能會變得更加昂貴。另一個範例是受管服務，其會依受管資源成本的一定百分比計費。較大的執行

個體大小不一定需要更多的管理工作，但收費會更高。請確認這些服務定價安排在其服務中包含成本最佳化計劃或功能，以提升效率。

客戶可能會發現市場上的這些產品更先進或更易於使用。您需要考慮這些產品的成本，並考慮長遠的潛在成本最佳化成果。

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 叢集。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)。

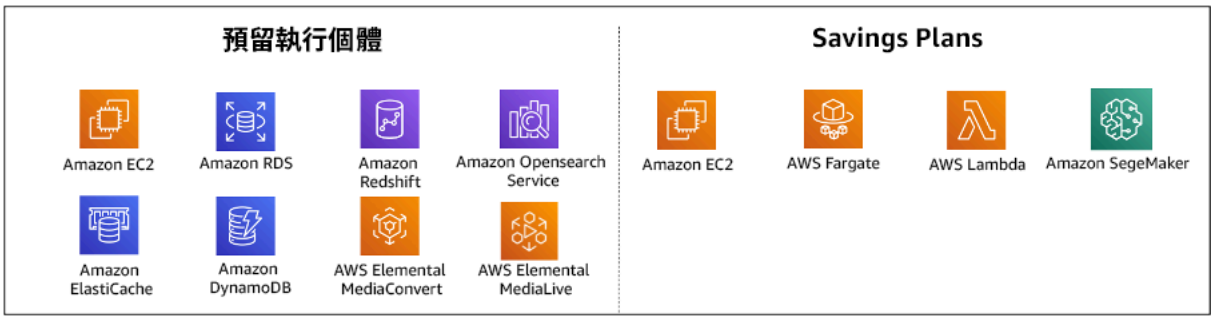
未建立此最佳實務時的風險暴露等級：低

實作指引

為了提高成本效率，AWS 會根據您過去的用量提供多個承諾建議。您可以使用這些建議來了解您可以節省的成本，以及如何使用承諾。您可以將這些服務作為隨需服務、Spot 服務，也可以承諾一定時間，並使用預留執行個體 (RI) 和 Savings Plans (SP) 降低隨需成本。您不僅需要了解每個工作負載元件和多項 AWS 服務，還需要了解這些服務的承諾折扣、購買選項和 Spot 執行個體，才能將工作負載最佳化。

考慮工作負載元件的要求，並了解這些服務的不同定價模式。定義這些元件的可用性要求。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。

例如，讓我們看看 AWS 上的這個 Web 應用程式架構。此範例工作負載包括多個 AWS 服務，例如 Amazon Route 53、AWS WAF、Amazon CloudFront、Amazon EC2 執行個體、Amazon RDS 執行個體、負載平衡器、Amazon S3 儲存和 Amazon Elastic File System (Amazon EFS)。您需要檢閱這些服務中的每一項，並透過不同的定價模式找出潛在的成本節省機會。其中有些服務可能符合 RI 或 SP 的資格，有些則可能只會隨需提供。如下圖所示，部分 AWS 服務可以使用 RI 或 SP 來重諾。



使用預留執行個體和 Savings Plans 所承諾的 AWS 服務

實作步驟

- 實作定價模式：使用分析結果購買 Savings Plans、預留執行個體或實作 Spot 執行個體。如果是第一次購買承諾，請選擇清單中的前五項或前十項建議，然後監控和分析未來一兩個月的結果。AWS Cost Management Console 會引導您完成該過程。從主控台檢閱 RI 或 SP 建議、自訂建議 (類型、付款和期限)，並檢閱每小時承諾 (例如每小時 20 美元)，然後加入到購物車。折扣會自動套用到符合資格的用量。定期購買少量承諾折扣 (例如每 2 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。最後，選取隨需 Amazon EC2 執行個體，並為其餘要求配置資源。
- 工作負載審查週期：實作會具體分析定價模式涵蓋範圍的工作負載審查週期。一旦工作負載達到所需的涵蓋範圍，請部分購買額外的承諾折扣 (每隔幾個月)，或隨著組織用量的變更進行購買。

資源

相關文件：

- 了解您的 Savings Plans 建議
- 存取預留執行個體的推薦
- 如何購買預留執行個體
- 執行個體購買選項

- Spot 執行個體
- 其他 AWS 服務的保留模式
- Savings Plans 支援的服務

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：

- 在購買 Savings Plans 前應考量哪些事項？
- 如何使用 Cost Explorer 分析開支和用量？

COST07-BP05 在管理帳戶層級執行定價模式分析

查看計費和成本管理工具，並檢視承諾和保留的建議折扣，在管理帳戶層級定期執行分析。

未建立此最佳實務時的曝險等級： 低

實作指引

執行定期成本建模可讓您有機會進行多個工作負載間的優化。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能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，在您的管理帳戶中尋找承諾折扣的機會。管理帳戶層級的建議在計算過程中會考量您的 AWS 組織中已啟用預留執行個體 (RI) 或 Savings Plans (SP) 折扣分享的帳戶。計算過程也會在折扣分享啟用時啟動，以推薦可盡量節省整體帳戶成本的承諾。

雖然在許多情況下，在管理帳戶層級購買可省下最多成本，但在某些情況下，您可以考慮在連結帳戶層級購買 SP，例如，您希望先將折扣套用至該連結帳戶中的用量時。成員帳戶建議會在個別帳戶層級上進行計算，以盡可能節省各個獨立帳戶的成本。如果您的帳戶同時擁有 RI 和 SP 承諾，則會按以下順序套用這些承諾：

1. 區域 RI
2. 標準 RI
3. 可轉換 RI

4. Instance Savings Plan
5. Compute Savings Plan

如果您在管理帳戶層級購買 SP，則將根據最高到最低的折扣百分比來套用節省的金額。管理帳戶層級的 SP 會查看所有連結帳戶，並以最高的折扣套用節省的金額。如果您希望限定節省金額的套用項目，您可以在連結的帳戶層級購買 Savings Plan，如此，每當該帳戶執行符合資格的運算服務時，就會先為該項目套用折扣。當帳戶未執行符合資格的運算服務時，折扣將會分享到相同管理帳戶下的其他連結帳戶。折扣分享預設為開啟，但可視需要關閉。

在合併帳單系列中，Savings Plans 會先套用至擁有者帳戶的用量，然後套用至其他帳戶的用量。只有在折扣分享啟用時，才會執行此模式。您的 Savings Plans 會先套用至您最高的節省金額百分比。如果有多種用法皆具有相同的節省百分比，Savings Plans 會套用至使用最低 Savings Plans 率的第一個用量。Savings Plans 會繼續套用，直到沒有剩餘用量或承諾用量耗盡。任何剩餘用量均按隨需費率收費。您可以隨時重新整理 AWS Cost Management 中的 Savings Plans 建議，以產生新的 Savings Plans 建議。

分析執行個體的彈性後，您可以採納建議的承諾。使用可能的不同資源選項分析工作負載的短期成本、分析 AWS 定價模型，並使其符合您的業務要求，以找出總體擁有成本和 成本最佳化 機會，進而建立成本模型。

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。請確實了解 Saving Plan 建議，並估計您的每月支出和每個月節省的成本。審查管理帳戶層級的建議；其計算過程中考量到您的 AWS 組織中已啟用 RI 或 Savings Plans 折扣分享，以盡可能節省帳戶成本的所有成員帳戶間的整體用量。您可以依照 Well-Architected 實驗室的指示，確定在所需的折扣與風險方面，採用了正確的建議。

資源

相關文件：

- AWS 定價的運作方式為何？
- 執行個體購買選項
- Saving Plan 概觀
- Saving Plan 建議
- 存取預留執行個體的推薦
- 了解您的 Saving Plans 建議

- Savings Plans 如何套用至您的 AWS 用量
- Saving Plans 與合併帳單
- 開啟共用的預留執行個體和 Savings Plans 折扣

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：

- AWS Well-Architected 實驗室：定價模式 (Level 200)
- AWS Well-Architected 實驗室：定價模式分析 (Level 200)
- 在購買 Savings Plan 前，我應考量哪些事項？
- 如何利用滾動 Savings Plans 降低承諾風險？
- 何時應使用 Spot 執行個體

COST 8.如何規劃資料傳輸費？

確實規劃和監控資料傳輸費，以便做出盡量減少成本的架構決策。小規模而有效的架構變更能夠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 選取元件以將資料傳輸成本最佳化
- 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。

未建立此最佳實務時的風險暴露等級：高

實作指引

在設計雲端解決方案時，由於習慣使用內部部署資料中心來設計架構或缺乏知識，通常會忽略掉資料傳輸費用。AWS 中的資料傳輸費用會由來源、目的地和流量的數量來決定。在設計階段考慮這些費用能

夠讓您省下成本。了解資料傳輸在工作負載中的發生位置、傳輸成本及其相關效益，對於準確估算總體擁有成本 (TCO) 來說非常重要。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。

您要為會在工作負載中傳輸資料的服務元件建模，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。

在為資料傳輸建模時，請考慮所擷取的資料量以及資料的來源。此外，也請考慮所處理的資料量以及需要的儲存或運算容量。在建模期間，請遵循工作負載架構的網路最佳實務，以將潛在的資料傳輸成本最佳化。

AWS Pricing Calculator 可以幫助您查看特定 AWS 服務的預估成本和預期的資料傳輸。如果您有已在執行的工作負載 (用於測試目的或在生產前環境中)，請使用 AWS Cost Explorer 或 AWS Cost and Usage Report (CUR) 來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

- 確定要求：來源和目的地之間所計劃資料傳輸的主要目標和業務要求是什麼？ 所預期的最終業務成果是什麼？ 收集業務要求並定義預期的成果。
- 確定來源和目的地：資料傳輸的資料來源和目的地是什麼 (例如在 AWS 區域 內、到 AWS 服務，或向外傳輸到網際網路)？
  - AWS 區域 內的資料傳輸
  - AWS 區域 之間的資料傳輸
  - 向外傳輸到網際網路的資料傳輸
- 確定資料分類：此資料傳輸的資料分類是什麼？ 這是什麼種類的資料？ 資料有多大？ 資料必須以何種頻率進行傳輸？ 資料敏感嗎？
- 確定要使用的 AWS 服務或工具：哪些 AWS 服務會用於此資料傳輸？ 是否可將已佈建的服務用於其他工作負載？
- 計算資料傳輸成本：使用先前建立的 AWS 定價資料傳輸模型來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。
- 將成本與成果連結：對於產生的每筆資料傳輸成本，請指定工作負載達到的成果。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，則可能是用於備援。
- 建立資料傳輸模型：在收集所有資訊後，為多個使用案例和不同工作負載建立概念性基礎資料傳輸模型。

資源

相關文件：

- AWS 快取解決方案
- AWS 定價
- Amazon EC2 定價
- Amazon VPC 定價
- 了解資料傳輸費用

相關影片：

- 監控並最佳化您的資料傳輸成本
- S3 Transfer Acceleration

相關範例：

- 常見架構的資料傳輸成本概觀
- AWS 網路方案指引

COST08-BP02 選取元件以將資料傳輸成本最佳化

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) 最佳化和多可用區域 (AZ) 組態等元件

未建立此最佳實務時的風險暴露等級：中

實作指引

資料傳輸建構可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。

將資料傳輸到 AWS 雲端 或於其中傳輸資料時，重要的是根據不同的使用案例來了解目的地、資料性質和可用的網路資源，以便選取合適的 AWS 服務來將資料傳輸最佳化。AWS 提供了一系列針對各種資料遷移要求量身打造的資料傳輸服務。根據組織內的業務需求，選取合適的資料儲存和資料傳輸選項。

在計劃或檢閱工作負載架構時，請考慮下列事項：

- 在 AWS 內使用 VPC 端點：VPC 端點可讓您在 VPC 與支援的 AWS 服務之間建立私人連線。這可讓您避免使用可能會產生資料傳輸成本的公用網際網路。
- 使用 NAT 閘道：使用 NAT 閘道，讓私有子網路中的執行個體可以連線到網際網路或 VPC 外的服務。檢查 NAT 閘道後方傳送最多流量的資源是否與 NAT 閘道位於相同的可用區域。如果沒有，請在與該資源相同的可用區域中建立新的 NAT 閘道，以降低跨 AZ 資料傳輸費用。
- 使用 AWS Direct Connect AWS Direct Connect 繞過公用網際網路，並在內部部署網路與 AWS 之間建立直接的私有連線。這可能會比透過網際網路傳輸大量資料更具成本效益和一致性。
- 避免跨區域界限傳輸資料：AWS 區域 之間的資料傳輸 (從某個區域到另一個區域) 通常會產生費用。請深思熟慮後再決定是否追求多區域路徑。如需詳細資訊，請參閱多區域案例。
- 監控資料傳輸：使用 Amazon CloudWatch 和 VPC Flow Logs 來擷取有關資料傳輸和網路用量的詳細資訊。分析 VPC 中擷取到的網路流量資訊，例如進出網路介面的 IP 地址或範圍。
- 分析您的網路用量：使用計量和報告工具 (例如 AWS Cost Explorer、CUDOS 儀表板或 CloudWatch) 以了解工作負載的資料傳輸成本。

實作步驟

- 選取用於資料傳輸的元件：使用 COST08-BP01 執行資料傳輸建模 中所說明的資料傳輸模型時，請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 (或降低其成本)。

資源

相關的最佳實務：

- COST08-BP01 執行資料傳輸建模
- COST08-BP03 實作可降低資料傳輸成本的服務

相關文件：

- 雲端資料遷移
- AWS 快取解決方案
- 使用 Amazon CloudFront 更快地交付內容

相關範例：

- 常見架構的資料傳輸成本概觀
- AWS 網路最佳化要訣
- 使用 Apache Parquet 格式的 VPC 流程日誌針對網路分析最佳化效能和降低成本

COST08-BP03 實作可降低資料傳輸成本的服務

實作服務以減少資料傳輸。例如，使用邊緣節點或內容交付網路 (CDN) 將內容提供給終端使用者、在應用程式伺服器或資料庫前面建置快取層，以及使用專用網路連線而非 VPN 來連線至雲端。

未建立此最佳實務時的曝險等級： 中

實作指引

有許多 AWS 服務可以協助您最佳化網路資料傳輸用量。根據您的工作負載元件、類型和雲端架構，這些服務可以協助您在雲端上壓縮、快取、共用和分配流量。

- 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 Well-Architected 安全節省搭售方案 可以在您計劃隨著時間的推移增加使用量，幫助您節省高達 30% 的 CloudFront 使用率。
- 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，因為它提供簡便的連線，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 成本。 閘道 VPC 端點 不收取小時費用，且支援 Amazon S3 和 Amazon DynamoDB。 界面 VPC 端點 由 AWS PrivateLink 提供，收取每小時費用和每 GB 使用費。
- NAT 閘道的 提供內建擴展和管理功能，與獨立 NAT 執行個體相比，成本更低。將 NAT 閘道放置在與高流量執行個體相同的可用區域中，並考慮為需要存取 Amazon DynamoDB 或 Amazon S3 的執行個體使用 VPC 端點，來降低資料傳輸和處理成本。
- 使用 AWS Snow Family 裝置，其中的運算資源可以用來收集與處理在邊緣 AWS Snow Family 裝置上的資料 (Snowcone、 Snowball 和 Snowmobile) 讓您能夠以具成本效益的方式將數 PB 的資料離線移至 AWS 雲端。

實作步驟

- 實作服務： 根據您的服務選擇適用的 AWS 網路服務、使用資料傳輸建模的工作負載類型，以及檢閱 VPC Flow Logs。查看成本最高和磁碟區流量最大的情況。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
- AWS Snow Family
- Amazon CloudFront 安全節省搭售方案

相關影片：

- 監控並最佳化您的資料傳輸成本
- AWS 成本最佳化系列：CloudFront
- 如何降低 NAT 閘道的資料傳輸費用？

相關範例：

- 如何退款共享服務：AWS Transit Gateway 範例
- 使用 Athena 查詢和 QuickSight，從成本和用量報告深入了解 AWS 資料傳輸詳細資訊
- 常見架構的資料傳輸成本概觀
- 使用 AWS Cost Explorer 分析資料傳輸成本
- 利用 Amazon CloudFront 功能，針對您的 AWS 架構進行成本最佳化
- 如何降低 NAT 閘道的資料傳輸費用？

管理需求與供應資源

問題

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，確認您購買的每一個項目都用到，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。

未建立此最佳實務時的曝險等級： 高

實作指引

要分析工作負載對雲端運算的需求，就必須了解雲端環境中啟動的運算工作模式和特性。這類分析可協助使用者優化資源配置、管理成本，並確保效能符合所需等級。

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。針對足夠長的時間執行分析，以納入任何季節變化，例如月底處理或節假日尖峰。

分析工作應反映實作擴展的潛在效益。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。

以下是執行雲端運算的工作負載需求分析時需要考慮的一些關鍵事項：

1. 資源使用和效能指標：分析 AWS 資源在一段時間內的使用情形。確認尖峰和離峰使用模式，以最佳化資源配置和擴展策略。監控效能指標，例如回應時間、延遲、輸送量和錯誤率。這些指標有助於評估雲端基礎架構的整體運作狀態和效率。

2. 使用者和應用程式擴展行為：了解使用者行為及其對工作負載需求的影響。檢查使用者流量的模式，有助於提高交付內容的完整性和應用程式的回應能力。分析工作負載如何隨著需求增加而擴展。判斷是否已正確、有效地設定自動擴展參數，以處理負載波動。
3. 工作負載類型：識別出在雲端中執行的不同工作負載類型，例如批次處理、即時資料處理、Web 應用程式、資料庫或機器學習。每種工作負載類型可能有不同的資源需求和效能資料。
4. 服務水準協議 (SLA)：將實際效能與 SLA 進行比較，以確保合規性並找出需要改進的部分。

您可以使用 Amazon CloudWatch 收集和追蹤指標、監控日誌檔、設定警示，以及自動對 AWS 資源的變更做出反應。您也可以利用 Amazon CloudWatch 來全面了解整個系統的資源使用率、應用程式效能和運作狀態。

透過 AWS Trusted Advisor，您可以根據最佳實務佈建資源，以改善系統效能和可靠性、提高安全性，並尋找節省成本的機會。您也可以關閉非生產執行個體，並使用 Amazon CloudWatch 和 Auto Scaling 來因應需求增加或減少。

最後，您可以使用 AWS Cost Explorer 或者 Amazon QuickSight 搭配 AWS Cost and Usage Report CUR 檔案或應用程式日誌，以執行工作負載需求的進階分析。

整體而言，全面的工作負載需求分析可讓組織在資源佈建、擴展和最佳化方面做出明智決策，進而提高效能、成本效益和使用者滿意度。

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。使用 Amazon CloudWatch、日誌檔和監控資料來深入了解工作負載的使用情況。分析工作負載的完整週期，並收集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。
- 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：

- Amazon CloudWatch

- AWS Trusted Advisor
- AWS X-Ray
- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon SQS 入門
- AWS Cost Explorer
- Amazon QuickSight

相關影片：

相關範例：

- 監控、追蹤和分析，以達到成本最佳化
- 搜尋和分析 CloudWatch 中的日誌

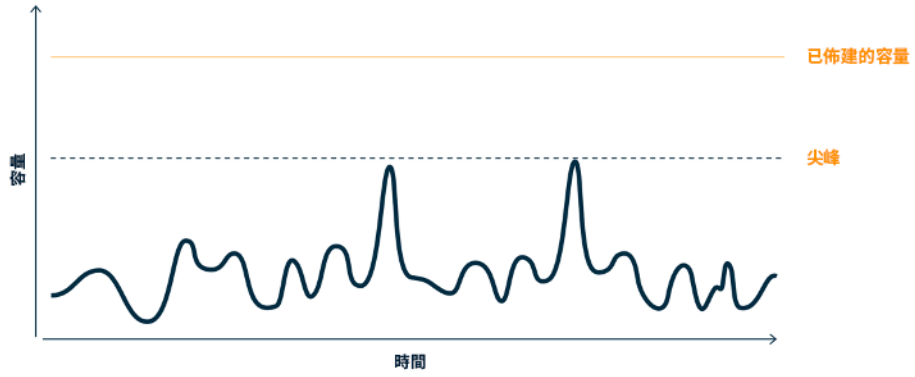
COST09-BP02 實作緩衝或調節機制來管理需求

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。確認調節和緩衝機制經過設計，以便讓用戶端在所需時間內收到回應。

未建立此最佳實務時的風險暴露等級：中

實作指引

在雲端運算中實作緩衝或調節機制至關重要，如此才能管理需求並降低工作負載所需的佈建容量。為了獲得最佳效能，請務必評估總需求，包括峰值、請求變更速度以及必要的回應時間。當用戶端能夠重新發送他們的請求時，套用限流就變得很實用。相反地，對於缺少重試功能的用戶端，最理想的方法是實作緩衝解決方案。這類緩衝機制簡化了請求的湧入作業，並且會將有不同操作速度之應用程式的互動最佳化。



需求曲線圖，內含兩個需要大量已佈建容量的相異尖峰

假設某個工作負載的需求曲線如上圖所示。此工作負載有兩個尖峰，為了處理這些尖峰，已佈建了資源容量 (以橙色線顯示)。用於此工作負載的資源和能源並非由需求曲線底下的區域表示，而是已佈建的容量底下的區域，因為這兩個尖峰必須用已佈建的容量處理。使工作負載需求曲線扁平化，有助於減少工作負載所需的已佈建容量，以及降低對環境造成的影響。若要消除尖峰時段，請考慮實作限流或緩衝解決方案。

為了深入了解，讓我們探索一下限流和緩衝機制。

限流：如果需求來源具有重試功能，則您可以實作限流。限流會告知來源，如果目前無法服務請求，則應稍後再試。來源會等待一段時間，然後重試請求。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 來實作限流。

緩衝為主：緩衝為主的方法會使用生產者 (會將訊息傳送到佇列的元件)、取用者 (會從佇列接收訊息的元件) 和佇列 (保存訊息) 來儲存訊息。消費者可讀取訊息並進行處理，允許以符合取用者業務要求的速度運作訊息。透過使用緩衝為主的方法，生產者的訊息會儲存在佇列或串流中，隨時可供取用者以符合其操作需求的速度來存取。

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) 是受管服務，會提供可讓單一取用者讀取個別訊息的佇列。Amazon Kinesis 會提供可讓許多取用者讀取相同訊息的串流。

緩衝和限流可透過修改工作負載的需求來消除任何尖峰時段。當用戶端會重試動作時請使用限流，並使用緩衝機制來保存請求以供稍後處理。使用緩衝為主的方法時，請將工作負載建構為可在所需的時間內為請求提供服務，並確認您能夠處理重複的工作請求。分析整體需求、變更率及所需的回應時間，以適當調整所需的調節或緩衝區大小。

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，則需要實作緩衝機制。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。
- 實作緩衝或調節機制：在工作負載中實作緩衝或調節機制。Amazon Simple Queue Service (Amazon SQS) 等佇列可為工作負載元件提供緩衝機制。Amazon API Gateway 可為工作負載元件提供限流。

資源

相關的最佳實務：

- SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化
- REL05-BP02 限流請求

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon API Gateway
- Amazon Simple Queue Service
- Amazon SQS 入門
- Amazon Kinesis

相關影片：

- 為分散式應用程式選擇適當的傳訊服務

相關範例：

- 管理和監控工作負載中的 API 調節
- 使用 API Gateway 大規模地限流分級的多租用戶 REST API
- 使用 Amazon API Gateway 在多租用戶 Amazon EKS SaaS 解決方案中啟用分級和限流
- 使用佇列和訊息進行應用程式整合

COST09-BP03 動態提供資源

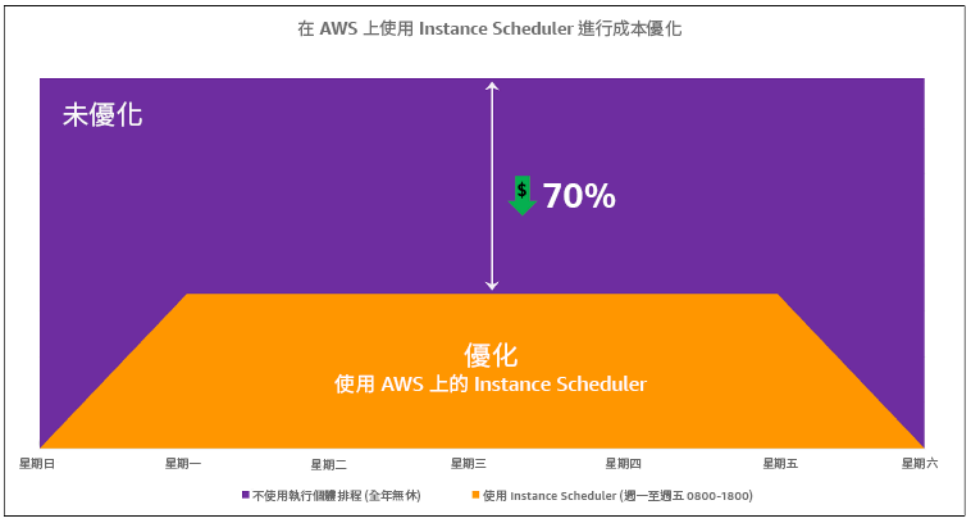
資源是按計畫進行佈建。這可以是以需求為基礎 (例如，透過自動調整規模)，或是以時間為基礎，其中需求可預測，並且根據時間提供資源。這些方法可盡量減少過度佈建或佈建不足的數量。

未建立此最佳實務時的曝險等級： 低

實作指引

AWS 客戶有數種方法可以增加應用程式的可用資源並提供資源，以滿足需求。其中一個選項是使用 AWS Instance Scheduler，以自動執行 Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon Relational Database Service (Amazon RDS) 執行個體的啟動及停止。另一個選項是使用 AWS Auto Scaling，這可讓您根據應用程式或服務的需求自動擴展運算資源。根據需求提供資源可讓您僅為自己使用的資源付費，以及在需要時啟動資源，並在不需要資源時將其終止，藉以降低成本。

AWS Instance Scheduler 可讓您將 Amazon EC2 和 Amazon RDS 執行個體設定為在已定義的時間停止及啟動，以便在一致的時間模式下達到相同資源的需求，例如，使用者在每天早上八點存取 Amazon EC2 執行個體，而晚上六點後則不需存取。此解決方案可停止非使用中的資源，並在需要時才加以啟動，藉以降低營運成本。



使用 AWS Instance Scheduler 優化成本。

您也可以使用 AWS Systems Manager 快速設定，透過簡單的使用者介面 (UI) 輕鬆設定跨帳戶和區域的 Amazon EC2 執行個體排程。您可以使用 AWS Instance Scheduler 來排程 Amazon EC2 或 Amazon RDS 執行個體，也可以停止和啟動現有的執行個體。不過，您無法停止及啟動屬於 Auto Scaling 群組 (ASG) 或管理 Amazon Redshift 或 Amazon OpenSearch Service 等服務的執行個體。Auto Scaling 群組對於群組內的執行個體有其本身的排程，據以建立這些執行個體。

AWS Auto Scaling 可協助您調整容量，盡可能以最低的成本維持穩定、可預測的效能，以因應持續變動的需求。這是全受管的免費服務，可與 Amazon EC2 執行個體和 Spot 機群、Amazon ECS、Amazon DynamoDB 與 Amazon Aurora 整合，以擴展應用程式的容量。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。

有多個擴展選項可用來擴展您的 Auto Scaling 群組：

- 一律保持目前的執行個體層級
- 手動擴展
- 根據排程進行擴展
- 根據需求進行擴展
- 使用預測擴展

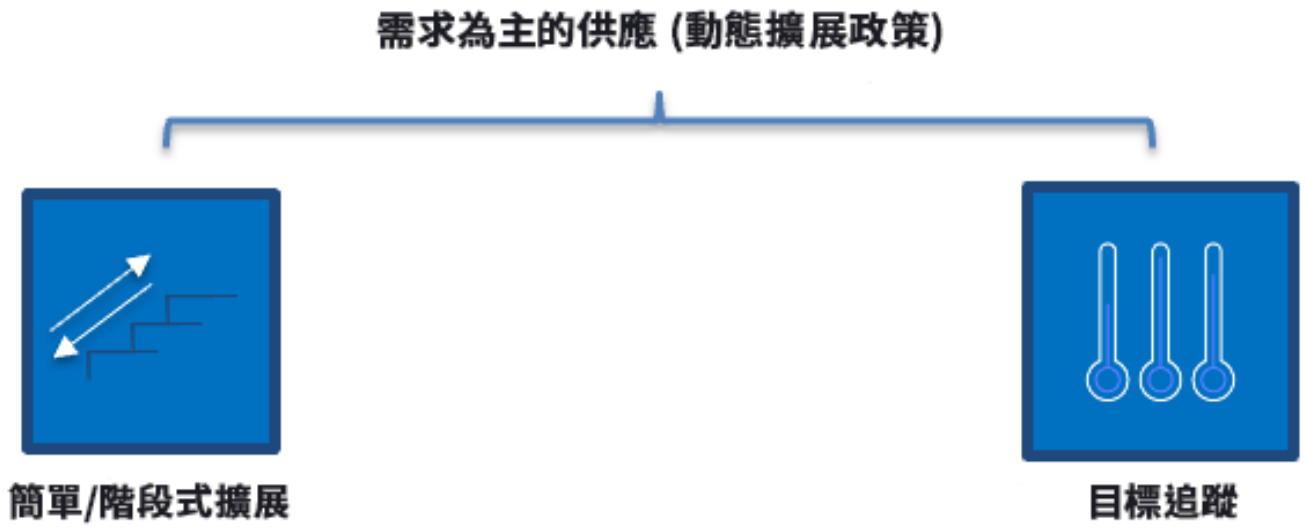
Auto Scaling 政策不同，可分類為動態和排程擴展政策。動態政策是手動或動態擴展，屬於排程或預測擴展。您可以使用擴展政策來進行動態、排程和預測擴展。您也可以使用 Amazon CloudWatch 的指標和警示，來觸發工作負載的擴展事件。我們建議您使用 啟動範本，這可讓您存取最新的功能和改進內容。當您使用啟動組態時，並非所有的 Auto Scaling 功能都可供使用。例如，您無法建立同時啟動 Spot 和隨需執行個體，或指定多個執行個體類型的 Auto Scaling 群組。您必須使用啟動範本來設定這些功能。使用啟動範本時，建議您對每個範本進行版本控制。使用啟動範本的版本控制，可以建立完整參數集的子集。然後，您可加以重複使用，以建立相同啟動範本的其他版本。

您可以使用 AWS Auto Scaling，或使用 AWS API 或 SDK 在您的程式碼中納入擴展。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，且變更的執行速度更快。這也可讓您隨時依據需求做出相應的工作負載資源配置。為了遵循此最佳實務，並且為組織動態提供資源，您應了解 AWS 雲端 中的水平和垂直擴展，以及在 Amazon EC2 執行個體中執行的應用程式有何性質。建議讓您的雲端財務管理團隊與技術團隊相互合作，以遵循此最佳實務。

Elastic Load Balancing (Elastic Load Balancing) 可將需求分散到多個資源以協助您進行擴展。藉由使用 ASG 和 Elastic Load Balancing，您可以用最佳方式路由流量以管理傳入請求，讓 Auto Scaling 群組中沒有任何執行個體不堪負荷。請求會以循環方式散佈在目標群組的所有目標之間，而不考量容量或使用率。

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，以及 Elastic Load Balancing 觀察到的請求與回應延遲。若可行的話，您應該使用可指示客戶體驗的指標，這通常是自訂指標，可能源自您工作負載內的應用程式程式碼。為了在本文件中詳細說明如何動態滿足需求，我們將 Auto Scaling 分類為需求為主和時間為主的供應模式，並深入探討這兩種模式。

需求為主的供應： 依賴幾近即時的需求狀態，充分利用雲端的彈性來供應資源，以滿足不斷變化的需求。對於需求為主的供應，請使用 API 或服務功能，以程式設計方式更動架構中的雲端資源量。這樣可讓您增減架構中元件的規模，在需求激增時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。

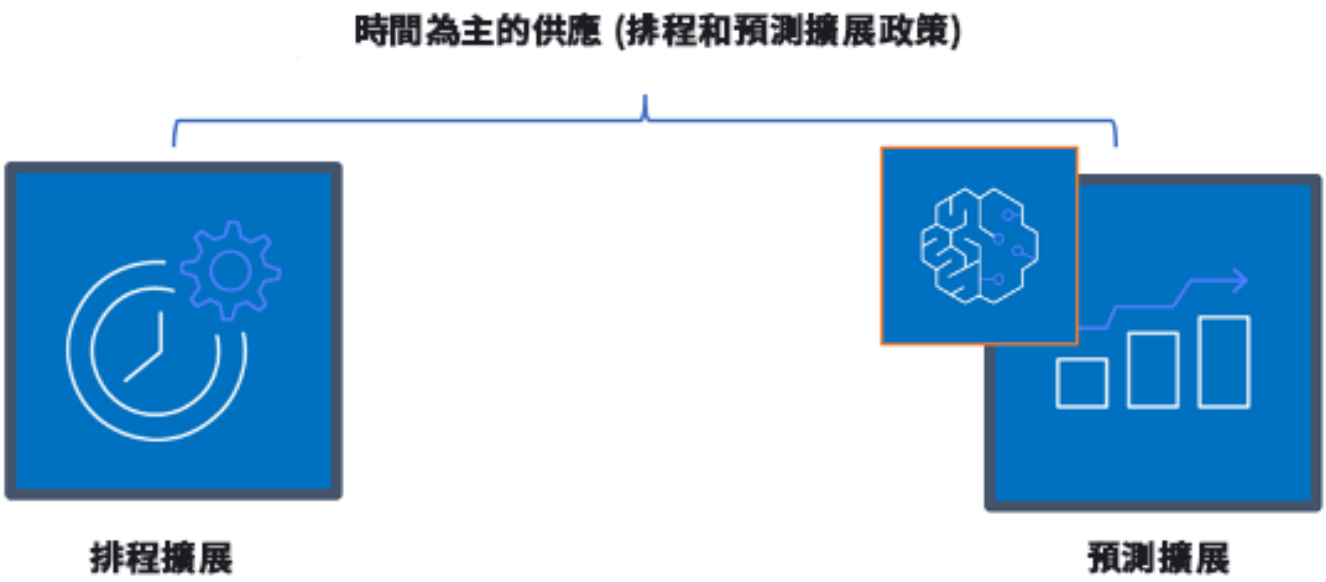


需求為主的動態擴展政策

- 簡單/階段式擴展： 根據客戶手動定義的步驟，監控指標及新增/移除執行個體。
- 目標追蹤： 類似恆溫器的控制機制，可自動新增或移除執行個體，以在客戶定義的目標上維護指標。

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。

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。採用時間為主的方法，您可在忙碌期提供更多資源或增加容量。



時間為主的擴展政策

您可以使用排程或預測自動擴展來實作時間為主的方法。可排定工作負載於定義的時間橫向擴展或縮減 (例如在營業時段開始時)，以便在使用者到來或需求增加時有資源可用。預測擴展會使用模式進行橫向擴展，而排程的擴展則使用先定義的時間進行橫向擴展。您也可以使用 屬性型執行個體類型選取 (ABS) 策略 (在 Auto Scaling 群組中)，以透過一組屬性 (例如 vCPU、記憶體和儲存) 來表達您的執行個體要求。這也可讓您自動使用新發行的新世代執行個體類型，並使用 Amazon EC2 Spot 執行個體來存取更大範圍的容量。Amazon EC2 機群和 Amazon EC2 Auto Scaling 會選取和啟動符合指定屬性的執行個體，您不必再手動挑選執行個體類型。

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 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

- 設定排程擴展： 針對可預測的需求變更，以時間為主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程的擴展。若要設定以時間為主的排程，您可以根據預期或可預測的負載變更，事先使用排程擴展的預測擴展來增加 Auto Scaling 群組中的 Amazon EC2 執行個體數目。
- 設定預測擴展： 預測擴展可讓您事先在 Auto Scaling 群組中增加每日和每週流量模式的 Amazon EC2 執行個體數目。如果您有定期流量尖峰和啟動耗時的應用程式，則應考慮使用預測擴展。預測擴展可在預估的負載之前初始化容量，協助您以優於單純動態擴展 (本質上是被動的) 的速度進行擴展。例如，如果使用者在營業時間開始時開始使用您的工作負載，且在營業時間結束後不使用，則預測擴展可在營業時間之前新增容量，以消除動態擴展為了回應變動的流量而產生的延遲。
- 設定動態自動擴展： 若要根據作用中的工作負載指標來設定擴展，請使用 Auto Scaling。使用分析和設定 Auto Scaling 以在正確的資源層級上啟動，並確認工作負載在所需的時間內擴展。您可以啟動並自動擴展單一 Auto Scaling 群組內的一組隨需執行個體和 Spot 執行個體。除了獲得使用 Spot 執行個體的折扣外，您還可以使用預留執行個體或 Savings Plan 來獲得常規隨需執行個體定價的折扣費率。將這些因素全部結合在一起，可協助您將 Amazon EC2 執行個體所能節省的成本優化，並確定應用程式所需的規模和效能。

資源

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- 擴展 Auto Scaling 群組的大小
- Amazon EC2 Auto Scaling 入門
- Amazon SQS 入門
- Amazon EC2 Auto Scaling 的排程擴展
- Amazon EC2 Auto Scaling 的預測擴展

相關影片：

- Auto Scaling 的目標追蹤擴展政策
- AWS Instance Scheduler

相關範例：

- Amazon EC2 機群的 Auto Scaling 屬性型執行個體類型選取
- 使用排程的擴展將 Amazon Elastic Container Service 的成本優化
- Amazon EC2 Auto Scaling 的預測擴展
- 如何搭配使用 Instance Scheduler 與 AWS CloudFormation 來排程 Amazon EC2 執行個體？

隨時間優化

問題

- COST 10.如何評估新服務？
- COST 11.如何評估工作的成本？

COST 10.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，以確認其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帳單 10% 的工作負載每季或每六個月審查一次，而低於 10% 的工作負載則每年審查一次。

未建立此最佳實務時的風險暴露等級：高

實作指引

為了擁有最符合成本效益的工作負載，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。若要實現較低的整體成本，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，您應更頻繁且更徹底地審查佔整體支出 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。

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，進而優化您的工作負載。如同效能要素審查和可靠性要素審查程序，請進行優化和改進活動與問題修復的識別、驗證及優先順序排定，並將其併入您的積存中。

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。配置時間和資源給持續性改進與審查頻率，以改進工作負載的效率和優化。這結合了許多因素，可能隨著組織內的工作負載而異，也可能隨著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：定義耗費於審查工作負載或工作負載元件的工作量。與審查頻率類似，這需在多個因素之間取得平衡。評估改進機會並制定其優先順序，以將精力集中在可以帶來最大收益的機會上，同時預估這些活動需要多少工作量。如果預期成果未能達到目標，且所需的工作量成本較高，請使用替代行動方案重複進行。您的審查程序應包含專用的時間和資源，用於持續的漸進式改善。例如，您可以決定花費一週分析來資料庫元件、一週分析運算資源，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

相關範例：

- AWS 支援主動服務
- SAP 工作負載的定期工作負載審查

COST10-BP02 定期審查和分析此工作負載

現有的工作負載會根據每個定義的程序定期接受審查，以確認是否可採用新服務、是否可取代現有服務、或是否可重新建構工作負載。

未建立此最佳實務時的風險暴露等級：中

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。AWS 最新消息會詳述 AWS 執行這項工作的情形，並且在 AWS 服務、功能和區域性擴充公告發行時提供其快速概覽。您可以深入探討已公告推出的項目，並將其用來審查和分析現有的工作負載。若要取得新的 AWS 服務和功能帶來的效益，您必須對工作負載進行審查，並視需要實作新的服務和功能。這表示您可能需要取代用於工作負載的現有服務，或將工作負載現代化，以採用這些新的 AWS 服務。例如，您可以審查工作負載，並使用 Amazon Simple Email Service 取代傳訊元件。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。

若要分析工作負載並凸顯潛在機會，您不僅應考慮使用新服務，也應使用新方法來建置解決方案。觀看 AWS 上的 This is My Architecture 影片以了解其他客戶的架構設計，及其面臨的挑戰和解決方案。查看 All-In 系列，以了解 AWS 服務的實際應用和客戶案例。您也可以觀看 Back to Basics 影片系列，其中包含對基本雲端架構模式的最佳實務所做的說明、探討和解析。另一個來源是 How to Build This 影片，其用意是要協助人們大致了解如何利用 AWS 服務 (MVP) 將其最低可行產品 (MVP) 推出上市。全球各地的建置人員只要有強烈意願想獲得經驗豐富的 AWS 解決方案架構師提供的架構指引，都可循此途徑。最後，您可以檢閱入門資源素材，其中包含逐步教學課程。

執行審查程序之前，請遵循您的企業在工作負載、安全和資料隱私權等方面的要求，以期在執行您同意的審查程序時，能夠採用特定的服務或區域和效能要求。

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
- AWS 最新消息
- AWS 文件
- AWS 入門
- AWS 一般資源

相關影片：

- AWS - This is My Architecture
- AWS - Back to Basics
- AWS - All-In 系列
- How to Build This

COST 11.如何評估工作的成本？

最佳實務

- COST11-BP01 執行操作自動化

COST11-BP01 執行操作自動化

評估雲端上的營運成本，專注於透過自動化，量化在管理任務、部署、減少人為錯誤風險、合規性和其他作業方面所省下的時間和精力。評估營運所需的時間和相關成本，並實行管理任務自動化，以盡可能地減少手動工作。

未建立此最佳實務時的風險暴露等級：低

實作指引

自動化操作可減少手動作業的頻率、提升效率，且客戶可在部署、管理或操作工作負載時享有一致且可靠的體驗。您可以從手動操作任務中釋出基礎設施資源，並將其用於價值更高的任務與創新，進而提高商業價值。企業需要以經過實證和測試的方式來管理其雲端中的工作負載。該解決方案必須安全、快速、符合成本效益，且具有最低的風險和最高的可靠性。

首先，請查看整體營運成本，再根據所需的工作量安排營運活動的優先順序。例如，在雲端中部署新資源、對現有資源進行優化變更，或實作所需的組態，分別需要多久的時間？ 將營運和管理成本納入考量，以查看人為活動的整體成本。排定管理任務自動化的優先順序，以減少人力付出。

審查工作應反映潛在的效益。例如，檢查手動和自動執行任務花費的時間。優先將重複性、高價值、耗時和複雜的活動自動化。具高價值或風險較高會發生人為錯誤的活動，通常適合作為自動化的起點，因為這類風險常伴隨著我們不樂見的額外營運成本 (例如營運團隊的加班費)。

使用自動化工具 (例如 AWS Systems Manager 或 AWS Config) 簡化作業、合規性、監控、生命週期和終止程序。您可以使用 AWS 服務、工具和第三方產品，自訂您要實作的自動化作業以滿足您的特定要求。下表顯示您可以透過 AWS 服務取得哪些核心操作功能與能力，以自動執行管理與操作：

- AWS Audit Manager：持續稽核 AWS 用量以簡化風險和合規的評估
- AWS Backup：集中管理及自動執行資料保護。
- AWS Config：設定運算資源、評估、稽核及衡量組態和資源清查。
- AWS CloudFormation：使用基礎設施即程式碼啟動高度可用的資源。
- AWS CloudTrail：IT 變更管理、合規和控制。
- Amazon EventBridge：排程事件並觸發 AWS Lambda 以採取行動。
- AWS Lambda：透過事件觸發或使用 AWS EventBridge 按固定排程來執行重複性程序，藉以將這些程序自動化。
- AWSSystems Manager：啟動和停止工作負載、修補作業系統、自動進行設定和持續管理。
- AWS Step Functions：排程任務並自動執行工作流程。
- AWS Service Catalog：符合規範並受到控制的範本取用和基礎設施即程式碼。

如果您想要使用 AWS 產品和服務立即採用自動化，且您的組織不具備相關技能，請聯繫 AWS Managed Services (AMS)、AWS Professional Services 或 AWS 合作夥伴，以提升您採用自動化的能力，並改善您在雲端中的營運效能。

AWS Managed Services (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。

AWS Professional Services 也可協助您透過 AWS 達成預期的商業成果和操作的自動化。這些實務準則可協助客戶部署自動化、穩健而靈活的 IT 營運，以及已針對雲端進行最佳化的管控能力。如需詳細的監控範例和建議的最佳實務，請參閱營運卓越支柱白皮書。

實作步驟

- 建置一次即可多次部署：使用 CloudFormation、AWS SDK 或 AWS CLI 之類的基礎設施即程式碼部署一次，並針對相似的環境或災難復原案例使用多次。在部署時加上標籤以追蹤您的使用量，如其他最佳實務所定義。使用 AWS Launch Wizard 可縮短部署許多常用企業工作負載所需的時間。AWS Launch Wizard 會引導您依據 AWS 最佳實務完成企業工作負載的大小調整、設定和部署。您也可以使用 Service Catalog，這有助於建立及管理基礎設施即程式碼核准用於 AWS 的範本，讓任何人都可探索已核准的自助服務雲端資源。
- 自動化持續合規：考慮根據預先定義的標準自動評估和矯正記錄的組態。當您結合 AWS Organizations 和 AWS Config 和 AWS CloudFormation 的功能時，您可以有效率地大規模管理和自動執行數百個成員帳戶的組態合規性。您可以檢閱組態變更和 AWS 資源之間的關係，並深入了解資源組態的歷史記錄。
- 自動化監控工作 AWS 提供各種工具，可用來監控服務。您可以將這些工具設定為自動執行監控工作。建立並實作監控計畫，從工作負載中的所有零件收集監控資料，以便在發生多點故障時，讓您能更輕鬆地除錯。例如，您可以使用自動監控工具觀察 Amazon EC2，該工具會在系統狀態檢查、執行個體狀態檢查和 Amazon CloudWatch 警示發生問題時向您報告。
- 自動化維護和操作：自動執行例行性操作而無需人工介入。使用 AWS 服務和工具時，您可以選擇要實作及自訂哪些 AWS 自動化以滿足您的特定要求。例如，使用 EC2 Image Builder 來建置、測試及部署虛擬機器和容器映像，以用於 AWS 或內部部署，或使用 AWS SSM 修補 EC2 執行個體。若您所需的動作無法以 AWS 服務完成，或您需要使用篩選資源的複雜動作，請使用 AWS Command Line Interface (AWS CLI) 或 AWS SDK 工具將操作自動化。AWS CLI 可讓您透過指令碼自動執行控制及管理 AWS 服務的完整程序，而無需使用 AWS Management Console。選取您慣用的AWS SDK 與 AWS 服務互動。如需其他程式碼範例，請參閱 AWS SDK 程式碼範例儲存庫。
- 利用自動化建立持續的生命週期：建立和保留成熟的生命週期政策非常重要，這不僅是為了法規或備援，也是為了實現成本最佳化。您可以使用 AWS Backup 集中管理和自動化資料存放區的資料保護，例如儲存貯體、磁碟區、資料庫和檔案系統。您也可以使用 Amazon Data Lifecycle Manager 自動建立、保留和刪除 EBS 快照與 EBS 支援的 AMI。
- 刪除不必要的資源：在沙盒或開發 AWS 帳戶 中累積未使用的資源，是相當常見的做法。在正常開發週期，開發人員會建立和實驗各種服務和資源，然後當不再需要那些資源時，也不會將其刪除。未使用的資源可能會對組織造成不必要且有時過高的成本。刪除這些資源可以降低運作這些環境的成本。如果不確定，請確保您不需要資料，或已備份資料。您可以用 AWS CloudFormation 來清理已部署的堆疊，這會自動刪除範本中定義的大多數資源。或者，您可以使用 aws-nuke 等工具來建立刪除 AWS 資源的自動化作業。

資源

相關文件：

- 將 AWS 雲端 中的運作現代化
- 用於自動化的 AWS 服務
- 基礎設施和自動化
- AWS Systems Manager 自動化
- 自動化和手動監控
- 用於 SAP 管理和運作的 AWS 自動化
- AWS Managed Services
- AWS Professional Services

相關影片：

- 在 AWS 中大規模自動化持續合規
- AWS Backup 示範：跨帳戶和跨區域備份
- 為您的 Amazon EC2 執行個體修補

相關範例：

- 重塑自動化操作 (第 I 部分)
- 重塑自動化操作 (第 II 部分)
- 使用 aws-nuke 自動刪除 AWS 資源
- 使用 AWS Config 和 AWS SSM 刪除未曾使用的 Amazon EBS 磁碟區
- 在 AWS 中大規模自動化持續合規
- AWS Lambda 的 IT 自動化

永續性

在建置雲端工作負載時，永續性實務是關於了解所使用服務的影響、量化整體工作負載生命週期的影響，以及應用設計原則和最佳實務來減少這些影響。您可以在下列白皮書中找到規範指引：永續性支柱白皮書。

最佳實務領域

- 區域選擇
- 因應需求
- 軟體和架構
- 資料
- 硬體和服務
- 程序和文化

區域選擇

問題

- SUS 1 如何為您的工作負載選取區域？

SUS 1 如何為您的工作負載選取區域？

您為工作負載選擇的區域會極大程度地影響其 KPI，包括效能、成本和碳足跡。若要有效改進這些 KPI，請根據您的業務要求和永續性目標，選擇工作負載的區域。

最佳實務

- SUS01-BP01 根據您的業務要求和永續性發展目標選擇區域

SUS01-BP01 根據您的業務要求和永續性發展目標選擇區域

根據您的業務要求和永續性發展目標選擇工作負載的區域，以將其 KPI 最佳化，包括效能、成本和碳足跡。

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。
- 您可以將所有工作負載資源合併到單一地理位置。

建立此最佳實務的優勢：將工作負載放在 Amazon 可再生能源專案附近或所公佈的碳強度較低的區域附近，有助於降低雲端工作負載的碳足跡。

未建立此最佳實務時的風險暴露等級：中

實作指引

AWS 雲端 是會持續擴張的區域和連接點 (POP) 網路，並透過全球網路基礎設施彼此連結起來。您為工作負載選擇的區域會極大程度地影響其 KPI，包括效能、成本和碳足跡。若要有效改進這些 KPI，請根據您的業務要求和永續性發展目標，選擇工作負載的區域。

實作步驟

- 請遵循這些步驟，根據您的業務要求 (包括合規、可用功能、成本和延遲) 評估工作負載的可能區域，並將這些區域列入候選清單：
  - 根據必須遵守的當地法規，確認這些區域符合規範。
  - 使用 AWS 區域服務清單來檢查區域是否有您執行工作負載時所需的服務和功能。
  - 使用 AWS Pricing Calculator 計算工作負載在每個區域的成本。
  - 測試終端使用者所在位置和每個 AWS 區域 之間的網路延遲。
- 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。
  - 識別相關的永續性指導方針，根據溫室氣體協定 (市場型和位置型方法) 來追蹤和比較逐年的碳排放。
  - 根據您用來追蹤碳排放的方法來選擇區域。如需根據永續性指導方針來選擇區域的詳細資訊，請參閱如何根據永續性目標來選取工作負載的區域。

資源

相關文件：

- 了解您的碳排放預估值
- 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

相關影片：

- AWS re:Invent 2023 - AWS 全球基礎設施的永續創新
- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- AWS re:Invent 2022 - 提供永續且高效能的架構
- AWS re:Invent 2022 - 永續架構並降低您的 AWS 碳足跡

- AWS re:Invent 2023 - AWS 全球基礎設施的永續性

因應需求

問題

- SUS 2 如何根據您的需求取得適當的雲端資源？

SUS 2 如何根據您的需求取得適當的雲端資源？

使用者和應用程式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。擴展基礎架構以持續符合需求，並確認您僅使用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，以限制使用者和應用程式使用資源所需的網路。移除未使用的資產。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 動態擴展工作負載基礎架構
- SUS02-BP02 讓 SLA 符合永續性目標
- SUS02-BP03 停止建立和維護未使用的資產
- SUS02-BP04 根據使用者的聯網要求優化其工作負載的地理位置
- SUS02-BP05 為執行的活動優化團隊成員資源
- SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化

SUS02-BP01 動態擴展工作負載基礎架構

利用雲端的彈性動態擴展您的基礎架構，以達到雲端資源的供需平衡，避免工作負載出現過度佈建的容量。

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：設定並測試工作負載彈性有助於有效達到雲端資源的供需平衡，並避免過度佈建的容量。您可以利用雲端中的彈性，在需求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業務所需。

未建立此最佳實務時的風險暴露等級：中

實作指引

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。平衡供需關係可將工作負載受到的影響降到最低。

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成為繁重的工作。應用程式可藉由修改執行個體大小進行垂直調整 (縱向擴展或縮減規模)、藉由修改執行個體數目進行水平調整 (縮減或橫向擴展)，或進行兩者的合併調整。

您可以使用多種不同的方法達到資源的供需平衡。

- 目標追蹤法：監控您的擴展指標，並視需要自動增加或減少容量。
- 預測擴展：縮減每日和每週趨勢的預期。
- 排程法：根據可預測的負載變化設定您自己的擴展排程。
- 服務擴展： 挑選按設計原本就會擴展的服務 (例如無伺服器)，或提供自動擴展功能。

辨別使用率低或無使用率的時期，並調整資源規模以移除過剩容量、提高效率。

實作步驟

- 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。AWS 提供了多種自動擴展機制，以確保工作負載可在使用者負載較低時迅速輕易地縮減規模。以下是自動擴展機制的幾個範例：

Auto scaling mechanism	Where to use
Amazon EC2 Auto Scaling	用來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。
Application Auto Scaling	用來自動將個別 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。

Auto scaling mechanism	Where to use
Kubernetes Cluster Autoscaler	用來自動擴展 AWS 上的 Kubernetes 叢集。

- 擴展常與 Amazon EC2 執行個體或 AWS Lambda 函數等服務一起討論。請考慮設定非運算服務 (例如 Amazon DynamoDB 讀取和寫入容量單位或 Amazon Kinesis Data Streams 碎片) 以符合需求。
- 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。
- 對於 Auto Scaling 群組請使用動態擴展，而非手動擴展。我們也建議您在動態擴展中使用目標追蹤擴展政策。
- 確認工作負載部署可處理橫向擴展和縮減事件。建立縮減事件的測試案例，以確認工作負載的行為符合預期，且不會對使用者體驗造成影響 (例如失去黏性工作階段)。您可以使用活動歷史來驗證 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。透過預測擴展，可以避免過度佈建容量的需求。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 的預測擴展。

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？
- 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
- 介紹 Karpenter - 一個開放原始碼的高效能 Kubernetes Cluster Autoscaler
- 深入探討 Amazon ECS 叢集 Auto Scaling

相關影片：

- AWS re:Invent 2023 - 為前一千萬名使用者擴展 AWS
- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- AWS re:Invent 2022 - 建置具成本、能源和資源效率的運算環境
- AWS re:Invent 2022 - 將容器從一個使用者擴展到數百萬名使用者
- AWS re:Invent 2023 - 使用 Amazon SageMaker 將 FM 推論擴展到數百種模型
- AWS re:Invent 2023 - 利用 Karpenter 的力量來擴展、最佳化和升級 Kubernetes

相關範例：

- Autoscaling

SUS02-BP02 讓 SLA 符合永續性目標

根據您的永續性目標審查並優化工作負載的服務水準協議 (SLA)，例如可用性或資料保留期，以盡可能減少支援工作負載所需的資源，同時繼續滿足商業需求。

常見的反模式：

- 工作負載 SLA 不明或語意不清。
- 您僅針對可用性和效能定義 SLA。
- 您對所有的工作負載使用相同的設計模式 (例如多可用區域架構)。

建立此最佳實務的效益：讓 SLA 符合永續性目標，可達到最佳資源用量並滿足商業需求。

未建立此最佳實務時的風險暴露等級：低

實作指引

SLA 會定義雲端工作負載應有的服務水準，例如回應時間、可用性和資料保留。其影響範圍涵蓋雲端工作負載的架構、資源用量和環境影響。請定期審查 SLA，並做出能大幅降低資源用量的取捨，換取可接受的服務水準降低。

實作步驟

- 了解永續性目標：確定組織中的永續性目標，例如減碳或提高資源使用率。
- 審核 SLA：評估您的 SLA，以評估它們是否支援您的業務需求。如果會超過 SLA，請執行進一步檢閱。

- 了解權衡取捨：了解工作負載複雜度 (例如大量並行使用者)、效能 (例如延遲) 和永續性影響 (例如所需資源) 之間的衡量取捨。通常，如果優先考慮兩個因素，將會犧牲第三個因素。
- 調整 SLA：藉由在大幅降低永續性影響中做出權衡取捨，以換取可接受的服務水準降低的做法來調整 SLA。
  - 永續性和可靠性：高可用性的工作負載往往會耗用較多資源。
  - 永續性和效能：使用較多資源以提升效能，可能會對環境造成較大的影響。
  - 永續性和安全性：保護過度的工作負載可能會對環境造成較大的影響。
- 盡可能定義永續性 SLA：為您的工作負載納入永續性 SLA。例如，將最低使用率層級定義為運算執行個體的永續性 SLA。
- 使用高效率的設計模式：使用優先執行業務關鍵功能的設計模式 (例如 AWS 上的微型服務)，對於非關鍵功能允許採用較低的服務水準 (例如回應時間或復原時間目標)。
- 傳達並建立責任：與所有相關利害關係人分享 SLA，包括您的開發團隊及客戶。使用報告來追蹤和監控 SLA。指派責任以達到 SLA 的永續性目標。
- 使用激勵和獎勵：使用激勵和獎勵來實現或超越與永續性目標一致的 SLA。
- 檢閱並反覆：定期審查和調整 SLA，確保它們與不斷演變的永續發展和效能目標一致。

資源

相關文件：

- 了解能在雲端中快速建構架構的彈性模式和權衡
- 服務水準協議對 SaaS 供應商的重要性

相關影片：

- AWS re:Invent 2023 - 容量、可用性、成本效率：聚焦三件事
- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- AWS re:Invent 2023 - 適用於鬆散耦合系統的進階整合模式與權衡
- AWS re:Invent 2022 - 提供永續且高效能的架構
- AWS re:Invent 2022 - 建置具成本、能源和資源效率的運算環境

SUS02-BP03 停止建立和維護未使用的資產

將您工作負載中未使用的資產除役，以降低支援您個人需求所需的雲端資源數量，並盡可能減少浪費。

常見的反模式：

- 您未分析應用程式是否有冗餘或不再需要的資產。
- 您未移除冗餘或不再需要的資產。

建立此最佳實務的優勢：移除未使用的資產可釋出資源，並改善工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

未使用的資產會耗用儲存空間和運算能力等資源。識別這些資產並將其消除可以釋出這類資源，進而提升雲端架構的效能。定期分析應用程式資產 (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。移除這類冗餘資產，避免工作負載中的資源浪費。

實作步驟

- 執行清查：進行全面清查，以識別工作負載中的所有資產。
- 分析用量：使用持續監控功能識別不再需要的靜態資產。
- 移除未使用的資產：制定計畫來移除不再需要的資產。
  - 移除任何資產之前，均應先評估該移除對架構的影響。
  - 合併重疊產生的資產以消除冗餘處理。
  - 更新您的應用程式，使其不再產生及儲存不需要的資產。
- 與第三方通訊：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。請求合併冗餘資產。
- 使用生命週期政策：使用生命週期政策來自動刪除未使用的資產。
  - 您可以使用 Amazon S3 生命週期，以在物件的整個生命週期中管理物件。
  - 您可以使用 Amazon Data Lifecycle Manager 自動建立、保留和刪除 Amazon EBS 快照與 Amazon EBS 支援的 AMI。
- 檢閱和最佳化：定期檢閱工作負載，以識別並移除任何未使用的資產。

資源

相關文件：

- 優化您的 AWS 永續性基礎設施，第 II 部分：儲存
- 如何終止我的 AWS 帳戶 上不再需要的作用中資源？

相關影片：

- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- AWS re:Invent 2022 - 使用 Amazon S3 數位媒體資產保存和發揮最大價值
- AWS re:Invent 2023 - 最佳化在多帳戶環境中的成本

SUS02-BP04 根據使用者的聯網要求優化其工作負載的地理位置

為您的工作負載選取可減少網路流量傳輸距離的區域和服務，並減少支援工作負載所需的整體網路資源。

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。
- 您可以將所有工作負載資源合併到單一地理位置。
- 通過現有資料中心的所有流量。

建立此最佳實務的優勢：將工作負載分配到使用者附近的位置，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風險暴露等級：中

實作指引

AWS 雲端 基礎設施是根據如下的位置選項而建置的：區域、可用區域、放置群組和邊緣位置 (例如 AWS Outposts 和 AWS Local Zones)。這些位置選項負責維護應用程式元件、雲端服務、邊緣網路和內部部署資料中心之間的連線。

分析工作負載中的網路存取模式，以識別如何使用這些雲端位置選項，以及減少網路流量必須輸送的距離。

實作步驟

- 分析您工作負載中的網路存取模式，以識別使用者如何使用您的應用程式。
  - 使用監控工具 (例如 Amazon CloudWatch 和 AWS CloudTrail) 收集網路活動的相關資料。
  - 分析資料以識別網路存取模式。
- 根據下列關鍵元素，為您的工作負載部署選取區域：
  - 您的永續性目標：相關說明請見區域選擇。

  - 資料的所在位置：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。
  - 使用者的所在位置：對於面向使用者的應用程式，請選擇接近工作負載使用者的一或多個區域。
  - 其他限制：請考量成本和合規性之類的限制，相關說明請見為工作負載選取區域時應考量的事項。
- 使用本機快取或 AWS 快取解決方案取得常用資產，以提升效能、減少資料移動，以及降低環境影響。

Service	When to use
Amazon CloudFront	用來快取靜態內容 (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)。
Amazon ElastiCache	用來快取 Web 應用程式的內容。
DynamoDB Accelerator	用來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：

Service	When to use
Lambda@Edge	用於在物件未經快取時起始的大量運算作業。
Amazon CloudFront 函數	用於簡單的使用案例，例如可由短期函數起始的 HTTP(s) 請求或回應操作。
AWS IoT Greengrass	用來為連線的裝置執行本機運算、傳訊和資料快取。

- 使用連線共用來支援連線重複使用，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 永續性基礎設施，第 III 部分：聯網
- Amazon ElastiCache 文件
- 什麼是 Amazon CloudFront？
- Amazon CloudFront 主要功能
- AWS 全球基礎設施
- AWS Local Zones 和 AWS Outposts，為您的邊緣工作負載選擇正確的技術
- 置放群組
- AWS Local Zones
- AWS Outposts

相關影片：

- 揭密 AWS 上的資料傳輸
- 擴展新一代 Amazon EC2 執行個體的網路效能
- AWS Local Zones 解說影片
- AWS Outposts：概觀和運作方式
- AWS re:Invent 2023 - 邊緣與內部部署工作負載的遷移策略
- AWS re:Invent 2021 - AWS Outposts：在內部部署環境帶來 AWS 體驗
- AWS re:Invent 2020 - AWS Wavelength：在 5G 邊緣以極低延遲執行應用程式
- AWS re:Invent 2022 - AWS Local Zones：為分散的邊緣建置應用程式
- AWS re:Invent 2021 - 使用 Amazon CloudFront 建置低延遲網站
- AWS re:Invent 2022 - 使用 AWS Global Accelerator 改善效能與可用性
- AWS re:Invent 2022 - 使用 AWS 建置您的全球廣域網路
- AWS re:Invent 2020：使用 Amazon Route 53 進行全球流量管理

相關範例：

- AWS 聯網研討會
- 永續性架構 - 盡可能減少跨網路的資料移動

SUS02-BP05 為執行的活動優化團隊成員資源

優化提供給團隊成員的資源，以盡量減少對環境永續性的影響，同時支援他們的需求。

常見的反模式：

- 您忽略團隊成員所使用的裝置對雲端應用程式的整體效率產生的影響。
- 您手動管理及更新團隊成員所使用的資源。

建立此最佳實務的優勢：優化團隊成員資源，可為具備雲端功能的應用程式改善整體效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

了解團隊成員用來使用您的服務的資源、其預期生命週期，以及財務和永續性的影響。實作將這些資源優化的策略。例如，在使用率高的可擴展基礎設施上執行複雜的操作 (例如轉譯和編譯)，而不是在使用率低的高功率單一使用者系統上執行。

實作步驟

- 使用節能工作站：為團隊成員提供節能的工作站和周邊裝置。在這些裝置中使用高效電源管理功能 (例如低功耗模式)，以減少其能源用量
- 使用虛擬化：使用虛擬桌面和應用程式串流來限縮升級與裝置要求。
- 鼓勵遠端協作：鼓勵團隊成員使用遠端協作工具 (如 Amazon Chime 或 AWS Wickr) 以減少出差需求和相關聯的碳排放。
- 使用節能軟體：移除或關閉不必要的功能和流程，為團隊成員提供節能軟體。
- 管理生命週期：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。定期維護和更新工作站或軟體，以維護和提高效率。
- 遠端裝置管理：實作裝置的遠端管理，以降低商務旅行需求。
  - AWS Systems ManagerFleet Manager 是一種整合式使用者介面 (UI) 體驗，可協助您從遠端管理在 AWS 或內部部署環境執行的節點。

資源

相關文件：

- 什麼是 Amazon WorkSpaces？
- Amazon WorkSpaces 的成本優化器
- Amazon AppStream 2.0 文件
- NICE DCV

相關影片：

- 在 AWS 上管理 Amazon WorkSpaces 的成本

SUS02-BP06 實作緩衝或調節使需求曲線趨於扁平化

緩衝和調節可讓需求曲線趨於扁平化，並減少您的工作負載所需的已佈建容量。

常見的反模式：

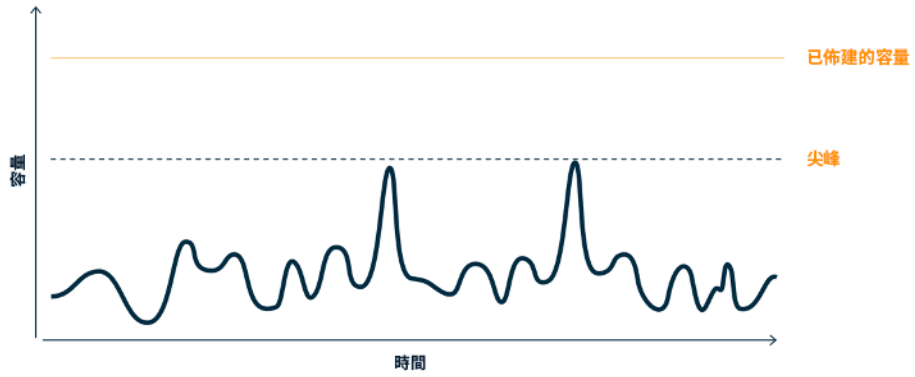
- 您非必要地立即處理用戶端要求。
- 您未分析用戶端要求的需求。

建立此最佳實務的優勢：讓需求曲線趨於扁平化，可減少工作負載所需的已佈建容量。減少已佈建的容量意味著較低的能源耗用量和環境影響。

未建立此最佳實務時的風險暴露等級：低

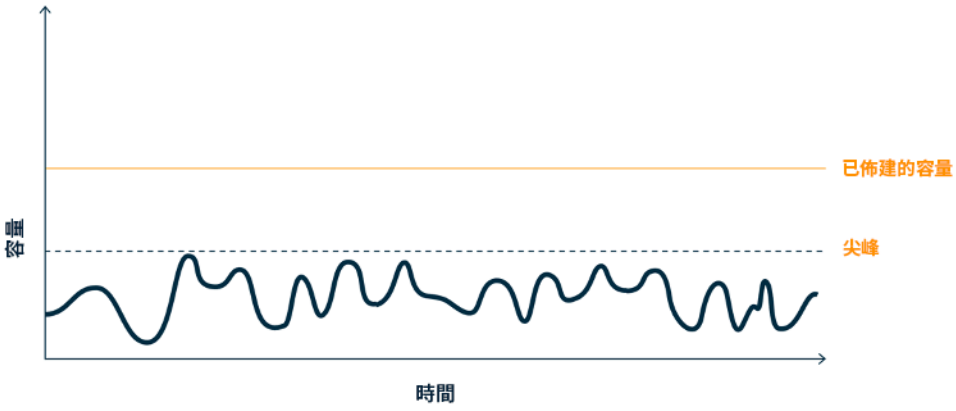
實作指引

使工作負載需求曲線扁平化，有助於減少工作負載所需的已佈建容量，以及降低對環境造成的影響。假設某個工作負載的需求曲線如下圖所示。此工作負載有兩個尖峰，為了處理這些尖峰，已佈建了資源容量 (以橙色線顯示)。用於此工作負載的資源和能源並非由需求曲線底下的區域表示，而是已佈建的容量底下的區域，因為這兩個尖峰必須用已佈建的容量處理。



需求曲線圖，內含兩個需要大量已佈建容量的相異尖峰。

您可以使用緩衝或調節來修正需求曲線，並使尖峰趨緩，意即減少已佈建的容量和耗用的能源。在用戶端可執行重試時實作調節。實作緩衝機制以儲存請求，並將處理的時間往後延遲。



調節對需求曲線和已佈建容量的影響。

實作步驟

- 分析用戶端要求以判斷如何予以回應。應考量的問題包括：
  - 此要求是否可進行非同步處理？
  - 用戶端是否有重試能力？
- 如果用戶端有重試功能，您可以實作調節，以告知來源若目前無法處理要求，則應稍後再試。
  - 您可以使用 Amazon API Gateway 來實作調節。
- 針對無法執行重試的用戶端，必須實作緩衝區使需求曲線扁平化。緩衝會延遲要求處理，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或串流來接受生產者傳出的訊息。消費者可讀取訊息並進行處理，允許以符合取用者業務要求的速度運作訊息。
  - Amazon Simple Queue Service (Amazon SQS) 是一個受管服務，可提供佇列，允許單一取用者讀取個別訊息。
  - Amazon Kinesis 可提供串流，允許許多取用者讀取相同訊息。
- 分析整體需求、變更率及所需的回應時間，以適當調整所需的調節或緩衝區大小。

資源

相關文件：

- Amazon SQS 入門
- 使用佇列和訊息進行應用程式整合
- 管理和監控工作負載中的 API 調節

- 使用 API Gateway 大規模地限流分級的多租用戶 REST API
- 使用佇列和訊息進行應用程式整合

相關影片：

- AWS re:Invent 2022 - 微型服務的應用程式整合模式
- AWS re:Invent 2023 - 智慧型節約：Amazon EC2 成本最佳化策略
- AWS re:Invent 2023 - 適用於鬆散耦合系統的進階整合模式與權衡

軟體和架構

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讓所部署資源保持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。留意客戶用來存取服務的裝置，並實施盡量減少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
- SUS03-BP02 移除或重構使用量低或完全未使用的工作負載元件
- SUS03-BP03 優化程式碼中耗用最多時間或資源的部分
- SUS03-BP04 優化對裝置和設備的影響
- 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

SUS03-BP01 最佳化非同步與排程任務的軟體和架構

使用有效率的軟體和架構模式 (例如佇列驅動)，讓所部署的資源一直保持高使用率。

常見的反模式：

- 在雲端工作負載中過度佈建資源以滿足未預料到的突增需求。

- 您的架構未透過傳訊元件將非同步訊息的傳送者與接受者分離。

建立此最佳實務的優勢：

- 有效率的軟體和架構模式可盡量減少工作負載中的未使用資源，並改善整體效率。
- 您可以將非同步訊息的處理與接收分開擴展。
- 透過傳訊元件，可用性要求會比較寬鬆，不用太多資源即可滿足。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用有效率的架構模式 (例如事件驅動架構)，以便能平均地使用元件，並盡量避免工作負載過度佈建。使用有效率的架構模式可盡量地讓閒置資源不會因為需求隨時間發生變化而有乏人問津的情形。

了解工作負載元件的要求，並採用能夠提升整體資源使用率的架構模式。淘汰不再需要的元件。

實作步驟

- 分析工作負載需求以判斷如何回應。
- 如果請求或作業不需要同步回應，請使用佇列驅動的架構和 Auto Scaling 工作節點，以將使用率最大化。以下是您可能會考慮使用佇列驅動架構的一些範例：

Queuing mechanism	Description
AWS Batch 作業佇列	AWS Batch 作業會提交至作業佇列並停留其中，直到能夠排定在運算環境中執行為止。
Amazon Simple Queue Service 和 Amazon EC2 Spot 執行個體	搭配使用 Amazon SQS 與 Spot 執行個體即可建置能容錯且有效率的架構。

- 對於可以隨時處理的佇列或作業，請使用排程機制來批次處理作業，以提升效率。以下是 AWS 上排程機制的幾個範例：

Scheduling mechanism	Description
Amazon EventBridge 排程器	這是 Amazon EventBridge 的一項功能，可讓您大規模地建立、執行和管理已排定的任務。

Scheduling mechanism	Description
AWS Glue 時間型排程	在 AWS Glue 中定義爬蟲程式和作業的時間型排程。
Amazon Elastic Container Service (Amazon ECS) 排定的任務	Amazon ECS 支援建立排定的任務排定的任務會使用 Amazon EventBridge 規則，依排程執行任務或是在 EventBridge 事件的回應中執行任務。
Instance Scheduler	設定 Amazon EC2 和 Amazon Relational Database Service 執行個體的開始和停止排程。

- 如果您的架構中使用輪詢和 Webhook 機制，請將其更換為事件。使用事件驅動的架構可建置高效率的工作負載。
- 利用 AWS 上的無伺服器來消除過度佈建的基礎設施。
- 將架構的個別元件調整為適當大小，避免閒置資源等待輸入。
  - 您可以使用AWS Cost Explorer 中的適當調整大小建議或 AWS Compute Optimizer 找出適當調整大小的機會。
  - 如需詳細資訊，請參閱適當調整大小：佈建執行個體以符合工作負載。

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？
- 使用 REST API 管理非同步工作流程

相關影片：

- AWS re:Invent 2023 - 將旅程導向無伺服器事件驅動架構
- AWS re:Invent 2023 - 針對事件驅動架構和網域驅動設計使用無伺服器
- AWS re:Invent 2023 - 搭配 Amazon EventBridge 的進階事件驅動模式
- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- 異步訊息模式 | AWS 事件

相關範例：

- 具有 AWS Graviton 處理器和 Amazon EC2 Spot 執行個體的事件驅動架構

SUS03-BP02 移除或重構使用量低或完全未使用的工作負載元件

移除未使用且不再需要的元件，並重構使用率低的元件，以盡可能避免工作負載中的浪費。

常見的反模式：

- 您未定期檢查個別工作負載元件的使用率水準。
- 您未查看並分析 AWS 適當調整大小的工具 (例如 AWS Compute Optimizer) 所提供的建議。

建立此最佳實務的優勢：移除未使用的元件可盡量避免浪費，並改善雲端工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

審查您的工作負載以識別閒置或未使用的元件。有一個迭代改進程序可由需求的變更或新雲端服務的發行來啟動。例如，AWS Lambda 函數執行時間的大幅下降可能意味著必須降低記憶體大小。此外，隨著 AWS 發行新的服務和功能，工作負載的最佳服務與架構可能會改變。

持續監控工作負載活動，並找機會改善個別元件的使用率水準。藉由移除閒置元件和執行適當調整大小的活動，您將可用最少的雲端資源達到業務要求。

實作步驟

- 您的 AWS 資源尚有庫存。在 AWS 中，可以開啟 AWS 資源總管 探索和組織您的 AWS 資源。如需詳細資訊，請參閱 AWS re:Invent 2022 - 在 AWS 上如何大規模管理資源和應用程式。

- 監控及擷取工作負載關鍵元件的使用率指標 (例如 Amazon CloudWatch 指標中的 CPU 使用率、記憶體使用率或網路輸送量)。
- 識別架構中完全未使用或使用率不足的元件。
  - 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)，以識別閒置、未使用或未充分利用的元件。
  - 對於暫時性工作負載，請評估使用率指標以識別閒置、未使用或未充分利用的元件。
- 淘汰不再需要的元件和相關聯的資產 (例如 Amazon ECR 映像)。
  - 自動清理 Amazon ECR 中未曾使用的映像
  - 使用 AWS Config 和 AWS Systems Manager 刪除未曾使用的 Amazon Elastic Block Store (Amazon EBS) 磁碟區
- 重構或整合未充分利用的元件與其他資源，以提高利用效率。例如，您可將多個小資料庫佈建至單一 Amazon RDS 資料庫執行個體，而不要在未充分利用的個別執行個體上執行資料庫。
- 了解您的工作量佈建以完成工作單位的資源。

資源

相關文件：

- AWS Trusted Advisor
- 什麼是 Amazon CloudWatch？
- 適當調整大小：佈建執行個體以符合工作負載
- 利用精簡化建議將您的成本最佳化

相關影片：

- AWS re:Invent 2023 - 容量、可用性、成本效率：聚焦三件事

相關範例：

- 最佳化硬體模式和觀察永續性 KPI

SUS03-BP03 優化程式碼中耗用最多時間或資源的部分

優化您的架構不同元件中執行的程式碼，將資源使用量降至最低，同時發揮最大效能。

常見的反模式：

- 您略過資源用量的程式碼優化。
- 您通常藉由增加資源來因應效能問題。
- 您的程式碼審查和開發程序未追蹤效能變更。

建立此最佳實務的優勢：使用有效率的程式碼可將資源用量壓到最低，並改善效能。

未建立此最佳實務時的風險暴露等級：中

實作指引

請務必檢查各個功能領域 (包括雲端架構應用程式的程式碼)，以優化其資源用量和效能。持續監控您的工作負載在建置環境和生產環境中的效能，並找機會改進資源用量特別高的程式碼片段。採用定期審查程序，在您的程式碼內識別低效使用資源的錯誤或反模式。使用簡單有效的演算法為您的使用案例產生相同結果。

實作步驟

- 使用高效率的編程語言：使用適用於工作負載的高效率作業系統和程式設計語言。如需關於高能效程式設計語言 (包括 Rust) 的詳細資料，請參閱 Rust 的永續性。
- 使用 AI 編碼配套：考慮使用 AI 編碼配套 (例如Amazon CodeWhisperer) 以有效率地撰寫程式碼。
- 自動執行程式碼審查：在擬定工作負載時採用自動化程式碼審查程序，以改善品質並識別錯誤和反模式。
  - 使用 Amazon CodeGuru Reviewer 自動化程式碼檢閱
  - 使用 Amazon CodeGuru 偵測並行錯誤
  - 使用 Amazon CodeGuru 提升 Python 應用程式的程式碼品質
- 使用程式碼分析工具：使用程式碼分析工具識別程式碼中耗用最多時間或資源的部分，作為最佳化目標。
  - 透過 Amazon CodeGuru Profiler 降低組織的碳足跡
  - 透過 Amazon CodeGuru Profiler 了解 Java 應用程式中的記憶體用量
  - 透過 Amazon CodeGuru Profiler 改善客戶體驗並降低成本
- 監控和最佳化：使用持續監控資源來識別資源需求高或組態不夠好的元件。
  - 將需要大量運算資源的演算法取代為會產生相同結果、但更簡單有效率的版本。
  - 移除不必要程式碼，例如排序和格式化。

- 使用程式碼重構或轉換：探索用於應用程式維護和升級的 Amazon Q 程式碼轉換的可能性。
  - 使用 Amazon Q 程式轉換將語言版本升級
  - AWS re:Invent 2023 - 使用 Amazon Q 程式碼轉換以自動執行應用程式升級與維護

資源

相關文件：

- 什麼是 Amazon CodeGuru Profiler？
- FPGA 執行個體
- 在 AWS 上建立的工具中的 AWS SDK

相關影片：

- 使用 Amazon CodeGuru Profiler 改善程式碼效率
- AWS re:Invent 2023 - Amazon CodeWhisperer 的最佳實務
- 使用 Amazon CodeGuru 自動進行程式碼審查和應用程式效能建議

相關範例：

- 使用 Amazon CodeGuru 最佳化程式碼

SUS03-BP04 優化對裝置和設備的影響

了解您的架構中使用的裝置和設備，並使用策略降低其用量。這樣可以盡量減輕對雲端工作負載的整體環境影響。

常見的反模式：

- 您忽略了客戶使用的裝置所受到的環境影響。
- 您手動管理及更新客戶所使用的資源。

建立此最佳實務的優勢：實作為客戶裝置優化的軟體模式和功能，可降低雲端工作負載的整體環境影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

實作為客戶裝置優化的軟體模式和功能，可透過數種方式降低環境影響：

- 實作具回溯相容性的新功能，可減少硬體更換的數量。
- 將應用程式優化以有效執行於裝置上，有助於降低其能源耗用量及延長電池使用壽命 (若是由電池供電)。
- 優化裝置的應用程式也可減少網路上的資料傳輸。

了解客戶您的架構中使用的裝置和設備、其預期生命週期，以及更換這些元件的影響。實作適當的軟體模式和功能，以盡可能減少裝置能源耗用量，以及客戶更換裝置和手動加以升級的需求。

實作步驟

- 執行清查：清查架構中使用的裝置。裝置可以是行動裝置、平板裝置、IOT 裝置、智慧電燈，甚或是工廠的智慧裝置。
- 使用節能裝置：考慮在您的架構中使用節能裝置。在不使用時，使用裝置上的電源管理組態進入低功耗模式。
- 執行高效應用程式：最佳化裝置上執行的應用程式：
  - 採用在背景執行任務之類的策略來降低能源耗用量。
  - 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。
  - 將承載和檔案轉換為裝置所需的優化格式。例如，您可以使用 Amazon Elastic Transcoder 或 AWS Elemental MediaConvert 將較大的高品質數位媒體檔案轉換為使用者可在行動裝置、平板電腦、Web 瀏覽器和聯網電視上播放的格式。
  - 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。
- 鼓勵供應商參與其中：與使用永續性材料的裝置供應商合作，並與在供應鏈中提供透明度和環境認證的裝置供應商合作
- 使用空中下載技術 (OTA) 更新：使用自動化空中下載技術 (OTA) 機制將更新部署到一或多個裝置。
  - 您可以使用 CI/CD 管道更新行動應用程式。
  - 您可以使用 AWS IoT Device Management 從遠端大規模管理連網裝置。

- 使用受管 Device Farm：若要測試新功能和更新，請使用具有代表性硬體集的受管 Device Farm，並迭代開發以最大化支援的裝置。如需詳細資訊，請參閱 SUS06-BP04 使用受管 Device Farm 進行測試。
- 繼續監控和改善：追蹤裝置的能源使用情況，以確定需要改善的區域。使用新技術或最佳實務來增強這些裝置對環境的影響。

資源

相關文件：

- 什麼是 AWS Device Farm？
- AppStream 2.0 文件
- NICE DCV
- 在執行 FreeRTOS 的裝置上更新韌體的 OTA 教學
- 最佳化您的 IoT 裝置以實現環境永續性

相關影片：

- AWS re:Invent 2023 - 使用 AWS Device Farm 改善行動裝置和 Web 應用程式品質

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。使用最能支援資料存取和儲存的軟體模式與架構，以盡可能減少支援工作負載所需的運算、聯網和儲存資源。

常見的反模式：

- 您假設所有工作負載具有類似的資料儲存和存取模式。
- 您只使用一個存儲層 – 假設所有工作負載都適合該層。
- 您假設資料存取模式不會隨著時間改變。
- 您的架構支援潛在的高資料存取爆量，這會導致資源在大部分的時間處於閒置狀態。

建立此最佳實務的優勢：根據資料存取和儲存模式選取及優化您的架構，有助於降低開發複雜性並提升整體使用率。了解何時使用全域表、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。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用與您的資料特性和存取模式最相符的軟體和架構模式。例如，使用 AWS 上的現代資料架構 (可讓您使用針對個人獨特分析使用案例而優化的專用服務)。這些架構模式有利於高效率資料處理並降低資源用量。

實作步驟

- 分析您的資料特性和存取模式，以識別雲端資源的正確組態。應考量的重要特性包括：
  - 資料類型：結構化、半結構化、非結構化
  - 資料成長：有界限、無界限
  - 資料耐用性：持續性、暫時性、臨時
  - 存取模式：讀取或寫入、更新頻率、尖峰或一致
- 使用最能支援資料存取和儲存模式的架構模式。
  - 啟用資料持續性的模式
  - 開始建構吧！ 現代資料架構
  - AWS 上的資料庫：使用正確的工具完成任務
- 利用可原生處理壓縮資料的技術。
  - Athena 壓縮支援檔案格式
  - AWS Glue 中 ETL 輸入和輸出的格式選項
  - 使用 Amazon Redshift 從 Amazon S3 載入壓縮的資料檔案
- 使用專用分析服務進行架構中的資料處理。如需有關 AWS 專用分析服務的詳細資訊，請參閱 AWS re:Invent 2022 - 在 AWS 上建置現代資料架構。
- 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，確保高效率查詢。如需進一步詳細資訊，請參閱AWS資料庫和 AWS re:Invent 2022 - 使用專用資料庫將應用程式現代化。
- 選取可減少架構中網路容量消耗的網路通訊協定。

資源

相關文件：

- 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

- 轉換為單欄格式，提高 Amazon Athena 的查詢效能
- 在 Amazon Aurora 上使用 Performance Insights 監控資料庫負載
- 在 Amazon RDS 上使用 Performance Insights 監控資料庫負載
- Amazon S3 Intelligent-Tiering 儲存類別
- 使用 Amazon DynamoDB 建置 CQRS 事件存放區

相關影片：

- AWS re:Invent 2022 - 在 AWS 上建置資料網格架構
- AWS re:Invent 2023 - 深入探索 Amazon Aurora 及其創新
- AWS re:Invent 2023 - 提高 Amazon EBS 效率並更具成本效益
- AWS re:Invent 2023 - 利用 Amazon S3 最佳化儲存價格和效能
- AWS re:Invent 2023 - 在 Amazon S3 上建置資料湖並進行最佳化
- AWS re:Invent 2023 - 搭配 Amazon EventBridge 的進階事件驅動模式

相關範例：

- AWS 專用資料庫研討會
- AWS 現代資料架構 Immersion Day
- 在 AWS 上建置資料網格

資料

問題

- SUS 4 如何利用資料管理政策和模式來支持您的永續性目標？

SUS 4 如何利用資料管理政策和模式來支持您的永續性目標？

實作資料管理實務來減少支援工作負載所需的佈建儲存，以及減少為了使用它所需的資源。了解您的資料，並使用更有效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

- SUS04-BP02 使用支援資料存取和儲存模式的技術
- SUS04-BP03 使用政策來管理資料集的生命週期
- SUS04-BP04 使用彈性和自動化擴充區塊儲存或檔案系統
- SUS04-BP05 移除不需要或多餘的資料
- SUS04-BP06 使用共用檔案系統或儲存體存取通用資料
- SUS04-BP07 盡可能減少跨網路的資料移動
- SUS04-BP08 僅在難以重新建立時才備份資料

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關鍵性，並選擇適當的 節能儲存層來儲存資料。

常見的反模式：

- 您未以正在處理或已儲存的類似特性來識別資料資產 (例如敏感性、業務關鍵性或法規要求)。
- 您未實作資料目錄以清查資料資產。

建立此最佳實務的優勢：實作資料分類政策，可讓您確認最節能的資料儲存層。

未建立此最佳實務時的風險暴露等級：中

實作指引

資料分類的工作之一，是識別正在處理的資料類型，以及由組織擁有或操作的資訊系統中儲存的資料類型。此外也須確認資料的關鍵性，以及資料損毀、遺失或誤用可能造成的影響。

若要實作資料分類政策，請從資料的情境使用採取逆向思維，並建立適當的分類機制，將指定資料集的關鍵性程度納入組織操作的考量中。

實作步驟

- 執行資料清查：對您工作負載現有的各種資料類型執行清查。
- 將資料分組：根據組織面臨的風險，判定資料的關鍵性、機密性、完整性和可用性。使用這些要求，將資料分組為您採用的其中一個資料分類層。範例請見分類資料及保護新創公司的四個簡單步驟。
- 定義資料分類層級和政策：針對每一個資料群組定義資料分類層級 (例如公開或機密) 和處理政策。據以標記資料。如需關於資料分類類別的詳細資訊，請參閱《資料分類白皮書》。

- 定期審查：定期審查與稽核您的環境，以尋找未標記及未分類的資料。使用自動化功能來識別這些資料，並適當地分類和標示資料。範例請見 AWS Glue 中的資料目錄和編目程式。
- 建立資料目錄：建立提供稽核及管控能力的資料目錄。
- 文件：記錄每個資料類別的資料分類政策和處理程序。

資源

相關文件：

- 使用 AWS 雲端 以支援資料分類
- 標記來自 AWS Organizations 的政策

相關影片：

- AWS re:Invent 2022 - 在 AWS 上發揮敏捷性與資料管控
- AWS re:Invent 2023- 透過 AWS 儲存裝置提供資料保護與彈性

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技術，以在支援工作負載的同時，也將佈建的資源降至最低。

常見的反模式：

- 您假設所有工作負載具有類似的資料儲存和存取模式。
- 您只使用一個存儲層 – 假設所有工作負載都適合該層。
- 您假設資料存取模式不會隨著時間改變。

建立此最佳實務的效益：根據資料存取和儲存模式來選取及優化您的儲存技術，可協助您降低達成商業需求所需的雲端資源，並改善雲端工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

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，從而達到最大的效能效率。

實作步驟

- 評估資料和存取特性：評估您的資料特性和存取模式，以收集儲存需求的重要特性。應考量的重要特性包括：
  - 資料類型：結構化、半結構化、非結構化
  - 資料成長：有界限、無界限
  - 資料耐用性：持續性、暫時性、臨時
  - 存取模式：讀取或寫入、頻率、尖峰或一致
- 選擇適當的儲存技術：將資料遷移至支援您的資料特性和存取模式的適當儲存技術。以下提供 AWS 儲存技術及其重要特性的一些範例：

Type	Technology	Key characteristics
物件儲存	Amazon S3	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 Transfer Acceleration 或 Access Points 之類的服務來支援您的位置、安全需求和存取模式。
封存儲存	Amazon S3 Glacier	針對資料封存而建置的 Amazon S3 儲存類別。
共用檔案系統	Amazon Elastic File System (Amazon EFS)	可供多種類型的運算解決方案存取的可掛載檔案系統。Amazon EFS 會自動增長及縮減儲存體，藉以進行效能優化而提供一致的低延遲。
共用檔案系統	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

Type	Technology	Key characteristics
		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
區塊儲存	Amazon Elastic Block Store (Amazon EBS)	針對 Amazon Elastic Compute Cloud (Amazon EC2) 而設計的可擴展、高效能區塊儲存服務。Amazon EBS 包含支援 SSD 的儲存系統 (用於交易、IOPS 密集工作負載)，以及支援 HDD 的儲存系統 (用於輸送量密集工作負載)。
關聯式資料庫	Amazon Aurora、Amazon RDS、Amazon Redshift	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) 和電子商務系統都使用關聯式資料庫來儲存資料。
鍵值資料庫	Amazon DynamoDB	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。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。

- 自動執行儲存空間配置：對於固定大小的儲存系統 (例如 Amazon EBS 或 Amazon FSx)，請監控可用儲存空間，並且在接近閾值時自動執行儲存空間分配。您可以利用 Amazon CloudWatch 來收集及分析 Amazon EBS 和 Amazon FSx 的不同指標。
- 選擇適當的儲存類別：選擇適當的資料儲存類別。
  - Amazon S3 儲存類別可以在物件層級設定。單一儲存貯體可以包含存放於所有儲存類別的物件。

- 您可以使用 Amazon S3 生命週期政策在儲存類別之間自動轉換物件或移除資料，而無需變更任何應用程式。在考量這些儲存機制時，您通常需要在資源效率、存取延遲與可靠性之間做出取捨。

資源

相關文件：

- Amazon EBS 磁碟區類型
- Amazon EC2 執行個體存放區
- Amazon S3 Intelligent-Tiering
- Amazon EBS I/O 特性
- 使用 Amazon S3 儲存類別
- 什麼是 Amazon S3 Glacier？

相關影片：

- AWS re:Invent 2023 - 提高 Amazon EBS 效率並更具成本效益
- AWS re:Invent 2023 - 利用 Amazon S3 最佳化儲存價格和效能
- AWS re:Invent 2023 - 在 Amazon S3 上建置資料湖並進行最佳化
- AWS re:Invent 2022 - 在 AWS 上建置現代化資料架構
- AWS re:Invent 2022 - 透過專用資料庫建置現代化應用程式
- AWS re:Invent 2022 - 在 AWS 上建置資料網格架構
- AWS re:Invent 2023 - 深入探索 Amazon Aurora 及其創新
- AWS re:Invent 2023 - 使用 Amazon DynamoDB 的進階資料建模

相關範例：

- Amazon S3 範例
- AWS 專用資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構 Immersion Day
- 在 AWS 上建置資料網格

SUS04-BP03 使用政策來管理資料集的生命週期

管理所有資料的生命週期並自動執行刪除，將工作負載所需的儲存總量降至最低。

常見的反模式：

- 您手動刪除資料。
- 您未刪除任何工作負載資料。
- 您未根據資料的保留和存取要求，將資料移至更節能的儲存層。

建立此最佳實務的優勢：使用資料生命週期政策可確保工作負載中的資料會以有效率的方式存取和保留。

未建立此最佳實務時的風險暴露等級：中

實作指引

資料集在其生命週期內，通常會有不同的保留和存取要求。例如，應用程式可能需要在一段時間內頻繁存取某些資料集。這段時間過後，便不會頻繁存取這些資料集。

為了在資料集的完整生命週期內有效率地管理資料集，請設定生命週期政策，也就是定義了資料集處理方式的規則。

有了生命週期組態規則後，便能指示特定儲存服務將資料集轉移至更節能的儲存層、將其封存，或加以刪除。

實作步驟

- 對工作負載內的資料集進行分類。
- 定義每個資料類別的處理程序。Define handling procedures for each data class.
- 設定自動生命週期政策以強制執行生命週期規則。下面幾個範例會說明如何為不同的 AWS 儲存服務設定自動化的生命週期政策：

Storage service	How to set automated lifecycle policies
Amazon S3	您可以使用 Amazon S3 生命週期，以在物件的整個生命週期中管理物件。如果存取模式不明、會變化或是無法預測，則可以使用 Amazon S3 Intelligent-Tiering，讓其監控存取

Storage service	How to set automated lifecycle policies
	模式，並自動將未存取的物件移至成本較低的存取層。您可以利用 Amazon S3 Storage Lens 指標來識別生命週期管理中的最佳化機會和落差。
Amazon Elastic Block Store	您可以使用 Amazon Data Lifecycle Manager 自動建立、保留和刪除 Amazon EBS 快照與 Amazon EBS 支援的 AMI。
Amazon Elastic File System	Amazon EFS 生命週期管理會自動管理檔案系統的檔案儲存。
Amazon Elastic Container Registry	Amazon ECR 生命週期政策會根據存留時間長短或計數讓映像過期，藉此自動清理容器映像。
AWS Elemental MediaStore	您可以使用物件生命週期政策來管控物件儲存在 MediaStore 容器內的時間長度。

- 請刪除已超過保留期間的未使用磁碟區、快照和資料。利用原生服務功能 (例如Amazon DynamoDB Time To Live 或 Amazon CloudWatch 日誌保留) 來執行刪除作業。
- 根據生命週期規則，在適用的情況下彙總和壓縮資料。

資源

相關文件：

- 使用 Amazon S3 儲存類別分析將 Amazon S3 生命週期規則最佳化
- 使用 AWS Config 規則 評估資源

相關影片：

- AWS re:Invent 2021 - Amazon S3 最佳化儲存支出的生命週期最佳實務
- AWS re:Invent 2023 - 利用 Amazon S3 最佳化儲存價格和效能
- 使用 Amazon S3 生命週期來簡化資料生命週期並將儲存成本最佳化
- 使用 Amazon S3 Storage Lens 降低儲存成本

SUS04-BP04 使用彈性和自動化擴充區塊儲存或檔案系統

隨著資料的增長使用彈性和自動化擴充區塊儲存或檔案系統，以盡可能縮小整體的已佈建儲存。

常見的反模式：

- 您為了日後的需求購買大型區塊儲存或檔案系統。
- 您過度佈建了檔案系統的每秒輸入/輸出作業數 (IOPS)。
- 您未監控資料磁碟區的使用率。

建立此最佳實務的優勢：盡可能減少儲存系統的過度佈建可減少閒置資源，並改善工作負載的整體效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

使用適合工作負載的大小分配、輸送量和延遲，建立區塊儲存和檔案系統。隨著資料的增長使用彈性和自動化擴充區塊儲存或檔案系統，而無須過度佈建這些儲存服務。

實作步驟

- 對於固定大小的儲存體 (例如 Amazon EBS)，請確認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。
- 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。例如，您可以使用 Amazon EBS 彈性磁碟區來變更磁碟區大小、磁碟區類型，或調整 Amazon EBS 磁碟區的效能。
- 為您的檔案系統選擇適當的儲存類別、效能模式和輸送量模式以因應商業需求 (勿過量)。
  - Amazon EFS 效能
  - Linux 執行個體上的 Amazon EBS 磁碟區效能
- 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。
- 根據資料適當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。
- 定期審查彈性磁碟區和檔案系統以終止閒置磁碟區，並縮減過度佈建的資源以符合目前的資料大小。

資源

相關文件：

- 調整 EBS 磁碟區大小後擴展檔案系統
- 使用 Amazon EBS 彈性磁碟區以修改磁碟區
- Amazon FSx 文件
- 什麼是 Amazon Elastic File System？

相關影片：

- 深入探討 Amazon EBS 彈性磁碟區
- 有助於提升效能和節省成本的 Amazon EBS 與快照優化策略
- 使用最佳實務優化 Amazon EFS 的成本與效能

SUS04-BP05 移除不需要或多餘的資料

移除不需要或多餘的資料，以盡量降低儲存資料集時所需的儲存資源。

常見的反模式：

- 您複製可以輕鬆取得或建立的資料。
- 您備份所有資料，而不考慮該資料是否重要。
- 您只會不定期地刪除資料、在發生營運事件時刪除資料，或完全不刪除資料。
- 您重複儲存資料，而不理會儲存服務的耐用性。
- 您在沒有任何商務理由的情況下啟用 Amazon S3 版本控制。

建立此最佳實務的優勢：移除不需要的資料會降低工作負載所需的儲存大小，以及工作負載環境所受到的影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

請勿儲存您不需要的資料。請自動刪除不需要的資料。使用會在檔案層級和區塊層級刪除重複資料的技術。利用服務原生的資料複寫和備援功能。

實作步驟

- 評估您是否可以藉由使用 AWS Data Exchange 和 AWS 上的開放資料登錄檔中現有的公開提供的資料集，以避免儲存資料。

- 使用可在區塊和物件層級刪除重複資料的機制。下面幾個範例會說明如何在 AWS 上刪除重複資料：

Storage service	Deduplication mechanism
Amazon S3	使用 AWS Lake Formation FindMatches，透過新的 FindMatches ML Transform 來尋找整個資料集內的相符記錄。
Amazon FSx	在適用於 Windows 的 Amazon FSx 上使用重複資料刪除。
Amazon Elastic Block Store 快照	快照是增量備份，這表示只會儲存最近一次快照後裝置上發生變更的區塊。

- 分析資料存取以識別不需要的資料。將生命週期政策自動化。利用原生服務功能 (例如 Amazon DynamoDB Time To Live、Amazon S3 Lifecycle 或 Amazon CloudWatch 日誌保留) 來執行刪除作業。
- 使用 AWS 上的資料虛擬化功能將資料留在其來源上，並避免資料重複。
  - AWS 上的雲端原生資料虛擬化
  - 使用 Amazon Redshift 資料共用來優化資料模式
- 使用可以進行增量備份的備份計數。
- 利用 Amazon S3 的耐久性和 Amazon EBS 的複寫功能來滿足耐久性目標，而非利用自我管理的技術 (例如獨立硬碟冗餘陣列 (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：

- 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
- 使用 AWS Lake Formation 整合及刪除重複資料集

相關影片：

- Amazon Redshift 資料共享使用案例

相關範例：

- 我要如何使用 Amazon Athena 分析 Amazon S3 伺服器存取日誌？

SUS04-BP06 使用共用檔案系統或儲存體存取通用資料

採用共用檔案系統或儲存體以避免資料重複，並且讓工作負載有更高效的基礎設施。

常見的反模式：

- 您為每個用戶端佈建儲存體。
- 您未從非作用中用戶端卸離資料磁碟區。
- 您未提供跨平台和系統的儲存體存取。

建立此最佳實務的優勢：使用共用檔案系統或儲存裝置，無需複製資料即可與一或多個取用者共用資料。這有助於減少工作負載所需的儲存資源。

未建立此最佳實務時的風險暴露等級：中

實作指引

如果有多個使用者或應用程式在存取相同的資料集，則務必使用共用儲存技術，讓您的工作負載有高效的基礎設施。共用儲存技術提供了集中儲存和管理資料的位置，可避免資料重複。此外也可強制執行跨不同系統的資料一致性。再者，共用儲存技術可讓您更有效率地使用運算能力，因為多個運算資源可同時平行存取及處理資料。

請在必要時才從這些共用儲存服務擷取資料，且應卸離未使用的磁碟區以釋出資源。

實作步驟

- 當資料有多個取用者時，將資料遷移到共用儲存體。以下是 AWS 共用儲存技術的一些範例：

Storage option	When to use
Amazon EBS Multi-Attach	Amazon EBS Multi-Attach 可讓您將單一佈建 IOPS SSD (io1 或 io2) 磁碟區連接至位於相同可用區域中的多個執行個體。
Amazon EFS	請參閱何時應選擇 Amazon EFS。
Amazon FSx	請參閱選擇 Amazon FSx 檔案系統。
Amazon S3	不需要檔案系統結構、且設計為使用物件儲存的應用程式，可使用 Amazon S3 作為可大規模擴展、耐久、低成本的物件儲存解決方案。

- 有需要時才將資料複製到共用檔案系統或從中擷取資料。例如，您可以建立由 Amazon FSx for Lustre 支援的 Amazon S3 檔案系統，並僅將處理任務所需的資料子集載入至 Amazon FSx。
- 根據您的使用模式適當刪除資料，如 SUS04-BP03 使用政策來管理資料集的生命週期 中所述。
- 將磁碟區與未積極使用它們的用戶端分開。

資源

相關文件：

- 將檔案系統連結至 Amazon S3 儲存貯體
- 在無伺服器應用程式中對 AWS Lambda 使用 Amazon EFS
- Amazon EFS Intelligent-Tiering 優化變更存取模式的工作負載成本
- 將 Amazon FSx 用於內部部署資料儲存庫

相關影片：

- 透過 Amazon EFS 將儲存成本優化
- AWS re:Invent 2023：AWS 檔案儲存最新消息
- AWS re:Invent 2023 - Amazon Elastic File System 上適用於建構者和資料科學家的檔案儲存

SUS04-BP07 盡可能減少跨網路的資料移動

使用共用檔案系統或物件儲存體存取通用資料，將支援工作負載資料移動所需的整體網路資源降至最低。

常見的反模式：

- 無論資料使用者位於何處，您都將所有資料儲存在相同的 AWS 區域 中。
- 您未優化資料大小和格式即將其移至網路。

建立此最佳實務的優勢：優化整個網路間的資料移動，可減少工作負載所需的整體網路資源，並降低其環境影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

要在您的組織移動資料，需要運算、聯網和儲存資源。使用相關技術盡可能減少資料移動，並改善工作負載的整體效率。

實作步驟

- 選取工作負載的區域時，可將區域與資料或使用者的鄰近性視為決策因素。
- 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。
- 使用有效率的檔案格式 (例如 Parquet 或 ORC)，並在透過網路移動資料之前先壓縮資料。
- 請勿移動未使用的資料。一些有助於避免移動未使用資料的範例：
  - 減少 API 回應 (僅回應相關資料)。
  - 彙總詳細的資料 (不需要記錄層級資訊)。
  - 請參閱 Well-Architected 實驗室 - 使用 Amazon Redshift 資料共用來優化資料模式
  - 考慮在 AWS Lake Formation 中使用跨帳戶資料共用。
- 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。

Service	When to use
Lambda@Edge	用於在物件未經快取時執行的大量運算作業。
CloudFront Functions	處理簡單的使用案例，例如可由短期函數起始的 HTTP(s) 請求/回應操作。

Service	When to use
AWS IoT Greengrass	為連線的裝置執行本機運算、傳訊和資料快取。

資源

相關文件：

- 優化您的 AWS 永續性基礎設施，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 要求
- 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

相關影片：

- 揭密 AWS 上的資料傳輸

相關範例：

- 永續性架構 - 盡可能減少跨網路的資料移動

SUS04-BP08 僅在難以重新建立時才備份資料

避免備份沒有商業價值的資料，以盡可能降低工作負載的儲存資源需求。

常見的反模式：

- 您沒有資料的備份策略。
- 您備份了可輕易重新建立的資料。

建立此最佳實務的優勢：避免備份非關鍵資料可減少工作負載所需的儲存資源，並降低其環境影響。

未建立此最佳實務時的風險暴露等級：中

實作指引

避免備份非必要的資料，有助於降低成本和工作負載所使用的儲存資源。僅備份具有商業價值或需要滿足合規要求的資料即可。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。

實作步驟

- 實作如 SUS04-BP01 實作資料分類政策 所列的資料分類政策。
- 根據您的復原時間目標 (RTO) 和復原點目標 (RPO) 使用資料分類的關鍵性，並設計備份策略。避免備份非關鍵資料。
  - 排除可輕易重新建立的資料。
  - 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。
- 使用自動化解決方案或受管服務來備份業務關鍵資料。
  - AWS Backup 是一種全受管服務，可讓您在雲端和內部部署環境輕鬆集中化和自動化 AWS 服務的資料保護。如需如何使用 AWS Backup 建立自動化備份的實作指引，請參閱 Well-Architected 實驗室 - 測試備份並還原資料。
  - 使用 AWS Backup 自動進行 Amazon EFS 的備份及優化備份成本。

資源

相關的最佳實務：

- 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-BP03 自動執行資料備份
- REL13-BP02 使用定義的復原策略達到復原目標

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
- 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
- 備份 Amazon EFS
- 備份 Amazon FSx for Windows File Server
- Amazon ElastiCache for Redis 的備份和還原

相關影片：

- AWS re:Invent 2023 - 提升恢復能力的備份與災難復原策略
- AWS re:Invent 2023 - AWS Backup 最新消息
- AWS re:Invent 2021 - 使用 AWS 進行備份、災難復原和勒索軟體防護

相關範例：

- Well-Architected 實驗室 - 備份資料

硬體和服務

問題

- SUS 5 如何選取雲端硬體和服務並在您的架構中使用，以支持您的永續性目標？

SUS 5 如何選取雲端硬體和服務並在您的架構中使用，以支持您的永續性目標？

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取最高效率的硬體和服務。

最佳實務

- SUS05-BP01 使用最低數量的硬體來滿足需求
- SUS05-BP02 使用影響最小的執行個體類型
- SUS05-BP03 使用受管服務
- SUS05-BP04 將硬體型運算加速器的使用方式優化

SUS05-BP01 使用最低數量的硬體來滿足需求

使用最低數量的硬體讓您的工作負載有效達成商業需求。

常見的反模式：

- 您未監控資源使用率。
- 您的架構中有資源處於低使用率水準。
- 您未審查靜態硬體的使用率以判斷是否應調整其大小。
- 您未根據業務 KPI 設定運算基礎設施的硬體使用率目標。

建立此最佳實務的優勢：適當調整雲端資源大小有助於降低工作負載的環境影響、節省金錢並維護效能基準。

未建立此最佳實務時的風險暴露等級：中

實作指引

建議選取您的工作負載所需的硬體總數，以改善其整體效率。AWS 雲端 提供的彈性可透過各種機制 (例如 AWS Auto Scaling) 動態擴充或減少資源數量，以因應需求的變化。此外也提供 API 和 SDK，讓修改資源變得非常輕鬆。使用這些功能可以頻繁變更工作負載實作。此外，使用 AWS 工具提供的適當調整大小指導方針可讓您有效操作雲端資源，而達到您的商業需求。

實作步驟

- 選擇執行個體類型：選擇最適合您的需求的執行個體類型。若要了解如何選擇 Amazon Elastic Compute Cloud 執行個體以及使用屬性型執行個體選擇等機制，請參閱下列內容：
  - 如何為工作負載選擇適當的 Amazon EC2 執行個體類型？
  - Amazon EC2 機群的屬性型執行個體類型選取。
  - 使用屬性型執行個體類型選取建立 Auto Scaling 群組。
- 擴展：使用小增量來擴展變數工作負載。
- 使用多種運算購買選項：利用多種運算購買選項平衡執行個體彈性、可擴展性和成本節省。
  - Amazon EC2 隨需執行個體最適用於新的有狀態尖峰工作負載 (其執行個體類型、位置或時間不具彈性)。
  - Amazon EC2 Spot 執行個體很適合用來補強具有容錯能力和彈性的應用程式適用的其他選項。
  - 對於狀態穩定、允許隨著您的需求變更保有彈性 (例如 AZ、區域、執行個體系列或執行個體類型) 的工作負載，請使用 Compute Savings Plans。
- 使用執行個體和可用區域多樣性：利用多樣化執行個體和可用區域來最大化應用程式可用性，並善用多餘容量。

- 適當調整執行個體大小：使用 AWS 工具提供的適當調整大小建議，調整您的工作負載。如需詳細資訊，請參閱利用精簡化建議將您的成本最佳化和適當調整大小：佈建執行個體以符合工作負載
  - 您可以使用 AWS Cost Explorer 或 AWS Compute Optimizer 中的適當調整大小建議，找出適當調整大小的機會。
- 協商服務水準協議 (SLA)：協商允許暫時減少容量的 SLA，同時自動化部署替換資源。

資源

相關文件：

- 優化您的 AWS 永續性基礎設施，第 I 部分：運算
- Amazon EC2 機群的 Auto Scaling 屬性型執行個體類型選取
- AWS Compute Optimizer 文件
- 操作 Lambda：效能優化
- Auto Scaling 文件

相關影片：

- AWS re:Invent 2023 - Amazon EC2 最新消息
- AWS re:Invent 2023 - 智慧型節約：Amazon Elastic Compute Cloud 成本最佳化策略
- AWS re:Invent 2022 - 在 AWS 上針對效能和成本最佳化 Amazon Elastic Kubernetes Service
- AWS re:Invent 2023 - 永續性運算：利用 AWS 降低成本和碳排放量

SUS05-BP02 使用影響最小的執行個體類型

持續監控並使用新的執行個體類型，讓能源效率方面的改進充分發揮效用。

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。

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風險暴露等級：中

實作指引

在雲端工作負載中使用高效執行個體，是降低資源用量和提高成本效益的關鍵。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 (例如機器學習訓練和推論以及影片轉碼) 的執行個體類型。

實作步驟

- 了解並探索執行個體類型：尋找可降低工作負載對環境之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 執行個體與深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體。
- 使用影響最小的執行個體類型：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。如需有關將工作負載移至 AWS Graviton 的詳細資訊，請參閱 AWS Graviton Fast Start 和將工作負載轉移至基於 AWS Graviton 的 Amazon Elastic Compute Cloud 執行個體時的注意事項。
  - 考慮選取 AWS Graviton 選項 (在您使用的 AWS Managed Services 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。
  - 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。 AWS Inferentia 執行個體 (例如 Inf2 執行個體) 所提供的效能功耗比最多會比同類 Amazon EC2 執行個體高出 50%。
  - 使用 Amazon SageMaker Inference Recommender 適當調整 ML 推論端點的大小。

  - 對於尖峰工作負載 (不常需要額外容量的工作負載)，請使用爆量效能執行個體。
  - 對於無狀態和容錯工作負載請使用 Amazon EC2 Spot 執行個體，以提高雲端整體使用率，並減少未使用資源的永續性影響。
- 操作和優化：操作和優化工作負載執行個體。
  - 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。
    - Well-Architected 實驗室 - 適當調整大小的建議
    - Well-Architected 實驗室：使用 Compute Optimizer 適當調整大小
    - Well-Architected 實驗室 - 優化硬體模式和觀察永續性 KPI

資源

相關文件：

- 優化您的 AWS 永續性基礎設施，第 I 部分：運算
- AWS Graviton
- Amazon EC2 DL1
- Amazon EC2 容量保留機群
- Amazon EC2 Spot 機群
- 函數：Lambda 函數組態
- Amazon EC2 機群的屬性型執行個體類型選取
- 在 AWS 上建置永續性、高效且成本優化的應用程式
- Contino 永續性儀表板如何協助客戶優化其碳足跡

相關影片：

- AWS re:Invent 2023 - AWS Graviton：AWS 工作負載的最佳性價比
- AWS re:Invent 2023 - 在 AWS Management Console 中新建 Amazon Elastic Compute Cloud 生成式 AI 功能
- AWS re:Invent 2023 - Amazon Elastic Compute Cloud 的最新消息
- AWS re:Invent 2023 - 智慧型節約：Amazon Elastic Compute Cloud 成本優化策略

- AWS re:Invent 2021 - 深入探討 AWS Graviton3 和 Amazon EC2 C7g 執行個體
- AWS re:Invent 2022 - 建置具成本、能源和資源效率的運算環境

相關範例：

- 在 AWS 上優化深度學習工作負載以維持永續性的指引
- 將 Amazon Relational Database Service 資料庫遷移到 Graviton

SUS05-BP03 使用受管服務

使用受管服務以提高雲端中的操作效率。

常見的反模式：

- 您使用具有低使用率的 Amazon EC2 執行個體來執行應用程式。
- 您的內部團隊僅管理工作負載，而沒有時間專注於創新或簡化。
- 您為在受管服務上可更高效執行的任務部署及維護技術。

建立此最佳實務的優勢：

- 使用受管服務可將責任轉移給 AWS，他們擁有從數百萬客戶積累而成的洞察，可帶來新的創新和效率動能。
- 基於多租用戶控制平面，受管服務將服務產生的環境影響分散到眾多使用者。

未建立此最佳實務時的風險暴露等級：中

實作指引

受管服務可將維持已部署硬體的高使用率和永續性優化的責任轉移給 AWS。受管服務也免除了維護服務的營運和管理重擔，讓您的團隊有更多時間可專注於創新。

審查您的工作負載，以識別可能被 AWS 受管服務取代的元件。例如，Amazon RDS、Amazon Redshift 和 Amazon ElastiCache 提供受管分析服務。Amazon Athena、Amazon EMR 和 Amazon OpenSearch Service 提供受管分析服務。

實作步驟

1. 清查工作負載：清查工作負載中的服務和元件。

2. 確定候選項：評估並識別可能被受管服務取代的元件。以下舉例說明您可能會考慮使用受管服務的時機：

Task	What to use on AWS
託管資料庫	使用受管 Amazon Relational Database Service (Amazon RDS) 執行個體，而非在 Amazon Elastic Compute Cloud (Amazon EC2) 上維護您自己的 Amazon RDS 執行個體。
託管容器工作負載	使用 AWS Fargate，而非實作您自己的容器基礎設施。
託管 Web 應用程式	使用 AWS Amplify 託管，供靜態網站和伺服器端轉譯的 Web 應用程式作為全受管 CI/CD 和託管服務。

3. 建立遷移計畫：識別相依項並建立遷移計畫。據以更新執行手冊和程序手冊。
    - AWS Application Discovery Service 會自動收集並提供有關應用程式相依性和利用率的詳細資訊，協助您在計劃遷移時做出更明智的決定。
4. 執行測試：在遷移至受管服務之前先測試服務。
5. 取代自我託管服務：使用遷移計畫將自我託管服務取代為受管服務。
6. 監控和調整：在遷移完成後繼續監控服務，以便在必要時進行調整及優化服務。

資源

相關文件：

- AWS 雲端 產品
- AWS 總體擁有成本 (TCO) 計算器
- Amazon DocumentDB
- Amazon Elastic Kubernetes Service (EKS)
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)

相關影片：

- AWS re:Invent 2021 - 使用 AWS Managed Services 大規模進行雲端操作
- AWS re:Invent 2023 - 在 AWS 上操作的最佳實務

SUS05-BP04 將硬體型運算加速器的使用方式優化

將加速運算執行個體的使用方式優化，以降低工作負載的實體基礎設施需求。

常見的反模式：

- 未監控 GPU 使用率。
- 針對工作負載使用一般用途的執行個體，但專用執行個體可以提供更高的效能、較低的成本，以及更優異的效能功耗比。
- 您使用硬體型運算加速器來執行任務，但使用 CPU 型運算加速器來執行時會更有效率。

建立此最佳實務的優勢：藉由將硬體型加速器的使用方式優化，您可以降低工作負載的實體基礎設施需求。

未建立此最佳實務時的風險暴露等級：中

實作指引

如果需要高處理能力，使用加速運算執行個體可讓您獲得好處，因為其可讓您存取硬體型運算加速器，例如圖形處理單元 (GPU) 和現場可程式化邏輯閘陣列 (FPGA)。這些硬體加速器在執行某些功能 (例如圖形處理或資料模式比對) 時，會比 CPU 型加速器更有效率。許多加速工作負載 (例如轉譯、轉碼和機器學習) 在資源用量方面極為變化不定。只在需要時執行此硬體，不需要時便將其自動除役，以將資源消耗降至最低。

實作步驟

- 識別哪些加速運算執行個體可以滿足您的要求。
- 針對機器學習工作負載，請利用專供工作負載使用的專用硬體，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。AWS Inferentia 執行個體 (例如 Inf2 執行個體) 所提供的效能功耗比最多會比同類 Amazon EC2 執行個體高出 50%。
- 請收集加速運算執行個體的用量指標。例如，您可以使用 CloudWatch 代理程式來收集 GPU 的 `utilization_gpu` 和 `utilization_memory` 等指標，如使用 Amazon CloudWatch 收集 NVIDIA GPU 指標中所示。
- 將硬體加速器的程式碼、網路運作和設定優化，以確保系統會充分利用基礎硬體。

  - 將 GPU 設定優化
  - Deep Learning AMI 中的 GPU 監控和優化
  - 將 I/O 優化以針對 Amazon SageMaker 中的深度學習訓練來調校 GPU 效能
- 使用最新的高效能程式庫和 GPU 驅動程式。
- 使用自動化來釋出不使用的 GPU 執行個體。

資源

相關文件：

- 加速運算
- 開始建構吧！ 使用自訂晶片和加速器來進行建構
- 如何為工作負載選擇適當的 Amazon EC2 執行個體類型？
- Amazon EC2 VT1 執行個體
- 選擇最佳的 AI 加速器和模型編譯來以 Amazon SageMaker 推斷電腦視覺

相關影片：

- AWS re:Invent 2021 - 如何為深度學習選取 Amazon EC2 GPU 執行個體
- AWS 線上技術講座 - 部署具成本效益的深度學習推論
- AWS re:Invent 2023 - 搭配 AWS 和 NVIDIA 的尖端 AI
- AWS re:Invent 2022 - [全新推出！] 介紹 AWS Inferentia2 型的 Amazon EC2 Inf2 執行個體
- AWS re:Invent 2022 - 利用 AWS Trainium 加速深度學習和創新
- AWS re:Invent 2022 - 利用 NVIDIA 進行 AWS 深度學習：從訓練到部署

程序和文化

問題

- SUS 6 您的組織程序如何支持您的永續性目標？

SUS 6 您的組織程序如何支持您的永續性目標？

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 採用可快速導入永續性改進的方法
- SUS06-BP02 讓工作負載保持在最新狀態
- SUS06-BP03 提高建置環境的使用率
- SUS06-BP04 使用受管 Device Farm 進行測試

SUS06-BP01 採用可快速導入永續性改進的方法

採用相關方法和程序來驗證潛在改善、盡可能降低測試成本，以及提供小幅改善。

常見的反模式：

- 審查應用程式的永續性是僅需在專案開始時執行一次的任務。
- 您的工作負載已過時，因為發行程序太繁瑣而無法導入資源效率的小幅變更。
- 您沒有改善工作負載以維持永續性的機制。

建立此最佳實務的優勢：建立導入和追蹤永續性改善的程序後，您將可持續採用新的特性和功能、消除問題，並改善工作負載效率。

未建立此最佳實務時的風險暴露等級：中

實作指引

在將潛在永續性改善部署到生產環境之前，先加以測試和驗證。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提供小幅改善。

實作步驟

- 了解並傳達組織永續發展目標：了解您的組織的永續發展目標，例如減碳或水資源管理。將這些目標轉化為雲端工作負載的永續需求。將這些需求傳達給主要利害關係人。
- 將永續發展需求加入到待辦事項：在開發待辦事項中新增永續改善需求。
- 迭代和改善：使用迭代改善程序對這些改善進行識別、評估、優先順序設定、測試及部署。
- 使用最簡可行產品 (MVP) 進行測試：使用最簡可行的代表元件開發並測試可能的改善，以降低測試成本和對環境的衝擊。
- 簡化流程：持續改進並簡化您的開發流程。例如，使用持續整合與持續交付 (CI/CD) 管道自動執行軟體交付流程，以測試及部署可能的改善，進而減少工作量和手動流程導致的錯誤。

- 培訓和認知：為您的團隊成員執行培訓計畫，帶他們了解永續發展及其活動如何影響組織的永續發展目標。
- 評估和調整：持續評估改善的影響，並視需要進行調整。

資源

相關文件：

- AWS 提供永續性解決方案
- 以 AWS CodeCommit 為基礎的可擴展敏捷開發實務

相關影片：

- AWS re:Invent 2023 - 永續架構：過去、現在和未來
- AWS re:Invent 2022 - 提供永續且高效能的架構
- AWS re:Invent 2022 - 永續架構並降低您的 AWS 碳足跡
- AWS re:Invent 2022 - AWS 全球基礎設施的永續性
- AWS re:Invent 2023 - AWS 可觀測性和營運最新消息

相關範例：

- Well-Architected 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

將工作負載保持在最新狀態，以採用高效功能、去除問題，以及改善工作負載的整體效率。

常見的反模式：

- 您假設目前的架構是靜態的，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。

建立此最佳實務的優勢：建立讓工作負載保持在最新狀態的程序後，您將可採用新的特性和功能、解決問題，並改善工作負載效率。

未建立此最佳實務時的風險暴露等級：低

實作指引

最新的作業系統、執行階段、中介軟體、程式庫和應用程式可改善工作負載效率，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。定期以最新的功能和版本將工作負載保持在最新狀態。

實作步驟

- 定義程序：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。
- 執行清查：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 Inventory，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。
- 了解更新程序：了解如何更新工作負載的元件。

Workload component	How to update
機器影像	使用 EC2 Image Builder 管理適用於 Linux 或 Windows Server 映像的 Amazon Machine Image (AMI) 的更新。
容器影像	搭配使用 Amazon Elastic Container Registry (Amazon ECR) 與您現有的管道來管理 Amazon Elastic Container Service (Amazon ECS) 映像。
AWS Lambda	AWS Lambda 包含版本管理功能。

- 使用自動化作業：將更新程序自動化，以減少部署新功能的工作量，並避免手動程序引起的錯誤。
  - 您可以使用 CI/CD 自動更新 AMI、容器映像，以及其他與您的雲端應用程式有關的成品。
  - 您可以使用 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

相關影片：

- AWS re:Invent 2022 - 利用最佳實務指引優化您的 AWS 工作負載
- 所有物件修補程式：AWS Systems 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

提高資源的使用率以開發、測試及建置您的工作負載。

常見的反模式：

- 您以手動方式佈建或終止您的建置環境。
- 您讓建置環境在測試、建置或發行活動以外執行 (例如，在開發團隊成員的非上班時間執行環境)。
- 您為建置環境過度佈建資源。

建立此最佳實務的優勢：藉由提高建置環境的使用率，您將可改善雲端工作負載的整體效率，同時為建置人員配置有效開發、測試和建置所需的資源。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用自動化和基礎設施即程式碼，在需要時啟動建置環境，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。您的測試環境應該會與生產組態近似。不過，請找

機會使用具有高載容量的執行個體類型、Amazon EC2 Spot 執行個體、自動調整規模資料庫服務、容器和無伺服器技術，以根據使用量調整開發和測試容量。將資料量限定為剛好達到測試要求。如果在測試中使用生產資料，請尋求從生產環境共用資料的可能性，而不要移動資料。

實作步驟

- 使用基礎設施做為程式碼：使用基礎設施做為程式碼來佈建您的組建環境。
- 使用自動化：使用自動化來管理開發和測試環境的生命週期，並且讓建置資源發揮最大效益。
- 最大化使用率：利用策略讓開發和測試環境達到最大的使用率。
  - 使用最低可行的代表環境來開發和測試潛在改善。
  - 在情況允許時使用無伺服器技術。
  - 使用隨需執行個體補充開發人員裝置。
  - 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。
  - 根據您的建置任務自動調整建置資源規模。

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？
- 什麼是 AWS CodeBuild？
- AWS 上的 Instance Scheduler

相關影片：

- AWS re:Invent 2023 - AWS 的持續整合與交付

SUS06-BP04 使用受管 Device Farm 進行測試

使用受管 Device Farm 有效測試代表性硬體集上的新功能。

常見的反模式：

- 您在個別實體裝置上手動測試及部署應用程式。
- 您未在真正的實體裝置上使用應用程式測試服務來測試及操作應用程式 (例如 Android、iOS 和 Web 應用程式)。

建立此最佳實務的優勢：使用受管 Device Farm 來測試具備雲端功能的應用程式有許多好處：

- 將有更多功能可用來測試各種裝置上的應用程式。
- 無須再以內部基礎設施進行測試。
- 提供多種裝置類型 (包括較舊且較不熱門的硬體)，因而無須再進行不必要的裝置升級。

未建立此最佳實務時的風險暴露等級：低

實作指引

使用受管 Device Farm 有助於簡化對代表性硬體集上的新功能進行測試的程序。受管 Device Farm 提供多種裝置類型 (包括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。

實作步驟

- 定義測試要求：定義您的測試要求和計畫 (例如測試類型、作業系統和測試排程)。
  - 您可以使用 Amazon CloudWatch RUM 來收集和分析用戶端資料，並研擬您的測試計畫。
- 選取受管 Device Farm：選取可支援測試要求的受管 Device Farm。例如，您可以使用 AWS Device Farm 來測試和了解您的變更對代表性硬體集有何影響。
- 使用自動化：使用自動化和持續整合/持續部署 (CI/CD) 來排程和執行測試。
  - 整合 AWS Device Farm 與您的 CI/CD 管道以執行跨瀏覽器 Selenium 測試
  - 使用 AWS DevOps 和行動服務建置及測試 iOS 和 iPadOS 應用程式
- 審查與調整：持續審查測試結果並進行必要的改進。

資源

相關文件：

- AWS Device Farm 裝置清單
- 檢視 CloudWatch RUM 儀表板

相關範例：

- Android 的 AWS Device Farm 範例應用程式
- iOS 的 AWS Device Farm 範例應用程式
- AWS Device Farm 的 Appium Web 測試

相關影片：

- AWS re:Invent 2023 - 使用 AWS Device Farm 改善行動裝置和 Web 應用程式品質
- AWS re:Invent 2021 - 透過最終使用者洞察與 Amazon CloudWatch RUM 優化應用程式

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。