架構

AWS 建構良好的架構

AWS 建構良好的架構: 架構

Table of Contents

AWS Well-Architected 架構

出版日期：2024 年 6 月 27 日 (文件修訂)

AWS Well-Architected 架構可協助您了解在 上建置系統時所做決策的優缺點 AWS。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 架構可協助您了解在 上建置系統時所做決策的優缺點 AWS。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。其可讓您根據最佳實務以一致的方式來衡量架構，並識別需要改善的區域。檢閱架構的程序是與架構決策相關的建設性討論，而不是稽核機制。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 解決方案架構師在跨各種商業垂直和使用案例架構解決方案方面擁有多年的經驗。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，我們已識別在雲端建構系統的最佳實務和核心策略。

AWS Well-Architected Framework 會記錄一組基礎問題，協助您了解特定架構是否與雲端最佳實務相符。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。隨著 AWS 不斷發展，我們繼續從與客戶合作中進一步了解，我們將繼續完善架構良好的定義。

此架構適用於擔任技術角色的人員，例如技術長（CTOs）、架構師、開發人員和營運團隊成員。它描述了設計和操作雲端工作負載時要使用的 AWS 最佳實務和策略，並提供進一步實作詳細資訊和架構模式的連結。如需詳細資訊，請參閱 AWS Well-Architected 首頁。

AWS 也提供免費檢閱工作負載的服務。AWS Well-Architected Tool（AWS WA Tool） 是雲端中的服務，提供一致的程序，讓您使用 AWS Well-Architected Framework 來檢閱和測量您的架構。 AWS WA Tool 提供建議，讓您的工作負載更可靠、安全、高效且符合成本效益。

為協助您套用最佳實務，我們已建立 AWS Well-Architected 實驗室，它可為您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） 合作夥伴合作，這些合作夥伴是 AWS Well-Architected Partner Program 的成員。這些 AWS 合作夥伴擁有深入 AWS 的知識，可協助您檢閱和改善工作負載。

定義

每天， 的專家 AWS 都會協助客戶建構系統，以利用雲端的最佳實務。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。

根據我們學到的內容，我們建立了 AWS Well-Architected 架構，該架構為客戶和合作夥伴提供一組一致的最佳實務，以評估架構，並提供一組問題供您用來評估架構與 AWS 最佳實務的一致性。

AWS Well-Architected Framework 以六大支柱為基礎：卓越營運、安全性、可靠性、效能效率、成本最佳化和永續性。

表 1. AWS Well-Architected Framework 的支柱

名稱	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	安全支柱描述如何利用雲端技術來保護資料、系統和資產，從而改善您的安全狀態。
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。本白皮書提供在 上實作可靠工作負載的深入最佳實務指南 AWS。
效能效率	能夠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。
成本最佳化	能夠以最低的價格執行系統來提供商業價值。
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 架構中，我們使用下列詞彙：

- 元件是針對 需求一起交付的程式碼、組態 AWS 和資源。一個元件往往是技術擁有的單元，並自其他元件所解偶。
- 術語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。
- 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。這些業務決策可以讓您了解工程設計的優先順序。您可以最佳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最佳化。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。安全和卓越營運通常不會因其他要件而被犧牲。

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、聯網架構師和安全架構師。這些團隊通常會使用 TOGAF或 Zachman 架構作為企業架構功能的一部分。

在 AWS，我們偏好將功能分發給團隊，而不是擁有具有該功能的集中式團隊。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 領導方針的支援，遍及所有角色培養一種返向工作從客戶需求出發的文化。返向工作是我們創新程序的基礎部分。我們從客戶及其期望著手，根據之定義並主導我們的工作方向。以客戶為尊的團隊會因應客戶的需要建置產品。

對架構而言，這表示我們期望每個團隊皆有能力建立架構，並且遵照最佳實務。為了協助新團隊取得這些功能或現有團隊提升其標準，我們會啟用主要工程師的虛擬社群的存取權，這些工程師可以檢閱其設計，並協助他們了解什麼是 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 最佳實務源自於我們以網際網路規模執行數千個系統的經驗。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 CTOs或 開發經理）在所有工作負載中執行 Well-Architected 檢閱，可讓您更了解技術產品組合中的風險。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：如果您在部署工作負載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，並自動擴展和縮減。
- 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。

架構的六大支柱

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
- 可靠性
- 效能效率
- 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。

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：

- 圍繞業務成果組織團隊：團隊實現業務成果的能力來自於領導力願景、高效運營以及與業務保持一致的運營模式。領導力應完全投入並致力於具有適當雲端操作模型的 CloudOps 轉型，以激勵團隊以最

有效率的方式運作並滿足業務成果。正確的營運模式會利用人員、流程和技術能力來擴展、最佳化生產力，並透過敏捷性、回應性和適應性來實現差異化。組織的長期願景轉化為目標，並在整個企業內傳達給雲端服務的利益相關者和消費者。目標和操作在所有層級KPIs都保持一致。這種做法維持了實作下列設計原則所帶來的長期價值。

- 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 （KPIs），並利用可觀測性遙測，做出明智的決策，並在業務結果處於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。
- 盡可能安全地自動化：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相同的工程原則。可以將整個工作負載及其操作 (應用程式、基礎設施、組態和程序) 定義為程式碼，然後進行更新。然後，可以透過回應事件來啟動工作負載操作，從而將其自動化。在雲端中，可以透過設定防護機制來採用自動化安全性，包括速率控制、錯誤閾值和核准。透過有效的自動化，可以實現對事件的一致回應，限制人為錯誤並減少操作員的辛勞。
- 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。這增加了信心，可以為您的工作負載提供有益的變化，同時保持品質並快速適應市場情況的變化。
- 經常改進營運程序：隨著工作負載的進化，適當地發展您的營運。在使用營運程序時，尋找機會予以改善。定期審查並驗證所有程序是否有效以及團隊是否熟悉這些程序。如果發現差距，請相應地更新程序。向所有利益相關者和團隊傳達程序更新。將營運遊戲化以分享最佳實務並教導團隊。
- 預料失敗：透過推動故障情境來了解工作負載的風險狀況及其對業務成果的影響，從而最大程度提高營運成功率。針對這些模擬失敗，測試程序的有效性和團隊的回應。制定明智的決策，以管理您的測試所識別的開放式風險。
- 從所有營運事件和指標中學習：透過從所有營運事件和失敗中學習的經驗來推動改進。跨團隊及在整個組織中分享獲得的經驗。學習應強調有關營運如何為業務成果做出貢獻的資料和軼事。
- 使用 受管服務：盡可能使用 AWS 受管服務來降低營運負擔。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。您的組織必須了解需求和優先順序，並使用這些來組織和執行工作以支援業務成果的實現。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，實作服務以實現工作負載的整合、部署和交付將為生產帶來更多有利的變更。

工作負載的操作本質上就可能存在著風險。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，將允許您了解工作負載的運作狀態、營運活動，並回應事件。您的優先事項會隨著業務需求和業務環境的變化而改變。運用這些方面做為回饋迴圈，以持續改善貴組織和工作負載的運作。

最佳實務

> **Note**
>
> 所有卓越操作問題都有字OPS首作為支柱的簡稱。

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，以便設定能實現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，並讓關鍵利益相關者 (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。評估客戶需求將驗證您對實現業務成果所需的支援有透徹的了解。確保您了解由貴組織管控所定義的、可能要求或強調特定重點的準則或義務以及外部因素，例如法規合規要求和產業標準。確認您是否設有識別內部管控和外部合規要求變更的機制。如果未識別要求，請確保您已對此決定進行盡職調查。定期審查您的優先事項，以便在需求變更時更新優先事項。

評估對業務的威脅 (例如，業務風險和負債以及資訊安全威脅)，並將此資訊保存在風險登記表內。評估風險的影響，以及利益衝突或替代方法之間的權衡。例如，新功能加速上市可能是成本最佳化所強調的重點，或您可為非關聯式資料選擇關聯式資料庫，以簡化系統遷移工作而無需重構。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。團隊必須了解自己在促成其他團隊成功的過程中所扮演的角色、其他團隊在促進其成功的過程中所扮演的角色，以及擁有共同目標。了解責任、擁有權、決策方式，以及誰有權制定決策，將有助於找到工作重點，並充分發揮團隊的優勢。團隊的需求將由其所支援的客戶、組織、團隊組成，以及工作負載的特性形塑而成。合理來說，無法要求單一操作模式支援貴組織中的所有團隊及其工作負載。

驗證每個應用程式、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。明確界定團隊成員的責任，以便他們可以採取適當行動，並具有識別責任和擁有權的機制。擁有請求增加、變更和例外狀況的機制，這樣您就不會限制創新。定義團隊之間的協議，說明他們如何協同合作以互相支援並實現業務成果。

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。當成果出現風險時，讓團隊成員採取行動，將影響降到最低，同時鼓勵他們在遇到風險時，向決策者和利益相關者呈報，以便處理問題並避免事故。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。驗證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，以便他們可以判斷對您的優先事項的影響。Well-Architected 架構強調學習、衡量和改善。它提供一致的方法來評估架構，並實作會隨時間擴展的設計。 AWS 提供 AWS Well-Architected Tool ，協助您在開發前檢閱方法、生產前的工作負載狀態，以及生產中工作負載的狀態。您可以比較工作負載與最新的 AWS 架構最佳實務、監控其整體狀態，並深入了解潛在風險。 AWS Trusted Advisor 是一項工具，可讓您存取一組核心檢查，建議可能有助於調整優先順序的最佳化。商業和企業支援客戶可存取針對安全性、可靠性、效能、成本優化以及永續性的其他檢查，從而進一步協助確定他們的優先事項。

AWS 可協助您教育團隊有關 AWS 及其 服務，以進一步了解其選擇如何影響您的工作負載。使用 AWS Support （ AWS 知識中心、 AWS 討論論壇和 AWS Support 中心 ） 和 AWS 文件提供的資源來教育您的團隊。透過 AWS Support AWS Support Center 取得 AWS 問題協助。 AWS 也會分享我們在 Amazon Builders' Library AWS 中的 操作所學到的最佳實務和模式。部落格 AWS 和官方 AWS

Podcast 提供了各種其他有用的資訊。 AWS 訓練和認證透過有關 AWS 基礎知識的自定進度數位課程提供一些訓練。您也可以註冊講師引導式訓練，以進一步支援團隊 AWS 技能的發展。

使用可讓您集中管理 等帳戶環境的工具或服務 AWS Organizations，以協助管理您的操作模型。例如透過可讓您定義帳戶設定的藍圖 （支援您的操作模型）、使用 套用持續的治理 AWS Organizations，以及自動佈建新帳戶等服務來 AWS Control Tower 擴展此管理功能。 AWS 合作夥伴網路中的 Managed Services Provider，例如 AWS Managed Services、 AWS Managed Services Partners 或 Managed Services Providers，提供實作雲端環境的專業知識，並支援您的安全和合規要求和業務目標。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：如何判斷優先順序？

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

OPS 2：您如何建構組織以支援您的業務成果？

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

OPS 3：您的組織文化如何支援您的業務成果？

為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。

您可能會發現，您在某個時間點會想要強調一小部分的優先事項。長期利用平衡的方法，以驗證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，以實現業務成功。必需要經由試驗才能實現創新，並讓想法轉化為成果。辨識不想要的結果是代表這是一場成功的試驗，因為可以判斷出不會通往成功途徑。

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。基於指標、日誌和追蹤，可觀測性為系統行為和動態提供深刻的見解。透過有效的可觀測性，團隊可以辨別模式、異常情況和趨勢，讓他們能夠主動解決潛在問題並維持最佳的系統運作狀態。識別關鍵績效指標 （KPIs） 至關重要，以確保監控活動與業務目標之間保持一致。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，從而最佳化系統效能和業務成果。此外，可觀測性使企業能夠主動出擊，而不是被動應對。團隊可以了解 cause-and-effect其系統內的關係，預測和預防問題，而不只是對問題做出反應。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進入生產環境的流程，並實現重構、快速品質意見回饋及錯誤修復。這會加快有助益的變更進入生產環境的速度、限制部署問題，並快速識別和修復部署活動所導致或在您的環境中所發現的問題。

採用可快速提供品質意見回饋，並從成果不盡理想的改變中快速復原的方法。使用這些實務可緩解部署變更所帶來問題的影響。為變更失敗做好規劃，以便在必要時能夠快速回應，同時測試並驗證所做變更。了解環境中的計劃內活動，以便管理會影響計劃內活動的變更風險。強調頻繁、細微、可逆的變更，以限制變更範圍。透過回復變更，可以更快地進行疑難排解和修復。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 可讓您將整個工作負載 （應用程式、基礎設施、政策、治理和操作） 檢視為程式碼。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、範本化、沙盒開發、測試和生產環境，並提高操作控制層級。

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：如何減少瑕疵、簡化修復並改善進入生產的流程？

採用可改進生產變更流程的方法，從而重構快速的品質回饋及錯誤修復。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：如何降低部署風險？

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：您如何知道自己已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。使用 Resource Tags 套用中繼資料，並 AWS Resource Groups 遵循一致的標記策略來識別您的資源。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。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。透過專注於基本洞察並消除不必要的資料，可以維持一種簡單的方法來了解工作負載效能。不僅要收集資料，還要正確解譯資料，這至關重要。定義明確的基準，設定適當的警示閾值，並主動監控任何偏差。關鍵指標的變化，特別是與其他資料相關時，可以查明特定的問題區域。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，以確定工作負載和營運是否成功。運作狀態包括工作負載的運作狀態以及為支援工作負載而執行的營運活動的運作狀態和成功情況 (例如，部署和事故回應)。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，必須高效且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眾所周知的事件，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。驗證如因回應事件而發出提醒，則將由明確識別的擁有者執行關聯程序。事先定義解決事件所需的人員，並納入向上呈報程序，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 中 AWS，您可以產生從工作負載和原生從 收集的指標的儀表板檢視 AWS。您可以利用 CloudWatch 或第三方應用程式來彙總和呈現操作活動的業務、工作負載和操作層級檢視。 透過記錄功能 AWS 提供工作負載洞察，包括 AWS X-Ray CloudWatch、 CloudTrail、 和 VPC Flow Logs，以識別工作負載問題，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：如何在組織中運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：您如何了解營運的運作狀態？

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：如何管理工作負載和操作事件？

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。將工作週期用於進行幾乎持續的逐漸改善。針對所有影響客戶的事件執行事件後分析。確定促成因素和預防措施，以限制或防止再次發生。適當地與受影響的社區溝通促成因素。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正在執行的營運中獲得經驗。

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。分析經驗教訓中的趨勢，並對運營指標執行跨團隊回顧性分析，以確定改進的機會和方法。實作旨在帶來改善的變更，並評估結果以判斷是否成功。

在 上 AWS，您可以將日誌資料匯出至 Amazon S3，或直接將日誌傳送至 Amazon S3 以進行長期儲存。使用 AWS Glue，您可以在 Amazon S3 中探索和準備日誌資料以供分析，並將相關聯的中繼資料儲存在 中 AWS Glue Data Catalog。然後，Amazon Athena AWS Glue可透過其與 的原生整合來分析您的日誌資料，並使用標準 進行查詢SQL。使用 Amazon 等商業智慧工具 QuickSight，您可以視覺化、探索和分析資料。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：如何發展營運？
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 和 AWS

白皮書

- 卓越運作支柱

影片

- DevOps 在 Amazon

安全

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。可以在安全支柱白皮書中找到實作指引。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端中有一些原則能協助您強化工作負載的安全：

- 實作強大的身分基礎：實作最低權限的原則，並針對每次與 AWS 資源的互動，以適當的授權強制執行職責分離。集中進行身分管理，旨在消除對長期靜態憑證的倚賴。
- 維持可追蹤性：即時監控、提醒和稽核您環境中發生的動作和變更。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。
- 在所有層級套用安全：透過多個安全控制，套用深度防禦方法。套用至所有層（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）。
- 將安全最佳實務自動化：將基於軟體的安全機制自動化，以提高您安全、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、記號化及存取控制。

- 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。
- 為安全事件作準備：為事故做好萬全準備，建立與您組織的要求吻合的事件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

雲端安全有七個最佳實務領域：

- 安全基礎
- 身分與存取管理
- 偵測
- 基礎設施保護
- 資料保護
- 事件回應
- 應用程式安全

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 共同責任模型可協助採用雲端的組織實現其安全和合規目標。因為 AWS 實體保護支援我們雲端服務的基礎設施，所以身為 AWS 客戶，您可以專注於使用 服務來完成您的目標。 AWS 雲端也提供對安全資料的更多存取，以及回應安全事件的自動化方法。

最佳實務

主題

- 安全
- 身分與存取管理
- 偵測
- 基礎設施保護
- 資料保護

- 事件回應
- 應用程式安全

安全

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，您必須將整體的最佳實務套用到每個安全區域。採用您在組織和工作負載層級於卓越營運中定義的要求和程序，並將其應用於所有區域。

掌握來自 AWS、產業來源和威脅情報的最新建議，可協助您發展威脅模型和控制目標。將安全程序、測試和驗證自動化可讓您擴展安全操作。

在 中 AWS，根據不同工作負載的函數和合規或資料敏感度需求，將不同的工作負載分隔為建議的方法。

身分與存取管理

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 中 AWS， Identity and Access Management AWS （ IAM ） 服務主要支援權限管理，可讓您控制使用者和程式設計存取 AWS 服務和資源。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。您也可以要求強大的密碼實務，例如複雜性等級、避免重複使用，以及強制執行多重要素身分驗證 （ MFA ）。您可以將聯合身分驗證與現有目錄服務一起使用。對於需要系統存取 的工作負載 AWS， IAM 允許透過角色、執行個體設定檔、身分聯合和臨時憑證進行安全存取。

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？

接近操作安全 AWS 工作負載時，您需要管理兩種類型的身分。了解您需要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。

SEC 2：如何管理人員和機器的身分？

人類身分：您的管理員、開發人員、運算子和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些是您組織的成員，或是與您合作的外部使用者，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 資源互動的外部使用者。

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對 AWS 服務提出請求，例如讀取資料。這些身分包括在 Amazon EC2執行個體或 AWS Lambda 函數等 AWS 環境中執行的機器。您也可以為需要存取權的外部各方管理其機器身分。此外，您可能還有 以外的機器 AWS 需要存取您的 AWS 環境。

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 和工作負載的人員 AWS 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。應使用最低權限方法授予使用者存取權，並採用最佳實務，包括密碼要求和MFA強制執行。應使用暫時性和有限權限的憑證執行程式設計存取，包括API呼叫 AWS 服務，例如 發行的憑證 AWS Security Token Service。

如果使用者想要與 AWS 外部互動，則需要程式設計存取權 AWS Management Console。授予程式設計存取權的方式取決於存取 的使用者類型 AWS。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 （在 IAM Identity Center 中管理的使用者）	使用暫時憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 • 對於 AWS CLI，請參閱 使用者指南 中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 AWS Command Line Interface

哪個使用者需要程式設計存取權？	到	By
		• 如需 AWS SDKs、工具和 AWS APIs，請參閱 AWS SDKs和 工具參考指南 中的IAM身分中心身分驗證。
IAM	使用暫時憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請遵循 IAM 使用者指南 中的將臨時憑證與 AWS 資源搭配使用中的指示。
IAM	(不建議使用) 使用長期憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 • 對於 AWS CLI，請參閱 AWS Command Line Interface 使用者指南 中的使用IAM使用者憑證進行驗證。 • 如需 AWS SDKs 和 工具，請參閱 AWS SDKs和 工具參考指南 中的使用長期憑證進行驗證。 • 對於 AWS APIs，請參閱 IAM 使用者指南 中的管理IAM使用者的存取金鑰。

AWS 提供可協助您進行身分和存取管理的資源。若要協助學習最佳實務，請探索我們的實作實驗室，了解如何管理憑證和驗證、控制人員存取以及控制程式設計式存取。

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，以確認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 中 AWS，您可以透過處理日誌、事件和監控來實作偵測控制，允許稽核、自動分析和警示。CloudTrail 日誌、 AWS API呼叫， CloudWatch 並提供警示指標的監控， AWS Config 並提供組態歷史記錄。Amazon GuardDuty 是一種受管威脅偵測服務，可持續監控惡意或未經授權的行為，以協助您保護 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：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，以獲得可見性。請針對安全事件和潛在威脅採取行動，以協助保護工作負載。

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 中 AWS，您可以使用 AWS原生技術或使用 提供的合作夥伴產品和服務，實作具狀態和無狀態封包檢查 AWS Marketplace。您應該使用 Amazon Virtual Private Cloud （ Amazon VPC ） 來建立私有、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：如何保護運算資源？

工作負載中的運算資源需要多層防禦，以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日誌記錄、監控和提醒，對於有效的資訊安全計畫均很重要。

AWS 客戶可以自訂或強化 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，並將此組態保留至不可變的 Amazon Machine Image （AMI）。然後，無論是由 Auto Scaling 啟動還是手動啟動，所有使用此方式啟動的新虛擬伺服器 （執行個體） 都會AMI收到強化的組態。

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 中 AWS，下列實務有助於資料保護：

- 身為 AWS 客戶，您可以完全控制資料。
- AWS 可讓您更輕鬆地加密資料和管理金鑰，包括定期金鑰輪換，這些金鑰輪換可由您輕鬆自動化 AWS 或由您維護。
- 提供了包含重要內容 (例如檔案存取和變更) 的詳細日誌記錄。
- AWS 設計了具有卓越彈性的儲存系統。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 永遠不會啟動區域之間的資料移動。除非您明確使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：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：如何保護靜態資料？

實作多重控制來保護靜態資料，以降低未經授權存取或處理不當的風險。

SEC 9：如何保護傳輸中的資料？

實作多重控制來保護傳輸中的資料，以降低未經授權存取或遺失的風險。

AWS 提供多種方法來加密靜態和傳輸中的資料。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，我們已為 Amazon S3 實作伺服器端加密（SSE），讓您更輕鬆地以加密形式儲存資料。您也可以安排 Elastic Load Balancing 處理整個HTTPS加密和解密程序（通常稱為SSL終止）（）ELB。

事件回應

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，將幫助您確認架構可以適應即時調查和復原。

在 中 AWS，下列實務有助於有效的事件回應：

- 提供了包含重要內容的詳細日誌記錄，例如檔案存取和變更。
- 事件可以自動處理，並啟動工具，透過使用 自動回應 AWS APIs。
- 您可以使用 AWS CloudFormation預先佈建工具和「無塵室」。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：您如何預測、回應及復原事件？

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。

確認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。

應用程式安全

應用程式安全（AppSec）說明您設計、建置和測試所開發工作負載之安全屬性的整體程序。您應該安排組織成員接受適當訓練，了解您的建置與發佈基礎結構的安全屬性，以及應用自動化來識別出安全問題。

將應用程式安全測試作為軟體開發生命週期（SDLC）和發佈後程序的常規部分，有助於驗證您是否具有結構式機制，以識別、修正和防止應用程式安全問題進入生產環境。

您的應用程式開發方法應該在設計、建置與操作工作負載期間納入安全控制。過程當中，可以調整程序，達到持續減少缺陷和最低技術負債。例如，在設計階段中應用威脅建模有助於提早發現設計瑕疵，修正更加簡單，成本節省更多，而不需要等到日後才能進行緩解。

解決瑕疵的成本和複雜性通常較低，您越早在 中SDLC。最簡單的解決問題方法就是別讓問題發生，因此從使用威脅模型開始，有助於您專注在設計階段的正確成果。隨著 AppSec 程式的成熟，您可以增加使用自動化執行的測試量、提高對建置者的意見回饋保真度，並減少安全審查所需的時間。這些動作全都可以改善所建置軟體的品質，並且加快功能進入生產階段。

這些實作準則著重於四個領域：組織與文化、管道的安全性、管道中的安全性以及相依性管理。每個區域都提供一組原則，您可以實作這些原則， end-to-end並提供如何設計、開發、建置、部署和操作工作負載的檢視。

在 中 AWS，您可以在解決應用程式安全程式時採用多種方法。當中有一些方法依賴技術，而其他方法則著重在應用程式安全計畫的人員和組織層面。

下列問題著重於這些應用程式安全方面的考量。

SEC 11：如何在整個設計、開發和部署生命週期中整合和驗證應用程式的安全屬性？

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。

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
- AWS 安全成熟度模型

白皮書

- 安全支柱
- AWS 安全性概觀
- AWS 風險與合規

影片

- AWS 工會的安全狀態
- 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。本白皮書提供在 上實作可靠工作負載的深入最佳實務指南 AWS。

可靠性支柱概述了設計原則、最佳實務和相關問題。可以在可靠性支柱白皮書中找到實作指引。

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：透過監控關鍵效能指標（KPIs）的工作負載，您可以在違反閾值時啟動自動化。這些KPIs應該是商業價值的指標，而不是服務操作的技術層面。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正的失敗路徑，從而降低風險。
- 水平擴展，以增加彙總工作負載的可用性：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，以確認其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更有效水平，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 (請參閱管理服務配額和限制)。
- 透過自動化管理變更：應透過自動化來執行對基礎架構的變更。必須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
- 故障管理

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

最佳實務

主題

基礎

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

使用 AWS時，大多數的基礎需求都已納入或可視需要加以解決。雲端的設計幾乎是無限的，因此 的責任是 AWS 滿足足夠的聯網和運算容量的需求，允許您隨需變更資源大小和配置。

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：如何管理Service Quotas和限制？

雲端型工作負載架構具有服務配額 (也稱為服務限制)。這些配額的存在是為了防止意外佈建超過您需要的資源，並限制API操作的請求率，以防止服務遭到濫用。另外還有一些資源限制，例如可將位元下推到光纖纜線的速率，或實體磁碟的儲存量。

REL 2：如何規劃網路拓撲？

工作負載通常存在於多個環境中。其中包括多個雲端環境 (可公開存取與私有)，也可能包含您現有的資料中心基礎設施。這些計畫必須包含網路考量事項，例如系統內部與系統間連線能力、公有 IP 位址管理、私有 IP 位址管理以及網域名稱解析。

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。

透過 AWS，工作負載開發人員可以選擇使用的語言和技術。 AWS SDKs 透過為 AWS 服務提供特定語言，來降低編碼APIs的複雜性。這些 SDKs加上語言選擇，可讓開發人員實作此處列出的可靠性最佳實務。開發人員也可在 Amazon 建置者資料中心中閱讀並學習 Amazon 如何構建和操作軟體。

下列問題著重於可靠性方面的這些考量。

REL 3：如何設計工作負載服務架構？

使用服務導向架構（SOA）或微服務架構建置高度可擴展且可靠的工作負載。服務導向架構（SOA）是透過服務介面讓軟體元件可重複使用的實務。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互動以防止失敗？

分散式系統仰賴通訊網路將伺服器或服務等元件互相連線。儘管這些網路中出現資料遺失或延遲，但工作負載仍必須可靠地運作。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故障並改善故障之間的平均時間（MTBF）。

REL 5：如何設計分散式系統中的互動，以減輕或承受故障？

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。結果是改善復原的平均時間（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，並自動回應 KPIs。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？

日誌和指標是可深入洞察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的變化？

可擴展的工作負載提供可自動新增或移除資源的彈性，讓資源能夠在任何特定時間點充分滿足目前的需求。

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，那麼就很難預測這些變更的影響，也很難解決由於這些變更而產生的問題。

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，而且還能驗證企業成功不會成為負擔。在監控到位後，您的團隊在KPIs偏離預期規範時會自動收到提醒。自動日誌記錄對環境的變更，允許您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。

故障管理

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。

透過 AWS，您可以利用自動化來回應監控資料。例如，當特定指標超過閾值時，您可以啟動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：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（ ）的需求RPO。

REL 10：如何使用故障隔離來保護工作負載？

錯誤隔離界限可將工作負載的故障影響限制在有限數量的元件內。界限外部的元件不會受到故障影響。您可以使用多個錯誤隔離界限來限制對工作負載的影響。

REL 11：如何設計工作負載來承受元件故障？

需要高可用性和低平均復原時間（MTTR）的工作負載必須進行彈性架構。

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，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。

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。

定期備份資料並測試備份檔案，從而確認您可以從邏輯和實際錯誤復原。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，然後觀察其可如何復原。定期執行此操作，並確認在出現重大工作負載變更後也能啟動此類測試。主動追蹤 KPIs，以及復原時間目標（RTO）和復原點目標（RPO），以評估工作負載的復原能力（特別是在故障測試情況下）。追蹤KPIs將協助您識別和緩解單一故障點。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：擴展計劃的運作方式
- 什麼是 AWS Backup？

白皮書

- 可靠性支柱： AWS Well-Architected
- 在 上實作 Microservices AWS

效能效率

效能效率支柱包括能夠有效率地使用雲端資源，以滿足效能需求，並隨著需求變更與技術發展來保持該效率需求。

效能達成效率支柱概述了設計原則、最佳實務和相關問題。可以在效能達成效率支柱白皮書中找到實作指引。

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，沒有任何SQL資料庫、媒

體轉碼和機器學習都是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。

- 幾分鐘內即可全球化：在全球多個 AWS 區域中部署工作負載可讓您以最低成本為客戶提供更低的延遲和更好的體驗。
- 使用無伺服器架構：採用無伺服器架構，您便無需執行和維護實體伺服器來完成傳統運算活動。例如，無伺服器儲存服務可以充當靜態網站 (因此無需 Web 伺服器)，而事件服務可以為您託管程式碼。如此一來，即可減輕管理實體伺服器的營運負擔，而且由於這些受管服務是在雲端規模上運行，因此還可以降低交易成本。
- 提高試驗頻率：使用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：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：

- 架構選取
- 運算與硬體
- 資料管理
- 聯網與內容交付
- 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。

定期檢閱您的選擇可驗證您是否利用不斷發展的 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

- 聯網與內容交付
- 程序和文化

架構選擇

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

AWS 資源提供多種類型和組態，讓您更輕鬆地找到最符合您需求的方法。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，諸如 Amazon DynamoDB 的受管服務提供完全受管的無SQL資料庫，在任何規模下都具有單位數毫秒延遲。

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

PERF 1：如何為工作負載選取適當的雲端資源和架構模式？

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

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 中 AWS，運算有三種形式：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中，這些虛擬伺服器執行個體具有不同的系列和大小，且提供廣泛的功能，包括固態硬碟（SSDs）和圖形處理單元（）GPUs。
- 容器是一種作業系統虛擬化方法，可讓您在資源隔離的程序中執行應用程式及其相依性。如果您需要控制運算環境的安裝、組態和管理，EC2則可以使用容器或 Amazon 的無 AWS Fargate 伺服器運算。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）。
- 函數則從您想套用的程式碼中將執行環境抽象化。例如， AWS Lambda 允許您在不執行執行個體的情況下執行程式碼。

下列問題著重於效能達成效率方面的這些考量。

PERF 2：如何在工作負載中選取和使用運算資源？

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

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）、必要的輸送量、存取頻率（線上、離線、封存）、更新頻率（WORM、動態），以及可用性和耐久性限制而有所不同。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 中 AWS，儲存有三種形式：物件、區塊和檔案：

- 物件儲存提供可擴展且耐用的平台，以利從任何網際網路位置存取資料，例如使用者產生的內容、作用中存檔、無伺服器運算、大數據儲存或備份與復原。Amazon Simple Storage Service (Amazon S3) 是一項物件儲存服務，提供領先業界的可擴展性、資料可用性、安全性和效能。Amazon S3 旨在提供 99.999999999% 的耐久性，並為全球公司存放數百萬個應用程式的資料。
- 區塊式儲存為每個虛擬主機提供高可用性、一致、低延遲的區塊式儲存，類似於直接連接的儲存（DAS）或儲存區域網路（）SAN。Amazon Elastic Block Store（Amazon EBS）專為需要EC2執行個體可存取持久性儲存的工作負載而設計，可協助您調整具有適當儲存容量、效能和成本的應用程式。
- 檔案儲存可讓您跨多個系統存取共用檔案系統。Amazon Elastic File System（Amazon EFS）等檔案儲存解決方案非常適合使用案例，例如大型內容儲存庫、開發環境、媒體存放區或使用者主目錄。Amazon FSx讓啟動和執行熱門檔案系統更有效率且符合成本效益，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：如何在工作負載中儲存、管理和存取資料？

系統更有效率的儲存解決方案會根據存取操作類型（區塊、檔案或物件）、存取模式（隨機或循序）、必要的輸送量、存取頻率（線上、離線、封存）、更新頻率（WORM、動態），以及可用

PERF 3：如何在工作負載中儲存、管理和存取資料？

性和耐久性限制而有所不同。Well-Architected 系統使用多重儲存解決方案，並開啟不同功能以提升效能並有效使用資源。

聯網與內容交付

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 上 AWS，聯網是虛擬化的，並提供多種不同的類型和組態。這可讓您更輕鬆地符合聯網需求。AWS 提供產品功能（例如增強型網路、Amazon EC2 網路最佳化執行個體、Amazon S3 傳輸加速和動態 Amazon CloudFront）來最佳化網路流量。 AWS 也提供聯網功能（例如 Amazon Route 53 延遲路由、Amazon VPC端點 AWS Direct Connect和 AWS Global Accelerator），以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：如何在工作負載中選取和設定聯網資源？

此問題包括在雲端中設計、設定和操作高效聯網和內容交付解決方案的指引和最佳實務。

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協助您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：

- 基礎設施為程式碼：使用 AWS CloudFormation 範本等方法將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) 來部署您的基礎架構。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：設定和監控指標，以擷取關鍵績效指標（KPIs）。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，關鍵指標正在擷取 time-to-first-byte或轉譯。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，在成功通過快速執行測試之後，會自動啟動效能測試。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。這些指令碼應該是冪等及非耦合的形式，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 software-as-a-service （ SaaS ） 解決方案來產生負載。可以考慮使用 AWS Marketplace 解決方案和 Spot 執行個體 — 這些可能是經濟實惠的負載產生方式。
- 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。
- 持續優化：培養文化以持續優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：您使用什麼程序來支援工作負載的更多效能效率？

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。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。

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：領導工作階段：工會的儲存狀態 STG2（01-L）
- AWS re：Invent 2019：領導工作階段： AWS 用途建置的資料庫 DAT2（09-L）
- AWS re：Invent 2019：與 AWS 混合 AWS 網路架構的連線能力 （NET317-R1）
- AWS re：Invent 2019：推動新一代 AmazonEC2：深入探索 Nitro 系統 （CMP303-R2）
- AWS re：Invent 2019：擴展到前 1，000 萬使用者 （ARC211-R）

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。

成本最佳化支柱概述了設計原則、最佳實務和相關問題。您可以在成本最佳化支柱白皮書中找到有關實作的規定性指引。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

雲端成本最佳化有五個設計原則：

- 實作雲端財務管理：為了達成財務成功並加速在雲端實現商業價值，請投資雲端財務管理和成本最佳化。您的組織應投入時間和資源，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。
- 停止在未區分的繁重負載上花費金錢： AWS 會繁重的資料中心操作，例如機架、堆疊和驅動伺服器。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：雲端可讓您輕鬆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。這有助於測量投資報酬率 （ROI），並讓工作負載擁有者有機會最佳化資源並降低成本。

定義

雲端成本最佳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
- 隨時間最佳化

與 Well-Architected Framework 中的其他支柱一樣，需要考慮權衡，例如是否針對成本最佳化 speed-to-market 或 。在某些情況下，更有效的方式是針對速度來最佳化，例如快速上市、推出新功能，或滿足截止日期，而不是投資預付成本最佳化。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最佳化不足的部署。不過，若必須將內部部署環境內的資源「平移」至雲端，然後再實施最佳化，這是理性的選擇。前期對成本最佳化策略進行適當投資，並達成一致奉行最佳實務，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最佳化的技術和最佳實務。

最佳實務

主題

實作雲端財務管理

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

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

在 中， AWS 您可以使用 Cost Explorer 、 以及選用的 Amazon Athena 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意識。 AWS Budgets 可針對成本和用量提供主動通知。 AWS 部落格提供有關新服務和功能的資訊，以驗證您是否掌握最新的服務版本。

下列問題著重於成本最佳化方面的這些考量。(如需成本最佳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？

實作 Cloud Financial Management 可協助組織在最佳化成本和用量時實現商業價值和財務成功，並在 上擴展規模 AWS。

建置成本最佳化函數時，請使用 成員，並使用 CFM和 成本最佳化的專家來補充團隊。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。

了解支出和用量

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。它減少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 中 AWS，您可以使用 AWS Organizations 或 建立帳戶結構 AWS Control Tower，該結構提供分離，並協助您配置成本和用量。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 來查看成本和用量，或使用 Amazon Athena 和 Amazon 建立自訂儀表板和分析 QuickSight。透過 AWS Budgets 的通知來控制您的成本和用量，並使用 AWS Identity and Access Management （ IAM ） Service Quotas來控制。

下列問題著重於成本最佳化方面的這些考量。

COST 2：如何管理用量？

制訂政策和機制以驗證產生合理的成本，同時達成目標。透過採用方法 checks-and-balances，您可以創新，而不會過度花費。

COST 3：如何監控用量和成本？

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。

COST 4：如何停止使用資源？

實作從專案啟動到 的變更控制和資源管理 end-of-life。這有助於關閉未使用的資源，以減少浪費。

您可以使用成本分配標籤為 AWS 用量和成本進行分類和追蹤。當您將標籤套用至 AWS 資源 （ 例如 EC2執行個體或 S3 儲存貯體 ） 時， 會根據您的用量和標籤 AWS 產生成本和用量報告。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。

確認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。如需高層級的洞察和趨勢，請透過 AWS Cost Explorer使用每日精細度。如需更深入的分析和檢查 AWS Cost Explorer，請在 、 或 Amazon Athena 和 Amazon 中使用每小時精細度 QuickSight ，並以每小時精細度搭配成本和用量報告（CUR）。

將加有標籤的資源結合實體生命週期追蹤 (員工、專案)，可識別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 提供多種靈活且符合成本效益的定價選項，以更有效符合您需求的方式從 Amazon EC2和其他 服務取得執行個體。隨需執行個體可讓您按時數為運算容量付費，無最低承諾的要求。Savings Plans 和預留執行個體可提供高達 75% 的隨需定價折扣。使用 Spot 執行個體，您可以利用未使用的 Amazon EC2容量，並提供最高 90% 的隨需定價折扣。Spot 執行個體適合使用一組伺服器，其中個別伺服器可以動態進出，例如無狀態 Web 伺服器、批次處理，或使用 HPC和大數據時。

適當的服務選擇也可以降低用量和成本；例如 CloudFront 將資料傳輸降至最低，或降低成本，例如在 Amazon 上使用 Amazon Aurora RDS來移除昂貴的資料庫授權成本。

下列問題著重於成本最佳化方面的這些考量。

COST 5：當您選取 服務時，如何評估成本？

Amazon EC2、Amazon EBS和 Amazon S3 是建置區塊 AWS 服務。受管服務，例如 Amazon RDS和 Amazon DynamoDB是更高層級的服務，或應用程式層級 AWS 的服務。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。

COST 6：當您選取資源類型、大小和數字時，如何達到成本目標？

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。

COST 7：如何使用定價模型來降低成本？

使用最適合您資源的定價模式，就能盡量減少支出。

COST 8：如何規劃資料傳輸費用？

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。

透過在選擇服務期間考慮成本，並使用 Cost Explorer 等工具 AWS Trusted Advisor ，並定期檢閱您的 AWS 用量，您可以主動監控您的使用率並相應地調整部署。

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，減少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 中 AWS，您可以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，可讓您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，並驗證資源符合工作負載需求。您可以使用 Amazon API Gateway 實作限流，或使用 Amazon 在工作負載中SQS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最佳化方面的這些考量。

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，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。任一方向的偏差使用率指標都會對您的組織造成負面影響，無論是營運成本（因過度使用而降低效能）或浪費 AWS 支出（因過度佈建而）。

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，確認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 推出新的服務和功能，最佳實務是檢閱您現有的架構決策，以驗證它們是否繼續最具成本效益。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最佳化工作負載，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，並讓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最佳化方面的這些考量。

COST 10：如何評估新服務？

隨著 AWS 推出新的服務和功能，最佳實務是檢閱您現有的架構決策，以驗證它們是否繼續最具成本效益。

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，Amazon Aurora on Amazon RDS可以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。

COST 11：如何評估工作成本？

評估雲端操作的工作成本、檢閱耗時的雲端操作，並透過採用相關 AWS 服務、第三方產品或自訂工具來自動化這些操作，以降低人力工作量和成本。

資源

請參閱以下資源，進一步了解我們成本最佳化的最佳實務。

文件

- AWS 文件

白皮書

- 成本最佳化支柱

永續性

永續性支柱的重點在於環境影響，尤其是能源消耗和效率，因為這方面是架構師採取直接行動來減少資源使用的重要手段。可以在永續性支柱白皮書中找到實作指引。

主題

- 設計原則
- 定義
- 最佳實務
- 資源

設計原則

在雲端中實現永續性有六個設計原則：

- 了解您的影響：衡量您雲端工作負載的影響，並建立工作負載未來影響的模型。包含所有影響來源，包括客戶使用您產品所產生的影響，以及產品最後除役和淘汰所產生的影響。審視每個工作單元所需的資源和排放量，將生產輸出與雲端工作負載的總體影響進行比較。使用此資料來建立關鍵績效指標（KPIs）、評估提高生產力同時降低影響的方法，以及估算提議變更隨著時間的影響。
- 建立永續性目標：對於每個雲端工作負載，建立長期永續性目標，例如減少每項交易所需的運算和儲存資源。針對改進現有工作負載永續性的投資回報建立模型，並為業主提供必須投資於永續性目標所需的資源。針對成長著手規劃，以及建構您的工作負載，讓成長導致的每個使用者或每項交易 (根據適當的單位測量) 影響強度降低。這些目標有助於支持貴企業或組織更廣泛的永續性發展目標、識別迴歸，以及排定潛在改進領域的優先順序。
- 最大化使用率：調整合適的工作負載規模並實作高效率設計，以確認高使用率，並將底層硬體的能源效率發揮到最大。由於每部主機會有基準耗電量，因此兩部以 30% 使用率執行的主機效率，會低於一部以 60% 執行的主機效率。在此同時減少或最小化閒置資源、處理和儲存，以減少為工作負載供電所需的能源總量。
- 預測並採用新的、更有效率的軟硬體產品：支援合作夥伴和供應商進行上游改進，以協助您減少雲端工作負載的影響。持續監控和評估更有效率的新軟硬體產品。針對靈活性進行設計，以允許快速採用高效率的新技術。

- 使用受管服務：在廣泛的客戶群中共用服務，有助於最大化資源利用，進而減少支援雲端工作負載所需的基礎設施數量。例如，客戶可以將工作負載遷移至 AWS 雲端 並採用受管服務，例如 AWS Fargate for serverless Container，其中 會大規模 AWS 運作，並負責其高效操作，以共享電力和聯網等常見資料中心元件的影響。使用 受管服務，有助於將影響降至最低，例如使用 Amazon S3 生命週期組態或 Amazon EC2 Auto Scaling 自動將不常存取的資料移至冷儲存，以調整容量以符合需求。
- 減少雲端工作負載的下游影響：減少使用您的服務所需的能源或資源量。減少客戶為了使用您的服務而升級裝置的需求。使用 Device Farm 進行測試以了解預期影響，並與客戶進行測試以了解使用您服務的實際影響。

定義

雲端永續性有六個最佳實務領域：

- 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

雲端中的永續性是一項近乎持續的工作，主要專注於透過從佈建的資源中實現最大效益並最大限度地減少所需的總資源，來減少工作負載所有元件的節能和效率。此工作的範圍包括最初選擇高效的程式設計語言、採用現代演算法、使用高效的資料儲存技術、部署至正確大小和高效的運算基礎設施，以及最大限度地減少對高效能最終使用者硬體的要求。

最佳實務

主題

- 程序和文化

區域選擇

工作負載的 區域選擇會大幅影響其 KPIs，包括效能、成本和碳足跡。若要改善這些 KPIs，您應該根據業務需求和永續性目標，為工作負載選擇區域。

下列問題著重於這些永續性方面的考量。(如需永續性問題清單和最佳實務，請參閱附錄。)

SUS 1：如何為工作負載選取區域？

工作負載的 區域選擇會大幅影響其 KPIs，包括效能、成本和碳足跡。若要改善這些 KPIs，您應該根據業務需求和永續性目標，為工作負載選擇區域。

因應需求

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。

下列問題著重於永續性方面的考量：

SUS 2：如何根據需求調整雲端資源？

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。

隨使用者負載擴展基礎設施：識別使用率低或無使用率的時期，並調整資源規模以減少過剩容量、提高效率。

SLAs 與永續性目標保持一致：定義和更新服務層級協議 （SLAs），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數量，同時繼續滿足業務需求。

停止建立和維護未使用的資產：分析應用程式資產 (例如預先編譯的報告、資料集和靜態影像) 和資產存取模式，識別冗餘、未充分利用和可以除役的目標。合併具有冗餘內容的產生資產 (例如，具有重疊

或通用資料集與輸出的每月報告)，以減少重複輸出時消耗資源。將未使用的資產除役 (例如不再販售產品的影像) 以釋放消耗的資源，並減少用於支援工作負載的資源數量。

針對使用者位置最佳化工作負載的地理位置：分析網路存取模式，識別客戶的地理連接位置。選取可減少網路流量傳輸距離的區域和服務，以減少支援工作負載所需的總網路資源。

為執行的活動最佳化團隊成員資源：最佳化提供給團隊成員的資源，以盡量減少對永續性的影響，同時支援他們的需求。例如，在使用率高的共用雲端桌面上執行複雜的操作 (例如渲染和編譯)，而不是在使用率低的高功率單一使用者系統上執行。

軟體和架構

實施可執行負載順暢並保持已部署資源一致高使用率的模式，將資源消耗降至最低。由於使用者行為隨時間改變，元件可能會因缺乏使用而閒置。修改模式和架構來整合未充分利用的元件，提高整體使用率。淘汰不再需要的元件。了解工作負載元件的效能，並最佳化消耗最多資源的元件。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。

下列問題著重於這些永續性方面的考量：

SUS 3：如何利用軟體和架構模式來支援永續性目標？

實施可執行負載順暢並保持已部署資源一致高使用率的模式，將資源消耗降至最低。由於使用者行為隨時間改變，元件可能會因缺乏使用而閒置。修改模式和架構來整合未充分利用的元件，提高整體使用率。淘汰不再需要的元件。了解工作負載元件的效能，並最佳化消耗最多資源的元件。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。

最佳化非同步與排程任務的軟體和架構：使用高效率的軟體設計和架構，將每個工作單元所需的平均資源降至最低。實作可平均利用元件的機制，減少任務之間的閒置資源，並將負載尖峰的影響降至最低。

移除或重構使用量低或完全未使用的工作負載元件：監控工作負載活動，識別各元件使用率隨時間的變化。移除未使用且不再需要的元件，並重構使用率低的元件，減少資源浪費。

最佳化程式碼中耗用最多時間或資源的區域：監控工作負載活動，識別消耗最多資源的應用程式元件。最佳化這些元件中執行的程式碼，將資源使用量降至最低，同時將效能發揮至最大。

最佳化對客戶裝置和設備的影響：了解客戶用來使用您服務的裝置和設備、其預期生命週期，以及更換這些元件對財務和永續性的影響。實作軟體模式和架構，將客戶更換裝置和升級設備的需求降至最低。例如，實作使用與較早硬體和作業系統版本向後相容的程式碼的新功能，或管理承載的大小，不讓其超過目標裝置的儲存容量。

使用最有效支援資料存取和儲存模式的軟體模式和架構：了解資料在工作負載中的使用方式、使用者的使用方式、傳輸方式以及儲存方式。選取可將資料處理和儲存要求降至最低的技術。

資料管理

下列問題著重於這些永續性方面的考量：

SUS 4：如何利用資料管理政策和模式來支援永續性目標？

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，並使用最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。

實作資料分類政策：將資料分類，以了解其對業務成果的重要性。使用此資訊來確定何時可將資料移動到更節能的儲存，或是可以安全刪除它。

使用支援資料存取和儲存模式的技術：使用最有效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。例如，固態裝置（SSDs）比磁性磁碟機更耗能，且應僅用於作用中的資料使用案例。針對不常存取的資料，使用節能的存檔類別儲存。

使用生命週期政策來刪除不需要的資料：管理所有資料的生命週期並自動執行刪除時間表，將工作負載的總儲存需求降至最低。

將區塊儲存中的過度佈建降至最低：若要最小化總佈建儲存，請建立具有適合工作負載之大小分配的區塊儲存。使用彈性磁碟區，隨著資料成長擴展儲存，無需調整連接到運算資源的儲存大小。定期審查彈性磁碟區，並縮減過度佈建的磁碟區以符合目前的資料大小。

移除不需要或多餘的資料：必要時才複製資料，將消耗的總儲存空間降至最低。使用在檔案和區塊層級刪除重複資料的備份技術。限制使用獨立磁碟機的備援陣列（RAID）組態，除非需要滿足 SLAs。

使用共用檔案系統或物件儲存體存取通用資料：採用共用儲存和單一真實來源，避免資料重複並降低工作負載的總儲存需求。僅在需要時從共用儲存體擷取資料。分離未使用的磁碟區以釋放資源。最小化跨網路的資料移動：使用共用儲存，並從區域資料存放區存取資料，將支援工作負載資料移動所需的總聯網資源降至最低。

僅在難以重新建立時才備份資料：為了將儲存消耗降至最低，僅備份具有業務價值或需要滿足合規要求的資料。檢查備份政策，並在復原案例中排除沒有價值的暫時性儲存。

硬體和服務

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。

下列問題著重於這些永續性方面的考量：

SUS 5：如何在架構中選擇並使用雲端硬體和服務來支援永續性目標？

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。

使用最低數量的硬體來滿足需求：使用雲端功能，您可以頻繁變更工作負載實作。隨著需求變更，更新已部署的元件。

使用影響最小的執行個體類型：持續關注新執行個體類型的發佈，並運用能源效率改進，包括旨在支援特定工作負載 (例如機器學習訓練和推論以及影片轉碼) 的執行個體類型。

使用 受管服務： 受管服務會將維持部署硬體高平均使用率和永續性最佳化的責任轉移到 AWS。使用受管服務，將服務的永續性影響分散給服務的所有租用戶，降低您的個人佔比。

最佳化您對 的使用GPUs：圖形處理單元 （GPUs） 可以是高耗電的來源，而且許多GPU工作負載具有高度可變性，例如轉譯、轉碼和機器學習訓練和建模。只在所需的時間內執行GPUs執行個體，並在不需要將耗用的資源降至最低時，透過自動化停用執行個體。

程序和文化

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。

下列問題著重於這些永續性方面的考量：

SUS 6：您的組織程序如何支援您的永續性目標？

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。

採用可快速導入永續性改進的操作：在將潛在改善部署到生產環境之前，先對其進行測試和驗證。在計算改善所帶來的未來潛在利益時，應考慮測試成本。開發低成本測試操作，以推動小改進的交付。

讓您的工作負載保持最新狀態： Up-to-date作業系統、程式庫和應用程式可以提高工作負載效率，並建立更有效率的技術採用。 Up-to-date 軟體也可能包含功能，以更精確地測量工作負載的永續性影響，因為廠商提供功能來滿足自己的永續性目標。

提高建置環境的使用率：使用自動化和基礎設施即程式碼，在需要時啟動生產前環境，並在不使用時將其關閉。常見的模式是排程可用性時間，使之與開發團隊成員的工作時間一致。休眠是一種有用的工具，可保留狀態，並在需要時快速讓執行個體上線。使用具有高載容量的執行個體類型、Spot 執行個體、彈性資料庫服務、容器和其他技術，以根據使用量調整開發和測試容量。

使用受管 Device Farm 進行測試：受管 Device Farm 可將硬體製造和資源使用的永續性影響分散給多個租用戶。受管 Device Farm 提供多種裝置類型，因此您可以支援較早且較不熱門的硬體，並避免不必要的裝置升級對客戶的永續性造成影響。

資源

請參閱下列資源，進一步了解我們的永續性最佳實務。

白皮書

- 永續性支柱

影片

- 氣候承諾

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識別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。採取近乎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 架構與內部審核系統和服務的方式 AWS 一致。它以影響架構方法的一組設計原則為基礎，以及驗證人員不會忽略根本原因分析 （RCA） 中經常出現的區域的問題。每當內部系統、AWS 服務或客戶發生重大問題時，我們會查看 RCA，看看我們是否可以改善我們使用的審核程序。

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。您必須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。

如果您想以審查作為一次性的快照或獨立測量，建議確認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 out-of-band研究才能回答的問題動作清單 （例如，「我們是否啟動加密？」）

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。您還需要考慮這些問題對您團隊day-to-day工作的影響。如果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畫。
- 「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 架構中的疑問，他們就能看出這些疑問完全不會顯露商業或技術上的專屬資訊。

在您與組織內的團隊實施多重審查之時，可能會識別主題上的問題。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，並識別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 提供跨六個支柱的架構最佳實務，用於設計和操作雲端中可靠、安全、高效、經濟實惠且永續的系統。該架構提供一套問題，允許您審查現有或提議的架構。它也為每個支柱提供一組 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。

貢獻者

下列個人和組織為本文件作出了貢獻：

- 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，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，Amazon Web Services 解決方案架構師
- Ben Mergen，Amazon Web Services 資深成本主管與解決方案架構師
- Chris Kozlowski，Amazon Web Services 企業支援資深專家與技術客戶經理
- Alex Livingstone，Amazon Web Services 雲端操作首席專家與解決方案架構師
- Paul Moran，Amazon Web Services 企業支援首席技術專家
- Peter Mullen，Amazon Web Services 專業服務諮詢顧問
- Chris Pates，Amazon Web Services 企業支援資深專家與技術客戶經理
- Arvind Raghunathan，Amazon Web Services 企業支援首席專家與技術客戶經理
- Sam Mokhtari，Amazon Web Services 資深效率主管與解決方案架構師

深入閱讀

AWS 架構中心

AWS 雲端合規

AWS Well-Architected Partner 計畫

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

成本最佳化支柱白皮書

永續性支柱白皮書

Amazon 建置者資料中心

文件修訂

若要收到此白皮書的更新通知，請訂閱RSS摘要。

變更	描述	日期
已更新最佳實務指引	整個支柱進行了大規模最佳實務更新。安全性和成本都獲得了新的最佳實務。	2024 年 6 月 27 日
主要更新	主要支柱更新。	2023 年 10 月 3 日
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。	2023 年 4 月 10 日
次要更新	已在附錄中新增工作量的定義和更新最佳實務。	2022 年 10 月 20 日
白皮書已更新	已新增永續性支柱和更新了連結。	2021 年 12 月 2 日
主要更新	永續性支柱已新增到框架中。	2021 年 11 月 20 日
次要更新	已移除非包容性語言。	2021 年 4 月 22 日
次要更新	已修正數個連結。	2021 年 3 月 10 日
次要更新	整體的小幅度編輯變更。	2020 年 7 月 15 日
新框架的更新	檢閱和重寫大多數問題和答案。	2020 年 7 月 8 日
白皮書已更新	新增 AWS Well-Architected Tool、 AWS Well-Architected Labs 連結，以及 AWS Well-Architected Partners、啟用多語言版本架構的次要修正。	2019 年 7 月 1 日

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。	2018 年 11 月 1 日
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。	2018 年 6 月 1 日
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的演變 AWS。	2017 年 11 月 1 日
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。	2016 年 11 月 1 日
次要更新	以目前的 Amazon CloudWatch Logs 資訊更新附錄。	2015 年 11 月 1 日
初次出版	AWS Well-Architected Framework 已發佈。	2015 年 10 月 1 日

> **Note**
>
> 若要訂閱RSS更新，您必須為正在使用的瀏覽器啟用RSS外掛程式。

框架版本：

- 2023-10-03 (目前)
- 2023-04-10
- 2022-03-31

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
- 可靠性
- 效能效率
- 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。您可以在卓越營運支柱白皮書中找到實作指引。

最佳實務領域

- 組織
- 準備
- 營運
- 演進

組織

問題

- OPS 1. 如何判斷優先順序？
- 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. 如何判斷優先順序？

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 評估治理要求
- OPS01-BP04 評估合規要求
- OPS01-BP05 評估威脅態勢
- OPS01-BP06 在管理效益和風險時評估權衡

OPS01-BP01 評估客戶需求

讓關鍵利益相關者 (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。這驗證了您對實現預期業務成果所需的營運支援有透徹的了解。

預期成果：

- 您可以透過客戶成果逆向工作。
- 您了解營運實務如何支援業務成果和目標。
- 您與所有相關各方接觸。
- 您擁有捕捉客戶需求的機制。

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。您不知道這是否會對您的客戶產生影響。
- 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：需求得到滿足的客戶更有可能繼續成為客戶。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曝險等級：高

實作指引

了解業務需求：只有業務、開發及營運團隊等利益相關者擁有共同的目標並達成共識，方能讓業務取得成功。

審查外部客戶的業務目標、需求和優先事項：與關鍵利益相關者 (包括業務、開發和營運團隊) 進行互動，以討論外部客戶的目標、需求和優先事項。這將確保您對實現業務和客戶成果所需的營運支援有透徹的了解。

建立共識：在以下方面建立共識：工作負載的業務功能、每個團隊在工作負載營運過程中的角色，以及這些因素如何支援內外部客戶的共同業務目標。

資源

相關的最佳實務：

- OPS11-BP03 實作意見回饋循環

OPS01-BP02 評估內部客戶需求

讓關鍵利益相關者 (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。

預期成果：

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

常見的反模式：

- 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產生什麼影響。
- 您正在實作新的開發工具，但尚未讓內部客戶了解是否需要它，或者它是否與他們現有的實務相容。
- 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，可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曝險等級：高

實作指引

- 了解業務需求：只有業務、開發及營運團隊等利益相關者擁有共同的目標並達成共識，方能實現業務成功。
- 審查內部客戶的業務目標、需求和優先事項：與關鍵利益相關者 (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，即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 實作意見回饋循環

OPS01-BP03 評估治理要求

管控是政策、規則或架構的集合，供公司用來達成其業務目標。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畫。您未建立該計畫，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曝險等級：高

實作指引

與利益相關者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 零售，雲端營運團隊會與整個組織的利益相關者合作，以開發治理要求。例如，它們禁止SSH存取 Amazon EC2執行個體。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. 識別工作負載的利益相關者，包括任何集中團隊。
2. 與利益相關者共同識別管控要求。
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 等服務AWS Config來建立 governance-as-code和驗證遵循了治理要求。
   b. 如果您使用 AWS Organizations，則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 - 合規性類似於管控，但來自組織外部。

相關文件：

- AWS 管理和治理雲端環境指南

- 多帳戶環境中 AWS Organizations 服務控制政策的最佳實務
- 中的治理 AWS 雲端：敏捷性與安全之間的正確平衡
- 什麼是治理、風險與合規（GRC）？

相關影片：

- AWS 管理和治理：組態、合規和稽核 - AWS 線上技術講座
- AWS re：Inforce 2019：雲端時代的治理（DEM12-R1）
- AWS re：Invent 2020：使用 實現合規作為程式碼 AWS Config
- AWS re：Invent 2020：靈活管理 AWS GovCloud (US)

相關範例：

- AWS Config Conformance Pack 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。合規標準的範例包括 PCI DSS、Fed RAMP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：

- 您的部分工作負載屬於支付卡產業資料安全標準 （PCI-DSS） 架構，但您的工作負載會儲存未加密的信用卡資料。
- 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。
- 年度系統和組織控制 （SOC2） 類型 II 稽核即將進行，您無法驗證控制項是否就位。

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，有助於證明您確實遵循合規架構。

未建立此最佳實務時的曝險等級：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 零售會為客戶儲存信用卡資訊。卡片儲存團隊的開發人員了解他們需要遵守 PCI架構DSS。他們已採取步驟來驗證信用卡資訊是否已根據 PCI架構DSS安全地儲存和存取。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 等服務驗證 AWS 資源的持續合規性AWS Security Hub。
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. 使用諸如 AWS Audit Manager 等服務來產生驗證合規性並產生稽核報告。
   b. 您可以使用 下載 AWS 安全與合規文件AWS Artifact。

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的重要部分。
- SEC01-BP06 自動化管道中安全控制的測試和驗證 - 作為管道的一部分，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 - 鑑識功能有時可用於稽核合規性。

相關文件：

- AWS 合規中心
- AWS 合規資源
- AWS 風險與合規白皮書
- AWS 共同責任模型
- AWS 依合規計劃在範圍內的服務

相關影片：

- AWS re：Invent 2020：使用 以程式碼形式達成合規 AWS Compute Optimizer
- AWS re：Invent 2021 - 雲端合規、保證和稽核
- AWS Summit ATL 2022 - 在 AWS （COP202） 實作合規、保證和稽核

相關範例：

- PCI DSS 和 AWS 基礎安全最佳實務 AWS

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。它提供一致的方法來評估架構，並實作會隨著時間擴展的設計。 AWS 提供 AWS Well-Architected Tool，協助您在開發之前檢閱方法、生產前的工作負載狀態，以及生產中工作負載的狀態。您可以將它們與最新的 AWS 架構最佳實務進行比較，監控工作負載的整體狀態，並深入了解潛在風險。

AWS 客戶有資格接受其任務關鍵工作負載的引導式 Well-Architected Review，以根據 AWS 最佳實務測量其架構。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 是一款可存取核心檢查集的工具，這些檢查提出了優化建議，可能有助您確定優先事項。商業和企業支援客戶可存取針對安全性、可靠性、效能和成本優化的其他檢查，從而進一步協助確定他們的優先事項。

預期成果：

- 您可以定期檢閱 Well-Architected 和 Trusted Advisor 輸出並採取行動
- 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，並採取相應的行動
- 視需要實作緩和措施
- 對行動和背景進行溝通

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會對工作負載產生意外影響。
- 競爭對手剛剛發布了其產品的一個版本，它可以解決許多客戶對您產品的投訴。您並未優先處理這些已知問題。
- 監管機構一直在追查像你們這樣不符合法律監管合規要求的公司。您尚未排定處理任何未解決合規要求之事項的優先順序。

建立此最佳實務的優勢：識別並了解組織和工作負載所面臨的威脅，有助於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曝險等級：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)，以便您可以在決定工作重點時考量其影響。
  - AWS 最新安全公告
  - AWS Trusted Advisor
- 維護威脅模型：建立和維護用於識別潛在威脅、已規劃和就地緩解措施及其優先順序的威脅模型。審查顯示為事件的威脅的機率、從這些事件中復原的成本、導致的預期傷害，以及防止這些事件的成本。當威脅模型的內容變更時，修改優先順序。

資源

相關的最佳實務：

- SEC01-BP07 使用威脅模型識別威脅並排定緩解優先順序

相關文件：

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

相關影片：

- AWS re:Inforce 2023 - 協助改善威脅模型的工具

OPS01-BP06 在管理效益和風險時評估權衡

來自多方的競爭利益可能會使安排工作的優先順序、建立能力、交付與業務策略一致的成果變得具有挑戰性。例如，您可能被要求加速 speed-to-market新的功能，而不是最佳化 IT 基礎設施成本。這會使利益雙方發生衝突。在這種情況下，需要將決策提交給更高的權利層來解決衝突。需要使用資料來消除決策過程中的情感依戀。

同樣的挑戰可能會發生在戰術層面。例如，選擇使用關聯式或非關聯式資料庫技術，可能會對應用程式的操作產生重大影響。了解各種選擇的可預測結果至關重要。

AWS 可協助您教育團隊有關 AWS 及其 服務的資訊，以進一步了解其選擇如何影響您的工作負載。使用 AWS Support (AWS 知識中心、AWS 論壇和 AWS Support 中心) 和 AWS 文件提供的資源來教育您的團隊。如有其他問題，請聯絡 AWS Support。

AWS 也會在 Amazon Builders' Library 中分享操作最佳實務和模式。部落格AWS和官方 AWS Podcast 提供了各種其他有用的資訊。

預期成果：您擁有明確定義的決策管控框架，可以促進雲端交付組織內每個層級的重要決策。該框架包括很多功能，例如風險登記表、有權做出決策的已定義角色以及可以做出的每個決策級別的定義模型。此框架預先定義了如何解決衝突，需要呈現哪些資料，以及如何確定選項的優先級，以便一旦做出決定，就可以立即提交。決策框架包括一個標準化方法，可審查和衡量每個決策的利益和風險，以了解權衡。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遵循支付卡產業資料安全標準 （PCI DSS）。您沒有考量滿足要求和繼續您目前開發工作之間的權衡取捨。相反地，您繼續開發工作，而不證明合規性。由於對平台安全性及其投資的擔憂，您的投資者會停止對公司的支援。
- 您已決定包含一個開發人員在網際網路上發現的程式庫。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。
- 遷移的最初業務理由是基於應用程式工作負載 60% 的現代化。然而，由於技術上的困難，決定只對 20% 進行現代化，導致長期計畫收益減少，基礎架構團隊手動支援舊式系統的操作員工作量增加，並且更依賴於在沒有規劃此變更的基礎架構團隊中開發新技能。

建立此最佳實務的優勢：充分協調和支援董事會層級的業務優先事項、了解取得成功的風險、制定明智決策，以及在風險阻礙成功機會時採取適當行動。了解決策的影響和後果有助於您優先考慮您的選擇，並更快地讓領導者達成一致，從而改善業務成果。確定您的選擇的可用好處並了解組織的風險，可以幫助您制定資料驅動型決策，而不是依賴於軼事。

未建立此最佳實務時的曝險等級：中

實作指引

管理利益和風險應由推動關鍵決策要求的管理機構來定義。您希望根據決策如何使組織受益來做出決策並確定優先級，並了解所涉及的風險。準確的資訊對於制定組織決策至關重要。這應該基於可靠的測量結果，並由成本效益分析的常見行業實務進行定義。要做出這些類型的決策，請在集中式和分散式授權之間取得平衡。總有一種權衡，了解每個選擇如何影響定義的策略和期望的業務成果至關重要。

實作步驟

1. 在整體式雲端治理架構中正式確定效益衡量實務。
   a. 在決策的中央控制與某些決策的分散權力之間取得平衡。
   b. 了解強加給每項決策的繁瑣決策流程可能會減緩您的速度。
   c. 將外部因素納入決策流程中 (例如合規要求)。
2. 為各級決策建立一個商定的決策框架，其中包括誰需要為受衝突利益影響的決策清除障礙。
   a. 集中化可能不可逆轉的單向門決策。
   b. 允許由較低級別的組織領導者做出雙向門決策。
3. 了解和管理利益和風險。在決策的收益與所涉及的風險之間取得平衡。
   a. 確定收益：根據業務目標、需求和優先事項確定收益。範例包括業務案例影響、 time-to-market、安全性、可靠性、效能和成本。
   b. 確定風險：根據業務目標、需求和優先事項確定風險。範例包括 time-to-market、安全性、可靠性、效能和成本。
   c. 根據風險評估收益並做出明智決策：根據利益相關者 (包括業務、開發和營運團隊) 的目標、需求和優先事項，確定收益和風險的影響。評價收益的價值時要考慮發生風險的可能性及其代價。例如，強調 speed-to-market可靠性可能會提供競爭優勢。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 Value Stream Analysis 和 等已知產業架構和功能LEAN，以基準化目前狀態效能、業務指標，並定義改善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 的 Day 1 文化要素 | 做出高品質、高速度的決策
- 雲端治理
- 管理與治理雲端環境

- 雲端和數位時代的治理：第一部分和第二部分

相關影片：

- 播客 | Jeff Bezos | 關於如何制定決策

相關範例：

- 使用資料（ DevOps Sagas ）做出明智的決策
- 使用開發值串流映射來識別 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 資源已識別擁有者
- OPS02-BP02 程序已識別擁有者
- OPS02-BP03 操作活動已識別負責其績效的擁有者
- OPS02-BP04 存在機制來管理責任和所有權
- OPS02-BP05 機制用於請求新增、變更和例外狀況
- 團隊之間的 OPS02-BP06 責任已預先定義或協商

OPS02-BP01 資源已識別擁有者

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。

預期成果：

- 資源已使用中繼資料或中央寄存器識別擁有者。
- 團隊成員可以識別誰擁有資源。
- 帳戶在可能的情況下擁有單一擁有者。

常見的反模式：

- 您 的替代聯絡人 AWS 帳戶 不會填入。
- 資源缺少可識別其屬於哪個團隊的標籤。
- 您有一個沒有電子郵件映射的ITSM佇列。
- 兩個團隊對基礎設施的關鍵部分的擁有權重疊。

建立此最佳實務的優勢：

- 透過指派擁有權，資源的變更控制很簡單。
- 疑難排解問題時，可以讓合適的擁有者參與。

未建立此最佳實務時的曝險等級：高

實作指引

定義擁有權對環境中的資源使用案例的意義。擁有權可能意味著誰監督資源的變更、在疑難排解期間支援資源或者是負有財務責任的人員。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 零售將所有權定義為擁有變更和資源支援之團隊或個人。他們利用 AWS Organizations 來管理其 AWS 帳戶。備選帳戶連絡人正在使用群組收件匣進行設定。每個ITSM佇列都會對應至電子郵件別名。標籤會識別誰擁有 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。擁有權可能表示誰擁有資源的風險、誰擁有資源的變更，或誰在疑難排解時支援資源。擁有權也可能意味著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 AWS 帳戶中將這些設定為帳戶、安全和營運聯絡人。即使有人正在休假、變更角色或離開公司，多人也會收到 AWS 通知並能夠回應。
   b. 如果帳戶不是由 AWS Organizations 管理，備選帳戶聯絡人便會協助 AWS 適時與相關人員取得聯繫。設定帳戶的備選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。您可以在不同的標籤中指定擁有者及其聯絡資訊。

   a. 可以使用 AWS Config 規則來強制資源具有所需的擁有權標籤。
   b. 如需有關如何為組織建立標記策略的深入指引，請參閱 AWS Tagging Best Practices whitepaper。
4. 使用 Amazon Q Business，這是一個使用生成式 AI 的對話式助理，可提高員工生產力、回答問題並根據企業系統中的資訊完成任務。
   a. 將 Amazon Q Business 連接到您公司的資料來源。Amazon Q Business 為超過 40 個支援的資料來源提供預先建置的連接器，包括 Amazon Simple Storage Service（Amazon S3）、Microsoft SharePoint、Salesforce 和 Atlassian Confluence。如需詳細資訊，請參閱企業版 Amazon Q 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。所有團隊成員都應可以存取。

實作計劃的工作量：低。利用帳戶聯絡資訊和標籤來指派 AWS 資源的所有權。對於其他資源，您可以使用 Wiki 中的資料表來記錄擁有權和聯絡資訊，或使用ITSM工具映射擁有權。

資源

相關的最佳實務：

- OPS02-BP02 程序已識別擁有者
- OPS02-BP04 存在機制來管理責任和所有權

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 標記最佳實務》白皮書
- 使用 Amazon Q Business and Identity Center 建置私有且 AWS IAM安全的企業生成 AI 應用程式
- Amazon Q Business 現已正式推出，可透過生成式 AI 提升員工生產力
- AWS 雲端 Operations & Migrations 部落格 - 使用 和 實作自動化 AWS Config 和集中式標記控制項 AWS Organizations
- AWS 安全部落格 - 使用 擴展您的預先遞交掛鉤 AWS CloudFormation Guard
- AWS DevOps 部落格 - AWS CloudFormation Guard 整合至 CI/CD 管道

相關研討會：

- AWS 研討會 - 標記

相關範例：

- AWS Config 規則 - EC2具有必要標籤和有效值的 Amazon

相關服務：

- AWS Config 規則 - 必要標籤
- AWS Organizations

OPS02-BP02 程序已識別擁有者

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

預期成果：您的組織擁有一套明確定義且維護良好的操作任何流程和程序。流程和程序儲存在中心位置，並可供您的團隊成員使用。透過明確指定的擁有權經常更新流程和程序。如果可能，指令碼、範本和自動化文件會以程式碼的形式實作。

常見的反模式：

- 未記錄處理程序。隔離的操作員工作站上可能存在碎片化指令碼。
- 有關如何使用指令碼的知識由少數個人掌握，或非正式地作為團隊知識。
- 舊版流程由於更新而到期，但更新的擁有權不清楚，原始著作人不再是組織的一部分。
- 流程和指令碼不容易發現，因此在需要時 (例如，回應事故) 無法立即使用。

建立此最佳實務的優勢：

- 流程和程序可大幅提升您操作工作負載的效率。
- 新的團隊成員更快速地變得高效。
- 減少事故緩解的時間。
- 不同的團隊成員 (和團隊) 可以以一致的方式使用相同的流程和程序。
- 團隊可以透過可重複的流程擴展其流程。
- 標準化流程和程序有助於減輕團隊之間轉移工作負載責任的影響。

未建立此最佳實務時的曝險等級：高

實作指引

- 流程和程序已經確定負責其定義的擁有者。
  - 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 AWS Systems Manager、文件和 等服務，擷取活動成品中繼資料的所有權 AWS Lambda。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，並擷取所有權和聯絡資訊。
- 隨著時間的推移，這些程序應逐步發展為可以作為程式碼執行，從而減少人為干預的需求。
  - 例如，請考慮 AWS Lambda 函數 CloudFormation、範本或 AWS Systems Manager 自動化文件。
  - 在適當的儲存庫中執行版本控制。
  - 包括合適的資源標記，以便可以很容易地識別擁有者和文件。

客戶範例

AnyCompany 零售將擁有權定義為擁有應用程式或應用程式群組（共用常見封存實務和技術）程序的團隊或個人。最初，程序和程序會記錄為 step-by-step文件管理系統中的指南，可使用 AWS 帳戶 託管應用程式的 上的標籤和帳戶中的特定資源群組來探索。他們利用 AWS Organizations 來管理其 AWS 帳戶。隨著時間的推移，這些程序會轉換為程式碼，並使用基礎設施作為程式碼（例如 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本）來定義資源。操作程序會在 AWS Systems Manager 或 AWS Lambda 函數中成為自動化文件，這些文件可以啟動為排程任務、回應 CloudWatch 警示或 AWS EventBridge 事件等 AWS 事件，或由 IT 服務管理（ITSM）平台內的請求啟動。所有流程都有標籤來識別擁有權。在流程的程式碼儲存庫所產生的 wiki 頁面中維護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. 檢閱並保留它們 up-to-date。
   b. 識別每個流程或程序的擁有者。
   c. 將其置於版本控制之下。

d. 在可能的情況下，在共用架構設計的工作負載和環境之間共用流程和程序。

2. 建立回饋和改進機制。

a. 定義檢閱流程頻率的政策。

b. 定義檢閱者與核准者的流程。

c. 實作問題或票證隊列，以提供和追蹤意見回饋。

d. 在可能的情況下，流程和程序應具有變更核准委員會的預先核准和風險分類 （CAB）。

3. 驗證流程和程序是否可供需要執行流程和程序的人員來存取和探索。

a. 使用標籤來指示可以針對工作負載存取流程和程序的位置。

b. 使用有意義的錯誤和事件訊息來指出可解決問題的適當流程或程序。

c. 使用 Wiki 和文件管理，讓整個組織可一致搜尋流程和程序。

4. 適時進行自動化。

a. 當服務和技術提供 時，應該開發自動化API。

b. 對流程進行充分的教育。制定使用者故事和要求以自動化這些流程。

c. 成功衡量流程和程序的使用情況，並追蹤問題以支援反覆改進。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS02-BP01 資源已識別擁有者
- OPS02-BP04 存在機制來管理責任和所有權
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 - DevOps 上的 簡介 AWS
- AWS 白皮書 - 標記 AWS 資源的最佳實務
- AWS 白皮書 - 使用多個帳戶組織 AWS 您的環境
- AWS 雲端 Operations & Migrations 部落格 - 建立卓越營運雲端自動化實務： 最佳實務 AWS Managed Services

- AWS 雲端 Operations & Migrations 部落格 - 使用 和 實作自動化 AWS Config 和集中式標記控制項 AWS Organizations
- AWS 安全部落格 - 使用 擴展您的預先遞交掛鉤 AWS CloudFormation Guard
- AWS DevOps 部落格 - AWS CloudFormation Guard 整合至 CI/CD 管道

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 上自動化 IT 操作 AWS
- AWS re：Invent 2020 - 透過 AWS Systems Manager 自動化任何項目
- AWS re：Inforce 2022 - 使用 AWS （NIS306） 自動化修補程式管理和合規
- AWS Support s You - 深入挖掘 AWS Systems Manager

相關服務：

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

OPS02-BP03 操作活動已識別負責其績效的擁有者

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。

預期成果：

您的組織清楚地規定了對已定義的工作負載執行特定活動並回應工作負載所產生事件的責任。該組織記錄了流程和履行的擁有權，並使此資訊可被發現。您可以在組織變更發生時檢閱並更新責任，而團隊會追蹤並衡量缺陷和低效率識別活動的效能。可以實作回饋機制來追蹤缺陷和改進並支援迭代改進。

常見的反模式：

- 您不記錄責任。

- 隔離的操作員工作站上存在碎片化指令碼。只有少數人知道如何進行使用，或非正式地將其稱為團隊知識。
- 舊版流程由於更新而到期，但沒有人知道誰擁有該流程，並且原始著作人不再是組織的一部分。
- 流程和指令碼無法被發現，在需要時 (例如，回應事故) 無法立即使用。

建立此最佳實務的優勢：

- 了解誰負責執行活動，在需要採取動作時通知誰，以及誰會執行動作、驗證結果並為活動擁有者提供意見回饋。
- 流程和程序可大幅提升您操作工作負載的效率。
- 新的團隊成員更快速地變得高效。
- 可以減少緩解事故所需的時間。
- 不同的團隊可採取一致的方式來使用相同的流程和程序。
- 團隊可以透過可重複的流程擴展其流程。
- 標準化流程和程序有助於減輕團隊之間轉移工作負載責任的影響。

未建立此最佳實務時的曝險等級：高

實作指引

若要開始定義職責，請先從現有文件開始，例如責任矩陣、流程與程序、角色與責任，以及工具與自動化。審查並主持有關文件化流程責任的討論。與團隊一起審查，以識別文件責任和流程之間的不一致。與該團隊的內部客戶討論提供的服務，以確定團隊之間的期望差距。

分析並解決差異。找出改進的機會，並尋找經常請求的資源密集型活動，這些活動通常需要改進。探索最佳實務、模式和規範性指引，以簡化和標準化改進。記錄改善機會，並追蹤改進完成情況。

隨著時間的推移，這些程序應逐步發展為可以作為程式碼執行，從而減少人為干預的需求。例如，程序可以啟動為 AWS Lambda 函數 AWS CloudFormation 、範本或 AWS Systems Manager 自動化文件。驗證這些程序是否在適當的儲存庫中受版本控制，並包含適當的資源標記，以便團隊能夠輕鬆識別擁有者和文件。記錄執行活動的責任，然後監控用於成功啟動和操作的自動化，以及期望成果的績效。

客戶範例

AnyCompany Retail 將擁有權定義為擁有共用常見架構實務和技術之應用程式或應用程式群組程序的團隊或個人。一開始，公司會將程序和程序記錄為 step-by-step文件管理系統中的指南。它們使用 AWS 帳戶 託管應用程式的 和 帳戶內特定資源群組上的標籤來探索程序，使用 AWS Organizations 來

管理其 AWS 帳戶。隨著時間的推移， AnyCompany Retail 會將這些程序轉換為程式碼，並使用基礎設施作為程式碼來定義資源 （透過類似 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本的服務）。操作程序會成為 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，這些文件可以作為排程任務啟動，以回應 Amazon CloudWatch 警示或 Amazon EventBridge 事件等事件，或 IT 服務管理 （ITSM） 平台內的請求。所有流程都有標籤來識別誰擁有它們。團隊會在流程的程式碼儲存庫所產生的 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. 檢閱並確認它們是 up-to-date。
    b. 確認每個流程或程序都有擁有者。
    c. 將程序置於版本控制之下。
    d. 在可能的情況下，在共用架構設計的工作負載和環境之間共用流程和程序。
2. 建立回饋和改進機制。
    a. 定義檢閱流程頻率的政策。
    b. 定義檢閱者與核准者的流程。
    c. 實作問題或票證隊列，以提供並追蹤意見回饋。
    d. 盡可能提供變更核准委員會 （ ） 的程序和程序的預先核准和風險分類CAB。
3. 讓流程和程序可供需要執行它們的人員進行存取和探索。
    a. 使用標籤來指示可以針對工作負載存取流程和程序的位置。
    b. 使用有意義的錯誤和事件訊息來指出可解決問題的適當流程或程序。
    c. 使用 Wiki 或文件管理，讓整個組織可一致搜尋流程和程序。
4. 在適當的時候實現自動化。
    a. 如果服務和技術提供 API，則開發自動化。
    b. 驗證是否充分理解流程，制定使用者故事和要求以自動化這些流程。
    c. 衡量流程和程序的成功使用情況，並追蹤問題以支援反覆改進。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS02-BP01 資源已識別擁有者

- OPS02-BP02 程序已識別擁有者
- OPS02-BP04 存在機制來管理責任和所有權
- OPS02-BP05 存在機制來識別責任和所有權
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 | DevOps 上的 簡介 AWS
- AWS 白皮書 | 標記 AWS 資源的最佳實務
- AWS 白皮書 | 使用多個帳戶組織 AWS 您的環境
- AWS 雲端 Operations & Migrations 部落格 | 建立卓越營運雲端自動化實務：最佳實務 AWS Managed Services
- AWS 研討會 - 標記
- AWS Service Management Connector

相關影片：

- AWS 知識中心即時 | 標記 AWS 資源
- AWS re：Invent 2020 | 透過 AWS Systems Manager 自動化任何項目
- AWS re：Inforce 2022 | 使用 AWS （NIS306） 自動化修補程式管理和合規
- AWS Support s You | 深入探索 AWS Systems Manager

相關範例：

- AWS Well-Architected 卓越營運研討會

OPS02-BP04 存在機制來管理責任和所有權

了解您角色的責任以及您為業務成果做出貢獻的方式，因為這種了解可明確任務的優先順序以及您的角色為何重要。這有助於團隊成員辨識需求並適當地回應。當團隊成員知道自己的角色時，他們可以建立擁有、找出改進機會，以及了解如何影響或進行適當的變更。

有時候，責任可能沒有明確的擁有者。在這些情況下，設計一種機制可解決此漏洞。為有權指派擁有權或計劃解決需求的人員建立明確定義的上報路徑。

預期成果：組織內的團隊擁有明確定義的職責，其中包括他們與資源、要執行的動作、流程及程序的關係。這些職責符合團隊的責任和目標，以及其他團隊的責任。您可以用一致且可探索的方式記錄上報路徑，並將這些決策輸入到文件成品中，例如責任矩陣、團隊定義或 Wiki 頁面。

常見的反模式：

- 團隊的責任含糊不清或定義不佳。
- 團隊沒有將角色與責任保持一致。
- 該團隊沒有調整其總體目標和具體目標以及其責任，這使得它難以衡量成功。
- 團隊成員的責任與團隊和更廣泛的組織不一致。
- 您的團隊不會保留責任 up-to-date，這會使他們與團隊執行的任務不一致。
- 確定職責的上報路徑尚未定義或不清楚。
- 上報路徑沒有單一執行緒擁有者，以確保及時回應。
- 角色、責任和上報路徑不容易發現，在需要時 (例如，回應事故) 無法立即使用。

建立此最佳實務的優勢：

- 了解誰擁有責任或擁有權時，可讓您聯絡適當的團隊或團隊成員，以提出請求或轉換任務。
- 為了降低不作為和未解決需求的風險，您已經確定了有權指派責任或擁有權的人員。
- 當您明確定義責任範圍時，團隊成員將獲得自主權和擁有權。
- 您的責任決定了您所做的決定、您採取的動作，以及如何將活動交給其適當的擁有者。
- 很容易識別被放棄的責任，因為您清楚地了解哪些責任不在團隊責任範圍內，這有助於您上報以進行澄清。
- 團隊可以避免混亂和緊張，並且可以更充分地管理工作負載和資源。

未建立此最佳實務時的曝險等級：高

實作指引

確定團隊成員的角色和責任，並確保他們了解其角色的期望。讓此資訊可供探索，如此組織的成員便能夠確定他們針對特定需求需要聯絡的人員 (團隊或個人)。隨著組織尋求利用在 上遷移和現代化的機會 AWS，角色和責任也可能會變更。讓您的團隊及其成員了解他們的責任，並對其進行適當的培訓，以便在此次變更期間執行其任務。

確定應接收上報的角色或團隊，以確認責任和擁有權。此團隊可以與各種利益相關者互動以做出決定。但是，他們應該擁有決策制定流程的管理權。

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。這些機制教導他們應該聯絡誰以滿足特定需求。

客戶範例

AnyCompany 零售業最近完成了工作負載從內部部署環境遷移到其 的登陸區域， AWS 並採用提升和轉移方法。他們執行了操作審查以反映他們如何完成一般操作任務，並驗證其現有的責任矩陣是否反映了新環境中的操作。當他們從內部部署遷移到 時 AWS，他們減少了與硬體和實體基礎設施相關的基礎設施團隊責任。此舉還揭示了為其工作負載發展營運模式的新機會。

他們在確定、處理並記錄大部分職責的同時，還定義了任何遺漏或隨著營運實務演變而可能需要變更的任何責任的上報路徑。若要探索新機會來標準化和改善工作負載的效率，請存取 AWS Systems Manager 等操作工具，以及 AWS Security Hub 和 Amazon 等安全工具 GuardDuty。 AnyCompany 零售根據他們想要先解決的改進，彙整責任和策略的審核。由於公司採用新的工作方式和技術模式，他們會更新其責任矩陣以進行匹配。

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、負責、諮詢和知情（RACI）矩陣
   b. 團隊定義或 Wiki 頁面
   c. 服務定義和產品
   d. 角色或職位描述
2. 審查並主持有關文件化責任的討論：
   a. 與團隊進行審查，以確定文件化責任與團隊通常執行的責任之間的不一致。
   b. 討論內部客戶提供的潛在服務，以確定團隊之間的期望差距。
3. 分析並解決差異。
4. 找出改進機會。
   a. 確定頻繁請求的資源密集型請求，這些請求通常需要改進。
   b. 尋找最佳實務、模式和規範性指引，並透過本指引簡化和標準化改進。
   c. 記錄改進機會，並追蹤完成情況。
5. 如果團隊尚未負責管理和追蹤職責指派，請指定團隊中負責此職責的人員。
6. 為團隊定義一個流程以請求澄清責任。
   a. 檢閱流程，並確認其清晰且易用。
   b. 確保有人擁有並追蹤他們結論的上報。

c. 建立運營指標以衡量有效性。

d. 建立意見回饋機制，以確認團隊是否能突顯改進機會。

e. 實作定期審查機制。

7. 文件存放在可發現且可存取的位置。

a. Wiki 或文件入口網站是常見選擇。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS01-BP06 評估權衡
- OPS03-BP02 團隊成員有權在結果處於風險時採取行動
- OPS鼓勵 03-BP03 升級
- OPS03-BP07 資源團隊適當
- OPS09-BP01 測量操作目標和 KPIs 指標
- OPS09-BP03 檢閱操作指標並排定改善優先順序
- OPS11-BP01 擁有持續改善的程序

相關文件：

- AWS 白皮書 - DevOps 上的 簡介 AWS
- AWS 白皮書 - AWS 雲端 採用架構：Operations Perspective
- AWS Well-Architected Framework 卓越營運 - 工作負載層級操作模型拓撲
- AWS 方案指引 - 建置雲端操作模式
- AWS 規範指南 - 為雲端操作模型建立 RACI或 RASCI矩陣
- AWS 雲端 Operations & Migrations 部落格 - 透過雲端平台團隊提供商業價值
- AWS 雲端 Operations & Migrations 部落格 - 為什麼選擇雲端營運模型？
- AWS DevOps 部落格 - 組織如何針對雲端操作進行現代化

相關影片：

- AWS Summit Online - 加速轉型的雲端操作模式

- AWS re:Invent 2023 - 面向未來的雲端安全性：一種新的操作模式

OPS02-BP05 機制用於請求新增、變更和例外狀況

您可以向流程、程序和資源的擁有者提出請求。請求包含新增、變更和例外狀況。這些請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請求。

預期成果：

- 可以根據分配的擁有權請求變更流程、程序和資源。
- 變更是經過深思熟慮的，權衡了利益和風險。

常見的反模式：

- 必須更新部署應用程式的方式，但無法向操作團隊請求變更部署程序。
- 必須更新災難復原計畫，但沒有確定的擁有者可以請求變更。

建立此最佳實務的優勢：

- 流程、程序和資源可隨需求的變化而發展。
- 擁有者可以在進行變更時做出明智決策。
- 變更是經過深思熟慮的。

未建立此最佳實務時的曝險等級：中

實作指引

若要實作此最佳實務，必須能夠要求變更流程、程序和資源。變更管理流程可以很簡單。記錄變更管理流程。

客戶範例

AnyCompany 零售使用責任指派（RACI）矩陣來識別誰擁有程序、程序和資源的變更。他們擁有記錄在案的變更管理流程，簡單且易於遵循。任何人都可以使用 RACI 矩陣和 程序提交變更請求。

實作步驟

1. 確定工作負載的流程、程序和資源，以及各自的擁有者。將其記錄在您的知識管理系統中。

   a. 如果尚未實作 OPS02-BP01 資源已識別擁有者、OPS02-BP02 程序已識別擁有者 或 OPS02-BP03 操作活動已識別負責其績效的擁有者，請先從這些項目開始。
2. 與組織中的利益相關者合作，制定變更管理流程。此流程應涵蓋資源、流程及程序的新增、變更及例外狀況。
   a. 可以使用 AWS Systems Manager Change Manager 做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流程。

實作計劃的工作量：中。制定變更管理流程需要與組織中的多個利益相關者保持一致。

資源

相關的最佳實務：

- OPS02-BP01 資源已識別擁有者 - 在建立變更管理流程之前，資源需要確定的擁有者。
- OPS02-BP02 程序已識別擁有者 - 在建立變更管理流程之前，流程需要確定的擁有者。
- OPS02-BP03 操作活動已識別負責其績效的擁有者 - 在建立變更管理流程之前，操作活動需要確定的擁有者。

相關文件：

- AWS 規範指南 - 適用於 AWS 大型遷移的基礎手冊：建立RACI矩陣
- 雲端白皮書中的變更管理

相關服務：

- AWS Systems Manager Change Manager

團隊之間的 OPS02-BP06 責任已預先定義或協商

團隊間已定義或協商說明如何相互配合及支援的協議 (例如，回應時間、服務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商業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。

預期成果：

- 團隊間的工作或支援協議已達成一致並記錄在案。
- 彼此支援或合作的團隊已定義了溝通渠道和回應期望。

常見的反模式：

- 生產環境中發生問題，而且兩個不同的團隊開始彼此獨立地進行疑難排解。他們各自為政的工作延長了中斷時間。
- 運營團隊需要開發團隊的幫助，但沒有商定回應時間。請求卡在待辦事項中。

建立此最佳實務的優勢：

- 團隊知道如何互動和互相支援。
- 對回應的期望是眾所周知的。
- 溝通渠道已明確定義。

未建立此最佳實務時的曝險等級：低

實作指引

實作此最佳實務意味著團隊之間的合作方式沒有歧義。正式協議規定了團隊如何一起工作或相互支援。團隊間的溝通渠道記錄於文件中。

客戶範例

AnyCompany 零售SRE團隊與開發團隊簽訂了服務層級協議。每當開發團隊在其票證系統中提出請求時，他們可以期望在十五分鐘內得到回應。如果出現網站中斷，SRE團隊會在開發團隊的支援下帶領調查。

實作步驟

1. 與組織中的利益相關者合作，根據流程和程序來制定團隊之間的協議。
   a. 如果在兩個團隊之間共享一個流程或程序，則制定有關團隊將如何一起工作的執行手冊。
   b. 如果團隊之間存在相依性，請同意SLA回應請求。
2. 在知識管理系統中記錄責任。

實作計劃的工作量：中。如果團隊之間沒有現有協議，則可能需要努力與組織中的利益相關者達成協議。

資源

相關的最佳實務：

- OPS02-BP02 程序已識別擁有者 - 在團隊之間達成協議之前，必須確定流程所有權。
- OPS02-BP03 操作活動已識別負責其績效的擁有者 - 在團隊之間達成協議之前，必須確定運營活動所有權。

相關文件：

- AWS Executive Insights - 使用 Two-Pizza 團隊增強創新
- DevOps on 的 簡介 AWS - Two-Pizza Teams

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 提供執行贊助
- OPS03-BP02 團隊成員有權在結果處於風險時採取行動
- 鼓勵 OPS03-BP03 升級
- OPS03-BP04 通訊是及時、清晰且可操作的
- 鼓勵 OPS03-BP05 實驗
- 0OPS03-BP06 鼓勵團隊成員維護和發展其技能集
- OPS03-BP07 資源團隊適當

OPS03-BP01 提供執行贊助

在最高層面上，高級領導層充當執行倡議者，以清楚地設定組織成果的期望和方向，包括評估其成功。倡議者倡導並推動最佳實務的採用和組織的發展。

預期成果：努力採用、轉型和最佳化雲端操作的組織可針對預期成果設定明確的領導層和問責製。該組織了解組織實現新成果所需的每種能力，並將擁有權分配給職能團隊進行開發。領導階層會主動設定此方向、指派所有權、負責並定義工作。因此，整個組織的個人都可以調動起來、備受鼓舞並積極朝著預期目標努力。

常見的反模式：

- 在沒有明確的雲端營運倡議者和計畫的情況下，工作負載擁有者必須將工作負載遷移至 AWS 。這導致團隊無法自覺地協作以改善和充分發展其營運能力。缺乏營運最佳實務標準使團隊不堪重負 (例如營運商辛苦工作、隨叫隨到以及技術債務)，這會限制創新。
- 一個新的全組織目標已設定，即在不提供領導倡議者和策略的情況下採用新興技術。團隊以不同的方式闡述目標，這會導致對於將精力集中在哪裡、為什麼重要以及如何衡量影響感到困惑。因此，組織在採用技術方面失去了動力。

建立此最佳實務的優勢：當高層支援明確傳達並分享願景、方向和目標時，團隊成員知道他們的期望是什麼。當領導者積極參與時，個人和團隊開始將精力集中在相同的方向上，以實現定義的目標。因此，組織最大限度地提高了成功的能力。當您評估成功時，您可以更好地識別成功障礙，以便透過高層支援的干預來解決這些障礙。

未建立此最佳實務時的曝險等級：高

實作指引

- 在雲端之旅的每個階段 (遷移、採用或最佳化)，成功都需要最高領導層與指定執行倡議者的積極參與。執行倡議者可根據定義的策略來調整團隊的心態、技能組合和工作方式。
  - 解釋原因：澄清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布目標，包括如何衡量進度和成功。
  - 追蹤目標的達成情況：定期衡量目標逐步達成的情況 (不只是完成任務)。分享結果，以便在成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，共同合作並構建正確的解決方案，以實現定義的成果。這可減少或消除組織摩擦。
  - 倡導您的團隊：與您的團隊保持互動，以便了解他們的表現以及是否有影響他們的外部因素。找出阻礙您團隊進度的障礙。代表您的團隊來協助解決障礙並消除不必要的負擔。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。
  - 鼓勵團隊發展：創造持續改進的文化，並主動從取得的進步和失敗中學習。鼓勵人員和組織的成長和發展。利用資料和軼事來發展願景和策略。

客戶範例

AnyCompany 零售正在透過快速重塑客戶體驗、提高生產力，以及透過生成式 AI 加速成長來實現業務轉型。

實作步驟

1. 建立單一執行緒領導，並指派主要執行倡議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。讓主要執行人員具有領導和做出關鍵決策的權限。
3. 驗證您的轉型策略非常清晰，並且由執行倡議者廣泛傳達到組織的各個層級。
    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. 將願景一致地傳達給負責策略各部分的所有團隊和個人。
4. 制定溝通規劃矩陣，指定需要傳遞給特定領導人、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. 一致且可靠地履行溝通計畫。
    b. 定期透過面對面活動設定並管理期望。
    c. 接受有關溝通有效性的反饋，並相應地調整溝通並進行計劃。
    d. 安排溝通活動以主動了解團隊的挑戰，並建立一致的回饋迴圈，以便在必要時糾正過程。
5. 從領導角度積極參與每項舉措，以驗證所有受影響的團隊都了解他們應負責實現的成果。
6. 在每次狀態會議上，執行倡議者都應該查找阻礙因素，檢查既定的指標、軼事或團隊的反饋，並衡量實現目標的進展情況。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS03-BP04 通訊是及時、清晰且可操作的
- OP11-BP01 擁有持續改善的流程
- OPS11-BP07 執行操作指標檢閱

相關文件：

- 解決組織困擾：高度一致
- 正在實施轉型：務實地應對變化
- 成為面向未來的企業

- 建置 時要避免的 7 個陷阱 CCOE
- 導覽雲端：成功的關鍵績效指標

相關影片：

- AWS re：Invent 2023：生成 AI 的領導者指南：使用歷史記錄塑造未來 （SEG204）

相關範例：

- Prosci：主要倡議者的角色和重要性

OPS03-BP02 團隊成員有權在結果處於風險時採取行動

由領導層灌輸的擁有權文化行為使所有員工都感到有權代表整個公司行事，超出了其定義的角色和責任範圍。員工可以採取行動，在風險出現時主動識別風險，並採取適當的措施。這樣的文化使員工能夠在了解情況的同時做出高價值決策。

例如，Amazon 使用領導方針作為指引，以推動員工在各種情況下前進、解決問題、處理衝突並採取行動的期望行為。

預期成果：領導層已經影響了一種新的文化，允許個人和團隊做出關鍵決策，即使在組織的較低級別(因為長期決策是使用可審計的權限和安全機制來定義的)。失敗並不氣餒，團隊反覆學習以改善他們的決策和應對措施，從而解決未來的類似情況。如果某人的行動導致可以使其他團隊受益的改進，則他們會主動分享此類行動中的知識。領導層會衡量運營改進，並激勵個人和組織採用此類模式。

常見的反模式：

- 在識別風險時，組織中沒有明確的指引或機制來進行應對。例如，當員工發現網路釣魚攻擊時，他們無法向安全團隊報告，導致組織中很大一部分人受到攻擊。這會導致資料外洩。
- 客戶抱怨服務無法使用，這主要源於部署失敗。SRE 您的團隊負責部署工具，且部署的自動復原已在其長期藍圖中。在最近推出的應用程式中，其中一位工程師設計了一種解決方案，將其應用程式自動還原到舊版本。雖然他們的解決方案可以成為SRE團隊的模式，但其他團隊不會採用，因為沒有追蹤此類改進的程序。組織繼續受到部署失敗的困擾，影響了客戶並導致進一步的負面情緒。
- 為了保持合規，您的資訊保護團隊會監督長期建立的程序，以代表連線至其 Amazon EC2 Linux 執行個體的運算子定期輪換共用SSH金鑰。資訊安全團隊需要幾天的時間才能完成輪換金鑰，而且會阻止您連線到這些執行個體。資訊區段內部或外部的任何人都不會建議在 上使用其他選項 AWS 來達成相同的結果。

建立此最佳實務的優勢：透過分散權限來制定決策，並授權團隊決定關鍵決策，您可以更快地解決問題，提高成功率。此外，團隊開始意識到主人翁精神，並且失敗是可以接受的。實驗成為文化中流砥柱。經理和董事並不覺得他們在工作的各個方面都受到了微觀管理。

未建立此最佳實務時的曝險等級：中

實作指引

1. 培養一種預期可能會發生失敗的文化。
2. 為組織內各個職能區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感，以便個人知道誰可以幫助他們促進分散式決策。
4. 定義您的單向和雙向門決策，以幫助個人了解何時需要升級到更高的領導層級。
5. 建立組織意識，讓所有員工有能力在結果出現風險時，在不同層面採取行動。為您的團隊成員提供管控文件、權限級別、工具和機會，以練習有效回應所需的技能。
6. 讓您的團隊成員有機會練習應對各種決策所需的技能。定義決策等級後，請執行演練日，以驗證所有個別參閱者是否了解並能夠演示該過程。
   a. 提供替代的安全環境，在其中可測試和培訓流程及程序。
   b. 承認並意識到，當結果出現預先定義的風險等級時，團隊成員有權採取行動。
   c. 透過指派權限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權限。
7. 讓團隊能夠分享學習經驗 (運營成功和失敗)。
8. 使團隊能夠挑戰現狀，並提供機制來追蹤和衡量改進，以及這些改進對組織的影響。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS01-BP06 在管理效益和風險時評估權衡
- OPS02-BP05 存在機制以識別責任和所有權

相關文件：

- AWS 部落格文章 | 敏捷企業
- AWS 部落格文章 | 衡量成功：悖論和計畫
- AWS 部落格文章 | 放手：實現團隊的自主權

- 集中化還是分散化？

相關影片：

- re：Invent 2023 | 如何避免破壞轉換（SEG201）
- re:Invent 2021 | Amazon 建置者資料中心：Amazon 的卓越營運
- 集中化與分散化

相關範例：

- 使用架構決策記錄來簡化軟體開發專案的技術決策

鼓勵 OPS03-BP03 升級

如果團隊成員認為預期成果存在風險並且達不到預期標準，領導層鼓勵他們將問題和疑慮向上呈報給更高層級的決策者和利益相關者。這是組織文化的一個特徵，並且在各個層面推動。應該儘早且經常向上呈報，以便識別風險，並防止風險引發事件。領導層不會譴責向上呈報問題的個人。

預期成果：整個組織的個人都很樂意將問題呈報給他們的直接和更高級別的領導層。領導層刻意和有意識地建立了期望，即他們的團隊在呈報任何問題時感到非常安全。存在一種機制來呈報組織內每個層級的問題。當員工呈報到經理時，他們共同決定影響的程度，以及是否應該呈報問題。為了啟動呈報，員工需要包含建議的工作計畫以解決問題。如果直接管理層沒有及時採取行動，則鼓勵員工在強烈認為組織面臨的風險需要呈報時，將問題呈報給最高領導層。

常見的反模式：

- 在雲端轉型計畫狀態會議期間，高層主管沒有提出足夠的探究性問題，以找出問題和阻礙發生的位置。只有好消息被呈現為狀態。CIO 已明確表示她只喜歡聽到好消息，因為提出的任何挑戰都會讓 CEO 認為程式失敗。
- 您是雲端操作工程師，您注意到應用程式團隊並未廣泛採用新的知識管理系統。該公司投入了一年時間和數百萬美元來實作這個新的知識管理系統，但人們仍然在本地編寫他們的執行手冊，並在組織的雲端共享中進行分享，這使得很難找到與所支援的工作負載相關的知識。您試圖引起領導層的注意，因為持續使用此系統可以提高運營效率。當您將其提交給領導該知識管理系統實作的主管時，她會譴責您，因為它讓投資受到質疑。
- 負責強化運算資源的資訊技術團隊已決定制定程序，該程序需要執行必要的掃描，以確保EC2執行個體在運算團隊釋出資源以供使用之前完全安全。這為要部署的資源建立了額外一週的時間延遲，這會中斷其 SLA。運算團隊害怕透過雲端將此問題呈報給 VP，因為這會讓資訊安全 VP 出醜。

建立此最佳實務的優勢：

複雜或重大問題在影響業務之前得到解決。浪費的時間更少。風險最小化。在解決問題時，團隊變得更加積極主動並注重結果。

未建立此最佳實務時的曝險等級：高

實作指引

在組織的各個層面自由呈報的意願和能力是一個組織和文化基礎，應該透過強調培訓、領導溝通、期望設定以及在組織各個層面的機制部署來有意識地進行發展。

實作步驟

1. 定義組織的政策、標準和期望。
   a. 確保廣泛採用和理解政策、期望和標準。
2. 在不符合標準時，鼓勵、培訓和授權工人儘早和頻繁呈報。
3. 組織認可儘早且頻繁呈報是最佳實務。接受向上呈報可能經證明是毫無根據的，然而有機會防止事件的發生好過於不向上呈報而錯過機會。
   a. 建立呈報機制 (例如 Andon Cord 系統)。
   b. 制定書面程序，定義向上呈報的時機與方式。
   c. 定義一系列具有越來越多權限可採取或批准行動的人員，以及每個利益相關者的聯絡資訊。
4. 當進行呈報時，它應該繼續下去，直到團隊成員確信透過領導層推動的行動緩解了風險。
   a. 呈報應包括：
      i. 情況描述和風險性質
      ii. 情況嚴重性
      iii. 誰或什麼受到影響
      iv. 影響有多大
      v. 發生影響時的緊迫性
      vi. 建議的補救措施和緩解計畫
   b. 保護向上呈報的員工。如果團隊成員圍繞無回應決策制定者或利益相關者向上呈報，則制定保護團隊成員免受報復的政策。制定機制以識別是否發生此情況，並適當地做出回應。
5. 鼓勵在組織生產的一切產品中建立持續改進回饋迴圈的文化。回饋迴圈充當負責個人的小型呈報，他們可確定改進機會，即使不需要升級。持續改進的文化迫使每個人都更加積極主動。
6. 領導層應定期重新強調政策、標準、機制，以及對公開呈報和持續回饋迴圈而不受報復的願望。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS02-BP05 機制用於請求新增、變更和例外狀況

相關文件：

- 如何培養一種持續改進並向 Andon 和呈報系統學習的文化？
- Andon Cord (IT 革命)
- AWS DevOps 指引 | 建立明確的呈報路徑，並鼓勵建設性的分歧

相關影片：

- Jeff Bezos 如何制定決策 (並提高速度)
- Toyota 產品系統：停止生產，一個按鈕以及一個 Andon 電氣板
- LEAN製造業的 Andon Cord

相關範例：

- 在 Incident Manager 中使用呈報計畫

OPS03-BP04 通訊是及時、清晰且可操作的

領導層負責建立強大而有效的溝通，尤其是當組織採用新策略、技術或工作方式時。領導者應該為所有員工設定期望，以實現公司目標。設計溝通機制，在負責運行由領導資助和贊助計劃的團隊之間可建立和保持意識。利用跨組織的多樣性，並用心聆聽多個獨特觀點。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。

預期成果：您的組織設計溝通策略以解決變更對組織的影響。團隊保持知情並積極繼續彼此合作，而不是彼此對抗。個人了解他們的角色對於實現既定目標是多麼重要。電子郵件只是一種被動的溝通機制，並相應地使用。管理層花時間與他們的個人貢獻者溝通，以幫助他們了解其責任、要完成的任務，以及他們的工作如何為整體使命做出貢獻。必要時，領導者直接在較小的場所與人們互動，以傳達訊息並確認這些訊息是否有效地傳遞。由於良好的溝通策略，組織的表現達到或超出領導層的期望。領導層鼓勵和尋求團隊內部和團隊之間的不同意見。

常見的反模式：

- 您的組織有五年計畫，可將所有工作負載遷移至 AWS。雲端的業務案例包括所有工作負載的 25% 進行現代化，以充分利用無伺服器技術。會將此策略CIO傳達給直屬員工，並期望每個領導者將此簡報逐級傳遞給經理、主管和個別貢獻者，而不需要任何當面溝通。退CIO一步並期望他的組織執行新策略。
- 領導層不提供或使用回饋機制，並且預期差距不斷增長，導致專案停滯。
- 系統會要求您對安全群組進行變更，但不會為您提供任何詳細資訊，說明需要進行哪些變更、變更可能對所有工作負載造成什麼影響，以及何時發生變更。管理員轉送來自 VP 的電子郵件， InfoSec 並新增「讓這種情況發生」訊息。
- 對您的遷移策略進行了變更，將規劃的現代化數量從 25% 降低到 10%。這對營運組織具有下游影響。他們沒有被告知這一策略變化，因此，他們還沒有準備好足夠的技術能力來支援更多的工作負載平移到 AWS中。

建立此最佳實務的優勢：

- 您的組織充分了解新的或變更的策略，他們會以強烈的動力採取相應的行動，以幫助彼此實現領導層設定的總體目標和指標。
- 存在的機制可用來及時通知團隊成員已知的風險和計畫的事件。
- 組織會更有效地採用新的工作方式 (包括對人員或組織、程序或技術的變更) 以及所需技能，而且您的組織可更快速地實現企業利益。
- 團隊成員了解溝通事項，他們可以更有效地完成工作。

未建立此最佳實務時的曝險等級：高

實作指引

若要實作此最佳實務，您必須與組織中的利益相關者合作，以同意溝通標準。在組織內將這些標準公告週知。對於任何重大的 IT 轉型，與忽略此實務的組織相比，已確立的規劃團隊可以更成功地管理變更對其人員的影響。大型組織在管理變更時更具挑戰性，因為與所有個別貢獻者一起建立對新策略的強有力支援至關重要。如果沒有此類轉型規劃團隊，領導層 100% 負責有效溝通。建立轉型規劃團隊時，指派團隊成員與所有組織領導層合作，以定義和管理各個層面的有效溝通。

客戶範例

AnyCompany 零售已註冊 AWS Enterprise Support，並取決於其他第三方供應商的雲端操作。該公司透過聊天和 chatops 作為其運營活動的主要溝通媒介。特定管道會填入提醒和其他資訊。人們必須展

開行動時，他們會明確說明預期成果，且在許多情況下他們會接收執行手冊或程序手冊以供使用。他們可使用變更行事曆來排程生產系統的重大變更。

實作步驟

1. 在組織內建立一個核心團隊，負責為組織內多個層面發生的變更制定和啟動溝通計畫。
2. 建立單線程擁有權以實現監督。賦予各個團隊獨立創新的能力，並平衡使用一致的機制，從而實現適當的檢查水平和方向性願景。
3. 與組織中的利益相關者合作，以同意溝通標準、實務和計畫。
4. 確認核心溝通團隊是否與組織和計畫領導層協作，代表領導者為適當的員工製作訊息。
5. 透過公告、共用行事曆、全手會議和面對面或 one-on-one方法，建立策略溝通機制來管理變革，讓團隊成員對應該採取的動作有適當的期望。
6. 提供必要的背景知識、詳細資訊和時間 (如果可能的話)，以確定是否需要採取行動。當需要採取行動時，請提供所需的動作及其影響。
7. 實作可促進戰術溝通的工具，例如內部聊天、電子郵件和知識管理。
8. 實施機制以衡量和驗證所有溝通是否都能達到預期成果。
9. 建立一個回饋迴圈，以衡量所有溝通的有效性，尤其是當溝通與整個組織中的變革抵制有關時。
10. 對於所有 AWS 帳戶，請建立帳單、安全和操作的替代聯絡人。理想情況下，每個聯絡人應該是電子郵件分發，而不是特定的個人聯絡人。
11. 建立升級和反向升級通訊計劃，與您的內部和外部團隊互動，包括 AWS 支援和其他第三方提供者。
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. 排定可重複動作的優先順序，以便大規模安全地自動化。
14. 在具有自動化動作的情況下進行通訊時，通訊目的應該是通知團隊、進行稽核或作為變更管理流程的一部分。
15. 分析來自提醒系統的通訊，找出不斷建立的誤報或提醒。移除或變更這些提醒，使其僅在需要人為介入時啟動。如果啟動了提醒，請提供執行手冊或程序手冊。
    a. 您可以使用 AWS Systems Manager 文件，為提醒建立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劃事件的通知，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，在計劃性事件發之前傳送通知。
    a. AWS Chatbot 可用於在組織訊息傳遞平台中傳送提醒並回應事件。
17. 提供可存取的資訊來源，您可以在其中發現計劃的事件。提供來自相同系統之計劃事件的通知。

a. AWS Systems Manager 變更行事曆可用於在可能發生變更時建立變更時段。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，以便讓他們可以採取動作。

a. 您可以訂閱 AWS 安全公告，以接收 AWS上的漏洞通知。

19尋求不同的意見和觀點：鼓勵每個人的貢獻。為代表人數不夠的群體提供溝通機會。在會議中輪換角色和職責。

a. 詳細闡述角色和職責：為團隊成員提供機會，讓他們承擔他們可能不會承擔的角色。他們會透過角色，以及與他們可能不會與之互動的新團隊成員互動，而獲得經驗和觀點。他們還將自己的經驗和觀點帶到新的角色，並帶給和他們互動的團隊成員。隨著觀點增加，確定新興的業務機會或者新的改進機會。在團隊成員之間輪流處理其他人通常執行的常見任務，以了解執行這些任務的需求和影響。

b. 提供安全且友善的環境：制定政策和控制措施，以保護組織內團隊成員的心理和身體安全。團隊成員應該能夠在不擔心報復行為的情況下進行互動。當團隊成員感到安全且受歡迎時，他們才更有可能參與進來並具備生產力。您的組織越多樣化，您就越能了解所支援的人員，包括您的客戶。當您的團隊成員感到安心、可以自在的暢所欲言，而且有信心他們的聲音不會被淹沒，他們才更有可能分享寶貴的洞見 (例如，行銷機會、可及性的需求、尚未有服務的市場區段、環境中未確認的風險)。

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會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的便利性支援，將可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 提供執行贊助
- OPS07-BP03 使用 Runbook 執行程序
- OPS07-BP04 使用教戰手冊調查問題

相關文件：

- AWS 部落格文章 | 問責制和授權是高效能敏捷組織的關鍵
- AWS Executive Insights | 學習擴展創新，而不是複雜性 | 單線程領導者
- AWS 安全公告

- 開啟 CVE
- AWS Support Slack 中的應用程式以管理支援案例
- 使用 管理 Slack 頻道中的 AWS 資源 AWS Chatbot

相關範例：

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

相關服務：

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

鼓勵 OPS03-BP05 實驗

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曝險等級：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 零售鼓勵實驗。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. 您可以使用單獨的 AWS 帳戶 來建立沙盒環境以供實驗。 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 提供建立特徵旗標的功能。
   b. Amazon CloudWatch Evidently 可用於在有限的部署上執行 A/B 測試。
   c. 可以使用 AWS Lambda 版本部署新版本的函數以進行 beta 測試。

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 執行事後分析 - 從事件中學習是創新以及實驗的重要驅動力。
- OPS11-BP03 實作意見回饋循環 - 回饋迴圈是實驗的重要組成部分。

相關文件：

- Amazon 文化內貌：實驗、失敗和客戶至上

- 在 中建立和管理沙盒帳戶的最佳實務 AWS
- 建立由雲端啟用的實驗文化
- 在 SulAmérica Seguros 啟用雲端的實驗和創新
- 實驗越多，失敗越少
- 使用多個帳戶組織您的 AWS 環境 - Sandbox OU
- 使用 AWS AppConfig 特徵旗標

相關影片：

- AWS 在 Air ft. Amazon CloudWatch Evidently | AWS Events
- AWS 與 Jira 整合的 On Air San Fran Summit 2022 ft. AWS AppConfig Feature Flags
- AWS re：Invent 2022 - 部署不是版本：控制具有功能旗標的啟動 BOA3（05-R）
- 以程式設計方式建立 AWS 帳戶 具有 的 AWS Control Tower
- 設定使用最佳實務的多帳戶 AWS 環境 AWS Organizations

相關範例：

- AWS 創新沙盒
- End-to-end 適用於電子商務的個人化 101

相關服務：

- Amazon CloudWatch Evidently
- AWS AppConfig
- AWS Control Tower

0OPS03-BP06 鼓勵團隊成員維護和發展其技能集

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 線上技術講座、AWS 事件和網路研討會，以及 AWS Well-Architected Labs，提供指導、範例和詳細的演練來教育您的團隊。

諸如 AWS Support (AWS re:Post、AWS Support Center) 等資源和 AWS 文件有助於消除技術障礙並改善操作。 AWS Support 請透過 AWS Support Center 聯絡 以取得問題的相關協助。

AWS 也會透過 Amazon Builders' Library AWS 中的 操作，以及透過AWS 部落格和官方 AWS Podcast 的各種其他實用教育材料，分享我們學到的最佳實務和模式。

AWS 培訓 和 認證包括透過自定進度數位課程進行的免費訓練，以及依角色或網域區分的學習計畫。您也可以註冊講師引導式訓練，以進一步支援團隊 AWS 技能的發展。

預期成果：您的組織不斷評估技能差距，並透過結構化預算和投資進行彌補。團隊透過提高技能的活動來鼓勵和激勵其會員，例如獲得領先的行業認證。團隊利用專用的跨共享知識計畫，例如 lunch-and-learns、臨場日、駭客松和遊戲日。您組織的 會保留其知識系統 up-to-date，並與跨訓練團隊成員相關，包括新進人員入職訓練。

常見的反模式：

- 在沒有結構化培訓計畫和預算的情況下，團隊會遇到不確定性，因為他們試圖跟上技術發展的步伐，從而增加損耗。
- 作為遷移至 的一部分 AWS，您的組織在團隊之間表現出技能差距和不同的雲端流暢性。如果不努力提高技能，團隊會發現自己被雲端環境的傳統和效率低下的管理所累，這會導致操作員的工作量增加。這種倦怠會加劇員工的不滿。

建立此最佳實務的優勢：當您的組織有意識地投資於改善其團隊的技能時，它也有助於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建立營運能力，讓團隊為處理各種事件做好準備。團隊有意識地投資於最佳實務的實作和發展。團隊士氣高漲，團隊成員重視自己對業務的貢獻。

未建立此最佳實務時的曝險等級：中

實作指引

為了採用新技術、推動創新、並跟上需求和責任變化的步伐來支援您的工作負載，請不斷投資於團隊的專業成長。

實作步驟

1. 使用結構化雲端宣傳計畫：AWS Skills Guild 提供諮詢培訓，以增加雲端技能信心並激發持續學習的文化。
2. 提供教育資源：提供專門的結構化時間，以及培訓教材和實驗室資源的存取權，並支援參與會議和專業組織，這些會議和組織可為教育工作者和同儕提供學習的機會。為資淺團隊成員提供接近資深

團隊成員的機會，讓資深團隊成員成為導師，或允許資淺團隊成員參觀資深團隊成員的工作，並接觸他們的方法和技能。鼓勵學習與工作不直接相關的內容，以便取得更廣泛的視野。

3. 鼓勵使用專家技術資源：利用諸如 AWS re:Post 之類的資源來存取精選知識和充滿活力的社群。
4. 建置和維護 up-to-date知識儲存庫：使用知識共享平台，例如 wiki 和 Runbook。使用 AWS re:Post Private 建立您自己的可重複使用的專家知識來源，以簡化協同合作、提高生產力並加速員工入職。
5. 團隊教育和跨團隊參與：規劃團隊成員的持續教育需求。為團隊成員提供機會 (暫時或永久地) 加入其他團隊，以分享讓整個組織受益的技能和最佳實務。
6. 支援追求和維持產業認證：支援團隊成員取得與維持可驗證所學知識並認可其成就的產業認證。

實作計劃的工作量：高

資源

相關的最佳實務：

- OPS03-BP01 提供執行贊助
- OPS11-BP04 執行知識管理

相關文件：

- AWS 白皮書 | 雲端採用架構：個人視角
- 投資持續學習以發展組織的未來
- AWS Skills Guild
- AWS 培訓 和 認證
- AWS Support
- AWS re：Post
- AWS 資源中心入門
- AWS 部落格
- AWS 雲端 合規
- AWS 文件
- 官方 AWS Podcast 。
- AWS 線上技術講座
- AWS 事件和網路研討會
- AWS Well-Architected 實驗室

- Amazon 建置者資料中心

相關影片：

- AWS re:Invent 2023 | 以雲端的速度重塑技能：將員工變為企業家
- WS re:Invent 2023 | 透過遊戲化建立好奇心文化

OPS03-BP07 資源團隊適當

提供適量訓練有素的團隊成員，並提供工具和資源來支援工作負載需求。負擔過重的團隊成員會增加人為錯誤的風險。對工具和資源的投資 (例如自動化) 可以提升團隊的效率，並協助他們支援更多工作負載，而不需要額外的生產能力。

預期成果：

- 您已適當地配置團隊，以取得他們 AWS 根據您的遷移計劃操作工作負載所需的技能組合。由於您的團隊在遷移專案期間自我擴展，因此他們已精通業務計劃在遷移或現代化其應用程式時使用的核心 AWS 技術。
- 您已經仔細調整人員配置計畫，以利用自動化和工作流程來有效利用資源。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。
- 隨著營運優先順序的轉變，會主動識別任何資源人員配置限制，以保護業務計畫的成功。
- 審核報告操作辛勞的操作指標 (例如待命疲勞或過度呼叫)，以驗證員工是否不堪重負。

常見的反模式：

- 當您接近多年雲端遷移計畫時，您的員工尚未提高 AWS 技能，這些風險支援工作負載並降低員工士氣。
- 您的整個 IT 組織正在轉向敏捷的工作方式。企業正優先考慮產品組合，並為需要首先開發的功能設定指標。敏捷流程不需要團隊將故事點分配給他們的工作計畫。因此，不可能了解下一工作量所需的能力水平，或者您是否擁有分配給該工作的正確技能。
- 您正在讓 AWS 合作夥伴遷移工作負載，而且一旦合作夥伴完成遷移專案，您就沒有團隊的支援轉換計畫。您的團隊難以高效地支援工作負載。

建立此最佳實務的優勢：您的組織中有適當技能的團隊成員來支援工作負載。資源配置可適應不斷變化的優先順序，而不會影響效能。結果是團隊精通工作負載的支援，同時最大限度地利用時間專注於為客戶創新，從而提高員工滿意度。

未建立此最佳實務時的曝險等級：中

實作指引

雲端遷移的資源規劃應該在組織層級進行，它符合遷移計畫以及為支援新雲端環境而實作的所需操作模型。這應該包括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者應該為領導雲端採用的工程師規劃技能差距分析、培訓和角色定義。

實作步驟

1. 使用諸如員工生產力等相關營運指標 (例如，支援工作負載的成本或事件期間所花費的操作員時數)，定義團隊成功的標準。
2. 定義資源容量計畫與檢驗機制，以驗證合格容量的適當平衡在需要時可用，並且可隨時間進行調整。
3. 建立機制 (例如，向團隊傳送每月調查問卷)，以了解影響團隊的工作相關挑戰 (例如責任增加、技術變化、人員損失或支援的客戶增加)。
4. 使用這些機制與團隊互動，並發現可能導致員工生產力挑戰的趨勢。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。找出阻礙您團隊進度的障礙。
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。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS我們鼓勵 03-BP06 團隊成員維護和提升其技能組合
- OPS09-BP03 檢閱操作指標並排定改善優先順序
- OPS10-BP01 使用事件、事件和問題管理的程序
- OPS10-BP07 自動化對事件的回應

相關文件：

- AWS 雲端 採用架構：人員觀點
- 成為面向未來的企業
- 優先考慮員工的技能以推動業務增長

- 高效能組織 - Amazon 雙披薩團隊
- 雲端成熟企業如何成功

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型決策。確保監控活動與業務目標保持一致的最有效方法之一，是透過定義和監控關鍵績效指標（KPIs）。

預期成果：有效率的可觀測性實務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 KPIs：在沒有明確的情況下工作KPIs可能會導致監控太多或太少，遺失重要訊號。
- 靜態 KPIs：不會KPIs隨著工作負載或業務目標的發展而重新檢視或精簡。
- 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：

- 問題識別的易用性：企業KPIs通常會比技術指標更清晰地呈現問題。企業的下降KPI可以比篩選多個技術指標更有效找出問題。
- 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

若要有效定義工作負載KPIs：

1. 從業務成果開始著手：在深入研究指標之前，請先了解預期業務成果。想要增加銷售量、提高使用者參與度，還是加快回應時間？
2. 將技術指標與業務目標相關聯：並非所有技術指標都會直接影響業務成果。識別這樣做的人，但使用 業務 識別問題通常更為簡單KPI。
3. 使用 Amazon CloudWatch：使用 CloudWatch 來定義和監控代表您 的指標KPIs。
4. 定期檢閱和更新 KPIs：隨著工作負載和業務的發展，請保持KPIs關聯。
5. 讓利益相關者參與：讓技術和業務團隊參與定義和檢閱 KPIs。

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 可觀測性技能建置器課程

相關影片：

- 研擬可觀測性策略

相關範例：

- 一個可觀測性研討會

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。發出遙測至關重要，它為您的應用程式狀態以及技術和業務成果的實現提供了可行洞見。從疑難排解到測量新功能的影響，或確保與業務金鑰效能指標（KPIs）保持一致，應用程式遙測會告知您建置、操作和發展工作負載的方式。

指標、日誌和追蹤構成了可觀測性的三個主要支柱。其可作為描述應用程式狀態的診斷工具。隨著時間的推移，它們有助於建立基準並識別異常。不過，為了確保監控活動與業務目標之間的一致性，定義和監控 至關重要KPIs。與技術指標相比，企業KPIs通常更容易識別問題。

其他遙測類型，例如真實使用者監控（RUM）和合成交易，可補充這些主要資料來源。RUM 提供即時使用者互動的洞見，而合成交易模擬潛在的使用者行為，有助於在實際使用者遇到瓶頸之前進行偵測。

預期成果：獲得工作負載效能且可付諸行動的洞見。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。
- 使用者報告問題：缺乏透過遙測和業務KPI監控主動偵測問題的跡象。

建立此最佳實務的優勢：

- 知情決策：透過遙測和業務的洞察KPIs，您可以做出資料驅動的決策。
- 改善運作效率：資料驅動的資源利用率可帶來成本效益。
- 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曝險等級：高

實作指引

若要為工作負載實作應用程式遙測，請使用 AWS Amazon CloudWatch 和 等服務AWS X-Ray。Amazon CloudWatch 提供全方位的監控工具套件，可讓您在內部部署環境中觀察資源 AWS 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。在串聯中， AWS X-Ray 可讓您追蹤、分析和偵錯應用程式，讓您深入了解工作負載的行為。透過服務地圖、延遲分佈和追蹤時間表等功能， AWS X-Ray 可讓您深入了解工作負載的效能和影響工作負載的瓶頸。

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 ：代理 CloudWatch 程式對於從您的工作負載及其基礎基礎設施中取得系統和應用程式指標和日誌至關重要。 CloudWatch 代理程式也可以用來收集 OpenTelemetry 或 X-Ray 追蹤，並將其傳送至 X-Ray。
3. 實作日誌和指標的異常偵測：使用CloudWatch 日誌異常偵測和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異常並發出提醒，進而提升監控能力，並加快對潛在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. 安全敏感日誌資料：使用 Amazon CloudWatch Logs 資料保護來遮罩日誌中的敏感資訊。此功能可在存取敏感資料前進行自動偵測和遮罩，從而有助於維護隱私權與合規性。實作資料遮罩，以安全地處理和保護敏感詳細資訊，例如個人識別資訊 （PII）。
5. 定義和監控業務 KPIs：建立與您的業務成果相符的自訂指標。
6. 使用 來測試您的應用程式 AWS X-Ray：除了部署 CloudWatch代理程式之外，測試應用程式以發出追蹤資料也很重要。此程序可提供工作負載行為和效能的進一步洞見。
7. 將整個應用程式的資料收集標準化：將整個應用程式的資料收集實務標準化。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： AWS 帳戶 使用 Amazon CloudWatch 跨帳戶可觀測性增強跨多個 的監控效率。使用此功能，您可以將不同帳戶的指標、日誌和警示合併為單一檢視，可簡化管理並改善組織 AWS 環境中已識別問題的回應時間。
9. 分析資料並採取行動：資料收集和標準化完成後，請使用 Amazon CloudWatch 進行指標和日誌分析，以及AWS X-Ray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。

實作計劃的工作量：高

資源

相關的最佳實務：

- OPS04-BP01 定義工作負載 KPIs
- OPS04-BP03 實作使用者活動遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作交易可追蹤性

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 的新功能 CloudWatch
- 新功能 AWS X-Ray

相關影片：

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

相關範例：

- 一個可觀測性研討會
- AWS 解決方案庫：使用 Amazon 進行應用程式監控 CloudWatch

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）和合成交易可做為此目的的強大工具。RUM 提供有關真實使用者互動的資料，授予使用者滿意度的未篩選觀點，而合成交易模擬使用者互動，即使在影響真實使用者之前，也有助於偵測潛在問題。

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的應用程式（RUM）：
  - 延遲問題偵測：如果沒有 RUM，在使用者投訴之前，您可能不會發現效能瓶頸或問題。這種被動回應的方式可能導致客戶不滿意。
  - 缺乏使用者體驗洞察：不使用 RUM表示您遺失了關鍵資料，這些資料顯示真實使用者如何與您的應用程式互動，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協助精簡和增強整體使用者體驗的持續意見回饋。
- 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 合成交易進行使用者活動遙測， AWS 提供 Amazon CloudWatch RUM 和 Amazon CloudWatch Synthetics 等服務。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 合成：建立 Canary 或指令碼常式，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. 分析資料並對資料採取行動：利用 RUM和 合成交易中的資料來取得洞察，並在偵測到異常時採取修正措施。使用 CloudWatch 儀表板和警示來隨時掌握最新資訊。

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：

- 使用 Amazon 透過最終使用者洞察最佳化應用程式 CloudWatch RUM
- AWS Air ft 上的 。 Amazon 的真實使用者監控 CloudWatch

相關範例：

- 一個可觀測性研討會
- Amazon CloudWatch RUM Web Client 的 Git 儲存庫
- 使用 Amazon CloudWatch Synthetics 來測量頁面載入時間

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，相依性遙測至關重要，它提供與相依性相關的可連線性、逾時和其他重要事件的寶貴洞見DNS，例如 、 資料庫或第三方 APIs。檢測應用程式以產生有關這些相依性的指標、日誌和追蹤，便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。

預期成果：確保工作負載所依賴的相依性如預期般正常運作，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：

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

建立此最佳實務的優勢：

- 改善工作負載可靠性：確保外部相依性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性相關問題，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：了解外部相依性的可擴展性限制與效能特性。

未建立此最佳實務時的曝險等級：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，實作相依性遙測。將相依性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性狀態的洞見。使用 AWS 工具來探索和量化相依性無法視需要交付時的影響。持續重新檢視您的策略，以考量優先順序、目標和獲得的洞見的變化。

實作步驟

若要有效實作相依性遙測：

1. 識別外部相依性：與利益相關者協作，共同找出工作負載所依賴的外部相依性。外部相依性可以涵蓋外部資料庫、第三方 APIs、其他環境的網路連線路由，以及 DNS服務等服務。實現有效相依性遙測的第一步，就是徹底了解這些相依性。
2. 擬訂監控策略：清楚了解外部相依性之後，就可以為其量身打造監控策略。這涉及了解每個相依性的重要性、其預期行為，以及任何相關聯的服務層級協議或目標（SLA 或 SLTs）。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：使用網際網路監控和網路監控，全面了解全球網際網路和網路狀況。這些工具可協助您了解並回應影響外部相依性的停機、中斷或效能降低。
4. 透過 隨時掌握最新資訊AWS Health Dashboard：它在 AWS 遇到可能會影響您服務的事件時提供提醒和修復指引。
    a. AWS Health 使用 Amazon EventBridge 規則 監控事件，或以程式設計方式與 AWS Health API 整合，以便在接收 AWS Health 事件時自動執行動作。這些動作可以是一般動作 (例如將所有規劃的生命週期事件訊息傳送至聊天介面) 或是特定動作 (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，請在帳戶之間彙總 AWS Health 事件。
5. 透過 AWS X-Ray：提供應用程式及其基礎相依性執行方式的洞見，為您的應用程式進行測試 AWS X-Ray 。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 DevOpsGuru ：此機器學習驅動服務可識別操作問題、預測可能發生重大問題的時間，並建議要採取的特定動作。對於獲得相依性洞見並確保其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性相關的指標和日誌。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性更新或變更，便驗證其效能並檢查是否符合您的應用程式需求。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS04-BP01 定義工作負載 KPIs
- OPS04-BP02 實作應用程式遙測
- OPS04-BP03 實作使用者活動遙測
- OPS04-BP05 實作交易可追蹤性
- OP08-BP04 建立可執行的提醒

相關文件：

- Amazon 個人 AWS Health Dashboard 使用者指南
- AWS 網際網路監控使用者指南
- AWS X-Ray 開發人員指南
- AWS DevOpsGuru 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 DevOpsGuru 簡介
- 使用 大規模管理資源生命週期事件 AWS Health

相關範例：

- AIOps使用 Amazon DevOpsGuru 取得營運洞見
- AWS Health 察覺
- 使用標籤型篩選來管理大規模 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。計算所有元件後，請利用 AWS X-Ray 和 等工具 OpenTelemetry 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。與開發人員進行定期檢閱，並使用 Amazon DevOpsGuru、X-Ray Analytics 和 X-Ray Insights 等工具來補充這些討論，以協助探索更深入的調查結果。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. 測試您的服務：確認從 AWS Lambda函數到EC2執行個體 的每個服務都會傳送追蹤資料。您測試的服務越多，檢視越清晰 end-to-end。
3. 整合CloudWatch 實際使用者監控和合成監控 ：將實際使用者監控 （RUM） 和合成監控與 X-Ray 整合。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 ：代理程式可以從 X-Ray 或 傳送追蹤 OpenTelemetry，增強所取得洞見的深度。
5. 使用 Amazon DevOpsGuru ： DevOpsGuru 使用 X-Ray 的資料 CloudWatch AWS Config， AWS CloudTrail 並提供可行的建議。

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 中設定警示CloudWatch是否有異常模式或延長延遲，允許主動解決問題。
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 DevOpsGuru 使用者指南

相關影片：

- 使用 AWS X-Ray Insights
- AWS Air ft 上的 。 可觀測性：Amazon CloudWatch 和 AWS X-Ray

相關範例：

- 測試您的應用程式 AWS X-Ray

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 測試和驗證變更
- 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 實作實務來改善程式碼品質
- OPS05-BP08 使用多個環境
- OPS05-BP09 進行頻繁、小型、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) 來管理程式碼和其他成品，例如基礎結構的版本控制 AWS CloudFormation 範本。

期望的結果：您的團隊在程式碼上進行協作。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，以規劃您未來的工作。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，您可以輕鬆還原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：

- 簡介 AWS CodeCommit

OPS05-BP02 測試和驗證變更

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。除了應用程式碼變更之外，測試應包括基礎設施、組態、安全控制和操作程序。測試有多種形式，從單位測試到軟體元件分析 （SCA）。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。

您的組織必須針對所有軟體成品制定測試標準。自動化測試可減少辛勞並避免手動測試錯誤。在某些情況下可能需要手動測試。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。

預期成果：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時未進行任何測試。它無法在生產環境中執行，這會導致中斷。
- 新的安全群組使用 部署， AWS CloudFormation 無需在生產前環境中進行測試。安全群組會讓您的客戶無法存取您的應用程式。
- 方法被修改，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：降低了軟體部署的變更失敗率。軟體品質獲得改善。開發人員更能感知其程式碼的可行性。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曝險等級：高

實作指引

作為持續整合實務的一部分，對從應用程式碼到基礎設施的所有變更進行測試。發布測試結果，以便開發人員快速獲得意見回饋。您的組織具有所有變更都必須通過的測試標準。

透過 Amazon Q Developer，利用生成式 AI 的強大功能，提升開發人員生產力和程式碼品質。Amazon Q Developer 包括程式碼建議的產生 (以大型語言模型為基礎)、單元測試的生產 (包括邊界條件)，以及透過偵測和修復安全漏洞增強程式碼安全性。

客戶範例

作為持續整合管道的一部分， AnyCompanyRetail 會對所有軟體成品執行多種類型的測試。他們實行測試驅動型開發，所以所有軟體都有單元測試。成品建置完成後，它們會執行 end-to-end測試。在第一輪測試完成後，他們會執行靜態應用程式安全性掃描，以尋找已知的漏洞。通過每個測試門時，開發人員會收到訊息。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. 與組織中的利益相關者合作制定軟體成品的測試標準。所有成品都應該通過哪些標準測試？ 測試涵蓋範圍中是否必須包含合規性或管控要求？ 您是否需要進行程式碼品質測試？ 測試完成時，誰需要得知？
    1. AWS 部署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。每組測試應在十分鐘內完成。測試應執行為整合管道的一部分。
    a. 使用 Amazon Q Developer，這是一種生成式 AI 工具，可協助建立單元測試案例 (包括邊界條件)、使用程式碼和註解產生函數，以及實作眾所周知的演算法。
    b. 使用 Amazon CodeGuru Reviewer 測試您的應用程式程式碼是否有瑕疵。
    c. 可使用 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 實作實務來改善程式碼品質
- 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
- Amazon Q Developer 現已正式推出，包含可重新構想開發人員體驗的新功能預覽
- 在您的 中使用 Amazon Q 開發人員的最終備忘單 IDE
- 左移工作負載，利用 AI 建立測試
- Amazon Q 開發人員中心
- 使用 Amazon 更快建置應用程式的 10 種方法 CodeWhisperer
- 使用 Amazon 超越程式碼涵蓋範圍 CodeWhisperer
- Amazon Prompt Engineering 的最佳實務 CodeWhisperer
- 使用 TaskCat 和 的自動 AWS CloudFormation 測試管道 CodePipeline
- 使用開放原始碼 SCA、 SAST和 DAST工具建置 end-to-end AWS DevSecOps CI/CD 管道
- 開始測試無伺服器應用程式
- CI/CD 管道是我的發行主管
- 《在 AWS 上實行持續整合和持續交付》白皮書

相關影片：

- API使用 Amazon Q Developer Agent for Software Development 實作
- 安裝、設定和搭配 使用 Amazon Q 開發人員 JetBrains IDEs（方法）
- 掌握 Amazon CodeWhisperer YouTube 播放清單的藝術
- AWS re：Invent 2020：可測試的基礎設施： 上的整合測試 AWS
- AWS ANZ Summit 2021 - 使用 CDK和 測試驅動開發推動測試優先策略
- 使用 測試您的基礎設施作為程式碼 AWS CDK

相關資源：

- 使用生成 AI 與 Amazon 建置應用程式 CodeWhisperer
- 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
- GitHub 使用 對 Node.js 應用程式執行單位測試 AWS CodeBuild
- 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。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，或在事件期間變更日誌詳細資訊等級。

組態應以已知且一致的狀態部署。應該使用自動化檢測來持續監控跨環境和區域的資源組態。這些控制項應定義為已自動化的程式碼和管理，以確保規則在各個環境中一致套用。組態變更應透過商定的變更控制程序進行更新，並一致地套用，以遵守版本控制。應用程式組態的管理應該獨立於應用程式和基礎設施程式碼。這允許在多個環境中進行一致的部署。組態變更不會導致重建或重新部署應用程式。

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。

在 上 AWS，您可以使用 AWS Config 持續監控帳戶和區域 之間的 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 Conformance Packs。

對於在 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的動態組態，您可以使用 AWS AppConfig來設定、驗證、部署和監控整個環境。

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 Canary 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 計畫變更失敗
- OPS06-BP02 測試部署
- OPS06-BP03 採用安全部署策略
- 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 工作負載的主動管理和合規
- AWS re：Invent 2020：使用 以程式碼形式達成合規 AWS Config
- 使用 管理和部署應用程式組態 AWS AppConfig

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

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。

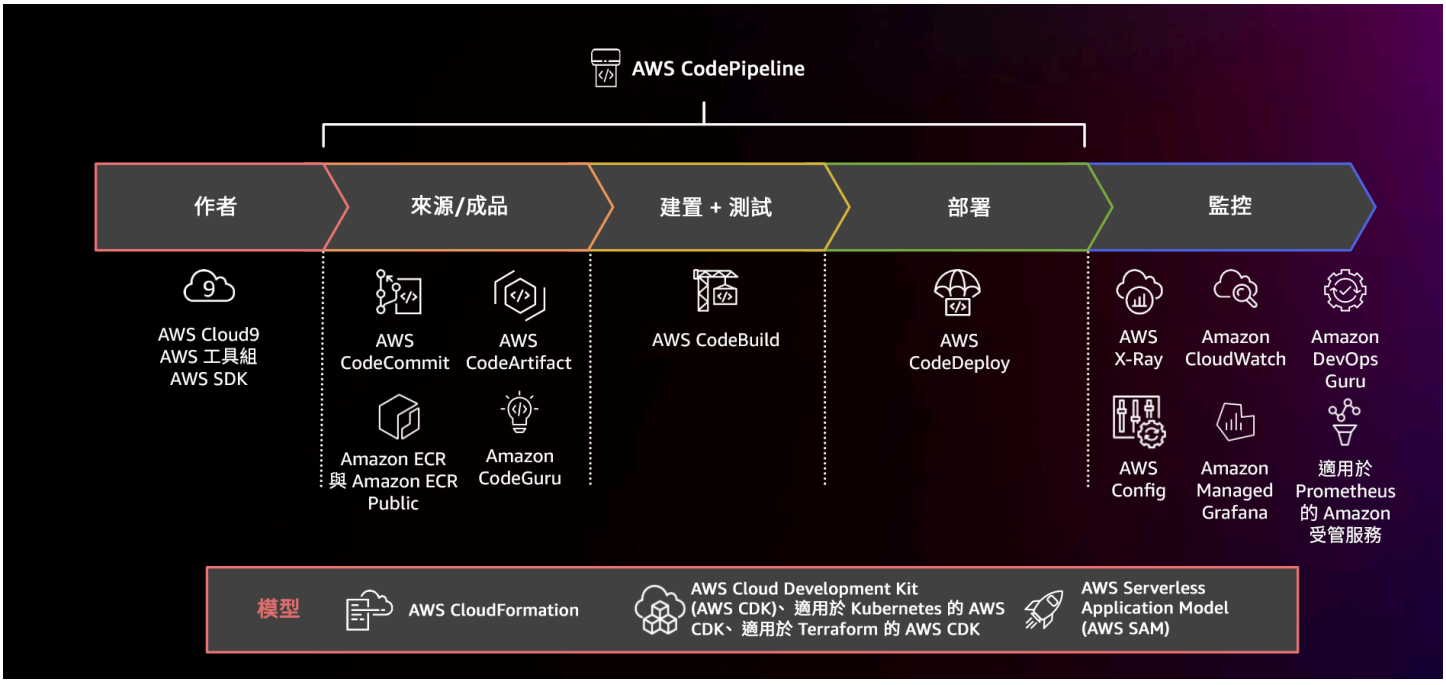
建立此最佳實務的優勢：透過提供用於管理建置和部署活動的機制，您可以減少執行重複性任務的工作量，讓團隊成員專注於高價值的創意任務，並減少手動程序導致的錯誤。

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



顯示使用 AWS CodePipeline 和 相關服務的 CI/CD 管道的圖表

1. AWS CodeCommit 用於版本控制、儲存和管理資產（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、執行單位測試，並產生準備好部署的成品。
3. 使用 CodeDeploy 作為部署服務，將應用程式部署自動化至 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 - AWS Well-Architected 的最佳實務 DevOps AWS

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 提供管道來更新機器映像。作為修補程式管理的一部分，請考慮使用AMI 映像管道或容器映像搭配 Docker 映像管道 的 Amazon Machine Images （AMIs），同時 AWS Lambda 提供自訂執行期的模式和其他程式庫來移除漏洞。

您應該使用 Amazon Image Builder 管理 Amazon Machine Images for Linux 或 Windows Server EC2 映像的更新。您可以搭配現有的管道使用 Amazon Elastic Container Registry （ Amazon ECR ） 來管理 Amazon ECS映像和管理 Amazon EKS映像。Lambda 包含版本管理功能。

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 上 AWS，您可以使用 AWS Systems Manager Patch Manager 自動化修補受管系統的程序，並使用 Systems Manager Maintenance Windows 排程活動。

預期結果：您的 AMI 和容器映像會修補 up-to-date，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。
- 您尚未修補持久性執行個體上的商業 off-the-shelf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

對於 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。

對於 Systems Manager Patch Manager：

1. 建立修補基準。
2. 選取修補操作方法。
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 Image Builder 建立EC2映像管道
- 建立容器映像管道
- AWS Systems Manager 修補程式管理員
- 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：

- 上的無伺服器應用程式的 CI/CD AWS
- 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。它們會隨著最佳實務的演變進行記錄和保存 up-to-date。

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。記錄和更新設計標準可讓組織 up-to-date符合最佳實務、安全和合規要求。

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、操作程序以及指引和管控要求。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。讓團隊得知發布的內容。擁有機制，以在新最佳實務出現時保持設計標準 up-to-date。

客戶範例

AnyCompany Retail 有一個跨職能架構團隊，可建立軟體架構模式。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益相關者合作，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. 在識別新的最佳實務時，有適當的機制來保持設計標準 up-to-date。
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益相關者協調和合作。

資源

相關的最佳實務：

- OPS01-BP03 評估治理要求 - 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 確保對操作就緒狀態進行一致審核 - 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 擁有持續改善的流程 - 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：

- AWS Backup使用 自動化 AWS Service Catalog

- AWS Service Catalog 帳戶工廠增強型
- Expedia Group 如何使用 建置資料庫即服務 （ DBaaS ） 方案 AWS Service Catalog
- 維護使用雲端架構模式的可見性
- 簡化在 AWS Organizations 設定中共用 AWS Service Catalog 產品組合

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 產品組合

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 實作實務來改善程式碼品質

實作相關實務以提高程式碼品質，並盡可能減少缺陷。部分範例包括測試驅動的開發、程式碼審查、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。

預期成果：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。變更會自動部署至生產環境，並導致中斷。
- 新應用程式是在沒有任何單位 end-to-end或整合測試的情況下開發的。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，以解決問題。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。程式碼品質有助於使用最佳實務，例如配對程式設計、程式碼檢閱以及 AI 生產力工具的實作。

未建立此最佳實務時的曝險等級：中

實作指引

實作實務以提高程式碼品質，在部署之前將故障降至最低。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。

透過 Amazon Q Developer，利用生成式 AI 的強大功能，提升開發人員生產力和程式碼品質。Amazon Q Developer 包括程式碼建議的產生 (以大型語言模型為基礎)、單元測試的生產 (包括邊界條件)，以及透過偵測和修復安全漏洞增強程式碼安全性。

客戶範例

AnyCompany 零售採用數種做法來改善程式碼品質。它們採用了測試驅動的開發做為撰寫應用程式的標準。對於某些新功能，它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，採用像是測試驅動開發、程式碼檢閱和配對程式設計的程式碼品質實務。使用這些技術來改善軟體品質。
   a. 使用 Amazon Q Developer，這是一種生成式 AI 工具，可協助建立單元測試案例 (包括邊界條件)、使用程式碼和註解產生函數、實作眾所周知的演算法、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)、記錄程式碼以及更快速地學習第三方程式碼程式庫。
   b. Amazon CodeGuru Reviewer 可以使用機器學習提供 Java 和 Python 程式碼的程式設計建議。
   c. 可以使用 AWS Cloud9 建立共用的開發環境，讓您可以在其中協同合作開發程式碼。

實作計劃的工作量：中。有許多方式可以實作此最佳實務，但是要讓整個組織採用可能會是一項挑戰。

資源

相關的最佳實務：

- OPS05-BP02 測試和驗證變更
- 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
- Amazon Q Developer 現已正式推出，包含可重新構想開發人員體驗的新功能預覽
- 在您的 中使用 Amazon Q 開發人員的終極備忘單 IDE
- 左移工作負載，利用 AI 建立測試
- Amazon Q 開發人員中心
- 使用 Amazon 更快建置應用程式的 10 種方法 CodeWhisperer
- 使用 Amazon 超越程式碼涵蓋範圍 CodeWhisperer
- Amazon Prompt Engineering 的最佳實務 CodeWhisperer
- 敏捷式軟體指南
- CI/CD 管道是我的發行主管
- 使用 Amazon CodeGuru Reviewer 自動化程式碼檢閱
- 採用測試驅動的開發方法
- 如何使用 Amazon DevFactory 建置更好的應用程式 CodeGuru
- 關於配對程式設計
- RENGA Inc. 使用 Amazon 自動化程式碼檢閱 CodeGuru
- 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：

- API使用 Amazon Q Developer Agent for Software Development 實作
- 搭配 JetBrains 安裝、設定和使用 Amazon Q 開發人員 IDEs (方法)
- 掌握 Amazon CodeWhisperer YouTube 播放清單的藝術
- AWS re：Invent 2020：使用 Amazon 持續改善程式碼品質 CodeGuru
- AWS ANZ Summit 2021 - 使用 CDK和 測試驅動開發推動測試優先策略

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。在環境中測試並推廣程式碼，以逐步進行生產。

常見的反模式：

- 您在共用開發環境中進行開發，而另一名開發人員覆寫了您的程式碼變更。
- 對共用開發環境的限制性安全控制可防止您試驗新服務和功能。
- 您對生產系統執行負載測試，並造成使用者停機。
- 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，並防止它再次發生。為防止更多資料在測試期間遺失，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：

- 上的執行個體排程器 AWS
- 什麼是 AWS CloudFormation？

OPS05-BP09 進行頻繁、小型、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：頻繁部署小型變更，可加快開發工作的速度。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：

- 在 上實作 Microservices AWS
- 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。

使用資源標籤和 AWS Resource Groups 並遵循一致的標記策略，來套用中繼資料，以幫助識別您的資源。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。開發人員不需要登入 AWS Management Console 即可提供更新。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。
- 身為開發人員，您登入 AWS Management Console ，使用非標準方法和系統建立新的開發環境。

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 - AWS Well-Architected 的最佳實務 DevOps AWS

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 計畫變更失敗
- OPS06-BP02 測試部署
- OPS06-BP03 採用安全部署策略
- OPS06-BP04 自動化測試和復原

OPS06-BP01 計畫變更失敗

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詳細的恢復計畫。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。
- 您未使用基礎設施即程式碼 (IaC)，並且以手動方式更新了基礎設施，從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。

建立此最佳實務的好處：擁有從失敗變更中復原的計畫，可最大限度地減少復原的平均時間（MTTR），並減少您的業務影響。

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畫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 | 確保部署期間的復原安全
- AWS 白皮書 | 雲端的變更管理

相關影片：

- re:Invent 2019 | Amazon 的高可用性部署方法

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型開發文化，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

作為持續整合和持續交付 （CI/CD） 管道的一部分， AnyCompany Retail 會執行在類似生產環境中為其客戶發佈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. 使用偏離偵測來偵測 資源是否已在 之外變更 AWS CloudFormation。
   b. 使用變更集來驗證堆疊更新的意圖是否符合啟動變更集時 AWS CloudFormation 採取的動作。
2. 這會觸發 AWS CodePipeline 中的手動核准步驟，以授權生產前環境的部署。
3. 使用AWS CodeDeploy AppSpec檔案等部署組態來定義部署和驗證步驟。
4. 在適用的情況下，AWS CodeDeploy 與其他 AWS 服務整合，或與AWS CodeDeploy 合作夥伴產品和服務整合。
5. 使用 Amazon CloudWatch AWS CloudTrail、 和 Amazon SNS事件通知來監控部署。
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. 對部署問題進行故障診斷。
8. 成功驗證前述步驟後，應該啟動手動核准工作流程，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 測試和驗證變更

相關文件：

- AWS Builders' Library | 自動化安全、移交部署 | 測試部署
- AWS 白皮書 | 在 上練習持續整合和持續交付 AWS
- 阿波羅的故事 - Amazon 的部署引擎
- 如何在運送程式碼之前在 AWS CodeDeploy 本機進行測試和偵錯
- 整合網路連線測試與基礎設施部署

相關影片：

- re:Invent 2020 |在 Amazon 測試軟體和系統

相關範例：

- 教學課程 | 透過驗證測試部署 和 Amazon ECS服務

OPS06-BP03 採用安全部署策略

安全的生產部署可控制有益變更的流程，目的是將這些變更對客戶造成的任何影響降到最低。安全控制提供檢查機制，以驗證所需的結果，並限制變更引入的任何缺陷或部署失敗造成的影響範圍。安全推出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持續交付 (CI/CD) 系統，可提供自動化安全部署的功能。團隊必須使用適當的安全推出策略。

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，所有客戶都會同時受到影響。
- 同時部署到所有系統中引入的缺陷需要緊急釋放。為所有客戶修正它需要數天的時間。
- 管理生產發行需要多個團隊的規劃和參與。這會限制您為客戶經常更新功能的能力。
- 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：自動化部署在推出速度和持續為客戶提供有益變更之間取得了平衡。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。

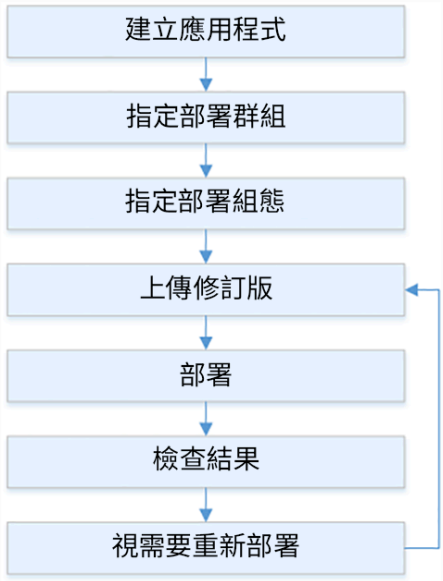
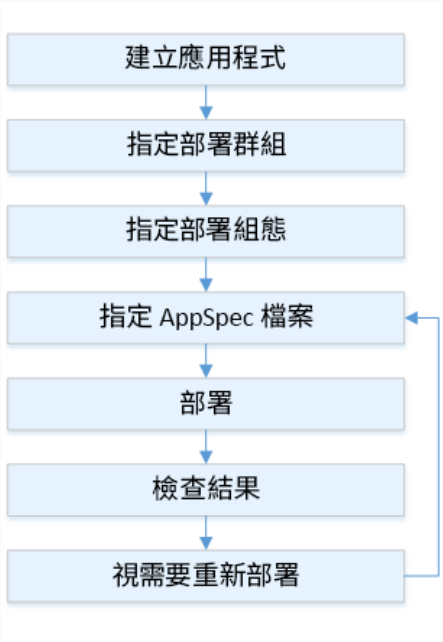
未建立此最佳實務時的曝險等級：中

實作指引

持續交付失敗可能導致服務可用性降低和糟糕的客戶體驗。為了最大限度地提高成功部署的速率，請在end-to-end發行程序中實作安全控制，以將部署錯誤降至最低，目標是實現零部署失敗。

客戶範例

AnyCompany Retail 的任務是實現最低到零的停機時間部署，這表示部署期間不會對使用者造成可察覺的影響。為此，公司已建立了部署模式 (請參閱下列工作流程圖表)，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。

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

實作步驟

1. 推廣到生產之後，使用核准工作流程可啟動生產推出步驟的一系列動作。
2. 使用自動化部署系統，例如 AWS CodeDeploy。 AWS CodeDeploy 部署選項包括 EC2/內部部署的就地部署，以及 EC2/內部部署的藍/綠部署 AWS Lambda，以及 Amazon ECS（請參閱先前的工作流程圖表）。
   a. 在適用的情況下，AWS CodeDeploy 與其他 AWS 服務整合，或與AWS CodeDeploy 合作夥伴產品和服務整合。
3. 針對 Amazon Aurora 和 Amazon RDS等資料庫使用藍/綠部署。
4. 使用 Amazon CloudWatch AWS CloudTrail、 和 Amazon Simple Notification Service（AmazonSNS） 事件通知來監控部署。
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. 對部署問題進行故障診斷。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS05-BP02 測試和驗證變更
- OPS05-BP09 進行頻繁、小型、可逆的變更
- OPS05-BP10 完全自動化整合和部署

相關文件：

- AWS Builders Library | 自動化安全的實作部署 | 生產部署
- AWS Builders Library | 我的 CI/CD 管道是我的版本負責人 | 安全的自動生產版本
- AWS 白皮書 | 實作持續整合和持續交付 AWS | 部署方法
- AWS CodeDeploy 使用者指南
- 在 中使用部署組態 AWS CodeDeploy
- 設定 API Gateway Canary 版本部署
- Amazon ECS 部署類型
- Amazon Aurora 和 Amazon 中完全受管的藍/綠部署 RDS
- 藍/綠部署搭配 AWS Elastic Beanstalk

相關影片：

- re:Invent 2020 | 無人為介入：Amazon 的自動化持續交付管道
- re:Invent 2019 | Amazon 的高可用性部署方法

相關範例：

- 在 中嘗試藍色/綠色部署範例 AWS CodeDeploy
- Worlshop | 適用於 Lambda Canary 部署的 Buiding CI/CD 管道，使用 AWS CDK
- 研討會 | EKS適用於 和 的藍/綠和 Canary 部署 ECS
- 研討會 | 建立跨帳戶 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，這些功能會減少 time-to-delivery，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. Canary 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式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 計畫變更失敗
- OPS06-BP02 測試部署

相關文件：

- AWS Builders Library | 確保部署期間的復原安全
- 使用 重新部署和復原部署 AWS CodeDeploy
- 使用 自動化部署時的 8 個最佳實務 AWS CloudFormation

相關範例：

- 使用 Selenium AWS LambdaAWS 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：Amazon 的自動化持續交付管道
- re:Invent 2019 | Amazon 的高可用性部署方法

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 確保對操作就緒狀態進行一致審核
- OPS07-BP03 使用 Runbook 執行程序

- OPS07-BP04 使用教戰手冊調查問題
- OPS07-BP05 做出明智的決策以部署系統和變更
- OPS07-BP06 建立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有助於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曝險等級：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 零售部門會確保支援工作負載的團隊有適當的人員配置和訓練。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 訓練和認證具有有關 的課程庫 AWS。它們提供免費和付費的線上或面授課程。
   b. AWS 託管您向專家學習的事件和網路研討會。 AWS
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 事件和網路研討會
- AWS 訓練和認證

OPS07-BP02 確保對操作就緒狀態進行一致審核

使用操作整備檢閱 （ ORRs ） 來驗證您可以操作工作負載。ORR 是在 Amazon 開發的一種機制，用於驗證團隊是否可以安全地操作工作負載。ORR 是使用 需求清單的檢閱和檢查程序。ORR 是團隊用來驗證工作負載的自助式體驗。ORRs 包含從我們多年建置軟體中學到的最佳實務。

ORR 檢查清單由架構建議、操作程序、事件管理和發行品質組成。錯誤糾正 (CoE) 程序是這些項目的主要驅動要素。您自己的事後分析應該會推動您自己的 演變ORR。ORR 不僅要遵循最佳實務，還要防止您先前看到的事件重複發生。最後，安全、管理和合規要求也可以包含在 中ORR。

在工作負載啟動到一般可用性ORRs之前執行，然後在整個軟體開發生命週期中執行。在啟動ORR之前執行 可提高您安全操作工作負載的能力。定期在工作負載ORR上重新執行您的 ，以捕捉最佳實務中的任何偏離。您可以擁有新服務啟動和定期檢閱ORRs的ORR檢查清單。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。當您使用雲端成熟時，您可以將ORR需求建置到架構中做為預設值。

預期結果：您有一個ORR檢查清單，其中包含組織的最佳實務。ORRs 在工作負載啟動之前執行。ORRs 會在工作負載生命週期期間定期執行。

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。
- 所學課程會納入您的ORR程序中。
- 工作負載啟動時，已設置必要的程序。
- ORRs 會在工作負載的軟體生命週期中執行。

若未建立此最佳實務的風險等級：高

實作指引

ORR 是兩件事：程序和檢查清單。您的ORR程序應由您的組織採用，並由執行發起人支援。工作負載啟動到一般可用性之前，至少ORRs必須執行 。ORR 在整個軟體開發生命週期中執行 ，以保持其符合最佳實務或新要求的最新狀態。ORR 檢查清單應包含組態項目、安全和治理要求，以及組織的最佳實務。隨著時間的推移，您可以使用 AWS Config、 AWS Security Hub和 AWS Control Tower Guardrails 等 服務，將最佳實務從 建置ORR到 Guardrails，以自動偵測最佳實務。

客戶範例

發生多次生產事件後， 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。新的工作負載會在啟動ORRs之前執行。每個工作負載每年都會執行一次ORR最佳實務子集，以整合新增至ORR檢查清單的新最佳實務和要求。隨著時間的推移， AnyCompany Retail AWS Config 用於偵測一些最佳實務，加速ORR程序。

實作步驟

若要進一步了解 ORRs，請閱讀操作就緒審核（ORR）白皮書。它提供有關ORR程序歷史記錄、如何建立您自己的ORR實務以及如何開發ORR檢查清單的詳細資訊。以下步驟是該文件的精簡版本。若要深入了解什麼ORRs是 以及如何建置自己的 ，我們建議您閱讀該白皮書。

1. 召集關鍵利益相關者，包含安全性、營運和開發等團隊的代表人員。
2. 請每位利益相關者提供至少一個需求。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：Operational Readiness Reviews（ ）白皮書中的範例ORR問題包含可用來開始使用的範例問題。ORR
3. 將需求集中放在試算表中。
   - 您可以在 中使用自訂鏡頭AWS Well-Architected Tool來開發您的 ORR，並在您的帳戶和 AWS 組織之間共用。
4. 識別一個要在 ORR 上執行的工作負載。啟動前的工作負載或內部工作負載是理想的選擇。
5. 執行ORR檢查清單，並記下所做的任何發現。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. 隨著時間的推移，繼續將最佳實務和要求新增至ORR檢查清單。

AWS Support 擁有企業支援的客戶可以向技術客戶經理請求操作就緒審核研討會。研討會是互動式向後工作工作階段，用於開發您自己的ORR檢查清單。

實作計劃的工作量：高。在組織中採用ORR實務需要高階主管贊助和利益相關者認同。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 評估治理要求 – 治理要求自然適合ORR檢查清單。
- OPS01-BP04 評估合規要求 – 合規要求有時包含在ORR檢查清單中。有些時候，它們會是獨立的程序。
- OPS03-BP07 資源團隊適當 – 團隊能力是ORR滿足需求的良好候選者。
- OPS06-BP01 計畫變更失敗 – 啟動工作負載前，必須先建立回復或向前回復計畫。
- 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 – 安全控制目標具有卓越的ORR要求。
- REL13-BP01 定義停機時間和資料遺失的復原目標 – 災難復原計劃是很好ORR的要求。

- COST02-BP01 根據您的組織需求制定政策 – 成本管理政策非常適合包含在ORR檢查清單中。

相關文件：

- AWS Control Tower - 中的護欄 AWS Control Tower
- AWS Well-Architected Tool - 自訂鏡頭
- Adrian Hornsby 提供的營運準備度審查範本
- 操作就緒審核 （ ORR ） 白皮書

相關影片：

- AWS Support s You | 建立有效的營運就緒狀態檢閱 （ ORR ）

相關範例：

- 範例操作準備檢閱 （ ORR ） 鏡頭

相關服務：

- AWS Config
- AWS Control Tower
- AWS Security Hub
- AWS Well-Architected Tool

OPS07-BP03 使用 Runbook 執行程序

執行手冊是為了達成特定成果而記錄的程序。執行手冊由一系列可供遵循以完成某項工作的步驟組成。早在航空業早期，就已使用執行手冊。在雲端操作中，我們使用執行手冊來降低風險及達到預期成果。簡言之，執行手冊就是完成一項工作的檢查清單。

執行手冊是操作工作負載的重要組成部分。從新團隊成員入職到部署重大版本，執行手冊是經過編纂的流程，無論誰使用這些執行手冊，都能提供一致的成果。應該在中央位置發布執行手冊，並隨著流程的發展進行更新，因為更新執行手冊是變更管理流程的關鍵組成部分。它們還應該包括當發生問題時有關錯誤處理、工具、權限、異常以及向上呈報的指引。

隨著組織的成熟，開始自動化執行手冊。從簡短且經常使用的執行手冊開始。使用指令碼語言來自動化步驟或讓步驟更容易執行。當您自動化前幾個執行手冊時，將花費時間來自動化更複雜的執行手冊。隨著時間的推移，大多數執行手冊都應該以某種方式自動化。

預期結果：您的團隊有一系列 step-by-step執行工作負載任務的指南。執行手冊中包含預期成果、必要的工具和許可，以及錯誤處理指示。它們會集中存放 (版本控制系統)，並且經常更新。例如，您的 Runbook 為您的團隊提供在應用程式警示、操作問題和規劃的生命週期 AWS Health 事件期間監控、通訊和回應重要帳戶事件的功能。

常見的反模式：

- 依靠記憶體來完成流程的每個步驟。
- 手動部署變更，無需檢查清單。
- 不同的團隊成員執行相同的過程，但具有不同的步驟或成果。
- 讓執行手冊脫離系統變更和自動化。

建立此最佳實務的優勢：

- 降低手動任務的錯誤率。
- 以一致的方式執行操作。
- 新的團隊成員可以更快地開始執行任務。
- 可以自動化執行手冊以減少辛勞。

未建立此最佳實務時的曝險等級：中

實作指引

執行手冊可以根據組織的成熟度等級採用多種形式。至少應該包含 step-by-step文字文件。應明確指出預期成果。清楚記錄必要的特殊權限或工具。如果發生問題，提供有關錯誤處理和呈報的詳細指引。列出執行手冊擁有者並將其發布在中央位置。執行手冊被記錄下來之後，透過讓團隊中的其他人執行它來進行驗證。隨著程序的發展，請根據您的變更管理流程來更新執行手冊。

隨著組織的成熟，應自動化文字執行手冊。使用 AWS Systems Manager Automation 等服務，可以將純文字轉換為可針對工作負載執行的自動化功能。這些自動化可以用來回應事件，減少維護工作負載的操作負擔。 AWS Systems Manager Automation 還提供低程式碼視覺化設計體驗，以更輕鬆地建立自動化 Runbook。

客戶範例

AnyCompany 零售必須在軟體部署期間執行資料庫結構描述更新。雲端操作團隊與資料庫管理團隊合作，建立用於手動部署這些變更的執行手冊。執行手冊以檢查清單的形式列出流程中的每個步驟。它包括發生問題時關於錯誤處理的部分。他們在其內部 wiki 中發布該執行手冊以及其他執行手冊。雲端操作團隊計劃在未來的衝刺中自動化該執行手冊。

實作步驟

如果您沒有現有的文件儲存庫，版本控制儲存庫是開始建置執行手冊庫的好地方。可以使用 Markdown 來構建執行手冊。我們提供了一個執行手冊範本範例，您可以使用它來開始構建執行手冊。

```
# 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流程。理想的流程是半定期執行，步驟數量少，並且具有低影響故障。
3. 在文件儲存庫中，使用範本建立新的 Markdown 草稿文件。填寫執行手冊標題和執行手冊資訊下的必填欄位。
4. 從第一個步驟開始，填寫執行手冊的「步驟」部分。
5. 將執行手冊交給團隊成員。讓他們使用執行手冊來驗證步驟。如果缺少某些內容或需要澄清，請更新執行手冊。
6. 將執行手冊發布到您的內部文件存放區。發布後，告知您的團隊和其他利益相關者。
7. 隨著時間的推移，您將建置執行手冊的程式庫。隨著程式庫的增長，開始努力自動化執行手冊。

實作計劃的工作量：低。Runbook 的最低標準是 step-by-step文字指南。自動化執行手冊可以增加實作工作量。

資源

相關的最佳實務：

- OPS02-BP02 程序已識別擁有者
- OPS07-BP04 使用教戰手冊調查問題
- OPS10-BP01 使用事件、事件和問題管理的程序
- OPS10-BP02 每個提醒都有一個程序
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected Framework：概念：執行手冊開發
- 使用自動化程序手冊和執行手冊實現卓越的營運
- AWS Systems Manager：使用 Runbook
- AWS 大型遷移的遷移手冊 - 任務 4：改善遷移執行手冊
- 使用 AWS Systems Manager Automation 執行手冊來解決操作任務

相關影片：

- AWS re：Invent 2019： Runbook、事件報告和事件回應DIY指南
- 如何在 上自動化 IT 操作 AWS | Amazon Web Services
- 將指令碼整合到 AWS Systems Manager

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：建立卓越營運雲端自動化實務：最佳實務 AWS Managed Services
- AWS Systems Manager：自動化演練
- AWS Systems Manager：從最新的快照 Runbook 還原根磁碟區
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事件回應 Runbook
- Gitlab - 執行手冊
- Rubix - 用於在 Jupyter 筆記本中構建執行手冊的 Python 庫
- 使用文件建置器建立自訂執行手冊

相關服務：

- AWS Systems Manager 自動化

OPS07-BP04 使用教戰手冊調查問題

Playbook 是 step-by-step用來調查事件的指南。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。程序手冊適用於各種情況，從失敗的部署到安全性事故。在許多案例中，程序手冊可釐清根本原因，而執行手冊則用來緩解該根本原因。程序手冊是組織事件應變計劃的關鍵要素。

一個好的程序手冊有幾個關鍵功能。它循序漸進地引導使用者完成探索過程。從外到內思考，應該遵循哪些步驟來診斷事件？ 在程序手冊中明確定義程序手冊中是否需要特殊工具或更高權限。制定溝通計畫，向利益相關者通報調查進展情況，這非常關鍵。在無法確定根本原因的情況下，程序手冊應具有升級計畫。如果確定了根本原因，程序手冊應該指向說明如何解決問題的執行手冊。程序手冊應集中存放並定期維護。如果程序手冊用於特定提醒，請在提醒中為您的團隊提供指向程序手冊的指引。

隨著組織的成熟，會將您的程序手冊自動化。從涵蓋低風險事件的程序手冊開始。使用指令碼自動執行探索步驟。確認您有配套的執行手冊來減輕常見根本原因。

預期成果：您的組織擁有常見事件的程序手冊。程序手冊存放在中心位置，並可供您的團隊成員使用。程序手冊會經常更新。對於任何已知的根本原因，都會構建配套的執行手冊。

常見的反模式：

- 調查事件沒有標準方法。
- 團隊成員依賴肌肉記憶或機構知識來疑難排解失敗的部署。
- 新團隊成員學習如何透過試驗和錯誤來調查問題。
- 調查問題的最佳實務不會跨團隊共用。

建立此最佳實務的優勢：

- 程序手冊可加強您減輕事故的努力。
- 不同的團隊成員可以使用相同的程序手冊，以一致的方式識別根本原因。
- 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。
- 程序手冊有助於團隊成員更快地開始做出貢獻。
- 團隊可以透過可重複的程序手冊擴展其程序。

未建立此最佳實務時的曝險等級：中

實作指引

建置和使用程序手冊的方式取決於組織的成熟度。如果您是雲端新手，請在中央文件儲存庫中以文字形式建立程序手冊。隨著組織的成熟，程序手冊可以使用 Python 之類的指令碼語言進行半自動化。這些指令碼可以在 Jupyter 筆記本內部運行，以加快發現速度。進階組織具有完全自動化的程序手冊，可解決使用執行手冊自動修復的常見問題。

列出工作負載發生的常見事件，開始建置程序手冊。為低風險並且根本原因已縮小到幾個問題的事件選擇程序手冊以開始。在您擁有更簡單案例的程序手冊之後，請轉到風險較高的案例或根本原因尚不明確的案例。

隨著組織的成熟，應自動化文字程序手冊。使用 AWS Systems Manager Automation 等服務，可以將純文字轉換為自動化功能。可以針對您的工作負載執行這些自動化，以加快調查速度。可以啟動這些自動化以回應事件，減少發現和解決事故的平均時間。

客戶可以使用 AWS Systems Manager Incident Manager 來回應事故。此服務提供單一介面來分類事故、在發現和緩解期間通知利益相關者，並在整個事故中進行協同合作。它使用 AWS Systems Manager Automations 來加速偵測和復原。

客戶範例

生產事件影響 AnyCompany 了零售。隨時待命的工程師使用程序手冊來調查問題。隨著他們逐步完成這些步驟，他們會讓程序手冊中確定的關鍵利益相關者了解最新狀況。工程師將根本原因確定為後端服務中的競爭條件。工程師使用 Runbook 重新啟動服務，讓 AnyCompany零售重新上線。

實作步驟

如果您沒有現有的文件儲存庫，建議您為程序手冊庫建立版本控制儲存庫。您可以使用 Markdown 構建程序手冊，它可與大多數程序手冊自動化系統相容。如果您是從頭開始，請使用下列範例程序手冊範本。

```
# 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您的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。這種情況應該是根本原因僅限於少數問題且解決方案風險較低。
3. 使用 Markdown 範本，填寫 [程序手冊名稱] 部分和 [程序手冊資訊] 下方的欄位。
4. 填寫疑難排解步驟。盡可能明確要執行哪些操作或應該調查哪些領域。
5. 將程序手冊交給團隊成員，讓他們仔細閱讀以驗證。如果有任何遺漏或不清楚的內容，請更新程序手冊。
6. 在文件儲存庫中發布程序手冊，並通知您的團隊和任何利益相關者。
7. 此程序手冊庫會隨著您新增更多程序手冊而增加。擁有多個教戰手冊後，開始使用 AWS Systems Manager Automations 等工具來自動化這些手冊，以保持自動化和教戰手冊的同步。

實作計劃的工作量：低。您的程序手冊應該是存放在中央位置的文字文件。更成熟的組織將推進程序手冊自動化。

資源

相關的最佳實務：

- OPS02-BP02 程序已識別擁有者
- OPS07-BP03 使用 Runbook 執行程序
- OPS10-BP01 使用事件、事件和問題管理的程序
- OPS10-BP02 每個警示都有一個程序
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected Framework：概念：程序手冊開發
- 使用自動化程序手冊和執行手冊實現卓越的營運
- AWS Systems Manager：使用 Runbook
- 使用 AWS Systems Manager Automation 執行手冊來解決操作任務

相關影片：

- AWS re：Invent 2019： Runbooks、事件報告和事件回應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

相關範例：

- AWS 客戶程序手冊架構
- AWS Systems Manager：自動化演練
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事件回應 Runbook
- Rubix - 用於在 Jupyter 筆記本中構建執行手冊的 Python 庫
- 使用文件建置器建立自訂執行手冊
- 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：Jupyter 的事件回應手冊

相關服務：

- AWS Systems Manager 自動化
- AWS Systems Manager Incident Manager

OPS07-BP05 做出明智的決策以部署系統和變更

為成功和失敗的工作負載變更建立程序。事前剖析是一種演練，團隊可藉此模擬失敗，制定緩解策略。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曝險等級：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 零售會定期執行預審，以驗證其程序是否有變更不成功。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。開發會啟動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。

實作計畫的工作量：中。實作事前剖析的實務需要貴組織利益相關者的協調和努力

資源

相關的最佳實務：

- OPS01-BP03 評估治理要求 - 管控要求是判斷是否部署變更的關鍵因素。
- OPS06-BP01 計畫變更失敗 - 建立計畫來緩解失敗的部署並且使用事前剖析來進行驗證。
- 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 共同責任模型
- 中的治理 AWS 雲端：敏捷性與安全之間的正確平衡

OPS07-BP06 建立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。這些相依性的支援計劃的存在有其必要性，以便應對服務中斷或軟體問題。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畫。
- 根據服務層級需求選擇適當的支援計畫。
- 記錄支援計畫、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畫。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曝險等級：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畫。設定適當的支援計畫以滿足服務層級需求。對於 AWS 客戶，這表示在您擁有生產工作負載的任何帳戶上啟用 AWS Business Support 或更高版本。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 零售，所有商業軟體和服務相依性都有支援計畫。例如，它們已在具有生產工作負載的所有 帳戶上啟用 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. 與組織中的利益相關者合作，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畫。
3. 針對商業軟體和服務，與供應商共同建立支援計畫。
    a. 為所有生產帳戶訂閱 AWS Business Support 或更高版本，可提供更快的回應時間 AWS Support ，而且強烈建議您這麼做。如果您沒有進階支援，您必須有行動計畫來處理問題，這需要 的協助 AWS Support。 AWS Support 提供工具和技術、人員和計畫的組合，旨在主動協助您最佳化效能、降低成本並更快速地創新。 AWS 業務支援提供其他好處，包括存取 AWS Trusted Advisor 和 AWS Personal Health Dashboard，以及更快的回應時間。
4. 在您的知識管理工具中記錄支援計畫。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畫。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 程序已識別擁有者

相關文件：

- AWS Support 計劃

相關服務：

- AWS 業務支援
- AWS 企業支援

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 建立可操作的警示
- 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 Amazon 等工具 CloudWatch 執行指標分析。 CloudWatch 異常偵測和 Amazon DevOpsGuru 等 AWS 服務可用來偵測異常，特別是靜態閾值未知或行為模式更適合異常偵測時。

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. 使用 Amazon CloudWatch：使用 Amazon CloudWatch 進行集中式檢視和深入分析。
   a. 設定 CloudWatch 儀表板以視覺化您的指標，並隨時間進行比較。
   b. 使用 中的百分位數 CloudWatch來取得指標分佈的清晰檢視，這有助於定義SLAs和了解異常值。
   c. 設定CloudWatch 異常偵測以識別異常模式，而不必依賴靜態閾值。
   d. 實作CloudWatch 跨帳戶可觀測性，以監控和疑難排解跨區域內多個帳戶的應用程式。
   e. 使用 CloudWatch Metric Insights 查詢和分析帳戶和區域的指標資料，識別趨勢和異常。
   f. 套用CloudWatch 指標數學來轉換、彙總或執行指標的計算，以取得更深入的洞見。
3. 使用 Amazon DevOpsGuru：將 Amazon DevOpsGuru 納入其機器學習增強型異常偵測，以識別無伺服器應用程式的早期操作問題跡象，並在影響客戶之前對其進行修復。
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 中啟用跨帳戶可觀測性 CloudWatch
- Amazon DevOpsGuru 簡介
- 使用 持續分析指標 AWS Cost Anomaly Detection

相關範例：

- 一個可觀測性研討會
- AIOps使用 Amazon DevOpsGuru 取得操作洞見

OPS08-BP02 分析工作負載日誌

定期分析工作負載日誌對於深入了解應用程式的操作層面至關重要。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。

預期成果：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。
- 沒有使用可用於日誌分析的完整工具套件，錯過了關鍵洞見。
- 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，以實現持續的應用程式最佳化。
- 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曝險等級：中

實作指引

Amazon CloudWatch Logs 是日誌分析的強大工具。 CloudWatch Logs Insights 和 Contributor Insights 等整合功能，讓從日誌中擷取有意義的資訊的過程變得直覺且有效。

實作步驟

1. 設定 CloudWatch 日誌 ：設定應用程式和服務將日誌傳送至 CloudWatch 日誌。
2. 使用日誌異常偵測：利用 Amazon CloudWatch Logs 異常偵測自動識別並提醒異常日誌模式。此工具可協助您主動管理日誌中的異常，並儘早偵測潛在問題。
3. 設定 CloudWatch Logs Insights ：使用 CloudWatch Logs Insights 以互動方式搜尋和分析您的日誌資料。
   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常用日誌模式。此功能可協助您了解日誌資料中常見的操作趨勢和潛在的異常值。
   c. 使用 CloudWatch Logs compare （ diff ） 在不同時段或不同日誌群組之間執行差異分析。使用此功能可精確找出變更，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 即時檢視日誌資料。您可以在應用程式的操作活動發生時進行主動監控，以便立即掌握系統效能和潛在問題。
5. 利用 Contributor Insights ：使用 CloudWatch Contributor Insights 來識別 IP 地址或使用者代理等高基度維度的熱門發言者。
6. 實作 CloudWatch 日誌指標篩選條件 ：設定CloudWatch 日誌指標篩選條件，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 ：監控和疑難排解跨區域內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
- OPS08-BP01 分析工作負載指標

相關文件：

- 使用 Logs Insights 分析 CloudWatch 日誌資料
- 使用 CloudWatch 貢獻者洞察
- 建立和管理 CloudWatch 日誌指標篩選條件

相關影片：

- 使用 Logs Insights 分析 CloudWatch 日誌資料
- 使用 CloudWatch 貢獻者洞察分析高基數資料

相關範例：

- CloudWatch 記錄範例查詢
- 一個可觀測性研討會

OPS08-BP03 分析工作負載追蹤

分析追蹤資料對於實現應用程式營運歷程的全面檢視至關重要。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。

預期成果：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。
- 不會將追蹤資料與相關日誌建立關聯。
- 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：

- 改善故障診斷並減少解決的平均時間（MTTR）。

- 深入了解依賴性及其影響。
- 快速識別和糾正效能問題。
- 利用追蹤衍生的指標制定明智的決策。
- 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曝險等級：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。 ServiceLens、X-Ray Insights、X-Ray Analytics 和 Amazon DevOpsGuru 等功能可增強從追蹤資料衍生的可操作洞察深度。

實作步驟

下列步驟提供結構化方法，可有效使用 AWS 服務實作追蹤資料分析：

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：深入研究 X-Ray 追蹤衍生的指標，例如延遲、請求率、錯誤率和回應時間分佈，使用服務地圖來監控應用程式運作狀態。
3. 使用 ServiceLens：利用ServiceLens地圖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，以自動偵測追蹤中的異常。
   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 Insights 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、精確定位模式並擷取洞見。
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. 合併 Amazon DevOpsGuru ：讓 Amazon DevOpsGuru 受益於機器學習模型，以找出追蹤中的操作異常。
8. 使用 CloudWatch Synthetics ：使用 CloudWatch Synthetics 建立 Canary，以持續監控您的端點和工作流程。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. 使用實際使用者監控（RUM）：使用 AWS X-Ray 和 CloudWatch RUM，您可以從應用程式的最終使用者開始透過下游 AWS 受管服務分析和偵錯請求路徑。這樣做有助於找出影響最終使用者的延遲趨勢和錯誤。
10. 與日誌建立關聯：將追蹤資料與 X-Ray 追蹤檢視中的相關日誌建立關聯，以深入了解應用程式行為。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：監控和疑難排解跨區域內多個帳戶的應用程式。

實作計劃的工作量：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態
- 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 & 分析和偵錯應用程式 AWS X-Ray
- 使用 AWS X-Ray Insights

相關範例：

- 一個可觀測性研討會
- 使用 實作 X-Ray AWS Lambda
- CloudWatch Synthetics Canary 範本

OPS08-BP04 建立可操作的警示

及時偵測並回應您的應用程式行為中的偏差至關重要。尤其重要的是根據關鍵績效指標 （KPIs） 來識別結果何時處於風險狀態，或何時出現非預期異常。開啟警示KPIs可確保您收到的訊號直接與業務或營運影響相關聯。這種可採取動作的提醒方法可促進主動回應，並有助於維持系統效能與可靠性。

預期結果：接收及時、相關且可採取行動的提醒，以快速識別和緩解潛在問題，尤其是當KPI結果處於風險狀態時。

常見的反模式：

- 設定太多非嚴重性提醒會導致提醒疲勞。
- 不根據 排定警示的優先順序KPIs，因此很難了解問題的業務影響。
- 忽視解決根本原因導致同一問題的重複提醒。

建立此最佳實務的優勢：

- 透過專注於可操作且相關的提醒來減少提醒疲勞。
- 透過主動偵測和緩解問題，改善系統運作時間和可靠性。
- 透過與熱門的提醒和通訊工具整合，強化團隊協同作業並加快解決問題的速度。

未建立此最佳實務時的曝險等級：高

實作指引

若要建立有效的警示機制，請務必使用指標、日誌和追蹤資料，在偵測到基於 的結果KPIs處於風險或異常時標記。

實作步驟

1. 判斷關鍵效能指標 （KPIs）：識別應用程式的 KPIs。警示應與這些警示相關聯KPIs，以準確反映業務影響。
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異常模式，這可協助您僅產生真實異常的警示。
   - 使用 AWS X-Ray Insights：
     a. 設定 X-Ray Insights 以偵測追蹤資料中的異常。

   b. 設定 X-Ray Insights 的通知，以便在偵測到問題時收到提醒。

- 與 Amazon DevOpsGuru 整合：
   a. 利用 Amazon DevOpsGuru 的機器學習功能來偵測現有資料的操作異常。
   b. 導覽至 DevOpsGuru 中的通知設定，以設定異常警示。

3. 實作可執行的提醒：設計提醒，為立即採取行動提供足夠資訊。
   1. AWS Health 使用 Amazon EventBridge 規則 監控事件，或以程式設計方式與 整合， AWS Health API以便在接收 AWS Health 事件時自動執行動作。這些動作可以是一般動作 (例如將所有規劃的生命週期事件訊息傳送至聊天介面) 或是特定動作 (例如在 IT 服務管理工具中啟動工作流程)。
4. 減少提醒疲勞：將非嚴重性提醒降至最低。當團隊對眾多微不足道的提醒感到不知所措時，他們可能會失去對重大問題的監督，從而降低提醒機制的整體有效性。
5. 設定複合警示 ：使用 Amazon CloudWatch 複合警示來合併多個警示。
6. 與警示工具整合 ：整合 Ops Genie 和 等工具PagerDuty。
7. 使用 AWS Chatbot：整合AWS Chatbot以將警示轉送至 Amazon Chime、Microsoft Teams 和 Slack。
8. 基於日誌的警示：使用 中的日誌指標篩選條件 CloudWatch ，根據特定日誌事件建立警示。
9. 審查並反覆：定期重新檢視並調整提醒組態。

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
- 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
- DevOpsGuru 通知
- X-ray Insights 通知
- 透過互動式監控、操作和疑難排解您的 AWS 資源 ChatOps
- Amazon CloudWatch 整合指南 | PagerDuty
- 將 Opsgenie 與 Amazon 整合 CloudWatch

相關影片：

- 在 Amazon 中建立複合警示 CloudWatch
- AWS Chatbot 概觀
- AWS 在 Air ft. 中的互變命令 AWS Chatbot

相關範例：

- 使用 Amazon 的雲端警示、事件管理和修復 CloudWatch
- 教學課程：建立將通知傳送至 的 Amazon EventBridge 規則 AWS Chatbot
- 一個可觀測性研討會

OPS08-BP05 建立儀表板

儀表板是以人為本的工作負載遙測資料檢視。雖然它們提供了重要的視覺介面，但它們不應該取代警報機制，而是補充它們。經過精心打造的儀表板不僅能提供快速了解系統運作狀況和效能的洞見，還能對利益相關者呈現有關業務成果和問題影響層面的即時資訊。

預期成果：

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。

- 仰賴沒有異常偵測提醒的儀表板。
- 儀表板未隨著工作負載發展而更新。

建立此最佳實務的優勢：

- 立即查看關鍵系統指標 和 KPIs。
- 增強利益相關者的溝通和理解。
- 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

為企業量身打造的儀表板可KPIs吸引更廣泛的利益相關者。儘管這些人可能對系統指標不感興趣，但他們熱衷於了解這些數字的業務含義。以業務為中心的儀表板可確保所有受監控和分析的技術和營運指標都與總體業務目標保持同步。這種一致性提供了清晰度，確保每個人在什麼重要以及什麼不重要的問題上意見一致。此外，強調業務的儀表板KPIs往往更可行。利益相關者可以快速了解營運的運作狀態、需要注意的領域以及對業務成果的潛在影響。

考慮到這一點，在建立儀表板時，請確保技術指標與業務 之間存在平衡KPIs。兩者都至關重要，但兩者迎合不同的受眾。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 Dashboards 是 CloudWatch 主控台中的可自訂首頁，您可以使用它在單一檢視中監控您的資源，甚至是分散在不同 AWS 區域 和 帳戶的資源。

實作步驟

1. 建立基本儀表板：在 中建立新的儀表板 CloudWatch，為其提供描述性名稱。
2. 使用 Markdown 小工具:在深入研究指標之前，請使用 Markdown 小工具在儀表板頂端新增文字內容。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：在適當位置合併儀表板變數，以允許動態且靈活的儀表板檢視。
4. 建立儀表板小工具：新增儀表板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：使用 CloudWatch Log Insights 從您的日誌中衍生可執行的指標，並在儀表板上顯示這些洞察。
6. 設定警示：將CloudWatch警示整合到您的儀表板中，以快速檢視違反閾值的任何指標。
7. 使用貢獻者洞察：整合CloudWatch貢獻者洞察來分析高基數欄位，並更清楚地了解資源的主要貢獻者。
8. 設計自訂小工具：對於標準小工具未滿足的特定需求，請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：使用 AWS Health Dashboard可更深入地了解您的帳戶運作狀態、事件，以及可能影響服務和資源的近期變更。也可以在 AWS Organizations 中集中檢視運作狀態事件，或建立自己的自訂儀表板 (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 建立可操作的警示

相關文件：

- 建置用於檢視營運狀況的儀表板
- 使用 Amazon CloudWatch Dashboards

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 了解企業 (營運儀表板)

相關範例：

- 一個可觀測性研討會

- 使用 Amazon 進行應用程式監控 CloudWatch
- AWS Health Events Intelligence Dashboards 和 Insights
- 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 測量操作目標和 KPIs 指標
- OPS09-BP02 傳達狀態和趨勢，以確保操作的可見性
- OPS09-BP03 檢閱操作指標並排定改善優先順序

OPS09-BP01 測量操作目標和 KPIs 指標

從組織取得目標和 KPIs 來定義操作成功，並判斷指標是否反映這些目標。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：

- KPIs 組織營運團隊的目標 和 已發佈並共用。
- KPIs 會建立反映這些指標的指標。範例可能包括：
  - 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
  - 使用或不使用標準化操作程序所花費的時間 （SOP）
  - 從失敗的程式碼推送復原所花的時間長度
  - 通話音量

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益相關者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。使用這些方法，集思廣益可能反映這些操作目標的關鍵效能指標 （KPIs）。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。

使用 KPIs，識別最能反映這些目標的指標和資料來源。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 KPIs
- Amazon CloudWatch - 使用指標
- 建置儀表板
- 如何使用 KPIs KPI Dashboard 追蹤您的成本最佳化

OPS09-BP02 傳達狀態和趨勢，以確保操作的可見性

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，利益相關者和使用者社群就會收到警示。
- 組織領導階層和利益相關者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。
- 領導階層和其他利益相關者會收到報告，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，利益相關者就能迅速獲得資訊，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。

此外，儀表板和報告可以為決策者和利益相關者提供洞見，以了解營運團隊如何能夠回應業務需求以及如何分配其資源。這對於確定是否有足夠的資源來支援業務至關重要。

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指標和報告，這些指標和報告最能反映目標KPIs，以及他們在推動變革方面發揮影響力的位置。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 檢閱操作指標並排定改善優先順序

預留專用的時間和資源來檢閱操作狀態，可確保服務 day-to-day業務線仍是首要任務。召集營運主管和利益相關者定期檢閱指標、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。
- 利益相關者和業務領導者會定期收到有關營運狀態的簡報，並徵求有關目標KPIs、 和未來倡議的意見。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

花時間與利益相關者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。判斷KPIs這會影響哪些目標，以及哪些目標應該成功。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 使用事件、事件和問題管理的程序
- OPS10-BP02 每個提醒都有一個程序
- OPS10-BP03 根據業務影響排定操作事件的優先順序
- OPS10-BP04 定義升級路徑
- OPS10-BP05 定義影響服務事件的客戶通訊計劃

- OPS10-BP06 透過儀表板傳達狀態
- OPS10-BP07 自動化對事件的回應

OPS10-BP01 使用事件、事件和問題管理的程序

有效管理事件、事故和問題的能力是維持工作負載運作狀態和效能的關鍵。識別和理解這些元素之間的差異，以制定有效的回應和解決策略至關重要。為每個方面建立並遵循明確定義的流程，有助於您的團隊迅速且有效地處理出現的任何運營挑戰。

預期成果：您的組織透過詳細記錄且集中儲存的流程，有效地管理營運事件、事故和問題。這些流程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：

- 您會反應性地 (而非主動) 回應事件。
- 對不同類型的事件或事故採取不一致的方法。
- 您的組織不會分析事件並從中學習，以防止未來再次發生。

建立此最佳實務的優勢：

- 簡化且標準化的回應流程。
- 減少事件對服務和客戶的影響。
- 加速解決問題。
- 持續改善營運流程。

未建立此最佳實務時的曝險等級：高

實作指引

實作此最佳實務表示您正在追蹤工作負載事件。您有處理事件和問題的程序。會經常記錄、共用和更新這些程序。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：事件是對動作、狀況或狀態變化的觀察。事件可以經過計劃或未計劃，並且事情可以在工作負載內部或外部產生。
- 事故：事故是指需要回應的事件，例如意外中斷或服務品質下降。它們表示需要立即注意以恢復正常工作負載操作的中斷。

- 問題：問題是一個或多個事故的根本原因。識別和解決問題涉及更深入地研究事故，以防止將未來再次發生。

實作步驟

事件

1. 監控事件：
   - 實作可觀測性並利用工作負載可觀測性。
   - 使用者、角色 AWS 或服務採取的監控動作會在 中記錄為事件AWS CloudTrail。
   - 使用 Amazon EventBridge即時回應應用程式中的操作變更。
   - 使用 AWS Config 持續評估、監控和記錄資源組態變更。
2. 建立程序：
   - 制定一個程序來評估哪些事件重要並需要監控。這涉及設定正常和異常活動的閾值和參數。
   - 確定將事件升級為事故的條件。這可以基於嚴重性、對使用者的影響或與預期行為的偏差。
   - 定期檢閱事件監控和回應程序。這包括分析過去的事件、調整閾值以及完善警示機制。

事故

1. 回應事故：
   - 使用可觀測性工具的洞察力，快速識別並回應事故。
   - 實作 AWS Systems Manager Ops Center 以彙總、組織營運項目和事故，並排定優先順序。
   - 使用 Amazon CloudWatch 和 等服務AWS X-Ray進行更深入的分析和疑難排解。
   - 考慮 AWS Managed Services （AMS） 以增強事件管理，並利用其主動、預防性和偵測功能。AMS 透過監控、事件偵測和回應以及安全管理等服務擴展營運支援。
   - Enterprise Support 客戶可利用 AWS 事件偵測與回應功能，為生產工作負載提供持續的主動監控和事件管理。
2. 建立事件管理程序：
   - 建立結構化的事件管理流程，包括清晰的角色、通訊協定和解決步驟。
   - 將事件管理與諸如 AWS Chatbot 等工具整合，以實現有效率的回應和協調。
   - 依嚴重性將事件分類，並針對每個類別預先定義事件回應計畫。
3. 學習和改進：
   - 進行事件後分析以了解根本原因和解決方案有效性。

- 根據審查和不斷發展的實務，持續更新和改進回應計畫。
- 記錄並分享跨團隊所學到的經驗教訓，以增強營運彈性。
- Enterprise Support 客戶可向其技術客戶經理請求參加事件管理研討會。這個指導性研討會可測試您現有的事件回應計畫，並協助您找出需要改進的領域。

問題

1. 識別問題：
   - 使用先前事件的資料來識別可能指出更深層次系統性問題的週期性模式。
   - 利用 AWS CloudTrail和 Amazon CloudWatch 等工具來分析趨勢並發現潛在問題。
   - 與包括營運、開發和業務單位在內的跨職能團隊合作，以獲得有關根本原因的不同觀點。
2. 建立問題管理程序：
   - 制定問題管理的結構化程序，專注於長期解決方案，而不是快速修復。
   - 結合根本原因分析（RCA）技術，以調查和了解事件的根本原因。
   - 根據調查結果更新營運政策、程序和基礎設施，以防止重複發生。
3. 持續改善：
   - 培養不斷學習和改進的文化，鼓勵團隊積極識別和解決潛在問題。
   - 定期審查和修訂問題管理程序和工具，以配合不斷發展的業務和技術環境。
   - 在整個組織中分享見解和最佳實務，以建立更具彈性且更有效率的營運環境。
4. 使用 AWS Support：
   - 使用 AWS 支援資源，例如 AWS Trusted Advisor，以取得主動指引和最佳化建議。
   - Enterprise Support 客戶可以在重大事件期間存取 AWS Countdown 等專業計畫以取得支援。

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 Runbook 執行程序
- OPS07-BP04 使用教戰手冊調查問題

- OPS08-BP01 分析工作負載指標
- OPS11-BP02 執行事後分析

相關文件：

- AWS 安全事件回應指南
- AWS 事件偵測和回應
- AWS 雲端採用架構：Operations Perspective - 事件和問題管理
- DevOps 和 時代的事件管理 SRE
- PagerDuty - 什麼是事件管理？

相關影片：

- 來自 的熱門事件回應提示 AWS
- AWS re：Invent 2022 - Amazon Builders' Library25 年 Amazon 卓越營運
- AWS re：Invent 2022 - AWS 事件偵測和回應 （ SUP201 ）
- 從 推出 Incident Manager AWS Systems Manager

相關範例：

- AWS 主動式服務 – 事件管理研討會
- 如何使用 PagerDuty 和 自動化事件回應 AWS Systems Manager Incident Manager
- 將事件回應者與 中的隨時待命排程互動 AWS Systems Manager Incident Manager
- 在 中改善事件處理期間的可見性和協作 AWS Systems Manager Incident Manager
- 中的事件報告和服務請求 AMS

相關服務：

- Amazon EventBridge

OPS10-BP02 每個提醒都有一個程序

為系統中的每個提醒建立清晰明確的程序，對於有效且高效的事件管理至關重要。此做法可確保每個提醒都能產生特定且可行的回應，從而改善操作的可靠性和回應能力。

預期成果：每個提醒都會啟動特定且明確定義的回應計畫。在可能的情況下，回應會自動化，具有明確的擁有權和定義的呈報路徑。警示會連結至 up-to-date知識庫，以便任何運算子都能一致且有效地回應。回應迅速且全面一致，可提升營運效率和可靠性。

常見的反模式：

- 提醒沒有預定義的回應流程，導致臨時和延遲的解決方案。
- 提醒過載會導致重要提醒被忽略。
- 由於缺乏明確的擁有權和責任，提醒的處理不一致。

建立此最佳實務的優勢：

- 透過僅提高可操作的提醒來減少提醒疲勞。
- 減少操作問題的平均解決時間（MTTR）。
- 減少調查的平均時間（MTTI），這有助於減少 MTTR。
- 增強擴展操作回應的能力。
- 提高了處理操作事件中的一致性和可靠性。

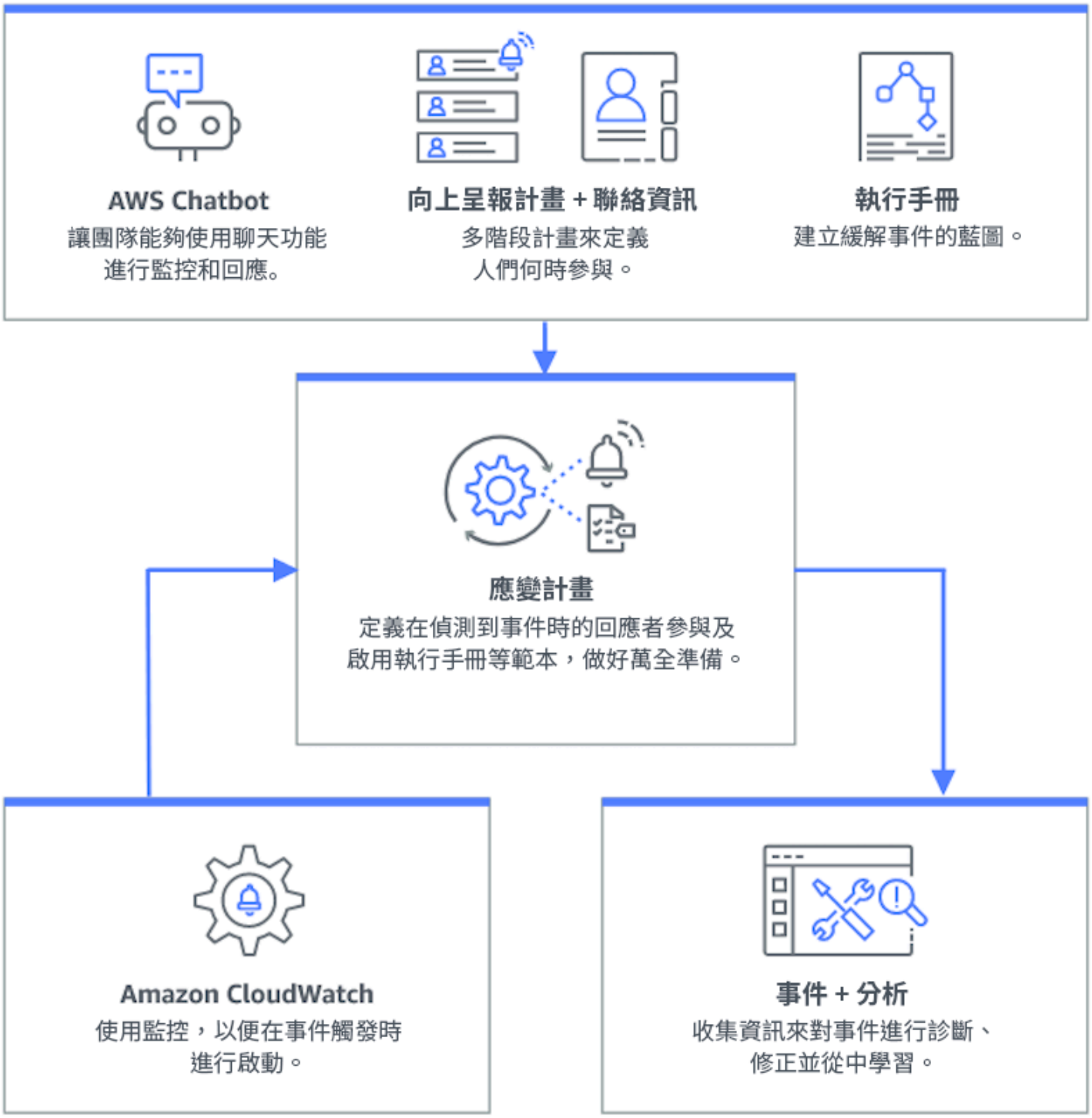
未建立此最佳實務時的曝險等級：高

實作指引

為每個提醒制定一個流程，包括：為每個提醒建立清晰的回應計畫；在可能的情況下自動化回應；並根據營運意見回饋和不斷發展的需求持續完善這些流程。

實作步驟

下圖說明 AWS Systems Manager Incident Manager 中的事件管理工作流程。它旨在透過自動建立事件以回應 Amazon CloudWatch 或 Amazon EventBridge的特定事件，快速回應操作問題。自動或手動建立事件時，Incident Manager 會集中管理事件、組織相關 AWS 資源資訊，並啟動預先定義的回應計劃。這包括為立即動作執行 Systems Manager Automation Runbook，以及在 中建立父項操作工作項目 OpsCenter ，以追蹤相關任務和分析。此簡化程序可加快並協調整個 AWS 環境的事件回應。



1. 使用複合警示：在 中建立複合警示 CloudWatch ，以分組相關警示、減少雜訊，並允許更有意義的回應。
2. 將 Amazon CloudWatch 警示與 Incident Manager 整合 設定 CloudWatch 警示，以在 中自動建立事件AWS Systems Manager Incident Manager。

3. 將 Amazon EventBridge 與 Incident Manager 整合：建立EventBridge 規則來回應事件，並使用定義的回應計劃建立事件。
4. 為 Incident Manager 中的事件做好準備：
   - 在 Incident Manager 中針對每種提醒類型建立詳細的回應計畫。
   - 透過與 Incident Manager 中的回應計畫相連的 AWS Chatbot 來建立聊天頻道，以便在 Slack、Microsoft Teams 和 Amazon Chime 等平台的事件期間進行即時通訊。
   - 將 Systems Manager Automation 執行手冊納入 Incident Manager 中，以推動對事件的自動回應。

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS08-BP04 建立可操作的警示

相關文件：

- AWS 雲端採用架構：Operations Perspective - 事件和問題管理
- 使用 Amazon CloudWatch 警示
- 設定 AWS Systems Manager Incident Manager
- 為 Incident Manager 中的事件做好準備

相關影片：

- 來自 的熱門事件回應提示 AWS

相關範例：

- AWS 研討會 - AWS Systems Manager Incident Manager - 自動化對安全事件的事件回應

OPS10-BP03 根據業務影響排定操作事件的優先順序

及時回應操作事件至關重要，但並非所有事件都是平等的。當您根據業務影響排定優先順序時，也可以優先處理可能產生重大後果的事件，例如安全、財務損失、違反法規或聲譽損害。

預期成果：根據對業務營運和目標的潛在影響，對營運事件的回應進行優先級排序。這使得回應高效且有效。

常見的反模式：

- 每個事件都按相同的緊急程度處理，會導致解決關鍵問題時出現混亂和延遲。
- 您無法區分高影響和低影響事件，導致資源分配錯誤。
- 您的組織缺乏明確的優先順序排定架構，導致對營運事件的回應不一致。
- 事件的優先級基於其報告的順序，而不是其對業務成果的影響。

建立此最佳實務的優勢：

- 確保關鍵業務功能首先獲得關注，將潛在損害降至最低。
- 改善多個並行事件期間的資源配置。
- 增強組織維持信任並遵守法規要求的能力。

未建立此最佳實務時的曝險等級：高

實作指引

當面對多個營運事件時，根據影響和緊迫性來確定優先順序的結構化方法至關重要。這種方法可幫助您做出明智的決策，直接在最需要的地方做出努力，並降低業務持續性的風險。

實作步驟

1. 評估影響：制定分類系統，根據事件對業務營運和目標的潛在影響來評估事件的嚴重性。下列範例顯示了影響類別：

影響層級	描述
高	影響許多員工或客戶，財務影響大，聲譽受損或受傷嚴重。
中	影響一組員工或客戶，中度財務影響，或中度聲譽受損。
低	影響個別員工或客戶，財務影響小，聲譽受損不嚴重。

2. 評估緊急性：在考量安全、財務影響和服務層級協議 （ ） 等因素的情況下，定義事件需要回應的速度緊急程度SLAs。下列範例示範緊急類別：

緊急程度	描述
高	大幅增加損壞、影響的時間敏感型工作、即將升級，或受影響的VIP使用者或群組。
中	損壞會隨著時間增加，或單一VIP使用者或群組受到影響。
低	邊際損害會隨著時間增加，或 non-time-sensitive受影響的工作。

3. 建立一個優先級矩陣：
   - 使用矩陣來交叉參考影響和緊迫性，將優先級別分配給不同的組合。
   - 讓負責營運事件回應的所有團隊成員都能存取和理解矩陣。
   - 下列範例矩陣會根據緊急性和影響來顯示事件嚴重性：

緊迫性和影響	高	中	低
高	嚴重	緊急	高
中	緊急	高	正常
低	高	正常	低

4. 培訓與溝通：對回應團隊進行優先級矩陣的培訓，並強調在事件期間遵循該矩陣的重要性。向所有利益相關者傳達優先級排序過程，以設定明確的期望。
5. 與事件回應整合：
   - 將優先級矩陣整合到您的事件回應計畫和工具中。
   - 盡可能自動化事件的分類和優先順序，以加快回應時間。
   - 企業支援客戶可利用 AWS 事件偵測與回應功能，為生產工作負載提供全年無休的主動監控和事件管理。
6. 審查和調整：定期審查優先順序排定程序的有效性，並根據業務環境中的意見回饋和變化進行調整。

資源

相關的最佳實務：

- 鼓勵 OPS03-BP03 升級
- OPS08-BP04 建立可操作的警示
- OPS09-BP01 測量操作目標和 KPIs 指標

相關文件：

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

OPS10-BP04 定義升級路徑

在您的事件回應協定中建立明確的呈報路徑，以促進及時且有效的活動。這包括指定升級提示、詳細說明升級程序，以及預先核准動作，以加快決策並縮短解決的平均時間（MTTR）。

預期成果：結構化且有效率的流程，可將事件呈報給適當的人員，將回應時間和影響降到最低。

常見的反模式：

- 復原程序不明確會導致在關鍵事件期間採取臨時應對措施。
- 當需要緊急行動時，缺少已定義的權限和擁有權會導致延遲。
- 利益相關者和客戶沒有按照預期得到通知。
- 重要決策被推遲。

建立此最佳實務的優勢：

- 透過預先定義的呈報程序來簡化事件回應。
- 透過預先核准的動作和明確的擁有權，減少停機時間。
- 根據事件嚴重性來改善資源配置和支援層級調整。
- 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

正確定義的升級路徑對於快速事件回應至關重要。 AWS Systems Manager Incident Manager 支援設定結構化升級計劃和隨叫隨到排程，這些排程會提醒適當的人員，讓他們準備好在事件發生時採取行動。

實作步驟

1. 設定升級提示：設定CloudWatch 警示以在 中建立事件AWS Systems Manager Incident Manager。
2. 設定隨時待命的排程：在 Incident Manager 中建立與您的呈報路徑保持一致的隨時待命的排程。為隨時待命的人員提供必要的權限和工具，以迅速採取行動。
3. 詳細說明呈報程序：
   - 確定應在哪些特定條件下呈報事件。
   - 在 Incident Manager 中建立呈報計畫。
   - 呈報渠道應包括聯絡人或隨時待命的時間表。
   - 定義團隊在每個呈報級別的角色和職責。
4. 預先核准的緩解措施：與決策者協同合作，針對預期情況預先核准動作。使用與 Incident Manager 整合的 Systems Manager Automation 執行手冊，加快事件解決速度。
5. 指定擁有權：針對呈報路徑的每個步驟，清楚識別內部擁有者。
6. 詳細說明第三方呈報：
   - 記錄第三方服務層級協議 （SLAs），並將其與內部目標保持一致。
   - 為事件期間的供應商溝通制定明確的協定。
   - 將供應商聯絡資訊整合至事件管理工具，以便直接存取。
   - 定期進行演練，包括第三方回應方案。
   - 保持供應商呈報資訊有據可查且易於存取。
7. 培訓和演練呈報計畫：對您的團隊進行呈報流程培訓，並定期進行事件回應演習或練習。企業支援客戶可申請事件管理研討會。
8. 持續改善：定期檢閱呈報路徑的有效性。根據事件發生後的經驗教訓和持續回饋來更新您的流程。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS08-BP04 建立可操作的警示
- OPS10-BP02 每個提醒都有一個程序
- OPS11-BP02 執行事後分析

相關文件：

- AWS Systems Manager Incident Manager 升級計畫
- 在 Incident Manager 中使用隨時待命的時間表
- 建立和管理執行手冊
- 使用 暫時提升存取管理 AWS IAM Identity Center
- Atlassian - 有效事件管理的呈報政策

OPS10-BP05 定義影響服務事件的客戶通訊計劃

在影響服務的事件中，有效的溝通對於維護與客戶的信任和透明度至關重要。明確定義的溝通計劃可協助您的組織在事件發生期間快速且清楚地分享資訊，包括內部和外部。

預期成果：

- 強大的溝通計畫可在影響服務的事件中有效地通知客戶和利益相關者。
- 溝通中的透明度可建立信任並減少客戶焦慮。
- 盡量減少服務影響事件對客戶體驗和業務運營的影響。

常見的反模式：

- 溝通不充分或延遲會導致客戶困惑和不滿。
- 過於技術化或模糊的消息傳遞無法傳達對使用者的實際影響。
- 沒有預先定義的溝通策略，導致不一致且被動的消息傳遞。

建立此最佳實務的優勢：

- 透過主動和清晰的溝通，增強客戶的信任和滿意度。
- 透過搶先解決客戶問題，減輕支援團隊的負擔。
- 改善有效管理事件並從中復原的能力。

未建立此最佳實務時的曝險等級：中

實作指引

為影響服務的事件制定全面的溝通計畫涉及多個方面，從選擇正確的渠道到精心製作消息和基調。該計畫應具有適應性、可擴展性，並適應不同的停機情況。

實作步驟

1. 定義角色和責任：
   - 指派一名主要事件管理者來監督事件回應活動。
   - 指定一名溝通管理者，其負責協調所有外部與內部溝通。
   - 包括支援管理者，以透過支援票證提供一致的溝通。
2. 識別通訊管道：選取管道，例如工作場所聊天、電子郵件、SMS、社交媒體、應用程式內通知和狀態頁面。這些渠道應具有彈性，並且能夠在影響服務的事件期間獨立運作。
3. 快速、清晰、定期地與客戶溝通：
   - 為各種服務損害場景開發模板，強調簡單性和基本細節。包括有關服務損害、預期解決時間和影響等資訊。
   - 透過推播通知、應用程式內通知、電子郵件、文字訊息、語音訊息和自訂渠道上的訊息，使用 Amazon Pinpoint 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、行動推播通知和簡訊提醒訂閱者。
   - 透過儀表板公開共用 Amazon CloudWatch 儀表板來傳達狀態。
   - 鼓勵社交媒體參與：
     - 積極監控社交媒體以了解客戶情緒。
     - 在社交媒體平台上發布公共更新和社區參與情況。
     - 準備範本以進行一致且清晰的社交媒體溝通。
4. 協調內部通訊：使用 AWS Chatbot 團隊協調和通訊等工具實作內部通訊協定。使用 CloudWatch 儀表板來通訊狀態。
5. 使用專用工具和服務協調溝通：
   - AWS Systems Manager Incident Manager 搭配 使用 AWS Chatbot ，設定專用聊天頻道，以在事件期間進行即時內部通訊和協調。
   - 使用 AWS Systems Manager Incident Manager Runbook，透過 Amazon Pinpoint 、Amazon 或第三方工具自動處理客戶通知SNS，例如在事件期間使用社交媒體平台。
   - 在執行手冊中合併核准工作流程，以便在傳送前選擇性地檢閱和授權所有外部通訊。

6. 實踐和改善：

- 針對溝通工具和策略的使用進行培訓。讓團隊能夠在事件發生時及時做出決策。
- 透過定期演習或演練日測試溝通計畫。使用這些測試來精簡消息傳遞並評估渠道的有效性。
- 實作意見回饋機制，以評估事件期間的溝通效率。根據意見回饋和不斷變化的需求不斷發展溝通計畫。

實作計劃的工作量：高

資源

相關的最佳實務：

- OPS07-BP03 使用 Runbook 執行程序
- OPS10-BP06 透過儀表板傳達狀態
- OPS11-BP02 執行事後分析

相關文件：

- Atlassian - 事件溝通最佳實務
- Atlassian - 如何編寫良好的狀態更新
- PagerDuty - 事件通訊指南

相關影片：

- Atlassian - 建立您自己的事件溝通計畫：事件範本

相關範例：

- AWS Health 儀表板
- AWS 狀態更新範例

OPS10-BP06 透過儀表板傳達狀態

使用儀表板作為戰略工具，將即時營運狀態和關鍵指標傳達給不同的受眾，包括內部技術團隊、領導層和客戶。這些儀表板提供系統運作狀態和業務績效的集中式視覺化呈現，從而提高透明度和決策效率。

預期成果：

- 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，減少頻繁的狀態請求。
- 在正常操作和事件期間增強實時決策。

常見的反模式：

- 加入事件管理通話的工程師要求更新狀態以加快速度。
- 依靠手動報告進行管理，這會導致延遲和潛在的不准確性。
- 事件發生期間，營運團隊經常因為狀態更新而受到干擾。

建立此最佳實務的優勢：

- 使利益相關者能夠立即存取關鍵資訊，有助於制定明智決策。
- 透過最大限度地減少手動報告和頻繁狀態查詢，減少操作效率低下問題。
- 透過即時掌握系統效能和業務指標，提高透明度和信任度。

未建立此最佳實務時的曝險等級：中

實作指引

儀表板可有效地傳達系統和業務指標的狀態，並可根據不同受眾群體的需求進行量身打造。Amazon CloudWatch 儀表板和 Amazon 等工具 QuickSight 可協助您建立互動式即時儀表板，以進行系統監控和商業智慧。

實作步驟

1. 確定利益相關者的需求：確定不同受眾群體的特定資訊需求，例如技術團隊、領導層和客戶。
2. 選擇正確的工具：選取適當的工具，例如用於系統監控的 Amazon CloudWatch 儀表板，以及用於互動式商業智慧的 Amazon QuickSight。
3. 設計高效儀表板：
   - 設計儀表板，以清楚呈現相關指標和 KPIs，確保其可理解且可操作。
   - 視需要整合系統層級與企業層級檢視。
   - 包括高階 (用於廣泛概述) 和低階 (用於詳細分析) 儀表板。

- 在儀表板中整合自動警示，以突顯重大問題。
- 使用重要指標閾值和目標為儀表板加上註釋，以實現即時可見性。

4. 整合資料來源：
   - 使用 Amazon CloudWatch 從各種 AWS 服務彙總和顯示指標，並從其他資料來源查詢指標，建立系統運作狀態和業務指標的統一檢視。
   - 使用 CloudWatch Logs Insights 之類的功能來查詢和視覺化來自不同應用程式和服務的記錄資料。
5. 提供自助服務存取：
   - 使用 CloudWatch 儀表板共用功能 與相關的利益相關者共用儀表板，以進行自助式資訊存取。
   - 確保儀表板易於存取並提供即時 up-to-date資訊。
6. 定期更新和完善：
   - 持續更新和完善儀表板，以滿足不斷變化的業務需求和利益相關者的意見回饋。
   - 定期檢閱儀表板，使其保持相關性並可有效傳達必要資訊。

資源

相關的最佳實務：

- OPS08-BP05 建立儀表板

相關文件：

- 建置用於檢視營運狀況的儀表板
- 使用 Amazon CloudWatch 儀表板
- 使用儀表板變數建立彈性儀表板
- 共用 CloudWatch 儀表板
- 從其他資料來源中查詢指標
- 將自訂小工具新增至 CloudWatch 儀表板

相關範例：

- 一個可觀測性研討會 - 儀表板

OPS10-BP07 自動化對事件的回應

自動化事件回應是快速、一致且無誤操作處理的關鍵。建立簡化的流程，並使用工具自動管理和回應事件，將手動干預降至最低，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。
- 一致且可靠的操作事件處理。
- 提高運營效率和系統可靠性。

常見的反模式：

- 手動事件處理會導致延遲和錯誤。
- 在重複的關鍵任務中，自動化被忽略。
- 重複的手動任務會導致警示疲勞，並遺漏重大問題。

建立此最佳實務的優勢：

- 加速事件回應，減少系統停機時間。
- 可靠的操作，自動化且一致的事件處理。

未建立此最佳實務時的曝險等級：中

實作指引

整合自動化以建立有效的操作工作流程，並將手動干預降至最低。

實作步驟

1. 識別自動化機會：確定自動化的重複性任務，例如問題修復、工單擴充、容量管理、擴展、部署和測試。
2. 識別自動化提示：
   - 使用 Amazon CloudWatch 警示動作 來評估和定義啟動自動回應的特定條件或指標。
   - 使用 Amazon EventBridge 來回應 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 、效能指標閾值 或 AWS 資源的狀態變更。
3. 實作事件驅動型自動化：

- 使用 AWS Systems Manager Automation Runbook 簡化維護、部署和修復任務。
- 在 Incident Manager 中建立事件會自動收集有關事件所涉及 AWS 資源的詳細資訊，並將其新增至事件。
- 使用 Quota Monitor for AWS 主動監控配額。
- 使用 AWS Auto Scaling 自動調整容量，以維持可用性和效能。
- 使用 Amazon CodeCatalyst將開發管道自動化。
- 煙霧測試或持續監控端點，APIs並使用合成監控 。

4. 透過自動化執行風險緩解：
    - 實作自動化安全回應，迅速解決風險。
    - 使用 AWS Systems Manager狀態管理器來減少組態偏差。
    - 使用 修復不合規資源 AWS Config 規則。

實作計劃的工作量：高

資源

相關的最佳實務：

- OPS08-BP04 建立可操作的警示
- OPS10-BP02 每個提醒都有一個程序

相關文件：

- 搭配 Incident Manager 使用 Systems Manager Automation 執行手冊
- 在 Incident Manager 中建立事件
- AWS 服務配額
- 監控資源使用情況並在接近配額時傳送通知
- AWS Auto Scaling
- 什麼是 Amazon CodeCatalyst？
- 使用 Amazon CloudWatch 警示
- 使用 Amazon CloudWatch 警示動作
- 使用 修復不合規資源 AWS Config 規則
- 使用篩選條件從日誌事件建立指標

- AWS Systems Manager State Manager

相關影片：

- 使用 建立 Automation Runbook AWS Systems Manager
- 如何在 上自動化 IT 操作 AWS
- AWS Security Hub 自動化規則
- 使用 Amazon CodeCatalyst 藍圖快速啟動軟體專案

相關範例：

- Amazon CodeCatalyst 教學課程：使用現代三層式 Web 應用程式藍圖建立專案
- 一個可觀測性研討會
- 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 擁有持續改善的流程
- OPS11-BP02 執行事後分析
- OPS11-BP03 實作意見回饋循環
- OPS11-BP04 執行知識管理
- OPS11-BP05 定義改進的驅動因素
- OPS11-BP06 驗證洞察
- OPS11-BP07 執行操作指標檢閱
- OPS11-BP08 記錄和分享所學課程

- OPS11-BP09 分配時間進行改善

OPS11-BP01 擁有持續改善的流程

根據內部和外部架構最佳實務評估您的工作負載。進行頻繁、有目的的工作負載審查。根據您的軟體開發步調制定改進機會的優先順序。

預期成果：

- 根據架構最佳實務頻繁分析工作負載。
- 在軟體開發過程中，改進機會與功能獲得同等優先級。

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。
- 您給予改進機會較低的優先級。與新功能相比，這些機會仍在待辦事項中。
- 沒有針對組織的最佳實務實作修改標準。

建立此最佳實務的優勢：

- 您的工作負載保留 up-to-date在架構最佳實務上。
- 您以故意的方式發展工作負載。
- 您可以利用組織最佳實務來改進所有工作負載。
- 您可以獲得具有累積影響的邊際收益，從而提高效率。

未建立此最佳實務時的曝險等級：高

實作指引

經常對工作負載進行架構審查。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特定最佳實務的記錄架構標準。
   a. 使用內部定義的標準進行這些審查。如果沒有內部標準，可使用 AWS Well-Architected Framework。

b. 使用 建立內部最佳實務的自訂鏡頭 AWS Well-Architected Tool ，並進行架構檢閱。

c. 請聯絡您的 AWS Solution Architect 或 Technical Account Manager，對您的工作負載執行引導式 Well-Architected Framework Review。

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。

實作計劃的工作量：低。您可以使用 AWS Well-Architected 架構來執行年度架構檢閱。

資源

相關的最佳實務：

- OPS11-BP02 執行事後分析
- OPS11-BP08 記錄和分享所學課程
- OPS04 實作可觀測性

相關文件：

- AWS Well-Architected Tool - 自訂鏡頭
- AWS Well-Architected 白皮書 - 審查程序
- 使用自訂鏡頭和 自訂 Well-Architected Reviews AWS Well-Architected Tool
- 在組織中實作 AWS Well-Architected Custom Lens 生命週期

相關影片：

- Well-Architected Labs - 100 級：自訂鏡頭開啟 AWS Well-Architected Tool
- AWS re：Invent 2023 - 擴展整個組織的 AWS Well-Architected 最佳實務

相關範例：

- AWS Well-Architected Tool

OPS11-BP02 執行事後分析

審查影響客戶的事件，並識別造成問題的因素和預防性動作。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。

預期成果：

- 您已建立包含事件後分析的事件管理程序。
- 您已制定可觀測性計畫，可收集有關事件的資料。
- 透過這些資料，您就可以了解並收集支援事件後分析程序的指標。
- 您可以從事件中學習，以改善未來的成果。

常見的反模式：

- 管理應用程式伺服器。大約每 23 小時 55 分鐘，您的所有活動工作階段都會終止。您試圖確定應用程式伺服器出了什麼問題。您懷疑這可能是網路問題，但由於網路團隊太忙而無法支援您，因此無法與他們合作。您缺乏可遵循的預定義流程來獲得支援並收集所需資訊以確定發生了什麼。
- 您的工作負載中有資料遺失。這是第一次發生，原因尚不清楚。您認為這並不重要，因為您可以重新建立資料。資料遺失開始更頻繁地出現，從而影響客戶。當您還原遺失的資料時，這也會為您帶來額外的操作負擔。

建立此最佳實務的優勢：

- 您有一個預先定義的程序來判斷造成事故的元件、條件、動作和事件，這有助於您找出改進機會。
- 可以使用事件後分析的資料進行改善。

未建立此最佳實務時的曝險等級：高

實作指引

使用程序判斷成因。審查所有影響客戶的事件。建立程序來識別和記錄事件的成因，以便您可以制定緩解措施來限制或防止事件再次發生。另外，您還可以制定快速有效地做出回應的程序。酌情溝通事件根本原因，並根據目標受眾量身定制溝通方式。在組織內公開分享學習成果。

實作步驟

1. 收集諸如部署變更、組態變更、事件開始時間、警示時間、參與時間、緩解開始時間和事件解決時間等指標。
2. 描述時間軸上的關鍵時間點，以了解事故的事件。
3. 請提出以下問題：
    a. 您可以縮短偵測時間嗎？

b. 是否有指標和警示的更新，可以更快地檢測到事件？
c. 可以改善診斷時間嗎？
d. 回應計畫或呈報計劃是否有更新，可以更快地吸引合適的回應方？
e. 可以改善緩解時間嗎？
f. 是否有可以新增或改善的執行手冊或說明手冊步驟？
g. 可以防止未來的事件發生嗎？

4. 建立檢查清單和動作。追蹤並傳遞所有動作。

實作計劃的工作量：中

資源

相關的最佳實務：

- OPS11-BP01 擁有持續改善的流程
- OPS 4 - 實作可觀測性

相關文件：

- 在 Incident Manager 中執行事件後分析
- 營運準備情況審核

OPS11-BP03 實作意見回饋循環

回饋迴圈提供可推動決策的可行洞見。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。

回饋迴圈分為兩類：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。

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畫。

客戶範例

AnyCompany 零售建立了一個 Web 表單，客戶可以提供意見回饋或報告問題。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 來建立和追蹤這些改進，如 OpsItems。

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益相關者，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。
        - 停止是指您希望團隊停止做的任何事。
        - 開始是指您希望開始執行的想法。
        - 保持是指您希望持續執行的項目。
    - 詢問在場人士的想法，收集利益相關者的回饋。
    - 排列回饋的優先順序。將動作和利益相關者指派至任何「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益相關者告知最新的狀態。

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：內部利益相關者可以使用回饋迴圈來表達需要和需求。
- OPS11-BP02 執行事後分析：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 執行操作指標檢閱：營運指標審查會找出趨勢和待改善的地方。

相關文件：

- 建置 時要避免的 7 個陷阱 CCOE
- 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 Review 建立意見回饋循環
- IBM 車庫方法 - 保留回溯性
- 投資 – PDCS週期
- 最大化開發人員的效能 (作者：Tim Cochran)

- 操作就緒審核（ORR）白皮書 - 迭代
- ITIL CSI - 持續服務改進
- 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 - Q nABot on AWS
- 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曝險等級：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 零售託管儲存所有知識的內部 Wiki。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。跨組織取得利益相關者的同意。
   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。與利益相關者合作來排列新增知識的優先順序。
4. 定期與利益相關者合作，以識別 out-of-date資訊並將其封存或更新。

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 記錄和分享所學課程 - 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 定義改進的驅動因素

確定改進驅動因素，以幫助您依據資料和回饋迴圈來評估改進機會並排定其優先順序。探索系統和程序中的改進機會，並在適當的情況下自動化。

預期成果：

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

常見的反模式：

- 可以從整個環境中收集資料，但不會關聯事件和活動。
- 您可以從您的整個資產中收集詳細資訊，並驅動高 Amazon CloudWatch 和 AWS CloudTrail 活動和成本。但是，您不會有目的地使用此資料。
- 在定義改進驅動因素時，您不會考慮業務成果。
- 您不會測量新功能的效果。

建立此最佳實務的優勢：

- 透過確定改進標準，可以將事件型動機或情緒投資的影響降到最低。
- 您可以回應商業活動，而不僅僅是技術事件。
- 測量您的環境，以確定需要改進的領域。

未建立此最佳實務時的曝險等級：中

實作指引

- 了解改進驅動因素：僅在支援理想結果時才對系統進行變更。
  - 所需能力：在評估改進機會時，評估所需的功能和能力。
    - 新功能 AWS
  - 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。追蹤合適的選項，並尋求優化機會。
    - AWS 最新安全公告
    - AWS Trusted Advisor
    - 雲端智慧儀表板
  - 合規要求：在審查改進機會時，評估保持法規、政策的遵從性或保持受到第三方支援所需的更新和變更。
    - AWS 合規
    - AWS 合規計劃
    - AWS 合規性最新資訊

資源

相關的最佳實務：

- OPS01 組織優先順序
- OPS02 關係和所有權
- OPS04-BP01 識別關鍵績效指標
- OPS08 利用工作負載可觀測性
- OPS09 了解營運運作狀態
- OPS11-BP03 實作意見回饋循環

相關文件：

- Amazon Athena
- Amazon QuickSight
- AWS 合規

- AWS 合規性最新資訊
- AWS 合規計劃
- AWS Glue
- AWS 最新安全公告
- AWS Trusted Advisor
- 將日誌資料匯出至 Amazon S3
- AWS最新消息
- 以客戶為中心的創新的必要條件
- 數位轉型：炒作還是戰略需要？

相關影片

- AWS re：Invent 2023 - 使用 AWS Support （ SUP310 ） 提高操作效率和彈性

OPS11-BP06 驗證洞察

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。

預期成果：

- 定期與企業擁有者審查洞見。企業擁有者為新獲得的洞察提供額外的內容。
- 可以檢閱洞見並請求技術同儕們的意見回饋，並在各個團隊之間分享您的學習成果。
- 可以發布資料和洞見，供其他技術和業務團隊審核。將所學知識納入其他部門的新實務中。
- 與資深主管一起總結和審查新洞見。資深主管使用新洞見來定義策略。

常見的反模式：

- 您發佈了一個新功能。此功能會變更部分客戶行為。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。
- 您推送新的更新並忽略重新整理您的 CDN。CDN 快取不再與最新版本相容。測量發生錯誤的請求百分比。您的所有使用者在與後端伺服器通訊時，都會報告 HTTP 400 個錯誤。調查用戶端錯誤，並發現由於您測量了錯誤的維度，所以時間被浪費了。
- 您的服務水準協議規定正常執行時間為 99.9%，而您的復原點目標為四小時。服務擁有者維護系統為零停機時間。實作昂貴且複雜的複寫解決方案，這會浪費時間和金錢。

建立此最佳實務的優勢：

- 與企業擁有者和領域專家驗證洞見時，建立共識並更有效地引導改進。
- 您會發現隱藏的問題，並將其納入未來決策中。
- 重點從技術成果轉移到業務成果。

未建立此最佳實務時的曝險等級：中

實作指引

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

資源

相關的最佳實務：

- OPS01-BP06 在管理效益和風險時評估權衡
- OPS團隊之間的 02-BP06 責任已預先定義或協商
- OPS11-BP03 實作意見回饋循環

相關文件：

- 設計 Cloud Center of Excellence （ CCOE ）

相關影片：

- 建立可觀測性以提高復原能力

OPS11-BP07 執行操作指標檢閱

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。

預期成果：

- 經常審查影響業務的指標

- 可以透過可觀測性功能來偵測和審查異常
- 可以使用資料來支援業務成果和目標

常見的反模式：

- 維護時段會中斷重要的零售促銷活動。如果還有其他影響企業的事件，企業仍然不知道是否有可能會延遲的標準維護時段。
- 您經歷了長時間的中斷，因為您經常使用組織中過時的程式庫。之後您已遷移到支援的程式庫。組織中的其他團隊不知道他們正面臨風險。
- 您不會定期檢閱客戶 的達成情況SLAs。您即將不符合您的客戶 SLAs。未滿足客戶 會受到財務處罰SLAs。

建立此最佳實務的優勢：

- 當您定期開會以審查營運指標、事件和事故時，可以在團隊之間保持共識。
- 您的團隊會定期會面，以檢閱指標和事件，這些指標和事件可讓您針對風險採取行動並識別客戶SLAs。
- 您分享學到的經驗教訓，為業務成果的優先順序和有針對性的改進提供資料。

未建立此最佳實務時的曝險等級：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益相關者進行互動，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 檢閱操作指標並排定改善優先順序
- OPS10-BP01 使用事件、事件和問題管理的程序

相關文件：

- Amazon CloudWatch
- Amazon CloudWatch 指標和維度參考
- 發佈自訂指標
- 使用 Amazon CloudWatch 指標
- 使用 的儀表板和視覺化 CloudWatch

OPS11-BP08 記錄和分享所學課程

記錄並分享在操作活動中獲得的經驗，以便您可以在內部以及跨團隊使用它們。應分享您的團隊獲得的經驗，以提高整個組織的效益。共用資訊和資源，以防止可避免的錯誤及簡化開發工作，並專注於交付所需的功能。

使用 AWS Identity and Access Management （ IAM ） 來定義許可，允許控制存取您想要在帳戶內和帳戶間共用的資源。

預期成果：

- 使用版本控制的儲存器來分享應用程式程式庫、執行指令碼的程序、程序文件及其他系統文件。
- 可以將基礎設施標準共用為版本控制的 AWS CloudFormation 範本。
- 審核團隊學到的經驗教訓。

常見的反模式：

- 您經歷了長時間的中斷，因為您的組織經常使用錯誤的程式庫。之後您已遷移到可靠的程式庫。組織中的其他團隊不知道他們正面臨風險。沒有人記錄和分享有關此程式庫的經驗，他們沒有意識到風險。
- 您已在內部共用的微型服務中找出導致工作階段終止的邊緣案例。您已更新對服務的呼叫，以避免此邊緣案例。組織中的其他團隊不知道他們正面臨風險。
- 您已找到一種方法，可大幅降低其中一個微服務的CPU使用率需求。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：分享經驗教訓以支援改進並最大限度地發揮經驗的優勢。

未建立此最佳實務時的曝險等級：低

實作指引

- 記錄和分享獲得的經驗：制定程序來記錄從執行營運活動和追溯性分析中學到的經驗教訓，以便其他團隊可以使用。
- 分享經驗：制定程序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、指南、管控和最佳實務；透過公共儲存庫分享指令碼、程式碼和程式庫。
  - 委派對您 AWS 環境的存取
  - 共用 AWS CodeCommit 儲存庫

資源

相關的最佳實務：

- OPS團隊之間的 02-BP06 責任已預先定義或協商
- OPS05-BP01 使用版本控制
- OPS05-BP06 共用設計標準
- OPS11-BP03 實作意見回饋循環
- OPS11-BP07 執行操作指標檢閱

相關文件：

- 使用 docs-as-code 解決方案減少專案延遲

相關影片：

- 委派對您 AWS 環境的存取
- AWS Support為您提供支援 | 探索事件管理桌上模擬演練

OPS11-BP09 分配時間進行改善

在流程中投入時間和資源，以持續逐漸改善。

預期成果：

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

- 您會執行遊戲日，並使用 Fault Injection Service （FIS） 提供團隊在類似生產環境中執行實驗所需的控制項和防護機制。

常見的反模式：

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

建立此最佳實務的優勢：透過在程序中投入時間和資源，您可以實現持續逐漸改善。

未建立此最佳實務時的曝險等級：低

實作指引

- 分配時間進行改進：在流程中投入時間和資源，以持續逐漸改善。
- 實作變更以改進和評估結果，從而確定成功與否。
- 如果結果未能達到目標，並且改進仍然是優先事項，則應採取替代行動方案。
- 在演練日模擬生產工作負載，並利用這些模擬中的知識進行改進。

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 - 使用 AWS Fault Injection Service 改善應用程式彈性

安全

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。可以在安全支柱白皮書中找到實作指引。

最佳實務領域

- 安全基礎
- 身分與存取管理
- 偵測
- 基礎設施保護
- 資料保護
- 事件回應
- 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，您必須將整體的最佳實務套用到每個安全區域。採用您在組織和工作負載層級於卓越營運中定義的要求和程序，並將其應用於所有區域。隨時掌握 AWS 和 產業建議和威脅情報，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 帳戶分隔工作負載
- SEC01-BP02 安全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
- SEC01-BP05 減少安全管理範圍
- SEC01-BP06 自動化標準安全控制的部署
- SEC01-BP07 使用威脅模型識別威脅並排定緩解優先順序
- SEC01-BP08 定期評估和實作新的安全服務和功能

SEC01-BP01 使用 帳戶分隔工作負載

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：一種帳戶結構，可將雲端作業、不相關的工作負載和環境隔離為單獨的帳戶，從而提高整個雲端基礎設施的安全性。

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。
- AWS 對服務、資源和區域的存取進行集中管理。
- 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 在在不同敏感層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雲端工作負載，以利用此隔離界限。如需在 上多帳戶策略的概念、模式和實作指引 AWS，請參閱使用多個帳戶組織您的 AWS 環境。

當您在中央管理 AWS 帳戶 下擁有多個 時，您的帳戶應組織為由組織單位層 （ ） 定義的階層OUs。然後，安全控制可以組織並套用到 OUs和成員帳戶，為組織中的成員帳戶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。

AWS Organizations 和 AWS Control Tower是兩項服務，可用來在 AWS 環境中實作和管理此多帳戶結構。 AWS Organizations 可讓您將帳戶組織到由一或多個 層定義的階層OUs，每個 OU 都包含數個成員帳戶。服務控制政策 （ SCPs ） 允許組織管理員在成員帳戶上建立精細的預防性控制，AWS

Config並可用於在成員帳戶上建立主動和偵測性控制。許多 AWS 服務與 整合 AWS Organizations，以提供委派的管理控制，並跨組織中所有成員帳戶執行服務特定的任務。

在 之上分層 AWS Organizations，為具有登陸區域 的多帳戶 AWS 環境AWS Control Tower提供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。與 AWS Organizations相比，Control Tower 具有數個好處。提供改進的帳戶管控的三個優點是：

- 整合式強制性安全控制，會自動套用至獲准加入組織的帳戶。
- 可以為指定 集開啟或關閉的選用控制項OUs。
- AWS Control Tower Account Factory 會在您的組織中自動部署包含預先核准基準和組態選項的帳戶。

實作步驟

1. 設計組織單位結構：設計合理的組織單位結構可減少建立及維護服務控制政策及其他安全性控制所需的管理負擔。您的組織單位結構應與業務需求、資料敏感度和工作負載結構保持一致。
2. 為多帳戶環境建立登陸區域：登陸區域可提供一致的安全性和基礎設施基礎，您的組織可以從中快速開發、啟動和部署工作負載。可以使用自訂建置的登陸區域或 AWS Control Tower 來協調您的環境。
3. 建立防護機制：透過登陸區域為您的環境實作一致的安全防護機制。 AWS Control Tower 提供可部署的強制性控制與選擇性控制清單。實作 Control Tower 時會自動部署強制性控制。檢閱強烈建議的控制和選擇性控制清單，並實作符合您需求的控制。
4. 限制對新新增區域 的存取：對於新 AWS 區域，使用者和角色等IAM資源只會傳播到您指定的區域。使用 Control Tower 時，可以透過主控台執行此動作，或在 IAM 中調整許可政策 AWS Organizations。
5. 考慮 AWS CloudFormation StackSets： StackSets 協助您從核准的範本將資源，包括IAM政策、角色和群組部署到不同的 AWS 帳戶 和 區域。

資源

相關的最佳實務：

- SEC02-BP04 依賴集中式身分提供者

相關文件：

- AWS Control Tower

- 《AWS 安全性稽核指南》
- IAM 最佳實務
- 使用 CloudFormation StackSets 跨多個 AWS 帳戶 和 區域佈建資源
- 組織 FAQ
- AWS Organizations 術語和概念
- AWS Organizations 多帳戶環境中的服務控制政策最佳實務
- AWS 帳戶管理參考指南
- 使用多個帳戶組織您的 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 建置和管理多個帳戶 AWS Control Tower
- 為現有組織啟用 Control Tower

相關研討會：

- Control Tower Immersion Day

SEC01-BP02 安全帳戶根使用者和屬性

根使用者是 中最高權限的使用者 AWS 帳戶，具有帳戶內所有資源的完整管理存取權，在某些情況下，安全政策不會限制這些權限。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減少因濫用根使用者憑證而造成意外或蓄意損壞的機會。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畫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。

- 不將 DNS、電子郵件伺服器和電話供應商作為關鍵安全周邊的一部分處理，因為這些會在帳戶復原流程中使用。

建立此最佳實務的優勢：保護對根使用者的存取，可建立對帳戶中的動作加以控制和稽核的信心。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 提供許多工具來協助保護您的帳戶。然而，由於預設情況下不會開啟其中一些措施，因此您必須採取直接行動加以實作。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。

當您第一次建立 時 AWS 帳戶，您會從一個身分開始，該身分可完全存取 帳戶中的所有 AWS 服務和資源。此身分稱為 AWS 帳戶 根使用者。您可以使用您用來建立帳戶的電子郵件地址和密碼，以根使用者的身分登入。由於授予 AWS 根使用者的存取提升，您必須限制 AWS 根使用者的使用，才能執行特別需要 的任務。根使用者登入憑證必須受到嚴密保護，且 AWS 帳戶 根使用者應一律使用多重要素身分驗證 （MFA）。

除了使用使用者名稱、密碼和多重要素身分驗證 （MFA） 裝置登入根使用者的正常身分驗證流程之外，還有帳戶復原流程，可讓您登入 AWS 帳戶 根使用者，並存取與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。此外，也請考量與根使用者相關聯的電子郵件地址託管在相同 的電子郵件伺服器或網域名稱服務 （DNS） 資源上的潛在循環相依性 AWS 帳戶。

使用 時 AWS Organizations， AWS 帳戶 每個都有多個具有根使用者的 。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。如適用，建議會交互參照至 CIS AWS Foundations 基準 1.4.0 版。 除了這些步驟之外，請參閱保護您 AWS 帳戶 和資源的AWS 最佳實務準則。

預防性控制

1. 為帳戶設定準確的聯絡資訊。
    a. 此資訊用於遺失密碼復原流程、遺失MFA裝置帳戶復原流程，以及與團隊進行關鍵安全相關通訊。

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。如果存在存取金鑰，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. 如果根使用者存取金鑰已存在，您應該使用這些金鑰轉換程序，以使用來自 AWS Identity and Access Management（IAM）角色的臨時存取金鑰，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 建立新的成員帳戶，則新成員帳戶上根使用者的初始密碼會設定為不會公開給您的隨機值。如有需要，請考慮使用來自您的 AWS Organization 管理帳戶的密碼重設流程來存取成員帳戶。

b. 對於獨立 AWS 帳戶 或管理 AWS 組織帳戶，請考慮為根使用者建立並安全地儲存憑證。針對根使用者MFA使用 。

4. 在 AWS 多帳戶環境中為成員帳戶根使用者使用預防性控制。

a. 考慮針對成員帳戶使用不允許為根使用者建立根存取金鑰預防性防護機制。

b. 考慮針對成員帳戶使用不允許以根使用者身分執行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. 為根使用者開啟多重要素驗證（MFA），特別是 AWS Organizations 管理（付款人）帳戶（CIS 1.5）。

c. 考慮硬體MFA裝置以確保復原能力和安全性，因為單次使用裝置可以降低包含您MFA程式碼的裝置可能重複使用用於其他目的的機會。確認以電池供電的硬體MFA裝置定期更換。（CIS 1.6）

- 若要MFA為根使用者設定 ，請遵循建立虛擬MFA或硬體MFA裝置 的指示。

d. 考慮註冊多個MFA裝置進行備份。每個帳戶最多允許 8 個MFA裝置。

- 請注意，如果MFA裝置遺失 ，為根使用者註冊多個裝置會自動關閉復原帳戶的流程。 MFA

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。請勿以需要存取密碼 AWS 帳戶 來取得密碼的方式存放密碼。

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 保護的根使用者MFA不依賴密碼作為身分驗證的單一因素。
- 定期變更根使用者密碼可降低不慎公開密碼遭到濫用的風險。

偵測性控制

- 建立警示以偵測根憑證的使用（CIS 1.7）。Amazon GuardDuty 可以透過RootCredentialUsage調查結果監控和提醒根使用者API憑證用量。
- 評估和實作 AWS Well-Architected Security Pillar 一致性套件 AWS Config中包含的偵測性控制項，或者如果使用 AWS Control Tower，則建議使用 Control Tower 中可用的強烈建議控制項。

操作指引

- 確定組織內誰應該存取根使用者憑證。
  - 使用兩人規則，讓任何人都無法存取所有必要的憑證MFA，並取得根使用者存取權。
  - 確認組織（用於密碼重設和MFA重設流程）維持對與帳戶相關聯的電話號碼和電子郵件別名的控制，而非單一個人。
- 僅透過例外狀況使用根使用者（CIS 1.7）。
  - AWS 根使用者不得用於日常任務，即使是管理任務。只有以根使用者身分登入，才能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，以及替代連絡人下列出的電子郵件地址。監控這些電子郵件收件匣，查看您可能接收的來自 <abuse@amazon.com> 的安全通知。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。請參閱 AWS Security Incident Response Guide 和安全支柱白皮書中「事件回應」一節中的最佳實務，了解如何為 AWS 帳戶建立事件回應策略的詳細資訊。

資源

相關的最佳實務：

- SEC01-BP01 使用 帳戶分隔工作負載
- SEC02-BP01 使用強大的登入機制
- SEC03-BP02 授予最低權限存取
- SEC03-BP03 建立緊急存取程序
- SEC10-BP05 佈建前存取

相關文件：

- AWS Control Tower
- 《AWS 安全性稽核指南》
- IAM 最佳實務
- Amazon GuardDuty – 根憑證用量提醒
- Step-by-step 監控根憑證使用的指引，透過 CloudTrail
- MFA 權杖已核准搭配 使用 AWS
- 在 上實作休息玻璃存取 AWS
- 您的 中要改進的前 10 個安全項目 AWS 帳戶
- 如果發現我的 AWS 帳戶中有未經授權的活動，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 re：inforce 2022 的 AWS 根憑證 – 安全最佳實務搭配 AWS AWS IAM

相關範例和實驗室：

- 實驗室： AWS 帳戶 設定和根使用者

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：企業的安全控制目標是明確定義的，並符合您的合規要求。控制項是透過自動化和政策來實作和強制執行的，並持續評估其在達成目標方面的有效性。在一個時間點和一段時間內的有效性證據可以很容易地向稽核人員報告。

常見的反模式：

- 您的企業對可保證安全的法規要求、市場預期和業界標準並未充分了解
- 網路安全架構和控制目標不符合業務需求

- 控制措施的實作並未以可衡量的方式與您的控制目標保持一致
- 您不使用自動化來報告控制措施的有效性

未建立此最佳實務時的曝險等級：高

實作指引

有許多常見的網路安全框架可以構成安全控制目標的基礎。考慮企業的法規要求、市場期望和業界標準，以決定哪些架構最能支援您的需求。範例包括 AICPA SOC 2、HITRUST、PCI-DSS、ISO27001 和 NIST SP 800-53。

對於您識別的控制目標，了解您取用 AWS 的服務如何協助您實現這些目標。使用 AWS Artifact 尋找與您的目標架構一致的文件和報告，描述您負責的其餘範圍所涵蓋的責任範圍 AWS 和指引。如需符合各種架構控制聲明的詳細服務特定指引，請參閱 AWS Customer Compliance Guides。

當您定義實現目標的控制措施時，使用預防性控制措施對執行進行整理，並使用偵測性控制來自動化緩解措施。 AWS Organizations 使用 服務控制政策 （SCP），協助防止整個 的不合規資源組態和動作。在 AWS Config 中實作規則以監控和報告不合規的資源，然後在對其行為有信心後，將規則切換為強制執行模型。若要部署符合網路安全架構的預定義和受控規則集，請首先評估 AWS Security Hub 標準的使用。 AWS 基礎服務最佳實務 （FSBP） 標準和CIS AWS 基礎基準是很好的起點，其控制與多個標準架構之間共用的許多目標保持一致。如果 Security Hub 本質上沒有所需的控制偵測，則可以使用 AWS Config 一致性套件來補充它。

視需要使用 AWS 全球安全與合規加速 （GSCA） 團隊建議的APN合作夥伴套件，向安全顧問、諮詢機構、證據收集和報告系統、稽核人員和其他補充服務取得協助。

實作步驟

1. 評估常見的網路安全架構，並使您的控制目標與選擇的目標保持一致。
2. 使用 取得架構指南和責任的相關文件 AWS Artifact。了解哪些合規部分屬於共同責任模型 AWS ，以及哪些部分是您的責任。
3. 使用 SCPs、資源政策、角色信任政策和其他防護機制，以防止不合規的資源組態和動作。
4. 評估部署符合您控制目標的 Security Hub 標準和 AWS Config 一致性套件。

資源

相關的最佳實務：

- SEC03-BP01 定義存取要求
- SEC04-BP01 設定服務和應用程式記錄
- SEC07-BP01 了解您的資料分類方案
- OPS01-BP03 評估治理要求
- OPS01-BP04 評估合規要求
- PERF01-BP05 使用政策和參考架構
- COST02-BP01 根據您的組織需求制定政策

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

透過監控業界威脅情報刊物和資料摘要以取得更新，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。 在識別新威脅時，使用自動化技術偵測潛在的漏洞和暴露。您可以採取緩解措施對抗這些威脅。 您採用 AWS 的服務會自動更新為最新的威脅情報。

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庫存清單，而這些都需要人員審查才能得知是否有潛在的漏洞和暴露。
- 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：使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

的潛在漏洞或暴露，如此就能協助您事先預測並快速降低風險。 這有助於控制與漏洞緩解相關的時間和成本。

未建立此最佳實務時的曝險等級：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。 如需已知對手戰術、技術和程序的文件（），請參閱 MITRE ATT&CK 知識庫TTPs。檢閱 MITRE的常見漏洞與暴露（CVE）清單，以隨時掌握您所依賴產品中已知漏洞的資訊。使用 Open Worldwide Application Security Project（OWASP）的熱門OWASP前 10 大專案，了解 Web 應用程式的關鍵風險。

使用 AWS 適用於 AWS 的安全公告，隨時掌握安全事件和建議的修復步驟CVEs。

為了減少您保持最新狀態的整體工作量和額外負荷，請考慮使用 AWS 服務，隨著時間自動整合新的威脅情報。 例如，Amazon GuardDuty 會隨時掌握最新的產業威脅情報，以偵測帳戶中的異常行為和威脅簽章。 Amazon Inspector 會自動將CVEs用於其持續掃描功能的 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時自動更新。

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。
- 考慮採用自動整合新威脅情報的服務，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排定緩解的優先順序
- OPS01-BP05 評估威脅態勢
- OPS11-BP01 擁有持續改善的程序

SEC01-BP05 減少安全管理範圍

判斷您是否可以使用將某些控制項的管理轉移到 AWS（受管服務）的服務 AWS 來減少安全範圍。這些服務有助於減少安全維護任務，例如基礎設施佈建、軟體設定、修補或備份。

預期結果：您在為工作負載選取 AWS 服務時，會考慮安全管理的範圍。除了其他 Well-Architected 考量之外，管理額外負荷和維護任務的成本（總擁有成本，或 TCO）會權衡您選擇的服務成本。您可以將 AWS 控制和合規文件納入控制評估和驗證程序。

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，未將安全管理任務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：使用受管服務可以減輕您管理營運安全控制的整體負擔，進而降低安全風險和總體擁有成本。否則，時間可能會花費在某些安全任務上，而無法轉投入其他為企業創造更多價值的任務。受管服務也可以透過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曝險等級：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。在 Amazon EC2執行個體上安裝和執行技術通常需要您承擔整體安全責任的最大份額。為了協助減少操作特定控制項的負擔，請識別可降低您共同責任模型範圍的 AWS 受管服務，並了解如何在現有架構中使用它們。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。受管服務可以將某些要求的合規轉移到 AWS。與您的合規團隊討論其稽核相關稽核 AWS 報告中所操作之服務的各個層面，以及管理和接受控制陳述式的自在程度。您可以AWS Artifact向稽核人員或監管機構提供 中的稽核成品，作為 AWS 安全控制的證據。您也可以使用某些 AWS 稽核成品提供的責任指南，以及AWS 客戶合規指南來設計您的架構。本指引可協助您確定應採取哪些額外的安全控制，以便支援系統的特定使用案例。

使用 受管服務時，請熟悉將資源更新至較新版本的程序（例如，更新 Amazon 管理的資料庫版本 RDS，或 AWS Lambda 函數的程式設計語言執行時間）。雖然受管服務可能會自動為您執行此操作，但設定更新的時間並了解對操作的影響仍然是您的責任。AWS Health 等工具可以協助您在整個環境中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遷移至 AWS，請考慮減少管理（時間和費用）並降低風險，同時評估您是否應該重新託管、重構、修改、重建或取代工作負載。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. 請考慮實作 受管服務，例如 Amazon RDS，而不是安裝和管理您自己的技術部署。
3. 使用 中的責任指引 AWS Artifact ，協助判斷您應該為工作負載設定的安全控制。
4. 保留使用中的資源清單，並持續 up-to-date使用新的服務和方法，以識別減少範圍的新機會。

資源

相關的最佳實務：

- PERF02-BP01 為您的工作負載選取最佳運算選項
- PERF03-BP01 使用最能夠支援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：

- 的計劃生命週期事件 AWS Health

相關工具：

- AWS Health
- AWS Artifact
- AWS 客戶合規指南

相關影片：

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

SEC01-BP06 自動化標準安全控制的部署

在開發和部署整個 AWS 環境的標準安全控制時，套用現代 DevOps 實務。 使用基礎架構作為程式碼 IaC）範本定義標準安全控制和組態、擷取版本控制系統中的變更、測試變更作為 CI/CD 管道的一部分，以及自動部署 AWS 環境的變更。

預期成果：IaC 範本會擷取標準化的安全控制，並將其遞交至版本控制系統。 CI/CD 管道位於可偵測變更、自動測試和部署 AWS 環境的位置。 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提醒。 工作負載會部署到已採取標準控制的環境中。 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。
- 仰賴個別工作負載團隊手動實作中央團隊定義的控制。
- 仰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。
- 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，可讓您使用版本控制系統追蹤和比較一段時間的變更。 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，從而提高成功部署的機會，並減少手動重複任務。 為工作負載團隊提供自助服務機制來部署核准的服務和組態，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。

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 中所述的實務 使用帳戶 分隔工作負載時，您會針對使用 管理的不同環境，最終獲得多個 AWS 帳戶 AWS Organizations。 雖然其中每個環境和工作負載可能需要不同的安全控制，但您可以將整個組織的某些安全控制標準化。 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套用至基礎設施佈建，也可以使用 IaC 來定義和部署標準安全控制。

請盡可能以宣告的方式定義安全控制 (例如在 AWS CloudFormation 中)，並將其儲存在來源控制系統中。 使用 DevOps 實務來自動化部署控制項以取得更可預測的版本、使用 等工具進行自動測試AWS CloudFormation Guard，以及偵測部署控制項與所需組態之間的偏離。 您可以使用 AWS CodePipeline、AWS CodeBuild 和 AWS CodeDeploy 等服務來建構 CI/CD 管道。請考慮使用多個帳戶組織您的 AWS 環境，在與其他部署管道分開的自己的帳戶中設定這些服務的指南。

您也可以定義範本，以標準化定義和部署 、 AWS 帳戶服務和組態。 此技術可讓中央安全團隊管理這些定義，並透過自助服務方法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，讓工作負載團隊可以將這些產品納入自己的管道部署中。 如果您使用的是 AWS Control Tower，某些範本和控制可以作為起點。 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 。 此功能有助於消除對中央團隊的依賴，以在工作負載團隊視需要識別新帳戶時核准和建立新帳戶。 您可能需要這些帳戶，以便根據諸如提供的功能、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. 確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. 為工作負載團隊建立標準化範本的目錄，以根據您的需求部署 AWS 帳戶 和服務。
4. 針對控制組態、指令碼和相關資料實作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 透過自動化部署變更
- SUS06-BP01 採用可快速引入永續發展改進的方法

相關文件：

- 使用多個帳戶組織您的 AWS 環境

相關範例：

- 使用 Service Catalog 自動化帳戶建立和資源佈建 AWS Organizations，以及 AWS Lambda
- 使用 AWS Secrets Manager、 和 增強 DevOps 管道 AWS KMS並保護資料 AWS Certificate Manager

相關工具：

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

SEC01-BP07 使用威脅模型識別威脅並排定緩解優先順序

執行威脅模型，以識別和維護 up-to-date工作負載的潛在威脅和相關緩解措施的註冊。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態勢，重新檢視和維護此工作。

未建立此最佳實務時的曝險等級：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。」 – Open Web Application Security Project （ OWASP ） Application Threat Modeling

為何使用威脅模型？

系統本身錯綜複雜，並且隨時間變得更複雜且更具能力，從而實現更大的商業價值及更高的客戶滿意度和參與度。這意味著 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，以及制定緩解措施，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狀態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決策。

何時應執行威脅建模？

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

我們如何執行威脅建模？

有許多不同的方式來執行威脅建模。就跟程式設計語言一樣，各有優缺點，而您應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？

此問題的目的是協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詳細資訊。建立模型或圖表是回答此問題最受歡迎的方法，因為這可協助您將正在建置的項目視覺化，例如使用資料流程圖。寫下關於您的系統的假設和重要詳細資訊也有助於您定義涵蓋的範圍。這可讓對威脅模型做出貢獻的每個人專注於相同的事物，並避免對 out-of-scope主題進行耗時的繞道（包括系統過時版本）。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。威脅是意外或故意的動作或事件，會帶來不必要的影響，並且可能會影響系統安全。如果對可能出錯之處沒有清楚的了解，則無法對症下藥。

   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 模型來識別威脅，例如 來協助您進行腦力激盪STRIDE，它建議評估不同的類別：詐騙、竄改、複寫、資訊揭露、拒絕服務和提升權限。此外，您可能想要透過檢閱現有清單和研究以取得啟發來協助腦力激盪，包括OWASP前 10 HiTrust 名、威脅目錄，以及您組織自己的威脅目錄。

3. 我們要做何處理？

   就跟上一個問題一樣，對於所有可能的緩解措施並沒有標準的清單可循。此步驟的輸入是上一步識別的威脅、動作和改進之處。

   安全與合規是您和 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 和 之間的責任平衡， AWS 協助您將威脅建模練習範圍擴展到您控制下的緩解措施，這些緩解措施通常是 AWS 服務組態選項和您自己的系統特定緩解措施的組合。

   對於共同責任 AWS 的部分，您會發現 AWS 服務在許多合規計畫的範圍內。這些程式可協助您了解位於 的強大控制項 AWS ，以維護雲端的安全性和合規性。這些程式的稽核報告可供 AWS 客戶從下載AWS Artifact。

   無論您使用哪些 AWS 服務，一律存在客戶責任的要素，且與您的責任相符的緩解措施應包含在威脅模型中。對於 AWS 服務本身的安全控制緩解措施，您想要考慮跨網域實作安全控制，包括身分和存取管理（身分驗證和授權）、資料保護（靜態和傳輸中）、基礎設施安全、記錄和監控等網域。每個 AWS 服務的文件都有一個專用的安全章節，提供可視為緩解措施的安全控制指引。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這些威脅的控制。這些控制可以是輸入驗證、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？

目標是讓您的團隊和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、教導和審查的組合。若要更加深入並實作，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，如果您正在尋找如何將威脅模型整合至組織應用程式開發生命週期的指引，請參閱 AWS 安全部落格上的如何處理威脅模型文章。

威脅編寫器

為了協助您和引導您執行威脅模型，請考慮使用 Threat Composer 工具，該工具旨在減少 time-to-value威脅模型建構時的情況。此工具可協助您執行下列操作：

- 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，以讓您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
- SEC01-BP05 減少安全管理範圍
- SEC01-BP08 定期評估和實作新的安全服務和功能

相關文件：

- 如何處理威脅模型（AWS 安全部落格）
- NIST：以資料為中心系統威脅模型的指南

相關影片：

- AWS Summit ANZ 2021 – 如何處理威脅模型
- AWS Summit ANZ 2022 – 擴展安全性 – 最佳化以實現快速且安全的交付

相關訓練：

- 為建置者正確建立威脅模型 – AWS 技能建置器虛擬自定進度訓練
- 為建置者正確建立威脅模型 – AWS Workshop

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

評估和實作來自 和 AWS AWS 合作夥伴的安全服務和功能，協助您發展工作負載的安全狀態。

預期結果：您有標準實務，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的設計有何影響。

常見的反模式：

- 您不會訂閱 AWS 部落格和RSS摘要，以快速了解相關的新功能和服務
- 您仰賴第二手來源得知安全服務和功能的最新消息和更新
- 您不鼓勵組織中 AWS 的使用者隨時掌握最新的更新

建立此最佳實務的優勢：如果您能隨時掌握新的安全服務和功能，就可以在雲端環境和工作負載中實作控制方面做出明智的決策。這些來源有助於提高對不斷演變的安全態勢的意識，以及如何利用 AWS 服務來防範新的和新興的威脅。

未建立此最佳實務時的曝險等級：低

實作指引

AWS 透過幾個管道通知客戶新的安全服務和功能：

- AWS 新功能
- 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，以取得完整的每日更新摘要。有些安全服務，例如 Amazon GuardDuty 和 AWS Security Hub，提供自己的SNS主題，以隨時了解這些特定服務的新標準、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，於會中宣佈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大會公告，您可以在 上的AWS 事件頻道線上檢視深入探討教育分組會議 YouTube。

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 Enterprise Support，您會收到來自 Technical Account Manager （ TAM ） 的每週更新，並可以與其安排定期檢閱會議。

實作步驟

1. 使用您最愛的RSS閱讀器訂閱各種部落格和公告，或訂閱每日功能更新SNS主題。
2. 評估要參加 AWS 的事件，以第一手了解新功能和服務。
3. 與 AWS 帳戶 您的團隊設定會議，以解決任何有關更新安全服務和功能的問題。
4. 考慮訂閱 Enterprise Support，定期諮詢 Technical Account Manager （ TAM ）。

資源

相關的最佳實務：

- PERF01-BP01 了解和了解可用的雲端服務和功能
- COST01-BP07 up-to-date保留新的服務版本

身分與存取管理

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？

接近操作安全 AWS 工作負載時，您必須管理兩種類型的身分。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。

人類身分：您的管理員、開發人員、運算子和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些是您組織的成員，或是與您合作的外部使用者，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 資源互動的外部使用者。

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對 AWS 服務提出請求，例如讀取資料。這些身分包括在 Amazon EC2執行個體或 AWS Lambda 函數等 AWS 環境中執行的機器。您也可以為需要存取權的外部各方管理其機器身分。此外，您可能還有 以外的機器 AWS ，需要存取您的 AWS 環境。

最佳實務

- SEC02-BP01 使用強大的登入機制
- SEC02-BP02 使用臨時憑證
- SEC02-BP03 安全地存放和使用秘密
- SEC02-BP04 依賴集中式身分提供者
- SEC02-BP05 定期稽核和輪換憑證
- SEC02-BP06 使用使用者群組和屬性

SEC02-BP01 使用強大的登入機制

登入（使用登入憑證進行身分驗證）可能會在不使用多重要素身分驗證（MFA）等機制時帶來風險，特別是在不慎公開或容易猜到登入憑證的情況下。使用強大的登入機制，透過要求 MFA和強大的密碼政策來降低這些風險。

預期結果： AWS 使用 AWS Identity and Access Management（IAM） 使用者、AWS 帳戶 根使用者、 AWS IAM Identity Center（單一登入的後繼者 AWS ）和第三方身分提供者的強大登入機制，降低中意外存取憑證的風險。這表示需要 MFA、強制執行強大的密碼政策，以及偵測異常的登入行為。

常見的反模式：

- 未針對身分強制執行強式密碼政策，包括複雜的密碼和 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曝險等級：高

實作指引

人類身分登入 AWS的方法有很多。驗證至 時， AWS 最佳實務是使用聯合 （直接聯合或使用 AWS IAM Identity Center ） 依賴集中式身分提供者 AWS。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 時 AWS 帳戶，您會從 AWS 帳戶 根使用者開始。您應僅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 後立即MFA為帳戶根使用者開啟 ， AWS 帳戶 並使用 AWS 最佳實務指南 來保護根使用者。

如果您在 中建立使用者 AWS IAM Identity Center，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 使用者集區並保護該服務中的登入程序，或使用 Amazon Cognito 使用者集區支援的其中一個身分提供者。

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

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該由您的公司政策設定或使用 NIST800-63 等標準。

- 需要 MFA。這是人類身分和工作負載IAM所需的最佳實務MFA。開啟 MFA 提供額外的安全層，要求使用者提供登入憑證和一次性密碼 （ OTP ） 或從硬體裝置進行密碼編譯驗證和產生的字串。
- 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：

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

Amazon Cognito 使用者集區目錄建議：

- 設定密碼強度設定。
- 使用者需要 MFA 。
- 針對適應性身分驗證 (這可封鎖可疑登入) 等功能使用 Amazon Cognito 使用者集區進階安全設定。

IAM 使用者建議：

- 理想情況下，您使用 IAM Identity Center 或直接聯合。不過，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政策以強制執行MFA登入，讓使用者可以管理自己的密碼和MFA裝置。

資源

相關的最佳實務：

- SEC02-BP03 安全地存放和使用秘密
- SEC02-BP04 依賴集中式身分提供者
- SEC03-BP08 在組織內安全地共用資源

相關文件：

- AWS IAM Identity Center 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 大規模管理使用者許可 AWS IAM Identity Center
- 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，最好是使用暫時憑證，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，請盡可能使用暫時憑證。長期憑證會產生許多風險，例如，它們可以在程式碼中上傳到公有 GitHub 儲存庫。透過使用暫時憑證，您可大幅降低憑證遭到入侵的可能性。

常見的反模式：

- 開發人員使用使用者的長期存取金鑰IAM，而不是CLI使用聯合從 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暫時憑證時，卻使用長期存取金鑰。

未建立此最佳實務時的曝險等級：高

實作指引

針對所有 AWS API 和 CLI請求使用臨時安全憑證，而非長期憑證。API 和 對 AWS 服務的CLI請求，幾乎在每個案例中都必須使用AWS 存取金鑰 簽署。您可以使用暫時或長期憑證簽署這些請求。您應該使用長期憑證，也稱為長期存取金鑰，唯一的時機是您使用IAM使用者或AWS 帳戶 根使用者 。當您透過其他方法聯合 AWS 或擔任IAM角色時，會產生臨時憑證。即使您 AWS Management Console 使用登入憑證存取 ，也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，並能夠使用暫時憑證完成幾乎所有任務。

避免使用長期憑證來支持臨時憑證，應該與減少IAM使用者使用的策略並行，以支持聯合和IAM角色。雖然IAM使用者過去曾用於人類和機器身分，但現在建議不要使用這些身分，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：

- 您應該依賴集中式身分提供者，並要求人類使用者與身分提供者使用聯合，以 AWS 使用臨時憑證存取 。您可以直接聯合至各個 AWS 帳戶 或使用 AWS IAM Identity Center 和自選的身分提供者，為您的使用者進行聯合。除了消除長期憑證之外，聯合還提供了與使用IAM使用者相比的一些優勢。

您的使用者也可以從命令列請求臨時憑證以進行直接聯合或使用 IAM Identity Center 。這表示很少使用需要IAM使用者或使用者長期憑證的案例。

- 授予第三方，例如軟體即服務 SaaS ）供應商存取 中的資源時 AWS 帳戶，您可以使用跨帳戶角色和資源型政策 。
- 如果您需要授予應用程式給取用者或客戶存取您的 AWS 資源，您可以使用 Amazon Cognito 身分集區或 Amazon Cognito 使用者集區來提供臨時憑證。憑證的許可是透過IAM角色設定。您也可以為未驗證的訪客使用者定義具有有限許可的獨立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IAM具有角色的臨時憑證來存取 AWS。

- 對於 Amazon Elastic Compute Cloud （ Amazon EC2 ），您可以使用 Amazon 的 角色EC2。

- AWS Lambda 可讓您設定 Lambda 執行角色，以授予服務使用臨時憑證執行動作的許可。 AWS 服務有許多其他類似的模型 AWS ，可使用 IAM 角色授予臨時憑證。
- 對於 IoT 裝置，您可以使用 AWS IoT Core 憑證提供者來請求暫時憑證。
- 對於內部部署系統，或在需要存取 AWS 資源的 AWS 之外執行的系統，您可以使用 IAM Roles Anywhere 。

有些情況無法使用暫時憑證，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。某些可能需要長期憑證的範例包括 WordPress 外掛程式和第三方 AWS 用戶端。在您必須使用長期憑證，或用於資料庫登入等 AWS 存取金鑰以外的憑證的情況下，您可以使用專為處理秘密管理而設計的服務，例如 AWS Secrets Manager。Secrets Manager 讓您可輕鬆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 安全地存放和使用秘密
- SEC02-BP04 依賴集中式身分提供者
- SEC03-BP08 在組織內安全地共用資源

相關文件：

- 暫時安全憑證
- AWS 登入資料
- IAM 安全最佳實務
- IAM 角色
- IAM 身分中心
- 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 大規模管理使用者許可 AWS IAM Identity Center
- 在每一層都能掌握身分

SEC02-BP03 安全地存放和使用秘密

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。這是使用秘密存取憑證來完成的，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這些憑證遭到入侵的可能性。

預期成果：實作安全管理應用程式憑證的機制，以達成下列目標：

- 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 封鎖對憑證的存取 API (將其視為憑證販賣機) 。
- 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

過去，用於驗證資料庫、第三方 、APIs權杖和其他秘密的憑證可能內嵌在原始程式碼或環境檔案中。AWS 提供數種機制，以安全地存放這些憑證、自動輪換憑證，以及稽核其使用量。

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儲存、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，可能有機會以暫時或短期憑證取代長期憑證。例如，考慮使用IAM角色將長期憑證取代為臨時憑證，而不是硬式編碼 AWS 秘密存取金鑰。

部分長期存留的機密可能無法移除或取代。您可以將這些機密儲存在 AWS Secrets Manager 之類的服務中，進行集中儲存、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：

憑證類型	描述	建議策略
IAM 存取金鑰	AWS IAM 用於在工作負載中擔任IAM角色的存取和秘密金鑰	取代：改為使用指派給運算執行個體 (例如 Amazon EC2 或 AWS Lambda) IAM的角色。如需與需要存取您 中資源的第三方互操作性 AWS 帳戶，請詢問他們是否支援AWS

憑證類型	描述	建議策略
		跨帳戶存取 。對於行動應用程式，請考慮透過 Amazon Cognito 身分集區 (聯合身分) 使用暫時憑證。對於在 之外執行的工作負載 AWS，請考慮 IAM Roles Anywhere 或 AWS Systems Manager Hybrid Activations。
SSH 金鑰	用於手動或作為自動化程序一部分登入 Linux EC2執行個體的安全 Shell 私有金鑰	取代：使用 AWS Systems Manager 或 EC2 Instance Connect 提供使用IAM角色對EC2執行個體的程式設計和人工存取。
應用程式和資料庫憑證	密碼 – 純文字字串	輪換：將憑證儲存在 AWS Secrets Manager 中並建立自動輪換 (如果可能)。
Amazon RDS和 Aurora Admin 資料庫憑證	密碼 – 純文字字串	取代：使用 Secrets Manager 與 Amazon 或 Amazon Aurora 的整合RDS。 https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/rds-secrets-manager.html此外，某些RDS資料庫類型可以在某些使用案例中使用IAM角色而非密碼 （ 如需更多詳細資訊，請參閱IAM資料庫身分驗證 ）。
OAuth 權杖	私密字符 – 純文字字串	輪換：將字符儲存在 AWS Secrets Manager 中並設定自動輪換。

憑證類型	描述	建議策略
API 權杖和金鑰	私密字符 – 純文字字串	輪換：儲存 AWS Secrets Manager 中，並在可能的情況下建立自動輪換。

常見的反模式是在原始程式碼、組態檔案或行動應用程式內內嵌IAM存取金鑰。當需要IAM存取金鑰才能與服務通訊時 AWS ，請使用暫時（短期）安全憑證。這些短期憑證可以透過執行個體IAM的角色EC2、Lambda 函數的執行角色、行動使用者存取的 Cognito IAM角色，以及 IoT 裝置的 IoT Core 政策提供。 IoT 與第三方互動時，偏好將存取權委派給具有帳戶資源必要存取權IAM的角色，而不是設定IAM使用者，並將該使用者的秘密存取金鑰傳送給第三方。

在許多情況下，工作負載需要儲存與其他 服務和資源互操作所需的秘密。 AWS Secrets Manager 旨在安全地管理這些憑證，以及權API杖、密碼和其他憑證的儲存、使用和輪換。

AWS Secrets Manager 提供五種金鑰功能，以確保敏感憑證的安全儲存和處理：靜態加密、傳輸中的加密、全面稽核、精細存取控制和可擴展憑證輪換。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等自動化工具識別包含硬式編碼憑證的程式碼路徑。
   a. 使用 Amazon CodeGuru 掃描程式碼儲存庫。檢閱完成後，在 `Type=Secrets` 中篩選 CodeGuru 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 秘密金鑰，請將它們取代為與必要資源相關聯的IAM角色。如果部分工作負載在外部，AWS 但需要IAM憑證才能存取 AWS 資源，請考慮 IAM Roles Anywhere 或 AWS Systems Manager Hybrid Activations。
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期擷取第三方機密。
   a. CodeGuru 主控台可以使用探索的憑證，在 Secrets Manager 中自動建立秘密。
   b. 將 Secrets Manager 的機密擷取整合至您的應用程式程式碼中。
      i. 無伺服器 Lambda 函數可以使用與語言無關的 Lambda 延伸。
      ii. 對於EC2執行個體或容器， AWS 提供範例用戶端程式碼，用於以多種熱門程式設計語言從 Secrets Manager 擷取秘密。

4. 定期審查您的程式碼庫並重新掃描，以確認程式碼中未加入新的機密。
    a. 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。將讀取、寫入和修改憑證的存取權限制為專門用於此目的IAM的角色，並且只提供存取以擔任操作使用者的一小部分的角色。

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：

- 入門 AWS Secrets Manager
- 身分提供者與聯合
- Amazon CodeGuru Introduces Secrets 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 整合 AWS Secrets Manager

相關影片：

- 大規模管理、擷取和輪換機密的最佳實務
- 使用 Amazon Secrets 偵測器尋找硬編碼 CodeGuru 秘密
- 使用 保護混合工作負載的秘密 AWS Secrets Manager

相關研討會：

- 在 中存放、擷取和管理敏感憑證 AWS Secrets Manager
- AWS Systems Manager 混合啟動

SEC02-BP04 依賴集中式身分提供者

人力身分 (員工和承包商) 可仰賴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。

預期結果：您擁有集中式身分提供者，可讓您集中管理人力資源使用者、身分驗證政策（例如需要多重要素身分驗證（MFA）），以及系統和應用程式的授權（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，如果有人離開您的組織，您可以自動撤銷對聯合應用程式和系統的存取權（包括 AWS）。您已在身分提供者中啟用詳細稽核日誌記錄，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風險暴露等級：高

實作指引

員工使用者存取 AWS的指引

組織中的員工和承包商這類人力使用者可能需要 AWS 使用 AWS Management Console 或 AWS Command Line Interface（AWS CLI）來執行其工作職能。您可以透過從您的集中式身分提供者聯合到 AWS 兩個層級，將 AWS 存取權授予您的人力資源使用者：將聯合導向至組織中的每個帳戶，AWS 帳戶 或聯合至AWS 組織中的多個帳戶。

- 若要將人力資源使用者直接與每個 聯合 AWS 帳戶，您可以使用集中身分提供者在該AWS Identity and Access Management帳戶中聯合到 。的彈性IAM可讓您為每個 啟用單獨的 SAML 2.0 或 Open

ID Connect （OIDC） Identity Provider， AWS 帳戶 並使用聯合使用者屬性進行存取控制。您的人力資源使用者將使用其 Web 瀏覽器，提供其憑證（例如密碼和MFA權杖代碼）來登入身分提供者。身分提供者會向其瀏覽器發出SAML宣告，並提交至 AWS Management Console 登入，URL以允許使用者AWS Management Console 透過擔任IAM角色 進行單一登入。您的使用者也可以透過使用身分提供者的SAML聲明來擔任IAM角色，取得臨時 AWS API憑證，以便在 AWS CLI或AWS SDKs來自 AWS STS 使用。

- 若要將人力資源使用者與 AWS 組織中的多個帳戶聯合，您可以使用 AWS IAM Identity Center 來集中管理人力資源使用者對 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 提供預設的身分來源目錄，可用來管理使用者和群組。或者，您可以使用 2.0 SAML 連接至外部身分提供者，並使用 自動佈建使用者和群組，或使用 連接至 Microsoft AD Directory，以選擇外部身分來源。 https://docs.aws.amazon.com/singlesignon/latest/userguide/provision-automatically.html SCIM https://docs.aws.amazon.com/singlesignon/latest/userguide/manage-your-identity-source-ad.html AWS Directory Service設定身分來源後，您可以透過在許可集中 AWS 帳戶 定義最低權限政策，將存取權指派給 使用者和群組。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。您的使用者可以將 AWS CLI v2 設定為與 Identity Center 進行身分驗證，並取得憑證來執行 AWS CLI 命令。Identity Center 也允許在存取 Amazon SageMaker Studio 和 AWS IoT Sitewise Monitor 入口網站 等 AWS 應用程式時進行單一登入。

遵循上述指引之後，您的人力資源使用者在管理 上的工作負載時，不再需要將IAM使用者和群組用於一般操作 AWS。反之，您的使用者和群組是在 外部進行管理， AWS 使用者能夠以聯合身分 的形式存取 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。您應該識別並移除不再需要的IAM群組、IAM使用者和長期使用者憑證（密碼和存取金鑰） AWS 帳戶。您可以使用憑證報告找到未使用的IAM憑證、刪除對應的IAM使用者，以及刪除IAM群組。您可以將 Service Control 政策（SCP） 套用至您的組織，以協助防止建立新的IAM使用者和群組，強制執行存取 AWS 是透過聯合身分。

應用程式使用者的指引

您可以使用 Amazon Cognito 作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 提供了可擴展至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新增使用者身分驗證和存取控制。Amazon Cognito 以開放式身分標準為基礎，例如 SAML和 Open ID Connect （OIDC），支援各種合規法規，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，聯合您的人力資源使用者 AWS 使用集中式身分提供者：
  - 使用 IAM Identity Center 透過與您的身分提供者聯合，啟用 AWS 帳戶 AWS 組織中的多個單一登入。
  - 使用 IAM將您的身分提供者直接連接到每個 AWS 帳戶，啟用聯合精細存取。
- 識別並移除被聯合身分取代IAM的使用者和群組。

應用程式使用者的步驟

- 使用 Amazon Cognito 作為應用程式的集中式身分提供者。
- 使用 OpenID Connect 和 將自訂應用程式與 Amazon Cognito 整合OAuth。您可以使用 Amplify 程式庫來開發自訂應用程式，該程式庫提供簡單的介面來與各種 AWS 服務整合，例如 Amazon Cognito 進行身分驗證。

資源

相關 Well-Architected 的最佳實務：

- SEC02-BP06 使用使用者群組和屬性
- SEC03-BP02 授予最低權限存取
- SEC03-BP06 根據生命週期管理存取權

相關文件：

- 中的身分聯合 AWS
- IAM 的安全最佳實務
- AWS Identity and Access Management 最佳實務
- Identity IAM Center 委派管理入門
- 如何在 IAM Identity Center 中使用客戶受管政策來處理進階使用案例
- 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） 深入探討
- AWS re：Invent 2022 - 透過 IAM Identity Center 簡化現有的人力資源存取
- AWS re：Invent 2018：在蛋糕的每一層掌握身分

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 安全職能合作夥伴：身分和存取管理
- AWS IAM Identity Center

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：實作憑證輪換以協助降低與使用長期憑證關聯的風險。定期稽核和修復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曝險等級：高

實作指引

當您無法依賴臨時憑證且需要長期憑證時，請稽核憑證，以確認多重要素身分驗證 （MFA） 等定義的控制項已強制執行、定期輪換，並具有適當的存取層級。

定期驗證 (最好是透過自動化工具) 是確認強制執行正確的控制項的必要項目。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暫時憑證。當您從 AWS Identity and Access Management （IAM） 使用者移至集中身分時，您可以產生憑證報告來稽核您的使用者。

我們也建議您在身分提供者MFA中強制執行和監控 。您可以設定 AWS Config 規則，或使用AWS Security Hub 安全標準 來監控使用者是否已設定 MFA。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，需要頻繁稽核和輪換存取金鑰。

實作步驟

- 定期稽核憑證：稽核身分提供者中設定的身分，並IAM協助驗證只有授權身分可以存取您的工作負載。這些身分可以包括但不限於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。制定程序以定期稽核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 設定環境中呼叫的特定API呼叫的警示。 AWS Amazon GuardDuty 也可以提醒您非預期的活動 ，這可能表示過度允許存取或非預期存取IAM憑證。
- 定期輪換憑證：當您無法使用臨時憑證時，請定期輪換長期IAM存取金鑰 （最多每 90 天輪換一次）。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。如需輪換IAM使用者存取金鑰的相關資訊，請參閱輪換存取金鑰 。
- 檢閱IAM許可：為了改善 的安全性 AWS 帳戶，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 可自動化許多IAM任務，例如角色和政策管理。或者， AWS CloudFormation 可以用來自動化IAM資源的部署，包括角色和政策，以減少人為錯誤的機會，因為範本可以經過驗證並控制版本。
- 使用 IAM Roles Anywhere 取代機器身分IAM的使用者：IAMRaters Anywhere 可讓您在傳統上無法使用的區域中使用角色，例如內部部署伺服器。IAM Roles Anywhere 使用受信任的 X.509 憑證來驗證 AWS 和接收臨時憑證。使用 IAM Roles Anywhere 可避免輪換這些憑證的需求，因為長期憑證不再存放在您的內部部署環境中。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 安全地存放和使用秘密

相關文件：

- 入門 AWS Secrets Manager
- 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
- 暫時安全憑證
- 取得 的憑證報告 AWS 帳戶

相關影片：

- 大規模管理、擷取和輪換機密的最佳實務
- 使用 大規模管理使用者許可 AWS IAM Identity Center
- 在每一層都能掌握身分

相關範例：

- Well-Architected Lab - 自動化IAM使用者清除
- Well-Architected Lab - IAM群組和角色的自動部署

SEC02-BP06 使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曝險等級：中

實作指引

AWS 許可是在名為政策的文件中定義，這些政策與主體相關聯，例如使用者、群組、角色或資源。 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群組可能已連接政策，用於在 CloudFront 開發帳戶中設定 Amazon 等服務。AutomationDeveloper群組可能具有與該WebAppDeveloper群組相同的一些 CloudFront 許可。您可在個別政策中擷取這些許可，並讓這些許可同時與這兩個群組關聯，而不是讓這兩種職務的使用者屬於 CloudFrontAccess 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。 如果使用此技術，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。 您參考許可政策中的屬性的方式是以其來源為基礎，無論是定義為聯合通訊協定的一部分（例如 SAML、 或 SCIM）OIDC、自訂SAML宣告，還是在 IAM Identity Center 中設定。

實作步驟

1. 確定您將定義群組和屬性的位置。
   a. 遵循 中的指引SEC02-BP04 依賴集中式身分提供者，您可以判斷是否需要在身分提供者、IAM身分中心內，還是在特定帳戶中使用IAM使用者群組來定義群組和屬性。
2. 定義群組。
   a. 根據職務和所需的存取範圍確定您的群組。
   b. 如果在 IAM Identity Center 中定義，請建立群組並使用許可集建立所需的存取層級。
   c. 如果在外部身分提供者中定義，請判斷提供者是否支援SCIM通訊協定，並考慮在IAM身分中心中啟用自動佈建。 此功能會同步提供者和 IAM Identity Center 之間群組的建立、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 通訊協定預設會提供特定屬性。 其他屬性可以使用屬性https://aws.amazon.com/SAML/Attributes/PrincipalTag名稱，使用SAML宣告來定義和傳遞。
   b. 如果在 IAM Identity Center 中定義，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. 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 依賴集中式身分提供者
- SEC03-BP02 授予最低權限存取
- 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 AWS？
- ABAC 在IAM身分中心

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 和工作負載的人員 AWS 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 定義存取要求
- SEC03-BP02 授予最低權限存取
- 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
- SEC03-BP06 根據生命週期管理存取權
- SEC03-BP07 分析公有和跨帳戶存取
- SEC03-BP08 在組織內安全地共用資源
- SEC03-BP09 與第三方安全地共用資源

SEC03-BP01 定義存取要求

管理員、終端使用者或其他元件都需要存取工作負載的每個元件或資源。請明確定義應該有權存取每個元件的人員和機器，選擇適當的身分類型及驗證和授權方法。

常見的反模式：

- 將機密硬式編碼或儲存在應用程式中。
- 為每名使用者授予自訂許可。
- 使用長期憑證。

未建立此最佳實務時的曝險等級：高

實作指引

管理員、終端使用者或其他元件都需要存取工作負載的每個元件或資源。請明確定義應該有權存取每個元件的人員和機器，選擇適當的身分類型及驗證和授權方法。

應使用聯合存取或集中式身分提供者提供組織內 AWS 帳戶 的定期存取。https://aws.amazon.com/identity/federation/您也應該集中身分管理，並確保已建立實務來整合對員工存取生命週期的 AWS 存取。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。建議使用以最低權限存取模型建置IAM的角色。AWS 受管政策提供預先定義的IAM政策，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 ，可在無法使用IAM角色時，協助安全地將秘密與應用程式或工作負載分離。在 Secrets Manager 中，您可以為憑證建立自動輪換。您可以使用建立參數時指定的唯一名稱，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。

您可以使用 AWS Identity and Access Management Roles Anywhere 在 中為在 之外執行的工作負載取得臨時安全憑證IAM AWS。您的工作負載可以使用與 AWS 應用程式搭配使用的相同IAM政策和IAM角色來存取 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，請使用存取金鑰上次使用的資訊來輪換和移除存取金鑰。

如果使用者想要與 AWS 外部互動，則需要程式設計存取權 AWS Management Console。授予程式設計存取權的方式取決於存取 的使用者類型 AWS。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 （在 IAM Identity Center 中管理的使用者）	使用暫時憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 • 對於 AWS CLI，請參閱 使用者指南 中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 AWS Command Line Interface • 如需 AWS SDKs、工具和 AWS APIs，請參閱 AWS SDKs和 工具參考指南 中的IAM身分中心身分驗證。
IAM	使用暫時憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請遵循 IAM 使用者指南 中的將臨時憑證與 AWS 資源搭配使用中的指示。
IAM	(不建議使用) 使用長期憑證簽署對 AWS CLI AWS SDKs、 或 的程式設計請求 AWS APIs。	請依照您要使用的介面所提供的指示操作。 • 對於 AWS CLI，請參閱 AWS Command Line Interface 使用者指南 中的使用IAM使用者憑證進行驗證。 • 如需 AWS SDKs 和 工具，請參閱 AWS SDKs和 工具參考指南 中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱 IAM 使用者指南 中的管理IAM使用者的存取金鑰。

資源

相關文件：

- 屬性型存取控制（ABAC）
- AWS IAM Identity Center
- IAM Roles Anywhere
- AWS IAM Identity Center 的受管政策
- AWS IAM 政策條件
- IAM 使用案例
- 移除不必要的憑證
- 使用 政策
- 如何根據 AWS 帳戶、OU 或組織控制對 AWS 資源的存取
- 使用增強型搜尋輕鬆識別、安排和管理秘密 AWS Secrets Manager

相關影片：

- 在 60 分鐘內成為IAM政策主檔
- 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

最佳實務是僅授予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，而無需變更基礎存取政策。

預期成果：使用者應僅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。管理員特權僅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。

- 使用根使用者進行 day-to-day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曝險等級：高

實作指引

最低權限原則指出，僅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 使用者和角色預設沒有許可。根使用者預設擁有完整存取權，應受到嚴格監控，並僅用於需要根存取權的任務。

IAM 政策用於明確授予IAM角色或特定資源的許可。例如，身分型政策可以連接到IAM群組，而 S3 儲存貯體可以由資源型政策控制。

建立IAM政策時，您可以指定 AWS 允許或拒絕存取所需的服務動作、資源和條件。 AWS 支援各種條件，協助您縮小存取範圍。例如，如果請求者不是您 AWS 組織的一部分，您可以使用`PrincipalOrgID`條件索引鍵 拒絕動作。

您也可以使用 `CalledVia` 條件金鑰來控制 AWS 服務代表您提出的請求，例如 AWS CloudFormation 建立 AWS Lambda 函數。您應該分層不同的政策類型，以建立 defense-in-depth和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其建置的系統建立自己的IAM政策，但也必須套用許可界限，以限制系統可接收的最大許可。

實作步驟

- 實作最低權限政策 ：將IAM具有最低權限的存取政策指派給群組和角色，以反映您定義的使用者角色或函數。
  - 根據API用量 的基本政策：判斷所需許可的一種方法是檢閱 AWS CloudTrail 日誌。此檢閱可讓您根據使用者在 內實際執行的動作建立自訂許可 AWS。IAM Access Analyzer 可以根據活動 自動產生IAM政策。 https://aws.amazon.com/blogs/security/delegate-permission-management-to-developers-using-iam-permissions-boundaries/您可以使用組織或帳戶層級的 IAM Access Advisor來追蹤特定政策 的上次存取資訊。
- 考慮使用適用於各工作職能的AWS 受管政策。開始建立精細的許可政策時，可能很難知道要從何處開始。 AWS 具有常見任務角色的受管政策，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。
- 考慮許可界限：許可界限是使用受管政策的功能，可設定身分型政策可授予IAM實體的最大許可。實體的許可界限可讓其只執行由身分類型政策和其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 的服務控制政策 AWS Organizations。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。另請考慮使用 AWS Control Tower，其提供豐富 的規範性受管控制 AWS Organizations。您也可以在 Control Tower 中定義自己的控制。
- 為您的組織建立使用者生命週期政策：使用者生命週期政策定義使用者加入 時要執行的任務 AWS、變更任務角色或範圍，或不再需要存取 AWS。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。
- 建立定期排程來檢閱許可並移除任何不需要的許可：您應該定期檢閱使用者存取權，以確認使用者沒有過度允許存取。 AWS Config和 IAM Access Analyzer 可以在稽核使用者許可時提供協助。
- 建立任務角色矩陣：任務角色矩陣可視覺化您的 AWS 足跡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
- 撰寫最低權限IAM政策的技術
- IAM Access Analyzer 可根據存取活動產生政策，讓您更輕鬆地實作最低權限許可 IAM https://aws.amazon.com/blogs/security/iam-access-analyzer-makes-it-easier-to-implement-least-privilege-permissions-by-generating-iam-policies-based-on-access-activity/
- 使用許可界限將IAM許可管理委派給開發人員
- 使用上次存取的資訊精簡許可

- IAM 政策類型和使用時機
- 使用IAM政策模擬器測試IAM政策
- 中的護欄 AWS Control Tower
- 零信任架構： AWS 觀點
- 如何使用 實作最低權限原則 CloudFormation StackSets
- 屬性型存取控制 （ ABAC ）
- 檢視使用者活動以縮小政策範圍
- 檢視角色存取
- 使用標記來組織您的環境並推動問責制
- AWS 標記策略
- 標記 AWS 資源

相關影片：

- 下一代許可管理
- 零信任：一個 AWS 觀點

相關範例：

- 實驗室：委派角色建立的IAM許可界限
- 實驗室： 的IAM標籤型存取控制 EC2

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，在正常情況下，您的人力資源使用者會使用集中式身分提供者（ SEC02-BP04 ）來管理其工作負載，並聯合至雲端。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，則發佈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提醒。

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，並發出提醒。

未建立此最佳實務時的風險暴露等級：中

實作指引

本節提供針對部署在 上的工作負載建立緊急存取程序的指南 AWS，從適用於所有失敗模式的通用指南開始，然後根據失敗模式類型提供特定指引。

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。例如，失敗模式 2 的程序假設身分提供者可用，但 上的組態 AWS 已修改或已過期。
- 預先建立緊急存取程序（SEC10-BP05）所需的資源。例如， AWS 帳戶 使用IAM使用者和角色，以及所有工作負載帳戶中的跨帳戶IAM角色預先建立緊急存取。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，您不依賴 AWS 緊急情況下可能無法使用的控制平面 APIs（用來建立和修改 AWS 資源）。此外，透過預先建立IAM資源，您不需要考慮最終一致性可能造成的延遲。
- 將緊急存取程序納入事件管理計畫（SEC10-BP02）的一部分。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提醒。例如，您應該監控緊急存取 AWS 帳戶中的活動並發出提醒，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作為事件回應模擬（SEC10-BP07）和災難復原測試（REL13-BP03）的一部分進行測試。

失敗模式 1：用於與 聯合的身分提供者 AWS 無法使用

如 SEC02-BP04 倚賴集中式身分提供者 所述，我們建議依賴集中式身分提供者來聯合您的人力資源使用者，以授予對 的存取權 AWS 帳戶。您可以使用 IAM Identity Center 聯合到 AWS 組織中 AWS 帳戶的多個，也可以使用 聯合到個人 AWS 帳戶 IAM。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，您可以為一小群管理員提供緊急存取程序，以 AWS 帳戶 執行關鍵任務，這些任務

無法等到集中式身分提供者恢復線上狀態。例如，您的身分提供者無法使用 4 小時，在此期間，您需要修改生產帳戶中 Amazon EC2 Auto Scaling 群組的上限，以處理客戶流量的意外尖峰。您的緊急管理員應遵循緊急存取程序，以取得特定生產的存取權， AWS 帳戶 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 ，僅用於緊急存取，並具有支援緊急存取程序 AWS 的資源 （ 例如IAM角色和IAM使用者 ）。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提醒 (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，其許可可在需要緊急存取 AWS 帳戶 的 中擔任跨帳戶角色。這些IAM角色是預先建立並設定信任政策，可信任緊急帳戶IAM的角色。

緊急存取程序可以使用下列其中一種方法：

- 您可以在具有關聯高強度密碼和MFA字符的緊急存取帳戶中，為緊急管理員預先建立一組IAM使用者。這些IAM使用者具有擔任IAM角色的許可，然後允許跨帳戶存取需要緊急存取的 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字符代碼登入緊急存取帳戶，切換至緊急帳戶中的緊急存取IAM角色，最後切換至工作負載帳戶中的緊急存取IAM角色，以執行緊急變更動作。此方法的優點是，每個IAM使用者都會指派給一個緊急管理員，而且您可以透過檢閱 CloudTrail 事件來了解登入的使用者。缺點是，您必須使用其關聯的長期密碼和MFA權杖來維護多個IAM使用者。
- 您可以使用緊急存取AWS 帳戶 根使用者登入緊急存取帳戶、擔任緊急存取IAM的角色，以及在工作負載帳戶中擔任跨帳戶角色。我們建議為根使用者設定強式密碼和多個MFA權杖。我們也建議將密碼和MFA權杖存放在安全的企業憑證保存庫中，以強制執行嚴格的身分驗證和授權。您應該保護密碼和MFA字符重設因素：將帳戶的電子郵件地址設定為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。

失敗模式 2：在 上的身分提供者組態 AWS 已修改或已過期

若要允許人力資源使用者聯合 AWS 帳戶，您可以使用外部IAM身分提供者設定 Identity Center 或建立 IAM Identity Provider （ SEC02-BP04 ）。通常，您可以透過匯入身分提供者提供的SAML中繼資料XML文件來設定這些文件。中繼資料XML文件包含對應至私有金鑰的 X.509 憑證，身分提供者用來簽署其SAML聲明。

管理員可能會錯誤地修改或刪除 AWS端的這些組態。在另一個案例中，匯入 的 X.509 憑證 AWS 可能會過期，且XML具有新憑證的新中繼資料尚未匯入 AWS。這兩種情況都可能中斷 AWS 您人力資源使用者的聯合到 ，導致緊急狀況。

在此類緊急事件中，您可以提供身分管理員對 的存取權 AWS ，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、切換至 Identity Center SAML 管理員帳戶中的角色，以及透過從您的身分提供者匯入最新的中繼資料XML文件以重新啟用聯合來更新外部身分提供者組態。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

在極少數情況下，如果 IAM Identity Center 或 AWS 區域 中斷，我們建議您設定組態，以用於提供暫時存取 AWS Management Console。

緊急存取程序會使用身分提供者到IAM緊急帳戶中的直接聯合。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 AWS 帳戶 專用於緊急存取程序的 。預先建立帳戶中所需的IAM資源，例如IAM角色或IAM使用者，以及選用的IAM身分提供者。此外，在工作負載中預先建立跨帳戶IAM角色 AWS 帳戶 ，其信任關係與緊急存取帳戶中的對應IAM角色。您可以使用 AWS CloudFormation StackSets 搭配 AWS Organizations ，在組織中的成員帳戶中建立此類資源。

- 建立 AWS Organizations 服務控制政策 （ SCPs ），以拒絕刪除和修改成員 中的跨帳戶IAM角色 AWS 帳戶。
- CloudTrail 為緊急存取啟用 ， AWS 帳戶 並將追蹤事件傳送至日誌集合 中的中央 S3 儲存貯體 AWS 帳戶。如果您使用 AWS Control Tower 來設定和管理您的 AWS 多帳戶環境，則您使用 AWS Control Tower 或 註冊建立的每個帳戶 AWS Control Tower 都會預設 CloudTrail 啟用，並傳送至專用日誌封存中的 S3 儲存貯體 AWS 帳戶。
- 透過建立符合緊急IAM角色在主控台登入和活動上的 EventBridge 規則，來監控緊急存取帳戶API的活動。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。

失敗模式 1 的其他步驟：用於 聯合的身分提供者 AWS 無法使用，而失敗模式 2：在 上的身分提供者組態 AWS 已修改或已過期

- 根據您選擇的緊急存取機制預先建立資源：

- 使用IAM使用者：使用強式密碼和相關聯的MFA裝置預先建立IAM使用者。
- 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，並將裝置存放在緊急管理員團隊成員可以快速存取的位置。

適用「故障模式 3：Identity Center 中斷」的其他步驟

- 如設定 的緊急存取 AWS Management Console中所述，在緊急存取 中 AWS 帳戶，建立IAM身分提供者以啟用來自身分提供者的直接SAML聯合。
- 在 IdP 中建立緊急操作群組，但不新增任何成員。
- 建立與緊急存取帳戶中的緊急操作群組對應的IAM角色。

資源

相關 Well-Architected 的最佳實務：

- SEC02-BP04 依賴集中式身分提供者
- SEC03-BP02 授予最低權限存取
- SEC10-BP02 制定事件管理計劃
- SEC10-BP07 執行遊戲天數

相關文件：

- 設定 的緊急存取 AWS Management Console
- 啟用 SAML 2.0 聯合使用者存取 AWS Management Console
- 緊急存取

相關影片：

- AWS re：Invent 2022 - 透過 IAM Identity Center 簡化現有的人力資源存取
- AWS re：Inforce 2022 - AWS Identity and Access Management （ IAM ） 深入探討

相關範例：

- AWS 緊急存取角色

- AWS 客戶程序手冊架構
- AWS 事件回應程序手冊範例

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，並建立審查程序以達到最低權限的許可。持續監視人類和機器存取權，並移除不使用的身分和許可。

預期成果：許可政策應遵循最低權限原則。隨著工作職責和角色的定義變得更具體，您需要審查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：

- 預設授予使用者管理員許可。
- 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曝險等級：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，開發人員可以存取廣泛的 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這些服務和服務動作。我們建議對人類和機器身分進行此項評估。機器身分，有時稱為系統或服務帳戶，是 AWS 可存取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，以協助識別未使用的使用者、角色、許可和憑證。 AWS 也可以協助分析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 產生政策可協助您根據主體互動的實際服務和動作建立限制性許可政策。屬性型存取控制（ABAC）可協助簡化許可管理，因為您可以使用其屬性提供許可給使用者，而不是將許可政策直接連接到每個使用者。

實作步驟

- 使用 AWS Identity and Access Management Access Analyzer：IAMAccess Analyzer 有助於識別組織和帳戶中的資源，例如 Amazon Simple Storage Service（Amazon S3）儲存貯體或與外部實體共用IAM的角色。

- 使用 IAM Access Analyzer 政策產生 ：IAMAccess 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。
- 判斷IAM使用者和角色可接受的時間範圍和使用政策：使用上次存取的時間戳記來識別未使用的使用者和角色，並將其移除。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範圍。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這些動作。上一次存取的資訊功能可在 中使用， AWS Management Console 並以程式設計方式可讓您將它們整合到您的基礎設施工作流程和自動化工具中。
- 考慮在 中記錄資料事件 AWS CloudTrail：根據預設， CloudTrail 不會記錄資料事件，例如 Amazon S3 物件層級活動（例如 `GetObject`和 `DeleteObject`） 或 Amazon DynamoDB 資料表活動（例如 `PutItem`和 `DeleteItem`）。考慮對這些事件使用日誌記錄功能，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
- 移除不必要的憑證
- 什麼是 AWS CloudTrail？
- 使用 政策
- DynamoDB 中的日誌記錄和監控
- 針對 Amazon S3 儲存貯體和物件使用 CloudTrail 事件記錄
- 取得 的憑證報告 AWS 帳戶

相關影片：

- 在 60 分鐘內成為IAM政策主檔
- 責任區隔、最低權限、委派和 CI/CD
- AWS re：Inforce 2022 - AWS Identity and Access Management （ IAM ） 深入探討

SEC03-BP05 為您的組織定義許可防護

使用許可防護機制縮小可授予主體的可用許可範圍。許可政策評估鏈包括您的防護機制，可在進行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，另外對暫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 帳戶對環境進行明確的隔離。 服務控制政策 （SCPs） 用於定義整個組織的許可護欄。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，以將許可管理委派給個別工作負載擁有者。

常見的反模式：

- 在AWS 組織 AWS 帳戶 中建立成員，但不使用 SCPs 來限制其根憑證可用的使用和許可。
- 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 的隱含拒絕基礎 AWS IAM來限制許可，信任該政策不會授予不想要的明確許可。
- 在相同的 中執行多個工作負載環境 AWS 帳戶，然後依賴 VPCs、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曝險等級：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。此方法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。這可協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。

建立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，一個帳戶的主體無法存取另一個帳戶中的資源，即使兩個帳戶都位於同一個 AWS 組織或同一組織單位 （OU） 下。 您可以使用 OUs 將要管理的帳戶分組為單一單位。

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。 您可以為此使用服務控制政策 （SCPs），您可以將其套用至 OU 或 帳戶。 SCPs 可以強制執行常見的存取控制，例如限制對特定的存取 AWS 區域、協助防止資源遭到刪除，或停用可能有風險的服務動作。 SCPs 您套用至組織的根目錄只會影響其成員帳戶，不會影響管理帳戶。 SCPs 僅管理組織中的主體。您的 SCPs不會管理組織外部存取您資源的主體。

另一個步驟是使用IAM資源政策，以範圍限制您可以對他們管理的資源採取的可用動作，以及代理主體必須滿足的任何條件。 只要主體是組織的一部分 （使用 PrincipalOrgId條件索引鍵 ），或只允許特定IAM角色的特定動作，這就可能非常廣泛。 您可以採取類似的方法，其中包含IAM角色信任政策中的條

件。 如果資源或角色信任政策明確指定與其管理的角色或資源相同的帳戶中的主體，則該主體不需要授予相同許可的附加IAM政策。 如果委託人位於與資源不同的帳戶中，則委託人確實需要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。 這可能需要他們建立新的IAM角色和許可政策。 您可以擷取允許團隊在許可IAM界限 中授予的最大許可範圍，並將本文件與團隊可用來管理其IAM角色和許可IAM的角色建立關聯。 這種方法可以讓他們完成工作，同時降低擁有IAM管理存取權的風險。

更精細的步驟是實作權限存取管理 （PAM） 和暫時提升的存取管理 （TEAM） 技術。 的一個範例PAM是要求主體在採取特權動作之前執行多重要素驗證。 如需詳細資訊，請參閱設定 MFA受保護的API存取 。TEAM 需要一個解決方案，用於管理允許主體擁有更高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較高存取權之IAM角色的角色信任政策。 另一種方法是，在正常操作下，IAM使用工作階段政策 來縮小角色授予主體的許可範圍，然後在核准的時段內暫時解除此限制。 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，請參閱暫時提升的存取權。

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 SCPs減少可授予您組織成員帳戶中主體的許可集上限。
   a. 我們建議您使用允許清單方法，撰寫拒絕所有動作SCPs的 ，但您允許的動作除外，以及允許執行這些動作的條件。首先定義您要控制的資源，然後將「效果」設定為「拒絕」。使用 NotAction 元素來拒絕您指定的動作以外的所有動作。將此與 NotLike 條件合併，以定義何時允許這些動作，如適用，例如 StringNotLike 和 ArnNotLike。
   b. 參閱服務控制政策範例。
3. 使用IAM資源政策來縮小範圍，並指定資源上允許動作的條件。 使用IAM角色信任政策中的條件來建立擔任角色的限制。
4. 將IAM許可界限指派給工作負載團隊可用來管理自己的工作負載IAM角色和許可IAM的角色。
5. 根據您的需求評估 PAM和 TEAM解決方案。

資源

相關文件：

- 上的資料周邊 AWS
- 使用資料周邊建立許可防護機制
- Policy 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：暫時提升存取管理
- 已驗證的安全合作夥伴解決方案 TEAM

SEC03-BP06 根據生命週期管理存取權

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。

預期成果您可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。您可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。您可以使用自動化將變更傳播到 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。
- 您未移除非作用中或已終止身分的作用中存取權限。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 內管理IAM使用者的生命週期 AWS 帳戶，或透過聯合，從人力資源身分提供者到 AWS IAM身分中心。對於IAM使用者，您可以在 中建立、修改和刪除使用者及其相關聯的許可 AWS 帳戶。對於聯合身分使用者，您可以使用 跨網域IAM身分管理 （SCIM） 通訊協定，從組織的身分提供者同步使用者和群組資訊，藉此使用身分中心來管理其生命週期。

SCIM 是開放標準通訊協定，用於跨不同系統自動佈建和取消佈建使用者身分。透過使用 將您的身分提供者與 IAM Identity Center 整合SCIM，您可以自動同步使用者和群組資訊，協助根據組織的權威身分來源的變更來驗證授予、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任務角色或責任，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，包括授予初始存取權、定期審查和離職的程序。
2. 實作IAM角色、群組和許可界限，以共同管理存取並強制執行允許的存取層級上限。
3. 使用 Identity IAM Center 與聯合身分提供者 （ 例如 Microsoft Active Directory、Okta、Ping Identity ） 整合，做為使用者和群組資訊的權威來源。
4. 使用 SCIM通訊協定，將身分提供者的使用者和群組資訊同步到 IAM Identity Center 的 Identity Store。
5. 在 IAM Identity Center 中建立代表組織中不同任務角色或責任的許可集。為每個許可集定義適當的IAM政策和許可。
6. 實作定期存取權審查、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 依賴集中式身分提供者

相關文件：

- 管理您的身分來源
- 在 IAM Identity Center 中管理身分
- 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 管理暫時提升的存取
- AWS re：Invent 2022 - 使用 IAM Identity Center 簡化現有的人力資源存取
- AWS re：Invent 2022 - 利用IAM政策的強大功能，並在具有 Access Analyzer 的許可中控制

SEC03-BP07 分析公有和跨帳戶存取

持續監控強調公有和跨帳戶存取權的調查結果。減少僅對需要此存取之特定資源的公有存取權和跨帳戶存取權。

預期結果：了解您的哪些 AWS 資源是共享的，以及與誰共享。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曝險等級：低

實作指引

如果您的 帳戶位於 中 AWS Organizations，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。您可以使用資源型政策，例如 Amazon Simple Storage Service （ Amazon S3 ） 儲存貯體政策，或透過允許另一個帳戶中的主體擔任您帳戶中IAM的角色，授予直接跨帳戶存取權。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識別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調查結果，讓您輕鬆分析潛在的各種存取。考慮使用 設定 IAM Access Analyzer AWS Organizations ，以確認您能夠查看所有帳戶。IAM Access Analyzer 也可讓您在部署資源許可之前預覽調查結果。這可讓您驗證政策變更是否僅授予對您資源的預期公有和跨帳戶存取權。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 可以報告設定錯誤的資源，並透過 AWS Config 政策檢查，可以偵測已設定公有存取權的資源。例如 AWS Control Tower和 AWS Security Hub 簡化跨 部署偵測控制項和防護欄的服務 AWS Organizations ，以識別和修復公開暴露的資源。例如， AWS Control Tower 具有受管防護機制，可偵測是否有任何 Amazon EBS快照可由 還原 AWS 帳戶。

實作步驟

- 考慮將 AWS Config 用於 AWS Organizations： AWS Config 可讓您將 內多個帳戶的調查結果彙總 AWS Organizations 到委派的管理員帳戶。這提供全面的檢視，並可讓您AWS Config 規則 跨帳戶部署 ，以偵測可公開存取的資源 。
- Configure AWS Identity and Access Management Access Analyzer IAM Access Analyzer 可協助您識別組織和帳戶中的資源，例如 Amazon S3 儲存貯體或與外部實體共用IAM的角色。
- 使用 中的自動修正 AWS Config 來回應 Amazon S3 儲存貯體的公有存取組態變更：您可以自動開啟 Amazon S3 儲存貯體的區塊公有存取設定。
- 實作監控和提醒以識別 Amazon S3 儲存貯體是否已變為公有：您必須設立監控和提醒以識別何時關閉 Amazon S3 封鎖公開存取，以及 Amazon S3 儲存貯體是否變為公有。此外，如果您使用 AWS Organizations，您可以建立服務控制政策，以防止對 Amazon S3 公有存取政策進行變更。 AWS Trusted Advisor 檢查是否有開放存取許可的 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。此 Trusted Advisor 檢查會檢查明確的儲存貯體許可和相關聯的儲存貯體政策，這些政策可能會覆寫儲存貯體許可。您也可以使用 AWS Config 來監控您的 Amazon S3 儲存貯體以供公開存取。如需詳細資訊，請參閱如何使用 AWS Config 來監控和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和 憑證，例如私有或 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 監控 AWS Trusted Advisor 檢查結果 EventBridge
- 管理組織中所有帳戶的 AWS Config 規則
- AWS Config 而且 AWS Organizations
- 將您的 AMI 公開用於 Amazon EC2

相關影片：

- 保護多帳戶環境的最佳實務
- 深入探索 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對資源的存取，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方案指引，請參閱 SEC03-BP07 分析公有和跨帳戶存取。

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曝險等級：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提醒。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，以及建立程序持續監控並自動提供提醒。

內的跨帳戶共用受許多 AWS 服務 AWS Organizations 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 可以將調查結果從個別帳戶轉移到中央帳戶，您可以在其中檢視所有調查結果。 AWS Backup 可以備份資源並跨帳戶共用。您可以使用 AWS Resource Access Manager（AWS RAM） 來共用其他常用資源，例如VPC子網路和 Transit Gateway 附件 、AWS Network Firewall或 Amazon SageMaker 管道 。

若要限制您的帳戶只共用組織內的資源，請使用服務控制政策 （SCPs） 以防止存取外部主體。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，以協助預防意外存取。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網路存取可信的資源。這些控制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，作為資料周邊的一部分，您可以使用VPC端點政策和`AWS:PrincipalOrgId`條件，以確保存取 Amazon S3 儲存貯體的身分屬於您的組織。請注意， SCPs 不適用於服務連結角色 AWS 或服務主體 。

使用 Amazon S3 時，請關閉 Amazon S3 儲存貯體ACLs的 ，並使用IAM政策來定義存取控制。若要限制從 Amazon 對 Amazon S3 原始伺服器的存取，請從原始伺服器存取身分 （OAI） 遷移至原始伺服器存取控制 （OAC），以支援其他功能，包括使用 進行伺服器端加密AWS Key Management Service。 CloudFront

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方案指引，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 合併到您建立並集中管理的組織。您可以將帳戶分組為組織單位 （OUs），並將不同的政策連接到每個 OU，以協助您滿足預算、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） 服務如何收集和儲存資料，並使用與 Organizations 整合 AWS 之服務的多帳戶管理。

2. AWS Organizations 與 AWS 服務整合。

   當您使用 AWS 服務在組織的成員帳戶中代表您執行任務時， 會在每個成員帳戶中為該服務 AWS Organizations 建立IAM服務連結角色 （SLR）。您應該使用 AWS Management Console、 AWS APIs或 管理受信任的存取 AWS CLI。如需開啟受信任存取的規範性指引，請參閱搭配使用 AWS Organizations 與可與 Organizations 搭配使用的其他 AWS服務和 。 AWS

3. 建立資料周邊。

   AWS 周邊通常表示為 管理的組織 AWS Organizations。除了內部部署網路和系統之外，存取 AWS 資源也是許多人認為我的周邊 AWS。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。

      請遵循在 AWS 白皮書上針對每個授權條件建立周長https://docs.aws.amazon.com/whitepapers/latest/building-a-data-perimeter-on-aws/perimeter-implementation.html中所述的步驟。如需有關保護網路層的方案指引，請參閱保護網路。

   b. 持續監控和提醒。

AWS Identity and Access Management Access Analyzer 可協助您識別組織和帳戶中那些與外部實體共用的資源。您可以將 IAM Access Analyzer 與 AWS Security Hub 整合，以將資源的調查結果從 IAM Access Analyzer 傳送至 Security Hub，以協助分析環境的安全狀態。若要整合，請在每個帳戶中的每個區域中開啟 IAM Access Analyzer 和 Security Hub。您也可以使用 AWS Config 規則 來稽核組態，並使用 AWS Chatbot 與 AWS Security Hub提醒適當對象。然後，您可以使用 AWS Systems Manager Automation 文件來修復不合規的資源。

c. 如需有關持續監控與提醒外部共用的資源的方案指引，請參閱分析公開和跨帳戶存取權。

4. 在 AWS 服務中使用資源共用並相應地限制。

   許多 AWS 服務可讓您與其他 帳戶共用資源，或鎖定另一個 帳戶中的資源，例如 Amazon Machine Images （AMIs） 和 AWS Resource Access Manager （AWS RAM）。限制 `ModifyImageAttribute`API以指定要AMI共用的受信任帳戶。使用 時，請指定 `ram:RequestedAllowsExternalPrincipals`條件 AWS RAM ，以限制與組織的共用，以協助防止來自不受信任身分的存取。如需方案指引和考量，請參閱資源共用和外部目標。

5. 使用 在 帳戶或其他 中安全地 AWS RAM 共用 AWS 帳戶。

   AWS RAM 可協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。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。此方法有助於降低您的營運開銷，同時透過與 Amazon 和 CloudWatch 的整合提供一致性、可見性和可稽核性 AWS CloudTrail，而您在使用跨帳戶存取時不會收到這些項目。

   如果您有先前使用資源型政策共用的資源，您可以使用 `PromoteResourceShareCreatedFromPolicy`API或 等價物，將資源共用提升為完整的 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，若要共用加密快照，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
- SEC03-BP09 與第三方安全地共用資源
- 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
- 如何搭配 使用信任政策 IAM
- 在 上建置資料周邊 AWS
- 如何在授予第三方存取您的 AWS 資源時使用外部 ID
- AWS 您可以搭配 使用 服務 AWS Organizations
- 在 上建立資料周邊 AWS：僅允許受信任身分存取公司資料

相關影片：

- 使用 AWS Resource Access Manager進行精密的存取
- 使用VPC端點保護您的資料周邊
- 在 上建立資料周邊 AWS

相關工具：

- 資料周邊政策範例

SEC03-BP09 與第三方安全地共用資源

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。第三方受管系統的許可管理應遵循使用具有臨時憑證的最低權限原則的 just-in-time存取實務。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。

預期結果：只要憑證有效且有效，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存取金鑰和秘密金鑰。使用 IAM角色和臨時憑證可協助您改善整體安全狀態，減少維護長期憑證的努力，包括這些敏感詳細資訊的管理和操作額外負荷。透過為IAM信任政策中的外部 ID 使用通用唯一識別碼 （ UUID ），並保持連接到您控制之IAM角色IAM的政策，您可以稽核和驗證授予第三方的存取權不會太寬鬆。如需有關分析外部共用的資源的方案指引，請參閱 SEC03-BP07 分析公有和跨帳戶存取。

常見的反模式：

- 在沒有任何條件的情況下使用預設IAM信任政策。
- 使用長期IAM憑證和存取金鑰。

- 重複使用外部 IDs。

未建立此最佳實務時的曝險等級：中

實作指引

您可能想要允許在 外部共用資源， AWS Organizations 或授予第三方存取您帳戶的權限。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。在這些情況下，建立僅具有第三方所需權限的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。

您也可以使用 IAM Roles Anywhere 來管理 AWS 使用 之外應用程式IAM的角色 AWS APIs。

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。保持對其他支援共享之 AWS 服務的認識。例如， AWS Well-Architected Tool 允許與其他 共用工作負載 AWS 帳戶，並AWS Resource Access Manager協助您安全地與其他 帳戶共用您擁有 AWS 的資源。

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。跨帳戶角色可讓您安全地將帳戶中 AWS 資源的存取權授予第三方，例如 AWS Partner或您組織中的其他帳戶，同時保有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 可協助您允許第三方工作負載安全地與 AWS 工作負載互動，並進一步減少對長期憑證的需求。

   您不應該使用與使用者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 可讓您指定誰可以在IAM信任政策中擔任角色。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，它還為帳戶擁有者提供一種方法來允許僅在特定情況下擔任該角色。外部 ID 的主要功能是解決並防止混淆代理人問題。

   如果您是 AWS 帳戶 擁有者，且已為除了您 AWS 帳戶 之外存取其他 的第三方設定角色，或者您擔任代表不同客戶擔任角色的職位，請使用外部 ID。與第三方合作 AWS Partner 或建立外部 ID 條件，以包含在IAM信任政策中。

3. 使用通用唯一的外部 IDs。

   實作為外部 ID 產生隨機唯一值的程序，例如通用唯一識別碼 （UUID）。第三方IDs在不同的客戶之間重複使用外部 無法解決混淆的代理問題，因為客戶 A 可能可以使用客戶 B 的角色以及重複ARN的外部 ID 檢視客戶 B 的資料。在多租戶環境中，第三方支援多個具有不同 的客戶 AWS 帳戶，第三方必須使用不同的唯一 ID 作為每個 的外部 ID AWS 帳戶。第三方負責偵測重複的外部，IDs並安全地將每個客戶對應至各自的外部 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。在需要外部 ID 之前，第三方應避免儲存客戶角色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。

   取代長期憑證的使用，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。也請與您的 AWS 帳戶 團隊和第三方合作，建立風險緩解 Runbook。如需有關回應和緩解潛在安全事件的衝擊的方案指引，請參閱事件回應。

5. 確認設定具有方案指引且已自動化。

   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自動設定機制，該文件會為您使用 AWS CloudFormation 範本或同等項目。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。如需使用 AWS CloudFormation 範本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 範本或同等工具，您應該監控是否有偏離偵測的變更，作為稽核實務的一部分。

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。監控和稽核外部 的角色和資料存放區的使用IDs。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方案指引，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
- SEC03-BP06 根據生命週期管理存取權
- SEC03-BP07 分析公有和跨帳戶存取
- 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
- 如何將信任政策與IAM角色搭配使用
- AWS 帳戶 使用IAM角色委派跨 的存取權
- 如何使用 AWS 帳戶 存取另一個 中的資源IAM？
- 中的安全最佳實務 IAM
- 跨帳戶政策評估邏輯
- 將 AWS 資源的存取權授予第三方時，如何使用外部 ID
- 使用自訂 AWS CloudFormation 資源從外部帳戶中建立的資源收集資訊
- 安全地使用外部 ID 存取其他人擁有 AWS 的帳戶
- IAM使用 IAM Roles Anywhere 將IAM角色擴展至 以外的工作負載

相關影片：

- 如何允許個別 AWS 帳戶 存取我的 的使用者或角色 AWS 帳戶？
- AWS re：Invent 2018：在 60 分鐘內成為IAM政策主檔
- AWS 知識中心現場：IAM最佳實務和設計決策

相關範例：

- Well-Architected Lab - Lambda 跨帳戶IAM角色假設 （ 300 級 ）
- 設定對 Amazon DynamoDB 的跨帳戶存取權
- 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，以獲得可見性。請針對安全事件和潛在威脅採取行動，以協助保護工作負載。

最佳實務

- SEC04-BP01 設定服務和應用程式記錄
- SEC04-BP02 在標準化位置擷取日誌、調查結果和指標
- SEC04-BP03 關聯和豐富安全提醒
- SEC04-BP04 啟動不合規資源的修復

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本安全原則，也是由治理、風險和合規（GRC）標準、政策和程序驅動的常見安全要求。

預期結果：組織應能夠可靠且一致地在需要時從 AWS 服務和應用程式擷取安全事件日誌，以履行內部程序或義務，例如安全事件回應。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。

建立此最佳實務的好處：實作安全事件的根本原因分析（RCA）機制，以及治理、風險和合規義務的證據來源。

未建立此最佳實務時的曝險等級：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。產生提醒也需要日誌，以指出特定關注的動作已發生。選取、開啟、儲存和設定查詢與擷取機制和提醒至關重要。

實作步驟

- 選取並使用日誌來源。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。選取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。為每個 AWS 帳戶 使用 AWS CloudTrail 或追蹤建立 AWS Organizations 追蹤，並為其設定 Amazon S3 儲存貯體。

  AWS CloudTrail 是一種記錄服務，可追蹤針對 AWS 帳戶 擷取 AWS 服務活動進行的API呼叫。依預設，它會開啟 ，並保留 90 天的管理事件，這些 CloudTrail 事件可以透過使用 、 或 的事件歷史記錄擷取 AWS SDK。 AWS Management Console AWS CLI若要延長資料事件的保留和可見性，請建立 CloudTrail 追蹤並將其與 Amazon S3 儲存貯體建立關聯，並選擇性地與 Amazon CloudWatch 日誌群組建立關聯。或者，您可以建立 CloudTrail Lake ，該 Lake 可保留 CloudTrail 日誌長達七年，並提供SQL以 為基礎的查詢設施

  AWS 建議客戶分別使用 VPC Flow Logs 和 Amazon Route 53 解析器查詢日誌 VPC開啟網路流量和DNS日誌，並將其串流至 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流程日誌，您可以選擇如何使用流程日誌以及在何處降低成本。

  AWS CloudTrail 日誌、VPC流程日誌和 Route 53 解析程式查詢日誌是支援 中安全調查的基本日誌來源 AWS。您也可以使用 Amazon Security Lake 來收集、標準化和儲存 Apache Parquet 格式的日誌資料，以及準備好進行查詢的 Open Cybersecurity 結構描述架構 （ OCSF ）。Security Lake 也支援來自第三方來源的其他 AWS 日誌和日誌。

  AWS 服務可以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 調查結果、Amazon Elastic Kubernetes Service （ AmazonEKS ） 稽核日誌，以及 Amazon EC2執行個體作業系統和應用程式日誌。如需記錄和監控選項的完整清單，請參閱 AWS Security Incident Response Guide 的附錄 A：雲端功能定義 – 日誌記錄和事件。
- 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，每個選項都有自己的保留和生命週期功能。兩種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。如果主要記錄選項是 Amazon CloudWatch Logs，您應該考慮將較不常存取的日誌封存到 Amazon S3。

- 選取日誌儲存：日誌儲存的選擇通常與您使用的查詢工具、保留功能、熟悉度和成本相關。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。儲存在 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 提供耐用的儲存體和內建的查詢設施。

- 識別適當的日誌保留：當您使用 Amazon S3 儲存貯體或 CloudWatch 日誌群組儲存日誌時，您必須為每個日誌來源建立足夠的生命週期，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。

- 針對具有適當保留和生命週期政策的每個 AWS 服務和應用程式使用記錄：針對組織中的每個 AWS 服務或應用程式，尋找特定的記錄組態指南：

  - 設定 AWS CloudTrail 追蹤
  - 設定VPC流程日誌
  - 設定 Amazon GuardDuty Finding Export
  - 設定 AWS Config 錄製
  - 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 Control Plane 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

- 選取並實作日誌的查詢機制：對於日誌查詢，您可以將 CloudWatch Logs Insights 用於儲存在 CloudWatch 日誌群組中的資料，並將 Amazon Athena 和 Amazon OpenSearch Service 用於儲存在 Amazon S3 中的資料。您也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。

  選取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。選取符合營運、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，您可以使用第三方安全資訊和事件管理（SIEM）工具來執行過去 90 天的查詢，但由於 的日誌擷取成本，您可以使用 Athena 執行超過 90 天的查詢SIEM。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。

- 使用日誌進行警示： AWS 透過數個安全服務提供警示：
  - AWS Config 會監控和記錄您的 AWS 資源組態，並可讓您針對所需的組態自動化評估和修復。
  - Amazon GuardDuty 是一種威脅偵測服務，會持續監控惡意活動和未經授權的行為，以保護您 AWS 帳戶 和工作負載。 擷取、 GuardDuty 彙總和分析來源的資訊，例如 AWS CloudTrail 管理和資料事件、DNS日誌、VPCFlow Logs 和 Amazon EKS Audit Logs。直接從 CloudTrail、VPCFlow Logs、DNS查詢日誌和 Amazon GuardDuty 提取獨立資料串流EKS。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 提供單一位置，可彙總、組織和排序來自多個 AWS 服務和選用第三方產品的安全警示或調查結果，讓您全面檢視安全警示和合規狀態。

  您還可以使用自訂提醒產生引擎，取得這些服務未涵蓋的安全提醒或與您的環境相關的特定提醒。如需建置這些警示和偵測的相關資訊，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 部署前工具

相關文件：

- AWS 安全事件回應指南
- Amazon Security Lake 入門
- 入門：Amazon CloudWatch Logs
- 安全合作夥伴解決方案：日誌記錄和監控

相關影片：

- AWS re：Invent 2022 - Amazon Security Lake 簡介

相關範例：

- 的 Assisted Log Enabler AWS
- AWS Security Hub 調查結果歷史匯出

相關工具：

- Snowflake for Cybersecurity

SEC04-BP02 在標準化位置擷取日誌、調查結果和指標

安全團隊仰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：您採用標準化方法來收集、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，藉此探索可能發生的安全事件並識別異常。整合安全資訊和事件管理（SIEM）系統或其他機制，以查詢和分析日誌資料，以便及時回應、追蹤和上報安全事件。

常見的反模式：

- 多個團隊各自擁有並管理日誌記錄和指標收集工作，而他們的工作方式卻與組織的日誌記錄策略不一致。
- 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：擁有標準化的日誌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

組織中 AWS 使用量的增長會導致分散式工作負載和環境的數量增加。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。安全團隊使用安全資訊和事件管理（SIEM）系統等工具從分散式來源收集資料，並進行關聯、分析和回應工作流程。這需要管理一組複雜的許可，以存取各種資料來源，以及在操作擷取、轉換和載入（ETL）程序時額外額外負荷。

若要克服這些挑戰，請考慮將安全日誌資料的所有相關來源彙總到 Log Archive 帳戶，如使用多個帳戶組織 AWS 您的環境中所述。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 AWS 帳戶 具有適當跨帳戶許可的個別 中，在標準化位置擷取此資料有幾個優點。此實務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。您可以設定 Security Lake 自動從常見來源擷取資料，例如 CloudTrail、Route 53、Amazon EKS和 VPC Flow Logs。您也可以將 AWS Security Hub 設定為 Security Lake 的資料來源，讓您將 Amazon GuardDuty 和 Amazon Inspector等 AWS 其他服務的調查結果與日誌資料建立關聯。 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開放網路安全結構描述架構（OCSF）格式，並存放在 Amazon S3 儲存貯體中作為 Parquet 檔案，無需ETL處理。

將安全資料存放在標準化位置可提供進階分析功能。AWS 建議您將安全分析的工具部署 AWS 到與 Log Archive 帳戶分開的安全工具帳戶。 此方法可讓您深入實作控制，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 Log Archive 和 Security Tooling 帳戶。如果您使用 AWS Control Tower 來管理組織，系統會自動為您建立 Log Archive 和 Security Tooling 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。 您可以透過常見資料湖架構方法、第三方資料產品或 Amazon Security Lake 等選項來實現此目標。AWS 建議您從為帳戶選擇加入 AWS 區域 的擷取安全資料，即使 未使用。

3. 設定將資料來源發佈到標準化位置

a. 識別您的安全資料的來源，並將其設定為發佈到標準化位置。評估選項，以所需格式自動匯出資料，而不是需要開發ETL程序的格式。使用 Amazon Security Lake，您可以從支援的 AWS 來源和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 AthenaAmazon QuickSight或第三方解決方案等工具，以擁有標準化位置所需的存取權。 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 帳戶分隔工作負載
- SEC07-BP04 定義資料生命週期管理
- SEC08-BP04 強制執行存取控制
- OPS08-BP02 分析工作負載日誌

相關文件：

- AWS 白皮書：使用多個帳戶組織 AWS 您的環境
- AWS 規範指南： AWS 安全參考架構 （AWS SRA）
- AWS 方案指引：應用程式擁有者的日誌記錄和監控指南

相關範例：

- 使用 Amazon Athena 和 Amazon 從分散式來源彙總、搜尋和視覺化日誌資料 QuickSight
- 如何使用 Amazon 視覺化 Amazon Security Lake 調查結果 QuickSight
- 使用 Amazon SageMaker Studio 和 Amazon Bedrock 為 Amazon Security Lake 產生 AI 支援的洞見
- 使用 Amazon 識別 Amazon Security Lake 資料中的網路安全異常 SageMaker
- 擷取、轉換和交付 Amazon Security Lake 發佈的事件至 Amazon OpenSearch Service

- 如何使用 AWS Security Hub 和 Amazon OpenSearch Service SIEM

相關工具：

- Amazon Security Lake
- Amazon Security Lake 合作夥伴整合
- 開放網路安全結構描述架構（OCSF）
- Amazon Athena
- Amazon QuickSight
- Amazon Bedrock

SEC04-BP03 關聯和豐富安全提醒

非預期的活動可能會導致不同來源產生多個安全提醒，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。實作安全提醒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提醒時，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，進而協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提醒進行調查 (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提醒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。
- 您只仰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：自動建立提醒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「事件」及正式回應所需的時間。額外的內容還可協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提醒表明的嚴重性。

未建立此最佳實務時的曝險等級：低

實作指引

安全提醒可能來自 中的許多不同來源 AWS，包括：

- Amazon GuardDuty、AWS Security Hub、Amazon Macie、Amazon Inspector、AWS Identity and Access Management Access Analyzer、 AWS Config和 Network Access Analyzer 等服務
- 自動分析 AWS 服務、基礎設施和應用程式日誌的警示，例如來自 Security Analytics for Amazon OpenSearch Service 的警示。
- 回應來自 Amazon CloudWatch、Amazon EventBridge或 等來源之帳單活動變更的警示AWS Budgets。
- 來自 的威脅情報摘要和安全合作夥伴解決方案等第三方來源 AWS Partner Network
- AWS Trust & Safety 或其他來源的聯絡，例如客戶或內部員工。

提醒基本上包含有關誰 (主體或身分)、做了什麼 (採取的動作)，以及對象是誰 (受影響的資源) 的資訊。對於其中每個來源，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。這可以採用將警示來源與安全資訊和事件管理 （ SIEM ） 工具整合的形式，以為您執行自動關聯、建置您自己的資料管道和處理，或兩者的組合。

可為您建立相互關聯的服務範例為 Amazon Detective。Detective 會執行持續從各種和 AWS 第三方來源擷取警示，並使用不同形式的情報來組合其關係的視覺化圖形，以協助調查。

雖然提醒的初始關鍵性可協助判斷優先順序，但提醒發生的環境則決定了其真實的關鍵性。例如，Amazon GuardDuty 可以提醒工作負載中的 Amazon EC2執行個體正在查詢非預期的網域名稱。GuardDuty 可能會自行為此提醒指派低嚴重性。不過，警示發生時與其他活動的自動關聯可能會發現有數百個EC2執行個體透過相同的身分部署，這增加了整體營運成本。在此情況下， GuardDuty 可能會將此相關事件內容發佈為新的安全警示，並將重要性調整為高，這將加快進一步的動作。

實作步驟

1. 識別安全提醒資訊的來源。了解來自這些系統的提醒如何表示身分、動作和資源，以確定可能的相互關聯。
2. 建立一個機制來擷取不同來源的提醒。為此，請考慮 Security Hub EventBridge、 和 CloudWatch 等服務。
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、VPCFlow Logs、Amazon Security Lake 以及基礎設施和應用程式日誌。
4. 將提醒與資料相互關聯和增添豐富性的來源整合在一起，以建立更詳細的安全事件內容並構成關鍵性。
    a. Amazon Detective、SIEM模具或其他第三方解決方案可以自動執行特定層級的擷取、關聯和擴充。

b. 您也可以使用 AWS 服務來建置您自己的 。例如，您可以叫用 AWS Lambda 函數，針對 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 建立可操作的警示
- REL06-BP03 傳送通知（即時處理和警示）

相關文件：

- AWS 安全事件回應指南

相關範例：

- 如何使用帳戶中繼資料豐富 AWS Security Hub 調查結果
- 如何使用 AWS Security Hub 和 Amazon OpenSearch Service SIEM

相關工具：

- Amazon Detective
- Amazon EventBridge
- AWS Lambda
- Amazon Athena

SEC04-BP04 啟動不合規資源的修復

您的偵測控制可能針對不符合您組態需求的資源發出提醒。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。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提醒。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。
- 您只仰賴補救措施，而不是將補救措施視為更廣泛的事件回應和復原計畫的一部分。

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提醒，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曝險等級：中

實作指引

如 SEC01-BP03 識別和驗證控制目標 中所述，AWS Config 等服務可協助您監控帳戶中資源的組態，以確保符合您的需求。 偵測到不合規的資源時，建議您設定將警示傳送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修正動作。

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，對設定錯誤VPC安全群組的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、AWS Systems Manager Automation 文件中，或透過您慣用的其他程式碼環境來定義回應。確保環境能夠 AWS 使用具有採取修正動作所需最低許可量IAM的角色來驗證 。

定義所需的修復之後，您就可以決定偏好的啟動方法。 AWS Config 可以為您啟動修復。如果您使用的是 Security Hub，您可以透過自訂動作 來執行此操作，這會將調查結果資訊發佈至 Amazon EventBridge。然後， EventBridge 規則可以啟動您的修復。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，並識別改進之處。擷取 Amazon CloudWatch Logs 中的日誌和修復結果，作為 Security Hub 中的調查結果備註。

首先，請考慮 上的自動安全回應 AWS，其已預先建置修正來解決常見的安全錯誤設定。

實作步驟

1. 分析提醒並排定優先順序。
   a. 將各種 AWS 服務的安全提醒合併到 Security Hub，以實現集中可見性、優先順序和修復。
2. 制定補救措施。
   a. 使用 Systems Manager 和 等服務 AWS Lambda 來執行程式設計修復。
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 的自訂動作 EventBridge。設定手動或自動啟動這些動作。
   b. 您也可以視需要使用 Amazon Simple Notification Service （ SNS ） 傳送通知和提醒給相關利益相關者 （ 例如安全團隊或事件回應團隊 ），以進行手動介入或升級。
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 安全事件回應指南 - 偵測

相關範例：

- 上的自動安全回應 AWS
- 使用 監控EC2執行個體金鑰對 AWS Config
- 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 自動化
- 上的自動安全回應 AWS

基礎設施保護

問題

- SEC 5. 如何保護網路資源？
- SEC 6. 如何保護運算資源？

SEC 5. 如何保護網路資源？

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 實作以檢查為基礎的保護
- 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分組為基礎，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。

預期結果：您網路的圖層是整體 defense-in-depth安全方法的一部分，可補充工作負載的身分驗證和授權策略。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：

- 您可以在單一 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。
- 您使用 VPCs和 子網路作為所有網路層考量的預設，且不考慮 AWS 受管服務如何影響您的拓撲。

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曝險等級：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。基礎網路控制可協助強制執行工作負載的資料存取需求。例如，在三層 Web 應用程式架構中，您可以將靜態呈現層檔案存放在 Amazon S3 上，並從內容交付網路（CDN）提供這些檔案，例如 Amazon CloudFront。應用程式層可以具有 Application Load Balancer（ALB）在 Amazon VPC 公有子網路（類似非隔離區域或 DMZ）中服務的公有端點，並將後端服務部署到私有子網路。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。在每個層邊界（CDN、公有子網路、私有子網路）上，您可以部署控制項，僅允許授權流量周遊這些邊界。

類似於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。在此情況下，根據您的控制需求，使用多個私有子網路分割應用程式層、在相同的 VPCs中不同 AWS 帳戶，或甚至 AWS 帳戶 在每個資料敏感度層級VPCs中不同。

對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提醒。

在設計過程中，請考慮使用 AWS 受管服務如何影響您的網路拓撲。探索 Amazon VPC Lattice 等 服務如何協助簡化跨網路層的工作負載元件互通性。使用 時AWS Lambda，除非有特定原因，否則請在VPC子網路中部署 。判斷VPC端點的位置，並AWS PrivateLink可以簡化遵守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代理來提供公有端點。使用 、Amazon API Gateway CloudFront、Elastic Load Balancing 和 等受管服務AWS Amplify來託管公有端點，以探索安全控制項的轉移。

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，請將這些元件部署到私有子網路中，以第一步驟中的群組為基礎。
4. 對於 Amazon DynamoDB 、Amazon Kinesis或 Amazon SQS等完全受管 AWS 服務，請考慮使用 VPC端點作為透過私有 IP 地址存取的預設。

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 networking 基礎

相關範例：

- VPC 範例
- ECS使用 AWS FargateAWS PrivateLink、 和 Network Load Balancer 私下存取 Amazon 上的容器應用程式
- 透過 VPC使用 Amazon 在 Amazon S3 儲存貯體中提供靜態內容 CloudFront

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、區域法規以及協定需求等因素。在可能的情況下，您會偏好 point-to-point透過網路對等進行流程，作為最低權限設計原則的一部分。

常見的反模式：

- 您採用以周邊為基礎的網路安全方法，僅控制網路層邊界處的流量流程。

- 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，而不是對兩者都實施。
- 您只仰賴工作負載元件和網路控制項來驗證和授權流量。

建立此最佳實務的優勢：此實務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。透過執行流量流程控制，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曝險等級：高

實作指引

雖然網路層有助於建立工作負載中提供類似函數、資料敏感度層級和行為的元件邊界，但您可以使用技術來進一步分割這些層內遵循最低權限原則的元件，從而建立更精細的流量控制層級。在 內 AWS，網路層主要是根據 Amazon 內的 IP 地址範圍使用子網路定義VPC。也可以使用不同的 定義層VPCs，例如依業務網域分組微服務環境。使用多個 時VPCs，請使用 調節路由AWS Transit Gateway。雖然這使用安全群組和路由表提供第 4 層 （IP 地址和連接埠範圍） 的流量控制，但您可以使用其他服務獲得進一步控制，例如 AWS PrivateLink、Amazon Route 53 Resolver DNS Firewall 、 AWS Network Firewall和 AWS WAF。

了解並清查工作負載的資料流程和通訊需求，包括連線起始方、連接埠、通訊協定和網路層。評估可用於建立連線和傳輸資料的通訊協定，以選取符合保護需求的通訊協定 （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。一個好的起點是在您的 中使用安全群組VPC，因為它們可以連接到使用彈性網路介面 （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。與第 4 層防火牆不同的是，安全群組可以設置一項規則來依識別碼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傳入和傳出規則來篩選流量。

當流量在 之間移動時VPCs，通常會使用VPC對等進行簡單路由，或使用 AWS Transit Gateway 進行複雜路由。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。不過，如果您的工作負載只需要不同 中特定元件之間的流量，VPCs請考慮使用 point-to-point連線AWS PrivateLink。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。部署生產者的相容負載平衡器，相應地 PrivateLink開啟，然後接受消費者的連線請求。然後，生產者服務會從消費者的私有 IP 地址指派VPC，供消費者用來提出後續請求。這種方法減少了對等網路的需求。包含資料處理和負載平衡的成本，作為評估 的一部分 PrivateLink。

雖然安全群組和 PrivateLink 協助控制工作負載元件之間的流程，但另一個主要考量是如何控制資源允許存取的DNS網域 （如果有的話）。根據 的DHCP組態VPCs，您可以為此考慮兩種不同的 AWS 服

務。大多數客戶會使用其CIDR範圍VPCs的 +2 地址可用的預設 Route 53 Resolver DNS服務（也稱為 Amazon DNS 伺服器或 AmazonProvidedDNS）。透過此方法，您可以建立DNS防火牆規則並將其與建立關聯VPC，以決定要針對您提供的網域清單採取哪些動作。

如果您不使用 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，以確定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傳入 Web 流量。如需這些服務的進一步指引，請參閱 SEC05-BP03 實作檢查型保護 。

實作步驟

1. 識別工作負載元件之間的必要資料流程。
2. 使用傳入和傳出流量的方法 defense-in-depth套用多個控制項，包括使用安全群組和路由表。
3. 使用防火牆來定義對 內、外和整個 的網路流量的精細控制VPCs，例如 Route 53 Resolver DNS Firewall 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 選擇如何分割工作負載
- SEC09-BP02 在傳輸中強制執行加密

相關文件：

- 您 的安全最佳實務 VPC
- AWS 網路最佳化秘訣
- 上的網路安全指南 AWS
- 在 中保護 VPC的傳出網路流量 AWS 雲端

相關工具：

- AWS Firewall Manager

相關影片：

- AWS Transit Gateway 許多 的參考架構 VPCs
- 使用 Amazon 進行應用程式加速和保護 CloudFront AWS WAF，以及 AWS Shield
- AWS re:Inforce 2023：防火牆和設置位置

相關範例：

- 實驗室： CloudFront 適用於 Web Application

SEC05-BP03 實作以檢查為基礎的保護

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。 允許和拒絕決策取決於明確的規則、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：

- 僅仰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目前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，以識別行為異常。

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。根據最新的威脅情報，隨著威脅態勢隨著時間的變化，從 AWS 和 合作夥伴的受管規則集中受益。 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曝險等級：中

實作指引

使用 AWS Network Firewall或其他防火牆和入侵預防系統（IPS）精細控制具狀態和無狀態網路流量 AWS Marketplace ，您可以在 Gateway Load Balancer（GWLB）後方部署。AWS Network Firewall 支援 Suricata 相容的開放原始碼IPS規格，以協助保護您的工作負載。

AWS Network Firewall 和廠商解決方案都支援GWLB不同的內嵌檢查部署模型。 例如，您可以根據基準VPC執行檢查、集中在檢查 中VPC，或在混合模型中部署，其中東西流量會流經檢查，VPC而

網際網路傳入會按檢查VPC。 另一個考量是解決方案是否支援取消包裝 Transport Layer Security（TLS），可針對任一方向啟動的流量進行深度封包檢查。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。

如果您使用的是執行 out-of-band檢查的解決方案，例如從以半透明模式操作的網路介面進行封包資料的 pcap 分析，則可以設定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。您可以查看這些設備的虛擬版本是否可在 上使用AWS Marketplace，這可能支援在 後方的內嵌部署GWLB。

對於透過 HTTP型通訊協定進行交易的元件，請使用 Web 應用程式防火牆 （WAF） 保護您的應用程式免受常見威脅。 AWS WAF 是一種 Web 應用程式防火牆，可讓您在傳送至 Amazon API Gateway、Amazon CloudFront AWS AppSync 或 Application Load Balancer 之前，監控並封鎖符合您可設定規則的 HTTP（S） 請求。當您評估 Web 應用程式防火牆的部署時，請考慮進行深度封包檢查，因為有些 會要求您在流量檢查TLS之前終止。若要開始使用 AWS WAF，您可以AWS 受管規則搭配自己的 使用，或使用現有的合作夥伴整合 。

您可以使用 集中管理組織中的 AWS Network Firewall AWS AWS WAF AWS Shield Advanced、 和 Amazon VPC安全群組AWS Firewall Manager。

實作步驟

1. 判斷您是否可以廣泛地範圍檢查規則，例如透過檢查 VPC，或者是否需要每個VPC方法更精細。
2. 對於內嵌檢測解決方案：
   a. 如果使用 AWS Network Firewall，請建立規則、防火牆政策和防火牆本身。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. 如果搭配 Gateway Load Balancer （GWLB） 使用第三方設備，請在一或多個可用區域中部署和設定您的設備。然後，建立您的 GWLB、端點服務、端點，並設定流量的路由。
3. 對於 out-of-band檢查解決方案：
   1. 在應鏡像傳入和傳出流量的介面上開啟VPC流量鏡像。您可以使用 Amazon EventBridge 規則來叫用 AWS Lambda 函數，以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. 對於傳入 Web 流量解決方案：
   a. 若要設定 AWS WAF，請先設定 Web 存取控制清單 （web ACL）。Web ACL是具有序列處理預設動作 （ALLOW 或 DENY） 的規則集合，可定義您的 WAF 如何處理流量。您可以在 Web 中建立自己的規則和群組，或使用 AWS 受管規則群組ACL。
   b. 設定 Web ACL 後，請將 Web ACL與 AWS 資源 （例如 Application Load Balancer、APIGateway 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 具有路由的範例架構
- 使用 AWS Gateway Load Balancer 和 的集中式檢查架構 AWS Transit Gateway

相關範例：

- 部署 Gateway Load Balancer 的最佳實務
- TLS 加密輸出流量和 的檢查組態 AWS Network Firewall

相關工具：

- AWS Marketplace IDS/IPS

SEC05-BP04 自動化網路保護

使用 DevOps實務自動化網路保護的部署，例如基礎設施，例如程式碼（IaC和 CI/CD 管道。 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，並將其遞交至版本控制系統中。 當進行新的變更時，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，以驗證控制是否如預期運作。 控制得到核准後，也會自動將其部署到實際執行環境中。

常見的反模式：

- 仰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。
- 仰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。

未建立此最佳實務時的曝險等級：中

實作指引

SEC05-BP02 控制網路層內的流量流和 SEC05-BP03 實作檢查型保護中所述的一些網路保護控制項，隨附可依據最新威脅情報自動更新的受管規則系統。 保護 Web 端點的範例包括AWS WAF 受管規則和AWS Shield Advanced 自動應用程式層DDoS緩解。 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，我們建議您使用 DevOps 實務來自動部署網路資源、保護和您指定的規則。 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，將它們遞交至版本控制系統，並使用 CI/CD 管道部署它們。 使用此方法取得 DevOps 管理網路控制項的傳統優點，例如更可預測的版本、使用 等工具進行自動測試AWS CloudFormation Guard，以及偵測部署環境與所需組態之間的偏離。

根據您在 SEC05-BP01 建立網路層 中所做的決策，您可能有建立專用VPCs於輸入、輸出和檢查流程的中央管理方法。 如AWS 安全參考架構 （AWS SRA） 中所述，您可以在VPCs專用網路基礎設施帳戶中定義這些屬性。 您可以使用類似的技術，集中定義其他帳戶中工作負載VPCs使用的 、其安全群組、 AWS Network Firewall 部署、Route 53 Resolver 規則和DNS防火牆組態，以及其他網路資源。 您可以透過 AWS Resource Access Manager 將這些資源與其他帳戶共用。 使用此方法，您可以簡化對網路控制的自動測試並將該控制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，並將其他控制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. 確定如何在版本控制系統中儲存和維護範本。 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。 定義測試以檢查是否有組態錯誤，以及範本是否符合您公司的標準。

資源

相關的最佳實務：

- SEC01-BP06 自動化標準安全控制的部署

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
- NetDevSecOps 將 AWS 網路部署現代化
- 將 AWS CloudFormation 安全測試與 AWS Security Hub 和 AWS CodeBuild 報告整合

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要多層防禦，以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 從強化的影像佈建運算
- 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 （AmazonECS） 容器和 Amazon Elastic Kubernetes Service （AmazonEKS） 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。
- 使用已超過廠商提供之生命週期結束 （EOL） 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。

未建立此最佳實務時的曝險等級：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 共同責任模型下， AWS 負責保護 的基礎設施 AWS 雲端。此基礎設施由執行 AWS 雲端 服務的硬體、軟體、聯網和設施組成。您要負責雲端的安全，例如 Amazon EC2執行個體的實際資料、安全組態和管理任務，並驗證您的 Amazon S3 物件是否已正確分類和設定。您著手漏洞管理的方法也可能視取用的服務而異。例如， AWS 管理我們受管關聯式資料庫服務的修補，Amazon RDS，但您將負責修補自我託管資料庫。

AWS 具有一系列服務，可協助您管理漏洞。Amazon Inspector 會持續掃描 AWS 工作負載是否有軟體問題和非預期的網路存取。AWS Systems Manager Patch Manager 可協助管理 Amazon EC2執行個體的修補。Amazon Inspector 和 Systems Manager 可以在 中檢視AWS Security Hub，這是雲端安全狀態管理服務，可協助自動化 AWS 安全檢查並集中安全提醒。

Amazon CodeGuru 可以使用靜態程式碼分析，協助識別 Java 和 Python 應用程式中的潛在問題。

實作步驟

- 設定 Amazon InspectorAmazon Inspector 會自動偵測新啟動的 Amazon EC2執行個體、Lambda 函數和推送至 Amazon 的合格容器映像，ECR並立即掃描是否有軟體問題、潛在瑕疵和非預期的網路暴露。
- 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，以修復 Java 和 Python 應用程式的常見安全問題。OWASP 基金會發佈來源程式碼分析工具的清單（也稱為SAST工具）。
- 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。您的方法可能會根據您所使用的 AWS 服務而有所不同。若要檢查在 Amazon EC2執行個體中執行之軟體中是否有潛在問題，請將 Amazon Inspector 新增至您的管道，以便在偵測到問題或潛在瑕疵時提醒您並停止建置程序。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 、Open VAS、套件管理員和 AWS Partner 工具進行漏洞管理。
- 使用 AWS Systems Manager：您必須負責 AWS 資源的修補程式管理，包括 Amazon Elastic Compute Cloud（Amazon EC2）執行個體、Amazon Machine Images（AMIs）和其他運算資源。AWS Systems Manager Patch Manager 以安全相關和其他類型的更新自動化以修補受管執行個體。Patch Manager 可用於在 Amazon EC2執行個體上為作業系統和應用程式套用修補程式，包括 Microsoft 應用程式、Windows 服務套件，以及 Linux 型執行個體的次要版本升級。除了 Amazon 之外EC2，Patch Manager 也可用於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，僅查看遺漏的修補程式報告，或者掃描並自動安裝所有遺漏的修補程式。
- 使用 AWS Security Hub：Security Hub 提供 中安全狀態的全面檢視 AWS。它收集跨多個 AWS 服務的安全資料，並以標準化格式提供這些調查結果，允許您優先處理跨 AWS 服務的安全調查結果。
- 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
- 的安全概觀 AWS Lambda

- 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
- AWS 使用 Amazon Inspector 和 AWS Systems Manager - 第 1 部分自動化漏洞管理和修復

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 從強化的影像佈建運算

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。
- 僅仰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。

您可以使用受信任來源提供的指南來減輕強化系統的負擔，例如網際網路安全中心（CIS）和國防資訊系統局（DISA）安全技術實作指南（STIGs）。我們建議您從 AWS 或 APN合作夥伴發佈的 Amazon Machine Image（AMI）開始，並使用AWS EC2 Image Builder 根據適當的 CIS和 STIG控制項組合來自動化組態。

雖然有可用的強化映像和EC2映像建置器配方會套用 CIS或 DISASTIG建議，但您可能會發現其組態阻止軟體順利執行。在這種情況下，您可以從非強化基本映像開始，安裝軟體，然後逐步套用CIS控制項來測試其影響。對於防止軟體執行的任何CIS控制項，請測試您是否可以在 DISA 中實作更精細的強化建議。追蹤您能夠成功套用的不同CIS控制項和DISASTIG組態。使用這些選項，相應地在 Image Builder 中定義您的EC2映像強化配方。

對於容器化工作負載，來自 Docker 的強化影像可在 Amazon Elastic Container Registry（ECR）公有儲存庫 上取得。您可以使用 EC2 Image Builder 搭配 來強化容器映像AMIs。

與作業系統和容器映像類似，您可以透過 pip、npm、Maven 和 等工具，從公有儲存庫取得程式碼套件（或程式庫 ）NuGet。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。此整合可以 up-to-date為您處理擷取、儲存和保留套件。然後，您的應用程式建置程序可以使用 Software Composition Analysis（SCA）、Static Application Security Testing（SAST）和 Dynamic Application Security Testing（ ）等技術，取得並測試這些套件的最新版本DAST。

對於使用 的無伺服器工作負載 AWS Lambda， 可簡化使用 Lambda 層管理套件相依性。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。您可以透過自己的建置程序來建立和維護層，為您的函數提供保留 的中央方式 up-to-date。

實作步驟

- 強化作業系統。使用信任來源的基本映像作為建置強化 的基礎AMIs。使用 EC2 Image Builder 協助自訂安裝在映像上的軟體。
- 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，請在建置管道中實作ECR映像掃描，並定期針對映像儲存庫實作映像掃描，以便在CVEs容器中尋找。
- 搭配 使用無伺服器實作時 AWS Lambda，請使用 Lambda 層來分隔應用程式函數程式碼和共用相依程式庫。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 安全性

相關範例：

- AMI使用 EC2 Image Builder 快速建置 STIG合規
- 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 end-to-end 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：

- 使用 SSH或 等通訊協定互動式存取 Amazon EC2執行個體RDP。
- 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。
- 登入執行個體以對問題進行疑難排解。

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。移除使用 Secure Shell （SSH） 和遠端桌面通訊協定 （RDP） 進行互動式存取會減少對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曝險等級：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、變更組態及對問題進行疑難排解。

但是，手動存取伴隨著許多風險。它需要接聽請求的伺服器，例如 SSHRDP或服務，其可提供未經授權存取的潛在路徑。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件、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。

若要降低這些風險，您可以實作以代理程式為基礎的遠端存取解決方案，例如 AWS Systems Manager . AWS Systems Manager Agent （ SSM 代理程式） 會啟動加密頻道，因此不依賴聆聽外部啟動的請求。請考慮設定SSM客服人員，透過VPC端點 建立此頻道。

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 也可以提供遠端 shell 的存取權，並在工作階段期間記錄調用的每個命令及其輸出，以記錄日誌和 Amazon S3。 會AWS CloudTrail記錄 Systems Manager 的調用APIs以供檢查。

實作步驟

1. 在 Amazon EC2執行個體上安裝AWS Systems Manager 代理程式 （ SSM 代理程式 ）。檢查SSM客服人員是否包含在基本AMI組態中並自動啟動。
2. 驗證與您的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. 停用在執行個體上執行的 RDP、 SSH和其他遠端存取服務。您可以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AMIs工具建置自訂的指令碼來達成此目的。
4. 確認適用於EC2執行個體的安全群組傳入規則不允許存取連接埠 22/tcp （ SSH ） 或連接埠 3389/tcp （ ）RDP。使用 AWS Config等服務實作偵測，並對設定錯誤的安全群組發出提醒。
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政策來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 中維護稽核追蹤。

資源

相關的最佳實務：

- REL08-BP04 使用不可變的基礎設施部署

相關範例：

- 使用 AWS Systems Manager 取代SSH存取以減少管理和安全額外負荷

相關工具：

- AWS Systems Manager

相關影片：

- 在 Session Manager 中控制使用者 AWS Systems Manager 對執行個體的存取

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。
- 僅仰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，請檢查提供者是否提供了用於驗證數位簽章的公有金鑰。 以下一些範例說明 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：驗證安裝下載的 AWS TOE簽章
- AWS Systems Manager：驗證SSM客服人員的簽章
- Amazon CloudWatch：驗證 CloudWatch 客服人員套件的簽章

將數位簽章驗證納入您用於取得和強化影像的程序，如 SEC06-BP02 佈建來自強化影像的運算中所述。

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 納入您的持續整合和交付 (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 協助保護容器映像建置管道的最佳實務 AWS Signer
- 宣佈使用 AWS Signer 和 Amazon 進行容器映像簽署 EKS
- 設定 的程式碼簽署 AWS Lambda
- Lambda 程式碼簽署的最佳實務和進階模式
- 使用 AWS Certificate Manager 私有 CA AWS Key Management Service 和非對稱金鑰進行程式碼簽署

相關範例：

- 使用 Amazon CodeCatalyst 和 自動化 Lambda 程式碼簽署 AWS Signer
- 使用 簽署和驗證OCI偽影 AWS Signer

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。 在 CI/CD 程序中整合自動化，以部署具有 up-to-date相依性的可信任工作負載。

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。您可以使用自動驗證來檢查軟體映像和相依性是否來自信任來源，以及是否遭到竄改。工作負載會自動檢查 up-to-date相依性，並簽署以在運算環境中建立可信度 AWS 。 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取代實際執行系統。
- 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。 它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曝險等級：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 CI/CD 管道和 中使用 Amazon Inspector，以持續掃描執行期環境是否有已知的常見漏洞

和暴險（CVEs）。 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以進一步了解。

自動化在部署值得信任的工作負載中也扮演了重要角色，如 SEC06-BP02 佈建從強化映像和 06-BP04 驗證軟體完整性 的運算中所述。 SEC06-BP04 您可以使用 EC2 Image Builder 、AWS CodeArtifact、 AWS Signer和 Amazon Elastic Container Registry（ECR）等服務來下載、驗證、建構和存放強化和核准的映像和程式碼相依性。 除了 Inspector 之外，這些都可以在您的 CI/CD 程序中扮演角色，因此您的工作負載只有在確認其相依性是 up-to-date來自可信任來源時才開始生產。 您的工作負載也會經過簽署，因此 AWS 運算環境，例如 AWS Lambda和 Amazon Elastic Kubernetes Service（EKS） 可以在允許執行之前驗證其未遭到竄改。

除了這些預防性控制之外，您還可以在偵測控制中針對運算資源使用自動化。 例如， AWS Security Hub提供 NIST 800-53 修訂版 5 標準，其中包含檢查，例如 【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 使用工作階段身分驗證技術，封鎖包含 X-Forwarded-For HTTP 標頭的請求，以及 1 TTL的網路，以停止來自外部來源的流量，以擷取EC2執行個體的相關資訊。Security Hub 中的此檢查可以偵測EC2執行個體何時使用IMDSv1並啟動自動修復。進一步了解 SEC04-BP04 中的自動偵測和修復 啟動不合規資源的修復 。

實作步驟

1. AMIs 使用 EC2 Image Builder 自動化建立安全、合規和強化。 您可以產生包含來自 Center for Internet Security（CIS）Benchmarks 或 Security Technical Implementation Guide（STIG）標準之控制項的映像，這些標準來自基礎映像 AWS 和APN合作夥伴映像。
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 自動化安全和合規狀態管理
3. 自動化修補或取代 Amazon Elastic Compute Cloud（Amazon EC2）執行個體。 AWS Systems Manager Patch Manager 會使用安全相關和其他類型的更新，自動修補受管執行個體的程序。您可以使用修補程式管理員以套用適用於作業系統和應用程式的修補程式。
   a. AWS Systems Manager 修補程式管理員
4. 自動掃描運算資源是否有常見漏洞和暴露（CVEs），並在建置管道中嵌入安全掃描解決方案。
   a. Amazon Inspector
   b. ECR 映像掃描

5. 考慮使用 Amazon GuardDuty 進行自動惡意軟體和威脅偵測，以保護運算資源。 GuardDuty 也可以在您的 AWS 環境中調用AWS Lambda函數時識別潛在問題。
   a. Amazon GuardDuty
6. 考慮 AWS 合作夥伴解決方案。 AWS 合作夥伴提供與內部部署環境中現有控制項同等、相同或整合的業界領先產品。這些產品可補充現有的 AWS 服務，讓您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. 基礎設施安全性

資源

相關的最佳實務：

- SEC01-BP06 自動化標準安全控制的部署

相關文件：

- 取得IMDSv1基礎設施 AWS 的完整優勢IMDSv2並停用

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. 如何保護傳輸中的資料？

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 了解您的資料分類方案

- SEC07-BP02 根據資料敏感度套用資料保護控制
- SEC07-BP03 自動化識別和分類
- SEC07-BP04 定義可擴展的資料生命週期管理

SEC07-BP01 了解您的資料分類方案

了解您的工作負載要處理的資料分類、其處理需求、關聯的業務流程、資料儲存在何處，以及誰是資料擁有者。 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需求，以及需要何種資料控制。 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，可根據資料分類來保護敏感資料。 這些控制左右著下列考量：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需求、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所述的資料敏感程度和需求，對資料實施適當的控制
- 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：此實務可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，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。這些實務還可針對每一種資料類型實施適當的控制，進而有助於降低成本。

未建立此最佳實務時的曝險等級：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。然後，您可以確定是否設置了正確的控制、是否可稽核控制，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，您可以套用標籤索引鍵為 `Classification`且標籤值為 的標籤，PHI以取得受保護的健康資訊 （ PHI ），以及另一個標籤索引鍵為 `Sensitivity`且標籤值為 的標籤`High`。 然後，您可以使用 AWS Config 等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) 的情況時發出提醒。 您可以使用標籤政策 (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. 識別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 標籤政策可用於強制執行標記標準。

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 根據資料敏感度套用資料保護控制

實施資料保護控制，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。 此實務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布了清楚的指引，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關聯成本，針對每一種敏感程度實施控制。 您設置了監控和提醒，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。這可能會導致對低敏感性資料過度佈建安全控制，或對高敏感性資料的保護不足。
- 在定義資料保護控制時，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伴隨的營運支出和成本。
- 未定期審查資料保護控制，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，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。 在適度採取較少控制的情況下，您可以改善員工、客戶或成員使用的資料存取性和完整性。 此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曝險等級：高

實作指引

根據資料敏感程度實施資料保護控制的方式包含幾個重要的步驟。首先，識別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。我們建議您使用服務控制政策 （SCPs） 將資料分隔為不同的 AWS 帳戶，以限制每個資料敏感度層級允許的服務和動作。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。請參閱保護靜態資料和保護傳輸中的資料的最佳實務，以實作加密、存取控制和稽核等相關控制。考慮採用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作控制的有效性。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。實作的資料保護控制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，協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. 識別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。

資源

相關的最佳實務：

- PERF03-BP01 使用最能夠支援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、身分及合規最佳實務
- 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 自動化識別和分類

將資料的識別和分類自動化，可協助您實作正確的控制。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。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，以偵測並提醒未經授權的資料儲存或處理行為，以便快速採取更正動作。

常見的反模式：

- 僅仰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，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，並採取更正動作。

未建立此最佳實務時的曝險等級：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。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 中 AWS，您可以將資料集上傳到 Amazon S3，並使用 Amazon Macie、Amazon Comprehend或 Amazon Comprehend Medical 進行掃描。 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提醒，就能在部署到實際執行環境之前，讓防護措施的落差浮現。其他功能，例如 中的敏感資料偵測、Amazon SNS和 Amazon CloudWatchAWS Glue，也可以用來偵測PII和採取緩解動作。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，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案或複製到資料分析環境中的情

形。 您可以使用 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. 對您的環境執行初始掃描，以進行自動識別和分類。
   a. 初始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，您可以設定 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。 然後，您可以使用敏感資料探索工作來深入分析 S3 儲存貯體。也可以將其他資料存放區匯出至 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提醒，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 中使用受管資料識別碼 SNS
- Amazon CloudWatch Logs：使用遮罩協助保護敏感日誌資料

相關範例：

- 使用 Macie 啟用 Amazon RDS 資料庫的資料分類
- 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 one-size-fits-all資料生命週期管理方法，而不考慮不同的敏感度等級和存取要求。
- 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。
- 仰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來自動化跨 AWS 服務的資料備份。 Amazon EBS快照提供複製EBS磁碟區並使用 S3 功能儲存磁碟區的方法，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 資料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。 這樣做將協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：透過資料驗證追蹤改善可追蹤性

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 建置資料湖的資料譜系

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？

實作多重控制來保護靜態資料，以降低未經授權存取或處理不當的風險。

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 強制執行存取控制

SEC08-BP01 實作安全金鑰管理

安全金鑰管理包括儲存、輪換、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換。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，以安全儲存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 ） 為 AWS KMS 金鑰提供耐用、安全且備援的儲存。許多 AWS 服務與 整合 AWS KMS以支援資料加密。 AWS KMS 會使用 FIPS 140-2 第 3 級已驗證的硬體安全模組來保護您的金鑰。沒有以純文字匯出 AWS KMS 金鑰的機制。

使用多帳戶策略部署工作負載時，最佳實務是將 AWS KMS 金鑰保留在與使用它們的工作負載相同的帳戶中。在此分散式模型中，管理 AWS KMS 金鑰的責任由應用程式團隊承擔。在其他使用案例中，組織可以選擇將 AWS KMS 金鑰存放到集中式帳戶。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。

無論金鑰材料存放在何處，都應透過使用金鑰政策和政策來嚴格控制對金鑰的存取。 IAM金鑰政策是控制 AWS KMS 金鑰存取的主要方式。此外， AWS KMS 金鑰授予可以提供 AWS 服務的存取權，以代表您加密和解密資料。請花時間檢閱對AWS KMS 金鑰 進行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 AWS 中存放的受管金鑰和客戶受管金鑰執行的操作 AWS KMS 可以登入 AWS CloudTrail ，並應定期檢閱。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。在 中刪除金鑰的嘗試 AWS KMS 會受到等待期 的限制，此期間預設為 30 天，讓管理員有時間檢閱這些動作，並在必要時復原請求。

大多數 AWS 服務 AWS KMS 使用的方式都對您透明，您唯一的要求是決定是否使用 AWS 受管金鑰或客戶受管金鑰。如果您的工作負載需要直接使用 AWS KMS 來加密或解密資料，最佳實務是使用信封加密來保護您的資料。AWS 加密SDK可以為您的應用程式提供用戶端加密基本概念，以實作信封加密並與 整合 AWS KMS。

實作步驟

1. 判斷金鑰的適當金鑰管理選項 （ AWS 受管或客戶受管 ）。
    - 為了方便使用， 為大多數 服務 AWS 提供 AWS 擁有和管理的 AWS 金鑰，提供 encryption-at-rest 功能，而無需管理金鑰材料或金鑰政策。
    - 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 如何與 服務 AWS KMS 整合。例如，EC2執行個體可以使用加密的EBS磁碟區，驗證從這些磁碟區建立的 Amazon EBS快照也會使用客戶受管金鑰加密，並減少意外揭露未加密的快照資料。
    - AWS 服務使用方式 AWS KMS

- 如需 AWS 服務提供的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：可讓您 AWS KMS 輕鬆建立和管理金鑰，並控制在各種 AWS 服務和應用程式中使用加密。
   - 入門： AWS Key Management Service （ AWS KMS ）
   - 檢閱對 AWS KMS 金鑰 進行存取控制的最佳實務。
4. 考慮 AWS Encryption SDK：當您的應用程式需要加密資料用戶端時，請使用 AWS Encryption SDK 搭配 AWS KMS 整合。
   - AWS Encryption SDK
5. 讓 IAM Access Analyzer 自動檢閱並通知是否有過於廣泛的 AWS KMS 金鑰政策。
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. 判斷適合您 AWS KMS 金鑰的記錄層級。由於記錄了對 的呼叫 AWS KMS，包括唯讀事件，因此與相關聯的 CloudTrail 日誌 AWS KMS 可能會變得大量。
   - 有些組織偏好將 AWS KMS 記錄活動隔離為單獨的追蹤。如需詳細資訊，請參閱 AWS KMS 開發人員指南中的使用 記錄 AWS KMS API通話 CloudTrail一節。

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
- 信封加密
- 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 密碼編譯詳細資訊

相關影片：

- Encryption 如何在 中運作 AWS
- 在 上保護您的區塊儲存 AWS

- AWS 資料保護：使用鎖定、金鑰、簽章和憑證

相關範例：

- 使用 實作進階存取控制機制 AWS KMS

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：

- 不使用 encrypt-by-default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曝險等級：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 了解您的資料分類方案)。

AWS Key Management Service （ AWS KMS ） 與許多 AWS 服務整合，讓您更輕鬆地加密靜態資料。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。使用 時 AWS KMS，請考慮需要多嚴格地限制資料。預設和服務控制 AWS KMS 金鑰由 代您管理及使用 AWS。對於需要精細存取基礎加密金鑰的敏感資料，請考慮客戶受管金鑰 （ CMKs ）。您可以完全控制 CMKs，包括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 透過設定預設加密來支援加密的強制執行。您可以使用 AWS Config 規則 自動檢查您是否使用加密，例如 Amazon Elastic Block

Store （ Amazon EBS ） 磁碟區 、Amazon Relational Database Service （ Amazon RDS ） 執行個體 和 Amazon S3 儲存貯體 。

AWS 也提供用戶端加密的選項，可讓您在資料上傳至雲端之前對其進行加密。 AWS Encryption SDK 提供使用信封加密 加密資料的方法。您會提供包裝金鑰，而 會為其加密的每個資料物件 AWS Encryption SDK 產生唯一的資料金鑰。考慮您 AWS CloudHSM 是否需要受管單一租戶硬體安全模組（ HSM ）。 AWS CloudHSM 可讓您在經過驗證FIPS的 140-2 層級 3 上產生、匯入和管理密碼編譯金鑰HSM。的一些使用案例 AWS CloudHSM 包括保護私有金鑰以發出憑證授權機構 （ CA ），以及開啟 Oracle 資料庫的透明資料加密 （ TDE ）。 AWS CloudHSM 用戶端SDK提供軟體，可讓您在將資料上傳至 AWS CloudHSM 之前，使用內部存放的金鑰加密資料用戶端 AWS。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。

  設定新 Amazon EBS磁碟區的預設加密：指定您希望以加密形式建立所有新建立的 Amazon EBS磁碟區，並可選擇使用 提供的預設金鑰 AWS 或您建立的金鑰。

  設定加密的 Amazon Machine Images （ AMIs ）：複製已設定加密AMI的現有 會自動加密根磁碟區和快照。

  設定 Amazon RDS加密 ：使用加密選項設定 Amazon RDS 資料庫叢集和靜態快照的加密。

  建立和設定具有政策的 AWS KMS 金鑰，以限制對每個資料分類的適當主體的存取：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。您也可以提供對其他的金鑰存取權 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CMK的政策，讓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 AWS 其他服務中設定加密：針對 AWS 您使用的其他服務，請檢閱該服務的安全文件，以確定服務的加密選項。

資源

相關文件：

- AWS 加密工具
- AWS Encryption SDK

- AWS KMS 密碼編譯詳細資訊白皮書
- 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
- 如何啟用 Amazon S3 儲存貯體的預設加密？
- 使用加密保護 Amazon S3 資料

相關影片：

- Encryption 如何在 中運作 AWS
- 在 上保護您的區塊儲存 AWS

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。 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，以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。

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。這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曝險等級：中

實作指引

自動化是貫穿保護靜態資料的實務的主題。SEC01-BP06 自動化部署標準安全控制項描述如何使用基礎設施作為程式碼 IaC ）範本來擷取資源的組態，例如使用 AWS CloudFormation。 這些範本會傳送至版本控制系統，並用於 AWS 透過 CI/CD 管道在 上部署資源。 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。 您可以使用 監控 或其他 IaC 工具中 CloudFormation尚未可用的設定，以避免組態錯誤AWS Config。 Config 為錯誤組態產生的警示可以自動修復，如 SEC04-BP04 所述，針對不合規資源啟動修復。

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 授予最低權限存取和 SEC03-BP04 減少許可持續描述設定最低權限存取政策，這些政策由 持續監控AWS Identity and Access Management Access Analyzer，以便在許可可以減少時產生調查結果。 除了監控許可的自動化之外，您還可以設定 Amazon GuardDuty 來監控磁碟EBS區 （透過EC2執行個體）、S3 儲存貯體 和支援的 Amazon Relational Database Service 資料庫 的異常資料存取行為。

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 Automate 識別和分類說明 Amazon Macie 如何監控 S3 儲存貯體是否有非預期的敏感資料，並產生可啟動自動回應的警示。

遵循 REL09 備份資料中的實務，開發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. 您可以使用 作為 IaC 範本，以及 CloudFormationGuard 來檢查範本是否組態錯誤。
   b. 使用 AWS Config 在主動評估模式下執行規則。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提醒，以調用修復動作。
   b. 如需自動化SEC修復的詳細資訊，請參閱 04-BP04 啟動不合規資源的修復。
3. 透過自動化持續監控並減少資料存取許可。
   a. IAM 當許可可能降低時，Access Analyzer 可以持續執行以產生提醒。

4. 監控並提醒異常資料存取行為。
    a. GuardDuty 同時監控磁碟EBS區、S3 儲存貯體和RDS資料庫等資料儲存資源的已知威脅簽章和與基準存取行為的偏差。
5. 監控並在敏感資料儲存於非預期位置時發出提醒。
    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 是一項受管服務，可在 上建立各種資料來源的加密和安全備份 AWS。 Elastic Disaster Recovery 可讓您複製完整的伺服器工作負載，並透過以秒為單位測量的復原點目標（RPO）來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 自動化標準安全控制的部署
- SEC03-BP02 授予最低權限存取
- SEC03-BP04 持續減少許可
- SEC04-BP04 啟動不合規資源的修復
- SEC07-BP03 自動化識別和分類
- REL09-BP02 保護和加密備份
- REL09-BP03 自動執行資料備份

相關文件：

- AWS 規範指南：自動加密現有和新的 Amazon EBS磁碟區
- AWS 使用NIST網路安全架構的勒索軟體風險管理（CSF）

相關範例：

- 如何使用 AWS Config 主動規則和 AWS CloudFormation Hooks 來防止建立不合規的雲端資源
- 使用 自動化並集中管理 Amazon S3 的資料保護 AWS Backup
- AWS re：Invent 2023 - 使用 Amazon EBS快照實作主動資料保護

- AWS re:Invent 2022 - 利用現代化資料保護建置並自動化以強化恢復能力

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 強制執行存取控制

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。

預期結果：確認只有授權使用者才能根據基準 need-to-know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儲存在一起。
- 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曝險等級：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 Object Lock 為 Amazon S3 中的物件提供強制存取控制，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：強制執行最低權限存取控制，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。
- 檢閱 AWS Key Management Service（AWS KMS）政策 ：檢閱政策中授予的存取層級。 AWS KMS
- 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。請考慮使用 AWS Config 來偵測可公開取得的儲存貯體，以及使用 Amazon CloudFront 來提供 Amazon S3 的內容。確認不允許公開存取的儲存貯體已正確設定為防止公開存取。依預設，所有 S3 儲存貯體皆為私有，只有明確獲得存取權的使用者才能存取。
- 使用 AWS IAM Access Analyzer ：IAMAccess Analyzer 會分析 Amazon S3 儲存貯體，並在 S3 政策授予外部實體存取權時產生調查結果。
- 適當時，使用 Amazon S3 版本控制和物件鎖定。
- 使用 Amazon S3 庫存清單：Amazon S3 庫存清單可用來稽核和報告 S3 物件的複寫和加密狀態。
- 檢閱 Amazon EBS 和AMI共用許可：共用許可可以允許與工作負載 AWS 帳戶 外部的影像和磁碟區共用。
- 檢閱 AWS Resource Access Manager 定期共用以確定是否應該持續共用資源。Resource Access Manager 可讓您在 Amazon 內共用資源，例如 AWS Network Firewall 政策、Amazon Route 53 解析器規則和子網路VPCs。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 定義存取要求
- SEC03-BP02 授予最低權限存取

相關文件：

- AWS KMS 密碼編譯詳細資訊白皮書
- 管理對 Amazon S3 資源的存取權限的簡介
- 管理 AWS KMS 資源存取權的概觀
- AWS Config 規則
- Amazon S3 + Amazon CloudFront：雲端中的配對
- 使用版本控制
- 使用 Amazon S3 Object Lock 鎖定物件
- 共用 Amazon EBS 快照
- 已共用 AMIs
- 在 Amazon S3 上託管單頁應用程式

相關影片：

- 在 上保護您的區塊儲存 AWS

SEC 9. 如何保護傳輸中的資料？

實作多重控制來保護傳輸中的資料，以降低未經授權存取或遺失的風險。

最佳實務

- SEC09-BP01 實作安全金鑰和憑證管理
- SEC09-BP02 在傳輸中強制執行加密
- 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） 憑證用於保護網路通訊，並透過網際網路以及私有網路建立網站、資源和工作負載的身分。

預期結果：安全憑證管理系統，可在公有金鑰基礎設施 （ ） 中佈建、部署、存放和更新憑證PKI。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，忽略憑證授權單位 (CA) 階層。
- 針對公有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
- 增加憑證授權機構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

現代工作負載會使用 等PKI通訊協定廣泛使用加密網路通訊TLS。PKI 憑證管理可能很複雜，但自動憑證佈建、部署和續約可以減少與憑證管理相關聯的摩擦。

AWS 提供兩種 服務來管理一般用途PKI憑證： AWS Certificate Manager和 AWS Private Certificate Authority （ AWS Private CA ）。ACM 是客戶用來佈建、管理和部署憑證的主要服務，可用於面向公有和私有 AWS 工作負載。ACM 使用 發行憑證， AWS Private CA 並與許多其他 受管服務整合，為工作負載提供安全TLS憑證。 AWS

AWS Private CA 可讓您建立自己的根或下級憑證授權機構，並透過 發行TLS憑證API。您可以在控制和管理TLS連線用戶端的信任鏈的情況下使用這些類型的憑證。除了TLS使用案例之外， AWS Private CA 還可以用來向 Kubernetes Pod、Matter 裝置產品證明、程式碼簽署，以及其他具有自訂範本 的使用案例發出憑證。您也可以使用 IAM Roles Anywhere 為已由 Private CA 簽署的 X.509 憑證發行的內部部署工作負載提供臨時IAM憑證。

除了 ACM和 之外 AWS Private CA， AWS IoT Core還提供專門的支援，以佈建、管理和將PKI憑證部署至 IoT 裝置。 AWS IoT Core 提供專門的機制，以將 IoT 裝置大規模加入您的公有金鑰基礎設施。

建立私有 CA 階層時的考量

在需要建立私有 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 AWS 帳戶 時，最佳實務是將 CA 階層的每個層級部署到個別。此刻意步驟可減少 CA 階層中每個層級的表面積，因此更輕鬆地探索日誌資料中的 CloudTrail異常，並在未經授權存取其中一個帳戶時減少存取範圍或影響。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。

然後，CAs在與根 CA 帳戶分開的帳戶中建立一或多個中繼，為最終使用者、裝置或其他工作負載發出憑證。最後，從您的根 CA 向中繼 發出憑證CAs，這會繼而向最終使用者或裝置發出憑證。如需規劃 CA 部署和設計 CA 階層的詳細資訊，包括規劃彈性、跨區域複寫、CAs跨組織共用等，請參閱規劃 AWS Private CA 部署。

實作步驟

1. 判斷您的使用案例所需的相關 AWS 服務：
   - 許多使用案例可以使用 來利用現有的 AWS 公有金鑰基礎設施AWS Certificate Manager。ACM 可用於部署 Web 伺服器、負載平衡器的TLS憑證，或公開信任憑證的其他用途。
   - 當您需要建立自己的私有憑證授權機構階層或需要存取可匯出的憑證時，考慮 AWS Private CA。ACM 然後， 可用來使用 發行許多類型的最終實體憑證 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. 盡可能實作自動憑證續約：
   - 針對 發行的憑證，ACM搭配整合的ACM受管服務使用受管續約。 AWS
3. 建立日誌記錄和稽核記錄：
   - 啟用CloudTrail日誌以追蹤持有憑證授權單位的帳戶存取權。請考慮在 中設定日誌檔案完整性驗證 CloudTrail ，以驗證日誌資料的真實性。
   - 定期產出並檢閱稽核報告，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 儲存貯體來存放憑證撤銷清單 （CRL）。如需根據您的工作負載需求設定此 S3 儲存貯體的指引，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 中託管和管理整個私有憑證基礎設施 AWS
- 如何保護適用於汽車和製造的企業規模ACM私有 CA 階層

- 私有 CA 最佳實務
- 如何使用 AWS RAM 來共用您的 ACM Private CA 跨帳戶

相關影片：

- 啟動 AWS Certificate Manager 私有 CA (研討會)

相關範例：

- 私有 CA 研討會
- IOT 裝置管理研討會（包括裝置佈建）

相關工具：

- 要使用的 Kubernetes cert-manager 外掛程式 AWS Private CA

SEC09-BP02 在傳輸中強制執行加密

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，以符合組織、法律和合規上的要求。在虛擬私有雲端（）之外傳輸敏感資料時，僅使用具有加密的通訊協定VPC。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。

預期結果：所有資料都應使用安全TLS通訊協定和密碼套件在傳輸過程中加密。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。應TLS盡可能使用 加密僅限內部 AWS 環境中的網路流量。 AWS 內部網路預設會加密，除非未經授權方已存取正在產生流量的任何資源（例如Amazon EC2執行個體和 Amazon ECS容器），否則 內的網路流量VPC將無法被欺騙或探查。考慮使用IPsec虛擬私有網路（）保護 network-to-network流量VPN。

常見的反模式：

- 使用已棄用版本的 SSL、 TLS和 密碼套件元件（例如 v3SSL.0、1024 位元RSA金鑰和 RC4 密碼）。
- 允許未加密的（HTTP）流量進出面向公有的資源。
- 未監控 X.509 憑證並在到期前更換。
- 針對 使用自我簽署的 X.509 憑證TLS。

未建立此最佳實務時的曝險等級：高

實作指引

AWS 服務提供使用 進行通訊TLS的HTTPS端點，在與 通訊時提供傳輸中的加密 AWS APIs。類似 等不安全通訊協定HTTP可以透過使用安全群組VPC在 中稽核和封鎖。HTTP 也可以在 Amazon CloudFront 或 Application Load Balancer 中自動重新導向至 HTTPS 的 請求。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，您可以使用VPC從外部網路VPN連線至您的 ，或AWS Direct Connect促進流量加密。確認您的用戶端至少使用 TLS 1.2 呼叫 AWS APIs ，因為AWS 在 2023 年 TLS6 月已棄用舊版 。 AWS 建議使用 TLS 1.3。 AWS Marketplace 如果您有特殊需求，中會提供第三方解決方案。

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，將安全群組設定為僅允許對應用程式負載平衡器或 Amazon EC2執行個體的HTTPS通訊協定。
- 在邊緣服務中設定安全通訊協定：HTTPS使用 Amazon 設定 CloudFront ，並使用適合您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 point-to-point 或 network-to-network 連線，以協助提供資料隱私權和完整性。
- 在負載平衡器中設定安全協定：選取安全政策，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 中設定安全通訊協定：將叢集設定為需要安全通訊端層 （ SSL ） 或傳輸層安全 （ TLS ） 連線 。
- 設定安全通訊協定：檢閱 AWS 服務文件以判斷 encryption-in-transit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、管理和部署公有TLS憑證，以搭配 AWS 服務使用。
- 考慮AWS Private Certificate Authority針對私有PKI需求使用 ： AWS Private CA 允許您建立私有憑證授權機構 （ CA ） 階層，以發行可用於建立加密TLS頻道的終端實體 X.509 憑證。

資源

相關文件：

- HTTPS搭配 使用 CloudFront
- 使用 將 VPC連線至遠端網路 AWS Virtual Private Network

- 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 加密資料庫執行個體的連線
- 設定連線的安全選項

SEC09-BP03 驗證網路通訊

使用支援身分驗證的通訊協定來驗證通訊的身分，例如 Transport Layer Security （ TLS ） 或 IPsec。

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、經驗證的網路協定。使用支援驗證和授權的網路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流程。在技術允許的情況下，流量流程要使用經過驗證和加密的網路協定。

常見的反模式：

- 工作負載內有未經加密或驗證的流量流程。
- 在多個使用者或實體之間重複使用驗證憑證。
- 僅仰賴網路控制作為存取控制機制。
- 建立自訂驗證機制，而非仰賴產業標準的驗證機制。
- 服務元件或 中的其他資源之間流動過度寬鬆的流量VPC。

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、日誌記錄和事件回應。
- 將網路控制項與身分驗證和授權控制項結合，為您的工作負載提供 defense-in-depth。

未建立此最佳實務時的曝險等級：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流程。
- 南北流量代表工作負載和取用者之間的流量流程。

雖然加密南北流量是常見的做法，但是使用經過驗證的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，最佳實務仍是對這些服務之間的通訊進行加密、驗證和授權。

例如， AWS 服務APIs使用 AWS Signature 第 4 版 （ SigV4 ） 簽章通訊協定來驗證來電者，無論請求來自哪個網路。此身分驗證可確保 AWS APIs可以驗證請求動作的身分，然後該身分可以與政策結合，以做出授權決定，以判斷是否應允許該動作。

Amazon VPC Lattice 和 Amazon API Gateway 等服務可讓您使用相同的 SigV4 簽章通訊協定，將身分驗證和授權新增至您工作負載中的東西流量。如果 AWS 環境外的資源需要與需要 SigV4-based 身分驗證和授權的服務通訊，您可以在非AWS 資源上使用 AWS Identity and Access Management （ IAM ） Roles Anywhere 來取得臨時 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。

驗證東西流量的另一個常見機制是TLS相互身分驗證 （ m TLS ）。許多物聯網 IoT ） business-to-business、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通訊的兩側身分。這些憑證可由 AWS Private Certificate Authority （ ） 發出AWS Private CA。您可以使用 Amazon API Gateway 等服務AWS App Mesh，並為工作負載間或內部通訊提供 mTLS 身分驗證。雖然 mTLS 提供TLS通訊兩側的身分驗證資訊，但它不會提供授權機制。

最後，2.0 OAuth 和 OpenID Connect （ OIDC ） 是兩種通訊協定，通常用於控制使用者對 服務的存取，但現在也越來越受流量歡迎 service-to-service。API Gateway JSON 提供 Web 權杖 （ JWT ） 授權方 ，允許工作負載使用從 OIDC或 2.0 身分提供者JWTs發行的 來限制對API路由OAuth的存取。OAuth2 範圍可以用作基本授權決策的來源，但授權檢查仍需要在應用程式層中實作，且僅OAuth2範圍無法支援更複雜的授權需求。

實作步驟

- 定義並記錄工作負載網路流程：實作 defense-in-depth策略的第一步是定義工作負載的流量流程。
  - 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時期的行為。
  - 執行威脅模型練習時，資料流程圖也很有用，如 SEC01-BP07 所述使用威脅模型 識別威脅並排定緩解優先順序。

- 建立網路控制：考慮建立與資料流程一致的網路控制 AWS 功能。雖然網路邊界不應是唯一的安全控制，但它們在策略中提供 defense-in-depth一層來保護工作負載。
  - 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 與支援 AWS 的第三方服務進行AWS PrivateLink通訊 AWS PrivateLink。透過 AWS PrivateLink 介面端點傳送的資料會保留在 AWS 網路骨幹內，而不會周遊公有網際網路。
- 在您的工作負載中跨服務實作身分驗證和授權：選擇最適合的一組 AWS 服務，以便在工作負載中提供已驗證的加密流量流程。
  - 考慮使用 Amazon VPC Lattice 來保護 service-to-service通訊。VPC Lattice 可以使用 SigV4 身分驗證結合身分驗證政策來控制 service-to-service存取。
  - 對於 service-to-service使用 m 的通訊TLS，請考慮 API Gateway 或 App Mesh 。 AWS Private CA 可用來建立能夠發出憑證以搭配 m 使用的私有 CA 階層TLS。
  - 使用 OAuth 2.0 或 與 服務整合時OIDC，請考慮API使用 JWT授權方 的 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 來管理對服務的存取，請考慮啟用和監控 VPC Lattice 存取日誌。這些存取日誌包含請求實體的相關資訊、包括來源和目的地 的網路資訊VPC，以及請求中繼資料。
  - 考慮啟用VPC流程日誌來擷取網路流程上的中繼資料，並定期檢閱異常情況。
  - 請參閱AWS 安全事件回應指南和 AWS Well-Architected Framework 安全支柱的事件回應區段，以取得規劃、模擬和回應安全事件的更多指引。

資源

相關的最佳實務：

- SEC03-BP07 分析公有和跨帳戶存取
- SEC02-BP02 使用臨時憑證
- SEC01-BP07 使用威脅模型識別威脅並排定緩解的優先順序

相關文件：

- 評估存取控制方法以保護 Amazon API Gateway APIs
- 設定 的相互TLS身分驗證 REST API

- 如何使用JWT授權方保護API閘道HTTP端點
- 使用 AWS IoT Core 憑證提供者授權對 AWS 服務的直接呼叫
- AWS 安全事件回應指南

相關影片：

- AWS re：invent 2022：簡介 VPC Lattice
- AWS re：invent 2020：適用於 HTTPAPIs的無伺服器API身分驗證 AWS

相關範例：

- Amazon VPC Lattice 研討會
- 零信任第 1 集 - Phantom Service Perimeter 研討會

事件回應

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 識別關鍵人員和外部資源
- SEC10-BP02 制定事件管理計劃
- SEC10-BP03 準備鑑識功能
- SEC10-BP04 開發和測試安全事件回應教戰手冊
- SEC10-BP05 佈建前存取
- SEC10-BP06 部署前工具

- SEC10-BP07 執行模擬
- SEC10-BP08 建立從事件中學習的架構

SEC10-BP01 識別關鍵人員和外部資源

識別可以協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。您可定期審查並更新此資訊，以在內部和外部工具中反映人員變更。在記錄此資訊時，您會考慮所有第三方服務供應商和供應商，包括安全合作夥伴、雲端供應商和 software-as-a-service（SaaS ）應用程式。SaaS 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，定期審查並更新此資訊。這對於事件管理者、事件回應者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授權。
- 事件回應者：事件回應者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（SMEs）：對於分散式和自主團隊，我們建議您SME為任務關鍵工作負載識別。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、定義回應計畫、自動排定待命時間表，以及制定向上呈報計畫。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示閾值。

識別外部合作夥伴：企業使用獨立軟體供應商（ISVs）、合作夥伴和分包商建置的工具，為客戶建立差異化解決方案。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的 層級， AWS Support 以便透過支援案例立即存取 AWS 主題專家。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。維護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. 為每個聯絡人（例如 SMS、電話或電子郵件）定義至少兩種類型的聯絡管道，以確保事件期間的可連線性。
3. 定義回應計畫。
   a. 識別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) 的向上呈報計畫。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 操作活動已識別負責其績效的擁有者

相關文件：

- AWS 安全事件回應指南

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 制定事件管理計劃

為事件回應制定的第一份文件是事件回應計畫。事件回應計畫應是您事件回應計畫和策略的基礎。

建立此最佳實務的優勢：開發全面且明確定義的事件回應程序，是成功且可擴展的事件回應計畫的關鍵。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回應程序。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回應程序。

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，可及時識別、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回應及復原策略納入考量。例如，如果您在 中操作符合美國聯邦 AWS 法規RAMP的工作負載，則遵守 NIST SP 800-61 電腦安全處理指南 很有用。同樣地，在操作具有歐洲個人身分識別資訊（PII）資料的工作負載時，請考慮以下案例：根據歐盟一般資料保護法規（GDPR）規定，您可以如何保護和回應與資料駐留相關的問題。

在 中為工作負載建立事件管理計劃時 AWS，請從AWS 共同責任模型開始， defense-in-depth以建立事件回應的方法。在此模型中， 會 AWS 管理雲端的安全性，而您要負責雲端的安全。此表示您保有控制權，並對您選擇實作的安全控制項負責。AWS 安全事件回應指南詳細說明在建置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，以與您的雲端維運目標保持同步。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。雖然為您的安全回應計劃建立負責任、負責、諮詢和知情的（RACI）圖表似乎很官僚，但這樣做可以促進快速和直接的溝通，並清楚地概述事件不同階段的領導。

在事件期間，包括受影響應用程式和資源的擁有者和開發人員，都是關鍵，因為他們是主題專家（SMEs），可以提供資訊和內容以協助測量影響。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回應之前，請務必先與他們建立關係。應用程式擁有者或 SMEs，例如您的雲端管理員或工程師，可能需要在環境不熟悉或複雜，或回應者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協助。

了解 AWS 回應團隊和支援

- AWS Support
  - AWS Support 提供各種計劃，讓您存取支援 AWS 解決方案成功和運作運作狀態的工具和專業知識。如果您需要技術支援和更多資源來協助規劃、部署和最佳化 AWS 環境，您可以選取最符合您 AWS 使用案例的支援計畫。
  - 將 中的支援中心 AWS Management Console（需要登入）視為中心聯絡點，以取得影響 AWS 資源問題的支援。對 的存取由 AWS Support 控制 AWS Identity and Access Management。如需有關存取 AWS Support 功能的詳細資訊，請參閱 入門 AWS Support。
- AWS 客戶事件回應團隊（CIRT）
  - AWS 客戶事件回應團隊（CIRT）是一個專門的 24 小時全年無休全球 AWS 團隊，在 AWS 共同責任模型 的客戶端的作用中安全事件期間為客戶提供支援。
  - 當 AWS CIRT支援您時，它們會協助分類和復原 上的作用中安全事件 AWS。他們可以使用 AWS 服務日誌協助進行根本原因分析，並提供復原建議。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 客戶可以透過 案例 AWS CIRT與 互動。 AWS Support
- DDoS 回應支援
  - AWS 提供 AWS Shield，其提供受管分散式拒絕服務（DDoS）保護服務，可保護在 上執行的 Web 應用程式 AWS。Shield 提供永遠在線的偵測和自動內嵌緩解功能，可將應用程式停機時間和延遲降至最低，因此不需要為了從DDoS保護中受益 AWS Support 而進行互動。Shield： AWS

Shield Standard 和 有兩個層級 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 ） 會持續管理 AWS 基礎設施，以便您可以專注於應用程式。透過實作最佳實務來維護您的基礎設施， AMS 有助於降低您的營運開銷和風險。AMS 自動化常見的活動，例如變更請求、監控、修補管理、安全性和備份服務，並提供全生命週期服務來佈建、執行和支援您的基礎設施。
  - AMS 負責部署安全偵測控制套件，並提供 24 小時全年無休的第一道警示回應。啟動警示時，AMS會遵循一組標準自動和手動劇本來驗證一致的回應。這些教戰手冊會在上線期間AMS與客戶共用，以便他們可以與 開發和協調回應AMS。

制定事件回應計畫

事件回應計畫應是您事件回應計畫和策略的基礎。事件回應計畫應納入正式文件中。事件回應計畫通常包含下列章節：

- 事件回應團隊概觀：概述事件回應團隊的目標和職能。
- 角色和責任：列出事件回應利害關係人，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。
- 備份通訊方法：將 out-of-band通訊作為事件通訊的備份是最佳實務。提供安全 out-of-band通訊管道的應用程式範例為 AWS Wickr。
- 事件回應的階段和應採取的行動：列舉事件回應的階段 (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，但每個組織的事件回應計畫都是獨一無二的。您必須建立最適合貴組織的事件回應計畫。

資源

相關的最佳實務：

- SEC04 （ 如何偵測和調查安全事件？ ）

相關文件：

- AWS 安全事件回應指南
- NIST：電腦安全事件處理指南

SEC10-BP03 準備鑑識功能

在安全事件發生之前，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。如需在 中開始建置鑑識能力的重要資訊 AWS 雲端，請參閱 中的鑑識調查環境策略 AWS 雲端。

設定取證的環境和 AWS 帳戶 結構後，請定義在四個階段有效執行取證合理方法所需的技術：

- 集合：收集相關 AWS 日誌，例如 AWS CloudTrail AWS Config、、VPCFlow Logs 和主機層級日誌。收集可用時受影響 AWS 資源的快照、備份和記憶體傾印。
- 檢查：檢查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，以了解事件並從中得出結論。
- 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 可協助您在資源成長和擴展時集中管理和管理 AWS 環境 AWS 。 AWS 組織會合併您的 ， AWS 帳戶 以便您以單一單位管理它們。您可以使用組織單位 （ OUs ） 將帳戶分組在一起，以單一單位進行管理。

對於事件回應，擁有支援事件回應功能的 AWS 帳戶 結構很有幫助，其中包括安全 OU 和鑑識 OU。在安全性 OU 中，您應該擁有下列項目的帳戶：

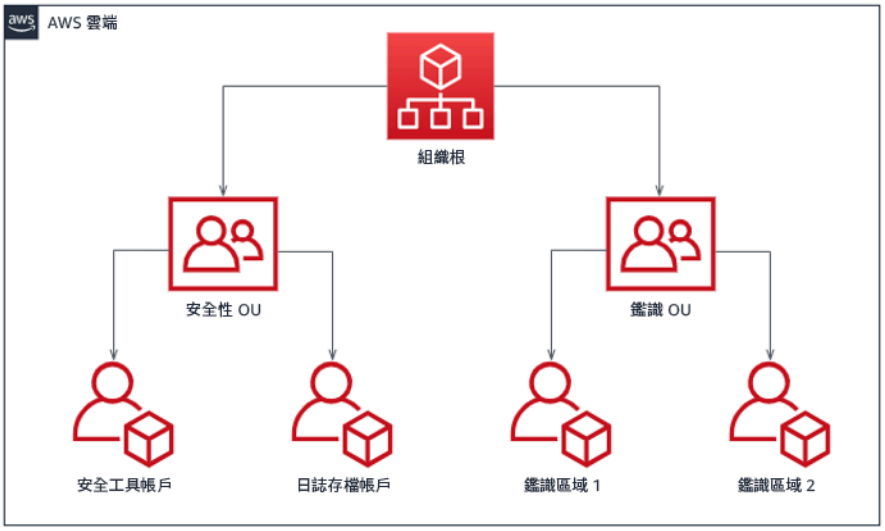
- 日誌封存：在許可 AWS 帳戶 有限的日誌封存中彙總日誌。
- 安全工具：在安全工具 中集中安全服務 AWS 帳戶。此帳戶會以安全性服務的委派系統管理員身分運作。

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，則可以封鎖在該區域之外建立 AWS 資源，並降低資源複製到非預期區域的風險。例如，如果您只在美國東部 (維吉尼亞北部) 區域 (us-east-1) 和

美國西部 (奧勒岡) (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 AWS 帳戶 多個區域建立鑑識。在將 AWS 資源複製到該帳戶時，您應該小心謹慎，以確認您是否符合資料主權要求。佈建新帳戶需要一些時間，因此必須在事件之前建立和檢測鑑識帳戶，以便回應者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：



事件回應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，對於從安全事件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。在 上 AWS，您可以拍攝各種資源的快照。快照可為您提供 point-in-time這些資源的備份。有許多 AWS 服務可以支援您進行備份和復原。如需有關這些備份和復原之服務和方法的詳細資訊，請參閱備份和復原方案指引和使用備份從安全事件中復原。

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。如需有關保護備份的指引，請參閱在 AWS中保護備份的 10 大安全最佳實務。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，您的事件回應團隊必須能夠快速收集和分析證據，同時在事件周圍的期間內維持準確性 （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）。事件回應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 為鑑識提供了許多自動化資源，這些資源列於下列資源區段中。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 安全事件回應指南 - 開發鑑識能力
- AWS 安全事件回應指南 - 鑑識資源
- 中的鑑識調查環境策略 AWS 雲端
- 如何在 中自動化鑑識磁碟收集 AWS
- AWS 規範指南 - 自動化事件回應和鑑識

相關影片：

- 自動化事件回應和鑑識

相關範例：

- 自動化事件回應和鑑識架構
- 適用於 Amazon 的自動 Forensics Orchestrator EC2

SEC10-BP04 開發和測試安全事件回應教戰手冊

準備事件回應流程的關鍵部分是制定程序手冊。事件回應程序手冊提供一系列方案指引和安全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

應針對事件案例建立程序手冊，例如：

- 預期事件：應針對您預期的事件建立程序手冊。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警示 ：應該為已知的安全調查結果和警示建立 Playbook，例如 GuardDuty 調查結果。您可能會收到 GuardDuty 調查結果並思考：「現在什麼？」 為了防止錯誤處理或忽略

GuardDuty 調查結果，請為每個潛在 GuardDuty 調查結果建立行動手冊。某些修復詳細資訊和指引可在GuardDuty 文件中找到。值得注意的是，預設情況下 GuardDuty 未啟用 ，且確實會產生成本。如需 的詳細資訊 GuardDuty，請參閱附錄 A：雲端功能定義 - 可見性和警示 。

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件。

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件？ 程序手冊的目標是什麼？
- 先決條件：此事件案例需要哪些日誌、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：在事件回應的各個階段，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：事件的偵測方式為何？
  - 分析：判斷影響範圍的方式為何？
  - 包含：隔離事件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 - 開發事件管理計劃

相關文件：

- 事件回應程序手冊的架構
- 制定您自己的事件回應程序手冊
- 事件回應程序手冊範例
- 使用 Jupyter 手冊和 CloudTrail Lake 建置 AWS 事件回應 Runbook

SEC10-BP05 佈建前存取

確認事件回應者已預先佈建正確的存取權 AWS ，以減少調查到復原所需的時間。

常見的反模式：

- 使用事件回應的根帳戶。
- 更改現有的帳戶。
- 在提供 just-in-time權限提升時直接控制IAM許可。

未建立此最佳實務時的曝險等級：中

實作指引

AWS 建議盡可能減少或消除對長期憑證的依賴，以偏好臨時憑證和just-in-time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，以及事件回應任務，我們建議您實作聯合身分以及適用於管理存取權的暫時權限提升。在此模型中，使用者會請求提升至較高層級的權限(例如事件回應角色)；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暫時 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回應情境中，使用暫時權限提升。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。
- 惡意活動，例如分散式拒絕服務 （ DDoS ） 事件或導致系統無法使用。

在上述案例中，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使用者、群組或角色來執行任務和存取 AWS 資源。僅將根憑證用於需要根使用者存取權的任務。若要驗證事件回應者對 AWS 和其他相關系統的存取層級是否正確，我們建議預先佈建專用帳戶。此類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畫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政策暫時提升使用者或角色的存取權，兩者都會讓使用者不清楚在事件期間擁有哪些存取權，並可能使升級的許可未被撤銷。

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。若要支援此功能，請建立行動手冊，以確認事件回應使用者是在專用安全帳戶中建立為使用者，而不是透過任何現有的聯合或單一登入（SSO）解決方案進行管理。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）。如果事件回應教戰手冊只需要存取 AWS Management Console，則使用者不應設定存取金鑰，並且應明確禁止建立存取金鑰。這可以使用IAM政策或服務控制政策（SCPs）進行設定，如 AWS 安全最佳實務中所述AWS Organizations SCPs。除了在其他帳戶中擔任事件回應角色的能力外，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。

若要確認事件回應角色的使用可以受到適當的監控和稽核，為此目的建立IAM的帳戶不得在個人之間共用，除非特定任務 需要，否則 AWS 帳戶根使用者 不會使用 。 https://docs.aws.amazon.com/accounts/latest/reference/root-user-tasks.html如果需要根使用者（例如IAM，無法存取特定帳戶），請使用單獨的程序搭配可用的 Playbook，以驗證根使用者登入憑證和MFA權杖的可用性。

若要設定事件回應角色IAM的政策，請考慮使用 IAM Access Analyzer 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回應角色的存取權，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回應角色。您可能想要為每個行動手冊建立單獨的IAM政策，以便更輕鬆地管理和稽核。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回應計畫。

使用事件回應帳戶，IAM在其他 中擔任專用事件回應角色 AWS 帳戶。這些角色必須設定為只能由安全帳戶中的使用者擔任，且信任關係必須要求呼叫主體已使用 進行身分驗證MFA。這些角色必須使用嚴格範圍IAM的政策來控制存取。確保這些角色的所有`AssumeRole`請求都已登入 CloudTrail 並收到提醒，而且使用這些角色採取的任何動作都會記錄下來。

強烈建議IAM帳戶和IAM角色都清楚命名，以便輕鬆地在 CloudTrail 日誌中找到它們。其中一個範例是命名IAM帳戶*<USER_ID>*-BREAK-GLASS和IAM角色 BREAK-GLASS-ROLE。

CloudTrail 用於記錄您 AWS 帳戶中API的活動，並應用於設定事件回應角色 的使用提醒。請參閱部落格貼文，其中會說明使用根金鑰如何設定提醒。您可以修改指示，以 filter-to-filter針對與事件回應IAM角色相關的`AssumeRole`事件設定 Amazon CloudWatch 指標：

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

由於事件回應角色可能具備很高的存取權限，因此必須將這些提醒傳送給多個群組，並據此快速採取行動。

在事件期間，回應者可能需要存取非 直接保護的系統IAM。這些可能包括 Amazon Elastic Compute Cloud 執行個體、Amazon Relational Database Service 資料庫或 software-as-a-service（SaaS ）平台。強烈建議您不要使用原生通訊協定，例如 SSH或 RDP，AWS Systems Manager Session Manager而是用於對 Amazon EC2執行個體的所有管理存取。此存取權可以使用 控制IAM，這是安全且經過稽核的。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。若要存取資料庫和第三方工具，建議您將存取憑證儲存在中， AWS Secrets Manager 並授予事件回應者角色的存取權。

最後，應將事件回應IAM帳戶的管理新增至您的加入者、Moverrs 和 Leavers 會定期處理和檢閱和測試，以確認僅允許預期的存取。

資源

相關文件：

- 管理暫時提升 AWS 對環境的存取
- AWS 安全事件回應指南
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- 為IAM使用者設定帳戶密碼政策
- 在 中使用多重要素驗證（MFA） AWS
- 使用 設定跨帳戶存取 MFA
- 使用 IAM Access Analyzer 產生IAM政策
- 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 受管政策建立精細的工作階段許可

相關影片：

- 自動化 中的事件回應和鑑識 AWS
- DIY Runbook、事件報告和事件回應指南
- 準備和回應 AWS 環境中的安全事件

相關範例：

- Lab： AWS 帳戶設定和根使用者

- 實驗室：使用 AWS 主控台和 進行事件回應 CLI

SEC10-BP06 部署前工具

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函數，您可以使用 的 APIs和 工具的完整集 AWS。您可以將身分管理、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。

如果您的事件回應團隊持續以相同方式回應提醒，可能會形成提醒疲勞的風險。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，或是錯過不尋常的提醒。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 異常偵測，可以減輕常見閾值型警示的負擔。

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件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提醒至關重要。此外，提供搜尋日誌資料之工具的有效方法是 Amazon Detective。

AWS 提供超過 200 個雲端服務和數千種功能。我們建議您檢閱可支援並簡化事件回應策略的服務。

除了日誌記錄之外，您還應該開發和實作標記策略。標記有助於提供有關 AWS 資源用途的背景。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件回應日誌記錄的文件：

- 安全事件回應的日誌記錄策略
- 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、預防性和回應功能，而其他 服務可用於架構自訂安全解決方案。如需安全事件回應最相關的服務清單，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與 AWS 資源相關的內部利益相關者的內文資訊可能很困難。其中一種方法是標籤形式，將中繼資料指派給您的 AWS 資源，並包含使用者定義的金鑰和值。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以加快回應時間，並允許您快速識別和辨別 AWS 資源的上下文資訊，從而將花費在組織內容上的時間降至最低。標籤也可以作為啟動回應自動化的機制。如需有關要標記的內容的詳細資訊，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實作 AWS 資源標記策略 （ SCPs ）。

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 在標準化位置擷取日誌、調查結果和指標

相關文件：

- 安全事件回應的日誌記錄策略
- 事件回應雲端功能定義

相關範例：

- Amazon 和 Amazon Detective 的威脅偵測 GuardDuty 和回應
- Security Hub 研討會
- 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，因此持續審查事件回應能力是很重要的。執行模擬 (也稱為比賽日) 是可用於執行此評估的一種方法。模擬使用真實世界的安全事件案例，旨在模擬

威脅發動者的戰術、技術和程序（TTPs），並允許組織透過回應這些模擬網路事件來練習和評估其事件回應能力，因為這些事件可能實際發生。

建立此最佳實務的優勢：模擬具有多種優勢：

- 驗證網路整備程度和培養事件回應人員的信心。
- 測試工具和工作流程的正確性及效率。
- 根據您的事件回應計畫，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曝險等級：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，涉及各種事件回應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，但事件回應工具或指令碼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件回應人員 (藍隊) 和模擬威脅參與者 (紅隊) 之間的協作層級。藍色團隊由安全操作中心（SOC）的成員組成，但也可以包括實際網路事件期間涉及的其他利益相關者。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。在紫色團隊練習期間，主要重點是偵測機制、工具和支援事件回應工作的標準操作程序（SOPs）。
- 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取模擬類型。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：定義模擬情境與模擬的目標。這兩者都應獲得領導階層的允許。
2. 識別關鍵利害關係人：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. 建置和測試情境：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. 開發動作後報告 （AAR）：識別表現良好的領域、可以使用改善的領域，以及潛在的差距。AAR 應測量模擬的有效性，以及團隊對模擬事件的回應，以便在未來的模擬中追蹤進度。

資源

相關文件：

- AWS 事件回應指南

相關影片：

- AWS GameDay - Security Edition

SEC10-BP08 建立從事件中學習的架構

實作經驗教訓的架構和根本原因分析能力，不僅有助改善事件回應能力，還有助防止事件重複發生。透過學習每個事件，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？

- 如何識別待改善之處？
- 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：

- 事件是什麼？
- 第一次識別事件的時間？
- 事件的識別方式？
- 哪些系統對活動發出提醒？
- 涉及哪些系統、服務和資料？
- 具體發生的事件？
- 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件？
- 在以下幾個領域有哪些可以改善之處：
  - 人物
    - 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件所需的培訓或能力？
    - 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件的流程和程序？
    - 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？
    - 如何減少 time-to-detection 50%？

- 是否需要改善現有提醒，或是需要針對此類事件建立新的提醒？
- 現有的工具是否允許對事件進行有效的調查 (搜尋/分析)？
- 可以做什麼來協助加快這類事件的識別速度？
- 可以做什麼來協助避免這類事件再次發生？
- 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，以便最有效地從事件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件回應流程，文件和利害關係人期望的標準部分。

資源

相關文件：

- AWS 安全事件回應指南 - 建立從事件中學習的架構
- NCSC CAF 指引 - 學到的課程

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 訓練，確保應用程式安全
- SEC11-BP02 自動化整個開發和發行生命週期的測試
- SEC11-BP03 執行定期滲透測試
- SEC11-BP04 手動程式碼檢閱

- SEC11-BP05 集中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 建置在工作負載團隊中嵌入安全擁有權的程式

SEC11-BP01 訓練，確保應用程式安全

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。

為了此最佳實務，安全開發是指正在寫入的軟體，以及支援軟體開發生命週期的工具或系統（SDLC）。

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。
- 未傳達 中採取的決策與組織的整體安全期望或政策有何SDLC關聯。
- 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曝險等級：中

實作指引

提供組織內建置人員的訓練。一開始上威脅建模相關課程，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建模相關課程，奠定良好基礎，並有助於進行考量安全層面的訓練。
- 提供對 AWS 培訓 的存取權和 認證、產業或 AWS 合作夥伴訓練。
- 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需求的自助式指南，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 建置在工作負載團隊中嵌入安全擁有權的程式

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 技能聯盟

相關影片：

- 預防性安全：考量與方法

相關範例：

- 威脅建模相關的研討會
- 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
- Service Catalog
- AWS BugBust

SEC11-BP02 自動化整個開發和發行生命週期的測試

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識別潛在問題，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）會分析您的原始程式碼是否有異常的安全模式，並提供易出現缺陷程式碼的指示。SAST 依賴靜態輸入，例如文件（需求規格、設計文件和設計規格）和應用程式原始程式碼，來測試一系列已知的安全問題。靜態程式碼分析器可協助加快大量程式碼的分析作業。NIST Quality Group 提供 Source Code Security Analyzers 的比較，其中包含 Byte Code Scanners 和 Binary Code Scanners 的開放原始碼工具。

使用動態分析安全測試（DAST）方法來補充靜態測試，該方法針對執行中的應用程式執行測試，以識別潛在的非預期行為。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 在建置器的 中提供程式碼建議，包括安全掃描IDE。Amazon CodeGuru Reviewer 可以在應用程式開發期間識別關鍵問題、安全問題和 hard-to-find錯誤，並提供改善程式碼品質的建議。

Security for Developers 研討會使用 AWS 開發人員工具，例如 AWS CodeCommit、 AWS CodeBuild和 AWS CodePipeline，來發佈包含 SAST 和 DAST測試方法的管道自動化。

隨著 的進展SDLC，請建立迭代程序，其中包含定期與您的安全團隊進行應用程式檢閱。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、程式碼檢閱和包含安全測試的 CI/CD 工具。
- 考慮在 中哪些位置SDLC適合封鎖管道，而不只是通知建置者問題需要修復。
- 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，例如 Amazon CodeWhisperer 與開發人員整合IDEs，以及 Amazon CodeGuru Reviewer 提交掃描程式碼，有助於建置者在正確的時間取得意見回饋。
- 使用 建置 時 AWS Lambda，您可以使用 Amazon Inspector 掃描函數中的應用程式程式碼。
- 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。

- 如果是可能會產生調查結果的安全測試，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調查結果，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
- 監控 Amazon EventBridge 和 Amazon CodeCommit Events 中的 CloudWatch 事件
- Amazon CodeGuru Review 中的秘密偵測
- AWS 透過有效治理加速 上的部署
- 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 治理 （ GitHub ）
- 開發人員安全研討會

SEC11-BP03 執行定期滲透測試

定期對您的軟體進行滲透測試。這項機制有助於識別自動化測試或手動程式碼審查時，未能偵測到的潛在軟體問題。此外還有助於了解偵測控制的效用。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，或是授予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。定期和排定的滲透測試應作為軟體開發生命週期的一部分執行（SDLC）。從滲透測試找到的調查結果應事先解決，才能安排軟體發行。您應該分析從滲透測試得到的調查結果，並識別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。
- 可識別偏好應用程式模式，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，您會執行計畫的安全性缺口情境，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調查結果預期哪些內容，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。
- 分析滲透測試調查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 訓練，確保應用程式安全
- SEC11-BP02 自動化整個開發和發行生命週期的測試

相關文件：

- AWS 滲透測試提供 滲透測試的詳細指引 AWS
- AWS 透過有效治理加速 上的部署
- AWS 安全能力合作夥伴
- 在 上現代化您的滲透測試架構 AWS Fargate
- AWS 故障注入模擬器

相關範例：

- 使用 （ ） 自動化API測試 AWS CodePipeline 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，而且有機會識別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曝險等級：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。您可能正在使用 AWS CodeCommit 或第三方解決方案 GitHub，例如 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 等工具可以更輕鬆地協調程式碼檢閱程序。

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。
- 考慮使用 Amazon CodeGuru Reviewer 來管理和協助程式碼檢閱。
- 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 自動化整個開發和發行生命週期的測試

相關文件：

- 在 AWS CodeCommit 儲存庫中使用提取請求

- 在 中使用核准規則範本 AWS CodeCommit
- 關於在 中提取請求 GitHub
- 使用 Amazon CodeGuru Reviewer 自動化程式碼檢閱
- 使用 Amazon CodeGuru Reviewer 自動偵測 CI/CD 管道中的安全漏洞和錯誤 CLI

相關影片：

- 使用 Amazon 持續改善程式碼品質 CodeGuru

相關範例：

- 開發人員安全研討會

SEC11-BP05 集中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。這使得重複使用的功能實作變得簡單，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。您應該實作一種有效率的方式，在發生更新或出現新需求時將套件新增至儲存庫。例如 AWS CodeArtifact或類似 AWS 合作夥伴解決方案之類的 AWS 服務提供一種交付此功能的方式。

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 自動化整個開發和發行生命週期的測試

相關文件：

- AWS 透過有效的治理加速 上的部署
- 使用套件原始伺服器控制工具組來加強 CodeArtifact 套件安全性
- 使用 Amazon CodeGuru Reviewer 偵測日誌中的安全問題
- 軟體偽影的供應鏈層級 （ 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：處理 Log4Shell

相關範例：

- 多區域套件發佈管道（GitHub）
- AWS CodeArtifact 使用（）在 上發佈 Node.js 模組 AWS CodePipeline GitHub
- AWS CDK Java CodeArtifact 管道範例（GitHub）
- 使用（）分發私有 .NET NuGet package AWS CodeArtifact 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曝險等級：高

實作指引

建置您的 AWS 帳戶 結構，從環境中移除持續的人工存取，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 Parameter

Store。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 AWS CloudFormation 或 AWS CDK 等工具來定義基礎設施，然後使用 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。
- 遵循使用多個帳戶組織 AWS 環境白皮書中有關環境分離的指導。
- 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 自動化整個開發和發行生命週期的測試

相關文件：

- AWS CI/CD 研討會
- AWS 使用有效的治理加速 上的部署
- 自動化安全、無人為介入的部署
- 使用 AWS Certificate Manager Private CA AWS Key Management Service 和非對稱金鑰簽署程式碼
- 程式碼簽署， 的信任和完整性控制 AWS Lambda

相關影片：

- 無人為介入：Amazon 的自動化持續交付管道

相關範例：

- 藍/綠部署搭配 AWS Fargate

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。管道不應只仰賴長期憑證資訊，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。
- 未定期審查與 CI/CD 基礎設施關聯的許可。
- 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，可降低憑證洩漏的風險。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。定期審查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 以程式設計方式為管道產生最低權限IAM政策。
- 將管道與監控和警示整合，以便通知您非預期或異常的活動，對於 AWS 受管服務，Amazon EventBridge 可讓您將資料路由至 AWS Lambda或 Amazon Simple Notification Service（Amazon）等目標SNS。

資源

相關文件：

- AWS 部署管道參考架構
- 監控 AWS CodePipeline
- 的安全最佳實務 AWS CodePipeline

相關範例：

- DevOps 監控儀表板（GitHub）

SEC11-BP08 建置在工作負載團隊中嵌入安全擁有權的程式

打造一項計畫或一種機制，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，但是讓建置人員團隊與生俱來擁有安全決策權，就能建置更快速、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關聯安全部門人員，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畫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。
- 有機會識別並了解具備高度改善價值的系統性問題或區域。
- 減少因安全檢閱調查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曝險等級：低

實作指引

從 SEC11-BP01 訓練，確保應用程式安全 的指引開始。接著識別您認為最適合組織的計畫操作模式。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，有助於計畫達成與成功實施。隨著這項計畫不斷建置，您一定要選擇可以用來顯示計畫價值的矩陣。從 AWS 如何解決此問題中學習是良好的學習體驗。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畫。
- 挑選計畫名稱。守門人、擁護者或倡導者是常見手法。
- 識別要應用的模式：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，識別專案贊助者。
- 計畫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 訓練，確保應用程式安全

- SEC11-BP02 自動化整個開發和發行生命週期的測試

相關文件：

- 如何建立威脅模型
- 如何思考雲端安全管控

相關影片：

- 預防性安全：考量與方法

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。可以在可靠性支柱白皮書中找到實作指引。

最佳實務領域

- 基礎
- 工作負載架構
- 變更管理
- 故障管理

基礎

問題

- REL 1. 如何管理服務配額和限制？
- REL 2. 如何規劃您的網路拓撲？

REL 1. 如何管理服務配額和限制？

雲端型工作負載架構具有服務配額 (也稱為服務限制)。這些配額的存在是為了防止意外佈建超過您需要的資源，並限制API操作的請求率，以防止服務遭到濫用。另外還有一些資源限制，例如可將位元下推到光纖纜線的速率，或實體磁碟的儲存量。

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 透過架構調節固定服務配額和限制
- REL01-BP04 監控和管理配額
- REL01-BP05 自動化配額管理
- REL01-BP06 確保目前配額與最大用量之間存在足夠的間隙，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。

預期結果：客戶可以 AWS 帳戶 實作適當的準則來監控關鍵指標、基礎設施檢閱和自動化修復步驟，以驗證未達到可能導致服務降級或中斷的服務配額和限制，從而防止服務降級或中斷。

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。

建立此最佳實務的優勢：監控和自動化服務配額和資源限制的管理可以主動減少故障。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畫性事件時將會有更高的彈性。

未建立此最佳實務時的曝險等級：高

實作指引

Service Quotas 是一種 AWS 服務，可協助您從一個位置管理超過 250 個 AWS 服務的配額。除了查詢配額值之外，您也可以從 Service Quotas 主控台或使用 AWS SDK. AWS Trusted Advisor offer 來請求和追蹤配額增加，以顯示某些服務某些方面的用量和配額。每個服務的預設服務配額也位於每個個別服務的 AWS 文件中（例如，請參閱 Amazon VPC Quotas ）。

某些服務限制，例如節流的速率限制APIs，是透過設定用量計畫，在 Amazon API Gateway 本身內設定。在其各自服務上設定為組態的一些限制包括佈建的 、已配置的 IOPSAmazon RDS儲存體和 Amazon EBS磁碟區配置。Amazon Elastic Compute Cloud 擁有自己的服務限制儀表板，有助於您管理執行個體、Amazon Elastic Block Store 和彈性 IP 位址限制。如果您有服務配額影響應用程式效能的使用案例，且無法根據您的需求調整，請聯絡 AWS Support ，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 AWS 的服務在正常使用中不會如預期般運作，並可能導致服務中斷或降級。例如，服務配額會限制區域中使用的 DL Amazon EC2執行個體數量。流量擴展事件期間，可以使用 Auto Scaling 群組（ ）達到此限制ASG。

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。它們也用於限制API操作的請求率，以保護服務免受濫用。

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。這些可能是儲存容量（例如，gp2 的大小限制為 1 GB - 16 TB）或磁碟輸送量。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。

如果存在服務配額影響應用程式效能的使用案例，且無法根據所需需求調整，請聯絡 AWS Support ，了解是否有緩解措施。如需有關調整固定配額的詳細資訊，請參閱 REL01-BP03 透過架構調節固定服務配額和限制。

有許多 AWS 服務和工具可協助監控和管理 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，以顯示某些 服務的某些方面的用量和配額。這有助於識別接近配額的服務。
- AWS Management Console 提供顯示服務配額值、管理、請求新配額、監控配額請求狀態和顯示配額歷史記錄的方法。
- AWS CLI 和 CDKs提供程式設計方法，以自動管理和監控服務配額層級和用量。

實作步驟

對於 Service Quotas：

- 檢閱 AWS Service Quotas。
- 若要了解現有的服務配額，請判斷所使用的服務（例如 IAM Access Analyzer）。大約有 250 個 AWS 服務由服務配額控制。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 增強此配額分析 AWS Config ，以尋找 中使用的所有AWS 資源 AWS 帳戶。
- 使用AWS CloudFormation 資料來判斷您的 AWS 資源已使用。查看在 中建立或使用 `list-stack-resources` AWS CLI 命令 AWS Management Console 建立的資源。您也可以查看設定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。
- 確認適用的服務配額。使用來自 Trusted Advisor 和 Service Quotas 的可程式設計存取資訊。
- 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)，以警示並通知服務配額是否接近或已達到上限。
- 建立自動化和程式化方法，以檢查某個區域中的服務配額是否已變更，但在同一帳戶的其他區域中未已變更 (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。
- 一旦確定特定服務需要更大的配額，請建立工程程序，以計算配額中所需的變更 (請參閱 REL01-BP05 自動化配額管理)。
- 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，以在佈建和使用新 AWS 服務之前檢閱服務配額，然後再推出生產或載入環境。（例如，負載測試帳戶）。

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。 CloudWatch 適當利用指標或日誌監控。
- 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 透過架構調節固定服務配額和限制
- REL01-BP04 監控和管理配額
- REL01-BP05 自動化配額管理
- REL01-BP06 確保目前配額與最大用量之間存在足夠的間隙，以適應容錯移轉
- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原
- REL12-BP05 使用混亂工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性支柱：可用性
- AWS Service Quotas（先前稱為服務限制）
- AWS Trusted Advisor 最佳實務檢查 （請參閱服務限制區段）
- AWS 限制對 AWS 答案的監控
- Amazon EC2 Service 限制
- 什麼是 Service Quotas？
- 如何請求提高配額
- 服務端點和配額
- Service Quotas 使用者指南
- 的配額監控 AWS
- AWS 故障隔離界限
- 備援的可用性
- AWS 適用於資料
- 什麼是持續整合？

- 什麼是持續交付？
- APN 合作夥伴：可協助進行組態管理的合作夥伴
- 在 上管理 account-per-tenant SaaS 環境中的帳戶生命週期 AWS
- 管理和監控工作負載中的API限流
- 使用 大規模檢視 AWS Trusted Advisor 建議 AWS Organizations
- 使用 自動化服務限制增加和企業支援 AWS Control Tower

相關影片：

- AWS Live re：Inforce 2019 – Service Quotas
- 使用 AWS 服務配額檢視和管理Service Quotas
- AWS IAM Quotas 示範

相關工具：

- Amazon CodeGuru Reviewer
- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOpsGuru
- 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：對於跨帳戶或區域的組態，或使用地區、區域或帳戶容錯移轉具有彈性設計的組態，服務和應用程式不應受到服務配額耗盡的影響。

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。
- 不要利用配額請求範本在多個區域和帳戶之間請求增加。
- 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。

建立此最佳實務的優勢：確認如果區域服務無法使用時，您可以處理次要區域或帳戶中目前的負載。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曝險等級：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 區域特定的。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)。

由於採用主動/主動、主動/被動 - 熱、主動/被動 - 冷，以及主動/被動- 指示燈方法實作設計，跨區域的工作負載越來越多，因此了解所有區域和帳戶配額級別至關重要。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畫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。

服務配額漂移是指特定指定配額的服務配額限制在一個區域而非所有區域中發生變更的情況，這對於追蹤和評估非常重要。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 透過架構調節固定服務配額和限制
- REL01-BP04 監控和管理配額
- REL01-BP05 自動化配額管理
- REL01-BP06 確保目前配額與最大用量之間存在足夠的間隙，以適應容錯移轉
- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原
- REL12-BP05 使用混亂工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性支柱：可用性
- AWS Service Quotas（先前稱為服務限制）
- AWS Trusted Advisor 最佳實務檢查 （請參閱服務限制區段）
- AWS 限制對 AWS 答案的監控

- Amazon EC2 Service 限制
- 什麼是 Service Quotas？
- 如何請求提高配額
- 服務端點和配額
- Service Quotas 使用者指南
- 的配額監控 AWS
- AWS 故障隔離界限
- 備援的可用性
- AWS 適用於資料
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可協助進行組態管理的合作夥伴
- 在 上管理 account-per-tenant SaaS 環境中的帳戶生命週期 AWS
- 管理和監控工作負載中的API限流
- 使用 大規模檢視 AWS Trusted Advisor 建議 AWS Organizations
- 使用 自動化服務限制增加和企業支援 AWS Control Tower

相關影片：

- AWS Live re：Inforce 2019 – Service Quotas
- 使用 AWS 服務配額檢視和管理Service Quotas
- AWS IAM Quotas 示範

相關服務：

- Amazon CodeGuru Reviewer
- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOpsGuru

- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

REL01-BP03 透過架構調節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。

範例包括網路頻寬、無伺服器函數調用承載大小、API閘道的限流爆量速率，以及與資料庫的並行使用者連線。

預期成果：在正常和高流量條件下，應用程式或服務會如預期般執行。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閾值

建立此最佳實務的優勢：確認應用程式是否會在所有預計的服務負載層級下執行，而不會中斷或降級。

未建立此最佳實務時的曝險等級：中

實作指引

與可用更高容量單位取代的軟服務配額或資源不同，無法變更 AWS 服務的固定配額。這表示在應用程式設計中使用時，必須評估所有此類 AWS 服務的潛在硬容量限制。

硬性限制會顯示在 Service Quotas 主控台中。如果資料欄顯示 `ADJUSTABLE = No`，則服務有硬性限制。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畫的一部分。

實作步驟

- 檢閱 AWS 可在應用程式設計階段使用的服務完整清單。
- 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。有些服務會在替代位置描述這些限制。
- 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 自動化配額管理
- REL01-BP06 確保目前配額與最大用量之間存在足夠的間隙，以適應容錯移轉
- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原

- REL12-BP05 使用混亂工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性支柱：可用性
- AWS Service Quotas（先前稱為服務限制）
- AWS Trusted Advisor 最佳實務檢查 （請參閱服務限制區段）
- AWS 限制對 AWS 答案的監控
- Amazon EC2 Service 限制
- 什麼是 Service Quotas？
- 如何請求提高配額
- 服務端點和配額
- Service Quotas 使用者指南
- 的配額監控 AWS
- AWS 故障隔離界限
- 備援的可用性
- AWS 適用於資料
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可協助進行組態管理的合作夥伴
- 在 上管理 account-per-tenant SaaS 環境中的帳戶生命週期 AWS
- 管理和監控工作負載中的API限流
- 使用 大規模檢視 AWS Trusted Advisor 建議 AWS Organizations
- 使用 自動化服務限制增加和企業支援 AWS Control Tower
- 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 AWS 服務配額檢視和管理Service Quotas

- AWS IAM Quotas 示範
- AWS re：Invent 2018：閉環和開場思維：如何控制大大小小的系統

相關工具：

- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOpsGuru
- 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：已經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閾值。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閾值
- 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。
- 僅設定 AWS Service Quotas 所支援之服務的警示，而不監控其他服務 AWS 。
- 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。
- 不利用範本進行多區域配額管理。

建立此最佳實務的優點：自動追蹤 AWS Service Quotas，並根據這些配額監控用量，可讓您在接近配額限制時查看 。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曝險等級：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 CloudWatch、Amazon 或 從 叫用 AWS Trusted Advisor。您也可以在 CloudWatch 日誌上使用指標篩選條件來搜尋和擷取日誌中的模式，以判斷用量是否接近配額閾值。

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來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。
- 使用 Trusted Advisor 服務限制來監控您目前在各種閾值的服務限制。
- 使用服務配額歷史記錄 （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，跨區域掃描服務配額，並比較差異。
- 自動化：設定排定的 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，以跨區域掃描服務配額並比較差異。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 透過架構調節固定服務配額和限制
- REL01-BP05 自動化配額管理
- REL01-BP06 確保目前配額與最大用量之間存在足夠的間隙，以適應容錯移轉
- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原
- REL12-BP05 使用混亂工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性支柱：可用性
- AWS Service Quotas（先前稱為服務限制）
- AWS Trusted Advisor 最佳實務檢查 （請參閱服務限制區段）
- AWS 限制對 AWS 答案的監控
- Amazon EC2 Service 限制
- 什麼是 Service Quotas？
- 如何請求提高配額

- 服務端點和配額
- Service Quotas 使用者指南
- 的配額監控 AWS
- AWS 故障隔離界限
- 備援的可用性
- AWS 適用於資料
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可協助進行組態管理的合作夥伴
- 在 上管理 account-per-tenant SaaS 環境中的帳戶生命週期 AWS
- 管理和監控工作負載中的API限流
- 使用 大規模檢視 AWS Trusted Advisor 建議 AWS Organizations
- 使用 自動化服務限制增加和企業支援 AWS Control Tower
- 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 AWS 服務配額檢視和管理Service Quotas
- AWS IAM Quotas 示範
- AWS re：Invent 2018：閉環和開場思維：如何控制大大小小的系統

相關工具：

- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOpsGuru
- AWS Config

- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

REL01-BP05 自動化配額管理

實作工具以在接近閾值時獲得提醒。您可以使用 AWS Service Quotas 來自動化配額增加請求APIs。

如果您將 Configuration Management Database （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配額。除了 之外 AWS SDK，Service Quotas 也使用 AWS Command Line Interface （AWS CLI） 提供自動化。

常見的反模式：

- 在試算表中追蹤配額和使用量。
- 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點：自動追蹤 AWS 服務配額，並監控該配額的使用量，可讓您在接近配額時看到 。您可以設定自動化，協助您在需要時請求增加配額。當您的用量趨勢與實現風險降低 (憑證遭入侵時) 和成本節省的優勢背道而馳時，您可以考慮降低部分配額。

未建立此最佳實務時的曝險等級：中

實作指引

- 設定自動監控 使用 實作工具SDKs，以便在接近閾值時提醒您。
  - 使用 Service Quotas 並透過自動化配額監控解決方案來增強服務，例如 AWS 限制監視器或來自的 產品 AWS Marketplace。
    - 什麼是 Service Quotas？
    - AWS - AWS Solution 上的配額監控
  - 使用 Amazon SNS和服務 AWS 配額 ，根據配額閾值設定自動回應APIs。 Service Quotas
  - 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。

- 設置自動化操作，對通知採取適當操作,並在必要時聯絡 AWS Support 。
- 手動啟動變更事件。
- 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：可協助進行組態管理的合作夥伴
- AWS Marketplace：協助追蹤限制CMDB的產品
- AWS Service Quotas (先前稱為 Service Limits)
- AWS Trusted Advisor 最佳實務檢查 （ 請參閱服務限制區段 ）
- AWS - AWS Solution 上的配額監控
- Amazon EC2 Service 限制
- 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用量之間存在足夠的間隙，以適應容錯移轉

本文說明如何維護資源配額與使用量之間的空間，以及如何讓您的組織受益。在完成使用資源之後，使用量配額可能會繼續佔用該資源。這可能會導致資源失敗或無法存取。透過確認您的配額是否涵蓋無法存取資源及其替換項目的重疊，來防止資源失敗。計算此差距時，應考慮諸如網路失敗、可用區域失敗或區域失敗等案例。

預期成果：資源或資源可存取性中的小型或大型故障可涵蓋在目前的服務閾值內。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。

- 不考慮某些 AWS 服務的服務故障隔離界限及其潛在的異常使用模式。

建立此最佳實務的優勢：當服務中斷事件影響應用程式可用性時，請使用雲端來實作策略，以便從這些事件中復原。一個範例策略是建立額外的資源來取代無法存取的資源，以適應容錯移轉條件，而不會耗盡您的服務限制。

未建立此最佳實務時的曝險等級：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。請考慮下列容錯移轉情況。

- 已中斷或無法存取的 VPC。
- 無法存取的子網路。
- 影響資源可存取性的降級可用區域。
- 聯網路由或輸入和輸出點遭到封鎖或變更。
- 影響資源可存取性的降級區域。
- 受區域或可用區域中的失敗所影響的資源子集。

容錯移轉的決策對於每個情況都是獨一無二的，因為業務影響有所不同。在決定容錯移轉應用程式或服務之前，先處理容錯移轉位置中的資源容量規劃和資源的配額。

檢閱每個服務的配額時，請考慮高於正常的活動峰值。這些峰值可能與由於聯網或權限而無法存取但仍處於活動狀態的的資源相關。未終止的作用中資源會計入服務配額限制。

實作步驟

- 維持服務配額和最大用量之間的空間，以適應容錯移轉或可存取性的喪失。
- 確定服務配額。說明典型的部署模式、可用性需求和使用量增長。
- 視需要請求增加配額。預計配額增加請求的等待時間。
- 確定您的可靠性需求 (也稱為「幾個 9」)。
- 了解可能的故障案例，例如元件遺失、可用區域或區域。
- 建立您的部署方法 (範例包括 Canary、藍/綠、紅/黑或滾動)。
- 為當前配額限制新增適當的緩衝。範例緩衝為 15%。
- 適當時包含靜態穩定性的計算 (區域 (Zonal) 和區域 (Regional))。

- 規劃使用量增長並監控使用量趨勢。
- 考慮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。這是針對關鍵業務排程而實作的有用策略，可以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 透過架構調節固定服務配額和限制
- REL01-BP04 監控和管理配額
- REL01-BP05 自動化配額管理
- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原
- REL12-BP05 使用混亂工程測試彈性

相關文件：

- AWS Well-Architected Framework 的可靠性支柱：可用性
- AWS Service Quotas（先前稱為服務限制）
- AWS Trusted Advisor 最佳實務檢查 （請參閱服務限制區段）
- AWS 限制對 AWS 答案的監控
- Amazon EC2 Service 限制
- 什麼是 Service Quotas？
- 如何請求提高配額
- 服務端點和配額
- Service Quotas 使用者指南
- 的配額監控 AWS

- AWS 故障隔離界限
- 備援的可用性
- AWS 適用於資料
- 什麼是持續整合？
- 什麼是持續交付？
- APN 合作夥伴：可協助進行組態管理的合作夥伴
- 在 上管理 account-per-tenant SaaS 環境中的帳戶生命週期 AWS
- 管理和監控工作負載中的API限流
- 使用 大規模檢視 AWS Trusted Advisor 建議 AWS Organizations
- 使用 自動化服務限制增加和企業支援 AWS Control Tower
- 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 AWS 服務配額檢視和管理Service Quotas
- AWS IAM Quotas 示範
- AWS re：Invent 2018：閉環和開場思維：如何控制大大小小的系統

相關工具：

- AWS CodeDeploy
- AWS CloudTrail
- Amazon CloudWatch
- Amazon EventBridge
- Amazon DevOpsGuru
- AWS Config
- AWS Trusted Advisor
- AWS CDK
- AWS Systems Manager
- AWS Marketplace

REL 2. 如何規劃您的網路拓撲？

工作負載通常存在於多個環境中。其中包括多個雲端環境 (可公開存取與私有)，也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 位址管理、私有 IP 位址管理和網域名稱解析等網路考量因素。

最佳實務

- REL02-BP01 為您的工作負載公有端點使用高可用性網路連線
- REL02-BP02 佈建雲端和內部部署環境中私有網路之間的備援連線
- REL02-BP03 確保 IP 子網路配置考量擴充性和可用性
- REL02-BP04 透過網格偏好 hub-and-spoke拓撲 many-to-many
- REL02-BP05 在所有連線的私有地址空間中強制執行不重疊的私有 IP 地址範圍

REL02-BP01 為您的工作負載公有端點使用高可用性網路連線

建立與工作負載公有端點的高可用性網路連線，可協助您減少因連線中斷而導致的停機時間，並改善工作負載SLA的可用性和。若要達成此目標，請使用高可用性 DNS、內容交付網路 （CDNs）、API閘道、負載平衡或反向代理。

預期成果：為您的公用端點規劃、建置和操作高可用性網路連線至關重要。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 GatewayURLs AWS AppSync APIs、AWS Lambda Function 和 Elastic Load Balancing （ELB） 都提供高可用性的公有端點。Amazon Route 53 為網域名稱解析提供高可用性DNS服務，以確認您的公有端點地址可以解決。

您也可以評估 AWS Marketplace 軟體設備進行負載平衡和代理。

常見的反模式：

- 設計高可用性的工作負載，無需規劃DNS和網路連線，以實現高可用性。
- 在個別執行個體或容器上使用公有網際網路地址，並透過 管理與其的連線DNS。
- 使用 IP 位址，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。
- 不規劃分散式拒絕服務 （DDoS） 攻擊。DDoS 攻擊會冒著關閉合法流量和降低使用者可用性的風險。

建立此最佳實務的優勢：針對高可用性和高彈性的網路連線進行設計，確保您的工作負載可供使用者存取且可用。

未建立此最佳實務時的曝險等級：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。若要驗證流量是否能夠到達端點， DNS 必須能夠將網域名稱解析為其對應的 IP 地址。使用高可用性和可擴展的網域名稱系統（DNS），例如 Amazon Route 53 來管理網域DNS的記錄。也可以使用 Amazon Route 53 提供的運作狀態檢查。運作狀態檢查會驗證您的應用程式是否可連線、可用和正常運作，而且可以按照模仿使用者行為的方式設定，例如請求網頁或特定 URL。若發生故障，Amazon Route 53 會回應DNS解決方案請求，並將流量導向僅運作狀態良好的端點。您也可以考慮使用 Amazon Route 53 提供的地理DNS和延遲型路由功能。

若要驗證工作負載本身是否具有高度可用性，請使用 Elastic Load Balancing （ELB）。Amazon Route 53 可用於將流量以 為目標ELB，將流量分散至目標運算執行個體。您也可以使用 Amazon API Gateway 搭配 AWS Lambda 進行無伺服器解決方案。客戶也可以在多個 中執行工作負載 AWS 區域。透過多站台主動/主動模式，工作負載可以為來自多個區域的流量提供服務。使用多站台主動/被動模式時，工作負載可為來自主動區域的流量提供服務，同時將資料複製到次要區域，並在主要區域發生故障時變為作用中狀態。然後，Route 53 運作狀態檢查可用於控制從主要區域的任何端點DNS容錯移轉到次要區域的端點，確認您的工作負載是否可連線且可供您的使用者使用。

Amazon 透過使用世界各地的邊緣位置網路來提供請求， CloudFront API以低延遲和高資料傳輸率分發內容。內容交付網路 （CDNs） 透過提供位於或快取在使用者附近的位置的內容來服務客戶。這也會改善應用程式的可用性，因為內容的負載會從伺服器轉移到 CloudFront的邊緣位置。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。

對於使用者分佈在地理上的工作負載， 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 Anycast 靜態 IP 地址，作為託管在一或多個 中的應用程式固定進入點 AWS 區域。這可讓流量盡可能接近您的使用者傳入 AWS 全球網路，進而改善工作負載的可連線性和可用性。 AWS Global Accelerator 也會使用 TCP、 和 運作狀態檢查HTTP來監控應用程式端點

的運作HTTPS狀態。您的端點的運作狀態或組態的任何變更都允許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 具有故障隔離設計，它使用兩個靜態IPv4地址，由獨立網路區域提供服務，以提高應用程式的可用性。

為了協助保護客戶免受DDoS攻擊， AWS 提供 AWS Shield Standard。Shield Standard 會自動開啟，並防止常見的基礎設施（第 3 層和第 4 層）攻擊，例如 SYN/UDP 洪水和反射攻擊，以支援 上應用程式的高可用性 AWS。如需針對更複雜和大型攻擊（例如UDP洪水）、狀態耗盡攻擊（例如TCPSYN洪水）的額外保護，以及協助保護在 Amazon Elastic Compute Cloud（AmazonEC2）、Elastic Load Balancing（ELB）CloudFront AWS Global Accelerator、Amazon 和 Route 53 上執行的應用程式，您可以考慮使用 AWS Shield Advanced。為了防止應用程式層攻擊，例如 HTTPPOST或 GET洪水，請使用 AWS WAF。 AWS WAF 可以使用 IP 地址、HTTP標頭、HTTP內文、URI字串、SQL注入和跨網站指令碼條件來判斷請求是否應該封鎖或允許。

實作步驟

1. 設定高可用性 DNS：Amazon Route 53 是高可用性和可擴展的網域名稱系統（DNS）Web 服務。Route 53 將使用者請求連接到在內部部署 AWS 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。首先建立 Route 53 運作狀態檢查並設定DNS容錯移轉 。以下是設定運作狀態檢查時要考慮的重要層面：
   a. Amazon Route 53 決定運作狀態檢查是否良好的方式
   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 的最佳實務 DNS
3. 將您的DNS服務連接至端點。
   a. 使用 Elastic Load Balancing 作為流量的目標時，請使用指向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. 對於無伺服器工作負載或使用 API Gateway APIs時私有，請使用 Route 53 將流量導向 API Gateway 。
4. 決定內容交付網路。
   a. 若要使用更接近使用者的邊緣位置交付內容，請先了解如何 CloudFront 交付內容 。
   b. 開始使用簡單的 CloudFront 分發 。 CloudFront 然後， 會知道您要從何處交付內容，以及如何追蹤和管理內容交付的詳細資訊。設定 CloudFront 分發時，請務必了解並考量下列各方面：
      i. 快取如何與 CloudFront 邊緣位置搭配使用
      ii. 增加直接從 CloudFront 快取提供的請求比例（快取命中率）

iii. 使用 Amazon CloudFront Origin Shield

iv. 透過 CloudFront 原始伺服器容錯移轉最佳化高可用性

5. 設定應用程式層保護： AWS WAF 協助您防範可能影響可用性、危及安全性或消耗過多資源的常見 Web 漏洞和機器人。若要深入了解，請檢閱AWS WAF 運作方式，以及何時準備好實作應用程式層HTTPPOSTANDGET洪水防護，請檢閱入門。 AWS WAF您也可以 AWS WAF 將 與 搭配使用， CloudFront 請參閱 Amazon CloudFront 功能 AWS WAF 使用方式的文件。
6. 設定額外DDoS保護：根據預設，所有 AWS 客戶都會獲得保護，免於以您的網站或應用程式為目標的常見、最常發生的網路和傳輸層DDoS攻擊， AWS Shield Standard 無需額外付費。如需對在 Amazon EC2、Elastic Load Balancing 、Amazon 和 Amazon Route 53 上執行的面向網際網路的應用程式提供額外保護 CloudFront AWS Global Accelerator，您可以考慮AWS Shield Advanced並檢閱DDoS彈性架構的範例。為了保護您的工作負載和公有端點免受DDoS攻擊，請參閱 入門 AWS Shield Advanced。

資源

相關的最佳實務：

- REL10-BP01 將工作負載部署到多個位置
- REL10-BP02 為您的多位置部署選取適當的位置
- REL11-BP04 在復原期間依賴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：可協助規劃聯網的合作夥伴
- AWS Marketplace 適用於網路基礎設施
- 什麼是 AWS Global Accelerator？
- 什麼是 Amazon CloudFront？
- 什麼是 Amazon Route 53？
- 什麼是 Elastic Load Balancing？
- 網路連線能力 - 建立您的雲端基礎
- 什麼是 Amazon API Gateway？
- 什麼是 AWS WAFAWS Shield、 和 AWS Firewall Manager？

- 什麼是 Amazon Application Recovery Controller？
- 設定DNS容錯移轉的自訂運作狀態檢查

相關影片：

- AWS re：Invent 2022 - 使用 改善效能和可用性 AWS Global Accelerator
- AWS re：Invent 2020：使用 Amazon Route 53 進行全域流量管理
- AWS re：Invent 2022 - 操作高可用性的多可用區域應用程式
- AWS re：Invent 2022 - 深入探索 AWS 網路基礎設施
- AWS re：Invent 2022 - 建置彈性網路

相關範例：

- 使用 Amazon Application Recovery Controller （ ARC ） 進行災難復原
- 可靠性研討會
- AWS Global Accelerator 研討會

REL02-BP02 佈建雲端和內部部署環境中私有網路之間的備援連線

在雲端和內部部署環境中的私有網路之間的連線中實作備援，以實現連線恢復能力。這可以透過部署兩個或多個連結和流量路徑來實現，從而在發生網路故障時保持連接。

常見的反模式：

- 您只依賴一個網路連線，這會產生單點故障。
- 您僅使用一個VPN通道或多個通道，這些通道結束於相同的可用區域。
- 您依賴一個 ISP 進行VPN連線，這可能會導致ISP中斷期間完全失敗。
- 未實作動態路由通訊協定，例如 BGP，這些通訊協定對於網路中斷期間重新路由流量至關重要。
- 您可以忽略VPN通道的頻寬限制，並高估其備份功能。

建立此最佳實務的優勢：透過在您的雲端環境與您的公司或內部部署環境之間實作備援連線，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曝險等級：高

實作指引

使用 AWS Direct Connect 將內部部署網路連線至 時 AWS，您可以使用在多個內部部署位置和多個 AWS Direct Connect 位置的不同裝置上結束的個別連線，來達到最大的網路彈性（SLA 99.99%）。此拓撲可針對裝置故障、連線問題和完全的位置中斷提供復原能力。或者，您也可以使用兩個個別連線至多個位置（SLA每個內部部署位置都連接至單一 Direct Connect 位置），以達到高彈性（ 99.9%）。此方法可防止因光纖斷裂或裝置故障而造成的連線中斷，並有助於減輕完全的位置故障。 AWS Direct Connect 復原工具組可協助設計您的 AWS Direct Connect 拓撲。

您也可以考慮在 上 AWS Site-to-Site VPN 結束 ， AWS Transit Gateway 作為主要 AWS Direct Connect 連線的成本效益備份。此設定可實現跨多個VPN通道的同等成本多路徑（ECMP）路由，允許高達 50Gbps 的輸送量，即使每個VPN通道上限為 1.25 Gbps。不過，請務必注意，這仍然 AWS Direct Connect 是將網路中斷降至最低並提供穩定連線的最有效選擇。

VPNs 使用網際網路將雲端環境連線至內部部署資料中心時，請將兩個VPN通道設定為單一 site-to-siteVPN連線的一部分。每個通道都應該在不同的可用區域結束，以獲得高可用性，並使用備援硬體來防止內部部署裝置故障。此外，請考慮您內部部署位置來自不同網際網路服務供應商（ISPs）的多個網際網路連線，以避免因單一ISP中斷而導致完全VPN連線中斷。選擇ISPs具有各種路由和基礎設施的 ，特別是具有 AWS 端點個別實體路徑的 ，可提供高度的連線可用性。

除了具有多個 AWS Direct Connect 連線和多個VPN通道（或兩者的組合）的實體備援之外，實作邊界閘道通訊協定（BGP）動態路由也很重要。Dynamic 會根據即時網路條件和設定的政策BGP，自動將流量從一個路徑重新路由到另一個路徑。這種動態行為對於在發生連結或網路故障時維持網路可用性和服務連續性特別有益。它可以快速選擇替代路徑，提高網路的彈性和可靠性。

實作步驟

- 在 AWS 和 內部部署環境之間取得高可用性連線。
  - 在個別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 位置以獲得高可用性。
  - 如果使用多個 AWS 區域，請在其中至少兩個 中建立備援。
- AWS Transit Gateway盡可能使用 來結束VPN連線 。
- 評估 AWS Marketplace 設備以結束VPNs或將 SD-WAN 擴展至 AWS。如果您使用 AWS Marketplace 設備，請在不同的可用區域中部署冗餘執行個體以實現高可用性。
- 提供內部部署環境的備援連線。
  - 您可能需要多個 的備援連線 AWS 區域 ，才能達成您的可用性需求。
  - 使用 AWS Direct Connect 彈性工具組以開始使用。

資源

相關文件：

- AWS Direct Connect 恢復力建議
- 使用備援 Site-to-SiteVPN連線提供容錯移轉
- 路由政策和BGP社群
- 中的主動/主動和主動/被動組態 AWS Direct Connect
- APN 合作夥伴：可協助規劃聯網的合作夥伴
- AWS Marketplace 適用於網路基礎設施
- Amazon Virtual Private Cloud 連線選項白皮書
- 建置可擴展且安全的多VPC AWS 網路基礎設施
- 使用備援 Site-to-SiteVPN連線提供容錯移轉
- 使用 AWS Direct Connect 彈性工具組開始
- VPC 端點和VPC端點服務 （ AWS PrivateLink ）
- 什麼是 AmazonVPC？
- 什麼是傳輸閘道？
- 什麼是 AWS Site-to-Site VPN？
- 使用 Direct Connect 閘道

相關影片：

- AWS re：Invent 2018：Amazon 的進階VPC設計和新功能 VPC
- AWS re：Invent 2019： AWS Transit Gateway reference architecture for many VPCs

REL02-BP03 確保 IP 子網路配置考量擴充性和可用性

Amazon VPC IP 地址範圍必須足夠大，以適應工作負載需求，包括考慮未來擴展，以及將 IP 地址配置到可用區域的子網路。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 位址空間。應為每個 配置私有 IP 地址範圍 （遵循 RFC 1918 年準則）VPC。在此流程中請滿足下列要求：

- 允許VPC每個區域的 IP 地址空間超過一個。
- 在 中VPC，為多個子網路留出空間，以便您可以涵蓋多個可用區域。

- 考慮將未使用的CIDR區塊空間保留在 中VPC，以供未來擴充。
- 確保有 IP 地址空間來滿足您可能使用的任何暫時性 Amazon EC2執行個體機群的需求，例如用於機器學習的 Spot Fleet、Amazon EMR叢集或 Amazon Redshift 叢集。由於每個 Kubernetes Pod 依預設都會從VPCCIDR區塊中指派一個可路由地址，因此應該對 Kubernetes 叢集進行類似的考量，例如 Amazon Elastic Kubernetes Service （Amazon EKS）。
- 請注意，每個子網路CIDR區塊的前四個 IP 地址和最後一個 IP 地址都已保留，且無法使用。
- 請注意，配置到 的初始VPCCIDR區塊VPC無法變更或刪除，但您可以將其他非重疊CIDR區塊新增至 VPC。不過，子網路IPv4CIDRs無法變更IPv6CIDRs。
- 最大的可能VPCCIDR區塊為 /16，最小區塊為 /28。
- 考慮其他連線網路（VPC、內部部署或其他雲端供應商），並確保 IP 地址空間不重疊。如需詳細資訊，請參閱 REL02-BP05 在所有連線的私有地址空間中強制執行不重疊的私有 IP 地址範圍。

預期成果：可擴展的 IP 子網路可以幫助您適應未來的成長，並避免不必要的浪費。

常見的反模式：

- 未考慮未來的成長，導致CIDR區塊太小且需要重新設定，進而可能導致停機時間。
- 錯誤預估 Elastic Load Balancer 可以使用的 IP 位址數量。
- 在相同的子網路中部署許多高流量負載平衡器
- 使用自動擴展機制，同時無法監控 IP 位址使用情況。
- 定義過大CIDR的範圍遠遠超過未來的成長預期，這可能會導致難以與其他地址範圍重疊的網路互連。

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。

- 根據您的服務需求、延遲、法規 AWS 帳戶 和災難復原（DR）需求選取相關 和 區域。
- 識別區域VPC部署的需求。
- 識別 的大小VPCs。
  - 判斷您是否要部署多VPC連線能力。

    - 什麼是 Transit Gateway？
    - 單一區域多VPC連線能力
- 確定您是否需要區隔聯網以滿足法規要求。
- VPCs 使用適當大小的CIDR區塊進行 ，以滿足您的目前和未來需求。
    - 如果您有未知的成長預測，您可能想要在較大的CIDR區塊側邊錯誤，以減少未來重新設定的可能性
- 請考慮使用子網路IPv6定址作為雙堆疊 的一部分VPC。IPv6 非常適合用於包含暫時性執行個體或容器機群的私有子網路，否則需要大量IPv4地址。

資源

相關 Well-Architected 的最佳實務：

- REL02-BP05 在所有連線的私有地址空間中強制執行不重疊的私有 IP 地址範圍

相關文件：

- APN 合作夥伴：可協助規劃聯網的合作夥伴
- AWS Marketplace 適用於網路基礎設施
- Amazon Virtual Private Cloud 連線選項白皮書
- 多個資料中心 HA 網路連線能力
- 單一區域多VPC連線能力
- 什麼是 AmazonVPC？
- IPv6 在 上 AWS
- IPv6 在參考架構上
- Amazon Elastic Kubernetes Service 啟動IPv6支援
- 對您的 - Classic Load Balancer VPC 的建議
- 可用區域子網路 - Application Load Balancer
- 可用區域 - Network Load Balancer

相關影片：

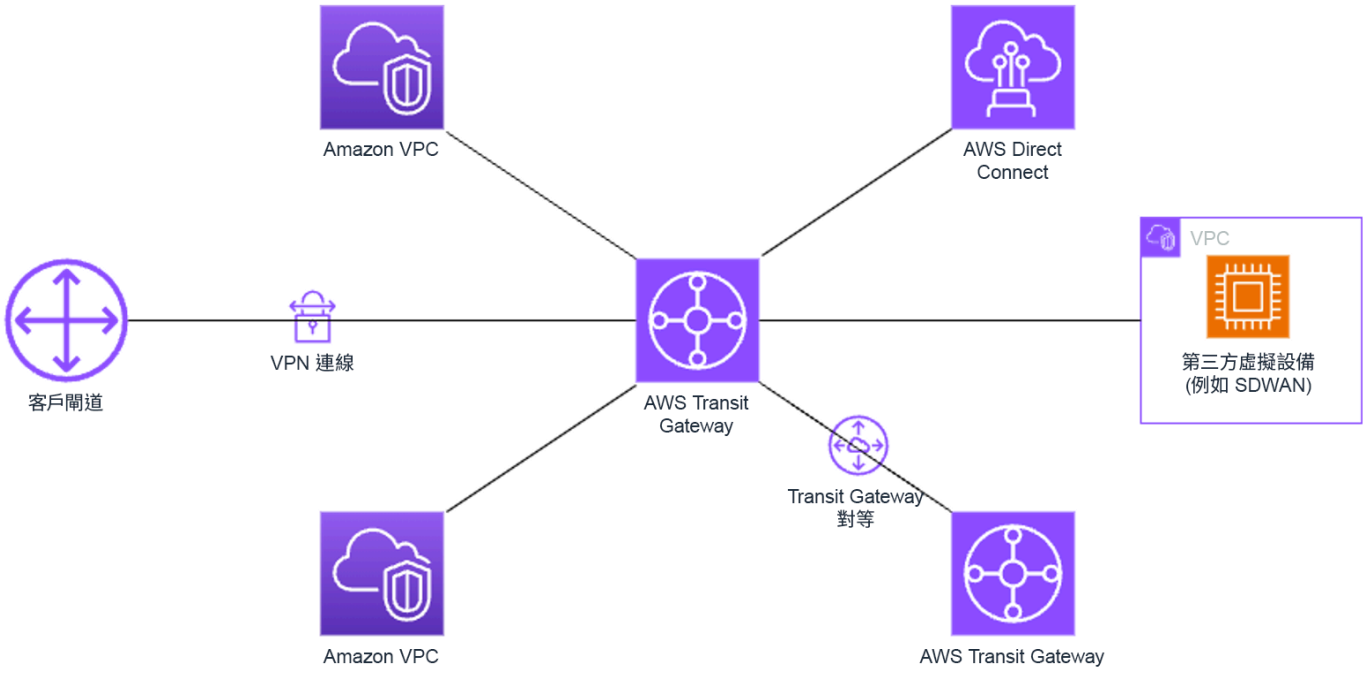
- AWS re：Invent 2018：Amazon 的進階VPC設計和新功能 VPC（NET303）

- AWS re：Invent 2019： AWS Transit Gateway 許多 VPCs（NET406-R1）的參考架構
- AWS re：Invent 2023： AWS 準備下一步？ 為成長和靈活性設計網路 （NET310）

REL02-BP04 透過網格偏好 hub-and-spoke拓撲 many-to-many

連接多個私有網路時，例如虛擬私有雲端 （VPCs） 和內部部署網路，請選擇透過網格網路的 hub-and-spoke拓撲。與網狀拓撲不同，每個網路直接連接到其他網路並提高複雜性和管理負荷，此 hub-and-spoke架構透過單一中樞集中連線。這種集中化簡化了網路結構，並增強了其可操作性、可擴展性和控制能力。

AWS Transit Gateway 是一項受管、可擴展且高可用性的服務，專為在 上建構 hub-and-spoke網路而設計 AWS。它可作為網路的中心樞紐，提供網路分段、集中式路由以及與雲端和內部部署環境的簡化連線。下圖說明如何使用 AWS Transit Gateway 建置 hub-and-spoke拓撲。



常見的反模式：

- 您會過度複雜化架構中的 hub-and-spoke路由政策，進而降低網路效率，並複雜化疑難排解和主動管理。
- 中樞內的路由式分段不足，這會導致漏洞，進而可能導致網路遭到未經授權的存取。
- 如果沒有經過仔細的最佳化，透過中樞路由的流量可能會導致更高的資料傳輸成本，尤其是跨可用區域和區域的流量。有效的流量管理策略對於控制費用至關重要。

建立此最佳實務的優點：隨著連線網路的數量增加，網狀連線的管理和擴展變得越來越具有挑戰性。為拓撲的 hub-and-spoke建構和操作 AWS Transit Gateway 提供可擴展且可靠的受管中樞。使用 時 AWS Transit Gateway，您可以建立連線並集中多個網路的流量路由。

未建立此最佳實務時的曝險等級：中

實作指引

- 規劃您的網路。
- 建立您的 AWS Transit Gateway。
- 連接您的 VPCs。
- 如有需要，請建立VPN連線或 Direct Connect 閘道，並將其與 Transit Gateway 建立關聯。
- 定義如何透過 Transit Gateway 路由表的組態，在連線VPCs與其他連線之間路由流量。
- 視需要使用 Amazon 監控和調整組態 CloudWatch ，以最佳化效能和成本。

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
- 使用區域 AWS Transit Gateway 間互連建立全球網路
- Amazon Virtual Private Cloud 連線選項
- APN 合作夥伴：可協助規劃聯網的合作夥伴
- AWS Marketplace 適用於網路基礎設施

相關影片：

- AWS re：Invent 2023 - AWS networking 基礎
- AWS re：Invent 2023 - 進階VPC設計和新功能

REL02-BP05 在所有連線的私有地址空間中強制執行不重疊的私有 IP 地址範圍

對等、透過 Transit Gateway 連線或透過 連線時，每個 的 IP 地址範圍VPCs不得重疊VPN。避免 IP 地址在 VPC和內部部署環境之間或您使用的其他雲端提供者之間發生衝突。您也必須有一種在需要時分配私有 IP 位址範圍的方法。IP 地址管理 （IPAM） 系統可協助自動化此作業。

預期成果：

- VPCs、內部部署環境或其他雲端提供者之間沒有 IP 地址範圍衝突。
- 適當的 IP 位址管理可以更輕鬆地擴展網路基礎設施，以適應網路需求的成長和變化。

常見的反模式：

- 在 中使用VPC與內部部署、公司網路或其他雲端供應商相同的 IP 範圍
- 不追蹤VPCs用於部署工作負載的 IP 範圍。
- 仰賴手動 IP 位址管理程序，例如試算表。
- 大小過大或過小的CIDR區塊，這會導致 IP 地址浪費或工作負載的地址空間不足。

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。這可防止使用不同應用程式的工作負載部分發生路由問題。

未建立此最佳實務時的曝險等級：中

實作指引

使用 IPAM，例如 Amazon VPC IP Address Manager ，來監控和管理您的CIDR使用。您也可以IPAMs從 取得數個 AWS Marketplace。在 上評估您的潛在用量 AWS、將CIDR範圍新增至現有 VPCs，並建立 VPCs 以允許規劃的使用量成長。

實作步驟

- 擷取電流CIDR消耗 （ 例如 VPCs和 子網路 ）。
  - 使用 服務API操作來收集目前CIDR消耗。
  - 使用 Amazon VPC IP Address Manager 來探索資源 。
- 記錄當前的子網路用量。
  - 使用服務API操作來收集每個區域中每個 的子網路。 VPC
  - 使用 Amazon VPC IP Address Manager 來探索資源 。
- 記錄當前用量。
- 確定是否建立了任何重疊的 IP 範圍。
- 計算備用容量。
- 識別重疊的 IP 範圍。您可以遷移到新的地址範圍，也可以考慮使用私有NAT閘道之類的技術，或者AWS PrivateLink如果您需要連接重疊範圍。

資源

相關的最佳實務：

- 保護網路

相關文件：

- APN 合作夥伴：可協助規劃聯網的合作夥伴
- AWS Marketplace 適用於網路基礎設施
- Amazon Virtual Private Cloud 連線選項白皮書
- 多個資料中心 HA 網路連線能力
- 連線具有重疊 IP 範圍的網路
- 什麼是 AmazonVPC？
- 什麼是 IPAM？

相關影片：

- AWS re：Invent 2023 - 進階VPC設計和新功能
- AWS re：Invent 2019： AWS Transit Gateway reference architecture for many VPCs
- AWS re：Invent 2023 - 為下一步做好準備？ 設計網路實現增長和靈活性
- AWS re：Invent 2021 - {New Launch} 大規模管理您的 IP 地址 AWS

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減輕或承受故障？

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）或微服務架構建置高度可擴展且可靠的工作負載。服務導向架構（SOA）是透過服務介面讓軟體元件可重複使用。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 選擇如何分割工作負載
- REL03-BP02 建置專注於特定業務網域和功能的服務
- REL03-BP03 提供每個 的服務合約 API

REL03-BP01 選擇如何分割工作負載

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。根據您的應用程式需求，這可能最終會盡可能結合服務導向架構（SOA）與微服務。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：

- 微服務 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 微服務架構（或在極少數情況下，選擇單片架構）。即使您選擇從整體架構開始，仍必須確保其為模組化，並隨著產品隨著使用者採用而擴展，最終可以發展為 SOA或 微服務。SOA 和 微服務分別提供較小的分割，這是現代可擴展且可靠的架構，但需要考慮權衡，特別是部署微服務架構時。

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 微服務分別提供較小的分割，這是現代可擴展且可靠的架構。SOA 對於實現更小的分割，同時避免微服務的一些複雜性，可能是一個很好的妥協。如需詳細資訊，請參閱 Microservice Trade-Offs。
- 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。如需更多詳細資訊，請參閱在 上實作 Microservices AWS。
- 考慮遵循 Strangler Fig 模式，將整體重構為較小的組件。這涉及逐步以新的應用程式和服務取代特定的應用程式元件。AWS Migration Hub Refactor Spaces 可充當增量重構的起點。如需詳細資訊，親參閱 Seamlessly migrate on-premises legacy workloads using a strangler pattern。
- 實作微服務可能需要服務探索機制，以允許這些分散式服務彼此通訊。 AWS App Mesh 可與服務導向的架構搭配使用，以提供可靠的服務探索和存取。 AWS Cloud Map也可用於動態DNS的 型服務探索。
- 如果您要從整體遷移至 SOA，Amazon MQ 可以在重新設計雲端中的舊版應用程式時，協助橋接差距，作為服務匯流排。
- 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。在重構程序中，您應該選擇使用關聯式或非關聯式（否 SQL）類型的資料庫繼續。如需詳細資訊，請參閱從 SQL到否 SQL。

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網域和功能的服務

相關文件：

- Amazon API Gateway：設定RESTAPI使用開啟API
- 什麼是服務導向架構？
- 有界限的環境 (領域驅動型設計的集中模式)
- 在 上實作 Microservices AWS
- 微型服務權衡
- 微型服務 - 此新架構術語的定義
- 上的微服務 AWS
- 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：

- 在 上使用 Microservices 實現卓越 AWS

REL03-BP02 建置專注於特定業務網域和功能的服務

服務導向架構（SOA）定義具有業務需求定義之詳細函數的服務。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：工程師和業務利益相關者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。現有單體會分解為邊界內容，而系統設計則朝向 SOA或 微服務架構邁進。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：

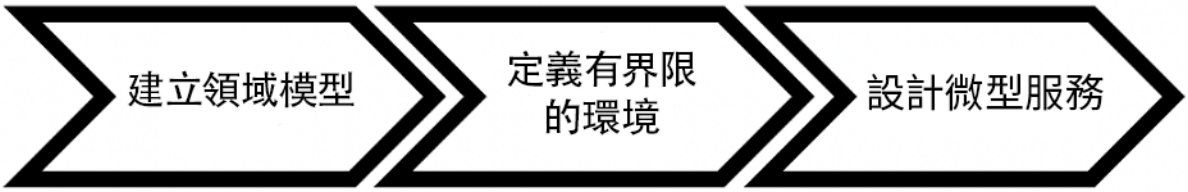
- 團隊是依據特定技術領域 (例如 UI 和 UX、中介軟體或資料庫) 組成的，而不是特定的業務領域。
- 應用程式跨多個領域責任。跨有界限環境的服務可能更難以維護，需要較大量的測試工作，且需要多個領域團隊參與軟體更新。
- 領域相依性 (例如領域實體程式庫) 會跨服務共用，因此一個服務領域出現變更時，需要變更其他服務領域
- 服務合約和商業邏輯無法以通用且一致的領域語言來表達實體，因此會導致翻譯層級使系統複雜化，並增加偵錯工作。

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

網域驅動設計（DDD）是圍繞業務網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。



領域驅動的設計

實作步驟

- 團隊可舉行事件風暴研討會，以便箋格式快速識別事件、命令、彙總和領域。
- 在網域環境中形成網域實體和函數之後，可以使用有界限的環境將網域劃分為服務，其中共用相似特徵和屬性的實體會分組在一起。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。
- 泡泡內容等戰術技術可讓您DDD在現有或舊版應用程式中導入 ，而無需預先重寫和對 的完整承諾 DDD。在 bubble 環境方法中，使用服務映射和協調或防損毀層來建立一個小的有界限環境，可保護新定義的網域模型免受外部影響。

在團隊執行網域分析和定義的實體和服務合約之後，他們可以利用 AWS 服務來實作其以雲端為基礎的服務。

- 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 ） 可協助團隊讓服務專注於解決業務問題。
- 選取最符合您的企業網域需求和微服務架構的 AWS 服務：
  - AWS 無伺服器可讓您的團隊專注於特定網域邏輯，而不是管理伺服器和基礎設施。
  - AWS的容器可簡化基礎設施的管理，讓您得以專注在您的網域要求上。
  - 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，它概述了一個框架，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業務邏輯分開的模式，可協助團隊專注於網域功能並改善軟體品質。

資源

相關的最佳實務：

- REL03-BP01 選擇如何分割工作負載
- REL03-BP03 提供每個 的服務合約 API

相關文件：

- AWS Microservices
- 在 上實作 Microservices AWS
- 如何將整合型服務分成微型服務
- 由 Legacy Systems 包圍DDD時入門

- 網域驅動設計：解決軟體核心的複雜性
- 在 上建置六邊形架構 AWS
- 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：

- 在 AWS （從 DDD/EventStormingWorkshop ） 上設計雲端原生微服務

相關工具：

- AWS 雲端 資料庫
- 上的無伺服器 AWS
- 的容器 AWS

REL03-BP03 提供每個 的服務合約 API

服務合約是機器可讀取API定義中定義的API生產者和消費者之間的書面協議。合約版本控制策略可讓消費者繼續使用現有的 ，API並在準備好API時將其應用程式遷移至較新的應用程式。只要遵守合約，就隨時可執行生產者部署。服務團隊可以使用他們選擇的 技術堆疊來滿足API合約。

預期結果：使用服務導向或微服務架構建置的應用程式可以獨立運作，同時具有整合的執行時間相依性。當雙方遵循共同API合約時，部署到API消費者或生產者的變更不會中斷整體系統的穩定性。透過服務通訊的元件APIs可以執行獨立的功能版本、升級至執行時間相依性，或容錯移轉至災難復原（DR）站台，彼此幾乎沒有影響。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。

常見的反模式：

- 建立APIs不含強烈輸入結構描述的服務。這會導致 APIs無法用來產生無法以程式設計方式驗證的API繫結和承載。

- 不採用版本控制策略，這會強制API消費者在服務合約演進時更新和發行或失敗。
- 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。
- 不使用API合約來開發測試案例和模擬API實作，以允許獨立測試服務元件。

建立此最佳實務的好處：由透過API服務合約通訊的元件組成的分散式系統可以提高可靠性。開發人員可以在開發程序的早期發現潛在的問題，並在編譯期間進行類型檢查，以確認請求和回應遵循API合約和必填欄位。API 合約為 APIs和 提供者提供了明確的自我記錄介面，讓不同的系統和程式設計語言之間具有更好的互通性。

未建立此最佳實務時的曝險等級：中

實作指引

識別業務網域並確定工作負載分割後，您就可以開發服務 APIs。首先，定義 的機器可讀取服務合約 APIs，然後實作API版本控制策略。當您準備好透過 REST、GraphQL 或非同步事件等常見通訊協定整合 服務時，您可以將 AWS 服務整合到您的架構中，以將元件與強式API合約整合。

AWS 服務API對象的服務

將 Amazon API Gateway 、 AWS AppSync和 Amazon EventBridge 等 AWS 服務整合到您的架構中，以在應用程式中使用API服務合約。Amazon API Gateway 可協助您與直接原生 AWS 服務和其他 Web 服務整合。API Gateway 支援 OpenAPI 規格和 versioning。 AWS AppSync 是您透過定義 GraphQL 結構描述來定義查詢、突變和訂閱的服務界面所設定的受管 GraphQL 端點。Amazon EventBridge 使用事件結構描述來定義事件，並為您的事件產生程式碼繫結。

實作步驟

- 首先，為您的 定義合約API。合約會表達 的功能API，並定義API輸入和輸出的強烈輸入資料物件和欄位。
- 當您APIs在 API Gateway 中設定 時，您可以匯入和匯出端點的開放API規格。
  - 匯入 OpenAPI 定義可簡化 的建立，API並可整合 AWS 基礎設施作為程式碼工具，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，並為服務消費者提供整合規格。
- 您可以使用 APIs AWS AppSync 定義 GraphQL 結構描述檔案來定義和管理 GraphQL，以產生合約界面，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 與 整合的專案 AWS AppSync 會產生強式 JavaScript 查詢檔案，用於您的應用程式，以及 Amazon DynamoDB 資料表的 AWS AppSync GraphQL 用戶端程式庫。

- 當您從 Amazon 取用服務事件時 EventBridge，事件會遵守已存在於結構描述登錄檔中的結構描述，或是您使用 OpenAPI Spec 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。
- 擴展或版本您的 API。新增可使用選用欄位或必要欄位的預設值設定的欄位時，延伸 API 是更簡單的選項。
  - JSON REST和 GraphQL 等通訊協定的 型合約非常適合合約延伸。
  - XML 等通訊協定的 型合約SOAP應與 服務消費者進行測試，以判斷合約延伸的可行性。
- 版本控制 時API，請考慮實作 Proxy 版本控制，其中使用外觀來支援版本，以便在單一程式碼庫中維護邏輯。
  - 使用 API Gateway，您可以使用請求和回應映射，透過建立外觀來提供新欄位的預設值，或從請求或回應中刪除的欄位，以簡化吸收合約變更。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 選擇如何分割工作負載
- REL03-BP02 建置專注於特定業務網域和功能的服務
- REL04-BP02 實作鬆散耦合相依性
- 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（應用程式程式設計介面）？
- 在 上實作 Microservices AWS
- 微型服務權衡
- 微型服務 - 此新架構術語的定義
- 上的微服務 AWS
- 使用 API Gateway 擴充功能開啟API
- 開啟 API-規格
- GraphQL：結構描述和類型
- Amazon EventBridge 程式碼繫結

相關範例：

- Amazon API Gateway：設定RESTAPI使用開啟API
- 使用 Open 的 Amazon API Gateway 至 Amazon DynamoDB CRUD 應用程式API
- 無伺服器時代的現代應用程式整合模式：APIGateway Service Integration
- 使用 Amazon 實作標頭型API閘道版本控制 CloudFront
- AWS AppSync：建置用戶端應用程式

相關影片：

- 使用在 中開啟API AWS SAM 管理API閘道

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？

分散式系統仰賴通訊網路將伺服器或服務等元件互相連線。儘管這些網路中出現資料遺失或延遲，但工作負載仍必須可靠地運作。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故障，並改善故障間隔的平均時間（MTBF）。

最佳實務

- REL04-BP01 識別您依賴的分散式系統類型
- REL04-BP02 實作鬆散耦合相依性
- REL04-BP03 持續工作
- REL04-BP04 使所有回應都具有同位性

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、非同步或批次處理。同步系統必須盡快處理請求，並使用 HTTP/SREST、 或遠端程序呼叫（RPC）通訊協定進行同步請求和回應呼叫，以彼此通訊。非同步系統透過中間服務以

非同步方式交換資料來相互通訊，而不需要耦合個別系統。批處理系統接收大量輸入資料，無需人工介入即可執行自動化資料處理，並產生輸出資料。

預期成果：設計與同步、非同步和批次相依性有效互動的工作負載。

常見的反模式：

- 工作負載會無限期地等待來自其相依性的回應，這可能會導致工作負載用戶端逾時，而不知道是否已收到其請求。
- 工作負載使用可同步呼叫彼此的一系列相依系統。這需要每個系統都可供使用，並在整個系列成功之前成功處理請求，從而導致潛在的脆弱行為和整體可用性。
- 工作負載會以非同步方式與其相依性進行通訊，並依賴於僅保證傳遞訊息一次的概念，而且通常仍然可以接收重複的訊息。
- 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：對於特定的工作負載來說，在同步、非同步和批次之間實作一種或多種通訊方式很常見。此最佳實務可協助您識別與每種通訊方式相關的不同權衡，讓您的工作負載能夠容忍其任何相依性中斷。

未建立此最佳實務時的風險暴露等級：高

實作指引

下列各節包含每種相依性的一般和特定實作指引。

一般指引

- 請確定您的相依性提供的效能和可靠性服務層級目標（SLOs）符合工作負載的效能和可靠性需求。
- 使用 AWS 可觀測性服務來監控回應時間和錯誤率，以確保您的相依性在工作負載所需的層級提供服務。
- 識別工作負載在與其相依性通訊時可能面臨的潛在挑戰。分散式系統面臨各種挑戰，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日誌記錄，以協助您在相依性遇到問題時對問題進行疑難排解。

同步相依性

在同步通訊中，工作負載會將請求傳送至其相依性，並封鎖等待回應的操作。當其相依性收到請求時，它會嘗試盡快處理它，並將回應傳送回您的工作負載。同步通訊的一個重大挑戰是它會導致時間耦合，

這需要您的工作負載及其相依性同時可用。當您的工作負載需要與其相依性同步通訊時，請考慮下列指引：

- 您的工作負載不應該依賴多個同步相依性來執行單一函數。此相依性系列增加了整體的脆弱性，因為路徑中的所有相依性都必須可用，才能順利完成請求。
- 當相依性狀態不良或無法使用時，請確定處理錯誤並重試策略。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。如需雙模式行為的詳細資訊，請參閱 REL11-BP05 使用靜態穩定性來防止雙模式行為。
- 請記住，快速檢錯好於讓工作負載等待。例如，AWS Lambda 開發人員指南說明如何在呼叫 Lambda 函數時處理重試和失敗。
- 當工作負載呼叫其相依性時設定逾時。此技術可避免等待太長時間或無限期等待回應。如需此問題的實用討論，請參閱調整具有延遲感知的 Amazon DynamoDB 應用程式 Java AWS SDKHTTP請求設定。
- 盡可能減少從工作負載到其相依性的呼叫次數，以滿足單一請求。在兩者之間進行聊天呼叫會增加耦合和延遲。

異步相依性

若要暫時將工作負載與其相依性分離，它們應該以非同步方式進行通訊。使用非同步方法，您的工作負載可以繼續執行任何其他處理，而不必等待其相依性或相依性系列來傳送回應。

當您的工作負載需要與其相依性異步通訊時，請考慮下列指引：

- 根據您的使用案例和需求來決定是使用訊息傳遞還是事件串流。訊息傳遞可讓您的工作負載透過訊息代理程式傳送和接收訊息，藉此與其相依性進行通訊。事件串流可讓您的工作負載及其相依性使用串流服務來發布和訂閱事件 (以連續資料串流形式傳送)，這些事件需要盡快處理。

- 訊息傳遞和事件串流處理訊息的方式不同，因此您需要根據下列內容做出權衡決策：
  - 訊息優先級：訊息代理程式可以在正常訊息之前處理高優先級訊息。在事件串流中，所有訊息都有相同的優先級。
  - 訊息消耗：訊息代理程式確保消費者收到消息。事件串流消費者必須追蹤他們讀取的最後一則訊息。
  - 訊息排序：除非您使用 first-in-first-out （FIFO） 方法，否則無法保證透過訊息，以傳送的確切順序接收訊息。事件串流永遠會保留產生資料的順序。
  - 訊息刪除：使用訊息傳遞時，消費者必須在處理訊息後刪除它。事件串流服務會將訊息附加至串流，並保留在其中，直到訊息的保留期過期為止。此刪除政策使得事件串流適合重播訊息。

- 定義工作負載如何知道其相依性何時完成其工作。例如，當工作負載以非同步方式調用 Lambda 函數時，Lambda 會將事件放在佇列中，並傳回成功回應，但沒有額外資訊。處理完成後，Lambda 函數可以將結果傳送至目的地，並根據成功或失敗進行設定。
- 透過利用冪等性來構建工作負載以處理重複訊息。冪等性意味著即使為相同訊息產生多次工作負載，工作負載的結果也不會變更。重要的是要指出，如果發生網路故障或尚未收到確認，訊息傳遞或串流服務將重新傳遞訊息。
- 如果您的工作負載未從其相依性中取得回應，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 文件說明了如何處理非同步調用的錯誤。
- 運用適當的可觀測性、偵錯和追蹤工具，管理和操作工作負載的非同步通訊及其相依性。您可以使用 Amazon CloudWatch 監控訊息和事件串流服務。還可以使用 AWS X-Ray 檢測工作負載，以快速獲得疑難排解問題的洞見。

批處理相依性

批處理系統會取得輸入資料，啟動一系列作業來處理它，並產生一些輸出資料，無需人工介入。視資料大小而定，工作可能會執行幾分鐘，在某些情況下執行數天。當您的工作負載需要與其批處理相依性通訊時，請考慮下列指引：

- 定義工作負載執行批次工作的時間範圍。工作負載可以設定週期性模式來調用批次系統，例如，每小時或每月結束時。
- 確定資料輸入和已處理的資料輸出的位置。選擇儲存服務，例如 Amazon Simple Storage Services（Amazon S3）、Amazon Elastic File System （Amazon EFS） 和 Amazon FSx for Lustre，讓您的工作負載大規模讀取和寫入檔案。
- 如果您的工作負載需要叫用多個批次工作，您可以利用 AWS Step Functions 來簡化在 AWS 或內部部署中執行的批次工作協調。此範例專案示範使用 Step Functions、AWS Batch 和 Lambda 協同運作批次任務。
- 監控批次任務以尋找異常情況，例如任務所花費的時間超過應該完成的時間。您可以使用 CloudWatch Container Insights 等工具來監控 AWS Batch 環境和任務。在這種情況下，工作負載將從一開始就停止下一個任務，並通知相關人員有關例外情況。

資源

相關文件：

- AWS 雲端 操作：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式行為
- AWS Lambda 開發人員指南： 中的錯誤處理和自動重試 AWS Lambda
- 調整可感知延遲的 Amazon DynamoDB 應用程式的 AWS Java SDKHTTP請求設定
- AWS 訊息傳遞
- 什麼是串流資料？
- AWS Lambda 開發人員指南：非同步調用
- Amazon Simple Queue ServiceFAQ：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 的可用 CloudWatch指標 SQS
- Amazon Kinesis Data Streams 開發人員指南：使用 Amazon 監控 Amazon Kinesis Data Streams Service CloudWatch
- AWS X-Ray 開發人員指南： AWS X-Ray 概念
- AWS GitHub： AWS Step 函數 Complex Orchestrator 應用程式上的範例
- AWS Batch 使用者指南： AWS Batch CloudWatch Container Insights

相關影片：

- AWS Summit SF 2022 - 透過 AWS （ COP310 ） 進行全堆疊可觀測性和應用程式監控

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。

解除相依性 (例如佇列系統、串流系統和工作流程) 有助於將變更或失敗對系統造成的影響降到最低。這種分離使組件的行為不會影響依賴它的其他組件，從而提高了彈性和敏捷性。

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布的界面。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，您可以橫向擴充，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直接 point-to-point互動整合，而是通常透過中繼耐用儲存層整合，例如 Amazon SQS佇列、串流資料平台，例如 Amazon Kinesis 或 AWS Step Functions。

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 AWS Step Functions 只是新增中繼層進行鬆散耦合的兩種方式。事件驅動的架構也可以 AWS 雲端 使用 Amazon 建置在 中 EventBridge，這可以從用戶端 （事件生產者） 仰賴的服務 （事件消費者） 中抽象用戶端 （事件生產者）。當您需要高輸送量、推送型訊息時， many-to-manyAmazon 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播到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將元件層級故障的影響降到最低，這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充。

常見的反模式：

- 部署整合型工作負載。
- 直接在工作負載層APIs之間調用，而沒有容錯移轉或非同步處理請求的能力。

- 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。

建立此最佳實務的優勢：鬆耦合有助於將某個元件的行為與依賴它的其他元件隔離，進而提高彈性和敏捷性。避免一個元件中的失敗影響其他元件。

未建立此最佳實務時的曝險等級：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。其中包括實作完全受管佇列、自動化工作流程、對事件做出反應APIs等服務，這些服務可協助隔離元件與其他元件的行為，進而提高復原能力和靈活性。

- 建置事件驅動架構：Amazon EventBridge 可協助您建置鬆散耦合和分散式事件驅動架構。
- 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。
- 將元件容器化為微服務：微服務可讓團隊建置由小型獨立元件組成的應用程式，這些元件透過定義明確的 進行通訊APIs。Amazon Elastic Container Service （Amazon ECS） 和 Amazon Elastic Kubernetes Service （Amazon EKS） 可協助您更快開始使用容器。
- 使用 Step Functions 管理工作流程：Step Functions 可協助您將多項 AWS 服務協調為彈性工作流程。
- 利用發佈訂閱 （pub/sub） 訊息架構：Amazon Simple Notification Service （Amazon SNS） 提供從發佈者到訂閱者 （也稱為生產者和消費者） 的訊息傳遞。

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 建置事件驅動應用程式 EventBridge
  - AWS re：Invent 2022 - 使用 Amazon 設計事件驅動整合 EventBridge
- 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。其中包括分散式系統的元件、用於解耦的佇列 （在 Amazon SQS 伺服器上分佈），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。佇列會在多個 Amazon SQS 伺服器之間存放訊息，以進行備援。

- 基本 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 上實作 Microservices AWS
  - 開始建構吧！使用容器建構微型服務
- AWS Step Functions 您可以透過 建置分散式應用程式、自動化程序、協調微服務等。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 建立無伺服器工作流程 AWS Lambda
  - 入門 AWS Step Functions

資源

相關文件：

- Amazon EC2：確保錯位
- Amazon 建置者資料中心：分散式系統的挑戰
- Amazon 建置者資料中心：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？
- 結束您的整合型架構
- 使用 AWS Step Functions 和 Amazon 協調佇列型 Microservices SQS
- 基本 Amazon SQS架構
- 佇列式架構

相關影片：

- AWS 2019 年紐約高峰會：事件驅動架構和 Amazon 簡介 EventBridge （MAD205）
- AWS re：Invent 2018：閉環和開場思維：如何控制系統，無論大小 ARC337（包括鬆散的耦合、固定工作、靜態穩定性）
- AWS re：Invent 2019：移至事件驅動的架構 （SVS308）

- AWS re：Invent 2019：使用 Amazon SQS和 Lambda 可擴展的無伺服器事件驅動應用程式
- AWS re：Invent 2022 - 使用 Amazon 設計事件驅動整合 EventBridge
- AWS re：Invent 2017：Elastic Load Balancing Deep Dive 和最佳實務

REL04-BP03 持續工作

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。實際上，這是 Amazon Route 53 處理端點運作狀態檢查 (例如 IP 位址) 的方式，以判斷最終使用者如何路由到端點。

未建立此最佳實務時的曝險等級：低

實作指引

- 執行持續工作，以便當負載大量快速變更時，系統不會發生故障。
- 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。
  - Amazon 建置者資料中心：可靠性、持續工作以及咖啡時刻
  - AWS re：Invent 2018：閉環和開場思維：如何控制大大小小的系統 ARC337（包括持續工作）
    - 針對監控 100,000 部伺服器的運作狀態檢查系統範例，請設計工作負載，以便無論成功或失敗次數為何，承載大小都能保持不變。

資源

相關文件：

- Amazon EC2：確保錯位
- Amazon 建置者資料中心：分散式系統的挑戰
- Amazon 建置者資料中心：可靠性、持續工作以及咖啡時刻

相關影片：

- AWS 2019 年紐約高峰會：事件驅動架構和 Amazon 簡介 EventBridge （MAD205）
- AWS re：Invent 2018：閉環和開場思維：如何控制大大小小的系統 ARC337（包括持續工作）
- AWS re：Invent 2018：閉環和開場思維：如何控制系統，大大小小 ARC337（包括鬆散的耦合、固定工作、靜態穩定性）
- AWS re：Invent 2019：移至事件驅動的架構 （SVS308）

REL04-BP04 使所有回應都具有同位性

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。若要這樣做，用戶端可以使用不透明度權杖發出API請求，每當重複請求時，都會使用相同的權杖。無能性服務API會使用權杖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。在 中使用隱含性字符APIs，服務可以接收一或多次靜音請求，而不會建立重複記錄或副作用。

未建立此最佳實務時的曝險等級：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。
  - 用戶端可以使用意識模糊權杖發出API請求，每當重複請求時，都會使用相同的權杖。無能性服務API會使用權杖傳回與第一次完成請求時傳回之回應相同的回應。
    - Amazon EC2：確保錯位

資源

相關文件：

- Amazon EC2：確保錯位
- Amazon 建置者資料中心：分散式系統的挑戰
- Amazon 建置者資料中心：可靠性、持續工作以及咖啡時刻

相關影片：

- AWS 2019 年紐約高峰會：事件驅動架構和 Amazon 簡介 EventBridge （MAD205）
- AWS re：Invent 2018：閉環和開場思維：如何控制系統，無論大小 ARC337（包括鬆散的耦合、固定工作、靜態穩定性）
- AWS re：Invent 2019：移至事件驅動的架構 （SVS308）

REL 5. 如何設計分散式系統中的互動以減輕或承受故障？

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。結果是改善復原的平均時間 （MTTR）。

最佳實務

- REL05-BP01 實作優雅降級，將適用的硬相依性轉換為軟相依性
- REL05-BP02 節流請求
- REL05-BP03 控制和限制重試呼叫
- REL05-BP04 快速失敗並限制佇列
- REL05-BP05 設定用戶端逾時
- REL05-BP06 盡可能讓系統處於無狀態
- REL05-BP07 實作緊急槓桿

REL05-BP01 實作優雅降級，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一直失敗，則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，這些預設值需要保留 up-to-date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，長時間不日誌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 與幾乎所有關聯式資料庫一樣，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 Serverless 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充的能力，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：提高輸送量的限流API請求
- CircuitBreaker（總結「發行！」書籍中的斷路器）
- 中的錯誤重試和指數退避 AWS

- Michael Nygard “Release It! Design and Deploy Production-Ready Software”
- 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：介紹 Amazon Builders 程式庫 （DOP328）

相關範例：

- Well-Architected 實驗室：Level 300：實作運作狀態檢查和管理相依性以提升可靠性

REL05-BP02 節流請求

限流請求以減輕因需求非預期地增加而耗盡資源。低於限流率的請求會進行處理，而超過定義限制的請求會遭到拒絕，並顯示傳回訊息指出請求已限流。

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 未實作端點限流，或保留預設值，而不考慮預期的磁碟區。
- 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。
- 尚未為應用程式對應用程式 （A2A） API取用者設定或考慮用量計劃。
- 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 位址的速率限制。

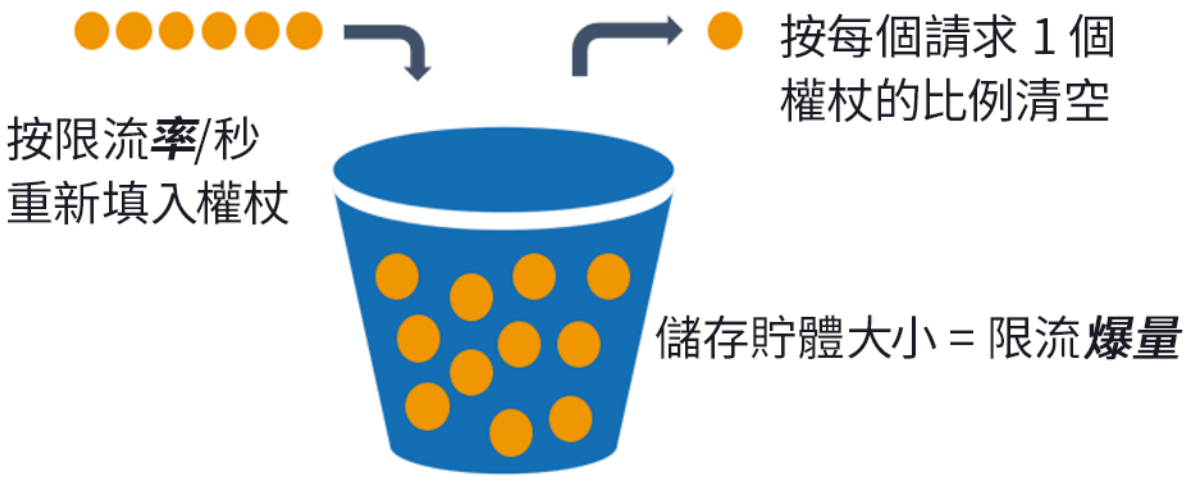
建立此最佳實務的優勢：設定限流限制的工作負載能夠在非預期的數量尖峰情況下正常運作，並成功處理已接受的請求負載。對 APIs和 佇列的請求突然或持續激增會受到限流，不會耗盡請求處理資源。速率會限制個別請求者，以便來自單一 IP 地址或API消費者的大量流量不會影響其他消費者。

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，請考慮實作記號儲存貯體演算法 (在此演算法中，記號對於請求具重要性)。記號會按每秒的限流率重新填入，並依照每個請求一個記號的比例非同步清空。



記號儲存貯體演算法。

Amazon API Gateway 會根據帳戶和區域限制實作權杖儲存貯體演算法，並可透過用量計劃為每個用戶端進行設定。此外，Amazon Simple Queue Service （ Amazon SQS ） 和 Amazon Kinesis 可以緩衝請求，以平穩化請求速率，並允許更高的限流速率來處理請求。最後，您可以使用 實作速率限制AWS WAF，以調節產生異常高負載的特定API取用者。

實作步驟

您可以為 設定API閘道，並在超過限制時APIs傳回429 Too Many Requests錯誤。您可以 AWS WAF 搭配 AWS AppSync 和 API Gateway 端點使用 ，以啟用每個 IP 地址的速率限制。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。

使用非同步處理時，當您將 Amazon 設定為 SQS的事件來源時 AWS Lambda，您可以設定最大並行速率，以避免工作負載或帳戶中其他服務所需的高事件速率耗用可用帳戶並行執行配額。

雖然 API Gateway 提供權杖儲存貯體的受管實作，但如果您無法使用 API Gateway，則可以利用 服務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）。

- 了解並設定每個區域、API每個階段的帳戶層級API閘道限流限制，以及每個用量計劃層級的API金鑰。
- 將AWS WAF 速率限制規則套用至 API Gateway 和 AWS AppSync 端點，以防止洪水並封鎖惡意 IPs。也可以在 A2A 取用者的金鑰上 AWS AppSync API設定速率限制規則。
- 請考慮您是否需要比 的速率限制更多的限流控制 AWS AppSync APIs，如果需要，請在 AWS AppSync 端點前方設定API閘道。
- 當 Amazon SQS佇列設定為 Lambda 佇列取用者的觸發條件時，請將並行上限設定為足以滿足服務層級目標的值，但不會使用影響其他 Lambda 函數的並行限制。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 搭配原生服務與 Amazon SQS或 Kinesis 整合，以緩衝請求。
- 如果您無法使用 API Gateway，請查看語言特定的程式庫，以實作工作負載的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 持續工作
- REL05-BP03 控制和限制重試呼叫

相關文件：

- Amazon API Gateway：提高輸送量的限流API請求
- AWS WAF：以速率為基礎的規則陳述式
- 引入使用 Amazon SQS 作為事件來源 AWS Lambda 時的最大並行
- AWS Lambda：並行上限

相關範例：

- 三個最重要的 AWS WAF 費率型規則
- Java Bucket4j
- Python 記號儲存貯體
- 節點記號儲存貯體
- 。NET 系統執行速率限制

相關影片：

- 使用 實作 GraphQL API安全最佳實務 AWS AppSync

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。

預期結果：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。
- 實作重試，而不測試其效果，或假設重試已內建到 中，SDK而不測試重試案例。
- 未能了解從相依性發布的錯誤代碼，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 SDKs依預設會實作重試和指數退避。在工作負載中適用的情況下，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 SDKs 會針對您選擇的語言實作經過驗證的重試策略，並具有指數退避和抖動，並且比編寫您自己的重試實作更有利。
- 在實作重試之前，請確認服務是冪等的。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。
- 呼叫 AWS 服務 時APIs，請使用 AWS SDKs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 使所有回應都具有同位性
- REL05-BP02 節流請求
- REL05-BP04 快速失敗並限制佇列
- 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：

- 中的錯誤重試和指數退避 AWS
- Amazon 建置者資料中心：逾時、重試、退避與抖動
- 指數退避和抖動
- 以無能方式確保重試安全 APIs

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：介紹 Amazon Builders 程式庫 （DOP328）

相關工具：

- AWS SDKs 和 工具：重試行為
- AWS Command Line Interface： AWS CLI 重試

REL05-BP04 快速失敗並限制佇列

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：

- 實作訊息佇列，但不在DLQ磁碟區上設定無效字母佇列（DLQ）或警示，以偵測系統何時故障。
- 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。
- 當最後的先出（FIFO）佇列更能滿足用戶端需求時，設定先出（LIFO）佇列，例如，當不需要嚴格排序，且待辦項目處理正在延遲所有新的和時間敏感請求，導致所有用戶端發生違反的服務層級時。
- 將內部佇列暴露至用戶端APIs，而不是將管理工作接收並將請求放置在內部佇列中的暴露。
- 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，並且在發布至生產環境之前，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 提供功能來監控故障並發出

警示。已知系統失敗時，可以調用緩解策略，包括背離受損的資源。當系統使用 Amazon SQS 和其他佇列技術實作佇列以順利載入時，他們必須考慮如何管理佇列待處理項目，以及訊息消耗失敗。

實作步驟

- 在軟體中實作程式化判斷提示或特定指標，並使用這些提示或指標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儀器建立指標和警示。
- 使用 CloudWatch 指標和警示來避免增加處理延遲或重複處理請求的受損資源。
- 使用非同步處理，方法是設計APIs接受請求，並使用 Amazon 將請求附加到內部佇列，SQS然後使用成功訊息回應訊息產生用戶端，以便用戶端可以在後端佇列取用者處理請求時釋出資源並繼續進行其他工作。
- 測量和監控佇列處理延遲，方法是在每次從佇列中提取訊息時產生 CloudWatch 指標，方法為立即與訊息時間戳記進行比較。
- 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。此技術是LIFO處理方法的近似值，並允許所有新工作的正常系統處理。
- 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 實作鬆散耦合相依性
- REL05-BP02 節流請求
- REL05-BP03 控制和限制重試呼叫
- REL06-BP02 定義和計算指標 （彙總）
- REL06-BP07 監控 end-to-end透過您的系統追蹤請求

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佇列中訊息的待處理項目增加？

- Elastic Load Balancing：區域轉移
- Amazon Application Recovery Controller：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、目標服務受損或聯網分割失敗，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。

利用如重試和斷路器等建立良好的設計模式，優雅地處理逾時，並支援快速失敗的方法。AWS SDKs 和 AWS CLI 允許設定連線和請求逾時，以及使用指數退避和抖動的重試。 AWS Lambda函數支援逾

時組態，而使用 AWS Step Functions，您可以建置低程式碼斷路器，以利用與服務 AWS 和 預先建置的整合SDKs。 AWS App MeshEnvoy 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 呼叫 時 AWS SDK，請檢閱文件，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 或 AWS CLI 命令時 AWS SDKs，請設定 `connectTimeoutInMillis`和 的 AWS 組態預設值，以設定預設逾時值`tlsNegotiationTimeoutInMillis`。
- 將命令列選項 `cli-connect-timeout`和 套用至 AWS 服務`cli-read-timeout`，以控制一次性 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。
- 對呼叫錯誤率、延遲的服務層級目標和延遲異常值實作CloudWatch 指標和CloudWatch 異常偵測，以深入了解如何管理過度激進或寬鬆的逾時。
- 設定 Lambda 函數的逾時。
- API Gateway 用戶端在處理逾時時必須實作自己的重試。API Gateway 支援下游整合的 50 毫秒至 29 秒整合逾時，在整合請求逾時時不會重試。
- 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。
- 對於基於容器的工作負載，請查看 App Mesh Envoy 功能以利用內建的逾時和斷路器。
- 使用 AWS Step Functions 建置用於遠端服務呼叫的低程式碼斷路器，特別是在呼叫 AWS 原生 SDKs和支援的 Step Functions 整合時，以簡化工作負載。

資源

相關的最佳實務：

- REL05-BP03 控制和限制重試呼叫
- REL05-BP04 快速失敗並限制佇列
- REL06-BP07 監控 end-to-end透過您的系統追蹤請求

相關文件：

- AWS SDK：重試和逾時
- Amazon 建置者資料中心：逾時、重試、退避與抖動
- Amazon API Gateway 配額和重要備註
- AWS Command Line Interface：命令列選項
- AWS SDK for Java 2.x：設定API逾時
- AWS 使用組態物件和組態參考的 Botocore
- AWS SDK for .NET：重試與逾時
- AWS Lambda：設定 Lambda 函數選項

相關範例：

- 搭配 AWS Step Functions 和 Amazon DynamoDB 使用斷路器模式
- Martin Fowler： CircuitBreaker

相關工具：

- AWS SDKs
- AWS Lambda
- Amazon SQS
- AWS Step Functions
- AWS Command Line Interface

REL05-BP06 盡可能讓系統處於無狀態

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，不依賴磁碟和記憶體中本機儲存的資料。這允許伺服器任意置換，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，您就可以使用無伺服器運算服務，例如 AWS Lambda 或 AWS Fargate。

除了伺服器替換之外，無狀態應用程式的另一個優點是他們可以水平擴展，因為任何可用的運算資源（例如EC2執行個體和 AWS Lambda 函數）都可以服務任何請求。

建立此最佳實務的優勢：設計為無狀態的系統更適合水平擴展，因此可以根據波動的流量和需求來新增或移除容量。其本質上也具有抵抗故障的能力，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曝險等級：中

實作指引

讓您的應用程式無狀態。無狀態應用程式支援水平擴展，並且可以容忍單個節點的失敗。分析並了解在架構中維持狀態的應用程式元件。這可協助您評估轉換為無狀態設計的潛在影響。無狀態架構會分離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，並最佳化資源使用率。

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進行分離和管理來解耦資料。
  - 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分離。
  - 可以將密碼儲存在安全的集中位置，使用 AWS Secrets Manager 來分離使用者資料。這意味著應用程式的程式碼不需要存儲密碼，這使得它更安全。
  - 請考慮使用 Amazon S3 來存放大型非結構化資料，例如影像和文件。應用程式可以在需要時擷取此資料，而無需將其存儲在記憶體中。
  - 使用 Amazon DynamoDB 來存放使用者設定檔等資訊。應用程式可以近乎即時的速度查詢這些資料。
- 將工作階段資料卸載至資料庫、快取或外部檔案。
  - Amazon ElastiCache、Amazon DynamoDB 、Amazon Elastic File System （Amazon EFS） 和 Amazon MemoryDB 是可用來卸載工作階段資料 AWS 的服務範例。
- 在確定需要使用所選儲存解決方案維持哪些狀態和使用者資料之後，設計一個無狀態架構。

資源

相關的最佳實務：

- REL11-BP03 在所有圖層上自動復原

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
- 上的無狀態 Web 層最佳實務 AWS

REL05-BP07 實作緊急槓桿

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，可以建立已知的良好流程，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。如需優雅降級的詳細資訊，請參閱 REL05-BP01 實作優雅降級，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：實作緊急控制桿可透過為解析程式提供已確立的程序來應對意外的需求峰值或非關鍵相依性失敗，從而提高工作負載中關鍵元件的可用性。

未建立此最佳實務時的曝險等級：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 關鍵和非關鍵功能的範例，請參閱 Any Day Can Be Prime Day: How Amazon.com Search Uses Chaos Engineering to Handle Over 84K Requests Per Second。
  - 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。如需更多詳細資訊，請參閱 REL11-BP05 使用靜態穩定性來防止雙模式行為 。
- 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)。
  - 這可能包括諸如降載、限流請求或實作適度降級等機制。

資源

相關的最佳實務：

- REL05-BP01 實作優雅降級，將適用的硬相依性轉換為軟相依性
- REL05-BP02 節流請求
- REL11-BP05 使用靜態穩定性來防止雙模式行為

相關文件：

- 自動化安全、無人為介入的部署
- 任何一天都可能是黃金日：Amazon.com 搜尋功能如何使用混沌工程每秒處理 84K 以上的請求

相關影片：

- AWS re：Invent 2020：透過不可變性的可靠性、一致性和信心

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因應需求的變化？
- REL 8. 如何實作變更？

REL 6. 如何監控工作負載資源？

日誌和指標是可深入洞察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 分析日誌
- REL06-BP06 定期執行審核
- REL06-BP07 監控 end-to-end透過您的系統追蹤請求

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 Dashboard 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，描述如何從日誌中擷取指標 (如果需要)，並設定調用對應警示事件的閾值。確保指標與工作負載的關鍵效能指標（KPIs）相關，並使用指標和日誌來識別服務降級的早期警告訊號。例如，與業務結果相關的指標，

例如每分鐘成功處理的訂單數量，可以比技術指標更快地指出工作負載問題，例如CPU使用率。使用 AWS Health Dashboard 來個人化檢視 AWS 資源基礎 AWS 之服務的效能和可用性。

雲端監控提供新機遇。大多數雲端供應商已開發可自訂掛鉤，並提供洞見，以協助您監控工作負載的多個層面。Amazon 等 AWS 服務會 CloudWatch 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、判斷正常基準，並以最少的使用者介入來確定表面異常。異常偵測演算法會考慮指標的季節性和趨勢變化。

AWS 提供大量監控和日誌資訊以供取用，可用於定義工作負載特定的指標、程序和採用機器學習技術， change-in-demand而不論 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，它會定期運行工作負載的用戶端執行的許多常見任務匹配動作。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 可讓您建立合成 Canary 來監控端點和 APIs。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、故障或限流率等問題。

預期成果：

從工作負載的所有元件中收集並使用關鍵指標，以確保工作負載的可靠性和最佳的使用者體驗。偵測工作負載未達成業務成果，可讓您快速宣告災難並從事件中復原。

常見的反模式：

- 僅監控工作負載的外部界面。
- 不會產生任何工作負載特定的指標，僅依賴工作負載使用 AWS 的服務提供給您的指標。
- 僅在工作負載中使用技術指標，而不監控與KPIs工作負載貢獻的非技術相關指標。
- 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：在工作負載中的所有層級進行監控，可讓您更快速地預測和解決構成工作負載的元件中的問題。

未建立此最佳實務時的曝險等級：高

實作指引

1. 在可用的地方開啟日誌記錄。應從工作負載的所有元件中取得監控資料。開啟其他日誌記錄，例如 S3 Access Logs，並允許您的工作負載記錄工作負載特定資料。從 Amazon

CPU、Amazon 、Amazon 、ECSEKSEC2Elastic Load Balancing 和 Amazon 等服務收集 AWS Auto Scaling、網路 I/O 和磁碟 I/O 平均值的指標EMR。如需AWS 將 CloudWatch 指標發佈至 的服務清單，請參閱發佈指標 AWS 的服務 CloudWatch。

2. 檢閱所有預設指標，並探索任何資料收集漏洞。每個服務都會產生預設指標。收集預設指標可讓您進一步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 CloudWatch 使用 AWS CLI 或 來建立和發佈自己的指標API。
3. 評估所有指標，以決定要針對工作負載中的每個 AWS 服務提醒哪些指標。您可以進行選擇，以選取對工作負載可靠性有重大影響的指標子集。專注於重要指標和閾值，可讓您調整提醒的數量，並協助將誤報率降至最低。
4. 在調用提醒後，定義工作負載的提醒和復原程序。定義警示可讓您快速通知、升級和遵循從事件復原的必要步驟，並滿足您指定的復原時間目標（RTO）。您可以使用 Amazon CloudWatch Alarms 來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。綜合監控遵循相同的路由並執行與客戶相同的動作，即使您的工作負載沒有任何客戶流量，也能持續驗證您的客戶體驗。透過使用綜合交易，您可以在客戶之前發現問題。

資源

相關的最佳實務：

- REL11-BP03 自動化所有層的復原

相關文件：

- AWS Health 儀表板入門 – 您的帳戶運作狀態
- AWS 發佈 CloudWatch 指標的服務
- Network Load Balancer 的存取日誌
- Application Load Balancer 的存取日誌
- 存取 的 Amazon CloudWatch Logs AWS Lambda
- Amazon S3 伺服器存取日誌記錄
- 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
- 發佈自訂指標

- 使用 Amazon CloudWatch Dashboards
- 使用 Amazon CloudWatch 指標
- 使用 Canary （ Amazon CloudWatch Synthetics ）
- 什麼是 Amazon CloudWatch Logs？

使用者指南：

- 建立追蹤
- 監控 Amazon EC2 Linux 執行個體的記憶體和磁碟指標
- 將 CloudWatch 日誌與容器執行個體搭配使用
- VPC 流量日誌
- 什麼是 Amazon DevOpsGuru？
- 什麼是 AWS X-Ray？

相關部落格：

- 使用 Amazon CloudWatch Synthetics 和 進行偵錯 AWS X-Ray

相關範例和研討會：

- AWS Well-Architected 實驗室：卓越營運 - 相依性監控
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態
- 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 作為主要彙總和儲存層。對於某些服務，例如 AWS Auto Scaling 和 Elastic Load Balancing ，預設會針對叢集或執行個體的CPU負載或平均請求延遲提供預設指標。對於串流服務，例如VPC流程日誌和 AWS CloudTrail，事件資料會轉送至 CloudWatch 日誌，您需要定義和套用指標篩選條件，從事件資料中擷取指標。這為您提供時間序列資料，可做為您定義叫用警示的 CloudWatch 警示輸入。

未建立此最佳實務時的曝險等級：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在傳送至 CloudWatch Logs 的日誌資料中要尋找的術語和模式。CloudWatch Logs 使用這些指標篩選條件將日誌資料轉換為數值 CloudWatch指標，您可以繪製或設定警示。
    - 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。大多數第三方產品都與 CloudWatch 和 Amazon S3 整合。
  - 某些 AWS 服務可以直接將日誌發佈至 Amazon S3。如果日誌的主要需求是儲存在 Amazon S3 中，則您可以輕鬆讓產生日誌的服務直接將其傳送到 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 進行偵錯 AWS X-Ray
- 一個可觀測性研討會
- 搜尋和篩選日誌資料
- 直接將日誌傳送至 Amazon S3
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態

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 可讓您使用基於靜態閾值、異常偵測和其他條件的 CloudWatch 警示來建立指標和複合警示。如需使用 可設定的警示類型的詳細資訊 CloudWatch，請參閱 CloudWatch 文件 的警示區段。

您可以使用CloudWatch 儀表板 為團隊建構指標和 AWS 資源提醒的自訂檢視。 CloudWatch 主控台中的可自訂首頁可讓您在多個 區域的單一檢視中監控資源。

警示可以執行一或多個動作，例如傳送通知至 Amazon SNS主題 、執行 Amazon EC2 動作或 Amazon EC2 Auto Scaling 動作，或在 中建立 OpsItem或 事件 AWS Systems Manager。

Amazon CloudWatch 使用 Amazon SNS 在警示變更狀態時傳送通知，將訊息從發佈者 （生產者） 傳遞給訂閱者 （消費者）。如需設定 Amazon SNS通知的詳細資訊，請參閱設定 Amazon SNS。

CloudWatch 會在建立、更新、刪除 CloudWatch 警示或其狀態變更時傳送EventBridge事件。您可以使用 EventBridge 這些事件來建立執行動作的規則，例如在警示狀態變更時通知您，或使用 Systems Manager 自動化 自動觸發帳戶中的事件。

何時應使用 EventBridge 或 Amazon SNS？

EventBridge 和 Amazon SNS都可以用來開發事件驅動的應用程式，您的選擇將取決於您的特定需求。

當您想要建置可對來自自己的應用程式、SaaS 應用程式 AWS 和服務的事件做出反應的應用程式時，EventBridge 建議使用 Amazon。 EventBridge 是唯一直接與第三方 SaaS 合作夥伴整合的事件型服

務。 EventBridge 也會自動從 200 多個 AWS 服務擷取事件，而無需開發人員在其帳戶中建立任何資源。

EventBridge 使用定義的 JSON型結構來建立套用在整個事件內文的規則，以選取要轉送至目標的事件。 EventBridge 目前支援超過 20 個 AWS 服務做為目標，包括 AWS Lambda、Amazon SQS、Amazon SNS、Amazon Kinesis Data Streams 和 Amazon Data Firehose 。

對於需要高扇出（數千或數百萬個端點）的應用程式，SNS建議使用 Amazon。我們看到的常見模式是，客戶使用 Amazon SNS 作為規則的目標，以篩選他們所需的事件，並擴展到多個端點。

訊息是非結構化的，可以是任何格式。Amazon SNS支援將訊息轉送至六種不同類型的目標，包括 Lambda、Amazon SQS、HTTP/S 端點、SMS、行動推送和電子郵件。Amazon SNS典型延遲低於 30 毫秒。各種 AWS 服務透過設定服務來傳送 Amazon SNS 訊息（超過 30 個，包括 Amazon EC2、Amazon S3 和 Amazon RDS）。

實作步驟

1. 使用 Amazon 警示 建立 CloudWatch 警示。
   a. 指標警示會根據 CloudWatch 指標監控單一 CloudWatch 指標或表達式。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。此動作可能包含傳送通知至 Amazon SNS主題 、執行 Amazon EC2 動作或 Amazon EC2 Auto Scaling 動作，或在 中建立 OpsItem或 事件 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。複合警示可以在狀態變更時傳送 Amazon SNS通知，並在進入警示狀態時建立 Systems Manager OpsItems或事件，但無法執行 Amazon EC2或 Auto Scaling 動作。
2. 設定 Amazon SNS通知 。建立 CloudWatch 警示時，您可以包含 Amazon SNS主題，以便在警示變更狀態時傳送通知。
3. 在 中建立符合指定警示的規則 EventBridge。 CloudWatch 每個規則都支援多個目標，包括 Lambda 函數。例如，您可以定義當可用磁碟空間不足時啟動的警示，這會透過 EventBridge 規則觸發 Lambda 函數來清理空間。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 使用教戰手冊調查失敗

相關文件：

- Amazon CloudWatch
- CloudWatch 記錄洞察
- 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 記錄資料保護
- Amazon EventBridge
- Amazon Simple Notification Service

相關影片：

- 重塑 2022 年可觀測性影片
- AWS re：Invent 2022 - Amazon 的可觀測性最佳實務

相關範例：

- 一個可觀測性研討會
- AWS Lambda 使用 Amazon CloudWatch Alarms 的意見回饋控制將 Amazon EventBridge 傳送至

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。

預期結果：識別即時警示，並設定警示的自動處理，以調用為維護服務層級目標和服務層級協議而採取的適當動作 （SLAs）。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。
- 運算子收到警示通知時，沒有要遵循的標準操作程序 （ SOPs ）。
- 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：自動化警示處理可改善系統復原能力。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曝險等級：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 中記錄手動步驟，SOP並在這些步驟上訓練運算子。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. 建立警示清查：若要取得所有警示的清單，您可以使用 Amazon CloudWatch AWS CLI 命令 使用 `describe-alarms`。根據您設定的警示數量，您可能必須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來使用API呼叫 來取得警示。
2. 記錄所有警報動作：更新執行手冊與所有警示及其動作，無論其為手動還是自動。AWS Systems Manager 可提供預先定義的執行手冊。如需有關執行手冊的詳細資訊，請參閱 Working with runbooks。如需有關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：針對任何需要動作的警示，請使用 指定自動動作 CloudWatch SDK。例如，您可以建立和啟用 CloudWatch 警示上的動作，或停用警示上的動作，以根據警示自動變更 Amazon EC2執行個體的狀態。

   您也可以使用 Amazon EventBridge 自動回應系統事件，例如應用程式可用性問題或資源變更。您可建立規則來指示您在意的事件，以及當事件符合規則時執行的動作。可自動啟動的動作包括叫

用AWS Lambda函數、叫用 Amazon EC2 Run Command、將事件轉送至 Amazon Kinesis Data Streams 以及EC2使用 來查看 Automate Amazon EventBridge。

4. 標準操作程序 （SOPs）：根據您的應用程式元件， AWS Resilience Hub建議多個SOP範本 。您可以使用這些SOPs來記錄操作員在發出警示時應遵循的所有程序。您也可以根據 Resilience Hub 建議建置 SOP，其中您需要具有相關復原政策的 Resilience Hub 應用程式，以及針對該應用程式的歷史復原評估。您的 建議是由彈性評估所SOP產生。

   Resilience Hub 與 Systems Manager 合作，提供許多文件SOPs來自動執行 的步驟，您可以用這些SSM文件作為 的基礎SOPs。例如，Reilience Hub 可能會建議根據現有SSM自動化文件SOP新增磁碟空間。

5. 使用 Amazon DevOpsGuru 執行自動動作：您可以使用 Amazon DevOpsGuru 自動監控應用程式資源是否有異常行為，並提供目標性建議，以加快問題識別和修復時間。使用 DevOpsGuru，您可以近乎即時地監控來自多個來源的操作資料串流，包括 Amazon CloudWatch 指標、AWS Config、AWS CloudFormation和 AWS X-Ray。您也可以使用 DevOpsGuru 自動在 OpsItems中建立事件，OpsCenter 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 Runbook 用於部署等標準活動

相關文件：

- AWS Systems Manager 自動化
- 從資源建立在事件 AWS 上觸發的 EventBridge 規則
- 一個可觀測性研討會
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態
- 什麼是 Amazon DevOpsGuru？
- 使用自動化文件 (手冊)

相關影片：

- AWS re：Invent 2022 - Amazon 的可觀測性最佳實務
- AWS re：Invent 2020：自動化任何 AWS Systems Manager
- 簡介 AWS Resilience Hub
- 為 Amazon DevOpsGuru Notifications 建立自訂票證系統
- 使用 Amazon DevOpsGuru 啟用多帳戶洞見彙總

相關範例：

- 可靠性研討會
- Amazon CloudWatch 和 Systems Manager 研討會

REL06-BP05 分析日誌

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，可用於分析日誌資料。Amazon CloudWatch Logs 也支援訂閱，允許資料順暢流至 Amazon S3，您可以在其中使用 或 Amazon Athena 查詢資料。它也支援對大量格式的查詢。如需詳細資訊，請參閱 Amazon Athena 使用者指南中的支援 SerDes 和資料格式。若要分析大型日誌檔案集，您可以執行 Amazon EMR叢集來執行 PB 規模分析。

AWS 合作夥伴和第三方提供許多工具，允許彙總、處理、儲存和分析。這些工具包括 New Relic、Splunk、Logly、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。

未建立此最佳實務時的曝險等級：中

實作指引

- CloudWatch Logs Insights 可讓您以互動方式搜尋和分析 Amazon Logs 中的 CloudWatch 日誌資料。

  - 使用 Logs Insights 分析 CloudWatch 日誌資料
  - Amazon CloudWatch Logs Insights 範例查詢
- 使用 Amazon CloudWatch Logs 將日誌傳送至 Amazon S3，您可以在其中使用 或 Amazon Athena 查詢資料。
  - 我要如何使用 Athena 來分析 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。這麼做可降低 Athena 分析每個查詢時的資料量。
      - 如何建立 S3 儲存貯體的生命週期政策？

資源

相關文件：

- Amazon CloudWatch Logs Insights 範例查詢
- 使用 Logs Insights 分析 CloudWatch 日誌資料
- 使用 Amazon CloudWatch Synthetics 和 進行偵錯 AWS X-Ray
- 如何建立 S3 儲存貯體的生命週期政策？
- 我要如何使用 Athena 來分析 Amazon S3 伺服器存取日誌？
- 一個可觀測性研討會
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態

REL06-BP06 定期執行審核

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故障發生時出現的情況。 AWS 提供的服務可讓您追蹤事件期間的服務狀態：

- Amazon CloudWatch Logs：您可以將日誌存放在此服務中，並檢查其內容。
- Amazon CloudWatch Logs Insights ： 是一項完全受管的服務，可讓您在幾秒鐘內分析大量日誌。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 AWS APIs在什麼時間和什麼主體叫用哪些 。

在 AWS，我們會進行每週會議，以檢閱營運績效，並在團隊之間分享學習成果。由於 中有這麼多團隊 AWS，因此我們建立了 Wheel 來隨機挑選要檢閱的工作負載。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。

常見的反模式：

- 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反應通知。

未建立此最佳實務時的曝險等級：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。您也應有可以檢查各種應用程式層和相依性的儀表板。
  - 使用 Amazon CloudWatch Dashboards
- 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，以查看是否有趨勢可能指出某項需要調查的事務。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。平均值或中位數可能會掩蓋異常值和異常。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 進行偵錯 AWS X-Ray
- 一個可觀測性研討會
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態

- 使用 Amazon CloudWatch Dashboards

REL06-BP07 監控 end-to-end透過您的系統追蹤請求

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。

預期結果：具有所有元件全面追蹤的工作負載易於偵錯，透過簡化根本原因探索來改善錯誤和延遲的平均解決時間（MTTR）。 End-to-end追蹤可減少探索受影響元件所需的時間，並深入探索錯誤或延遲的詳細根本原因。

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，如果沒有追蹤 AWS Lambda，團隊可能無法清楚了解因頻繁工作負載冷啟動所造成的延遲。
- 合成 Canary 或真實使用者監控（RUM）未設定追蹤。如果沒有 Canary 或 RUM，追蹤分析會省略用戶端互動遙測，產生不完整的效能描述檔。
- 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，雲端原生追蹤SDKs應用於非雲端原生或第三方工具的儀器元件，應設定為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，包括個別元件與日誌記錄、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。諸如何時應調用災難復原 (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)。在追蹤組態中 end-to-end包含合成 Canary 和真實使用者監控，以測量遠端用戶端互動和延遲，以便您可以根據服務層級協議和目標準確評估系統效能。

您可以使用 AWS X-Ray和 Amazon CloudWatch Application Monitoring 測試服務，在請求通過應用程式時提供完整的檢視。X-Ray 會收集應用程式遙測，並可讓您針對無程式碼或低程式碼的系統元件，將 APIs和 視覺化並進行篩選。 CloudWatch 應用程式監控包括 ServiceLens 整合您的追蹤與指標、日

誌和警示。 CloudWatch 應用程式監控還包括用於監控端點和 的合成APIs，以及用於測試 Web 應用程式用戶端的真實使用者監控。

實作步驟

- 在 Amazon S3 和 Amazon Gateway 等所有支援的原生服務 AWS X-Ray 上使用 。 Amazon S3 AWS Lambda API AWS 這些服務使用基礎設施作為程式碼 AWS SDKs或 啟用具有組態切換的 X-Ray AWS Management Console。
- 檢測應用程式 AWS Distro for Open Telemetry 和 X-Ray 或第三方收集代理程式。
- 檢閱 AWS X-Ray 開發人員指南，了解程式設計語言特定實作。這些文件章節詳細說明如何針對您的應用程式程式設計語言執行HTTP請求、SQL查詢和其他程序。
- 使用適用於 Amazon CloudWatch Synthetic Canary 和 Amazon CloudWatch RUM 的 X-Ray 追蹤，透過下游 AWS 基礎設施來分析最終使用者用戶端的請求路徑。
- 根據資源運作狀態和 Canary 遙測設定 CloudWatch 指標和警示，以便團隊快速收到問題警示，然後可以透過 深入探索追蹤和服務地圖 ServiceLens。
- 如果將第三方工具用於主要追蹤解決方案，請為第三方追蹤工具 (例如 Datadog、New Relic 或 Dynatrace) 啟用 X-Ray 整合。

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 進行偵錯 AWS X-Ray
- Amazon 建置者資料中心：偵測分散式系統，以了解運作狀態
- AWS X-Ray 與其他 AWS 服務整合
- AWS 適用於 OpenTelemetry 和 的 Distro AWS X-Ray
- Amazon CloudWatch：使用合成監控
- Amazon CloudWatch：使用 CloudWatch RUM

- 設定 Amazon CloudWatch 合成 Canary 和 Amazon CloudWatch 警示
- 可用性及更多：了解和改善 上分散式系統的復原能力 AWS

相關範例：

- 一個可觀測性研討會

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因應需求的變化？

可擴展的工作負載提供可自動新增或移除資源的彈性，讓資源能夠在任何特定時間點充分滿足目前的需求。

最佳實務

- REL07-BP01 在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 在偵測到工作負載需要更多資源時取得資源
- REL07-BP04 Load 測試您的工作負載

REL07-BP01 在取得或擴展資源時使用自動化

在取代資源受損或擴展工作負載時，請使用 Amazon S3 和 等受管 AWS 服務來自動化程序 AWS Auto Scaling。您也可以使用第三方工具 和 AWS SDKs 來自動化擴展。

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 可讓您偵測和取代受損的執行個體。它還可讓您為包括 Amazon EC2執行個體和 Spot Fleets、Amazon ECS任務、Amazon DynamoDB 資料表和索引，以及Amazon Aurora 複本在內的資源建立擴展計劃。

擴展EC2執行個體時，請確定您使用多個可用區域（最好至少三個），並新增或移除容量以維持這些可用區域的平衡。ECS 任務或 Kubernetes Pod（使用 Amazon Elastic Kubernetes Service 時）也應分散到多個可用區域。

使用 時 AWS Lambda，執行個體會自動擴展。每次收到您函數的事件通知時， 都會 AWS Lambda 快速找到其運算機群中的可用容量，並將程式碼執行到配置的並行。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，您的應用程式可以實現儲存貯體中每個字首每秒至少 3，500 PUTCOPY/POST/DELETE 或 5，500 GET/HEAD 請求。在儲存貯體中的字首數不受限制。您可以並行讀取以提升您的讀取或寫入的效能 例如，如果您在 Amazon S3 儲存貯體裡建立 10 個字首，平行讀取，您可以縮放讀取效能至每秒 55,000 讀取要求。

設定和使用 Amazon CloudFront 或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，並可以處理快取中內容的請求，因此減少擴展工作負載的需求。

常見的反模式：

- 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：自動化可消除在部署和停用資源時可能出現的手動錯誤。自動化可消除因部署或停用需求回應緩慢而造成成本超支和拒絕服務的風險。

未建立此最佳實務時的曝險等級：高

實作指引

- 配置和使用 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。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？
    - 在適用的 Amazon EC2執行個體和 Spot Fleets、Amazon ECS任務、Amazon DynamoDB 資料表和索引、Amazon Aurora 複本和 AWS Marketplace 設備上設定 Auto Scaling。

    - 使用 DynamoDB Auto Scaling 功能自動管理輸送容量
        - 使用服務API操作來指定警示、擴展政策、暖機時間和冷卻時間。
- 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 Balancer 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 Balancer 可用於以與 整合的方式分配負載 AWS Auto Scaling ，以管理需求。
                    - 將負載平衡器與 Auto Scaling 群組配合使用
        - 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 將流量分配至不同的工作負載TCP，或為您的工作負載設定一組固定的 IP 地址。
                - Network Load Balancer 可用來以與 整合的方式分配負載 AWS Auto Scaling ，以管理需求。
- 使用高可用性DNS提供者。DNS 名稱可讓您的使用者輸入名稱，而不是 IP 地址來存取工作負載，並將此資訊分發到定義的範圍，通常針對工作負載的使用者全域分發。
    - 使用 Amazon Route 53 或信任的DNS提供者。
        - 什麼是 Amazon Route 53？
    - 使用 Route 53 管理您的 CloudFront 分佈和負載平衡器。
        - 確定要管理的網域和子網域。
        - 使用 ALIAS或 記錄建立適當的CNAME記錄集。
            - 處理記錄
- 使用 AWS 全域網路最佳化從使用者到應用程式的路徑。 AWS Global Accelerator 會持續監控應用程式端點的運作狀態，並在不到 30 秒內將流量重新導向至運作狀態良好的端點。
    - AWS Global Accelerator 是一種服務，可改善本機或全域使用者的應用程式的可用性和效能。它提供靜態 IP 地址，作為單一或多個 中應用程式端點的固定進入點 AWS 區域，例如 Application Load Balancer、Network Load Balancer 或 Amazon EC2執行個體。
        - 什麼是 AWS Global Accelerator？
- 設定和使用 Amazon CloudFront 或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 分佈，或使用第三方 CDN。
    - 您可以限制對工作負載的存取，以便只能 CloudFront 使用端點安全群組或存取政策 CloudFront 中的 的 IP 範圍從 存取工作負載。

資源

相關文件：

- APN 合作夥伴：可協助您建立自動化運算解決方案的合作夥伴
- AWS Auto Scaling：擴展計劃的運作方式
- AWS Marketplace：可與 Auto Scaling 結合使用的產品
- 使用 DynamoDB Auto Scaling 功能自動管理輸送容量
- 將負載平衡器與 Auto Scaling 群組配合使用
- 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 修改 DynamoDB 資料表的輸送量 AWS CLI）。但是，應盡可能使用自動化 (請參閱取得或擴展資源時使用自動化)。

預期成果：啟動擴展活動 (自動或手動)，以在偵測到故障或客戶體驗降級時恢復可用性。

未建立此最佳實務時的曝險等級：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。定義手動或自動化程序，以擴展所需的資源。 o 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失敗。

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 的元件 AWS Auto Scaling 可以使用擴展計畫來設定一組擴展資源的指示。如果您使用 AWS CloudFormation 或將標籤新增至 AWS 資源，您可以為每個應用程式設定不同資源集的擴展計劃。Auto Scaling 為針對每個資源自訂的擴展策略提供建議。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。有關詳細資訊，請參閱 How scaling plans work。
    - Amazon EC2 Auto Scaling 會驗證您是否擁有正確數量的 Amazon EC2執行個體，以處理應用程式的負載。您可以建立EC2執行個體集合，稱為 Auto Scaling 群組。您可以在每個 Auto Scaling 群組中指定執行個體數量下限和上限，Amazon EC2 Auto Scaling 可確保您的群組永遠不會低於或高於這些限制。如需更多詳細資訊，請參閱什麼是 Amazon EC2 Auto Scaling ？
    - Amazon DynamoDB Auto Scaling 功能使用 Application Auto Scaling 服務代您動態調整佈建的輸送容量，以此回應實際流量模式。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，以處理突然增加的流量，而不需限流。如需詳細資訊，請參閱 Managing throughput capacity automatically with DynamoDB auto scaling。

資源

相關的最佳實務：

- REL07-BP01 在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：擴展計劃的運作方式

- 使用 DynamoDB Auto Scaling 功能自動管理輸送容量
- 什麼是 Amazon EC2 Auto Scaling

REL07-BP03 在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 服務會自動擴展以滿足需求。如果使用 Amazon EC2執行個體或 Amazon ECS叢集，您可以根據與工作負載需求對應的用量指標，設定自動擴展。對於 Amazon EC2，可以使用平均CPU使用率、負載平衡器請求計數或網路頻寬來擴展（或縮減）EC2執行個體。對於 Amazon ECS，可以使用平均CPU使用率、負載平衡器請求計數和記憶體使用率來擴展（或縮減）ECS任務。在 上使用目標 Auto Scaling AWS，自動擴展器的作用就像家用調溫器，新增或移除資源以維持您指定的目標值（例如 70% CPU使用率）。

Amazon EC2 Auto Scaling 也可以 doPredictive Auto Scaling ，它使用機器學習來分析每個資源的歷史工作負載，並定期預測未來的負載。

Little's Law 有助於計算您需要的運算執行個體數量（EC2 執行個體、並行 Lambda 函數等）。

L = λW

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。

未建立此最佳實務時的曝險等級：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自動擴展的排程擴展。
    - Amazon EC2 Auto Scaling 的排程擴展
  - 使用 AWS 預測擴展。

- Amazon EC2 Auto Scaling 的預測擴展

資源

相關文件：

- AWS Marketplace：可與 Auto Scaling 結合使用的產品
- 使用 DynamoDB Auto Scaling 功能自動管理輸送容量
- 採用Machine LearningEC2技術的 Predictive Scaling
- 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 Load 測試您的工作負載

採用負載測試方法來衡量擴展活動是否滿足工作負載要求。

重要的是執行持續的負載測試。負載測試應探索突破點，並測試工作負載的效能。 AWS 可讓您輕鬆設定臨時測試環境，以建立生產工作負載規模的模型。在雲端，您可隨需建立生產規模的測試環境、完成測試，然後再停用資源。因為您只為執行中的測試環境付費，所以能以與內部部署測試相較之下相當微小比例的成本來模擬即時環境。

在生產系統承受壓力的演練日，以及客戶使用量較低的時間內，您也應考慮在生產中進行負載測試，並且讓可用的所有人員解釋結果並解決所發生的任何問題。

常見的反模式：

- 在與生產組態不同的部署上執行負載測試。
- 僅對工作負載的個別部分執行負載測試，而非整個工作負載。
- 使用一部分請求而非代表性的一組實際請求來執行負載測試。
- 對高於預期負載的小型安全係數執行負載測試。

建立此最佳實務的優勢：您會知道架構中的哪些元件在負載時失敗，並能夠識別要監看哪些指標，指出您正在及時處理該負載來解決問題，避免受到該故障的影響。

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須新增或移除資源。
  - 上的分散式負載測試 AWS：模擬數千個連線使用者
    - 識別請求混合。您可能會有不同的請求混合，因此您應該在識別流量混合時查看各種時間範圍。
    - 實作載入驅動程式。您可以使用自訂程式碼、開放原始碼或商業軟體實作載入驅動程式。
    - 最初使用小容量的負載測試。您將負載驅動到較小容量 (可能和單一執行個體或容器一樣小)，看到一些立即的影響。
    - 針對較大容量的負載測試。在分散式負載上的效果會有所不同，因此您必須盡可能在接近產品環境的條件下進行測試。

資源

相關文件：

- 上的分散式負載測試 AWS：模擬數千個連線的使用者
- 負載測試應用程式

相關影片：

- AWS Summit ANZ 2023：透過 AWS 分散式負載測試，放心加速

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，那麼就很難預測這些變更的影響，也很難解決由於這些變更而產生的問題。

最佳實務

- REL08-BP01 將 Runbook 用於部署等標準活動
- REL08-BP02 整合功能測試作為部署的一部分
- REL08-BP03 整合彈性測試作為部署的一部分
- REL08-BP04 使用不可變的基礎設施部署

- REL08-BP05 透過自動化部署變更

REL08-BP01 將 Runbook 用於部署等標準活動

執行手冊是實現特定結果的預定義程序。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、修補工作負載或修改DNS。

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，從有效的手動過程開始，以程式碼實作它，並在適當時調用它以自動執行。

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。

常見的反模式：

- 在生產環境中對組態執行非計劃中的變更。
- 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增強信心。

未建立此最佳實務時的曝險等級：高

實作指引

- 透過在執行手冊中記錄程序，對熟知的事件提供一致且迅速的回應。
  - 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）定義您的基礎設施，您可以使用版本控制軟體來版本控制和追蹤變更。
  - 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，建立單一且分離的範本。

- 確定實作的許可、範本和負責方。
  - 使用 控制存取 AWS Identity and Access Management
- 使用來源控制，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：可協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： Runbook
- 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 整合功能測試作為部署的一部分

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。

預期成果：您可以使用自動化來執行功能測試，相關的測試資料可減少測試持續時間和費用，並提高測試結果的準確性。可以在部署過程中整合功能測試，以協助您自動化發行管道，以快速且可靠地更新應用程式和基礎設施。

常見的反模式：

- 可以在部署管道之外手動執行測試。
- 可以透過手動緊急工作流程，略過自動化中的測試步驟。
- 您不會遵循既定的測試計畫和流程，以加快時間表。

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需求運作。其用於一致地驗證 元件的預期工作順序APIs，例如使用者介面、、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

會驗證每個功能的行為是否如預期，以保護使用者期望和軟體的完整性。將功能測試整合為常規部署的一部分，並使用自動化來部署所有變更，從而減少引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。如果未符合成功條件，則管道會停止或復原。 AWS CodePipeline 提供自動化測試的持續交付管道，讓測試人員能夠自動化整個測試和部署程序。它與 AWS CodeBuild 和 等 AWS 服務整合 AWS CodeDeploy ，以自動化軟體開發生命週期的建置、測試和部署階段。

實作步驟

- 設定管道：使用 AWS CodePipeline 主控台或 AWS Command Line Interface （ ） 設定來源、建置、測試和部署階段CLI。
  - 定義來源：使用 AWS CodePipeline，您可以自動從版本控制系統擷取來源程式碼，例如 GitHub、 AWS CodeCommit或 Bitbucket，這會驗證最新的程式碼是否一律用於測試。
  - 自動化建置和測試： AWS CodeBuild 可以自動建置和測試程式碼，並產生測試報告。它支援常見的測試架構JUnit，例如 NUnit、 和 TestNG 。
  - 部署您的程式碼：程式碼建立並測試完成後， AWS CodeDeploy 就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以及每個階段的狀態。還可以根據測試執行狀態使用質量檢查來阻止管道。也可以接收任何管道階段失敗或管線完成的通知。

資源

相關文件：

- AWS CodePipeline 搭配 使用 AWS CodeBuild 來測試程式碼和執行組建
- 在 中登入和監控 AWS CodeBuild
- 功能測試指標

REL08-BP03 整合彈性測試作為部署的一部分

通過有意識地在系統中引入故障來整合彈性測試，以在破壞性情況下衡量其能力。彈性測試與通常整合在部署週期中的單元和功能測試不同，因為它們專注於識別系統中的意外故障。雖然從生產前的彈性測試整合開始是安全的，但應設定一個目標，在生產環境中實作這些測試，作為演練日的一部分。

預期成果：彈性測試有助於建立系統承受生產降級能力的信心。實驗可識別會導致故障的弱點，這有助於您改善系統，以自動且有效地緩解故障和降級。

常見的反模式：

- 在部署過程中缺乏可觀測性和監控
- 依賴人類解決系統故障
- 品質不佳的分析機制
- 專注於系統中的已知問題，缺乏識別任何未知因素的實驗
- 可識別故障，但沒有解決方案
- 沒有調查結果和執行手冊文件

建立最佳實務的優勢：在部署中整合的彈性測試有助於識別系統中未知的問題，否則這些問題會被忽視，從而導致生產中停機。在系統中識別這些未知因素可協助您記錄調查結果、將測試整合到您的 CI/CD 程序中以及建置執行手冊，透過高效率、可重複的機制簡化緩解作業。

未建立此最佳實務時的曝險等級：中

實作指引

可以整合到系統部署中的最常見彈性測試表單是災難復原和混沌工程。

- 在任何重大部署中，包含災難復原計劃和標準操作程序 （SOPs） 的更新。
- 將可靠性測試整合至您的自動化部署管道。諸如 AWS Resilience Hub 的服務可整合到您的 CI/CD 管道中，以建立持續的彈性評估，並在每次部署中自動進行評估。
- 在 中定義您的應用程式 AWS Resilience Hub。復原能力評估會產生程式碼片段，協助您將復原程序建立為應用程式的 AWS Systems Manager 文件，並提供建議的 Amazon CloudWatch 監控器和警示清單。
- 您的 DR 計劃和SOPs更新後，請完成災難復原測試，以確認它們是否有效。災難復原可協助您判斷是否可以在事件發生後還原系統並恢復正常操作。您可以模擬各種災難復原策略，並確定您的規劃是否足以滿足您的正常運行時間需求。常見的災難復原策略包括備份與還原、指示燈、冷待命、暖待命、熱待命和主動-主動式，而且成本和複雜性都有所不同。在災難復原測試之前，建議您定義復原時間目標 （RTO） 和復原點目標 （RPO），以簡化要模擬的策略選擇。 AWS 提供災難復原工具AWS Elastic Disaster Recovery，例如協助您開始規劃和測試。
- 混沌工程實驗引入了系統中斷，例如網路中斷和服務故障。透過模擬受控故障，您可以發現系統的漏洞，同時控制注入故障的影響。就像其他策略一樣，在非生產環境中使用諸如 AWS Fault Injection Service 等服務執行受控故障模擬，以在部署到生產環境之前獲得信心。

資源

相關文件：

- 使用彈性測試進行故障實驗以建立復原備
- 使用 AWS Resilience Hub 和 持續評估應用程式復原能力 AWS CodePipeline
- 上的災難復原 （DR） 架構 AWS，第 1 部分：雲端復原的策略
- 使用混沌工程確認工作負載的彈性
- 混沌工程的原則
- 混沌工程研討會

相關影片：

- AWS re:Invent 2020：使用混沌工程測試彈性
- 透過 AWS Fault Injection Service 改善應用程式彈性
- 使用 準備和保護您的應用程式免受中斷 AWS Resilience Hub

REL08-BP04 使用不可變的基礎設施部署

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，不允許就地修改以執行工作負載內的基礎設施資源。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：由於跨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取代為已知且版本控制的組態，可將基礎設施設為已知、經過測試且可信狀態，以避免組態偏移。
- 可靠的不可分割部署：部署可以順利完成，也可以保持不變，從而提高部署流程的一致性和可靠性。
- 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。
- 增強型安全狀態：不允許變更基礎設施，即可停用遠端存取機制（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曝險等級：中

實作指引

自動化

定義不可變基礎設施部署策略時，建議盡可能使用自動化來提高可重複性並將人為錯誤的可能性降至最低。如需更多詳細資訊，請參閱 REL08-BP05 使用自動化和自動化安全的移手部署來部署變更https://aws.amazon.com/builders-library/automating-safe-hands-off-deployments/。

使用基礎設施即程式碼 (IaC)，基礎設施佈建、協同運作和部署步驟會以程式設計、描述性和宣告式的方式定義，並儲存在原始檔控制系統中。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：

Canary 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 (Canary) 上執行。之後，您可以仔細檢查所產生的任何行為變更或錯誤。如果遇到嚴重問題，可以從 Canary 中刪除流量，然後將使用者傳送回以前的版本。如果部署成功，您可以繼續以所需的速度部署，同時監控變更是否有錯誤，直到您完全部署為止。 AWS CodeDeploy 可以使用允許 Canary 部署的部署組態進行設定。

藍/綠部署：與 Canary 部署類似，不同之處在於整個應用程式須並行部署。您可在兩個堆疊 (藍色和綠色) 之間交替部署。再次強調，您可以將流量傳送到新版本，且如果發現部署問題，則可以回復到舊版本。通常所有流量都會一次切換，但您也可以使用每個版本的流量部分，使用 Amazon Route 53 加權DNS路由功能來調用新版本的採用， AWS CodeDeploy 並且AWS Elastic Beanstalk可以使用允許藍/綠部署的部署組態進行設定。

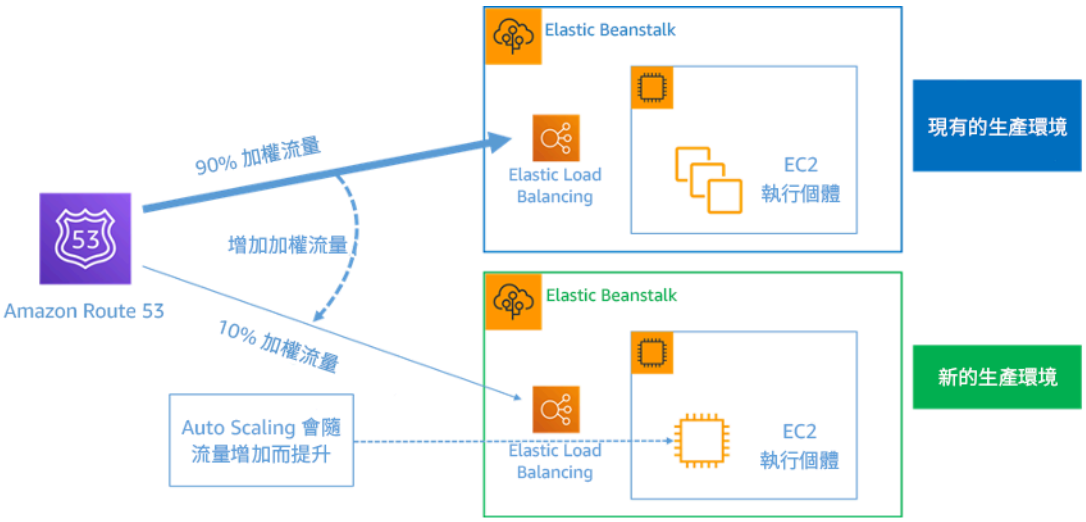


圖 8：使用 AWS Elastic Beanstalk 和 Amazon Route 53 進行藍/綠部署

漂移偵測

漂移被定義為導致基礎設施資源具有與預期不同的狀態或配置的任何變更。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 ） 來指定誰或什麼可以存取 中的服務和資源 AWS、集中管理精細許可，以及分析跨 精簡許可的存取權 AWS。
- 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AWS 白皮書 DevOps 上的 簡介中所述，自動化是 AWS 服務的基石，且在所有 服務、功能和產品中都支援內部。
  - 預先封裝 Amazon Machine Image （ AMI ） 可以加快啟動的時間。EC2 Image Builder 是一項完全受管 AWS 的服務，可協助您自動建立、維護、 up-to-date驗證、共用和部署自訂、安全和 Linux 或 Windows 自訂 AMI。
  - 支援自動化的一些服務包括：
    - AWS Elastic Beanstalk 是一項服務，可在熟悉的伺服器上快速部署和擴展使用 Java、.NET、PHP、Node.js、Python、Ruby、Go 和 Docker 開發的 Web 應用程式，例如 Apache、NGINX、 Passenger 和 IIS。
    - AWS Proton 協助平台團隊連線和協調開發團隊在基礎設施佈建、程式碼部署、監控和更新所需的所有不同工具。 AWS Proton 啟用自動化基礎設施，作為無伺服器和容器型應用程式的程式碼佈建和部署。

- 將基礎設施用作程式碼可讓您輕鬆自動化基礎設施部署，並協助實現基礎設施不可變性。 AWS 提供以程式設計、描述性且明確的方式啟用基礎設施建立、部署和維護的服務。
  - AWS CloudFormation 協助開發人員以有序且可預測的方式建立 AWS 資源。資源使用 JSON或YAML 格式以文字檔案寫入。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。您可以在 JSON中或YAML使用任何程式碼編輯器撰寫資源，例如 AWS Cloud9，檢查它是否進入版本控制系統，然後以安全、可重複的方式 CloudFormation 建置指定的服務。
  - AWS Serverless Application Model （ AWS SAM ） 是一種開放原始碼架構，可用來在上建置無伺服器應用程式 AWS。與其他 AWS 服務 AWS SAM 整合，也是 的延伸 AWS CloudFormation。
  - AWS Cloud Development Kit (AWS CDK) 是一個開放原始碼軟體開發架構，可用來建模，並使用熟悉的程式設計語言來佈建您的雲端應用程式資源。您可以使用 AWS CDK 在背景 AWS CloudFormation 中使用 TypeScript、Python、Java 和 .NET. AWS CDK uses 來建立應用程式基礎設施的模型，以安全且可重複的方式佈建資源。
  - AWS Cloud Control API 推出一組常見的建立、讀取、更新、刪除和清單 （ CRUDL ） APIs，協助開發人員以簡單且一致的方式管理其雲端基礎設施。雲端控制API通用APIs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。
  - Canary 部署：
    - 設定 API Gateway Canary Release 部署
    - ECS使用 為 Amazon 建立具有 Canary 部署的管道 AWS App Mesh
  - 藍/綠部署：AWS 白皮書上的藍/綠部署說明實作藍/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 Detecting unmanaged configuration changes to stacks and resources。

資源

相關的最佳實務：

- REL08-BP05 透過自動化部署變更

相關文件：

- 自動化安全、無人為介入的部署
- 利用 AWS CloudFormation Nubank 建立不可變的基礎設施

- 基礎設施即程式碼
- 實作警示以自動偵測 AWS CloudFormation 堆疊中的偏離

相關影片：

- AWS re：Invent 2020：透過不可變性的可靠性、一致性和信心

REL08-BP05 透過自動化部署變更

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。

期望成果：可以透過廣泛的生產前測試、自動復原和交錯的生產部署，在發佈過程中建置自動化部署安全性。這種自動化可將部署失敗所造成的潛在影響降到最低，而且開發人員不再需要主動觀察部署到生產環境的情況。

常見的反模式：

- 可以執行手動變更。
- 可以透過手動緊急工作流程，略過自動化中的步驟。
- 您不會遵循既定的計畫和流程，以加快時間表。
- 可以在不考慮封裝時間的情況下執行快速後續部署。

建立此最佳實務的優勢：當您使用自動化來部署所有變更時，可以移除導致人為錯誤的可能性，並能夠在變更生產之前進行測試。在生產推送之前執行此程序可驗證您的計畫是否已完成。此外，自動復原至您的發佈程序可以識別生產問題，並將工作負載恢復到先前的作業狀態。

未建立此最佳實務時的曝險等級：中

實作指引

自動化您的部署管道。部署管道讓您可以調用自動測試、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。其中不可或缺的一部分就是採用持續整合和持續交付/部署 (CI/CD) 的文化，在這種文化中，提交或程式碼變更會經過各種自動化階段，從建置和測試階段到生產環境的部署。

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，但是出於這個原因，我們建議您能自動化最困難的程序。

實作步驟

可以依照下列步驟自動化部署以移除手動作業：

- 設定程式碼儲存庫以安全地存儲程式碼：使用 AWS CodeCommit 建立安全的基於 Git 的儲存庫。
- 設定持續整合服務以編譯原始程式碼、執行測試和建立部署成品：若要為此設定建置專案，請參閱AWS CodeBuild 開始使用主控台 。
- 設定部署服務以自動化應用程式部署，並處理應用程式更新的複雜性，而不依賴易出錯的手動部署：將軟體部署AWS CodeDeploy自動化至各種運算服務，例如 Amazon EC2、AWS Lambda、AWS Fargate和您的內部部署伺服器。若要設定這些步驟，請參閱 入門 CodeDeploy。
- 設定持續交付服務，自動化您的發行管道，以便實現更快且更可靠的應用程式和基礎設施更新：請考慮使用 AWS CodePipeline 來協助您自動化發行管道。如需更多詳細資訊，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 持續交付巢狀 AWS CloudFormation 堆疊 AWS CodePipeline
- 使用 AWS CodeCommit、 AWS CodeBuildAWS CodeDeploy和 完成 CI/CD AWS CodePipeline
- APN 合作夥伴：可協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 修補程式管理員
- 什麼是 AmazonSES？
- 什麼是 Amazon Simple Notification Service？

相關影片：

- AWS 2019 年高峰會： 上的 CI/CD AWS

故障管理

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 （ ） 的需求RPO。

最佳實務

- REL09-BP01 識別和備份需要備份的所有資料，或從來源複製資料
- REL09-BP02 安全並加密備份
- REL09-BP03 自動執行資料備份
- REL09-BP04 定期復原資料，以確認備份完整性和程序

REL09-BP01 識別和備份需要備份的所有資料，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：已根據重要性識別並分類資料來源。然後，根據 建立資料復原策略RPO。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。在資料遺失的情況下，實作的策略允許復原或複製已定義 RPO和 內的資料RTO。

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。
- 沒有定義的 RPO，或備份頻率不符合 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：確定需要備份的位置並實作建立備份的機制，或是能夠從外部來源重製資料，這可提升在中斷時還原及復原資料的能力。

未建立此最佳實務時的曝險等級：高

實作指引

AWS 所有資料存放區都提供備份功能。Amazon RDS和 Amazon DynamoDB 等服務另外支援自動備份，允許 point-in-time復原 （ PITR ），這可讓您在目前時間之前最多五分鐘或更短的時間還原備份。許多 AWS 服務提供將備份複製到另一個 的功能 AWS 區域。 AWS Backup 是一項工具，可讓您集中並自動化跨 AWS 服務的資料防護。 AWS Elastic Disaster Recovery可讓您複製完整的伺服器工作負載，並維持對內部部署、跨可用區域或跨區域的持續資料保護，並以秒為單位測量復原點目標（ RPO ）。

Amazon S3 可以用作自我管理和受 AWS管資料來源的備份目的地。Amazon EBS、Amazon RDS和 Amazon DynamoDB 等 AWS 服務已內建建立備份的功能。也可以使用第三方備份軟體。

AWS 雲端 內部部署資料可以使用 AWS Storage Gateway或 備份至 AWS DataSync。Amazon S3 儲存貯體可用來在 AWS中存放此資料。Amazon S3 提供多種儲存層級，例如 Amazon S3 Glacier 或 S3 Glacier Deep Archive，以降低資料儲存成本。

您能夠從其他資源重現資料來符合資料復原需求。例如，如果主要 遺失，Amazon ElastiCache 複本節點或 Amazon RDS僅供讀取複本可用來重現資料。如果像這樣的來源可用來達成復原點目標 （ RPO ）和復原時間目標 （ RTO ），您可能不需要備份。另一個範例是，如果使用 Amazon EMR，則可能不需要備份HDFS資料存放區，只要您可以從 Amazon S3 將資料重製到 EMR AmazonAmazon S3。

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。此時間應落在工作負載RTO的 內。

實作步驟

1. 識別工作負載的所有資料來源。資料可以存儲在許多資源上，例如資料庫、磁碟區、檔案系統、日誌記錄系統和物件儲存。請參閱 資源區段，以尋找不同 AWS 服務上儲存資料的相關文件，以及這些服務提供的備份功能。
2. 根據重要性對資料來源進行分類。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，某些資料可能很關鍵，需要RPO接近零，而其他資料可能較不重要，可以容忍較高RPO和某些資料遺失。同樣地，不同的資料集也可能有不同的RTO需求。
3. 使用 AWS 或第三方服務來建立資料 的備份。 AWS Backup 是一項 受管服務，允許在 上建立各種資料來源的備份 AWS。 會AWS Elastic Disaster Recovery處理自動次秒資料複寫至 AWS 區域。大多數 AWS 服務也具有原生功能來建立備份。也 AWS Marketplace 有許多解決方案可提供這些功能。有關如何從各種 AWS 服務建立資料備份的資訊，請參閱下面列出的資源。
4. 對於未備份的資料，請建立資料複製機制。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，從備份還原所需的時間比從來源重新產生資料更長，從而導致 中的入侵RTO。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 載入資料倉儲 （ 如 Amazon Redshift ） 或 MapReduce 叢集 （ 如 AmazonEMR ） 以對資料進行分析，這可能是可以從其他來源重製的資料範例。只要這些分析的結果存放在某個位置或可重現，您就不會因為資料倉儲或 MapReduce 叢集中的失敗而遭受資料遺失。可以從來源重現的其他範例包括快取 （ 例如 Amazon ElastiCache ） 或僅供RDS讀取複本。
5. 建立備份資料的節奏。建立資料來源備份是一個定期的程序，頻率應取決於 RPO。

實作計畫的工作量：中

資源

相關的最佳實務：

REL13-BP01 定義停機時間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 Volume Gateway？
- APN 合作夥伴：可協助備份的合作夥伴
- 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
- 適用於 Redis ElastiCache 的 備份和還原
- 在 Neptune 中建立資料庫叢集快照
- 建立資料庫快照
- 建立在排程上觸發的 EventBridge 規則
- 使用 Amazon S3 進行跨區域複寫
- EFS-to-EFS AWS Backup
- 將日誌資料匯出至 Amazon S3
- 物件生命週期管理
- DynamoDB 的隨需備份與還原
- DynamoDB 的 Point-in-time 復原
- 使用 Amazon OpenSearch Service Index 快照
- 什麼是 AWS Elastic Disaster Recovery？

相關影片：

- AWS re：Invent 2021 - 備份、災難復原和勒索軟體保護搭配 AWS
- AWS Backup 示範：跨帳戶和跨區域備份
- AWS re：Invent 2019：深度挖掘 AWS Backup， 英尺。 Rackspace （STG341）

相關範例：

- Well-Architected Lab - 為 Amazon S3 實作雙向跨區域複寫 （CRR）
- Well-Architected 實驗室 - 測試資料的備份和還原

- Well-Architected 實驗室 - 針對分析工作負載，透過容錯恢復進行備份與還原
- Well-Architected 實驗室 - 災難復原 - 備份與還原

REL09-BP02 安全並加密備份

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曝險等級：高

實作指引

使用身分驗證和授權控制和偵測對備份的存取，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。這兩種方法都允許您使用 AWS Key Management Service（AWS KMS）來建立和儲存資料金鑰，或者您可以提供自己的金鑰，然後由您負責。使用 AWS KMS，您可以在誰可以和無法存取您的資料金鑰和解密的資料IAM上，使用 設定政策。

對於 Amazon RDS，如果您選擇加密資料庫，則備份也會加密。DynamoDB 備份一律會加密。使用 時 AWS Elastic Disaster Recovery，傳輸中和靜態的所有資料都會加密。透過 Elastic Disaster Recovery，靜態資料可以使用預設的 Amazon EBS加密磁碟區加密金鑰或自訂客戶管理金鑰進行加密。

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 中使用加密RDS。您可以在建立RDS執行個體時使用 設定靜態 AWS Key Management Service 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 擁有的 AWS KMS 金鑰或 AWS 受管KMS金鑰，指定儲存在帳戶中的金鑰。
- 加密儲存在 Amazon 中的資料EFS。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 中存放的金鑰在 Amazon S3 中設定靜態加密KMS，但金鑰是區域特定的。您可以在設定複寫時指定目的地金鑰。
- 選擇是否使用 Elastic Disaster Recovery 的預設或自訂 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 其他組態：使用 中存放的加密金鑰複寫使用伺服器端加密建立的物件 （ SSE ） AWS KMS
- DynamoDB 靜態加密
- 加密 Amazon RDS 資源
- 在 Amazon 中加密資料和中繼資料 EFS
- 中的備份加密 AWS
- 管理加密資料表
- Security Pillar - AWS Well-Architected Framework
- 什麼是 AWS Elastic Disaster Recovery？

相關範例：

- Well-Architected Lab - 為 Amazon S3 實作雙向跨區域複寫 （ CRR ）

REL09-BP03 自動執行資料備份

根據復原點目標 （ RPO ） 或資料集中的變更通知的定期排程，將備份設定為自動進行。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：一種自動化過程，可以按既定的節奏建立資料來源的備份。

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點：自動化備份會根據您的 定期進行備份RPO，並在未進行備份時提醒您。

未建立此最佳實務時的曝險等級：中

實作指引

AWS Backup 可用於建立各種資料來源的自動 AWS 資料備份。Amazon RDS執行個體幾乎每五分鐘可以持續備份一次，Amazon S3 物件幾乎每十五分鐘可以持續備份一次，提供 point-in-time 復原（PITR）至備份歷史記錄中特定時間點。對於其他 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 可以每小時執行自動備份一次。這些服務也提供原生備份功能。提供具有復原功能的 point-in-time自動備份 AWS 的服務包括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中的特定時間點。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 提供具有復原功能的 point-in-time持續備份。Amazon S3 版本控制一旦開啟，便會自動執行。Amazon Data Lifecycle Manager 可用來自動建立、複製和刪除 Amazon EBS快照。它也可以自動建立、複製、棄用和取消註冊 Amazon EBS後端 Amazon Machine Images（AMIs）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）到目標復原區域的連續區塊層級複寫。 Point-in-time 服務會自動建立和管理 Amazon EBS快照。

為了集中檢視備份自動化和歷史記錄， AWS Backup 提供完全受管、以政策為基礎的備份解決方案。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. 識別目前正在手動備份的資料來源。如需詳細資訊，請參閱REL09-BP01 識別和備份需要備份的所有資料，或從來源複製資料。

2. 判斷工作負載RPO的 。如需詳細資訊，請參閱REL13-BP01 定義停機時間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 是完全受管的服務，可讓您輕鬆集中和自動化跨 AWS 服務、雲端和內部部署的資料保護。使用 AWS Backup中的備份計畫建立規則，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RPO建立的 告知。如需如何使用建立自動備份的實作指南 AWS Backup，請參閱測試資料備份和還原 。原生備份功能由大多數存放資料的 AWS 服務提供。例如， RDS 可用於具有 point-in-time復原 （ ） 的自動備份PITR。
4. 對於自動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)，請考慮使用受信任的第三方解決方案來建立自動備份。或者，您可以使用 AWS CLI 或 建立自動化來執行此操作SDKs。您可以使用 AWS Lambda 函數 或 AWS Step Functions 來定義建立資料備份時涉及的邏輯，並使用 Amazon EventBridge 根據您的 以頻率叫用它RPO。

實作計畫的工作量：低

資源

相關文件：

- APN 合作夥伴：可協助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- 建立在排程上觸發的 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：深度挖掘 AWS Backup， 英尺。 Rackspace （STG341）

相關範例：

- Well-Architected 實驗室 - 測試資料的備份和還原

REL09-BP04 定期復原資料，以確認備份完整性和程序

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）。

預期結果：使用定義明確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目標（RTO）內復原。確認從備份還原會導致資源包含原始資料，而不會損毀或無法存取，且在復原點目標（ ）內遺失資料RPO。

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。
- 假設從備份還原或復原資料的時間落在工作負載RTO的 內。
- 假設備份中包含的資料落在工作負載RPO的 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。

建立此最佳實務的好處：測試備份的復原可驗證資料在需要時可以還原，而不必擔心資料可能遺失或損毀、工作負載RTO的 內可能進行還原和復原，以及工作負載的任何資料遺失都落在 RPO 內。

未建立此最佳實務時的曝險等級：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。應執行的一些常見測試是檢查所有資料是否可用、未損毀、可存取，以及是否有任何資料遺失落在工作負載RPO的 內。此類測試也可以協助判斷復原機制是否足夠快速，以容納工作負載的 RTO。

使用 AWS，您可以建立測試環境並還原備份以評估 RTO和 RPO功能，以及對資料內容和完整性執行測試。

此外，Amazon RDS和 Amazon DynamoDB 允許 point-in-time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。

如果所有資料都可用，表示 未損毀、可存取，而且任何資料遺失都屬於工作負載RPO的 。此類測試也可以協助判斷復原機制是否足夠快速，以容納工作負載的 RTO。

AWS Elastic Disaster Recovery 提供 Amazon EBS磁碟區的持續 point-in-time復原快照。當來源伺服器複寫時， point-in-time狀態會根據設定的政策隨時間而變長期。彈性災難復原可透過啟動執行個體進行測試和演練，而無需重新導向流量，從而協助您驗證這些快照的完整性。

實作步驟

1. 識別目前正在備份的資料來源，以及這些備份的儲存位置。如需實作指引，請參閱 REL09-BP01 識別和備份需要備份的所有資料，或從來源複製資料。
2. 針對每個資料來源建立資料驗證條件。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 以根據資料重要性還原資料。如需實作指引，請參閱 REL13-BP01 定義停機時間和資料遺失的復原目標。
4. 評估您的復原能力。檢閱您的備份和還原策略，以了解其是否可滿足您的 RTO 和 RPO，並視需要調整策略。使用 AWS Resilience Hub，可以執行工作負載的評估。評估會根據彈性政策評估您的應用程式組態，並報告是否可以達到您的 RTO和RPO目標。
5. 使用生產中用於資料還原的當前已建立的流程進行測試還原。這些程序取決於原始資料來源的備份方式、備份本身的格式和儲存位置，或是否已從其他源重現資料。例如，如果您使用的是諸如 AWS Backup 等受管服務，這可能就像將備份還原到新資源一樣簡單。如果使用 AWS Elastic Disaster Recovery ，則可以啟動復原演練。
6. 根據先前建立的資料驗證條件，從已還原的資源中驗證資料復原。還原和復原的資料是否包含備份時最新的記錄/項目？ 此資料是否屬於工作負載RPO的 ？
7. 測量還原和復原所需的時間，並將其與您已建立的 進行比較RTO。此程序是否屬於工作負載RTO的 ？ 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呼叫都會加上時間戳記，此資訊可在 中使用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。如果使用自動流程，則可以使用 Amazon DynamoDB 之類的服務來存放此資訊。此外，許多 AWS 服務提供事件歷史記錄，可在發生特定動作時提供時間戳記資訊。在 中 AWS Backup，備份和還原動作稱為任務 ，而這些任務包含時間戳記資訊作為其中繼資料的一部分，可用於測量還原和復原所需的時間。
8. 如果資料驗證失敗，或還原和復原所需的時間超過RTO為工作負載建立的時間，請通知利益相關者。實作自動化來執行此操作時，例如在此實驗室 中，Amazon Simple Notification Service（Amazon SNS）之類的服務可用來傳送電子郵件或SMS向利益相關者傳送推播通知。這些訊息也可以發佈至傳訊應用程式，例如 Amazon Chime、Slack 或 Microsoft Teams，或用於建立 OpsItems 使用 AWS Systems Manager 的任務 OpsCenter。
9. 將此流程自動化以定期執行。例如， 中的類似 AWS Lambda 或 狀態機器的服務 AWS Step Functions 可用於自動還原和復原程序，而 Amazon EventBridge 可用於定期調用此自動化工作流程，如以下架構圖所示。了解如何使用 自動化資料復原驗證 AWS Backup。此外，此 Well-Architected 實驗室為這裡的幾個步驟提供了一種自動化方法的實際操作體驗。

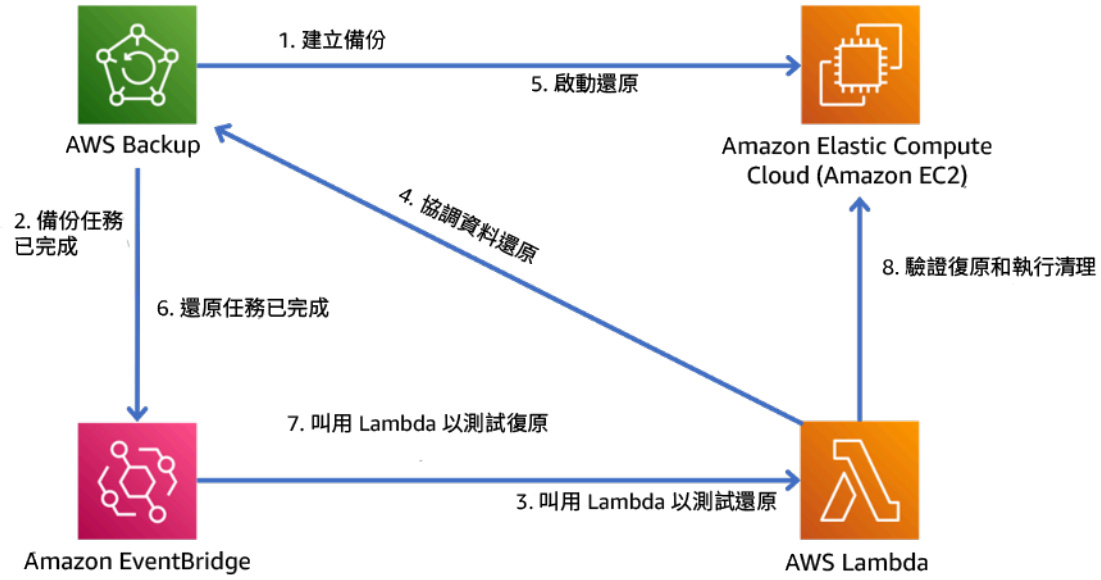


圖 9. 自動備份和還原流程

實施計畫的工作量：中到高，取決於驗證條件的複雜性。定。

資源

相關文件：

- 使用 自動化資料復原驗證 AWS Backup
- APN 合作夥伴：可協助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- 建立在排程上觸發的 EventBridge 規則
- DynamoDB 的隨需備份與還原
- 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：測試資料的備份和還原

REL 10. 如何使用故障隔離來保護工作負載？

錯誤隔離界限可將工作負載的故障影響限制在有限數量的元件內。界限外部的元件不會受到故障影響。您可以使用多個錯誤隔離界限來限制對工作負載的影響。

最佳實務

- REL10-BP01 將工作負載部署到多個位置
- REL10-BP02 為您的多位置部署選取適當的位置
- REL10-BP03 自動復原限制在單一位置的元件
- REL10-BP04 使用隔板架構來限制影響範圍

REL10-BP01 將工作負載部署到多個位置

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。這些位置可視需要多樣化。

中的服務設計基礎原則之一 AWS 是避免基礎實體基礎設施中的單一故障點。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。建置依賴備援元件的系統時，請務必確保元件獨立運作，在 的情況下 AWS 區域，則自主運作。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（AZs）

AWS 區域 由多個可得區域組成，這些區域旨在彼此獨立。每個可用區域與其他可用區域之間的實體距離較大，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。這種設計將這些系統中的故障限制為僅限於影響 AZ 的系統故障。儘管地理位置分開，但可用區域位於相同的區域，允許高輸送量、低延遲的聯網。整個 AWS 區域（所有可用區域，包含多個實體獨立資料中心）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（例如，資料庫之間）的能力。這允許您在主動/主動或主動/待命組態中使用可用區域。

可用區域是獨立的，因此當將工作負載架構為使用多個區域時，可提高工作負載的可用性。某些 AWS 服務（包括 Amazon EC2執行個體資料平面）會部署為嚴格區域服務，這些服務已與其所在的可用區域共用命運。AZs 不過，另一個 中的 Amazon EC2執行個體不會受到影響並繼續運作。同樣，如果可用區域中的故障導致 Amazon Aurora 資料庫失敗，則未受影響的可用區域中的僅供讀取複本 Aurora 執行個體可自動升級為主要執行個體。另一方面，區域 AWS 服務 (例如 Amazon DynamoDB) 則在主動/主動組態中在內部使用多個可用區域，以實現該服務的可用性設計目標，而無需設定可用區域放置。

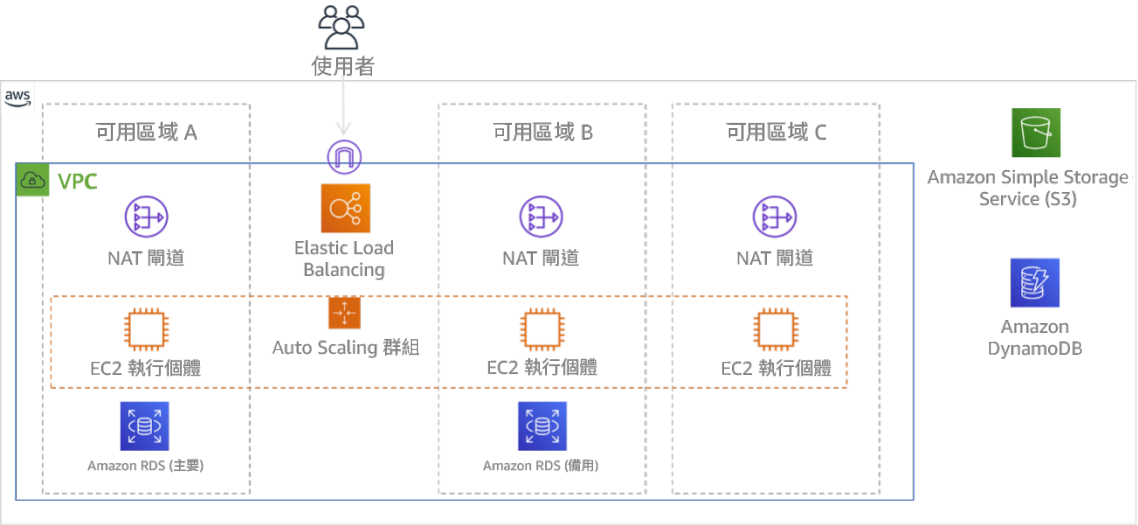


圖 9：跨三個可用區域部署的多層架構。請注意，Amazon S3 和 Amazon DynamoDB 一律自動採用異地同步備份策略。ELB也會部署到所有三個區域。

雖然 AWS 控制平面通常提供管理整個區域（多個可用區域）內資源的能力，但某些控制平面（包括 Amazon EC2和 Amazon EBS）能夠將結果篩選到單一可用區域。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 本地區域

AWS 本機區域的作用類似於其各自 AWS 區域 中的可用區域，因為它們可以被選取為區域 AWS 資源的置放位置，例如子網路和EC2執行個體。讓他們特別之處在於，它們不在關聯的 中 AWS 區域，但接近目前 AWS 區域 不存在的大型人口、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 包含全球各城市的邊緣節點。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 可讓您在這些邊緣位置建立工作負載端點，以為接近使用者 AWS 的全球網路提供入門。Amazon API Gateway 允許使用 CloudFront 分佈進行邊緣最佳化的API端點，以透過最接近的邊緣位置促進用戶端存取。

AWS 區域

AWS 區域 因此， 設計為自主，為了使用多區域方法，您將為每個區域部署專用的服務複本。

在發生一次性大規模事件時，災難復原策略通常採用多區域方法來實現復原目標。如需這些策略的詳細資訊，請參閱災難復原 (DR) 計畫。然而，在這裡，我們專注於可用性，這試圖隨著時間的推移實現平均執行時間目標。對於高可用性目標，多區域架構通常設計為主動/主動，其中每個服務副本 (在其各自的區域中) 都處於活動狀態 (服務請求)。

> **建議**
>
> 使用單一 AWS 區域內的多可用區域策略，可滿足大多數工作負載的可用性目標。只有在工作負載具有極高可用性需求或需要多區域架構的其他業務目標時，才考慮使用多區域架構。

AWS 為您提供跨區域操作服務的功能。例如，使用 Amazon Simple Storage Service （ Amazon S3 ） 複寫、Amazon RDS Read Replicas （ 包括 Aurora Read Replicas ） 和 Amazon DynamoDB Global Tables AWS 提供連續、非同步的資料複寫。透過連續複寫，您的資料版本幾乎可以在每個活動區域中立即使用。

使用 AWS CloudFormation，您可以定義您的基礎設施，並在 AWS 帳戶 和 之間持續部署 AWS 區域。透過單一操作，您可以在多個帳戶和區域之間建立、更新或刪除 AWS CloudFormation 堆疊， AWS CloudFormation StackSets 並擴展此功能。對於 Amazon EC2執行個體部署， AMI （ Amazon Machine Image ） 用於提供硬體組態和已安裝軟體等資訊。您可以實作 Amazon EC2 Image Builder 管道，建立AMIs所需的 ，並將其複製到作用中區域。這可確保這些 Golden AMIs 擁有在每個新區域中部署和擴展工作負載所需的一切。

若要路由流量，Amazon Route 53 和 AWS Global Accelerator 都允許定義政策，以決定哪些使用者前往哪個作用中區域端點。使用 Global Accelerator，您可以設定流量調節盤，以控制導向至每個應用程式端點的流量百分比。Route 53 支援此百分比方法，以及多個其他可用政策，包括地理位置鄰近政策和基於延遲的政策。Global Accelerator 會自動利用廣泛的 AWS 邊緣伺服器網路，盡快將流量加入 AWS 網路骨幹，進而降低請求延遲。

所有這些能力的運作都是為了維護每個地區的自主權。此方法的例外極少，包括提供全域邊緣交付 （ 例如 Amazon CloudFront 和 Amazon Route 53 ） 的服務，以及 AWS Identity and Access Management （ IAM ） 服務的控制平面。大多數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中執行的工作負載， 會盡可能建構混合式體驗。 AWS Direct Connect 提供來自您內部部署的專用網路連線， AWS 讓您同時在兩者中執行。

另一個選項是使用 執行內部部署的 AWS 基礎設施和服務 AWS Outposts。 AWS Outposts 是完全受管的服務，可將 AWS 基礎設施、 AWS 服務、 APIs和工具擴展到您的資料中心。中 AWS 雲端 所使用的相同硬體基礎設施會安裝在您的資料中心。 AWS Outposts 然後連接至最近的 AWS 區域。然後，您可以使用 AWS Outposts 來支援低延遲或本機資料處理需求的工作負載。

未建立此最佳實務時的曝險等級：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。這些位置可視需要多樣化。
  - 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 時VPC）
  - 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。使用多區域資料儲存，包括快取。使用 Amazon EC2 Auto Scaling 的功能、Amazon ECS任務置放、在 中執行時 AWS Lambda 的功能組態VPC，以及 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：跨可用區域分佈執行個體
      - 選擇區域與可用區域
    - 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
    - 當您將 函數設定為在 中執行時，請在多個可用區域中使用子網路VPC。
      - 設定 AWS Lambda 函數以存取 Amazon 中的資源 VPC
    - 將多個可用區域與 ElastiCache 叢集搭配使用。
      - 選擇區域與可用區域
- 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。大多數可靠性需求都可以 AWS 區域使用多可用區域策略在單一 中滿足。視需要使用多區域策略，以符合您的業務需求。
  - AWS re：Invent 2018：多區域主動應用程式的架構模式 （ARC209-R2）
    - 備份到另一個 AWS 區域 可以新增另一層保證，以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。
- AWS Outposts 評估工作負載。如果您的工作負載需要內部部署資料中心達到低延遲，或有本機資料處理要求。然後使用 在內部部署上執行 AWS 基礎設施和服務 AWS Outposts
  - 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，請參閱 AWS Local Zones 是否位於使用者附近。如果是，請用它將工作負載部署在更靠近這些使用者的位置。
  - AWS Local Zones FAQ

資源

相關文件：

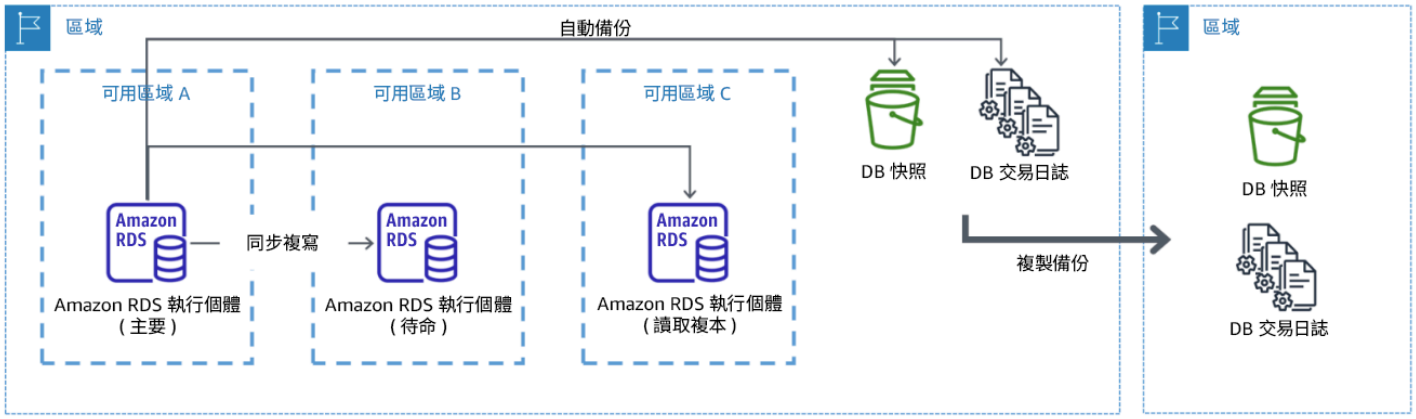
- AWS 全球基礎設施
- AWS Local Zones FAQ
- Amazon ECS任務置放策略
- 選擇區域與可用區域
- 範例：跨可用區域分佈執行個體
- 全域資料表：使用 DynamoDB 進行多區域複寫
- 使用 Amazon Aurora 全球資料庫
- 使用 AWS 服務建立多區域應用程式部落格系列
- 什麼是 AWS Outposts？

相關影片：

- AWS re：Invent 2018：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操作（NET339）

REL10-BP02 為您的多位置部署選取適當的位置

預期結果：為了實現高可用性， 一律 （如果可能） 將工作負載元件部署到多個可用區域 （AZs）。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。



備份到另一個 AWS 區域的彈性多可用區域資料庫部署

常見的反模式：

- 選擇當多可用區域架構滿足需求時設計多區域架構。
- 如果這些元件之間的彈性和多位置需求不同，則不會考慮應用程式元件之間的相依性。

建立此最佳實務的優勢：為了進行復原，應使用可建立多層防禦的方法。透過使用多個 建置高可用性架構，單層可避免較小、更常見的中斷AZs。另一防禦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。第二層涉及建構您的應用程式以跨多個 AWS 區域。

- 99.5% 的可用性與 99.99% 的可用性之間的差異在於每月 3.5 小時以上。如果工作負載位於多個中，則其預期的可用性只能達到「四九」AZs。
- 透過在多個 中執行工作負載AZs，您可以隔離電源、冷卻和聯網的故障，以及火災和洪水等大多數自然災難。
- 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。

未建立此最佳實務時的曝險等級：高

實作指引

對於基於一個可用區域中斷或部分遺失的災難事件，在單一可用區域中實作高可用性工作負載 AWS 區域 有助於緩解自然和技術災難。每個 AWS 區域 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，並相隔一段距離。不過，對於存在可能丟失多個可用區域元件 (彼此之間

距離很遠) 風險的災難事件，應該實作災難復原選項，以減輕整個區域範圍的故障。對於需要極高彈性的工作負載 (關鍵基礎設施、健康相關應用程式、金融系統基礎設施等)，可能需要採用多區域策略。

實作步驟

1. 評估您的工作負載，並判斷彈性需求是否可以透過多可用區方法（單一 AWS 區域）滿足，或是否需要多區域方法。實作多區域架構以滿足這些需求會帶來額外的複雜性，因此請仔細考量您的使用案例及其需求。使用單個 AWS 區域幾乎總是可以滿足彈性需求。判斷是否需要使用多個區域時，請考慮下列可能的需求：
   a. 災難復原（DR）：對於基於一個可用區域中斷或部分遺失的災難事件，在單一 中的多個可用區域中實作高可用性工作負載 AWS 區域 有助於緩解自然和技術災難。對於存在可能丟失多個可用區域元件 (彼此之間距離很遠) 風險的災難事件，應該在多個區域實作災難復原選項，以減輕整個區域範圍的自然災害或技術故障。
   b. 高可用性（HA）：多區域架構（AZs在每個區域中使用多個）可用來實現大於四個 9 的（> 99.99%）可用性。
   c. 堆疊在地化 ：將工作負載部署至全球受眾時，您可以在不同的 中部署在地化堆疊 AWS 區域 ，為這些區域中的受眾提供服務。本地化可以包括語言、貨幣和儲存的資料類型。
   d. 使用者鄰近性：將工作負載部署至全球受眾時，您可以透過在 AWS 區域 接近最終使用者的位置部署堆疊來降低延遲。
   e. 資料落地：某些工作負載受到資料落地要求的約束，其中某些使用者的資料必須保留在特定國家/地區的邊界內。根據所涉及的法規，您可以選擇將整個堆疊或僅將資料部署到這些邊界 AWS 區域 內的 。
2. 以下是 AWS 服務提供的一些多可用區域功能範例：
   a. 若要使用 EC2或 保護工作負載ECS，請在運算資源前部署 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 Balancer 入門
      ii. Network Load Balancers 入門
   b. 如果EC2執行個體執行不支援負載平衡的商業 off-the-shelf軟體，您可以實作多可用區域災難復原方法，以達到容錯狀態。
      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，將服務平均部署到三個 AZs，以實現可用性和成本的平衡。
      i. Amazon ECS可用性最佳實務 | 容器

d. 對於非 Aurora Amazon RDS，您可以選擇多可用區作為組態選項。主要資料庫執行個體失敗時，Amazon RDS會自動提升待命資料庫，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提升彈性。

   i. Amazon RDS Multi AZ 部署

   ii. 在不同的 中建立僅供讀取複本 AWS 區域

3. 以下是 AWS 服務提供的多區域功能的一些範例：

a. 對於由服務自動提供多可用區域可用性的 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. 對於由服務自動提供多可用區域可用性的 DynamoDB 資料表，您可輕鬆地將現有資料表轉換為全域資料表，以利用多個區域。

   i. 將您的單一區域 Amazon DynamoDB 資料表轉換為全域資料表

c. 如果您的工作負載由 Application Load Balancer 或 Network Load Balancer 預付，請使用 AWS Global Accelerator 將流量導向包含運作狀態良好的端點的多個區域，以改善應用程式的可用性。

   i. AWS Global Accelerator - AWS Global Accelerator （amazon.com） 中標準加速器的端點

d. 對於利用 的應用程式 AWS EventBridge，請考慮跨區域匯流排將事件轉送到您選擇的其他區域。

   i. 在 之間傳送和接收 Amazon EventBridge 事件 AWS 區域

e. 對於 Amazon Aurora 資料庫，請考慮橫跨多個 AWS 區域的 Aurora 全域資料庫。也可以修改現有叢集以新增區域。

   i. Amazon Aurora 全球資料庫入門

f. 如果您的工作負載包含 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. 中的多區域金鑰 AWS KMS

g. 如需其他服務 AWS 功能，請參閱此部落格系列，了解如何使用 AWS 服務建立多區域應用程式系列

實作計畫的工作量：中高

資源

相關文件：

- 使用 AWS Services 系列建立多區域應用程式
- 上的災難復原 （DR） 架構 AWS，第 IV 部分：多站台主動/主動

- AWS 全球基礎設施
- AWS Local Zones FAQ
- 上的災難復原 （DR） 架構 AWS，第 I 部分：雲端中的復原策略
- 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：多區域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：

- 上的災難復原 （DR） 架構 AWS，第 I 部分：雲端中的復原策略
- DTCC 達到的復原能力遠遠超過其可在內部部署執行的操作
- Expedia Group 使用具有專屬DNS服務的多區域、多可用區域架構，為應用程式增加彈性
- Uber：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 建置資料落地
- Intuit 跨兩個區域 TurboTax 執行

REL10-BP03 自動復原限制在單一位置的元件

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曝險等級：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 會針對相同可用區域中的指定叢集EMR啟動所有節點，因為在相同區域中執行叢集可改善工作流程的效能，因為其提供更高的資料存取率。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，除了使用多可用區域之外，您也應該以其他方式佈建備援。可以佈建多個節點。使用EMR檔案系統（EMRFS），中的資料EMR可以儲存在 Amazon S3 中，進而可以在多個可用區域或 之間複寫 AWS 區域。

同樣地，對於 Amazon Redshift，依預設，它會在 AWS 區域 您選擇的 內隨機選取的可用區域中佈建叢集。所有叢集節點將佈建在相同的區域中。

對於部署到內部部署資料中心的狀態型伺服器工作負載，您可以使用 AWS Elastic Disaster Recovery 來保護 中的工作負載 AWS。如果您已在 中託管 AWS，您可以使用 Elastic Disaster Recovery 來保護替代可用區域或區域的工作負載。Elastic Disaster Recovery 使用輕量型暫存區的持續區塊層級複寫，以提供內部部署應用程式和雲端應用程式的快速且可靠的復原。

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。如果您無法使用自動擴展，請針對EC2執行個體使用自動復原，或根據 Amazon EC2或ECS容器生命週期事件實作自我修復自動化。
   - 針對沒有單一執行個體 IP 地址、私有 IP 地址、彈性 IP 地址和執行個體中繼資料需求的執行個體和容器工作負載，請使用 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。
   - 針對需要單一EC2執行個體 ID 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 執行個體的自動復原。
     - 當偵測到執行個體失敗時，自動復原會將復原狀態警示傳送至SNS主題。
   - 使用 Amazon EC2執行個體生命週期事件或 Amazon ECS事件，在無法使用自動擴展或EC2復原的情況下，自動進行自我修復。
     - 使用事件來調用自動化，以根據您所需的過程邏輯來修復您的元件。
   - 使用 AWS Elastic Disaster Recovery 保護僅限於單一位置的有狀態工作負載。

資源

相關文件：

- Amazon ECS事件
- Amazon EC2 Auto Scaling 生命週期掛鉤
- 復原您的執行個體。
- 服務自動擴展

- 什麼是 Amazon EC2 Auto Scaling
- AWS Elastic Disaster Recovery

REL10-BP04 使用隔板架構來限制影響範圍

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：小組型架構使用工作負載的多個獨立執行個體，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：使用小組型架構時，小組本身包含許多常見的故障類型，從而提供額外的故障隔離。這些故障界限可針對難以控制的失敗類型提供復原能力，例如失敗的程式碼部署或已損毀或調用特定失敗模式 (也稱為毒藥請求) 的請求。

未建立此最佳實務時的曝險等級：高

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實現故障隔離。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。界限外部的元件不會受到故障影響。您可以使用多個錯誤隔離界限來限制對工作負載的影響。在 上 AWS，客戶可以使用多個可用區域和區域來提供故障隔離，但故障隔離的概念也可以擴展到工作負載的架構。

整體工作負載是依分割區索引鍵的分割區小組。此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的跨小組互動進行細分。分割區金鑰的範例包括客戶 ID、資源 ID 或任何其他參數，在大多數API通話中都能輕鬆存取。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。

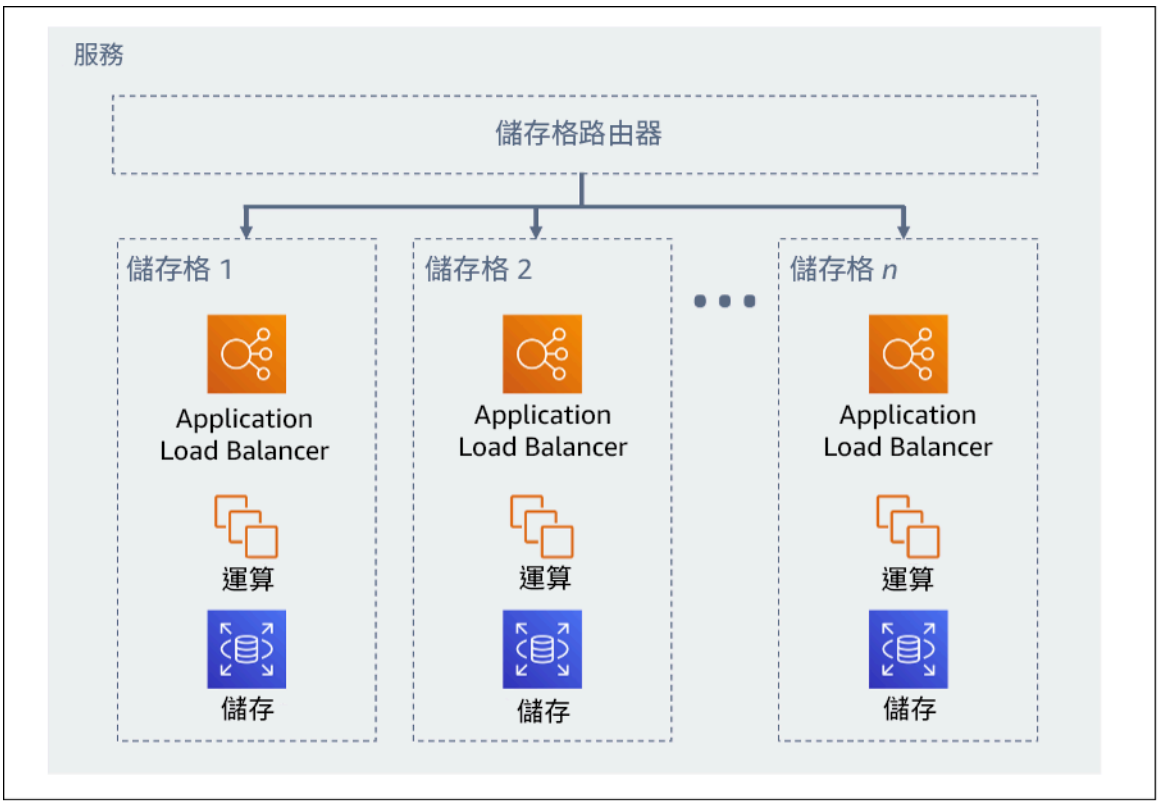


圖 11：小組型架構

實作步驟

設計小組型架構時，要考慮數個設計考量：

1. 分割區索引鍵：選擇分割區索引鍵時，應特別考慮。
   - 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。例如 `customer ID` 或 `resource 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. 持久性小組映射：上游服務應該僅在其資源的生命週期內與單個小組進行互動。
   - 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：路由器層是儲存格之間的共用元件，因此無法遵循與儲存格相同的區隔策略。

- 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。正如 Colm MacCárthaigh 在可靠性、持續工作以及一杯好咖啡中所述，簡單的設計和持續的工作模式會產生可靠的系統並降低抗脆弱性。

4. 儲存格大小：儲存格應具有最大的大小，且不允許超過它。
   - 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 Load 測試您的工作負載
   - 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：應利用多層彈性來防範不同的故障網域。
   - 對於彈性，您應該使用建置防禦層的方法。透過使用多個 建置高可用性架構，單層可避免較小、更常見的中斷AZs。另一防禦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。第二層涉及建構您的應用程式以跨多個 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的位置。
6. 程式碼部署：交錯的程式碼部署策略應優先於同時將程式碼變更部署到所有單元格。
   - 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，請參閱安全且無需人為干預的自動化部署。

資源

相關的最佳實務：

- REL07-BP04 Load 測試您的工作負載
- REL10-BP02 為您的多位置部署選取適當的位置

相關文件：

- 可靠性、持續工作以及一杯好咖啡
- AWS 和室化
- 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：閉環和開場思維：如何控制大大小小的系統
- AWS re：Invent 2018：如何將故障的 Blast Radius AWS 降至最低（ARC338）
- Shuffle-sharding： AWS re：Invent 2019：介紹 Amazon Builders 的程式庫（DOP328）
- AWS ANZ Summit 2021 - 一切都失敗，一直：為恢復能力而設計

相關範例：

- Well-Architected 實驗室：透過隨機切換分區來隔離故障

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）的工作負載必須進行彈性架構。

最佳實務

- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP02 容錯移轉至健全的資源
- REL11-BP03 自動化所有層的復原
- REL11-BP04 在復原期間依賴資料平面而非控制平面
- REL11-BP05 使用靜態穩定性來防止雙模式行為
- REL11-BP06 當事件影響可用性時傳送通知
- REL11-BP07 建構您的產品以滿足可用性目標和運作時間服務等級協議（SLAs）

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（KPIs）。

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，因此衡量此值的關鍵效能指標（KPIs）需要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，但在此閾值下無法提供足夠的回應時間。
- 指標的收集頻率不足以達到復原時間目標（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）是從發生故障到開始修復操作之間的時間量。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《進階多可用區域彈性模式》白皮書中的 Gray failures。

實作步驟

- 您的監控間隔取決於復原必須多快完成。復原時間取決於復原所需的時間，因此您必須考慮此時間和復原時間目標（ ）來決定收集頻率RTO。
- 設定元件和受管服務的詳細監控。
  - 判斷是否需要詳細的EC2執行個體監控和 Auto Scaling。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。
  - 判斷是否需要增強對 的監控RDS。增強型監控會在RDS執行個體上使用代理程式，以取得不同程序或執行緒的有用資訊。
  - 判斷 Lambda 、APIGateway 、Amazon EKS、Amazon ECS和所有類型負載平衡器 的重要無伺服器元件的監控需求。
  - 判斷 Amazon S3、Amazon 、Amazon FSx EFS和 Amazon EBS儲存元件的監控需求。
- 建立自訂指標以測量業務金鑰效能指標（KPIs）。工作負載會實作關鍵業務函數，這些函數應用作KPIs協助識別間接問題發生的時間。

- 以使用者 Canary 監控使用者的故障體驗。可執行和模擬客戶行為的綜合交易測試 (也稱為 Canary 測試，但請別與 Canary 部署混淆)，是最重要的測試程序之一。針對來自不同遠端位置的工作負載端點持續執行這些測試。
- 建立追蹤使用者體驗的自訂指標。如果您可以檢測客戶的體驗，則可以判斷消費者體驗何時變差。
- 設定警示以偵測工作負載的任何部分何時未正常運作，並指示何時自動擴展資源。警示可以視覺化顯示在儀表板上，透過 Amazon SNS或電子郵件傳送警示，並使用 Auto Scaling 將工作負載資源向上或向下擴展。
- 建立儀表板以視覺化指標。儀表板可以讓您以視覺化方式查看趨勢、極端值和其他潛在問題的指標，或指出您可能想要調查的問題。
- 為您的服務建立分散式追蹤監控。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。
- 在個別區域和帳戶中建立監控系統（使用或 CloudWatch X-Ray） 儀表板和資料收集。
- 建立 Amazon Health Aware 監控的整合，以允許監控可能降級之 AWS 資源的可見性。對於業務必要工作負載，此解決方案可讓您存取 AWS 服務的主動和即時警示。

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
- Enhanced Monitoring (增強型監控)
- 使用 Amazon 監控 Auto Scaling 群組和執行個體 CloudWatch
- 發佈自訂指標
- 使用 Amazon CloudWatch 警示
- 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：Level 300：實作運作狀態檢查和管理相依性以提升可靠性
- 一個可觀測性研討會：探索 X-Ray

相關工具：

- CloudWatch
- CloudWatch X-Ray

REL11-BP02 容錯移轉至健全的資源

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），請確定您已備妥系統，以容錯移轉至未受損位置中的健全資源。

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，以將從故障和對資料的影響中復原的時間降到最低。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。我們也最佳化功能 replace-and-restart，以快速從中斷中復原。

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 原生受管服務都是原生多個可用區域（例如 Lambda 或 API Gateway）。其他服務 AWS（例如 EC2和 EKS）需要特定的最佳實務設計，以支援跨 的資源或資料儲存的容錯移轉AZs。

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。
- RTO 和 RPO 未建立。
- 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 ，可協助分散資源和可用區域的負載。因此，將流量轉移到保持運作狀態良好的資源，可以減輕個別資源 （ 例如EC2執行個體 ） 的故障或可用區域受損。

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至多個 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、Amazon Application Recovery Controller （ ARC ）、Amazon CloudFront和 AWS Global Accelerator 可協助跨 路由流量 AWS 區域。

AWS 服務，例如 Amazon S3、Lambda、APIGateway、Amazon SQS、Amazon SNS、Amazon SES、Amazon 、Amazon Pinpoint 、Amazon ECR AWS Certificate Manager EventBridge、 或 Amazon DynamoDB ，都會由 自動部署到多個可用區域 AWS。發生故障時， AWS 這些服務會自動將流量路由至運作狀態良好的位置。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，多可用區是組態選項。如果啟動容錯移轉， AWS 可以將流量導向至運作狀態良好的執行個體。此容錯移轉動作可能由客戶採取， AWS 或依客戶要求採取

對於 Amazon EC2執行個體、Amazon Redshift、Amazon ECS任務或 Amazon EKS Pod，您可以選擇要部署的可用區域。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，重新路由可以利用 Amazon Route 53、Amazon Application Recovery Controller AWS Global Accelerator、Route 53 Private DNS for VPCs或 CloudFront 提供一種方法來定義網際網路網域和指派路由政策，包括運作狀態檢查，以將流量路由至運作狀態良好的區域。 AWS Global Accelerator 提供靜態 IP 地址，作為應用程式固定進入點，然後使用 AWS 全域網路而不是網際網路路由至 AWS 區域 您選擇的端點，以提高效能和可靠性。

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RPO每個元件建立容錯移轉設計會議RTO和 。
- 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。
- 決定哪些容錯移轉計劃由 自動執行 AWS，哪些計劃可以由 DevOps 程序自動執行，哪些計劃可能是手動的。記錄並測量每個服務的 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 ），請確定您已備妥系統，無法容錯移轉至未受損位置中的健全資源。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- DR 的計劃
- REL10 - 使用故障隔離來保護工作負載

相關文件：

- 設定RTO和RPO目標
- 使用 Route 53 加權路由進行容錯移轉
- 使用 Amazon Application Recovery Controller 進行災難復原
- EC2 使用自動擴展
- EC2 部署 - 多可用區域
- ECS 部署 - 多可用區
- 使用 Amazon Application Recovery Controller 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 EFS和 容錯移轉的跨區域複寫
- EFS 跨區域複寫和容錯移轉
- 聯網容錯移轉
- S3 端點容錯移轉使用 MRAP
- 為 S3 建立跨區域複寫
- 跨區域容錯移轉和容錯性容錯回復的指南 AWS
- 使用多區域 Global Accelerator 進行容錯移轉
- 使用 進行容錯移轉 DRS
- 使用 Amazon Route 53 建立災難復原機制

相關範例：

- 上的災難復原 AWS
- 上的彈性災難復原 AWS

REL11-BP03 自動化所有層的復原

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，復原設計和自動修復程序可從現有最佳實務中提取。

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：執行自動化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 服務的設計應包含最小和最大複本或狀態集，以及最小叢集和節點群組大小調整。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散到一個或多個可用區域中的多個目標和虛擬設備（AZs）。

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。範例服務為 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 自動執行 AWS Systems Manager ，以快速修復問題。

Amazon EventBridge 可用來監控和篩選事件，例如 CloudWatch 警示或其他 服務的狀態 AWS 變更。根據事件資訊，它可以叫用 AWS Lambda、Systems Manager Automation 或其他目標，以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執行個體運作狀態。如果執行個體處於執行以外的任何狀態，或者系統狀態受損，Amazon EC2 Auto Scaling 會將執行個體視為運作狀態不佳，並啟動替換執行個體。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，請使用負載平衡並選擇適當的負載平衡器類型。
- 請考慮為 Amazon 進行復原RDS。對於待命執行個體，請設定待命執行個體的自動容錯移轉。對於 Amazon RDS Read Replica，需要自動化工作流程才能將僅供讀取複本設為主要。
- 在已部署應用程式且無法部署在多個位置的EC2執行個體上實作自動復原，並可在失敗時容忍重新啟動。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。會保留執行個體中繼資料和相關聯的 IP 地址，以及 Amazon Elastic File System 或 File Systems for Lustre 和 Windows EBS的磁碟區和掛載點。使用 AWS OpsWorks，您可以在層層級設定EC2執行個體的自動修復。
- 當您無法使用自動擴展或自動復原，或自動復原失敗時，則使用 AWS Step Functions 和 AWS Lambda 實作自動復原。當您無法使用自動擴展，且無法使用自動復原或自動復原失敗時，您可以使用 AWS Step Functions 和 自動復原 AWS Lambda。
- Amazon EventBridge 可用來監控和篩選事件，例如CloudWatch警示或其他 AWS 服務的狀態變更。根據事件資訊，它接著可以調用 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 Automatic Recovery
- 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出現故障的執行個體
- 什麼是 AWS Step Functions？
- 什麼是 AWS Lambda？
- 什麼是 Amazon EventBridge？
- 使用 Amazon CloudWatch 警示
- 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 服務
- Amazon RDS 自動容錯移轉

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 在復原期間依賴資料平面而非控制平面

控制平面提供APIs用於建立、讀取和描述、更新、刪除和列出（CRUDL）資源的管理，而資料平面則處理 day-to-day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：

- 取決於變更DNS記錄以重新路由流量。
- 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。
- 依賴廣泛的多服務多API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風險暴露等級：中。對於某些類型的服務降級，則會影響控制平面。廣泛使用控制平面進行修復的相依性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 Application Recovery Controller 來轉移DNS流量。這些功能會持續監控應用程式從故障中復原的能力，並可讓您控制跨多個 AWS 區域、可用區域和內部部署的應用程式復原。

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 資料平面會回答DNS查詢，並執行和評估運作狀態檢查。它們是全域分佈的，專為 100% 可用性服務層級協議（SLA）而設計。

您建立、更新和刪除 Route 53 資源的 Route 53 管理和APIs主控台，在控制平面上執行，其設計旨在優先考慮管理 時所需的強大一致性和耐用性DNS。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)。雖然這兩個系統都建置得非常可靠，但 中不包含控制平面SLA。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。

將運算基礎設施設計為靜態穩定，以避免在事件期間使用控制平面。例如，如果您使用 Amazon EC2 執行個體，請避免手動佈建新執行個體，或指示 Auto Scaling 群組在回應中新增執行個體。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，請在整體 end-to-end系統上設定限流，以安全地限制到達有限資源集的總流量。

對於 Amazon DynamoDB 、Amazon API Gateway、負載平衡器和無 AWS Lambda 伺服器等服務，使用這些服務會利用資料平面。不過，建立新的函數、負載平衡器、API閘道或 DynamoDB 資料表是控制平面動作，應該在降級之前完成，作為事件的準備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、控制平面以及如何 AWS 建置服務以滿足高可用性目標的詳細資訊，請參閱使用可用區域 的靜態穩定性。

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） 至預先擴展的 Amazon EC2 資源 （資料平面）
- Amazon EC2執行個體擴展 （控制平面） 到 AWS Lambda 擴展 （資料平面）
- 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 Amazon Application Recovery Controller （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EKS的 Amazon EC2 Auto Scaling 或 Amazon 與次要區域中的 Amazon EC2 Auto Scaling 或 Amazon 相比EKS，並將流量路由至次要區域（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：可協助自動化容錯能力的合作夥伴
- AWS Marketplace：可用於容錯的產品
- Amazon 建置者資料中心：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（分割至控制平面和資料平面）
- AWS Elemental MediaStore 資料平面
- 使用 Amazon Application Recovery Controller 建置高彈性的應用程式，第 1 部分：單一區域堆疊
- 使用 Amazon Application Recovery Controller 建置高彈性的應用程式，第 2 部分：多區域堆疊
- 使用 Amazon Route 53 建立災難復原機制
- 什麼是 Amazon Application Recovery Controller
- 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台工作負載

相關範例：

- Amazon Application Recovery Controller 簡介
- Amazon 建置者資料中心：控管較小服務，避免分散式系統過載

- 使用 Amazon Application Recovery Controller 建置高彈性的應用程式，第 1 部分：單一區域堆疊
- 使用 Amazon Application Recovery Controller 建置高彈性的應用程式，第 2 部分：多區域堆疊
- 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式行為

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。

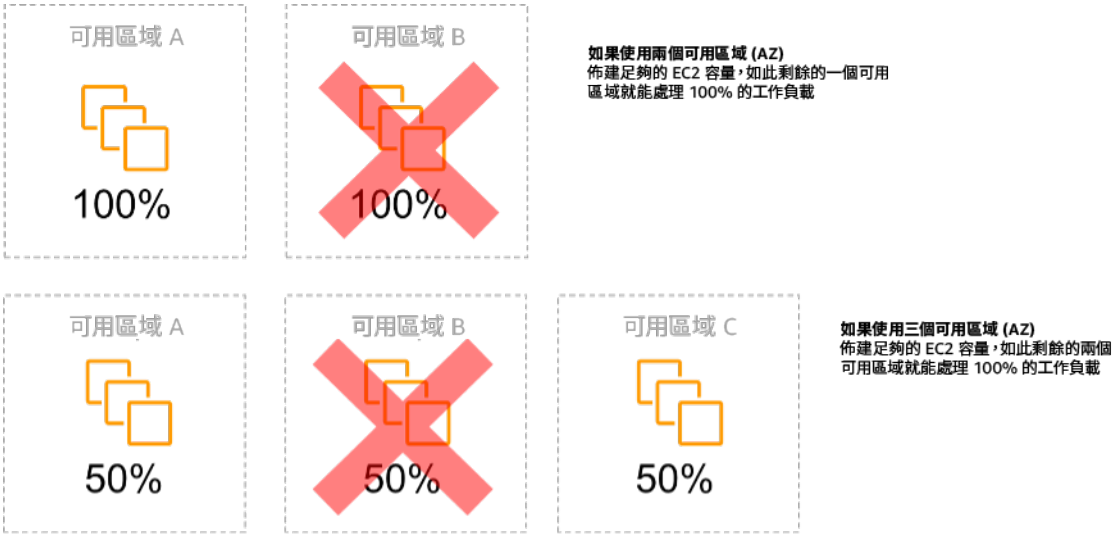
建立此最佳實務的優勢：使用靜態穩定設計執行的工作負載，能夠在正常和故障事件發生時產生可預測的結果。

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。雙模式行為的範例是當穩定的 Amazon EC2設計在每個可用區域中佈建足夠的執行個體，以便在移除一個 AZ 時處理工作負載負載。Elastic Load Balancing 或 Amazon Route 53 運作

狀態會進行檢查，將負載從受損的執行個體中移出。流量轉移後，請使用 AWS Auto Scaling 以非同步方式取代失敗區域中的執行個體，並在運作狀態良好的區域中啟動執行個體。運算部署的靜態穩定性（例如EC2執行個體或容器）會產生最高的可靠性。



跨可用區域的EC2執行個體靜態穩定性

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。靜態穩定性架構適用於各種架構，包括跨可用區域分佈的運算執行個體、資料庫僅供讀取複本設計、Kubernetes（AmazonEKS）叢集設計和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。呼叫失敗時，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
- 儲存體：Amazon S3 （單區域）、Amazon EFS（安裝）、Amazon FSx（安裝）
- 負載平衡器：在某些設計下

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 MultiRegion Routing
  - AWS Global Accelerator
  - Amazon Application Recovery Controller
- 設定資料庫讀取複本以考慮單一主要執行個體或讀取複本的遺失情況。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。如果使用 Amazon S3 One Zone-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。
- Load balancers 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，靜態穩定的設計可能是在更複雜的設計AZs中將工作負載分散到多個。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 在復原期間依賴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 建置者資料中心：使用可用區域實現靜態穩定性

- 故障隔離界限
- 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 MultiRegion Routing
- AWS Global Accelerator
- Amazon Application Recovery Controller
- 單區域 Amazon S3
- 跨區域負載平衡

相關影片：

- AWS： AWS re：Invent 2019 中的靜態穩定性：介紹 Amazon Builders' Library （DOP328）

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。

預期結果：違反閾值時，警示會立即傳送至營運團隊，例如錯誤率、延遲或其他關鍵金鑰效能指標（KPI）指標，以便儘快解決這些問題，並避免或將使用者影響降至最低。

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。
- 在設計監控和警示時，不考慮微小故障。

- 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點：復原通知可讓營運和業務團隊了解服務降級，以便他們能夠立即回應，將平均偵測時間（MTTD）和平均修復時間（）降至最低MTTR。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風險暴露等級：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 `IN ALARM` 或 `OK`) 以及觸發原因詳情。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，應在超過閾值時將警示傳送至適當的團隊。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。複合式警示會使用數個 KPI監控警示，根據操作業務邏輯建立警示。 CloudWatch 警示可以設定為傳送電子郵件，或使用 Amazon SNS整合或 Amazon 將事件記錄到第三方事件追蹤系統中 EventBridge。

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。
- 基礎設施警示會指出何時擴展資源。警示可以視覺化顯示在儀表板上，透過 Amazon SNS或電子郵件傳送警示，並使用 Auto Scaling 來擴展工作負載資源。
- 可建立簡單的靜態指示，以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。您可以直接叫用 Amazon SNSAPI傳送通知，並連結任何自動化以進行修復或通訊。
- 整合 Amazon Health Aware 監控，以監控可能降級 AWS 的資源可見性。對於業務必要工作負載，此解決方案可讓您存取 AWS 服務的主動和即時警示。

資源

相關 Well-Architected 的最佳實務：

- 可用性定義

相關文件：

- 根據靜態閾值建立 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 警示
- Amazon Health Aware （AHA）
- 設定 CloudWatch 複合警示
- re：Invent 2022 可 AWS 觀測性的新功能

相關工具：

- CloudWatch
- CloudWatch X-Ray

REL11-BP07 建構您的產品以滿足可用性目標和運作時間服務等級協議 （SLAs）

建構您的產品，以符合可用性目標和運作時間服務層級協議 （SLAs）。如果您發佈或私下同意可用性目標或運作時間 SLAs，請確認您的架構和操作程序是設計來支援這些目標。

預期結果：每個應用程式都有一個已定義的可用性和SLA效能指標目標，這些指標可以監控和維護，以滿足業務結果。

常見的反模式：

- 設計和部署工作負載，而不設定任何 SLAs。
- SLA 指標設定過高，沒有理由或業務需求。
- 在SLAs未考量相依性及其基礎 的情況下進行設定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，可協助您達成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。

未建立此最佳實務時的曝險等級：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。每個工作負載都應考慮其相依性及其 SLAs。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。
- 透過SLAs減少平均偵測時間（MTTD）和平均維修時間（），可觀測性和儀表對於實現至關重要MTTR。
- 較不頻繁的故障（較長的）MTBF、較短的故障偵測時間（較短的 MTTD）和較短的維修時間（較短的 MTTR）是用來改善分散式系統可用性的三個因素。
- 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？

- 與利益相關者合作時，根據工作負載架構定義SLA指標。考慮工作負載使用SLAs的所有相依性。
- 設定SLA目標後，請最佳化架構以符合 SLA。
- 設定將符合 的設計後SLA，請實作操作變更、程序自動化和 Runbook，這些也會專注於減少 MTTD 和 MTTR。
- 部署後，請在 上監控和報告 SLA。

資源

相關的最佳實務：

- REL03-BP01 選擇如何分割工作負載
- REL10-BP01 將工作負載部署到多個位置
- REL11-BP01 監控工作負載的所有元件以偵測故障
- REL11-BP03 自動化所有層的復原
- REL12-BP05 使用混亂工程測試彈性
- REL13-BP01 定義停機時間和資料遺失的復原目標
- 了解工作負載運作狀態

相關文件：

- 備援的可用性
- 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 服務層級協議（SLAs）
- 上的儲存格型架構指南 AWS
- 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 使用教戰手冊調查失敗
- REL12-BP02 執行事後分析
- REL12-BP03 測試功能需求
- REL12-BP04 測試擴展和效能需求
- REL12-BP05 使用混亂工程測試彈性
- REL12-BP06 定期執行遊戲日

REL12-BP01 使用教戰手冊調查失敗

藉由在程序手冊中記錄調查程序，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。在識別或呈報問題之前，任何程序步驟的結果都會用來決定要採取的後續步驟。

程序手冊是您必須進行的主動規劃，以便能夠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。

常見的反模式：

- 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流程得到一致遵循。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。

未建立此最佳實務時的曝險等級：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調用或執行，或者可能提示執行程序手冊，以回應已識別的事件。
    - 使用 AWS Systems Manager 自動化您的操作手冊
    - AWS Systems Manager 執行命令
    - 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 自動化
- AWS Systems Manager 執行命令
- 使用 AWS Systems Manager 自動化您的操作手冊
- 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。一種機制是修正錯誤（COE）程序。COE 此程序可協助您的團隊識別、了解和解決事件的根本原因，同時建立機制和防護機制，以限制再次發生相同事件的機率。

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，以讓其他實作了相同成因的工作負載減輕風險，並讓工作負載能夠在事件發生前實作減輕措施或自動復原。

未建立此最佳實務時的曝險等級：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。

程序的基石COE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。使用諸如五個為什麼等技巧深入研究潛在問題。

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。這些業務決策可以讓您了解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：

- 為什麼您應該制定錯誤修正（COE）

相關影片：

- Amazon 成功失敗的方法
- AWS re：Invent 2021 - Amazon Builders 程式庫：Amazon 的卓越營運

REL12-BP03 測試功能需求

使用可驗證所需功能的技術，例如單元測試和整合測試。

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，開發人員會將變更遞交至來源儲存庫，其中 CodePipeline 會自動偵測變更。系統會建置這些變更，並執行測試。在測試完成後，該建置程式碼會部署至開發中伺服器以進行測試。從預備伺服器 CodePipeline 執行更多測試，例如整合或載入測試。成功完成這些測試後， CodePipeline 會將已測試和核准的程式碼部署到生產執行個體。

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 可讓您建立 Canary 來監控端點和 APIs。

未建立此最佳實務時的曝險等級：高

實作指引

- 測試功能需求。這包括驗證所需功能的單位測試和整合測試。
  - CodePipeline 搭配 使用 AWS CodeBuild 來測試程式碼和執行組建
  - AWS CodePipeline 使用 新增對 裝置和自訂整合測試的支援 AWS CodeBuild
  - 持續交付和持續整合
  - 使用 Canary（Amazon CloudWatch Synthetics）
  - 軟體測試自動化

資源

相關文件：

- APN 合作夥伴：可協助實作持續整合管道的合作夥伴
- AWS CodePipeline 使用 新增對 裝置和自訂整合測試的支援 AWS CodeBuild
- AWS Marketplace：可用於持續整合的產品
- 持續交付和持續整合
- 軟體測試自動化
- CodePipeline 搭配 使用 AWS CodeBuild 來測試程式碼和執行組建
- 使用 Canary （ Amazon CloudWatch Synthetics ）

REL12-BP04 測試擴展和效能需求

使用諸如負載測試等技術來驗證工作負載符合擴展和效能需求。

在雲端，您可以隨需建立工作負載的生產規模測試環境。如果在已縮減規模的基礎設施上執行這些測試，則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。

未建立此最佳實務時的曝險等級：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。
  - 上的分散式負載測試 AWS：模擬數千個連線使用者
  - 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：

- 上的分散式負載測試 AWS：模擬數千個連線使用者
- Apache JMeter

REL12-BP05 使用混亂工程測試彈性

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，還可以藉由在故障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。如此一來，混亂實驗應作為系統開發生命週期 （ SDLC ） 的一部分執行，並作為 CI/CD 管道的一部分執行。

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，嘗試遺失 Amazon EC2 執行個體或容錯移轉主要 Amazon RDS 資料庫執行個體，並確認您的工作負載未受影響（或僅受影響極小）。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。

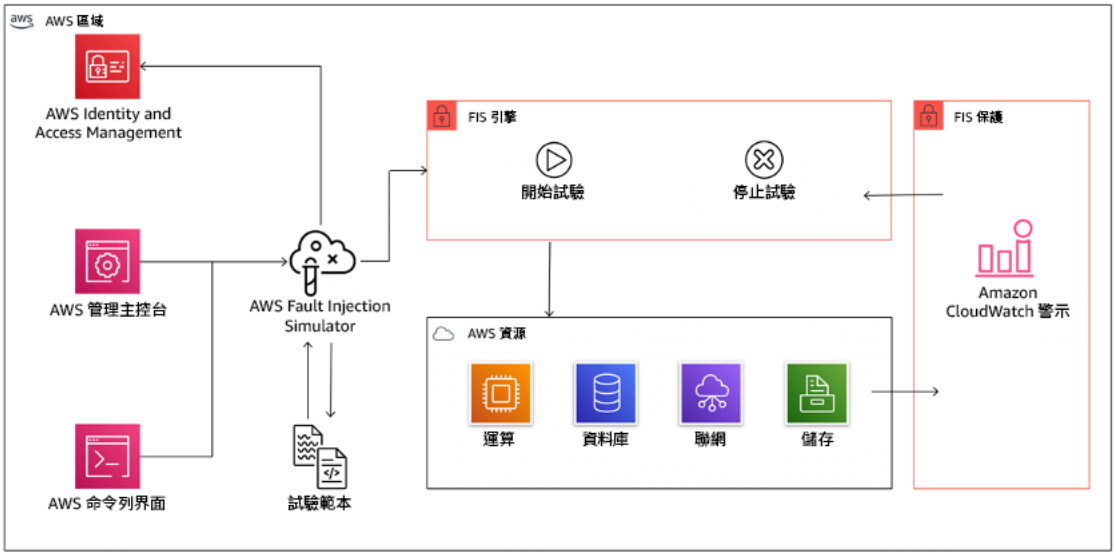
對於應用程式層級故障（例如當機），您可以從記憶體和CPU耗盡等壓力因素開始。

為了驗證由於間歇性網路中斷導致的外部相依性的備用或容錯移轉機制，您的元件應在指定的時間內 (可延續數秒到數小時) 阻止對第三方供應商的存取來模擬這類事件。

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)。這些故障的實驗，包括延遲、訊息中斷和DNS故障等聯網效果，可能包括無法解析名稱、聯絡DNS服務或建立與相依服務的連線。

混沌工程工具：

AWS Fault Injection Service（AWS FIS）是一種完全受管的服務，用於執行故障注入實驗，可作為 CD 管道的一部分或管道外部使用。 AWS FIS 是混亂工程遊戲日使用的好選擇。它支援同時引入不同類型資源的故障，包括 Amazon EC2、Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic Kubernetes Service（Amazon EKS）和 Amazon RDS。這些故障包括終止資源、強制容錯移轉、強調CPU或記憶體、限流、延遲和封包遺失。由於其與 Amazon CloudWatch Alarms 整合，因此您可以將停止條件設定為防護機制，以便在實驗造成非預期的影響時復原實驗。



AWS Fault Injection Service 與 AWS 資源整合，可讓您執行工作負載的故障注入實驗。

故障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，以及諸如 Gemlin 之類的商業選項。若要擴展可在 上注入的故障範圍 AWS， AWS

FIS 與 Chaos Mesh 和 Litmus Chaos 整合，可讓您在多個工具之間協調故障注入工作流程。例如，您可以使用 CPU Chaos Mesh 或 Litmus 故障在 Pod 上執行壓力測試，同時使用 AWS FIS 故障動作終止隨機選取的叢集節點百分比。

實作步驟

1. 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) 會根據關鍵相依性、過去的事件、已知問題和合規性要求來考量風險。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。如需有關建立此類清單的詳細資訊，請參閱 Operational Readiness Review 白皮書，指導您如何建立程序以防止先前事件再次發生。失敗模式和效果分析 （FMEA） 程序為您提供架構，用於執行失敗的元件層級分析，以及它們如何影響您的工作負載。FMEA 由 Adrian Cockcroft 在失敗模式和持續恢復 中更詳細地概述。

2. 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故障注入試驗。

3. 對於您所執行的每個試驗，均應依循下圖中的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。

a. 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，付款系統的穩定狀態可以定義為處理 300 次TPS，成功率為 99%，往返時間為 500 毫秒。

b. 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**
>
> If *specific fault* 發生，*workload name* 工作負載將 *describe mitigating controls* 以維護 *business or technical metric impact*.

例如：

- 如果 Amazon 節點群組中的 20% EKS節點遭到刪除，則 Transaction Create API將繼續為 100 毫秒 （ 穩定狀態 ） 以下請求的第 99 百分位數提供服務。Amazon EKS節點將在五分鐘內復原，且 Pod 將在實驗開始後八分鐘內排定並處理流量。提醒將在三分鐘內引發。
- 如果發生單一 Amazon EC2執行個體故障，訂單系統的 Elastic Load Balancing 運作狀態檢查將導致 Elastic Load Balancing 僅在 Amazon EC2 Auto Scaling 取代故障執行個體時，將請求傳送至其餘運作狀態良好的執行個體，並維持伺服器端 （ 5xx ） 錯誤 （ 穩定狀態 ） 增加低於 0.01%。
- 如果主要 Amazon RDS 資料庫執行個體失敗，供應鏈資料收集工作負載將容錯移轉並連線至待命 Amazon RDS 資料庫執行個體，以維持資料庫讀取或寫入錯誤 （ 穩定狀態 ） 不到 1 分鐘。

c. 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。已知的未知是您知道但不完全理解的事情，未知的未知是您不知道也不完全理解的事情。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 提供許多預先定義的錯誤模擬，稱為動作。您也可以 AWS FIS 使用AWS Systems Manager 文件 定義在 中執行的自訂動作。

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。從已建立的服務開始 AWS FIS ，這種服務可讓您執行具有明確定義範圍和安全機制的實驗，如果實驗引入非預期的湍流，則會復原實驗。若要了解使用 的更多各種實驗

AWS FIS，另請參閱具有 Chaos Engineering lab 的彈性和建構良好的應用程式。此外，AWS Resilience Hub 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。

在可行的情況下，實驗應該使用 Canary 部署在實際負載下在生產環境中執行，這可加速控制和實驗系統部署。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。綜合監測 (也稱為使用者 Canary)，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。

執行故障注入試驗時，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。無論是自動化（例如使用 AWS FIS）還是手動，這些貼文動作都應該是說明如何偵測和處理失敗的教戰手冊的一部分。

d. 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，在工作負載用戶端的任何 APIs 或URIs直接存取上包含合成監視器（也稱為使用者 Canary）。

e. 改善工作負載設計的彈性。

如果未維持穩定狀態，請採用 AWS Well-Architected 可靠性支柱的最佳實務，調查如何改進工作負載設計來緩解故障。可以在 AWS 建置者資料中心中找到其他指引和資源，其中包含有關如何改善運作狀態檢查或在應用程式碼中透過輪詢進行重試等文章。

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。

4. 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。若要了解如何執行此操作，請參閱此部落格，了解如何使用 執行 AWS FIS 實驗 AWS CodePipeline。這個關於 CI/CD 管道中的經常性AWS FIS 實驗的實驗室可讓您親手操作。

故障注入試驗也是演練日的一部分 (請參閱 REL12-BP06 定期執行遊戲日)。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。

5. 擷取並儲存試驗結果。

故障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。結果範例可能包括儀表板螢幕擷取畫面、指標資料庫中的CSV傾印，或實驗中事件和觀察的手動輸入記錄。使用 AWS FIS試驗日誌記錄可以是此資料擷取的一部分。

資源

相關的最佳實務：

- REL08-BP03 整合彈性測試作為部署的一部分
- 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 Canary 部署

相關影片：

- AWS re：Invent 2020：使用混亂工程測試彈性（ARC316）
- AWS re：Invent 2019：改善混亂工程的復原能力（DOP309-R1）
- AWS re：Invent 2019：在無伺服器世界中執行混亂工程（CMY301）

相關範例：

- 建構良好的實驗室：300 級：測試 Amazon EC2、Amazon RDS和 Amazon S3 的彈性
- AWS 實驗室的混沌工程
- 混沌工程實驗室的彈性和 Well-Architected 應用程式
- 「無伺服器混沌」實驗室
- 使用 AWS Resilience Hub 實驗室測量和改善您的應用程式彈性

相關工具：

- AWS Fault Injection Service
- AWS Marketplace：Gremlin 混沌工程平台
- Chaos Toolkit
- Chaos Mesh
- Litmus

REL12-BP06 定期執行遊戲日

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。

演練日會模擬失敗或事件，以測試系統、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。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，以便您的團隊建立應對的肌肉記憶。

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。執行手冊已經到位。模擬事件會以規定的方式在生產系統中執行，包括可能發生的故障事件，並會評估影響。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)。由於演練日經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。

常見的反模式：

- 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。

未建立此最佳實務時的曝險等級：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，然後注入故障。
  - 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。如果您執行程序手冊，請識別應使用的執行手冊，或建立新的執行手冊。

資源

相關影片：

- AWS re：Invent 2019：改善混亂工程的復原能力（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 定義停機時間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 在 DR 站台或區域管理組態偏離
- REL13-BP05 自動化復原

REL13-BP01 定義停機時間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）是服務中斷和服務還原之間的可接受延遲上限。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以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

RTO 和 RPO值是為您的工作負載選擇適當的災難復原（DR）策略時的重要考量。這些目標由業務決定，然後由技術團隊用於選擇和實作 DR 策略。

預期成果：

每個工作負載都有指派的 RTO和 RPO，根據業務影響定義。工作負載會指派給預先定義的 層，定義服務可用性和可接受的資料遺失，並具有關聯的 RTO和 RPO。如果這種分層不可行，那麼可以為每個工作負載進行定制分配，以便稍後建立分層。RTO 和 RPO 用作為工作負載選擇災難復原策略實作的主要考量之一。挑選 DR 策略的其他考量是成本限制、工作負載相依性和操作需求。

對於 RTO，請根據中斷持續時間了解影響。它是線性的，還是有非線性影響？(例如，四個小時後，將生產線關閉，直到下一個班次開始)。

災難復原矩陣 (如下所示) 可協助您了解工作負載關鍵性與復原目標之間的關聯性。(請注意，X 軸和 Y 軸的實際值應根據您的組織需求來自訂)。

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：災難復原矩陣

常見的反模式：

- 沒有定義的復原目標。
- 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。
- 不了解停機和資料遺失的影響。
- 選取不切實際的復原目標，例如零復原時間和零資料損失，這對於您的工作負載組態而言可能無法實現。
- 選擇比實際業務目標更嚴格的復原目標。這會強制進行比工作負載所需更昂貴和更複雜的 DR 實作。
- 選取與相依工作負載不相容的復原目標。

- 您的復原目標不會考慮法規遵循要求。
- RTO 並為工作負載RPO定義，但從未測試過。

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。

未建立此最佳實務時的曝險等級：高

實作指引

對於指定的工作負載，您必須了解停機和資料遺失對業務造成的影響。停機時間或資料遺失越大，影響通常也會越大，但是這種增長形式可能會因工作負載類型而有所不同。例如，您可能會容忍影響很小的一小時停機，但是在此之後，影響很快就會上升。對業務的影響表現為多種形式，包括貨幣成本 (例如收入損失)、客戶信任 (以及對聲譽的影響)、營運問題 (例如缺少薪資單或生產力下降) 以及監管風險。使用下列步驟來了解這些影響，並RPO為您的工作負載設定 RTO和 。

實作步驟

1. 確定此工作負載的業務利益相關者，並與他們互動以實作這些步驟。工作負載的復原目標是一項業務決策。然後，技術團隊與業務利益相關者合作，使用這些目標來選擇 DR 策略。

   > **Note**
   >
   > 對於步驟 2 和 3，您可以使用 the section called “實作工作表”。

2. 透過回答以下問題來收集必要的資訊以做出決定。
3. 對於組織中的工作負載影響，您是否有關鍵性類別或層級？
   a. 如果有，請將此工作負載指派到某個類別
   b. 如果沒有，請建立這些類別。建立五個或更少的類別，並調整每個類別的復原時間點目標範圍。範例類別包括：嚴重、高、中、低。若要了解工作負載如何映射至類別，請考慮工作負載是關鍵任務、重要業務還是非業務驅動。
   c. 根據類別設定工作負載 RTO和 RPO 。始終選擇比輸入此步驟計算的原始值更嚴格的類別 （較低RTO和 RPO ）。如果這會導致值發生不合適的大幅變更，請考慮建立新類別。
4. 根據這些答案，將 RTO和 RPO值指派給工作負載。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. 將此工作負載的災難復原計劃 （DRP） 記錄在工作負載團隊和利益相關者可存取的位置，這是組織業務持續性計劃 （BCP） 的一部分
   a. 記錄 RTO和 RPO，以及用於判斷這些值的資訊。包括用於評估工作負載對業務的影響的策略

   b. 除了記錄其他指標，RTORPO您是否正在追蹤或計劃追蹤災難復原目標
   c. 當您建立 DR 策略和執行手冊的詳細資訊時，將這些詳細資訊新增至此計畫。
6. 透過查閱矩陣中的工作負載關鍵性，如圖 15，就可以開始建立為組織定義的預先定義服務層級。
7. 根據 實作 DR 策略 （或 DR 策略的概念驗證） 之後the section called “REL13-BP02 使用定義的復原策略來滿足復原目標”，請測試此策略以判斷實際工作負載 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的，則可以與您的業務利益相關者合作以調整這些目標，或者對 DR 策略進行變更以實現目標。

主要問題

1. 在對業務造成嚴重影響之前，工作負載可以停止的最長時間是多少
   a. 如果工作負載中斷，請確定每分鐘對業務的貨幣成本 (直接財務影響)。
   b. 考慮到影響並不總是線性的。影響起初可能會受到限制，然後在臨界點後迅速增加。
2. 在對業務造成嚴重影響之前，可能遺失的最大資料量是多少
   a. 針對最關鍵的資料存放區考慮此值。確定其他資料存放區的各自關鍵性。
   b. 如果遺失工作負載資料，是否可以重新建立？ 如果此操作比備份和還原更容易操作，則RPO根據用於重新建立工作負載資料的來源資料的重要性進行選擇。
3. 此工作負載所依賴的工作負載 (下游) 或者依賴此工作負載的工作負載 (上游) 的復原目標和可用性期望為何？
   a. 選擇允許此工作負載以符合上游相依性需求的復原目標
   b. 根據下游相依性的復原能力，選擇可實現的復原目標。可以排除非關鍵的下游相依性 (您可以「解決」的相依性)。或者，您也可以在必要時使用重要的下游相依性來改善其復原能力。

其他問題

考慮以下問題，及其如何套用於此工作負載：

4. 您是否有不同的 RTO ，RPO取決於中斷類型 （區域與 AZ 等）？
5. 當您的 RTO/RPO 可能變更時，是否有特定的時間 （季節性、銷售事件、產品推出）？ 如果是這樣，不同的測量和時間邊界是什麼？
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，對聲譽有什麼影響？
8. 如果工作負載中斷，可能會產生哪些其他營運影響？ 例如，如果電子郵件系統無法使用，或薪資系統無法提交交易，對員工生產力的影響。

9. 工作負載如何RTO與業務單位和組織 DR 策略RPO保持一致？

10提供服務是否有內部合約義務？ 未滿足這些要求是否會受到懲罰？

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行步驟 2 和 3。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

實作計畫的工作量：低

資源

相關的最佳實務：

- the section called “REL09-BP04 定期復原資料，以確認備份完整性和程序”
- 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
- 上工作負載的災難復原 AWS：雲端中的復原 （AWS 白皮書）
- 使用 Resilience Hub AWS 管理復原政策
- APN 合作夥伴：可協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：多區域主動主動應用程式的架構模式 （ARC209-R2）
- 上工作負載的災難復原 AWS

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，都有已定義並實作的 DR 策略，可讓工作負載實現災難復原目標。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。
- 沒有災難復原的計畫。
- 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援您工作負載的能力。最常見的復原目標為 RTO和 RPO，如 中所述REL13-BP01 定義停機時間和資料遺失的復原目標。

單一 中跨多個可用區域的 DR 策略 （AZs） AWS 區域，可以針對火災、洪水和重大停電等災難事件提供緩解。如果必須針對不太可能發生的事件實作保護，以防止工作負載在指定的 中執行 AWS 區域，您可以使用使用多個區域的 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。它們會依成本和複雜性的增加順序列出，以及 RTO和 的減少順序RPO。復原區域是指 AWS 區域 以外的 ，而非用於工作負載的主要區域。

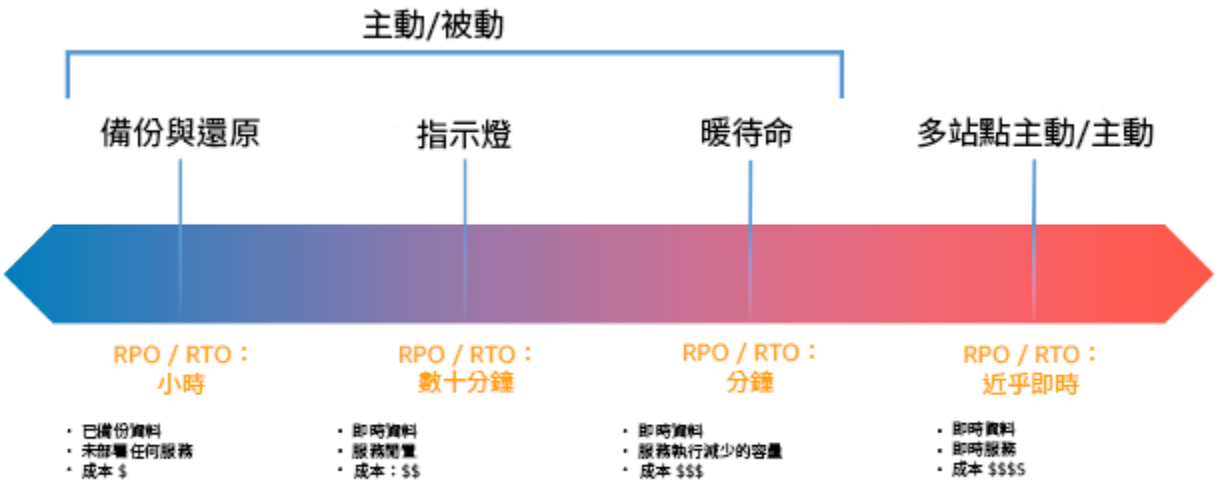


圖 17：災難復原 (DR) 策略

- 備份和還原 （RPO 小時以RTO內，24 小時以內）：將資料和應用程式備份到復原區域。使用自動或連續備份將允許時間點復原 （PITR），在某些情況下，可能低至 RPO 5 分鐘。發生災難時，您將部署基礎設施 （使用基礎設施做為程式碼來減少 RTO）、部署程式碼，以及還原備份資料以從復原區域中的災難中復原。
- 指示燈 （RPO以分鐘為單位，RTO以十分鐘為單位）：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以秒為單位，RTO以分鐘為單位）：維持縮減但功能完整的工作負載版本，一律在復原區域中執行。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。擴展程度越高，暖待命就越低，RTO控制平面依賴性也就越低。當完全擴展時，這稱為熱待命。
- 多區域（多站台）作用中（RPO 接近零，RTO可能為零）：您的工作負載會部署到多個 ，並主動提供來自 的流量 AWS 區域。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，可保護您免受某些類型的災難，但除非您的解決方案也包含 point-in-time復原選項，否則無法保護您免受資料損毀或損毀。

> **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) 基礎設施並向上擴展，而暖待命只需要您向上擴展 (一切都已部署並執行中)。根據您的 RTO和 RPO需求，在這些選項之間進行選擇。
>
> 當成本受到關注，且您希望達成暖備援策略中定義的類似RPO和RTO目標時，您可以考慮採用先導燈方法RPO並提供改進和RTO目標的雲端原生解決方案 AWS Elastic Disaster Recovery，例如 。

實作步驟

1. 決定將滿足此工作負載復原需求的 DR 策略。

   選擇 DR 策略是減少停機時間和資料遺失（RTO 和 RPO）以及實作策略的成本和複雜性之間的權衡。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，企業已確定其允許的上限RTO，以及其服務還原策略上可花費的額度限制。基於業務目標，DR 策略指示燈或暖待機將同時符合 RTO和 成本標準。

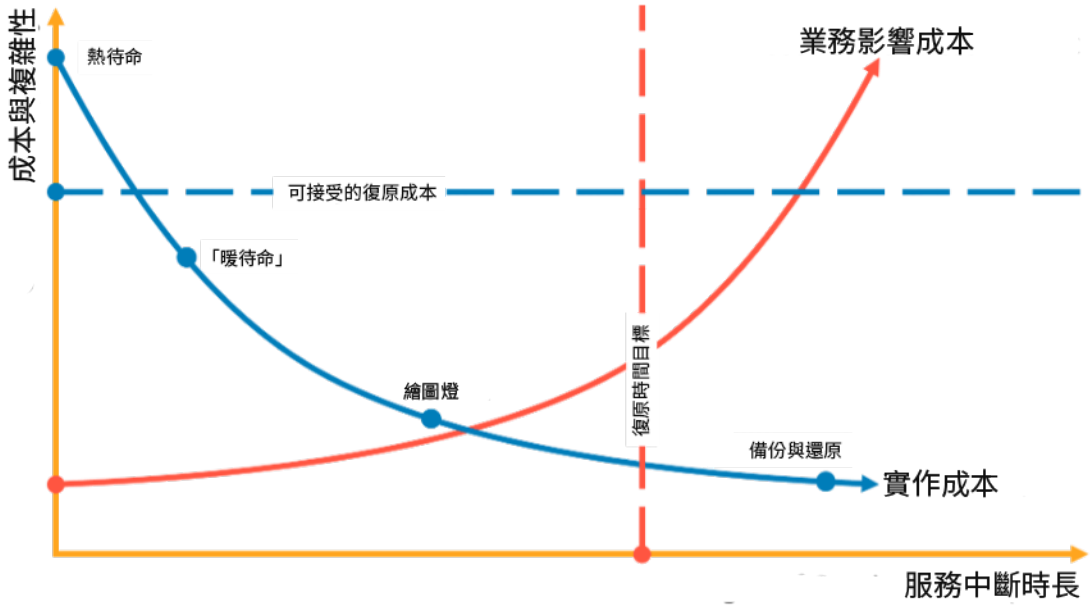


圖 18：根據 RTO和 成本選擇 DR 策略

若要進一步了解，請參閱 Business Continuity Plan （ 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 做為主要和復原站台來解釋策略。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。

備份和還原

備份和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，進而提高 RTO和 RPO。始終備份您的資料，並將其複製到另一個網站 （ 例如另一個 ） 是很好的做法 AWS 區域。

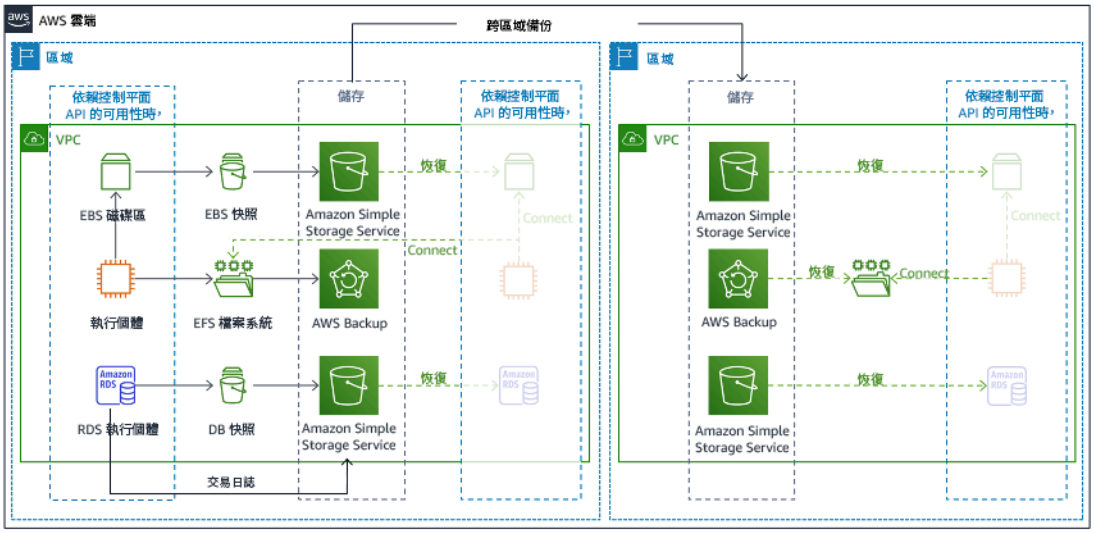


圖 19：備份和還原架構

如需此策略的詳細資訊，請參閱 上的災難復原 （DR） 架構 AWS，第二部分：使用快速復原進行備份和還原。

指示燈

使用指示燈方法，可以將資料從主要區域複製到復原區域。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。

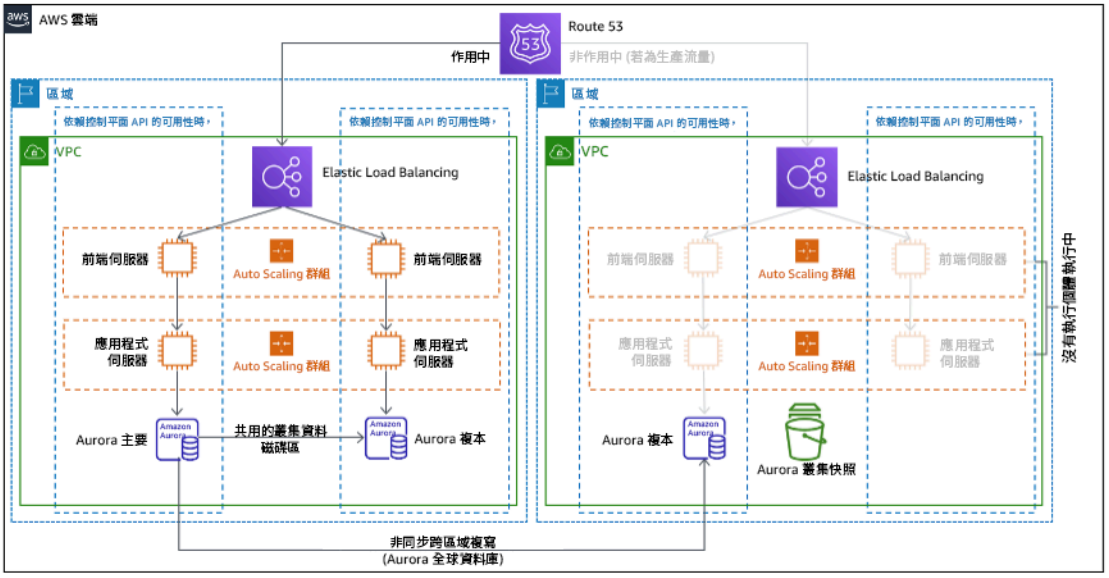


圖 20：指示燈架構

如需此策略的詳細資訊，請參閱 上的災難復原 （DR） 架構 AWS，第 部分III：Pilot Light 和 Warm Standby。

暖待命

暖待命方法包括確保在另一個區域中有規模縮減但功能完整的生產環境副本。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。如果復原區域已部署全部容量，則稱為熱待命。

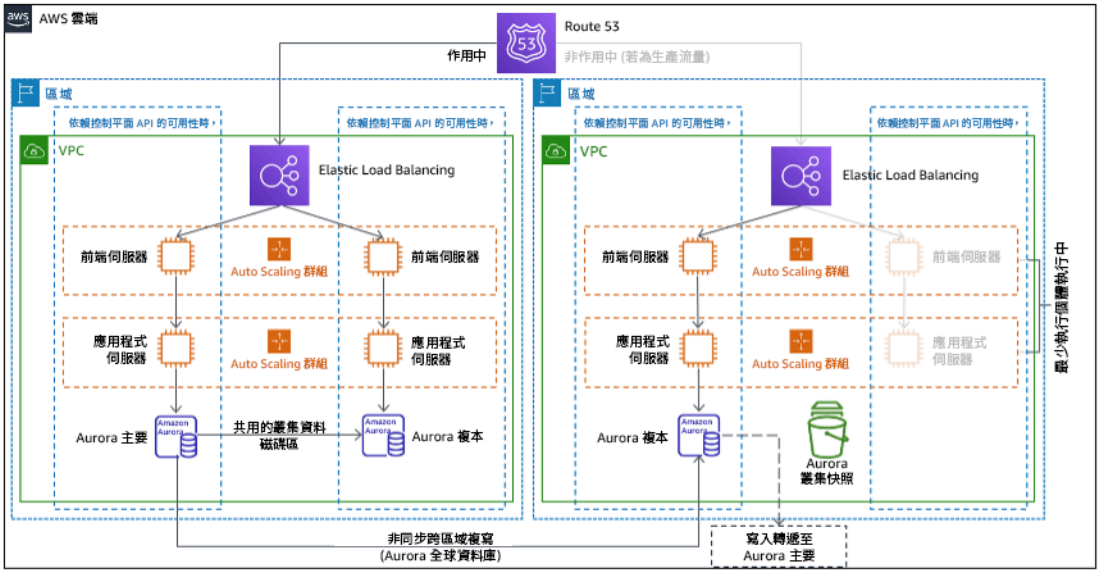


圖 21：暖待命架構

使用暖待命或指示燈需要縱向擴展復原區域中的資源。若要驗證容量是否在需要時可用，請考慮使用 為EC2執行個體保留容量。如果使用 AWS Lambda，則佈建的並行可以提供執行期環境，以便他們準備好立即回應函數的叫用。

如需此策略的詳細資訊，請參閱 上的災難復原 （DR） 架構 AWS，第 部分III：Pilot Light 和 Warm Standby。

多站點主動/主動

作為多站台主動/主動策略的一部分，您可以在多個區域同時執行工作負載。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。客戶可以出於 DR 以外的原因選擇此策略。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，如果工作負載在部署 AWS 區域 到其中的其中一個 中無法支援，則該區域會疏散，而其餘區域則用於維護可用性。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。

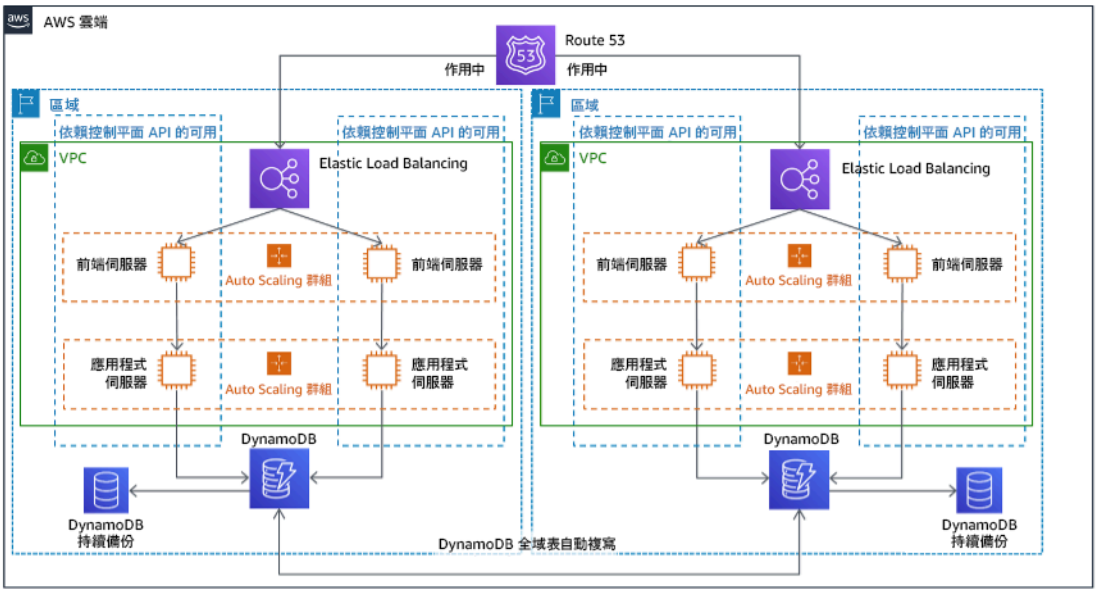


圖 22：多站點主動/主動架構

如需此策略的詳細資訊，請參閱 上的災難復原 （DR） 架構 AWS，第四部分：多站台主動/主動 。

AWS Elastic Disaster Recovery

如果您正在考慮災難復原的指示燈或熱待命策略， AWS Elastic Disaster Recovery 可以提供改善效益的替代方法。Elastic Disaster Recovery 可以提供類似暖待機的 RPO和 RTO 目標，但可維持低成本的試驗燈方法。Elastic Disaster Recovery 會使用持續資料保護，從主要區域將資料複寫至復原區域，以秒為單位RPO的測量結果，以及RTO以分鐘為單位的測量結果。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。

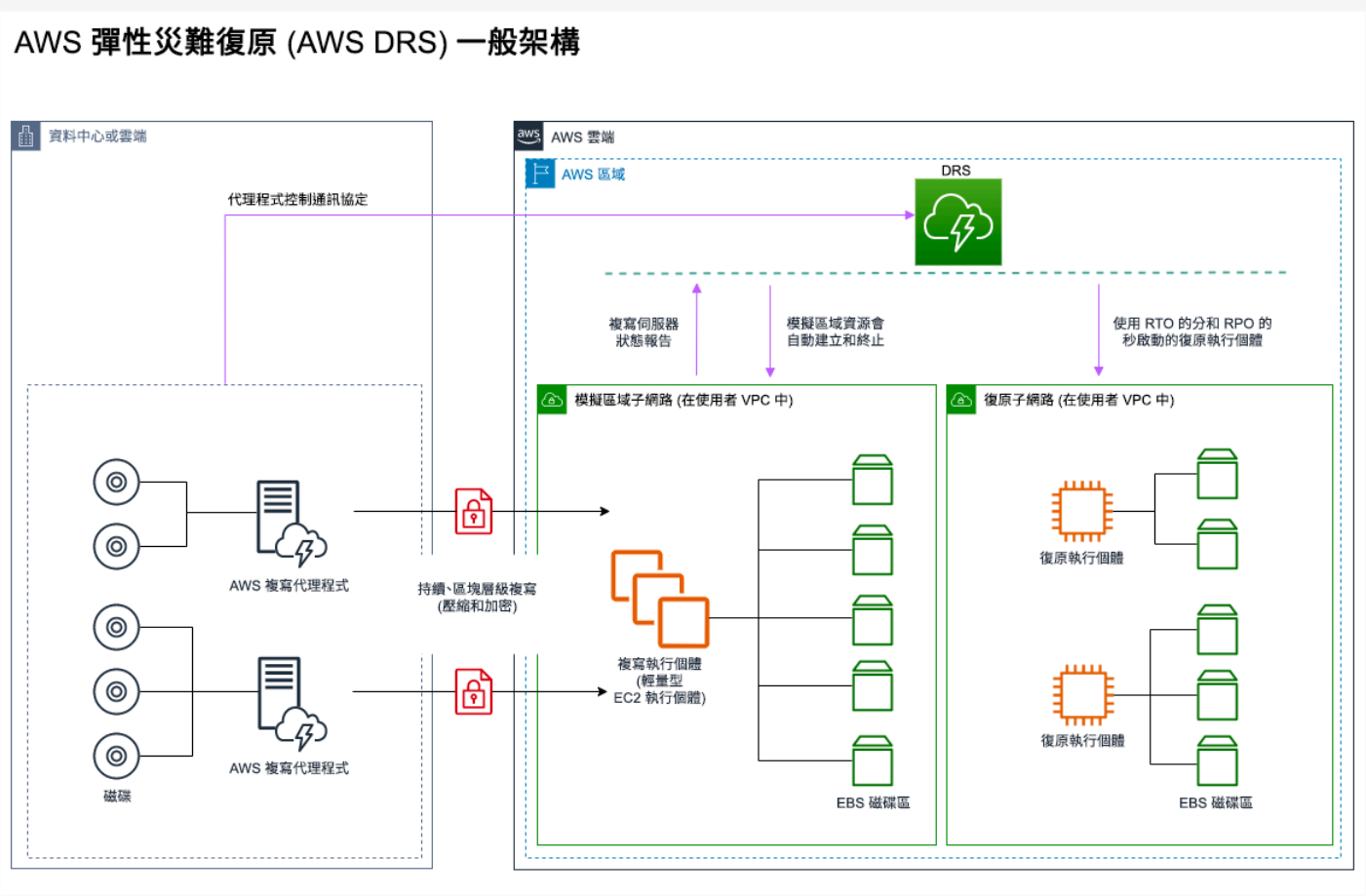


圖 23： AWS Elastic Disaster Recovery 架構

保護資料的其他實務

使用所有策略時，您還必須緩解資料災難。持續資料複寫可保護您免受某些類型的災難，但除非您的策略也包含儲存資料版本控制或復原選項 point-in-time，否則可能無法保護您免受資料損毀或損毀。除了複本之外，您還必須備份復原站台中的複寫資料，才能建立 point-in-time備份。

在單一 中使用多個可用區域 （ AZs ） AWS 區域

在AZs單一區域中使用多個 時，您的 DR 實作會使用上述策略的多個元素。首先，您必須使用多個建立高可用性 （ HA ） 架構，AZs如圖 23 所示。此架構使用多站台主動/主動方法，因為 Amazon EC2執行個體和 Elastic Load Balancer 具有部署在多個 中的資源AZs，可主動處理請求。架構也會示範熱待命，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為主要執行個體。

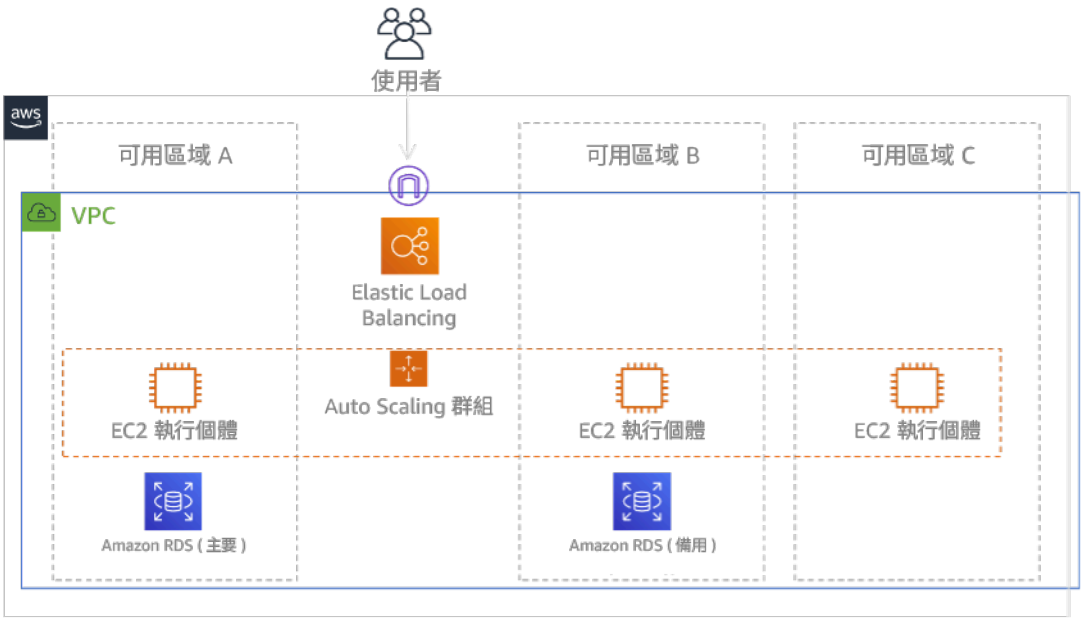


圖 24：多可用區域架構

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。這對限制為 Amazon EBS磁碟區或 Amazon Redshift 叢集 等單一區域的資料尤其重要。如果 AZ 失敗，您需要將此資料還原至另一個 AZ。在可能的情況下，您也應該將資料備份複製到另一個備份， AWS 區域 作為額外的保護層。

在部落格文章中，使用 Amazon Application Recovery Controller 建置高彈性的應用程式，第 1 部分：單一區域堆疊 中，說明了單一區域、多可用區域 DR 的較不常見替代方法。在這裡，策略是 AZs盡可能在 之間維持盡可能多的隔離，例如區域的運作方式。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。如果這適用於目前只有一個 的區域工作負載 AWS 區域，則多區域將無法滿足您的業務需求。異地同步備份策略提供良好的保護，可防範大部分災難。

3. 在容錯移轉之前 (在正常操作期間)，評估工作負載的資源，以及其在復原區域中的組態。

   對於基礎設施 AWS 和資源，請使用基礎設施作為程式碼，例如 Hashicorp Terraform AWS CloudFormation 等第三方工具。若要使用單一操作跨多個帳戶和區域部署，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 來控制部署的堆疊是作用中還是待命。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 策略都需要在 內備份資料來源 AWS 區域，然後將這些備份複製到復原區域。 AWS Backup提供集中式檢視，您可以在其中設定、排程和監控這些資源的備份。對於 Pilot Light、暖待命和多站台作用中/作用中，您也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。

若要進一步了解 AWS 服務如何在 區域間運作，請參閱此部落格系列，了解如何使用 AWS 服務建立多區域應用程式。

4. 確定並實作如何讓復原區域在需要時 (在災難事件發生時) 做好容錯移轉的準備。

   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 Pilot Light 和 Warm Standby 策略，您的復原動作將需要部署缺少的資源，例如圖 20 中的EC2執行個體，以及任何其他缺少的資源。

   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磁碟EBS區、資料庫執行個體和 DynamoDB RDS 資料表等資料建立資源。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。如需詳細資訊，請參閱REL09-BP01 識別和備份需要備份的所有資料，或從來源複製資料。除了所需的Amazon Virtual Private Cloud （ AmazonVPC ）、子網路和安全群組之外，重建基礎設施還包括建立EC2執行個體等資源。您可以將大部分還原程序自動化。若要了解如何操作，請參閱此部落格文章。

5. 確定並實作如何在需要時 (在災難事件發生時) 重新路由流量以進行容錯移轉。

   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，需要考慮幾個流量管理選項。一種選擇是使用 Amazon Route 53。使用 Amazon Route 53，您可以將一個或多個 IP 端點 AWS 區域 與 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 Application Recovery Controller ，它提供高可用性的資料平面API，將流量重新路由至復原區域。實作容錯移轉時，使用資料平面操作並避免控制平面操作，如 REL11-BP04 在復原期間依賴資料平面而非控制平面 中所述。

   若要進一步了解此選項和其他選項，請參閱《災難復原白皮書》中的此章節。

6. 設計一個工作負載故障恢復計畫。

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為即時資料庫，且具有 up-to-date資料。然後，目標是從復原區域重新同步到主要區域，確保其為 up-to-date。

有些 AWS 服務會自動執行此操作。如果使用 Amazon DynamoDB 全域表，即使主要區域中的資料表變得無法使用，當它重新上線時，DynamoDB 會繼續傳播任何擱置的寫入。如果使用 Amazon Aurora 全球資料庫並使用受管計劃容錯移轉，則維護 Aurora 全球資料庫的現有複寫拓撲。因此，主要區域中先前的讀取/寫入執行個體將成為複本，並從復原區域中接收更新。

如果這不是自動的，您將需要在主要區域中重建資料庫，做為復原區域中資料庫的複本。在許多情況下，這將涉及刪除舊的主要資料庫並建立新的複本。

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，該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 Performing a failback。

實作計畫的工作量：高

資源

相關的最佳實務：

- the section called “REL09-BP01 識別和備份需要備份的所有資料，或從來源複製資料”
- the section called “REL11-BP04 在復原期間依賴資料平面而非控制平面”
- the section called “REL13-BP01 定義停機時間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
- 上工作負載的災難復原 AWS：雲端中的復原（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容錯移轉
- S3：跨區域複寫
- 什麼是 AWS Backup？
- 什麼是 Amazon Application Recovery Controller？
- AWS 彈性災難復原
- HashiCorp Terraform：開始使用 - AWS
- APN 合作夥伴：可協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：

- 上工作負載的災難復原 AWS
- AWS re：Invent 2018：多區域主動應用程式的架構模式（ARC209-R2）
- AWS 彈性災難復原入門 | Amazon Web Services

相關範例：

- Well-Architected 實驗室 - 災難復原 - 說明 DR 策略的系列研討會

REL13-BP03 測試災難復原實作以驗證實作

定期測試復原站台的容錯移轉，以確認其運作正常且RPO符合 RTO和。

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畫，可驗證該計畫能在需要時運作，也能讓您的團隊知道如何執行策略。

未建立此最佳實務時的曝險等級：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. 對於EC2以 Amazon 為基礎的工作負載，請使用 AWS Elastic Disaster Recovery實作並啟動 DR 策略的演練執行個體。 AWS Elastic Disaster Recovery 提供有效率地執行演練的能力，協助您為容錯移轉事件做好準備。您也可以使用 Elastic Disaster Recover 頻繁啟動您的執行個體進行測試和演練，而不需要重新導向流量。

資源

相關文件：

- APN 合作夥伴：可協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
- 在 上災難復原工作負載 AWS：雲端復原 （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：多區域主動應用程式的架構模式
- AWS re：Invent 2019： Backup-and-restore 和災難復原解決方案搭配 AWS

相關範例：

- Well-Architected 實驗室 - 彈性測試

REL13-BP04 在 DR 站台或區域管理組態偏離

確保 DR 站點或區域的基礎設施、資料和組態符合需要。例如，檢查 AMIs和服務配額是否為最新版本。

AWS Config 會持續監控和記錄您的 AWS 資源組態。它可以偵測偏離並叫用 AWS Systems Manager Automation 來修正偏離並發出警示。 AWS CloudFormation 還可以另外偵測已部署堆疊中的偏離。

常見的反模式：

- 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。
- 未考量主要位置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。

未建立此最佳實務時的曝險等級：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。
- 允許 AWS Config 追蹤潛在的偏離位置。使用 AWS Config 規則來建立強制執行災難復原策略的系統，並在偵測到偏離時產生警示。
  - 透過 修復不合規 AWS 資源 AWS Config 規則
  - AWS Systems Manager 自動化
- 使用 AWS CloudFormation 部署您的基礎設施。 AWS CloudFormation 可以偵測範本指定內容 CloudFormation與實際部署內容之間的偏離。

- AWS CloudFormation：偵測整個 CloudFormation 堆疊上的漂移

資源

相關文件：

- APN 合作夥伴：可協助災難復原的合作夥伴
- AWS 架構部落格：災難復原系列
- AWS CloudFormation：偵測整個 CloudFormation堆疊上的漂移
- AWS Marketplace：可用於災難復原的產品
- AWS Systems Manager 自動化
- 在 上災難復原工作負載 AWS：雲端復原 （AWS 白皮書）
- 如何在 AWS上實作基礎設施組態管理解決方案？
- 透過 修復不合規 AWS 資源 AWS Config 規則

相關影片：

- AWS re：Invent 2018：多區域主動應用程式的架構模式 （ARC209-R2）

REL13-BP05 自動化復原

使用 AWS 或第三方工具將系統復原自動化，並將流量路由至 DR 網站或 區域。

根據設定的運作狀態檢查，Elastic Load Balancing 和 等 AWS 服務 AWS Auto Scaling可以將負載分散至運作狀態良好的可用區域，而 Amazon Route 53 和 AWS Global Accelerator 等服務可以將負載路由至運作狀態良好的 AWS 區域。Amazon Application Recovery Controller 可協助您使用就緒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故障中復原的能力，因此您可以控制跨多個 AWS 區域、可用區域和內部部署的應用程式復原。

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 Elastic Disaster Recovery 可讓組織在AWS中設定自動化災難復原策略。彈性災難復原也支援 AWS中的跨區域和跨可用區域災難復原。

常見的反模式：

- 實作相同的自動化容錯移轉和容錯恢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。

未建立此最佳實務時的曝險等級：中

實作指引

- 自動執行復原路徑。對於較短的復原時間，請遵循災難復原計畫，以便在發生中斷時快速恢復 IT 系統。
  - 使用 Elastic Disaster Recovery 進行自動化容錯移轉和容錯恢復。Elastic Disaster Recovery 會持續將機器（包括作業系統、系統狀態組態、資料庫、應用程式和檔案）複寫到目標 AWS 帳戶 和偏好區域中的低成本暫存區域。如果發生災難，在選擇使用 Elastic Disaster Recovery 進行復原之後，Elastic Disaster Recovery 會將已複寫的伺服器的轉換自動化到 AWS上復原區域中完全佈建的工作負載中。
    - 使用 Elastic Disaster Recovery 進行容錯移轉和容錯恢復
    - AWS Elastic Disaster Recovery resources

資源

相關文件：

- APN 合作夥伴：可協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 自動化
- AWS Elastic Disaster Recovery
- 上工作負載的災難復原 AWS：雲端中的復原（AWS 白皮書）

相關影片：

- AWS re：Invent 2018：多區域主動應用程式的架構模式（ARC209-R2）

效能效率

效能效率支柱包括能夠有效率地使用雲端資源，以滿足效能需求，並隨著需求變更與技術發展來保持該效率需求。可以在效能達成效率支柱白皮書中找到實作指引。

最佳實務領域

- 架構選擇

- 運算與硬體
- 資料管理
- 聯網與內容交付
- 程序和文化

架構選擇

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 了解和了解可用的雲端服務和功能
- PERF01-BP02 使用來自雲端供應商或適當合作夥伴的指導，了解架構模式和最佳實務
- 0PERF01-BP03 將成本納入架構決策
- 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 使用基準測試來推動架構決策
- PERF01-BP07 使用資料驅動方法進行架構選擇

PERF01-BP01 了解和了解可用的雲端服務和功能

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。
- 移轉到雲端後，您不會將應用程式現代化。
- 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：透過考慮新服務和設定，您可以大幅提升效能、降低成本並最佳化維護工作負載所需的工作量。它也可以協助您加速啟用 time-to-value雲端產品的 。

未建立此最佳實務時的曝險等級：高

實作指引

AWS 持續推出新的服務和功能，可改善效能並降低雲端工作負載的成本。繼續使用 up-to-date這些新服務和功能對於在雲端維持效能效能至關重要。將工作負載架構現代化也可協助您提升生產力、推動創新並釋放更多成長機會。

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 產品，以識別並了解相關的服務和組態選項，協助您改善效能並降低成本和操作複雜性。
  - Amazon Web Services 雲端
  - AWS 學院
  - 有什麼新功能 AWS？
  - AWS 部落格
  - AWS 技能建置器
  - AWS 事件和網路研討會
  - AWS 培訓 和 憑證
  - AWS Youtube 頻道
  - AWS 研討會
  - AWS 社群
- 使用 Amazon Q 取得有關服務的相關資訊和建議。
- 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。
- 持續了解新雲端服務和功能。

資源

相關文件：

- Amazon Web Services 概觀

- Amazon EC2功能
- step-by-step使用 AWS 合作夥伴學習計劃學習
- AWS 訓練和認證
- 我成為 AWS 解決方案架構師的學習路徑
- AWS 架構中心
- AWS Partner Network
- AWS 解決方案程式庫
- AWS 知識中心
- 在 上建置現代應用程式 AWS

相關影片：

- AWS re：Invent 2023 - Amazon 的新功能 EC2
- AWS re：Invent 2022 - 使用 Amazon 降低您的營運和基礎設施成本 ECS
- AWS re：Invent 2023 - 使用 建立具有效率、敏捷性和創新的雲端 AWS
- AWS re：Invent 2022 - 部署 ML 模型，以高效能和低成本進行推論
- This is my Architecture

相關範例：

- AWS 範例
- AWS SDK 範例

PERF01-BP02 使用來自雲端供應商或適當合作夥伴的指導，了解架構模式和最佳實務

使用文件、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。這些資源可協助檢閱和改善架構，以實現最佳效能。

常見的反模式：

- 您可以使用 AWS 作為常見的雲端提供者。
- 您使用 AWS 服務的方式並非為其設計。
- 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：使用雲端供應商或適當合作夥伴的指引，可協助您針對工作負載做出正確的架構選擇，並讓您對決策充滿信心。

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，可協助您建置和管理高效的雲端工作負載。 AWS 文件提供程式碼範例、教學課程和詳細的服務說明。除了文件之外， AWS 還提供訓練和認證計畫、解決方案架構師和專業服務，可協助客戶探索雲端服務的不同層面，並在 上實作高效率的雲端架構 AWS。

利用這些資源來深入了解寶貴的知識和最佳實務、節省時間並在 AWS 雲端中取得更好的成果。

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，並取得更好的效能。
  - AWS 文件（例如使用者指南和白皮書）
  - AWS 部落格
  - AWS 培訓 和 憑證
  - AWS Youtube 頻道
- 加入 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 AWS 合作夥伴學習計劃學習 step-by-step
  - AWS 事件和網路研討會
  - AWS 研討會
  - AWS 社群
- 當您需要額外指引或產品資訊時，請聯絡 AWS 尋求協助。 AWS Solutions Architects 和 AWS Professional Services 提供解決方案實作的指引。AWS 合作夥伴提供 AWS 專業知識，協助您為企業解鎖敏捷性和創新能力。
- 如果您需要技術支援才能有效使用服務，請使用 AWS Support。我們的支援計劃旨在為您提供工具和專業知識的正確組合，讓您可以成功使用 ， AWS 同時最佳化效能、管理風險並控制成本。

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 - 使用 Amazon 的進階事件驅動模式 EventBridge
- AWS re：Invent 2023 - 在 上實作分散式設計模式 AWS
- AWS re：Invent 2023 - 應用程式架構作為程式碼

相關範例：

- AWS 範例
- AWS SDK 範例
- AWS 分析參考架構

0PERF01-BP03 將成本納入架構決策

將成本納入架構決策中，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，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。
- 您不會針對開放原始碼解決方案評估授權解決方案。
- 您不會定義儲存區生命週期政策。
- 您不會檢閱 的新服務和功能 AWS 雲端。
- 您只能使用區塊儲存。

建立此最佳實務的優勢：將成本納入到決策中可讓您使用更有效率的資源並探索其他投資。

未建立此最佳實務時的曝險等級：中

實作指引

優化工作負載成本可以提高資源利用率並避免雲端工作負載中的浪費。將成本納入架構決策中，通常包括適當調整工作負載元件大小以及啟用彈性，進而提高雲端工作負載效能的效率。

實作步驟

- 確立成本目標，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)。可使用 AWS Pricing Calculator 和 AWS Cost Explorer 找出工作負載中的關鍵成本驅動因素。
- 了解雲端中的定價模式，例如隨需執行個體、預留執行個體、Savings Plans 和 Spot 執行個體。
- 使用 Well-Architected 成本最佳實務，針對成本最佳化這些關鍵元件。
- 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 取得不可接受成本的警示。
  - 使用 AWS Compute Optimizer 或 AWS Trusted Advisor 得成本最佳化建議。
  - 使用 AWS Cost Anomaly Detection 取得自動化成本異常偵測和根本原因分析。

資源

相關文件：

- 什麼是 AWS Billing and Cost Management？
- 使用 進行成本最佳化 AWS
- 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 成本最佳化的新功能

- AWS re：Invent 2023 - 最佳化成本和效能，並追蹤緩解進度
- 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說明手冊
- 啟動最佳化：調整應用程式效能以實現最高效率
- 無伺服器最佳化研討會 (效能與成本)
- 擴充經濟高效的架構

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進時，判斷哪些選擇會影響客戶和工作負載效率。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：評估潛在的效能相關改善項目時，必須判斷技術變更的權衡是否符合工作負載要求。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

根據效能和客戶影響，識別架構中的關鍵領域。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

- 了解您的工作負載需求 和 SLAs。

- 清楚定義評估因素。因素可能與工作負載的成本、可靠性、安全性和效能有關。
- 選擇可滿足需求的架構和服務。
- 執行實驗和概念驗證（POCs），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、效能和成本的權衡。通常情況下，優先考慮其中兩個因素會以犧牲第三個因素為代價。

資源

相關文件：

- Amazon 建置者資料中心
- Amazon QuickSight KPIs
- Amazon CloudWatch RUM
- X-Ray 文件
- 了解恢復模式和權衡取捨以便在雲端中高效進行架構

相關影片：

- 透過 Amazon 最佳化應用程式 CloudWatch RUM
- AWS re：Invent 2023 - 容量、可用性、成本效益：挑選三項
- AWS re：Invent 2023 - 鬆散耦合系統的進階整合模式和權衡

相關範例：

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

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提高效率。

常見的反模式：

- 您允許各種可能會影響公司管理開銷的技術。

建立此最佳實務的優勢：為架構、技術和供應商選擇制定政策，可讓您快速做出決策。

未建立此最佳實務時的曝險等級：中

實作指引

在選擇資源和架構方面擁有內部政策，提供在選擇架構時要遵循的標準和準則。這些準則可簡化在選擇合適的雲端服務時的決策過程，並有助於提高效能效率。使用政策或參考架構來部署工作負載。將服務整合到您的雲端部署，然後使用效能測試以確認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，以識別最相關的政策。
- 使用 AWS 提供的適當參考架構或您的業界最佳實務。
- 針對常見情況，建立包含政策、標準、參考架構和規範指引的連續體。這樣做可以讓您的團隊更快地行動。如果適用，為您的垂直發展量身打造資產。
- 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 up-to-date掌握業界標準和 AWS 更新，確保您的政策和參考架構有助於最佳化您的雲端工作負載。

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 架構部落格

相關影片：

- This is my Architecture
- AWS re：Invent 2022 - 使用 SAP & AWS reference 架構為您的企業加速價值

相關範例：

- AWS 範例

- AWS SDK 範例

PERF01-BP06 使用基準測試來推動架構決策

對現有工作負載的效能進行基準化分析，以了解工作負載在雲端的效能，並根據該資料推動架構決策。

常見的反模式：

- 您倚賴不代表工作負載特性的常見基準。
- 您將客戶的意見回饋和看法作為唯一基準。

建立此最佳實務的優勢：對目前的實作進行基準測試可讓您衡量效能改進。

未建立此最佳實務時的曝險等級：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的效能。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測試，或使用業界標準測試，例如 TPC-DS ，對您的工作負載進行基準測試。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，以獲得有效結果。多次執行相同的基準化分析，以確認您已擷取到隨時間推移出現的任何變化。

由於基準化分析的速度通常比負載測試要快，因此可以在部署管道中盡早使用基準化分析，以便能更快提供有關效能偏差的回饋。當您評估元件或服務中的重大變更時，藉助基準化分析，您可以更快速地查看所做的變更是否合理。請務必使用基準化分析搭配負載測試，因為負載測試將告訴您工作負載在生產中的效能。

實作步驟

- 規劃和定義：
  - KPIs 為您的基準定義目標、基準、測試案例、指標（例如CPU使用率、延遲或輸送量）和 。
  - 關注使用者體驗方面的使用者需求，以及回應時間和可存取性等因素。
  - 找出工作負載適用的基準化分析工具。您可以使用 Amazon CloudWatch 之類的 AWS 服務，或是與工作負載相容的第三方工具。

- 配置並檢測：
  - 設定環境並配置資源。
  - 實作監控和日誌記錄以擷取測試結果。
- 基準化分析和監控：
  - 在測試期間執行基準化分析並監控指標。
- 分析並記錄：
  - 記錄基準化分析過程和調查結果。
  - 分析結果以找出瓶頸、趨勢和需要改善的領域。
  - 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：
  - 根據您的基準化分析來調整資源配置和分配。
  - 調整後重新測試您的工作負載，以驗證改進。
  - 記錄您的學習，並重複此過程以確定其他有待改進的領域。

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
- 基因體工作流程，第 5 部分：自動化基準測試
- 對 Amazon 中的端點部署進行基準測試和最佳化 SageMaker JumpStart

相關影片：

- AWS re：Invent 2023 - 冷 AWS Lambda 啟動基準
- 在雲端中對有狀態服務進行基準測試
- This is my Architecture
- 透過 Amazon 最佳化應用程式 CloudWatch RUM

- Amazon CloudWatch Synthetics 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 Client

PERF01-BP07 使用資料驅動方法進行架構選擇

為架構選擇定義清晰、資料驅動型方法，以確認是否使用正確的雲端服務和組態，來滿足特定業務需求。

常見的反模式：

- 您假設目前的架構是靜態的，且不應隨著時間而更新。
- 您的架構選擇基於猜測和假設。
- 您會隨時間導入架構變更，而且無需理由佐證。

建立此最佳實務的優勢：透過採用明確定義的方法來做出架構選擇，您可以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

使用雲端或外部資源 (例如已發布的使用案例或白皮書) 的內部經驗和知識，在架構中選擇資源和服務。您應擁有一個明確定義的流程，鼓勵對工作負載中可能使用的服務進行實驗和基準化分析。

關鍵工作負載的待辦項目不僅應包括可提供與業務和使用者相關的功能的使用者故事，還包括構成工作負載架構跑道的技術故事。這條跑道了解科技和新服務的新進展，並根據資料和適當理由採用這些技術和新服務。這證明該架構仍然面向未來，不會停滯不前。

實作步驟

- 與關鍵利益相關者互動，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。

- 建立架構跑道或技術待辦項目，系統會優先處理這些項目與功能待辦事項。
- 評價和評估不同的雲端服務 (如需詳細資訊，請參閱 PERF01-BP01 了解和了解可用的雲端服務和功能)。
- 探索符合效能需求的不同架構模式，例如微型服務或無伺服器 (如需詳細資訊，請參閱 PERF01-BP02 使用來自雲端供應商或適當合作夥伴的指導，了解架構模式和最佳實務)。

- 諮詢其他團隊、架構圖和資源，例如 AWS 解決方案架構師、AWS 架構中心 和 AWS Partner Network，以協助您為工作負載選擇正確的架構。

- 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。
- 實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。
- 根據學習、新技術和指標 (其指出目前方法中需要的變更或問題)，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
- 在 上建置 End-to-End資料驅動應用程式的架構模式 AWS

相關影片：

- This is my Architecture
- AWS re：Invent 2021 - 資料驅動的企業：從願景到價值
- AWS re：Invent 2022 - 提供永續、高效能的架構
- AWS re：Invent 2023 - 最佳化成本和效能，並追蹤緩解進度
- AWS re：Invent 2022 - AWS optimization：立即結果的可行步驟

相關範例：

- AWS 範例

- AWS SDK 範例

運算與硬體

問題

- PERF 2. 如何在工作負載中選取和使用運算資源？

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 為您的工作負載選取最佳運算選項
- PERF02-BP02 了解可用的運算組態和功能
- PERF02-BP03 收集運算相關指標
- PERF02-BP04 設定和正確大小的運算資源
- 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 為您的工作負載選取最佳運算選項

為工作負載選擇最合適的運算選項，可讓您改善效能、減少不必要的基礎設施成本，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。
- 缺乏對雲端運算選項、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的認識。
- 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。

建立此最佳實務的優勢：可以透過找出運算需求並根據可用選項進行評估，提高工作負載的資源效率。

未建立此最佳實務時的曝險等級：高

實作指引

為了最佳化您的雲端工作負載以取得效能效率，請務必針對您的使用案例和效能需求選擇最適當的運算選項。 AWS 提供各種運算選項，以因應雲端中的不同工作負載。例如，您可以使用 Amazon EC2 來啟動和管理虛擬伺服器、AWS Lambda執行程式碼，而不必佈建或管理伺服器、Amazon ECS 或 Amazon EKS 來執行和管理容器，或平行AWS Batch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。也可以考慮在單一工作負載中使用多種類型的運算解決方案，因為每種運算解決方案都有自己的優點和缺點。

下列步驟會引導您選取正確的運算選項，以符合您的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。需要考慮的關鍵需求包括處理需求、流量模式、資料存取模式、擴展需求和延遲需求。
- 了解適用於工作負載的不同 AWS 運算服務。如需詳細資訊，請參閱PERF01-BP01 了解和了解可用的雲端服務和功能。以下是一些關鍵的 AWS 運算選項、其特性和常見使用案例：

AWS 服務	重要特性	常用案例
Amazon Elastic Compute Cloud （Amazon EC2）	擁有專為硬體、授權要求、大規模選取的不同執行個體系列、處理器類型與運算加速器設計的選項	平移遷移、整合型應用程式、混合環境、企業應用程式
Amazon Elastic Container Service （Amazon ECS）、Amazon Elastic Kubernetes Service （Amazon EKS）	輕鬆的部署、一致的環境、可擴展	微型服務、混合環境
AWS Lambda	無伺服器運算服務可執行程式碼以回應事件，並自動管理基礎運算資源。	微型服務、事件驅動型應用程式
AWS Batch	有效且動態地佈建和擴展Amazon Elastic Container Service （Amazon ECS）、Amazon Elastic	HPC，訓練 ML 模型

AWS 服務	重要特性	常用案例
	Kubernetes Service（Amazon EKS）和AWS Fargate運算資源，並根據您的任務需求選擇使用隨需或 Spot 執行個體	
Amazon Lightsail	預先設定用於執行小型工作負載的 Linux 和 Windows 應用程式	簡易網路應用程式、自訂的網站

- 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 (例如修補和擴展)。
- 在非生產環境中執行實驗和基準測試，以確定哪個運算選項最能滿足您的工作負載需求。
- 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 Amazon CloudWatch 和 最佳化服務等 AWS 監控工具AWS Compute Optimizer，根據實際使用情況模式持續最佳化您的運算資源。

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 Containers：Amazon EKS Worker Nodes
- Amazon ECS Containers：Amazon ECS Container Instances
- 函數：Lambda 函數組態
- 容器的規範性指引
- 無伺服器的規範性指引

相關影片：

- AWS re：Invent 2023 - AWS Graviton：工作負載 AWS 的最佳價格效能
- AWS re：Invent 2023 - 中的新 Amazon Elastic Compute Cloud 生成 AI 功能 AMS
- 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 - 為新一代 Amazon Elastic Compute Cloud 提供支援：深入研究 Nitro 系統
- 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 ML 模型，以高效能和低成本進行推論
- AWS re：Invent 2019 - 最佳化 AWS 運算的效能和成本
- Amazon EC2基礎
- 以高效能和低成本部署機器學習 (ML) 模型以進行推論

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 自動擴展實現高效且彈性的工作負載
- 使用 Container Services 遷移至 AWS Graviton

PERF02-BP02 了解可用的運算組態和功能

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。
- 過度佈建運算資源以符合尖峰需求。

建立此最佳實務的優點：熟悉 AWS 運算功能和組態，以便您可以使用最佳化的運算解決方案來滿足您的工作負載特性和需求。

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功能，以支援不同的工作負載特性和需求。了解這些選項如何與您的工作負載互補，並確定哪種組態選項最適合您的應用程式。這些選項的範例包括執行個體系列、大

小、功能（GPU、I/O）、爆量、逾時、函數大小、容器執行個體和並行。如果您的工作負載已使用相同的運算選項超過四週，且您預期未來特性將保持不變，您可以使用 AWS Compute Optimizer來了解目前的運算選項是否適合 CPU和 記憶體角度的工作負載。

實作步驟

- 了解工作負載需求（例如CPU需求、記憶體和延遲）。
- 檢閱 AWS 文件和最佳實務，以了解有助於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較高 vCPU 與記憶體比率的工作負載。 • 記憶體最佳化執行個體提供大量記憶體，以支援記憶體密集型工作負載。 • 儲存體最佳化執行個體專為需要對本機儲存體進行高連續讀取和寫入存取權（IOPS）的工作負載而設計。
定價方式	• 隨需執行個體允許您按秒數或時數來使用運算能力，無須簽訂長期合約。這些執行個體適合於超過效能基準需求的爆量。 • Savings Plans 可大幅節省隨需執行個體，以換取在一年或三年內使用特定運算能力的承諾。 • Spot 執行個體可讓您以折扣價利用未使用的執行個體容量，用於無狀態、容錯的工作負載。
Auto Scaling	使用 Auto Scaling 設定，使運算資源與流量模式相符。
規模調整	• 使用 Compute Optimizer 取得機器學習支援的建議，了解哪些運算組態最符合您的運算特性。

組態選項	範例
	• 使用 AWS Lambda Power Tuning 為您的 Lambda 函數選擇最佳組態。
硬體型運算加速器	• 加速運算執行個體會比 CPU型替代方案更有效率地執行圖形處理或資料模式比對等函數。 • 對於機器學習工作負載，請善用工作負載特有的專用硬體，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 Containers：Amazon EKS Worker Nodes
- Amazon ECS Containers：Amazon ECS Container Instances
- 函數：Lambda 函數組態

相關影片：

- AWS re：Invent 2023 – AWS Graviton： AWS 工作負載的最佳價格效能
- AWS re：Invent 2023 – 中的新 Amazon EC2 生成 AI 功能 AWS Management Console
- AWS re：Invent 2023 – Amazon 的新功能 EC2
- AWS re：Invent 2023 – 智慧節省：Amazon EC2成本最佳化策略
- AWS re：Invent 2021 – 推動下一代 AmazonEC2：Nitro 系統的深度探索
- AWS re：Invent 2019 – Amazon EC2基礎
- AWS re：Invent 2022 – 最佳化 Amazon EKS 的效能和成本 AWS

相關範例：

- 運算最佳化工具示範程式碼
- Amazon EC2 Spot 執行個體研討會
- Amazon 的高效率和彈性工作負載 EC2 AWS Auto Scaling
- Graviton 開發人員研討會
- AWS 適用於 Microsoft 工作負載沉浸式日
- AWS 適用於 Linux 工作負載沉浸式日
- AWS Compute Optimizer 示範程式碼
- Amazon EKS研討會

PERF02-BP03 收集運算相關指標

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。
- 您只會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：收集效能相關指標有助於使應用程式效能與業務需求保持一致，確保符合工作負載需求。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 中 AWS 雲端，收集指標是改善安全性、成本效益、效能和永續性的重要步驟。 使用 Amazon CloudWatch 等監控服務 AWS ，提供廣泛的效能相關指標，為您提供寶貴的洞見。CPU 使用率、記憶體使用率、磁碟 I/O 和網路傳入和傳出等指標可以提供使用率層級或效能瓶頸的洞察。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，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

- 識別與您的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)。

  - Amazon EC2 預設指標
  - Amazon ECS 預設指標
  - Amazon EKS 預設指標
  - Lambda 預設指標
  - Amazon EC2記憶體和磁碟指標
- 為工作負載選擇並設定合適的日誌記錄和監控解決方案。
  - AWS 原生可觀測性
  - AWS Distro for OpenTelemetry
  - Amazon Managed Service for Prometheus
- 根據工作負載需求，為指標定義必要的篩選條件和彙總。
  - 使用 Amazon CloudWatch Logs 和指標篩選條件量化自訂應用程式指標
  - 使用 Amazon CloudWatch 策略標記收集自訂指標
- 為指標設定資料保留政策，以符合安全性和營運目標。
  - CloudWatch 指標的預設資料保留
  - CloudWatch 日誌的預設資料保留
- 如有必要，為指標建立警示和通知，可協助您主動回應效能相關問題。
  - 使用 Amazon CloudWatch 異常偵測建立自訂指標的警示
  - 使用 Amazon 為特定網頁建立指標和警示 CloudWatch RUM
- 使用自動化來部署指標和記錄彙總代理程式。
  - AWS Systems Manager 自動化
  - OpenTelemetry 收集器

資源

相關文件：

- 監控與可觀測性
- 最佳實務：使用 實作可觀測性 AWS
- Amazon CloudWatch 文件
- 使用 CloudWatch 代理程式從 Amazon EC2執行個體和內部部署伺服器收集指標和日誌
- 存取 的 Amazon CloudWatch Logs AWS Lambda
- 將 CloudWatch 日誌與容器執行個體搭配使用

- 發佈自訂指標
- AWS Answers：集中式日誌記錄
- AWS 發佈 CloudWatch 指標的服務
- 在 EKS上監控 Amazon AWS Fargate

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 – 建立有效的可觀測性策略
- AWS re：Invent 2023 – AWS Distro for 的無縫可觀測性 OpenTelemetry
- 上的應用程式效能管理 AWS

相關範例：

- AWS 適用於 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 設定和正確大小的運算資源

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，只能使用一個執行個體系列。
- 您可以忽略 AWS Cost Explorer 或 Compute Optimizer 的建議，以進行正確調整大小。
- 您未重新評估工作負載是否適用於新執行個體類型。
- 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：透過避免資源的過度佈建和佈建不足，適當調整運算資源的大小可確保雲端中的最佳操作。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的成本，而佈建不足可能會導致效能不佳和負面的客戶體驗。 AWS 提供AWS Compute Optimizer和 等工具AWS Trusted Advisor，使用歷史資料提供建議，以正確調整運算資源的大小。

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 Fleet 的屬性型執行個體類型選擇
  - 使用屬性型執行個體類型選取範圍來建立 Auto Scaling 群組
  - 利用 Karpenter 整合來最佳化 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用量之間的關係。使用此資料，可以選擇最適合您工作負載描述檔和效能目標的資源。
- 使用 Amazon 等監控工具來 AWS 監控資源用量 CloudWatch。
- 為運算資源選取適合的組態。
  - 對於暫時性工作負載，評估執行個體 Amazon CloudWatch 指標，例如 `CPUUtilization`，以識別執行個體是否使用不足或過度使用。
  - 對於穩定的工作負載，定期檢查AWS Compute Optimizer和 等工具 AWS 的許可化 AWS Trusted Advisor ，以識別最佳化和調整運算資源大小的機會。
- 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 進行雲端運算 AWS
- Amazon EC2執行個體類型

- Amazon ECS Containers：Amazon ECS Container Instances
- Amazon EKS Containers：Amazon EKS Worker Nodes
- 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 工作負載的最佳價格效能
- AWS re：Invent 2023 – 中的新 Amazon EC2 生成 AI 功能 AWS Management Console
- AWS re：Invent 2023 – Amazon 的新功能 EC2
- AWS re：Invent 2023 – 智慧節省：Amazon EC2成本最佳化策略
- AWS re：Invent 2021 – 推動下一代 AmazonEC2：Nitro 系統的深度探索
- 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。
- 使用與內部部署相同的大小規模準則 (通常是靜態基礎設施)。
- 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，並隨著流量變化提升可靠性。

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，提供彈性來動態擴展或縮減資源，以滿足需求的變化。結合與運算相關的指標，動態擴展允許工作負載自動回應變更，並使用最佳運算資源集來實現目標。

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。
- 基於排程的方法：按照排程來擴展可讓您根據可預測的負載變化來設定自己的擴展排程。
- 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向上擴展和縮減規模事件。

實作步驟

- 運算執行個體、容器和函數提供了彈性機制，可與自動擴展功能結合使用，或是作為服務功能提供。以下是自動擴展機制的幾個範例：

自動擴展機制	在哪裡使用
Amazon EC2 Auto Scaling	為確保您擁有正確數量的 Amazon EC2 執行個體，可處理應用程式的使用者負載。
Application Auto Scaling	自動擴展 Amazon 以外的個別 AWS 服務資源，EC2例如 AWS Lambda 函數或 Amazon Elastic Container Service （AmazonECS） 服務。
Kubernetes Cluster Autoscaler/Karpenter	自動擴展 Kubernetes 叢集。

- 擴展通常與運算服務相關，例如 Amazon EC2執行個體或 AWS Lambda 函數。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。如果您正在部署影片轉碼應用程式，則預期 100% CPU使用率，且不應是您的主要指標。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，請考慮下列 Amazon 指南EC2：
  - 指標應為有效的使用率指標，並說明執行個體的忙碌程度。

  - 指標值必須與 Auto Scaling 群組中的執行個體數成比例增加或減少。
- 對於 Auto Scaling 群組，確保使用動態擴展，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同時處理擴展事件 (擴充和縮減)。例如，您可以使用活動歷史記錄來驗證 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。透過預測性擴展，可以消除過度佈建容量的需求。如需詳細資訊，請參閱使用 Amazon EC2 Auto Scaling 進行預測擴展。

資源

相關文件：

- 使用 進行雲端運算 AWS
- Amazon EC2執行個體類型
- Amazon ECS Containers：Amazon ECS Container Instances
- Amazon EKS Containers：Amazon EKS Worker Nodes
- 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 Cluster 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 工作負載的最佳價格效能
- AWS re：Invent 2023 – AWS 管理主控台中的新 Amazon EC2 生成 AI 功能
- AWS re：Invent 2023 – Amazon 的新功能 EC2
- AWS re：Invent 2023 – 智慧節省：Amazon EC2成本最佳化策略
- AWS re：Invent 2021 – 推動下一代 AmazonEC2：Nitro 系統的深度探索
- 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
- 透過在 上執行來擴展 Amazon EKS工作負載 IPv6

PERF02-BP06 使用最佳化的硬體型運算加速器

使用硬體加速器比CPU以 為基礎的替代方案更有效率地執行某些函數。

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改善效能和降低成本。
- 您正在使用硬體型運算加速器執行任務，這些任務可以使用 CPU型替代方案更有效率。
- 您未監控GPU用量。

建立此最佳實務的優點：透過使用硬體型加速器，例如圖形處理單元（GPUs）和現場可程式設計閘道陣列（FPGAs），您可以更有效率地執行某些處理函數。

未建立此最佳實務時的曝險等級：中

實作指引

加速運算執行個體可讓您存取硬體型運算加速器，例如 GPUs和 FPGAs。這些硬體加速器會比CPU以為基礎的替代方案更有效率地執行圖形處理或資料模式比對等特定功能。許多加速的工作負載 (例如轉譯、轉碼和機器學習) 在資源使用方面變化很大。只在需要時執行此硬體，並在不需要時自動停用它們，以提高整體效能的效率。

實作步驟

- 確定哪些加速運算執行個體可以滿足您的需求。
- 對於機器學習工作負載，請善用工作負載特有的專用硬體，例如 AWS Trainium 、AWS Inferentia 和 Amazon EC2 DL1。Inf2 執行個體等 AWS Inf2 執行個體比類似的 Amazon EC2執行個體提供高達 50% 的效能/瓦特。
- 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 的指標，例如 `utilization_memory` `utilization_gpu`和 GPUs，如使用 Amazon 收集NVIDIAGPU指標 CloudWatch中所示。
- 優化硬體加速器的程式碼、網路運作和設定，以確保系統會充分利用基礎硬體。
  - 最佳化GPU設定

  - GPU 深度學習中的監控和最佳化 AMI
  - 最佳化 I/O 以在 Amazon 中進行深度學習訓練GPU的效能調校 SageMaker
- 使用最新的高效能程式庫和GPU驅動程式。
- 使用自動化在不使用時釋出GPU執行個體。

資源

相關文件：

- 在 Amazon Elastic Container Service GPUs上使用
- GPU 執行個體
- 具有 AWS Trainium 的執行個體
- 具有 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 EC2VT1執行個體
- 如何為工作負載選擇適當的 Amazon EC2執行個體類型？
- 選擇使用 Amazon 進行電腦視覺推論的最佳 AI 加速器和模型編譯 SageMaker

相關影片：

- AWS re：Invent 2021 - 如何選取 Amazon Elastic Compute Cloud GPU執行個體進行深度學習
- AWS re：Invent 2022 - 【NEW LAUNCH！】 介紹 AWS Inferentia2-based Amazon EC2 Inf2 執行個體
- AWS re:Invent 2022 - 使用 AWS Trainium 加速深度學習並加快創新速度
- AWS re：Invent 2022 - AWS 使用 進行深度學習NVIDIA：從訓練到部署

相關範例：

- Amazon SageMaker 和 NVIDIA GPU Cloud （ NGC ）
- SageMaker 與 Trainium 和 Inferentia 搭配使用，以最佳化深度學習訓練和推論工作負載
- 在 Amazon 中使用 Amazon Elastic Compute Cloud Inf1 執行個體最佳化NLP模型 SageMaker

資料管理

問題

- PERF 3. 如何在工作負載中儲存、管理和存取資料？

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）、必要的輸送量、存取頻率（線上、離線、封存）、更新頻率（WORM、動態），以及可用性和耐久性限制而有所不同。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 使用專門建置的資料存放區，最適合支援您的資料存取和儲存需求
- PERF03-BP02 評估資料存放區的可用組態選項
- PERF03-BP03 收集並記錄資料存放區效能指標
- PERF03-BP04 實作策略來改善資料存放區中的查詢效能
- PERF03-BP05 實作使用快取的資料存取模式

PERF03-BP01 使用專門建置的資料存放區，最適合支援您的資料存取和儲存需求

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。

建立此最佳實務的優勢：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曝險等級：高

實作指引

在選擇和實作資料儲存時，請確定查詢、擴展和儲存特性支援工作負載資料需求。 AWS 提供許多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、關聯性、索引鍵值、文件、記憶體內、圖形、時間序列和分類帳資料庫。每個資料管理解決方案都有為您提供的選項和組態，以支援您的使用案例和資料模型。透過了解資料特性和需求，您可以擺脫單片儲存技術和限制性 one-size-fits-all 的方法，以適當地管理資料。

實作步驟

- 對您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 （原子性、一致性、隔離、耐久性） 需求
  - 資料存取模式 (大量讀取或大量寫入)
  - Latency (延遲)
  - 輸送量
  - IOPS （每秒的輸入/輸出操作）
  - 資料保留期間
- 了解適用於 工作負載的不同資料存放區 （儲存和資料庫服務） AWS ，其可滿足您的資料特性，如所述PERF01-BP01 了解和了解可用的雲端服務和功能。 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
物件儲存	Amazon Simple Storage Service (Amazon S3)	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。

類型	AWS 服務	重要特性
封存儲存	Amazon S3 Glacier	專為資料封存而打造。
串流儲存空間	Amazon Kinesis Amazon Managed Streaming for Apache Kafka（Amazon MSK）	快速地擷取和儲存串流資料。
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。
共用檔案系統	Amazon FSx	以最新的 AWS 運算解決方案為基礎，支援四個常用的檔案系統： NetApp ONTAP、Open ZFS、Windows File Server 和 Lustre。FSxAmazonlatency、輸送量和 IOPS會因檔案系統而異，因此在選擇適合您工作負載需求的檔案系統時，應該考慮這一點。
區塊儲存	Amazon Elastic Block Store（Amazon EBS）	專為 Amazon Elastic Compute Cloud（Amazon ）設計的可擴展、高效能區塊儲存服務EC2。Amazon EBS包含交易IOPS型 密集型工作負載的 SSD後端儲存體，以及輸送量密集型工作負載的 HDD後端儲存體。

類型	AWS 服務	重要特性
關聯式資料庫	Amazon Aurora、Amazon RDS、Amazon Redshift 。	旨在支援 ACID（原子性、一致性、隔離、耐久性）交易，並維持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）和電子商務會使用關聯式資料庫來存放其資料。
鍵值資料庫	Amazon DynamoDB	已針對常見的存取模式進行最佳化，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。
文件資料庫	Amazon DocumentDB	旨在將半結構化資料儲存為JSON類似 文件。這些資料庫可協助開發人員快速建置和更新應用程式，例如內容管理、目錄和使用者設定檔。
記憶體資料庫	Amazon ElastiCache、Amazon MemoryDB for Redis	適用於需要即時存取資料、最低延遲和最高輸送量的應用程式。您可以將記憶體資料庫用於應用程式快取、工作階段管理、遊戲排行榜、低延遲 ML 特徵存放區、微型服務簡訊系統，以及高輸送量串流機制

類型	AWS 服務	重要特性
圖形資料庫	Amazon Neptune	適用於此類應用程式：必須在高度連線圖形資料集之間，大規模導覽和查詢數百萬個關係，並且在過程中僅有毫秒延遲。許多公司使用圖形資料庫進行詐騙偵測、社交聯網和推薦引擎。
時間序列資料庫	Amazon Timestream	可快速地從隨時間變化的資料收集、合成和衍生洞見。IoT 應用程式 DevOps和工業遙測可以利用時間序列資料庫。
寬欄	Amazon Keyspaces (適用於 Apache Cassandra)	可使用表格、列和欄，但與關聯式資料庫不同，在同一個表格中，欄的名稱和格式會因列而異。您通常會在大規模工業應用程式中看到寬欄存放區，用於設備維護、叢集管理和路由優化。
總帳	Amazon Quantum Ledger 資料庫（Amazon QLDB）	可提供集中化且受信任的機構，為每個應用程式維護可擴展、不可變且以密碼編譯方式驗證的交易記錄。我們會看到用於記錄、供應鏈、註冊甚至銀行交易系統的總帳資料庫。

- 如果您要建置資料平台，請在 上利用現代資料架構 AWS 來整合您的資料湖、資料倉儲和專用資料存放區。
- 為工作負載選擇資料存放區時，需要考慮的關鍵問題如下：

問題	重要考慮事項
如何建構資料？	• 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 之類的無SQL資料庫 • 對於鍵值資料，請考慮 DynamoDB、Amazon ElastiCache （Redis OSS） 或 Amazon MemoryDB
需要哪種層級的參考完整性？	• 對於外部金鑰限制，Amazon RDS 和 Aurora 等關聯式資料庫可以提供此完整性層級。 • 一般而言，在無SQL資料模型中，您會將資料還原成單一文件或要在單一請求中擷取的文件集合，而不是跨文件或資料表加入。
是否需要符合 ACID（原子性、一致性、隔離、耐久性） 要求？	• 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。 • 如果沒有SQL資料庫 需要強烈的一致性，您可以搭配 DynamoDB使用強烈的一致性讀取。
儲存要求如何隨時間變更？ 這如何影響可擴展性？	• 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體上有上限，而且一旦達到這些限制，通常必須使用碎片等機制進行水平分割。
讀取查詢與寫入查詢的比例是多少？ 快取可能改善效能嗎？	• 如果DAX資料庫是 DynamoDB ，則大量讀取工作負載可以從快取層中受益，例如 ElastiCache或 。 DynamoDB • 讀取也可以卸載，以讀取具有關聯式資料庫的複本，例如 Amazon RDS。

問題	重要考慮事項
儲存和修改（OLTP - 線上交易處理）或擷取和報告（OLAP - 線上分析處理）是否具有更高的優先順序？	• 對於高輸送量讀取即交易處理，請考慮無SQL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 Amazon Redshift 之類的資料欄式資料庫，或將資料匯出至 Amazon S3，並使用 Athena 或 Amazon QuickSight執行分析。
資料需要哪種層級的耐久性？	• Aurora 會自動跨區域內的三個可用區域複寫資料，這表示資料高度耐用且資料遺失的機會較低。 • DynamoDB 會自動跨多個可用區域複寫，具有高可用性和資料耐久性。 • Amazon S3 提供 11 個九的耐用性。許多資料庫服務，例如 Amazon RDS和 DynamoDB ，都支援將資料匯出至 Amazon S3 以進行長期保留和封存。
是否希望擺脫商務資料庫引擎、或授權成本？	• 考慮開放原始碼引擎，例如 PostgreSQL 和 MySQL on Amazon RDS或 Aurora。 • 利用 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
對資料庫的操作期望是什麼？ 移至受管服務是否為主要問題？	• 利用 Amazon RDS而非 Amazon EC2，以及使用 DynamoDB 或 Amazon DocumentDB 而非自我託管無SQL資料庫，可以降低營運開銷。

問題	重要考慮事項
目前如何存取資料庫？ 它是否僅存取應用程式，還是有商業智慧（BI）使用者和其他連線off-the-shelf的應用程式？	• 如果您依賴於外部工具，則可能必須保持與其所支援之資料庫的相容性。Amazon與其支援的引擎版本RDS完全相容，包括Microsoft SQL Server、Oracle、My SQL和Postgre SQL。

- 在非生產環境中執行實驗和基準測試，以確定哪個資料存放區最能滿足您的工作負載需求。

資源

相關文件：

- Amazon EBS磁碟區類型
- Amazon EC2 Storage
- AmazonEFS：Amazon EFS Performance
- 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
- 使用 的雲端儲存 AWS
- 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 之間選擇 RDS
- 實作 Amazon 的最佳實務 ElastiCache

相關影片：

- AWS re：Invent 2023：改善 Amazon Elastic Block Store 效率並提高成本效益
- AWS re：Invent 2023：使用 Amazon Simple Storage Service 最佳化儲存價格和效能
- AWS re：Invent 2023：在 Amazon Simple Storage Service 上建置和最佳化資料湖
- AWS re：Invent 2022：在 上建置現代資料架構 AWS
- AWS re：Invent 2022：在 上建置資料網格架構 AWS
- AWS re：Invent 2023：深入探索 Amazon Aurora 及其創新
- AWS re：Invent 2023：使用 Amazon DynamoDB 進行進階資料建模
- AWS re：Invent 2022：使用專用資料庫現代化應用程式
- 深入探討 Amazon DynamoDB：進階設計模式

相關範例：

- AWS 目的建構資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構沉浸式日
- 在 上建置資料網格 AWS
- 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 評估資料存放區的可用組態選項

了解並評估資料存放區可用的各種功能和組態選項，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EBS，例如 Amazon 。
- 您可以IOPS針對所有工作負載使用佈建的 ，而不需要針對所有儲存層進行實際測試。
- 您不知道所選資料管理解決方案的組態選項。

- 您完全依賴於增加執行個體大小，而不查看其他可用的組態選項。
- 您並不測試資料存放區的擴展特性。

建立此最佳實務的優勢：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曝險等級：中

實作指引

工作負載可以根據資料儲存和存取需求，使用一個或多個資料存放區。要優化效能達成效率和成本，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、一致性要求、連線集區以及快取選項。嘗試使用這些不同的組態選項來改善效能達成效率指標。

實作步驟

- 了解資料存放區的目前組態 (例如執行個體類型、儲存體大小或資料庫引擎版本)。
- 檢閱 AWS 文件和最佳實務，以了解建議組態選項，以協助改善資料存放區的效能。要考慮的關鍵資料存放區選項如下：

組態選項	範例
卸載讀取 (例如讀取複本和快取)	• 對於 DynamoDB 資料表，您可以使用 卸載用於快取DAX的讀取。 • 您可以建立 Amazon ElastiCache （Redis OSS） 叢集，並將應用程式設定為先從快取讀取，如果請求的項目不存在，則會傳回資料庫。 • 諸如 Amazon RDS和 Aurora 等關聯式資料庫，以及佈建的SQL Neptune 和 Amazon DocumentDB 等資料庫都不支援新增僅供讀取複本來卸載工作負載的讀取部分。 • 諸如 DynamoDB 等無伺服器資料庫將自動擴展。確保您已佈建足夠的讀取容量單位（RCU），以處理工作負載。

組態選項	範例
擴展寫入 (例如分區金鑰碎片或引進佇列)	• 對於關聯式資料庫，您可以增加執行個體的大小以容納增加的工作負載，或增加佈建的 IOPs 以允許基礎儲存的輸送量增加。 • 也可以在資料庫前面引入佇列，而不是直接寫入資料庫。此模式可讓您將擷取與資料庫分離並控制流速，這樣資料庫就不會不堪重負。 • 批次處理寫入請求，而不是建立許多短期交易，這有助於改善高寫入量關聯式資料庫的輸送量。 • DynamoDB 等無伺服器資料庫可以自動擴展寫入輸送量，也可以根據容量模式調整佈建的寫入容量單位 （WCU）。 • 當您達到指定分割區索引鍵的輸送量限制時，仍然可能會遇到熱分割區的問題。透過選擇更均勻分佈的分割區索引鍵，或分片寫入分割區索引鍵，來緩解此問題。
使用政策來管理資料集的生命週期	• 可以使用 Amazon S3 生命週期，以在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。可以利用 Amazon S3 Storage Lens 指標，找出生命週期管理中的最佳化機會和差距。 • Amazon EFS生命週期管理會自動管理檔案系統的檔案儲存。

組態選項	範例
連線管理與集區	• Amazon RDS Proxy 可與 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，但請考慮已佈建的容量和自動擴展政策來處理負載中的高峰。

- 在非生產環境中執行實驗和基準測試，以確定哪個組態選項能滿足您的工作負載需求。
- 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，使用實際用量模式持續最佳化您的資料存放區。

資源

相關文件：

- AWS的雲端儲存
- Amazon EBS磁碟區類型
- Amazon EC2 Storage
- AmazonEFS：Amazon EFS Performance
- 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：使用 Amazon Simple Storage Service 最佳化儲存價格和效能
- AWS re:Invent 2023：在 Amazon Simple Storage Service 上建置和最佳化資料湖
- AWS re：Invent 2023： AWS 檔案儲存的新功能
- AWS re:Invent 2023：深入了解 Amazon DynamoDB

相關範例：

- AWS 目的建構資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構沉浸式日
- Amazon EBS Autoscale
- 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績效指標，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。
- 您只會將指標發布到團隊使用的內部工具，而不會全面了解您的工作負載。
- 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。

- 您只監控系統層級指標，而沒有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工作負載的正常行為和需求。異常模式可以更快地識別和偵錯，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

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。這可讓您偵測異常情況，並根據業務指標衡量效能，以確認您是否滿足工作負載需求。

指標應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、可用磁碟儲存體、磁碟 I/O、快取命中率，以及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、索引用量、資料表鎖定、查詢逾時，以及開啟的連線數量。此資料對於了解工作負載的執行情況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，以調整和優化工作負載的資源。

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

- 找出要追蹤的資料存放區關鍵效能指標。
  - Amazon S3 指標和維度
  - 在 Amazon RDS執行個體中監控 的指標
  - 使用 Amazon 上的 Performance Insights 監控資料庫負載 RDS
  - 增強型監視概觀
  - DynamoDB 指標和維度
  - 監控 DynamoDB Accelerator
  - 使用 Amazon 監控 Amazon MemoryDB CloudWatch
  - 應監控哪些指標？
  - 監控 Amazon Redshift 叢集效能
  - Timestream 指標和維度
  - Amazon Aurora 的 Amazon CloudWatch 指標
  - Amazon Keyspaces (適用於 Apache Cassandra) 中的日誌記錄和監控
  - 監控 Amazon Neptune 資源

- 使用核准的日誌記錄和監控解決方案來收集這些指標。Amazon CloudWatch 可以收集架構中資源的指標。您還可以收集和發布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警示，指出何時違反閾值。
- 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。
  - Amazon DevOpsGuru for Amazon RDS提供效能問題的可見性，並提出修正動作的建議。
- 在監控和日誌記錄解決方案中設定資料保留，以符合安全性和營運目標。
  - CloudWatch 指標的預設資料保留
  - CloudWatch 日誌的預設資料保留

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
- 使用 的雲端資料庫 AWS
- Amazon RDS Performance Insights

相關影片：

- AWS re：Invent 2022 - Amazon RDS和 Aurora 的效能監控，採用 Autodesk
- 使用 Amazon DevOpsGuru for Amazon 進行資料庫效能監控和調整 RDS
- AWS re：Invent 2023 - AWS 檔案儲存的新功能
- AWS re：Invent 2023 - 深入探索 Amazon DynamoDB
- AWS re：Invent 2023 - 在 Amazon S3 上建置和最佳化資料湖
- AWS re：Invent 2023 - AWS 檔案儲存的新功能
- AWS re：Invent 2023 - 深入探索 Amazon DynamoDB

- 在 Amazon 上監控 Redis 工作負載的最佳實務 ElastiCache

相關範例：

- AWS 資料集擷取指標收集架構
- Amazon RDS監控研討會
- AWS 目的建構資料庫研討會

PERF03-BP04 實作策略來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：

- 您沒有分割資料存放區中的資料。
- 您在資料存放區中僅以一種檔案格式儲存資料。
- 您沒有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。這也有助於提高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，這些都有助最佳化資料的儲存和存取。

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。
- 找出資料存放區中執行速度緩慢的查詢，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 EXPLAINANALYZE中使用 EXPLAIN和
- 實作策略以改善查詢效能。有些關鍵策略包括下列情況：

- 使用資料欄檔案格式（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並縮短資料掃描時間。
  - 在 Athena 中分割資料
  - 分割區與資料分配
- 在查詢中對共同欄進行資料索引編制。
- 針對頻繁查詢使用具體化視觀表。
  - 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。當您聯結兩個資料表時，請在聯結左側指定較大的資料表，並在聯結右側指定較小的資料表。
- 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，例如清空、重新索引以及執行統計。
- 在非生產環境中實驗和測試策略。

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 的最佳實務 ElastiCache
- 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 - Amazon RDS和 Aurora 的效能監控，採用 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 - 在無伺服器或資料庫的情形下查詢資料
- AWS 目的建構資料庫研討會

PERF03-BP05 實作使用快取的資料存取模式

實作可受益於快取資料的存取模式，以便快速擷取經常存取的資料。

常見的反模式：

- 快取頻繁變更的資料。
- 您依賴快取資料，就好像它是持久存儲並始終可用一樣。
- 您不考慮快取資料的一致性。
- 您不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風險暴露等級：中

實作指引

快取是旨在存儲資料的軟體或硬體組件，以便更快或更有效地滿足未來對相同資料的請求。如果存儲在快取中的資料丟失，可以透過重複之前的計算或從另一個資料存放區中擷取來進行重建。

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。可以在應用程式的多個層級快取資料，例如在進行遠端呼叫的應用程式內 (稱為用戶端快取)，或使用快速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) 都可以在指定的時間內，在本機儲存其唯一查詢的結果。這可以先檢查本機用戶端快取，來減少網路對資料儲存的請求數量。如果結果不存在，則應用程式便可查詢資料儲存，並將這些結果儲存在本機。此模式允許每個用戶端將資料儲存在最接近的位置 (用戶端本身)，從而達到最低的延遲。當後端資料儲存無法使用時，用戶端也可以繼續提供某些查詢，從而提高整體系統的可用性。

這種方法的一個缺點是，當涉及多個用戶端時，它們可能會在本地存儲相同的快取資料。這會導致這些用戶端之間的重複儲存使用量和資料不一致。一個用戶端可能會快取查詢結果，一分鐘後，另一個用戶端可以執行相同查詢並獲得不同結果。

遠端快取

為了解決用戶端之間的重複資料問題，可以使用快速外部服務或遠端緩存來存儲查詢的資料。每個用戶端都會在查詢後端資料儲存之前檢查遠端快取，而非檢查本機資料存放區。此策略可實現用戶端之間更一致的回應、更好的儲存資料效率以及更高的快取資料量，因為儲存空間會獨立於用戶端進行擴展。

遠端快取的缺點是整個系統可能會遇到較高延遲，因為需要額外的網路跳轉來檢查遠端快取。用戶端快取可以與遠端快取一起用於多層級快取，以改善延遲。

實作步驟

- 識別可能受益於快取的資料庫APIs和網路服務。具有繁重讀取工作負載、比率高 read-to-write或規模昂貴的服務是快取的候選者。
  - 資料庫快取
  - 啟用API快取以增強回應能力
- 找出最適合您的存取模式的適當快取策略類型。
  - 快取策略
  - AWS 快取解決方案
- 遵循資料存放區的快取最佳實務。
- 針對平衡資料新鮮度和降低後端資料存放區壓力的所有資料，設定快取失效策略，例如 time-to-live（TTL）。
- 在用戶端中啟用自動連線重試、指數退避、用戶端逾時和連線集區等功能 (如果可用)，因為它們可以改善效能和可靠性。
  - 最佳實務：Redis 用戶端和 Amazon ElastiCache（RedisOSS）
- 監控快取命中率，目標為 80% 或更高。較低的值可能表示快取大小不足，或者無法從快取中受益的存取模式。
  - 應監控哪些指標？
  - 在 Amazon 上監控 Redis 工作負載的最佳實務 ElastiCache
  - 使用 Amazon 監控 Amazon ElastiCache（RedisOSS）的最佳實務 CloudWatch
- 實作資料複寫，將讀取卸載至多個執行個體，並提高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 監控 Amazon ElastiCache （RedisOSS） 的最佳實務 CloudWatch
- 應監控哪些指標？
- Amazon ElastiCache 白皮書的大規模效能
- 快取挑戰和策略

相關影片：

- Amazon ElastiCache Learning Path
- Amazon ElastiCache 最佳實務的成功設計
- AWS re：Invent 2020 - Amazon ElastiCache 最佳實務的成功設計
- AWS re：Invent 2023 - 【LAUNCH】 介紹 Amazon ElastiCache Serverless
- AWS re：Invent 2022 – 使用 Redis 重新構想資料層的 5 種絕佳方法
- AWS re：Invent 2021 - 在 Amazon 上深入探索 ElastiCache （Redis OSS）

相關範例：

- 使用 Amazon 提升我的SQL資料庫效能 ElastiCache （Redis OSS）

聯網與內容交付

問題

- PERF 4. 如何在工作負載中選取和設定聯網資源？

PERF 4. 如何在工作負載中選取和設定聯網資源？

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。

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 VPN為您的工作負載選擇適當的專用連線能力
- PERF04-BP04 使用負載平衡將流量分散到多個資源
- 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，以提供高效的效能並改善使用者體驗。

常見的反模式：

- 所有流量都流經現有資料中心。
- 可以透過中央防火牆路由所有流量，而非使用雲端原生網路安全工具。
- 您不了解實際用量需求的情況下佈建 AWS Direct Connect 連線。
- 定義聯網解決方案時，不會考慮工作負載特性和加密開銷。
- 對於雲端中的聯網解決方案，您可以使用內部部署概念和策略。

建立此最佳實務的優勢：了解聯網如何影響工作負載效能，有助於您識別潛在瓶頸、改善使用者體驗、提高可靠性並在工作負載變更時減少操作維護。

未建立此最佳實務時的風險暴露等級：高

實作指引

網路負責處理應用程式元件、雲端服務、邊緣網路和內部部署資料之間的連線，因此對工作負載效能可能有嚴重影響。除了工作負載效能外，使用者體驗也會受到網路延遲、頻寬、通訊協定、位置、網路擁塞、抖動、輸送量和路由規則的影響。

取得工作負載的已記錄在案的聯網需求清單，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，確定哪些服務符合您的工作負載網路特性。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提高效能達成效率的必要條件。

實作步驟：

- 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 (尖峰和頻率)、輸送量、加密、檢查和路由規則等指標。
- 了解金鑰 AWS 聯網服務，例如 VPCs、AWS Direct Connect、Elastic Load Balancing （ ELB ） 和 Amazon Route 53。
- 擷取下列主要聯網特性：

特性	工具與指標
基礎聯網特性	• VPC 流程日誌 • 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 ShieldAWS WAF、 和 AWS Network Firewall 指標
追蹤特性	• AWS X-Ray • VPC 可連線性分析工具 • 網路存取分析器 • Amazon Inspector • Amazon CloudWatch RUM

- 基準測試並測試網路效能：
  - 基準網路輸送量，因為當執行個體位於相同 時，某些因素可能會影響 Amazon EC2 網路效能 VPC。測量相同 中 Amazon EC2 Linux 執行個體之間的網路頻寬VPC。
  - 執行負載測試，試驗聯網解決方案和選項。

資源

相關文件：

- Application Load Balancer
- EC2 Linux 上的增強型網路
- EC2 Windows 上的增強型網路
- EC2 置放群組
- 在 Linux 執行個體上使用彈性網路轉接器 （ENA） 啟用增強型網路
- Network Load Balancer
- 使用 網路產品 AWS
- 轉換閘道
- 轉換到 Amazon Route 53 中的以延遲為基礎的路由
- VPC 端點

相關影片：

- AWS re：Invent 2023 - AWS networking 基礎
- AWS re：Invent 2023 - 聯網可以為您的應用程式做什麼？
- AWS re：Invent 2023 - 進階VPC設計和新功能
- AWS re：Invent 2023 - 開發人員雲端聯網指南
- AWS re：Invent 2019 - 與 AWS 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 Summit Online - 改善應用程式的全球網路效能
- AWS re：Invent 2020 - 使用 Well-Architected Framework 建立網路最佳實務和秘訣
- AWS re：Invent 2020 - AWS networking 大規模遷移的最佳實務

相關範例：

- AWS Transit Gateway 和可擴展安全解決方案
- AWS 網路研討會
- 網路防火牆實際操作研討會

- 在 上觀察並診斷您的網路 AWS
- 在 上尋找和解決網路設定錯誤 AWS

PERF04-BP02 評估可用的聯網功能

評估雲端中可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。
- 可以使用防火牆而非安全群組來篩選流量。
- 您TLS中斷流量檢查，而不是依賴安全群組、端點政策和其他雲端原生功能。
- 您只會使用子網路來分隔，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 全域 AWS 骨幹為客戶提供最佳的聯網體驗。

未建立此最佳實務時的風險暴露等級：高

實作指引

AWS 提供 AWS Global Accelerator和 Amazon CloudFront 等服務，可協助改善網路效能，而大多數 AWS 服務具有產品功能（例如 Amazon S3 Transfer Acceleration 功能），可最佳化網路流量。

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，以及它們對衡量的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。
  - 在建置統一的全球網路時，請考慮使用 AWS 雲端 WAN 來建置、管理和監控組織的網路。
  - 使用 Amazon CloudWatch Logs 指標 監控您的全域和核心網路。利用 Amazon CloudWatch RUM提供洞見，以協助識別、了解和增強使用者的數位體驗。
  - 檢視 AWS 區域 和可用區域之間以及每個可用區域內的彙總網路延遲，使用 AWS Network Manager深入了解應用程式效能與基礎 AWS 網路效能之間的關係。

  - 使用現有的組態管理資料庫（CMDB）工具或服務，例如 AWS Config來建立工作負載的庫存及其設定方式。
- 如果這是現有的工作負載，則請識別並記錄效能指標的基準，並著重於瓶頸和要改善的領域。效能相關聯網指標會依據業務需求和工作負載特性而有所不同。首先，這些指標對於檢閱工作負載可能很重要：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，則請執行負載測試以識別效能瓶頸。
- 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。查看下列主要聯網選項和功能：

改進機會	解決方案
網路的路徑或路由	使用網路存取分析器來識別路徑或路由。
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AWS Transit Gateway時，評估VPC對等和 之間的操作和效能權衡。AWS Transit Gateway 可簡化如何將所有互連VPCs，其可以跨越數千個 AWS 帳戶和 內部部署網路。使用 在多個帳戶 AWS Transit Gateway 之間共用您的 AWS Resource Access Manager。 請參閱PERF04-BP03 VPN為您的工作負載選擇適當的專用連線能力
網路服務	AWS Global Accelerator 是一種聯網服務，使用 AWS 全球網路基礎設施將使用者流量的效能提升高達 60%。 Amazon CloudFront 可以改善全球工作負載內容交付和延遲的效能。 使用 Lambda@edge 執行函數，以自訂 CloudFront 提供更接近使用者的內容、減少延遲並改善效能。

改進機會	解決方案
	Amazon Route 53 提供以延遲為基礎的路由、地理位置路由、地理位置鄰近性路由和以 IP 為基礎的路由選項，可協助您為全球使用者改善工作負載的效能。當工作負載分佈在全球範圍時，請檢閱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。
儲存功能資源	Amazon S3 Transfer Acceleration 是一項功能，可讓外部使用者受益於 的網路最佳化 CloudFront ，將資料上傳至 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製到多個區域，並透過提供一個存取點來簡化工作負載。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。

改進機會	解決方案
運算資源功能	Amazon EC2執行個體、容器和 Lambda 函數使用的彈性網路介面（ENA）以每個流程為基礎加以限制。檢閱置放群組，以最佳化EC2您的網路輸送量。若要避免個別流程的瓶頸，請將應用程式設計為使用多個流程。若要監控並取得運算相關網路指標的可見性，請使用 CloudWatch 指標 和 ethtool。`ethtool` 命令包含在ENA驅動程式中，並將其他可做為自訂指標發佈的網路相關指標公開給 CloudWatch。 Amazon Elastic Network Adapters（ENA）透過為叢集置放群組 內的執行個體提供更好的輸送量，進一步最佳化。 Elastic Fabric Adapter（EFA）是 Amazon EC2執行個體的網路介面，可讓您在 上執行需要大規模內部通訊的工作負載 AWS。 Amazon EBS最佳化執行個體使用最佳化的組態堆疊，並提供額外的專用容量來增加 Amazon EBS I/O。

資源

相關文件：

- Application Load Balancer
- EC2 Linux 上的增強型網路
- EC2 Windows 上的增強型網路
- EC2 置放群組
- 在 Linux 執行個體上使用彈性網路轉接器（ENA）啟用增強型網路
- Network Load Balancer
- 使用 網路產品 AWS

- 轉換到 Amazon Route 53 中的以延遲為基礎的路由
- VPC 端點
- VPC 流量日誌

相關影片：

- AWS re：Invent 2023 – 為下一步做好準備？ 設計網路實現增長和靈活性
- AWS re：Invent 2023 – 進階VPC設計和新功能
- AWS re：Invent 2023 – 雲端聯網開發人員指南
- AWS re：Invent 2022 – 深入探索 AWS 網路基礎設施
- AWS re：Invent 2019 – 與 AWS 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 和可擴展安全解決方案
- AWS 網路研討會
- 觀察並診斷您的網路
- 在 上尋找和解決網路設定錯誤 AWS

PERF04-BP03 VPN為您的工作負載選擇適當的專用連線能力

需要混合式連線來連接內部部署資源和雲端資源時，請佈建足夠的頻寬以滿足您的效能要求。預估混合工作負載的頻寬和延遲需求。這些數字將促進調整大小需求。

常見的反模式：

- 您僅評估網路加密要求VPN的解決方案。
- 不會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：選擇和設定適當的連線解決方案將提高工作負載的可靠性並將效能最大化。透過識別工作負載需求、提前規劃和評估混合解決方案，您可以最大限度地減少昂貴的實體網路變更和操作額外負荷，同時增加您的 time-to-value。

未建立此最佳實務時的曝險等級：高

實作指引

根據頻寬需求開發混合式聯網架構。AWS Direct Connect 允許您私下將內部部署網路與 AWS連線。需要高頻寬、低延遲同時可達到一致效能時，適合這個選項。VPN 連線會透過網際網路建立安全連線。在以下情況使用它：當只需要臨時連線時、當成本是一個考慮因素時、或者在使用 AWS Direct Connect時等待建立彈性物理網路連線作為應急措施時。

如果您的頻寬需求很高，您可以考慮多個 AWS Direct Connect VPN或服務。流量可以在服務之間進行負載平衡，但VPN由於延遲和頻寬差異，我們不建議在 AWS Direct Connect 和 之間進行負載平衡。

實作步驟

- 預估現有應用程式的頻寬和延遲要求。
  - 對於要移至 的現有工作負載 AWS，請利用內部網路監控系統的資料。
  - 針對您沒有監控資料的新工作負載或現有工作負載，請諮詢產品擁有者以確定足夠的效能指標，並且提供良好的使用者體驗。
- 選取專用連線或 VPN 做為連線選項。根據所有工作負載需求（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。
  - AWS Direct Connect 使用專用連線或託管連線，提供 AWS 環境的專用連線，範圍從 50 Mbps 到 100 Gbps。這可為您提供受管和受控的延遲以及佈建頻寬，因此您的工作負載可以有效率地連線到其他環境。使用 AWS Direct Connect 合作夥伴，您可以從多個環境進行 end-to-end連線，提供具有一致效能的延伸網路。 AWS 提供使用原生 100 Gbps、連結彙總群組（LAG）或BGP同等成本多路徑（ECMP）擴展直接連線連線頻寬。
  - 提供支援網際網路通訊協定安全的受管VPN服務 AWS Site-to-Site VPN（IPsec）。建立VPN連線時，每個VPN連線都包含兩個通道，以實現高可用性。
- 遵循 AWS 文件選擇適當的連線選項：
  - 如果您決定使用 AWS Direct Connect，請為您的連線選擇適當的頻寬。
  - 如果您使用 AWS Site-to-Site VPN 跨多個位置的 來連線至 AWS 區域，請使用加速 Site-to-SiteVPN連線來改善網路效能。
  - 如果您的網路設計包含透過 的IPSecVPN連線AWS Direct Connect，請考慮使用私有 IP VPN 來提高安全性並實現分割。AWS Site-to-Site 私有 IP VPN部署在傳輸虛擬介面（VIF）之上。

  - AWS Direct Connect SiteLink允許在全球各地的資料中心之間建立低延遲和冗餘連線，方法是透過AWS Direct Connect 位置 之間的最快路徑傳送資料，繞過 AWS 區域。
- 在部署到生產環境之前驗證連線設定。執行安全性和效能測試，以確保其符合您的頻寬、可靠性、延遲和合規性要求。
- 定期監控您的連線效能和使用情況，並視需要進行最佳化。

確定性效能流程圖

資源

相關文件：

- 使用 網路產品 AWS
- 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 建置混合網路連線 AWS
- AWS re：Invent 2023 – 安全遠端連線至 AWS
- AWS re：Invent 2022 – 使用 Amazon 最佳化效能 CloudFront
- AWS re：Invent 2019 – 與 AWS 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 和可擴展安全解決方案
- AWS 網路研討會

PERF04-BP04 使用負載平衡將流量分散到多個資源

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未考慮工作負載需求。
- 您不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不同的負載，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曝險等級：高

實作指引

負載平衡器會做為您的工作負載的進入點，從那裡將您的流量分散到後端目標，例如運算執行個體或容器，以提高利用率。

選擇正確的負載平衡器類型是最佳化架構的第一步。首先列出您的工作負載特性，例如通訊協定（例如 TCP、HTTP、 或 WebSockets）TLS、目標類型（例如執行個體、容器或無伺服器）、應用程式需求（例如長時間執行的連線、使用者身分驗證或黏性），以及置放（例如區域、本機區域、Outpost 或區域隔離）。

AWS 提供多個模型，讓您的應用程式使用負載平衡。Application Load Balancer 最適合 HTTP和 HTTPS 流量的負載平衡，並提供進階請求路由，以交付現代應用程式架構為目標，包括微服務和容器。

Network Load Balancer 最適合需要極端效能的TCP流量負載平衡。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，可讓您靈活地集中管理負載平衡器SSL的設定，並從工作負載卸載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，這是避免目標完成 CPU密集型TLS交握，以及改善憑證管理的機會。

當您在負載平衡器中設定SSL/TLS卸載時，它會負責加密來自用戶端的流量，同時將未加密的流量交付至後端，釋放後端資源並改善用戶端的回應時間。

Application Load Balancer 也可以提供 HTTP/2 流量，而不需要在目標上支援。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，應該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，您可能會決定讓工作負載更靠近您的客戶，也就是利用 AWS Local Zones 或甚至 AWS Outposts 中的 Application Load Balancer。

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大小有關。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。負載平衡器也可以與 Amazon ECS 和 Amazon EKS 整合，用於容器化工作負載。

- Amazon ECS - 服務負載平衡
- Amazon 上的應用程式負載平衡 EKS
- Amazon 上的網路負載平衡 EKS

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 上執行的非HTTP工作負載使用 Network Load BalancerUDP。
  - 如果您想要利用兩個 產品的功能，請使用兩者的組合ALB （ 作為 的目標NLB ）。例如，如果您想要將 靜態 與來自 IPs的HTTP標頭型路由NLB搭配使用ALB，或者如果您想要將HTTP工作負載公開至 ，則可以這樣做AWS PrivateLink。
  - 如需負載平衡器的完整比較，請參閱ELB產品比較 。
- 如果可能，請使用 SSL/TLS 卸載。

  - 使用與 整合Application Load Balancer和網路Load Balancer來設定 HTTPS/TLS 接聽程式AWS Certificate Manager。
  - 請注意，某些工作負載可能因為合規原因而需要 end-to-end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SEC傳輸中的 09-BP02 強制執行加密。
- 選取正確的路由演算法 （僅限 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 提供兩種路由演算法的選項：
  - 最低未解決請求：針對應用程式的請求因複雜性而異，或目標因處理功能而異的情況，用來讓負載更妥善地分散到您的後端目標。
  - 循環配置：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)。依預設，它會在 中關閉NLB，而在中ALB，您可以關閉每個目標群組 。
  - 使用跨區域開啟來增加可用性和彈性。根據預設， 的跨區域會開啟，ALB您可以在 中NLB為每個目標群組 開啟。
- 開啟HTTP工作負載的 HTTP keep-alives （僅限 ALB）。透過此功能，負載平衡器可以重複使用後端連線，直到保持連線逾時到期為止，藉此改善您的HTTP請求和回應時間，並減少後端目標的資源使用率。如需如何為 Apache 和 Nginx 執行此操作的詳細資訊，請參閱使用 Apache 或NGINX作為後端伺服器的最佳設定為何ELB？
- 開啟負載平衡器的監控功能。
  - 開啟 Application Load Balancer 和 Network Load Balancer 的存取記錄。
  - 要考慮的主要欄位ALB為 `request_processing_timerequest_processing_time`、 和 `response_processing_time`。
  - 要考慮的主要欄位NLB為 `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 。
  - 為效能相關指標建立警示，例如`TargetResponseTime`針對 ALB。

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 Application Load Balancer
- 監控 Network Load Balancer
- 使用 Elastic Load Balancing 在 Auto Scaling 群組的執行個體中分配流量

相關影片：

- AWS re：Invent 2023：聯網可以為您的應用程式做什麼？
- AWS re：Inforce 20：如何使用 Elastic Load Balancing 大規模增強您的安全狀態
- AWS re：Invent 2018：Elastic Load Balancing：Deep Dive 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：為不同工作負載充分利用 Elastic Load Balancing

相關範例：

- Gateway Load Balancer
- CDK 和 AWS CloudFormation 範例，用於使用 Amazon Athena 進行日誌分析

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通訊協定 （TCP），較高的延遲很可能降低整體輸送量。有方法可透過TCP調校和最佳化傳輸通訊協定來修正此問題，但其中一個解決方案是使用使用者資料包通訊協定 （UDP）。

常見的反模式：

- 無論效能需求為何，您都會TCP在所有工作負載中使用 。

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允許高速，但它不提供重新傳輸或高可靠性。TCP 是全功能通訊協定，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

如果您有能力為應用程式選擇不同的通訊協定，而且您具備此領域的專業知識，請使用不同的通訊協定來最佳化應用程式和使用者體驗。請注意，這種方法有很大的困難，如果您已先用其他方法最佳化應用程式，才可嘗試。

改善您的工作負載效能的主要考慮是了解延遲和輸送量需求，然後選擇可最佳化效能的網路通訊協定。

何時考慮使用 TCP

TCP 提供可靠的資料交付，可用於工作負載元件之間的通訊，其中資料的可靠性和保證交付至關重要。許多 Web 型應用程式依賴 TCP型通訊協定，例如 HTTP和 HTTPS來開啟通訊TCP端，以便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也使用 的常見應用程式TCP，因為它是應用程式元件之間簡單且可靠的傳輸機制。TLS 搭配 使用 TCP可以為通訊增加一些額外負荷，這可能會導致延遲增加和輸送量減少，但它具有安全性的優勢。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。

何時考慮使用 UDP

UDP 是 connection-less-oriented通訊協定，因此適用於需要快速、高效傳輸的應用程式，例如日誌、監控和 VoIP 資料。此外，UDP如果您的工作負載元件可回應來自大量用戶端的小型查詢，請考慮使用 ，以確保工作負載的最佳效能。Datagram Transport Layer Security （DTLS） UDP等同於 Transport Layer Security （TLS）。DTLS 搭配 使用 時UDP，額外負荷來自加密和解密資料，因為交握程序已簡化。DTLS 也會將少量額外負荷新增至UDP封包，因為其中包含其他欄位，以指示安全參數並偵測竄改。

何時考慮使用 SRD

可擴展的可靠資料包 （SRD） 是一種針對高輸送量工作負載進行最佳化的網路傳輸通訊協定，因為它能夠跨多個路徑載入平衡器流量，並從封包捨棄或連結失敗中快速復原。SRD 因此，最適合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

- 使用 AWS Global Accelerator 和 AWS Transfer Family 服務來改善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 上的工作負載之間達到較低的延遲 AWS。透過 AWS Transfer Family，您可以使用 Secure Shell File Transfer Protocol （ SFTP ） 和 File Transfer Protocol over SSL （ FTPS ） 等 TCP型通訊協定，安全地擴展和管理檔案傳輸至 AWS 儲存服務。
- 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。如果用戶端應用程式與伺服器之間的網路延遲很高，則TCP三向交握可能需要一些時間，進而影響應用程式的回應能力。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，請考慮使用 Amazon CloudFront，這會為動態內容建立每個原始伺服器的持續性連線，以移除可能拖慢每個用戶端請求的連線設定時間。
- 由於加密和解密的影響，TLS搭配 TCP或 使用 UDP 可能會導致工作負載的延遲增加和輸送量降低。對於此類工作負載，請考慮 SSL/TLS 卸載 onElastic Load Balancing，透過允許負載平衡器處理 SSL/TLS 加密和解密程序來改善工作負載效能，而不是讓後端執行個體這樣做。這有助於減少後端執行個體的使用CPU率，進而改善效能並增加容量。
- 使用Network Load Balancer （ NLB ） 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，以提高工作負載的效能和可靠性。會將傳入UDP流量NLB分散到多個目標，讓您水平擴展工作負載、增加容量並減少單一目標的負荷。
- 對於高效能運算 （ HPC ） 工作負載，請考慮使用彈性網路轉接器 （ ENA ） Express 功能，該功能使用 SRD 通訊協定，透過為EC2執行個體之間的網路流量提供更高的單一流量頻寬 （ 25 Gbps ） 和更低的尾部延遲 （ 99.9 百分位數 ） 來改善網路效能。
- 使用Application Load Balancer （ ALB ） 在工作負載元件之間或 gRPC 用戶端和服務之間路由和負載平衡 gRPC （ 遠端程序呼叫 ） 流量。gRPC 使用 TCP型 HTTP/2 通訊協定進行傳輸，並提供效能優勢，例如更輕量的網路佔用、壓縮、高效的二進位序列化、對多種語言的支援，以及雙向串流。

資源

相關文件：

- 如何將UDP流量路由至 Kubernetes
- Application Load Balancer
- EC2 Linux 上的增強型網路
- EC2 Windows 上的增強型聯網
- EC2 置放群組

- 在 Linux 執行個體上使用彈性網路轉接器 （ENA） 啟用增強型網路
- Network Load Balancer
- 使用 網路產品 AWS
- 轉換到 Amazon Route 53 中的以延遲為基礎的路由
- VPC 端點

相關影片：

- AWS re：Invent 2022 – 在新一代 Amazon Elastic Compute Cloud 執行個體上擴展網路效能
- 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 AWS 私有全域網路，您可以減少延遲，並為遠端使用者提供更好的體驗。

未建立此最佳實務時的風險暴露等級：中

實作指引

資源，例如 Amazon EC2執行個體，會放置在 AWS 區域、 AWS Local Zones 、 AWS Outposts或 AWS Wavelength 區域中的可用區域中。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 等邊緣服務AWS Global Accelerator也可用於透過快取邊緣位置的內容，或透過 AWS 全球網路為使用者提供工作負載的最佳路徑來改善網路效能。

Amazon EC2提供置放群組以進行聯網。置放群組是執行個體的邏輯分組，用於減少延遲。搭配支援的執行個體類型和彈性網路轉接器（ENA）使用置放群組，可讓工作負載參與低延遲、減少抖動的 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。

延遲敏感服務會使用 AWS 全球網路在邊緣位置提供，例如 Amazon CloudFront。這些邊緣位置通常提供內容交付網路（CDN）和網域名稱系統（ ）等服務DNS。透過在邊緣使用這些服務，工作負載可以對內容或DNS解決方案的請求做出低延遲的回應。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。正確設定 DNS和 HTTP/HTTPS 的快取控制，以從這些方法中獲益最多。

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流程日誌記錄 IP 流量
  - 如何保留用戶端 IP 地址 AWS Global Accelerator
- 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：
  - 資料所在位置：對於資料密集型應用程式 (例如大數據和機器學習)，應用程式碼執行時應盡可能接近資料。
  - 使用者所在位置：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：考慮諸如成本和合規性之類的限制，如為工作負載選取區域時應考慮的事項中所述。
- 使用 AWS Local Zones 執行諸如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。
- 5G 裝置需要 ultra-low-latency高解析度即時影片串流、高擬真度音訊和擴增實境或虛擬實境（AR/VR）等應用程式。對於此類應用程式，請考慮在 5G 網路內AWS Wavelength AWS Wavelength 內嵌 AWS 運算和儲存服務，提供用於開發、部署和擴展 ultra-low-latency應用程式的行動邊緣運算基礎設施。
- 針對常用資產，使用本機快取或 AWS 快取解決方案以提升效能、減少資料移動以及降低環境影響。

服務	使用情況
Amazon CloudFront	使用 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：

服務	使用情況
Lambda@Edge	用於在物件未經快取時起始的大量運算作業。
Amazon CloudFront Functions	用於可由短期函數啟動的簡單使用案例，例如 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。這些應用程式可以受益於提供靜態廣播 IP 地址和在邊緣位置TCP終止的網路服務。 AWS Global Accelerator 可以為您的應用程式提升效能達 60%，並為多區域架構提供快速容錯移轉。AWS Global Accelerator 為您提供靜態廣播 IP 地址，作為託管於一或多個 的應用程式固定進入點 AWS 區域。這些 IP 地址允許流量盡可能接近您的使用者傳入 AWS 全域網路。透過在用戶端與最接近用戶端的 AWS 邊緣位置之間建立TCP連線， AWS Global Accelerator 以減少初始連線設定時間。檢閱 的使用 AWS Global Accelerator ，以改善 TCP/UDP 工作負載的效能，並為多區域架構提供快速容錯移轉。

資源

相關的最佳實務：

- COST07-BP02 根據成本實作區域
- COST08-BP03 實作 服務以降低資料傳輸成本

- REL10-BP01 將工作負載部署到多個位置
- REL10-BP02 為您的多位置部署選取適當的位置
- SUS01-BP01 根據業務需求和永續性目標選擇區域
- SUS02-BP04 根據其聯網需求最佳化工作負載的地理定位
- SUS04-BP07 將網路之間的資料移動降至最低

相關文件：

- AWS 全球基礎設施
- AWS 本機區域 和 AWS Outposts，為您的邊緣工作負載選擇正確的技術
- 置放群組
- AWS 本機區域
- 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 解說器影片
- AWS Outposts：概觀和運作方式
- AWS re：Invent 2023 - 邊緣和內部部署工作負載的遷移策略
- AWS re：Invent 2021 - AWS Outposts：將 AWS 體驗帶到內部部署
- AWS re：Invent 2020： AWS Wavelength：在 5G 邊緣以超低延遲執行應用程式
- AWS re：Invent 2022 - AWS Local Zones：為分散式邊緣建置應用程式
- AWS re：Invent 2021 - 使用 Amazon 建置低延遲網站 CloudFront
- AWS re：Invent 2022 - 透過 改善效能和可用性 AWS Global Accelerator
- AWS re：Invent 2022 - 使用 建置您的全球廣域網路 AWS
- AWS re：Invent 2020：使用 Amazon Route 53 進行全域流量管理

相關範例：

- AWS Global Accelerator 自訂路由研討會
- 使用邊緣函數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優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。
- 將預設組態用於所有網路服務。
- 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：收集必要的 AWS 網路指標並實作網路監控工具，可讓您了解網路效能並最佳化網路組態。

未建立此最佳實務時的曝險等級：低

實作指引

監控往返 VPCs、子網路或網路介面的流量，對於了解如何利用 AWS 網路資源和最佳化網路組態至關重要。使用下列 AWS 網路工具，您可以進一步檢查流量使用量、網路存取和日誌的相關資訊。

實作步驟

- 識別關鍵效能指標，例如要收集的延遲或封包遺失。 AWS 提供數種工具，可協助您收集這些指標。藉由使用下列工具，您可以進一步檢查流量用量、網路存取和日誌的相關資訊：

AWS 工具	在哪裡使用
Amazon VPC IP Address Manager 。	使用 IPAM 來規劃、追蹤和監控您 AWS 和內部部署工作負載的 IP 地址。這是最佳化 IP 位址用量和分配的最佳實務。
VPC 流程日誌	使用VPC流程日誌來擷取 中往返網路介面的流量詳細資訊VPCs。透過VPC流量日誌，您可以

AWS 工具	在哪裡使用
	診斷過度限制或寬鬆的安全群組規則，並判斷往返網路介面的流量方向。
AWS Transit Gateway Flow Logs	使用 AWS Transit Gateway 流程日誌來擷取進出傳輸閘道的 IP 流量相關資訊。
DNS 查詢記錄	Route 53 收到的公有或私有DNS查詢的日誌資訊。使用 DNS 日誌，您可以透過了解請求的網域或子網域，或回應DNS查詢的 Route 53 EDGE位置來最佳化DNS組態。
Reachability Analyzer	Reachability Analyzer 可幫助您分析和偵錯網路可達性。Reachability Analyzer 是一種組態分析工具，可讓您在 中的來源資源和目的地資源之間執行連線測試VPCs。此工具可協助您確認您的組態符合您預期的連線能力。
網路存取分析器	網路存取分析器可協助您了解對資源的網路存取。您可以使用網路存取分析器來指定您的網路存取需求，並識別未符合您的指定需求的潛在網路路徑。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 並開啟網路選項的適當指標。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 Network Address Usage、VPCNATGateway AWS Transit Gateway、VPNTunle、 AWS Network Firewall Elastic Load Balancing 和 的指標 AWS Direct Connect。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，並且可以協助您根據您的觀察來最佳化網路組態。

AWS 工具	在哪裡使用
AWS Network Manager	使用 AWS Network Manager，您可以監控AWS 全球網路的即時和歷史效能，以用於操作和規劃。Network Manager 會在 AWS 區域 和 可用區域之間以及每個可用區域內提供彙總網路延遲，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，為您提供洞見，協助您識別、了解和改善使用者體驗。

- 使用 和 AWS Transit Gateway Flow Logs 識別熱門發言者VPC和應用程式流量模式。
- 評估和最佳化您目前的網路架構VPCs，包括 、子網路和路由。例如，您可以評估不同的VPC互連方式，或 AWS Transit Gateway 協助您改善架構中的聯網。
- 評估網路中的路由路徑，以確認始終使用目的地之間的最短路徑。網路存取分析器可協助您執行此操作。

資源

相關文件：

- 公開DNS查詢記錄
- 什麼是 IPAM？
- 什麼是 Reachability Analyzer？
- 什麼是網路存取分析器？
- CloudWatch 指標 VPCs
- 使用 Apache Parquet 格式的 VPC Flow Logs 來最佳化效能並降低網路分析的成本
- 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 – 雲端聯網開發人員指南
- AWS re：Invent 2023 – 為下一步做好準備？ 設計網路實現增長和靈活性

- AWS re：Invent 2023 – 進階VPC設計和新功能
- AWS re：Invent 2022 – 深入探索 AWS 網路基礎設施
- AWS re：Invent 2020 – 使用 AWS Well-Architected Framework 建立網路最佳實務和秘訣
- AWS re：Invent 2020 – 監控和疑難排解網路流量

相關範例：

- AWS 聯網研討會
- AWS 網路監控
- 在 上觀察並診斷您的網路 AWS
- 在 上尋找和解決網路設定錯誤 AWS

程序和文化

問題

- 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？

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。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：

最佳實務

- PERF05-BP01 建立關鍵效能指標 （KPIs），以測量工作負載運作狀態和效能
- PERF05-BP02 使用監控解決方案來了解效能最關鍵的領域
- PERF05-BP03 定義改善工作負載效能的程序
- PERF05-BP04 Load 測試工作負載
- PERF05-BP05 使用自動化主動修復效能相關問題
- PERF05-BP06 保留工作負載和服務 up-to-date
- PERF05-BP07 定期檢閱指標

PERF05-BP01 建立關鍵效能指標 （KPIs），以測量工作負載運作狀態和效能

識別KPIs以定量和定性方式測量工作負載效能的 。KPIs 協助您測量與業務目標相關的工作負載的運作狀態和效能。

常見的反模式：

- 您只能監控系統層級指標，以深入了解工作負載，而不了解這些指標的業務影響。
- 您假設KPIs您的 已作為標準指標資料發佈和共用。
- 您未定義量化、可測量的 KPI。
- 您KPIs不符合業務目標或策略。

建立此最佳實務的好處：識別代表工作負載運作狀態和效能的特定KPIs項目，有助於使團隊與其優先順序保持一致，並定義成功的業務成果。與所有部門共用這些指標可提供閾值、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曝險等級：高

實作指引

KPIs 允許業務和工程團隊在目標和策略的衡量上保持一致，以及這些因素如何結合以產生業務結果。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您的使用者，降低其使用者體驗等級，並可能導致客戶流失。當您定義KPI閾值時，請結合產業基準和最終使用者期望。例如，如果目前的產業基準是在兩秒內載入網頁，但最終使用者預期網頁會在一秒內載入，則在建立 時，您應該考慮這兩個資料點KPI。

您的團隊必須使用KPIs即時精細資料和歷史資料來評估工作負載，以供參考，並建立儀表板，在KPI資料上執行指標數學，以得出操作和使用率洞察。KPIs 應記錄並包含支援業務目標和策略的閾值，並應映射到監控的指標。KPIs 當業務目標、策略或最終使用者需求變更時，應重新檢視。

實作步驟

- 識別利益相關者：識別和記錄關鍵業務利益相關者，包括開發和運營團隊。
- 定義目標：與利益相關者合作，以定義和記錄工作負載的目標。考慮工作負載的關鍵效能層面，例如輸送量、回應時間和成本，以及業務目標，例如使用者滿意度。
- 檢閱產業最佳實務：檢閱產業最佳實務，以識別與您的工作負載目標KPIs相符的相關 。
- 識別指標：找出符合您工作負載目標的指標，可協助您衡量效能和業務目標。KPIs 根據這些指標建立 。範例指標是諸如平均回應時間或並發使用者數量等測量值。

- 定義和記錄 KPIs：使用產業最佳實務和工作負載目標來設定工作負載 的目標KPI。使用此資訊設定嚴重性或警示層級的KPI閾值。識別和記錄KPI未達到 的風險和影響。
- 實作監控：使用監控工具，例如 Amazon CloudWatch 或 AWS Config 來收集指標和測量 KPIs。
- 視覺化通訊 KPIs：使用 Amazon QuickSight 等儀表板工具視覺化並與KPIs利益相關者通訊。
- 分析和最佳化：定期檢閱和分析KPIs，以識別工作負載中需要改進的區域。與利益相關者合作以實作這些改進。
- 重新檢視和改進：定期檢閱指標KPIs並評估其有效性，特別是在業務目標或工作負載效能變更時。

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
- 大型雲端遷移的關鍵效能指標 （ KPIs ） 的重要性
- 如何使用 KPI Dashboard KPIs 追蹤您的成本最佳化
- X-Ray 文件
- 使用 Amazon CloudWatch 儀表板
- Amazon QuickSight KPIs

相關影片：

- AWS re：Invent 2023 - 最佳化成本和效能，並追蹤緩解進度
- AWS re：Invent 2023 - 使用 大規模管理資源生命週期事件 AWS Health
- AWS re：Invent 2023 - Pinterest 的效能和效率：最佳化最新執行個體
- AWS re：Invent 2022 - AWS optimization：立即結果的可行步驟
- AWS re：Invent 2023 - 建立有效的可觀測性策略
- AWS Summit SF 2022 - 全堆疊可觀測性和應用程式監控 AWS
- AWS re：Invent 2023 - AWS 為前 1，000 萬使用者擴展
- AWS re：Invent 2022 - Amazon 如何使用更好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 事件

相關範例：

- 使用 Amazon 建立儀表板 QuickSight

PERF05-BP02 使用監控解決方案來了解效能最關鍵的領域

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CPU使用率或記憶體壓力等標準運算指標足以解決效能問題。
- 您只會使用所選監控軟體記錄的預設指標。
- 您只會在有問題時審查指標。

建立此最佳實務的優點：了解效能的關鍵領域有助於工作負載擁有者監控KPIs和排定高影響改善的優先順序。

未建立此最佳實務時的曝險等級：高

實作指引

設定 end-to-end追蹤以識別流量模式、延遲和關鍵效能區域。監控您的資料存取模式，以確定是否有緩慢的查詢或分段及分割不佳的資料。使用負載測試或監控來找出工作負載受限領域。

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。調查這些領域後，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

- 設定 end-to-end監控以擷取所有工作負載元件和指標。以下是 上監控解決方案的範例 AWS。

服務	在哪裡使用
Amazon CloudWatch Real-User Monitoring（RUM）	擷取來自實際使用者用戶端和前端工作階段的應用程式效能指標。

服務	在哪裡使用
AWS X-Ray	透過應用程式層追蹤流量，並找出組成部分和相依性之間的延遲。使用 X-Ray 服務地圖，查看工作負載組成部分之間的關係和延遲。
Amazon Relational Database Service 效能洞見	檢視資料庫效能指標並找出效能待改善之處。
Amazon RDS增強型監控	檢視資料庫 OS 效能指標。
Amazon DevOpsGuru	偵測異常作業模式，以便在營運問題影響客戶之前識別。

- 執行測試，來產生指標、確定流量模式、瓶頸和關鍵效能區域。以下是如何進行測試的一些範例：
  - 設定CloudWatch合成 Canary，以程式設計方式使用 Linux cron 任務或速率表達式模擬瀏覽器型使用者活動，以隨時間產生一致的指標。
  - 使用 AWS 分散式負載測試解決方案，來產生尖峰流量或以預期成長速率測試工作負載。
- 評估指標和遙測，來找出關鍵的效能領域。與您的團隊一起檢閱這些領域，討論監控和解決方案，以避免瓶頸。
- 進行效能改善的實驗，並透過資料來衡量這些變更。例如，您可以使用CloudWatch Evidently 來測試對工作負載的新改進和效能影響。

資源

相關文件：

- re：Invent 2023 可 AWS 觀測性的新功能
- Amazon 建置者資料中心
- X-Ray 文件
- Amazon CloudWatch RUM
- Amazon DevOpsGuru

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 - 建立有效的可觀測性策略
- AWS Summit SF 2022 - 全堆疊可觀測性和應用程式監控 AWS
- AWS re：Invent 2022 - AWS optimization：立即結果的可行步驟
- AWS re：Invent 2022 - Amazon Builders 程式庫：Amazon 卓越營運 25 年
- AWS re：Invent 2022 - Amazon 如何使用更好的指標來改善網站效能
- 使用 Amazon CloudWatch Synthetics 視覺化監控應用程式

相關範例：

- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間
- Amazon CloudWatch RUM Web Client
- X-Ray SDK for Python
- 上的分散式負載測試 AWS

PERF05-BP03 定義改善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。

建立此最佳實務的優勢：定義進行架構變更的程序後，您就能使用收集的資料，以隨著時間影響工作負載。

未建立此最佳實務時的曝險等級：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。

識別工作負載的關鍵效能限制。記錄工作負載的效能限制，讓您知道哪些類型的創新可能會改善工作負載的效能。

實作步驟

- 識別 KPIs：識別 中KPIs概述的工作負載效能PERF05-BP01 建立關鍵效能指標 （KPIs），以測量工作負載運作狀態和效能，以基準化工作負載。
- 實作監控：使用AWS 可觀測性工具收集效能指標並測量 KPIs。
- 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的程式碼)，步驟請參閱 PERF05-BP02 使用監控解決方案來了解效能最關鍵的領域。使用分析和效能工具，來確定效能改進策略。
- 驗證改進：使用沙盒或生產前環境，來驗證改進策略的有效性。
- 實作變更：實作生產中的變更，並持續監控工作負載的效能。記錄改進項目並與利益相關者溝通這些變更。
- 重新檢視和完善：定期檢視您的績效改善程序，以找出需要提高的領域。

資源

相關文件：

- AWS 部落格
- 新功能 AWS
- AWS 技能建置器

相關影片：

- AWS re：Invent 2022 - 提供永續、高效能的架構
- AWS re：Invent 2023 - 最佳化成本和效能，並追蹤緩解進度
- AWS re：Invent 2022 - AWS optimization：立即結果的可行步驟
- AWS re：Invent 2022 - 透過最佳實務指引最佳化 AWS 工作負載

相關範例：

- AWS Github

PERF05-BP04 Load 測試工作負載

對工作負載執行負載測試，以確認它可以處理生產負載並識別任何效能瓶頸。

常見的反模式：

- 可以對工作負載的個別部分進行負載測試，而非整個工作負載。
- 可以在與生產環境不同的基礎設施中進行負載測試。
- 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。
- 您可以在未諮詢 Amazon 測試政策的情況下執行負載EC2測試，並提交模擬事件提交表單。這會導致您的測試無法執行，因為它看起來像事件 denial-of-service。

建立此最佳實務的優勢：在負載測試過程中測量效能時，會顯示您將在負載增加到何種程度時受到影響。這可讓您能夠在工作負載受到影響之前預測所需的變更。

未建立此最佳實務時的曝險等級：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，以及分析指標以評估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。自動執行負載測試作為交付管道的一部分，並將結果與預先定義的KPIs閾值進行比較。此程序有助於您持續達到所需的效能。

實作步驟

- 定義測試目標：確定您要評估的工作負載效能層面，例如輸送量和回應時間。
- 選擇測試工具：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：根據生產環境設定測試環境。您可以使用 AWS 服務來執行生產規模環境，以測試您的架構。
- 實作監控：使用 Amazon CloudWatch 等監控工具，收集架構中各項資源的指標。也可以收集和發布自訂指標。
- 定義方案：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：大規模執行測試方案。利用 AWS 雲端 來測試工作負載，以探索其無法擴展的位置，或它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。
- 記錄和分享調查結果：記錄並報告調查結果和建議。與利益相關者分享此資訊，協助他們做出有關效能最佳化策略的明智決策。

- 不斷反覆執行：負載測試應定期執行，尤其是在系統更新變更之後。

資源

相關文件：

- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- 上的分散式負載測試 AWS

相關影片：

- AWS Summit ANZ 2023：透過 AWS 分散式負載測試，放心加速
- AWS re：Invent 2022 - AWS 為前 1，000 萬使用者擴展
- 使用 AWS 解決方案解決：分散式負載測試

- AWS re：Invent 2021 - 透過使用 Amazon 的終端使用者洞察最佳化應用程式 CloudWatch RUM
- Amazon CloudWatch Synthetics 示範

相關範例：

- 上的分散式負載測試 AWS

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 （KPIs） 結合監控和警示系統，主動解決與績效相關的問題。

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。

建立此最佳實務的優勢：主動修復警示動作能夠讓支援人員專注在無法自動採取行動的項目上。這有助於操作人員無須疲於處理所有警示，而僅專注於關鍵警示。

未建立此最佳實務時的曝險等級：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，您可能有一個系統，可以預測預期的金鑰效能指標（KPI）值，並在違反特定閾值時發出警示，或者如果超出KPIs預期值，則工具可以自動停止或復原部署。

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 AWS 監控解決方案，例如 Amazon CloudWatch 或 AWS X-Ray 來協助您進一步了解問題的根本原因。
- 定義自動化程序：建立 step-by-step可用於自動修正問題的修復程序。
- 設定啟動事件：將事件設定為自動啟動修復程序。例如，您可以定義觸發條件，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 服務和技術來自動化修復程序。例如，AWS Systems Manager Automation 提供安全且可擴展的方式，來自動化修復程序。如果變更無法成功解決問題，則請務必使用自我修復邏輯來還原變更。
- 測試工作流程：在生產前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。
- 制定說明手冊：制定並記錄說明手冊，其中概述了補救計畫的步驟，包括啟動事件、補救邏輯和採取的動作。確保培訓利益相關者，以協助他們有效地應對自動補救事件。
- 審查和完善：定期評估自動補救工作流程的有效性。如有必要，請調整啟動事件和補救邏輯。

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
- 在 中使用警示和警示動作 CloudWatch
- 建立卓越營運雲端自動化實務：最佳實務來自 AWS Managed Services
- 使用自動表格優化來自動調整您的 Amazon Redshift 效能

相關影片：

- AWS re：Invent 2023 - 自動化擴展、修復和智慧自我修復的策略
- 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 - 智慧自動化雲端操作
- AWS re：Invent 2022 - 在 AWS 環境中大規模設定控制項
- AWS re：Invent 2022 - 使用 自動化修補程式管理和合規 AWS
- AWS re：Invent 2022 - Amazon 如何使用更好的指標來改善網站效能
- AWS re：Invent 2023 - 卸載：診斷並解決 Amazon 的效能問題 RDS
- AWS re：Invent 2021 -{New Launch} 自動偵測並解決 Amazon DevOpsGuru 的問題
- AWS re：Invent 2023 - 集中您的操作

相關範例：

- CloudWatch 記錄自訂警示

PERF05-BP06 保留工作負載和服務 up-to-date

繼續使用 up-to-date新的雲端服務和功能，以採用有效率的功能、移除問題，並改善工作負載的整體效能效率。

常見的反模式：

- 假設您目前的架構為靜態，且不會隨著時間的推移而更新。
- 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點：透過建立程序以持續 up-to-date使用新服務和產品，您可以採用新功能和功能、解決問題並改善工作負載效能。

未建立此最佳實務時的曝險等級：低

實作指引

在新服務、設計模式和產品功能推出時，評估提升效能的方法。透過評估、內部討論或外部分析，確定哪些方法可以提高工作負載效能或效率。定義程序來評估與工作負載相關的更新、新功能和服務。例

如，建立使用新技術的概念證明或與內部小組協商。嘗試新的想法或服務時，執行效能測試以衡量其對工作負載效能的影響。

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。
- 識別更新來源：找出與工作負載組成部分相關的新聞和更新來源。例如，您可以訂閱符合您工作負載元件的產品的 AWS 部落格最新消息。您可以訂閱RSS摘要或管理您的電子郵件訂閱 。
- 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軟體政策所需的軟體和組態，以及哪些執行個體需要更新。
- 評估最新更新：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 AMIs、容器映像，以及與雲端應用程式相關的其他成品。
  - 可以使用 AWS Systems Manager Patch Manager 之類的工具來自動化系統更新流程，並使用 AWS Systems Manager Maintenance Windows 來排程活動。
- 記錄過程：記錄用於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。參考文件化的業務需求KPIs，並協助排定哪些更新將對業務產生正面影響的優先順序。

資源

相關文件：

- AWS 部落格
- 新功能 AWS
- 使用自動化映像建置器管道實作 up-to-dateEC2映像

相關影片：

- AWS re：Inforce 2022 - 使用 自動化修補程式管理和合規 AWS
- 所有事項修補程式： AWS Systems Manager | AWS Events

相關範例：

- 庫存和修補程式管理
- 一個可觀測性研討會

PERF05-BP07 定期檢閱指標

作為日常維護的一部分或對事件或事故的回應，審查收集了哪些指標。透過這些審查來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) 有助於識別、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。

建立此最佳實務的優勢：持續審查正在收集的指標，以確認指標正確識別、處理或防止問題發生。如果讓指標長時間持續處於警示狀態，指標也會變得過時。

未建立此最佳實務時的曝險等級：中

實作指引

不斷改進指標收集和監控。作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，從而可以防止事故發生或更快地解決將來的事故。

作為對事故或事件的回應的一部分，評估哪些指標有助於解決問題，哪些指標可以幫助解決問題但未被追蹤。使用此方法提高所收集指標的品質，從而可以防止事故發生或更快地解決將來的事故。

實作步驟

- 定義指標：定義與您的工作負載目標一致的關鍵效能指標以進行監控，包括回應時間和資源使用率等指標。
- 建立基準：設定各指標的基準和期望值。基準應提供參考點以識別偏差或異常。
- 設定規律：設定規律 (例如每週或每月一次) 以檢閱重要指標。
- 識別效能問題：每次審查期間都會評估趨勢，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。
- 識別修正動作：使用您的分析來識別修正動作。這可能包括參數調整、修正錯誤和擴展資源。

- 記錄調查結果：記錄您的調查結果，包括已識別的問題、根本原因和修正動作。
- 反覆執行並改善：持續評估並改善指標檢閱過程。使用從以前的審核中學到的經驗教訓，隨著時間的推移提升程序。

資源

相關文件：

- CloudWatch 文件
- 使用 CloudWatch 代理程式從 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 如何使用更好的指標來改善網站效能
- AWS re：Invent 2023 - 建立有效的可觀測性策略
- AWS Summit SF 2022 - 全堆疊可觀測性和應用程式監控 AWS
- AWS re：Invent 2023 - 卸載：診斷並解決 Amazon 的效能問題 RDS

相關範例：

- 使用 Amazon 建立儀表板 QuickSight
- 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。您可以在成本最佳化支柱白皮書中找到有關實作的規定性指引。

最佳實務領域

- 實作雲端財務管理

- 了解支出和用量
- 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？

實作 Cloud Financial Management 可協助組織在最佳化成本和用量時實現商業價值和財務成功，並在上擴展規模 AWS。

最佳實務

- COST01-BP01 建立成本最佳化的擁有權
- COST01-BP02 建立金融與技術之間的合作關係
- COST01-BP03 建立雲端預算和預測
- COST01-BP04 在組織流程中實作成本意識
- COST01-BP05 報告並通知成本最佳化
- COST01-BP06 主動監控成本
- COST01-BP07 up-to-date保留新的服務版本
- COST01-BP08 建立成本感知文化
- COST01-BP09 從成本最佳化量化商業價值

COST01-BP01 建立成本最佳化的擁有權

建立團隊（Cloud Business Office、Cloud Center of Excellence 或 FinOps 團隊），負責建立和維護整個組織的成本意識。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

這是 Cloud Business Office （CBO） 或 Cloud Center of Excellence （CCOE） 函數或團隊的推出，負責建立和維護雲端運算的成本意識文化。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、技術和組織利益相關者組成的新團隊。

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：透過 FinOps 團隊、雲端財務管理 （CFM） 團隊、雲端商業辦公室 （CBO） 或 Cloud Center of Excellence （CCoE） 等指定團隊，客戶可以設計和實作治理機制，並推動全公司最佳實務。
- 分散式：影響技術團隊，進行成本最佳化。
- 混合式：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 業務或企業支援計劃， Enterprise-On-Ramp且需要協助建置此團隊或函數，請透過您的 帳戶團隊聯絡您的 Cloud Financial Management （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 的個人或團隊CFM需要以下一組技能：
  - 軟體開發：如果正在建構指令碼和自動化。
  - 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透過測量、監控、修改、規劃和擴展雲端的有效使用，來有效率地在雲端上操作。

- 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。

  您可以定義值型或成本型金鑰效能指標 （ KPIs ）。當您定義 時KPIs，您可以根據效率和預期業務結果來計算預期成本。以價值為基礎的將成本和用量指標與商業價值驅動因素KPIs聯繫起來，並協助合理化 AWS 支出的變化。衍生以價值為基礎的第一步KPIs是跨組織合作，以選取並商定一組標準KPIs。

- 確立定期規律：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。這些定期節奏呼叫可協助團隊識別為整個組織KPIs提供意義的值。

資源

相關文件：

- AWS CCOE 部落格
- 建立雲端商業辦公室
- CCOE - Cloud Center of Excellence

相關影片：

- Vanguard CCOE成功案例

相關範例：

- 使用 Cloud Center of Excellence （ CCOE ） 轉換整個企業
- 建置 CCOE以轉換整個企業
- 建置時要避免的 7 個陷阱 CCOE

COST01-BP02 建立金融與技術之間的合作關係

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。

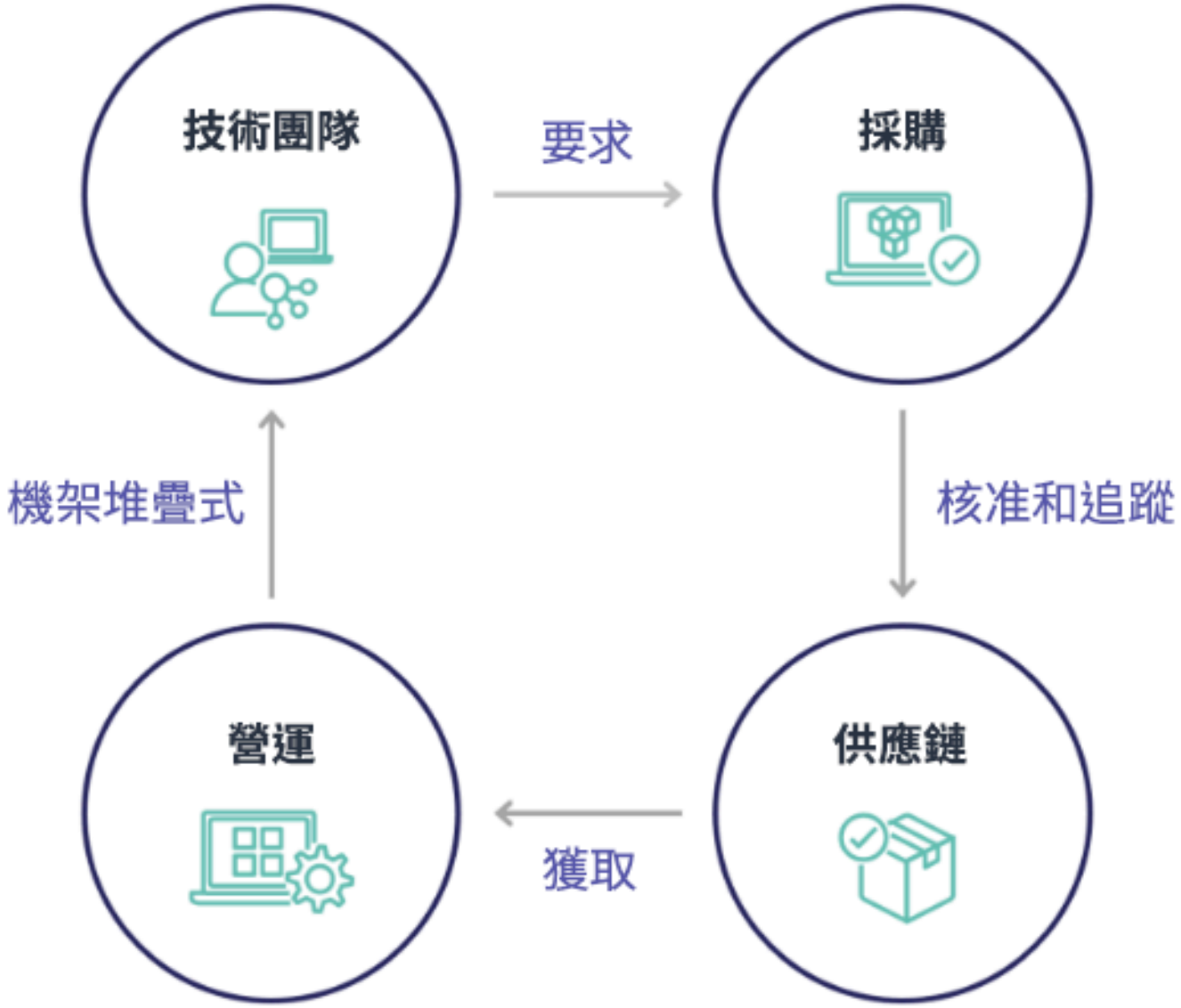
未建立此最佳實務時的曝險等級：高

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
- 營運團隊機架、堆疊和移交 ready-to-use基礎設施



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，所需的其實就是一個帳戶以及一組適當的權限。這也是降低 IT 和財務風險的原因；這表示團隊永遠只需按幾下滑鼠或API通話，即可終止閒置或不必要的雲端資源。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益相關者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：

- 財務領導：CFOs、財務控制者、財務規劃人員、業務分析師、採購、採購和應付帳款必須了解耗用、購買選項和每月發票開立程序的雲端模型。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、依用量計費定價、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。
- 財務利益相關者擔任策略顧問，以了解如何使用資本購買承諾折扣（例如預留執行個體或 AWS Savings Plans），以及如何使用雲端來擴展組織。
- 現有的應付帳款和採購程序會與雲端搭配使用。
- 財務和技術團隊會合作預測未來 AWS 成本和用量，以調整和建立組織預算。
- 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益相關者包括：

- 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對於建立成本感知文化至關重要，該文化鼓勵工程師對 Cloud Financial Management（）採取行動CFM。CFM 或財務營運從業人員和財務團隊的其中一個常見問題，就是讓工程師了解雲端上的整個業務、遵循最佳實務，並採取建議的動作。
- 第三方：如果您的組織使用第三方（例如顧問或工具），請確保他們與您的財務目標保持一致，並且可以透過他們的參與模型和投資回報（）來證明其一致性ROI。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。

實現CFM和成功需要跨財務、技術和業務團隊的協作，以及跨組織傳達和評估雲端支出的方式轉變。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。這通常是 CFOs、財務控制者、財務規劃人員、業務分析師、採購和採購。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、整體財務和最佳化指標。然後，會更詳細地報告關鍵工作負載。

資源

相關文件：

- AWS 新聞部落格

COST01-BP03 建立雲端預算和預測

調整現有的組織預算編列和預測程序，使其與本質會高度變動的雲端成本和用量相容。程序必須是動態的，並使用以趨勢為基礎和/或以業務驅動因素為基礎的演算法。

未建立此最佳實務時的曝險等級：高

實作指引

在傳統的內部部署 IT 設定中，客戶通常會面臨規劃固定成本的挑戰，這些成本只是偶爾變化，通常是購買新的 IT 硬體和服務以滿足尖峰需求。相反地， AWS 雲端 會採用不同的方法，客戶會依實際 IT 和業務需求來支付他們使用的資源。在雲端環境中，需求可能會每月、每天甚至每小時波動。

使用雲端可帶來了效率、速度和敏捷性，進而產生高度變化的成本和使用模式。成本會隨著工作負載效率的增加或部署新的工作負載和功能而降低或增加。隨著工作負載擴展以滿足不斷擴大的客戶群，由於資源的可存取性增加，雲端使用量和成本也隨之增加。雲端服務的這種靈活性延伸到成本和預測，這創造了一定程度的彈性。

與這些不斷變化的業務需求和需求驅動因素緊密保持一致至關重要，並盡可能準確地規劃。傳統的組織預算流程需要適應這種變化。

在預測新工作負載的成本時，請考慮成本建模。成本建模可建立對預期雲端成本的基準理解，這可協助您執行總擁有成本 （TCO）、投資報酬率 （ROI） 和其他財務分析、設定目標和期望，以及識別成本最佳化的機會。

您的組織應了解成本定義和接受的分組。您預測的詳細程度會根據組織的結構與內部工作流程而有所不同。選取符合您特定需求和組織設定的精細度。了解在什麼層級執行預測非常重要：

- 管理帳戶或 AWS Organizations 層級：管理帳戶是您用於建立 AWS Organizations的帳戶。組織預設有一個管理帳戶。

- 連結或成員帳戶： Organizations 中的帳戶是 AWS 帳戶 包含您的 AWS 資源和可存取這些資源之身分的標準。
- 環境：環境是執行應用程式版本的 AWS 資源集合。可以使用多個連結帳戶或成員帳戶建立一個環境。
- 專案：專案是指固定期間內要完成的既定目標或任務的組合。在預測期間考慮專案生命週期非常重要。
- AWS 服務：運算或儲存服務等群組或類別，您可以在其中將預測 AWS 的服務分組。
- 自訂分組：您可以根據組織的需求建立自訂群組，例如業務單位、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。

識別出會影響使用量成本的業務驅動因素，並分別預測每個因素，以預先計算預期使用量。部分驅動因素可能與組織內的 IT 和產品團隊相關。您的銷售、行銷和業務主管已經熟知行銷活動、促銷、地理擴張、合併與收購等其他業務驅動因素，進行協作並考慮所有這些需求驅動因素也很重要。

您可以根據過去的支出，在定義的未來時間範圍內使用AWS Cost Explorer 進行趨勢型預測。 AWS Cost Explorer的預測引擎會根據費用類型 （例如預留執行個體） 分割歷史資料，並使用機器學習和規則型模型的組合，個別預測所有費用類型的支出。

建立預測程序和建置模型後，您可以使用AWS Budgets 來指定時段、循環或金額 （固定或變數），並新增 服務和標籤等篩選條件 AWS 區域，以精細設定自訂預算。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，預測更加靈活，也可以全年隨時調整，並提供一年、兩年或三年的動態預測。在技術和商業利益相關者之間建立財務期望時，預算和預測至關重要。準確的預測和實作，不僅讓直接負責佈建成本的利益相關者更能掌握狀況，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，以定期向您和利益相關者傳送電子郵件。您還可以根據實際成本 (為主動式) 或預測成本建立AWS Budgets 提醒，從而為採取措施緩解潛在成本超支提供了時間。當您的成本或用量實際超出某個級別，或預計超出預算額度時，系統會提醒您。

使用基於趨勢的演算法 (使用歷史成本作為輸入) 和基於驅動因素的演算法 (例如：新產品推出、區域擴展或工作負載的新環境) 調整現有預算並預測流程，使其更具動態性，這是動態和可變支出環境的理想選擇。使用 Cost Explorer 或任何其他工具確定趨勢型預測後，請使用 ，根據預期用量 AWS Pricing Calculator（流量 requests-per-second或必要的 Amazon EC2執行個體） 來估算您的 AWS 使用案例和未來成本。

追蹤預測的準確度，因為可以根據這些預測計算和估計來設定預算。監控整合式雲端成本預測的準確性和有效性。定期檢查與預測相比的實際支出，並根據需要進行調整以提高預測精確度。追蹤預測差異，並對報告的差異執行根本原因分析，以採取行動並調整預測。

如 COST01-BP02 建立金融與技術之間的合作關係 中所述，重要的是在 IT、財務和其他利益相關者之間建立合作關係和規律，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，提高接觸頻率可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：在具有多個維度和分組的組織內建立通用 AWS 的成本語言。確保利益相關者了解預測精細度、定價模型和成本預測的水平。
- 分析基於趨勢的預測：使用基於趨勢的預測工具，例如 AWS Cost Explorer 和 Amazon Forecast。從服務、帳戶、標籤和成本類別等多個角度分析您的使用成本。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（將線性迴歸套用為預測的機器學習形式）。
- 分析基於驅動因素的預測：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，以預先計算預期使用成本。與業務單位主管和利益相關者密切合作，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流程：根據所採用的預測方法 (例如基於趨勢、基於業務驅動因素、或兩種預測方法的組合)，定義您的預測和預算流程。預算應經過計算、切合實際並基於您的預測。
- 設定警示和通知：使用 AWS Budgets 警示和成本異常偵測來取得警示和通知。
- 與利益相關者一起執行定期審查：例如，與 IT、財務、平台團隊和其他業務部門的利益相關者一起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 Forecasting
- Amazon Forecast
- AWS Budgets

相關影片：

- 如何使用 AWS Budgets 來追蹤我的支出和用量
- AWS 成本最佳化系列： AWS Budgets

相關範例：

- 了解並建置基於驅動程式的預測
- 如何建立和推動預測文化
- 如何改善雲端成本預測
- 使用適當的工具進行雲端成本預測

COST01-BP04 在組織流程中實作成本意識

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。向技術團隊和業務和財務團隊的決策者報告雲端成本，以提高成本意識，並為財務和業務利益相關者建立效率關鍵績效指標（KPIs）。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件管理程序，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。在考慮實作的成本時，請架構對話以包含投資報酬率（ROI）元件，以證明時間或金錢投資的合理性。
- 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。
- 利用免費的 AWS 原生工具AWS Budgets，例如 AWS Cost Anomaly Detection、 和 AWS Budgets 報告 。

當組織持續採用 Cloud Financial Management （CFM） 實務時，這些行為會陷入工作和決策中。其結果是文化更具成本意識，從建構新 born-in-the-cloud應用程式的開發人員，到分析這些新雲端投資ROI的金融經理。

實作步驟

- 識別相關的組織程序：每個組織單位均審查其程序，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，例如事件管理和培訓。
- 建立自我維持的成本感知文化：確保所有相關的利益相關者都與 cause-of-change成本保持一致並產生影響，以便他們了解雲端成本。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：每個程序都會經過修改，以提高成本感知。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。

若要取得協助，請透過您的帳戶團隊聯絡CFM專家，或探索下列資源和相關文件。

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
- 入門指南 AWS Cost Management

COST01-BP05 報告並通知成本最佳化

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。

使用 AWS Cost Explorer，透過多種篩選條件和精細度來檢視成本和用量時，這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。使用 AWS Budgets Reports，根據設定的預算追蹤成本使用和用量狀況時可使用。

使用 AWS Budgets設定自訂預算，以追蹤您的成本和用量，並在超過閾值時快速回應從電子郵件或 Amazon Simple Notification Service （ AmazonSNS ） 通知收到的提醒。將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，以隨時了解實際或預測的成本和用量如何朝預算閾值進展。也可根據這些提醒來設定提醒和動作以自動執行，或在超出預算目標時透過核准程序執行。

實作成本和用量的通知，以確保在成本和用量發生意外時可以快速採取行動。 AWS Cost Anomaly Detection可讓您減少成本意外並增強控制，而不會拖慢創新。 AWS Cost Anomaly Detection 識別異常支出和根本原因，這有助於降低帳單意外的風險。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。

您也可以將 Amazon QuickSight 與 AWS Cost and Usage Report （ CUR ） 資料搭配使用，以更精細的資料提供高度自訂的報告。Amazon QuickSight 可讓您排程報告並定期接收成本報告電子郵件，以取得歷史成本和用量或節省成本的機會。檢查我們在 Amazon 上建置的成本智慧儀表板 （ CID ） 解決方案 QuickSight，這可讓您獲得進階可見性。

使用 AWS Trusted Advisor提供指引，以驗證佈建的資源是否符合成本最佳化的 AWS 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。

定期建立報告，其中包含 的 Savings Plans 、預留執行個體和 Amazon EC2許可化建議的反白顯示，AWS Cost Explorer 以開始降低與穩定狀態工作負載、閒置和未充分利用資源相關聯的成本。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 最佳實務以減少浪費，或要求帳戶團隊和合作夥伴協助您最佳化並節省雲端成本。

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深的成本節省，並允許利益相關者 （ 企業擁有者、財務和技術團隊 ） 參與這些承諾討論。

共用包含機會或新發行公告的報告，這可協助您降低雲端的總擁有成本 （ TCO ）。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：針對工作負載在所有 帳戶 AWS Budgets 上設定 。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
- 開始最佳化 AWS 雲端成本的關鍵方法

COST01-BP06 主動監控成本

實作工具和儀表板以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加以推廣。

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。

建立每日或頻繁的例行程序來使用 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。使用分組和篩選來分析 AWS 帳戶層級、工作負載層級或特定 AWS 服務層級 AWS 的服務用量和成本，並驗證是否預期。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 Cost Intelligence Dashboard 建置自己的報告，這是由 AWS Solutions Architects 建置的 Amazon QuickSight 解決方案，並將您的預算與實際成本和用量進行比較。

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。
- 建立並啟用成本和用量AWS Budgets的每日精細程度，以及時採取行動，防止任何潛在的成本超支： AWS Budgets 允許您設定警示通知，以便在任何預算類型超出預先設定的閾值時隨時掌握最新資訊。最佳利用方式 AWS Budgets 是將預期的成本和用量設定為限制，以便將超出預算的任何項目視為超支。
- AWS Cost Anomaly Detection 為成本監控建立：AWS Cost Anomaly Detection使用進階Machine Learning技術來識別異常支出和根本原因，以便您可以快速採取行動。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 AWS Cost and Usage Report （ CUR ） 資料或將資料與 Amazon QuickSight 儀表板整合，以視覺化組織的成本： AWS Cost Explorer 具有可讓您視覺化、了解和管理成本 AWS 和用量的 easy-to-use介面。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
- AWS Cost Anomaly Detection 使用 Slack 提醒

COST01-BP07 up-to-date保留新的服務版本

定期諮詢專家或 AWS 合作夥伴，以考慮哪些服務和功能提供較低的成本。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 不斷增加新功能，因此您可以利用最新的技術更快地進行實驗和創新。您可以實作新的 AWS 服務和功能，以提高工作負載的成本效率。定期檢閱 AWS 成本管理、AWS 新聞部落格、AWS 成本管理部落格、AWS最新消息，以取得新的服務和功能版本的相關資訊。新文章提供發佈所有 AWS 服務、功能和區域擴展公告的簡短概觀。

實作步驟

- 訂閱部落格：前往 AWS 部落格頁面並訂閱 What's New Blog 和其他相關部落格。可以使用您的電子郵件地址在通訊偏好頁面註冊。
- 訂閱 AWS 新聞：定期檢閱AWS 新聞部落格和 What's New with AWS，以取得新服務和功能版本的相關資訊。訂閱RSS摘要，或使用電子郵件追蹤公告和版本。
- 遵循 AWS 折價：我們所有服務的一般降價，對於 從規模中獲得的客戶 AWS 來說，傳遞經濟效率是標準的方式。截至 2023 年 9 月 20 日，自 2006 年以來 AWS ，價格已降低 134 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。您可以在AWS 新聞部落格 的降價類別中了解先前的降價工作，包括 Amazon Elastic Compute Cloud （Amazon EC2）執行個體。

- AWS 事件和會議：參加您的本機 AWS 高峰會，以及與您所在區域的其他組織的任何本機會議。如果您無法親自出席，請嘗試參加虛擬活動，以聽取 AWS 專家和其他客戶商業案例的更多訊息。
- 與您的客戶團隊會面：與您的客戶團隊排定一個定期規律，與他們開會並討論產業趨勢和 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
- 新功能 AWS
- AWS 新聞部落格

相關範例：

- Amazon EC2 – 15 年的 IT 成本最佳化和節省
- AWS 新聞部落格 - 折價

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，或比較跨團隊標準化成本和用量的報告（例如 cost-per-workload和 cost-per-transaction）來完成。
- 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，且已經過利益相關者的確認，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：提高成本意識，並為KPIs財務和業務利益相關者建立效率。
- 通知利益相關者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：分享有關降低任何工作負載成本的成功案例 AWS 帳戶，或組織建立積極的態度並鼓勵成本最佳化。
- 訓練：確保技術團隊或團隊成員接受訓練，以認識 上的資源成本 AWS 雲端。
- AWS 事件和會議：參加當地 AWS 峰會，以及與您所在區域的其他組織的任何當地會議。
- 訂閱部落格：前往 AWS 部落格頁面並訂閱 What's New Blog 和其他相關部落格，以追蹤 共用的新版本、實作、範例和變更 AWS。

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 Labs：雲端財務管理

COST01-BP09 從成本最佳化量化商業價值

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益相關者解釋投資報酬率。量化商業價值有助於您在未來就成本優化投資獲得利益相關者更多的支持，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曝險等級：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，如果公司花費 100，000 美元在 上部署工作負載， AWS 然後對其進行最佳化，則新成本只會變成 80，000 美元，而不會犧牲品質或輸出。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。利益相關者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 時量化 Amazon Elastic Compute Cloud（Amazon EC2） 的成本節省，這可降低成本並維持工作負載輸出層級。您可以在移除閒置的 Amazon EC2執行個體，或刪除未連接的 Amazon Elastic Block Store （Amazon EBS） 磁碟區時，量化 AWS 支出的成本降低。

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

- 評估商業利益：這是分析和調整 AWS 雲端 成本的程序，以最大限度地提高從每一美元花費中獲得的利益。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 AWS 成本：預測有助於財務利益相關者與其他內部和外部組織利益相關者設定期望，並可以改善組織的財務可預測性。 AWS Cost Explorer可用於執行成本和用量的預測。

資源

相關文件：

- AWS 雲端 經濟效益
- 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 Windows 開啟時解除鎖定商業價值 AWS

相關範例：

- 測量並最大化 Customer 360 的商業價值
- 採用 Amazon Web Services 受管資料庫的商業價值
- Amazon Web Services 對於獨立軟體供應商的商業價值
- 雲端現代化的商業價值
- 遷移到 Amazon Web Services 的商業價值

了解支出和用量

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. 如何停用資源？

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。透過採用方法 checks-and-balances，您可以創新，而不會過度花費。

最佳實務

- COST02-BP01 根據您的組織需求制定政策
- COST02-BP02 實作目標
- 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
- COST02-BP06 追蹤專案生命週期

COST02-BP01 根據您的組織需求制定政策

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、修改和停用。

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，這是推動變革的關鍵，因為用量變化會帶來成本變化。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、修改和停用。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治理政策，其重點為成本優化。確實根據組織的要求和利益相關者的請求來調整政策。

- 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。
- 用途：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：明確定義應該使用此政策的人員及其使用時機，例如 DevOps X Team 將此政策用於 X 環境（生產或非生產）的美國東部客戶。

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. 排程 Amazon EC2和 Amazon RDS執行個體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 24 小時無活動後停止未使用的 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）在生產環境中檢閱其停止的 Amazon EC2執行個體，並通知他們，如果其 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，然後使用 CPU 調整執行個體的大小 AWS Compute Optimizer。
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. 監控資源使用率，並使用 Amazon 設定警示來觸發擴展事件 CloudWatch。
11. 對於每個 AWS 帳戶，請使用 來根據成本中心和業務單位設定帳戶 AWS Budgets 的成本和用量預算。
12. 使用 為您的帳戶 AWS Budgets 設定成本和用量預算，可協助您掌握支出並避免意外帳單，讓您更妥善地控制成本。

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 step-by-step執行政策要求的指示。

若要實作此政策，您可以使用各種第三方工具或 AWS Config 規則來檢查政策陳述式的合規性，並使用 AWS Lambda 函數觸發自動修復動作。您也可以使用 AWS Organizations 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

- 與利益相關者會面：若要制定政策，請要求組織內的利益相關者 (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。採取反復的方法，從廣泛討論開始，然後在每個步驟持續細化至最小的單位。團隊成員包括對工作負載有直接關係的人員，例如組織單位或應用程式擁有者，以及支援群組，例如安全和財務團隊。
- 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：定義工作負載運作的位置，包括國家和國家中的區域。此資訊用於對應至 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 受管政策作為指南。
- 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、營運和停用) 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：工作負載和組織需求可能會隨時間變更。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

- AWS 任務函數的受管政策
- AWS 多個帳戶計費策略
- AWS 服務的動作、資源和條件索引鍵
- AWS 管理和治理
- AWS 區域 使用IAM政策控制對 的存取
- 全球基礎設施區域和 AZs

相關影片：

- AWS 大規模管理和治理

相關範例：

- VMware - 什麼是雲端政策？

COST02-BP02 實作目標

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預期成果的方向，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曝險等級：高

實作指引

為您的組織制定成本與用量總目標和具體目標。作為 上不斷成長的組織 AWS，設定和追蹤成本最佳化的目標很重要。這些目標或關鍵效能指標（KPIs）可以包括支出百分比或採用某些最佳化服務，例如 AWS Graviton 執行個體或 gp3 EBS磁碟區類型。設定可衡量和可實現的總目標有助於衡量效率的改善情況，這對於業務營運非常重要。總目標可為您的組織提供預期結果的指引和方向。

具體目標是要實現的具體可衡量成果。簡而言之，目標就是您想要往的方向前進，而目標則是該方向的前進距離，以及應該實現該目標的時間（使用特定、可測量、可指派、實際和及時的指導，或 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。平台用量增加 20%，成本增加少於百分之五，則是具體目標範例。另一個常見的總目標是工作負載每六個月必須更有效率。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。使用正確的指標，並KPIs為您的組織進行計算。您可以從基本開始KPIs，並在稍後根據業務需求進行演變。

成本優化的總目標是提高工作負載效率，這對應於工作負載的每個業務成果的成本隨著時間而降低。為所有工作負載實作這個總目標，並設定具體目標，例如每六個月至一年將效率提高百分之五。在雲端中，可以透過建立成本最佳化功能以及發行新服務和功能來達成此目標。

具體目標是您希望達到以達到的可量化基準，以實現總體目標，而基準則會將您的實際結果與具體目標進行比較。使用 建立每個運算服務單位KPIs成本的基準（例如 Spot 採用、Graviton 採用、最新執行個體類型和隨需涵蓋範圍）、儲存服務（例如EBSGP3採用、淘汰EBS快照和 Amazon S3 標準儲存）或資料庫服務使用量（例如RDS開放原始碼引擎、Graviton 採用和隨需涵蓋範圍）。這些基準 和 KPIs 可協助您驗證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。每個組織可以具有這些 的不同目標值KPIs。

類別	KPI (%)	描述
運算	EC2 用量涵蓋範圍	EC2 使用 SP+RI+Spot 的執行個體（以成本或小時為單位），相較於執行個體的總數量（以成本或小時為單位）EC2
運算	計算 SP/RI 使用率	與總體可用的 SP 或 RI 小時數相比，已使用的 SP 或 RI 小時數
運算	EC2/小時成本	EC2 成本除以該小時內執行的 EC2執行個體數量
運算	vCPU 成本	所有執行個體的每個 vCPU 成本
運算	最新一代執行個體	Graviton (或其他新一代執行個體類型) 上的執行個體百分比
資料庫	RDS 涵蓋範圍	RDS 使用 RI 的執行個體（以成本或小時為單位），相較於RDS執行個體的總數量（以成本或小時為單位）

類別	KPI (%)	描述
資料庫	RDS 使用率	與總體可用的 RI 小時數相比，已使用的 RI 小時數
資料庫	RDS 運作時間	RDS 成本除以該小時內執行的 RDS執行個體數量
資料庫	最新一代執行個體	Graviton (或其他現代執行個體類型) 上的執行個體百分比
儲存	儲存使用率	最佳化儲存成本 (例如 Glacier、Deep Archive 或 Infrequent Access) 除以總儲存成本
標記	未標記資源	Cost Explorer： 1. 篩選掉抵用金、折扣、稅金、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您的每月成本。

使用此表格，包括目標或基準值，應根據組織目標計算這些值。您需要測量業務的特定指標，並了解該工作負載的業務結果，才能定義準確且實際的 KPIs。當您評估組織內的績效指標時，請區分服務於不同目的之不同類型的指標。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，它們可能會追蹤伺服器響應時間、網路延遲或系統正常運行時間。這些指標對於評估基礎設施如何支援組織的技術操作至關重要。但是，它們不能直接洞察更廣泛的業務目標，例如客戶滿意度，收入增長或市場份額。為了全面了解業務績效，請使用與業務成果直接相關的策略性業務指標來補充這些效率指標。

為您的 KPIs和相關節省機會建立近乎即時的可見性，並追蹤您的進度。若要開始定義和追蹤KPI目標，建議您從 Cloud Intelligence KPI Dashboards （ ） 中取得儀表板CID。 https://

wellarchitectedlabs.com/cloud-intelligence-dashboards/根據成本和用量報告（CUR）中的資料，KPI儀表板提供一系列建議的成本最佳化 KPIs，能夠設定自訂目標並追蹤一段時間內的進度。

如果您有其他解決方案來設定和追蹤KPI目標，請確定組織中所有雲端財務管理利益相關者都採用這些方法。

實作步驟

- 定義預期的用量等級：首先，請關注用量等級。與應用程式擁有者、行銷團隊和更大的業務團隊互動，以了解工作負載的預期用量等級。客戶需求如何隨著時間而變更，以及因季節性增加或行銷活動會發生哪些變更？
- 定義工作負載資源與成本：定義用量等級後，量化達成這些用量等級所需的工作負載資源變更。您可能需要為工作負載元件增加資源的大小或數量、增加資料傳輸，或將工作負載元件變更為特定等級的不同服務。指定每個要點的成本，並預測當用量發生變化時成本會有什麼變化。
- 定義業務總目標：從預期用量和成本變更中取得輸出，將此項目與預期的技術變更或任何您正在執行的計畫結合，並制定工作負載的總目標。總目標必須涵蓋用量和成本，以及兩者之間的關係。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，確保將未使用的資源控制在特定成本水位以下)。無須為每個未使用的資源類型定義總目標，也不需要為總目標和具體目標定義造成損失的成本。如果預期有成本變更但用量不變，請確認制定有組織計畫 (例如培訓和教育等能力打造計畫)。
- 定義具體目標：對於定義的每個總目標，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，具體目標將會量化改善的程度 (通常是所有經費所獲得的業務輸出)，及其達成時間。例如，可設定一個總目標，以盡量減少因過度佈建而造成的浪費。有了這個總目標後，您的具體目標可能是生產工作負載第一層中的運算過度佈建產生的浪費不應超過分層運算成本的 10%。此外，第二個具體目標可能是生產工作負載第二層中的運算過度佈建產生的浪費不應超過分層運算成本的 5%。

資源

相關文件：

- 工作職能的AWS 受控政策
- AWS 多帳戶帳單策略
- AWS 區域 使用IAM政策控制對 的存取
- S.M.A.R.T. 目標
- 如何使用 CID KPI Dashboard KPIs 追蹤您的成本最佳化

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？
- 選擇單位指標以支援您的業務
- 實務中的單位指標 — 經驗教訓
- 單位指標如何幫助在業務職能之間建立一致性
- Well-Architected 實驗室：停用資源 (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曝險等級：高

實作指引

AWS Organizations 可讓您建立多個 AWS 帳戶 ，協助您在 上擴展工作負載時集中管理環境 AWS。您可以分組組織單位 （OU） 結構 AWS 帳戶 ，並在每個 OU AWS 帳戶 下建立多個，以建立組織階層的模型。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，您可以按照管理帳戶最佳實務和成員帳戶最佳實務，根據設計的帳戶結構建立新的或 AWS 帳戶 選取現有帳戶作為成員帳戶。

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。對於 AWS 帳戶 您應該有多少個，沒有一個大小適合所有答案。評估您目前和未來的營運和成本模型，以確保 的結構 AWS 帳戶 反映組織的目標。有些公司 AWS 帳戶 會基於業務原因建立多個 ，例如：

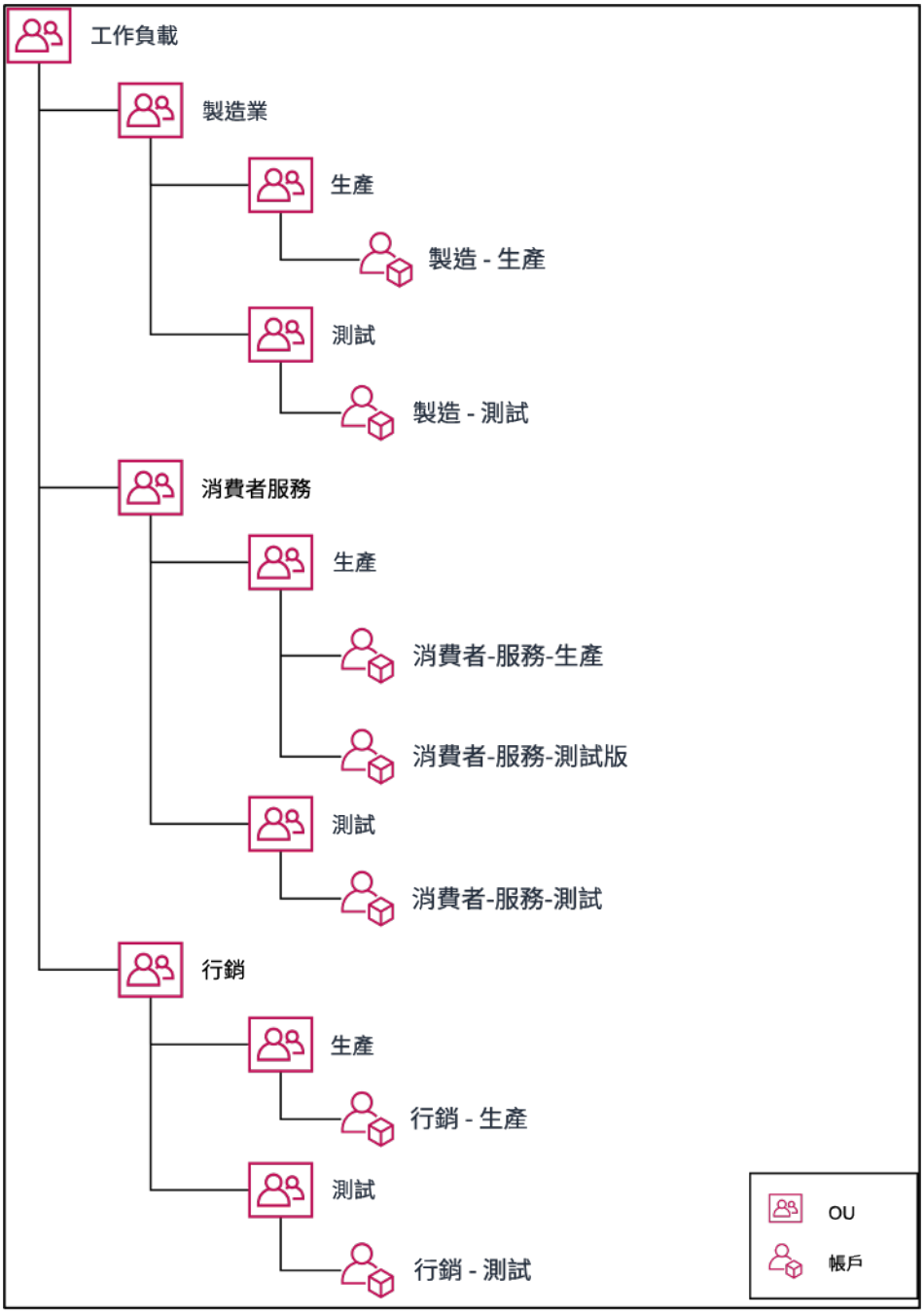
- 組織單位、成本中心或特定工作負載之間需要行政管理或會計年度和帳單上的區隔。
- AWS 服務限制設定為特定工作負載專用。
- 工作負載和資源之間需要區隔和隔離。

在 AWS Organizations 中，合併帳單會在一個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 AWS Organizations 搭配組織單位 （ OU ） 使用 來分組多個帳戶，並在每個 OU AWS 帳戶 下放置多個帳戶。建議OUs用於各種使用案例和工作負載，這些案例和工作負載提供組織帳戶的模式。



在組織單位 AWS 帳戶 下分組多個的範例。

AWS Control Tower 可以快速設定多個 AWS 帳戶，確保治理符合您組織的需求。

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，或是擁有帳戶的利益相關者。

- 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：使用這些分隔和分組，為每個群組指定一個帳戶，並維護分隔要求。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。如果成員帳戶不得讓任何其他帳戶看到其使用或帳單資料，或者 AWS 需要與 分開的帳單，請定義多個管理或付款人帳戶。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
- 工作職能的AWS 受控政策
- AWS 多帳戶帳單策略
- AWS 區域 使用IAM政策控制對 的存取
- AWS Control Tower
- AWS Organizations
- 管理帳戶和成員帳戶的最佳實務
- 使用多個帳戶組織您的 AWS 環境
- 開啟共享的預留執行個體和 Savings Plans 折扣
- 合併帳單
- 合併帳單

相關範例：

- 分割 CUR和共用存取權

相關影片：

- 介紹 AWS Organizations
- 設定使用最佳實務的多帳戶 AWS 環境 AWS Organizations

相關範例：

- Well-Architected Labs：建立 AWS 組織 （ 100 級 ）
- 分割 AWS Cost and Usage Report 和共用存取權
- 定義電信公司的 AWS 多帳戶策略
- 最佳化的最佳實務 AWS 帳戶
- 使用 的組織單位最佳實務 AWS Organizations

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、修改或停用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曝險等級：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 Security Pillar 的身分和存取管理區段中建議，您需要強大的身分管理和許可，以便在正確的條件下為正確的人員提供正確的資源存取權。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。如需使用者、群組和身分驗證的最佳實務，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：

- 工作職能的AWS 受控政策
- AWS 多帳戶帳單策略
- AWS 建構良好的架構安全支柱
- AWS Identity and Access Management (IAM)
- AWS Identity and Access Management 政策

相關影片：

- 為何使用 Identity and Access Management

相關範例：

- Well-Architected 實驗室基本身分識別與存取
- AWS 區域 使用IAM政策控制對 的存取
- 開始您的雲端財務管理之旅：雲端成本操作

COST02-BP05 實作成本控制

根據組織政策以及定義的群組和角色實作控制。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曝險等級：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。了解工作負載和環境限制後，您可以強制執行管理。 AWS Budgets 可讓您設定通知，並定義 AWS 成本、用量和承諾折扣 （ Savings Plans和預留執行個體 ） 的每月預算。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。

使用 設定預算限制後 AWS Budgets，請使用 AWS Cost Anomaly Detection 來降低意外成本。 AWS Cost Anomaly Detection 是一種成本管理服務，使用機器學習來持續監控您的成本和用量，以偵測異常支出。其可協助您識別異常支出與根本原因，以便您迅速因應。首先，在 中建立成本監控 AWS Cost Anomaly Detection，然後設定金額閾值 （ 例如影響大於 $1，000 的異常警示 ），選擇您的提醒偏好設定。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 AWS Identity and Access Management和 AWS Organizations Service Control 政策 （ SCP ） 強制執行 中的治理政策。IAM 可讓您安全地管理對 AWS 服務和資源的存取。使用 IAM，您可以控制誰可以建立或管理 AWS 資源、可以建立的資源類型，以及可以在哪裡建立資源。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政策來強制執行正確的用量。SCP 可讓您集中控制組織中所有帳戶的可用許可上限，讓帳戶保持在存取控制準則內。SCPs 僅在已開啟所有功能的組織中可用，而且您可以設定 SCPs，依預設拒絕或允許成員帳戶的動作。如需實作存取管理的詳細資訊，請參閱 Well-Architected 安全支柱白皮書。

亦可透過管理 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 (包括資源的建立與停用) 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 以在支出超出政策時通知您。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 ） 或服務 AWS 。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。您也可以預先設定預算 AWS 動作，以強制執行特定IAM或SCP政策，或停止目標 Amazon EC2或 Amazon RDS執行個體。預算操作可以開始，也可以要求工作流程核准。
- 實施異常支出的通知：使用 AWS Cost Anomaly Detection 減少組織中的意外成本，並分析潛在異常支出的根本原因。建立成本監控以識別指定精細度的異常支出，並在 中設定通知後 AWS Cost

Anomaly Detection，它會在偵測到異常支出時傳送警示給您。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。在設定時使用 AWS Cost Categories AWS Cost Anomaly Detection ，以識別哪個專案團隊或業務單位團隊可以分析意外成本的根本原因，並及時採取必要的動作。

- 對用量實作控制：使用定義的組織政策，實作IAM政策和角色來指定使用者可以執行的動作，以及不能執行的動作。政策中可能包含多個組織 AWS 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：

- 工作職能的AWS 受控政策
- AWS 多帳戶帳單策略
- AWS 區域 使用IAM政策控制對 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 來追蹤我的支出和用量

相關範例：

- IAM存取管理政策範例
- 服務控制政策範例
- AWS 預算動作
- 建立IAM政策以使用 Tags 控制對 Amazon EC2 資源的存取
- 限制對特定 Amazon EC2 資源的 IAM Identity 存取
- 建立IAM政策，以限制依系列EC2使用的 Amazon
- 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 進行成本異常偵測的 Slack 整合 AWS Chatbot

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曝險等級：低

實作指引

透過有效追蹤專案生命週期，組織可以透過強化的規劃、管理和資源最佳化來實現更好的成本控制。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正在使用中，但 AWS 當發行新的服務或功能時，它們可以停用或採用。檢查工作負載的先前階段。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來標記資源，以固定審核工作負載的時間。舉例來說，如果開發環境上次是在幾個月前進行審核，那麼現在是時候再次審核，以探索是否可以採用新的服務，或是環境是否正在使用中。您可以在 myApplications 上將應用程式分組和標記 AWS ，以管理和追蹤重要性、環境、上次檢閱和成本中心等中繼資料。可以追蹤工作負載的生命週期，監控和管理應用程式的成本、運作狀態、安全狀態和效能。

AWS 提供各種管理和治理服務，可用於實體生命週期追蹤。您可以使用 AWS Config或 AWS Systems Manager 提供 AWS 資源和組態的詳細清查。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 提供的豐富事件和指標集相結合， AWS 可讓您建立重大生命週期事件的檢視，並主動管理 資源以減少不必要的成本。

與 Application Lifecycle Management （ ALM ） 類似，追蹤專案生命週期應涉及多個程序、工具和團隊合作，例如設計和開發、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洞見和增強控制，促進成功的專案規劃、實作和完成。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。

如需有關實作實體生命週期追蹤的詳細資訊，請參閱 AWS Well-Architected 卓越營運支柱白皮書。

實作步驟

- 建立專案生命週期監控程序：雲端卓越中心團隊必須建立專案生命週期監控程序。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。

- 執行工作負載審核：根據組織政策所定義，設定一個定期節奏以稽核現有專案並執行工作負載審核。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，例如完整或部分停用。

資源

相關文件：

- 上的標記指南 AWS
- 什麼是 ALM（應用程式生命週期管理）？
- 工作職能的AWS 受控政策

相關範例：

- AWS 區域 使用IAM政策控制對 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
- COST03-BP02 將組織資訊新增至成本和用量
- COST03-BP03 識別成本屬性類別
- COST03-BP04 建立組織指標

- COST03-BP05 設定帳單和成本管理工具
- COST03-BP06 根據工作負載指標分配成本

COST03-BP01 設定詳細資訊來源

設定成本管理和報告工具，以增強分析以及成本和用量資料的透明度。設定您的工作負載以建立日誌項目，以便追蹤和隔離成本和用量。

未建立此最佳實務時的曝險等級：高

實作指引

詳細的帳單資訊 (例如成本管理工具中的每小時精細度) 可讓組織更詳細地追蹤其耗用量，並協助他們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。這是從 接收詳細成本和用量資料的新建議方法 AWS。它為所有計費 AWS 服務 （ 與 相同的資訊CUR ） 提供每日或每小時用量精細度、費率、成本和用量屬性，以及一些改進。所有可能的維度都在 中，CUR例如標記、位置、資源屬性和帳戶 IDs。

根據您要建立的匯出類型，有三種匯出類型：標準資料匯出、使用 Amazon QuickSight 整合匯出至成本和用量儀表板，或舊版資料匯出。

- 標準資料匯出：表格的自訂匯出，可定期交付給 Amazon S3。
- 成本和用量儀表板：匯出和整合至 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 AWS Cost and Usage Report （ ） 的匯出CUR。

可以使用下列自訂來建立資料匯出：

- 包含資源 IDs
- 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 上執行容器的工作負載EKS，請啟用分割成本分配資料，以便根據容器工作負載使用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引入 AWS Cost and Usage Report。分割成本分配資料是透過計算叢集上執行之個別ECS服務和任務的成本來計算。

成本和用量儀表板會定期將成本和用量儀表板表格匯出至 S3 儲存貯體，並將預先建置的成本和用量儀表板部署至 Amazon QuickSight。如果想要在不進行自訂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。

如果需要，您仍可CUR以舊版模式匯出 ，其中您可以整合其他處理服務AWS Glue，例如準備資料以供分析，並使用 Amazon Athena 使用 查詢資料SQL來執行資料分析。

實作步驟

- 建立資料匯出：使用您想要的資料建立自訂匯出，並控制匯出結構描述。使用基本 建立帳單和成本管理資料匯出SQL，並透過與 Amazon 整合來視覺化您的帳單和成本管理資料 QuickSight。也可以使用標準模式匯出資料，以使用 Amazon Athena 等其他處理工具來分析資料。
- 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。設定包含所有識別符和資源的每小時精細程度報告IDs。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：若要存取過去 14 天內每小時精細度的成本和用量資料，請考慮在帳單主控台中啟用每小時和資源層級資料。
- 設定應用程式日誌記錄：確認您的應用程式會記錄其交付的每個業務成果，以便追蹤和衡量相應成果。確保此資料的精細度至少為每小時，以便與成本和用量資料相符。如需有關日誌記錄和監控的詳細資訊，請參閱卓越營運支柱 - AWS Well-Architected Framework。

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
- AWS 帳單和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和用量

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曝險等級：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：鍵為 `Environment`，其值為 `Production`。生產環境中的所有資源都會有此鍵/值對。標記可讓您使用有意義且相關的組織資訊，來分類和追蹤成本。您可以套用代表組織類別 (例如成本中心、應用程式名稱、專案或擁有者) 的標籤，並識別工作負載及其特性 (例如，測試或生產)，以在整個組織中劃分成本和用量歸屬。

當您將標籤套用至 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， 會將此資訊 AWS 新增至您的成本和用量報告。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。

在組織的帳戶中建立和實作 AWS 標記標準，可協助您以一致且統一的方式管理和管理 AWS 環境。使用 中的標籤政策 AWS Organizations 來定義如何在 帳戶中 AWS 的資源上使用標籤的規則 AWS Organizations。標籤政策可讓您輕鬆採用標準化方法來標記 AWS 資源

AWS 標籤編輯器可讓您新增、刪除和管理多個資源的標籤。利用標籤編輯器，您會搜尋要加標籤的資源，然後為搜尋結果中的資源管理標籤。

AWS Cost Categories可讓您將組織意義指派給成本，而不需要資源上的標籤。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：收集業務中的所有利益相關者，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，根據類別在工作負載中的所有資源上放置標籤。使用 CLI、標籤編輯器或 等工具 AWS Systems Manager 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：為驗證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。使用諸如 AWS CloudFormation 等服務來驗證資源在建立時是否已加上標記。您也可以使用 Lambda 函數建立自訂解決方案以進行自動標記，或者使用可定期掃描工作負載並移除任何未標記資源的微型服務，這非常適合用於測試和開發環境。
- 監控和報告標記：為驗證您可在整個組織中保有高層級標記，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。定期審查未標記資源的數量，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 Budgets 分析您的成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本與用量報告

相關影片：

- 如何標記我的 AWS 資源，以按成本中心或專案劃分帳單
- 標記 AWS 資源

COST03-BP03 識別成本屬性類別

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。

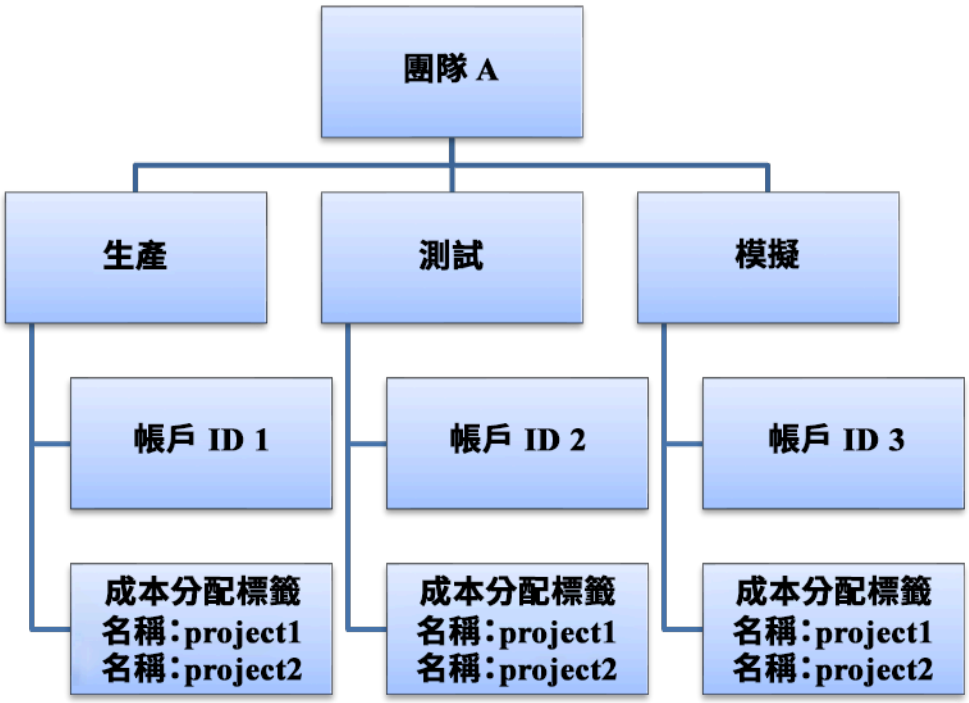
雲端支出權責劃分為有紀律的需求和成本管理建立了有力的誘因。對於將大部分雲端支出分配給取用業務單位或團隊的組織，這樣可以大幅節省雲端成本。此外，分配雲端支出有助於組織採用更多集中式雲端控管的最佳實務。

在定期會議中與財務團隊和其他相關利益相關者合作，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、生產和停用。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。

使用組織中的利益相關者定義成本屬性類別後，請使用AWS Cost Categories將成本和用量資訊分組到中有意義的類別 AWS 雲端，例如特定專案或 AWS 帳戶 部門或業務單位的成本。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。這些類別也可見於 AWS Cost Explorer AWS Budgets、 和 AWS Cost and Usage Report 。

例如，為您的業務單位（DevOps 團隊）建立成本類別，而且在每個類別下，會根據您定義的群組，建立具有多個維度（AWS 帳戶、成本分配標籤、服務或費用類型的規則）的多個規則（每個子類別的規則）。有了成本類別，即可使用以規則為基礎的引擎來整理成本。您設定的規則會將您的成本整理至各個類別。在這些規則中，您可以使用多個維度來篩選每個類別，例如特定 AWS 帳戶、 AWS 服務或費用類型。然後，您就可以在 AWS Billing and Cost Management 和成本管理主控台中使用多個產品中的這些類別。這包括 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 中。在 AWS Cost Explorer 和 中 AWS Budgets，成本類別會顯示為額外的帳單維度。您可以使用該值來篩選特定的成本類別值，或依成本類別分組。

實作步驟

- 定義您的組織類別：與內部利益相關者和業務單位會談，定義可反映組織結構和要求的類別。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：與內部利益相關者和業務單位會談，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，使用成本AWS Cost Categories和用量資訊，並將 AWS 成本和用量映射到有意義的類別 。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，以便可使用 AWS Cost Categories 在分類的結構中管理您的成本。

資源

相關文件：

- 標記 AWS 資源
- 使用成本分配標籤
- 使用 分析您的成本 AWS Budgets
- 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
- 在成本類別中拆分費用
- AWS Cost Categories 功能

相關範例：

- 使用成本 AWS Cost Categories組織成本和用量資料
- 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：成本類別

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曝險等級：高

實作指引

了解工作負載的輸出是否算得上業務成功。每個工作負載通常都有少數幾個能夠指出效能的主要輸出。如果您有包含許多元件的複雜工作負載，則可以排定清單的優先順序，或定義和追蹤每個元件的指標。與您的團隊合作，以了解要使用哪些指標。此單位將用於了解工作負載的效率，或每個業務輸出的成本。

實作步驟

- 定義工作負載成果：與業務中的利益相關者會面，並定義工作負載的成果。這些是客戶用量的主要衡量方式，並且必須是業務指標而非技術指標。每個工作負載應該有少量的高層級指標 (少於五個)。如果工作負載為不同的使用案例產生多個成果，請將它們分組為單一指標。
- 定義工作負載元件成果：或者，如果您有大型且複雜的工作負載，或者可以用明確定義的輸入和輸出，輕鬆地將工作負載分成元件 (例如微型服務)，則請為每個元件定義指標。工作應該反映元件的價值和成本。從最大的元件開始，並向較小的元件運行。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS Budgets 分析您的成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本與用量報告

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，以管理及優化雲端支出。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優化降低成本。

未建立此最佳實務時的風險暴露等級：高

實作指引

為了建立健全的權責劃分，應先將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。

為了鼓勵雲端支出的權責劃分，使用者應有權存取可讓他們檢視成本與用量的工具。 AWS 建議您基於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 Organization 等工具建立多個 AWS 帳戶。標記支援 AWS 的資源，並根據組織結構（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，並將其與 AWS Cost Categories對應，將業務單位的帳戶分組到其成本中心，以便業務單位擁有者可以在同一個位置查看多個帳戶的耗用。
- 存取：在合併帳單中追蹤整個組織的帳單資訊。確認適當的利益相關者和企業擁有者具有存取權。

- 控制：使用正確的防護機制建立有效的治理機制，以防止在使用 Service Control 政策（SCP）、標籤政策、IAM政策和預算提醒時出現意外情況。例如，您可以允許團隊只使用有效的控制機制在首選區域中建立特定資源，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：設定儀表板，顯示目前的成本和用量級別。儀表板應在工作環境中顯眼的位置提供，類似於營運儀表板。可以匯出資料，並使用 AWS 成本最佳化中心的成本和用量儀表板或任何支援產品來建立此可見性。您可能需要為不同的角色建立不同的儀表板。例如，管理員儀表板可能與工程儀表板不同。
- 通知：當成本或用量超過定義的限制，且使用 AWS Budgets 或 AWS Cost Anomaly Detection 發生異常時，提供通知。
- 報告：彙總所有成本和用量資訊。利用詳細的可歸因成本資料，提高雲端支出的意識和責任。建立與使用這些報告的團隊相關且包含建議的報告。
- 追蹤：顯示相對於設定的總目標或具體目標目前成本和用量的狀況。
- 分析：允許團隊成員使用不同的篩選條件 (資源、帳戶、標籤等) 執行自訂和深度分析，精確到每小時、每日或每月。
- 檢查：隨時掌握資源部署和成本最佳化商機的最新資訊。使用 Amazon CloudWatch、Amazon SNS 或 Amazon 取得通知SES，以便在組織層級進行資源部署。使用 AWS Trusted Advisor 或 檢閱成本最佳化建議 AWS Compute Optimizer。
- 趨勢報告：以所需的精細度顯示所需期間內的成本與用量變化。
- 預測：使用您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。

您可以使用 AWS 成本最佳化中心，了解從中央位置整合的潛在成本節省機會，並建立資料匯出以便與 Amazon Athena 整合。您也可以使用 AWS Cost Optimization Hub 部署成本和用量儀表板，該儀表板利用 Amazon QuickSight 進行互動式成本分析並保護成本洞察分享。

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，但請務必驗證價值主張。

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群組，以存取所需的成本和用量報告以供其使用，並使用 AWS Identity and Access Management 來控制諸如 AWS Cost Explorer等工具的存取權。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。
- 管理成本標籤和類別：跨團隊、業務單位、應用程式、環境和專案來管理成本。使用資源標籤依成本分配標籤來管理成本。使用標籤、帳戶、服務等，依據維度建立成本類別以對應您的成本。

- 設定 AWS 預算：針對工作負載的所有帳戶設定 AWS 預算。使用標籤和成本類別，設定整體帳戶支出的預算以及工作負載的預算。在 AWS Budgets 中設定通知，以便在超過預算金額或預估成本超過預算時收到提醒。
- 設定 AWS 成本異常偵測：針對您建立的帳戶、核心服務或成本類別使用AWS 成本異常偵測，以監控您的成本和用量，並偵測異常支出。您可以在彙總報告中個別接收警示，並在電子郵件或 Amazon SNS主題中接收警示，可讓您分析和判斷異常的根本原因，並識別推動成本增加的因素。
- 使用成本分析工具：針對您的工作負載和帳戶來設定 AWS Cost Explorer，將成本資料視覺化以進行深入分析。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。
- 使用節省成本的分析工具：使用 AWS 成本最佳化中心，透過量身打造的建議來識別節省機會，包括刪除未使用的資源、權利化、節省計劃、保留和運算最佳化工具建議。
- 設定進階工具：您可以選擇性地建立視覺效果以促進交互式分析和成本洞見分享。透過 AWS Cost Optimization Hub 上的資料匯出，您可以 QuickSight 為組織建立由 Amazon 提供支援的成本和用量儀表板，以提供其他詳細資訊和精細度。您也可以使用 Amazon Athena 中的資料匯出來實作進階分析功能，並在 Amazon QuickSight上建立儀表板。與 AWS 合作夥伴進行合作，採用雲端管理解決方案，進行整合式雲端帳單監控與最佳化。

資源

相關文件：

- 什麼是 AWS Billing and Cost Management 和 Cost Management？
- 建立最佳實務 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 分析您的成本
- 使用 分析您的成本 AWS Cost Explorer
- 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 的提醒 KPI

相關範例：

- Amazon 支援的成本和用量儀表板 QuickSight
- AWS 成本與用量管控研討會

COST03-BP06 根據工作負載指標分配成本

依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曝險等級：低

實作指引

成本優化意味著以最低的價格提供業務成果，只有依工作負載指標 (按工作負載效率測量) 來分配工作負載成本才能達成。透過日誌檔案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。如果您擁有靜態成本元件 (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，有助於您確定優化活動的重點。

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，可以透過某些方法對使用情況進行分類並追蹤成本。如果您需要追蹤 Amazon ECS和 AWS Batch 共用資源，您可以在 中啟用分割成本分配資料 AWS Cost Explorer。使用分割成本分配資料，您可以了解並優化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 Amazon Athena 和 Amazon 等分析服務，為整體工作負載和任何元件 QuickSight 建立效率儀表板。

資源

相關文件：

- 標記 AWS 資源
- 使用 AWS Budgets 分析您的成本

- 使用 Cost Explorer 分析成本
- 管理 AWS 成本與用量報告

相關範例：

- AWS Batch 使用 AWS 分割成本分配資料改善 Amazon ECS和 的成本可見性

COST 4. 如何停用資源？

實作從專案啟動到 的變更控制和資源管理 end-of-life。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 追蹤其生命週期的資源
- COST04-BP02 實作停用程序
- COST04-BP03 停用資源
- COST04-BP04 自動停用資源
- COST04-BP05 強制執行資料保留政策

COST04-BP01 追蹤其生命週期的資源

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曝險等級：高

實作指引

停用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) 可協助您識別要停用的資產 (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，或標記停用日期。然後，即可對這些標籤執行報告。功能標記的範例值可以是 `feature-X testing`，可識別資源在工作負載生命週期的用途。另一個範例是TTL針對 資源使用 `LifeSpan`或 ，例如 to-be-deleted標籤金鑰名稱和值，以定義停用的時段或特定時間。

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，確保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。有關標記使用案例、策略和技巧的更多詳細資訊，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，在輸入請求或輸出完成時啟動。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。如需有關未使用或未充分利用資源的詳細資訊，請參閱 AWS Trusted Advisor 成本最佳化檢查。
- 群組 AWS 資源：建立 AWS 資源的群組。您可以使用 AWS Resource Groups 來組織和管理位於相同 中的 AWS 資源 AWS 區域。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。使用標籤編輯器將標籤大量新增至支援的資源。考慮使用 AWS Service Catalog 來建立、管理並向最終使用者分發批准的產品組合，並管理產品生命週期。

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
- 發佈自訂指標

相關影片：

- 如何使用 最佳化成本 AWS Trusted Advisor

相關範例：

- 組織 AWS 資源
- 使用 最佳化成本 AWS Trusted Advisor

COST04-BP02 實作停用程序

實作識別和停用未使用資源的程序。

未建立此最佳實務時的曝險等級：高

實作指引

在您的組織中實作標準化程序，以識別並移除未使用的資源。此程序應該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

- 建立並實作停用程序：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停用程序。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。詳述停用資源的必要步驟，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停用程序已經執行。

  使用下列停用步驟來引導您了解過程中應檢查的事項：

  - 識別要停用的資源：識別在 AWS 雲端中有資格停用的資源。記錄所有必要資訊，並排定停用時間。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。
  - 協調與溝通：與工作負載擁有者合作，確認要停用的資源
  - 記錄中繼資料和建立備份：如果生產環境中的資源需要，或者它們是關鍵資源，則記錄中繼資料（例如公有 IPs、區域、AZVPC、子網路和安全群組）並建立備份（例如 Amazon Elastic Block Store 快照或接受 AMI、金鑰匯出和憑證匯出）。
  - 驗證 infrastructure-as-code：判斷資源是否使用 、 AWS CloudFormation Terraform 或任何其他infrastructure-as-code部署工具部署 AWS Cloud Development Kit (AWS CDK)，以便在必要時重新部署。
  - 防止存取：在一段時間內套用限制性控制，以防止在判斷是否需要資源時使用資源。確認資源環境可在必要時恢復為原始狀態。
  - 遵循您的內部停用程序：遵循組織的管理任務和停用程序，例如從組織網域中移除資源、移除DNS記錄，以及從組態管理工具、監控工具、自動化工具和安全工具中移除資源。

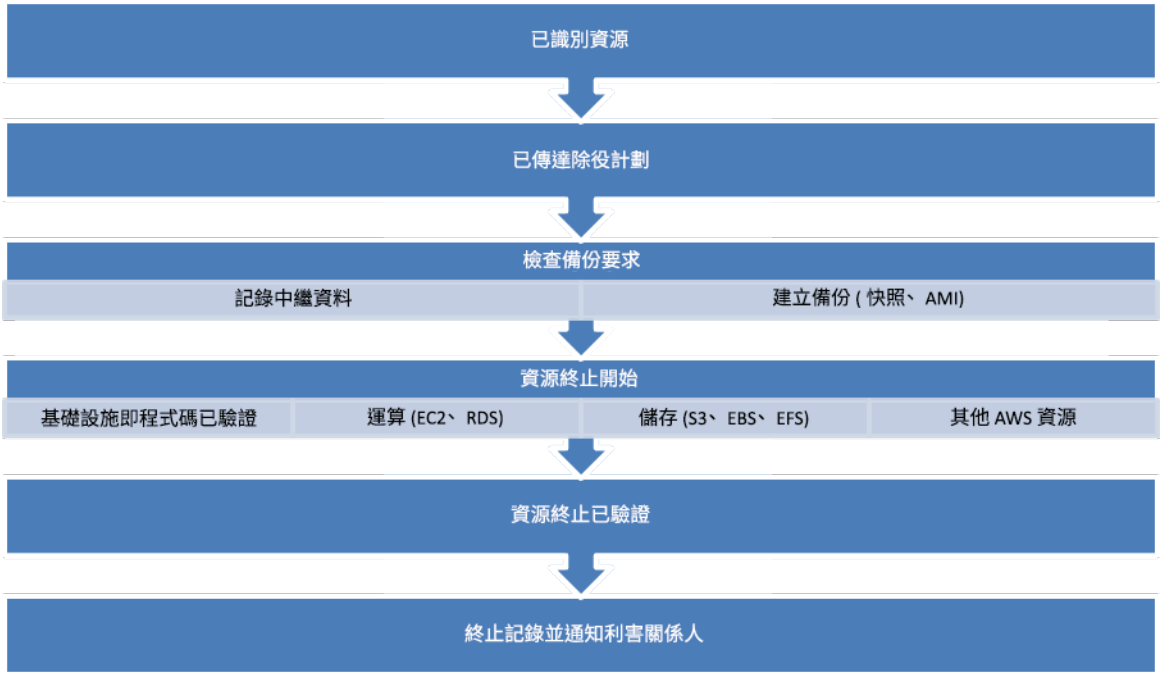
  如果資源是 Amazon EC2執行個體，請參閱下列清單。如需更多詳細資訊，請參閱如何刪除或終止我的 Amazon EC2 資源？

  - 停止或終止所有 Amazon EC2執行個體和負載平衡器。Amazon EC2執行個體在終止之後，會在主控台中顯示一小段時間。您不需要為任何未處於執行中狀態的執行個體付費
  - 刪除 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 位址。
  - 取消註冊所有 Amazon Machine Images（AMIs）。

- 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 的最短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短儲存持續時間摘要，請參閱 Amazon S3 儲存類別的效能。如需有關如何計算提前刪除費用的詳細資訊，請參閱 Amazon S3 定價。

下面的簡單停用程序流程圖會概述停用步驟。在停用資源之前，請先確認您確定要停用的資源沒有被組織使用。



資源停用流程。

資源

相關文件：

- AWS Auto Scaling
- AWS Trusted Advisor
- AWS CloudTrail

相關影片：

- 刪除 CloudFormation 堆疊，但保留一些資源

- 了解哪些使用者啟動了 Amazon EC2執行個體

相關範例：

- 刪除或終止 Amazon EC2 資源
- 了解哪個使用者啟動了 Amazon EC2執行個體

COST04-BP03 停用資源

停用由諸如定期稽核或用量變更等事件觸發的資源。通常會定期執行停用，其執行方式可以手動，也可以自動。

未建立此最佳實務時的曝險等級：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。搜尋和停用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。搜尋和停用事件也可由外部事件啟動，例如市場條件變化或產品終止。

實作步驟

- 停用資源：這是不再被需要或授權協議結束的 AWS 資源的折舊階段。請先完成所有最終檢查，再移至處置階段並停用資源，以防止發生任何不需要的中斷，例如擷取快照或備份。使用停用程序，停用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：停用資源 (Level 100)

COST04-BP04 自動停用資源

設計工作負載，在識別和停用非關鍵資源、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曝險等級：低

實作指引

使用自動化來降低或消除停用程序的相關成本。將工作負載設計為執行自動停用，可降低工作負載生命週期內的整體成本。您可以使用 Amazon EC2 Auto Scaling 或 Application Auto Scaling 來執行停用程序。您也可以使用 API或 SDK 實作自訂程式碼，以自動停用工作負載資源。

現代應用程式是先建置的無伺服器應用程式，這是優先採用無伺服器服務的策略。針對堆疊的三個層 AWS 開發無伺服器服務：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 Amazon EC2 Auto Scaling 或 Application Auto Scaling ：針對支援的資源，請使用 Amazon EC2 Auto Scaling 或 Application Auto Scaling 進行設定。這些服務可協助您在使用 AWS 服務時最佳化使用率和成本效率。當需求下降時，這些服務會自動移除超額的資源容量，以免您超支。
- 設定 CloudWatch 以終止執行個體：執行個體可設定為使用CloudWatch 警示 終止。使用來自於停用程序的指標，透過 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。
- 在工作負載中實作程式碼：您可以使用 AWS SDK或 AWS CLI 來停用工作負載資源。在與 整合的應用程式內實作程式碼， AWS 並終止或移除不再使用的資源。
- 使用無伺服器服務：優先在 上建置無伺服器架構和事件驅動架構 AWS ，以建置和執行應用程式。AWS 提供多項無伺服器技術服務，可本質上提供自動最佳化的資源使用率和自動除役 （擴展和擴展）。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：

- Amazon EC2 Auto Scaling
- Amazon EC2 Auto Scaling 入門
- Application Auto Scaling
- AWS Trusted Advisor

- 上的無伺服器 AWS
- 建立警示以停止、終止、重新啟動或復原執行個體
- 將終止動作新增至 Amazon CloudWatch 警示

相關範例：

- 排程自動刪除 AWS CloudFormation 堆疊
- Well-Architected 實驗室 — 自動停用資源 (Level 100)
- Servian AWS Auto Cleanup

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曝險等級：中

使用資料保留政策和生命週期政策，降低已識別資源的停用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 自動化建立和刪除 Amazon Elastic Block Store 快照和 Amazon EBS 後端 Amazon Machine Images （AMIs），並使用 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 實作自訂程式碼，為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：在 Amazon Data Lifecycle Manager 上使用生命週期政策，自動刪除 Amazon EBS快照和 Amazon EBS後端 AMIs。
- 在儲存貯體上設定生命週期組態：在儲存貯體上使用 Amazon S3 生命週期組態，定義 Amazon S3 在物件生命週期中採取的動作，以及根據您的業務需求在物件生命週期結束時進行刪除。

資源

相關文件：

- AWS Trusted Advisor
- Amazon Data Lifecycle Manager

- 如何在 Amazon S3 儲存貯體上設定生命周期組態

相關影片：

- 使用 Amazon Data Lifecycle Manager 自動化 Amazon EBS 快照
- 使用生命周期組態規則來清空 Amazon S3 儲存貯體

相關範例：

- 使用生命周期組態規則來清空 Amazon S3 儲存貯體
- Well-Architected 實驗室：自動停用資源 (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 是建置區塊 AWS 服務。受管服務，例如 Amazon RDS和 Amazon DynamoDB是更高層級的服務，或應用程式層級 AWS 的服務。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 識別組織的成本要求
- COST05-BP02 分析工作負載的所有元件
- COST05-BP03 對每個元件執行徹底分析
- COST05-BP04 選取具有成本效益授權的軟體
- COST05-BP05 選取此工作負載的元件，以符合組織優先順序來最佳化成本
- COST05-BP06 執行隨時間不同用量的成本分析

COST05-BP01 識別組織的成本要求

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曝險等級：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 支柱之間建立平衡，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開會，包括產品管理人員、應用程式擁有者、開發和營運團隊、管理層和財務部人員。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。
- 解決技術債務並將其記錄在案：在工作負載檢閱期間，處理技術債務。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。向其他利益相關者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 建構您的產品以滿足可用性目標和運作時間服務等級協議 （ SLAs ）

- OPS01-BP06 評估權衡

相關文件：

- AWS 總擁有成本（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，例如目前和預計的成本。

未建立此最佳實務時的曝險等級：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包含各種服務。對於每個元件，都可以選擇特定 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。

如 COST05-BP01 所述識別組織需求 找出成本 的組織需求後，請對工作負載中的所有元件執行徹底分析。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省來考慮分析成本。在分析此工作負載的所有元件上所花費的努力應與最佳化該特定元件所預期的潛在節省或改進相當。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估算，會為此元件建立最佳整體結果。

工作負載可能會隨時間改變，而且如果工作負載架構或用量變化，適當的服務組合可能並非最佳。選擇服務的分析必須納入目前和未來的工作負載狀態以及用量水平。為未來的工作負載狀態或用量實作服務，可減少或消除未來變更所需的工作量，藉此降低整體成本。例如，使用 EMR Serverless 最初可能是適當的選擇。不過，隨著該服務的耗用增加，在 上轉換 EMREC2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）可以分析概念驗證（PoC或執行中環境的成本。也可以使用 AWS Pricing Calculator 來估算工作負載成本。

撰寫工作流程，供技術團隊檢閱其工作負載。讓此工作流程保持簡單，同時也涵蓋所有必要步驟，以確保團隊了解工作負載的每個元件及其定價。然後，您的組織可以根據每個團隊的特定需求來遵循和自訂此工作流程。

1. 列出工作負載使用的每個服務：這是一個很好的起點。確定目前使用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、資料傳輸和功能特定定價等因素，不同的 AWS 服務具有不同的定價模型。
3. 專注於具有非預期工作負載成本，且不符合預期用量和業務結果的服務：識別與使用 或 AWS Cost Explorer 的價值或用量不成比例的異常值或服務 AWS Cost and Usage Report。將成本與業務成果相互關聯以優先考慮最佳化工作至關重要。
4. AWS Cost Explorer、 CloudWatch Logs、VPCFlow Logs 和 Amazon S3 Storage Lens 以了解這些高成本的根本原因：這些工具對於診斷高成本至關重要。每項服務都可提供不同的視角來檢視和分析使用情況和成本。例如，Cost Explorer 有助於判斷整體成本趨勢、 CloudWatch Logs 提供營運洞察、VPCFlow Logs 顯示 IP 流量，以及 Amazon S3 Storage Lens 可用於儲存分析。
5. 使用 AWS Budgets 為服務或帳戶設定特定金額的預算：設定預算是管理成本的主動方式。使用 AWS Budgets 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。CloudWatch alarms 可以在違反特定閾值時通知您，進而改善介入回應時間。

透過對所有工作負載元件進行策略審查 (無論其目前屬性為何)，可隨著時間的推移帶來顯著的改進和財務方面的節省。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：建立工作負載元件清單。使用此清單可確認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。按功能將資源分組在一起以提高效率 (例如，生產資料庫儲存 (若有多個資料庫的話))。
- 設定元件清單的優先順序：取得元件清單並按照工作順序排列其優先順序。這通常是依最昂貴到最便宜的元件成本排序，或依貴組織優先事項所定義的關鍵性排序。
- 執行分析：對於清單上的每個元件，檢閱可用的選項和服務並選擇最適合您組織優先事項的選項。

資源

相關文件：

- AWS Pricing Calculator
- AWS Cost Explorer
- Amazon S3 儲存類別
- 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曝險等級：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「平移」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。上的受管服務可 AWS 消除維護服務的營運和管理負擔，例如修補或升級作業系統，並可讓您專注於創新和業務。

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，您可能希望透過遷移至 Amazon Relational Database Service （ Amazon RDS ） 等 database-as-a-service平台，或遷移應用程式至 等完全受管平台，以減少管理資料庫執行個體的時間AWS Elastic Beanstalk。

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。您可以使用 AWS Managed Services AWS Management Console 或 AWS APIs 和 修改 屬性SDKs，使資源需求與不斷變化的需求保持一致。例如，您可以增加或減少 Amazon EMR叢集 （ 或 Amazon Redshift 叢集 ） 上的節點數量以縮減或縮減。

您也可以在 AWS 資源上封裝多個執行個體，以啟用更高密度的用量。例如，您可以在單一 Amazon Relational Database Service （ Amazon RDS ） 資料庫執行個體上佈建多個小型資料庫。隨著用量的增加，您可以使用快照和還原程序將其中一個資料庫遷移至專用的 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。使用 APIs和 與系統和監控工具整合SDKs，例如 Amazon ，可以將修改服務所需的持續工作減少到幾乎零 CloudWatch。

Amazon RDS、Amazon Redshift 和 Amazon ElastiCache 提供受管資料庫服務。Amazon AthenaAmazon EMR和 Amazon OpenSearch Service 提供受管分析服務。

AMS 是一項代表企業客戶和合作夥伴操作 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 使用具有自動化的企業雲端操作模型，讓您符合組織需求、更快地進入雲端，並降低持續的管理成本。

實作步驟

- 執行徹底的分析：使用元件清單，從最高優先順序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：考慮您所管理資源的營運成本，並將其與受 AWS 管資源進行比較。例如，檢閱在 Amazon EC2執行個體上執行的資料庫，並與在 Amazon EMR上執行 Apache Spark 的 Amazon RDS選項（ AWS 受管服務）或 Amazon 進行比較EC2。從自我管理工作負載移至 AWS 完全受管工作負載時，請仔細研究您的選項。要考慮的三個最重要的因素是您要使用的受管服務類型、將用來遷移資料的程序，以及了解 AWS 共同責任模型。

資源

相關文件：

- AWS 總擁有成本（TCO）計算器
- Amazon S3 儲存類別
- AWS 雲端 產品
- AWS 共同責任模型

相關影片：

- 為什麼要移至受管資料庫？
- 什麼是 AmazonEMR？我該如何使用它來處理資料？

相關範例：

- 為什麼要移至受管資料庫
- 使用 將來自相同SQL伺服器資料庫的資料合併到單一 Amazon RDS for SQL Server 資料庫 AWS DMS
- 將大規模資料交付至 Amazon Managed Streaming for Apache Kafka（Amazon MSK）
- 將 ASP.NET Web 應用程式遷移至 AWS Elastic Beanstalk

COST05-BP04 選取具有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。如果需要授權軟體，請避免與任意屬性繫結的授權，例如 CPUs，尋找與輸出或結果繫結的授權。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曝險等級：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。根據業務需求、工程師的技能、預測用量或其他技術相依性，組織可以考慮在 上使用開放原始碼軟體 AWS ，以將授權成本降至最低。換句話說，使用開放原始碼軟體可降低軟體授權的成本。隨著工作負載的大小擴展，這可能會對工作負載成本產生重大影響。

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。授權成本也可以獨立於輸送量或用量來擴展，例如依硬體擴展的授權（CPU 受限制的授權）。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，使用 Linux 作業系統在 us-east-1 中操作 Amazon EC2執行個體，相較於在 Windows 上執行的另一個 Amazon EC2執行個體，可讓您將成本降低約 45%。

AWS Pricing Calculator 提供了一種綜合方法，將各種資源的成本與不同的授權選項進行比較，例如 Amazon RDS執行個體和不同的資料庫引擎。此外， 為現有工作負載的成本 AWS Cost Explorer 提供了寶貴的觀點，特別是具有不同授權的工作負載。對於許可證管理，AWS License Manager 提供一種簡化的方法來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：

- 開放原始碼位於 AWS

- AWS 總擁有成本（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，以符合組織優先順序來最佳化成本

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進行運算，例如 AWS Lambda 及針對靜態網站的 Amazon Simple Storage Service (Amazon S3)。盡可能將應用程式容器化，並使用 AWS Managed Container Services，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 Lambda、Amazon Simple Queue Service（Amazon SQS）、Amazon SNS和 Amazon SES等無伺服器或應用程式層級服務。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。

無伺服器服務也可以使用事件驅動型架構。事件驅動型架構是推送架構，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。這表示減少網路頻寬消耗、減少CPU使用率、減少閒置機群容量，以及減少 SSL/TLS 交握。

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，如果您需要在 上檢閱資料庫的預期流量 AWS，請考慮增加執行個體大小或使用 Amazon ElastiCache 服務（Redis 或 Memcached）為您的資料庫提供快取機制。
- 評估事件驅動型架構：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 總擁有成本（TCO）計算器
- AWS Serverless
- 什麼是事件驅動型架構
- Amazon S3 儲存類別
- 雲端產品
- Amazon ElastiCache（Redis OSS）

相關範例：

- 事件驅動型架構入門
- 事件驅動型架構
- Statsig 如何使用 Amazon ElastiCache（Redis OSS）以更具成本效益的方式執行 100 倍
- 使用 AWS Lambda 函數的最佳實務

COST05-BP06 執行隨時間不同用量的成本分析

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曝險等級：中

實作指引

隨著新服務和功能 AWS 發行，工作負載的最佳服務可能會變更。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。另一個需要檢閱的方面是使用模式的變更。用量的重大變更可能表示替代服務更理想。

如果您需要將資料移至 AWS 雲端，您可以選擇任何種類的服務 AWS 優惠和合作夥伴工具，以協助您遷移資料集，無論是檔案、資料庫、機器映像、區塊磁碟區，甚至是磁帶備份。例如，若要在邊緣來回移動大量資料 AWS 或在邊緣處理資料，您可以使用其中一個 AWS 專用裝置，以符合成本效益的方式離線移動 PB 的資料。另一個範例是更高的資料傳輸率，直接連線服務可能比為您的企業VPN提供所需一致連線能力的 便宜。

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。

透過與組織中的利益相關者討論，針對不同使用情況執行成本分析，並使用 AWS Cost Explorer 的預測功能來預測服務變更的潛在影響。使用 AWS Budgets、 CloudWatch 帳單警示監控用量層級的啟動 AWS Cost Anomaly Detection ，並更快識別和實作最具成本效益的服務。

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。與利益相關者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。識別您希望更多使用者使用您的 AWS 資源的日曆日、週或月，這表示您應該增加現有資源的容量或採用其他服務來降低成本並提高效能。
- 根據預測用量執行成本分析：使用定義的使用模式，在上述每個點執行分析。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、大小和數字
- COST06-BP03 根據指標自動選取資源類型、大小和數字
- COST06-BP04 考慮使用共用資源

COST06-BP01 執行成本建模

識別組織要求 (例如業務需求和現有承諾)，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曝險等級：高

實作指引

為您的工作負載及其每個元件執行成本建模，以了解資源之間的平衡，並根據特定效能等級，找出工作負載中每個資源的合適大小。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 效能效率支柱的檢閱一節。

例如，若要為包含運算資源的工作負載建立成本建模，AWS Compute Optimizer 可協助正在執行的工作負載的成本建模。它根據歷史用量，提供運算資源的合適大小建議。確定 CloudWatch 代理程式已部署到 Amazon EC2執行個體，以收集記憶體指標，協助您在 中提供更準確的建議 AWS Compute Optimizer。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。

您可以搭配自訂日誌使用多種服務作為資料來源，以針對其他服務和工作負載元件授權操作，例如 AWS Trusted Advisor、Amazon CloudWatch 和 Amazon CloudWatch Logs。 AWS Trusted Advisor 會檢查資源，並以低使用率標記資源，協助您調整資源大小並建立成本模型。

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。
- 考量您的現有承諾、其他工作負載的選定定價模型，以及更快速創新和專注於核心業務價值的能力，為您的計劃工作負載選擇合適的 AWS 服務。

實作步驟

- 針對資源執行成本建模：將工作負載或概念驗證部署到具有要測試之特定資源類型和大小的獨立帳戶。使用測試資料執行工作負載，並記錄輸出結果以及測試執行時的成本資料。然後，重新部署工作負載或變更資源類型和大小，並再次執行測試。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
- 找出機會進行適當調整
- Amazon CloudWatch 功能
- 成本最佳化：Amazon EC2 Right Sizing
- AWS Compute Optimizer
- AWS 定價計算器

相關範例：

- 執行資料驅動的成本建模
- 估算規劃 AWS 資源組態的成本
- 選擇正確的 AWS 工具

COST06-BP02 根據資料選取資源類型、大小和數字

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曝險等級：中

實作指引

Amazon EC2提供多種執行個體類型，具有不同層級的 CPU、記憶體、儲存體和聯網容量，以適應不同的使用案例。這些執行個體類型具有不同的 CPU、記憶體、儲存體和聯網功能組合，讓您在為專案選擇正確的資源組合時，具有多樣性。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 核心數目
- GPU 核心
- 系統記憶體數量 （RAM）
- 儲存類型和空間
- 網路頻寬要求

識別運算需求的目的和需要的執行個體，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
- 儲存優化

- 加速運算
- HPC 最佳化

如需特定 Amazon EC2執行個體系列可以實現的特定目的和使用案例的更深入了解，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針對執行個體類型、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使用量層級，然後選擇指定的資源類型和大小。依賴成本建模資料，在考慮執行個體所需的資料傳輸率的情況下，決定虛擬 CPUs、總記憶體 GiB）、本機執行個體存放磁碟區 （GB）、Amazon EBS磁碟區和網路效能層級的數量。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
- 成本最佳化：EC2正確調整大小

相關影片：

- 為您的工作負載選取正確的 Amazon EC2執行個體
- 調整您的服務

相關範例：

- 更輕鬆地探索和比較 Amazon EC2執行個體類型

COST06-BP03 根據指標自動選取資源類型、大小和數字

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，以優化成本。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曝險等級：低

實作指引

在工作負載中建立意見回饋迴圈，使用執行中工作負載的作用中指標來變更該工作負載。您可以使用 受管服務，例如 AWS Auto Scaling，您可以設定 為您執行正確的大小調整操作。 AWS 還提供 APIs、 SDKs和 功能，允許資源以最少的努力進行修改。您可以將工作負載程式設計為 stop-and-start 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。

某些 AWS 服務已內建自動類型或大小選擇，例如 Amazon Simple Storage Service Intelligent-Tiering。Amazon S3 Intelligent-Tiering 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

- 透過設定工作負載指標來提高您的可觀測性：擷取工作負載的關鍵指標。這些指標提供客戶體驗的指示，例如工作負載輸出，並符合資源類型和大小之間的差異，例如 CPU和記憶體用量。針對運算資源，分析效能資料以正確調整 Amazon EC2執行個體的大小。識別閒置的執行個體，以及未充分使用的執行個體。要尋找的關鍵指標是CPU用量和記憶體使用率 （ 例如 90% 的時間 40% CPU使用率，如使用 AWS Compute Optimizer 和啟用記憶體使用率 進行授權中所述 ）。識別四週內最大CPU用量和記憶體使用率低於 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。對於 Amazon S3 等儲存資源，您可以使用 Amazon S3 Storage Lens，預設情況下，您可以在儲存貯體層級查看各種類別的 28 個指標，以及在儀表板中查看 14 天的歷史資料。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。
- 檢視許可化建議：使用 Cost Management 主控台中的 AWS Compute Optimizer 和 Amazon EC2 許可化工具中的許可化建議，或檢閱 AWS Trusted Advisor 資源的正確大小，以調整工作負載。正確調整不同資源的大小時，請務必使用適當的工具，並遵循正確的大小調整指南，無論是 Amazon EC2執行個體、 AWS 儲存類別或 Amazon RDS執行個體類型。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 從整個組織的指標分析衍生出來的情境式建議，您可以立即採取步驟來最佳化儲存。

- 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。您可以在單一 Auto Scaling 群組內啟動和自動擴展隨需執行個體和 Spot 執行個體組成的機群。除了獲得使用 Spot 執行個體的折扣之外，您還可以使用預留執行個體或 Savings Plan，以獲得定期隨需執行個體定價的折扣費率。所有這些因素組合可協助您最佳化 Amazon EC2執行個體的成本節省，並判斷應用程式所需的規模和效能。您也可以在 Auto Scaling 群組（ABS）中使用以屬性為基礎的執行個體類型選取（）策略，這可讓您將執行個體需求表達為一組屬性，例如 v CPU、記憶體和儲存體。 Auto Scaling ASG 您可以在較新一代的執行個體類型發佈時自動使用，並使用 Amazon EC2 Spot 執行個體存取更廣泛的容量範圍。Amazon EC2 Fleet 和 Amazon EC2 Auto Scaling 會選取並啟動符合指定屬性的執行個體，無需手動挑選執行個體類型。對於儲存資源，您可以使用 Amazon S3 Intelligent Tiering 和 Amazon EFS Infrequent Access 功能，這可讓您自動選取儲存類別，以便在資料存取模式變更時自動節省儲存成本，而不會影響效能或營運開銷。

資源

相關文件：

- AWS Auto Scaling
- AWS 大小適中
- AWS Compute Optimizer
- Amazon CloudWatch 功能
- CloudWatch 設定
- CloudWatch 發佈自訂指標
- Amazon EC2 Auto Scaling 入門
- Amazon S3 Storage Lens
- Amazon S3 Intelligent-Tiering
- Amazon EFS 不常存取
- 使用 啟動 Amazon EC2執行個體 SDK

相關影片：

- 適當調整服務的大小

相關範例：

- Amazon EC2 Fleet Auto Scaling 的屬性型執行個體類型選擇
- 使用已排程的擴展，針對成本最佳化 Amazon Elastic Container Service
- 使用 Amazon EC2 Auto Scaling 進行預測性擴展
- 使用 Amazon S3 Storage Lens 將成本最佳化並了解使用情況
- Well-Architected 實驗室：適當調整建議的大小 (Level 100)

COST06-BP04 考慮使用共用資源

對於已部署的多個業務單位的組織層級服務，請考慮使用共用資源來增加使用率並降低總擁有成本（TCO）。使用共用資源可能是一個具成本效益的選項，可透過使用現有解決方案、共用元件或兩者來集中化管理和成本。在帳戶界限內或專用帳戶中管理常用功能，例如監控、備份和連線。還可以透過實作標準化、減少重複及降低複雜性來降低成本。

未建立此最佳實務時的曝險等級：中

實作指引

當多個工作負載導致相同的功能時，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。請考慮使用現有資源 (尤其是共用資源)，例如非生產資料庫伺服器或目錄服務，透過遵循安全性最佳實務和組織法規來降低雲端成本。為了實現最佳價值和效率，將成本 (使用回報 (showback) 和計費 (chargeback)) 分配到推動消費的相關業務領域至關重要。

回報 (showback) 是指將雲端成本分解為可歸因類別的報告，例如消費者、業務單位、總賬帳戶或其他負責實體。回報的目標是向團隊、業務單位或個人展示其所用雲端資源的成本。

計費 (chargeback) 是指根據適合特定財務管理程序的策略，將中央服務支出分配給成本單位。對於客戶而言，計費 (chargeback) 會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。透過建立計費機制，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，對於較不重要的工作負載，使用具有一般組態的共用資源。為了進一步最佳化成本，請僅為關鍵工作負載預留專用伺服器。共用資源或在多個帳戶之間佈建資源，以便有效地管理它們。即使在不同的開發、測試和生產環境中，安全共用也是可行的，而且不會影響組織結構。

為了提高您對容器化應用程式的了解並最佳化其成本和用量，請使用分割成本分配資料，它可幫助您根據應用程式使用共用運算和記憶體的方式，將成本分配給個別業務實體。分割成本分配資料可協助

您在 Amazon Elastic Container Service （Amazon ECS） 或 Amazon Elastic Kubernetes Service （Amazon ） 上執行的容器工作負載中，達成任務層級的顯示和退款EKS。

對於分散式架構，請建置共用服務 VPC，該服務可讓您集中存取每個 中工作負載所需的共用服務 VPCs。這些共用服務可能包含目錄服務或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 中建置資源VPC。

當您使用共用資源時，可以節省營運成本、最大化資源利用率並提高一致性。在多帳戶設計中，您可以集中託管一些 AWS 服務，並使用中樞中的多個應用程式和帳戶來存取它們，以節省成本。您可以使用 AWS Resource Access Manager （AWS RAM） 來共用其他常用資源，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 建立資源一次，並與其他帳戶共用。

組織應有效地標記共用成本，並確認大部分成本已標記或分配。如果未有效地分配共用成本，而且沒有人負責共用成本管理，則共用雲端成本可能會螺旋式上升。您應該知道在資源、工作負載、團隊或組織層級產生了哪些成本，因為相較於達成的業務成果，這項知識可增強您對適用層級所提供的價值的了解。最終，組織可以從共用雲端基礎設施的成本節約中獲益。鼓勵共用雲端資源的成本分配，以最佳化雲端支出。

實作步驟

- 評估現有資源：檢閱針對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，如果業務邏輯或技術需求允許，請考慮現有平台。
- 使用 中的資源共用 AWS RAM 並相應限制：使用 與組織內的其他 AWS 帳戶 AWS RAM 共用資源。共用資源時，無需在多個帳戶中重複資源，這樣可將資源維護的作業負擔降到最低。此流程也可協助您安全地與帳戶中的角色和使用者以及其他 AWS 帳戶共用您所建立的資源。
- 標記資源：標記屬於成本報告的候選資源，並在成本分類中將其分類。啟用這些與成本相關的資源標籤進行成本分配，以提供 AWS 資源用量的可見性。專注於在成本和用量可見性方面建立適當的精細程度，並透過成本分配報告和KPI追蹤來影響雲端消耗行為。

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？

- AWS 您可以與 搭配使用的服務 AWS Organizations
- 可共用 AWS 的資源
- AWS 成本和用量 （ 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：

- 如何退款共用服務： AWS Transit Gateway 範例
- 如何使用 建置 Savings Plans 的扣款/顯示模型 CUR
- 使用VPC共用進行符合成本效益的多帳戶微服務架構
- EKS使用 AWS 分割成本分配資料改善 Amazon 的成本可見性
- AWS Batch 使用 AWS 分割成本分配資料改善 Amazon ECS和 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 執行定價模型分析
- COST07-BP02 根據成本選擇區域
- COST07-BP03 選取具有成本效益條款的第三方協議
- COST07-BP04 為此工作負載的所有元件實作定價模型
- COST07-BP05 在管理帳戶層級執行定價模型分析

COST07-BP01 執行定價模型分析

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曝險等級：高

實作指引

AWS 有多個定價模型，可讓您以最符合成本效益的方式支付資源，以符合組織的需求，並取決於產品。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

隨需執行個體允許您按照時數或秒數 (最少 60 秒) 支付運算或資料庫容量的費用，視您執行的執行個體而定，而無須支付長期的固定款項或預付款。

Savings Plans 是一種靈活的定價模型，可在 Amazon EC2、Lambda 和 AWS Fargate 用量上提供低價，以換取承諾一年或三年內一致用量（以每小時美元為單位）。

Spot 執行個體是一種 Amazon EC2定價機制，可讓您以折價每小時費率（最高 90% 的隨需價格）請求備用運算容量，而無需預先承諾。

透過預付容量，預留執行個體可讓您獲得高達 75% 的折扣。如需詳細資訊，請參閱透過預留來最佳化成本。

您可能會選擇為生產、品質和開發環境的相關資源納入 Savings Plans。或者，由於沙盒資源僅在需要時開啟，因此您可以為該環境中的資源選擇隨需模型。使用 Amazon Spot 執行個體來降低 Amazon EC2成本，或使用 Compute Savings Plans 來降低 Amazon EC2、 Fargate 和 Lambda 成本。AWS Cost Explorer 建議工具透過 Savings Plans 提供承諾折扣的機會。

如果您EC2過去曾為 Amazon 購買預留執行個體，或已在組織內部建立成本分配實務，則可以暫時繼續使用 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。您可以在 中重新整理 Savings Plans（SP）建議 AWS Cost Management ，以隨時產生新的 Savings Plans 建議。使用預留執行個體（RI）來降低 Amazon RDS、Amazon Redshift ElastiCache、Amazon 和 Amazon OpenSearch Service 的成本。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 執行個體用於具備容錯能力和靈活性的各種應用程式。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 和其他彈性工作負載。

分析您的 Amazon EC2和 Amazon RDS執行個體，在您不使用時是否可以關閉它們（非營業時間和週末）。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。當 Amazon Redshift 叢集或 Amazon EC2和 Amazon RDS Instance 停止時，運算帳單會停止，且只會收取儲存費用。

請注意，隨需容量保留（ODCR）不是定價折扣。無論您是否以預留容量執行執行個體，都需要支付按隨需費率計算的容量保留費用。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。ODCRs 不必與長期承諾相關聯，因為當您不再需要它們時，它們可以取消，但也可以從 Savings Plans 或預留執行個體提供的折扣中獲益。

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 機群的候選項目。
  - Well-Architected 實驗室：Cost Explorer
  - Well-Architected 實驗室：成本視覺化
- 檢閱現有定價合約：針對長期需求，檢閱目前的合約或承諾。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。企業協議可讓客戶量身打造最符合其需求的協議。對於長期承諾，請考慮特定執行個體類型、執行個體系列 AWS 區域和可用區域的預留定價折扣、預留執行個體或 Savings Plans。
- 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。要驗證您是否以所需的折扣和風險實作了正確的建議，請遵循 Well-Architected 實驗室。

資源

相關文件：

- 存取預留執行個體建議
- 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：定價模型

COST07-BP02 根據成本選擇區域

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，有助於讓您針對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

AWS 雲端 基礎設施是全球通用的，託管於全球多個位置，並以 AWS 區域、可用區域、本機區域、AWS Outpost 和 Wavelength 區域為基礎建置。區域是世界上的實體位置，每個區域都是 AWS 具有多個可用區域的個別地理區域。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、聯網和連線能力。

每個 都在當地市場條件下 AWS 區域 運作，而且每個區域的資源定價因土地、光纖、電力和稅金成本的差異而不同。您可以選擇特定區域以操作解決方案的元件或全部，以便以最低價格於全球執行。使用 AWS Calculator，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、安全和業務需求方面沒有義務，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亞太區域（雪梨）（ap-southwest-2），並且沒有使用其他區域的限制（例如資料隱私權、安全性），則在美國東部（維吉尼亞北部）（ ）部署非關鍵（開發和測試us-east-1）Amazon EC2執行個體的成本將更低。

	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 6 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：分析目前區域中的工作負載成本。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，請考慮資料傳輸成本。將資料存放在接近客戶與資源之處。選擇 AWS 區域 資料流和資料傳輸最少的成本較低。根據您的資料傳輸業務需求，您可以使用 Amazon CloudFront、AWS PrivateLink、 AWS Direct Connect和 AWS Virtual Private Network來降低聯網成本、改善效能並增強安全性。

資源

相關文件：

- 存取預留執行個體建議

- 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 選取具有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曝險等級：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，如果您使用沒有提供利益的其他 服務，提供 Amazon 建議EC2並收取整個帳單一定百分比費用的解決方案可能會變得更昂貴。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

- 分析第三方協議與條款：審核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：

- 存取預留執行個體建議
- 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 為此工作負載的所有元件實作定價模型

永久執行的資源應使用預留容量，例如 Savings Plans 或預留執行個體。設定短期容量以使用 Spot 執行個體或 Spot 機群。隨需執行個體僅用於無法中斷且執行時間不夠長，以及不適合使用預留容量的短期工作負載 (介於 25% 到 75% 之間的時間，視資源類型而定)。

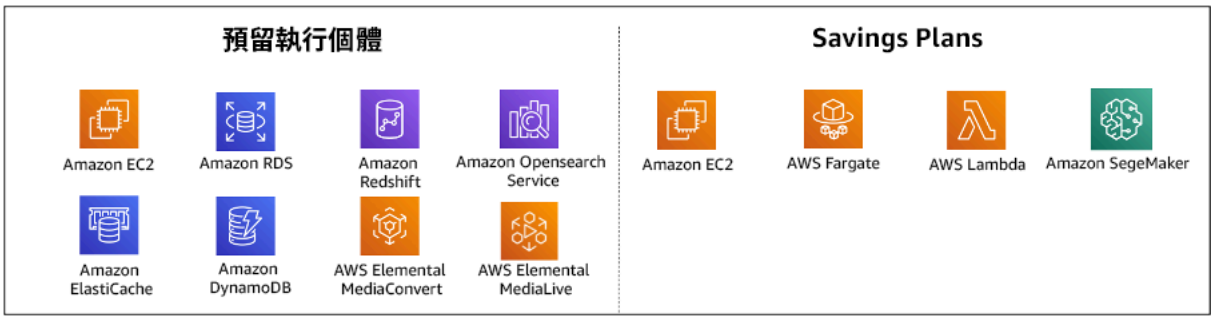
未建立此最佳實務時的曝險等級：低

實作指引

為了提高成本效益， 會根據您過去的使用 AWS 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。您可以使用這些服務做為隨需、Spot 或在特定期間內做出承諾，並透過預留執行個體 （ RIs ） 和 Savings Plans （ ） 降低隨需成本SPs。您不僅需要了解每個工作負載元件和多個 AWS 服務，還要承諾這些服務的折扣、購買選項和 Spot 執行個體，以最佳化您的工作負載。

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。此範例工作負載包含多個 AWS 服務，例如 Amazon Route 53 AWS WAF、Amazon CloudFront、Amazon EC2執行個體、Amazon RDS執行個體、Load Balancer、Amazon S3 儲存體和 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。其中有些可能符合 RIs或的資格SPs，有些可能只能隨需提供。如下圖所示，部分 AWS 服務可以使用 RIs或 遞交SPs。



AWS 使用預留執行個體和 Savings Plans 遞交的服務

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。如果這是您的第一次承諾購買，請選擇清單中的前五或十個建議，然後監控和分析下個月或兩個月的結果。AWS Cost Management Console 會引導您完成整個程序。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，並為其餘需求配置資源。
- 工作負載審查週期：實作工作負載的審查週期，特別分析定價模型涵蓋範圍。一旦工作負載達到所需的涵蓋範圍，請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
- 存取預留執行個體建議
- 如何購買預留執行個體
- 執行個體購買選項

- Spot 執行個體
- AWS 其他服務的預留模型
- 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：

- 購買 Savings Plans 前應考量哪些事項？
- 如何使用 Cost Explorer 來分析我的支出和用量？

COST07-BP05 在管理帳戶層級執行定價模型分析

查看計費和成本管理工具，並檢視承諾和保留的建議折扣，在管理帳戶層級執行定期分析。

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。

雖然在管理帳戶層級購買可最佳化在許多情況下的最大節省金額，但在某些情況下，您可能會考慮SPs在連結帳戶層級購買，例如您希望折扣先套用到該特定連結帳戶中的使用。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。SPs 在管理帳戶層級查看所有連結帳戶，並在折扣最高的地方套用節省。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。如果有多種用量具有相同的節省金額百分比，則 Savings Plans 會套用至第一個具有最低 Savings Plans 費率的用量。Savings Plans 將繼續套用，直至沒有剩餘用量或您的承諾用盡為止。任何剩餘用量均按隨需費率收費。您可以在 AWS Cost Management 中重新整理 Savings Plans 建議，以隨時產生新的 Savings Plans 建議。

分析執行個體的彈性後，您可以採納建議的承諾。透過使用潛在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，並與您的業務需求保持一致，以找出擁有權的總成本和成本最佳化機會，來建立成本模型。

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 定價如何運作？
- 執行個體購買選項
- Saving Plan 概觀
- Saving Plan 建議

- 存取預留執行個體建議
- 了解 Savings Plans 建議
- Savings Plans 如何套用至您的 AWS 用量
- 具有合併帳單功能的 Savings Plans
- 開啟共享的預留執行個體和 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 選取元件以最佳化資料傳輸成本
- COST08-BP03 實作 服務以降低資料傳輸成本

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曝險等級：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。中的資料傳輸費用 AWS 取決於流量的來源、目的地和數量。在設計階段考慮這些費用能夠讓您省下成本。了解工作負載中資料傳輸的發生位置、傳輸成本及其相關效益，對於準確估算總擁有成本（ ）非常重要TCO。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聯網最佳實務，以將潛在的資料傳輸成本最佳化。

AWS Pricing Calculator 可協助您查看特定 AWS 服務和預期資料傳輸的預估成本。如果您已經有執行中的工作負載（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）來了解和模擬資料傳輸成本。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

- 確定需求：在來源與目的地之間規劃的資料傳輸的主要目標和業務需求是什麼？ 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。
- 識別來源和目的地：資料傳輸的資料來源和目的地是什麼，例如在 內 AWS 區域、到 AWS 服務或到網際網路？
  - 內的資料傳輸 AWS 區域
  - 之間的資料傳輸 AWS 區域
  - 資料傳出到網際網路
- 識別資料分類：此資料傳輸的資料分類為何？ 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？
- 識別要使用 AWS 的服務或工具：此資料傳輸使用哪些 AWS 服務？ 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：使用先前建立的資料傳輸模型 AWS Pricing 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。

- 將成本與結果連結：對於產生的每筆資料傳輸成本，請指定其為工作負載達到的結果。如果在元件之間傳輸，可能是用於解耦，如果在可用區域之間傳輸，則可能是用於備援。
- 建立資料傳輸模型：收集所有資訊後，為多個使用案例和不同的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：

- 監控和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 網路規範指南

COST08-BP02 選取元件以最佳化資料傳輸成本

選擇所有元件，並設計架構以降低資料傳輸成本。這包括使用 （WAN） 最佳化和多可用區域 （AZ） 組態等 wide-area-network元件

未建立此最佳實務時的曝險等級：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最佳化和應用程式最佳化來減少元件之間傳輸的資料量。

將資料傳輸至 或在 內時 AWS 雲端，請務必根據不同的使用案例、資料的性質和可用的網路資源來了解目的地，以選擇適當的 AWS 服務來最佳化資料傳輸。 AWS 提供針對各種資料遷移需求量身打造的一系列資料傳輸服務。根據組織內的業務需求，選擇正確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 內使用VPC端點 AWS：VPC端點允許您的 VPC和 支援 AWS 的服務之間的私有連線。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至網際網路或 外部的服務 VPC。檢查傳送最多流量的NAT閘道後方資源是否與NAT閘道位於相同的可用區域中。如果不是，請在與 資源相同的可用區域中建立新的NAT閘道，以減少跨可用區域資料傳輸費用。
- 使用 AWS Direct Connect AWS Direct Connect 會略過公有網際網路，並在內部部署網路與 之間建立直接的私有連線 AWS。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。
- 避免跨區域邊界傳輸資料：在 AWS 區域 （從一個區域到另一個區域） 之間傳輸資料通常會產生費用。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流程日誌來擷取資料傳輸和網路用量的詳細資訊。分析 中擷取的網路流量資訊VPCs，例如往返網路介面的 IP 地址或範圍。
- 分析您的網路用量：使用計量和報告工具 AWS Cost Explorer，例如 、 CUDOS Dashboards 或 CloudWatch 來了解工作負載的資料傳輸成本。

實作步驟

- 選擇用於資料傳輸的元件：使用 COST08-BP01 執行資料傳輸建模 中所述的資料傳輸模型，專注於資料傳輸成本最高的位置或工作負載用量變更時資料傳輸成本最高的位置。尋找替代架構或其他元件，以消除或降低資料傳輸需求 (或降低其成本)。

資源

相關的最佳實務：

- COST08-BP01 執行資料傳輸建模
- COST08-BP03 實作 服務以降低資料傳輸成本

相關文件：

- 雲端資料遷移

- AWS 快取解決方案
- 使用 Amazon 更快交付內容 CloudFront

相關範例：

- 常見架構的資料傳輸成本概觀
- AWS 網路最佳化秘訣
- 使用 Apache Parquet 格式的 VPC Flow Logs 來最佳化效能並降低網路分析的成本

COST08-BP03 實作 服務以降低資料傳輸成本

實作服務以減少資料傳輸。例如，使用邊緣位置或內容交付網路（CDN）將內容交付給最終使用者、在應用程式伺服器或資料庫前建置快取層，並使用專用網路連線，而不是VPN用於連線至雲端。

未建立此最佳實務時的曝險等級：中

實作指引

有各種 AWS 服務可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 是全球內容交付網路，可提供低延遲和高傳輸速度的資料。其快取位於全球節點的資料，能減輕您的資源所受的負載。透過使用 CloudFront，您可以減少管理工作，以最低延遲將內容交付給全球大量使用者。如果您計劃隨著時間增加 CloudFront 用量，安全節省套件可協助您節省高達 30% 的用量。
- AWS Direct Connect 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 端點允許透過私有聯網在服務之間 AWS 進行連線，並可用於降低公有資料傳輸和NAT閘道成本。閘道VPC端點沒有每小時費用，並支援 Amazon S3 和 Amazon DynamoDB 。介面VPC端點由提供，AWS PrivateLink並收取每小時費用和每 GB 用量成本。
- NAT 閘道提供內建擴展和管理，相較於獨立NAT執行個體，可降低成本。將NAT閘道放置在與高流量執行個體相同的可用區域中，並考慮將VPC端點用於需要存取 Amazon DynamoDB 或 Amazon S3 的執行個體，以降低資料傳輸和處理成本。

- 使用具有運算資源AWS Snow Family的裝置在 edge 收集和處理資料。 AWS Snow Family devices （ Snowcone、Snowball 和 Snowmobile ） 可讓您將 PB 的資料移至具 AWS 雲端 成本效益且離線的狀態。

實作步驟

- 實作服務：使用資料傳輸建模和檢閱VPC流程日誌，根據您的服務工作負載類型選取適用的 AWS 網路服務。查看成本最高和磁碟區流量最大的情況。檢閱 AWS 服務，並評估是否有減少或移除傳輸的服務，特別是聯網和內容交付。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 Security Savings 套件

相關影片：

- 監控和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 從成本和用量報告中深入了解 AWS 資料傳輸詳細資訊 QuickSight
- 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
- 使用 Amazon CloudFront 功能以成本最佳化您的 AWS 架構
- 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？

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。任一方向的偏斜使用率指標都會對您的組織造成負面影響，包括營運成本（因過度使用而降低效能）或浪費 AWS 支出（由於過度佈建）。

最佳實務

- COST09-BP01 執行工作負載需求的分析
- COST09-BP02 實作緩衝區或限流來管理需求
- COST09-BP03 動態供應資源

COST09-BP01 執行工作負載需求的分析

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。在足夠長的時間內執行分析，以納入任何季節性差異，例如 end-of-month處理或假日峰值。

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. 資源使用率和效能指標 ：分析資源如何隨時間 AWS 使用。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. 服務層級協議 （SLAs）：將實際效能與 進行比較SLAs，以確保合規性並識別需要改進的領域。

您可以使用 Amazon CloudWatch 收集和追蹤指標、監控日誌檔案、設定警示，並自動回應資源的變更 AWS 。您也可以使用 Amazon CloudWatch 來全面了解資源使用率、應用程式效能和運作狀態。

使用 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 來符合需求的增加或減少。

最後，您可以將 AWS Cost Explorer或 Amazon QuickSight 與 AWS Cost and Usage Report （CUR）檔案或應用程式日誌搭配使用，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、日誌檔案和監控資料來深入了解工作負載的使用方式。分析工作負載的完整週期，並收集任何季節性變更的資料，例如 end-of-month 或 end-of-year 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 入門 SQS
- AWS Cost Explorer
- Amazon QuickSight

相關影片：

相關範例：

- 監控、追蹤和分析以實現成本最佳化
- 搜尋和分析登入 CloudWatch

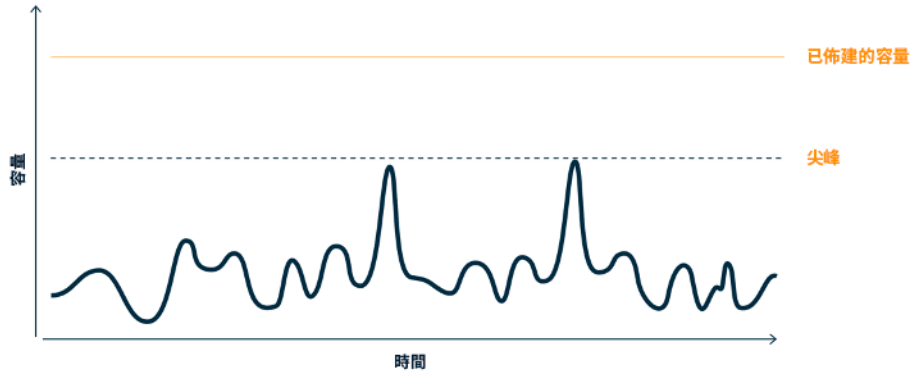
COST09-BP02 實作緩衝區或限流來管理需求

緩衝和限流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限流機制。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曝險等級：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。實作限流的優點是限制最大資源量和工作負載成本。在 中 AWS，您可以使用 Amazon API Gateway 實作限流。

基於緩衝區：基於緩衝區的方法會使用生產者 (將訊息傳送至佇列的元件)、取用者 (從佇列接收訊息的元件) 和佇列 (保留訊息) 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 中 AWS，您可以選擇多個 服務來實作緩衝方法。Amazon Simple Queue Service （ Amazon SQS ） 是一種受管服務，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。

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。
- 實作緩衝區或限流：在工作負載中實作緩衝區或限流。Amazon Simple Queue Service（Amazon SQS）等佇列可以為您的工作負載元件提供緩衝。Amazon API Gateway 可為工作負載元件提供限流功能。

資源

相關的最佳實務：

- SUS02-BP06 實作緩衝或調節，以扁平需求曲線
- REL05-BP02 節流請求

相關文件：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon API Gateway
- Amazon Simple Queue Service
- Amazon 入門 SQS
- Amazon Kinesis

相關影片：

- 為分散式應用程式選擇正確的訊息傳遞服務

相關範例：

- 管理和監控工作負載中的API限流
- 使用 API Gateway REST API 大規模限流分層多租戶
- 使用 Amazon API Gateway 在多租戶 Amazon EKS SaaS 解決方案中啟用分層和限流
- 使用佇列與訊息進行應用程式整合

COST09-BP03 動態供應資源

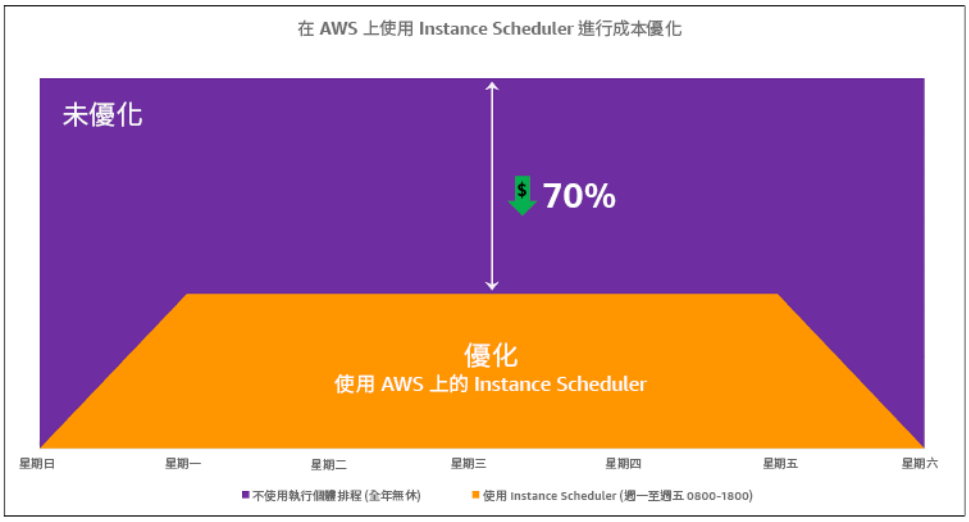
資源會按計劃進行佈建。這可以是以需求為基礎 (例如，透過自動調整規模)，或是以時間為基礎，其中需求可預測，並且根據時間提供資源。這些方法可盡量減少過度佈建或佈建不足的數量。

未建立此最佳實務時的曝險等級：低

實作指引

AWS 客戶有多種方式可以增加其應用程式可用的資源，並提供資源以滿足需求。其中一個選項是使用 AWS Instance Scheduler，它可自動啟動和停止 Amazon Elastic Compute Cloud （Amazon EC2） 和 Amazon Relational Database Service （Amazon RDS） 執行個體。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 可讓您在定義的時間設定 Amazon 和 Amazon RDS執行個體的停止EC2和啟動，以便您可以在一致的時間模式內滿足對相同資源的需求，例如每天使用者在早上八點存取 Amazon EC2執行個體，他們在晚上六點後不需要。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 執行個體排程器進行成本最佳化。

您也可以使用 AWS Systems Manager 快速設定，透過簡單的使用者介面 （UI），輕鬆地在帳戶和區域間設定 Amazon EC2執行個體的排程。您可以使用 AWS 執行個體排程器來排程 Amazon EC2或 Amazon RDS執行個體，並且可以停止和啟動現有的執行個體。不過，您無法停止和啟動屬於 Auto Scaling 群組 （ASG） 或管理 Amazon Redshift 或 Amazon OpenSearch Service 等服務的執行個體。Auto Scaling 群組對群組中的執行個體有自己的排程，並且會建立這些執行個體。

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。它是一項完全受管且免費的服務，可擴展應用程式與 Amazon EC2執行個體和 Spot Fleets、Amazon ECS、Amazon DynamoDB 和 Amazon Aurora 整合的容量。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：

- 隨時維持目前執行個體層級
- 手動擴展
- 依據排程擴展
- 依據需求擴展
- 使用預測擴展

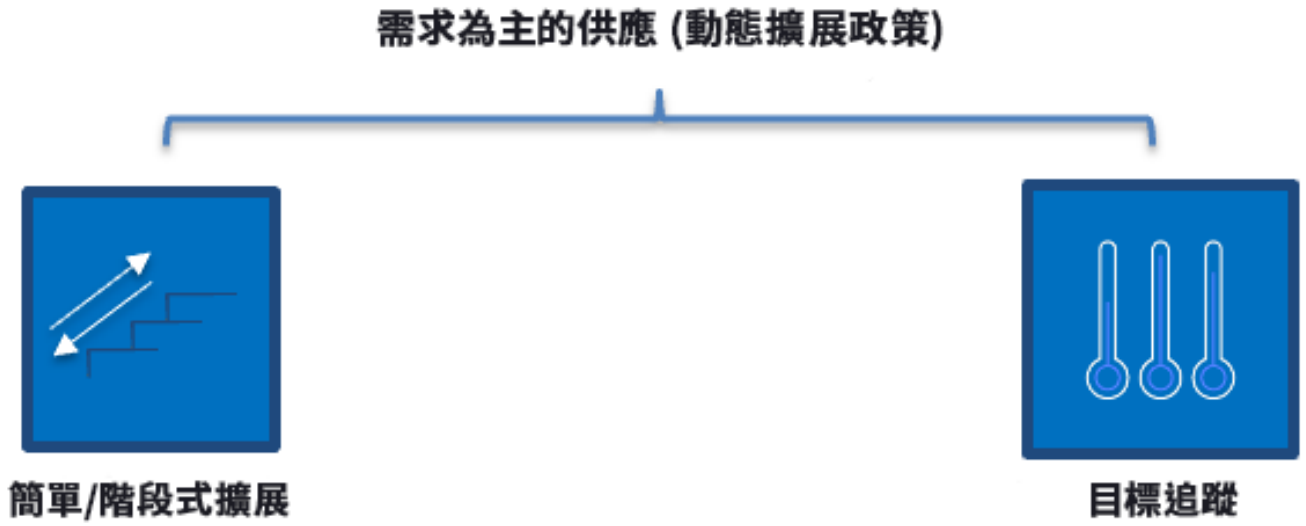
Auto Scaling 政策有所不同，可分類為動態和排程擴展政策。動態政策是手動或動態擴展，屬於排程或預測擴展。您可以使用擴展政策來進行動態、排程和預測擴展。您也可以使用 Amazon CloudWatch 的指標和警示來觸發工作負載的擴展事件。建議您使用啟動範本，它允許您存取最新功能和改善項目。即便使用啟動組態，也並非所有 Auto Scaling 功能都可用。例如：您無法建立同時啟動 Spot 及隨需執行個體的 Auto Scaling 群組或指定多個執行個體類型的群組。您必須使用啟動範本來設定這些功能。使用啟動範本時，建議您對每個範本進行版本控制。藉由啟動範本的版本控制，您可以建立一組完整的參數子集，之後可以重複使用子集來建立相同啟動範本的其他版本。

您可以使用 AWS Auto Scaling 或 ，AWS APIs在程式碼中使用或SDKs合併擴展。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。為了動態遵循此最佳實務並為組織提供資源，您應該了解 中的水平和垂直擴展 AWS 雲端，以及 Amazon EC2執行個體上執行的應用程式性質。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。

彈性負載平衡 (Elastic Load Balancing) 可跨多個資源分配需求以協助您進行擴展。使用 ASG和 Elastic Load Balancing 時，您可以透過最佳方式路由流量來管理傳入的請求，以便在 Auto Scaling 群組中不會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。

典型指標可以是標準 Amazon EC2指標，例如CPU使用率、網路輸送量和 Elastic Load Balancing 觀察到的請求和回應延遲。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。對於需求型供應，使用 APIs 或服務功能以程式設計方式改變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。

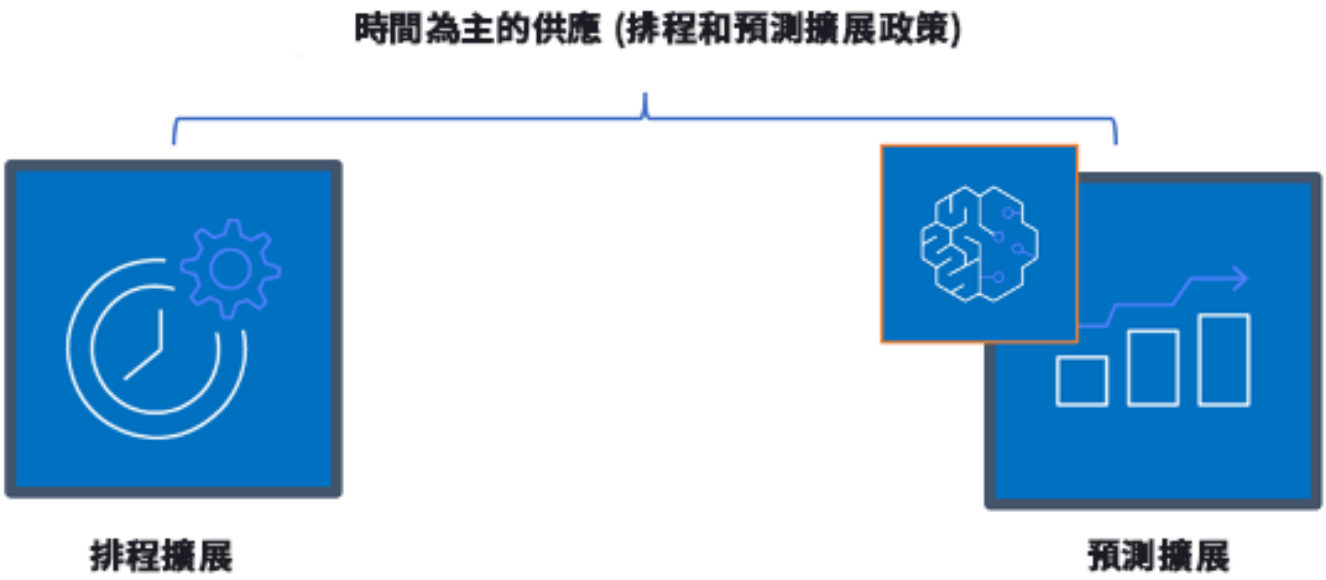


需求為主的動態擴展政策

- 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。您也可以在 Auto Scaling 群組中使用屬性型執行個體類型選取（ABS）策略，這可讓您將執行個體需求表達為一組屬性，例如 v CPU、記憶體和儲存體。這也可讓您在發佈較新一代的執行個體類型時自動使用，並使用 Amazon EC2 Spot 執行個體存取更廣泛的容量範圍。Amazon EC2 Fleet 和 Amazon EC2 Auto Scaling 選擇並啟動符合指定屬性的執行個體，無需手動挑選執行個體類型。

您也可以利用 AWS APIs和 SDKs AWS CloudFormation ，視需要自動佈建和停用整個環境。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。您可以使用 APIs來擴展環境中的資源大小（垂直擴展）。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。此技術也可以套用至其他資源，例如 Amazon EBS Elastic Volumes，這些資源可以修改以增加大小、調整效能（IOPS）或使用時變更磁碟區類型。

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。若要設定以時間為基礎的排程，您可以使用排程擴展的預測擴展，根據預期或可預測的負載變更預先增加 Auto Scaling 群組中的 Amazon EC2執行個體數量。
- 設定預測擴展：預測擴展可讓您在流量流量的每日和每週模式之前，增加 Auto Scaling 群組中的 Amazon EC2執行個體數量。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。使用分析並設定 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。您可以在單一 Auto Scaling 群組內啟動和自動擴展隨需執行個體和 Spot 執行個體組成的機群。除了獲得使用 Spot 執行個體的折扣之外，您還可以使用預留執行個體或 Savings Plan，以獲得定期隨需執行個體定價的折扣費率。所有這些因素組合可協助您最佳化 Amazon EC2執行個體的成本節省，並協助您取得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 入門 SQS
- 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 執行個體排程器

相關範例：

- Amazon EC2 Fleet Auto Scaling 的屬性型執行個體類型選擇
- 使用已排程的擴展，針對成本最佳化 Amazon Elastic Container Service
- 使用 Amazon EC2 Auto Scaling 進行預測擴展
- 如何搭配 使用執行個體排程器 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 推出新的服務和功能，最佳實務是檢閱您現有的架構決策，以驗證它們是否繼續最具成本效益。

最佳實務

- COST10-BP01 開發工作負載審核程序
- COST10-BP02 定期檢閱和分析此工作負載

COST10-BP01 開發工作負載審核程序

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曝險等級：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。與績效支柱審查和可靠性支柱審查流程類似，識別、驗證及優先處理最佳化和改進活動以及問題修復，並將其納入您的待辦項目中。

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。
- 定義審查徹底性：定義審查工作負載或工作負載元件所需的工作量。與審查頻率類似，這需在多個因素之間取得平衡。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 支援主動式服務
- 工作負載的定期SAP工作負載檢閱

COST10-BP02 定期檢閱和分析此工作負載

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曝險等級：中

實作指引

AWS 不斷新增新功能，因此您可以使用最新的技術更快地進行實驗和創新。AWS 新功能詳細說明 AWS 了如何執行此操作，並在發佈 AWS 服務、功能和區域擴展公告時提供快速概觀。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。為了實現新 AWS 服務和功能的優點，您可以檢閱工作負載並視需要實作新服務和功能。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 AWS 服務。例如，您可以檢閱工作負載，並使用 Amazon Simple Email Service 取代簡訊元件。這消除了營運和維護執行個體叢集的成本，同時以較低的成本提供所有功能。

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。檢閱 上的這是我的架構影片 AWS ，以了解其他客戶的架構設計、挑戰及其解決方案。檢查 All-In 系列，了解 AWS 服務和客戶案例的真實應用程式。也可以觀看回歸基礎影片系列，其中說明、檢查和細分基本的雲端架構模式最佳實務。另一個來源是如何建置此影片，其設計旨在協助人們提出有關如何使用 AWS 服務實現最小可行產品（MVP）的重大想法。對於來自世界各地的建置者來說，這是一種從經驗豐富的 AWS 解決方案架構師取得架構指導的強烈想法。最後，可以檢閱入門資源材料，其中包含逐步教學課程。

在開始審查程序之前，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 - 這是我的架構
- AWS - 返回基本作業要點
- AWS - All-In 系列
- 如何建置此方法

COST 11. 如何評估工作的成本？

最佳實務

- COST11-BP01 執行操作的自動化

COST11-BP01 執行操作的自動化

評估雲端上的營運成本，著重於量化管理任務、部署中節省的時間和精力，並透過自動化降低人為錯誤、法規遵循和其他操作的風險。評估營運工作所需的時間和相關成本，並實作管理任務的自動化，以盡可能地減少手動工作量。

未建立此最佳實務時的曝險等級：低

實作指引

將操作自動化可減少人工作業的頻率、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而穩定的體驗。您可以將基礎設施資源從手動操作任務中解放出來，並將其用於價值更高的任務與創新，這可提升商業價值。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、快速且具有成本效益，並具有最低的風險和最大的可靠性。

首先，考慮整體營運成本，根據所需的工作量確定操作活動的優先順序。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？ 透過考慮運營和管理成本來確定人為行動的總成本。排定管理任務的自動化優先順序，以減少人力。

審查工作量應反映潛在的效益。例如，檢查手動執行任務所花費的時間 (對照自動執行)。優先考慮自動化重複、高價值、耗時且複雜的活動。具有高價值或高人為錯誤風險的活動通常是開始自動化的起點，因為這類風險通常會產生不必要的額外營運成本 (例如營運團隊的加班費)。

使用 AWS Systems Manager 或 等自動化工具 AWS Config 來簡化操作、合規、監控、生命週期和終止程序。透過 AWS 服務、工具和第三方產品，您可以自訂實作的自動化，以符合您的特定需求。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：持續稽核您的 AWS 用量，以簡化風險和合規評估
- AWS Backup：集中管理和自動化資料保護。
- AWS Config：配置計算資源，評定、審核、評估組態和資源清單。
- AWS CloudFormation：使用基礎設施即程式碼啟動高可用性資源。
- AWS CloudTrail：IT 變更管理、合規性和控制。
- Amazon EventBridge Schedule 事件和觸發程序 AWS Lambda 以採取動作。
- AWS Lambda：使用 事件觸發重複程序，或使用 以固定排程執行這些程序，以自動化重複程序 AWS EventBridge。
- AWS Systems Manager：啟動和停止工作負載、修補作業系統、自動化組態和持續管理。
- AWS Step Functions：排程工作並自動化工作流程。
- AWS Service Catalog：範本使用，具有合規性和控制的基礎設施即程式碼。

如果您想要使用 AWS 產品和服務立即採用自動化，而且您的組織中沒有技能，請聯絡 AWS Managed Services （AMS）、AWS 專業服務 或 AWS 合作夥伴，以增加自動化的採用並改善您在雲端中的卓越營運。

AWS Managed Services （ AMS ） 是一種代表企業客戶和合作夥伴操作 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 使用具有自動化的企業雲端操作模型，讓您符合組織需求、更快地進入雲端，並降低持續的管理成本。

AWS 專業服務也可以協助您達成所需的業務成果，並使用 將操作自動化 AWS。它們可協助客戶部署已針對雲端進行優化的自動化、穩健而靈活的 IT 營運及管控能力。如需詳細的監控範例和建議的最佳實務，請參閱《卓越營運支柱》白皮書。

實作步驟

- 建置一次並部署許多 ：使用 infrastructure-as-code 例如 CloudFormation AWS SDK、 或 AWS CLI 部署一次，並在類似環境或災難復原案例中多次使用。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可縮短部署許多熱門企業工作負載的時間。 AWS Launch Wizard 會依照 AWS 最佳實務，引導您完成企業工作負載的大小、組態和部署。您也可以使用 Service Catalog ，這可協助您建立和管理可用於 的 infrastructure-as-code已核准範本， AWS 以便任何人都能探索已核准的自助式雲端資源。
- 自動化持續合規性：考慮根據預先定義的標準，自動化記錄的組態的評估和修復。當您 AWS Organizations 結合 和 的功能 AWS Config 時AWS CloudFormation，您可以為數百個成員帳戶大規模有效地管理和自動化組態合規。您可以檢閱 AWS 資源之間的組態和關係變更，並深入了解資源組態的歷史記錄。
- 自動化監控任務 AWS 提供各種可用來監控服務的工具。您可以設定這些工具來自動執行監控工作。建立和實作監控計畫來收集工作負載的全面監控資料，以便在出現多點故障時能更輕鬆地偵錯。例如，您可以使用自動化監控工具來觀察 Amazon，EC2並在系統狀態檢查、執行個體狀態檢查和 Amazon CloudWatch 警示發生錯誤時回報給您。
- 自動化維護和操作：自動執行例行操作，無需人為介入。使用 AWS 服務和工具，您可以選擇要實作哪些 AWS 自動化並根據您的特定需求自訂。例如，使用 EC2 Image Builder 建置、測試和部署虛擬機器和容器映像，以便在內部部署 AWS 或內部部署使用，或使用 修補EC2執行個體 AWS SSM。如果所需的動作無法使用 AWS 服務完成，或者您需要更複雜的動作搭配篩選資源，請使用 AWS Command Line Interface （ AWS CLI ） 或 AWS SDK 工具來自動化操作。 AWS CLI 可讓您使用指令碼來自動化整個控制和管理 AWS 服務的程序，而無需使用 AWS Management Console。選取您偏好的 AWS SDKs 以與 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 。
- 使用自動化建立持續的生命週期：建立並保留成熟的生命週期政策非常重要，這不僅適用於法規或備援，還適用於成本最佳化。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 自動化建立、保留和刪除EBS快照和 EBS後端 AMIs。

- 刪除不必要的資源：在沙盒或開發中累積未使用的資源相當常見 AWS 帳戶。開發人員會在正常開發週期中建立並試驗各種服務和資源，然後在不再需要這些資源時不會刪除它們。未使用的資源可能會為組織帶來不必要的、有時甚至很高的成本。刪除這些資源可以降低操作這些環境的成本。如果不確定，請確保不再需要資料或已備份。可以使用 AWS CloudFormation 來清理已部署的堆疊，這會自動刪除範本中定義的大部分資源。或者，您可以使用 aws-nuke 等工具建立刪除 AWS 資源的自動化。

資源

相關文件：

- 中的現代化操作 AWS 雲端
- 用於自動化的AWS Services
- 基礎設施和自動化
- AWS Systems Manager 自動化
- 自動和手動監控
- AWS SAP管理和操作的自動化
- AWS Managed Services
- AWS Professional Services

相關影片：

- 在 中大規模自動化持續合規 AWS
- AWS Backup 示範：跨帳戶和跨區域備份
- 針對 Amazon EC2執行個體的修補

相關範例：

- 重塑自動化操作 (第一部分)
- 重塑自動化操作 (第二部分)
- 使用 aws-nuke 自動刪除 AWS 資源
- 使用 AWS Config 和 刪除未使用的 Amazon EBS磁碟區 AWS SSM
- 自動化大規模的持續合規 AWS
- 使用 的 IT 自動化 AWS Lambda

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。可以在永續性支柱白皮書中找到實作指引。

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 如何為工作負載選取區域？

SUS 1 如何為工作負載選取區域？

工作負載的 區域選擇會大幅影響其 KPIs，包括效能、成本和碳足跡。若要有效改善這些 KPIs，您應該根據業務需求和永續性目標，為工作負載選擇區域。

最佳實務

- SUS01-BP01 根據業務需求和永續性目標選擇區域

SUS01-BP01 根據業務需求和永續性目標選擇區域

根據您的業務需求和永續性目標，為您的工作負載選擇區域，以最佳化其 KPIs，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曝險等級：中

實作指引

AWS 雲端 是一個持續擴展的區域和存在點網路 PoP ），具有將它們連結在一起的全域網路基礎設施。工作負載的 區域選擇會大幅影響其 KPIs，包括效能、成本和碳足跡。若要有效改善這些 KPIs，您應該根據您的業務需求和永續性目標，為您的工作負載選擇區域。

實作步驟

- 遵循以下步驟，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。
  - 測試最終使用者位置與每個 之間的網路延遲 AWS 區域。
- 選擇 Amazon 可再生能源專案附近的區域，以及電網公佈的碳強度低於其他位置 (或區域) 的區域。
  - 根據溫室氣體協定 （以市場和位置為基礎的方法），識別您的相關永續性指導方針，以追蹤和比較 year-to-year碳排放。
  - 根據您用來追蹤碳排放的方法來選擇區域。如需根據永續性指引來選擇區域的詳細資訊，請參閱 How to select a Region for your workload based on sustainability goals。

資源

相關文件：

- 了解您的碳排放估算
- 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發展創新

- AWS re：Invent 2023 - 永續架構：過去、現在和未來
- AWS re：Invent 2022 - 提供永續、高效能的架構
- AWS re：Invent 2022 - 永續架構並減少 AWS 碳足跡
- AWS re：Invent 2022 - AWS 全球基礎設施的永續性

因應需求

問題

- SUS 2 如何將雲端資源與需求保持一致？

SUS 2 如何將雲端資源與需求保持一致？

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設施
- SUS02-BP02 SLAs與永續發展目標保持一致
- SUS02-BP03 停止建立和維護未使用的資產
- SUS02-BP04 根據其聯網需求最佳化工作負載的地理定位
- SUS02-BP05 針對執行的活動最佳化團隊成員資源
- SUS02-BP06 實作緩衝或調節，以扁平需求曲線

SUS02-BP01 動態擴展工作負載基礎設施

使用雲端的彈性並動態擴展您的基礎設施，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設施。
- 您一律手動擴展基礎設施。

- 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曝險等級：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、容器和函數提供彈性機制，可結合自動擴展或作為 服務的功能。 AWS 提供各種自動擴展機制，以確保工作負載在低使用者負載期間可以快速輕鬆地擴展。以下是自動擴展機制的幾個範例：

自動擴展機制	在哪裡使用
Amazon EC2 Auto Scaling	使用 來確認您有正確數量的 Amazon EC2執行個體，可用於處理應用程式的使用者負載。
Application Auto Scaling	使用 自動將個別 AWS 服務的資源擴展至 Amazon 之外EC2，例如 Lambda 函

自動擴展機制	在哪裡使用
	數或 Amazon Elastic Container Service（AmazonECS）服務。
Kubernetes Cluster Autoscaler	使用 在 上自動擴展 Kubernetes 叢集 AWS。

- 擴展通常與運算服務相關，例如 Amazon EC2執行個體或 AWS Lambda 函數。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。如果您正在部署影片轉碼應用程式，則預期 100% CPU使用率，且不應成為您的主要指標。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，請考慮下列 Amazon 指南EC2：
  - 指標應為有效的使用率指標，並說明執行個體的忙碌程度。
  - 指標值必須與 Auto Scaling 群組中的執行個體數成比例增加或減少。
- 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記錄來驗證 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。透過預測性擴展，可以消除過度佈建容量的需求。如需更多詳細資訊，請參閱使用 Amazon EC2 Auto Scaling 進行預測擴展。

資源

相關文件：

- Amazon EC2 Auto Scaling 入門
- 採用Machine LearningEC2技術的 Predictive Scaling
- 使用 Amazon OpenSearch Service、Amazon Data Firehose 和 Kibana 分析使用者行為
- 什麼是 Amazon CloudWatch？
- 使用 Amazon 上的 Performance Insights 監控資料庫負載 RDS
- 推出使用 Amazon EC2 Auto Scaling 進行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 Cluster Auto Scaling

相關影片：

- AWS re：Invent 2023 - AWS 為前 1，000 萬使用者擴展
- AWS re：Invent 2023 - 永續架構：過去、現在和未來
- AWS re：Invent 2022 - 建置經濟、高能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
- AWS re：Invent 2023 - 使用 Amazon 將 FM 推論擴展至數百個模型 SageMaker
- AWS re：Invent 2023 - 利用 Karpenter 的強大功能來擴展、最佳化和升級 Kubernetes

相關範例：

- 自動擴展

SUS02-BP02 SLAs與永續發展目標保持一致

根據您的永續性目標檢閱和最佳化工作負載服務層級協議 （SLA），以盡可能減少支援工作負載所需的資源，同時繼續滿足業務需求。

常見的反模式：

- 工作負載SLAs未知或不明確。
- 您可以為可用性和效能定義您的 SLA 。
- 您對所有工作負載使用相同的設計模式 (例如多可用區域架構)。

建立此最佳實務的好處：SLAs與永續發展目標保持一致，可在滿足業務需求的同時獲得最佳資源使用量。

未建立此最佳實務時的曝險等級：低

實作指引

SLAs 定義從雲端工作負載預期的服務層級，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。在定期檢視SLAs和進行權衡，以顯著降低資源使用量，以換取可接受的服務水準降低。

實作步驟

- 了解永續性目標：識別組織中的永續性目標，例如減碳或提高資源使用率。
- 檢閱 SLAs：評估您的 SLAs以評估它們是否支援您的業務需求。如果您超過 SLAs，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。通常情況下，優先考慮其中兩個因素會以犧牲第三個因素為代價。
- 調整 SLAs：SLAs透過進行權衡來大幅降低永續性影響，以換取可接受的服務水準降低來調整您的 。
  - 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。
- 如果可能SLAs，請定義永續性：包含工作負載SLAs的永續性。例如，將最低使用率層級定義為運算執行個體SLA的永續性。
- 使用有效的設計模式：在 上使用微服務等設計模式 AWS ，以排定業務關鍵函數的優先順序，並允許非關鍵函數較低的服務層級 （例如回應時間或復原時間目標）。
- 溝通和建立責任：SLAs與所有相關利益相關者共用 ，包括您的開發團隊和客戶。使用報告來追蹤和監控 SLAs。指派問責，以符合 的永續性目標SLAs。
- 使用激勵和獎勵：使用激勵和獎勵來實現或超越SLAs永續發展目標。
- 檢閱和迭代：定期檢閱和調整您的 SLAs ，以確保其符合不斷變化的永續性和效能目標。

資源

相關文件：

- 了解恢復模式和權衡取捨以便在雲端中高效進行架構
- 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、成本效益：挑選三個
- AWS re：Invent 2023 - 永續架構：過去、現在和未來
- AWS re：Invent 2023 - 鬆散耦合系統的進階整合模式和權衡
- AWS re：Invent 2022 - 提供永續、高效能的架構

- AWS re：Invent 2022 - 建置具成本、能源和資源效益的運算環境

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曝險等級：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、靜態影像和資產存取模式)，以識別冗餘、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，以識別工作負載內的所有資產。
- 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 自動化建立、保留和刪除 Amazon EBS快照和 Amazon EBS後端 AMIs。
- 審查和最佳化：定期審查工作負載，以識別並移除任何未使用的資產。

資源

相關文件：

- 最佳化 AWS 基礎設施以實現永續性，第 II 部分：儲存
- 如何終止不再需要的作用中資源 AWS 帳戶？

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 保留和最大化數位媒體資產的價值
- AWS re：Invent 2023 - 最佳化多帳戶環境中的成本

SUS02-BP04 根據其聯網需求最佳化工作負載的地理定位

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。
- 所有流量都流經現有資料中心。

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曝險等級：中

實作指引

AWS 雲端 基礎設施是以區域、可用區域、置放群組和邊緣位置等位置選項為基礎，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。
  - 資料所在位置：對於資料密集型應用程式 (例如大數據和機器學習)，應用程式碼執行時應盡可能接近資料。
  - 使用者所在位置：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：考慮諸如成本和合規性之類的限制，如為工作負載選取區域時應考慮的事項中所述。
- 針對常用資產，使用本機快取或 AWS 快取解決方案以提升效能、減少資料移動以及降低環境影響。

服務	使用情況
Amazon CloudFront	使用 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：

服務	使用情況
Lambda@Edge	用於在物件未經快取時起始的大量運算作業。
Amazon CloudFront Functions	用於可由短期函數啟動的簡單使用案例，例如 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。

- 使用連線集區來支援連線重複使用，減少所需資源。

- 使用不仰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：

- 最佳化 AWS 基礎設施以實現永續性，第 部分III：聯網
- Amazon ElastiCache 文件
- 什麼是 Amazon CloudFront？
- Amazon CloudFront 金鑰功能
- AWS 全球基礎設施
- AWS 本機區域 和 AWS Outposts，為您的邊緣工作負載選擇正確的技術
- 置放群組
- AWS 本機區域
- AWS Outposts

相關影片：

- 在 上解密資料傳輸 AWS
- 在新一代 Amazon EC2執行個體上擴展網路效能
- AWS Local Zones 解說器影片
- AWS Outposts：概觀和運作方式
- AWS re：Invent 2023 - 邊緣和內部部署工作負載的遷移策略
- AWS re：Invent 2021 - AWS Outposts：將 AWS 體驗帶到內部部署
- AWS re：Invent 2020 - AWS Wavelength：在 5G 邊緣以超低延遲執行應用程式
- AWS re：Invent 2022 - AWS Local Zones：為分散式邊緣建置應用程式
- AWS re：Invent 2021 - 使用 Amazon 建置低延遲網站 CloudFront
- AWS re：Invent 2022 - 使用 改善效能和可用性 AWS Global Accelerator
- AWS re：Invent 2022 - 使用 建置您的全球廣域網路 AWS
- AWS re：Invent 2020：使用 Amazon Route 53 進行全域流量管理

相關範例：

- AWS 網路研討會
- 永續性架構 - 盡可能減少跨網路的資料移動

SUS02-BP05 針對執行的活動最佳化團隊成員資源

最佳化提供給團隊成員的資源，以盡量減少對環境永續性的影響，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：最佳化團隊成員資源，可為啟用雲端的應用程式改善整體效率。

未建立此最佳實務時的曝險等級：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。實作將這些資源最佳化的策略。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制升級與裝置要求。
- 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，並選取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：為裝置實作遠端管理，以減少必要商務差旅時間。
  - AWS Systems Manager Fleet Manager 是一種統一使用者介面 （UI） 體驗，可協助您遠端管理在內部部署 AWS 或內部部署上執行的節點。

資源

相關文件：

- 什麼是 Amazon WorkSpaces？
- Amazon 的成本最佳化工具 WorkSpaces
- Amazon AppStream 2.0 文件
- NICE DCV

相關影片：

- 管理 Amazon WorkSpaces on 的成本 AWS

SUS02-BP06 實作緩衝或調節，以扁平需求曲線

緩衝和限流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：

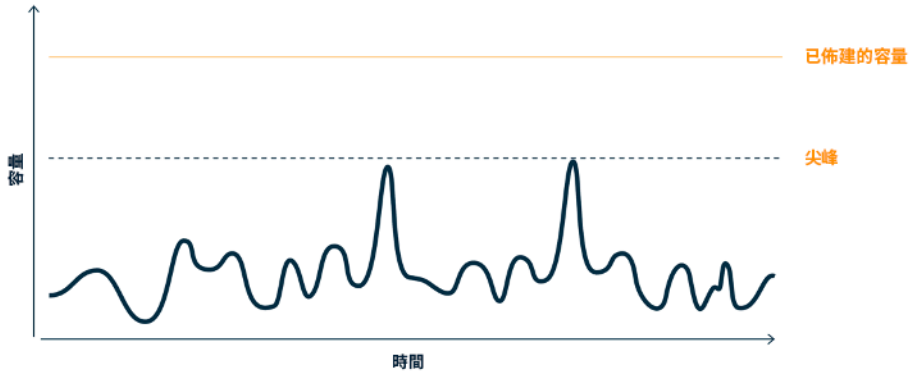
- 您非必要地立即處理用戶端請求。
- 您未分析用戶端要求的需求。

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曝險等級：低

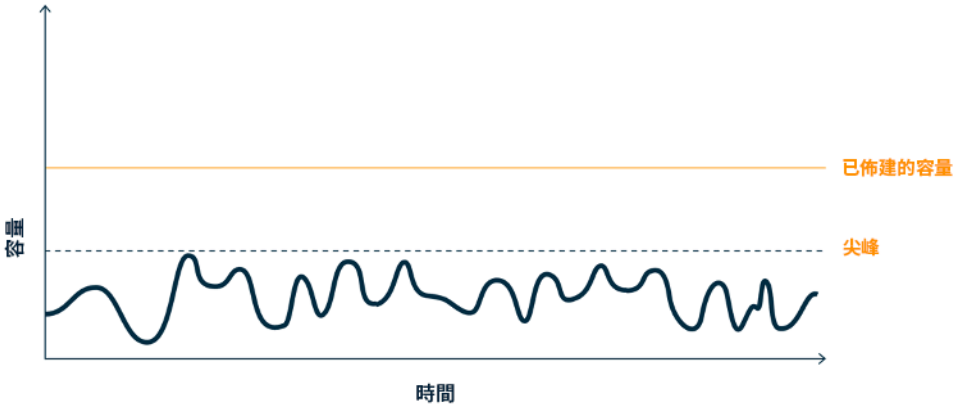
實作指引

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，內含兩個需要大量佈建容量的相異尖峰。

您可以使用緩衝或限流來修改需求曲線，並使尖峰趨緩，意即減少佈建容量和耗用的能源。在用戶端可以執行重試時實作限流。實作緩衝機制以儲存請求，並將處理的時間往後延遲。



限流對需求曲線和佈建容量的影響。

實作步驟

- 分析用戶端請求以確定如何予以回應。要考慮的問題包括：
  - 此請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，您可以實作限流，以告知來源若目前無法處理請求，則應稍後再試。
  - 您可以使用 Amazon API Gateway 實作限流。

- 針對無法執行重試的用戶端，需要實作緩衝區使需求曲線扁平化。緩衝會延遲請求處理，讓以不同速率執行的應用程式能夠有效地通訊。緩衝型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 ） 是一種受管服務，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 入門 SQS
- 使用佇列與訊息進行應用程式整合
- 管理和監控工作負載中的API限流
- 使用 API Gateway REST API 大規模限流分層多租戶
- 使用佇列與訊息進行應用程式整合

相關影片：

- AWS re：Invent 2022 - 微服務的應用程式整合模式
- AWS re：Invent 2023 - 智慧節省：Amazon EC2成本最佳化策略
- AWS re：Invent 2023 - 鬆散耦合系統的進階整合模式和權衡

軟體和架構

問題

- SUS 3 如何利用軟體和架構模式來支援永續性目標？

SUS 3 如何利用軟體和架構模式來支援永續性目標？

實施可執行負載順暢並保持已部署資源一致高使用率的模式，將資源消耗降至最低。由於使用者行為隨時間改變，元件可能會因缺乏使用而閒置。修改模式和架構來整合未充分利用的元件，提高整體使用率。淘汰不再需要的元件。了解工作負載元件的效能，並最佳化消耗最多資源的元件。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 針對非同步和排程任務最佳化軟體和架構
- SUS03-BP02 移除或重構低使用率或不使用的工作負載元件
- SUS03-BP03 最佳化消耗最多時間或資源的程式碼區域
- SUS03-BP04 最佳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 針對非同步和排程任務最佳化軟體和架構

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：

- 高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曝險等級：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。使用高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確定如何回應。
- 如果請求或任務不需要同步回應，請使用佇列驅動的架構和自動擴展工作節點，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：

佇列機制	描述
AWS Batch 任務佇列	AWS Batch 任務會提交至其所在的任務佇列，直到可以排程在運算環境中執行為止。
Amazon Simple Queue Service 和 Amazon EC2 Spot 執行個體	配對 Amazon SQS和 Spot 執行個體，以建置容錯且高效率的架構。

- 對於可以隨時處理的佇列或任務，請使用排程機制來批次處理任務，以提升效率。以下是在 上排程機制的一些範例 AWS：

排程機制	描述
Amazon EventBridge Scheduler	Amazon EventBridge 的功能，可讓您大規模建立、執行和管理排程任務。
AWS Glue 以時間為基礎的排程	在 中為您的爬蟲程式和任務定義以時間為基礎的排程 AWS Glue。
Amazon Elastic Container Service （Amazon ECS） 排程任務	Amazon ECS支援建立排程任務。排程的任務會使用 Amazon EventBridge 規則，在排程或 EventBridge 事件回應中執行任務。
Instance Scheduler	設定 Amazon 和 Amazon Relational Database Service 執行個體的啟動EC2和停止排程。

- 如果您的架構中使用輪詢和 Webhook 機制，請將其取代為事件。使用事件驅動的架構來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 識別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 擴展 SQS
- 什麼是 AWS Step Functions？
- 什麼是 AWS Lambda？
- AWS Lambda 搭配 Amazon 使用 SQS
- 什麼是 Amazon EventBridge？
- 使用 管理非同步工作流程 REST API

相關影片：

- AWS re：Invent 2023 - 導覽至無伺服器事件驅動架構的旅程
- AWS re：Invent 2023 - 將無伺服器用於事件驅動的架構和網域驅動的設計
- AWS re：Invent 2023 - 使用 Amazon 的進階事件驅動模式 EventBridge
- AWS re：Invent 2023 - 永續架構：過去、現在和未來
- 非同步訊息模式 | AWS 事件

相關範例：

- 使用 AWS Graviton 處理器和 Amazon EC2 Spot 執行個體的事件驅動架構

SUS03-BP02 移除或重構低使用率或不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。
- 您不會檢查和分析來自 AWS 授權工具的建議，例如 AWS Compute Optimizer。

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曝險等級：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，隨著新服務和功能 AWS 發行，工作負載的最佳服務和架構可能會變更。

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

- 清查您的 AWS 資源。在 中 AWS，您可以開啟 AWS 資源總管來探索和組織您的 AWS 資源。如需更多詳細資訊，請參閱 AWS re：Invent 2022 - 如何在 上大規模管理資源和應用程式 AWS。
- 監控和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 授權工具，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。
- 不再需要的淘汰元件和相關資產（例如 Amazon ECR映像）。
  - 自動清除 Amazon 中未使用的映像 ECR
  - 使用 AWS Config 和 刪除未使用的 Amazon Elastic Block Store（Amazon EBS）磁碟區 AWS Systems Manager
- 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，您可以在單一 Amazon RDS 資料庫執行個體上佈建多個小型資料庫，而不是在個別未充分利用的執行個體上執行資料庫。
- 了解您的工作負載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
- 使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、成本效益：選擇三

相關範例：

- 最佳化硬體模式並觀察永續性 KPIs

SUS03-BP03 最佳化消耗最多時間或資源的程式碼區域

最佳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最佳化。
- 您通常藉由增加資源來回應效能問題。
- 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曝險等級：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)，以最佳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

- 使用高效率的程式設計語言：使用適用於工作負載的高效率作業系統和程式設計語言。如需有關高能效程式設計語言 (包括 Rust) 的詳細資訊，請參閱 Rust 的永續性。
- 使用 AI 編碼搭配：考慮使用 AI 編碼搭配，例如 Amazon CodeWhisperer 來有效率地編寫程式碼。
- 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 偵測並行錯誤 CodeGuru

  - 使用 Amazon 提升 Python 應用程式的程式碼品質 CodeGuru
- 使用程式碼分析工具：使用程式碼分析工具來識別程式碼中使用最多時間或資源的區域，作為最佳化目標。
  - 使用 Amazon CodeGuru Profiler 減少組織的碳足跡
  - 使用 Amazon CodeGuru Profiler 了解 Java 應用程式中的記憶體用量
  - 使用 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 程式碼轉換升級語言版本
  - AWS re：Invent 2023 - 使用 Amazon Q Code Transformation 自動化應用程式升級和維護

資源

相關文件：

- 什麼是 Amazon CodeGuru Profiler？
- FPGA 執行個體
- 要在 上建置 AWS SDKs的工具 AWS

相關影片：

- 使用 Amazon CodeGuru Profiler 改善程式碼效率
- AWS re：Invent 2023 - Amazon 最佳實務 CodeWhisperer
- 使用 Amazon 自動化程式碼檢閱和應用程式效能建議 CodeGuru

相關範例：

- 使用 Amazon 最佳化程式碼 CodeGuru

SUS03-BP04 最佳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：實作為客戶裝置最佳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曝險等級：中

實作指引

實作為客戶裝置最佳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。
- 最佳化應用程式以在裝置上有效執行，有助於降低能源耗用量及延長電池使用壽命 (若是由電池供電)。
- 最佳化裝置的應用程式也可減少網路上的資料傳輸。

了解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：清查您的架構中使用的裝置。裝置可以是行動裝置、平板電腦、IOT裝置、智慧燈，甚至工廠中的智慧裝置。
- 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最佳化格式。例如，您可以使用 Amazon Elastic Transcoder 或 AWS Elemental MediaConvert 將大型高品質數位媒體檔案轉換為使用者可以在行動裝置、平板電腦、Web 瀏覽器和外接電視機上播放的格式。
    - 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：與使用永續性材料並提供供應鏈和環境認證透明度的裝置供應商合作。
- 使用 over-the-air （OTA） 更新：使用自動化 over-the-air （OTA） 機制將更新部署至一或多個裝置。
    - 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 使用受管裝置陣列進行測試。
- 繼續監控和改善：追蹤裝置的能源使用情況，以識別需要改善的區域。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
- OTA 在執行免費之裝置上更新韌體的教學課程RTOS
- 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 改善您的行動和 Web 應用程式品質 AWS Device Farm

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最佳化您的架構，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曝險等級：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最佳化的專用服務)。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
  - 上的資料庫 AWS：正確任務的正確工具
- 利用可原生處理壓縮資料的技術。
  - Athena 壓縮支援檔案格式
  - 中的ETL輸入和輸出格式選項 AWS Glue
  - 使用 Amazon Redshift 從 Amazon S3 載入壓縮資料檔案

- 使用專用分析服務進行架構中的資料處理。如需 AWS 專用分析服務的詳細資訊，請參閱 AWS re：Invent 2022 - 在 上建置現代資料架構 AWS。
- 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協定。

資源

相關文件：

- COPY Amazon Redshift 的資料欄格式
- 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
- 使用 Amazon 上的 Performance Insights 監控資料庫負載 RDS
- 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 上建置資料網格架構 AWS
- AWS re：Invent 2023 - 深入探索 Amazon Aurora 及其創新
- AWS re：Invent 2023 - 改善 Amazon EBS效率並提高成本效益
- AWS re：Invent 2023 - 使用 Amazon S3 最佳化儲存價格和效能
- AWS re：Invent 2023 - 在 Amazon S3 上建置和最佳化資料湖
- AWS re：Invent 2023 - 使用 Amazon 的進階事件驅動模式 EventBridge

相關範例：

- AWS 目的建構資料庫研討會
- AWS 現代資料架構沉浸式日
- 在 上建置資料網格 AWS

資料

問題

- SUS 4 如何利用資料管理政策和模式來支援永續性目標？

SUS 4 如何利用資料管理政策和模式來支援永續性目標？

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 使用彈性和自動化來擴展區塊儲存或檔案系統
- SUS04-BP05 移除不需要或備援的資料
- SUS04-BP06 使用共用檔案系統或儲存體來存取常用資料
- SUS04-BP07 將跨網路的資料移動降至最低
- SUS04-BP08 僅在難以重新建立時備份資料

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曝險等級：中

實作指引

資料分類涉及識別組織擁有或營運的資訊系統中正在處理和儲存的資料類型。此外涉及確認資料的關鍵性，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，確定資料的關鍵性、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。相應地標記資料。如需有關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，並適當地分類和標記資料。範例請見 AWS Glue中的資料型錄和編目程式。
- 建立資料型錄：建立提供稽核及管控能力的資料型錄。
- 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：

- 利用 AWS 雲端 以支援資料分類
- 來自 的標籤政策 AWS Organizations

相關影片：

- AWS re：Invent 2022 - 在 上啟用資料治理的靈活性 AWS
- AWS re：Invent 2023 - 資料保護和 AWS 儲存的彈性

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。

建立此最佳實務的優勢：根據資料存取和儲存模式來選取及最佳化您的儲存技術，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曝險等級：低

實作指引

選取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。以下是一些 AWS 儲存技術及其關鍵特性的範例：

Type	技術	重要特性
物件儲存	Amazon Simple Storage Service (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。

Type	技術	重要特性
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會自動擴充和縮減儲存體，並針對效能進行最佳化，以提供一致的低延遲。
共用檔案系統	Amazon FSx	以最新的 AWS 運算解決方案為基礎，支援四個常用的檔案系統： NetApp ONTAP、Open ZFS、Windows File Server 和 Lustre。Amazon FSx延遲、輸送量和 會因檔案系統 IOPS而異，因此在選擇適合您工作負載需求的檔案系統時，應該考慮這一點。
區塊儲存	Amazon Elastic Block Store （Amazon EBS）	專為 Amazon Elastic Compute Cloud （Amazon ） 設計的可擴展、高效能區塊儲存服務EC2。Amazon EBS包含交易IOPS型 密集型工作負載的 SSD後端儲存體，以及輸送量密集型工作負載的 HDD後端儲存體。

Type	技術	重要特性
關聯式資料庫	Amazon Aurora、Amazon RDS、Amazon Redshift	旨在支援 ACID（原子性、一致性、隔離、耐久性）交易，並維持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其資料。
鍵值資料庫	Amazon DynamoDB	已針對常見的存取模式進行最佳化，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。

- 自動化儲存分配：對於 Amazon EBS或 Amazon 等固定大小的儲存系統FSx，監控可用的儲存空間，並在達到閾值時自動化儲存分配。您可以利用 Amazon CloudWatch 來收集和分析 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儲存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，而無需進行任何應用程式變更。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 - 改善 Amazon EBS效率並提高成本效益
- AWS re：Invent 2023 - 使用 Amazon S3 最佳化儲存價格和效能
- AWS re：Invent 2023 - 在 Amazon S3 上建置和最佳化資料湖
- AWS re：Invent 2022 - 在 上建置現代資料架構 AWS
- AWS re：Invent 2022 - 使用專用資料庫現代化應用程式
- AWS re：Invent 2022 - 在 上建置資料網格架構 AWS
- AWS re：Invent 2023 - 深入探索 Amazon Aurora 及其創新
- AWS re：Invent 2023 - 使用 Amazon DynamoDB 進行進階資料建模

相關範例：

- Amazon S3 範例
- AWS 目的建構資料庫研討會
- 專為開發人員打造的資料庫
- AWS 現代資料架構沉浸式日
- 在 上建置資料網格 AWS

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曝險等級：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。
- 設定自動生命週期政策以強制執行生命週期規則。以下是如何為不同 AWS 儲存服務設定自動生命週期政策的一些範例：

儲存服務	如何設定自動化生命週期原則
Amazon Simple Storage Service (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，讓其監控存取模式，並自動將未存取的物件移至成本較低的存取層。可以利用 Amazon S3 Storage Lens 指標，找出生命週期管理中的最佳化機會和差距。
Amazon Elastic Block Store	您可以使用 Amazon Data Lifecycle Manager 自動化建立、保留和刪除 Amazon EBS快照和 Amazon EBS後端 AMIs。
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年齡或計數過期的影像，自動清除容器映像。

儲存服務	如何設定自動化生命週期原則
AWS Elemental MediaStore	您可以使用物件生命週期政策來管理物件應該存放在 MediaStore 容器中的時間長度。

- 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 存留時間或 Amazon CloudWatch 日誌保留以進行刪除。
- 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 儲存類別分析最佳化 Amazon S3 生命週期規則
- 使用 評估資源 AWS Config 規則

相關影片：

- AWS re：Invent 2021 - Amazon S3 生命週期最佳實務，以最佳化您的儲存支出
- AWS re：Invent 2023 - 使用 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最佳化儲存成本
- 使用 Amazon S3 Storage Lens 降低儲存成本

SUS04-BP04 使用彈性和自動化來擴展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。
- 您過度佈建檔案系統的每秒輸入和輸出操作（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曝險等級：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，而無需過度佈建這些儲存服務。

實作步驟

- 對於 Amazon EBS等固定大小的儲存體，請確認您正在監控使用的儲存體數量與整體儲存體大小的比較，並在可能的情況下建立自動化，以在達到閾值時增加儲存體大小。
- 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 Elastic Volumes 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資料大小。

資源

相關文件：

- 在調整EBS磁碟區大小後擴展檔案系統
- 使用 Amazon EBS Elastic Volumes 修改磁碟區
- Amazon FSx 文件
- 什麼是 Amazon Elastic File System？

相關影片：

- 深入探索 Amazon EBS Elastic Volumes
- Amazon EBS和 Snapshot 最佳化策略可提供更好的效能和節省成本
- 使用最佳實務來最佳化 Amazon EFS 的成本和效能

SUS04-BP05 移除不需要或備援的資料

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重新建立的資料。
- 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開啟 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曝險等級：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上的開放資料中現有的公開提供的資料集，來避免儲存資料。
- 使用可在區塊和物件層級刪除重複資料的機制。以下是如何在 上刪除重複資料的一些範例 AWS：

儲存服務	重複資料刪除機制
Amazon Simple Storage Service (Amazon S3)	使用 AWS Lake Formation FindMatches 尋找跨資料集（包括沒有識別符的記錄）的相符記錄，方法是使用新的 FindMatches ML 轉換。
Amazon FSx	在 Amazon FSx for Windows 上使用重複資料刪除。
Amazon Elastic Block Store 快照	快照為遞增備份，這表示只會儲存您上次執行裝置快照後發生變更的區塊。

- 分析資料存取以識別不需要的資料。自動化生命週期政策。利用原生服務功能，例如 Amazon DynamoDB 存留時間 、Amazon S3 生命週期 或 Amazon CloudWatch 日誌保留以進行刪除。
- 在 上使用資料虛擬化功能 AWS ，以維護來源的資料，並避免資料重複。
  - 上的雲端原生資料虛擬化 AWS
  - 使用 Amazon Redshift 資料共用來最佳化資料模式
- 使用可以進行增量備份的備份技術。
- 利用 Amazon S3 的耐用性和 Amazon 複寫EBS來實現您的耐用性目標，而不是自我管理的技術（例如冗餘的獨立磁碟陣列（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。
- 在推送工作負載的新版本時，從物件存放區和邊緣快取中移除 out-of-date部署和資產。

資源

相關文件：

- 變更日誌中的 CloudWatch 日誌資料保留
- Amazon FSx for Windows File Server 上的重複資料刪除
- Amazon 的功能FSx，用於ONTAP包含重複資料刪除
- 對 Amazon 上的檔案進行驗證 CloudFront
- 使用 來 AWS Backup 備份和還原 Amazon EFS 檔案系統
- 什麼是 Amazon CloudWatch Logs？
- 在 Amazon 上使用備份 RDS
- 使用 整合和刪除重複資料集 AWS Lake Formation

相關影片：

- Amazon Redshift 資料共用使用案例

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來存取常用資料

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曝險等級：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，讓您的工作負載使用高效的基礎設施。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 上共用儲存技術的一些範例 AWS：

儲存選項	使用情況
Amazon EBS Multi-Attach	Amazon EBS Multi-Attach 可讓您將單一佈建 IOPSSSD（io1 或 io2） 磁碟區連接至位於相同可用區域的多個執行個體。
Amazon EFS	請參閱選擇 Amazon 的時機EFS。
Amazon FSx	請參閱選擇 Amazon FSx File System 。

儲存選項	使用情況
Amazon Simple Storage Service (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。

- 僅在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 S3 支援的 Amazon FSx for Lustre 檔案系統，並且只將處理任務所需的資料子集載入 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：

- 將您的檔案系統連結到 Amazon S3 儲存貯體
- 在無伺服器應用程式中 AWS Lambda 使用 Amazon EFS for
- Amazon EFS Intelligent-Tiering 透過變更存取模式來最佳化工作負載的成本
- 將 Amazon FSx與內部部署資料儲存庫搭配使用

相關影片：

- Amazon 的儲存成本最佳化 EFS
- AWS re：Invent 2023 - AWS 檔案儲存的新功能
- AWS re：Invent 2023 - Amazon Elastic File System 上建置者和資料科學家的檔案儲存

SUS04-BP07 將跨網路的資料移動降至最低

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聯網資源降至最低。

常見的反模式：

- 您可以將所有資料存放在與資料使用者所在位置 AWS 區域 相同的 中。
- 您未最佳化資料大小和格式，便將其移至網路。

建立此最佳實務的優勢：最佳化整個網路間的資料移動，可減少工作負載所需的整體聯網資源，並降低其環境影響。

未建立此最佳實務時的曝險等級：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。
- 使用有效的檔案格式（例如 Parquet 或 ORC）並壓縮資料，然後再透過網路移動。
- 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：
  - 僅減少對相關資料的API回應。
  - 彙總詳細的資料 (不需要記錄層級資訊)。
  - 參閱 Well-Architected 實驗室 - 使用 Amazon Redshift 資料共用來最佳化資料模式。
  - 考慮 中的跨帳戶資料共用 AWS Lake Formation。
- 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。

服務	使用情況
Lambda@Edge	用於在物件未經快取時執行的大量運算作業。
CloudFront 函數	用於可由短期函數啟動的簡單使用案例，例如 HTTP（s）請求/回應操作。
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：

- 最佳化 AWS 基礎設施以實現永續性，第 部分III：聯網

- AWS 全球基礎設施
- Amazon CloudFront 主要功能，包括 CloudFront Global Edge Network
- 在 Amazon OpenSearch Service 中壓縮HTTP請求
- 使用 Amazon 進行中繼資料壓縮 EMR
- 從 Amazon S3 載入壓縮資料檔案至 Amazon Redshift
- 使用 Amazon 提供壓縮檔案 CloudFront

相關影片：

- 在 上解密資料傳輸 AWS

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曝險等級：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 所述的資料分類政策。
- 根據您的復原時間目標 （RTO） 和復原點目標 （ ），使用資料分類和RPO設計備份策略的重要性。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。
    - 排除資料的本機複本，除非從常見位置還原資料所需的時間超過您的服務層級協議（SLAs）。
- 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 是一項完全受管的服務，可讓您輕鬆集中和自動化跨 AWS 服務、雲端和內部部署的資料保護。如需如何使用 建立自動備份的實作指南 AWS Backup，請參閱 Well-Architected Labs – 測試資料備份和還原。
    - EFS 使用 自動化備份並最佳化 Amazon 的備份成本 AWS Backup。

資源

相關的最佳實務：

- REL09-BP01 識別和備份需要備份的所有資料，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來滿足復原目標

相關文件：

- 使用 來 AWS Backup 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：可協助備份的合作夥伴
- AWS Marketplace：可用於備份的產品
- 備份 Amazon EFS
- 備份 Amazon FSx for Windows File Server
- Amazon 的備份和還原 ElastiCache（RedisOSS）

相關影片：

- AWS re：Invent 2023 - 備份和災難復原策略，以提高復原能力
- AWS re：Invent 2023 - 新功能 AWS Backup
- AWS re：Invent 2021 - 備份、災難復原和勒索軟體保護搭配 AWS

相關範例：

- Well-Architected 實驗室 - 備份資料

硬體和服務

問題

- SUS 5 如何在架構中選擇並使用雲端硬體和服務來支援永續性目標？

SUS 5 如何在架構中選擇並使用雲端硬體和服務來支援永續性目標？

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 使用最少的硬體量來滿足您的需求
- 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 最佳化硬體型運算加速器的使用

SUS05-BP01 使用最少的硬體量來滿足您的需求

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確定是否應調整其大小。
- 您不會根據業務 為運算基礎設施設定硬體使用率目標KPIs。

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曝險等級：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 提供了透過各種機制動態擴展或減少資源數量的靈活性，例如 AWS Auto Scaling，並滿足需求的變化。它也提供 APIs和 SDKs ，允許以最少的努力修改資源。使用這些功能可以頻繁變更工作負載實作。此外，請使用 AWS 工具的許可化指南，有效率地操作您的雲端資源並滿足您的業務需求。

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 Fleet 的屬性型執行個體類型選擇。
  - 使用屬性型執行個體類型選取來建立 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：使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 On-Demand Instances 最適合新的、具狀態和繁重的工作負載，這些工作負載不能是執行個體類型、位置或時間彈性。
  - Amazon EC2 Spot 執行個體是補充容錯且彈性應用程式其他選項的絕佳方式。
  - 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。
- 授權執行個體：使用來自 AWS 工具的 授權建議，對工作負載進行調整。如需詳細資訊，請參閱利用精簡化建議最佳化您的成本和適當調整大小：佈建執行個體以符合工作負載
  - 在 AWS Cost Explorer 或 中使用許可化建議AWS Compute Optimizer來識別許可化機會。
- 協商服務層級協議 （ SLAs ）：協商SLAs，允許在自動化部署替代資源時暫時減少容量。

資源

相關文件：

- 最佳化 AWS 基礎設施的永續性，第 I 部分：運算
- Amazon EC2 Fleet Auto Scaling 的屬性型執行個體類型選擇
- AWS Compute Optimizer 文件

- 操作 Lambda：效能最佳化
- Auto Scaling 文件

相關影片：

- AWS re：Invent 2023 - Amazon 的新功能 EC2
- AWS re：Invent 2023 - 智慧節省：Amazon Elastic Compute Cloud 成本最佳化策略
- AWS re：Invent 2022 - 最佳化 Amazon Elastic Kubernetes Service 的效能和成本 AWS
- AWS re：Invent 2023 - 永續運算：使用 降低成本和碳排放 AWS

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。
- 您可以在 Amazon EC2 Auto Scaling 組態中指定一個執行個體類型。
- 您使用 AWS 執行個體的方式並非為其設計 （例如，您將運算最佳化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。
- 您不會檢查對 等工具 AWS 進行許可化的建議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曝險等級：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 What's New with AWS，以掌握 up-to-date最新的 AWS 技術和執行個體。
  - 了解不同的 AWS 執行個體類型。
  - EC2 觀看 re：Invent 2020 - Deep dive on AWS Graviton2 處理器驅動的 Amazon 執行個體，以及 Deep dive in AWS Graviton3 和 Amazon C7g 執行個體，了解 Graviton EC2 C7g 型執行個體在 Amazon 中提供每瓦能源使用的最佳效能。 AWS Graviton2 EC2
- 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。
  - 如果可能，請修改工作負載以使用不同數量 vCPUs 和不同數量的記憶體，以最大化您選擇的執行個體類型。
  - 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。如需將工作負載移至 AWS Graviton 的詳細資訊，請參閱 AWS Graviton 快速入門和將工作負載轉換為 AWS Graviton 型 Amazon Elastic Compute Cloud 執行個體時的考量事項。
  - 考慮在使用 受管服務時選取 AWS Graviton 選項。 AWS
  - 將工作負載遷移至有執行個體對永續性影響最小，且仍符合業務要求的區域。
  - 對於機器學習工作負載，請善用專為工作負載量身打造的專用硬體，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。 AWS 相較於類似的 Amazon 執行個體，Inf2 執行個體等 Inferentia EC2執行個體可提供高達每瓦 50% 的效能。
  - 使用 Amazon SageMaker Inference Recommender 來正確調整 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 執行個體來增加雲端的整體使用率，並減少未使用資源的永續性影響。
- 操作和最佳化：操作並最佳化您的工作負載執行個體。
  - 對於暫時性工作負載，請評估執行個體 Amazon CloudWatch 指標，例如 `CPUUtilization`，以識別執行個體是閒置還是未充分利用。
  - 對於穩定的工作負載，請定期檢查 AWS 授權工具，例如 AWS Compute Optimizer ，以識別最佳化和調整執行個體大小的機會。如需更多範例和建議，請參閱下列實驗室：
    - 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
    - 建構良好的實驗室 - 最佳化硬體模式和 Observice 永續性 KPIs

資源

相關文件：

- 最佳化 AWS 基礎設施以實現永續性，第 1 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 Fleet 的屬性型執行個體類型選擇
- 在 AWS上建置永續性、高效且成本最佳化的應用程式
- Contino 永續性儀表板如何協助客戶最佳化碳足跡

相關影片：

- AWS re：Invent 2023 - AWS Graviton： AWS 工作負載的最佳價格效能
- AWS re：Invent 2023 - 中的新 Amazon Elastic Compute Cloud 生成 AI 功能 AWS Management Console
- AWS re：Invent 2023 = Amazon Elastic Compute Cloud 的新功能
- AWS re：Invent 2023 - 智慧節省：Amazon Elastic Compute Cloud 成本最佳化策略
- AWS re：Invent 2021 - 深入探索 AWS Graviton3 和 Amazon EC2 C7g 執行個體
- AWS re：Invent 2022 - 建置具成本、能源和資源效益的運算環境

相關範例：

- 解決方案： 上永續發展的深度學習工作負載最佳化指南 AWS
- 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：

- 您可以使用低使用率的 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 受管服務將責任轉移到 AWS，這具有數百萬個客戶之間的洞察，有助於推動新的創新和效率。
- 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曝險等級：中

實作指引

受管服務將責任轉移到 AWS ，以維持已部署硬體的高使用率和永續性最佳化。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。

檢閱您的工作負載，以識別可以由 AWS 受管服務取代的元件。例如，Amazon RDS、Amazon Redshift 和 Amazon ElastiCache 提供受管資料庫服務。Amazon AthenaAmazon 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. 識別候選項：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：

任務	要在 上使用什麼 AWS
託管資料庫	使用受管 Amazon Relational Database Service （ Amazon RDS ） 執行個體，而不是在 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。

任務	要在 上使用什麼 AWS
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最佳化服務。

資源

相關文件：

- AWS 雲端 產品
- AWS 總擁有成本 （ 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 - 大規模雲端操作 AWS Managed Services
- AWS re：Invent 2023 - 在 上操作的最佳實務 AWS

SUS05-BP04 最佳化硬體型運算加速器的使用

將加速運算執行個體的使用方式最佳化，以降低工作負載的實體基礎設施需求。

常見的反模式：

- 您未監控GPU用量。

- 針對工作負載使用一般用途執行個體，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。
- 您正在將硬體型運算加速器用於使用 CPU型替代方案更有效率的任務。

建立此最佳實務的優勢：藉由將硬體型加速器的使用方式最佳化，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曝險等級：中

實作指引

如果您需要高處理能力，則可以受益於使用加速運算執行個體，這可讓您存取硬體型運算加速器，例如圖形處理單元（GPUs）和現場可程式設計閘道陣列（）FPGAs。這些硬體加速器會比 CPU型替代方案更有效率地執行圖形處理或資料模式比對等特定功能。許多加速的工作負載 (例如轉譯、轉碼和機器學習) 在資源使用方面變化很大。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

- 確定哪些加速運算執行個體可以滿足您的需求。
- 對於機器學習工作負載，請善用工作負載特有的專用硬體，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。相較於類似的 Amazon 執行個體，Inf2 執行個體等 AWS Inferentia 執行個體可提供高達每瓦 50% 的效能。 EC2
- 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 的指標，例如 `utilization_memory` `utilization_gpu`和 GPUs，如使用 Amazon 收集NVIDIAGPU指標 CloudWatch中所示。
- 優化硬體加速器的程式碼、網路運作和設定，以確保系統會充分利用基礎硬體。
  - 最佳化GPU設定
  - GPU 深度學習中的監控和最佳化 AMI
  - 最佳化 I/O 以在 Amazon 中進行深度學習訓練GPU的效能調校 SageMaker
- 使用最新的高效能程式庫和GPU驅動程式。
- 不使用時，請使用自動化來釋出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 EC2VT1執行個體
- 選擇使用 Amazon 進行電腦視覺推論的最佳 AI 加速器和模型編譯 SageMaker

相關影片：

- AWS re：Invent 2021 - 如何選取 Amazon EC2GPU執行個體進行深度學習
- 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 - 使用 AWS 和 的尖端 AI NVIDIA
- AWS re：Invent 2022 - 【NEW LAUNCH！】 介紹 AWS Inferentia2-based Amazon EC2 Inf2 執行個體
- AWS re：Invent 2022 - 使用 加速深度學習並更快速地創新 AWS Trainium
- AWS re：Invent 2022 - AWS 使用 進行深度學習NVIDIA：從訓練到部署

程序和文化

問題

- SUS 6 您的組織程序如何支援您的永續性目標？

SUS 6 您的組織程序如何支援您的永續性目標？

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 採用可快速引入永續發展改進的方法
- SUS06-BP02 保留工作負載 up-to-date
- SUS06-BP03 增加建置環境的使用率
- SUS06-BP04 使用受管裝置陣列進行測試

SUS06-BP01 採用可快速引入永續發展改進的方法

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曝險等級：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。
- 使用最低可行產品進行測試（MVP）：使用最低可行的代表性元件開發和測試潛在的改進，以減少測試的成本和環境影響。
- 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 啟用永續性解決方案
- 可擴展的靈活開發實務，以 為基礎 AWS CodeCommit

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、高效能的架構
- AWS re：Invent 2022 - 永續架構並減少 AWS 碳足跡
- AWS re：Invent 2022 - AWS 全球基礎設施的永續性
- AWS re：Invent 2023 - AWS 可觀測性和操作的新功能

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 保留工作負載 up-to-date

保留工作負載 up-to-date以採用有效率的功能、移除問題，並改善工作負載的整體效率。

常見的反模式：

- 假設您目前的架構為靜態，且不會隨著時間的推移而更新。
- 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曝險等級：低

實作指引

最新的作業系統、執行時期、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軟體和軟體政策所需的組態，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。

工作負載元件	如何更新
機器映像	使用 EC2 Image Builder 管理 Linux 或 Windows 伺服器映像的 Amazon Machine Images（AMIs）更新。
容器映像	將 Amazon Elastic Container Registry（Amazon ECR）與現有的管道搭配使用，以管理 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：自動化更新，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 AMIs、容器映像，以及與雲端應用程式相關的其他成品。
  - 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
- 新功能 AWS
- AWS 開發人員工具

相關影片：

- AWS re：Invent 2022 - 透過最佳實務指引最佳化 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 - 庫存和修補程式管理
- 實驗室： AWS Systems Manager

SUS06-BP03 增加建置環境的使用率

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曝險等級：低

實作指引

使用 自動化 和 infrastructure-as-code 在需要時建立環境，並在不使用時將其移除。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，請尋找機會使用具有爆量容量的執行個體類型、Amazon EC2 Spot 執行個體、自動擴展資料庫服務、容器和無伺服器技術，以將開發和測試容量與使用量保持一致。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

- 使用基礎設施即程式碼：使用基礎設施即程式碼來佈建您的建置環境。
- 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：使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 Burstable 效能執行個體
- 什麼是 AWS CloudFormation？
- 什麼是 AWS CodeBuild？
- 上的執行個體排程器 AWS

相關影片：

- AWS re：Invent 2023 - 的持續整合和交付 AWS

SUS06-BP04 使用受管裝置陣列進行測試

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 來測試啟用雲端功能的應用程式有許多優勢：

- 將有更多高效率功能可用來測試各種裝置上的應用程式。
- 無需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)，因而無需再進行不必要的裝置升級。

未建立此最佳實務時的曝險等級：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，並制定測試計劃。
- 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。
  - 將 AWS Device Farm 與您的 CI/CD 管道整合，以執行跨瀏覽器硒測試
  - 使用 和 行動服務建置和測試 iOS AWS DevOps 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 CloudWatchRUM儀表板

相關影片：

- AWS re：Invent 2023 - 使用 AWS Device Farm 改善您的行動和 Web 應用程式品質
- AWS re：Invent 2021 - 透過使用 Amazon 的終端使用者洞察最佳化應用程式 CloudWatch RUM

相關範例：

- AWS Device Farm 適用於 Android 的 App 範例
- AWS Device Farm iOS 版範例應用程式
- 的 Appium Web 測試 AWS Device Farm

注意

客戶有責任對本文件中的資訊進行自己的獨立評定。本文件：（a）僅供參考，（b）代表目前的 AWS 產品產品和實務，這些產品和實務可能隨時變更，恕不另行通知，以及（c）不會從 AWS 及其關聯機構、供應商或授權方建立任何承諾或保證。 AWS 產品或服務「原樣」提供，無論明示還是暗示，均無任何保證、陳述或條件。 AWS 對其客戶的責任和責任受 AWS 協議控制，本文件不屬於 與其 AWS 客戶之間的任何協議，也未對其進行修改。

AWS 詞彙表

如需最新的 AWS 術語，請參閱AWS 詞彙表 參考 中的AWS 詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。