AWS Well-Architected 架構

2020 年 7 月

This paper has been archived.

The latest version is now available at:

https://docs.aws.amazon.com/zh_tw/wellarchitected/latest/framework/welcome.html

本文件介紹 AWS Well-Architected 架構；該架構讓您可以審查和改進雲端架構，更了解您的設計決策對業務的影響。我們依照定義成為 Well-Architected 架構支柱的五大概念性領域，提出一般設計原則以及特定的最佳實務和指導。

聲明

客戶應負責對本文件中的資訊自行進行獨立評估。本文件：(a) 僅供參考之用，(b) 代表目前的 AWS 產品供應與實務，如有變更恕不另行通知，以及 (c) 不構成 AWS 及其附屬公司、供應商或授權人的任何承諾或保證。AWS 產品或服務以「現況」提供，不提供任何明示或暗示的擔保、主張或條件。AWS 對其客戶之責任與義務，應受 AWS 協議之約束，且本文件並不屬於 AWS 與其客戶間之任何協議的一部分，亦非上述協議之修改。

引言

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

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

「AWS Well-Architected 架構」記錄一份基本問題，能讓您了解特定架構是否妥善符合雲端最佳實務的條件。該架構提供一致的方針，可依照您預計自現代雲端系統可獲得的品質來評估系統，並能得知欲達到此等品質會需要的修補措施。AWS 持續在演進當中，我們也不斷地從與客戶一同工作之中學到更多，因此架構完善的定義會始終精進下去。

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

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

為協助您應用最佳實務，我們特別成立 AWS Well-Architected 實驗室，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。我們也與身為 AWS Well-Architected 合作夥伴計劃成員，也是精選 AWS 合作夥伴網路 (APN) 中的合作夥伴組成團隊並肩合作。這些 APN 合作夥伴對 AWS 擁有深入的知識，能協助您審查和改進工作負載。

定義

AWS 的專家每一天都在輔助客戶進行系統架構，善用雲端的最佳實務。隨著您的設計演進，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有多好，以及權衡形成的後果。

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

AWS Well-Architected 架構以五個支柱為基礎：卓越營運、安全性、可靠性、效能達成效率和成本優化。

Table 1. AWS Well-Architected 架構的支柱

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	安全性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。
可靠性	工作負載如預期般正確、一致地執行其預期功能的能力，這包括在其整個生命週期中操作和測試工作負載的能力。
效能達成效率	有效率地使用運算資源以滿足系統需求，並隨著需求變更與技術發展來保持該效率需求的能力。
成本優化	在最低價格之下執行系統以產生商業價值的能力

在 AWS Well-Architected 架構中我們所稱

- 元件代表應一項要求所一同遞送的程式碼、設定和 AWS 資源。一個元件往往是技術擁有的單元，並自其他元件所解偶。
- 我們所稱的工作負載是用以識別一同帶來商業價值的一組元件。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。
- 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、測試、上線，投入生產)。
- 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。
- 在組織內，技術組合是業務運作所需的工作負載的集合。

建立工作負載的架構時，您可依照業務環境，在各支柱之間作出權衡。這些業務決定可主導您工程設計的優先順序。您可以選擇在開發環境中以可靠性作為代價最佳化成本，或者針對關鍵任務解決方案，以較高成本達到可靠性的最佳化。在電子商務解決方案中，效能能影響營收和客戶購買的傾向。安全和卓越營運一般不會為了其他支柱而權衡妥協。

論架構

在內部部署環境中，客戶經常設置集中團隊負責技術架構，疊覆在其他產品或功能團隊上，以確保其遵照最佳實務。技術架構團隊經常以一組角色所組成，例如技術架構師 (基礎設施)、解決方案架構師 (軟體)、資料架構師、網聯架構師，和安全架構師。這類團隊經常採取 TOGAF 或 Zachman 框架作為企業架構能力的部分。

在 AWS，我們偏好將能力分散至團隊中，不以集中團隊具備該項能力。選擇將決策權分散有其風險存在，確保團隊符合內部標準即為一例。我們以兩種方式降低這類風險。首先，我們演練 專注在使得各個團隊具備該項能力，並且請到專家，確保該團隊提高所需[1]

[1] 做事方式、程序、標準，及可接受的規範。

符合標準的標竿。第二，我們實作機制[3]實施自動化檢查，以確保符合標準。這種分散式的作法受到 Amazon 領導方針的支持，遍及所有角色培養一種文化能起反向作用[2]出自客戶。以客戶為尊的團隊會因應客戶的需要建置產品。

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

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

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

[2] 反向作用是我們創新程序的基礎部分。我們從客戶及其期望著手，根據之定義並主導我們的工作方向。

[3] “立意良好是不夠的，需要以良好的機制才能有所實現” Jeff Bezos 言。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)。

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: 消除猜想基礎設施容量的需要。當您在部署系統之前做出容量的決定時，可能最後變成坐擁昂貴的閒置資源，或處理容量有限的效能影響。而利用雲端運算，這些問題都會消失。您可依照需要使用大小不拘的容量，自動上下調整。
- 生產規模測試系統: 在雲端，您可隨需建立生產規模的測試環境、完成測試，再將資源除役。因為您只為執行中的測試環境付費，所以能以與內部部署測試相較之下相當微小比例的成本來模擬即時環境。
- 自動化讓架構試驗更容易: 自動化可讓您用低成本建立並複製系統，避免產生人工開支。您可追蹤自動化的變更，稽核其影響，並可視需要還原為先前參數。
- 允許演進的架構: 允許演進的架構。在傳統環境中，架構上的決策往往實作成為靜態的一次性活動，其生命週期當中只有系統的少數主要版本。隨著業務及其環境持續改變，這些初始決定可能妨礙系統，無法符合不斷改變的業務要求。在雲端，按需自動化與測試的能力，可降低因設計變更而形成衝擊的風險。如此可允許系統隨時間演進，因此企業能以標準實務的形式享有創新的優勢。
- 使用資料來驅動架構: 在雲端，您可收集架構上的選擇對於工作負載的行為有何影響的資料。如此可讓您為如何提升工作負載，做出以事實為根據的決策。您的雲端基礎設施為程式碼，因此可隨時間利用該資料得知基礎設施的適當選擇及提升。
- 透過演練日進行改進: 為了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。

架構的五大支柱

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、效能效率和成本優化這五大支柱，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。

卓越營運

卓越營運 支柱包含 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。

卓越營運支柱概述了設計原則、最佳實務和相關問題。您可以在卓越營運支柱白皮書中找到實作的指引。

設計原則

有 five 項雲端 卓越營運 設計原則：

- 以程式碼執行營運: 在雲端，您可以在整個環境中套用與您應用程式程式碼所用相同的工程原則。您可將整個工作負載 (應用程式、基礎架構) 定義為程式碼，並以程式碼加以更新。您可以程式碼實作營運程序，並透過觸發這些程式碼來自動化執行，進而回應事件。透過以程式碼執行營運，您可限制人為錯誤並實現對事件的一致回應。
- 進行頻繁、細微和可逆的變更: 設計工作負載以允許定期更新元件。進行小增量變更，以便在變更失敗時能撤回變更 (盡可能不影響客戶)。
- 經常完善營運程序: 在使用營運程序時，尋找機會予以改善。發展工作負載，同時適當發展程序。設定定期演練日，以審查並驗證所有程序是否有效以及團隊是否熟悉這些程序。
- 預期失敗: 執行「事前剖析」演練，以識別潛在的失敗來源，進而排除或減少這些來源。測試您的失敗情境並驗證您對它們的影響的理解。測試您的回應程序，以確保它們確實有效且團隊熟悉程序的執行。設定定期演練日，以測試工作負載和團隊對模擬事件的回應。
- 從所有營運失敗中學習經驗: 從所有營運事件和失敗中學習經驗，進而不斷推動改善。跨團隊及在整個組織中分享獲得的經驗。

定義

有 four 個雲端 卓越營運 最佳實務方面：

- 組織

- 準備
- 操作
- 演進

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

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

最佳實務

組織

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

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

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

確保每個應用程式、工作負載、平台和基礎架構元件都有已識別擁有者，而且每個流程和程序都有負責其定義的已識別擁有者，以及負責其執行的擁有者。透過了解每個元件、流

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

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

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

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

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

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

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

OPS 1: 如何決定您的優先事項？
每個人都必須了解自己在實現商業價值過程中的角色。擁有共同目標以設定資源優先順序。這會充分發揮您所做努力的優勢。
OPS 2: 如何建構組織以支援業務成果？
您的團隊必須了解其在達成業務成果中所扮演的角色。團隊需要了解自己在促成其他團隊成功的過程中所扮演的角色、其他團隊在促進其成功的過程中所扮演的角色，以及擁有共同目標。了解責任、擁有權、決策方式，以及誰有權制定決策，將有助於找到工作重點，並充分發揮團隊的優勢。
OPS 3: 您的組織文化如何支援您的業務成果？
為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。

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

準備

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

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

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

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

評估工作負載、流程、程序及人員的營運準備度，以了解與工作負載相關的營運風險。您應使用一致的程序 (包括手動或自動檢查清單) 來獲悉工作負載或變更執行就緒的時間。

這樣一來，您也將能尋找任何需要您制定解決方案的領域。具備可記錄例行活動的執行手冊，以及可指引問題解決程序的程序手冊。了解收益和風險，以做出明智決策，讓變更順利進入生產環境。

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

下列問題著重於 卓越營運 方面的這些考量。

OPS 4: 您如何設計工作負載以便了解其狀況？
設計工作負載，以便它為您提供了解其內部狀態所需的跨全部元件 (例如指標、日誌和追蹤) 的資訊。這讓您能在適當時機提供有效回應。
OPS 5: 您如何減少缺陷、幫助輕鬆修復，以及改善生產流程？
採用改善改變生產流程的方法，藉此重構、快速提供品質意見回饋及修復錯誤。這會加快有助益的改變發揮作用的速度、限制部署問題，並快速識別和修復部署活動造成的問題。
OPS 6: 您如何緩解部署風險？
採用可快速提供品質意見回饋，並從成果不盡理想的改變中快速復原的方法。使用這些實務可緩解部署變更所帶來問題的影響。
OPS 7: 您如何知道自己準備好支援工作負載？
評估工作負載、流程和程序及人員的營運準備度，了解工作負載相關營運風險。

對以程式碼實作營運活動進行投資，從而最大程度地提高營運人員的生產力，將錯誤率降至最低以及實現自動回應。使用「事前剖析」可預測失敗並適時建立程序。依照一致的標記策略，使用資源標籤和 AWS Resource Groups 來套用中繼資料，以識別您的資源。標記您的資源，以用於組織、成本會計、存取控制，以及將自動執行營運活動設為目標。採用可利用雲端彈性的部署實務，以促進開發活動和系統的預部署，進而加快實作速度。當您變更您用於評估工作負載的檢查清單時，請計劃如何處理不再合規的即時系統。

操作

我們可根據業務和客戶成果的實現情況，衡量是否成功運作工作負載。定義預期成果，確定如何衡量成功，並識別可用於這些計算的指標，以判斷您的工作負載和營運是否成功。營運運作狀態包括工作負載的運作狀態，以及為支援工作負載所執行營運活動 (例如，部署和事件回應) 的運作狀態和成功情況。建立指標基準以便進行改善、調查和介入；收集並分析指標；然後，驗證您對營運成功及其隨著時間的變化情況的理解。使用收集的指標來判斷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已充分了解的事件，並使用程序手冊協助調查和解決問題。根據事件對業務和客戶的影響來確定回應事件的優先順序。確保如因回應事件而發出

提醒，則將由明確識別的擁有者執行關聯程序。事先定義解決事件所需的人員，並納入向上呈報觸發條件，以在必要時根據緊迫性和影響力，在其中新增額外的參與人員。識別並邀請具有權限的個人來決定行動方案，該方案將受到先前未解決的事件回應的業務影響。

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

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

下列問題著重於 卓越營運 方面的這些考量。

OPS 8: 您如何了解工作負載的運作狀態？
定義、擷取和分析工作負載指標，掌握工作負載事件，以便採取適當行動。
OPS 9: 您如何了解營運狀況？
定義、擷取和分析營運指標，掌握營運事件，以便採取適當行動。
OPS 10: 您如何管理工作負載和營運事件？
準備和驗證回應事件的程序，大幅降低工作負載中斷情形。

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

演進

您必須學習、分享和持續改善以維持卓越營運。投入工作週期以持續逐漸改善。針對所有影響客戶的事件執行事件後分析。確定成因和預防措施，限制或防止其再次發生。視情況與受影響的社群溝通成因。定期評估改進機會 (例如，功能請求、問題修復和合規要求) 並確定其優先順序，包括工作負載和營運程序。在您的程序中納入回饋迴圈，以快速識別有待改善的領域並從營運執行中獲得經驗。

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

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

下列問題著重於 卓越營運 方面的這些考量。

OPS 11: 您如何改善營運？ 投入時間和資源持續逐漸改善，以加強營運的效果和效率。

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

資源

請參閱以下資源，進一步了解我們的 卓越營運 最佳實務：

文件

- DevOps and AWS

白皮書

- Operational Excellence Pillar

影片

- DevOps at Amazon

安全性

安全性支柱包含安全性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。

安全性支柱概述了設計原則、最佳實務和相關問題。您可以在安全性支柱白皮書中找到實作的指引。

設計原則

有 seven 項雲端安全性設計原則：

- 建立強大的身份識別基礎: 實作最低授權原則，並對於每個與 AWS 資源的互動強制執行職責與適當的授權分離。集中化身分管理，旨在消除對長期靜態登入資料的依賴。
- 啟用可追溯性: 即時監控、提醒和稽核動作和對您環境的變更。將日誌和指標收集與系統進行整合，以自動調查並採取動作。
- 在所有層套用安全性: 使用多個安全控制套用深度防禦方法。套用至所有層級 (例如，網路邊緣、VPC、負載平衡、每個執行個體和運算服務、作業系統、應用程式和程式碼)。

- 自動化安全最佳實務: 將基於軟體的安全性機制自動化，可提高您安全、快速和以具成本效益的方式擴展的能力。建立安全架構 (包括實作控制) 在版本控制的範本中作為程式碼定義和管理。
- 保護傳輸中資料和靜態資料: 將您的資料分為不同的敏感性等級，並使用適當的機制，例如加密、權杖化及存取控制。
- 讓人員遠離資料: 使用機制和工具，來降低或消除對直接存取或手動處理資料的需要。在處理敏感資料時，這降低了處理不當或修改以及人為錯誤的風險。
- 為安全事件做準備: 為事故做好萬全準備，建立與您組織的要求吻合的事故管理和調查政策與程序。執行失敗回應模擬和使用工具與自動化，以提高偵測、調查和復原的速度。

定義

有 six 個雲端安全性最佳實務方面：

- 安全性
- 身份和存取管理
- 偵測
- 基礎設施保護
- 資料保護
- 事故回應

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

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

最佳實務

安全性

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

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

下列問題著重於安全性方面的這些考量。 (如需安全性問題清單和最佳實務，請參閱附錄。).

SEC 1: 如何安全地操作工作負載？

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

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

身份和存取管理

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

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

下列問題著重於安全性方面的這些考量。

SEC 2: 如何管理人員和機器的身分？

處理操作安全的 AWS 工作負載時，您需要管理兩種身分類型。了解您需要管理和授予存取權的身分類型，有助於確保正確的身分在適當的條件下存取正確的資源。人員身分：您的管理員、開發人員、操作員和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些人是組織的成員，或與您協作的外部使用者，以及透過 Web 瀏覽器、用戶端應用程式或互動式命令列工具與 AWS 資源互動的使用者。機器身分：您的服務應用程式、操作工具和工作負載需要身分，才能向 AWS 服務發出請求，例如讀取資料。這些身分包括在 AWS 環境中執行的機器，例如 Amazon EC2 執行個體或 AWS Lambda 函數。您也可以為需要存取權的外部人員管理機器身分。此外，您可能也有在 AWS 外部，需要存取 AWS 環境的機器。

SEC 3: 如何管理人員和機器的許可？

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

登入資料不得在任何使用者或系統之間共用。應使用最低權限的方法以及最佳實務 (包括密碼要求和強制執行 MFA) 來授予使用者存取權限。包括對 AWS 服務的 API 呼叫在內的程

式設計存取應使用臨時和有限權限的登入資料 (例如由 AWS Security Token Service 發出的登入資料) 執行。

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

偵測

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

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

下列問題著重於安全性方面的這些考量。

SEC 4: 您如何偵測和調查安全事件？
從日誌和指標中擷取並分析事件以掌握情況。針對安全事件和潛在威脅採取行動，有助於保護工作負載。

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

基礎設施保護

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

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

下列問題著重於安全性方面的這些考量。

SEC 5: 如何保護您的網路資源？

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

SEC 6: 您如何保護運算資源？

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

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

AWS 客戶能夠量身訂製或強化 Amazon Elastic Compute Cloud (Amazon EC2)、Amazon EC2 Container Service (Amazon ECS) 容器或 AWS Elastic Beanstalk 執行個體的組態，並將組態持久保留到一個不變的 Amazon Machine Image (AMI)。然後，無論是由 Auto Scaling 觸發還是手動啟動，使用此 AMI 啟動的所有新虛擬伺服器 (執行個體) 都將獲得此強化組態。

資料保護

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

在 AWS 中，以下實務有助於保護資料：

- 作為 AWS 客戶，您對資料保持完全控制。
- AWS 讓您可以更輕鬆地加密資料和管理金鑰，包括常規的金鑰輪換。這些可以透過 AWS 輕鬆地自動化或由您手動維護。
- 提供了包含重要內容 (例如檔案存取和變更) 的詳細記錄。
- AWS 設計的儲存系統具有卓越彈性。例如，Amazon S3 Standard、S3 Standard-IA、S3 One Zone-IA 和 Amazon Glacier 都在給定年份內提供 99.999999999% 的物件耐用性。此耐用性等級相當於 0.000000001% 物件年平均預期損失率。
- 版本控制可以作為更大的資料生命週期管理過程的一部分，可以防止意外的覆寫、刪除和類似損害。
- AWS 永遠不會主動移動區域之間的資料。除非您明確啟用相關功能或利用提供相關功能的服務，否則放置在某個區域中的內容將保留在該區域中。

下列問題著重於安全性方面的這些考量。

SEC 7: 您如何分類資料？
資料分類可讓您根據關鍵性和敏感度將資料分類，協助判定適當的保護和保留控制。
SEC 8: 您如何保護靜態資料？
實作多個控制來保護您的靜態資料，以降低未經授權的存取或不當處理的風險。
SEC 9: 您如何保護傳輸中資料？
實作多個控制以保護傳輸中的資料，減少未經授權的存取或遺失的風險。

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

事故回應

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

在 AWS 中，以下實務有助於有效地回應事故：

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

下列問題著重於安全性方面的這些考量。

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

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

資源

請參閱以下資源，進一步了解我們的安全性最佳實務：

文件

- AWS Cloud Security
- AWS Compliance
- AWS Security Blog

白皮書

- Security Pillar
- AWS Security Overview
- AWS Security Best Practices
- AWS Risk and Compliance

影片

- AWS Security State of the Union
- Shared Responsibility Overview

可靠性

可靠性支柱包含工作負載如預期般正確、一致地執行其預期功能的能力，這包括在其整個生命週期中操作和測試工作負載的能力。

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

設計原則

有 five 項雲端可靠性設計原則：

- 自動從失敗中復原: 透過監控工作負載的關鍵績效指標 (KPI)，您可在達到臨界值時觸發自動化。這些 KPI 應為業務價值的衡量指標，而非服務營運的技術方面。如此一來，即可自動通知和追蹤失敗，以及自動化可解決或修復失敗的復原程序。藉助更複雜的自動化功能，您可以在發生失敗前進行預測和修補。
- 測試復原程序: 在內部部署環境中，經常執行測試以證明工作負載可在特定情況下正常工作。測試通常不可用於驗證復原策略。在雲端，您可測試工作負載會發生哪些失敗情境，同時您可驗證復原程序。您可使用自動化來模擬不同的失敗情境或重新建立會導致之前失敗的情境。此方法會在實際的失敗情境發生前公開您可以測試和修復的失敗路徑，從而降低風險。
- 水平擴展以提高總體工作負載可用性: 使用多個小資源取代一個大資源，以降低整體工作負載上發生單一失敗時造成的影響。將請求分散在多個較小的資源中，以確保它們不會共享常見失敗點。

- 停止猜測容量: 內部部署工作負載失敗的一個常見原因是資源飽和，即當對工作負載的需求超出該工作負載的容量時發生的情況 (這通常為阻斷服務攻擊的目標)。在雲端，您可以監控需求和工作負載利用率，並自動新增或刪除資源，以保持可滿足需求的最佳水平，而不會過度佈建或佈建不足。仍然存在限制，但是某些配額可以控制，而其他限制則可管理 (請參閱管理服務配額和限制)。
- 管理自動化變更: 應使用自動化來執行對基礎設施的變更。需要管理的變更包括之後可以追蹤和審查的自動化變更。

定義

有 four 個雲端可靠性最佳實務方面：

- 基礎
- 工作負載架構
- 變更管理
- 失敗管理

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

最佳實務

基礎

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

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

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

REL 1: 您如何管理服務配額和限制？

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

REL 2: 如何規劃您的網路拓撲？

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

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

工作負載架構

可靠的工作負載始自於軟體和基礎設施的前期設計決策。您的架構選擇會對所有五大 Well-Architected 支柱的工作負載行為產生影響。為求可靠性，您必須依循特定模式。

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

下列問題著重於可靠性方面的這些考量。

REL 3: 如何設計您的工作負載服務架構？
使用服務導向架構 (SOA) 或微型服務架構，建置擴展性與可靠性高的工作負載。服務導向架構 (SOA) 是透過服務界面讓軟體元件可重複使用的做法。微型服務架構則進一步讓元件變得更小、更簡單。
REL 4: 如何在分散式系統中設計防止失敗的互動？
分散式系統倚賴通訊網路來互連元件，例如伺服器或服務。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務可防止失敗，並延長平均失敗間隔時間 (MTBF)。
REL 5: 如何設計分散式系統中的互動以緩解或承受故障？
分散式系統倚賴通訊網路來互連元件 (例如，伺服器或服務)。即使這些網路上的資料遺失或延遲，您的工作負載仍必須可靠運作。分散式系統的元件必須以不會對其他元件或工作負載造成負面影響的方式運作。這些最佳實務讓工作負載能夠承受壓力或故障，更快速地從其中復原，並減輕這類受損的影響。最終縮短平均復原時間 (MTTR)。

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。

變更管理

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

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

下列問題著重於可靠性方面的這些考量。

REL 6: 如何監控工作負載資源？
日誌和指標是深入了解工作負載運作狀態的強大工具。您可以設定工作負載以監控日誌和指標，並在超過閾值或發生重大事件時傳送通知。監控可讓您的工作負載識別何時會超過低效能閾值或發生故障，以便自動復原來回應。
REL 7: 如何設計工作負載以適應需求變更？
可擴展工作負載提供自動新增或移除資源的彈性，以便隨時盡可能符合目前需求。
REL 8: 您如何實作變更？
需有控制變更以部署新功能，並確保工作負載和運作環境執行已知軟體，且能以可預測的方式修補或取代。如果這些變更不受控制，則難以預測這些變更的效果，或是解決肇因於這些變更的問題。

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

失敗管理

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

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

下列問題著重於可靠性方面的這些考量。

REL 9: 您如何備份資料？
備份資料、應用程式和組態，以符合復原時間目標 (RTO) 和復原點目標 (RPO) 的要求。
REL 10: 如何使用故障隔離來保護您的工作負載？
故障隔離界限會在工作負載內將失敗影響限制至有限數量的元件。界限外的元件不受失敗影響。使用多個故障隔離界限時，您可以限制對工作負載的影響。
REL 11: 如何設計工作負載以承受元件失敗？
需要高可用性和低平均復原時間 (MTTR) 的工作負載必須建立彈性架構。
REL 12: 如何測試可靠性？
在將工作負載設計為可彈性應對生產壓力之後，測試是確保其依設計運作並交付您預期之彈性的唯一方法。
REL 13: 您如何規劃災難復原 (DR)？
備妥備份和冗餘工作負載元件是 DR 策略的開始。RTO 和 RPO 是您還原可用性的目標。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。

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

資源

請參閱以下資源，進一步了解我們的可靠性最佳實務：

文件

- AWS Documentation
- AWS Global Infrastructure
- AWS Auto Scaling: How Scaling Plans Work
- What Is AWS Backup?

白皮書

- Reliability Pillar: AWS Well-Architected
- Implementing Microservices on AWS

效能達成效率

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

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

設計原則

有 five 項雲端效能達成效率設計原則：

- 讓進階技術變得更普及: 將複雜的任務委派給雲端廠商，讓團隊更輕鬆地實作進階技術。與其要求 IT 團隊了解新技術的託管和執行方式，不如考慮使用技術即服務。例如，NoSQL 資料庫、媒體轉碼和機器學習均為需要專業知識的技術。在雲端，這些技術成為團隊可以使用的服務，讓團隊可專注於產品開發，而非資源佈建及管理。
- 在最短的時間部署到全球: 在全球多個 AWS 區域部署工作負載，讓您以最低的成本，為客戶提供較低的延遲和更好的體驗。

- 使用無伺服器架構: 採用無伺服器架構，您便無需執行和維護實體伺服器來完成傳統運算活動。例如，無伺服器儲存服務可以充當靜態網站 (因此無需 Web 伺服器)，而事件服務可以為您託管程式碼。如此一來，即可減輕管理實體伺服器的營運負擔，而且由於這些受管服務是在雲端規模上運行，因此還可以降低交易成本。
- 提高試驗頻率: 藉助虛擬及可自動化的資源，您可以使用不同類型的執行個體、儲存設備或組態，迅速完成比較測試。
- 考慮機械共鳴作用: 了解雲端服務的使用方式，並一律使用最符合工作負載目標的技術方法。例如，在您選擇資料庫或儲存方法時，請考慮資料存取模式。

定義

有 four 個雲端效能達成效率最佳實務方面：

- 選擇
- 審查
- 監控
- 權衡

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

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

最佳實務

選擇

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

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

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

PERF 1: 您如何選擇效能最佳的架構？
欲讓工作負載達到最佳效能通常需要採用多種方法。Well-Architected 系統會使用多重解決方案和功能以提升效能。

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

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

運算

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

AWS 中提供了三種運算形式：執行個體、容器和函數：

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

下列問題著重於效能達成效率方面的這些考量。

PERF 2: 您如何選擇運算解決方案？

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

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

儲存

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

在 AWS 中，儲存有三種形式：物件、區塊和檔案：

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

下列問題著重於效能達成效率方面的這些考量。

PERF 3: 您如何選擇儲存解決方案？

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

當您要選取儲存解決方案時，務必確保其符合您的存取模式，以達到您想要的效能。

資料庫

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

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

下列問題著重於效能達成效率方面的這些考量。

PERF 4: 您如何選擇資料庫解決方案？

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

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

網路

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

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

下列問題著重於效能達成效率方面的這些考量。

PERF 5: 您如何設定聯網解決方案？

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

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

審查

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

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

下列問題著重於效能達成效率方面的這些考量。

PERF 6: 您如何發展工作負載，以運用新版本的優勢？

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

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

監控

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

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

下列問題著重於效能達成效率方面的這些考量。

PERF 7: 您如何監控資源來確保達成預期效能？

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

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

權衡

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

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

下列問題著重於效能達成效率方面的這些考量。

PERF 8: 您如何採用權衡來增進效能？
架構解決方案時，判斷權衡項目可讓您選擇最佳方法。您通常可以透過權衡一致性、耐用性和時間與延遲的空間來提升效能。

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

資源

請參閱以下資源，進一步了解我們的效能達成效率最佳實務：

文件

- Amazon S3 Performance Optimization
- Amazon EBS Volume Performance

白皮書

- Performance Efficiency Pillar

影片

- AWS re:Invent 2019: Amazon EC2 foundations (CMP211-R2)
- AWS re:Invent 2019: Leadership session: Storage state of the union (STG201-L)
- AWS re:Invent 2019: Leadership session: AWS purpose-built databases (DAT209-L)
- AWS re:Invent 2019: Connectivity to AWS and hybrid AWS network architectures (NET317-R1)
- AWS re:Invent 2019: Powering next-gen Amazon EC2: Deep dive into the Nitro system (CMP303-R2)
- AWS re:Invent 2019: Scaling up to your first 10 million users (ARC211-R)

成本優化

成本優化支柱包含在最低價格之下執行系統以產生商業價值的能力

成本優化支柱概述了設計原則、最佳實務和相關問題。您可以在成本優化支柱白皮書中找到實作的指引。

設計原則

有 five 項雲端成本優化設計原則：

- 實作雲端財務管理: 為實現財務成功並加速在雲端實現商業價值，您需要投資雲端財務管理/成本優化。您的組織需要投入時間和資源，在這個新的技術與使用管理領域中打造能力。與您的安全性或營運能力類似，您需要透過知識累積、計畫、資源和程序打造能力，以成為具成本效率的組織。
- 採用消費模式: 僅為您需要的運算資源付費，依照業務要求增減用量，不必倚賴複雜的預測。例如，開發與測試環境通常僅於一週工作日的一天八小時當中使用。您可在不使用這些資源時加以停止，有潛力可節省 75% 成本 (40 小時相對於 168 小時)。
- 衡量整體效率: 測量工作負載的商業輸出和遞送的相關成本。以此測量值可得知您從增加輸出與降低成本獲取的增益。
- 停止將金錢花在繁重的無差別工作上: AWS 會處理資料中心營運的繁重工作，例如架設、堆疊和支援伺服器。通過受管服務，同時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 基礎架構。
- 分析和歸因支出: 採雲端式能更容易準確識別系統的用量和成本，繼而允許將 IT 成本透明化地歸因至個別工作負載擁有者。如此有助於測量投資報酬率 (ROI)，並且讓工作負載擁有者有機會優化資源和降低成本。

定義

有 five 個雲端成本優化最佳實務方面：

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

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

當地體現雲端的經濟效益。以下各節提供初始和持續實作工作負載雲端財務管理和成本優化的技術和最佳實務。

最佳實務

實作雲端財務管理

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

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

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

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

COST 1: 如何實作雲端財務管理？
透過實作雲端財務管理，組織可以透過優化成本和用量以及在 AWS 上進行規模調整，實現商業價值和財務上的成功。

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

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

支出和用量感知

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

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

在 AWS 中，您可以使用 AWS Organizations 或 AWS Control Tower 來建立帳戶結構，如此可實現區隔並協助您分配成本和用量。您也可以對資源使用標記，利用商業和組織資訊確定用量和成本情況。使用 AWS Cost Explorer 查看您的成本和用量，或使用 Amazon Athena 和 Amazon QuickSight 建立自訂儀表板和分析。透過 AWS 預算的通知，以及使用 AWS Identity and Access Management (IAM) 和 Service Quotas 的控制措施，控制成本和用量。

下列問題著重於成本優化方面的這些考量。

COST 2: 您如何管控用量？
建立原則和機制以確保產生的成本合理，同時達成目標。您可以運用相互制衡的方法，在不超支的情況下創新。
COST 3: 您如何監控用量和成本？
建立原則和程序以監控並適當分配成本。這可讓您衡量並改善此工作負載的成本效益。
COST 4: 如何進行資源除役？
從啟動到結束專案期間，控制變更並管理資源。這可確保您關閉或終止未使用的資源，以減少浪費。

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

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

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

具有經濟效益的資源

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

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

AWS 備有各種具有彈性且經濟的定價選項，讓您以最符合需要的方式獲取 EC2 和其他服務的執行個體。隨需執行個體讓您可以按時數為運算容量付費，無最低承諾的要求。Savings Plans 和預留執行個體與隨需定價相較，可節省高達 75% 的成本。使用 Spot 執行個體，您可善用未用的 Amazon EC2 容量，與隨需定價相較可節省高達 90% 的成本。Spot

執行個體適合用在系統能耐受使用伺服器叢集之處，其中個別伺服器能動態性地來去，例如無狀態 Web 伺服器、批次處理，或使用 HPC 和大型資料時。

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

下列問題著重於成本優化方面的這些考量。

COST 5: 您選擇服務時如何評估成本？
Amazon EC2、Amazon EBS 和 Amazon S3 是基礎 AWS 服務。Amazon RDS 和 Amazon DynamoDB 等受管服務為更高等級或應用程式等級的 AWS 服務。選擇適當的基礎和受管服務，您便可為成本最佳化此工作負載。舉例來說，您可以使用受管服務減少或省下大部分管理和營運開銷，讓您從事應用程式或企業相關活動。
COST 6: 您選擇資源類型、大小和數量時，如何達成成本目標？
確保您為手上的任務選擇適當的資源大小和資源數量。您透過選擇最具成本效益的類型、大小和數量，最大限度地減少浪費。
COST 7: 您如何使用定價模式降低成本？
使用最適合您資源的定價模式，大幅減少支出。
COST 8: 您如何規劃資料傳輸費？
務必規劃和監控資料傳輸費，以便做出可大幅減少成本的架構決策。小但有效的架構變更可隨時間大幅減少營運成本。

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

管理需求與供應資源

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

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

下列問題著重於成本優化方面的這些考量。

COST 9: 如何管理需求和供應資源？
針對支出和效能達到平衡的工作負載，請確保使用購買的每個項目，並避免極少使用執行個體。往任一端傾斜的使用指標，對您組織在營運成本 (因過度使用而降低效能) 或浪費的 AWS 花費 (因過度佈建) 方面會造成負面影響。

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

隨時間優化

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

透過實作新功能或資源類型可逐步優化工作負載，同時盡量減少實作變更所需的工作量。這可隨著時間持續提高效率，並確保您持續使用最新的技術來降低營運成本。您也可以使用新的服務來取代工作負載中的元件，或將新元件新增至工作負載中。這可以大幅提高效率，因此定期檢閱工作負載並實作新服務和功能至關重要。

下列問題著重於成本優化方面的這些考量。

COST 10: 您如何評估新服務？ 隨著 AWS 推出新服務和功能，最佳實務是檢視現有架構決策，以確保持續發揮最大成本效益。

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

資源

請參閱以下資源，進一步了解我們的成本優化最佳實務：

文件

- AWS Documentation

白皮書

- Cost Optimization Pillar

審查程序

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

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

AWS Well-Architected 符合 AWS 於內部審查系統與服務的方式。其所根據的前提為能影響架構方針的一套設計原則，並提出問題，確保人員不致於忽略根本原因分析 (RCA) 中經常列為重點的領域。每當內部系統、AWS 服務或客戶有明顯問題，我們都會查看 RCA，了解是否能提升所使用的審查程序。[1]

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。正式運作之後，您的工作負載可隨著新增功能和變更技術實作而繼續演進。工作負載的架構會隨時間而變化。您需要遵守良好的衛生實務，以阻止您推動演進的同時，其架構上的特性隨之衰退。在您作出重要的架構變更時，應遵照一套衛生程序，包括 Well-Architected 審查。

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

開會時的一些建議項目如下：

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

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

[1] 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。

- “我們太忙！” (團隊預備進行盛大推出時，往往會這麼說。)
  - 既然預備進行盛大推出，一定希望過程能夠順利。審查可讓您了解可能漏掉的任何問題。
  - 建議您在產品生命週期之中及早實施審查，以發現風險並開發配合功能遞送藍圖的減緩計劃。
- “就算有結果，我們也沒有時間作出任何行動！” (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
- “We don’t want others to know the secrets of our solution implementation!”
  - 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限閱資訊。

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

結論

AWS Well-Architected 架構提供了遍及五大支柱的架構最佳實務，用於設計和營運可靠、安全、有效率且經濟實惠的雲端系統。該架構提供一套問題，允許您審查現有或提議的架構。並且也為各支柱提供一套 AWS 最佳實務。在您的架構中使用該架構可協助您產生穩定且有效率的系統，讓您能夠專注於功能需求。

作者群

協力完成本文件的個人與組織如下：

- Rodney Lester: 任Amazon Web Services Well-Architected 資深經理
- Brian Carlson: Amazon Web Services Well-Architected 營運主管
- Ben Potter: Amazon Web Services Well-Architected 安全主管
- Eric Pullen: Amazon Web Services Well-Architected 效能主管
- Seth Eliot: Amazon Web Services Well-Architected 可靠性主管
- Nathan Besh: Amazon Web Services Well-Architected 成本主管
- Jon Steele: Amazon Web ServicesAmazon Web Services 資深技術客戶經理
- Ryan King: Amazon Web Services 技術計劃經理
- Erin Rifkin: Amazon Web Services 資深產品經理
- Max Ramsay: Amazon Web Services 首席安全解決方案架構師
- Scott Paddock: Amazon Web Services 安全解決方案架構師
- Callum Hughes: Amazon Web Services 解決方案架構師

深入閱讀

AWS Cloud Compliance

AWS Well-Architected Partner program

AWS Well-Architected Tool

AWS Well-Architected homepage

Cost Optimization Pillar whitepaper

Operational Excellence Pillar whitepaper

Performance Efficiency Pillar whitepaper

Reliability Pillar whitepaper

Security Pillar whitepaper

The Amazon Builders' Library

文件修訂

Table 2. 主要修訂：

日期	描述
2020 年 7 月	審查並重新撰寫大多數的問題和答案。
2019 年 7 月	新增 AWS Well-Architected Tool，連結至 AWS Well-Architected 實驗室及 AWS Well-Architected 合作夥伴、小處修復以促成架構有多種語言版本。
2018 年 11 月	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 6 月	更新以簡化問題文字，將答案標準化，並提升可讀性。
2017 年 11 月	卓越營運移至支柱前端並重新撰寫，使其成為其他支柱的框架。重新整理其他支柱，以反映 AWS 的演進。
2016 年 11 月	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。
2015 年 11 月	以目前的 Amazon CloudWatch Logs 資訊更新附錄。
2015 年 10 月	原始發布。

附錄：問題與最佳實務

卓越營運

組織

OPS 1 如何決定您的優先事項？

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

最佳實務:

- 評估外部客戶需求: 讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定將工作重點放在哪些外部客戶需求上。這將確保您對實現想要的業務成果所需的營運支援有透徹的了解。
- 評估內部客戶需求: 讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定將工作重點放在哪些內部客戶需求上。這將確保您對實現業務成果所需的營運支援有透徹的了解。
- 評估管控要求: 確保您了解由貴組織所定義的、可能要求或強調特定重點的準則或義務。評估內部因素，例如組織政策、標準和要求。確認您設有確定管控變更的機制。如果未確定管控要求，請確保您已對此決定進行盡職調查。
- 評估合規要求: 評估外部因素，例如合規要求和產業標準，以確保您了解可能要求或強調特定重點的準則或義務。如果未確定合規要求，請確保對此決定進行盡職調查。
- 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。
- 評估權衡: 評估在相互衝突的利益或替代方法之間做出權衡的影響，以幫助您在確定工作重點或選擇行動方案時做出明智的決定。例如，新功能加速上市可能是成本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，以便更輕鬆地遷移系統，而非遷移至針對您的資料類型最佳化的資料庫並更新您的應用程式。
- 管理收益和風險: 管理收益和風險，以便在確定工作重點時做出明智的決定。例如，部署具有未解決問題的工作負載可能有益，以便可以為客戶提供重要的新功能。相關風險可能得以減輕，也可能出現無法接受風險存在的事實，在此情況下，您將需要採取動作來解決風險。

OPS 2 如何建構組織以支援業務成果？

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

最佳實務:

- 已為資源識別擁有者: 了解誰擁有各個應用程式、工作負載、平台和基礎架構元件、該元件提供什麼商業價值，以及該擁有權為何存在。透過了解這些個別元件的商業價值，以及其如何支援業務成果，可得知對元件套用的流程和程序。
- 已為流程和程序識別擁有者: 了解誰具有個別流程和程序的擁有權、為何使用特定流程和程序，以及為何該擁有權存在。了解使用特定流程和程序的原因，能夠幫助發現改進機會。
- 已為營運活動識別負責其效能的擁有者: 了解誰負責在已定義的工作負載上執行特定活動，以及為什麼該責任存在。透過了解誰負責執行活動，可得知誰將會進行活動、驗證結果，以及提供回饋給活動擁有者。
- 團隊成員知道他們負責的項目: 透過了解您角色的責任以及您為業務成果做出貢獻的方式，可得知任務的優先順序以及您的角色為何很重要。如此可讓團隊成員辨識需求並適當地回應。
- 存在機制用來識別責任和擁有權: 如果沒有識別個人或團隊，就會有定義的向某人向上呈報的路徑，該人員有權指派擁有權或為需解決的需求進行規劃。
- 存在用於請求新增、變更和例外狀況的機制: 您可以向流程、程序和資源的擁有者提出請求。評估收益和風險後，若可行並經判斷是合適的行為，則應制定明智的決策以核准請求。
- 團隊之間的責任為預先定義或經過協商: 團隊間對於如何相互配合及支援會定義或協商協議 (例如，回應時間、服務等級目標或服務等級協議)。透過了解團隊工作對於業務成果和其他團隊及組織成果的影響，可得知其任務的優先順序，並讓他們能做出適當的回應。

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:

- 高層的支持: 資深領導階層清楚地設定對組織的期望並評估成功情況。資深領導階層是採用最佳實務和組織演進的發起者、倡導者和推動者
- 授權團隊成員在成果有風險時採取動作: 工作負載擁有者已定義指引和範圍，授權團隊成員在成果有風險時做出回應。當事件超出定義的範圍時，採取向上呈報機制來取得方向。
- 鼓勵向上呈報: 如果團隊成員認為成果有風險，則其機制可協助將疑慮向上呈報至決策制定者和利害關係人，而且我們鼓勵這麼做。應該儘早且經常向上呈報，以便識別風險，並防止風險引發事件。
- 溝通需及時、清楚且可行: 存在的機制可用來及時通知團隊成員已知的風險和計劃的事件。提供必要的內容、詳細資訊和時間 (如果可能) 來支援判斷是否需要採取動作、需要什麼動作，並及時採取動作。例如，提供軟體漏洞的通知，以便加快修補的速度，或提供計劃的銷售促銷活動的通知，如此就能實作變更凍結，避免服務中斷的風險。
- 鼓勵進行試驗: 試驗可加速學習，讓團隊成員保持興趣和參與度。不理想的結果是成功的試驗，因其已識別出不會助力成功的路徑。團隊成員不會因取得不理想結果的成功試驗而受懲罰。必需要經由試驗才能實現創新，並讓想法轉化為成果。
- 團隊成員得以並受到鼓勵來維持和發展自己的技能集: 團隊必須發展自己的技能集，以採用新技術，並支援需求和責任的變更，以支援您的工作負載。新技術的技能成長通常是團隊成員滿意度的來源，並可支援創新。支援團隊成員追求和維持產業認證，以驗證和認可他們不斷成長的技能。交叉培訓以促進知識轉移，並在失去熟練的、經驗豐富且具備機構知識的成員時，降低重大影響的風險。提供學習專用的結構化時間。
- 適當地為團隊提供資源: 維持團隊成員能力，並提供工具和資源，以支援您的工作負載需求。為團隊成員指派過多的任務會增加因人為錯誤所造成的事件風險。對工具和資源的投資 (例如，為經常執行的活動提供自動化) 可以提高團隊的有效性，讓他們能夠支援其他的活動。
- 鼓勵並尋求來自團隊內部和跨團隊的多樣化建議: 利用跨組織的多樣性，尋求多種獨特的觀點。使用此觀點來增加創新、挑戰假設，並降低確認偏差的風險。在團隊中增加包容性、多樣性和可及性，以獲得有益的觀點。

準備

OPS 4 您如何設計工作負載以便了解其狀況？

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

最佳實務:

- 實作應用程式遙測: 在您的應用程式程式碼中部署監控機制，以發出有關其內部狀態、狀況和業務成果實現的資訊。例如，佇列深度、錯誤訊息和回應時間。使用此資訊來確定何時需要回應。
- 實作和設定工作負載遙測: 設計和設定您的工作負載，以發出有關其內部狀態和當前狀況的資訊。例如，API 呼叫量、HTTP 狀態碼和擴展事件。使用此資訊來協助確定何時需要回應。
- 實作使用者活動遙測: 檢測您的應用程式程式碼，以發出有關使用者活動的資訊 (例如，點按流或已開始、已放棄和已完成的交易)。使用此資訊來了解應用程式如何被使用、使用模式以及確定何時需要回應。
- 實作相依性遙測: 設計和設定您的工作負載，以發出有關其相依資源之狀態 (例如，可達性或回應時間) 的資訊。外部相依性的範例可包含外部資料庫、DNS 和網路連線。使用此資訊來確定何時需要回應。
- 實作交易可追溯性: 實作您的應用程式程式碼並設定您的工作負載元件，以發出有關整個工作負載交易流的資訊。使用此資訊來確定何時需要回應，並幫助確定問題的根本原因。

OPS 5 您如何減少缺陷、幫助輕鬆修復，以及改善生產流程？

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

最佳實務:

- 使用版本控制: 使用版本控制來追蹤變更和發佈。
- 測試並驗證變更: 測試和驗證變更以幫助限制和偵測錯誤。自動化測試以減少由手動程序引起的錯誤，並減少測試工作量。
- 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。
- 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。
- 執行修補程式管理: 執行修補程式管理以獲取功能，解決問題並保持遵循管控。自動化修補程式管理，以減少由手動程序引起的錯誤，並減少修補工作量。
- 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。
- 實作用於提高程式碼品質的實務: 實作實務以提高程式碼品質並將缺陷降至最少。例如，測試驅動的開發、程式碼檢閱和標準採用。
- 使用多個環境: 使用多個環境進行實驗、開發和測試您的工作負載。當環境接近生產環境時使用更高的控制等級，以確保您的工作負載在部署後將按預期執行。
- 進行頻繁、細微和可逆的變更: 頻繁、細微和可逆的變更會縮小變更的範圍和影響。這樣可以簡化故障診斷，實現更快的修復，並提供回復變更的選項。
- 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。

OPS 6 您如何緩解部署風險？

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

最佳實務:

- 為失敗變更進行規劃: 計劃在變更未達到理想成果時，恢復到已知的良好狀態，或者在生產環境中進行補救。透過這樣準備可加快回應速度，以縮短復原時間。
- 測試並驗證變更: 在生命週期所有階段測試變更並驗證結果，以確認新功能，並將失敗部署的風險和影響降至最低。
- 使用部署管理系統: 使用部署管理系統來追蹤和實作變更。此舉可減少由手動程序引起的錯誤，以及部署變更的工作量。
- 使用有限的部署進行測試: 繼續現有系統現有系統的同時，在有限的部署中進行測試，以在大規模部署之前確認理想成果達成與否。例如，使用部署 Canary 測試或一體式部署。
- 使用平行環境進行部署: 在平行環境中實作變更，然後轉換到新環境。維護先前的環境，直到確認已成功部署為止。此舉可透過復原到先前的環境來將復原時間縮短至最少。
- 部署頻繁、細微和可逆的變更: 透過頻繁、細微和可逆的變更來縮小變更範圍。透過回復變更，可以更輕鬆地進行故障診斷並加快修復速度。
- 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。
- 自動化測試和復原: 自動測試部署的環境，以確認理想成果達成與否。當無法實現結果時，自動還原到先前的良好狀態，以最大限度縮短還原時間，並減少由手動程序引起的錯誤。

OPS 7 您如何知道自己準備好支援工作負載？

評估工作負載、流程和程序及人員的營運準備度，了解工作負載相關營運風險。

最佳實務:

- 確保人員能力: 建立一種機制，用於驗證您有適當數量受過培訓的人員來為營運需求提供支援。培訓人員並根據需要調整人員能力，以保持有效的支援。
- 確保對營運準備度進行一致的審查: 確保對營運工作負載的準備度進行一致的審查。審查必須至少包括團隊和工作負載的營運準備度，以及安全性要求。在程式碼中實作審查活動，並在適當的情況下觸發自動審查來回應事件，以確保一致性、執行速度並減少由手動程序引起的錯誤。
- 使用執行手冊執行程序: 執行手冊是實現特定結果的書面程序。透過在執行手冊中記錄程序，對熟知的事件做出一致且迅速的回應。將執行手冊實作為程式碼，並在適當的情況下觸發執行手冊的執行來回應事件，以確保一致性、加快回應速度並減少由手動程序引起的錯誤。
- 使用程序手冊來調查問題: 在程序手冊中記錄調查程序，以對未充分了解的問題實現一致且迅速的回應。程序手冊是為識別造成失敗情境的因素所執行的預先定義步驟。在確定或向上呈報問題之前，任何程序步驟的結果都用於確定要採取的後續步驟。
- 做出部署系統和變更的明智決策: 評估團隊支援工作負載的能力以及工作負載對管控的遵從性。在確定是否轉換系統或將系統投入生產時，比照這些評估部署的收益。了解收益和風險，以做出明智決策。

操作

OPS 8 您如何了解工作負載的運作狀態？

定義、擷取和分析工作負載指標，掌握工作負載事件，以便採取適當行動。

最佳實務:

- 識別關鍵績效指標: 根據所需的業務成果 (例如，訂單率、客戶保留率以及獲利與營運支出的對比) 與客戶成果 (例如，客戶滿意度)，識別關鍵績效指標 (KPI)。評估 KPI 以確定工作負載是否成功。
- 定義工作負載指標: 定義工作負載指標以衡量 KPI 的實現情況 (例如，捨棄的購物車、下單的訂單、成本、價格和分配的工作負載支出)。定義工作負載指標以衡量工作負載的運作狀態 (例如，界面回應時間、錯誤率、提出的請求、完成的請求和使用率)。評估指標以判斷工作負載是否取得了預期的成果，並了解工作負載的運作狀態。
- 收集和分析工作負載指標: 定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。
- 建立工作負載指標基準: 為指標建立基準，以提供期望值，做為比較和識別效能欠佳和過剩的元件的基礎。識別用於改善、調查和介入的閾值。
- 了解工作負載的預期活動模式: 建立工作負載活動模式以識別異常行為，以便您可以在需要時做出適當回應。
- 在工作負載結果有風險時發出提醒: 當工作負載結果有風險時發出提醒，以便您可以在必要時做出適當的回應。
- 在偵測到工作負載異常時發出提醒: 當偵測到工作負載異常時發出提醒，以便您可以在必要時做出適當的回應。
- 驗證結果的實現以及 KPI 和指標的有效性 : 建立工作負載營運的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。

OPS 9 您如何了解營運狀況？

定義、擷取和分析營運指標，掌握營運事件，以便採取適當行動。

最佳實務:

- 識別關鍵績效指標: 根據所需的業務 (例如，交付的新功能) 和客戶成果 (例如，客戶支援案例)，識別關鍵績效指標 (KPI)。評估 KPI 以確定營運是否成功。
- 定義營運指標: 定義營運指標以衡量 KPI 的實現情況 (例如，成功部署和失敗部署)。定義營運指標以衡量營運活動的運作狀態 (例如，偵測事件所需的平均時間 (MTTD)，以及從事件中復原所需的平均時間 (MTTR))。評估指標以判斷營運是否取得理想成果，並了解您的營運活動的運作狀態。
- 收集和分析營運指標: 定期對指標進行主動審查，以確定趨勢並確定需要在哪些地方採取適當回應。
- 建立營運指標基準: 為指標建立基準，以提供期望值，做為比較和識別效能欠佳和過剩的營運活動的基礎。
- 了解營運活動的預期模式: 建立營運活動模式以識別異常活動，以便您可以在必要時做出適當的回應。
- 在營運成果有風險時發出提醒: 當營運成果有風險時發出提醒，以便您可以在必要時做出適當的回應。
- 在偵測到營運異常時發出提醒: 在偵測到營運異常時發出提醒，以便您可以在必要時做出適當的回應。
- 驗證結果的實現以及 KPI 和指標的有效性 : 建立營運活動的業務層級檢視，以幫助您確定需求是否得到滿足，並確定需要改進以實現業務目標的領域。驗證 KPI 和指標的有效性，並在必要時進行修訂。

OPS 10 您如何管理工作負載和營運事件？

準備和驗證回應事件的程序，大幅降低工作負載中斷情形。

最佳實務:

- 使用程序進行事件、事故和問題管理: 建立處理已觀察到的事件、需要介入的事件 (事故) 和需要介入且重複發生或當前無法解決的事件 (問題) 的程序。透過這些程序，做出及時和適當的回應，以減輕這些事件對業務和客戶的影響。
- 每個提醒建立一個程序: 對於引發提醒的任何事件，建立明確定義的回應 (執行手冊或程序手冊)，並指明。此舉可確保對營運事件的有效而迅速的回應，並防止需採取動作的事件被無價值的通知所淹沒。
- 根據業務影響確定營運事件的優先順序: 確保在有多個事件需要介入時，首先解決對業務最重要的事件。例如，影響可能包括人員傷亡、經濟損失或聲譽或信用受損。
- 定義向上呈報路徑: 在您的執行手冊和程序手冊中定義向上呈報路徑，包括觸發向上呈報的條件以及向上呈報的程序。明確確定每個動作的擁有者，以確保對營運事件做出迅速有效的回應。
- 啟用推送通知: 就您的使用者所用之服務受到影響以及服務再次恢復正常，直接與使用者溝通 (例如，透過電子郵件或簡訊)，以便使用者能夠採取適當動作。
- 透過儀表板傳達狀態: 提供針對其目標受眾 (例如，內部技術團隊、領導和客戶) 量身定制的儀表板，以傳達業務的當前營運狀態，並提供感興趣的指標。
- 自動回應事件: 自動對事件進行回應，以減少由手動程序引起的錯誤，並確保快速一致的回應。

演進

OPS 11 您如何改善營運？

投入時間和資源持續逐漸改善，以加強營運的效果和效率。

最佳實務:

- 建立持續改進程序: 定期評估改進機會並排定其優先順序，以專注於它們可在其中提供最大效益的工作。
- 執行事件後分析: 審查影響客戶的事件，並識別成因和預防性措施。使用此資訊來制定緩解措施，以限制或避免事件再次發生。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。
- 實作回饋迴圈: 在程序和工作負載中包含回饋迴圈，以幫助您識別問題和需要改進的領域。
- 執行知識管理: 存在的機制讓您的團隊成員可以及時探索他們所需的資訊、存取資訊，並識別其是否為最新且完整的資訊。存在的機制是用來識別所需的內容、需要重新整理的內容，以及應存檔的內容，以便該內容不再供其他人參考。
- 定義改進驅動因素: 確定改進驅動因素，以幫助您評估改進機會並排定其優先順序。
- 驗證洞見: 與跨職能團隊和企業擁有者一起審查您的分析結果和回應。透過這些審查建立共識，確定其他影響並確定行動方案。適當調整回應。
- 執行營運指標審查: 與來自不同業務領域的跨團隊參與者定期進行營運指標的追溯性分析。透過這些審查確定改進機會、可能的行動方案並分享獲得的經驗。
- 記錄和分享獲得的經驗: 記錄並分享從執行營運活動中獲得的經驗，以便您可以在內部以及跨團隊使用它們。
- 分配改進時間: 在流程中投入時間和資源，以持續逐漸改善。

安全性

安全性

SEC 1 如何安全地操作工作負載？

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

最佳實務:

- 使用帳戶區隔工作負載: 根據函數或一組常用的控制項，在個別的帳戶和群組帳戶中管理工作負載，而不是複製公司的報告結構。先從安全與基礎設施開始，讓您的組織隨著工作負載的成長設定常見的防護機制。
- 安全的 AWS 帳戶: 例如，透過啟用 MFA 和限制根使用者的使用，來保護帳戶的存取權，並設定帳戶聯絡人。
- 識別和驗證控制目標: 根據合規要求以及從威脅模型識別的風險，獲得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。
- 及時了解安全威脅: 透過隨時得知最新安全威脅來辨識攻擊媒介，協助您定義並實作適當的控制。
- 及時了解安全建議的最新資訊: 隨時掌握 AWS 和產業安全建議的最新資訊，以發展工作負載的安全狀態。
- 自動化管道中安全控制的測試和驗證: 為安全機制建立安全的基準和範本，這些機制會在建置、管道和程序中進行測試和驗證。使用工具和自動化，持續測試和驗證所有安全控制。例如，掃描機器圖像和基礎設施即程式碼範本，檢查是否有安全漏洞、異常和偏離各階段既定基準。
- 使用威脅模型識別風險並確定優先級: 使用威脅模型來識別和維護對潛在威脅的最新紀錄。排定威脅的優先順序並調整安全控制，以防止、偵測和回應威脅。在不斷演變的安全形勢下，重新審視和維護此事項。
- 定期評估和實作新的安全服務和功能: AWS 和 APN 合作夥伴會持續發佈新的功能和服務，讓您發展工作負載的安全狀態。

身份和存取管理

SEC 2 如何管理人員和機器的身分？

處理操作安全的 AWS 工作負載時，您需要管理兩種身分類型。了解您需要管理和授予存取權的身分類型，有助於確保正確的身分在適當的條件下存取正確的資源。人員身分：您的管理員、開發人員、操作員和最終使用者需要身分才能存取您的 AWS 環境和應用程式。這些人是組織的成員，或與您協作的外部使用者，以及透過 Web 瀏覽器、用戶端應用程式或互動式命令列工具與 AWS 資源互動的使用者。機器身分：您的服務應用程式、操作工具和工作負載需要身分，才能向 AWS 服務發出請求，例如讀取資料。這些身分包括在 AWS 環境中執行的機器，例如 Amazon EC2 執行個體或 AWS Lambda 函數。您也可以為需要存取權的外部人員管理機器身分。此外，您可能也有在 AWS 外部，需要存取 AWS 環境的機器。

最佳實務:

- 使用強式登入機制: 強制執行密碼長度下限，並教育使用者避免使用常見密碼或重複使用密碼。透過軟體或硬體機制強制使用 Multi-Factor Authentication (MFA)，以提供額外的保護層。
- 使用臨時登入資料: 需要身分才能動態取得臨時登入資料。若是人力身分，請使用 AWS Single Sign-On 或與 IAM 角色的聯合身分來存取 AWS 帳戶。若是機器身分，要求使用 IAM 角色，而非長期存取金鑰。
- 安全地存放和使用機密: 對於需要第三方應用程式密碼等機密的人力和機器身分，請在專業服務中使用最新的產業標準，以自動輪換的方式存放機密。
- 倚賴集中化的身分供應商: 若是人力身分，請倚賴可讓您在集中位置管理身分的身分供應商。這可讓您從單一位置建立、管理和撤銷存取權，以便更輕鬆地管理存取權。這可減少多個登入資料的需求，並提供與 HR 程序整合的機會。
- 定期稽核和輪換登入資料: 當您無法倚賴臨時登入資料且需要長期登入資料時，請稽核登入資料以確保定義的控制 (例如 MFA) 會定期強制執行、輪換，且具有適當的存取層級。
- 利用使用者群組和屬性: 將具有共同安全需求的使用者放在身分供應商定義的群組中，並設置機制，以確保可用於存取控制的使用者屬性 (例如，部門或位置) 正確並已更新。使用這些群組和屬性 (而非個別使用者) 來控制存取情形。這可讓您透過一次變更使用者的群組成員資格或屬性集中管理存取權，而不是在使用者存取需求變更時更新許多個別政策。

SEC 3 如何管理人員和機器的許可？

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

最佳實務:

- 定義存取需求: 工作負載的每個元件或資源都需要由管理員、最終使用者或其他元件存取。您需要清楚定義哪些人或哪些項目應該可以存取每個元件，然後選擇適當的身分類型與身份驗證和授權方法。
- 授予最低權限存取權: 允許在特定條件下對特定 AWS 資源執行特定動作的存取權，來僅授予身分所需的存取權。倚賴群組和身分屬性，大規模動態設定許可，而不是定義個別使用者的許可。例如，您可以允許一組開發人員的存取權，以只管理其專案的資源。如此一來，將開發人員從群組移除時，在使用該群組來進行存取控制的任何地方都會撤銷該開發人員的存取權，而不需要對存取政策進行任何變更。
- 建立緊急存取程序: 在極少數的狀況下，自動化程序或管道發生問題時，允許緊急存取工作負載的程序。這可協助您倚賴最低權限的存取權，但確保使用者可在需要時取得適當的存取層級。例如，建立一個程序，讓管理員驗證和核准他們的請求。
- 持續減少許可: 當團隊和工作負載決定他們需要的存取時，請移除他們不再使用的許可，並建立檢閱程序以達到最低權限的許可。持續監控和減少未使用的身分和許可。
- 為您的組織定義許可防護機制: 建立通用控制項，限制對組織中所有身分的存取權。例如，您可以限制對特定 AWS 區域的存取權，或防止操作員刪除常見資源，例如用於中央安全團隊的 IAM 角色。
- 根據生命週期管理存取: 將存取控制與操作員和應用程式之生命週期以及集中化的聯合身分供應商相整合。例如：在使用者離開組織或變更角色時移除其存取權。
- 分析公有和跨帳戶存取: 持續監控強調公有和跨帳戶存取的問題清單。減少對需要此類存取之資源的公有存取和跨帳戶存取。
- 安全地共用資源: 管理跨帳戶或 AWS 組織內共用資源的使用量。監控共用資源並檢閱共用資源存取。

偵測

SEC 4 您如何偵測和調查安全事件？

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

最佳實務:

- 設定服務和應用程式記錄: 設定整個工作負載中的記錄，包括應用程式日誌、資源日誌和 AWS 服務日誌。例如：確定組織內的所有帳戶已啟用 AWS CloudTrail、Amazon CloudWatch Logs、Amazon GuardDuty 和 AWS Security Hub。
- 集中分析日誌、問題清單和指標: 應集中收集所有日誌、指標和遙測，並自動分析以偵測異常和未經授權活動的指標。儀表板可讓您輕鬆存取運作狀態的即時洞見。例如：確保 Amazon GuardDuty 和 Security Hub 日誌傳送到中央位置以進行提醒和分析。
- 自動回應事件: 使用自動化來調查和修復事件可減少人工作業和人為錯誤，還可讓您擴展調查功能。定期檢閱將協助您調整自動化工具並持續反覆運算。例如：將 Amazon GuardDuty 事件的回應自動化，方法是自動化第一個調查步驟，然後反覆運算以逐步消除人工作業。
- 實作可行的安全事件: 建立傳送給團隊並能讓團隊據此採取行動的提醒。確保提醒包含讓團隊採取動作的相關資訊。例如：確保 Amazon GuardDuty 和 AWS Security Hub 提醒傳送給團隊採取動作，或傳送到回應自動化工具，且團隊仍透過自動化架構的簡訊收到通知。

基礎設施保護

SEC 5 如何保護您的網路資源？

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

最佳實務:

- 建立網路層: 將共用連線能力需求的元件分組成許多層。例如：不需存取網際網路的 VPC 中的資料庫叢集，應放置在沒有往返網際網路路由的子網路中。在沒有 VPC 的情況下操作的無伺服器工作負載中，與微型服務類似的分層和區隔可以達到相同的目標。
- 控制所有層級的流量: 針對傳入和傳出流量套用深入防禦方法的控制項。例如：對於 Amazon Virtual Private Cloud (VPC)，這包括安全群組、網路 ACL 和子網路。對於 AWS Lambda，請考慮使用以 VPC 為基礎的控制在您的私有 VPC 中執行。
- 自動化網路保護: 自動化保護機制，以借助威脅情報和異常偵測提供自衛網路。例如：可以主動適應目前威脅並降低其影響的入侵偵測和預防工具。
- 實作檢查和保護: 檢查和篩選每一層的流量。例如：使用網頁應用程式防火牆，以協助防止應用程式網路層意外存取。對於 Lambda 函數，第三方工具可以新增應用程式層防火牆到您的執行時間環境。

SEC 6 您如何保護運算資源？

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

最佳實務:

- 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。
- 減少受攻擊面: 透過強化作業系統以及盡量減少使用中的元件、程式庫和外部消耗性服務來減少受攻擊面。
- 實作受管服務: 實作管理資源的服務 (例如 Amazon RDS、AWS Lambda 和 Amazon ECS)，能為您減少共同責任模式中的安全維護任務。
- 自動化運算保護: 將您的保護性運算機制自動化，包括漏洞管理、受攻擊面縮小和資源管理。
- 讓人員在遠距離執行動作: 移除互動式存取功能可降低人為錯誤的風險，並降低手動設定或管理的可能性。例如：使用變更管理工作流程以利用基礎設施即程式碼部署 EC2 執行個體，然後使用工具而不是直接存取或堡壘主機來管理 EC2 執行個體。
- 驗證軟體完整性: 實作機制 (例如程式碼簽署) 以驗證工作負載中使用的軟體、程式碼和程式庫，確保它們來自信任的來源且未遭到篡改。

資料保護

SEC 7 您如何分類資料？

資料分類可讓您根據關鍵性和敏感度將資料分類，協助判定適當的保護和保留控制。

最佳實務:

- 識別工作負載內的資料: 這包括資料的類型和分類、相關的業務程序、資料擁有者、適用的法律和合規要求、存放的位置，以及需要強制執行的結果控制項。這可能包括分類以指出資料是否打算供公開取得；資料是否僅供內部使用，例如客戶的個人識別資訊 (PII)；資料是否用於更受限制的存取，例如智慧財產、依法特權或標示為敏感資料等等。
- 定義資料保護控制: 根據資料的分類層級保護資料。例如：使用相關建議來保護歸類為公有的資料，同時實作額外的控制以保護敏感資料。
- 自動識別和分類: 自動識別和分類資料，以減少人為作業導致錯誤的風險。
- 定義資料生命週期管理: 您已定義的生命週期策略應以敏感性等級以及法律和組織的要求作為依據。您應考慮保留資料的期間、資料銷毀程序、資料存取管理、資料轉換和資料共享等方面。

SEC 8 您如何保護靜態資料？

實作多個控制來保護您的靜態資料，以降低未經授權的存取或不當處理的風險。

最佳實務:

- 實作安全金鑰管理: 加密金鑰必須安全儲存，並受到嚴格的存取控制；例如使用 AWS KMS 等金鑰管理服務。考慮使用不同的金鑰並對金鑰採取存取控制，同時結合 AWS IAM 和資源政策，以符合資料分類層級和隔離要求。
- 強制執行靜態加密: 根據最新的標準和建議，強制執行您的加密要求，以幫助保護您的靜態資料。
- 自動化靜態資料保護: 使用自動化工具以持續驗證並強制執行靜態資料控制；例如，驗證以確認只有加密的儲存資源存在。
- 強制執行存取控制: 強制執行最低權限存取控制和機制 (包括備份、隔離和版本控制)，以保護靜態資料。防止操作員授予您資料的公有存取權。
- 使用限制人員存取資料的機制: 在正常運作情況下，讓所有使用者遠離直接存取敏感資料和系統的權限。例如，針對執行查詢提供儀表板，而不是直接存取資料存放區。未使用 CI/CD 管道時，請判斷需要哪些控制和程序，才能充分提供一般停用時的緊急存取機制。

SEC 9 您如何保護傳輸中資料？

實作多個控制以保護傳輸中的資料，減少未經授權的存取或遺失的風險。

最佳實務:

- 實作安全金鑰和憑證管理: 安全地儲存加密金鑰和憑證，並透過施行嚴格的存取控制在合宜的時刻進行輪換；例如，使用 AWS Certificate Manager (ACM) 等憑證管理服務。
- 強制執行傳輸中加密: 根據適當的標準和建議強制執行已定義的加密要求，協助您符合組織、法律和合規上的要求。
- 自動偵測意外的資料存取: 您可使用 GuardDuty 這類工具，根據資料分類層級，自動偵測將資料移到所定義邊界以外的嘗試；例如使用 DNS 通訊協定，偵測出將資料複製到未知或不信任網路的木馬程式。
- 驗證網路通訊: 使用支援身份驗證的通訊協定 (Transport Layer Security (TLS) 或 IPsec) 來驗證通訊的身分。

事故回應

SEC 10 您如何預估、回應事件以及從事件中復原？

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

最佳實務:

- 確定關鍵人員和外部資源: 確定可以幫助您的組織回應事件的內部和外部人員、資源及法律義務。
- 制定事件管理計畫: 建立計畫以協助您回應事件、在事件期間進行溝通，以及從事件中復原。例如您可以從工作負載和組織最可能發生的情境，開始建立事件回應計畫。包括您在內部和外部進行溝通和向上呈報的流程。
- 準備鑑識功能: 識別和準備適合的鑑識調查功能，包括外部專家、工具和自動化。
- 自動化遏制能力: 自動化事件遏制與復原，以縮短回應時間和減少對組織的影響。
- 預先佈建存取權限: 確保事件回應者具有預先佈建到 AWS 中的正確存取權限，以縮短調查直至復原的時間。
- 預先部署工具: 確保安全人員具有預先部署到 AWS 中的適當工具，以縮短調查直至復原的時間。
- 執行演練日: 定期練習事件回應演練日 (模擬)，將汲取的教訓納入計畫中，並不斷改進。

可靠性

基礎

REL 1 您如何管理服務配額和限制？

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

最佳實務:

- 了解服務配額和限制: 您需了解工作負載架構的預設配額和配額增加要求。您也需知道哪些資源限制 (例如，磁碟或網路) 具有潛在影響。
- 管理跨帳戶和區域的服務配額: 如果您使用多個 AWS 帳戶或 AWS 區域，確保在生產工作負載執行的所有環境中都要求合適的配額。
- 透過架構適應固定服務配額和限制: 瞭解不可變更的服務配額和實體資源及架構，以防止這些因素影響可靠性。
- 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。
- 自動執行配額管理: 實作工具以在接近閾值時獲得提醒。您可以使用 AWS Service Quotas API，自動化配額增加請求。
- 確保目前配額與最大使用量之間存在足夠差距以適應容錯移轉: 當資源失敗時，在成功終止之前，仍會被計入配額。在終止失敗的資源之前，確保您的配額涵蓋所有失敗的資源與替換資源的重疊部分。計算此差距時，應考慮可用區域失敗。

REL 2 如何規劃您的網路拓撲？

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

最佳實務:

- 針對工作負載公有端點使用高可用性網路連線: 這些端點及其路由必須具備高可用性。為達成此目的，請使用高度可用的 DNS、內容交付網路 (CDN)、API Gateway、負載平衡或反向代理。
- 在雲端中的私有網路與內部部署環境之間佈建冗餘連線能力: 在單獨部署的私有網路之間使用多個 AWS Direct Connect (DX) 連線和多個 VPN 通道。使用多個 DX 位置以實現高可用性。如果使用多個 AWS 區域，請至少在其中兩個區域中確保冗餘。您可能需要評估終止 VPN 的 AWS Marketplace 設備。如果您使用 AWS Marketplace 設備，可在不同的可用區域中部署冗餘執行個體以實現高可用性。
- 確保 IP 子網路分配帳戶具有擴展性和可用性: Amazon VPC IP 地址範圍必須足夠大，以適應工作負載的要求，包括考慮將來擴展 IP 地址以及跨可用區域將 IP 地址分配給子網路。這包括負載平衡器、EC2 執行個體和容器型應用程式。
- 偏好軸幅式拓撲而非多對多網狀拓撲: 如果兩個以上的網路地址空間 (例如，VPC 和內部部署網路) 透過 VPC 對等互連、AWS Direct Connect 或 VPN 連線，則使用軸幅式模型，例如 AWS Transit Gateway 提供的此類模型。
- 在連線的所有私有地址空間中強制使用不重疊的私有 IP 地址範圍: 如果透過 VPN 對等互連或連線，則每個 VPC 的 IP 地址範圍不得重疊。同樣地，您必須避免 VPC 與內部部署環境或您所使用之其他雲端供應商之間出現 IP 地址衝突。您也須有一種在需要時分配私有 IP 地址範圍的方法。

工作負載架構

REL 3 如何設計您的工作負載服務架構？

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

最佳實務:

- 選擇如何劃分工作負載: 應避免整合型架構。反之，您應在 SOA 與微型服務之間做出選擇。做出各種選擇時，請在效益與複雜性之間取得平衡，即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。使用較小的區段的優勢包括更高的靈活性、組織彈性及擴展性。複雜情況包括延遲可能增加、偵錯更複雜，以及運作負擔增加
- 建置專注於特定業務領域和功能的服務: SOA 會建置具有依業務需求定義之明確描述功能的服務。微型服務運用領域模型和有界限的環境來對此項業務進一步限縮，因此各服務僅做一件事。專注於特定功能讓您能夠區別不同服務的可靠性要求，並更集中瞄準投資目標。簡要的業務問題和與各服務相關的小型團隊，也更容易讓組織擴展。
- 每個 API 都提供服務合約: 服務合約為服務整合團隊間的明訂記載的協議，並包括電腦可讀取的 API 定義、速率限制和效能期望。版本控制策略可讓用戶端繼續使用現有 API，並在準備好時將應用程式遷移至更新的 API。只要不違反合約，隨時都可進行部署。服務供應商團隊可以使用自己選擇的技術堆疊，以滿足 API 合約要求。同樣地，服務取用者可以使用自有的技術。

REL 4 如何在分散式系統中設計防止失敗的互動？

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

最佳實務:

- 確定需要哪種分散式系統: 硬式即時分散式系統需要同步、快速給予回應，而軟式即時系統則可以在更長的時段 (分鐘) 內來回應。離線系統會透過批次或非同步處理來處理回應。硬式即時分散式系統具有最嚴格的可靠性要求。
- 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，進而提高彈性和靈活性
- 將所有回應設為等冪: 等冪服務承諾每個請求只完成一次，使得發出多個相同請求與發出單一請求具有相同的效果。等冪服務可讓用戶端更輕鬆地實作重試，而不用擔心錯誤地多次處理請求。為此，用戶端可以使用等冪權杖發出 API 請求，即每次重複請求時，都會使用相同的權杖。等冪服務 API 會使用權杖來傳回與第一次完成請求時傳回之回應相同的回應。
- 持續執行工作: 負載大幅快速變更時，系統可能會發生故障。例如，監控數千部伺服器運作狀態的運作狀態檢查系統，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。

REL 5 如何設計分散式系統中的互動以緩解或承受故障？

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

最佳實務:

- 實作適度降級以將適用的硬相依性轉換為軟相依性: 當元件的相依性狀況不良，元件本身仍可運作，但以降級的方式運作。例如，當相依性呼叫失敗時，容錯移轉為預先決定的靜態回應。
- 調節請求: 這是一種緩解模式，用於回應意外增加的需求。有些請求會接受，但超過定義限制的請求會遭到拒絕，並傳回訊息，指出它們已受到調節。預期用戶端會退避並放棄請求，或以較慢的速率再試一次。
- 控制和限制重試呼叫: 使用指數退避以在逐漸延長間隔後重試。引進抖動來隨機化這些重試間隔，並限制重試次數上限。
- 快速失敗和限制佇列: 如果工作負載無法成功回應請求，則快速失敗。如此將可釋放與請求關聯的資源，並且使服務在資源用盡時復原。如果工作負載能成功回應，但請求率太高，則改為使用佇列來緩衝請求。不過，請勿允許可能導致處理用戶端已放棄的過時請求之長佇列。
- 設定用戶端逾時: 適當設定逾時、系統性對其進行驗證，並且不要依賴預設值，因為它們通常設定得太高
- 盡可能讓服務無狀態: 服務不應要求狀態，或應該卸載狀態，以便在不同的用戶端請求之間，不依賴磁碟或記憶體中本機儲存的資料。這讓伺服器能夠任意置換，而不會對可用性造成影響。Amazon ElastiCache 或 Amazon DynamoDB 是卸載狀態的適當目的地。
- 實作緊急控制桿: 這是可緩解對工作負載的可用性影響的快速程序。它們可以在沒有根本原因的情況下操作。理想的緊急控制桿會提供完全決定性啟用和停用準則，將解析器的認知負擔降至零。範例控制桿包括封鎖所有機器人流量或提供靜態回應。控制桿通常是手動的，但也可以自動化。

變更管理

REL 6 如何監控工作負載資源？

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

最佳實務:

- 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 Personal Health Dashboard 監控 AWS 服務
- 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
- 傳送通知 (即時處理和警示): 當重大事件發生時，需要知道的組織會收到通知
- 自動化回應 (即時處理和警示): 當偵測到事件時，使用自動化來採取措施，例如更換故障元件
- 儲存與分析: 收集日誌檔和指標歷史記錄，並分析這些檔案和歷史記錄，以瞭解更廣泛的趨勢和工作負載洞見
- 定期進行審查: 經常審查工作負載監控的實作方式，並根據重大事件和變更進行更新
- 透過您的系統監控請求的端對端追蹤: 使用 AWS X-Ray 或第三方工具，讓開發人員能夠更輕鬆地分析和偵錯分散式系統，以瞭解其應用程式及其基礎服務的執行成效。

REL 7 如何設計工作負載以適應需求變更？

可擴展工作負載提供自動新增或移除資源的彈性，以便隨時盡可能符合目前需求。

最佳實務:

- 取得或擴展資源時使用自動化: 替換受損的資源或擴展工作負載時，請使用 Amazon S3 和 AWS Auto Scaling 等受管的 AWS 服務進行自動化程序。您還可以使用第三方工具和 AWS 開發套件來自動調整規模。
- 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。
- 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。
- 對工作負載執行負載測試: 採用負載測試方法來衡量擴展活動是否滿足工作負載要求。

REL 8 您如何實作變更？

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

最佳實務:

- 將執行手冊用於部署等標準活動: 執行手冊是實現特定成果的預定義步驟。使用執行手冊執行手動或自動進行的標準活動。範例包括部署工作負載、修補工作負載或進行 DNS 修改。
- 將功能測試整合為部署的一部分: 功能測試會作為自動化部署的一部分執行。如果未符合成功條件，則會終止或回復管道。
- 將彈性測試整合為部署的一部分: 彈性測試 (做為混沌工程的一部分) 會在生產前環境中做為自動化部署管道的一部分執行。
- 使用不可變基礎設施進行部署: 此模型會強制規定生產工作負載上不會就地進行更新、安全性修補程式或組態方面的變更。需要進行變更時，會在新的基礎設施上建置架構並部署到生產環境。
- 使用自動化部署變更: 部署和修補經過自動化以消除負面影響。

失敗管理

REL 9 您如何備份資料？

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

最佳實務:

- 識別並備份所有需要備份的資料，或從來源複製資料: Amazon S3 可做為多個資料來源的備份目的地。Amazon EBS、Amazon RDS 和 Amazon DynamoDB 等 AWS 服務已內建用於建立備份的功能。也可以使用第三方備份軟體。或者，如果可以從其他來源複製資料以滿足 RPO，則可能不需要備份
- 保護和加密備份: 透過 AWS IAM 等身份驗證和授權來偵測存取，並透過使用加密來偵測資料完整性受損情況。
- 自動執行資料備份: 設定備份以根據定期排程或資料集中的變更自動執行。RDS 執行個體、EBS 磁碟區、DynamoDB 表和 S3 物件都可以設定為自動備份。您也可以使用 AWS Marketplace 解決方案或第三方解決方案。
- 定期執行資料復原以驗證備份的完整性和程序: 透過執行復原測試，驗證您的備份程序實作是否符合復原時間目標 (RTO) 和復原點目標 (RPO)。

REL 10 如何使用故障隔離來保護您的工作負載？

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

最佳實務:

- 部署工作負載至多個位置: 跨多個可用區域或視需要跨 AWS 區域，分配工作負載資料和資源。這些位置可以根據需要多樣化。
- 針對限制在單一位置的元件將復原自動化: 如果工作負載的元件只能在單一可用區域或內部部署資料中心執行，您必須在定義的復原目標內實作完整重建工作負載的功能。
- 使用隔板架構: 如同船舶上的隔板一樣，此模式可確保失敗限制在一小部分的請求/使用者中，如此一來才能限制受損的請求數量，讓大部分的請求可以繼續運行，而不會發生錯誤。資料的隔板通常稱為分割區或分區，而服務的隔板稱為儲存格。

REL 11 如何設計工作負載以承受元件失敗？

需要高可用性和低平均復原時間 (MTTR) 的工作負載必須建立彈性架構。

最佳實務:

- 監控工作負載的所有元件以偵測失敗: 持續監控工作負載的運作狀態，讓您和自動化系統在發生效能降低或完全失敗時能夠察覺。根據商業價值監控關鍵績效指標 (KPI)。
- 容錯移轉至未影響位置中運作良好的資源: 確保在某個位置發生失敗時，運作良好位置中的資料和資源可以繼續處理請求。這對多區域工作負載而言更輕鬆，因為 Elastic Load Balancing 和 AWS Auto Scaling 等 AWS 服務可協助跨可用區域分配負載。對於多區域工作負載，這會更複雜。例如，跨區域僅供讀取複本讓您可以將資料部署至多個 AWS 區域，但您仍須將僅供讀取複本升階為主節點，並在主要位置發生失敗時將流量指向該主節點。Amazon Route 53 和 AWS Global Accelerator 也可協助跨 AWS 區域路由流量。
- 將所有分層的修復自動化: 偵測到失敗時，使用自動化功能執行動作來進行修復。
- 使用靜態穩定性來防止雙模態行為: 雙模態行為是您的工作負載在正常和失敗模式下展現出不同的行為，例如，當可用區域失敗時，仰賴啟動新的執行個體。您應改為建置靜態穩定且僅以一種模式操作的工作負載。在這種情況下，如果移除一個可用區域，則在每個可用區域佈建足夠的執行個體來處理工作負載負載，然後使用 Elastic Load Balancing 或 Amazon Route 53 運作狀態檢查，將負載從受損的執行個體移出。
- 當事件影響可用性時傳送通知: 當偵測到重大事件時傳送通知，即使事件造成的問題已自動解決。

REL 12 如何測試可靠性？

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

最佳實務:

- 使用程序手冊調查失敗: 透過在程序手冊中記錄調查程序，實現對無法充分理解的失敗情境進行快速一致的回應。程序手冊是為識別造成失敗情境的因素所執行的預先定義步驟。在確定或向上呈報問題之前，任何程序步驟的結果都用於確定要採取的後續步驟。
- 執行事件後分析: 審查影響客戶的事件，並識別成因和預防性行動項目。使用此資訊來開發緩解措施，以限制或防止事件再次發生。制定可快速有效回應的程序。適當地傳達成因和為目標受眾量身打造的糾正措施。建立一種可以根據需要將這些原因傳達給其他人的方法。
- 測試功能要求: 這包括驗證所需功能的單位測試和整合測試。
- 測試擴展和效能需求: 這包括進行負載測試，以驗證工作負載是否滿足擴展和效能需求。
- 使用混沌工程測試彈性: 執行定期將失敗注入生產前和生產環境中的測試。假設工作負載對失敗的反應，然後將您的假設與測試結果進行比較，並在不相符時反覆進行測試。請確保生產測試不會影響使用者。
- 定期進行演練日: 使用演練日定期執行失敗程序，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，以確保生產測試不會影響使用者。

REL 13 您如何規劃災難復原 (DR)？

備妥備份和冗餘工作負載元件是 DR 策略的開始。RTO 和 RPO 是您還原可用性的目標。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。

最佳實務:

- 定義停機和資料遺失的復原目標: 工作負載具有復原時間目標 (RTO) 和復原點目標 (RPO)。
- 使用定義的復原策略來滿足復原目標: 已經定義了災難復原 (DR) 策略來實現目標。
- 測試災難復原實作以驗證實作: 定期測試容錯移轉到災難復原，以確保滿足 RTO 和 RPO。
- 管理 DR 站點或區域的組態偏移: 確保根據需要在 DR 站點或區域提供基礎設施、資料和組態。例如，檢查 AMI 和服務配額是否為最新版本。
- 自動化復原: 使用 AWS 或第三方工具自動化系統復原，並將流量路由到 DR 站點或區域。

效能達成效率

選擇

PERF 1 您如何選擇效能最佳的架構？

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

最佳實務:

- 了解可用的服務和資源: 了解並熟悉雲端中可用的廣泛服務和資源。確定與工作負載相關的服務和組態選項，並了解如何獲得最佳效能。
- 定義架構選擇程序: 使用內部經驗和雲端知識，或使用外部資源 (例如已發佈的使用案例、相關文件或白皮書)，定義選擇資源和服務的程序。您定義的程序應該鼓勵對可在工作負載中使用的服務進行實驗和基準化分析。
- 將成本需求因素納入決策 : 工作負載通常具有營運的成本需求。使用內部成本控制，根據預測的資源需求選取資源類型和大小。
- 使用政策或參考架構: 透過評估內部政策和現有參考架構，並使用分析來選取工作負載的服務和組態，來將效能和效率提升至最大。
- 使用雲端供應商或適當的合作夥伴提供的指導: 使用解決方案架構師、專業服務或適當的合作夥伴等雲端公司資源，來引導您做出決策。這些資源可協助檢閱和改善架構，以實現最佳效能。
- 對現有工作負載進行基準化分析: 對現有工作負載的效能進行基準化分析，以了解工作負載在雲端的效能。使用從基準化分析中收集的資料，來推動架構決策。
- 對工作負載執行負載測試: 使用不同類型和大小的資源，在雲端部署最新的工作負載架構。監控部署，以擷取可識別瓶頸或過多容量的效能指標。使用此效能資訊，來設計或改善您的架構和資源選擇。

PERF 2 您如何選擇運算解決方案？

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

最佳實務:

- 評估可用的運算選項: 了解您可以使用的運算選項的效能特性。了解執行個體、容器和函數運作的方式，以及它們會給您的工作負載帶來什麼優勢或劣勢。
- 了解可用的運算組態選項: 了解各種選項如何與您的工作負載互補，以及哪種組態選項最適合您的系統。這些選項的範例包括：執行個體系列、大小、功能 (GPU、I/O)、函數大小、容器執行個體、單租用與多租用。
- 收集與運算相關的指標: 了解運算系統效能的最好方法之一是記錄和追蹤各種資源的真實使用情況。此資料可用來更準確地判斷資源需求。
- 透過適當調整大小來確定所需的組態: 分析工作負載的各種效能特性，以及這些特性與記憶體、網路和 CPU 使用量的關係。使用此資料，可以選擇最適合您工作負載描述檔的資源。例如，執行個體的 R 系列可以為記憶體密集型工作負載 (例如資料庫) 提供最佳服務。不過，高載工作負載從彈性容器系統中獲益的程度更高。
- 利用資源的可用彈性: 雲端提供的彈性可透過各種機制來動態擴展或減少資源，以滿足需求的變化。結合與運算相關的指標，工作負載可以自動回應變更，並利用最佳資源集來實現其目標。
- 根據指標重新評估運算需求: 使用系統層級指標來確定工作負載隨時間的行為和要求。透過將可用資源與這些需求進行比較來評估您的工作負載需求，並對運算環境進行變更，以達到最適合工作負載描述檔的狀態。例如，隨著時間的流逝，系統可能會比最初想像的要消耗更多的記憶體，因此轉換到不同的執行個體系列或大小，可以同時提高效能和效率。

PERF 3 您如何選擇儲存解決方案？

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

最佳實務:

- 了解儲存特性和要求: 了解選擇最適合工作負載的服務 (例如物件儲存、區塊儲存、檔案儲存或執行個體儲存體) 所需的不同特性 (例如可共享、檔案大小、快取大小、存取模式、延遲、輸送量和資料持久性)。
- 評估可用的組態選項: 評估各種特性和組態選項，以及它們與儲存的關係。了解如何在何處使用佈建 IOPS、SSD、磁帶儲存、物件儲存、存檔儲存或暫時性儲存，以優化工作負載的儲存空間和效能。
- 根據存取模式和指標制定決策: 根據工作負載的存取模式選擇儲存系統，並透過決定工作負載存取資料的方式來設定儲存系統。選擇物件儲存而不是區塊儲存，以提高儲存效率。設定您選擇的儲存選項以匹配資料存取模式。

PERF 4 您如何選擇資料庫解決方案？

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

最佳實務:

- 了解資料特性: 了解工作負載中資料的不同特性。確定工作負載是否需要交易、如何與資料互動、其效能需求為何。使用此資料為您的工作負載選擇最佳效能資料庫方法 (例如，關聯式資料庫、NoSQL 鍵值、文件、寬欄、圖形、時間序列或記憶體內儲存)。
- 評估可用選項: 評估在工作負載的儲存機制選擇過程中可用的服務和儲存選項。了解使用給定的服務或系統來存放資料的方法及時間。了解可優化資料庫效能或效率的可用組態選項，例如佈建 IOPS、記憶體和運算資源，以及快取。
- 收集並記錄資料庫效能指標: 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。例如，測量存取資料庫時每秒交易、慢查詢或引入的系統延遲。使用這些資料以了解資料庫系統的效能。
- 根據存取模式選擇資料儲存: 根據工作負載的存取模式確定要使用的服務和技術。例如，利用關聯式資料庫處理需要交易的工作負載，或者提供更高輸送量的鍵值存儲 (但最終在適用時保持一致)。
- 根據存取模式和指標優化資料儲存: 使用效能特性和存取模式來優化資料儲存或查詢，以實現最佳效能。測量索引編制、鍵值分佈、資料倉儲設計或快取策略此類的優化，會對系統效能或整體效率造成何種影響。

PERF 5 您如何設定聯網解決方案？

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

最佳實務:

- 了解聯網如何影響效能: 分析並了解與網路相關的決策如何影響工作負載效能。例如，網路延遲通常會影響使用者體驗，並且使用錯誤的協定可能會因過多的開銷而使網路容量不足。
- 評估可用的聯網功能: 評估雲端中可能提升效能的聯網功能。透過測試、指標和分析來測量這些功能的影響。例如，利用可用的網路層級功能來降低延遲、網路距離或抖動。
- 為混合式工作負載選擇適當大小的專用連線或 VPN: 當需要內部部署通訊時，請確定您有足夠的頻寬可容納工作負載效能。根據頻寬需求，單一專用連線或單一 VPN 可能不足，而且您必須啟用跨多個連線的流量負載平衡。
- 利用負載平衡和加密卸載: 在多個資源或服務之間分配流量，以讓您的工作負載能夠利用雲端提供的彈性。您也可以使用負載平衡來卸載加密終止，以提升效能及有效管理和路由流量。
- 選擇可提高效能的網路通訊協定: 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。
- 根據網路要求選擇工作負載的位置: 使用可用的雲端位置選項來降低網路延遲或提高輸送量。利用 AWS 區域、可用區域、置放群組和節點 (例如 Outposts、Local Regions 和 Wavelength) 來降低網路延遲或改善輸送量。
- 根據指標優化網路組態: 使用收集和分析的資料來做出有關優化網路組態的明智決策。測量這些變更的影響，並利用這些測量結果來做出未來的決策。

審查

PERF 6 您如何發展工作負載，以運用新版本的優勢？

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

最佳實務:

- 掌握最新的資源和服務: 當新服務、設計模式和產品供應項目推出時，評估提升效能的方法。透過臨時評估、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。
- 定義提高工作負載效能的程序: 定義一個程序，以在新的服務、設計模式、資源類型和組態可用時對其進行評估。例如，對新的執行個體方案執行現有的效能測試，以判斷其是否可能改善工作負載。
- 隨時間提升工作負載效能: 作為一個組織，使用評估過程中收集的資訊，在新服務或資源可用時主動推動採用。

監控

PERF 7 您如何監控資源來確保達成預期效能？

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

最佳實務:

- 記錄效能相關指標: 使用監控和可觀察性服務來記錄效能相關指標。例如，記錄資料庫交易、慢速查詢、I/O 延遲、HTTP 請求輸送量、服務延遲或其他關鍵資料。
- 分析事件或事故發生時的指標: 為回應事件或事故 (或在事件或事故期間)，使用監控儀表板或報告來了解和診斷影響。這些檢視可讓您深入了解工作負載的哪些部分未如預期執行。
- 建立用於測量工作負載效能的關鍵績效指標 (KPI): 確定指示工作負載效能是否達到預期的 KPI。例如，以 API 為基礎的工作負載可能使用整體回應延遲來表示整體效能，而電子商務網站可能會選擇將購買數用作其 KPI。
- 使用監控來產生警示型通知: 使用監控系統和您定義的效能相關關鍵績效指標 (KPI)，當這些測量結果超出預期範圍時自動產生警示。
- 定期審查指標: 作為日常維護或對事件或事故的回應，審查收集了哪些指標。透過這些審查來確定哪些指標是解決問題的關鍵，以及哪些其他指標 (如果被追蹤) 將有助於識別、解決或預防問題。
- 主動監控和警示: 使用關鍵績效指標 (KPI) 搭配監控和提醒系統，主動處理效能相關的問題。使用警示觸發自動化動作，盡可能修復問題。如果無法自動回應，則將警示上報給能夠回應的人員。例如，您可能有一個可以預測關鍵績效指標 (KPI) 預期值並在超過特定閾值時發出警示的系統，或者在 KPI 超出預期值時可以自動停止或回復部署的工具。

權衡

PERF 8 您如何採用權衡來增進效能？

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

最佳實務:

- 了解效能至關重要的領域: 了解並確定提高工作負載效能將對效率或客戶體驗產生積極影響的領域。例如，具有大量客戶互動的網站可受益於邊緣服務，因為這樣可以將內容交付移至更接近客戶的地方。
- 了解設計模式和服務: 研究並了解有助於提高工作負載效能的各種設計模式和服務。作為分析的一部分，確定您為了實現更高效能而可能付出的代價。例如，使用快取服務可以幫助減少資料庫系統上的負載。但是，它需要一些工程來實作安全的快取，或者在某些區域可能需要引入最終一致性。
- 確定權衡如何影響客戶和效率: 在評估與效能相關的改進時，判斷哪些選擇將如何影響客戶和工作負載效率。例如，如果使用鍵值資料存放區提高系統效能，請務必評估其最終一致性本質對客戶的影響。
- 衡量效能改進的影響: 在進行變更以提高效能時，請評估所收集的指標和資料。使用此資訊來判斷效能提升對工作負載、工作負載元件和客戶所造成的影響。此測量可協助您了解權衡所帶來的改善，並協助您判斷是否產生任何負面影響。
- 使用各種與效能相關的策略: 在適用的情況下，可利用多種策略來提升效能。這些策略包括：快取資料以防止過多的網路或資料庫呼叫、使用資料庫引擎的唯讀複本來提高讀取速率、在可能的情況下對資料進行分區或壓縮以減少資料量，以及對結果進行緩衝和串流以避免阻塞。

成本優化

實作雲端財務管理

COST 1 如何實作雲端財務管理？

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

最佳實務:

- 建立成本優化職能部門: 建立一支團隊，負責建立並維護整個組織的成本感知。該團隊需要在組織中擔任財務、技術和業務角色的人員。
- 在財務與技術之間建立合作夥伴關係: 讓財務和技術團隊參與討論雲端之旅各個階段的成本和用量。各團隊定期碰面並討論相關主題，例如，組織總目標和具體目標、成本和用量的目前狀態，以及財務和會計實務。
- 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。
- 在組織程序中實作成本感知: 在會影響用量的全新或現有程序中實作成本感知，並利用現有程序取得成本感知。在員工培訓中實作成本感知。
- 就成本優化提供報告和通知: 設定 AWS 預算以針對目標提供有關成本和用量的通知。定期召開會議以分析此工作負載的成本效率並推廣成本感知文化。
- 主動監控成本: 實作工具和儀表板來主動監控工作負載的成本。當您收到通知時，不要只看成本和類別。這有助於識別正面趨勢，並在整個組織中推廣它們。
- 及時了解新的服務版本: 定期諮詢專家或 APN 合作夥伴，以了解哪些服務和功能可以降低成本。審查 AWS 部落格和其他資訊來源。

支出和用量感知

COST 2 您如何管控用量？

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

最佳實務:

- 根據您的組織要求制定政策: 制定定義組織如何管理資源的政策。政策應涵蓋資源和工作負載的成本方面，包括在資源生命週期中資源的建立、修改和除役。
- 實作總目標和具體目標: 為您的工作負載實作成本和用量目標。總目標可為您的組織提供成本和用量的方向，而具體目標可為您的工作負載提供可測量的結果。
- 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。
- 實作群組和角色: 實作符合您政策的群組和角色，並控制哪些人員可以建立、修改或除役每個群組中的執行個體和資源。例如，實作開發、測試和生產群組。這適用於 AWS 服務和第三方解決方案。
- 實作成本控制措施: 根據組織政策以及定義的群組和角色實作控制措施。這些控制措施可確保成本的發生始終符合組織要求：例如，使用 IAM 政策控制對區域或資源類型的存取。
- 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。

COST 3 您如何監控用量和成本？

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

最佳實務:

- 設定詳細資訊來源: 設定 AWS 成本和用量報告，以及 Cost Explorer 每小時精細度，以提供詳細的成本和用量資訊。設定您的工作負載，使其具有每個交付業務成果的日誌項目。
- 識別成本歸因類別: 識別可用於在組織內分配成本的組織類別。
- 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。
- 設定帳單和成本管理工具: 根據組織政策設定 AWS Cost Explorer 和 AWS 預算。
- 將組織資訊新增至成本與用量: 根據組織、工作負載屬性和成本分配類別來定義標記結構描述。在所有資源上實作標記。使用 Cost Categories 以根據組織屬性將成本與用量分組。
- 根據工作負載指標分配成本: 按指標或業務成果分配工作負載的成本，以衡量工作負載的成本效率。實作程序以使用 Amazon Athena 分析 AWS 成本和用量報告，從而獲得洞見和退款功能。

COST 4 如何進行資源除役？

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

最佳實務:

- 在資源的生命週期中追蹤資源: 定義並實作一種方法，在資源的生命週期中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。
- 實作除役程序: 實作識別和除役孤立資源的程序。
- 除役資源: 除役由諸如定期稽核或用量變更等事件觸發的資源。除役通常會定期執行，而且是手動或自動化的。
- 自動除役資源: 設計工作負載，在識別和除役非關鍵資源、不需要的資源或低利用率資源時，妥善處理資源終止。

具有經濟效益的資源

COST 5 您選擇服務時如何評估成本？

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

最佳實務:

- 確定組織的成本要求: 與團隊成員一起為此工作負載定義成本最佳化與其他支柱 (例如效能和可靠性) 之間的平衡。
- 分析此工作負載的所有元件: 確保分析每個工作負載元件，無論當前大小或當前成本如何。檢閱工作應反映潛在的效益，例如當前和預計的成本。
- 對每個元件執行徹底的分析: 查看每個元件的組織整體成本。透過考慮營運和管理成本來查看整體擁有成本，尤其是在使用受管服務時。檢閱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。
- 選取具成本效益授權的軟體: 開放原始碼軟體將剔除對工作負載增加大量成本的軟體授權成本。請在需要授權軟體時，避免繫結至任意屬性 (例如 CPU) 的授權，尋找繫結至輸出或成果的授權。這些授權的成本會更接近其提供的效益。
- 選取此工作負載的元件，以按照組織優先事項來最佳化成本: 選取所有元件時需考慮成本因素。這包括使用應用程式層級和受管服務，例如 Amazon RDS、Amazon DynamoDB、Amazon SNS 和 Amazon SES，以降低整體組織成本。使用無伺服器和容器執行運算，例如 AWS Lambda、用於靜態網站的 Amazon S3 和 Amazon ECS。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。
- 對一段時間內的不同用量進行成本分析: 工作負載可能隨時間變更。某些服務或功能在不同的用量層級上更具成本效益。按預計用量隨時間對每個元件執行分析，可以確保此工作負載在其整個生命週期內保持成本效益。

COST 6 您選擇資源類型、大小和數量時，如何達成成本目標？

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

最佳實務:

- 執行成本建模: 確定組織要求並對工作負載及其每個元件執行成本建模。在不同預測負載下對工作負載執行基準測試活動，並比較成本。建模工作應反映潛在效益：例如，花費的時間與元件成本成正比。
- 根據資料選擇資源類型和大小: 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來進行此選擇。
- 根據指標自動選擇資源類型和大小: 使用目前執行的工作負載中的指標來選擇正確的大小和類型，以最佳化成本。為服務 (例如 Amazon EC2、Amazon DynamoDB、Amazon EBS (PIOPS)、Amazon RDS、Amazon EMR 和聯網) 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。

COST 7 您如何使用定價模式降低成本？

使用最適合您資源的定價模式，大幅減少支出。

最佳實務:

- 執行定價模式分析: 分析工作負載的每個元件。判斷元件與資源是否會執行較長期間 (針對承諾折扣)，或動態與短期執行 (針對 Spot 或隨需)。使用 AWS Cost Explorer 中的建議功能對工作負載執行分析。
- 根據成本實作區域: 每個區域的資源定價可能不同。考慮區域成本，可以確保您為此工作負載支付最低的總價
- 選擇具成本效益條款的第三方協議: 具成本效益的協議和條款可確保這些服務的成本隨其提供的優勢而擴展。選擇可在為您的組織提供額外優勢時擴展的協議和定價。
- 針對此工作負載的所有元件實作定價模式: 永久執行的資源應使用預留容量，例如 Savings Plans 或預留執行個體。設定短期容量以使用 Spot 執行個體或 Spot 叢集。隨需執行個體僅用於無法中斷且執行時間不夠長，不適合使用預留容量的短期工作負載：這段時間介於 25% 到 75% 之間 (視資源類型而定)。
- 在主要帳戶層級執行定價模式分析: 使用 Cost Explorer Savings Plans 和預留執行個體建議，在主要帳戶層級針對承諾折扣執行定期分析。

COST 8 您如何規劃資料傳輸費？

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

最佳實務:

- 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。
- 選擇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 最佳化和異地同步備份組態等元件
- 實作可降低資料傳輸成本的服務: 實作可降低資料傳輸成本的服務：例如，使用 Amazon CloudFront 之類的 CDN 向最終使用者交付內容，使用 Amazon ElastiCache 快取層，或者使用 AWS Direct Connect 代替 VPN 連線到 AWS。

管理需求與供應資源

COST 9 如何管理需求和供應資源？

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

最佳實務:

- 對工作負載需求進行分析: 分析工作負載隨時間的需求。確保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。
- 實作緩衝或調節機制來管理需求: 緩衝和調節機制會修改工作負載的需求，以消除任何尖峰時段。在用戶端執行重試時實作調節機制。實作緩衝機制以儲存請求，並將處理的時間往後延遲。確保調節和緩衝區經過設計，以便讓用戶端在所需時間內收到回應。
- 動態提供資源: 資源是按計畫進行佈建。這可以是以需求為基礎 (例如，透過自動調整規模)，或是以時間為基礎，其中需求可預測，並且根據時間提供資源。這些方法可儘量減少過度佈建或佈建不足的數量。

隨時間優化

COST 10 您如何評估新服務？

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

最佳實務:

- 制定工作負載審查程序: 制定一個程序，用於定義工作負載審查的標準和程序。審查工作應反映潛在效益；例如，核心工作負載或價值超過賬單 10% 的工作負載每季度進行審查，而低於 10% 的工作負載則每年進行審查。
- 定期審查和分析此工作負載: 根據定義的程序定期審查現有的工作負載。