

AWS Well-Architected Framework

卓越運作支柱



卓越運作支柱: AWS Well-Architected Framework

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

摘要和介紹	1
簡介	1
卓越營運	2
設計原則	2
定義	3
組織	4
組織優先事項	6
OPS01-BP01 評估外部客戶需求	6
OPS01-BP02 評估內部客戶需求	7
OPS01-BP03 評估治理要求	8
OPS01-BP04 評估合規要求	10
OPS01-BP05 評估威脅態勢	13
OPS01-BP06 評估權衡的同時管理利益和風險	15
操作模型	17
操作模式 2 x 2 表示法	18
關係和擁有權	25
組織文化	41
OPS03-BP01 提供高層支持	41
OPS03-BP02 授權團隊成員在成果有風險時採取動作	44
鼓勵 OPS03-BP03 升級	46
OPS03-BP04 溝通需及時、清楚且可行	48
OPS03-BP05 鼓勵進行試驗	52
OPS03-BP06 管理團隊成員維持和發展自己的技能集	54
OPS03-BP07 資源團隊適當	57
準備	60
實作可觀測性	61
OPS04-BP01 識別關鍵績效指標	61
OPS04-BP02 實作應用程式遙測	63
OPS04-BP03 實作使用者體驗遙測	66
OPS04-BP04 實作相依性遙測	68
OPS04-BP05 實作分散式追蹤	71
營運設計	73
OPS05-BP01 使用版本控制	73
OPS05-BP02 測試並驗證變更	75

OPS05-BP03 使用組態管理系統	78
OPS05-BP04 使用建置和部署管理系統	80
OPS05-BP05 執行修補程式管理	82
OPS05-BP06 共用設計標準	85
OPS05-BP07 實作用於提高程式碼品質的實務	87
OPS05-BP08 使用多個環境	89
OPS05-BP09 進行頻繁、小型、可逆的變更	91
OPS05-BP10 完全自動化整合和部署	92
緩解部署風險	93
OPS06-BP01 計畫變更失敗	94
OPS06-BP02 測試部署	96
OPS06-BP03 採用安全的部署策略	98
OPS06-BP04 自動化測試和復原	101
營運準備度和變更管理	103
OPS07-BP01 確保人員能力	104
OPS07-BP02：確保對營運準備度進行一致的審查	106
OPS07-BP03 使用 Runbook 執执行程序	109
OPS07-BP04 使用程序手冊來調查問題	112
OPS07-BP05 做出明智的決策以部署系統和變更	116
OPS07-BP06 建立生產工作負載的支援計劃	117
營運	120
利用工作負載可觀測性	121
OPS08-BP01 分析工作負載指標	121
OPS08-BP02 分析工作負載日誌	123
OPS08-BP03 分析工作負載追蹤	125
OPS08-BP04 建立可執行的提醒	128
OPS08-BP05 建立儀表板	130
了解運作狀態	133
OPS09-BP01 使用指標衡量營運目標與 KPI	133
OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況	135
OPS09-BP03 檢閱操作指標並排定改善優先順序	137
回應事件	138
OPS10-BP01 使用程序進行事件、事故和問題管理	139
OPS10-BP02 每個提醒建立一個程序	143
OPS10-BP03 根據業務影響排定操作事件的優先順序	147
OPS10-BP04 定義升級路徑	149

OPS10-BP05 為影響服務的事件定義客戶溝通計劃	151
OPS10-BP06 透過儀表板傳達狀態	154
OPS10-BP07 自動化對事件的回應	156
演進	159
學習、分享和改進	159
OPS11-BP01 建立持續改進程序	160
OPS11-BP02 執行事後分析	162
OPS11-BP03 實作意見回饋循環	163
OPS11-BP04 執行知識管理	166
OPS11-BP05 定義改進驅動因素	168
OPS11-BP06 驗證洞察	170
OPS11-BP07 執行操作指標檢閱	172
OPS11-BP08 記錄和分享獲得的經驗	173
OPS11-BP09 分配時間進行改善	175
結論	177
貢獻者	178
深入閱讀	179
文件修訂	180
注意	182
AWS 詞彙表	183

卓越營運支柱 - AWS Well-Architected Framework

發布日期：2024 年 11 月 6 日 ([文件修訂](#))

本白皮書的重點是 AWS Well-Architected Framework 的卓越營運支柱。當中提供了相關指引，可協助您在設計、交付和維護 AWS 工作負載時套用最佳實務。

簡介

[AWS Well-Architected Framework](#) 可協助您了解在 AWS 上建置工作負載時所做決策的優勢和風險。透過使用該架構，您將了解關於在雲端設計和操作可靠、安全、有效率、經濟實惠且永續的工作負載的營運和架構最佳實務。藉助該架構，即可根據最佳實務一致地量測營運和架構，並找出需要改進的方面。我們相信，擁有重視營運的 Well-Architected 工作負載可大幅提高企業成功的可能性。

此架構以六大支柱為基礎：

- 操作效能
- 安全
- 可靠性
- 效能效率
- 成本最佳化
- 永續性

本白皮書重點介紹了卓越營運支柱，以及如何將其套用為您架構良好的解決方案的基礎。營運被認為是與其支援的業務線和開發團隊截然不同的獨立職能，因此要在此環境中實現卓越營運極具挑戰性。透過採用本白皮書中的實務，您可以建置能夠洞見營運狀態的架構、實現有效的營運和事件回應，並且可以繼續改善和支援您的業務目標。

本白皮書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。閱讀本白皮書之後，您將了解在設計卓越營運雲端架構時要使用的 AWS 最佳實務和策略。本白皮書並未提供實作的詳細資訊或架構模式。不過，其確實包含有關此資訊的適當資源的參考資料。

卓越營運

卓越營運 (OE) 是一項承諾，代表致力於妥善設計軟體，同時持續提供絕佳的客戶體驗。卓越營運支柱包括組織團隊、設計工作負載、大規模運作工作負載及促使其進化的最佳實務。

卓越營運的目標是要快速且可靠地為客戶提供新功能和錯誤修正。持續投入卓越營運的組織不僅可滿足其客戶，同時也能打造新功能、勇於改變，並且積極面對失敗。在此過程中，卓越營運可協助開發人員持續達成高品質的成果，進而實現持續整合與持續交付 (CI/CD)。

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。領導力應該充分投入，並致力於使用合適的雲端作業模式進行 CloudOps 轉型，以鼓勵團隊以最有效的方式營運並達成業務成果。正確的營運模式會利用人員、流程和技术能力來擴展、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害關係人和消費者。目標和營運 KPI 在各個層面都一致。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 (KPI)，並利用可觀測性遙測來做出明智的決策，並在業務成果有風險時迅速採取行動。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。向所有利害關係人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：盡可能地使用 AWS 受管服務以降低營運負擔。圍繞與這些服務的互動建置營運程序。

定義

雲端有四種最佳實務領域可實現卓越營運：

- 組織
- 準備
- 營運
- 演進

組織的領導階層定義業務目標。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的方式，實作啟用整合、部署及交付工作負載的服務將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，將協助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。

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須重構。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 以協助您在開發前審核您的方法、生產前的工作負載狀態以及生產中工作負載的狀態。您可以將工作負載與最新的 AWS 架構最佳實務做比較、監控工作負載的整體狀態，以及深入了解潛在風險。AWS Trusted Advisor 是一款可存取核心檢查集的工具，這些檢查提出了優化建議，可能有助您確定優先事項。商業和企業支援客戶可存取針對安全性、可靠性、效能、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。使用 AWS 支援 (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。請透過 AWS 支援 Center 聯絡 AWS 支援，尋求 AWS 相關問題的協助。AWS 還分享了我們透過在 Amazon 建置者資料中心的 AWS 操作中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 基礎原理中的自主進度數位課程提供一些培訓。您還可以報名參加講師指導下的培訓，以進一步協助開發團隊的 AWS 技能。

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，允許您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。

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。

主題

- [組織優先事項](#)

- [操作模型](#)
- [組織文化](#)

組織優先事項

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。定期審查您的優先事項，以便在組織的需求變更時更新優先事項。

最佳實務

- [OPS01-BP01 評估外部客戶需求](#)
- [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 評估權衡的同時管理利益和風險](#)

OPS01-BP01 評估外部客戶需求

讓關鍵利害關係人 (包括業務、開發和營運團隊) 參與進來，以確定將工作重點放在哪些外部客戶需求上。這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。
- 您擁有捕捉外部客戶需求的機制。

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害關係人擁有共同的目標並達成共識，方能讓業務取得成功。

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

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

資源

相關的最佳實務：

- [OPS11-BP03 實作回饋迴圈](#)

OPS01-BP02 評估內部客戶需求

讓關鍵利益相關者 (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。合規標準的例子包括 PCI DSS、FedRAMP 和 HIPAA。適用的合規標準取決於各種因素，例如解決方案存放或傳輸的資料類型，以及解決方案支援的地理區域。

預期成果：

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

常見的反模式：

- 您的工作負載有部分屬於支付卡產業資料安全標準 (PCI-DSS) 架構下，但您的工作負載儲存信用卡資料時並未予以加密。
- 您的軟體開發人員和架構師不知道您的組織必須遵循的合規架構。
- 年度 Systems and Organizations Control (SOC2) Type II 稽核即將到來，但您無法驗證已設置控制。

建立此最佳實務的優勢：

- 評估和了解套用到工作負載的合規要求，可讓您了解如何安排工作的優先順序來實現商業價值。
- 您可以選擇與合規架構相符的適當位置和技术。
- 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 Retail 儲存客戶的信用卡資訊。卡片儲存團隊的開發人員了解他們必須遵從 PCI-DSS 架構。他們執行了相關步驟，驗證信用卡資訊以安全方式儲存和存取，並遵從 PCI-DSS 架構。他們每年都會與安全團隊共同驗證合規性。

實作步驟

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

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

資源

相關的最佳實務：

- [SEC01-BP03 識別並驗證控制目標](#) - 安全控制目標是整體合規性的重要組成部分。
- [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](#) - 許多合規性框架都基於資料處理和儲存政策。
- [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](#)
- [合規計劃範圍內的 AWS 服務](#)

相關影片：

- [AWS re:Invent 2020：使用 AWS Compute Optimizer 實現合規即程式碼](#)

- [AWS re:Invent 2021 - 雲端合規性、保證和稽核](#)
- [AWS Summit ATL 2022 - 在 AWS 上實作合規性、保證和稽核 \(COP202\)](#)

相關範例：

- [AWS 上的 PCI DSS 和 AWS 基礎安全最佳實務](#)

相關服務：

- [AWS Artifact](#)
- [AWS Audit Manager](#)
- [AWS Compute Optimizer](#)
- [AWS Security Hub](#)

OPS01-BP05 評估威脅態勢

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

[Well-Architected Framework](#) 強調學習、衡量和改善。它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 評估權衡的同時管理利益和風險

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，您可能會被要求加快新功能的上市速度，而不是最佳化 IT 基礎架構成本。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。使用 [支援](#) ([AWS 知識中心](#)、[AWS 論壇](#)和 [支援 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 支援。

AWS 還可以在 [Amazon 建置者資料中心](#)中分享操作最佳實務和模式。您可透過 [AWS 部落格](#)和[官方 AWS 播客](#)獲得其他各種實用資訊。

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支付卡產業資料安全標準 (PCI DSS) 的合規性。您沒有考量滿足要求和繼續您目前開發工作之間的權衡取捨。相反地，您繼續開發工作，而不證明合規性。由於對平台安全性及其投資的擔憂，您的投資者會停止對公司的支援。
- 您已決定包含其中一位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，導致長期計劃收益減少，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。範例包括企劃案影響、上市時間、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。範例包括上市時間、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，強調上市速度優先於可靠性，可能提供競爭優勢。不過，如果發生可靠性問題，則可能會縮短正常執行時間。

4. 以程式設計方式強制執行關鍵決策，讓您自動遵守合規要求。
5. 利用已知的產業架構和功能，例如「價值串流分析」和 LEAN，對目前的狀態效能、業務指標進行基準分析，並定義這些指標的進度迭代。

實作計劃的工作量：中高

資源

相關的最佳實務：

- [OPS01-BP05 評估威脅環境](#)

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：

- [使用資料做出明智決策 \(DevOps Sagas\)](#)
- [使用開發價值串流映射來識別 DevOps 結果的限制](#)

操作模型

在本節中，我們提供了一種方法來了解您所使用的操作模型、如何視覺化模型，以及在團隊層級上您應該如何發展以從雲端服務投資中獲取最大價值。透過這樣做，您可以增強操作實務，建置敏捷的團隊和工作負載，並為業務成果做出積極貢獻。

您的團隊通常存在於多個組織層級中，而這些層級都具有現有的工作方式。與您的團隊一起實現業務成果意味著了解您的團隊在這些層級中的位置、與您互動的團隊的位置以及他們的工作方式。此外，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、知道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。

這些層級構成了組織的整體操作模式。組織如何發揮作用來交付業務成果取決於許多因素，例如類型、產業、地理位置、規模和自治程度。但是，它可能分為三大類：

- 集中
- 分散式
- 聯合

這些組織層級拓撲在[成功組織](#)中加以說明。

您的團隊和工作負載存在於您組織的操作模式中。但是，無法要求單一操作模式支援所有團隊及其工作負載。因此，您的團隊也需要自己的操作模式。這種工作方式取決於您的組織、您的部門、團隊組成，以及工作負載本身的特性。

大多數遷移至雲端的組織都是將其作為企業轉型計畫的一部分，該計畫旨在解鎖新的工作方式 (操作模式) 以支援長期策略目標。此旅程不是一個時間點演練，而是一個需要不斷演進、逐步推進策略目標的過程。這可讓工作負載擁有者以最少的干擾來適應不斷發展的操作模式。

Amazon 經常用作大型組織如何大規模創新的範例，方法是讓團隊貼近客戶、快速推出創新產品和服務，以及利用支援速度和敏捷性的技術架構。這要求我們重組團隊的組織方式，現在稱為雙披薩團隊。一個雙披薩團隊內嵌了所有適當的資源 (工程、測試、產品和計畫管理及營運) 來擁有和執行端對端的工作負載。

我們建議將此操作模式作為一種行之有效的方法，讓工作負載團隊快速行動，並以最好的方式為客戶服務，為整體業務成果做出貢獻。

尋求效仿這種成功的組織可能需要在整個轉型之旅中調整其操作模式。在組織和團隊層級，這都需要考慮、規劃和溝通。下節提供了一種方式，視覺化說明這些團隊層級操作模式及其如何發展為誰開發誰執行。

操作模式 2 x 2 表示法

這些操作模式 2 x 2 表示法屬於圖示，可協助您了解您環境中團隊間的關係。這些圖表著重於相應人員負責的相應工作和團隊間的關係，但我們也將討論在這些範例中的管控和決策制定。

視支援的工作負載而定，您的團隊可能在數種模式中擔負數個部分的責任。您可能希望打破更專業的學科領域，而非描述的高階領域。隨著您劃分或彙整活動，或重疊團隊並提供更具體的詳細資訊，這些模式可能會有無限的變化。

您可能會發現跨團隊間會有能夠提供其他優勢或發揮效率的重疊或尚未辨識的功能。您也可識別貴組織內尚未滿足，但自己打算解決的需求。

評估組織變更時，請檢視模式間的權衡、您的個別團隊位於模式中的哪個階段 (目前和變更後)、您團隊的關係和責任將如何變更，以及效益是否能夠影響貴組織。

您可以使用以下四種操作模式獲得成功。部分模式更適用於特定使用案例，或部署中的特定時間點。在您的環境中使用時，部分模式可能比其他模式更具優勢。

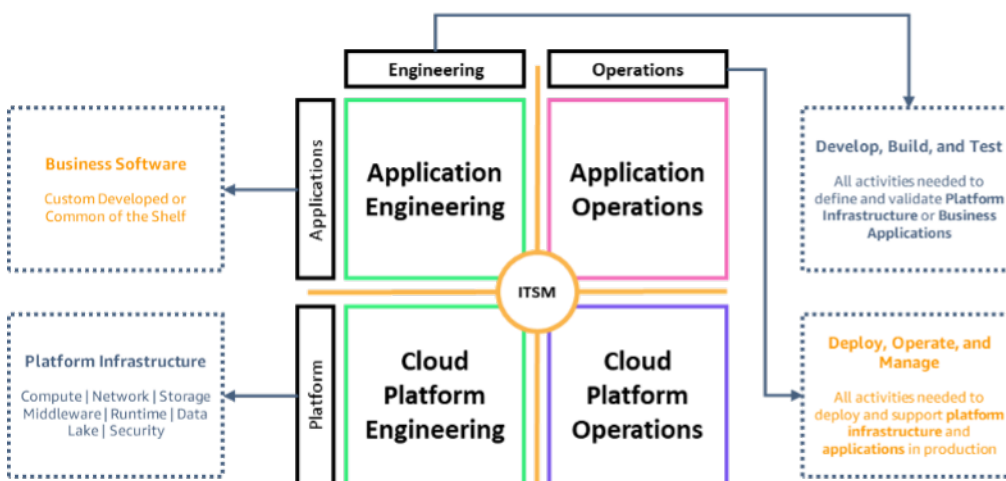
主題

- [完全獨立的操作模式](#)
- [DevOps 搭配雲端受管服務供應商](#)
- [雲端維運和平台啟用 \(COPE\)](#)
- [分散式 DevOps](#)
- [分散式 DevOps](#)
- [改進您的操作模式](#)

完全獨立的操作模式

應用程式和平台位於下圖的垂直軸上。應用程式是指幫助實現業務成果的工作負載，可以是自訂開發或採購的軟體。平台是指實體和虛擬基礎設施，以及支援該工作負載的其他軟體。

工程和操作位於水平軸上。工程是指開發、建置和測試應用程式和基礎設施。操作是指部署、更新及持續支援應用程式和基礎設施。



傳統模式

從歷史上看，組織採用了 ITIL 等架構或 ISO 等標準，並圍繞這些架構或標準制定營運活動，這通常會導致完全分離的拓撲。在此模式中，各象限中的活動由個別團隊執行。工作透過工作請求、佇列、票證或使用 IT 服務管理 (ITSM) 系統等機制，在團隊之間轉交。

任務轉給團隊或團隊間的任務轉交會讓事情變得更複雜，並形成瓶頸與延誤。除非是要優先處理的請求，否則請求可能延誤處理。如果晚發現缺陷，可能需要大量重新作業，也可能需要再次交由相同的團隊和職務處理。如果發生需由工程團隊採取行動的事件，其回應會因遞交活動而受到延誤。

當業務、開發和營運團隊根據正在執行的活動或職務組織時，目標不一致的風險會升高。如此會導致讓團隊專注於其特定責任，而非專注於達成業務成果。團隊可能專精於狹隘的領域，或在實際或邏輯空間上遭到分離，因而阻礙溝通與合作。

DevOps 搭配雲端受管服務供應商

「DevOps 搭配雲端受管服務供應商」模式遵循應用程式團隊的誰開發誰執行方法。但是，貴組織現有的技能或團隊成員可能無法支援專門的平台工程和營運團隊，或您可能不適合投入時間和人力物力來進行此工作。

或者，您可能想要擁有一個專注於建立讓您的企業脫穎而出之功能的平台團隊，但希望將無差異化的日常作業外包。

[AWS Managed Services](#) 等受管服務供應商或 [AWS Partner Network](#) 中的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。

DevOps 搭配雲端受管服務供應商

就此變化而言，我們將透過平台團隊進行集中管控和管理，使用 AWS Organizations 和 AWS Control Tower 建立帳戶並管理政策。

此模式需要您修改機制，以與您的服務供應商的機制合作。它不會處理因團隊間 (包括您的服務供應商) 轉交任務而導致的瓶頸和延誤，或因晚發現缺陷而導致的潛在重新作業。

您可以獲得供應商標準、最佳實務、流程和專業知識的優勢。您也可以從其服務產品持續開發中獲得效益。

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。它還可以為您提供時間來建置和成熟自己的平台功能，而不會降低雲端遷移計畫的速度。

雲端維運和平台啟用 (COPE)

此雲端維運和平台啟用 (COPE) 模式旨在透過支援應用程式團隊採用 DevOps 文化為工作負載執行工程和營運活動，從而建立誰開發誰執行方法。

您的應用程式團隊可能負責遷移、採用雲端或現代化工作負載，但可能不具有充分支援雲端架構和營運的現有技能。應用程式團隊能力和熟悉度的缺乏可能會降低組織的敏捷性並影響業務成果。

為了解決此問題，請使用貴組織內現有的營運專業知識來支援應用程式團隊的雲端維運之旅。這可以是一個專門的專家團隊，也可以是一個虛擬團隊，參與者是從整個組織中選出的。但是，目標保持不變，即提供營運支援，以培養工作負載團隊的能力，使用雲端優先的自動化原則，消除無差別的繁重工作，提供標準化模式，並促進自主性。目標是跨雲端功能培養足夠的成熟度並降低營運責任的障礙，以便應用程式團隊不再需要額外的支援。

COPE 模式著重於工作負載層級。如果多個團隊同時需要此方法，如果您正在執行複雜、大規模的多年遷移專案，或者您要建置支援這些計畫的平台，請考慮使用雲端卓越中心 (CCoE)。許多人在尋求加速遷移到雲端並廣泛轉型組織時，發現這種機制很成功。

雲端維運和平台啟用 (COPE)

您的平台工程團隊會建置一層薄型的核心共用平台功能，這些功能是以預先定義的標準為基礎，供應用程式團隊採用並由 COPE 團隊提供。平台工程團隊編制透過自助服務機制提供給應用程式團隊的企業參考架構和模式。使用 AWS Service Catalog 等服務，應用程式團隊可以部署已核准的參考架構、模式、服務和組態，預設情況下符合集中控管和安全標準。

平台工程團隊還為應用程式團隊提供一組標準化服務 (例如，開發工具、可觀測性工具、備份和復原工具及聯網)。

COPE 團隊管理和支援標準化服務，並為應用程式團隊根據參考架構和模式建立雲端服務提供協助。他們與應用程式團隊合作，協助他們建立基準作業。在此過程中，應用程式團隊隨著時間的推移逐步對系統和資源承擔更多責任。COPE 團隊與平台工程團隊一起推動持續改進，並擔任應用程式團隊的擁護者。

應用程式團隊在設定環境、CI/CD 管道、變更管理、可觀測性和監控以及建立事件和事件管理程序方面取得協助，並視需要整合 COPE 團隊。COPE 團隊與應用程式團隊一起參與進行這些操作活動，隨著應用程式團隊取得擁有權，COPE 團隊會隨著時間的推移逐步停止參與。

應用程式團隊受益於 COPE 團隊的技能和組織所學到的經驗。其受到透過集中控管建立的防護機制保護。應用程式團隊以公認的成功案例為基礎，並從他們採用的組織標準的持續發展中獲益。透過建立可

觀測性和監控的程序，他們可以更深入地了解工作負載的操作，並且能夠更好地了解他們對工作負載所做的變更的影響。

COPE 團隊還可以保留支援操作活動所需的存取權，提供跨應用程式團隊的企業操作視圖，並提供關鍵事件管理支援。COPE 團隊保留對視為無差別繁重工作的活動的責任，他們透過可大規模支援的標準解決方案來滿足這些活動。他們還繼續為應用程式團隊管理易於理解的程式設計和自動化操作活動，以便他們能夠專注於差異化其應用程式。

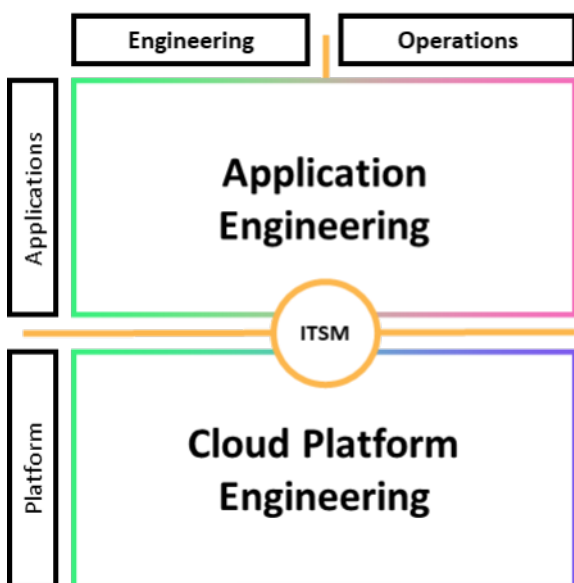
您可以取得貴組織源自團隊成功案例的標準、最佳實務、程序和專業知識的優勢。可建立一種機制來複寫這些成功模式，以便新團隊在雲端中採用或實現現代化。此模式強調 COPE 團隊協助應用程式團隊建立及轉換知識和成品的能力。它減輕了應用程式團隊的操作負擔，但也帶來了應用程式團隊無法變得獨立的風險。它在平台工程、COPE 和應用程式團隊之間建立關係，建立回饋迴圈以支援進一步發展和創新。

建立平台工程和 COPE 團隊，同時定義組織範圍內的標準，可以促進雲端採用並支援現代化工作。透過提供 COPE 團隊作為應用程式團隊的顧問和合作夥伴的額外支援，您可以消除工作負載層級的障礙，這些障礙會減緩應用程式團隊採用有益的雲端功能的速度。

分散式 DevOps

分散式 DevOps 模式遵循 [COPE 方法](#)，在工程團隊中分隔 (或分配) 應用程式工程操作和基礎設施工程操作職責。

您的應用程式工程師會執行其工作負載的工程和操作。相同地，您的基礎設施工程師會執行其用於支援應用程式團隊的平台工程和操作。



分散式 DevOps

在此範例中，我們將管控視為集中在組織內的其他地方。標準會分發、提供給應用程式和平台團隊，或與之共用。

使用可協助您集中管控跨帳戶環境的工具或服務，例如 [AWS Organizations](#)、[AWS Control Tower](#) 等服務會擴大此管理功能，協助您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。

誰開發誰執行並不表示應用程式團隊負責全堆疊、工具鏈和平台。

平台工程團隊為應用程式團隊提供一組標準化服務 (例如，開發工具、監控工具、備份和復原工具及聯網)。平台團隊也可將核准之雲端供應商服務、相同的特定組態或兩者的存取權提供給應用程式團隊。

提供部署核准之服務和組態的自助服務功能的機制 (例如 Service Catalog) 有助於限制在強制執行管控時與履行請求關聯的延遲。

平台團隊可啟用全堆疊可見性，以便應用程式團隊可以區分其應用程式元件和服務，以及其應用程式耗用之基礎設施元件所發生的問題。平台團隊也可提供設定這些服務的協助，以及如何改善應用程式團隊營運的指引。

如之前所述，機制存在的目的應為請求標準新增、變更及例外狀況，以支援活動及其應用程式的創新。

分散式 DevOps 模式提供健全的意見回饋迴圈給應用程式團隊。工作負載的日常操作會透過直接互動，或間接透過支援和功能請求，而增加與客戶的接觸。如此提高的可見性有助於應用程式團隊更迅速處理問題。透過更深入參與和更密切的關係能深入了解客戶需求，並更迅速地實現創新。

所有這一切對於支援應用程式團隊的平台團隊也是如此，因為平台團隊應將這些應用程式團隊視為客戶。

採用的標準可能事先經過核准可以使用，進而減少進入生產所需的審查工作量。使用平台團隊提供的支援和經過測試的標準，可能會減少這些服務發生問題的頻率。採用標準可協助應用程式團隊專注於差異化工作負載。

分散式 DevOps

分散式 DevOps 模式是誰開發誰執行方法的變體，其中操作主要由工作負載團隊負責。

您的應用程式工程師會執行工作負載的工程和操作。相同地，您的基礎設施工程師會執行其用於支援應用程式團隊的平台工程和操作。

分散式 DevOps

在此範例中，我們進行分散管控。標準仍會由平台團隊分發、提供給應用程式團隊，或與之共用，但應用程式團隊可自由設計和操作新的平台功能，以支援其工作負載。

在此模式中，應用程式團隊受到的限制較少，但伴隨而來的卻是責任大幅增加。必須擁有其他技能 (和潛在的團隊成員)，以支援其他平台功能。如果技能組合不足，且無法及早辨識缺陷，則大量重新作業的風險會提高。

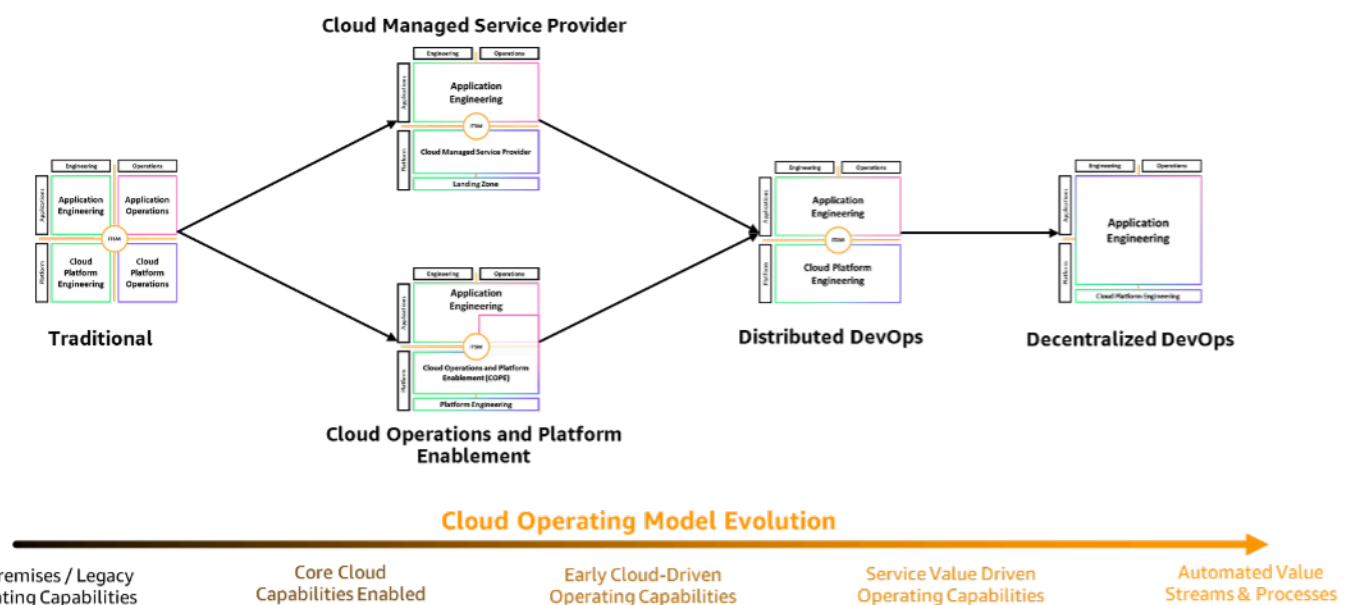
強制執行並非專門委派給應用程式團隊的政策。使用可協助您集中管控跨帳戶環境的工具或服務，例如 [AWS Organizations](#)、[AWS Control Tower](#) 等服務會擴大此管理功能，協助您定義帳戶設定的藍圖 (支援您的操作模式)、使用 AWS Organizations 套用持續管控，以及自動化新帳戶的佈建。

讓應用程式團隊設有請求標準新增和變更的機制，將會帶來好處。他們可以貢獻可使其他應用程式團隊受益的新標準。平台團隊可能會認為，為這些其他功能直接提供支援，能有效幫助實現業務成果。

此模式透過重要技能和團隊成員要求，減少對於創新的限制。它解決了團隊間轉交任務所導致的許多瓶頸和延誤，同時仍促進團隊與客戶之間建立有效的關係。

改進您的操作模式

提供的模式在工作負載層級逐步邁向更多自主性，符合「雙比薩團隊」原則。重要的是要了解，從傳統方法到分散式 DevOps (作為持續發展到「雙比薩團隊」模式的基礎) 的旅程可能需要時間，並且需要在許多功能方面建置成熟度。因此，我們提供了一個範例，說明當您的團隊和組織在企業轉型之旅中如何在模式之間轉換。在每個變更或模式中，您都在朝著一個更自主，但仍然與組織保持一致的團隊發展。



雲端操作模式發展

在評估您的團隊如何支援貴組織發展時，請檢查模式之間的權衡、您的個別團隊位於模式中的哪個階段(在轉型和發展時)、您團隊的關係和責任可能如何變更，以及效益是否能夠影響貴組織。請記住，變化從來都不是線性的。某些模式更適合旅程中的特定使用案例或時間點，而其中一些模式可能比您環境中的模式具有優勢。

關係和擁有權

您的操作模式定義團隊之間的關係，並支援可識別的擁有權和責任。

最佳實務

- [OPS02-BP01 資源已識別擁有者](#)
- [OPS02-BP02 流程和程序已識別擁有者](#)
- [OPS02-BP03 已為營運活動識別負責其效能的擁有者](#)
- [OPS02-BP04 存在管理責任和擁有權的機制](#)
- [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 流程和程序已識別擁有者

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 等服務，使用文件和 AWS Lambda 來擷取活動成品中繼資料中的擁有權。使用標籤或資源群組擷取資源擁有權，並指定擁有權和聯絡資訊。使用 AWS Organizations 建立標記政策，並擷取擁有權和聯絡資訊。
- 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 Retail 將擁有權定義為擁有應用程式或應用程式群組 (共享共同的架構實務和技術) 流程的團隊或個人。最初，流程和程序作為分步指南記錄在文件管理系統中，可使用託管應用程式的 AWS 帳戶中的標籤以及帳戶內特定資源群組中的標籤進行探索。他們利用 AWS Organizations 來管理他們的 AWS 帳戶。隨著時間的推移，這些流程會轉換為程式碼，並使用基礎設施即程式碼 (例如 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本) 來定義資源。操作流程會變成 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，這些自動化文件可以作為排程任務啟動，以回應諸如 AWS CloudWatch 警示或 AWS EventBridge 事件之類的事件，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. 審核並將其維持在最新狀態。
 - 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. 定義審核流程頻率的策略。
 - b. 定義審核者與核准者的流程。
 - 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，流程和程序應得到變更核准委員會 (CAB) 的預先批准和風險分類。
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. 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 40 多個受支援的資料來源提供預先建置的連接器，其中包括 Amazon S3、Microsoft SharePoint、Salesforce 和 Atlassian Confluence。如需詳細資訊，請參閱 [Amazon Q 連接器](#)。
5. 適時進行自動化。
 - a. 當服務和技術提供 API 時，應該開發自動化。
 - 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，並建立問題或票證以支援反覆改進。

實作計劃的工作量：中

資源

相關的最佳實務：

- [OPS02-BP01 已為資源識別擁有者](#)
- [OPS02-BP04 存在管理責任和擁有權的機制](#)
- [OPS11-BP04 執行知識管理](#)

相關文件：

- [AWS 白皮書 - DevOps on AWS 簡介](#)
- [AWS 白皮書 - 標記 AWS 資源的最佳實務](#)
- [AWS 白皮書 - 使用多個帳戶組織您的 AWS 環境](#)
- [AWS 雲端 操作和遷移部落格 - 使用 Amazon Q Business 簡化您的操作](#)
- [AWS 雲端 操作和遷移部落格 - 建立雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務](#)

- [AWS 雲端 操作和遷移部落格 - 使用 AWS Config 和 AWS Organizations 實作自動化和集中式標記控制](#)
- [AWS 安全部落格 - 使用 AWS CloudFormation Guard 擴展您的預先提交勾點](#)
- [AWS DevOps 部落格 - 將 AWS CloudFormation Guard 整合到 CI/CD 管道中](#)

相關研討會：

- [AWS Well-Architected 卓越營運研討會](#)
- [AWS 研討會 - 標記](#)

相關影片：

- [如何在 AWS 上自動執行 IT 操作](#)
- [AWS re:Invent 2020 - 使用 AWS Systems Manager 自動執行任何操作](#)
- [AWS re:Inforce 2022 - 使用 AWS 自動化修補程式管理和合規性 \(NIS306\)](#)
- [支援 支援您 - 深入了解 AWS Systems Manager](#)

相關服務：

- [AWS Systems Manager - 自動化](#)
- [AWS Service Management Connector](#)

OPS02-BP03 已為營運活動識別負責其效能的擁有者

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

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審核並更新責任，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應用程式或應用程式群組 (共享共同的架構實務和技術) 流程的團隊或個人。最初，公司將流程和程序記錄為文件管理系統中的分步指南。其會使用託管應用程式的 AWS 帳戶中的標籤以及帳戶內特定資源群組中的標籤來探索程序，並使用 AWS Organizations 來管理

其 AWS 帳戶。隨著時間的推移，AnyCompany Retail 會將這些流程轉換為程式碼，並使用基礎設施即程式碼 (例如 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本) 來定義資源。操作流程會變成 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，它們可以作為排程任務啟動，以回應諸如 Amazon CloudWatch 警示或 Amazon EventBridge 事件之類的事件，或由 IT 服務管理 (ITSM) 平台內的請求啟動。所有流程都有標籤，可用來識別其擁有者。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. 審核並確認其是否為最新狀態。
 - 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. 定義審核流程頻率的政策。
 - b. 定義審核者與核准者的流程。
 - c. 實作問題或票證隊列，以提供並追蹤意見回饋。
 - d. 在可能的情況下，為流程和程序提供變更核准委員會 (CAB) 的預先批准和風險分類。
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. 當服務和技術提供 API 時，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 已為資源識別擁有者](#)

- [OPS02-BP02 流程和程序已識別擁有者](#)
- [OPS02-BP04 存在管理責任和擁有權的機制](#)
- [OPS02-BP05 存在識別責任和擁有權的機制](#)
- [OPS11-BP04 執行知識管理](#)

相關文件：

- [AWS 白皮書 | DevOps on AWS 簡介](#)
- [AWS 白皮書 | 標記 AWS 資源的最佳實務](#)
- [AWS 白皮書 | 使用多個帳戶組織您的 AWS 環境](#)
- [AWS 雲端 操作和遷移部落格 | 建立雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務](#)
- [AWS 研討會 - 標記](#)
- [AWS Service Management Connector](#)

相關影片：

- [AWS 知識中心直播 | 標記 AWS 資源](#)
- [AWS re:Invent 2020 | 使用 AWS Systems Manager 自動執行任何操作](#)
- [AWS re:Inforce 2022 | 使用 AWS 自動化修補程式管理和合規性 \(NIS306\)](#)
- [支援 支援您 | 深入了解 AWS Systems Manager](#)

OPS02-BP04 存在管理責任和擁有權的機制

了解您角色的責任以及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，設計一種機制來解決此差距。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。
- 您的團隊不會更新職責，這導致其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)。隨著組織尋求利用機會在 AWS 上進行遷移和現代化，角色和職責也可能會改變。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 Retail 最近使用平移方法，完成了在 AWS 中將工作負載從內部部署環境遷移到其登陸區域。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 AWS 時，他們減少了與硬體和實體基礎結構相關的基礎架構團隊的責任。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。若要探索標準化和改善工作負載效率的新機會，請提供 AWS Systems Manager 等操作工具以及 AWS Security Hub 和 Amazon GuardDuty 等安全工具的存取權。AnyCompany Retail 根據他們希望首先解決的改進來審查責任和策略。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. 責任或負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣
 - b. 團隊定義或 Wiki 頁面
 - c. 服務定義和產品
 - 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. 找出頻繁發出的資源密集型請求，這類請求通常表示需要改進。
 - b. 尋找最佳實務、了解模式、遵循規範性指引，並簡化和標準化改進。
 - 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. 審核流程，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 授權團隊成員在成果有風險時採取動作](#)
- [OPS03-BP03 鼓勵向上呈報](#)
- [OPS03-BP07 適當地為團隊提供資源](#)
- [OPS09-BP01 使用指標衡量營運目標與 KPI](#)
- [OPS09-BP03 審核營運指標並優先改進](#)
- [OPS11-BP01 建立持續改進程序](#)

相關文件：

- [AWS 白皮書 - DevOps on AWS 簡介](#)
- [AWS 白皮書 - AWS 雲端 採用架構：營運觀點](#)
- [AWS Well-Architected Framework 卓越營運 - 工作負載層級操作模型拓撲](#)
- [AWS 方案指引 - 建置雲端操作模式](#)
- [AWS 方案指引 - 為雲端操作模式建立 RACI 或 RASCI 矩陣](#)
- [AWS 雲端 操作和遷移部落格 - 透過雲端平台團隊提供商業價值](#)
- [AWS 雲端 操作和遷移部落格 - 為何選擇雲端操作模式？](#)
- [AWS DevOps 部落格 - 組織如何實現雲端操作的現代化](#)

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 流程和程序已識別擁有者](#) 或 [OPS02-BP03 已為營運活動識別負責其效能的擁有者](#)，請先從這些項目開始。

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 流程和程序已識別擁有者](#) - 在建立變更管理流程之前，流程需要確定的擁有者。
- [OPS02-BP03 已為營運活動識別負責其效能的擁有者](#) - 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領導調查。

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 流程和程序已識別擁有者](#) - 在團隊之間達成協議之前，必須確定流程所有權。
- [OPS02-BP03 已為營運活動識別負責其效能的擁有者](#) - 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 - 透過 Two-Pizza 團隊促進創新](#)
- [DevOps on 的簡介 AWS - Two-Pizza Teams](#)

組織文化

為您的團隊成員提供支援，讓他們能夠更有效地採取動作以及支援業務成果。

最佳實務

- [OPS03-BP01 提供高層支持](#)
- [OPS03-BP02 授權團隊成員在成果有風險時採取動作](#)
- [鼓勵 OPS03-BP03 升級](#)
- [OPS03-BP04 溝通需及時、清楚且可行](#)
- [OPS03-BP05 鼓勵進行試驗](#)
- [OPS03-BP06 管理團隊成員維持和發展自己的技能集](#)
- [OPS03-BP07 資源團隊適當](#)

OPS03-BP01 提供高層支持

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。領導層積極制定此方向、分配擁有權、承擔責任並定義工作。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劃的情況下，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：當高層的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，組織得以運用最強大的能力來獲得成功。當您評估成功時，您可以更好地識別成功障礙，以便透過高層支持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佈目標，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 Retail 正在透過快速重新打造客戶體驗、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劃。
 - 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。

實作計劃的工作量：中

資源

相關的最佳實務：

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

相關文件：

- [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 CCOE 時應避開的 7 大陷阱](#)
- [導覽雲端：成功的關鍵績效指標](#)

相關影片：

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

相關範例：

- [Prosci：主要倡議者的角色和重要性](#)

OPS03-BP02 授權團隊成員在成果有風險時採取動作

由領導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：領導層的影響力擴大，培養出新的文化，讓組織內較低層級的個人和團隊也能做出關鍵決策 (只要是透過可稽核的許可和安全機制來定義決策即可)。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 團隊負責部署工具，而部署的自動復原則在他們的長期藍圖中。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流程。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。
- 為了保持一致，資訊安全團隊會監督一個歷史悠久的流程，以代表連線至其 Amazon EC2 Linux 執行個體的營運商定期輪換共用的 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。資訊安全團隊內外沒有人建議在 AWS 上使用其他選項來實現相同的結果。

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 評估權衡，同時管理收益和風險](#)
- [OPS02-BP05 存在識別責任和擁有權的機制](#)

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 | 如何不破壞轉型 \(SEG201\)](#)
- [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安全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

- 定義組織的政策、標準和期望。
 - 確保廣泛採用和理解政策、期望和標準。
- 在不符合標準時，鼓勵、培訓和授權工人儘早和頻繁呈報。
- 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。
 - 建立呈報機制 (例如 Andon Cord 系統)。
 - 制定書面程序，定義向上呈報的時機與方式。
 - 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。
- 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。
 - 呈報應包括：
 - 情況描述和風險性質
 - 情況嚴重性
 - 誰或什麼受到影響
 - 影響有多大
 - 發生影響時的緊迫性
 - 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 溝通需及時、清楚且可行

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：

- 您的組織有五年計劃，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。CIO 將此策略傳達給直接下屬，並期望每位領導者將此演示文稿提交給經理、董事和個人貢獻者，而無須進行任何面對面溝通。CIO 退後一步，並期望其組織執行新的策略。
- 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。經理會轉發來自資訊安全網副總裁的電子郵件，並新增訊息「去實現 (Make this happen)」。
- 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劃的事件。
- 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 Retail 已註冊 AWS Enterprise Support，並依賴其他第三方提供商進行其雲端運營。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劃領導層協作，代表領導者為適當的員工製作訊息。
5. 建立策略性溝通機制，透過公告、共用行事曆、全體會議，以及面對面或一對一的方法來管理變更，讓團隊成員對應採取的行動有適當的期望。
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. 建立一種回饋循環，用以衡量所有溝通的有效性，尤其是與整個組織中抗拒改變的力量相關的溝通。
10. 對於所有 AWS 帳戶，建立帳單、安全性和操作的[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. 建立呈報和逆向呈報溝通計畫，以與您的內部和外部團隊 (包括 AWS Support 和其他第三方供應商) 互動。
12. 在每個轉型計劃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](#) 可用於在發生變更時建立變更時段。這可為團隊成員提供有關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、尚未提供服務的市場區隔，以及環境中未確認的風險)。
 - c. 鼓勵團隊成員充分參與：提供員工充分參與所有與工作相關的活動所需的資源。面對日常挑戰的團隊成員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 提供高層支持](#)
- [OPS07-BP03 使用執行手冊執行程序](#)
- [OPS07-BP04 使用程序手冊調查問題](#)

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](#)
- [開放式 CVE](#)
- [Slack 中用於管理支援案例的 支援 應用程式](#)
- [使用聊天應用程式中的 Amazon Q Developer 來管理 Slack 頻道中的 AWS 資源](#)

相關服務：

- [聊天應用程式中的 Amazon Q Developer](#)
- [AWS Systems Manager 變更行事曆](#)
- [AWS Systems Manager 文件](#)

OPS03-BP05 鼓勵進行試驗

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

預期成果：

- 您的組織鼓勵試驗以促進創新。
- 試驗被視為一種學習機會。

常見的反模式：

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

建立此最佳實務的優勢：

- 試驗可帶動創新。

- 透過試驗，您可以更快回應使用者的意見反映。
- 組織可培養學習文化。

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

實作步驟

1. 與組織中的領導階層共同推行試驗風氣。應鼓勵團隊成員以安全的方式執行試驗。
2. 為團隊成員提供可安全進行試驗的環境。他們必須能夠存取類似生產環境的環境。
 - a. 可以使用單獨的 AWS 帳戶來建立用於實驗的沙盒環境。[AWS Control Tower](#) 可用於佈建這些帳戶。
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](#) 能夠建立功能旗標。
 - b. 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 AWS 中建立和管理沙盒帳戶的最佳實務](#)
- [建立由雲端啟用的實驗文化](#)
- [在 SulAmérica Seguros 實現雲端實驗和創新](#)
- [實驗越多，失敗越少](#)
- [使用多個帳戶來組織您的 AWS 環境 - 沙盒 OU](#)
- [使用 AWS AppConfig 功能旗標](#)

相關影片：

- [AWS On Air ft. Amazon CloudWatch Evidently | AWS 活動](#)
- [AWS On Air San Fran Summit 2022 ft. AWS AppConfig 功能旗標與 Jira 整合](#)
- [AWS re:Invent 2022 - 部署不是發行版本：透過功能旗標控制您的啟動 \(BOA305-R\)](#)
- [使用 AWS Control Tower 以程式設計方式建立 AWS 帳戶](#)
- [設定使用 AWS Organizations 最佳實務的多帳戶 AWS 環境](#)

相關範例：

- [AWS 創新沙盒](#)
- [電子商務端對端個人化 101](#)

相關服務：

- [Amazon CloudWatch Evidently](#)
- [AWS AppConfig](#)
- [AWS Control Tower](#)

OPS03-BP06 管理團隊成員維持和發展自己的技能集

團隊必須發展自己的技能集，以採用新技術，並支援需求和責任的變更，以支援您的工作負載。新技術的技能成長通常是團隊成員滿意度的來源，並可支援創新。支援團隊成員追求和維持產業認證，以驗證和認可他們不斷成長的技能。交叉培訓以促進知識轉移，並在失去熟練的、經驗豐富且具備機構知識的成員時，降低重大影響的風險。提供學習專用的結構化時間。

AWS 可提供資源，包括 [AWS 入門資源中心](#)、[AWS 部落格](#)、[AWS 線上技術講座](#)、[AWS 活動和網路研討會](#) 以及 [AWS Well-Architected 實驗室](#)，提供指引、範例和詳細的逐步解說來教育您的團隊。

諸如 [支援](#) ([AWS re:Post](#)、[支援 Center](#)) 等資源和 [AWS 文件](#) 有助於消除技術障礙並改善操作。透過 [支援中心](#) 聯絡支援，以獲取相關問題的幫助。

AWS 還分享了我們透過在 [Amazon 建置者資料中心](#) 操作 AWS 所學到的最佳實務和模式，以及透過 [AWS 部落格](#) 和 [官方 AWS 部落格](#) 提供的各種其他有用的教育材料。

[AWS 培訓和認證](#) 透過自定進度的數位課程提供免費培訓，以及按角色或領域制定的學習計劃。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。

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。團隊利用專門的交叉分享知識計劃，例如午餐學習會、Immersion Day、黑客松和演練日。您的組織將其知識系統保持在最新狀態，並與交叉培訓團隊成員相關，包括新員工入職培訓。

常見的反模式：

- 在沒有結構化培訓計劃和預算的情況下，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。
- 作為遷移到 AWS 的一部分，您的組織展現了團隊間的技能差距和不同的雲端流暢度。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。針對性的學習計劃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劃：[AWS Skills Guild](#) 提供諮詢培訓，以提高雲端技能信心並培養持續學習的文化。

2. 提供教育資源：提供專門的結構化時間，以及取用培訓教材和實驗室資源的權限，並支援參與會議和專業組織，從中為教育工作者和同儕提供學習機會。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. 建置並維護最新的知識儲存庫：使用知識共享平台，例如 Wiki 和執行手冊。使用 [AWS re:Post Private](#) 建立您自己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 提供高層支持](#)
- [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](#)
- [支援](#)
- [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 播客](#)。

- [AWS 線上技術講座](#)
- [AWS 活動和研討會](#)
- [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。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

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。確定關鍵績效指標 (KPI) 對於確保監控活動與業務目標之間的一致性至關重要。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。團隊可以了解系統內的因果關係，預測和預防問題，而不僅僅是對問題做出反應。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 允許您以程式碼檢視您的整個工作負載 (應用程式、基礎設施、原則、管控和營運)。這表示您可以將用於應用程式程式碼的相同工程規則套用到堆疊的每個元素，並在團隊或組織之間分享這些元素，以擴大開發工作的優勢。在雲端以程式碼執行營運，並利用安全進行試驗的能力，開發工作負載、營運程序以及實務失敗案例。使用 AWS CloudFormation 允許您擁有一致的範本化沙盒開發、測試和生產環境，同時還能提高營運控制等級。

對以程式碼形式實作營運活動進行投資，從而最大程度地提高營運人員的生產力，將錯誤率降至最低以及實現自動回應。使用「事前剖析」可預測失敗並適時建立程序。依照一致的標記策略，使用資源標籤和 AWS Resource Groups 來套用中繼資料，以識別您的資源。標記您的資源，以用於組織、成本會

計、存取控制，以及將自動執行營運活動設為目標。採用可利用雲端彈性的部署實務，以促進開發活動和系統的預部署，進而加快實作速度。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。

主題

- [實作可觀測性](#)
- [營運設計](#)
- [緩解部署風險](#)
- [營運準備度和變更管理](#)

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。

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。

確定關鍵績效指標 (KPI) 對於確保監控活動與業務目標之間的一致性至關重要。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。團隊可以了解系統內的因果關係，預測和預防問題，而不僅僅是對問題做出反應。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。

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，您可能不會意識到效能瓶頸或問題。這種被動回應的方式可能導致客戶不滿意。
 - 缺乏使用者體驗洞察：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](#) 來設計 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 的實際使用者監控 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，可提供連線能力、逾時，以及像是 DNS、資料庫或第三方 API 等其他與相依性相關重要事件的寶貴洞見。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。相依性無法按需求運作時，使用 AWS 工具探索並量化其影響。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。外部相依性可能包含各種服務，像是外部資料庫、第三方 API、前往其他環境的網路連線能力路由，以及 DNS 服務。實現有效相依性遙測的第一步，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。這包括了解每一項相依性的重要性、預期行為，以及任何相關的服務層級協議或目標 (SLA 或 SLT)。設定主動警示，以便在發生狀態變更或效能偏差時通知您。
3. 使用[網路監控](#)：使用[網際網路監控](#)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. 利用 [AWS Health](#) 隨時掌握新知：AWS Health 是 AWS 雲端 資源運作狀態的權威資訊來源。使用 AWS Health 視覺化並接收有關任何目前服務事件和近期變更的通知 (例如規劃的生命週期事件)，如此您就能採取行動來緩解衝擊。
 - a. 透過 [AWS 使用者通知](#) 建立符合用途的 [AWS Health 事件通知](#)，以利用電子郵件和聊天管道傳送，並[透過 Amazon EventBridge 或 AWS Health API 以程式設計方式與您的監控和警示工具整合](#)。
 - b. 透過 Amazon EventBridge 或 AWS Health API 整合變更管理或您可能已在使用的 ITSM 工具 (如 [Jira](#) 或 [ServiceNow](#))，以規劃並追蹤需要採取行動的運作狀態事件進度。

- c. 如果您使用 AWS Organizations，請啟用 [AWS Health 的組織檢視](#)，以彙總帳戶之間的 AWS Health 事件。
5. 使用 [AWS X-Ray](#) 檢測您的應用程式：AWS X-Ray 提供了深入了解應用程式及其基礎相依性運作效能的洞見。透過從頭到尾追蹤請求，您就可以找出應用程式所依賴的外部服務或元件的瓶頸或故障。
6. 使用 [Amazon DevOps Guru](#)：這項機器學習驅動的服務可識別操作問題，預測重大問題可能在什麼時候發生，並且建議可採取的特定行動。對於獲得相依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 KPI](#)
- [OPS04-BP02 實作應用程式遙測](#)
- [OPS04-BP03 實作使用者活動遙測](#)
- [OPS04-BP05 實作交易可追溯性](#)
- [OP08-BP04 建立可執行的提醒](#)

相關文件：

- [Amazon Personal 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》<https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html>
- [AWS DevOps Guru 使用者指南](#)

相關影片：

- [深入了解影響應用程式效能的網際網路問題](#)

- [Amazon DevOps Guru 簡介](#)
- [使用 AWS Health 大規模管理資源生命週期事件](#)

相關範例：

- [AWS Health Aware](#)
- [使用以標籤為基礎的篩選來大規模管理 AWS Health 監控和提醒](#)

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。說明所有元件後，請利用 AWS X-Ray 和 等工具 OpenTelemetry，透過 X-Ray 和 Amazon CloudWatch ServiceLens Map 等工具收集追蹤資料進行分析。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，以增強所取得洞見的深度。
5. 使用 [Amazon DevOpsGuru](#)：DevOpsGuru 使用來自 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](#)

營運設計

採用的方法需能夠改善變更發揮作用的流程，並有助於重構、快速提供品質意見回饋，以及修復錯誤。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，並能快速識別和修復部署活動造成的問題。

在 AWS 中，您可以程式碼檢視您的整個工作負載 (應用程式、基礎設施、原則、管控和營運)。所有這些均可在其中予以定義並使用程式碼進行更新。這表示您可以在您堆疊的每個元素中套用與您應用程式程式碼所用相同的工程原則。

最佳實務

- [OPS05-BP01 使用版本控制](#)
- [OPS05-BP02 測試並驗證變更](#)
- [OPS05-BP03 使用組態管理系統](#)
- [OPS05-BP04 使用建置和部署管理系統](#)
- [OPS05-BP05 執行修補程式管理](#)
- [OPS05-BP06 共用設計標準](#)
- [OPS05-BP07 實作用於提高程式碼品質的實務](#)
- [OPS05-BP08 使用多個環境](#)
- [OPS05-BP09 進行頻繁、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布。

許多 AWS 服務都提供版本控制功能。使用修訂版或[原始程式碼控制](#)系統 (例如 [Git](#)) 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](#)

相關影片：

- [AWS re:Invent 2023 - Lockheed Martin 如何採用 DevSecOps 技術更快建置軟體](#)
- [AWS re:Invent 2023 - GitHub 如何將 AI 營運化以應用到團隊協作和生產力](#)

OPS05-BP02 測試並驗證變更

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

您的組織必須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。
- 新的安全群組透過 AWS CloudFormation 進行部署，而未在生產前環境中測試。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

在其持續整合管道中，AnyCompany Retail 對所有軟體成品執行了數種類型的測試。他們實行測試驅動型開發，所以所有軟體都有單元測試。建置成品後，它們會執行端到端測試。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](#) 來測試您的應用程式碼是否存在故障。
 - 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 實作用於提高程式碼品質的實務](#)
- [OPS05-BP10 完全自動化整合和部署](#)

相關文件：

- [採用測試驅動的開發方法](#)
- [使用 Amazon Q 加速您的軟體開發生命週期](#)
- [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 IDE 中使用 Amazon Q Developer 的終極速查表](#)
- [左移工作負載，利用 AI 建立測試](#)
- [Amazon Q 開發人員中心](#)

- [使用 Amazon CodeWhisperer 快速建置應用程式的 10 種方式](#)
- [使用 Amazon CodeWhisperer 超越程式碼覆蓋範圍](#)
- [使用 Amazon CodeWhisperer 進行提示詞工程的最佳實務](#)
- [使用 TaskCat 和 CodePipeline 的自動化 AWS CloudFormation 測試管道](#)
- [使用開放原始碼 SCA、SAST 和 DAST 工具建置端對端 AWS DevSecOps CI/CD 管道](#)
- [開始測試無伺服器應用程式](#)
- [CI/CD 管道是我的發行主管](#)
- [《在 AWS 上實行持續整合和持續交付》白皮書](#)

相關影片：

- [使用適用於軟體開發的 Amazon Q Developer 代理程式來實作 API](#)
- [使用 JetBrains IDE 安裝、設定和使用 Amazon Q Developer \(操作說明\)](#)
- [掌握 Amazon CodeWhisperer 的藝術 - YouTube 播放清單](#)
- [AWS re:Invent 2020：可測試的基礎設施：在 AWS 上進行整合測試](#)
- [AWS Summit ANZ 2021 - 透過 CDK 和測試驅動的開發施行測試優先策略](#)
- [使用 AWS CDK 測試基礎設施即程式碼](#)

相關資源：

- [AWS 部署管道參考架構 - 應用程式](#)
- [AWS Kubernetes DevSecOps 管道](#)
- [使用 AWS CodeBuild 為 GitHub 中的 Node.js 應用程式執行單元測試](#)
- [使用 Serverspec 進行基礎設施程式碼的測試驅動開發](#)

相關服務：

- [Amazon Q Developer](#)
- [Amazon CodeGuru Reviewer](#)
- [AWS CodeBuild](#)
- [AWS CodePipeline](#)

OPS05-BP03 使用組態管理系統

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

靜態組態管理會在初始化資源時設定值，這些值預期會在資源的整個生命週期內保持一致。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 AWS 上，可以使用 [AWS Config](#) 持續監控 [跨帳號和區域](#) 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，可以使用 [AWS AppConfig](#) 在整個環境中進行設定、驗證、部署和監控。

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 採用安全的部署策略](#)
- [OPS06-BP04 自動化測試和復原](#)

相關文件：

- [AWS Control Tower](#)
- [AWS 登陸區域加速器](#)
- [AWS Config](#)
- [什麼是 AWS Config ?](#)
- [AWS AppConfig](#)
- [什麼是 AWS CloudFormation ?](#)
- [AWS 開發人員工具](#)

- [AWS CodeBuild](#)
- [AWS CodePipeline](#)
- [AWS CodeDeploy](#)

相關影片：

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

OPS05-BP04 使用建置和部署管理系統

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

在 AWS 中，可以使用 [AWS 開發人員工具](#) (例如 [AWS CodeBuild](#)、[AWS CodePipeline](#) 和 [AWS CodeDeploy](#)) 等服務來建置持續整合/持續部署 (CI/CD) 管道。

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

常見的反模式：

- 在開發系統中編譯程式碼之後，您將可執行檔複製到生產系統中，卻無法啟動。本機日誌檔案指出其因缺少相依性而失敗。
- 您在開發環境中使用新功能成功建置應用程式，並提供程式碼以進行品質保證 (QA)。它未通過 QA，因為缺少靜態資產。
- 週五，在經過一番努力之後，您成功在開發環境中手動建置應用程式，包括新編碼的功能。到了週一，您卻無法重複成功建置應用程式的步驟。
- 您執行為新版本建立的測試。然後，您會在下週設定測試環境，並執行所有現有的整合測試，接著執行效能測試。新的程式碼具有無法接受的效能影響，必須重新開發及測試。

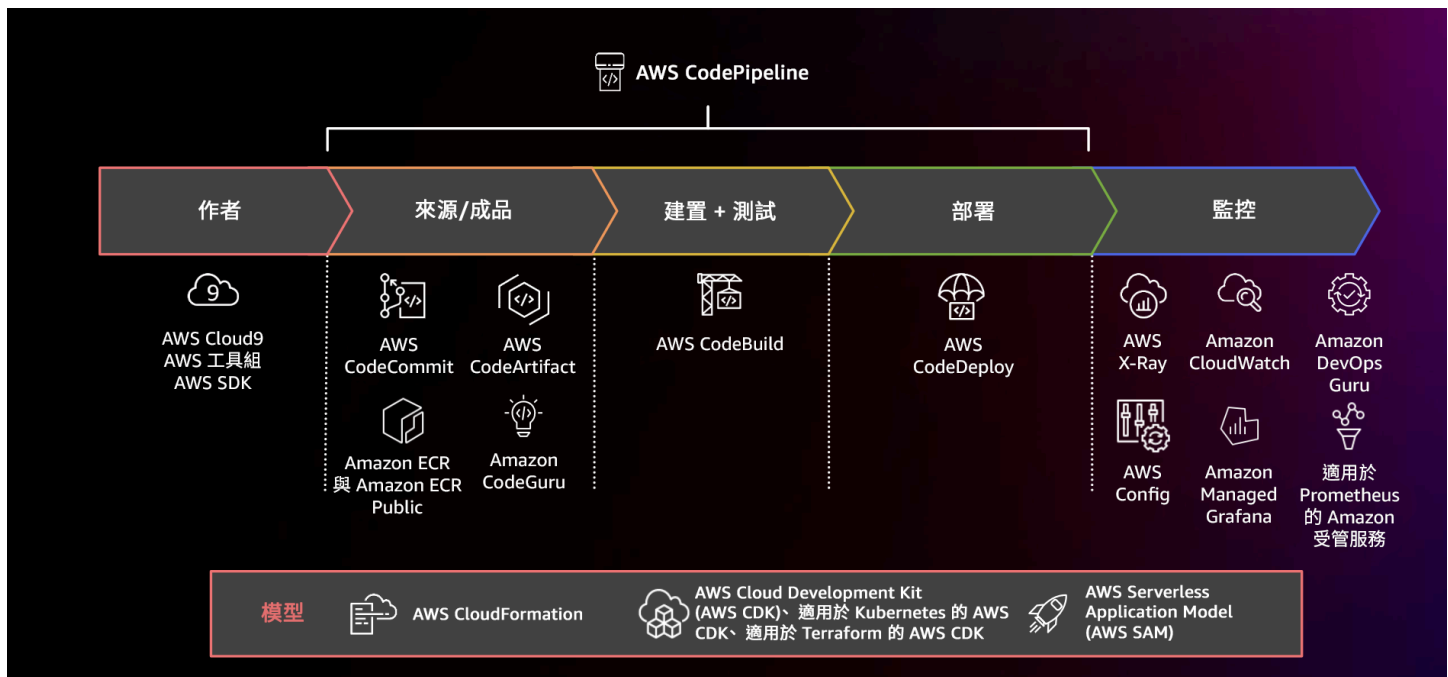
建立此最佳實務的優勢：透過提供用於管理建置和部署活動的機制，您可以減少執行重複性任務的工作量，讓團隊成員專注於高價值的創意任務，並減少手動程序導致的錯誤。

未建立此最佳實務時的曝險等級：中

實作指引

建置和部署管理系統可用來追蹤和實作變更、減少手動程序導致的錯誤，以及減少安全部署所需的工作量。從程式碼簽入到建置、測試、部署和驗證，完全自動化整合和部署管道。此舉可縮短前置時間、降低成本、促進增加變更頻率、減少工作量，並且增進協作。

實作步驟



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

1. 使用版本控制系統來儲存和管理資產 (例如文件、原始程式碼和二進位檔案)。
2. 使用 CodeBuild 可編譯原始碼、執行單元測試，並產生可立即部署的成品。
3. 將 CodeDeploy 用作一項部署服務，可自動將應用程式部署至 [Amazon EC2](#) 執行個體、內部部署執行個體、[無伺服器 AWS Lambda 函數](#) 或 [Amazon ECS](#)。
4. 監控您的部署。

資源

相關的最佳實務：

- [OPS06-BP04 自動化測試和復原](#)

相關文件：

- [AWS 開發人員工具](#)
- [什麼是 AWS CodeBuild？](#)
- [AWS CodeBuild](#)
- [什麼是 AWS CodeDeploy？](#)

相關影片：

- [AWS re:Invent 2022 - 適用於 DevOps on AWS 的 AWS Well-Architected 最佳實務](#)

OPS05-BP05 執行修補程式管理

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

修補程式和漏洞管理屬於您利益和風險管理活動的一部分。最好擁有不可變的基礎設施，並在已驗證的已知良好狀態下部署工作負載。如果這種方法不可行，剩下的方法就是進行修補。

[AWS Health](#) 是有關規劃的生命週期事件，以及其他影響 AWS 雲端 資源運作狀態且須採取行動的事件的權威資訊來源。您應得知即將進行的變更和更新。主要的規劃生命週期事件至少會提前六個月傳送。

[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](#) 的 [Amazon Machine Images](#) (AMI) 或具有 [Docker 映像管道](#) 的容器映像，同時 AWS Lambda 會為 [自訂執行時期和其他程式庫](#) 提供模式以移除漏洞。

應使用 [Amazon EC2 Image Builder](#) 管理適用於 Linux 或 Windows Server 映像的 [Amazon Machine Images](#) 的更新。可以使用 [Amazon Elastic Container Registry \(Amazon ECR\)](#) 搭配現有管道來管理 Amazon ECS 映像並管理 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。在開啟 AWS 時，可以使用 [AWS Systems Manager Patch Manager](#) 來自動化修補受管系統的流程，並使用 [Systems Manager 維護時段](#) 來排程活動。

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

常見的反模式：

- 您必須在兩小時內套用所有新的安全修補程式，結果導致應用程式與修補程式不相容而發生多次停機。
- 未修補的程式庫導致意外後果發生，因為有不明對象利用其中的漏洞來存取您的工作負載。
- 您自動修補開發人員環境，而未通知開發人員。您收到來自開發人員的多次投訴，表示其環境如預期停止運作。
- 您尚未在持續執行的執行個體上修補商用現成軟體。當軟體發生問題而您聯絡廠商時，他們會通知您不支援該版本，您必須修補至特定程度才能獲得協助。
- 您使用的加密軟體近期發佈了修補程式，使效能獲得大幅改善。未修補的系統因未修補仍存在效能問題。
- 收到發生零時差漏洞的通知時，需緊急修正並手動修補所有環境。
- 您不知道維護資源所需的重大行動，例如強制性版本更新，因為您未檢閱即將進行的規劃生命週期事件和其他資訊。您錯失規劃和執行的關鍵時間，導致團隊須緊急變更，且可能造成影響或非預期的停機時間。

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. 審核設定。
 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 EC2 Image Builder 建立映像管道](#)
- [建立容器映像管道](#)
- [AWS Systems Manager Patch Manager](#)
- [使用 Patch Manager](#)
- [使用修補程式合規報告](#)
- [AWS 開發人員工具](#)

相關影片：

- [AWS 上適用於無伺服器應用程式的 CI/CD](#)
- [設計時考量 Ops](#)

相關範例：

- [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 確保對營運準備度進行一致的審查](#) - 營運準備度檢查清單是在設計您的工作負載時實作設計標準的機制。
- [OPS11-BP01 建立持續改進程序](#) - 更新設計標準是持續改善的一部分。
- [OPS11-BP04 執行知識管理](#) - 在您的知識管理實務中，記錄和共用設計標準。

相關文件：

- [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 實作用於提高程式碼品質的實務

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審核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，而未進程式碼審核。變更會自動部署至生產環境，並導致中斷。
- 會開發一個新的應用程式，沒有進行任何單元、端到端或整合測試。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審核，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。程式碼品質最佳實務包括配對程式設計、程式碼審核以及 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、程式碼審核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 Retail 採用數個實務來改善程式碼品質。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審核，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、程式碼審核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](#) 可以使用機器學習針對 Java 和 Python 程式碼提供程式設計建議。

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 測試並驗證變更](#)
- [OPS05-BP06 共用設計標準](#)

相關文件：

- [採用測試驅動的開發方法](#)
- [使用 Amazon Q 加速您的軟體開發生命週期](#)
- [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 IDE 中使用 Amazon Q Developer 的終極速查表](#)
- [左移工作負載，利用 AI 建立測試](#)
- [Amazon Q 開發人員中心](#)
- [使用 Amazon CodeWhisperer 快速建置應用程式的 10 種方式](#)
- [使用 Amazon CodeWhisperer 超越程式碼覆蓋範圍](#)

- [使用 Amazon CodeWhisperer 進行提示詞工程的最佳實務](#)
- [敏捷式軟體指南](#)
- [CI/CD 管道是我的發行主管](#)
- [使用 Amazon CodeGuru Reviewer 自動進程式碼審核](#)
- [採用測試驅動的開發方法](#)
- [DevFactory 如何使用 Amazon CodeGuru 建置更佳的應用程式](#)
- [關於配對程式設計](#)
- [RENGA Inc. 使用 Amazon CodeGuru 自動進程式碼審核](#)
- [敏捷開發的藝術：測試驅動的開發](#)
- [程式碼審核為何重要 \(而且確實可節省時間！\)](#)

相關影片：

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

相關服務：

- [Amazon Q Developer](#)
- [Amazon CodeGuru Reviewer](#)
- [Amazon CodeGuru Profiler](#)

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。

1. 您的組織可透過建立登陸區域來執行此操作，該區域提供治理、控制、帳戶自動化、聯網、安全和營運可觀測性。使用多個環境來管理這些登陸區域功能。常見的範例是沙盒組織，可用於開發和測

試 [AWS Control Tower](#) 型登陸區域的變更，其中包括 [AWS IAM Identity Center](#) 和政策 (例如[服務控制政策 \(SCP\)](#))。這些元素全都可能對登陸區域內 AWS 帳戶 的存取和操作產生重大影響。

- 除了這些服務之外，您的團隊還可利用 AWS 和 AWS 合作夥伴發布的解決方案，或您組織內開發的自訂解決方案來擴展登陸區域功能。AWS 所發布解決方案的範例包括 [Customizations for AWS Control Tower \(CfCT\)](#) 和 [AWS Control Tower Account Factory for Terraform \(AFT\)](#)。
- 在付諸實際執行之前，您的組織會在過程中透過環境針對登陸區域套用相同的測試、推廣程式碼和政策變更原則。此策略為您的應用程式和工作負載團隊提供了穩定且安全的登陸區域環境。

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。
- 您升級一項重要的登陸區域功能，但變更卻有損您的團隊為新專案或現有工作負載供應帳戶的能力。
- 您將新的控制項套用至 AWS 帳戶，但變更卻影響工作負載團隊在其 AWS 帳戶 內部署變更的能力。

建立此最佳實務的優勢：當您部署多個環境時，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。對於像是登陸區域等複雜功能，它可大幅降低變更的風險、簡化改善程序，並降低對環境進行重大更新的風險。使用登陸區域的組織自然受益於其 AWS 環境中的多個帳戶，因為具有帳戶結構、治理、網路和安全組態。經過一段時間後，隨著組織成長，登陸區域可能會進一步保護和組織您的工作負載和資源。

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。

平台工程、聯網和安全營運等團隊通常會在擁有不同需求的組織層級管理各種功能。僅是將帳戶分隔
開來並不足以提供及維護實驗、開發和測試各自所需的環境。在這種情況下，可建立個別的 AWS
Organizations 執行個體。

資源

相關文件：

- [Instance Scheduler on AWS](#)
- [什麼是 AWS CloudFormation ?](#)
- [使用多個帳戶來組織您的 AWS 環境 - 多個組織 - 測試整體 AWS 環境的變更](#)
- [AWS Control Tower 指南](#)

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

常見的反模式：

- 週五，您完成了為功能分支編寫新程式碼。週一，執程式碼品質測試指令碼和每個單位測試指令碼之後，請為下一排程版本檢查程式碼。
- 系統會指派您編寫修正程式碼，以解決影響生產環境中大量客戶的重大問題。測試修正後，您遞交程式碼和電子郵件變更管理內容，以請求核准將其部署到生產環境中。
- 您以開發人員身分登入 AWS Management Console，以使用非標準方法和系統來建立新的開發環境。

建立此最佳實務的優勢：透過實作自動化建置和部署管理系統，您可以減少手動程序引起的錯誤，以及部署變更的工作量，協助您的團隊成員專注於提供商業價值。您在推廣至生產環境的同時，加快了交付速度。

未建立此最佳實務時的曝險等級：低

實作指引

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

資源

相關的最佳實務：

- [OPS05-BP03 使用組態管理系統](#)
- [OPS05-BP04 使用建置和部署管理系統](#)

相關文件：

- [什麼是 AWS CodeBuild？](#)
- [什麼是 AWS CodeDeploy？](#)

相關影片：

- [AWS re:Invent 2022 - 適用於 DevOps on AWS 的 AWS Well-Architected 最佳實務](#)

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。

工作負載的設計應包括如何部署、更新及操作。您希望實作符合減少缺陷和快速安全修復的工程實務。

最佳實務

- [OPS06-BP01 計畫變更失敗](#)
- [OPS06-BP02 測試部署](#)
- [OPS06-BP03 採用安全的部署策略](#)
- [OPS06-BP04 自動化測試和復原](#)

OPS06-BP01 計畫變更失敗

計畫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[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 測試並驗證變更](#)

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 採用安全的部署策略

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。

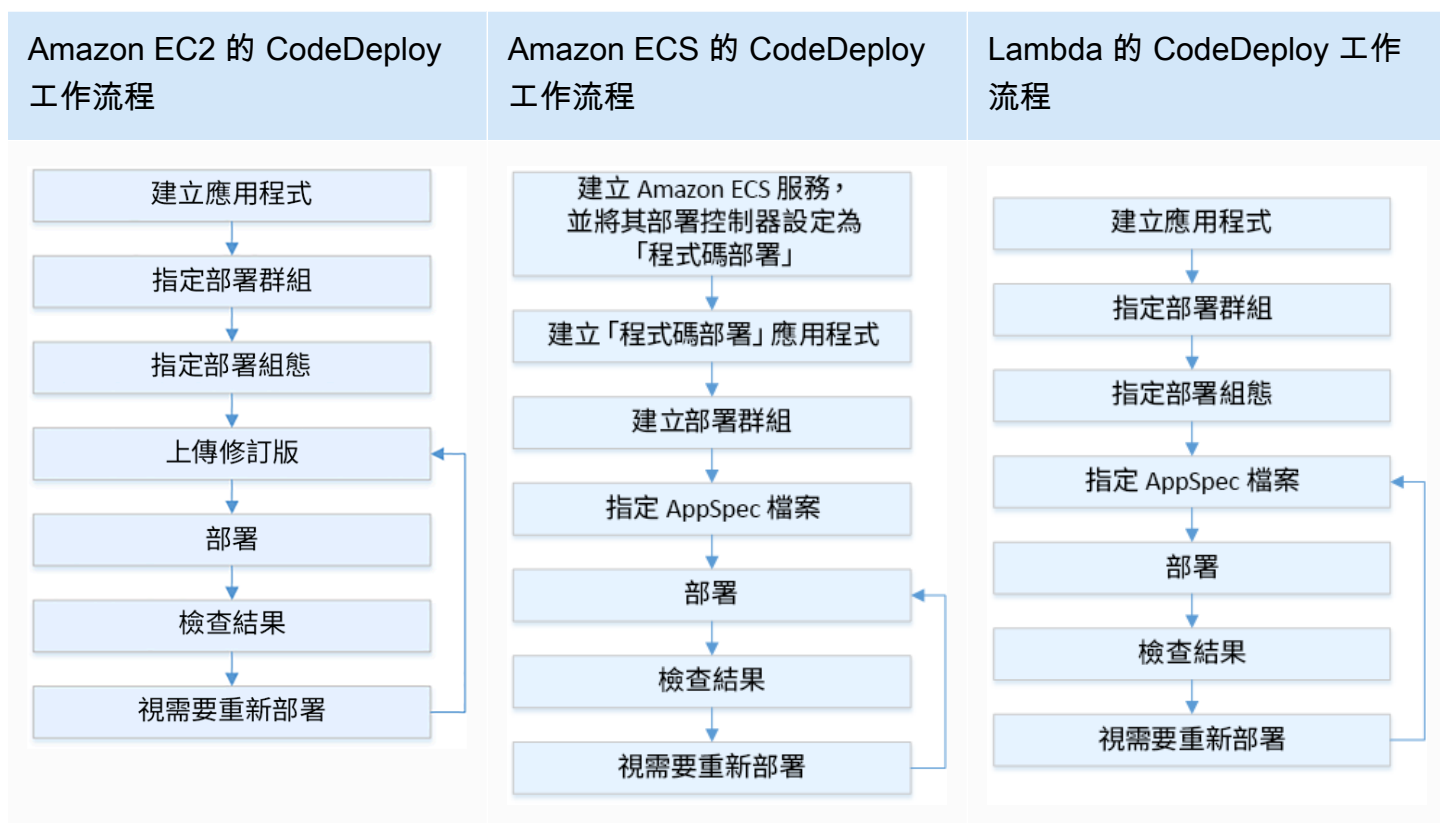
未建立此最佳實務時的曝險等級：中

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。為了盡可能提高部署成功率，請在端對端發行程序中實施安全控制，盡量減少部署錯誤，而目標則是實現零部署失敗。

客戶範例

AnyCompany Retail 的使命是實現最小至零停機時間的部署，這表示部署期間沒有對使用者產生任何可感知的負面影響。為此，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。



實作步驟

- 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。
- 使用自動化部署系統，例如 [AWS CodeDeploy](#)。AWS CodeDeploy [部署選項](#) 包括適用於 EC2/內部部署的就地部署，以及適用於 EC2/內部部署、AWS Lambda 和 Amazon ECS 的藍/綠部署 (請參閱上述工作流程圖)。
 - 在適用的情況下，[將 AWS CodeDeploy 與其他 AWS 服務整合](#)，或者[將 AWS CodeDeploy 與合作夥伴的產品和服務整合](#)。
- 對 [Amazon Aurora](#) 和 [Amazon RDS](#) 等資料庫使用藍/綠部署。
- 使用 Amazon CloudWatch、AWS CloudTrail 和 Amazon Simple Notification Service (Amazon SNS) 事件通知來[監控部署](#)。
- 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。
- 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 測試並驗證變更](#)
- [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 建置者資料中心 | 自動化安全、無人為介入的部署 | 生產部署](#)
- [AWS 建置者資料中心 | 我的 CI/CD 管道是我的發行主管 | 安全、自動的生產發行](#)
- [AWS 白皮書 | 在 AWS 上實行持續整合和持續交付 | 部署方法](#)
- [AWS CodeDeploy 使用者指南](#)
- [使用 AWS CodeDeploy 中的部署組態](#)
- [設定 API Gateway Canary 發行部署](#)
- [Amazon ECS 部署類型](#)
- [Amazon Aurora 和 Amazon RDS 中的全受管藍/綠部署](#)
- [使用 AWS Elastic Beanstalk 的藍/綠部署](#)

相關影片：

- [re:Invent 2020 | 無人為介入：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 AWS CodeDeploy 中嘗試使用藍/綠部署範例](#)
- [研討會 | 使用 AWS CDK 建置 Lambda Canary 部署的 CI/CD 管道](#)
- [研討會 | 利用 Amazon ECS 建置您的第一個 DevOps 藍/綠管道](#)
- [研討會 | 利用 Amazon EKS 建置您的第一個 DevOps 藍/綠管道](#)
- [研討會 | EKS GitOps 搭配 ArgoCD](#)
- [研討會 | AWS 上的 CI/CD 研討會](#)

- [針對容器型 Lambda 函數使用 AWS SAM 實作跨帳戶 CI/CD](#)

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 Lambda AWS 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](#)

營運準備度和變更管理

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。管理環境的變更流程。

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。這樣也有助於尋找需要您制定計畫以解決問題的任何領域。您將擁有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。使用一種機制來管理變更，此機制支援提供商業價值的並協助降低與變更相關的風險。

最佳實務

- [OPS07-BP01 確保人員能力](#)
- [OPS07-BP02 確保對營運準備度進行一致的審查](#)
- [OPS07-BP03 使用 Runbook 執执行程序](#)
- [OPS07-BP04 使用程序手冊來調查問題](#)
- [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 Retail 確保支援工作負載的團隊有適當的配備人員且訓練有素。他們有足夠的工程師可以支援待命輪替。人員會獲得工作負載建置基礎的軟體和平台的培訓，並且鼓勵他們考取認證。有足夠的人員讓員工可以休假，同時仍然支援工作負載和待命輪替。

實作步驟

1. 指派足夠的人員來操作和支援工作負載，包括待命職責、安全問題和生命週期事件，例如終止支援和憑證輪換任務。
2. 為您的人員提供組成您的工作負載的軟體和平台的培訓。
 - a. [AWS 培訓和認證](#)有一個關於 AWS 的課程庫。它們提供免費和付費的線上或面授課程。
 - b. [AWS 舉辦活動和網路研討會](#)，您可向 AWS 專家學習。
3. 定期執行下列工作：
 - 隨著操作條件和工作負載變更，評估團隊規模和技能。
 - 調整團隊大小和技能以符合操作要求。
 - 透過 AWS Health 驗證[處理規劃的生命週期事件](#)、意外的安全性和操作通知的能力和容量。

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

資源

相關的最佳實務：

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

相關文件：

- [AWS 活動和研討會](#)
- [AWS 培訓和認證](#)

OPS07-BP02 確保對營運準備度進行一致的審查

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

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

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

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

常見的反模式：

- 您啟動工作負載，但不知道自己是否能夠運行工作負載。
- 啟動工作負載的認證中未納入管控和安全性需求。
- 不會定期重新評估工作負載。
- 工作負載啟動，但不需設置必要的程序。
- 您可以在多個工作負載中看到重複出現的相同根本原因失敗。

建立此最佳實務的優勢：

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

若未建立此最佳實務的風險等級：高

實作指引

ORR 有兩個部分：程序和檢查清單。貴組織應採用 ORR 程序，並由執行主辦人支援此程序。至少，必須在工作負載啟動以全面供應前執行 ORR。在整個軟體開發生命週期執行 ORR，使其與最佳實務或新需求保持同步。ORR 檢查清單應包含組態項目、安全性和管控需求，以及來自貴組織的最佳實務。隨著時間的推移，可以使用諸如 [AWS Config](#)、[AWS Security Hub](#) 和 [AWS Control Tower Guardrails](#) 之類的服務，將 ORR 的最佳實務建置為防護機制，以自動偵測最佳實務。

客戶範例

在發生數個生產事件後，AnyCompany Retail 決定實作 ORR 程序。他們建立了一份檢查清單，其中由最佳實務、管控和合規需求，以及從中斷中汲取的經驗教訓所組成。在工作負載啟動前，新的工作負載會執行 ORR。每個工作負載每年都會使用一部分的最佳實務來執行 ORR，以便納入在 ORR 檢查清單中新增的最佳實務和需求。隨著時間的推移，AnyCompany Retail 使用 [AWS Config](#) 來偵測一些最佳實務，從而加快 ORR 流程。

實作步驟

要了解有關 ORR 的更多資訊，請閱讀 [《營運準備度審查 \(ORR\)》白皮書](#)。其中提供詳細的資訊，說明 ORR 程序的歷史、如何建立您專屬的 ORR 實務，以及如何制定 ORR 檢查清單。以下步驟是該文件的精簡版本。如需深入了解 ORR 是什麼，以及如何建立您專屬的 ORR，我們建議閱讀該白皮書。

1. 召集關鍵利益相關者，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。
 - 《營運準備度審查 (ORR)》白皮書的 [附錄 B：ORR 問題範例](#) 包含您可以開始使用的範例問題。
3. 將需求集中放在試算表中。
 - 您可以在 [AWS Well-Architected Tool](#) 中使用 [自訂聚焦](#) 來開發 ORR，並在您的帳戶和 AWS 組織中進行共用。
4. 找出要在其中執行 ORR 的一個工作負載。啟動前的工作負載或內部工作負載是理想的選擇。
5. 演練 ORR 檢查清單，並記下任何所探索的項目。如果採取緩解措施，那就可能接受探索。對於缺少緩解措施的任何探索，請將那些探索新增至項目的待辦清單中，然後在啟動前加以實作。
6. 隨著時間持續在 ORR 檢查清單中新增最佳實務和需求。

擁有 Enterprise Support 的支援客戶可向其技術客戶經理請求 [營運準備度審查研討會](#)。該研討會是一個交互式回溯會議，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 實務需要高層和利益相關者的支持。使用貴組織提供的各方意見，來建立和更新檢查清單。

資源

相關的最佳實務：

- [OPS01-BP03 評估治理要求](#) – ORR 檢查清單原本就很適合用來管控需求。
- [OPS01-BP04 評估合規要求](#) – ORR 檢查清單中有時會包含合規需求。有些時候，它們會是獨立的程序。
- [OPS03-BP07 資源團隊適當](#) – 團隊能力是 ORR 需求的絕佳候選項。
- [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劃。
- [OPS07-BP01 確保人員能力](#) – 若要支援工作負載，您必須具備所需的人員。
- [SEC01-BP03 識別和驗證控制目標](#) — 安全性控制目標是絕佳的 ORR 需求。
- [REL13-BP01 定義停機時間和資料遺失的復原目標](#) — 災難復原計劃是絕佳的 ORR 需求。
- [COST02-BP01 根據您的組織需求制定政策](#) — 成本管理政策適合包含在 ORR 檢查清單中。

相關文件：

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

相關影片：

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

相關範例：

- [營運準備度審查 \(ORR\) 聚焦範例](#)

相關服務：

- [AWS Config](#)

- [AWS Control Tower](#)
- [AWS Security Hub](#)
- [AWS Well-Architected Tool](#)

OPS07-BP03 使用 Runbook 執执行程序

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。它們至少應該包含 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：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 使用程序手冊來調查問題

程序手冊是用來調查事件的逐步指南。事件發生時，我們會使用程序手冊來調查、確認影響範圍和找出根本原因。程序手冊適用於各種情況，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。制定溝通計劃，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劃。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 Automation 來加速偵測和復原。

客戶範例

生產事故影響了 AnyCompany Retail。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。使用執行手冊，工程師重新啟動了服務，使 AnyCompany Retail 重新上線。

實作步驟

如果您沒有現有的文件儲存庫，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. 在文件儲存庫中發佈程序手冊，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。當您擁有多本程序手冊之後，請使用 AWS Systems Manager Automation 等工具開始進行自動化，讓自動化和程序手冊保持同步。

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 流程和程序已識別擁有者](#)
- [OPS07-BP03 使用執行手冊執执行程序](#)

- [OPS10-BP01 使用程序進行事件、事故和問題管理](#)
- [OPS10-BP02 每個提醒建立一個程序](#)
- [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：執行手冊、事件報告和事件回應的 DIY 指南 \(SEC318-R1\)](#)
- [AWS Systems Manager Incident Manager - AWS 虛擬研討會](#)
- [將指令碼整合到 AWS Systems Manager](#)

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 Automation](#)
- [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畫

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級：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劃。設定適當的支援計劃以滿足服務層級需求。對於 AWS 客戶而言，這意味著在任何有生產工作負載的帳戶上啟動 AWS Business Support 或更高等級。定期與支援供應商聯繫，取得關於支援優惠、程序和聯絡人的更新。記錄如何要求軟體和服務供應商的支援，包括如何在中斷發生時加以呈報。實作相關機制以保有最新的支援聯絡資料。

客戶範例

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

實作步驟

1. 與組織中的利益相關者合作，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劃。
3. 針對商業軟體和服務，與供應商共同建立支援計劃。
 - a. 為所有生產帳戶訂閱 AWS Business Support 或更高等級可獲得 AWS 支援 更快的回應時間，極力建議這麼做。如果您沒有付費支援，則必須有處理問題的行動計劃，而這需要 AWS 支援 的協助。AWS 支援 提供了多種工具和技術、人員及方案，旨在主動協助您最佳化效能、降低成本和加快創新速度。此外，AWS Business Support 提供額外的權益，包括透過 API 存取 AWS

Trusted Advisor 和 AWS Health，以便與您的系統進程式設計整合，以及其他存取方法，例如 AWS Management Console 和 Amazon EventBridge 管道。

4. 在您的知識管理工具中記錄支援計劃。納入如何要求支援、在提出支援案例時應通知誰，以及在事件發生時如何加以呈報等資訊。任何人在得知支援程序或聯絡資料有所變更時，都可以利用 Wiki 這項機制對文件進行必要的更新。

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

資源

相關的最佳實務：

- [OPS02-BP02 流程和程序已識別擁有者](#)

相關文件：

- [AWS 支援 計劃](#)

相關服務：

- [AWS 商業支援](#)
- [AWS 企業支援](#)

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 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呈現營運活動的業務、工作負載和營運層級檢視。AWS 透過 AWS X-Ray、CloudWatch、CloudTrail 以及 VPC Flow Logs 等日誌記錄功能來提供工作負載洞見，以識別工作負載問題，支援根本原因分析和修復。

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。

主題

- [利用工作負載可觀測性](#)
- [了解運作狀態](#)
- [回應事件](#)

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。

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。

AWS 提供特定的工具，如 [Amazon CloudWatch](#) (用於監控和日誌記錄) 和 [AWS X-Ray](#) (用於分散式追蹤)。這些服務可與各種不同的 AWS 資源輕鬆整合，進而有效率地收集資料、根據預先定義的閾值設定提醒，以及在儀表板上呈現資料以便進行解讀。利用這些洞察就能讓您做出明智的資料驅動決策，以滿足您的營運目標。

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 建立可執行的提醒](#)
- [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 服務可用於偵測異常，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，並隨著時間比較它們。
 - b. 使用 [中的百分位數 CloudWatch](#) 來清楚了解指標分佈，這有助於定義SLAs和理解異常值。
 - c. 設定 [CloudWatch 異常偵測](#) 以識別異常模式，而不依賴靜態閾值。
 - 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 建立可執行的提醒

及時偵測並回應您的應用程式行為中的偏差至關重要。尤其重要的是要了解基於關鍵績效指標 (KPI) 的結果何時處於危險之中，或者何時出現意外異常。以 KPI 為基礎的提醒可確保您收到的訊號直接與業務或營運影響產生關係。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。

預期成果：接收及時、相關且可行的提醒，以便快速識別和緩解潛在問題，尤其是在 KPI 結果面臨風險時。

常見的反模式：

- 設定太多非嚴重性提醒會導致提醒疲勞。
- 不會根據 KPI 來排定提醒的優先順序，因此難以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

若要建立有效的提醒機制，使用指標、日誌和追蹤資料至關重要，其會在基於 KPI 的結果出現風險或偵測到異常時進行標記。

實作步驟

1. 確定關鍵績效指標 (KPI)：確定應用程式的 KPI。提醒應與這些關鍵績效指標相關聯，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，這可協助您僅針對真正的異常產生提醒。
- 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 DevOps Guru 整合：
 - a. 利用 [Amazon DevOps Guru](#) 的機器學習功能，偵測現有資料的操作異常情況。
 - b. 導覽至 DevOps Guru 中的[通知設定](#)，以設定異常提醒。
- 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. 使用 [Amazon EventBridge 規則監控 AWS Health 事件](#)，或以程式設計方式與 AWS Health API 整合，以便在您收到 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. 與提醒工具整合：整合諸如 [Ops Genie](#) 和 [PagerDuty](#) 等工具。
- 7. 採用聊天應用程式中的 Amazon Q Developer：整合[聊天應用程式中的 Amazon Q Developer](#)，以便將警示轉送至 Amazon Chime、Microsoft Teams 和 Slack。
- 8. 基於日誌的提醒：使用 CloudWatch 中的 [日誌指標篩選器](#)，根據特定的日誌事件建立警示。
- 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](#)
- [DevOps Guru 通知](#)
- [X-ray Insights 通知](#)
- [透過互動式 ChatOps 監控和操作您的 AWS 資源並進行疑難排解](#)
- [Amazon CloudWatch 整合指南 | PagerDuty](#)
- [整合 Opsgenie 與 Amazon CloudWatch](#)

相關影片：

- [在 Amazon CloudWatch 中建立複合警示](#)
- [聊天應用程式中的 Amazon Q Developer 概觀](#)
- [聊天應用程式中的 Amazon Q Developer 中的 AWS On Air ft. 可變命令](#)

相關範例：

- [使用 Amazon CloudWatch 在雲端進行警示、事件管理和修復](#)
- [教學課程：建立將通知傳送至聊天應用程式中的 Amazon Q Developer 的 Amazon EventBridge 規則](#)
- [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：

- 立即掌握關鍵系統指標和 KPI。
- 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

專為業務 KPI 量身打造的儀表板，可與更廣泛的利益相關者進行互動。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，突出顯示業務 KPI 的儀表板往往更具可操作性。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，請確保技術指標和業務 KPI 之間保持平衡。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 儀表板是 CloudWatch 主控台中可自訂的首頁，可讓您在單一檢視中監控資源，甚至是分散在不同的 AWS 區域和帳戶中的那些資源。

實作步驟

1. 建立基本儀表板：[在 CloudWatch 中建立新儀表板](#)，並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：利用 [CloudWatch Log Insights](#) 從日誌中導出可操作的指標，並在儀表板上顯示這些洞見。
6. 設定警示：將 [CloudWatch 警示](#) 整合到儀表板中，以便快速查看違反其閾值的任何指標。
7. 使用 Contributor Insights：整合 [CloudWatch Contributor Insights](#) 以分析高基數欄位，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：AWS Health 是 AWS 雲端 資源運作狀態的權威資訊來源。使用現成的 [AWS Health Dashboard](#)，或使用您自己的儀表板和工具中的 AWS Health 資料，以便擁有正確的資訊來做出明智的決策。
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 建立可執行的提醒](#)

相關文件：

- [建置用於檢視營運狀況的儀表板](#)
- [使用 Amazon CloudWatch 儀表板](#)

相關影片：

- [建立跨帳戶和跨區域 CloudWatch 儀表板](#)
- [AWS re:Invent 2021 - 透過 AWS 雲端 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 CloudWatch 監控應用程式](#)
- [AWS Health 事件智慧儀表板和洞見](#)
- [使用 Amazon Managed Grafana 視覺化 AWS Health 事件](#)

了解運作狀態

定義、擷取和分析營運指標，掌握營運團隊的活動，以便採取適當行動。

您的組織應該要能輕鬆了解營運狀況。您會希望訂出營運團隊的業務目標，確定反映這些目標的關鍵績效指標。然後根據營運成果開發並使用指標，以獲得有用的洞察。您應使用這些指標來實作涵蓋業務和技術觀點的儀表板和報告，進而協助主管和利害關係人做出明智的決策。

AWS 可輕鬆彙集和分析您的營運日誌，如此一來，您便能產生指標、了解自己的營運狀態並從長期的營運中取得洞察。

最佳實務

- [OPS09-BP01 使用指標衡量營運目標與 KPI](#)
- [OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況](#)
- [OPS09-BP03 檢閱操作指標並排定改善優先順序](#)

OPS09-BP01 使用指標衡量營運目標與 KPI

從您的組織取得定義營運成功的目標和 KPI，並決定反映這些目標的指標。設定基準做為參考點，並定期重新評估。制定機制以便從團隊收集這些指標以進行評估。[DevOps Research and Assessment \(DORA\)](#) 指標提供了衡量軟體交付的 DevOps 實務進度的常用方法。

預期成果：

- 組織會發布並分享營運團隊的目標和 KPI。

- 您負責建立反映這些 KPI 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
 - 處理問題所花的時間，無論是否有標準作業程序 (SOP)
 - 從失敗的程式碼推送復原所花的時間長度
 - 通話音量

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害關係人一起確定服務未來的整體目標。確定各個不同營運團隊應負責的任務，以及能夠應對哪些挑戰。使用這些來集思廣益，找出能夠反映這些營運目標的關鍵績效指標 (KPI)。這些目標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，以及成本效率。

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

資源

相關文件：

- [QuickSight - 使用 KPI](#)
- [Amazon CloudWatch - 使用指標](#)
- [建置儀表板](#)
- [如何使用 KPI 儀表板追蹤成本最佳化 KPI](#)
- [AWS DevOps 指引](#)

相關範例：

- [使用原生 AWS 監控和可觀測性工具來監控軟體交付的效能](#)
- [利用 DORA 指標在部署速度與穩定性之間取得平衡](#)
- [金融服務產業中的 MLOps 營運指標範例](#)
- [如何使用 KPI 儀表板追蹤成本最佳化 KPI](#)

OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況

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

預期成果：

- 主管對團隊處理的通話量類型和正在進行的工作 (例如部署) 可以一目瞭然。
- 發生影響擴及正常營運的情況時，利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最能反映目標和 KPI 的指標和報告，以及這些資訊在推動變革方面的影響力。花時間進行這些活動，以在團隊內部和團隊之間提高營運的重要性。

使用 [AWS Health](#) 搭配您擁有的儀表板，或將 AWS Health 事件整合到其中，讓您的團隊能夠找出應用程式問題與 AWS 服務狀態之間的相互關聯。

資源

相關的最佳實務：

- [OPS09-BP01 使用指標衡量營運目標與 KPI](#)

相關文件：

- [測量進度](#)
- [建置用於檢視營運狀況的儀表板](#)

相關範例：

- [資料操作](#)
- [如何使用 KPI 儀表板追蹤成本最佳化 KPI](#)
- [大規模雲端遷移關鍵績效指標 \(KPI\) 的重要性](#)

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](#)

回應事件

您應可以預測營運事件，不論是計劃 (如銷售促銷、部署和故障測試) 或非計劃 (如使用率突增和元件故障) 中的事件。回應提醒時，應使用現有的執行手冊和程序手冊以實現一致的結果。定義的提醒應由負責回應和向上呈報的角色或團隊擁有。您還將希望了解系統元件的業務影響，並在需要時使用它來確定工作目標。您應在事件發生後執行根本原因分析 (RCA)，然後防止再次發生失敗或文件因應措施。

AWS 提供的工具可以程式碼支援您工作負載及營運的各個方面，進而簡化您的事件回應。這些工具可讓您編寫營運事件回應的指令碼，並進行初始化以回應監控資料。

在 AWS 中，您可透過將失敗的元件取代為已知良好的版本來縮短復原時間，而不是嘗試進行修復。然後，您可以對失敗的頻外資源執行分析。

最佳實務

- [OPS10-BP01 使用程序進行事件、事故和問題管理](#)

- [OPS10-BP02 每個提醒建立一個程序](#)
- [OPS10-BP03 根據業務影響排定操作事件的優先順序](#)
- [OPS10-BP04 定義升級路徑](#)
- [OPS10-BP05 為影響服務的事件定義客戶溝通計劃](#)
- [OPS10-BP06 透過儀表板傳達狀態](#)
- [OPS10-BP07 自動化對事件的回應](#)

OPS10-BP01 使用程序進行事件、事故和問題管理

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、角色或 AWS 服務所採取的監控動作會記錄為 [AWS CloudTrail](#) 中的事件。
- 使用 [Amazon EventBridge](#) 即時對應用程式中的操作變更進行回應。
- 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。
- 定期審核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 [AWS X-Ray](#) 等服務，進行更深入的分析 and 疑難排解。
- 考慮使用 [AWS Managed Services \(AMS\)](#) 來加強事件管理，利用其主動、預防性和偵測功能。AMS 透過監控、事故偵測和回應以及安全管理等服務擴展了營運支援。
- 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。
- 將事件管理與聊天應用程式中的 [Amazon Q Developer](#) 這類工具整合，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 [事件回應計劃](#)。

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劃。
- 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。
- 運用諸如 [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。
- 整合根本原因分析 (RCA) 技術，以調查並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. 聯絡 AWS 支援：

- 使用 AWS 支援資源，例如 [AWS Trusted Advisor](#)，取得主動式指引和最佳化建議。
- 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 執行情序](#)
- [OPS07-BP04 使用程序手冊來調查問題](#)
- [OPS08-BP01 分析工作負載指標](#)
- [OPS11-BP02 執行事後分析](#)

相關文件：

- [AWS 安全事件回應指南](#)
- [AWS 事件偵測與回應](#)
- [AWS Cloud Adoption Framework：營運角度 - 事件與問題管理](#)
- [DevOps 和 SRE 時代的事件管理](#)
- [PagerDuty - 什麼是事件管理？](#)

相關影片：

- [來自 AWS 的重要事件回應秘訣](#)
- [AWS re:Invent 2022 - Amazon 建置者資料中心：25 年 Amazon 卓越營運](#)
- [AWS re:Invent 2022 - AWS 事故偵測與回應 \(SUP201\)](#)
- [介紹 AWS Systems Manager 中的 Incident Manager](#)

相關範例：

- [AWS 主動服務 – 事件管理研討會](#)
- [如何使用 PagerDuty 和 AWS Systems Manager Incident Manager 來自動化事故回應](#)
- [使用 AWS Systems Manager Incident Manager 中的隨時待命的時間表與事故回應人員互動](#)
- [改善 AWS Systems Manager Incident Manager 中事故處理期間的可見性和協作](#)
- [AMS 中的事故報告和服務請求](#)

相關服務：

- [Amazon EventBridge](#)

OPS10-BP02 每個提醒建立一個程序

為系統中的每個提醒建立清晰明確的程序，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劃。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。提醒會連結至最新的知識庫，以便任何操作員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)。
- 減少平均調查時間 (MTTI)，有助於降低 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。

例如，您已有既定流程來處理重要帳戶的 AWS Health 事件，包括應用程式警示、營運問題及規劃的生命週期事件 (例如，在叢集自動更新之前更新 Amazon EKS 版本)，而且您為團隊提供主動監控、溝通和回應這些事件的能力。這些動作有助於防止 AWS 端變更造成的服務中斷，或是在發生非預期的問題時更快緩解。

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劃；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。它的設計目的是透過自動建立事件來回應 [Amazon CloudWatch](#) 或 [Amazon EventBridge](#) 中的特定事件，迅速回應營運問題。自動或手動建立事件時，Incident Manager 會集中管理事件，組織相關的 AWS 資源資訊，並啟動預先定義的回應計劃。這包括執行 Systems Manager Automation 執行手冊以立即採取行動，以及在 OpsCenter 中建立父作業工作項目以追蹤相關任務和分析。此簡化的流程可加速並協調整個 AWS 環境中的事件回應。



1. 使用複合警示：在 CloudWatch 中建立[複合警示](#)，將相關警示分組，從而降低噪音並允許更有意義的回應。

2. 利用 [AWS Health](#) 隨時掌握新知：AWS Health 是 AWS 雲端 資源運作狀態的權威資訊來源。使用 AWS Health 視覺化並接收有關任何目前服務事件和近期變更的通知 (例如規劃的生命週期事件)，如此您就能採取行動來緩解衝擊。
 - a. 透過 [AWS 使用者通知](#) 建立符合用途的 [AWS Health 事件通知](#)，以利用電子郵件和聊天管道傳送，並透過 [Amazon EventBridge](#) 或 [AWS Health API](#) 以程式設計方式與您的監控和警示工具整合。
 - b. 透過 [Amazon EventBridge](#) 或 [AWS Health API](#) 整合變更管理或您可能已在使用的 ITSM 工具 (如 [Jira](#) 或 [ServiceNow](#))，以規劃並追蹤需要採取行動的運作狀態事件進度。
 - c. 如果您使用 AWS Organizations，請啟用 [AWS Health 的組織檢視](#)，以彙總帳戶之間的 AWS Health 事件。
3. 整合 Amazon CloudWatch 警示與 Incident Manager 設定 CloudWatch 警示，以便在 [AWS Systems Manager Incident Manager](#) 中自動建立事件。
4. 整合 Amazon EventBridge 與 Incident Manager：建立 [EventBridge 規則](#) 以回應事件並使用定義的回應計劃建立事件。
5. 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 [回應計劃](#)。
 - 透過與 Incident Manager 中回應計劃相連的 [聊天應用程式中的 Amazon Q Developer](#) 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 建立可執行的提醒](#)

相關文件：

- [AWS Cloud Adoption Framework：營運角度 - 事件與問題管理](#)
- [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 AWS 的重要事件回應秘訣](#)
- [re:Invent 2023 | 使用 AWS Health 大規模管理資源生命週期事件](#)

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，針對事件需要回應的速度定義緊急程度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

緊迫性和影響	高	中	低
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 建立可執行的提醒](#)
- [OPS09-BP01 使用指標衡量營運目標與 KPI](#)

相關文件：

- [Atlassian - 了解事件嚴重性層級](#)
- [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畫和待命排程，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 建立可執行的提醒](#)
- [OPS10-BP02 每個提醒建立一個程序](#)
- [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 為影響服務的事件定義客戶溝通計畫

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。明確定義的溝通計畫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劃涉及多個方面，從選擇正確的渠道到精心製作消息和基調。該計劃應具有適應性、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. 確定溝通渠道：選取諸如工作場所聊天、電子郵件、簡訊、社交媒體、應用程式內通知和狀態頁面等渠道。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、行動推送通知和文字訊息來提醒訂閱用戶。
 - 透過公開共用 Amazon CloudWatch 儀表板，使用儀表板溝通狀態。
 - 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. 協調內部溝通：使用如聊天應用程式中的 Amazon Q Developer 等工具實作內部通訊協定，以進行團隊協調和溝通。使用 CloudWatch 儀表板來溝通狀態。
5. 使用專用工具和服務協調溝通：
- 搭配使用 AWS Systems Manager Incident Manager 和聊天應用程式中的 Amazon Q Developer 即可設定專屬的聊天頻道，以便在事件發生時進行即時內部溝通和協調。
 - 在事件發生期間，透過 Amazon Pinpoint、Amazon SNS 或第三方工具 (例如社交媒體平台)，使用 AWS Systems Manager Incident Manager 執行手冊將客戶通知自動化。
 - 在執行手冊中合併核准工作流程，以便在傳送前選擇性地審核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劃。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劃。

實作計劃的工作量：高

資源

相關的最佳實務：

- [OPS07-BP03 使用 Runbook 執执行程序](#)
- [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 - 事件溝通指南](#)

相關影片：

- [Atlassian - 建立您自己的事件溝通計劃：事件範本](#)

相關範例：

- [AWS Health 儀表板](#)

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準確性。
- 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 QuickSight 等工具可協助您建立用於系統監控和商業智慧的互動式即時儀表板。

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，使用 [Amazon CloudWatch 儀表板](#) 進行系統監控，以及使用 [Amazon QuickSight](#) 獲得互動式商業情報。[AWS Health](#) 在 [AWS Health Dashboard](#) 中提供了立即可用的體驗，您也可以使用 Amazon EventBridge 中的運作狀態事件或透過 AWS Health API 來擴增您的儀表板。
3. 設計高效儀表板：
 - 設計儀表板以清楚呈現相關指標和 KPI，確保其易於理解和操作。
 - 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](#) 彙總和顯示來自各種 AWS 服務的指標，並[查詢其他資料來源的指標](#)，建立系統運作狀態和商業指標的統一檢視。
 - 使用 [CloudWatch Logs Insights](#) 等功能，查詢並視覺化來自不同應用程式和服務的日誌資料。
 - 使用 AWS Health 事件透過 [AWS Health API](#) 或 [Amazon EventBridge 上的 AWS Health 事件](#)，隨時掌握 AWS 服務的操作狀態和已確認的操作問題。
5. 提供自助服務存取：
 - 與相關利益相關者共用 CloudWatch 儀表板，以使用[儀表板共用功能](#)進行自助式資訊存取。
 - 確保儀表板易於存取，並提供即時、最新的資訊。
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。
 - 定期審核儀表板，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](#)
- [共享 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 建立可執行的提醒](#)
- [OPS10-BP02 每個提醒建立一個程序](#)

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

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。

分享在團隊中學到的經驗教訓。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 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，並將相關聯的中繼資料儲存在 AWS Glue Data Catalog 中。Amazon Athena，透過與 AWS Glue 的原生整合，可用來分析日誌資料，並使用標準 SQL 進行查詢。您可以使用 Amazon QuickSight 這類商業智慧工具來視覺化、探索和分析資料。探索可能推動改善的感興趣趨勢和事件。

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，以及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。

主題

- [學習、分享和改進](#)

學習、分享和改進

您必須定期留有時間，以進行營運活動分析、失敗情境分析、試驗及作出改善。當事情失敗時，您將要確保您的團隊以及整個工程師社群都能從這些失敗中獲得經驗。您應分析失敗以識別獲得的經驗並規劃改善。您希望與其他團隊定期審查獲得的經驗，以驗證您的洞見。

最佳實務

- [OPS11-BP01 建立持續改進程序](#)
- [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 記錄和分享獲得的經驗](#)
- [OPS11-BP09 分配時間進行改善](#)

OPS11-BP01 建立持續改進程序

根據內部和外部架構最佳實務評估您的工作負載。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：

- 您的工作負載在架構最佳實務中保持最新狀態。
- 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 特定最佳實務的已記載架構標準。
 - 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 AWS Well-Architected Tool 來建立內部最佳實務的自訂聚焦，並執行架構審查。
 - c. 請聯絡您的 AWS 解決方案架構師或技術客戶經理，在引導下執行工作負載的 Well-Architected Framework 審查。
2. 在您的軟體開發程序中，為在審查期間找出的改進機會制定優先順序。

實作計劃的工作量：低。可以使用 AWS Well-Architected Framework 進行每年的架構審查。

資源

相關的最佳實務：

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

相關文件：

- [AWS Well-Architected Tool - 自訂聚焦](#)
- [AWS Well-Architected 白皮書 - 審查程序](#)
- [使用自訂聚焦和 AWS Well-Architected Tool 來自訂 Well-Architected 審查](#)
- [在您的組織中實作 AWS Well-Architected 自訂聚焦生命週期](#)

相關影片：

- [AWS re:Invent 2023 - 擴展組織的 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. 回應計畫或呈報計畫是否有更新，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 建立持續改進程序](#)
- [OPS 4 - 實作可觀測性](#)

相關文件：

- [在 Incident Manager 中執行事件後分析](#)
- [營運準備情況審核](#)

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 評估外部客戶需求](#)：回饋迴圈是一種機制，可收集外部客戶的需求。
- [OPS01-BP02 評估內部客戶需求](#)：內部利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 – PDICS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内容管理系統開始。跨組織取得利益相關者的同意。
 - a. 如果您沒有現有内容管理系統，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内容管理系統中。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 記錄和分享獲得的經驗](#) - 知識管理可促進所學習課程的資訊共用。

相關文件：

- [Atlassian - 知識管理](#)

相關範例：

- [DokuWiki](#)
- [Gollum](#)
- [MediaWiki](#)
- [Wiki.js](#)

OPS11-BP05 定義改進驅動因素

確定改進驅動因素，以幫助您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。

- 可以從整個資產中收集詳細資料，這會導致較高的 Amazon CloudWatch 和 AWS CloudTrail 活動與成本。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。
 - [AWS 最新消息](#)
 - 不可接受的問題：在評估改進機會時，評估不可接受的問題、錯誤和漏洞。追蹤合適的選項，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 組織優先事項](#)

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

相關文件：

- [Amazon Athena](#)
- [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 - 透過 支援 \(SUP310\) 提高營運效率和彈性](#)

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 記錄和分享獲得的經驗

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) 來定義權限，允許對您希望在帳戶內及帳戶間分享的資源的受控存取。

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。
- 您已找到一個方法，可大幅降低其中一個微型服務所需的 CPU 使用率。您不知道是否有任何其他團隊可以利用此技術。

建立此最佳實務的優勢：分享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。
 - 利用 [AWS re:Post Private](#) 做為知識服務，讓組織內的協作和知識共享更順暢。

資源

相關的最佳實務：

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

相關文件：

- [利用 AWS re:Post Private 增進協作並安全地共享雲端知識](#)
- [利用文件即程式碼解決方案減少專案延遲](#)

相關影片：

- [AWS re:Invent 2023 - 利用 AWS re:Post Private 在您的公司內協作並與 AWS 合作](#)
- [支援 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](#)

結論

卓越營運需要堅持不懈的努力方能達成。

透過共同的目標來讓組織做好準備，邁向成功。確保每個人都了解自己在達成業務成果方面的角色，以及他們如何影響他人成功的能力。為團隊成員提供支援，讓他們能夠協助達成業務成果。

每個營運事件和失敗均應被視為改善您架構營運的機會。透過了解工作負載的需求、預先定義例行活動的 Runbook，以及引導問題解決方案的 Playbook、使用中的操作作為程式碼功能 AWS，以及維持情境感知，您的操作將能更好地做好準備，並在事件發生時更有效地回應。

在優先事項變更時，透過專注於以優先事項為基礎的增量改善，以及從事件回應和追溯性分析中獲得的經驗，您將透過加強活動的效率和效果，來協助業務成功。

AWS 致力於協助您建置和操作可最大化效率的架構，同時建置高回應性和適應性部署。若要提高工作負載的卓越營運，您應該使用本白皮書中所述的最佳實務。

貢獻者

- Rich Boyd , Amazon Web Services Well-Architected 卓越營運支柱主管
- Jon Steele , Amazon Web Services Well-Architected 首席解決方案架構師
- Ryan King , Amazon Web Services 資深技術計畫經理
- Chris Kunselman , Amazon Web Services 諮詢顧問
- Peter Mullen , Amazon Web Services 諮詢顧問
- Brian Quinn , Amazon Web Services 資深諮詢顧問
- David Stanley , Amazon Web Services 雲端營運模式主管
- 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
- Fatih (Ben) Mergen , Amazon Web Services 資深成本首席解決方案架構師

深入閱讀

如需其他指引，請參閱下列資源：

- [AWS Well-Architected 架構](#)
- [AWS 架構中心](#)

文件修訂

若要收到此白皮書更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
已更新最佳實務指引	最佳實務已更新為以下領域中的新指引：OPS 2、OPS 5、OPS 9 和 OPS 10。指引包括有關 AWS 服務和生成式 AI 的新建議。	2024 年 11 月 6 日
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。對 OPS 1、OPS 2 和 OPS 3 中的內容進行多次整合。OPS 10 中的風險評等變更。	2024 年 6 月 27 日
主要內容更新與整合	有多個最佳實務領域的內容已更新並整合。有兩個最佳實務領域 (OPS 4 和 OPS 8) 已重新撰寫，納入了新的內容和重點。 下列領域的最佳實務已更新並整合：營運設計、緩解部署風險和了解運作狀態。最佳實務領域 OPS 04 已更新至實作可觀測性。最佳實務領域 OPS 08 已更新至利用工作負載可觀測性。	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。	2023 年 4 月 10 日

白皮書已更新	最佳實務更新了新的實作指引。	2022 年 12 月 15 日
白皮書已更新	已擴充最佳實務並新增了改善計畫。	2022 年 10 月 20 日
次要更新	小幅度編輯更新。	2022 年 8 月 8 日
白皮書已更新	更新以反映新的 AWS 服務和功能以及最新的最佳實務。	2022 年 2 月 2 日
新框架的更新	更新以反映新的 AWS 服務和功能以及最新的最佳實務。	2020 年 7 月 8 日
白皮書已更新	更新以反映新的 AWS 服務和功能以及更新的參考。	2018 年 7 月 1 日
初次出版	卓越營運支柱 – AWS Well-Architected Framework 已發布。	2017 年 11 月 1 日

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 產品和實務，這些產品和實務可能隨時變更，恕不另行通知，且（c）不會從 AWS 及其附屬公司、供應商或授權方提供「原樣」的任何承諾 AWS 或保證，而沒有任何明示或暗示的保證、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。

© 2023 Amazon Web Services, Inc. 或其附屬公司。保留所有權利。

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](#)。