AWS Well-Architected Framework

卓越營運支柱

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

卓越營運支柱: AWS Well-Architected Framework

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

摘要與簡介 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 4
OPS01-BP01 評估客戶需求 4
OPS01-BP02 評估內部客戶需求 5
OPS01-BP03 評估管控要求 7
OPS01-BP04 評估合規要求 9
OPS01-BP05 評估威脅態勢 11
OPS01-BP06 在管理效益和風險的同時評估權衡 13
操作模式 16
操作模式 2 x 2 表示法 16
關係和擁有權 24
組織文化 40
OPS03-BP01 提供高層的支持 40
OPS03-BP02 授權團隊成員在成果有風險時採取動作 43
OPS03-BP03 鼓勵向上呈報 45
OPS03-BP04 溝通需及時、清楚且可行 48
OPS03-BP05 鼓勵進行試驗 51
OPS03-BP06 團隊成員受到鼓勵來維持和培養自己的技能集 54
OPS03-BP07 適當地為團隊提供資源 56
準備 59
實作可觀測性 59
OPS04-BP01 識別關鍵績效指標 60
OPS04-BP02 實作應用程式遙測 61
OPS04-BP03 實作使用者體驗遙測 64
OPS04-BP04 實作相依性遙測 66
OPS04-BP05 實作分散式追蹤 69
營運設計 71
OPS05-BP01 使用版本控制 71
OPS05-BP02 測試並驗證變更 73

OPS05-BP03 使用組態管理系統 76
OPS05-BP04 使用建置和部署管理系統 78
OPS05-BP05 執行修補程式管理 80
OPS05-BP06 共用設計標準 83
OPS05-BP07 實作用於提高程式碼品質的實務 85
OPS05-BP08 使用多個環境 87
OPS05-BP09 進行頻繁、細微和可逆的變更 88
OPS05-BP10 完全自動化整合和部署 89
緩解部署風險 91
OPS06-BP01 為失敗變更進行規劃 91
OPS06-BP02 測試部署 93
OPS06-BP03 採用安全的部署策略 95
OPS06-BP04 自動化測試和復原 98
營運準備度和變更管理 101
OPS07-BP01 確保人員能力 101
OPS07-BP02：確保對營運準備度進行一致的審查 103
OPS07-BP03 使用執行手冊執行程序 106
OPS07-BP04 使用程序手冊來調查問題 109
OPS07-BP05 做出部署系統和變更的明智決策 113
OPS07-BP06 啟用生產工作負載的支援計劃 114
營運 117
利用工作負載可觀測性 117
OPS08-BP01 分析工作負載指標 118
OPS08-BP02 分析工作負載日誌 120
OPS08-BP03 分析工作負載追蹤 122
OPS08-BP04 建立可付諸行動的警示 124
OPS08-BP05 建立儀表板 127
了解運作狀態 129
OPS09-BP01 使用指標衡量營運目標與 KPI 130
OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況 131
OPS09-BP03 檢閱營運指標並優先改進 133
回應事件 134
OPS10-BP01 使用程序進行事件、事故和問題管理 135
OPS10-BP02 每個提醒建立一個程序 139
OPS10-BP03 根據業務影響確定營運事件的優先順序 142
OPS10-BP04 定義向上呈報路徑 144

OPS10-BP05 定義處理影響服務事件的客戶通訊計畫 146
OPS10-BP06 透過儀表板傳達狀態 149
OPS10-BP07 自動回應事件 151
演進 154
學習、分享和改進 154
OPS11-BP01 建立持續改進程序 154
OPS11-BP02 執行事件後分析 156
OPS11-BP03 實作回饋迴圈 158
OPS11-BP04 執行知識管理 161
OPS11-BP05 定義改進驅動因素 163
OPS11-BP06 驗證洞見 165
OPS11-BP07 執行營運指標審查 166
OPS11-BP08 記錄和分享獲得的經驗 168
OPS11-BP09 分配改進時間 169
結論 171
作者群 172
深入閱讀 173
文件修訂 174

卓越營運支柱 - AWS Well-Architected Framework

出版日期： 2024 年 6 月 27 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

此 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效益和風險。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：

- 卓越營運
- 安全性
- 可靠性
- 效能達成效率
- 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

在 Amazon，我們將卓越營運視為一項承諾，致力妥善設計軟體，並提供絕佳的客戶體驗。其中包括組織團隊、設計工作負載、大規模運作工作負載，以及促使其隨時間進化的最佳實務。卓越營運有助於讓您的團隊專心設計對客戶有利的新功能，避免將時間耗費在維護與緊急應變上。為了正確建置，我們依循相關最佳實務，為您與團隊造就妥善運作的系統、平衡的工作負載，且最重要的是，為客戶創造絕佳體驗。

卓越營運的總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：

- 以業務成果為中心來組織團隊：團隊可以從領導願景、有效的營運和業務一致的營運模式，獲取業務成果。領導階層應充分投入並致力於 CloudOps 轉型，採用合適的雲端營運模式，激勵團隊以最有效率的方式運作，並獲得業務成果。正確的營運模式會利用人員、流程和技術能力，並且擴展、最佳化生產力，並於敏捷性、回應能力和適應性脫穎而出。組織的長期願景會轉化為橫跨整個企業的目標，並傳達給利益相關者和雲端服務消費者。目標和營運 KPI 於所有層級皆一致。這種做法可以維持實作下列設計原則所獲得的長期價值。
- 實作可觀測性以獲得可採取行動的見解：全面了解工作負載的行為、效能、可靠性、成本和運作狀態。建立關鍵績效指標 (KPI) 並利用可觀測性遙測，以做出明智決策並在業務成果面臨風險時立即採取行動。根據可採取行動的可觀測性資料，主動改善效能、可靠性和成本。
- 視情況盡可能自動化：在雲端，您可以在整個環境中套用與您應用程式程式碼所用的相同工程原則。您可以將整個工作負載及其營運 (應用程式、基礎架構、組態和程序) 定義為程式碼，並將其更新。接著，便可自動化工作負載營運，在回應事件時啟動這些作業。在雲端中，自動化安全可以透過設定防護機制完成部署，包括錯誤率控制、錯誤臨界值和核准。透過有效的自動化，您就可以達到一致的事件回應、限制人為錯誤，並減少操作人員疲累情況。
- 進行頻繁、細微和可逆的變更：設計可擴展且鬆散耦合的工作負載以允許定期更新元件。自動化部署技術加上較細微的增量變更可縮減影響範圍，並在發生故障時更快地反轉情況。這可讓您更有信心您能為工作負載帶來有益的變更，同時維持品質並迅速適應市場情況的變化。
- 經常完善營運程序：隨著您工作負載的演進，您的營運也應該適當演進。在使用營運程序時，尋找機會予以改善。保持定期檢閱，並驗證所有程序是否有效以及團隊是否熟悉這些程序。如果發現漏洞，

請相應地更新程序。向所有利害關係人和團隊傳達程序更新消息。將營運遊戲化以分享最佳實務並教導團隊。

- 預期失敗：盡量提升營運的成果，透過失敗案例了解工作負載的風險概況，及其對於業務成果的影響。測試程序的有效性，以及團隊面對這些模擬失敗的反應。做出明智的決策，並管理經由測試所識別的開放風險。
- 從所有營運事件和指標中學習：從所有營運事件和失敗中學習經驗，進而不斷改善。跨團隊及在整個組織中分享獲得的經驗。學習內容應強調有關營運如何有助於業務成果的資料與軼事。
- 使用受管服務：盡可能地使用 AWS 受管服務以降低營運負擔。圍繞與這些服務的互動建置營運程序。

定義

雲端有四種最佳實務領域可實現卓越營運：

- 組織
- 準備
- 營運
- 演進

組織的領導階層定義業務目標。貴組織必須了解要求和優先順序，並運用這些資訊規劃和進行用以幫助達成業務成果的工作。您的工作負載必須提供支援工作負載所需的資訊。透過自動化重複程序的方式實作服務以整合、部署及交付工作負載，將使得更多有益的變更發揮作用。

工作負載的操作本質上就可能存在著風險。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，並回應事故。您的優先事項會隨著業務需求和業務環境的變化而改變。運用這些方面做為回饋迴圈，以持續推動貴組織的改善和工作負載的操作。

組織

您需要了解組織的優先事項、組織結構，以及組織如何支援您的團隊成員，進而助您實現業務成果。

若要實現卓越營運，您必須了解以下事項：

主題

- 組織優先事項
- 操作模式
- 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，以便在組織需求變更時將其更新。

最佳實務

- OPS01-BP01 評估客戶需求
- OPS01-BP02 評估內部客戶需求
- OPS01-BP03 評估管控要求
- OPS01-BP04 評估合規要求
- OPS01-BP05 評估威脅態勢
- OPS01-BP06 在管理效益和風險的同時評估權衡

OPS01-BP01 評估客戶需求

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

期望的結果：

- 您可以從客戶成果進行反向操作。
- 您了解您的營運實務如何支援業務成果和目標。

- 您與所有相關方交流。
- 您有掌握客戶需求的機制。

常見的反模式：

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

建立此最佳實務的優勢：需求被滿足的客戶更有可能成為忠實客戶。評估和了解外部客戶的需求，將讓您了解如何安排工作的優先順序來實現商業價值。

未建立此最佳實務時的風險暴露等級：高

實作指引

了解業務需求：只有業務、開發及營運團隊等利害關係人擁有共同的目標並達成共識，才能造就企業的成功。

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

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

資源

相關的最佳實務：

- OPS11-BP03 實作回饋迴圈

OPS01-BP02 評估內部客戶需求

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

期望的結果：

- 您使用既定的優先事項，將改善工作集中在能發揮最大的影響力的地方 (例如，發展團隊技能、改善工作負載效能、降低成本、自動化執行手冊或提升監控力)。
- 隨著需求的變化來更新您的優先順序。

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

資源

相關的最佳實務：

- OPS11-BP03 實作回饋迴圈

OPS01-BP03 評估管控要求

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

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

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

實作計劃的工作量：中。實作遺漏的管控要求可能會導致工作負載重新作業。

資源

相關的最佳實務：

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

相關文件：

- AWS 管理與管控雲端環境指南
- 多帳戶環境中的 AWS Organizations 服務控制政策的最佳實務
- AWS 雲端 中的管控：敏捷和安全之間的正確平衡
- 什麼是管控、風險和合規 (GRC)？

相關影片：

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

相關範例：

- AWS Config 合規套件範例

相關服務：

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

OPS01-BP04 評估合規要求

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

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

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

- 評估和了解套用到工作負載的合規要求，可讓您了解如何安排工作的優先順序來實現商業價值。
- 您可以選擇與合規架構相符的適當位置和技術。
- 針對可稽核性設計工作負載，可讓您證明您確實遵循合規架構。

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

- AWS 合規中心

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

相關影片：

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

相關範例：

- AWS 上的 PCI DSS 和 AWS 基礎安全最佳實務

相關服務：

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

OPS01-BP05 評估威脅態勢

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

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

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

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

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

期望的結果：

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

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

資源

相關的最佳實務：

- SEC01-BP07 使用威脅模型識別威脅並優先考慮緩解措施

相關文件：

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

相關影片：

- AWSre:Inforce 2023 - 協助改善威脅建模的工具

OPS01-BP06 在管理效益和風險的同時評估權衡

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

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

AWS 可以協助您教育您的團隊有關 AWS 及其服務的知識，從而讓他們更加了解自己的選擇會如何影響工作負載。使用 AWS Support (AWS 知識中心、AWS 論壇和 AWS Support 中心) 和 AWS 文件中的資源來教育您的團隊。如有更多問題，請聯絡 AWS Support。

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

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

常見的反模式：

- 您的投資者要求您證明支付卡產業資料安全標準 (PCI DSS) 的合規性。您沒有考量滿足要求和繼續您目前開發工作之間的權衡取捨。相反地，您在不證明合規性的情況下，繼續開發工作。由於對平台安全性及其投資的擔憂，您的投資者會停止對公司的支援。
- 您決定採用您的一位開發人員在網際網路上找到的一個程式庫。您尚未評估從未知來源採用此程式庫的風險，並且不知道它是否包含弱點或惡意程式碼。
- 需要遷移的最初商務理由是基於 60% 的應用程式工作負載的現代化。但是，由於有技術上的困難，最後決定只進行 20% 的現代化，因而降低了計畫的長期效益，使基礎設施團隊以人工方式支援舊式系統的操作員勞動工作增加，導致組織更加依賴未針對此變更做準備的基礎設施團隊開發新技能。

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

未建立此最佳實務時的風險暴露等級：中

實作指引

管理效益和風險應由治理機構定義，而該機構可以驅動關鍵決策要求。您希望根據對組織有利的方式來做決策和訂立優先順序，並了解其中可能牽涉到的風險。準確的資訊對於做出組織決策至關重要。這應該要以紮實的評估測量為基礎，並須由常見的產業成本效益分析實務定義。若要做出這些類型的決策，請在集中化和分散權限之間取得平衡。權衡是無可避免的，而了解每項選擇對於定義的策略和想要達成的業務成果的影響，是非常重要的一環。

實作步驟

1. 在整體雲端管控架構中，使優勢評估做法變成一項例行作業。
   a. 在某些決策執行的集中控制與分散權限之間取得平衡。
   b. 了解加諸在每一項決策上的重擔及其決策流程，都會拖慢您的腳步。
   c. 將外部因素納入您的決策過程中 (如合規要求)。

2. 為不同層級的決策建立有共議的決策架構，其中包括誰需要解鎖受利益衝突影響的決策。
   a. 集中於可能無法逆轉的單向門決策。
   b. 允許較低職階的組織領導者做出雙向門決策。
3. 了解和管理效益和風險。在決策的收益與所涉及的風險之間取得平衡。
   a. 識別效益：根據業務目標、需求和優先事項識別效益。範例包括業務案例影響、上市時間、安全性、可靠性、效能和成本。
   b. 識別風險：根據業務目標、需求和優先事項識別風險。範例包括上市時間、安全性、可靠性、效能和成本。
   c. 評估風險與效益並做出明智決策：根據關鍵利害關係人 (包括業務、開發和營運團隊) 的目標、需求和優先事項，確定效益和風險的影響。評價收益的價值時要考慮發生風險的可能性及其代價。例如，強調上市速度優先於可靠性，可能提供競爭優勢。不過，如果發生可靠性問題，則可能會縮短正常執行時間。
4. 以程式設計方式強制執行關鍵決策，讓您自動遵守合規要求。
5. 利用已知的產業架構和功能 (例如價值流分析和 LEAN)，以目前狀態績效和業務指標為基準，並定義改進這些指標的進度迭代。

實作計畫的工作量：中高

資源

相關的最佳實務：

- OPS01-BP05 評估威脅態勢

相關文件：

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

相關影片：

- 播客 | Jeff Bezos | 關於如何做決策

相關範例：

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

操作模式

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

團隊的需求將由其產業、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所有團隊及其工作負載。

存在於組織內的操作模式數量可能會隨著開發團隊數量增長。您可能需要使用一組操作模式。

採用標準和消耗服務可幫助簡化操作，並限制操作模式的支援重擔。根據共同標準進行的開發工作所得到的效益，會隨著已採用該標準的團隊數目，以及誰將採用新功能而放大。

機制存在的目的應為請求標準新增、變更及例外狀況，以支援團隊的活動。如果沒有此選項，標準就會限制創新。評估效益和風險後，若可行則應核准請求，並判斷請求是否合適。

明確定義的一組責任將可減少工作發生衝突或重複的頻率。當業務、開發和營運團隊之間合作無間、關係密切時，則更容易達成業務成果。

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。

視他們支援的工作負載而定，我們的團隊可能在數種模式中擔負數個部份的責任。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。

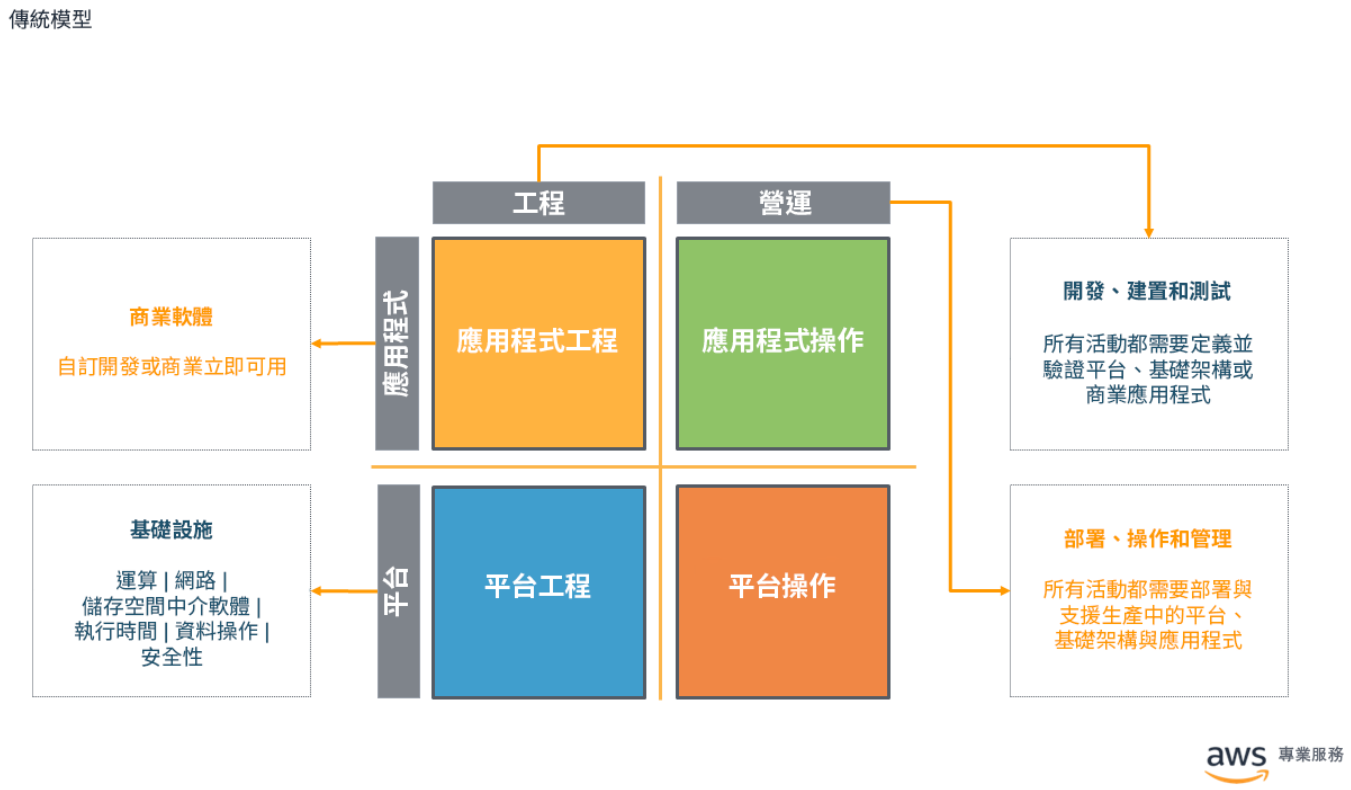
主題

- 完全獨立的操作模式
- 採用集中管控的獨立應用程式工程和操作 (AEO) 和基礎設施工程和操作 (IEO)
- 採用集中管控和服務供應商的獨立 AEO 和 IEO
- 採用集中管控和內部服務供應商諮詢合作夥伴的獨立 AEO 和 IEO
- 採用分散管控的獨立 AEO 和 IEO

完全獨立的操作模式

應用程式和基礎設施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。基礎設施是指實體和虛擬的基礎設施，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。



此「完全獨立」的模式存在於許多組織中。各象限中的活動由個別的團隊執行。 工作透過工作請求、工作佇列、票證或使用 IT 服務管理 (ITSM) 系統等機制，在團隊之間轉交。

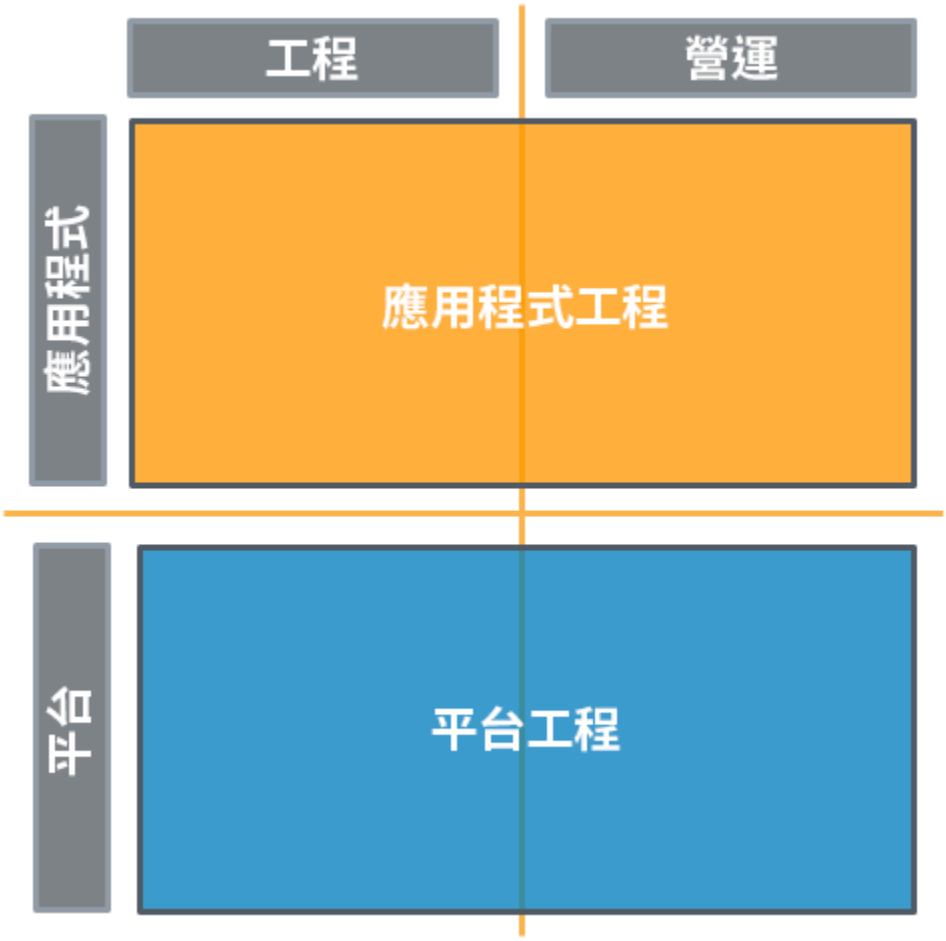
任務轉給團隊或團隊間的任務轉交會讓事情變得更複雜，並形成瓶頸與延誤。除非是要優先處理的請求，否則請求可能延誤處理。如果晚發現缺陷，可能需要大量重新作業，也可能需要再次交由相同的團隊和職務處理。如果發生需由工程團隊採取行動的事件，其回應會因遞交活動而受到延誤。

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。

採用集中管控的獨立應用程式工程和操作 (AEO) 和基礎設施工程和操作 (IEO)

此「獨立 AEO 和 IEO」模式依循「誰開發誰執行」方法。

您的應用程式工程師和開發人員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。



在此範例中，我們將進行集中管控。標準會分發、提供給應用程式團隊，或與之共用。

您應該使用能集中管控跨帳戶環境的工具或服務，例如 AWS Organizations。諸如 AWS Control Tower 等服務會擴大此管理功能，讓您能定義帳戶設定的藍圖 (支援您的操作模式)、使用 AWS Organizations 套用持續管控，以及自動化新帳戶的佈建作業。

「誰開發誰執行」並不表示應用程式團隊負責完整的堆疊、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、備份和復原工具及網路)。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，有助於限縮在強制執行管控時履行請求的相關延誤。

平台團隊可提供完整的堆疊能見度，以便應用程式團隊可以區分其應用程式元件和服務，以及其應用程式消耗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導。

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，以支援團隊的活動及其應用程式的創新。

獨立 AEO IEO 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。如此提高的能見度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。

上述原則對於支援應用程式團隊的平台團隊也同樣適用。

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。應用程式團隊透過採用標準可以專注於差異化其工作負載。

採用集中管控和服務供應商的獨立 AEO 和 IEO

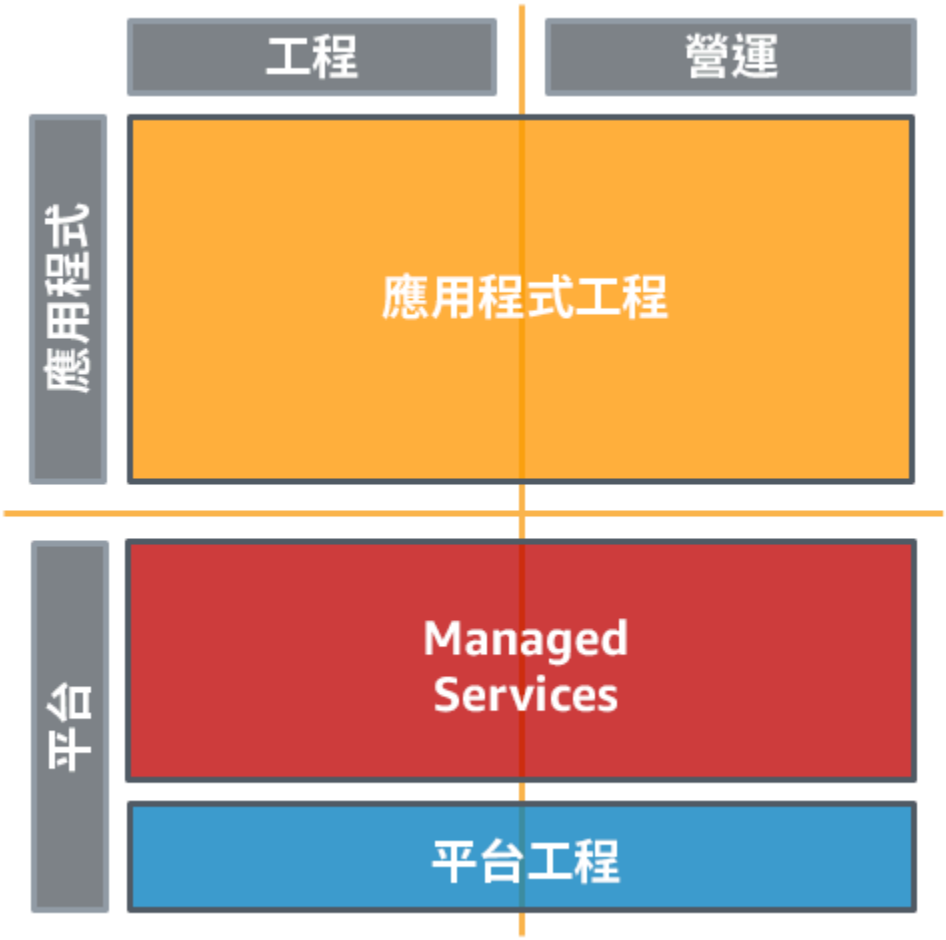
此「獨立 AEO 和 IEO」模式依循「誰開發誰執行」方法。

您的應用程式工程師和開發人員會執行其工作負載的工程和操作。

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，或您可能不想要投入時間和人力物力來進行此工作。

或者，您可能想要永遠專注於建立讓您的企業脫穎而出之功能的平台團隊，但卻想要擺脫一成不變的日常作業，交給外包商執行。

諸如 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。



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。

此模式的確需要您修改機制，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。

採用集中管控和內部服務供應商諮詢合作夥伴的獨立 AEO 和 IEO

此「獨立 AEO 和 IEO」模式企圖建立「誰開發誰執行」方法。

您想要應用程式團隊為其工作負載執行工程和操作活動，並採用更類似於 DevOps 的文化。

您的應用程式團隊可能正在進行遷移、採用雲端或將工作負載現代化，但目前沒有技能足以支援雲端和雲端操作。在應用程式團隊能力或熟悉度方面的缺乏可能會成為您工作的障礙。

為了解決這個問題，您應建立一個雲端啟用中心團隊 (CCoE)，讓其提供一個論壇以便提出問題、討論需求，和識別解決方案。根據您組織的需求，CCoE 可以是一個專屬的專家團隊，也可以是一個虛擬團隊，其中參與者選自您的整個組織。CCoE 可讓團隊進行雲端轉型、建立集中式雲端管控，以及定義帳戶和組織管理標準。他們也會識別成功的參考架構和模式，供企業使用。

我們將 CCoE 稱為雲端啟用中心，而不是更常見的雲端卓越中心，以強調讓支援團隊能夠成功並實現商務成果。

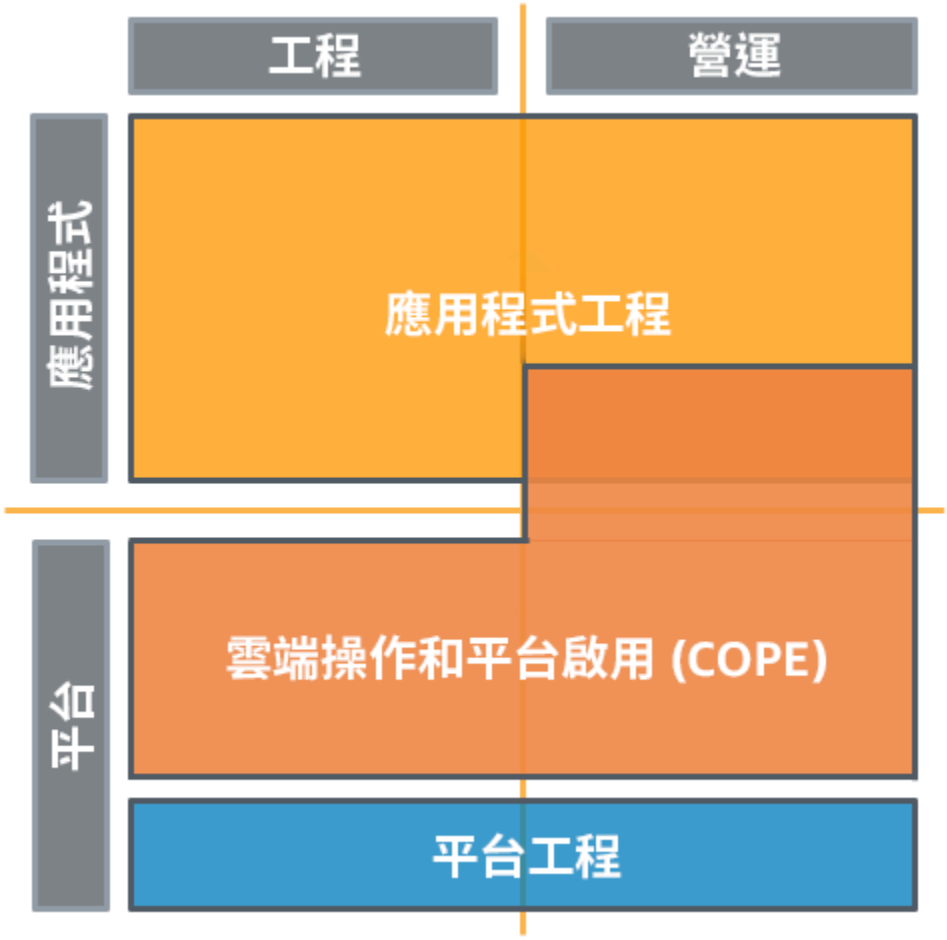
您的平台工程團隊會根據這些標準建置核心共用平台功能，供應用程式團隊採用。他們編纂了透過自助服務機制提供給應用程式團隊的企業參考架構和模式。使用 AWS Service Catalog 這類服務，應用程式團隊可以部署已核准的參考架構、模式、服務和組態，並預設符合集中管控和安全標準。

平台工程團隊也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、備份和復原工具及網路)。

您的組織擁有一個「內部 MSP 和諮詢合作夥伴」，用來管理和支援標準化服務，並協助應用程式團隊根據參考架構和模式建立其雲端服務。這個「雲端操作和平台啟用 (COPE)」團隊會與應用程式團隊合作，以協助他們建立基準操作，但隨著時間的推移，應用程式團隊逐漸對其系統和資源承擔更多責任。COPE 團隊與 CCoE 和平台工程團隊一起推動持續改進，並充當應用程式團隊的支持者。

應用程式團隊獲得協助設定環境、CICD 管道、變更管理、可觀察性和監控，以及視需要與 COPE 團隊一起建立事故和事件管理程序，而這些程序會與企業的類似程序整合。COPE 團隊與應用程式團隊一起參與這些操作活動的執行，一旦應用程式團隊獲得擁有權，就會取消 COPE 團隊的參與。

應用程式團隊受益於 COPE 團隊的技能和組織學到的經驗教訓。他們受到透過集中管控建立的防護機制保護。應用程式團隊建立在公認的成功之上，並在持續發展他們所採用的組織標準中獲益。他們透過建立可觀察性和監控的過程更深入地洞悉其工作負載的操作，並且能夠更好地了解他們對其工作負載所做變更的影響。



COPE 團隊保留支援操作活動、提供跨應用程式團隊的企業營運檢視，以及提供關鍵事故管理支援所需的存取權。COPE 團隊對視為無差別繁重工作的活動保留責任，其會透過可大規模支援的標準解決方案來滿足這些活動。他們也會繼續針對應用程式團隊管理熟知的程式設計和自動化操作活動，以便他們專注於區分其應用程式。

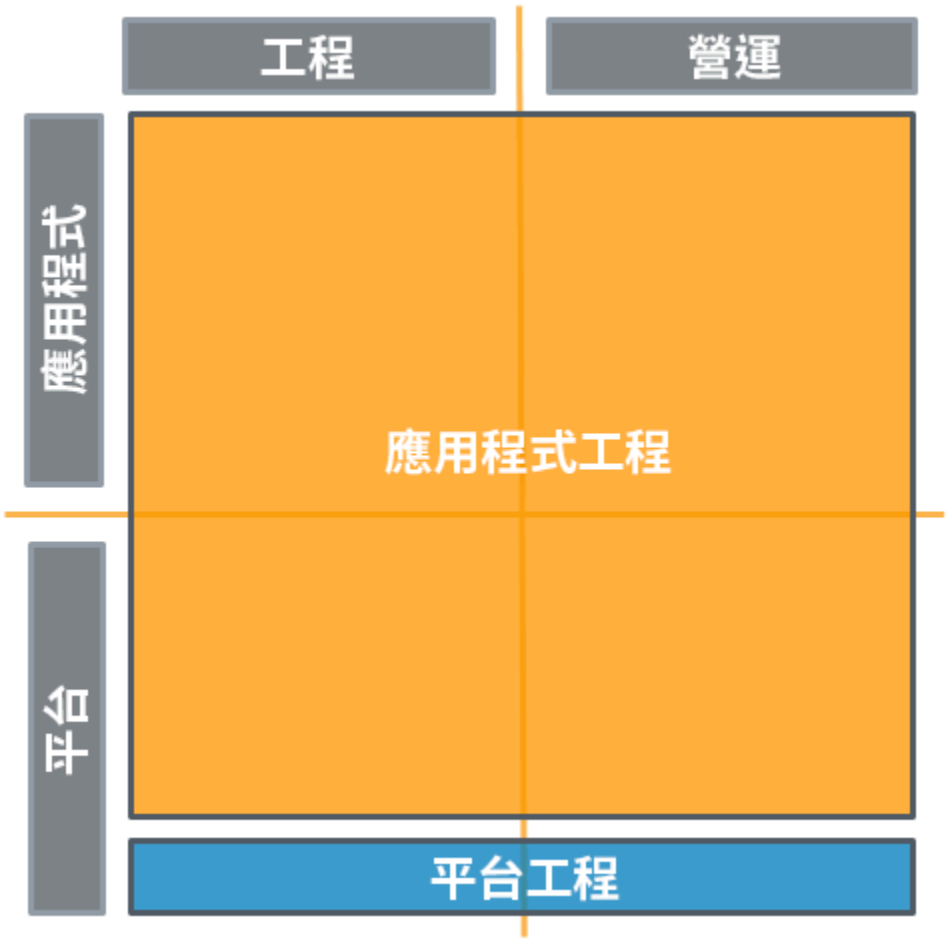
您可從組織標準、最佳實務、程序，以及衍生自團隊成功的專業知識獲得優勢。您可建立一種機制來複製這些成功模式，讓新團隊在雲端上採用或進行現代化。此模型強調 COPE 團隊能夠協助建立應用程式團隊，並轉移知識和成品。它會減輕應用程式團隊的操作負擔，但存在應用程式團隊無法獨立自主的風險。它會在 CCoE、COPE 和應用程式團隊之間建立關係，從而建立反饋迴圈以支援進一步的演進和創新。

建立您的 CCoE 和 COPE 團隊，同時定義全組織標準，可以促進雲端採用並支援現代化工作。藉由額外支援 COPE 團隊作為應用程式團隊的顧問和合作夥伴，您可以移除減緩應用程式團隊採用有益雲端功能的障礙。

採用分散管控的獨立 AEO 和 IEO

此「獨立 AEO 和 IEO」模式依循「誰開發誰執行」方法。

您的應用程式工程師和開發人員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。



在此範例中，我們將進行分散管控。

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。

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

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。他們能夠貢獻可以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。

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 資源已確認擁有者
- OPS02-BP02 流程和程序已確認擁有者
- OPS02-BP03 已為營運活動識別負責其效能的擁有者
- OPS02-BP04 存在管理責任和擁有權的機制
- OPS02-BP05 存在用於要求新增、變更和例外狀況的機制
- OPS02-BP06 團隊之間的責任是預先定義或經過協商的

OPS02-BP01 資源已確認擁有者

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

期望的結果：

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

常見的反模式：

- AWS 帳戶 的替代聯絡人未填入。

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

1. 首先為您的組織定義擁有權。擁有權可暗示資源的風險由誰承擔、誰擁有資源的變更，或誰支援疑難排解期間的資源。擁有權也可暗示資源的財務或管理擁有權。
2. 使用 AWS Organizations 管理帳戶。您可以集中管理帳戶的替代聯絡人。
   a. 只要使用公司擁有的電子郵件地址和電話號碼作為聯絡資訊，即使聯絡資訊所屬的個人已離職，您仍可存取這些資訊。例如，為帳單、營運和安全建立各別的電子郵件分發清單，在每個作用中的 AWS 帳戶 中將這些設定為帳戶、安全和營運聯絡人。即使某人休假、職務變動或離職，仍有多人會收到 AWS 通知並且能有所回應。
   b. 如果帳戶未由 AWS Organizations 管理，替代帳戶聯絡人可協助 AWS 在必要時聯繫到適當人員。設定帳戶的替代聯絡人以將其指向團體而非個人。
3. 使用標籤來識別 AWS 資源的擁有者。您可以用個別的標籤指定擁有者及其聯絡資訊。
   a. 您可以使用 AWS Config 規則強制資源要有必要的擁有權標籤。

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

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

資源

相關的最佳實務：

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

相關文件：

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

相關研討會：

- AWS 研討會 - 標記

相關範例：

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

相關服務：

- AWS Config 規則 - required-tags
- AWS Organizations

OPS02-BP02 流程和程序已確認擁有者

了解誰具有個別流程和程序的擁有權、為何使用特定流程和程序，以及為何該擁有權存在。了解使用特定流程和程序的原因，有助於找出改進機會。

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

常見的反模式：

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

建立此最佳實務的優勢：

- 流程和程序可以大幅提高工作負載的工作量。
- 新的團隊成員工作更快、效率更高。
- 縮短緩解事件的時間。
- 不同的團隊成員 (和團隊) 可以採用一致方式，使用相同的流程和程序。
- 團隊可用可重複的程序來擴展其程序。
- 標準化的流程和程序有助於減輕團隊之間轉移工作負載職務的影響。

未建立此最佳實務時的風險暴露等級：高

實作指引

- 流程和程序已確認擁有者負責其定義。
  - 識別為支援工作負載所執行的營運活動。將這些活動記錄在可探索的位置中。
  - 唯一識別負責活動規格的個人或團隊。他們負責確認具備適當技能的團隊成員能夠成功執行該活動，且該團隊成員具備正確許可、存取權和工具。如果執行該活動時發生問題，執行該活動的團隊成員需負責提供改善活動所需的詳細回饋。
  - 透過 AWSSystems Manager 等服務、文件與 AWS Lambda，擷取活動成品中繼資料中的擁有權。使用標籤或資源群組擷取資源擁有權，並指定擁有權和聯絡資訊。使用 AWS Organizations 建立標記政策，並確保擷取擁有權和聯絡資訊。
- 長期下來，這些程序應該會獲得改善，而達到可依程式碼執行，減少人工介入的必要性。
  - 例如，考慮 AWS Lambda 函數、CloudFormation 範本或 AWSSystems Manager 自動化文件。
  - 在適當的儲存庫中執行版本控制。
  - 包含合適的資源標籤，以便識別擁有者和文件。

客戶範例

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

實作步驟

1. 記錄現有的流程和程序。
   a. 檢閱並保持為最新狀態。
   b. 識別每個流程或程序的擁有者。
   c. 實施版本控制。
   d. 在可能情況下，於共用架構設計的工作負載和環境之間共用流程和程序。
2. 建立意見回饋和改善的機制。

   a. 定義程序應多久檢閱一次的政策。
   b. 定義檢閱者與核准者適用的程序。
   c. 實作問題或票務佇列，以便提供與追蹤意見回饋。
   d. 在可能的情況下，流程和程序應由變更核准委員會 (CAB) 進行預先核准和風險分類。
3. 確認執行流程和程序可以供需要執行的人存取和探索。
   a. 使用標籤，指出可供工作負載存取流程和程序的所在位置。
   b. 使用有意義的錯誤和事件訊息，指出可解決問題的適當流程或程序。
   c. 使用 Wiki 和文件管理，讓這些流程和程序一致可供整個組織的人搜尋。
4. 適當時機進行自動化。
   a. 當服務和技術提供 API 時，應該開發自動化。
   b. 充分教育有關程序的相關知識。發展那些流程的用戶故事和自動化需求。
   c. 成功評估流程和程序的使用情況，並提出支援反覆改進的問題。

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

- AWS 白皮書 - AWS 上的 DevOps 簡介
- AWS 白皮書 - 標記 AWS 資源的最佳實務
- AWS 白皮書 - 使用多個帳戶整理您的 AWS 環境
- AWS 雲端 作業與遷移部落格 - 建置雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- AWS 雲端 作業與遷移部落格 - 使用 AWS Config 和 AWS Organizations 實作自動化和集中化的標記控制
- AWS 安全性部落格 - 使用 AWS CloudFormation Guard 擴展您的預先提交勾點

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

相關研討會：

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

相關影片：

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

相關服務：

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

OPS02-BP03 已為營運活動識別負責其效能的擁有者

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

期望的結果：

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

常見的反模式：

- 您不記錄職責。
- 隔離的操作員工作站存在片段指令碼。只有少數個人知道如何使用這些指令碼，或將其非正式稱為團隊知識。
- 舊版處理序的更新日期已到期，但沒有人知道該處理序的擁有者，而原始作者已離開組織。

- 處理序和指令碼均無法找到，因此無法適時供人使用 (例如，要回應事故時)。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

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

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

客戶範例

AnyCompany Retail 將擁有權定義為負責應用程式或應用程式群組 (而群組中的應用程式共用常見的架構實務和技術) 之程序的團隊或個人。起初公司將流程和程序記錄為文件管理系統中的逐步指南。公司使用託管應用程式的 AWS 帳戶 和帳戶內特定資源組上的標籤來探索程序，並使用 AWS Organizations 來管理其 AWS 帳戶。過了一段時間，AnyCompany Retail 將這些流程轉換為程式碼，並使用基礎設施即程式碼 (透過 CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 範本等服務) 定義資源。作業流程會成為在 AWS Systems Manager 或 AWS Lambda 函數中的自動化文件，

可以啟動為排程任務，以回應 Amazon CloudWatch 警示或 Amazon EventBridge 事件等事件，或是因應 IT 服務管理 (ITSM) 平台內的請求而啟動。所有流程都有標籤用來識別擁有者。團隊會在流程程式碼儲存庫產生的 Wiki 頁面中管理自動化和流程的文件。

實作步驟

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

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS02-BP01 資源已確認擁有者
- OPS02-BP02 流程和程序已確認擁有者
- OPS02-BP04 存在管理責任和擁有權的機制

- OPS02-BP05 存在用來識別責任和擁有權的機制
- OPS11-BP04 執行知識管理

相關文件：

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

相關影片：

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

相關範例：

- AWS Well-Architected 卓越營運研討會

OPS02-BP04 存在管理責任和擁有權的機制

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

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

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

為您的組織成員提供可存取的機制，以探索和識別擁有權和責任。這些機制會教導他們有特定需求時應聯絡誰。

客戶範例

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

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

實作步驟

1. 從現有文件開始。一些典型的來源文件可能包括：
   a. 責任或負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣
   b. 團隊定義或 Wiki 頁面
   c. 服務定義和供應項目
   d. 角色或工作描述
2. 審查並主持有關記錄的責任討論：
   a. 與團隊一起審查，識別團隊通常執行的記錄責任和責任之間的不一致性。
   b. 討論內部客戶提供的潛在服務，以識別團隊之間的期望落差。
3. 分析並解決差異。
4. 識別改進機會。
   a. 識別經常提出的資源密集型請求，這些請求通常是最需要改進的對象。
   b. 尋找最佳實務、模式和規範指引，透過本指引簡化和標準化改進事宜。
   c. 記錄改進機會，並追蹤直至完成。
5. 如果團隊尚未進行管理和追蹤責任分派，請確定團隊中擔負此職責的人。
6. 定義團隊請求解釋責任的流程。
   a. 檢閱該流程，並確認流程是否夠清晰且易於使用。
   b. 確保有人扛責並追蹤呈報至得出結論。
   c. 建立營運指標以衡量效用。
   d. 建立回饋機制，確認團隊可以突顯改進機會。

   e. 實施定期檢討的機制。
7. 以可探索且可存取的位置記錄文件。
   a. Wiki 或文件入口網站是共同的選擇。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS01-BP06 評估權衡
- OPS03-BP02 授權團隊成員在成果有風險時採取動作
- OPS03-BP03 鼓勵向上呈報
- OPS03-BP07 適當地為團隊提供資源
- OPS09-BP01 使用指標衡量營運目標與 KPI
- OPS09-BP03 檢閱營運指標並優先改進
- OPS11-BP01 建立持續改進程序

相關文件：

- AWS 白皮書 - AWS 上的 DevOps 簡介
- AWS 白皮書 - AWS 雲端 採用架構：營運觀點
- AWS Well-Architected Framework 卓越營運 - 工作負載層級作業模式拓撲
- AWS 規範性指引 - 建置您的雲端作業模式
- AWS 規範性指引 - 為雲端作業模式建立 RACI 或 RASCI 矩陣
- AWS 雲端 營運與遷移部落格 - 透過雲端平台團隊提供商業價值
- AWS 雲端 營運與遷移部落格 - 為什麼要使用雲端作業模式？
- AWS DevOps 部落格 - 組織如何將雲端作業現代化

相關影片：

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

OPS02-BP05 存在用於要求新增、變更和例外狀況的機制

您可以向流程、程序和資源的擁有者提出要求。要求包含新增、變更和例外狀況。這些要求會經歷變更管理程序。評估收益和風險後，若可行並經判斷是合適的行為，則應制定明智的決策以核准要求。

期望的結果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

客戶範例

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

實作步驟

1. 識別您工作負載的流程、程序和資源，及其各自的擁有者。在您的知識管理系統中加以記錄。
   a. 如果您尚未實作 OPS02-BP01 資源已確認擁有者、OPS02-BP02 流程和程序已確認擁有者 或 OPS02-BP03 已為營運活動識別負責其效能的擁有者，請先予以實作。

2. 與組織中的利害關係人合作制定變更管理程序。此程序應涵蓋資源、流程和程序的新增、變更與例外狀況。
   a. 您可以使用 AWS Systems Manager Change Manager 作為工作負載資源的變更管理平台。
3. 在您的知識管理系統中記錄變更管理程序。

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

資源

相關的最佳實務：

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

相關文件：

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

相關服務：

- AWS Systems Manager Change Manager

OPS02-BP06 團隊之間的責任是預先定義或經過協商的

團隊間已定義或協商說明如何相互配合及支援的協議 (例如，回應時間、服務水準目標或服務水準協議)。團隊間的溝通管道記錄於文件中。透過了解團隊工作對於業務成果和其他團隊及組織成果的影響，可得知其任務的優先順序，並協助他們適當地回應。

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

期望的結果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：低

實作指引

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

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

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 提供高層的支持
- OPS03-BP02 授權團隊成員在成果有風險時採取動作
- OPS03-BP03 鼓勵向上呈報
- OPS03-BP04 溝通需及時、清楚且可行
- OPS03-BP05 鼓勵進行試驗
- OPS03-BP06 團隊成員受到鼓勵來維持和培養自己的技能集
- OPS03-BP07 適當地為團隊提供資源

OPS03-BP01 提供高層的支持

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

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：高

實作指引

- 若要在雲端旅程的每個階段 (遷移、採用或最佳化) 中取得成功，最高層領導就必須透過指定的執行任務發起者主動參與過程。執行任務發起者會根據定義的策略調整團隊的思維、技能和工作方式。
  - 解釋原因：闡明並解釋願景和策略背後的原因。
  - 設定期望：為您的組織定義和發佈目標，包括衡量進度和成功的方式。
  - 追蹤目標的達成情況：定期評估目標的遞增實現情況 (不僅是完成任務)。分享結果，以便在結果有風險時，可以採取適當的行動。
  - 提供實現目標所需的資源：將人員和團隊聚集在一起，共同合作建立合適的解決方案，以達成所設定的結果。如此便可減少或消除組織摩擦。
  - 支持您的團隊：與團隊保持合作，讓您了解團隊的表現，以及是否有外部因素正影響著他們。找出阻礙您團隊前進的障礙。代表您的團隊來協助解決障礙並消除不必要的負擔。當您的團隊受到外部因素影響時，請重新評估目標並適當地調整目標。
  - 推動採用最佳實務：確認提供量化效益的最佳實務，並認可建立者和採用者。鼓勵進一步採用，以擴大已達成的效益。
  - 鼓勵團隊的進化：建立持續改進的文化，並主動從進步和失敗中學習。鼓勵人員和組織的成長和發展。利用資料和故事發展願景和策略。

客戶範例

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

實作步驟

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

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

- 清理組織內的阻礙：高度一致
- 現實中的轉型：務實地接近變革

- 成為具前瞻性的企業
- 建置 CCOE 時應避開的 7 大陷阱
- 瀏覽雲端：成功的關鍵績效指標

相關影片：

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

相關範例：

- Prosci：主要發起者的角色和重要性

OPS03-BP02 授權團隊成員在成果有風險時採取動作

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

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

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

常見的反模式：

- 組織中沒有明確的指引或機制來說明確定風險時該怎麼做。例如，當員工注意到網路釣魚攻擊時，因他們未及時向資安防護團隊通報，導致組織內大部分成員都受到攻擊。這會導致資料洩露。
- 您的客戶抱怨服務無法使用，這主要是由於部署失敗所致。您的 SRE 團隊負責部署工具，其長期藍圖中包含部署作業自動復原。在最近推出的一款應用程式中，其中一名工程師設計了一種解決方案，可自動將其應用程式恢復到舊版本。儘管他們的解決方案可以變成 SRE 團隊的模式，但由於沒有流程可追蹤此類改進，其他團隊並未採用。該組織繼續受到部署失敗的困擾，影響客戶並導致進一步的負面情緒。

- 為了保持合規性，您的 infosec 團隊會監督一個長期的流程，代表連接到其 Amazon EC2 Linux 執行個體的操作員定期輪換共享 SSH 金鑰。infosec 團隊需要幾天才能完成輪換金鑰，而且您無法連線到這些執行個體。infosec 內部或外部沒有人建議使用 AWS 上的其他選項來達到相同的結果。

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

- 使用架構式決策記錄來簡化軟體開發專案的技術決策

OPS03-BP03 鼓勵向上呈報

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

期望的結果：組織內所有的人都很樂意將問題向上呈報至其直屬主管和更高層領導者。領導階層已經特意為團隊成員打好預防針，讓他們能安心呈報任何問題。在組織裡，有一個機制可以讓每個職級的人向上呈報問題。當員工想要呈報給他們的主管時，會共同判定問題的影響程度，以及是否應向上呈報。為了開始向上呈報的流程，員工需要同時提交一份包含解決問題建議的工作計畫。如果直屬主管沒有及時採取行動，且如果員工認為組織面臨的風險很高，則員工應將問題向上呈報給最高職級的領導者。

常見的反模式：

- 在雲端轉型計畫近況討論會議上，高層執行領導不會提出足夠的探查性問題來尋找發生問題和阻礙之處。只有好消息才會納入近況報告。CIO 明確表示，她只喜歡聽好消息，因為任何提出的挑戰都會使 CEO 認為該計畫不成功。
- 您是一名雲端營運工程師，您注意到應用團隊並未廣泛採用新知識管理系統。公司投入了一年時間和數百萬美元來實作這個新知識管理系統，但員工仍在本機編寫他們的執行手冊，並分享到組織雲端共

享區上，因此其他人很難找到與支援工作負載相關的知識。您嘗試讓領導階層注意這個問題，因為唯有一致地使用這個系統才能提高營運效率。當您將這個問題呈報給領導知識管理系統實作的總監時，她會問責於您，因為她認為您的呈報使公司對這項投資產生質疑。

- 負責加強運算資源的 infosec 團隊已決定制定一個流程，就是在運算團隊釋放資源供組織使用之前，需要執行必要的掃描以確保 EC2 執行個體完全受保護。這導致部署資源的時間延遲一週，進而破壞了他們的 SLA。運算團隊害怕將這個問題呈報給雲端副總裁，因為這會讓資安副總裁難堪。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

1. 為您的組織定義政策、標準和期望。
   1. 確保廣泛採用和了解政策、期望和標準。
2. 鼓勵、培訓並賦予員工能力，以在不符合標準時及早向上呈報。
3. 組織認可儘早且經常呈報是最佳實務。接受呈報經證明可能是無根據的，然而有機會防止事件的發生，總好過於不呈報而錯過該機會。
   a. 建立一個向上呈報機制 (如安燈線系統)。
   b. 制定定義進行向上呈報的時機與方式的記錄程序。
   c. 定義一序列擁有逐漸升高的採取或核准動作權限的人員，以及每個利害關係人的聯絡資訊。
4. 當向上呈報後應持續追蹤，直到團隊成員確定在領導階層採取行動後已使風險降低。
   a. 向上呈報應包括：
      i. 情況及風險性質描述
      ii. 情況的嚴重性
      iii. 誰或什麼會受影響
      iv. 影響程度有多大

v. 發生衝擊時的緊急情況

vi. 建議的補救措施和緩解計畫

b. 保護向上呈報的員工如果團隊成員越過無回應的決策制定者或利害關係人進行越級呈報，則組織應制定保護團隊成員免受報復的政策。制定機制以識別是否發生此情況，並適當地做出回應。

5. 鼓勵將持續改進回饋的文化注入組織內產生的一切事務中，並且不斷地反覆循環下去。回饋迴圈對負責方而言算是微級呈報程序，即使不需要向上呈報，也能找到改進的機會。持續改進文化會迫使每個人更積極主動。
6. 領導者應定期重新強調政策、標準、機制，以及對開放、不畏報復的向上呈報和持續回饋循環機制的渴望。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS02-BP05 存在用於要求新增、變更和例外狀況的機制

相關文件：

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

相關影片：

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

相關範例：

- 在事件管理員中處理向上呈報計畫

OPS03-BP04 溝通需及時、清楚且可行

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

1. 在組織內建立一個須為工作結果負責的核心團隊，他們會針對組織內多個層級發生的變化建立和啟動溝通計畫。
2. 建立單一執行團隊擁有權以利於監督。提供個別團隊獨立創新的能力，並平衡一致性機制的使用情況，以在正確的層級上進行檢查，並確定正確的展望方向。
3. 與組織中的利害關係人共同達成溝通標準、實務和計畫方面的協議。
4. 確認核心溝通團隊是否與組織和計畫領導者合作，並代表領導者向適當的員工傳達訊息。
5. 建立策略性溝通機制，以透過公告、共享行事曆、全公司會議，以及親自或一對一的方法來管理變革，以便團隊成員對他們應該採取的行動有適當的期望。
6. 提供必要的背景內容、詳細資訊和時間 (如果可能)，以判斷是否需要採取行動。當需要採取行動時，提供所需的行動及其影響力。
7. 實作可促進策略性溝通的工具，例如內部聊天、電子郵件和知識管理。
8. 實作機制來衡量和驗證所有通訊是否都導致預期的結果。
9. 建立一個回饋迴圈，用以衡量所有通訊的有效性，尤其是當整個組織中的變革引起了反對聲時。
10. 對於所有 AWS 帳戶，請針對帳單、安全性和營運建立替代聯絡人。在理想的情況中，每個聯絡人都應該列入電子郵件分發名單，而不是當成特定的個別聯絡人。
11. 建立向上呈報及其反向的通訊計畫，以用來與您的內部和外部團隊 (包括 AWS 支援和其他第三方供應商) 互動。
12. 在每個轉型計畫的生命週期中，始終如一地啟動和執行溝通策略。
13. 在可行情況下，優先處理可重複的動作，以大規模安全地執行自動化。

14. 當人員在自動化操作過程中需要通訊時，通訊的目的應該是通知團隊、進行稽核或變更管理流程的一部分。
15. 分析來自警示系統的通訊，以尋找不斷建立的誤報或警示。移除或變更這些警示，以便在需要人工干預時啟動。如果起始一項提醒，請提供執行手冊或程序手冊。
    a. 您可以使用 AWS Systems Manager Documents 來建置提示用的程序手冊和執行手冊。
16. 已設立機制，以清楚且可行的方式提供風險或計畫事件的通知，並提供足夠的通知，以便適當的回應。使用電子郵件清單或聊天管道，在計畫性事件發生之前傳送通知。
    a. AWS 聊天機器人可在您的組織傳訊平台內用來傳送提醒及回應事件。
17. 提供可存取的資訊來源，您可以在其中發現計畫的事件。提供來自相同系統之計畫事件的通知。
    a. AWS Systems Manager 變更行事曆可用來建立可進行變更的變更時段。這可為團隊成員提供有關於何時可安全進行變更的通知。
18. 監控漏洞通知和修補程式資訊，了解外部漏洞以及與工作負載元件相關的潛在風險。提供通知給團隊成員，讓他們可以採取動作。
    a. 您可以訂閱 AWS 安全公告，以接收 AWS 相關漏洞的通知。
19. 尋求多樣化的意見和觀點：鼓勵每個人做出貢獻。為代表性不足的團體提供溝通機會。在會議中輪換角色和責任。
    a. 擴展角色和責任：為團隊成員提供機會，以擔任他們可能不會擔任的角色。他們可以透過角色，以及與他們可能不會與之交涉的新團隊成員互動，從中獲得經驗和新觀點。他們也可以將自己的經驗和觀點帶到新的角色，並帶給和他們互動的團隊成員。隨著觀點的增加，找出新興商機或新的改進機會。在團隊內的成員之間輪換其他人通常會執行的常見任務，以讓全員了解執行這些任務的需求和影響。
    b. 提供安全且友善的環境：建立政策與控制措施，保護組織內團隊成員身心上的安全。團隊成員應該能夠在不擔心報復行為的情況下進行互動。當團隊成員感到安全且受歡迎時，他們才更有可能積極參與並具備生產力。您的組織越多樣化，您就越能了解所支援的人員，包括您的客戶。當您的團隊成員感到安心、可以暢所欲言，而且有信心他們的聲音不會被淹沒，他們才更有可能分享寶貴的洞見 (例如，行銷機會、可及性的需求、尚未有服務的市場區段，以及環境中未確認的風險)。
    c. 鼓勵團隊成員充分參與：提供員工充分參與所有與工作相關的活動所需的資源。面對日常挑戰的團隊成員可發展出解決挑戰的技能。這些以獨特方式發展的技能可為組織提供顯著的效益。為團隊成員提供必要便利性支援，以便從他們的貢獻中獲得更高的效益。

資源

相關的最佳實務：

- OPS03-BP01 提供高層的支持
- OPS07-BP03 使用執行手冊執行程序
- OPS07-BP04 使用程序手冊來調查問題

相關文件：

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

相關範例：

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

相關服務：

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

OPS03-BP05 鼓勵進行試驗

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

預期成果：

- 您的組織鼓勵試驗以促進創新。
- 試驗被視為一種學習機會。

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

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

相關服務：

- Amazon CloudWatch Evidently
- AWS AppConfig

- AWS Control Tower

OPS03-BP06 團隊成員受到鼓勵來維持和培養自己的技能集

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

AWS 提供資源，包括 AWS 入門資源中心、AWS 部落格、AWS線上技術會談、AWS 活動和網路研討會和 AWSWell-Architected Labs，而這些資源提供了可教育您團隊的說明、範例和詳細演練。

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

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

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

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

實作步驟

1. 使用結構化的雲端宣傳計畫： AWS Skills Guild 提供顧問式培訓，以提高人員在雲端技能方面的信心，並激發持續學習文化。
2. 為教育提供資源：提供專門的結構化時間、培訓教材和實驗室資源存取權，並支持員工參與會議和專業組織，這些會議和組織可為教育工作者和同儕提供學習的機會。為資淺團隊成員提供接近資深團隊成員的機會，讓資深團隊成員成為導師，或允許資淺團隊成員伴隨資深團隊成員工作，藉以學習他們的做法和技能。鼓勵學習與工作不直接相關的內容，以便取得更廣泛的視野。
3. 鼓勵使用專家技術資源：利用 AWSre:Post 等資源來存取精心挑選的知識，並加入充滿活力的社群。
4. 建立和維護最新的知識庫：使用知識共享平台，例如 Wiki 和執行手冊。使用 AWS re:Post Private 建立您自己可重複使用的專家知識來源，以簡化協作、提高生產力，並加快員工上線速度。
5. 團隊教育和跨團隊參與：針對團隊成員持續的教育需求進行規劃。為團隊成員提供加入其他團隊機會 (暫時或永久地)，以分享讓整個組織受益的技能和最佳實務。
6. 支援對產業認證的追求和維持：支援團隊成員取得與維護可驗證所學知識並認可其成就的產業認證。

實作計畫的工作量：高

資源

相關的最佳實務：

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

相關文件：

- AWS 白皮書 | 雲端採用架構：人員觀點
- 投資於持續學習以發展組織的未來
- AWS Skills Guild

- AWS 培訓 and Certification
- AWS Support
- AWS re:Post
- AWS 開始使用資源中心
- AWS 部落格
- AWS 雲端 合規
- AWS 文件
- 官方 AWS播客。
- AWS 線上技術會談
- AWS 活動和研討會
- AWS Well-Architected 實驗室
- 在 Amazon Builders' Library 中

相關影片：

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

OPS03-BP07 適當地為團隊提供資源

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

期望的結果：

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

實作步驟

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

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

準備

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。

要為卓越營運做好準備，您需要考慮以下事項：

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。以指標、日誌和追蹤為根基，可觀測性提供了系統行換動態的洞見。有效的可觀測性能夠讓團隊辨別模式、異常情況和趨勢，以便主動解決潛在問題並維持最佳的系統運作狀態。

確定關鍵績效指標 (KPI) 至關重要，可確保監控活動與業務目標保持一致。這種一致性可確保團隊使用真正重要的指標來做出資料驅動的決策，進而最佳化系統效能和業務成果。

此外，可觀測性使得企業能夠化被動為主動。團隊能夠了解系統內的因果關係，預測並預防問題，而不只是被動回應問題。隨著工作負載的演進，務必重新檢視並改進可觀測性策略，以保持相關性和有效性。

最佳實務

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

OPS04-BP01 識別關鍵績效指標

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 高

實作指引

若要有效地定義工作負載 KPI：

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

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

- 研擬可觀測性策略

相關範例：

- One Observability 研討會

OPS04-BP02 實作應用程式遙測

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

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

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

1. 確定要收集的資料：確定可提供工作負載運作狀況、效能和行為實質洞見的重要指標、日誌和追蹤。
2. 部署 CloudWatch 代理程式： CloudWatch 代理程式的作用在於，方便您從工作負載及其基礎設施中取得系統和應用程式指標和日誌。CloudWatch 代理程式也可用來收集 OpenTelemetry 或 X-Ray 追蹤，並傳送至 X-Ray。

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

實作計畫的工作量：高

資源

相關的最佳實務：

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

相關文件：

- AWS 可觀測性最佳實務
- CloudWatch 使用者指南
- AWS X-Ray 開發人員指南

- 檢測分散式系統，以了解運作狀態
- AWS 可觀測性 Skill Builder 課程
- Amazon CloudWatch 最新消息
- AWS X-Ray 最新消息

相關影片：

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

相關範例：

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

OPS04-BP03 實作使用者體驗遙測

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 高

實作指引

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

實作步驟

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

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

- Amazon CloudWatch RUM 指南
- Amazon CloudWatch Synthetics 指南

相關影片：

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

相關範例：

- One Observability 研討會
- Amazon CloudWatch RUM Web 用戶端的 Git 儲存庫
- 使用 Amazon CloudWatch Synthetics 測量頁面載入時間

OPS04-BP04 實作相依性遙測

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

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

常見的反模式：

- 忽略外部相依項：僅關注內部應用程式指標，而忽略與外部相依項相關的指標。
- 缺乏主動監控：等待問題出現，而非持續監控相依項的運作狀況與效能。

- 單獨運作的監控：使用多種分散的監控工具，如此可能導致僅掌握相依項的部分運作狀況且獲得不一致的資訊。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

若要有效實作相依項遙測：

1. 識別外部相依項：與利害關係人協作，共同找出工作負載所依賴的外部相依項。外部相依項可能包含各種服務，像是外部資料庫、第三方 API、前往其他環境的網路連線能力路由，以及 DNS 服務。實現有效相依項遙測的第一步，就是徹底了解這些相依項。
2. 擬訂監控策略：清楚了解外部相依項之後，就可以為其量身打造監控策略。這包括了解每一項相依項的重要性、預期行為，以及任何相關的服務層級協議或目標 (SLA 或 SLT)。設定主動警示，以便在發生狀態變更或效能偏差時通知您。
3. 使用網路監控：使用網際網路監視器和網路監視器，提供全球網際網路和網路狀況的全方位洞見。這些工具可協助您了解並回應影響外部相依項的中斷、干擾或效能降低。
4. 使用 AWS Health Dashboard 隨時掌握資訊：它會在 AWS 遇到可能影響服務的事件時，發出警示並提供修復指引。
   a. 監控使用 Amazon EventBridge 規則的 AWS Health 事件，或以程式設計方式與 AWS Health API 整合，以在您收到 AWS Health 事件時自動執行動作。這些可能是一般動作 (例如將所有計畫的生命週期事件訊息傳送到聊天介面) 或特定動作 (例如在 IT 服務管理工具中啟動工作流程)。

b. 如果您使用 AWS Organizations，請跨帳戶彙總 AWS Health 事件。

5. 使用 AWS X-Ray 檢測您的應用程式：AWS X-Ray 提供關於應用程式及其基礎相依項如何運作的洞察。透過從頭到尾追蹤請求，您就可以找出應用程式所依賴的外部服務或元件的瓶頸或故障。
6. 使用 Amazon DevOps Guru：這項機器學習驅動的服務可識別操作問題，預測重大問題可能在什麼時候發生，並且建議可採取的特定行動。對於獲得相依項洞見並確保它們不是造成操作問題的根源來說，這項服務非常寶貴。
7. 定期監控：持續監控與外部相依項相關的指標和日誌。針對非預期的行為或效能降低的情況設定警示。
8. 變更後驗證：每當有任何外部相依項更新或變更，便驗證其效能並檢查是否符合您的應用程式需求。

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

- 深入了解影響應用程式效能的網際網路問題
- Amazon DevOps Guru 簡介

- 使用 AWS Health 以大規模管理資源生命週期事件

相關範例：

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

OPS04-BP05 實作分散式追蹤

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

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

常見的反模式：

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

建立此最佳實務的優勢：

- 全方位的系統概觀：從進入到退出，徹底視覺化整個請求路徑。
- 強化偵錯：快速識別失敗或效能問題發生的位置。
- 改善使用者體驗：根據實際使用者資料進行監控與最佳化，確保系統符合實際需求。

未建立此最佳實務時的曝險等級： 高

實作指引

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

實作步驟

若要有效實作分散式追蹤：

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

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

- 使用 AWS X-Ray 檢測您的應用程式

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、細微和可逆的變更
- OPS05-BP10 完全自動化整合和部署

OPS05-BP01 使用版本控制

使用版本控制來追蹤變更和發佈。

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

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

常見的反模式：

- 您已在工作站上開發和儲存程式碼。您的工作站發生無法復原的儲存錯誤，造成程式碼遺失。
- 變更覆寫現有的程式碼之後，您重新啟動應用程式卻無法運作。您無法還原變更。
- 您對其他人要編輯的報告檔案加上了寫入鎖定。他們會與您聯絡，要求您停止處理該檔案，以便完成任務。
- 您的研究團隊一直在進行詳細的分析，以塑造您未來的工作。某人意外地將自己的購物清單儲存在最終報告中。您無法還原變更，且必須重新建立報告。

建立此最佳實務的優勢： 透過使用版本控制功能，您可以輕鬆回復為已知的良好狀態和舊版本，並有效降低資產遺失的風險。

未建立此最佳實務時的曝險等級： 高

實作指引

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

資源

相關的最佳實務：

- OPS05-BP04 使用建置和部署管理系統

相關文件：

- 什麼是 AWS CodeCommit？

相關影片：

- AWS CodeCommit 簡介

OPS05-BP02 測試並驗證變更

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

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

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

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

客戶範例

在其持續整合管道中，AnyCompany Retail 對所有軟體成品執行了數種類型的測試。他們實行了測試驅動的開發，因此所有軟體都有測試單元。在成品建置後，他們執行了端對端測試。這個第一輪測試完

成後，他們執行了靜態應用程式安全掃描，以尋找已知漏洞。開發人員在每個測試門檻通過後均收到訊息。所有測試都完成後，軟體成品即儲存在成品儲存庫中。

實作步驟

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

資源

相關的最佳實務：

- OPS05-BP01 使用版本控制
- OPS05-BP06 共用設計標準
- OPS05-BP07 實作用於提高程式碼品質的實務
- OPS05-BP10 完全自動化整合和部署

相關文件：

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

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

相關影片：

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

相關資源：

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

相關服務：

- Amazon Q Developer
- Amazon CodeGuru Reviewer

- AWS CodeBuild
- AWS CodePipeline

OPS05-BP03 使用組態管理系統

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

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

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

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

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

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

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

常見的反模式：

- 您手動更新整個機群的 Web 伺服器組態，但由於更新錯誤，導致多部伺服器無法回應。
- 您在數小時內手動更新應用程式伺服器機群。變更期間的組態不一致會導致未預期的行為。
- 某人已更新您的安全群組，無法再存取您的 Web 伺服器。若不知道進行了哪些變更，您就需要花大量時間來調查問題，復原時間也會跟著拉長。
- 您可以透過 CI/CD 將生產前組態推送到生產環境中，而不需進行驗證。您讓使用者和客戶面臨使用不正確的資料和服務。

建立此最佳實務的優勢： 採用組態管理系統可減少進行和追蹤變更的工作量，以及手動程序造成的錯誤頻率。組態管理系統提供了管控、合規和法規需求方面的保證。

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

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

資源

相關的最佳實務：

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

相關文件：

- AWS Control Tower
- AWS 登陸區域加速器

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

相關影片：

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

OPS05-BP04 使用建置和部署管理系統

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

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

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

常見的反模式：

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

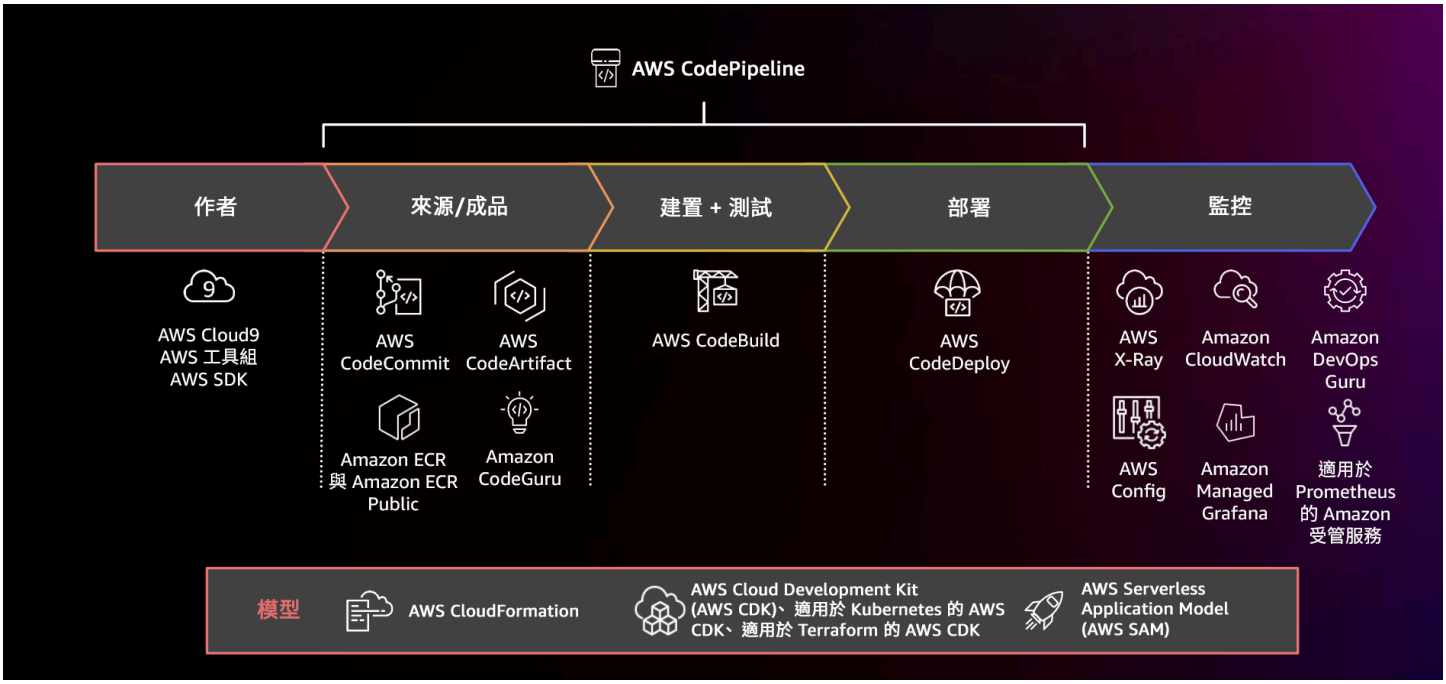
建立此最佳實務的優勢： 透過提供用於管理建置和部署活動的機制，您可以減少執行重複性任務的工作量，讓團隊成員專注於高價值的創意任務，並減少手動程序導致的錯誤。

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟



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

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

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

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

相關影片：

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

OPS05-BP05 執行修補程式管理

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

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

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

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

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

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

常見的反模式：

- 您必須在兩小時內套用所有新的安全修補程式，結果導致應用程式與修補程式不相容而發生多次停機。

- 未修補的程式庫導致意外後果發生，因為有不明對象利用其中的漏洞來存取您的工作負載。
- 您自動修補開發人員環境，而未通知開發人員。您收到來自開發人員的多次投訴，表示其環境如預期停止運作。
- 您尚未在持續執行的執行個體上修補商用現成軟體。當軟體發生問題而您聯絡廠商時，他們會通知您不支援該版本，您必須修補至特定程度才能獲得協助。
- 您使用的加密軟體近期發佈了修補程式，使效能獲得大幅改善。未修補的系統因未修補仍存在效能問題。
- 收到發生零時差漏洞的通知時，需緊急修正並手動修補所有環境。

建立此最佳實務的優勢： 透過建立修補程式管理程序 (包括修補準則和在各環境中散佈的方法)，您就能擴展和報告修補程度。這樣可保證修補過程安全無虞，並確保能清楚看見已知修正的狀態。如此可促進採用所需的功能、迅速消除問題，並持續遵循管控要求。實作修補程式管理系統和自動化，以減少部署修補程式的工作量，並限制手動程序引起的錯誤。

未建立此最佳實務時的曝險等級： 中

實作指引

修補系統以補救問題，獲得所需的功能，並保持符合管控政策和廠商支援需求。在不可變系統中，部署適當的修補程式集以實現所需的結果。自動化修補程式管理機制，以縮短修補時間、避免手動程序引起的錯誤，並減少修補工作量。

實作步驟

針對 Amazon EC2 Image Builder：

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

3. 選用 - 定義您的基礎設施組態。
4. 選用 - 定義組態設定。
5. 檢閱設定。
6. 定期維護配方乾淨度。

針對 Systems Manager Patch Manager：

1. 建立修補基準。
2. 選取路徑操作方法。
3. 啟用合規報告和掃描。

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

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

相關影片：

- AWS 上適用於無伺服器應用程式的 CI/CD
- 設計時考量 Ops

相關範例：

- Well-Architected 實驗室 - 庫存和修補程式管理

- AWS Systems Manager Patch Manager 教學課程

OPS05-BP06 共用設計標準

在團隊之間共用最佳實務，以提高認識並最大化開發工作的效益。記載它們並且隨著您的架構演進讓它們保持在最新狀態。如果您的組織中強制執行共用標準，則必須存在用於請求標準新增、變更及例外狀況的機制。如果沒有此選項，標準就會限制創新。

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

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

客戶範例

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

實作步驟

1. 識別擁有開發和更新設計標準的跨部門團隊。這個團隊應與整個組織的利害關係人合作，共同開發設計標準、操作程序、檢查清單、指引和管控需求。記錄設計標準並且在組織內共用。

   a. AWS Service Catalog 可以用來建立套裝服務，代表使用基礎設施即程式碼的設計標準。您可以與所有帳戶共用套裝服務。
2. 設立機制讓設計標準隨著新的最佳實務出現而保持在最新狀態。
3. 如果設計標準是集中強制執行，設立程序來請求變更、更新和豁免。

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

- AWS Service Catalog 參考架構

- AWS Service Catalog 研討會

相關服務：

- AWS Service Catalog

OPS05-BP07 實作用於提高程式碼品質的實務

實作實務以提高程式碼品質並將缺陷降至最少。部分範例包括測試驅動的開發、程式碼檢閱、標準採用和配對程式設計。將這些實務併入您的持續整合和交付程序。

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

常見的反模式：

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

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

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

- Agile 軟體指南
- CI/CD 管道是我的發行隊長
- 使用 Amazon CodeGuru Reviewer 自動化程式碼審查
- 採用測試驅動的開發方法
- DevFactory 如何使用 Amazon CodeGuru 組建更好的應用程式
- 關於配對程式設計
- RENGA Inc. 使用 Amazon CodeGuru 自動進行程式碼審查
- 敏捷開發的藝術：測試驅動的開發
- 程式碼檢閱為何重要 (而且確實可節省時間！)

相關影片：

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

相關服務：

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

OPS05-BP08 使用多個環境

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

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

常見的反模式：

- 您在共享開發環境中進行開發，而另一名開發人員覆寫您的程式碼變更。
- 對共享開發環境的限制性安全控制，讓您無法試驗新服務和功能。
- 您對生產系統執行負載測試，並給使用者造成停機。
- 在生產環境中發生導致資料遺失的嚴重錯誤。在您的生產環境中，您試圖重建導致資料遺失的條件，以便了解此情況如何發生，並防止再次發生。為防止更多資料在測試期間遺失，您必須讓使用者無法使用應用程式。
- 您正在操作多租用戶服務，且無法支援客戶對專用環境的要求。
- 您不一定會進行測試，但要測試時，您會在生產環境中進行。
- 您認為簡單的單一環境會覆寫環境內變更的影響範圍。

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

資源

相關文件：

- AWS 上的 Instance Scheduler
- 什麼是 AWS CloudFormation？

OPS05-BP09 進行頻繁、細微和可逆的變更

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

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 低

實作指引

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

資源

相關的最佳實務：

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

相關文件：

- 實作 AWS 上的微型服務
- 微型服務 - 可觀測性

OPS05-BP10 完全自動化整合和部署

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

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

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

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 低

實作指引

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

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

OPS06-BP01 為失敗變更進行規劃

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

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

常見的反模式：

- 您執行了部署，而您的應用程式變得不穩定，但系統中似乎有作用中使用者。您必須決定是否要復原變更並影響作用中使用者，或在知道使用者無論如何都會受到影響的情況下，等待復原變更。
- 在進行路由變更後，您可以存取新的環境，但其中一個子網路變成無法連線。您必須決定是否要復原所有項目，或嘗試修正無法存取的子網路。當您做出該決定時，子網路仍無法連線。
- 您的系統架構並不允許以較小版本進行更新。因此，在部署失敗期間，您無法回復這些大量變更。
- 您未使用基礎架構即程式碼 (IaC)，並且您以手動方式更新了基礎架構，而造成不理想的組態。您無法有效追蹤和還原手動變更。

- 由於您尚未測量部署的增加頻率，您的團隊不會想降低變更規模和改善每次變更的回復計畫，進而造成更高的風險和失敗率。
- 請不要測量因變更失敗而導致中斷的總持續時間。您的團隊無法排定優先順序並改善其部署程序和回復計畫的效能。

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

未建立此最佳實務時的曝險等級： 高

實作指引

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

實作步驟

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

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS06-BP04 自動化測試和復原

相關文件：

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

相關影片：

- re:Invent 2019 | Amazon 的高可用部署方式

OPS06-BP02 測試部署

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

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

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

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 高

實作指引

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

客戶範例

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

實作步驟

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

實作計劃的工作量： 高

資源

相關的最佳實務：

- OPS05-BP02 測試並驗證變更

相關文件：

- AWS 建置者資料中心 | 自動化安全、無人為介入的部署 | 測試部署
- AWS 白皮書 | 在 AWS 上實行持續整合和持續交付
- Apollo 的故事 - Amazon 部署引擎
- 在交付程式碼之前，如何在本機測試和偵錯 AWS CodeDeploy
- 整合網路連線能力測試和基礎設施部署

相關影片：

- re:Invent 2020 | 在 Amazon 測試軟體和系統

相關範例：

- 教學 | 具驗證測試的部署和 Amazon ECS 服務

OPS06-BP03 採用安全的部署策略

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

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

常見的反模式：

- 您一次性將失敗的變更部署至所有生產環境。因此，所有客戶會同時受影響。
- 同時部署到所有系統的負面影響必須以緊急版本因應。需要數天時間為所有客戶進行錯誤修正。

- 管理生產發行需要多個團隊的規畫和參與。這會限制您為客戶積極提供更新功能的能力。
- 您透過修改現有系統來執行可變部署。發現變更失敗之後，您必須再次修改系統以還原舊版本，這會延長回復時間。

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

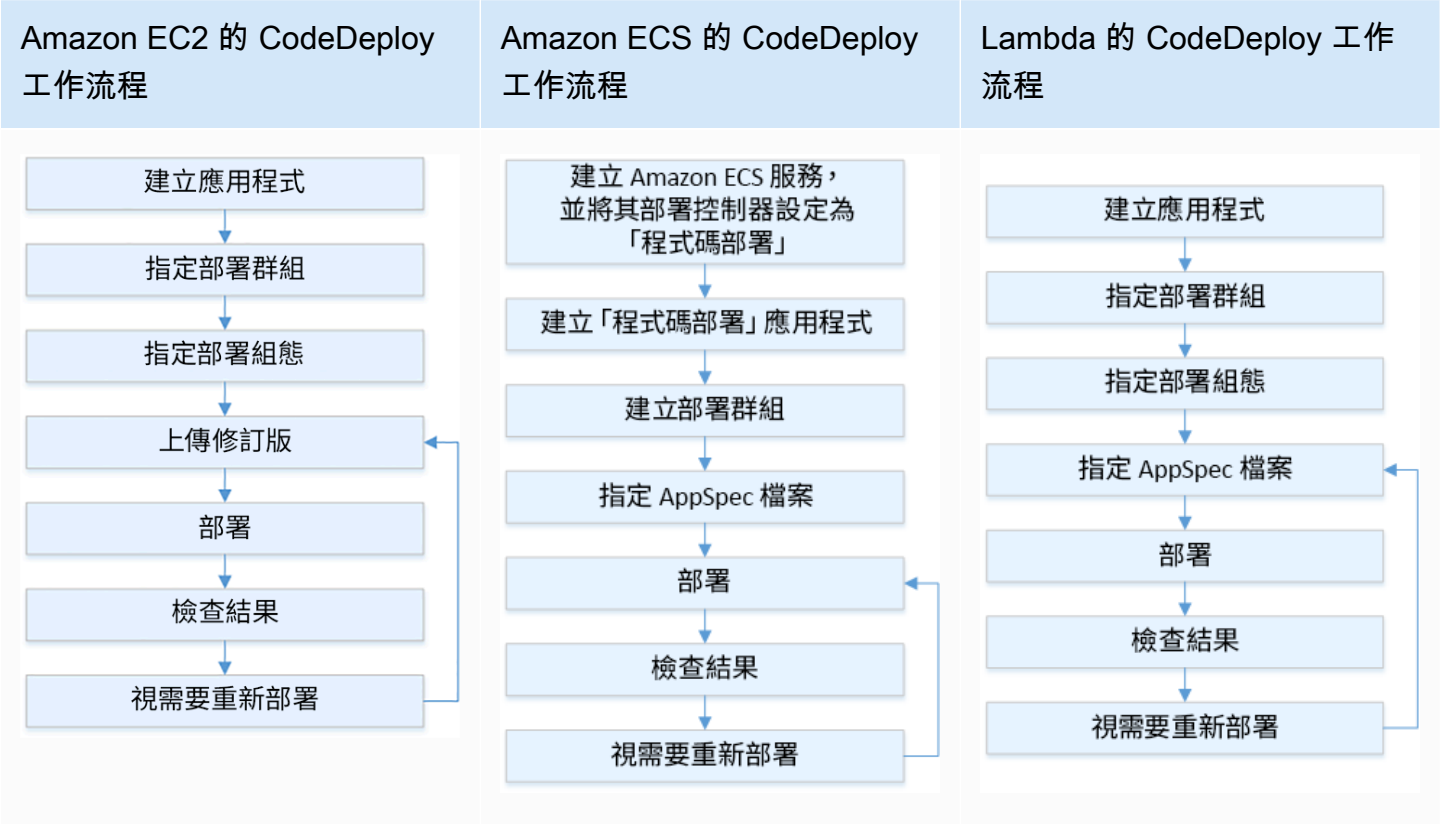
未建立此最佳實務時的曝險等級： 中

實作指引

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

客戶範例

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



實作步驟

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

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS05-BP02 測試並驗證變更
- OPS05-BP09 進行頻繁、細微和可逆的變更
- OPS05-BP10 完全自動化整合和部署

相關文件：

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

相關影片：

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

相關範例：

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

OPS06-BP04 自動化測試和復原

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

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

常見的反模式：

- 您的系統架構並不允許以較小版本進行更新。因此，在部署失敗期間，您無法回復這些大量變更。
- 您的部署程序包含一系列手動步驟。將變更部署到工作負載之後，即可開始部署後測試。測試之後，您會發現工作負載無法運作，且客戶中斷連線。然後您開始回復到之前的版本。所有這些手動步驟都會延遲整體系統回復，並對客戶造成長期影響。
- 您花時間為應用程式中不常使用的功能開發自動化測試案例，因而大幅降低了自動化測試功能的投資報酬率。
- 您的版本包含彼此獨立的應用程式、基礎設施、修補程式和組態更新。但是，您有一個 CI/CD 管道可以一次交付所有變更。一個元件故障會強迫您還原所有變更，進而使回復過程變得複雜且效率低下。

- 您的團隊在第一個衝刺階段完成編碼，並開始衝刺兩項工作，但直到第三個衝刺階段，計畫中都不包括測試。最終，自動化測試找出第一個衝刺階段的缺漏，必須在測試第二個衝刺階段前解決，才能啟動交付項目，因此整個版本延遲，進而降低您的自動化測試效率。
- 生產版本的自動迴歸測試案例已經完成，但您並未監控工作負載的運作狀況。由於無法查看是否已重啟服務，您不確定是否需要回復或已啟動回復。

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

1. 為您的開發生命週期建立測試生命週期，定義需求規劃到測試案例開發、工具配置、自動化測試和測試案例結案等每個測試程序階段。
   a. 根據您的整體測試策略建立針對特定工作負載的測試方式。
   b. 在整個開發生命週期中，考慮適當的連續測試策略。
2. 根據您的業務需求和管道投資，選擇用於測試和回復的自動化工具。
3. 決定您應該分別自動化和手動執行哪些測試案例。這些內容皆可以根據受測功能的業務價值優先順序來決定。使所有團隊成員隨時接收計畫最新資訊，並確認執行手動測試的權責分配。
   a. 將自動化測試功能應用於對自動化有意義的特定測試案例，例如可重複或經常執行的案例、需要重複作業的案例，或跨多個組態所需的案例。
   b. 在自動化工具中定義測試自動化指令碼和成功條件，如此一來，當特定案例失敗時，可以啟動持續的工作流程自動化。
   c. 定義自動回復的特定失敗條件。
4. 測試案例其中複雜度和人工互動具較高的失敗風險，因此必須排定測試自動化的優先順序，透過詳盡的測試案例開發來產生一致的結果。

5. 將您的自動化測試和回復工具整合到 CI/CD 管道。
   a. 為變更制定明確的成功條件。
   b. 監控觀察以偵測這些條件，並在符合特定回復條件時自動回復變更。
6. 執行不同類型的自動化生產測試，例如：
   a. A/B 測試，以顯示結果比較兩個使用者測試組之間的當前版本。
   b. 金絲雀測試讓您能在將變更發佈給所有使用者之前，先將其發佈給一部分使用者。
   c. 功能旗標測試允許您從應用程序外部標記新版本的功能 (每次僅限一個)，進而使每個新功能皆能逐一進行驗證。
   d. 迴歸測試，以現有的關聯元件驗證新功能。
7. 透過其他應用程式和元件，監控應用程式、交易和互動的操作面向。開發報告，以便按照部銅工作負載顯示變更成功率，讓您得以識別出能夠進一步最佳化的自動化和工作流程部分。
   a. 開發測試結果報告，協助您快速決定是否應該調用回復程序。
   b. 實施策略，允許根據一個或多個測試方法導出的預定義失敗條件進行自動回復。
8. 開發自動化測試用例，以便在未來可重複的變更中重複使用。

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

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

相關範例：

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

相關影片：

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

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 使用執行手冊執行程序
- OPS07-BP04 使用程序手冊來調查問題
- OPS07-BP05 做出部署系統和變更的明智決策
- OPS07-BP06 啟用生產工作負載的支援計劃

OPS07-BP01 確保人員能力

建立一種機制，用於驗證您有適當數量受過培訓的人員來支援工作負載。他們必須受過組成您的工作負載的平台和服務的培訓。為他們提供操作工作負載所需的知識。您必須擁有足夠受過培訓的人員，才能支援工作負載的一般操作，並且針對會發生的任何事件進行疑難排解。擁有足夠的人員，以便您可以輪替待命和休假的人員，避免倦怠。

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

- 擁有熟練的團隊成員可有效支援您的工作負載。
- 具有足夠的團隊成員，您可以支援工作負載和待命輪替，同時降低倦怠風險。

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

- AWS 活動和研討會
- AWS 培訓與認證

OPS07-BP02 確保對營運準備度進行一致的審查

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

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

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

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

常見的反模式：

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

建立此最佳實務的優勢：

- 工作負載包含架構、程序和管理最佳實務。
- 經驗已納入 ORR 程序中。

- 工作負載啟動時，已設置必要的程序。
- ORR 會在工作負載的整個軟體生命週期執行。

若未建立此最佳實務的風險等級： 高

實作指引

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

客戶範例

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

實作步驟

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

1. 召集關鍵利害關係人，包含安全性、營運和開發等團隊的代表人員。
2. 請每位利害關係人提供至少一個需求。對於第一次的反覆測試，請嘗試將項目數限制在三十個以下。
   - 附錄 B：來自「營運準備度審查 (ORR)」白皮書的 ORR 問題範例包含您可以開始使用的範例問題。
3. 將需求集中放在試算表中。
   - 您可以使用 在 AWS Well-Architected Tool 中 使用自訂聚焦 來制定 ORR 並在帳戶和 AWS 組織之間進行共用。
4. 找出要在其中執行 ORR 的一個工作負載。啟動前的工作負載或內部工作負載是理想的選擇。
5. 演練 ORR 檢查清單，並記下任何所探索的項目。如果採取緩解措施，那就可能無法進行探索。對於缺少緩解措施的任何探索，請將那些探索新增至項目的待辦清單中，然後在啟動前加以實作。

6. 隨著時間持續在 ORR 檢查清單中新增最佳實務和需求。

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

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

- 營運準備度審查 (ORR) 聚焦範例

相關服務：

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

OPS07-BP03 使用執行手冊執行程序

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

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

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

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

常見的反模式：

- 憑藉記憶完成程序中的每個步驟。
- 手動部署變更而不使用檢查清單。
- 不同的團隊成員執行相同程序，但使用的步驟不同，或結果不同。
- 執行手冊失去與系統變更和自動化的同步。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

隨著組織逐漸成熟，您的文字執行手冊應該要自動化。您可以使用 AWS Systems Manager 自動化等服務，將平面文字轉換為可以根據工作負載執行的自動化程序。這些自動化可作為事件的應變動作來執行，以降低您維持工作負載的操作負擔。AWS Systems Manager 自動化還提供低程式碼的 視覺設計體驗，以更輕鬆地建立自動化執行手冊。

客戶範例

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

實作步驟

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

```
# Runbook Title
## Runbook Info
| Runbook ID | Description | Tools Used | Special Permissions | Runbook Author | Last
 Updated | Escalation POC |
|-------|-------|-------|-------|-------|-------|-------|
| RUN001 | What is this runbook for? What is the desired outcome? | Tools | Permissions
 | Your Name | 2022-09-21 | Escalation Name |
```

```
## Steps
1. Step one
2. Step two
```

1. 如果您沒有現有的文件儲存庫或 Wiki，請在您的版本控制系統中建立新的版本控制儲存庫。
2. 識別沒有執行手冊的程序。經常執行、步驟數較少，且失敗的影響程度不高的程序，就是理想的程序。
3. 在您的文件儲存庫中，使用範本建立新的草稿 Markdown 文件。填寫「執行手冊標題」和「執行手冊資訊」下的必要欄位。
4. 從第一個步驟開始，填寫執行手冊的「步驟」部分。
5. 將執行手冊提供給團隊成員。讓他們使用執行手冊來驗證步驟。如有任何事項缺漏或需要釐清，請更新執行手冊。
6. 將執行手冊發佈至您的內部文件存放區。發佈後，請告知團隊和其他利害關係人。
7. 一段時間後，您會建置執行手冊程式庫。隨著該程式庫的擴增，您應開始設法將執行手冊自動化。

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

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者
- OPS07-BP04 使用程序手冊來調查問題
- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP02 每個提醒建立一個流程
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected 架構：概念：執行手冊研製
- 使用自動化的程序手冊和執行手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- 用於 AWS 大型遷移的遷移執行手冊 - 任務 4：改進您的遷移執行手冊

- 使用 AWS Systems Manager 自動化執行手冊完成營運任務

相關影片：

- AWS re:Invent 2019：執行手冊、事故報告和事故應變的 DIY 指南
- 如何在 AWS 上將 IT 作業自動化 | Amazon Web Services
- 將指令碼整合到 AWS Systems Manager 中

相關範例：

- Well-Architected 實驗室：使用程序手冊和執行手冊將操作自動化
- AWS 部落格文章：建立雲端自動化實務以實現卓越營運：AWS Managed Services 的最佳實務
- AWS Systems Manager：自動化演練
- AWS Systems Manager：從最新的快照執行手冊還原根磁碟區
- 使用 Jupyter 筆記本和 CloudTrail Lake 建置 AWS 事故應變執行手冊
- Gitlab - 執行手冊
- Rubix - 用來在 Jupyter 筆記本中建置執行手冊的 Python 程式庫
- 使用 Document Builder 建立自訂執行手冊

相關服務：

- AWS Systems Manager 自動化

OPS07-BP04 使用程序手冊來調查問題

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

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

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

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

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

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

客戶範例

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

實作步驟

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

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

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

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

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者
- OPS07-BP03 使用執行手冊執行程序
- OPS10-BP01 使用程序進行事件、事故和問題管理
- OPS10-BP02 每個提醒建立一個流程
- OPS11-BP04 執行知識管理

相關文件：

- AWS Well-Architected 架構：概念：程序手冊開發研製
- 使用自動化的程序手冊和執行手冊達成卓越營運
- AWS Systems Manager：使用執行手冊
- 使用 AWS Systems Manager 自動化執行手冊完成營運任務

相關影片：

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

相關範例：

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

- Well-Architected 實驗室：使用 Jupyter 事件應變程序手冊

相關服務：

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

OPS07-BP05 做出部署系統和變更的明智決策

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

預期成果：

- 您在將變更部署到您的工作負載時做出明智決策。
- 變更符合管控。

常見的反模式：

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

建立此最佳實務的優勢：

- 您對工作負載的失敗變更已做好準備。
- 變更您的工作負載符合管控政策。

未建立此最佳實務時的風險暴露等級：低

實作指引

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

客戶範例

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

實作步驟

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

實作計劃的工作量：中。實作事前剖析的實務需要貴組織利害關係人的協調和努力

資源

相關的最佳實務：

- OPS01-BP03 評估管控要求 - 管控要求是判斷是否部署變更的關鍵因素。
- OPS06-BP01 為失敗變更進行規劃 - 建立計劃來緩解失敗的部署並且使用事前剖析來驗證它們。
- OPS06-BP02 測試部署 - 每個軟體變更都應該在部署之前先適當的進行測試，以便在生產中減少缺陷。
- OPS07-BP01 確保人員能力 - 擁有支援工作負載的足夠受過培訓的人員，對於為部署系統變更做出明智決策相當重要。

相關文件：

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

OPS07-BP06 啟用生產工作負載的支援計劃

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

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：低

實作指引

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

客戶範例

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

實作步驟

1. 與組織中的利害關係人合作，識別您的工作負載所依賴的軟體和服務供應商。記錄這些相依性。
2. 確認工作負載的服務層級需求。選取相對應的支援計劃。

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

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

資源

相關的最佳實務：

- OPS02-BP02 流程和程序已確認擁有者

相關文件：

- AWS Support Plans

相關服務：

- AWS Business Support
- AWS Enterprise Support

營運

成功是業務成果的實現，並根據您定義的指標來衡量。透過了解工作負載和營運運作狀態，您可以確定組織和業務成果何時可能處於風險狀態或目前正處於風險狀態，並適當地做出回應。

為取得成功，必須能夠：

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。透過專注於基本洞見並消除不必要的資料，您就能持續使用簡單直接的方式來了解工作負載效能。

重點不只是收集資料，還要正確解譯資料。定義清楚的基準、設定適當的警示閾值，並主動監控任何偏差情況。一旦關鍵指標稍有變化，尤其是與其他資料相關時，就能精確指出特定問題所在。

有了可觀測性，您就具備更優異的預測能力，並且能應付潛在的挑戰，進而確保工作負載順利運行並滿足業務需求。

AWS 提供特定的工具，如 Amazon CloudWatch ，可用於監控和記錄，以及 AWS X-Ray ，可用於分散式追蹤。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警示，以及在儀表板上呈現資料以便進行解讀。利用這些洞見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 建立可付諸行動的警示

- OPS08-BP05 建立儀表板

OPS08-BP01 分析工作負載指標

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

1. 分析與檢閱： 定期檢閱和解讀您的工作負載指標。
   a. 將業務成果指標視為優先於純粹技術指標的事項。
   b. 了解資料中峰值、下降或模式的重要性。
2. 利用 Amazon CloudWatch： 使用 Amazon CloudWatch 集中檢視並進行深入分析。

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

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

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

OPS08-BP02 分析工作負載日誌

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

實作步驟

1. 設定 CloudWatch Logs：應用程式和服務以將日誌傳送至 CloudWatch Logs。
2. 使用日誌異常偵測：利用 Amazon CloudWatch Logs 異常偵測自動識別不尋常日誌模式並予以警示。此工具可協助您主動管理日誌檔中的異常狀況，並及早偵測潛在的問題。

3. 設定 CloudWatch Logs Insights：使用 CloudWatch Logs Insights，以互動方式搜尋和分析日誌資料。
   a. 製作查詢以找出模式、視覺化日誌資料，並產生可付諸行動的洞見。
   b. 使用 CloudWatch Logs Insights 模式分析來分析和視覺化頻繁的日誌模式。此功能可協助您了解日誌資料中常見的運作趨勢和潛在異常值。
   c. 使用 CloudWatch Logs 比較 (diff) 在不同時間週期之間，或在不同日誌群組之間執行差異分析。使用此功能精確找出變更，並評估其對系統效能或行為的影響。
4. 使用 Live Tail 即時監控日誌：使用 Amazon CloudWatch Logs Live Tail 即時查看日誌資料。您可以主動監控應用程式的運作活動，從中了解系統效能和潛在問題。
5. 利用 Contributor Insights：使用 CloudWatch Contributor Insights 識別 IP 位址或使用者代理程式等高基數維度中的最高用量者。
6. 實作 CloudWatch Logs 指標篩選條件：設定 CloudWatch Logs 指標篩選條件，將日誌資料轉換為可操作的指標。如此您就能設定警報或進一步分析模式。
7. 實作 CloudWatch 跨帳戶可觀測性：監控和疑難排解區域內跨多個帳戶的應用程式。
8. 定期檢閱和改進：定期檢閱您的日誌分析策略，以擷取所有相關資訊並持續最佳化應用程式效能。

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

- 使用 CloudWatch Logs Insights 分析日誌資料

- 使用 CloudWatch Contributor Insights 分析高基數資料

相關範例：

- CloudWatch Logs 範例查詢
- One Observability 研討會

OPS08-BP03 分析工作負載追蹤

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

實作步驟

下列步驟提供了結構化的方法，以使用 AWS 服務有效實作追蹤資料分析：

1. 整合 AWS X-Ray：確保 X-Ray 與您的應用程式整合以擷取追蹤資料。
2. 分析 X-Ray 指標：使用服務地圖監控應用程式健全狀態，深入了解從 X-Ray 追蹤衍生的指標，例如延遲、請求率、故障率和回應時間分佈。
3. 使用 ServiceLens：利用服務地圖來增強服務和應用程式的可觀測性。如此就能將追蹤、指標、日誌、警報和其他運作狀況資訊整合在一起檢視。
4. 啟用 X-Ray Insights：
   a. 開啟 X-Ray Insights 以在追蹤中自動偵測異常狀況。
   b. 檢查洞見以找出明確的模式並確定根本原因，例如故障率或延遲增加。
   c. 請參考洞察時間軸，依時間順序查看所偵測到問題的分析。
5. 使用 X-Ray Analytics：X-Ray Analytics 可讓您徹底探索追蹤資料、找出明確的模式，以及擷取洞見。
6. 使用 X-Ray 中的群組：在 X-Ray 中建立群組，即可根據如高延遲等條件篩選追蹤，以進行更針對性的分析。
7. 併入 Amazon DevOps Guru：參與 Amazon DevOps Guru 以便利用機器學習模型的優勢，從追蹤中找出明確的操作異常狀況。
8. 使用 CloudWatch Synthetics：使用 CloudWatch Synthetics建立可持續監控端點和工作流程的 Canary。這些 Canary 可與 X-Ray 整合，以提供追蹤資料，用來對要測試的應用程式進行深入分析。
9. 使用真實使用者監控 (RUM)：使用 AWS X-Ray 和 CloudWatch RUM，您可以透過下游受 AWS 管理服務，從應用程式的最終使用者開始分析和除錯請求路徑。這樣做有助於找出影響最終使用者的延遲趨勢和錯誤。
10. 與日誌檔關聯：將 追蹤資料與 X-Ray 追蹤檢視中的相關日誌相關聯，以獲得應用程式行為的精細視角。如此可讓您檢視與追蹤的交易直接相關的日誌事件。
11. 實作 CloudWatch 跨帳戶可觀測性：監控和疑難排解區域內跨多個帳戶的應用程式。

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

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

OPS08-BP04 建立可付諸行動的警示

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

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

常見的反模式：

- 設定太多非嚴重警示，導致警示疲勞。
- 未根據 KPI 排定警示的優先順序，因此難以了解問題對業務造成的影響。

- 忽略解決根本原因，導致一再出現相同問題的警示。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

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

6. 與提醒工具整合：結合 Ops Genie 和 PagerDuty 等工具。
7. 與 AWS Chatbot「接合 」：整合 AWS Chatbot 以將警示轉送到 Amazon Chime、Microsoft Teams 和 Slack。
8. 基於日誌的提醒：使用 CloudWatch 中的日誌指標篩選條件，根據特定日誌事件建立警示。
9. 檢閱和迭代：定期重新檢視並改善警示組態。

實作計畫的工作量：中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- OPS04-BP03 實作使用者體驗遙測
- OPS04-BP04 實作相依性遙測
- OPS04-BP05 實作分散式追蹤
- OPS08-BP01 分析工作負載指標
- OPS08-BP02 分析工作負載日誌
- OPS08-BP03 分析工作負載追蹤

相關文件：

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

相關影片：

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

相關範例：

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

OPS08-BP05 建立儀表板

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

期望的結果：

使用視覺呈現的方式，提供清楚、深入系統與業務運作狀況且可付諸行動的洞見。

常見的反模式：

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

本最佳實務的優勢：

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

未建立此最佳實務時的風險等級：中

實作指引

以業務為中心的儀表板

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

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

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

實作步驟

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

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

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，然後使用指標並根據營運成果制定指標，以獲得有用的洞見。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獲得洞見。

最佳實務

- OPS09-BP01 使用指標衡量營運目標與 KPI
- OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況
- OPS09-BP03 檢閱營運指標並優先改進

OPS09-BP01 使用指標衡量營運目標與 KPI

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

預期成果：

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

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

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

資源

相關文件：

- Amazon QuickSight - 使用 KPI
- Amazon CloudWatch - 使用指標
- 建置儀表板
- 如何使用 KPI 儀表板追蹤成本最佳化 KPI

OPS09-BP02 傳達狀態和趨勢以確實掌控營運狀況

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

預期成果：

- 主管對團隊處理的通話量類型和正在進行的工作 (例如部署)可以一目瞭然。
- 發生影響擴及正常營運的情況時，利害關係人和使用者社群就會收到警示。

- 組織領導階層和利害關係人可查看狀態頁面以便回應警示或影響，並且獲得有關營運事件的資訊，例如聯絡窗口、票證資訊及預估的復原時間。
- 領導階層和其他利害關係人會收到報告，報告中會顯示營運統計資料，例如某一段時間內的通話量、使用者滿意度分數、待處理票證數量及其存留時間。

常見的反模式：

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

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

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

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

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

資源

相關文件：

- 測量進度

- 建置用於檢視營運狀況的儀表板

相關解決方案：

- 資料操作

OPS09-BP03 檢閱營運指標並優先改進

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

預期成果：

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

常見的反模式：

- 新產品已推出，但 1 級和 2 級營運團隊未接受足夠的培訓來提供支援，或未配置額外的人員。領導者未看見指出支援單解決次數減少且事故量增加的指標。訂閱數量隨著不滿的使用者離開平台而開始減少，但數週後才採取行動。
- 長久以來一直採用手動程序來執行工作負載維護工作。雖然渴望自動化，但由於系統的重要性較低，因此優先順序較低。然而經過一段時間後，系統的重要性已提高，而現在這些手動程序佔用了大多數營運時間。未安排資源來提供更多營運工具，導致員工隨著工作負載增加而倦怠。等到有員工離職並加入其他競爭對手，領導階層才察覺到此情況。

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

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

資源

相關文件：

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

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 根據業務影響確定營運事件的優先順序
- OPS10-BP04 定義向上呈報路徑
- OPS10-BP05 定義處理影響服務事件的客戶通訊計畫
- OPS10-BP06 透過儀表板傳達狀態
- OPS10-BP07 自動回應事件

OPS10-BP01 使用程序進行事件、事故和問題管理

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 高

實作指引

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

了解事件、事故和問題

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

實作步驟

事件

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

事故

1. 回應事故：
   - 利用可觀測性工具的洞察，快速識別事故並進行回應。
   - 實作 AWS Systems Manager 操作中心 以彙總、組織及排序營運項目和事件的優先順序。
   - 使用多項服務，例如 Amazon CloudWatch 和 AWS X-Ray 進行更深入的分析和疑難排解。
   - 考慮 AWS Managed Services (AMS) 利用其主動、預防和偵測功能，進而增強事件管理。AMS 會擴展營運支援，過程中透過監控、事件偵測和回應以及安全管理等服務。
   - Enterprise Support 客戶可以要求 AWS 事件偵測與回應，為生產工作負載提供持續主動監控和事件管理。
2. 建立事件管理程序：

- 建立結構化事件管理程序，包括清晰的角色、通訊協定和解決步驟。
- 整合事件管理與多項工具，像是 AWS Chatbot 達到快速的回應和協調。
- 根據嚴重性進行事件分類，且預先定義 每個類別適用的 事件回應計畫。

3. 學習和改善：
   - 舉行 事件後分析 了解根本原因和解決有效成果。
   - 根據檢閱和不斷變化的實務，持續更新和改善回應計畫。
   - 記錄並分享跨團隊所獲得的經驗，以增強營運韌性。
   - Enterprise Support 客戶可以要求 事件管理研討會 (透過其技術客戶經理)。這個指導研討會將測試您現有的事故應變計劃，並協助您識別改善的領域。

問題

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

實作計劃的工作量： 中

資源

相關的最佳實務：

- OPS04-BP01 識別關鍵績效指標
- OPS04-BP02 實作應用程式遙測
- OPS07-BP03 使用執行手冊執行程序
- OPS07-BP04 使用程序手冊來調查問題
- OPS08-BP01 分析工作負載指標
- OPS11-BP02 執行事件後分析

相關文件：

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

相關影片：

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

相關範例：

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

相關服務：

- Amazon EventBridge

OPS10-BP02 每個提醒建立一個程序

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 高

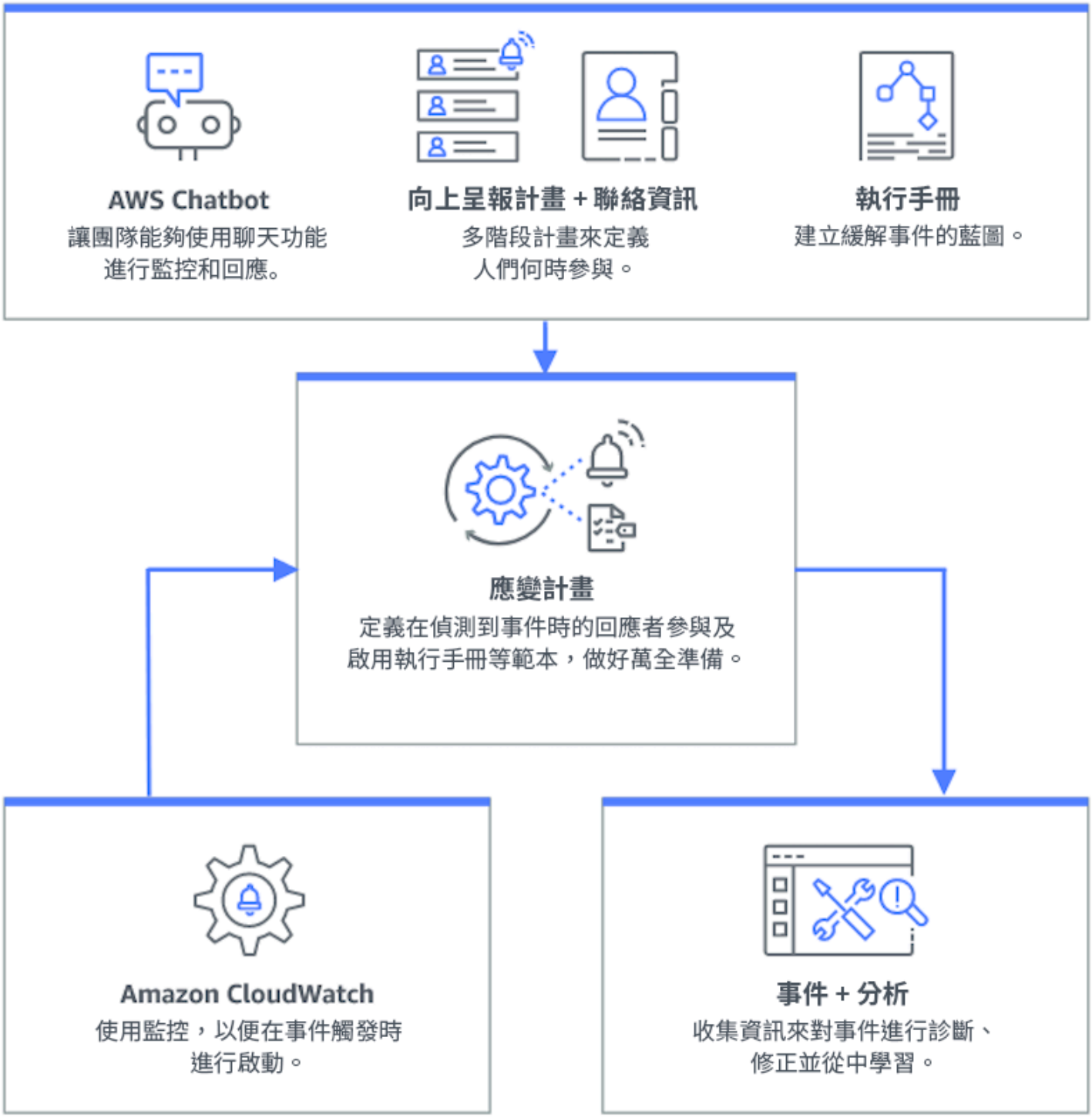
實作指引

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

實作步驟

下圖說明事件管理工作流程於 AWS Systems Manager Incident Manager。它可快速回應營運相關問題，期間透過自動建立事件，以便回應來自 Amazon CloudWatch 或者 Amazon EventBridge。建

立事件之後，Incident Manager 可能以自動或手動方式，集中管理事件、組織相關 AWS 資源資訊，並啟動預定義的回應計畫。這包括執行 Systems Manager 自動化執行手冊以立即採取行動，同時在 OpsCenter 中建立上層營運工作項目，以利追蹤相關的工作和分析。這個簡化程序加快並協調在您整體 AWS 環境中的所有事件回應。



1. 使用複合警報： 建立 複合警報 於 CloudWatch，以便分組相關警報，減少雜訊，並做出更有意義的回應。
2. 整合 Amazon CloudWatch 警報與 Incident Manager 設定 CloudWatch 警報以自動建立事件於 AWS Systems Manager Incident Manager。
3. 整合 Amazon EventBridge 與 Incident Manager： 建立 EventBridge 個規則 以回應事件，並建立使用定義回應計畫的事件。
4. 為 Incident Manager 中的事件做準備：
   - 建立詳細的 回應計畫 於 Incident Manager 中，以因應各種提醒類型。
   - 建立聊天管道經由 AWS Chatbot 已連接到 Incident Manager 中的回應計畫，促進在 Slack、Microsoft Teams 和 Amazon Chime 等平台之間發生事件的即時通訊。
   - 納入 Systems Manager 自動化執行手冊 於 Incident Manager 以推動自動化事件回應。

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

- 常用事件回應提示來自 AWS

相關範例：

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

OPS10-BP03 根據業務影響確定營運事件的優先順序

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

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

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

影響層級	描述
高	影響許多員工或客戶，高度財務影響、高度聲譽受損或傷害。

影響層級	描述
中	影響特定群組的員工或客戶，中度財務影響或中度聲譽受損。
低	影響個別員工或客戶，低度財務影響或低度聲譽受損。

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

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

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

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

4. 訓練和溝通： 訓練回應團隊認識優先順序矩陣，以及在事件發生期間遵循矩陣的重要性。向所有利益相關者傳達優先順序，確定明確的期望。
5. 整合事件回應：

- 將優先順序矩陣納入您的事件回應計畫和工具。
- 在可能情況下，自動化事件的分類和優先順序，加快回應時間。
- 企業支援客戶可以利用 AWS 事件偵測與回應，為生產工作負載提供全年無休 24x7 的主動監控和事件管理。

6. 檢閱和適應： 定期檢閱優先順序程序的有效性，並根據業務環境的意見回應和變化進行調整。

資源

相關的最佳實務：

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

相關文件：

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

OPS10-BP04 定義向上呈報路徑

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

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

常見的反模式：

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

建立此最佳實務的優勢：

- 簡化的事件回應會透過預先定義的向上呈報程序。

- 縮短停機時間，期間搭配預先核准的動作和清晰的擁有權。
- 根據事件嚴重性，改善資源分配和支援層級調整。
- 改善與利益相關者和客戶的溝通。

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

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

8. 持續改善： 定期檢閱向上呈報路徑的有效性。根據事件事後分析獲得的經驗與持續意見反映，更新您的程序。

實作計劃的工作量： 中

資源

相關的最佳實務：

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

相關文件：

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

OPS10-BP05 定義處理影響服務事件的客戶通訊計畫

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

預期成果：

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

常見的反模式：

- 通訊不足或延遲會導致客戶混淆和心情不悅。
- 訊息過於技術性或模糊，無法傳達使用者受到的實際影響。

- 沒有預先定義的通訊策略，導致訊息不一致和反應性不佳。

建立此最佳實務的優勢：

- 透過主動和清晰的通訊，增強客戶信任感和滿意度。
- 透過預先解決客戶的疑慮，減少支援團隊的負擔。
- 提高可有效管理事件和從中復原的能力。

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

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

4. 協調內部通訊： 實作內部協定，使用像是 AWS Chatbot 等工具進行團隊協調和通訊。使用 CloudWatch 儀表板以傳達狀態。
5. 協調通訊，期間搭配專用工具和服務：
   - 使用 AWS Systems Manager Incident Manager 搭配 AWS Chatbot，並設定專用聊天通道，以便在事件期間即時進行內部通訊和協調。
   - 使用 AWS Systems Manager Incident Manager 執行手冊，在事件期間，自動透過 Amazon Pinpoint、Amazon SNS，或社群媒體平台等第三方工具處理客戶通知。
   - 在執行手冊中納入核准工作流程，即可選擇性地檢閱，並在授權所有外部通訊之後，再進行傳送。
6. 實務和改善：
   - 舉行有關使用通訊工具和策略的訓練。增強團隊能力，使其能夠在事件期間及時做出決定。
   - 透過定期鑽研或演練日測試通訊計畫。使用這些測試來完善訊息，並評估通道的有效性。
   - 實作意見反映機制，以便評估事件期間的通訊效果。持續根據意見反映和不斷變化的需求來改善通訊計畫。

實作計劃的工作量： 高

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

- Atlassian - 建立您自己的事件通訊計畫：事件範本

相關範例：

- AWS Health 儀表板
- 範例 AWS 狀態更新

OPS10-BP06 透過儀表板傳達狀態

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

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 中

實作指引

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

實作步驟

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

資源

相關的最佳實務：

- OPS08-BP05 建立儀表板

相關文件：

- 建置用於檢視營運能見度的儀表板
- 使用 Amazon CloudWatch 儀表板
- 使用儀表板變數建立靈活的儀表板

- 共用 CloudWatch 儀表板
- 查詢來自其他資料來源的指標
- 新增自訂小工具至 CloudWatch 儀表板

相關範例：

- One Observability 研討會 - 儀表板

OPS10-BP07 自動回應事件

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

預期成果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的曝險等級： 中

實作指引

納入自動化以建立有效營運工作流程，並將手動介入降到最低。

實作步驟

1. 識別自動化機會： 判斷重複的自動化工作，例如問題修復、票證增強功能、容量管理、擴展、部署和測試。
2. 識別自動化命令：
   - 評估並定義可啟動自動回應的特定條件或指標，過程中搭配 Amazon CloudWatch 警報動作。
   - 使用 Amazon EventBridge 以回應 AWS 服務、自訂工作負載和 SaaS 應用程式中的事件。
   - 考慮啟動事件，例如 特定日誌項目、效能指標臨界值，或 狀態變化 (關於 AWS 資源)。
3. 實作事件驅動型自動化：
   - 使用 AWS Systems Manager 自動化執行手冊，簡化維護、部署和修復工作。
   - 建立事件於 Incident Manager 自動收集，並將相關 AWS 資源的詳細資訊新增到事件中。
   - 主動監控配額，透過使用 AWS 的配額監視器。
   - 自動調整容量，搭配 AWS Auto Scaling 以維持可用性和效能。
   - 自動化開發管道，搭配 Amazon CodeCatalyst。
   - 煙霧測試，或持續監控端點和 API， 過程中使用綜合監控。
4. 透過自動化執行風險緩解：
   - 實作 自動化安全性回應 迅速解決風險。
   - 使用 AWS Systems Manager 狀態管理員 以減少組態偏離。
   - 使用 AWS Config 規則 修復不合規的資源。

實作計劃的工作量： 高

資源

相關的最佳實務：

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

相關文件：

- 使用系統管理員自動化執行手冊搭配 Incident Manager
- 在 Incident Manager 中建立事件
- AWS Service Quotas

- 監控資源用量，並在接近配額時傳送通知
- AWS Auto Scaling
- 什麼是 Amazon CodeCatalyst？
- 使用 Amazon CloudWatch 警報
- 使用 Amazon CloudWatch 警報動作
- 依 AWS Config 規則 修補不合規的資源
- 使用篩選條件從日誌事件建立指標
- AWS Systems Manager 狀態管理員

相關影片：

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

相關範例：

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

演進

演進是隨著時間的推移持續改善的過程。根據從營運活動中獲得的經驗，實作頻繁的小量循序漸進的變更，並評估其是否成功帶來改善。

若要讓營運隨著時間演進，您必須能夠：

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 執行事件後分析
- OPS11-BP03 實作回饋迴圈
- OPS11-BP04 執行知識管理
- OPS11-BP05 定義改進驅動因素
- OPS11-BP06 驗證洞見
- OPS11-BP07 執行營運指標審查
- OPS11-BP08 記錄和分享獲得的經驗
- OPS11-BP09 分配改進時間

OPS11-BP01 建立持續改進程序

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

期望的結果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

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

相關影片：

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

相關範例：

- AWS Well-Architected Tool

OPS11-BP02 執行事件後分析

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

期望的結果：

- 您已建立包括事件後分析的事件管理程序。
- 您已實施可觀測性計畫以收集事件相關資料。
- 有了這些資料，您可以了解並收集能支援事件後分析程序的指標。

- 您會從事件中學習經驗，進一步改善未來成果。

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

實作步驟

1. 收集像是部署變更、設定變更、事件開始時間、警報時間、人員介入時間、緩解開始時間和事件解決時間等指標。
2. 描述在時間軸上的關鍵時間點，以便目標對象了解事件發生的相關活動。
3. 提出下列問題：
   a. 您是否能改善偵測時間？
   b. 指標和警報是否有任何更新而能盡快偵測出事件？
   c. 您是否能改善診斷時間？
   d. 您的回應計畫或升級計畫是否有更新而能讓適當回應人員盡快介入？
   e. 您是否能改善緩解時間？

f. 是否有任何執行手冊或教戰手冊步驟可讓您新增或加以改進？

g. 您是否能防止未來事件發生？

4. 建立檢查清單和措施。追蹤與交付所有措施。

實作計畫的工作量：中

資源

相關的最佳實務：

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

相關文件：

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

OPS11-BP03 實作回饋迴圈

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

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

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

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

常見的反模式：

- 您推出新功能，但沒有辦法收到客戶對該功能的回饋。

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

建立此最佳實務的優勢：

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

若未建立此最佳實務，暴露的風險等級： 高

實作指引

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

客戶範例

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

實作步驟

1. 即時回饋
   - 您需要制定機制來接收來自客戶和團隊成員的回饋。您也可以設定營運活動來提供自動化的回饋。
   - 組織需要制定程序來審查此回饋、判斷需要改善的項目，並安排改善項目。
   - 您必須將回饋新增至軟體開發程序。
   - 在您著手改善後，請與回饋提交者追蹤後續進展。
     - 您可以使用 AWS Systems Manager OpsCenter， 以 OpsItems 的形式 建立和追蹤這些改善。

2. 追溯性分析
    - 在開發週期結束時，以固定的規律或在主要版本之後，執行追溯性分析。
    - 召集工作負載中參與的利害關係人，進行回顧會議。
    - 在白板或試算表建立三個欄位：停止、開始和持續。
        - 停止 是您希望團隊停止做的任何事。
        - 開始 是您希望開始執行的想法。
        - 持續 是您希望持續執行的項目。
    - 詢問在場人士的想法，收集利害關係人的回饋。
    - 排列回饋的優先順序。將動作和利害關係人指派至任何「開始」或「持續」項目。
    - 將動作新增至軟體開發程序中，並在您執行改善項目時向利害關係人告知最新的狀態。

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

資源

相關的最佳實務：

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

相關文件：

- 建置 CCOE 時應避開的 7 大陷阱
- Atlassian 團隊程序手冊 - 追溯性
- 電子郵件定義：回饋迴圈
- 根據 AWS Well-Architected Framework 審查建立回饋迴圈
- IBM Garage Methodology - 進行回顧
- Investopedia – PDCS 週期
- 最大化開發人員的效能 (作者：Tim Cochran)
- 營運準備度審查 (ORR) 白皮書 - 反覆執行

- TIL CSI - 持續服務改善
- 當 Toyota 遇見電子商務：Amazon 的精實原則

相關影片：

- 建立有效的客戶回饋迴圈

相關範例：

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

相關服務：

- AWS Systems Manager OpsCenter

OPS11-BP04 執行知識管理

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

預期成果：

- 團隊成員可以存取及時、準確的資訊。
- 資訊是可搜尋的。
- 存在機制以新增、更新和封存資訊。

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：高

實作指引

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

客戶範例

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

實作步驟

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

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

資源

相關的最佳實務：

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

相關文件：

- Atlassian - 知識管理

相關範例：

- DokuWiki
- Gollum
- MediaWiki
- Wiki.js

OPS11-BP05 定義改進驅動因素

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

期望的結果：

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

常見的反模式：

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

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

- 了解改進驅動因素：僅在理想結果受支援時才對系統進行變更。
  - 所需能力：在評估改進機會時，評估所需的功能和能力。
    - AWS 最新消息
  - 不可接受的問題：在評估改進機會時，評估不可接受的問題、錯誤和漏洞。追蹤調整大小的選項，並尋找最佳化機會。
    - AWS 安全佈告欄
    - AWS Trusted Advisor
    - Cloud Intelligence Dashboards
  - 合規要求：在審查改進機會時，評估保持對法規、政策的合規性或保持受到第三方支援所需的更新和變更。
    - AWS 合規
    - AWS 合規計畫
    - AWS 合規最新消息

資源

相關的最佳實務：

- OPS01 組織優先事項
- OPS02 關係和擁有權
- OPS04-BP01 識別關鍵績效指標
- OPS08 利用工作負載可觀測性
- OPS09 了解運作狀態
- OPS11-BP03 實作回饋迴圈

相關文件：

- Amazon Athena
- Amazon QuickSight
- AWS 合規

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

相關影片：

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

OPS11-BP06 驗證洞見

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

期望的結果：

- 您可以定期與企業主審查洞見。企業主為新獲得的洞見提供額外的背景資訊。
- 您可以審查洞見並要求技術同仁提供意見回饋，然後在團隊之間分享您的學習心得。
- 您可以發布資料和洞見，供其他技術和業務團隊檢閱。您在策劃其他部門的新做法時，將自己的學習心得納入考量。
- 與資深領導者一起總結並審查新洞見。資深領導者利用新的洞見來擬定策略。

常見的反模式：

- 您發布了一項新功能。此功能會改變某些客戶行為。您的可觀測性不會考慮這些變更。您不會量化這些變更的好處。
- 您推送新的更新並疏於重新整理 CDN。CDN 快取不再與最新版本相容。您可以衡量含錯誤請求的百分比。您的所有使用者都在與後端伺服器通訊時報告 HTTP 400 錯誤。您調查用戶端錯誤，發現由於測錯維度，而造成時間的浪費。

- 您的服務層級協議規定 99.9% 的正常運作時間，而您的復原點目標是四小時。服務擁有者堅決表示系統達到了零停機時間。您實作昂貴且複雜的複寫解決方案，浪費時間和金錢。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

資源

相關的最佳實務：

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

相關文件：

- 設計雲端卓越中心 (CCOE)

相關影片：

- 建置可觀測性以強化恢復能力

OPS11-BP07 執行營運指標審查

與來自不同業務領域的跨團隊參與者定期進行營運指標的追溯性分析。透過這些審查確定改進機會、可能的行動方案並分享獲得的經驗。尋找所有環境 (例如開發、測試和生產) 中的改善機會。

期望的結果：

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

常見的反模式：

- 您的維護時段造成重要的零售促銷活動中斷。如果還有其他影響企業的事件，企業仍然不知道是否有可能會延遲的標準維護時段。
- 由於您在組織中常用過時的資源庫，因此長期遭受停機之苦。之後您便遷移到受支援的資源庫。組織中的其他團隊不知道他們正面臨風險。
- 您不會定期審查客戶 SLA 達成率。您有不符合客戶 SLA 的傾向。若不符合客戶 SLA，會產生相關的財務罰責。

建立此最佳實務的優勢：

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

未建立此最佳實務時的風險暴露等級：中

實作指引

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

資源

相關的最佳實務：

- OPS08-BP05 建立儀表板

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

相關文件：

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

OPS11-BP08 記錄和分享獲得的經驗

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

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

期望的結果：

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

常見的反模式：

- 由於您的組織通常使用錯誤資源庫，致使您的組織遭受長期停機之苦。之後您便遷移到可靠的資源庫。組織中的其他團隊不知道他們正面臨風險。沒有人記錄和分享使用這個資源庫的經驗，而且他們不知道風險的存在。
- 您已在內部共用的微型服務中找出導致工作階段終止的邊緣案例。您已更新對服務的呼叫，以避免此邊緣案例。組織中的其他團隊不知道他們正面臨風險。
- 您已找到一個方法，可大幅降低其中一個微型服務所需的 CPU 使用率。您不知道是否有任何其他團隊可以利用此技術。

建立此最佳實務的優勢：分享獲得的經驗以協助改進並將經驗的好處發揮到最大。

未建立此最佳實務時的風險暴露等級：低

實作指引

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

資源

相關的最佳實務：

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

相關文件：

- 使用文件即程式碼解決方案減少專案延遲

相關影片：

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

OPS11-BP09 分配改進時間

在流程中投入時間和資源，以持續逐漸改善。

期望的結果：

- 您可以建立臨時環境複本，進而降低試驗和測試的風險、工作量及成本。
- 這些重複的環境可用於測試從您的分析、試驗和開發得出的結論，以及測試計畫的改善。
- 您執行「演練日」，並使用故障注入服務 (FIS) 提供團隊在類似生產環境中執行實驗所需的控制和防護機制。

常見的反模式：

- 您的應用程式伺服器存在已知的效能問題。將問題加入到每個規劃功能實作的待辦項目中。如果規劃功能的新增速率保持不變，則效能問題永遠不會解決。
- 為協助持續改進，您核准管理員和開發人員使用他們額外的時間來選取和實作改進項目。改進永遠不會有完成的一天。
- 作業驗收已完成，所以您未再進行測試。

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

未建立此最佳實務時的風險暴露等級：低

實作指引

- 分配改進時間：在流程中投入時間和資源，以持續、逐漸地做出改善。
- 實作變更以改進和評估結果，從而確定成功與否。
- 如果結果未能達到目標，且改進仍然是優先事項，則應尋求替代行動方案。
- 在演練日期間模擬生產工作負載，並將從這些模擬學得的經驗用於改進。

資源

相關的最佳實務：

- OPS05-BP08 使用多個環境

相關影片：

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

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。透過了解您的工作負載需求，預定義例行活動的執行手冊，和可指引問題解決方案的程序手冊，在 AWS 中將營運用作程式碼功能以及維持狀況認知，則在發生事件時，您的營運團隊將可做好更完善的準備並能夠更有效回應。

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，您將能加強活動的效率和效果，進而實現業務成功。

AWS 致力於協助您建置和營運架構，以便在您建置快速回應且適應性高的部署時能發揮最大效率。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。

作者群

- Rich Boyd，Amazon Web Services Well-Architected 卓越營運支柱主管
- Jon Steele，Amazon Web Services Well-Architected 解決方案架構師
- Ryan King，Amazon Web Services 資深技術計劃經理
- 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 Enterprise Support 資深專家技術客戶經理
- Alex Livingstone，Amazon Web Services Cloud Operations 首席專家解決方案架構師
- Paul Moran，Amazon Web Services Enterprise Support 首席技術專家
- Peter Mullen，Amazon Web Services Professional Services 諮詢顧問
- Chris Pates，Amazon Web Services Enterprise Support 資深專家技術客戶經理
- Arvind Raghunathan，Amazon Web Services Enterprise Support 首席專家技術客戶經理
- Ben Mergen，Amazon Web Services 資深解決方案架構師

深入閱讀

如需其他指引，請參考以下資源：

- AWS Well-Architected Framework
- 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
白皮書已更新	最佳實務更新了新的實作指引。	June 27, 2024
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 04 和 OPS 08) 已重新撰寫，納入了新的內容和重點。 以下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。	October 3, 2023
新框架的更新	最佳實務已更新，納入了規範性指引，並增加了新的最佳實務。	April 10, 2023
白皮書已更新	最佳實務更新了新的實作指引。	December 15, 2022
白皮書已更新	已擴充最佳實務並新增了改善計劃。	October 20, 2022
小幅度更新	小幅度編輯更新。	August 8, 2022
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。	February 2, 2022

小幅度更新	已將永續性支柱新增至簡介。	December 2, 2021
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。	July 8, 2020
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。	July 1, 2018
初版	卓越營運支柱 – AWS Well-Architected Framework 已發佈。	November 1, 2017