使用 Amazon CloudWatch 設計和實作記錄和監控

AWS 方案指引

AWS 方案指引: 使用 Amazon CloudWatch 設計和實作記錄和監控

Table of Contents

使用 Amazon CloudWatch 設計和實作記錄和監控

Khurram Nizami，Amazon Web Services (AWS)

2023 年 4 月 (文件歷史記錄)

本指南可協助您針對使用 Amazon Elastic Compute Cloud (Amazon EC2 AWS) 執行個體、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS)AWS Lambda、 和內部部署伺服器的工作負載，使用 Amazon Amazon CloudWatch 和相關的 Amazon Web Services () 管理和控管服務來設計和實作記錄和監控。 Amazon EC2 https://docs.aws.amazon.com//eks/latest/userguide/what-is-eks.html 本指南適用於在 AWS 雲端管理工作負載的營運團隊、DevOps 工程師和應用程式工程師。

您的記錄和監控方法應以 AWS Well-Architected Framework 的六個支柱為基礎。這些支柱是卓越營運、安全性、可靠性、效能效率和成本最佳化。架構良好的監控和警示解決方案可協助您主動分析和調整基礎設施，進而改善可靠性和效能。

本指南不會廣泛討論安全性或成本最佳化的記錄和監控，因為這些是需要深入評估的主題。有許多 AWS 服務支援安全記錄和監控，包括 AWS CloudTrail、AWS Config、Amazon Inspector、Amazon Detective、Amazon Macie、Amazon GuardDuty 和 AWS Security Hub CSPM。您也可以使用 AWS Cost Explorer、 AWS Budgets和 CloudWatch 帳單指標進行成本最佳化。

下表概述記錄和監控解決方案應處理的六個區域。

擷取和擷取日誌檔案和指標	識別、設定系統和應用程式日誌和指標 AWS ，並將其傳送至來自不同來源的服務。
搜尋和分析日誌	搜尋和分析 日誌以進行操作管理、問題識別、疑難排解和應用程式分析。
監控指標和警示	識別工作負載中的觀察和趨勢並對其採取行動。
監控應用程式和服務可用性	透過持續監控服務可用性，減少停機時間並改善您達到服務水準目標的能力。
追蹤應用程式	追蹤系統和外部相依性中的應用程式請求，以微調效能、執行根本原因分析，以及疑難排解問題。

建立儀表板和視覺化	建立專注於系統和工作負載相關指標和觀察的儀表板，這有助於持續改善和主動發現問題。

CloudWatch 可以滿足大多數的記錄和監控需求，並提供可靠、可擴展且靈活的解決方案。除了用於監控和分析的 CloudWatch 記錄整合之外，許多 AWS 服務也會自動提供 CloudWatch 指標。CloudWatch 也提供代理程式和日誌驅動程式，以支援各種運算選項，例如伺服器（雲端和內部部署）、容器和無伺服器運算。本指南也涵蓋下列用於記錄和監控 AWS 的服務：

- AWS Systems Manager Distributor、Systems Manager State Manager 和 Systems Manager Automation 可自動化、設定和更新 EC2 執行個體和內部部署伺服器的 CloudWatch 代理程式
- Amazon OpenSearch Service 用於進階日誌彙總、搜尋和分析
- Amazon Route 53 運作狀態檢查和 CloudWatch Synthetics 來監控應用程式和服務可用性
- Amazon Managed Service for Prometheus 用於大規模監控容器化應用程式
- AWS X-Ray 用於應用程式追蹤和執行時間分析
- Amazon Managed Grafana 可視覺化和分析來自多個來源的資料（例如 CloudWatch、Amazon OpenSearch Service 和 Amazon Timestream)

您選擇的 AWS 運算服務也會影響記錄和監控解決方案的實作和組態。例如，Amazon EC2、Amazon ECS、Amazon EKS 和 Lambda 的 CloudWatch 實作和組態不同。

應用程式和工作負載擁有者通常會忘記記錄和監控，或是設定和實作不一致。這表示工作負載進入生產時可觀測性有限，這會導致識別問題的延遲，並增加故障診斷和解決這些問題所需的時間。您的記錄和監控解決方案至少必須解決作業系統 (OS) 層級日誌和指標的系統層，以及應用程式日誌和指標的應用程式層。本指南提供建議的方法，以解決不同運算類型的這兩個層，包括下表中概述的三種運算類型。

長時間執行和不可變的 EC2 執行個體	在多個 AWS 區域或帳戶中跨多個作業系統 OSs) 的系統和應用程式日誌和指標。
容器	Amazon ECS 和 Amazon EKS 叢集的系統與應用程式日誌和指標，包括不同組態的範例。
無伺服器	Lambda 函數的系統與應用程式日誌和指標，以及自訂的考量。

本指南提供記錄和監控解決方案，可解決 CloudWatch 和下列領域的相關 AWS 服務：

- 規劃 CloudWatch 部署 – 規劃 CloudWatch 部署的考量事項，以及集中 CloudWatch 組態的指導方針。
- 為 EC2 執行個體和內部部署伺服器設定 CloudWatch 代理程式 – 系統層級和應用程式層級記錄和指標的 CloudWatch 組態詳細資訊。
- Amazon EC2 和內部部署伺服器的 CloudWatch 代理程式安裝方法 – 安裝 CloudWatch 代理程式的方法，包括跨多個區域和帳戶使用 Systems Manager 自動部署。
- 在 Amazon ECS 上記錄和監控 – Amazon ECS 中針對叢集層級和應用程式層級記錄和指標設定 CloudWatch 的指引。
- 在 Amazon EKS 上記錄和監控 – 在 Amazon EKS 中為叢集層級和應用程式層級記錄和指標設定 CloudWatch 的指引。
- Amazon EKS 上的 Prometheus 監控 – 介紹並比較 Amazon Managed Service for Prometheus 與 Prometheus 的 CloudWatch Container Insights 監控。
- 的日誌記錄和指標 AWS Lambda – 為您的 Lambda 函數設定 CloudWatch 的指引。
- 搜尋和分析 CloudWatch 中的日誌 – 使用 Amazon CloudWatch Application Insights、CloudWatch Logs Insights 分析日誌，以及將日誌分析擴展至 Amazon OpenSearch Service 的方法。
- 使用 CloudWatch 的警示選項 – 推出 CloudWatch 警示和 CloudWatch 異常偵測，並提供警示建立和設定的指引。
- 監控應用程式和服務可用性 – 介紹並比較 CloudWatch Synthetics 和 Route 53 運作狀態檢查，以進行自動可用性監控。
- 使用 追蹤應用程式 AWS X-Ray – 使用 X-Ray for Amazon EC2、Amazon ECS、Amazon EKS 和 Lambda 進行應用程式追蹤的簡介和設定
- 使用 CloudWatch 的儀表板和視覺化 – CloudWatch Dashboards 簡介，可改善 AWS 工作負載的可觀測性。
- CloudWatch 與 AWS 服務的整合 – 說明 CloudWatch 如何與各種 AWS 服務整合。
- 用於儀表板和視覺化的 Amazon Managed Grafana – 介紹並比較 Amazon Managed Grafana 與 CloudWatch 的儀表板和視覺化。

本指南中會針對這些領域使用實作範例，也可以從 AWS Samples GitHub 儲存庫取得。

目標業務成果

建立專為 AWS 雲端設計的記錄和監控解決方案，對於實現雲端運算的六個優勢至關重要。您的記錄和監控解決方案應可協助您的 IT 組織實現有益於業務流程、業務合作夥伴、員工和客戶的業務成果。在實作與 AWS Well-Architected Framework 一致的記錄和監控解決方案之後，您可以預期以下四個結果：

加速操作準備

啟用記錄和監控解決方案是準備工作負載以進行生產支援和使用的重要元件。如果您過度依賴手動程序，並且也可以減少 IT 投資的價值實現時間 (TTV)，則營運準備可能會很快成為瓶頸。無效的方法也會導致工作負載的可觀測性受到限制。這可能會增加長時間中斷、客戶不滿意和業務流程失敗的風險。

您可以使用本指南的方法來標準化和自動化您在 AWS 雲端上的記錄和監控。然後，新的工作負載需要最少的手動準備和介入才能進行生產記錄和監控。這也有助於減少跨多個帳戶和區域為不同工作負載大規模建立記錄和監控標準所需的時間和步驟。

改善卓越營運

本指南提供多個記錄和監控的最佳實務，協助各種工作負載達成業務目標和卓越營運。本指南也提供詳細的範例和開放原始碼、可重複使用的範本，您可以搭配基礎設施即程式碼 (IaC) 方法使用 AWS 服務實作架構良好的記錄和監控解決方案。改善卓越營運是反覆的，需要持續改進。本指南提供如何持續改善記錄和監控實務的建議。

增強操作可見性

您的業務流程和應用程式可能受到不同的 IT 資源支援，並託管在不同的運算類型上，無論是內部部署或 AWS 雲端。您的操作可見性可能受限於記錄和監控策略的實作不一致和不完整。採用全面的記錄和監控方法可協助您快速識別、診斷和回應工作負載的問題。本指南可協助您設計和實作方法來改善完整的操作可見性，並減少解決 (MTTR) 失敗的平均時間。全方位的記錄和監控方法也有助於您的組織改善服務品質、增強最終使用者體驗，以及滿足服務層級協議 (SLAs)。

擴展操作並降低額外負荷成本

您可以從本指南擴展記錄和監控實務，以支援多個區域和帳戶、短期資源和多個環境。本指南提供自動化手動步驟的方法和範例（例如安裝和設定客服人員、監控指標，以及在發生問題時通知或採取行

動）。當您的雲端採用變得成熟和成長，而且您需要在不增加雲端管理活動或資源的情況下擴展操作功能時，這些方法很有幫助。

規劃 CloudWatch 部署

記錄和監控解決方案的複雜性和範圍取決於幾個因素，包括：

- 使用多少環境、區域和帳戶，以及此數字可能如何增加。
- 現有工作負載和架構的多樣性和類型。
- 必須記錄和監控的運算類型和OSs。
- 是否有內部部署位置和 AWS 基礎設施。
- 多個系統和應用程式的彙總和分析需求。
- 防止未經授權暴露日誌和指標的安全要求。
- 必須與您的記錄和監控解決方案整合，以支援操作程序的產品和解決方案。

您必須使用新的或更新的工作負載部署，定期檢閱和更新記錄和監控解決方案。發現問題時，應識別並套用記錄、監控和警示的更新。然後，可以主動識別這些問題，並在未來防止這些問題。

您必須確保持續安裝和設定軟體和服務，以擷取和擷取日誌和指標。已建立的記錄和監控方法針對不同的網域（例如安全性、效能、聯網或分析），使用多個 AWS 或獨立的軟體廠商 (ISV) 服務和解決方案。每個網域都有自己的部署和組態需求。

我們建議您使用 CloudWatch 來擷取和擷取多個OSs和運算類型的日誌和指標。許多 AWS 服務使用 CloudWatch 來記錄、監控和發佈日誌和指標，而不需要進一步的組態。CloudWatch 提供軟體代理程式，可針對不同的OSs和環境進行安裝和設定。下列各節概述如何為多個帳戶、區域和組態部署、安裝和設定 CloudWatch 代理程式：

主題

- 在集中式或分散式帳戶中使用 CloudWatch
- 管理 CloudWatch 代理程式組態檔案

在集中式或分散式帳戶中使用 CloudWatch

雖然 CloudWatch 旨在監控一個帳戶和區域中 AWS 的服務或資源，但您可以使用中央帳戶從多個帳戶和區域擷取日誌和指標。如果您使用多個帳戶或區域，您應該評估是否使用集中式帳戶方法或個別帳戶來擷取日誌和指標。一般而言，多帳戶和多區域部署需要混合式方法，以支援安全、分析、操作和工作負載擁有者的需求。

下表提供選擇使用集中式、分散式或混合式方法時應考慮的領域。

帳戶結構	您的組織可能有數個不同的帳戶（例如，非生產和生產工作負載的帳戶），或特定環境中單一應用程式的數千個帳戶。我們建議您在工作負載執行所在的帳戶中維護應用程式日誌和指標，讓工作負載擁有者能夠存取日誌和指標。這可讓他們在記錄和監控中具有作用中的角色。我們也建議您使用個別的記錄帳戶來彙總所有工作負載日誌，以進行分析、彙總、趨勢和集中式操作。單獨的記錄帳戶也可以用於安全性、封存和監控和分析。
存取要求	團隊成員（例如，工作負載擁有者或開發人員）需要存取日誌和指標，以排除故障並進行改善。日誌應該保留在工作負載的帳戶中，讓存取和故障診斷更容易。如果日誌和指標維護在與工作負載不同的帳戶中，使用者可能需要定期在帳戶之間切換。 使用集中式帳戶為授權使用者提供日誌資訊，而不授予工作負載帳戶的存取權。這可以簡化分析工作負載的存取需求，其中需要多個帳戶中執行的工作負載進行彙總。集中式記錄帳戶也可以有替代的搜尋和彙總選項，例如 Amazon OpenSearch Service 叢集。Amazon OpenSearch Service 提供精細的存取控制，可低至日誌的欄位層級。當您擁有需要特殊存取和許可的敏感或機密資料時，精細存取控制非常重要。
操作	許多組織都有集中式操作和安全團隊，或外部組織提供需要存取日誌以進行監控的操作支援。集中式記錄和監控可讓您更輕鬆地識別趨勢、搜尋、彙總和執行所有帳戶和工作負載的分析。如果您的組織使用 DevOps 的「建置」方法 DevOps ，則工作負載擁有者需要在其帳戶中記錄和監控資訊。除了分散式工作負載擁有權之外，可能需要混合式方法來滿足中央操作和分析。
Environment (環境)	您可以選擇在生產帳戶的中央位置託管日誌和指標，並將其他環境的日誌和指標（例如開發或測試）保留在相同或個別的帳戶中，視安全需求和帳戶架構而定。這有助於防止更廣泛的受眾存取生產期間建立的敏感資料。

CloudWatch 提供多種選項，可讓您使用 CloudWatch 訂閱篩選條件即時處理日誌。您可以使用訂閱篩選條件將日誌即時串流至 AWS 服務，以進行自訂處理、分析和載入至其他系統。如果您採取混合方式，在個別帳戶和區域中提供日誌和指標，以及集中式帳戶和區域，這可能會特別有用。下列清單提供可用於此項目 AWS 的服務範例：

- Amazon Data Firehose – Firehose 提供串流解決方案，可根據產生的資料磁碟區自動擴展和調整大小。您不需要管理 Amazon Kinesis 資料串流中的碎片數量，而且可以直接連線至 Amazon Simple Storage Service (Amazon S3)、Amazon OpenSearch Service 或 Amazon Redshift，無需額外的編碼。如果您想要將日誌集中在 AWS 這些服務中，Firehose 是有效的解決方案。
- Amazon Kinesis Data Streams – 如果您需要與 Firehose 不支援並實作其他處理邏輯的服務整合，Kinesis Data Streams 是適當的解決方案。您可以在您的帳戶和區域中建立 Amazon CloudWatch Logs 目的地，指定中央帳戶中的 Kinesis 資料串流，以及授予其在串流中放置記錄的許可的 AWS Identity and Access Management (IAM) 角色。Kinesis Data Streams 為您的日誌資料提供彈性的開放式登陸區域，然後可供不同的選項使用。您可以將 Kinesis Data Streams 日誌資料讀取至您的帳戶、執行預先處理，並將資料傳送至您選擇的目的地。

  不過，您必須設定串流的碎片，以便針對產生的日誌資料適當調整大小。Kinesis Data Streams 可做為日誌資料的臨時媒介或佇列，您可以將資料存放在 Kinesis 串流中 1 到 365 天。Kinesis Data Streams 也支援重播功能，這表示您可以重播未使用的資料。
- Amazon OpenSearch Service – CloudWatch Logs 可以將日誌群組中的日誌串流到個別或集中帳戶中的 OpenSearch 叢集。當您設定日誌群組將資料串流至 OpenSearch 叢集時，會在與日誌群組相同的帳戶和區域中建立 Lambda 函數。Lambda 函數必須與 OpenSearch 叢集具有網路連線。除了自訂 Amazon OpenSearch Service 的擷取之外，您還可以自訂 Lambda 函數來執行額外的預先處理。使用 Amazon OpenSearch Service 的集中式記錄可讓您更輕鬆地分析、搜尋和疑難排解雲端架構中多個元件的問題。
- Lambda – 如果您使用 Kinesis Data Streams，則需要佈建和管理耗用串流資料的運算資源。若要避免這種情況，您可以直接將日誌資料串流至 Lambda 進行處理，並根據邏輯將其傳送至目的地。這表示您不需要佈建和管理運算資源，即可處理傳入的資料。如果您選擇使用 Lambda，請確定您的解決方案與 Lambda 配額相容。

您可能需要以檔案格式處理或共用存放在 CloudWatch Logs 中的日誌資料。您可以建立匯出任務，將特定日期或時間範圍的日誌群組匯出至 Amazon S3。例如，您可以選擇每天將日誌匯出至 Amazon S3 進行分析和稽核。Lambda 可用來自動化此解決方案。您也可以將此解決方案與 Amazon S3 複寫結合，將日誌從多個帳戶和區域運送和集中到一個集中帳戶和區域。

CloudWatch 代理程式組態也可以在 agent區段中指定credentials欄位。這會指定將指標和日誌傳送到不同帳戶時要使用的 IAM 角色。如果指定，此欄位會包含 role_arn 參數。當您只需要在特定集中帳戶和區域中集中記錄和監控時，即可使用此欄位。

您也可以使用 AWS SDK 以您選擇的語言撰寫自己的自訂處理應用程式、從您的帳戶讀取日誌和指標，以及將資料傳送至集中式帳戶或其他目的地，以進行進一步的處理和監控。

管理 CloudWatch 代理程式組態檔案

我們建議您建立標準 Amazon CloudWatch 代理程式組態，其中包含您要在所有 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和內部部署伺服器中擷取的系統日誌和指標。您可以使用 CloudWatch 代理程式組態檔案精靈來協助您建立組態檔案。您可以多次執行組態精靈，為不同的系統和環境產生唯一的組態。您也可以使用組態檔案結構描述來修改組態檔案或建立變化。CloudWatch 代理程式組態檔案可以存放在 AWS Systems Manager 參數存放區參數中。 如果您有多個 CloudWatch 代理程式組態檔案，您可以建立個別的參數存放區參數。如果您使用多個 AWS 帳戶或 AWS 區域，您必須管理和更新每個帳戶和區域中的參數存放區參數。或者，您可以將 CloudWatch 組態作為 Amazon S3 中的檔案或您選擇的版本控制工具集中管理。

CloudWatch 代理程式隨附的amazon-cloudwatch-agent-ctl指令碼可讓您指定組態檔案、參數存放區參數或代理程式的預設組態。預設組態符合基本的預先定義指標集，並設定代理程式向 CloudWatch 報告記憶體和磁碟空間指標。不過，它不包含任何日誌檔案組態。如果您使用 CloudWatch 代理程式的Systems Manager Quick Setup，也會套用預設組態。

由於預設組態不包含記錄，也未根據您的需求自訂，因此建議您建立並套用自訂的 CloudWatch 組態，以符合您的需求。

管理 CloudWatch 組態

根據預設，CloudWatch 組態可以儲存並套用為參數存放區參數或 CloudWatch 組態檔案。最佳選擇取決於您的需求。在本節中，我們會討論這兩個選項的優缺點。代表性解決方案也詳細說明如何管理多個 AWS 帳戶和 AWS 區域的 CloudWatch 組態檔案。

Systems Manager 參數存放區參數

如果您有單一、標準 CloudWatch 代理程式組態檔案，想要在一組小型 AWS 帳戶和區域中套用和管理，請使用參數存放區參數來管理 CloudWatch 組態。當您將 CloudWatch 組態儲存為參數存放區參數時，您可以使用 CloudWatch 代理程式組態工具 (amazon-cloudwatch-agent-ctl在 Linux 上) 從參數存放區讀取和套用組態，而無需將組態檔案複製到執行個體。您可以使用 AmazonCloudWatch-ManageAgent Systems Manager 命令文件，在單一執行中更新多個 EC2 執行個體上的 CloudWatch 組態。由於參數存放區參數是區域參數，因此您必須更新和維護每個 AWS 區域和 AWS 帳戶中的

CloudWatch 參數存放區參數。如果您要將多個 CloudWatch 組態套用至每個執行個體，則必須自訂 AmazonCloudWatch-ManageAgent Command 文件 以包含這些參數。

CloudWatch 組態檔案

如果您有多個 AWS 帳戶和區域，而且您正在管理多個 CloudWatch 組態檔案，則將 CloudWatch 組態作為檔案管理得很好。使用此方法，您可以在資料夾結構中瀏覽、組織和管理它們。 您可以將安全規則套用至個別資料夾或檔案，以限制和授予存取權，例如更新和讀取許可。您可以在 AWS 外部共用和轉移它們以進行協同合作。您可以控制檔案的版本，以追蹤和管理變更。您可以將組態檔案複製到 CloudWatch 代理程式組態目錄，而無需個別套用每個組態檔案，即可共同套用 CloudWatch 組態。對於 Linux，CloudWatch 組態目錄位於 `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d`。對於 Windows，組態目錄位於 `C:\ProgramData\Amazon\AmazonCloudWatchAgent\Configs`。

當您啟動 CloudWatch 代理程式時，代理程式會自動附加這些目錄中找到的每個檔案，以建立 CloudWatch 複合組態檔案。組態檔案應存放在中央位置 （ 例如，S3 儲存貯體），可供您所需的帳戶和區域存取。提供使用此方法的範例解決方案。

組織 CloudWatch 組態

無論用來管理 CloudWatch 組態的方法為何，請組織您的 CloudWatch 組態。您可以使用如下所示的方法，將組態整理成檔案或參數存放區路徑。

/config/standard/windows/ec2	存放 Amazon EC2 的標準 Windows 特定 CloudWatch 組態檔案。您可以在此資料夾下進一步分類不同 Windows 版本、EC2 執行個體類型和環境的標準作業系統 (OS) 組態。
/config/standard/windows/onpremises	存放內部部署伺服器的標準 Windows 特定 CloudWatch 組態檔案。您也可以在此資料夾下進一步分類不同 Windows 版本、伺服器類型和環境的標準作業系統組態。
/config/standard/linux/ec2	儲存 Amazon EC2 的標準 Linux 特定 CloudWatch 組態檔案。您可以在此資料夾下進一步分類不同 Linux 發行版本、EC2 執行個體類型和環境的標準作業系統組態。
/config/standard/linux/onpremises	儲存適用於內部部署伺服器的標準 Linux 特定 CloudWatch 組態檔案。您可以在此資料夾下進

	一步分類不同 Linux 發行版本、伺服器類型和環境的標準作業系統組態。
/config/ecs	如果您使用 Amazon ECS 容器執行個體，請存放 Amazon Elastic Container Service (Amazon ECS) 特有的 CloudWatch 組態檔案。這些組態可以附加到 Amazon ECS 特定系統層級記錄和監控的標準 Amazon EC2 組態。
/config/<application_name>	存放應用程式特定的 CloudWatch 組態檔案。您可以使用環境和版本的其他資料夾和字首進一步分類您的應用程式。

範例：將 CloudWatch 組態檔案存放在 S3 儲存貯體

本節提供使用 Amazon S3 存放 CloudWatch 組態檔案的範例，以及用於擷取和套用 CloudWatch 組態檔案的自訂 Systems Manager Runbook。此方法可以解決大規模使用 Systems Manager 參數存放區參數進行 CloudWatch 組態的一些挑戰：

- 如果您使用多個區域，則必須同步每個區域的參數存放區中的 CloudWatch 組態更新。參數存放區是區域服務，必須在使用 CloudWatch 代理程式的每個區域中更新相同的參數。
- 如果您有多個 CloudWatch 組態，您必須啟動每個參數存放區組態的擷取和應用程式。您必須從參數存放區個別擷取每個 CloudWatch 組態，並在每次新增組態時更新擷取方法。相反地，CloudWatch 提供用於儲存組態檔案的組態目錄，並套用目錄中的每個組態，而不需要個別指定這些組態。
- 如果您使用多個帳戶，則必須確保每個新帳戶在其參數存放區中具有所需的 CloudWatch 組態。您也必須確保未來會將任何組態變更套用至這些帳戶及其區域。

您可以將 CloudWatch 組態存放在可從所有帳戶和區域存取的 S3 儲存貯體中。然後，您可以使用 Systems Manager Automation Runbooks 和 Systems Manager State Manager，將這些組態從 S3 儲存貯體複製到 CloudWatch 組態目錄。您可以使用cloudwatch-config-s3-bucket.yamlAWS CloudFormation 範本來建立可從 AWS Organizations 中組織內多個帳戶存取的 S3 儲存貯體。範本包含 `OrganizationID`參數，可授予您組織內所有帳戶的讀取存取權。

本指南的設定狀態管理員和適用於 CloudWatch 代理程式的 Distributor 部署和組態區段中提供的擴增範例 Systems Manager Runbook，設定為使用由cloudwatch-config-s3-bucket.yamlAWSAWS CloudFormation 範本建立的 S3 儲存貯體擷取檔案。

或者，您可以使用版本控制系統 （例如 GitHub) 來存放您的組態檔案。如果您想要自動擷取儲存在版本控制系統中的組態檔案，您必須管理或集中登入資料儲存，並更新 Systems Manager Automation Runbook，此手冊用於跨您的帳戶和擷取登入資料 AWS 區域。

為 EC2 執行個體和內部部署伺服器設定 CloudWatch 代理程式

許多組織會在實體伺服器和虛擬機器 (VMs上執行工作負載。這些工作負載通常會在不同作業系統上執行，每個OSs都有擷取和擷取指標的唯一安裝和組態需求。

如果您選擇使用 EC2 執行個體，您可以高度控制執行個體和作業系統組態。不過，這種更高層級的控制和責任需要您監控和調整組態，以實現更有效率的使用。您可以建立記錄和監控的標準，並套用標準安裝和組態方法來擷取和擷取日誌和指標，以改善您的營運效率。

將 IT 投資遷移或擴展到 AWS 雲端的組織可以利用 CloudWatch 來實現統一的記錄和監控解決方案。CloudWatch 定價表示您要擷取的指標和日誌會遞增付費。您也可以使用與 Amazon EC2 類似的 CloudWatch 代理程式安裝程序，擷取內部部署伺服器的日誌和指標。

開始安裝和部署 CloudWatch 之前，請確定您評估系統和應用程式的日誌記錄和指標組態。請確定您定義了要為要使用OSs擷取的標準日誌和指標。系統日誌和指標是記錄和監控解決方案的基礎和標準，因為它們是由作業系統產生，Linux 和 Windows 不同。除了 Linux 版本或發行版本特有的指標和日誌檔案之外，Linux 發行版本也有一些重要的指標和日誌檔案可用。此差異也會在不同 Windows 版本之間發生。

設定 CloudWatch 代理程式

CloudWatch 會使用 CloudWatch 代理程式和每個作業系統特有的代理程式組態檔案，擷取 Amazon EC2 和內部部署伺服器的指標和日誌。 CloudWatch 建議您先定義組織的標準指標和日誌擷取組態，再開始在帳戶中大規模安裝 CloudWatch 代理程式。

您可以結合多個 CloudWatch 代理程式組態，以形成複合 CloudWatch 代理程式組態。建議的方法之一是在系統和應用程式層級定義和分割日誌和指標的組態。下圖說明如何結合適用於不同需求的多個 CloudWatch 組態檔案類型，以形成複合 CloudWatch 組態：

這些日誌和指標也可以針對特定環境或需求進一步分類和設定。例如，您可以針對未受管制的開發環境，以較低的精確度定義較小的日誌和指標子集，以及針對受管制的生產環境，以更高的精確度定義更大、更完整的集合。

設定 EC2 執行個體的日誌擷取

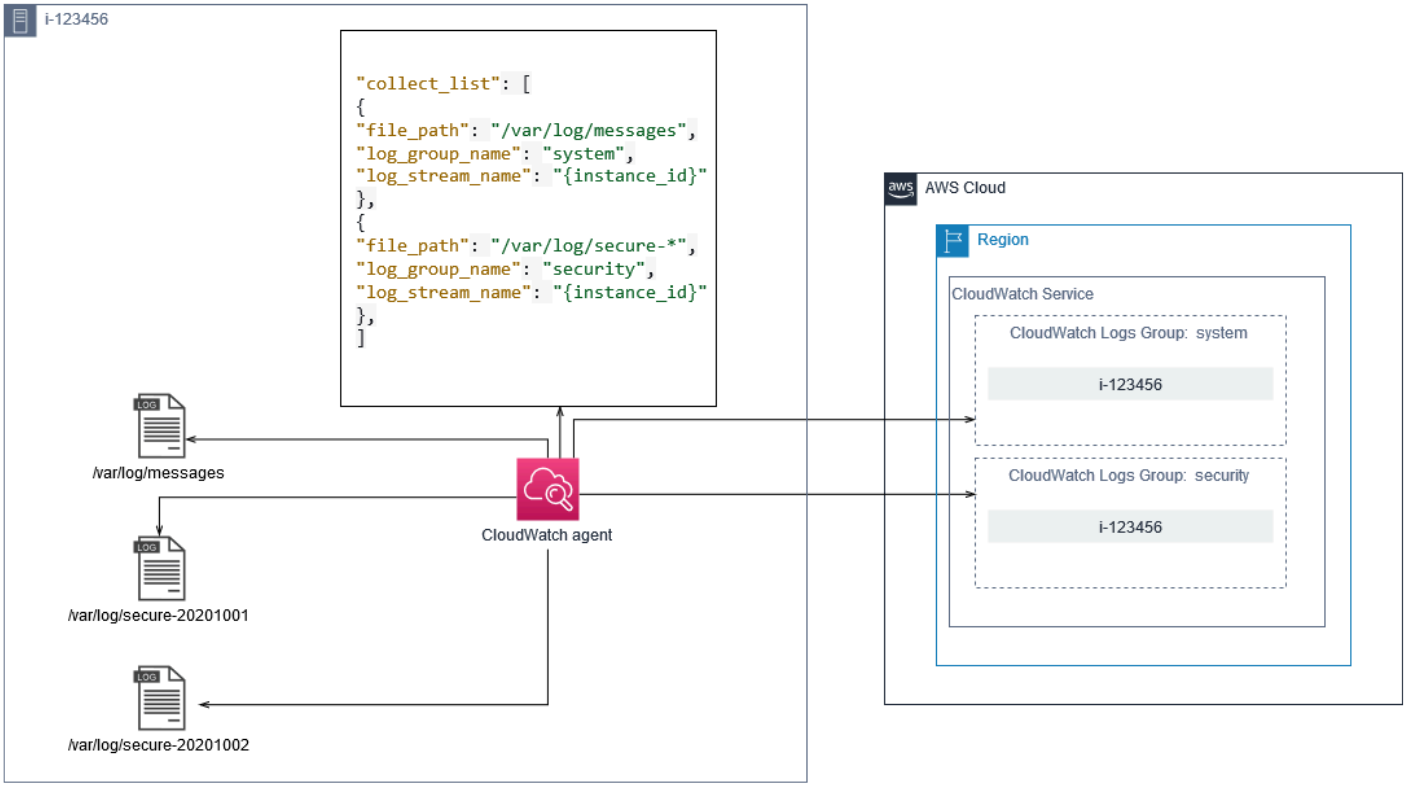
根據預設，Amazon EC2 不會監控或擷取日誌檔案。相反地，日誌檔案是由安裝在 EC2 執行個體、 AWS API 或 AWS Command Line Interface () 上的 CloudWatch 代理程式軟體擷取並擷取至 CloudWatch Logs AWS CLI。我們建議使用 CloudWatch 代理程式，將日誌檔案擷取到 Amazon EC2 和內部部署伺服器的 CloudWatch Logs。

您可以搜尋和篩選日誌，以及擷取指標，並根據 CloudWatch 中日誌檔案中的模式修補來執行自動化。CloudWatch 支援純文字、空格分隔和 JSON 格式的篩選條件和模式語法選項，搭配 JSON 格式的日誌可提供最大的彈性。若要增加篩選和分析選項，您應該使用格式化的日誌輸出，而不是純文字。

CloudWatch 代理程式使用組態檔案來定義要傳送至 CloudWatch 的日誌和指標。然後CloudWatch 會將每個日誌檔案擷取為日誌串流，並將這些日誌串流分組為日誌群組。這可協助您從 EC2 執行個體跨日誌執行操作，例如搜尋相符的字串。

預設日誌串流名稱與 EC2 執行個體 ID 相同，預設日誌群組名稱與日誌檔案路徑相同。日誌串流的名稱在 CloudWatch 日誌群組中必須是唯一的。您可以在日誌串流和日誌群組名稱中使用 `instance_id`、`local_hostname`、 `hostname`或 `ip_address`進行動態替換，這表示您可以在多個 EC2 執行個體間使用相同的 CloudWatch 代理程式組態檔案。

下圖顯示用於擷取日誌的 CloudWatch 代理程式組態。日誌群組是由擷取的日誌檔案定義，並包含每個 EC2 執行個體的個別日誌串流，因為 `{instance_id}`變數用於日誌串流名稱和 EC2 執行個體 IDs是唯一的。



日誌群組定義其包含之日誌串流的保留、標籤、安全性、指標篩選條件和搜尋範圍。根據日誌檔案名稱的預設分組行為可協助您搜尋、建立指標，以及對帳戶和區域中跨 EC2 執行個體之日誌檔案的特定資料發出警示。您應該評估是否需要進一步的日誌群組精簡。例如，您的帳戶可能由多個業務單位共用，並且擁有不同的技術或操作擁有者。這表示您必須進一步精簡日誌群組名稱，以反映分離和擁有權。此方法可讓您將分析和故障診斷集中在相關的 EC2 執行個體上。

如果多個環境使用一個帳戶，您可以為每個環境中執行的工作負載分開記錄。下表顯示包含業務單位、專案或應用程式和環境的日誌群組命名慣例。

日誌群組名稱	/<Business unit>/<Project or application name>/<Environment>/<Log file name>
日誌串流名稱	<EC2 instance ID>

您也可以將所有 EC2 執行個體的日誌檔案分組到相同的日誌群組。這可讓您更輕鬆地搜尋和分析單一 EC2 執行個體的一組日誌檔案。如果您的大多數 EC2 執行個體為一個應用程式或工作負載提供服務，

而且每個 EC2 執行個體都具有特定用途，這很有用。下表顯示如何格式化您的日誌群組和日誌串流命名，以支援此方法。

日誌群組名稱	/<Business unit>/<Project or application name>/<Environment>/<EC2 instance ID>
日誌串流名稱	<Log file name>

設定 EC2 執行個體的指標擷取

根據預設，您的 EC2 執行個體會啟用基本監控，並且每五分鐘會自動將一組標準指標（例如 CPU、網路或儲存相關指標）傳送至 CloudWatch。CloudWatch 指標可能會因執行個體系列而有所不同，例如，爆量效能執行個體具有 CPU 點數的指標。Amazon EC2 標準指標包含在執行個體價格中。如果您啟用 EC2 執行個體的詳細監控，您可以在一分鐘期間內接收資料。期間頻率會影響 CloudWatch 成本，因此請務必評估所有或僅部分 EC2 執行個體是否需要詳細監控。例如，您可以啟用生產工作負載的詳細監控，但對非生產工作負載使用基本監控。

內部部署伺服器不包含 CloudWatch 的任何預設指標，且必須使用 CloudWatch 代理程式 AWS CLI、或 AWS SDK 來擷取指標。這表示您必須定義要在 CloudWatch 組態檔案中擷取的指標（例如 CPU 使用率）。您可以建立唯一的 CloudWatch 組態檔案，其中包含內部部署伺服器的標準 EC2 執行個體指標，除了標準 CloudWatch 組態之外，還會套用它。

CloudWatch 中的指標由指標名稱和零個或多個維度唯一定義，並在指標命名空間中唯一分組。 AWS 服務提供的指標具有開頭為 的命名空間 AWS（例如 AWS/EC2），非AWS 指標則視為自訂指標。您使用 CloudWatch 代理程式設定和擷取的指標都視為自訂指標。由於建立的指標數目會影響 CloudWatch 成本，因此您應該評估所有或僅部分 EC2 執行個體是否需要每個指標。例如，您可以為生產工作負載定義一組完整的指標，但針對非生產工作負載使用這些指標的較小子集。

CWAgent 是 CloudWatch 代理程式發佈之指標的預設命名空間。與日誌群組類似，指標命名空間會組織一組指標，以便在同一個位置找到它們。您應該修改命名空間，以反映業務單位、專案或應用程式，以及環境（例如 /<Business unit>/<Project or application name>/<Environment>）。如果多個不相關的工作負載使用相同的帳戶，此方法非常有用。您也可以將命名空間命名慣例與 CloudWatch 日誌群組命名慣例建立關聯。

指標也會透過其維度來識別，這可協助您根據一組條件來分析指標，並且是記錄觀察的屬性。Amazon EC2 包含具有 InstanceId和 AutoScalingGroupName維度之 EC2 執行個體的個別指標。如果您

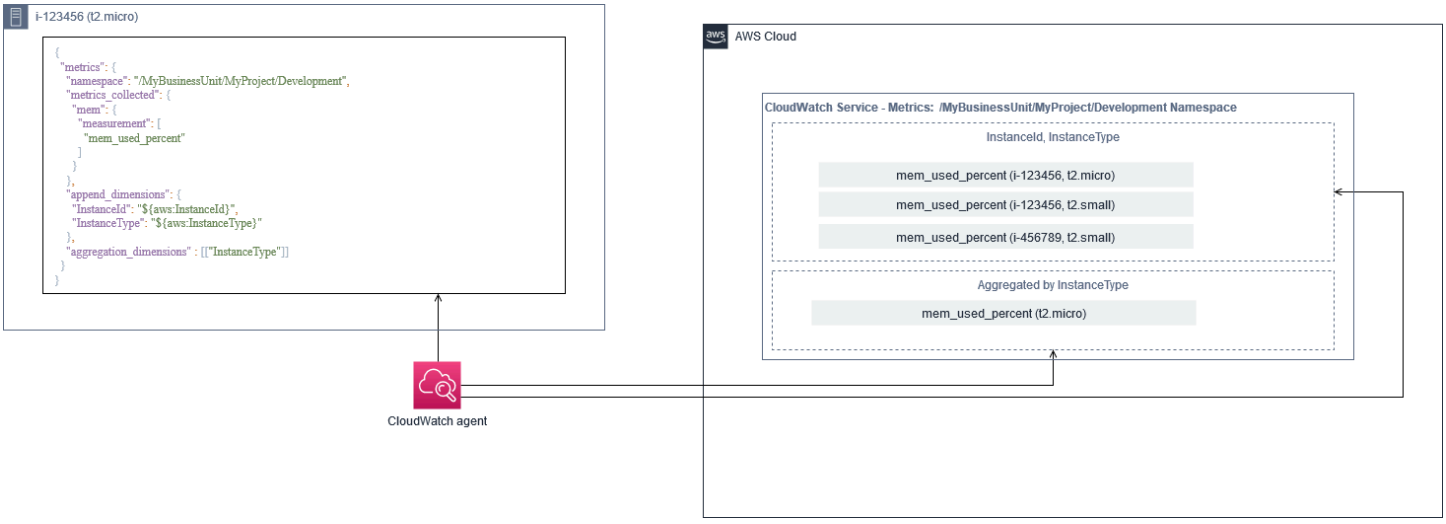
啟用詳細監控，也會收到具有 ImageId和 InstanceType維度的指標。例如，Amazon EC2 為 CPU 使用率提供單獨的 EC2 執行個體指標與InstanceId維度，以及為InstanceType維度提供單獨的 CPU 使用率指標。這可協助您分析每個唯一 EC2 執行個體的 CPU 使用率，以及特定執行個體類型的所有 EC2 執行個體。 https://docs.aws.amazon.com//AWSEC2/latest/UserGuide/instance-types.html

新增更多維度可增加您的分析功能，但也會增加整體成本，因為每個指標和唯一的維度值組合都會產生新的指標。例如，如果您針對維InstanceId度建立記憶體使用率百分比的指標，則這是每個 EC2 執行個體的新指標。如果您的組織執行數千個 EC2 執行個體，這會導致數千個指標並產生更高的成本。若要控制和預測成本，請確定您決定指標的基數，以及哪些維度會新增最大值。例如，您可以為生產工作負載指標定義一組完整的維度，但為非生產工作負載定義這些維度的較小子集。

您可以使用 append_dimensions 屬性，將維度新增至 CloudWatch 組態中定義的一個或所有指標。您也可以動態將 ImageId、InstanceType、 InstanceId和 附加AutoScalingGroupName至 CloudWatch 組態中的所有指標。或者，您可以使用該指標上的 append_dimensions 屬性來附加特定指標的任意維度名稱和值。CloudWatch 也可以彙總您使用 aggregation_dimensions 屬性定義的指標維度統計資料。

例如，您可以彙總針對InstanceType維度使用的記憶體，以查看所有 EC2 執行個體針對每個執行個體類型使用的平均記憶體。如果您使用在區域中執行的t2.micro執行個體，您可以判斷使用 t2.micro類別的工作負載是否過度利用或過度利用提供的記憶體。利用率不足可能是使用 EC2 類別搭配不需要記憶體容量之工作負載的跡象。相反地，過度使用可能是使用記憶體不足之 Amazon EC2 類別的工作負載跡象。

下圖顯示使用自訂命名空間、新增維度和 彙總的範例 CloudWatch 指標組態InstanceType。



系統層級 CloudWatch 組態

系統層級指標和日誌是監控和記錄解決方案的核心元件，而 CloudWatch 代理程式具有適用於 Windows 和 Linux 的特定組態選項。

我們建議您使用 CloudWatch 組態檔案精靈或組態檔案結構描述，為您計劃支援的每個作業系統定義 CloudWatch 代理程式組態檔案。其他工作負載特定的作業系統層級日誌和指標可以在單獨的 CloudWatch 組態檔案中定義，並附加至標準組態。這些唯一的組態檔案應分別存放在 S3 儲存貯體中，以供 EC2 執行個體擷取。此用途的 S3 儲存貯體設定範例說明於本指南的 管理 CloudWatch 組態一節。您可以使用 State Manager 和 Distributor 自動擷取和套用這些組態。

設定系統層級日誌

系統層級日誌對於診斷和疑難排解內部部署或 AWS 雲端的問題至關重要。您的日誌擷取方法應包含作業系統產生的任何系統和安全日誌。根據作業系統版本，作業系統產生的日誌檔案可能會有所不同。

CloudWatch 代理程式支援透過提供事件日誌名稱來監控 Windows 事件日誌。您可以選擇要監控的 Windows 事件日誌（例如 `System`、 `Application`或 `Security`)。

Linux 系統的系統、應用程式和安全日誌通常存放在 `/var/log`目錄中。下表定義了您應該監控的常見預設日誌檔案，但您應該檢查 `/etc/rsyslog.conf`或 `/etc/syslog.conf` 檔案，以確定系統日誌檔案的特定設定。

Fedora 分佈	`/var/log/boot.log*` – 開機日誌
(Amazon Linux、CentOS、Red Hat Enterprise Linux)	`/var/log/dmesg` – 核心日誌
	`/var/log/secure` – 安全性和身分驗證日誌
	`/var/log/messages` – 一般系統日誌
	`/var/log/cron*` – Cron 日誌
	`/var/log/cloud-init-output.log` – 從Userdata啟動指令碼輸出
Debian	`/var/log/syslog` – 開機日誌
(Ubuntu)	

		/var/log/cloud-init-output.log – 從Userdata啟動指令碼輸出
		/var/log/auth.log – 安全性和身分驗證日誌
		/var/log/kern.log – 核心日誌

您的組織可能也有其他代理程式或系統元件，可產生您要監控的日誌。您應該評估並決定這些代理程式或應用程式產生的日誌檔案，並透過識別它們的檔案位置將其包含在組態中。例如，您應該在組態中包含 Systems Manager 和 CloudWatch 代理程式日誌。下表提供這些 Windows 和 Linux 代理程式日誌的位置。

Windows	CloudWatch 代理程式	$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log
	Systems Manager 代理程式	%PROGRAMDATA%\Amazon\SSM\Logs\amazon-ssm-agent.log %PROGRAMDATA%\Amazon\SSM\Logs\errors.log %PROGRAMDATA%\Amazon\SSM\Logs\audits\amazon-ssm-agent-audit-YYYY-MM-DD
Linux	CloudWatch 代理程式	/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log

	Systems Manager 代理程式	`/var/log/amazon/ssm/amazon-ssm-agent.log`
		`/var/log/amazon/ssm/errors.log`
		`/var/log/amazon/ssm/audits/amazon-ssm-agent-audit-YYYY-MM-DD`

如果日誌檔案是在 CloudWatch 代理程式組態中定義，但找不到，則 CloudWatch 會忽略日誌檔案。當您想要維護 Linux 的單一日誌組態，而不是每個發行版本的個別組態時，這會很有用。在代理程式或軟體應用程式開始執行之前，日誌檔案不存在也很有用。

設定系統層級指標

記憶體和磁碟空間使用率不包含在 Amazon EC2 提供的標準指標中。若要包含這些指標，您必須在 EC2 執行個體上安裝和設定 CloudWatch 代理程式。CloudWatch 代理程式組態精靈會使用預先定義的指標建立 CloudWatch 組態，您可以視需要新增或移除指標。請務必檢閱預先定義的指標集，以判斷您需要的適當層級。

最終使用者和工作負載擁有者應根據伺服器或 EC2 執行個體的特定需求發佈其他系統指標。這些指標定義應該存放在單獨的 CloudWatch 代理程式組態檔案中，並進行版本控制和維護，並在中央位置（例如 Amazon S3) 共用以重複使用和自動化。

標準 Amazon EC2 指標不會在內部部署伺服器中自動擷取。這些指標必須在內部部署執行個體所使用的 CloudWatch 代理程式組態檔案中定義。您可以使用 CPU 使用率等指標為內部部署執行個體建立個別的指標組態檔案，並將這些指標附加至標準指標組態檔案。

應用程式層級 CloudWatch 組態

應用程式日誌和指標是由執行中的應用程式產生，且特定於應用程式。請確定您定義了充分監控組織定期使用之應用程式所需的日誌和指標。例如，您的組織可能已針對 Web 應用程式在 Microsoft Internet Information Server (IIS) 上標準化。您可以為 IIS 建立標準日誌和指標 CloudWatch 組態，也可以在您的組織中使用。應用程式特定的組態檔案可以存放在集中位置（例如 S3 儲存貯體），並由工作負載擁有者或透過自動擷取存取，並複製到 CloudWatch 組態目錄。CloudWatch 代理程

式會自動將每個 EC2 執行個體或伺服器的組態檔案目錄中找到的 CloudWatch 組態檔案合併為複合 CloudWatch 組態。最終結果是包含組織標準系統層級組態的 CloudWatch 組態，以及所有相關應用程式層級 CloudWatch 組態。

工作負載擁有者應識別和設定所有關鍵應用程式和元件的日誌檔案和指標。

設定應用程式層級日誌

應用程式層級記錄會根據應用程式是商用off-the-shelf(COTS) 還是自訂開發的應用程式而有所不同。COTS 應用程式及其元件可能會為日誌組態和輸出提供數個選項，例如日誌詳細資訊層級、日誌檔案格式和日誌檔案位置。不過，大多數 COTS 或第三方應用程式不允許您從根本上變更記錄（例如，更新應用程式的程式碼以包含無法設定的其他日誌陳述式或格式）。您至少應該設定 COTS 或第三方應用程式的記錄選項，以記錄警告和錯誤層級資訊，最好是 JSON 格式。

您可以在 CloudWatch 組態中包含應用程式的日誌檔案，將自訂開發的應用程式與 CloudWatch Logs 整合。自訂應用程式提供更好的日誌品質和控制，因為除了包含任何其他必要的詳細資訊之外，您還可以自訂日誌輸出格式、分類和分隔元件輸出以分隔日誌檔案。請務必檢閱並標準化記錄程式庫，以及組織所需的資料和格式，以便更輕鬆地分析和處理。

您也可以使用 CloudWatch Logs `PutLogEvents` API 呼叫或使用 AWS SDK 寫入 CloudWatch 日誌串流。您可以使用 API 或 SDK 進行自訂記錄需求，例如協調跨分散式元件和伺服器的單一日誌串流記錄。不過，維護最簡單且最廣泛適用的解決方案是設定您的應用程式寫入日誌檔案，然後使用 CloudWatch 代理程式讀取日誌檔案並將其串流至 CloudWatch。

您也應該考慮要從應用程式日誌檔案測量的指標類型。您可以使用指標篩選條件來測量、繪製和警示 CloudWatch 日誌群組中的此資料。例如，您可以使用指標篩選條件，透過在日誌中識別失敗的登入嘗試來計算這些嘗試。

您也可以在應用程式日誌檔案中使用 CloudWatch 內嵌指標格式，為自訂開發的應用程式建立自訂指標。

設定應用程式層級指標

自訂指標是由 AWS 服務直接提供給 CloudWatch 的指標，它們會在 CloudWatch 指標的自訂命名空間中發佈。所有應用程式指標都視為自訂 CloudWatch 指標。應用程式指標可能符合 EC2 執行個體、應用程式元件、API 呼叫，甚至是商業函數。您還必須考慮您為指標選擇的維度的重要性和基數。具有高基數的維度會產生大量自訂指標，並可能增加您的 CloudWatch 成本。

CloudWatch 可協助您以多種方式擷取應用程式層級指標，包括下列項目：

- 透過定義您要從 procstat 外掛程式擷取的個別程序來擷取程序層級指標。
- 應用程式會將指標發佈至 Windows Performance Monitor，此指標會在 CloudWatch 組態中定義。
- 指標篩選條件和模式會套用到 CloudWatch 中的應用程式日誌。
- 應用程式會使用 CloudWatch 內嵌指標格式寫入 CloudWatch 日誌。
- 應用程式會透過 API 或 AWS SDK 將指標傳送至 CloudWatch。
- 應用程式會使用設定的 CloudWatch 代理程式，將指標傳送至已收集或 StatsD 協助程式。

您可以使用 procstat 來監控和測量 CloudWatch 代理程式的關鍵應用程式程序。這可協助您在應用程式不再執行關鍵程序時發出警示並採取動作（例如，通知或重新啟動程序）。您也可以測量應用程式程序的效能特性，並在特定程序運作異常時發出警示。

如果您無法使用其他自訂指標更新 COTS 應用程式，Procstat 監控也很有用。例如，您可以建立測量 `my_process` `cpu_time`並包含自訂`application_version`維度的指標。如果您的不同指標有不同的維度，您也可以為應用程式使用多個 CloudWatch 代理程式組態檔案。

如果您的應用程式在 Windows 上執行，您應該評估應用程式是否已將指標發佈至 Windows Performance Monitor。許多 COTS 應用程式與 Windows Performance Monitor 整合，可協助您輕鬆監控應用程式指標。CloudWatch 也與 Windows Performance Monitor 整合，您可以擷取其中現有的任何指標。

請務必檢閱應用程式提供的記錄格式和日誌資訊，以判斷可以使用指標篩選條件擷取哪些指標。您可以檢閱應用程式的歷史日誌，以判斷錯誤訊息和異常關機的呈現方式。您也應該檢閱先前報告的問題，以判斷是否可以擷取指標，以防止問題重複發生。您也應該檢閱應用程式的文件，並要求應用程式開發人員確認如何識別錯誤訊息。

對於自訂開發的應用程式，請與應用程式的開發人員合作，定義可使用 CloudWatch 內嵌指標格式、AWS SDK 或 AWS API 實作的重要指標。建議的方法是使用內嵌指標格式。您可以使用 AWS 提供的開放原始碼內嵌指標格式程式庫，協助您以所需的格式撰寫陳述式。您也需要更新應用程式特定的 CloudWatch 組態，以包含內嵌指標格式代理程式。這會導致 EC2 執行個體上執行的代理程式充當本機內嵌指標格式端點，將內嵌指標格式指標傳送至 CloudWatch。

如果您的應用程式已支援將指標發佈至收集或統計，您可以利用這些指標將指標擷取至 CloudWatch。

Amazon EC2 和內部部署伺服器的 CloudWatch 代理程式安裝方法

自動化 CloudWatch 代理程式的安裝程序可協助您快速且一致地部署它，並擷取所需的日誌和指標。有數種方法可自動化 CloudWatch 代理程式安裝，包括多帳戶和多區域支援。討論了下列自動化安裝方法：

- 使用 Systems Manager Distributor 和 Systems Manager State Manager 安裝 CloudWatch 代理程式 – 如果您的 EC2 執行個體和內部部署伺服器正在執行 Systems Manager 代理程式，我們建議您使用此方法。這可確保 CloudWatch 代理程式保持更新，而且您可以在沒有 CloudWatch 代理程式的伺服器上進行報告和修復。此方法也會擴展以支援多個帳戶和區域。
- 在 EC2 執行個體佈建期間，將 CloudWatch 代理程式部署為使用者資料指令碼的一部分 – Amazon EC2 可讓您定義在第一次開機或重新啟動時執行的啟動指令碼。您可以定義指令碼，以自動化代理程式的下載和安裝程序。這也可以包含在 CloudFormation 指令碼和 AWS Service Catalog 產品中。如果特定工作負載有自訂代理程式安裝和組態方法偏離您的標準，則此方法可能視需要適用。
- 在 Amazon Machine Image (AMIs) 中包含 CloudWatch 代理程式 – 您可以在 Amazon EC2 的自訂 AMIs 中安裝 CloudWatch 代理程式。使用 AMI 的 EC2 執行個體會自動安裝並啟動代理程式。不過，您必須確保代理程式及其組態定期更新。

使用 Systems Manager Distributor 和 State Manager 安裝 CloudWatch 代理程式

您可以使用 Systems Manager State Manager 搭配 Systems Manager Distributor，在伺服器和 EC2 執行個體上自動安裝和更新 CloudWatch 代理程式。Distributor 包含安裝最新 CloudWatch 代理程式版本的`AmazonCloudWatchAgent` AWS 受管套件。

此安裝方法具有下列先決條件：

- Systems Manager 代理程式必須在您的伺服器或 EC2 執行個體上安裝並執行。Systems Manager 代理程式預先安裝在 Amazon Linux、Amazon Linux 2 和一些 AMIs 上。代理程式也必須安裝在其他映像或內部部署 VMs 和伺服器上並進行設定。

> **Note**
>
> Amazon Linux 2 即將終止支援。如需詳細資訊，請參閱 Amazon Linux 2 FAQs。

- 具有所需 CloudWatch 和 Systems Manager 許可的 IAM 角色或登入資料必須連接到 EC2 執行個體，或在內部部署伺服器的登入資料檔案中定義。例如，您可以建立包含 AWS 受管政策的 IAM 角色：`AmazonSSMManagedInstanceCore`適用於 Systems Manager 和`CloudWatchAgentServerPolicy`適用於 CloudWatch 的 。您可以使用 ssm-cloudwatch-instance-role.yaml CloudFormation 範本來部署包含這兩個政策的 IAM 角色和執行個體描述檔。您也可以修改此範本，以包含 EC2 執行個體的其他標準 IAM 許可。對於內部部署伺服器或 VMs， 應該設定 CloudWatch 代理程式使用為內部部署伺服器設定的 Systems Manager 服務角色。如需詳細資訊，請參閱 AWS 知識中心中的如何設定使用 Systems Manager 代理程式和統一 CloudWatch 代理程式的現場部署伺服器，以僅使用暫時登入資料？。

下列清單提供數個使用 Systems Manager Distributor 和 State Manager 方法來安裝和維護 CloudWatch 代理程式的優點：

- 多個OSs的自動安裝 – 您不需要為每個作業系統撰寫和維護指令碼，即可下載和安裝 CloudWatch 代理程式。
- 自動更新檢查 – State Manager 會自動並定期檢查每個 EC2 執行個體是否具有最新的 CloudWatch 版本。
- 合規報告 – Systems Manager 合規儀表板會顯示哪些 EC2 執行個體無法成功安裝 Distributor 套件。
- 為新啟動的 EC2 執行個體自動安裝 – 在帳戶中啟動的新 EC2 執行個體會自動接收 CloudWatch 代理程式。

不過，在選擇此方法之前，您也應該考慮以下三個領域：

- 與現有關聯的衝突 – 如果另一個關聯已安裝或設定 CloudWatch 代理程式，則兩個關聯可能會互相干擾，並可能導致問題。使用此方法時，您應該移除安裝或更新 CloudWatch 代理程式和組態的任何現有關聯。
- 更新自訂代理程式組態檔案 – 經銷商會使用預設組態檔案執行安裝。如果您使用自訂組態檔案或多個 CloudWatch 組態檔案，您必須在安裝後更新組態。
- 多區域或多帳戶設定 – 必須在每個帳戶和區域中設定 State Manager 關聯。多帳戶環境中的新帳戶必須更新，以包含狀態管理員關聯。您需要集中或同步 CloudWatch 組態，以便多個帳戶和區域可以擷取和套用所需的標準。

設定 CloudWatch 代理程式部署和組態的 State Manager 和 Distributor

您可以使用 Systems Manager Quick Setup 快速設定 Systems Manager 功能，包括在 EC2 執行個體上自動安裝和更新 CloudWatch 代理程式。快速設定會部署 CloudFormation 堆疊，以根據您的選擇部署和設定 Systems Manager 資源。

下列清單提供由 Quick Setup 執行的兩個重要動作，用於自動安裝和更新 CloudWatch 代理程式：

1. 建立 Systems Manager 自訂文件 – 快速設定會建立下列 Systems Manager 文件，以搭配 State Manager 使用。文件名稱可能會有所不同，但內容保持不變：
   - `CreateAndAttachIAMToInstance` – 建立不存在`AmazonSSMRoleForInstancesQuickSetup`的角色和執行個體描述檔，並將`AmazonSSMManagedInstanceCore`政策連接到角色。這不包含必要的 `CloudWatchAgentServerPolicy` IAM 政策。您必須更新此政策並更新此 Systems Manager 文件，以包含此政策，如下節所述。
   - `InstallAndManageCloudWatchDocument` – 使用 Systems `AWS-ConfigureAWSPackage` Manager 文件安裝 CloudWatch 代理程式與 Distributor，並使用預設 CloudWatch 代理程式組態設定每個 EC2 執行個體一次。
   - `UpdateCloudWatchDocument` – 使用 `AWS-ConfigureAWSPackage` Systems Manager 文件安裝最新的 CloudWatch 代理程式，以更新 CloudWatch 代理程式。 CloudWatch 更新或解除安裝代理程式不會從 EC2 執行個體移除現有的 CloudWatch 組態檔案。
2. 建立狀態管理員關聯 – 狀態管理員關聯會建立並設定為使用自訂建立的 Systems Manager 文件。狀態管理員關聯名稱可能會有所不同，但組態保持不變：
   - `ManageCloudWatchAgent` – 為每個 EC2 執行個體執行 Systems `InstallAndManageCloudWatchDocument` Manager 文件一次。
   - `UpdateCloudWatchAgent` – 每 30 天針對每個 EC2 執行個體執行 `UpdateCloudWatchDocument` Systems Manager 文件。
   - 為每個 EC2 執行個體執行一次 `CreateAndAttachIAMToInstance` Systems Manager 文件。

您必須擴增並自訂完成的快速設定組態，以包含 CloudWatch 許可並支援自訂 CloudWatch 組態。特別是， `CreateAndAttachIAMToInstance`和 `InstallAndManageCloudWatchDocument` 文件將需要更新。您可以手動更新 Quick Setup 建立的 Systems Manager 文件。或者，您可以使用自己的 CloudFormation 範本來佈建具有必要更新的相同資源，以及設定和部署其他 Systems Manager 資源，而不使用快速設定。

> **Important**
>
> 快速設定會建立 CloudFormation 堆疊，以根據您的選擇部署和設定 Systems Manager 資源。如果您更新快速設定選擇，您可能需要手動重新更新 Systems Manager 文件。

下列各節說明如何手動更新 Quick Setup 建立的 Systems Manager 資源，以及使用您自己的 CloudFormation 範本來執行更新的 Quick Setup。我們建議您使用自己的 CloudFormation 範本，以避免手動更新 Quick Setup 和 建立的資源 CloudFormation。

使用 Systems Manager Quick Setup 並手動更新建立的 Systems Manager 資源

快速設定方法建立的 Systems Manager 資源必須更新，以包含所需的 CloudWatch 代理程式許可，並支援多個 CloudWatch 組態檔案。本節說明如何更新 IAM 角色和 Systems Manager 文件，以使用包含可從多個帳戶存取之 CloudWatch 組態的集中式 S3 儲存貯體。本指南的 管理 CloudWatch 組態區段討論建立 S3 儲存貯體以存放 CloudWatch 組態檔案。

更新 **`CreateAndAttachIAMToInstance`** Systems Manager 文件

快速設定建立的此 Systems Manager 文件會檢查 EC2 執行個體是否已連接現有的 IAM 執行個體描述檔。如果是這樣，它會將`AmazonSSMManagedInstanceCore`政策連接到現有角色。這可保護您現有的 EC2 執行個體，避免失去可能透過現有執行個體設定檔指派的 AWS 許可。您需要在此文件中新增步驟，將 `CloudWatchAgentServerPolicy` IAM 政策連接至已連接執行個體描述檔的 EC2 執行個體。如果 IAM 角色不存在且 EC2 執行個體未連接執行個體描述檔，Systems Manager 文件也會建立該角色。您必須更新文件的此區段，以同時包含 IAM `CloudWatchAgentServerPolicy` 政策。

檢閱完成的 CreateAndAttachIAMToInstance.yaml 範例文件，並將其與 Quick Setup 建立的文件進行比較。編輯現有的文件，以包含必要的步驟和變更。根據您的快速設定選擇，快速設定建立的文件可能與提供的範例文件不同，因此請務必進行必要的調整。範例文件包含每日掃描執行個體是否有遺失修補程式的快速設定選項選項，因此包含 Systems Manager Patch Manager 的政策。

更新 **`InstallAndManageCloudWatchDocument`** Systems Manager 文件

此由 Quick Setup 建立的 Systems Manager 文件會安裝 CloudWatch 代理程式，並使用預設 CloudWatch 代理程式組態進行設定。預設 CloudWatch 組態符合基本、預先定義的指標集。您必須取代預設組態步驟，並新增從 CloudWatch 組態 S3 儲存貯體下載 CloudWatch 組態檔案的步驟。

檢閱已完成的 InstallAndManageCloudWatchDocument.yaml 更新文件，並將其與 Quick Setup 建立的文件進行比較。快速設定建立的文件可能不同，因此請確定您已進行必要的調整。編輯現有的文件，以包含必要的步驟和變更。

使用 CloudFormation 而非快速設定

您可以使用 CloudFormation 來設定 Systems Manager，而不是使用快速設定。此方法可讓您根據您的特定需求自訂 Systems Manager 組態。此方法也可避免手動更新 Quick Setup 建立的 Systems Manager 資源，以支援自訂 CloudWatch 組態。

快速設定功能也會使用 CloudFormation 和建立 CloudFormation 堆疊集，以根據您的選擇部署和設定 Systems Manager 資源。您必須先建立 CloudFormation StackSets 用來支援跨多個帳戶或區域的部署的 IAM 角色，才能使用 CloudFormation 堆疊集。快速設定會建立使用 CloudFormation StackSets 支援多區域或多帳戶部署所需的角色。如果您想要在多個區域或從單一帳戶和區域設定和部署 Systems Manager 資源，您必須完成 CloudFormation StackSets 的先決條件。如需詳細資訊，請參閱 CloudFormation 文件中的堆疊集操作的先決條件。

檢閱自訂快速設定的 AWS-QuickSetup-SSMHostMgmt.yaml 範本。 CloudFormation

您應該檢閱 CloudFormation 範本中的資源和功能，並根據需求進行調整。您應該控制使用的 CloudFormation 範本版本，並逐步測試變更以確認所需的結果。此外，您應該執行雲端安全審查，以根據組織的需求判斷是否需要任何政策調整。

您應該在單一測試帳戶和區域中部署 CloudFormation 堆疊，並執行任何必要的測試案例來自訂和確認所需的結果。然後，您可以將部署擴展到單一帳戶中的多個區域，然後擴展到多個帳戶和多個區域。

使用 CloudFormation 堆疊在單一帳戶和區域中自訂快速設定

如果您只使用單一帳戶和區域，您可以將完整範例部署為 CloudFormation 堆疊，而不是 CloudFormation 堆疊集。不過，如果可能，我們建議您使用多帳戶、多區域堆疊集方法，即使只使用單一帳戶和區域。Using CloudFormation StackSets 可讓您在未來更輕鬆地擴展到其他帳戶和區域。

使用下列步驟，將 AWS-QuickSetup-SSMHostMgmt.yaml CloudFormation 範本部署為單一帳戶中的 CloudFormation 堆疊，以及 AWS 區域：

1. 下載範本並將其檢查到您偏好的版本控制系統（例如 GitHub)。
2. 根據組織的需求自訂預設 CloudFormation 參數值。
3. 自訂狀態管理員關聯排程。

4. 使用邏輯 ID `InstallAndManageCloudWatchDocument` 自訂 Systems Manager 文件。確認 S3 儲存貯體字首符合包含 CloudWatch 組態的 S3 儲存貯體字首。
5. 擷取並記錄包含 CloudWatch 組態之 S3 儲存貯體的 Amazon Resource Name (ARN)。如需詳細資訊，請參閱本指南的 管理 CloudWatch 組態一節。提供範例 cloudwatch-config-s3-bucket.yaml CloudFormation 範本，其中包含儲存貯體政策，以提供 AWS Organizations 帳戶的讀取存取權。
6. 將自訂的快速設定 CloudFormation 範本部署至與 S3 儲存貯體相同的帳戶：
   - 針對 `CloudWatchConfigBucketARN` 參數，輸入 S3 儲存貯體的 ARN。
   - 根據您要為 Systems Manager 啟用的功能，調整參數選項。
7. 使用和不使用 IAM 角色部署測試 EC2 執行個體，以確認 EC2 執行個體可與 CloudWatch 搭配使用。

- 套用`AttachIAMToInstance`狀態管理員關聯。這是設定為依排程執行的 Systems Manager Runbook。使用 Runbook 的 State Manager 關聯不會自動套用至新的 EC2 執行個體，並可設定為按排程執行。如需詳細資訊，請參閱 Systems Manager 文件中的使用狀態管理員執行具有觸發條件的自動化。
- 確認 EC2 執行個體已連接必要的 IAM 角色。
- 透過確認 EC2 執行個體顯示在 Systems Manager 中，確認 Systems Manager 代理程式正常運作。
- 根據來自 S3 儲存貯體的 CloudWatch 組態，檢視 CloudWatch 日誌和指標，確認 CloudWatch 代理程式正常運作。

使用 CloudFormation StackSets 在多個區域和多個帳戶中自訂快速設定

如果您使用多個帳戶和區域，則可以將 AWS-QuickSetup-SSMHostMgmt.yaml CloudFormation 範本部署為堆疊集。您必須先完成 CloudFormation StackSet 先決條件，才能使用堆疊集。這些需求會根據您要部署具有自我管理或服務管理許可的堆疊集而有所不同。

我們建議您部署具有服務受管許可的堆疊集，讓新帳戶自動接收自訂的快速設定。您必須從 AWS Organizations 管理帳戶或委派管理員帳戶部署服務受管堆疊集。您應該從用於自動化的集中式帳戶部署堆疊集，該帳戶具有委派的管理員權限，而不是 AWS Organizations 管理帳戶。我們也建議您使用單一區域中的單一或少量帳戶，以測試組織單位 (OU) 為目標來測試堆疊集部署。

1. 從本指南的 使用 CloudFormation 堆疊在單一帳戶和區域中自訂快速設定區段完成步驟 1 到 5。
2. 登入 AWS 管理主控台，開啟 CloudFormation 主控台，然後選擇建立 StackSet：
   - 選擇範本已就緒，並上傳範本檔案。上傳您根據需求自訂的 CloudFormation 範本。

- 指定堆疊集詳細資訊：
  - 輸入堆疊集名稱，例如 `StackSet-SSM-QuickSetup`。
  - 根據您要為 Systems Manager 啟用的功能，調整參數選項。
  - 針對 `CloudWatchConfigBucketARN` 參數，輸入 CloudWatch 組態 S3 儲存貯體的 ARN。
  - 指定堆疊集選項，選擇是否將服務受管許可與 AWS Organizations 或自我管理許可搭配使用。
    - 如果您選擇自我管理許可，請輸入 AWSCloudFormationStackSetAdministrationRole 和 AWSCloudFormationStackSetExecutionRole IAM 角色詳細資訊。管理員角色必須存在於帳戶中，而執行角色必須存在於每個目標帳戶中
  - 對於使用 的服務受管許可 AWS Organizations，我們建議您先部署到測試 OU，而不是整個組織。
    - 選擇是否要啟用自動部署。建議您選擇已啟用。對於帳戶移除行為，建議設定為刪除堆疊。
  - 針對自我管理許可，輸入您要設定 AWS 之帳戶的帳戶 IDs。如果您使用自我管理許可，則必須為每個新帳戶重複此程序。
  - 輸入您將使用 CloudWatch 和 Systems Manager 的區域。
  - 在堆疊集的操作和堆疊執行個體索引標籤中檢視狀態，確認部署成功。
  - 依照本指南 使用 CloudFormation 堆疊在單一帳戶和區域中自訂快速設定 一節的步驟 7，測試 Systems Manager 和 CloudWatch 在已部署帳戶中是否正常運作。

設定內部部署伺服器的考量事項

現場部署伺服器和 VMs 的 CloudWatch 代理程式是使用與 EC2 執行個體類似的方法來安裝和設定。不過，下表提供在內部部署伺服器和 VMs 上安裝和設定 CloudWatch 代理程式時必須評估的考量。

將 CloudWatch 代理程式指向用於 Systems Manager 的相同臨時登入資料。	當您在包含內部部署伺服器的混合環境中設定 Systems Manager 時，可以使用 IAM 角色啟用 Systems Manager。您應該使用為 EC2 執行個體建立的角色，其中包含 `CloudWatchAgentServerPolicy` 和 `AmazonSSMManagedInstanceCore` 政策。 這會導致 Systems Manager 代理程式擷取臨時登入資料並將其寫入本機登入資料檔案。您可以將 CloudWatch 代理程式組態指向相同的檔案。您可以從使用 Systems Manager 代理程式和統

	一 CloudWatch 代理程式設定內部部署伺服器使用 程序，以僅使用 知識中心的臨時登入資料。AWS 您也可以定義個別的 Systems Manager Automation Runbook 和 State Manager 關聯，並以具有標籤的現場部署執行個體為目標，以自動化此程序。當您為現場部署執行個體建立 Systems Manager 啟用時，您應該包含一個標籤，以將執行個體識別為現場部署執行個體。
考慮使用具有 VPN 或 Direct Connect 存取 and AWS PrivateLink 的帳戶和區域。	您可以使用 AWS Direct Connect 或 AWS Virtual Private Network (Site-to-Site VPN) 在內部部署網路和虛擬私有雲端 (VPC) 之間建立私有連線。 AWS PrivateLink 會使用介面 VPC 端點建立 CloudWatch Logs 的私有連線。如果您有限制以防止資料透過公有網際網路傳送到公有服務端點，此方法會很有用。
所有指標都必須包含在 CloudWatch 組態檔案中。	Amazon EC2 包含標準指標 （ 例如 CPU 使用率 ），但必須針對內部部署執行個體定義這些指標。您可以使用單獨的平台組態檔案來定義內部部署伺服器的這些指標，然後將組態附加到平台的標準 CloudWatch 指標組態。

暫時性 EC2 執行個體的考量事項

如果 EC2 執行個體是由 Amazon EC2 Auto Scaling、Amazon EMR、Amazon EC2 Spot 執行個體或佈建，則為暫時性或暫時性執行個體 AWS Batch。Ephemeral EC2 執行個體可能會在一般日誌群組下產生非常大量的 CloudWatch 串流，而不需要有關其執行時間原始伺服器的其他資訊。

如果您使用暫時性 EC2 執行個體，請考慮在日誌群組和日誌串流名稱中新增其他動態內容資訊。例如，您可以包含 Spot 執行個體請求 ID、Amazon EMR 叢集名稱或 Amazon EC2 Auto Scaling 群組名稱。此資訊可能因新啟動的 EC2 執行個體而有所不同，您可能需要在執行時間擷取和設定。您可以在開機時撰寫 CloudWatch 代理程式組態檔案，並重新啟動代理程式以包含更新後的組態檔案。這可讓您使用動態執行時間資訊，將日誌和指標交付至 CloudWatch。

您也應該確保在暫時性 EC2 執行個體終止之前，CloudWatch 代理程式會傳送您的指標和日誌。CloudWatch 代理程式包含 `flush_interval` 參數，可設定為定義清除日誌和指標緩衝區的時間間隔。您可以根據您的工作負載降低此值，並停止 CloudWatch 代理程式，並在 EC2 執行個體終止之前強制緩衝區排清。

使用自動化解決方案來部署 CloudWatch 代理程式

如果您使用自動化解決方案 （例如 Ansible 或 Chef)，您可以利用它來自動安裝和更新 CloudWatch 代理程式。如果您使用此方法，您必須評估下列考量：

- 驗證自動化是否涵蓋您支援的OSs和作業系統版本。如果自動化指令碼不支援組織的所有OSs，您應該為不支援OSs定義替代解決方案。
- 驗證自動化解決方案是否定期檢查 CloudWatch 代理程式更新和升級。您的自動化解決方案應定期檢查 CloudWatch 代理程式的更新，或定期解除安裝和重新安裝代理程式。您可以使用排程器或自動化解決方案功能來定期檢查和更新代理程式。
- 驗證您可以確認代理程式安裝和組態合規。您的自動化解決方案應該可讓您判斷系統何時未安裝代理程式，或代理程式何時無法運作。您可以在自動化解決方案中實作通知或警示，以便追蹤失敗的安裝和組態。

使用使用者資料指令碼在執行個體佈建期間部署 CloudWatch 代理程式

如果您不打算使用 Systems Manager，並想要選擇性地將 CloudWatch 用於 EC2 執行個體，則可以使用此方法。一般而言，此方法以一次性或需要特殊化組態時使用。 AWS 提供 CloudWatch 代理程式的直接連結，可在啟動或使用者資料指令碼中下載。代理程式安裝套件可以在沒有使用者互動的情況下無提示地執行，這表示您可以在自動化部署中使用它們。如果您使用此方法，您應該評估下列考量：

- 提高使用者不會安裝代理程式或設定標準指標的風險。使用者可以佈建執行個體，而無需包含安裝 CloudWatch 代理程式的必要步驟。它們也可能錯誤設定代理程式，這可能會導致記錄和監控不一致。
- 安裝指令碼必須是作業系統特定的，且適用於不同的作業系統版本。如果您打算同時使用 Windows 和 Linux，則需要單獨的指令碼。Linux 指令碼也應該根據發行版本有不同的安裝步驟。
- 若有新版本，您必須定期更新 CloudWatch 代理程式。如果您使用 Systems Manager 搭配 State Manager，這可以自動化，但您也可以設定使用者資料指令碼在執行個體啟動時重新執行。然後 CloudWatch 代理程式會在每次重新開機時更新和重新安裝。

- 您必須自動化標準 CloudWatch 組態的擷取和應用程式。如果您使用 Systems Manager 搭配 State Manager，這可以自動化，但您也可以設定使用者資料指令碼，以在開機時擷取組態檔案，並重新啟動 CloudWatch 代理程式。

在您的 AMIs 中包含 CloudWatch 代理程式

使用此方法的優點是您不需要等待 CloudWatch 代理程式安裝和設定，而且您可以立即開始記錄和監控。這可協助您更好地監控執行個體佈建和啟動步驟，以防執行個體無法啟動。如果您不打算使用 Systems Manager 代理程式，此方法也適用。如果您使用此方法，您應該評估下列考量：

- 更新程序必須存在，因為 AMIs可能不包含最新的 CloudWatch 代理程式版本。安裝在 AMI 中的 CloudWatch 代理程式僅是上次建立 AMI 時的最新版本。您應該包含額外的方法來定期更新代理程式，以及佈建 EC2 執行個體時。如果您使用 Systems Manager，則可以使用本指南中提供使用 Systems Manager Distributor 和 State Manager 安裝 CloudWatch 代理程式的解決方案。如果您不使用 Systems Manager，則可以使用使用者資料指令碼在執行個體啟動和重新啟動時更新代理程式。
- 您的 CloudWatch 代理程式組態檔案必須在執行個體啟動時擷取。如果您不使用 Systems Manager，您可以設定使用者資料指令碼在開機時擷取組態檔案，然後重新啟動 CloudWatch 代理程式。
- 更新 CloudWatch 組態後，必須重新啟動 CloudWatch 代理程式。
- AWS 登入資料不得儲存在 AMI 中。請確定 AMI 中未存放任何本機 AWS 登入資料。如果您使用 Amazon EC2，您可以將必要的 IAM 角色套用至執行個體，並避免本機登入資料。如果您使用內部部署執行個體，您應該在啟動 CloudWatch 代理程式之前自動化或手動更新執行個體登入資料。

在 Amazon ECS 上記錄和監控

Amazon Elastic Container Service (Amazon ECS) 提供兩種啟動類型，用於執行容器，並決定託管任務和服務的基礎結構類型；這些啟動類型為 AWS Fargate 和 Amazon EC2。這兩種啟動類型都與 CloudWatch 整合，但組態和支援會有所不同。

下列各節可協助您了解如何使用 CloudWatch 在 Amazon ECS 上記錄和監控。

主題

- 使用 EC2 啟動類型設定 CloudWatch
- EC2 和 Fargate 啟動類型的 Amazon ECS 容器日誌
- 搭配 FireLens for Amazon ECS 使用自訂日誌路由
- Amazon ECS 的指標

使用 EC2 啟動類型設定 CloudWatch

使用 EC2 啟動類型，您可以佈建使用 CloudWatch 代理程式記錄和監控的 EC2 執行個體 Amazon ECS 叢集。Amazon ECS 最佳化 AMI 已預先安裝 Amazon ECS 容器代理程式，並為 Amazon ECS 叢集提供 CloudWatch 指標。

這些預設指標包含在 Amazon ECS 的成本中，但 Amazon ECS 的預設組態不會監控日誌檔案或其他指標（例如，可用磁碟空間）。您可以使用 AWS 管理主控台 來佈建具有 EC2 啟動類型的 Amazon ECS 叢集，這會建立 CloudFormation 堆疊，以部署具有啟動組態的 Amazon EC2 Auto Scaling 群組。不過，這種方法表示您無法選擇自訂 AMI，也無法使用不同的設定或其他開機指令碼來自訂啟動組態。

若要監控其他日誌和指標，您必須在 Amazon ECS 容器執行個體上安裝 CloudWatch 代理程式。您可以從本指南的 使用 Systems Manager Distributor 和 State Manager 安裝 CloudWatch 代理程式區段使用 EC2 執行個體的安裝方法。不過，Amazon ECS AMI 不包含必要的 Systems Manager 代理程式。當您建立 Amazon ECS 叢集時，您應該使用自訂啟動組態搭配安裝 Systems Manager 代理程式的使用者資料指令碼。這可讓您的容器執行個體向 Systems Manager 註冊，並套用 State Manager 關聯來安裝、設定和更新 CloudWatch 代理程式。當 State Manager 執行和更新 CloudWatch 代理程式組態時，也會套用 Amazon EC2 的標準系統層級 CloudWatch 組態。您也可以將 Amazon ECS 的標準化 CloudWatch 組態存放在 CloudWatch 組態的 S3 儲存貯體中，並使用狀態管理員自動套用。

您應該確保套用至 Amazon ECS 容器執行個體的 IAM 角色或執行個體描述檔包含必要的 `CloudWatchAgentServerPolicy`和 `AmazonSSMManagedInstanceCore`政策。您可以使用

ecs_cluster_with_cloudwatch_linux.yaml CloudFormation 範本來佈建 Linux 型 Amazon ECS 叢集。此範本會建立具有自訂啟動組態的 Amazon ECS 叢集，該組態會安裝 Systems Manager 並部署自訂 CloudWatch 組態，以監控 Amazon ECS 特定的日誌檔案。

您應該擷取 Amazon ECS 容器執行個體的下列日誌，以及標準 EC2 執行個體日誌：

- Amazon ECS 代理程式啟動輸出 – `/var/log/ecs/ecs-init.log`
- Amazon ECS 代理程式輸出 – `/var/log/ecs/ecs-agent.log`
- IAM 登入資料提供者請求日誌 – `/var/log/ecs/audit.log`

如需輸出層級、格式和其他組態選項的詳細資訊，請參閱 Amazon ECS 文件中的 Amazon ECS 日誌檔案位置。

> **Important**
>
> Fargate 啟動類型不需要代理程式安裝或組態，因為您不執行或管理 EC2 容器執行個體。

Amazon ECS 容器執行個體應使用最新的 Amazon ECS 最佳化 AMIs 和容器代理程式。 會將公有 Systems Manager 參數存放區參數與 Amazon ECS 最佳化 AMI 資訊一起 AWS 存放，包括 AMI ID。您可以使用 Amazon ECS 最佳化 AMI 的參數存放區參數格式，從參數存放區擷取最新的最佳化 AMIs。您可以參考公有參數存放區參數，參考範本中 CloudFormation 最新的 AMI 或特定 AMI 版本。

AWS 會在每個支援的區域中提供相同的參數存放區參數。這表示參考這些參數的 CloudFormation 範本可以在區域和帳戶之間重複使用，而無需更新 AMI。您可以參考特定版本來控制將較新的 Amazon ECS AMIs 部署到您的組織，這可協助您在測試之前避免使用新的 Amazon ECS 最佳化 AMI。

EC2 和 Fargate 啟動類型的 Amazon ECS 容器日誌

Amazon ECS 使用任務定義，將容器部署和管理為任務和服務。您可以在任務定義中設定要在 Amazon ECS 叢集中啟動的容器。記錄是在容器層級使用日誌驅動程式設定。視您使用的是 EC2 或 Fargate 啟動類型而定，多個日誌驅動程式選項為您的容器提供不同的日誌系統（例如 `syslog`、、`splunk`、`awslogs`、`fluentd`、`gelf`、`json-file`、`journald`、`logentries`、、 或 `awsfirelens`)。Fargate 啟動類型提供下列日誌驅動程式選項的子集：`awslogs`、 `splunk`和 `awsfirelens`。 AWS 提供`awslogs`日誌驅動程式，以擷取容器輸出並將其傳輸至 CloudWatch Logs。日誌驅動程式設定可讓您自訂日誌群組、區域和日誌串流字首，以及許多其他選項。

日誌群組的預設命名，以及 上自動設定 CloudWatch Logs AWS 管理主控台 選項所使用的選項為 `/ecs/<task_name>`。Amazon ECS 使用的日誌串流名稱具有 `<awslogs-stream-prefix>/<container_name>/<task_id>` 格式。我們建議您使用群組名稱，根據組織的需求將日誌分組。在下表中， `image_name`和 `image_tag`包含在日誌串流的名稱中。

日誌群組名稱	`/<Business unit>/<Project or application name>/<Environment>/<Cluster name>/<Task name>`
日誌串流名稱字首	`/<image_name>/<image_tag>`

此資訊也可在任務定義中使用。不過，任務會定期更新為新的修訂版，這表示任務定義可能使用不同於`image_name`image_tag任務定義目前使用的版本。如需詳細資訊和命名建議，請參閱本指南的 規劃 CloudWatch 部署一節。

如果您使用持續整合和持續交付 (CI/CD) 管道或自動化程序，您可以使用每個新的 Docker 映像組建為您的應用程式建立新的任務定義修訂。例如，您可以在任務定義修訂和記錄組態中包含 Docker 映像名稱、映像標籤、GitHub 修訂版或其他重要資訊，做為 CI/CD 程序的一部分。

搭配 FireLens for Amazon ECS 使用自訂日誌路由

FireLens for Amazon ECS 可協助您將日誌路由到 Fluentd 或 Fluent Bit，以便您可以將容器日誌直接傳送到 AWS 服務和 AWS 合作夥伴網路 (APN) 目的地，以及支援將日誌運送到 CloudWatch Logs。

AWS 為 Fluent Bit 提供 Docker 映像，其中包含 Amazon Kinesis Data Streams、Amazon Data Firehose 和 CloudWatch Logs 的預先安裝外掛程式。您可以使用 FireLens 日誌驅動程式，而不是`awslogs`日誌驅動程式，以進一步自訂和控制傳送至 CloudWatch Logs 的日誌。

例如，您可以使用 FireLens 日誌驅動程式來控制日誌格式輸出。這表示 Amazon ECS 容器的 CloudWatch 日誌會自動格式化為 JSON 物件`container_name`，並包含 `ecs_cluster`、`ecs_task_arn`、`ecs_task_definition`container_id、 和 的 JSON 格式屬性`ec2_instance_id`。當您指定`awsfirelens`驅動程式時，流利的主機會透過 `FLUENT_HOST`和 `FLUENT_PORT`環境變數公開到您的容器。這表示您可以使用流暢的日誌程式庫，從程式碼直接登入日誌路由器。例如，您的應用程式可能包含程式`fluent-logger-python`庫，使用環境變數提供的值來記錄到 Fluent Bit。

如果您選擇使用 FireLens for Amazon ECS，您可以設定與`awslogs`日誌驅動程式相同的設定，也可以使用其他設定。例如，您可以使用 ecs-task-nginx-firelense.json Amazon ECS 任務定義，啟動設定

為使用 FireLens 記錄至 CloudWatch 的 NGINX 伺服器。它也會啟動 FireLens Fluent Bit 容器做為記錄的附屬項目。

Amazon ECS 的指標

Amazon ECS 使用 Amazon ECS 容器代理程式，在叢集和服務層級為 EC2 和 Fargate 啟動類型提供標準 CloudWatch 指標（例如 CPU 和記憶體使用率）。 EC2 您也可以使用 CloudWatch Container Insights 擷取服務、任務和容器的指標，或使用內嵌指標格式擷取您自己的自訂容器指標。

Container Insights 是一種 CloudWatch 功能，可在叢集、容器執行個體、服務和任務層級提供 CPU 使用率、記憶體使用率、網路流量和儲存空間等指標。Container Insights 也會建立自動儀表板，協助您分析服務和任務，並查看容器層級的平均記憶體或 CPU 使用率。Container Insights 會將自訂指標發佈到自訂`ECS/ContainerInsights`命名空間，可用於繪製圖形、警示和儀表板。

您可以為每個個別 Amazon ECS 叢集啟用 Container Insights，以開啟 Container Insight 指標。如果您也想要在容器執行個體層級查看指標，您可以啟動 CloudWatch 代理程式做為 Amazon ECS 叢集上的協助程式容器。您可以使用 cwagent-ecs-instance-metric-cfn.yaml CloudFormation 範本，將 CloudWatch 代理程式部署為 Amazon ECS 服務。重要的是，此範例假設您已建立適當的自訂 CloudWatch 代理程式組態，並將其存放在具有金鑰 的參數存放區中`ecs-cwagent-daemon-service`。

部署為 CloudWatch Container Insights 協助程式容器的 CloudWatch 代理程式包含其他磁碟、記憶體和 CPU 指標，例如 `instance_cpu_reserved_capacity`和 `instance_memory_reserved_capacity`搭配 `ClusterName`、`ContainerInstanceId`、`InstanceId`維度。 CloudWatch Container Insights 會使用 CloudWatch 內嵌指標格式來實作容器執行個體層級的指標。您可以使用本指南 設定 CloudWatch 代理程式部署和組態的 State Manager 和 Distributor區段中的 方法，為您的 Amazon ECS 容器執行個體設定其他系統層級指標。

在 Amazon ECS 中建立自訂應用程式指標

您可以使用 CloudWatch 內嵌指標格式，為您的應用程式建立自訂指標。`awslogs` 日誌驅動程式可以解譯 CloudWatch 內嵌指標格式陳述式。

下列範例中`CW_CONFIG_CONTENT`的環境變數設定為 `cwagentconfig` Systems Manager 參數存放區參數的內容。您可以使用此基本組態執行代理程式，將其設定為內嵌指標格式端點。不過，已不再需要。

```
{
```

```
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

如果您有跨多個帳戶和區域的 Amazon ECS 部署，您可以使用 AWS Secrets Manager 秘密來存放 CloudWatch 組態，並設定秘密政策以與您的組織共用。您可以使用任務定義中的秘密選項來設定CW_CONFIG_CONTENT變數。

您可以在應用程式中使用 AWS 提供的開放原始碼內嵌指標格式程式庫，並指定AWS_EMF_AGENT_ENDPOINT環境變數，以連接至做為內嵌指標格式端點的 CloudWatch 代理程式附屬容器。例如，您可以使用 ecs_cw_emf_example 範例 Python 應用程式，將內嵌指標格式的指標傳送至設定為內嵌指標格式端點的 CloudWatch 代理程式附屬容器。

適用於 CloudWatch 的 Fluent Bit 外掛程式也可用於傳送內嵌指標格式訊息。您也可以使用 ecs_firelense_emf_example 範例 Python 應用程式，將內嵌指標格式的指標傳送至 Amazon ECS 附屬容器的 Firelens。

如果您不想使用內嵌指標格式，您可以透過 AWS API 或 AWS SDK 建立和更新 CloudWatch 指標。除非您有特定的使用案例，否則我們不建議使用此方法，因為它會為您的程式碼增加維護和管理開銷。

在 Amazon EKS 上記錄和監控

Amazon Elastic Kubernetes Service (Amazon EKS) 與 Kubernetes 控制平面的 CloudWatch Logs 整合。控制平面由 Amazon EKS 做為受管服務提供，您可以開啟記錄功能，而無需安裝 CloudWatch 代理程式。也可以部署 CloudWatch 代理程式來擷取 Amazon EKS 節點和容器日誌。Fluent Bit 和 Fluentd 也支援將您的容器日誌傳送至 CloudWatch Logs。

CloudWatch Container Insights 在叢集、節點、Pod、任務和服務層級為 Amazon EKS 提供全面的指標監控解決方案。Amazon EKS 也支援使用 Prometheus 擷取指標的多個選項。Amazon EKS 控制平面提供指標端點，以 Prometheus 格式公開指標。您可以將 Prometheus 部署到您的 Amazon EKS 叢集，以使用這些指標。

您也可以設定 CloudWatch 代理程式來抓取 Prometheus 指標，並建立 CloudWatch 指標，以及使用其他 Prometheus 端點。Prometheus 的 Container Insights 監控也可以自動從支援的容器化工作負載和系統探索和擷取 Prometheus 指標。

您可以在 Amazon EKS 節點上安裝和設定 CloudWatch 代理程式，方法類似於搭配 Distributor 和 State Manager 用於 Amazon EC2 的方法，讓您的 Amazon EKS 節點與您的標準系統記錄和監控組態保持一致。

Amazon EKS 的日誌記錄

Kubernetes 記錄可以分為控制平面記錄、節點記錄和應用程式記錄。Kubernetes 控制平面是一組元件，可管理 Kubernetes 叢集並產生用於稽核和診斷目的的日誌。使用 Amazon EKS，您可以開啟不同控制平面元件的日誌，並將其傳送至 CloudWatch。

Kubernetes 也會在每個執行 Pod 的 Kubernetes 節點`kube-proxy`上執行系統元件，例如`kubelet`和 。這些元件會在每個節點內寫入日誌，您可以設定 CloudWatch 和 Container Insights 來擷取每個 Amazon EKS 節點的這些日誌。

容器會分組為 Kubernetes 叢集內的 Pod，並排定在您的 Kubernetes 節點上執行。大多數容器化應用程式會寫入標準輸出和標準錯誤，而容器引擎會將輸出重新導向至記錄驅動程式。在 Kubernetes 中，容器日誌位於節點的 `/var/log/pods`目錄中。您可以設定 CloudWatch 和 Container Insights 來擷取每個 Amazon EKS Pod 的這些日誌。

Amazon EKS 控制平面記錄

Amazon EKS 叢集包含 Kubernetes 叢集的高可用性、單一租用戶控制平面，以及執行容器的 Amazon EKS 節點。控制平面節點會在 管理的 帳戶中執行 AWS。Amazon EKS 叢集控制平面節點已與 CloudWatch 整合，您可以開啟特定控制平面元件的記錄。

日誌會針對每個 Kubernetes 控制平面元件執行個體提供。 會 AWS 管理控制平面節點的運作狀態，並為 Kubernetes 端點提供服務層級協議 (SLA)。

Amazon EKS 節點和應用程式記錄

我們建議您使用 CloudWatch Container Insights 來擷取 Amazon EKS 的日誌和指標。Container Insights 使用 CloudWatch 代理程式和 Fluent Bit 或 Fluentd 實作叢集、節點和 Pod 層級指標，以將日誌擷取至 CloudWatch。Container Insights 也提供自動儀表板，其中包含所擷取 CloudWatch 指標的分層檢視。Container Insights 部署為在每個 Amazon EKS 節點上執行的 CloudWatch DaemonSet 和 Fluent Bit DaemonSet。Container Insights 不支援 Fargate 節點，因為節點是由 管理 AWS 且不支援 DaemonSets。本指南會分別介紹 Amazon EKS 的 Fargate 記錄。

下表顯示 Amazon EKS 預設 Fluentd 或 Fluent Bit 日誌擷取組態擷取的 CloudWatch 日誌群組和日誌。 https://docs.aws.amazon.com//AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html

`/aws/containerinsights/Cluster_Name/application`	中的所有日誌檔案`/var/log/containers` 。此目錄提供`/var/log/pods` 目錄結構中所有 Kubernetes 容器日誌的符號連結。這會擷取寫入`stdout`或 的應用程式容器日誌`stderr`。它還包含 Kubernetes 系統容器的日誌`aws-vpc-cni-init` ，例如 `kube-proxy` 、 和 `coreDNS`。
`/aws/containerinsights/Cluster_Name/host`	來自 `/var/log/dmesg` 、 `/var/log/secure` 和 的日誌`/var/log/messages` 。
`/aws/containerinsights/Cluster_Name/dataplane`	`/var/log/journal` 中適用於 `kubelet.service` 、`kubeproxy`

	.service 和 docker.service 的日誌。

如果您不想使用 Container Insights 搭配 Fluent Bit 或 Fluentd 進行記錄，您可以使用安裝在 Amazon EKS 節點上的 CloudWatch 代理程式來擷取節點和容器日誌。Amazon EKS 節點是 EC2 執行個體，這表示您應該將其包含在 Amazon EC2 的標準系統層級記錄方法中。如果您使用 Distributor 和 State Manager 安裝 CloudWatch 代理程式，Amazon EKS 節點也會包含在 CloudWatch 代理程式安裝、組態和更新中。

下表顯示 Kubernetes 特有的日誌，如果您未使用 Container Insights 搭配 Fluent Bit 或 Fluentd 進行記錄，則必須擷取這些日誌。

/var/log/containers	此目錄提供/var/log/pods 目錄結構下所有 Kubernetes 容器日誌的符號連結。這可有效地擷取寫入 stdout或 的應用程式容器日誌stderr。這包括 Kubernetes 系統容器的日誌aws-vpc-cni-init ，例如 kube-proxy 、 和 coreDNS。重要：如果您使用 Container Insights，則不需要這樣做。
var/log/aws-routed-eni/ipamd.log /var/log/aws-routed-eni/plugin.log	您可以在此處找到 L-IPAM 協助程式的日誌

您必須確保 Amazon EKS 節點安裝並設定 CloudWatch 代理程式，以傳送適當的系統層級日誌和指標。不過，Amazon EKS 最佳化 AMI 不包含 Systems Manager 代理程式。透過使用啟動範本，您可以自動化 Systems Manager 代理程式安裝和預設 CloudWatch 組態，以透過使用者資料區段實作的啟動指令碼擷取重要的 Amazon EKS 特定日誌。Amazon EKS 節點使用 Auto Scaling 群組做為受管節點群組或自我管理節點進行部署。

使用受管節點群組，您可以提供包含使用者資料區段的啟動範本，以自動化 Systems Manager 代理程式安裝和 CloudWatch 組態。您可以自訂並使用 amazon_eks_managed_node_group_launch_config.yaml CloudFormation 範本來建立啟動範本，以安裝 Systems Manager 代理程式、CloudWatch 代理程式，也可以將 Amazon EKS 特定記錄組態新增至 CloudWatch 組態目錄。此範本可用於使用infrastructure-as-code(IaC) 方法更新 Amazon

EKS 受管節點群組啟動範本。範本的每次更新都會 CloudFormation 佈建新版本的啟動範本。然後，您可以更新節點群組以使用新的範本版本，並讓受管生命週期程序更新您的節點，而不會停機。確定套用至受管節點群組的 IAM 角色和執行個體描述檔包含 `CloudWatchAgentServerPolicy`和 `AmazonSSMManagedInstanceCore` AWS 受管政策。

透過自我管理節點，您可以直接佈建和管理 Amazon EKS 節點的生命週期和更新策略。自我管理節點可讓您在 Amazon EKS 叢集和 Bottlerocket 上執行 Windows 節點，以及其他選項。您可以使用 CloudFormation 將自我管理節點部署到 Amazon EKS 叢集，這表示您可以使用 Amazon EKS 叢集的 IaC 和受管變更方法。 AWS 提供 amazon-eks-nodegroup.yaml CloudFormation 範本，您可以依原狀使用或自訂。範本為叢集中的 Amazon EKS 節點佈建所有必要的資源 （例如，單獨的 IAM 角色、安全群組、Amazon EC2 Auto Scaling 群組和啟動範本）。amazon-eks-nodegroup.yaml CloudFormation 範本是更新的版本，可安裝所需的 Systems Manager 代理程式、CloudWatch 代理程式，並將 Amazon EKS 特定的記錄組態新增至 CloudWatch 組態目錄。

記錄 Fargate 上的 Amazon EKS

使用 Fargate 上的 Amazon EKS，您可以部署 Pod，而無需配置或管理您的 Kubernetes 節點。這樣就不需要擷取 Kubernetes 節點的系統層級日誌。若要從 Fargate Pod 擷取日誌，您可以使用 Fluent Bit 將日誌直接轉送至 CloudWatch。這可讓您自動將日誌路由到 CloudWatch，而無需進一步設定或 Fargate 上 Amazon EKS Pod 的附屬容器。如需詳細資訊，請參閱 Amazon EKS 文件中的 Fargate 記錄，以及 AWS 部落格上的適用於 Amazon EKS 的 Fluent Bit。此解決方案會從容器擷取 `STDOUT` 和`STDERR`輸入/輸出 (I/O) 串流，並根據 Fargate 上為 Amazon EKS 叢集建立的 Fluent Bit 組態，透過 Fluent Bit 將其傳送至 CloudWatch。

Amazon EKS 和 Kubernetes 的指標

Kubernetes 提供指標 API，可讓您存取資源用量指標 （例如節點和 Pod 的 CPU 和記憶體用量），但 API 僅提供point-in-time資訊，不提供歷史指標。Kubernetes 指標伺服器通常用於 Amazon EKS 和 Kubernetes 部署，以彙總指標、提供有關指標的短期歷史資訊，以及支援 Horizontal Pod Autoscaler 等功能。

Amazon EKS 透過 Kubernetes API 伺服器以 Prometheus 格式公開控制平面指標，CloudWatch 可以擷取和擷取這些指標。CloudWatch 和 Container Insights 也可以設定為為您的 Amazon EKS 節點和 Pod 提供全面的指標擷取、分析和警示。

Kubernetes 控制平面指標

Kubernetes 使用 `/metrics` HTTP API 端點，以 Prometheus 格式公開控制平面指標。您應該在 Kubernetes 叢集中安裝 Prometheus，以使用 Web 瀏覽器繪製和檢視這些指標。您也可以將 Kubernetes API 伺服器公開的指標擷取至 CloudWatch。

Kubernetes 的節點和系統指標

Kubernetes 提供 Prometheus 指標伺服器 Pod，您可以在叢集、節點和 Pod 層級 CPU 和記憶體統計資料的 Kubernetes 叢集上部署和執行。這些指標會與 Horizontal Pod Autoscaler 和 Vertical Pod Autoscaler 搭配使用。CloudWatch 也可以提供這些指標。

如果您使用 Kubernetes Dashboard 或水平和垂直 Pod 自動擴展器，則應安裝 Kubernetes Metrics Server。Kubernetes Dashboard 可協助您瀏覽和設定 Kubernetes 叢集、節點、Pod 和相關組態，並從 Kubernetes Metrics Server 檢視 CPU 和記憶體指標。

Kubernetes Metrics Server 提供的指標無法用於非自動擴展目的（例如監控）。這些指標適用於point-in-time分析，而非歷史分析。Kubernetes Dashboard 會部署 `dashboard-metrics-scraper`，以在短時間內從 Kubernetes Metrics Server 存放指標。

Container Insights 使用在 Kubernetes DaemonSet 中執行的 CloudWatch 代理程式的容器化版本，來探索叢集中的所有執行中容器，並提供節點層級指標。它會收集效能堆疊每一層的效能資料。您可以使用 Quick Starts 中的 AWS Quick Start 或分別設定 Container Insights。Quick Start 會使用 CloudWatch 代理程式設定指標監控，並使用 Fluent Bit 記錄，因此您只需要部署一次即可記錄和監控。

由於 Amazon EKS 節點是 EC2 執行個體，除了 Container Insights 擷取的指標之外，您應該使用為 Amazon EC2 定義的標準來擷取系統層級指標。您可以從本指南的 設定 CloudWatch 代理程式部署和組態的 State Manager 和 Distributor區段使用相同的方法來安裝和設定 Amazon EKS 叢集的 CloudWatch 代理程式。您可以更新 Amazon EKS 特定的 CloudWatch 組態檔案，以包含指標以及 Amazon EKS 特定的日誌組態。

具有 Prometheus 支援的 CloudWatch 代理程式可以自動從支援的容器化工作負載和系統中探索和抓取 Prometheus 指標。它以內嵌指標格式將其擷取為 CloudWatch Logs Insights 進行分析，並自動建立 CloudWatch 指標。

> **Important**
>
> 您必須部署 CloudWatch 代理程式的專用版本，才能收集 Prometheus 指標。 CloudWatch 這是與針對 Container Insights 部署的 CloudWatch 代理程式不同的代理程式。您可以使用

prometheus_jmx 範例 Java 應用程式，其中包含 CloudWatch 代理程式和 Amazon EKS Pod 部署的部署和組態檔案，以示範 Prometheus 指標探索。如需詳細資訊，請參閱 CloudWatch 文件中的在 Amazon EKS 和 Kubernetes 上設定 Java/JMX 範例工作負載。您也可以設定 CloudWatch 代理程式，從 Amazon EKS 叢集中執行的其他 Prometheus 目標擷取指標。

應用程式指標

您可以使用 CloudWatch 內嵌指標格式建立自己的自訂指標。若要擷取內嵌指標格式陳述式，您需要將內嵌指標格式項目傳送至內嵌指標格式端點。CloudWatch 代理程式可以設定為 Amazon EKS Pod 中的附屬容器。CloudWatch 代理程式組態會儲存為 Kubernetes ConfigMap，並由 CloudWatch 代理程式附屬容器讀取，以啟動內嵌指標格式端點。

您也可以將應用程式設定為 Prometheus 目標，並透過 Prometheus 支援設定 CloudWatch 代理程式，以探索、抓取指標並將其擷取至 CloudWatch。例如，您可以將開放原始碼 JMX 匯出工具與 Java 應用程式搭配使用，以公開 CloudWatch 代理程式用於 Prometheus 消耗的 JMX Beans。

如果您不想使用內嵌指標格式，也可以使用 AWS API 或 AWS SDK 建立和更新 CloudWatch 指標。不過，我們不建議使用此方法，因為它會混合監控和應用程式邏輯。

Fargate 上的 Amazon EKS 指標

Fargate 會自動佈建 Amazon EKS 節點來執行 Kubernetes Pod，因此您不需要監控和收集節點層級指標。不過，您必須監控 Fargate 上 Amazon EKS 節點上執行的 Pod 指標。Container Insights 目前不適用於 Fargate 上的 Amazon EKS，因為它需要下列目前不支援的功能：

- 目前不支援 DaemonSets。Container Insights 透過在每個叢集節點上執行 CloudWatch 代理程式做為 DaemonSet 部署。
- 不支援 HostPath 持久性磁碟區。CloudWatch 代理程式容器使用 hostPath 持久性磁碟區做為收集容器指標資料的先決條件。
- Fargate 可防止特權容器和主機資訊的存取。

您可以使用 Fargate 的內建日誌路由器，將內嵌指標格式陳述式傳送至 CloudWatch。日誌路由器使用 Fluent Bit，其具有可設定為支援內嵌指標格式陳述式的 CloudWatch 外掛程式。

您可以在 Amazon EKS 叢集中部署 Prometheus 伺服器，以從 Fargate 節點收集指標，藉此擷取和擷取 Fargate 節點的 Pod 層級指標。由於 Prometheus 需要持久性儲存，如果您使用 Amazon Elastic File System (Amazon EFS) 進行持久性儲存，則可以在 Fargate 上部署 Prometheus。您也可以在

Amazon EC2 支援的節點上部署 Prometheus。如需詳細資訊，請參閱部落格上的 AWS [AWS Fargate 使用 Prometheus 和 Grafana 監控 上的 Amazon EKS]。

Amazon EKS 上的 Prometheus 監控

Amazon Managed Service for Prometheus 為開放原始碼 Prometheus 提供可擴展、安全、受 AWS 管的服務。您可以使用 Prometheus 查詢語言 (PromQL) 來監控容器化工作負載的效能，而無需管理基礎基礎設施來擷取、儲存和查詢操作指標。您可以使用 AWS Distro for OpenTelemetry (ADOT) 或 Prometheus 伺服器做為收集代理程式，從 Amazon EKS 和 Amazon ECS 收集 Prometheus 指標。

Prometheus 的 CloudWatch Container Insights 監控可讓您設定和使用 CloudWatch 代理程式，從 Amazon ECS、Amazon EKS 和 Kubernetes 工作負載探索 Prometheus 指標，並將其擷取為 CloudWatch 指標。如果 CloudWatch 是您的主要可觀測性和監控解決方案，則此解決方案是適當的。不過，以下清單概述了 Amazon Managed Service for Prometheus 為擷取、儲存和查詢 Prometheus 指標提供更多彈性的使用案例：

- Amazon Managed Service for Prometheus 可讓您使用 Amazon EKS 或自我管理 Kubernetes 中部署的現有 Prometheus 伺服器，並將其設定為寫入 Amazon Managed Service for Prometheus，而不是本機設定的資料存放區。這消除了為 Prometheus 伺服器及其基礎設施管理高可用性資料存放區的繁重工作。當您想要在 AWS 雲端中利用成熟的 Prometheus 部署時，Amazon Managed Service for Prometheus 是適合的選擇。
- Grafana 直接支援 Prometheus 做為視覺化的資料來源。如果您想要使用 Grafana 搭配 Prometheus 而非 CloudWatch Dashboards 進行容器監控，則 Amazon Managed Service for Prometheus 可以滿足您的需求。Amazon Managed Service for Prometheus 與 Amazon Managed Grafana 整合，以提供受管的開放原始碼監控和視覺化解決方案。
- Prometheus 可讓您使用 PromQL 查詢，對操作指標進行分析。相反地，CloudWatch 代理程式會將內嵌指標格式的 Prometheus 指標擷取至 CloudWatch Logs，進而產生 CloudWatch 指標。您可以使用 CloudWatch Logs Insights 查詢內嵌指標格式日誌。
- 如果您不打算使用 CloudWatch 進行監控和指標擷取，則應搭配 Prometheus 伺服器和視覺化解決方案使用 Amazon Managed Service for Prometheus，例如 Grafana。您需要將 Prometheus 伺服器設定為從 Prometheus 目標抓取指標，並將伺服器設定為遠端寫入 Amazon Managed Service for Prometheus 工作區。如果您使用 Amazon Managed Grafana，則可以使用隨附的外掛程式，直接將 Amazon Managed Grafana 與您的 Amazon Managed Service for Prometheus 資料來源整合。由於指標資料存放在 Amazon Managed Service for Prometheus 中，因此沒有部署 CloudWatch 代理程式或將資料擷取至 CloudWatch 的需求。Prometheus 的 Container Insights 監控需要 CloudWatch 代理程式。

您也可以使用 ADOT 收集器從 Prometheus 檢測的應用程式進行湊集，並將指標傳送至 Amazon Managed Service for Prometheus。如需 ADOT Collector 的詳細資訊，請參閱 AWS Distro for OpenTelemetry 文件。

的日誌記錄和指標 AWS Lambda

Lambda 不需要為工作負載管理和監控伺服器，並自動使用 CloudWatch Metrics 和 CloudWatch Logs，而無需進一步設定或檢測應用程式的程式碼。本節可協助您了解 Lambda 所用系統的效能特性，以及您的組態選擇如何影響效能。它還可協助您記錄和監控 Lambda 函數，以最佳化效能並診斷應用程式層級的問題。

Lambda 函數記錄

Lambda 會自動將標準輸出和標準錯誤訊息從 Lambda 函數串流至 CloudWatch Logs，而不需要記錄驅動程式。Lambda 也會自動佈建容器來執行 Lambda 函數，並將它們設定為在個別日誌串流中輸出日誌訊息。

Lambda 函數的後續調用可以重複使用相同的容器和輸出至相同的日誌串流。Lambda 也可以佈建新的容器，並將調用輸出到新的日誌串流。

第一次叫用 Lambda 函數時，Lambda 會自動建立日誌群組。Lambda 函數可以有多個版本，您可以選擇要執行的版本。Lambda 函數調用的所有日誌都會存放在相同的日誌群組中。名稱無法變更，且格式為 `/aws/lambda/<YourLambdaFunctionName>` 。每個 Lambda 函數執行個體的日誌群組中都會建立個別的日誌串流。Lambda 具有使用 `YYYY/MM/DD/[<FunctionVersion>]<InstanceId>` 格式之日誌串流的標準命名慣例。由 `InstanceId`產生 AWS ，以識別 Lambda 函數執行個體。

我們建議您以 JSON 格式格式化日誌訊息，因為您可以使用 CloudWatch Logs Insights 更輕鬆地查詢它們。它們也可以更輕鬆地篩選和匯出。您可以使用記錄程式庫來簡化此程序或撰寫您自己的日誌處理函數。我們建議您使用日誌程式庫來協助格式化和分類日誌訊息。例如，如果您的 Lambda 函數是以 Python 撰寫，您可以使用 Python 記錄模組來記錄訊息並控制輸出格式。Lambda 原生使用 Python 日誌程式庫處理以 Python 撰寫的 Lambda 函數，而且您可以在 Lambda 函數中擷取和自訂記錄器。AWS Labs 已建立AWS Lambda 適用於 Python 開發人員的 Powertools 工具組，讓您更輕鬆地使用冷啟動等金鑰資料來豐富日誌訊息。此工具組適用於 Python、Java、Typescript 和 .NET。

另一個最佳實務是使用 變數來設定日誌輸出層級，並根據環境和您的需求進行調整。除了使用的程式庫之外，Lambda 函數的程式碼可能會根據日誌輸出層級輸出大量日誌資料。這可能會影響您的記錄成本並影響效能。

Lambda 可讓您為 Lambda 函數執行期環境設定環境變數，而無需更新程式碼。例如，您可以建立`LAMBDA_LOG_LEVEL`環境變數，定義可從程式碼擷取的日誌輸出層級。下列範例會嘗試擷取`LAMBDA_LOG_LEVEL`環境變數，並使用 值來定義記錄輸出。如果未設定環境變數，則會預設為`INFO`層級。

```
import logging
from os import getenv

logger = logging.getLogger()
log_level = getenv("LAMBDA_LOG_LEVEL", "INFO")
level = logging.getLevelName(log_level)
logger.setLevel(level)
```

從 CloudWatch 將日誌傳送至其他目的地

您可以使用訂閱篩選條件將日誌傳送至其他目的地 （例如 Amazon OpenSearch Service 或 Lambda 函數）。如果您不使用 Amazon OpenSearch Service，您可以使用 Lambda 函數來處理日誌，並使用 AWS SDKs 將其傳送到您選擇的 AWS 服務。

您也可以在 Lambda 函數中將 SDKs用於 AWS 雲端外部的日誌目的地，以直接將日誌陳述式傳送到您選擇的目的地。如果您選擇此選項，建議您考慮延遲的影響、額外的處理時間、錯誤和重試處理，以及操作邏輯與 Lambda 函數的耦合。

Lambda 函數指標

Lambda 可讓您在不管理或擴展伺服器的情況下執行程式碼，這幾乎消除了系統層級稽核和診斷的負擔。不過，了解 Lambda 函數系統層級的效能和調用指標仍然很重要。這可協助您最佳化資源組態並改善程式碼效能。透過適當調整 Lambda 函數的大小，有效監控和測量效能可以改善使用者體驗並降低成本。一般而言，做為 Lambda 函數執行的工作負載也具有需要擷取和分析的應用程式層級指標。Lambda 直接支援內嵌指標格式，讓擷取應用程式層級 CloudWatch 指標變得更輕鬆。

系統層級指標

Lambda 會自動與 CloudWatch 指標整合，並為 Lambda 函數提供一組標準指標。Lambda 也為具有這些指標的每個 Lambda 函數提供單獨的監控儀表板。您需要監控的兩個重要指標是錯誤和叫用錯誤。了解調用錯誤和其他錯誤類型之間的差異，可協助您診斷和支援 Lambda 部署。

叫用錯誤會導致 Lambda 函數無法執行。這些錯誤發生在程式碼執行之前，因此您無法在程式碼中實作錯誤處理來識別它們。反之，您應該為 Lambda 函數設定警示，以偵測這些錯誤並通知操作和工作負載擁有者。這些錯誤通常與組態或許可錯誤有關，並可能因組態或許可的變更而發生。調用錯誤可能會啟動重試，這會導致函數多次調用。

成功叫用 Lambda 函數會傳回 HTTP 200 回應，即使函數擲回例外狀況。您的 Lambda 函數應實作錯誤處理並引發例外狀況，讓`Errors`指標擷取並識別 Lambda 函數的失敗執行。您應該從 Lambda 函數叫用傳回格式化的回應，其中包含判斷執行是否完全失敗、部分失敗或成功的資訊。

CloudWatch 提供 CloudWatch Lambda Insights，您可以啟用個別 Lambda 函數。Lambda Insights 會收集、彙總和摘要系統層級指標（例如 CPU 時間、記憶體、磁碟和網路使用量）。Lambda Insights 也會收集、彙總和摘要診斷資訊（例如冷啟動和 Lambda 工作者關機），以協助您隔離並快速解決問題。

Lambda Insights 會根據 Lambda 函數的名稱，使用`/aws/lambda-insights/`日誌串流名稱字首，使用內嵌指標格式自動將效能資訊傳送到日誌群組。這些效能日誌事件會建立 CloudWatch 指標，這是自動 CloudWatch 儀表板的基礎。我們建議您為效能測試和生產環境啟用 Lambda Insights。Lambda Insights 建立的其他指標包括`memory_utilization`有助於正確調整 Lambda 函數大小，讓您避免支付不必要的容量。

應用程式指標

您也可以使用內嵌指標格式，在 CloudWatch 中建立和擷取自己的應用程式指標。您可以利用AWS 提供的內嵌指標格式程式庫，建立內嵌指標格式陳述式並將其發出至 CloudWatch。整合的 Lambda CloudWatch 記錄設施已設定為處理和擷取適當格式的內嵌指標格式陳述式。

搜尋和分析 CloudWatch 中的日誌

將日誌和指標擷取到一致的格式和位置之後，您可以搜尋和分析它們，以協助改善營運效率，以及識別和疑難排解問題。我們建議您以格式正確的格式擷取日誌（例如 JSON)，以便更輕鬆地搜尋和分析日誌。大多數工作負載都使用一組 AWS 資源，例如網路、運算、儲存和資料庫。如果可能，您應該從這些資源集中分析指標和日誌，並將其相互關聯，以有效地監控和管理所有 AWS 工作負載。

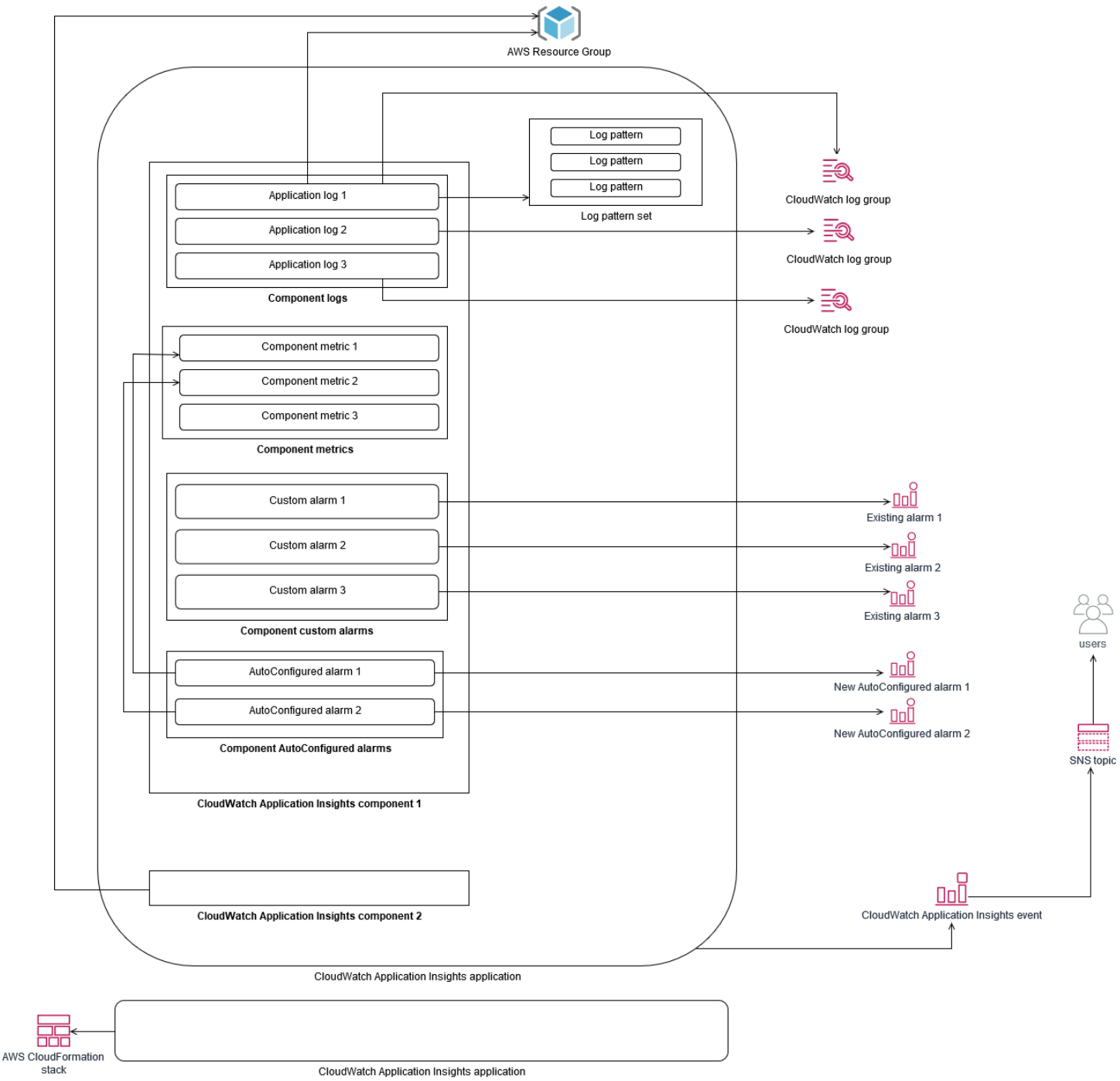
CloudWatch 提供數種功能來協助分析日誌和指標，例如 CloudWatch Application Insights 跨不同 AWS 資源共同定義和監控應用程式的指標和日誌、CloudWatch 異常偵測以呈現指標的異常，以及 CloudWatch Log Insights 以互動方式搜尋和分析 CloudWatch Logs 中的日誌資料。

使用 CloudWatch Application Insights 集體監控和分析應用程式

應用程式擁有者可以使用 Amazon CloudWatch Application Insights 來設定工作負載的自動監控和分析。除了針對 帳戶中的所有工作負載設定的標準系統層級監控之外，還可以設定此功能。透過 CloudWatch Application Insights 設定監控，也可以協助應用程式團隊主動與營運保持一致，並減少復原的平均時間 (MTTR)。CloudWatch Application Insights 可協助減少建立應用程式層級記錄和監控所需的工作量。它還提供以元件為基礎的架構，可協助團隊區分記錄和監控責任。

CloudWatch Application Insights 使用資源群組來識別應共同監控為應用程式的資源。資源群組中支援的資源會成為 CloudWatch Application Insights 應用程式個別定義的元件。CloudWatch Application Insights 應用程式的每個元件都有自己的日誌、指標和警示。

對於日誌，您可以定義應該用於元件和 CloudWatch Application Insights 應用程式內的日誌模式集。日誌模式集是根據規則表達式搜尋的日誌模式集合，以及偵測到模式時的低、中或高嚴重性。對於指標，您可以從服務特定和支援的指標清單中，選擇要監控每個元件的指標。對於警示，CloudWatch Application Insights 會自動為受監控的指標建立和設定標準或異常偵測警示。CloudWatch Application Insights 具有指標的自動組態，以及 CloudWatch 文件中 CloudWatch Application Insights 支援的日誌和指標中所概述技術的日誌擷取。下圖顯示 CloudWatch Application Insights 元件與其記錄和監控組態之間的關係。每個元件都定義了自己的日誌和指標，以使用 CloudWatch 日誌和指標進行監控。



CloudWatch Application Insights 監控的 EC2 執行個體需要 Systems Manager 和 CloudWatch 代理程式和許可。如需詳細資訊，請參閱 CloudWatch 文件中的使用 CloudWatch Application Insights 設定應用程式的先決條件。 CloudWatch CloudWatch Application Insights 使用 Systems Manager 來安裝和更新 CloudWatch 代理程式。CloudWatch Application Insights 中設定的指標和日誌會建立儲存在 Systems Manager 參數中的 CloudWatch 代理程式組態檔案，其中包含每個 CloudWatch Application Insights 元件的`AmazonCloudWatch-ApplicationInsights-SSMParameter`字首。這會導致另一個 CloudWatch 代理程式組態檔案新增至 EC2 執行個體上的 CloudWatch 代理程式

組態目錄。Systems Manager 命令會執行，將此組態附加至 EC2 執行個體上的作用中組態。使用 CloudWatch Application Insights 不會影響現有的 CloudWatch 代理程式組態設定。除了您自己的系統和應用程式層級 CloudWatch 代理程式組態之外，您還可以使用 CloudWatch Application Insights。不過，您應該確保組態不會重疊。

使用 CloudWatch Logs Insights 執行日誌分析

CloudWatch Logs Insights 可讓您使用簡單的查詢語言，輕鬆搜尋多個日誌群組。如果您的應用程式日誌以 JSON 格式建構，CloudWatch Logs Insights 會自動探索多個日誌群組中日誌串流的 JSON 欄位。您可以使用 CloudWatch Logs Insights 來分析您的應用程式和系統日誌，這會儲存您的查詢以供日後使用。CloudWatch Logs Insights 的查詢語法支援彙總函數，例如 sum()、avg()、count()、min() 和 max()，這些函數有助於疑難排解應用程式或效能分析。

如果您使用內嵌指標格式來建立 CloudWatch 指標，則可以使用支援的彙總函數查詢內嵌指標格式日誌來產生一次性指標。這有助於降低 CloudWatch 監控成本，方法是擷取必要的資料點，以視需要產生特定指標，而不是主動將其擷取為自訂指標。這對於高基數的維度特別有效，這會導致大量的指標。CloudWatch Container Insights 也會採用此方法，並擷取詳細的效能資料，但只會產生此資料子集的 CloudWatch 指標。

例如，下列內嵌指標項目只會從內嵌指標格式陳述式中擷取的指標資料產生一組有限的 CloudWatch 指標：

```
{
 "AutoScalingGroupName": "eks-e0bab7f4-fa6c-64ba-dbd9-094aee6cf9ba",
 "CloudWatchMetrics": [
 {
 "Metrics": [
 {
 "Unit": "Count",
 "Name": "pod_number_of_container_restarts"
 }
 ],
 "Dimensions": [
 [
 "PodName",
 "Namespace",
 "ClusterName"
 ]
 ],
 "Namespace": "ContainerInsights"
 }
```

```
],
"ClusterName": "eksdemo",
"InstanceId": "i-03b21a16b854aa4ca",
"InstanceType": "t3.medium",
"Namespace": "amazon-cloudwatch",
"NodeName": "ip-172-31-10-211.ec2.internal",
"PodName": "cloudwatch-agent",
"Sources": [
"cadvisor",
"pod",
"calculated"
],
"Timestamp": "1605111338968",
"Type": "Pod",
"Version": "0",
"pod_cpu_limit": 200,
"pod_cpu_request": 200,
"pod_cpu_reserved_capacity": 10,
"pod_cpu_usage_system": 3.268605094109382,
"pod_cpu_usage_total": 8.899539221131045,
"pod_cpu_usage_user": 4.160042847048305,
"pod_cpu_utilization": 0.44497696105655227,
"pod_cpu_utilization_over_pod_limit": 4.4497696105655224,
"pod_memory_cache": 4096,
"pod_memory_failcnt": 0,
"pod_memory_hierarchical_pgfault": 0,
"pod_memory_hierarchical_pgmajfault": 0,
"pod_memory_limit": 209715200,
"pod_memory_mapped_file": 0,
"pod_memory_max_usage": 43024384,
"pod_memory_pgfault": 0,
"pod_memory_pgmajfault": 0,
"pod_memory_request": 209715200,
"pod_memory_reserved_capacity": 5.148439982463127,
"pod_memory_rss": 38481920,
"pod_memory_swap": 0,
"pod_memory_usage": 42803200,
"pod_memory_utilization": 0.6172094650851303,
"pod_memory_utilization_over_pod_limit": 11.98828125,
"pod_memory_working_set": 25141248,
"pod_network_rx_bytes": 3566.4174629544723,
"pod_network_rx_dropped": 0,
"pod_network_rx_errors": 0,
"pod_network_rx_packets": 3.3495665260575094,
```

```
 "pod_network_total_bytes": 4283.442421354973,
 "pod_network_tx_bytes": 717.0249584005006,
 "pod_network_tx_dropped": 0,
 "pod_network_tx_errors": 0,
 "pod_network_tx_packets": 2.6964010534762948,
 "pod_number_of_container_restarts": 0,
 "pod_number_of_containers": 1,
 "pod_number_of_running_containers": 1,
 "pod_status": "Running"
}
```

不過，您可以查詢擷取的指標，以取得進一步的洞見。例如，您可以執行下列查詢，以查看具有記憶體頁面故障的最新 20 個 Pod：

```
fields @timestamp, @message
| filter (pod_memory_pgfault > 0)
| sort @timestamp desc
| limit 20
```

使用 Amazon OpenSearch Service 執行日誌分析

CloudWatch 與 Amazon OpenSearch Service 整合，可讓您使用訂閱篩選條件，將日誌資料從 CloudWatch 日誌群組串流至您選擇的 Amazon OpenSearch Service 叢集。您可以使用 CloudWatch 進行主要日誌和指標擷取和分析，然後針對下列使用案例使用 Amazon OpenSearch Service 進行擴增：

- 精細資料存取控制 – Amazon OpenSearch Service 可讓您將資料存取限制為欄位層級，並協助根據使用者許可匿名化欄位中的資料。如果您想要在不公開敏感資料的情況下支援故障診斷，這會很有用。
- 跨多個帳戶、區域和基礎設施彙總和搜尋日誌 – 您可以將日誌從多個帳戶和區域串流到常見的 Amazon OpenSearch Service 叢集。您的集中式營運團隊可以分析趨勢、問題，並跨帳戶和區域執行分析。將 CloudWatch 日誌串流到 Amazon OpenSearch Service 也可協助您在中央位置搜尋和分析多區域應用程式。
- 使用 ElasticSearch 代理程式將日誌直接運送並擴充至 Amazon OpenSearch Service ElasticSearch – 您的應用程式和技術堆疊元件可以使用 CloudWatch 代理程式不支援的OSs。您可能也想要先擴充和轉換日誌資料，再將其運送到您的日誌解決方案。Amazon OpenSearch Service 支援標準 Elasticsearch 用戶端，例如支援日誌擴充和轉換的 Elastic Beats 系列資料寄件人和 Logstash，再將日誌資料傳送至 Amazon OpenSearch Service。

- 現有的操作管理解決方案使用 ElasticSearch、Logstash、Kibana (ELK) Stack 進行記錄和監控 – 您可能已經對 Amazon OpenSearch Service 或開放原始碼 Elasticsearch 進行了大量投資，其中已設定許多工作負載。您可能也有已在 Kibana 中建立的操作儀表板，而您想要繼續使用。

如果您不打算使用 CloudWatch 日誌，您可以使用 Amazon OpenSearch Service 支援的代理程式、日誌驅動程式和程式庫（例如 Fluent Bit、Fluentd、logstash 和 Open Distro for ElasticSearch API) 將您的日誌直接運送到 Amazon OpenSearch Service 並略過 CloudWatch。不過，您也應該實作解決方案來擷取 AWS 服務產生的日誌。CloudWatch Logs 是許多 AWS 服務的主要日誌擷取解決方案，而多個 服務會自動在 CloudWatch 中建立新的日誌群組。例如，Lambda 會為每個 Lambda 函數建立新的日誌群組。您可以設定日誌群組的訂閱篩選條件，將其日誌串流至 Amazon OpenSearch Service。您可以為要串流到 Amazon OpenSearch Service 的每個個別日誌群組手動設定訂閱篩選條件。或者，您可以部署自動將新日誌群組訂閱到 ElasticSearch 叢集的解決方案。您可以將日誌串流到相同帳戶或集中帳戶中的 ElasticSearch 叢集。將日誌串流到相同帳戶中的 ElasticSearch 叢集，可協助工作負載擁有者更妥善地分析和支援其工作負載。

您應該考慮在集中或共用帳戶中設定 ElasticSearch 叢集，以彙總帳戶、區域和應用程式的日誌。例如， AWS Control Tower 設定用於集中式記錄的 Log Archive 帳戶。在 中建立新帳戶時 AWS Control Tower，其 AWS CloudTrail 和 AWS Config 日誌會交付到此集中式帳戶中的 S3 儲存貯體。檢測的記錄 AWS Control Tower 是用於組態、變更和稽核記錄。

若要使用 Amazon OpenSearch Service 建立集中式應用程式日誌分析解決方案，您可以將一或多個集中式 Amazon OpenSearch Service 叢集部署至集中式日誌帳戶，並在其他帳戶中設定日誌群組，將日誌串流至集中式 Amazon OpenSearch Service 叢集。

您可以建立個別的 Amazon OpenSearch Service 叢集，來處理可能分散到您帳戶的不同應用程式或雲端架構層。使用個別的 Amazon OpenSearch Service 叢集可協助您降低安全性和可用性風險，而擁有常見的 Amazon OpenSearch Service 叢集可讓您更輕鬆地搜尋和關聯相同叢集中的資料。

使用 CloudWatch 的警示選項

執行重要指標的一次性和自動化分析，可協助您在問題影響工作負載之前偵測並解決問題。CloudWatch 可讓您在特定期間內使用多個統計資料，輕鬆繪製和比較多個指標。您可以使用 CloudWatch 搜尋具有所需維度值的所有指標，以尋找分析所需的指標。

我們建議您透過包含一組初始指標和維度來開始指標擷取方法，以做為監控工作負載的基準。隨著時間的推移，工作負載會成熟，您可以新增其他指標和維度，以協助您進一步分析和支援它。您的應用程式或工作負載可能會使用多個 AWS 資源，並擁有自己的自訂指標，您應該將這些資源分組在命名空間下，讓它們更容易識別。

您也應該考慮記錄和監控資料的關聯性，以便快速識別相關的記錄和監控資料，以診斷特定問題。您可以使用AWS X-Ray 追蹤映射來關聯追蹤、指標、日誌和警示，以診斷問題。您也應考慮在工作負載的日誌中包含指標和識別符的其他維度，以協助您快速搜尋和識別跨系統和服務的問題。

使用 CloudWatch 警示來監控和警示

您可以使用 CloudWatch 警示來減少工作負載或應用程式中的手動監控。您應該先檢閱每個工作負載元件擷取的指標，並判斷每個指標的適當閾值。請確定您識別在違反閾值時必須通知哪些團隊成員。您應該建立和鎖定分佈群組，而不是個別團隊成員。

CloudWatch 警示可以與您的服務管理解決方案整合，以自動建立新的票證並執行操作工作流程。例如， AWS 提供 AWS Service Management Connector for ServiceNowAWS Service Management Connector，並協助您快速設定整合。此方法對於確保已認可引發的警示，並與這些產品中可能已定義的現有操作工作流程保持一致至關重要。

您也可以為具有不同閾值和評估期間的相同指標建立多個警示，這有助於建立呈報程序。例如，如果您有一個追蹤客戶訂單的`OrderQueueDepth`指標，您可以在短一分鐘的平均期間內定義較低的閾值，透過電子郵件或 Slack 通知應用程式團隊成員。您也可以在較長的 15 分鐘內，以相同的閾值定義相同指標的另一個警示，並通知應用程式團隊和應用程式團隊的主管。最後，您可以為 30 分鐘期間的硬平均閾值定義第三個警示，以通知上級管理並通知所有團隊成員。建立多個警示可協助您針對不同的條件採取不同的動作。您可以從簡單的通知程序開始，然後視需要調整和改善。

使用 CloudWatch 異常偵測來監控和警示

如果您不確定要套用至特定指標的閾值，或者您希望警示根據觀察到的歷史值自動調整閾值，則可以使用 CloudWatch 異常偵測。CloudWatch 異常偵測對於活動可能有定期、可預測變更的指標特別有用，

例如，在截止時間之前，當日交付的每日採購訂單會增加。異常偵測可啟用自動調整的閾值，並有助於減少錯誤警示。您可以為每個指標和統計資料啟用異常偵測，並將 CloudWatch 設定為根據極端值發出警示。

例如，您可以在 EC2 `CPUUtilization` 執行個體上啟用 指標和 `AVG`統計資料的異常偵測。然後，異常偵測會使用最多 14 天的歷史資料來建立機器學習 (ML) 模型。您可以使用不同的異常偵測頻帶建立多個警示，以建立警示提升程序，類似於使用不同的閾值建立多個標準警示。

如需本節的詳細資訊，請參閱 CloudWatch 文件中的根據異常偵測建立 CloudWatch 警示。
CloudWatch

跨多個區域和帳戶的警示

應用程式和工作負載擁有者應為跨越多個區域的工作負載建立應用程式層級警示。我們建議您在工作負載部署所在的每個帳戶和區域中建立個別警示。您可以使用帳戶和區域無關 CloudFormation StackSets 和 範本來簡化和自動化此程序，以部署具有必要警示的應用程式資源。 templateYou設定警示動作以鎖定常見的 Amazon Simple Notification Service (Amazon SNS) 主題，這表示無論帳戶或區域為何，都會使用相同的通知或修補動作。

在多帳戶和多區域環境中，我們建議您為帳戶和區域建立彙總警示，以使用 CloudFormation StackSets 和彙總指標來監控帳戶和區域問題，例如所有 EC2 執行個體`CPUUtilization`的平均值。

您也應考慮為針對您擷取的標準 CloudWatch 指標和日誌設定的每個工作負載建立標準警示。例如，您可以為每個監控 CPU 使用率指標的 EC2 執行個體建立個別警示，並在每天平均 CPU 使用率超過 80% 時通知中央操作團隊。您也可以建立標準警示，每天監控低於 10% 的平均 CPU 使用率。這些警示可協助中央操作團隊與特定工作負載擁有者合作，在需要時變更 EC2 執行個體的大小。

使用 EC2 執行個體標籤自動建立警示

為您的 EC2 執行個體建立一組標準警示可能會耗時、不一致且容易出錯。您可以使用 amazon-cloudwatch-auto-alarms 解決方案，為您的 EC2 執行個體自動建立一組標準 CloudWatch 警示，並根據 EC2 執行個體標籤建立自訂警示，以加速警示建立程序。解決方案消除了手動建立標準警示的需求，在大規模遷移使用 CloudEndure 等工具的 EC2 執行個體期間非常有用。您也可以使用 CloudFormation StackSets 部署此解決方案，以支援多個區域和帳戶。如需詳細資訊，請參閱部落格上的 AWS 使用標籤來建立和維護 Amazon EC2 執行個體的 Amazon CloudWatch 警示Amazon EC2。

監控應用程式和服務可用性

CloudWatch 可協助您監控和分析應用程式和工作負載的效能和執行時間層面。您也應該監控應用程式和工作負載的可用性和可連線性方面。您可以搭配 Amazon Route 53 運作狀態檢查和 CloudWatch Synthetics 使用主動監控方法來達成此目的。

當您想要監控透過 HTTP 或 HTTPS 與網頁的連線，或透過 TCP 與公有網域名稱系統 (DNS) 名稱或 IP 地址的網路連線時，您可以使用 Route 53 運作狀態檢查。Route 53 運作狀態檢查會以十秒或 30 秒的間隔從您指定的區域啟動連線。您可以選擇多個區域來執行運作狀態檢查，每個運作狀態檢查都會獨立執行，而且您必須至少選擇三個區域。如果特定子字串出現在傳回用於運作狀態檢查評估的前 5，120 個位元組的資料中，您可以搜尋 HTTP 或 HTTPS 請求的回應內文。如果傳回 2xx 或 3xx 回應，HTTP 或 HTTPS 請求會被視為正常運作。Route 53 運作狀態檢查可透過檢查其他運作狀態檢查來建立複合運作狀態檢查。如果您有多個服務端點，而且想要在其中一個服務端點運作狀態不佳時執行相同的通知，則可以執行此操作。如果您將 Route 53 用於 DNS，則可以將 Route 53 設定為如果運作狀態檢查變成運作狀態不佳，則容錯移轉至另一個 DNS 項目。對於每個關鍵工作負載，您應該考慮為對正常操作至關重要的外部端點設定 Route 53 運作狀態檢查。Route 53 運作狀態檢查可協助您避免將容錯移轉邏輯寫入應用程式。

CloudWatch 合成可讓您將 Canary 定義為指令碼，以評估工作負載的運作狀態和可用性。Canary 是以 Node.js 或 Python 撰寫的指令碼，並透過 HTTP 或 HTTPS 通訊協定運作。Canary 會使用 Node.js 或 Python 作為架構，在您的帳戶中建立 Lambda 函數。您定義的每個 Canary 可以對不同的端點執行多個 HTTP 或 HTTPS 呼叫。這表示您可以監控一系列步驟的運作狀態，例如使用案例或具有下游相依性的端點。Canary 會建立 CloudWatch 指標，其中包含執行的每個步驟，以便您可以獨立警示和測量不同的步驟。雖然 Canary 需要比 Route 53 運作狀態檢查更多的規劃和開發工作，但它們為您提供高度可自訂的監控和評估方法。Canary 也支援在虛擬私有雲端 (VPC) 中執行的私有資源，這使得它們非常適合在端點沒有公有 IP 地址時監控可用性。您也可以使用 Canary 來監控內部部署工作負載，只要您有從 VPC 到端點的連線。當您的工作負載包含內部部署中存在的端點時，這尤其重要。

使用 追蹤應用程式 AWS X-Ray

透過應用程式的請求可能包含對內部部署伺服器、Amazon EC2、容器或 Lambda 中執行的資料庫、應用程式和 Web 服務的呼叫。透過實作應用程式追蹤，您可以快速識別應用程式中使用分散式元件和服務之問題的根本原因。您可以使用 AWS X-Ray 跨多個元件追蹤應用程式請求。當請求流經您的應用程式元件，且每個元件都以區段表示時，X-Ray 會在服務圖表上取樣並視覺化請求。X-Ray 會產生追蹤識別符，以便您可以在請求流過多個元件時將其關聯，從而協助您從頭到尾檢視請求。您可以透過包含註釋和中繼資料來進一步增強此功能，以協助唯一地搜尋和識別請求的特性。

我們建議您使用 X-Ray 設定和檢測應用程式中的每個伺服器或端點。透過呼叫 X-Ray 服務在您的應用程式程式碼中實作 X-Ray。X-Ray 還提供多種語言 AWS SDKs，包括自動將資料傳送至 X-Ray 的經檢測用戶端。X-Ray SDK 為用於呼叫其他服務 (例如 HTTP、MySQL、PostgreSQL 或 MongoDB) 的常見程式庫提供修補程式。

X-Ray 提供 X-Ray 協助程式，您可以在 Amazon EC2 和 Amazon ECS 上安裝和執行，將資料轉送至 X-Ray。X-Ray 會為您的應用程式建立追蹤，從執行服務請求的 X-Ray 協助程式的伺服器和容器擷取效能資料。X-Ray 會自動透過修補 AWS SDK，將您對 Amazon DynamoDB 等 AWS 服務的呼叫作為子區段來測試。X-Ray 也可以自動與 Lambda 函數整合。

如果您的應用程式元件對無法設定和安裝 X-Ray 協助程式或檢測程式碼的外部服務發出呼叫，您可以建立子區段來包裝對外部服務的呼叫。如果您使用 ，X-Ray 會將 CloudWatch 日誌和指標與您的應用程式追蹤建立關聯 適用於 JAVA 的 AWS X-Ray SDK，這表示您可以快速分析請求的相關指標和日誌。

部署 X-Ray 協助程式以追蹤 Amazon EC2 上的應用程式和服務

您需要在應用程式元件或微服務執行所在的 EC2 執行個體上安裝和執行 X-Ray 協助程式。您可以在佈建 EC2 執行個體時使用使用者資料指令碼來部署 X-Ray 協助程式，或者如果您建立自己的 AMIs，則可以將其包含在 AMI 建置程序中。這在 EC2 執行個體為暫時性時特別有用。

您應該使用 State Manager 來確保 X-Ray 協助程式一致地安裝在 EC2 執行個體上。對於 Amazon EC2 Windows 執行個體，您可以使用 Systems Manager AWS-RunPowerShellScript 文件來執行下載和安裝 X-Ray 代理程式的 Windows 指令碼。對於 Linux 上的 EC2 執行個體，您可以使用 AWS-RunShellScript 文件來執行 Linux 指令碼，以下載和安裝代理程式做為服務。

您可以使用 Systems Manager AWS-RunRemoteScript 文件在多帳戶環境中執行指令碼。您必須建立可從所有帳戶存取的 S3 儲存貯體，如果您使用 ，建議您使用組織型儲存貯體政策來建立 S3 儲存貯體

AWS Organizations。然後，您將指令碼上傳至 S3 儲存貯體，但請確定 EC2 執行個體的 IAM 角色具有存取儲存貯體和指令碼的許可。

您也可以設定 State Manager，將指令碼與已安裝 X-Ray 代理程式的 EC2 執行個體建立關聯。由於所有 EC2 執行個體可能不需要或使用 X-Ray，因此您可以鎖定與執行個體標籤的關聯。例如，您可以根據 `InstallAWSXRayDaemonWindows`或 `InstallAWSXRayDaemonLinux`標籤的存在來建立狀態管理員關聯。

部署 X-Ray 協助程式以追蹤 Amazon ECS 或 Amazon EKS 上的應用程式和服務

您可以將 X-Ray 協助程式部署為容器型工作負載的附屬容器，例如 Amazon ECS 或 Amazon EKS。然後，如果您使用 Amazon ECS，您的應用程式容器可以使用容器連結連接到附屬容器，或者如果您使用 awsvpc 網路模式，則容器可以直接連接到 localhost 上的附屬容器。

對於 Amazon EKS，您可以在應用程式的 Pod 定義中定義 X-Ray 協助程式，然後您的應用程式可以透過您指定的容器連接埠上的 localhost 連線到協助程式。

設定 Lambda 追蹤對 X-Ray 的請求

您的應用程式可能包含對 Lambda 函數的呼叫。您不需要安裝適用於 Lambda 的 X-Ray 協助程式，因為協助程式程序完全由 Lambda 管理，使用者無法設定。您可以使用 AWS 管理主控台 並檢查 X-Ray 主控台中的主動追蹤選項，為 Lambda 函數啟用此功能。

如需進一步檢測，您可以將 X-Ray 開發套件與 Lambda 函數綁定，以記錄外撥通話並新增註釋或中繼資料。

檢測您的 X-Ray 應用程式

您應該評估符合應用程式程式設計語言的 X-Ray 開發套件，並將應用程式對其他系統進行的所有呼叫分類。檢閱您選擇的程式庫提供的用戶端，並查看開發套件是否可以自動檢測應用程式的請求或回應的追蹤。判斷軟體開發套件提供的用戶端是否可用於其他下游系統。對於應用程式呼叫且無法使用 X-Ray 檢測的外部系統，您應該建立自訂子區段，以在追蹤資訊中擷取和識別它們。

當您檢測應用程式時，請確定您建立註釋，以協助您識別和搜尋請求。例如，您的應用程式可能會為客戶使用識別符，例如 `customer id`，或根據其在應用程式中的角色來分割不同的使用者。

您可以為每個追蹤建立最多 50 個註釋，但您可以建立包含一或多個欄位的中繼資料物件，只要區段文件不超過 64 KB。您應該選擇性地使用註釋來尋找資訊，並使用中繼資料物件來提供更多內容，以協助在請求找到之後進行故障診斷。

設定 X-Ray 取樣規則

透過自訂抽樣規則，您可以控制記錄的資料量，並修改抽樣行為，而無需修改或重新部署程式碼。取樣規則會告知 X-Ray SDK 要針對一組條件記錄多少請求。根據預設，X-Ray 開發套件會記錄每秒的第一個請求，以及任何額外請求的百分之五。每秒一個請求是儲槽。這可確保只要服務持續提供請求，每秒都會記錄至少一個追蹤。5% 是超出儲槽大小的額外請求取樣率。

您應該檢閱並更新預設組態，以判斷您帳戶的適當值。您的需求在開發、測試、效能測試和生產環境中可能會有所不同。您可能有應用程式，根據其接收的流量或其重要性層級，需要自己的抽樣規則。您應該從基準開始，並定期重新評估基準是否符合您的需求。

使用 CloudWatch 的儀表板和視覺化

儀表板可協助您快速專注於應用程式和工作負載的關注領域。CloudWatch 提供自動儀表板，您也可以輕鬆建立使用 CloudWatch 指標的儀表板。CloudWatch 儀表板提供比單獨檢視指標更多的洞見，因為它們可協助您關聯多個指標並識別趨勢。例如，包含收到的訂單、記憶體、CPU 使用率和資料庫連線的儀表板，可協助您在訂單計數增加或減少時，將多個 AWS 資源的工作負載指標變更相互關聯。

您應該在帳戶和應用程式層級建立儀表板，以監控工作負載和應用程式。您可以開始使用 CloudWatch 自動儀表板，這是 AWS 預先設定服務特定指標的服務層級儀表板。自動服務儀表板會顯示服務的所有標準 CloudWatch 指標。自動儀表板會繪製每個服務指標使用的所有資源，並協助您快速識別帳戶中的極端值資源。這可協助您識別使用率高和低的資源，這可協助您最佳化成本。

建立跨服務儀表板

您可以透過檢視服務的自動服務層級儀表板，並使用動作功能表中的新增至儀表板選項來建立跨 AWS 服務儀表板。 然後，您可以從其他自動儀表板將指標新增至新的儀表板，並移除指標以縮小儀表板的焦點。您也應該新增自己的自訂指標，以追蹤金鑰觀察（例如，收到的訂單或每秒交易數）。建立您自己的自訂跨服務儀表板，可協助您專注於工作負載最相關的指標。我們建議您建立帳戶層級的跨服務儀表板，涵蓋關鍵指標並顯示帳戶中的所有工作負載。

如果您的雲端營運團隊有中央辦公室空間或共用區域，您可以在全螢幕模式下在大型電視監視器上顯示 CloudWatch 儀表板，並自動重新整理。

建立應用程式或工作負載特定的儀表板

我們建議您建立應用程式和工作負載特定的儀表板，專注於生產環境中每個關鍵應用程式或工作負載的關鍵指標和資源。應用程式和工作負載特定的儀表板著重於您的自訂應用程式或工作負載指標，以及會影響其效能的重要 AWS 資源指標。

您應該定期評估和自訂 CloudWatch 應用程式或工作負載儀表板，以便在事件發生後追蹤關鍵指標。引進或淘汰功能時，您也應該更新應用程式或工作負載特定的儀表板。除了記錄和監控之外，更新工作負載和應用程式特定的儀表板應該是持續改善品質的必要活動。

建立跨帳戶或跨區域儀表板

AWS 資源主要是區域性，指標、警示和儀表板專屬於資源部署所在的區域。這可能需要您變更區域，以檢視跨區域工作負載和應用程式的指標、儀表板和警示。如果您將應用程式和工作負載分隔為多個帳

戶，您可能還需要重新驗證並登入每個帳戶。不過，CloudWatch 支援從單一帳戶檢視跨帳戶和跨區域資料，這表示您可以在單一帳戶和區域中檢視指標、警示、儀表板和日誌小工具。如果您有集中式記錄和監控帳戶，這非常有用。

帳戶擁有者和應用程式團隊擁有者應為帳戶特定的跨區域應用程式建立儀表板，以有效地監控集中位置中的關鍵指標。CloudWatch 儀表板會自動支援跨區域小工具，這表示您可以建立儀表板，其中包含來自多個區域的指標，而無需進一步設定。

一個重要的例外是 CloudWatch Logs Insights 小工具，因為日誌資料只能針對您目前登入的帳戶和區域顯示。您可以使用指標篩選條件從日誌建立區域特定的指標，這些指標可以顯示在跨區域儀表板上。當您需要進一步分析這些日誌時，您可以切換到特定區域。

營運團隊應建立集中式儀表板，以監控重要的跨帳戶和跨區域指標。例如，您可以建立跨帳戶儀表板，其中包含每個帳戶和區域中的彙總 CPU 使用率。您也可以使用指標數學，跨多個帳戶和區域彙總和儀表板資料。

使用指標數學微調可觀測性和警示

您可以使用指標數學來協助計算與工作負載相關的格式和表達式指標。計算的指標可以在儀表板上儲存和檢視，以供追蹤之用。例如，標準 Amazon EBS 磁碟區指標提供在特定期間內執行的讀取 (`VolumeReadOps`) 和寫入 (`VolumeWriteOps`) 操作數目。

不過， AWS 提供 IOPS 中 Amazon EBS 磁碟區效能的指導方針。您可以新增 和 ，`VolumeWriteOps`然後除以為這些指標選擇的期間，以指標數學來繪製`VolumeReadOps`和計算 Amazon EBS 磁碟區的 IOPS。

在此範例中，我們加總該期間的 IOPS，然後除以期間長度以取得 IOPS。然後，您可以針對此指標數學表達式設定警示，以便在磁碟區的 IOPS 接近磁碟區類型的最大容量時提醒您。如需使用指標數學來監控具有 CloudWatch 指標的 Amazon Elastic File System (Amazon EFS) 檔案系統的詳細資訊和範例，請參閱 AWS 部落格上的 Amazon CloudWatch 指標數學可簡化 Amazon EFS 檔案系統的近乎即時監控。

將 Amazon ECS、Amazon EKS 和 Lambda 的自動儀表板與 CloudWatchContainer Insights 和 CloudWatch Lambda Insights 搭配使用

CloudWatch Container Insights 會為在 Amazon ECS 和 Amazon EKS 上執行的容器工作負載建立動態的自動儀表板。您應該讓 Container Insights 能夠觀察 CPU、記憶體、磁碟、網路和診斷資訊，例

如容器重新啟動失敗。Container Insights 會產生動態儀表板，您可以快速篩選叢集、容器執行個體或節點、服務、任務、Pod 和個別容器層級。Container Insights 是在叢集和節點或容器執行個體層級設定，視服務 AWS 而定。

與 Container Insights 類似，CloudWatch Lambda Insights 會為您的 Lambda 函數建立動態的自動儀表板。此解決方案會收集、彙總和摘要系統層級指標，包括 CPU 時間、記憶體、磁碟和網路。它也會收集、彙總和摘要診斷資訊，例如冷啟動和 Lambda 工作者關閉，以協助您隔離和快速解決 Lambda 函數的問題。Lambda 在函數層級啟用，不需要任何代理程式。

Container Insights 和 Lambda Insights 也可協助您快速切換到應用程式或效能日誌、X-Ray 追蹤和服務地圖，以視覺化您的容器工作負載。它們都使用 CloudWatch 內嵌指標格式來擷取 CloudWatch 指標和效能日誌。

您可以為工作負載建立共用 CloudWatch 儀表板，使用 Container Insights 和 Lambda Insights 擷取的指標。您可以透過 CloudWatch Container Insights 篩選和檢視自動儀表板，然後選擇新增至儀表板選項，以將顯示的指標新增至標準 CloudWatch 儀表板。然後，您可以移除或自訂指標，並新增其他指標以正確代表您的工作負載。

CloudWatch 與 AWS 服務的整合

AWS 提供許多 服務，其中包含記錄和指標的其他組態選項。這些服務通常可讓您為日誌輸出設定 CloudWatch Logs，並為指標輸出設定 CloudWatch 指標。用於提供這些服務的基礎基礎設施由 AWS 管理，且無法存取，但您可以使用佈建服務的記錄和指標選項，以取得進一步的洞見和故障診斷問題。例如，您可以將 VPC 流程日誌發佈至 CloudWatch，也可以設定 Amazon Relational Database Service (Amazon RDS) 執行個體將日誌發佈至 CloudWatch。

大多數 AWS 服務會記錄其與 整合的 API AWS CloudTrail呼叫。CloudTrail 也支援與 CloudWatch Logs 整合，這表示您可以在 AWS 服務中搜尋和分析活動。您也可以使用 或 Amazon EventBridge，針對在 AWS 服務中執行的特定動作，使用事件規則來建立和設定自動化和通知。某些 服務會直接與 EventBridge 整合。 EventBridge 您也可以建立透過 CloudTrail 交付的事件。

用於儀表板和視覺化的 Amazon Managed Grafana

Amazon Managed Grafana 可用來觀察和視覺化 AWS 工作負載。Amazon Managed Grafana 可協助您大規模視覺化和分析營運資料。Grafana 是一種開放原始碼分析平台，可協助您查詢、視覺化、提醒和了解存放指標的任何地方。如果您的組織已使用 Grafana 視覺化現有工作負載，而且您想要將涵蓋範圍擴展到 AWS 工作負載，Amazon Managed Grafana 特別有用。您可以將 Amazon Managed Grafana 與 CloudWatch 搭配使用，方法是將其新增為資料來源，這表示您可以使用 CloudWatch 指標建立視覺化。Amazon Managed Grafana 支援 AWS Organizations ，您可以使用來自多個帳戶和區域的 CloudWatch 指標來集中儀表板。

下表提供使用 Amazon Managed Grafana 而非 CloudWatch 進行儀表板的優勢和考量。根據最終使用者、工作負載和應用程式的不同需求，混合式方法可能是合適的。

建立與 Amazon Managed Grafana 和開放原始碼 Grafana 支援的資料來源整合的視覺化和儀表板	Amazon Managed Grafana 可協助您從許多不同的資料來源建立視覺化和儀表板，包括 CloudWatch 指標。Amazon Managed Grafana 包含多個內建資料來源，涵蓋 AWS 服務、開放原始碼軟體和 COTS 軟體。如需詳細資訊，請參閱 Amazon Managed Grafana 文件中的內建資料來源。您也可以將工作區升級至 Grafana Enterprise，以新增對更多資料來源的支援。Grafana 也支援資料來源外掛程式，可讓您與不同的外部系統通訊。CloudWatch 儀表板需要 CloudWatch 指標或 CloudWatch Logs Insights 查詢，才能在 CloudWatch 儀表板上顯示資料。
與 AWS 您的帳戶存取分開管理對儀表板解決方案的存取	Amazon Managed Grafana 需要使用 AWS IAM Identity Center (IAM Identity Center) 和 AWS Organizations 進行身分驗證和授權。這可讓您使用可能已與 IAM Identity Center 或 搭配使用的聯合身分，向 Grafana 驗證使用者身分 AWS Organizations。不過，如果您不是使用 IAM Identity Center 或 AWS Organizations，則會將其設定為 Amazon Managed Grafana 設定程序的一部分。如果您的組織限制使用 IAM Identity

	Center 或 ，這可能會成為問題 AWS Organizations。
透過 AWS Organizations 整合跨多個帳戶和區域擷取和存取資料	Amazon Managed Grafana 與 整合 AWS Organizations ，可讓您在所有帳戶中從 CloudWatch 和 Amazon OpenSearch Service 等 AWS 來源讀取資料。這可讓您建立儀表板，使用跨帳戶的資料顯示視覺化效果。若要自動啟用跨 的資料存取 AWS Organizations，您需要在 AWS Organizations 管理帳戶中設定 Amazon Managed Grafana 工作區。根據AWS Organizations 管理帳戶的最佳實務，不建議這麼做。相反地，CloudWatch 也支援 CloudWatch 指標的跨帳戶、跨區域儀表板。
使用開放原始碼社群中可用的進階視覺化小工具和 Grafana 定義	Grafana 提供大量的視覺效果集合，您可以在建立儀表板時使用。還有一個大型的社群貢獻儀表板程式庫，您可以根據您的需求編輯和重複使用。
搭配新的和現有的 Grafana 部署使用儀表板	如果您已使用 Grafana，您可以從 Grafana 部署匯入和匯出儀表板，並自訂儀表板以在 Amazon Managed Grafana 中使用。Amazon Managed Grafana 可讓您在 Grafana 上標準化為儀表板解決方案。
工作區、許可和資料來源的進階設定和組態	Amazon Managed Grafana 可讓您建立多個 Grafana 工作區，這些工作區具有自己的設定資料來源、使用者和政策集。這可協助您滿足更進階的使用案例需求，以及進階安全組態。如果您的團隊尚未具備這些技能，則進階功能可能會要求他們增加與 Grafana 的體驗。

使用 CloudWatch 常見問答集設計和實作記錄和監控

本節提供有關使用 CloudWatch 設計和實作記錄和監控解決方案的常見問題解答。

要存放 CloudWatch 組態檔案的位置？

適用於 Amazon EC2 的 CloudWatch 代理程式可以套用儲存在 CloudWatch 組態目錄中的多個組態檔案。理想情況下，您應該將 CloudWatch 組態儲存為一組檔案，因為您可以在多個帳戶和環境中進行版本控制並再次使用它們。如需詳細資訊，請參閱本指南的 管理 CloudWatch 組態一節。或者，您可以將組態檔案存放在 GitHub 的儲存庫中，並在佈建新的 EC2 執行個體時自動擷取組態檔案。

發出警示時，如何在我的服務管理解決方案中建立票證？

您可以將服務管理系統與 Amazon Simple Notification Service (Amazon SNS) 主題整合，並設定 CloudWatch 警示以在發出警示時通知 SNS 主題。您的整合系統會收到 SNS 訊息，並且可以使用您的服務管理系統 APIs或 SDKs建立票證。

如何使用 CloudWatch 擷取容器中的日誌檔案？

Amazon ECS 任務和 Amazon EKS Pod 可以設定為自動將 STDOUT 和 STDERR 輸出傳送至 CloudWatch。記錄容器化應用程式的建議方法是讓容器將其輸出傳送到 STDOUT 和 STDERR。十二要素應用程式資訊清單也涵蓋了這一點。

不過，如果您想要將特定日誌檔案傳送至 CloudWatch，您可以在 Amazon EKS Pod 或 Amazon ECS 任務定義中掛載磁碟區，您的應用程式將寫入其批次檔案，並使用 Fluentd 或 Fluent Bit 的附屬容器將日誌傳送至 CloudWatch。您應該考慮將容器中的特定日誌檔案以符號方式連結至 `/dev/stdout`和 `/dev/stderr`。如需詳細資訊，請參閱 Docker 文件中的檢視容器或服務日誌。

如何監控 AWS 服務的運作狀態問題？

您可以使用 AWS Health Dashboard來監控 AWS 運作狀態事件。您也可以參考 aws-health-tools GitHub 儲存庫，以取得與 AWS 運作狀態事件相關的範例自動化解決方案。

沒有客服人員支援時，如何建立自訂 CloudWatch 指標？

您可以使用內嵌指標格式，將指標擷取至 CloudWatch。您也可以使用 AWS SDK（例如 put_metric_data)、 AWS CLI（例如 put-metric-data) 或 AWS API（例如 PutMetricData) 來建立自訂

指標。您應該考慮如何長期維護任何自訂邏輯。其中一種方法是使用 Lambda 搭配整合式內嵌指標格式支援來建立指標，以及 CloudWatch Events 事件排程規則來建立指標的期間。

如何將現有的記錄和監控工具與 整合 AWS？

您應該參考軟體或服務供應商提供的指南，以便與 整合 AWS。您可能可以使用代理程式軟體、軟體開發套件或提供的 API，將日誌和指標傳送到其解決方案。您也可以使用設定為廠商規格的開放原始碼解決方案，例如 Fluentd 或 Fluent Bit。您也可以使用 AWS SDK 和 CloudWatch Logs 訂閱篩選條件搭配 Lambda 和 Kinesis Data Streams 來建立自訂日誌處理器和寄件人。最後，如果您使用多個帳戶和區域，也應該考慮如何整合軟體。

Resources

簡介

- AWS Well-Architected

目標業務成果

- logging-monitoring-apg-guide-examples
- 雲端運算的六個優點

規劃您的 CloudWatch 部署

- AWS Organizations 術語與概念
- AWS Systems Manager 快速設定
- 使用 CloudWatch 代理程式從 Amazon EC2 執行個體和內部部署伺服器收集指標和日誌
- cloudwatch-config-s3-bucket.yaml
- 使用精靈建立 CloudWatch 代理程式組態檔案
- 企業 DevOps：為什麼您應該執行您建置的內容
- 將日誌資料匯出至 Amazon S3
- Amazon OpenSearch Service 中的精細存取控制
- Lambda 配額
- 手動建立或編輯 CloudWatch 代理程式組態檔案
- 使用訂閱即時處理日誌資料
- 要建置的工具 AWS

為 EC2 執行個體和內部部署伺服器設定 CloudWatch 代理程式

- Amazon EC2 指標維度
- 爆量效能執行個體

- CloudWatch 代理程式預先定義的指標集
- 使用 procstat 外掛程式收集程序指標
- 設定 procstat 的 CloudWatch 代理程式
- 管理 EC2 執行個體的詳細監控
- 使用 CloudWatch 內嵌指標格式擷取高基數日誌和產生指標
- 使用日誌群組和日誌串流
- 列出執行個體可用的 CloudWatch 指標
- PutLogEvents
- 使用收集的 擷取自訂指標
- 使用 StatsD 擷取自訂指標

Amazon EC2 和內部部署伺服器的 CloudWatch 代理程式安裝方法

- 在混合多雲端環境中建立 Systems Manager 所需的 IAM 服務角色
- 為混合環境建立受管執行個體啟用
- 建立 IAM 角色和使用者以搭配 CloudWatch 代理程式使用
- 使用命令列下載和設定 CloudWatch 代理程式
- 如何設定使用 Systems Manager 代理程式和統一 CloudWatch 代理程式的現場部署伺服器只使用臨時憑證？
- 堆疊集操作的先決條件
- 使用 Spot 執行個體

在 Amazon ECS 上記錄和監控

- amazon-cloudwatch-logs-for-fluent-bit
- Amazon ECS CloudWatch 指標
- Amazon ECS Container Insights 指標
- Amazon ECS 容器代理程式
- Amazon ECS 啟動類型
- 部署 CloudWatch 代理程式以在 Amazon ECS 上收集 EC2 執行個體層級指標

- ecs_cluster_with_cloudwatch_linux.yaml
- ecs_cw_emf_example
- ecs_firelense_emf_example
- ecs-task-nginx-firelense.json
- 擷取 Amazon ECS 最佳化 AMI 中繼資料
- 使用 awslogs 日誌驅動程式
- 使用用戶端程式庫產生內嵌指標格式日誌

在 Amazon EKS 上記錄和監控

- Amazon EKS 控制平面記錄
- amazon_eks_managed_node_group_launch_config.yaml
- Amazon EKS 節點
- amazon-eks-nodegroup.yaml
- Amazon EKS 服務水準協議
- Container Insights Prometheus 指標監控
- 使用 Prometheus 控制平面指標
- Fargate 記錄
- Fargate 上適用於 Amazon EKS 的 Fluent Bit
- 在 Fargate 上使用 Amazon EKS 時如何擷取應用程式日誌
- 安裝 CloudWatch 代理程式以收集 Prometheus 指標
- 安裝 Kubernetes 指標伺服器
- kubernetes/儀表板
- Kubernetes Horizontal Pod Autoscaler
- Kubernetes 控制平面元件
- Kubernetes Pod
- 啟動範本支援
- 受管節點群組
- 受管節點更新行為
- metrics-server

- 使用 Prometheus 和 Grafana 在 Fargate 上監控 Amazon EKS
- prometheus_jmx
- prometheus / jmx_exporter
- 擴展其他 Prometheus 來源並匯入這些指標
- 自我管理節點
- 將日誌傳送至 CloudWatch Logs
- 將 FluentD 設定為 DaemonSet，以將日誌傳送至 CloudWatch Logs
- 在 Amazon EKS 和 Kubernetes 上設定 Java/JMX 範例工作負載
- 新增 Prometheus 湊集目標的教學課程：Prometheus API Server 指標
- 垂直 Pod Autoscaler

的日誌記錄和指標 AWS Lambda

- Lambda 調用錯誤
- 記錄 – Python 的記錄設施
- 使用用戶端程式庫產生內嵌指標格式日誌
- 使用 Lambda 函數指標

搜尋和分析 CloudWatch 中的日誌

- Beats 系列
- Elastic Logstash
- 彈性堆疊
- 將 CloudWatch Logs 資料串流至 Amazon OpenSearch Service

使用 CloudWatch 的警示選項

- amazon-cloudwatch-auto-alarms
- AWS 適用於 Jira Service Management Cloud 的服務管理連接器
- AWS 適用於 Jira Service Management Data Center 的服務管理連接器
- AWS Service Management Connector for ServiceNow

監控應用程式和服務可用性

- 設定 DNS 容錯移轉

使用 追蹤應用程式 AWS X-Ray

- Amazon ECS 任務聯網
- 在 X-Ray 主控台中設定取樣規則
- 執行 Windows PowerShell 命令或指令碼
- 在 Amazon EC2 上執行 X-Ray 協助程式
- 將追蹤資料傳送至 X-Ray
- X-Ray 中的服務圖表

使用 CloudWatch 的儀表板和視覺化

- Amazon CloudWatch 指標數學可簡化 Amazon EFS 檔案系統的近乎即時監控
- 設定 CloudWatch Container Insights
- 使用指標數學

CloudWatch 與 AWS 服務的整合

- AWS CloudTrail 支援的 服務和整合
- Amazon EventBridge AWS 服務 中來自 的事件
- 透過 傳遞的 AWS 服務事件 AWS CloudTrail
- 使用 CloudWatch Logs 監控 CloudTrail 日誌檔案 CloudWatch
- 將資料庫日誌發佈至 CloudWatch Logs
- 將流程日誌發佈至 CloudWatch Logs

用於儀表板和視覺化的 Amazon Managed Grafana

- 中的管理帳戶的最佳實務 AWS Organizations
- Amazon Managed Grafana 的內建資料來源

- CloudWatch 中的跨帳戶和跨區域儀表板
- Grafana 外掛程式

文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 RSS 摘要。

變更	描述	日期
更新記錄資訊	更新了 記錄的 AWS Lambda 章節。	2023 年 4 月 17 日
已更新組態資訊	更新並重新命名有關建立和儲存 CloudWatch 組態的章節。	2023 年 2 月 9 日
更新指標資訊	已更新 Amazon ECS 指標區段中的自訂應用程式指標資訊。	2023 年 1 月 31 日
已移除預覽通知	Amazon Managed Grafana 已全面推出。	2022 年 5 月 25 日
已移除區段	不再支援 CloudWatch 軟體開發套件指標。	2022 年 1 月 7 日
初次出版	—	2021 年 4 月 30 日

AWS 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

數字

7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的現場部署 Oracle 資料庫遷移至 Amazon Aurora PostgreSQL 相容版本。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將內部部署 Oracle 資料庫遷移至 中的 Amazon Relational Database Service (Amazon RDS) for Oracle AWS 雲端。
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 (CRM) 系統遷移至 Salesforce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將您的現場部署 Oracle 資料庫遷移至 中 EC2 執行個體上的 Oracle AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移到相同平台的雲端服務。範例：將 Microsoft Hyper-V應用程式遷移至 AWS。
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

A

ABAC

請參閱屬性型存取控制。

抽象服務

請參閱 受管服務。

ACID

請參閱原子性、一致性、隔離性、耐久性。

主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但比主動-被動遷移需要更多的工作。

主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫會在資料複寫至目標資料庫時處理來自連線應用程式的交易。目標資料庫在遷移期間不接受任何交易。

彙總函數

在一組資料列上運作的 SQL 函數，會計算群組的單一傳回值。彙總函數的範例包括 SUM和 MAX。

AI

請參閱人工智慧。

AIOps

請參閱人工智慧操作。

匿名化

永久刪除資料集中個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

反模式

經常用於重複性問題的解決方案，其中解決方案具有反生產力、無效或比替代解決方案更有效。

應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體攻擊。

應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是產品組合探索和分析程序的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱什麼是人工智慧？

人工智慧操作 (AIOps)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需有關如何在 AWS 遷移策略中使用 AIOps 的詳細資訊，請參閱操作整合指南。

非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

原子性、一致性、隔離性、耐久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱《 AWS Identity and Access Management (IAM) 文件》中的 ABAC for AWS。

授權資料來源

存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、修訂或假名化資料。

可用區域

中的不同位置 AWS 區域 ，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、低延遲網路連線能力。

AWS 雲端採用架構 (AWS CAF)

的指導方針和最佳實務架構 AWS ，可協助組織制定高效且有效的計劃，以成功地移至雲端。 AWS CAF 將指導方針組織到六個重點領域：業務、人員、治理、平台、安全和營運。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。因此， AWS CAF 為人員開發、訓練和通訊提供指引，協助組織做好成功採用雲端的準備。如需詳細資訊，請參閱 AWS CAF 網站和 AWS CAF 白皮書。

AWS 工作負載資格架構 (AWS WQF)

評估資料庫遷移工作負載、建議遷移策略並提供工作預估值的工具。 AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

B

錯誤的機器人

旨在中斷或傷害個人或組織的機器人。

BCP

請參閱業務持續性規劃。

行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以將行為圖與 Amazon Detective 搭配使用來檢查失敗的登入嘗試、可疑的 API 呼叫和類似動作。如需詳細資訊，請參閱偵測文件中的行為圖中的資料。

大端序系統

首先儲存最高有效位元組的系統。另請參閱 Endianness。

二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境 （藍色） 中執行目前的應用程式版本，並在另一個環境 （綠色） 中執行新的應用程式版本。此策略可協助您快速復原，並將影響降至最低。

機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。有些機器人有用或有益，例如在網際網路上編製資訊索引的 Web 爬蟲程式。某些其他機器人稱為惡意機器人，旨在中斷或傷害個人或組織。

殭屍網路

受到惡意軟體感染且受單一方控制之機器人的網路，稱為機器人繼承器或機器人運算子。殭屍網路是擴展機器人及其影響的最佳已知機制。

分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱關於分支 (GitHub 文件)。

碎片存取

在特殊情況下，以及透過核准的程序，讓使用者快速取得他們通常無權存取 AWS 帳戶 之 的存取權。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS 實作打破玻璃程序指標。

棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

緩衝快取

儲存最常存取資料的記憶體區域。

業務能力

業務如何創造價值 (例如，銷售、客戶服務或營銷)。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 AWS上執行容器化微服務白皮書的圍繞業務能力進行組織部分。

業務連續性規劃 (BCP)

一種解決破壞性事件 (如大規模遷移) 對營運的潛在影響並使業務能夠快速恢復營運的計畫。

C

CAF

請參閱AWS 雲端採用架構。

Canary 部署

版本對最終使用者的緩慢和增量版本。當您有信心時，您可以部署新版本並完全取代目前的版本。

CCoE

請參閱 Cloud Center of Excellence。

CDC

請參閱變更資料擷取。

變更資料擷取 (CDC)

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以將 CDC 用於各種用途，例如稽核或複寫目標系統中的變更以保持同步。

混沌工程

故意引入故障或破壞性事件，以測試系統的彈性。您可以使用 AWS Fault Injection Service (AWS FIS) 執行實驗，為您的 AWS 工作負載帶來壓力，並評估其回應。

CI/CD

請參閱持續整合和持續交付。

分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

用戶端加密

在目標 AWS 服務 接收資料之前，在本機加密資料。

雲端卓越中心 (CCoE)

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端 企業策略部落格上的 CCoE 文章。

雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲端運算通常連接到邊緣運算技術。

雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱建置您的雲端操作模型。

採用雲端階段

組織在遷移至 時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展雲端採用 (例如，建立登陸區域、定義 CCoE、建立營運模型)

- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段由 Stephen Orban 在部落格文章 The Journey Toward Cloud-First 和企業策略部落格上的採用階段中定義。 AWS 雲端 如需有關它們如何與 AWS 遷移策略相關的詳細資訊，請參閱遷移整備指南。

CMDB

請參閱組態管理資料庫。

程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲端儲存庫包括 GitHub或 Bitbucket Cloud。程式碼的每個版本都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

冷資料

很少存取且通常是歷史資料的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且成本較低的儲存層或類別，可以降低成本。

電腦視覺 (CV)

AI 欄位???，使用機器學習從數位影像和影片等視覺化格式分析和擷取資訊。例如，Amazon SageMaker AI 提供 CV 的影像處理演算法。

組態偏離

對於工作負載，組態會從預期狀態變更。這可能會導致工作負載不合規，而且通常是漸進和無意的。

組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常在遷移的產品組合探索和分析階段使用 CMDB 中的資料。

一致性套件

您可以組合的 AWS Config 規則和修補動作集合，以自訂您的合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶 和 區域中或整個組織的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的一致性套件。

持續整合和持續交付 (CI/CD)

自動化軟體發行程序的來源、建置、測試、暫存和生產階段的程序。CI/CD 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。如需詳細資訊，請參閱持續交付的優點。CD 也可表示持續部署。如需詳細資訊，請參閱持續交付與持續部署。

CV

請參閱電腦視覺。

D

靜態資料

網路中靜止的資料，例如儲存中的資料。

資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected Framework 中安全支柱的元件。如需詳細資訊，請參閱資料分類。

資料偏離

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變更。資料偏離可以降低 ML 模型預測的整體品質、準確性和公平性。

傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

資料網格

架構架構，提供分散式、分散式資料擁有權與集中式管理。

資料最小化

僅收集和處理嚴格必要資料的原則。在 中實作資料最小化 AWS 雲端 可以降低隱私權風險、成本和分析碳足跡。

資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱在 上建置資料周邊 AWS。

資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

資料來源

在整個生命週期中追蹤資料的原始伺服器和歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

資料主體

正在收集和處理資料的個人。

資料倉儲

支援商業智慧的資料管理系統，例如 分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

資料庫處理語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

DDL

請參閱資料庫定義語言。

深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

深度防禦

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在 上採用此策略時 AWS，您可以在 AWS Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，defense-in-depth方法可能會結合多重要素驗證、網路分割和加密。

委派的管理員

在 中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶來管理組織的帳戶，並管理該服務的許可。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的可搭配 AWS Organizations運作的服務。

deployment

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

開發環境

請參閱 環境。

偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS上實作安全控制中的偵測性控制。

開發值串流映射 (DVSM)

一種程序，用於識別對軟體開發生命週期中的速度和品質造成負面影響的限制並排定優先順序。DVSM 擴展了原本專為精簡製造實務設計的價值串流映射程序。它著重於透過軟體開發程序建立和移動價值所需的步驟和團隊。

數位分身

真實世界系統的虛擬呈現，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

維度資料表

在星星結構描述中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常是文字欄位或離散數字，其行為類似於文字。這些屬性通常用於查詢限制、篩選和結果集標記。

災難

防止工作負載或系統在其主要部署位置中實現其業務目標的事件。這些事件可能是自然災難、技術故障或人類動作的結果，例如意外設定錯誤或惡意軟體攻擊。

災難復原 (DR)

您用來將災難造成的停機時間和資料遺失降至最低的策略和程序。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的 上的工作負載災難復原 AWS：雲端中的復原。

DML

請參閱資料庫處理語言。

領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何將領域驅動的設計與 strangler fig 模式搭配使用的資訊，請參閱使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET (ASMX) Web 服務。

DR

請參閱災難復原。

偏離偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源中的偏離，也可以使用 AWS Control Tower 來偵測登陸區域中可能影響控管要求合規性的變更。 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html

DVSM

請參閱開發值串流映射。

E

EDA

請參閱探索性資料分析。

EDI

請參閱電子資料交換。

邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與雲端運算相比，邊緣運算可以減少通訊延遲並改善回應時間。

電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱什麼是電子資料交換。

加密

將人類可讀取的純文字資料轉換為加密文字的運算程序。

加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

端點

請參閱 服務端點。

端點服務

您可以在虛擬私有雲端 (VPC) 中託管以與其他使用者共用的服務。您可以使用 建立端點服務，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management (IAM) 委託人。這些帳戶或主體可以透過建立介面 VPC 端點私下連接至您的端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud (Amazon VPC) 文件中的建立端點服務。

企業資源規劃 (ERP)

一種系統，可自動化和管理企業的關鍵業務流程 （例如會計、MES 和專案管理）。

信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service (AWS KMS) 文件中的信封加密。

環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如， AWS CAF 安全概念包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱計畫實作指南。

ERP

請參閱企業資源規劃。

探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。透過計算摘要統計並建立資料可視化來執行 EDA。

F

事實資料表

星狀結構描述中的中央資料表。它存放有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含度量的資料，以及包含維度資料表外部索引鍵的資料欄。

快速失敗

一種使用頻繁和增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

故障隔離界限

在 中 AWS 雲端，像是可用區域 AWS 區域、控制平面或資料平面等邊界會限制故障的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱AWS 故障隔離界限。

功能分支

請參閱分支。

特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

功能重要性

特徵對於模型的預測有多重要。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱 的機器學習模型可解譯性 AWS。

特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

少量擷取提示

在要求 LLM 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。少量的提示對於需要特定格式、推理或網域知識的任務來說非常有效。另請參閱零鏡頭提示。

FGAC

請參閱精細存取控制。

精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。

閃切遷移

一種資料庫遷移方法，透過變更資料擷取使用連續資料複寫，以盡可能在最短的時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

FM

請參閱基礎模型。

基礎模型 (FM)

大型深度學習神經網路，已在廣義和未標記資料的大量資料集上進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱什麼是基礎模型。

G

生成式 AI

已針對大量資料進行訓練的 AI 模型子集，可使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱什麼是生成式 AI。

地理封鎖

請參閱地理限制。

地理限制 (地理封鎖)

Amazon CloudFront 中的選項，可防止特定國家/地區的使用者存取內容分發。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的限制內容的地理分佈。

Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程被視為舊版，而以幹線為基礎的工作流程是現代、偏好的方法。

黃金影像

系統或軟體的快照，做為部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為棕地。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

防護機制

有助於跨組織單位 (OU) 來管控資源、政策和合規的高層級規則。預防性防護機制會強制執行政策，以確保符合合規標準。透過使用服務控制政策和 IAM 許可界限來將其實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。它們是透過使用 AWS Config AWS Security Hub CSPM、Amazon GuardDuty、、Amazon Inspector AWS Trusted Advisor和自訂 AWS Lambda 檢查來實作。

H

HA

請參閱高可用性。

異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如，Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分，而轉換結構描述可能是一項複雜任務。AWS 提供有助於結構描述轉換的 AWS SCT。

高可用性 (HA)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統的設計目的是自動容錯移轉、持續提供高品質的效能，並處理不同的負載和故障，並將效能影響降至最低。

歷史現代化

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種來源收集和存放資料。

保留資料

從用於訓練機器學習模型的資料集中保留的部分歷史標記資料。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

異質資料庫遷移

將您的來源資料庫遷移至共用相同資料庫引擎的目標資料庫 (例如，Microsoft SQL Server 至 Amazon RDS for SQL Server)。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

熱資料

經常存取的資料，例如即時資料或最近的轉譯資料。此資料通常需要高效能儲存層或類別，才能提供快速的查詢回應。

修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性，通常會在典型 DevOps 發行工作流程之外執行修補程式。

超級護理期間

在切換後，遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常，此期間的長度為 1-4 天。在超級護理期間結束時，遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

I

IaC

將基礎設施視為程式碼。

身分型政策

連接至一或多個 IAM 主體的政策，可定義其在 AWS 雲端 環境中的許可。

閒置應用程式

90 天期間 CPU 和記憶體平均使用率在 5% 至 20% 之間的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

IIoT

請參閱工業物聯網。

不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有的基礎設施。不可變基礎設施本質上比可變基礎設施更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的使用不可變基礎設施的部署最佳實務。

傳入 (輸入) VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

工業 4.0

由 Klaus Schwab 於 2016 年推出的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進展，指製造程序的現代化。

基礎設施

應用程式環境中包含的所有資源和資產。

基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱建立工業物聯網 (IIoT) 數位轉型策略。

檢查 VPC

在 AWS 多帳戶架構中，集中式 VPC，可管理 VPCs 之間 （在相同或不同的 中 AWS 區域)、網際網路和內部部署網路之間的網路流量檢查。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱什麼是 IoT？

可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱 的機器學習模型可解譯性 AWS。

IoT

請參閱物聯網。

IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 為 ITSM 提供了基礎。

IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需有關將雲端操作與 ITSM 工具整合的資訊，請參閱操作整合指南。

ITIL

請參閱 IT 資訊庫。

ITSM

請參閱 IT 服務管理。

L

標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會決定使用者可以看到哪些資料列和資料欄。

登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱設定安全且可擴展的多帳戶 AWS 環境。

大型語言模型 (LLM)

預先訓練大量資料的深度學習 AI 模型。LLM 可以執行多個任務，例如回答問題、摘要文件、將文字翻譯成其他語言，以及完成句子。如需詳細資訊，請參閱什麼是 LLMs。

大型遷移

遷移 300 部或更多伺服器。

LBAC

請參閱標籤型存取控制。

最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的套用最低權限許可。

隨即轉移

請參閱 7 個 R。

小端序系統

首先儲存最低有效位元組的系統。另請參閱 Endianness。

LLM

請參閱大型語言模型。

較低的環境

請參閱 環境。

M

機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱機器學習。

主要分支

請參閱分支。

惡意軟體

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊，或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬程式、間諜軟體和鍵盤記錄器。

受管服務

AWS 服務 會 AWS 操作基礎設施層、作業系統和平台，而您會存取端點來存放和擷取資料。Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

製造執行系統 (MES)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

MAP

請參閱遷移加速計劃。

機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的建置機制。

成員帳戶

除了屬於組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。帳戶一次只能是一個組織的成員。

製造執行系統

請參閱製造執行系統。

訊息佇列遙測傳輸 (MQTT)

根據發佈/訂閱模式的輕量型machine-to-machine(M2M) 通訊協定，適用於資源受限的 IoT 裝置。

微服務

一種小型的獨立服務，它可透過定義明確的 API 進行通訊，通常由小型獨立團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱使用無 AWS 伺服器服務整合微服務。

微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務會使用輕量型 API，透過明確定義的介面進行通訊。此架構中的每個微服務都可以進行更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱在 上實作微服務 AWS。

Migration Acceleration Program (MAP)

一種 AWS 計畫，提供諮詢支援、訓練和服務，協助組織建立強大的營運基礎，以移至雲端，並協助抵銷遷移的初始成本。MAP 包括用於有條不紊地執行舊式遷移的遷移方法以及一組用於自動化和加速常見遷移案例的工具。

大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是 AWS 遷移策略的第三階段。

遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。遷移工廠團隊通常包括營運、業務分析師和擁有者、遷移工程師、開發人員以及從事 Sprint 工作的 DevOps 專業人員。20% 至 50% 之間的企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的遷移工廠的討論和雲端遷移工廠指南。

遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。遷移中繼資料的範例包括目標子網路、安全群組和 AWS 帳戶。

遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

遷移組合評定 (MPA)

線上工具，提供驗證商業案例以遷移至 的資訊 AWS 雲端。MPA 提供詳細的組合評定 (伺服器適當規模、定價、總體擁有成本比較、遷移成本分析) 以及遷移規劃 (應用程式資料分析和資料收集、應用程式分組、遷移優先順序，以及波次規劃)。MPA 工具 （需要登入） 可供所有 AWS 顧問和 APN 合作夥伴顧問免費使用。

遷移準備程度評定 (MRA)

使用 AWS CAF 取得組織雲端整備狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序。如需詳細資訊，請參閱遷移準備程度指南。MRA 是 AWS 遷移策略的第一階段。

遷移策略

用來將工作負載遷移至 的方法 AWS 雲端。如需詳細資訊，請參閱此詞彙表中的 7 個 Rs 項目，並請參閱調動您的組織以加速大規模遷移。

機器學習 (ML)

請參閱機器學習。

現代化

將過時的 (舊版或單一) 應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱《》中的現代化應用程式的策略 AWS 雲端。

現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱《》中的評估應用程式的現代化準備 AWS 雲端程度。

單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱將單一體系分解為微服務。

MPA

請參閱遷移產品組合評估。

MQTT

請參閱訊息佇列遙測傳輸。

多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」 或者「這個客戶對哪種產品類別最感興趣？」

可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性， AWS Well-Architected Framework 建議使用不可變基礎設施做為最佳實務。

O

OAC

請參閱原始存取控制。

OAI

請參閱原始存取身分。

OCM

請參閱組織變更管理。

離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

OI

請參閱 操作整合。

OLA

請參閱操作層級協議。

線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

OPC-UA

請參閱開放程序通訊 - 統一架構。

開放程序通訊 - 統一架構 (OPC-UA)

用於工業自動化的machine-to-machine(M2M) 通訊協定。OPC-UA 提供資料加密、身分驗證和授權機制的互通性標準。

操作水準協議 (OLA)

一份協議，闡明 IT 職能群組承諾向彼此提供的內容，以支援服務水準協議 (SLA)。

操作整備審查 (ORR)

問題和相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的操作準備審查 (ORR)。

操作技術 (OT)

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造中，整合 OT 和資訊技術 (IT) 系統是工業 4.0 轉型的關鍵重點。

操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱操作整合指南。

組織追蹤

由 建立的線索 AWS CloudTrail ，會記錄 AWS 帳戶 組織中所有 的所有事件 AWS Organizations。在屬於組織的每個 AWS 帳戶 中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱 CloudTrail 文件中的建立組織追蹤。

組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。在 AWS 遷移策略中，此架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱 OCM 指南。

原始存取控制 (OAC)

CloudFront 中的增強型選項，用於限制存取以保護 Amazon Simple Storage Service (Amazon S3) 內容。OAC 支援所有 S3 儲存貯體、使用 AWS KMS (SSE-KMS) 的伺服器端加密 AWS 區域，以及對 S3 儲存貯體的動態PUT和DELETE請求。

原始存取身分 (OAI)

CloudFront 中的一個選項，用於限制存取以保護 Amazon S3 內容。當您使用 OAI 時，CloudFront 會建立一個可供 Amazon S3 進行驗證的主體。經驗證的主體只能透過特定 CloudFront 分發來存取 S3 儲存貯體中的內容。另請參閱 OAC，它可提供更精細且增強的存取控制。

ORR

請參閱操作整備審核。

OT

請參閱操作技術。

傳出 (輸出) VPC

在 AWS 多帳戶架構中，處理從應用程式內啟動之網路連線的 VPC。AWS 安全參考架構建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

P

許可界限

附接至 IAM 主體的 IAM 管理政策，可設定使用者或角色擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的許可界限。

個人身分識別資訊 (PII)

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。PII 的範例包括名稱、地址和聯絡資訊。

PII

請參閱個人身分識別資訊。

手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

PLC

請參閱可程式設計邏輯控制器。

PLM

請參閱產品生命週期管理。

政策

可定義許可的物件（請參閱身分型政策）、指定存取條件（請參閱資源型政策），或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱服務控制政策）。

混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱在微服務中啟用資料持久性。

組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱評估遷移準備程度。

述詞

傳回 `true`或 的查詢條件`false`，通常位於 `WHERE` 子句中。

述詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這可減少必須從關聯式資料庫擷取和處理的資料量，並改善查詢效能。

預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS上實作安全控制中的預防性控制。

委託人

中可執行動作和存取資源 AWS 的實體。此實體通常是 AWS 帳戶、IAM 角色或使用者的根使用者。如需詳細資訊，請參閱 IAM 文件中角色術語和概念中的主體。

依設計的隱私權

透過整個開發程序將隱私權納入考量的系統工程方法。

私有託管區域

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 查詢的資訊。如需詳細資訊，請參閱 Route 53 文件中的使用私有託管區域。

主動控制

旨在防止部署不合規資源的安全控制。這些控制項會在佈建資源之前對其進行掃描。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的控制項參考指南，並參閱實作安全控制項中的主動控制項。 AWS

產品生命週期管理 (PLM)

管理產品整個生命週期的資料和程序，從設計、開發和啟動，到成長和成熟，再到拒絕和移除。

生產環境

請參閱 環境。

可程式設計邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

提示鏈結

使用一個 LLM 提示的輸出做為下一個提示的輸入，以產生更好的回應。此技術用於將複雜任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和相關性，並允許更精細、個人化的結果。

擬匿名化

將資料集中的個人識別符取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

發佈/訂閱 (pub/sub)

一種模式，可啟用微服務之間的非同步通訊，以改善可擴展性和回應能力。例如，在微服務型 MES 中，微服務可以將事件訊息發佈到其他微服務可訂閱的頻道。系統可以新增新的微服務，而無需變更發佈服務。

Q

查詢計劃

一系列步驟，如指示，用於存取 SQL 關聯式資料庫系統中的資料。

查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

R

RACI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RAG

請參閱擷取增強生成。

勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

RASCI 矩陣

請參閱負責、負責、諮詢、告知 (RACI)。

RCAC

請參閱資料列和資料欄存取控制。

僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

重新架構師

請參閱 7 個 R。

復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

復原時間目標 (RTO)

服務中斷與服務還原之間的可接受延遲上限。

重構

請參閱 7 個 R。

區域

地理區域中的 AWS 資源集合。每個 AWS 區域 都獨立於其他 ，以提供容錯能力、穩定性和彈性。如需詳細資訊，請參閱指定 AWS 區域 您的帳戶可以使用哪些。

迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

重新託管

請參閱 7 個 R。

版本

在部署程序中，它是將變更提升至生產環境的動作。

重新放置

請參閱 7 個 R。

Replatform

請參閱 7 個 R。

回購

請參閱 7 個 R。

彈性

應用程式抵禦中斷或從中斷中復原的能力。在 中規劃彈性時，高可用性和災難復原是常見的考量 AWS 雲端。如需詳細資訊，請參閱AWS 雲端 彈性。

資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣

矩陣，定義所有涉及遷移活動和雲端操作之各方的角色和責任。矩陣名稱衍生自矩陣中定義的責任類型：負責人 (R)、責任 (A)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除它，則稱為 RACI 矩陣。

回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS上實作安全控制中的回應性控制。

保留

請參閱 7 個 R。

淘汰

請參閱 7 個 R。

檢索增強生成 (RAG)

一種生成式 AI 技術，其中 LLM 會在產生回應之前參考訓練資料來源以外的授權資料來源。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱什麼是 RAG。

輪換

定期更新秘密的程序，讓攻擊者更難存取登入資料。

資料列和資料欄存取控制 (RCAC)

使用已定義存取規則的基本、彈性 SQL 表達式。RCAC 包含資料列許可和資料欄遮罩。

RPO

請參閱復原點目標。

RTO

請參閱復原時間目標。

執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

S

SAML 2.0

許多身分提供者 (IdP) 使用的開放標準。此功能會啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS 管理主控台 或呼叫 AWS API 操作，而不必為您組織中的每個人在 IAM 中建立使用者。如需有關以 SAML 2.0 為基礎的聯合詳細資訊，請參閱 IAM 文件中的關於以 SAML 2.0 為基礎的聯合。

SCADA

請參閱監督控制和資料擷取。

SCP

請參閱服務控制政策。

秘密

您以加密形式存放的 AWS Secrets Manager機密或限制資訊，例如密碼或使用者登入資料。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 Secrets Manager 文件中的 Secrets Manager 秘密中的什麼內容？。

設計安全性

透過整個開發程序將安全性納入考量的系統工程方法。

安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全控制有四種主要類型：預防性、偵測性、回應性和主動性。

安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

安全資訊與事件管理 (SIEM) 系統

結合安全資訊管理 (SIM) 和安全事件管理 (SEM) 系統的工具與服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生提醒。

安全回應自動化

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為偵測或回應式安全控制，協助您實作 AWS 安全最佳實務。自動化回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換登入資料。

伺服器端加密

由 AWS 服務 接收資料的 在其目的地加密資料。

服務控制政策 (SCP)

為 AWS Organizations中的組織的所有帳戶提供集中控制許可的政策。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的服務控制政策。

服務端點

的進入點 URL AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考 中的 AWS 服務 端點。

服務水準協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

服務層級指標 (SLI)

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

服務層級目標 (SLO)

代表服務運作狀態的目標指標，由服務層級指標測量。

共同責任模式

描述您與 共同 AWS 承擔雲端安全與合規責任的模型。 AWS 負責雲端的安全，而 負責雲端的安全。如需詳細資訊，請參閱共同責任模式。

SIEM

請參閱安全資訊和事件管理系統。

單一故障點 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

SLA

請參閱服務層級協議。

SLI

請參閱服務層級指標。

SLO

請參閱服務層級目標。

先拆分後播種模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱 中的階段式應用程式現代化方法 AWS 雲端。

SPOF

請參閱單一故障點。

星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構旨在用於資料倉儲或商業智慧用途。

Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由 Martin Fowler 引入，作為重寫單一系統時管理風險的方式。如需有關如何套用此模式的範例，請參閱使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET (ASMX) Web 服務。

子網

您 VPC 中的 IP 地址範圍。子網必須位於單一可用區域。

監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

合成測試

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 Amazon CloudWatch Synthetics 來建立這些測試。

系統提示

一種向 LLM 提供內容、指示或指導方針以指示其行為的技術。系統提示有助於設定內容，並建立與使用者互動的規則。

T

標籤

做為中繼資料以組織 AWS 資源的鍵值對。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱標記您的 AWS 資源。

目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

測試環境

請參閱 環境。

訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

傳輸閘道

可以用於互連 VPC 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的什麼是傳輸閘道。

主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

受信任的存取權

將許可授予您指定的服務，以代表您在組織中 AWS Organizations 及其帳戶中執行任務。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱 文件中的 AWS Organizations 搭配使用 AWS Organizations 與其他 AWS 服務。

調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

雙比薩團隊

兩個比薩就能吃飽的小型 DevOps 團隊。雙披薩團隊規模可確保軟體開發中的最佳協作。

U

不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱量化深度學習系統的不確定性指南。

未區分的任務

也稱為繁重工作，這是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、維護和容量規劃。

較高的環境

請參閱 環境。

V

清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

VPC 對等互連

兩個 VPC 之間的連線，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 Amazon VPC 文件中的什麼是 VPC 對等互連。

漏洞

危害系統安全性的軟體或硬體瑕疵。

W

暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

暖資料

不常存取的資料。查詢這類資料時，通常可接受中等緩慢的查詢。

視窗函數

SQL 函數，對與目前記錄在某種程度上相關的資料列群組執行計算。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器和應用程式。

WORM

請參閱寫入一次，讀取許多。

WQF

請參閱AWS 工作負載資格架構。

寫入一次，讀取許多 (WORM)

儲存模型，可一次性寫入資料，並防止刪除或修改資料。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為不可變。

Z

零時差入侵

利用零時差漏洞的攻擊，通常是惡意軟體。

零時差漏洞

生產系統中未緩解的瑕疵或漏洞。威脅行為者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

零鏡頭提示

提供 LLM 執行任務的指示，但沒有可協助引導任務的範例 (快照)。LLM 必須使用其預先訓練的知識來處理任務。零鏡頭提示的有效性取決於任務的複雜性和提示的品質。另請參閱少量擷取提示。

殭屍應用程式

CPU 和記憶體平均使用率低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。