簡化 Amazon EKS 可觀測性的最佳實務

AWS 方案指引

AWS 方案指引: 簡化 Amazon EKS 可觀測性的最佳實務

Table of Contents

簡化 Amazon EKS 可觀測性的最佳實務

Ishwar Chauthaiwale、Naveen Suthar 和 Pratap Kumar Nanda，Amazon Web Services (AWS)

2025 年 4 月 (文件歷史記錄)

Amazon Elastic Kubernetes Service (Amazon EKS) 需要全方位的可觀測性解決方案，才能有效監控容器化工作負載並進行疑難排解。分散式系統和微服務在 Amazon EKS 環境中具有複雜的架構，因此實作適當的可觀測性實務對於維護可靠的操作至關重要。Amazon EKS 環境中的有效可觀測性可讓團隊深入了解應用程式效能、有效率地疑難排解問題，並維持最佳的叢集運作狀態。

挑戰在於導覽適用於 Amazon EKS 可觀測性的工具和技術龐大生態系統，同時遵守符合組織目標和產業標準的最佳實務。有效的可觀測性策略必須平衡全面的資料收集與效能考量、成本效益和可擴展性。

本指南旨在協助組織在下列領域最佳化其 Amazon EKS 可觀測性：

- 建立有效率的記錄機制
- 實作強大的監控解決方案
- 針對複雜架構使用分散式追蹤
- 實作警示和事件回應策略

透過採用這些最佳實務，您的組織可以增強其深入了解 Amazon EKS 環境的能力，進而提高可靠性、效能和營運效率。這種簡化的可觀測性方法有助於故障診斷和維護，並支援資料驅動型決策，以持續改善以 Kubernetes 為基礎的應用程式和基礎設施。（如需 Amazon EKS 的詳細資訊，請參閱 服務文件。）

本指南深入探討 Amazon EKS 可觀測性的各個層面，並探索您可以量身打造的工具和策略，以滿足 Amazon EKS 部署的特定需求，從小型應用程式到大型、複雜的微服務架構。

在本指南中：

目標

本指南可協助您和組織達成下列業務目標：

- 增強的營運可見性 – 透過有效的可觀測性實務，全面深入了解 Amazon EKS 叢集和應用程式。

  此目標強調在 Amazon EKS 環境中維持完整可見性的重要性。AWS X-Ray、Amazon CloudWatch Container Insights 和 AWS Distro for OpenTelemetry 等工具可協助您了解系統行為、快速識別問題，以及維持最佳效能。
- 改善故障診斷效率 – 透過有效的追蹤和監控策略，減少平均偵測時間 (MTTD) 和平均解決時間 (MTTR)。

  此目標著重於實作可觀測性實務，以快速識別和解決問題。分散式追蹤、有效記錄和全方位指標收集等技術是實現此目標的關鍵。
- 主動式效能管理 – 在潛在問題影響最終使用者之前，讓潛在問題能夠進行早期偵測。

  主動監控對於維持高可用性和效能至關重要。此目標說明實作適當警示、趨勢分析和預測監控以防止服務中斷的重要性。
- 具成本效益的可觀測性 – 最佳化可觀測性成本，同時維持全面的系統可見性。

  成本最佳化包括實作有效率的抽樣策略、適當的資料保留政策和最佳檢測方法。目標是在可觀測性需求與成本考量之間取得平衡，同時確保有效的系統監控。
- 可擴展的監控架構 – 確保您的可觀測性解決方案可與您的 Amazon EKS 環境無縫擴展。

  此目標著重於實作可隨應用程式成長的監控解決方案。無論您是執行單一叢集還是多叢集、多區域部署，您的可觀測性策略都應相應地擴展

在 Amazon EKS 中記錄

記錄是管理和維護在 Amazon EKS 上執行之應用程式的關鍵層面。在 Amazon EKS 環境中有效的記錄實務，可協助開發人員、營運團隊和系統管理員深入了解其容器化應用程式及其基礎基礎設施的行為、效能和運作狀態。

在 Amazon EKS 中實作強大的記錄策略至關重要，原因如下：

- 故障診斷：日誌有助於快速識別和診斷問題，從而減少停機時間並改善整體系統可靠性。
- 合規：許多產業需要全方位記錄以進行稽核和法規。
- 安全性：日誌分析可協助您偵測和調查潛在的安全威脅或違規。
- 效能最佳化：日誌提供應用程式和系統效能的洞見，因此您可以識別瓶頸並最佳化資源使用率。
- 監控和提醒：日誌資料可用來設定監控系統，並針對特定事件或條件觸發提醒。

在本節中：

Amazon EKS 中的記錄類型

在 Amazon EKS 中，記錄涉及擷取、儲存和分析 Kubernetes 叢集不同元件所產生的各種日誌資料類型，包括：

- 系統日誌：基礎 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或AWS Fargate節點的相關資訊
- Kubernetes 元件日誌：來自核心 Kubernetes 元件的資料，例如 API 伺服器、排程器和控制器管理員
- 容器執行期日誌：來自容器執行期的資訊，例如 Docker 或 containerd
- 應用程式日誌：來自容器化應用程式的輸出

若要有效管理 Amazon EKS 環境中的日誌，您通常會採用 AWS 服務第三方工具和最佳實務的組合。這可能包括使用 Amazon CloudWatch、Fluent Bit、Elasticsearch、Kibana 和其他記錄和分析工具來收集、存放和視覺化日誌資料。

下列各節探討在 Amazon EKS 中記錄的各種層面，包括實作 Kubernetes 叢集中完整記錄策略的最佳實務、工具和技術 AWS。

系統日誌

Amazon EKS 中基礎 EC2 執行個體或 Fargate 節點的記錄涉及不同的方法，具體取決於節點類型。

若要在 Amazon EKS 中實作 EC2 執行個體的記錄，您可以使用下列工具：

- CloudWatch 代理程式：在您的 EC2 執行個體上安裝和設定 CloudWatch 代理程式。將其設定為收集系統日誌，例如 `/var/log/messages`和 `/var/log/secure`。您可以使用使用者資料指令碼或組態管理工具來自動化此程序。
- Fluent Bit：將 Fluent Bit 部署為 DaemonSet，從所有節點收集日誌。將其設定為將日誌轉送至 CloudWatch Logs 或其他集中式日誌系統。
- Container Insights：在您的 EKS 叢集中啟用 Container Insights，以自動從 EC2 執行個體收集指標和日誌。
- 自訂指令碼：開發自訂指令碼以收集特定日誌，並將其傳送至您偏好的日誌目的地。
- SSM 代理程式：使用 AWS Systems Manager 代理程式 (SSM 代理程式） 收集日誌並將其轉送至 CloudWatch Logs。

若要在 Amazon EKS 中實作 Fargate 節點的記錄，請使用下列工具：

- Fargate 記錄：Fargate 會自動從您的容器收集 `stdout`和 `stderr` 日誌。設定 Fargate 設定檔以將這些日誌傳送至 CloudWatch Logs。
- 適用於 Fargate 的 Fluent Bit： AWS 提供專門用於 Fargate 記錄的 Fluent Bit 映像。將其部署為 Fargate Pod 中的附屬容器，以收集和轉送日誌。
- 適用於 Fargate 的 Container Insights：啟用 Container Insights 以從 Fargate 節點收集指標和日誌。

Kubernetes 元件日誌

從 Amazon EKS 中的 API 伺服器、排程器和控制器管理員等 Kubernetes 元件收集日誌，需要稍微不同於應用程式記錄的方法。這些元件會做為由 管理的 Amazon EKS 控制平面的一部分執行 AWS。以下是您可以收集和存取這些日誌的方式：

- 啟用控制平面記錄：您可以透過 AWS 管理主控台、 AWS Command Line Interface (AWS CLI) 或基礎設施即程式碼 (IaC) 工具，例如 AWS CloudFormation或 Terraform，為 EKS 叢集啟用控制平面記

錄。當您啟用控制平面記錄時，日誌會傳送至 CloudWatch Logs。您可以在 CloudWatch 主控台中下列日誌群組下檢視它們：

  - `/aws/eks/<cluster-name>/cluster`
  - `/aws/eks/<cluster-name>/kube-apiserver`
  - `/aws/eks/<cluster-name>/kube-scheduler`
  - `/aws/eks/<cluster-name>/kube-controller-manager`
  - `/aws/eks/<cluster-name>/authenticator`
- 使用 CloudWatch Logs Insights：您可以使用 CloudWatch Logs Insights 對日誌執行複雜的查詢。
- 將日誌匯出至 Amazon S3：對於長期儲存或進一步分析，您可以將日誌匯出至 Amazon Simple Storage Service (Amazon S3)。
- 使用第三方工具：您可以使用 Fluent Bit 等工具收集這些日誌，並將其轉送至 Elasticsearch 或 Splunk 等其他日誌系統。
- 使用 AWS CloudTrail： AWS CloudTrail服務可以提供對 EKS 叢集進行 API 呼叫的其他洞見。

容器執行期日誌

在 Amazon EKS 中記錄容器執行期日誌涉及從容器執行期擷取和管理日誌，這通常`containerd`適用於 Amazon EKS。以下是如何在 Amazon EKS 中記錄容器執行期日誌的方法：

- 直接存取 Amazon EC2 節點上的日誌。對於自我管理的 EC2 節點，您可以從這些位置直接存取主機上的容器執行期日誌：
  - `containerd` 日誌： `/var/log/containers/`
  - Docker 日誌（如果您使用的是 Docker 執行時間）： `/var/log/docker.log`
- 使用 DaemonSet 進行日誌收集。
- 將日誌收集代理程式（例如 Fluent Bit) 部署為 DaemonSet，以從所有節點收集日誌。
- 設定 CloudWatch 代理程式以收集容器執行期日誌。
- 啟用 Container Insights 以收集容器執行期指標和日誌。
- 使用 Fargate。對於 Fargate 節點，容器執行期日誌會自動收集，並且可以透過 CloudWatch Logs 存取。
- 使用 Fluent Bit 或 Logstash 等工具實作自訂記錄解決方案。設定 CloudWatch 警示或使用 Prometheus 等工具來監控容器執行期日誌中的特定模式或問題。請考慮使用與 Kubernetes 和 Amazon EKS 完美整合的第三方記錄解決方案，例如 Datadog、Splunk 或 Elastic Stack (ELK Stack)。使用日誌彙總工具從多個來源收集日誌，並將其轉送至集中式日誌系統。

應用程式記錄

Amazon EKS 中的應用程式日誌是維護和疑難排解應用程式的重要部分。若要在 Amazon EKS 中實作應用程式記錄，您可以從下列選項中選擇：

- 將日誌寫入 `stdout/stderr`：處理應用程式日誌最簡單且最原生的 Kubernetes 方法是將日誌寫入 `stdout`和 `stderr`。Kubernetes 會自動擷取這些串流。
- 實作日誌彙總：使用諸如 Fluent Bit 的日誌彙總器，從您的所有 Pod 收集日誌。
- 設定日誌路由：設定您的日誌彙總器，將日誌路由到所需的目的地 （例如 CloudWatch Logs 或 Elasticsearch)。
- 使用 CloudWatch Container Insights：啟用 Container Insights 進行全面的記錄和監控。

登入 Amazon EKS 的最佳實務

下列最佳實務有助於為您的 Amazon EKS 環境建立強大、可擴展且高效的記錄系統，並為您的 Kubernetes 叢集提供更好的疑難排解、監控和整體管理。

- 集中日誌收集：使用集中式日誌解決方案，例如 CloudWatch Logs、Elasticsearch 或第三方服務，從所有元件彙總日誌。這可為日誌分析提供單一存取點，並簡化管理。
- 實作結構化記錄：使用 JSON 等結構化日誌格式，以便更輕鬆地剖析和搜尋日誌。包含相關中繼資料，例如時間戳記、日誌層級和來源識別符。
- 適當使用日誌層級：在您的應用程式中實作適當的日誌層級 （例如 `DEBUG`、`WARN`、 `INFO`和 `ERROR`)。設定生產環境以適當層級記錄，以避免過度記錄。
- 啟用容器記錄：將容器設定為登入 `stdout`和 `stderr`。這可讓 Kubernetes 擷取這些日誌並將其轉送至您選擇的記錄解決方案。
- 實作日誌輪換：設定日誌輪換以管理日誌檔案大小並防止磁碟空間問題。在應用程式中使用 `logrotate`或 實作輪換等工具。
- 使用 Kubernetes DaemonSets 進行日誌收集：將日誌收集代理程式 （例如 Fluent Bit) 部署為 DaemonSets，以確保它們在您的叢集中的每個節點上執行。
- 實作保留政策：定義和強制執行日誌保留政策，以符合法規和管理儲存成本。
- 安全日誌資料：加密傳輸中和靜態的日誌。實作存取控制以限制誰可以檢視和管理日誌。
- 監控日誌擷取：設定日誌擷取失敗或延遲的提醒，以確保持續記錄。
- 使用 Kubernetes 註釋和標籤：使用 Kubernetes 註釋和標籤將中繼資料新增至日誌，以改善可搜尋性和篩選。

- 實作分散式追蹤：使用 AWS X-Ray或 Jaeger 等分散式追蹤工具，跨微服務關聯日誌。
- 最佳化日誌磁碟區：選擇您的日誌內容，以避免不必要的成本和效能問題。對大量、低值日誌使用抽樣。
- 實作日誌彙總：使用 Logstash 等工具從多個來源彙總日誌，再將它們傳送到您的中央記錄系統。
- AWS 服務 盡可能使用：CloudWatch Logs 和 Container Insights 等服務可與其他 無縫整合 AWS 服務。
- 實作日誌分析和視覺化：使用 CloudWatch Logs Insights、Elasticsearch with Kibana 或第三方解決方案等工具進行日誌分析和視覺化。
- 實作自動化日誌分析：使用機器學習和 AI 支援的工具自動偵測日誌中的異常和模式。
- 記錄您的記錄策略：為您的團隊維護記錄架構、實務和工具的明確文件。

登入 Amazon EKS 的重要考量事項

本節討論在 Amazon EKS 中實作記錄時應謹記的重要考量事項。

- 效能影響：記錄過多可能會影響應用程式效能。請注意產生的日誌數量和頻率。
- 成本管理：日誌儲存和處理可能會產生大量成本，尤其是大規模成本。實作日誌保留政策，並考慮使用日誌彙總來降低成本。
- 安全與合規：確保日誌不包含敏感資訊，例如密碼或個人資料。實作傳輸中和靜態日誌的加密。當您處理日誌時，請考慮合規要求，例如一般資料保護法規 (GDPR) 或健康保險流通與責任法案 (HIPAA)。
- 可擴展性：確保您的記錄解決方案可以隨著叢集大小和日誌磁碟區進行擴展。考慮使用緩衝和批次處理進行日誌傳輸。
- 日誌保留：定義和實作適當的日誌保留期間。平衡合規要求與儲存成本。
- 存取控制：實作日誌存取的適當 AWS Identity and Access Management (IAM) 角色和政策。遵循日誌管理的最低權限原則。
- 日誌一致性：在不同應用程式和服務中使用一致的日誌格式。使用結構化記錄，以便於剖析和分析。
- 時間同步：同步所有節點的時間，以取得日誌中的一致時間戳記。
- 資源配置：配置記錄代理程式的適當資源 （例如 CPU 和記憶體）。監控記錄元件的資源用量。
- Fargate 考量：Fargate 具有與 EC2-based節點不同的特定記錄機制。了解 Fargate 記錄的限制和功能。
- 多租用戶叢集：在多租用戶環境中，請確保日誌在租用戶之間正確隔離。
- 日誌剖析和分析：考慮有效日誌分析所需的工具和技能。實作結構化資料擷取的日誌剖析。

- 監控記錄系統：設定記錄基礎設施本身的監控。產生記錄系統故障或待處理項目的提醒。
- 網路影響：請注意日誌傳輸所使用的網路頻寬。考慮對日誌資料使用壓縮。
- Kubernetes 事件：不要忽略 Kubernetes 事件做為重要資訊來源。
- 控制平面記錄：了解啟用控制平面記錄的影響和成本。
- 日誌輪換：實作日誌輪換來管理節點上的檔案大小。
- 除錯功能：請確定您的日誌記錄解決方案允許輕鬆除錯和故障診斷。
- 與現有工具的整合：考慮 Amazon EKS 記錄解決方案如何與現有監控和提醒工具整合。
- 測試：定期測試您的記錄設定，特別是在叢集升級之後。
- 文件：維護記錄架構和實務的清楚文件。
- 日誌彙總延遲：請注意日誌彙總中的任何延遲，以及它如何影響即時監控。

在 Amazon EKS 中監控

在 Amazon EKS 中進行監控，可對 Kubernetes 工作負載的運作狀態、效能和安全性提供重要的可見性。如果沒有適當的監控，您會面臨服務中斷、安全漏洞，以及可能影響業務營運和增加成本的低效率資源使用率的風險。有效的監控可讓您主動識別和解決問題、最佳化資源用量，以及維持容器化應用程式的合規要求。透過實作全方位的監控解決方案，您可以確保高可用性、及早偵測異常狀況，並針對擴展和改善 Amazon EKS 基礎設施做出資料驅動型決策。

本節探討 Amazon EKS 監控的各個層面，包括不同的監控類型、可用的工具和最佳實務，以協助您為 Kubernetes 環境建置強大的監控策略。

在本節中：

- Amazon EKS 中的監控類型
- Amazon EKS 的監控工具
- 實作 Amazon EKS 監控解決方案的高可用性
- 在 Amazon EKS 中監控的最佳實務
- Amazon EKS 中的進階監控考量事項

Amazon EKS 中的監控類型

Amazon EKS 中的有效可觀測性涉及基礎設施、應用程式和安全監控活動。

基礎設施監控

基礎設施監控是 Amazon EKS 可觀測性的基本元件，可讓您深入了解 Kubernetes 叢集基礎元素的運作狀態和效能。其核心涉及追蹤控制平面元件和工作者節點的生命體徵，並確保基礎平台保持穩定和高效。

- 控制平面監控至關重要，因為它會監督 API 伺服器、等資料庫和排程器等關鍵元件。透過監控 API 伺服器延遲，您可以快速識別可能影響應用程式部署或擴展操作的效能瓶頸。Etcd 效能監控會驗證叢集的狀態資料庫是否有效運作，並防止可能影響整個叢集的資料一致性問題。
- 節點層級監控同樣重要，因為它專注於執行容器化工作負載的運算資源。這包括追蹤 CPU 使用率、記憶體使用量、磁碟 I/O，以及所有工作者節點的網路效能。了解這些指標有助於防止資源耗盡、最佳化節點擴展決策，並確保適當的容量規劃。

- 網路監控在維持 Pod、服務和外部資源之間的可靠通訊中扮演重要角色。透過監控網路輸送量、延遲和連線狀態，您可以及早識別連線問題，並確保順暢的應用程式通訊。儲存監控透過追蹤磁碟區效能、容量使用率和 I/O 模式來補充網路監控，以協助防止資料相關的瓶頸。

基礎設施監控可做為潛在問題的早期警告系統、啟用主動維護，並確保最佳資源配置。如果沒有強大的基礎設施監控，您會面臨意外停機時間、效能降低和資源使用效率低落的風險，這可能會大幅影響業務營運和成本。

應用程式監控

應用程式監控對於在您的 Amazon EKS 環境中維護運作狀態良好、效能良好且可靠的容器化應用程式至關重要。此監控層級著重於叢集內實際執行的工作負載，並提供應用程式如何運作、執行和與其他服務互動的重要洞見。

應用程式監控包括容器層級監控、服務層級監控和分散式追蹤。

- 在容器層級，應用程式監控會追蹤重要指標，例如容器運作狀態、重新啟動計數和資源耗用模式。這些指標可協助您識別可能耗用過多資源或頻繁重新啟動的問題容器，這可能表示潛在的問題，例如記憶體流失或組態問題。透過監控容器生命週期事件，您可以確保適當的應用程式行為，並快速疑難排解部署問題。
- 服務層級監控提供應用程式效能和可靠性指標的可見性，例如回應時間、錯誤率和請求輸送量。這些指標對於維護服務層級目標 (SLOs) 並確保正面的最終使用者體驗至關重要。您可以追蹤不同服務端點的延遲、識別效能瓶頸，以及監控錯誤模式以維護應用程式可靠性。
- 分散式追蹤是應用程式監控的另一個關鍵層面，特別是在微服務架構中。透過實作追蹤，您可以在請求通過不同服務時遵循請求、了解相依性，並識別效能瓶頸。這種end-to-end可見性可協助您最佳化服務互動，並針對跨越多個元件的複雜問題進行疑難排解。

自訂應用程式指標在提供業務特定洞見方面扮演重要角色。這些可能包括訂單處理率、使用者登入頻率或交易成功率等指標。您可以將這些自訂指標與基礎設施和容器指標建立關聯，以更加了解基礎設施效能如何影響業務營運，並做出資料驅動的擴展和最佳化決策。

應用程式監控的重要性在於能夠提供應用程式運作狀態和效能的完整檢視。此監控可讓您維持高品質服務、快速解決問題，並持續最佳化您的應用程式以符合業務目標。

安全監控

Amazon EKS 中的安全監控是一項關鍵活動，可協助組織維護其 Kubernetes 環境的完整性、機密性和合規性。這種全面的安全方法結合了持續監控、威脅偵測和合規監控，以保護容器化工作負載免於潛在的安全風險和未經授權的存取。它包含身分驗證和授權監控、網路安全監控，以及組態和合規監控。

- 身分驗證和授權監控會追蹤所有存取叢集的嘗試，形成第一道防線。這包括監控 API 伺服器請求、追蹤成功和失敗的登入嘗試，以及稽核角色型存取控制 (RBAC) 變更。透過維護存取哪些資源和何時存取的詳細稽核日誌，您可以快速偵測潛在的安全漏洞、未經授權的存取嘗試或權限提升活動。這在維護嚴格存取控制至關重要的多租戶環境中尤為重要。
- 網路安全監控著重於偵測和防止 Pod 和服務之間未經授權的通訊。透過監控網路政策違規和不尋常的流量模式，您可以識別潛在的安全威脅，例如容器逸出嘗試或叢集內的橫向移動。這包括同時追蹤內部叢集通訊和外部流量模式，以確保容器僅與授權端點通訊，並遵循定義的安全政策。
- 組態和合規監控對於維護安全基準和滿足法規要求至關重要。它涉及持續掃描容器映像是否有漏洞、監控執行期安全性，以及追蹤可能影響安全性狀態的組態變更。定期合規稽核可確保遵守產業標準和組織安全政策，而組態偏離偵測有助於防止可能導致安全風險的未經授權變更。

Amazon EKS 中的安全監控提供必要的可見性和控制，有助於防範現代安全威脅，同時確保符合法規要求。透過實作全面的安全監控，您的組織可以維持強大的安全狀態、快速回應安全事件，並證明符合各種法規標準。

Amazon EKS 的監控工具

本節討論三種 Amazon EKS 監控工具： AWS 監控服務、開放原始碼或專屬解決方案，以及專用工具。

AWS 服務

- Amazon CloudWatch：全方位監控和記錄服務

  CloudWatch 構成 AWS 監控解決方案的骨幹，並為 Amazon EKS 環境提供廣泛的功能。它為精細容器和叢集指標提供 Container Insights，因此您可以監控效能、資源使用率和應用程式運作狀態。此服務在日誌彙總和分析方面表現卓越，並支援跨容器和節點的集中式記錄。CloudWatch 自然與 整合 AWS 服務。它提供自動化警示組態，並支援自訂指標和儀表板，這使得它成為 Amazon EKS 監控的必要工具。
- AWS X-Ray：進階分散式追蹤平台

X-Ray 透過提供複雜的分散式追蹤功能來提升可觀測性。其服務地圖視覺化提供應用程式架構和相依性的清晰洞見，詳細的請求追蹤有助於識別跨服務的效能瓶頸。X-Ray 可以透過複雜的微服務架構追蹤請求，這使得它對於故障診斷和最佳化非常寶貴，尤其是在跨越多個 的分散式系統中 AWS 服務。

- AWS Distro for OpenTelemetry：統一可觀測性架構

  Distro for OpenTelemetry 提供統一的資料收集功能與跨平台支援，因此非常適合混合環境。此服務與其他 整合 AWS 服務，支援自訂檢測，並提供實作全方位監控解決方案的彈性，同時維持與業界標準的相容性。

- Amazon Managed Grafana：企業級視覺化

  Amazon Managed Grafana 為資料視覺化和分析提供全受管服務。它提供與其他 AWS 服務內建安全功能的無縫整合，以及企業級可擴展性。服務可簡化儀表板的建立和管理，同時提供進階功能，例如跨帳戶資料來源存取和與 的整合 AWS IAM Identity Center。

- Amazon Managed Service for Prometheus：高可用性、安全、受管監控

  Amazon Managed Service for Prometheus 是一種全受管、與 Prometheus 相容的監控服務。它提供自動化擴展、高可用性和安全指標擷取和查詢。此服務與 Amazon EKS 無縫整合，並消除管理 Prometheus 伺服器的操作開銷。

開放原始碼或專屬解決方案

上一節所述的 AWS 工具提供無縫整合和受管服務。本節中列出的開放原始碼工具透過提供彈性和廣泛的自訂選項 AWS 服務 來補充 。了解每個工具的功能和使用案例，可協助您設計最符合您特定需求的監控策略。

- Prometheus：指標收集工具組

  Prometheus 是在 Kubernetes 環境中收集指標的開放原始碼解決方案。其時間序列資料庫和 PromQL 查詢語言可實現複雜的指標分析。平台的服務探索功能會自動適應動態 Kubernetes 環境，其警示管理系統可讓您隨時掌握重大問題。Prometheus 提供廣泛的整合選項，使其成為全面指標監控的多樣化選擇。

- Grafana：進階視覺化引擎

  Grafana 透過其視覺化功能，將複雜的監控資料轉換為可行的洞察。平台會建立自訂儀表板，結合來自多個來源的資料，並提供基礎設施和應用程式指標的統一檢視。其支援各種資料來源和警示管理功

能，可提供全方位的監控。Grafana 可協助您視覺化即時和歷史資料，以便識別趨勢並做出明智的決策。

- Fluent Bit：統一記錄層

  此日誌記錄解決方案提供 Kubernetes 環境的日誌收集和管理。其原生 Kubernetes 整合可確保從容器和節點無縫收集日誌，且其對多個輸出目的地的支援可提供日誌儲存和分析的彈性。日誌剖析和篩選等進階功能可讓您根據特定需求處理和路由日誌。Fluent Bit 的輕量性質使其特別適合容器化環境。

- Datadog：全堆疊可觀測性

  Datadog 提供具有原生 Kubernetes 支援的全方位監控功能。它提供基礎設施監控、應用程式效能監控 (APM)、日誌管理和即時分析。您可以使用平台的自動服務探索和廣泛的整合目錄進行 Amazon EKS 監控，以及其機器學習功能來偵測異常並預測潛在問題。

- 新複本：應用程式效能監控

  New Relic 提供應用程式效能和基礎設施運作狀態的可見性。其 Kubernetes 整合提供詳細的容器洞見、分散式追蹤和自訂儀表板。平台可協助您將應用程式效能與基礎設施指標相互關聯，以便快速識別和解決問題。

- Elastic Stack (ELK Stack)：日誌分析和搜尋

  ELK Stack 結合了 Elasticsearch、Logstash 和 Kibana，以提供日誌管理和分析功能。它提供進階搜尋功能、視覺化工具和機器學習功能。您可以使用 堆疊來處理來自 Amazon EKS 環境的大量日誌資料。

專用工具

您可以根據您的特定監控需求、操作規模和組織偏好設定，混合和比對下列工具。關鍵是建立監控堆疊，提供全面的可見性，同時保持可管理且符合成本效益。

- kube-state-metrics (KSM)：Kubernetes 狀態監控

  此附加元件服務會接聽 Kubernetes API 伺服器，並產生物件狀態的指標。它提供部署、Pod 和其他 Kubernetes 資源運作狀態的洞見。

- Kubernetes 指標伺服器：資源指標

  此指標伺服器會從 kubelet 收集資源指標，並透過 Kubernetes 指標 API 公開這些指標。它提供水平 Pod 自動擴展和基本 CPU 和記憶體指標。

- Kubecost：Kubernetes 成本監控

  Kubecost 等工具為 EKS 叢集提供詳細的成本分析和最佳化建議。它們可協助您了解和最佳化不同命名空間、部署和服務之間的雲端支出。

實作 Amazon EKS 監控解決方案的高可用性

Amazon EKS 監控的強大高可用性 (HA) 策略對於確保 Kubernetes 環境的持續可見性至關重要。本節討論在監控基礎設施的不同層面實作 HA 的全方位方法。

架構備援和可擴展性

建置高度可用的監控系統從適當的架構設計開始。監控元件應分散到多個 AWS 可用區域，以防止區域故障。這包括為 Prometheus 伺服器、日誌收集器和警示管理員等關鍵監控元件實作水平擴展。您可以使用 Amazon Managed Service for Prometheus 和 Amazon Managed Grafana 等 AWS 受管服務，協助降低營運開銷，同時確保高可用性。設定自動容錯移轉機制以在元件故障期間維持服務連續性，並實施運作狀態檢查和自動復原程序。

彈性資料儲存策略

資料儲存彈性是維護監控系統可靠性的基礎。實作分散式儲存解決方案可確保即使個別儲存節點故障，指標資料和日誌仍可存取。這包括跨多個可用區域設定適當的資料複寫，以及使用不同的儲存後端進行備援。為歷史資料建立定期備份程序，並針對各種失敗案例記錄復原程序。對於 Prometheus 等時間序列資料庫，實作遠端儲存解決方案有助於將儲存考量與資料收集分開，並改善整體系統可靠性。

備援提醒管理

警示管理需要特別注意 HA 設定。部署備援警示管理員可確保即使在系統故障期間，重要通知仍可送達預期的收件人。設定多個通知管道，例如電子郵件、簡訊、Slack 和 PagerDuty，以提供替代通訊路徑。使用警示重複資料刪除機制，以防止在部分系統故障期間產生警示風暴，以及確保永遠不會遺漏關鍵警示的備用通知方法。實作警示相互關聯有助於在容錯移轉案例期間維護內容，並防止來自備援系統的重複通知。

負載平衡和服務探索

適當的負載平衡對於維持穩定的監控服務至關重要。 AWS Application Load Balancer 會將傳入監控流量分散到多個端點，運作狀態檢查可確保流量只會路由至運作狀態良好的執行個體。服務探索機制可協助監控元件自動適應環境中的變更，例如新增節點或服務。使用 DaemonSets 在所有節點上一致地部署監控代理程式，以確保叢集擴展時的全面涵蓋。

其他 HA 考量事項

網路彈性：

- 實作備援網路路徑。
- 跨可用區域設定適當的子網路設計。
- AWS Direct Connect 搭配備份路由使用 。
- 設定適當的安全群組和網路存取控制清單 (網路 ACLs)。

監控監視器：

- 部署次要監控系統。
- 實作跨區域監控。
- 設定無回應系統的提醒。
- 定期測試容錯移轉程序。

容量規劃：

- 監控資源用量趨勢。
- 實作預測擴展。
- 定期測試效能。

資料管理：

- 實作資料保留政策。
- 設定指標彙總。
- 規劃資料生命週期管理。
- 定期最佳化儲存體。

復原程序：

- 文件復原程序。
- 定期測試災難復原。
- 盡可能實作自動化復原。

- 識別並實作明確的呈報路徑。

透過實作這些高可用性實務，您可以確保 Amazon EKS 監控基礎設施保持可靠性和彈性，即使在各種故障情況下，也能持續了解 Kubernetes 環境。這些 HA 組態的定期測試和更新可確保它們在環境演進時保持有效。

在 Amazon EKS 中監控的最佳實務

策略實作方法

成功的 Amazon EKS 監控策略從規劃良好的分階段實作方法開始。

- 首先識別和監控直接影響業務營運和應用程式可靠性的關鍵指標。此基礎應包含基本基礎設施指標、關鍵應用程式效能指標和關鍵安全性指標。根據營運需求和經驗教訓逐漸擴展監控涵蓋範圍，並確保每個新增項目都提供有意義的價值。
- 使用基礎設施即程式碼 (IaC) 工具，例如 Terraform 或 來實作自動化部署程序 CloudFormation ，以確保一致性和可重複性。
- 測試和驗證監控系統，以協助維持可靠性和準確性。
- 持續精簡監控參數，以符合不斷變化的業務需求。

有效的資料管理

適當的資料管理對於維護高效且符合成本效益的監控解決方案至關重要。

- 實作明確的資料保留政策，以平衡歷史分析需求與儲存成本。
- 為不同的指標類型設定適當的取樣率：關鍵指標的頻率較高，較不關鍵指標的頻率較低。
- 使用指標彙總來減少資料量，同時保持有意義的洞察，尤其是長期趨勢分析。
- 實作系統性日誌輪換和封存程序來管理儲存成本，同時確保重要資料仍可存取。
- 考慮為日誌儲存實作hot-warm-cold架構，以最佳化存取速度和成本效益。

警示組態和管理

警示組態需要仔細考慮以維持有效性，而不會造成警示疲勞。

- 根據服務水準目標 (SLOs) 和歷史效能模式，定義明確、可行的閾值。

- 實作分層警示嚴重性系統，以清楚區分需要立即關注的關鍵問題，以及較不緊急的問題。
- 確保提醒提供足夠的內容和可行資訊，以便快速解決問題。
- 建立明確的呈報程序，並定義不同警示嚴重性的擁有權和回應時間。
- 定期檢閱和精簡警示組態，以協助維持其相關性和有效性。

資源優化

持續監控資源使用率對於維護經濟實惠的操作至關重要。

- 在所有叢集元件中實作全面的資源監控，包括節點、Pod 和持久性磁碟區。
- 根據實際使用模式和效能需求設定自動擴展，以確保有效率的資源使用率，同時維持效能。
- 使用成本分配標籤來追蹤不同團隊、應用程式或環境的資源消耗。
- 定期分析資源效率指標，以識別最佳化機會並實作改善。
- 考慮實作成本管理工具來追蹤和最佳化雲端支出。

安全性

安全考量應該是監控策略不可或缺的一部分。

- 為所有監控元件實作最低權限存取原則，以確保使用者和服務只有他們所需的許可。
- 啟用全面的稽核記錄，以追蹤監控系統的所有存取和變更。
- 定期執行監控組態和存取模式的安全審查，以識別潛在的漏洞。
- 對傳輸中和靜態的敏感監控資料實作加密。
- 將安全監控與現有的安全資訊和事件管理 (SIEM) 系統整合，以獲得全面的安全可見性。

Amazon EKS 中的進階監控考量事項

效能最佳化：

- 最佳化指標收集間隔。
- 設定有效率的查詢模式。
- 實作指標預先彙總。
- 使用適當的儲存解決方案。

合規與控管：

- 維護稽核線索。
- 實作合規監控。
- 提供定期合規報告。
- 文件監控程序。

災難復原：

- 定期備份監控組態。
- 文件復原程序。
- 測試復原程序。

持續改進：

- 定期監控檢閱工作階段。
- 最佳化效能週期。
- 根據事件更新監控。
- 納入使用者意見回饋。

這些最佳實務提供架構，以實作和維護 Amazon EKS 環境的有效監控解決方案。定期檢閱和更新這些實務，使其與您的組織需求和業界標準保持一致。監控不是一次性設定，而是需要定期關注和改進的持續程序。

在 Amazon EKS 中追蹤

追蹤是 Amazon EKS 中應用程式可觀測性的重要元件。追蹤透過收集、處理和視覺化在 EKS 叢集上部署的各種微服務中的請求路徑，提供請求流程和服務互動的詳細可見性。此功能可協助您了解系統行為、識別瓶頸，以及有效疑難排解 Amazon EKS 環境中的問題。透過提供請求流程end-to-end 可見性，有效的追蹤可消除對分散式系統進行偵錯的複雜性。這可讓您跨服務界限追蹤交易，並識別 Amazon EKS 工作負載內的效能問題或故障。

Amazon EKS 的整體追蹤實作可讓您了解系統行為、最佳化效能，以及維護容器化應用程式的可靠性。最後，追蹤功能可增強 Amazon EKS 環境中的操作可見性和系統可維護性。

AWS X-Ray 在追蹤應用程式相關資料中扮演重要角色。追蹤涉及監控服務互動的各個層面，包括下列項目：

- 請求路徑和相依性提供分散式系統行為的重要洞見。當請求周遊不同的微服務和元件時，他們會追蹤請求的完整旅程。映射服務相依性可協助您了解通訊模式，並識別應用程式架構中的關鍵路徑。如需實作詳細資訊，請參閱 X-Ray 文件中的使用 AWS X-Ray 服務追蹤映射。
- 服務延遲和瓶頸是維持最佳系統效能的重要指標。透過測量和分析服務之間的回應時間，您可以有效地識別效能問題。此資料可讓您精確找出導致請求鏈延遲的特定服務或操作，並實現目標最佳化工作。若要進一步了解延遲分析，請參閱 X-Ray 文件中的與 Analytics 主控台互動。
- 錯誤傳播模式可協助您了解系統可靠性和容錯能力。透過追蹤跨 服務的錯誤路徑，了解失敗如何透過系統串聯，您可以更好地建構您的應用程式。這種可見性可協助您識別錯誤的根本原因及其對相依服務的影響，這會導致更具彈性的系統。如需實作詳細資訊，請參閱 X-Ray 文件中的追蹤。
- 跨服務的資源使用率提供對系統效率和成本最佳化的洞察。您可以監控與追蹤資料相關的 CPU、記憶體和網路使用模式，以了解資源需求。此資料可協助您分析資源消耗趨勢，以最佳化 EKS 叢集的服務效能和成本。如需監控設定，請參閱 Amazon EKS 文件中的監控叢集效能和檢視日誌。
- 最終使用者交易流程對於了解和改善使用者體驗至關重要。透過追蹤從前端到後端服務的完整使用者互動，您可以確保最佳的應用程式效能。您可以測量和最佳化關鍵使用者旅程的end-to-end回應時間，這將直接影響客戶滿意度。若要實作最終使用者監控，請使用程式設計語言的 AWS X-Ray SDK。
- API 閘道互動形成應用程式效能和安全性的前線。您可以在 API 進入點監控請求模式和效能，以確保最佳服務交付。此可見性可協助您追蹤身分驗證、授權和速率限制對請求流程的影響，以同時維持安全和效能需求。進一步了解 Amazon API Gateway with X-Raydocumentation 中的 API 追蹤。

Amazon EKS 中的有效追蹤不僅止於收集範圍和追蹤。它需要結構良好的策略，在可觀測性需求與系統效能之間取得平衡。此策略應著重於：

- 實作適當的抽樣率：根據流量模式和業務優先順序設定抽樣規則，以最佳化成本，同時保持關鍵交易的可見性。若要進一步了解，請參閱 X-Ray 文件中的設定抽樣規則。
- 定義要追蹤的關鍵路徑和服務：識別需要詳細追蹤以確保最佳效能監控的基本服務和使用者旅程，並排定其優先順序。如需詳細資訊，請參閱 Amazon EKS 文件中的使用 ADOT Operator 傳送指標和追蹤資料。
- 建立適當的資料保留政策：設定資料生命週期管理規則，以平衡可觀測性需求與儲存成本和合規需求。若要檢視 CloudWatch 保留政策，請參閱 CloudWatch Logs 文件中的使用日誌群組和日誌串流。
- 設定有效的視覺化和分析工具：部署和設定視覺化工具，例如 AWS X-Ray Analytics 主控台或 Amazon Managed Grafana，以有效分析追蹤資料。如需詳細資訊，請參閱 X-Ray 文件中的與 Analytics 主控台互動。

在本節中：

Amazon EKS 的追蹤工具

Amazon EKS 支援數個 AWS 和第三方選項來實作分散式追蹤。

AWS 服務

- AWS X-Ray：進階分散式追蹤平台

  X-Ray 是全受管 AWS 服務 的，可提供end-to-end追蹤功能。它會自動檢測 AWS 服務 ，並為在 Amazon EKS 上執行的應用程式提供詳細的服務地圖和分析。X-Ray 與其他 整合 AWS 服務，包括 Amazon CloudWatch，並提供追蹤與 AWS 服務 呼叫的自動關聯性。

- AWS Distro for OpenTelemetry：統一可觀測性架構

  Distro for OpenTelemetry 是適用於雲端原生應用程式的安全、生產就緒和 AWS支援的 OpenTelemetry 分佈。它提供廠商中立的檢測功能，同時維持原生 AWS 服務 整合，因此非常適合混合雲端環境。Distro for OpenTelemetry 支援多個可觀測性後端，並提供與 AWS 監控服務的無縫整合。

開放原始碼解決方案

- OpenTelemetry：開放原始碼可觀測性架構

  OpenTelemetry 提供標準化的可觀測性架構，其中包含支援多種程式設計語言的全方位檢測程式庫。其靈活的後端選項和廠商中立方法，非常適合需要跨不同環境一致性的工作負載。框架的廣泛生態系統可確保與各種監控解決方案的廣泛相容性。

- Jaeger：開放原始碼分散式追蹤平台

  Jaeger 提供具有即時分散式內容傳播的全方位追蹤功能。它透過詳細的服務相依性視覺化來提供根本原因分析和效能最佳化。Jaeger 的架構專為高可擴展性而設計，並支援各種儲存後端，因此適合大規模的 Amazon EKS 部署。EKS 設定的 ViewJaeger

- Grafana Tempo：分散式追蹤

  Tempo 是一種 Grafana 實驗室解決方案，可提供大規模追蹤儲存，並與 Prometheus 指標無縫整合。其具有成本效益的追蹤保留模型以及與 Grafana 的原生整合，使其適用於已使用 Grafana 進行視覺化的組織。Tempo 的架構專為 Amazon EKS 等雲端原生環境而設計。

在 Amazon EKS 中追蹤的最佳實務

本節提供建立有效追蹤系統的完整最佳實務和技術清單，以增強 Amazon EKS 中以 Kubernetes 為基礎的應用程式的可觀測性和故障診斷。

- 策略抽樣：根據您應用程式的流量模式和您正在使用之服務的重要性，設定不同的抽樣率。為關鍵路徑實作更高的取樣率，同時減少大量、較不關鍵的路由取樣，以最佳化成本。如需指引，請參閱 AWS X-Ray 文件中的設定抽樣規則。
- 檢測設定：使用 X-Ray SDK 或 AWS Distro for OpenTelemetry 收集器等自動檢測工具，將手動檢測工作降至最低。維持跨 服務的一致命名慣例和內容傳播，以獲得更好的追蹤關聯性。如需詳細資訊，請參閱 Distro for OpenTelemetry 收集器文件。
- 資料管理：實作適當的保留期和壓縮策略，以平衡儲存成本與您的可觀測性需求。建立明確的資料隱私權控制和備份程序，以保護敏感的追蹤資料。如需詳細資訊，請參閱 CloudWatch Logs 文件中的變更 CloudWatch Logs 中的日誌資料保留。 CloudWatch
- 效能最佳化：監控和最佳化追蹤額外負荷，將對應用程式效能的影響降至最低。使用有效的緩衝和非同步處理，以減少延遲影響。如需詳細資訊，請參閱 X-Ray 文件中的設定 AWS X-Ray 協助程式。
- 安全控制：使用 IAM 角色和政策實作適當的存取控制和資料保護措施。定期安全稽核和合規審查有助於確保追蹤資料保持安全。如需詳細資訊，請參閱 X-Ray 文件中的 安全 AWS X-Ray。

- 監控和提醒：設定追蹤集合運作狀態的全面監控，並設定集合問題的提醒。追蹤取樣率和系統效能指標，以確保最佳操作。如需詳細資訊，請參閱 CloudWatch 文件中的 Container Insights。
- 高可用性：跨可用區域部署備援收集器，並設定適當的容錯移轉機制。定期測試高可用性設定可確保可靠的追蹤收集。如需詳細資訊，請參閱《Amazon Managed Service for Prometheus 文件》中的使用 AWS Distro for OpenTelemetry 做為收集器。

透過遵循這些最佳實務，您可以為您的 Amazon EKS 環境建立強大、有效率且有效的追蹤系統。這將有助於確保 Kubernetes 應用程式的完整可觀測性、有效率的故障診斷和最佳效能。

Amazon EKS 中的提醒

警示是管理和維護在 Amazon EKS 上執行之應用程式的關鍵元件。它可做為早期警告系統，在潛在問題、異常或效能降低情況升級為可能影響服務可用性或使用者體驗的嚴重問題之前，通知操作員和開發人員。提醒涉及監控 Kubernetes 叢集的各個層面，包括：

- 基礎設施運作狀態
- 應用程式效能
- 容器指標
- 自訂業務指標

Amazon EKS 中的有效提醒不僅止於設定通知。它需要一個well-thought-out的策略，在及時資訊的需求與提醒疲勞的可能性之間取得平衡。此策略應該：

- 定義有意義的閾值和條件。
- 根據嚴重性和影響排定警示的優先順序。
- 實作適當的路由和呈報程序。
- 與事件管理和通訊工具整合。

在本節中：

- Amazon EKS 警示工具
- 在 Amazon EKS 中提醒的最佳實務

Amazon EKS 警示工具

Amazon EKS 支援數個 AWS 和第三方選項來實作提醒。當您選擇 Amazon EKS 警示的工具時，請考慮整合功能、可擴展性、易用性、成本和符合監控和警示需求的特定功能等因素。許多組織使用這些工具的組合，為其 Amazon EKS 環境建立全面的監控和提醒解決方案。

- Amazon CloudWatch： AWS 服務 用於監控和可觀測性

  CloudWatch 為 EKS 叢集提供指標、日誌和警示，並與其他叢集完美整合 AWS 服務。
- Prometheus：Kubernetes 的開放原始碼監控和提醒工具

  Prometheus 提供強大的查詢語言 (PromQL) 來定義警示條件。

- 警示管理員：搭配 Prometheus 處理警示

  Alertmanager 提供重複資料刪除、分組和警示路由。它支援各種通知管道，包括電子郵件、Slack 和 PagerDuty。

- Grafana：用於監控和可觀測性的開放原始碼平台

  Grafana 提供視覺化和提醒功能。它可以與各種資料來源整合，包括 Prometheus 和 CloudWatch。

- Elastic Stack (ELK Stack)：Elasticsearch、Logstash 和 Kibana 的組合

  此工具適用於日誌彙總、分析和提醒。它可以透過 Elastic 的可觀測性功能進行擴展。

- 第三方解決方案

  市場上有許多可用的工具，包括 Datadog、New Relic、Sysdig、Dynatrace、Zabbix、Nagios、Splunk、IBM Instana 和 AppDynamics。

在 Amazon EKS 中提醒的最佳實務

本節說明建立強大警示系統的最佳實務，以增強 Amazon EKS 中以 Kubernetes 為基礎的應用程式的可靠性和效能。

定義清除提醒閾值：

- 根據歷史資料和業務需求設定有意義的閾值。
- 在適當的情況下使用動態閾值，以考慮不同的工作負載。

實作警示優先順序：

- 依嚴重性分類警示（例如，嚴重、高、中、低）。
- 使警示優先順序與業務影響保持一致。

避免警示疲勞：

- 透過消除備援或低值警示來減少雜訊。
- 將警示與群組相關問題建立關聯。

使用多階段提醒：

- 在達到關鍵層級之前實作警告閾值。
- 針對不同的提醒嚴重性使用不同的通知管道。

實作適當的提醒路由：

- 確定警示已傳送給正確的團隊或個人。
- 使用全天、每天涵蓋範圍的隨需排程和輪換。

利用 Kubernetes 原生指標：

- 監控核心 Kubernetes 元件（節點、Pod、服務）。
- 針對其他 Kubernetes 物件指標使用 kube-state-metrics (KSM)。

監控基礎設施和應用程式：

- 設定叢集運作狀態、節點狀態和資源使用率的提醒。
- 實作應用程式特定的提醒，例如錯誤率和延遲。

使用 Prometheus 和 Alertmanager：

- 使用 Prometheus 收集指標和 PromQL 定義警示條件。
- 使用 Alertmanager 進行警示路由和重複資料刪除。

與 Amazon CloudWatch 整合：

- 針對 Amazon EKS 特定指標使用 CloudWatch Container Insights。
- 設定關鍵 AWS 資源指標的 CloudWatch 警示。

實作內容豐富的提醒：

- 在警示訊息中包含相關資訊，例如叢集名稱、命名空間和 Pod 詳細資訊。
- 在提醒中提供相關儀表板或 Runbook 的連結。

使用異常偵測：

- 針對複雜模式實作機器學習型異常偵測。
- 使用 CloudWatch 異常偵測或第三方工具等服務。

實作提醒抑制和靜音：

- 允許暫時抑制已知問題。
- 實作維護時段，以減少規劃停機時間期間的噪音。

監控警示效能：

- 追蹤警示頻率、解決時間和誤報率等指標。
- 根據這些指標定期檢閱和精簡提醒規則。

實作呈報程序：

- 定義未解決警示的明確呈報路徑。
- 使用 PagerDuty 或 Opsgenie 等工具進行自動呈報。

定期測試提醒系統：

- 定期測試您的警示管道。
- 在災難復原演練中包含提醒測試。

使用範本來確保警示一致性：

- 為常見案例建立標準化提醒範本。
- 確保所有警示的格式和資訊一致。

實作速率限制：

- 對經常觸發的警示實作速率限制，以防止警示風暴。

使用自訂指標：

- 實作自訂指標以進行應用程式特定的監控。

- 使用 Kubernetes 自訂指標 API 根據這些指標進行自動擴展。

實作記錄整合：

- 將警示與相關日誌建立關聯，以加快故障診斷速度。
- 搭配提醒系統使用 Grafana Loki 或 ELK Stack 等工具。

考慮成本提醒：

- 設定資源用量或成本意外尖峰的提醒。
- 使用 AWS Budgets 或第三方成本管理工具。

使用分散式追蹤：

- 整合分散式追蹤工具，例如 Jaeger 或 AWS X-Ray。
- 設定異常追蹤模式或延遲的提醒。

文件提醒 Runbook：

- 為每個警示類型建立清晰、可行的 Runbook。
- 在 Runbook 中包含故障診斷步驟和呈報程序。

透過遵循這些最佳實務，您可以為 Amazon EKS 環境建立強大、有效率且有效的提醒系統。這將有助於確保以 Kubernetes 為基礎的應用程式的高可用性、快速問題解決和最佳效能。

後續步驟

本指南提供在 Amazon EKS 環境中實作強大可觀測性的完整架構，著重於指標收集、記錄基礎設施、分散式追蹤和成本最佳化。透過了解並套用這些核心元件，您可以建置高度可觀測、可維護且符合成本效益的容器環境，以深入了解應用程式和基礎設施行為。Amazon CloudWatch Container Insights 和 AWS 服務 等 的整合AWS X-Ray，結合 Prometheus 和 OpenTelemetry 等開放原始碼解決方案，為監控和疑難排解容器化應用程式建立了強大的基礎。

實作成功倚賴分階段的方法，從核心指標收集開始，逐步擴展到全面的記錄和分散式追蹤功能。我們建議您先評估目前的監控功能、找出差距，以及選擇符合您營運需求和團隊專業知識的適當工具組合。這種有條不紊的方法可確保可觀測性堆疊的每個元件都已正確實作和整合，而團隊則開發必要的技能和程序，以有效地使用這些工具。

Amazon EKS 可觀測性的長期永續性取決於成本、資源和程序的定期最佳化。您應該持續檢閱和調整可觀測性基礎設施，包括資料保留政策、取樣率和資源配置，以維持全面監控與營運效率之間的適當平衡。這種改善的反覆方法結合持續的團隊訓練和文件更新，可讓您的組織維持有效的可觀測性，同時支援業務成長並適應不斷變化的應用程式架構。

資源

AWS 文件

- Amazon EKS 最佳實務指南
- Amazon CloudWatch Container Insights
- Amazon Managed Service for Prometheus
- Amazon Managed Grafana
- AWS Distro for OpenTelemetry 和 AWS X-Ray
- Amazon OpenSearch Service

AWS 部落格文章

- Amazon EKS 增強了 Kubernetes 控制平面可觀測性
- 使用 Amazon Managed Service for Prometheus 受管湊集器在 Amazon EKS 上自動化指標收集
- 使用 CloudWatch Container Insights 自動化 Amazon EKS 叢集的監控
- 使用 Amazon EKS 的受管監控解決方案增強可觀測性

其他資源

- OpenTelemetry 文件
- Prometheus 文件
- Fluent Bit 文件
- Kubernetes 文件中的監控、記錄和偵錯

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
初次出版	—	2025 年 4 月 10 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 儲存貯體中的所有伺服器端加密 AWS KMS (SSE-KMS) 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)

當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，以提高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。

重新定位

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

定義所有涉及遷移活動和雲端操作之各方的角色和責任的矩陣。矩陣名稱衍生自矩陣中定義的責任類型：負責人 (R)、責任 (A)、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 Rs。

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。此功能可啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS 管理主控台 或呼叫 AWS API 操作，而無需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。