模式

AWS 方案指引

AWS 方案指引: 模式

Table of Contents

AWS 規定指引模式

Amazon Web Services (AWS) 規範指導模式提供實作特定雲端遷移、現代化和部署案例的指 step-by-step 示、架構、工具和程式碼。這些模式經過主題專家的審核 AWS，適用於計劃或正在遷移到的建設者和實際操作使用者。 AWS他們還支持已經在使用 AWS 並正在尋找優化雲端操作或現代化的方法的使用者。

無論您是處於專案的概念驗證、規劃還是實作階段 AWS ，都可以使用這些模式將不同複雜度的內部部署或雲端工作負載移至並加速雲端採用、最佳化和現代化工作。例如，對於雲端移轉專案：

- 在規劃階段中，您可以評估可移轉至的不同選項 AWS。您可以根據您要重新定位、重新裝載、重新平台或重新架構，選擇符合您需求的正確模式。您也可以瞭解可用於移轉的各種工具，並開始規劃取得授權或開始與廠商的初始對話。
- 在概念驗證和實作階段中，您可以依照模式中提供的 step-by-step 指示將工作負載移轉至 AWS。每個模式都包含諸如先決條件、目標參考架構、工具、工 step-by-step 作、最佳實務、疑難排解和程式碼等詳細資料。
- 如果您已經在使用 AWS 雲端，您可以找到可協助您現代化、最佳化、擴充和保護雲端資源使用的模式。

若要依技術領域檢視模式清單，請使用下列連結或「AWS 規範指引」首頁上的篩選和搜尋選項。

- 分析
- 企業生產力
- 雲端原生
- 容器與微服務
- 內容交付
- 成本管理
- 資料湖
- 資料庫
- DevOps
- 使用者運算
- 高效能運算
- 混合雲
- 基礎建

- IoT
- 機器學習與人工智慧
- 大型机
- 管理與治理
- 訊息與通訊
- 移民
- 現代化
- 聯網
- 作業系統
- 操作
- SaaS 服務
- 安全性、身分識別、合規
- 無伺服器
- 軟體開發與測試
- 儲存與備份
- 網頁及行動應用程式

若要檢視所有出版品，包括指南、策略和模式，請參閱AWS 規範指引首頁。

分析

主題

- 在 Microsoft SQL 服務器分析服務中分析 Amazon Redshift 數據
- 使用 Amazon 雅典娜和亞馬遜分析和視覺化嵌套 JSON 數據 QuickSight
- 使用 AWS CloudFormation 範本在 AWS Glue 中自動執行加密
- 使用 AWS Glue 建立 ETL 服務管道，以遞增方式將資料從 Amazon S3 載入到亞馬遜紅移
- 使用 AWS 服務計算風險值 (VaR)
- 將太數據標準化時間功能轉換為 Amazon Redshift SQL
- 將太數據重置功能轉換為 Amazon Redshift SQL
- 啟動時強制標記 Amazon EMR 叢集
- 確保啟動時已啟用 Amazon S3 的亞馬遜 EMR 記錄功能
- 使用 AWS AWS Glue 任務和 Python 產生測試資料
- 使用 Lambda 函數在暫態 EMR 叢集中啟動星火工作
- 使用 AWS Glue 將阿帕奇卡桑德拉工作負載遷移到 Amazon Keyspaces
- 將 Oracle 商業智慧 12c 從現場部署伺服器遷移到 AWS 雲端
- 使用以下方式將現場部署阿帕奇卡夫卡叢集遷移到 Amazon MSK MirrorMaker
- 將 ELK 堆疊遷移到 AWS 上的彈性雲端
- 使用星爆將資料遷移到 AWS 雲端
- 優化 AWS 上輸入檔案大小的 ETL 擷取
- 使用 AWS Step Functions 透過驗證、轉換和分割協調 ETL 管道
- 使用 Amazon Redshift ML 執行進階分析
- 使用 Athena 存取、查詢和加入 Amazon DynamoDB 資料表
- 設定最小的可行資料空間，以便在組織之間共用資料
- 使用標量 Python UDF 為 Amazon Redshift 查詢結果設置特定語言排序
- 訂閱 Lambda 函數，以便從不同 AWS 區域的 S3 儲存貯體發出的事件通知
- 三種 AWS Glue ETL 任務類型，可將資料轉換為 Apache 實木地板
- 使用 Amazon 雅典娜和亞馬遜視覺化亞馬遜 Redshift 審核日 QuickSight
- 使用 Amazon 將所有 AWS 帳戶的 IAM 登入資料報告視覺化 QuickSight
- 更多模式

在 Microsoft SQL 服務器分析服務中分析 Amazon Redshift 數據

創建者蘇尼爾·沃拉 (AWS)

環境：PoC 或試點	來源：Amazon Redshift	目標：Microsoft SQL 伺服器分析服務
R 類型：不適用	工作量：Microsoft	技術：分析
AWS 服務：Amazon Redshift		

Summary

此模式描述如何連接和分析 Amazon Redshift 數據庫中的數據，通過使用智能軟 OLE 數據庫提供程序或 CDATA ADO.NET 提供程序進行數據庫訪問。

Amazon Redshift 是一種在雲端中完全受管的 PB 級資料倉儲服務。SQL Server 分析服務是一種線上分析處理 (OLAP) 工具，可用來分析來自資料集市和資料倉儲 (例如 Amazon Redshift) 的資料。您可以使用 SQL Server 分析服務，從您的資料建立 OLAP 多維資料集，以進行快速、進階的資料分析。

先決條件和限制

假設

- 此模式說明如何在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上為 Amazon Redshift 設定 SQL 伺服器分析服務和智慧軟體 OLE 資料庫提供者或 CDATA ADO.NET 提供者。或者，您也可以將兩者安裝在公司資料中心的主機上。

先決條件

- 有效的 AWS 帳戶
- 具有登入資料的 Amazon Redshift 叢集

架構

源, 技術, 堆棧

- 亞 Amazon Redshift 集群

目標技術堆疊

- Microsoft SQL 伺服器分析服務

來源與目標架構

工具

- Microsoft 視覺工作室 2019（社區版）
- Amazon Redshift 的智能軟 OLE 數據庫提供商（試用）或 Amazon Redshift 的 CDATA ADO.NET 提供商（試用）

史诗

分析表格

任務	描述	所需技能
分析要匯入的表格和資料。	識別要匯入的 Amazon Redshift 表格及其大小。	DBA

設定 EC2 執行個體並安裝工具

任務	描述	所需技能
設定 EC2 執行個體。	在您的 AWS 帳戶中，在私有或公有子網路中建立 EC2 執行個體。	系統管理員
安裝用於數據庫訪問的工具。	下載並安裝適用於 Amazon Redshift 的智能軟 OLE 數據庫提供商（或 Amazon Redshift	系統管理員

任務	描述	所需技能
	的 CDATA ADO.NET 提供商）。	
安裝視覺工作室。	下載並安裝視覺工作室 2019（社區版）。	系統管理員
安裝擴充功能。	安裝 Microsoft 分析服務項目擴展視覺工作室。	系統管理員
建立專案。	在視覺工作室中建立新的表格式模型專案，以存放您的 Amazon Redshift 資料。在 Visual Studio 中，在建立專案時，選擇 [分析服務表格式專案] 選項。	DBA

建立資料來源並匯入資料表

任務	描述	所需技能
建立一個 Amazon Redshift 資料來源。	使用適用於 Amazon Redshift 的智慧型 OLE 資料庫提供者 (或 Amazon Redshift 的 CDATA ADO.NET 提供者) 和您的 Amazon Redshift 登入資料，建立 Amazon Redshift 資料來源。	Amazon Redshift, DBA
匯入資料表。	從 Amazon Redshift 選擇表格和視圖並將其導入到您的 SQL 服務器分析服務項目中。	Amazon Redshift, DBA

移轉後清理

任務	描述	所需技能
刪除 EC2 執行個體。	刪除先前啟動的 EC2 執行個體。	系統管理員

相關資源

- Amazon Redshift（AWS 文檔）
- 安裝 SQL 伺服器分析服務 (Microsoft 文件)
- 表格式模型設計器（Microsoft 文檔）
- 用於進階分析的 OLAP 多維資料集概觀 (Microsoft 文件)
- Microsoft 視覺工作室 2019（社區版）
- Amazon Redshift 智能軟件 OLE 數據庫提供商（試用）
- Amazon Redshift 的 CDATA 提供商（試用）

使用 Amazon 雅典娜和亞馬遜分析和視覺化嵌套 JSON 數據 QuickSight

創建者：阿諾普·辛格 (AWS)

環境：PoC 或試點	技術：分析；資料庫	AWS 服務：Amazon Athena；Amazon QuickSight

Summary

此模式說明如何使用 Amazon Athena 將巢狀 JSON 格式的資料結構轉換為表格式檢視，然後在 Amazon 中將資料視覺化。 QuickSight

您可以將 JSON 格式的資料用於作業系統的 API 驅動資料饋送，以建立資料產品。這些數據還可以幫助您更好地了解客戶及其與產品的互動情況，以便您量身定制用戶體驗並預測結果。

先決條件和限制

先決條件

- 一個活躍的 AWS 帳戶
- 代表巢狀資料結構的 JSON 檔案 (此模式提供範例檔案)

限制：

- JSON 功能與 Athena 現有的 SQL 導向函數完美整合。但是，它們與 ANSI SQL 不兼容，並且 JSON 文件預計將每個記錄在單獨的行上進行。您可能需要使用 Athena 中的`ignore.malformed.json`屬性來指出格式錯誤的 JSON 記錄是否應轉換為空字元或產生錯誤。如需詳細資訊，請參閱 Athena 文件中讀取 JSON 資料的最佳做法。
- 這種模式只考慮簡單和少量 JSON 格式的數據。如果您想要大規模使用這些概念，請考慮套用資料分割，然後將資料合併為較大的檔案。

架構

下圖顯示此模式的架構和工作流程。嵌套數據結構以 JSON 格式存儲在 Amazon Simple Storage Service (Amazon S3) 中。在 Athena，JSON 資料會對應至 Athena 資料結構。然後，您可以建立檢視來分析資料，並將中的資料結構視覺化 QuickSight。

工具

AWS 服務

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。這種模式使用 Amazon S3 來存儲 JSON 文件。
- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。此模式使用 Athena 來查詢和轉換 JSON 資料。只要在中執行一些動作 AWS Management Console，您就可以將 Athena 指向 Amazon S3 中的資料，並使用標準 SQL 執行一次性查詢。Athena 是無伺服器服務，因此無需設定或管理基礎結構，而且您只需為執行的查詢付費。Athena 會自動擴充並並行執 parallel 查詢，因此即使是大型資料集和複雜的查詢，結果也很快。
- Amazon QuickSight 是雲端規模商業智慧 (BI) 服務，可協助您在單一儀表板上視覺化、分析和報告資料。 QuickSight 可讓您輕鬆建立和發佈包含機器學習 (ML) 深入解析的互動式儀表板。您可以從任何裝置存取這些儀表板，並將其嵌入到您的應用程式、入口網站和網站中。

範例程式碼

下列 JSON 檔案提供您可以在此模式中使用的巢狀資料結構。

```
{
  "symbol": "AAPL",
  "financials": [
    {
      "reportDate": "2017-03-31",
      "grossProfit": 20591000000,
      "costOfRevenue": 32305000000,
      "operatingRevenue": 52896000000,
      "totalRevenue": 52896000000,
      "operatingIncome": 14097000000,
      "netIncome": 11029000000,
```

```
      "researchAndDevelopment": 2776000000,
      "operatingExpense": 6494000000,
      "currentAssets": 101990000000,
      "totalAssets": 334532000000,
      "totalLiabilities": 200450000000,
      "currentCash": 15157000000,
      "currentDebt": 13991000000,
      "totalCash": 67101000000,
      "totalDebt": 98522000000,
      "shareholderEquity": 134082000000,
      "cashChange": -1214000000,
      "cashFlow": 12523000000,
      "operatingGainsLosses": null
    }
  ]
}
```

史诗

設定 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	若要建立儲存貯體來存放 JSON 檔案，請登入 AWS Management Console，開啟 Amazon S3 主控台，然後選擇 [建立儲存貯體]。如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。	系統管理員
新增巢狀的 JSON 資料。	將您的 JSON 檔案上傳到 S3 儲存貯體。如需 JSON 檔案範例，請參閱上一節。如需指示，請參閱 Amazon S3 文件中的上傳物件。	系統管理員

分析 Athena 的資料

任務	描述	所需技能
創建一個用於映射 JSON 數據的表。	1. 開啟 A thena 主控台。 2. 依照 Athena 文件中的指示建立資料庫。 3. 從 [資料庫] 功能表中，選擇您建立的資料庫。 4. 在查詢編輯器中，輸入如下CREATE TABLE陳述式： `CREATE EXTERNAL TABLE financials_json (symbol string, financials array< struct<reportdate: string, grossprofit: bigint, totalrevenue: bigint, totalcash: bigint, totaldebt: bigint, researchanddevelopment: bigint>>) ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe' LOCATION 's3://s3bucket-for-athena/'` 其中LOCATION指定包含 JSON 檔案的 S3 儲存貯體的位置。 5. 選擇「執行」以建立表格。	開發人員

任務	描述	所需技能
	如需有關建立資料表的詳細資訊，請參閱 Athena 文件。	

任務	描述	所需技能
建立用於資料分析的檢視。	1. 開啟 A thena 主控台。 2. 依照 Athena 文件中的指示建立資料庫。 3. 從 [資料庫] 功能表中，選擇您建立的資料庫。 4. 在查詢編輯器中，輸入如下CREATE VIEW陳述式： `CREATE OR REPLACE VIEW financial_json_view AS SELECT symbol, financials[1].reportdate one_report_date, -- indexes start with 1 financials[1].totalrevenue one_total_revenue, financials[1].reportdate another_report_date, financials[1].totalrevenue another_total_revenue FROM financials_json where symbol='AAPL' ORDER BY 1` 5. 選擇 Run (執行) 以建立檢視。 如需有關建立檢視的詳細資訊，請參閱 Athena 文件。	開發人員

任務	描述	所需技能
分析和驗證數據。	1. 開啟 A thena 主控台。 2. 在查詢編輯器中，使用您在上一個步驟中建立的檢視來執行查詢。 3. 根據 JSON 檔案驗證資料，以確認資料行名稱和資料類型已正確對應。	開發人員

將資料視覺化 QuickSight

任務	描述	所需技能
將 Athena 設定為中的資料來源 QuickSight。	1. 開啟 QuickSight 主控台。 2. 選擇資料集，再選擇新增資料集。 3. 選擇 Athena 作為資料來源。 4. 選擇包含您所建立之檢視的資料庫。 5. 選擇您要為其建立資料集的檢視表。 6. 在 [完成資料集建立] 頁面上，選擇 [直接查詢您的資料]。 7. 選擇 Visualize (視覺化)。	系統管理員
視覺化中的資料 QuickSight。	1. 將資料集視覺化後，請從左窗格中選擇視覺效果，然後選擇資料集的欄位。若要取得更多資訊，請參閱 QuickSight 文件中的自學課程。	資料分析

任務	描述	所需技能
	2. 將變更儲存至分析。 3. 選擇 [發佈儀表板] 以發佈您建立的視覺效果。	

相關資源

- Amazon Athena 文
- Amazon QuickSight 教程
- 使用嵌套 JSON（博客文章）

使用 AWS CloudFormation 範本在 AWS Glue 中自動執行加密

創建者：迪奧戈·蓋德斯（AWS）

程式碼儲存庫：AWS Glue 加密強制	環境：生產	技術：分析；安全性、身分識別、合規
工作負載：所有其他工作	AWS 服務：Amazon EventBridge；AWS AWS Glue；AWS KMS；AWS Lambda；AWS CloudFormation	

Summary

此模式說明如何使用 AWS CloudFormation 範本在 AWS Glue 中設定和自動執行加密。範本會建立強制加密的所有必要組態和資源。這些資源包括初始組態、Amazon EventBridge 規則建立的預防控制，以及 AWS Lambda 函數。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 部署 CloudFormation 範本及其資源的權限

限制

這種安全控制是區域性的。您必須在要在 AWS Glue 中設定加密強制執行的每個 AWS 區域部署安全控制。

架構

目標技術堆疊

- Amazon CloudWatch 日誌（來自 AWS Lambda）
- Amazon EventBridge 法則

- AWS CloudFormation 堆疊
- AWS CloudTrail
- AWS Identity and Access Management (IAM) 受管角色和政策
- AWS Key Management Service (AWS KMS)
- AWS KMS 別名
- AWS Lambda 功能
- AWS Systems Manager 參數存放區

目標架構

下圖顯示如何在 AWS Glue 中自動執行加密。

該圖顯示以下工作流程：

1. CloudFormation 範本會建立所有資源，包括 AWS Glue 中加密強制執行的初始組態和偵探控制。
2. EventBridge 規則會偵測加密組態中的狀態變更。
3. 系統會叫用 Lambda 函數，以便透過記錄進 CloudWatch 行評估和記錄。對於不合規的偵測，會使用 AWS KMS 金鑰的 Amazon 資源名稱 (ARN) 來復原參數存放區。在啟用加密的情況下，服務會修復為合規狀態。

自動化和規模

如果您使用 AWS Organizations，可以使用 AWS CloudFormation StackSets 在多個帳戶中部署此範本，以便在 AWS Glue 中啟用加密強制執行。

工具

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，Lambda 函數、使用 API 目標的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS 可 CloudTrail協助您啟用 AWS 帳戶的操作和風險稽核、管理和合規。

- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。

Code

此模式的代碼可在 GitHub aws 自定義護欄事件驅動的存儲庫中找到。

最佳實務

AWS Glue 支援靜態資料加密，可在 AWS Glue 中撰寫任務，以及使用開發端點開發指令碼。

請考慮下列最佳作法：

- 設定 ETL 任務和開發端點，以使用 AWS KMS 金鑰寫入靜態加密資料。
- 使用您透過 AWS KMS 管理的金鑰，加密存放在 AWS Glue 資料型錄中的中繼資料。
- 使用 AWS KMS 金鑰加密任務書籤以及爬網程式和 ETL 任務產生的日誌。

史诗

啟動 CloudFormation 範本

任務	描述	所需技能
部署 CloudFormation 範本。	從 GitHub 存放庫下載aws-custom-guardrail-event-driven.yaml 範本，然後部署範本。狀CREATE_COMPLETE 態表示您的範本已成功部署。 注意：範本不需要輸入參數。	雲端架構師

驗證 AWS Glue 中的加密設定

任務	描述	所需技能
檢查 AWS KMS 金鑰組態。	1. 登入 AWS 管理主控台，然後開啟 AWS Glue 主控台。 2. 在導覽窗格的 [資料目錄] 下，選擇 [目錄設定]。 3. 確認「中繼資料加密」和「加密」連線密碼設定已標記並設定為使用KMSKeyGlue 。	雲端架構師

測試加密強制

任務	描述	所需技能
識別中的加密設定 CloudFormation。	1. 登入 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 在導覽窗格中，選擇 [堆疊]，然後選擇您的堆疊。 3. 選擇 Resources (資源) 標籤。 4. 在「資源」表中，依「邏輯 ID」尋找加密設定。	雲端架構師
將佈建的基礎結構切換至不符合標準的狀態。	1. 登入 AWS 管理主控台，然後開啟 AWS Glue 主控台。 2. 在導覽窗格的 [資料目錄] 下，選擇 [目錄設定]。 3. 清除「中繼資料加密」核取方塊。	雲端架構師

任務	描述	所需技能
	4. 清除 [加密連線密碼] 核取方塊。 5. 選擇儲存。 6. 重新整理 AWS Glue 主控台。 清除核取方塊後，防護會偵測到 AWS Glue 中的不合規狀態，然後透過自動修復加密錯誤設定來強制執行合規。因此，在重新整理頁面之後，應再次選取加密核取方塊。	

相關資源

- 在 AWS CloudFormation 主控台建立堆疊 (AWS CloudFormation 文件)
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail (Amazon CloudWatch 文件)
- 在 AWS Glue 中設定加密 (AWS Glue 文件)

使用 AWS Glue 建立 ETL 服務管道，以遞增方式將資料從 Amazon S3 載入到亞馬遜紅移

創建者：羅漢牙買加尼（AWS）和阿魯納巴達塔（AWS）

環境：生產	技術：分析、資料湖、儲存與備份	AWS 服務：Amazon Redshift；Amazon S3；AWS AWS Glue；AWS Lambda

Summary

此模式提供有關如何設定 Amazon Simple Storage Service (Amazon S3) 以獲得最佳資料湖效能，然後使用 AWS Glue 將增量資料變更從 Amazon S3 載入 Amazon Redshift，以及執行擷取、轉換和載入 (ETL) 操作。

Amazon S3 中的來源檔案可以有不同的格式，包括逗號分隔值 (CSV)、XML 和 JSON 檔案。此模式說明如何使用 AWS Glue 將來源檔案轉換成成本最佳化且效能最佳化的格式，例如 Apache Parquet。您可以直接從亞馬遜雅典娜和亞馬 Amazon Redshift Spectrum 查詢實木複合地板文件。您也可以將實木複合地板檔案載入 Amazon Redshift、彙總這些檔案，並與消費者共用彙總的資料，或使用 Amazon QuickSight 將資料視覺化。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 具有適當權限且包含 CSV、XML 或 JSON 檔案的 S3 來源儲存貯體。

假設

- CSV、XML 或 JSON 來源檔案已經載入到 Amazon S3，而且可以從設定 AWS Glue 和 Amazon Redshift 的帳戶存取。
- 我們遵循載入檔案、分割檔案、壓縮和使用資訊清單的最佳實務，如 Amazon Redshift 文件所述。
- 來源檔案結構未變更。

- 來源系統能夠遵循 Amazon S3 中定義的資料夾結構，將資料內嵌到 Amazon S3。
- Amazon Redshift 叢集跨越單一可用區域。(這個架構是適當的，因為 AWS Lambda、AWS Glue 和 Amazon Athena 都是無伺服器的。) 為了達到高可用性，叢集快照會以一般頻率拍攝。

限制

- 檔案格式僅限於 AWS Glue 目前支援的檔案格式。
- 不支援即時下游報告。

架構

源, 技術, 堆棧

- 包含 CSV、XML 或 JSON 檔案的 S3 儲存貯體

目標技術堆疊

- S3 資料湖 (使用分割的 Parquet 檔案儲存)
- Amazon Redshift

目標架構

資料流

工具

- Amazon S3 — 亞馬遜簡單儲存服務 (Amazon S3) 是可高度擴展的物件儲存服務。Amazon S3 可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- AWS Lambda — AWS Lambda 可讓您執行程式碼，而無需佈建或管理伺服器。AWS Lambda 是一項事件驅動型服務；您可以設定程式碼以從其他 AWS 服務自動啟動。
- Amazon Redshift — Amazon Redshift 是一種全受管的 PB 級資料倉儲服務。使用 Amazon Redshift，您可以使用標準 SQL 跨資料倉儲和資料湖查詢數 PB 的結構化和半結構化資料。

- AWS Glue — AWS Glue 是全受管 ETL 服務，可讓您更輕鬆地準備和載入資料以進行分析。AWS Glue 會探索您的資料，並將相關的中繼資料 (例如，表格定義和結構描述) 存放在 AWS Glue 資料型錄中。您的目錄數據可以立即搜索，可以查詢，並且可用於 ETL。
- AWS Secrets Manager — AWS Secrets Manager 可協助保護和集中管理應用程式或服務存取所需的機密。此服務會儲存資料庫認證、API 金鑰和其他機密，並且不需要以純文字格式對敏感資訊進行硬式編碼。Secrets Manager 還提供金鑰輪替，以滿足安全性和合規性需求。它具有 Amazon 紅移，亞馬 Amazon Relational Database Service（亞馬遜 RDS）和 Amazon DocumentDB 的內置集成。您可以使用秘密管理員主控台、命令列介面 (CLI) 或機密管理員 API 和軟體開發套件來儲存和集中管 Secrets Manager。
- Amazon Athena 娜 — 亞馬遜雅典娜是一種互動式查詢服務，可讓您輕鬆分析存放在 Amazon S3 中的資料。Athena 是無伺服器且與 AWS Glue 整合，因此可以直接查詢使用 AWS Glue 編目的資料。Athena 經過彈性調整，可提供互動式查詢效能。

史诗

建立 S3 儲存貯體和資料夾結構

任務	描述	所需技能
分析來源系統的資料結構和屬性。	針對貢獻給 Amazon S3 資料湖的每個資料來源執行此任務。	數據工程師
定義分區和訪問策略。	此策略應根據數據捕獲的頻率，增量處理和消耗需求。確定 S3 儲存貯體不對外開放，而且存取權限僅由特定服務角色型政策控制。如需詳細資訊，請參閱 Amazon S3 說明文件。	數據工程師
為每個資料來源類型建立單獨的 S3 儲存貯體，並為已處理的 (Parquet) 資料建立每個來源個別的 S3 儲存貯體。	為每個來源建立個別值區，然後根據來源系統的資料擷取頻率建立資料夾結構；例如，`s3://source-system-name/date/hour` 。對於已處理 (轉換為 Parquet 格	數據工程師

任務	描述	所需技能
	式) 的檔案，請建立類似的結構；例如，`s3://source-processed-bucket/date/hour` 。如需建立 S3 儲存貯體的詳細資訊，請參閱 Amazon S3 文件。	

在 Amazon Redshift 中建立資料倉儲

任務	描述	所需技能
使用適當的參數群組以及維護和備份策略來啟動 Amazon Redshift 叢集。	在建立 Amazon Redshift 叢集時，請將密碼管理員資料庫密碼用於管理員使用者登入資料。如需建立和調整 Amazon Redshift 叢集大小的相關資訊，請參閱 Amazon Redshift 文件和調整雲端資料倉儲大小白皮書。	數據工程師
建立 IAM 服務角色並將其連接至 Amazon Redshift 叢集。	AWS Identity and Access Management (IAM) 服務角色可確保存取 Secrets Manager 和來源 S3 儲存貯體。如需詳細資訊，請參閱有關授權和新增角色的 AWS 文件。	數據工程師
建立資料庫結構描述。	請遵循表格設計的 Amazon Redshift 最佳實務。根據使用案例，選擇適當的排序和分佈索引鍵，以及可能的最佳壓縮編碼。如需最佳實務，請參閱 AWS 文件。	數據工程師

任務	描述	所需技能
設定工作負載管理。	根據您的需求，確定工作負載管理 (WLM) 佇列、短查詢加速 (SQA) 或並行擴展。如需詳細資訊，請參閱 Amazon Redshift 文件中的實作工作負載管理。	數據工程師

在密碼管理員中建立密碼

任務	描述	所需技能
創建一個新的密碼來存儲在 Secrets Manager Amazon Redshift 登錄憑據。	此密碼會儲存 admin 使用者以及個別資料庫服務使用者的認證。如需指示，請參閱 Secrets Manager 文件。選擇 Amazon Redshift 叢集作為密碼類型。此外，在「秘密」旋轉頁面上，開啟旋轉。這會在 Amazon Redshift 叢集中建立適當的使用者，並且會在定義的間隔輪換金鑰密碼。	數據工程師
建立 IAM 政策以限制 Secrets Manager 存取權。	限制 Secrets Manager 員只能存取 Amazon Redshift 管理員和 AWS Glue。	數據工程師

設定 AWS AWS Glue

任務	描述	所需技能
在 AWS Glue 資料型錄中，為 Amazon Redshift 新增連線。	如需指示，請參閱 AWS Glue 文件。	數據工程師

任務	描述	所需技能
為 AWS Glue 建立和附加 IAM 服務角色，以存取 Secrets Manager、Amazon Redshift 和 S3 儲存貯體。	如需詳細資訊，請參閱 AWS Glue 文件。	數據工程師
定義來源的 AWS Glue 資料型錄。	這個步驟涉及在 AWS Glue 資料型錄中建立資料庫和必要的資料表。您可以使用爬蟲程式將 AWS Glue 資料庫中的表格分類，或將其設定為 Amazon Athena 外部表格。您也可以透過 AWS Glue 資料型錄存取在 Athena 設定的外部表格。請參閱 AWS 文件，以取得有關在 Athena 清除資料目錄和建立外部表格的詳細資訊。	數據工程師
建立 AWS Glue 任務以處理來源資料。	AWS Glue 任務可以是 Python 殼層，也可 PySpark 以是標準化、重複資料刪除和清理來源資料集。若要優化效能並避免查詢整個 S3 來源儲存貯體，請按日期對 S3 儲存貯體進行分割，並按年、月、日和小時劃分，做為 AWS Glue 任務的下推述詞。如需詳細資訊，請參閱 AWS Glue 文件。以 Parquet 格式將已處理和轉換的資料載入已處理的 S3 儲存貯體分割區。您可以查詢 Athena 的鑲木地板。	數據工程師

任務	描述	所需技能
建立 AWS Glue 任務以將資料載入 Amazon Redshift。	AWS Glue 任務可以是 Python 殼層，也可以透過提升資料 PySpark 來載入資料，然後進行完整的重新整理。如需詳細資訊，請參閱 AWS Glue 文件和其他資訊一節。	數據工程師
(選擇性) 視需要使用觸發程式來排程 AWS Glue 任務。	增量資料載入主要由 Amazon S3 事件所驅動，該事件會導致 AWS Lambda 函數呼叫 AWS Glue 任務。針對需要以時間為基礎的任何資料載入使用 AWS Glue 觸發器型排程，而不是以事件為基礎的排程。	數據工程師

建立 Lambda 函數

任務	描述	所需技能
為 AWS Lambda 建立並附加 IAM 服務連結角色，以存取 S3 儲存貯體和 AWS AWS Glue 任務。	建立 AWS Lambda 的 IAM 服務連結角色，其中包含讀取 Amazon S3 物件和儲存貯體的政策，以及存取 AWS Glue API 以開始 AWS Glue 任務的政策。如需詳細資訊，請參閱知識中心。	數據工程師
建立 Lambda 函數以根據定義的 Amazon S3 事件執行 AWS AWS Glue 任務。	Lambda 函數應該由建立 Amazon S3 資訊清單資訊清單來啟動。Lambda 函數應將 Amazon S3 資料夾位置 (例如，來源儲存貯體/年/月/日/小時) 作為參數傳遞給 AWS Glue 任務。AWS Glue 任務將使用此參數做為下推述詞，以優化	數據工程師

任務	描述	所需技能
	存取和任務處理效能。如需詳細資訊，請參閱 AWS Glue 文件。	
建立 Amazon S3 PUT 物件事件以偵測物件建立，並呼叫相應的 Lambda 函數。	Amazon S3 PUT 物件事件應該只能透過建立資訊清單資訊清單來啟動。資訊清單可控制 Lambda 函數和 AWS Glue 任務並行處理，並以批次方式處理載入，而不是處理到達 S3 來源儲存貯體特定分區的個別資料。如需詳細資訊，請參閱 L ambda 文件。	數據工程師

相關資源

- Amazon S3 文件
- AWS AWS Glue 文件
- Amazon Redshift 文檔
- AWS Lambda
- Amazon Athena
- AWS Secrets Manager

其他資訊

提供更新和完整重新整理的詳細方法

Upsert：這適用於需要歷史彙總的資料集，視業務使用案例而定。根據您的業務需求，遵循更新和插入新資料 (Amazon Redshift 文件) 中所述的方法之一。

完整重新整理：這適用於不需要歷史彙總的小型資料集。請遵循下列其中一種方法：

1. 截斷 Amazon Redshift 表。
2. 從暫存區域載入目前的分割區

或：

1. 使用目前的分割區資料建立暫存資料表。
2. 放下目標 Amazon Redshift 表。
3. 將暫存資料表重新命名為目標資料表。

使用 AWS 服務計算風險值 (VaR)

創建者：蘇門薩曼塔 (AWS)

環境：PoC 或試點	技術：分析；無伺服器	AWS 服務：Amazon Kinesis Data Streams；AWS Lambda；Amazon SQS；Amazon ElastiCache

Summary

此模式說明如何使用 AWS 服務實作風險值 (VaR) 計算系統。在內部部署環境中，大多數 VaR 系統都使用大型專用基礎架構，以及內部或商業網格排程軟體來執行批次程序。此模式提供簡單、可靠且可擴展的架構，以處理 AWS 雲端中的 VaR 處理。它建立的無伺服器架構使用 Amazon Kinesis 資料串流作為串流服務、使用 Amazon Simple Queue Service (Amazon SQS) 做為受管佇列服務、Amazon ElastiCache 做為快取服務，以及 AWS Lambda 處理訂單和計算風險。

風險價值是一種統計指標，交易者和風險經理用於估計其投資組合中的潛在損失超出一定信心水平。大多數 VaR 系統都涉及運行大量的數學和統計計算並存儲結果。這些計算需要大量的運算資源，因此 VaR 批次程序必須分成較小的運算工作集。將大批量拆分為較小的任務是可能的，因為這些任務大多是獨立的（也就是說，一個任務的計算不依賴於其他任務）。

VaR 架構的另一個重要需求是運算延展性。此模式使用無伺服器架構，該架構會根據運算負載自動向內或向外擴展。由於批次或線上運算需求難以預測，因此需要動態擴展才能在服務層級協定 (SLA) 所強制的時間表內完成程序。此外，一旦資源上的工作完成，成本最佳化架構應該能夠縮減每個運算資源的規模。

AWS 服務非常適合 VaR 計算，因為它們提供可擴展的運算和儲存容量、以成本最佳化方式進行處理的分析服務，以及執行風險管理工作流程的不同類型排程器。此外，您只需為在 AWS 上使用的運算和儲存資源付費。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶

- 輸入文件，這取決於您的業務需求。典型的使用案例包含下列輸入檔案：
  - 市場資料檔案 (輸入 VaR 計算引擎)
  - 交易數據文件（除非交易數據通過流傳輸）。
  - 配置數據文件（模型和其他靜態配置數據）
  - 計算引擎模型檔案 (定量資料庫)
  - 時間序列資料檔案 (用於過去五年的歷史資料，例如股票價格)
- 如果市場資料或其他輸入是透過串流傳入，則會設定 Amazon Kinesis Data Streams，並設定 Amazon Identity and Access Management (IAM) 許可以寫入串流。

這種模式構建了一個架構，在其中將交易數據從交易系統寫入 Kinesis 數據流。您可以將交易資料儲存在小批次檔案中，將它們存放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中，然後叫用事件以開始處理資料，而不是使用串流服務。

限制

- Kinesis 資料串流排序可保證在每個碎片上，因此寫入多個碎片的交易訂單無法保證以與寫入作業相同的順序交付。
- AWS Lambda 執行階段限制目前為 15 分鐘。如需詳細資訊，請參閱 Lambda 常見問題集。)

架構

目標架構

下列架構圖顯示風險評估系統的 AWS 服務和工作流程。

此圖展示了以下要點：

1. 交易從訂單管理系統流入。
2. 工單位置淨值 Lambda 函數會處理訂單，並將每個股票代碼的合併訊息寫入 Amazon SQS 中的風險佇列。
3. 風險計算引擎 Lambda 函數會處理來自 Amazon SQS 的訊息、執行風險計算，以及更新 Amazon 風險快取中的 VaR 損益 (PnL) 資訊。 ElastiCache
4. 讀取 ElastiCache 資料 Lambda 函數擷取風險結果， ElastiCache 並將其存放在資料庫和 S3 儲存貯體中。

如需這些服務和步驟的詳細資訊，請參閱 Epics 一節。

自動化和規模

您可以使用 AWS Cloud Development Kit (AWS CDK) 或 AWS CloudFormation 範本來部署整個架構。該架構可以同時支持批處理和日內（實時）處理。

擴展是內置在架構中。隨著越來越多的交易寫入 Kinesis 資料串流並等待處理，您可以叫用其他 Lambda 函數來處理這些交易，然後在處理完成後縮減規模。您也可以選擇透過多個 Amazon SQS 風險計算佇列進行處理。如果佇列之間需要嚴格的排序或合併，則無法平行處理。但是，對於 end-of-the-day 批次或小型日內批次，Lambda 函數可以 parallel 處理並將最終結果儲存在中 ElastiCache。

工具

AWS 服務

- Amazon Aurora 與 MySQL 相容版本是完全受管、與 MySQL 相容的關聯式資料庫引擎，可協助您設定、操作和擴展 MySQL 部署。這種模式使用 MySQL 作為一個例子，但你可以使用任何 RDBMS 系統來存儲數據。
- Amazon 可 ElastiCache協助您在 AWS 雲端中設定、管理和擴展分散式記憶體內快取環境。
- Amazon Kinesis Data Streams 可協助您即時收集和處理大量資料記錄串流。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

此模式為 AWS 雲端中的 VaR 系統提供範例架構，並說明如何使用 Lambda 函數進行 VaR 計算。若要建立 Lambda 函數，請參閱 L ambda 文件中的程式碼範例。如需協助，請聯絡 AWS Professional Services。

最佳實務

- 讓每個 VaR 運算工作盡可能小且輕量化。在每個計算任務中嘗試不同數量的交易，以查看哪個交易最適合計算時間和成本。

- 將可重複使用的物件存放在 Amazon ElastiCache 使用 Apache 箭頭之類的框架來減少序列化和反序列化。
- 考慮拉姆達的時間限制。如果您認為運算工作可能超過 15 分鐘，請嘗試將它們分解為較小的工作，以避免 Lambda 逾時。如果這是不可能的，您可以考慮使用 AWS Fargate，Amazon 彈性容器服務（Amazon ECS）和亞馬遜彈性 Kubernetes 服務（亞馬遜 EKS）的容器協調解決方案。

史诗

交易流向風險系統

任務	描述	所需技能
開始寫交易。	新的，結算或部分結算的交易從訂單管理系統寫入風險流。此模式使用 Amazon Kinesis 做為受管串流服務。交易訂單股票代碼的哈希值用於在多個碎片中放置交易訂單。	Amazon Kinesis

執行 Lambda 函數進行訂單處理

任務	描述	所需技能
使用 Lambda 開始風險處理。	針對新訂單執行 AWS Lambda 函數。Lambda 將根據待處理的交易訂單數量自動擴展。每個 Lambda 執行個體都有一或多個訂單，並從 Amazon ElastiCache 擷取每個股票代碼的最新倉位。(您可以使用 CUSIP ID、曲線名稱或其他金融衍生產品的索引名稱作為儲存和擷取資料的金鑰 ElasticCache。) 在中 ElastiCache，總頭寸（數量）和鍵值對 < 股票代碼，淨頭寸 > (其中淨	Amazon Kinesis，AWS Lambda，Amazon ElastiCache

任務	描述	所需技能
	頭寸是縮放因子）會對每個股票更新一次。	

將每個股票代碼的消息寫入隊列

任務	描述	所需技能
將合併的訊息寫入風險佇列。	將訊息寫入佇列。此模式使用 Amazon SQS 做為受管佇列服務。單一 Lambda 執行個體可能會在任何給定時間收到一個小批次的交易訂單，但只會為每個股票代碼寫入一則訊息給 Amazon SQS。計算比例係數：（舊淨位置 + 當前位置）/舊淨位置。	Amazon SQS，AWS Lambda

呼叫風險引擎

任務	描述	所需技能
開始風險計算。	會叫用風險引擎 lambda 的 Lambda 函數。每個位置都由單個 Lambda 函數處理。不過，基於最佳化目的，每個 Lambda 函數都可以處理來自 Amazon SQS 的多則訊息。	Amazon SQS，AWS Lambda

從快取擷取風險結果

任務	描述	所需技能
擷取和更新風險快取。	Lambda 從 ElastiCache中檢索每個股票代碼的當前淨頭寸。它還從中檢索每個股票代碼的 VaR 損益（PnL）數組。ElastiCache 如果 PnL 陣列已經存在，Lambda 函數會使用擴展來更新陣列和 VaR，這是來自網絡 Lambda 函數所撰寫的 Amazon SQS 訊息。如果盈虧陣列不在 ElasticCache，則會使用模擬股票價格系列資料來計算新的盈虧和 VaR。	Amazon SQS，AWS Lambda，Amazon ElastiCache

更新彈性緩存中的數據並存儲在數據庫中

任務	描述	所需技能
存儲風險結果。	在中更新 VaR 和盈虧數字之後 ElastiCache，每五分鐘就會叫用一個新的 Lambda 函數。此函數會從讀取所有儲存的資料， ElastiCache 並將其存放在 Aurora 與 MySQL 相容的資料庫和 S3 儲存貯體中。	AWS Lambda，Amazon ElastiCache

相關資源

- 巴塞爾風值框架

將太數據標準化時間功能轉換為 Amazon Redshift SQL

資料來源：泰瑞資料倉儲	目標：Amazon Redshift	R 型：重新建築
環境：生產	技術：分析；資料庫；移轉	工作負載：所有其他工作
AWS 服務：Amazon Redshift		

Summary

規範化是 ANSI SQL 標準的太數據擴展。當 SQL 表格包含具有「期間」資料類型的資料行時，NORMALIZE 會合併該欄中符合或重疊的值，以形成合併多個個別週期值的單一期間。若要使用標準化，SQL SELECT 清單中至少有一個資料行必須是 Teradata 的暫時週期資料類型。如需有關規範化的詳細資訊，請參閱 Teradata 文件。

Amazon Redshift 不支援正常化，但您可以使用原生 SQL 語法和 Amazon Redshift 中的 LAG 視窗函數來實作此功能。這種模式著重於使用 Teradata 規範化擴展與「開符合或重疊」條件，這是最流行的格式。本文說明此功能在 Teradata 中的運作方式，以及如何將其轉換為 Amazon Redshift 原生 SQL 語法。

先決條件和限制

先決條件

- 基本的 SQL 知識和經驗
- Amazon Redshift 知識和經驗

架構

源, 技術, 堆棧

- 太级数据仓储

目標技術堆疊

- Amazon Redshift

目標架構

如需將 Teradata 資料庫遷移到 Amazon Redshift 的高階架構，請參閱使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 的模式。遷移不會自動將太數據標準化短語轉換為 Amazon Redshift SQL。您可以按照此模式中的準則轉換此 Teradata 擴展。

工具

Code

為了說明正常化的概念和功能，請考慮 Teradata 中的下列表格定義：

```
CREATE TABLE systest.project
     (    emp_id        INTEGER,
          project_name  VARCHAR(20),
          dept_id       INTEGER,
          duration      PERIOD(DATE)
     );
```

執行下列 SQL 程式碼，將範例資料插入資料表中：

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000,  PERIOD(DATE '2010-01-10',
 DATE '2010-03-20') );
INSERT INTO systest.project VALUES (10, 'First Phase', 2000,  PERIOD(DATE '2010-03-20',
 DATE '2010-07-15') );

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000,  PERIOD(DATE
 '2010-06-15', DATE '2010-08-18') );
INSERT INTO systest.project VALUES (20, 'First Phase', 2000,  PERIOD(DATE '2010-03-10',
 DATE '2010-07-20') );

INSERT INTO systest.project VALUES (20, 'Second Phase', 1000,  PERIOD(DATE
 '2020-05-10', DATE '2020-09-20') );

END TRANSACTION;
```

結果：

```
select * from systest.project order by 1,2,3;
```

```
 *** Query completed. 4 rows found. 4 columns returned.
 *** Total elapsed time was 1 second.

     emp_id  project_name              dept_id  duration
-----------  --------------------  -----------  ------------------------
         10  First Phase                  1000  ('10/01/10', '10/03/20')
         10  First Phase                  2000  ('10/03/20', '10/07/15')
         10  Second Phase                 2000  ('10/06/15', '10/08/18')
         20  First Phase                  2000  ('10/03/10', '10/07/20')
         20  Second Phase                 1000  ('20/05/10', '20/09/20')
```

太數據標準化用例

現在，將太元數據標準化 SQL 子句添加到 SELECT 語句中：

```
SELECT NORMALIZE ON MEETS OR OVERLAPS emp_id, duration
FROM systest.project
ORDER BY 1,2;
```

這個規範化操作是在一個單一的列（emp_id）上執行。對於 emp_id=10，持續時間中的三個重疊期間值會合併為單一週期值，如下所示：

```
     emp_id  duration
-----------  ------------------------
         10  ('10/01/10', '10/08/18')
         20  ('10/03/10', '10/07/20')
         20  ('20/05/10', '20/09/20')
```

下面的 SELE CT 語句執行對項目名稱和 DEPT_ID 標準化操作。請注意，SELE CT 列表只包含一個期間列，持續時間。

```
SELECT NORMALIZE project_name, dept_id, duration
FROM systest.project;
```

輸出：

```
project_name              dept_id  duration
--------------------  -----------  ------------------------
First Phase                  1000  ('10/01/10', '10/03/20')
```

```
Second Phase                 1000  ('20/05/10', '20/09/20')
First Phase                  2000  ('10/03/10', '10/07/20')
Second Phase                 2000  ('10/06/15', '10/08/18')
```

Amazon Redshift 等效 SQL

Amazon Redshift 目前不支持表中的期間數據類型。相反地，您需要將 Teradata 週期資料欄位分成兩部分：起始日期、結束日期，如下所示：

```
CREATE TABLE systest.project
     (    emp_id        INTEGER,
          project_name  VARCHAR(20),
          dept_id       INTEGER,
          start_date  DATE,
          end_date    DATE
     );
```

將示例數據插入表中：

```
BEGIN TRANSACTION;

INSERT INTO systest.project VALUES (10, 'First Phase', 1000,  DATE '2010-01-10', DATE
 '2010-03-20' );
INSERT INTO systest.project VALUES (10, 'First Phase', 2000,  DATE '2010-03-20', DATE
 '2010-07-15');

INSERT INTO systest.project VALUES (10, 'Second Phase', 2000,  DATE '2010-06-15', DATE
 '2010-08-18' );
INSERT INTO systest.project VALUES (20, 'First Phase', 2000,  DATE '2010-03-10', DATE
 '2010-07-20' );

INSERT INTO systest.project VALUES (20, 'Second Phase', 1000,  DATE '2020-05-10', DATE
 '2020-09-20' );

END TRANSACTION;
```

輸出：

```
 emp_id | project_name | dept_id | start_date |  end_date
--------+--------------+---------+------------+------------
     10 | First Phase  |    1000 | 2010-01-10 | 2010-03-20
     10 | First Phase  |    2000 | 2010-03-20 | 2010-07-15
```

```
     10 | Second Phase |    2000 | 2010-06-15 | 2010-08-18
     20 | First Phase  |    2000 | 2010-03-10 | 2010-07-20
     20 | Second Phase |    1000 | 2020-05-10 | 2020-09-20
(5 rows)
```

要重寫太元數據的正常化子句，您可以在 Amazon Redshift 使用 LAG 窗口函數。此函數會傳回位於分割區中目前資料列上方 (之前) 指定偏移處的資料列值。

您可以使用 LAG 函數來識別開始新期間的每個資料列，方法是判斷期間是否符合或與先前期間重疊 (若為 0，若否則為 1)。當此旗標累計總結時，它會提供一個群組識別碼，可在外部 Group By 子句中使用，以便在 Amazon Redshift 中達到所需的結果。

以下是使用 LAG () 的 Amazon Redshift SQL 陳述式範例：

```
SELECT emp_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
 start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project
ORDER BY 1,2;
```

輸出：

```
 emp_id | start_date |  end_date  | groupstartflag
--------+------------+------------+----------------
     10 | 2010-01-10 | 2010-03-20 |              1
     10 | 2010-03-20 | 2010-07-15 |              0
     10 | 2010-06-15 | 2010-08-18 |              0
     20 | 2010-03-10 | 2010-07-20 |              1
     20 | 2020-05-10 | 2020-09-20 |              1
(5 rows)
```

以下 Amazon Redshift SQL 語句僅在 emp_id 列上標準化：

```
SELECT T2.emp_id, MIN(T2.start_date) as new_start_date, MAX(T2.end_date) as
 new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY emp_id ORDER BY start_date ROWS
 UNBOUNDED PRECEDING) As GroupID
FROM ( SELECT emp_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY emp_id ORDER BY
 start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
```

```
FROM systest.project ) T1
) T2
GROUP BY T2.emp_id, T2.GroupID
ORDER BY 1,2;
```

輸出：

```
 emp_id | new_start_date | new_end_date
--------+----------------+-------------------------------------
     10 | 2010-01-10     | 2010-08-18
     20 | 2010-03-10     | 2010-07-20
     20 | 2020-05-10     | 2020-09-20
(3 rows)
```

以下 Amazon Redshift SQL 語句在項目名稱和 dept_id 列上正常化：

```
SELECT T2.project_name, T2.dept_id, MIN(T2.start_date) as new_start_date,
 MAX(T2.end_date) as new_end_date
FROM
( SELECT T1.*, SUM(GroupStartFlag) OVER (PARTITION BY project_name, dept_id ORDER BY
 start_date ROWS UNBOUNDED PRECEDING) As GroupID
FROM ( SELECT project_name, dept_id, start_date, end_date,
            (CASE WHEN start_date <= LAG(end_date) OVER (PARTITION BY project_name,
 dept_id ORDER BY start_date, end_date) THEN 0 ELSE 1 END) AS GroupStartFlag
FROM systest.project ) T1
) T2
GROUP BY T2.project_name, T2.dept_id, T2.GroupID
ORDER BY 1,2,3;
```

輸出：

```
 project_name | dept_id | new_start_date | new_end_date
--------------+---------+----------------+--------------
 First Phase  |    1000 | 2010-01-10     | 2010-03-20
 First Phase  |    2000 | 2010-03-10     | 2010-07-20
 Second Phase |    1000 | 2020-05-10     | 2020-09-20
 Second Phase |    2000 | 2010-06-15     | 2010-08-18
(4 rows)
```

史诗

將標準化轉換為 Amazon Redshift SQL

任務	描述	所需技能
建立您的 SQL 程式碼。	根據您的需要使用正常化短語。	SQL Developer
將代碼轉換為 Amazon Redshift SQL。	要轉換您的代碼，請遵循此模式的「工具」部分中的準則。	SQL Developer
在 Amazon Redshift 中運行代碼。	在 Amazon Redshift 中建立您的資料表、將資料載入資料表，然後在資料表中執行程式碼。	SQL Developer

相關資源

參考

- 太級數據標準化時間功能（Teradata 文檔）
- LAG 視窗功能 (Amazon Redshift 文件)
- 遷移到 Amazon Redshift（AWS 網站）
- 使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 移 (AWS Prescriptive Guidance)
- 將功能重設時間轉換為 Amazon Redshift SQL (AWS Prescriptive Guidance)

工具

- AWS Schema Conversion Tool

合作夥伴

- AWS 遷移能力合作夥伴

將太數據重置功能轉換為 Amazon Redshift SQL

資料來源：泰瑞資料倉儲	目標：Amazon Redshift	R 型：重新建築
環境：生產	技術：分析；資料庫；移轉	工作負載：所有其他工作
AWS 服務：Amazon Redshift		

Summary

重置時間是 SQL 分析窗口函數中使用的 Teradata 功能。它是一個擴展到 ANSI SQL 標準。RESET WHERE 決定 SQL 視窗函式根據某些指定條件來運作的分割區。如果條件評估為 TRUE，則在現有窗口分區內創建一個新的動態子分區。如需有關重設時機的詳細資訊，請參閱 Teradata 文件。

Amazon Redshift 不支持 SQL 窗口函數中的重置時間。若要實作此功能，您必須在 Amazon Redshift 中將重設時機轉換為原生 SQL 語法，並使用多個巢狀函數。此模式示範如何使用 Teradata 重設時機功能，以及如何將其轉換為 Amazon Redshift SQL 語法。

先決條件和限制

先決條件

- Teradata 資料倉儲及其 SQL 語法的基本知識
- 對 Amazon Redshift 及其 SQL 語法有很好的理解

架構

源, 技術, 堆棧

- 太级数据仓储

目標技術堆疊

- Amazon Redshift

架構

如需將 Teradata 資料庫遷移到 Amazon Redshift 的高階架構，請參閱使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移至 Amazon Redshift 移的模式。遷移不會自動將太數據重置時短語轉換為 Amazon Redshift SQL。您可以按照下一節中的準則轉換此 Teradata 擴展。

工具

Code

若要說明「重設時機」的概念，請在 Teradata 中考慮下列資料表定義：

```
create table systest.f_account_balance
( account_id integer NOT NULL,
  month_id integer,
  balance integer )
unique primary index (account_id, month_id);
```

執行下列 SQL 程式碼，將範例資料插入資料表：

```
BEGIN TRANSACTION;
Insert Into systest.f_account_balance values (1,1,60);
Insert Into systest.f_account_balance values (1,2,99);
Insert Into systest.f_account_balance values (1,3,94);
Insert Into systest.f_account_balance values (1,4,90);
Insert Into systest.f_account_balance values (1,5,80);
Insert Into systest.f_account_balance values (1,6,88);
Insert Into systest.f_account_balance values (1,7,90);
Insert Into systest.f_account_balance values (1,8,92);
Insert Into systest.f_account_balance values (1,9,10);
Insert Into systest.f_account_balance values (1,10,60);
Insert Into systest.f_account_balance values (1,11,80);
Insert Into systest.f_account_balance values (1,12,10);
END TRANSACTION;
```

範例資料表包含下列資料：

account_id	月份	平衡
1	1	60
1	2	99

1	3	94
1	4	90
1	5	80
1	6	88
1	7	90
1	8	92
1	9	10
1	10	60
1	11	80
1	12	10

對於每個帳戶，假設您要分析連續每月餘額增加的順序。當一個月的餘額小於或等於上個月的餘額時，需求是將計數器重置為零並重新啟動。

使用案例時重置

為了分析此資料，Teradata SQL 會使用具有巢狀彙總的視窗函數和「重設時機」片語，如下所示：

```
SELECT account_id, month_id, balance,
 ( ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY month_id
RESET WHEN balance <= SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS
 BETWEEN 1 PRECEDING AND 1 PRECEDING) ) -1 ) as balance_increase
FROM systest.f_account_balance
ORDER BY 1,2;
```

輸出：

account_id	月份	平衡	餘額 _ 增加
1	1	60	0
1	2	99	1

1	3	94	0
1	4	90	0
1	5	80	0
1	6	88	1
1	7	90	2
1	8	92	3
1	9	10	0
1	10	60	1
1	11	80	2
1	12	10	0

查詢在 Teradata 中的處理方式如下：

1. SUM（餘額）彙總函數計算給定月份中給定帳戶的所有餘額的總和。
2. 我們會檢查指定月份 (針對特定帳戶) 的餘額是否大於上個月的餘額。
3. 如果餘額增加，我們跟踪累積計數值。如果 RESET WH EN 條件評估為 false，這意味著餘額在連續幾個月內增加，我們將繼續增加計數。
4. ROW_NUMBER () 有序的解析函數計算計數值。當我們到達一個月的餘額小於或等於上個月的餘額時，RESET WHEN 條件評估為 true。如果是這樣，我們啟動一個新的分區，ROW_NUMBER () 從 1 重新啟動計數。我們使用前 1 和 1 之間的行來訪問前一行的值。
5. 我們減去 1 以確保計數值從 0 開始。

Amazon Redshift 等效 SQL

Amazon Redshift 不支持 SQL 分析窗口函數中的重置時間短語。 若要產生相同的結果，您必須使用 Amazon Redshift 原生 SQL 語法和巢狀子查詢來重新撰寫 Teradata SQL，如下所示：

```
SELECT account_id, month_id, balance,
   (ROW_NUMBER() OVER(PARTITION BY account_id, new_dynamic_part ORDER BY month_id) -1)
 as balance_increase
```

```
FROM
( SELECT account_id, month_id, balance, prev_balance,
SUM(dynamic_part) OVER (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN
 UNBOUNDED PRECEDING AND CURRENT ROW) As new_dynamic_part
FROM ( SELECT account_id, month_id, balance,
SUM(balance) over (PARTITION BY account_id ORDER BY month_id ROWS BETWEEN 1 PRECEDING
 AND 1 PRECEDING) as prev_balance,
(CASE When balance <= prev_balance Then 1 Else 0 END) as dynamic_part
FROM systest.f_account_balance ) A
) B
ORDER BY 1,2;
```

由於 Amazon Redshift 在單一 SQL 陳述式的 SELE CT 子句中不支援巢狀視窗函數，因此您必須使用兩個巢狀子查詢。

- 在內部子查詢 (別名 A) 中，會建立並填入動態磁碟分割指示器 (dynamic_part)。 如果一個月的餘額小於或等於上個月的餘額，dynamic_part 會設定為 1；否則，它會設定為 0。
- 在下一層（別名 B）中，一個新的動態 _ 部分屬性作為一個 SUM 窗口函數的結果生成。
- 最後，您將 new_dynamic_part 作為新的分割區屬性 (動態磁碟分割) 新增至現有的分割區屬性 (account_id)，並套用與 Teradata 中相同的 ROW_NUMBER () 視窗函數 (和減去一)。

在這些變更之後，Amazon Redshift SQL 產生與太數據相同的輸出。

史诗

將重置時間轉換為 Amazon Redshift SQL

任務	描述	所需技能
建立您的視窗功能。	根據您的需要使用嵌套聚合和 RESET 時短語。	SQL Developer
將代碼轉換為 Amazon Redshift SQL。	要轉換您的代碼，請遵循此模式的「工具」部分中的準則。	SQL Developer
在 Amazon Redshift 中運行代碼。	在 Amazon Redshift 中建立您的資料表、將資料載入資料表，然後在資料表中執行程式碼。	SQL Developer

相關資源

參考

- 當短語重置（太數據文檔）
- 解釋時重置（堆棧溢出）
- 遷移到 Amazon Redshift（AWS 網站）
- 使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 移 (AWS Prescriptive Guidance)
- 將太數據標準化時間功能轉換為 Amazon Redshift SQL (AWS Prescriptive Guidance)

工具

- AWS Schema Conversion Tool

合作夥伴

- AWS 遷移能力合作夥伴

啟動時強制標記 Amazon EMR 叢集

創建者：普里揚卡喬達瑞 (AWS)

環境：生產	技術：分析；安全性、身分識別、合規性	AWS 服務：Amazon EMR；AWS Lambda；Amazon CloudWatch 活動

Summary

此模式提供安全控制，可確保 Amazon EMR 叢集在建立時加上標記。

Amazon EMR 是一種 Amazon Web Services（AWS）服務，用於處理和分析大量數據。Amazon EMR 提供可擴充、低組態的服務，是執行內部叢集運算的更輕鬆替代方案。您可以使用標記以不同的方式分類 AWS 資源，例如按用途、擁有者或環境。例如，您可以透過將自訂中繼資料指派給每個叢集來標記 Amazon EMR 叢集。標籤由您定義的鍵和值組成。我們建議您建立一組一致的標籤，以符合組織的需求。當您將標籤新增至 Amazon EMR 叢集時，標籤也會傳播到與叢集關聯的每個使用中 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。同樣地，當您從 Amazon EMR 叢集移除標籤時，該標籤也會從每個關聯的作用中 EC2 執行個體中移除。

偵測控制項會監控 API 呼叫，並針對RunJobFlowAddTags、RemoveTags和 CreateTagsAPI 啟 CloudWatch 動 Amazon 事件事件。該事件調用 AWS Lambda，該腳本運行一個 Python 腳本。Python 函數會從事件的 JSON 輸入取得 Amazon EMR 叢集識別碼，並執行下列檢查：

- 檢查 Amazon EMR 叢集是否使用您指定的標籤名稱進行設定。
- 如果沒有，請傳送 Amazon 簡單通知服務 (Amazon SNS) 通知給使用者，其中包含相關資訊：此通知來源於 Lambda 的 Amazon EMR 叢集名稱、違規詳細資訊、AWS 區域、AWS 帳戶和亞馬遜資源名稱 (ARN)。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一個 Amazon Simple Storage Service (Amazon S3) 存儲桶，用於上傳提供的 Lambda 代碼。或者，您可以為此目的建立 S3 儲存貯體，如 Epics 一節所述。

- 您希望接收違規通知的作用中電子郵件地址。
- 您要檢查的強制性標籤列表。

限制

- 此安全控制是區域性的。您必須在要監控的每個 AWS 區域中部署它。

產品版本

- Amazon EMR 版本 4.8.0 及更高版本。

架構

工作流架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- Amazon CloudWatch 活動-Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- Amazon EMR-Amazon EMR 是一種網路服務，可簡化大數據架構的執行，並有效率地處理大量資料。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。

- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

此模式包括下列附件：

- `EMRTagValidation.zip`— 安全控制的 Lambda 程式碼。
- `EMRTagValidation.yml`— 設定事件和 Lambda 函數的 CloudFormation 範本。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立 S3 儲存貯體來託管 Lambda 程式碼 .zip 檔案。此 S3 儲存貯體必須與您要監控的 Amazon EMR 叢集位於相同的 AWS 區域。Amazon S3 儲存貯體的名稱必須是全域唯一，且命名空間會由所有 AWS 帳戶共享。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 .zip 檔案上傳至 S3 儲存貯體。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
啟動 AWS CloudFormation 範本。	在與 S3 儲存貯體相同的 CloudFormation AWS 區域中開啟 AWS 主控台，然後部署範本。如需部署 AWS CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的在 AWS CloudFormation 主控台建立堆疊。	雲端架構師
完成範本中的參數。	當您啟動範本時，系統會提示您輸入下列資訊： • S3 儲存貯體：指定您在第一個史詩中建立或選取的儲存貯體。這是您上傳附加的 Lambda 程式碼 (.zip 檔案) 的地方。 • S3 金鑰：指定 S3 儲存貯體中 Lambda .zip 檔案的位置 (例如檔案名稱 .zip 或控制項/檔案名稱 .zip)。請勿包含前導斜線。 • 通知電子郵件：提供您要接收 Amazon SNS 通知的作用中電子郵件地址。 • 標記金鑰名稱：在逗號分隔清單中提供您要檢查的標籤 (例如，`ApplicationID` 、`Environment` 、`Owner`)。E CloudWatch vents 事件會監視叢集中的這些標記，並在找不到時傳送通知。	雲端架構師

任務	描述	所需技能
	• Lambda 記錄層級：指定 Lambda 函數的記錄層級和頻率。使用「資訊」(Info) 可記錄進度的詳細資訊訊息、針對仍允許部署繼續的錯誤事件發生錯誤，以及針對潛在有害情況發出警告。	

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱，才能開始接收違規通知。	雲端架構師

相關資源

- AWS Lambda 開發人員指南
- 在 Amazon EMR 中標記叢集

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確保啟動時已啟用 Amazon S3 的亞馬遜 EMR 記錄功能

環境：生產	技術：安全性、身分識別、合規性、無伺服器、分析	工作負載：開源
AWS 服務：Amazon EMR; Amazon S3; Amazon SNS; Amazon CloudWatch		

Summary

此模式提供安全控制，用於監控 Amazon 網路服務 (AWS) 上執行之 Amazon EMR 叢集的記錄組態。

Amazon EMR 是用於大數據處理和分析的 AWS 工具。Amazon EMR 提供可擴充的低組態服務，作為執行內部叢集運算的替代方案。Amazon EMR 提供兩種類型的 EMR 叢集。

- 暫時性 Amazon EMR 叢集：暫時性 Amazon EMR 叢集會在處理完成時自動關閉並停止產生成本。
- 持續性 Amazon EMR 叢集：持續性 Amazon EMR 叢集會在資料處理任務完成後繼續執行。

Amazon EMR 和 Hadoop 都會產生報告叢集狀態的日誌檔案。根據預設，這些檔案會寫入 /mnt/var/log/ 目錄中的主節點。根據啟動叢集時的配置方式，您也可以將這些日誌儲存到 Amazon Simple Storage Service (Amazon S3)，並透過圖形化偵錯工具檢視它們。請注意，只有在叢集啟動時才能指定 Amazon S3 日誌記錄。使用此組態時，每 5 分鐘會將日誌從主節點傳送到 Amazon S3 位置。對於暫時性叢集而言，Amazon S3 記錄非常重要，因為叢集在處理完成時會消失，而且這些日誌檔可用於偵錯任何失敗的任務。

該模式使用 AWS CloudFormation 範本部署安全控制，以監控 API 呼叫並在「RunJobFlow」上啟 CloudWatch 動 Amazon 事件。觸發程序會叫用 AWS Lambda，它會執行 Python 指令碼。Lambda 函數會從事件 JSON 輸入擷取 EMR 叢集識別碼，並檢查 Amazon S3 日誌 URI。如果找不到 Amazon S3 URI，Lambda 函數會傳送亞馬遜簡單通知服務 (Amazon SNS) 通知，詳細說明該通知來源於 EMR 叢集名稱、違規詳細資訊、AWS 區域、AWS 帳戶以及 Lambda Amazon 資源名稱 (ARN)。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 用於 Lambda 代碼 .zip 文件的 S3 存儲桶
- 您想要接收違規通知的電子郵件地址

限制

- 此偵探控制是區域性的，必須部署在您要監控的 AWS 區域中。

產品版本

- Amazon EMR 版本 4.8.0 及更新版本

架構

目標技術堆疊

- Amazon CloudWatch 活動事件
- Amazon EMR
- Lambda 函數
- S3 儲存貯體
- Amazon SNS

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS CloudFormation StackSets 在要監控的多個帳戶中部署此範本。

工具

工具

- AWS CloudFormation — AWS 可 CloudFormation 協助您使用基礎設施即程式碼來建立 AWS 資源的模型和設定。
- AWS Cloudwatch 活動 — AWS CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- Amazon EMR — Amazon EMR 是一個受管叢集平台，可簡化大數據架構的執行作業。
- AWS Lambda — AWS Lambda 支援執行程式碼，無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — Amazon S3 是一個 Web 服務界面，可用來存放和擷取任意數量的資料，從 Web 上的任何位置。
- Amazon SNS — Amazon SNS 是一種 Web 服務，可協調和管理發佈者與用戶端之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。

Code

- 專案的 .zip 檔案可作為附件使用。

史诗

定義 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	若要託管 Lambda 程式碼 .zip 檔案，請選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用命名空間。您的 S3 儲存貯體必須與正在評估的 Amazon EMR 叢集位於相同的 AWS 區域。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」一節中提供的 Lambda 程式碼 .zip 檔案上傳至 S3 儲存貯體。S3 儲存貯體必須與正在評估的 Amazon EMR 叢集位於相同的區域。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
部署 AWS CloudFormation 範本。	在 AWS CloudFormation 主控台與 S3 儲存貯體相同的區域中，將以附件形式提供的 AWS CloudFormation 範本部署到此模式。在下一個史詩中，提供參數的值。如需部署 AWS CloudFormation 範本的詳細資訊，請參閱「相關資源」一節。	雲端架構師

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 Amazon S3 密鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如<directory>/<file-name>.zip)。	雲端架構師

任務	描述	所需技能
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。「信息」指定了有關應用程序進度的詳細信息消息。「Error」指定仍可允許應用程式繼續執行的錯誤事件。「警告」表示可能有害的情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件訊息傳送至提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

AWS Lambda

Amazon EMR 記錄

部署 AWS CloudFormation 範本

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS AWS Glue 任務和 Python 產生測試資料

環境：生產	技術：分析；雲端原生；資料湖；軟體開發與測試；無伺服器；大數據	AWS 服務：AWS AWS Glue；Amazon S3

Summary

此模式說明如何透過建立以 Python 撰寫的 AWS Glue 任務，快速輕鬆地同時產生數百萬個範例檔案。範例檔案存放在亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體中。快速產生大量範例檔案的能力對於在 AWS 雲端測試或評估服務而言非常重要。例如，您可以對 Amazon S3 前置詞中的數百萬個小檔案執行資料分析，以測試 AWS Glue DataBrew 工作室或 AWS Glue 任務的效能。

雖然您可以使用其他 AWS 服務來產生範例資料集，但我們建議您使用 AWS Glue。您不需要管理任何基礎設施，因為 AWS Glue 是無伺服器資料處理服務。您只需攜帶程式碼，然後在 AWS Glue 叢集中執行即可。此外，AWS Glue 還可佈建、設定和擴展執行任務所需的資源。您只需為工作在執行時使用的資源付費。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI)，已安裝並設定為與 AWS 帳戶搭配使用

產品版本

- Python 3.9
- AWS CLI 第 2 版

限制

每個觸發的 AWS Glue 任務數目上限為 50 個。如需詳細資訊，請參閱 AWS Glue 端點和配額。

架構

下圖說明以 AWS Glue 任務為中心的範例架構，該任務會將其輸出 (亦即範例檔案) 寫入 S3 儲存貯體。

圖表包括下列工作流程：

1. 您可以使用 AWS CLI、AWS 管理主控台或 API 來啟動 AWS Glue 任務。AWS CLI 或 API 可讓您自動執行叫用任務的平行化，並縮短產生範例檔案的執行時間。
2. AWS Glue 任務會隨機產生檔案內容，將內容轉換為 CSV 格式，然後將內容以 Amazon S3 物件形式存放在通用前綴下。每個文件小於一千字節。AWS Glue 任務接受兩個使用者定義的任務參數：START_RANGE和END_RANGE。您可以使用這些參數來設定檔案名稱，以及每次執行任務在 Amazon S3 中產生的檔案數目。您可以 parallel 執行此工作的多個執行個體 (例如 100 個執行個體)。

工具

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

最佳實務

實作此模式時，請考慮下列 AWS Glue 最佳實務：

- 使用正確的 AWS Glue 工作者類型來降低成本。建議您瞭解 Worker 類型的不同內容，然後根據 CPU 和記憶體需求選擇適合您工作負載的 Worker 類型。對於這種模式，我們建議您使用 Python 殼層工作作為工作類型，以最小化 DPU 並降低成本。如需詳細資訊，請參閱 AWS Glue 開發人員指南中的在 AWS Glue 中新增任務。

- 使用正確的並行限制來擴展您的工作。我們建議您根據您的時間需求和所需的檔案數量，根據 AWS Glue 任務的最大並行處理。
- 首先開始生成少量文件。若要在建立 AWS Glue 任務時降低成本並節省時間，請從少量檔案開始 (例如 1,000 個)。這可以使故障排除更容易。如果成功產生少量檔案，則可以縮放至更多檔案。
- 首先在本地運行。若要在建立 AWS Glue 任務時降低成本並節省時間，請在本機啟動開發並測試程式碼。如需設定可協助您在殼層和整合式開發環境 (IDE) 中撰寫 AWS Glue 擷取、轉換和載入 (ETL) 任務的 Docker 容器的指示，請參閱 AWS 大數據部落格上使用容器貼文在本機開發 AWS Glue ETL 任務。

如需更多 AWS Glue 最佳實務，請參閱 AWS Glue 文件中的最佳實務。

史诗

建立目的地 S3 儲存貯體和 IAM 角色

任務	描述	所需技能
建立用於存放檔案的 S3 儲存貯體。	建立 S3 儲存貯體及其中的前置詞。 注意：此模式將`s3://{your-s3-bucket-name}/small-files/` 位置用於演示目的。	應用程式開發人員
建立和設定 IAM 角色。	您必須建立 AWS Glue 任務可用來寫入 S3 儲存貯體的 IAM 角色。 1. 建立 IAM 角色 (例如，呼叫`"AWSGlueServiceRole-smallfiles"`)。 2. 選擇 AWS Glue 作為政策的信任實體。 3. 將呼叫的 AWS 受管政策附加`"AWSGlueServiceRole"` 至該角色。	應用程式開發人員

任務	描述	所需技能
	4. 建立"s3-small-file-access" 根據下列組態呼叫的內嵌政策或客戶管理政策。請"{bucket}" 以儲存貯體名稱取代。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "s3:GetObject", "s3:PutObject" ], "Resource": [ "arn:aws:s3:::{bucket}/small-files/input/*" ] } ] }` 5. 將"s3-small-file-access" 原則附加至您的角色。	

建立和設定 AWS Glue 任務以處理並行執行

任務	描述	所需技能
建立 AWS AWS Glue 任務。	您必須建立可產生內容並將其存放在 S3 儲存貯體的 AWS Glue 任務。 建立 AWS Glue 任務，然後完成以下步驟來設定任務： 1. 登入 AWS 管理主控台並開啟 AWS Glue 主控台。 2. 在導覽窗格的 [資料整合和 ETL] 下，選擇 [工作]。 3. 在「建立工作」區段中，選擇「Python 殼層指令碼編輯器」。 4. 在 [選項] 區段中，選取 [使用樣板程式碼建立新指令碼]，然後選擇 [建立]。 5. 選擇 [Job 明細]。 6. 在「名稱」中，輸入建立小型檔案。 7. 對於 IAM 角色，請選取您先前建立的 IAM 角色。 8. 在「此工作執行」段落中，選擇您要編寫的新命令檔。 9. 展開進階屬性。 10對於「最大並行」，請輸入 100 以進行示範。注意：最大並行性會定義您可以 parallel 執行的工作執行處理數目。 11選擇儲存。	應用程式開發人員

任務	描述	所需技能
更新工作代碼。	1. 開啟 AWS AWS Glue 主控台。 2. 在導覽窗格中，選擇 Jobs (任務)。 3. 在「您的工作」區段中，選擇您先前建立的工作。 4. 選擇 [指令碼] 索引標籤，然後根據下列程式碼更新指令碼。使用BUCKET_NAME 您的值更新PREFIX、和text_str變數。 `from awsglue.utils import getResolvedOptions import sys import boto3 from random import randrange # Two arguments args = getResolvedOptions(sys.argv, ['START_RANGE', 'END_RANGE']) START_RANGE = int(args['START_RANGE']) END_RANGE = int(args['END_RANGE']) BUCKET_NAME = '{BUCKET_NAME}' PREFIX = 'small-files/input/' s3 = boto3.resource('s3')`	應用程式開發人員

任務	描述	所需技能
	`` for x in range(STA RT_RANGE, END_RANGE): # generate file name file_name = f"input_{x}.txt" # generate text text_str = str(randrange(1000 00))+","+str(randr ange(100000))+", " + str(randrange(1000 0000)) + "," + str(randrange(1000 0)) # write in s3 s3.Object(BUCKE T_NAME, PREFIX + file_name).put(Bod y=text_str) ``` 5. 選擇儲存。	

從命令列或主控台執行 AWS Glue 任務

任務	描述	所需技能
從命令列執行 AWS Glue 任務。	若要從 AWS CLI 執行 AWS Glue 任務，請使用您的值執行下列命令： ``` cmd:~$ aws glue start- job-run --job-name create_small_files --arguments '{"--STAR T_RANGE":"0","--EN D_RANGE":"1000000"}' ```	應用程式開發人員

任務	描述	所需技能
	cmd:~$ aws glue start-job-run --job-name create_small_files --arguments '{"--START_RANGE":"1000000","--END_RANGE":"2000000"}' 注意：如需從 AWS 管理主控台執行 AWS Glue 任務的相關說明，請參閱此模式中 AWS 管理主控台中的執行 AWS Glue 任務故事。 提示：如果您想要一次使用不同的參數執行多個執行，建議您使用 AWS CLI 執行 AWS Glue 任務，如上例所示。 若要產生使用特定平行化因子產生已定義檔案數目所需的所有 AWS CLI 命令，請執行下列 bash 程式碼 (使用您的值)： # define parameters NUMBER_OF_FILES=10000000; PARALLELIZATION=50; # initialize _SB=0; # generate commands for i in $(seq 1 $PARALLELIZATION); do echo aws glue start-job-run --job-name create_sm	

任務	描述	所需技能
	<pre>all_files --argumen ts "'"'{"--START_RANG E":"'$(((NUMBER_OF _FILES/PARALLELIZA TION) * (i-1) + _SB))'","--END_RAN GE":"'$(((NUMBER_O F_FILES/PARALLELIZ ATION) * (i)))'"}'"'"; _SB=1; done</pre> 如果您使用上述指令碼，請考慮下列事項： • 該腳本簡化了大規模小文件的調用和生成。 • 更新NUMBER_OF_FILES 並PARALLELIZATION 使用您的價值觀。 • 上面的腳本打印必須運行的命令列表。複製這些輸出命令，然後在終端中運行它們。 • 如果您想要直接從指令碼中執行命令，請移除第 11 行中的echo陳述式。 注意：若要查看上述指令碼輸出的範例，請參閱此模式的其他資訊一節中的 Shell 指令碼輸出。	

任務	描述	所需技能
在 AWS 管理主控台中執行 AWS Glue 任務。	1. 登入 AWS 管理主控台並開啟 AWS Glue 主控台。 2. 在導覽窗格的 [資料整合和 ETL] 下，選擇 [工作]。 3. 在「您的工作」區段中，選擇您的工作。 4. 在「參數 (選用)」區段中，更新您的參數。 5. 選擇 [動作]，然後選擇 [執行工作]。 6. 根據需要多次重複步驟 3-5。例如，要創建 1000 萬個文件，請重複此過程 10 次。	應用程式開發人員
檢查 AWS AWS Glue 任務的狀態。	1. 開啟 AWS AWS Glue 主控台。 2. 在導覽窗格中，選擇 Jobs (任務)。 3. 在 [您的工作] 區段中，選擇您先前建立的工作 (也就是`create_small_files`)。 4. 若要深入瞭解檔案的進度和產生，請檢閱執行 ID、執行狀態和其他欄。	應用程式開發人員

相關資源

參考

- AWS 上的開放資料登錄
- 用於分析的資料集

- AWS 上的開放資料
- 在 AWS AWS Glue 中新增任務
- 開始使用 AWS AWS Glue

指南和模式

- AWS AWS Glue 最佳實務
- 負載測試應用

其他資訊

基準測試

此模式用於使用不同的並行化參數生成 1000 萬個文件作為基準測試的一部分。下表顯示了測試的輸出：

平行化	工作執行所產生的檔案數目	Job 期間	Speed (速度)
10	1,000,000	六小時四十分鐘	很慢
50	200,000	八十分鐘	適中
100	100,000	40 分鐘	快速

如果您想要加快處理速度，可以在工作組態中設定更多並行執行。您可以根據自己的需求輕鬆調整任務組態，但請記住，AWS Glue 服務配額有限制。如需詳細資訊，請參閱 AWS Glue 端點和配額。

外殼腳本輸出

下列範例會顯示從此模式的命令列故事中執行 AWS Glue 任務的 shell 指令碼輸出。

```
user@MUC-1234567890 MINGW64 ~
  $ # define parameters
  NUMBER_OF_FILES=10000000;
  PARALLELIZATION=50;
  # initialize
  _SB=0;
```

```
  # generate commands
  for i in $(seq 1 $PARALLELIZATION);
   do
         echo aws glue start-job-run --job-name create_small_files --arguments
 "'"'{"--START_RANGE":"'$(((NUMBER_OF_FILES/PARALLELIZATION)  (i-1) + SB))'","--
ENDRANGE":"'$(((NUMBER_OF_FILES/PARALLELIZATION)  (i)))'"}'"'";
         _SB=1;
   done

  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"0","--END_RANGE":"200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"200001","--END_RANGE":"400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"400001","--END_RANGE":"600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"600001","--END_RANGE":"800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"800001","--END_RANGE":"1000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1000001","--END_RANGE":"1200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1200001","--END_RANGE":"1400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1400001","--END_RANGE":"1600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1600001","--END_RANGE":"1800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"1800001","--END_RANGE":"2000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2000001","--END_RANGE":"2200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2200001","--END_RANGE":"2400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2400001","--END_RANGE":"2600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2600001","--END_RANGE":"2800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"2800001","--END_RANGE":"3000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3000001","--END_RANGE":"3200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3200001","--END_RANGE":"3400000"}'
```

```
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3400001","--END_RANGE":"3600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3600001","--END_RANGE":"3800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"3800001","--END_RANGE":"4000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4000001","--END_RANGE":"4200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4200001","--END_RANGE":"4400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4400001","--END_RANGE":"4600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4600001","--END_RANGE":"4800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"4800001","--END_RANGE":"5000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5000001","--END_RANGE":"5200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5200001","--END_RANGE":"5400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5400001","--END_RANGE":"5600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5600001","--END_RANGE":"5800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"5800001","--END_RANGE":"6000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6000001","--END_RANGE":"6200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6200001","--END_RANGE":"6400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6400001","--END_RANGE":"6600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6600001","--END_RANGE":"6800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"6800001","--END_RANGE":"7000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7000001","--END_RANGE":"7200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7200001","--END_RANGE":"7400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7400001","--END_RANGE":"7600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7600001","--END_RANGE":"7800000"}'
```

```
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"7800001","--END_RANGE":"8000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8000001","--END_RANGE":"8200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8200001","--END_RANGE":"8400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8400001","--END_RANGE":"8600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8600001","--END_RANGE":"8800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"8800001","--END_RANGE":"9000000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9000001","--END_RANGE":"9200000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9200001","--END_RANGE":"9400000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9400001","--END_RANGE":"9600000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9600001","--END_RANGE":"9800000"}'
  aws glue start-job-run --job-name create_small_files --arguments '{"--
START_RANGE":"9800001","--END_RANGE":"10000000"}'

  user@MUC-1234567890 MINGW64 ~
```

常見問答集

我應該使用多少個並行運行或並行作業？

並行執行和 parallel 作業的數量取決於您的時間需求和所需的測試檔案數量。我們建議您檢查正在建立的檔案大小。首先，檢查 AWS Glue 任務需要多少時間來產生所需的檔案數量。然後，使用正確的並發運行數量來滿足您的目標。例如，如果您假設 100,000 個檔案需要 40 分鐘才能完成執行，但目標時間為 30 分鐘，則必須增加 AWS Glue 任務的並行設定。

我可以使用此模式創建什麼類型的內容？

您可以建立任何類型的內容，例如使用不同分隔符號 (例如 PIPE、JSON 或 CSV) 的文字檔案。此模式使用 Boto3 寫入檔案，然後將檔案儲存在 S3 儲存貯體中。

S3 儲存貯體需要什麼級別的 IAM 許可？

您必須具有以身分識別為基礎的政策，才能`Write`存取 S3 儲存貯體中的物件。如需詳細資訊，請參閱 Amazon S3 文件中的 Amazon S3：允許對 S3 儲存貯體中物件的讀取和寫入存取。

使用 Lambda 函數在暫態 EMR 叢集中啟動星火工作

創建者：德魯巴約提穆克吉 (AWS)

環境：生產	技術：分析	工作負載：開源
AWS 服務：Amazon EMR；AWS Identity and Access Management；AWS Lambda；Amazon VPC		

Summary

此模式使用 Amazon EMR RunJobFlow API 動作來啟動暫時性叢集，以便從 Lambda 函數執行星火任務。暫時性 EMR 叢集的設計目的是在工作完成或發生任何錯誤時立即終止。暫時性叢集可節省成本，因為它僅在運算時間內執行，而且在雲端環境中提供可擴充性和彈性。

暫時性 EMR 叢集是使用 Boto3 API 和 Python 程式設計語言在 Lambda 函數中啟動的。使用 Python 編寫的 Lambda 函數提供了在需要時啟動叢集的額外靈活性。

為了示範範例批次計算和輸出，此模式會從 Lambda 函數在 EMR 叢集中啟動 Spark 工作，並針對虛構公司的範例銷售資料執行批次計算。Spark 任務的輸出將是 Amazon 簡單存儲服務（亞馬遜 S3）中的逗號分隔值（CSV）文件。輸入資料檔案、Spark .jar 檔案、程式碼片段，以及用於執行運算的虛擬私有雲端 (VPC) 和 AWS 身分與存取管理 (IAM) 角色的 AWS CloudFormation 範本都會以附件形式提供。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

限制

- 一次只能從程式碼啟動一個 Spark 工作。

產品版本

- 在 Amazon EMR 6.0.0 上進行了測試

架構

目標技術堆疊

- Amazon EMR
- AWS Lambda
- Amazon S3
- Apache Spark

目標架構

自動化和規模

若要自動執行 Sparkk-EMR 批次計算，您可以使用下列其中一個選項。

- 實作 Amazon EventBridge 規則，該規則可以在 Cron 排程中啟動 Lambda 函數。如需詳細資訊，請參閱教學課程：使用 EventBridge.
- 設定 Amazon S3 事件通知，以在檔案送達時啟動 Lambda 函數。
- 透過事件主體和 Lambda 環境變數，將輸入參數傳遞至 AWS Lambda 函數。

工具

AWS 服務

- Amazon EMR 是一種受管叢集平台，可簡化在 AWS 上執行大數據架構以處理和分析大量資料的過程。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

- 阿帕奇星火是用於大規模數據處理的多語言分析引擎。

史诗

建立 Amazon EMR 和 Lambda IAM 角色和 VPC

任務	描述	所需技能
建立 IAM 角色和 VPC。	如果您已經擁有 AWS Lambda 和 Amazon EMR IAM 角色和 VPC，則可以略過此步驟。若要執行程式碼，EMR 叢集和 Lambda 函數都需要 IAM 角色。EMR 叢集也需要具有公用子網路的 VPC 或具有 NAT 閘道的私有子網路。若要自動建立所有 IAM 角色和 VPC，請依原樣部署附加的 AWS CloudFormation 範本，或者您可以依照其他資訊一節中的指定手動建立角色和 VPC。	雲端架構師
請注意 AWS CloudFormation 範本輸出金鑰。	成功部署 CloudFormation 範本後，瀏覽至 AWS CloudFormation 主控台中的 [輸出] 索引標籤。請注意五個輸出鍵： • `S3Bucket` • `LambdaExecutionRole` • `ServiceRole` • `JobFlowRole` • `Ec2SubnetId` 當您建立 Lambda 函數時，您將使用這些索引鍵中的值。	雲端架構師

上傳星火 .jar 文件

任務	描述	所需技能
上傳星火 .jar 文件。	將星火 .jar 檔案上傳到 AWS CloudFormation 堆疊建立的 S3 儲存貯體。值區名稱與輸出金鑰相同`S3Bucket`。	一般 AWS

建立 Lambda 函數以啟動 EMR 叢集

任務	描述	所需技能
建立 Lambda 函數。	在 Lambda 主控台上，建立具有執行角色的 Python 3.9+ Lambda 函數。執行角色原則必須允許 Lambda 啟動 EMR 叢集。請參閱隨附的 AWS CloudFormation 範本。)	資料工程師、雲端工程師
複製並粘貼代碼。	將`lambda_function.py` 檔案中的程式碼取代為此模式「其他資訊」區段中的程式碼。	資料工程師、雲端工程師
變更程式碼中的參數。	遵循程式碼中的註解，變更參數值以符合您的 AWS 帳戶。	資料工程師、雲端工程師
啟動函數以啟動叢集。	啟動函數，以使用提供的 Spark .jar 檔案來啟動暫態 EMR 叢集的建立。它將運行星火作業，並在作業完成時自動終止。	資料工程師、雲端工程師
檢查 EMR 叢集狀態。	啟動 EMR 叢集之後，叢集會顯示在 Amazon EMR 主控台的「叢集」索引標籤下。啟動	資料工程師、雲端工程師

任務	描述	所需技能
	叢集或執行作業時的任何錯誤都可以進行相應的檢查。	

設定並執行範例示範

任務	描述	所需技能
上傳星火 .jar 文件。	從附件部分下載 Spark .jar 文件，並將其上傳到 S3 存儲桶。	資料工程師、雲端工程師
上傳輸入資料集。	將附加的`fake_sales_data.csv` 檔案上傳到 S3 儲存貯體。	資料工程師、雲端工程師
貼上 Lambda 程式碼並變更參數。	從「工具」部分複製代碼，然後將代碼粘貼到 Lambda 函數中，替換代碼`lambda_function.py` 文件。變更參數值以符合您的帳戶。	資料工程師、雲端工程師
啟動函數並驗證輸出。	Lambda 函數使用提供的 Spark 任務初始化叢集之後，它會在 S3 儲存貯體中產生一個 .csv 檔案。	資料工程師、雲端工程師

相關資源

- 建築火花
- 阿帕奇星火和 Amazon EMR
- 博托 3 文檔運行工作流程文檔
- 阿帕奇星火的信息和文檔

其他資訊

Code

```
"""
Copy paste the following code in your Lambda function. Make sure to change the
 following key parameters for the API as per your account

-Name (Name of Spark cluster)
-LogUri (S3 bucket to store EMR logs)
-Ec2SubnetId (The subnet to launch the cluster into)
-JobFlowRole (Service role for EC2)
-ServiceRole (Service role for Amazon EMR)

The following parameters are additional parameters for the Spark job itself. Change the
 bucket name and prefix for the Spark job (located at the bottom).

-s3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar (Spark jar file)
-s3://your-bucket-name/prefix/fake_sales_data.csv (Input data file in S3)
-s3://your-bucket-name/prefix/outputs/report_1/ (Output location in S3)
"""
import boto3

client = boto3.client('emr')


def lambda_handler(event, context):
    response = client.run_job_flow(
        Name='spark_job_cluster',
        LogUri='s3://your-bucket-name/prefix/logs',
        ReleaseLabel='emr-6.0.0',
        Instances={
            'MasterInstanceType': 'm5.xlarge',
            'SlaveInstanceType': 'm5.large',
            'InstanceCount': 1,
            'KeepJobFlowAliveWhenNoSteps': False,
            'TerminationProtected': False,
            'Ec2SubnetId': 'subnet-XXXXXXXXXXXXX'
        },
        Applications=[{'Name': 'Spark'}],
        Configurations=[
            {'Classification': 'spark-hive-site',
             'Properties': {
```

```
                'hive.metastore.client.factory.class':
'com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory'}
            }
        ],
        VisibleToAllUsers=True,
        JobFlowRole='EMRLambda-EMREC2InstanceProfile-XXXXXXXXX',
        ServiceRole='EMRLambda-EMRRole-XXXXXXXXX',
        Steps=[
            {
                'Name': 'flow-log-analysis',
                'ActionOnFailure': 'TERMINATE_CLUSTER',
                'HadoopJarStep': {
                    'Jar': 'command-runner.jar',
                    'Args': [
                        'spark-submit',
                        '--deploy-mode', 'cluster',
                        '--executor-memory', '6G',
                        '--num-executors', '1',
                        '--executor-cores', '2',
                        '--class', 'com.aws.emr.ProfitCalc',
                        's3://your-bucket-name/prefix/lambda-emr/SparkProfitCalc.jar',
                        's3://your-bucket-name/prefix/fake_sales_data.csv',
                        's3://your-bucket-name/prefix/outputs/report_1/'
                    ]
                }
            }
        ]
    )
```

IAM 角色和 VPC 建立

若要在 Lambda 函數中啟動 EMR 叢集，則需要 VPC 和 IAM 角色。您可以使用此模式「附件」區段中的 AWS CloudFormation 範本來設定 VPC 和 IAM 角色，也可以使用下列連結手動建立角色。

執行 Lambda 和 Amazon EMR 需要下列 IAM 角色。

Lambda 行角色

Lambda 函數的執行角色授予其存取 AWS 服務和資源的權限。

Amazon EMR 的服務角色

Amazon EMR 角色可定義 Amazon EMR 在佈建資源和執行未在叢集內執行的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體環境中執行的服務層級任務時，允許的動作。例如，服務角色用於在叢集啟動時佈建 EC2 執行個體。

EC2 執行個體的服務角色

叢集 EC2 執行個體的服務角色 (也稱為 Amazon EMR 的 EC2 執行個體設定檔) 是一種特殊類型的服務角色，會在執行個體啟動時指派給 Amazon EMR 叢集中的每個 EC2 執行個體。在 Apache Hadoop 之上執行的應用程式處理程序會擔任此角色，以便與其他 AWS 服務互動的許可。

建立 VPC 和子網路

您可以從 VPC 主控台建立 VPC。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Glue 將阿帕奇卡桑德拉工作負載遷移到 Amazon Keyspaces

創建者：尼古拉·科列斯尼科夫（AWS），卡思加普里亞·錢德蘭（AWS）和薩米爾·帕特爾（AWS）

環境：生產	來源：卡桑德拉	目標：Amazon Keyspaces
R 類型：不適用	工作負載：開放原始碼；所有其他	技術：分析；移轉；無伺服器；大數據
AWS 服務：AWS AWS Glue；Amazon Keyspaces；Amazon S3；AWS CloudShell		

Summary

這種模式向您展示如何將現有的 Apache 卡桑德拉工作負載遷移到 Amazon Keyspaces（對於阿帕奇卡桑德拉）通過使用 AWS Glue CqlReplicator。您可以在 AWS Glue 上使用 CQlReplicator，將移轉工作負載的複寫延遲降到幾分鐘。您也會學到如何使用 Amazon Simple Storage Service (Amazon S3) 儲存貯體來存放遷移所需的資料，包括 Apache Parquet 檔案、組態檔案和指令碼。此模式假設您的 Cassandra 工作負載託管在虛擬私有雲 (VPC) 中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上。

先決條件和限制

先決條件

- 卡桑德拉群集與源表
- Amazon Keyspaces 中的目標資料表以複寫工作負載
- S3 儲存貯體，用於存放包含增量資料變更的中間 Parquet 檔案
- S3 儲存貯體來存放任務組態檔案和指令碼

限制

- AWS Glue 上的 CQLL 複製器需要一些時間來為卡桑德拉工作負載佈建資料處理單元 (DPU)。Cassandra 叢集與 Amazon 金鑰空間中的目標金鑰空間和表格之間的複寫延遲可能只持續幾分鐘。

架構

源, 技術, 堆棧

- 阿帕奇·卡桑德拉
- DataStax 伺服器
- 南瓜屬

目標技術堆疊

- Amazon Keyspaces

移轉架構

下圖顯示了一個示例架構，其中 Cassandra 叢集託管在 EC2 執行個體上並分散在三個可用區域。該卡桑德拉節點託管在私有子網。

該圖顯示以下工作流程：

1. 自訂服務角色可讓您存取 Amazon Keyspaces 和 S3 儲存貯體。
2. AWS Glue 任務會讀取 S3 儲存貯體中的任務組態和指令碼。
3. AWS AWS Glue 任務透過連接埠 9042 連線，以讀取卡桑德拉叢集中的資料。
4. AWS Glue 任務會透過連接埠 9142 連線，將資料寫入 Amazon Keyspaces。

工具

AWS 服務和工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

- AWS CloudShell 是以瀏覽器為基礎的殼層，您可以使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 和一系列預先安裝的開發工具來管理 AWS 服務。
- AWS Glue 是全受管的 ETL 服務，可協助您在資料存放區和資料串流之間可靠地分類、清理、豐富和移動資料。
- Amazon Keyspaces（適用於 Apache Cassandra）是一種受管的資料庫服務，可協助您在 AWS 雲端中遷移、執行和擴展 Cassandra 工作負載。

Code

此模式的程式碼可在 GitHub CQL Replicator 存放庫中取得。

最佳實務

- 若要判斷移轉所需的 AWS Glue 資源，請預估來源 Cassandra 資料表中的資料列數目。例如，每 0.25 個 DPU (2 個 vCPUs、4 GB 記憶體) 以及 84 GB 磁碟的每個資料列有 250 K 個資料列。
- 在運行 CQL 複製器之前預熱 Amazon Keyspaces 表。例如，八個 CQL複製器圖塊 (AWS Glue 任務) 每秒最多可寫入 22 K WCU，因此目標應預先加熱至每秒 25-30 K 的 WCU。
- 若要啟用 AWS Glue 元件之間的通訊，請對安全群組中的所有 TCP 連接埠使用自我參照的輸入規則。
- 使用增量流量策略隨時間分配移轉工作負載。

史诗

部署 CQL 複製器

任務	描述	所需技能
創建一個目標密鑰空間和表。	1. 在 Amazon 密鑰空間中創建一個 Keyspaces 間和表。 有關寫入容量的詳細資訊，請參閱此模式的其他資訊一節中的寫入單位計算。 您也可以通過使用卡桑德拉查詢語言（C QL）創建一個	應用程式擁有者、AWS 管理員、DBA、應用程式開發人

任務	描述	所需技能
	密鑰空間。如需詳細資訊，請參閱此模式的其他資訊一節中的使用 CQL 建立金鑰空間。 注意：建立表格之後，請考慮將表格切換為隨需容量模式，以避免不必要的費用。 2. 若要更新為輸送量模式，請執行下列指令碼： `ALTER TABLE target_keyspace.target_table WITH CUSTOM_PROPERTIES = { 'capacity_mode': { 'throughput_mode': 'PAY_PER_REQUEST'} }`	

任務	描述	所需技能
配置卡桑德拉驅動程序連接到卡桑德拉。	使用下列組態指令碼： `Datastax-java-driver {` `  basic.request.cons` `istency = “LOCAL_QU` `ORUM”` `  basic.contact-points` ` = [“127.0.0.1:9042”]` `   advanced.reconnect-` `on-init = true` `   basic.load-balanci` `ng-policy {` `        local-dat` `acenter = “datacent` `er1”` `}` `advanced.auth-provid` `er = {` `       class = PlainText` `AuthProvider` `       username = “user-` `at-sample”` `       password =` ` “S@MPLE=PASSWORD=”` `}` `}` 注意：前面的腳本使用星火卡桑德拉連接器。有關更多信息，請參閱卡桑德拉的參考配置。	DBA

任務	描述	所需技能
配置卡桑德拉驅動程序連接到 Amazon Keyspaces。	使用下列組態指令碼： `datastax-java-driver { basic { load-balancing-pol icy { local-datacenter = us-west-2 } contact-points = ["cassandr a.us-west-2.amazon aws.com:9142"] request { page-size = 2500 timeout = 360 seconds consistency = LOCAL_QUORUM } } advanced { control-connection { timeout = 360 seconds } session-leak.thres hold = 6 connection { connect-timeout = 360 seconds init-query-timeout = 360 seconds warn-on-init-error = false } auth-provider = { class = software. aws.mcs.auth.SigV4 AuthProvider aws-region = us- west-2`	DBA

任務	描述	所需技能
	<pre>} ssl-engine-factory { class = DefaultSs lEngineFactory } } }</pre> 注意：前面的腳本使用星火卡桑德拉連接器。有關更多信息，請參閱卡桑德拉的參考配置。	
為 AWS AWS Glue 任務建立 IAM 角色。	建立以 AWS Glue 命名`glue-cassandra-migration` 的新 AWS 服務角色，做為受信任的實體。 注意：`glue-cassandra-migration` 應該提供對 S3 存儲桶和 Amazon Keyspaces 的讀寫訪問權限。S3 存儲桶包含 .jar 文件，Amazon Keyspaces 和卡桑德拉的配置文件，以及中間實木複合地板文件。例如，它包含`AWSGlueServiceRole` `AmazonS3FullAccess` 、和`AmazonKeyspacesFullAccess` 受管理的策略。	AWS DevOps

任務	描述	所需技能
在 AWS 中下載 CQL 複製器。CloudShell	執行下列命令，將專案下載到您的主資料夾： `git clone https://github.com/aws-samples/cql-replicator.git` `cd cql-replicator/glue` `# Only for AWS CloudShell, the bc package includes bc and dc. Bc is an arbitrary precision numeric processing arithmetic language` `sudo yum install bc -y`	
修改參考組態檔案。	複製`CassandraConnector.conf` 和`Keyspaces Connector.conf` 到項`../glue/conf` 目文件夾中的目錄。	AWS DevOps

任務	描述	所需技能
啟動遷移程序。	下列指令會初始化 CQLReplicator 環境。初始化涉及複製 .jar 成品，以及建立 AWS Glue 連接器、S3 儲存貯體、AWS Glue 任務、migration 金鑰空間和資料表：ledger ``` cd cql-replicator/glue/bin ./cqlreplicator --state init --sg '"sg-1","sg-2"' \ --subnet "subnet-XXXXXXXXXXXX" \ --az us-west-2a --region us-west-2 \ --glue-iam-role glue-cassandra-migration \ --landing-zone s3://cql-replicator-1234567890-us-west-2 ``` 該指令碼包括下列參數： • --sg— 允許從 AWS Glue 存取 Cassandra 叢集的安全群組，並包含所有流量的自我參照入埠規則 • --subnet— 子網到其中卡桑德拉集群所屬 • --az— 子網路的可用區域 • --region— 部署卡桑德拉叢集的 AWS 區域	AWS DevOps

任務	描述	所需技能
	• `--glue-iam-role` — 代表您呼叫 Amazon Keyspaces 和 Amazon S3 時，AWS Glue 可以承擔的 IAM 角色許可 • `--landing zone`— 重複使用 S3 儲存貯體的選用參數 (如果您未提供`--landing zone`參數值，則`init`程序會嘗試建立新的儲存貯體來儲存組態檔、.jar 成品和中繼檔案。)	
驗證部署。	執行上一個命令之後，AWS 帳戶應包含下列項目： • AWS AWS Glue 中的 CQL 複製器 AWS AWS Glue 任務和 AWS AWS Glue 連接器 • 存放成品的 S3 儲存貯體 • 目標密鑰空間`migration` 和 Amazon Keyspaces 間中的`ledger`表	AWS DevOps

執行 CQL 複製器

任務	描述	所需技能
啟動移轉程序。	要在 AWS Glue 上操作 CQL 複製器，您需要使用`--state run`命令，然後是一系列參數。這些參數的精確組態主要由您獨特的移轉需求決定。例	AWS DevOps

任務	描述	所需技能
	如，如果您選擇複寫存留時間 (TTL) 值和更新，或者將超過 1 MB 的物件卸載到 Amazon S3，則這些設定可能會有所不同。 若要將工作負載從 Cassandra 叢集複寫到 Amazon Keyspaces，請執行下列命令： `./cqlreplicator --state run --tiles 8 \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --region us-west-2 \ --src-keyspace source_keyspace \ --src-table source_table \ --trg-keyspace taget_keyspace \ --writetime-column column_name \ --trg-table target_table --inc-traffic` 您的源密鑰空間和表 `source_keyspace.source_table` 在卡桑德拉集群中。您的目標密鑰空間和表	

任務	描述	所需技能
	位target_keyspace.target_table 於 Amazon Keyspaces 間中。此參數--inc-traffic 有助於防止增量流量因大量請求而使 Cassandra 叢集和 Amazon Keyspaces 超載。 若要複製更新，請新增--writetime-column regular_column_name 至您的命令列。常規列將被用作寫入時間戳的源。	

監控移轉程序

任務	描述	所需技能
在歷史遷移階段驗證遷移卡桑德拉行。	若要取得回填階段期間複製的資料列數目，請執行下列命令： ``` ./cqlreplicator --state stats \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --src-keyspace source_keyspace --src-table source_table --region us-west-2 ```	AWS DevOps

停止移轉程序

任務	描述	所需技能
使用命cqlreplicator 令或 AWS Glue 主控台。	若要正常停止移轉程序，請執行下列命令： `./cqlreplicator --state request-stop --tiles 8 \ --landing-zone s3://cql-replicator-1234567890-us-west-2 \ --region us-west-2 \ --src-keyspace source_keyspace --src-table source_table` 若要立即停止遷移程序，請使用 AWS Glue 主控台。	AWS DevOps

清除

任務	描述	所需技能
刪除已部署的資源。	下列命令會刪除 AWS Glue 任務、連接器、S3 儲存貯體和 Keyspaces 表格ledger： `./cqlreplicator --state cleanup --landing-zone s3://cql-replicato`	AWS DevOps

任務	描述	所需技能
	`r-1234567890-us-we st-2`	

故障診斷

問題	解決方案
AWS Glue 任務失敗並傳回記憶體不足 (OOM) 錯誤。	1. 變更 Worker 類型 (向上擴充)。例如，`G0.25X`將變更為`G.1X`或`G.1X`為`G.2X`。或者，在 CQL複製器中增加每個 AWS Glue 任務 (向外擴充) 的 DPU 數量。 2. 從移轉程序中斷的點開始。若要重新啟動失敗的 CQLReplicator 工作，請使用相同的參數重新執行`--state run`命令。

相關資源

- CQL 複製器，含 AWS AWS Glue 讀取器
- AWS AWS Glue 文件
- Amazon Keyspaces 文檔
- 阿帕奇·卡桑德拉

其他資訊

移轉考量

您可以使用 AWS Glue 將您的 Cassandra 工作負載遷移到 Amazon Keyspaces 間，同時保持 Cassandra 來源資料庫在遷移過程中完全正常運作。複寫完成後，您可以選擇將應用程式切斷到 Amazon Keyspaces 間，而 Cassandra 叢集和 Amazon 金 Keyspaces 間之間的複寫延遲最小 (不到分鐘)。為了保持資料一致性，您也可以使用類似的管道將資料從 Amazon Keyspaces 複寫回 Cassandra 叢集。

寫入單位計算

例如，假設您打算在一小時內寫入 500,000,000 的行大小為 1 KiB。您需要的 Amazon Keyspaces 寫入單位 (WCU) 總數是根據以下計算：

```
(number of rows/60 mins 60s) 1 WCU per row = (500,000,000/(60*60s) * 1 WCU) = 69,444 WCUs required
```

每秒 69,444 WCU 是 1 小時的速率，但您可以為開銷增加一些緩衝。 例如，`69,444 * 1.10 = 76,388 WCUs`有 10% 的額外負荷。

通過使用 CQL 創建密鑰空間

若要使用 CQL 建立金鑰空間，請執行下列命令：

```
CREATE KEYSPACE target_keyspace WITH replication = {'class': 'SingleRegionStrategy'}
CREATE TABLE target_keyspace.target_table ( userid uuid, level text, gameid int,
 description text, nickname text, zip text, email text, updatetime text, PRIMARY KEY
 (userid, level, gameid) ) WITH default_time_to_live = 0 AND CUSTOM_PROPERTIES =
 {'capacity_mode':{ 'throughput_mode':'PROVISIONED', 'write_capacity_units':76388,
 'read_capacity_units':3612 }} AND CLUSTERING ORDER BY (level ASC, gameid ASC)
```

將 Oracle 商業智慧 12c 從現場部署伺服器遷移到 AWS 雲端

創建者：蘭雷（藍雷）展覽（AWS）和帕特里克·黃（AWS）

環境：生產	來源：內部部署	目標：Amazon EC2，Amazon RDS，Amazon ALB，Amazon EFS
R 類型：重新平台	工作量：甲骨文	技術：分析；資料庫
AWS 服務：Amazon EBS；Amazon EC2；Amazon EFS；AWS CloudFormation；Elastic Load Balancing（ELB）；AWS Certificate Manager（ACM）		

Summary

此模式顯示如何使用 AWS 將 Oracle 商業智慧企業版 12c 從現場部署伺服器遷移到 AWS CloudFormation 雲端。它也說明如何使用其他 AWS 服務來實作 Oracle BI 12c 元件，以提供高可用性、安全性、彈性，以及動態擴展的能力。

如需將 Oracle BI 12c 遷移到 AWS 雲端相關的最佳實務清單，請參閱此模式的其他資訊一節。

附註：最佳做法是在將現有 Oracle BI 12c 資料傳輸到雲端之前執行多個測試移轉。這些測試可協助您微調移轉方法、識別和修正潛在問題，以及更準確地預估停機時間需求。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 透過 AWS 虛擬私有網路 (AWS VPN) 服務或 AWS 直接連接，在現場部署伺服器與 AWS 之間的安全網路連線
- 甲骨文操作系統，甲骨文 BI 12c，甲骨文數據庫，甲骨文服務器和甲骨文 HTTP WebLogic 服務器的軟件許可證

限制

如需儲存大小限制的相關資訊，請參閱 Amazon Relational Database Service 服務 (Amazon RDS) 的 Oracle 說明文件。

產品版本

- Oracle 智慧型商業管理系統企業版 12c
- 甲骨文 WebLogic 服务器
- 甲骨文 HTTP 服務器
- Oracle 資料庫 12c (或更新版本)
- 甲骨文爪哇 8

架構

下圖顯示在 AWS 雲端中執行 Oracle BI 12c 元件的範例架構：

此圖顯示下列架構：

1. Amazon Route 53 提供域名服務（DNS）配置。
2. Elastic Load Balancing (ELB) 可分配網路流量，以改善跨多個可用區域之 Oracle BI 12c 元件的延展性和可用性。
3. 亞馬遜彈性運算雲端 (Amazon EC2) Auto Scaling 群組在多個可用區域託管 Oracle HTTP 伺服器、Weblogic 管理伺服器和受管商業智慧伺服器。
4. Amazon Relational Database Service 服務 (Amazon RDS)，適用於跨多個可用區域的 Oracle 資料庫商業智慧伺服器中繼資
5. Amazon Elastic File System (Amazon EFS) 掛載到每個 Oracle BI 12c 元件上，以進行共用檔案儲存。

技術堆疊

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic File System (Amazon EFS)
- Amazon RDS for Oracle

- AWS Certificate Manager (ACM)
- Elastic Load Balancing (ELB)
- 甲骨文
- 甲骨文 WebLogic 服务器
- 甲骨文 HTTP 服務器（OHS）

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Certificate Manager (ACM) 可協助您建立、存放和更新公有和私有 SSL/TLS X.509 憑證和金鑰，以保護您的 AWS 網站和應用程式。
- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以根據需要啟動任意數量的虛擬伺服器，並快速擴展或縮減它們。
- Amazon EC2 Auto Scaling 可協助您維持應用程式的可用性，並允許您根據定義的條件自動新增或移除 Amazon EC2 執行個體。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。
- E@@ lastic Load Balancing 可將傳入的應用程式或網路流量分配到多個目標。例如，您可以在一個或多個可用區域中將流量分配到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。
- Oracle 資料汲取可協助您以高速將資料和中繼資料從一個資料庫移至另一個資料庫。
- Oracle Fusion 中間件是一套應用程序開發工具和集成解決方案，以身份管理，協作和商業智能報告。
- Oracle 可協 GoldenGate助您在 Oracle 雲端基礎架構中設計、執行、協調和監控資料複製和串流資料處理解決方案。

- 「Oracle 命 WebLogic 令檔工具」(WLST) 提供指令行介面，可協助您 WebLogic 水平向外擴充叢集。

史诗

評估來源環境

任務	描述	所需技能
收集軟體清查資訊。	識別來源技術堆疊之每個軟體元件的版本和修補程式層級，包括下列項目： • 甲骨文操作系統 • Oracle Database • 甲骨文 • Oracle WebLogic 伺服器 • 甲骨文伺服器 • Java	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管
收集運算和儲存庫存資訊。	在來源環境中，檢閱下列項目的目前和歷史使用率指標： • CPU 用量 • 記憶體用量 • 儲存使用量 重要:請確保您考慮使用量的歷史峰值。	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管理員、系統管
收集來源環境架構及其需求的相關資訊。	完整瞭解來源環境的架構及其需求，包括下列知識： • Oracle WebLogic 伺服器網域組態 • 聚類	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管

任務	描述	所需技能
	• 負載平衡 • 連線能力 • 可用性 • 災難復原需求	
識別 Java 資料庫連線能力 (JDBC) 資料來源。	針對來源環境所使用的每個資料庫引擎，收集來源環境 JDBC 資料來源和驅動程式的相關資訊。	移轉架構師、應用程式擁有者、Oracle BI 管理員、資料庫工程師或管理
收集有關環境特定設定的資訊。	收集來源環境特定之設定和組態的相關資訊，包括： • 自訂啟動和關閉指令碼 • Java 和其他環境變量 • 憑證	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管
識別其他應用程式的任何相依性。	收集來源環境中與其他應用程式建立相依性之整合的相關資訊。 重要：請確定您識別任何輕量型目錄存取通訊協定 (LDAP) 整合和其他網路需求。	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管

設計您的目標環境

任務	描述	所需技能
建立高階設計文件。	建立目標架構設計文件。請務必使用您在評估來源環境時收集的資訊來通知設計文件。	解決方案架構師、應用程式架構師、資料庫工程師、
取得設計文件的核准。	與利益相關者檢閱設計文件，並取得所需的核准。	應用程式或服務負責人、解決方案架構師、應用

部署基礎架構

任務	描述	所需技能
準備中的基礎結構程式碼 CloudFormation。	建立 CloudFormation 範本以在 AWS 雲端佈建您的 Oracle BI 12c 基礎設施。 如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的使用 AWS CloudFormation 範本。 注意：最佳做法是為每個 Oracle BI 12c 層建立 CloudFormation 模組化樣版，而不是針對所有資源建立一個大型樣板。如需 CloudFormation 最佳實務的詳細資訊，請參閱 AWS 部落格 CloudFormation上的 AWS 自動化部署時的 8 個最佳實務。	雲端基礎架構實現、解決方案架構師、應用程式架構
下載所需的軟體。	d 從 Oracle 網站下載下列軟體以及必要的版本和修補程式： • 爪哇 • 甲骨文 WebLogic 服务器 • 甲骨文	移轉架構師、資料庫工程師、應用架構
準備安裝指令碼。	建立執行無訊息安裝的軟體安裝指令碼。這些指令碼可簡化部署自動化。 如需詳細資訊，請參閱 OBIEE 12c：如何執行無訊息安裝？在「Oracle Support 部」網站上。您需要「Oracle 客戶	移轉架構師、資料庫工程師、應用架構

任務	描述	所需技能
	Support 部」帳戶才能檢視說明文件。	
為您的網頁和應用程式層建立一個以 Amazon EBS 為基礎的 Linux AMI。	1. 為您的 Web 和應用程式層級部署和設定 Amazon EC2 執行個體。請確定執行個體符合執行下列項目的先決條件： • Oracle 作業系統環境設定 • Oracle 作業系統使用者帳戶設定 • 安裝軟體 2. 建立執行個體的 Amazon 機器映像 (AMI)，並儲存副本以備 future 使用。如需指示，請參閱 Amazon EC2 執行個體使用者指南中的建立亞馬遜 EBS 支援 Linux AMI。	移轉架構師、資料庫工程師、應用架構
使用啟動 AWS 基礎設施 CloudFormation。	使用您建立的 CloudFormation 範本，在模組中部署 Oracle BI 12c Web 和應用程式層。 如需指示，請參閱 AWS CloudFormation 使用者指南 CloudFormation中的 AWS 入門。	雲基礎架構架構師、解決方案架構師、應用

使用全新安裝將甲骨文 BI 12c 遷移到 AWS

任務	描述	所需技能
暫存所需的軟體。	將所需軟體暫存在 Amazon EC2 執行個體可存取的位置。例如，您可以在 Amazon S3 或其他可供 Web 和應用程式伺服器存取的 Amazon EC2 執行個體中暫存軟體。	移轉架構師、Oracle BI 架構師、雲端基礎架構實現、解決方案架構師、應用程式架構師
準備您的儲存庫資料庫以進行 Oracle BI 12c 安裝。	透過針對新的亞馬遜 RDS 資料庫執行個體執行個體執行 Oracle 儲存庫建立公用程式 (RCU) 來建立甲骨文 BI 12c 架構。	雲基礎架構實現，解決方案架構師，應用程序架構師，遷移架構師，Oracle BI 架構師
安裝甲骨文融合中間件 12c 和甲骨文 BI 12c。	1. 從一個 Amazon EC2 實例開始，安裝甲骨文融合中間件 12c 基礎設施和 OBIEE 12c。如需詳細資訊，請參閱「Oracle 智慧型商業管理系統」的「Oracle Fusion 中介軟體企業建置指南」的下列章節： • 在 BIHOST1 上啟動基礎結構安裝程式 • 安裝「Oracle 智慧型商業管理系統」以準備企業部署 備註：使用 Amazon EFS 託管將在 Oracle BI 12c 叢集節點之間共用的目錄。 2. 將任何必要的修補程式套用至安裝。	遷移建築師，甲骨文 BI 架構師

任務	描述	所需技能
	3. 建立執行個體的 AMI 並儲存副本以供 future 使用。	
為甲骨文商業智能 12c 配置您的甲骨文 WebLogic 服務器域。	將您的 Oracle BI 12c 網域設定為非叢集部署。 如需詳細資訊，請參閱《Oracle Fusion 中介軟體企業部署指南》中的《Oracle 智慧型商業管理系統》中的設定 BI 網域。	遷移建築師，甲骨文 BI 架構師
執行水平擴展出甲骨文 BI 12c 的。	水平向外擴展單個節點到所需的節點數量。 如需更多資訊，請參閱《Oracle 融合中介軟體企業部署指南》中的「Oracle 智慧型商業管理系統」中的「向外延展 Oracle 智慧	遷移建築師，甲骨文 BI 架構師
安裝甲骨文 HTTP 服務器 12c。	1. 在甲骨文網絡層 Amazon EC2 實例上安裝甲骨文 HTTP 服務器 12c。如需指示，請參閱針對 Oracle 存取管理系統 12c 安裝和設定 Oracle HTTP 伺服器中的安裝 Oracle HTTP 伺服器。 2. 將任何必要的修補程式套用至安裝。 3. 建立執行個體的 AMI 並儲存副本以供 future 使用。	遷移建築師，甲骨文 BI 架構師

任務	描述	所需技能
設定 SSL 終止的負載平衡器。	1. 在 ACM 中建立或匯入 SSL 憑證。 2. 將 SSL 憑證與 ELB 建立關聯。	雲端基礎架構架構師、移轉架
將商業智慧中繼資料成品遷移到 AWS。	1. 從內部部署 Oracle BI 12c 安裝匯出「Oracle 智慧型商業管理系統」應用程式存檔 (BAR) 檔案。若要匯出 BAR 檔案，請使用指WebLogic 令碼處理工具 (WLST) 來執行命令。exportServiceInstance 2. 將現場部署 BAR 檔案匯入 AWS 甲骨文 BI 12c 安裝。若要匯入 BAR 檔案，請執行 importServiceInstanceWLST 指令。	遷移建築師，甲骨文 BI 架構師

任務	描述	所需技能
執行移轉後工作。	匯入 BAR 檔案後，請執行下列動作： • 設定任何其他 JDBC 資料來源。 • 為 PostgreSQL 或 Amazon Redshift 等其他資料來源安裝驅動程式。 • 設定 Oracle LDAP、SSL、單一登入 (SSO) 和WebLogic 安全性存放區。 • 設定 AWS Identity and Access Management (IAM) 政策。 • 啟用使用狀況追蹤。 • 設定與其他系統的整合。 • 移轉任何自訂指令碼。	遷移建築師，甲骨文 BI 架構師

測試新環境

任務	描述	所需技能
測試新的甲骨文 BI 12c 環境。	在新的甲骨文 BI 12c 環境進行 end-to-end 測試。盡可能使用自動化。 測試活動的例子包括以下內容： • 驗證控制面板、報表和 URL • 使用者驗收測試 (UAT) • 操作驗收測試（OAT）	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管

任務	描述	所需技能
	注意：根據需要進行其他測試和驗證。	

切換到新的環境

任務	描述	所需技能
中斷與內部部署 Oracle BI 12c 環境的流量連線。	在指定的切換視窗中，停止內部部署 Oracle BI 12c 環境的所有流量。	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管
重新同步新的 Oracle BI 12c 儲存庫資料庫與來源資料庫。	將 Amazon RDS 甲骨文 BI 12c 儲存庫資料庫與現場部署資料庫重新同步。 若要同步資料庫，您可以使用 Oracle 資料泵重新整理或 AWS DMS 變更資料擷取 (CDC)。	甲骨文 BI 管理員, 數據庫工程師/管理員
將您的甲骨文 BI 12c 網址切換為指向新的 AWS 環境。	更新內部 DNS 伺服器上的甲骨文 BI 12c 網址，以便它們指向新的 AWS 安裝。	移轉架構師、解決方案架構師、應用程式擁有者、Oracle BI 管
監控新環境。	監視新的甲骨文 BI 12c 環境，通過使用下列任何工具： • Amazon CloudWatch • Amazon RDS Performance Insights • Oracle Enterprise Manager	Oracle BI 管理員，數據庫工程師/管理員，應用管理員
取得專案的簽署。	與利益相關者檢閱測試結果，並取得必要的核准，以整理遷移作業。	應用程式擁有者、服務擁有者、雲端基礎架構設計師、移轉架構師、Oracle BI 架

相關資源

- 在 Oracle 版 RDS 上使用 Oracle 儲存庫建立公用程式 (Amazon RDS 使用者指南)
- Amazon RDS 上的甲骨文 (Amazon RDS 用戶指南)
- AWS 上的甲骨文 WebLogic 伺服器 12c (AWS 白皮書)
- 針對高可用性建置 Oracle 智慧型商業管理系統 (Oracle 說明中心)
- Oracle 智慧型商業管理系統應用模組存檔 (BAR) 檔案 (Oracle 說明中心)
- 如何在環境之間移轉 OBI 12c (Oracle 客 Support 服務中心)

其他資訊

以下是與將 Oracle BI 12c 遷移到 AWS 雲端相關的最佳實務清單。

儲存庫資料

最佳做法是在亞馬遜 RDS 為甲骨文執行個體上託管甲骨文 BI 12c 資料庫結構描述。此執行個體類型提供符合成本效益且可調整大小的容量，同時自動執行硬體佈建、資料庫設定、修補和備份等管理工作。

如需詳細資訊，請參閱 Amazon RDS 使用者指南中的在 RDS 版 Oracle 上使用 Oracle 儲存庫建立公用程式。

Web 和應用程式層

記憶體優化的 Amazon EC2 執行個體通常非常適合甲骨文 BI 12c 伺服器。無論您選擇何種執行個體類型，請確定您佈建的執行個體符合系統的記憶體使用需求。此外，請確定根據 Amazon EC2 執行個體的可用記憶體設定足夠的 WebLogic Java 虛擬機器 (JVM) 堆積大小。

本機儲存

I/O 扮演甲骨文 BI 12c 應用程序的整體性能的重要組成部分。Amazon Elastic Block Store (Amazon EBS) 提供針對不同工作負載模式進行優化的不同儲存類別。請務必選擇適合您使用案例的 Amazon EBS 磁碟區類型。

如需 EBS 磁碟區類型的詳細資訊，請參閱 Amazon EBS 文件中的 Amazon EBS 功能。

共用儲存

叢集的 Oracle BI 12c 網域需要下列資源的共用儲存空間：

- 組態檔案
- 甲骨文 BI 12c 單例數據目錄（SDD）
- 全域快取
- 甲骨文 BI 調度器腳本
- WebLogic 服务器二进制文件

您可以使用 Amazon EFS 來滿足此共用儲存需求，該 EFS 提供可擴展的全受管彈性網路檔案系統 (NFS) 檔案系統。

微調共用儲存空間效能

Amazon EFS 具有兩種輸送量模式：佈建和成組分解。此服務也有兩種效能模式：「一般用途」和「最大 I/O」。

若要微調效能，請先在一般用途效能模式和佈建輸送量模式下測試工作負載。執行這些測試可協助您判斷這些基準模式是否足以滿足您想要的服務等級。

如需詳細資訊，請參閱 Amazon EFS 使用者指南中的 Amazon EFS 效能。

可用性和災難復原

最佳做法是跨多個可用區域部署 Oracle BI 12c 元件，以便在可用區域故障時保護這些資源。以下是 AWS 雲端託管之特定 Oracle BI 12c 資源的可用性和災難復原最佳實務清單：

- 甲骨文 BI 12c 儲存庫資料庫：將異地同步備份 Amazon RDS 資料庫執行個體部署到您的 Oracle 商業智慧 12 機構資料庫。在異地同步備份部署中，Amazon RDS 會在不同的可用區域中自動佈建和維護同步備用複本。跨可用區域執行 Oracle BI 12c 儲存庫資料庫執行個體可增強規劃的系統維護期間的可用性，並協助保護資料庫免受執行個體和可用區域故障的影響。
- Oracle BI 12c 受管伺服器：為了實現容錯，最佳做法是在設定為跨多個可用區域的 Amazon EC2 Auto Scaling 群組中的受管理伺服器上部署 Oracle BI 12c 系統元件。Auto Scaling 會根據 Amazon EC2 運作狀態檢查取代故障執行個體。如果可用區域失敗，Oracle HTTP 伺服器會繼續將流量導向運作中可用區域中的「受管理的伺服器」。然後，Auto Scaling 會啟動執行個體，以跟上您的主機數量需求。建議您啟動 HTTP 工作階段狀態複寫，以協助確保現有工作階段可順利容錯移轉至正常運作的受管理伺服器。
- Oracle BI 12c 管理伺服器：為確保您的管理伺服器具有高可用性，請將其託管在設定為跨多個可用區域的 Amazon EC2 Auto Scaling 群組中。然後，將組的最小和最大大小設置為 1。如果發生可用區域故障，Amazon EC2 Auto Scaling 會在替代可用區域中啟動更換的管理伺服器。若要復原相同可用區域內任何故障的基礎主機，您可以啟用 Amazon EC2 自動復原。

- Oracle 網頁層伺服器：將 Oracle HTTP 伺服器與您的 Oracle WebLogic 伺服器網域建立關聯是最佳作法。為了獲得高可用性，請在設定為擴展多個可用區域的 Amazon EC2 Auto Scaling 群組中部署 Oracle HTTP 伺服器。然後，將伺服器置於 ELB 彈性負載平衡器後方。若要提供額外的主機故障保護，您可以啟用 Amazon EC2 自動復原。

擴充性

AWS 雲端的彈性可協助您水平或垂直擴展應用程式，以回應工作負載需求。

垂直縮放

若要垂直擴展應用程式，您可以變更執行 Oracle BI 12c 元件之 Amazon EC2 執行個體的大小和類型。您不需要在部署開始時過度佈建執行個體，而且會產生不必要的費用。

水平縮放

Amazon EC2 Auto Scaling 可根據工作負載需求自動新增或移除受管伺服器，協助您水平擴展應用程式。

注意：使用 Amazon EC2 Auto Scaling 進行水平擴展需要指令碼技能和完整的測試才能實作。

Backup 與復原

以下是 AWS 雲端託管之特定 Oracle BI 12c 資源的備份和復原最佳實務清單：

- Oracle 商業智慧型中繼資料儲存庫：Amazon RDS 會自動建立並儲存資料庫執行個體的備份。這些備份會保留您指定的一段時間。請務必根據資料保護需求設定 Amazon RDS 備份持續時間和保留設定。如需詳細資訊，請參閱 Amazon RDS 備份和還原。
- 受管伺服器、管理伺服器和 Web 層伺服器：確保根據資料保護和保留需求設定 Amazon EBS 快照。
- 共用儲存：您可以使用 AWS Backup 管理存放在 Amazon EFS 中的檔案的備份和復原。AWS Backup 服務也可以部署到集中管理其他服務的備份和復原，包括 Amazon EC2、Amazon EBS 和 Amazon RDS。如需詳細資訊，請參閱什麼是 AWS Backup？ 在 AWS Backup 開發人員指南中。

安全性與合規性

以下是可協助您在 AWS 雲端保護 Oracle BI 12c 應用程式的安全最佳實務和 AWS 服務清單：

- 靜態加密：Amazon RDS、Amazon EFS 和 Amazon EBS 均支援業界標準加密演算法。您可以使用 AWS Key Management Service (AWS KMS) 建立和管理加密金鑰，並控制其在 AWS 服務和應用程

式中的使用情況。您也可以在託管 Oracle BI 12c 儲存庫資料庫的 Amazon RDS for Oracle 資料庫執行個體上設定甲骨文透明資料加密 (TDE)。

- 傳輸中加密：最佳做法是啟動 SSL 或 TLS 通訊協定，以保護 Oracle BI 12c 安裝各層之間傳輸中的資料。您可以使用 AWS Certificate Manager (ACM) 為您的 Oracle BI 12c 資源佈建、管理和部署公有和私有 SSL 和 TLS 憑證。
- 網路安全性：請確定您已在 Amazon VPC 中部署 Oracle BI 12c 資源，且該虛擬私人雲端已針對您的使用案例設定了適當的存取控制。設定您的安全群組，以篩選執行安裝之 Amazon EC2 執行個體的入站和出站流量。此外，請務必設定網路存取控制清單 (NACL)，以根據定義的規則允許或拒絕流量。
- 監控和記錄：您可以使用 AWS CloudTrail 追蹤對 AWS 基礎設施的 API 呼叫，包括 Oracle BI 12c 資源。此功能在追蹤基礎結構的變更或進行安全性分析時非常有用。您也可以使用 Amazon CloudWatch 檢視操作資料，這些資料可以為您提供可行的 Oracle BI 12c 應用程式效能和運作狀態的深入解析。您也可以配置警報並根據這些警報採取自動化操作。Amazon RDS 提供其他監控工具，包括增強型監控和 Perfor mance Insights。

使用以下方式將現場部署阿帕奇卡夫卡叢集遷移到 Amazon MSK MirrorMaker

由張漢（AWS）和坦納普拉特（AWS）創建

環境：PoC 或試點	來源：內部部署或自我管理的 Apache 卡夫卡群集	目標：阿帕奇卡夫卡（Amazon MSK）的 Amazon 託管流
R 類型：重新平台	工作負載：開放原始碼；所有其他	技術：分析；大數據；遷移
AWS 服務：Amazon MSK		

Summary

此模式提供了將現場部署、自我管理或託管的 Apache Kafka 叢集遷移至適用於 Apache Kafka (Amazon MSK) 的 Amazon 受管串流的指導。您也可以使用此模式從一個 Amazon MSK 叢集遷移到另一個叢集。

阿帕奇卡夫卡包括的 MirrorMaker 功能，它複製兩個卡夫卡集群之間的數據。 MirrorMaker 由消費者，這是一個消費者組的一部分的集合。取用者會從來源叢集中的主題讀取資料，然後將此資料傳遞給生產者，這些產生器會將資料寫入目標叢集。

Amazon MSK 文件包含程序的高階概觀，可使用 1.0 MirrorMaker 版將現場部署 Kafka 叢集遷移至 Amazon MSK。此模式通過提供使用 2.0 MirrorMaker 版的全面 step-by-step 說明來補充此信息。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Kafka 來源叢集是下列其中一項：
  - 在內部部署資料中心
  - 雲端中的自我管理
  - 透過合作夥伴託管

限制

- 若要使用 2.0 MirrorMaker 版，來源叢集必須執行 Apache 卡夫卡 2.4.0 或更新版本。如需舊版，請參閱 Amazon MSK 文件中的指示，以便使用 1.0 MirrorMaker 版。

產品版本

- MirrorMaker 2.0 版本
- 阿帕奇卡夫卡 2.4.0 版本或更高版本。如需 Amazon MSK 支援的 Apache 卡夫卡版本的詳細資訊，請參閱支援的 Apache 卡夫卡版本。

架構

源, 技術, 堆棧

- 內部部署或自我管理的卡夫卡叢集

目標技術堆疊

- Amazon MSK 叢集

目標架構

該圖顯示了以下過程：

1. MirrorMaker 讀取來源卡夫卡叢集中的主題和用戶群組的資料。
2. MirrorMaker 將資料和消費者資訊複寫到目標 Amazon MSK 叢集。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

- 適用 Managed Streaming for Apache Kafka (Amazon MSK) 是一項全受管服務，可協助您建置和執行使用 Apache Kafka 處理串流資料的應用程式。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他工具

- 阿帕奇卡夫卡是一個開源的事件流媒體平臺。在這種模式中，您可以使用 Kafka 的MirrorMaker功能來執行跨群集遷移。

最佳實務

您可以 MirrorMaker 在來源環境或目標環境中執行，但建議您盡可能靠近目標叢集執行它。如需詳細資訊，請參閱 Apache Kafka 文件中的最佳作法：從遠端使用、產品至本機。

史诗

建立 VPC 並以 Amazon MSK 叢集為目標

任務	描述	所需技能
建立 VPC。	1. 在目標 AWS 帳戶中建立 VPC。如需指示，請參閱建立 VPC。 2. 在新 VPC 中的不同可用區域中建立三個私有子網路。如需指示，請參閱建立子網路。使用不同的可用區域可提供高可用性和容錯能力。 注意：如果您使用公用網際網路連線來移轉 Kafka 叢集，請建立公用子網路並啟用 Amazon MSK 叢集的公用存取權。	AWS 系統管理員、 DevOps 工程師、雲端管理員

任務	描述	所需技能
建立 Amazon MSK 叢集。	建立 Amazon MSK 叢集。如需指示，請參閱使用 AWS 管理主控台建立叢集或使用 AWS CLI 建立叢集。將叢集設定為使用您先前建立的 VPC 和子網路。	AWS 系統管理員、 DevOps 工程師、雲端管理員

設定 MirrorMaker

任務	描述	所需技能
安裝 MirrorMaker。	1. 啟動 EC2 執行個體。 2. Connect 至您的 EC2 執行個體。 3. 在 EC2 執行個體上，下載並擷取最新的 Kafka 版本。有關說明，請參閱快速入門（Kafka 文檔）。 注意：在此模式中，您將 MirrorMaker 2.0 安裝為 Amazon EC2 執行個體上的專用 MirrorMaker 叢集。此選項適用於開發環境，並且是此模式中使用的方法。如需 MirrorMaker 2.0 其他部署選項的詳細資訊，請參閱此模式的其他資訊一節。	AWS 系統管理員、雲端管理員、 DevOps 工程師
指定卡夫卡叢集資訊。	在卡夫卡客戶端安裝文件bin夾中，創建一個 mm2.properties 文件，並為您的源卡夫卡集群配置它。	AWS 系統管理員、雲端管理員、 DevOps 工程師

任務	描述	所需技能
	如需指示，請參閱執行專用 MirrorMaker 叢集 (Kafka 文件)。	
開始 MirrorMaker。	輸入下列命令以啟動 MirrorMaker 並傳遞 mm2.properties 檔案。 `$ ./bin/connect-mirror-maker.sh mm2.properties`	AWS 系統管理員、雲端管理員、 DevOps 工程師
監控進度。	檢查進度，方法 MirrorMaker 是檢查每個主題的最後一個偏移量與正在消耗的主題目前偏移之間的延遲。如需指示，請參閱 Kafka 文件中的監視異地複寫。	AWS 系統管理員、雲端管理員、 DevOps 工程師

切過

任務	描述	所需技能
停止消費者應用程序。	停止使用來源叢集資料的所有用戶應用程式。	應用程式開發人員
啟動消費者應用程式。	變更應用程式啟動程序組態，以指向目的地叢集。然後開始在目標叢集上消耗。	應用程式開發人員
停止來源叢集上的產生器。	當用戶應用程式在目標叢集上成功使用時，請停止來源叢集上的產生器。	應用程式開發人員
啟動目標叢集上的產生器。	變更生產者的組態啟動程序伺服器，並指向目標叢集。在啟	應用程式開發人員

任務	描述	所需技能
	動生產者之前，請等待來源叢集中的所有資料完成鏡像。MirrorMaker	
停止 MirrorMaker。	在生產者移至目標叢集之後，請停止 MirrorMaker。	AWS 系統管理員、雲端管理員、 DevOps 工程師

相關資源

AWS 資源

- 使用以下方式遷移叢集 MirrorMaker (Amazon MSK 文件)
- Amazon MSK 遷移實驗室 (AWS 工作坊工作室)

其他資源

- MirrorMaker 2.0（阿帕奇卡夫卡改善提案）
- 異地複寫：跨叢集資料鏡像 (Apache 卡夫卡文件)

其他資訊

此模式在 Amazon EC2 上以專用 MirrorMaker 叢集的形式執行 MirrorMaker 2.0。此選項適用於開發環境。雖然它沒有在這種模式中討論，你也可以在卡夫卡 Connect 集群運行 MirrorMaker 2.0。此部署選項使用 Kafka 生態系統中的框架，以改善擴展和維護。您可以將連接器部署到 Kafka Connect 叢集中，並使用關聯的組態來執行應用程式。連接器可以在獨立模式下執行以進行開發或測試，或以分散式模式執行以進行生產。如需詳細資訊，請參閱 MirrorMaker 在 Connect 叢集中執行 (Apache Kafka 說明文件)。如需其他 MirrorMaker 2.0 部署選項的詳細資訊，請參閱逐步解說：執行 MirrorMaker 2.0 (Kafka 說明文件)。

將 ELK 堆疊遷移到 AWS 上的彈性雲端

創建者：巴圖爾加·普雷瓦拉查（AWS），烏代·雷迪和安東尼·普拉薩德·泰瓦拉治（AWS）

環境：生產	來源：彈性搜索	目標：彈性雲
R 類型：重新平台	工作負載：所有其他工作	技術：分析；安全性、身分識別、合規性
AWS 服務：Amazon EC2；Amazon EC2 Auto Scaling；Elastic Load Balancing（ELB）；Amazon S3；Amazon Route 53		

Summary

E@@ lastic 多年來一直提供服務，其使用者和客戶通常會在內部部署自行管理 Elastic。彈性雲端是受管理的 Elasticsearch 服務，提供了一種使用彈性堆疊 (ELK 堆疊) 和企業搜尋、可觀察性和安全性解決方案的方法。您可以使用日誌、指標、APM（應用程式效能監控）和 SIEM（安全性資訊和事件管理）等應用程式存取彈性解決方案。您可以使用整合式功能，例如機器學習、索引生命週期管理、Kibana Lens (用於拖放視覺效果)。

當您從自我管理的彈性搜索轉移到彈性雲時，Elasticsearch 服務會處理以下事項：

- 佈建及管理基礎架構
- 建立和管理彈性搜尋叢集
- 向上和向下調整叢集
- 升級、修補和擷取快照

這讓您有更多時間專注於解決其他挑戰。

此模式定義如何將現場部署彈性搜尋 7.13 遷移至 Amazon Web Services (AWS) 上彈性雲端上的彈性搜尋。其他版本可能需要對此模式中描述的過程進行輕微修改。如需詳細資訊，請聯絡您的彈性代表。

先決條件和限制

先決條件

- 可存取亞馬遜簡單儲存服務 (Amazon S3) 快照的有效 AWS 帳戶
- 安全、足夠高頻寬的私有連結，可將快照資料檔案複製到 Amazon S3
- Amazon S3 Transfer Acceleration
- 彈性快照政策可確保定期將資料擷取存檔到足夠大的本機資料存放區或遠端儲存 (Amazon S3)

在啟動遷移之前，您必須了解隨附索引的快照集和生命週期政策在內部部署的大小。如需更多資訊，請聯絡彈性。

角色和技能

移轉程序也需要下表所述的角色和專業知識。

Role	專业	責任
應用支援	熟悉彈性雲端和內部部署彈性	所有彈性相關任務
系統管理員或 DBA	深入了解內部部署彈性環境及其組態	能夠佈建儲存、安裝和使用 AWS Command Line Interface (AWS CLI) (AWS CLI)，以及識別在現場部署提供彈性的所有資料來源
網路管理員	具備現場部署到 AWS 網路連線、安全性和效能的知識	建立從內部部署到 Amazon S3 的網路連結，並了解連線頻寬

限制

- 彈性雲端上的彈性搜尋僅適用於支援的 AWS 區域 (2021 年 9 月)。

產品版本

- 彈性搜索 7.13

架構

源, 技術, 堆棧

內部部署彈性搜尋 7.13 或更新版本：

- 叢集快照
- 索引快照
- 節拍配置

源代碼技術架構

下圖顯示具有不同擷取方法、節點類型和 Kibana 的典型內部部署架構。不同的節點類型會反映 Elasticsearch 叢集、驗證和視覺化角色。

1. 從節拍攝到記錄儲存
2. 從節拍攝到阿帕奇卡夫卡消息隊列
3. 從文件節拍攝到洛格斯塔什
4. 從阿帕奇卡夫卡消息隊列攝入到 Logstash
5. 從記錄儲存擷取至彈性搜尋叢集
6. 彈性搜索集群
7. 認證和通知節點
8. 木花和斑點節點

目標技術堆疊

彈性雲端會透過跨叢集複寫部署到多個 AWS 區域中的軟體即服務 (SaaS) 帳戶。

- 叢集快照
- 索引快照
- 節拍配置
- 彈性雲
- Network Load Balancer

- Amazon Route 53
- Amazon S3

目標架構

受管理的彈性雲端基礎架構為：

- 高可用性，存在於多個可用區域和多個 AWS 區域。
- 因為資料 (索引和快照) 是使用彈性雲端跨叢集複寫 (CCR) 來複寫，因此具有區域容錯能力
- 存檔，因為快照已存檔在 Amazon S3 中
- 透過網路負載平衡器和 Route 53 的組合來容忍網路磁碟分割
- 資料擷取源自 (但不限於) 彈性 APM、節拍、記錄儲存

高階移轉步驟

Elastic 已經開發了自己的規範方法，用於將內部部署彈性群集遷移到彈性雲。彈性方法與 AWS 遷移指導和最佳實務直接配合，包括 Well-Architected 的框架和 AWS Migration Acceleration Program (MAP)。一般而言，三個 AWS 遷移階段如下：

- 評估
- 調動
- 遷移和現代化

Elastic 遵循類似的遷移階段以及互補術語：

- 啟動
- 計畫
- 實施
- 交付
- Close (關閉)

Elastic 使用彈性實施方法來促進項目成果的交付。這在設計上具有包容性，以確保 Elastic，諮詢團隊和客戶團隊能夠清晰地合作以共同提供預期的結果。

彈性方法在實施階段結合了傳統的瀑布階段與 Scrum。技術需求的組態會以協同合作的方式反覆提供，同時將風險降至最低。

工具

AWS 服務

- Amazon Route 53 — Amazon Route 53 是一種高可用性和可擴展的域名系統（DNS）網絡服務。您可以使用 Route 53 執行三個主要功能的任意組合：網域註冊、DNS 路由和運作狀態檢查。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。此模式使用 S3 儲存貯體和 Amazon S3 Transfer Accel eration。
- E@@ lastic Load Balancing — Elastic Load Balancing 會在一或多個可用區域中自動將傳入流量分配到多個目標，例如 EC2 執行個體、容器和 IP 地址。

其他工具

- 節拍-擊敗來自 Logstash 或彈性搜索的船舶數據
- 彈性雲 — 彈性雲是一種託管彈性搜索的託管服務。
- 彈性搜索 — Elasticsearch 是一種搜索和分析引擎，使用彈性堆棧集中存儲您的數據，以進行可擴展的搜索和分析。此模式也會使用快照建立和跨叢集複寫。
- Logstash — Logstash 是伺服器端資料處理管道，可從多個來源擷取資料、對其進行轉換，然後將其傳送至您的資料儲存體。

史诗

準備移轉

任務	描述	所需技能
識別執行內部部署彈性解決方案的伺服	確認支援彈性移轉。	應用所有者
瞭解內部部署伺服器組態。	若要瞭解在內部部署成功驅動工作負載所需的伺服器組態，	應用 Support

任務	描述	所需技能
	請尋找目前使用中的伺服器硬體佔用空間、網路組態和儲存特性	
收集使用者和應用程式帳戶資訊。	識別內部部署彈性環境所使用的使用者名稱和應用程式名稱。	系統管理員，應用支持
文件節拍和資料寄件人配置。	若要記錄組態，請查看現有資料來源和接收器。如需詳細資訊，請參閱彈性文件。	應用支援
決定資料的速度和體積。	建立叢集處理多少資料的基準。	系統管理員，應用支持
記錄 RPO 和 RTO 案例。	文件復原點目標 (RPO) 與復原時間目標 (RTO) 案例 (以中斷與服務層次協定 (SLA) 為基礎。	應用程式擁有者、系統管理員、App 支援
決定最佳快照生命週期設定。	定義移轉期間和移轉後使用彈性快照保護資料的頻率。	應用程式擁有者、系統管理員、App 支援
定義移轉後的效能預期。	根據目前和預期的螢幕重新整理、查詢執行階段和使用者介面行為產生指標。	系統管理員，應用支持
記錄網際網路存取傳輸、頻寬和可用性需求。	確定將快照複製到 Amazon S3 的網際網路連線速度、延遲和彈性。	網路管理員
記錄彈性內部部署執行階段的目前成本。	確保 AWS 目標環境的大小設計既高效能又符合成本效益。	DBA、系統管理員、應用程式支援

任務	描述	所需技能
識別驗證和授權需求。	彈性堆疊安全性功能提供內建領域，例如輕量型目錄存取通訊協定 (LDAP)、安全性宣告標記語言 (SAML) 和 OpenID Connect (OIDC)。	DBA、系統管理員、應用程式支援
根據地理位置瞭解特定法規要求。	確保根據您的要求和任何國家相關要求導出和加密數據。	DBA、系統管理員、應用程式支援

實施遷移

任務	描述	所需技能
在 Amazon S3 上準備暫存區域。	若要在 Amazon S3 上接收快照，請建立 S3 儲存貯體和臨時 AWS Identity and Access Management (IAM) 角色，以完全存取新建立的儲存貯體。如需詳細資訊，請參閱建立角色以委派許可給 IAM 使用者。使用 AWS Security Token Service 要求臨時安全登入資料。保持訪問密鑰 ID，秘密訪問密鑰和會話令牌的安全。 在儲存貯體上啟用 Amazon S3 Transfer Acceleration。	AWS 管理員
在現場部署安裝 AWS CLI 和 Amazon S3 外掛程式。	在每個彈性搜尋節點上，執行下列命令。 `sudo bin/elasticsearch-plugin install repository-s3`	AWS 管理員

任務	描述	所需技能
	然後重新啟動節點。	
設定 Amazon S3 用戶端存取。	新增先前透過執行下列命令建立的金鑰。 `elasticsearch-keystore add s3.client.default.access_key` `elasticsearch-keystore add s3.client.default.secret_key` `elasticsearch-keystore add s3.client.default.session_token`	AWS 管理員
註冊彈性資料的快照儲存庫	使用 Kibana 開發工具告知現場部署本機叢集要寫入哪個遠端 S3 儲存貯體。	AWS 管理員
設定快照原則。	若要設定快照生命週期管理，請在 Kibana 政策索引標籤上選擇 SLM 原則，然後定義應包含哪些時間、資料串流或索引，以及要使用的名稱。 設定需要頻繁快照的策略。快照是增量的，可以有效利用儲存裝置。符合您的準備程度評估決定。政策也可以指定保留政策，並在不再需要快照時自動刪除快照。	應用支援

任務	描述	所需技能
確認快照可正常運作。	在 Kibana 開發工具中，執行下列命令。 `GET _snapshot/<your_repo_name>/_all`	AWS 管理員、應用程式支援、
在彈性雲端上部署新叢集。	登入 Elastic，並選擇叢集，以取得從整備程度評估中的業務發現所衍生的「可觀察性、搜尋或安全性」。	AWS 管理員、應用程式支援
設定叢集金鑰存放區存取權。	新叢集需要存取將儲存快照的 S3 儲存貯體。在 Elasticsearch 服務主控台上，選擇 [安全性]，然後輸入您先前建立的存取權和秘密 IAM 金鑰。	AWS 管理員
設定彈性雲端託管叢集以存取 Amazon S3。	針對先前在 Amazon S3 中建立的快照儲存庫設定新的叢集存取權。使用木花，請執行以下操作： 1. 選擇 [堆疊管理]、[快照設定] RegisterRepo。 2. 在「別名」欄位中，輸入存放庫的名稱。 3. 對於 S3 用戶端名稱，請選擇次要。 4. 將您先前建立的 S3 儲存貯體新增至存放庫。 5. 選擇壓縮快照。 6. 對於「加密」設定，請保留預設值。	AWS 管理員、應用程式 Support

任務	描述	所需技能
驗證新的 Amazon S3 儲存庫。	確保您可以存取 Elastic Cloud 叢集中託管的新存放庫。	AWS 管理員
初始化彈性搜尋服務叢集。	在彈性搜尋服務主控台上，從 S3 快照初始化彈性搜尋服務叢集。 運行以下命令作為 POST。 `*/_close?expand_wildcards=all` `/_snapshot/<your-repo-name>/ <your-snapshot-name>/_restore` `*/_open?expand_wildcards=all`	應用 Support

完成移轉

任務	描述	所需技能
確認快照還原成功。	使用 Kibana 開發工具，執行下列命令。 `GET _cat/indices`	應用支援
Redploy 攝入服務。	將節拍和 Logstash 的端點 Connect 到新的彈性搜索服務端點。	應用支援

測試叢集環境並清理

任務	描述	所需技能
驗證叢集環境。	將現場部署 Elastic 叢集環境遷移到 AWS 後，您可以連線到該環境，並使用自己的使用者接受度測試 (UAT) 工具來驗證新環境。	應用支援
清理資源。	驗證叢集已成功遷移後，請移除用於遷移的 S3 儲存貯體和 IAM 角色。	AWS 管理員

相關資源

彈性參考

- 彈性雲
- AWS 上受管彈性搜尋和基巴納
- 彈性企搜
- 彈性整合
- 彈性可觀測性
- 彈性安全性
- 節拍
- 彈性 APM
- 移轉至索引生命週期管理
- 彈性訂閱
- 觸點彈性

彈性博客文章

- 如何在 AWS 上從自我管理的彈性搜尋遷移到彈性雲端 (部落格文章)
- 移轉至彈性雲端 (部落格文章)

彈性文檔

- 教學課程：使用 SLM 自動備份
- ILM：管理索引生命週期
- 长尾
- 跨叢集複寫 (CCR)
- 內嵌管道
- 執行彈性搜尋 API 要求
- 快照保留

彈性視訊和網路研討

- 彈性雲端移轉
- 彈性雲端：客戶為何要遷移 (網路研討會)

AWS 參考資料

- AWS Marketplace 上的彈性雲端
- AWS 命令列界面
- AWS Direct Connect
- Migration Acceleration Program (MAP)
- Network Load Balancer
- 區域與可用區域
- Amazon Route 53
- Amazon Simple Storage Service
- Amazon S3 Transfer Acceleration
- VPN 連線
- Well-Architected 的框架

其他資訊

如果您打算移轉複雜的工作負載，請參與彈性諮詢服務。如果您有與組態和服務相關的基本問題，請聯絡彈性 Support 團隊。

使用星爆將資料遷移到 AWS 雲端

創建者：安東尼·普拉薩德·特瓦拉治 (AWS)、肖恩·范·斯塔登 (星爆) 和蘇雷什維拉戈尼 (AWS)

環境：生產	技術：分析；資料湖；資料庫	工作負載：所有其他工作
AWS 服務：Amazon EKS		

Summary

Starburst 透過提供企業查詢引擎，將現有資料來源整合到單一存取點中，協助您加快資料遷移到 Amazon Web Services (AWS) 的速度。在完成任何移轉計劃之前，您可以跨多個資料來源執行分析，以取得寶貴的見解。在不中斷 business-as-usual 分析的情況下，您可以使用 Starburst 引擎或專用的擷取、轉換和載入 (ETL) 應用程式來移轉資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 虛擬私有雲 (VPC)
- 亞 Amazon Elastic Kubernetes Service（Amazon EKS）集群
- Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling 組
- 需要移轉的目前系統工作負載清單
- 從 AWS 到現場部署環境的網路連線

架構

參考架構

下列高階架構圖顯示星爆企業在 AWS 雲端中的典型部署：

1. 星爆企業叢集會在您的 AWS 帳戶內執行。
2. 使用者使用輕量型目錄存取通訊協定 (LDAP) 或開放授權 (OAuth) 進行驗證，並直接與 Starburst 叢集互動。

3. 星爆可以連接到多個 AWS 數據源，例如 AWS Glue，Amazon Simple Storage Service (Amazon S3)，Amazon Relational Database Service（Amazon RDS）和 Amazon Redshift。Starburst 針對 AWS 雲端、內部部署或其他雲端環境中的資料來源提供聯合查詢功能。
4. 您可以使用頭盔圖表在 Amazon EKS 集群中啟動星爆企業。
5. 星爆企業使用 Amazon EC2 Auto Scaling 群組和 Amazon EC2 競價型執行個體來優化基礎設施。
6. 星爆企業版會直接連線到您現有的內部部署資料來源，以便即時讀取資料。此外，如果您在此環境中有現有的 Starburst 企業部署，則可以將 AWS 雲端中的新 Starburst 叢集直接連接到此現有叢集。

請注意以下內容：

- 星爆不是一個數據虛擬化平台。它是以 SQL 為基礎的大規模 parallel 處理 (MPP) 查詢引擎，可構成用於分析的整體資料網格策略的基礎。
- 當 Starburst 部署為遷移的一部分時，它可以直接連接到現有的內部部署基礎結構。
- Starburst 提供數個內建的企業和開放原始碼連接器，可促進各種舊系統的連線能力。如需連接器及其功能的完整清單，請參閱 Starburst 企業版使用者指南中的連接器。
- Starburst 可以從內部部署資料來源即時查詢資料。這樣可以防止在遷移資料時中斷一般業務作業。
- 如果您要從現有的現場部署 Starburst 企業部署遷移，則可以使用特殊的連接器星爆星際之門，將 AWS 中的星爆企業叢集直接連接到現場部署叢集。當商業使用者和資料分析師將 AWS 雲端的查詢聯合到現場部署環境時，這會提供額外的效能優勢。

高階程序概觀

您可以使用 Starburst 加快資料移轉專案的速度，因為 Starburst 在移轉資料之前就能對所有資料提供深入分析。下圖顯示使用星爆移轉資料的典型程序。

Roles (角色)

使用 Starburst 完成移轉通常需要下列角色：

- 雲管理員 — 負責使雲資源可用於運行星爆企業應用程序
- 星爆管理員 — 負責安裝、配置、管理和支援星爆應用程式

- 數據工程師 — 負責：
  - 將舊版資料移轉至雲端
  - 建立語義檢視以支援分析
- 解決方案或系統擁有者 — 負責整體解決方案的實作

工具

AWS 服務

- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。
- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，可在 AWS 上執行 Kubernetes，而不需要站立或維護自己的 Kubernetes 控制平面。Kubernetes 是一套開放原始碼系統，用於容器化應用程式的自動化部署、擴展與管理。

其他工具

- 掌舵 — Helm 是 Kubernetes 的套件管理員，可協助您在 Kubernetes 叢集上安裝及管理應用程式。
- 星爆企業 — 星爆企業是以 SQL 為基礎的大規模 parallel 處理 (MPP) 查詢引擎，可構成分析整體資料網格策略的基礎。
- 星爆星際之門 — 星爆星之門將一個星爆企業環境中的目錄和資料來源 (例如現場部署資料中心的叢集) 連結到另一個星爆企業環境中的目錄和資料來源，例如 AWS 雲端中的叢集。

史诗

評估資料

任務	描述	所需技能
識別資料並排定優先順序。	識別您要移動的資料。大型的內部部署舊版系統可以包含您要遷移的核心資料，以及因為合規原因而不想移動或無法移動的資料。從資料清查開始，可協助您排定應優先定位哪些資料。如需詳細資訊，請參	數據工程師，DBA

任務	描述	所需技能
	閱開始使用自動化產品組合探索。	
探索、清查和備份您的資料。	針對您的使用案例驗證資料的品質、數量和相關性。視需要備份或建立資料的快照，並完成資料的目標環境。	數據工程師，DBA

設置星爆企業環境

任務	描述	所需技能
在 AWS 雲端中設定星爆企業。	在編目資料時，請在受管 Amazon EKS 叢集中設定星爆企業。如需詳細資訊，請參閱星爆企業參考文件中的使用 Kubernetes 進行部署。這可讓您在資料移轉過程中進行 business-as-usual 分析。	AWS 管理員、應用程式開發
將星爆 Connect 到數據源。	識別資料並設定星爆企業之後，請將星爆連線至資料來源。星爆直接從數據源讀取數據作為 SQL 查詢。如需詳細資訊，請參閱星爆企業參考文件。	AWS 管理員、應用程式開發

遷移數據

任務	描述	所需技能
建置並執行 ETL 管線。	開始資料移轉程序。此活動可與 business-as-usual 分析同時發生。對於移轉，您可以使用	數據工程師

任務	描述	所需技能
	第三方產品或星爆。Starburst 具有跨不同來源讀取和寫入數據的能力。如需詳細資訊，請參閱星爆企業參考文件。	
驗證資料。	移轉資料之後，請驗證資料，以確保所有必要的資料均已移動且完整無缺。	數據工程師， DevOps 工程師

切成薄片並推出

任務	描述	所需技能
切過的數據。	資料移轉和驗證完成後，您可以切除資料。這涉及更改星爆中的數據連接鏈接。您不必指向內部部署來源，而是指向新的雲端來源並更新語意檢視。如需詳細資訊，請參閱星爆企業參考文件中的連接器。	數據工程師，切換領導
向使用者推出。	資料取用者開始使用移轉的資料來源。這個過程是不可見的分析最終用戶。	資料工程師切換主管

相關資源

AWS Marketplace

- 星爆星系
- 星爆企業
- 星爆數據 JumpStart
- 帶引力彈的星爆企業

星爆文件

- 星爆企業版用戶指南
- 星爆企業參考文件

其他 AWS 文件

- 開始使用自動化產品組合探索 (AWS Prescriptive Guidance)
- 使用 AWS 上的 Starburst 優化雲端基礎設施的成本和效能 (部落格文章)

優化 AWS 上輸入檔案大小的 ETL 擷取

環境：PoC 或試點	技術：分析；資料湖	工作負載：開源
AWS 服務：AWS AWS Glue；Amazon S3		

Summary

此模式說明如何在處理資料之前最佳化檔案大小，在 AWS Glue 上針對大數據和 Apache Spark 工作負載優化擷取、轉換和載入 (ETL) 程序的擷取步驟。使用此模式可以防止或解決小檔案的問題。也就是說，由於檔案的彙總大小，大量的小型檔案會拖慢資料處理速度。例如，數百個檔案只有幾百 KB，每個檔案可能會大幅降低 AWS Glue 任務的資料處理速度。這是因為 AWS Glue 必須在 Amazon 簡單儲存服務 (Amazon S3) 和 YARN (然而另一個資源協商者) 上執行內部清單功能，必須存放大量中繼資料。為了提高數據處理速度，您可以使用分組來啟用 ETL 任務將一組輸入文件讀入單個內存分區中。分區會自動將較小的文件分組在一起。或者，您可以使用自訂程式碼將批次邏輯新增至現有檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一個或多個 AWS 粘合任務
- 一或多個巨量資料或 Apache 星火工作負載
- S3 儲存貯體

架構

以下模式顯示 AWS Glue 任務如何處理不同格式的資料，然後存放在 S3 儲存貯體中，以便瞭解效能。

該圖顯示以下工作流程：

1. AWS Glue 任務會將 CSV、JSON 和鑲木地板格式的小檔案轉換為動態框架。注意：輸入檔案的大小對 AWS Glue 任務的效能有最大的影響。

2. AWS Glue 任務會在 S3 儲存貯體中執行內部清單功能。

工具

- AWS Glue 是全受管的 ETL 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

史诗

使用分組來最佳化讀取期間的 ETL 擷取

任務	描述	所需技能
指定群組大小。	如果您有超過 50,000 個檔案，則依預設會進行分組。但是，您也可以透過在connectionOptions 參數中指定群組大小，對少於 50,000 個檔案使用群組。connectionOptions 參數位於方create_dynamic_frame.from_options 法中。	數據工程師
編寫分組代碼。	使用此create_dynamic_frame 方法建立動態影格。例如： `S3bucket_node1 = glueContext.create_dynamic_frame.from_options( format_options={"multiline": False}, connection_type="s3", format="json",`	數據工程師

任務	描述	所需技能
	`connection_options ={ "paths": ["s3://bucket/prefix/file.json"], "recurse": True, "groupFiles": 'inPartition', "groupSize": 1048576 }, transformation_ctx="S3bucket_node1", )` 注意：用於`groupFiles` 將 Amazon S3 分割區群組中的檔案分組。用於設定`groupSize` 要在記憶體中讀取之群組的目標大小。以`groupSize` 位元組為單位指定 (1048576 = 1 MB)。	
將程式碼新增至工作流程。	在 AWS Glue 中將分組代碼新增至您的任務工作流程。	數據工程師

使用自訂邏輯最佳化 ETL 擷取

任務	描述	所需技能
選擇語言和處理平台。	選擇針對您的使用案例量身打造的指令碼語言和處理平台。	雲端架構師
撰寫程式碼。	編寫自定義邏輯以將文件批處理在一起。	雲端架構師

任務	描述	所需技能
將程式碼新增至工作流程。	在 AWS Glue 中將程式碼新增至您的任務工作流程。這可讓您在每次執行工作時套用自訂邏輯。	數據工程師

轉換後寫入資料時重新分割

任務	描述	所需技能
分析沖銷模式。	瞭解下游應用程式將如何使用您撰寫的資料。例如，如果他們每天查詢資料，而您只對每個區域進行資料分割，或者輸出檔案非常小，例如每個檔案 2.5 KB，則這並非最適合使用。	DBA
在寫入之前重新分區數據。	根據處理期間 (根據處理邏輯) 和處理後 (根據消耗) 的聯結或查詢重新分割。例如，根據位元組大小重新分割，例如`.repartition(100000)` ，或根據資料行重新分割，例如. `.repartition("column_name")`	數據工程師

相關資源

- 讀取較大群組中的輸入檔
- 監控 AWS AWS Glue
- 使用 Amazon CloudWatch 指標監控 AWS AWS Glue
- 任務監控與偵錯
- 開始使用 AWS Glue 上的無伺服器 ETL

其他資訊

決定檔案大小

沒有直接的方法可以確定文件大小是太大還是太小。檔案大小對處理效能的影響取決於叢集的配置。在核心 Hadoop 中，我們建議您使用 128 MB 或 256 MB 的檔案，以充分利用區塊大小。

對於 AWS Glue 上的大多數文字檔工作負載，我們建議使用 5-10 個 DPU 叢集的檔案大小介於 100 MB 到 1 GB 之間。若要找出輸入檔案的最佳大小，請監控 AWS Glue 任務的預處理區段，然後檢查任務的 CPU 使用率和記憶體使用率。

其他考量

如果早期 ETL 階段的效能是瓶頸，請在處理之前考慮將資料檔分組或合併。如果您擁有檔案產生程序的完整控制權，在原始資料傳送到 AWS 之前，彙總來源系統本身上的資料點會更有效率。

使用 AWS Step Functions 透過驗證、轉換和分割協調 ETL 管道

創建者：桑迪普甘加帕帝伊 (AWS)

代碼存儲庫：aws-step-functions-etl-流水線模式	環境：生產	技術：分析；大數據；資料湖 DevOps；無伺服器
AWS 服務：Amazon Athena；AWS AWS Glue；AWS Lambda；AWS Step Functions		

Summary

此模式說明如何建置無伺服器擷取、轉換和載入 (ETL) 管道，以驗證、轉換、壓縮和分割大型 CSV 資料集，以達到效能和成本最佳化。該管道由 AWS Step Functions 協調，包括錯誤處理、自動重試和使用者通知功能。

將 CSV 檔案上傳到 Amazon Simple Storage Service (Amazon S3) 儲存貯體來源資料夾時，ETL 管道開始執行。管線會驗證來源 CSV 檔案的內容和結構描述，將 CSV 檔案轉換為壓縮的 Apache Parquet 格式，依年、月和日分割資料集，並將其儲存在分析工具的獨立資料夾中以供處理。

自動化此模式的程式碼可在 GitHub具有 AWS Step Functions 儲存庫的 ETL 管道中取得。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 使用 AWS 帳戶安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)，以便您可以透過部署 AWS CloudFormation 堆疊來建立 AWS 資源。建議使用 AWS CLI 第 2 版。如需安裝指示，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI 第 2 版。如需 AWS CLI 組態指示，請參閱 AWS CLI 文件中的組態和登入資料檔案設定。
- Amazon S3 儲存貯體。
- 具有正確結構描述的 CSV 資料集。(此模式隨附的程式碼存放庫提供範例 CSV 檔案，其中包含您可以使用的正確結構描述和資料類型。)
- 支援與 AWS 管理主控台搭配使用的網頁瀏覽器。請參閱支援的瀏覽器清單。)

- AWS Glue 主控台存取權。
- AWS Step Functions 主控台存取。

限制

- 在 AWS Step Functions 中，保留歷程記錄的最大限制為 90 天。如需詳細資訊，請參閱 AWS Step Functions 文件中的標準工作流程的配額和配額。

產品版本

- Python 3.11 適用於 AWS Lambda
- AWS AWS Glue 2.0 版

架構

圖中所示的工作流程包含下列高階步驟：

1. 使用者將 CSV 檔案上傳到 Amazon S3 中的來源資料夾。
2. Amazon S3 通知事件會啟動 AWS Lambda 函數，以啟動 Step Functions 數狀態機器。
3. Lambda 函數會驗證原始 CSV 檔案的結構描述和資料類型。
4. 根據驗證結果：
   a. 如果來源檔案驗證成功，檔案會移至 stage 資料夾以供進一步處理。
   b. 如果驗證失敗，檔案會移至錯誤資料夾，並透過 Amazon 簡單通知服務 (Amazon SNS) 傳送錯誤通知。
5. AWS Glue 爬行者程式會從 Amazon S3 中的階段資料夾建立原始檔案的結構描述。
6. AWS Glue 任務會將原始檔案轉換、壓縮和分割成實木複合地板格式。
7. AWS Glue 任務也會將檔案移至 Amazon S3 中的轉換資料夾。
8. AWS Glue 爬行者程式會從轉換後的檔案建立結構描述。產生的結構描述可供任何分析工作使用。您也可以使用 Amazon Athena 執行臨時查詢。
9. 如果配管完成時沒有發生錯誤，則結構描述檔案會移至封存資料夾。如果遇到任何錯誤，則會將檔案移至錯誤資料夾。
10. Amazon SNS 會根據管道完成狀態傳送通知，指出成功或失敗。

此模式中使用的所有 AWS 資源都是無伺服器的。沒有要管理的伺服器。

工具

AWS 服務

- AWS Glue — AWS Glue 是全受管 ETL 服務，可讓客戶輕鬆準備和載入資料以進行分析。
- AWS 步驟函數 — AWS Step Functions 是一種無伺服器協調服務，可讓您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。透過 AWS Step Functions 圖形主控台，您可以將應用程式的工作流程視為一系列事件驅動的步驟。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種高可用性、耐用、安全且全受管的 Pub/sub 簡訊服務，可讓您分離微型服務、分散式系統和無伺服器應用程式。
- AWS Lambda — AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。

Code

此模式的程式碼可在 GitHub具有 AWS Step Functions 儲存庫的 ETL 管道中取得。代碼存儲庫包含以下文件和文件夾：

- `template.yml`— 使用 AWS 步驟函數建立 ETL 管道的 AWS CloudFormation 範本。
- `parameter.json`— 包含所有參數和參數值。您可以更新此檔案以變更參數值，如 Epics 一節中所述。
- `myLayer/python`資料夾 — 包含為此專案建立所需 AWS Lambda 層所需的 Python 套件。
- `lambda`資料夾 — 包含下列 Lambda 函數：
  - `move_file.py`— 將來源資料集移至封存、轉換或錯誤資料夾。
  - `check_crawler.py`— 在傳送失敗訊息之前，根據`RETRYLIMIT` 環境變數設定的次數，檢查 AWS Glue 爬行程式的狀態。
  - `start_crawler.py`— 啟動 AWS Glue 爬蟲程式。
  - `start_step_function.py`— 啟動 AWS Step Functions。
  - `start_codebuild.py`— 啟動 AWS CodeBuild 專案。
  - `validation.py`— 驗證輸入原始資料集。
  - `s3object.py`— 在 S3 儲存貯體內建立所需的目錄結構。

- `notification.py`— 在管道結束時傳送成功或錯誤通知。

若要使用範例程式碼，請依照 Epics 一節中的指示操作。

史诗

準備來源檔案

任務	描述	所需技能
複製範例程式碼儲存庫。	1. 使用 AWS Step Functions 存放庫開啟 ETL 管道。 2. 在檔案清單上方的主儲存庫頁面上選擇「程式碼」，然後複製「使用 HTTPS 複製」下列出的 URL。 3. 將工作目錄變更為要儲存範例檔案的位置。 4. 在終端機或命令提示字元中，輸入命令： `git clone <repoURL>` 其中`<repoURL>` 指的是您在步驟 2 中複製的 URL。	開發人員
更新參數值。	在存放庫的本機副本中，編輯`parameter.json` 檔案並更新預設參數值，如下所示： • `pS3BucketName` — 用於存放資料集的 S3 儲存貯體名稱。範本會為您建立此值區。儲存貯體名稱必須是全域唯一的。	開發人員

任務	描述	所需技能
	• pSourceFolder – S3 儲存貯體內將用來上傳來源 CSV 檔案的資料夾名稱。 • pStageFolder – S3 儲存貯體內的資料夾名稱，此資料夾在程序期間將用作暫存區域。 • pTransformFolder – S3 儲存貯體內的資料夾名稱，用於存放已轉換和分割的資料集。 • pErrorFolder – 如果無法驗證來源 CSV 檔案，將會移至 S3 儲存貯體內的資料夾。 • pArchiveFolder – S3 儲存貯體內將用來存檔來源 CSV 檔案的資料夾名稱。 • pEmailforNotification – 用於接收成功/錯誤通知的有效電子郵件地址。 • pPrefix– 將在 AWS Glue 編目程式名稱中使用的前置字串。 • pDatasetSchema – 將驗證來源檔案的資料集結構描述。地獄犬 Python 包用於源數據集驗證。如需詳細資訊，請參閱地獄犬網站。	

任務	描述	所需技能
將原始程式碼上傳至 S3 儲存貯體。	在部署自動化 ETL 管道的 CloudFormation 範本之前，您必須先封裝 CloudFormation 範本的來源檔案，並將它們上傳到 S3 儲存貯體。若要這麼做，請使用預先設定的設定檔執行下列 AWS CLI 命令： `aws cloudformation package --template-file template.yml --s3-bucket <bucket_name> --output-template-file packaged.template --profile <profile_name>` 其中： • `<bucket_name>` 是您要在其中部署堆疊的 AWS 區域中現有 S3 儲存貯體的名稱。此存儲桶用於存儲 CloudFormation 模板的源代碼包。 • `<profile_name>` 是您在設定 AWS CLI 時預先設定的有效 AWS CLI 設定檔。	開發人員

建立 堆疊。

任務	描述	所需技能
部署 CloudFormation 範本。	若要部署 CloudFormation 範本，請執行下列 AWS CLI 命令： `aws cloudformation deploy --stack-name <stack_name> --template-file packaged.template --parameter-overrides file://parameter.json --capabilities CAPABILITY_IAM --profile <profile_name>` 其中： • `<stack_name>` 是 CloudFormation 堆疊的唯一識別碼。 • `<profile-name>` 是您預先設定的 AWS CLI 設定檔。	開發人員
檢查進度。	在 AWS 主 CloudFormation 控台上，檢查堆疊開發的進度。當狀態為時`CREATE_COMPLETE` ，堆疊已成功部署。	開發人員
請記下 AWS Glue 資料庫名稱。	堆疊的 [輸出] 索引標籤會顯示 AWS Glue 資料庫的名稱。金鑰名稱為`GlueDBOutput` 。	開發人員

測試管道

任務	描述	所需技能
啟動 ETL 管線。	1. 導覽至 S3 儲存貯體內的來源資料夾 (`source`或您在`parameter.json` 檔案中設定的資料夾名稱)。 2. 將範例 CSV 檔案上傳至此資料夾。（代碼存儲庫提供了一個名為的示例文件`Sample_Bank_Transaction_Raw_Dataset.csv` ，您可以使用它。）上傳文件將通過 Step Functions 啟動 ETL 管道。 3. 在 Step Functions 主控台上，檢查 ETL 管線狀態。	開發人員
檢查已分割的資料集。	當 ETL 管道完成時，請確認分區資料集可在 Amazon S3 轉換資料夾 (`transform` 或您在`parameter.json` 檔案中設定的資料夾名稱) 中使用。	開發人員
檢查已分割的 AWS Glue 資料庫。	1. 在 AWS Glue 主控台上，選取堆疊建立的 AWS Glue 資料庫 (這是您在上一篇史詩中記下的資料庫)。 2. 確認 AWS Glue 資料型錄中提供分區資料表。	開發人員
執行查詢。	(選擇性) 使用 Amazon Athena 在分區和轉換的資料庫上執行臨機操作查詢。如需指示，請參閱 AWS 文件中的使用	数据库分析

任務	描述	所需技能
	Amazon Athena 執行 SQL 查詢。	

故障診斷

問題	解決方案
適用於 AWS Glue 任務和爬蟲的 AWS Identity and Access Management (IAM) 許可	如果您進一步自訂 AWS Glue 任務或爬蟲程式，請務必在 AWS Glue 任務使用的 IAM 角色中授予適當的 IAM 許可，或向 AWS Lake Formation 提供資料許可。如需詳細資訊，請參閱 AWS 文件。

相關資源

AWS 服務文件

- AWS Step Functions
- AWS Glue
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

其他資訊

下圖顯示 Step Functions Inspector 面板中成功 ETL 管道的 AWS 步驟函數工作流程。

下圖顯示由於輸入驗證錯誤而失敗的 ETL 管道的 AWS Step Functions 數工作流程，從「步驟函數 Inspector」面板。

使用 Amazon Redshift ML 執行進階分析

環境：PoC 或試點	技術：分析、機器學習與人工智慧	工作負載：所有其他工作
AWS 服務：Amazon Redshift; Amazon SageMaker		

Summary

在 Amazon Web Services (AWS) 雲端上，您可以使用 Amazon Redshift 機器學習 (亞馬遜 Redshift ML) 對存放在 Amazon Redshift 叢集或亞馬遜簡單儲存服務 (Amazon S3) 上的資料執行 ML 分析。Amazon Redshift ML 支援受監督式學習，這通常用於進階分析。Amazon Redshift ML 的使用案例包括收入預測、信用卡詐騙偵測和客戶生命週期價值 (CLV) 或客戶流失預測。

Amazon Redshift ML 可讓資料庫使用者使用標準 SQL 命令輕鬆建立、訓練和部署機器學習模型。Amazon Redshift ML 使用 Amazon SageMaker Autopilot 自動輔助駕駛功能，根據您的資料自動訓練和調整最佳機器學習模型以進行分類或回歸，同時保有控制權和能見度。

Amazon Redshift，Amazon S3 和亞馬遜之間的所有互動 SageMaker 都被抽象化並自動化。ML 模型經過訓練和部署之後，它就會在 Amazon Redshift 中以使用者定義函數 (UDF) 的形式提供，並可用於 SQL 查詢。

此模式與 AWS 部落格中的 Amazon Redshift ML 使用 SQL 在 Amazon Redshift 中建立、訓練和部署機器學習模型相輔相成，以及使用入門資源中心的 Amazon SageMaker 教學課程建立、訓練和部署機器學習模型。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon Redshift 表中的現有數據

技能

- 熟悉 Amazon Redshift ML 使用的術語和概念，包括機器學習、訓練和預測。如需這方面的詳細資訊，請參閱 Amazon 機器學習 (Amazon ML) 文件中的訓練機器學習模型。
- 具備 Amazon Redshift 使用者設定、存取管理和標準 SQL 語法的經驗。如需這方面的詳細資訊，請參閱 Amazon Redshift 文件中的開始使用 Amazon Redshift。
- Amazon S3 和 AWS Identity and Access Management (IAM) 的知識和經驗。
- 在 AWS Command Line Interface (AWS CLI) (AWS CLI) 中執行命令的經驗也很有幫助，但不是必需的。

限制

- Amazon Redshift 叢集和 S3 儲存貯體必須位於相同的 AWS 區域。
- 這種模式的方法僅支持監督學習模型，例如回歸，二進制分類和多類分類。

架構

下列步驟說明 Amazon Redshift ML 如何搭配 SageMaker 建置、訓練和部署機器學習模型：

1. Amazon Redshift 會將訓練資料匯出到 S3 儲存貯體。
2. SageMaker 自動輔助駕駛會自動預先處理訓練資料。
3. 在叫用`CREATE MODEL`陳述式之後，Amazon Redshift ML 會用 SageMaker 於訓練。
4. SageMaker Autopilot 會搜尋並建議 ML 演算法和最佳化超參數，以最佳化評估指標。
5. Amazon Redshift ML 將輸出 ML 模型註冊為 Amazon Redshift 集群中的 SQL 函數。
6. ML 模型的函數可以在 SQL 陳述式中使用。

技術, 堆棧

- Amazon Redshift
- SageMaker
- Amazon S3

工具

- Amazon Redshift — Amazon Redshift 是一種企業級、PB 級規模的全受管資料倉儲服務。

- 亞馬遜 Redshift ML — Amazon Redshift 機器學習 (Amazon Redshift ML) 是一種強大的雲端服務，可讓各種技能等級的分析師和資料科學家輕鬆使用機器學習技術。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- Amazon SageMaker — SageMaker 是一個全受管的 ML 服務。
- Amazon SageMaker 自動輔助駕駛 — SageMaker 自動輔助駕駛是一套功能集，可自動執行自動化機器學習 (AutoML) 程序的關鍵任務。

Code

您可以使用下列程式碼在 Amazon Redshift 中建立受監管的 ML 模型：

```
“CREATE MODEL customer_churn_auto_model
FROM (SELECT state,
             account_length,
             area_code,
             total_charge/account_length AS average_daily_spend,
             cust_serv_calls/account_length AS average_daily_cases,
             churn
      FROM customer_activity
      WHERE  record_date < '2020-01-01'
     )
TARGET churn
FUNCTION ml_fn_customer_churn_auto
IAM_ROLE 'arn:aws:iam::XXXXXXXXXXXX:role/Redshift-ML'
SETTINGS (
  S3_BUCKET 'your-bucket'
);”)
```

注意：該SELECT州可以參考 Amazon Redshift 常規表，Amazon Redshift Spectrum 外部表，或兩者兼而有之。

史诗

準備訓練和測試資料集

任務	描述	所需技能
準備訓練和測試資料集。	登入 AWS 管理主控台並開啟 Amazon 主 SageMaker 控台。	資料科學家

任務	描述	所需技能
	遵循建置、訓練和部署機器學習模型教學課程中的指示，建立包含標籤欄 (受監督訓練) 且沒有標題的 .csv 或 Apache Parquet 檔案。 注意：我們建議您將原始資料集隨機排列並分割為模型訓練的訓練集 (70%)，以及用於模型效能評估的測試集 (30%)。	

準備和設定技術堆疊

任務	描述	所需技能
建立和設定 Amazon Redshift 叢集。	在 Amazon Redshift 主控台上，根據您的需求建立叢集。如需這方面的詳細資訊，請參閱 Amazon Redshift 文件中的建立叢集。 重要事項：必須使用`SQL_PREVIEW` 維護追蹤建立 Amazon Redshift 叢集。如需有關預覽曲目的詳細資訊，請參閱 Amazon Redshift 文件中的選擇叢集維護追蹤。	DBA、雲端架構師
建立 S3 儲存貯體以存放訓練資料和模型成品。	在 Amazon S3 主控台上，為訓練和測試資料建立 S3 儲存貯體。如需建立 S3 儲存貯體的詳細資訊，請參閱從 AWS 快速入門建立 S3 儲存貯體。	DBA、雲端架構師

任務	描述	所需技能
	重要事項：請確定您的 Amazon Redshift 叢集和 S3 儲存貯體位於同一個區域。	
建立 IAM 政策並將其附加到 Amazon Redshift 叢集。	建立 IAM 政策以允許 Amazon Redshift 叢集存取 SageMaker 和 Amazon S3。如需指示和步驟，請參閱亞馬遜紅移文件中的叢集設定以使用 Amazon Redshift ML。	DBA、雲端架構師
允許 Amazon Redshift 使用者和群組存取結構描述和資料表。	授與許可以允許 Amazon Redshift 中的使用者和群組存取內部和外部結構描述和表格。如需步驟和指示，請參閱 Amazon Redshift 文件中的管理許可和擁有權。	DBA

在 Amazon Redshift 中創建和訓練 ML 模型

任務	描述	所需技能
在 Amazon Redshift 中創建和訓練 ML 模型。	在 Amazon Redshift ML 中建立和訓練您的機器學習模型。如需詳細資訊，請參閱 Amazon Redshift 文件中的`CREATE MODEL`陳述式。	資料科學家開發人員

在 Amazon Redshift 中執行批次推論和預測

任務	描述	所需技能
使用產生的 ML 模型函數執行推論。	如需使用產生的 ML 模型函數執行推論的詳細資訊，請參閱	資料科學家、商業智慧使用者

任務	描述	所需技能
	Amazon Redshift 文件中的預測。	

相關資源

準備訓練和測試資料集

- 使用 Amazon 建置、訓練和部署機器學習模型 SageMaker

準備和設定技術堆疊

- 創建一個 Amazon Redshift 集群
- 選擇 Amazon Redshift 群集維護跟踪
- 建立 S3 儲存貯體
- 設置 Amazon Redshift 集群以使用 Amazon Redshift ML
- 管理 Amazon Redshift 中的許可和所有權

在 Amazon Redshift 中創建和訓練 ML 模型

- 在 Amazon Redshift 中創建模型聲明

在 Amazon Redshift 中執行批次推論和預測

- Amazon Redshift 中的預測

其他資源

- 開始使用 Amazon Redshift ML
- 使用 SQL 搭配亞馬遜紅移 ML 在 Amazon Redshift 中建立、訓練和部署機器學習模型
- Amazon Redshift 合作夥伴
- AWS 機器學習能力合作夥伴

使用 Athena 存取、查詢和加入 Amazon DynamoDB 資料表

創建者：穆努爾·阿爾馬蒙（AWS）

環境：生產	技術：分析；資料庫；無伺服器；大數據	AWS 服務：Amazon Athena；Amazon DynamoDB；AWS Lambda；Amazon S3

Summary

此模式說明如何使用 Amazon Athena DynamoDB 連接器設定亞馬遜雅典娜和亞馬遜 DynamoDB 之間的連接。連接器會使用 AWS Lambda 函數來查詢動態資料 B 中的資料。您不需要編寫任何代碼來設置連接。建立連線之後，您可以使用 Athena 聯合查詢執行來自 Athena 的 SQL 命令，快速存取和分析 DynamoDB 表格。您也可以將一或多個 DynamoDB 表格彼此聯結，或聯結至其他資料來源，例如 Amazon Redshift 或 Amazon Aurora。

先決條件和限制

先決條件

- 具有管理 DynamoDB 表、Athena 資料來源、Lambda 和 AWS 身分和存取管理 (IAM) 角色許可的有效 AWS 帳戶
- 亞馬遜簡易儲存服務 (Amazon S3) 儲存貯體，Athena 可以存放查詢結果
- 一個 S3 儲存貯體，其中 Athena DynamoDB 連接器可以在短期內儲存資料
- 支援 Athena 引擎第 2 版的 AWS 區域
- 存取 Athena 和所需 S3 儲存貯體的 IAM 許可
- Amazon Athena DynamoDB 連接器, 已安裝

限制

查詢 DynamoDB 資料表需要支付費用。資料表大小超過幾 GB (GB) 可能會產生很高的成本。我們建議您在執行任何全表掃描操作之前考慮成本。如需詳細資訊，請參閱 Amazon DynamoDB 定價。為了降低成本並達到高效能，建議您一律在查詢中使用 LIMIT (例如`SELECT * FROM table1 LIMIT 10`)。此外，在生產環境中執行 JOIN 或 GROUP BY 查詢之前，請考慮資料表的大小。如果您的資料表太大，請考慮其他選項，例如將表格遷移到 Amazon S3。

架構

下圖顯示使用者如何在 Athena 的 DynamoDB 資料表上執行 SQL 查詢。

該圖顯示以下工作流程：

1. 若要查詢 DynamoDB 資料表，使用者會從 Athena 執行 SQL 查詢。
2. Athena 啟動一個 Lambda 函數。
3. Lambda 函數會在 DynamoDB 資料表中查詢要求的資料。
4. DynamoDB 要求的資料傳回至 Lambda 函數。然後，函數會透過 Athena 將查詢結果傳送給使用者。
5. Lambda 函數會將資料存放在 S3 儲存貯體中。

技術, 堆

- Amazon Athena
- Amazon DynamoDB
- Amazon S3
- AWS Lambda

工具

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- Amazon Athena DynamoDB 連接器是一種 AWS 工具，可讓 Athena 與 DynamoDB 連線，並使用 SQL 查詢存取您的資料表。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

史诗

建立範 DynamoDB 料表

任務	描述	所需技能
建立第一個範例資料表。	1. 登入 AWS 管理主控台，然後開啟 DynamoDB 主控台。 2. 選擇 建立資料表 。 3. 對於「表格名稱」，請輸入資料表 1。 4. 針對分割區索引鍵，輸入 PK 1。 5. 針對「排序」索引鍵，輸入 SK1。 6. 在 [表格設定] 區段中，選擇 [自訂設定]。 7. 在 [資料表類別] 區段中，選擇 [DynamoDB 標準]。 8. 在 [讀取/寫入容量設定] 區段中，針對 [容量模式] 選擇 [隨選 9. 在「靜態加密」區段中，選擇「Amazon DynamoDB 擁有者」。 10選擇 建立資料表 。	開發人員
將樣本數據插入到第一個表中。	1. 開啟 DynamoDB 主控台。 2. 在導覽窗格中，選擇 [表格]，然後在 [名稱] 欄中選擇您的表格。 3. 選擇「操作」，然后選擇「創建物件」。 4. 選擇 [JSON 檢視]。	開發人員

任務	描述	所需技能
	5. 在屬性編輯器的標題列中，關閉檢視 DynamoDB JSON。 6. 在「屬性」編輯器中，逐一輸入下列範例資料： `{ "PK1": "1234", "SK1": "info", "Salary": "5000" }` `{ "PK1": "1235", "SK1": "info", "Salary": "5200" }`	

任務	描述	所需技能
建立第二個範例資料表。	1. 開啟 DynamoDB 主控台。 2. 選擇 建立資料表 。 3. 對於「表格名稱」，輸入可表格 2。 4. 針對分割區索引鍵，輸入 PK2。 5. 針對「排序」鍵，輸入 SK2。 6. 在 [表格設定] 區段中，選擇 [自訂設定]。 7. 在 [資料表類別] 區段中，選擇 [DynamoDB 標準]。 8. 在 [讀取/寫入容量設定] 區段中，針對 [容量模式] 選擇 [隨選 9. 在「靜態加密」區段中，選擇「Amazon DynamoDB 擁有者」。 10選擇 建立資料表 。	開發人員

任務	描述	所需技能
將樣本數據插入到第二個表中。	1. 開啟 DynamoDB 主控台。 2. 在導覽窗格中，選擇 [表格]，然後在 [名稱] 欄中選擇您的表格。 3. 選擇「操作」，然后選擇「創建物件」。 4. 在屬性編輯器的標題列中，關閉檢視 DynamoDB JSON。 5. 在「屬性」編輯器中，逐一輸入下列範例資料： `{ "PK2": "1234", "SK2": "bonus", "Bonus": "500" }` `{ "PK2": "1235", "SK2": "bonus", "Bonus": "1000" }`	開發人員

在 Athena 中為 DynamoDB 建立資料來源

任務	描述	所需技能
設定資料來源連接器。	為 DynamoDB 建立資料來源，然後建立 Lambda 函數以連接至該資料來源。	開發人員

任務	描述	所需技能
	1. 登入 AWS 管理主控台並開啟 Athena 主控台。 2. 在導覽窗格中，選擇 [資料來源]，然後選擇 [建立資料來源]。 3. 選擇 Amazon DynamoDB 資料來源，然後選擇 [下一步]。 4. 在 [資料來源詳細資料] 區段中，對於資料來源名稱，輸入 TestDynamoDB。 5. 在「連線詳細資料」區段中，選取已部署的 Lambda 函數，或者如果沒有可用於此模式的 Lambda 函數，請選擇「建立 Lambda 函數」。注意：如需建立 Lambda 函數的詳細資訊，請參閱 L ambda 開發人員指南中的入門使用 Lambda。 6. (選擇性) 如果您選擇建立 Lambda 函數，則必須先設定 Java 應用程式包含的 AWS CloudFormation 範本，然後再部署該堆疊。範本包括 ApplicationName、SpillBucket AthenaCatalogName、和其他應用程式設定。備註：部署此 Java 應用程式之後，堆疊會建立 Lambda 函數，讓 Athena 能夠與 DynamoDB 通訊。這使您的表可以通過 SQL 命令訪問。	

任務	描述	所需技能
	7. 部署您的 Lambda 函數。 8. 選擇下一步。	
確認 Lambda 函數可以存取 S3 溢出儲存貯體。	1. 開啟 Lambda 主控台。 2. 在導覽窗格中，選擇 [函數]，然後選擇您先前建立的函數。 3. 選擇 Configuration (組態) 索引標籤。 4. 在左窗格中，選擇 [環境變數]，然後確認機碼的值為`spill_bucket` 。 5. 在左窗格中，選擇 [權限]，然後在 [執行角色] 區段中選擇附加的 IAM 角色。附註：系統會將您導向至 IAM 主控台中附加至 Lambda 函數的 IAM 角色。 6. 確認您擁有`spill_bucket` 值區的寫入權限。 如果您遇到錯誤，請參閱此模式中的其他資訊一節以取得指引。	開發人員

從 Athena 存取 DynamoDB 資料表

任務	描述	所需技能
查詢動 DynamoDB 料表。	1. 登入 AWS 管理主控台並開啟 Athena 主控台。	開發人員

任務	描述	所需技能
	2. 在導覽窗格中，選擇 [資料來源]，然後選擇 [建立資料來源]。 3. 在導覽窗格中，選擇 Query Editor (查詢編輯器)。 4. 在 [編輯器] 索引標籤的 [資料] 區段中，對於 [資料來源]，選擇資料來源的資料來源。 5. 如需資料庫，請選擇您的資料庫。 6. 針對「查詢 1」，輸入下列查詢：`SELECT * FROM dydbtable1 t1;` 7. 選擇 [執行]，然後確認表格中的輸出。 8. 針對「查詢 2」，輸入下列查詢：`SELECT * FROM dydbtable2 t2;` 9. 選擇 [執行]，然後確認表格中的輸出。	

任務	描述	所需技能
加入兩個 DynamoDB 資料表。	DynamoDB 是一個 NoSQL 資料存放區，不支援 SQL 聯結作業。因此，您必須對兩個 DynamoDB 表格執行聯結作業： 1. 選擇加號圖示以建立另一個查詢。 2. 針對「查詢 3」，輸入下列查詢： `SELECT pk1, salary, bonus FROM dydbtable1 t1 JOIN dydbtable2 t2 ON t1.pk1 = t2.pk2;`	開發人員

相關資源

- Amazon Athena 連接器 (AWS 實驗室)
- 使用 Amazon Athena 的新聯合查詢 (AWS 大數據部落格) 查詢任何資料來源
- Athena 引擎版本參考 (Athena 使用者指南)
- 使用 AWS Glue 和亞馬遜雅典娜 (AWS 資料庫部落格) 簡化 Amazon DynamoDB 資料擷取和分析

其他資訊

如果您在 Athena 中以`{bucket_name}/folder_name/`格式執行查詢，則可能會收到下列錯誤訊息：`spill_bucket`

```
"GENERIC_USER_ERROR: Encountered an exception[java.lang.RuntimeException] from your
 LambdaFunction[arn:aws:lambda:us-east-1:xxxxxx:function:testdynamodb] executed in
 context[retrieving meta-data] with message[You do NOT own the spill bucket with the
 name: s3://test-bucket-dynamodbconnector/athena_dynamodb_spill_data/]
```

```
This query ran against the "default" database, unless qualified by the query. Please
 post the error message on our forum  or contact customer support with Query Id:
 [query-id]"
```

若要解決此錯誤，請將 Lambda 函數的環境變數更新spill_bucket為{bucket_name_only}，然後將儲存貯體寫入存取權限的下列 Lambda IAM 政策更新為：

```
{
             "Action": [
                 "s3:GetObject",
                 "s3:ListBucket",
                 "s3:GetBucketLocation",
                 "s3:GetObjectVersion",
                 "s3:PutObject",
                 "s3:PutObjectAcl",
                 "s3:GetLifecycleConfiguration",
                 "s3:PutLifecycleConfiguration",
                 "s3:DeleteObject"
             ],
             "Resource": [
                 "arn:aws:s3:::spill_bucket",
                 "arn:aws:s3:::spill_bucket/*"
             ],
             "Effect": "Allow"
         }
```

或者，您可以移除先前建立的 Athena 資料來源連接器，然後僅使用{bucket_name}於重新建立spill_bucket。

設定最小的可行資料空間，以便在組織之間共用資料

創建者：拉米·赫奇尼（想想），伊斯梅爾·阿卜杜勒勞伊（Think-It），馬爾特·加塞林（Think-It），豪爾赫·埃爾南德斯蘇亞雷斯（AWS）和邁克爾·米勒（AWS）

環境：PoC 或試點	技術：分析；容器與微服務；資料湖；資料庫；基礎架構	工作負載：開源
AWS 服務：Amazon Aurora; AWS Certificate Manager (ACM); AWS CloudFormation; Amazon EC2; Amazon EFS; Amazon EKS; Elastic Load Balancing (ELB); Amazon RDS; Amazon S3; AWS Systems Manager		

Summary

數據空間是用於數據交換的聯合網絡，以信任和控制數據作為核心原則。透過提供符合成本效益且與技術無關的解決方案，讓組織能夠大規模地共用、交換和協作資料。

數據空間有可能通過使用涉及所有相關利益相關者的 end-to-end 方法來解決數據驅動的問題，以顯著推動可持續發展的 future 努力。

此模式會引導您瞭解兩家公司如何使用 Amazon Web Services (AWS) 上的資料空間技術來推動碳排放和減少策略向前發展的範例。在這種情況下，X 公司提供了 Y 公司消耗的碳排放數據。如需下列資料空間規格詳細資訊，請參閱「其他資訊」一節：

- 參加者
- 商業案例
- 数据空间权威
- 資料空間元件
- 資料空間服務
- 要交換的數據

- 資料模型
- 氣管-X EDC 連接器

該陣列包括以下步驟：

- 部署執行兩個參與者的基本資料空間所需的基礎結構 AWS。
- 以安全的方式使用連接器交換碳排放強度資料。

此模式會部署將透過 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 託管資料空間連接器及其服務的 Kubernetes 叢集。

Eclipse 資料空間元件 (EDC) 控制平面和資料平面都部署在 Amazon EKS 上。Tractus-X 掌舵官方圖表將 PostgreSQL 和保管庫服務作為依賴項部署。 HashiCorp

此外，身分識別服務部署在 Amazon Elastic Compute Cloud (Amazon EC2) 上，以複寫最小可行資料空間 (MVDS) 的真實案例。

先決條件和限制

先決條件

- 在您選擇的基礎架構中部署活 AWS 帳戶 動 AWS 區域
- 具有 Amazon S3 存取權的 AWS Identity and Access Management (IAM) 使用者，該使用者將暫時作為技術使用者使用 (嵌入式設計中心連接器目前不支援使用角色。 我們建議您專門為此示範建立一個 IAM 使用者，並且此使用者將擁有與其相關聯的有限許可。)
- AWS Command Line Interface （ AWS CLI ）在您選擇的安裝和配置 AWS 區域
- AWS 安全認證
- 在您的工作站上顯示
- Git 在你的工作站上
- 庫貝克特爾
- 頭盔
- 郵递员
- 一個 AWS Certificate Manager (ACM) SSL/TLS 憑證
- 將指向 Application Load Balancer 的 DNS 名稱 (ACM 憑證必須涵蓋 DNS 名稱)

- HashiCorp 保管箱 (如需使用管理密碼 AWS Secrets Manager 的相關資訊，請參閱其他資訊一節。)

產品版本

- AWS CLI 版本 2+
- 郵遞員集合 2.1 版

限制

- 連接器選擇-此部署使用 EDC 型連接器。但是，請務必考慮嵌入式設計中心和 FIWARE True 連接器的優勢和功能，以做出符合部署特定需求的明智決策。
- 嵌入式設計中心連接器構建-選擇的部署解決方案依賴 Tractus-X EDC 連接器頭盔圖，這是一個完善且經過廣泛測試的部署選項。使用此圖表的決定取決於它的常見用法以及在提供的組建中包含基本擴充功能。雖然 PostgreSQL 和文件 HashiCorp 庫是預設元件，但您可以根據需要彈性自訂自己的連接器組建。
- 私有叢集存取-對已部署 EKS 叢集的存取僅限於私有通道。與叢集的互動僅透過使用`kubectl`和 IAM 來執行。您可以使用負載平衡器和網域名稱來啟用叢集資源的公用存取權，這些平衡器和網域名稱必須選擇性地實作，才能將特定服務公開給更廣泛的網路。但是，我們不建議提供公共訪問權限。
- 安全性焦點-強調將安全性組態抽象成預設規格，以便您可以專注於 EDC 連接器資料交換所涉及的步驟。雖然會維護預設安全性設定，但必須先啟用安全通訊，然後再將叢集公開給公用網路。此預防措施可確保安全資料處理的穩健基礎。
- 基礎設施成本-基礎設施成本的估計可以使用 AWS Pricing Calculator. 一個簡單的計算表明，部署的基礎設施每月成本可以高達 162.92 美元。

架構

MVDS 架構由兩個虛擬私有雲 (VPC) 組成，一個用於動態屬性佈建系統 (DAPS) 身分識別服務，另一個用於 Amazon EKS。

DAP 架構

下圖顯示在由 Auto Scaling 群組控制的 EC2 執行個體上執行的 DAPS。應用程式負載平衡器和路由資料表會公開 DAPS 伺服器。Amazon Elastic File System (Amazon EFS) 可在 DAPS 執行個體之間同步資料。

Amazon EKS 架構

數據空間被設計為與技術無關的解決方案，並且存在多個實現。此模式使用 Amazon EKS 叢集部署資料空間技術元件。下圖顯示 EKS 叢集的部署。工作者節點安裝在私有子網路中。Kubernetes 網繭可存取適用於 PostgreSQL 執行個體的 Amazon Relational Database Service 服務 (Amazon RDS)，該執行個體也位於私有子網路中。Kubernetes 網繭會在 Amazon S3 中存放共用資料。

工具

AWS 服務

- AWS CloudFormation協助您設定 AWS 資源、快速且一致地佈建資源，以及跨區域的整個生命週期進 AWS 帳戶 行管理。
- Amazon Elastic Compute Cloud (Amazon EC2) 在 AWS 雲端中提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Elastic File System (Amazon EFS) 協助您在 AWS 雲端中建立和設定共用檔案系統。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在上執行 Kubernetes， AWS 而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 EC2 執行個體、容器和 IP 地址。

其他工具

- eksctl 是一個命令列公用程式，用於在 Amazon EKS 上建立和管理 Kubernetes 叢集。
- Git 是一個開源的，分佈式版本控制系統。
- HashiCorp 保管箱提供安全的儲存空間，並可控制憑證和其他敏感資訊的存取
- Helm 是 Kubernetes 的開放原始碼套件管理員，可協助您在 Kubernetes 叢集上安裝和管理應用程式。
- kubectl 是一種命令列介面，可協助您針對 Kubernetes 叢集執行命令。
- 郵遞員是一個 API 平台。

代碼存儲庫

此模式的 Kubernetes 設定 YAML 檔案和 Python 指令碼可在 AWS 模式 edc 儲存庫中取得。 GitHub
該模式還使用 Tractus-X 嵌入式設計中心存儲庫。

最佳實務

Amazon EKS 和參與者基礎設施的隔離

Kubernetes 中的命名空間將在此模式中將公司 X 提供者的基礎結構與公司 Y 消費者的基礎結構分開。如需詳細資訊，請參閱 EKS 最佳做法指南。

在更現實的情況下，每個參與者都會有獨立的 Kubernetes 叢集在自己的範圍內執行。 AWS 帳戶數據空間參與者可以訪問共享基礎設施（DAPS 在這種模式中），同時與參與者的基礎設施完全分開。

史诗

設定環境並佈建 EKS 叢集和 EC2 執行個體

任務	描述	所需技能
複製儲存庫。	若要將儲存庫複製到工作站，請執行下列命令： `git clone https://github.com/Think-iT-Labs/aws-patterns-edc` 工作站必須能夠存取您的 AWS 帳戶.	DevOps 工程師
佈建 Kubernetes 叢集並設定命名空間。	若要在您的帳戶中部署簡化的預設 EKS 叢集，請在複製存放庫的工作站上執行下列`eksctl`命令： `eksctl create cluster` 此命令會建立跨越三個不同可用區域的 VPC 以及私有和公用子網路。建立網路層之後，命	DevOps 工程師

任務	描述	所需技能
	令會在一個 Auto Scaling 群組中建立兩個 `m5.large` EC2 執行個體。 如需詳細資訊和範例輸出，請參閱 eksctl 指南。 佈建私人叢集之後，請執行下列命令，將新的 EKS 叢集新增至您的本機 Kubernetes 組態： `aws eks update-kubeconfig --name <EKS CLUSTER NAME> --region <AWS REGION>` 此模式使用`eu-west-1` AWS 區域 來執行所有命令。但是，您可以在首選中運行相同的命令 AWS 區域。 若要確認您的 EKS 節點正在執行且處於就緒狀態，請執行下列命令： `kubectl get nodes`	

任務	描述	所需技能
設定命名空間。	若要為提供者和取用者建立命名空間，請執行下列命令： `kubectl create ns provider` `kubectl create ns consumer` 在這種模式中，使用`provider`和`consumer`作為命名空間以適應後續步驟中的配置非常重要。	DevOps 工程師

部署身分識別服務

任務	描述	所需技能
使 AWS CloudFormation用部署 DAPS。	為了便於管理 DAPS 操作，DAPS 伺服器安裝在 EC2 執行個體上。 若要安裝 DAPS，請使用AWS CloudFormation 範本。您將需要 [必要條件] 區段中的 ACM 憑證和 DNS 名稱。範本會部署並設定下列項目： • Application Load Balancer • Auto Scaling 群組 • 設定使用者資料的 EC2 執行個體，以安裝所有必要套件 • IAM 角色 • DAP	DevOps 工程師

任務	描述	所需技能
	您可以登入並使用AWS CloudFormation 主控台 AWS Management Console來部署 AWS CloudFormation 範本。您也可以使用如下 AWS CLI 命令來部署範本： `aws cloudformation create-stack --stack-name daps \ --template-body file://aws-patterns-edc/cloudformation.yml --parameters \ ParameterKey=CertificateARN,ParameterValue=<ACM Certificate ARN> \ ParameterKey=DNSName,ParameterValue=<DNS name> \ ParameterKey=InstanceType,ParameterValue=<EC2 instance type> \ ParameterKey=EnvironmentName,ParameterValue=<Environment Name> --capabilities CAPABILITY_NAMED_IAM`  環境名稱是您自己的選擇。我們建議使用有意義的術語`DapsInfrastructure`，例如，因為它會反映在 AWS 資源標籤中。	

任務	描述	所需技能
	對於此模式，t3.small足以執行具有三個 Docker 容器的 DAPS 工作流程。 範本會在私有子網路中部署 EC2 執行個體。這表示執行個體無法透過網際網路透過 SSH (安全殼層) 直接存取。這些執行個體會佈建必要的 IAM 角色和 AWS Systems Manager 代理程式，以透過工作階段管理員存取執行中的執行個體，此功能具有的功能 AWS Systems Manager。 我們建議使用工作階段管理員來存取。或者，您可以佈建防禦主機以允許從網際網路存取 SSH。使用防禦主機方法時，EC2 執行個體可能需要幾分鐘的時間才能開始執行。 成功部署 AWS CloudFormation 範本後，請將 DNS 名稱指向您的 Application Load Balancer DNS 名稱。若要確認，請執行下列命令： `dig <DNS NAME>` 輸出格式應類似以下內容： `; <<>> DiG 9.16.1-Ubuntu <<>> edc-pattern.think-it.io` `;; global options: +cmd` `;; Got answer:`	

任務	描述	所需技能
	;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42344 ;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 65494 ;; QUESTION SECTION: ;edc-pattern.think-it.io. IN A ;; ANSWER SECTION: edc-pattern.think-it.io. 276 IN CNAME daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaws.com. daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaws.com. 36 IN A 52.208.240.129 daps-alb-iap9zmwy3kn8-1328773120.eu-west-1.elb.amazonaws.com. 36 IN A 52.210.155.124	

任務	描述	所需技能
將參與者的連接器註冊到 DAPS 服務。	在為 DAPS 佈建的任何 EC2 執行個體中，註冊參與者： 1. 使用 root 使用者在 EC2 執行個體上執行可用的指令碼： `cd /srv/mvds/omejdn-daps` 2. 註冊提供者： `bash scripts/register_connector.sh <provider_name>` 3. 註冊消費者： `bash scripts/register_connector.sh <consumer_name>` 名稱的選擇不會影響後續步驟。我們建議使用 `provider` and `consumer` 或`companyx`和`companyy`。 註冊命令還將自動配置 DAPS 服務，並使用從創建的證書和密鑰中獲取所需的信息。 當您登入 DAPS 伺服器時，請收集安裝後續步驟所需的資訊： 1. 從`omejdn-daps/config/clients.yml` 獲取	DevOps 工程師

任務	描述	所需技能
	提client id供者和消費者。這些client id值是十六進制數字的長字符串。 2. 從目omejdn-daps/keys 錄中複製、consumer.cert consumer.key provider.cert 、和provider.key 檔案的內容。 我們建議您將文字複製並貼到工作站daps-上以前綴的類似名稱檔案中。 您應該有提供者和消費者的用戶端 ID，而且工作站上的工作目錄中應該有四個檔案： • 來源檔案名稱consumer.cert 會變成工作站檔案名稱daps-consumer.cert 。 • 來源檔案名稱consumer.key 會變成工作站檔案名稱daps-consumer.key 。 • 來源檔案名稱provider.cert 會變成工作站檔案名稱daps-provider.cert 。 • 來源檔案名稱provider.key 會變成工作站檔	

任務	描述	所需技能
	案名稱daps-provider.key 。	

部署參與者的連接器

任務	描述	所需技能
複製 Tractus-X 嵌入式設計中心儲存庫並使用 0.4.1 版本。	Tractus-X 嵌入式設計中心連接器的組建需要 PostgreSQL (資產資料庫) 和文件 HashiCorp 庫 (密碼管理) 服務才能部署並可供使用。 有許多不同版本的 Tractus-X EDC 頭盔圖表。此病毒碼會指定 0.4.1 版，因為它使用 DAPS 伺服器。 最新版本使用受管理的身分識別錢包 (MIW) 搭配分散式實作的身分識別服務。 在您建立兩個 Kubernetes 命名空間的工作站上，複製 tractusx-edc 儲存庫，然後簽出分支。`release/0.4.1` `git clone https://github.com/eclipse-tractusx/tractusx-edc` `cd tractusx-edc` `git checkout release/0.4.1`	DevOps 工程師

任務	描述	所需技能
設定 Tractus-X 嵌入式設計中心頭盔圖表。	修改 Tractus-X Helm 圖表範本組態，使兩個連接器能夠一起互動。 若要這麼做，您可以將命名空間新增至服務的 DNS 名稱，以便叢集中的其他服務可以解析該名稱空間。應對`charts/tractusx-connector/templates/_helpers.tpl` 檔案進行這些修改。此病毒碼提供此檔案的最終修改版本供您使用。複製它並將其放在文件的`daps`部分中`charts/tractusx-connector/templates/_helpers.tpl` 。 確保在`charts/tractusx-connector/Chart.yaml` 以下位置註釋所有 DAPS 依賴關係： `dependencies:` `  # IDS Dynamic Attribute Provisioning Service (IAM)` `  #  - name: daps` `  # version: 0.0.1` `  # repository: "file://./subcharts/omejdn"` `  # alias: daps` `  # condition: install.daps`	DevOps 工程師

任務	描述	所需技能
設定連接器以在 Amazon RDS 上使用 PostgreSQL。	(選用) 此模式不需要 Amazon 關聯式資料庫服務 (Amazon RDS) 執行個體。不過，我們強烈建議您使用 Amazon RDS 或 Amazon Aurora，因為它們提供高可用性以及備份和復原等功能。 若要以 Amazon RDS 取代庫伯尼特斯上的 PostgreSQL，請執行下列動作： 1. 佈建適用於 PostgreSQL 執行個體的亞馬遜 RDS。 2. 在中`Chart.yaml` ，為`PostgreSQL` 區段加上註解。 3. 在`provider_values.yml` 和中`consumer_values.yml` ，如下所示設定`postgresql` 區段： `postgresql:` `  auth:` `    database: edc` `    password: <RDS PASSWORD>` `    username: <RDS Username>` `  jdbcUrl: jdbc:postgresql://<RDS DNS NAME>:5432/edc` `  username: <RDS Username>` `  password: <RDS PASSWORD>`	DevOps 工程師

任務	描述	所需技能
	<pre>primary: persistence: enabled: false readReplicas: persistence: enabled: false</pre>	

任務	描述	所需技能
設定和部署提供者連接器及其服務。	若要設定提供者連接器及其服務，請執行下列動作： 1. 若要將provider_edc.yaml 檔案從edc_helm_configs 目錄下載到目前的 Helm 圖表資料夾，請執行下列命令： `wget -q https://raw.githubusercontent.com/Think-iT-Labs/aws-patterns-edc/main/edc_helm_configs/provider_edc.yaml> -P charts/tractusx-connector/` 2. 將下列變數 (也標記在檔案中) 取代為其值： • CLIENT_ID – DAPS 所產生的識別碼。CLIENT_ID 應該位/srv/mvds/omejdn-daps/config/clients.yml/config/clients.yml 於 DAPS 伺服器上。它應該由十六進制字符組成的字符串。 • DAPS_URL– DAPS 伺服器的網址 它應該https://{DNS name}使用您在運行	DevOps 工程師

任務	描述	所需技能
	AWS CloudFormation 模板時設置的 DNS 名稱。 • VAULT_TOKEN – 用於保管庫授權的令牌。選擇任何值。 • vault.fullnameOverride – vault-provider 。 • vault.hashicorp.url – http://vault-provider:8200/ 。 先前的值假設部署名稱和命名空間名稱是 Provider。 3. 若要從工作站執行 Helm 圖表，請使用下列指令： `cd charts/tractusx-connector` `helm dependency build` `helm upgrade --install provider ./ -f provider_edc.yaml -n provider`	

任務	描述	所需技能
將憑證和金鑰新增至提供者儲存庫。	為避免混淆，請在`tractusx-edc/charts` 目錄之外產生下列憑證。 例如，執行下列命令以變更為您的主目錄： `cd ~` 您現在需要將提供者所需的密碼新增至 Vault。 Vault 中密碼的名稱是`provider_edc.yml` 檔案`secretNames:` 區段中金鑰的值。依預設，它們的設定方式如下： `secretNames:` `transferProxyTokenSignerPrivateKey: transfer-proxy-token-signer-private-key` `transferProxyTokenSignerPublicKey: transfer-proxy-token-signer-public-key` `transferProxyTokenEncryptionAesKey: transfer-proxy-token-encryption-aes-key` `dapsPrivateKey: daps-private-key`	DevOps 工程師

任務	描述	所需技能
	<pre>dapsPubli cKey: daps-public-key</pre> 一開始會產生進階加密標準 (AES) 金鑰、私密金鑰、公開金鑰和自我簽署憑證。這些隨後會作為密碼新增至 Vault。 此外，此目錄應包含您從 DAPS 伺服器複製的`daps-provider.cert` 和`daps-provider.key` 檔案。 1. 執行下列命令： <pre># generate a private key openssl ecparam -name prime256v1 -genkey -noout -out provider- private-key.pem # generate correspon ding public key openssl ec -in provider-private-k ey.pem -pubout -out provider-public-ke y.pem # create a self-sign ed certificate openssl req -new - x509 -key provider- private-key.pem -out provider-cert.pem - days 360 # generate aes key openssl rand -base64 32 > provider- aes.key</pre>	

任務	描述	所需技能
	2. 在將密碼加入 Vault 之前，請透過以下方式取代換行符號，將密碼從多行轉換為單行\n： `cat provider-private-key.pem | sed 's/$/\\\n/'\|tr -d '\\n' > provider-private-key.pem.line` `cat provider-public-key.pem \| sed 's/$/\\\n/'|tr -d '\\n' > provider-public-key.pem.line` `cat provider-cert.pem | sed 's/$/\\\\n/'\|tr -d '\\n' > provider-cert.pem.line` `cat provider-aes.key \| sed 's/$/\\\\n/'|tr -d '\\n' > provider-aes.key.line` `## The following block is for daps certificate and key` `openssl x509 -in daps-provider.cert -outform PEM | sed 's/$/\\\\n/'\|tr -d '\\n' > daps-provider.cert.line` `cat daps-provider.key \| sed 's/$/\\\\n/'| tr -d '\\n' > daps-provider.key.line`	

任務	描述	所需技能
	3. 若要格式化將加入至 Vault 的密碼，請執行下列命令： `JSONFORMAT='{"content": "%s"}'` `#create a single line in JSON format` `printf "${JSONFORMAT}\\n" "`cat provider-private-key.pem.line`" > provider-private-key.json` `printf "${JSONFORMAT}\\n" "`cat provider-public-key.pem.line`" > provider-public-key.json` `printf "${JSONFORMAT}\\n" "`cat provider-cert.pem.line`" > provider-cert.json` `printf "${JSONFORMAT}\\n" "`cat provider-aes.key.line`" > provider-aes.json` `printf "${JSONFORMAT}\\n" "`cat daps-provider.key.line`" > daps-provider.key.json` `printf "${JSONFORMAT}\\n" "`cat daps-provider.cert.line`" > daps-provider.cert.json`	

任務	描述	所需技能
	密碼現在為 JSON 格式，可以新增至保存庫。 4. 若要取得儲存庫的網繭名稱，請執行下列命令： `kubectl get pods -n provider|egrep "vault|NAME"` 網繭名稱將類似於"vault-provider-0" 。建立轉寄至資料保險箱的連接埠時會使用此名稱。端口轉發使您可以訪問文件庫以添加密碼。您應該從已設定 AWS 登入資料的工作站執行此指令。 5. 若要存取資料保險箱，請使用kubectl來規劃連接埠轉發： `kubectl port-forward <VAULT_POD_NAME> 8200:8200 -n provider` 您現在應該可以透過瀏覽器或 CLI 存取保管庫。 瀏覽器 1. 使用瀏覽器瀏覽至 http://127.0.0.1:8200，這將使用您設定的連接埠轉送。	

任務	描述	所需技能
	2. 使用您先前設定的權杖登入provider_edc.yml 。在秘密引擎中，建立三個密碼。每個密碼都會有一個Path for this secret值，這是下列清單中顯示的秘密名稱。在該secret data部分中，密鑰的名稱將是content，該值將是命名的相應文件中的單行文本.line。 3. 密碼名稱來自provider_edc.yml 檔案中的secretNames 區段。 4. 建立下列密碼： • transfer-proxy-token-signer-private-key 使用檔案名稱的密碼 provider-private-key.pem.line • transfer-proxy-token-signer-public-key 使用檔案名稱的密碼 provider-cert.pem.line • transfer-proxy-token-encryption-aes-key 使用檔案名稱的密碼 provider-aes.key.line	

任務	描述	所需技能
	• daps-private-key 使用檔案名稱的密碼 daps-provider.key.line • daps-public-key 使用檔案名稱的密碼 daps-provider.cert .line 文件庫 CLI CLI 也會使用您設定的連接埠轉送。 1. 在您的工作站上，依照 Vault 文件中的指示安裝 HashiCorp Vault CLI。 2. 若要使用您設定的權杖登入 Vaultprovider_edc.yml ，請執行下列命令： `vault login -address= http://127.0.0.1:8 200` 使用正確的令牌，您應該會看到該消息 "Success! You are now authenticated." 3. 若要使用先前建立的 JSON 格式檔案來建立密碼，請執行下列程式碼： `vault kv put -address= http://127.0.0.1:8`	

任務	描述	所需技能
	200 secret/transfer-proxy-token-signer-private-key @provider-private-key.json vault kv put -address=http://127.0.0.1:8200 secret/transfer-proxy-token-signer-public-key @provider-cert.json vault kv put -address=http://127.0.0.1:8200 secret/transfer-proxy-token-encryption-aes-key @provider-aes.json vault kv put -address=http://127.0.0.1:8200 secret/daps-private-key @daps-provider.key.json vault kv put -address=http://127.0.0.1:8200 secret/daps-public-key @daps-provider.cert.json	

任務	描述	所需技能
設定和部署用戶連接器及其服務。	設定和部署用戶的步驟與您為提供者完成的步驟類似： 1. 要將 aws-模式-edc 存儲庫複製到 `tractusx-edc/charts/tractusx-connector` 文件夾`consumer_edc.yaml` 中，請運行以下命令： `cd tractusx-edc` `wget -q https://raw.githubusercontent.com/Think-iT-Labs/aws-patterns-edc/main/edc_helm_configs/consumer_edc.yaml -P charts/tractusx-connector/` 2. 使用其實際值更新下列變數： • `CONSUMER_CLIENT_ID` – DAPS 所產生的識別碼。`CONSUMER_CLIENT_ID` 應該位`config/clients.yml` 於 DAPS 伺服器上。 • `DAPS_URL`– 與您用於提供者的 DAPS 網址相同。 • `VAULT_TOKEN` – 用於保管庫授權的令牌。選擇任何值。	

任務	描述	所需技能
	• vault.fullnameOverride – vault-consumer • vault.hashicorp.url – http://vault-provider:8200/ 先前的值假設部署名稱和命名空間名稱為consumer。 3. 要運行頭盔圖，請使用以下命令： `cd charts/tractusx-connector` `helm upgrade --install consumer ./ -f consumer_edc.yaml -n consumer`	

任務	描述	所需技能
將憑證和金鑰新增至取用者保存庫。	從安全的角度來看，我們建議為每個資料空間參與者重新產生憑證和金鑰。此病毒碼會重新產生取用者的憑證和金鑰。 這些步驟與提供者的步驟非常相似。您可以驗證consumer_edc.yml 檔案中的密碼名稱。 儲存庫內的密碼名稱是的一secretNames: 節中的金鑰值consumer_edc.yml file。依預設，它們的設定方式如下： `secretNames:` `transferProxyTokenSignerPrivateKey: transfer-proxy-token-signer-private-key` `transferProxyTokenSignerPublicKey: transfer-proxy-token-signer-public-key` `transferProxyTokenEncryptionAesKey: transfer-proxy-token-encryption-aes-key` `dapsPrivateKey: daps-private-key` `dapsPublicKey: daps-public-key`	DevOps 工程師

任務	描述	所需技能
	您從 DAPS 伺服器複製的daps-consumer.cert 和daps-consumer.key 檔案應該已存在於此目錄中。 1. 執行下列命令： ``` # generate a private key openssl ecparam -name prime256v1 -genkey -noout -out consumer-private-key.pem # generate corresponding public key openssl ec -in consumer-private-key.pem -pubout -out consumer-public-key.pem # create a self-signed certificate openssl req -new -x509 -key consumer-private-key.pem -out consumer-cert.pem -days 360 # generate aes key openssl rand -base64 32 > consumer-aes.key ``` 2. 手動編輯要取代換行符號的檔案\n，或使用類似下列的三個指令： ``` cat consumer-private-key.pem | sed 's/$/\\\n/'|tr -d '\\n' >	

任務	描述	所需技能
	consumer-private-key.pem.line cat consumer-public-key.pem | sed 's/$/\\\n/'\|tr -d '\\n' > consumer-public-key.pem.line cat consumer-cert.pem \| sed 's/$/\\\\n/'|tr -d '\\n' > consumer-cert.pem.line cat consumer-aes.key | sed 's/$/\\\\n/'\|tr -d '\\n' > consumer-aes.key.line cat daps-consumer.cert \| sed 's/$/\\\\n/'|tr -d '\\n' > daps-consumer.cert.line cat daps-consumer.key | sed 's/$/\\\\n/'\|tr -d '\\n' > daps-consumer.key.line 3. 若要格式化將加入至 Vault 的密碼，請執行下列命令： JSONFORMAT='{"content": "%s"}' #create a single line in JSON format printf "${JSONFORMAT}\\n" "`cat consumer-private-key.pem.line`" >	

任務	描述	所需技能
	consumer-private-key.json printf "${JSONFORMAT}\\n" "`cat consumer-public-key.pem.line`" > consumer-public-key.json printf "${JSONFORMAT}\\n" "`cat consumer-cert.pem.line`" > consumer-cert.json printf "${JSONFORMAT}\\n" "`cat consumer-aes.key.line`" > consumer-aes.json printf "${JSONFORMAT}\\n" "`cat daps-consumer.key.line`" > daps-consumer.key.json printf "${JSONFORMAT}\\n" "`cat daps-consumer.cert.line`" > daps-consumer.cert.json 密碼現在為 JSON 格式，可以新增至保存庫。 4. 若要取得用戶儲存庫的網繭名稱，請執行下列命令： kubectl get pods -n consumer | egrep "vault|NAME"	

任務	描述	所需技能
	網繭名稱將類似於"vault-consumer-0" 。建立轉寄至資料保險箱的連接埠時會使用此名稱。端口轉發使您可以訪問文件庫以添加密碼。您應該從已設定 AWS 認證的工作站執行此指令。 5. 若要存取資料保險箱，請使用kubectl來規劃連接埠轉發： `kubectl port-forward <VAULT_POD_NAME> 8201:8200 -n consumer` 這次本地端口是 8201，因此您可以為生產者和消費者提供端口轉發。 瀏覽器 您可以使用瀏覽器連接到 http://localhost:8201/ 訪問消費者保管庫，並按照概述使用名稱和內容創建密碼。 包含內容的密碼和檔案如下： • `transfer-proxy-token-signer-private-key` 使用檔案名稱的密碼 `consumer-private-key.pem.line` • `transfer-proxy-token-signer-public-`	

任務	描述	所需技能
	key 使用檔案名稱的密碼 consumer-cert.pem.line • transfer-proxy-token-encryption-aes-key 使用檔案名稱的密碼 consumer-aes.key.line 文件庫 CLI 使用 Vault CLI，您可以執行下列命令來登入儲存庫並建立密碼： 1. 使用您在以下內容中設定的權杖登入 Vaultconsumer_edc.yml ： `vault login -address=http://127.0.0.1:8201` 使用正確的令牌，您應該會看到該消息 "Success! You are now authenticated." 2. 若要使用先前建立的 JSON 格式檔案建立密碼，請執行下列程式碼： `vault kv put -address=http://127.0.0.1:8201 secret/transfer-proxy-token-signer-p`	

任務	描述	所需技能
	rivate-key @consumer -private-key.json vault kv put - address=http://12 7.0.0.1:8201 secret/ transfer-proxy-token -signer-public-key @consumer-cert.json vault kv put -address= http://127.0.0.1:8 201 secret/transfer- proxy-token-encrypti on-aes-key @consumer -aes.json vault kv put -address= http://127.0.0.1:8 201 secret/daps- private-key @daps-con sumer.key.json vault kv put - address=http://12 7.0.0.1:8201 secret/ daps-public-key @daps-consumer.cer t.json	

設定 HTTP 用戶端以與連接器的管理 API 互動

任務	描述	所需技能
設定連接埠轉送。	1. 若要檢查網繭的狀態，請執行下列命令： kubectl get pods -n provider kubectl get pods -n consumer	DevOps 工程師

任務	描述	所需技能
	2. 若要確定 Kubernetes 部署成功，請執行下列命令來查看提供者和取用者 Kubernetes 網繭的記錄檔： `kubectl logs -n provider <producer control plane pod name>` `kubectl logs -n consumer <consumer control plane pod name>` 叢集是私有的，不可公開存取。若要與連接器互動，請使用 Kubernetes 連接埠轉送功能將機器產生的流量轉送至連接器控制平面。 1. 在第一個終端機上，透過連接埠 8300 將消費者的要求轉寄至管理 API： `kubectl port-forward deployment/consumer-tractusx-connector-controlplane 8300:8081 -n consumer` 2. 在第二個終端機上，透過連接埠 8400 將提供者的要求轉寄至管理 API： `kubectl port-forward deployment/provider-tractusx-connect`	

任務	描述	所需技能
	`or-controlplane 8400:8081 -n provider`	

任務	描述	所需技能
為提供者和消費者建立 S3 儲存貯體。	EDC 連接器目前不使用臨時 AWS 登入資料，例如假定角色所提供的登入資料。嵌入式設計中心僅支援使用 IAM 存取金鑰 ID 和秘密存取金鑰組合。 稍後的步驟需要兩個 S3 儲存貯體。一個 S3 儲存貯體用於存放供應商提供的資料。另一個 S3 儲存貯體用於取用者收到的資料。 IAM 使用者應該只有在兩個具名值區中讀取和寫入物件的權限。 需要創建訪問密鑰 ID 和秘密訪問密鑰對並保持安全。解除委任此 MVDS 之後，應該刪除 IAM 使用者。 下列程式碼是使用者的 IAM 政策範例： `{ "Version": "2012-10-17", "Statement": [ { "Sid": "Stmt1708699805237", "Action": [ "s3:GetObject", "s3:GetObjectVersion", "s3:ListAllMyBuckets",`	DevOps 工程師

任務	描述	所需技能
	"s3:ListB ucket", "s3:ListB ucketMultipartUplo ads", "s3:ListB ucketVersions", "s3:PutObject"], "Effect": "Allow", "Resource": ["arn:aws: s3:::<S3 Provider Bucket>", "arn:aws: s3:::<S3 Consumer Bucket>", "arn:aws: s3:::<S3 Provider Bucket>/*", "arn:aws: s3:::<S3 Consumer Bucket>/*"] }] }	
設定郵差以與連接器互動。	您現在可以透過 EC2 執行個體與連接器互動。使用 Postman 做為 HTTP 用戶端，並為提供者和取用者連接器提供郵遞員集合。 將集合從aws-pattern-edc儲存庫匯入 Postman 執行個體。 此模式使用 Postman 集合變量為您的請求提供輸入。	應用程式開發人員、資料

通過連接器提供公司 X 碳排放足跡數據

任務	描述	所需技能
準備要共享的碳排放強度數據。	首先，您需要決定要共用的資料資產。X 公司的數據表示其車隊的碳排放足跡。重量是以公噸為單位的車輛總重（GVW），根據輪到井（WTW）測量結果，排放量單位為每噸公里二氧化碳的克數（克 CO2 e/t-km）： • 車輛類型:廂型車; 重量:< 3.5; 排放:800 • 車輛類型:城市卡車; 重量:3.57.5; 排放:315 • 車輛類型:中型貨車; 重量:7.520; 廢氣排放:195 • 車輛類型:重型貨車; 重量:> 20; 廢氣排放:115 範例資料位於`aws-patterns-edc` 儲存庫中的`carbon_emissions_data.json` 檔案中。 X 公司使用 Amazon S3 存放對象。 建立 S3 儲存貯體並將範例資料物件存放在該處。下列命令會建立具有預設安全性設定的 S3 儲存貯體。我們強烈建議您諮詢 Amazon S3 的安全性最佳實務。	資料工程師、App 開發人員

任務	描述	所需技能
	`aws s3api create-bucket <BUCKET_NAME> --region <AWS_REGION>` `# You need to add '--create-bucket-configuration` `# LocationConstraint=<AWS_REGION>' if you want to create # the bucket outside of us-east-1 region` `aws s3api put-object --bucket <BUCKET_NAME> \` `--key <S3 OBJECT NAME> \` `--body <PATH OF THE FILE TO UPLOAD>` S3 儲存貯體名稱應該是全域唯一的。如需命名規則的詳細資訊，請參閱 AWS 文件。	

任務	描述	所需技能
使用 Postman 將資料資產註冊到提供者的連接器。	EDC 連接器資料資產會保留資料的名稱及其位置。在此情況下，EDC 連接器資料資產將指向 S3 儲存貯體中建立的物件： • 連接器：供應商 • 請求：建立資產 • 集合變數：更新ASSET_NAME 。選擇代表資產的有意義名稱。 • 要求主體：使用您為提供者建立的 S3 儲存貯體更新要求主體。 `"dataAddress": { "edc:type": "AmazonS3", "name": "Vehicle Carbon Footprint", "bucketName": "<REPLACE WITH THE SOURCE BUCKET NAME>", "keyName": "<REPLACE WITH YOUR OBJECT NAME>", "region": "<REPLACE WITH THE BUCKET REGION>", "accessKeyId": "<REPLACE WITH YOUR ACCESS KEY ID>", "secretAccessKey": "<REPLACE WITH SECRET ACCESS KEY>" }`	應用程式開發人員、資料

任務	描述	所需技能
	• 回應:成功的要求會傳回建立的時間和新建立資產的資產 ID。 `{ "@id": "c89aa31c-ec4c-44ed-9e8c-1647f19d7583" }` • 收集變數 ASSET_ID：ASSET_ID使用嵌入式設計中心連接器在建立後自動產生的識別碼來更新郵遞員集合變數。	

任務	描述	所需技能
定義資產的使用政策。	嵌入式設計中心資產必須與明確的使用政策相關聯。首先，在提供者連接器中建立原則定義。 X 公司的政策是允許數據空間的參與者使用碳排放足跡數據。 • 請求主體： • 連接器：供應商 • 要求：建立原則 • 收集**Policy Name**變數：以原則名稱更新變數。 • 回應：成功的要求會傳回建立的時間和新建立之原則的原則識別碼。在建立之後，POLICY_ID 使用 EDC 連接器所產生的原則識別碼來更新收集變數。	應用程式開發人員、資料
定義資產的嵌入式設計中心合約方案及其使用政策。	若要允許其他參與者要求存取您的資料，請在指定使用條件和權限的合約中提供資料： • 連接器：供應商 • 請求：建立合約定義 • 收集**Contract Name**變數：使用合約提案或定義的名稱更新變數。	應用程式開發人員、資料

發現資產並在定義的合同達成協議

任務	描述	所需技能
索取 X 公司共用的資料目錄。	身為資料空間中的資料消費者，Y 公司首先需要探索其他參與者共用的資料。 在此基本設定中，您可以要求取用者連接器直接向提供者連接器要求可用資產目錄來執行此操作。 • 連接器：消費者 • 索取：索取目錄 • 回應：來自提供者的所有可用資料資產及其附加的使用政策。身為資料使用者，請尋找您感興趣的合約，並相應地更新下列收集變數。 • CONTRACT_OFFER_ID – 消費者希望談判的合同提供的 ID • ASSET_ID– 消費者想要談判的資產 ID • PROVIDER_CLIENT_ID – 與之交涉的供應商連接器 ID	應用程式開發人員、資料
就 X 公司的碳排放強度數據展開合同談判。	現在您已經識別出要使用的資產，請啟動取用者與提供者連接器之間的合約協商程序。 • 連接器：消費者 • 請求：合同談判	應用程式開發人員、資料

任務	描述	所需技能
	• 集合**CONSUMER_CLIENT_ID** 變數：使用要交涉之取用者連接器的 ID 更新變數。 該過程可能需要一些時間才能到達「已驗證」狀態。 您可以使用`Get Negotiation` 請求來檢查「合約協議」的狀態和對應的「合約 ID」。	

使用合同協議使用數據

任務	描述	所需技能
使用來自 HTTP 端點的資料。	（選項 1）要使用 HTTP 數據平面消耗數據空間中的數據，可以使用 webhook.site 模擬 HTTP 服務器，並在消費者連接器中啟動傳輸過程： • 連接器：消費者 • 請求：合同談判 • 收集**Contract Agreement ID**變數：使用 EDC 連接器所產生之合約的 ID 來更新變數。 • 請求主體：更新請求主體以指定HTTP為`dataDestination` 與 webhook URL 旁邊的一個： `{`	應用程式開發人員、資料

任務	描述	所需技能
	<pre>"dataDestination": { "type": "HttpProxy" }, "privateProperties": { "receiverHttpEndpoint": "<WEBHOOK URL>" } }</pre> 連接器會將直接下載檔案所需的資訊傳送至 Webhook URL。 接收到的裝載類似於以下內容： <pre>{ "id": "dcc90391-3819-4b54-b401-1a005a029b78", "endpoint": "http://consumer-tractusx-connector-dataplane.consumer:8081/api/public", "authKey": "Authorization", "authCode": "<AUTH CODE YOU RECEIVE IN THE ENDPOINT>", "properties": { "https://w3id.org/edc/v0.0.1/ns/cid": "vehicle-carbon-footprint-contract:4563abf7-5</pre>	

任務	描述	所需技能
	dc7-4c28-bc3d-97f4 5e32edac:b073669b- db20-4c83-82df-46b 583c4c062" } } 使用接收到的登入資料取得供應商共用的 S3 資產。 在最後一個步驟中，您必須將請求發送到消費者數據平面（正確轉發端口），如有效負載（endpoint）中所述。	

任務	描述	所需技能
直接使用 S3 儲存貯體中的資料。	(選項 2) 使用 Amazon S3 與 EDC 連接器整合，並直接指向消費者基礎設施中的 S3 儲存貯體做為目的地： • 要求主體：更新要求主體，將 S3 儲存貯體指定為資料目標。 這應該是您先前為存放取用者接收到的資料而建立的 S3 儲存貯體。 `{ "dataDestination": { "type": "AmazonS3", "bucketName": "{{ REPLACE WITH THE DESTINATION BUCKET NAME }}", "keyName": "{{ REPLACE WITH YOUR OBJECT NAME }}", "region": "{{ REPLACE WITH THE BUCKET REGION }}", "accessKeyId": "{{ REPLACE WITH YOUR ACCESS KEY ID }}", "secretAccessKey": "{{ REPLACE WITH SECRET ACCESS KEY }}" } } }`	資料工程師、App 開發人員

故障診斷

問題	解決方案
連接器可能會引發有關憑證 PEM 格式的問題。	通過添加將\n每個文件的內容連接到單行。

相關資源

- DSSC
- 為永續性使用案例建置資料空間 (Thin k-IT 提供的 AWS Prescriptive Guidance 策略)
- 適用於資料空間的 AWS
- 拖拉卡斯-X 文檔
- DAP
- 透過資料空間和 AWS 啟用資料共用 (部落格文章)

其他資訊

資料空間規格

參加者

參與者	公司描述	公司焦點
公司 X	經營整個歐洲和南美洲的車隊運輸各種貨物。	旨在做出數據驅動的決策，以減少碳排放強度。
Y 公司名稱	環境監管機構	執行環境法規和政策，以監察和減輕企業和行業的環境影響，包括碳排放強度。

商業案例

X 公司使用數據空間技術與合規審核員 Y 公司共享碳足跡數據，以評估和解決 X 公司物流運營對環境的影響。

数据空间权威

數據空間權威是管理數據空間的組織的聯盟。在此模式中，X 公司和 Y 公司都會組成治理主體，並代表聯合資料空間授權單位。

資料空間元件

元件	選擇的實作	其他資訊
數據集交換協議	資料空間通訊協定 0.8 版	• JSON-LD • 資料目錄詞彙 (DCAT)
資料空間連接器	拖曳-X 嵌入式設計中心連接器 0.4.1 版	• 嵌入式設計中心
資料交換政策	預設使用原則	• 开放数字权利语言

資料空間服務

服務	實施	其他資訊
身份服務	動態屬性佈建系統 (DAPS)	「動態屬性佈建系統 (DAPS) 旨在確定組織和連接器的某些屬性。因此，只要他們信任 DAPS 斷言，第三方就不需要信任後者。」 — DAP 為了專注於連接器的邏輯，資料空間會使用 Docker 撰寫在 Amazon EC2 機器上部署。
探索服務	蓋亞-X 聯邦產品目錄	「聯合目錄構成了 Gaia-X 自我描述的索引存儲庫，以便發現和選擇提供者及其服務產品。自我描述是參與者以屬性和索賠形式提供的有關自己和他們的服務的信息。」 — 蓋亞-X 生態系統啟動器

要交換的數據

資料資產	Description	Format (格式)
碳排放數據	指定地區（歐洲和南美洲）中不同車輛類型的強度值來自整個車隊	JSON 檔案

資料模型

```
{
  "region": "string",
  "vehicles": [
// Each vehicle type has its Gross Vehicle Weight (GVW) category and its emission
 intensity in grams of CO2 per Tonne-Kilometer (g CO2 e/t-km) according to the "Well-
to-Wheel" (WTW) measurement.
    {
      "type": "string",
      "gross_vehicle_weight": "string",
      "emission_intensity": {
        "CO2": "number",
        "unit": "string"
      }
    }
  ]
}
```

氣管-X EDC 連接器

如需每個 Tractus-X 嵌入式設計中心參數的說明文件，請參閱原始值檔案。

下表列出所有服務，以及其對應的暴露連接埠和端點，以供參考。

服務名稱	連接埠和路徑
控制平台	• 管理：-港口：8081 路徑：/management • 控制-港口：8083 路徑：/control

	• 通訊協定連接埠：8084 路徑：/api/v1/dsp • 指標-港口：9090 路徑：/metrics • 可觀察性-港口：8085 路徑：/observability
資料平面	預設-連接埠：8080 路徑：/api 公共-港口：8081 路徑：/api/dataplane/control 代理伺服器-連接埠:8186 路徑:/proxy 度量-港口：9090 路徑：/metrics 可觀測性-港口：8085 路徑：/observability
保存庫	連接埠：
PostgreSQL	連接埠：

使用 AWS Secrets Manager 管理員

可以使用 Secrets Manager 而不是 HashiCorp 保管庫作為秘密管理器。若要這麼做，您必須使用或建置 AWS Secrets Manager EDC 延伸模組。

您將負責創建和維護自己的圖像，因為 Tractus-X 不為 Secrets Manager 提供支持。

為此，您需要通過引入 AWS Secrets Manager EDC 擴展來修改控制平面和連接器數據平面的構建 Gradle 文件（請參閱此 Maven 工件的示例），然後構建，維護和引用 Docker 映像。

如需有關重構 Tractus-X 連接器泊塢視窗影像的詳細資訊，請參閱重構 Tractus-X 嵌入式設計中心頭盔圖表。

為了簡單起見，我們避免以此樣式重建連接器影像並使用 HashiCorp Vault。

使用標量 Python UDF 為 Amazon Redshift 查詢結果設置特定語言排序

由伊桑·斯塔克（AWS）創建

環境：生產	技術：分析	AWS 服務：Amazon Redshift

Summary

此模式提供使用純量 Python UDF (使用者定義函數) 為 Amazon Redshift 查詢結果設定不區分大小寫的語言排序的步驟和範例程式碼。有必要使用標量 Python UDF，因為 Amazon Redshift 會根據二進制 UTF-8 排序返回結果，並且不支持特定於語言的排序。一個 Python 的 UDF 是基於一個 Python 2.7 程序，並在數據倉庫中運行非 SQL 處理代碼。您可以在單一查詢中使用 SQL 陳述式來執行 Python UDF 程式碼。如需詳細資訊，請參閱 AWS 大數據部落格中的 Python UDF 簡介。

此模式中的樣本數據基於土耳其語字母，用於演示目的。此模式中的純量 Python UDF 是為了使 Amazon Redshift 的預設查詢結果符合土耳其語中字元的語言排序而建置。如需詳細資訊，請參閱此模式的其他資訊一節中的土耳其文語言範例。您可以在此模式中修改其他語言的標量 Python UDF。

先決條件和限制

先決條件

- 具有資料庫、結構描述和表格的 Amazon Redshift 叢集
- 具有創建表和創建功能許可的 Amazon Redshift 用戶
- Python 2.7 或更高版本

限制

此模式中的查詢使用的語言排序不區分大小寫。

架構

技術堆疊

- Amazon Redshift
- Python

工具

AWS 服務

- Amazon Redshift 是 AWS 雲端中的受管 PB 級資料倉儲服務。Amazon Redshift 與您的資料湖整合，可讓您使用資料為您的企業和客戶取得新的見解。

其他工具

- Python (UDF) 使用者定義函式是您可以使用 Python 撰寫，然後在 SQL 陳述式中呼叫的函式。

史诗

開發程式碼，以語言順序排序查詢結果

任務	描述	所需技能
為範例資料建立資料表。	若要在 Amazon Redshift 中建立資料表並將範例資料插入資料表，請使用下列 SQL 陳述式： `CREATE TABLE my_table (first_name varchar(30));` `INSERT INTO my_table (first_name) VALUES ('ali'), ('Ali'), ('ırmak'), ('IRMAK'), ('irem'), ('İREM'), ('oğuz'), ('OĞUZ'), ('ömer'), ('ÖMER'), ('sedat'),`	數據工程師

任務	描述	所需技能
	`` ('SEDAT'), ('şule'), ``` 注意：樣本數據中的名字包括土耳其語字母中的特殊字符。如需此範例中土耳其文語言考量的詳細資訊，請參閱此模式的其他資訊一節中的土耳其文語言範例。	

任務	描述	所需技能
檢查樣本數據的默認排序。	若要在 Amazon Redshift 中查看範例資料的預設排序方式，請執行下列查詢： `SELECT first_name FROM my_table ORDER BY first_name;` 查詢會傳回您先前建立的資料表中的名字清單： `first_name` `---------------` `Ali` `IRMAK` `OĞUZ` `SEDAT` `ali` `irem` `oğuz` `sedat` `ÖMER` `ömer` `İREM` `ırmak` `ŞULE` `şule` 查詢結果的順序不正確，因為預設的二進位 UTF-8 排序不適合土耳其文特殊字元的語言順序。	數據工程師

任務	描述	所需技能
創建一個標量 Python UDF。	要創建一個標量 Python UDF，使用下面的 SQL 代碼： `CREATE OR REPLACE FUNCTION collate_sort (value varchar) RETURNS varchar IMMUTABLE AS $$ def sort_str(val): import string dictionary = { 'I': 'ı', 'ı': 'h~', 'İ': 'i', 'Ş': 's~', 'ş': 's~', 'Ğ': 'g~', 'ğ': 'g~', 'Ü': 'u~', 'ü': 'u~', 'Ö': 'o~', 'ö': 'o~', 'Ç': 'c~', 'ç': 'c~' } for key, value in dictionary.items(): val = val.replace(key, value) return val.lower()`	數據工程師

任務	描述	所需技能
	<pre> return sort_str(value) $$ LANGUAGE plpythonu;</pre>	
查詢範例資料。	若要使用 Python UDF 查詢範例資料，請執行下列 SQL 查詢： <pre>SELECT first_name FROM my_table ORDER BY collate_order(firs t_name);</pre> 查詢現在會以土耳其語言順序傳回範例資料： <pre>first_name --------------- ali Ali ırmak IRMAK irem İREM oğuz OĞUZ ömer Ömer sedat SEDAT şule ŞULE</pre>	數據工程師

相關資源

- 訂單 BY 條款（Amazon Redshift 文檔）

- 創建一個標量 Python UDF（Amazon Redshift 文檔）

其他資訊

土耳其語示例

Amazon Redshift 會根據二進位 UTF-8 排序順序傳回查詢結果，而非語言特定的排序順序。這表示如果您查詢包含土耳其文字元的 Amazon Redshift 資料表，則查詢結果不會根據土耳其語的語言順序排序。土耳其語包含六個不出現在拉丁字母中的特殊字符（ç，ı，，ö，ş 和 ü）。這些特殊字元會根據二進位 UTF-8 順序放置在排序結果集的結尾，如下表所示。

二進制 UTF-8 排序	土耳其語語言排序
a	a
b	b
c	c
d	ç (*)
e	d
f	e
g	f
h	g
i	(*)
j	h
k	英 (*)
l	i
m	j
n	k

o	l
p	m
r	n
s	o
t	ö (*)
u	p
v	r
y	s
z	英寸 (*)
ç (*)	t
(*)	u
英 (*)	ü (*)
ö (*)	v
英寸 (*)	y
ü (*)	z

注意：星號 (*) 表示土耳其語中的特殊字符。

如上表所示，在土耳其語語言排序中，特殊字符 ç 在 c 和 d 之間，但以二進制 UTF-8 順序顯示在 z 之後。此模式中的純量 Python UDF 使用下列字元取代字典，以對應的對應拉丁文字元取代土耳其文特殊字元。

土耳其特殊字符	拉丁等效字符
ç	c ‿
ı	H ‿

ğ	g ~
ö	o~
ş	s ~
ü	u~

注意：波浪符號 (~) 字元會附加到拉丁文字元的結尾，以取代其對應的土耳其文特殊字元。

修改一個標量 Python 的 UDF 函數

若要從這個模式修改純量 Python UDF 函式，讓函式接受定位參數並支援多重交易字典，請使用下列 SQL 程式碼：

```
CREATE OR REPLACE FUNCTION collate_sort (value varchar, locale varchar)
RETURNS varchar
IMMUTABLE
AS
$$
    def sort_str(val):
       import string
       # Turkish Dictionary
       if locale == 'tr-TR':
            dictionary = {
               'I': 'ı',
               'ı': 'h~',
               'İ': 'i',
               'Ş': 's~',
               'ş': 's~',
               'Ğ': 'g~',
               'ğ': 'g~',
               'Ü': 'u~',
               'ü': 'u~',
               'Ö': 'o~',
               'ö': 'o~',
               'Ç': 'c~',
               'ç': 'c~'
        }
        # German Dictionary
        if locale == 'de-DE':
            dictionary = {
```

```
                ....
                ....
        }

        for key, value in dictionary.items():
            val = val.replace(key, value)

        return val.lower()

    return sort_str(value)

$$ LANGUAGE plpythonu;
```

下面的示例代碼演示了如何查詢修改後的 Python UDF：

```
SELECT first_name FROM my_table ORDER BY collate_order(first_name, 'tr-TR');
```

訂閱 Lambda 函數，以便從不同 AWS 區域的 S3 儲存貯體發出的事件通知

由蘇雷什科納塔拉（AWS）和阿林多姆薩卡（AWS）創建

環境：生產	技術：分析	AWS 服務：AWS Lambda；Amazon S3；Amazon SNS；Amazon SQS

Summary

Amazon Simple Storage Service (Amazon S3) 事件通知會針對 S3 儲存貯體中的某些事件 (例如，物件建立的事件、物件移除事件或還原物件事件) 發佈通知。您可以使用 AWS Lambda 函數根據應用程式的需求來處理這些通知。但是，Lambda 函數無法直接訂閱來自不同 AWS 區域託管的 S3 儲存貯體的通知。

此模式的方法透過針對每個區域使用 Amazon 簡單通知服務 (Amazon SNS) 主題，部署散發案例以處理跨區域 S3 儲存貯體的 Amazon S3 通知。這些區域 SNS 主題會將 Amazon S3 事件通知傳送至也包含 Lambda 函數的中央區域中的 Amazon 簡單佇列服務 (Amazon SQS) 佇列。Lambda 函數會訂閱此 SQS 佇列，並根據您組織的需求處理事件通知。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 位於多個區域的現有 S3 儲存貯體，包括用於託管 Amazon SQS 佇列的中央區域和 Lambda 函數。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需有關這方面的詳細資訊，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI。
- 熟悉 Amazon SNS 中的扇出場景。如需有關這方面的詳細資訊，請參閱 Amazon SNS 文件中的常見 Amazon SNS 案例。

架構

下圖顯示了這種模式的方法的體系結構。

該圖顯示以下工作流程：

1. Amazon S3 會將有關 S3 儲存貯體的事件通知 (例如，建立的物件、移除物件或還原物件) 傳送至相同區域中的 SNS 主題。
2. SNS 主題會將事件發佈至中央區域的 SQS 佇列。
3. SQS 佇列會設定為 Lambda 函數的事件來源，並為 Lambda 函數緩衝事件訊息。
4. Lambda 函數會輪詢 SQS 佇列中是否有訊息，並根據您的應用程式需求處理 Amazon S3 事件通知。

技術, 堆

- Lambda
- Amazon SNS
- Amazon SQS
- Amazon S3

工具

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可透過命令列殼層中的命令與 AWS 服務互動。只要使用最少的組態，您就可以執行 AWS CLI 命令，從命令提示字元實作與以瀏覽器為基礎的 AWS 管理主控台所提供的功能相同。
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

- Amazon SQS — Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可讓您整合和分離分散式軟體系統和元件。Amazon SQS 同時支援標準佇列和 FIFO 佇列。

史诗

在您的中央區域建立 SQS 佇列和 Lambda 函數

任務	描述	所需技能
使用 Lambda 觸發程序建立 SQS 佇列。	登入 AWS 管理主控台，並在 AWS Lambda Lambda 文件中使用 Lambda 搭配 Amazon SQS 使用教學中的指示，在您的中央區域建立下列資源： • Lambda 執行角色 • 處理 Amazon S3 事件的 Lambda 函數 • 一個 SQS 隊列 備註：請務必將 SQS 佇列設定為 Lambda 函數的事件來源。	AWS DevOps、雲端架構師

建立 SNS 主題並為每個所需區域中的 S3 儲存貯體設定事件通知

任務	描述	所需技能
建立 SNS 主題以接收 Amazon S3 事件通知。	在您想要接收 Amazon S3 事件通知的區域中建立 SNS 主題。如需有關這方面的詳細資訊，請參閱 Amazon SNS 文件中的建立 SNS 主題。	AWS DevOps、雲端架構師

任務	描述	所需技能
	重要事項：請務必記錄 SNS 主題的 Amazon 資源名稱 (ARN)。	
將 SNS 主題訂閱到中央 SQS 佇列。	將您的 SNS 主題訂閱到中央區域託管的 SQS 佇列。如需有關此項目的詳細資訊，請參閱 Amazon SNS 文件中的訂閱 SNS 主題。	AWS DevOps、雲端架構師

任務	描述	所需技能
更新 SNS 主題的存取政策。	1. 開啟 Amazon SNS 主控台，選擇「主題」，然後選擇您先前建立的 SNS 主題。 2. 選擇 [編輯]，然後展開 [存取原則-選用] 區段。 3. 將下列存取政策附加到您的 SNS 主題以允許 Amazon S3 的`sns:publish`許可，然後選擇「儲存」： `{ "Version": "2012-10-17", "Statement": [ { "Sid": "0", "Effect": "Allow", "Principal": { "Service": "s3.amazonaws.com" }, "Action": "sns:Publish", "Resource": "arn:aws:sns:us-west-2::s3Events-SNSTopic-us-west-2" } ] }`	AWS DevOps、雲端架構師

任務	描述	所需技能
為區域中的每個 S3 儲存貯體設定通知。	為區域中的每個 S3 儲存貯體設定事件通知。如需這方面的詳細資訊，請參閱 Amazon S3 文件中的使用 Amazon S3 主控台啟用和設定事件通知。 注意：在「目的地」段落中，選擇 SNS 主題，然後指定您先前建立之 SNS 主題的 ARN。	AWS DevOps、雲端架構師
對所有必要的區域重複此史詩。	重要事項：針對您想要接收 Amazon S3 事件通知的每個區域 (包括您的中央區域) 重複此史詩中的任務。	AWS DevOps、雲端架構師

相關資源

- 設定存取政策 (Amazon SQS 文件)
- 將 SQS 佇列設定為事件來源 (AWS Lambda 文件)
- 設定 SQS 佇列以啟動 Lambda 函數 (Amazon SQS 文件)
- AWS::Lambda::Function 資源 (AWS CloudFormation 文件)

三種 AWS Glue ETL 任務類型，可將資料轉換為 Apache 實木地板

由阿德南阿爾維（AWS），卡爾蒂安·拉馬干德蘭和尼特·戈文達西文（AWS）創建

環境：PoC 或試點	技術：分析	工作負載：所有其他工作
AWS 服務：AWS AWS Glue		

Summary

在 Amazon Web Services (AWS) 雲端上，AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。AWS Glue 讓您能夠以符合成本效益的方式，將資料分類、清理、豐富資料，以及在各種資料存放區和資料串流之間可靠地移動資料。

此模式在 AWS Glue 中提供不同的任務類型，並使用三種不同的指令碼來示範編寫 ETL 任務。

您可以使用 AWS Glue 在 Python 殼層環境中撰寫 ETL 任務。您也可以在託管的 Apache 星火環境中使用 Python（PySpark）或斯卡拉創建批處理和流 ETL 任務。為了讓您開始編寫 ETL 作業，此模式著重於使用 Python 殼層和 Scala 的批次 ETL 作業。 PySparkPython 殼層作業適用於需要較低運算能力的工作負載。受管理的 Apache Spark 環境適用於需要高運算能力的工作負載。

阿帕奇實木複合地板是建立支持高效的壓縮和編碼方案。它可以加速您的分析工作負載，因為它以單欄式方式儲存資料。在長期運行中將數據轉換為 Parquet 可以節省您的存儲空間，成本和時間。要了解有關鑲木地板的更多信息，請參閱博客文章 Apache Parquet：如何使用開源柱狀數據格式成為英雄。

先決條件和限制

先決條件

- AWS Identity and Access Management (IAM) 角色 (如果您沒有角色，請參閱其他資訊一節)。

架構

目標技術堆疊

- AWS Glue
- Amazon Simple Storage Service (Amazon S3)

- Apache Parquet

自動化和規模

- AWS Glue 工作流程支援 ETL 管道的完全自動化。
- 您可以變更資料處理單位 (DPU) 或 Worker 類型的數目，以水平和垂直縮放。

工具

AWS 服務

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Glue 是全受管 ETL 服務，可在各種資料存放區和資料串流之間分類、清理、豐富和移動資料。

其他工具

- Apache 的實木複合地板是一種開源的面向列的數據文件格式，專為存儲和檢索。

組態

使用下列設定來設定 AWS Glue ETL 的運算能力。若要降低成本，請在執行此模式中提供的工作負載時使用最小設定。

- Python 外殼 — 您可以使用 1 個 DPU 來利用 16 GB 的記憶體，或使用 0.0625 個 DPU 來利用 1 GB 的記憶體。此模式使用 0.0625 DPU，這是 AWS Glue 主控台中的預設值。
- 適用於 Spark 的 Python 或 Scala — 如果您在主控台中選擇與火花相關的任務類型，AWS Glue 預設會使用 10 個工作程式和 G.1X 工作者類型。此模式使用兩個 Worker，這是允許的最小數目，具有標準 Worker 類型，這足夠且具有成本效益。

下表顯示 Apache Spark 環境的不同 AWS AWS Glue 工作者類型。因為 Python 殼層作業不使用 Apache 星火環境來執行 Python，所以它不會包含在資料表中。

	標準	G.1X	G.2X

vCPU	4	4	8
記憶體	16 GB	16 GB	32 GB
磁碟空間	50 GB	64 GB	128 GB
每名工人的執行人	2	1	1

Code

如需此模式中使用的程式碼 (包括 IAM 角色和參數設定)，請參閱其他資訊一節。

史诗

上傳資料

任務	描述	所需技能
將資料上傳到新的或現有的 S3 儲存貯體。	在您的帳戶中建立或使用現有的 S3 儲存貯體。從「附件」區段上傳 sample_data.csv 檔案，並記下 S3 儲存貯體和前置碼的位置。	一般 AWS

建立並執行 AWS AWS Glue 任務

任務	描述	所需技能
建立 AWS AWS Glue 任務。	在 AWS Glue 主控台的 ETL 區段下，新增 AWS AWS Glue 任務。選取適當的任務類型、AWS Glue 版本，以及對應的 DPU/ 工作者類型和工作者數目。如需詳細資訊，請參閱「組態」一節。	開發人員、雲端或資料

任務	描述	所需技能
更改輸入和輸出位置。	複製與 AWS Glue 任務對應的程式碼，然後變更您在上傳資料史詩中記下的輸入和輸出位置。	開發人員、雲端或資料
設定參數。	您可以使用「其他資訊」區段中提供的片段來設定 ETL 工作的參數。AWS Glue 在內部使用四個引數名稱： • `--conf` • `--debug` • `--mode` • `--JOB_NAME` 必須在 AWS Glue 主控台上明確輸入`--JOB_NAME` 參數。選擇 「Job」、「編輯工作」、「安全性組態」、「命令檔程式庫」和「工作參數」(選輸入`--JOB_NAME` 作為鍵並提供一個值。您也可以使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 或 AWS Glue API 來設定此參數。該`--JOB_NAME` 參數由星火使用，並且不需要在 Python 外殼環境作業。 您必須`--`在每個參數名稱之前加入；否則，程式碼將無法運作。例如，對於程式碼片段，位置參數必須由`--input_loc` 和叫用`--output_loc` 。	開發人員、雲端或資料

任務	描述	所需技能
執行 ETL 工作。	運行您的作業並檢查輸出。請注意原始檔案減少了多少空間。	開發人員、雲端或資料

相關資源

參考

- Apache Spark
- AWS AWS Glue：它是如何工作的
- AWS AWS Glue 定價

教學課程和影片

- 什麼是 AWS AWS Glue？

其他資訊

IAM 角色

建立 AWS Glue 任務時，您可以使用具有下列程式碼片段中顯示許可的現有 IAM 角色，或使用新角色。

若要建立新角色，請使用下列 YAML 程式碼。

```
# (c) 2022 Amazon Web Services, Inc. or its affiliates. All Rights Reserved. This AWS
 Content is provided subject to the terms of the AWS Customer
# Agreement available at https://aws.amazon.com/agreement/ or other written agreement
 between Customer and Amazon Web Services, Inc.

AWSTemplateFormatVersion: "2010-09-09"

Description: This template will setup IAM role for AWS Glue service.

Resources:
  rGlueRole:
```

```
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "glue.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
      Policies:
        - PolicyName: !Sub "${AWS::StackName}-s3-limited-read-write-inline-policy"
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - "s3:PutObject"
                  - "s3:GetObject"
                Resource: "arn:aws:s3:::*/*"
      Tags:
        - Key  : "Name"
          Value : !Sub "${AWS::StackName}"

Outputs:
  oGlueRoleName:
    Description: AWS Glue IAM role
    Value:
      Ref: rGlueRole
    Export:
      Name: !Join [ ":", [ !Ref "AWS::StackName", rGlueRole ] ]
```

AWS AWS Glue Python 殼

Python 代碼使用熊貓和 PyArrow 庫將數據轉換為實木複合地板。熊貓圖書館已經可用。當您執行病毒碼時，會下載程式 PyArrow 庫，因為這是一次性執行。您可以使用 wheel 檔案轉換 PyArrow 為資源庫，並將檔案做為資源庫套件提供。如需有關封裝輪子檔案的詳細資訊，請參閱提供您自己的 Python 程式庫。

AWS Glue Python 外殼參數

```
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["input_loc", "output_loc"])
```

AWS Glue Python 殼代碼

```
from io import BytesIO
import pandas as pd
import boto3
import os
import io
import site
from importlib import reload
from setuptools.command import easy_install
install_path = os.environ['GLUE_INSTALLATION']
easy_install.main( ["--install-dir", install_path, "pyarrow"] )
reload(site)
import pyarrow


input_loc = "bucket-name/prefix/sample_data.csv"
output_loc = "bucket-name/prefix/"


input_bucket = input_loc.split('/', 1)[0]
object_key = input_loc.split('/', 1)[1]


output_loc_bucket = output_loc.split('/', 1)[0]
output_loc_prefix = output_loc.split('/', 1)[1]


s3 = boto3.client('s3')
obj = s3.get_object(Bucket=input_bucket, Key=object_key)
df = pd.read_csv(io.BytesIO(obj['Body'].read()))


parquet_buffer = BytesIO()
s3_resource = boto3.resource('s3')
df.to_parquet(parquet_buffer, index=False)
s3_resource.Object(output_loc_bucket, output_loc_prefix +  'data' +
 '.parquet').put(Body=parquet_buffer.getvalue())
```

使用 Python 的 AWS AWS Glue 火花任務

若要搭配 Python 使用 AWS AWS Glue 星火工作類型，請選擇星火做為任務類型。選擇火花 3.1、Python 3，並改善任務啟動時間 (Glue 3.0 版) 做為 AWS AWS Glue 版本。

AWS Glue Python 參數

```
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME", "input_loc", "output_loc"])
```

使用 Python 代碼進行 AWS AWS Glue 火花任務

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import *
from awsglue.dynamicframe import DynamicFrame
from awsglue.utils import getResolvedOptions
from awsglue.job import Job


sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

input_loc = "bucket-name/prefix/sample_data.csv"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(\
    connection_type = "s3", \
    connection_options = {
        "paths": [input_loc]}, \
    format = "csv",
    format_options={
        "withHeader": True,
        "separator": ","
    })


outputDF = glueContext.write_dynamic_frame.from_options(\
    frame = inputDyf, \
```

```
    connection_type = "s3", \
    connection_options = {"path": output_loc \
        }, format = "parquet")
```

對於大量壓縮的大型檔案 (例如，1,000 個檔案大約 3 MB)，請使用compressionType參數搭配recurse參數來讀取前置碼內可用的所有檔案，如下列程式碼所示。

```
input_loc = "bucket-name/prefix/"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
                    connection_type = "s3",
                    connection_options = {"paths": [input_loc],
                                          "compressionType":"gzip","recurse" :"True",
                                          },
                    format = "csv",
                    format_options={"withHeader": True,"separator": ","}
                    )
```

對於大量壓縮的小檔案 (例如，1,000 個檔案大約 133 KB)，請使用groupFiles參數以compressionType及和recurse參數。groupFiles參數會將小型檔案群組成多個大檔案，而groupSize參數則控制群組為以位元組為單位的指定大小 (例如 1 MB)。下列程式碼片段提供在程式碼中使用這些參數的範例。

```
input_loc = "bucket-name/prefix/"
output_loc = "bucket-name/prefix/"

inputDyf = glueContext.create_dynamic_frame_from_options(
                    connection_type = "s3",
                    connection_options = {"paths": [input_loc],
                                          "compressionType":"gzip","recurse" :"True",
                                           "groupFiles" :"inPartition",
 "groupSize" :"1048576",
                                          },
                    format = "csv",
                    format_options={"withHeader": True,"separator": ","}
                    )
```

這些設定可讓 AWS Glue 任務讀取多個檔案 (大或小，無論壓縮或不含壓縮)，並以 Parquet 格式將檔案寫入目標中，無論是否有任何變更，都可以使用 Parquet 格式將檔案寫入目標。

AWS AWS Glue 火花與斯卡拉的任務

若要搭配 Scala 使用 AWS AWS Glue 星火工作類型，請選擇 S park 作為任務類型，選擇語言為 Scala。選擇 S park 3.1，斯卡拉 2 改善任務啟動時間 (Glue 版本 3.0) 做為 AWS AWS Glue 版本。為了節省儲存空間，下列 AWS Glue 與 Scala 範例也使用此`applyMapping`功能來轉換資料類型。

AWS AWS Glue 斯卡拉參數

```
import com.amazonaws.services.glue.util.GlueArgParser val args =
 GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME", "inputLoc",
 "outputLoc").toArray)
```

AWS AWS Glue 星火任務與斯卡拉代碼

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.DynamicFrame
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueScalaApp {
  def main(sysArgs: Array[String]) {

    @transient val spark: SparkContext = SparkContext.getOrCreate()
    val glueContext: GlueContext = new GlueContext(spark)

    val inputLoc = "s3://bucket-name/prefix/sample_data.csv"
    val outputLoc = "s3://bucket-name/prefix/"

    val readCSV = glueContext.getSource("csv", JsonOptions(Map("paths" ->
 Set(inputLoc)))).getDynamicFrame()

    val applyMapping = readCSV.applyMapping(mappings = Seq(("_c0", "string", "date",
 "string"), ("_c1", "string", "sales", "long"),
    ("_c2", "string", "profit", "double")), caseSensitive = false)

    val formatPartition = applyMapping.toDF().coalesce(1)

    val dynamicFrame = DynamicFrame(formatPartition, glueContext)
```

```
    val dataSink = glueContext.getSinkWithFormat(
        connectionType = "s3",
        options = JsonOptions(Map("path" -> outputLoc )),
        transformationContext = "dataSink", format =
 "parquet").writeDynamicFrame(dynamicFrame)
  }
}
```

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon 雅典娜和亞馬遜視覺化亞馬遜 Redshift 審核日 QuickSight

創建者：桑凱特·蘇斯卡（AWS）和戈帕爾克里希納·巴蒂亞（AWS）

環境：PoC 或試點	技術：分析、大數據、資料湖	AWS 服務：Amazon Athena; Amazon Redshift; Amazon S3; Amazon QuickSight

Summary

安全性是 Amazon Web Services (AWS) 雲端上資料庫操作不可或缺的一部分。您的組織應確保其監控資料庫使用者活動和連線，以偵測潛在的安全事件和風險。此病毒碼可協助您監視資料庫的安全性和疑難排解目的，這是一種稱為資料庫稽核的處理程序。

此模式提供 SQL 指令碼，可自動建立 Amazon Athena 表格和 Amazon 中報告儀表板的檢視，QuickSight 以協助您稽核 Amazon Redshift 日誌。這可確保負責監視資料庫活動的使用者可以方便地存取資料安全性功能。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的 Amazon Redshift 叢集。如需有關此功能的詳細資訊，請參閱 Amazon Redshift 文件中的建立 Amazon Redshift 叢集。
- 存取現有的 Athena 工作群組。如需詳細資訊，請參閱 Amazon Athena 文件中的工作群組如何運作。
- 具有所需 AWS Identity and Access Management (IAM) 許可的現有 Amazon 簡單儲存貯體 (Amazon S3) 來源儲存貯體。如需詳細資訊，請參閱 Amazon Redshift 說明文件中的資料庫稽核記錄的 Amazon Redshift 稽核記錄的儲存貯體許可。

架構

技術, 堆

- Athena
- Amazon Redshift
- Amazon S3
- QuickSight

工具

- Amazon Athena 娜 — Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon S3 中的資料。
- Amazon QuickSight — QuickSight 是可擴展的無伺服器、可嵌入式、機器學習支援的商業智慧 (BI) 服務。
- Amazon Redshift — Amazon Redshift 是一種企業級、PB 級規模的全受管資料倉儲服務。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。

史诗

設定 Amazon Redshift 叢集

任務	描述	所需技能
啟用 Amazon Redshift 叢集的稽核記錄。	1. 登入 AWS 管理主控台，開啟 Amazon Redshift 主控台，選擇 [叢集]，然後選擇要啟用記錄的叢集。 2. 選擇「內容」索引標籤，然後遵循 Amazon Redshift 文件中使用主控台設定稽核中的指示來啟用稽核。	DBA, 資料工程師
在 Amazon Redshift 叢集參數群組中啟用記錄功能。	您可以使用 AWS 管理主控台、Amazon Redshift API 參考或 AWS Command Line Interface (AWS CLI) (AWS	DBA, 資料工程師

任務	描述	所需技能
	CLI)，同時啟用連線日誌、使用者日誌和使用者活動日誌的稽核功能。 若要稽核使用者活動記錄檔，您必須啟用資`enable_user_activity_logging` 料庫參數。如果您只啟用稽核記錄功能而不啟用相關參數，則資料庫稽核會記錄連線和使用者記錄的記錄資訊，但不會記錄使用者活動記錄的記錄資訊。依預設，`enable_user_activity_logging` 參數不會啟用，但您可以透過`false`將其從變更為來啟用此參數`true`。 重要事項：您需要在啟用該參數的情況下建立新的叢集`user_activity_logging` 參數群組，並將其附加到 Amazon Redshift 叢集。如需這方面的詳細資訊，請參閱修改叢集中的 Amazon Redshift 文件。 如需有關此任務的詳細資訊，請參閱 Amazon Redshift 參數群組和 Amazon Redshift 說明文件中的使用主控台設定稽核。	

任務	描述	所需技能
為 Amazon Redshift 叢集記錄設定 S3 儲存貯體許可。	啟用日誌記錄時，Amazon Redshift 會收集日誌資訊，並將其上傳到存放在 S3 儲存貯體的日誌檔。您可以使用現有的 S3 儲存貯體或建立新儲存貯體。 重要事項：請確定 Amazon Redshift 具有存取 S3 儲存貯體所需的 IAM 許可。如需這方面的詳細資訊，請參閱 Amazon Redshift 說明文件中的資料庫稽核記錄所提供的 Amazon Redshift 稽核記錄的儲存貯體許可。	DBA, 資料工程師

建立 Athena 表格和檢視

任務	描述	所需技能
建立 Athena 表格和檢視，以查詢 S3 儲存貯體中的 Amazon Redshift 稽核日誌資料。	開啟 Amazon Athena 主控台，然後使用 `AuditLogging.sql` SQL 指令碼 (附加) 中的資料定義語言 (DDL) 查詢來建立使用者活動日誌、使用者日誌和連線日誌的表格和檢視。 如需詳細資訊和指示，請參閱 Amazon Athena 工作坊中的建立資料表和執行查詢教學課程。	數據工程師

在 QuickSight 儀表板中設置日誌監控

任務	描述	所需技能
使用 Athena 做為資料來源建立 QuickSight 儀表板。	開啟 Amazon QuickSight 主控台，然後按照 Amazon Athena Athena 工作坊的視覺化 QuickSight 使用雅典娜教學中的指示，建立 QuickSight 儀表板。	DBA, 資料工程師

相關資源

- 在 Athena 中建立資料表並執行查詢
- QuickSight 使用 Athena 視覺化

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon 將所有 AWS 帳戶的 IAM 登入資料報告視覺化 QuickSight

創建者：帕拉格納格韋卡 (AWS) 和阿倫·錢達皮萊 (AWS)

程式碼儲存庫：讓您的 IAM 登入資料報告全組織能見度	環境：生產	技術：分析；諮詢；管理與治理；安全性、身分識別、合規
工作負載：所有其他工作	AWS 服務：Amazon Athena；AWS CloudFormation；Amazon EventBridge；AWS Identity and Access Management；Amazon QuickSight	

Summary

警告：IAM 使用者擁有長期登入資料，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。

您可以使用 AWS Identity and Access Management (IAM) 登入資料報告，協助您滿足組織的安全、稽核和合規要求。認證報告提供 AWS 帳戶中所有使用者的清單，並顯示其登入資料的狀態，例如密碼、存取金鑰和多因素身份驗證 (MFA) 裝置。您可以針對由 AWS AWS Organ izations 管理的多個 AWS 帳戶使用登入資料報告。

此模式包含步驟和程式碼，可協助您使用 Amazon QuickSight 儀表板為組織中所有 AWS 帳戶建立和共用 IAM 登入資料報告。您可以與組織中的利害關係人共用儀表板。這些報告可協助您的組織達成下列目標業務成果：

- 識別與 IAM 使用者相關的安全事件
- 追蹤 IAM 使用者至單一登入 (SSO) 身份驗證的即時遷移
- 追蹤 IAM 使用者存取的 AWS 區域
- 保持合規
- 與其他利益相關者分享資

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 擁有成員帳戶的組織
- 具有存取 Organizations 帳戶權限的 IAM 角色
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已安裝和設定
- 訂閱 Amazon QuickSight 企業版

架構

技術, 堆

- Amazon Athena
- Amazon EventBridge
- Amazon QuickSight
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue
- AWS Identity and Access Management (IAM)
- AWS Lambda
- AWS Organizations

目標架構

下圖顯示設定工作流程的架構，該架構可擷取來自多個 AWS 帳戶的 IAM 登入資料報告資料。

1. EventBridge 每天調用一個 Lambda 函數。
2. Lambda 函數會在整個組織的每個 AWS 帳戶中擔任 IAM 角色。然後，該函數會建立 IAM 登入資料報表，並將報表資料存放在集中式 S3 儲存貯體中。您必須在 S3 儲存貯體上啟用加密並停用公開存取。
3. AWS Glue 爬行程式每天都會檢索 S3 儲存貯體，並相應地更新 Athena 表。
4. QuickSight 匯入和分析認證報告中的資料，並建立可供利益相關者視覺化並與其共用的儀表板。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可使用標準 SQL 輕鬆分析 Amazon S3 中的資料。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，Lambda 函數、使用 API 目標的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- Amazon QuickSight 是雲端規模商業智慧 (BI) 服務，可協助您在單一儀表板中視覺化、分析和報告資料。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

Code

此模式的代碼可在 GitHub getiamcredsreport-allaccounts-org存儲庫中找到。您可以使用此儲存庫中的程式碼，跨 Organizations 中的 AWS 帳戶建立 IAM 登入資料報告，並將其存放在中央位置。

史诗

設定基礎架構

任務	描述	所需技能
設置 Amazon QuickSight 企業版。	1. 在您的 AWS 帳戶中啟用 Amazon QuickSight 企業版。如需詳細資訊，請參閱 QuickSight 文件 QuickSight 中的管理 Amazon 內部的使用者存取權限。 2. 若要授予儀表板許可，請取得 QuickSight 使用者的 Amazon 資源名稱 (ARN)。	AWS 管理員、AWS DevOps、雲端管理員、雲端架構師

任務	描述	所需技能
將 Amazon QuickSight 與 Amazon S3 和 Athena 集成。	在部署 AWS CloudFormation 堆疊之前，您必須先授權QuickSight 使用 Amazon S3 和 Athena。	AWS 管理員、AWS DevOps、雲端管理員、雲端架構師

部署基礎架構

任務	描述	所需技能
克隆存 GitHub 儲庫。	1. 執行下列命令，將 GitHub getiamcredsreport-allaccounts-org存放庫複製到您的本機電腦：`git clone https://github.com/aws-samples/getiamcredsreport-allaccounts-org`	AWS 管理員
部署基礎結構。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 在瀏覽窗格中，選擇 [建立堆疊]，然後選擇 [使用新資源 (標準)]。 3. 在 [識別資源] 頁面上，選擇 [下一步]。 4. 在 [指定範本] 頁面上，對於 [範本來源]，選取 [上傳範本檔案]。 5. 選擇 [選擇檔案]，從複製的 GitHub 儲存庫中選取`Cloudformation-cre`	AWS 管理員

任務	描述	所需技能
	atecredrepo.yaml 檔案，然後選擇 [下一步]。 6. 在參數中，使IAMRoleName 用您的 IAM 角色進行更新。 這應該是您希望 Lambda 在組織的每個帳戶中承擔的 IAM 角色。此角色會建立認證報告。注意：在堆疊建立的這個步驟中，角色不一定要出現在所有帳戶中。 7. 在參數中，更新 S3BucketName S3 儲存貯體的名稱，Lambda 可以在其中存放所有帳戶的登入資料。 8. 在堆疊名稱中，輸入您的堆疊名稱。 9. 選擇提交。 10請注意 Lambda 函數的角色名稱。	

任務	描述	所需技能
建立 IAM 權限政策。	使用下列許可為組織中的每個 AWS 帳戶建立 IAM 政策： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:GenerateCredentialReport", "iam:GetCredentialReport" ], "Resource": "*" } ] }`	AWS DevOps、雲端管理員、雲端架構師、資料工程師

任務	描述	所需技能
建立具有信任政策的 IAM 角色。	1. 為 AWS 帳戶建立 IAM 角色，並附加您在上一步中建立的許可政策。 2. 將下列信任政策附加到 IAM 角色： `{ "Version":"2012-10-17", "Statement":[ { "Effect": "Allow", "Principal":{ "AWS":[ "arn:aws:iam::<MasterAccountID>:role/<LambdaRole>" ] }, "Action": "sts:AssumeRole" } ] }` 重要事項：請`arn:aws:iam::<MasterAccountID>:role/<LambdaRole>` 以您先前記下的 Lambda 角色的 ARN 取代。 注意：Organizations 通常使用自動化來為其 AWS 帳戶建立 IAM 角色。我們建議您使用此自動化操作 (如果有的話)。或	雲端管理員、雲端架構師、AWS 管理員

任務	描述	所需技能
	者，您可以使用代碼存儲庫中的`CreateRoleforOrg.py` 腳本。指令碼需要現有的管理角色或任何其他具有在每個 AWS 帳戶建立 IAM 政策和角色權限的 IAM 角色。	
配置 Amazon QuickSight 以可視化數據。	1. QuickSight使用您的認證登入。 2. 使用 Ath@@ ena 建立資料集 (使用資料`iamcredreportdb` 庫和資料“`cfn_iamcredreport`” 表)，然後自動重新整理資料集。 3. 在中建立分析 QuickSight。 4. 建立 QuickSight 儀表板。	AWS DevOps、雲端管理員、雲端架構師、資料工程師

其他資訊

其他考量

考慮下列各項：

- 使用 CloudFormation 基礎設施部署後，您可以等待在 Amazon S3 中建立並由 Athena 進行分析的報告，直到 Lambda 和 AWS Glue 按照其排程執行為止。或者，您也可以手動執行 Lambda 以取得 Amazon S3 中的報告，然後執行 AWS Glue 爬行程式以取得從資料建立的 Athena 表格。
- QuickSight 是一個功能強大的工具，可根據您的業務需求分析和視覺化資料。您可以使用中的參數，根據您選擇的資料欄位 QuickSight 來控制 Widget 資料。此外，您也可以使用 QuickSight 分析從資料集建立參數 (例如「帳戶」、「日期」和「使用者」欄位 `partition_0`、`partition_1`，例如、和`user`)，以新增「帳戶」、「日期」和「使用者」參數的控制項。
- 若要建立自己的 QuickSight 儀表板，請參閱 AWS 工作坊工作室網站上的QuickSight 研討會。
- 若要查看範例 QuickSight 儀表板，請參閱程 GitHub getiamcredsreport-allaccounts-org式碼儲存庫。

目標業務成果

您可以使用此模式來實現以下目標業務成果：

- 識別與 IAM 使用者相關的安全事件 — 使用單一窗格調查組織中每個 AWS 帳戶中的每個使用者。您可以追蹤 IAM 使用者最近存取的個別 AWS 區域的趨勢，以及他們使用的服務。
- 追蹤 IAM 使用者至 SSO 身份驗證的即時遷移 — 使用者可以使用單一登入資料登入一次，並存取多個 AWS 帳戶和應用程式。如果您打算將 IAM 使用者遷移到 SSO，此模式可協助您轉換為 SSO，並追蹤所有 AWS 帳戶的所有 IAM 使用者登入資料使用情況 (例如存取 AWS 管理主控台或存取金鑰的使用情況)。
- 追蹤 IAM 使用者存取的 AWS 區域 — 您可以基於各種目的控制 IAM 使用者對區域的存取權限，例如資料主權和成本控制。您也可以追蹤任何 IAM 使用者對區域的使用情況。
- 保持合規 — 遵循最低權限原則，您可以僅授予執行特定任務所需的特定 IAM 許可。此外，您還可以追蹤 AWS 服務、AWS 管理主控台和長期登入資料使用情況。
- 與其他利益相關者共用資訊 — 您可以與其他利益相關者共用精選儀表板，而無需授予他們 IAM 登入資料報告或 AWS 帳戶的存取權。

更多模式

- ???
- 使用亞馬遜文本提取自動從 PDF 文件中提取內容
- 使用 AWS DataOps 開發套件建立資料管道以擷取、轉換和分析 Google 分析資料
- ???
- 使用 AWS IoT 資料以符合成本效益的方式，將物聯網資料直接導入 Amazon S3
- 使用 AWS Cost Explorer 為 Amazon EMR 叢集建立詳細的成本和用量報告
- 為 Amazon RDS 和 Amazon Aurora 創建詳細的成本和用量報告
- 使用 AWS Cost Explorer 為 AWS Glue 任務建立詳細的成本和用量報告
- 跨帳戶資料共用自動化
- 使用基礎設施即程式碼在 AWS 雲端部署和管理無伺服器資料湖
- 在本地角度應用程序中嵌入 Amazon QuickSight 儀表板
- 確保亞 Amazon Redshift 叢集在建立時已加密
- 確保啟動時已啟用 Amazon EMR 靜態資料的加密
- 擷取和查詢資料湖中的 AWS IoT 中 SiteWise 繼資料屬性
- 在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight
- 讓 SageMaker 筆記本執行個體暫時存取另一個 AWS 帳戶中的 CodeCommit 儲存庫
- 在未使用 AWS KMS 金鑰加密 Amazon 資料 Firehose 資源時識別並發出警示
- 將自我託管的 MongoDB 環境遷移到 AWS 雲端上的 MongoDB 地圖集
- 使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移
- 使 DistCp 用 PrivateLink 適用於 Amazon S3 的 AWS，將資料從現場部署 Hadoop 環境遷移到 Amazon S3
- ???
- 將現場部署工作負載遷移到 AWS 上的 Cloudera 資料平台
- 在啟動時監控 Amazon EMR 叢集的傳輸中加密
- 為 AWS 設定 Grafana 監控儀表板 ParallelCluster
- 確認新的 Amazon Redshift 叢集具有必要的 SSL 端點
- 確認新的 Amazon Redshift 叢集是否在 VPC 中啟動
- ???

企業生產力

主題

- 在 AWS 上設定高可用性 PeopleSoft 架構
- 更多模式

在 AWS 上設定高可用性 PeopleSoft 架構

環境：生產	技術：企業生產力; 基礎架構; Web 和移動應用程序; 數據庫	工作量：甲骨文
AWS 服務：Amazon EC2 Auto Scaling；Amazon EFS；Elastic Load Balancing（ELB）；Amazon RDS		

Summary

將 PeopleSoft 工作負載遷移到 AWS 時，備援是一個重要的目標。它可確保您的 PeopleSoft 應用程式始終具有高可用性，並且能夠快速從故障中復原。

此模式為 AWS 上的 PeopleSoft 應用程式提供架構，以確保網路、應用程式和資料庫層的高可用性 (HA)。它使用 Amazon Relational Database Service（Amazon RDS）的甲骨文或 Amazon RDS for SQL Server 數據庫的數據庫層。此架構還包括 AWS 服務，例如 Amazon Route 53、亞馬遜彈性運算雲端 (Amazon EC2) Linux 執行個體、Amazon Elastic Block Storage (Amazon EBS)、Amazon Elastic File System (Amazon EFS) 和 Ap plication Load Balancer，並具有可擴展性。

Oracle PeopleSoft 為人力管理和其他業務營運提供了一套工具和應用程式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有在 AWS 上設定所需授權的 PeopleSoft 環境
- 使用下列資源在 AWS 帳戶中設定的虛擬私有雲端 (VPC)：
  - 至少兩個可用區域
  - 每個可用區域中有一個公用子網路和三個私有子網路
  - NAT 閘道和網際網路閘道
  - 用於路由流量的每個子網路的路由表

- 定義的網路存取控制清單 (網路 ACL) 和安全群組，以協助確保符合貴組織標準的 PeopleSoft 應用程式安全

限制

- 此模式提供高可用性 (HA) 解決方案。它不支援災難復原 (DR) 案例。在罕見的情況下，用於 HA 實作的整個 AWS 區域發生故障時，應用程式將無法使用。

產品版本

- PeopleSoft 應用程式執行 PeopleTools 8.52 及更新版本

架構

目標架構

PeopleSoft 生產應用程式的停機或中斷會影響應用程式的可用性，並對您的業務造成重大中斷。

我們建議您設計 PeopleSoft 生產應用程式，使其始終具有高可用性。您可以通過消除單點故障，添加可靠的交叉或故障轉移點以及檢測故障來實現這一目標。下圖說明適用於 AWS PeopleSoft 的 HA 架構。

此架構部署使用適用於甲骨文的 Amazon RDS 做為 PeopleSoft 資料庫，以及在 RHEL (RHEL) 上執行的 EC2 執行個體。您也可以使用 Amazon RDS for SQL Server 做為人民資料庫。

此架構包含下列元件：

- Amazon Route 53 用作網域名稱伺服器 (DNS)，用於將請求從網際網路路由到 PeopleSoft 應用程式。
- AWS WAF 可協助您防範可能會影響可用性、危及安全性或耗用過多資源的常見 Web 入侵程式和機器人。AWS Shield 進階 (未說明) 提供更廣泛的保護。
- 應用程式負載平衡器會透過針對 Web 伺服器的進階要求路由，來平衡 HTTP 和 HTTPS 流量的負載。
- 支援應用程式的網路伺服器、應用程式伺服器、程序排程器伺服器和 Elasticsearch 伺服器會在多個可 PeopleSoft 用區域中執行，並使用 Amazon EC2 Auto Scaling。

- 應用 PeopleSoft 程式使用的資料庫會以異地同步備份組態在 Amazon RDS 上執行。
- 應用程式使用的檔案共 PeopleSoft 用是在 Amazon EFS 上設定的，可用來跨執行個體存取檔案。
- Amazon EC2 Auto Scaling 使用亞馬遜機器映像 (AMI)，以確保在需要時快速複製 PeopleSoft 元件。
- NAT 閘道會將私有子網路中的執行個體連線到 VPC 外部的服務，並確保外部服務無法起始與這些執行個體的連線。
- 網際網路閘道是水平擴充、備援且高可用性的 VPC 元件，可讓您的 VPC 與網際網路之間進行通訊。
- 公用子網路中的堡壘主機可讓您從外部網路 (例如網際網路或內部部署網路) 存取私人子網路中的伺服器。堡壘主機提供對私有子網路中伺服器的控制和安全存取。

架構, 細節

該 PeopleSoft 資料庫位於異地同步備份組態中的 Amazon RDS for Oracle 文 (或 Amazon RDS for SQL Server) 資料庫中。Amazon RDS 異地同步備份功能可跨兩個可用區域複寫資料庫更新，以提高耐用性和可用性。Amazon RDS 會自動容錯移轉到待命資料庫，以進行計劃的維護和意外中斷。

PeopleSoft Web 和中間層安裝在 EC2 執行個體上。這些執行個體分散在多個可用區域，並由 Auto Scaling 群組綁定。這可確保這些元件始終具有高可用性。系統會維持最少數目的必要例證，以確保應用程式始終可用，並可在需要時進行調整。

建議您針對 OEM EC2 執行個體使用目前一代的 EC2 執行個體類型。目前一代的執行個體類型 (例如，在 AWS Nitro 系統上建置的執行個體) 支援硬體虛擬機器 (HVM)。HVM AMI 必須利用增強型網路功能，而且還提供更高的安全性。屬於每個 Auto Scaling 群組的 EC2 執行個體在更換或擴展執行個體時，會使用自己的 AMI。建議您根據您希望應用程式處理的負載，以及 Oracle 針對 PeopleSoft 應用程式和 PeopleTools 版本建議的最小值來選取 EC2 執行個體類型。 PeopleSoft 如需有關硬體和軟體需求的詳細資訊，請參閱 Oracle 客戶服務中心網站。

PeopleSoft Web 層和中間層共用 Amazon EFS 掛載，以共用報告、資料檔案和 (如果需要) 目`PS_HOME`錄。基於效能和成本原因，Amazon EFS 在每個可用區域設定掛接目標。

系統會佈建「Application Load Balancer 器」，以支援存取 PeopleSoft 應用程式的流量，並對跨不同可用區域之 Web 伺服器之間的流量進行負載平衡。應用程式負載平衡器是在至少兩個可用區域中提供 HA 的網路裝置。Web 伺服器會使用負載平衡組態，將流量分配到不同的應用程式伺服器。Web 伺服器和應用程式伺服器之間的負載平衡可確保負載平均分散到各個執行個體，並協助避免因執行個體超載而導致的瓶頸和服務中斷。

Amazon Route 53 用作 DNS 服務，用於將流量從網際網路路由到 Application Load Balancer。Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

醫管局詳情

- 資料庫：Amazon RDS 的異地同步備份功能使用同步複寫在多個可用區域中操作兩個資料庫。這會建立具有自動容錯移轉的高可用性環境。Amazon RDS 具有容錯移轉事件偵測功能，並在這些事件發生時啟動自動容錯移轉。您也可以透過 Amazon RDS API 啟動手動容錯移轉。如需詳細說明，請參閱部落格文章 Amazon RDS 引擎蓋下：異地同步備份。容錯移轉是無縫的，應用程式會在發生時自動重新連線到資料庫。不過，容錯移轉期間的任何處理序排程器工作都會產生錯誤，而且必須重新提交。
- PeopleSoft 應用程式伺服器：應用程式伺服器分散在多個可用區域，並為其定義了 Auto Scaling 群組。如果執行個體失敗，Auto Scaling 群組會立即使用從應用程式伺服器範本 AMI 複製的運作狀態良好的執行個體取代該執行個體。具體而言，啟用了 Jolt 共用，因此當應用程式伺服器執行個體停止運作時，工作階段會自動容錯移轉到另一個應用程式伺服器，而 Auto Scaling 群組會自動啟動另一個執行個體、啟動應用程式伺服器，並將其註冊到 Amazon EFS 掛載中。新建立的應用程式伺服器會使用 Web 伺服器中的`PSSTRSETUP.SH`指令碼自動新增至 Web 伺服器。這樣可確保應用程式伺服器始終具備高可用性，並可快速從失敗中復原。
- 程序排程器：程序排程器伺服器分散在多個可用區域，並且已為其定義 Auto Scaling 群組。如果執行個體失敗，Auto Scaling 群組會立即使用從處理序排程器伺服器範本 AMI 複製的運作狀態良好的執行個體取代該執行個體。具體來說，當程序排程器執行個體停止運作時，Auto Scaling 群組會自動啟動另一個執行個體並啟動程序排程器。執行個體失敗時執行的任何工作都必須重新提交。這樣可確保處理程序排程器始終具有高可用性，並可快速從失敗中復原。
- 彈性搜尋伺服器：Elasticsearch 伺服器具有為其定義的「Auto Scaling」群組。如果執行個體失敗，Auto Scaling 群組會立即使用從 Elasticsearch 伺服器範本 AMI 複製的運作狀態良好的執行個體取代該執行個體。具體來說，當 Elasticsearch 執行個體發生故障時，提供要求的 Application Load Balancer 會偵測失敗並停止向其傳送流量。「自 Auto Scaling 例」群組會自動啟動另一個執行個體，並顯示 Elasticsearch 執行個體。Elasticsearch 執行個體備份後，Application Load Balancer 會偵測到其狀態良好，並開始再次向其傳送要求。這樣可確保 Elasticsearch 伺服器始終具有高可用性，並可快速從故障中復原。
- Web 伺服器：Web 伺服器已為其定義了「Auto Scaling」群組。如果執行個體失敗，Auto Scaling 群組會立即使用從 Web 伺服器範本 AMI 複製的運作狀態良好的執行個體取代它。具體來說，當 Web 伺服器執行個體發生故障時，提供要求的 Application Load Balancer 器會偵測失敗並停止向其傳送流量。Auto Scaling 群組會自動啟動另一個執行個體，並啟動 Web 伺服器執行個體。當 Web 伺服器執行個體備份時，Application Load Balancer 器會偵測到它是否正常，並開始再次向其傳送要求。這樣可確保 Web 伺服器始終具有高可用性，並可快速從失敗中復原。

工具

AWS 服務

- 應用程式負載平衡器將傳入的應用程式流量分配到多個可用區域中的多個目標，例如 EC2 執行個體。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

最佳實務

營運最佳做法

- PeopleSoft 在 AWS 上執行時，請使用 Route 53 從網際網路和本機路由流量。如果主要資料庫執行個體無法使用，請使用容錯移轉選項將流量重新路由到災難復原 (DR) 站台。
- 請務必在 PeopleSoft 環境前使用 Application Load Balancer。這樣可以確保流量以安全的方式與 Web 服務器進行負載平衡。
- 在「Application Load Balancer」目標群組設定中，確定已透過負載平衡器產生的 Cookie 開啟黏性。

  注意：如果您使用外部單一登入 (SSO)，您可能需要使用應用程式型 Cookie。這可確保 Web 伺服器和應用程式伺服器之間的連線保持一致。
- 對於 PeopleSoft 生產應用程式，應用程式負載平衡器閒置逾時必須與您使用的 Web 設定檔中設定的值相符。這可防止使用者工作階段在負載平衡器層過期。
- 對於 PeopleSoft 生產應用程式，請將應用程式伺服器回收計數設定為可將記憶體洩漏降到最低的值。
- 如果您將 Amazon RDS 資料庫用於 PeopleSoft 生產應用程式 (如此模式中所述)，請以異地同步備份格式執行該資料庫以取得高可用性。

- 如果您的資料庫在 PeopleSoft 生產應用程式的 EC2 執行個體上執行，請確定待命資料庫正在另一個可用區域上執行，以取得高可用性。
- 對於 DR，請確保您的 Amazon RDS 資料庫或 EC2 執行個體已在與生產資料庫不同的 AWS 區域中設定備用。這樣可以確保在該地區發生災難時，您可以將應用程序切換到另一個區域。
- 對於 DR，請使用 Amazon 彈性災難復原在不同的區域中設定應用程式層級元件，以及生產元件。這樣可以確保在該地區發生災難時，您可以將應用程序切換到另一個區域。
- 使用 Amazon EFS (適用於中等 I/O 需求) 或 Amazon FSx (針對高 I/O 需求) 來存放您的 PeopleSoft 報告、附件和資料檔案。這樣可確保內容儲存在一個集中位置，而且可以從基礎結構內的任何位置存取。
- 使用 Amazon CloudWatch (基本和詳細資訊) 以近乎即時的方式監控 PeopleSoft 應用程式正在使用的 AWS 雲端資源。這樣可確保您立即收到問題的警示，並且可以在問題影響環境的可用性之前快速解決問題。
- 如果您使用 Amazon RDS 資料庫做為資料 PeopleSoft 庫，請使用增強型監控。此功能可存取超過 50 個指標，包括 CPU、記憶體、檔案系統 I/O 和磁碟 I/O。
- 使用 AWS CloudTrail 監控 PeopleSoft 應用程式正在使用的 AWS 資源上的 API 呼叫。這可協助您執行安全性分析、資源變更追蹤及法規遵循稽核。

安全性最佳做法

- 若要保護您的 PeopleSoft 應用程式免受常見的入侵，例如 SQL 插入或跨網站指令碼 (XSS)，請使用 AWS WAF。請考慮使用 AWS Shield 進階來量身打造的偵測和緩解服務。
- 將規則新增至 Application Load Balancer，以自動將流量從 HTTP 重新導向至 HTTPS，以協助保護您的 PeopleSoft 應用程式。
- 為應用程式負載平衡器設定個別的安全性群組。此安全性群組應該只允許 HTTPS/HTTP 輸入流量，而不允許輸出流量。這可確保只允許預期的流量，並有助於保護您的應用程式。
- 針對應用程式伺服器、Web 伺服器和資料庫使用私人子網路，並針對輸出網際網路流量使用 NAT 閘道。這樣可確保支援應用程式的伺服器無法公開存取，同時只提供公開存取權給需要它的伺服器。
- 使用不同的 VPC 來執行您的 PeopleSoft 生產環境和非生產環境。使用 AWS Transit Gateway、VPC 對等互連、網路 ACL 和安全群組來控制 VPC 和現場部署資料中心 (如有必要) 之間的流量。
- 遵循最小特權的原則。僅將應用 PeopleSoft 程式使用之 AWS 資源的存取權授予絕對需要的使用者。僅授與執行工作所需的最低權限。如需詳細資訊，請參閱 AWS Well-Architected Framework 的安全性支柱。
- 盡可能使用 AWS Systems Manager 存取應用 PeopleSoft 程式使用的 EC2 執行個體。

可靠性最佳做法

- 當您使用應用程式負載平衡器時，請為每個已啟用的可用區域註冊單一目標。這使得負載平衡器最有效。
- 我們建議您為每個 PeopleSoft 生產環境使用三個不同的 URL：一個用於存取應用程式的 URL，一個用於提供整合代理程式，另一個用於檢視報表的 URL。如果可能的話，每個 URL 都應該有自己的專用 Web 伺服器和應用程式伺服器。這種設計有助於使您的 PeopleSoft 應用程式更安全，因為每個 URL 都有不同的功能和受控的存取。如果基礎服務失敗，它也會將影響範圍降至最低。
- 建議您針對應用程式的負載平衡器目標群組設定健康狀態檢 PeopleSoft 查。運作狀態檢查應在 Web 伺服器上執行，而不是執行這些伺服器的 EC2 執行個體。如此可確保 Web 伺服器當機或託管 Web 伺服器的 EC2 執行個體發生故障時，Application Load Balancer 器會準確地反映該資訊。
- 對於生 PeopleSoft 產應用程式，我們建議您將 Web 伺服器分散至至少三個可用區域。這可確保即使其中一個可用區域故障， PeopleSoft 應用程式始終具有高可用性。
- 對於生 PeopleSoft 產應用程式，啟用震動共用 () `joltPooling=true`。如此可確保當伺服器出於修補目的或因為虛擬機器故障而關閉時，您的應用程式容錯移轉至另一部應用程式伺服器。
- 對於生 PeopleSoft 產應用程式，設`DynamicConfigReload` 定為 1。8.52 版及更新 PeopleTools 版本支援此設定。它將新的應用程序服務器動態地添加到 Web 服務器，而無需重新啟動服務器。
- 若要將套用 PeopleTools 修補程式時的停機時間降到最低，請針對 Web 和應用程式伺服器的 Auto Scaling 群組啟動設定使用藍色/綠色部署方法。如需詳細資訊，請參閱 AWS 上的部署選項概觀白皮書。
- 使用 AWS Backup 在 AWS 上備份您的 PeopleSoft 應用程式。AWS Backup 是符合成本效益、全受管、以政策為基礎的服務，能夠大規模簡化資料保護程序。

效能最佳做法

- 在 Application Load Balancer 終止 SSL，以獲得最佳的 PeopleSoft 環境效能，除非您的企業需要整個環境中的加密流量。
- 為 AWS 服務 (例如亞馬遜簡單通知服務 (Amazon SNS) 建立界面 VPC 端點，CloudWatch以便流量始終保持內部。這是符合成本效益的，有助於保護您的應用程式

成本最佳化最佳做法

- 標記 PeopleSoft 環境使用的所有資源，並啟用成本分配標籤。這些標籤可協助您檢視和管理資源成本。

- 對於 PeopleSoft 生產應用程式，請為 Web 伺服器和應用程式伺服器設定 Auto Scaling 群組。這樣可以維護最少數量的 Web 和應用程式伺服器，以支援您的應用程式。您可以使用 Auto Scaling 群組原則，視需要擴展和縮減伺服器。
- 當成本超過您指定的預算閾值時，請使用帳單警示來收到警示。

可持續發展最佳

- 使用基礎結構即程式碼 (IaC) 來維護您的 PeopleSoft 環境。這可協助您建置一致的環境並維持變更控制。

史诗

將您的 PeopleSoft 數據庫遷移到 Amazon RDS

任務	描述	所需技能
建立資料庫子網路群組。	在 Amazon RDS 主控台的導覽窗格中，選擇子網路群組，然後建立具有多個可用區域中子網路的 Amazon RDS 資料庫子網路群組。這是 Amazon RDS 資料庫在異地同步備份組態中執行的必要條件。	雲端管理員
創建 Amazon RDS 數據庫。	在您為高可用性管理 PeopleSoft 局環境選取的 AWS 區域的可用區域中建立 Amazon RDS 資料庫。建立 Amazon RDS 資料庫時，請務必選取異地同步備份選項 (建立待命執行個體) 和您在上一步中建立的資料庫子網路群組。如需詳細資訊，請參閱 Amazon RDS 文件。	雲端管理員、Oracle 資料庫管理員
將您的 PeopleSoft 資料庫遷移到 Amazon RDS。	使用 AWS PeopleSoft Database Migration Service	雲端管理員， PeopleSoft DBA

任務	描述	所需技能
	(AWS DMS)，將您現有的資料庫遷移到 Amazon RDS 資料庫。如需詳細資訊，請參閱 AWS DMS 文件和 AWS 部落格文章使用 AWS DMS 遷移 Oracle 資料庫，且停機時間幾乎為零。	

設定您的 Amazon EFS 檔案系統

任務	描述	所需技能
建立檔案系統。	在 Amazon EFS 主控台上，為每個可用區域建立檔案系統並掛接目標。如需指示，請參閱 Amazon EFS 文件。建立檔案系統後，請記下其 DNS 名稱。當您掛載檔案系統時，將會使用此資訊。	雲端管理員

設定您的 PeopleSoft 應用程式和檔案系統

任務	描述	所需技能
啟動 EC2 執行個體。	為您的應用程式啟動 EC2 執行個 PeopleSoft 體。如需指示，請參閱 Amazon EC2 文件。 • 針對名稱，輸入 APP_TEMPLATE 。 • 對於作業系統影像，請選擇「紅帽」。 • 針對執行個體類型，選擇適合您 PeopleSoft 應用程式的	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
	執行個體類型。如需詳細資訊，請參閱架構區段中的架構詳細資料。	
在執行個體 PeopleSoft 上安裝。	在您建立的 EC2 執行個體 PeopleTools 上安裝 PeopleSoft 應用程式。如需相關指示，請參閱 Oracle 說明文件。	雲端管理員、 PeopleSoft 管理員
建立應用程式伺服器。	為 AMI 範本建立應用程式伺服器，並確定該範本已成功連線至 Amazon RDS 資料庫。	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
掛載 Amazon EFS 檔案系統。	以根使用者身分登入 EC2 執行個體，然後執行下列命令，將 Amazon EFS 檔案系統掛載到伺服器PSFTMNT上名為的資料夾。 `sudo su -` `mkdir /psftmnt` `cat /etc/fstab` 將以下行附加到文/etc/fstab 件中。使用您在建立檔案系統時記下的 DNS 名稱。 `fs-09e064308f11453` `88.efs.us-east-1.a` `mazonaws.com:/ /` `psftmnt nfs4 nfsvers=4` `.1,rsize=1048576,w` `size=1048576,hard,` `timeo=600,retrans=` `2,noresvport,_netdev` ` 0 0` `mount -a`	雲端管理員、 PeopleSoft 管理員
檢查權限。	確保該PSFTMNT文件夾具有適當的權限，以便用 PeopleSoft 戶可以正確訪問它。	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
建立其他執行個體。	重複此史詩中的前面步驟，為處理程序排程器、網頁伺服器和 Elasticsearch 伺服器建立範本執行個體。命名這些例證PRCS_TEMPLATE WEB_TEMPLATE 、和SRCH_TEMPLATE 。對於 Web 伺服器，請設定joltPooling=true 和DynamicConfigReload=1 。	雲端管理員、 PeopleSoft 管理員

建立指令碼以設定伺服器

任務	描述	所需技能
建立指令碼以安裝應用程式伺服器。	在 Amazon APP_TEMPLATE EC2 執行個體中，以 PeopleSoft 使用者身分建立下列指令碼。命名它appstart.sh 並將其放置在目PS_HOME錄中。您將使用此指令碼啟動應用程式伺服器，並在 Amazon EFS 掛載上記錄伺服器名稱。 `#!/bin/ksh` `. /usr/homes/hcmdemo/.profile.` `psadmin -c configure -d HCMDEMO` `psadmin -c parallelboot -d HCMDEMO`	PeopleSoft 管理員

任務	描述	所需技能
	`touch /psftmnt/`echo $HOSTNAME`	
建立指令碼以安裝程序排程器伺服器。	在 Amazon PRCS_TEMPLATE EC2 執行個體中，以 PeopleSoft 使用者身分建立下列指令碼。命名它`prcsstart.sh` 並將其放置在目`PS_HOME`錄中。您將使用此指令碼來啟動處理序排程器伺服器。 `#!/bin/ksh` `. /usr/homes/hcmdemo/.profile` `/* The following line ensures that the process scheduler always has a unique name during replacement or scaling activity. */` `sed -i "s/.*PrcsServerName.*/`hostname -I | awk -F. '{print "PrcsServerName=PSUNX"$3$4}'`/" $HOME/appserv/prcs/*/psprcs.cfg` `psadmin -p configure -d HCMDEMO` `psadmin -p start -d HCMDEMO`	PeopleSoft 管理員

任務	描述	所需技能
建立指令碼以安裝彈性搜尋伺服器。	在亞馬遜 SRCH_TEMPLATE EC2 執行個體中，以彈性搜尋使用者身分建立下列指令碼。命名它srchstart.sh 並將其放置在目HOME錄中。 `#!/bin/ksh` `/* The following line` ` ensures that the` ` correct IP is indicated` ` in the elasticse` `arch.yaml file. */` `sed -i "s/.*netw` `ork.host.*/`hostna` `me -I	awk '{print` ` "host:"$0}'`/"` ` $ES_HOME_DIR/config/` `elasticsearch.yaml` `nohup $ES_HOME_DIR/bin/` `elasticsearch &`	PeopleSoft 管理員

任務	描述	所需技能
建立指令碼以安裝 Web 伺服器。	在 Amazon WEB_TEMPLATE EC2 執行個體中，以 Web 伺服器使用者身分在HOME目錄中建立下列指令碼。 renip.sh：此指令碼可確保 Web 伺服器在從 AMI 複製時具有正確的 IP。 `#!/bin/ksh` `hn=`hostname`` `/* On the following` `line, change the IP` `with the hostname with` `the hostname of the web` `template. */` `for text_file in `find` `* -type f -exec grep` `-l '<hostname-of-the-` `web-template>' {} \;` `do` `sed -e 's/<hostn` `ame-of-the-web-tem` `plate>/'$hn'/g'` `$text_file > temp` `mv -f temp $text_file` `done`  psstrsetup.sh ：此指令碼可確保 Web 伺服器使用目前正在執行的正確應用程式伺服器 IP。它嘗試連接到 jolt 端口上的每個應用程序服務器，並將其添加到配置文件中。  `#!/bin/ksh` `c2=""`	PeopleSoft 管理員

任務	描述	所需技能
	for ctr in `ls -1 /psftmnt/*.internal` do c1=`echo $ctr | awk -F "/" '{print $3}'` /* In the following lines, 9000 is the jolt port. Change it if necessary. */ if nc -z $c1 9000 2> /dev/null; then if [[$c2 = ""]]; then c2="psserver="`echo $c1`":9000" else c2=`echo $c2`","`echo $c1`":9000" fi fi done webstart.sh ：此腳本運行前兩個腳本並啟動 Web 服務器。 #!/bin/ksh /* Change the path in the following if necessary. */ cd /usr/homes/hcmdemo ./renip.sh ./psstrsetup.sh webserv/peoplesoft/bin/startPIA.sh	

任務	描述	所需技能
新增一個定位表項目。	在 Amazon EC2 WEB_TEMPLATE 實例中，作為網絡服務器用戶，將以下行添加到 crontab。變更時間和路徑以反映您需要的值。此項目可確保您的 Web 伺服器在configuration.properties 檔案中始終具有正確的應用程式伺服器項目。 `* * * * * /usr/homes/hcmdemo/psstrsetup.sh`	PeopleSoft 管理員

建立 AMI 和 Auto Scaling 群組範本

任務	描述	所需技能
為應用程式伺服器範本建立 AMI。	在 Amazon EC2 主控台上，建立亞 Amazon APP_TEMPLATE EC2 執行個體的 AMI 映像。命名 AMI PSAPPSRV-SCG-VER1 。如需指示，請參閱 Amazon EC2 文件。	雲端管理員、 PeopleSoft 管理員
為其他伺服器建立 AMI。	重複上一個步驟，為處理程序排程器、Elasticsearch 伺服器和網頁伺服器建立 AMI。	雲端管理員、 PeopleSoft 管理員
為應用程式伺服器「自動調整」群組建立啟動範本。	為應用程式伺服器「自動調整」群組建立啟動範本。命名範本PSAPPSRV_TEMPLATE。 在範本中，選擇您為APP_TEMPLATE 執行個	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
	體建立的 AMI。如需指示，請參閱 Amazon EC2 文件。 • 在啟動範本中，根據您的需求選取執行個體類型。 • 在 [進階詳細資料] 區段的 [使用者資料] 欄位中，新增下列項目。請確定路徑和使用者資訊正確無誤。您在上一個步驟中建立了`appstart.sh` 指令碼。 `#! /bin/ksh` `su -c "/usr/homes/hcmdemo/appstart.sh" - hcmdemo`	
為程序排程器伺服器 Auto Scaling 群組建立啟動範本。	重複上一個步驟，為程序排程器伺服器 Auto Scaling 群組建立啟動範本。為範本命名`PSPRCS_TEMPLATE` 。在範本中，選擇您為程序排程器建立的 AMI。 • 在 [進階詳細資料] 區段的 [使用者資料] 欄位中，新增下列項目。請確定路徑和使用者資訊正確無誤。您在上一個步驟中建立了`prcsstart.sh` 指令碼。 `#! /bin/ksh` `su -c "/usr/homes/hcmdemo/prcsstart.sh" - hcmdemo`	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
為 Elasticsearch 伺服器 Auto Scaling 群組建立啟動範本。	重複上述步驟，為 Elasticsearch 伺服器「自動調整比例」群組建立啟動範本。為範本命名SRCH_TEMPLATE 。在範本中，選擇您為搜尋伺服器建立的 AMI。 • 在 [進階詳細資料] 區段的 [使用者資料] 欄位中，新增下列項目。請確定路徑和使用者資訊正確無誤。您在上一個步驟中建立了srchstart.sh 指令碼。 `#! /bin/ksh` `su -c “/usr/homes/essearch/srchstart.sh” - essearch`	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
為 Web 伺服器 Auto Scaling 群組建立啟動範本。	重複上述步驟，為 Web 伺服器 Auto Scaling 群組建立啟動範本。為範本命名`WEB_TEMPLATE` 。在範本中，選擇您為 Web 伺服器建立的 AMI。 • 在 [進階詳細資料] 區段的 [使用者資料] 欄位中，新增下列項目。請確定路徑和使用者資訊正確無誤。您在上一個步驟中建立了`webstart.sh` 指令碼。 `#! /bin/ksh` `su -c "/usr/homes/hcmdemo/webstart.sh"` `- hcmdemo`	雲端管理員、 PeopleSoft 管理員

建立 Auto Scaling 群組

任務	描述	所需技能
為應用程式伺服器建立「Auto Scaling」群組。	在 Amazon EC2 主控台上，使用`PSAPPSRV_TEMPLATE` 範本建立`PSAPPSRV_ASG` 為應用程式伺服器呼叫的 Auto Scaling 群組。如需指示，請參閱 Amazon EC2 文件。 • 在 [選擇執行個體啟動選項] 頁面上，選取正確的 VPC，然後從不同可用區域選取多個子網路。 • 在 [設定進階選項] 頁面上，請勿選取負載平衡器。	雲端管理員、 PeopleSoft 管理員

任務	描述	所需技能
	• 在 [設定群組大小和擴展原則] 頁面上，根據您要架構系統的負載量，以及是否要使用擴展原則來選擇設定。我們建議您至少將所需容量和最小容量設定為 2，以便至少有一個執行個體可用於在任何時間點為流量提供服務。如需有關 Auto Scaling 政策的詳細資訊，請參閱 Amazon EC2 文件。	
為其他伺服器建立「Auto Scaling」群組。	重複上一個步驟，為程序排程器、Elasticsearch 伺服器和網頁伺服器建立「Auto Scaling」群組。	雲端管理員、 PeopleSoft 管理員

建立和設定目標群組

任務	描述	所需技能
建立 Web 伺服器的目標群組。	在 Amazon EC2 主控台上，為網頁伺服器建立目標群組。如需指示，請參閱 E lastic Load Balancing 文件。將連接埠設定為 Web 伺服器監聽的連接埠。	雲端管理員
設定健康狀態檢查。	確認健康狀態檢查具有正確的值，以反映您的業務需求。如需詳細資訊，請參閱 Elastic Load Balancing 說明文件。	雲端管理員
建立彈性搜尋伺服器的目標群組。	重複上述步驟，以建立呼叫 Elasticsearch 伺服	雲端管理員

任務	描述	所需技能
	器PSFTSRCH的目標群組，並設定正確的彈性搜尋連接埠。	
將目標群組新增至「Auto Scaling」群組。	開啟您先前建立的 Web 伺服器「PSPIA_ASG Auto Scaling」群組。在 [負載平衡] 索引標籤上，選擇 [編輯]，然後將目PSFTWEB標群組新增至 [Auto Scaling] 群組。 對 Elasticsearch Auto Scaling 群組重複此步驟，PSSRCH_ASG 以新增PSFTSRCH您先前建立的目標群組。	雲端管理員
設定工作階段黏性。	在目標群組中PSFTWEB，選擇「屬性」標籤，選擇「編輯」，然後設定工作階段黏著性。對於粘性類型，請選擇負載平衡器生成的 cookie，並將持續時間設置為 1。如需詳細資訊，請參閱 Elastic Load Balancing 說明文件。 針對目標群組重複此步驟PSFTSRCH。	雲端管理員

建立和設定應用程式負載平衡器

任務	描述	所需技能
為 Web 伺服器建立負載平衡器。	建立名為的 Application Load Balancer 器，PSFTLB以平衡 Web 伺服器的流量。如需	雲端管理員

任務	描述	所需技能
	指示，請參閱 E lastic Load Balancing 文件。 • 提供負載平衡器名稱。 • 對於 Scheme (結構描述)，選擇 Internet-facing (面向網際網路)。 • 在 [網路對應] 區段中，選取正確的 VPC 和來自不同可用區域的至少兩個公用子網路。 • 在「監聽器和路由」段落中，選取目標群組，PSFTWEB然後指定正確的協定和連接埠號碼。	
為彈性搜尋伺服器建立負載平衡器。	建立名為 Application Load Balancer 器，PSFTSCH以平衡 Elasticsearch 伺服器的流量。 • 提供負載平衡器名稱。 • 針對「配置」，選擇「內部」 • 在 [網路對應] 區段中，選取正確的 VPC 和私人子網路。 • 在「監聽器和路由」段落中，選取目標群組，PSFTSRCH然後指定正確的協定和連接埠號碼。	雲端管理員

任務	描述	所需技能
設定 Route 53。	在 Amazon Route 53 主控台上，在託管區域中建立記錄，以便為 PeopleSoft 應用程式提供服務。有關說明，請參閱 Amazon Route 53 文檔。這可確保所有流量都通過PSFTLB負載平衡器。	雲端管理員

相關資源

- 甲骨文 PeopleSoft 網站
- AWS 文件

更多模式

- 使用 AWS 副駕駛員將叢集應用程式部署到 Amazon ECS
- 使用地形部署 CloudWatch Synthetics 金絲雀
- 使用 Amazon 基岩和 Amazon Transcribe 來記錄語音輸入的機構知識

雲端原生

主題

- 使用亞馬遜 Kinesis 影片串流和 AWS Fargate 建立影片處理管道
- 使用 AWS 服務監控 SAP RHEL 起搏器叢集
- 成功將 S3 儲存貯體匯入為 AWS CloudFormation 堆疊
- 更多模式

使用亞馬遜 Kinesis 影片串流和 AWS Fargate 建立影片處理管道

由皮奧特·喬特科夫斯基 (AWS) 和普希帕勞·唐加維爾 (AWS) 創建

環境：PoC 或試點	技術：雲端原生；軟體開發與測試；媒體服務	AWS 服務：AWS Fargate；Amazon Kinesis；Amazon S3

Summary

此模式示範如何使用 Amazon Kinesis Video Streams 和 AWS Fargate 從影片串流擷取畫面，並將其存放為影像檔案，以便在 Amazon 簡單儲存服務 (Amazon S3) 中進行進一步處理。

該模式提供了一個 Java Maven 項目的形式的示例應用程序。此應用程式使用 AWS Cloud Development Kit (AWS CDK) 來定義 AWS 基礎設施。無論是幀處理邏輯和基礎結構定義都寫在 Java 編程語言。您可以使用此範例應用程式作為開發自己的即時視訊處理管道的基礎，或建立機器學習管道的視訊預處理步驟。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 已安裝開發套件 11
- 阿帕奇 Maven 的，安裝
- 已安裝 AWS Cloud Development Kit (AWS CDK)
- 已安裝 AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版
- Docker (建置要在 AWS Fargate 任務定義中使用的 Docker 映像時所需)，已安裝

限制

這種模式旨在作為概念證明，或作為進一步發展的基礎。在生產部署中，不應以目前的形式使用它。

產品版本

- 此病毒碼已透過 AWS CDK 版本 1.77.0 進行測試 (請參閱 AWS CD K 版本)
- JDK 11

- AWS CLI 第 2 版

架構

目標技術堆疊

- Amazon Kinesis Video Streams
- AWS Fargate 任務
- Amazon Simple Queue Service (Amazon SQS) 佇列
- Amazon S3 儲存貯體

目標架構

使用者建立 Kinesis 視訊串流、上傳影片，並將包含輸入 Kinesis 視訊串流和輸出 S3 儲存貯體的詳細資訊的 JSON 訊息傳送至 SQS 佇列。AWS Fargate 是在容器中執行主要應用程式，會從 SQS 佇列中提取訊息並開始擷取框架。每個框架都儲存在映像檔中，並存放在目標 S3 儲存貯體中。

自動化和規模

範例應用程式可在單一 AWS 區域內水平和垂直擴展。透過增加從 SQS 佇列讀取的已部署 AWS Fargate 任務數量，即可達成水平擴展。透過增加應用程式中的框架分割和影像發佈執行緒數目，可以實現垂直縮放。這些設定會以環境變數的形式傳遞至 AWS CDK QueueProcessingFargateService資源定義中的應用程式。由於 AWS CDK 堆疊部署的性質，您可以在多個 AWS 區域和帳戶中部署此應用程式，而無需額外費力。

工具

工具

- AWS CDK 是一種軟體開發架構，可透過使用程式設計語言 (例如 TypeScript、Python、 JavaScript Java 和 C#/.net) 來定義雲端基礎設施和資源。
- Amazon Kinesis Video Streams 是一種全受管 AWS 服務，可用來將即時影片從裝置串流到 AWS 雲端，或建立用於即時影片處理或批次導向影片分析的應用程式。
- AWS Fargate 是適用於容器的無伺服器運算引擎。Fargate 無需佈建和管理伺服器，並讓您專注於開發應用程式。

- Amazon S3 是一種物件儲存服務，提供可擴展性、資料可用性、安全性和效能。
- Amazon SQS 是全受管訊息佇列服務，可讓您分離和擴展微型服務、分散式系統和無伺服器應用程式。

Code

- 隨即附加範例應用程式專案 (`frame-splitter-code.zip`) 的 .zip 檔案。

史诗

部署基礎架構

任務	描述	所需技能
啟動 Docker 常駐程式。	在您的本機系統上啟動 Docker 常駐程式。AWS CDK 使用泊塢視窗建立 AWS Fargate 任務中使用的映像。您必須先執行 Docker，才能繼續進行下一個步驟。	開發者、 DevOps 工程師
建置專案。	下載`frame-splitter-code` 範例應用程式 (附件)，並將其內容解壓縮至本機電腦上的資料夾中。在您可以部署基礎設施之前，您必須構建 Java Maven 項目。在命令提示字元中，瀏覽至專案的根目錄，然後執行以下命令來建置專案： `mvn clean install`	開發者、 DevOps 工程師
啟動 AWS CDK 的啟動程序。	(僅限首次使用 AWS CDK 使用者) 如果這是您第一次使用 AWS CDK，則可能必須執行 AWS CLI 命令來啟動環境：	開發者、 DevOps 工程師

任務	描述	所需技能
	`cdk bootstrap --profile "$AWS_PROFILE_NAME"` 其中$AWS_PROFILE_NAME 包含 AWS 登入資料中的 AWS 設定檔名稱。或者，您可以移除此參數以使用預設設定檔。如需詳細資訊，請參閱 AWS CDK 文件。	

任務	描述	所需技能
部署 AWS CDK 堆疊。	在此步驟中，您可以在 AWS 帳戶中建立所需的基礎設施資源 (SQS 佇列、S3 儲存貯體、AWS 遠門任務定義)、建立 AWS Fargate 任務所需的 Docker 映像，以及部署應用程式。在命令提示字元中，瀏覽至專案的根目錄，然後執行以下命令： `cdk deploy --profile "$AWS_PROFILE_NAME" --all` 其中$AWS_PROFILE_NAME 包含 AWS 登入資料中的 AWS 設定檔名稱。或者，您可以移除此參數以使用預設設定檔。確認部署。請記下 CDK 部署輸出中的QueueUrl和值區值；稍後的步驟將需要這些值。AWS CDK 會建立資產、將資產上傳到您的 AWS 帳戶，然後建立所有基礎設施資源。您可以在 AWS CloudFormation 主控台觀察資源建立程序。如需詳細資訊，請參閱 AWS CloudFormation 文件和 AWS CDK 文件。	開發者、 DevOps 工程師

任務	描述	所需技能
建立視訊串流。	在此步驟中，您會建立 Kinesis 視訊串流，做為視訊處理的輸入串流。請確定您已安裝並設定 AWS CLI。在 AWS CLI 中，執行： `aws kinesisvideo --profile "$AWS_PROFILE_NAME" create-stream --stream-name "$STREAM_NAME" --data-retention-in-hours "24"` 其中`$AWS_PROFILE_NAME` 包含 AWS 登入資料中的 AWS 設定檔名稱 (或移除此參數以使用預設設定檔)，並且`$STREAM_NAME` 是任何有效的串流名稱。 或者，您也可以按照 Kinesis 影片串流說明文件中的步驟，使用 Kinesis 主控台建立視訊串流。請記下建立串流的 AWS 資源名稱 (ARN)；稍後您將需要它。	開發者、 DevOps 工程師

執行範例

任務	描述	所需技能
將影片上傳至串流。	在範例`frame-splitter-code` 應用程式的專案資料夾中，開啟資`src/test/`	開發者、 DevOps 工程師

任務	描述	所需技能
	java/amazon/awscdk/examples/splitter 料夾中的ProcessingTaskTest.java 檔案。將profileName 和streamName 變數取代為您在先前步驟中使用的值。若要將範例影片上傳到您在上一個步驟中建立的 Kinesis 視訊串流，請執行： `amazon.awscdk.examples.splitter.ProcessingTaskTest#testExample test` 或者，您也可以使用 Kinesis 影片串流說明文件中所述的其中一種方法來上傳影片。	

任務	描述	所需技能
啟動視頻處理。	現在您已將視訊上傳到 Kinesis 視訊串流，就可以開始處理它了。若要啟動處理邏輯，您必須將包含詳細資訊的訊息傳送至 AWS CDK 在部署期間建立的 SQS 佇列。若要使用 AWS CLI 傳送訊息，請執行： `aws sqs --profile "$AWS_PROFILE_NAME" send-message --queue-url QUEUE_URL --message-body MESSAGE` 其中`$AWS_PROFILE_NAME` 包含 AWS 登入資料中的 AWS 設定檔名稱 (移除此參數以使用預設設定檔)，`QUEUE_URL` 是 AWS CDK 輸出的`QueueUrl`值，並且`MESSAGE`是以下格式的 JSON 字串： `{ "streamARN": "STREAM_ARN", "bucket": "BUCKET_NAME", "s3Directory": "test-output" }` 其中`STREAM_ARN` 是您在先前步驟中建立的影片串流的 ARN，`BUCKET_NAME` 是 AWS CDK 輸出的儲存貯體值。	開發者、 DevOps 工程師

任務	描述	所需技能
	傳送此訊息會啟動視訊處理。或者，您也可以使用 Amazon SQS 主控台傳送訊息，如 Amazon SQS 文件所述。	
查看視頻幀的圖像。	您可以在 S3 輸出儲存貯體中看到產生的映像，`s3://BUCKET_NAME/test-output` 其中`BUCKET_NAME` 是 AWS CDK 輸出的儲存貯體值。	開發者、 DevOps 工程師

相關資源

- AWS CDK 文件
- AWS CDK API 參考
- AWS CDK 介紹性研討會
- Amazon Kinesis Video Streams 文件
- 範例：使用識別視訊串流中的物件 SageMaker
- 範例：剖析和呈現 Kinesis Video Streams 片段
- 使用 Amazon Kinesis 影片串流和亞馬遜 SageMaker (AWS Machine Learning 部落格文章)，即時大規模分析即時影片
- AWS Fargate 開始使用

其他資訊

選擇一個 IDE

我們建議您使用自己喜歡的 Java IDE 來構建和探索此項目。

清除

執行完此範例後，請移除所有已部署的資源，以避免產生額外的 AWS 基礎設施成本。

若要移除基礎設施和影片串流，請在 AWS CLI 中使用以下兩個命令：

```
cdk destroy --profile "$AWS_PROFILE_NAME" --all
```

```
aws kinesisvideo --profile "$AWS_PROFILE_NAME" delete-stream --stream-arn "$STREAM_ARN"
```

或者，您也可以使用 AWS CloudFormation 主控台移除 AWS CloudFormation 堆疊，以及使用 Kinesis 主控台移除 Kinesis 影片串流，以手動移除資源。請注意，`cdk destroy`不會移除輸出 S3 儲存貯體或亞馬遜彈性容器登錄 (Amazon ECR) 儲存庫 (`aws-cdk/assets`) 中的映像。您必須手動刪除它們。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS 服務監控 SAP RHEL 起搏器叢集

由刺耳索里亞（AWS），蘭迪·德拉曼（AWS）和雷文德拉沃爾（AWS）創建

環境：生產	技術：雲端原生；基礎架構；作業系統	工作負載：SAP
AWS 服務：Amazon CloudWatch；Amazon SNS；Amazon CloudWatch 日誌		

Summary

此模式概述了使用 Amazon 和亞馬遜簡單通知服務 (Amazon SNS)，為 SAP 應用程式和 SAP HANA 資料庫服務的 RHEL (RHEL) 起搏器叢集監控 CloudWatch 和設定警示的步驟。

此組態可讓您在 SAP SCS 或 ASCS、排入佇列複寫伺服器 (ERS) 以及 SAP HANA 叢集資源時，藉由 CloudWatch 記錄串流、指標篩選器和警示的協助來監視這些資源處於「已停止」狀態。Amazon SNS 會向基礎設施或 SAP 基礎設施團隊傳送有關已停止叢集狀態的電子郵件。

您可以使用AWS CloudFormation指令碼或AWS服務主控台來建立此模式的AWS資源。此模式假設您正在使用主控台；它不提供 CloudWatch 和 Amazon SNS 的 CloudFormation 指令碼或涵蓋基礎設施部署。起搏器指令用於設定叢集警示配置。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶。
- Amazon SNS 已設定為傳送電子郵件或行動通知。
- 適用於爪哇的 SAP ASCS/ERS 或 SCS/ERS，以及 SAP HANA 資料庫 RHEL 起搏器叢集。如需詳細說明，請參閱下列主題：
  - 集群設置
  - SAP 網路weaver 網路通訊/Java 叢集設定

限制

- 此解決方案目前適用於 RHEL 7.3 版及更新版本的起搏器叢集。尚未在 SUSE 作業系統上進行測試。

產品版本

- RHEL 7.3 及更高版本

架構

目標技術堆疊

- RHEL 起搏器警示事件驅動代理程式
- Amazon Elastic Compute Cloud (Amazon EC2)
- CloudWatch 警報
- CloudWatch 日誌群組和量度篩選
- Amazon SNS

目標架構

下圖說明此解決方案的元件和工作流程。

自動化和規模

- 您可以使用 CloudFormation 指令碼自動建立AWS資源。您也可以使用其他度量篩選器來縮放和涵蓋多個叢集。

工具

AWS 服務

- Amazon 可 CloudWatch協助您即時監控AWS資源和執行應用程式的指標。AWS
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。

工具

- CloudWatch 代理程式 (統一) 是一種工具，可從 EC2 執行個體收集系統層級指標、日誌和追蹤，並從應用程式擷取自訂指標。
- Pacemaker 警示代理程式 (適用於 RHEL 7.3 及更新版本) 是一種工具，可在 Pacemaker 叢集中發生變更時 (例如資源停止或重新啟動時) 啟動動作。

最佳實務

- 如需在上使用 SAP 工作負載的最佳做法AWS，請參閱 SAP Lens 以取AWS得 Well-Architected 的架構。
- 考慮為 SAP HANA 叢集設定 CloudWatch 監控所需的成本。如需詳細資訊，請參閱CloudWatch 文件。
- 請考慮針對 Amazon SNS 警示使用呼叫器或票務機制。
- 請務必檢查 RPM 套件的 RPM 套件是否適用於個人電腦、心臟起搏器和AWS隔離代理程式的 RHEL 高可用性 (HA) 版本。

史诗

設定 Amazon SNS

任務	描述	所需技能
建立 SNS 主題。	1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。 2. 在 Amazon SNS 儀表板上，在 Common actions (常見的動作) 下，選擇 Create Topic (建立主題)。 3. 在「建立新主題」對話方塊中，選擇「標準」做為「類型」。	AWS 管理員

任務	描述	所需技能
	4. 在「主題名稱」中，輸入主題的名稱 (例如，`my-topic`)。 5. 請選擇 建立主題。 這會建立具有可讓您發佈通知的資源原則的 SNS 主題。 6. 複製主題 ARN (例如，`arn:aws:sns:us-east-1:111122223333:my-topic`)。您將在稍後的步驟中使用此 ARN。	

任務	描述	所需技能
修改 SNS 主題的存取原則。	1. 在 Amazon SNS 主控台的導覽窗格中，選擇「主題」，然後選擇您建立的主題。 2. 選擇 [編輯]，然後移至 [存取原則] 區段。 3. 請確定存取原則包含 CloudWatch 為允許發行至此主題的其中一個服務主體。例如： `{ "Sid": "Allow AWS CloudWatch to Publish to this SNS topic", "Effect": "Allow", "Principal": { "Service": [ "cloudwatch.amazonaws.com" ] }, "Action": "SNS:Publish", "Resource": "arn:aws:sns:us-east-1:111122223333:my-topic" }` 4. 選擇儲存變更。	AWS 系統管理員

任務	描述	所需技能
訂閱 SNS 主題。	1. 在 Amazon SNS 主控台的導覽窗格中，選擇訂閱，建立訂閱。 2. 針對主題 ARN，貼上您在第一個工作中建立的 ARN。 3. 對於通訊協定，選擇電子郵件。 4. 針對端點，輸入負責 SAP Pacemaker 叢集且應該接收通知的人員或團隊的電子郵件地址。例如，這可以是 SAP Basis 或基礎結構小組的通訊群組清單的電子郵件地址。 5. 選擇建立訂閱。 6. 從電子郵件應用程式中，開啟 AWS 通知傳來的訊息，然後確認您的訂閱。 您的 Web 瀏覽器顯示自 Amazon SNS 的確認回覆。	AWS 系統管理員

確認叢集的設定

任務	描述	所需技能
檢查叢集狀態。	使用 pcs 狀態指令來確認資源是否處於線上狀態。	SAP 基礎管理員

設定心臟起搏器警示

任務	描述	所需技能
在主要叢集執行處理上設定 Pacemaker 警示代理程式。	登入 pimary 叢集中的 EC2 執行個體並執行下列命令： `install --mode=0755 /usr/share/pacemaker/alerts/alert_file.sh.sample` `touch /var/lib/pacemaker/alert_file.sh` `touch /var/log/pcmk_alert_file.log` `chown hacluster:haclient /var/log/pcmk_alert_file.log` `chmod 600 /var/log/pcmk_alert_file.log` `pcs alert create id=alert_file description="Log events to a file." path=/var/lib/pacemaker/alert_file.sh` `pcs alert recipient add alert_file id=my-alert_logfile value=/var/log/pcmk_alert_file.log`	SAP 基礎管理員
在次要叢集執行個體上設定 Pacemaker 警示代理程式。	登入次要叢集中的次要叢集 EC2 執行個體，並執行下列命令： `install --mode=0755 /usr/share/pacemaker/alerts/alert_file.sh.sample`	SAP 基礎管理員

任務	描述	所需技能
	`touch /var/lib/pacemaker/alert_file.sh` `touch /var/log/pcmk_alert_file.log` `chown hacluster:haclient /var/log/pcmk_alert_file.log` `chmod 600 /var/log/pcmk_alert_file.log`	
確認已建立 RHEL 警示資源。	使用下列命令確認已建立警示資源： `pcs alert` 命令的輸出將如下所示： `[root@xxxxxxx ~]# pcs alert` `Alerts:` ` Alert: alert_file (path=/var/lib/pacemaker/alert_file.sh)` `  Description: Log events to a file.` `  Recipients:` `   Recipient: my-alert_logfile (value=/var/log/pcmk_alert_file.log)`	SAP 基礎管理員

設定 CloudWatch 代理程式

任務	描述	所需技能
安裝代 CloudWatch 理程式。	有幾種方法可以在 EC2 執行個體上安裝 CloudWatch 代理程式。若要使用指令行： 1. 下載 CloudWatch 代理程式套件： `wget https://s3.<region>.amazonaws.com/amazoncloudwatch-agent-region/redhat/amd64/latest/amazon-cloudwatch-agent.rpm` AWS 區域其中`<region>`是 EC2 執行個體所在的位置 (例如，`us-west-2`)。 2. 選擇性) 驗證封裝簽章。如需指示，請參閱文件中的驗證 CloudWatch 代理程式套 CloudWatch 件的簽章。 3. 在第一個執行個體上安裝套件： `sudo rpm -U ./amazon-cloudwatch-agent.rpm` 4. 對次要例證重複此步驟。 如需詳細資訊，請參閱CloudWatch 文件。	AWS 系統管理員

任務	描述	所需技能
將 IAM 角色附加到 EC2 執行個體。	若要讓 CloudWatch 代理程式從執行個體傳送資料，您必須將 IAM CloudWatchAgentServerRole角色附加至每個執行個體。或者，您可以將 CloudWatch 代理程式的政策新增至現有的 IAM 角色。如需詳細資訊，請參閱CloudWatch 文件。	AWS 管理員
設定 CloudWatch 代理程式以監視主要叢集執行處理上的 Pacemaker 警示代理程式記錄檔。	1. 執行以下命令來設定主要叢集執行個體： `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard` 2. 針對 Linux 選擇 1，然後選取監視策略的選項。 3. 對於 「是否要監視任何記錄檔」的問題，請選擇「是」，並從 pcs 警示指令提供 Pacemaker 記錄檔的路徑。在我們的例子中，它是`var/log/pcmk_alert_file.log` 。 4. 提供記錄群組和記錄資料流的名稱。如果您未指定記錄串流，則會使用AWS執行個體 ID 做為預設值。 5. 針對次要叢集執行個體重複步驟 1-4。	AWS 管理員

任務	描述	所需技能
在主要和次要叢集執行個體上啟動 CloudWatch 代理程式。	若要啟動代理程式，請在主要和次要叢集中的 EC2 執行個體上執行下列命令： `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json`	AWS 管理員

設定 CloudWatch 資源

任務	描述	所需技能
設定 CloudWatch 記錄群組。	1. 請在以下位置開啟 CloudWatch 主控台 https://console.aws.amazon.com/cloudwatch/ 2. 在功能窗格中，選擇 [記錄群組] > [建立記錄群組]。 3. 輸入記錄群組的名稱，然後選擇 [建立記錄群組]。 CloudWatch 代理程式會將 Pacemaker 警示檔案作為 CloudWatch 記錄串流傳輸到日誌群組。	AWS 管理員
設定 CloudWatch 量度篩選器。	量度篩選器可協助您搜尋 CloudWatch 記錄資料流stop	AWS 管理員、SAP 基礎管理員

任務	描述	所需技能
	`<cluster-resource-name>` 中的樣式。識別出此病毒碼時，度量篩選器會更新自訂量度。 1. 在 CloudWatch 主控台的導覽窗格中，選擇 [記錄群組]。 2. 選擇您在先前工作中建立的記錄群組名稱。 3. 選擇 Actions (動作) > Create metric filter (建立指標篩選條件)。 4. 針對篩選器模式，輸入要使用的篩選器模式，例如`stop ABC_scs`，以符合名`ABC_scs`為的 SAP SCS 叢集資源的停止事件。 如需詳細資訊，請參閱 CloudWatch 文件中的篩選器模式語法。 5. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件，用以測試模式。每個記錄事件都必須在個別的行上指定，因為分行符號是用來分隔記錄事件在 [記錄事件訊息] 方塊中。 6. 選擇 Next (下一步)，然後輸入篩選條件的名稱。 7. 在「測量結果詳細資訊」下，針對測量結果	

任務	描述	所需技能
	CloudWatch 命名空間，輸入要在其中發行測量結果的命名空間名稱 (例如`sapcluster_monitoring`)。如果此命名空間尚未存在，請選取 [建立新的]。 8. 在「測量結果名稱」中，輸入新測量結果的名稱 (例如`sapcluster_<sid>` ，其中`<sid>`是 SAP 系統識別名稱)。 9. 在「測量結果」值中，輸入 1。 或者，您也可以輸入權杖，例如`$size`。如此會針對包含 `size` 欄位的每個日誌事件，以 `size` 欄位中的數值遞增指標。 10對於「預設值」，輸入 0。 11選擇 Create metric filter (建立指標篩選條件)。 當度量篩選器識別步驟 4 中的模式時，它會將 CloudWatch 自訂量度的值更新`sapcluster_abc` 為 1。 CloudWatch 警示會`SAP-Cluster-QA1-ABC` 監控量度，`sapcluster_abc` 並在指標值變更為 1 時傳送 SNS	

任務	描述	所需技能
	通知。這表示叢集資源已停止，需要採取動作。	

任務	描述	所需技能
為 SAP ASCS/SCS 和 ERS CloudWatch 量度設定量度警示。	若要根據單一量度建立警示： 1. 在 CloudWatch 主控台的功能窗格中，選擇 [警報]、[所有鬧鐘]。 2. 選擇 Create alarm (建立警示)。 3. 選擇 Select Metric (選取指標)。 4. 搜尋在先前工作中建立的自訂量度`sapcluster_monitoring` 。 5. 選擇 SAP SCS 的測量結果名稱 (例如，`sapcluster_<abc>`)，這個名稱也是在上一個作業中建立的。 6. 在「圖形化量度」標籤上，設定下列項目： • 對於 Statistic (統計數字)，選擇 Maximum (最大值)。 • 對於期間，選擇 1 分鐘。 • 對於臨界值類型，選擇靜態並**`sapcluster_<sid>`** 將臨界值設定為大於或等於 1 的值。 7. 選擇下一步。 8. 在「通知」中，選取您在第一個史詩中建立的 SNS 主題。 9. 在 [名稱] 和 [說明] 中，提供警示名稱和簡短描述，然後選擇 [下一步]。	AWS 管理員

任務	描述	所需技能
	10選擇 Create Alarm (建立警示)。	
為 SAP HANA CloudWatch 量度設定量度警示。	透過以下變更，重複上一個工作設定 CloudWatch 量度警示的步驟： • 對於步驟 5，請選擇 SAP HANA 的測量結果名稱 (例如，`sapcluster_db_<abc>`)。 • 對於步驟 6，`sapcluster_<sid>` 將的臨界值設定為大於 0 的值。	AWS 管理員

相關資源

- 叢集事件的觸發指令碼 (RHEL 文件)
- 使用精靈建立 CloudWatch 代理程式組態檔 (CloudWatch 說明文件)
- 在伺服器上安裝和執行 CloudWatch 代理程式 (CloudWatch 說明文件)
- 根據靜態閾值建立 CloudWatch 警示 (CloudWatch 文件)
- 使用高可用性叢集在 AWS 上手動部署 SAP HANA (AWS網站上的 SAP 文件)
- SAP NetWeaver 指南 (AWS網站上的 SAP 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

成功將 S3 儲存貯體匯入為 AWS CloudFormation 堆疊

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：雲端原生；儲存與備份	AWS 服務：Amazon S3; AWS CloudFormation

Summary

如果您使用 Amazon Web Services (AWS) 資源，例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體，並且想要使用基礎設施即程式碼 (IaC) 方法，則可以將資源匯入 AWS CloudFormation 並以堆疊形式管理。

此模式提供成功將 S3 儲存貯體匯入為 AWS CloudFormation 堆疊的步驟。透過使用此模式的方法，您可以避免在單一動作匯入 S3 儲存貯體時可能發生的錯誤。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的 S3 儲存貯體和 S3 儲存貯體政策。如需詳細資訊，請參閱 AWS 知識中心中的 AWS Config 規則 s3-，我應該使用哪些 S3 儲存貯體政策來符合 AWS Config 規則 s3-bucket-ssl-requests-only。
- 現有的 AWS Key Management Service (AWS KMS) 金鑰及其別名。如需這方面的詳細資訊，請參閱 AWS KMS 文件中的使用別名。
- 範例 `CloudFormation-template-S3-bucket` AWS CloudFormation 範本 (隨附)，下載到您的本機電腦。

架構

該圖顯示以下工作流程：

1. 使用者會建立 JSON 或 YAML 格式的 AWS CloudFormation 範本。
2. 範本會建立用於匯入 S3 儲存貯體的 AWS CloudFormation 堆疊。

3. AWS CloudFormation 堆疊會管理您在範本中指定的 S3 儲存貯體。

技術, 堆

- AWS CloudFormation
- AWS Identity and Access Management (IAM)
- AWS KMS
- Amazon S3

工具

- AWS CloudFormation — AWS 可 CloudFormation 協助您以預測和重複的方式建立和佈建 AWS 基礎設施部署。
- AWS Identity and Access Management (IAM) — IAM 是一種用於安全控制 AWS 服務存取的 Web 服務。
- AWS KMS — AWS Key Management Service (AWS KMS) 是一種針對雲端擴展的加密和金鑰管理服務。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。

史诗

將具有以 CMK 為基礎的加密的 S3 儲存貯體匯入為 AWS 堆疊 CloudFormation

任務	描述	所需技能
建立範本以匯入 S3 儲存貯體和 CMK。	在您的本機電腦上，使用下列範本範本建立範本以匯入 S3 儲存貯體和 CMK： `AWSTemplateFormatVersion: 2010-09-09` `Parameters:` `  bucketName:`	AWS DevOps

任務	描述	所需技能
	Type: String Resources: S3Bucket: Type: 'AWS::S3::Bucket' DeletionPolicy: Retain Properties: BucketName: !Ref bucketName BucketEncryption: ServerSideEncryptionConfiguration: - ServerSideEncryptionByDefault: SSEAlgorithm: 'aws:kms' KMSMasterKeyID: !GetAtt - KMSS3Encryption - Arn KMSS3Encryption: Type: 'AWS::KMS::Key'	

任務	描述	所需技能
	DeletionPolicy: Retain Properties: Enabled: true KeyPolicy: !Sub |- { "Id": "key-consolepolicy-3", "Version": "2012-10-17", "Statement": [{ "Sid": "Enable IAM User Permissions", "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::${AWS::AccountId}:root"] }, "Action": "kms:*",	

任務	描述	所需技能
	<pre>"Resource": "*" } }] } EnableKey Rotation: true</pre>	
建立堆疊。	1. 登入 AWS 管理主控台，開啟 AWS 主控台，選擇 [檢視堆疊]，選擇 [建立堆疊]，然後選擇 [使用現有資源] (匯入資源)。 CloudFormation 2. 選擇 [上傳範本檔案]，然後上傳您先前建立的範本檔案。 3. 為您的堆疊輸入名稱，並根據您的需求設定剩餘的選項。 4. 選擇「建立堆疊」，然後等待堆疊的狀態變更為IMPORT_COMPLETE 。	AWS DevOps

任務	描述	所需技能
建立 KMS 金鑰別名。	1. 在 AWS CloudFormation 主控台上，選擇 [堆疊]，選擇您先前建立的堆疊名稱，選擇 [範本] 窗格，然後選擇 [在設計師中檢視]。 2. 將下列程式碼片段新增至範本`Resource`區段，然後選擇 [建立堆疊] 並完成精靈： `KMSS3EncryptionAlias:` `  Type: 'AWS::KMS ::Alias'` `  DeletionPolicy: Retain` `  Properties:` `  AliasName: alias/ S3BucketKey` `  TargetKeyId: !Ref KMSS3Encryption` 如需有關這方面的詳細資訊，請參閱 AWS CloudFormation 文件中的 AWS CloudFormation 堆疊更新。	AWS DevOps

任務	描述	所需技能
更新堆疊以包含 S3 儲存貯體政策。	1. 在 AWS CloudFormation 主控台上，選擇 [堆疊]，選擇您先前建立的堆疊名稱，選擇 [範本] 窗格，然後選擇 [在設計師中檢視]。 2. 將下列程式碼片段新增至範本Resource區段，然後選擇 [建立堆疊] 並完成精靈： S3BucketPolicy: Type: 'AWS::S3::BucketPolicy' Properties: Bucket: !Ref S3Bucket PolicyDocument: !Sub |- { "Version": "2008-10-17", "Id": "restricthttp", "Statement": [{ "Sid": "denyhttp",	AWS DevOps

任務	描述	所需技能
	<pre>"Effect": "Deny", "Principal": { "AWS": "*" }, "Action": "s3:*", "Resource": ["arn:aws:s3:::${S3Bucket}","arn:aws:s3:::${S3Bucket}/*"], "Condition": { "Bool": { "aws:SecureTransport": "false" } } }]</pre>	

任務	描述	所需技能
	} 注意：此 S3 儲存貯體政策具有拒絕陳述式，可限制不安全的 API 呼叫。	
更新金鑰原則。	1. 在 AWS CloudFormation 主控台上，選擇 [堆疊]，選擇您先前建立的堆疊名稱，選擇 [範本] 窗格，然後選擇 [在設計師中檢視]。 2. 修改範本的 KMS 資源，以包含可讓系統管理員管理 CMK 的金鑰原則。 3. 選擇 [建立堆疊]，選擇 [下一步]，然後根據您的需求完成精靈。 如需詳細資訊，請參閱 AWS KMS 文件中的使用金鑰政策和允許 AWS KMS 鑰管理員管理 CMK。	AWS 管理員

任務	描述	所需技能
新增資源層級標籤。	1. 在 AWS CloudFormation 主控台上，選擇 [堆疊]，選擇您先前建立的堆疊名稱，選擇 [範本] 窗格，然後選擇 [在設計師中檢視]。 2. 將下列程式碼片段新增至範本的 Amazon S3 資源`Properties` 區段，然後選擇「建立堆疊」並完成精靈： `Tags:` `  - Key: createdBy` `    Value: Cloudformation`	AWS DevOps

相關資源

- 將現有資源納入 AWS CloudFormation 管理
- AWS RE: 發明 2017 年：深入探討 AWS CloudFormation (影片)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

更多模式

- 使用工作階段管理員和 Amazon EC2 執行個體 Connect 存取防禦主機
- 將一個 AWS 帳戶中的 AWS CodeCommit 儲存庫與另一個帳戶中的 SageMaker 工作室建立關聯
- 使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目
- 自動執行 Amazon Lookout for Vision 訓練和部署，以進行異常偵測
- 使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation
- 使用 CI/CD 管道自動建置 Java 應用程式並將其部署到 Amazon EKS
- 使用 Python 在 AMS 中自動創建一個 RFC
- ???
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建置微焦點企業伺服器 PAC
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 啟動時檢查 EC2 執行個體是否有強制標籤
- 在 AWS 上設定適 NetBackup 用於 VMware 雲端的雲端
- 使用工作階段管理員 Connect 到 Amazon EC2 執行個體
- ???
- ???
- 使用 Amazon CloudWatch 異常偵測為自訂指標建立警示
- 使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統
- 自動為 Java 和 Python 項目創建動態 CI 管道
- 自動建立基於標籤的 Amazon CloudWatch 儀表板
- 使用 AWS 副駕駛員將叢集應用程式部署到 Amazon ECS
- 將反應型單頁應用程式部署到 Amazon S3 和 CloudFront
- 部署和偵錯 Amazon EKS 叢集
- 使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制 CloudFormation
- 使用地形表單部署和管理 AWS Control Tower 控制
- 使用 Elastic Beanstalk 部署容器
- 使用容器映像部署 Lambda 函數
- 使用 Amazon 基岩和 Amazon Transcribe 來記錄語音輸入的機構知識
- 啟動時強制執行 Amazon RDS 資料庫的自動標記
- 估算隨需容量的 DynamoDB 表格的成本

- 透過 Green Boost 探索全堆疊雲端原生 Web 應用程式開發
- 使用 AWS DMS 將 Amazon RDS for SQL Server 資料表匯出到 S3 儲存貯體
- 使用 Amazon Personalize 個人化產生個人化和重新排名的建議
- 使用 AWS AWS Glue 任務和 Python 產生測試資料
- 當 AWS KMS 金鑰的金鑰狀態變更時，取得 Amazon SNS 通知
- ???
- 在未使用 AWS KMS 金鑰加密 Amazon 資料 Firehose 資源時識別並發出警示
- 使用 AWS Step Functions 實作無伺服器傳奇模式
- 透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 管理多個 AWS 帳戶和 AWS 區域的 AWS 服務目錄產品
- 通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB
- 將 DNS 記錄批量遷移到 Amazon Route 53 私有託管區域
- 使用 SharePlex 和 AWS DMS 從甲骨文 8i 或 9i 遷移到適用於甲骨文的亞馬遜 RDS
- 監控 Amazon ElastiCache 叢集以進行靜態加密
- 在啟動時監控 Amazon EMR 叢集的傳輸中加密
- 監控安全群組的 ElastiCache 叢集
- 使用精確 Connect 將大型主機資料庫複寫到 AWS
- 在多區域、多帳戶組織中設定 AWS CloudFormation 漂移偵測
- 使用 AWS Lambda 在六角形架構中建構 Python 專案
- 在 SaaS 架構中使用 C# 和 AWS CDK 進行筒倉模型的租用戶上線
- 使用以下方式從 AWS IAM 身分中心更新 AWS CLI 登入資料 PowerShell
- 使用地形表單為組織自動啟 GuardDuty 用 Amazon
- 使用 Splunk 檢視 AWS Network Firewall 日誌和指標

容器與微服務

主題

- 使用 AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用 AWS PrivateLink 和 Network Load Balancer 在 Amazon EKS 上私下存取容器應用程式
- 使用 Amazon EKS 上的 AWS 私有 CA 在 AWS App Mesh 中啟用 MTL
- 使用 AWS Batch 為 Amazon RDS for PostgreSQL 資料庫執行個體自動備份
- 使用 CI/CD 管道，在 Amazon EKS 中自動部署節點終止處理程式
- 使用 CI/CD 管道自動建置 Java 應用程式並將其部署到 Amazon EKS
- 使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統
- 使用 AWS Fargate 在 Amazon ECS 上部署 Java 微服務
- 使用 Amazon ECR 和 AWS Fargate 在 Amazon ECS 上部署 Java 微服務
- 使用 Amazon ECR 和負載平衡在 Amazon ECS 上部署 Java 微服務
- 使用 Amazon EKS 和 Amazon S3 中的頭盔圖儲存庫來部署 Kubernetes 資源和套件
- 使用容器映像部署 Lambda 函數
- 在 Amazon EKS 上部署範例 Java 微服務，並使用應用程式負載平衡器公開微服務
- 使用 AWS 副駕駛員將叢集應用程式部署到 Amazon ECS
- 在 Amazon EKS 叢集上部署以 gRPC 為基礎的應用程式，並使 Application Load Balancer 存取
- 部署和偵錯 Amazon EKS 叢集
- 使用 Elastic Beanstalk 部署容器
- 使用 Lambda 函數、Amazon VPC 和無伺服器架構產生靜態輸出 IP 地址
- 使用 Kubernetes 在 Amazon EKS 工作者節點上安裝 SSM 代理程式 DaemonSet
- 在 Amazon EKS 工作者節點上安裝 SSM CloudWatch 代理程式和代理程式 preBootstrapCommands
- 優化 AWS 應用程序容器生成的碼頭映像
- 使用節點相似性、污點和容許，將 Kubernetes 網繭放置在 Amazon EKS 上
- 跨帳戶或區域複寫篩選過的 Amazon ECR 容器映像
- 輪換資料庫認證而不重新啟動

- WorkSpaces 使用 Amazon ECS 隨時隨地在 Amazon 上運行 Amazon ECS Anywhere 務
- 在 Amazon EC2 Linux 實例上運行一個 ASP.NET 核心網絡 API 碼頭容器
- 使用 AWS Fargate 大規模執行訊息導向工作負載
- 搭配 AWS Fargate 使用 Amazon EKS 上的 Amazon EFS，以持續性資料儲存執行可設定狀態工作負載
- 更多模式

使用 AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：生產	技術：容器與微服務；網路；安全性、身分識別、合規性；Web 和行動應用程式	工作負載：所有其他工作
AWS 服務：Amazon EC2; Amazon EC2 Auto Scaling; Amazon EC2 容器註冊表; Amazon EFS; Amazon RDS; Amazon VPC; Amazon ECS; Elastic Load Balancing (ELB); AWS Lambda		

Summary

此模式描述如何在 Network Load Balancer 後方的 Amazon Elastic Container Service (Amazon ECS) 上私有託管 Docker 容器應用程式，以及如何使用 AWS 存取應用程式。 PrivateLink然後，您可以使用私有網路安全地存取 Amazon Web Services (AWS) 雲端上的服務。Amazon Relational Database Service 服務 (Amazon RDS) 託管在具有高可用性 (HA) 的 Amazon ECS 上執行之應用程式的關聯式資料庫。如果應用程式需要持續性儲存，則會使用 Amazon Elastic File System (Amazon EFS)。

執行 Docker 應用程式的 Amazon ECS 服務 (前端有 Network Load Balancer) 可與虛擬私有雲端 (VPC) 端點建立關聯，以便透過 AWS 存取。 PrivateLink然後，可以使用其他 VPC 端點與其他 VPC 共用此 VPC 端點服務。

您也可以使用 AWS Fargate 取代 Amazon EC2 Auto Scaling 群組。如需詳細資訊，請參閱使用 AWS Fargate、AWS 和 Network Load Balancer 在 Amazon ECS 上私下存取容器應用程式。 PrivateLink

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已在 Linux、macOS 或視窗上安裝和設定
- 泊塢視窗, 安裝和配置在 Linux, macOS, 或視窗
- 在碼頭上運行的應用程序

架構

技術堆疊

- Amazon CloudWatch
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon EC2 Auto Scaling
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon RDS
- Amazon Simple Storage Service (Amazon S3)
- AWS Lambda
- AWS PrivateLink
- AWS Secrets Manager
- Application Load Balancer
- Network Load Balancer
- VPC

自動化和規模

- 您可以使用 AWS 使用基礎設施即程式碼 CloudFormation來建立此模式。

工具

- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。

- Amazon EC2 Auto Scaling — Amazon EC2 Auto Scaling 可協助您確保擁有正確數量的 Amazon EC2 執行個體可用來處理應用程式的負載。
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，可讓您輕鬆執行、停止和管理叢集上的容器。
- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種受管 AWS 容器映像登錄服務，安全、可擴展且可靠。
- Amazon EFS — Amazon Elastic File System (Amazon EFS) 提供簡單、可擴展且全受管的彈性 NFS 檔案系統，可與 AWS 雲端服務和現場部署資源搭配使用。
- AWS Lambda — L ambda 是一種運算服務，可用來執行程式碼，無需佈建或管理伺服器
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。此服務旨在降低開發人員進行網路規模運算的難度。
- AWS Secrets Manager — Secrets Manager 提供 API 呼叫以程式設計方式擷取密碼，協助您取代程式碼中的硬式編碼登入 Secrets Manager 料 (包括密碼)。
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。
- E@@ lastic Load Balancing — Elastic Load Balancing 可將傳入的應用程式或網路流量分散到多個可用區域中的多個目標，例如 Amazon EC2 執行個體、容器和 IP 地址。
- Docker — Doc ker 幫助開發人員將任何應用程序打包，運送和運行作為一個輕量級，便攜和自給自足的容器。

史诗

建立網路元件

任務	描述	所需技能
建立 VPC。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。選擇創建 VPC，然後選擇 VPC 等等。 2. 輸入 VPC 的名稱，然後選擇適當的 CIDR 區塊範圍。	雲端管理員

任務	描述	所需技能
	3. 指定兩個可用區域、兩個公用子網路、四個私有子網路。兩個私有子網路用於 Amazon ECS 任務，而兩個私有子網路則用於 Amazon RDS 資料庫。 4. 為每個可用區域指定一個 NAT 閘道。 5. 選擇建立 VPC。	

建立負載平衡器

任務	描述	所需技能
建立 Network Load Balancer。	1. 開啟 Amazon EC2 主控台，然後選擇包含 VPC 的 AWS 區域。 2. 在負載平衡下，選擇負載平衡器，然後選擇建立負載平衡器。 3. 選擇 Network Load Balancer，然後選擇建立。 4. 在 [設定負載平衡器] 頁面上，設定您的 Network Load Balancer 和接聽程式。 重要:請確定您選擇網路負載平衡器的配置為內部。 5. 選擇適用的安全性設定、設定安全群組和目標群組。在 [設定路由] 區段中，選擇 [執行個體] 或 [IP] 做為 [目標] 請確定您沒有註冊目標。	雲端管理員

任務	描述	所需技能
	6. 設定完所有設定後，請選擇 [下一步：複查]，然後選擇 [建立]。	
建立應用程式負載平衡器。	1. 在 Amazon EC2 主控台上，選擇包含您 VPC 的相同區域。 2. 在負載平衡下，選擇負載平衡器，然後選擇建立負載平衡器。 3. 選擇 Application Load Balancer，然後選擇建立。 4. 設定應用程式負載平衡器及其監聽器。 重要：請確定您選擇應用程式負載平衡器的配置為內部。 5. 選擇適用的安全性設定、設定安全群組和目標群組。在 [設定路由] 區段中，選擇 [執行個體] 或 [IP] 做為 [目標] 請確定您沒有註冊目標。 6. 設定完所有設定後，請選擇 [下一步：複查]，然後選擇 [建立]。	雲端管理員

建立 Amazon EFS 檔案系統

任務	描述	所需技能
建立一個 Amazon EFS 檔案系統。	1. 開啟 Amazon EFS 主控台，然後選擇「建立檔案系統」。	雲端管理員

任務	描述	所需技能
	2. 在 [建立檔案系統] 對話方塊中，輸入檔案系統的名稱，然後選擇您的 VPC。 3. 選擇 [建立] 以建立檔案系統。 4. 設定和設定您的 Amazon EFS 檔案系統。	
掛載子網路的目標。	1. 返回 Amazon EFS 主控台，然後選擇檔案系統。檔案系統頁面會顯示您帳戶中的 Amazon EFS 檔案系統。 2. 選擇您建立的檔案系統，然後選擇管理以顯示可用區域。若要新增掛載目標，請選擇 [新增掛載目標]，然後新增您建立的四個私人子網路。	雲端管理員
確認子網路已裝載為目標。	1. 在 Amazon EFS 主控台上，選擇檔案系統。 2. 選擇網路以顯示現有掛載目標的清單。請確定這些子網路包括您建立的四個子網路。	雲端管理員

建立 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	如有需要，開啟 Amazon S3 主控台並建立 S3 儲存貯體來存放應用程式的靜態資產。	雲端管理員

建立密碼管理員密碼

任務	描述	所需技能
建立 AWS KMS 金鑰以加密密碼 Secrets Manager 密碼。	開啟 AWS Key Management Service (AWS KMS) 主控台，然後建立 KMS 金鑰。	雲端管理員
創建一個 Secrets Manager 密碼來存儲 Amazon RDS 密碼。	1. 開啟 AWS Secrets Manager 主控台，然後選擇存放新密碼來建立新密碼。 2. 選擇您建立的 KMS 金鑰，然後儲存新密碼。	雲端管理員

創建一個 Amazon RDS 實例

任務	描述	所需技能
建立資料庫子網路群組。	1. 開啟 Amazon RDS 主控台，然後選擇子網路群組。 2. 選擇 [建立資料庫子網路群組]，然後輸入資料庫子網路群組的名稱和說明。 3. 選擇您先前建立的 VPC，然後選擇可用區域和子網路。然後選擇 Create (建立)。	雲端管理員
創建一個 Amazon RDS 實例。	在私有子網路中建立和設定 Amazon RDS 執行個體。確定已開啟 HA 的異地同步備份。	雲端管理員
將資料載入 Amazon RDS 執行個體。	將應用程式所需的關聯式資料載入 Amazon RDS 執行個體。此過程將根據應用程序的需求以及數據庫模式的定義和設計方式而有所不同。	雲端管理員，DBA

創建 Amazon ECS 組件

任務	描述	所需技能
建立 ECS 叢集。	1. 開啟 Amazon ECS 主控台，然後選擇「叢集」。 2. 選擇 [建立叢集]，然後根據您所需的規格設定 ECS 叢集。	雲端管理員
創建碼頭圖像。	依照「相關資源」一節中的指示建立 Docker 映像檔。	雲端管理員
建立 Amazon ECR 儲存庫。	1. 在 Amazon ECR 主控台上，選擇儲存庫。 2. 選擇 [建立儲存庫]，然後輸入儲存庫的唯一名稱。 3. 根據您的規格設定存放庫，包括必要時的 AWS KMS 加密。	雲端管理員、 DevOps 工程師
驗證 Amazon ECR 儲存庫的 Docker 用戶端。	若要驗證 Amazon ECR 儲存庫的 Docker 用戶端，請在 AWS CLI 中執行「`aws ecr get-login-password` 命令。	雲端管理員
將碼頭映像推送到 Amazon ECR 存儲庫。	1. 識別您要推送的 Docker 映像，然後在 AWS CLI 中執行`docker images`命令。 2. 使用 Amazon ECR 登錄、儲存庫和選用的映像標籤名稱組合來標記您的映像。 3. 通過運行`docker push`命令推送 Docker 映像。 4. 針對所有必要的影像重複這些步驟。	雲端管理員

任務	描述	所需技能
建立 Amazon ECS 任務定義。	在 Amazon ECS 中執行 Docker 容器所需的任務定義。 1. 返回 Amazon ECS 主控台，選擇「任務定義」，然後選擇「建立新的任務定義」。 2. 在 [選取相容性] 頁面上，選取工作應使用的啟動類型，然後選擇 [下一步]。 如需設定任務定義的說明，請參閱「相關資源」一節中的「建立任務定義」。 重要:請務必提供您推送至 Amazon ECR 的泊塢視窗映像檔。	雲端管理員
建立 Amazon ECS 服務。	使用您先前建立的 ECS 叢集來建立 Amazon ECS 服務。請務必選擇 Amazon EC2 做為啟動類型，並選擇在上一步中建立的任務定義，以及 Application Load Balancer 的目標群組。	雲端管理員

創建一個 Amazon EC2 Auto Scaling 組

任務	描述	所需技能
建立啟動組態。	開啟 Amazon EC2 主控台，然後建立啟動組態。確定使用者資料具有允許 EC2 執行個體加入所需 ECS 叢集的程式碼。如	雲端管理員

任務	描述	所需技能
	需所需程式碼的範例，請參閱「相關資源」一節。	
創建一個 Amazon EC2 Auto Scaling 組。	返回 Amazon EC2 主控台，然後在「Auto Scaling」下選擇「Auto Scaling」群組。設置一個 Amazon EC2 Auto Scaling 組。請確定您選擇私有子網路，並啟動先前建立的組態。	雲端管理員

設定 AWS PrivateLink

任務	描述	所需技能
設定 AWS PrivateLink 端點。	1. 在 Amazon VPC 端主控台上，建立 AWS PrivateLink 端點。 2. 將此端點與 Network Load Balancer 建立關聯，讓 Amazon ECS 上託管的應用程式可私下供客戶使用。 如需詳細資訊，請參閱相關資源一節。	雲端管理員

建立 VPC 端點

任務	描述	所需技能
建立 VPC 端點。	為您先前建立的 AWS PrivateLink 端點建立 VPC 端點。VPC 端點完整網域名稱 (FQDN) 將指向 AWS PrivateLi	雲端管理員

任務	描述	所需技能
	nk 端點 FQDN。這會為 DNS 端點可存取的 VPC 端點服務建立 elastic network interface。	

建立 Lambda 函式

任務	描述	所需技能
建立 Lambda 函數。	在 AWS Lambda 主控台上，建立 Lambda 函數，將 Application Load Balancer IP 地址更新為 Network Load Balancer 的目標。如需詳細資訊，請參閱「相關資源」一節中的「針對應用程式負載平衡器使用靜態 IP 位址」部落格文章。	應用程式開發人員

相關資源

建立負載平衡器：

- 建立 Network Load Balancer
- 建立應用程式負載平衡器

建立 Amazon EFS 檔案系統：

- 建立 Amazon EFS 檔案系統
- 在 Amazon EFS 中建立掛載目標

建立 S3 儲存貯體：

- 建立 S3 儲存貯體

建立密碼管理員密碼：

- 在 AWS KMS 中建立金鑰
- 在 AWS Secrets Manager 中建立密碼

創建一個 Amazon RDS 實例：

- 建立 Amazon RDS 資料庫執行個體

創建 Amazon ECS 組件：

- 建立 Amazon ECS 叢集
- 建立泊塢視窗映像
- 建立 Amazon ECR 儲存庫
- 使用 Amazon ECR 儲存庫驗證碼頭工人
- 將映像推送到 Amazon ECR 存儲庫
- 建立 Amazon ECS 任務定義
- 創建一個 Amazon ECS 服務

創建一個 Amazon EC2 Auto Scaling 組：

- 建立啟動組態
- 使用啟動組態建立 Auto Scaling 群組
- 使用 Amazon EC2 使用者資料啟動容器執行個體

設定 AWS PrivateLink：

- VPC 端端點服務 (AWS PrivateLink)

建立 VPC 端點：

- 接口 VPC 端端點 (AWS PrivateLink)

創建 Lambda 函數：

- 創建一個 Lambda 函數

其他資源：

- 針對應用程式負載平衡器使用靜態 IP 位址
- 透過 AWS 安全地存取服務 PrivateLink

使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：生產	技術：容器與微服務；網路；安全性、身分識別、合規性；Web 和行動應用程式	工作負載：所有其他工作
AWS 服務：Amazon EC2 容器註冊表；Amazon ECS；Amazon EFS；Amazon RDS；Amazon VPC；Elastic Load Balancing（ELB）；AWS Lambda		

Summary

此模式描述如何透過使用 Amazon 彈性容器服務 (Amazon ECS) 搭配 AWS Fargate 啟動類型，在 Network Load Balancer 後面使用 Amazon 彈性容器服務 (AWS)，在網路負載平衡器後面私有託管 Docker 容器應用程式，以及使用 AWS 存取應用程式。 PrivateLinkAmazon Relational Database Service 服務 (Amazon RDS) 託管在具有高可用性 (HA) 的 Amazon ECS 上執行之應用程式的關聯式資料庫。如果應用程式需要持續性儲存，您可以使用 Amazon Elastic File System (Amazon EFS)。

此模式針對執行 Docker 應用程式的 Amazon ECS 服務使用 Fargate 啟動類型，並在前端使用 Network Load Balancer。然後，它可以與虛擬私有雲端 (VPC) 端點建立關聯，以便透過 AWS PrivateLink 存取。然後，可以使用其他 VPC 端點與其他 VPC 共用此 VPC 端點服務。

您可以將 Fargate 與 Amazon ECS 搭配使用來執行容器，而不必管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的叢集。您也可以使用 Amazon EC2 Auto Scaling 組而不是 Fargate。如需詳細資訊，請參閱使用 AWS PrivateLink 和 Network Load Balancer 在 Amazon ECS 上私下存取容器應用程式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已在 Linux、macOS 或視窗上安裝和設定
- 泊塢視窗, 安裝和配置在 Linux, macOS, 或視窗
- 在碼頭上運行的應用程序

架構

技術堆疊

- Amazon CloudWatch
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- Amazon EFS
- Amazon RDS
- Amazon Simple Storage Service (Amazon S3)
- AWS Fargate
- AWS Lambda
- AWS PrivateLink
- AWS Secrets Manager
- Application Load Balancer
- Network Load Balancer
- VPC

自動化和規模

- 您可以使用 AWS 使用基礎設施即程式碼 CloudFormation來建立此模式。

工具

- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，可讓您輕鬆執行、停止和管理叢集上的容器。
- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種受管 AWS 容器映像登錄服務，安全、可擴展且可靠。
- Amazon EFS — Amazon Elastic File System (Amazon EFS) 提供簡單、可擴展且全受管的彈性 NFS 檔案系統，可與 AWS 雲端服務和現場部署資源搭配使用。
- AWS Fargate — AWS Fargate 是一項技術，您可以與 Amazon ECS 搭配使用來執行容器，而不必管理伺服器或 Amazon EC2 執行個體的叢集。
- AWS Lambda — Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。此服務旨在降低開發人員進行網路規模運算的難度。
- AWS Secrets Manager — Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。
- E@@ lastic Load Balancing — Elastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個可用區域中的多個目標，例如 EC2 執行個體、容器和 IP 地址。
- Docker — Doc ker 幫助開發人員輕鬆打包，運送和運行任何應用程序作為一個輕量級，便攜和自給自足的容器。

史诗

建立網路元件

任務	描述	所需技能
建立 VPC。	1. 登入 AWS 管理主控台，然後開啟 Amazon VPC 主控台。選擇創建 VPC，然後選擇 VPC 等等。	雲端管理員

任務	描述	所需技能
	2. 輸入 VPC 的名稱，然後選擇適當的 CIDR 區塊範圍。 3. 指定兩個可用區域、兩個公用子網路、四個私有子網路。兩個私有子網路用於 Amazon ECS 任務，而兩個私有子網路則用於 Amazon RDS 資料庫。 4. 為每個可用區域指定一個 NAT 閘道。 5. 選擇建立 VPC。	

建立負載平衡器

任務	描述	所需技能
建立 Network Load Balancer。	1. 開啟 Amazon EC2 主控台，然後選擇包含您 VPC 的 AWS 區域。 2. 在負載平衡下，選擇負載平衡器，然後選擇建立負載平衡器。 3. 選擇 Network Load Balancer，然後選擇建立。 4. 在 [設定負載平衡器] 頁面上，設定您的 Network Load Balancer 和接聽程式。重要事項：請確定您選擇網路負載平衡器的配置為「內部」。 5. 選擇適用的安全性設定、設定安全群組和目標群組。在 [設定路由] 區段中選擇 [IP]	雲端管理員

任務	描述	所需技能
	作為 [目標類型]。請確定您沒有註冊目標。 6. 設定完所有設定後，請選擇 [下一步：複查]，然後選擇 [建立]。 如需此和其他故事的說明，請參閱「相關資源」一節。	
建立應用程式負載平衡器。	1. 在 Amazon EC2 主控台上，選擇包含 VPC 的相同區域。 2. 在負載平衡下，選擇負載平衡器，然後選擇建立負載平衡器。 3. 選擇 Application Load Balancer，然後選擇建立。 4. 設定應用程式負載平衡器及其監聽器。重要：請確定您選擇應用程式負載平衡器的配置為內部。 5. 選擇適用的安全性設定、設定安全群組和目標群組。在 [設定路由] 區段中選擇 [IP] 作為 [目標類型]。請確定您沒有註冊目標。 6. 設定完所有設定後，請選擇 [下一步：複查]，然後選擇 [建立]。	雲端管理員

建立 Amazon EFS 檔案系統

任務	描述	所需技能
建立一個 Amazon EFS 檔案系統。	1. 開啟 Amazon EFS 主控台，然後選擇「建立檔案系統」。 2. 在 [建立檔案系統] 對話方塊中，輸入檔案系統的名稱，然後選擇您的 VPC。 3. 選擇 [建立] 以建立檔案系統。 4. 設定和設定您的 Amazon EFS 檔案系統。	雲端管理員
掛載子網路的目標。	1. 返回 Amazon EFS 主控台，然後選擇檔案系統。檔案系統頁面會顯示您帳戶中的 Amazon EFS 檔案系統。 2. 選擇您建立的檔案系統，然後選擇管理以顯示可用區域。 3. 若要新增掛載目標，請選擇 [新增掛載目標]，然後新增您建立的四個私人子網路。	雲端管理員
確認子網路已裝載為目標。	1. 在 Amazon EFS 主控台上，選擇檔案系統。 2. 選擇網路以顯示現有掛載目標的清單。請確定這些子網路包括您建立的四個子網路。	雲端管理員

建立 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	如有需要，開啟 Amazon S3 主控台並建立 S3 儲存貯體來存放應用程式的靜態資產。	雲端管理員

建立密碼管理員密碼

任務	描述	所需技能
建立 AWS KMS 金鑰以加密密碼 Secrets Manager 密碼。	開啟 AWS Key Management Service (AWS KMS) 主控台並建立 KMS 金鑰。	雲端管理員
創建一個 Secrets Manager 密碼來存儲 Amazon RDS 密碼。	1. 開啟 AWS Secrets Manager 主控台，然後選擇存放新密碼來建立新密碼。 2. 選擇您建立的 KMS 金鑰，然後儲存新密碼。	雲端管理員

創建一個 Amazon RDS 實例

任務	描述	所需技能
建立資料庫子網路群組。	1. 開啟 Amazon RDS 主控台，然後選擇子網路群組。 2. 選擇 [建立資料庫子網路群組]，然後輸入資料庫子網路群組的名稱和說明。 3. 選擇您先前建立的 VPC，然後選擇可用區域和子網路。然後選擇 Create (建立)。	雲端管理員

任務	描述	所需技能
創建一個 Amazon RDS 實例。	在私有子網路中建立和設定 Amazon RDS 執行個體。確定已開啟異地同步備份以取得高可用性 (HA)。	雲端管理員
將資料載入 Amazon RDS 執行個體。	將應用程式所需的關聯式資料載入 Amazon RDS 執行個體。此過程將根據應用程序的需求以及數據庫模式的定義和設計方式而有所不同。	DBA

創建 Amazon ECS 組件

任務	描述	所需技能
建立 ECS 叢集。	1. 開啟 Amazon ECS 主控台，然後選擇「叢集」。 2. 選擇 [建立叢集]，然後根據您所需的規格設定 ECS 叢集。	雲端管理員
創建碼頭圖像。	依照「相關資源」一節中的指示建立 Docker 映像檔。	雲端管理員
建立 Amazon ECR 儲存庫。	1. 開啟 Amazon ECR 主控台，然後選擇儲存庫。 2. 選擇 [建立儲存庫]，然後輸入儲存庫的唯一名稱。 3. 根據您的規格設定存放庫，包括必要時的 AWS KMS 加密。	雲端管理員、 DevOps 工程師
將碼頭映像推送到 Amazon ECR 存儲庫。	1. 識別您要推送的 Docker 映像，然後在 AWS CLI 中執行`docker images`命令。	雲端管理員

任務	描述	所需技能
	2. 使用 Amazon ECR 登錄、儲存庫和選用的映像標籤名稱組合來標記您的映像。 3. 通過運行`docker push`命令推送 Docker 映像。 4. 針對所有必要的影像重複這些步驟。	
建立 Amazon ECS 任務定義。	在 Amazon ECS 中執行 Docker 容器所需的任務定義。 1. 返回 Amazon ECS 主控台，選擇「任務定義」，然後選擇「建立新的任務定義」。 2. 在 [選取相容性] 頁面上，選取工作應使用的啟動類型，然後選擇 [下一步]。 如需設定任務定義的說明，請參閱「相關資源」一節中的「建立任務定義」。重要:請務必提供您推送至 Amazon ECR 的泊塢視窗映像檔。	雲端管理員
建立 ECS 服務並選擇 Fargate 作為啟動類型。	1. 使用您先前建立的 ECS 叢集來建立 Amazon ECS 服務。確保選擇 Fargate 作為啟動類型。 2. 選擇在上一個步驟中建立的作業定義，然後選擇「Application Load Balancer」的目標群組。	雲端管理員

設定 AWS PrivateLink

任務	描述	所需技能
設定 AWS PrivateLink 端點。	1. 開啟 Amazon VPC 人雲端主控台，然後建立 AWS PrivateLink 端點。 2. 將此端點與 Network Load Balancer 建立關聯，這可讓 Amazon ECS 上託管的應用程式私下提供給客戶。 如需詳細資訊，請參閱相關資源一節。	雲端管理員

建立 VPC 端點

任務	描述	所需技能
建立 VPC 端點。	為您先前建立的 AWS PrivateLink 端點建立 VPC 端點。VPC 端點完整網域名稱 (FQDN) 將指向 AWS PrivateLink 端點 FQDN。這會建立網域名稱服務端點可存取的 VPC 端點服務的 elastic network interface。	雲端管理員

建立 Lambda 函式

任務	描述	所需技能
建立 Lambda 函數。	開啟 Lambda 主控台並建立 Lambda 函數，將 Application Load Balancer IP 位址更新為	應用程式開發人員

任務	描述	所需技能
	Network Load Balancer 的目標。如需詳細資訊，請參閱「相關資源」一節中的「針對應用程式負載平衡器使用靜態 IP 位址」部落格文章。	

相關資源

建立負載平衡器：

- 建立 Network Load Balancer
- 建立應用程式負載平衡器

建立 Amazon EFS 檔案系統：

- 建立 Amazon EFS 檔案系統
- 在 Amazon EFS 中建立掛載目標

建立 S3 儲存貯體：

- 建立 S3 儲存貯體

建立密碼管理員密碼：

- 在 AWS KMS 中建立金鑰
- 在 AWS Secrets Manager 中建立密碼

創建一個 Amazon RDS 實例：

- 建立 Amazon RDS 資料庫執行個體

創建 Amazon ECS 組件：

- 建立 Amazon ECS 叢集

- 建立泊塢視窗映像
- 創建一個 Amazon ECR 存儲庫
- 使用 Amazon ECR 儲存庫驗證碼頭工人
- 將映像推送到 Amazon ECR 存儲庫
- 建立 Amazon ECS 任務定義
- 創建一個 Amazon ECS 服務

設定 AWS PrivateLink：

- VPC 端端點服務 (AWS PrivateLink)

建立 VPC 端點：

- 接口 VPC 端端點 (AWS PrivateLink)

創建 Lambda 函數：

- 創建一個 Lambda 函數

其他資源：

- 針對應用程式負載平衡器使用靜態 IP 位址
- 透過 AWS 安全地存取服務 PrivateLink

使用 AWS PrivateLink 和 Network Load Balancer 在 Amazon EKS 上私下存取容器應用程式

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：生產	技術：容器與微服務；現代化 DevOps；安全性、身分識別、合規	工作負載：所有其他工作
AWS 服務：Amazon EKS; Amazon VPC		

Summary

此模式說明如何在 Network Load Balancer 後方的 Amazon Elastic Kubernetes Service (Amazon EKS) 上以私密方式託管 Docker 容器應用程式，以及如何使用 AWS 存取應用程式。 PrivateLink然後，您可以使用私有網路安全地存取 Amazon Web Services (AWS) 雲端上的服務。

執行 Docker 應用程式的 Amazon EKS 叢集 (前端有 Network Load Balancer) 可與虛擬私有雲端 (VPC) 端點建立關聯，以便透過 AWS 存取。 PrivateLink然後，可以使用其他 VPC 端點與其他 VPC 共用此 VPC 端點服務。

此模式描述的設定是在 VPC 和 AWS 帳戶之間共用應用程式存取權的安全方式。它不需要特殊的連線或路由組態，因為消費者和供應商帳戶之間的連線位於全球 AWS 骨幹網上，而且不會周遊公用網際網路。

先決條件和限制

先決條件

- 碼頭, 安裝和配置在 Linux 上, macOS, 或視窗.
- 在碼頭上運行的應用程序。
- 作用中的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已在 Linux、macOS 或視窗上安裝和設定。

- 具有標記私有子網路且設定為託管應用程式的現有 Amazon EKS 叢集。如需詳細資訊，請參閱 Amazon EKS 文件中的子網路標記。
- 已安裝和設定以存取 Amazon EKS 叢集上的資源。如需詳細資訊，請參閱 Amazon EKS 文件中的安裝 kubectl。

架構

技術堆疊

- Amazon EKS
- AWS PrivateLink
- Network Load Balancer

自動化和規模

- Kubernetes 資訊清單可在 Git 型儲存庫 (例如，在 AWS 上) 追蹤和管理，並在 AWS 中使用持續整合和持續交付 (CI/CD CodeCommit) 進行部署。 CodePipeline
- 您可以使用 AWS 使用基礎設施 CloudFormation 即程式碼 (IaC) 來建立此模式。

工具

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- E@@ lastic Load Balancing — Elastic Load Balancing 可將傳入的應用程式或網路流量分散到多個目標，例如 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、容器和 IP 地址，分散在一或多個可用區域中。

- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，您可以使用它在 AWS 上執行 Kubernetes，而無需安裝、操作和維護自己的 Kubernetes 控制平面或節點。
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。
- Kubectl — Kubectl 是一個命令列公用程式，可針對 Kubernetes 叢集執行命令。

史诗

部署 Kubernetes 部署和服務資訊清單檔案

任務	描述	所需技能
建立 Kubernetes 部署資訊清單檔案。	根據您的需求修改下列範例檔案，以建立部署資訊清單檔案。 `apiVersion: apps/v1` `kind: Deployment` `metadata:` `  name: sample-app` `spec:` `  replicas: 3` `  selector:` `  matchLabels:` `  app: nginx` `  template:` `  metadata:` `  labels:` `  app: nginx` `  spec:` `  containers:` `  - name: nginx` `  image:` ` public.ecr.aws/z9d2n7e1/nginx:1.19.5` `  ports:` `  - name: http` `  container Port: 80` 注意：這是使用 NGINX 碼頭映像部署的 NGINX 範例組態檔案。有關更多信息，請參閱 Docker 文檔中的如何使用官方的 NGINX Docker 映像。	DevOps 工程師

任務	描述	所需技能
部署 Kubernetes 部署資訊清單檔案。	執行下列命令，將部署資訊清單檔案套用至 Amazon EKS 叢集： `kubectl apply -f <your_deployment_file_name>`	DevOps 工程師

任務	描述	所需技能
建立 Kubernetes 服務資訊清單檔案。	根據您的需求修改下列範例檔案，以建立服務資訊清單檔案。 `apiVersion: v1` `kind: Service` `metadata:` `  name: sample-service` `  annotations:` `  service.beta.kubernetes.io/aws-load-balancer-type: nlb` `  service.beta.kubernetes.io/aws-load-balancer-internal: "true"` `spec:` `  ports:` `  - port: 80` `  targetPort: 80` `  protocol: TCP` `  type: LoadBalancer` `  selector:` `  app: nginx` 重要：請確定您包含下列項目，`annotations` 以定義內部 Network Load Balancer： `service.beta.kubernetes.io/aws-load-balancer-type: nlb` `service.beta.kubernetes.io/aws-load-balancer-internal: "true"`	DevOps 工程師

任務	描述	所需技能
部署 Kubernetes 服務資訊清單檔案。	執行下列命令，將服務資訊清單檔案套用至 Amazon EKS 叢集： `kubectl apply -f <your_service_file_name>`	DevOps 工程師

建立端點

任務	描述	所需技能
記錄網路負載平衡器的名稱。	執行下列命令以擷取 Network Load Balancer 的名稱： `kubectl get svc sample-service -o wide` 記錄建立 AWS PrivateLink 端點所需的網路負載平衡器名稱。	DevOps 工程師
建立 AWS PrivateLink 端點。	登入 AWS 管理主控台，開啟 Amazon VPC 主控台，然後建立 AWS PrivateLink 端點。將此端點與 Network Load Balancer 建立關聯，這樣可讓客戶以私密方式使用應用程式。如需詳細資訊，請參閱 Amazon VPC 文件中的 VPC 端點服務 (AWS PrivateLink)。 重要事項：如果消費者帳戶需要存取應用程式，則必須將消	雲端管理員

任務	描述	所需技能
	費者帳戶的 AWS 帳戶 ID 新增到 AWS PrivateLink 端點組態的允許主體清單中。如需詳細資訊，請參閱 Amazon VPC 文件中的新增和移除端點服務的許可。	
建立 VPC 端點。	在 Amazon VPC 主控台上，選擇端點服務，然後選擇建立端點服務。為 AWS 端點建立 VPC PrivateLink 端點。 VPC 端點的完整網域名稱 (FQDN) 會指向 AWS 端點的 FQDN。 PrivateLink 這會為 DNS 端點可存取的 VPC 端點服務建立 elastic network interface。	雲端管理員

相關資源

- 使用官方的 NGINX 泊塢視窗映像
- Amazon EKS 上的網絡負載平衡
- 建立 VPC 端端點服務 (AWS PrivateLink)
- 新增和移除端點服務的權限

使用 Amazon EKS 上的 AWS 私有 CA 在 AWS App Mesh 中啟用 MTL

由奧馬爾·卡希爾（AWS），伊曼紐爾·薩利烏（AWS）和穆罕默德·沙赫扎德（AWS）創建

環境：PoC 或試點	技術：容器與微服務	AWS 服務：AWS App Mesh；Amazon EKS；AWS Certificate Manager (ACM)

Summary

此模式顯示如何在 AWS 應用程式 Mesh 中使用 AWS 私有憑證授權機構 (AWS 私有 CA) 的憑證在 Amazon Web Services (AWS) 上實作相互傳輸層安全性 (MTL)。它通過適用於所有人的安全生產身份識別框架（SPIFE）使用特使秘密發現服務（SDS）API。SPIFFE 是雲端原生運算基礎 (CNCF) 開放原始碼專案，具有廣泛的社群支援，可提供精細且動態的工作負載身分識別管理。若要實作 SPIFE 標準，請使用 SPIRE SPIFE 執行階段環境。

在 App Mesh 中使用 MTL 可提供雙向對等驗證，因為它會透過 TLS 增加一層安全性，並允許網狀中的服務驗證正在建立連線的用戶端。用戶端與伺服器關係中的用戶端也會在工作階段交涉程序期間提供 X.509 憑證。伺服器會使用此憑證來識別和驗證用戶端。這有助於驗證證書是否由受信任的證書頒發機構（CA）以及證書是否為有效證書。

先決條件和限制

先決條件

- 具有自我管理或受管節點群組的亞馬 Amazon Elastic Kubernetes Service (Amazon EKS) 叢集
- 部署在已啟動 SDS 的叢集上的 App Mesh 控制器
- 由 AWS 私有 CA 核發的 AWS Certificate Manager (ACM) 的私有憑證

限制

- SPIRE 無法安裝在 AWS Fargate 上，因為 SPIRE 代理程式必須以庫 DaemonSet伯尼特人的身分執行。

產品版本

- AWS App Mesh 控制器圖表 1.3.0 或更新版本

架構

下圖顯示了 VPC 中具有應用程序網格的 EKS 集群。一個工作節點中的 SPIRE 伺服器會與其他工作節點中的 SPIRE 代理程式以及 AWS 私有 CA 通訊。特使用於 SPIRE 代理工作節點之間的 MTL 通信。

此圖說明了下列步驟：

1. 憑證已發行。
2. 請求證書簽名和證書。

工具

AWS 服務

- AWS 私有 CA — AWS 私有憑證授權單位 (AWS Private CA) 可讓您建立私有憑證授權單位 (CA) 階層，包括根 CA 和從屬 CA，而無須支付操作現場部署 CA 的投資和維護成本。
- AWS App Mesh — AWS App Mesh 是一種服務網格，可讓您更輕鬆地監控和控制服務。App Mesh 可標準化您的服務通訊方式，為應用程式中的每個服務提供一致的可見性和網路流量控制。
- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，您可以使用它在 AWS 上執行 Kubernetes，而無需安裝、操作和維護自己的 Kubernetes 控制平面或節點。

其他工具

- 掌舵 — Helm 是 Kubernetes 的套件管理員，可協助您在 Kubernetes 叢集上安裝及管理應用程式。此模式使用 Helm 來部署 AWS App Mesh 控制器。
- AWS 應用程式網狀控制器圖表 — 此模式使用 AWS 應用程式網狀控制器圖表在 Amazon EKS 上啟用 AWS App Mesh。

史诗

設定環境

任務	描述	所需技能
使用 Amazon EKS 設置 App Mesh。	遵循儲存庫中提供的基本部署步驟。	DevOps 工程師
安裝尖塔。	在 EKS 叢集上使用安裝尖塔。	DevOps 工程師
安裝 AWS 私有 CA 憑證。	依照 AWS 文件中的指示，為您的私有根 CA 建立並安裝憑證。	DevOps 工程師
授與叢集節點執行個體角色的權限。	若要將原則附加至叢集節點執行個體角色，請使用 [其他資訊] 區段中的程式碼。	DevOps 工程師
為 AWS 私有 CA 新增 SPIRE 外掛程式。	要將插件添加到 SPIRE 服務器配置中，請使用其他信息部分中的代碼。將 `certificate_authority_arn` Amazon 資源名稱（ARN）替換為您的私有 CA ARN。使用的簽署演算法必須與私有 CA 上的簽署演算法相同。`your_region` 以您的 AWS 區域取代。 如需有關外掛程式的詳細資訊，請參閱伺服器外掛程式：UpstreamAuthority 「aws_pca」。	DevOps 工程師
更新包。證書。	建立 SPIRE 伺服器之後，將會建立一個`spire-bundle.yaml` 檔案。將`spire-bundle.yaml` 檔案中	DevOps 工程師

任務	描述	所需技能
	的bundle.crt 值從私有 CA 變更為公用憑證。	

部署和註冊工作負載

任務	描述	所需技能
使用 SPIRE 註冊節點和工作負載項目。	若要向 SPIRE 伺服器註冊節點和工作負載 (服務)，請使用儲存庫中的程式碼。	DevOps 工程師
在啟用 MTL 的 App Mesh 中建立網格。	使用適用於微服務應用程式 (例如虛擬服務、虛擬路由器和虛擬節點) 的所有元件，在 App Mesh 中建立新的網格。	DevOps 工程師
檢查註冊的條目。	您可以執行下列命令來檢查節點和工作負載的已註冊項目。 `kubectl exec -n spire spire-server-0 -- /opt/spire/bin/spire-server entry show` 這將顯示 SPIRE 特工的條目。	DevOps 工程師

驗證 MTL 流量

任務	描述	所需技能
驗證 MTL 流量。	1. 從前端服務，將 HTTP 標頭傳送至後端服務，並使用在 SPIRE 中註冊的服務驗證是否成功回應。	DevOps 工程師

任務	描述	所需技能
	2. 對於相互 TLS 驗證，您可以執行下列命令來檢查ssl.handshake 統計資料。 `kubectl exec -it $POD -n $NAMESPACE -c envoy -- curl http://localhost:9901/stats | grep ssl.handshake` 執行上一個命令之後，您應該會看到偵聽程式ssl.handshake 計數，看起來類似下列範例： `listener.0.0.0.0_15000.ssl.handshake: 2`	
確認憑證是從 AWS 私有 CA 核發。	您可以檢視 SPIRE 伺服器中的記錄，以檢查外掛程式是否已正確設定，並從上游私有 CA 核發憑證。執行下列命令。 `kubectl logs spire-server-0 -n spire` 然後檢視產生的記錄檔。此代碼假定您的服務器已命名spire-server-0 並託管在您的 spire 名稱空間中。您應該會看到成功載入外掛程式，以及與上游私有 CA 建立連線。	DevOps 工程師

相關資源

- 在 Amazon EKS 上的 AWS 應用程序網格中將 MTL 與 SPIRE 一起使用
- 在多帳戶 Amazon EKS 環境中使用 SPIFE/SPIRE 在 AWS App Mesh 中啟用 MTL
- 此模式中使用的逐步解說
- 伺服器外掛程式: UpstreamAuthority 「aws_pca」
- 庫伯尼特人快速入門課程

其他資訊

將權限附加至叢集節點執行個體角色

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ACMPCASigning",
            "Effect": "Allow",
            "Action": [
                "acm-pca:DescribeCertificateAuthority",
                "acm-pca:IssueCertificate",
                "acm-pca:GetCertificate",
                "acm:ExportCertificate"
            ],
            "Resource": "*"
        }
    ]
}
AWS Managed Policy: "AWSAppMeshEnvoyAccess"
```

新增 ACM 的 SPIRE 外掛程式

```
Add the SPIRE plugin for ACM
Change certificate_authority_arn to your PCA ARN. The signing algorithm used must be
 the same as the signing algorithm on the PCA. Change your_region to the appropriate
 AWS Region.
UpstreamAuthority "aws_pca" {
    plugin_data {
        region = "your_region"
        certificate_authority_arn = "arn:aws:acm-pca:...."
```

```
        signing_algorithm = "your_signing_algorithm"
    }
}
```

使用 AWS Batch 為 Amazon RDS for PostgreSQL 資料庫執行個體自動備份

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：PoC 或試點	技術：容器和微服務; 數據庫; DevOps	工作負載：所有其他工作
AWS 服務：Amazon RDS; AWS Batch; Amazon CloudWatch; AWS Lambda; Amazon S3		

Summary

備份 PostgreSQL 資料庫是一項重要的工作，通常可以使用 pg_dump 公用程式來完成，該公用程式預設會使用 COPY 命令來建立 PostgreSQL 資料庫的結構描述和資料傾印。但是，如果您需要定期備份多個 PostgreSQL 資料庫，則此程序可能會變得重複。如果您的 PostgreSQL 資料庫託管在雲端，您也可以利用 Amazon Relational Database Service 服務 (Amazon RDS) 為 PostgreSQL 提供的自動備份功能。此模式說明如何使用 pg_dump 公用程式為 Amazon RDS for PostgreSQL 資料庫執行個體自動化定期備份。

注意：這些說明假設您使用的是 Amazon RDS。不過，您也可以針對在 Amazon RDS 外部託管的 PostgreSQL 資料庫使用此方法。若要進行備份，AWS Lambda 函數必須能夠存取您的資料庫。

以時間為基礎的 Amazon CloudWatch 活動事件會啟動 Lambda 函數，該函數會搜尋套用至 Amazon RDS 上 PostgreSQL 資料庫執行個體中繼資料的特定備份標籤。如果 PostgreSQL 資料庫執行個體具有 BKP: 自動 DDBDdump = 作用中標籤和其他必要的備份標籤，則 Lambda 函數會針對每個資料庫備份提交個別任務至 AWS B atch。

AWS Batch 處理這些任務，並將備份資料上傳到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。此模式使用 Docker 檔案和 entrypoint.sh 檔案來建立用於在 AWS Batch 任務中進行備份的 Docker 容器映像。備份程序完成後，AWS Batch 會將備份詳細資訊記錄到 Amazon DynamoDB 上的庫存表格中。作為額外的保護措施，如果 AWS Batch 中的任務失敗， CloudWatch 事件事件會啟動 Amazon 簡單通知服務 (Amazon SNS) 通知。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有受管或未受管理的運算環境。如需詳細資訊，請參閱 AWS Batch 文件中的受管和非受管運算環境。
- 已安裝並設定 AWS 命令列界面 (CLI) 第 2 版泊塢視窗映像。
- 適用於 Amazon RDS for PostgreSQL 現有資料庫執行個體。
- 現有的 S3 儲存貯體。
- 碼頭, 安裝和配置在 Linux 上, macOS, 或視窗.
- 熟悉 Lambda 中的編碼。

架構

技術, 堆

- Amazon CloudWatch 活動
- Amazon DynamoDB
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon RDS
- Amazon SNS
- Amazon S3
- AWS Batch
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- Docker

工具

- Amazon CloudWatch 活動 — CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- Amazon DynamoDB — DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種受管 AWS 容器映像登錄服務，安全、可擴展且可靠。
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon SNS — 亞馬遜簡單通知服務 (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者的訊息傳遞。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- AWS Batch — AWS Batch 可協助您在 AWS 雲端上執行批次運算工作負載。
- AWS KMS — AWS Key Management Service (AWS KMS) 是一項受管服務，可讓您輕鬆建立和控制用於加密資料的加密金鑰。
- AWS Lambda — L ambda 是一種運算服務，可協助您執行程式碼，而無需佈建或管理伺服器。
- AWS Secrets Manager — Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- Docker — Docker 可協助開發人員輕鬆打包、運送和執行任何應用程式，成為輕量、可攜式且自給自足的容器。

Amazon RDS 上的 PostgreSQL 資料庫執行個體必須有標籤套用至其中繼資料。Lambda 函數會搜尋標籤以識別應備份的資料庫執行個體，通常會使用下列標籤。

標籤	Description
BKP: 自動轉儲 = 活動	將 Amazon RDS 資料庫執行個體識別為備份的候選項。
bkp：= AutomatedBackupSecret <secret_name>	識別包含 Amazon RDS 登入登入資料的密碼管理員密碼。
BKP: 自動轉儲 3 桶 = <s3_bucket_name>	識別要將備份傳送至的 S3 儲存貯體。

BKP: 自動化 DumpFrequency	識別應備份資料庫的頻率和時間。
BKP: 自動化 DumpTime	
bkp: 文本轉儲命令 = <pgdump_command>	識別需要進行備份的資料庫。

史诗

在 DynamoDB 中建立詳細目錄表

任務	描述	所需技能
在 DynamoDB 中建立資料表。	登入 AWS 管理主控台，開啟 Amazon DynamoDB 主控台，然後建立一個表格。如需此和其他故事的說明，請參閱「相關資源」一節。	雲管理員，數據庫管理員
確認已建立資料表。	執行 `aws dynamodb describe-table --table-name <table-name> | grep TableStatus` 命令。如果表存在，該命令將返回`"TableStatus": "ACTIVE",` 結果。	雲管理員，數據庫管理員

在 AWS Batch 中為失敗的任務事件建立 SNS 主題

任務	描述	所需技能
建立 SNS 主題。	開啟 Amazon SNS 主控台，選擇「主題」，然後使用該名稱建立 SNS 主題`JobFailedAlert` 。訂閱主題的使用中電子郵件地址，並檢查您的電子郵件收件匣以確認來自 AWS	雲端管理員

任務	描述	所需技能
	Novents 的 SNS 訂閱電子郵件。	
為 AWS Batch 建立失敗的任務事件規則。	開啟 Amazon 主 CloudWatch 控台，選擇 [事件]，然後選擇 [建立規則]。選擇顯示進階選項，然後選擇編輯。若要建立可選取目標處理事件的模式，請使用 [其他資訊] 區段中的 [失敗工作事件] 程式碼取代現有的文字。此程式碼定義了在 AWS Batch 發生事件時啟動的CloudWatch `Failed`事件規則。	雲端管理員
新增事件規則目標。	在目標中，選擇新增目標，然後選擇 `JobFailedAlert` SNS 主題。設定其餘詳細資料並建立 Cloudwatch 事件規則。	雲端管理員

建立碼頭映像並將其推送至 Amazon ECR 儲存庫

任務	描述	所需技能
建立 Amazon ECR 儲存庫。	開啟 Amazon ECR 主控台，然後選擇要在其中建立儲存庫的 AWS 區域。選擇儲存庫，然後選擇 [建立儲存庫]。根據您的需求配置存放庫。	雲端管理員
撰寫 Dockerfile。	登入 Docker 並使用 [其他資訊] 區段中的 [範例碼頭檔案] 和 [範例 entrypoint.sh 檔案] 來建立 Docker 檔案。	DevOps 工程師

任務	描述	所需技能
建立泊塢視窗映像並將其推送至 Amazon ECR 儲存庫。	將 Docker 檔案建置到碼頭映像中，並將其推送至 Amazon ECR 儲存庫。如需此故事的說明，請參閱「相關資源」一節。	DevOps 工程師

建立 AWS Batch 元件

任務	描述	所需技能
建立 AWS Batch 任務定義。	開啟 AWS Batch 主控台並建立任務定義，其中包含 Amazon ECR 儲存庫的統一資源識別碼 (URI) 做為屬性`Image`。	雲端管理員
設定 AWS Batch 任務佇列。	在 AWS Batch 主控台上，選擇「Job 務佇列」，然後選擇「建立佇列」。建立將存放任務的任務佇列，直到 AWS Batch 在運算環境中的資源上執行它們為止。重要事項：請務必為 AWS Batch 撰寫邏輯，以便將備份詳細資訊記錄到 DynamoDB 詳細目錄表。	雲端管理員

建立並排程 Lambda 函數

任務	描述	所需技能
建立 Lambda 函數以搜尋標籤。	建立 Lambda 函數，以搜尋 PostgreSQL 資料庫執行個體上的標籤，並識別備份候選項目。確保您的 Lambda 函數可以識別`bkp:Autom`	DevOps 工程師

任務	描述	所需技能
	`atedDBDump = Active` 標籤和所有其他必要的標籤。重要事項：Lambda 函數也必須能夠將任務新增至 AWS Batch 任務佇列。	
建立以時間為基礎的事件 CloudWatch 事件。	開啟 Amazon CloudWatch 主控台並建立使用 cron 運算式定期執行 Lambda 函數的 CloudWatch 事件事件。重要事項：所有排程的事件都使用 UTC 時區。	雲端管理員

測試備份自動化

任務	描述	所需技能
建立 Amazon KMS 金鑰。	開啟 Amazon KMS 主控台並建立 KMS 金鑰，以用來加密存放在 AWS 秘密管理員中的 Amazon RDS 登入資料。	雲端管理員
建立 AWS 密碼管理員密碼。	開啟 AWS Secrets Manager 主控台，並將您的 Amazon RDS 資料庫登入資料存放為機密資料。	雲端管理員
將必要的標籤新增至 PostgreSQL 資料庫執行個體。	開啟 Amazon RDS 主控台，然後將標籤新增到您要自動備份的 PostgreSQL 資料庫執行個體。您可以使用「工具」區段中表格中的標籤。如果您需要從同一個 Amazon RDS 執行個體內的多個 PostgreSQL 資料庫進行	雲端管理員

任務	描述	所需技能
	備份，請使用`-d test:-d test1`作為標籤的`bkp:pgdumpcommand` 值。重要：`test`和`test1`是數據庫名稱。請確定冒號 (:) 後面沒有空格。	
驗證備份自動化。	若要驗證備份自動化，您可以叫用 Lambda 函數，或等待備份排程開始。備份程序完成後，請檢查 DynamoDB 詳細目錄表是否具有適用於 PostgreSQL 資料庫執行個體的有效備份項目。如果它們匹配，則備份自動化過程成功。	雲端管理員

相關資源

在 DynamoDB 中建立詳細目錄表

- 建立 Amazon DynamoDB 料表

在 AWS Batch 中為失敗的任務事件建立 SNS 主題

- 創建一個 Amazon SNS 主題
- 針對 AWS Batch 中失敗的任務事件傳送 SNS 警示

建立碼頭映像並將其推送至 Amazon ECR 儲存庫

- 創建一個 Amazon ECR 存儲庫
- 編寫一個碼頭文件，創建一個碼頭圖像，並將其推送到 Amazon ECR

建立 AWS Batch 元件

- 建立 AWS Batch 任務定義
- 設定運算環境和 AWS Batch 任務佇列
- 在 AWS Batch 中建立任務佇列

創建一個 Lambda 函數

- 建立 Lambda 函數並撰寫程式碼
- 搭配使用 Lambda 搭配使用

創建一個 CloudWatch 事件事件

- 建立以時間為基礎的 CloudWatch 事件
- 在事件中使用 Cron 運算式

測試備份自動化

- 建立 Amazon KMS 金鑰
- 建立密碼管理員密碼
- 將標籤新增至 Amazon RDS 執行個體

其他資訊

失敗的工作事件：

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
```

```
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

示例碼頭文件：

```
FROM alpine:latest
RUN apk --update add py-pip postgresql-client jq bash && \
pip install awscli && \
rm -rf /var/cache/apk/*
ADD entrypoint.sh /usr/bin/
RUN chmod +x /usr/bin/entrypoint.sh
ENTRYPOINT ["entrypoint.sh"]
```

entrypoint.sh 檔案範例：

```
 #!/bin/bash
set -e
DATETIME=`date +"%Y-%m-%d_%H_%M"`
FILENAME=RDS_PostGres_dump_${RDS_INSTANCE_NAME}
FILE=${FILENAME}_${DATETIME}

aws configure --profile new-profile set role_arn arn:aws:iam::${TargetAccountId}:role/
${TargetAccountRoleName}
aws configure --profile new-profile set credential_source EcsContainer

echo "Central Account access provider IAM role is: "
aws sts get-caller-identity

echo "Target Customer Account access provider IAM role is: "
aws sts get-caller-identity --profile new-profile

securestring=$(aws secretsmanager get-secret-value --secret-id $SECRETID --output json
 --query 'SecretString' --region=$REGION --profile new-profile)

if [[ ${securestring} ]]; then
    echo "successfully accessed secrets manager and got the credentials"
```

```
    export PGPASSWORD=$(echo $securestring | jq --raw-output | jq -r '.DB_PASSWORD')
    PGSQL_USER=$(echo $securestring | jq --raw-output | jq -r '.DB_USERNAME')
    echo "Executing pg_dump for the PostGres endpoint ${PGSQL_HOST}"
    # pg_dump -h $PGSQL_HOST -U $PGSQL_USER -n dms_sample | gzip -9 -c  | aws s3 cp -
 --region=$REGION  --profile new-profile s3://$BUCKET/$FILE
    # in="-n public:-n private"
    IFS=':' list=($EXECUTE_COMMAND);
    for command in "${list[@]}";
      do
        echo $command;
        pg_dump -h $PGSQL_HOST -U $PGSQL_USER ${command} | gzip -9 -c  | aws s3 cp - --
region=$REGION --profile new-profile s3://${BUCKET}/${FILE}-${command}".sql.gz"
        echo $?;
        if  [[ $? -ne 0 ]]; then
            echo "Error occurred in database backup process. Exiting now....."
            exit 1
        else
            echo "Postgresql dump was successfully taken for the RDS endpoint
 ${PGSQL_HOST} and is uploaded to the following S3 location s3://${BUCKET}/${FILE}-
${command}.sql.gz"
            #write the details into the inventory table in central account
            echo "Writing to DynamoDB inventory table"
            aws dynamodb put-item --table-name ${RDS_POSTGRES_DUMP_INVENTORY_TABLE} --
region=$REGION --item '{ "accountId": { "S": "'"${TargetAccountId}"'" }, "dumpFileUrl":
 {"S": "'"s3://${BUCKET}/${FILE}-${command}.sql.gz"'" }, "DumpAvailableTime": {"S":
 "'"`date +"%Y-%m-%d::%H::%M::%S"` UTC"'"}}'
            echo $?
            if  [[ $? -ne 0 ]]; then
                echo "Error occurred while putting item to DynamoDb Inventory Table.
 Exiting now....."
                exit 1
            else
                echo "Successfully written to DynamoDb Inventory Table
 ${RDS_POSTGRES_DUMP_INVENTORY_TABLE}"
            fi
        fi
      done;
else
    echo "Something went wrong {$?}"
    exit 1
fi

exec "$@"
```

使用 CI/CD 管道，在 Amazon EKS 中自動部署節點終止處理程式

由桑迪普甘加帕迪伊 (AWS)、約翰·瓦爾加斯 (AWS)、實務迪普·辛格 (AWS)、桑迪普·加萬德 (AWS) 和維約瑪薩克德瓦 (AWS) 所建立

程式碼儲存庫：將 NTH 部署至 EKS	環境：生產	技術：容器和微服務；DevOps
AWS 服務：AWS CodePipeline；Amazon EKS；AWS CodeBuild		

Summary

在 Amazon Web Services (AWS) 雲端上，您可以使用 AWS 節點終止處理常式 (一種開放原始碼專案)，以適當的方式處理 Kubernetes 內的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體關閉。AWS 節點終止處理常式可協助確保 Kubernetes 控制平面能適當回應可能導致 EC2 執行個體無法使用的事件。此類事件包括以下內容：

- EC2 執行個體排程維護
- Amazon EC2 競價型執行個體中斷
- Auto Scaling 群組縮放
- 跨可用區域 Auto Scaling 群組重新平衡
- 透過 API 或 AWS 管理主控台終止 EC2 執行個體

如果未處理事件，您的應用程式程式碼可能無法正常停止。還可能需要更長的時間才能恢復完整的可用性，或者可能會意外地將工作排程到正在關閉的節點。`aws-node-termination-handler`(NTH) 可以在兩種不同的模式下運作：執行個體中繼資料服務 (IMDS) 或佇列處理器。如需有關這兩種模式的詳細資訊，請參閱讀我檔案。

此模式會透過持續整合和持續傳遞 (CI/CD) 管線使用佇列處理器來自動部署 NTH。

注意：如果您使用的是 EKS 受管節點群組，則不需要. `aws-node-termination-handler`

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 支援與 AWS 管理主控台搭配使用的網頁瀏覽器。請參閱支援的瀏覽器清單。
- 已安裝 AWS Cloud Development Kit (AWS CDK)。
- `kubectl`，已安裝 Kubernetes 命令列工具。
- `eksctl`，已安裝適用於 Amazon 彈性 Kubernetes 服務 (亞馬遜 EKS) 的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 具有 1.20 版或更新版本的執行中 EKS 叢集。
- 連接至 EKS 叢集的自我管理節點群組。若要使用自我管理的節點群組建立 Amazon EKS 叢集，請執行下列命令。

```
eksctl create cluster --managed=false --region <region> --name <cluster_name>
```

  如需有關的詳細資訊`eksctl`，請參閱 eksctl 文件。
- 適用於您叢集的 AWS Identity and Access Management (IAM) OpenID Connect (OIDC) 供應商。如需詳細資訊，請參閱為叢集建立 IAM OIDC 提供者。

限制

- 您必須使用支援 Amazon EKS 服務的 AWS 區域。

產品版本

- 庫伯尼特斯版本 1.20 或更新版本
- `eksctl`版本 0.107.0 或更新版本
- AWS CDK 版本 2.27.0 或更新版本

架構

目標技術堆疊

- 虛擬私有雲 (VPC)

- 一個 EKS 叢集
- Amazon Simple Queue Service (Amazon SQS)
- IAM
- Kubernetes

目標架構

下圖顯示節點終止啟動時 end-to-end 步驟的高階檢視。

圖表中顯示的工作流程包含下列高階步驟：

1. 自動調整規模 EC2 執行個體終止事件會傳送至 SQS 佇列。
2. 第 N 個網繭會監控 SQS 佇列中是否有新訊息。
3. 第 N 個網繭會收到新訊息，並執行下列動作：
   - 接上節點，以便新的網繭不會在節點上執行。
   - 排空節點，以便撤除現有的網繭
   - 將生命週期掛接訊號傳送至「Auto Scaling」群組，以便終止節點。

自動化和規模

- 程式碼由 AWS CDK 管理和部署，並由 AWS CloudFormation 巢狀堆疊提供支援。
- Amazon EKS 控制平面可跨多個可用區域執行，以確保高可用性。
- 對於自動擴展，Amazon EKS 支援 Kubernetes 叢集自動配置器和卡彭特器。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。

- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- Amazon EC2 Auto Scaling 可協助您維持應用程式的可用性，並允許您根據定義的條件自動新增或移除 Amazon EC2 執行個體。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。

其他工具

- kubectl 是一種 Kubernetes 命令列工具，可針對 Kubernetes 叢集執行命令。您可以使用 kubectl 部署應用程式、檢查和管理叢集資源，以及檢視記錄。

Code

此模式的代碼可在 GitHub .com 上的deploy-nth-to-eks回購中獲得。代碼存儲庫包含以下文件和文件夾。

- `nth folder`— Helm 圖表、值檔案和用於掃描和部署節點終止處理常式 AWS CloudFormation 範本的指令碼。
- `config/config.json`— 應用程式的組態參數檔案。此檔案包含要部署的 CDK 所需的所有參數。
- `cdk`— AWS CDK 原始程式碼。
- `setup.sh`— 用於部署 AWS CDK 應用程式以建立所需 CI/CD 管道和其他必要資源的指令碼。
- `uninstall.sh`— 用於清理資源的指令碼。

要使用示例代碼，請按照 Epics 部分中的說明進行操作。

最佳實務

如需自動化 AWS 節點終止處理常式時的最佳實務，請參閱下列內容：

- EKS 最佳做法指南
- 節點終止處理程序-配置

史诗

設定您的環境

任務	描述	所需技能
克隆回購。	若要使用 SSH (安全殼層) 複製存放庫，請執行下列命令。 `git clone git@github.com:aws-samples/deploy-nth-to-eks.git` 若要使用 HTTPS 複製存放庫，請執行下列命令。 `git clone https://github.com/aws-samples/deploy-nth-to-eks.git` 克隆回購會創建一個名為`deploy-nth-to-eks` 。 切換到該目錄。 `cd deploy-nth-to-eks`	應用開發人員、AWS DevOps、 DevOps 工程師
設定庫貝設定檔案。	在終端機中設定 AWS 登入資料，並確認您有權擔任叢集角色。您可以使用下列範例程式碼。 `aws eks update-kubeconfig --name <Cluster_Name> --region <region>--role-arn <Role_ARN>`	AWS DevOps、 DevOps 工程師、應用程式開發者

部署 CI/CD 管線

任務	描述	所需技能
設定參數。	在config/config.json 檔案中，設定下列必要參數。 • pipelineName ：要由 AWS CDK 建立的 CI/CD 管線名稱 (例如，)。deploy-nth-to-eks-pipeline AWS CodePipeline 將建立具有此名稱的管道。 • repositoryName ：要建立的 AWS CodeCommit 存放庫 (例如，deploy-nth-to-eks-repo)。AWS CDK 會建立此存放庫，並將其設定為 CI/CD 管道的來源。 注意：此解決方案將創建此 CodeCommit 回購和分支（在以下 branch 參數中提供）。 • branch：存儲庫中的分支名稱（例如，main）。對此分支的提交將啟動 CI/CD 管線。 • cfn_scan_script ：用來掃描 AWS CloudFormation 範本是否有 NTH (scan.sh) 的指令碼路徑。此指令碼存在於將成為 AWS 存 CodeCommit 放庫一部分的nth資料夾中。	應用開發人員、AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	• cfn_deploy_scrip t：用於部署 NTH 之 AWS CloudFormation 範本的指令碼路徑 (installAp p.sh)。 • stackName ：要部署的 CloudFormation 堆疊名稱。 • eksClusterName ：現有 EKS 叢集的名稱。 • eksClusterRole ：將用來存取所有 Kubernetes API 呼叫之 EKS 叢集的 IAM 角色 (例如，)。clusterad min 通常會在中新增此角色aws-authConfigMap 。 • create_cluster_rol e ：若要建立 eksCluste rRole IAM 角色，請輸入 yes。如果您要在eksClusterRole 參數中提供現有的叢集角色，請輸入 no。 • create_iam_oidc_pr ovider ：若要為您的叢集建立 IAM OIDC 提供者，請輸入 yes。如果 IAM OIDC 提供者已存在，請輸入否。如需詳細資訊，請參閱為叢集建立 IAM OIDC 提供者。 • AsgGroupName ：屬於 EKS 叢集一部分的「Auto Scaling	

任務	描述	所需技能
	例」群組名稱的逗號分隔清單 (例如)。ASG_Group_1,ASG_Group_2 • region：叢集所在的 AWS 區域名稱 (例如，us-east-2)。 • install_cdk ：如果機器上目前未安裝 AWS CDK，請輸入 yes。執行命cdk --version 令以檢查已安裝的 AWS CDK 版本是否為 2.27.0 或更新版本。在這種情況下，請輸入 no。 如果您輸入 yes，setup.sh 指令碼會執行命sudo npm install -g cdk@2.27.0 令，在機器上安裝 AWS CDK。該腳本需要 sudo 權限，因此請在出現提示時提供帳戶密碼。	

任務	描述	所需技能
建立要部署第 N 個的 CI/CD 管線。	執行 setup.sh 指令碼。 `./setup.sh` 該指令碼將部署 AWS CDK 應用程式，該應用程式將根據檔案中的使用者輸入參數，使用範例程式碼、管道和 CodeBuild 專`config/config.json` 案來建立 CodeCommit 存放庫。 這個腳本會要求輸入密碼，因為它使用 sudo 命令安裝 npm 軟件包。	應用開發人員、AWS DevOps、 DevOps 工程師

任務	描述	所需技能
檢閱 CI/CD 管線。	開啟 AWS 管理主控台，並檢閱以下在堆疊中建立的資源。 • CodeCommit 回購與文nth件夾的內容 • AWS CodeBuild 專案cfn-scan，將掃描 CloudFormation 範本中是否有漏洞。 • CodeBuild 專案Nth-Deploy ，該專案將透過 AWS CodePipeline 管道部署 AWS CloudFormation 範本和對應的 NTH Helm 圖表。 • 部署 NTH 的 CodePipeline 管道。 管線成功執行之後，Helm 發行版aws-node-termination-handler 本會安裝在 EKS 叢集中。此外，名為的 Pod aws-node-termination-handler 正在叢集的kube-system 命名空間中執行。	應用開發人員、AWS DevOps、 DevOps 工程師

測試第 N 個部署

任務	描述	所需技能
模擬「Auto Scaling」群組縮放事件。	若要模擬自動縮放縮放事件，請執行下列操作：	

任務	描述	所需技能
	1. 在 AWS 主控台上，開啟 EC2 主控台，然後選擇 Auto Scaling 群組。 2. 選取與中提供的名稱相同的「Auto Scaling」群組`config/config.json`，然後選擇「編輯」。 3. 將所需容量和最小容量減少 1。 4. 選擇更新。	
檢閱記錄檔。	在擴充事件期間，NTH Pod 將會警戒並清空對應的工作節點 (將在擴充事件中終止的 EC2 執行個體)。若要檢查記錄，請使用 [其他資訊] 區段中的程式碼。	應用開發人員、AWS DevOps、 DevOps 工程師

清除

任務	描述	所需技能
清理所有 AWS 資源。	若要清除此模式所建立的資源，請執行下列命令。 `./uninstall.sh` 這將通過刪除 CloudFormation 堆棧來清理在此模式中創建的所有資源。	DevOps 工程師

故障診斷

問題	解決方案
npm 登錄未正確設定。	在安裝此解決方案期間，指令碼會安裝 npm install 以下載所有必要的套件。如果在安裝期間看到「找不到模組」的訊息，則可能未正確設定 npm 登錄。若要查看目前的登錄設定，請執行下列命令。 `npm config get registry` 若要使用設定登錄`https://registry.npmjs.org/`，請執行下列命令。 `npm config set registry https://registry.npmjs.org`
延遲 SQS 訊息傳遞。	做為疑難排解的一部分，如果您想要延遲 SQS 郵件傳遞至 NTH Pod，您可以調整 SQS 傳遞延遲參數。如需詳細資訊，請參閱 Amazon SQS 延遲佇列。

相關資源

- AWS 節點終止處理常式原始碼
- EC2 工作坊
- AWS CodePipeline
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Cloud Development Kit
- AWS CloudFormation

其他資訊

1. 找到第 N 個網繭名稱。

```
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7   1/1     Running   0          26m
kubectl get pods -n kube-system |grep aws-node-termination-handler
aws-node-termination-handler-65445555-kbqc7   1/1     Running   0          26m
```

2. 檢查日誌。範例記錄檔如下所示。這表明節點在發送 Auto Scaling 組生命週期掛鉤完成信號之前已被封鎖和排空。

```
kubectl -n kube-system logs aws-node-termination-handler-65445555-kbqc7
022/07/17 20:20:43 INF Adding new event to the event store
 event={"AutoScalingGroupName":"eksctl-my-cluster-target-nodegroup-
ng-10d99c89-NodeGroup-ZME36IGAP7O1","Description":"ASG Lifecycle Termination
 event received. Instance will be interrupted at 2022-07-17 20:20:42.702
 +0000 UTC \n","EndTime":"0001-01-01T00:00:00Z","EventID":"asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564","InProgress":fal
east-2.compute.internal","NodeProcessed":false,"Pods":null,"ProviderID":"aws:///us-
east-2c/i-0409f2a9d3085b80e","StartTime":"2022-07-17T20:20:42.702Z","State":""}
2022/07/17 20:20:44 INF Requesting instance drain event-id=asg-lifecycle-
term-33383831316538382d353564362d343332362d613931352d383430666165636334333564
 instance-id=i-0409f2a9d3085b80e kind=SQS_TERMINATE node-name=ip-192-168-75-60.us-
east-2.compute.internal provider-id=aws:///us-east-2c/i-0409f2a9d3085b80e
2022/07/17 20:20:44 INF Pods on node node_name=ip-192-168-75-60.us-
east-2.compute.internal pod_names=["aws-node-qchsw","aws-node-termination-
handler-65445555-kbqc7","kube-proxy-mz5x5"]
2022/07/17 20:20:44 INF Draining the node
2022/07/17 20:20:44 ??? WARNING: ignoring DaemonSet-managed Pods: kube-system/aws-node-
qchsw, kube-system/kube-proxy-mz5x5
2022/07/17 20:20:44 INF Node successfully cordoned and drained
 node_name=ip-192-168-75-60.us-east-2.compute.internal reason="ASG Lifecycle
 Termination event received. Instance will be interrupted at 2022-07-17 20:20:42.702
 +0000 UTC \n"
2022/07/17 20:20:44 INF Completed ASG Lifecycle Hook (NTH-K8S-TERM-HOOK) for instance
 i-0409f2a9d3085b80e
```

使用 CI/CD 管道自動建置 Java 應用程式並將其部署到 Amazon EKS

由馬赫什·拉格南（AWS），詹姆斯·拉德克（AWS）和喬姆西·帕佩亨（AWS）創建

代碼存儲庫：aws-cicd-java-eks	環境：生產	技術：容器與微服務；雲端原生； DevOps現代化
工作負載：所有其他工作	AWS 服務：AWS CloudFormation; AWS CodeCommit; AWS CodePipeline; Amazon EC2 容器註冊表; Amazon EKS	

Summary

此模式說明如何建立持續整合和持續交付 (CI/CD) 管道，以使用建議的 DevSecOps 實務自動建置 Java 應用程式並將其部署到亞馬遜網路服務 (AWS) 雲端上的亞馬遜彈性 Kubernetes 服務 (Amazon EKS) 叢集。這種模式使用一個春季啟動 Java 框架開發的問候應用程序，並使用 Apache 的 Maven。

您可以使用此模式的方法為 Java 應用程式建置程式碼、將應用程式成品封裝為 Docker 映像、安全掃描映像，以及將影像作為 Amazon EKS 上的工作負載容器上傳。如果您想要從緊密結合的單體架構遷移到微服務架構，則此模式的方法非常有用。該方法還可以幫助您監視和管理 Java 應用程序的整個生命週期，從而確保更高級別的自動化並有助於避免錯誤或錯誤。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版。如需有關這方面的詳細資訊，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI 第 2 版。
- AWS CLI 第 2 版必須使用建立 Amazon EKS 叢集的相同 IAM 角色進行設定，因為只有該角色獲得授權，才能將其他 IAM 角色新增到 `aws-authConfigMap`. 如需設定 AWS CLI 的資訊和步驟，請參閱 AWS CLI 文件中的組態基礎知識。

- 具有 AWS 完整存取權的 AWS Identity and Access Management (IAM) 角色和許可 CloudFormation。有關這方面的詳細資訊，請參閱 AWS CloudFormation 文件中的使用 IAM 控制存取。
- 現有的 Amazon EKS 叢集，其中包含 EKS 叢集中工作者節點的 IAM 角色名稱和 IAM 角色 Amazon 資源名稱 (ARN) 的詳細資訊。
- Kubernetes 叢集自動配置器 (已在您的 Amazon EKS 叢集中安裝和設定)。如需詳細資訊，請參閱 Amazon EKS 文件中的叢集自動配置器。
- 存取 GitHub 儲存庫中的程式碼。

重要備註

AWS Security Hub 已啟用為程式碼中 AWS CloudFormation 範本的一部分。根據預設，啟用 Security Hub 之後，會提供 30 天的免費試用期，之後會產生與此 AWS 服務相關的費用。如需有關定價的詳細資訊，請參閱 AWS Security Hub 定價。

產品版本

- 頭盔版本 3.4.2 或更新版本
- 阿帕奇 Maven 版本 3.6.3 或更高版本
- BridgeCrew 切科夫 2.2 版或更高版本
- 水族安全三維版本 0.37 或更高版本

架構

技術堆疊

- AWS CodeBuild
- AWS CodeCommit
- Amazon CodeGuru
- AWS CodePipeline
- Amazon Elastic Container Registry
- Amazon Elastic Kubernetes Service
- Amazon EventBridge
- AWS Security Hub
- Amazon Simple Notification Service (Amazon SNS)

目標架構

該圖顯示以下工作流程：

1. 開發人員更新 CodeCommit 存儲庫的基底分支中的 Java 應用程序代碼，該代碼創建了一個提取請求（PR）。
2. 提交 PR 後，Amazon 審核者會自動 CodeGuru 審核程式碼、根據 Java 的最佳實務進行分析，並向開發人員提供建議。
3. 將 PR 合併到基本分支之後，就會建立 Amazon EventBridge 事件。
4. 該 EventBridge 事件啟動 CodePipeline 管道，並啟動。
5. CodePipeline 執行 CodeSecurity 掃描階段（連續安全性）。
6. CodeBuild 啟動使用 Checkov 掃描 Dockerfile 和 Kubernetes 部署 Helm 檔案的安全性掃描程序，並根據增量程式碼變更掃描應用程式原始碼。應用程式原始程式碼掃描是由CodeGuru 審核者命令列介面 (CLI) 包裝函式執行。
7. 如果安全性掃描階段成功，就會啟動「建置」階段 (持續整合)。
8. 在「建置」階段中， CodeBuild 建置成品、將成品封裝至 Docker 映像、使用 Aqua Security Trivy 掃描映像中是否有安全漏洞，然後將映像儲存在 Amazon ECR 中。
9. 從步驟 8 檢測到的漏洞已上傳到 Security Hub，供開發人員或工程師進一步分析。Security Hub 提供修復弱點的概觀和建議。
10. CodePipeline 管道內各個階段的電子郵件通知會透過 Amazon SNS 傳送。
11. 持續整合階段完成後， CodePipeline 進入部署階段 (持續交付)。
12. Docker 映像會使用頭盔圖表，以容器工作負載 (網繭) 的形式部署至 Amazon EKS。
13. 應用程式網繭設定了 Amazon 效能分 CodeGuru 析工具代理程式，該代理程式會將應用程式的效能分析資料 (CPU、堆集使用量和延遲) 傳送至 Amazon CodeGuru Profiler，以協助開發人員瞭解應用程式的行為。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。

- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- Amazon CodeGuru Profiler 會從您的即時應用程式收集執行階段效能資料，並提供可協助您微調應用程式效能的建議。
- Amazon CodeGuru Reviewer 使用程式分析和機器學習來偵測開發人員難以找到的潛在缺陷，並提供改善 Java 和 Python 程式碼的建議。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。它也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他服務

- Helm 是 Kubernetes 的開源軟件包管理器。
- Apache Maven 是軟體專案管理和理解工具。
- BridgeCrew Checkov 是一種靜態代碼分析工具，用於掃描基礎結構作為代碼（IaC）文件，以查找可能導致安全性或合規性問題的錯誤配置。
- Aqua 安全 Trivy 是一個全面的掃描器，用於容器映像，文件系統和 Git 存儲庫中的漏洞，除了配置問題。

Code

此模式的代碼可在 GitHub aws-codepipeline-devsecops-amazoneks存儲庫中找到。

最佳實務

- IAM 實體在此解決方案的所有階段都遵循了最低特權原則。如果您想要使用其他 AWS 服務或第三方工具來擴充解決方案，建議您遵循最低權限原則。
- 如果您有多個 Java 應用程式，建議您為每個應用程式建立個別的 CI/CD 管線。
- 如果您有一個整體應用程序，我們建議盡可能將應用程序分解為微服務。微型服務更具彈性，可讓您更輕鬆地將應用程式部署為容器，並提供應用程式整體建置與部署的能見度。

史诗

設定環境

任務	描述	所需技能
克隆存 GitHub 儲庫。	若要複製存放庫，請執行下列命令。 `git clone https://github.com/aws-samples/aws-codepipeline-devsecops-amazoneks`	應用 DevOps 程式開發人員、
建立 S3 儲存貯體並上傳程式碼。	1. 登入 AWS 管理主控台，開啟 Amazon S3 主控台，然後在計劃部署此解決方案的 AWS 區域中建立 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。 2. 在 S3 儲存貯體中，建立名為code. 3. 導覽至您複製儲存庫的位置。若要使用 .zip 副檔名	AWS DevOps、 DevOps 工程師、雲端管理員、 DevOps

任務	描述	所需技能
	(cicdstack.zip) 建立整個程式碼的壓縮版本，並驗證 .zip 檔案，請依序執行下列命令。 注意：如果命python令失敗並指出未找到 Python，請python3改用。 `cd aws-codepipeline-devsecops-amazoneks` `python -m zipfile -c cicdstack.zip *` `python -m zipfile -t cicdstack.zip` 4. 將cicdstack.zip 檔案上傳到您先前在 S3 儲存貯體中建立的程式碼資料夾。	

任務	描述	所需技能
建立 AWS CloudFormation 堆疊。	1. 開啟 AWS 主 CloudFormation 控台，然後選擇建立堆疊。 2. 在 [指定範本] 中，選擇 [上傳範本檔案]、[上傳`cf_templates/codecommit_ecr.yaml` 檔案]，然後選擇 [下一步]。 3. 在指定堆疊詳細資料中，輸入堆疊名稱，然後提供下列輸入參數值： • CodeCommitRepositoryBranchName：您的代碼將駐留的分支名稱（默認為 main） • CodeCommitRepositoryName：要創建的 CodeCommit 存儲庫的名稱。 • CodeCommitRepositoryS3Bucket：您在其中建立程式碼資料夾的 S3 儲存貯體的名稱 • CodeCommitRepository中三 BucketObj Key：`code/cicdstack.zip` • ECR RepositoryName：要創建的 Amazon ECR 回購的名稱 4. 選擇 [下一步]，使用 [設定堆疊] 選項的預設設定，然後選擇 [下一步]。	AWS DevOps， DevOps

任務	描述	所需技能
	5. 在「複查」區段中，確認範本和堆疊詳細資料，然後選擇「建立堆疊」。然後會建立堆疊，包括 CodeCommit 和 Amazon ECR 儲存庫。 6. 請記下 Java CI/CD 管道設定所需的儲存庫 CodeCommit 和 Amazon ECR 儲存庫的名稱。	
驗證 CloudFormation 堆疊部署。	1. 在 CloudFormation 主控台的 [堆疊] 下，確認您已部署的 CloudFormation 堆疊狀態。堆疊的狀態應為「建立完成」。 2. 此外，從主控台驗證 Amazon ECR 是 CodeCommit 否已佈建且已準備就緒。	DevOps 工程師
刪除 S3 儲存貯體。	清空並刪除您先前建立的 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的刪除儲存貯體。	AWS DevOps， DevOps

配置頭盔圖

任務	描述	所需技能
設定 Java 應用程式的頭盔圖表。	1. 在複製 GitHub 存放庫的位置中，導覽至資料夾`helm_charts/aws-proserve-java-greeting` 。在此資	DevOps

任務	描述	所需技能
	料夾中，values.de v.yaml 檔案包含有關 Kubernetes 資源組態的相關資訊，您可以針對容器部署到 Amazon EKS 進行修改。提供您的 AWS 帳戶 ID、AWS 區域和 Amazon ECR 儲存庫名稱，以更新 Docker 儲存庫參數。 `image:` `  repository:` `  <account-id>.dkr.e` `cr.<region>.amazon` `aws.com/<app-ecr-r` `epo-name>` 2. Java 網繭的服務類型已設定為LoadBalancer 。 `service:` `  type: LoadBalancer` `  port: 80` `  targetPort: 8080` `  path: /hello` `  initialDelaySecond` `s: 60` `  periodSeconds: 30` 若要使用不同的服務 (例如，NodePort)，您可以變更參數。如需詳細資訊，請參閱 Kubernetes 文件。 3. 您可以將參數變更為，以啟動 Kubernetes 水平網繭自動配置器。autoscaling enabled: true	

任務	描述	所需技能
	<pre>autoscaling: enabled: true minReplicas: 1 maxReplicas: 100 targetCPUUtilizati onPercentage: 80 # targetMem oryUtilizationPerc entage: 80</pre> 您可以透過變更`values.<ENV>.yaml` 檔案 (開發、生產、UAT 或 QA 環境的位置`<ENV>`) 中的值，為 Kubernetes 工作負載啟用不同的功能。	

任務	描述	所需技能
驗證 Helm 圖表的語法錯誤。	1. 從終端機執行下列命令，確認 Helm v3 已安裝在本機工作站中。 `helm --version` 如果沒有安裝頭盔 v3，請安裝它。 2. 在終端機中，導覽至 Helm 圖表目錄 (`helm_charts/aws-proserve-java-greeting`)，然後執行下列命令。 `helm lint . -f values.dev.yaml` 這將檢查頭盔圖表的任何語法錯誤。	DevOps 工程師

設定爪哇 CI/CD 管線

任務	描述	所需技能
建立 CI/CD 管線。	1. 開啟 AWS 主 CloudFormation 控台，然後選擇「建立堆疊」。 2. 在 [指定範本] 中，選擇 [上傳範本檔案]、上傳`cf_templates/build_deployment.yaml` 範本，然後選擇 [下一步]。	AWS DevOps

任務	描述	所需技能
	3. 在指定堆疊詳細資料中，指定堆疊名稱，然後為輸入參數提供下列值： • CodeBranchName：代碼所在的 CodeCommit 回購分支名稱 • EKSClusterName：您的 EKS 叢集的名稱 (而非識別碼) `EKSCluster` • EKS CodeBuild AppName：應用程序的名稱頭盔圖（ ）`aws-proserve-java-greeting` • EKS WorkerNodeRole ARN：Amazon EKS 工作者節點 IAM 角色的 ARN • EKS WorkerNodeRoleName：指派給 Amazon EKS 工作者節點的 IAM 角色的名稱 • EcrDockerRepository: Amazon ECR 存儲庫的名稱，您的代碼的 Docker 映像將被存儲 • EmailRecipient：需要傳送組建通知的電子郵件地址 • EnvType：環境（例如，開發，測試或生產） • SourceRepoName：代碼所在的 CodeCommit 存儲庫的名稱	

任務	描述	所需技能
	4. 選擇下一步。使用 [設定堆疊選項] 中的預設設定，然後選擇 [下一步]。 5. 在「檢閱」區段中，確認 AWS CloudFormation 範本和堆疊詳細資料，然後選擇「下一步」。 6. 選擇建立堆疊。 7. 在 CloudFormation 堆疊部署期間，您在參數中提供之電子郵件地址的擁有者會收到訂閱 SNS 主題的訊息。若要訂閱 Amazon SNS，擁有者必須選擇訊息中的連結。 8. 建立堆疊之後，開啟堆疊的 [輸出] 索引標籤，然後記錄 `EksCodeBuildkubeRoleARN` 輸出金鑰的 ARN 值。稍後將需要此 IAM ARN 值，才能為 CodeBuild IAM 角色提供權限，以便在 Amazon EKS 叢集中部署工作負載。	

啟動安全中心與 Aqua 安全之間的整合

任務	描述	所需技能
開啟 Aqua 安全性整合功能。	要將 Trivy 報告的 Docker 映像漏洞發現項目上傳到 Security Hub，需要執行此步驟。由於 AWS CloudFormation 不支援	AWS 管理員、 DevOps 工程師

任務	描述	所需技能
	Security Hub 整合，因此必須手動完成此程序。 1. 開啟 AWS Security Hub 主控台，然後瀏覽至整合。 2. 搜尋「水上安全」，然後選擇「水上安全：水上安全」。 3. 選擇接受搜尋結果。	

配置 CodeBuild 以運行頭盔或 kubectl 命令

任務	描述	所需技能
CodeBuild 允許在 Amazon EKS 叢集中執行頭盔或 kubectl 命令。	若 CodeBuild 要經過驗證，才能在 EKS 叢集中使用 Helm 或*kubectl*命令，您必須將 IAM 角色新增至. *aws-auth ConfigMap* 在此情況下，請新增 IAM 角色的 ARNEksCodeBuildkubeRoleARN ，這是為 CodeBuild 服務建立的 IAM 角色，可存取 EKS 叢集並在其上部署工作負載。這是一次性的活動。 重要事項：必須先完成下列程序，才能進行中的部署核准階段 CodePipeline。 1. 在您的 Amazon Linux 或 macOS 環境中打開`cf_templates/kube_aws_auth_configmap`	DevOps

任務	描述	所需技能
	_patch.sh 外圍程序腳本。 2. 透過執行下列命令向 Amazon EKS 叢集進行驗證。 `aws eks --region <aws-region> update-kubeconfig --name <eks-cluster-name>` 3. 使用下列命令執行 shell 指令碼，並<rolearn-eks-codebuild-kubectl> 以您先前記錄EksCodeBuildkubeRoleARN 的 ARN 值取代。 `bash cf_templates/kube_aws_auth_configmap_patch.sh <rolearn-eks-codebuild-kubectl>` aws_authConfigMap 已設定，並授與存取權。	

驗證 CI/CD 管線

任務	描述	所需技能
確認 CI/CD 管線是否自動啟動。	1. 如果 Checkov 偵測到 Dockerfile 或 Helm 圖表中的漏洞，管道中的 CodeSecurity 掃描階段通	DevOps

任務	描述	所需技能
	常會失敗。不過，這個範例的目的在於建立識別潛在安全性弱點的程序，而不是透過 CI/CD 管道 (通常是處理序) 修正。 DevSecOps 在檔案中buildspec/buildspec_secscan.yaml ，checkov命令會使用--soft-fail 旗標來避免管線失敗。 `- echo -e "\n Running Dockerfile Scan"` `- checkov -f code/app/Dockerfile --framework dockerfile --soft-fail --summary-position bottom` `- echo -e "\n Running Scan of Helm Chart files"` `- cp -pv helm_charts/$EKS_CODEBUILD_APP_NAME/values.dev.yaml helm_charts/$EKS_CODEBUILD_APP_NAME/values.yaml` `- checkov -d helm_charts/$EKS_CODEBUILD_APP_NAME --framework helm --soft-fail --summary-position bottom` `- rm -rfv helm_charts/$EKS_C`	

任務	描述	所需技能
	ODEBUILD_APP_NAME/values.yaml 若要讓管線在報告 Dockerfile 和 Helm 圖表的弱點時失敗，必須從命令中移除該--soft-fail checkov選項。然後，開發人員或工程師可以修復這些漏洞，並將更改提交到 CodeCommit 源代碼存儲庫。 2. 與 CodeSecurity 掃描類似，構建階段使用 Aqua 安全 Trivy 在推送應用程序之前識別高和關鍵 Docker 圖像漏洞。到 Amazon ECR。在此示例中，我們不會因 Docker 映像漏洞導致管道失敗。在檔案中buildspec/buildspec.yml ，該trivy命令包含**--exit-code** 帶有值的旗標 **0**，這就是為什麼在報告 HIGH 或 CRITIAL Docker 映像弱點時管道不會失敗的原因。 - AWS_REGION=$AWS_DEFAULT_REGION AWS_ACCOUNT_ID=$AWS_ACCOUNT_ID trivy -d image --no-progress --ignore-unfixed --exit-code 0 --severity HIGH,CRITICAL --format template --template "@securityhub/asff.tpl" -o	

任務	描述	所需技能
	securityhub/report .asff $AWS_ACCO UNT_ID.dkr.ecr.$AW S_DEFAULT_REGION.a mazonaws.com/$IMAG E_REPO_NAME:$CODEB UILD_RESOLVED_SOUR CE_VERSION 若要讓管道在報告HIGH, CRTICAL弱點時失敗，請--exit-code 將的值變更為1。 然後，開發人員或工程師可以修復這些漏洞，並將更改提交到 CodeCommit 源代碼存儲庫。 3. Aqua 安全 Trivy 回報的 Docker 映像漏洞已上傳到 Security Hub。在 AWS Security Hub 主控台上，導覽至發現項目。使用「記錄狀態」=「活動」和「產品 = Aqua 安全性」過濾發現項 這將列出 Security Hub 中的 Docker 映像漏洞。Security Hub 心可能需要 15 分鐘 — 1 小時才會出現弱點。 有關使用啟動管道的詳細資訊 CodePipeline，請參閱 AWS CodePipeline 文件中 CodePipeline的啟動管道、手	

任務	描述	所需技能
	動啟動管道和按排程啟動管道。	
核准部署。	1. 建置階段完成之後，就會有部署核准閘門。審核者或發行管理員應該檢查組建，如果滿足所有需求，則核准它。對於使用持續交付進行應用程式部署的團隊，建議使用這種方法。 2. 核准後，管線會啟動「部署」階段。 3. 部署階段成功之後，此階段的 CodeBuild 記錄會提供應用程式的 URL。使用 URL 來驗證應用程式的準備程度。	DevOps
驗證應用程式分析。	部署完成並將應用程式網繭部署到 Amazon EKS 後，應用程式中設定的 Amazon 效能分 CodeGuru 析工具代理程式會嘗試將應用程式的效能分析資料 (CPU、堆積摘要、延遲和瓶頸) 傳送至 Amazon 效能分析工具。 CodeGuru 對於應用程式的初始部署，Amazon 效能分 CodeGuru 析工具需要大約 15 分鐘的時間來視覺化分析資料。	AWS DevOps

相關資源

- AWS CodePipeline 文件
- 在 AWS 中使用 Trivy 掃描影像 CodePipeline (部落格文章)
- 使用 Amazon CodeGuru 效能分析工具改善您的 Java 應用程式 (部落格文章)
- AWS 安全性尋找格式 (ASFF) 語法
- Amazon EventBridge 事件模式
- 頭盔升級

其他資訊

CodeGuru 就功能而言，效能分析工具不應與 AWS X-Ray 服務混淆。 CodeGuru Profiler 最適合識別最昂貴的代碼行，這可能會導致瓶頸或安全問題，並在它們成為潛在風險之前加以修復。AWS X-Ray 服務適用於應用程式效能監控。

在此模式中，事件規則與預設事件匯流排相關聯。如果需要，您可以擴充模式以使用自訂事件匯流排。

此病毒碼會使用 CodeGuru Reviewer 做為應用程式程式碼的靜態應用程式安全性測試 (SAST) 工具。您也可以將此管線用於其他工具，例如 SonarQube 或 Checkmarx。您可以在中新增任何這些工具的對應掃描設定指示`buildspec/buildspec_secscan.yaml`，以取代的掃描指示 CodeGuru。

使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統

創建者杜爾加普拉薩德奇普里（AWS）

環境：PoC 或試點	技術：容器與微服務；雲端原生；管理與治理；儲存與備份；Web 和行動應用程式	AWS 服務：Amazon ECS; Amazon EFS

Summary

此模式提供程式碼範例和步驟來建立 Amazon Elastic Container Service (Amazon ECS) 任務定義，該定義可在亞馬遜網路服務 (AWS) 雲端的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上執行，同時使用 Amazon Elastic File System (Amazon EFS) 在這些 EC2 執行個體上掛載檔案系統。使用 Amazon EFS 的 Amazon ECS 任務會自動掛接您在任務定義中指定的檔案系統，並讓這些檔案系統可供 AWS 區域中所有可用區域的任務容器使用。

為了滿足您的持續性儲存和共用儲存需求，您可以同時使用 Amazon ECS 和 Amazon EFS。例如，您可以使用 Amazon EFS 存放應用程式的永久性使用者資料和應用程式資料，並在不同的可用區域中執行作用中和待命 ECS 容器配對，以實現高可用性。您也可以使用 Amazon EFS 存放可由 ECS 容器和分散式任務工作負載 parallel 存取的共用資料。

若要將 Amazon EFS 與 Amazon ECS 搭配使用，您可以在任務定義中新增一或多個磁碟區定義。磁碟區定義包括 Amazon EFS 檔案系統 ID、存取點 ID，以及 AWS Identity and Access Management (IAM) 授權或傳輸中傳輸層安全 (TLS) 加密的組態。您可以使用工作定義中的容器定義來指定在容器執行時裝載的工作定義磁碟區。執行使用 Amazon EFS 檔案系統的任務時，Amazon ECS 會確保檔案系統已掛載，並可供需要存取檔案系統的容器使用。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有虛擬私人網路 (VPN) 端點或路由器的虛擬私有雲 (VPC)

- (建議使用) Amazon ECS 容器代理程式 1.38.0 或更新版本，以便與 Amazon EFS 存取點和 IAM 授權功能相容 (如需詳細資訊，請參閱 AWS 部落格文章適用於 Amazon EFS 的新功能 — IAM 授權和存取點)。

限制

- 舊於 1.35.0 的 Amazon ECS 容器代理程式版本不支援 Amazon EFS 檔案系統來執行使用 EC2 啟動類型的任務。

架構

下圖顯示使用 Amazon ECS 建立任務定義並在 ECS 容器中的 EC2 執行個體上掛載 Amazon EFS 檔案系統的應用程式範例。

該圖顯示以下工作流程：

1. 建立一個 Amazon EFS 檔案系統。
2. 使用容器建立工作定義。
3. 設定容器執行個體以掛接 Amazon EFS 檔案系統。任務定義參考磁碟區掛載，因此容器執行個體可以使用 Amazon EFS 檔案系統。無論在哪個容器執行個體上建立這些任務，ECS 任務都可以存取相同的 Amazon EFS 檔案系統。
4. 使用任務定義的三個執行個體建立 Amazon ECS 服務。

技術, 堆棧

- Amazon EC2
- Amazon ECS
- Amazon EFS

工具

- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。

- Amazon ECS — 亞馬遜彈性容器服務 (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，用於在叢集上執行、停止和管理容器。您可以在 AWS Fargate 管理的無伺服器基礎設施上執行任務和服務。或者，若要進一步控制基礎設施，您可以在您管理的 EC2 執行個體叢集上執行任務和服務。
- Amazon EFS — Amazon Elastic File System (Amazon EFS) 提供簡單、可擴展且全受管的彈性 NFS 檔案系統，可與 AWS 雲端服務和現場部署資源搭配使用。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可透過命令列殼層中的命令與 AWS 服務互動。只要使用最少的組態，您就可以執行 AWS CLI 命令，從命令提示字元實作與以瀏覽器為基礎的 AWS 管理主控台所提供的功能相同。

史诗

建立 Amazon EFS 檔案系統

任務	描述	所需技能
使用 AWS 管理主控台建立 Amazon EFS 檔案系統。	1. 建立 Amazon EFS 檔案系統，然後選擇包含您容器的 VPC。注意：如果您使用不同的 VPC，請設定 VPC 對等連線。 2. 請注意檔案系統 ID。	AWS DevOps

使用 Amazon EFS 檔案系統或 AWS CLI 建立 Amazon ECS 任務定義

任務	描述	所需技能
使用 Amazon EFS 檔案系統建立任務定義。	使用新的 Amazon ECS 主控台或具有下列組態的傳統 Amazon ECS 主控台來建立任務定義： • 如果您使用新的主控台，請針對應用程式環境選擇 Amazon EC2 執行個體。如果您使用傳統主控台，請選擇 EC2 做為啟動類型。	AWS DevOps

任務	描述	所需技能
	• 新增磁碟區。輸入磁碟區的名稱，選擇 EFS 做為磁碟區類型，然後選擇您先前記下的檔案系統 ID。對於根目錄，請選擇您要在 Amazon ECS 容器主機上託管的 Amazon EFS 檔案系統路徑。	
使用 AWS CLI 建立任務定義。	1. 若要為您的工作定義建立含有輸入參數預留位置的 JSON 範本，請執行下列命令： `aws ecs register-task-definition --generate-cli-skeleton` 2. 若要使用 JSON 範本建立工作定義，請執行下列命令： `aws ecs register-task-definition --cli-input-json file://<path_to_your_json_file>` 3. 根據檔案 (附`task_definition_parameters.json` 件) 在 JSON 範本中輸入輸入參數。附註：如需有關輸入參數的詳細資訊，請參閱任務定義參數 (Amazon ECS 文件) 和 register-task-definition(AWS CLI 命令參考)。	AWS DevOps

相關資源

- Amazon ECS 任務定義
- Amazon EFS 卷

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Fargate 在 Amazon ECS 上部署 Java 微服務

由維傑湯普森（AWS）和桑迪普邦杜古拉（AWS）創建

環境：PoC 或試點	來源：容器	目標：Amazon ECS
R 類型：不適用	技術：容器與微服務；Web 與行動應用程式	AWS 服務：Amazon ECS

Summary

此模式提供使用 AWS Fargate 在亞馬遜彈性容器服務 (Amazon ECS) 上部署容器化 Java 微型服務的指導。該模式不使用 Amazon Elastic Container Registry (Amazon ECR) 進行容器管理; 相反，Docker 映像從碼頭集線器提取。

先決條件和限制

先決條件

- 碼頭集線器上現有的 Java 微服務應用程式
- 一個公共碼頭存儲庫
- 有效的 AWS 帳戶
- 熟悉 AWS 服務，包括 Amazon ECS 和 Fargate
- 碼頭工人，Java 和春季啟動框架
- Amazon Relational Database Service（Amazon RDS）啟動並運行（可選）
- 虛擬私有雲 (VPC) (如果應用程式需要 Amazon RDS) (選用)

架構

源, 技術, 堆棧

- Java 微服務（例如，在春季啟動中實現）並部署在碼頭上

來源架構

目標技術堆疊

- 使用 Fargate 託管每個微服務的 Amazon ECS 叢集
- 用於託管 Amazon ECS 叢集和相關安全群組的 VPC 擬私人雲端網路
- 每個微服務的群集/任務定義，使用 Fargate 啟動容器

目標架構

工具

工具

- Amazon ECS 無需安裝和操作自己的容器協調軟體、管理和擴展虛擬機器叢集，或在這些虛擬機器上排程容器。
- AWS Fargate 可協助您執行容器，而不需要管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。它與 Amazon Elastic Container Service (Amazon ECS) 一起使用。
- Docker 是一個軟件平台，可讓您快速構建，測試和部署應用程序。Docker 將軟體封裝到稱為容器的標準化單元中，這些單元包含軟體所需的一切，包括程式庫、系統工具、程式碼和執行階段。

泊塢視窗代碼

下列 Docker 檔案會指定所使用的 Java 開發套件 (JDK) 版本、Java 封存檔 (JAR) 檔案所在的位置、公開的連接埠號碼，以及應用程式的進入點。

```
FROM openjdk:11
ADD target/Spring-docker.jar Spring-docker.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","Spring-docker.jar"]
```

史诗

建立新的工作定義

任務	描述	所需技能
建立任務定義。	在 Amazon ECS 中執行泊塢視窗容器需要任務定義。在 https://console.aws.amazon.com/ecs/ 開啟 Amazon ECS 主控台，選擇「任務定義」，然後建立新的任務定義。如需詳細資訊，請參閱 Amazon ECS 文件。	AWS 系統管理員、應用程式開發
選擇啟動類型。	選擇 Fargate 作為啟動類型。	AWS 系統管理員、應用程式開發
設定工作。	定義作業名稱，並使用適當數量的工作記憶體和 CPU 來設定應用程式。	AWS 系統管理員、應用程式開發
定義容器。	指定容器名稱。對於映像檔，請輸入 Docker 網站名稱、儲存庫名稱和 Docker 映像檔的標籤名稱 () `docker.io/sample-repo/sample-application:sample-tag-name` 。設定應用程式的記憶體限制，並為允許的連接埠設定連接埠映射 (`8080, 80`)。	AWS 系統管理員、應用程式開發
建立工作。	當工作和容器設定就位時，請建立工作。如需詳細指示，請參閱「相關資源」一節中的連結。	AWS 系統管理員、應用程式開發

配置叢集

任務	描述	所需技能
建立和配置叢集。	選擇 [僅限網路連線] 做為叢集類型、設定名稱，然後建立叢集或使用現有叢集 (如果有的話)。如需詳細資訊，請參閱 Amazon ECS 文件。	AWS 系統管理員、應用程式開發

設定工作

任務	描述	所需技能
建立任務。	在叢集內，選擇 [執行新工作]。	AWS 系統管理員、應用程式開發
選擇啟動類型。	選擇 Fargate 作為啟動類型。	AWS 系統管理員、應用程式開發
選擇作業定義、修訂版和平台版本。	選擇您要執行的作業、作業定義的修訂版本，以及平台版本。	AWS 系統管理員、應用程式開發
選取 叢集。	選擇您要從中執行工作的叢集。	AWS 系統管理員、應用程式開發
指定工作數目。	設定應執行的工作數目。如果您要啟動兩個或兩個以上的工作，則需要負載平衡器才能在工作之間分配流量。	AWS 系統管理員、應用程式開發
指定任務群組。	(選擇性) 指定任務群組名稱，以將一組相關工作識別為任務群組。	AWS 系統管理員、應用程式開發
設定叢集 VPC、子網路和安全群組。	設定叢集 VPC 和您要在其上部署應用程式的子網路。建立或	AWS 系統管理員、應用程式開發

任務	描述	所需技能
	更新安全群組 (HTTP、HTTPS 和通訊埠 8080)，以提供對輸入和輸出連線的存取權。	
設定公用 IP 設定。	啟用或停用公用 IP，視您是否要使用公用 IP 位址執行 Fargate 工作而定。預設的建議選項為 [已啟用]。	AWS 系統管理員、應用程式開發
檢閱設定並建立工作	檢閱您的設定，然後選擇 [執行工作]。	AWS 系統管理員、應用程式開發

切過

任務	描述	所需技能
複製應用程式 URL。	當工作狀態已更新為「執行中」時，請選取工作。在 [網路] 區段中，複製公用 IP。	AWS 系統管理員、應用程式開發
測試您的應用程式。	在瀏覽器中，輸入公用 IP 以測試應用程式。	AWS 系統管理員、應用程式開發

相關資源

- Amazon ECS 碼頭基礎知識（Amazon ECS 文檔）
- AWS Fargate 上的 Amazon ECS（Amazon ECS 文檔）
- 建立任務定義 (Amazon ECS 文件)
- 建立叢集 (Amazon ECS 文件)
- 設定基本服務參數 (Amazon ECS 文件)
- 設定網路 (Amazon ECS 文件)
- 在 Amazon ECS 上部署 Java 微服務 (部落格文章)

使用 Amazon ECR 和 AWS Fargate 在 Amazon ECS 上部署 Java 微服務

由維傑湯普森（AWS）和桑迪普邦杜古拉（AWS）創建

環境：PoC 或試點	來源：容器	目標：Amazon ECS
R 類型：不適用	技術：容器與微服務；Web 與行動應用程式	AWS 服務：Amazon ECS

Summary

此模式會引導您完成在 Amazon 彈性容器服務 (Amazon ECS) 中將 Java 微服務部署為容器化應用程式的步驟。該模式還使用 Amazon Elastic Container Registry (Amazon ECR) 來管理您的容器，並使用 AWS Fargate 運行您的容器。

先決條件和限制

先決條件

- 在 Docker 上內部部署執行的現有 Java 微服務應用程式
- 有效的 AWS 帳戶
- 熟悉 Amazon ECR、Amazon ECS、AWS Fargate 和 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 熟悉 Java 和碼頭軟件

產品版本

- AWS CLI 版本 1.7 或更新版本

架構

源, 技術, 堆棧

- Java 微服務（例如，使用春季啟動開發）並部署在內部

- Docker

來源架構

目標技術堆疊

- Amazon ECR
- Amazon ECS
- AWS Fargate

目標架構

工具

工具

- Amazon Elastic Container Registry (Amazon ECR) 是全受管的登錄，可讓開發人員輕鬆存放、管理和部署 Docker 容器映像。Amazon ECR 與 Amazon ECS 整合，以簡化您的 development-to-production 工作流程。Amazon ECR 將映像託管在高可用性和可擴展的架構中，因此您可以可靠地為應用程式部署容器。與 AWS Identity and Access Management (IAM) 整合可提供每個儲存庫的資源層級控制。
- Amazon Elastic Container Service (Amazon ECS) 是可高度擴展的高效能容器協調服務，可支援 Docker 容器，並可讓您在 AWS 上輕鬆執行和擴展容器化應用程式。Amazon ECS 無需安裝和操作自己的容器協調軟體、管理和擴展虛擬機器叢集，或在這些虛擬機器上排程容器。
- AWS Fargate 是 Amazon ECS 的運算引擎，可讓您執行容器，而不必管理伺服器或叢集。使用 AWS Fargate，您不再需要佈建、設定和擴展虛擬機器叢集來執行容器。這樣一來即無須選擇伺服器類型、決定何時擴展叢集，或最佳化叢集壓縮。
- Docker 是一個平台，可讓您在稱為容器的套件中建置、測試和交付應用程式。

Code

以下內容 DockerFile 指定所使用的 Java 開發套件 (JDK) 版本、Java 封存 (JAR) 檔案所在的位置、公開的連接埠號碼以及應用程式的進入點。

```
FROM openjdk:8
 ADD target/Spring-docker.jar Spring-docker.jar
 EXPOSE 8080
ENTRYPOINT ["java","-jar","Spring-docker.jar"]
```

史诗

創建一個 Amazon ECR 存儲庫

任務	描述	所需技能
建立 儲存庫。	登入 AWS 管理主控台，然後開啟 Amazon ECR 主控台，網址為 https://console.aws.amazon.com/ecr/repositories。創建一個私人存儲庫。如需指示，請參閱 Amazon ECR 文件中的建立私有存放庫。	開發人員, 系統管理
上傳專案。	開啟儲存庫，然後選擇「檢視推送指令」。按照顯示的步驟上傳專案。這些步驟只有在您使用 AWS CLI 1.7 版或更新版本時才有作用。) 上傳完成後，將組建的 URL 複製到存放庫中。當您在 Amazon ECS 中建立容器時，您將使用此 URL。	開發人員, 系統管理

創建並旋轉容器

任務	描述	所需技能
建立任務定義。	在 Amazon ECS 中執行泊塢視窗容器需要任務定義。在 https://console.aws.amazon	開發人員, 系統管理

任務	描述	所需技能
	.com/ecs/ 開啟 Amazon ECS 主控台，選擇任務定義，然後建立新的任務定義。如需詳細資訊，請參閱 Amazon ECS 文件中的建立任務定義。	
選擇啟動類型。	選擇 Fargate 作為啟動類型。	開發人員, 系統管理
設定工作。	定義作業名稱，並使用適當數量的工作記憶體和 CPU 來設定應用程式。	開發人員, 系統管理
定義容器。	新增容器，提供名稱、Amazon ECR 儲存庫的 URL、記憶體限制和連接埠對應。連接埠 8080 和 80 已針對連接埠對應設定。根據您的應用程式需求設定其餘設定。	開發人員, 系統管理
建立工作。	當工作和容器設定就位時，請建立工作。如需詳細指示，請參閱「相關資源」一節中的連結。	開發人員, 系統管理

建立 Amazon ECS 叢集並設定服務

任務	描述	所需技能
建立或選擇叢集。	Amazon ECS 叢集提供任務或服務的邏輯分組。您可以選擇使用現有叢集或建立新叢集。如果您決定建立新叢集，請根據需求選擇叢集類型。在我們的範例中，我們選取了網路叢集。提供叢集的名稱，並選擇	開發人員, 系統管理

任務	描述	所需技能
	是否要建立新的虛擬私有雲端 (VPC) 以用於 Fargate 工作。	
建立服務。	在叢集內，選擇 [建立服務]。	開發人員, 系統管理
選擇啟動類型。	選擇 Fargate 作為啟動類型。	開發人員, 系統管理
選擇作業定義、修訂版和平台版本。	選擇您要執行的工作，然後選擇作業定義的修訂版本和平台版本。	開發人員, 系統管理
選取 叢集。	從下拉式清單中選取要在其中建立服務的叢集。	開發人員, 系統管理
提供服務名稱。	為您正在建立的服務提供唯一的名稱。	開發人員, 系統管理
指定工作數目。	設定服務啟動時應執行的工作數目。如果您要啟動兩個或多個任務，則需要負載平衡器來平衡任務。要配置的任務的最小數量是一個。	開發人員, 系統管理
設定最小和最大健康百分比。	設定應用程式的最小和最大健全狀況百分比，或接受提供的預設選項。	開發人員, 系統管理
設定部署設定。	根據您的需求選擇部署類型。您可以選擇滾動式更新或藍/綠部署。	開發人員, 系統管理
設定叢集 VPC、子網路和安全群組。	設定叢集 VPC、您要在其上部署應用程式的子網路，以及用於提供入站/輸出連線存取權的安全群組 (HTTP、HTTPS 和連接埠 8080)。	開發人員, 系統管理

任務	描述	所需技能
設定公用 IP 設定。	啟用或停用公用 IP，視您是否要使用公用 IP 位址執行 Fargate 工作而定。	開發人員, 系統管理
設定負載平衡。	設定負載平衡器 (如果您要透過多項工作啟動服務)。您必須先建立負載平衡器及其目標群組，才能啟動服務。	開發人員, 系統管理
設定自動調整規模。	將您的服務設定為使用 Amazon ECS 服務 Auto Scaling，根據您的需求調整所需的任務數量。	開發人員, 系統管理
檢閱設定並建立服務。	檢閱您的服務設定，然後選擇 [建立服務]。	開發人員, 系統管理

切過

任務	描述	所需技能
測試您的應用程式。	使用部署工作時建立的公用 DNS 來測試應用程式。如果應用程式具有負載平衡器，請使用它來測試應用程式，然後切斷應用程式。	開發人員, 系統管理

相關資源

- Amazon ECS 碼頭基礎知識（Amazon ECS 文檔）
- AWS Fargate 上的 Amazon ECS（Amazon ECS 文檔）
- 建立私有儲存庫 (Amazon ECR 文件)
- 建立任務定義 (Amazon ECS 文件)

- 容器定義 (Amazon ECS 文件)
- 建立叢集 (Amazon ECS 文件)
- 設定基本服務參數 (Amazon ECS 文件)
- 設定網路 (Amazon ECS 文件)
- 將您的服務設定為使用負載平衡器 (Amazon ECS 文件)
- 將您的服務設定為使用服務 Auto Scaling (Amazon ECS 文件)

使用 Amazon ECR 和負載平衡在 Amazon ECS 上部署 Java 微服務

R 類型：不適用	資料來源：爪哇	目標：Amazon ECS
創建者：AWS	環境：PoC 或試點	技術：Web 和移動應用程序; 容器和微服務
AWS 服務：Amazon ECS		

Summary

此模式概述了在 Amazon 彈性容器服務 (Amazon ECS) 上部署容器化 Java 微服務架構的步驟，以便於擴展和更快速地開發應用程式。這有助於實現創新並 time-to-market 加速新功能。

該模式還使用 Amazon Elastic Container Registry (Amazon ECR) 來存放和管理以碼頭為基礎的容器，以及使用 Python 指令碼的 AWS CloudFormation 範本來自動化基礎設施的設定。該模式基於在 Amazon 彈性容器服務上部署 Java 微服務後的文章，該服務發佈在 AWS 運算部落格上。

微型服務為軟體開發提供架構和組織方法，其中軟體由小型獨立的服務組成，這些服務會透過定義明確的應用程式設計介面 (API) 進行通訊。小型、獨立的團隊擁有這些服務。

Amazon ECS 是可高度擴展、高效能的容器協調服務。它支援 Docker 容器，可讓您在 AWS 上快速執行和擴展容器化應用程式。使用 Amazon ECS，您不再需要安裝和操作容器協調軟體、管理和擴展虛擬機器 (VM) 叢集，或在這些虛擬機器上排程容器。

透過簡單的 API 呼叫，您可以啟動和停止啟用 Docker 的應用程式、查詢請求的完整狀態，以及存取許多自然功能，例如 AWS Identity and Access Management (IAM) 角色、安全群組、負載平衡器、Amazon CloudWatch 事件、AWS CloudFormation 範本和 AWS 日誌。 CloudTrail

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Java 微服務原始程式碼，搭配 Java 開發套件 1.7 版或更新版本
- 帳戶中使用者的存取金鑰和秘密存取金鑰
- AWS 命令列界面 (AWS CLI)

- Java、AWS 軟體開發套件 (開發套件)，以及碼頭軟體
- 熟悉先前技術的使用
- 熟悉 AWS 服務，例如 Amazon ECS、AWS CloudFormation 和 Elastic Load Balancing

架構

源, 技術, 堆棧

- 在 Java 中實現並部署在內部部署環境中的 Apache Tomcat 上的微服務

目標技術堆疊

- 檢查用戶端要求的應用程式負載平衡器。根據路由規則，負載平衡器會將要求導向至符合狀態之目標群組的執行個體和連接埠。
- 每個微服務的目標群組。對應的服務會使用目標群組來註冊可用的容器執行個體。每個目標群組都有一個路徑，因此當您針對特定微服務呼叫方式時，它會對應至正確的目標群組。這可讓您使用一個 Application Load Balancer 來為路徑存取的所有微服務提供服務。例如，https:///owner/ * 會對應並導向至擁有者微服務。
- 為每個微服務託管容器的 Amazon ECS 叢集。
- 用於託管 Amazon ECS 叢集和相關安全群組的 Amazon Virtual Private Cloud 端 (Amazon VPC) 網路。
- 適用於每個微服務的亞馬遜彈性容器註冊表 (Amazon ECR) 儲存庫。
- 每個微服務的服務或任務定義，用於啟動 Amazon ECS 叢集執行個體上的容器。

目標架構

工具

- Amazon ECS — Amazon ECS 可讓您透過簡單的 API 呼叫啟動和停止容器型應用程式，讓您從集中式服務取得叢集的狀態，並讓您存取許多熟悉的 Amazon Elastic Compute Cloud (Amazon EC2) 功能。
- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種全受管的註冊表，可讓開發人員輕鬆存放、管理和部署 Docker 容器映像。Amazon ECR 與 Amazon ECS 整合，以簡化您的

development-to-production 工作流程。Amazon ECR 將映像託管在高可用性和可擴展的架構中，因此您可以可靠地為應用程式部署容器。與 AWS Identity and Access Management (IAM) 整合可提供每個儲存庫的資源層級控制。

史诗

建立 AWS CloudFormation 範本以設定 Amazon ECS 叢集來託管 Java 微服務

任務	描述	所需技能
佈建 Amazon EC2 Linux 執行個體、安裝泊塢視窗，並為每個微服務建立碼頭檔案。		行動
在 Amazon ECR 上設置碼頭圖像。	使用 Docker 檔案來推送、建立映像檔，並為您的新儲存庫加上標記。對每個微服務執行相同的動作。將新標記的映像推送到存放庫。	行動
建立 AWS CloudFormation 範本。	建立 AWS CloudFormation 範本以佈建虛擬私有雲端 (VPC)、Amazon ECS 叢集和 Amazon Relational Database Service 服務 (Amazon RDS)。	行動

佈建 AWS 服務

任務	描述	所需技能
使用您之前建立的 CloudFormation 範本建立 AWS 基礎設施。	使用 Python 指令碼叫PetClinic用您先前建立的 AWS CloudFormation 範本。https://github.com/awslabs/amazon-ecs-java-microservices/blob/master/2_ECS_Java_Spring_	行動

任務	描述	所需技能
	_Microservices/setup.py 此範本會建立目標環境所需的 AWS 基礎設施。	
建立 Amazon ECR 儲存庫、任務、服務、應用程式負載平衡器和目標群組。	Python 指令碼會讀取 AWS CloudFormation 範本的輸出，並使用 BOTO3 API 呼叫建立 Amazon ECR 儲存庫、任務、服務、Application Load Balancer 和目標群組。	行動

相關資源

- 在 Amazon 彈性容器服務上部署 Java 微服務 (AWS 運算部落格文章)
- Python 腳本
- Amazon ECS 文檔
- Amazon ECS 碼頭基礎知識
- 適用於 Python 的 AWS 開發套件
- Amazon VPC 文件
- Amazon ECR 文件

使用 Amazon EKS 和 Amazon S3 中的頭盔圖儲存庫來部署 Kubernetes 資源和套件

由薩加爾·帕尼格拉希（AWS）創建

環境：PoC 或試點	技術：容器和微服務；DevOps	AWS 服務：Amazon EKS

Summary

此模式可協助您有效管理 Kubernetes 應用程式，無論其複雜性為何。該模式將 Helm 整合到您現有的持續整合和持續交付 (CI/CD) 管道中，以便將應用程式部署到 Kubernetes 叢集中。掌舵是一個 Kubernetes 軟件包管理器，可幫助您管理 Kubernetes 應用程序。掌舵圖有助於定義、安裝和升級複雜的 Kubernetes 應用程式。圖表可以版本化並存儲在 Helm 存儲庫中，這可以改善中斷期間的平均還原時間（MTTR）。

此模式使用 Amazon Elastic Kubernetes Service (Amazon EKS) 的 Kubernetes 叢集。它使用 Amazon Simple Storage Service (Amazon S3) 做為 Helm 圖儲存庫，因此整個組織的開發人員都可以集中管理和存取圖表。

先決條件和限制

先決條件

- 具有虛擬私有雲 (VPC) 的有效亞馬遜網路服務 (AWS) 帳戶
- Amazon EKS 集群
- 在 Amazon EKS 叢集中設定工作者節點並準備好執行工作負載
- Kubectl 用於為用戶端機器中的目標叢集設定 Amazon EKS 庫貝配置檔案
- 用於建立 S3 儲存貯體的 AWS Identity and Access Management (IAM) 存取
- 從用戶端機器存取 Amazon S3 的 IAM (程式設計或角色)
- 源代碼管理和 CI/CD 管道

限制

- 目前不支援升級、刪除或管理自訂資源定義 (CRD)。

- 如果您使用參照 CRD 的資源，則必須單獨安裝 CRD（在圖表之外）。

產品版本

- 頭盔

架構

目標技術堆疊

- Amazon EKS
- Amazon VPC
- Amazon S3
- 源代碼管理
- Helm
- 庫貝克特爾

目標架構

自動化和規模

- AWS CloudFormation 可用於自動化基礎設施的建立。如需詳細資訊，請參閱 Amazon EKS 文件 CloudFormation中的使用 AWS 建立 Amazon EKS 資源。
- Helm 將被納入您現有的 CI/CD 自動化工具中，以自動化 Helm 圖表的包裝和版本控制（超出此模式的範圍）。
- GitVersion 或者 Jenkins 內建編號可用於自動化圖表的版本控制。

工具

工具

- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，可在 AWS 上執行 Kubernetes，而不需要站立或維護自己的 Kubernetes 控制平面。Kubernetes 是一套開放原始碼系統，用於容器化應用程式的自動化部署、擴展與管理。

- 掌舵 — Helm 是 Kubernetes 的套件管理員，可協助您在 Kubernetes 叢集上安裝及管理應用程式。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Kubectl — Kubectl 是一個命令列公用程式，可針對 Kubernetes 叢集執行命令。

Code

範例程式碼已附加。

史诗

配置和初始化頭盔

任務	描述	所需技能
安裝 Helm 客戶端。	要在本地系統上下載並安裝 Helm 客戶端，請使用以下命令。 `sudo curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash`	DevOps 工程師
驗證頭盔的安裝。	若要驗證 Helm 是否能夠與 Amazon EKS 叢集內的 Kubernetes API 伺服器通訊，請執行。`helm version`	DevOps 工程師

在 Amazon EKS 群集中創建並安裝頭盔圖

任務	描述	所需技能
為 NGINX 創建一個頭盔圖表。	要創建在客戶端機器`my-nginx`上命名的頭盔圖，請	DevOps 工程師

任務	描述	所需技能
	運行`helm create my-nginx`。	
檢閱圖表的結構。	若要檢閱圖表的結構，請執行 tree 指令`tree my-nginx/` 。	DevOps 工程師
停用圖表中的服務帳戶建立。	在`values.yaml` 區`serviceAccount` 段下方，將`create`金鑰設定為`false`。此功能已關閉，因為不需要為此模式建立服務帳戶。	DevOps 工程師
驗證 (lint) 修改過的圖表是否存在語法錯誤。	若要在目標叢集中安裝之前驗證圖表是否有任何語法錯誤，請執行`helm lint my-nginx/` 。	DevOps 工程師
安裝圖表以部署 Kubernetes 資源。	要運行 Helm 圖安裝，請使用以下命令。 `helm install --name my-nginx-release --debug my-nginx/ --namespace helm-space` 選用`debug`旗標會在安裝期間輸出所有除錯訊息。該 `namespace` 標誌指定將在其中創建此圖表的資源部分的命名空間。	DevOps 工程師

任務	描述	所需技能
檢閱 Amazon EKS 叢集中的資源。	若要檢閱作為命helm-space 名空間中 Helm 圖表一部分建立的資源，請使用下列命令。 `kubectl get all -n helm-space`	DevOps 工程師

還原至先前版本的 Kubernetes 應用程式

任務	描述	所需技能
修改並升級發行版本。	若要修改圖表，請在中values.yaml ，將replicaCount 值變更為2。然後通過運行以下命令升級已安裝的版本。 `helm upgrade my-nginx-release my-nginx/ --namespace helm-space`	DevOps 工程師
查看頭盔版本的歷史記錄。	若要列出已使用 Helm 安裝之特定版本的所有修訂版本，請執行下列命令。 `helm history my-nginx-release`	DevOps 工程師
檢閱特定修訂的詳細資訊。	在切換或復原至工作中版本之前，以及在安裝修訂版之前進行額外的驗證層，請使用下列指令檢視哪些值已傳遞給每個修訂版本。	DevOps 工程師

任務	描述	所需技能
	`helm get --revision=2 my-nginx-release`	
回滾到以前的版本。	若要復原至先前的修訂版，請使用下列指令。 `helm rollback my-nginx-release 1` 此範例將復原至修訂編號 1。	DevOps 工程師

將 S3 儲存貯體初始化為 Helm 儲存庫

任務	描述	所需技能
為舵圖表創建 S3 存儲桶。	建立唯一的 S3 儲存貯體。在值區中，建立名為的資料夾 `charts`。此模式中的範例使用`s3://my-helm-charts/charts` 作為目標圖表儲存庫。	雲端管理員
安裝 Amazon S3 的頭盔插件。	要在客戶端計算機上安裝 helm-s3 插件，請使用以下命令。 `helm plugin install https://github.com/hypnoglow/helm-s3.git --version 0.10.0` 注：頭盔 V3 支持可與插件版本 0.9.0 及以上。	DevOps 工程師

任務	描述	所需技能
初始化 Amazon S3 掌舵存儲庫。	要將目標文件夾初始化為 Helm 存儲庫，請使用以下命令。 `helm S3 init s3://my-helm-charts/charts` 命令會在目標中建立 `index.yaml` 檔案，以追蹤儲存在該位置的所有圖表資訊。	DevOps 工程師
將 Amazon S3 存儲庫添加到掌舵。	若要在用戶端機器中新增存放庫，請使用下列命令。 `helm repo add my-helm-charts s3://my-helm-charts/charts` 此命令將別名添加到 Helm 客戶端機器中的目標存儲庫。	DevOps 工程師
檢閱儲存庫清單。	若要檢視 Helm 用戶端機器中的存放庫清單，請執行`helm repo list`。	DevOps 工程師

在 Amazon S3 掌舵存儲庫中 Package 和存儲圖表

任務	描述	所需技能
封裝圖表。	若要封裝您建立的`my-nginx`圖表，請執行`helm package ./my-nginx/` 。此命令會將`my-nginx`圖表資料夾的所有內容封裝到封存檔案中，該檔案會使用	DevOps 工程師

任務	描述	所需技能
	檔Chart.yaml 案中提到的版本號碼來命名。	
將套件存放在 Amazon S3 掌舵儲存庫中。	若要將套件上傳到 Amazon S3 中的 Helm 儲存庫，請使用.tgz檔案的正確名稱執行下列命令。 `helm s3 push ./my-nginx-0.1.0.tgz my-helm-charts`	DevOps 工程師
搜尋「頭盔」圖表。	若要確認圖表同時出現在本機和 Amazon S3 的 Helm 儲存庫中，請執行下列命令。 `helm search repo my-nginx`	DevOps 工程師

修改、編列版本和封裝圖表

任務	描述	所需技能
修改並封裝圖表。	在中values.yaml ，將replicaCount 值設定為1。然後通過運行來打包圖表helm package ./my-nginx/ ，這次將版本更改Chart.yaml 為0.1.1。 版本控制最理想地通過使用 CI/CD 管道中的工具（例如 GitVersion Jenkins 構建編號）進行自動化更新。自動化版本號超出此模式的範圍。	DevOps 工程師

任務	描述	所需技能
將新版本推送到 Amazon S3 中的頭盔存儲庫。	若要將版本為 0.1.1 的新套件推送至 Amazon S3 中的 `my-helm-charts` Helm 儲存庫，請執行下列命令。 `helm s3 push ./my-nginx-0.1.1.tgz my-helm-charts`	DevOps 工程師

從 Amazon S3 掌舵存儲庫搜索並安裝圖表

任務	描述	所需技能
搜索我的 nginx 圖表的所有版本。	若要檢視圖表的所有可用版本，請使用`--versions` 旗標執行下列命令。 `helm search repo my-nginx --versions` 如果沒有旗標，Helm 預設會顯示圖表的最新上傳版本。	DevOps 工程師
從 Amazon S3 掌舵存儲庫安裝圖表。	先前工作的搜尋結果會顯示`my-nginx`圖表的多個版本。若要從 Amazon S3 掌舵儲存庫安裝新版本 (0.1.1)，請使用以下命令。 `helm upgrade my-nginx-release my-helm-charts/my-nginx --version 0.1.1 --namespace helm-space`	DevOps 工程師

相關資源

- 頭盔文件
- 幫助 -3 插件（麻省理工學院許可證）
- HEM 客戶端二進
- Amazon EKS 文檔

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用容器映像部署 Lambda 函數

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：容器與微服務；雲端原生；軟體開發與測試；無伺服器	工作負載：所有其他工作
AWS 服務：Amazon EC2 容器登錄；AWS Lambda		

Summary

AWS Lambda 支援容器映像檔做為部署模型。此模式示範如何透過容器映像部署 Lambda 函數。

Lambda 是一種無伺服器、事件驅動的運算服務，您可以使用它為幾乎任何類型的應用程式或後端服務執行程式碼，而無需佈建或管理伺服器。透過 Lambda 函數的容器映像支援，您可獲得應用程式成品最高 10 GB 儲存空間的優勢，以及使用熟悉的容器映像開發工具的能力。

此模式中的範例使用 Python 做為基礎程式設計語言，但您可以使用其他語言，例如 Java、Node.js 或 Go。該模式使用 AWS CodeCommit 作為來源，但您也可以使用 GitHub Bitbucket 或 Amazon Simple Storage Service (Amazon S3)。

先決條件和限制

先決條件

- Amazon Elastic Container Registry (Amazon ECR) 激活
- 應用程式碼
- 帶有運行時界面客戶端和最新版本的 Python 的碼頭圖像

限制

- 支援的最大影像大小為 10 GB。
- 以 Lambda 為基礎的容器部署的執行時間上限為 15 分鐘。

架構

目標技術堆疊

- Python 编程语言
- AWS CodeBuild
- AWS CodeCommit
- Docker 映像檔
- Amazon ECR
- AWS Identity and Access Management (IAM)
- AWS Lambda
- Amazon CloudWatch 日誌

目標架構

1. 您可以建立儲存庫並使用提交應用程式程式碼 CodeCommit。
2. 對進行變更時，會啟動 CodeBuild 專案 CodeCommit，並將其用作來源提供者。
3. 該 CodeBuild 項目創建碼頭映像並將圖像發佈到 Amazon ECR。
4. 您可以使用 Amazon ECR 中的映像來建立 Lambda 函數。

自動化和規模

您可以使用 AWS CloudFormation、AWS Cloud Development Kit (AWS CDK) 或開發套件中的 API 操作來自動化此模式。Lambda 可以根據請求的數量自動擴展，您可以使用並發參數對其進行調整。如需詳細資訊，請參閱 L ambda 文件。

工具

AWS 服務

- AWS CloudFormation 設計師提供整合的 JSON 和 YAML 編輯器，可協助您檢視和編輯 CloudFormation 範本。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。

- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS CodeStar 是一種雲端服務，用於在 AWS 上建立、管理和使用軟體開發專案。對於此模式，您可以使用 AWS CodeStar 或其他開發環境。
- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

其他工具

- Docker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中提供軟體。

最佳實務

- 使您的功能盡可能高效和小，以避免加載不必要的文件。
- 努力在 Docker 文件列表中使靜態圖層更高，並將更頻繁變化的圖層放置在較低的位置。這改善了緩存，從而提高了性能。
- 映像擁有者負責更新和修補映像。將該更新節奏添加到您的操作流程中。如需詳細資訊，請參閱 AWS Lambda 文件。

史诗

在中建立專案 CodeBuild

任務	描述	所需技能
創建一個 CodeCommit 存儲庫。	創建一個包含 Docker 文件，文件和應用程序源代碼的 CodeCommit 存儲庫。`buildspec.yaml` 如需詳細資訊，請參閱 AWS CodeCommit 文件。	開發人員

任務	描述	所需技能
建立 CodeBuild 專案。	在 CodeBuild 控制台上，創建一個使用 CodeCommit repo 和`buildspec.yaml` 文件的新項目。您將使用該 CodeBuild 項目來創建映像。 確認已啟用特權模式。要構建碼頭圖像，這是必要的。否則，映像檔將無法成功建置。 提供專案名稱和描述的值。對於來源提供者，請選擇 CodeCommit。如需詳細資訊，請參閱 AWS 文件。	開發人員
編輯碼頭文件。	Docker 文件應位於您正在開發應用程序的頂級目錄中。Python 代碼應該在`src`文件夾中。 建立映像檔時，請使用 Lambda 官方支援的映像檔。否則，將發生引導錯誤，從而使打包過程更加困難。 如需詳細資訊，請參閱其他資訊一節。	開發人員

任務	描述	所需技能
在 Amazon ECR 中創建一個存儲庫。	在 Amazon ECR 中建立容器儲存庫。在下面的示例命令中，創建的存儲庫的名稱是`cf-demo`。儲存庫將在`buildspec.yaml` 檔案中重複使用。 `aws ecr create-repository --cf-demo`	AWS 管理員、開發人員
將映像推送至 Amazon ECR。	您可以使用執 CodeBuild 行映像構建過程。 CodeBuild 需要與 Amazon ECR 互動並與 S3 合作的權限。作為流程的一部分，Docker 映像檔會建立並推送至 Amazon ECR 登錄。如需範本和程式碼的詳細資訊，請參閱其他資訊一節。	開發人員
確認映像檔位於存放庫中。	若要驗證映像檔是否位於儲存庫中，請在 Amazon ECR 主控台上選擇儲存庫。如果在 Amazon ECR 設定中開啟了該功能，則應列出影像並附有標籤，並附上弱點掃描報告的結果。 如需詳細資訊，請參閱 AWS 文件。	開發人員

建立 Lambda 函數以執行映像檔

任務	描述	所需技能
建立 Lambda 函數。	在 Lambda 主控台上，選擇 [建立函數]，然後選擇 [容器	應用程式開發人員

任務	描述	所需技能
	映像檔]。輸入 Amazon ECR 儲存庫中映像檔的函數名稱和 URI，然後選擇 [建立函數]。如需詳細資訊，請參閱 AWS Lambda 文件。	
測試 Lambda 函數。	若要叫用並測試函數，請選擇 [測試]。如需詳細資訊，請參閱 AWS Lambda 文件。	應用程式開發人員

故障診斷

問題	解決方案
構建不成功。	1. 檢查項目的特權模式是否已打 CodeBuild 開。 2. 確保 Docker 相關命令具有必要的權限。嘗試添sudo加到命令。 3. 確認與相關聯的 IAM 角色具 CodeBuild 有包含適當動作的政策，可與 Amazon ECR、Amazon S3 和 CloudWatch 日誌互動。

相關資源

- Lambda 的基本映像
- 碼頭工人樣本 CodeBuild
- 傳遞臨時憑證

其他資訊

編輯碼頭文件

下列程式碼顯示您在 Docker 檔案中編輯的命令。

```
FROM public.ecr.aws/lambda/python:3.11

# Copy function code
COPY app.py ${LAMBDA_TASK_ROOT}
COPY requirements.txt  ${LAMBDA_TASK_ROOT}

# install dependencies
RUN pip3 install --user -r requirements.txt

# Set the CMD to your handler (could also be done as a parameter override outside of
 the Dockerfile)
CMD [ "app.lambda_handler" ]
```

該FROM命令值對應於在公共 Amazon ECR 映像存儲庫中使用 Lambda 函數的 Python 3.11 基本映像。

此命COPY app.py ${LAMBDA_TASK_ROOT}令會將程式碼複製到 Lambda 函數將使用的工作根目錄。此命令使用環境變量，因此我們不必擔心實際路徑。要運行的函數作為參數傳遞給命CMD ["app.lambda_handler"]令。

該COPY requirements.txt命令捕獲代碼所需的依賴關係。

此命RUN pip install --user -r requirements.txt令會將相依性安裝至本機使用者目錄。

若要建立映像檔，請執行下列命令。

```
docker build -t <image name> .
```

在 Amazon ECR 中添加圖像

在下面的代碼中，替換aws_account_id為帳戶號碼，us-east-1如果您使用的是不同的區域進行替換。buildspec檔案會使用 CodeBuild 組建編號，將映像版本唯一識別為標籤值。您可以變更此選項以符合您的需求。

構建規格的自定義代碼

```
phases:
  install:
    runtime-versions:
```

```
      python: 3.11
  pre_build:
    commands:
      - python3 --version
      - pip3 install --upgrade pip
      - pip3 install --upgrade awscli
      - sudo docker info
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - ls
      - cd app
      - docker build -t cf-demo:$CODEBUILD_BUILD_NUMBER .
      - docker container ls
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin aws_account_id.dkr.ecr.us-east-1.amazonaws.com
      - docker tag cf-demo:$CODEBUILD_BUILD_NUMBER aws_account_id.dkr.ecr.us-
east-1.amazonaws.com/cf-demo:$CODEBUILD_BUILD_NUMBER
      - docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/cf-demo:
$CODEBUILD_BUILD_NUMBER
```

在 Amazon EKS 上部署範例 Java 微服務，並使用應用程式負載平衡器公開微服務

由維傑·湯普森（AWS）和阿卡瑪哈德維（AWS）創建

環境：PoC 或試點	技術：容器與微服務	工作負載：開源
AWS 服務：Amazon EC2 容器註冊表; Amazon EKS; Amazon ECR		

Summary

此模式說明如何使用`eksctl`命令列公用程式和亞馬遜彈性容器登錄 (Amazon ECR)，在 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 上將範例 Java 微服務部署為容器化應用程式。您可以使用應用程式負載平衡器負載平衡應用程式流量。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 macOS、Linux 或視窗上安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI) 1.7 版或更新版本
- 正在運行的碼頭守護進程
- 在 macOS、Linux 或視窗上安裝和設定的`eksctl`命令列公用程式 (如需詳細資訊，請參閱 Amazon EKS 文件中的開始使用 — 例如。)
- 在 macOS、Linux 或視窗上安裝和設定的`kubectl`命令列公用程式 (如需詳細資訊，請參閱 Amazon EKS 說明文件中的安裝或更新 kubectl)。

限制

- 此模式不涵蓋應用程式負載平衡器的 SSL 憑證安裝。

架構

目標技術堆疊

- Amazon ECR
- Amazon EKS
- Elastic Load Balancing

目標架構

下圖顯示了在 Amazon EKS 上容器化 Java 微服務的架構。

工具

- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- E@@ lastic Load Balancing 會在一個或多個可用區域中自動將傳入流量分配到多個目標，例如 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。
- eksctl 可協助您在 Amazon EKS 上建立叢集。
- kubectl 可讓您針對 Kubernetes 叢集執行命令。
- Docker 可協助您在稱為容器的套件中建置、測試和交付應用程式。

史诗

通過使用插件創建一個 Amazon EKS 集群

任務	描述	所需技能
建立 Amazon EKS 叢集。	若要建立使用兩個 t2.small Amazon EC2 執行個體做為節	開發人員, 系統管理

任務	描述	所需技能
	點的 Amazon EKS 叢集，請執行下列命令： `eksctl create cluster --name <your-cluster-name> --version <version-number> --nodes=1 --node-type=t2.small` 備註：該過程可能需要 15 到 20 分鐘。建立叢集之後，適當的 Kubernetes 組態就會新增至您的 kubeconfig 檔案。您可以在稍後的步驟中使用該`kubeconfig` 檔案來部署應用程式。`kubectl`	
驗證 Amazon EKS 叢集。	若要驗證叢集是否已建立，以及您是否可以連線到叢集，請執行`kubectl get nodes`命令。	開發人員, 系統管理

創建一個 Amazon ECR 存儲庫並推送碼頭映像。

任務	描述	所需技能
建立 Amazon ECR 儲存庫。	請遵循 Amazon ECR 文件中建立私有存放庫中的指示進行操作。	開發人員, 系統管理
創建一個聚甲醛 XML 文件。	根據此模`pom.xml`式的其他信息部分中的示例 POM 文件代碼創建文件。	開發人員, 系統管理
建立來源檔案。	根據下列範例，建立`HelloWorld.java` 在`src/`	

任務	描述	所需技能
	main/java/eksExample 路徑中呼叫的來源檔案： `package eksExample;` `import static spark.Spark.get;` `public class HelloWorld {` `  public static void main(String[] args) {` `  get("/", (req, res) -> {` `  return "Hello World!";` `  });` `  }` `}` 請務必使用下列目錄結構： `### Dockerfile` `### deployment.yaml` `### ingress.yaml` `### pom.xml` `### service.yaml` `### src` `  ### main` `  ### java` `  ### eksExample` `  ### HelloWorld.java`	
建立 Dockerfile。	Dockerfile 根據此模式的其他信息部分中的示例 Dockerfile 代碼創建一個。	開發人員, 系統管理

任務	描述	所需技能
構建並推送碼頭映像。	在您要建立、標記影像並Dockerfile 將其推送至 Amazon ECR 的目錄中，執行下列命令： `aws ecr get-login -password --region <region>| docker login --username <username> --password-stdin <account_number>.dkr.ecr.<region>.amazonaws.com` `docker buildx build --platform linux/amd64 -t hello-world-java:v1 .` `docker tag hello-world-java:v1 <account_number>.dkr.ecr.<region>.amazonaws.com/<repository_name>:v1` `docker push <account_number>.dkr.ecr.<region>.amazonaws.com/<repository_name>:v1` 注意：在上述命令中修改 AWS 區域、帳戶編號和儲存庫詳細資訊。請務必記下圖片 URL 以供日後使用。 重要事項：配備 M1 晶片的 macOS 系統在建置與 AMD64 平台上執行的 Amazon EKS 相容的映像時發生問題。要解決此問題，請使用 docker buildx	

任務	描述	所需技能
	構建可在 Amazon EKS 上運行的 Docker 映像。	

部署 Java 微服務

任務	描述	所需技能
建立部署檔案。	建立`deployment.yaml` 根據此病毒碼「其他資訊」區段中的範例部署檔案程式碼呼叫的 YAML 檔案。 備註：使用您先前複製的映像 URL 做為 Amazon ECR 儲存庫的映像檔案路徑。	開發人員, 系統管理
在 Amazon EKS 叢集上部署 Java 微服務。	若要在 Amazon EKS 叢集中建立部署，請執行命`kubectl apply -f deployment.yaml` 令。	開發人員, 系統管理
驗證網繭的狀態。	1. 若要驗證網繭的狀態，請執行`kubectl get pods`命令。 2. 等待狀態變更為 [就緒]。	開發人員, 系統管理
建立服務。	1. `service.yaml` 根據此模式的「其他資訊」區段中的「範例」服務檔案程式碼建立名為的檔案。 2. 執行 `kubectl apply -f service.yaml` 命令。	開發人員, 系統管理

任務	描述	所需技能
安裝 AWS Load Balancer 控制器附加元件。	請遵循 Amazon EKS 文件中安裝 AWS Load Balancer 控制器外掛程式的指示。 備註：您必須安裝附加元件，才能為 Kubernetes 服務建立 Application Load Balancer 載平衡器或 Network Load Balancer。	開發者，系統管理員
建立輸入資源。	`ingress.yaml` 根據此模式的其他資訊區段中的範例輸入資源檔案程式碼，建立一個 YAML 檔案。	開發人員, 系統管理
建立應用程式負載平衡器。	若要部署輸入資源並建立應用程式負載平衡器，請執行命`kubectl apply -f ingress.yaml` 令。	開發人員, 系統管理

測試應用程式。

任務	描述	所需技能
測試並驗證應用程式。	1. 若要從 ADDRESS 欄位取得負載平衡器的 DNS 名稱，請執行`kubectl get ingress.networking.k8s.io/java-microservice-ingress` 命令。 2. 在與 Amazon EKS 節點位於相同 VPC 中的 EC2 執行個體上，執行命令。`curl -`	開發人員, 系統管理

任務	描述	所需技能
	v <DNS address from previous command>	

相關資源

- 建立私有儲存庫 (Amazon ECR 文件)
- 推送碼頭圖像（Amazon ECR 文檔）
- 入口控制器（Amazon EKS 工作坊）
- 碼頭構建（碼頭文檔）

其他資訊

聚甲醛文件示例

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>


  <groupId>helloWorld</groupId>
  <artifactId>helloWorld</artifactId>
  <version>1.0-SNAPSHOT</version>


  <dependencies>
    <dependency>
      <groupId>com.sparkjava</groupId><artifactId>spark-core</
artifactId><version>2.0.0</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
```

```
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-jar-plugin</
artifactId><version>2.4</version>
        <configuration><finalName>eksExample</finalName><archive><manifest>
              <addClasspath>true</addClasspath><mainClass>eksExample.HelloWorld</
mainClass><classpathPrefix>dependency-jars/</classpathPrefix>
            </manifest></archive>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-compiler-plugin</
artifactId><version>3.1</version>
        <configuration><source>1.8</source><target>1.8</target></configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId><artifactId>maven-assembly-plugin</
artifactId>
        <executions>
          <execution>
            <goals><goal>attached</goal></goals><phase>package</phase>
            <configuration>
              <finalName>eksExample</finalName>
              <descriptorRefs><descriptorRef>jar-with-dependencies</descriptorRef></
descriptorRefs>
              <archive><manifest><mainClass>eksExample.HelloWorld</mainClass></
manifest></archive>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

示例碼頭文件

```
FROM bellsoft/liberica-openjdk-alpine-musl:17

RUN apk add maven
WORKDIR /code

# Prepare by downloading dependencies
ADD pom.xml /code/pom.xml
RUN ["mvn", "dependency:resolve"]
```

```
RUN ["mvn", "verify"]

# Adding source, compile and package into a fat jar
ADD src /code/src
RUN ["mvn", "package"]

EXPOSE 4567
CMD ["java", "-jar", "target/eksExample-jar-with-dependencies.jar"]
```

部署檔案範例

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservice-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app.kubernetes.io/name: java-microservice
  template:
    metadata:
      labels:
        app.kubernetes.io/name: java-microservice
    spec:
      containers:
      - name: java-microservice-container
        image: .dkr.ecr.amazonaws.com/:
        ports:
        - containerPort: 4567
```

範例服務檔案

```
apiVersion: v1
kind: Service
metadata:
  name: "service-java-microservice"
spec:
  ports:
    - port: 80
      targetPort: 4567
      protocol: TCP
  type: NodePort
```

```
  selector:
    app.kubernetes.io/name: java-microservice
```

輸入資源檔案範例

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: "java-microservice-ingress"
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/load-balancer-name: apg2
    alb.ingress.kubernetes.io/target-type: ip
  labels:
    app: java-microservice
spec:
  rules:
    - http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: "service-java-microservice"
                port:
                  number: 80
```

使用 AWS 副駕駛員將叢集應用程式部署到 Amazon ECS

由讓·巴蒂斯特·吉盧瓦 (AWS)、馬修·喬治 (AWS) 和托馬斯·斯科特 (AWS) 創建

程式碼儲存庫：叢集範例應用程式	環境：生產	技術：容器與微服務；企業生產力；雲端原生產力；軟體開發與測試
AWS 服務：Amazon ECS; AWS Fargate; Amazon ECR		

Summary

此模式示範如何以兩種方式在 Amazon 彈性容器服務 (Amazon ECS) 叢集中部署容器，方法是使用 Amazon Web Services (AWS) 管理主控台，並使用 AWS 副駕駛員，以示範 AWS Copilot 如何簡化部署任務。

Amazon ECS 是可高度擴展、快速的容器管理服務，可讓您輕鬆執行、停止和管理叢集上的容器。您可用來在服務中執行個別任務或任務的任務定義中會對您的容器進行定義。您可以在由 AWS Fargate 管理的無伺服器基礎設施上執行任務和服務。或者，若要進一步控制基礎設施，您可以在您管理的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體叢集上執行任務和服務。

AWS Copilot 命令列界面 (CLI) 命令可簡化從本機開發環境在 Amazon ECS 上生產就緒容器化應用程式的建置、釋放和操作。AWS Copilot CLI 與支援現代應用程式最佳實務的開發人員工作流程保持一致：從使用基礎設施即程式碼到建立代表使用者佈建的持續整合和持續交付 (CI/CD) 管道。您可以在日常開發和測試週期中使用 AWS Copilot CLI 作為 AWS 管理主控台的替代方案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 在本機安裝和設定為使用您的 AWS 帳戶 (請參閱 AWS CLI 文件中的安裝說明和組態說明)
- AWS 副駕駛員已在本機安裝 (請參閱 Amazon ECS 文件中的安裝說明)

- 安裝在本地計算機上的 Docker（請參閱 Docker 文檔）

限制

- Docker 強制執行免費方案中每個 IP 位址每 6 小時 100 張容器映像的提取限制。

架構

目標技術堆疊

- AWS 環境透過虛擬私有雲端 (VPC)、公有和私有子網路以及安全群組進行設定
- Amazon ECS 叢集
- Amazon ECS 服務和任務定義
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon DynamoDB
- Application Load Balancer
- AWS Fargate
- Amazon Identity and Access Management (IAM)
- Amazon CloudWatch
- AWS CloudTrail

目標架構

當您為此病毒碼部署範例應用程式時，會在不同的可用區域中建立和部署多個工作。每個任務都會將資料存放在 Amazon DynamoDB 中。當您訪問任務的網頁時，您可以查看所有其他任務的數據。

工具

AWS 服務

- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種 AWS 受管容器映像登錄服務，安全、可擴展且可靠。Amazon ECR 支援私有儲存庫，其具有使用 IAM 的資源型許可。
- Amazon ECS — 亞馬遜彈性容器服務 (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，用於在叢集上執行、停止和管理容器。您可以在由 AWS Fargate 管理的無伺服器基礎設施上執行

任務和服務。或者，若要進一步控制基礎設施，您可以在您管理的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體叢集上執行任務和服務。

- AWS Copilot — AWS Copilot 提供命令列界面，可協助您在 AWS 上啟動和管理容器化應用程式，包括推送到登錄、建立任務定義以及建立叢集。
- AWS Fargate — AWS Fargate 是無伺服器 pay-as-you-go 運算引擎，可讓您專注於建置應用程式，而無需管理伺服器。AWS Fargate 與 Amazon ECS 和亞馬 Amazon Elastic Kubernetes Service（Amazon EKS）兼容。當您使用 Fargate 啟動類型或 Fargate 容量提供者執行 Amazon ECS 任務和服務時，將會在容器中封裝應用程式、指定 CPU 和記憶體需求、定義聯網和 IAM 政策，並啟動應用程式。每個 Fargate 任務都有自己的隔離邊界，不會與其他任務共享基礎內核，CPU 資源，內存資源或 elastic network interface。
- 亞馬遜 DynamoDB — Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- E@@ lastic Load Balancing (ELB) — Elastic Load Balancing 會自動將傳入流量分配到一或多個可用區域中的多個目標，例如 EC2 執行個體、容器和 IP 地址。其會監控已註冊目標的運作狀態，並且僅將流量路由至運作狀態良好的目標。當傳入流量隨著時間發生變化，Elastic Load Balancing 會擴展您的負載平衡器。他可以自動擴展以因應絕大多數的工作負載。

工具

- 泊塢工人命令行界面
- AWS Command Line Interface (AWS CLI)
- AWS 副駕駛命令列界面

Code

您可以在「叢集範例應用程式」存放庫中找到此模式中使用的範例應用程式的程式碼。 GitHub請遵循下一節中的指示來使用範例檔案。

史诗

部署應用程式堆疊-選項 1 (AWS 管理主控台)

任務	描述	所需技能
克隆存 GitHub 儲庫。	使用以下命令克隆示例代碼存儲庫：	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	`git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app`	
建立您的 Amazon ECR 儲存庫。	1. 登入 AWS 管理主控台，然後開啟 Amazon ECR 主控台，網址為 https://console.aws.amazon.com/ecr/repositories。 2. 選擇 Create repository (建立儲存庫)。 3. 對於存放庫名稱，請輸入cluster-sample-app。 4. 對於所有其他設定，請保留預設值。 5. 選擇 Create repository (建立儲存庫)。 如需詳細資訊，請參閱 Amazon ECR 文件中的建立私有存放庫。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
建置、標記您的 Docker 映像檔，並將其推送至您的 Amazon ECR 儲存庫。	1. 選取您剛建立的儲存庫，然後選擇 [檢視推送指令]。 2. 複製顯示的命令並在本地運行它們以構建，標記和推送碼 docker 映像。這些命令將類似於以下內容。 要向註冊表驗證您的 Docker 客戶端： `aws ecr get-login -password --region <YOUR_AWS_REGION> | docker login --username AWS --password-stdin <YOUR_AWS_ACCOUNT>.dkr.ecr.<YOUR_AWS_REGION>.amazonaws.com` 若要建立您的泊塢視窗映像檔： `docker build -t cluster-sample-app .` 要標記您的碼頭圖像： `docker tag cluster-sample-app:latest <YOUR_AWS_ACCOUNT>.dkr.ecr.<YOUR_AWS_REGION>.amazonaws.com/cluster-sample-app:latest`	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	要將 Docker 映像推送到您的存儲庫： `docker push <YOUR_AWS_ACCOUNT>.dkr.ecr.<YOUR_AWS_REGION>.amazonaws.com/cluster-sample-app:latest`	

任務	描述	所需技能
部署應用程式堆疊。	1. 開啟 AWS 主 CloudFormation 控台，網址為 https://console.aws.amazon.com/cloudformation/。 2. 選擇建立堆疊。 3. 在「準備範本」區段中，選擇「範本已就緒」。 4. 在 Specify template (指定範本) 區段中，選擇 Upload a template file (上傳範本檔案)。 5. 選擇您從 GitHub 儲存庫複製`cluster-sample-app-stack.yml` 的本機檔案做為 CloudFormation 範本，然後選擇 [下一步]。 6. 輸入堆疊的名稱，然後選擇 [下一步]。 7. 保留所有預設選項，然後選擇 [下一步]。 8. 檢閱所有選項、確認 IAM 資源的建立，然後選擇 [建立堆疊]。 9. 部署應用程式堆疊後，請選擇 [輸出] 索引標籤，複製 URL，然後在瀏覽器中開啟它以存取應用程式。 如需部署 CloudFormation 範本的詳細資訊，請參閱 AWS CloudFormation 文件中的建立堆疊。	AWS 應用程 DevOps式開發人員

部署應用程式堆疊 — 選項 2 (AWS 副駕駛 CLI)

任務	描述	所需技能
克隆存 GitHub 儲庫。	使用以下命令克隆示例代碼存儲庫： `git clone https://github.com/aws-samples/cluster-sample-app cluster-sample-app && cd cluster-sample-app`	AWS 應用程式開發人員 DevOps
使用 AWS 副駕駛 CLI 將您的容器映像部署到 AWS。	在專案的根目錄中使用下列指令，只需一個步驟即可部署應用程式： `copilot init --app cluster-sample-app --name demo --type "Load Balanced Web Service" --dockerfile ./Dockerfile --port 8080 --deploy` 然後，您應該可以使用作為輸出提供的 DNS 名稱來訪問應用程序。	AWS 應用程式開發人員 DevOps

刪除創建的資源

任務	描述	所需技能
刪除透過 AWS 管理主控台建立的資源。	如果您使用選項 1 (AWS 管理主控台) 部署應用程式堆疊，請在準備好刪除所建立的資源時遵循下列步驟：	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	1. 請在以下位置開啟 CloudFormation 主控台。 https://console.aws.amazon.com/cloudformation/ 2. 選取您建立的堆疊，然後選擇 [刪除]。 3. 在 https://console.aws.amazon.com/ecr/repositories 開啟 Amazon ECR 主控台。 4. 選取您建立的存放庫，然後選擇 [刪除]。	
刪除 AWS 副駕駛建立的資源。	如果您使用選項 2 (AWS Copilot CLI) 部署應用程式堆疊，請在準備刪除所建立的資源時，從專案的根目錄執行下列命令： `copilot app delete`	AWS 應用程式開發人員 DevOps

相關資源

- 安裝或更新最新版本的 AWS CLI (AWS CLI 文件)
- 使用 AWS 副駕駛員命令列界面 (Amazon ECS 文件)
- AWS Fargate 上的 Amazon ECS（Amazon ECR 文檔）
- Amazon ECS 文件
- Amazon ECR 文件
- Amazon CloudFormation 文檔
- 碼頭桌面（碼頭文檔）

在 Amazon EKS 叢集上部署以 gRPC 為基礎的應用程式，並使 Application Load Balancer 存取

由基蘭庫馬爾·錢德拉什卡（AWS）和 Huy 阮（AWS）創建

代碼存儲庫：grpc-traffic-on-alb到 eks	環境：PoC 或試點	技術：容器與微服務；內容傳遞；Web 和行動應用程式
工作負載：所有其他工作	AWS 服務：Amazon EKS；Elastic Load Balancing (ELB)	

Summary

此模式說明如何在 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 叢集上託管以 gRPC 為基礎的應用程式，並透過應用程式負載平衡器安全地存取該應用程式。

GrPC 是可在任何環境中執行的開放原始碼遠端程序呼叫 (RPC) 架構。您可以將其用於微服務整合和用戶端伺服器通訊。如需 GrPC 的詳細資訊，請參閱 AWS 部落格文章 end-to-end HTTP/2 和 gRPC 的 Application Load Balancer 支援。

此模式說明如何託管在 Amazon EKS 上 Kubernetes 網繭上執行的基於 gRPC 的應用程式。gRPC 用戶端透過具有 SSL/TLS 加密連線的 HTTP/2 通訊協定連線到 Application Load Balancer。應用程式負載平衡器會將流量轉送至在 Amazon EKS 網繭上執行的 gRPC 應用程式。您可以使用 Kubernetes 水平網繭自動配置器，根據流量自動調整 gRPC 網繭的數目。應用程式負載平衡器的目標群組會在 Amazon EKS 節點上執行運作狀態檢查、評估目標是否運作良好，以及僅將流量轉送至運作良好的節點。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 碼頭, 安裝和配置在 Linux 上, macOS, 或視窗.
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已在 Linux、macOS 或視窗上安裝和設定。

- 例如, 在 Linux 上安裝和配置, macOS 系統, 或視窗.
- `kubectl`、已安裝並設定為存取 Amazon EKS 叢集上的資源。如需詳細資訊，請參閱 Amazon EKS 文件中的安裝或更新 kubectl。
- 安裝和配置的 GrPCurl。
- 全新或現有的 Amazon EKS 叢集。如需詳細資訊，請參閱開始使用 Amazon EKS。
- 設定為存取 Amazon EKS 叢集的電腦終端機。如需詳細資訊，請參閱 Amazon EKS 文件中的設定電腦與叢集通訊。
- AWS Load Balancer 控制器，佈建於 Amazon EKS 叢集中。
- 具有有效 SSL 或 SSL/TLS 憑證的現有 DNS 主機名稱。您可以使用 AWS Certificate Manager (ACM) 或將現有憑證上傳至 ACM 來取得網域的憑證。如需這兩個選項的詳細資訊，請參閱 ACM 文件中的要求公用憑證和將憑證匯入 AWS Certificate Manager。

架構

下圖顯示了這種模式實現的體系結構。

下圖顯示了從將負載卸載到應用程式負載平衡器的 GrPC 用戶端接收 SSL/TLS 流量的工作流程。流量是以明文形式轉送到 GrPC 伺服器，因為它來自虛擬私有雲 (VPC)。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- E@@ lastic Load Balancing 可將傳入的應用程式或網路流量分配到多個目標。例如，您可以在一個或多個可用區域中將流量分配到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。
- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。

工具

- eksctl 是一個簡單的 CLI 工具，用於在 Amazon EKS 上創建集群。
- kubectl 是一個命令列公用程式，可針對 Kubernetes 叢集執行命令。
- AWS Load Balancer 控制器可協助您管理 Kubernetes 叢集的 AWS 彈性負載平衡器。
- GrPCurl 是一個命令列工具，可協助您與 GrPC 服務互動。

代碼存儲庫

此模式的代碼可在 GitHub grpc-traffic-on-alb-to-eks 存儲庫中找到。

史诗

建置 GrPC 伺服器的泊塢視窗映像並推送至 Amazon ECR

任務	描述	所需技能
建立 Amazon ECR 儲存庫。	登入 AWS 管理主控台，開啟 Amazon ECR 主控台，然後建立 Amazon ECR 儲存庫。如需詳細資訊，請參閱 Amazon ECR 文件中的建立儲存庫。確保您記錄了 Amazon ECR 存儲庫的 URL。 您也可以透過執行下列命令，使用 AWS CLI 建立 Amazon ECR 儲存庫： `aws ecr create-repository --repository-name helloworld-grpc`	雲端管理員
建置 Docker 影像。	1. 克隆 GitHub grpc-traffic-on-alb到 eks 存儲庫。 `git clone https://github.com/aws-samp`	DevOps 工程師

任務	描述	所需技能
	`les/grpc-traffic-o n-alb-to-eks.git` 2. 從存儲庫的根目錄中，確保 Docker 文件存在，然後運行以下命令來構建 Docker 映像： `docker build -t <amazon_ecr_reposi tory_url>:<Tag> .` 重要事項：請務必<amazon_ecr_reposi tory_url> 以先前建立之 Amazon ECR 儲存庫的 URL 取代。	

任務	描述	所需技能
將碼頭圖像推送到 Amazon ECR。	1. 執行下列命令以登入 Amazon ECR 儲存庫： `aws ecr get-login -password --region us-east-1 --no-cli-auto-prompt | docker login --username AWS --password-stdin <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com` 2. 執行下列命令，將 Docker 映像推送至 Amazon ECR 儲存庫： `docker push <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/helloworld-grpc:1.0` 重要事項：請務必以 AWS 帳戶 ID 取代`<your_aws_account_id>` 。	DevOps 工程師

將 Kubernetes 資訊清單部署到 Amazon EKS 叢集

任務	描述	所需技能
修改 Kubernetes 資訊清單檔案中的值。	1. 根據您的需求，修改存放`grpc-sample.yaml` 庫 Kubernetes 資料夾中的 Kubernetes 資訊清單檔案。您必須修改輸入資源中的註	DevOps 工程師

任務	描述	所需技能
	釋和主機名稱。如需範例輸入資源，請參閱其他資訊一節。如需有關輸入註釋的詳細資訊，請參閱 Kubernetes 文件中的輸入註釋。 2. 在 Kubernetes 部署資源中，將部署資源變更為您將 Docker 映像推送`image`到的 Amazon ECR 儲存庫的統一資源識別碼 (URI)。如需範例部署資源，請參閱其他資訊一節。	
部署 Kubernetes 資訊清單檔案。	執行下列`kubectl`命令，將`grpc-sample.yaml` 檔案部署到 Amazon EKS 叢集： `kubectl apply -f ./kubernetes/grpc-sample.yaml`	DevOps 工程師

建立應用程式負載平衡器 FQDN 的 DNS 記錄

任務	描述	所需技能
記錄應 Application Load Balancer 的 FQDN。	1. 執行下列`kubectl`命令，以說明管理應用程式負載平衡器的 Kubernetes 輸入資源： `kubectl get ingress -n grpcserver` 範例輸出位於「其他資訊」區段。在輸出中，`HOSTS`欄	DevOps 工程師

任務	描述	所需技能
	位會顯示為其建立 SSL 憑證的 DNS 主機名稱。 2. 從Address輸出欄位記錄應用程式負載平衡器的完整網域名稱 (FQDN)。 3. 建立指向應用程式負載平衡器 FQDN 的 DNS 記錄。如果您的 DNS 提供者是 Amazon Route 53，您可以建立指向應用程式負載平衡器 FQDN 的別名記錄。如需有關此選項的詳細資訊，請參閱 Route 53 說明文件中的別名與非別名記錄之間進行選擇。	

測試解決方案

任務	描述	所需技能
測試 gRPC 伺服器。	透過執行下列命令，使用 GrPCurl 來測試端點： `grpcurl grpc.example.com:443 list grpc.reflection.v1alpha.ServerReflection helloworld.helloworld` 注意：請`grpc.example.com` 以您的 DNS 名稱取代。	DevOps 工程師

任務	描述	所需技能
使用 GrPC 用戶端測試 gRPC 伺服器。	在`helloworld_client_ssl.py` 範例 gRPC 用戶端中，將來自的主機名稱取`grpc.example.com` 代為用於 GrPC 伺服器的主機名稱。 下列程式碼範例顯示 GrPC 伺服器針對用戶端要求的回應： `python ./app/helloworld_client_ssl.py` `message: "Hello to gRPC server from Client"` `message: "Thanks for talking to gRPC server!! Welcome to hello world. Received message is \"Hello to gRPC server from Client\""` `received: true` 這表明客戶端可以與服務器通話，並且連接成功。	DevOps 工程師

清除

任務	描述	所需技能
移除 DNS 記錄。	移除指向您先前建立之應用程式負載平衡器 FQDN 的 DNS 記錄。	雲端管理員
移除負載平衡器。	在 Amazon EC2 主控台上，選擇負載平衡器，然後移除	雲端管理員

任務	描述	所需技能
	Kubernetes 控制器為您的輸入資源建立的負載平衡器。	
刪除 Amazon EKS 叢集。	使eksctl用以下命令刪除 Amazon EKS 叢集： `eksctl delete cluster -f ./eks.yaml`	AWS DevOps

相關資源

- Amazon EKS 上的網絡負載平衡
- 應用程式負載平衡器的目標群組

其他資訊

範例輸入資源：

```
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
    alb.ingress.kubernetes.io/ssl-redirect: "443"
    alb.ingress.kubernetes.io/backend-protocol-version: "GRPC"
    alb.ingress.kubernetes.io/listen-ports: '[{"HTTP": 80}, {"HTTPS":443}]'
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/target-type: ip
    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:<AWS-
Region>:<AccountId>:certificate/<certificate_ID>
    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP
  labels:
    app: grpcserver
    environment: dev
  name: grpcserver
  namespace: grpcserver
```

```
spec:
  ingressClassName: alb
  rules:
  - host: grpc.example.com # <----- replace this as per your host name for which the
 SSL certtficate is available in ACM
    http:
      paths:
      - backend:
          service:
            name: grpcserver
            port:
              number: 9000
        path: /
        pathType: Prefix
```

範例部署資源：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grpcserver
  namespace: grpcserver
spec:
  selector:
    matchLabels:
      app: grpcserver
  replicas: 1
  template:
    metadata:
      labels:
        app: grpcserver
    spec:
      containers:
      - name: grpc-demo
        image: <your_aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/helloworld-
grpc:1.0   #<------- Change to the URI that the Docker image is pushed to
        imagePullPolicy: Always
        ports:
        - name: grpc-api
          containerPort: 9000
        env:
        - name: POD_IP
          valueFrom:
```

```
            fieldRef:
              fieldPath: status.podIP
    restartPolicy: Always
```

輸出範例：

```
NAME              CLASS           HOSTS                           Address
 PORTS          AGE
 grpcserver       <none>      <DNS-HostName>                  <ELB-address>
 80             27d
```

部署和偵錯 Amazon EKS 叢集

由瑞典雷迪斯 (AWS) 和馬修·喬治 (AWS) 創建

環境：PoC 或試點	技術：容器與微服務；基礎架構；現代化；無伺服器；雲端原生	工作負載：所有其他工作
AWS 服務：Amazon EKS；AWS Fargate		

Summary

容器正在成為雲端原生應用程式開發的重要組成部分。Kubernetes 提供了一種有效的方式來管理和協調容器。Amazon Elastic Kubernetes Service (Amazon EKS) 是一項全受管且經過認證的 Kubernetes 一致性服務，可用於在 Amazon Web Services (AWS) 上建置、保護、操作和維護 Kubernetes 叢集。它支援在 AWS Fargate 上執行網繭，以提供隨需、大小適中的運算容量。

對於開發人員和管理員而言，在執行容器化工作負載時，瞭解偵錯選項很重要。此模式會引導您使用 AWS Fargate 在 Amazon EKS 上部署和偵錯容器的過程。其中包括建立、部署、存取、偵錯和清理 Amazon EKS 工作負載。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Identity and Access Management (IAM) 角色設定具有足夠許可，可建立 Amazon EKS、IAM 角色和服務連結角色並與之互動
- 在本機電腦上安裝的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- EKSCTL
- 庫貝克特爾
- 頭盔

限制

- 這種模式為開發人員提供了有用的開發環境調試實踐。它沒有說明生產環境的最佳實踐。
- 如果您正在執行 Windows，請使用作業系統特定的指令來設定環境變數。

使用的產品版本

- AWS CLI 第 2 版
- 在您使用的 Amazon EKS 控制平面的一個次要版本差異之內的 kubectl 版本
- 查看最新版本
- 頭盔第 3 版

架構

技術, 堆

- Application Load Balancer
- Amazon EKS
- AWS Fargate

目標架構

圖表中顯示的所有資源均透過使用`eksctl`和從本機機器發出的`kubectl`命令來佈建。私有叢集必須從私有 VPC 內的執行個體執行。

目標架構由使用 Fargate 啟動類型的 EKS 叢集組成。這可提供隨需、大小適中的運算容量，而不需要指定伺服器類型。EKS 叢集具有控制平面，用於管理叢集節點和工作負載。網繭會佈建到跨多個可用區域的私人 VPC 子網路中。參照 Amazon ECR 公用圖庫，可擷取 NGINX 網路伺服器映像並將其部署到叢集的網繭。

此圖顯示如何使用`kubectl`命令存取 Amazon EKS 控制平面，以及如何使用應用程式負載平衡器存取應用程式。

.

1. AWS 雲端以外的本機機器會將命令傳送到 Amazon EKS 受管 VPC 內的 Kubernetes 控制平面。
2. Amazon EKS 會根據 Fargate 設定檔中的選取器排程網繭。

3. 本機電腦會在瀏覽器中開啟 Application Load Balancer 器 URL。
4. 應用程式負載平衡器會在跨多個可用區域的私有子網路中部署的 Fargate 叢集節點中的 Kubernetes 網繭之間劃分流量。

工具

AWS 服務

- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。此模式也使用 eksctl 命令列工具與 Amazon EKS 上的 Kubernetes 叢集搭配使用。
- AWS Fargate 可協助您執行容器，而不需要管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。它與 Amazon Elastic Container Service (Amazon ECS) 一起使用。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以在一個或多個可用區域中將流量分配到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。佈建 Kubernetes 輸入時，此模式會使用 AWS Load Balancer 控制元件來建立應用程式負載平衡器。應用程式負載平衡器會在多個目標之間分散傳入流量。

其他工具

- Helm 是 Kubernetes 的開源軟件包管理器。在此模式中，Helm 用於安裝 AWS Load Balancer 控制器。
- Kubernetes 是一套開放原始碼系統，可自動化容器化應用程式的部署、擴展和管理。
- NGINX 是一個高性能的 Web 和反向代理服務器。

史诗

建立一個 EKS 叢集

任務	描述	所需技能
建立檔案。	使用「其他資訊」區段中的程式碼，建立下列檔案：	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
	• clusterconfig-fargate.yaml • nginx-deployment.yaml • nginx-service.yaml • nginx-ingress.yaml • index.html	
設定環境變數。	注意：如果命令因為先前未完成的工作而失敗，請等待幾秒鐘，然後再次執行命令。 此模式使用檔案中定義的 AWS 區域和叢集名稱clusterconfig-fargate.yaml 。設定與環境變數相同的值，以便在其他指令中參照它們。 `export AWS_REGION="us-east-1"` `export CLUSTER_NAME="my-fargate"`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
建立 EKS 叢集。	若要建立使用`clusterconfig-fargate.yaml` 檔案規格的 EKS 叢集，請執行下列命令。 `eksctl create cluster -f clusterconfig-fargate.yaml` 該文件包含`ClusterConfig` ，它佈建一個在`us-east-1` 區域`my-fargate-cluster` 中命名的新 EKS 集群和一個默認的 Fargate 配置文件（`fp-default` ）。 默認的 Fargate 配置文件配置有兩個選擇器（`default`和`kube-system` ）。	應用程式開發人員、AWS DevOps、AWS 管理員

任務	描述	所需技能
檢查已建立的叢集。	若要檢查建立的叢集，請執行下列命令。 `eksctl get cluster --output yaml` 輸出應該是以下內容。 `- Name: my-fargate` `  Owned: "True"` `  Region: us-east-1` 使用檢查建立的 Fargate 設定檔。CLUSTER_NAME `eksctl get fargatepr ofile --cluster  $CLUSTER_NAME --output  yaml` 此命令顯示有關資源的信息。您可以使用這些資訊來驗證建立的叢集。輸出應該是以下內容。 `- name: fp-default` `  podExecutionRoleAR N: arn:aws:iam::<YOUR -ACCOUNT-ID>:role/ eksctl-my-fargate- cluster-FargatePod ExecutionRole-xxx` `  selectors:` `  - namespace: default` `  - namespace: kube- system` `  status: ACTIVE` `  subnets:`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	<pre>- subnet-aaa - subnet-bbb - subnet-ccc</pre>	

部署容器

任務	描述	所需技能
部署 NGINX 網頁伺服器。	若要在叢集上套用 NGINX Web 伺服器部署，請執行下列命令。 <pre>kubectl apply -f ./ nginx-deployment.yaml</pre>輸出應該是以下內容。 <pre>deployment.apps/nginx- deployment created</pre>部署包括從 Amazon ECR 公共圖庫擷取的三個 NGINX 映像複本。映像會部署至預設命名空間，並公開在執行中網繭上的連接埠 80 上。	應用程式開發人員、AWS DevOps、AWS 系統管理員
檢查部署和網繭。	(選擇性) 檢查部署。您可以使用以下命令驗證部署的狀態。 <pre>kubectl get deployment</pre>輸出應該是以下內容。	應用程式開發人員、AWS DevOps、AWS 管理員

任務	描述	所需技能
	`NAME READY UP-TO-DATE AVAILABLE AGE nginx-deployment 3/3 3 3 7m14s` 網繭是 Kubernetes 中的可部署物件，其中包含一或多個容器。若要列出所有網繭，請執行下列命令。 `kubectl get pods` 輸出應該是以下內容。 `NAME READY STATUS RESTARTS AGE nginx-deployment-xxxx-aaa 1/1 Running 0 94s nginx-deployment-xxxx-bbb 1/1 Running 0 94s nginx-deployment-xxxx-ccc 1/1 Running 0 94s`	

任務	描述	所需技能
擴展部署。	若要將部署從中指定的三個複本擴展deployment.yaml 到四個複本，請使用下列命令。 `kubectl scale deployment nginx-deployment --replicas 4` 輸出應該是以下內容。 `deployment.apps/nginx-deployment scaled`	應用程式開發人員、AWS DevOps、AWS 系統管理員

部署 AWS Load Balancer 控制器

任務	描述	所需技能
設定環境變數。	描述叢集的 CloudFormation 堆疊，以擷取其 VPC 的相關資訊。 `aws cloudformation describe-stacks --stack-name eksctl-$CLUSTER_NAME-cluster --query "Stacks[0].Outputs[?OutputKey==\`VPC\`].OutputValue"` 輸出應該是以下內容。 `[ "vpc-<YOUR-VPC-ID>" ]`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	複製 VPC ID 並將其匯出為環境變數。 `export VPC_ID="vpc-<YOUR-VPC-ID>"`	
設定叢集服務帳戶的 IAM。	使用舊史詩CLUSTER_NAME 中的AWS_REGION 和，為叢集建立 IAM 開放 ID Connect 提供者。 `eksctl utils associate-iam-oidc-provider \` `--region $AWS_REGION \` `--cluster $CLUSTER_NAME \` `--approve`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
下載並建立 IAM 政策。	下載 AWS Load Balancer 控制器的 IAM 政策，該控制器可讓其代表您撥打 AWS API。 `curl -o iam-policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/main/docs/install/iam_policy.json` 使用 AWS CLI 在您的 AWS 帳戶中建立政策。 `aws iam create-policy \ --policy-name AWSLoadBalancerControllerIAMPolicy \ --policy-document file://iam-policy.json` 您應該會看到下列輸出。 `{ "Policy": { "PolicyName": "AWSLoadBalancerControllerIAMPolicy", "PolicyId": "<YOUR_POLICY_ID>", "Arn": "arn:aws:iam::<YOUR-ACCOUNT-ID>:policy/AWSLoadBalancerControllerIAMPolicy", "Path": "/",`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	<pre>"DefaultV ersionId": "v1", "Attachme ntCount": 0, "Permissi onsBoundaryUsageCo unt": 0, "IsAttachable": true, "CreateDate": "<YOUR-DATE>", "UpdateDate": "<YOUR-DATE>" } }</pre> 將政策的 Amazon 資源名稱（ARN）另存為$POLICY_ARN 。 <pre>export POLICY_AR N=”arn:aws:iam::<Y OUR-ACCOUNT-ID>:po licy/AWSLoadBalanc erControllerIAMPol icy”</pre>	

任務	描述	所需技能
建立 IAM 服務帳戶。	建立命名kube-system 空間aws-load-balancer-controller 中指定的 IAM 服務帳戶。使用CLUSTER_NAME AWS_REGION 、和POLICY_ARN 您先前設定的。 `eksctl create iamserviceaccount \` `--cluster=$CLUSTER_NAME \` `--region=$AWS_REGION \` `--attach-policy-arn=$POLICY_ARN \` `--namespace=kube-system \` `--name=aws-load-balancer-controller \` `--override-existing-serviceaccounts \` `--approve` 確認建立。 `eksctl get iamserviceaccount \` `--cluster $CLUSTER_NAME \` `--name aws-load-balancer-controller \` `--namespace kube-system \` `--output yaml` 輸出應該是以下內容。 `- metadata:`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	<pre> name: aws-load- balancer-controller namespace: kube- system status: roleARN: arn:aws:i am::<YOUR-ACCOUNT- ID>:role/eksctl-my -fargate-addon-iam serviceaccount-ku- Role1-<YOUR-ROLE-ID> wellKnownPolicies: autoScaler: false awsLoadBalancerCon troller: false certManager: false ebsCSIController: false efsCSIController: false externalDNS: false imageBuilder: false</pre>	

任務	描述	所需技能
安裝 AWS Load Balancer 控制器。	更新頭盔儲存庫。 `helm repo update` 將 Amazon EKS 圖表存儲庫添加到掌舵回購。 `helm repo add eks https://aws.github.io/eks-charts` 在背景套用 AWS Load Balancer 控制器 EK 圖表所使用的 Kubernetes 自訂資源定義 (CRD)。 `kubectl apply -k "github.com/aws/eks-charts/stable/aws-load-balancer-controller//crds?ref=master"` 輸出應該是以下內容。 `customresourcedefinition.apiextensions.k8s.io/ingressclassparams.elbv2.k8s.aws created customresourcedefinition.apiextensions.k8s.io/targetgroupbindings.elbv2.k8s.aws created`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	使用您先前設定的環境變數來安裝 Helm 圖表。 `helm install aws-load-balancer-controller eks/aws-load-balancer-controller \ --set clusterName=$CLUSTER_NAME \ --set serviceAccount.create=false \ --set region=$AWS_REGION \ --set vpcId=$VPC_ID \ --set serviceAccount.name=aws-load-balancer-controller \ -n kube-system` 輸出應該是以下內容。 `NAME: aws-load-balancer-controller LAST DEPLOYED: <YOUR-DATE> NAMESPACE: kube-system STATUS: deployed REVISION: 1 TEST SUITE: None NOTES: AWS Load Balancer controller installed!`	

任務	描述	所需技能
創建一個 NGINX 服務。	使用檔案建立服務以公開 NGINX 網繭。`nginx-service.yaml` `kubectl apply -f nginx-service.yaml` 輸出應該是以下內容。 `service/nginx-service created`	應用程式開發人員、AWS DevOps、AWS 系統管理員
建立 Kubernetes 輸入資源。	使用檔案建立服務以公開 Kubernetes NGINX 輸入。`nginx-ingress.yaml` `kubectl apply -f nginx-ingress.yaml` 輸出應該是以下內容。 `ingress.networking.k8s.io/nginx-ingress created`	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
取得負載平衡器 URL。	若要擷取輸入資訊，請使用下列命令。 `kubectl get ingress nginx-ingress` 輸出應該是以下內容。 `NAME CLASS HOSTS ADDRESS PORTS AGE nginx-ingress <none> * k8s-default-nginxing-xxx.us-east-1.elb.amazonaws.com 80 80s` 從輸出中複製ADDRESS（例如，`k8s-default-nginxing-xxx.us-east-1.elb.amazonaws.com`），然後將其粘貼到瀏覽器中以訪問該`index.html` 文件。	應用程式開發人員、AWS DevOps、AWS 系統管理員

偵錯執行中容器

任務	描述	所需技能
選取網繭。	列出所有網繭，並複製所需網繭的名稱。 `kubectl get pods` 輸出應該是以下內容。	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	`NAME READY STATUS RESTARTS AGE nginx-deployment-xxxx-aaa 1/1 Running 0 55m nginx-deployment-xxxx-bbb 1/1 Running 0 55m nginx-deployment-xxxx-ccc 1/1 Running 0 55m nginx-deployment-xxxx-ddd 1/1 Running 0 42m` 此命令會列出現有的網繭和其他資訊。 如果您對特定網繭感興趣，請為POD_NAME變數填入您感興趣的網繭名稱，或將其設定為環境變數。否則，省略此參數以查詢所有資源。 `export POD_NAME="nginx-deployment-<YOUR-POD-NAME>"`	
存取記錄檔。	從您要偵錯的網繭取得記錄檔。 `kubectl logs $POD_NAME`	應用程式開發人員、AWS 系統管理員、AWS DevOps

任務	描述	所需技能
轉發 NGINX 端口。	使用連接埠轉送將用於存取 NGINX Web 伺服器的網繭連接埠對應至本機電腦上的連接埠。 `kubectl port-forward deployment/nginx-deployment 8080:80` 在瀏覽器中，開啟下列 URL。 `http://localhost:8080` 該`port-forward` 命令提供對`index.html` 文件的訪問，而無需通過負載平衡器公開提供該文件。這對於在調試時訪問正在運行的應用程序非常有用。您可以通過按鍵盤命令 Ctrl+ C 來停止端口轉發。	應用程式開發人員、AWS DevOps、AWS 系統管理員
在網繭內執行命令。	要查看當前`index.html` 文件，請使用以下命令。 `kubectl exec $POD_NAME -- cat /usr/share/nginx/html/index.html` 您可以使用命`exec`令直接在網繭中發出任何命令。這對於偵錯執行中的應用程式很有用。	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
將檔案複製到網繭。	移除此網繭上的預設 `index.html` 檔案。 `kubectl exec $POD_NAME -- rm /usr/share/nginx/html/index.html` 將自訂的本機檔案上傳`index.html` 至網繭。 `kubectl cp index.html $POD_NAME:/usr/share/nginx/html/` 您可以使用`cp`命令將檔案直接變更或新增至任何網繭。	應用程式開發人員、AWS DevOps、AWS 系統管理員
使用連接埠轉送來顯示變更。	使用連接埠轉送來驗證您對此網繭所做的變更。 `kubectl port-forward pod/$POD_NAME 8080:80` 在瀏覽器中打開以下 URL。 `http://localhost:8080` 套用至`index.html` 檔案的變更應在瀏覽器中可見。	應用程式開發人員、AWS DevOps、AWS 系統管理員

刪除資源

任務	描述	所需技能
刪除負載平衡器。	刪除輸入。 `kubectl delete ingress/nginx-ingress` 輸出應該是以下內容。 `ingress.networking.k8s.io "nginx-ingress" deleted` 刪除服務。 `kubectl delete service/nginx-service` 輸出應該是以下內容。 `service "nginx-service" deleted` 刪除負載平衡器控制器。 `helm delete aws-load-balancer-controller -n kube-system` 輸出應該是以下內容。 `release "aws-load-balancer-controller" uninstalled` 刪除服務帳戶。	應用程式開發人員、AWS DevOps、AWS 系統管理員

任務	描述	所需技能
	`eksctl delete iamservic eaccount --cluster $CLUSTER_NAME -- namespace kube-syst em --name aws-load- balancer-controller`	
刪除部署。	若要刪除部署資源，請使用下列命令。 `kubectl delete deploy/ng inx-deployment` 輸出應該是以下內容。 `deployment.apps "nginx- deployment" deleted`	應用程式開發人員、AWS DevOps、AWS 系統管理員
刪除叢集。	使用下列命令刪除 EKS 叢集，其中`my-fargate` 是叢集名稱。 `eksctl delete cluster -- name $CLUSTER_NAME` 此命令會刪除整個叢集，包括所有關聯的資源。	應用程式開發人員、AWS DevOps、AWS 系統管理員
刪除 IAM 政策。	使用 AWS CLI 刪除先前建立的政策。 `aws iam delete-policy --policy-arn $POLICY_A RN`	應用程式開發人員、AWS 管理員、AWS DevOps

故障診斷

問題	解決方案
您會在叢集建立時收到錯誤訊息，指出您的目標可用區域沒有足夠的容量來支援叢集。您應該會看到類似下列內容的訊息。 `Cannot create cluster 'my-fargate' because us-east-1e, the targeted availability zone, does not currently have sufficient capacity to support the cluster. Retry and choose from these availability zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1f`	使用錯誤訊息中建議的可用區域再次建立叢集。在`clusterconfig-fargate.yaml` 檔案的最後一行指定可用區域清單 (例如，`availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]`)。

相關資源

- Amazon EKS 文檔
- Amazon EKS 上的應用程式負載平衡
- EKS 最佳做法指南
- AWS Load Balancer 控制器文件
- 查看文件
- Amazon ECR 公共畫廊 NGINX 圖片
- 頭盔文件
- 偵錯執行中的網繭 (Kubernetes 文件)
- Amazon EKS 工作坊
- EKS 叢集建立錯誤

其他資訊

群集配置-遠程. 亞姆

```
apiVersion: eksctl.io/v1alpha5
```

```
kind: ClusterConfig

metadata:
  name: my-fargate
  region: us-east-1

fargateProfiles:
  - name: fp-default
    selectors:
      - namespace: default
      - namespace: kube-system
```

尼金克斯部署. 羊

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: "nginx-deployment"
  namespace: "default"
spec:
  replicas: 3
  selector:
    matchLabels:
      app: "nginx"
  template:
    metadata:
      labels:
        app: "nginx"
    spec:
      containers:
      - name: nginx
        image: public.ecr.aws/nginx/nginx:latest
        ports:
        - containerPort: 80
```

尼金克斯服務.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    alb.ingress.kubernetes.io/target-type: ip
  name: "nginx-service"
```

```
  namespace: "default"
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
  type: NodePort
  selector:
    app: "nginx"
```

尼金克斯含量. 羊

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  namespace: "default"
  name:  "nginx-ingress"
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
spec:
  rules:
    - http:
        paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: "nginx-service"
              port:
                number: 80
```

index.html

```
<!DOCTYPE html>
<html>

<body>
    <h1>Welcome to your customized nginx!</h1>
    <p>You modified the file on this running pod</p>
</body>
```

```
</html>
```

使用 Elastic Beanstalk 部署容器

由托馬斯·斯科特（AWS）和讓·巴蒂斯特·吉盧瓦（AWS）創建

程式碼儲存庫：叢集範例應用	環境：生產	技術：容器與微服務；雲端原生；現代化
AWS 服務：AWS Elastic Beanstalk		

Summary

在 Amazon Web Services (AWS) 雲端上，AWS Elastic Beanstalk 支援 Docker 做為可用平台，讓容器可以在建立的環境中執行。此模式顯示如何使用 Elastic Beanstalk 服務部署容器。此模式的部署將使用基於 Docker 平台的 Web 服務器環境。

若要使用 Elastic Beanstalk 部署和擴展 Web 應用程式和服務，您必須上傳程式碼，並自動處理部署作業。此外，還包括容量佈建、負載平衡、自動調整規模和應用程式健康狀態監控。當您使用 Elastic Beanstalk 時，您可以完全控制它代表您建立的 AWS 資源。使用 Elastic Beanstalk 並不收取其他費用。您只需為用於存放和執行應用程式的 AWS 資源付費。

此模式包括使用 AWS Elastic Beanstalk 命令列界面 (EB CLI) 和 AWS 管理主控台進行部署的說明。

使用案例

Elastic Beanstalk 的使用案例包括以下內容：

- 部署原型環境以示範前端應用程式。（此模式使用 Docker 文件作為示例。）
- 部署 API 以處理指定網域的 API 要求。
- 使用 Docker 撰寫部署協調解決方案（`docker-compose.yml`不作為此模式中的實際示例）。

先決條件和限制

先決條件

- 一個 AWS 帳戶
- AWS EB CLI 已在本機安裝

- 在本地計算機上安裝的泊塢窗

限制

- 在免費計劃中，每個 IP 地址每 6 小時有 100 次提取的 Docker 提取限制。

架構

目標技術堆疊

- Amazon Elastic Compute Cloud (Amazon EC2) 執行個體
- 安全群組
- Application Load Balancer
- Auto Scaling 群組

目標架構

自動化和規模

AWS Elastic Beanstalk 可以根據發出的請求數量自動擴展。為環境建立的 AWS 資源包括一個 Application Load Balancer、一個 Auto Scaling 群組，以及一或多個 Amazon EC2 執行個體。

負載平衡器位於 Amazon EC2 執行個體前面，這是 Auto Scaling 群組的一部分。Amazon EC2 Auto Scaling 會自動啟動額外的 Amazon EC2 執行個體，以容納您的應用程式增加的負載。如果應用程式的負載減少，Amazon EC2 Auto Scaling 會停止執行個體，但會保持至少一個執行個體的執行中。

自動縮放觸發

Elastic Beanstalk 環境中的 Auto Scaling 組使用兩個 Amazon CloudWatch 警報來啟動擴展操作。當每個執行個體的平均傳出網路流量，在五分鐘期間高於 6 MB 或低於 2 MB 時，預設的觸發條件就會擴展。如要有效地使用 Amazon EC2 Auto Scaling，請根據您的應用程式、執行個體類型和服務需求，設定適用的觸發。您可以根據多項統計資料來進行擴展，包括延遲、磁碟 I/O、CPU 使用率和請求計數。如需詳細資訊，請參閱 Auto Scaling 觸發器。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS EB 命令列界面 (EB CLI) 是一種命令列用戶端，可用來建立、設定和管理 Elastic Beanstalk 環境。
- E@@ lastic Load Balancing 可將傳入的應用程式或網路流量分配到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。

其他服務

- Docker 將軟體封裝成稱為容器的標準化單元，其中包括程式庫、系統工具、程式碼和執行階段。

Code

此模式的程式碼可在 GitHub 叢集範例應用程式存放庫中取得。

史诗

使用碼頭文件構建

任務	描述	所需技能
克隆遠程存儲庫。	• 若要複製存放庫，請執行指令`git clone https://github.com/aws-samples/cluster-sample-app.git 。 </p>`	應用程式開發人員、AWS 管理員、AWS DevOps
初始化 Elastic Beanstalk 泊塢窗項目。	1. 創建一個名為`aws.json`根目錄的文件。 2. 在`aws.json`檔案中，新增下列程式碼。 `{ "AWSEBDockerrunVersion":"1", "Image":{ "Name":"cluster-sample-app"`	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
	<pre>}, "Ports":[{ "ContainerPort":80 , "HostPort":8080 }] }</pre> 3. `eb init -p docker` 在專案的根目錄執行命令。	
在本機測試專案。	1. `eb local run`在專案的根目錄執行命令。 2. 瀏覽至以測試應用程式`http://localhost` 。	應用程式開發人員、AWS 管理員、AWS DevOps

使用 EB CLI 進行部署

任務	描述	所需技能
執行部署命令	1. `eb create docker-sample-cluster-app` 在專案的根目錄執行命令。	應用程式開發人員、AWS 管理員、AWS DevOps
存取已部署的版本。	部署指令完成後，請使用`eb open`指令存取專案。	應用程式開發人員、AWS 管理員、AWS DevOps

使用主控台進行部署

任務	描述	所需技能
使用瀏覽器部署應用程式。	1. 開啟 主控台。 2. 導航到 Elastic Beanstalk 控制台。 3. 選擇建立應用程式。 4. 針對「應用程式名稱」，輸入叢集範例應用程式。 5. 選擇泊塢工人作為平台。 6. 選擇 [上傳程式碼]。 7. 選擇您的本機 .zip 檔案 (位於複製專案的根目錄中) 或公用的 Amazon Simple Storage Service (Amazon S3) URL。	應用程式開發人員、AWS 管理員、AWS DevOps
存取已部署的版本。	部署後，存取已部署的應用程式，然後選擇提供的 URL。	應用程式開發人員、AWS 管理員、AWS DevOps

相關資源

- 網頁伺服器環境
- 在 macOS 上安裝 EB CLI
- 手動安裝 EB CLI

其他資訊

使用 Elastic Beanstalk 的優點

- 自動化基礎架構
- 基礎平台的自動管理
- 自動修補和更新以支援應用程式

- 應用程式的自動縮放
- 能夠自定義節點的數量
- 如有需要，可存取基礎架構元件
- 比其他容器部署解決方案更容易部署

使用 Lambda 函數、Amazon VPC 和無伺服器架構產生靜態輸出 IP 地址

創建者湯瑪斯·斯科特 (AWS)

環境：生產	技術：容器與微服務；軟體開發與測試	AWS 服務：AWS Lambda

Summary

此模式說明如何使用無伺服器架構在 Amazon Web Services (AWS) 雲端中產生靜態輸出 IP 地址。如果您的組織想要使用安全檔案傳輸通訊協定 (SFTP) 將檔案傳送至個別的業務實體，就可以從這個方法中受益。這表示企業實體必須能夠存取允許檔案透過其防火牆的 IP 位址。

該模式的方法可協助您建立使用彈性 IP 地址作為輸出 IP 地址的 AWS Lambda 函數。透過遵循此模式中的步驟，您可以建立 Lambda 函數和虛擬私有雲 (VPC)，透過具有靜態 IP 位址的網際網路閘道路由輸出流量。若要使用靜態 IP 位址，請將 Lambda 函數附加至 VPC 及其子網路。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Identity and Access Management (IAM) 許可，可建立和部署 Lambda 函數，以及建立 VPC 及其子網路。如需詳細資訊，請參閱 AWS Lambda 文件中的執行角色和使用者許可。
- 如果您計劃使用基礎設施即程式碼 (IaC) 來實作此模式的方法，則需要整合式開發環境 (IDE)，例如 AWS Cloud9。如需這方面的詳細資訊，請參閱什麼是 AWS Cloud9？ 在 AWS Cloud9 文件中。

架構

下圖顯示此模式的無伺服器架構。

該圖顯示以下工作流程：

1. 出站流量離NAT gateway 1開Public subnet 1。

2. 出站流量離NAT gateway 2開Public subnet 2。
3. Lambda 函數可以在Private subnet 1或中執行Private subnet 2。
4. Private subnet 1並將流量Private subnet 2路由到公用子網路中的 NAT 閘道。
5. NAT 閘道會從公用子網路將輸出流量傳送至網際網路閘道。
6. 輸出數據從互聯網網關傳輸到外部服務器。

技術堆疊

- Lambda
- Amazon Virtual Private Cloud (Amazon VPC)

自動化和規模

您可以在不同的可用區域中使用兩個公用子網路和兩個私有子網路，以確保高可用性 (HA)。即使有一個可用區域無法使用，模式的解決方案仍會繼續運作。

工具

- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 佈建 AWS 雲端的邏輯隔離部分，您可以在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 AWS 可擴展基礎設施的優勢。

史诗

建立新 VPC

任務	描述	所需技能
建立新 VPC	登入 AWS 管理主控台，開啟 Amazon VPC 主控台，然後建	AWS 管理員

任務	描述	所需技能
	立名10.0.0.0/25 為 IPv4 CIDR 範圍Lambda VPC的 VPC。 如需有關建立 VPC 的詳細資訊，請參閱 Amazon VPC 文件中的開始使用 Amazon VPC。	

建立兩個公用子網路

任務	描述	所需技能
建立第一個公用子網路。	1. 在 Amazon VPC 主控台上，選擇「子網路」，然後選擇「建立子網路」。 2. 針對名稱標籤，輸入 public-one 。 3. 對於 VPC，請選擇 Lambda VPC。 4. 選擇可用區域並進行記錄。 5. 對於 IPv4 CIDR 區塊，請輸入，10.0.0.0/28 然後選擇 [建立子網路]。	AWS 管理員
建立第二個公用子網路。	1. 在 Amazon VPC 主控台上，選擇「子網路」，然後選擇「建立子網路」。 2. 針對名稱標籤，輸入 public-two 。 3. 對於 VPC，請選擇 Lambda VPC。 4. 選擇可用區域並進行記錄。重要：您無法使用包	AWS 管理員

任務	描述	所需技能
	含public-one 子網路的可用區域。 5. 對於 IPv4 CIDR 區塊，請輸入，10.0.0.16/28 然後選擇 [建立子網路]。	

建立兩個私有子網路

任務	描述	所需技能
建立第一個私有子網路。	1. 在 Amazon VPC 主控台上，選擇「子網路」，然後選擇「建立子網路」。 2. 針對名稱標籤，輸入 private-one 。 3. 對於 VPC，請選擇 Lambda VPC。 4. 選擇包含您先前建立之public-one 子網路的可用區域。 5. 對於 IPv4 CIDR 區塊，請輸入，10.0.0.32/28 然後選擇 [建立子網路]。	AWS 管理員
建立第二個私有子網路。	1. 在 Amazon VPC 主控台上，選擇「子網路」，然後選擇「建立子網路」。 2. 針對名稱標籤，輸入 private-two 。 3. 對於 VPC，請選擇 Lambda VPC。	AWS 管理員

任務	描述	所需技能
	4. 選擇包含您先前建立之public-two 子網路的相同可用區域。 5. 對於 IPv4 CIDR 區塊，請輸入，10.0.0.64/28 然後選擇 [建立子網路]。	

為 NAT 閘道建立兩個彈性 IP 位址

任務	描述	所需技能
建立第一個彈性 IP 位址。	1. 在 Amazon VPC 主控台上，選擇彈性 IP，然後選擇配置新地址。 2. 選擇配置並記錄新建立的彈性 IP 位址的配置 ID。 備註：此彈性 IP 位址用於您的第一個 NAT 閘道。	AWS 管理員
建立第二個彈性 IP 位址。	1. 在 Amazon VPC 主控台上，選擇彈性 IP，然後選擇配置新地址。 2. 選擇配置並記錄第二個彈性 IP 位址的配置 ID。 備註：此彈性 IP 位址用於您的第二個 NAT 閘道。	AWS 管理員

建立網際網路閘道

任務	描述	所需技能
建立網際網路閘道	1. 在 Amazon VPC 主控台上，選擇 [網際網路閘道]，然後選擇 [建立網際網路閘道]。 2. 輸入名`Lambda internet gateway`稱，然後選擇 [建立網際網路閘道]。請確定您已記錄網際網路閘道 ID。	AWS 管理員
將網際網路閘道連接至 VPC。	選取您剛建立的網際網路閘道，然後選擇 Actions, Attach to VPC (動作、連接到 VPC)。	AWS 管理員

建立兩個 NAT 閘道

任務	描述	所需技能
建立第一個 NAT 閘道。	1. 在 Amazon VPC 主控台上，選擇「NAT 閘道」，然後選擇「建立 NAT 閘道」。 2. 輸`nat-one`入 NAT 閘道名稱。 3. 選擇`public-one` 作為要在其中建立 NAT 閘道的子網路。 4. 對於連線類型，選擇公用。 5. 針對彈性 IP 配置識別碼，請選擇您先前建立的第一個彈性 IP 位址，並將其與 NAT 閘道建立關聯。	AWS 管理員

任務	描述	所需技能
	6. 選擇建立 NAT 閘道。	
建立第二個 NAT 閘道。	1. 在 Amazon VPC 主控台上，選擇「NAT 閘道」，然後選擇「建立 NAT 閘道」。 2. 輸nat-two入 NAT 閘道名稱。 3. 選擇public-two 作為要在其中建立 NAT 閘道的子網路。 4. 對於連線類型，選擇公用。 5. 針對彈性 IP 配置識別碼，請選擇您先前建立的第二個彈性 IP 位址，並將其與 NAT 閘道建立關聯。 6. 選擇建立 NAT 閘道。	AWS 管理員

為您的公用和私有子網路建立路由表

任務	描述	所需技能
建立公用一子網路的路由表。	1. 在 Amazon VPC 主控台上，選擇「路由表」，然後選擇「建立路由表」。 2. 輸入public-one-subnet作為路由表名稱，然後選擇建立路由表格。 3. 選擇public-one-subnet路由表格，選擇編輯路由，然後選擇新增路由。	AWS 管理員

任務	描述	所需技能
	4. 0.0.0.0在 [目的地] 方塊中指定，然後在 [目標] 清單中選擇網際網路閘道 ID。 5. 在 [子網路關聯] 索引標籤上，選擇 [編輯子網路關聯]，選擇具有 10.0.0.0/28 CIDR 範圍的public-one 子網路，然後選擇 [儲存關聯]。 6. 選擇 Save Changes (儲存變更)。	
建立公用兩個子網路的路由表。	1. 在 Amazon VPC 主控台上，選擇「路由表」，然後選擇「建立路由表」。 2. 輸入public-two-subnet作為路由表名稱，然後選擇建立路由表格。 3. 選擇public-two-subnet路由表格，選擇編輯路由，然後選擇新增路由。 4. 0.0.0.0在 [目的地] 方塊中指定，然後在 [目標] 清單中選擇網際網路閘道 ID。 5. 在 [子網路關聯] 索引標籤上，選擇 [編輯子網路關聯]，選擇具有 10.0.0.16/28 CIDR 範圍的public-two 子網路，然後選擇 [儲存關聯]。 6. 選擇 Save Changes (儲存變更)。	AWS 管理員

任務	描述	所需技能
為私有一個子網路建立路由表。	1. 在 Amazon VPC 主控台上，選擇「路由表」，然後選擇「建立路由表」。 2. 輸入`private-one-subnet` 作為路由表名稱，然後選擇建立路由表格。 3. 選擇`private-one-subnet` 路由表格，選擇編輯路由，然後選擇新增路由。 4. `0.0.0.0`在「目的地」方塊中指定，然後在「目標」清單中選擇`public-one` 子網路中的 NAT 閘道。 5. 在 [子網路關聯] 索引標籤上，選擇 [編輯子網路關聯]，選擇具有 `10.0.0.32/28` CIDR 範圍的`private-one` 子網路，然後選擇 [儲存關聯]。 6. 選擇 Save Changes (儲存變更)。	AWS 管理員

任務	描述	所需技能
建立私用兩個子網路的路由表。	1. 在 Amazon VPC 主控台上，選擇「路由表」，然後選擇「建立路由表」。 2. 輸入`private-two-subnet` 作為路由表名稱，然後選擇建立路由表格。 3. 選擇`private-two-subnet` 路由表格，選擇編輯路由，然後選擇新增路由。 4. `0.0.0.0`在「目的地」方塊中指定，然後在「目標」清單中選擇`public-two` 子網路中的 NAT 閘道。 5. 在 [子網路關聯] 索引標籤上，選擇 [編輯子網路關聯]，選擇具有 `10.0.0.64/28` CIDR 範圍的`private-two` 子網路，然後選擇 [儲存關聯]。 6. 選擇 Save Changes (儲存變更)。	AWS 管理員

建立 Lambda 函數，將其新增至 VPC，然後測試解決方案

任務	描述	所需技能
建立新 Lambda 函數。	1. 開啟 AWS Lambda 主控台，然後選擇建立函數。 2. 在 [基本資訊] `Lambda test` 下，輸入 [函數名稱] 底下，然後在 [執行階段] 下選擇您選擇的語言。	AWS 管理員

任務	描述	所需技能
	3. 選擇建立函數。	
將 Lambda 函數新增至您的 VPC。	1. 在 AWS Lambda 主控台上，選擇「函數」，然後選擇您先前建立的函數。 2. 選擇 Configuration (組態)，然後選擇 VPC。 3. 選擇 [編輯]，然後選擇Lambda VPC和兩個私人子網路。 4. 選擇用於測試的預設安全性群組，然後選擇 [儲存]。	AWS 管理員
撰寫程式碼以呼叫外部服務。	1. 使用您選擇的程式設計語言，撰寫程式碼以呼叫會傳回 IP 位址的外部服務。 2. 確認傳回的 IP 位址是否符合其中一個彈性 IP 位址。	AWS 管理員

相關資源

- 設定 Lambda 函數以存取 VPC 中的資源

使用 Kubernetes 在 Amazon EKS 工作者節點上安裝 SSM 代理程式 DaemonSet

創建者：馬亨德拉·西達帕 (AWS)

環境：PoC 或試點	技術：容器與微服務 DevOps；基礎架構	AWS 服務：Amazon EKS；AWS Systems Manager

Summary

請注意，2021 年 9 月：最新的 Amazon EKS 最佳化 AMI 會自動安裝 SSM 代理程式。如需詳細資訊，請參閱 2021 年 6 月 AMI 的發行說明。

在 Amazon Elastic Kubernetes Service (Amazon EKS) 中，基於安全準則的原因，工作者節點沒有附加安全殼層 (SSH) 金鑰對。此模式顯示如何使用 Kubernetes DaemonSet 資源類型在所有工作節點上安裝 AWS Systems Manager 代理程式 (SSM 代理程式)，而不需手動安裝或取代節點的 Amazon 機器映像 (AMI)。 DaemonSet 使用背景工作節點上的 cron 工作來排程 SSM 代理程式的安裝。您也可以使用此模式在 Worker 節點上安裝其他套件。

當您對叢集中的問題進行疑難排解時，依需求安裝 SSM Agent 可讓您在不使用安全殼層金鑰配對的情況下與 Worker 節點建立 SSH 工作階段、收集記錄或查看執行個體設定。

先決條件和限制

先決條件

- 具有亞馬遜彈性運算雲端 (Amazon EC2) 工作者節點的現有 Amazon EKS 叢集。
- 容器執行個體應具有與 SSM 服務通訊的必要權限。AWS Identity and Access Management (IAM) 受管角色 AmazonSSM ManagedInstanceCore 提供所需的許可，讓 SSM 代理程式在 EC2 執行個體上執行。如需詳細資訊，請參閱 AWS Systems Manager 文件。

限制

- 此模式不適用於 AWS Fargate，因為 Fargate 平台 DaemonSets 不支援。
- 此模式僅適用於以 Linux 為基礎的工作者節點。

- DaemonSet 網繭會以特殊權限模式執行。如果 Amazon EKS 叢集具有以特殊權限模式封鎖網繭的 Webhook，將不會安裝 SSM 代理程式。

架構

下圖說明此模式的架構。

工具

工具

- kubectl 是用來與 Amazon EKS 叢集互動的命令列公用程式。此模式用`kubectl`於在 Amazon EKS 叢集 DaemonSet 上部署，該叢集將在所有工作節點上安裝 SSM 代理程式。
- Amazon EKS 可讓您輕鬆在 AWS 上執行 Kubernetes，而無需安裝、操作和維護自己的 Kubernetes 控制平面或節點。Kubernetes 是一套開放原始碼系統，用於容器化應用程式的自動化部署、擴展與管理。
- AWS Systems Manager 工作階段管理員可讓您透過互動式的一鍵式瀏覽器型殼層或 AWS Command Line Interface (AWS CLI) (AWS CLI) 管理 EC2 執行個體、現場部署執行個體和虛擬機器 (VM)。

Code

使用下列程式碼建立將在 Amazon EKS 叢集上安裝 SSM 代理程式的 DaemonSet 組態檔案。請按照「史詩」部分中的說明進行操作。

```
cat << EOF > ssm_daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  labels:
    k8s-app: ssm-installer
  name: ssm-installer
  namespace: kube-system
spec:
  selector:
    matchLabels:
      k8s-app: ssm-installer
  template:
```

```
    metadata:
      labels:
        k8s-app: ssm-installer
    spec:
      containers:
      - name: sleeper
        image: busybox
        command: ['sh', '-c', 'echo I keep things running! && sleep 3600']
      initContainers:
      - image: amazonlinux
        imagePullPolicy: Always
        name: ssm
        command: ["/bin/bash"]
        args: ["-c","echo '* * * * * root yum install -y https://s3.amazonaws.com/
ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm & rm -rf /etc/
cron.d/ssmstart' > /etc/cron.d/ssmstart"]
        securityContext:
          allowPrivilegeEscalation: true
        volumeMounts:
        - mountPath: /etc/cron.d
          name: cronfile
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      volumes:
      - name: cronfile
        hostPath:
          path: /etc/cron.d
          type: Directory
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      terminationGracePeriodSeconds: 30
EOF
```

史诗

設置庫貝克特爾

任務	描述	所需技能
安裝並設定 kubectl 以存取 EKS 叢集。	如果尚kubectl未安裝並設定為存取 Amazon EKS 叢集，請	DevOps

任務	描述	所需技能
	參閱 Amazon EKS 說明文件中的安裝 kubectl。	

部署 DaemonSet

任務	描述	所需技能
建立組 DaemonSet 態檔案。	使用此模式稍早「程式碼」區段中的程式碼建立名為的 DaemonSet 組態檔案`ssm_daemonset.yaml`，該檔案將部署至 Amazon EKS 叢集。 由啟動的網繭 DaemonSet 具有主容器和`init`容器。主容器有一個`sleep`命令。`init`容器包含一個`command`區段，可建立 Cron 工作檔案，以便在路徑上安裝 SSM 代理程式。`/etc/cron.d/` Cron 工作只會執行一次，並在工作完成後自動刪除它建立的檔案。 init 容器完成後，主容器會等待 60 分鐘，然後再結束。60 分鐘後，會啟動新的 Pod。此網繭會安裝 SSM 代理程式 (如果遺失)，或將 SSM 代理程式更新為最新版本。 如果需要，您可以修改命`sleep`令以每天重新啟動網繭一次或更頻繁地執行。	DevOps

任務	描述	所需技能
在 Amazon EKS 叢集 DaemonSet 上部署。	若要在 Amazon EKS 叢集上部署您在上一個步驟中建立的 DaemonSet 組態檔案，請使用下列命令： `kubectl apply -f ssm_daemonset.yaml` 此命令會建立一個 DaemonSet 在背景工作節點上執行網繭以安裝 SSM 代理程式。	DevOps

相關資源

- 安裝庫貝克特爾（Amazon EKS 文檔）
- 設定工作階段管理員 (AWS Systems Manager 文件)

在 Amazon EKS 工作者節點上安裝 SSM CloudWatch 代理程式和代理程式 preBootstrapCommands

創建者：阿卡瑪哈德維哈德維（AWS）

環境：生產	技術：容器與微服務；基礎設施；營運	AWS 服務：Amazon EKS; AWS Systems Manager; Amazon CloudWatch

Summary

此模式提供程式碼範例和步驟，以便在 Amazon EKS 叢集建立期間，在 Amazon Amazon Web Services 服務 (AWS) 雲 CloudWatch 端上在 Amazon Elastic Kubernetes Service (Amazon EKS) 工作者節點上安裝 AWS Systems Manager 代理程式 (SSM 代理程式) 和亞馬遜代理程式。您可以使用`eksctl`組態檔結構描述 (Weaveworks 文件) 中的`preBootstrapCommands`屬性來安裝 SSM CloudWatch 代理程式和代理程式。然後，您可以使用 SSM 代理程式連線到工作者節點，而無需使用 Amazon Elastic Compute Cloud (Amazon EC2) key pair。此外，您可以使用 CloudWatch 代理程式監控 Amazon EKS 工作者節點上的記憶體和磁碟使用率。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 macOS、Linux 或視窗上安裝和配置的 eksctl 命令列公用程式
- 在 macOS、Linux 或視窗上安裝和設定的命令列公用程式

限制

- 建議您避免將長時間執行的指令碼新增至`preBootstrapCommands`屬性，因為這會在擴展活動期間延遲節點加入 Amazon EKS 叢集。我們建議您改為建立自訂的 Amazon 機器映像 (AMI)。
- 此模式僅適用於 Amazon EC2 Linux 執行個體。

架構

技術, 堆

- Amazon CloudWatch
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Systems Manager 參數存放區

目標架構

下圖顯示使用者使用安裝的 SSM 代理程式連線至 Amazon EKS 工作者節點的範例。`preBootstrapCommands`

該圖顯示以下工作流程：

1. 使用者使用具有安裝 SSM 代理程式和 CloudWatch 代理程式的`preBootstrapCommands`屬性的`eksctl`組態檔案來建立 Amazon EKS 叢集。
2. 任何稍後因擴展活動而加入叢集的新執行個體，都會使用預先安裝的 SSM 代理程式和 CloudWatch 代理程式建立。
3. 使用者使用 SSM 代理程式連線至 Amazon EC2，然後使用代理程式監控記憶體和磁碟使用 CloudWatch 率。

工具

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- AWS Systems Manager Parameter Store 為組態資料管理和機密管理提供安全的階層式儲存。
- AWS Systems Manager 工作階段管理員透過互動式的一鍵式瀏覽器型殼層或 AWS Command Line Interface (AWS CLI) (AWS CLI)，協助您管理 EC2 執行個體、現場部署執行個體和虛擬機器。
- eksctl 是一個命令列公用程式，用於在 Amazon EKS 上建立和管理 Kubernetes 叢集。
- kubectl 是用於與叢集 API 伺服器通訊的命令列公用程式。

史诗

創建一個 Amazon EKS 集群

任務	描述	所需技能
儲存代 CloudWatch 理程式組態檔。	將 CloudWatch 代理程式組態檔案存放在您想要建立 Amazon EKS 叢集的 AWS 區域中的 AWS Systems Manager Parameter Store 中。若要這麼做，請在 AWS Systems Manager Parameter Store 中建立參數，並記下參數的名稱 (例如，`AmazonCloudwatch-linux`)。 如需詳細資訊，請參閱此病毒碼的其他資訊一節中的範例 CloudWatch 代理程式組態檔案程式碼。	DevOps 工程師
建立 eksctl 配置檔案和叢集。	1. 建立包含 CloudWatch 代理程式和 SSM 代理程式安裝步驟的`eksctl`組態檔。如需詳細資訊，請參閱此模式的其他資訊一節中的範例 eksctl 組態檔案程式碼。 2. 執行`eksctl create cluster -f cluster.yaml` 指令以建立叢集。	AWS DevOps

確認 SSM 代理程式和 CloudWatch 代理程式是否正常運作

任務	描述	所需技能
測試 SSM 代理程式。	使用 SSH 連線到 Amazon EKS 叢集節點，方法是使用 AWS Systems Manager 文件開始工作階段中涵蓋的任何方法。	AWS DevOps
測試代 CloudWatch 理程式。	使用主 CloudWatch 控台驗證 CloudWatch 代理程式： 1. 登入 AWS 管理主控台，並開啟 CloudWatch 主控台。 2. 在導覽窗格中，展開 [量度]，然後選擇 [所有量度]。 3. 在「瀏覽」索引標籤的搜尋方塊中，輸入並選擇 CWAgent 度量，以查看記憶體和磁碟度量。	AWS DevOps

相關資源

- 在您的伺服器上安裝和執行 CloudWatch 代理程式 (Amazon CloudWatch 文件)
- 建立 Systems Manager 參數 (主控台) (AWS Systems Manager 文件)
- 建立 CloudWatch 代理程式組態檔案 (Amazon CloudWatch 文件)
- 開始工作階段 (AWS CLI) (AWS Systems Manager 文件)
- 啟動工作階段 (Amazon EC2 主控台) (AWS Systems Manager 文件)

其他資訊

範例 CloudWatch 代理程式組態檔

在下列範例中， CloudWatch 代理程式設定為監控 Amazon Linux 執行個體上的磁碟和記憶體使用率：

```
{
    "agent": {
        "metrics_collection_interval": 60,
        "run_as_user": "cwagent"
    },
    "metrics": {
        "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
            "ImageId": "${aws:ImageId}",
            "InstanceId": "${aws:InstanceId}",
            "InstanceType": "${aws:InstanceType}"
        },
        "metrics_collected": {
            "disk": {
                "measurement": [
                    "used_percent"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "mem": {
                "measurement": [
                    "mem_used_percent"
                ],
                "metrics_collection_interval": 60
            }
        }
    }
}
```

設定檔案範例

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: test
  region: us-east-2
  version: "1.24"
managedNodeGroups:
  - name: test
    minSize: 2
```

```
    maxSize: 4
    desiredCapacity: 2
    volumeSize: 20
    instanceType: t3.medium
    preBootstrapCommands:
    - sudo yum install amazon-ssm-agent -y
    - sudo systemctl enable amazon-ssm-agent
    - sudo systemctl start amazon-ssm-agent
    - sudo yum install amazon-cloudwatch-agent -y
    - sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c ssm:AmazonCloudwatch-linux
    iam:
      attachPolicyARNs:
        - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
        - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
        - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
        - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

其他代碼詳細信

- 在preBootstrapCommands屬性的最後一行，AmazonCloudwatch-linux是在 AWS 系統管理員參數存放區中建立的參數名稱。您必須AmazonCloudwatch-linux在建立 Amazon EKS 叢集的相同 AWS 區域中加入參數存放區。您也可以指定檔案路徑，但我們建議您使用 Systems Manager，以便於自動化和重複使用。
- 如果您在eksctl組態檔preBootstrapCommands中使用，則會在 AWS 管理主控台中看到兩個啟動範本。第一個啟動範本包括中指定的指令preBootstrapCommands。第二個範本包括在 Amazon EKS 使用者資料中指定的命令preBootstrapCommands和預設值。需要此數據才能獲得加入集群的節點。節點群組的 Auto Scaling 群組會使用此使用者資料來啟動新執行個體。
- 如果您在eksctl組態檔案中使用iam屬性，則必須列出預設的 Amazon EKS 政策，以及附加的 AWS Identity and Access Management (IAM) 政策中所需的任何其他政策。在 [建立 eksctl 組態檔和叢集] 步驟的程式碼片段中，*CloudWatchAgentServerPolicy*並AmazonSSMMangedInstanceCore新增其他原則，以確保代 CloudWatch 理程式和 SSM 代理程式能如預期般運作。AmazonEKSWorkerNodePolicy、AmazonEKS_CNI_Policy、AmazonEC2ContainerRegistryR策是 Amazon EKS 叢集正常運作所需的強制性政策。

優化 AWS 應用程序容器生成的碼頭映像

由瓦倫·夏爾馬（AWS）創建

環境：PoC 或試點	技術：容器與微服務；現代化； DevOps	AWS 服務：Amazon ECS

Summary

AWS App2Container 是一種命令列工具，可協助將在內部部署或虛擬機器上執行的現有應用程式轉換為容器，而不需要變更程式碼。

基於應用程序類型，App2Container 採用保守的方法來識別依賴關係。對於處理程序模式，應用程式伺服器上的所有非系統檔案都會包含在容器映像中。在這種情況下，可能會生成相當大的圖像。

此模式提供了一種優化 App2Container 生成的容器映像的方法。它適用於由 App2Container 在處理模式下發現的所有 Java 應用程序。在模式中定義的工作流程被設計為在應用程序服務器上運行。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 Linux 伺服器上的應用程式伺服器上執行的 Java 應用程式
- App2Container 在 Linux 伺服器上安裝和設定，且符合所有先決條件

架構

源, 技術, 堆棧

- 在 Linux 伺服器上執行的 Java 應用程式

目標技術堆疊

- 由 App2Container 生成的碼頭圖像

目標架構流程

1. 探索應用程式伺服器上執行的應用程式，並分析應用程式。
2. 容器化應用程式。
3. 評估碼頭圖像的大小。如果影像太大，請繼續執行步驟 4。
4. 使用 shell 腳本（附加）來識別大文件。
5. 更新`analysis.json`檔案中的`appExcludedFiles`和`appSpecificFiles`清單。

工具

工具

- AWS App2Container — AWS App2Container (A2C) 是一種命令列工具，可協助您提升和轉移在現場部署資料中心或虛擬機器上執行的應用程式，以便在由 Amazon 彈性容器服務 (Amazon ECS) 或亞馬遜彈性 Kubernetes 服務 (Amazon EKS) 管理的容器中執行。

Code

附加了 `optimizeImage.sh` shell 腳本和示例`analysis.json`文件。

該`optimizeImage.sh`文件是一個實用程序腳本，用於查看 App2Container 生成的文件的內容，. `ContainerFiles.tar` 檢閱會識別大型且可排除的檔案或子目錄。該腳本是以下 tar 命令的包裝器。

```
tar -Ptvf <path>|tr -s ' '|cut -d ' ' -f3,6| awk '$2 ~/<filetype>$/'| awk '$2 ~/
^<toplevel>/'| cut -f1-<depth> -d'/'|awk '{ if ($1>= <size>) arr[$2]+=$1 } END { for
 (key in arr) { if(<verbose>) printf("%-50s\t%-50s\n", key, arr[key]) else printf("%s,
\n", key) } } '|sort -k2 -nr
```

在 tar 命令中，指令碼會使用下列值：

`path`	的路徑 `ContainerFiles.tar`
`filetype`	要比對的檔案類型
`toplevel`	要比對的頂層目錄

`depth`	絕對路徑的深度
`size`	每個文件的大小

指令碼會執行以下操作：

1. 它使`tar -Ptvf`用列出文件而不提取它們。
2. 它按文件類型過濾文件，從頂級目錄開始。
3. 基於深度，它生成絕對路徑作為索引。
4. 根據索引和存儲，它提供了子目錄的總大小。
5. 它打印子目錄的大小。

您也可以在 tar 指令中手動取代這些值。

史诗

探索、分析和容器化應用程式

任務	描述	所需技能
探索內部部署 Java 應用程式。	若要探索應用程式伺服器上執行的所有應用程式，請執行下列命令。 `sudo app2container inventory`	AWS DevOps
分析發現的應用程序。	若要使用在詳細目錄階段中取得`application-id` 的應用程式來分析每個應用程式，請執行下列命令。 `sudo app2container analyze --application-id <java-app-id>`	AWS DevOps

任務	描述	所需技能
將分析的應用程式容器化。	若要將應用程式容器化，請執行下列命令。 `sudo app2container containerize --application-id <application-id>` 此命令會在工作區位置產生 Docker 映像以及 tar 套裝軟體。 如果 Docker 映像檔太大，請繼續執行下一個步驟。	AWS DevOps

識別 appExcludedFiles 並 appSpecificFiles 從應用程序容器中提取的 tar 文件

任務	描述	所需技能
識別工件 tar 檔案大小。	識別中的`ContainerFiles.tar` 檔案`{workspace}/{java-app-id}/Artifacts` ，其中`workspace` 是 App2Container 工作區，而且`java-app-id` 是應用程式識別碼。 `./optimizeImage.sh -p /{workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 0 -t / -v` 這是最佳化後 tar 檔案的總大小。	AWS DevOps

任務	描述	所需技能
列出/目錄下的子目錄及其大小。	若要識別/頂層目錄下主要子目錄的大小，請執行下列命令。 `./optimizeImage.sh -p /{workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 1 -t / -s 1000000 -v` `/var` `554144711` `/usr` `2097300819` `/tmp` `18579660` `/root` `43645397` `/opt` `222320534` `/home` `65212518` `/etc` `11357677`	AWS DevOps

任務	描述	所需技能
識別/目錄下的大型子目錄。	對於上一個命令中列出的每個主要子目錄，識別其子目錄的大小。用-d於增加深度並-t指示頂層目錄。 例如，用/var作頂層目錄。在下/var，識別所有大型子目錄及其大小。 `./optimizeImage.sh -p /{workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 2 -t /var -s 1000000 -v` 對上一個步驟中列出的每個子目錄重複此程序 (例如/usr/tmp、/opt、和/home)。	AWS DevOps

任務	描述	所需技能
分析/目錄下每個子目錄中的大型資料夾。	針對上一個步驟中列出的每個子目錄，識別執行應用程式所需的任何資料夾。 例如，使用上一個步驟中的子目錄，列出目錄中的所有子目/var錄及其大小。識別應用程式所需的任何子目錄。 `/var/tmp 237285851 /var/lib 24489984 /var/cache 237285851` 若要排除應用程式不需要的子目錄，請在analysis.json 檔案中將這些子目錄新增至下的appExcludedFiles 區段。containerParameters 隨即附加範例analysis.json 檔案。	AWS DevOps

任務	描述	所需技能
從「應用程式排除」清單中識別所需的檔案。	針對新增至 AppExclude 清單的每個子目錄，識別該子目錄中應用程式所需的任何檔案。在 Analys.json 檔案中，在下節中新增特定檔案或子目錄。appSpecificFiles containerParameters 例如，如果目/usr/lib錄已新增至排除清單，但/usr/lib/jvm 應用程式需要，請新增/usr/lib/jvm 至appSpecificFiles 區段。	AWS DevOps

再次擷取和容器化應用程式

任務	描述	所需技能
將分析的應用程式容器化。	若要將應用程式容器化，請執行下列命令。 `sudo app2container containerize --application-id <application-id>` 此命令會在工作區位置產生 Docker 映像以及 tar 套裝軟體。	AWS DevOps
識別工件 tar 檔案大小。	識別中的Container Files.tar 檔案{workspace}/{java-app-id}/Artifacts ，	AWS DevOps

任務	描述	所需技能
	其中 `workspace` 是 App2Container 工作區，而且 `java-app-id` 是應用程式識別碼。 `./optimizeImage.sh -p /{workspace}/{java-app-id}/Artifacts/ContainerFiles.tar -d 0 -t / -v` 這是最佳化後 tar 檔案的總大小。	
運行碼頭映像。	若要確認映像檔啟動時沒有錯誤，請使用下列命令在本機執行 Docker 映像。 若要識別 `imageId`容器，請使用`docker images |grep java-app-id` 。 若要執行容器，請使用`docker run -d <image id>`。	AWS DevOps

相關資源

- 什麼是應用程序容器？
- AWS 應用程式容器 — 適用於 Java 和 .NET 應用程式的全新容器化工具 (部落格文章)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用節點相似性、污點和容許，將 Kubernetes 網繭放置在 Amazon EKS 上

由希特斯·帕里赫（AWS）和拉格·比米迪馬里（AWS）創建

環境：PoC 或試點	技術：容器與微服務	工作負載：開源
AWS 服務：Amazon EKS		

Summary

此模式示範如何使用 Kubernetes 節點相似性、節點污點和網繭容許，在 Amazon Web 服務 (AWS) 雲端上的 Amazon Elastic Kubernetes Service (Amazon EKS) 叢集中有意排程應用程式 Amazon Web Services 繭上的特定工作者節點。

污染是一種節點屬性，可讓節點拒絕一組網繭。容許是一種網繭內容，可讓 Kubernetes 排程器在具有相符污染的節點上排程網繭。

不過，單獨的容許無法防止排程器將 Pod 放置在沒有任何污染的背景工作節點上。例如，具有容許功能的運算密集型 Pod 可能會無意中排程在一般用途未受污染的節點上。在該案例中，網繭的節點相似性內容會指示排程器將網繭置於符合節點相似性中指定之節點選取準則的節點上。

污染、容許和節點相似性一起指示排程器在具有相符污點的節點上一致地排程網繭，以及符合在網繭上指定的節點相似性節點選取準則的節點標籤。

此模式提供 Kubernetes 部署資訊清單檔案範例，以及建立 EKS 叢集、部署應用程式和驗證網繭放置的步驟。

先決條件和限制

先決條件

- 已設定登入資料的 AWS 帳戶，可在您的 AWS 帳戶上建立資源
- AWS 命令列界面 (AWS CLI)
- eksctl
- kubectl

- Docker 已安裝 (針對正在使用的作業系統)，並啟動引擎 (如需 Docker 授權需求的相關資訊，請參閱 Docker 網站)
- Java 版本 11 或更新版本
- 在您最喜歡的集成開發環境（IDE）上運行的 Java 微服務; 例如，AWS Cloud9，IntelliJ 理想社區版或 E clipse（如果您沒有 Java 微服務，請參閱在 Amazon EKS 模式和微服務與春季部署示例 Java 微服務以獲取有關創建微服務的幫助）

限制

- 此模式不提供 Java 代碼，並假定您已經熟悉 Java。若要建立基本的 Java 微服務，請參閱在 Amazon EKS 上部署範例 Java 微服務。
- 本文中的步驟建立可累積成本的 AWS 資源。請務必在完成實作和驗證模式的步驟之後清理 AWS 資源。

架構

目標技術堆疊

- Amazon EKS
- Java
- Docker
- Amazon Elastic Container Registry (Amazon ECR)

目標架構

解決方案架構圖顯示具有兩個網繭 (部署 1 和部署 2) 的 Amazon EKS，以及兩個節點群組 (ng1 和 ng2)，每個節點各有兩個節點。網繭和節點具有下列屬性。

	部署 1 個網繭	部署 2 網繭	節點群組 1	節點群組 2
寬容	鍵:分類工作負載, 值:真, 效果: NoSchedule 關鍵：機器學習工作負載，	無		

	值：真，效果：NoSchedule		
節點相似性	鍵：阿爾法 .eksctl.io /節點組名稱 = ng1;	無	節點群組. 名稱 = ng1
污點			鍵:分類工作負載, 值:真, 效果: NoSchedule 關鍵：機器學習工作負載，值：真，效果：NoSchedule

1. 部署 1 網繭已定義容許和節點相似性，這會指示 Kubernetes 排程器將部署網繭放在節點群組 1 (ng1) 節點上。
2. 節點群組 2 (ng2) 沒有符合部署 1 的節點相似性節點選取器運算式的節點標籤，因此 Pod 不會排程在 ng2 節點上。
3. 部署 2 Pod 沒有在部署資訊清單中定義任何容許或節點相似性。由於節點上的污點，排程器將拒絕排程節點群組 1 上的部署 2 Pod。
4. 部署 2 Pod 將改為放置在節點群組 2 上，因為節點沒有任何污點。

此模式示範，透過使用污點和容許結合節點相似性，您可以控制 Pod 在特定工作節點集上的放置。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。

- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- eksctl 是 AWS 相當於庫貝克特爾，並有助於創建 EKS。

其他工具

- Docker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中提供軟體。
- kubectl 是一種命令列介面，可協助您針對 Kubernetes 叢集執行命令。

史诗

建立 EKS 叢集

任務	描述	所需技能
建立叢集 .yaml 檔案。	`cluster.yaml` 使用下面的代碼創建一個名為的文件。 `apiVersion: eksctl.io/v1alpha5` `kind: ClusterConfig` `metadata:` `  name: eks-taint-demo` `  region: us-west-1` `# Unmanaged nodegroups with and without taints.` `nodeGroups:` `  - name: ng1` `  instanceType: m5.xlarge` `  minSize: 2` `  maxSize: 3` `  taints:` `  - key: classified_workload` `  value: "true"`	應用程式擁有者、AWS DevOps、雲端管理員、DevOps 工程師

任務	描述	所需技能
	<pre>effect: NoSchedule - key: machine_l earning_workload value: "true" effect: NoSchedule - name: ng2 instanceType: m5.xlarge minSize: 2 maxSize: 3</pre>	
使用 eksctl 建立叢集。	執行`cluster.yaml` 檔案以建立 EKS 叢集。建立叢集可能需要幾分鐘的時間。 <pre>eksctl create cluster -f cluster.yaml</pre>	AWS DevOps、AWS 系統管理員、應用程式開發人員

創建一個圖像並將其上傳到 Amazon ECR

任務	描述	所需技能
建立 Amazon ECR 私有儲存庫。	若要建立 Amazon ECR 儲存庫，請參閱建立私有存放庫。請注意回購的 URI。	AWS DevOps、 DevOps 工程師、應用程式開發者
創建碼頭文件。	如果您有要用來測試模式的現有 Docker 容器映像檔，則可以略過此步驟。 要創建一個 Dockerfile，請使用以下代碼片段作為參考。如果您遇到錯誤，請參閱疑難排解一節。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	`FROM adoptopenjdk/openjdk11:jdk-11.0.14.1_1-alpine` `RUN apk add maven` `WORKDIR /code` `# Prepare by downloading dependencies` `ADD pom.xml /code/pom.xml` `RUN ["mvn", "dependency:resolve"]` `RUN ["mvn", "verify"]` `# Adding source, compile and package into a fat jar` `ADD src /code/src` `RUN ["mvn", "package"]` `EXPOSE 4567` `CMD ["java", "-jar", "target/eksExample-jar-with-dependencies.jar"]`	
創建 pom.xml 和源文件，並構建和推送碼頭映像。	若要建立`pom.xml`檔案和 Java 來源檔案，請參閱在 Amazon EKS 模式上部署範例 Java 微服務。 使用該模式中的說明來構建和推送 Docker 映像。	AWS DevOps、 DevOps 工程師、應用程式開發者

部署到 Amazon EKS

任務	描述	所需技能
建立部署的 .yaml 檔案。	若要建立`deployment.yaml` 檔案，請使用 [其他資訊] 區段中的程式碼。 在程式碼中，節點相似性的索引鍵是您在建立節點群組時建立的任何標籤。此模式使用 eksctl 創建的默認標籤。如需自訂標籤的相關資訊，請參閱 Kubernetes 文件中的將網繭指派給節點。 節點相似性索引鍵的值是由建立的節點群組名稱`cluster.yaml` 。 若要取得污點的索引鍵和值，請執行下列命令。 `kubectl get nodes -o json | jq '.items[].spec.taints'` 映像檔是您在先前步驟中建立的 Amazon ECR 儲存庫的 URI。	AWS DevOps、 DevOps 工程師、應用程式開發者
部署檔案。	若要部署到 Amazon EKS，請執行下列命令。 `kubectl apply -f deployment.yaml`	AWS 應用程式開發人員、DevOps 工程師 DevOps

任務	描述	所需技能
檢查部署。	1. 若要檢查網繭是否已就緒，請執行下列命令。 `kubectl get pods -o wide` 如果 POD 已準備就緒，輸出應類似下列內容，並顯示STATUS為「執行中」。 `NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES` `<pod_name> 1/1 Running 0 12d 192.168.18.50 ip-192-168-20-110.us-west-1.compute.internal <none> <none>` 記下網繭的名稱和節點的名稱。您可以跳過下一步。 2. (選擇性) 若要取得有關網繭的其他詳細資料並檢查網繭上的容許，請執行下列命令。 `kubectl describe pod <pod_name>` 輸出的範例位於 [其他資訊] 區段中。	AWS 應用程式開發人員、DevOps 工程師 DevOps

任務	描述	所需技能
	3. 若要驗證節點上的 Pod 放置是否正確，請執行下列命令。 `kubectl describe node <node name> | grep -A 1 "Taints"` 確認節點上的污點符合公差，且節點上的標籤符合中定義的節點相似性。`deployment.yaml` 具有容許和節點相似性的 Pod 應放置在具有相符污點和節點相似性標籤的節點上。上一個指令會提供節點上的污點。以下為範例輸出。 `kubectl describe node ip-192-168-29-181.us-west-1.compute.internal | grep -A 1 "Taints"` `Taints: classifled_workload=true:NoSchedule` `machine_learning_workload=true:NoSchedule` 此外，執行下列命令，以檢查所放置 Pod 的節點是否具有符合節點相似性節點標籤的標籤。	

任務	描述	所需技能
	`kubectl get node <node name> --show-labels` 4. 若要確認應用程式是否正在執行其所要執行的動作，請執行下列命令來檢查 Pod 記錄。 `kubectl logs -f <name-of-the-pod>`	

任務	描述	所需技能
建立沒有容許和節點相似性的第二個部署 .yaml 檔案。	此額外步驟是驗證在部署資訊清單檔案中未指定任何節點相似性或容許時，產生的 Pod 不會在具有污染物的節點上排程。（它應該安排在沒有任何污點的節點上）。使用下列程式碼建立名為的新部署檔案`deploy_no_taint.yaml` 。 `apiVersion: apps/v1` `kind: Deployment` `metadata:` `  name: microservice-deployment-non-tainted` `spec:` `  replicas: 1` `  selector:` `    matchLabels:` `      app.kubernetes.io/name: java-microservice-no-taint` `  template:` `    metadata:` `      labels:` `        app.kubernetes.io/name: java-microservice-no-taint` `    spec:` `      containers:` `        - name: java-microservice-container-2` `          image: <account_number>.dkr.ecr<region>.amazonaws.com/<repository_name>:latest` `          ports:`	應用開發人員、AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	`- container` `Port: 4567`	
部署第二個部署 .yaml 檔案，並驗證網繭放置	1. 執行下列命令。 `kubectl apply -f deploy_no_taint.yaml` 2. 部署成功後，請執行先前執行的相同命令，以檢查節點群組中的 Pod 位置，而不會造成污染。 `kubectl describe node <node_name> | grep "Taints"` 輸出應該是以下內容。 `Taints: <none>` 這樣就完成了測試。	應用開發人員、AWS DevOps、 DevOps 工程師

清除資源

任務	描述	所需技能
清除資源。	若要避免對仍在執行的資源產生 AWS 費用，請使用下列命令。 `eksctl delete cluster --name <Name of the`	AWS 應用程 DevOps式開發人員

任務	描述	所需技能
	`cluster> --region <region-code>`	

故障診斷

問題	解決方案
如果您的系統使用 arm64 架構（特別是在 M1 Mac 上運行此體系結構），則其中一些命令可能無法運行。以下行可能會出錯。 `FROM adoptopenjdk/openjdk11:jdk-11.0.14.1_1-alpine`	如果您在運行 Docker 文件時出現錯誤，請用以下FROM行替換該行。 `FROM bellsoft/liberica-openjdk-alpine-musl:17`

相關資源

- 在 Amazon EKS 上部署示例 Java 微服務
- 建立 Amazon ECR 私有儲存庫
- 將網繭指派給節點 (Kubernetes 文件)
- 污染和容忍 (庫伯尼特文件)
- Amazon EKS
- Amazon ECR
- AWS CLI
- Docker
- 智能理念公司
- Eclipse

其他資訊

部署. 羊

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservice-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: java-microservice
  template:
    metadata:
      labels:
        app.kubernetes.io/name: java-microservice
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: alpha.eksctl.io/nodegroup-name
                    operator: In
                    values:
                      - <node-group-name-from-cluster.yaml>
      tolerations: #only this pod has toleration and is viable to go to ng with taint
        - key: "<Taint key>" #classified_workload in our case
          operator: Equal
          value: "<Taint value>" #true
          effect: "NoSchedule"
        - key: "<Taint key>" #machine_learning_workload in our case
          operator: Equal
          value: "<Taint value>" #true
          effect: "NoSchedule"
      containers:
        - name: java-microservice-container
          image: <account_number>.dkr.ecr<region>.amazonaws.com/
<repository_name>:latest
          ports:
            - containerPort: 4567
```

描述 pod 示例輸出

```
Name:         microservice-deployment-in-tainted-nodes-5684cc495b-vpcfx
Namespace:    default
```

```
Priority:     0
Node:         ip-192-168-29-181.us-west-1.compute.internal/192.168.29.181
Start Time:   Wed, 14 Sep 2022 11:06:47 -0400
Labels:       app.kubernetes.io/name=java-microservice-taint
              pod-template-hash=5684cc495b
Annotations:  kubernetes.io/psp: eks.privileged
Status:       Running
IP:           192.168.13.44
IPs:
  IP:           192.168.13.44
Controlled By:  ReplicaSet/microservice-deployment-in-tainted-nodes-5684cc495b
Containers:
  java-microservice-container-1:
    Container ID:
 docker://5c158df8cc160de8f57f62f3ee16b12725a87510a809d90a1fb9e5d873c320a4
    Image:          934188034500.dkr.ecr.us-east-1.amazonaws.com/java-eks-apg
    Image ID:       docker-pullable://934188034500.dkr.ecr.us-east-1.amazonaws.com/
java-eks-apg@sha256:d223924aca8315aab20d54eddf3443929eba511b6433017474d01b63a4114835
    Port:           4567/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Wed, 14 Sep 2022 11:07:02 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-ddvvw (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-ddvvw:
    Type:                     Projected (a volume that contains injected data from
 multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 classifled_workload=true:NoSchedule
```

```
                            machine_learning_workload=true:NoSchedule
                            node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                            node.kubernetes.io/unreachable:NoExecute op=Exists for
 300s
Events:                     <none>
```

跨帳戶或區域複寫篩選過的 Amazon ECR 容器映像

創建者：阿卜達爾·加魯巴 (AWS)

環境：生產	技術：容器和微服務；DevOps	AWS 服務：Amazon EC2 容器註冊表；Amazon CloudWatch；AWS CodeBuild ；AWS Identity and Access Management；AWS CLI

Summary

Amazon Elastic Container Registry (Amazon ECR) 可以使用跨區域和跨帳戶複寫功能，以原生方式跨 Amazon Web Services (AWS) 區域和 AWS 帳戶複寫映像儲存庫中的所有容器映像檔。如需詳細資訊，請參閱 AWS 部落格文章 Amazon ECR 中的跨區域複寫已登陸。) 但是，無法根據任何條件篩選跨 AWS 區域或帳戶複製的映像。

此模式說明如何根據映像標籤模式，跨 AWS 帳戶和區域複寫存放在 Amazon ECR 中的容器映像。該模式使用 Amazon E CloudWatch vents 偵聽具有預先定義自訂標籤的映像的推送事件。推送事件會啟動 AWS CodeBuild 專案，並將映像詳細資料傳遞給該專案。 CodeBuild 專案會根據提供的詳細資訊，將來源 Amazon ECR 登錄中的映像複製到目的地登錄。

此模式會複製跨帳戶具有特定標記的影像。例如，您可以使用此模式僅將生產就緒的安全映像複製到生產 AWS 帳戶。在開發帳戶中，在完整測試映像之後，您可以將預先定義的標籤新增至安全映像，並使用此模式中的步驟將標記的映像複製到生產帳戶。

先決條件和限制

先決條件

- 來源和目的地 Amazon ECR 登錄的有效 AWS 帳戶
- 此模式中使用之工具的系統管理權限
- 安裝在本地計算機上進行測試的 Docker
- AWS Command Line Interface (AWS CLI) (AWS CLI)，用於在 Amazon ECR 中進行身份驗證

限制

- 此模式只會監視一個 AWS 區域中來源登錄的推送事件。您可以將此模式部署到其他區域，以觀看這些區域中的登錄。
- 在此模式中，一個 Amazon CloudWatch 事件規則會偵聽單一影像標籤模式。如果您想要檢查多個模式，您可以新增事件以偵聽其他影像標記模式。

架構

目標架構

自動化和規模

此模式可透過基礎架構即程式碼 (IaC) 指令碼自動化，並大規模部署。若要使用 AWS CloudFormation 範本部署此模式，請下載附件並遵循其他資訊一節中的指示進行操作。

您可以將多個 Amazon E CloudWatch vents 事件 (使用不同的自訂事件模式) 指向同一個 AWS CodeBuild 專案，以複寫多個映像標籤模式，但是您需要更新`buildspec.yaml`檔案中的次要驗證 (包含在附件和工具區段中)，如下所示，以支援多種模式。

```
...
if [[ ${IMAGE_TAG} != release-* ]]; then
...
```

工具

Amazon 服務

- IAM — AWS Identity and Access Management (IAM) 可讓您安全地管理 AWS 服務和資源的存取。在此模式中，您需要建立跨帳戶 IAM 角色，AWS 在 CodeBuild 將容器映像推送到目的地登錄時會承擔這個角色。
- Amazon ECR — Amazon Elastic Container Registry (Amazon ECR) 是一種全受管容器登錄，可讓您在任何地方輕鬆存放、管理、共用和部署容器映像和成品。映像推送動作至來源登錄會將系統事件詳細資訊傳送至 Amazon E CloudWatch vents 所拾取的事件匯流排。
- AWS CodeBuild — AWS CodeBuild 是全受管的持續整合服務，可提供運算能力來執行任務，例如編譯原始程式碼、執行測試，以及產生可供部署的成品。此模式使用 AWS CodeBuild 執行從來源 Amazon ECR 登錄到目的地登錄的複製動作。

- CloudWatch 活動 — Amazon CloudWatch 活動提供一系統事件串流，用來描述 AWS 資源中的變更。此模式使用規則將 Amazon ECR 推送動作與特定影像標籤模式相符。

工具

- 碼頭 CLI — Docker 是一種工具，可以更輕鬆地創建和管理容器。容器會將應用程式及其所有相依性封裝到一個單元或套件中，以便在任何支援容器執行階段的平台上輕鬆部署。

Code

您可以通過兩種方式實現此模式：

- 自動化設定：部署附件中提供的兩個 AWS CloudFormation 範本。如需指示，請參閱「其他資訊」一節。
- 手動設定：依照 Epics 區段中的步驟操作。

建置規格範例

如果您使用此 CloudFormation 模式提供的範本，`buildspec.yaml`檔案會包含在 CodeBuild 資源中。

```
  version: 0.2
  env:
    shell: bash
  phases:
    install:
      commands:
        - export CURRENT_ACCOUNT=$(echo ${CODEBUILD_BUILD_ARN} | cut -d':' -f5)
        - export CURRENT_ECR_REGISTRY=${CURRENT_ACCOUNT}.dkr.ecr.
${AWS_REGION}.amazonaws.com
        - export DESTINATION_ECR_REGISTRY=${DESTINATION_ACCOUNT}.dkr.ecr.
${DESTINATION_REGION}.amazonaws.com
    pre_build:
      on-failure: ABORT
      commands:
        - echo "Validating Image Tag ${IMAGE_TAG}"
        - |
          if [[ ${IMAGE_TAG} != release-* ]]; then
            aws codebuild stop-build --id ${CODEBUILD_BUILD_ID}
            sleep 60
```

```
            exit 1
          fi
        - aws ecr get-login-password --region ${AWS_REGION} | docker login -u AWS --
password-stdin ${CURRENT_ECR_REGISTRY}
        - docker pull ${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
    build:
      commands:
        - echo "Assume cross-account role"
        - CREDENTIALS=$(aws sts assume-role --role-arn ${CROSS_ACCOUNT_ROLE_ARN} --
role-session-name Rolesession)
        - export AWS_DEFAULT_REGION=${DESTINATON_REGION}
        - export AWS_ACCESS_KEY_ID=$(echo ${CREDENTIALS} | jq -r
 '.Credentials.AccessKeyId')
        - export AWS_SECRET_ACCESS_KEY=$(echo ${CREDENTIALS} | jq -r
 '.Credentials.SecretAccessKey')
        - export AWS_SESSION_TOKEN=$(echo ${CREDENTIALS} | jq -r
 '.Credentials.SessionToken')
        - echo "Logging into cross-account registry"
        - aws ecr get-login-password --region ${DESTINATION_REGION} | docker login -u
 AWS --password-stdin ${DESTINATION_ECR_REGISTRY}
        - echo "Check if Destination Repository exists, else create"
        - |
          aws ecr describe-repositories --repository-names ${REPO_NAME} --region
 ${DESTINATION_REGION} \
          || aws ecr create-repository --repository-name ${REPO_NAME} --region
 ${DESTINATION_REGION}
        - echo "retag image and push to destination"
        - docker tag ${CURRENT_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
 ${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
        - docker push ${DESTINATION_ECR_REGISTRY}/${REPO_NAME}:${IMAGE_TAG}
```

史诗

建立 IAM 角色

任務	描述	所需技能
建立 CloudWatch 事件角色。	在來源 AWS 帳戶中，為要假設的 Amazon CloudWatch 事件建立 IAM 角色。該角色應具有啟動 AWS CodeBuild 專案的許可。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師

任務	描述	所需技能
	若要使用 AWS CLI 建立角色，請遵循 IAM 文件中的指示。 範例信任原則 (`trustpolicy.json`)： `{ "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Principal": {"Service": "events.amazonaws.com"}, "Action": "sts:AssumeRole" } }` 範例權限原則 (`permissionpolicy.json`)： `{ "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Action": "codebuild:StartBuild", "Resource": "<CodeBuild Project ARN>" } }`	

任務	描述	所需技能
建立 CodeBuild 角色。	按照 IAM 文件中的說明，CodeBuild 為 AWS 建立要假設的 IAM 角色。角色應具有下列權限： • 承擔目標跨帳戶角色的權限 • 建立記錄群組和記錄資料流，以及放置記錄事件的權限 • 透過將 AmazonEC2ContainerRegistryReadOnly 受管政策新增至該角色，即可獲得所有 Amazon ECR 儲存庫的唯讀許可 • 停止權限 CodeBuild 範例信任原則 (`trustpolicy.json`)： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "codebuild.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師

任務	描述	所需技能
	範例權限原則 (permissionpolicy.json)： { "Version": "2012-10-17", "Statement": [{ "Action": ["codebuild:StartBuild", "codebuild:StopBuild", "codebuild:Get*", "codebuild:List*", "codebuild:BatchGet*"], "Resource": "*", "Effect": "Allow" }, { "Action": ["logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents"], "Resource": "*",	

任務	描述	所需技能
	<pre>"Effect": "Allow" }, { "Action": "sts:AssumeRole", "Resource": "<ARN of destination role>", "Effect": "Allow", "Sid": "AssumeCrossAccountArn" }] }</pre> 將受管理的政策附加 AmazonEC2ContainerRegistryReadOnly 到 CLI 命令，如下所示： <pre>~$ aws iam attach-role-policy \ --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \ --role-name <name of CodeBuild Role></pre>	

任務	描述	所需技能
建立跨帳戶角色。	在目的地 AWS 帳戶中，為要承擔的來源帳戶的 AWS CodeBuild 角色建立 IAM 角色。跨帳戶角色應允許容器映像建立新的儲存庫，並將容器映像上傳到 Amazon ECR。 若要使用 AWS CLI 建立 IAM 角色，請遵循 IAM 文件中的指示。 若要允許上一個步驟中的 AWS CodeBuild 專案，請使用下列信任政策： `{ "Version": "2012-10-17", "Statement": { "Effect": "Allow", "Principal": { "AWS": "<ARN of source codebuild role>" }, "Action": "sts:AssumeRole" } }` 若要允許上一步中的 AWS CodeBuild 專案將映像儲存在目的地登錄中，請使用下列許可政策： `{ "Version": "2012-10-17", "Statement": [`	AWS 管理員、AWS DevOps、雲端管理員、雲端架構師、 DevOps 工程師、AWS 系統管理員

任務	描述	所需技能
	<pre>{ "Action": ["ecr:GetDownloadUrlForLayer", "ecr:BatchCheckLayerAvailability", "ecr:PutImage", "ecr:InitiateLayerUpload", "ecr:UploadLayerPart", "ecr:CompleteLayerUpload", "ecr:GetRepositoryPolicy", "ecr:DescribeRepositories", "ecr:GetAuthorizationToken", "ecr:CreateRepository"], "Resource": "*", "Effect": "Allow" }] }</pre>	

建立 CodeBuild 專案

任務	描述	所需技能
建立 CodeBuild 專案。	按照 AWS 文 CodeBuild 件中的指示，在來源帳戶中建立 CodeBuild AWS 專案。專案應與來源登錄位於相同的區域。 設定專案的方式如下： • 環境類型：`LINUX CONTAINER` • 服務角色：`CodeBuild Role` • 特權模式：`true` • 環境圖像：`aws/codebuild/standard:x.x` （使用可用的最新圖像） • 環境變數： • `CROSS_ACCOUNT_ROLE_ARN` ：跨帳戶角色的 Amazon 資源名稱 (ARN) • `DESTINATION_REGION` ：跨帳戶區域的名稱 • `DESTINATION_ACCOUNT` ：目標帳戶的編號 • 構建規格：使用「工具」部分中列出的`buildspec.yaml` 文件。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師

建立活動

任務	描述	所需技能
建立事件規則。	由於該模式使用內容過濾功能，因此您需要使用 Amazon 創建事件 EventBridge。依照文件中的指示建立事 EventBridge 件和目標，並進行一些修改： • 在「定義模式」中，選擇「事件模式」，然後選擇「自訂模式」。 • 將下列自訂事件模式範例程式碼複製到提供的文字方塊中： `{ "source": ["aws.ecr"], "detail-type": ["ECR Image Action"], "detail": { "action-type": ["PUSH"], "result": ["SUCCESS"], "image-tag": [{ "prefix": "release-"}] } }` • 對於選取目標，請選擇 AWS CodeBuild 專案，然後貼上您在上一個史詩中建立的 AWS CodeBuild 專案的 ARN。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師

任務	描述	所需技能
	• 對於「設定輸入」，選擇「輸入變壓器 • 在「輸入路徑」文字方塊中，貼上： • 在「輸入範本」文字方塊中，貼上： • 選擇 [使用現有角色]，然後選擇您先前在建立 IAM 角色史詩中建立的 CloudWatch 事件角色名稱。	

```
 {"IMAGE_TAG":"$.de
tail.image-tag","R
EPO_NAME":"$.detai
l.repository-name"
}
```

```
 {"environmentVaria
blesOverride":
 [ {"name":
 "IMAGE_TAG",
 "value":<IMAGE_TAG
>},{"name":"REPO_N
AME","value":<REPO
_NAME>}]}
```

驗證

任務	描述	所需技能
與 Amazon ECR 進行身份驗證。	遵循 Amazon ECR 文件中的步驟，對來源和目的地登錄進行驗證。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、 DevOps 工程師、雲端架構師

任務	描述	所需技能
測試映像複製。	在您的來源帳戶中，將容器映像推送到新的或現有的 Amazon ECR 來源儲存庫，且影像標籤前置為。`release-`若要推送映像檔，請依照 Amazon ECR 文件中的步驟執行。 您可以在CodeBuild 主控台中監視 CodeBuild 專案的進度。 成功完成 CodeBuild 專案後，請登入目的地 AWS 帳戶、開啟 Amazon ECR 主控台，並確認該映像存在於目的地 Amazon ECR 登錄中。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師
測試圖像排除。	在您的來源帳戶中，將容器映像推送到新的或現有的 Amazon ECR 來源儲存庫，其中包含沒有自訂前置詞的映像標籤。 確認 CodeBuild 專案尚未啟動，且目標登錄中沒有顯示容器映像檔。	AWS 管理員、AWS DevOps、AWS 系統管理員、雲端管理員、雲端架構師、DevOps 工程師

相關資源

- 開始使用 CodeBuild
- 開始使用 Amazon EventBridge
- Amazon EventBridge 事件模式中基於內容的篩選
- 使用 IAM 角色委派跨 AWS 帳戶的存取
- 私人映像複製

其他資訊

若要自動部署此病毒碼的資源，請依照下列步驟執行：

1. 下載附件並解壓縮兩個 CloudFormation 範本：`part-1-copy-tagged-images.yaml`和`part-2-destination-account-role.yaml`。
2. 登入 AWS CloudFormation 主控台，並`part-1-copy-tagged-images.yaml`在與來源 Amazon ECR 登錄相同的 AWS 帳戶和區域中部署。視需要更新參數。範本會部署下列資源：
   - Amazon CloudWatch 活動 IAM 角色
   - AWS CodeBuild 專案 IAM 角色
   - AWS CodeBuild 專案
   - AWS CloudWatch 事件規則
3. 記下「輸出」索引標籤`SourceRoleName`中的值。在下一個步驟中，您將需要此值。
4. 在您要將 Amazon ECR 容器映像複製到的 AWS 帳戶中部署第二個 CloudFormation 範本。`part-2-destination-account-role.yaml`視需要更新參數。對於`SourceRoleName`參數，請指定步驟 3 中的值。此範本會部署跨帳戶 IAM 角色。
5. 驗證映像複製和排除，如 Epics 一節的最後一個步驟所述。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

輪換資料庫認證而不重新啟動

創建者：喬希·喬伊（AWS）

環境：生產	技術：容器與微服務；資料庫；基礎架構 DevOps；安全性、身分識別、合規性；管理與治理	AWS 服務：Amazon ECS; Amazon Aurora; AWS Fargate; AWS Secrets Manager; Amazon VPC

Summary

在 Amazon Web Services (AWS) 雲端上，您可以使用 AWS Secrets Manager 在資料庫的整個生命週期中輪換、管理和擷取資料庫登入資料。使用者和應用程式會透過呼叫 Secrets Manager API 擷取密碼，不需要以純文字形式對敏感資訊進行硬式編碼。

如果您將容器用於微服務工作負載，則可以在 AWS Secrets Manager 中安全地存放登入資料。若要將組態與程式碼分開，這些認證通常會插入到容器中。不過，定期且自動輪換您的認證很重要。支援撤銷後重新整理認證的能力也很重要。同時，應用程式需要輪換認證的能力，同時減少任何潛在的下游可用性影響。

此模式描述如何輪換容器內使用 AWS Secrets Manager 保護的密碼，而不需要重新啟動容器。此外，此病毒碼會使用 Secret Manager 用戶端快取元件，減少秘 Secrets Manager 的認證查閱次數。當您使用用戶端快取元件重新整理應用程式內的認證時，不需要重新啟動容器即可擷取輪換的認證。

這種方法適用於 Amazon Elastic Kubernetes Service（Amazon EKS）和 Amazon Elastic Container Service (Amazon ECS)。

涵蓋了兩種情況。在單一使用者案例中，會偵測到過期的認證，在密碼輪換時重新整理資料庫認證。系統會指示認證快取重新整理密碼，然後應用程式會重新建立資料庫連線。用戶端快取元件會快取應用程式內的認證，並協助避免針對每個認證查詢與 Secrets Manager 聯絡。認證會在應用程式內輪換，而不需要透過重新啟動容器強制重新整理認證。

第二個案例會在兩個使用者之間交替來旋轉密碼。擁有兩個作用中使用者可減少停機的可能性，因為一個使用者的認證永遠處於作用中狀態 雙使用者認證輪換在叢集的大型部署時很有幫助，其中認證更新可能會有很小的傳播延遲。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在 Amazon EKS 或 Amazon ECS 容器中運行的應用程序。
- 儲存在 Secrets Manager 中的認證，且已啟用輪換。
- 如果部署雙使用者解決方案，則會儲存在 Secrets Manager 中的第二組認證。代碼示例可以在 GitHub 回購 aws-secrets-manager-rotation-lambdas 中找到。
- Amazon Aurora 數據庫。

限制

- 這個例子是針對 Python 應用程序的目標。對於 Java 應用程式，您可以使用 Java 用戶端快取元件或機 Secrets Manager 的 JDBC 用戶端快取程式庫。

架構

目標架構

案例 1 — 單一使用者的認證輪換

在第一個案例中，Secrets Manager 會定期輪換單一資料庫認證。應用程序容器在 Fargate 中運行。建立第一個資料庫連線時，應用程式容器會擷取 Aurora 的資料庫認證。然後 Secrets Manager 快取元件會快取認證，以便 future 建立連線。輪替期間過後，認證會過期，而且資料庫會傳回驗證錯誤。接著應用程式會擷取輪換的認證、使快取無效，並透過 Secrets Manager 用戶端快取元件更新認證快取。

在這個案例中，當認證正在輪換且過時的連線正在使用過時的認證時，可能會發生最小的中斷。您可以使用雙使用者案例來解決此問題。

案例 2 — 輪替兩個使用者的認證

在第二個案例中，Secrets Manager 會定期輪換兩個資料庫使用者認證 (愛麗絲和鮑勃的)。應用程式容器會在 Fargate 叢集中執行。建立第一個資料庫連線時，應用程式容器會擷取第一位使用者的 Aurora 資料庫認證 (Alice)。然後 Secrets Manager 快取元件會快取認證，以便 future 建立連線。

雖然有兩個使用者和認證，但只有一個作用中的認證是由 Secrets Manager 管理。在這種情況下，緩存組件會定期過期並獲取最新的憑據。如果 Secrets Manager 輪替期間超過快取逾時，快取元件會為第二位使用者 (Bob) 取得輪換的認證。例如，如果快取到期時間是以分鐘為單位，而輪換期間是以天為單位，則快取元件會擷取新的認證，做為其定期快取重新整理的一部分。如此一來，停機時間就會降到最低，因為每個使用者的認證在一次 Secrets Manager 輪替中都處於作用中狀態。

自動化和規模

您可以使用 AWS 使用基礎設施即程式碼 CloudFormation來部署此模式。這會建立並建立應用程式容器、建立 Fargate 工作、將容器部署到 Fargate，以及使用 Aurora 設定和設定 Secrets Manager。如需 step-by-step 部署指示，請參閱讀我檔案。

工具

工具

- AWS Secrets Manager 可透過 API 呼叫秘密管 Secrets Manager 員來取代硬式編碼登入資料 (包括密碼) 以擷取密碼。由於 Secrets Manager 可以根據排程自動輪換密碼，因此您可以用短期密碼取代長期密碼，從而降低入侵的風險。
- Docker 幫助開發人員將任何應用程序打包，發貨和運行作為一個輕量級，便攜和自給自足的容器。

Code

示例 Python 代碼

此模式會使用 Secrets Manager 的 Python 用戶端快取元件，在建立資料庫連線時擷取驗證認證。用戶端快取元件有助於避免每次都與 Secrets Manager 聯絡。

現在，當輪換週期過後，緩存的憑據將過期，並且連接到數據庫將導致身份驗證錯誤。對於 MySQL，驗證錯誤代碼是 1045。此範例將 Amazon Aurora 用於 MySQL，不過您可以使用其他引擎，例如 PostgreSQL。發生驗證錯誤時，資料庫連線例外狀況處理程式碼會擷取錯誤。然後它會通知 Secrets Manager 用戶端快取元件重新整理密碼，然後重新驗證並重新建立資料庫連線。如果您使用 PostgreSQL 或其他引擎，則必須查找相應的身份驗證錯誤代碼。

容器應用程式現在可以使用旋轉的密碼更新資料庫密碼，而無需重新啟動容器。

將下列程式碼放在處理資料庫連線的應用程式程式碼中。這個例子使用 Django，它使用用於連接的數據庫包裝器對數據庫後端進行子類。如果您使用不同的程式設計語言或資料庫連線程式庫，請參閱您的資料庫連線程式庫，以檢閱如何子類別資料庫連線擷取。

```
    def get_new_connection(self, conn_params):
        try:
            logger.info("get connection")
            databasecredentials.get_conn_params_from_secrets_manager(conn_params)
            conn =super(DatabaseWrapper,self).get_new_connection(conn_params)
            return conn
        except MySQLdb.OperationalError as e:
            error_code=e.args[0]
            if error_code!=1045:
                raise e

            logger.info("Authentication error. Going to refresh secret and try again.")
            databasecredentials.refresh_now()
            databasecredentials.get_conn_params_from_secrets_manager(conn_params)
            conn=super(DatabaseWrapper,self).get_new_connection(conn_params)
            logger.info("Successfully refreshed secret and established new database
connection.")
            return conn
```

AWS CloudFormation 和 Python 程式碼

- https://github.com/aws-samples/ aws-secrets-manager-credential-rotation-without-container-restart

史诗

在認證輪替期間維持應用程式

任務	描述	所需技能
安裝快取元件。	下載並安裝 Python 的 Secrets Manager 客戶端緩存組件。如需下載連結，請參閱相關資源一節。	開發人員
緩存工作憑據。	使用 Secrets Manager 用戶端快取元件，在本機快取工作中的認證。	開發人員

任務	描述	所需技能
更新應用程式程式碼，以在資料庫連線發生未經授權的錯誤時重新整理認證。	更新應用程式程式碼，以使用 Secrets Manager 來擷取和重新整理資料庫認證。添加邏輯以處理未經授權的錯誤代碼，然後獲取新輪換的憑證。請參閱 Python 程式碼範例一節。	開發人員

相關資源

建立密碼管理員密碼

- 在 AWS KMS 中建立金鑰
- 使用 AWS 秘密管理員建立和管理機密

建立 Amazon Aurora 叢集

- 建立 Amazon RDS 資料庫執行個體

創建 Amazon ECS 組件

- 使用傳統主控台建立叢集
- 建立泊塢視窗映像
- 創建一個私有存儲庫
- Amazon ECR 私人註冊表
- 推送碼頭圖像
- Amazon ECS 任務定義
- 在傳統主控台中建立 Amazon ECS 服務

下載並安裝 Secrets Manager 用戶端快取元件

- Python 快取用戶端

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

WorkSpaces 使用 Amazon ECS 隨時隨地在 Amazon 上運行 Amazon ECS Anywhere 務

由阿卡什·庫馬爾（AWS）創建

環境：生產	技術：容器與微服務；現代化	工作負載：所有其他工作
AWS 服務：Amazon ECS; Amazon WorkSpaces; AWS Directory Service		

Summary

亞馬遜彈性容器服務 (Amazon ECS) 任何地方都支援在任何環境中部署 Amazon ECS 任務，包括 Amazon Web Services (AWS) 受管基礎設施和客戶管理的基礎設施。您可以在使用在雲端執行且始終保持最新狀態的完全 AWS 受管控平面時執行此操作。

企業經常使用 Amazon 來開 WorkSpaces 發容器型應用程式。這需要 Amazon Elastic Compute Cloud (Amazon EC2) 或 AWS Fargate 與 Amazon ECS 集群來測試和運行 ECS 任務。現在，透過使用 Amazon ECS Anywhere，您可以將 Amazon WorkSpaces 做為外部執行個體直接新增到 ECS 叢集，而且可以直接執行您的任務。這可以縮短您的開發時間，因為您可以在 Amazon WorkSpaces 上使用 ECS 叢集來測試容器。您也可以節省使用 EC2 或 Fargate 執行個體測試容器應用程式的成本。

這種模式展示了如何 WorkSpaces 使用 Amazon ECS 在亞馬遜任何地方部署 Amazon ECS Anywhere 務。它會設定 ECS 叢集，並使用 AWS Directory Service Simple AD 來啟動 WorkSpaces. 然後，範例 ECS 工作會在中啟動 NGINX。 WorkSpaces

先決條件和限制

- 有效的 AWS 帳戶
- AWS 命令列界面 (AWS CLI)
- 您的機器上設定的 AWS 登入資料

架構

目標技術堆疊

- 虛擬私有雲 (VPC)
- Amazon ECS 集群
- Amazon WorkSpaces
- 使用 Simple AD 的 AWS Directory Service

目標架構

該架構包括以下服務和資源：

- 在自訂 VPC 中具有公用和私有子網路的 ECS 叢集
- VPC 中的 Simple AD 可讓使用者存取 Amazon WorkSpaces
- Amazon 使用簡單的 AD 在 VPC 中 WorkSpaces 佈建
- AWS Systems Manager 已啟動將 Amazon 新增 WorkSpaces 為受管執行個體
- 使用 Amazon ECS 和 AWS 系統管理器代理（SSM 代理），Amazon WorkSpaces 添加到系統管理器和 ECS 集群
- 要在 ECS 叢集中執行的 ECS WorkSpaces 工作範例

工具

- AWS Directory Service 簡易作用中目錄 (Simple AD) 是獨立的受管目錄，由 Samba 4 作用中目錄相容伺服器提供支援。Simple AD 提供 AWS 受管 Microsoft AD 所提供的功能子集，包括管理使用者和安全連線至 Amazon EC2 執行個體的功能。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。
- Amazon WorkSpaces 幫助您為您的用戶佈建虛擬，基於雲的 Microsoft Windows 或 Amazon Linux 桌面，稱為 WorkSpaces. WorkSpaces 無需採購和部署硬體或安裝複雜的軟體。

史诗

設定 ECS 叢集

任務	描述	所需技能
建立和配置 ECS 叢集。	若要建立 ECS 叢集，請遵循 AWS 文件中的指示，包括下列步驟： • 對於選取叢集相容性，請選擇僅聯網，這將支援 Amazon WorkSpace 做為 ECS 叢集的外部執行個體。 • 選擇建立新的 VPC。	雲端架構師

推出 Amazon WorkSpaces

任務	描述	所需技能
設置 Simple AD 並啟動 Amazon WorkSpaces。	若要為新建立的 VPC 佈建 Simple AD 目錄並啟動 Amazon WorkSpaces，請遵循 AWS 文件中的指示。	雲端架構師

為混合式環境設定 AWS Systems Manager

任務	描述	所需技能
下載附加的腳本。	在本機電腦上，下載「附件」區段中的`ssm-trust-policy.json` 和`ssm-activation.json` 檔案。	雲端架構師
新增 IAM 角色。	根據您的業務需求新增環境變數。	雲端架構師

任務	描述	所需技能
	`export AWS_DEFAU LT_REGION=${AWS_RE GION_ID}` `export ROLE_NAME= ${ECS_TASK_ROLE}` `export CLUSTER_NAME= ${ECS_CLUSTER_NAME}` `export SERVICE_N AME=${ECS_CLUSTER_ SERVICE_NAME}` 執行下列命令。 `aws iam create-role -- role-name $ROLE_NAME --assume-role-policy- document file://ssm- trust-policy.json`	
將亞馬遜 SSM ManagedIn stanceCore 政策添加到 IAM 角色。	執行下列命令。 `aws iam attach-role- policy --role-name $ROLE_NAME --policy- arn arn:aws:iam::aws:p olicy/AmazonSSMMan agedInstanceCore`	雲端架構師
將亞馬遜 ContainerServicefo r EC2 角色政策添加到 IAM 角色。	執行下列命令。 `aws iam attach-role- policy --role-name $ROLE_NAME --policy- arn arn:aws:iam::aws:p olicy/service-role /AmazonEC2Containe rServiceforEC2Role`	雲端架構師

任務	描述	所需技能
驗證 IAM 角色。	若要驗證 IAM 角色，請執行下列命令。 `aws iam list-attached-role-policies --role-name $ROLE_NAME`	雲端架構師
啟用 Systems Manager。	執行下列命令。 `aws ssm create-activation --iam-role $ROLE_NAME | tee ssm-activation.json`	雲端架構師

新增 WorkSpaces 至 ECS 叢集

任務	描述	所需技能
Connect 到您的 WorkSpaces.	若要連接並設定您的工作區，請遵循 AWS 文件中的指示。	應用程式開發人員
下載 EC-任何地方安裝腳本。	在命令提示中，執行下列 命令。 `curl -o "ecs-anywhere-install.sh" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh" && sudo chmod +x ecs-anywhere-install.sh`	應用程式開發人員
檢查外圍程序檔的完整性。	(選擇性) 執行下列命令。	應用程式開發人員

任務	描述	所需技能
	`curl -o "ecs-anywhere-install.sh.sha256" "https://amazon-ecs-agent-packages-preview.s3.us-east-1.amazonaws.com/ecs-anywhere-install.sh.sha256" && sha256sum -c ecs-anywhere-install.sh.sha256`	
在 Amazon Linux 上新增 EPEL 儲存庫。	若要新增企業 Linux (EPEL) 儲存區域的額外套件，請執行命令`sudo amazon-linux-extras install epel -y`。	應用程式開發人員
在任何地方安裝亞馬遜 ECS。	若要執行安裝指令碼，請使用下列命令。 `sudo ./ecs-anywhere-install.sh --cluster $CLUSTER_NAME --activation-id $ACTIVATION_ID --activation-code $ACTIVATION_CODE --region $AWS_REGION`	

任務	描述	所需技能
檢查 ECS 叢集中的執行個體資訊。	若要檢查 Systems Manager 和 ECS 叢集執行個體資訊，並驗證 WorkSpaces 已新增至叢集，請從本機電腦執行下列命令。 `aws ssm describe-instance-informati on" && "aws ecs list-container-instances --cluster $CLUSTER_NAME`	應用程式開發人員

新增一個 ECS 任務 WorkSpaces

任務	描述	所需技能
建立工作執行 IAM 角色。	下載`task-execution-assume-role.json` 和`external-task-definition.json` 從附件部分。 在您的本機電腦上，執行下列命令。 `aws iam --region $AWS_DEFAULT_REGIO N create-role --role-name $ECS_TASK _EXECUTION_ROLE --assume-role-policy-document file://ta sk-execution-assume-role.json`	雲端架構師
將原則新增至執行角色。	執行下列命令。	雲端架構師

任務	描述	所需技能
	`aws iam --region $AWS_DEFAULT_REGION attach-role-policy --role-name $ECS_TASK_EXECUTION_ROLE --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy`	
建立工作角色。	執行下列命令。 `aws iam --region $AWS_DEFAULT_REGION create-role --role-name $ECS_TASK_EXECUTION_ROLE --assume-role-policy-document file://task-execution-assume-role.json`	雲端架構師
將任務定義註冊到叢集。	在您的本機電腦上，執行下列命令。 `aws ecs register-task-definition --cli-input-json file://external-task-definition.json`	雲端架構師

任務	描述	所需技能
執行工作。	在您的本機電腦上，執行下列命令。 `aws ecs run-task --cluster $CLUSTER_NAME --launch-type EXTERNAL --task-definition nginx`	雲端架構師
驗證工作執行狀態。	若要擷取工作 ID，請執行下列命令。 `export TEST_TASKID=$(aws ecs list-tasks --cluster $CLUSTER_NAME | jq -r '.taskArns[0]')` 使用工作 ID，執行下列命令。 `aws ecs describe-tasks --cluster $CLUSTER_NAME --tasks ${TEST_TASKID}`	雲端架構師
驗證上的工作 WorkSpace。	若要檢查 NGINX 是否在上執行 WorkSpace，請執行命令。 `curl http://localhost:8080`	應用程式開發人員

相關資源

- ECS 叢集
- 設定混合式環境
- Amazon WorkSpaces
- 簡易 AD

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在 Amazon EC2 Linux 實例上運行一個 ASP.NET 核心網絡 API 碼頭容器

由阿南德·拉馬林甘（AWS）和拜縣斯雷拉克斯米（AWS）創建

環境：PoC 或試點	技術：容器與微服務；軟體開發與測試；Web 和行動應用程式	工作量：Microsoft
AWS 服務：Amazon EC2；Elastic Load Balancing (ELB)		

Summary

此模式適用於開始在 Amazon Web Services (AWS) 雲端上容器化應用程式的人員。當您開始在雲端上將應用程式容器化時，通常沒有設定容器協調平台。此模式可協助您在 AWS 上快速設定基礎設施，以測試容器化應用程式，而不需要精心設計的容器協調基礎設施。

現代化之旅的第一步是轉換應用程式。如果它是傳統的 .NET 框架應用程序，則必須首先將運行時更改為 ASP.NET 核心。然後執行下列動作：

- 創建碼頭容器映像
- 使用構建的映像運行 Docker 容器
- 在任何容器協調平台上部署應用程式之前先驗證應用程式，例如亞馬遜彈性容器服務 (Amazon ECS) 或亞馬遜彈性 Kubernetes 服務 (Amazon EKS)。

此模式涵蓋在 Amazon 彈性運算雲端 (Amazon EC2) Linux 執行個體上建置、執行和驗證現代應用程式開發的各個層面。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶
- 具有足夠存取權限的 AWS Identity and Access Management (IAM) 角色，可針對此模式建立 AWS 資源

- 視覺工作室社區 2022 年或更高版本下載並安裝
- 一個 .NET 框架項目現代化為 ASP.NET 核心
- 一個 GitHub 存儲庫

產品版本

- 視覺工作室社區 2022 或更高版本

架構

目標架構

此模式使用 AWS CloudFormation 範本建立如下圖所示的高可用性架構。Amazon EC2 Linux 執行個體是在私有子網路中啟動的。AWS Systems Manager 器會話管理器用於訪問私有的 Amazon EC2 Linux 實例，並測試在 Docker 容器中運行的 API。

1. 透過工作階段管理員存取 Linux 執行個體

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) — AWS 命令列界面 (AWS CLI) 是一種開放原始碼工具，可透過命令列殼層中的命令與 AWS 服務互動。只要使用最少的組態，您就可以執行 AWS CLI 命令，以實作與瀏覽器型 AWS 管理主控台所提供的功能相同。
- AWS 管理主控台 — AWS 管理主控台是一種 Web 應用程式，其中包含和參照用於管理 AWS 資源的廣泛服務主控台集合。若是首次登入，這時主控台頁面將會顯示。首頁可讓您存取每個服務主控台，並提供單一位置來存取執行 AWS 相關任務所需的資訊。
- AWS Systems Manager 工作階段管理員 — 工作階段管理員是全受管 AWS Systems Manager 功能。使用工作階段管理員，您可以管理 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。工作階段管理員提供安全且可稽核的節點管理，無需開啟輸入連接埠、維護防禦主機或管理安全殼層金鑰。

其他工具

- 視覺工作室 2022 — 視覺工作室 2022 是一個整合式開發環境 (IDE)。
- Docker — D ocker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中交付軟體。

Code

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
 WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj", "DemoNetCoreWebAPI/"]
RUN dotnet restore "DemoNetCoreWebAPI/DemoNetCoreWebAPI.csproj"
COPY . .
WORKDIR "/src/DemoNetCoreWebAPI"
RUN dotnet build "DemoNetCoreWebAPI.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "DemoNetCoreWebAPI.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "DemoNetCoreWebAPI.dll"]
```

史诗

開發核心網頁應用程式介面

任務	描述	所需技能
使用視覺工作室創建一個示例 ASP.NET 核心網絡 API。	要創建一個示例 ASP.NET 核心 Web API，請執行以下操作： 1. 2022 年開放視覺工作室。 2. 選擇 Create new project (建立新的專案)。	應用程式開發人員

任務	描述	所需技能
	3. 選取 ASP.NET 核心網頁 API 專案範本，然後選擇 [下一步]。 4. 針對專案名稱，輸入 DemoNetCoreWebAPI，然後選擇「下一步」。 5. 選擇建立。 6. 若要在本端執行專案，請按 F5。 7. 確認預設 WeatherForecastAPI 端點正在使用 Swagger 傳回結果。 8. 打開命令提示符，導航到 .csproj 項目文件夾，然後運行以下命令以將新的 Web API 推送到存儲庫。 GitHub `git add --all` `git commit -m “Initial Version”` `git push`	

任務	描述	所需技能
建立 Dockerfile。	若要建立 Docker 檔案，請執行下列其中一個動作： • 使用「代碼」部分中的示例 Docker 文件手動創建 Docker 文件。根據需求，選取適當的 .NET 基礎映像。如需 .NET 和 ASP.NET 核心相關映像檔的詳細資訊，請參閱碼頭集線器。 • 使用視覺工作室和碼頭桌面創建碼頭文件。在解決方案資源管理器中，右鍵單擊該項目，選擇添加-> Docker Support。針對目標作業系統，選取 Linux。確保新的 Docker 文件與解決方案文件（.sln）位於相同的路徑中。 若要將變更推送至儲 GitHub 存庫，請執行下列命令。 `git add --all` `git commit -m “Dockerfile added”` `git push`	應用程式開發人員

設定 Amazon EC2 Linux 執行個體

任務	描述	所需技能
設定基礎結構。	啟動 AWS CloudFormation 範本以建立基礎設施，其中包括下列項目： • 使用 AWS VPC 快速入門的虛擬私有雲端 (VPC)，其中包含跨越兩個可用區域的兩個公有子網路和兩個私有子網路。 • 啟用 AWS Systems Manager 所需的 IAM 角色。 • 在其中一個私有子網路中，使用具有最新 SSM 代理程式的 Amazon Linux 2 示範執行個體。雖然這個執行個體沒有任何來自網際網路的直接連線，但是可以使用 AWS Systems Manager 工作階段管理員安全地存取它，而不需要防禦主機。 若要進一步了解如何使用工作階段管理員存取私有 Amazon EC2 執行個體，而不需要防禦主機，請參閱邁向無堡壘的世界部落格文章。	應用程式開發人員、AWS 管理員、AWS DevOps
登入 Amazon EC2 Linux 執行個體。	若要連線到私有子網路中的 Amazon EC2 Linux 執行個體，請執行下列動作： 1. 開啟 Amazon EC2 主控台。	應用程式開發人員

任務	描述	所需技能
	2. 在導覽窗格中，選擇 Instances (執行個體)。 3. 選取 Amazon Linux 2 示範執行個體，然後選擇 Connect。 4. 選擇 Session Manager (工作階段管理員)。 5. 選擇 [Connect] 以開啟新的終端機視窗。 6. 執行下列命令。 `sudo su`	
安裝並啟動泊塢視窗。	若要在 Amazon EC2 Linux 執行個體中安裝和啟動泊塢視窗，請執行以下操作： 1. 要安裝 Docker，請運行以下命令。 `yum install -y docker` 2. 若要啟動 Docker 服務，請執行下列命令。 `service docker start` 3. 若要驗證 Docker 安裝，請執行下列命令。 `docker info`	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
安裝 Git 並克隆存儲庫。	若要在 Amazon EC2 Linux 執行個體上安裝 Git 並從中複製儲存庫 GitHub，請執行下列動作。 1. 要安裝 Git，請運行以下命令。 `yum install git -y` 2. 若要複製存放庫，請執行下列命令。 `git clone https://github.com/<username>/<repo-name>.git` 3. 若要導覽至 Docker 檔案，請執行下列命令。 `cd <repo-name>/DemoNetCoreWebAPI/`	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
構建並運行碼頭容器。	若要在 Amazon EC2 Linux 執行個體內建置 Docker 映像檔並執行容器，請執行下列動作： 1. 若要建立 Docker 映像檔，請執行下列命令。 `docker build -t aspnetcorewebapiimage -f Dockerfile .` 2. 若要檢視所有 Docker 映像檔，請執行下列命令。 `docker images` 3. 若要建立並執行容器，請執行下列命令。 `docker run -d -p 80:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage`	應用程式開發人員、AWS 管理員、AWS DevOps

測試網頁 API

任務	描述	所需技能
使用 curl 命令測試網頁 API。	若要測試 Web API，請執行下列命令。 `curl -X GET "http://localhost/WeatherFo`	應用程式開發人員

任務	描述	所需技能
	`recast" -H  "accept: text/plain"` 驗證 API 回應。 注意：當您在本地運行時，您可以從 Swagger 獲取每個端點的 curl 命令。	

清除資源

任務	描述	所需技能
刪除所有資源。	刪除堆疊以移除所有資源。這樣可確保您不會為未使用的任何服務付費。	AWS 管理員 DevOps

相關資源

- 使用 PuTTY 從視窗 Connect 到您的 Linux 執行個體
- 創建一個網絡 API 與 ASP.NET 核心
- 走向一個沒有堡壘的世界

使用 AWS Fargate 大規模執行訊息導向工作負載

創建者：斯坦·祖巴雷夫 (AWS)

環境：PoC 或試點	技術：容器與微服務；訊息與通訊；資料庫	AWS 服務：AWS Fargate；Amazon SQS；Amazon DynamoDB

Summary

此模式示範如何使用容器和 AWS Fargate 在 AWS 雲端中大規模執行訊息導向工作負載。

當應用程式處理程序的資料量超過以功能為基礎的無伺服器運算服務的限制時，使用容器處理資料會很有幫助。例如，如果應用程式需要的運算容量或處理時間比 AWS Lambda 提供的更多，則使用 Fargate 可以改善效能。

下列範例設定使用 AWS Cloud Development Kit (AWS CDK) 在 TypeScript AWS 雲端中設定和部署下列資源：

- Fargate 服務
- Amazon Simple Queue Service (Amazon SQS) 隊列
- 一個 Amazon DynamoDB 表。
- Amazon CloudWatch 儀表板

Fargate 服務從 Amazon SQS 佇列接收和處理訊息，然後將它們存放在 Amazon DynamoDB 表格中。您可以使用儀表板監控要處理多少 Amazon SQS 訊息，以及 Fargate 建立了多少 DynamoDB 項目。CloudWatch

附註：您也可以使用此模式的範例程式碼，在事件驅動的無伺服器架構中建置更複雜的資料處理工作負載。如需詳細資訊，請參閱使用 AWS Fargate 大規模執行事件驅動和排程工作負載。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 最新版本的 AWS Command Line Interface (AWS CLI) (AWS CLI)，已在本機電腦上安裝和設定
- Git，在本地計算機上安裝和配置
- 在本機電腦上安裝和設定的 AWS CDK
- 在本地計算機上進行，安裝和配置
- Docker，在本地計算機上安裝和配置

架構

目標技術堆疊

- Amazon SQS
- AWS Fargate
- Amazon DynamoDB

目標架構

下圖顯示使用 Fargate 在 AWS 雲端大規模執行訊息導向工作負載的範例工作流程：

該圖顯示以下工作流程：

1. Fargate 服務使用 Amazon SQS 長輪詢來接收來自 Amazon SQS 佇列的訊息。
2. 接著，Fargate 服務會處理 Amazon SQS 訊息，並將它們存放在 DynamoDB 表格中。

自動化和規模

若要自動擴展 Fargate 任務計數，您可以設定 Amazon Elastic Container Service (Amazon ECS) 服務自 Auto Scaling。最佳做法是根據應用程式 Amazon SQS 佇列中的可見訊息數量來設定擴展政策。

如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的根據 Amazon SQS 進行擴展。

工具

AWS 服務

- AWS Fargate 可協助您執行容器，而不需要管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。它與 Amazon Elastic Container Service (Amazon ECS) 一起使用。

- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。

Code

此模式的代碼可在 GitHub sqs-fargate-ddb-cdk-go 存儲庫中找到。

史诗

使用 AWS CDK 建立和部署資源

任務	描述	所需技能
克隆存 GitHub 儲庫。	執行下列命令，將 GitHub sqs-fargate-ddb-cdk-go 儲存庫複製到您的本機電腦： `git clone https://github.com/aws-samples/sqs-fargate-ddb-cdk-go.git`	應用程式開發人員
確認 AWS CLI 已設定為正確的 AWS 帳戶，而且 AWS CDK 具有所需的許可。	若要檢查您的 AWS CLI 組態設定是否正確，您可以執行以下亞馬遜簡單儲存服務 (Amazon S3) ls 命令： `aws s3 ls` 此程序還要求 AWS CDK 具有在 AWS 帳戶內佈建基礎設施的許可。若要授予所需的許可，您必須在 AWS CLI 中建立具名的 AWS 設定檔，並將其匯出為 AWS_PRO FILE 環境變數。	應用程式開發人員

任務	描述	所需技能
	注意：如果您之前尚未在 AWS 帳戶中使用過 AWS CDK，則必須先佈建所需的 AWS CDK 資源。如需詳細資訊，請參閱 AWS CDK v2 開發人員指南中的啟動安裝。	
將 AWS CDK 堆疊部署到您的 AWS 帳戶。	1. 執行下列 AWS CLI 命令來建立容器映像： `docker build -t go-fargate .` 2. 執行下列命令以開啟 AWS CDK 目錄： `cd cdk` 3. 執行下列命令來安裝所需的 npm 模組： `npm i` 4. 執行下列命令，將 AWS CDK 模式部署到您的 AWS 帳戶： `cdk deploy --profile ${AWS_PROFILE}`	應用程式開發人員

測試設定

任務	描述	所需技能
將測試訊息傳送到 Amazon SQS 佇列。	如需指示，請參閱 Amazon SQS 開發人員指南中的將訊息傳送到佇列 (主控台)。 測試 Amazon SQS 訊息範例	應用程式開發人員

任務	描述	所需技能
	<pre>{ "message": "hello, Fargate" }</pre>	
確認測試訊息是否出現在 Fargate 服務的 CloudWatch 記錄檔中。	請遵循 Amazon ECS 開發人員指南中檢視 CloudWatch 日誌中的指示。請務必檢閱 go-service-clusterECS 叢集中記go-fargate-service錄群組的記錄。	應用程式開發人員
確認測試訊息是否顯示在 DynamoDB 表中。	1. 開啟 DynamoDB 主控台。 2. 在左側導覽窗格中，選擇 Tables (資料表)。然後，從清單中選取下列表格：sqs-fargate-ddb-table。 3. 選擇 探索資料表項目 。 4. 確認測試訊息出現在 [傳回的項目] 清單中。	應用程式開發人員
驗證 Fargate 服務正在將消息發送到 CloudWatch 日誌。	1. 開啟 CloudWatch 主控台。 2. 在左側導覽窗格中，選擇 [儀表板]。 3. 在「自訂儀表板」清單中，選取名為的儀表板go-service-dashboard。 4. 確認測試訊息出現在記錄檔中。 注意：AWS CDK 會自動在您的 AWS 帳戶中建立 CloudWatch 儀表板。	應用程式開發人員

清除

任務	描述	所需技能
刪除 AWS CDK 堆疊。	1. 執行下列命令，在 AWS CLI 中開啟您的 AWS CDK 目錄： `cd cdk` 2. 執行下列命令以刪除 AWS CDK 堆疊： `cdk destroy --profile ${AWS_PROFILE}`	應用程式開發人員
確認已刪除 AWS CDK 堆疊。	若要確定已刪除堆疊，請執行下列命令： `aws cloudformation list-stacks --query \ "StackSummaries[?contains(StackName,'SqsFargate')].StackStatus" \ --profile ${AWS_PROFILE}` 命令輸出中返回的`StackStatus` 值是`DELETE_COMPLETE` 如果堆棧被刪除。 如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的描述和列出堆疊。	應用程式開發人員

相關資源

- 設定 AWS CLI (第 2 版的 AWS CLI 使用者指南)
- API 參考 (AWS CDK API 參考)
- 適用於 Go v2 的 AWS 開發套件 (Go 文件)

搭配 AWS Fargate 使用 Amazon EKS 上的 Amazon EFS，以持續性資料儲存執行可設定狀態工作負載

由里卡多·莫拉斯（AWS），羅德里戈貝爾薩（AWS）和盧西奧·佩雷拉（AWS）創建

代碼存儲庫：Amazon EKS 與 Fargate 和 Amazon EFS	環境：PoC 或試點	技術：容器與微服務；儲存與備份
工作負載：開源	AWS 服務：Amazon EFS；Amazon EKS；AWS Fargate	

Summary

此模式提供有關啟用 Amazon Elastic File System (Amazon EFS) 作為在 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 上執行之容器的儲存裝置的指導，方法是使用 AWS Fargate 佈建您的運算資源。

此模式中描述的設定遵循安全性最佳做法，並預設提供靜態安全性和傳輸中的安全性。若要加密 Amazon EFS 檔案系統，它會使用 AWS Key Management Service (AWS KMS) 金鑰，但您也可以指定一個金鑰別名來分派建立 KMS 金鑰的程序。

您可以按照此模式中的步驟為 proof-of-concept (PoC) 應用程式建立命名空間和 Fargate 設定檔、安裝用於將 Kubernetes 叢集與 Amazon EFS 整合的 Amazon EFS 容器儲存界面 (CSI) 驅動程式、設定儲存類別，以及部署 PoC 應用程式。這些步驟會產生 Amazon EFS 檔案系統，該系統會在多個 Kubernetes 工作負載之間共用，並透過 Fargate 執行。該模式伴隨著可以自動執行這些步驟的腳本。

如果您想要在容器化應用程式中保存資料，並希望避免在擴展作業期間遺失資料，則可以使用此模式。例如：

- DevOps 工具 — 常見的案例是制定持續整合和持續交付 (CI/CD) 策略。在這種情況下，您可以使用 Amazon EFS 做為共用檔案系統，在 CI/CD 工具的不同執行個體之間存放組態，或為 CI/CD 工具的不同執行個體之間的管道階段存放快取 (例如 Apache Maven 儲存庫)。
- 網頁伺服器 — 常見的案例是使用 Apache 作為 HTTP 網頁伺服器。您可以使用 Amazon EFS 做為共用檔案系統來存放在 Web 伺服器的不同執行個體之間共用的靜態檔案。在此範例案例中，修改會直接套用至檔案系統，而不是將靜態檔案複製到 Docker 映像中。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有 1.17 版或更新版本的現有 Amazon EKS 叢集 (測試至 1.27 版)
- 現有的 Amazon EFS 檔案系統，可繫結 Kubernetes StorageClass 並動態佈建檔案系統
- 叢集管理權限
- 設定為指向所需 Amazon EKS 叢集的內容

限制

- 當您將 Amazon EKS 與 Fargate 搭配使用時，需要考慮一些限制。例如，不支援使用某些 Kubernetes 建構 (例如 DaemonSets 和有權限的容器)。如需有關 Fargate 限制的詳細資訊，請參閱 Amazon EKS 文件中的 AWS Fargate 考量事項。
- 此模式提供的程式碼支援執行 Linux 或 macOS 的工作站。

產品版本

- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版或更新版本
- Amazon EFS CSI 驅動程式 1.0 版或更新版本 (測試至 2.4.8 版)
- 版本為 0.24.0 或更高版本（測試至 0.158.0 版本）
- jq 版本 1.6 或更新版本
- 庫貝克特爾版本 1.17 或更新版本（測試至 1.27 版）
- 庫伯尼特斯版本 1.17 或更新版本（測試至 1.27 版）

架構

目標架構由下列基礎結構組成：

- 虛擬私有雲 (VPC)
- 兩個可用區域

- 具有 NAT 閘道的公用子網路，可提供網際網路存取
- 具有 Amazon EKS 叢集和 Amazon EFS 掛接目標 (也稱為掛接點) 的私有子網路
- VPC 層級的 Amazon EFS

以下是 Amazon EKS 叢集的環境基礎設施：

- 可在命名空間層級容納 Kubernetes 建構的 AWS Fargate 設定檔
- 具有以下內容的庫伯尼特斯命名空間：
  - 跨可用區域分配兩個應用程式網繭
  - 在叢集層級繫結至持續性磁碟區 (PV) 的一個持續性磁碟區宣告 (PVC)
- 與命名空間中的 PVC 繫結，並指向叢集外部私有子網路中 Amazon EFS 掛載目標的全叢集 PV

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可用來從命令列與 AWS 服務互動。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。在這種模式中，它提供了一個簡單、可擴展、全受管和共用的檔案系統，以搭配 Amazon EKS 使用。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或操作自己的叢集。
- AWS Fargate 是 Amazon EKS 的無伺服器運算引擎。它會為您的 Kubernetes 應用程式建立和管理運算資源。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。

其他工具

- Docker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中提供軟體。
- eksctl 是一個命令列公用程式，用於在 Amazon EKS 上建立和管理 Kubernetes 叢集。
- kubectl 是一種命令列介面，可協助您針對 Kubernetes 叢集執行命令。
- jq 是用於解析 JSON 的命令行工具。

Code

此模式的程式碼是使用 AWS Fargate GitHub 存放庫在 Amazon EKS 上使用 Amazon EFS 的持續性組態中提供。這些腳本由 Epic 組織，在文件夾中epic01通過epic06，對應於此模式中史詩部分中的順序。

最佳實務

目標架構包括下列服務和元件，並遵循 AWS Well-Architected Framework 構最佳實務：

- Amazon EFS 提供簡單、可擴展、全受管的彈性 NFS 檔案系統。在網繭中執行的 PoC 應用程式的所有複寫中，這些複寫會分散在所選 Amazon EKS 叢集的私有子網路中，作為共用檔案系統。
- 每個私有子網路的 Amazon EFS 掛載目標。這可在叢集的虛擬私有雲 (VPC) 內為每個可用區域提供備援。
- Amazon EKS，用於執行庫伯尼特工作負載。您必須先佈建 Amazon EKS 叢集，然後才能使用此模式，如先決條件一節中所述。
- AWS KMS 可為存放在 Amazon EFS 檔案系統中的內容提供靜態加密。
- Fargate 可管理容器的運算資源，讓您可以專注於業務需求，而不是基礎結構負擔。會針對所有私有子網路建立 Fargate 設定檔。它提供叢集虛擬私有雲 (VPC) 內每個可用區域的備援。
- Kubernetes 網繭，用於驗證內容是否可由不同的應用程式執行個體共用、使用和寫入。

史诗

佈建 Amazon EKS 叢集 (選用)

任務	描述	所需技能
建立 Amazon EKS 叢集。	如果您已部署叢集，請跳至下一個史詩。在您現有的 AWS 帳戶中建立 Amazon EKS 叢集。在GitHub 存放庫目錄中，使用其中一種模式，透過使用地形或 eksctl 來部署 Amazon EKS 叢集。如需詳細資訊，請參閱 Amazon EKS 文件中的建立 Amazon EKS 叢集。備註：在 Terraform 模式中，還有一些範例說明如何：將 Fargate	AWS 管理員、地形或其他管理員、Kubernetes 管理員

任務	描述	所需技能
	設定檔連結到 Amazon EKS 叢集、建立 Amazon EFS 檔案系統，以及在 Amazon EKS 叢集中部署 Amazon EFS CSI 驅動程式。	

任務	描述	所需技能
匯出環境變數。	執行 env.sh 指令碼。這會提供後續步驟所需的資訊。 `source ./scripts/env.sh` `Inform the AWS Account ID:` `<13-digit-account-id>` `Inform your AWS Region:` `<aws-Region-code>` `Inform your Amazon EKS Cluster Name:` `<amazon-eks-cluster-name>` `Inform the Amazon EFS Creation Token:` `<self-genereated-uuid>` 如果還沒有註明，您可以使用以下 CLI 命令獲取上述請求的所有信息。 `# ACCOUNT ID` `aws sts get-caller-identity --query "Account" --output text` `# REGION CODE` `aws configure get region` `# CLUSTER EKS NAME` `aws eks list-clusters --query "clusters" --output text` `# GENERATE EFS TOKEN`	AWS 系統管理員

任務	描述	所需技能
	`uuidgen`	

建立 Kubernetes 命名空間和連結的 Fargate 設定檔

任務	描述	所需技能
為應用程式工作負載建立 Kubernetes 命名空間和 Fargate 設定檔。	建立命名空間以接收與 Amazon EFS 互動的應用程式工作負載。執行 `create-k8s-ns-and-linked-fargate-profile.sh` 指令碼。您可以選擇使用自訂命名空間名稱或預設提供的命名空間`poc-efs-eks-fargate` 。 使用自訂應用程式命名空間名稱： `export $APP_NAMESPACE=<CUSTOM_NAME>` `./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \` `-c "$CLUSTER_NAME" -n "$APP_NAMESPACE"` 沒有自訂應用程式命名空間名稱： `./scripts/epic01/create-k8s-ns-and-linked-fargate-profile.sh \` `-c "$CLUSTER_NAME"`	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
	其中$CLUSTER_NAME 是您的 Amazon EKS 集群的名稱。該-n <NAMESPACE> 參數是可選的; 如果沒有通知，將提供默認生成的命名空間名稱。	

建立 Amazon EFS 檔案系統

任務	描述	所需技能
產生唯一的權杖。	Amazon EFS 需要建立權杖以確保冪等作業 (使用相同建立權杖呼叫作業沒有任何作用)。為了滿足此要求，您必須通過可用的技術生成唯一令牌。例如，您可以生成一個通用唯一標識符（UUID）以用作創建令牌。	AWS 系統管理員
建立一個 Amazon EFS 檔案系統。	建立檔案系統，以接收應用程式工作負載讀取和寫入的資料檔案。您可以建立加密或非加密的檔案系統。最佳作法是，此模式的程式碼會建立加密系統，預設啟用靜態加密。) 您可以使用唯一的對稱 AWS KMS 金鑰來加密檔案系統。如果未指定自訂金鑰，則會使用 AWS 受管金鑰。 在您為 Amazon EFS 產生唯一的權杖之後，請使用 create-efs.sh 指令碼建立加密或非加密的 Amazon EFS 檔案系統。	AWS 系統管理員

任務	描述	所需技能
	使用靜態加密，不使用 KMS 金鑰： `./scripts/epic02/create-efs.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"` 其中$CLUSTER_NAME 是 Amazon EKS 叢集的名稱，$EFS_CREATION_TOKEN 是檔案系統的唯一建立權杖。 使用 KMS 金鑰的靜態加密功能： `./scripts/epic02/create-efs.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN" \ -k "$KMS_KEY_ALIAS"` 其中$CLUSTER_NAME 是 Amazon EKS 叢集的名稱，$EFS_CREATION_TOKEN 是檔案系統的唯一建立權杖，$KMS_KEY_ALIAS 是 KMS 金鑰的別名。 沒有加密： `./scripts/epic02/create-efs.sh -d \`	

任務	描述	所需技能
	`-c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"` 其中`$CLUSTER_NAME` 是 Amazon EKS 叢集的名稱，`$EFS_CREATION_TOKEN` 是檔案系統的唯一建立權杖，並-d停用靜態加密。	
建立安全群組。	建立安全群組以允許 Amazon EKS 叢集存取 Amazon EFS 檔案系統。	AWS 系統管理員
更新安全性群組的輸入規則。	更新安全性群組的輸入規則，以允許下列設定的傳入流量： • 通訊協定 — 連接埠 2049 • 來源 — VPC 中包含 Kubernetes 叢集之私有子網路的 CIDR 區塊範圍	AWS 系統管理員
為每個私有子網路新增掛載目標。	針對 Kubernetes 叢集的每個私人子網路，為檔案系統和安全性群組建立掛載目標。	AWS 系統管理員

將 Amazon EFS 元件安裝到叢集

任務	描述	所需技能
部署 Amazon EFS CSI 驅動程式。	將 Amazon EFS CSI 驅動程式部署到叢集中。該驅動程序根據應用程序創建的持久性卷聲明佈建存儲。執行指`create-k8s-efs-csi-sc.sh` 令碼	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
	以將 Amazon EFS CSI 驅動程式和儲存類別部署到叢集中。 `./scripts/epic03/create-k8s-efs-csi-sc.sh` 此指令碼使用`kubectl`公用程式，因此請確定已設定內容，並指向所需的 Amazon EKS 叢集。	
部署儲存類別。	將儲存類別部署到 Amazon EFS 佈建程式的叢集中。	具有授與權限的 Kubernetes 使用者

將 PoC 應用程式安裝到 Kubernetes 叢集

任務	描述	所需技能
部署持續性磁碟區。	部署持續性磁碟區，並將其連結至建立的儲存類別和 Amazon EFS 檔案系統的 ID。應用程式會使用持續性磁碟區來讀取和寫入內容。您可以在儲存區欄位中為持續性磁碟區指定任何大小。Kubernetes 需要此欄位，但由於 Amazon EFS 是彈性檔案系統，因此不會強制執行任何檔案系統容量。您可以使用或不使用加密來部署持續性磁碟區。Amazon EFS CSI 驅動程式預設會啟用加密，這是最佳實務。) 執行指`deploy-poc-app.sh` 令碼以部署持續性磁碟區、持續	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
	性磁碟區宣告和兩個工作負載。 使用傳輸中的加密： `./scripts/epic04/deploy-poc-app.sh \ -t "$EFS_CREATION_TOKEN"` 其中$EFS_CREATION_TOKEN 是檔案系統的唯一建立權杖。 傳輸中沒有加密： `./scripts/epic04/deploy-poc-app.sh -d \ -t "$EFS_CREATION_TOKEN"` 其中$EFS_CREATION_TOKEN 是檔案系統的唯一建立權杖，並-d停用傳輸過程中的加密。	
部署應用程式要求的持續性磁碟區宣告。	部署應用程式要求的持續性磁碟區宣告，並將其連結至儲存區類別。使用與先前建立的持續性磁碟區相同的存取模式。您可以在儲存區欄位中為持續性磁碟區宣告指定任何大小。Kubernetes 需要此欄位，但由於 Amazon EFS 是彈性檔案系統，因此不會強制執行任何檔案系統容量。	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
部署工作負載 1.	部署代表應用程式工作負載 1 的網繭。此工作負載會將內容寫入檔案/data/out 1.txt 。	具有授與權限的 Kubernetes 使用者
部署工作負載 2.	部署代表應用程式工作負載 2 的網繭。此工作負載會將內容寫入檔案/data/out 2.txt 。	具有授與權限的 Kubernetes 使用者

驗證檔案系統的持續性、耐久性和可共用性

任務	描述	所需技能
檢查的狀態PersistentVolume 。	輸入下列命令以檢查的狀態PersistentVolume 。 `kubectl get pv` 如需範例輸出，請參閱其他資訊一節。	具有授與權限的 Kubernetes 使用者
檢查的狀態PersistentVolumeClaim 。	輸入下列命令以檢查的狀態PersistentVolumeClaim 。 `kubectl -n poc-efs-eks-fargate get pvc` 如需範例輸出，請參閱其他資訊一節。	具有授與權限的 Kubernetes 使用者
驗證工作負載 1 可以寫入檔案系統。	輸入以下命令以驗證工作負載 1 是否正在寫入/data/out 1.txt 。	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
	`kubectl exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -f /data/out1.txt` 結果類似於以下內容： `...` `Thu Sep  3 15:25:07 UTC 2023 - PoC APP 1` `Thu Sep  3 15:25:12 UTC 2023 - PoC APP 1` `Thu Sep  3 15:25:17 UTC 2023 - PoC APP 1` `...`	
驗證工作負載 2 可以寫入檔案系統。	輸入以下命令以驗證工作負載 2 是否正在寫入`/data/out2.txt` 。 `kubectl -n $APP_NAMESPACE exec -ti poc-app2 -- tail -f /data/out2.txt` 結果類似於以下內容： `...` `Thu Sep  3 15:26:48 UTC 2023 - PoC APP 2` `Thu Sep  3 15:26:53 UTC 2023 - PoC APP 2` `Thu Sep  3 15:26:58 UTC 2023 - PoC APP 2` `...`	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
驗證工作負載 1 可以讀取工作負載 2 所寫入的檔案。	輸入以下命令，以驗證工作負載 1 是否可讀取工作負載 2 所寫入的/data/out2.txt 檔案。 `kubectl exec -ti poc-app1 -n poc-efs-eks-fargate -- tail -n 3 /data/out2.txt` 結果類似於以下內容： `...` `Thu Sep  3 15:26:48 UTC 2023 - PoC APP 2` `Thu Sep  3 15:26:53 UTC 2023 - PoC APP 2` `Thu Sep  3 15:26:58 UTC 2023 - PoC APP 2` `...`	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
驗證工作負載 2 可以讀取工作負載 1 所寫入的檔案。	輸入以下命令以驗證工作負載 2 是否可讀取工作負載 1 所寫入的/data/out1.txt 檔案。 `kubectl -n $APP_NAMESPACE exec -ti poc-app2 -- tail -n 3 /data/out1.txt` 結果類似於以下內容： `... Thu Sep 3 15:29:22 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:27 UTC 2023 - PoC APP 1 Thu Sep 3 15:29:32 UTC 2023 - PoC APP 1 ...`	具有授與權限的 Kubernetes 使用者

任務	描述	所需技能
驗證在移除應用程式元件後保留檔案。	接下來，您可以使用指令碼移除應用程式元件 (持續性磁碟區、持續性磁碟區宣告和網繭)，並驗證檔案/data/out1.txt 和檔案/data/out2.txt 是否保留在檔案系統中。使用以下命令來執行 validate-efs-content.sh 指令碼。 `./scripts/epic05/validate-efs-content.sh \` `-t "$EFS_CREATION_TOKEN"` 其中$EFS_CREATION_TOKEN 是檔案系統的唯一建立權杖。 結果類似於以下內容： `pod/poc-app-validation created` `Waiting for pod get Running state...` `Waiting for pod get Running state...` `Waiting for pod get Running state...` `Results from execution of 'find /data' on validation process pod:` `/data` `/data/out2.txt` `/data/out1.txt`	具有授與權限的 Kubernetes 使用者，系統管理員

監控作業

任務	描述	所需技能
監控應用程式記錄。	作為第二天操作的一部分，請將應用程式日誌運送到 Amazon CloudWatch 進行監控。	AWS 系統管理員，具有授與許可的 Kubernetes 使用者
使用容器洞察來監控 Amazon EKS 和 Kubernetes 容器。	作為第二天操作的一部分，請使用 Amazon 容器洞察來監控 Amazon EKS 和 Kubernetes 系統。 CloudWatch 此工具會從不同層級和維度的容器化應用程式收集、彙總和摘要指標。如需詳細資訊，請參閱相關資源一節。	AWS 系統管理員，具有授與許可的 Kubernetes 使用者
使用監控 Amazon EFS CloudWatch。	作為第二天操作的一部分，使用 Amazon 監控檔案系統 CloudWatch，該 Amazon 會從 Amazon EFS 收集原始資料並將其處理為可讀的近即時指標。如需詳細資訊，請參閱相關資源一節。	AWS 系統管理員

清除資源

任務	描述	所需技能
清理模式的所有創建的資源。	完成此模式後，請清理所有資源，以避免產生 AWS 費用。使用 PoC 應用程式後，執行`clean-up-resources.sh` 指令碼以移除所有資源。完成下列其中一個選項。	具有授與權限的 Kubernetes 使用者，系統管理員

任務	描述	所需技能
	使用 KMS 金鑰的靜態加密功能： `./scripts/epic06/clean-up-resources.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN" \ -k "$KMS_KEY_ALIAS"` 其中$CLUSTER_NAME 是 Amazon EKS 叢集的名稱，$EFS_CREATION_TOKEN 是檔案系統的建立權杖，$KMS_KEY_ALIAS 是 KMS 金鑰的別名。 沒有靜態加密： `./scripts/epic06/clean-up-resources.sh \ -c "$CLUSTER_NAME" \ -t "$EFS_CREATION_TOKEN"` 其中$CLUSTER_NAME 是 Amazon EKS 叢集的名稱，$EFS_CREATION_TOKEN 是檔案系統的建立權杖。	

相關資源

參考

- 適用於 Amazon EKS 的 AWS Fargate 現在支援 Amazon EFS (公告)
- 如何在 AWS Fargate 上使用 Amazon EKS 時擷取應用程式日誌 (部落格文章)
- 使用容器洞察 (Amazon CloudWatch 文件)
- 在 Amazon EKS 和 Kubernetes 上設定容器洞察 (Amazon 文件) CloudWatch
- Amazon EKS 和 Kubernetes 容器洞察指標 (Amazon 文件) CloudWatch
- 使用 Amazon 監控 Amazon EFS CloudWatch (Amazon EFS 文件)

GitHub 教學課程和範例

- 靜態佈建
- 傳輸中加密
- 從多個網繭存取檔案系統
- 在中使用 Amazon EFS StatefulSets
- 安裝次路徑
- 使用 Amazon EFS 存取點
- 地形的 Amazon EKS 藍圖

所需工具

- 安裝 AWS CLI 第 2 版
- 安裝外掛程式
- 安裝庫貝克特爾
- 正在安裝 jq

其他資訊

以下是`kubectl get pv`命令的範例輸出。

```
NAME         CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
            STORAGECLASS   REASON   AGE
poc-app-pv   1Mi        RWX            Retain           Bound    poc-efs-eks-fargate/
poc-app-pvc   efs-sc                  3m56s
```

以下是`kubectl -n poc-efs-eks-fargate get pvc`命令的範例輸出。

```
NAME          STATUS   VOLUME       CAPACITY   ACCESS MODES   STORAGECLASS   AGE
poc-app-pvc   Bound    poc-app-pv   1Mi        RWX            efs-sc         4m34s
```

更多模式

- 使用 CAST 醒目提示評估應用程式移轉至 AWS 雲端的準備程度
- 使用 AWS CDK 為微型服務自動建置 CI/CD 管道和 Amazon ECS 叢集
- 使用 GitHub 動作和地形表單建置碼頭映像並將其推送到 Amazon ECR
- 容器化已由 Blu Age 現代化的大型主機工作負載
- 使用 Firelens 日誌路由器為 Amazon ECS 創建自定義日誌解析器
- 在 Amazon ECS 上部署適用於 Java 微服務的 CI/CD 管道
- 使用 EC2 執行個體設定檔從 AWS Cloud9 部署 Amazon EKS 叢集
- 使用 Terraform 為容器化的藍光時代應用程式部署環境
- 使用 Amazon 中的推論管道將預處理邏輯部署到單一端點中的 ML 模型 SageMaker
- 使用 AWS 程式碼服務和 AWS KMS 多區域金鑰，管理對多個帳戶和區域的微型服務的藍/綠部署
- 使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式
- 從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk
- 在 Amazon ECS 上從甲骨文遷移 WebLogic 到阿帕奇湯姆貓（TomEE）
- 在 AWS 上將 ASP.NET 網頁表單應用程式現代化
- 使用 AWS CloudFormation 和 AWS Config 監控 Amazon ECR 儲存庫是否有萬用字元許可
- 使用 AWS CDK 和在 Amazon ECS Anywhere 為混合式工作負載設定 CI/CD 管道 GitLab
- 在 Amazon S3 中設置頭盔 v3 圖表存儲庫
- ???
- 使用憑證管理員和讓我們 end-to-end 加密為 Amazon EKS 上的應用程式設定加密
- 使用 Flux 簡化 Amazon EKS 多租戶應用程式部署
- 使用 AWS Lambda 在六角形架構中建構 Python 專案
- 在 Amazon 上訓練和部署支援 GPU 的自訂機器學習模型 SageMaker

內容交付

主題

- 使用 AWS Firewall Manager 和 Amazon 資料防 Firehose，將 AWS WAF 日誌傳送到 Splunk
- 使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront
- 更多模式

使用 AWS Firewall Manager 和 Amazon 資料防 Firehose，將 AWS WAF 日誌傳送到 Splunk

創建者：邁克爾·弗里登塔爾（AWS）、阿曼·考爾甘地（AWS）和 JJ 約翰遜（AWS）

環境：PoC 或試點	技術：內容傳遞；安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：AWS Firewall Manager；Amazon Kinesis Data Firehose；AWS WAF		

Summary

從歷史上看，有兩種方法可以將數據移動到 Splunk：推送或拉動架構。提取架構可透過重試提供傳送資料保證，但 Splunk 中需要專用資源來輪詢資料。拉架構通常不是實時的，因為輪詢。中的推送架構通常具有較低的延遲、更具擴展性，並且可降低操作複雜性和成本。但是，它不保證傳送，通常需要代理程式。

Splunk 與 Amazon 資料 Firehose 整合可透過 HTTP 事件收集器 (HEC) 將即時串流資料提供給 Splunk。這種整合同時提供推送和拉取架構的優勢 — 它可確保透過重試方式傳遞資料、接近即時性，並且具有低延遲和低複雜性。港燈可快速且有效率地透過 HTTP 或 HTTPS 將資料直接傳送到 Splunk。HECs 是基於令牌的，這消除了在應用程序或支持文件中對憑據進行硬編碼的需要。

在 AWS Firewall Manager 政策中，您可以為所有帳戶中的所有 AWS WAF Web ACL 流量設定記錄，然後使用 Firehose 交付串流將該日誌資料傳送到 Splunk 以進行監控、視覺化和分析。此解決方案提供下列優點：

- 針對所有帳戶中的 AWS WAF 網路 ACL 流量進行集中管理和記錄
- Splunk 與單一 AWS 帳戶整合
- 可擴展性
- 近乎即時的日誌資料傳送
- 透過使用無伺服器解決方案進行成本最佳化，因此您不必為未使用的資源付費。

先決條件和限制

先決條件

- 屬於 AWS 組織組織一部分的有效 AWS Organizations 帳戶。
- 您必須具備下列權限，才能使用 Firehose 啟用記錄功能：
  - `iam:CreateServiceLinkedRole`
  - `firehose:ListDeliveryStreams`
  - `wafv2:PutLoggingConfiguration`
- 必須設定 AWS WAF 及其網路 ACL。如需指示，請參閱開始使用 AWS WAF。
- 必須設定 AWS Firewall Manager。如需指示，請參閱 AWS Firewall Manager 先決條件
- 必須設定 AWS WAF 的 Firewall Manager 員安全政策。如需指示，請參閱 AWS Firewall Manager AWS WAF 政策入門。
- Splunk 必須使用可由 Firehose 連線到的公開 HTTP 端點進行設定。

限制

- AWS 帳戶必須在 AWS Organizations 的單一組織中進行管理。
- Web ACL 必須與交付串流位於相同的區域。如果您要擷取 Amazon 的日誌 CloudFront，請在美國東部 (維吉尼亞北部) 區域建立 Firehose 交付串流。`us-east-1`
- 適用於 Firehose 的 Splunk 附加元件可用於付費 Splunk 雲端部署、分散式 Splunk 企業部署和單一執行個體 Splunk 企業部署。免費試用 Splunk 雲端部署不支援此附加元件。

架構

目標技術堆疊

- Firewall Manager
- Firehose
- Amazon S3
- AWS WAF
- Splunk

目標架構

下圖顯示如何使用 Firewall Manager 員集中記錄所有 AWS WAF 資料，並透過 Kinesis Data Firehose 將其傳送至 Splunk。

1. AWS WAF 網路 ACL 會將防火牆記錄檔資料傳送至 Firewall Manager 員。
2. 「Firewall Manager 員」會將記錄檔資料傳送至 Firehose。
3. Firehose 交付串流會將日誌資料轉送到 Splunk 和 S3 儲存貯體。如果 Firehose 交付串流發生錯誤，S3 儲存貯體會充當備份。

自動化和規模

此解決方案旨在擴展和容納組織內的所有 AWS WAF Web ALC。您可以將所有網頁 ACL 設定為使用相同的 Firehose 執行個體。不過，如果您想要設定和使用多個 Firehose 執行個體，您可以。

工具

AWS 服務

- AWS Firewall Manager 是一項安全管理服務，可協助您集中設定和管理 AWS Organizations 中帳戶和應用程式的防火牆規則。
- Amazon 資料 Firehose 可協助您將即時串流資料交付到其他 AWS 服務、自訂 HTTP 端點，以及受支援的第三方服務供應商 (例如 Splunk) 擁有的 HTTP 端點。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS WAF 是一種 Web 應用程式防火牆，可協助您監控轉寄至受保護 Web 應用程式資源的 HTTP 和 HTTPS 請求。

其他工具

- Splunk 可協助您監視、視覺化和分析記錄資料。

史诗

設定溢出

任務	描述	所需技能
安裝適用於 AWS 的 Splunk 應用程式。	1. 登錄到您的 Splunk 重型貨運代理商。預設網址為`http://<IP address>:8000` 。 2. 在左側導覽列中，選擇 [應用程式] 旁邊的 [齒輪] 按鈕。 3. 選擇瀏覽更多應用程式。 4. 搜索 AWS。 5. 對於適用於 AWS 的 Splunk 應用程式，請選擇安裝。 6. 輸入您的 Splunk.com 登入認證，接受條款與條件，然後選擇 [登入並安裝]。 7. 選擇完成。	安全管理員，Splunk 管理員
安裝適用於 AWS WAF 的附加元件。	重複上述指示，安裝適用於 Splunk 的 AWS Web 應用程式防火牆附加元件。	安全管理員，Splunk 管理員
安裝並設定 Firehose 的 Splunk 附加元件。	1. 安裝並設定 Firehose 的 Splunk 附加元件。作為安裝和配置的一部分，如果您的 Splunk 平台有必要，您可以設置 HTTP 事件收集器並準備基礎結構以將日誌數據發送到索引器。請參閱與您的 Splunk 部署相對應的指示： • Splunk 雲端部署 (Splunk 文件)	安全管理員，Splunk 管理員

任務	描述	所需技能
	• 分散式 Splunk 企業部署 (Splunk 文件) • 單一執行個體 Splunk 企業部署 (Splunk 說明文件) 重要:安裝並設定 Splunk 附加元件之後，請停止此程序。請勿繼續進行設定 Firehose 以將資料傳送至 Splunk 平台的指示。 2. 記下 HTTP 事件收集器令牌和 HTTP 端點。稍後設定傳遞串流時，您需要此值。	

建立 Firehose 交付串流

任務	描述	所需技能
授予 Firehose 存取 Splunk 目的地的權限。	設定允許 Firehose 存取 Splunk 目的地的存取政策，並將日誌資料備份到 S3 儲存貯體。如需詳細資訊，請參閱授與 Firehose 存取 Splunk 目的地。	安全管理員
建立 Firehose 交付串流。	在您管理 AWS WAF 網路 ACL 的相同帳戶中，在 Firehose 中建立交付串流。您在建立交付串流時必須擁有 IAM 角色。Firehose 會假設該 IAM 角色，並取得指定 S3 儲存貯體的存取權。如需指示，請參閱建立交付串流。注意下列事項：	安全管理員

任務	描述	所需技能
	• 交付串流名稱必須以開頭aws-waf-logs- 。 • 對於來源，請選擇「直接放入」。 • 對於 S3 Backup 模式，請選擇 [備份所有事件]，然後選擇現有儲存貯體或建立新儲存貯體。 • 針對目的地，請依照 Firehose 文件中針對目的地選擇 Splunk 中的指示進行。如需 Splunk 端點和端點類型值的相關資訊，請參閱 Splunk 說明文件中的設定 Amazon 資料 Firehose。 針對您在 HTTP 事件收集器中設定的每個 Token 重複此程序。	
測試交付串流。	測試交付串流以驗證是否已正確設定。如需指示，請參閱 Firehose 文件中的使用 Splunk 做為目的地進行測試。	安全管理員

設定 Firewall Manager 員以記錄資料

任務	描述	所需技能
設定 Firewall Manager 員策略。	Firewall Manager 員政策必須設定為啟用記錄，並將記錄檔轉寄至正確的 Firehose 傳遞串流。如需詳細資訊和指示，請	安全管理員

任務	描述	所需技能
	參閱設定 AWS WAF 政策的記錄。	

相關資源

AWS 資源

- 記錄網路 ACL 流量 (AWS WAF 文件)
- 設定 AWS WAF 政策的記錄日誌記錄 (AWS WAF 文件)
- 教學課程：使用 Amazon 資料 Firehose 將 VPC 流程日誌傳送至 Splunk (Firehose 文件)
- 如何使用 Amazon 數據 Firehose 將 VPC 流日誌推送到 Splunk？ (AWS 知識中心)
- 使用 Amazon 資料防 Firehose (AWS 部落格文章) 將資料擷取至 Splunk 提供支援

潑濺文件

- Amazon 數據 Firehose 的 Splunk 附加組件

使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront

創作者天使·埃曼紐爾·埃爾南德斯塞布萊恩

環境：PoC 或試點	技術：內容傳遞；網路；安全性、身分識別、合規性；無伺服器；Web 和行動應用程式	AWS 服務：Amazon CloudFront；Elastic Load Balancing (ELB)；AWS Lambda

Summary

當您提供在 Amazon Web Services (AWS) 上託管的靜態內容時，建議的方法是使用 Amazon Simple Storage Service (S3) 儲存貯體做為來源，並使用 Amazon CloudFront 分發內容。此解決方案有兩個主要優點：在邊緣位置快取靜態內容的便利性，以及為 CloudFront 散佈定義 Web 存取控制清單 (Web ACL) 的能力，協助您以最少的組態和管理額外負荷保護對內容的要求。

但是，建議的標準方法有一個共同的架構限制。在某些環境中，您希望部署在虛擬私有雲 (VPC) 中的虛擬防火牆應用裝置檢查所有內容，包括靜態內容。標準方法不會透過 VPC 路由流量進行檢查。此模式提供了替代的架構解決方案。您仍然使用 CloudFront 分發來提供 S3 儲存貯體中的靜態內容，但是流量會使用 Application Load Balancer 透過 VPC 路由傳送。然後，AWS Lambda 函數會從 S3 儲存貯體擷取並傳回內容。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- S3 儲存貯體中託管的靜態網站內容。

限制

- 此模式中的資源必須位於單一 AWS 區域，但可以在不同的 AWS 帳戶中佈建。
- 限制會分別套用至 Lambda 函數可接收和傳送的最大請求和回應大小。如需詳細資訊，請參閱 Lambda 函數作為目標的限制 (Elastic Load Balancing 文件)。

- 使用這種方法時，請務必在效能、可擴充性、安全性和成本效益之間取得良好的平衡。儘管 Lambda 具有很高的可擴展性，但如果並行 Lambda 叫用數量超過最大配額，則會限制某些請求。如需詳細資訊，請參閱 Lambda 配額 (Lambda 文件)。您還需要在使用 Lambda 時考慮定價。若要將 Lambda 叫用最小化，請確定您已正確定義散發的快取。 CloudFront 如需詳細資訊，請參閱最佳化快取和可用性 (CloudFront 說明文件)。

架構

目標技術堆疊

- CloudFront
- Amazon Virtual Private Cloud (Amazon VPC)
- Application Load Balancer
- Lambda
- Amazon S3

目標架構

下圖顯示當您需要使用 CloudFront 透過 VPC 提供 S3 儲存貯體的靜態內容時所建議的架構。

1. 客戶端請求 CloudFront 分發 URL 以獲取 S3 存儲桶中的特定網站文件。
2. CloudFront 將請求傳送至 AWS WAF。AWS WAF 會使用套用至分發的網路 ACL 來篩選請求。CloudFront 如果請求被判定為有效，則流程會繼續進行。如果要求判定為無效，用戶端會收到 403 錯誤。
3. CloudFront 檢查其內部緩存。如果存在與傳入請求匹配的有效密鑰，則關聯的值將作為響應發送回客戶端。如果沒有，流量將繼續。
4. CloudFront 將要求轉送至指定 Application Load Balancer 的 URL。
5. 應用程式負載平衡器具有與以 Lambda 函數為基礎的目標群組相關聯的接聽程式。應用程式負載平衡器會叫用 Lambda 函數。
6. Lambda 函數會連線至 S3 儲存貯體，在其上執行`GetObject`作業，然後傳回內容做為回應。

自動化和規模

若要使用此方法自動化靜態內容的部署，請建立 CI/CD 管道以更新託管網站的 Amazon S3 儲存貯體。

Lambda 函數會在服務的配額和限制範圍內自動擴展以處理並行請求。如需詳細資訊，請參閱 Lambda 函數擴展和 Lambda 配額 (Lambda 文件)。對於其他 AWS 服務和功能 (例如 CloudFront 和 Application Load Balancer)，AWS 會自動擴展這些服務和功能。

工具

- Amazon CloudFront 透過全球資料中心網路提供您的 Web 內容，加快 Web 內容的分發速度，進而降低延遲並提升效能。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。在此模式中，您可以使用透過 Elastic Load Balancing 佈建的應用程式負載平衡器，將流量導向至 Lambda 函數。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

史诗

用 CloudFront 於透過 VPC 提供來自 Amazon S3 的靜態內容

任務	描述	所需技能
建立 VPC。	建立 VPC 以託管此模式中部署的資源，例如應用程式負載平衡器和 Lambda 函數。 如需指示，請參閱建立 VPC (Amazon VPC 文件)。	雲端架構師
建立一個 AWS WAF 網路 ACL。	建立一個 AWS WAF 網路 ACL。稍後在此樣式中，您可以將此 Web ACL 套用至 CloudFront 分佈。如需相關指示，請參閱建立網路 ACL (AWS WAF 文件)。	雲端架構師

任務	描述	所需技能
建立 Lambda 函數。	建立 Lambda 函數，將 S3 儲存貯體中託管的靜態內容做為網站提供服務。使用此模式的「其他資訊」區段中提供的程式碼。自訂程式碼以識別您的目標 S3 儲存貯體。	一般 AWS
上傳 Lambda 函數。	輸入下列命令，將 Lambda 函數程式碼上傳至 Lambda 中的 .zip 檔案封存。 `aws lambda update-function-code \ --function-name \ --zip-file fileb://lambda-alb-s3-website.zip`	一般 AWS
建立應用程式負載平衡器。	建立指向 Lambda 函數的網際網路對向 Application Load Balancer。如需指示，請參閱建立 Lambda 函數的目標群組 (Elastic Load Balancing 文件)。對於高可用性組態，請建立 Application Load Balancer，並將其附加到不同可用區域中的私有子網路。	雲端架構師

任務	描述	所需技能
創建一個 CloudFront 分佈。	建立指向您建立的 Application Load Balancer 的 CloudFront 發佈。 1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/cloudfront/v3/home 開啟 CloudFront 主控台。 2. 選擇 Create Distribution (建立分佈)。 3. 在 Create Distribution Wizard (建立分佈精靈) 的第一頁上，在 Web (Web) 區段中選擇 Get Started (開始使用)。 4. 指定發行版的設定。如需詳細資訊，請參閱在建立或更新分佈時您指定的值。注意下列事項： a. 將「應用程式負載平衡器」設定為原點。 b. 在分發設定中，選擇您要透過 AWS WAF 套用的現有 Web ACL。如需詳細資訊，請參閱 AWS WAF 網路 ACL。 5. 儲存您的變更。 6. CloudFront 建立發行版之後，發佈的 [狀態] 欄的值會從變更InProgress為 [已部署]。如果您選擇啟用分佈，將會在狀態切換為 Deployed	雲端架構師

任務	描述	所需技能
	(已部署) 之後準備好處理請求。	

相關資源

AWS 文件

- 優化緩存和可用性（CloudFront 文檔）
- 作為目標的 Lambda 函數 (Elastic Load Balancing 文件)
- 配額 (Lambda 文件)

AWS 服務網站

- Application Load Balancer
- Lambda
- CloudFront
- Amazon Simple Storage Service (Amazon S3)
- AWS WAF
- Amazon VPC

其他資訊

Code

下面的示例 Lambda 函數是用 Node.js 編寫的。此 Lambda 函數充當 Web 伺服器，可對包含網站資源的 S3 儲存貯體執行`GetObject`操作。

```
/**

 * This is an AWS Lambda function created for demonstration purposes.

 * It retrieves static assets from a defined Amazon S3 bucket.

 * To make the content available through a URL, use an Application Load Balancer with a
Lambda integration.
```

```
 *
 * Set the S3_BUCKET environment variable in the Lambda function definition.
 */

var AWS = require('aws-sdk');

exports.handler = function(event, context, callback) {

    var bucket = process.env.S3_BUCKET;
    var key = event.path.replace('/', '');

    if (key == '') {
        key = 'index.html';
    }

    // Fetch from S3
    var s3 = new AWS.S3();
    return s3.getObject({Bucket: bucket, Key: key},
       function(err, data) {

            if (err) {
                return err;
            }

            var isBase64Encoded = false;
            var encoding = 'utf8';

            if (data.ContentType.indexOf('image/') > -1) {
                isBase64Encoded = true;
                encoding = 'base64'
            }

            var resp = {
                statusCode: 200,
                headers: {
                    'Content-Type': data.ContentType,
                },
                body: new Buffer(data.Body).toString(encoding),
                isBase64Encoded: isBase64Encoded
            };

            callback(null, resp);
        }
    );
```

```
};
```

更多模式

- 檢查 Amazon CloudFront 分佈的存取記錄、HTTPS 和 TLS 版本
- 在 Amazon EKS 叢集上部署以 gRPC 為基礎的應用程式，並使 Application Load Balancer 存取
- ???
- 使用地形表單部署 AWS WAF 解決方案的安全自動化
- 使用 Splunk 檢視 AWS Network Firewall 日誌和指標

成本管理

主題

- 使用 AWS Cost Explorer 為 AWS Glue 任務建立詳細的成本和用量報告
- 使用 AWS Cost Explorer 為 Amazon EMR 叢集建立詳細的成本和用量報告
- 更多模式

使用 AWS Cost Explorer 為 AWS Glue 任務建立詳細的成本和用量報告

由帕里加比德（AWS）和芳香拉吉傑亞拉揚（AWS）創建

環境：生產	技術：成本管理；分析	AWS 服務：AWS Billing and Cost Management；AWS AWS Glue

Summary

此模式說明如何透過設定使用者定義的成本分配標籤來追蹤 AWS Glue 資料整合任務的使用成本。您可以使用這些標籤，在 AWS Cost Explorer 中為跨多個維度的任務建立詳細的成本和用量報告。例如，您可以追蹤小組、專案或成本中心層級的使用成本。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一或多個已啟用使用者定義標籤的 AWS Glue 任務

架構

目標技術堆疊

- AWS Glue
- AWS Cost Explorer

下圖顯示如何套用標籤來追蹤 AWS Glue 任務的使用成本。

該圖顯示以下工作流程：

1. 資料工程師或 AWS 管理員會為 AWS Glue 任務建立使用者定義的成本分配標籤。
2. AWS 管理員會啟用標籤。
3. 標籤會向 AWS Cost Explorer 報告中繼資料。

工具

- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- AWS Cost Explorer 可協助您檢視和分析 AWS 成本和用量。

史诗

為 AWS Glue 任務建立和啟用標籤

任務	描述	所需技能
為 AWS Glue 任務建立使用者定義的成本分配標籤。	若要將標籤新增至現有的 AWS Glue 任務 1. 登入 AWS 管理主控台，然後開啟 AWS Glue 主控台。 2. 在左側導覽窗格的 [ETL] 下，選擇 [工作]。 3. 在「您的工作」區段中，選擇要標記的工作名稱。 4. 選擇 Job details (任務詳細資訊) 索引標籤。然後，展開「高級屬性」部分。 5. 在「標籤」中，選擇「新增標籤」。 6. 在 Key 中，輸入標籤的名稱。 7. (選擇性) 在值中，輸入您要與金鑰關聯的值。	數據工程師

任務	描述	所需技能
	8. (選擇性) 針對您要為工作建立的每個標籤重複步驟 5-7。 9. 選擇儲存。 若要將標籤新增至新的 AWS Glue 任務 1. 根據您的使用案例需求建立新的 AWS Glue 任務。如需指示，請參閱 AWS Glue 開發人員指南中的使用 AWS Glue 主控台上的任務。 2. 設定 Job 務詳細資料設定時，請遵循此任務的若要新增標籤至現有 AWS Glue 任務部分的步驟 4-9。 注意：如需詳細資訊，請參閱 AWS Glue 開發人員指南中的 AWS Glue 標籤。	
啟動使用者定義的成本配置標籤。	遵循 AWS 帳單使用者指南中的啟用使用者定義的成本分配標籤中的指示。	AWS 管理員

為您的 AWS Glue 任務建立成本和用量報告

任務	描述	所需技能
在 AWS Cost Explorer 中使用標籤篩選器，為 AWS Glue 任務建立成本和用量報告。	1. 登入 AWS 管理主控台並開啟 AWS 成本管理主控台。 2. 在左側導覽窗格中，請選擇報告。	一般 AWS、AWS 管理員

任務	描述	所需技能
	3. 選擇 [建立新報告]。 4. 針對 [選取報表類型]，選擇 [成本和使用量 (建議使用)]。然後，選擇「創建報告」。 5. 對於篩選器，請選擇服務。出現「服務」下拉列表。 6. 選取 [Glue] 旁邊的核取方塊。然後，選擇「應用過濾器」。 7. 針對「篩選」，選擇「標籤」 標籤下拉清單隨即出現。 8. 選擇 [團隊]。然後，選取您已指派標籤的團隊旁邊的核取方塊。排除任何您尚未指派標籤的團隊。然後，選擇「應用過濾器」。 9. 在圖表頂端，選擇「標籤」。然後，為您要建立報告的 AWS Glue 任務選擇標籤。 10.在圖表頂端，選擇「過去 3 個月」下拉式清單，然後選擇您希望報表涵蓋的時間範圍。然後，選擇「每月」下拉式清單，並根據時間範圍選擇要如何彙總報告中的行項目。 11選擇 Save as (另存為)。然後，輸入報告的標題。 12選擇「儲存報告」。	

任務	描述	所需技能
	如需詳細資訊，請參閱 AWS 成本管理使用者指南中的使用 Cost Explorer 探索資料。	

使用 AWS Cost Explorer 為 Amazon EMR 叢集建立詳細的成本和用量報告

由帕里加比德（AWS）和芳香拉吉傑亞拉揚（AWS）創建

環境：生產	技術：成本管理；分析；大數據	AWS 服務：AWS Billing and Cost Management；Amazon EMR

Summary

此模式顯示如何透過設定使用者定義的成本分配標籤來追蹤 Amazon EMR 叢集的使用成本。您可以使用這些標籤，在 AWS Cost Explorer 中為跨多個維度的叢集建立詳細的成本和用量報告。例如，您可以追蹤小組、專案或成本中心層級的使用成本。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 已啟動使用者定義標籤的一或多個 EMR 叢集

架構

目標技術堆疊

- Amazon EMR
- AWS Cost Explorer

目標架構

下圖顯示如何套用標籤來追蹤特定 Amazon EMR 叢集的使用成本。

該圖顯示以下工作流程：

1. 資料工程師或 AWS 管理員會為 Amazon EMR 叢集建立使用者定義的成本分配標籤。
2. AWS 管理員會啟用標籤。
3. 標籤會向 AWS Cost Explorer 報告中繼資料。

工具

工具

- Amazon EMR 是一種受管叢集平台，可簡化在 AWS 上執行大數據架構以處理和分析大量資料的過程。
- AWS Cost Explorer 可協助您檢視和分析 AWS 成本和用量。

史诗

為您的 Amazon EMR 叢集建立和啟用標籤

任務	描述	所需技能
為您的 Amazon EMR 叢集建立使用者定義的成本分配標籤。	若要將標籤新增至現有的 Amazon EMR 叢集 遵循《Amazon EMR 管理指南》中將標籤新增至現有叢集中的指示進行。 將標籤新增至新的 Amazon EMR 叢集 依照 Amazon EMR 管理指南中的將標籤新增至新叢集中的指示進行。 如需如何設定 Amazon EMR 叢集的詳細資訊，請參閱《Amazon EMR 管理指南》中的規劃和設定叢集。	數據工程師

任務	描述	所需技能
啟動使用者定義的成本配置標籤。	遵循 AWS 帳單使用者指南中啟用使用者定義的成本分配標籤中的指示。	AWS 管理員

為您的 Amazon EMR 叢集建立成本和用量報告

任務	描述	所需技能
在 AWS Cost Explorer 中使用標籤篩選器，為 Amazon EMR 叢集建立成本和用量報告。	1. 登入 AWS 管理主控台並開啟 AWS 成本管理主控台。 2. 在左側導覽窗格中，請選擇報告。 3. 選擇 [建立新報告]。 4. 針對 [選取報表類型]，選擇 [成本和使用量 (建議使用)]。然後，選擇「創建報告」。 5. 對於篩選器，請選擇服務。出現「服務」下拉列表。 6. 選取 EMR (彈性 MapReduce) 和 EC2 執行個體 (彈性計算雲端 — 運算) 旁邊的核取方塊。然後，選擇「應用過濾器」。 7. 針對「篩選」，選擇「標籤」 標籤下拉清單隨即出現。 8. 選擇 [團隊]。然後，選取您已指派標籤的團隊旁邊的核取方塊。排除任何您尚未指派標籤的團隊。然後，選擇「應用過濾器」。	一般 AWS、AWS 管理員

任務	描述	所需技能
	9. 在圖表頂端，選擇「標籤」。然後，選擇您要為其建立報告的 Amazon EMR 叢集標籤。 10. 在圖表頂端，選擇「過去 3 個月」下拉式清單，然後選擇您希望報表涵蓋的時間範圍。然後，選擇「每月」下拉式清單，並根據時間範圍選擇要如何彙總報告中的行項目。 11. 選擇 Save as (另存為)。然後，輸入報告的標題。 12. 選擇「儲存報告」。 如需詳細資訊，請參閱 AWS 成本管理使用者指南中的使用 Cost Explorer 探索資料。	

更多模式

- 使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation
- 使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3
- ???
- 為 Amazon RDS 和 Amazon Aurora 創建詳細的成本和用量報告
- 使用 AWS Config 和 AWS Systems Manager 刪除未使用的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區
- 估算 Amazon DynamoDB 表格的儲存成本
- 估算隨需容量的 DynamoDB 表格的成本

資料湖

主題

- 自動從 AWS Data Exchange 擷取到 Amazon S3 的資料
- 使用 AWS DataOps 開發套件建立資料管道以擷取、轉換和分析 Google 分析資料
- 使用 Amazon Athena 設定對共用 AWS Glue 資料目錄的跨帳戶存取
- 跨帳戶資料共用自動化
- 使用基礎設施即程式碼在 AWS 雲端部署和管理無伺服器資料湖
- 使用 AWS IoT 資料以符合成本效益的方式，將物聯網資料直接導入 Amazon S3
- 使用萬 LiveData 迪斯科遷移器將 Hadoop 資料遷移到 Amazon S3
- 更多模式

自動從 AWS Data Exchange 擷取到 Amazon S3 的資料

由阿德南海藻（AWS）和曼尼康塔·戈納（AWS）創建

技術：分析；資料湖	環境：生產	AWS 服務：Amazon S3; Amazon CloudWatch; AWS Lambda; Amazon SNS

Summary

此模式提供 AWS CloudFormation 範本，可讓您將 AWS Data Exchange 中的資料自動導入 Amazon Simple Storage Service (Amazon S3) 中的資料湖。

AWS Data Exchange 是一項服務，可讓您輕鬆在 AWS 雲端中安全地交換以檔案為基礎的資料集。AWS Data Exchange 資料集是以訂閱為基礎。身為訂閱者，您也可以在提供者發佈新資料時存取資料集修訂。

AWS CloudFormation 範本會建立 Amazon CloudWatch 活動事件和 AWS Lambda 函數。此事件會監視您已訂閱之資料集的任何更新。如果有更新，請 CloudWatch 啟動 Lambda 函數，將資料複製到您指定的 S3 儲存貯體。成功複製資料後，Lambda 會傳送 Amazon Simple Notification Service (Amazon SNS) 通知給您。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 訂閱 AWS Data Exchange 中的資料集

限制

- AWS CloudFormation 範本必須針對 AWS Data Exchange 中的每個訂閱資料集個別部署。

架構

目標技術堆疊

- AWS Lambda
- Amazon S3
- AWS Data Exchange
- Amazon CloudWatch
- Amazon SNS

目標架構

自動化和規模

您可以針對要擷取到資料湖中的資料集多次使用 AWS CloudFormation 範本。

工具

- AWS Data Exchange — 這項服務可讓 AWS 客戶輕鬆在 AWS 雲端安全地交換檔案型資料集。作為訂閱者，您可以從合格的數據提供商那裡找到並訂閱數百種產品。然後，您可以快速下載資料集或將其複製到 Amazon S3，以便在各種 AWS 分析和機器學習服務中使用。擁有 AWS 帳戶的任何人都可以是 AWS Data Exchange 訂閱者。
- AWS Lambda — 一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。您只需為使用的運算時間付費；程式碼未執行時不會收取任何費用。使用 AWS Lambda，您可以針對幾乎任何類型的應用程式或後端服務執行程式碼，而無需管理。AWS Lambda 在高可用性運算基礎設施上執行程式碼，並管理所有運算資源，包括伺服器和作業系統維護、容量佈建和自動擴展、程式碼監控和記錄。
- Amazon S3 — 互聯網存儲。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Amazon CloudWatch 活動 — 提供近乎即時的系統事件串流，描述 AWS 資源的變更。使用可快速設置的簡單規則，您可以匹配事件並將其路由到一個或多個目標函數或流。 CloudWatch 事件會在發生時意識到操作變化。它會回應這些作業變更，並在必要時採取修正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。您也可以使用 CloudWatch 事件來排程使用 cron 或速率運算式在特定時間自行啟動的自動化動作。
- Amazon SNS — 一種 Web 服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。Amazon SNS 針對高輸送量、以推送為基礎的簡訊提供主題 (通訊管道)。 many-to-many 使用 Amazon SNS 主題，發佈者可以將訊息分發給大量訂閱者以進行 parallel 處理，包括 Amazon

Simple Queue Service (Amazon SQS) 佇列、AWS Lambda 函數和 HTTP/S 網路掛鉤。您也可以使用 Amazon SNS 透過行動推送、簡訊和電子郵件傳送通知給最終使用者。

史诗

訂閱資料集

任務	描述	所需技能
訂閱資料集。	在 AWS Data Exchange 主控台中，訂閱資料集。如需相關指示，請參閱「相關資源」一節中的連結。	一般 AWS
請注意資料集屬性。	請記下資料集的 AWS 區域、ID 和修訂 ID。在下一個步驟中，您將需要此功能用於 AWS CloudFormation 範本。	一般 AWS

部署 AWS CloudFormation 範本

任務	描述	所需技能
建立 S3 儲存貯體和資料夾。	如果 Amazon S3 中已有資料湖，請建立資料夾來存放要從 AWS Data Exchange 擷取的資料。如果您要部署範本以進行測試，請建立新的 S3 儲存貯體，並記下下一個步驟的儲存貯體名稱和資料夾前置詞。	一般 AWS
部署 AWS CloudFormation 範本。	將以附件形式提供的 AWS CloudFormation 範本部署到此模式。設定下列參數以對應您的 AWS 帳戶、資料集和 S3 儲存貯體設定：資料集	一般 AWS

任務	描述	所需技能
	AWS 區域、資料集 ID、修訂 ID、S3 儲存貯體名稱 (例如，文件 EXAMPLE-BUCKET)、資料夾前綴 (例如，我的資料夾/) 以及 SNS 通知的電子郵件。您可以將「資料集名稱」參數設定為任何名稱。當您部署範本時，它會執行 Lambda 函數，以自動擷取資料集中的第一組可用資料。隨後，隨後的擷取會自動進行，因為新的資料到達資料集。	

相關資源

- 訂閱 AWS Data Exchange 中的資料產品 (AWS Data Exchange 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS DataOps 開發套件建立資料管道以擷取、轉換和分析 Google 分析資料

由安東·庫什金（AWS）和魯迪·普格（AWS）創建

程式碼儲存庫：AWS DDK 範例 — 使用 Amazon AppFlow、亞馬 Amazon Athena 和 AWS DataOps 開發套件分析谷歌分析資料	環境：PoC 或試點	技術：資料湖、分析 DevOps、基礎架構
工作負載：開源	AWS 服務：Amazon AppFlow；Amazon Athena；AWS CDK；AWS Lambda；Amazon S3	

Summary

此模式說明如何使用 AWS DataOps 開發套件 (DDK) 和其他 AWS 服務建立資料管道以擷取、轉換和分析 Google 分析資料。AWS DDK 是開放原始碼開發架構，可協助您在 AWS 上建立資料工作流程和現代資料架構。AWS DDK 的其中一個主要目標是為您節省通常用於勞動密集型資料管道任務的時間和精力，例如協調管道、建立基礎設施，以及建立該基礎設施的背後 DevOps 工作。您可以將這些勞動密集型任務卸載到 AWS DDK，以便專注於撰寫程式碼和其他高價值活動。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 谷歌分析的 Amazon AppFlow 連接器，配置
- 蟒蛇和點子（Python 的軟件包管理器）
- Git，已安裝和配置
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝 AWS Cloud Development Kit (AWS CDK)

產品版本

- Python 3.7 或更高版本
- 點子 9.0.3 或更高版本

架構

技術, 堆

- Amazon AppFlow
- Amazon Athena
- Amazon CloudWatch
- Amazon EventBridge
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queue Service (Amazon SQS)
- AWS DataOps 開發套件 (DDK)
- AWS Lambda

目標架構

下圖顯示了導入，轉換和分析 Google Analytics（分析）數據的事件驅動過程。

該圖顯示以下工作流程：

1. Amazon CloudWatch 計劃事件規則調用 Amazon AppFlow。
2. Amazon 將谷 AppFlow 歌分析數據導入 S3 存儲桶。
3. S3 儲存貯體擷取資料後，會產生中 EventBridge 的事件通知，並由 CloudWatch 事件規則擷取，然後放入 Amazon SQS 佇列中。
4. Lambda 函數會使用 Amazon SQS 佇列中的事件、讀取個別 S3 物件、將物件轉換為 Apache Parquet 格式、將轉換後的物件寫入 S3 儲存貯體，然後建立或更新 AWS Glue 資料型錄資料目錄資料表定義。
5. Athena 查詢會針對資料表執行。

工具

AWS 工具

- Amazon AppFlow 是全受管的整合服務，可讓您在軟體即服務 (SaaS) 應用程式之間安全地交換資料。
- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Cloud Development Kit (CDK) 是在程式碼中定義雲端基礎設施並透過 AWS CloudFormation 佈建雲端基礎設施的架構。
- AWS 開 DataOps 發套件 (DDK) 是開放原始碼開發架構，可協助您在 AWS 上建立資料工作流程和現代資料架構。

Code

此模式的程式碼可在 GitHub AWS DataOps 開發套件 (DDK) 中取得，以及使用 Amazon AppFlow、亞馬 Amazon Athena 和 AWS DataOps 開發套件儲存庫分析谷歌分析資料。

史诗

準備環境

任務	描述	所需技能
克隆源代碼。	若要克隆源代碼，請運行以下命令：	DevOps 工程師

任務	描述	所需技能
	`git clone https://github.com/aws-samples/aws-ddk-examples.git`	
建立虛擬環境。	瀏覽至原始程式碼目錄，然後執行下列命令以建立虛擬環境： `cd google-analytics-data-using-appflow/python && python3 -m venv .venv`	DevOps 工程師
安裝依賴關係。	若要啟動虛擬環境並安裝相依性，請執行下列命令： `source .venv/bin/activate && pip install -r requirements.txt`	DevOps 工程師

部署使用資料管線的應用程式

任務	描述	所需技能
引導環境。	1. 確認 AWS CLI 已為您的 AWS 帳戶設定有效登入資料。如需詳細資訊，請參閱 AWS CLI 文件中的使用命名設定檔。 2. 執行 `cdk bootstrap --profile [AWS_PROFILE]` 命令。	DevOps 工程師

任務	描述	所需技能
部署資料。	若要部署資料管線，請執行`cdk deploy --profile [AWS_PROFILE]` 命令。	DevOps 工程師

測試部署

任務	描述	所需技能
驗證堆疊狀態。	1. 開啟 AWS CloudFormation 主控台。 2. 在「堆疊」頁面上，確認堆疊的狀態`DdkAppflowAthenaStack` 為`CREATE_COMPLETE` 。	DevOps 工程師

故障診斷

問題	解決方案
建立`AWS::AppFlow::Flow` 資源期間部署失敗，並且您收到下列錯誤： `Connector Profile with name ga-connection does not exist`	確認您為谷歌分析創建了一個 Amazon AppFlow 連接器並命名它`ga-connection` 。 有關說明，請參閱 Amazon AppFlow 文檔中的谷歌分析。

相關資源

- AWS DataOps 開發套件 (DDK) GitHub
- AWS DDK 範例 () GitHub

其他資訊

AWS DDK 資料管道由一個或多個階段組成。在下列程式碼範例中，您可AppFlowIngestionStage以使用從 Google Analytics (分析) 擷取資料、SqsToLambdaStage處理資料轉換，AthenaSQLStage以及執行 Athena 查詢。

首先，會建立資料轉換和擷取階段，如下列程式碼範例所示：

```
        appflow_stage = AppFlowIngestionStage(
            self,
            id="appflow-stage",
            flow_name=flow.flow_name,
        )
        sqs_lambda_stage = SqsToLambdaStage(
            self,
            id="lambda-stage",
            lambda_function_props={
                "code": Code.from_asset("./ddk_app/lambda_handlers"),
                "handler": "handler.lambda_handler",
                "layers": [
                    LayerVersion.from_layer_version_arn(
                        self,
                        id="layer",
                        layer_version_arn=f"arn:aws:lambda:
{self.region}:336392948345:layer:AWSDataWrangler-Python39:1",
                    )
                ],
                "runtime": Runtime.PYTHON_3_9,
            },
        )
        # Grant lambda function S3 read & write permissions
        bucket.grant_read_write(sqs_lambda_stage.function)
        # Grant Glue database & table permissions
        sqs_lambda_stage.function.add_to_role_policy(
            self._get_glue_db_iam_policy(database_name=database.database_name)
        )
        athena_stage = AthenaSQLStage(
            self,
            id="athena-sql",
            query_string=[
                (
                    "SELECT year, month, day, device, count(user_count) as cnt "
                    f"FROM {database.database_name}.ga_sample "
```

```
                    "GROUP BY year, month, day, device "
                    "ORDER BY cnt DESC "
                    "LIMIT 10; "
                )
            ],
            output_location=Location(
                bucket_name=bucket.bucket_name, object_key="query-results/"
            ),
            additional_role_policy_statements=[
                self._get_glue_db_iam_policy(database_name=database.database_name)
            ],
        )
```

接下來，DataPipeline建構會使用 EventBridge 規則將階段連接在一起，如下列程式碼範例所示：

```
        (
            DataPipeline(self, id="ingestion-pipeline")
            .add_stage(
                stage=appflow_stage,
                override_rule=Rule(
                    self,
                    "schedule-rule",
                    schedule=Schedule.rate(Duration.hours(1)),
                    targets=appflow_stage.targets,
                ),
            )
            .add_stage(
                stage=sqs_lambda_stage,
                # By default, AppFlowIngestionStage stage emits an event after the flow
run finishes successfully
                # Override rule below changes that behavior to call the the stage when
data lands in the bucket instead
                override_rule=Rule(
                    self,
                    "s3-object-created-rule",
                    event_pattern=EventPattern(
                        source=["aws.s3"],
                        detail={
                            "bucket": {"name": [bucket.bucket_name]},
                            "object": {"key": [{"prefix": "ga-data"}]},
                        },
                        detail_type=["Object Created"],
                    ),
```

```
                    targets=sqs_lambda_stage.targets,
                ),
            )
            .add_stage(stage=athena_stage)
        )
```

如需更多程式碼範例，請參閱使用 Amazon AppFlow、亞馬 Amazon Athena 和 AWS DataOps 開發套件儲存庫 GitHub 分析谷歌分析資料。

使用 Amazon Athena 設定對共用 AWS Glue 資料目錄的跨帳戶存取

創建者丹尼斯·阿夫多寧 (AWS)

環境：生產	技術：資料湖、分析、大數據	工作負載：所有其他工作
AWS 服務：Amazon Athena；AWS AWS Glue		

Summary

此模式提供 step-by-step 指示 (包括 AWS Identity and Access Management (IAM) 政策範例，以使用 AWS Glue 資料型錄設定存放在 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體中的資料集的跨帳戶共用。您可以將資料集存放在 S3 儲存貯體中。中繼資料是由 AWS Glue 爬行程式收集並放入 AWS Glue 資料型錄中。S3 儲存貯體和 AWS Glue 資料型錄存放在稱為資料帳戶的 AWS 帳戶中。您可以在另一個稱為消費者帳戶的 AWS 帳戶中提供 IAM 主體的存取權。使用者可以使用 Amazon Athena 無伺服器查詢引擎，查詢消費者帳戶中的資料。

先決條件和限制

先決條件

- 兩個作用中的 AWS 帳戶
- 其中一個 AWS 帳戶中的 S3 儲存貯體
- Athena 引擎版本 2
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI) (或 CloudShell用於執行 AWS CLI 命令的 AWS)

產品版本

此模式僅適用於 Athena 引擎第 2 版和 Athena 引擎版本 3。我們建議您升級至 Athena 引擎版本 3。如果您無法從 Athena 引擎版本 1 升級為 Athena 引擎版本 3，請依照 AWS 大數據部落格中的 Amazon Athena 跨帳戶 AWS Glue 資料型錄存取方法進行操作。

架構

目標技術堆疊

- Amazon Athena
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)

下圖顯示的架構使用 IAM 許可，透過 AWS AWS Glue 資料型錄與另一個 AWS 帳戶 (消費者帳戶) 共用一個 AWS 帳戶 (資料帳戶) 中 S3 儲存貯體中的資料。

該圖顯示以下工作流程：

1. 資料帳戶中的 S3 儲存貯體政策授予對消費者帳戶中的 IAM 角色以及資料帳戶中 AWS Glue 編目程式服務角色的許可。
2. 資料帳戶中的 AWS KMS 金鑰政策會授予消費者帳戶中的 IAM 角色以及資料帳戶中 AWS Glue 爬蟲服務角色的許可。
3. 資料帳戶中的 AWS Glue 爬蟲程式會探索 S3 儲存貯體中存放之資料的結構描述。
4. 資料帳戶中 AWS Glue 資料型錄的資源政策可授予取用者帳戶中 IAM 角色的存取權。
5. 使用者使用 AWS CLI 命令在消費者帳戶中建立具名目錄參考。
6. IAM 政策授予消費者帳戶中的 IAM 角色，以存取資料帳戶中的資源。IAM 角色的信任政策允許消費者帳戶中的使用者擔任 IAM 角色。
7. 消費者帳戶中的使用者擔任 IAM 角色，並使用 SQL 查詢存取資料目錄中的物件。
8. Athena 無伺服器引擎會執行 SQL 查詢。

附註：IAM 最佳實務建議您將權限授與 IAM 角色，並使用聯合身分。

工具

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以保護資料。

史诗

在資料帳戶中設定權限

任務	描述	所需技能
授與 S3 儲存貯體中資料的存取權。	根據以下範本建立 S3 儲存貯體政策，並將政策指派給存放資料的儲存貯體。 `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": ["arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler"] }, "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*"`	雲端管理員

任務	描述	所需技能
	`        },` `        {` `            "Effect":` ` "Allow",` `            "Principa` `l": {` `                "AWS": [` `  "arn:aws:iam::<con` `sumer account id>:role/` `<role name>",` `  "arn:aws:iam::<dat` `a account id>:role/` `service-role/AWSGl` `ueServiceRole-data-` `bucket-crawler"` `                ]` `            },` `            "Action":` ` "s3:ListBucket",` `            "Resource":` ` "arn:aws:s3:::data-` `bucket"` `        }` `    ]` `}` 儲存貯體政策將許可授予消費者帳戶中的 IAM 角色，以及資料帳戶中的 AWS Glue 爬蟲服務角色。	

任務	描述	所需技能
(如果需要) 授與資料加密金鑰的存取權。	如果 S3 儲存貯體使用 AWS KMS 金鑰加密，請授與該金鑰的`kms:Decrypt` 權限給取用者帳戶中的 IAM 角色，以及授與資料帳戶中的 AWS Glue 爬蟲服務角色。 使用下列陳述式更新金鑰原則： `{ "Effect": "Allow", "Principal": { "AWS": [ "arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler" ] }, "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }`	雲端管理員

任務	描述	所需技能
授與爬行者程式對資料的存取權。	將下列 IAM 政策附加至爬行者程式的服務角色： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data-bucket" } ] }`	雲端管理員

任務	描述	所需技能
(必要時) 授與爬行者程式 (Crawler) 資料加密金鑰的存取權。	如果 S3 儲存貯體使用 AWS KMS 金鑰加密，請將下列政策附加至爬行者程式的服務角色，將該金鑰的`kms:Decrypt` 權限授予該金鑰： `{` `  "Effect": "Allow",` `  "Action": "kms:Decrypt",` `  "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>"` `}`	雲端管理員

任務	描述	所需技能
授與取用者帳戶中的 IAM 角色，並授與爬行者程式存取資料目錄。	1. 登入 AWS 管理主控台並開啟 AWS Glue 主控台。 2. 在導覽窗格的 [資料目錄] 下，選擇 [設定]。 3. 在 [權限] 區段中，新增下列陳述式，然後選擇 [儲存]。 `{ "Version" : "2012-10-17", "Statement" : [{ "Effect" : "Allow", "Principal" : { "AWS" : ["arn:aws:iam::<consumer account id>:role/<role name>", "arn:aws:iam::<data account id>:role/service-role/AWSGlueServiceRole-data-bucket-crawler"] }, "Action" : "glue:*", "Resource" : ["arn:aws:glue:<region>:<data account id>:catalog",`	雲端管理員

任務	描述	所需技能
	<pre>"arn:aws:glue:<region>:<data account id>:database/*", "arn:aws:glue:<region>:<data account id>:table/*"] }] }</pre> 此政策允許資料帳戶中所有資料庫和表格上的所有 AWS Glue 動作。您可以自訂原則，只將必要的權限授與用戶主體。例如，您可以提供資料庫中特定資料表或檢視表的唯讀存取權。	

從消費者帳戶訪問數據

任務	描述	所需技能
為資料目錄建立具名參考。	若要建立具名資料型錄參考，請使用CloudShell或在本機安裝的 AWS CLI 執行下列命令： <pre>aws athena create-data-catalog --name <shared catalog name> --type GLUE --parameters catalog-id=<data account id></pre>	雲端管理員

任務	描述	所需技能
授與消費者帳戶中的 IAM 角色對資料的存取權。	將以下政策附加到消費者帳戶中的 IAM 角色，以授予角色跨帳戶對資料的存取權限： { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::data-bucket/*" }, { "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "arn:aws:s3:::data-bucket" }, { "Effect": "Allow", "Action": "glue:*", "Resource": ["arn:aws:glue:<region>:<data account id>:catalog", "arn:aws:glue:<region>:<data account id>:database/*",	雲端管理員

任務	描述	所需技能
	<pre>"arn:aws:glue:<region>:<data account id>:table/*"] }] }</pre> 接下來，使用下列範本指定哪些使用者可以在其信任政策中接受 IAM 角色： <pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<consumer account id>:user/<IAM user>" }, "Action": "sts:AssumeRole" }] }</pre> 最後，透過將相同的政策附加到他們所屬的使用者群組，授予使用者假設 IAM 角色的權限。	

任務	描述	所需技能
（如果需要）授予消費者帳戶中的 IAM 角色對資料加密金鑰的存取權。	如果 S3 儲存貯體使用 AWS KMS 金鑰加密，請將下列政策附加至該儲存貯體，將該金鑰的`kms:Decrypt` 權限授與消費者帳戶中的 IAM 角色： `{ "Effect": "Allow", "Action": "kms:Decrypt", "Resource": "arn:aws:kms:<region>:<data account id>:key/<key id>" }`	雲端管理員
切換到消費者帳戶中的 IAM 角色以存取資料。	身為資料使用者，請切換至 IAM 角色以存取資料帳戶中的資料。	資料消費者

任務	描述	所需技能
存取資料。	使用 Athena 查詢資料。例如，開啟 Athena 查詢編輯器並執行下列查詢： `SELECT * FROM <shared catalog name>.<database name>.<table name>` 除了使用具名目錄參考外，您也可以依 Amazon 資源名稱 (ARN) 參考目錄。 附註：如果您在查詢或檢視中使用動態目錄參考，請以逸出雙引號 (\」) 括住參考。例如： `SELECT * FROM \"glue:arn:aws:glue:<region>:<data account id>:catalog\".<database name>.<table name>` 如需詳細資訊，請參閱 Amazon Athena 使用者指南中的跨帳戶存取 AWS Glue 資料目錄。	資料消費者

相關資源

- 跨帳戶存取 AWS Glue 資料型錄 (Athena 文件)
- create-data-catalog(AWS CLI 命令參考)
- 透過 Amazon Athena 存取跨帳戶 AWS Glue 資料目錄 (AWS 大數據部落格)

- IAM 中的安全最佳做法 (IAM 文件)

其他資訊

使用 Lake Formation 作為跨帳戶共享的替代方案

您也可以使用 AWS Lake Formation 跨帳戶共用 AWS Glue 目錄物件的存取權。Lake Formation 在欄和資料列層級提供精細的存取控制、以標籤為基礎的存取控制、ACID 交易的受控表格，以及其他功能。儘管 Lake Formation 與 Athena 完全集成，但與此模式的僅 IAM 方法相比，它確實需要額外的配置。我們建議您考慮在整體解決方案架構的更廣泛環境中使用 Lake Formation 或僅限 IAM 存取控制的決定。考量事項包括涉及哪些其他服務，以及它們如何與這兩種方法整合。

跨帳戶資料共用自動化

由伊薩姆·哈比比（AWS），路易斯·霍爾卡德（AWS）和瑪達萊娜卡爾沃（AWS）創建

環境：PoC 或試點	技術：資料湖、分析	工作負載：所有其他工作
AWS 服務：AWS AWS Glue；AWS Lake Formation；AWS RAM；Amazon Athena		

Summary

在一個組織中擁有多個獨立業務單位 (BUS) 意味著對資料湖存取權限的嚴格控制應該是首要任務，而且每個 BU 必須只存取自己的資料。但是，BU 的工作負載可能會對另一個 BU 感興趣，用於分析目的，這引起了對跨 BU 數據共享主題的興趣，並具有細微的權限控制。

在這個 apg 中，我們假設 BU 對應到託管其資料的 AWS 帳戶 (Glue 從 S3 抓取資料庫)，因此，跨 BU 資料共用會變成 AWS 跨帳戶資料共用問題。我們將提供一種自動化方式，使用 Lake Formation 與外部 AWS 帳戶的主體共用 Glue 資料庫的特定表格。此自動化可讓資料擁有者授與外部匯流排在定義的資料表上執行分析查詢 (例如使用 Athena) 的權限。

您可以使用此自動化解決方案來滿足典型的使用案例，例如：

人力資源資料團隊將託管在來源 AWS 帳戶中，該帳戶將與資料分析師團隊的目標 AWS 帳戶共用薪資表，以便使用 Athena 進一步查詢。

先決條件和限制

前提

對於此部署，您將需要：

- 具有足夠許可的兩個 AWS 帳戶 (來源帳戶和目標帳戶)，可部署此程式碼中封裝的 AWS 資源
- aws-cdk：全局安裝（故宮安裝-g aws-cdk）
- 混帳客戶端

- 至少一個包含表格的編目 Glue 資料庫。
- 在史詩部分展示了一些手動 Lake Formation 配置

限制

- 此解決方案需要 AWS 來源帳戶上已經編目的 Glue 資料庫。

- 此解決方案尚未提供撤銷授予權限的自動化方法。將來源帳戶中的資料共用到目標帳戶後，應在 Lake Formation 主控台上手動撤銷存取權。

架構

方案概觀

此 CDK 代碼部署了下圖中總結的體系結構

它特別包括：

來源帳戶堆疊：

- DynamoDb table：此表格包含使用者上傳的共用權限定義。它已啟用 DynamoDb 串流，並為新增至資料表的每個共用權限項目觸發 Lambda。
- lambda 函數：將資料表上的指定權限授與外部主體。

目標帳戶堆疊：

- Resource Access Manager（RAM）：接收來自 Lake Formation 的邀請。應該接受邀請，以便授予對共享數據的訪問權限。
- Amazon SQS：從來源帳戶接收訊息，指出共用程序已啟動
- EventBridge 規則：一旦接受 RAM 邀請，就會觸發此規則。
- 兩個 Lambda 函數：一個由自動接受 RAM 邀請的 SQS 佇列觸發，另一個由建立本機共用資料庫的 EventBridge 規則觸發的第二個函數，以及共用資源的資源連結。您可以向 Athena 進一步查詢這些資源連結。

該過程可以通過以下步驟進行總結：

1-使用者在來源帳戶的 DynamoDB 表格中上傳共用定義項目。

2- DynamoDb 串流會觸發來源帳戶 lambda，該帳戶使用湖泊形成與目標帳戶共用共用共用定義項目中指定的資料庫資料表。此共用會自動傳送 RAM 邀請至目標帳戶。

3-源帳戶 lambda 還將消息發送到目標帳戶中的 SQS 隊列，以提醒其共享過程的開始。

4-在目標帳戶上，SQS 隊列觸發接受收到的 RAM 邀請的 lambda。

5-接受邀請後， EventBridge 規則會觸發 lamdba，以創建本地數據庫和將包含共享表的資源鏈接。此 lambda 也會將共用資料的權限授予目標主體。

6-主體能夠使用 Athena 查詢數據。

工具

代碼存儲庫

此模式的代碼可在 G itlab 上找到

最佳實務

- 如前所述，這是強制性的，您的帳戶中有一個已經 Glue 抓取的數據庫。
- 資料庫名稱和表格名稱應與 Glue 編目資料庫中的名稱相符。
- 要插入 DynamoDB 的共用輸入項目應該是這樣的：

史诗

複製儲存庫並設定部署

任務	描述	所需技能
克隆存儲庫	克隆計算機上的 gitlab 存儲庫 `git clone git@ssh.gitlab.aws.dev:ihabibi/cross-account-data-sharing.git`	一般 AWS

任務	描述	所需技能
	`cd cross-account-data-sharing`	
設定您的部署	編輯`resources.py` 檔案，其中包含有關區域、您正在使用的來源/目標帳戶以及目標主體 arn 的資訊 `AWS_REGION = 'eu-west-1'` `AWS_SOURCE_ACCOUNT_ID = '111111111111'` `AWS_TARGET_ACCOUNT_ID = '222222222222'` `TARGET_PRINCIPAL_ARN = 'arn:aws:iam::222222222222:role/admin'`	一般 AWS

啟動您的 AWS 帳戶並部署程式碼

任務	描述	所需技能
啟動您的來源 AWS 帳戶	如果尚未完成，您需要在部署此 CDK 應用程式之前啟動 AWS 環境。 使用來源 AWS 帳戶的 AWS 登入資料執行下列命令： `cdk bootstrap aws://<source-account-id>/<aws-region>`	一般 AWS
部署來源 CDK 堆疊	現在您的來源 AWS 帳戶已啟動載入，而且您已設定部署，	一般 AWS

任務	描述	所需技能
	您可以使用下列命令部署 CDK 應用程式： （確保您位於 cross-account-data-sharing/目錄中） `cdk deploy SourceAccountStack`	
啟動您的目標 AWS 帳戶	如果尚未完成，您需要在部署此 CDK 應用程式之前啟動 AWS 環境。 使用目標 AWS 帳戶的 AWS 登入資料執行以下命令： `cdk bootstrap aws://<target-account-id>/<aws-region>`	一般 AWS
部署目標 CDK 堆疊	現在您的目標 AWS 帳戶已啟動載入，並且已設定部署，您可以使用下列命令部署 CDK 應用程式： （確保您位於 cross-account-data-sharing/目錄中） `cdk deploy TargetAccountStack`	一般 AWS

在源帳戶上設置 Lake Formation

任務	描述	所需技能
在源帳戶上設置 Lake Formation	• 在來源帳戶上，登入 Lake Formation 主控台，然後移至 [註冊並擷取] —> [資料湖位置]。註冊資料的 S3 位置。 • 移至 [權限] —> [資料湖權限]。撤銷所有 AllowedGroup IAM 許可。	

測試跨帳戶共享

任務	描述	所需技能
從來源帳戶共用資料表到目標帳戶	• 登錄到源帳戶的控制台轉到 DynamoDb 並查找「permissions_table」表並在此模式下插入一個項目。您也可以使用 AWS CLI `{ "share_id": "1", "table_name": "sample_data", "database_name": "database-ohio", "permissions": "DESCRIBE,SELECT", "source_acc_id": "111111111111", "target_acc_id": "222222222222" }`	一般 AWS

任務	描述	所需技能
	一旦項目被插入到表中，它會觸發整個過程，並且該表應該在幾秒鐘內在目標帳戶上進行查詢。 • 請注意，可能的權限是描述，選擇。他們應該用逗號分隔。	
查詢目標帳戶上的資料表	• 登錄到目標帳戶的控制台，您會發現 Lake Formation 已經識別了共享表，您可以使用 Athena 進行查詢。	

相關資源

代碼在吉特實驗室

其他資訊

主要使用的服務的文檔：

Amazon DynamoDb

AWS Lambda

AWS Lake Formation

AWS Glue

AWS Resource Access Manager

Amazon SQS

使用基礎設施即程式碼在 AWS 雲端部署和管理無伺服器資料湖

環境：生產	技術：資料湖；分析；無伺服器； DevOps	工作負載：所有其他工作
AWS 服務：Amazon S3; Amazon SQS; AWS; AWS AWS Glue CloudForm ation; Amazon; AWS Lambda CloudWatch; AWS Step Functions; Amazon DynamoDB		

Summary

此模式說明如何使用無伺服器運算和基礎設施即程式碼 (IaC) 在 Amazon Web Services (AWS) 雲端上實作和管理資料湖。此模式是以 AWS 開發的無伺服器資料湖架構 (SDLF) 研討會為基礎。

SDLF 是可重複使用的資源集合，可加速 AWS 雲端上企業資料湖的交付速度，並協助更快地部署到生產環境。它是用來通過遵循最佳實踐來實現數據湖的基礎結構。

SDLF 使用 AWS CodePipeline、AWS 和 AWS 等 AWS 服務，在整個程式碼和基礎設施部署中實作持續整合/持續部署 (CI/CD) 程序。 CodeBuild CodeCommit

此模式使用多個 AWS 無伺服器服務來簡化資料湖管理。其中包括用於儲存的亞馬遜簡單儲存服務 (Amazon S3) 和 Amazon DynamoDB、適用於運算的 AWS Lambda 和 AWS Glue，以及亞馬遜 CloudWatch 活動、亞馬遜簡單佇列服務 (Amazon SQS) 以及用於協調的 AWS Step Functions。

AWS CloudFormation 和 AWS 程式碼服務充當 IaC 層，透過輕鬆的操作和管理提供可重複且快速的部署。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。

- 一個 Git 客戶端，安裝和配置。
- SDLF 工作坊，在網頁瀏覽器視窗中開啟，即可使用。

架構

架構圖說明了事件驅動的過程，其中包含以下步驟。

1. 將檔案新增至原始資料 S3 儲存貯體後，Amazon S3 事件通知會放置在 SQS 佇列中。每個通知都以 JSON 檔案形式傳送，其中包含 S3 儲存貯體名稱、物件金鑰或時間戳記等中繼資料。
2. Lambda 函數會使用此通知，該函數會根據中繼資料將事件路由至正確的擷取、轉換和載入 (ETL) 程序。Lambda 函數也可以使用儲存在 Amazon DynamoDB 表格中的關聯式組態。此步驟可讓資料湖中的多個應用程式進行解耦和擴充。
3. 事件會路由至 ETL 程序中的第一個 Lambda 函數，該函數會將資料從原始資料區域轉換並移動到資料湖的暫存區。第一步是更新綜合目錄。這是一個 DynamoDB 表格，其中包含資料湖的所有檔案中繼資料。此表中的每一列都包含存放在 Amazon S3 中之單一物件的操作中繼資料。對 Lambda 函數進行同步呼叫，該函數在 S3 物件上執行輕度轉換是一種經濟實惠的作業 (例如，將檔案從一種格式轉換為另一種格式)。由於已將新物件新增至暫存 S3 儲存貯體，因此會更新完整目錄，並將訊息傳送至 SQS 佇列，等待 ETL 中的下一個階段。
4. CloudWatch 事件規則每 5 分鐘觸發一個 Lambda 函數。此函數會檢查訊息是否已從先前的 ETL 階段傳遞至 SQS 佇列。如果傳遞訊息，Lambda 函數會從 ETL 程序中的 AWS Step Functions 數開始第二個函數。
5. 然後在一批文件上應用繁重的轉換。這種繁重的轉型是一項運算成本高昂的操作，例如同步呼叫 AWS Glue 任務、AWS Fargate 任務、Amazon EMR 步驟或 Amazon 筆記本。 SageMaker 表中繼資料是使用 AWS Glue 編目程式 (更新 AWS Glue 目錄) 從輸出檔案擷取。檔案中繼資料也會新增至 DynamoDB 中的完整型錄表格。最後，也會執行利用 Deequ 的資料品質步驟。

技術, 堆棧

- Amazon CloudWatch 活動
- AWS CloudFormation
- AWS CodePipeline

- AWS CodeBuild
- AWS CodeCommit
- Amazon DynamoDB
- AWS Glue
- AWS Lambda
- Amazon S3
- Amazon SQS
- AWS Step Functions

工具

- Amazon CloudWatch 活動 — CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS CloudFormation — CloudFormation 協助您以預測和重複的方式建立和佈建 AWS 基礎設施部署。
- AWS CodeBuild — CodeBuild 是全受管的建置服務，可編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit — CodeCommit 是 AWS 託管的版本控制服務，可用於私有存放和管理資產 (例如原始碼和二進位檔案)。
- AWS CodePipeline — CodePipeline 是一種持續交付服務，可用於建立模型、視覺化和自動化持續發行軟體變更所需的步驟。
- Amazon DynamoDB — DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及可擴展性。
- AWS Glue — AWS Glue 是全受管 ETL 服務，可讓您更輕鬆地準備和載入資料以進行分析。
- AWS Lambda — L ambda 支援執行程式碼，無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — 亞馬遜簡單儲存服務 (Amazon S3) 是可高度擴展的物件儲存服務。Amazon S3 可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- AWS 步驟函數-AWS Step Functions 是無伺服器函數協調器，可讓您輕鬆地將 AWS Lambda 函數和多個 AWS 服務排序為關鍵業務應用程式。
- Amazon SQS — Amazon Simple Queue Service (Amazon SQS) 是全受管訊息佇列服務，可協助您分離和擴展微型服務、分散式系統和無伺服器應用程式。

- Deequ — Deequ 是一種工具，可協助您計算大型資料集的資料品質指標、定義和驗證資料品質限制，並隨時掌握資料分佈變更的通知。

Code

SDLF 的原始程式碼和資源可在 AWS 實驗室 GitHub 儲存庫中取得。

史诗

設定 CI/CD 管線以佈建 IAC

任務	描述	所需技能
設定 CI/CD 管線以管理資料湖的 IaC。	登入 AWS 管理主控台，並按照 SDLF 研討會初始設定章節中的步驟操作。這會建立初始的 CI/CD 資源，例如為資料湖佈建和 CodePipeline 管理 IaC 的 CodeCommit 儲存庫、CodeBuild 環境和管道。	DevOps 工程師

版本控制 IAC

任務	描述	所需技能
複製本機電腦上的 CodeCommit 存放庫。	按照 SDLF 研討會的部署基礎部分中的步驟進行操作。這可協助您將裝載 IaC 的 Git 儲存庫複製到本機環境中。 如需詳細資訊，請參閱 CodeCommit 文件中的連線至 CodeCommit 儲存庫。	DevOps 工程師
修改 CloudFormation 範本。	使用本機工作站和程式碼編輯器，根據您的使用案例或需求修改 CloudFormation 範本。將	DevOps 工程師

任務	描述	所需技能
	它們提交到本地克隆的 Git 存儲庫。 如需詳細資訊，請參閱 AWS 文件中的使用 CloudFormation AWS CloudFormation 範本。	
將更改推送到存 CodeCommit 儲庫。	您的基礎結構代碼現在受到版本控制，並跟踪對代碼庫的修改。當您將變更推送至 CodeCommit 儲存庫時，CodePipeline 會自動將其套用至您的基礎結構，並將其傳遞至 CodeBuild。 重要：如果您在中使用 AWS SAM CLI CodeBuild，請執行`sam package`和`sam deploy`命令。如果您使用 AWS CLI，請執行`aws cloudformation package`和`aws cloudformation deploy` 命令。	DevOps 工程師

相關資源

設定 CI/CD 管線以佈建 IAC

- SDLF 工作坊 — 初始設置

版本控制 IAC

- SDLF 工作坊 — 部署基礎
- 連接到 CodeCommit 存儲庫
- 使用 AWS CloudFormation 範本

其他資源

- AWS 無伺服器資料分析管道參考架構
- SDLF 文件

使用 AWS IoT 資料以符合成本效益的方式，將物聯網資料直接導入 Amazon S3

由塞巴斯蒂安·維維亞尼（AWS）和里茲旺·賽德（AWS）創建

環境：PoC 或試點	技術：資料湖、分析、IoT	工作負載：開源
AWS 服務：AWS IoT Greengrass；Amazon S3；亞馬 Amazon Athena		

Summary

此模式說明如何使用 AWS IoT Greengrass 第 2 版裝置，以符合成本效益的方式將物聯網 (IoT) 資料直接導入 Amazon 簡單儲存體 (Amazon S3) 儲存貯體。裝置執行可讀取 IoT 資料的自訂元件，並將資料儲存在永久性儲存裝置 (亦即本機磁碟或磁碟區) 中。然後，裝置會將 IoT 資料壓縮為 Apache Parquet 檔案，並定期將資料上傳到 S3 儲存貯體。

您擷取的 IoT 資料量和速度僅受邊緣硬體功能和網路頻寬的限制。您可以使用 Amazon Athena 以符合成本效益的方式分析擷取的資料。Athena 支持使用 Amazon 託管 Graf ana 壓縮 Apache 實木複合地板文件和數據可視化。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 AWS IoT Greengrass 第 2 版上執行並從感應器收集資料的邊緣閘道 (資料來源和資料收集程序超出此模式的範圍，但您幾乎可以使用任何類型的感應器資料。 此模式使用具有在本機發佈資料的感測器或閘道的本機 MQTT 代理程式)。
- AWS IoT Greengrass 件、角色和開發套件相依性
- 用於將資料上傳到 S3 儲存貯體的串流管理器元件
- AWS SDK for Java 發套件、AWS 開發套件或適用 JavaScript於 Python 的 AWS 開發套件 (Boto3) 以執行 API

限制

- 此模式中的資料不會即時上傳到 S3 儲存貯體。有一個延遲時間，您可以配置延遲時間。資料會暫時緩衝在 Edge 裝置中，然後在期限到期後上傳。
- 此開發套件僅適用於 Java、Node.js 和 Python。

架構

目標技術堆疊

- Amazon S3
- AWS IoT Greengrass
- MQTT 經紀商
- 流管理器組件

目標架構

下圖顯示旨在擷取 IoT 感應器資料並將該資料存放在 S3 儲存貯體的架構。

該圖顯示以下工作流程：

1. 多個感測器（例如溫度和閥門）更新發佈給當地的 MQTT 代理商。
2. 訂閱這些感應器的 Parquet 檔案壓縮程式會更新主題並接收這些更新。
3. Parquet 檔案壓縮程式會將更新儲存在本機。
4. 期限過後，儲存的檔案會壓縮到 Parquet 檔案中，並傳遞至串流管理員以上傳到指定的 S3 儲存貯體。
5. 流管理器將 Parquet 文件上傳到 S3 存儲桶。

注意：串流管理員 (`StreamManager`) 是受管理的元件。如需如何將資料匯出到 Amazon S3 的範例，請參閱 AWS IoT Greengrass 文件中的串流管理員。您可以使用本地 MQTT 代理作為組件或其他代理像 Eclipse 蚊子。

工具

AWS 工具

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS IoT Greengrass 是開放原始碼 IoT 邊緣執行階段和雲端服務，可協助您在裝置上建置、部署和管理 IoT 應用程式。

其他工具

- Apache 的實木複合地板是一種開源的面向列的數據文件格式，專為存儲和檢索。
- MQTT (訊息佇列遙測傳輸) 是專為受限裝置所設計的輕量型訊息通訊協定。

最佳實務

對上傳的數據使用正確的分區格式

S3 儲存貯體中的根前置詞名稱沒有特定需求 (例如，"myAwesomeDataSet/"或"dataFromSource")，但我們建議您使用有意義的分區和前置詞，以便輕鬆瞭解資料集的用途。

我們還建議您在 Amazon S3 中使用正確的分區，以便查詢在資料集上以最佳方式執行。在下列範例中，資料會以 HIVE 格式分割，以便最佳化每個 Athena 查詢掃描的資料量。這樣可以改善效能並降低成本。

```
s3://<ingestionBucket>/<rootPrefix>/year=YY/month=MM/day=DD/
HHMM_<suffix>.parquet
```

史诗

設定您的環境

任務	描述	所需技能
建立 S3 儲存貯體。	1. 建立 S3 儲存貯體或使用現有儲存貯體。 2. 為您要擷取 IoT 資料的 S3 儲存貯體建立有意義的前置詞 (例如，s3:\\<bucket>\<prefix>)。	應用程式開發人員

任務	描述	所需技能
	3. 記錄您的前綴以供日後使用。	

任務	描述	所需技能
將 IAM 許可新增至 S3 儲存貯體。	若要授予使用者對先前建立之 S3 儲存貯體和前置詞的寫入權限，請將以下 IAM 政策新增至您的 AWS IoT Greengrass 角色： { "Version": "2012-10-17", "Statement": [{ "Sid": "S3DataUpload", "Effect": "Allow", "Action": ["s3:List*", "s3:Put*"], "Resource": ["arn:aws:s3:::<ingestionBucket>", "arn:aws:s3:::<ingestionBucket>/<prefix>/*"] }] } 如需詳細資訊，請參閱 Aurora 文件中的建立 IAM 政策以存取 Amazon S3 資源。	應用程式開發人員

任務	描述	所需技能
	接下來，更新 S3 儲存貯體的資源政策 (如果需要)，以允許使用正確的 AWS 主體進行寫入存取。	

建置和部署 AWS IoT Greengrass 件

任務	描述	所需技能
更新元件的配方。	根據下列範例建立部署時，請更新元件組態： `{ "region": "<region>", "parquet_period": <period>, "s3_bucket": "<s3Bucket>", "s3_key_prefix": "<s3prefix>" }` 以您<region>的 AWS 區域、<period>定期間隔、<s3Bucket> S3 儲存貯體和<s3prefix> 前置詞取代。	應用程式開發人員
建立元件。	執行以下任意一項： • 建立元件。 • 將元件新增至 CI/CD 配管 (如果有的話)。請務必將成品從成品儲存庫複製到 AWS IoT Greengrass 成品儲存貯	應用程式開發人員

任務	描述	所需技能
	體。然後，建立或更新您的 AWS IoT Greengrass 件。 • 將 MQTT 代理程式新增為元件，或稍後手動新增。附註：此決定會影響您可以搭配 Broker 使用的驗證配置。手動新增代理程式會將代理程式與 AWS IoT Greengrass 分離，並啟用代理程式的任何受支援的身份驗證配置。AWS 提供的代理程式元件具有預先定義的身份驗證 如需詳細資訊，請參閱 MQTT 3.1.1 代理商 (模型) 和 MQTT 5 代理商 (EMQX)。	
更新 MQTT 用戶端。	範例程式碼不會使用驗證，因為元件會在本機連線至 Broker。如果您的案例不同，請視需要更新 MQTT 用戶端區段。此外，請執行下列動作： 1. 更新訂閱中的 MQTT 主題。 2. 視需要更新 MQTT 訊息剖析器，因為來自每個來源的訊息可能不同。	應用程式開發人員

將元件新增至 AWS IoT Greengrass 版本 2 核心裝置

任務	描述	所需技能
更新核心裝置的部署。	如果 AWS IoT Greengrass 第 2 版核心裝置的部署已經存	應用程式開發人員

任務	描述	所需技能
	在，請修改部署。如果部署不存在，請建立新部署。 若要為元件指定正確的名稱，請根據下列項目更新新元件的記錄檔管理員組態 (如果需要)： `{ "logsUploaderConfiguration": { "systemLogsConfiguration": { ... }, "componentLogsConfigurationMap": { "<com.iot.ingest.parquet>": { "minimumLogLevel": "INFO", "diskSpaceLimit": "20", "diskSpaceLimitUnit": "MB", "deleteLogFileAfterCloudUpload": "false" } ... } }, "periodicUploadIntervalSec": "300" }` 最後，完成 AWS IoT Greengrass 核心裝置的部署修訂。	

驗證資料擷取到 S3 儲存貯體

任務	描述	所需技能
檢查 AWS IoT 綠色磁碟區的日誌。	檢查以下各項： • MQTT 用戶端已成功連線至本機 MQTT 代理程式。 • MQTT 用戶端已訂閱正確的主題。 • 感應器更新訊息會傳送至 MQTT 主題的代理程式。 • 實木複合地板壓縮發生在每個週期間隔	應用程式開發人員
檢查 S3 存儲桶。	確認資料是否正在上傳到 S3 儲存貯體。您可以看到每個時期正在上傳的文件。 您也可以查詢下一節中的資料，以確認資料是否已上傳至 S3 儲存貯體。	應用程式開發人員

從 Athena 設定查詢

任務	描述	所需技能
創建一個數據庫和表。	1. 建立 AWS Glue 資料庫 (如有需要)。 2. 在 AWS Glue 中手動建立表格，或透過在 AWS Glue 中執行爬蟲程式來建立表格。	應用程式開發人員
授予 Athena 對資料的存取權。	1. 更新許可以允許 Athena 存取 S3 儲存貯體。如需詳細資訊，請參閱 Athena 文件	應用程式開發人員

任務	描述	所需技能
	中 AWS Glue 資料型錄中對資料庫和表格的精細存取。 2. 查詢資料庫中的資料表。	

故障診斷

問題	解決方案
MQTT 用戶端無法連線	• 驗證 MQTT 代理程式的權限。如果您擁有來自 AWS 的 MQTT 代理程式，請參閱 MQTT 3.1.1 代理商 (模型) 和 M QTT 5 代理商 (EMQX)。 • 驗證 MQTT 用戶端上的認證。如果您擁有來自 AWS 的 MQTT 代理程式，請參閱 MQTT 3.1.1 代理商 (模型) 和 M QTT 5 代理商 (EMQX)。
MQTT 用戶端無法訂閱	驗證 MQTT 代理程式的權限。如果您擁有來自 AWS 的 MQTT 代理程式，請參閱 MQTT 3.1.1 代理商 (模型) 和 M QTT 5 代理商 (EMQX)。
鑲木地板文件沒有被創建	• 驗證 MQTT 主題是否正確。 • 確認來自感應器的 MQTT 訊息格式正確。
物件未上傳至 S3 儲存貯體	• 確認您具有網際網路連線能力和端點連線能力。 • 確認 S3 儲存貯體的資源政策正確無誤。 • 驗證 AWS IoT Greengrass 第 2 版核心裝置角色的許可。

相關資源

- DataFrame（熊貓文檔）

- 阿帕奇拼花文檔（鑲木地板文檔）
- 開發 AWS IoT Greengrass 件 (AWS IoT Greengrass 人員指南，第 2 版)
- 將 AWS IoT Greengrass 件部署到裝置 (AWS IoT Greengrass 開發人員指南，第 2 版)
- 與本機 IoT 裝置互動 (AWS IoT 環境開發人員指南，第 2 版)
- MQTT 3.1.1 代理程式 (模型) (AWS IoT 大規模開發人員指南，第 2 版)
- MQTT 5 代理程式 (EMQX) (AWS 物聯網環境開發人員指南，第 2 版)

其他資訊

成本分析

下列成本分析案例示範此模式涵蓋的資料擷取方法如何影響 AWS 雲端中的資料擷取成本。此案例中的定價範例是以發佈時的價格為基礎。價格可能變動。此外，您的成本可能會因 AWS 區域、AWS 服務配額以及與雲端環境相關的其他因素而有所不同。

輸入訊號設定

此分析使用下列一組輸入訊號做為比較 IoT 擷取成本與其他可用替代方案的基礎。

信號數	Frequency (頻率)	每個訊號的資料
125	二十五赫茲	8 位元組

在這種情況下，系統接收 125 個信號。每個信號為 8 個字節，每 40 毫秒（25 Hz）發生一次。這些信號可以單獨出現或分組在一個共同的有效載荷中。您可以根據需要選擇拆分和打包這些信號。您還可以確定延遲。延遲包括接收、累積和擷取資料的時間段。

為了比較目的，此案例的擷取操作以 `us-east-1` AWS 區域為基礎。成本比較僅適用於 AWS 服務。其他成本 (例如硬體或連線能力) 不會計入分析中。

成本比較

下表顯示每個擷取方法的每月成本，以美元 (USD) 為單位。

方法	每月成本

AWS IoT SiteWise *	美元
含資料處理套件的 AWS IoT SiteWise Edge (將所有資料保留在邊緣)	二百美元
用於存取原始資料的 AWS IoT Core 和 Amazon S3 規則	美元
在邊緣實木複合地板文件壓縮並上傳到 Amazon S3	0.5 美元

* 資料必須縮減取樣才能符合服務配額。這意味著此方法存在一些數據丟失。

替代方法

本節顯示下列替代方法的等效成本：

- AWS IoT SiteWise — 每個訊號都必須在個別訊息中上傳。因此，每個月的消息總數是 125 × 25×3600×24×30，或者每月 81 億條消息。不過，AWS IoT 每秒只 SiteWise 能處理每個資產 10 個資料點。假設數據被縮減採樣到 10 赫茲，每月的消息數量減少到 125×10×3600×24 億或 3.24 十億。如果您使用的發行者元件會以 10 個群組封裝測量值 (每百萬則訊息 1 美元)，則每月收到 324 USD 的每月費用。假設每條消息都是 8 個字節（1 KB /125），那就是 25.92 GB 的數據存儲空間。這增加了每月 7.77 美元的每月費用。第一個月的總費用為 331.77 美元，每月增加 7.77 美元。
- AWS IoT SiteWise Edge 含資料處理套件，包括在邊緣完整處理的所有模型和訊號 (也就是無雲端擷取) — 您可以使用資料處理套件作為降低成本並設定在邊緣計算的所有模型的替代方案。即使沒有執行實際計算，這也可以僅用於存儲和可視化。在這種情況下，必須為邊緣閘道使用功能強大的硬體。每月有 200 美元的固定費用。
- MQTT 將原始資料直接擷取至 AWS IoT Core，以及將原始資料存放在 Amazon S3 的物聯網規則 — 假設所有訊號都發佈在一個共同的承載中，則發佈到 AWS IoT 核心的訊息總數為 25×3600×24×30，或每月 64.8 百萬。每百萬則訊息為 1 美元，每月的費用為 64.8 美元。每百萬個規則啟動 0.15 美元，每則訊息有一個規則，每月增加 19.44 美元的每月費用。Amazon S3 中每 GB 儲存的成本為 0.023 美元，每月再增加 1.5 美元 (每月增加以反映新資料)。第一個月的總費用為 84.54 美元，每月增加 1.5 美元。
- 壓縮 Parquet 檔案邊緣的資料並上傳到 Amazon S3 (建議的方法) — 壓縮率取決於資料類型。與 MQTT 相同的工業數據測試，整個月的總輸出數據為 1.2 Gb。這費用為每月 0.03 美元。其他基準測試中描述的壓縮比率（使用隨機數據）的順序為 66%（接近最壞情況）。總數據為 21 Gb，每月費用 0.5 美元。

拼花文件生成器

下列程式碼範例會示範以 Python 撰寫的 Parquet 檔案產生器的結構。程式碼範例僅用於說明目的，如果貼到您的環境中，將無法運作。

```
import queue
import paho.mqtt.client as mqtt
import pandas as pd

#queue for decoupling the MQTT thread
messageQueue = queue.Queue()
client = mqtt.Client()
streammanager = StreamManagerClient()

def feederListener(topic, message):
    payload = {
        "topic" : topic,
        "payload" : message,
    }
    messageQueue.put_nowait(payload)

def on_connect(client_instance, userdata, flags, rc):
    client.subscribe("#",qos=0)

def on_message(client, userdata, message):
    feederListener(topic=str(message.topic),
 message=str(message.payload.decode("utf-8")))


filename = "tempfile.parquet"
streamname = "mystream"
destination_bucket= "mybucket"
keyname="mykey"
period= 60

client.on_connect = on_connect
client.on_message = on_message
streammanager.create_message_stream(
            MessageStreamDefinition(name=streamname,
 strategy_on_full=StrategyOnFull.OverwriteOldestData)
        )


while True:
```

```
  try:
      message = messageQueue.get(timeout=myArgs.mqtt_timeout)
  except (queue.Empty):
      logger.warning("MQTT message reception timed out")

  currentTimestamp = getCurrentTime()
  if currentTimestamp >= nextUploadTimestamp:
      df = pd.DataFrame.from_dict(accumulator)
      df.to_parquet(filename)
      s3_export_task_definition = S3ExportTaskDefinition(input_url=filename,
bucket=destination_bucket, key=key_name)
      streammanager.append_message(streamname,
Util.validate_and_serialize_to_json_bytes(s3_export_task_definition))
      accumulator = {}
      nextUploadTimestamp += period
  else:
       accumulator.append(message)
```

使用萬 LiveData 迪斯科遷移器將 Hadoop 資料遷移到 Amazon S3

來源：內部部署 Hadoop 群集	目標：Amazon S3	R 類型：重新主機
環境：生產	技術：資料湖、大數據、混合雲、遷移	工作負載：所有其他工作
AWS 服務：Amazon S3		

Summary

此模式描述了將 Apache Hadoop 數據從 Hadoop 分佈式文件系統（HDFS）遷移到 Amazon Simple Storage Service (Amazon S3) 的過程。它使用 WanDisco LiveData 遷移程序來自動化數據遷移過程。

先決條件和限制

先決條件

- Hadoop 的集群邊緣節點，其中 LiveData 遷移將被安裝。節點應符合下列需求：
  - 最低規格：4 個 CPU，16 GB 內存，100 GB 存儲空間。
  - 最低 2 Gbps 的網路速度。
  - 可在邊緣節點上存取的連接埠 8081，以存取萬迪斯科使用者介面。
  - Java 1.8 64 位。
  - 安裝在邊緣節點上的 Hadoop 客戶端庫。
  - 能夠作為 HDFS 超級用戶進行身份驗證（例如，「hdfs」）。
  - 如果您的 Hadoop 叢集上已啟用 Kerberos，邊緣節點上必須有包含適合 HDFS 超級使用者主體的有效金鑰索引標籤。
  - 如需支援的作業系統清單，請參閱版本說明。
- 可存取 S3 儲存貯體的有效 AWS 帳戶。
- 在現場部署 Hadoop 叢集 (特別是邊緣節點) 和 AWS 之間建立的 AWS Direct Connect 連結。

產品版本

- LiveData 移民者

- 使用者介面 (一) 5.8.0

架構

源, 技術, 堆棧

- 內部部署 Hadoop 叢集

目標技術堆疊

- Amazon S3

架構

下圖顯示 LiveData 移轉程式解決方案架構。

此工作流程包含四個主要元件，用於將資料從現場部署 HDFS 移轉到 Amazon S3。

- LiveData 移轉程式 — 自動將資料從 HDFS 遷移到 Amazon S3，並駐留在 Hadoop 叢集的邊緣節點上。
- HDFS — 提供對應用程式資料的高輸送量存取的分散式檔案系統。
- Amazon S3 — 提供可擴展性、資料可用性、安全性和效能的物件儲存服務。
- AWS Direct Connect — 建立從現場部署資料中心到 AWS 的專用網路連線的服務。

自動化和規模

您通常會建立多個移轉，以便依路徑或目錄從來源檔案系統中選取特定的內容。您也可以透過定義多個移轉資源，同時將資料移轉至多個獨立的檔案系統。

史诗

在您的 AWS 帳戶中設定 Amazon S3 儲存

任務	描述	所需技能
登入 AWS 帳戶。	登入 AWS 管理主控台，然後前往 https://console.aws.amazon.com/s3/ 開啟 Amazon S3 主控台。	AWS 經驗
建立 S3 儲存貯體。	如果您尚未將現有的 S3 儲存貯體用作目標儲存，請在 Amazon S3 主控台上選擇「建立儲存貯體」選項，然後指定儲存貯體名稱、AWS 區域和儲存貯體設定以進行區塊公開存取。AWS 和 WanDisco 建議您啟用 S3 儲存貯體的區塊公開存取選項，並設定儲存貯體存取和使用者許可政策以符合組織的需求。一個 AWS 範例可在 https://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example1.html 上提供。	AWS 經驗

安裝 LiveData 移轉程式

任務	描述	所需技能
下載 LiveData 移轉程式安裝程式。	下載 LiveData 移轉程式安裝程式，並將其上傳到 Hadoop 邊緣節點。您可以在以下位置下載 LiveData 遷移者的免費試用版：https://www2.wand	Hadoop 管理員，應用程序所有

任務	描述	所需技能
	isco.com/ldm-trial。您也可以從 AWS Marketplace 取得 LiveData 移轉程式的存取權，網址為 https://aws.amazon.com/marketplace/pp/B07B8SZND9。	
安裝 LiveData 移轉程式。	使用下載的安裝程式並將 LiveData 移轉程式安裝為 Hadoop 叢集中邊緣節點上的 HDFS 超級使用者。如需安裝指令，請參閱「其他資訊」一節。	Hadoop 管理員，應用程序所有
檢查 LiveData 遷移程式和其他服務的狀態。	通過使用「其他信息」部 LiveData 分中提供的命令檢查遷移器，配置單元遷移器和 WanDisco UI 的狀態。	Hadoop 管理員，應用程序所有

透過使用者介面設定儲存

任務	描述	所需技能
註冊您的 LiveData 移民帳戶。	通過端口 8081（在 Hadoop 邊緣節點上）上的 Web 瀏覽器登錄到 WanDisco UI，並提供您的詳細信息進行註冊。例如，如果您在名為 myldmhost.example.com 的主 LiveData 機上執行移轉程式，網址將會是：http://myldmhost.example.com:8081	應用程式擁
設定您的來源 HDFS 儲存裝置。	提供來源 HDFS 儲存所需的組態詳細資料。這將包括「F	Hadoop 管理員，應用程序所有

任務	描述	所需技能
	S.DefaultFS」值和使用者定義的儲存名稱。如果啟用 Kerberos，請提供主參與者和索引標籤位置供 LiveData 移轉程式使用。如果叢集上已啟用 NameNode HA，請提供邊緣節點上 core-site.xml 和 hdfs-site.xml 檔案的路徑。	
設定您的目標 Amazon S3 儲存。	將目標儲存區新增為 s3a 類型。提供使用者定義的儲存名稱和 S3 儲存貯體名稱。針對登入資料提供者選項輸入「簡單AWSCredentialsProvider」，然後為 S3 儲存貯體提供 AWS 存取權和秘密金鑰。還需要其他 S3a 屬性。如需詳細資訊，請參閱 LiveData 移轉程式文件中的「S3a 屬性」一節，網址為 https://docs.wandisco.com/live-data-migrator/docs/command-reference/#filesystem-add-s3a。	AWS，應用程式擁有者

準備移轉

任務	描述	所需技能
新增排除項目 (如有需要)。	如果您想要從移轉中排除特定資料集，請為來源 HDFS 儲存區新增排除項目。這些排除可以基於文件大小，文件名（基於正則表達式模式）和修改日期。	Hadoop 管理員，應用程序所有

建立並開始移轉

任務	描述	所需技能
建立和設定移轉。	在 WanDisco 使用者介面的儀表板中建立移轉。選擇您的來源 (HDFS) 和目標 (S3 儲存貯體)。新增您在上一個步驟中定義的新排除項。選取「覆寫」或「大小相符時略過」選項。在所有欄位完成時建立移轉。	Hadoop 管理員，應用程序所有
開始移轉。	在儀表板上，選取您建立的移轉。按一下以開始移轉。您也可以在建立移轉時選擇自動啟動選項，以自動啟動移轉。	應用程式擁

管理頻寬 (選用)

任務	描述	所需技能
設定來源與目標之間的網路頻寬限制。	在儀表板的「儲存空間」清單中，選取您的來源儲存空間，然後在「分組」清單中選取「頻寬管理」。清除無限制選項，並定義最大頻寬限制和單位。選擇「應用」。	應用程式擁有者, 網

監視和管理移轉

任務	描述	所需技能
使用萬迪斯科使用者介面檢視移轉資訊。	使用 WanDisco 使用者介面來檢視授權、頻寬、儲存和移轉資訊。使用者介面也提供通知系統，讓您可以接收有關使用	Hadoop 管理員，應用程序所有

任務	描述	所需技能
	中錯誤、警告或重要里程碑的通知。	
停止、繼續和刪除移轉。	您可以將內容置於「已停止」狀態，以停止移轉將內容傳輸到其目標。停止的遷移可以恢復。您也可以刪除處於「已停止」狀態的移轉。	Hadoop 管理員，應用程序所有

相關資源

- LiveData 移轉程式文件
- LiveData AWS Marketplace 中的移轉工具
- 萬迪斯科支持社區
- 萬迪斯科 LiveData 遷徙者示範（視頻）

其他資訊

安裝 LiveData 移轉程式

假設安裝程式位於您的工作目錄中，您可以使用下列指令來安裝 LiveData Migrate 程式：

```
su - hdfs
chmod +x livedata-migrator.sh && sudo ./livedata-migrator.sh
```

安裝後檢查 LiveData 移轉程式和其他服務的狀態

使用下列命令來檢查 LiveData 移轉程式、蜂巢移轉程式和 WanDisco UI 的狀態：

```
service livedata-migrator status
service hivemigrator status
service livedata-ui status
```

更多模式

- 使用 AWS Glue 建立 ETL 服務管道，以遞增方式將資料從 Amazon S3 載入到亞馬遜紅移
- ???
- 確保亞 Amazon Redshift 叢集在建立時已加密
- 使用 AWS AWS Glue 任務和 Python 產生測試資料
- 使用星爆將資料遷移到 AWS 雲端
- 優化 AWS 上輸入檔案大小的 ETL 擷取
- 使用 AWS Step Functions 透過驗證、轉換和分割協調 ETL 管道
- ???
- 以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3
- 確認新的 Amazon Redshift 叢集具有必要的 SSL 端點
- 使用 Amazon 雅典娜和亞馬遜視覺化亞馬遜 Redshift 審核日 QuickSight

資料庫

主題

- 使用連結伺服器從 Amazon EC2 上的 Microsoft SQL 伺服器存取現場部署 Microsoft SQL 伺服器表
- 使用僅供讀取複本將 HA 新增至 Oracle PeopleSoft Amazon 自訂
- 評估將 SQL 伺服器資料庫遷移到 AWS 上的 MongoDB 地圖集的查詢效能
- 使用 DR 協調器架構自動化跨區域容錯移轉和容錯回復
- 自動化跨 AWS 帳戶複寫 Amazon RDS 執行個體
- 使用 Systems Manager 和自動備份 SAP HANA 資料庫 EventBridge
- 使用雲端託管人封鎖對 Amazon RDS 的公開存取
- 在 AWS 上的 SQL Server 中的「永遠開啟」可用性群組中設定唯讀路由
- 通過在 PGAdmin 中使用 SSH 隧道進行 Connect
- 將甲骨文查詢轉換為 SQL 數據庫
- 使用自訂實作跨帳戶複製 Amazon DynamoDB 表格
- 使用 AWS Backup 跨帳戶複製 Amazon DynamoDB 表
- 為 Amazon RDS 和 Amazon Aurora 創建詳細的成本和用量報告
- 使用 Aurora 中的自訂端點模擬 Oracle RAC 工作負載
- 在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線
- 加密現有的亞馬遜 RDS 資料庫執行個體
- 啟動時強制執行 Amazon RDS 資料庫的自動標記
- 估算隨需容量的 DynamoDB 表格的成本
- 估算 Amazon DynamoDB 表格的儲存成本
- 使用 AWR 報告估計甲骨文資料庫的 Amazon RDS 引擎大小
- 使用 AWS DMS 將 Amazon RDS for SQL Server 資料表匯出到 S3 儲存貯體
- 處理動態 SQL 語句中 Aurora PostgreSQL 塊
- 在 Aurora 兼容後處理過載的甲骨文功能
- 協助強制執行 DynamoDB 標記
- 使用 AWS DMS 和 Amazon Aurora 實作跨區域災難復原
- 將具有 100 個以上引數的甲骨文函數和程序遷移到 PostgreSQL
- 將適用於 Oracle 資料庫執行個體的 Amazon RDS 移轉到使用 AMS 的其他帳戶

- 將甲骨文輸出綁定變量遷移到 PostgreSQL 數據庫
- 使用具有相同主機名稱的 SAP 高速鐵路將 SAP HANA 遷移到 AWS
- 使用分散式可用性群組將 SQL 伺服器遷移到 AWS
- 使用 SharePlex 和 AWS DMS 從甲骨文 8i 或 9i 遷移到適用於甲骨文的亞馬遜 RDS
- 監控 Amazon Aurora 是否有沒有加密的
- 使用 Amazon 監控甲骨文 GoldenGate 日誌 CloudWatch
- 在適用於甲骨文的亞馬遜 RDS 上將 Oracle 數據庫企業版重新平台為標準版 2
- 使用精確 Connect 將大型主機資料庫複寫到 AWS
- 使用 Lambda 和機 Secrets Manager 為亞馬遜 RDS 和 Aurora PostgreSQL 安排任務
- 使用受信任的內容，在 AWS 上的 Db2 聯合資料庫中保護和簡化使用者存取
- 使用現場部署 SMTP 伺服器和資料庫郵件傳送 Amazon RDS for SQL Server 服器資料庫執行個體的通知
- 在 AWS 上的 IBM Db2 上為 SAP 設定災難復原
- 在 Amazon RDS 上為甲骨文電子商務套件設置 HA/DR 架構，並使用活動備用數據庫
- 使用 GTID 在亞馬遜 EC2 上設置 Amazon RDS for MySQL 和 MySQL 之間的數據複寫
- 甲骨文 PeopleSoft 應用程式在 Amazon RDS 自訂的轉換角色
- 依工作負載的資料庫移轉
- 更多模式

使用連結伺服器從 Amazon EC2 上的 Microsoft SQL 伺服器存取現場部署 Microsoft SQL 伺服器表

創建者：蒂魯馬拉·達薩里（AWS）和愛德華多·瓦倫丁（AWS）

環境：PoC 或試點	技術：資料庫	工作量：Microsoft

Summary

此模式描述如何從運行或託管在 Amazon 彈性計算雲（亞馬遜 EC2）視窗或 Linux 實例通過使用鏈接的服務器上運行的 Microsoft SQL Server 數據庫訪問在 Microsoft 視窗上運行的現場部署 Microsoft SQL Server 數據庫表。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon EC2 與 Microsoft SQL 服務器在 Amazon Linux AMI 上運行（Amazon 機器映像）
- AWS 在現場部署 Microsoft SQL 伺服器 (視窗) 伺服器與視窗或 Linux EC2 執行個體之間直接連接

產品版本

- SQL 伺服器 2016 年或更新版本

架構

源, 技術, 堆棧

- 在視窗上執行的內部部署 Microsoft SQL 伺服器
- Amazon EC2 與 Microsoft SQL 服務器在視窗 AMI 或 Linux AMI 上運行

目標技術堆疊

- Amazon EC2 與 Microsoft SQL 服務器在 Amazon Linux AMI 上運行

- Amazon EC2 與 Microsoft SQL 服務器在視窗 AMI 上運行

來源與目標資料庫架構

工具

- Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器基礎設施的集成環境。它提供了一個用戶界面和一組具有與 SQL Server 交互的豐富腳本編輯器的工具。

史诗

將 SQL 伺服器中 SQL 伺服器的驗證模式變更為視窗

任務	描述	所需技能
透過 SSMS Connect 到視窗 SQL 伺服器。		DBA
從 Windows SQL Server 執行個體的內容 (按一下滑鼠右鍵) 功能表，將驗證模式變更為 SQL Server 中的視窗。		DBA

重新啟動視窗服務

任務	描述	所需技能
重新啟動 SQL 服務。	1. 在「SSMS 物件總管」中，選擇 SQL 伺服器執行個體。 2. 開啟上下文 (按一下右鍵) 功能表。 3. 選擇重新啟動。	DBA

創建新的登錄並選擇數據庫訪問 Windows SQL 服務器

任務	描述	所需技能
在「安全性」標籤中，開啟「登入」的內容 (按一下滑鼠右鍵) 功能表，然後選取新的登入。		DBA
在 [一般] 索引標籤中，選擇 [SQL Server 驗證]、輸入使用者名稱、輸入密碼，然後確認密碼並清除下次登入時變更密碼的選項。		DBA
在伺服器角色索引標籤中，選擇公用。		DBA
在 [使用者對應] 索引標籤中，選擇要存取的資料庫和結構描述，然後反白顯示資料庫以選取資料庫角色。	選取公用和 db_datareader 以存取資料庫表格中的資料。	DBA
選擇「確定」以建立使用者。		DBA

將視窗 SQL 伺服器 IP 新增至 SQL 伺服器主機檔案

任務	描述	所需技能
透過終端機視窗 Connect 到 Linux SQL 伺服器方塊。		DBA
開啟 /etc/主機檔案，並使用 SQL 伺服器新增視窗機器的 IP 位址。		DBA
儲存主機檔案。		DBA

在 SQL 伺服器上建立連結伺服器

任務	描述	所需技能
通過使用存儲過程主服務器和主. dbo.sp_addlinkedsrvlogin 創建一個鏈接的服務器。	如需有關使用這些預存程序的詳細資訊，請參閱其他資訊一節。	DBA, 開發人員

驗證 SSMS 中創建的鏈接服務器和數據庫

任務	描述	所需技能
在 SSMS 中的 SQL 伺服器中，移至連結伺服器並重新整理。		DBA
展開左窗格中建立的連結伺服器和目錄。	您會看到選取的 SQL Server 資料庫與資料表和檢視表。	DBA

確認您可以存取視窗 SQL 伺服器資料庫資料表

任務	描述	所需技能
在 SSMS 查詢視窗中，執行以下查詢：「從 [方塊中選取前 3 名 *。	請注意，FROM 子句使用一個四部分的語法：計算機。數據庫（例如，選擇名稱「SQL2 數據庫」從 [SQL2]. 在我們的例子中，我們在 hosts 文件中創建了 SQL2 的別名，因此您不需要在方括號之間輸入實際的 NetBIOS 名稱。如果您確實使用了實際的 NetBIOS 名稱，請注意 AWS 預設使用 NetBIOS 名稱，例如贏-xxxx，而 SQL 伺服器則需要使用方括號來表示具有破折號的名稱。	DBA, 開發人員

相關資源

- 適用於 Linux 上 SQL 伺服器的版本資訊

其他資訊

使用預存程序建立連結伺服器

SSMS 不支援為 Linux SQL Server 建立連結伺服器，因此您必須使用這些預存程序來建立它們：

```
EXEC master.sys.sp_addlinkedserver @server= N'SQLLIN' , @srvproduct= N'SQL Server'
EXEC master.dbo.sp_addlinkedsrvlogin
 @rmtsrvname=N'SQLLIN',@useself=N'False',@locallogin=NULL,@rmtuser=N'username',@rmtpassword='Te
```

附註 1：在預存程序中輸入您先前在 Windows SQL 伺服器中建立的登入憑證`master.dbo.sp_addlinkedsrvlogin`。

注意 2：`@server`名稱`SQLLIN`和主機文件條目名稱`172.12.12.4 SQLLIN`應相同。

您可以使用此程序建立下列案例的連結伺服器：

- Linux SQL 伺服器到視窗 SQL 伺服器透過連結伺服器 (如此模式中所指定)
- 視窗 SQL 服務器到 Linux SQL 服務器通過鏈接的服務器
- Linux SQL 服務器通過鏈接服務器到另一個 Linux SQL 服務器

使用僅供讀取複本將 HA 新增至 Oracle PeopleSoft Amazon 自訂

創建者：自動化（AWS）

環境：生產	技術：資料庫；基礎架構	工作量：甲骨文
AWS 服務：Amazon RDS		

Summary

若要在 Amazon Web Services (AWS) 上執行 Oracle PeopleSoft 企業資源規劃 (ERP) 解決方案，您可以使用 Amazon Relational Database Service 服務 (Amazon RDS) 或 Amazon RDS 自訂軟體，該服務支援需要存取基礎作業系統和資料庫環境的傳統、自訂和封裝應用程式。如需規劃移轉時要考量的關鍵因素，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

在撰寫本文時，適用於 Oracle 的 RDS 自訂不支援異地同步備份選項，該選項適用於 Amazon RDS 適用於 Oracle，使用儲存複寫作為 HA 解決方案。而是使用待命資料庫建立並維護主要資料庫的實體副本，此模式可達到 HA。該模式著重於使用 Oracle 資料保全設定僅供讀取複本，在具有 HA 的 Amazon RDS 自訂上執行 PeopleSoft 應用程式資料庫的步驟。

此模式也會將僅供讀取複本變更為唯讀模式。讓僅供讀取複本處於唯讀模式可提供額外的好處：

- 從主要資料庫卸載唯讀工作負載
- 使用「Oracle Active 資料保全」功能，從待命資料庫擷取狀況良好的區塊，啟用自動修復損毀的區塊
- 使用「遠端同步」功能，讓遠端待命資料庫保持同步，而不會產生與長距離重做日誌傳輸相關的效能額外負荷。

以唯讀模式使用複本需要「Oracle 作用中的資料保全」選項，此選項需要額外付費，因為它是「Oracle 資料庫企業版」的個別授權功能。

先決條件和限制

前提

- Amazon RDS 自定義上的現有 PeopleSoft 應用程序。如果您沒有應用程式，請參閱將甲骨文遷移 PeopleSoft 到 Amazon RDS 自訂模式。

- 單一 PeopleSoft 應用程式層。不過，您可以調整此模式來處理多個應用程式層級。
- Amazon RDS 自訂設定了至少 8 GB 的交換空間。
- 一種 Oracle Active Data Guard 資料庫授權，用於將僅供讀取複本轉換為唯讀模式，並使用該授權將報告作業卸載至待命模式。如需更多資訊，請參閱 Oracle 技術管理系統的商業價目表。

限制

- 適用於 Oracle 的 RDS 自訂的一般限制和不支援的組態
- 與適用於甲骨文僅供讀取複本的 Amazon RDS 自訂相關限制

產品版本

- 如需 Amazon RDS 自訂支援的 Oracle 資料庫版本，請參閱適用於 Oracle 的 RDS 自訂。
- 如需 Amazon RDS 自訂支援的 Oracle 資料庫執行個體類別，請參閱適用於 Oracle RDS 自訂的資料庫執行個體類別支援。

架構

目標技術堆疊

- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Oracle Active Data Guard
- 甲骨文 PeopleSoft 應用

目標架構

下圖顯示 Amazon RDS 自訂資料庫執行個體和 Amazon RDS 自訂僅供讀取複本。僅供讀取複本使用「Oracle 作用中資料保全」複寫到另一個可用區域。您也可以使用僅供讀取複本卸載主要資料庫上的讀取流量，以及用於報告用途。

如需使用 AWS PeopleSoft 上 Oracle 的代表性架構，請參閱在 AWS 上設定高可用性 PeopleSoft 架構。

工具

AWS 服務

- Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。在此模式中，您可以使用密碼名稱從「秘密管理員」擷取資料庫使用者密碼`do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg`。`RDS_DATAGUARD`

其他工具

- 「Oracle 資料保全」可協助您建立、維護、管理及監督待命資料庫。

最佳實務

若要實現零資料遺失 (RPO=0) 目標，請使用「`MaxAvailability`資料保全」保護模式與重做傳輸`SYNC+NOAFFIRM`設定，以獲得更好的效能。如需有關選取資料庫保護模式的詳細資訊，請參閱其他資訊一節。

史诗

建立僅供讀取複本

任務	描述	所需技能
建立僅供讀取複本。	若要建立 Amazon RDS 自訂資料庫執行個體的僅供讀取複本，請遵循 Amazon RDS 文件中的指示，並使用您建立的 Amazon RDS 自訂資料庫執行個體 (請參閱先決條件一節) 做為來源資料庫。 依預設，Amazon RDS 自訂僅供讀取複本建立為實體待命，並處於掛接狀態。這是故意確	DBA

任務	描述	所需技能
	保符合 Oracle 主動資料保全授權的規定。 此模式包含用於設定多租用戶容器資料庫 (CDB) 或非 CDB 執行個體的程式碼。	

將「Oracle 資料保全」保護模式變更為 MaxAvailability

任務	描述	所需技能
存取主要資料庫上的「資料保全」中介組態。	在此範例中，Amazon RDS 自訂僅供讀取複本`RDS_CUSTOM_ORCL_D` 適用於非 CDB 執行個體和 `RDS_CUSTOM_RDSCDB_B` CDB 執行個體。非 CDB 的資料庫為 `orcl_a` (主要) 和 `orcl_d` (待命)。CDB 的資料庫名稱為 `rdscdb_a` (主要) 和 `rdscdb_b` (待命)。 您可以直接或透過主要資料庫連線到 RDS Custom 僅供讀取複本。您可以在`$ORACLE_HOME/network/admin` 目錄中的`tnsnames.ora` 檔案中找到資料庫的網路服務名稱。適用於 Oracle 的 RDS 自訂會自動為您的主要資料庫和僅供讀取複本填入這些項目。 `RDS_DATAGUARD` 使用者的密碼會以密碼名稱儲存在 AWS Secrets Manager 中`do-not-`	DBA

任務	描述	所需技能
	delete-rds-custom-+<<RDS Resource ID>>+-dg。如需有關如何使用從 Secrets Manager 擷取的安全殼層 (安全殼層) 金鑰連線至 RDS 自訂執行個體的詳細資訊，請參閱使用 SSH 連線至 RDS 自訂資料庫執行個體。 若要透過「資料保全」命令行 (dgmgrl) 存取「Oracle 資料保全」中介組態，請使用下列程式碼。 非國家開發行 `$ dgmgrl RDS_DATAGUARD@RDS_CUSTOM_ORCL_D` `DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 22:44:49 2022` `Version 19.10.0.0.0` `Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.` `Welcome to DGMGRL, type "help" for information.` `Password:` `Connected to "ORCL_D"` `Connected as SYSDG.` `DGMGRL>` `DGMGRL> show database orcl_d` `Database - orcl_d` `Role: PHYSICAL STANDBY`	

任務	描述	所需技能
	Intended State: APPLY-ON Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 11.00 KByte/s Instance(s): ORCL SUCCESS DGMGRL> **國家開發行** -bash-4.2$ dgmgrl C##RDS_DATAGUARD@RDS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 20:24:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for information. Password: Connected to "RDSCDB_B" Connected as SYSDG. DGMGRL> DGMGRL> show database rdscdb_b Database - rdscdb_b	

任務	描述	所需技能
	Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 2.00 KByte/s Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS DGMGRL>	

任務	描述	所需技能
透過從主要節點連線到 DGMGRL 來變更記錄傳輸設定。	將記錄傳輸模式變更為FastSync，對應於重做傳輸設定SYNC+NOAFFIRM 。若要確保在角色切換之後擁有有效的設定值，請同時針對主要資料庫和待命資料庫進行變更。 非國家開發行 `DGMGRL>` `DGMGRL> edit database orcl_d set property logxptmode=fastsync;` `Property "logxptmode" updated` `DGMGRL> show database orcl_d LogXptMode;` `LogXptMode = 'fastsync'` `DGMGRL> edit database orcl_a set property logxptmode=fastsync;` `Property "logxptmode" updated` `DGMGRL> show database orcl_a logxptmode;` `LogXptMode = 'fastsync'` `DGMGRL>` 國家開發行 `DGMGRL> edit database rdscdb_b set property logxptmode=fastsync;DGMGRL> edit database rdscdb_b set property logxptmode=fastsync;`	DBA

任務	描述	所需技能
	Property "logxptmode" updated DGMGRL> show database rdscdb_b LogXptMode; LogXptMode = 'fastsync' DGMGRL> edit database rdscdb_a set property logxptmode=fastsync; Property "logxptmode" updated DGMGRL> show database rdscdb_a logxptmode; LogXptMode = 'fastsync' DGMGRL>	

任務	描述	所需技能
將保護模式變更為 MaxAvailability。	透過DGMGRL從主要節點MaxAvailability 連線，將保護模式變更為。 非國家開發行 `DGMGRL> edit configuration set protection mode as maxavailability;` `Succeeded.` `DGMGRL> show configuration;` `Configuration - rds_dg` `Protection Mode: MaxAvailability` `Members:` `orcl_a - Primary database` `orcl_d - Physical standby database` `Fast-Start Failover: Disabled` `Configuration Status:` `SUCCESS (status updated 38 seconds ago)` `DGMGRL>` 國家開發行 `DGMGRL> show configuration` `Configuration - rds_dg` `Protection Mode: MaxAvailability` `Members:` `rdscdb_a - Primary database` `rdscdb_b - Physical standby database`	DBA

任務	描述	所需技能
	Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 57 seconds ago) DGMGRL>	

將複本狀態從裝載變更為唯讀，並啟用重做套用

任務	描述	所需技能
停止待命資料庫的重做套用。	依預設，僅供讀取複本是以MOUNT模式建立的。若要以唯讀模式開啟它，您首先需要透過DGMGRL從主節點或待命節點連線至來關閉 redo apply。 非國家開發行 DGMGRL> show database orcl_dDGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 11.00 KByte/s Real Time Query: OFF Instance(s): ORCL Database Status: SUCCESS	DBA

任務	描述	所需技能
	DGMGRL> edit database orcl_d set state=app ly-off; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY- OFF Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 42 seconds (computed 1 second ago) Average Apply Rate: (unknown) Real Time Query: OFF Instance(s): ORCL Database Status: SUCCESS DGMGRL> **國家開發行** DGMGRL> show configura tionDGMGRL> show configuration Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status:	

任務	描述	所需技能
	SUCCESS (status updated 57 seconds ago) DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 2.00 KByte/s Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> edit database rdscdb_b set state=apply-off; Succeeded. DGMGRL> show database rdscdb_b; Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-OFF Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: (unknown)	

任務	描述	所需技能
	Real Time Query: OFF Instance(s): RDSCDB Database Status: SUCCESS	

任務	描述	所需技能
以唯讀模式開啟僅供讀取複本執行個體。	使用 TNS 項目 Connect 到待命資料庫，然後從主要或待命節點連線至該資料庫，以唯讀模式開啟它。 非國家開發行 `$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg` `-bash-4.2$ sqlplus RDS_DATAGUARD@RDS_CUSTOM_ORCL_D as sysdg` `SQL*Plus: Release 19.0.0.0.0 - Production on Fri Sep 30 23:00:14 2022` `Version 19.10.0.0.0` `Copyright (c) 1982, 2020, Oracle. All rights reserved.` `Enter password:` `Last Successful login time: Fri Sep 30 2022 22:48:27 +00:00` `Connected to:` `Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production` `Version 19.10.0.0.0` `SQL> select open_mode from v$database;` `OPEN_MODE` `--------------------` `MOUNTED` `SQL> alter database open read only;` `Database altered.`	DBA

任務	描述	所需技能
	SQL> select open_mode from v$database; OPEN_MODE -------------------- READ ONLY SQL>	
	國家開發行 -bash-4.2$ sqlplus C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_B as sysdg SQL*Plus: Release 19.0.0.0.0 - Productio n on Wed Jan 11 21:14:07 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2022, Oracle. All rights reserved. Enter password: Last Successful login time: Wed Jan 11 2023 21:12:05 +00:00 Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.16.0.0.0 SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- RDSCDB MOUNTED SQL> alter database open read only; Database altered.	

任務	描述	所需技能
	SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- RDSCDB READ ONLY SQL>	

任務	描述	所需技能
在僅供讀取複本執行個體上啟用重做套用。	使用主節點或待命節點的 DGMGR L，在僅供讀取複本執行個體上啟動重做套用。 非國家開發行 `$ dgmgrl RDS_DATAG UARD@RDS_CUSTOM_OR CL_D DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Sep 30 23:02:16 2022 Version 19.10.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "ORCL_D" Connected as SYSDG. DGMGRL> edit database orcl_d set state=apply-on; DGMGRL> edit database orcl_d set state=app ly-on; Succeeded. DGMGRL> show database orcl_d Database - orcl_d Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 0 seconds ago)`	DBA

任務	描述	所需技能
	Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 496.00 KByte/s Real Time Query: ON Instance(s): ORCL Database Status: SUCCESS DGMGRL> **國家開發行** -bash-4.2$ dgmgrl C##RDS_DATAGUARD@RDS_CUSTOM_RDSCDB_B -bash-4.2$ dgmgrl C##RDS_DATAGUARD@RDS_CUSTOM_RDSCDB_B DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 11 21:21:11 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for information. Password: Connected to "RDSCDB_B" Connected as SYSDG. DGMGRL> edit database rdscdb_b set state=apply-on; Succeeded.	

任務	描述	所需技能
	DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Average Apply Rate: 35.00 KByte/s Real Time Query: ON Instance(s): RDSCDB Database Status: SUCCESS DGMGRL> show database rdscdb_b Database - rdscdb_b Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Average Apply Rate: 16.00 KByte/s Real Time Query: ON Instance(s): RDSCDB Database Status: SUCCESS DGMGRL>	

相關資源

- 將 Amazon RDS 設定為甲骨文 PeopleSoft 資料庫 (AWS 白皮書)
- Oracle 資料保全中介指南 (Oracle 參考文件)
- 資料保全概念與管理 (Oracle 參考文件)

其他資訊

選取您的資料庫保護模式

Oracle 資料保全提供三種保護模式，可根據您的可用性、保護和效能需求來設定「資料保全」環境。下表摘要說明這三種模式。

保護模式	重做傳輸設定	Description
最高效能	ASYNC	對於主要資料庫上發生的交易，重做資料會以非同步方式傳輸，並寫入待命資料庫重做日誌。因此，效能的影響是最小的。 MaxPerformance 由於非同步記錄傳送，無法提供 RPO=0。
最大保護	SYNC+AFFIRM	對於主要資料庫上的交易，在確認交易之前，會同步傳輸重做資料，並將其寫入磁碟上的待命資料庫重做日誌。如果待命資料庫無法使用，主要資料庫會自行關閉，以確保交易受到保護。
最大可用性	SYNC+AFFIRM	這與MaxProtection 模式類似，除非沒有從待命資料庫接收到確認。在這種情況下，它會像處於MaxPerfor

	mance 模式一樣，以保留主要資料庫可用性，直到能夠再次將其重做串流寫入同步處理的待命資料庫為止。
SYNC+NOAFFIRM	對於主要資料庫上的交易，redo 會同步傳輸至待命資料庫，且主要資料庫只會等待重做已在待命時接收到的確認，而不是已寫入待命磁碟。此模式 (也稱為) 可提供效能優勢FastSync，但在多次同時發生故障的特殊情況下，可能會遭受資料遺失的潛在風險。

RDS 自訂適用於 Oracle 的僅供讀取複本是以最高效能保護模式建立的，這也是 Oracle 資料保全的預設保護模式。最大效能模式對主要資料庫的效能影響最低，可協助您符合以秒為單位測量的復原點目標 (RPO) 需求。

若要達成零資料遺失 (RPO=0) 目標，您可以MaxAvailability使用重做傳輸的SYNC+NOAFFIRM設定，將「Oracle 資料保全」保護模式自訂為，以獲得更好的效能。由於只有在對應的重做向量順利傳輸至待命資料庫之後才會確認主要資料庫上的認可，因此主要執行個體和複本之間的網路延遲對於認可敏感性工作負載而言至關重要。建議您對工作負載執行負載測試，以評估自訂僅供讀取複本以在MaxAvailability模式下執行時的效能影響。

與在不同可用區域部署僅供讀取複本相比，在主要資料庫相同的可用區域中部署僅供讀取複本可提供較低的網路延遲。但是，在相同的可用區域中部署主要和僅供讀取複本可能不符合 HA 需求，因為在可用區域無法使用的情況下，主要執行個體和僅供讀取複本執行個體都會受到影響。

評估將 SQL 伺服器資料庫遷移到 AWS 上的 MongoDB 地圖集的查詢效能

創建者：巴圖爾加·普列夫拉查（AWS），克里希納庫瑪·薩蒂亞納拉亞納（美國公司）和巴布斯里尼瓦森（MongoDB）PeerIslands

環境：PoC 或試點	來源：Microsoft SQL 服務器	目標：MongoDB 地圖集或 MongoDB 企業進階
R 類型：重新平台	工作量：Microsoft	技術：資料庫；移轉

Summary

此模式提供了載入 MongoDB 的指導，並儘可能接近實際的資料評估 MongoDB 查詢效能。評估提供的輸入可協助您規劃從關聯式資料庫移轉至 MongoDB。此病毒碼使用PeerIslands 測試資料產生器和效能分析器來測試查詢效能。

這種模式是 Microsoft SQL Server 遷移到 MongoDB 特別有用，因為執行結構描述轉換和從當前 SQL Server 實例加載數據到 MongoDB 可以是非常複雜的。相反地，您可以將接近真實世界的資料載入 MongoDB、瞭解 MongoDB 效能，並在開始實際移轉之前微調結構描述設計。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 熟悉 MongoDB 地圖集
- 目標 MongoDB 模式
- 典型的查詢模式

限制

- 數據加載時間和性能將受到 MongoDB 集群實例大小的限制。我們建議您選擇建議用於生產環境的執行個體，以瞭解真實世界的效能。
- PeerIslands 測試資料產生器和效能分析器目前僅支援線上資料載入和查詢。尚不支援離線批次處理 (例如，使用 Spark 連接器將資料載入 MongoDB)。

- PeerIslands 測試資料產生器和效能分析器支援集合中的現場關係。它不支持跨集合的關係。

產品版本

- 這種模式同時支持 MongoDB 地圖集和 MongoDB 企業高級版。

架構

目標技術堆疊

- 蒙古數據庫地圖集或 MongoDB 企業進階

架構

PeerIslands 測試資料產生器和效能分析器是使用 Java 和 Angular 建置的，並將其產生的資料存放在亞馬遜彈性區塊存放區 (Amazon EBS) 上。該工具包含兩個工作流程：測試數據生成和性能測試。

- 在測試數據生成中，您可以創建一個模板，這是必須生成的數據模型的 JSON 表示。建立範本之後，您可以依據負載產生組態的定義，在目標集合中產生資料。
- 在效能測試中，您可以建立設定檔。設定檔是一種多階段測試案例，您可以在其中設定建立、讀取、更新和刪除 (CRUD) 作業、彙總管線、每個作業的加權，以及每個階段的持續時間。建立設定檔之後，您可以根據組態在目標資料庫上執行效能測試。

PeerIslands 測試資料產生器和效能分析器會將其資料儲存在 Amazon EBS 上，因此您可以使用任何 MongoDB 支援的連線機制 (包括對等互連、允許清單和私有端點)，將 Amazon EBS 連接到 MongoDB。預設情況下，該工具不包括操作組件；但是，如果需要，可以使用適用於 Prometheus，Amazon 託管 Grafana，Amazon 和 AWS Secrets Manager 的 Amazon CloudWatch 託管服務進行配置。

工具

- PeerIslands 測試資料產生器和效能分析器包括兩個元件。測試資料產生器元件可協助您根據 MongoDB 結構描述產生高度客戶特定的真實世界資料。該工具是完全 UI 驅動的，具有豐富的數據庫，可用於在 MongoDB 上快速生成數十億條記錄。該工具還提供了實現 MongoDB 架構中字段之間的關係的功能。性能分析器組件可幫助您生成高度特定於客戶的查詢和聚合，並在 MongoDB 上執行

現實的性能測試。您可以使用效能分析器，針對特定使用案例使用豐富的負載設定檔和參數化查詢來測試 MongoDB 效能。

最佳實務

請參閱下列資源：

- MongoDB 架構設計最佳實踐（MongoDB 開發人員網站）
- 在 AWS 上部署 MongoDB 地圖集的最佳實務 (MongoDB 網站)
- 使用 AWS 將應用程式安全地連接到 MongoDB 地圖集資料平面 PrivateLink (AWS 部落格文章)
- MongoDB 效能最佳做法指南 (MongoDB 網站)

史诗

瞭解您的來源資料

任務	描述	所需技能
瞭解目前 SQL 伺服器來源的資料庫佔用空間。	瞭解您目前的 SQL 伺服器佔用空間。這可以通過對數據庫的 `INFORMATION` 模式運行查詢來實現。確定表的數量和每個表的大小。分析與每個表關聯的索引。如需有關 SQL 分析的詳細資訊，請參閱網站上的 SQL2Mongo：資料移轉旅程的部落格文章。 PeerIslands	DBA
瞭解來源結構描述。	決定資料表結構描述和資料的商務表示法 (例如郵遞區號、名稱和貨幣)。使用現有的實體關係（ER）圖或從現有數據庫生成 ER 圖。如需詳細資訊，請參閱網站上的部落格文章 SQL2Mongo：資料移轉之旅。 PeerIslands	DBA

任務	描述	所需技能
瞭解查詢模式。	記錄您使用的前 10 個 SQL 查詢。您可以使用資料庫中可用的資料庫中可用的資料表，以瞭解最常見的查詢。如需詳細資訊，請參閱網站上的部落格文章 SQL2Mongo：資料移轉之旅。 PeerIslands	DBA
瞭解 SLA 承諾。	記錄資料庫作業的目標服務層級協定 (SLA)。典型的測量包括查詢延遲和每秒查詢次數。這些措施及其目標通常在非功能性要求（NFR）文檔中可用。	DBA

定義 MongoDB 模式

任務	描述	所需技能
定義目標結構描述。	定義目標 MongoDB 架構的各種選項。如需詳細資訊，請參閱 MongoDB 地圖集文件中的結構描述。根據表格關係來考慮最佳作法和設計模式。有關詳細信息，請參閱 MongoDB 文檔中的數據模型示例和模式。	MongoDB 工程
定義目標查詢模式。	定義 MongoDB 的查詢和聚合管道。這些查詢等同於您為 SQL Server 工作負載擷取的常用查詢。若要瞭解如何建構 MongoDB 彙總管線，請參閱 MongoDB 文件。	MongoDB 工程

任務	描述	所需技能
定義 MongoDB 執行個體類型。	決定您打算用於測試的執行個體大小。如需指引，請參閱 MongoDB 文件。	MongoDB 工程

準備目標資料庫

任務	描述	所需技能
設定 MongoDB 地圖集叢集。	若要在 AWS 上設定 MongoDB 叢集，請依照 Mon goDB 文件中的指示進行。	MongoDB 工程
在目標資料庫中建立使用者。	依照 MongoDB 文件中的指示，設定 Mon goDB 地圖集叢集的存取和網路安全性。	MongoDB 工程
在 AWS 中建立適當的角色，並為 Atlas 設定角色型存取控制。	如果需要，請依照 MongoDB 文件中的指示設定其他使用者。透過 AWS 角色設定身份驗證和授權。	MongoDB 工程
為 MongoDB 地圖集訪問設置指南針。	設置 MongoDB 的指南針圖形用戶界面實用程序，便於導航和訪問。	MongoDB 工程

使用測試資料產生器設定基本負載

任務	描述	所需技能
安裝測試數據生成器。	在您的環境中安裝PeerIsland測試資料產生器。	MongoDB 工程
配置測試數據生成器以生成適當的數據。	通過使用數據庫來生成 MongoDB 模式中的每個字段的特定數據創建模板。有關更	MongoDB 工程

任務	描述	所需技能
	多信息，請參閱 MongoDB 的數據生成器和性能。視頻分析儀。	
水平縮放測試數據生成器以生成所需的負載。	您可以使用您建立的樣板，設定必要的平行程度，針對目標集合開始產生負載。確定時間範圍和規模以生成必要的數據。	MongoDB 工程
驗證 MongoDB 地圖集中的負載。	檢查加載到 MongoDB 地圖集的數據。	MongoDB 工程
在 MongoDB 上生成所需的索引。	根據查詢模式，視需要定義索引。如需最佳作法，請參閱 MongoDB 文件。	MongoDB 工程

進行性能測試

任務	描述	所需技能
在效能分析器中設定負載設定檔。	透過設定特定查詢及其對應的加權、測試執行的持續時間和階段，在效能分析器中建立效能測試設定檔。有關更多信息，請參閱 MongoDB 的數據生成器和性能。視頻分析儀。	MongoDB 工程
執行效能測試。	使用您建立的效能測試設定檔，透過設定所需的平行程度，對目標集合開始測試。水平擴展性能測試工具以對 MongoDB 地圖集運行查詢。	MongoDB 工程
記錄測試結果。	記錄查詢的 P95，P99 延遲。	MongoDB 工程

任務	描述	所需技能
調整結構描述和查詢模式。	修改索引和查詢模式以解決任何效能問題。	MongoDB 工程

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。	刪除用於「測試資料產生器」和「效能分析器」的所有暫存資源。	AWS 管理員
更新效能測試結果。	了解 MongoDB 的查詢性能，並將其與您的 SLA 進行比較。如有必要，請微調 MongoDB 結構描述並重新執行程序。	MongoDB 工程
完成該項目。	關閉專案並提供意見反應。	MongoDB 工程

相關資源

- GitHub 存儲庫：
- 模式：MongoDB 的模式設計
- 聚合管道：MongoDB 聚合管道
- MongoDB 地圖集大小:調整層選擇
- 視頻：MongoDB 的數據生成器和性能。分析儀
- 參考 MongoDB：文件
- 教程:MongoDB 開發人員指南, Mongo DB
- AWS Marketplace：AWS 市場上的 MongoDB 地圖集
- AWS 合作夥伴解決方案：AWS 上的 MongoDB 地圖集參考部署

其他資源：

- SQL 分析
- 開發 MongoDB 社區論壇
- MongoDB 的性能調整問題
- 運用阿特拉斯和 Redshift 作業分析
- 使用 MongoDB 地圖集和 AWS Elastic Beanstalk 進行應用程式現代化

使用 DR 協調器架構自動化跨區域容錯移轉和容錯回復

創建者：吉滕德拉·庫馬爾 (AWS)、奧利弗·弗朗西斯 (AWS) 和巴拉蘇布拉馬尼亞 (AWS)

代碼存儲庫：aws-cross-region-dr-數據庫	環境：生產	技術：資料庫；基礎架構；移轉；現代化
AWS 服務：Amazon Aurora; AWS CloudFormation; Amazon ElastiCache; Amazon RDS; AWS Step Functions		

Summary

此模式說明如何使用 DR 協調器架構來協調和自動化手動且容易出錯的步驟，以便跨 Amazon Web Services () 區域執行災難復原。AWS該模式涵蓋了以下數據庫：

- 適用於 MySQL 的 Amazon Relational Database Service 服務 (Amazon RDS)、適用於 PostgreSQL 的亞馬遜 RDS 或亞馬遜 RDS
- Amazon Aurora MySQL 兼容版或 Amazon Aurora PostgreSQL 兼容版（使用集中文件）
- Amazon ElastiCache 的雷迪斯

若要示範 DR 協調器架構的功能，您可以建立兩個資料庫執行個體或叢集。主要位於中 AWS 區域 `us-east-1`，次要位於中`us-west-2`。要創建這些資源，您可以使用 aws-cross-region-dr-data GitHub 存儲庫的`App-Stack`文件夾中的 AWS CloudFormation 模板。

先決條件和限制

一般先決條

- DR 協調器框架部署在主要和次要 AWS 區域
- 兩個 Amazon 簡單存儲服務桶
- 具有兩個子網路和一個安 AWS 全群組的虛擬私人雲端 (VPC)

引擎特定先決條件

- Amazon Aurora — 至少一個 Aurora 全球資料庫必須有兩個可用 AWS 區域。您可以使用`us-east-1`作為主要區域，並用`us-west-2`作次要區域。
- 適用 ElastiCache 於 Redis 的 Amazon — ElastiCache 全域資料存放區必須有兩個 AWS 區域可用。您可以`use us-east-1`作為主要區域，並`us-west-2`作為次要區域使用。

Amazon RDS 限制

- DR Orchestrator 架構在執行容錯移轉或容錯回復之前，不會檢查複寫延遲。必須手動檢查複寫延遲。
- 此解決方案已使用主要資料庫執行個體與一個僅供讀取複本進行測試。如果您想要使用一個以上的僅供讀取複本，請先徹底測試解決方案，然後再在生產環境中實作。

Aurora 限制

- 功能可用性和支援會因每個資料庫引擎的特定版本而有所不同 AWS 區域。如需跨區域複寫功能和區域可用性的詳細資訊，請參閱跨區域僅供讀取複本。
- Aurora 全域資料庫對支援的 Aurora 資料庫執行個體類別有特定組態需求，以及最大數目 AWS 區域。如需詳細資訊，請參閱 Amazon Aurora 全球資料庫的組態需求。
- 此解決方案已使用主要資料庫執行個體與一個僅供讀取複本進行測試。如果您想要使用一個以上的僅供讀取複本，請先徹底測試解決方案，然後再在生產環境中實作。

ElastiCache 限制

- 如需有關全域資料存放區的區域可用性和 ElastiCache 組態需求的資訊，請參閱文件中的必要條 ElastiCache 件和限制。

Amazon RDS p 接頭版本

Amazon RDS 支援下列引擎版本：

- MySQL — Amazon RDS 支援執行下列版本的資料庫執行個體：MySQL 8.0 和 MySQL 5.7
- PostgreSQL — 如需支援的 Amazon RDS for PostgreSQL 版本的相關資訊，請參閱可用的資料庫版本。
- MariaDB — Amazon RDS 支援執行下列 MariaDB 版本的資料庫執行個體：
  - MariaDB 10.11

- MariaDB 10.6
- MariaDB 10.5

Aurora 產品版本

- Amazon Aurora 全球資料庫轉換需要 Aurora 與 MySQL 5.7 相容性與 MySQL 5.7 相容性、2.09.1 及更高版本

  如需詳細資訊，請參閱 Amazon Aurora 全域資料庫的限制。

ElastiCache 適用於產品版本

Amazon ElastiCache 適用於 Redis 的支持以下版本：

- Redis 7.1 (增強版)
- Redis 7.0 (增強版)
- Redis 6.2 (增強版)
- Redis 6.0 (增強版)
- Redis 5.0.6 (增強版)

如需詳細資訊，請參閱 Redis 版本支援 ElastiCache 。

架構

Amazon RDS 架構

Amazon RDS 架構包括以下資源：

- 在主要區域 (`us-east-1`) 中建立的主要 Amazon RDS 資料庫執行個體，具有用戶端讀取/寫入存取權
- 在次要區域 (`us-west-2`) 中建立的 Amazon RDS 僅供讀取複本，具有用戶端唯讀存取權
- 主要和次要區域中部署的 DR 協調器架構

上圖顯示以下項目：

1. 主要執行個體與次要執行個體之間的非同步複製
2. 主要區域中用戶端的讀取/寫入存取
3. 次要區域中用戶端的唯讀存取權

Aurora 建築

Amazon Aurora 架構包括下列資源：

- 在主要區域 (us-east-1) 中使用主動寫入器端點建立的主要 Aurora 資料庫叢集
- 在次要區域 (us-west-2) 中使用非作用中寫入器端點建立的 Aurora 資料庫叢集
- 主要和次要區域中部署的 DR 協調器架構

上圖顯示以下項目：

1. 主要叢集與次要叢集之間的非同步複製
2. 具有主動寫入器端點的主要資料庫叢集
3. 具有非作用中寫入器端點的次要資料庫叢集

ElastiCache 對於雷迪斯架構

ElastiCache 適用於 Redis 的 Amazon 架構包括以下資源：

- 針對使用兩個叢集建立 ElastiCache 的 Redis 全域資料存放區：
  1. 主要區域中的主要叢集 (us-east-1)
  2. 次要區域中的次要叢集 (us-west-2)
- 在兩個叢集之間使用 TLS 1.2 加密的 Amazon 跨區域連結
- 主要和次要區域中部署的 DR 協調器架構

自動化和規模

DR Orchestrator 架構具備擴充能力，並支援 parallel 多個 AWS 資料庫的容錯移轉或容錯回復。

您可以使用下列承載程式碼容錯移轉帳戶中的多個 AWS 資料庫。在此範例中，三個 AWS 資料庫 (兩個全域資料庫，例如：與 Aurora MySQL 相容或與 Aurora PostgreSQL 相容，以及一個 Amazon RDS for MySQL 執行個體) 容錯移轉至 DR 區域：

```
{
  "StatePayload": [
    {
      "layer": 1,
      "resources": [
        {
          "resourceType": "PlannedFailoverAurora",
          "resourceName": "Switchover (planned failover) of Amazon Aurora global
 databases (MySQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-mysql-global-
identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-mysql-cluster-
identifier"
          }
        },
        {
          "resourceType": "PlannedFailoverAurora",
          "resourceName": "Switchover (planned failover) of Amazon Aurora global
 databases (PostgreSQL)",
          "parameters": {
            "GlobalClusterIdentifier": "!Import dr-globaldb-cluster-postgres-global-
identifier",
            "DBClusterIdentifier": "!Import dr-globaldb-cluster-postgres-cluster-
identifier"
          }
        },
        {
          "resourceType": "PromoteRDSReadReplica",
          "resourceName": "Promote RDS for MySQL Read Replica",
          "parameters": {
            "RDSInstanceIdentifier": "!Import rds-mysql-instance-identifier",
            "TargetClusterIdentifier": "!Import rds-mysql-instance-global-arn"
          }
        }
      ]
    }
  ]
}
```

工具

AWS 服務

- Amazon Aurora 是全受管的關聯式資料庫引擎，專為雲端建置，並與 MySQL 和 PostgreSQL 相容。
- Amazon 可 ElastiCache協助您在中設定、管理和擴展分散式記憶體內快取環境。 AWS 雲端這種模式使用 Amazon ElastiCache 的 Redis。
- AWS Lambda 是一項運算服務，可協助您執行程式碼，無需佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。在這種模式中，Lambda 函數用 AWS Step Functions 於執行步驟。
- Amazon Relational Database Service (Amazon RDS) 可協助您在中設定、操作和擴展關聯式資料庫 AWS 雲端。此模式支援 Amazon RDS for MySQL (適用於 MySQL)、亞馬遜 RDS 以及適用於瑪利亞德的亞馬遜 RDS。
- AWS SDK for Python (Boto3)可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務. 在這種模式中，Boto3 API 用於與數據庫實例或全局數據庫進行通信。
- AWS Step Functions是一項無伺服器協調服務，可協助您結合 AWS Lambda 功能與其他功能，AWS 服務 以建置關鍵業務應用程式。在此模式中，Step Functions 狀態機器用於協調和執行資料庫執行個體或全域資料庫的跨區域容錯移轉和容錯回復。

代碼存儲庫

此模式的代碼可在上 GitHub的 aws-cross-region-dr-data 存儲庫中找到。

史诗

安裝 DR 協調器框架

任務	描述	所需技能
克隆存 GitHub 儲庫。	若要複製存放庫，請執行下列命令： `git clone https://github.com/aws-samples/aws-cross-region-dr-databases.git`	AWS DevOps，AWS 管理員

任務	描述	所需技能
將 Lambda 函數程式碼封 Package 在 .zip 檔案封存中。	建立 Lambda 函數的封存檔案，以包含 DR 協調器架構相依性： `cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts` `bash scripts/deploy-orchestrator-sh.sh`	AWS 管理員
建立 S3 儲存貯體。	S3 儲存貯體需要存放 DR 協調器架構以及您的最新組態。建立兩個 S3 儲存貯體，一個位於主要區域 (us-east-1)，另一個位於次要區域 (us-west-2)： • dr-orchestrator-xxxxxx-us-east-1 • dr-orchestrator-xxxxxx-us-west-2 以隨機值取代，以使值區名稱xxxxxx具唯一性。	AWS 管理員
建立子網路和安全性群組。	在主要區域 (us-east-1) 和次要區域 (us-west-2) 中，為 VPC 中的 Lambda 函數部署建立兩個子網路和一個安全群組： • subnet-XXXXXXX • subnet-YYYYYYY • sg-XXXXXXXXXXXX	AWS 管理員

任務	描述	所需技能
更新 DR 協調器參數檔案。	在<YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation 資料夾中，更新下列 DR 協調器參數檔： • Orchestrator-Deployer-parameters-us-east-1.json • Orchestrator-Deployer-parameters-us-west-2.json 使用下列參數值x，y取代資源名稱和： [{ "ParameterKey": "TemplateStoreS3BucketName", "ParameterValue": "dr-orchestrator-xxxxxx-us-east-1" }, { "ParameterKey": "TemplateVPCId", "ParameterValue": "vpc-xxxxxx" }, { "ParameterKey": "TemplateLambdaSubnetID1",	AWS 管理員

任務	描述	所需技能
	"Paramete rValue": "subnet-x xxxxx" }, { "ParameterKey": "TemplateLambdaSub netID2", "Paramete rValue": "subnet-y yyyyy" }, { "ParameterKey": "TemplateLambdaSec urityGroupID", "Paramete rValue": "sg-xxxxx xxxxx" }]	

任務	描述	所需技能
將 DR 協調器架構程式碼上傳至 S3 儲存貯體。	S3 存儲桶中的代碼將比在本地目錄中更安全。將目DR-Orchestration-artifacts 錄 (包括所有檔案和子資料夾) 上傳到 S3 儲存貯體。 若要上傳程式碼，請執行下列動作： 1. 登入 AWS Management Console。 2. 導覽至 Amazon S3 主控台。 3. 選取dr-orchestrator-xxxxxx-us-east-1 bucket。 4. 選擇 [上傳]，然後選擇 [新增資料夾]。 5. 選取 DR-Orchestration-artifacts 資料夾。 6. 選擇上傳。 7. 選取dr-orchestrator-xxxxxx-us-west-2 值區。 8. 重複步驟 4-7。	AWS 管理員

任務	描述	所需技能
在主要區域中部署 DR 協調器架構。	若要在主要區域 (us-east-1) 中部署 DR 協調器架構，請執行下列命令： `cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation` `aws cloudformation deploy \` `--region us-east-1 \` `--stack-name dr-orchestrator \` `--template-file Orchestrator-Deployer.yaml \` `--parameter-overrides file://Orchestrator-Deployer-parameters-us-east-1.json \` `--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \` `--disable-rollback`	AWS 管理員

任務	描述	所需技能
在次要區域中部署 DR 協調器架構。	在次要區域 (us-west-2) 中，執行下列命令： `cd <YOUR-LOCAL-GIT-FOLDER>/DR-Orchestration-artifacts/cloudformation` `aws cloudformation deploy \` `--region us-west-2 \` `--stack-name dr-orchestrator \` `--template-file Orchestrator-Deployer.yaml \` `--parameter-overrides file://Orchestrator-Deployer-parameters-us-west-2.json \` `--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \` `--disable-rollback`	AWS 管理員

任務	描述	所需技能
驗證部署。	如果命 AWS CloudFormation 令運行成功，它返回以下輸出： `Successfully created/ updated stack - dr-orchestrator` 或者，您可以瀏覽至主 AWS CloudFormation 控台並驗證`dr-orchestrator` 堆疊的狀態。	AWS 管理員

建立資料庫執行個體或叢集

任務	描述	所需技能
建立資料庫子網路和安全群組。	在您的 VPC 中，為主要 (`us-east-1`) 和次要 () 區域的資料庫執行個體或全域資料庫建立兩個子網路和一個安全群組：`us-west-2` • `subnet-XXXXXX` • `subnet-XXXXXX` • `sg-XXXXXXXXXX`	AWS 管理員
更新主要資料庫執行個體或叢集的參數檔案。	在`<YOUR LOCAL GIT FOLDER>/App-Stack` 資料夾中，更新主要「區域」的參數檔案。 Amazon RDS	AWS 管理員

任務	描述	所需技能
	在RDS-MySQL-parameter-us-east-1.json 檔案中，更新SubnetIds 並DBSecurityGroup 使用您建立的資源名稱： `{ "Parameters": { "SubnetIds": "subnet-xxxxxx,subnet-xxxxxx", "DBSecurityGroup": "sg-xxxxxxxxxx", "MySqlGlobalIdentifier":"rds-mysql-instance", "InitialDatabaseName": "mysqldb", "DBPortNumber": "3789", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2", "KMSKeyAliasName": "rds/rds-mysql-instance-KmsKeyId" } }`  Amazon Aurora  在Aurora-MySQL-parameter-us-east-1.json 檔案中，更新SubnetIds 並DBSecurityGroup 使用您建立的資源名稱：  `{`	

任務	描述	所需技能
	<pre> "Parameters": { "SubnetIds": "subnet1-xxxxxx,su bnet2-xxxxxx", "DBSecurityGroup": "sg-xxxxxxxxxx", "GlobalClusterIden tifier":"dr-globaldb- cluster-mysql", "DBClusterName":"d bcluster-01", "SourceDBClusterNa me":"dbcluster-02", "DBPortNumber": "3787", "DBInstanceClass": "db.r5.large", "InitialDatabaseNa me": "sampledb", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2", "KMSKeyAliasName": "rds/dr-globaldb-c luster-mysql-KmsKe yId" } }</pre> Amazon ElastiCache 的雷迪斯 在ElastiCache-parame ter-us-east-1.json 檔案中，更新SubnetIds 並DBSecurityGroup 使用 您建立的資源名稱。 <pre>{ "Parameters": {</pre>	

任務	描述	所需技能
	"CacheNodeType": "cache.m5.large", "DBSecurityGroup": "sg-xxxxxxxxxx", "SubnetIds": "subnet-xxxxxx,subnet-xxxxxx", "EngineVersion": "5.0.6", "GlobalReplicationGroupIdSuffix": "demo-redis-global-datastore", "NumReplicas": "1", "NumShards": "1", "ReplicationGroupId": "demo-redis-cluster", "DBPortNumber": "3788", "TransitEncryption": "true", "KMSKeyAliasName": "elasticache/demo-redis-global-datastore-KmsKeyId", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2" } }	

任務	描述	所需技能
在主要區域部署資料庫執行個體或叢集。	若要在主要 Region (us-east-1) 中部署執行個體或叢集，請根據資料庫引擎執行下列命令。 Amazon RDS `cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack` `aws cloudformation deploy \` `--region us-east-1 \` `--stack-name rds-mysql-app-stack \` `--template-file RDS-MySQL-Primary.yaml \` `--parameter-overrides file://RDS-MySQL-parameter-us-east-1.json \` `--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \` `--disable-rollback` Amazon Aurora `cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack` `aws cloudformation deploy \` `--region us-east-1 \` `--stack-name aurora-mysql-app-stack \` `--template-file Aurora-MySQL-Primary.yaml \`	AWS 管理員

任務	描述	所需技能
	--parameter-overrides file://Aurora-MySQ L-parameter-us-eas t-1.json \ --capabilities CAPABILITY_AUTO_EX PAND CAPABILIT Y_NAMED_IAM CAPABILIT Y_IAM \ --disable-rollback Amazon ElastiCache 的雷迪斯 cd <YOUR-LOCAL-GIT-FO LDER>/App-Stack aws cloudformation deploy \ --region us-east-1 -- stack-name elasticac he-ds-app-stack \ --template-file ElastiCache-Primar y.yaml \ --parameter-overrides file://ElastiCache -parameter-us-east -1.json \ --capabilities CAPABILITY_AUTO_EX PAND CAPABILIT Y_NAMED_IAM CAPABILIT Y_IAM \ --disable-rollback 確認已成功部署 AWS CloudFormation 資源。	

任務	描述	所需技能
更新次要資料庫執行個體或叢集的參數檔案。	在資<YOUR LOCAL GIT FOLDER>/App-Stack 料夾中，更新次要區域的參數檔案。 Amazon RDS 在RDS-MySQL-parameter-us-west-2.json 檔案中，更新SubnetIDs 並DBSecurityGroup 使用您建立的資源名稱。PrimaryRegionKMSKeyArn 使用從主資料庫執行個體 AWS CloudFormation 堆疊的「輸出」區段MySQLKmsKeyId 取得的值來更新： `{ "Parameters": { "SubnetIds": "subnet-aaaaaaaaa, subnet-bbbbbbbbb", "DBSecurityGroup": "sg-ccccccccc", "MySqlGlobalIdentifier":"rds-mysql-instance", "InitialDatabaseName": "mysqldb", "DBPortNumber": "3789", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2",`	AWS 管理員

任務	描述	所需技能
	<pre>"KMSKeyAliasName": "rds/rds-mysql-ins tance-KmsKeyId", "PrimaryRegionKMSK eyArn":"arn:aws:km s:us-east-1:xxxxxx xxx:key/mrk-xxxxxx xxxxxxxxxxxxxx" } }</pre> Amazon Aurora 在Aurora-MySQL-param eter-us-west-2.jso n 檔案中，DBSecurit yGroup 使用您建立的資 源名稱更新SubnetIDs 和。PrimaryRe gionKMSKeyArn 使用 從主資料庫執行個體 AWS CloudFormation 堆疊的 「輸出」區段AuroraKms KeyId 取得的值來更新： <pre>{ "Parameters": { "SubnetIds": "subnet1-aaaaaaaaa ,subnet2-bbbbbbbbb", "DBSecurityGroup": "sg-ccccccccc", "GlobalClusterIden tifier":"dr-globaldb- cluster-mysql", "DBClusterName":"d bcluster-01", "SourceDBClusterNa me":"dbcluster-02",</pre>	

任務	描述	所需技能
	<pre>"DBPortNumber": "3787", "DBInstanceClass": "db.r5.large", "InitialDatabaseName": "sampledb", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2", "KMSKeyAliasName": "rds/dr-globaldb-cluster-mysql-KmsKeyId" } }</pre> Amazon ElastiCache 的雷迪斯 在ElastiCache-parameter-us-west-2.json 檔案中，更新SubnetIDs 並DBSecurityGroup 使用您建立的資源名稱。PrimaryRegionKMSKeyArn 使用從主資料庫執行個體 AWS CloudFormation 堆疊的「輸出」區段ElastiCacheKmsKeyId 取得的值來更新： <pre>{ "Parameters": { "CacheNodeType": "cache.m5.large", "DBSecurityGroup": "sg-ccccccccc",</pre>	

任務	描述	所需技能
	"SubnetIds": "subnet-aaaaaaaaa, subnet-bbbbbbbbb", "EngineVersion": "5.0.6", "GlobalReplicationGroupIdSuffix": "demo-redis-global-datastore", "NumReplicas": "1", "NumShards": "1", "ReplicationGroupId": "demo-redis-cluster", "DBPortNumber": "3788", "TransitEncryption": "true", "KMSKeyAliasName": "elasticache/demo-redis-global-datastore-KmsKeyId", "PrimaryRegion": "us-east-1", "SecondaryRegion": "us-west-2" } }	

任務	描述	所需技能
在次要區域部署資料庫執行個體或叢集。	根據您的資料庫引擎執行下列命令。 Amazon RDS `cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack` `aws cloudformation deploy \` `--region us-west-2 \` `--stack-name rds-mysql-app-stack \` `--template-file RDS-MySQL-DR.yaml \` `--parameter-overrides file://RDS-MySQL-parameter-us-west-2.json \` `--capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_NAMED_IAM CAPABILITY_IAM \` `--disable-rollback` Amazon Aurora `cd <YOUR-LOCAL-GIT-FOLDER>/App-Stack` `aws cloudformation deploy \` `--region us-west-2 \` `--stack-name aurora-mysql-app-stack \` `--template-file Aurora-MySQL-DR.yaml \` `--parameter-overrides file://Aurora-MySQ`	AWS 管理員

任務	描述	所需技能
	L-parameter-us-wes t-2.json \ --capabilities CAPABILITY_AUTO_EX PAND CAPABILIT Y_NAMED_IAM CAPABILIT Y_IAM \ --disable-rollback	
	Amazon ElastiCache 的雷迪斯 cd <YOUR-LOCAL-GIT-FO LDER>/App-Stack aws cloudformation deploy \ --region us-west-2 \ --stack-name elasticac he-ds-app-stack \ --template-file ElastiCache-DR.yaml \ --parameter-overrides file://ElastiCache -parameter-us-west -2.json \ --capabilities CAPABILITY_AUTO_EX PAND CAPABILIT Y_NAMED_IAM CAPABILIT Y_IAM \ --disable-rollback	
	確認已成功部署 AWS CloudFormation 資源。	

相關資源

- 資料庫的災難復原策略 AWS (AWS 規範性指導策略)

- 將關聯式資料庫的 DR 解決方案自動化 AWS(AWS 規範指南指南)
- 使用 Amazon Aurora 全球數據
- AWS 區域 使用全域資料存放區間的複寫
- 將關聯式資料庫的 DR 解決方案自動化 AWS(AWS 規範指南指南)

自動化跨 AWS 帳戶複寫 Amazon RDS 執行個體

創建者：帕拉格納格韋卡 (AWS) 和阿倫·錢達皮萊 (AWS)

環境：生產	技術：資料庫；無伺服器 DevOps；基礎架構	工作負載：所有其他工作
AWS 服務：AWS Lambda；Amazon RDS；適用於 Python 的 AWS 開發套件（博多 3）；AWS Step Functions；Amazon SNS		

Summary

此模式說明如何使用 AWS AWS Step Functions 數和 AWS Lambda，將 Amazon 關聯式資料庫服務 (Amazon RDS) 資料庫執行個體跨不同 AWS 帳戶複寫、追蹤和復原的程序自動化。您可以使用此自動化功能來執行 RDS 資料庫執行個體的大規模複寫，而不會對效能造成任何影響或營運超負荷 — 無論您的組織規模如何。您也可以使用此模式來協助組織遵守強制性的資料管理策略或合規要求，這些策略要求在不同 AWS 帳戶和 AWS 區域之間複寫和備援您的資料。大規模跨帳戶複寫 Amazon RDS 資料是效率低下且容易出錯的手動程序，成本高昂且耗時，但此模式的自動化可協助您安全、有效且有效率地實現跨帳戶複寫。

先決條件和限制

先決條件

- 兩個 AWS 帳戶
- 在來源 AWS 帳戶中啟動並執行的 RDS 資料庫執行個體
- 目的地 AWS 帳戶中 RDS 資料庫執行個體的子網路群組
- 在來源 AWS 帳戶中建立並與目的地帳戶共用的 AWS Key Management Service (AWS KMS) 金鑰 (如需有關政策詳細資訊的詳細資訊，請參閱此模式的其他資訊一節)。
- 目的地 AWS 帳戶中的 AWS KMS 金鑰，用於加密目的地帳戶中的資料庫

產品版本

- Python 3.9（使用 AWS Lambda）
- PostgreSQL、13 倍和 14 倍

架構

技術, 堆棧

- Amazon Relational Database Service (Amazon RDS)
- Amazon Simple Notification Service (Amazon SNS)
- AWS Key Management Service (AWS KMS)
- AWS Lambda
- AWS Secrets Manager
- AWS Step Functions

目標架構

下圖顯示使用 Step Functions 協調 RDS 資料庫執行個體從來源帳戶 (帳戶 A) 到目的地帳戶 (帳戶 B) 的排程隨選複寫的架構。

在來源帳戶 (圖表中的帳戶 A) 中，「Step Functions」狀態機器會執行下列動作：

1. 從帳戶 A 中的 RDS 資料庫執行個體建立快照。
2. 使用帳戶 A 中的 AWS KMS 金鑰複製和加密快照，為確保傳輸過程中的加密，無論資料庫執行個體是否加密，都會加密快照。
3. 將快照存取帳戶 B 授與帳戶 B 共用資料庫快照。
4. 將通知推送至 SNS 主題，然後 SNS 主題叫用帳戶 B 中的 Lambda 函數。

在目標帳戶 (圖表中的帳戶 B) 中，Lambda 函數會執行 Step Functions 數狀態機器來協調下列項目：

1. 將共用快照從帳戶 A 複製到帳戶 B，同時先使用帳戶 A 中的 AWS KMS 金鑰解密資料，然後使用帳戶 B 中的 AWS KMS 金鑰加密資料。
2. 從 Secret 管理員讀取密碼，以擷取目前資料庫執行個體的名稱。
3. 使用適用於 Amazon RDS 的新名稱和預設 AWS KMS 金鑰，從快照還原資料庫執行個體。

4. 讀取新資料庫的端點，並使用新的資料庫端點更新 Secrets Manager 中的密碼，然後標記先前的資料庫執行個體，以便稍後將其刪除。
5. 保留資料庫的最新 N 個執行個體，並刪除所有其他執行個體。

工具

AWS 工具

- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- 適用於 Python 的 AWS 開發套件 (Boto3) 是一套軟體開發套件，可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 Lambda 函數和其他 AWS 服務，以建立關鍵業務應用程式。

Code

此模式的程式碼可在 GitHub 跨帳戶 RDS 複寫存放庫中取得。

史诗

只要按一下，即可在 AWS 帳戶自動複寫 RDS 資料庫執行個體

任務	描述	所需技能
在來源帳戶中部署 CloudFormation 堆疊。	1. 登入來源帳戶 (帳戶 A) 的 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 在導覽窗格中，選擇 Stacks (堆疊)。 3. 選擇 [建立堆疊]，然後選擇 [使用現有資源 (匯入資源)]。 4. 在 [識別資源] 頁面上，選擇 [下一步]。 5. 在 [指定範本] 頁面上，選取 [上傳範本]。 6. 選擇 [選擇檔案]，從 [GitHub 跨帳戶 RDS 複寫] 儲存庫中選取`Cloudformation-SourceAccountRDS.yaml` 檔案，然後選擇 [下一步]。 7. 在堆疊名稱中，輸入堆疊的名稱。 8. 在「參數」段落中，指定堆疊樣板中定義的參數： • 在中 DestinationAccount Number，輸入目的地 RDS 資料庫執行個體的帳戶號碼。 • 在中 KeyName，輸入您的 AWS KMS 金鑰。	雲端管理員、雲端架構師

任務	描述	所需技能
	• 對於 ScheduleExpression，輸入 Cron 表示式 (預設為每日上午 12:00)。 • 在來源標識符中，輸入來源資料庫的名稱。 • 對於 SourceDB SnapshotName，請輸入快照的名稱或接受預設值。 9. 選擇下一步。 10.在 [設定堆疊選項] 頁面上，保留預設值，然後選擇 [下一步]。 11.檢閱您的堆疊組態，然後選擇 [提交]。 12.選擇堆疊的 [資源] 索引標籤，然後記下 SNS 主題的 Amazon 資源名稱 (ARN)。	

任務	描述	所需技能
在目的地帳戶中部署 CloudFormation 堆疊。	1. 登入目的地帳戶 (帳戶 B) 的 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 在導覽窗格中，選擇 Stacks (堆疊)。 3. 選擇 [建立堆疊]，然後選擇 [使用現有資源 (匯入資源)]。 4. 在 [識別資源] 頁面上，選擇 [下一步]。 5. 在 [指定範本] 頁面上，選取 [上傳範本]。 6. 選擇檔案，從 GitHub 跨帳戶 RDS 複寫儲存庫中選取`Cloudformation-DestinationAccountRDS.yaml` 檔案，然後選擇下一步。 7. 在堆疊名稱中，輸入堆疊的名稱。 8. 在「參數」段落中，指定堆疊樣板中定義的參數： • 在中 DatabaseName，輸入資料庫的名稱。 • 在 Engine 中，輸入與來源資料庫相符的資料庫引擎類型。 • 對於 DB InstanceClass，請輸入偏好的資料庫執行處理類型，或接受預設值。 • 針對子網路群組，輸入現有的 VPC 子網路群組。	雲端架構師、 DevOps 工程師、雲端管理員

任務	描述	所需技能
	如需有關建立子網路群組的指示，請參閱 Amazon RDS 使用者指南中的步驟 2：建立資料庫子網路群組。 • 在中 SecretName，輸入路徑和密碼名稱，或接受預設名稱。 • 對於 SGID，請輸入目的地叢集的安全性群組識別碼。 • 若為 KMSKey，請在目的地帳戶中輸入 KMS 金鑰的 ARN。 • 在中 NoOfOlderInstances，輸入要保留用於復原的 RDS 資料庫執行個體舊副本數目。 9. 選擇下一步。 10. 在 [設定堆疊選項] 頁面上，保留預設值，然後選擇 [下一步]。 11. 檢閱您的堆疊組態，然後選擇 [提交]。 12. 選擇堆疊的 `InvokeStepFunction` [資源] 索引標籤，然後記下的 [實體 ID] 和 [ARN]。	

任務	描述	所需技能
確認目的地帳戶中 RDS 資料庫執行個體的建立。	1. 登入 AWS 管理主控台並開啟 Amazon RDS 主控台。 2. 在瀏覽窗格中，選擇 [資料庫]，然後確認新 RDS 資料庫執行個體出現在新叢集下方。	雲端管理員、雲端架構師、DevOps 工程師

任務	描述	所需技能
將 Lambda 函數訂閱至 SNS 主題。	您必須執行下列 AWS Command Line Interface (AWS CLI) (AWS CLI) 命令，才能將目標帳戶 (帳戶 B) 中的 Lambda 函數訂閱到來源帳戶 (帳戶 A) 中的 SNS 主題。 在帳戶 A 中，執行下列命令： `aws sns add-permission \ --label lambda-access --aws-account-id <DestinationAccount> \ --topic-arn <Arn of SNSTopic > \ --action-name Subscribe ListSubscriptionsB yTopic` 在帳戶 B 中，執行下列命令： `aws lambda add-permi ssion \ --function-name <Name of InvokeStepFunction > \ --source-arn <Arn of SNSTopic > \ --statement-id function-with-sns \ --action lambda:In vokeFunction \ --principal sns.amazo naws.com` 在帳戶 B 中，執行下列命令：	雲端管理員、雲端架構師、DBA

任務	描述	所需技能
	`aws sns subscribe \ --protocol "lambda" \ --topic-arn <Arn of SNSTopic> \ --notification-endpoint <Arn of InvokeStepFunction>`	
將來源帳戶中的 RDS 資料庫執行個體與目標帳戶同步。	啟動來源帳戶中的「Step Functions 數」狀態機器，以起始隨選資料庫複寫。 1. 開啟 Step Functions 主控台。 2. 在瀏覽窗格中，選擇 [狀態機器]。 3. 選擇您的狀態機。 4. 在 [執行] 索引標籤上，選取您的函數，然後選擇 [開始執行] 以啟動工作流程。 備註：排程器可協助您按排程自動執行複製，但排程器預設為關閉。您可以在目標帳戶 CloudFormation 堆疊的 [資源] 索引標籤中找到排程器的 Amazon CloudWatch 規則名稱。如需有關如何修改 CloudWatch 事件規則的指示，請參閱 CloudWatch 使用指南中的刪除或停用 CloudWatch 事件規則。	雲端架構師、 DevOps 工程師、雲端管理員

任務	描述	所需技能
視需要將資料庫復原至任何先前的副本。	1. 開啟 Secrets Manager 主控台。 2. 從密碼清單中，選擇您先前使用 CloudFormation 範本建立的密碼。您的應用程式會使用密碼來存取目的地叢集中的資料庫。 3. 若要從詳細資訊頁面更新密碼值，請在「機密值」段落中選擇擷取秘密值，然後選擇編輯。 4. 輸入資料庫端點的詳細資訊。	雲端管理員、DBA、工程師 DevOps

相關資源

- 跨區域僅供讀取複本 (Amazon RDS 使用者指南)
- 藍/綠部署 (Amazon RDS 使用者指南)

其他資訊

您可以使用下列範例政策在 AWS 帳戶之間共用 AWS KMS 金鑰。

```
{
    "Version": "2012-10-17",
    "Id": "cross-account-rds-kms-key",
    "Statement": [
        {
            "Sid": "Enable user permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<SourceAccount>:root"
            },
            "Action": "kms:*",
            "Resource": "*"
```

```
        },
        {
            "Sid": "Allow administration of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<DestinationAccount>:root"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::<DestinationAccount>:root",
                    "arn:aws:iam::<SourceAccount>:root"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

使用 Systems Manager 和自動備份 SAP HANA 資料庫 EventBridge

由安巴里薩塔卡（AWS）和高拉夫拉特（AWS）創建

程式碼儲存庫:備份文件	環境：生產	技術：資料庫、儲存與備份
工作負載：SAP	AWS 服務：Amazon EC2; Amazon EventBridge; Amazon S3; AWS Systems Manager	

Summary

此模式描述如何使用 AWS Systems Manager、Amazon、亞馬遜簡單儲存服務 (Amazon EventBridge S3) 和適用於 SAP HANA 的 AWS Backint Agent 理自動化 SAP HANA 資料庫備份。

此模式使用命令提供 shell `BACKUP DATA` 指令碼型方法，並且不需要為多個系統上的每個作業系統 (OS) 執行個體維護指令碼和工作組態。

注意：截至 2023 年 4 月，AWS Backup 宣布支援 Amazon 彈性運算雲端 (亞馬遜 EC2) 上的 SAP HANA 資料庫。如需詳細資訊，請參閱 Amazon EC2 執行個體上的 SAP HANA 資料庫備份。

根據組織的需求，您可以使用 AWS Backup 服務自動備份 SAP HANA 資料庫，也可以使用此模式。

先決條件和限制

先決條件

- 在針對 Systems Manager 設定的受管 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上，具有受支援版本處於執行狀態的現有 SAP HANA 執行個體
- 系統管理員代理程式 (SSM 代理程式) 2.3.274.0 或更新版本已安裝
- 未啟用公開存取權的 S3 儲存貯體
- 一個名為的`hdbuserstore`密鑰 `SYSTEM`
- 自動化執行手冊的 AWS Identity and Access Management (IAM) 角色，可按排程執行

- AmazonSSMManagedInstanceCore和ssm:StartAutomationExecution原則會附加至 Systems Manager 自動化服務角色。

限制

- 適用於 SAP HANA 的 AWS Backint Agent 不支援重複資料刪除功能。
- 適用於 SAP HANA 的 AWS Backint Agent 不支援資料壓縮。

產品版本

下列作業系統支援 AWS Backint Agent：

- SUSE Linux Enterprise Server
- 適用於 SAP 的 SUSE 企業伺服器
- 適用於 SAP 的紅帽企業

AWS Backint Agent 支援下列資料庫：

- 單節點和多節點
- SAP HANA 2.0 及更新版本 (單節點和多個節點)

架構

目標技術堆疊

- AWS 巴肯特代理程式
- Amazon S3
- AWS Systems Manager
- Amazon EventBridge
- SAP HANA

目標架構

下圖顯示安裝 AWS Backint Agent、S3 儲存貯體和 Systems Manager 的安裝指令碼，以及使用 Command 文件來排程定期備份的安裝指令碼。 EventBridge

自動化與規模

- 您可以使用系統管理員自動化執行手冊來安裝多個 AWS Backint 代理程式。
- Systems Manager 工作手冊的每次執行都可以根據目標選擇擴展到 n 個 SAP HANA 執行個體。
- EventBridge 可以自動化 SAP HANA 備份。

工具

- 適用於 SAP HANA 的 AWS Backint Agent 程式是與現有工作流程整合的獨立應用程式，可將 SAP HANA 資料庫備份到您在組態檔中指定的 S3 儲存貯體。AWS Backint Agent 支援 SAP HANA 資料庫的完整、增量和差異備份。它在 SAP HANA 資料庫伺服器上執行，其中備份和目錄會從 SAP HANA 資料庫傳輸到 AWS Backint Agent。
- Amazon EventBridge 是一種無伺服器事件匯流排服務，您可以使用它將應用程式與來自各種來源的資料連接起來。 EventBridge 將來自應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務的即時資料串流傳遞至 AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點或其他帳戶中的事件匯流排等目標。
- Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- AWS Systems Manager 可協助您在 AWS 上檢視和控制基礎設施。您可以使用 Systems Manager 主控台檢視來自多個 AWS 服務的操作資料，並自動化 AWS 資源的操作任務。

Code

此模式的代碼可在aws-backint-automated-backup GitHub 存儲庫中找到。

史诗

創建一個高清商店密鑰系統

任務	描述	所需技能
創建一個高清客存儲密鑰。	1. 導覽至 `/usr/sap/<SID>/HDB<InstNo>/exe`。	AWS 管理員、SAP HANA 管理員

任務	描述	所需技能
	2. 以 SAP HANA 資料庫執行個體編號的XX形式執行下列命令。 `hdbuserstore -i set SYSTEM <hostname>:3XX13@SYSTEMDB SYSTEM` 例如，對於saphanadb 具有執行個體編號的 SAP HANA 主機00，請執行下列命令。 `hdbuserstore -i set SYSTEM saphanadb:30013@SYSTEMDB SYSTEM`	

安裝 AWS 黑 Backint Agent

任務	描述	所需技能
安裝 AWS 黑 Backint Agent。	遵循 AWS Backint 代理程式文件中針對 SAP HANA 安裝和設定 AWS BBackint Agent 理程式中的說明進行操作。	AWS 管理員、SAP HANA 管理員

建立系 Systems Manager 指令文件

任務	描述	所需技能
建立系 Systems Manager 指令文件。	1. 登入 AWS 管理主控台並開啟 AWS Systems Manager 主控台。 2. 選擇「文件」，然後選擇「我擁有」。 3. 確認您與 SAP HANA 資料庫位於相同的 AWS 區域。 4. 選擇「建立文件」、「指令」或「工作階段」來建立文件 5. 使用唯一且具描述性的名稱，不含空格 (例如 SAP Hana Backup)。 6. 請確定 [文件類型] 設定為 [命令] 文件。 7. 在內容標題下，有一些示例代碼。確保您選擇 JSON 代碼類型，並用GitHub 存儲庫中HDB_Backup_SSM_Document.json 文件中的代碼替換代碼。 8. 選擇 Create document (建立文件)。 9. 在「我擁有」區段中檢查您的文件。	AWS 管理員、SAP HANA 管理員

按常規頻率排程備份

任務	描述	所需技能
使用 Amazon 安排定期備份 EventBridge。	1. 開啟 Amazon 主 EventBridge控台，選擇「規則」，然後選擇「建立規則」。 2. 在「定義規則詳細資料」畫面上，輸入規則的唯一名稱和說明，並使用預設事件匯流排。 3. 在規則類型下，選擇排程，然後選擇下一步。 4. 在「定義排程」畫面上，根據所需的頻率選擇適當的排程模式和 Cron 或速率運算式。 5. 在「選取目標」 畫面上，選擇「AWS 服務」做為「目標類型」。在 [選取目標] 下，選擇 [Systems Manager 執行命令]。 6. 選擇您之前創建的文檔。 7. 在 [目標索引鍵和目標值] 下，提供執行個體識別碼。您可以使用標籤名稱和標籤值來新增多個例證。 8. 在 [設定自動化參數] 下，選擇 [常數] 做為增量或差異備份 如果要完整備份，請選擇「無參數」。 9. 選擇要建立新角色還是使用現有角色。如果您使用現有角色，請確定該角色具有呼叫目標所需的原則。	AWS 管理員、SAP HANA 管理員

任務	描述	所需技能
	10.保留預設的其他設定，然後選擇「下一步」。 11.「設定標記」畫面是選擇性的。選擇下一步。 12.在「檢閱並建立」畫面上，檢閱規則設定，然後選擇「建立」。應該已成功建立規則。 您可以從 S3 儲存貯體路徑驗證備份成功。 `s3:/<your_bucket_name>/<target folder>/<SID>/usr/sap/<SID>/SYS/global/hdb/backint/DB_<SID>/` 您也可以從 SAP HANA 備份目錄驗證備份。	

相關資源

- 適用於 SAP HANA 的 AWS Backint Agent
- 安裝和設定適用於 SAP HANA 的 AWS Backint Agent

使用雲端託管人封鎖對 Amazon RDS 的公開存取

由阿比·庫瑪（AWS）和德瓦里卡·帕特雷（AWS）創建

環境：生產	技術：資料庫；安全性、身分識別、合規性	工作負載：所有其他的工作負載；
AWS 服務：Amazon RDS		

Summary

許多組織在多個雲端供應商上執行其工作負載和服務。在這些混合雲環境中，除了個別雲端供應商所提供的安全性外，雲端基礎架構還需要嚴格的雲端控管。像 Amazon 關聯式資料庫服務 (Amazon RDS) 這樣的雲端資料庫是一項重要的服務，必須監控是否有任何存取和權限漏洞。雖然您可以透過設定安全群組來限制對 Amazon RDS 資料庫的存取，但是您可以新增第二層保護以禁止諸如公用存取之類的動作。確保公共訪問被阻止將為您提供一般數據保護條例（GDPR），Health 保險可移植性和責任法案（HIPAA），美國國家標準與技術研究所（NIST）和支付卡行業數據安全標準（PCI DSS）合規性。

雲端託管人是一種開放原始碼規則引擎，可用來強制執行 Amazon Web Services (AWS) 資源 (例如 Amazon RDS) 的存取限制。透過雲端託管人，您可以設定規則，以根據定義的安全性和合規性標準來驗證環境。您可以使用 Cloud Doctordian 管理雲端環境，協助確保符合安全性原則、標籤原則，以及未使用資源的垃圾回收和成本管理。透過雲端託管人，您可以使用單一介面在混合雲環境中實作控管。例如，您可以使用雲端託管界面與 AWS 和 Microsoft Azure 互動，從而減少使用 AWS Config、AWS 安全群組和 Azure 政策等機制的工作量。

此模式提供如何在 AWS 上使用雲端託管人強制限制 Amazon RDS 執行個體上公有可存取性的指示。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- key pair
- 已安裝 AWS Lambda

架構

目標技術堆疊

- Amazon RDS
- AWS CloudTrail
- AWS Lambda
- Cloud Custodian

目標架構

下圖顯示雲端託管人在 Amazon RDS 上將政策部署到 Lambda、AWS CloudTrail 啟動`CreateDBInstance`事件，並將 Lambda 函數設定`PubliclyAccessible`為 false。

工具

AWS 服務

- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。

其他工具

- Cloud Doctordian 將許多組織用來管理其公有雲帳戶的工具和指令碼統一為一個開放原始碼工具。它使用無狀態規則引擎進行策略定義和強制執行，其中包含雲端基礎架構的指標、結構化輸出和詳細報告。它與無伺服器執行階段緊密整合，以提供即時補救和回應，同時降低營運成本。

史诗

設定 AWS CLI

任務	描述	所需技能
安裝 AWS CLI。	若要安裝 AWS CLI，請遵循 AWS 文件中的指示進行。	AWS 管理員
設定 AWS 登入資料。	設定 AWS CLI 用來與 AWS 互動的設定，包括 AWS 區域和您要使用的輸出格式。 `$>aws configure` `AWS Access Key ID` `[None]: <your_acc` `ess_key_id>` `AWS Secret Access Key` `[None]: <your_sec` `ret_access_key>` `Default region name` `[None]:` `Default output format` `[None]:` 如需詳細資訊，請參閱 AWS 文件。	AWS 管理員
建立 IAM 角色。	若要使用 Lambda 執行角色建立 IAM 角色，請執行下列命令。 `aws iam create-role --` `role-name lambda-ex --` `assume-role-policy-` `document '{"Version":` `"2012-10-17","Stat` `ement": [{ "Effect":` `"Allow", "Principal":` `{"Service": "lambda.a` `mazonaws.com"},`	AWS DevOps

任務	描述	所需技能
	`"Action": "sts:AssumeRole"}]}`	

設定雲端託管人

任務	描述	所需技能
安裝雲託管。	若要為您的作業系統和環境安裝雲端託管人，請依照雲端託管人說明文件中的指示進行。	DevOps 工程師
檢查雲託管架構。	若要查看可針對其執行政策的 Amazon RDS 資源的完整清單，請使用下列命令。 `custodian schema aws.rds`	DevOps 工程師
建立雲端託管人政策。	使用 YAML 副檔名儲存在「其他資訊」區段中的「雲端託管人」原則檔案下的程式碼。	DevOps 工程師
定義雲端託管人動作以變更可公開存取的旗標。	1. 找出託管人代碼 (例如，`/Users/abcd/custodian/lib/python3.9/site-packages/c7n/resources/rds.py`)。 2. 尋找中的`RDSSetPublicAvailability`類別`rds.py`，然後使用其他資訊區段中 c7n 資源 rds.py 檔案下的程式碼來修改此類別。	DevOps 工程師

任務	描述	所需技能
執行乾運行。	(選擇性) 若要檢查原則識別哪些資源，而不對資源執行任何動作，請使用下列命令。 `custodian run -dryrun <policy_name>.yaml -s <output_directory>`	DevOps 工程師

部署原則

任務	描述	所需技能
使用 Lambda 部署政策。	若要建立將執行原則的 Lambda 函數，請使用下列命令。 `custodian run -s policy.yaml` 然後，AWS CloudTrail `CreateDBInstance` 活動將啟動此政策。 因此，對於符合條件的執行個體，AWS Lambda 會將可公開存取的旗標設定為 false。	DevOps 工程師

相關資源

- AWS Lambda
- Amazon RDS
- Cloud Custodian

其他資訊

雲端託管人原則 YAML 檔案

```
policies:
  - name: "block-public-access"
    resource: rds
    description: |
      This Enforcement blocks public access for RDS instances.
    mode:
      type: cloudtrail
      events:
        - event: CreateDBInstance # Create RDS instance cloudtrail event
          source: rds.amazonaws.com
          ids: requestParameters.dBInstanceIdentifier
      role: arn:aws:iam::1234567890:role/Custodian-compliance-role
    filters:
      - type: event
        key: 'detail.requestParameters.publiclyAccessible'
        value: true
    actions:
      - type: set-public-access
        state: false
```

C7 安全資源 rds.py 文件

```
@actions.register('set-public-access')
 class RDSSetPublicAvailability(BaseAction):

     schema = type_schema(
         "set-public-access",
         state={'type': 'boolean'})
     permissions = ('rds:ModifyDBInstance',)

     def set_accessibility(self, r):
         client = local_session(self.manager.session_factory).client('rds')
         waiter = client.get_waiter('db_instance_available')
         waiter.wait(DBInstanceIdentifier=r['DBInstanceIdentifier'])
         client.modify_db_instance(
             DBInstanceIdentifier=r['DBInstanceIdentifier'],
             PubliclyAccessible=self.data.get('state', False))
```

```
    def process(self, rds):
        with self.executor_factory(max_workers=2) as w:
            futures = {w.submit(self.set_accessibility, r): r for r in rds}
            for f in as_completed(futures):
                if f.exception():
                    self.log.error(
                        "Exception setting public access on %s  \n %s",
                        futures[f]['DBInstanceIdentifier'], f.exception())
        return rds
```

Security Hub 整合

雲端託管人可與 AWS Security Hub 整合，以傳送安全發現結果並嘗試修復動作。如需詳細資訊，請參閱宣布雲端託管人與 AWS Security Hub 整合。

在 AWS 上的 SQL Server 中的「永遠開啟」可用性群組中設定唯讀路由

創建者：蘇哈尼·謝克 (AWS)

環境：PoC 或試點	技術：資料庫；基礎架構	工作量：Microsoft
AWS 服務：AWS 管理 Microsoft AD；Amazon EC2		

Summary

此模式涵蓋如何在 SQL Server 永遠開啟中使用待命次要複本，方法是將唯讀工作負載從主要複本卸載到次要複本。

資料庫鏡像具有 one-to-one 對應。您無法直接讀取次要資料庫，因此您必須建立快照集。永遠開啟可用性群組功能是在 Microsoft SQL 伺服器 2012 年引入。在更高版本中，主要功能已引入，包括只讀路由。在永遠開啟可用性群組中，您可以將複本模式變更為唯讀，直接從次要複本讀取資料。

永遠開啟可用性群組解決方案支援高可用性 (HA)、災難復原 (DR)，以及資料庫鏡像的替代方案。「永遠開啟」可用性群組會在資料庫層級運作，並將一組使用者資料庫的可用性最大化。

SQL Server 會使用唯讀路由機制，將傳入的唯讀連線重新導向至次要僅供讀取複本。為此，您應該在連接字符串中添加以下參數和值：

- `ApplicationIntent=ReadOnly`
- `Initial Catalog=<database name>`

先決條件和限制

先決條件

- 具有虛擬私有雲 (VPC)、兩個可用區域、私有子網路和安全群組的有效 AWS 帳戶
- 兩台 Amazon Elastic Compute Cloud (Amazon EC2) 機器搭配 SQL Server 2019 企業版 Amazon 機器映像，並在執行個體層級設定 Windows 伺服器容錯移轉叢集 (WSFC)，以及在主節點 () 和次要節點 (WSFCNODE1) 之間設定的 SQL Server 層級的永遠開啟可用性群組，這些群組是 AWS

Directory Service 的一部分，適用於 Microsoft Active Directory 目錄目錄的 AWS 目錄服務的一部分
WSFCNODE2 tagechtalk.com

- 設定為在次要複本中接受read-only的一或多個節點
- SQLAG1為「永遠開啟」可用性群組命名的監聽器
- SQL Server 資料庫引擎在兩個節點上以相同的服務帳戶執行
- SQL 伺服器管理工作室 (SSMS)
- 名為的測試資料庫 test

產品版本

- SQL 伺服器 2014 年及更新版本

架構

目標技術堆疊

- Amazon EC2
- AWS 受管 Microsoft AD
- Amazon FSx

目標架構

下圖顯示「永遠開啟」可用性群組 (AG) 接聽程式如何將連線中包含ApplicationIntent參數的查詢重新導向至適當的次要節點。

1. 會將要求傳送至「永遠開啟」可用性群組接聽程式。
2. 如果連接字串沒有ApplicationIntent參數，則會將要求傳送至主要執行個體。
3. 如果連接字串包含ApplicationIntent=ReadOnly，則會使用唯讀路由組態將要求傳送至次要執行個體，而 WSFC 具有「永遠開啟」可用性群組。

工具

AWS 服務

- 適用於 Microsoft 活動目錄的 AWS Directory Service 可讓您的目錄感知工作負載和 AWS 資源在 AWS 雲端中使用 Microsoft 活動目錄。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon FSx 提供支援業界標準連線協定的檔案系統，並在 AWS 區域提供高可用性和複寫功能。

其他服務

- SQL 伺服器管理工作室 (SSMS) 是用於連接、管理和管理 SQL 伺服器執行個體的工具。
- sqlcmd 是一個命令行實用程序。

最佳實務

如需永遠開啟可用性群組的詳細資訊，請參閱 SQL Server 說明文件。

史诗

設定唯讀路由

任務	描述	所需技能
將複本更新為唯讀。	若要將主要複本和次要複本更新為唯讀，請從 SSMS 連線到主要複本，然後從 [其他資訊] 區段執行步驟 1 程式碼。	DBA
建立路由網址。	若要為兩個複本建立路由 URL，請從 [其他資訊] 區段執行步驟 2 程式碼。在此代碼中，`tagechtalk.com` 是 AWS 受管 Microsoft AD 目錄的名稱。	DBA
建立路由清單。	若要為兩個複本建立路由清單，請從 [其他資訊] 區段執行步驟 3 程式碼。	DBA

任務	描述	所需技能
驗證路由清單。	從 SQL Server 管理 Studio Connect 到主要執行個體，然後從 [其他資訊] 區段執行步驟 4 程式碼，以驗證路由清單。	DBA

測試唯讀路由

任務	描述	所需技能
使用 ApplicationIntent 參數進行 Connect。	1. 在 SSMS 中，使`ApplicationIntent=ReadOnly;Initial Catalog=test` 用連線至永遠開啟可用性群組接聽程式名稱。 2. 連接與次要複本建立。若要測試此問題，請執行下列命令以顯示連線的伺服器名稱。 `SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios')` 輸出將顯示目前的次要複本名稱 (`WSFCNODE2`)。	DBA
執行容錯移轉。	1. 從 SSMS，連線到永遠開啟可用性群組接聽程式名稱。 2. 確認主要和次要資料庫是否同步，且不會遺失任何資料。	DBA

任務	描述	所需技能
	3. 執行容錯移轉，以便目前的主要複本成為次要複本，而次要複本會成為主要複本。 4. 在 SSMS 中，使ApplicationIntent=ReadOnly;Initial Catalog=test 用連線至永遠開啟可用性群組接聽程式名稱。 5. 連接與次要複本建立。若要測試此問題，請執行下列命令來顯示連線的伺服器名稱。 `SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios')` 它將顯示當前的次要副本名稱（WSFCNODE1 ）。	

使 Connect sqlcmd 命令列公用程式進行連線

任務	描述	所需技能
Connect 用 SQLCMD 進行連線。	若要從 sqlcmd 連線，請在命令提示字元中從其他資訊區段執行步驟 5 程式碼。連線之後，執行下列命令以顯示連線的伺服器名稱。 `SELECT SERVERPROPERTY('ComputerNamePhysicalNetBios') .`	DBA

任務	描述	所需技能
	輸出將顯示目前的次要複本名稱 (WSFCNODE1)。	

故障診斷

問題	解決方案
建立接聽程式失敗，並顯示「WSFC 叢集無法將網路名稱資源上線」訊息。	如需詳細資訊，請參閱 Microsoft 部落格文章建立接聽程式失敗，並顯示訊息「WSFC 叢集無法將網路名稱資源置於線上」。
潛在問題，包括其他接聽程式問題或網路存取問題。	請參閱 Microsoft 說明文件中的永遠在可用性群組組態 (SQL Server) 疑難排解。

相關資源

- 設定永遠開啟可用性群組的唯讀路由
- 疑難排解永遠開啟可用性群組組態 (SQL Server)

其他資訊

步驟 1. 將複本更新為唯讀

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
 (ALLOW_CONNECTIONS = READ_ONLY))
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
 (ALLOW_CONNECTIONS = READ_ONLY))
GO
```

步驟 2. 建立路由網址

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH (SECONDARY_ROLE
 (READ_ONLY_ROUTING_URL = N'TCP://WSFCNode1.tagechtalk.com:1433'))
```

```
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (SECONDARY_ROLE
 (READ_ONLY_ROUTING_URL = N'TCP://WSFCNode2.tagechtalk.com:1433'))
GO
```

步驟 3. 建立路由清單

```
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE1' WITH
 (PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=('WSFCNODE2','WSFCNODE1')));
GO
ALTER AVAILABILITY GROUP [SQLAG1] MODIFY REPLICA ON N'WSFCNODE2' WITH (PRIMARY_ROLE
 (READ_ONLY_ROUTING_LIST=('WSFCNODE1','WSFCNODE2')));
GO
```

步驟 4. 驗證路由清單

```
SELECT AGSrc.replica_server_name AS PrimaryReplica, AGRepl.replica_server_name AS
 ReadOnlyReplica, AGRepl.read_only_routing_url AS RoutingURL , AGRL.routing_priority
 AS RoutingPriority FROM sys.availability_read_only_routing_lists AGRL INNER JOIN
 sys.availability_replicas AGSrc ON AGRL.replica_id = AGSrc.replica_id INNER JOIN
 sys.availability_replicas AGRepl ON AGRL.read_only_replica_id = AGRepl.replica_id
 INNER JOIN sys.availability_groups AV ON AV.group_id = AGSrc.group_id ORDER BY
 PrimaryReplica
```

步驟 5. SQL 命令實用程序

```
sqlcmd -S SQLAG1,1433 -E -d test -K ReadOnly
```

通過在 PGAdmin 中使用 SSH 隧道進行 Connect

由吉萬·謝蒂（AWS）和巴努古迪瓦達（AWS）創建

環境：生產	技術：資料庫；安全性、身分識別、合規性	工作負載：開源
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

出於安全原因，將數據庫放置在私有子網中總是很好的。透過 Amazon Amazon Web Services 務 (AWS) 雲端上公有子網路中的 Amazon 彈性運算雲端 (Amazon EC2) 堡壘主機，可以對資料庫執行查詢。這需要在 Amazon EC2 主機上安裝開發人員或資料庫管理員常用的軟體，例如 pgAdmin 或 DBeaver。

在 Linux 伺服器上執行 pgAdmin 並透過網頁瀏覽器存取，需要安裝額外的相依性、權限設定和設定。

作為替代解決方案，開發人員或資料庫管理員可以使用 pgAdmin 連線到 PostgreSQL 資料庫，從其本機系統啟用 SSH 通道。在這種方法中，pgAdmin 會在連線到資料庫之前使用公有子網路中的 Amazon EC2 主機做為中介主機。「架構」區段中的圖表顯示了設定。

備註：請確保連接到 PostgreSQL 資料庫的安全群組允許從 Amazon EC2 主機在連接埠 5432 上進行連線。

先決條件和限制

先決條件

- 現有的 AWS 帳戶
- 具有公有子網路和私有子網路的虛擬私有雲 (VPC)
- 已連接安全群組的 EC2 執行個體
- 具有安全群組的 Amazon Aurora PostgreSQL 相容版本資料庫
- 用於設置隧道的安全殼層（SSH）key pair

產品版本

- pgAdmin 版本 6.2+
- Amazon Aurora 郵政兼容版 12.7+

架構

目標技術堆疊

- Amazon EC2
- Amazon Aurora 郵政兼容

目標架構

下圖顯示使用 pgAdmin 搭配 SSH 通道，透過網際網路閘道連線至連線至資料庫的 EC2 執行個體。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

其他服務

- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

建立連線

任務	描述	所需技能
建立伺服器。	在 pgAdmin 中，選擇創建，然後選擇服務器。有關設置 pgAdmin 以註冊服務器，配置連接以及使用服務器對話框通過 SSH 隧道進行連接的其他幫助，請參閱相關資源部分中的鏈接。	DBA
提供伺服器的名稱。	在「一般」頁籤上，輸入名稱。	DBA
輸入資料庫詳細資訊。	在「連線」頁籤上，輸入下列項目的值： • 主機名稱/位址 • 連接埠 • 維護資料庫 • 使用者名稱 • 密碼	DBA
輸入 Amazon EC2 服務器詳細信息。	在安全殼層通道索引標籤上，提供位於公有子網路中之 Amazon EC2 執行個體的詳細資訊。 • 將使用 SSH 通道設定為是，以指定 pgAdmin 在連線到指定的伺服器時應該使用 SSH 通道。	DBA

任務	描述	所需技能
	• 在「通道主機」欄位中，指定 SSH 主機的名稱或 IP 位址 (例如，10.x.x.x)。 • 在 [通道連接埠] 欄位中，指定 SSH 主機的連接埠 (例如 22)。 • 在「使用者名稱」欄位中，指定具有 SSH 主機登入權限的使用者名稱 (例如，ec2-user)。 • 將身份驗證類型指定為身份文件，以便 pgAdmin 在連接時使用私鑰文件。 • 在 [身分識別檔案] 欄位中包含 [隱私權增強型郵件] (PEM) 檔案的位置。.pem 檔案是 Amazon EC2 key pair。	
保存並連接。	選擇 [儲存] 以完成設定，並使用安全殼層通道連線至 Aurora PostgreSQL 相容的資料庫。	DBA

相關資源

- 伺服器對話
- Connect 到伺服器

將甲骨文查詢轉換為 SQL 數據庫

由皮尼許辛格（AWS）和洛克許古拉姆（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS
R 型：重新建築	工作量：甲骨文	技術：資料庫；移轉
AWS 服務：Amazon Aurora; Amazon RDS		

Summary

這個從現場部署遷移到 Amazon Web Services (AWS) 雲端的遷移程序使用 AWS 結構描述轉換工具 (AWS SCT) 將程式碼從 Oracle 資料庫轉換為 PostgreSQL 資料庫。大部分的程式碼都會由 AWS SCT 自動轉換。但是，JSON 相關的 Oracle 查詢不會自動轉換。

從 Oracle 12.2 版本開始，甲骨文數據庫支持各種 JSON 功能，在基於 JSON 的數據轉換為基於行的數據幫助。不過，AWS SCT 不會自動將以 JSON 為基礎的資料轉換成 PostgreSQL 支援的語言。

這種遷移模式主要著重於手動將 JSON 相關的 Oracle 查詢轉換為 PostgreSQL 數據庫等功能`JSON_OBJECT``JSON_ARRAYAGG`，以及`JSON_TABLE`從 Oracle 數據庫轉換為 PostgreSQL 數據庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署 Oracle 資料庫執行個體 (啟動並執行)
- 適用於 PostgreSQL 或 Amazon Aurora PostgreSQL 相容版本資料庫執行個體 (啟動並執行) 的 Amazon Relational Database Service 服務 (Amazon RDS)

限制

- JSON 相關查詢需要固定`KEY`和`VALUE`格式。不使用該格式返回錯誤的結果。
- 如果 JSON 結構中的任何更改在結果部分中添加了新的`KEY`和`VALUE`對，則必須在 SQL 查詢中更改相應的過程或函數。

- 一些 JSON 相關的功能在早期版本的甲骨文和 PostgreSQL 中受到支持，但功能較少。

產品版本

- 甲骨文資料庫 12.2 版及更新版本
- Amazon RDS for PostgreSQL SQL 或 Aurora 版本 9.5 及更新版本
- AWS SCT 最新版本 (使用 1.0.664 版進行測試)

架構

源, 技術, 堆棧

- 使用 19c 版本的 Oracle 資料庫執行處理

目標技術堆疊

- 具有第 13 版 Amazon RDS for PostgreSQL 或 Aurora 兼容的資料庫執行個體

目標架構

1. 使用 AWS SCT 搭配 JSON 函數程式碼，將原始程式碼從甲骨文轉換為 PostgreSQL。
2. 此轉換會產生支援 PostgreSQL 的移轉檔案。
3. 手動將未轉換的甲骨文 JSON 函數代碼轉換為 JSON 函數代碼。
4. 在目標 Aurora 與 PostgreSQL 相容的資料庫執行個體上執行 .sql 檔案。

工具

AWS 服務

- Amazon Aurora 是全受管的關聯式資料庫引擎，專為雲端建置，並與 MySQL 和 PostgreSQL 相容。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。

- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

其他服務

- Oracle SQL 開發人員是一個整合式開發環境，可簡化傳統與雲端式部署中 Oracle 資料庫的開發與管理。
- pgAdmin 或 DBAver。 pgAdmin 是一個開放源代碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。DBeaver 是一個通用的數據庫工具。

最佳實務

使用JSON_TABLE函數時，Oracle 查詢具有類型CAST作為默認值。最佳做法是在 PostgreSQL CAST 中使用，使用雙大於字符（ ）。>>

如需詳細資訊，請參閱其他資訊一節中的 < 附加資訊 > 一節。

史诗

在甲骨文和 PostgreSQL 數據庫中生成 JSON 數據

任務	描述	所需技能
將 JSON 數據存儲在甲骨文數據庫中。	在 Oracle 數據庫中創建一個表，並將 JSON 數據存儲在CLOB列中。 使用「其他資訊」區段中的「Oracle_ 表格 _ 建立 _ 插入程序檔」。	移民工程師
將 JSON 資料儲存在 PostgreSQL 中。	在 PostgreSQL 資料庫中建立資料表，並將 JSON 資料儲存在資料行中TEXT。使用「其他資訊」區段中的「建立 _ 插入程序檔」。	移民工程師

將 JSON 轉換為行格式

任務	描述	所需技能
轉換甲骨文數據庫上的 JSON 數據。	編寫甲骨文 SQL 查詢將 JSON 數據讀取為行格式。如需詳細資訊和範例語法，請參閱其他資訊一節中的 Oracle_SQL_READ_ JSON。	移民工程師
轉換資料庫上 PostgreSQL 料。	寫一 PostgreSQL 將 JSON 數據讀取為行格式。有關更多詳細信息和示例語法，請參閱其他信息部分中的 Postgre_SQL_READ_ JSON。	移民工程師

使用 SQL 查詢手動轉換 JSON 資料，並以 JSON 格式報告輸出

任務	描述	所需技能
在 Oracle SQL 查詢上執行彙總和驗證。	若要手動轉換 JSON 資料，請在 Oracle SQL 查詢上執行聯結、彙總和驗證，並以 JSON 格式報告輸出。在「其他資訊」區段中，使用「Oracle_SQL_JSON_ 彙總 _ 聯結」下的程式碼。 1. JOIN — JSON 格式的資料會當做輸入參數傳遞至查詢。一個內部聯接是這個靜態數據和 Oracle 數據庫表中的 JSON 數據之間進行的`aws_test_table` 。 2. 具有驗證的彙總 — JSON 資料具有`KEY`和`VALUE`參數，其值如`accountNu`	移民工程師

任務	描述	所需技能
	mber parentAcc ountNumbe r 、businessU nitId 和positionId ，這些值用於SUM和COUNT彙總。 3. JSON 格式 — 在聯結和彙總之後，會使用JSON_OBJE CT 和以 JSON 格式報告資料JSON_ARRAYAGG 。	

任務	描述	所需技能
對 Postgres SQL 查詢執行彙總和驗證。	若要手動轉換 JSON 資料，請在 PostgreSQL 查詢上執行聯結、彙總和驗證，並以 JSON 格式報告輸出。在 [其他資訊] 區段中，使用 [彙總] 加入下的程式碼。 1. JOIN — JSON 格式化的資料 (tab1) 會當做輸入參數傳遞至WITH子句查詢。一個 JOIN 是這個靜態數據和 JSON 數據，這是在tab表之間進行。一個 JOIN 也與子WITH句，該子句在aws_test_pg_table 表中具有 JSON 數據進行。 2. 彙總 — JSON 資料具有KEY和VALUE參數，其值如accountNumber parentAccountNumber 、businessUnitId 、和positionId ，這些值用於SUM和COUNT彙總。 3. JSON 格式 — 在聯結和彙總之後，會使用JSON_BUILD_OBJECT 和以 JSON 格式報告資料JSON_AGG。	移民工程師

將甲骨文程序轉換為包含 JSON 查詢的 PostgreSQL 函數

任務	描述	所需技能
將甲骨文程序中的 JSON 查詢轉換為行。	對於 Oracle 程序的範例，請使用先前的 Oracle 查詢，以及其他資訊區段中的 Oracle_程序 re_With_JSON_ Query 下的程式碼。	移民工程師
將具有 JSON 查詢的 PostgreSQL 函數轉換為以資料列為基礎的資料。	對於範例 PostgreSQL 函式，請使用先前的 PostgreSQL 查詢和其他資訊區段中的程式碼。	移民工程師

相關資源

- 甲骨文函数
- PostgreSQL 數
- 甲骨文 JSON 函數示例
- PostgreSQL 數範例
- AWS Schema Conversion Tool

其他資訊

要將 JSON 代碼從甲骨文數據庫轉換為 PostgreSQL 數據庫，請按順序使用以下腳本。

1。Oracle 表格 _ 建立 _ 插入 _ 腳本

```
create table aws_test_table(id number,created_on date default sysdate,modified_on
 date,json_doc clob);

REM INSERTING into EXPORT_TABLE
SET DEFINE OFF;
Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc)
values (1,to_date('02-AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022
 12:30:14','DD-MON-YYYY HH24:MI:SS'),TO_CLOB(q'[{
```

```
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -'",
    "a]')
|| TO_CLOB(q'[ccount" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
]')
|| TO_CLOB(q'[        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}]'));
Insert into aws_test_table (ID,CREATED_ON,MODIFIED_ON,json_doc) values (2,to_date('02-
AUG-2022 12:30:14','DD-MON-YYYY HH24:MI:SS'),to_date('02-AUG-2022 12:30:14','DD-MON-
YYYY HH24:MI:SS'),TO_CLOB(q'[{
  "metadata" : {
```

```
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "pqr@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -'",
    "account" : {
      "companyId" : "CARS",
      "busin]')
|| TO_CLOB(q'[essUnitId" : 6,
      "accountNumber" : 42001,
      "parentAccountNumber" : 32001,
      "firstName" : "terry",
      "lastName" : "whitlock",
      "street1" : "UO  123",
      "city" : "TOTORON",
      "region" : "NO",
      "postalcode" : "LKM 111",
      "country" : "Canada"
    },
    "products" : [
      {
        "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
        "id" : "0000000014",
        "name" : "ProView eLooseleaf",
      ]')
|| TO_CLOB(q'[  "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}]'));

commit;
```

2. 後表建立 _ 插入 _ 腳本

```
create table aws_test_pg_table(id int,created_on date ,modified_on date,json_doc text);
insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(1,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
      "accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}');
```

```
insert into aws_test_pg_table(id,created_on,modified_on,json_doc)
values(2,now(),now(),'{
  "metadata" : {
    "upperLastNameFirstName" : "PQR XYZ",
    "upperEmailAddress" : "pqr@gmail.com",
    "profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "54534343",
    "displayName" : "Xyz, pqr",
    "firstName" : "pqr",
    "lastName" : "Xyz",
    "emailAddress" : "a*b**@h**.k**",
    "productRegistrationStatus" : "Not registered",
    "positionId" : "0090",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "CARS",
      "businessUnitId" : 6,
      "accountNumber" : 42001,
      "parentAccountNumber" : 32001,
      "firstName" : "terry",
      "lastName" : "whitlock",
      "street1" : "UO  123",
      "city" : "TOTORON",
      "region" : "NO",
      "postalcode" : "LKM 111",
      "country" : "Canada"
    },
    "products" : [
      {
        "appUserGuid" : "ia744d7790000016899f8cf3f417d6df6",
        "id" : "0000000014",
        "name" : "ProView eLooseleaf",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}');
```

3. 阿拉克列 _ 讀取 JSON

下面的代碼塊顯示了如何將 Oracle JSON 數據轉換為行格式。

查詢和語法範例

```
SELECT   JSON_OBJECT(
 'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                                ) ) ) ) FROM
   (SELECT   tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
 clerk_count
       FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
  COLUMNS (
          parent_account_number NUMBER PATH
           '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( '{
       "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
```

```
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }', '$.accounts[*]' ERROR ON ERROR
      COLUMNS (
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data
      ON ( static_data.parent_account_number = tab_data.parent_account_number
           AND static_data.account_number = tab_data.account_number
           AND static_data.business_unit_id = tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number );
```

JSON 文檔將數據存儲為集合。每個集合可以有KEY和VALUE對。每個都VALUE可以有嵌套KEY和VALUE對。下表提供有關VALUE從 JSON 文件讀取特定內容的資訊。

鑰匙	層次結構或路徑被用來獲取值	價值
profileType	metadata -> profileType	「P」
positionId	data -> positionId	「0100」
accountNumber	data-> 帳號-> accountNumber	42000

在上一個表格中，KEYprofileType是VALUE的 metadataKEY. KEYpositionId是VALUE的一個 dataKEY. KEYaccountNumber是VALUE的一個 accountKEY，而且accountKEY是一個VALUE的dataKEY。

JSON 文件範例

```
{
  "metadata" : {
    "upperLastNameFirstName" : "ABC XYZ",
    "upperEmailAddress" : "abc@gmail.com",
```

```
"profileType" : "P"
  },
  "data" : {
    "onlineContactId" : "032323323",
    "displayName" : "Abc, Xyz",
    "firstName" : "Xyz",
    "lastName" : "Abc",
    "emailAddress" : "abc@gmail.com",
    "productRegistrationStatus" : "Not registered",
"positionId" : "0100",
    "arrayPattern" : " -",
    "account" : {
      "companyId" : "SMGE",
      "businessUnitId" : 7,
"accountNumber" : 42000,
      "parentAccountNumber" : 32000,
      "firstName" : "john",
      "lastName" : "doe",
      "street1" : "retOdertcaShr ",
      "city" : "new york",
      "postalcode" : "XY ABC",
      "country" : "United States"
    },
    "products" : [
      {
        "appUserGuid" : "i0acc4450000001823fbad478e2eab8a0",
        "id" : "0000000046",
        "name" : "ProView",
        "domain" : "EREADER",
        "registrationStatus" : false,
        "status" : "11"
      }
    ]
  }
}
```

SQL 查詢，用於從 JSON 文檔中獲取選定的字段

```
select parent_account_number,account_number,business_unit_id,position_id from
 aws_test_table aws,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
COLUMNS (
parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
account_number NUMBER PATH '$.data.account.accountNumber',
```

```
business_unit_id NUMBER PATH '$.data.account.businessUnitId',
position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'
)) as sc
```

在以前的查詢中，JSON_TABLE是 Oracle 中的內置函數，將 JSON 數據轉換為行格式。JSON_TABLE 函數需要 JSON 格式的參數。

中的每個項目都COLUMNS有一個預定義的PATH，並且有一個VALUE適合給KEY定的行格式返回。

先前查詢的結果

父項帳戶號碼	帳號 _ 號碼	商務單位 ID	位置識別碼
32000	42000	7	0100
32001	42001	6	0090

4. 下一篇文章

查詢和語法範例

```
select *
from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::VARCHAR as positionId
from aws_test_pg_table) d ;
```

在甲骨文中PATH，用於識別具體的KEY和VALUE. 但是，PostgreSQL 使用一個HIERARCHY模型來讀取KEY和VALUE從 JSON 讀取。下面的示例中使用了下Oracle_SQL_Read_JSON面提到的相同 JSON 數據。

不允許使用類型為 CAST 的 SQL 查詢

(如果您強制輸入CAST，查詢會失敗，並顯示語法錯誤。)

```
select *
from (
```

```
select (json_doc::json->'data'->'account'->'parentAccountNumber') as
 parentAccountNumber,
(json_doc::json->'data'->'account'->'accountNumber')as accountNumber,
(json_doc::json->'data'->'account'->'businessUnitId') as businessUnitId,
(json_doc::json->'data'->'positionId')as positionId
from aws_test_pg_table) d ;
```

使用單個大於運算符 (>) 將返回定VALUE義的。KEY例如KEY：positionId、和VALUE："0100"。

當您使用單一大於運算子 () > 時，不允許使用類CAST型。

允許使用 CAST 類型的 SQL 查詢

```
select *
from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar as positionId
from aws_test_pg_table) d ;
```

若要使用 typeCAST，您必須使用雙大於運算子。如果您使用單一大於運算子，查詢會傳回VALUE已定義的 (例如KEY:positionId、和VALUE:"0100")。使用雙大於運算子 (>>) 將傳回為該運算子定義的實際值 KEY (例如KEY:positionId、和VALUE:0100，不含雙引號)。

在前面的情況下，parentAccountNumber是類型CAST為INT，accountNumber類型CAST為INT，類型businessUnitIdCAST為INT，類型positionId為，類型CAST為VARCHAR。

下表顯示的查詢結果說明單一大於運算子 (>) 和雙大於運算子 () 的角色。>>

在第一個資料表資料表中，查詢會使用單一大於運算子 () >。每個資料行都是 JSON 類型，無法轉換成其他資料類型。

parentAccountNumber	帳號	businessUnitId	職位識別碼
2003565430	2003564830	7	「0100」
2005284042	2005284042	6	「0090」
2000272719	2000272719	1	「0100」

在第二個資料表中，查詢會使用雙大於運算子 () >>。每列支持CAST基於列值的類型。例如，INTEGER在此情況下。

parentAccountNumber	帳號	businessUnitId	職位識別碼
2003565430	2003564830	7	0100
2005284042	2005284042	6	0090
2000272719	2000272719	1	0100

5. Oracle_SQL_JSON 彙總 _ 加入

查詢範例

```
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               ) ) ) )
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
```

```
            SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
clerk_count
       FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
 COLUMNS (
         parent_account_number NUMBER PATH
          '$.data.account.parentAccountNumber',
           account_number NUMBER PATH '$.data.account.accountNumber',
           business_unit_id NUMBER PATH '$.data.account.businessUnitId',
           position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
           ) AS tab_data
           INNER JOIN JSON_TABLE ( '{
       "accounts": [{
         "accountNumber": 42000,
         "parentAccountNumber": 32000,
         "businessUnitId": 7
       }, {
         "accountNumber": 42001,
         "parentAccountNumber": 32001,
         "businessUnitId": 6
       }]
     }', '$.accounts[*]' ERROR ON ERROR
     COLUMNS (
     parent_account_number PATH '$.parentAccountNumber',
     account_number PATH '$.accountNumber',
     business_unit_id PATH '$.businessUnitId')
     ) static_data
     ON ( static_data.parent_account_number = tab_data.parent_account_number
          AND static_data.account_number = tab_data.account_number
          AND static_data.business_unit_id = tab_data.business_unit_id )
       GROUP BY
           tab_data.business_unit_id,
           tab_data.parent_account_number,
           tab_data.account_number
   );
```

要將行級數據轉換為 JSON 格式，甲骨文具有內置的功能JSON_OBJECT，
如，JSON_ARRAYJSON_OBJECTAGG，和。JSON_ARRAYAGG

- JSON_OBJECT接受兩個參數：KEY和VALUE。KEY參數本質上應該是硬編碼或靜態的。VALUE參數衍生自表格輸出。
- JSON_ARRAYAGG接受JSON_OBJECT作為參數。這有助於將JSON_OBJECT元素集合為列表。例如，如果您的JSON_OBJECT元素具有多個記錄 (資料集中有多個KEY和VALUE配對)，則

會JSON_ARRAYAGG附加資料集並建立清單。根據數據結構語言，LIST是一組元素。在此上下文中，LIST是一組JSON_OBJECT元素。

下面的例子顯示了一個JSON_OBJECT元素。

```
{
   "taxProfessionalCount": 0,
   "attorneyCount": 0,
   "nonAttorneyCount": 1,
   "clerkCount": 0
}
```

下一個範例顯示兩個JSON_OBJECT元素，以方括號 ([ ]) LIST 表示。

```
[
    {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
,
    {
        "taxProfessionalCount": 2,
        "attorneyCount": 1,
        "nonAttorneyCount": 3,
        "clerkCount":4
      }
]
```

SQL 查詢範例

```
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
```

```
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               )
                        )
                                        )
             )
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE   0 END
            )      tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN        1    ELSE
  0 END
            )      attorney_count,

            SUM(CASE    WHEN tab_data.position_id = '0090' THEN        1    ELSE
  0 END
            )      non_attorney_count,

            SUM(CASE    WHEN tab_data.position_id = '0050' THEN        1    ELSE
  0 END
            )      clerk_count

        FROM
            aws_test_table scco,  JSON_TABLE ( json_doc, '$' ERROR ON ERROR
            COLUMNS (
            parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( '{
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
```

```
          "businessUnitId": 6
        }]
      }', '$.accounts[*]' ERROR ON ERROR
      COLUMNS (
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data ON ( static_data.parent_account_number =
 tab_data.parent_account_number
                         AND static_data.account_number = tab_data.account_number

                         AND static_data.business_unit_id =
 tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number
    );
```

先前 SQL 查詢的輸出範例

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
```

```
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

6. 後的 SQL_JSON 彙總 _ 加入

PostgreSQL 的內置函數，JSON_BUILD_OBJECT並JSON_AGG將行級數據轉換為 JSON 格式。 PostgreSQL JSON_BUILD_OBJECT 和相當JSON_AGG於甲骨JSON_OBJECT文和. JSON_ARRAYAGG

查詢範例

```
select
JSON_BUILD_OBJECT ('accountCounts',
    JSON_AGG(
        JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
        ,'parentAccountNumber',parentAccountNumber
        ,'accountNumber',accountNumber
        ,'totalOnlineContactsCount',online_contacts_count,
        'countByPosition',
            JSON_BUILD_OBJECT (
            'taxProfessionalCount',tax_professional_count
            ,'attorneyCount',attorney_count
            ,'nonAttorneyCount',non_attorney_count
            ,'clerkCount',clerk_count
            )
        )
    )
)
from (
with tab as (select * from (
select (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER as
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER as accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER as businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar as positionId
from aws_test_pg_table) a ) ,
tab1 as ( select
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
```

```
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer
 businessUnitId,
(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
 parentAccountNumber
from (
select '{
        "accounts": [{
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }, {
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }]
      }'::json as jc) b)
select
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0  END)
  tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0  END)      attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0090' THEN      1  ELSE      0 END)
  non_attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0050' THEN      1  ELSE      0 END)
  clerk_count
from tab1,tab
where tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
and tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
and tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY      tab.businessUnitId::text,
            tab.parentAccountNumber::text,
            tab.accountNumber::text) a;
```

上述查詢的範例輸出

從甲骨文和 PostgreSQL 的輸出是完全一樣的。

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
```

```
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

7. 使用 _JSON_ 查詢的組織程序

此程式碼會將甲骨文程序轉換成具有 JSON SQL 查詢的 PostgreSQL 函式。它顯示了查詢如何將 JSON 轉換為行，反向。

```
CREATE OR REPLACE PROCEDURE p_json_test(p_in_accounts_json IN varchar2,
 p_out_accunts_json  OUT varchar2)
IS
BEGIN
/*
p_in_accounts_json paramter should have following format:
       {
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
```

```
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }
*/
SELECT
    JSON_OBJECT(
        'accountCounts' VALUE JSON_ARRAYAGG(
            JSON_OBJECT(
                'businessUnitId' VALUE business_unit_id,
                        'parentAccountNumber' VALUE parent_account_number,
                        'accountNumber' VALUE account_number,
                        'totalOnlineContactsCount' VALUE online_contacts_count,
                        'countByPosition' VALUE
                    JSON_OBJECT(
                        'taxProfessionalCount' VALUE tax_count,
                        'attorneyCount' VALUE attorney_count,
                        'nonAttorneyCount' VALUE non_attorney_count,
                        'clerkCount' VALUE clerk_count
                               ) ) ) )
into p_out_accunts_json
FROM
    (SELECT
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number,
            SUM(1) online_contacts_count,
            SUM(CASE WHEN tab_data.position_id = '0095' THEN  1 ELSE 0 END) tax_count,
            SUM(CASE    WHEN tab_data.position_id = '0100' THEN 1 ELSE 0 END)
 attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0090' THEN 1 ELSE 0 END)
 non_attorney_count,
            SUM(CASE    WHEN tab_data.position_id = '0050' THEN 1 ELSE 0 END)
 clerk_count
        FROM aws_test_table scco,JSON_TABLE ( json_doc, '$' ERROR ON ERROR
            COLUMNS (
            parent_account_number NUMBER PATH '$.data.account.parentAccountNumber',
            account_number NUMBER PATH '$.data.account.accountNumber',
            business_unit_id NUMBER PATH '$.data.account.businessUnitId',
            position_id VARCHAR2 ( 4 ) PATH '$.data.positionId'    )
            ) AS tab_data
            INNER JOIN JSON_TABLE ( p_in_accounts_json, '$.accounts[*]' ERROR ON ERROR

      COLUMNS (
```

```
      parent_account_number PATH '$.parentAccountNumber',
      account_number PATH '$.accountNumber',
      business_unit_id PATH '$.businessUnitId')
      ) static_data
      ON ( static_data.parent_account_number = tab_data.parent_account_number
           AND static_data.account_number = tab_data.account_number
           AND static_data.business_unit_id = tab_data.business_unit_id )
        GROUP BY
            tab_data.business_unit_id,
            tab_data.parent_account_number,
            tab_data.account_number
    );
EXCEPTION
WHEN OTHERS THEN
   raise_application_error(-20001,'Error while running the JSON query');
END;
/
```

執行程序

下列程式碼區塊說明如何使用範例 JSON 輸入來執行先前建立的 Oracle 程序。它還為您提供此過程的結果或輸出。

```
set serveroutput on;
declare
v_out varchar2(30000);
v_in varchar2(30000):= '{
        "accounts": [{
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }, {
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }]
      }';
begin
  p_json_test(v_in,v_out);
  dbms_output.put_line(v_out);
end;
/
```

程序輸出

```
{
  "accountCounts": [
    {
      "businessUnitId": 6,
      "parentAccountNumber": 32001,
      "accountNumber": 42001,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": 7,
      "parentAccountNumber": 32000,
      "accountNumber": 42000,
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

8. 使用 _JSON 查詢發佈函數

範例函數

```
CREATE OR REPLACE  FUNCTION f_pg_json_test(p_in_accounts_json  text)
RETURNS text
LANGUAGE plpgsql
AS
$$
DECLARE
 v_out_accunts_json   text;
BEGIN
```

```
SELECT
JSON_BUILD_OBJECT ('accountCounts',
    JSON_AGG(
        JSON_BUILD_OBJECT ('businessUnitId',businessUnitId
        ,'parentAccountNumber',parentAccountNumber
        ,'accountNumber',accountNumber
        ,'totalOnlineContactsCount',online_contacts_count,
        'countByPosition',
            JSON_BUILD_OBJECT (
            'taxProfessionalCount',tax_professional_count
            ,'attorneyCount',attorney_count
            ,'nonAttorneyCount',non_attorney_count
            ,'clerkCount',clerk_count
            ))))
INTO v_out_accunts_json
FROM (
WITH tab AS (SELECT * FROM (
SELECT (json_doc::json->'data'->'account'->>'parentAccountNumber')::INTEGER AS
 parentAccountNumber,
(json_doc::json->'data'->'account'->>'accountNumber')::INTEGER AS accountNumber,
(json_doc::json->'data'->'account'->>'businessUnitId')::INTEGER AS businessUnitId,
(json_doc::json->'data'->>'positionId')::varchar AS positionId
FROM aws_test_pg_table) a ) ,
tab1 AS ( SELECT
(json_array_elements(b.jc -> 'accounts') ->> 'accountNumber')::integer accountNumber,
(json_array_elements(b.jc -> 'accounts') ->> 'businessUnitId')::integer businessUnitId,
(json_array_elements(b.jc -> 'accounts') ->> 'parentAccountNumber')::integer
 parentAccountNumber
FROM (
SELECT p_in_accounts_json::json AS jc) b)
SELECT
tab.businessUnitId::text,
tab.parentAccountNumber::text,
tab.accountNumber::text,
SUM(1) online_contacts_count,
SUM(CASE WHEN tab.positionId::text = '0095' THEN 1 ELSE 0  END)
  tax_professional_count,
SUM(CASE WHEN tab.positionId::text = '0100' THEN 1 ELSE 0  END)      attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0090' THEN      1  ELSE      0 END)
  non_attorney_count,
SUM(CASE  WHEN tab.positionId::text = '0050' THEN      1  ELSE      0 END)
  clerk_count
FROM tab1,tab
WHERE tab.parentAccountNumber::INTEGER=tab1.parentAccountNumber::INTEGER
```

```
AND tab.accountNumber::INTEGER=tab1.accountNumber::INTEGER
AND tab.businessUnitId::INTEGER=tab1.businessUnitId::INTEGER
GROUP BY      tab.businessUnitId::text,
            tab.parentAccountNumber::text,
            tab.accountNumber::text) a;
RETURN v_out_accunts_json;
END;
$$;
```

執行函數

```
select    f_pg_json_test('{
        "accounts": [{
          "accountNumber": 42001,
          "parentAccountNumber": 32001,
          "businessUnitId": 6
        }, {
          "accountNumber": 42000,
          "parentAccountNumber": 32000,
          "businessUnitId": 7
        }]
      }')   ;
```

功能輸出

下列輸出與 Oracle 程序輸出類似。不同之處在於此輸出是文本格式。

```
{
  "accountCounts": [
    {
      "businessUnitId": "6",
      "parentAccountNumber": "32001",
      "accountNumber": "42001",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 0,
        "nonAttorneyCount": 1,
        "clerkCount": 0
      }
    },
    {
      "businessUnitId": "7",
```

```
      "parentAccountNumber": "32000",
      "accountNumber": "42000",
      "totalOnlineContactsCount": 1,
      "countByPosition": {
        "taxProfessionalCount": 0,
        "attorneyCount": 1,
        "nonAttorneyCount": 0,
        "clerkCount": 0
      }
    }
  ]
}
```

使用自訂實作跨帳戶複製 Amazon DynamoDB 表格

創建者：拉姆庫瑪·拉馬努加姆 (AWS)

環境：生產	資料來源 Amazon DynamoDB	目標：Amazon DynamoDB
R 類型：不適用	工作負載：所有其他工作	技術：資料庫
AWS 服務：Amazon DynamoDB		

Summary

在亞馬遜網路服務 (AWS) 上使用 Amazon DynamoDB 時，常見使用案例是使用生產環境中的表資料複製或同步處理開發、測試或暫存環境中的 DynamoDB 表。標準做法是，每個環境都使用不同的 AWS 帳戶。

DynamoDB 現在支援使用 AWS Backup 的跨帳戶備份。如需使用 AWS Backup 時相關儲存成本的相關資訊，請參閱 AWS Backup 定價。當您使用 AWS Backup 跨帳戶進行複製時，來源和目標帳戶必須是 AWS Organizations 組織的一部分。還有其他使用 AWS 服務 (例如 AWS Data Pipeline 或 AWS Glue) 的跨帳戶備份和還原解決方案。但是，使用這些解決方案會增加應用程式的佔用空間，因為需要部署和維護更多 AWS 服務。

您也可以使用 Amazon DynamoDB 串流來擷取來源帳戶中的表格變更。然後，您可以啟動 AWS Lambda 函數，並在目標帳戶的目標資料表中進行相應的變更。但是該解決方案適用於必須始終保持同步源表和目標表的用例。它可能不適用於經常更新資料的開發、測試和測試環境。

此模式提供實作自訂解決方案的步驟，將 Amazon DynamoDB 表格從一個帳戶複製到另一個帳戶。這種模式可以使用常見的編程語言，如 C# ，Java 和 Python 來實現。我們建議使用 AWS 開發套件支援的語言。

先決條件和限制

先決條件

- 兩個作用中的 AWS 帳戶
- 兩個帳戶中的 DynamoDB 表
- AWS Identity and Access Management (IAM) 角色和政策的知識

- 有關如何使用任何通用程式設計語言 (例如 C#、Java 或 Python) 存取亞馬遜動態資料表的知識

限制

此模式適用於大約 2 GB 或更小的 DynamoDB 表格。透過其他邏輯來處理連線或工作階段中斷、節流以及失敗和重試，可用於較大的資料表。

從來源資料表讀取項目的 DynamoDB 掃描作業只能在單一呼叫中擷取最多 1 MB 的資料。對於大於 2 GB 的大型資料表，此限制可能會增加執行完整表格複製的總時間。

架構

自動化和規模

此模式適用於大小較小 (約 2 GB) 的 DynamoDB 表格。

若要將此模式套用至較大的資料表，請解決下列問題：

- 在資料表複製作業期間，會使用不同的安全性權杖來維護兩個作用中工作階段。如果表複製操作花費的時間超過令牌到期時間，則必須設置邏輯才能刷新安全令牌。
- 如果未佈建足夠的讀取容量單位 (RCU) 和寫入容量單位 (WCU)，則來源或目標資料表上的讀取或寫入可能會受到限制。一定要 catch 並處理這些異常。
- 處理任何其他失敗或例外狀況，並設置重試機制，以便從複製作業失敗的位置重試或繼續。

工具

工具

- 亞馬遜 DynamoDB — Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- 所需的其他工具會根據您為實作選擇的程式設計語言而有所不同。例如，如果您使用 C#，您將需要 Microsoft 視覺工作室和以下軟 NuGet 件包：
  - `AWSSDK`
  - `AWSSDK.DynamoDBv2`

Code

下列 Python 程式碼片段會使用 Boto3 程式庫刪除並重新建立 DynamoDB 資料表。

請勿使用 IAM 使用者AWS_SECRET_ACCESS_KEY的AWS_ACCESS_KEY_ID和，因為這些是長期登入資料，應避免以程式設計方式存取 AWS 服務。如需有關暫時登入資料的詳細資訊，請參閱最佳做法一節。

下列程式碼片段中TEMPORARY_SESSION_TOKEN使用的AWS_SECRET_ACCESS_KEY、和是從 AWS 安全性權杖服務 (AWS STS) 擷取的臨時登入資料。AWS_ACCESS_KEY_ID

```
import boto3
import sys
import json

#args = input-parameters = GLOBAL_SEC_INDEXES_JSON_COLLECTION,
 ATTRIBUTES_JSON_COLLECTION, TARGET_DYNAMODB_NAME, TARGET_REGION, ...

#Input param: GLOBAL_SEC_INDEXES_JSON_COLLECTION
#[{"IndexName":"Test-index","KeySchema":[{"AttributeName":"AppId","KeyType":"HASH"},
{"AttributeName":"AppType","KeyType":"RANGE"}],"Projection":
{"ProjectionType":"INCLUDE","NonKeyAttributes":["PK","SK","OwnerName","AppVersion"]}}]

#Input param: ATTRIBUTES_JSON_COLLECTION
#[{"AttributeName":"PK","AttributeType":"S"},
{"AttributeName":"SK","AttributeType":"S"},
{"AttributeName":"AppId","AttributeType":"S"},
{"AttributeName":"AppType","AttributeType":"N"}]

region = args['TARGET_REGION']
target_ddb_name = args['TARGET_DYNAMODB_NAME']

global_secondary_indexes = json.loads(args['GLOBAL_SEC_INDEXES_JSON_COLLECTION'])
attribute_definitions = json.loads(args['ATTRIBUTES_JSON_COLLECTION'])

# Drop and create target DynamoDB table
dynamodb_client = boto3.Session(
        aws_access_key_id=args['AWS_ACCESS_KEY_ID'],
        aws_secret_access_key=args['AWS_SECRET_ACCESS_KEY'],
        aws_session_token=args['TEMPORARY_SESSION_TOKEN'],
    ).client('dynamodb')

# Delete table
print('Deleting table: ' + target_ddb_name + ' ...')
```

```
try:
    dynamodb_client.delete_table(TableName=target_ddb_name)

    #Wait for table deletion to complete
    waiter = dynamodb_client.get_waiter('table_not_exists')
    waiter.wait(TableName=target_ddb_name)
    print('Table deleted.')
except dynamodb_client.exceptions.ResourceNotFoundException:
    print('Table already deleted / does not exist.')
    pass

print('Creating table: ' + target_ddb_name + ' ...')

table = dynamodb_client.create_table(
    TableName=target_ddb_name,
    KeySchema=[
        {
            'AttributeName': 'PK',
            'KeyType': 'HASH'  # Partition key
        },
        {
            'AttributeName': 'SK',
            'KeyType': 'RANGE'  # Sort key
        }
    ],
    AttributeDefinitions=attribute_definitions,
    GlobalSecondaryIndexes=global_secondary_indexes,
    BillingMode='PAY_PER_REQUEST'
)

waiter = dynamodb_client.get_waiter('table_exists')
waiter.wait(TableName=target_ddb_name)

print('Table created.')
```

最佳實務

臨時憑證

作為安全最佳實務，以程式設計方式存取 AWS 服務時，請避免使用 IAM 使用者AWS_SECRET_ACCESS_KEY的AWS_ACCESS_KEY_ID和，因為這些都是長期登入資料。一律嘗試使用臨時登入資料以程式設計方式存取 AWS 服務。

舉例來說，開發人員在開發期間對應用程式中 IAM 使用者AWS_SECRET_ACCESS_KEY的AWS_ACCESS_KEY_ID和進行硬式編碼，但在將變更推送至程式碼儲存庫之前，無法移除硬式編碼的值。這些暴露的認證可能會被非預期或惡意的用戶使用，這些用戶可能會產生嚴重影響（特別是當公開的憑據具有管理員權限時）。應使用 IAM 主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 立即停用或刪除這些暴露的登入資料。

若要取得用於程式設計方式存取 AWS 服務的臨時登入資料，請使用 AWS STS。臨時身份證明僅在指定的時間內有效（從 15 分鐘到 36 小時）。允許的臨時證明資料持續時間上限會因角色設定和角色鏈結等因素而有所不同。如需 AWS STS 的詳細資訊，請參閱文件。

史诗

設定 DynamoDB 料表

任務	描述	所需技能
建立 DynamoDB 資料表。	在來源和目標 AWS 帳戶中建立具有索引的 DynamoDB 表格。 將容量佈建設定為隨需模式，這可讓 DynamoDB 根據工作負載動態擴展讀取/寫入容量。 或者，您可以將佈建的容量與 4000 個 RCU 和 4000 個 WCU 搭配使用。	應用程式開發人員、DBA、移轉工程師
填入來源表格。	將測試資料填入來源帳戶中的 DynamoDB 表格。至少有 50 MB 或更多的測試資料可協助您查看表格複製期間使用的尖峰和平均 RCU。然後，您可以視需要變更容量佈建。	應用程式開發人員、DBA、移轉工程師

設定認證以存取 DynamoDB 表格

任務	描述	所需技能
建立 IAM 角色以存取來源和目標 DynamoDB 表格。	在來源帳戶中建立具有存取 (讀取) 來源帳戶中 DynamoDB 表格的權限的 IAM 角色。 將來源帳戶新增為此角色的信任實體。 在目標帳戶中建立 IAM 角色，具有存取 (建立、讀取、更新、刪除) 目標帳戶中 DynamoDB 表的權限。 將目標帳戶新增為此角色的信任實體。	AWS 應用程式開發人員 DevOps

將表資料從一個帳戶複製到另一個帳戶

任務	描述	所需技能
取得 IAM 角色的臨時登入資料。	取得在來源帳戶中建立的 IAM 角色的臨時登入資料。 取得在目標帳戶中建立 IAM 角色的臨時登入資料。 取得 IAM 角色的臨時登入資料的一種方法是使用 AWS CLI 中的 AWS STS。 `aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/<role-name> --role-session-name`	應用程式開發人員、移轉

任務	描述	所需技能
	<session-name> --profile <profile-name> 使用適當的 AWS 設定檔 (對應於來源或目標帳戶)。 如需取得暫時登入資料之不同方式的詳細資訊，請參閱下列內容： • AWS Security Token Service API 參考 • 取得用於 CLI 存取的 IAM 角色登入資料	
針對來源和目標 DynamoDB 存取，初始化 DynamoDB 用戶端。	針對來源和目標 DynamoDB 表格，初始化 AWS 開發套件所提供的 DynamoDB 用戶端。 • 對於來源 DynamoDB 用戶端，請使用從來源帳戶擷取的臨時登入資料。 • 對於目標 DynamoDB 用戶端，請使用從目標帳戶擷取的臨時登入資料。 如需使用 IAM 臨時登入資料提出請求的詳細資訊，請參閱 AWS 文件。	應用程式開發人員

任務	描述	所需技能
卸除並重新建立目標資料表。	使用目標帳戶 DynamoDB 用戶端，刪除並重新建立目標帳戶中的目標 DynamoDB 表格 (以及索引)。 從 DynamoDB 表中刪除所有記錄是一項昂貴的操作，因為它會消耗佈建的 WCU。刪除和重新創建表可避免這些額外費用。 您可以在建立表格之後將索引新增至表格，但這需要花費 2 到 5 分鐘的時間。透過將索引集合傳遞至`createTable` 呼叫，在建立資料表期間建立索引會更有效率。	應用程式開發人員

任務	描述	所需技能
執行表格複製。	重複以下步驟，直到複製所有資料為止： • 使用來源 DynamoDB 用戶端對來源帳戶中的表格執行掃描。每個 DynamoDB 掃描只會從表格擷取 1 MB 的資料，因此您必須重複此操作，直到讀取所有項目或記錄為止。 • 對於每組掃描項目，請使用適用於 DynamoDB 的 AWS 開發套件中的`BatchWriteItem` 呼叫，使用目標 DynamoDB 用戶端將項目寫入目標帳戶中的表格。這樣可以減少向 DynamoDB 發出的`PutItem`請求數量。 • `BatchWriteItem` 寫入或置入的限制為 25 次，或者最多 16 MB。在呼叫之前，您必須新增邏輯，以 25 個計數累積掃描項目`BatchWriteItem` 。`BatchWriteItem` 返回無法成功複製的項目列表。使用此清單，新增重試邏輯，以僅針對未成功的項目執行另一個`BatchWriteItem` 呼叫。 如需詳細資訊，請參閱附件區段中 C# 中的參考實作 (用於卸除、建立和填入資料表)。還附	應用程式開發人員

任務	描述	所需技能
	加了示例表配置 JavaScript 對象符號（JSON）文件。	

相關資源

- Amazon DynamoDB 明文件
- 在您的 AWS 帳戶中建立 IAM 使用者
- AWS 開發套件
- 搭配 AWS 資源使用臨時登入資料

其他資訊

此模式是使用 C# 來複製含有 200,000 個項目的 DynamoDB 資料表 (平均項目大小為 5 KB，資料表大小為 250 MB) 來實作。目標 DynamoDB 料表已設定為具有 4000 個 RCU 和 4000 個 WCU 的佈建容量。

完整的資料表複製作業 (從來源帳戶到目標帳戶)，包括卸除和重新建立資料表，都需要 5 分鐘。消耗的總容量單位：30,000 個 RCU 和約 40 萬個 WCU。

如需 DynamoDB 容量模式的詳細資訊，請參閱 AWS 文件中的讀取/寫入容量模式。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Backup 跨帳戶複製 Amazon DynamoDB 表

創建者：拉姆庫瑪·拉馬努加姆 (AWS)

環境：PoC 或試點	技術：資料庫；移轉	AWS 服務：Amazon DynamoDB、AWS Backup

Summary

在亞馬遜網路服務 (AWS) 上使用 Amazon DynamoDB 時，常見使用案例是使用生產環境中的表格資料複製或同步開發、測試或暫存環境中的 DynamoDB 表。標準做法是，每個環境都使用不同的 AWS 帳戶。

AWS Backup 支援 DynamoDB、亞馬遜簡單儲存服務 (Amazon S3) 和其他 AWS 服務的跨區域和跨帳戶資料備份和還原。此模式提供使用 AWS Backup 跨帳戶備份和還原在 AWS 帳戶之間複製 DynamoDB 表的步驟。

先決條件和限制

先決條件

- 屬於相同 AWS 組織組織的兩個有效 AWS 帳戶
- 兩個帳戶中的 DynamoDB 資料表。
- 建立和使用 AWS 備份文件庫的 AWS Identity and Access Management (IAM) 許可

限制

- 來源和目標 AWS 帳戶應屬於同一個 AWS Organizations 組織。

架構

目標技術堆疊

- AWS Backup
- Amazon DynamoDB

目標架構

1. 在來源帳戶的 AWS Backup 保存庫中建立 DynamoDB 表格備份。
2. 將備份複製到目標帳戶中的備份儲存庫。
3. 使用目標帳戶備份保存庫中的備份還原目標帳戶中的 DynamoDb 表格。

自動化和規模

您可以使用 AWS Backup 來排定備份以特定的時間間隔執行。

工具

- AWS Backup — AWS Backup 是一種全受管服務，可集中和自動化 AWS 服務、雲端和現場部署的資料保護。使用此服務，您可以集中一處設定備份政策和監控 AWS 資源的活動。它可讓您自動化和合併先前執行的備份工作 service-by-service，並且不需要建立自訂指令碼和手動程序。
- 亞馬遜 DynamoDB — Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。

史诗

在來源和目標帳戶中開啟 AWS Backup 功能

任務	描述	所需技能
開啟 DynamoDB 和跨帳戶備份的進階功能。	在來源和目標 AWS 帳戶中，執行下列動作： 1. 在 AWS 管理主控台上，開啟 AWS Backup 主控台。 2. 選擇設定。 3. 在 Amazon DynamoDB 備份的進階功能下，確認已啟用進階功能，或選擇啟用。 4. 在 [跨帳戶管理] 下方，針對 [跨帳戶備份] 選擇 [啟用]。	AWS DevOps，移轉工程師

在來源和目標帳戶中建立備份儲存庫

任務	描述	所需技能
建立備份儲存庫。	在來源和目標 AWS 帳戶中，執行下列動作： 1. 在 AWS Backup 主控台上，選擇 Backup 保管庫。 2. 選擇 Create backup vault (建立備份文件庫)。 3. 複製備份儲存庫的 Amazon 資源名稱 (ARN) 並加以儲存。 在來源帳戶和目標帳戶之間複製 DynamoDB 表格備份時，將需要來源備份和目標備份儲存庫的 ARN。	AWS DevOps，移轉工程師

使用備份儲存庫執行備份與還原

任務	描述	所需技能
在來源帳戶中，建立 DynamoDB 資料表備份。	若要為來源帳戶中的 DynamoDB 表格建立備份，請執行下列動作： 1. 在 AWS Backup 儀表板頁面上，選擇建立隨需備份。 2. 在 [設定] 區段中，選取 DynamoDB 做為 [資源類型]，然後選取表格名稱。 3. 在「B ackup 保管庫」下拉式清單中，選取您在來源帳戶中建立的備份保管庫。	AWS DevOps、DBA、移轉工程師

任務	描述	所需技能
	4. 選取您要的「保留期間」。 5. 選擇 Create on-demand backup (建立隨需備份)。 隨即建立新的備份工作。 若要監控 Backup 任務的狀態，請在 AWS Backup 任務頁面上選擇備份任務標籤。所有使用中、進行中及已完成的備份工作都會列在此索引標籤中。	

任務	描述	所需技能
將備份從來源帳戶複製到目標帳戶。	備份工作完成後，將 DynamoDB 表備份從來源帳戶的備份保存庫複製到目標帳戶中的備份保存庫。 若要複製備份資料保險箱，請在來源帳戶中執行下列操作： 1. 在 AWS Backup 主控台上，選擇 Backup 保管庫。 2. 在 [備份] 下方，選擇 DynamoDB 資料表備份。 3. 選擇 Actions (動作)、Copy (複製)。 4. 輸入目標帳戶的 AWS 區域。 5. 對於「外部儲存庫 ARN」，請輸入您在目標帳戶中建立的備份儲存庫的 ARN。 6. 若要將備份從來源帳戶複製到目標帳戶，請在目標帳戶備份保存庫中啟用來自其他帳戶的存取。	AWS DevOps、遷移工程師、DBA
還原目標帳戶中的備份。	在目標 AWS 帳戶中，執行以下操作： 1. 在 AWS Backup 主控台上，選擇 Backup 保管庫。 2. 在 [備份] 下，選取您從來源帳戶複製的備份。 3. 選擇動作，還原。 4. 輸入您要還原的目標 DynamoDB 表格的名稱。	AWS DevOps、DBA、移轉工程師

相關資源

- 搭配使用 AWS Backup
- 跨 AWS 帳戶建立備份副本
- AWS Backup 定價

為 Amazon RDS 和 Amazon Aurora 創建詳細的成本和用量報告

創建者：拉什瑪南拉沙曼南（AWS）和蘇達山那拉西姆罕

環境：生產	技術：資料庫、成本管理、分析	AWS 服務：Amazon Athena；Amazon Aurora；Amazon RDS；AWS Billing and Cost Management

Summary

此模式顯示如何透過設定使用者定義的成本分配標籤來追蹤 Amazon Relational Database Service (Amazon RDS) 或 Amazon Aurora 叢集的使用成本。您可以使用這些標籤，在 AWS Cost Explorer 中為跨多個維度的叢集建立詳細的成本和用量報告。例如，您可以追蹤團隊、專案或成本中心層級的使用成本，然後分析 Amazon Athena 中的資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一或多個 Amazon RDS 或 Amazon Aurora 執行個體

限制

如需標記限制，請參閱 AWS 帳單使用者指南。

架構

目標技術堆疊

- Amazon RDS 或 Amazon Aurora
- AWS 成本和用量報告
- AWS Cost Explorer
- Amazon Athena

工作流程和架構

標籤和分析工作流程包含下列步驟：

1. 資料工程師、資料庫管理員或 AWS 管理員會為 Amazon RDS 或 Aurora 叢集建立使用者定義的成本分配標籤。
2. AWS 管理員會啟用標籤。
3. 標籤會向 AWS Cost Explorer 報告中繼資料。
4. 資料工程師、資料庫管理員或 AWS 管理員會建立每月成本分配報告。
5. 資料工程師、資料庫管理員或 AWS 管理員使用 Amazon Athena 分析每月成本分配報告。

下圖顯示如何套用標籤來追蹤 Amazon RDS 或 Aurora 執行個體的使用成本。

下列架構圖顯示成本分配報告如何與 Amazon Athena 整合以進行分析。

每月成本分配報告存放在您指定的 Amazon S3 儲存貯體中。使用 AWS CloudFormation 範本設定 Athena 時 (如史詩部分所述)，範本會佈建數個其他資源，包括 AWS Glue 爬蟲程式、AWS Glue 資料庫、亞馬遜簡單通知系統 (Amazon SNS) 事件、AWS Lambda 函數，以及 Lambda 函數的 AWS 身分與存取管理 (IAM) 角色，以及 Lambda 函數的 AWS 身分與存取管理 (IAM) 角色。當新的成本資料檔案送達 S3 儲存貯體時，會使用事件通知將這些檔案轉寄至 Lambda 函數以進行處理。Lambda 函數會啟動 AWS Glue 爬行程式任務，以建立或更新 AWS Glue 資料型錄中的表格。然後，此表格會用來查詢 Athena 中的資料。

工具

- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon S3 中的資料。
- Amazon Aurora 是全受管的關聯式資料庫引擎，專為雲端建置，並與 MySQL 和 PostgreSQL 相容。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。

- AWS CloudFormation 是一種基礎設施即程式碼 (IaC) 服務，可讓您輕鬆建立 AWS 和第三方資源的模型、佈建和管理。
- AWS Cost Explorer 可協助您檢視和分析 AWS 成本和用量。

史诗

為您的 Amazon RDS 或 Aurora 叢集建立和啟用標籤

任務	描述	所需技能
為您的 Amazon RDS 或 Aurora 叢集建立使用者定義的成本分配標籤。	若要將標籤新增至新的或現有的 Amazon RDS 或 Aurora 叢集，請按照 Amazon Aurora 使用者指南中新增、列出和移除標籤中的指示進行操作。 附註：如需如何設定 Amazon Aurora 叢集的詳細資訊，請參閱 Amazon Aur ora 使用者指南中的 MySQL 和 Postgre SQL 的說明。	AWS 管理員、資料工程師、DBA
啟動使用者定義的成本配置標籤。	遵循 AWS 帳單使用者指南中的啟用使用者定義的成本分配標籤中的指示。	AWS 管理員

建立成本和用量報告

任務	描述	所需技能
建立和設定叢集的成本和用量報告。	1. 登入 AWS 管理主控台並開啟 AWS 帳單主控台。 2. 在左側導航窗格中，選擇「成本和用量報表」。 3. 選擇 Create report (建立報告)。	應用程式擁有者、AWS 管理員、DBA、一般 AWS、資料工程師

任務	描述	所需技能
	4. 提供報告名稱，保留其他選項的預設設定，然後選擇 [下一步]。 5. 選擇設定並提供現有 S3 儲存貯體的詳細資訊。您也可以選擇從此畫面建立新的 S3 儲存貯體。選擇下一步。 6. 確認要套用至值區的預設政策，選取確認核取方塊，然後選擇 [儲存]。 7. 對於「報表路徑前置詞」，請指定您要在報表名稱前面加上的首碼。 8. 針對「時間粒度」，選擇「每小時」、「每日」或「每月」，視您希望為報表收集資料的頻率而定。 9. 對於「報告」版本控制，請選擇是要個別建立報告的新版本，還是以每個版本覆寫現有報告。 10對於「啟用報表資料整合」，請選擇 Amazon Athena。確認壓縮類型已設定為「實木地板」。 11選擇下一步。 12檢閱報告設定，然後選擇 [檢閱並完成]。 數據將在 24 小時內提供。	

分析成本和用量報告資料

任務	描述	所需技能
分析成本和用量報告資料。	1. 設定並使用 Athena 來分析報告資料。如需指示，請參閱 AWS 成本和用量報告使用者指南中的使用 Amazon Athena 查詢成本和用量報告。我們建議您使用 A thena 提供的 AWS CloudFormation 範本。 2. 執行 Athena 查詢。例如，您可以使用下列 SQL 查詢來檢查資料重新整理的狀態。 `select status from cost_and_usage_data_status` 如需詳細資訊，請參閱 AWS 成本和用量報告使用者指南中的執行 Amazon Athena 查詢。 注意：當您執行 SQL 查詢時，請確定已從下拉式清單中選取正確的資料庫。	應用程式擁有者、AWS 管理員、DBA、一般 AWS、資料工程師

相關資源

參考

- 使用 AWS CloudFormation 範本設定 Athena (建議使用)
- 手動設定 Athena
- Amazon Athena 查詢

- 將報表資料載入其他資源

教學課程和影片

- 使用 Amazon Athena 分析成本和用量報告 (YouTube 影片)

使用 Aurora 中的自訂端點模擬 Oracle RAC 工作負載

由 HariKrishna 博加達 (AWS) 創建

環境：PoC 或試點	來源：數據庫：關係	目標：Aurora
R 類型：重新平台	工作量：甲骨文	技術：資料庫；移轉
AWS 服務：Amazon Aurora; Amazon CloudWatch		

Summary

此模式說明如何使用 Amazon Aurora PostgreSQL 相容版本搭配使用可在單一叢集內的執行個體之間分配工作負載的自訂端點，以模擬 Oracle Real 應用程式叢集 (Oracle RAC) 工作負載中的服務。此模式會示範如何為 Amazon Aurora 資料庫建立自訂端點。自訂端點可讓您在 Aurora 叢集中的不同資料庫執行個體集之間分配和負載平衡工作負載。

在 Oracle RAC 環境中，服務可以跨越一或多個執行處理，並根據交易效能促進工作負載平衡。服務功能包括 end-to-end 自動復原、按工作負載滾動變更，以及完整的位置透明度。您可以使用此樣式來模擬其中一些特徵。例如，您可以模擬為報表應用程式路由連線的功能。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一 PostgreSQL 動程序
- Aurora 郵政兼容的數據庫
- 甲骨文 RAC 數據庫遷移到 Aurora 與 PostgreSQL 相容的數據庫

限制

- 如需適用於自訂端點的限制，請參閱 Amazon RDS 文件中的指定自訂端點的屬性。

架構

源, 技術, 堆棧

- 一個三節點的甲骨文 RAC 數據庫

目標技術堆疊

- 具有兩個僅供讀取複本的 Aurora PostgreSQL 相容資料庫

來源架構

下圖顯示三節點 Oracle RAC 資料庫的架構。

目標架構

下圖顯示具有兩個僅供讀取複本的 Aurora PostgreSQL 相容資料庫的架構。三個不同的應用程式/服務使用自訂端點，這些端點為不同的應用程式使用者提供服務，並在主要和僅供讀取複本之間重新導向流

工具

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

史诗

建立 Aurora 與 PostgreSQL 相容的叢集

任務	描述	所需技能
建立叢集。	若要建立叢集，請參閱 Amazon RDS 說明文件中的建立資料庫叢集並連接至 Aurora PostgreSQL 資料庫叢集上的資料庫。	AWS 管理員
為工作負載建立自訂參數群組。	若要建立參數群組，請參閱 Amazon RDS 文件中的建立資料庫叢集參數群組。	AWS 管理員
建立事件通知和警報。	您可以使用事件通知和 Amazon CloudWatch 警示，在叢集狀態變更時通知您，並在達到預先定義的閾值時擷取指標。 若要建立 CloudWatch 警示，請參閱 CloudWatch 文件中的根據靜態閾值建立 CloudWatch 警示。 若要建立事件通知，請參閱文件中的建立對 CloudWatch 事件觸發的事 CloudWatch 件規則。	AWS 管理員

將複本新增至與 Aurora PostgreSQL 相容的資料庫叢集

任務	描述	所需技能
將僅供讀取複本新增至叢集。	1. 建立僅供讀取複本。	AWS 管理員

任務	描述	所需技能
	2. 將僅供讀取複本新增至資料庫叢集所在的相同可用區域。備註：如果您的容錯移轉節點必須符合需求，則可以使用不同的可用區域。	
請注意僅供讀取複本端點。	記錄您的僅供讀取複本端點，以便稍後用於建立自訂端點。	AWS 管理員

建立自訂端點

任務	描述	所需技能
輸入自訂端點的名稱。	針對您需要的每個端點，建立與工作負載或應用程式相關的唯一端點名稱。	AWS 管理員
新增端點成員。	將僅供讀取複本端點新增至自訂群組。如需詳細資訊，請參閱 Amazon RDS 文件中的編輯自訂端點。	AWS 管理員
(選擇性) 將 future 的執行個體新增至叢集。	如果您想要將更多複本或端點新增到自訂群組，請參閱 Amazon RDS 說明文件中的將 Aurora 複本新增至資料庫叢集。	AWS 管理員
建立端點。	若要建立端點，請參閱 Amazon RDS 文件中的建立自訂端點。	AWS 管理員

使用自訂端點測試應用程式連線

任務	描述	所需技能
與指向工作負載的應用程式共用自訂端點詳細資料。	將您的自訂端點詳細資料新增至您計劃測試之報表應用程式中的資料庫連線詳細資料。	AWS 管理員
使用自訂端點 Connect 工作負載。	驗證報告應用程式中的自訂端點詳細資料。	AWS 管理員
檢查資料庫中的連線詳細資訊。	1. 測試應用程式的使用者名稱和連線計數。 2. 檢查工作負載之間的負載平衡，以確保連線分散在不同的自訂端點 (主要和僅供讀取複本)。	AWS 管理員

相關資源

- Aurora 端點的類型
- 自訂端點的成員資格規則
- E 適用於自訂端點的 nd-to-end AWS CLI 範例
- Amazon Aurora 作為甲骨文 RAC 的替代品
- 從甲骨文遷移到 PostgreSQL 的挑戰以及如何克服它們

在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線

創建者：羅希特·卡普爾 (AWS)

環境：PoC 或試點	技術：資料庫；網路；安全性、身分識別、合規性	工作負載：開源
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

Amazon Relational Database Service 服務 (Amazon RDS) 支援 PostgreSQL 資料庫執行個體使用 SSL 加密。您可以使用 SSL 加密應用程式和 Amazon RDS 適用於 PostgreSQL 資料庫執行個體之間的 PostgreSQL 連線。根據預設，亞馬遜 RDS 版使用 SSL/TLS，並預期所有用戶端都能使用 SSL/TLS 加密進行連線。Amazon RDS for PostgreSQL TLS 版本 1.1 和 1.2 版。

此模式說明如何為 Amazon RDS for PostgreSQL 資料庫執行個體啟用加密連線。您可以使用相同的程序為 Amazon Aurora PostgreSQL 相容版本啟用加密連線。

先決條件和限制

- 有效的 AWS 帳戶
- 一個亞馬遜 RDS 資料庫執行個體
- 一個 SSL 服務包

架構

工具

- pgAdmin 是一個開放原始碼的管理和開發平 PostgreSQL。您可以在 Linux、Unix、macOS 和視窗上使用 PgAdmin pgAdmin 來管理您在 PostgreSQL 10 及更新版本中的資料庫物件。
- PostgreSQL 編輯器提供更易於使用的介面，可協助您建立、開發和執行查詢，以及根據您的需求編輯程式碼。

最佳實務

- 監視不安全的資料庫連線。
- 稽核資料庫存取權限。
- 確保備份和快照在靜態時加密。
- 監控資料庫存取。
- 避免不受限制的存取群組。
- 使用 Amazon 增強您的通知 GuardDuty。
- 定期監控政策遵守情況。

史诗

下載受信任憑證並將其匯入您的信任存放區

任務	描述	所需技能
將受信任的憑證載入您的電腦。	若要將憑證新增至您電腦的受信任的根憑證授權單位存放區，請依照下列步驟執行。（這些說明使用窗口服務器作為一個例子。） 1. 在 [Windows 伺服器] 中，選擇 [開始]、[執行]，然後輸入 mmc。 2. 在主控台中，選擇 [檔案] > [新增/移除嵌入式管 3. 在 [可用嵌入式管理單元] 下，選擇 [憑證]，然後選擇 [新增] 4. 在 [這個嵌入式管理單元將永遠管理憑證] 下方，選擇 [電腦帳戶]、[下 5. 選擇本機電腦，完成。	DevOps 工程師，移民工程師，DBA

任務	描述	所需技能
	6. 如果您沒有其他嵌入式管理單元可新增至主控台，請選擇 [確定]。 7. 在主控台樹狀目錄中，按兩下憑證。 8. 在「受信任的根憑證授權單位 9. 選擇 [所有工作] > [匯入] 以匯入下載的憑證。 10請依照「憑證匯入精靈」中的步驟執行。	

強制使用 SSL 連線

任務	描述	所需技能
建立參數群組並設定 rds.force_ssl 參數。	如果 PostgreSQL 資料庫執行個體具有自訂參數群組，請編輯參數群組並變更`rds.force_ssl` 為 1。 如果資料庫執行個體使用未`rds.force_ssl` 啟用的預設參數群組，請建立新參數群組。您可以使用 Amazon RDS API 修改新參數群組，也可以按照以下指示手動修改新參數群組。 若要建立新參數群組： 1. 登入 AWS 管理主控台，然後針對託管資料庫執	DevOps 工程師，移民工程師，DBA

任務	描述	所需技能
	行個體的 AWS 區域開啟 Amazon RDS 主控台。 2. 在導覽窗格中，選擇 Parameter groups (參數群組)。 3. 選擇「建立參數群組」，然後設定下列值： • 選擇「參數群組族群」做為「參數群組」14。 • 針對群組名稱，請輸入 pgsql-SSL <database _instance>。 • 在「描述」中，輸入要新增之參數群組的任意格式描述。 • 選擇建立。 4. 選擇您建立的參數群組。 5. 從 Parameter group actions (參數群組動作)，選擇 Edit (編輯)。 6. 尋找並將其設定變更為 1。 注意：變更此參數之前，請先進行用戶端測試。 7. 選擇儲存變更。 若要將參數群組與 PostgreSQL 資料庫執行個體建立關聯，請執行 1. 在 Amazon RDS 主控台的導覽窗格中，選擇「資料	

任務	描述	所需技能
	庫」，然後選擇 PostgreSQL 資料庫執行個體。 2. 選擇 Modify (修改)。 3. 在「其他規劃」下，選擇新參數群組，然後選擇「繼續」。 4. 在排程修改下，選擇立即套用。 5. 選擇 Modify DB instance (修改資料庫執行個體)。 如需詳細資訊，請參閱 Amazon RDS 文件。	
強制 SSL 連線。	Connect 至 Amazon RDS for PostgreSQL 的資料庫執行個體。不使用 SSL 的連線嘗試會遭到拒絕，並顯示錯誤訊息。如需詳細資訊，請參閱 Amazon RDS 文件。	DevOps 工程師，移民工程師，DBA

安裝 SSL 擴充功能

任務	描述	所需技能
安裝 SSL 擴充功能。	1. 啟動一個 psql 或 pgAdmin 連接作為 DBA。 2. 呼叫 ssl_is_use () 函數來判斷是否正在使用 SSL。 `select ssl_is_used();`	DevOps 工程師，移民工程師，DBA

任務	描述	所需技能
	該函數返回，t如果連接使用 SSL; 否則返回f。 3. 安裝 SSL 擴充功能。 `create extension sslinfo;` `show ssl;` `select ssl_cipher();` 如需詳細資訊，請參閱 Amazon RDS 文件。	

為 SSL 設定您 PostgreSQL 用戶端

任務	描述	所需技能
設定 SSL 的用戶端。	透過使用 SSL，您可以啟動 PostgreSQL 伺服器，並支援使用 TLS 通訊協定的加密連線。伺服器會監聽同一個 TCP 連接埠上的標準和 SSL 連線，並與任何連線用戶端進行協商，以了解是否使用 SSL。依預設，這是用戶端選項。 如果您使用的是 psql 客戶端： 1. 請確定 Amazon RDS 憑證已載入您的本機電腦。 2. 新增下列項目以啟動 SSL 用戶端連線： `psql postgres -h SOMEHOST.amazonaws`	DevOps 工程師，移民工程師，DBA

任務	描述	所需技能
	.com -p 8192 -U someuser sslmode=v erify-full sslrootce rt=rds-ssl-ca-cert .pem select ssl_cipher(); 對於其他用戶端： • 修改相應的應用程式公開金鑰參數。這可能是作為一個選項，作為連接字符串的一部分，或者作為 GUI 工具連接頁面上的屬性。 檢閱這些用戶端的下列頁面： • pgAdmin 文檔 • JDBC 文件	

故障診斷

問題	解決方案
無法下載 SSL 憑證。	請檢查您與網站的連線，然後重試將憑證下載到您的本機電腦。

相關資源

- Amazon RDS for PostgreSQL 件
- 將 SSL 與 PostgreSQL 資料庫執行個體搭配使用 (Amazon RDS 文件)
- 使用 SSL 安全的 TCP/IP 連 PostgreSQL 件集)
- 使用 SSL (JDBC 文件)

加密現有的亞馬遜 RDS 資料庫執行個體

創建者：皮尤什·戈亞爾（AWS），肖巴納拉古（AWS）和亞斯拉賈（AWS）

環境：生產	技術：資料庫；安全性、身分識別、合規性	AWS 服務：Amazon RDS; AWS 公司; AWS DMS

Summary

此模式說明如何在亞馬遜網路服務 (AWS) 雲端中為 PostgreSQL 資料庫執行個體加密現有的 Amazon 關聯式資料庫服務 (Amazon RDS)，並將停機時間降至最低。此程序也適 Amazon RDS for MySQL 資料庫執行個體。

您可以在建立 Amazon RDS 資料庫執行個體時為其啟用加密，但在建立之後則不能啟用加密。不過，您可以建立資料庫執行個體的快照，然後建立該快照的加密副本，將加密新增至未加密的資料庫執行個體。然後，您可以從加密的快照還原資料庫執行個體，以取得原始資料庫執行個體的加密副本。如果您的項目允許在此活動期間停機（至少對於寫入事務），那麼這就是您需要做的。當資料庫執行個體的新加密副本可用時，您可以將應用程式指向新的資料庫。但是，如果您的項目不允許此活動的重大停機時間，則需要一種有助於最大程度地減少停機時間的替代方法。此模式使用 AWS Database Migration Service (AWS DMS) 來遷移和持續複寫資料，以便在最短的停機時間內完成切換到新的加密資料庫。

Amazon RDS 加密資料庫執行個體使用業界標準 AES-256 加密演算法，在託管 Amazon RDS 資料庫執行個體的伺服器上加密您的資料。加密資料後，Amazon RDS 會以透明方式處理資料的存取和解密身份驗證，對效能的影響降到最低。您不需要修改資料庫用戶端應用程式即可使用加密。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 未加密的 Amazon RDS for PostgreSQL 行個體
- 使用 (建立、修改或停止) AWS DMS 任務的經驗 (請參閱 AWS DMS 文件中的使用 AWS DMS 任務)
- 熟悉 AWS Key Management Service (AWS KMS) 以加密資料庫 (請參閱 AWS KMS 文件)

限制

- 您只能在建立 Amazon RDS 資料庫執行個體時啟用加密，而不能在建立資料庫執行個體之後啟用加密。
- 未記錄資料表中的資料將不會使用快照還原。如需詳細資訊，請參閱使用 PostgreSQL 的最佳做法。
- 未加密資料庫執行個體不可以有加密僅供讀取複本，加密資料庫執行個體也不可以有未加密僅供讀取複本。
- 您無法將未加密的備份或快照還原至已加密的資料庫執行個體。
- AWS DMS 不會自動傳輸序列，因此需要額外的步驟來處理此問題。

如需詳細資訊，請參閱 Amazon RDS 文件中的 Amazon RDS 加密資料庫執行個體限制。

架構

來源架構

- 未加密 RDS 資料庫執行個體

目標架構

- 加密 RDS 資料庫執行個
  - 目的地 RDS 資料庫執行個體是透過還原來源 RDS 資料庫執行個體的資料庫快照複本建立的。
  - 還原快照時會使用 AWS KMS 金鑰進行加密。
  - 使用 AWS DMS 複寫任務來遷移資料。

工具

用於啟用加密的工具：

- 用於加密的 AWS KMS 金鑰 — 建立加密的資料庫執行個體時，您可以選擇客戶受管金鑰或 Amazon RDS 的 AWS 受管金鑰來加密資料庫執行個體。如果您沒有為客戶受管金鑰指定金鑰識別碼，Amazon RDS 會將 AWS 受管金鑰用於您的新資料庫執行個體。Amazon RDS 為您的 AWS 帳戶為 Amazon RDS 創建一個 AWS 受管金鑰。您的 AWS 帳戶在每個 AWS 區域都有不同的 AWS 受管金鑰，適用於 Amazon RDS。如需使用 KMS 金鑰進行 Amazon RDS 加密的詳細資訊，請參閱加密 Amazon RDS 資源。

用於進行中複寫的工具：

- AWS DMS — 您可以使用 AWS Database Migration Service (AWS DMS) 將變更從來源資料庫複寫到目標資料庫。重要的是要保持源和目標數據庫的同步，以將停機時間降到最低。如需設定 AWS DMS 和建立任務的相關資訊，請參閱 AWS DMS 文件。

史诗

建立來源資料庫執行個體的快照並加密

任務	描述	所需技能
檢查來源 PostgreSQL 資料庫執行個體的詳細資料。	在 Amazon RDS 主控台上，選擇來源 PostgreSQL 資料庫執行個體。在 [設定] 索引標籤上，確定未啟用執行個體的加密功能。如需螢幕圖例，請參閱「其他資訊」一節。	DBA
建立資料庫快照集。	建立要加密之執行個體的資料庫快照。建立快照所需的時間取決於資料庫的大小。如需指示，請參閱 Amazon RDS 文件中的建立資料庫快照。	DBA
加密快照。	在 Amazon RDS 主控台導覽窗格中，選擇快照，然後選取您建立的資料庫快照。針對 Actions (動作) 選擇 Copy Snapshot (複製快照)。在對應欄位中提供目的地 AWS 區域和資料庫快照副本的名稱。選取「啟用加密」核取方塊。在 Master Key (主金鑰) 中，指定用來加密資料庫快照副本的 KMS 金鑰識別符。選擇 Copy Snapshot (複製快照)。如需詳	DBA

任務	描述	所需技能
	細資訊，請參閱 Amazon RDS 文件中的複製快照。	

準備目標資料庫執行個體

任務	描述	所需技能
還原資料庫快照。	在 Amazon RDS 主控台上，選擇快照。選擇您建立的加密快照。針對 Actions (動作)，選擇 Restore Snapshot (還原快照)。對於資料庫執行個體識別碼，請為新資料庫執行個體提供唯一的名稱。檢閱執行個體詳細資訊，然後選擇還原資料庫執行個體。將從您的快照建立新的加密資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 文件中的從資料庫快照還原。	DBA
使用 AWS DMS 遷移資料。	在 AWS DMS 主控台上，建立 AWS DMS 任務。對於移轉類型，請選擇移轉現有資料並複寫進行中的變更。在「工作設定」 中，對於「目標」表格準備模式，選擇「截斷」。如需詳細資訊，請參閱 AWS DMS 文件中的建立任務。	DBA
啟用資料驗證。	在 [工作設定] 中選擇 [啟用驗證]。這可讓您將來源資料與目標資料進行比較，以確認資料是否已正確移轉。	DBA

任務	描述	所需技能
停用目標資料庫執行個體的限制。	停用目標資料庫執行個體上的任何觸發程序和外部索引鍵限制，然後啟動 AWS DMS 任務。如需停用觸發程序和外部索引鍵限制的詳細資訊，請參閱 AWS DMS 文件。	DBA
驗證資料。	完成滿載後，請驗證目標資料庫執行個體上的資料，以查看其是否與來源資料相符。如需詳細資訊，請參閱 AWS DMS 文件中的 AWS DMS 資料驗證。	DBA

切斷至目標資料庫執行個體

任務	描述	所需技能
停止來源資料庫執行個體的寫入作業。	停止來源資料庫執行個體的寫入作業，以便開始應用程式停機時間。確認 AWS DMS 已完成管道中資料的複寫。在目標資料庫執行個體上啟用觸發器和外部索引鍵。	DBA
更新資料庫序列	如果來源資料庫包含任何序號，請驗證並更新目標資料庫中的序列。	DBA
設定應用程式端點。	將您的應用程式連線設定為使用新的 Amazon RDS 資料庫執行個體端點。資料庫執行個體現已加密。	DBA，應用程式擁有者

相關資源

- 建立 AWS DMS 任務
- 使用 Amazon 監控複寫任務 CloudWatch
- 監控 AWS DMS 任務
- 更新 Amazon RDS 加密金鑰

其他資訊

檢查來源 PostgreSQL 資料庫執行個體的加密：

此模式的其他注意事項：

- 透過將`rds.logical_replication`參數設定為 1，啟用 PostgreSQL 上的複寫。

重要注意事項：複寫插槽會保留預先寫入日誌 (WAL) 檔案，直到檔案被外部使用為止 (例如透`pg_recvlogical`過擷取、轉換和載入 (ETL) 工作；或透過 AWS DMS。當您將`rds.logical_replication`參數值設定為 1 時，AWS DMS 會設定`wal_level`、`max_wal_senders`、`max_replication_slots`、和`max_connections`參數。如果存在邏輯複寫插槽，但複寫插槽所保留的 WAL 檔案沒有用戶，您可能會看到交易記錄磁碟使用量增加，並且可用儲存空間不斷減少。如需解決此問題的詳細資訊和步驟，請參閱如何識別造成 Amazon RDS for PostgreSQL 上出現「裝置上沒有剩餘空間DiskFull」或「錯誤的原因？ 在 AWS Support 知識中心。

- 您在建立資料庫快照集之後對來源資料庫執行個體所做的任何結構描述變更都不會出現在目標資料庫執行個體上。
- 建立加密的資料庫執行個體後，就無法變更該資料庫執行個體使用的 KMS 金鑰。在建立加密的資料庫執行個體之前，請務必確定您的 KMS 金鑰需求。
- 在執行 AWS DMS 任務之前，您必須停用目標資料庫執行個體上的觸發器和外部索引鍵。您可以在任務完成時重新啟用這些功能。

啟動時強制執行 Amazon RDS 資料庫的自動標記

環境：生產	技術：資料庫；雲端原生；安全性、身分識別、合規性	AWS 服務：Amazon RDS; Amazon SNS; AWS CloudTrail; Amazon CloudWatch

Summary

Amazon Relational Database Service 服務 (Amazon RDS) 是一種 Web 服務，可讓您更輕鬆地在 Amazon Web Services (AWS) 雲端中設定、操作和擴展關聯式資料庫。其能為產業標準的關聯式資料庫提供具成本效益、可調整大小的容量，並管理常見的資料庫管理任務。

您可以使用標記以不同的方式對 AWS 資源進行分類。當您的帳戶中有許多資源，並且想要根據標籤快速識別特定資源時，關聯式資料庫標記非常有用。您可以使用 Amazon RDS 標籤將自訂中繼資料新增至 RDS 資料庫執行個體。標籤由使用者定義的索引鍵和值組成。我們建議您建立一組一致的標籤，以符合組織的需求。

此模式提供 AWS CloudFormation 範本，可協助您監控和標記 RDS 資料庫執行個體。範本會建立監視 AWS 建 CloudTrail CreateDB Instance 個體 CloudWatch 事件的 Amazon 活動事件。(CloudTrail 擷取 Amazon RDS 的 API 呼叫做為事件。) 偵測到此事件時，它會呼叫 AWS Lambda 函數，該函數會自動套用您定義的標籤金鑰和值。該模板還通過使用亞馬遜簡單通知服務（Amazon SNS）發送有關實例已標記的通知。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 用於上傳 Lambda 程式碼的 Amazon Simple Storage Service (Amazon S3) 儲存貯體。
- 您希望接收標記通知的電子郵件地址。

限制

- 該解決方案支持 CloudTrail CreateDB Instance 事件。它不會為任何其他事件建立通知。

架構

工作流架構

自動化和規模

- 您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需在每個區域或帳戶中執行一次範本。

工具

AWS 服務

- AWS CloudTrail — AWS CloudTrail 是一項 AWS 服務，可協助您對 AWS 帳戶進行管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。 CloudWatch 事件會在發生作業變更時瞭解作業變更，並視需要採取更正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而不需要佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種網路服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。

Code

該模式包括一個包含兩個文件的附件：

- `index.zip`是包含此模式之 Lambda 程式碼的壓縮檔案。
- `rds.yaml`是部署 Lambda 程式碼的 CloudFormation 範本。

有關如何使用這些文件的信息，請參見 Epics 部分。

史诗

部署 Lambda 程式碼

任務	描述	所需技能
將代碼上傳到 S3 存儲桶。	建立新的 S3 儲存貯體，或使用現有的 S3 儲存貯體上傳附加的`index.zip` 檔案 (Lambda 程式碼)。此儲存貯體必須與您要監控的資源 (RDS 資料庫執行個體) 位於相同的 AWS 區域。	雲端架構師
部署 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 Cloudformation 主控台，然後部署附件中提供的`rds.yaml`檔案。在下一個史詩中，提供模板參數的值。	雲端架構師

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選取的 S3 儲存貯體的名稱。此 S3 儲存貯體包含 Lambda 程式碼的 .zip 檔案，且必須與您要監控的 CloudFormation 範本和 RDS 資料庫執行個體位於相同的 AWS 區域。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如，`index.zip`	雲端架構師

任務	描述	所需技能
	或controls/index.zip)。	
提供電子郵件地址。	提供您要接收違規通知的作用中電子郵件地址。	雲端架構師
指定記錄日誌層級。	指定記錄日誌層級和詳細資訊。 Info指定應用程式進度的詳細資訊訊息，應該只用於偵錯。 Error指定仍然允許應用程式繼續執行的錯誤事件。 Warning指定潛在的有害情況。	雲端架構師
輸入 RDS 資料庫執行個體的標籤金鑰和值。	輸入要自動套用至 RDS 執行個體的必要標籤金鑰和值。如需詳細資訊，請參閱 AWS 文件中的標記 Amazon RDS 資源。	雲端架構師

確認訂閱

任務	描述	所需技能
確認電子郵件訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件訊息傳送至您提供的電子郵件地址。若要在執行個體標記時接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 建立儲存貯體 (Amazon S3 文件)
- 標記 Amazon RDS 資源（Amazon Aurora 文檔）

- 上傳物件 (Amazon S3 文件)
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail (Amazon CloudWatch 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

估算隨需容量的 DynamoDB 表格的成本

環境：生產	技術：資料庫、雲端原生、無伺服器、成本管理	AWS 服務：Amazon DynamoDB

Summary

Amazon DynamoDB 是一種 NoSQL 交易資料庫，即使在 PB 規模下也能提供 10 毫秒的延遲。此 Amazon Web Services (AWS) 無伺服器產品因其一致的效能和可擴展性而越來越受歡迎。 您不需要佈建基礎結構。您的單一資料表可以成長到 PB。

使用隨需容量模式時，您可以按請求支付應用程式在資料表上執行的資料讀取和寫入費用。AWS 費用是根據一個月累積的讀取請求單位 (RRU) 和寫入請求單位 (WRU) 計算。DynamoDB 會在整個月持續監控表格的大小，以決定您的儲存費用。它支持使用 point-in-time-recovery （ PITR ）的連續備份。DynamoDB 會在整個月持續監控已啟用 PITR 的表格大小，以決定您的備份費用。

若要估算專案的 DynamoDB 成本，請務必計算在產品生命週期的不同階段耗用多少 RRU、WRU 和儲存體。對於粗略的成本估算，您可以使用 AWS 定價計算器，但是您必須為表格提供大約數量的 RRU、WRU 和儲存需求。這些可能很難在項目開始時估計。AWS 定價計算器不會考慮資料成長率或項目大小，而且不會分別考慮基礎資料表和全域次要索引 (GSI) 的讀取和寫入次數。若要使用 AWS 定價計算器，您必須估算所有這些方面，以假設 WRU、RRU 和儲存大小的棒球場數字，以取得成本估算。

此模式提供機制和可重複使用的 Microsoft Excel 範本，以估算隨需容量模式的基本 DynamoDB 成本因素，例如寫入、讀取、儲存、備份和回復成本。它比 AWS 定價計算器更精細，而且會獨立考量基礎資料表和 GSI 需求。它還考慮了每月項目數據的增長率，並預測三年的成本。

先決條件和限制

先決條件

- 有關 DynamoDB 資料模型設計的基本知識
- DynamoDB 定價、WRU、RRU、儲存以及備份與復原的基本知識 (如需詳細資訊，請參閱隨需容量的定價)
- 瞭解 DynamoDB 中的資料、資料模型和項目大小

- 有關動 DynamoDB SI 的知識

限制

- 範本為您提供近似計算，但並不適用於所有組態。若要取得更準確的估計值，您必須測量基底資料表和 GSI 中每個項目的個別項目大小。
- 為了更準確的估計，您必須考慮平均月份中每個項目的預期寫入次數 (插入、更新和刪除) 和讀取次數。
- 此模式支援根據固定資料成長假設，估算未來幾年僅寫入、讀取、儲存以及備份和復原成本。

工具

AWS 服務

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。

其他工具

- AWS 定價計算器是一種基於 Web 的規劃工具，可用於為 AWS 使用案例建立估算值。

最佳實務

若要協助降低成本，請考慮下列 DynamoDB 設計最佳實務。

- 分割區索引鍵設計 – 使用高基數分割區索引鍵來均勻分配負載。
- 鄰接表設計模式-使用此設計模式進行管理 one-to-many 和 many-to-many 關係。
- 稀鬆索引 – 對 GSI 使用稀鬆索引。在您建立 GSI 時，指定一個分割區索引鍵和 (選用) 一個排序索引鍵。只有在基本資料表中包含對應 GSI 分割區索引鍵的項目才會出現在稀疏索引中。這有助於保持 GSI 更小。
- 索引過載 – 使用相同的 GSI 對各種類型的項目編製索引。
- GSI 寫入碎片 – 明智地進行碎片以跨分割區分佈資料，以實現高效、更快的查詢。
- 大型項目 – 僅將中繼資料儲存在表內，將 Blob 儲存在 Amazon S3 中，並將參考保留在 DynamoDB 中。將大型項目分解為多個項目，並使用排序索引鍵有效率地編製索引。

如需更多設計最佳實務，請參閱《Amazon DynamoDB 開發人員指南》。

史诗

從您的 DynamoDB 資料模型擷取項目資訊

任務	描述	所需技能
取得項目大小。	1. 檢查您要在桌子中存儲多少種不同類型的項目。 2. 若要計算每個項目的大小 (以 KB 為單位)，請新增每個屬性的 [索引鍵] 和 [值] 大小。 3. 計算基礎表格和每個 GSI 的項目大小。	數據工程師
估計寫入成本。	若要估計隨需容量模式下的寫入成本，首先您必須測量一個月內將使用多少 WRU。為此，您需要考慮以下因素： • 一個月內每個項目的建立、更新和刪除作業數目。 • 可用 GSI 的數量。獨立考慮每個索引。 • 索引項目的平均大小 • 索引上的同步處理次數 • 每個月會在表格中新增多少項新物件 (例如元件或產品)？ 每個月新增的項目數量可能會有所不同，但您可以根據您的業務案例假設平均增長率。 如需詳細資訊，請參閱其他資訊一節。	數據工程師

任務	描述	所需技能
估計讀取成本。	若要估算隨選模式下的讀取成本，首先您必須測量一個月將使用多少 RRU。為此，您需要考慮以下因素： • 可用 GSI 的數量。獨立考慮每個索引。 • 索引項目的平均大小 • 每個產品每月的平均讀取次數。 • DynamoDB 表格中可用物件 (元件或產品) 的總數。	資料工程師、App 開發人員

任務	描述	所需技能
估計儲存體大小和成本。	首先，根據表格中的項目大小估計每月平均儲存需求。然後，將儲存大小乘以 AWS 區域的每 GB 儲存價格來計算儲存成本。 如果您已輸入用於估算寫入成本的資料，則不需要再次輸入資料來計算儲存空間大小。否則，要估計存儲大小，您需要考慮以下因素： • 根據您的表格設計，模組 (產品) 中的資料項目數目。 • 平均項目大小 (以 KB 為單位)。 • 可用 GSI 的數量。獨立考慮每個索引。 • 索引項目的平均大小 • 每個月會在表格中新增多少個新產品？ 每個月的新產品數量可能不同，但您可以根據您的業務案例假設平均增長率。此範例平均每月使用 1000 萬個新產品。	數據工程師

在 Excel 模板中輸入項目和對象信息

任務	描述	所需技能
從「附件」部分下載 Excel 模板，然後根據您的用例表進行調整。	1. 下載 Excel 模板。 2. 根據您的表格設計調整業務模組和 GSI。	數據工程師

任務	描述	所需技能
在 Excel 範本中輸入資訊。	1. 更新圖紙中的項目資訊。僅在橙色儲存格中更新資料。 2. 調整對象編號：每個月可以添加到表中多少？ 3. 為您的 AWS 區域更新每百萬個 WRU 和 RRU 價格。 4. 更新 AWS 區域每月每 GB 的儲存和備份價格。 5. 更新 AWS 區域每 GB 的復原價格。 在範本中，有三個項目或實體：資訊、中繼資料和關係。有兩種 GSI。針對您的使用案例，如果您需要更多項目，請建立新列。如果您需要更多 GSI，請複製現有的 GSI 區塊，然後貼上以根據需要建立任意數量的 GSI 區塊。然後調整「總和」和「總計」列計算。	數據工程師

相關資源

參考

- 適用於隨需容量的 Amazon DynamoDB 定價
- 適用於 AWS 定價計算器
- 使用 DynamoDB 進行設計和架構的最佳實務
- DynamoDB 入門

指南和模式

- 使用 Amazon DynamoDB 擬資料
- 估算 Amazon DynamoDB 表格的儲存成本

其他資訊

寫入成本計算範例

DynamoDB 資料模型設計顯示一個產品的三個項目，平均項目大小為 4 KB。當您將新產品新增至 DynamoDB 基底資料表時，它會消耗項目數 * (項目大小/1 KB 寫入單位) = 3 * (4/1) = 12 WRU。在此範例中，對於寫入 1 KB，產品會耗用 1 WRU。

閱讀成本計算範例

要獲得 RRU 估計，請考慮一個月內每個項目將讀取多少次的平均值。例如，資訊項目平均會在一個月讀取 10 次，而中繼資料項目會讀取兩次，而關係項目會讀取五次。在範例範本中，所有元件的 RRU 總計 = 每月建立的新元件數 * 每月每個元件的 RRU = 1 千萬 * 17 RRU = 每月 1.7 億 RRU。

每個月，新的東西（組件或產品將被添加，並且產品總數將隨著時間的推移而增長。因此，RRU 要求也會隨著時間的推移而增長。

- 對於第一個月 RRU，消費將是 1.7 億。
- 第二個月，RRU 的消費將為 2* 1.7 億 = 3.4 億。
- 對於第三個月 RRU 消費將是 3* 1.7 億 = 510 百萬。

下圖顯示每月 RRU 耗用量與成本預測。

請注意，圖表中的價格僅供說明之用。若要針對您的使用案例建立準確的預測，請查看 AWS 定價頁面，並在 Excel 工作表中使用這些價格。

儲存、備份及回復成本計算範例

DynamoDB 儲存、備份和還原都會彼此連接。Backup 與存儲直接連接，恢復與備份大小直接連接。隨著資料表大小的增加，對應的儲存、備份和還原成本將按比例增加。

儲存空間大小和成本

儲存成本會根據您的資料成長率隨時間增加。例如，假設基底資料表和 GSI 中元件或產品的平均大小為 11 KB，而每個月將會在資料庫表格中新增一千萬個新產品。在這種情況下，您的 DynamoDB 資料

表大小將會增加 (11 KB * 10 百萬) /1024/1024 = 每月 105 GB。在第一個月，您的表格存儲大小將是 105 GB，在第二個月它將是 105 + 105 = 210 GB，依此類推。

- 第一個月的儲存成本為您 AWS 區域的每 GB 105 GB * 儲存價格。
- 第二個月，您所在區域的儲存空間成本為每 GB 210 GB * 儲存空間價格。
- 第三個月，您所在區域的儲存空間成本為每 GB 315 GB * 儲存空間價格。

如需未來三年的儲存空間大小和成本，請參閱儲存空間大小與預測一節。

Backup 成本

Backup 成本會根據您的資料成長率隨時間增加。當您使用 point-in-time-recovery (PITR) 開啟連續備份時，連續備份費用會根據每月儲存 GB 的平均值計算。在日曆月份，平均備份大小將與您的表格存儲大小相同，儘管實際大小可能會有所不同。由於每個月都會新增新產品，因此總儲存大小和備份大小會隨著時間的推移而增加。例如，在第一個月，105 GB 的平均備份大小可能會在第二個月增加到 210 GB。

- 第一個月的備份費用為您 AWS 區域的每 GB 105 GB * 持續備份價格。
- 第二個月的備份費用為您區域的每月 210 GB * 持續備份價格 (每 GB)。
- 第三個月的備份費用為您所在區域的每 GB 每月 315 GB * 持續備份價格。
- 以及，依此類推

Backup 成本包含在「儲存體大小與成本預測」區段的圖表中。

回收成本

在啟用 PITR 的情況下進行連續備份時，復原作業費用會根據還原的大小計算。每次還原時，您都會根據已還原的資料 GB 付費。如果您的表格大小很大，並且您在一個月內執行多次還原，則會很昂貴。

為了預估還原成本，此範例假設您每月在月底執行一次 PITR 復原。此範例使用每月平均備份大小作為該月的還原資料大小。第一個月的平均備份大小為 105 GB，而對於月底的復原，還原資料大小為 105 GB。第二個月，它將是 210 GB，依此類推。

根據您的資料成長率，回復成本會隨著時間的推移而增加。

- 第一個月的復原成本為您 AWS 區域的每 GB 105 GB * 還原價格。
- 第二個月的回復成本為您所在地區的每 GB 210 GB * 還原價格。
- 第三個月的回復成本為您所在地區的每 GB 315 GB * 還原價格。

如需詳細資訊，請參閱 Excel 範本中的 [儲存、備份和復原] 索引標籤和下一節中的圖形。

儲存空間大小和成本預測

在範本中，標準資料表類別每月減去免費方案 25 GB 的實際可計費儲存體大小來計算。在工作表中，您將獲得一個分為每月值的預測圖。

下列範例圖表會預測每月儲存體大小 (GB)、計費儲存成本、隨需備份成本，以及未來 36 個日曆月的回復成本。所有費用均以美元計算。從圖表中可以看出，儲存、備份和復原成本會隨著儲存體大小的增加而成比例增加。

請注意，圖表中使用的價格僅用於說明目的。若要為您的使用案例建立準確的價格，請查看 AWS 定價頁面，並在 Excel 範本中使用這些價格。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

估算 Amazon DynamoDB 表格的儲存成本

創建者穆努爾·阿爾馬蒙

環境：PoC 或試點	技術：資料庫、大數據、成本管理、儲存與備份	AWS 服務：Amazon DynamoDB

Summary

Amazon DynamoDB 是一種 NoSQL 交易資料庫，即使在 PB 規模下也能提供 10 毫秒的延遲。此 Amazon Web Services (AWS) 無伺服器產品因其一致的效能和可擴展性而越來越受歡迎。您不需要佈建儲存空間。您的單一資料表可以成長到 PB。

DynamoDB 會在整個月持續監控表格的大小，以決定您的儲存費用。然後，AWS 會向您收取平均儲存大小 (以 GB 為單位) 的費用。您的資料表隨著時間的推移成長越多，儲存成本就越大。若要計算儲存成本，您可以使用 AWS 定價計算器，但您需要提供資料表的大約大小，包括全域次要索引 (GSI)，在專案開始時確實很難估算。此外，AWS 定價計算器不會考慮資料成長率。

此模式提供了一種機制和可重複使用的 Microsoft Excel 範本來計算 DynamoDB 儲存體大小和成本。它會獨立考量基底資料表和 GSI 的儲存需求。它會考慮個別項目的大小以及隨時間推移的資料成長率來計算儲存體大小。

若要取得估計值，請在範本中插入兩條資訊：

- 基本資料表和 GSI 的個別項目大小 (以 KB 為單位)
- 平均一個月內可以將多少個新物件或產品新增至表格 (例如 1000 萬個)

範本會產生未來三年的儲存與成本預測圖表，如下列範例所示。

先決條件和限制

先決條件

- 有關 DynamoDB 儲存和定價的基本知識
- 瞭解 DynamoDB 中的資料、資料模型和項目大小

- 有關 DynamoDB 全域次要索引 (GSI) 的知識

限制

- 範本為您提供近似計算，但並不適用於所有組態。若要取得更準確的估計值，您必須測量基底資料表和 GSI 中每個項目的個別項目大小。
- 此模式僅支援根據固定資料成長假設來估算未來幾年的儲存大小和成本。

工具

AWS 服務

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。

其他工具

- AWS 定價計算器是一種基於 Web 的規劃工具，可用於為 AWS 使用案例建立估算值。

史诗

從您的 DynamoDB 資料模型擷取項目資訊

任務	描述	所需技能
取得項目大小。	1. 檢查您要在桌子中存儲多少種不同類型的項目。 2. 要計算每個項目的大小（以 KB 為單位），請添加每個屬性的「鍵」和「值」大小。 3. 計算基礎表格和每個 GSI 的項目大小。	數據工程師
獲取一個月內添加的對象數量。	估計平均在一個月內將多少個元件或物件新增至 DynamoDB 表格。	數據工程師

在 Excel 模板中輸入項目和對象信息

任務	描述	所需技能
從附加文檔下載 Excel 工作表，並根據您的用例表進行調整。	1. 下載 Excel 模板。 2. 根據您的表格設計調整業務模組和 GSI。	數據工程師
在 Excel 範本中輸入資訊。	1. 將項目資訊更新到圖紙中。 2. 調整對象編號：每個月可以添加到表中多少？ 3. 更新 AWS 區域每月每 GB 的儲存價格。	數據工程師

相關資源

- Amazon DynamoDB 需定價
- 適用於 AWS 定價計算器

其他資訊

請注意，附加的範本只會預測「標準」儲存表格類別的儲存大小與成本。根據儲存成本的預測，以及考量個別項目大小和產品或物件成長率，您可以估算下列項目：

- 資料匯出成本
- Backup 與回復成本
- 資料儲存需求。

Amazon DynamoDB 料儲存成本

DynamoDB 會持續監控資料表的大小，以判斷儲存費用。DynamoDB 會新增資料的原始位元組大小以及依據您啟用的功能而定的每個項目儲存額外負荷，來測量應計費資料的大小。如需詳細資訊，請參閱 DynamoDB 開發人員指南。

資料儲存的價格取決於您的表格類別。如果您使用的是 DynamoDB 標準表類別，則每個月存放的前 25 GB 是免費的。如需有關不同 AWS 區域中標準表格類別和標準不常存取表格類別的儲存成本的詳細資訊，請參閱隨需容量的定價。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWR 報告估計甲骨文資料庫的 Amazon RDS 引擎大小

由阿布舍克韋爾馬（AWS）和愛德華多·瓦倫丁（AWS）創建

環境：生產	來源：甲骨文數據庫	目標：Amazon RDS 或 Amazon Aurora
R 型：重新建築	工作量：甲骨文	技術：資料庫；移轉
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

當您將 Oracle 資料庫遷移到 Amazon Relational Database Service 服務 (Amazon RDS) 或 Amazon Aurora 時，計算目標資料庫的 CPU、記憶體和磁碟 I/O 是一項關鍵要求。您可以分析「Oracle 自動工作負載儲存區域 (AWR)」報表，來預估目標資料庫所需的容量。此模式說明如何使用 AWR 報表來估計這些值。

來源 Oracle 資料庫可以位於現場部署或託管在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上，也可以是適用於 Oracle 資料庫執行個體的 Amazon RDS。目標資料庫可以是任何 Amazon RDS 或 Aurora 資料庫。

備註：如果您的目標資料庫引擎是 Oracle，則容量預估會更精確。對於其他 Amazon RDS 資料庫，引擎大小可能會因資料庫架構的差異而有所不同。

建議您先執行效能測試，然後再移轉 Oracle 資料庫。

先決條件和限制

先決條件

- 可下載 AWR 報表的「Oracle 資料庫企業版」授權與「Oracle 診斷套件」授權。

產品版本

- 適用於版本 11g (11.2.0.3.v1 及更新版本) 以及最高至 12.2 和 18c、19 版的所有甲骨文資料庫版本。

- 此模式不包括 Oracle 工程系統或 Oracle 雲端基礎架構 (OCI)。

架構

源, 技術, 堆棧

下列其中一項：

- 內部部署 Oracle 資料庫
- EC2 執行個體上的甲骨文資料庫
- Amazon RDS for Oracle 數據庫實例

目標技術堆疊

- 任何 Amazon RDS 或 Amazon Aurora 數據庫

目標架構

如需完整遷移程序的相關資訊，請參閱使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移至 Aurora PostgreSQL 模式。

自動化和規模

如果您要遷移多個 Oracle 資料庫，並且想要使用其他效能指標，可以按照部落格文章中所述的步驟，根據 Oracle 效能指標大小調整大小大小的 Amazon RDS 執行個體來自動化處理程序。

工具

- 「Oracle 自動工作負載儲存區域 (AWR)」是 Oracle 資料庫內建的儲存區域。它會定期收集並儲存系統活動和工作負載資料，然後由「自動資料庫診斷監督器」(ADDM) 對其進行分析。AWR 會定期 (依預設，每 60 分鐘) 建立系統效能資料的快照，並儲存資訊 (依預設，最多 8 天)。 您可以使用 AWR 檢視和報表來分析此資料。

最佳實務

- 若要計算目標資料庫的資源需求，您可以使用單一 AWR 報表、多個 AWR 報表或動態 AWR 檢視表。我們建議您在尖峰負載期間使用多個 AWR 報告，以預估處理這些尖峰負載所需的資源。此外，動態檢視還提供更多資料點，協助您更精確地計算資源需求。

- 您應該只預估計劃移轉之資料庫的 IOPS，而不是其他使用該磁碟的資料庫和處理程序。
- 若要計算資料庫使用了多少 I/O，請勿使用 AWR 報表之「負載設定檔」區段中的資訊。請改為使用 [I/O 設定檔] 區段 (如果有的話)，或跳至「執行處理活動統計資料」段落，查看實體讀取和寫入作業的總值。
- 估計 CPU 使用率時，建議您使用資料庫測量結果方法而非作業系統 (OS) 統計資料，因為它是以資料庫使用的 CPU 為基礎。（操作系統統計信息還包括其他進程的 CPU 使用率。）您也應該在 ADDM 報表中檢查 CPU 相關建議，以改善移轉後的效能。
- 決定正確的執行個體類型時，請考慮特定執行個體大小的 I/O 輸送量限制 — Amazon 彈性區塊存放區 (Amazon EBS) 輸送量和網路輸送量。
- 在移轉前執行效能測試，以驗證引擎大小。

史诗

建立 AWR 報告

任務	描述	所需技能
啟用 AWR 報表。	若要啟用報表，請遵循 Oracle 說明文件中的指示。	DBA
檢查保留期。	若要檢查 AWR 報表的保留期間，請使用下列查詢。 `SQL> SELECT snap_interval,retention FROM dba_hist_wr_control;`	DBA
產生快照。	如果 AWR 快照間隔的精細度不夠，無法擷取尖峰工作負載的尖峰，您可以手動產生 AWR 報告。若要產生手動 AWR 快照，請使用下列查詢。 `SQL> EXEC dbms_workload_repository.create_snapshot;`	DBA

任務	描述	所需技能
檢查最近的快照。	若要檢查最近的 AWR 快照，請使用下列查詢。 `SQL> SELECT snap_id, to_char(begin_interval_time,'dd/MON/yy hh24:mi') Begin_Interval, to_char(end_interval_time,'dd/MON/yy hh24:mi') End_Interval FROM dba_hist_snapshot ORDER BY 1;`	DBA

估計磁碟 I/O 需求

任務	描述	所需技能
選擇一種方法。	IOPS 是儲存裝置上每秒輸入和輸出操作的標準測量方法，包括讀取和寫入作業。 如果要將現場部署資料庫遷移到 AWS，則需要判斷資料庫使用的尖峰磁碟 I/O。 您可以使用下列方法來估計目標資料庫的磁碟 I/O： • AWR 報表的載入設定檔段落 • AWR 報表的「執行處理活動統計資料」段落 (Oracle 資料庫 12c 或更新的版本請使用此段落) • AWR 報表的「I/O 設定檔」段落 (如果是 12c 之前的	DBA

任務	描述	所需技能
	「Oracle 資料庫」版本，請使用此段落) • AWR 檢視表 下列步驟說明這四種方法。	

任務	描述	所需技能
選項 1：使用負載輪廓。	下表顯示 AWR 報表之「負載設定檔」段落的範例。 重要事項：如需更準確的資訊，建議您使用選項 2 (I/O 設定檔) 或選項 3 (執行個體活動統計資料)，而非負載設定檔。 每秒 / 每宗交易 / 每位執行 / 每次通話 資料庫時間: 26.6 0.2 0.00 0.02 資料庫中央處理器： 18.0 0.1 0.00 0.01 背景中央處理器： 0.2 0.0 0.00 0.00	DBA

任務	描述		所需技能
	重做大小(位元組)：	2,458 17,09 .9	
	邏輯讀取（塊	3,37 23,4 .5	
	區塊變更：	21,6 150.5	
	物理讀取（塊	13,5 94.4	
	物理寫入（塊	3,46 24.1	
	讀取IO請求：	3,58 24.9	

任務	描述		所需技能
	寫入 IO 請求：	574.1	4.0
	讀取 IO (MB)	106.1	0.7
	寫入 IO (MB)	27.1	0.2
	IM 掃描行：	0.0	0.0
	會話邏輯讀取 IM：		
	使用者呼叫：	1,245	8.7

任務	描述	所需技能
	剖析: 4,62(32.2 硬解析（SC 8.9 0.1 SQL 工作區域 (MB) 824.9 5.7 登入： 1.7 0.0 執行 (SQL 136,(950.4 回滾： 22.9 0.2 交易： 143.8 根據這項資訊，您可以計算 IOP 和輸送量，如下所示： IOPS = 讀取 I/O 請求：+ 寫入 I/O 請求 = 輸送量 = 實體讀取 (區塊) + 實體寫入 (區塊) =	

任務	描述	所需技能
	由於 Oracle 中的區塊大小為 8 KB，因此您可以按如下方式計算總輸送量： 總輸送量 (以 MB 為單位) 為 17042.4 * 8 * 1024 警告：請勿使用負載設定檔來估計執行個體大小。它不如執行個體活動統計資料或 I/O 設定檔那麼精確。	

任務	描述	所需技能
選項 2：使用執行個體活動統計資料。	如果您使用的是 12c 之前的 Oracle 資料庫版本，可以使用 AWR 報表的「執行處理活動統計資料」區段來估計 IOPS 和輸送量。下表顯示本節的範例。 統計數字 / 總計 / 每秒 / 每次反式 物理讀取總 IO 請求 / 2,547,,217 / 3,610. / 25.11 物理讀取總字節 / 80,776,124, / 114,48 / 796,149 8 物理寫入總 IO 請求 / 534,19 08 / 757.1 / 5.27 實體寫入總位元組 / 25,517 8,849, / 36,165 1.84 / 251,508 8 根據這項資訊，您可以計算 IOPS 和輸送量的總計，如下所示：	DBA

任務	描述	所需技能
	總 IOPS 總數 = 3,610.28 + 757.11 = 4367 總兆比特 = 114,482,426.26 +	

任務	描述	所需技能
選項 3：使用 I/O 設定檔。	在 Oracle 資料庫 12c 中，AWR 報表包含「I/O 設定檔」段落，以單一表格顯示所有資訊，並提供更精確的資料庫效能資料。下表顯示本節的範例。 每秒讀取 + 寫入 / 每秒讀取 / 每秒寫入次數 請求總數： 4,367. / 3,610. / 757.1 資料庫要求： 4,161. / 3,586. / 574.7 優化請求： 0.0 / 0.0 / 0.0 重做請求： 179.3 / 2.8 / 176.6 總計 (MB): 143.7 / 109.2 / 34.5 資料庫 (MB)： 133.1 / 106.1 / 27.1	DBA

任務	描述	所需技能
	最佳化總計 (MB): 0.0 0.0 0.0 重做功能 (MB): 7.6 2.7 4.9 資料庫 (區塊) : 17,042 13,575 3,467.3 通過緩衝區緩存（塊）： 5,898. 5,360. 537.6 直接（塊） 11,143 8,214. 2,929.7 此表格提供下列輸送量和 IOPS 總數值： 輸送量 = 143 MBPS (從第五列開始，標示為「總計」，第二欄) IOPS = 4,367.4（從第一行，標記為「總請求」，第二列）	

任務	描述	所需技能
選項 4：使用 AWR 視觀表。	您可以使用 AWR 檢視來查看相同的 IOPS 和輸送量資訊。若要取得此資訊，請使用下列查詢： `break on report compute sum of Value on report select METRIC_NAME,avg(AVERAGE) as "Value" from dba_hist_sysmetric_summary where METRIC_NAME in ('Physical Read Total IO Requests Per Sec','Physical Write Total IO Requests Per Sec') group by metric_name;`	DBA

預估 CPU 需求

任務	描述	所需技能
選擇一種方法。	您可以透過三種方式估計目標資料庫所需的 CPU： • 使用處理器的實際可用核心 • 通過使用基於操作系統統計信息的使用內核 • 通過使用基於數據庫統計信息的使用內核 如果您要查看使用的核心，我們建議您使用資料庫指標方法	DBA

任務	描述	所需技能
	而非作業系統統計資料，因為它是以您計劃移轉的資料庫使用的 CPU 為基礎。（操作系統統計信息還包括其他進程的 CPU 使用率。）您也應該在 ADDM 報表中檢查 CPU 相關建議，以改善移轉後的效能。 您也可以根據 CPU 產生預估需求。如果您使用的是不同的 CPU 世代，則可以依照白皮書中的指示來預估目標資料庫所需的 CPU，以達到最佳工作負載效能的神秘面紗。	

任務	描述	所需技能
選項 1：根據可用核心預估需求。	在 AWR 報表中： • CPU 指的是邏輯和虛擬 CPU。 • 核心是實體 CPU 晶片組中的處理器數量。 • 插槽是將晶片連接至主機板的實體裝置。多核心處理器具有多個 CPU 核心的插槽。 您可以使用兩種方式估算可用的核心： • 通過使用 OS 命令 • 使用 AWR 報表 使用 OS 命令估計可用的核心 使用下列指令來計算處理器中的核心數。 `$ cat /proc/cpuinfo | grep "cpu cores"|uniq cpu cores : 4 cat /proc/cpuinfo | egrep "core id|physic al id" | tr -d "\n" | sed s/physical/\\nphys ical/g | grep -v ^$ | sort | uniq | wc -l` 使用以下命令來計算處理器中的插槽。 `grep "physical id" / proc/cpuinfo | sort -u`	DBA

任務	描述	所需技能
	physical id : 0 physical id : 1 注意：我們不建議使用作業系統命令 (例如 nmon 和 sar) 來擷取 CPU 使用率。這是因為這些計算包括其他處理序的 CPU 使用率，而且可能無法反映資料庫所使用的實際 CPU。 使用 AWR 報表預估可用核心 您也可以從 AWR 報告的第一個區段衍生 CPU 使用率。以下是報告的摘錄。 資料庫名稱 資料庫 ID 執行個體 研究所 啟動時間 發行版本 RA X <DE XX> 1 九月二十日 23:(12.1 0 NO Hos Nar (主機名稱) 平台 CPU 核心 插座 記憶體 (GB)	

任務	描述	所需技能
	<ho 64 80 80 2 441.7 e> 位 在此範例中，CPU 計數為 80，表示這些是邏輯 (虛擬) CPU。您也可以看到此組態有兩個插槽，每個插槽上有一個實體處理器 (總共兩個實體處理器)，以及每個實體處理器或插槽 40 個核心。	

<table>
<tr><th>任務</th><th>描述</th><th>所需技能</th></tr>
<tr><td>選項 2：使用作業系統統計資料預估 CPU 使用率。</td><td>您可以直接在作業系統中 (使用 sar 或其他主機作業系統公用程式) 檢查作業系統 CPU 使用率統計資料，或從 AWR 報告的「作業系統統計資料」段落檢查 IDLE/ (IDLE+BUSY) 值。您可以看到直接從 v$ osstat 消耗的 CPU 的秒數。AWR 和 Statspack 報表也會在「作業系統統計資料」段落中顯示此資料。

如果在同一個方塊上有多個資料庫，它們都具有相同的 v$osstat 值為 BUSY_TIME。

<table><tr><th>統計數字</th><th>Value</th><th>結束值</th></tr><tr><td>自由記憶體位元組</td><td>6,810,67,248</td><td>12,280,799,232</td></tr><tr><td>非作用中記憶體位元組</td><td>175,627,33,632</td><td>160,380,653,568</td></tr><tr><td>免費 _ 位元組交換</td><td>17,145,64,336</td><td>17,145,872,384</td></tr><tr><td>商務時間</td><td>1,305,56,937</td><td></td></tr></table></td><td>DBA</td></tr>
</table>

任務	描述		所需技能
	閒置時間	4,312,71 ,839	
	科威特時間	53,417,1 4	
	美好時光	29,815	
	系統時間	148,567, 70	
	使用者時間	1,146,91 ,783	
	載入	25 29	
	以位元組為單位	593,920	
	輸出位元組	327,680	
	實體記憶體位元組	474,362, 17,152	
	CPU 數	80	
	CPU 核心數	80	
	通訊端數	2	

任務	描述		所需技能
	全局接收 _ 大小 _ 最大	4194,304	
	全局發送大小 _ 最大	2,097,15	
	TCP_ 接收 _ 大小 _ 預設值	87,380	
	TCP_ 接收 _ 大小 _ 最大	6,291,45	
	TCP_ 接收 _ 大小 _ 分鐘	4,096	
	TCP_ 發送 _ 大小 _ 默認值	16,384	
	TCP_ 發送 _ 大小 _ 最大	4194,304	

任務	描述	所需技能
	傳送大小 _ 最小值 4,096 如果系統中沒有其他主要 CPU 消費者，請使用下列公式來計算 CPU 使用率的百分比： 使用率 = 忙碌時間/總時間 忙碌時間 = 要求 = V $ 操作時間 C = 總時間 (忙碌 + 閒置) C = 容量 = V $ 奧斯塔. 繁忙時間 + V $ 閒置時間 使用率 = 忙碌時間/(工作時間 + 閒置時間) = -1,305,569,937/(1,305,569,937 + 4,312,718,839) = 已使用 23%	

任務	描述	所需技能
選項 3：使用資料庫測量結果預估 CPU 使用率。	如果系統中有多個資料庫執行，您可以使用報表開頭顯示的資料庫測量結果。 		鎖點識別碼	捕捉時間	工作階段	光標/工作階段
---	---	---	---	---
開始鎖點：	1846	九月二十八日	1226	35.8
結束鎖點:	1854	十月六日至二十	1876	41.1
經過：		11,7 (分鐘)		
資料庫時間:		(分鐘)				DBA

任務	描述	所需技能
	若要取得 CPU 使用率測量結果，請使用下列公式： 資料庫 CPU 使用率 (CPU 可用電源的百分比) = CPU 時間/CPU 數量/經歷時間 其中 CPU 使用率由 CPU 時間描述，代表花在 CPU 上的時間，而不是等待 CPU 的時間。此計算結果如下： = 312,625.40/11,759.64/80 = 正在使用中央處理器的 33% 核心數目 (33%) * 80 = 26.4 個核心 核心總數 = 26.4 * (120%) = 您可以使用這兩個值中的較大值來計算 Amazon RDS 或 Aurora 資料庫執行個體的 CPU 使用率。 附註：在 IBM AIX 上，計算的使用率與作業系統或資料庫中的值不符。這些值在其他作業系統上確實相符。	

估計記憶體需求

任務	描述	所需技能
使用記憶體統計資料預估記憶體需求。	您可以使用 AWR 報表來計算來源資料庫的記憶體，並在目標資料庫中比對它。您也應該	DBA

任務	描述	所需技能
	檢查現有資料庫的效能，並降低記憶體需求以節省成本，或增加需求以提升效能。這需要對 AWR 響應時間和應用程序的服務級別協議（SLA）進行詳細分析。使用 Oracle 系統整體區域 (SGA) 與程式整體區域 (PGA) 使用量的總和，作為 Oracle 的預估記憶體使用率。為 OS 添加額外的 20% 以確定目標內存大小需求。對於 Oracle RAC，請在所有 RAC 節點上使用預估的記憶體使用率總和，並減少總記憶體，因為它儲存在一般區塊上。 1. 檢查「執行處理效率百分比」表格中的測量結果。此表格使用下列術語： • 「緩衝區命中百分比」是在緩衝區快取中找到特定區塊 (而非執行實體 I/O) 的百分比。為了獲得較佳效能，請以 100% 為目標。 • 緩衝區無等待百分比應接近 100%。 • 鎖存命中% 應接近 100%。 • % 非剖析 CPU 是非剖析活動所花費的 CPU 時間百分比。這個值應該接近 100%。.	

任務	描述	所需技能
	執行環境效率百分比 (目標 100%) 緩衝區未等待%: 99.99 重做 NoWa %: 100.00 緩衝區命中%: 99.84 記憶體內排序%: 100.00 程式庫命中率: 748.7 軟剖析%: 99.81 執行以解析% : 96.61 閂鎖命中%: 100.00 剖析 CPU 以剖 72.73 % 非剖析中 99.21	

任務	描述	所需技能
	析耗用%：央處理器： 快閃記憶體命中率：0.00 在此範例中，所有測量結果看起來都很好，因此您可以將 SGA 和 PGA 用於現有資料庫作為容量規劃需求。 2. 檢查記憶體統計資料區段並計算 SGA/PGA。 開始 結束 主機記憶體 (MB)：452,387 452,387.3 SGA 用途 (MB):220,544 220,544.0 PGA 使用36,874.9 45,270.0	

任務	描述	所需技能
	量 (MB): 使用中的執行個體記憶體總計 = SGA + PGA = 220 GB + 45 GB = 265 GB 新增 20% 的緩衝區： 執行個體記憶體總計 = 1.2 * 265 GB = 318 GB 由於 SGA 和 PGA 佔主機記憶體的 70%，因此總記憶體需求為： 主機記憶體總計 = 318/0.7 = 464 GB 附註：移轉至 Amazon RDS for Oracle 文時，會根據預先定義的公式預先計算 PGA 和 SGA。確保預先計算的值接近您的估計值。	

判斷目標資料庫的資料庫執行個體類型

任務	描述	所需技能
根據磁碟 I/O、CPU 和記憶體預估值判斷資料庫執行個體類型。	根據先前步驟中的預估值，目標 Amazon RDS 或 Aurora 資料庫的容量應為： • 68 核心處理器 • 每秒 143 兆比特的輸送量	DBA

任務	描述	所需技能
	• 4367 IOPS，適用於磁碟 I/O • 464 GB 的記憶體 在目標 Amazon RDS 或 Aurora 資料庫中，您可以將這些值對應至資料庫 .r5.16xlarge 執行個體類型，其容量為 32 個核心、512 GB 的記憶體和 13,600 Mbps 的輸送量。 如需詳細資訊，請參閱 AWS 部落格文章根據 Oracle 效能指標大小適當大小的 Amazon RDS 執行個體大小。	

相關資源

- Aurora 資料庫執行個體類別 (Amazon Aurora 文件
- Amazon RDS 資料庫執行個體儲存 (Amazon RDS 文件)
- AWS 礦工工具 (GitHub 儲存庫)

使用 AWS DMS 將 Amazon RDS for SQL Server 資料表匯出到 S3 儲存貯體

創建者：蘇哈尼·謝克 (AWS)

環境：PoC 或試點	資料來源:	目標：S3
R 類型：不適用	工作負載：Microsoft	技術：資料庫；雲端原生
AWS 服務：AWS DMS; Amazon RDS; Amazon S3; AWS Secrets Manager; AWS Identity and Access Management		

Summary

適用於 SQL Server 的亞馬 Amazon Relational Database Service（Amazon RDS）不支持將數據加載到 Amazon Web Services（AWS）雲上的其他數據庫引擎鏈接的服務器上。相反地，您可以使用 AWS Database Migration Service (AWS DMS) 將適用 Amazon RDS for SQL Server 資料表匯出到亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體，其中資料可供其他資料庫引擎使用。

AWS DMS 可協助您快速安全地將資料庫遷移到 AWS。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。AWS DMS 可以在使用最廣泛的商業和開放原始碼資料庫之間移轉您的資料。

此模式會在設定 AWS DMS 端點時使用 AWS 秘密管理員。Secrets Manager 可協助您保護存取應用程式、服務和 IT 資源所需的機密。您可以使用該服務在整個生命週期中輪換、管理和擷取資料庫登入資料、API 金鑰和其他機密。使用者和應用程式會透過呼叫 Secrets Manager 來擷取密碼，減少對敏感資訊進行硬式編碼的需求。Secrets Manager 提供秘密輪換與 Amazon RDS 內置集成, Amazon Redshift, 和 Amazon DocumentDB. 此外，該服務還可擴展到其他類型的密鑰，包括 API 密鑰和 OAuth 令牌。使用 Secrets Manager，您可以使用精細的許可來控制密碼的存取，並針對 AWS 雲端、第三方服務和內部部署中的資源集中稽核機密輪替。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- S3 儲存貯體
- 虛擬私有雲 (VPC)
- 資料庫子網路
- Amazon RDS for SQL Server
- 具有代表 Amazon RDS 執行個體存取 S3 儲存貯體存取 (列出、取得和放置物件) 的 AWS Identity and Access Management (IAM) 角色。
- 用來儲存 RDS 執行個體認證的 Secrets Manager。

架構

技術, 堆

- Amazon RDS for SQL Server
- AWS DMS
- Amazon S3
- AWS Secrets Manager

目標架構

下圖顯示透過 AWS DMS 協助，將資料從 Amazon RDS 執行個體匯入 S3 儲存貯體的架構。

1. 透過來源端點連接至來源 Amazon RDS 執行個體的 AWS DMS 遷移任務
2. 從來源 Amazon RDS 執行個體複製資料
3. 透過目標端點連接至目標 S3 儲存貯體的 AWS DMS 遷移任務
4. 以逗號分隔值 (CSV) 格式將複製的資料匯出到 S3 儲存貯體

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。

其他服務

- Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器，包括訪問，配置和管理 SQL 服務器組件的工具。

史诗

設定 Amazon RDS for SQL Server 執行個體

任務	描述	所需技能
建立適用於 SQL 伺服器的亞馬遜 RDS 執行個體。	1. 開啟 AWS 管理主控台，選擇 RDS，然後使用標準建立選項建立具有所需版本的 Amazon RDS 執行個體，例如 SQL 伺服器快速版、SQL 伺服器標準版或 SQL 伺服器企業版。對於版本，請選擇 2016 或更新版本。 2. 在「範本」下，選擇「開發/測試」。	DBA，工程師 DevOps
設定執行個體的認證。	1. 輸入執行個體的名稱。 2. 提供 Amazon RDS 執行個體的使用者名稱和密碼。	DBA，工程師 DevOps

任務	描述	所需技能
設定執行個體類別、儲存、auto 擴展和可用性。	1. 從清單中選取資料庫執行個體類別：標準、記憶體最佳化和高載類別。選擇資料庫執行個體類型，以配置針對此資料庫執行個體規劃的工作負載所需的運算、網路和記憶體容量。如需詳細資訊，請參閱 AWS 文件。 2. 從清單中選取儲存類型：一般用途 SSD、佈建 IOPS SSD 或磁帶。視需要配置預設儲存區大小。 3. 選擇啟用儲存自動調度資源，根據您的容量規劃增加 Amazon RDS 儲存。 4. AWS DMS 支援含複寫執行個體的異地同步備份部署。如果可用區域、內部硬體或網路發生中斷服務，AWS DMS 會建立待命執行個體，並透過自動容錯移轉至待命複本來提供高可用性 (HA)。根據匯入的大小，選取適當的選項。	DBA，工程師 DevOps

任務	描述	所需技能
指定 VPC、子網路群組、公用存取和安全性群組。	視需要選取 VPC 人雲端、資料庫子網路群組和 VPC 人雲端安全群組，以建立 Amazon RDS 執行個體。遵循最佳做法，例如： • 請勿啟用 RDS 資料庫執行個體的公開存取權。 • 請勿在安全性群組中使用 CIDR 0.0.0/0。 • 僅使用所需的 IP 位址和連接埠詳細資料來存取 RDS 執行個體。	DBA，工程師 DevOps
設定監控、備份和維護。	1. 指定所需的備份選項。依預設，會啟用自動備份，保留期為 7 天。 2. 選擇適當的 auto 次要版本升級和維護時段設定，以透過 Amazon RDS 將擱置的修改或維護套用至資料庫。 3. 選擇建立資料庫。	DBA，工程師 DevOps

設置數據庫和示例數據

任務	描述	所需技能
建立資料表並載入範例資料。	在新的數據庫中，創建一個表。使用 [其他資訊] 區段中的範例程式碼，將資料載入資料表。	DBA，工程師 DevOps

設定認證

任務	描述	所需技能
建立機密。	1. 在主控台上，選擇 Secrets Manager，然後選擇 [儲存新密碼]。 2. 輸入適用於 SQL 伺服器的亞馬遜 RDS 資料庫的使用者名稱和密碼。 此密碼將用於 AWS DMS 來源端點。	DBA，工程師 DevOps

設定資料庫和 S3 儲存貯體之間的存取

任務	描述	所需技能
建立 IAM 角色以存取 Amazon RDS。	1. 在主控台上，選擇 IAM，然後建立 IAM 角色，為 S3 儲存貯體提供 Amazon RDS 讀取/寫入存取權限。 2. 在功能下，選取 S3 整合。	DBA，工程師 DevOps

建立 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	若要從 Amazon RDS for SQL Server 儲存資料，請在主控台上選擇 S3，然後選擇 [建立儲存貯體]。確定 S3 儲存貯體未公開提供。	DBA，工程師 DevOps

設定 AWS DMS 和 S3 儲存貯體之間的存取

任務	描述	所需技能
為 AWS DMS 建立 IAM 角色以存取 Amazon S3。	建立 IAM 角色，讓 AWS DMS 列出、取得和放置 S3 儲存貯體中的物件。	DBA，工程師 DevOps

設定 AWS DMS

任務	描述	所需技能
建立 AWS DMS 來源端點。	1. 在主控台上，選擇 Database Migration Service，然後選擇端點。建立來源端點，然後選取選取 RDS 資料庫執行個體核取方塊。 2. 對於來源引擎，選取 Microsoft SQL 伺服器。 3. 在 [存取端點資料庫] 下，選擇 [AWS Secrets Manager]，然後輸入您先前建立的密碼和 IAM 角色，以及資料庫名稱。 4. 測試來源端點。	DBA，工程師 DevOps
建立 AWS DMS 目標端點。	建立目標端點，選取 Amazon S3 做為目標引擎。 為您先前建立的 IAM 角色提供 S3 儲存貯體名稱和資料夾名稱。	DBA，工程師 DevOps
建立 AWS DMS 複寫執行個體。	在相同的 VPC、子網路和安全群組中，建立 AWS DMS 複寫	DBA，工程師 DevOps

任務	描述	所需技能
	執行個體。如需選擇執行個體類別的詳細資訊，請參閱 AWS 文件。	
建立 AWS DMS 遷移任務。	若要將資料從 Amazon RDS 版 SQL 伺服器匯出到 S3 儲存貯體，請建立資料庫遷移任務。對於移轉類型，請選擇移轉現有資料。選取您建立的 AWS DMS 端點和複寫執行個體。	DBA，工程師 DevOps

將資料匯出到 S3 儲存貯體

任務	描述	所需技能
執行資料庫移轉工作。	若要匯出 SQL Server 資料表資料，請啟動資料庫移轉工作。此任務會以 CSV 格式將資料從 Amazon RDS for SQL Server 匯出到 S3 儲存貯體。	DBA，工程師 DevOps

清除資源

任務	描述	所需技能
刪除資源。	為避免產生額外費用，請依照下列順序使用主控台刪除資源： 1. 移轉任務 2. Replication instance (複寫執行個體) 3. 端點 4. S3 儲存貯體	DBA，工程師 DevOps

任務	描述	所需技能
	5. 資料庫執行個體	

相關資源

- AWS DMS
- Amazon Simple Storage Service (Amazon S3)
- Amazon RDS for SQL Server
- Amazon S3 整合

其他資訊

若要建立資料庫和資料表，並載入範例資料，請使用下列程式碼。

```
--Step1: Database creation in RDS SQL Server
CREATE DATABASE [Test_DB]
 ON  PRIMARY
( NAME = N'Test_DB', FILENAME = N'D:\rdsdbdata\DATA\Test_DB.mdf' , SIZE = 5120KB ,
 FILEGROWTH = 10%)
 LOG ON
( NAME = N'Test_DB_log', FILENAME = N'D:\rdsdbdata\DATA\Test_DB_log.ldf' , SIZE =
 1024KB , FILEGROWTH = 10%)
GO

--Step2: Create Table
USE Test_DB
GO
Create Table Test_Table(ID int, Company Varchar(30), Location Varchar(20))

--Step3: Load sample data.
USE Test_DB
GO
Insert into Test_Table values(1,'AnyCompany','India')
Insert into Test_Table values(2,'AnyCompany','USA')
Insert into Test_Table values(3,'AnyCompany','UK')
Insert into Test_Table values(4,'AnyCompany','Hyderabad')
Insert into Test_Table values(5,'AnyCompany','Banglore')
```

處理動態 SQL 語句中 Aurora PostgreSQL 塊

創建者：阿努拉達奇塔（AWS）

環境：PoC 或試點	來源：數據庫關係	目標：PostgreSQL
R 型：重新建築	工作負載：甲骨文; 開源	技術：資料庫；移轉
AWS 服務：Amazon Aurora; Amazon RDS		

Summary

此模式說明如何避免在動態 SQL 陳述式中處理匿名區塊時發生的錯誤。當您使用 AWS 結構描述轉換工具將 Oracle 資料庫轉換為 Aurora PostgreSQL 相容版本資料庫時，您會收到錯誤訊息。若要避免錯誤，您必須知道0UT繫結變數的值，但在執行 SQL 陳述式之後才能知道0UT繫結變數的值。AWS 結 Schema Conversion Tool (AWS SCT) 無法理解動態 SQL 陳述式中的邏輯所產生的錯誤。AWS SCT 無法轉換 PL/SQL 程式碼中的動態 SQL 陳述式 (也就是函式、程序和套件)。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- Aurora 資料庫 (資料庫) 執行個體
- 適用於甲骨文資料庫執行個體的 Amazon Relational Database Service 服務
- PostgreSQL 式終端
- SQL * 加
- AWS_ORACLE_EXT目標資料庫中的結構描述 (AWS SCT 延伸套件的一部分)
- 最新版本的 AWS Schema Conversion Tool (AWS SCT) 及其所需驅動程式

架構

源, 技術, 堆棧

- 內部部署 Oracle 資料庫 10g 及更新版本

目標技術堆疊

- Amazon Aurora PostgreSQL
- Amazon RDS for PostgreSQL
- AWS Schema Conversion Tool

移轉架構

下圖顯示如何使用 AWS SCT 和 Oracle 0UT 繫結變數掃描應用程式程式碼中的內嵌 SQL 陳述式，並將程式碼轉換為 Aurora 資料庫可以使用的相容格式。

該圖顯示以下工作流程：

1. 使用 Aurora PostgreSQL 做為目標資料庫，為來源資料庫產生 AWS SCT 報告。
2. 識別動態 SQL 程式碼區塊中的匿名區塊 (AWS SCT 引發錯誤)。
3. 手動轉換程式碼區塊，並在目標資料庫上部署程式碼。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分資料庫程式碼物件轉換為與目標資料庫相容的格式，協助您將異質資料庫遷移變得可預測。

其他工具

- pgAdmin 允許您連接到您的數據庫服務器並與之交互。
- 「Oracle SQL 開發人員」是一個整合式開發環境，您可以使用它來開發和管理 Oracle 資料庫中的資料庫。您可以使用 SQL *Plus 或甲骨文 SQL 開發人員來處理此模式。

史诗

設定 Oracle 來源資料庫

任務	描述	所需技能
在 Amazon RDS 或亞馬 Amazon EC2 上創建一個甲骨文實例。	若要在 Amazon RDS 上建立 Oracle 資料庫執行個體，請參閱 Amazon RDS 說明文件中的建立 Oracle 資料庫執行個體並連接到 Oracle 資料庫執行個體上的資料庫。 若要在亞馬遜彈性運算雲端 (Amazon EC2) 上建立 Oracle 資料庫執行個體，請參閱 AWS Prescriptive Guidance 文件中的適用於甲骨文的亞馬遜 EC2。	DBA
建立要移轉的資料庫結構描述和物件。	您可以使用 Amazon Cloud Directory 建立資料庫結構描述。如需詳細資訊，請參閱 Cloud Directory 說明文件中的建立結構描述。	DBA
設定輸入和輸出安全性群組。	若要建立和設定安全群組，請參閱 Amazon RDS 說明文件中的使用安全群組控制存取。	DBA
確認資料庫正在執行中。	若要檢查資料庫的狀態，請參閱 Amazon RDS 文件中的檢視 Amazon RDS 事件。	DBA

設定目標 Aurora 資料庫

任務	描述	所需技能
在 Amazon RDS PostgreSQL 建立一個 Aurora 執行個體。	若要建立 Aurora PostgreSQL 執行個體，請參閱 Amazon RDS 說明文件中的建立資料庫叢集並連接至 Aurora PostgreSQL 資料庫叢集上的資料庫。	DBA
設定輸入和輸出安全性群組。	若要建立和設定安全群組，請參閱 Aurora 說明文件中的透過建立安全群組來提供 VPC 中資料庫叢集的存取權。	DBA
確認 Aurora 資 PostgreSQL 在執行中。	若要檢查資料庫的狀態，請參閱 Aurora 文件中的檢視 Amazon RDS 事件。	DBA

設定 AWS SCT

任務	描述	所需技能
將 AWS SCT Connect 到來源資料庫。	若要將 AWS SCT 連接到來源資料庫，請參閱 AWS SCT 文件中的以來源形式連接到 PostgreSQL。	DBA
將 AWS SCT Connect 到目標資料庫。	若要將 AWS SCT 連接到目標資料庫，請參閱什麼是 AWS 結 Schema Conversion Tool？在 AWS 結 Schema Conversion Tool 使用者指南中。	DBA

任務	描述	所需技能
在 AWS SCT 中轉換資料庫結構描述，並將自動轉換的程式碼儲存為 SQL 檔案。	若要儲存 AWS SCT 轉換的檔案，請參閱 AWS 結構描述轉換工具使用者指南中的在 AWS SCT 中儲存和套用轉換後的結構描述。	DBA

遷移代碼

任務	描述	所需技能
獲取用於手動轉換的 SQL 文件。	在 AWS SCT 轉換的檔案中，提取需要手動轉換的 SQL 檔案。	DBA
更新指令碼。	手動更新 SQL 檔案。	DBA

相關資源

- Amazon RDS
- Amazon Aurora 功能

其他資訊

下列範例程式碼顯示如何設定 Oracle 來源資料庫：

```
CREATE or replace PROCEDURE calc_stats_new1 (
  a NUMBER,
  b NUMBER,
  result out NUMBER)
IS
BEGIN
result:=a+b;
END;
/
```

```
set serveroutput on ;

DECLARE
  a NUMBER := 4;
  b NUMBER := 7;
  plsql_block VARCHAR2(100);
  output number;
BEGIN
  plsql_block := 'BEGIN calc_stats_new1(:a, :b,:output); END;';
  EXECUTE IMMEDIATE plsql_block USING a, b,out output;
  DBMS_OUTPUT.PUT_LINE('output:'||output);

END;
```

下列範例程式碼示範如何設定目標 Aurora PostgreSQL 資料庫：

```
 w integer,
 x integer)
RETURNS integer
AS
$BODY$
DECLARE
begin
return w + x ;
end;
$BODY$
LANGUAGE  plpgsql;


CREATE OR REPLACE FUNCTION test_pg.init()
RETURNS void
AS
$BODY$
BEGIN
if aws_oracle_ext.is_package_initialized
      ('test_pg' ) then
      return;
    end if;
    perform aws_oracle_ext.set_package_initialized
      ('test_pg' );

PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', NULL::INTEGER);
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', NULL::text);
```

```
END;
$BODY$
LANGUAGE  plpgsql;


DO $$
declare
v_sql text;
v_output_loc int;
a integer :=1;
b integer :=2;
BEGIN
perform  test_pg.init();
--raise notice 'v_sql %',v_sql;
execute 'do $a$ declare v_output_l int; begin select * from test_pg.calc_stats_new1('||
a||','||b||') into v_output_l;
PERFORM aws_oracle_ext.set_package_variable(''test_pg'', ''v_output'', v_output_l) ;
 end; $a$'  ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
raise notice 'v_output_loc %',v_output_loc;
END ;
$$
```

在 Aurora 兼容後處理過載的甲骨文功能

創建者蘇曼娜亞南德拉 (AWS)

環境：PoC 或試點	來源：甲骨文數據庫	目標：Aurora 郵政兼容
R 類型：重新平台	工作量：甲骨文	技術：資料庫；移轉
AWS 服務：Amazon Aurora		

Summary

您從現場部署 Oracle 資料庫遷移到 Amazon Aurora PostgreSQL 相容版本的程式碼可能包含過載功能。這些函數具有相同的定義 — 也就是說，相同的函數名稱和 input (IN) 參數的數目和資料類型，但資料類型或 output () 參數的數目可能會有所不同。0UT

這些參數不匹配可能會導致 PostgreSQL 中的問題，因為很難確定要運行哪個函數。此模式說明當您將資料庫程式碼遷移至 Aurora PostgreSQL 相容時，如何處理多載函式。

先決條件和限制

先決條件

- 作為來源資料庫的 Oracle 資料庫執行處理
- 將 Aurora PostgreSQL 相容的資料庫執行個體做為您的目標資料庫 (請參閱 Aurora 文件中的指示)

產品版本

- Oracle 資料庫 9i 或更新版本
- 甲骨文 SQL 開發者版本 18.4.0.376
- pgAdmin 4 客戶端
- 與 Aurora PostgreSQL 相容的版本 11 或更新版本 (請參閱 Aurora 文件中的識別 Amazon Aurora PostgreSQL 版本)

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。

其他工具

- Oracle SQL 開發人員是一個免費的整合式開發環境，可在傳統和雲端部署中使用 Oracle 資料庫中的 SQL。
- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

創建一個簡單的函數

任務	描述	所需技能
在 PostgreSQL 中創建具有一個輸入參數和一個輸出參數的函數。	下面的例子說明了`test_overloading` 在 Aurora PostgreSQL 兼容命名的函數。該函數有兩個參數：一個輸入文本參數和一個輸出文本參數。 `CREATE OR REPLACE FUNCTION public.test_overloading( str1 text, OUT str2 text) LANGUAGE 'plpgsql' COST 100 VOLATILE AS $BODY$ DECLARE BEGIN`	資料工程師，兼容 Aurora

任務	描述	所需技能
	`str2 := 'Success'; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; $BODY$;`	
在執行 PostgreSQL。	執行您在上一個步驟中建立的函數。 `select public.test_overloading('Test');` 它應該顯示以下輸出。 `Success`	資料工程師，兼容 Aurora

重載功能

任務	描述	所需技能
使用相同的函數名稱在 PostgreSQL 中創建一個重載的函數。	在 Aurora PostgreSQL 相容中建立一個重載函數，該函數使用與之前的函數相同的函數名稱。下面的例子也被命名`test_overloading` ，但它有三個參數：一個輸入文本參數，一個輸出文本參數和一個輸出整數參數。 `CREATE OR REPLACE FUNCTION public.test_overloading(`	資料工程師，兼容 Aurora

任務	描述	所需技能
	str1 text, OUT str2 text, OUT num1 integer) LANGUAGE 'plpgsql' COST 100 VOLATILE AS $BODY$ DECLARE str3 text; BEGIN str2 := 'Success'; num1 := 100; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; $BODY$;	

任務	描述	所需技能
在執行 PostgreSQL。	當你運行這個函數，它失敗，並顯示以下錯誤消息。 `ERROR: cannot change return type of existing function` `HINT: Use DROP FUNCTION test_overloading(text) first.` 發生這種情況是因為 Aurora PostgreSQL 兼容不直接支持函數重載。它無法識別要運行哪個函數，因為輸出參數的數量在第二個版本的函數中是不同的，儘管輸入參數是相同的。	資料工程師，兼容 Aurora

套用因應措施

任務	描述	所需技能
將 INOUT 加入至第一個輸出參數。	因應措施是將第一個輸出參數表示為來修改函數程式碼INOUT。 `CREATE OR REPLACE FUNCTION public.test_overloading( str1 text, INOUT str2 text, OUT num1 integer) LANGUAGE 'plpgsql' COST 100`	資料工程師，兼容 Aurora

任務	描述	所需技能
	<pre> VOLATILE AS $BODY$ DECLARE str3 text; BEGIN str2 := 'Success'; num1 := 100; RETURN ; EXCEPTION WHEN others THEN RETURN ; END; $BODY$;</pre>	
執行修改後的函數。	使用下列查詢執行您更新的函數。您傳遞 null 值作為此函數的第二個引數，因為您宣告此參數INOUT為避免錯誤。 <pre>select public.te st_overloading('Te st', null);</pre> 該函數現在已成功創建。 <pre>Success, 100</pre>	資料工程師，兼容 Aurora
驗證結果。	驗證具有多載函數的代碼是否已成功轉換。	資料工程師，兼容 Aurora

相關資源

- 使用 Amazon Aurora PostgreSQL (Aurora 文件)
- 甲骨文中的函數重載 (Oracle 文檔管理系統)

- 函數 PostgreSQL (PostgreSQL)

協助強制執行 DynamoDB 標記

創建者：曼西蘇拉特瓦拉（AWS）

環境：生產	技術：資料庫；雲端原生；安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：Amazon CloudWatch；Amazon DynamoDB；AWS Lambda；Amazon SNS		

Summary

當亞馬遜網路服務 (AWS) 雲端上的 DynamoDB 資源遺失或移除預先定義的 Amazon DynamoDB 標籤時，此模式會設定自動通知。

DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及可擴充性。DynamoDB 可讓您減輕操作和擴展分散式資料庫的管理負擔。使用 DynamoDB 時，您不必擔心硬體佈建、設定和組態、複寫、軟體修補或叢集擴展問題。

該模式使用 AWS CloudFormation 範本，該範本會建立 Amazon CloudWatch 活動事件和 AWS Lambda 函數。此事件會使用 AWS 監視任何新的或現有的 DynamoDB 標記資訊。 CloudTrail如果遺失或移除預先定義的標籤，會 CloudWatch 觸發 Lambda 函數，該函數會傳送 Amazon Simple Notification Service (Amazon SNS) 通知，通知您違規事件。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 適用於 Lambda .zip 檔案的 Amazon Simple Storage Service (Amazon S3) 儲存貯體，其中包含用於執行 Lambda 函數的 Python 指令碼

限制

- 只有在TagResource或UntagResource CloudTrail 事件發生時，解決方案才有效。它不會為任何其他事件建立通知。

架構

目標技術堆疊

- Amazon DynamoDB
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon S3
- Amazon SNS

目標架構

自動化和規模

您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需在每個區域或帳戶中執行一次範本。

工具

工具

- Amazon DynamoDB — DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及可擴展性。
- AWS CloudTrail — CloudTrail 是一項 AWS 服務，可協助您對 AWS 帳戶進行管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS Lambda — Lambda 是一種運算服務，可支援執行程式碼，而不需要佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。

- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種網路服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。

Code

- 專案的 .zip 檔案可作為附件使用。

史诗

定義 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。此 S3 儲存貯體將託管 Lambda 程式碼 .zip 檔案。您的 S3 儲存貯體必須與受監控的 DynamoDB 資源位於相同的 AWS 區域。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將附件區段中提供的 Lambda 程式碼 .zip 檔案上傳至 S3 儲存貯體。S3 儲存貯體必須位於與受監控的 DynamoDB 資源相同的區域。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
部署 AWS CloudFormation 範本。	在 AWS 主 CloudFormation 控台上，部署「附件」區段中提供的 AWS CloudFormation 範本。在下一個史詩中，提供參數的值。	雲端架構師

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立或選擇的 S3 儲存貯體的名稱。	雲端架構師
提供 Amazon S3 密鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<folder>/<file-name>.zip`	雲端架構師
提供電子郵件地址	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師
輸入所需的標籤金鑰。	請確定標籤以逗號分隔，兩者之間沒有空格 (例如`ApplicationId,CreatedBy,Environment,`	雲端架構師

任務	描述	所需技能
	`Organization`)。E CloudWatch vents 事件會搜尋這些標籤，如果找不到標籤，則傳送通知。	

確認訂閱。

任務	描述	所需技能
確認訂閱。	範本成功部署後，會將訂閱電子郵件傳送至您提供的電子郵件地址。若要接收違規通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 建立 S3 儲存貯體
- 將檔案上傳到 S3 儲存貯體
- 在 DynamoDB 資源中標記
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS DMS 和 Amazon Aurora 實作跨區域災難復原

創建者馬克·哈德森 (AWS)

環境：生產	技術：資料庫	AWS 服務：AWS DMS；Amazon RDS；Amazon Aurora

Summary

自然或人為造成的災害可能隨時發生，並可能影響指定 Amazon Web Services (AWS) 區域中執行的服務和工作負載的可用性。為了減輕風險，您必須開發災難復原 (DR) 計劃，其中包含 AWS 服務的內建跨區域功能。對於本質上不提供跨區域功能的 AWS 服務，DR 計劃還必須提供解決方案來處理跨 AWS 區域的容錯移轉。

此模式會引導您完成災難復原設定，其中涉及單一區域中的兩個 Amazon Aurora MySQL 相容版本資料庫叢集。為了符合 DR 要求，資料庫叢集設定為使用 Amazon Aurora 全球資料庫功能，並具有跨多個 AWS 區域的單一資料庫。AWS Database Migration Service (AWS DMS) 任務會在本地區域的叢集之間複寫資料。不過，AWS DMS 目前不支援區域之間的任務容錯移轉。此模式包括解決該限制以及在兩個區域獨立設定 AWS DMS 所需的步驟。

先決條件和限制

先決條件

- 支援 Amazon Aurora 全球資料庫的選定主要和次要 AWS 區域。
- 在主要區域的單一帳戶中，有兩個獨立的 Amazon Aurora 與 MySQL 相容版本資料庫叢集。
- 資料庫執行處理類別 db.r5 或更高版本 (建議使用)。
- 主要區域中的 AWS DMS 任務，在現有資料庫叢集之間執行持續複寫。
- DR 區域資源可符合建立資料庫執行個體的需求。如需詳細資訊，請參閱在 VPC 中使用資料庫執行個體。

限制

- 如需 Amazon Aurora 全球資料庫限制的完整清單，請參閱 Amazon Aurora 全球資料庫的限制。

產品版本

- Amazon Aurora 與 MySQL 相容的版本 5.7 或 8.0。如需詳細資訊，請參閱 Amazon Aurora 版本。

架構

目標技術堆疊

- Amazon Aurora MySQL 兼容版全球數據庫集群
- AWS DMS

目標架構

下圖顯示兩個 AWS 區域的全域資料庫，其中一個包含主要主資料庫和記者資料庫以及 AWS DMS 複寫，另一個具有次要主資料庫和記者資料庫。

自動化和規模

您可以使用 AWS CloudFormation 在次要區域建立先決條件基礎設施，例如虛擬私有雲端 (VPC)、子網路和參數群組。您也可以使用 AWS CloudFormation 在 DR 區域中建立次要叢集，並將其新增至全域資料庫。如果您使用樣板在主要區域中建立資料庫叢集，則可以使用其他 CloudFormation 範本更新或擴充它們，以建立全域資料庫資源。如需詳細資訊，請參閱使用兩個資料庫執行個體建立 Amazon Aurora 資料庫叢集和為 Aurora MySQL 建立全域資料庫叢集。

最後，您可以在發生容錯移轉和容錯回復事件 CloudFormation 之後，在主要和次要區域中建立 AWS DMS 任務。如需詳細資訊，請參閱AWS::DMS::ReplicationTask。

工具

- Amazon Aurora-Amazon Aurora 是一個全受管的關聯式資料庫引擎，與 MySQL 和 PostgreSQL 相容。此模式使用 Amazon Aurora MySQL 兼容版。
- Amazon Aurora 全球資料庫-Amazon Aurora 全球資料庫專為全球分散式應用程式所設計 一個 Amazon Aurora 全球資料庫可以跨越多個 AWS 區域。它會複寫您的資料，而不會影響資料庫效能。它還可以在每個區域實現低延遲的快速本地讀取，並提供從全區域中斷的災難恢復。
- AWS DMS-AWS Database Migration Service (AWS DMS) 提供一次性移轉或持續複寫。進行中的複寫工作可讓來源和目標資料庫保持同步。設定完成後，進行中的複寫工作會以最小的延遲持續將來源變更套用至目標。所有 AWS DMS 功能 (例如資料驗證和轉換) 都可用於任何複寫任務。

史诗

準備主要區域中的現有資料庫叢集

任務	描述	所需技能
修改資料庫叢集參數群組。	在現有的資料庫叢集參數群組中，將**binlog_format** 參數設定為 row 值，以啟動資料列層級二進位記錄。 在執行持續複寫或變更資料擷取 (CDC) 時，AWS DMS 需要 MySQL 相容資料庫的資料列層級二進位記錄。如需詳細資訊，請參閱使用 AWS 受管的 MySQL 相容資料庫做為 AWS DMS 的來源。	AWS 管理員
更新資料庫二進位記錄保留期間。	使用安裝在最終使用者裝置或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 MySQL 用戶端，在主資料庫叢集的寫入器節點上執行 Amazon 關聯式資料庫服務 (Amazon RDS) 提供的下列存放程序，其中XX是保留記錄的小時數。 `call mysql.rds_set_configuration('binlog retention hours', XX)` 執行下列命令以確認設定。 `call mysql.rds_show_configuration;`	DBA

任務	描述	所需技能
	AWS 管理的與 MySQL 相容的資料庫會盡快清除二進位記錄。因此，保留期必須足夠長，以確保 AWS DMS 任務執行之前不會清除日誌。24 小時的值通常就足夠了，但值應根據 DR 區域中設定 AWS DMS 任務所需的時間而定。	

更新主要區域中的現有 AWS DMS 任務

任務	描述	所需技能
記錄 AWS DMS 任務 ARN。	使用 Amazon 資源名稱 (ARN) 獲取 AWS DMS 任務名稱以供日後使用。若要擷取 AWS DMS 任務 ARN，請在主控台中檢視任務或執行下列命令。 `aws dms describe-replication-tasks` ARN 看起來像下面這樣。 `arn:aws:dms:us-east-1:<accountid>:task:AN6HFFMPM246XOZVEUHCNSOVF7MQCLTOZUIRAMY` 最後一個冒號之後的字元對應於稍後步驟中使用的工作名稱。	AWS 管理員

任務	描述	所需技能
修改現有的 AWS DMS 任務以記錄檢查點。	AWS DMS 會建立包含資訊的檢查點，以便複寫引擎知道變更串流的復原點。若要記錄檢查點資訊，請在主控台中執行下列步驟： 1. 停止執行 AWS DMS 任務。 2. 使用工作中的 JSON 編輯器將 TaskRecoveryTableEnabled 參數設定為 true。 3. 啟動 AWS DMS 任務。	AWS 管理員
驗證檢查點資訊。	使用連線到叢集之寫入器端點的 MySQL 用戶端，查詢記者資料庫叢集中的新中繼資料表，以確認它是否存在並包含複寫狀態資訊。執行下列命令。 `select * from awsdms_control.awsdms_txn_state;` 來自 ARN 的任務名稱應該在`Task_Name` 列的這個表中找到。	DBA

將兩個 Amazon Aurora 叢集擴展到 DR 區域

任務	描述	所需技能
在 DR 區域中建立基礎結構。	建立和存取 Amazon Aurora 叢集所需的基礎元件： • 虛擬私有雲端 (VPC)	AWS 管理員

任務	描述	所需技能
	• 子網 • 安全群組 • 網路存取控制清單 • 子網路群組 • DB parameter group (資料庫參數群組) • DB cluster parameter group (資料庫叢集參數群組) 請確定兩個參數群組的組態與主要區域中的組態相符。	
將 DR 區域新增至兩個 Amazon Aurora 叢集。	將次要區域 (DR 區域) 新增至主要和記者 Amazon Aurora 叢集。如需詳細資訊，請參閱將 AWS 區域新增至 Amazon Aurora 全球資料庫。	AWS 管理員

執行錯移轉

任務	描述	所需技能
停止執行 AWS DMS 任務。	發生容錯移轉後，主要區域中的 AWS DMS 任務將無法正常運作，應停止以避免發生錯誤。	AWS 管理員
執行受管理的容錯移轉。	對 DR 區域執行主要資料庫叢集的受管理容錯移轉。如需指示，請參閱針對 Amazon Aurora 全球資料庫執行受管的計劃容錯移轉。完成主資料庫叢集的容錯移轉之後，請在報	AWS 管理員，DBA

任務	描述	所需技能
	告者資料庫叢集上執行相同的活動。	
將數據加載到主數據庫中。	將測試資料插入 DR 資料庫叢集中主要資料庫的寫入器節點。此資料將用於驗證複製是否正常運作。	DBA
建立 AWS DMS 複寫執行個體。	若要在 DR 區域中建立 AWS DMS 複寫執行個體，請參閱建立複寫執行個體。	AWS 管理員，DBA
建立 AWS DMS 來源和目標端點。	若要在 DR 區域中建立 AWS DMS 來源和目標端點，請參閱建立來源和目標端點。來源應指向主資料庫叢集的寫入器執行個體。目標應該指向報告者資料庫叢集的寫入器執行個體。	AWS 管理員，DBA
取得複寫檢查點。	若要取得複寫檢查點，請使用 MySQL 用戶端，針對 DR 區域中報告者資料庫叢集中的寫入器節點執行下列命令，以查詢中繼資料表。 `select * from awsdms_control.awsdms_txn_state;` 在表格中，找出與您在第二個史詩中取得的主要區域中存在的 AWS DMS 任務的 ARN 相對應的 task_name 值。	DBA

任務	描述	所需技能
建立 AWS DMS 任務。	使用主控台在 DR 區域中建立 AWS DMS 任務。在工作中，指定「僅複製資料變更」的移轉方法。如需詳細資訊，請參閱建立工作。 1. 在工作設定中，使用精靈指定下列項目： • 來源交易的 CDC 啟動模式 — 啟用自訂 CDC 啟動模式 • 來源交易的自訂 CDC 起點 — 指定復原檢查點 2. 在 [復原檢查點] 方塊中，輸入先前透過資料`awsdms_txn_state` 表上的資料庫查詢取得的複寫檢查點值。 3. 在工作設定區段中，選取 JSON 編輯器，然後將`TaskRecoveryTableEnabled`參數設定為 true。 將 AWS DMS 任務 [開始遷移任務] 設定設定為 [建立時自動]。	AWS 管理員，DBA
記錄 AWS DMS 任務 ARN。	使用 ARN 取得 AWS DMS 任務名稱，以供日後使用。若要擷取 AWS DMS 任務 ARN，請執行下列命令。 `aws dms describe-replication-tasks`	AWS 管理員，DBA

任務	描述	所需技能
驗證複製的資料。	查詢 DR 區域中的報告程式資料庫叢集，以確認您載入主資料庫叢集的測試資料是否已複寫。	DBA

執行容錯回復

任務	描述	所需技能
停止執行 AWS DMS 任務。	發生故障回復之後，DR 區域中的 AWS DMS 任務將無法正常運作，應停止以避免發生錯誤。	AWS 管理員
執行受管理的容錯回復。	將主資料庫叢集容錯回到主要區域。如需指示，請參閱針對 Amazon Aurora 全球資料庫執行受管的計劃容錯移轉。完成主資料庫叢集的容錯回復之後，請在報告者資料庫叢集上執行相同的活動。	AWS 管理員，DBA
取得複寫檢查點。	若要取得複寫檢查點，請使用 MySQL 用戶端，針對 DR 區域中報告者資料庫叢集中的寫入器節點執行下列命令，以查詢中繼資料表。 `select * from awsdms_control.awsdms_txn_state;` 在表格中，找出與您在第四個史詩中取得的 DR 區域中存在	DBA

任務	描述	所需技能
	的 AWS DMS 任務 ARN 相對應的`task_name` 值。	
更新 AWS DMS 來源和目標端點。	資料庫叢集失敗後，請檢查主要區域中的叢集，以判斷哪些節點是寫入器執行個體。然後驗證主要區域中現有的 AWS DMS 來源和目標端點是否指向寫入器執行個體。如果沒有，請使用寫入器執行個體網域名稱系統 (DNS) 名稱更新端點。	AWS 管理員

任務	描述	所需技能
建立 AWS DMS 任務。	使用主控台在主要區域中建立 AWS DMS 任務。在工作中，指定「僅複製資料變更」的移轉方法。如需詳細資訊，請參閱建立工作。 1. 在工作設定中，使用精靈並指定下列項目： • 來源交易的 CDC 啟動模式 — 啟用自訂 CDC 啟動模式 • 來源交易的自訂 CDC 起點 — 指定復原檢查點 2. 在 [復原檢查點] 方塊中，輸入先前透過資料 `awsdms_txn_state` 表上的資料庫查詢取得的複寫檢查點值。 3. 同樣在任務設置部分中，選擇 JSON 編輯器並將TaskRecoveryTableEnabled參數設置為 true。 4. 最後，將 AWS DMS 任務開始遷移任務設定設定設定為「建立時自動」。	AWS 管理員，DBA

任務	描述	所需技能
記錄 AWS DMS 任務 Amazon 資源名稱 (ARN)。	使用 ARN 取得 AWS DMS 任務名稱，以供日後使用。若要擷取 AWS DMS 任務 ARN，請執行下列命令： `aws dms describe-replication-tasks` 執行另一個受管理容錯移轉時或在 DR 案例期間，將需要工作名稱。	AWS 管理員，DBA
刪除 AWS DMS 任務。	刪除主要區域中的原始 (目前停止) AWS DMS 任務和次要區域中現有的 AWS DMS 任務 (目前已停止)。	AWS 管理員

相關資源

- 設定 Amazon Aurora 資料庫叢集
- 使用 Amazon Aurora 全球數據
- 使用 Amazon Aurora MySQL
- 使用 AWS DMS 複寫執行個體
- 使用 AWS DMS 端點
- 使用 AWS DMS 任務
- 什麼是 AWS CloudFormation？

其他資訊

此範例中使用 Amazon Aurora 全域資料庫適用於 DR，因為它們提供 1 秒的有效復原時間目標 (RTO) 以及少於 1 分鐘的復原點目標 (RPO)，兩者都低於傳統複寫解決方案，而且非常適合 DR 案例。

Amazon Aurora 全球資料庫還提供許多其他優點，包括：

- 具有本機延遲的全域讀取 — 全球消費者可以存取本機區域中的資訊，並具有本機延遲。
- 可擴展的次要 Amazon Aurora 資料庫叢集 — 次要叢集可獨立擴展，最多可新增 16 個唯讀複本。
- 從主要 Amazon Aurora 資料庫叢集快速複寫 — 複寫對主要叢集的效能影響很小。它發生在儲存層，典型的跨區域複寫延遲少於 1 秒。

此模式也會使用 AWS DMS 進行複寫。Amazon Aurora 資料庫提供建立僅供讀取複本的功能，以簡化複寫程序和 DR 設定。不過，當需要資料轉換或目標資料庫需要來源資料庫沒有的其他索引時，通常會使用 AWS DMS 進行複寫。

將具有 100 個以上引數的甲骨文函數和程序遷移到 PostgreSQL

創建者斯里瓦斯·波特拉赫沃 (AWS)

環境：PoC 或試點	來源：甲骨文	目標：PostgreSQL
R 類型：重新平台	工作負載：開放原始碼；	技術：資料庫；移轉
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

此模式顯示如何將具有 100 個以上引數的 Oracle 資料庫函數和程序移轉至 PostgreSQL。例如，您可以使用此模式將 Oracle 函數和程序遷移到以下與 PostgreSQL 相容的 AWS 資料庫服務之一：

- Amazon Relational Database Service 服務 (Amazon RDS)
- Amazon Aurora PostgreSQL-Compatible Edition

PostgreSQL 不支援具有超過 100 個引數的函數或程序。因應措施是，您可以定義具有符合來源函數引數之類型欄位的新資料類型。然後，您可以建立並執行使用自訂資料類型做為引數的 PL/pgSQL 函數。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一個 Amazon RDS 甲骨文數據庫（數據庫）實例
- Amazon RDS for PostgreSQL 的資料庫執行個體或相容於 Aurora 的資料庫執行個體

產品版本

- Amazon RDS Oracle 資料庫執行個體版本 10.2 及更新版
- Amazon RDS PostgreSQL 資料庫執行個體 9.4 版及更新版本，或與 Aurora 相容資料庫執行個體 9.4 及更新版本

- 甲骨文 SQL 開發者版本 18 及更新版本
- pgAdmin 版本 4 及更高版本

架構

源, 技術, 堆棧

- Amazon RDS Oracle 資料庫執行個體版本 10.2 及更新版

目標技術堆疊

- Amazon RDS PostgreSQL 資料庫執行個體 9.4 版及更新版本，或與 Aurora 相容資料庫執行個體 9.4 及更新版本

工具

AWS 服務

- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。

其他服務

- Oracle SQL 開發人員是一個整合式開發環境，可簡化傳統與雲端式部署中 Oracle 資料庫的開發與管理。
- pgAdmin 是一個開放源代碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

最佳實務

請確定您建立的資料類型與來源 Oracle 函數或程序中包含的類型欄位相符。

史诗

執行含有超過 100 個引數的 Oracle 函數或程序

任務	描述	所需技能
建立或識別具有 100 個以上引數的現有 Oracle/PLSQL 函數或程序。	建立具有超過 100 個引數的 Oracle/PLSQL 函數或程序。 -或- 識別具有 100 個以上引數的現有 Oracle/PLSQL 函數或程序。 如需詳細資訊，請參閱 Oracle 資料庫說明文件中的第 14.7 節建立函數陳述式和 14.11 建立程序陳述式。	Oracle/PLSQL 知識
編譯 Oracle /PLSQL 函數或程序。	編譯 Oracle /PLSQL 函數或程序。 如需詳細資訊，請參閱 Oracle 資料庫文件中的編譯函數。	Oracle/PLSQL 知識
執行 Oracle /PLSQL 函數。	執行 Oracle /PLSQL 函數或程序。然後，保存輸出。	Oracle/PLSQL 知識

定義符合來源函數或程序引數的新資料類型

任務	描述	所需技能
在 PostgreSQL 中定義一個新的數據類型。	在 PostgreSQL 中定義一個新的數據類型，其中包括源 Oracle 函數或過程的參數中出現的所有相同字段。	PL/PGSQL 知識

任務	描述	所需技能
	如需詳細資訊，請參閱 PostgreSQL 文件中的建立類型。	

創建一個包含新類型引數的 PostgreSQL 函數

任務	描述	所需技能
建立包含新資料類型的 PostgreSQL 函式。	建立包含新TYPE引數的 PostgreSQL 函數。 若要檢閱範例函數，請參閱此模式的其他資訊一節。	PL/PGSQL 知識
編 PostgreSQL。	在編譯 PostgreSQL。如果新的資料類型欄位與來源函式或程序的引數相符，則函式會成功編譯。	PL/PGSQL 知識
執行 PostgreSQL 函數。	執行 PostgreSQL 函數。	PL/PGSQL 知識

故障診斷

問題	解決方案
該函數返回以下錯誤： 錯誤:「」附近的語法錯誤 <statement>	請確定所有函式的陳述式都以分號 (;) 結尾。
該函數返回以下錯誤： 錯誤：「」不是已知變數 <variable>	確保函數體中使用的變量列在函數的DECLARE部分中。

相關資源

- 使用 Amazon Aurora PostgreSQL（Amazon Aurora 用戶指南 Aurora）
- 建立類型 (文件集)

其他資訊

包含類型引數的範例 PostgreSQL 函數

```
CREATE OR REPLACE FUNCTION test_proc_new
(
    IN p_rec type_test_proc_args
)
RETURNS void
AS
$BODY$
BEGIN

    /*
    **************
    The body would contain code to process the input values.
    For our testing, we will display couple of values.
    ***************
    */
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_acct_id: ', p_rec.p_acct_id);
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_ord_id: ', p_rec.p_ord_id);
    RAISE NOTICE USING MESSAGE = CONCAT_WS('', 'p_ord_date: ', p_rec.p_ord_date);

END;
$BODY$
LANGUAGE plpgsql
COST 100;
```

將適用於 Oracle 資料庫執行個體的 Amazon RDS 移轉到使用 AMS 的其他帳戶

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for Oracle 的 AWS Managed Services
R 類型：重新主機	工作量：甲骨文	技術：資料庫、移轉、儲存與備份
AWS 服務：Amazon RDS；AWS Managed Services		

Summary

此模式說明如何將 Oracle 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS) 從一個 AWS 帳戶遷移到另一個 AWS 帳戶。此模式適用於來源 AWS 帳戶不使用 AWS Managed Services (AMS)，但目標帳戶確實使用 AMS 的案例。您可以在 AMS 中使用變更請求 (RFC) 來完成移轉，而不是使用 AWS 管理主控台執行資料庫操作。對於具有大量交易的多 TB Oracle 來源資料庫，此方法可將停機時間降至最低。例如，400—900 GB 資料庫的停機時間可能會持續大約兩到三個小時。資料庫遷移時間與適用於 Oracle 資料庫執行個體的 Amazon RDS 大小成正比。

重要事項：此模式要求您在來源帳戶中建立 Amazon RDS for Oracle 資料庫執行個體的資料庫快照，將快照複製到使用 AMS 的目標帳戶，然後透過提高 RFC 從該快照建立新的資料庫執行個體。

先決條件和限制

先決條件

- 來源帳戶的有效 AWS 帳戶
- 使用 AMS 做為目標帳戶的作用中 AWS 帳戶
- 適用於甲骨文資料庫執行個體的 Amazon RDS，啟動

限制

- 來源帳戶中資料庫執行個體的相同屬性或組態會複製到 AMS 上的新目標資料庫執行個體。
- 此遷移方法中使用的 RFC 方法的功能有限，可支援適用於甲骨文的 Amazon RDS。您可以使用 AWS CloudFormation 範本來執行資料庫遷移，存取亞馬遜 RDS 適用於甲骨文的完整功能。
- 您可能會遇到應用程式中斷數小時，因為移轉必須在排定的停機時間內完成。在停機期間，您會停止來源帳戶中的資料庫執行個體，然後即時存取目標帳戶中的新資料庫執行個體。
- 此遷移方法不適用於將資料庫執行個體從一個 AWS 區域移轉到同一 AWS 帳戶內的另一個區域。

產品版本

- 甲骨文資料庫標準版 2 (SE2) 12.1.0.2.v2 執行個體及更新版本在 Amazon RDS for Oracle 上
- 不再支援 Amazon RDS 適用於甲骨文 (如需詳細資訊，請參閱 Amazon RDS for Oracle 文件中的亞馬遜 RDS)。

架構

源, 技術, 堆棧

- 適用於甲骨文的亞馬遜 RDS 上的甲骨文數據庫 SE2 12.1.0.2.v2 實例
- Amazon RDS 子網路群組
- Amazon RDS 選項組（如果需要）
- Amazon RDS 參數組（如果需要）
- Amazon Virtual Private Cloud（Amazon VPC）安全組
- AWS Key Management Service (AWS KMS) 搭配 AWS 受管金鑰或客戶受管金鑰
- AWS Identity and Access Management (IAM) 角色 (如有需要)

目標技術堆疊

- 適用於甲骨文的亞馬遜 RDS 上的甲骨文數據庫 SE2 12.1.0.2.v2 實例
- Amazon RDS 子網路群組
- Amazon RDS 選項組（如果需要）
- Amazon RDS 參數組（如果需要）
- Amazon VPC 安全組
- AWS Managed Services (AMS)

- AWS KMS 搭配 AWS 受管金鑰和客戶受管金鑰
- IAM 角色 (如果需要)

來源與目標移轉架構

下圖顯示將一個 AWS 帳戶中的 Amazon RDS for Oracle 文資料庫執行個體遷移到另一個使用 AMS 的 AWS 帳戶中的 Amazon RDS 適用 Oracle 資料庫執行個體。

該圖顯示以下工作流程：

1. 在來源帳戶中建立 Amazon RDS for Oracle 資料庫執行個體的資料庫快照。
2. 將快照複製到目標帳戶中的 AMS。
3. 從目標帳戶中的快照建立新的 Amazon RDS for Oracle 資料庫執行個體。

自動化和規模

您可以使用 CloudFormation 範本並在 AMS 中建立 RFC 來自動化和擴展移轉。 CloudFormation 可讓您使用 Amazon RDS for Oracle 的所有功能，包括在從快照建立 Amazon RDS for Oracle 資料庫執行個體時設定和還原資料庫執行個體的功能。

工具

- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Managed Services (AMS) 可協助您更有效率且安全地操作 AWS 基礎設施。

史诗

準備目標帳戶的切換

任務	描述	所需技能
建立自訂的 AWS KMS 金鑰。	1. 提升名為「建立 KMS 金鑰」的自動 RFC，以便從	AWS

任務	描述	所需技能
	目標帳戶建立自訂 KMS 金鑰。 2. 與來源帳戶共用您的自訂 KMS 金鑰。注意：您無法共用使用適用於 Amazon RDS 的預設 AWS 受管金鑰的 Oracle 資料庫執行個體的 Amazon RDS (`aws/rds`)。而是透過從 KMS 金鑰重新加密資料庫執行個體來共用資料庫執行個體。	
建立安全群組。	引發名為「建立安全群組」的自動 RFC，以從目標帳戶為您的 VPC 建立安全群組。 請務必指定下列項目： • 新安全性群組名稱 • TCP 和 UDP 輸入和輸出規則 • 標準標籤	AWS

任務	描述	所需技能
(選擇性) 檢閱您的 Amazon RDS 資源。	建立適用於 Oracle 資料庫的 Amazon RDS 執行個體時，會建立下列資源： • Amazon RDS 子網路群組 (根據子網路識別碼) • Amazon RDS 選項群組 (根據來源資料庫執行個體的快照) • Amazon RDS 參數群組 (根據資料庫執行個體的快照) 如果您想要檢閱建立資料庫執行個體時建立的 Amazon RDS 資源，則可以連線到 Oracle 資料庫執行個體，並在 Amazon RDS 主控台中尋找子網路群組、選項群組和參數群組。	AWS

切斷來源帳戶

任務	描述	所需技能
停止應用程式。	停止應用程式及其相依服務。您必須停止來源帳戶中資料庫的所有流量。	應用所有者
拍攝手動快照。	在來源帳戶中手動建立 Amazon RDS for Oracle 資料庫執行個體的資料庫快照。	AWS
停止資料庫執行個體。	停止適用於 Oracle 資料庫執行個體的亞馬遜 RDS。	AWS

任務	描述	所需技能
複製快照。	將資料庫快照複製到相同的來源帳戶，然後使用從目標帳戶共用的自訂 KMS 金鑰重新加密複製的資料庫快照檔案。	AWS
共用快照。	與目標帳戶共用新快照 (使用自訂 KMS 金鑰複製)。	AWS

切斷目標帳戶

任務	描述	所需技能
複製快照。	提升名為複製 RDS 快照的自動 RFC，將資料庫快照複製到相同的目標帳戶，並使用為重新加密而建立的預設 AWS 受管 KMS 金鑰。 若要將目標帳戶設為新快照的擁有者，並視需要啟用從快照建立的 Amazon RDS for Oracle 資料庫執行個體與選項群組建立關聯，這是必要的。	AWS
從快照建立資料庫執行個體。	提升名為「從快照建立資料庫」的自動化 RFC，以便從快照建立適用於 Oracle 資料庫的 Amazon RDS 執行個體。 請務必指定下列項目： • 在上一個步驟中建立的新快照 ID • VPC ID • 子網路 ID	AWS

任務	描述	所需技能
	• RDS 實例識別碼 • 標準標籤	
將執行個體附加至安全性群組，並進行設定更新。	1. 提出名為「其他更新」的手動 RFC，以連接您先前使用先前建立的 VPC 安全群組建立的 Amazon RDS for Oracle 資料庫執行個體。 2. 對 Amazon RDS for Oracle 資料庫執行個體組態進行任何其他變更。	AWS
測試資料庫執行個體。	登入託管在相同安全群組上的任何執行個體或應用程式伺服器，並使用 telnet 連接至 1521 連接埠，測試新的 Amazon RDS for Oracle 資料庫執行個體端點連線能力。如需詳細資訊，請參閱 Amazon RDS 文件中的連接至 Amazon RDS 資料庫執行個體。 備註：如果主要使用者登入身分證明可用，您可以從任何 SQL 用戶端 (例如 Oracle SQL 開發人員) 登入，以測試 Amazon RDS for Oracle 資料庫執行個體。	AWS

相關資源

- AWS Managed Services (AWS 文件)
- RFC 的運作方式 (AWS Managed Services 文件)
- 共用加密快照 (Amazon RDS 使用者指南)

- 如何與其他帳戶共用加密的 Amazon RDS 資料庫快照？ (AWS 知識中心)
- 什麼是 Amazon Relational Database Service（Amazon RDS）？ （Amazon RDS 用戶指南）
- Amazon RDS for Oracle（Amazon RDS 用戶指南）
- 使用 AMS 主控台 (AWS Managed Services 文件)

其他資訊

回滾遷移

如果您要復原移轉，請完成下列步驟：

1. 從目標帳戶引發手動 RFC (更新其他)，以刪除在目標帳戶中建立的資料庫堆疊。
2. 更新應用程式組態，以指向來源帳戶中的 Amazon RDS for Oracle 資料庫執行個體。
3. 在來源帳戶中啟動適用於甲骨文資料庫執行個體的 Amazon RDS。

將甲骨文輸出綁定變量遷移到 PostgreSQL 數據庫

由比卡什錢德拉魯（AWS）和維奈帕拉迪（AWS）創建

環境：PoC 或試點	來源：數據庫關係	目標：RDS/ Aurora
R 類型：重新平台	工作量：甲骨文	技術：資料庫；移轉
AWS 服務：Amazon Aurora; Amazon RDS; AWS SCT		

Summary

此模式顯示如何將 Oracle 資料庫OUT繫結變數遷移到下列其中一個與 PostgreSQL 相容的 AWS 資料庫服務：

- Amazon Relational Database Service 服務 (Amazon RDS)
- Amazon Aurora PostgreSQL-Compatible Edition

不支持OUT綁定變量。若要在 Python 陳述式中取得相同的功能，您可以建立自訂 PL/pgSQL 函式，改為使用GET和**SET**封裝變數。若要套用這些變數，此模式中提供的範例包裝函數指令碼使用 AWS Schema Conversion Tool (AWS SCT) 延伸套件。

注意：如果 Oracle EXECUTE IMMEDIATE 陳述式是最多可傳回一個資料列的SELECT陳述式，最佳作法是執行下列動作：

- 將OUT綁定變量（定義）放在子INTO句中
- 將IN綁定變量放在子USING句中

如需詳細資訊，請參閱 Oracle 文件中的立即執行陳述式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 內部部署資料中心中的 Oracle 資料庫 10g (或更新版本) 來源資料庫
- Amazon RDS for PostgreSQL 的資料庫執行個體或相容於 Aurora 的資料庫執行個體

架構

源, 技術, 堆棧

- 內部部署 Oracle 資料庫 10g (或更新版本) 資料庫

目標技術堆疊

- Amazon RDS for PostgreSQL 的資料庫執行個體或相容於 Aurora 的資料庫執行個體

目標架構

下圖顯示將 Oracle 資料庫0UT繫結變數遷移到 PostgreSQL 相容 AWS 資料庫的工作流程範例：

該圖顯示以下工作流程：

1. AWS SCT 會將來源資料庫結構描述和大部分自訂程式碼轉換為與目標 PostgreSQL 相容 AWS 資料庫的格式。
2. PL/pgSQL 函數會標記任何無法自動轉換的資料庫物件。然後會手動轉換標記的物件以完成移轉。

工具

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。
- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

使用自訂 PL/PgSQL 函式和 AWS SCT 遷移甲骨文輸出綁定變數

任務	描述	所需技能
Connect 到與 PostgreSQL 相容的 AWS 資料庫。	建立資料庫執行個體之後，您可以使用任何標準 SQL 用戶端應用程式連線到資料庫叢集中的資料庫。例如，您可以使用 pgAdmin 連線到資料庫執行個體。 如需詳細資訊，請參閱下列其中一項： • 在 Amazon RDS 使用者指南中連接到 Amazon RDS 資料庫執行個體 • 在 Amazon Aurora 使用者指南中連接到 Amazon Aurora 資料庫叢集	移民工程師
將此模式中的示例包裝函數腳本添加到目標數據庫的主模式中。	從此模式的其他資訊區段複製 PL/pgSQL 包裝函式指令碼範例。然後，將函數添加到目標數據庫的主模式中。 如需詳細資訊，請參閱 PostgreSQL 文件中的 CREATE FUNCTION。	移民工程師
(選擇性) 更新目標資料庫主要結構描述中的搜尋路徑，以便包含 Test_pg 結構描述。	若要改善效能，您可以更新 PostgreSQL 搜尋路徑變數，使其包含 TEST_ PG 結構描述名稱。如果您在搜尋路徑中包含結構描述名稱，則無論何時	移民工程師

任務	描述	所需技能
	呼叫 PL/pgSQL 函數，都不需要指定名稱。 如需詳細資訊，請參閱 PostgreSQL 文件中的〈結構描述搜尋路徑〉第 5.9.3 節。	

相關資源

- AWS Schema Conversion Tool
- 輸出連結變數 (Oracle 文件集)
- 使用連結變數來改善 SQL 查詢效能 (Oracle 部落格)

其他資訊

PL/PGSQL 函數示例

```
/* Oracle */

CREATE or replace PROCEDURE test_pg.calc_stats_new1 (
                                                      a NUMBER,
                                                      b NUMBER,
                                                      result out NUMBER
                                                    )

IS
BEGIN
result:=a+b;
END;
/
/* Testing */
set serveroutput on
DECLARE
  a NUMBER := 4;
  b NUMBER := 7;
  plsql_block VARCHAR2(100);
  output number;
```

```
BEGIN
  plsql_block := 'BEGIN test_pg.calc_stats_new1(:a, :b,:output); END;';
  EXECUTE IMMEDIATE plsql_block USING a, b,out output;  -- calc_stats(a, a, b, a)
  DBMS_OUTPUT.PUT_LINE('output:'||output);
END;

output:11


PL/SQL procedure successfully completed.

--Postgres--

/* Example : 1 */
CREATE OR REPLACE FUNCTION test_pg.calc_stats_new1(
                                                    w integer,
                                                    x integer
                                                  )
RETURNS integer
AS
$BODY$
begin
      return w + x ;
end;
$BODY$
LANGUAGE  plpgsql;


CREATE OR REPLACE FUNCTION aws_oracle_ext.set_package_variable(
                                                                 package_name name,
                                                                 variable_name name,
                                                                 variable_value
 anyelement
                                                               )
    RETURNS void
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
begin
    perform set_config
      ( format( '%s.%s',package_name, variable_name )
      , variable_value::text
```

```
      , false );
  end;
$BODY$;

CREATE OR REPLACE FUNCTION aws_oracle_ext.get_package_variable_record(
                                                                        package_name
 name,
                                                                        record_name name
                                                                      )
RETURNS text
LANGUAGE 'plpgsql'
    COST 100
    VOLATILE
AS $BODY$
begin
    execute 'select ' || package_name || '$Init()';

    return aws_oracle_ext.get_package_variable
        (
         package_name := package_name
        , variable_name := record_name || '$REC' );
end;
$BODY$;

--init()--
CREATE OR REPLACE FUNCTION test_pg.init()
RETURNS void
AS
$BODY$
BEGIN
if aws_oracle_ext.is_package_initialized('test_pg' ) then
      return;
    end if;
    perform aws_oracle_ext.set_package_initialized
      ('test_pg' );
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_output', NULL::INTEGER);
PERFORM aws_oracle_ext.set_package_variable('test_pg', 'v_status', NULL::text);
END;
$BODY$
LANGUAGE  plpgsql;

/* callable for 1st Example */

DO $$
```

```
declare
v_sql text;
v_output_loc int;
a integer :=1;
b integer :=2;
BEGIN
perform  test_pg.init();
--raise notice 'v_sql %',v_sql;
execute 'do $a$ declare v_output_l int; begin select * from test_pg.calc_stats_new1('||
a||','||b||') into v_output_l;
PERFORM aws_oracle_ext.set_package_variable(''test_pg'', ''v_output'', v_output_l) ;
 end; $a$'  ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
raise notice 'v_output_loc %',v_output_loc;
END ;
$$

/*In above Postgres example we have set the value of v_output using v_output_l in the
 dynamic anonymous block to mimic the
  behaviour of oracle out-bind variable .*/

--Postgres Example : 2 --
CREATE OR REPLACE FUNCTION test_pg.calc_stats_new2(
 w integer,
 x integer,
inout status text,
 out result integer)
AS
$BODY$
DECLARE
begin
result :=  w + x ;
status := 'ok';
end;
$BODY$
LANGUAGE  plpgsql;

/* callable for 2nd Example */
DO $$
declare
v_sql text;
v_output_loc int;
v_staus text:= 'no';
a integer :=1;
```

```
b integer :=2;
BEGIN
perform  test_pg.init();
execute 'do $a$ declare v_output_l int; v_status_l text; begin select * from
 test_pg.calc_stats_new2(''||a||'',''||b||'','''||v_staus||''') into v_status_l,v_output_l;
PERFORM aws_oracle_ext.set_package_variable(''test_pg'', ''v_output'', v_output_l) ;
PERFORM aws_oracle_ext.set_package_variable(''test_pg'', ''v_status'', v_status_l) ;
end; $a$'  ;
v_output_loc := aws_oracle_ext.get_package_variable('test_pg', 'v_output');
v_staus := aws_oracle_ext.get_package_variable('test_pg', 'v_status');
raise notice 'v_output_loc %',v_output_loc;
raise notice 'v_staus %',v_staus;
END ;
$$
```

使用具有相同主機名稱的 SAP 高速鐵路將 SAP HANA 遷移到 AWS

創建者普拉迪普普利揚帕塔（AWS）

環境：生產	來源：內部部署 SAP HANA 資料庫	目標：AWS 上的 SAP HANA 資料庫
R 類型：重新主機	工作負載：SAP	技術：資料庫；移轉
AWS 服務：AWS Client VPN；AWS Direct Connect；Amazon EBS		

Summary

SAP HANA 遷移至 Amazon Web Services (AWS) 可以使用多種選項執行，包括備份和還原、匯出和匯入，以及 SAP HANA 系統複寫 (HSR)。特定選項的選擇取決於來源與目標 SAP HANA 資料庫之間的網路連線、來源資料庫的大小、停機時間考量及其他因素。

如果來源和目標系統之間存在穩定的網路，以及整個資料庫 (SAP HANA 資料庫複寫快照) 之間存在穩定的網路，則可在 1 天內完全複寫 SAP HANA 工作負載至 AWS 的 SAP HSR 選項，如 SAP HSR 的網路輸送量要求，則可以在 1 天內完全複寫。此方法的停機時間需求僅限於在目標 AWS 環境、SAP HANA DB 備份和移轉後工作上執行接管。

SAP HSR 支援使用不同的主機名稱 (對應至不同 IP 位址的主機名稱) 來
複製主要系統或來源系統與次要系統或目標系統之間的流量。您可以在
中global.ini的[system_replication_hostname_resolution]章節下定義這些特定的主機名稱集來執行此操作。在此段落中，必須在每個主機上定義主要站台和次要站台的所有主機。如需詳細的設定步驟，請參閱 SAP 文件。

此設定的其中一個重要優點是，主要系統中的主機名稱必須與次要系統中的主機名稱不同。否則，可以觀察到以下錯誤。

- `"each site must have a unique set of logical hostnames"`
- `"remoteHost does not match with any host of the source site. All hosts of source and target site must be able to resolve all hostnames of both sites correctly"`

不過，在目標 AWS 環境中使用相同的 SAP HANA DB 主機名稱，可以減少移轉後步驟的數目。

此模式提供了在使用 SAP HSR 選項時，在來源和目標環境上使用相同主機名稱的因應措施。使用此模式，您可以使用 SAP HANA 主機名稱重新命名選項。您可以將暫時主機名稱指派給目標 SAP HANA 資料庫，以促進 SAP HSR 的主機名稱唯一性。移轉完成目標 SAP HANA 環境的接管里程碑之後，您可以將目標系統主機名稱還原回來源系統的主機名稱。

先決條件和限制

先決條件

- 一個活躍的 AWS 帳戶。
- 具有虛擬私人網路 (VPN) 端點或路由器的虛擬私有雲 (VPC)。
- AWS Client VPN 或 AWS Direct Connect 配置為將文件從源傳輸到目標。
- 來源環境和目標環境中的 SAP HANA 資料庫。在相同 SAP HANA 平台版本中，目標 SAP HANA 資料庫修補程式層級應等於或高於來源 SAP HANA 資料庫修補程式層級。例如，無法在 HANA 1.0 和 HANA 2.0 系統之間設定複寫。如需詳細資訊，請參閱 SAP 注意事項中的問題 15 — 常見問題集：SAP HANA 系統複寫。
- 目標環境中的 SAP 應用程式伺服器。
- 目標環境中的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區。

限制

下列 SAP 文件清單涵蓋了與此因應措施相關的已知問題，包括有關 SAP HANA 動態分層和向外延展移轉的限制：

- 2956397 — 重新命名資料庫系統失敗
- 2222694 — 嘗試重命名 HANA 系統時，出現以下錯誤「源文件不屬於原始用戶（uid = xxxx）」
- 2607227 — 高畫質管理:註冊名稱系統:重新命名 SAP HANA 執行個體失敗
- 2630562 — HANA 主機名稱重新命名失敗且 HANA 無法啟動
- 2935639 — 寄存器未使用 global.ini 部分中系統複製 _ 主機名稱解析度下指定的主機名稱
- 2710211 — 錯誤：來源系統和目標系統的邏輯主機名稱重疊
- 2693441 — 由於發生錯誤，無法重新命名 SAP HANA 系統
- 2519672 — HANA 主要和次要具有不同的系統 PKI SSFS 數據和密鑰或無法檢查
- 2457129 — 當動態分層是景觀的一部分時，不允許 SAP HANA 系統主機重新命名

- 2473002 — 使用 HANA 系統複寫移轉向外延展系統 (SAP 對於向外擴充 SAP HANA 系統使用此主機名稱重新命名方法沒有提供任何限制。 但是，必須在每個單獨的主機上重複此程序。 其他向外延展移轉限制也適用於此方法。)

產品版本

- 此解決方案適用於 SAP HANA 資料庫平台 1.0 和 2.0 版。

架構

來源設定

SAP HANA 資料庫已安裝在來源環境中。所有 SAP 應用程式伺服器連線和資料庫介面都使用相同的主機名稱來連線用戶端。下圖顯示範例來源主機名稱hdbhost及其對應的 IP 位址。

目標設定

AWS 雲端 目標環境使用相同的主機名稱來執行 SAP HANA 資料庫。AWS 上的目標環境包括下列項目：

- SAP HANA 資料庫
- SAP 應用伺服器
- EBS 磁碟區

中間配置

在下圖中，暫時重新命名 AWS 目標環境上的主機名稱，以temp-host便來源和目標上的主機名稱是唯一的。移轉完成目標環境的接管里程碑之後，會使用原始名稱重新命名目標系統虛擬主機名稱。hdbhost

中繼組態包括下列其中一個選項：

- AWS Client VPN 使用 Client VPN 端點
- AWS Direct Connect 連接到路由器

AWS 目標環境中的 SAP 應用程式伺服器可以在複製設定之前或接管之後安裝。不過，在複製設定之前先安裝應用程式伺服器，有助於減少安裝期間、設定高可用性及備份期間的停機時間。

工具

AWS 服務

- AWS Client VPN是受管理的用戶端型 VPN 服務，可讓您安全地存取內部部署網路中的資 AWS 源和資源。
- AWS Direct Connect透過標準乙太網路光纖纜線，將您的內部網路連結至某個 AWS Direct Connect 位置。透過此連線，您可以直接建立公用的虛擬介面 AWS 服務，繞過網路路徑中的網際網路服務供應商。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。EBS 磁碟區的行為與未格式化的原始區塊型儲存設備相似。您可以將這些磁碟區做為裝置，掛載在您的執行個體上。

其他工具

- SAP 應用程式伺服器 — SAP 應用程式伺服器為程式設計人員提供表達業務邏輯的方式 SAP 應用程序服務器執行基於業務邏輯的數據處理。實際數據存儲在數據庫中，這是一個單獨的組件。
- SAP HANA 駕駛艙和 SAP HANA 工作室 — SAP HANA 駕駛艙和 SAP HANA 工作室都為 SAP HANA 資料庫提供管理介面。在 SAP HANA 工作室中，SAP HANA 管理主控台是一種系統檢視，可為 SAP HANA 資料庫管理提供相關內容。
- SAP HANA 系統複寫 — SAP HANA 系統複製 (SAP 高鐵) 是 SAP 提供用於複製 SAP HANA 資料庫的標準程序。SAP 高鐵所需的可執行檔是 SAP HANA 伺服器核心本身的一部分。

史诗

準備來源和目標環境

任務	描述	所需技能
安裝和設定 SAP HANA 資料庫。	在來源和目標環境中，請確定 SAP HANA 資料庫已根據 SAP HANA 針對最佳實務進行安裝	SAP 基礎管理

任務	描述	所需技能
	和設定。如需詳細資訊，請參閱上的 SAP HANA AWS。	
對應 IP 位址。	在目標環境中，請確定暫存主機名稱已指派給內部 IP 位址。 1. 透過導覽至 EC2、執行個體、動作、聯網、管理 IP 地址、指派新 IP 地址，將次要 IPv4 地址指派給 AWS 管理主控台上的 EC2 執行個體。 2. 若要將相同的位址指派給 EC2 網路介面卡 (NIC)，請從作業系統以 root 使用者身分執行命令`ip addr add <IP>/32 dev eth0`，並以步驟 1 中的 IP 位址取代`<IP>`。	AWS 管理
解析目標主機名稱。	在次要 SAP HANA 資料庫上，透過更新檔案中的相關主機名稱，確認已為 SAP HANA 複寫網路解析兩個主機名稱 (`hdbhost`和`temp-host`)。`/etc/hosts`	Linux 系統管理
備份來源和目標 SAP HANA 資料庫。	使用 SAP HANA 工作室或 SAP HANA 駕駛艙在 SAP HANA 資料庫上執行備份。	SAP 基礎管理

任務	描述	所需技能
交換系統 PKI 證書。	(僅適用於 SAP HANA 2.0 及更新版本) 在主要和次要資料庫之間的檔案系統 (SSFS) 存放區中，系統公開金鑰基礎結構 (PKI) 安全存放區中交換憑證。如需詳細資訊，請參閱 SAP 注意事項 2369981 — 使用 SAP HANA 系統複寫進行驗證的必要設定步驟。	SAP 基礎管理

重新命名目標 SAP HANA 資料庫

任務	描述	所需技能
停止目標用戶端連線。	在目標環境中，關閉 SAP 應用程式伺服器和其他用戶端連線。	SAP 基礎管理
將目標 SAP HANA 資料庫重新命名為暫存主機名稱。	1. 以根使用者身分，使用常駐，將目標 SAP HANA DB 主機名稱重新命名為暫時主機名稱hdblcm。 `root $> cd /hana/shared/<SID/hdblcm` `root $> ./hdblcm` 2. 選擇選項9 | rename_system | Rename the SAP HANA Database System。 3. 提供新名稱： temp-host 。 4. 您可以根據需要驗證其他選項。不過，請確定您不要將	SAP 基礎管理

任務	描述	所需技能
	主機重新命名與 SID 變更混淆 (SAP 注意事項 2598814 — hdblcm：SID 重新命名失敗)。 SAP HANA 資料庫停止和啟動將由控制hdblcm。	
指派複製網路。	在來源系統global.ini 檔案的標[system_replication_hostname_resolution] 頭下，提供來源和目標複製網路詳細資訊。然後將條目複製到目標系統上的global.ini 文件中。	SAP 基礎管理
啟用主要的複製。	若要在來源 SAP HANA 資料庫上啟用複寫，請執行下列命令。 `hdbnsutil -sr_enable --name=siteA`	SAP 基礎管理

任務	描述	所需技能
將目標 SAP HANA 資料庫註冊為次要系統。	若要將目標 SAP HANA 資料庫註冊為 SAP HSR 來源的次要系統，請選擇非同步複寫。 `(sid)adm $> HDB stop` `(sid)adm $> hdbnsutil -sr_register –name=siteB –remotehost=hdbhost /` `--remoteInstance=00 –replicationMode=async –operationMode=logreplay` `(sid)adm $> HDB start` 或者，您可以選擇註冊–online選項。在這種情況下，您不需要停止並啟動 SAP HANA 數據庫。	SAP 基礎管理
驗證同步處理。	在來源 SAP HANA DB 上，確認所有記錄檔都已套用至目標系統 (因為它是非同步複寫)。 若要驗證複製，請在來源上執行下列命令。 `(sid)adm $> cdpy` `(sidadm $> python systemReplicationStatus.py`	SAP 基礎管理
關閉源 SAP 應用程序和 SAP HANA 數據庫。	在移轉切換期間，請執行來源系統 (SAP 應用程式和 SAP HANA 資料庫) 的關閉。	SAP 基礎管理

任務	描述	所需技能
對目標執行接管。	若要在 AWS 上對目標執行接管，請執行命令`hdbnsutil -sr_takeover` 。	SAP 基礎管理
在目標 SAP HANA 資料庫上，關閉複寫功能。	若要清除複寫中繼資料，請執行指令來停止目標系統上的複寫`hdbnsutil -sr_disable` 。 注意：這是根據 SAP 注意事項 2693441 — 由於錯誤而無法重新命名 SAP HANA 系統。	SAP 基礎管理
備份目標 SAP 花數據庫。	成功接管之後，我們建議您執行完整的 SAP HANA 資料庫備份。	SAP 基礎管理

還原到目標系統中的原始主機名稱

任務	描述	所需技能
將目標 SAP HANA 資料庫主機名稱還原為原始主機名稱。	1. 若要將目標 SAP HANA DB 主機名稱還原為原始虛擬主機名稱，請使用常駐`hdblcm`。 `root $> cd /hana/shared/<SID>/hdblcm` `root $> ./hdblcm` 2. 選擇選項`9 | rename_system | Rename the SAP HANA Database System`。 3. 提供新名稱：`hdbhost`。	SAP 基礎管理

任務	描述	所需技能
	您可以根據需要驗證其他選項。不過，請確定您不要將主機重新命名與 SID 變更混淆 (SAP 注意事項 2598814 — hdblcm：SID 重新命名失敗)。	
調整高清商店。	調整指向來源`hdbuserstore` 詳細資料的`schema/user` 詳細資料。如需詳細步驟，請參閱 SAP 文件。 若要驗證此步驟，請執行命令`R3trans -d`。結果應反映成功連線至 SAP HANA 資料庫。	SAP 基礎管理
啟動用戶端連線。	在目標環境中，啟動 SAP 應用程式伺服器和其他用戶端連線。	SAP 基礎管理

相關資源

SAP 參考資料

SAP 經常更新 SAP 文件參考資料。要保持最新狀態，請參閱 SAP 注意事項 2407186 — SAP HANA 高可用性的操作指南和白皮書。

其他 SAP 注意事項

- 2550327 — 如何重新命名
- 常見問題解答：哈娜系統複製
- 2078425 — 針對 SAP HANA 平台生命週期管理工具的疑難排解注意事項
- 2592227 — 哈納系統中的 FQDN 後綴更改
- 2048681 — 在沒有安全殼層或根登入資料的多主機系統上執行 SAP HANA 平台生命週期管理管理工作

SAP 文件

- 系統複製網路連線
- 系統複製的主機名稱解析

AWS 參考

- 將 SAP HANA 從其他平台遷移到 AWS

其他資訊

hdblcm作為主機名稱重新命名活動一部分所執行的變更會合併在下列詳細記錄中。

使用分散式可用性群組將 SQL 伺服器遷移到 AWS

創建者：普拉芬·馬薩拉（AWS）

來源：SQL 伺服器內部部署	目標：EC2 上的 SQL 伺服器	R 類型：重新主機
環境：PoC 或試點	技術：資料庫；移轉	工作量：Microsoft
AWS 服務：Amazon EC2		

Summary

Microsoft SQL Server 永遠在可用性群組提供 SQL Server 的高可用性 (HA) 和災難復原 (DR) 解決方案。可用性群組包含接受讀取/寫入流量的主要複本，以及最多八個接受讀取流量的次要複本所組成。可用性群組是在具有兩個或多個節點的 Windows 伺服器容錯移轉叢集 (WSFC) 上設定。

Microsoft SQL Server 永遠在分散式可用性群組提供解決方案，以設定兩個獨立的 WFSC 之間的兩個個別的可用性群組。屬於分散式可用性群組的可用性群組不一定要位於相同的資料中心。一個可用性群組可以位於內部部署，而另一個可用性群組則可以位於不同網域中的 Amazon Web Services (AWS) 雲端上的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上。

此模式概述了使用分散式可用性群組將屬於現有可用性群組一部分的現場部署 SQL Server 資料庫遷移至具有在 Amazon EC2 上設定可用性群組的 SQL Server 的步驟。遵循此模式，您可以將資料庫遷移到 AWS 雲端，並在切換期間將停機時間降至最低。資料庫在切換後立即在 AWS 上具有高可用性。您也可以使用此模式將基礎作業系統從現場部署變更為 AWS，同時保留相同版本的 SQL Server。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Direct Connect 或 AWS Site-to-Site VPN
- 在現場部署和 AWS 上的兩個節點上安裝的相同版本的 SQL Server

產品版本

- SQL 伺服器版本 2016 及更新版本

- SQL Server Enterprise Edition

架構

源, 技術, 堆棧

- Microsoft SQL Server 資料庫與永遠在可用性群組內部部署

目標技術堆疊

- 在 AWS 雲端上的 Amazon EC2 上具有永遠在線可用性群組的 Microsoft SQL 伺服器資料庫

移轉架構

術語

- 水務委員會 1 — 樓宇內的水務足球會
- WSFC 2 — AWS 雲端上的 WSFC
- AG 1 — 第一個可用性群組，這是在 WSFC 1
- AG 2 — 第二個可用性群組，這是在 WSFC 2 中
- SQL Server 主要複本 — AG 1 中的節點，被視為所有寫入的全域主要複本
- SQL Server 轉寄站 — AG 2 中的節點，以非同步方式從 SQL Server 主要複本接收資料
- SQL Server 次要複本 — AG 1 或 AG 2 中的節點，可從主要複本或轉寄站同步接收資料

工具

- AWS Direct Connect — AWS Direct Connect 透過標準乙太網路光纖電纜將您的內部網路連結到 AWS Direct Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，繞過網路路徑中的網際網路服務供應商。
- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。
- AWS Site-to-Site VPN — AWS Site-to-Site VPN 支援建立 site-to-site 虛擬私人網路 (VPN)。您可以設定 VPN 在 AWS 上啟動的執行個體和自己的遠端網路之間傳遞流量。

- Microsoft SQL 服務器管理工作室-Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器基礎設施的集成環境。它提供了一個用戶界面和一組具有與 SQL Server 交互的豐富腳本編輯器的工具。

史诗

在 AWS 上設定第二個可用性群組

任務	描述	所需技能
在 AWS 上建立 WSFC。	在具有兩個 HA 節點的 Amazon EC2 執行個體上建立 WSFC 2。您將使用此容錯移轉叢集在 AWS 上建立第二個可用性群組 (AG 2)。	系統管理員、 SysOps 管理員
在 WSFC 2 上建立第二個可用性群組。	使用 SSMS，在 WSFC 2 中的兩個節點上建立 AG 2。WSFC 2 中的第一個節點將充當轉發器。WSFC 2 中的第二個節點將充當 AG 2 的次要複本。 在此階段，AG 2 中沒有可用的資料庫。這是設定分散式可用性群組的起點。	DBA, 開發人員
在 AG 2 上建立沒有復原選項的資料庫。	備份內部部署可用性群組 (AG 1) 上的資料庫。 將資料庫還原至轉寄站和 AG 2 的次要複本，而不使用復原選項。還原資料庫時，請指定具有足夠磁碟空間供資料庫資料檔和記錄檔使用的位置。 在此階段，資料庫處於還原狀態。它們不是 AG 2 或分散式	DBA, 開發人員

任務	描述	所需技能
	可用性群組的一部分，而且不會同步處理。	

設定分散式可用性群組

任務	描述	所需技能
在 AG 1 上建立分散式可用性群組。	若要在 AG 1 上建立分散式可用性群組，請CREATE AVAILABILITY GROUP 搭配選DISTRIBUTED 項使用。 1. 使用 AG 1 和 AG 2 的LISTENER_URL 端點位址。 2. 對於AVAILABILITY-MODE，使用ASYNCHRONOUS_COMMIT 以避免網路延遲 (如果有的話)。這不會影響資料庫的效能。 3. 對於 FAILOVER_MODE ，請使用 MANUAL。這是唯一可與分散式可用性群組搭配使用的可用性模式。 4. 若要在 AG 2 上手動還原資料庫，並對較大的資料庫擁有更多控制權，請使MANUAL用SEEDING_MODE .	DBA, 開發人員
在 AG 2 上建立分散式可用性群組。	若要在 AG 2 上建立分散式可用性群組，請ALTER AVAILABILITY GROUP 搭配選DISTRIBUTED 項使用。	DBA, 開發人員

任務	描述	所需技能
	1. 使用 AG 1 和 AG 2 的LISTENER_URL 端點位址。 2. 對於AVAILABILITY-MODE，使用ASYNCHRONOUS_COMMIT 以避免網路延遲 (如果有的話)。這不會影響資料庫的效能。 3. 對於 FAILOVER_MODE ，請使用 MANUAL。這是唯一可與分散式可用性群組搭配使用的可用性模式。 4. 若要在 AG 2 上手動還原資料庫，並對較大的資料庫擁有更多控制權，請使MANUAL用SEEDING_MODE . 分散式可用性群組是在 AG 1 和 AG 2 之間建立的。 AG 2 中的資料庫尚未設定為參與從 AG 1 到 AG 2 的資料流程。	

任務	描述	所需技能
將資料庫新增至 AG 2 上的轉寄站和次要複本。	使用 AG 2 上的轉寄站和次要複本中`ALTER DATABASE`的`SET HADRAVAILABILITY GROUP`選項，將資料庫新增至分散式可用性群組。 這會啟動 AG 1 和 AG 2 上的資料庫之間的非同步資料流程。 全域主要執行寫入作業、同步傳送資料至 AG 1 上的次要複本，並以非同步方式將資料傳送至 AG 2 上的轉寄站。AG 2 上的轉寄站會同步傳送資料至 AG 2 上的次要複本。	DBA, 開發人員

監控 AG 1 和 AG 2 之間的非同步資料流

任務	描述	所需技能
使用 DMV 和 SQL 伺服器記錄檔。	使用動態管理檢視 (DMV) 和 SQL Server 記錄檔來監視兩個可用性群組之間的資料流程狀態。 有興趣監視的 DMV 包括`sys.dm_hadr_availability_replica_states` 和`sys.dm_hadr_automatic_seeding` 。	DBA, 開發人員

任務	描述	所需技能
	如需轉寄站同步處理的狀態，請監視轉寄站上 SQL Server 記錄檔中的同步處理狀態。	

執行最終移轉的切換活動

任務	描述	所需技能
停止主要複本的所有流量。	在 AG 1 中停止主要複本的傳入流量，這樣資料庫就不會發生寫入活動，而且資料庫已準備好進行移轉。	應用程式擁有者、開
變更 AG 1 上分散式可用性群組的可用性模式。	在主要複本上，將分散式可用性群組的可用性模式設定為同步。 將可用性模式變更為同步之後，資料會從 AG 1 中的主要複本同步傳送至 AG 2 中的轉寄站。	DBA, 開發人員
檢查兩個可用性群組中的 LSNS。	檢查 AG 1 和 AG 2 中的最後一個記錄序號 (LSNS)。因為 AG 1 中的主要複本中沒有發生任何寫入，因此資料會同步處理，而且兩個可用性群組的最後一個 LSNS 都應該相符。	DBA, 開發人員
將 AG 1 更新為次要角色。	當您將 AG 1 更新為次要角色時，AG 1 會遺失主要複本角色且不接受寫入，而且兩個可用性群組之間的資料流程會停止。	DBA, 開發人員

容錯移轉至第二個可用性群組

任務	描述	所需技能
手動容錯移轉至 AG 2。	在 AG 2 的轉寄站上，變更分散式可用性群組以允許資料遺失。因為您已經檢查並確認 AG 1 和 AG 2 上的最後一個 LSNS 相符，因此資料遺失並不是問題。 當您允許 AG 2 中轉寄站上的資料遺失時，AG 1 和 AG 2 的角色會變更： • AG 2 會成為具有主要複本和次要複本的可用性群組。 • AG 1 會成為具有轉寄站和次要複本的可用性群組。	DBA, 開發人員
變更 AG 2 上分散式可用性群組的可用性模式。	在 AG 2 的主要複本上，將可用性模式變更為非同步。 這會將資料移動從 AG 2 變更為 AG 1，從同步變更為非同步。為了避免 AG 2 和 AG 1 之間的網路延遲 (如果有的話)，必須執行此步驟，而且不會影響資料庫的效能。	DBA, 開發人員
開始將流量傳送到新的主要複本。	更新連接字串，以使用 AG 2 上的接聽程式 URL 端點傳送流量至資料庫。 AG 2 現在接受寫入並將資料傳送至 AG 1 中的轉寄站，並將資料傳送至 AG 2 中自己的次	應用程式擁有者、開

任務	描述	所需技能
	要複本。資料會以非同步方式從 AG 2 移至 AG 1。	

進行切換後的活動

任務	描述	所需技能
將分散式可用性群組卸除在 AG 2 上。	監視移轉的規劃時間長度。然後將分散式可用性群組放在 AG 2 上，以移除 AG 2 和 AG 1 之間的分散式可用性群組設定。這會移除分散式可用性群組組態，以及從 AG 2 到 AG 1 停止的資料流程。 此時，AG 2 在 AWS 上具有高可用性，其主要複本可在同一個可用性群組中進行寫入和次要複本。	DBA, 開發人員
解除委任內部部署伺服器。	解除委任屬於 AG 1 一部分的 WSFC 1 中的內部部署伺服器。	系統管理員、 SysOps 管理員

相關資源

- 分散可用性群組
- SQL 文件：分散式可用性群組
- SQL 文件：永遠在使用可用性群組：高可用性和災難回復解決方案

使用 SharePlex 和 AWS DMS 從甲骨文 8i 或 9i 遷移到適用於甲骨文的亞馬遜 RDS

由拉姆雅基尼 (AWS) 創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS
R 類型：重新平台	工作負載：開放原始碼；	技術：資料庫、雲端原生、移轉
AWS 服務：AWS DMS; Amazon RDS		

Summary

此模式說明如何將現場部署 Oracle 8i 或 9i 資料庫遷移到適用於 Oracle 資料庫的 Amazon Relational Database Service 服務 (Amazon RDS)。您可以使用此模式透過使用 Quest SharePlex 進行同步複寫，以減少停機時間來完成移轉作業。

您必須使用中繼 Oracle 資料庫執行個體進行遷移，因為 AWS Database Migration Service (AWS DMS) 不支援 Oracle 8i 或 9i 做為來源環境。您可以使用 SharePlex 7.6.3，從舊版 Oracle 資料庫複製到更新的 Oracle 資料庫版本。中繼 Oracle 資料庫執行個體可做為 SharePlex 7.6.3 的目標相容，並作為 AWS DMS 或更新版本的來源提供支援。 SharePlex此支援可繼續將資料複寫到適用於 Oracle 的亞馬遜 RDS 目標環境。

請考慮數種已淘汰的資料類型和功能可能會影響從 Oracle 8i 或 9i 移轉至最新版本的 Oracle 資料庫。為了減輕這種影響，此模式使用 Oracle 11.2.0.4 作為中繼資料庫版本，以協助在遷移到 Amazon RDS for Oracle 目標環境之前協助優化結構描述程式碼。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署環境中的來源 Oracle 8i 或 9i 資料庫
- 甲骨文資料庫 12c 版本 2 (12CR2)，用於在 Amazon Elastic Compute Cloud (Amazon EC2) 上進行安裝

- 任務 SharePlex 7.6.3（商業級）

限制

- 適用於 Oracle 的 RDS 限制

產品版本

- 來源資料庫的甲骨文 8i 或 9i
- 用於暫存資料庫的甲骨文 12CR2（必須與甲骨文版本的亞馬遜 RDS 相匹配）
- 目標資料庫的甲骨文 12CR2 或更新版本 (Amazon RDS for Oracle)

架構

源, 技術, 堆棧

- 甲骨文 8i 或 9i 数据库
- SharePlex

目標技術堆疊

- Amazon RDS for Oracle

移轉架構

下圖顯示如何將 Oracle 8i 或 9i 資料庫從現場部署環境遷移到 AWS 雲端中的 Amazon RDS for Oracle 文資料庫執行個體。

該圖顯示以下工作流程：

1. 使用存檔日誌模式、強制記錄日誌和補充記錄日誌來啟用 Oracle 來源資料庫。
2. 使用復原管理程式 (RMAN) 復原和回溯 _SCN，從 Oracle 來源資料庫 point-in-time 回復 Oracle 安裝資料庫。
3. 設定 SharePlex 使用 FLASHBACK_SCN (用於 RMAN)，從 Oracle 來源資料庫讀取重做日誌。
4. 啟動 SharePlex 複製，將資料從 Oracle 來源資料庫同步到 Oracle 安裝資料庫。

5. 使用 EXPDP 和 IMPDP 還原亞馬遜 RDS 適用於甲骨文目標資料庫。FLASHBACK_SCN
6. 使用 FLASHBACK_SCN (在 EXPDP 中使用)，將 AWS DMS 及其來源任務設定為甲骨文預備資料庫，將適用於甲骨文的 Amazon RDS 設定為目標資料庫。
7. 啟動 AWS DMS 任務，將資料從 Oracle 預備資料庫同步到 Oracle 目標資料庫。

工具

- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- Quest SharePlex 是 Oracle 對 Oracle 的資料複製工具，可在停機時間降至最低且不會遺失資料的情況下移動資料。
- 復原管理員 (RMAN) 是一種 Oracle 資料庫從屬端，可在您的資料庫上執行備份和復原作業。它大大簡化了備份，還原和恢復數據庫文件。
- 「資料汲取匯出」可協助您將資料和中繼資料上傳至一組稱為傾印檔案集的作業系統檔案。傾印檔案集只能由「資料汲取匯入」公用程式或 DBMS_ DATAPUMP 套件匯入。

史诗

在 Amazon EC2 上設置 SharePlex 和甲骨文測試數據庫

任務	描述	所需技能
建立 EC2 執行個體。	1. 建立 EC2 執行個體。 2. 在 EC2 執行個體上安裝甲骨文 12CR2 作為甲骨文預備資料庫。	甲骨文管理
準備暫存資料庫。	透過從 Oracle 8i 或 9i 資料庫來源環境取得 RMAN 備份，準備 Oracle 安裝資料庫以進行還原作為 Oracle 12CR2 上的升級。	甲骨文管理

任務	描述	所需技能
	如需詳細資訊，請參閱 Oracle 文件中的「Oracle 9i 復原管理員使用者指南」和「資料庫 Backup 與復原使用者指南」。	
配置 SharePlex。	將 SharePlex 來源設定為現場部署 Oracle 8i 或 9i 資料庫，並將目標設定為在 Amazon EC2 上託管的 Oracle 12CR2 暫存資料庫。	SharePlex、甲骨文管理

將 Amazon RDS for Oracle 設置為您的目標環境

任務	描述	所需技能
建立 Oracle 資料庫執行個體。	建立 Amazon RDS for Oracle 資料庫，然後將甲骨文 12CR2 連接到資料庫。 如需詳細資訊，請參閱 Amazon RDS 說明文件中的建立 Oracle 資料庫執行個體和連接 Oracle 資料庫執行個體上的資料庫。	DBA
從預備資料庫還原適用於甲骨文的亞馬遜 RDS。	1. 使FLASHBACK_SCN 用從 Oracle 暫存資料庫伺服器取得 EXPDP 備份。 2. 從預備資料庫還原適用於甲骨文的亞馬遜 RDS。 如需詳細資訊，請參閱甲骨文件中的 54 DBMS_DATAPUMP。	DBA

設定 AWS DMS

任務	描述	所需技能
建立資料庫的端點。	為 Oracle 暫存資料庫建立來源端點，並為適用於 Oracle 資料庫的 Amazon RDS 建立目標端點。 如需詳細資訊，請參閱如何使用 AWS DMS 建立來源或目標端點？ 在 AWS 知識中心。	DBA
建立複寫執行個體。	使用 AWS DMS 將 Oracle 暫存資料庫的複寫執行個體啟動到 Amazon RDS for Oracle 文資料庫。 如需詳細資訊，請參閱如何建立 AWS DMS 複寫執行個體？在 AWS 知識中心。	DBA
建立並開始複寫工作。	使FLASHBACK_SCN 用 EXPDP 建立用於變更資料擷取 (CDC) 的 AWS DMS 複寫任務 (因為已經透過 EXPDP 發生了滿載)。 如需詳細資訊，請參閱 AWS DMS 文件中的建立任務。	DBA

切換到 Amazon RDS for Oracle

任務	描述	所需技能
停止應用程式工作負載。	在計劃的切換視窗期間停止應用程式伺服器及其應用程式。	應用程式開發人員, DBA

任務	描述	所需技能
驗證現場部署 Oracle 預備資料庫與 EC2 執行個體的同步處理。	透過在現場部署來源資料庫上執行一些日誌切換，確認是否已將 SharePlex 複寫執行個體的複寫任務的所有訊息張貼到 Amazon EC2 上的 Oracle 預備資料庫。 如需詳細資訊，請參閱 Oracle 說明文件中的 6.4.2 切換記錄檔。	DBA
驗證 Oracle 預備資料庫與亞馬遜 RDS 適用於甲骨文資料庫的同步。	確認所有 AWS DMS 任務都沒有延遲且沒有錯誤，然後檢查任務的驗證狀態。	DBA
停止 SharePlex 和 Amazon RDS 的複寫。	如果 SharePlex 和 AWS DMS 複寫都沒有顯示任何錯誤，請停止這兩個複寫。	DBA
將應用程式重新對應至 Amazon RDS。	與應用程式伺服器及其應用程式共用 Amazon RDS for Oracle 端點詳細資訊，然後啟動應用程式以恢復業務操作。	應用程式開發人員, DBA

測試 AWS 目標環境

任務	描述	所需技能
在 AWS 上測試 Oracle 預備資料庫環境。	1. 測試 SharePlex 複製，並確認 Oracle 暫存資料庫上沒有同步處理間隙或複寫錯誤。 2. 透過內部部署環境中定義的基準測試，確認應用程式是否如預期般運作。	SharePlex、甲骨文管理

任務	描述	所需技能
測試 Amazon RDS 環境。	1. 確認複寫後傳播到 Amazon RDS 的所有資料都沒有錯誤。 2. 將另一個應用程式指向 Amazon RDS 資料庫執行個體，然後執行效能測試以驗證預期的行為。 如需詳細資訊，請參閱 Amazon RDS 文件中的亞馬遜 RDS 適用於甲骨文。	甲骨文管理

相關資源

- 自信地移轉
- Amazon EC2
- Amazon RDS for Oracle
- AWS Database Migration Service
- 對 AWS DMS 遷移進行偵錯：發生錯誤時該怎麼辦 (第 1 部分)
- 偵錯 AWS DMS 遷移：發生錯誤時該怎麼辦 (第 2 部分)
- 對 AWS DMS 遷移進行偵錯：發生錯誤時該怎麼辦？（第三部分）
- SharePlex 用於資料庫複製
- SharePlex：任何環境的資料庫複製

監控 Amazon Aurora 是否有沒有加密的

創建者：曼西蘇拉特瓦拉（AWS）

環境：生產	技術：安全性、身分識別、合規性、儲存與備份、資料庫	工作負載：開放原始碼；所有其他
AWS 服務：Amazon SNS；Amazon Aurora；AWS CloudTrail；Amazon CloudWatch；AWS Lambda		

Summary

此模式提供 Amazon Web Services (AWS) CloudFormation 範本，您可以部署這些範本，以便在建立 Amazon Aurora 執行個體時設定自動通知，而不會開啟加密。

Aurora 為全受管關聯式資料庫引擎，可與 MySQL 和 PostgreSQL 相容。透過一些工作負載，Aurora 可提供 MySQL 最多五倍的輸送量和 PostgreSQL 最多三倍的輸送量，而不需變更您的多數現有應用程式。

該 CloudFormation 模板創建一個 Amazon 事件 CloudWatch 事件和一個 AWS Lambda 函數。此事件使用 AWS CloudTrail 監控任何 Aurora 執行個體建立或現有執行個體的時間點還原。Cloudwatch 事件事件會啟動 Lambda 函數，該函數會檢查是否已啟用加密。如果未開啟加密，Lambda 函數會傳送 Amazon Simple Notification Service (Amazon SNS) 通知，通知您違規事件。

先決條件和限制

前提

- 有效的 AWS 帳戶

限制

- 此服務控制僅適用於 Amazon Aurora 執行個體。它不支援其他 Amazon Relational Database Service 服務 (Amazon RDS) 執行個體。

- CloudFormation 範本必須為`CreateDBInstance`和 `RestoreDBClusterToPointInTim e` 部署。

產品版本

- Amazon Aurora 支援的 PostgreSQL 版本
- Amazon Aurora 中支援的 MySQL 版本

架構

目標技術堆疊

- Amazon Aurora
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

目標架構

自動化和規模

您可以針對不同的區域和帳戶多次使用 CloudFormation 範本。您只需在每個區域或帳戶中執行一次。

工具

工具

- Amazon Aurora — Amazon Aurora 是全受管的關聯式資料庫引擎，與 MySQL 和 PostgreSQL 相容。
- AWS CloudTrail — AWS 可 CloudTrail 協助您管理 AWS 帳戶的管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供一系統事件 near-real-time 串流，用於描述 AWS 資源的變更。

- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — 亞馬遜簡單通知服務 (Amazon SNS) 是一種受管服務，可使用 Lambda、HTTP、電子郵件、行動推送通知和行動文字訊息 (SMS) 提供訊息交付。

Code

專案的 .zip 檔案可作為附件使用。

史诗

為 Lambda 指令碼建立 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	開啟 Amazon S3 主控台，然後選擇或建立 S3 儲存貯體。此 S3 儲存貯體將託管 Lambda 程式碼 .zip 檔案。您的 S3 儲存貯體必須與 Aurora 位於相同的區域。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 .zip 檔案上傳到您定義的 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
部署 CloudFormation 範本。	在 CloudFormation 主控台上，將以附件形式提供的`RDS_Aurora_Encryption_At_Rest.yml` CloudFormation 範本部署到此病毒碼。在下一個史詩中，提供模板參數的值。	雲端架構師

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選擇的 S3 儲存貯體的名稱。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<directory>/<file-name>.zip`	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件訊息傳送至提供的電子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 建立 S3 儲存貯體
- 將檔案上傳到 S3 儲存貯體
- 建立 Amazon Aurora 資料庫叢集
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon 監控甲骨文 GoldenGate 日誌 CloudWatch

創建者：奇特拉·克里希那穆錫（AWS）

環境：生產	技術：資料庫	工作量：甲骨文
AWS 服務：Amazon CloudWatch；Amazon SNS		

Summary

Oracle 為 Oracle 資料庫 GoldenGate 提供 Amazon Relational Database Service 服務 (Amazon RDS) 之間的即時複寫，或在亞馬遜彈性運算雲端 (Amazon EC2) 上託管的 Oracle 資料庫之間進行即時複寫。它支持單向和雙向複製。

當您用 GoldenGate 於複寫時，若要確認 GoldenGate 處理作業是否已啟動並執行，以確定來源和目標資料庫同步處於同步狀態，則監督至關重要。

此模式說明針對 GoldenGate 錯誤日誌實作 Amazon CloudWatch 監控的步驟，以及如何設定警示以傳送特定事件 (例如`STOP`或) 的通知，以`ABEND`便您可以採取適當的動作以快速恢復複寫。

先決條件和限制

前提

- GoldenGate 在 EC2 執行個體上安裝和設定，因此您可以在這些 EC2 執行個體上設定 CloudWatch 監控。如果您想要監控 GoldenGate AWS 區域以進行雙向複寫，則必須在執行該程序的每個 EC2 執行個體中安裝 CloudWatch 代理 GoldenGate 程式。

限制

- 此模式說明如何使用來監視 GoldenGate 程序 CloudWatch。 CloudWatch 不會監控複寫期間的複寫延遲或資料同步處理問題。您必須執行個別的 SQL 查詢來監視複寫延遲或資料相關錯誤，如GoldenGate 文件中所述。

產品版本

- 本文件是基于甲骨文 GoldenGate 19.1.0.0.4 在 Linux x86-64 上的實施情況。但是，此解決方案適用於所有主要版本的 GoldenGate.

架構

目標技術堆疊

- GoldenGate 安裝在 EC2 實例上的 Oracle 二進製文件
- Amazon CloudWatch
- Amazon Simple Notification Service (Amazon SNS)

目標架構

工具

AWS 服務

- Amazon CloudWatch 是一種監控服務，用於此模式來監視 GoldenGate 錯誤日誌。
- Amazon SNS 是一項訊息通知服務，用於此模式來傳送電子郵件通知。

其他工具

- 甲骨文 GoldenGate 是一種資料複寫工具，可用於 Amazon RDS for Oracle 資料庫或亞 Amazon EC2 上託管的甲骨文資料庫。

高階實作步驟

1. 為 CloudWatch 代理程式建立 AWS Identity and Access Management (IAM) 角色。
2. 將 IAM 角色附加到產生 GoldenGate 錯誤日誌的 EC2 執行個體。
3. 在 EC2 執行個體上安裝 CloudWatch 代理程式。
4. 設定 CloudWatch 代理程式組態檔：`awscli.conf`和`awslogs.conf`。
5. 啟動代 CloudWatch 理程式。
6. 在記錄群組中建立量度篩選器。
7. 設置 Amazon SNS。

8. 為度量篩選器建立警示。Amazon SNS 會在這些篩選器擷取事件時傳送電子郵件警示。

如需詳細指示，請參閱下一節。

史诗

步驟 1. 為 CloudWatch 代理程式建立 IAM 角色

任務	描述	所需技能
建立 IAM 角色。	存取 AWS 資源需要許可，因此您可以建立 IAM 角色，以包含每個伺服器執行 CloudWatch 代理程式所需的許可。 若要建立 IAM 角色： 1. 登入 AWS 管理主控台，然後前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。 2. 在導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。 3. 對於受信任的實體類型，請選擇 AWS 服務。 4. 對於常見使用案例，請選擇 EC2，然後選擇 [下一步]。 5. 在策略清單中，選取旁邊的核取方塊CloudWatchAgentServerPolicy。如有需要，請使用搜尋方塊來尋找政策。 6. 選擇下一步。 7. 對於 Role name (角色名稱)，輸入新角色的名稱，	AWS 一般資訊

任務	描述	所需技能
	例如 `goldengate-cw-monitoring-role` 或另一個您喜好的名稱。 8. (選用) 針對 Role description (角色描述)，輸入描述。 9. 確認CloudWatchAgentServerPolicy出現在策略名稱下方。 10. (選擇性) 新增一或多個標籤 (機碼值配對) 以組織、追蹤或控制此角色的存取權，然後選擇 [建立角色]。	

步驟 2. 將 IAM 角色附加到 GoldenGate EC2 執行個體

任務	描述	所需技能
將 IAM 角色附加到產生 GoldenGate 錯誤日誌的 EC2 執行個體。	GoldenGate 必須填入 CloudWatch 並監控所產生的錯誤日誌，因此您需要將在步驟 1 中建立的 IAM 角色附加到執行中的 EC2 執行 GoldenGate 個體。 若要將 IAM 角色附加至執行個體： 1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。 2. 在瀏覽窗格中，選擇 [執行個體]，然後尋找執行中的執行 GoldenGate 個體。	AWS 一般資訊

任務	描述	所需技能
	3. 選取執行個體，然後選擇 [動作]、[安全性]、[修改 IAM 角色]。 4. 選取在第一步中建立的 IAM 角色以附加至執行個體，然後選擇 [儲存]。	

步驟 3 至 5. 在金門 EC2 執行個體上安裝和設定 CloudWatch 代理程式

任務	描述	所需技能
在 GoldenGate EC2 執行個體上安裝 CloudWatch 代理程式。	若要安裝代理程式，請執行下列命令： `sudo yum install -y awslogs`	AWS 一般資訊
編輯代理程式組態檔。	1. 執行下列命令。 `sudo su -` 2. 視需要編輯此檔案以更新 AWS 區域。 `cat /etc/awslogs/conf [plugins] cwlogs = cwlogs [default] region = us-east-1` 3. 編輯`/etc/awslogs/awslogs.conf` 檔案以更新檔案名稱、記錄群組名稱和日期/時間格式。您必須指定日期/時間以符合中的日期格式`ggserror.log` ；	AWS 一般資訊

任務	描述	所需技能
	否則，記錄串流將不會流入 CloudWatch。例如： `datetime_format = %Y-%m-%dT%H:%M:%S%z` `file = /u03/oracle/oragg/ggserr.log` `log_group_name = goldengate_monitor`	
啟動代 CloudWatch 理程式。	若要啟動代理程式，請使用下列命令。 `$ sudo service awslogsd start` 啟動代理程式之後，您可以在 CloudWatch 主控台中檢視記錄群組。日誌流將具有文件的內容。	AWS 一般資訊

步驟 6. 為記錄群組建立量度篩選器

任務	描述	所需技能
為關鍵字異常結束和停止創建指標過濾器。	當您為日誌群組建立指標篩選器時，只要在錯誤日誌中識別篩選器，就會啟動警示並根據 Amazon SNS 組態傳送電子郵件通知。 若要建立量度篩選器： 1. 開啟主 CloudWatch 控台，網址為 https://c	CloudWatch

任務	描述	所需技能
	onsole.aws.amazon.com/cloudwatch/。 2. 選擇日誌群組的名稱。 3. 選擇 Actions (動作)，然後選擇 Create metric filter (建立指標篩選條件)。 4. 對於「濾鏡」樣式，請指定樣式，例如ABEND。 5. 選擇 Next (下一步)，然後輸入指標篩選條件的名稱。 6. 在「測量結果詳細資訊」下，針對測量結果命名 CloudWatch 空間，輸入要在其中發行測量結果的命名空間名稱 如果命名空間不存在，請務必選取 Create new (新建)。 7. 針對「度量」值，請輸入1，因為您的量度篩選器會計算篩選器中關鍵字的出現次數。 8. 將「單位」設為 「無」。 9. 選擇 Create metric filter (建立指標篩選條件)。可以從導覽窗格中找到您建立的指標篩選條件。 10.為STOPPED樣式建立另一個度量篩選器。在一個日誌組中，您可以創建多個指標過濾器並單獨設置警報。	

步驟 7. 設定 Amazon SNS

任務	描述	所需技能
建立 SNS 主題。	在此步驟中，您將 Amazon SNS 設定為為指標篩選器建立警示。 若要建立 SNS 主題： 1. 登入 Amazon SNS 主控台，網址為 https://console.aws.amazon.com/sns/home。 2. 在 [建立主題] 方塊中，輸入主題名稱，例如`goldengate-alert` ，然後選擇 [下一步]。 3. 針對 Type (類型)，選擇 Standard (標準)。 4. 捲動到表單結尾，然後選擇 Create topic (建立主題)。主控台會開啟新主題的 Details (詳細資料) 頁面。	Amazon SNS
建立訂閱。	若要建立主題的訂閱： 1. 在左導覽窗格中，選擇 Subscriptions (訂閱)。 2. 在 Subscriptions (訂閱) 頁面，選擇 Create subscription (建立訂閱)。 3. 在 [建立訂閱] 頁面上，選擇主題 ARN 欄位以查看 AWS 帳戶中的主題清單。 4. 選擇您在之前步驟所建立的主題。	Amazon SNS

任務	描述	所需技能
	5. 對於 Protocol (通訊協定)，選擇 Email (電子郵件)。 6. 針對 Endpoint (端點)，請輸入可用於接收通知的電子郵件地址。 7. 選擇 [建立訂閱]。 主控台會開啟新訂閱的 [詳細資料] 頁面。 8. 檢查您的電子郵件收件匣是否有來自 AWS Novents 的訊息，然後在電子郵件中選擇 [確認訂閱]。 Amazon SNS 會開啟您的 web 瀏覽器，並顯示含有您的訂閱 ID 的訂閱確認。	

步驟 8. 建立警示以傳送指標篩選器的通知

任務	描述	所需技能
建立 SNS 主題的警示。	若要根據記錄群組量度篩選器建立警示： 1. 開啟主 CloudWatch 控台，網址為 https://console.aws.amazon.com/cloudwatch/。 2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Log groups (日誌群組)。 3. 選擇包含指標篩選條件的日誌群組。	CloudWatch

任務	描述	所需技能
	4. 選擇 Metric filters (指標篩選條件)。 5. 在「度量篩選器」標籤中，選取您要依據的指標篩選器的核取方塊。 6. 選擇 Create alarm (建立警示)。 7. 對於「條件」，請在每個區段中指定下列項目： • 對於 Threshold type (閾值類型)，選擇 Static (靜態)。 • 對於無論何時如此。 <metric-name> ，選擇 [更大]。 • 對於比。 中，指定 0。 8. 選擇下一步。 9. 在通知之下： • 針對 Alarm state trigger (警示狀態觸發)，選擇 In Alarm (警示中)。 • 對於「傳送通知至下列 SNS 主題」，請選擇「選取現有主題」。 • 在電子郵件方塊中，選取您在上一個步驟中建立的 Amazon SNS 主題。 10選擇下一步。 11.在 Name and description (名稱和描述) 中，輸入警示的名稱和描述。	

任務	描述	所需技能
	注意：對於說明，您可以指定執行個體名稱，以便通知電子郵件具有描述性。 12針對 [預覽和建立]，檢查您的設定是否正確，然後選擇 [建立警示]。 完成這些步驟後，每當您正在監視的 GoldenGate 錯誤記錄檔 (`ggserr.log`) 中偵測到這些病毒碼時，您都會收到電子郵件通知。	

故障診斷

問題	解決方案
GoldenGate 錯誤日誌中的日誌流不會流入 CloudWatch。	檢查`/etc/awslogs/awslogs.conf` 檔案以確認檔案名稱、記錄群組名稱和日期/時間格式。您必須指定日期/時間以符合中的日期格式`ggserror.log` 。否則，記錄串流將不會流入 CloudWatch。

相關資源

- Amazon CloudWatch 文檔
- 使用 CloudWatch 代理程式收集指標和記錄檔
- Amazon SNS 文件

在適用於甲骨文的亞馬遜 RDS 上將 Oracle 數據庫企業版重新平台為標準版 2

由蘭雷秀恩米（AWS）和塔倫·查拉（AWS）創建

環境：生產	來源：內部部署	目標：Amazon RDS
R 類型：重新平台	工作量：甲骨文	技術：資料庫
AWS 服務：Amazon RDS		

Summary

Oracle 數據庫企業版（EE）是許多企業運行應用程序的熱門選擇。但是，在某些情況下，應用程式只使用很少或沒有 Oracle Database EE 功能，因此缺乏理由導致龐大的授權成本。當您遷移到 Amazon RDS 時，將這類資料庫降級為 Oracle 資料庫標準版 2 (SE2)，即可節省成本。

此模式說明如何在從現場部署遷移到適用於甲骨文的 Amazon RDS 時，如何從 Oracle 資料庫 EE 降級至 Oracle 資料庫 SE2。如果您的 EE Oracle 資料庫已經在 Amazon RDS 或亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上執行，則此模式中顯示的步驟也適用。

如需詳細資訊，請參閱 AWS Prescriptive Guidance 指南，了解如何在 AWS 上評估降級 Oracle 資料庫至標準版 2。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 甲骨文數據庫企業版
- 一種從屬端工具，例如 Oracle SQL 開發人員或 SQL*Plus，用於連接和執行 Oracle 資料庫上的 SQL 命令
- 用於執行評估的資料庫使用者；例如，下列其中一項：
  - 具有足夠權限執行 AWS Schema Conversion Tool (AWS SCT) 評估的使用者
  - 擁有足夠權限的使用者，可在 Oracle 資料庫說明表格上執行 SQL 查詢
- 用於執行資料庫移轉的資料庫使用者；例如，下列其中一項：

- 具有足夠權限執行 AWS Database Migration Service (AWS DMS) 的使用者
- 具有足夠權限的使用者可執行「Oracle 資料汲取」匯出與匯入
- 有足夠權限執行 Oracle 的使用者 GoldenGate

限制

- Amazon RDS for Oracle 有一個最大的數據庫大小。如需詳細資訊，請參閱 Amazon RDS 資料庫執行個體儲存體。

產品版本

本文件所述的一般邏輯適用於 9i 及更新版本的 Oracle 版本。如需支援的自我管理和適用於 Oracle 資料庫的 Amazon RDS 版本，請參閱 AWS DMS 文件。

若要在不支援 AWS SCT 的情況下識別功能使用情況，請在來源資料庫上執行 SQL 查詢。若要從不支援 AWS DMS 和 Oracle 資料泵浦的舊版 Oracle 遷移，請使用 Oracle 匯出和匯入公用程式。

如需目前支援的版本和版本清單，請參閱 AWS 文件中的 Oracle 在 Amazon RDS。如需定價和支援執行個體類別的詳細資訊，請參 Amazon RDS for Oracle 定價。

架構

源, 技術, 堆棧

- 在現場部署或 Amazon EC2 上運行的 Oracle 數據庫企業版

使用原生 Oracle 工具的目標技術堆疊

- Amazon RDS for Oracle 運行甲骨文數據庫 SE2

1. 使用「Oracle 資料汲取」來匯出資料。
2. 透過資料庫連結將傾印檔案複製到 Amazon RDS。
3. 使用甲骨文數據泵將轉儲文件導入 Amazon RDS。

使用 AWS DMS 的目標技術堆疊

- Amazon RDS for Oracle 運行甲骨文數據庫 SE2
- AWS DMS

1. 使用「Oracle 資料汲取」搭配「倒溯 _SCN」來匯出資料。
2. 透過資料庫連結將傾印檔案複製到 Amazon RDS。
3. 使用甲骨文數據泵將轉儲文件導入 Amazon RDS。
4. 使用 AWS DMS 變更資料擷取 (CDC)。

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。這種模式使用 Amazon RDS for Oracle。
- AWS SCT 提供以專案為基礎的使用者界面，可自動評估、轉換和複製來源 Oracle 資料庫的資料庫結構描述，並將其複製為與 Amazon RDS for Oracle 相容的格式。AWS SCT 可讓您分析將授權類型從企業版變更為 Oracle 標準版，藉此節省成本。AWS SCT 報告的授權評估和雲端 Support 部分提供有關使用中的 Oracle 功能的詳細資訊，讓您在遷移到 Amazon RDS 版 Oracle 時做出明智的決定。

其他工具

- 原生 Oracle 匯入和匯出公用程式支援將 Oracle 資料移入和移出 Oracle 資料庫。Oracle 提供兩種類型的資料庫匯入與匯出公用程式：「原始匯出與匯入」(適用於較早版本) 和 「Oracle 資料汲取匯出與匯入」(適用於 Oracle 資料庫 10g 版本 1 及更新版本)。
- Oracle GoldenGate 提供即時複寫功能，因此您可以在初始載入後同步目標資料庫。此選項有助於減少應用程式上線期間的停機時間。

史诗

進行移轉前評估

任務	描述	所需技能
驗證應用程式的資料庫需求。	確定您的應用程式已通過認證，可在 Oracle 資料庫 SE2 上執行。請直接洽詢軟體廠商、開發人員或應用程式文件。	應用程式開發者、DBA、應用程式擁有
直接在資料庫中調查 EE 功能的使用情況。	若要確定 EE 功能的使用，請執行下列其中一項作業： • 為您的 Oracle EE 資料庫產生 AWS SCT 評估報告。該報告告訴您如果要變更授權類型，應該從目前 EE 資料庫中移除哪些功能。 • 如果您有「Oracle 客戶 Support 務部」帳戶，請取得並執行 Sup port 文件 1317265.1 options_packs_usage_statistics.sql 中的指令碼，以產生 Oracle 資料庫正在使用的選項和功能的報告。 • 查詢 DBA_Feature_USAGE_ 統計，以顯示所有使用中功能的詳細資訊。	應用程式擁有者、DBA、應用程式開發
識別 EE 功能用於營運活動。	資料庫或應用程式管理員有時會依賴 Ee 專用功能來執行作業活動。常見的範例包括線上維護活動 (索引重建、表格移	應用程式開發者、DBA、應用程式擁有

任務	描述	所需技能
	動)，以及按批次工作分類的平行處理原則使用。 您可以在可能的情況下修改作業來減輕這些相依性。識別這些功能的使用方式，並根據成本與效益相比做出決定。 您可以使用「比較 Oracle 資料庫 EE 與 SE2 功能」表格作為指南，以識別「Oracle 資料庫 SE2」中可用的功能。	
複查 EE Oracle 資料庫的工作負載模式。	Oracle 資料庫 SE2 會隨時自動將使用量限制為最多 16 個 CPU 執行緒。 如果您的 Oracle EE 資料庫獲得使用「Oracle 診斷套件」的授權，請使用「自動工作負載儲存區域 (AWR)」工具或 DBA_HIST_* 視觀表來分析資料庫工作負載模式，以判斷 16 個 CPU 繫線的上限是否會在降級至 SE2 時對服務層次造成負面影響。 確保您的評估涵蓋尖峰活動的期間，例如一天結束、月份或年度處理。	應用程式擁有者、DBA、應用程式開發

在 AWS 上準備目標基礎設施

任務	描述	所需技能
部署和設定網路基礎結構。	建立虛擬私有雲 (VPC) 和子網路、安全群組和網路存取控制清單。	AWS 管理員、雲端架構師、網路管理員、 DevOps 工程師
為甲骨文 SE2 資料庫佈建亞馬遜 RDS。	佈建適用於 Oracle SE2 資料庫的目標 Amazon RDS，以符合應用程式的效能、可用性和安全性需求。建議生產工作負載使用異地同步備份組態。不過，若要改善移轉效能，您可以將啟用異地同步備份延遲到資料移轉之後。	雲端管理員、雲端架構師、DBA、 DevOps 工程師、AWS 管理員
自訂 Amazon RDS 環境。	設定自訂參數和選項，並啟用其他監控。如需詳細資訊，請參閱移轉至亞馬遜 RDS 適用於甲骨文的最佳實務。	AWS 管理員、AWS 系統管理員、雲端管理員、DBA、雲端架構師

執行移轉無效執行和應用程式測試

任務	描述	所需技能
遷移數據（空運行）。	使用最適合您特定環境的方法，將資料從來源 Oracle EE 資料庫遷移到亞馬遜 RDS for Oracle SE2 資料庫執行個體。根據大小、複雜度和可用停機時間等因素來選取移轉策略。請使用下列其中一項或組合： • Oracle 原生工具，例如「Oracle 資料汲取管理系統」(建議使用)、「Orac	DBA

任務	描述	所需技能
	le 匯入匯出」公用程式及 Oracle。 GoldenGate • AWS DMS，使用透過 CDC 連續複寫的完整負載。	
驗證目標資料庫。	執行資料庫儲存和程式碼物件的移轉後驗證。檢閱移轉記錄，並修正任何已識別的問題。如需詳細資訊，請參閱將 Oracle 資料庫移轉至 AWS 雲端的指南。	DBA
測試應用程式。	應用程式和資料庫管理員應適當地進行功能、效能和作業測試。如需詳細資訊，請參閱移轉至亞馬遜 RDS 適用於甲骨文的最佳實務。 最後，獲得利益相關者的測試結果的註冊。	應用程式開發人員、應用程式擁有者、DBA、移轉工程師、移轉主

切過

任務	描述	所需技能
從「Oracle 資料庫 EE」重新整理資料。	根據應用程式可用性需求選取資料重新整理方法。如需詳細資訊，請參閱將 Oracle 資料庫遷移到 AWS 的策略中的遷移方法。 例如，您可以透過使用 Oracle GoldenGate 或 AWS DMS 等工具與持續複寫，達到接近零的停機時間。如果停機時間視	應用程式擁有者、切換主管、DBA、移轉工程師、移轉主管

任務	描述	所需技能
	窗允許，您可以使用離線方法 (例如「Oracle 資料汲取」或「原始匯出匯入」公用程式) 來執行最終資料切換。	
將應用程式指向目標資料庫執行個體。	更新應用程式和其他用戶端中的連線參數，以指向適用於 Oracle SE2 的 Amazon RDS 資料庫。	應用程序開發人員，應用程序所有者，遷移工程師，遷移負責人，切換領導
執行移轉後活動。	執行資料後移轉工作，例如啟用異地同步備份、資料驗證和其他檢查。	DBA，移民工程師
執行切換後監控。	使用 Amazon CloudWatch 和 Amazon RDS Performance Insights 等工具來監控 Amazon RDS for Oracle 文 SE2 資料庫。	應用程式開發人員、應用程式擁有者、AWS 管理員、DBA、移轉工程

相關資源

AWS 方案指引

- 將 Oracle 資料庫遷移到 AWS 雲端 (指南)
- 在 AWS 上評估將 Oracle 資料庫降級為標準版 2 (指南)
- 將現場部署 Oracle 資料庫遷移到 Amazon RDS for Oracle (模式)
- 使用 Oracle Data Pump 將內部部署 Oracle 資料庫遷移至 Amazon RDS for Oracle (模式)

部落格文章

- 使用 AWS DMS 遷移 Oracle 資料庫，停機時間幾乎為零
- 使用亞馬遜 RDS 為甲骨文分析甲骨文 SE 中的績效管理
- 在甲骨文 SE 中使用 Amazon RDS for Oracle 管理您的 SQL 計劃

- 在 Oracle 標準版中實現表分區：第 1 部分

使用精確 Connect 將大型主機資料庫複寫到 AWS

創建者：盧西奧佩雷拉（AWS），巴拉吉莫罕（AWS）和薩丹吉里（AWS）

環境：生產	來源：內部部署大型主機	目標：AWS 資料庫
R 型：重新建築	工作負載：所有其他工作	技術：資料庫、雲端原生、大型主機、現代化
AWS 服務：Amazon DynamoDB；Amazon Keyspaces；Amazon MSK；Amazon RDS；Amazon ElastiCache		

Summary

這個模式概述了使用 External Connect 以近乎即時的方式，將大型主機資料庫中的資料複製到 Amazon 資料存放區的步驟。它使用適用於 Apache Kafka (Amazon MSK) 的 Amazon 受管串流和雲端中的自訂資料庫連接器來實作事件型架構，以改善可擴展性、彈性和效能。

Exact Connect 是一種複製工具，可擷取舊式大型主機系統中的資料，並將其整合到雲端環境中。透過變更資料擷取 (CDC)，透過使用具有低延遲和高輸送量異質資料管道的近即時訊息流程，將資料從大型主機複寫到 AWS。

此模式還涵蓋了具有多區域資料複寫和容錯移轉路由功能的復原資料管線的災難復原策略。

先決條件和限制

先決條件

- 您想要複寫到 AWS 雲端的現有大型主機資料庫 (例如 IBM DB2、IBM 資訊管理系統 (IMS) 或虛擬儲存存取方法 (VSAM)
- 有效的 AWS 帳戶
- 從您的公司環境到 AWS 的 AWS Direct Connect 或 AWS 虛擬私人網路 (AWS VPN)
- 具有可供您舊式平台存取之子網路的虛擬私有雲

架構

源, 技術, 堆棧

至少包含下列其中一個資料庫的大型主機環境：

- IMS 資料庫
- 数据库
- VSAM 文件

目標技術堆疊

- Amazon MSK
- Amazon Elastic Kubernetes Service (Amazon EKS) 和亞 Amazon EKS Anywhere
- Docker
- AWS 關聯式或 NoSQL 資料庫，如下所示：
  - Amazon DynamoDB
  - Amazon Relational Database Service 服務 (Amazon RDS)，適用於甲骨文、亞馬遜 RDS 或 Amazon Aurora
  - Amazon ElastiCache 的雷迪斯
  - Amazon Keyspaces (適用於 Apache Cassandra)

目標架構

將大型主機資料複寫到 AWS 資料庫

下圖說明將大型主機資料複寫到 AWS 資料庫，例如 DynamoDB、Amazon RDS、Amazon 或 Amazon ElastiCache Keyspaces。透過在現場部署大型主機環境中使用精確擷取和發佈者、現場部署分散式環境中 Amazon EKS Anywhere 上的精確調度器，以及在 AWS 雲端中精確套用引擎和資料庫連接器，以近乎即時的速度進行複寫。

該圖顯示以下工作流程：

1. 精確擷取會從 CDC 記錄取得大型主機資料，並將資料維護在內部暫態儲存裝置中。
2. 精確發行者偵聽內部數據存儲中的變化，並通過 TCP/IP 連接將 CDC 記錄發送到精確調度程序。

3. 精確調度員從發布商接收 CDC 記錄，並將其發送到 Amazon MSK。調度程序根據用戶配置和多個工作任務創建卡夫卡密鑰 parallel 推送數據。當記錄存放在 Amazon MSK 中時，調度程式會將通知傳回給發佈者。
4. Amazon MSK 在雲環境中保存 CDC 記錄。主題的分割區大小取決於您的交易處理系統 (TPS) 對輸送量的需求。該卡夫卡鍵是強制性的進一步轉換和交易排序。
5. 精確套用引擎會偵聽 Amazon MSK 的 CDC 記錄，並根據目標資料庫需求轉換資料 (例如，透過篩選或對應)。您可以將自訂邏輯新增至精確 SQD 指令碼。（SQD 是恰恰的專有語言。）精確套用引擎會將每個 CDC 記錄轉換為 Apache Avro 或 JSON 格式，並根據您的需求將其發佈至不同的主題。
6. 目標卡夫卡主題保持基於目標數據庫的多個主題 CDC 記錄，和卡夫卡促進基於定義的卡夫卡鍵事務排序。分割區索引鍵會與對應的分割區對齊，以支援順序處理。
7. 資料庫連接器 (自訂 Java 應用程式) 會接聽 Amazon MSK 的 CDC 記錄，並將它們儲存在目標資料庫中。
8. 您可以根據自己的需求選擇一個目標數據庫。此模式同時支援 NoSQL 和關聯式資料庫。

災難復原

業務持續性是組織成功的關鍵。AWS 雲端提供高可用性 (HA) 和災難復原 (DR) 的功能，並支援組織的容錯移轉和後援計劃。此模式遵循主動/被動 DR 策略，並針對實作符合 RTO 和 RPO 需求的 DR 策略提供高階指引。

下圖說明 DR 工作流程。

上圖顯示以下項目：

1. 如果 AWS 區域 1 發生任何故障，則需要半自動容錯移轉。如果區域 1 發生故障，系統必須啟動路由變更，才能將「精確調度程式」連接至「區域 2」。
2. Amazon MSK 透過區域之間的鏡像複寫資料。因此，在容錯移轉期間，區域 2 中的 Amazon MSK 叢集必須提升為主要領導者。
3. 精確套用引擎和資料庫連接器是可在任何地區運作的無狀態應用程式。
4. 資料庫同步處理取決於目標資料庫。例如，DynamoDB 可以使用全域資料表，而且 ElastiCache 可以使用全域資料存放區。

透過資料庫連接器進行低延遲和高輸送量處

資料庫連接器是此模式中的關鍵元件。連接器遵循以監聽器為基礎的方法，從 Amazon MSK 收集資料，並透過針對關鍵任務應用程式 (第 0 層和第 1 層) 的高輸送量和低延遲處理，將交易傳送到資料庫。下圖說明此程序。

此模式支援透過多執行緒處理引擎開發具有單執行緒耗用的自訂應用程式。

1. 連接器主執行緒會使用 Amazon MSK 的 CDC 記錄，並將它們傳送到執行緒集區進行處理。
2. 來自執行緒集區的執行緒處理 CDC 記錄，並將它們傳送至目標資料庫。
3. 如果所有執行緒都忙碌中，CDC 記錄會由執行緒佇列保持為保留狀態。
4. 主執行緒等待從執行緒佇列中清除所有記錄，並將偏移量提交至 Amazon MSK。
5. 子執行緒處理失敗。如果在處理期間發生失敗，失敗的訊息會傳送至 DLQ (無效字母佇列) 主題。
6. 子執行緒會根據大型主機時間戳記啟動條件式更新 (請參閱 DynamoDB 文件中的條件運算式)，以避免資料庫中的任何重複或 out-of-order 更新。

如需有關如何實作具有多執行緒功能的 Kafka 消費者應用程式的詳細資訊，請參閱 Confluent 網站上的 Apache Kafka 取用者的多執行緒訊息消耗部落格文章。

工具

AWS 服務

- 適用 Managed Streaming for Apache Kafka (Amazon MSK) 是一項全受管服務，可協助您建置和執行使用 Apache Kafka 處理串流資料的應用程式。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- Amazon EKS Anywhere 不在可協助您部署、使用和管理在您自己的資料中心執行的 Kubernetes 叢集。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon 可 ElastiCache協助您在 AWS 雲端中設定、管理和擴展分散式記憶體內快取環境。
- Amazon Keyspaces (適用於 Apache Cassandra) 是一種受管的資料庫服務，可協助您在 AWS 雲端中遷移、執行和擴展 Cassandra 工作負載。

其他工具

- 精確 Connect 將舊式大型主機系統 (例如 VSAM 資料集或 IBM 大型主機資料庫) 的資料整合到新一代雲端和資料平台中。

最佳實務

- 尋找 Kafka 分割區和多執行緒連接器的最佳組合，以平衡最佳效能與成本。由於 MIPS (每秒百萬個指令) 耗用量較高，因此多個精確擷取和調度程式執行個體可能會增加成本。
- 避免將資料操作和轉換邏輯新增至資料庫連接器。為此，請使用精確套用引擎，該引擎可提供以微秒為單位的處理時間。
- 在資料庫連接器中建立對資料庫 (活動訊號) 的定期要求或健康狀況檢查呼叫，以便頻繁地加熱連線並減少延遲。
- 實現線程池驗證邏輯以了解線程隊列中的待處理任務，並在下一次 Kafka 輪詢之前等待所有線程完成。這有助於避免節點、容器或處理序損毀時資料遺失。
- 透過健全狀況端點公開延遲指標，透過儀表板和追蹤機制增強可觀察性能。

史诗

準備來源環境 (內部部署)

任務	描述	所需技能
設定大型主機程序 (批次或線上公用程式)，以從大型主機資料庫啟動 CDC 程序。	1. 識別大型主機環境。 2. 識別將參與 CDC 程序的大型主機資料庫。 3. 在大型主機環境中，開發可啟動 CDC 工具以擷取來源資料庫中的變更的程序。如需指示，請參閱您的大型主機文件。 4. 記錄 CDC 過程，包括配置。	大型主機工程師

任務	描述	所需技能
	5. 在測試環境和生產環境中部署流程。	
啟動大型主機資料庫記錄串流。	1. 在大型主機環境中設定記錄串流以擷取 CDC 記錄檔。如需指示，請參閱您的大型主機文件。 2. 測試日誌串流，以確保它們擷取必要的資料。 3. 在測試和生產環境中部署日誌流。	大型主機資料庫專家
使用擷取元件擷取 CDC 記錄。	1. 在大型主機環境中安裝和設定精確擷取元件。如需指示，請參閱精確文件。 2. 測試組態，以確保 Capture 元件正常運作。 3. 設定複寫處理作業，透過擷取元件複寫擷取的 CDC 記錄。 4. 記錄每個來源資料庫的擷取組態。 5. 開發監視系統，以確保 Capture 元件會隨著時間的推移正確收集記錄檔。 6. 在測試和生產環境中部署安裝和配置。	大型主機工程師，精確 Connect 中小企業

任務	描述	所需技能
設定 [發行者] 元件以監聽擷取元件。	1. 在大型主機環境中安裝和設定精確發行者元件。如需指示，請參閱精確文件。 2. 測試組態，以確保「發行者」元件正常運作。 3. 設定複寫程序，將 CDC 記錄從發行者發佈至精確調度程式元件。 4. 記錄發行者組態。 5. 開發監視系統，以確保 Publisher 元件隨著時間的推移正常運作。 6. 在測試和生產環境中部署安裝和配置。	大型主機工程師，精確 Connect 中小企業
在現場部署分散式環境中的任何位置佈建 Amazon EKS。	1. 在現場部署基礎設施上隨處安裝 Amazon EKS，並確保已正確設定。如需指示，請參閱 Amazon EKS Anywhere 文件。 2. 為 Kubernetes 叢集設定安全的網路環境，包括防火牆。 3. 在 Amazon EKS Anywhere 不在叢集中實作和測試範例應用程式部署。 4. 實作叢集的自動擴展功能。 5. 開發和實施備份和災難恢復程序。	DevOps 工程師

任務	描述	所需技能
在分散式環境中部署和設定 Dispatcher 元件，以便在 AWS 雲端中發佈主題。	1. 設定和容器化精確調度程式元件。如需指示，請參閱精確文件。 2. 將調度程式泊塢視窗映像部署到現場部署 Amazon EKS Anywhere 不在環境中。 3. 在 AWS 雲端和調度程式之間設定安全連線。 4. 開發監控系統，以確保 Dispatcher 元件隨著時間的推移正常運作。 5. 在測試和生產環境中部署安裝和配置。	DevOps 工程師，精確 Connect 中小企

準備目標環境 (AWS)

任務	描述	所需技能
在指定的 AWS 區域佈建 Amazon EKS 叢集。	1. 登入您的 AWS 帳戶並進行設定，以確保擁有建立和管理 Amazon EKS 叢集所需的許可。 2. 在選取的 AWS 區域中建立虛擬私有雲端 (VPC) 和子網路。如需指示，請參閱 Amazon EKS 文件。 3. 建立和設定必要的網路安全群組，以允許 Amazon EKS 叢集與 VPC 中的其他資源之間進行通訊。如需詳細資訊，請參閱 Amazon EKS 文件。	DevOps 工程師，網絡管理員

任務	描述	所需技能
	4. 建立 Amazon EKS 叢集，並使用正確的節點群組大小和執行個體類型進行設定。 5. 透過部署範例應用程式來驗證 Amazon EKS 叢集。	
佈建 MSK 叢集並設定適用的 Kafka 主題。	1. 設定您的 AWS 帳戶，以確保具有建立和管理 MSK 叢集所需的許可。 2. 建立並設定必要的網路安全性群組，以允許 MSK 叢集與 VPC 中的其他資源之間進行通訊。如需詳細資訊，請參閱 Amazon VPC 文件。 3. 建立 MSK 叢集並將其設定為包含應用程式將使用的 Kafka 主題。如需詳細資訊，請參閱 Amazon MSK 文件。	DevOps 工程師，網絡管理員
設定套用引擎元件以監聽複寫的 Kafka 主題。	1. 設定和容器化精確套用引擎元件。 2. 將套用引擎泊塢視窗映像部署到 AWS 帳戶中的 Amazon EKS 叢集。 3. 設定套用引擎以聆聽 MSK 主題。 4. 在套用引擎中開發並設定 SQD 指令碼，以處理篩選和轉換。如需詳細資訊，請參閱精確文件。 5. 在測試和生產環境中部署套用引擎。	精確 Connect 中小企

任務	描述	所需技能
在 AWS 雲端佈建資料庫執行個體。	1. 設定您的 AWS 帳戶，以確保擁有建立和管理資料庫叢集和表格所需的許可。如需指示，請參閱您要使用的 AWS 資料庫服務的 AWS 文件。（請參閱資源部分以獲取鏈接。） 2. 在選取的 AWS 區域中建立 VPC 和子網路。 3. 建立並設定必要的網路安全群組，以允許資料庫執行個體與 VPC 中的其他資源之間進行通訊。 4. 建立資料庫並將其設定為包含應用程式將使用的資料表。 5. 設計和驗證資料庫結構描述。	數據工程師， DevOps 工程師
設定及部署資料庫連接器，以監聽套用引擎所發佈的主題。	1. 設計資料庫連接器，將 Kafka 主題與您在先前步驟中建立的 AWS 資料庫連接。 2. 根據目標資料庫開發連接器。 3. 設定連接器以接聽套用引擎所發佈的 Kafka 主題。 4. 將連接器部署到 Amazon EKS 叢集中。	應用開發人員、雲端架構師、資料工程

設定業務持續性和災難復原

任務	描述	所需技能
為您的業務應用程式定義災難復原目標。	1. 根據您的業務需求和影響分析，定義 CDC 管道的 RPO 和 RTO 目標。 2. 定義溝通和通知程序，以確保所有利益相關者都知道災難恢復計劃。 3. 確定實施災難恢復計劃所需的預算和資源。 4. 記錄災難復原目標，包括 RPO 和 RTO 目標。	雲端架構師、資料工程師、App 擁有者
根據定義的 RT/RPO 設計災難復原策略。	1. 根據您的重要性和復原需求，決定 CDC 管線最合適的災難復原策略。 2. 定義災難復原架構和拓撲。 3. 定義 CDC 管線的容錯移轉和容錯回復程序，以確保它們可以快速且無縫地切換到備份區域。 4. 記錄災難恢復策略和程序，並確保所有利益相關者對設計有清晰的了解。	雲端架構師、資料工程師
佈建災難復原叢集和組態。	1. 佈建用於災難復原的次要 AWS 區域。 2. 在次要 AWS 區域中，建立與主要 AWS 區域相同的環境。 3. 在主要和次要區域 MirrorMaker 之間配置阿帕奇卡夫卡。如需詳細資訊，	DevOps 工程師、網路管理員、雲端架構師

任務	描述	所需技能
	請參閱 Amazon MSK 文件。 4. 在次要區域中設定待命應用程式。 5. 設定主要和次要區域之間的資料庫複製。	
測試 CDC 管線以進行災難復原。	1. 為 CDC 管道定義災難復原測試的範圍和目標，包括要實現的測試案例和 RTO。 2. 識別用於執行災難復原測試的測試環境和基礎結構。 3. 準備測試資料集和指令碼以模擬失敗情境。 4. 驗證資料完整性和一致性，以確保不會遺失資料。	應用程序所有者，數據工程師，雲架構

相關資源

AWS 資源

- Amazon DynamoDB
- 使用亞馬遜動態 B 的條件運算式
- Amazon EKS
- Amazon EKS Anywhere
- Amazon ElasticCache
- Amazon Keyspaces
- Amazon MSK
- Amazon RDS 和 Amazon Aurora
- Amazon VPC

精確 Connect 資源

- 精確 Connect 概述
- 透過精確 Connect 變更資料擷取

融合資源

- 與阿帕奇卡夫卡消費者的多線程消費消費

使用 Lambda 和機 Secrets Manager 為亞馬遜 RDS 和 Aurora PostgreSQL 安排任務

創建者：亞斯拉賈（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：AWS 上的 PostgreSQL
R 類型：不適用	工作負載：開源	技術：資料庫
AWS 服務：AWS Lambda；Amazon RDS；AWS Secrets Manager；Amazon Aurora		

Summary

對於 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上託管的現場部署資料庫和資料庫，資料庫管理員通常使用 cron 公用程式來排程任務。

例如，可以使用 cr on 輕鬆地排程用於資料擷取的工作或資料清除工作。對於這些工作，資料庫證明資料通常是硬式編碼或儲存在特性檔中。但是，當您遷移到 Amazon Relational Database Service 服務 (Amazon RDS) 或亞馬遜 Aurora PostgreSQL 相容版本時，您無法登入主機執行個體來排程 Cron 任務。

此模式說明如何使用 AWS Lambda 和 AWS Secrets Manager 在遷移後為 Amazon RDS for PostgreSQL 和 Aurora PostgreSQL 相容資料庫排程任務。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon RDS for PostgreSQL 或 Aurora 相容資料庫

限制

- 工作必須在 15 分鐘內完成，也就是 Lambda 函數逾時限制。如需其他限制，請參閱 AWS Lambda 文件。

- Job 代碼必須使用 Lambda 支援的語言編寫。

架構

源, 技術, 堆棧

該堆棧具有以 Bash，Python 和 Java 等語言編寫的作業。資料庫認證會儲存在屬性檔案中，並使用 Linux cron 排程工作。

目標技術堆疊

此堆疊具有 Lambda 函數，該函數會使用儲存在 Secrets Manager 中的認證來連線到資料庫並執行活動。Lambda 函數是在排定的間隔使用 Amazon CloudWatch 事件啟動的。

目標架構

工具

- AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。您只需為使用的運算時間付費；程式碼未執行時不會收取任何費用。使用 AWS Lambda，您可以針對幾乎任何類型的應用程式或後端服務執行程式碼，而無需管理。AWS Lambda 在高可用性運算基礎設施上執行程式碼，並管理所有運算資源，包括伺服器和作業系統維護、容量佈建和自動擴展、程式碼監控和記錄。您只需要使用 AWS Lambda 支援的其中一種語言提供程式碼即可。
- Amazon E CloudWatch vents 提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。使用可快速設置的簡單規則，您可以匹配事件並將其路由到一個或多個目標函數或流。 CloudWatch 事件在發生時意識到操作變化。它會回應這些作業變更，並在必要時採取修正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。您也可以使用 CloudWatch 事件來排程使用 cron 或速率運算式在特定時間自行啟動的自動化動作。
- AWS Secrets Manager 可協助您保護存取應用程式、服務和 IT 資源的機密。您可以在其生命週期中輕鬆輪換、管理和擷取資料庫登入資料、API 金鑰和其他機密。使用者和應用程式透過呼叫 Secrets Manager API 擷取密碼，這樣就不需要以純文字硬式編碼敏感資訊。Secrets Manager 提供秘密輪換與 Amazon RDS 內置集成, Amazon Redshift, 和 Amazon DocumentDB. 該服務可擴展到其他類型的密鑰，包括 API 密鑰和 OAuth 令牌。Secrets Manager 可讓您使用精細的許可控制密碼的存取，並集中稽核 AWS 雲端、第三方服務和現場部署資源的機密輪替。

史诗

將資料庫認證儲存在 Secrets Manager

任務	描述	所需技能
建立 Lambda 函數的資料庫使用者。	在應用程式的不同部分使用不同的資料庫使用者是一個很好的作法。如果您的 cron 作業已經存在單獨的數據庫用戶，請使用它。否則，請建立新的資料庫使用者。如需詳細資訊，請參閱管理 PostgreSQL 使用者和角色 (AWS 部落格文章)。	DBA
在秘密管理員中將資料庫認證儲存為密碼。	請依照建立資料庫密碼 (Secrets Manager 說明文件) 中的指示進行。	DBA, DevOps

撰寫 Lambda 函數的程式碼

任務	描述	所需技能
選擇 AWS Lambda 支援的程式設計語言。	如需支援語言的清單，請參閱 Lambda 執行階段 (Lambda 文件)。	開發人員
撰寫邏輯以從 Secrets Manager 擷取資料庫認證。	如需範例程式碼，請參閱如何使用 AWS Secrets Manager 安全地為 Lambda 函數提供資料庫登入資料 (AWS 部落格文章)。	開發人員
撰寫邏輯以執行排程的資料庫活動。	將您在現場部署使用的排程任務的現有程式碼遷移到 AWS Lambda 函數。如需詳細資訊	開發人員

任務	描述	所需技能
	，請參閱部署 Lambda 函數 (Lambda 文件)。	

部署程式碼並建立 Lambda 函數

任務	描述	所需技能
建立 Lambda 函數部署套件。	此套件包含程式碼及其相依性。如需詳細資訊，請參閱部署套件 (Lambda 文件)。	開發人員
建立 Lambda 函數。	在 AWS Lambda 主控台中，選擇 [建立函數]、輸入函數名稱、選擇執行階段環境，然後選擇 [建立函數]。	DevOps
上傳部署套件。	選擇您建立的 Lambda 函數以開啟其組態。您可以直接在程式碼區段中撰寫程式碼，或上傳您的部署套件。若要上傳您的套件，請移至 [函式碼] 區段，選擇 [程式碼] 項目類型以上傳 .zip 檔案，然後選取套件。	DevOps
根據您的需求設定 Lambda 函數。	例如，您可以將逾時參數設定為預期 Lambda 函數所需的持續時間。如需詳細資訊，請參閱設定函數選項 (Lambda 文件)。	DevOps
設定 Lambda 函數角色的權限，以存取 Secrets Manager。	如需指示，請參閱 AWS Lambda 函數中的使用密碼 (秘密管理員文件)。	DevOps

任務	描述	所需技能
測試 Lambda 函數。	手動啟動該功能以確保其按預期工作。	DevOps

使用 CloudWatch 事件排程 Lambda 函數

任務	描述	所需技能
建立規則以依排程執行 Lambda 函數。	使用 CloudWatch 事件排程 Lambda 函數。如需指示，請參閱使用 CloudWatch 事件排程 Lambda 函數 (CloudWatch 事件教學課程)。	DevOps

相關資源

- AWS Secrets Manager
- 開始使用 Lambda
- 建立在 CloudWatch 事件上觸發的事件規則
- AWS Lambda 限制
- 從無伺服器應用程式查詢 AWS 資料庫 (部落格文章)

使用受信任的內容，在 AWS 上的 Db2 聯合資料庫中保護和簡化使用者存取

創建者：西帕薩拉迪（AWS）

環境：PoC 或試點	技術：資料庫；安全性、身分識別、合規性	工作負載：IBM
AWS 服務：Amazon EC2		

Summary

許多公司正在將其舊式大型主機工作負載遷移到亞馬遜網路服務 (AWS)。這項遷移包括在 Amazon 彈性運算雲端 (亞馬遜 EC2) 上將 IBM Db2 (適用於 z/OS) 資料庫轉移到適用於 Linux、Unix 和視窗 (LUW) 的 Db2。在從現場部署到 AWS 的階段遷移期間，使用者可能需要存取 IBM Db2 z/OS 和 Amazon EC2 上 Db2 LUW 中的資料，直到所有應用程式和資料庫都完全遷移到 Db2 LUW 為止。在此類遠端資料存取案例中，使用者驗證可能具有挑戰性，因為不同平台使用不同的驗證機制

此模式涵蓋如何在 Db2 上為 LUW 設定同盟伺服器，並將 Db2 for z/OS 設定為遠端資料庫。此模式會使用受信任的內容，將使用者的識別碼從 Db2 LUW 傳播至 Db2 z/OS，而不需在遠端資料庫上重新驗證。如需有關信任前後關聯的詳細資訊，請參閱其他資訊一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 Amazon EC2 執行個體上執行的 Db2 執行個體
- 在內部部署執行的遠端 Db2 for z/OS 資料庫
- 透過 AWS Site-to-Site VPN 或 AWS 直接連接至 AWS 的現場部署網路

架構

目標架構

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Site-to-Site VPN 可協助您在 AWS 上啟動的執行個體和自己的遠端網路之間傳遞流量。

其他服務

- db2cli 是 Db2 互動式指令列介面 (CLI) 命令。

史诗

在 AWS 上執行的 Db2 LUW 資料庫上啟用聯合

任務	描述	所需技能
在 DB2 LUW 資料庫上啟用聯合。	若要在 DB2 LUW 上啟用聯合，請執行下列命令。 `update dbm cfg using federated YES`	DBA
重新啟動資料庫。	若要重新啟動資料庫，請執行下列命令。 `db2stop force; db2start;`	DBA

編目遠端資料庫

任務	描述	所需技能
編目遠端 Db2 z/OS 子系統。	若要在 AWS 上執行的 Db2 LUW 上編目遠端 Db2 z/OS 資料庫，請使用下列範例命令。 `catalog TCPIP NODE tcpnode REMOTE mainframehost SERVER mainframeport`	DBA
編目遠端資料庫。	若要編目遠端資料庫，請使用下列範例命令。 `catalog db dbnam1 as ndbnam1 at node tcpnode`	DBA

建立遠端伺服器定義

任務	描述	所需技能
收集遠端 Db2 z/OS 資料庫的使用者認證。	繼續執行下列步驟之前，請先收集下列資訊： • Db2 z/OS 子系統名稱 — 上一個步驟中 LUW 上已編目的 Db2 z/OS 名稱 (例如，) `ndbnam1` • Db2 z/OS 版本 — Db2 z/OS 子系統版本 (例如) 12 • Db2 z/OS 使用者識別碼 — 具有 BIND 權限的使用者，只需要建立伺服器定義 (例如) `dbuser1`	DBA

任務	描述	所需技能
	• Db2 z/OS 密碼 — (例如，dbuser1) 的密碼 dbpasswd • Db2 z/OS 代理伺服器使用者 — 代理使用者的識別碼，用來建立信任連線 (例如) zproxy • Db2 z/OS 代理伺服器密碼 — zproxy 使用者的密碼 (例如，) zproxy	
建立 DRDA 包裝函式。	若要建立 DRDA 包裝函式，請執行下列命令。 `CREATE WRAPPER DRDA;`	DBA

任務	描述	所需技能
建立伺服器定義。	若要建立伺服器定義，請執行下列範例命令。 `CREATE SERVER ndbserver TYPE DB2/ZOS VERSION 12 WRAPPER DRDA AUTHORIZATION "dbuser1" PASSWORD "dbpasswd " OPTIONS ( DBNAME 'ndbnam1',FED_PROX Y_USER 'ZPROXY' );` 在此定義中，FED_PR0XY _USER 指定將用於建立與 Db2 z/OS 資料庫之信任連線的 Proxy 使用者。只有在 Db2 LUW 資料庫中建立遠端伺服器物件時，才需要授權使用者 ID 和密碼。它們將不會在以後運行時使用。	DBA

建立使用者對應

任務	描述	所需技能
建立代理主機使用者的使用者對應。	若要建立 Proxy 使用者的使用者對應，請執行下列命令。 `CREATE USER MAPPING FOR ZPROXY SERVER ndbserver OPTIONS (REMOTE_AUTHID 'ZPROXY', REMOTE_PA SSWORD 'zproxy');`	DBA
在 Db2 LUW 上為每個使用者建立使用者對應。	為 AWS 上 Db2 LUW 資料庫上需要透過代理使用者存取	DBA

任務	描述	所需技能
	遠端資料的所有使用者建立使用者對應。若要建立使用者對應，請執行下列命令。 `CREATE USER MAPPING FOR PERSON1 SERVER ndbserver OPTIONS (REMOTE_AUTHID 'USERZID', USE_TRUSTED_CONTEXT 'Y');` 此陳述式指定 Db2 LUW (PERSON1) 上的使用者可以建立與遠端 Db2 z/OS 資料庫的信任連線 ()。USE_TRUSTED_CONTEXT 'Y' 透過 Proxy 使用者建立連線之後，使用者可以使用 Db2 z/OS 使用者識別碼 () REMOTE_AUTHID 'USERZID' 來存取資料。	

創建受信任的上下文對象

任務	描述	所需技能
創建受信任的上下文對象。	若要在遠端 Db2 z/OS 資料庫上建立信任的內容物件，請使用下列範例命令。 `CREATE TRUSTED CONTEXT CTX_LUW_ZOS BASED UPON CONNECTION USING SYSTEM AUTHID ZPROXY ATTRIBUTES (`	DBA

任務	描述	所需技能
	ADDRESS '10.10.10.10') NO DEFAULT ROLE ENABLE WITH USE FOR PUBLIC WITHOUT AUTHENTIC ATION; 在此定義中，CTX_LUW_Z OS 是受信任內容物件的任意名稱。物件包含 Proxy 使用者識別碼和受信任連線必須來源之伺服器的 IP 位址。在這個例子中，服務器是 AWS 上的 Db2 LUW 數據庫。您可以使用網域名稱而非 IP 位址。該條款WITH USE FOR PUBLIC WITHOUT AUTHENTICATION 表示每個用戶 ID 都允許在受信任連接上切換用戶 ID。不需要提供密碼。	

相關資源

- IBM 資源存取控制設備
- LUW 联合会
- 信任的內容

其他資訊

Db2 信任的上下文

信任內容是 Db2 資料庫物件，可定義聯合伺服器與遠端資料庫伺服器之間的信任關係。若要定義信任關係，信任內容會指定信任屬性。信任屬性有三種類型：

- 提出初始資料庫連線要求的系統授權 ID
- 建立連線的 IP 位址或網域名稱
- 資料庫伺服器與資料庫用戶端之間資料通訊的加密設定

當連線要求的所有屬性都符合在伺服器上定義之任何信任內容物件中指定的屬性時，便會建立信任連線。受信任連線有兩種類型：隱含和明確連線。建立隱含信任連線之後，使用者會繼承在該信任連線定義範圍之外無法使用的角色。建立明確的信任連線之後，使用者可以開啟相同的實體連線 (無論是否驗證)。此外，Db2 使用者可以獲得指定權限的角色，這些角色只能在受信任的連線中使用。此病毒碼使用明確的信任連線。

此模式中的信任上下文

在模式完成之後，Db2 LUW 上的 PERSON1 會使用同盟信任內容存取來自 Db2 z/OS 的遠端資料。如果連線來自信任內容定義中指定的 IP 位址或網域名稱，則 PERSON1 的連線會透過 Proxy 使用者建立。建立連線之後，PERSON1 的對應 Db2 z/OS 使用者識別碼會在不重新驗證的情況下進行切換，而使用者可以根據為該使用者設定的 Db2 權限存取資料或物件。

同盟信任內容的優點

- 這種方法通過消除使用共同的用戶 ID 或應用程序 ID 來維護最小權限的原則，該 ID 或應用程序 ID 需要所有用戶所需的所有權限的超集合。
- 在同盟資料庫和遠端資料庫上執行交易的使用者的真實身分永遠是已知且可以稽核的。
- 效能提升，因為實體連線會在使用者間重複使用，而不需要聯合伺服器重新驗證。

使用現場部署 SMTP 伺服器和資料庫郵件傳送 Amazon RDS for SQL Server 服器資料庫執行個體的通知

創建者：尼沙德曼卡 (AWS)

環境：PoC 或試點	技術：資料庫、管理與治理	工作負載：Microsoft
AWS 服務：Amazon RDS		

Summary

資料庫郵件 (Microsoft 文件) 會使用簡易郵件傳送通訊協定 (SMTP) 伺服器，從 Microsoft SQL Server 資料庫傳送電子郵件訊息，例如通知或警示。適用於 Microsoft SQL 伺服器的 Amazon Relational Database Service 服務 (Amazon RDS) 文件提供使用 Amazon Simple Email Service (Amazon SES) 做為資料庫郵件 SMTP 伺服器的說明。如需詳細資訊，請參閱在 Amazon RDS for SQL Server 上使用 Database Mail。作為替代組態，此模式說明如何將資料庫郵件設定為使用現場部署 SMTP 伺服器做為郵件伺服器，將資料庫郵件設定為從 Amazon RDS for SQL Server 資料庫 (DB) 執行個體傳送電子郵件。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 執行標準版或企業版 SQL 伺服器的 Amazon RDS 資料庫執行個體
- 內部部署 SMTP 伺服器的 IP 位址或主機名稱
- 輸入安全群組規則，允許從 SMTP 伺服器的 IP 位址連線至 Amazon RDS for SQL Server 器資料庫執行個體
- 現場部署網路與包含 Amazon RDS 資料庫執行個體的虛擬私有雲端 (VPC) 之間的連線，例如 AWS Direct Connect 連線

限制

- 不支援快速版本的 SQL 伺服器。

- 如需有關限制的詳細資訊，請參閱 Amazon RDS 說明文件中的在 Amazon RDS for SQL Server 上使用資料庫郵件中的限制。

產品版本

- RDS 支援的 SQL 伺服器版本的標準版和企業版

架構

目標技術堆疊

- Amazon RDS for SQL Server 數據庫實例
- Amazon Route 53 轉發規則
- 資料庫郵件
- 內部部署 SMTP 伺服
- Microsoft SQL 伺服器管理工作室

目標架構

下圖顯示此模式的目標架構。當發生事件或動作並發出有關資料庫執行個體的通知或警示時，Amazon RDS for SQL Server 會使用資料庫郵件傳送電子郵件通知。資料庫郵件會使用內部部署 SMTP 伺服器來傳送電子郵件。

工具

AWS 服務

- 適用於 Microsoft SQL 伺服器的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 SQL Server 關聯式資料庫。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

其他工具

- 資料庫郵件是將電子郵件訊息 (例如通知和警示) 從 SQL Server 資料庫引擎傳送給使用者的工具。

- Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器，包括訪問，配置和管理 SQL 服務器組件的工具。在此模式中，您可以使用 SSMS 執行 SQL 命令，在適用於 SQL 伺服器資料庫執行個體的 Amazon RDS 上設定資料庫郵件。

史诗

啟用內部部署 SMTP 伺服器的網路連線

任務	描述	所需技能
從 RDS 資料庫執行個體移除異地同步備份。	如果您使用的是多區域 RDS 資料庫執行個體，請將異地同步備份執行個體轉換為單一可用區執行個體。完成資料庫郵件的設定後，您會將資料庫執行個體轉換回異地同步備份部署。然後，「資料庫郵件」組態可在主要和次要節點中使用。如需指示，請參閱從 Microsoft SQL Server 資料庫執行個體移除異地同步備份。	DBA
為現場部署 SMTP 伺服器上的 Amazon RDS 端點或 IP 位址建立允許清單。	SMTP 伺服器位於 AWS 網路之外。在現場部署 SMTP 伺服器上，建立允許清單，以允許伺服器與輸出端點或 Amazon RDS 執行個體或 Amazon RDS 上託管的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體進行通訊。此程序因組織而異。如需資料庫執行個體端點的詳細資訊，請參閱尋找資料庫執行個體端點和連接埠號碼。	DBA
移除連接埠 25 限制。	根據預設，AWS 會限制 EC2 執行個體上的連接埠 25。若要	一般 AWS

任務	描述	所需技能
	移除通訊埠 25 限制，請執行下列動作： 1. 使用您的 AWS 帳戶登入，然後開啟移除電子郵件傳送限制的要求表單。 2. 輸入您的電子郵件地址，以便 AWS Support 與您聯絡，提供有關請求的更新資訊。 3. 在「使用案例描述」欄位中輸入必要資訊。 4. 選擇提交。 請注意： • 如果您在多個 AWS 區域中有執行個體，請針對每個區域提交個別的請求。 • 處理您的請求最多可能需要 48 小時。	
新增 Route 53 規則以解析 SMTP 伺服器的 DNS 查詢。	使用 Route 53 解決 AWS 資源與現場部署 SMTP 伺服器之間的 DNS 查詢。您必須建立將 DNS 查詢轉寄至 SMTP 伺服器網域的規則，例如`example.com` 。如需指示，請參閱 Route 53 說明文件中的建立轉送規則。	網路管理員

在 Amazon RDS for SQL Server 資料庫執行個體上設定資料庫郵件

任務	描述	所需技能
啟用資料庫郵件。	為資料庫郵件建立參數群組，將`database mail xps`參數設定為`1`，然後將 Database Mail 參數群組與目標 RDS 資料庫執行個體建立關聯。如需指示，請參閱 Amazon RDS 文件中的啟用資料庫郵件。請勿繼續前往這些指示中的〈設定資料庫郵件〉一節。現場部署 SMTP 伺服器的組態與 Amazon SES 不同。	DBA
連線到資料庫執行個體。	從防禦主機，使用 Microsoft SQL 伺服器管理工作室 (SSMS) 連接到 Amazon RDS for SQL Server 的資料庫執行個體。如需指示，請參閱連線至執行 Microsoft SQL Server 資料庫引擎的資料庫執行個體。如果遇到任何錯誤，請參閱「相關資源」一節中的連線疑難排解參考資料。	DBA
建立設定檔。	在 SSMS 中，輸入下列 SQL 陳述式以建立資料庫郵件設定檔。取代以下的值： • 在中`profile_name` ，輸入新設定檔的名稱。 • 在中`description` ，輸入新設定檔的簡短描述。	DBA

任務	描述	所需技能
	如需有關此預存程序及其引數的詳細資訊，請參閱 Microsoft 文件中的 sysmail_add_profile_sp。 `EXECUTE msdb.dbo.sysmail_add_profile_sp @profile_name = 'SQL Alerts profile', @description = 'Profile used for sending outgoing notifications using OM SMTP Server.';`	

任務	描述	所需技能
將主參與者新增至設定檔。	輸入下列 SQL 陳述式，將公用或私用主體新增至資料庫郵件設定檔。主體是可以要求 SQL Server 資源的實體。取代以下的值： • 在中`profile_name` ，輸入您先前建立的設定檔的名稱。 • 在中`principal_name` ，輸入資料庫使用者或角色的名稱。這個值必須對應至 SQL Server 驗證使用者、視窗驗證使用者或視窗驗證群組。 如需有關此預存程序及其引數的詳細資訊，請參閱 Microsoft 說明文件中的 sysmail_add_add_Principalfile_ sp。 `EXECUTE msdb.dbo.sysmail_add_principalprofile_sp @profile_name = 'SQL Alerts profile', @principal_name = 'public', @is_default = 1 ;`	DBA

任務	描述	所需技能
創建帳戶。	輸入下列 SQL 陳述式，以建立「資料庫郵件」帳戶。取代以下的值： • 在中account_name ，輸入新帳戶的名稱。 • 在中description ，輸入新帳戶的簡短描述。 • 在中email_address ，輸入要從中傳送資料庫郵件訊息的電子郵件地址。 • 對於display_address ，輸入此帳戶外寄郵件所使用的顯示名稱，例如SQL Server Automated Notification 。您也可以使用輸入的值email_address 。 • 在中mailserver_name ，輸入 SMTP 郵件伺服器的名稱或 IP 位址。 • 對於port，保留的值25。 • 對於enable_ssl ，0如果您不想讓「資料庫郵件」使用 SSL 加密通訊，請將值保留在1或輸入。 • 在中username，輸入登入 SMTP 郵件伺服器的使用者名稱。如果伺服器不需要驗證，請輸入NULL。 • 在中password，輸入登入 SMTP 郵件伺服器的密碼。	DBA

任務	描述	所需技能
	如果伺服器不需要驗證，請輸入NULL。 如需有關此預存程序及其引數的詳細資訊，請參閱 Microsoft 文件中的 sysmail_add_account_ sp。 `EXECUTE msdb.dbo.sysmail_add_account_sp @account_name = 'SQL Alerts account', @description = 'Database Mail account for sending outgoing notifications.', @email_address = 'xyz@example.com', @display_name = 'xyz@example.com', @mailserver_name = 'test_smtp.example.com', @port = 25, @enable_ssl = 1, @username = 'SMTP-username', @password = 'SMTP-password';`	

任務	描述	所需技能
將帳戶新增至設定檔。	輸入下列 SQL 陳述式，將資料庫郵件帳戶新增至資料庫郵件設定檔。取代以下的值： • 在中`profile_name` ，輸入您先前建立的設定檔的名稱。 • 在中`account_name` ，輸入您先前建立的帳戶名稱。 如需有關此預存程序及其引數的詳細資訊，請參閱 Microsoft 說明文件中的 sysmail_add_profileaccount_ sp。 `EXECUTE msdb.dbo.sysmail_add_profileaccount_sp @profile_name = 'SQL Alerts profile', @account_name = 'SQL Alerts account', @sequence_number = 1;`	DBA
(選擇性) 將異地同步備份新增至 RDS 資料庫執行個體。	如果您想要使用資料庫鏡像 (DBM) 或永遠開啟可用性群組 (AGG) 來新增異地同步備份，請參閱將異地同步備份新增至 Microsoft SQL Server 資料庫執行個體中的指示。	DBA

相關資源

- 在適用 Amazon RDS for SQL Server 上使用資料庫郵件 (Amazon RDS 文件)

- 使用檔案附件 (Amazon RDS 文件)
- 疑難排解 SQL 伺服器資料庫執行個體的連線 (Amazon RDS 文件)
- 無法連線到 Amazon RDS 資料庫執行個體 (Amazon RDS 文件)

在 AWS 上的 IBM Db2 上為 SAP 設定災難復原

環境：生產	技術：資料庫；作業	工作負載：SAP
AWS 服務：Amazon EC2；AWS 彈性災難復原		

Summary

此模式概述了為 SAP 工作負載設定災難復原 (DR) 系統的步驟，並使用 IBM Db2 做為資料庫平台，並在 Amazon Web Services (AWS) 雲端上執行。我們的目標是提供低成本的解決方案，以便在發生中斷時提供業務連續性。

該圖案使用指示燈方法。透過在 AWS 上實作試驗燈 DR，您可以減少停機時間並維持業務連續性。試驗燈方法著重於在 AWS 中設定最小的 DR 環境，包括與生產環境同步的 SAP 系統和待命 Db2 資料庫。

該解決方案是可擴展的。您可以視需要將其延伸至完整的災難復原環境。

先決條件和限制

先決條件

- 在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上執行的 SAP 執行個體
- 一個數據庫
- SAP 產品可用性對照表 (PAM) 支援的作業系統
- 生產和待命資料庫主機的不同實體資料庫主機名稱
- 在每個 AWS 區域中啟用跨區域複寫 (CRR) 的 Amazon 簡易儲存服務 (Amazon S3) 儲存貯體

產品版本

- 資料庫 11.5.7 版或更新版本

架構

目標技術堆疊

- Amazon EC2
- Amazon Simple Storage Service (Amazon S3)
- Amazon Virtual Private Cloud（VPC 對等）
- Amazon Route 53
- IBM Db2 高可用性災難復原

目標架構

此架構以 Db2 做為資料庫平台，為 SAP 工作負載實作 DR 解決方案。生產資料庫部署在 AWS 區域 1 中，而待命資料庫則部署在第二個區域。待命資料庫稱為 DR 系統。Db2 資料庫支援多個待命資料庫 (最多三個)。它使用 Db2 HADR 來設定 DR 資料庫，並自動執行與待命資料庫之間的記錄傳送。

如果發生災難而使區域 1 無法使用，DR 區域中的待命資料庫會接管實際執行資料庫角色。SAP 應用程式伺服器可以預先建置，也可以使用 AWS 彈性災難復原或 Amazon 機器映像 (AMI) 來建置，以符合復原時間目標 (RTO) 要求。此模式使用 AMI。

Db2 HADR 實作生產待命設定，其中生產作為主要伺服器，而且所有使用者都連接至該伺服器。所有交易都會寫入記錄檔，這些記錄檔會使用 TCP/IP 傳輸到待命伺服器。待命伺服器會向前捲動傳輸的記錄檔記錄，以更新其本機資料庫，這有助於確保它與生產伺服器保持同步。

使用 VPC 對等互連，讓生產區域和 DR 區域中的執行個體可以彼此通訊。Amazon Route 53 最終用戶到互聯網應用程序。

1. 在區域 1 中建立應用程式伺服器的 AMI，並將 AMI 複製到區域 2。在發生災難時，使用 AMI 在區域 2 中啟動伺服器。
2. 設定實際執行資料庫 (位於區域 1) 與待命資料庫 (位於區域 2) 之間的 Db2 HADR 複寫。
3. 變更 EC2 執行個體類型，以便在發生災難時與生產執行個體相符。
4. 在「區域 1」中`LOGARCHMETH1`，設定為`db2remote: S3 path`。
5. 在「區域 2」中`LOGARCHMETH1`，設定為`db2remote: S3 path`。
6. 跨區域複寫會在 S3 儲存貯體之間執行。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。此病毒碼使用 VPC 對等互連。

最佳實務

- 在決定 HADR 複寫模式時，網路扮演著關鍵角色。對於跨 AWS 區域的 DR，我們建議您使用 Db2 HADR 非同步或超級異步模式。
- 如需 Db2 HADR 複寫模式的詳細資訊，請參閱 IBM 說明文件。
- 您可以使用 AWS 管理主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 為現有 SAP 系統建立新 AMI。然後，您可以使用 AMI 來復原現有的 SAP 系統或建立複製。
- AWS Systems Manager Automation 可協助處理 EC2 執行個體和其他 AWS 資源的常見維護和部署任務。
- AWS 提供多種原生服務來監控和管理 AWS 上的基礎設施和應用程式。Amazon CloudWatch 和 AWS 等服務 CloudTrail 可分別用於監控您的基礎設施和 API 操作。如需詳細資訊，請參閱 SAP on AWS — 搭配起搏器的 IBM Db2 HADR。

史诗

準備環境

任務	描述	所需技能
檢查系統和日誌。	1. 確認已設定 Db2 上的生產 SAP 系統。 2. 確認已開啟記錄備份，並設定為將日誌儲存在 S3 儲存貯體中。這可以通過 Db2 參	AWS 管理員、SAP 基礎管理員

任務	描述	所需技能
	數LOGARCHMETH1 進行檢查。 3. 建立其他應用程式伺服器的 AMI。	

設定伺服器和複製

任務	描述	所需技能
建立 SAP 和資料庫伺服器。	1. 若要部署 DR 區域的基礎設施，請使用 AWS CloudFormation 指令碼或使用生產執行個體的 AMI。作為指示燈方法的一部分，您可以在與生產執行個體相同的系列中使用較小的 EC2 執行個體。例如，如果您的生產實例類型是r6i.12xlarge ，則可以使用 DR 構建的r6i.xlarge 實例類型。不過，請務必在 DR 執行個體上配置相同的儲存容量，以還原生產資料庫備份。 2. 為其建立 Amazon Elastic File System (Amazon EFS) 掛載點/sapmnt/<SID>/ ，並確保將其設定為從主要系統複寫。 3. 從生產系統進行完整資料庫備份 (線上或離線)。您將使用此備份來建立 DR 資料庫。	SAP 基礎管理員

任務	描述	所需技能
	4. 在 DR 系統中，使用 SAP 軟體佈建管理員 (SWPM) 系統複製方法，搭配使用系統副本搭配備份/還原以進行 HA/DR 目的，以建置 DR SAP 系統。 5. 當 SWPM 詢問時，請使用從實際執行中取得的備份還原 DR 中的資料庫。DR 資料庫將處於向前復原擱置狀態。 還原完整備份之後，依預設會設定向前復原擱置狀態。向前復原擱置狀態表示資料庫正在還原，而且可能需要套用某些變更。如需詳細資訊，請參閱 IBM 說明文件。	

任務	描述	所需技能
檢查配置。	1. 若要設定 HADR 的記錄封存，生產環境和 DR 資料庫都必須能夠從所有記錄存檔位置自動擷取記錄檔。確認 DR 資料庫中的LOGARCHMETH1 參數設定為與實際執行資料庫中相同的位置。如果因為區域限制而無法存取相同的位置，請確定 DR 系統可以自動從主要系統擷取記錄檔。 2. 若要啟用 TCP/IP 連接埠以啟用資料庫複寫，請新增下列兩個項目，/etc/services 在生產環境和 DR 主機中進行修改。在程式碼中，<SID>指的是 Db2 資料庫的系統識別碼 (SID) (例如PR1)。 `<SID>_HADR_1 55001/tcp # DB2 HADR Port1` `<SID>_HADR_2 55002/tcp # DB2 HADR Port2` 確認這兩個連接埠都允許主要和待命之間的輸入和輸出流量。 3. 簽/etc/hosts 入生產環境和 DR 主機，以確認生產和待命主機的主機名稱指向正確的 IP 位址。	AWS 管理員、SAP 基礎管理員

任務	描述	所需技能
設定從生產資料庫到 DR 資料庫的複寫 (使用非同步模式)。	1. 在生產資料庫中，執行下列命令以更新參數。 `db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST HOST1` `db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_1` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST HOST2` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_2` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid>` `db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120` `db2 UPDATE DB CFG FOR <SID> USING HADR_SYNCMODE ASYNC` `db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT 1000` `db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240` `db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexbuild ON` 2. 在 DR 資料庫中，執行下列命令來更新參數。	SAP 基礎管理員

任務	描述	所需技能
	`db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST HOST2` `db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_2` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST HOST1` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_1` `db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid>` `db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120` `db2 UPDATE DB CFG FOR <SID> USING HADR_SYNCMODE ASYNC` `db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT 1000` `db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240` `db2 UPDATE DB CFG FOR <SID> USING indexrec RESTART logindexbuild ON` 需要這些參數才能將 HADR 相關資訊提供給兩個資料庫。在 Db2 資料庫中，HADR 會根據每個先前設定的參數的值來啟動。如需這些參數的詳細資訊，請參閱 IBM 說明文件。	

任務	描述	所需技能
	3. 使用下列命令，先在新建立的待命資料庫上啟動 HADR。 `db2 deactivate db <SID>` `db2 start hadr on db <SID> as standby` 4. 使用下列命令在生產資料庫上啟動 HADR。 `db2 deactivate db <SID>` `db2 start hadr on db <SID> as primary` 5. 檢查生產和待命 Db2 資料庫是否處於同步狀態，並且記錄傳送正在進行中。 若要監視 HADR 複寫狀態，請使用下列db2pd命令。 `db2pd -d <SID> -hadr` 如需監控 HADR 的詳細資訊，請參閱 IBM 說明文件。	

測試 DR 容錯移轉工作

任務	描述	所需技能
規劃 DR 測試的生產業務停機時間。	請務必在生產環境上規劃所需的業務停機時間，以測試 DR 容錯移轉案例。	SAP 基礎管理員

任務	描述	所需技能
建立測試使用者。	建立可在 DR 主機中驗證的測試使用者 (或任何測試變更)，以確認 DR 容錯移轉後的記錄複寫。	SAP 基礎管理員
在主控台上，停止生產 EC2 執行個體。	在此步驟中會啟動非正常關機，以模擬災難案例。	AWS 系統管理員
擴展 DR EC2 執行個體以符合需求。	在 EC2 主控台上，變更 DR 區域中的執行個體類型。 1. 停止執行個體：如果執行個體正在執行，您必須先停止執行個體，才能變更其執行個體類型。在 EC2 主控台上，選取執行個體，然後選擇停止。 2. 修改執行個體類型：在 EC2 主控台上選取執行個體，然後選擇 [動作]、[執行個體設定]、[變更執行個體類型]。選取與主要執行個體相符的執行個體類型，然後選擇「套用」。 3. 啟動執行個體：執行個體類型變更完成後，透過選取執行個體並選擇 Start，從 EC2 主控台啟動執行個體。 4. 要啟動 Db2 數據庫，請使用以下命令。 `db2start` `db2 start HADR on db <SID> as standby`	SAP 基礎管理員

任務	描述	所需技能
啟動接管。	從 DR 系統 (host2) 起始接管程序，並將 DR 資料庫顯示為主要資料庫。 `db2 takeover hadr on database <SID> by force` 您可以選擇性地設定下列參數，根據執行處理類型自動調整資料庫記憶體配置。該 INSTANCE_MEMORY 值可以根據要分配給 Db2 數據庫的內存的專用部分來決定。 `db2 update db cfg for <SID> using INSTANCE_MEMORY <FIXED VALUE> IMMEDIATE;` `db2 get db cfg for <SID> | grep -i DATABASE_MEMORY AUTOMATIC IMMEDIATE;` `db2 update db cfg for <SID> using self_tuning_mem ON IMMEDIATE;` 使用下列指令驗證變更。 `db2 get db cfg for <SID> | grep -i MEMORY` `db2 get db cfg for <SID> | grep -i self_tuning_mem`	SAP 基礎管理員
在 DR 區域中啟動 SAP 的應用程式伺服器。	使用您在生產系統中建立的 AMI，在 DR 區域中啟動新的其他應用程式伺服器。	SAP 基礎管理員

任務	描述	所需技能
在啟動 SAP 應用程式之前執行驗證。	1. 驗證/etc/hosts 和/etc/fstab 項目。 2. 掛載/sapmnt/<SID>/ 在 DR 系統上。 3. 驗證 DR 檔案系統/sapmnt/<SID>/ 是否與生產同步/sapmnt/<SID>/ 。 4. 登錄到<sid>adm用戶R3trans -d，運行並驗證文trans.log 件中的輸出。trans.log 檔案會在您執行R3trans -d指令的相同位置產生。	AWS 管理員、SAP 基礎管理員
在 DR 系統上啟動 SAP 應用程式。	使用使用者啟動 DR 系統上的 SAP 應<sid>adm用程式。請使用下列程式碼，其中代XX表 SAP ABAP SAP 中央服務 (ASCS) 伺服器的執行個體編號，並代YY表 SAP 應用程式伺服器的執行個體編號。 `sapconrol -nr XX -function StartService <SID>` `sapconrol -nr XX -function StartSystem` `sapconrol -nr YY -function StartService <SID>` `sapconrol -nr YY -function StartSystem`	SAP 基礎管理員

任務	描述	所需技能
執行 SAP 驗證。	這會以 DR 測試的形式執行，以提供證據或檢查 DR 區域的資料複寫是否成功。	測試工程師

執行 DR 容錯回復工作

任務	描述	所需技能
啟動生產 SAP 和資料庫伺服器。	在主控台上，啟動在生產系統中託管 SAP 和資料庫的 EC2 執行個體。	SAP 基礎管理員
啟動生產資料庫並設定 HADR。	登錄到生產系統（host1），並使用以下命令驗證數據庫處於恢復模式。 `db2start` `db2 start HADR on db P3V as standby` `db2 connect to <SID>` 確認 HADR 的狀態為connected 。複寫狀態應為peer。 `db2pd -d <SID> -hadr` 如果資料庫不一致connected 且peer狀態不一致，則可能需要進行備份和還原，才能使資料庫 (開啟host1) 與目前作用中的資料庫 (host2在 DR 區域中) 同步。在此情況下，請將資料庫備份從 host2 DR 區域	SAP 基礎管理員

任務	描述	所需技能
	中的資料庫還原至host1生產區域中的資料庫。	
將資料庫回復到生產區域。	在一般 business-as-usual 情況下，此步驟會在排定的停機時間內執行。DR 系統上執行的應用程式會停止，且資料庫無法回到生產區域 (區域 1)，以便從生產區域恢復作業。 1. 登入 DR 區域中的 SAP 應用程式伺服器，並停止 SAP 應用程式。 2. /sapmnt/<SID> 從 DR 系統卸載，確定變更已反向複製到生產/sapmnt/<SID> 系統。 3. 登入生產區域中的資料庫伺服器 (host1)，然後執行接管。 `db2 takeover hadr on database <SID>` 4. 檢查 HADR 狀態：HADR_ROLE 應該處於開啟狀態host1並PRIMARY開Stand `db2pd -d <SID> -hadr`	SAP 基礎管理員

任務	描述	所需技能
在啟動 SAP 應用程式之前執行驗證。	1. 驗證/etc/hosts 和/etc/fstab 項目。 2. 安裝/sapmnt/<SID>/ 在生產系統上。 3. 確保它與 DR 系統同步/sapmnt/<SID>/ 。 4. 登錄到<sid>adm用戶R3trans -d，運行並驗證文trans.log 件中的輸出。trans.log 檔案會在您執行R3trans -d指令的相同位置產生。	AWS 管理員、SAP 基礎管理員
啟動 SAP 應用程式。	1. 使用使用者在生產系統上啟動 SAP 應<sid>adm用程式。請使用下列程式碼，其中代XX表 SAP ASCS 伺服器的執行個體編號，並代YY表 SAP 應用程式伺服器的執行個體編號。 `sapconrol -nr XX -function StartService <SID>` `sapconrol -nr XX -function StartSystem` `sapconrol -nr YY -function StartService <SID>` `sapconrol -nr YY -function StartSystem` 2. 若要確認應用程式伺服器可供使用，請登入 SAP 並使用 SICK 和 SM51 交易執行檢查。	SAP 基礎管理員

故障診斷

問題	解決方案
用於疑難排解 HADR 相關問題的重要記錄檔和命令	• `db2 get db cfg | grep -i hadr` • `db2pd -d sid -hadr` • `Db2diag.log` （此文件通常位於目db2dump錄內，db2dump路徑由參數定義DIAGPATH。）
針對 Db2 UDB 上的 Hadr 問題進行疑難排解的 SAP 注意事項	請參閱 SAP 附註 1154013-DB6：哈德爾環境中的資料庫問題。(您需要 SAP 入口網站認證才能存取此備註。)

相關資源

- AWS 上 Db2 資料庫的災難復原方法 (部落格文章)
- SAP on AWS — 搭配起搏器的 IBM Db2 哈德爾
- 逐步設定 DB2 資料庫之間的 HADR 複寫的程序
- 哈德爾百科

其他資訊

使用此模式，您可以為 Db2 資料庫上執行的 SAP 系統設定嚴重損壞修復系統。在災難情況下，企業應該能夠在您定義的復原時間目標 (RTO) 和復原點目標 (RPO) 需求內繼續進行：

- RTO 是服務中斷和恢復服務之間的最大可接受延遲。這決定了當服務無法使用時，什麼被視為可接受的時間範圍。
- RPO 是自上次資料復原點以來可接受的時間上限。這決定了最後一個恢復點和服務中斷之間可接受的數據丟失。

如需 HADR 相關的常見問題集，請參閱 SAP 注意事項 #1612105-DB6：Db2 高可用性災難復原 (HADR) 的常見問題集。(您需要 SAP 入口網站認證才能存取此備註。)

在 Amazon RDS 上為甲骨文電子商務套件設置 HA/DR 架構，並使用活動備用數據庫

由西蒙·坎寧安（AWS）和尼廷薩克森納創建

環境：生產	技術：資料庫；基礎架構	工作量：甲骨文
AWS 服務：Amazon RDS		

Summary

此模式說明如何在 Amazon 關聯式資料庫服務 (Amazon RDS) 自訂高可用性 (HA) 和災難復原 (DR) 上架構 Oracle 電子商務解決方案，方法是在另一個 Amazon Web 服務 (AWS) 可用區域中設定 Amazon RDS 自訂僅供讀取複本資料庫，然後將其轉換為作用中的待命資料庫。Amazon RDS 自訂僅供讀取複本的建立是透過 AWS 管理主控台完全自動化的。

此模式不會討論新增其他應用程式層和共用檔案系統的步驟，這些檔案也可以是 HA/DR 架構的一部分。如需這些主題的相關資訊，請參閱下列「Oracle 客戶 Support 務部注意事項」：1375769.1、1375670.1 和 1383621.1 (節 5, 進階複製選項)。(存取需要「Oracle 客戶 Support 部」帳戶。)

若要將電子商務套件系統遷移到 Amazon Web Services (AWS) 上的單一可用區架構，請參閱將 Oracle 電子商務套件遷移至 Amazon RDS 自訂模式。

Oracle 電子商務套件是一種企業資源計劃 (ERP) 解決方案，用於自動化整個企業的處理，例如財務、人力資源、供應鏈和製造。它具有三層架構：客戶端，應用程序和數據庫。以前，您必須在自我管理的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上執行電子商務套件資料庫，但現在您可以從 Amazon RDS 自訂中受益。

先決條件和限制

先決條件

- Amazon RDS 自定義上的現有電子商務套件安裝; 看到模式遷移甲骨文電子商務套件到 Amazon RDS 自定義
- 如果您要將僅供讀取複本變更為唯讀，並使用此複本將報告卸載至待命狀態，請使用「Oracle Active Data Guard」資料庫授權 (請參閱 Oracle 技術管理系統的商業價目表)

限制

- Amazon RDS 自訂上 Oracle 資料庫的限制和不受支援的組態
- 與適用於甲骨文僅供讀取複本的 Amazon RDS 自訂相關限制

產品版本

如需 Amazon RDS 自訂支援的 Oracle 資料庫版本和執行個體類別，請參閱適用於 Oracle 的 Amazon RDS 自訂需求和限制。

架構

下圖說明 AWS 上 E-Business Suite 的代表性架構，在主動/被動設定中包含多個可用區域和應用程式層。該資料庫使用 Amazon RDS 自訂資料庫執行個體和 Amazon RDS 自訂僅供讀取複本。僅供讀取複本使用作用中資料保全來複寫到另一個可用區域。您也可以使用僅供讀取複本卸載主要資料庫上的讀取流量，以及用於報告用途。

如需詳細資訊，請參閱 Amazon RDS 文件中的使用適用於 Oracle 的僅供讀取複本。

Amazon RDS 自訂僅供讀取複本預設會建立為已掛接。不過，如果您想要將某些唯讀工作負載卸載至待命資料庫以減少主要資料庫的負載，您可以依照 E pics 段落中的步驟，手動將已掛接複本的模式變更為唯讀。一般使用案例是從待命資料庫執行報表。變更為唯讀需要作用中的待命資料庫授權。

在 AWS 上建立僅供讀取複本時，系統會在封面下使用 Oracle 資料保全代理程式。 此組態會在「最大效能」模式中自動產生並設定，如下所示：

```
DGMGRL> show configuration
Configuration - rds_dg
  Protection Mode: MaxPerformance
  Members:
  vis_a - Primary database
    vis_b - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS   (status updated 58 seconds ago)
```

工具

AWS 服務

- Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。它可以自動執行資料庫管理工作和作業，同時讓身為資料庫管理員的您可以存取和自訂資料庫環境和作業系統。

其他工具

- 「Oracle 資料保全」是可協助您建立及管理 Oracle 待命資料庫的工具。此病毒碼使用 Oracle 資料保全在 Amazon RDS 自訂上設定作用中待命資料庫。

史诗

建立僅供讀取複本

任務	描述	所需技能
建立 Amazon RDS 自訂資料庫執行個體的僅供讀取複本。	若要建立僅供讀取複本，請遵循 Amazon RDS 文件中的指示，並使用您建立的 Amazon RDS 自訂資料庫執行個體 (請參閱先決條件一節) 做為來源資料庫。 依預設，Amazon RDS 自訂僅供讀取複本建立為實體待命，並處於掛接狀態。這是故意確保符合 Oracle 主動資料保全授權的規定。請依照下列步驟將僅供讀取複本轉換為唯讀模式。	DBA

將僅供讀取複本變更為唯讀作用中待命

任務	描述	所需技能
Connect 到 Amazon RDS 自訂僅供讀取複本。	使用下列命令，將實體待命資料庫轉換為作用中待命資料庫。 重要：這些指令需要 Oracle 作用中的待命授權。若要取得授權，請聯絡您的 Oracle 代表。 `$ sudo su - rdsdb` `-bash-4.2$ sql` `SQL> select process,s` `tatus,sequence# from v` `$managed_standby;` `PROCESS STATUS` `SEQUENCE#` `--------- ------------` `----------` `ARCH CLOSING` `3956` `ARCH CONNECTED` `0` `ARCH CLOSING` `3955` `ARCH CLOSING` `3957` `RFS IDLE` `0` `RFS IDLE` `3958` `MRP0 APPLYING_LOG` `3958` `SQL> select name,` `database_role,` `open_mode from v` `$database;`	DBA

任務	描述	所需技能
	NAME DATABASE_ ROLE OPEN_MODE --------- --------- ------- --------- ----------- VIS PHYSICAL STANDBY MOUNTED SQL> alter database recover managed standby database cancel; Database altered. Open the standby database SQL> alter database open; Database altered. SQL> select name, database_role, open_mode from v $database; NAME DATABASE_ ROLE OPEN_MODE --------- --------- ------- --------- ----------- VIS PHYSICAL STANDBY READ ONLY	

任務	描述	所需技能
使用即時記錄套用開始媒體復原。	若要啟用即時記錄套用功能，請使用下列命令。這些會將待命 (僅供讀取複本) 轉換並驗證為作用中待命資料庫，因此您可以連線並執行唯讀查詢。 `SQL>  alter database recover managed standby database using current logfile disconnect from session;` `Database altered`	DBA
檢查資料庫狀態。	要檢查數據庫的狀態，請使用以下命令。 `SQL> select name, database_role, open_mode from v$database;` `NAME  DATABASE_ROLE  OPEN_MODE` `--------- ---------- ----------` `VIS  PHYSICAL STANDBY  READ ONLY WITH APPLY`	DBA

任務	描述	所需技能
核取重做套用模式。	要檢查重做應用模式，請使用以下命令。 `SQL> select process,status,sequence# from v$managed_standby;` `PROCESS STATUS SEQUENCE#` `--------- ------------ ----------` `ARCH CLOSING 3956` `ARCH CONNECTED 0` `ARCH CLOSING 3955` `ARCH CLOSING 3957` `RFS IDLE 0` `RFS IDLE 3958` `MRP0 APPLYING_LOG 3958` `SQL> select open_mode from v$database;` `OPEN_MODE` `--------------------` `READ ONLY WITH APPLY`	DBA

相關資源

- 將甲骨文電子商務套件遷移到 Amazon RDS 自定義（AWS Prescriptive Guidance）
- 使用 Amazon RDS 自定義（Amazon RDS 文檔）
- 使用 Amazon RDS 自訂專用僅供讀取複本 (Amazon RDS 文件)
- 適用於甲骨文的 Amazon RDS 自訂 — 資料庫環境中的新控制功能 (AWS 新聞部落格)

- 遷移 AWS 上的甲骨文電子商務套件 (AWS 白皮書)
- AWS 上的甲骨文電子商務套件架構 (AWS 白皮書)

使用 GTID 在亞馬遜 EC2 上設置 Amazon RDS for MySQL 和 MySQL 之間的數據複寫

創建者：拉傑什·馬迪瓦利 (AWS)

環境：PoC 或試點	技術：資料庫	工作負載：開源

Summary

此模式說明如何透過使 Amazon Web Services MySQL 原生全域交易識別碼 (GTID) 複寫，在適用於 MySQL 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS) 與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 MySQL 資料庫之間設定資料複寫。

使用 GTID 時，在原始伺服器上認可並由複本套用交易時，就會識別和追蹤交易。在容錯移轉期間啟動新複本時，不需要參考記錄檔。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 已部署 Amazon Linux 執行個體

限制

- 此設定需要內部小組執行唯讀查詢。
- 來源和目標 MySQL 版本必須相同。
- 在相同的 AWS 區域和虛擬私有雲 (VPC) 中設定複寫。

產品版本

- Amazon RDS 版本 5.7.23 及更新版本，這些是支援 GTID 的版本

架構

源, 技術, 堆棧

- Amazon RDS for MySQL

目標技術堆疊

- Amazon EC2

目標架構

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- 適用於 MySQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 MySQL 關聯式資料庫。

其他服務

- 「全域交易識別符 (GTID)」https://dev.mysql.com/doc/refman/5.7/en/replication-gtids.html是系統為遞交的 MySQL 交易所產生的唯一識別符。
- mysqldump 是用來執行邏輯備份的用戶端公用程式，產生 SQL 陳述式可執行來重現來源資料庫物件定義和資料表資料。
- MySQL 是 MySQL 的命令行客戶端。

史诗

建立和準備 Amazon RDS for MySQL 的資料庫執行個體

任務	描述	所需技能
建立適用於 MySQL 的 RDS 執行個體。	若要建立適用於 MySQL 的 RDS 執行個體，請使用下一個任務中涵蓋的參數值，遵循 Amazon RDS 文件中的步驟。	DBA，工程師 DevOps
啟用 DB 參數群組中的 GTID 相關設定。	在 Amazon RDS for MySQL 用於 MySQL 資料庫參數群組中啟用下列參數。 設定`enforce_gtid_consistency` 為`on`，並設定`gtid-mode` 為`on`。	DBA
重新啟動 Amazon RDS for MySQL 適用於 MySQL 執行個體。	需要重新開機，參數變更才會生效。	DBA
建立使用者並授與其複寫權限。	要安裝 MySQL，請使用以下命令。 `CREATE USER 'repl'@'%' IDENTIFIED BY 'xxxx';` `GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%' ;` `FLUSH PRIVILEGES;`	DBA

在亞馬遜 EC2 執行個體上安裝和準備 MySQL

任務	描述	所需技能
在 Amazon Linux 上安裝 MySQL。	要安裝 MySQL，請使用以下命令。 `sudo yum update` `sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm` `sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm` `sudo yum install mysql-community-server` `sudo systemctl start mysqld`	DBA
在 EC2 執行個體上登入 MySQL 並建立資料庫。	資料庫名稱應與 Amazon RDS for MySQL (MySQL 版) 中的資料庫名稱相同。在下列範例中，資料庫名稱為`replication` 。 `create database replication;`	DBA
編輯 MySQL 配置文件，然後重新啟動數據庫。	`/etc/`透過新增下列參數來編輯位於中的`my.conf`檔案。 `server-id=3` `gtid_mode=ON` `enforce_gtid_consistency=ON` `replicate-ignore-db=mysql`	DBA

任務	描述	所需技能
	`binlog-format=ROW` `log_bin=mysql-bin` 然後重新啟動`mysqld`服務。 `systemctl mysqld restart`	

設定複製

任務	描述	所需技能
從 Amazon RDS for MySQL 適用於 MySQL 資料庫匯出資料傾印。	若要從 Amazon RDS for MySQL 版 MySQL 匯出傾印，請使用下列命令。 `mysqldump --single-transaction -h mydb.xxxxxxx.amazonaws.com -uadmin -p --databases replication > replication-db.sql`	DBA
在 Amazon EC2 上的 MySQL 數據庫中恢復 .sql 轉儲文件。	要將轉儲導入 Amazon EC2 上的 MySQL 數據庫，請使用以下命令。 `mysql -D replication -uroot -p < replication-db.sql`	DBA
將 Amazon EC2 上的 MySQL 資料庫設定為複本。	若要開始複寫並檢查複寫狀態，請登入 Amazon EC2 上的 MySQL 資料庫，然後使用下列命令。	DBA

任務	描述	所需技能
	`CHANGE MASTER TO MASTER_HOST="mydb.xxxxxxx.amazonaws.com", MASTER_USER="repl", MASTER_PASSWORD="rep123", MASTER_PORT=3306, MASTER_AUTO_POSITION = 1;` `START SLAVE;` `SHOW SLAVE STATUS\G`	

相關資源

- Amazon EC2 Linux 執行個體使用者指南
- 使用 MySQL 百勝儲存庫在 Linux 上安裝 MySQL
- 使用全域交易識別碼複寫
- 使用以 GTID 為基礎的複寫適用於 Amazon RDS for MySQL

甲骨文 PeopleSoft 應用程式在 Amazon RDS 自訂的轉換角色

創建者：自動化（AWS）

環境：生產	技術：資料庫；基礎架構	工作量：甲骨文
AWS 服務：Amazon RDS		

Summary

若要在 Amazon Web Services (AWS) 上執行 Oracle PeopleSoft 企業資源規劃 (ERP) 解決方案，您可以使用 Amazon Relational Database Service 服務 (Amazon RDS) 或 Amazon RDS 自訂軟體，該服務支援需要存取基礎作業系統 (OS) 和資料庫環境的傳統、自訂和封裝應用程式。如需規劃移轉時要考量的關鍵因素，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

此模式著重於執行 Oracle 資料保全轉換或角色轉換的步驟，在 Amazon RDS Custom 上執行作為具有僅供讀取複本資料庫的主要資料庫的 PeopleSoft 應用程式資料庫執行 Oracle 資料保全轉換或角色轉換。此模式包含設定快速啟動容錯移轉 (FSFO) 的步驟。在此處理作業期間，「Oracle 資料保全」組態中的資料庫會繼續在其新角色中運作。Oracle Data Guard 轉換的典型使用案例包括災難復原 (DR) 演練、資料庫上的排定維護活動，以及待命優先的修正程式套用機動修正程式。如需詳細資訊，請參閱部落格文章減少 Amazon RDS 自訂中的資料庫修補停機時間。

先決條件和限制

前提

- 使用僅供讀取複本模式完成將 HA 新增至 Amazon RDS 自訂。 PeopleSoft

限制

- 適用於 Oracle 的 RDS 自訂的限制和不支援的組態
- 與適用於甲骨文僅供讀取複本的 Amazon RDS 自訂相關限制

產品版本

- 如需 Amazon RDS 自訂支援的 Oracle 資料庫版本，請參閱適用於 Oracle 的 RDS 自訂。

- 如需 Amazon RDS 自訂支援的 Oracle 資料庫執行個體類別，請參閱適用於 Oracle RDS 自訂的資料庫執行個體類別支援。

架構

技術, 堆

- Amazon RDS Custom for Oracle

目標架構

下圖顯示 Amazon RDS 自訂資料庫執行個體和 Amazon RDS 自訂僅供讀取複本。Oracle 資料保全為 DR 提供容錯移轉期間的角色轉換。

如需使用 AWS PeopleSoft 上 Oracle 的代表性架構，請參閱在 AWS 上設定高可用性 PeopleSoft 架構。

工具

AWS 服務

- Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。在此模式中，您可以使用密碼名稱從「秘密管理員」擷取資料庫使用者密碼`do-not-delete-rds-custom-+<<RDS Resource ID>>+-dg`。`RDS_DATAGUARD`

其他服務

- 「Oracle 資料保全」可協助您建立、維護、管理及監督待命資料庫。此病毒碼會使用「Oracle 資料保全最大效能」來轉換角色 (Oracle 資料保全切換)。

最佳實務

對於生產部署，我們建議在第三個可用區域中啟動觀察器執行個體，與主要和僅供讀取複本節點分開。

史诗

啟動角色轉換

任務	描述	所需技能
暫停主要和複本的資料庫自動化。	雖然 RDS 自訂自動化架構不會干擾角色轉換程序，但最好在 Oracle 資料保全切換期間暫停自動化。 若要暫停和繼續 RDS 自訂資料庫自動化，請遵循暫停和繼續 RDS 自訂自動化中的指示。	雲端管理員，DBA
檢查「Oracle 資料保全」狀態。	若要檢查「Oracle 資料保全」狀態，請登入主要資料庫。此模式包含使用多租戶容器資料庫 (CDB) 或非 CDB 執行個體的程式碼。 非國家開發行 `-bash-4.2$ dgmgrl RDS_DATAGUARD@RDS_CUSTOM_ORCL_A` `DGMGRL for Linux: Release 19.0.0.0.0 - Production on Mon Nov 28 20:55:50 2022` `Version 19.10.0.0.0` `Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.` `Welcome to DGMGRL, type "help" for informati on.` `Password:` `Connected to "ORCL_A"` `Connected as SYSDG.`	DBA

任務	描述	所需技能
	DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 59 seconds ago) DGMGRL> **國家開發行** CDB-bash-4.2$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_A DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 18 06:13:07 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_A " Connected as SYSDG. DGMGRL> show configura tion Configuration - rds_dg	

任務	描述	所需技能
	Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL>	
驗證執行個體角色。	開啟 AWS 管理主控台，然後導覽至 Amazon RDS 主控台。在資料庫的 [複寫] 區段的 [連線與安全性] 索引標籤上，確認主要和複本的執行個體角色。 主要角色應與「Oracle 資料保全」主要資料庫相符，且複本角色應符合「Oracle 資料保全」實體待命資料庫。	雲端管理員，DBA

任務	描述	所需技能
執行切換。	若要執行轉換，請DGMGRL從主要節點連接到。 非國家開發行 `DGMGRL> switchover to orcl_d;` `Performing switchover NOW, please wait...` `Operation requires a connection to database "orcl_d"` `Connecting ...` `Connected to "ORCL_D"` `Connected as SYSDG.` `New primary database "orcl_d" is opening...` `Operation requires start up of instance "ORCL" on database "orcl_a"` `Starting instance "ORCL"...` `Connected to an idle instance.` `ORACLE instance started.` `Connected to "ORCL_A"` `Database mounted.` `Database opened.` `Connected to "ORCL_A"` `Switchover succeeded, new primary is "orcl_d"` `DGMGRL>` 國家開發行 `DGMGRL> switchover to rdscdb_b` `Performing switchover NOW, please wait...`	DBA

任務	描述	所需技能
	New primary database "rdscdb_b" is opening... Operation requires start up of instance "RDSCDB" on database "rdscdb_a" Starting instance "RDSCDB"... Connected to an idle instance. ORACLE instance started. Connected to "RDSCDB_A" Database mounted. Database opened. Connected to "RDSCDB_A" Switchover succeeded, new primary is "rdscdb_b"	

任務	描述	所需技能
驗證「Oracle 資料保全」連線。	切換之後，請驗證從主節點到的「Oracle 資料保全」連線。DGMGRL 非國家開發行 `DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database orcl_a - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 60 seconds ago) DGMGRL>` `DGMGRL> show configura tion lag; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database orcl_a - Physical standby database Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Fast-Start Failover: Disabled`	DBA

任務	描述	所需技能
	Configuration Status: SUCCESS (status updated 44 seconds ago) DGMGRL> 國家開發行 DGMGRL> show configura tion DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_b - Primary database rdscdb_a - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 52 seconds ago) DGMGRL> DGMGRL> show configura tion lag Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_b - Primary database rdscdb_a - Physical standby database Transport Lag: 0 seconds (computed 0 seconds ago)	

任務	描述	所需技能
	`Apply Lag: 0 seconds (computed 0 seconds ago)` `Fast-Start Failover: Disabled` `Configuration Status: SUCCESS (status updated 53 seconds ago)` `DGMGRL>`	
在 Amazon RDS 主控台上驗證執行個體角色。	執行角色切換後，Amazon RDS 主控台會在「資料庫」下「連線與安全」索引標籤的「複寫」區段下顯示新角色。複寫狀態可能需要幾分鐘的時間才能從空白更新為複寫。	DBA

設定 FSFO

任務	描述	所需技能
重設切換。	將切換設定回主要節點。	DBA
安裝並啟動觀察者。	觀察者處理序是DGMGRL用戶端元件，通常在與主要資料庫和待命資料庫不同的機器上執行。觀察者的「ORACLE 本位目錄」安裝可以是「Oracle 客戶端管理員」安裝，或者您可以安裝「Oracle 資料庫企業版」或「個人版」。有關數據庫版本的觀察者安裝的詳細信息，請參閱安裝和啟動觀察者。要配置觀察者進程的高可	DBA

任務	描述	所需技能
	用性，您可能需要執行以下操作： • 為執行觀察者的 EC2 執行個體啟用 EC2 執行個體自動復原。作為操作系統啟動的一部分，您需要自動執行觀察者啟動過程。 • 在 EC2 執行個體中部署觀察者，並設定大小為一 (1) 的 Amazon EC2 Auto Scaling 群組。如果 EC2 執行個體發生故障，自動擴展群組會自動啟動另一個 EC2 執行個體。 對于 Oracle 12c 版本 2 和更高版本，您最多可以部署三名觀察者。一位觀察者是主要觀察者，其餘的則是備份觀察者。當主要觀察者失敗時，其中一位備份觀察者將扮演主要角色。	

任務	描述	所需技能
從觀察者主機 Connect 到 DGMGRL。	觀察者主機設定了主要和待命資料庫連線的`tnsnames.ora` 項目。只要資料遺失在FastStartFailoverLagLimit組態內 (以秒為單位)，您就可以啟用具有最大效能保護模式的 FSFO。不過，您必須使用最大可用性保護模式才能達到零資料遺失 (RPO=0)。 非國家開發行 `DGMGRL> show configura tion; Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 58 seconds ago) DGMGRL> show configura tion lag Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_a - Primary database orcl_d - Physical standby database`	DBA

任務	描述	所需技能
	Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 5 seconds ago) DGMGRL>	
	國家開發行 -bash-4.2$ dgmgrl C##RDS_DATAGUARD@R DS_CUSTOM_RDSCDB_A DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Jan 18 06:55:09 2023 Version 19.16.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for informati on. Password: Connected to "RDSCDB_A " Connected as SYSDG. DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database	

任務	描述	所需技能
	rdscdb_b - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 18 seconds ago) DGMGRL>	

任務	描述	所需技能
將待命資料庫修改為容錯移轉目標。	從主節點或觀察者節點 Connect 到一個待命資料庫。雖然您的設定可能有多個待命資料庫，但您目前只需要連線到一個資料庫。) 非國家開發行 `DGMGRL> edit database orcl_a set property FastStartFailoverTarget='orcl_d';` `Property "faststartfailovertarget" updated` `DGMGRL> edit database orcl_d set property FastStartFailoverTarget='orcl_a';` `Property "faststartfailovertarget" updated` `DGMGRL> show database orcl_a FastStartFailoverTarget;` `FastStartFailoverTarget = 'orcl_d'` `DGMGRL> show database orcl_d FastStartFailoverTarget;` `FastStartFailoverTarget = 'orcl_a'` `DGMGRL>` 國家開發行 `DGMGRL> edit database orcl_a set property`	DBA

任務	描述	所需技能
	FastStartFailoverT arget='rdscdb_b'; Object "orcl_a" was not found DGMGRL> edit database rdscdb_a set property FastStartFailoverT arget='rdscdb_b'; Property "faststar tfailovertarget" updated DGMGRL> edit database rdscdb_b set property FastStartFailoverT arget='rdscdb_a'; Property "faststar tfailovertarget" updated DGMGRL> show database rdscdb_a FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_b' DGMGRL> show database rdscdb_b FastStart FailoverTarget; FastStartFailoverT arget = 'rdscdb_a' DGMGRL>	

任務	描述	所需技能
設定 FastStartFailoverThreshold 與 DGMGRL 的連線。	在甲骨文 19c 中，默認值為 30 秒，最小值為 6 秒。較低的值可能會在容錯移轉期間縮短復原時間目標 (RTO)。較高的值有助於減少主要資料庫發生不必要的容錯移轉暫時性錯誤的機會。 適用於 Oracle 自動化架構的 RDS 自訂會監控資料庫健全狀況，並每隔幾秒鐘執行更正動作。因此，我們建議設 FastStartFailoverThreshold 定為大於 10 秒的值。下列範例會將臨界值設定為 35 秒。 非 CBD 或國家開發行 `DGMGRL> edit configuration set property FastStartFailoverThreshold=35;` `Property "faststartfailoverthreshold" updated` `DGMGRL> show configuration FastStartFailoverThreshold;` `FastStartFailoverThreshold = '35'` `DGMGRL>`	DBA

任務	描述	所需技能
透過從主要節點或觀察者節點連線至 DGMGRL 來啟用 FSFO。	如果資料庫未啟用「倒溯資料庫」，則ORA-16827 會顯示警告訊息。如果FastStart FailoverAutoReinstate組態特性設為 TRUE (預設值)，選擇性的倒溯資料庫可協助自動將失敗的主要資料庫恢復到容錯移轉之前的某個時間點。 非國家開發行 `DGMGRL> enable fast_start failover;` `Warning: ORA-16827: Flashback Database is disabled` `Enabled in Zero Data Loss Mode.` `DGMGRL>` `DGMGRL> show configuration` `Configuration - rds_dg` `Protection Mode: MaxAvailability` `Members:` `orcl_a - Primary database` `Warning: ORA-16819: fast-start failover observer not started` `orcl_d - (*) Physical standby database` `Warning: ORA-16819: fast-start failover observer not started` `Fast-Start Failover: Enabled in Zero Data Loss Mode` `Configuration Status:`	DBA

任務	描述	所需技能
	WARNING (status updated 29 seconds ago) DGMGRL> 國家開發行 DGMGRL> enable fast_start failover; Warning: ORA-16827: Flashback Database is disabled Enabled in Zero Data Loss Mode. DGMGRL> show configuration; Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database Warning: ORA-16819: fast-start failover observer not started rdscdb_b - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 11 seconds ago) DGMGRL>	

任務	描述	所需技能
啟動 FSFO 監視的觀察者，並驗證狀態。	您可以在啟用 FSFO 之前或之後啟動觀察者。如果 FSFO 已啟用，觀察者會立即開始監視主要和目標待命資料庫的狀態和連線。如果未啟用 FSFO，則在啟用 FSFO 之後，觀察者才會開始監視。 當你啟動觀察者時，主數據庫配置將被顯示，沒有任何錯誤消息，如前面的命令所證明。`show configuration` 非國家開發行 `DGMGRL> start observer;` `[W000 2022-12-0` `1T06:16:51.271+00:00]` `FSFO target standby is` `orcl_d` `Observer 'ip-10-0-` `1-89' started` `[W000 2022-12-0` `1T06:16:51.352+00:00]` `Observer trace level is` `set to USER` `DGMGRL> show configura` `tion` `Configuration - rds_dg` `Protection Mode:` `MaxAvailability` `Members:` `orcl_a - Primary` `database` `orcl_d - (*) Physical` `standby database`	DBA

任務	描述	所需技能
	Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: SUCCESS (status updated 56 seconds ago) DGMGRL> DGMGRL> show observer Configuration - rds_dg Primary: orcl_a Active Target: orcl_d Observer "ip-10-0- 1-89" - Master Host Name: ip-10-0-1 -89 Last Ping to Primary: 1 second ago Last Ping to Target: 1 second ago DGMGRL> **國家開發行** DGMGRL> start observer; Succeeded in opening the observer file "/home/oracle/fsfo _ip-10-0-1-56.dat". [W000 2023-01-1 8T07:31:32.589+00:00] FSFO target standby is rdscdb_b Observer 'ip-10-0- 1-56' started The observer log file is '/home/oracle/obse rver_ip-10-0-1-56. log'.	

任務	描述	所需技能
	DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: rdscdb_a - Primary database rdscdb_b - (*) Physical standby database Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: SUCCESS (status updated 12 seconds ago) DGMGRL> DGMGRL> show observer; Configuration - rds_dg Primary: rdscdb_a Active Target: rdscdb_b Observer "ip-10-0- 1-56" - Master Host Name: ip-10-0-1-56 Last Ping to Primary: 1 second ago Last Ping to Target: 2 seconds ago DGMGRL>	

任務	描述	所需技能
確認容錯移轉。	在這個案例中，可以透過手動停止主要 EC2 執行個體來執行容錯移轉測試。停止 EC2 執行個體之前，請使用`tail`指令根據您的組態監視觀察者記錄檔。用`DGMGRL`於與使用者一`orcl_d`起登入待命資料庫`RDS_DATAGUARD` ，並檢查「Oracle 資料保全」狀態。它應該表明這`orcl_d`是新的主數據庫。 注意：在此容錯移轉測試案例中，`orcl_d`是非 CDB 資料庫。 容錯移轉之前，已在`orcl_a`啟用倒溯資料庫。在前一個主要資料庫返回線上`MOUNT`狀態並啟動之後，觀察者會將其還原到新的待命資料庫中。復原的資料庫會做為新主要資料庫的 FSFO 目標。您可以在觀察者日誌中驗證詳細信息。 `DGMGRL> show configura tion Configuration - rds_dg Protection Mode: MaxAvailability Members: orcl_d - Primary database Warning: ORA-16824 : multiple warnings, including fast-star t failover-related`	DBA

任務	描述	所需技能
	warnings, detected for the database orcl_a - (*) Physical standby database (disabled) ORA-16661: the standby database needs to be reinstated Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status: WARNING (status updated 25 seconds ago) DGMGRL> 以下顯示中的範例輸出observer.log 。 $ tail -f /tmp/observer.log Unable to connect to database using rds_custom_orcl_a [W000 2023-01-18T07:50:32.589+00:00] Primary database cannot be reached. [W000 2023-01-18T07:50:32.589+00:00] Fast-Start Failover threshold has expired. [W000 2023-01-18T07:50:32.590+00:00] Try to connect to the standby. [W000 2023-01-18T07:50:32.590+00:00] Making a last connection attempt to primary database before	

任務	描述	所需技能
	proceeding with Fast-Start Failover. [W000 2023-01-18T07:50:32.591+00:00] Check if the standby is ready for failover. [S002 2023-01-18T07:50:32.591+00:00] Fast-Start Failover started... 2023-01-18T07:50:32.591+00:00 Initiating Fast-Start Failover to database "orcl_d"... [S002 2023-01-18T07:50:32.592+00:00] Initiating Fast-start Failover. Performing failover NOW, please wait... Failover succeeded, new primary is "orcl_d" 2023-01-18T07:55:32.101+00:00 [S002 2023-01-18T07:55:32.591+00:00] Fast-Start Failover finished... [W000 2023-01-18T07:55:32.591+00:00] Failover succeeded. Restart pinging. [W000 2023-01-18T07:55:32.603+00:00] Primary database has changed to orcl_d. [W000 2023-01-18T07:55:33.618+00:00] Try to connect to the primary.	

任務	描述	所需技能
	[W000 2023-01-18T07:55:33.622+00:00] Try to connect to the primary rds_custom_orcl_d. [W000 2023-01-18T07:55:33.634+00:00] The standby orcl_a needs to be reinstated [W000 2023-01-18T07:55:33.654+00:00] Try to connect to the new standby orcl_a. [W000 2023-01-18T07:55:33.654+00:00] Connection to the primary restored! [W000 2023-01-18T07:55:35.654+00:00] Disconnecting from database rds_custom_orcl_d. [W000 2023-01-18T07:55:57.701+00:00] Try to connect to the new standby orcl_a. ORA-12170: TNS:Connect timeout occurred	

設定 Oracle 應用程式與資料庫之間的連線

任務	描述	所需技能
在主要資料庫中建立並啟動服務。	您可以使用組態中同時包含主要和待命資料庫端點的 TNS 項目，避免在角色轉換期間變更應用程式組態。您可以定義兩個以角色為基礎的資料庫服	DBA

任務	描述	所需技能
	務，以同時支援讀取/寫入和唯讀 在下列範例中，orcl_rw是主要資料庫上作用中的讀取/寫入服務。 orcl_ro是唯讀服務，且在以唯讀模式開啟的待命資料庫上處於作用中狀態。 `SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- ORCL READ WRITE SQL> exec dbms_serv ice.create_service ('orcl_rw','orcl_r w'); PL/SQL procedure successfully completed . SQL> exec dbms_serv ice.create_service ('orcl_ro','orcl_r o'); PL/SQL procedure successfully completed .`  `SQL> exec dbms_serv ice.start_service('orcl_rw'); PL/SQL procedure successfully completed . SQL>`	

任務	描述	所需技能
啟動待命資料庫中的服務。	若要在唯讀待命資料庫中啟動服務，請使用下列程式碼。 `SQL> select name,open _mode from v$database; NAME OPEN_MODE --------- --------- ----------- ORCL READ ONLY WITH APPLY SQL> exec dbms_serv ice.start_service( 'orcl_ro'); PL/SQL procedure successfully completed . SQL>`	DBA

任務	描述	所需技能
重新啟動主資料庫時，自動啟動服務。	若要在重新啟動主要資料庫時自動啟動服務，請使用下列程式碼。 `SQL> CREATE OR REPLACE TRIGGER TrgDgServices after startup on database DECLARE db_role VARCHAR(30); db_open_mode VARCHAR(30); BEGIN SELECT DATABASE_ROLE, OPEN_MODE INTO db_role, db_open_mode FROM V$DATABASE; IF db_role = 'PRIMARY' THEN DBMS_SERV 2 ICE.START_SERVICE('orcl_rw'); END IF; IF db_role = 'PHYSICAL STANDBY' AND db_open_mode LIKE 'READ ONLY%' THEN DBMS_SERVICE.START_SERVICE('orcl_ro'); END IF; END; / Trigger created. SQL>`	DBA

任務	描述	所需技能
設定讀取/寫入和唯讀資料庫之間的連線。	您可以使用下列應用程式組態範例來執行讀取/寫入和唯讀連線。 `ORCL_RW = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=20)(RETRY_DELAY=3)(TRANSPORT_CONNECT_ TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftdb.******.us-west-2 .rds.amazonaws.com) (PORT=1521)) (ADDRESS = (PROTOCOL = TCP)(HOST=psftread .******.us-west-2. rds.amazonaws.com) (PORT=1521))) (CONNECT_DATA=(SERVIC E_NAME = orcl_rw))) ORCL_RO = (DESCRIPTION = (CONNECT_TIMEOUT= 120)(RETRY_COUNT=20)(RETRY_DELAY=3)(TRANSPORT_CONNECT_ TIMEOUT=3) (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST=devpsftdb.******.us-west-2 .rds.amazonaws.com) (PORT=1521)) (ADDRESS = (PROTOCOL = TCP)(HOST=psftread`	DBA

任務	描述	所需技能
	.******.us-west-2. rds.amazonaws.com) (PORT=1521))) (CONNECT_DATA=(SERVIC E_NAME = orcl_ro)))	

相關資源

- 在 Amazon RDS 上使用資料保全啟用高可用性 (AWS 技術指南)
- 將 Amazon RDS 設定為甲骨文 PeopleSoft 資料庫 (AWS 白皮書)
- Oracle 資料保全中介指南 (Oracle 參考文件)
- 資料保全概念與管理 (Oracle 參考文件)
- Oracle 資料保全特定風扇與 FCF 組態需求 (Oracle 參考文件)

依工作負載的資料庫移轉

主題

- IBM
- Microsoft
- N/A
- 开源
- Oracle
- SAP

IBM

- 使用 AWS DMS 將 Db2 資料庫從 Amazon EC2 遷移到與 MySQL 相容的 Aurora
- 使用日誌傳送將 LUW 的 Db2 移轉至 Amazon EC2，以減少中斷時間
- 透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2
- 使用 AWS DMS 和 AWS SCT，從 Amazon EC2 上的 IBM Db2 遷移到 Aurora 與 PostgreSQL 相容
- 從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的阿帕奇 Tomcat
- 使用受信任的內容，在 AWS 上的 Db2 聯合資料庫中保護和簡化使用者存取

Microsoft

- 加速 Microsoft 工作負載探索和移轉到 AWS
- 使用連結伺服器從 Amazon EC2 上的 Microsoft SQL 伺服器存取現場部署 Microsoft SQL 伺服器表
- 評估將 SQL 伺服器資料庫遷移到 AWS 上的 MongoDB 地圖集的查詢效能
- 更改 Python 和 Perl 應用程序以支持從 Microsoft SQL 服務器遷移到 Amazon Aurora PostgreSQL 兼容版本的數據庫
- 在 AWS 上的 SQL Server 中的「永遠開啟」可用性群組中設定唯讀路由
- 使用 Microsoft Excel 和 Python 為 AWS DMS 任務建立 AWS CloudFormation 範本
- 使用 AWS DMS 將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3
- 使用 AWS DMS 將 Amazon RDS for SQL Server 資料表匯出到 S3 儲存貯體
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 將簡訊佇列從 Microsoft Azure 服務匯流排遷移到 Amazon SQS
- 通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB
- 使用 AWS DMS 和 AWS SCT 將 Microsoft SQL 伺服器資料庫遷移到 Aurora MySQL
- 將 .NET 應用程式從 Microsoft Azure 應用程式服務遷移到 AWS Elastic Beanstalk
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon EC2
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server
- 使用連結伺服器將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server 服器
- 使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將現場部署 Microsoft SQL 伺服器資料庫遷移至 Amazon Redshift 移
- ???
- 使用複製將資料從 Microsoft Azure Blob 遷移到 Amazon S3
- 使用分散式可用性群組將 SQL 伺服器遷移到 AWS
- 使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器
- ???
- 使用現場部署 SMTP 伺服器和資料庫郵件傳送 Amazon RDS for SQL Server 服器資料庫執行個體的通知

- 使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施

N/A

- 在將主機移轉至 AWS 期間建立防火牆請求的核准程序
- 加密現有的亞馬遜 RDS 資料庫執行個體
- 估算 Amazon DynamoDB 表格的儲存成本
- 使用 AWS DMS 和 Amazon Aurora 實作跨區域災難復原

开源

- ???
- 在與 PostgreSQL 相容的 Aurora 中建立應用程式使用者和角色
- 在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線
- ???
- 將現場部署 MySQL 資料庫遷移到 Amazon EC2
- 將現場部署 MySQL 資料庫遷移到 Amazon RDS for MySQL
- 將內部部署 MySQL 資料庫遷移至 Aurora MySQL
- 將內部 PostgreSQL 料庫遷移至 Aurora
- 使用 Auto Scaling 能從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的 Apache Tomcat
- 使用 SharePlex 和 AWS DMS 從甲骨文 8i 或 9i 遷移到適用於甲骨文的亞馬遜 RDS
- 從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk
- 使用合格邏輯從 Amazon EC2 上的 PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS 應用程式容器將現場部署 Java 應用程式遷移到 AWS
- 使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL
- 將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容
- 將具有 100 個以上引數的甲骨文函數和程序遷移到 PostgreSQL
- 將 Redis 工作負載遷移到 AWS 上的 Redis 企業雲端
- 監控 Amazon Aurora 是否有沒有加密的
- 重新啟動 RHEL 來源伺服器後，自動重新啟動 AWS 複寫代理程式而不停用 SELinux
- 使用 Lambda 和機 Secrets Manager 為亞馬遜 RDS 和 Aurora PostgreSQL 安排任務
- 使用 GTID 在亞馬遜 EC2 上設置 Amazon RDS for MySQL 和 MySQL 之間的數據複寫
- 使用傳輸在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫

Oracle

- 使用僅供讀取複本將 HA 新增至 Oracle PeopleSoft Amazon 自訂
- 設定 Oracle 資料庫與 Aurora 相容之間的連結
- 將甲骨文查詢轉換為 SQL 數據庫
- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR
- 使用 Aurora 中的自訂端點模擬 Oracle RAC 工作負載
- 使用 AWR 報告估計甲骨文資料庫的 Amazon RDS 引擎大小
- 處理動態 SQL 語句中 Aurora PostgreSQL 塊
- 在 Aurora 兼容後處理過載的甲骨文功能
- 使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS
- ???
- 將適用於 Oracle 資料庫執行個體的 Amazon RDS 移轉到使用 AMS 的其他帳戶
- 使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS SCT 和 AWS DMS 將適用於甲骨文的亞馬遜 RDS 遷移到適用於 PostgreSQL 的 CLI 馬遜 RDS CloudFormation
- ???
- 將 Amazon RDS for Oracle 執行個體遷移到另一個 VPC
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到 Amazon EC2
- 使用 Logstash 將現場部署 Oracle 資料庫遷移至 Amazon OpenSearch 服務
- 使用 AWS DMS 和 AWS SCT 將現場部署甲骨文資料庫遷移到適用於 MySQL 的 Amazon RDS for MySQL
- 將現場部署甲骨文資料庫遷移到 Amazon RDS for Oracle
- 透過資料庫連結使用直接的 Oracle 資料汲取匯入，將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon RDS
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到亞馬遜 RDS
- 使用甲骨文旁觀者和 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS
- 將現場部署 Oracle 資料庫遷移到 Amazon EC2 上的甲骨文
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS

- 使用 AWS DMS 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 將甲骨文資料庫遷移到 Amazon DynamoDB 資料庫
- 使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Aurora
- 使用 Oracle 資料泵和 AWS DMS 將甲骨文 JD 愛德華資料 EnterpriseOne 庫遷移到 AWS
- 使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL
- 使用 AWS DMS 將甲骨文 PeopleSoft 資料庫遷移到 AWS
- 將資料從內部部署 Oracle 資料庫遷 PostgreSQL 至 Aurora
- 從 Amazon RDS for Oracle 遷移到 Amazon RDS for MySQL
- 使用具體化視圖和 AWS DMS，從甲骨文 8i 或 9i 遷移到亞馬遜 RDS
- 使用和 AWS DMS 從甲骨文 8i 或 9i 遷移到亞馬遜 RDS SharePlex
- 使用甲骨文從甲骨文數據庫遷移到亞馬遜 RDS GoldenGate
- ???
- 使用 AWS DMS 從甲骨文遷移到 Amazon DocumentDB
- 在 Amazon ECS 上從甲骨文遷移 WebLogic 到阿帕奇湯姆貓（TomEE）
- 將基於函數的索引從甲骨文遷移到 PostgreSQL
- 將舊有應用程式從 Oracle Pro*C 移轉至 ECPG
- 將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列
- 將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫
- 將 Oracle 電子商務套件遷移到 Amazon RDS 定制
- 使用擴充功能將甲骨文原生函數遷移至 PostgreSQL
- 將甲骨文輸出綁定變量遷移到 PostgreSQL 數據庫
- 將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制
- 將甲骨文功能遷移到 AWS 上的 PostgreSQL
- 將甲骨文序列遷移 _ 可重複使用的編譯包到 PostgreSQL
- 將虛擬生成的列從甲骨文遷移到 PostgreSQL
- 使用 Amazon 監控甲骨文 GoldenGate 日誌 CloudWatch
- 在適用於甲骨文的亞馬遜 RDS 上將 Oracle 數據庫企業版重新平台為標準版 2
- 在 Amazon RDS 上為甲骨文電子商務套件設置 HA/DR 架構，並使用活動備用數據庫

- 在 Aurora 相容上設定甲骨文 UTL_FILE 功能
- 甲骨文 PeopleSoft 應用程式在 Amazon RDS 自訂的轉換角色
- 從甲骨文遷移到 Amazon Aurora PostgreSQL 後驗證數據庫對象

SAP

- 使用 Systems Manager 和自動備份 SAP HANA 資料庫 EventBridge
- 將現場部署 SAP ASE 資料庫遷移至 Amazon EC2
- 使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器
- 使用 AWS SCT 和 AWS DMS 將亞馬 Amazon EC2 上的 SAP ASE 遷移到與 Amazon Aurora PostgreSQL 相容
- ???
- 使用應用程式遷移服務減少同質 SAP 移轉切換時間
- 在 AWS 上的 IBM Db2 上為 SAP 設定災難復原

更多模式

- 使用 Athena 存取、查詢和加入 Amazon DynamoDB 資料表
- 彙總 Amazon DynamoDB 中的資料，用於 Athena 的機器學習預測
- 允許 EC2 執行個體寫入 AMS 帳戶中 S3 儲存貯體的存取權
- 使用 Amazon 雅典娜和亞馬遜分析和視覺化嵌套 JSON 數據 QuickSight
- 使用 AWS Directory Service 在 Amazon EC2 上驗證 Microsoft SQL 服務器
- 使用 AWS Batch 為 Amazon RDS for PostgreSQL 資料庫執行個體自動備份
- 使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3
- 使用 Python 應用程式為亞馬遜動態 B 自動產生模型和 CRUD 函數
- 自動修復未加密的 Amazon RDS 資料庫執行個體和叢集
- ???
- 使用 DevOps 實務和 AWS Cloud9 建立鬆散結合的架構與微型服務
- 更改 Python 和 Perl 應用程序以支持從 Microsoft SQL 服務器遷移到 Amazon Aurora PostgreSQL 兼容版本的數據庫
- 設定對 Amazon DynamoDB 的跨帳戶存取權
- 設定 Oracle 資料庫與 Aurora 相容之間的連結
- 使用 Python 在 AWS 上將 EBCDIC 資料轉換並解壓縮為 ASCII
- 將太數據標準化時間功能轉換為 Amazon Redshift SQL
- 將太數據重置功能轉換為 Amazon Redshift SQL
- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 在與 PostgreSQL 相容的 Aurora 中建立應用程式使用者和角色
- 使用 Microsoft Excel 和 Python 為 AWS DMS 任務建立 AWS CloudFormation 範本
- ???
- 使用私有靜態 IP 在 Amazon EC2 上部署卡桑德拉集群，以避免重新平衡
- 使用 RAG 和提示，開發先進的生成式 AI 聊天助理 ReAct
- 使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR
- 在 Amazon RDS for SQL Server 中啟用透明資料加密
- 使用 AWS DMS 將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3
- 使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS
- ???

- 使用 AWS 秘密管理員來管理登入
- 使用 AWS DMS 將 Db2 資料庫從 Amazon EC2 遷移到與 MySQL 相容的 Aurora
- 通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB
- 使用 AWS DMS 和 AWS SCT 將 Microsoft SQL 伺服器資料庫遷移到 Aurora MySQL
- 將自我託管的 MongoDB 環境遷移到 AWS 雲端上的 MongoDB 地圖集
- 使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 移
- 使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS SCT 和 AWS DMS 將適用於甲骨文的亞馬遜 RDS 遷移到適用於 PostgreSQL 的 CLI 馬遜 RDS CloudFormation
- 將 Amazon RDS 資料庫執行個體遷移到另一個 VPC 端或帳戶
- ???
- 將 Amazon RDS for Oracle 執行個體遷移到另一個 VPC
- 將 Amazon Redshift 叢集遷移到中國的 AWS 區域
- ???
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon EC2
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server
- 使用連結伺服器將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server 服器
- 使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將現場部署 Microsoft SQL 伺服器資料庫遷移至 Amazon Redshift 移
- ???
- 將現場部署 MySQL 資料庫遷移到 Amazon EC2
- 將現場部署 MySQL 資料庫遷移到 Amazon RDS for MySQL
- 將內部部署 MySQL 資料庫遷移至 Aurora MySQL
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到 Amazon EC2
- 使用 Logstash 將現場部署 Oracle 資料庫遷移至 Amazon OpenSearch 服務
- 使用 AWS DMS 和 AWS SCT 將現場部署甲骨文資料庫遷移到適用於 MySQL 的 Amazon RDS for MySQL

- 將現場部署甲骨文資料庫遷移到 Amazon RDS for Oracle
- 透過資料庫連結使用直接的 Oracle 資料汲取匯入，將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon RDS
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到亞馬遜 RDS
- 使用甲骨文旁觀者和 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS
- 將現場部署 Oracle 資料庫遷移到 Amazon EC2 上的甲骨文
- 將內部 PostgreSQL 料庫遷移至 Aurora
- 將現場部署 SAP ASE 資料庫遷移至 Amazon EC2
- 將現場部署 ThoughtSpot 獵鷹資料庫遷移到 Amazon Redshift
- 使用 AWS SCT 資料擷取代理程式將現場部署 Vertica 資料庫遷移到 Amazon Redshift 移
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 將甲骨文資料庫遷移到 Amazon DynamoDB 資料庫
- 使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Aurora
- 使用 Oracle 資料泵和 AWS DMS 將甲骨文 JD 愛德華資料 EnterpriseOne 庫遷移到 AWS
- 使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL
- 使用 AWS DMS 將甲骨文 PeopleSoft 資料庫遷移到 AWS
- 將資料從內部部署 Oracle 資料庫遷 PostgreSQL 至 Aurora
- 使用星爆將資料遷移到 AWS 雲端
- 使用日誌傳送將 LUW 的 Db2 移轉至 Amazon EC2，以減少中斷時間
- 透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2
- 從 Amazon RDS for Oracle 遷移到 Amazon RDS for MySQL
- ???
- 使用 AWS DMS 和 AWS SCT，從 Amazon EC2 上的 IBM Db2 遷移到 Aurora 與 PostgreSQL 相容
- 使用具體化視圖和 AWS DMS，從甲骨文 8i 或 9i 遷移到亞馬遜 RDS
- 使用和 AWS DMS 從甲骨文 8i 或 9i 遷移到亞馬遜 RDS SharePlex
- 使用甲骨文從甲骨文數據庫遷移到亞馬遜 RDS GoldenGate
- ???
- 使用 AWS DMS 從甲骨文遷移到 Amazon DocumentDB

- 使用合格邏輯從 Amazon EC2 上的 PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器
- 將基於函數的索引從甲骨文遷移到 PostgreSQL
- 將舊有應用程式從 Oracle Pro*C 移轉至 ECPG
- 將現場部署工作負載遷移到 AWS 上的 Cloudera 資料平台
- 使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL
- 將 Oracle 商業智慧 12c 從現場部署伺服器遷移到 AWS 雲端
- 將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列
- 將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫
- 將 Oracle 電子商務套件遷移到 Amazon RDS 定制
- 將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容
- 使用擴充功能將甲骨文原生函數遷移至 PostgreSQL
- 將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制
- 將甲骨文功能遷移到 AWS 上的 PostgreSQL
- 將甲骨文序列遷移 _ 可重複使用的編譯包到 PostgreSQL
- 將 Redis 工作負載遷移到 AWS 上的 Redis 企業雲端
- 使用 AWS SCT 和 AWS DMS 將亞馬 Amazon EC2 上的 SAP ASE 遷移到與 Amazon Aurora PostgreSQL 相容
- 將虛擬生成的列從甲骨文遷移到 PostgreSQL
- 監控 Amazon ElastiCache 叢集以進行靜態加密
- 監控安全群組的 ElastiCache 叢集
- 使用應用程式遷移服務減少同質 SAP 移轉切換時間
- 輪換資料庫認證而不重新啟動
- 使用 AWS Fargate 大規模執行訊息導向工作負載
- 在 AWS 上設定高可用性 PeopleSoft 架構
- ???
- 在 Aurora 相容上設定甲骨文 UTL_FILE 功能
- 以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3
- 使用傳輸在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫
- 用 CloudEndure 於內部部署資料庫的嚴重損壞復原

- 從甲骨文遷移到 Amazon Aurora PostgreSQL 後驗證數據庫對象
- 確認新的 Amazon Redshift 叢集是否在 VPC 中啟動

DevOps

主題

- 自動化 AWS 資源評估
- 使用開放原始碼工具自動安裝 SAP 系統
- 使用 AWS CDK 自動化 AWS 服務目錄產品組合和產品部署
- 使用和事件將事件驅動的備份自動化 CodeCommit 到 Amazon S3 CodeBuild CloudWatch
- 使用 AWS CodePipeline 和 AWS 自動化堆疊集部署 CodeBuild
- 使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔
- 使用 AWS CDK 為微型服務自動建置 CI/CD 管道和 Amazon ECS 叢集
- 使用 DevOps 實務和 AWS Cloud9 建立鬆散結合的架構與微型服務
- 使用 GitHub 動作和地形表單建置碼頭映像並將其推送到 Amazon ECR
- 使用 AWS CodeCommit、AWS 和 AWS Device Farm 建置和測試 iOS 應用程式 CodePipeline
- 使用 cdk-nag 規則套件檢查 AWS CDK 應用程式或 CloudFormation 範本以取得最佳實務
- 設定對 Amazon DynamoDB 的跨帳戶存取權
- 為在 Amazon EKS 上執行的應用程式設定相互 TLS 身份驗證
- 使用 Firelens 日誌路由器為 Amazon ECS 創建自定義日誌解析器
- 使用和 HashiCorp 打包器創建管道 CodePipeline 和 AMI
- 使用建立管道並將成品更新部署到現場部署 EC2 執行個體 CodePipeline
- 自動為 Java 和 Python 項目創建動態 CI 管道
- 使用地形部署 CloudWatch Synthetics 金絲雀
- 在 Amazon ECS 上部署適用於 Java 微服務的 CI/CD 管道
- 使用 AWS CodeCommit 和 AWS 在多 CodePipeline 個 AWS 帳戶中部署 CI/CD 管道
- 使用 AWS Network Firewall 和 AWS Transit Gateway 部署防火牆
- 使用 AWS CodePipeline CI/CD 管道部署 AWS 膠合任務
- 使用 EC2 執行個體設定檔從 AWS Cloud9 部署 Amazon EKS 叢集
- 使用 AWS CodePipeline、AWS 和 AWS 在多個 AWS 區域部署程式碼 CodeCommit CodeBuild
- 將 AWS Organizations 中各個組織的 AWS Backup 報告匯出為 CSV 檔案
- 將 Amazon EC2 執行個體清單的標籤匯出為 CSV 檔案

- 使用對流圈產生包含 AWS 組態受管規則的 AWS CloudFormation 範本
- 讓 SageMaker 筆記本執行個體暫時存取另一個 AWS 帳戶中的 CodeCommit 儲存庫
- 為多帳戶環境實作 GitHub Flow 分支 DevOps 策略
- 為多帳戶環境實施 Gitflow 分支策略 DevOps
- 為多帳戶環境實作幹線分支 DevOps 策略
- 自動檢測更改並為中的壟斷啟動不同的 CodePipeline 管道 CodeCommit
- 使用 AWS 整合比特儲存庫與 AWS Amplify CloudFormation
- 使用 Step Functions 函數和 Lambda 代理函數在 AWS 帳戶之間啟動 CodeBuild 專案
- 使用 AWS 程式碼服務和 AWS KMS 多區域金鑰，管理對多個帳戶和區域的微型服務的藍/綠部署
- 使用 AWS CloudFormation 和 AWS Config 監控 Amazon ECR 儲存庫是否有萬用字元許可
- 從 AWS CodeCommit 事件執行自訂動作
- 將 Amazon CloudWatch 指標發佈到 CSV 檔案
- 使用最新的框架在 AWS Glue 中對 Python ETL 任務執行單元測試
- 在 Amazon S3 中設置頭盔 v3 圖表存儲庫
- 使用 AWS CodePipeline 和 AWS CDK 設定 CI/CD 管道
- 使用憑證管理員和讓我們 end-to-end 加密為 Amazon EKS 上的應用程式設定加密
- 使用 Flux 簡化 Amazon EKS 多租戶應用程式部署
- 使用自訂資源，將多個電子郵件端點訂閱至 SNS 主題
- 使用 Serverspec 進行基礎架構程式碼的測試驅動開發
- 在 AWS 中使用第三方 Git 來源儲存庫 CodePipeline
- 使用 AWS 建立 CI/CD 管道以驗證地形組態 CodePipeline
- 更多模式

自動化 AWS 資源評估

創建者：納文·薩塔爾 (AWS)、阿倫·巴格爾 (AWS)、馬尼什·加格 (AWS) 和桑迪普·加萬德 (AWS)

代碼存儲庫：infrastructure-assessment-iac-automation	環境：PoC 或試點	技術： DevOps；基礎架構；管理與治理；營運；無伺服器
AWS 服務：Amazon Athena；AWS CloudTrail；AWS Lambda；Amazon S3；Amazon QuickSight		

Summary

此模式描述使用 AWS Cloud Development Kit (AWS CDK) 設定資源評估功能的自動化方法。透過使用此模式，營運團隊會以自動方式收集資源稽核詳細資訊，並在單一儀表板上檢視 AWS 帳戶中部署的所有資源的詳細資料。這在下列使用案例中很有幫助：

- 識別基礎設施即程式碼 (IaC) 工具，並隔離由不同 I HashiCorp aC 解決方案 (例如地形、AWS、AWS CloudFormation CDK 和 AWS Command Line Interface (AWS CLI) (AWS CLI) 所建立的資源
- 擷取資源稽核資訊

此解決方案也可協助領導團隊從單一儀表板取得 AWS 帳戶中的資源和活動的深入解析。

注意：Amazon QuickSight 是一項付費服務。在執行資料分析和建立儀表板之前，請先檢閱 Amazon QuickSight 定價。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 具備佈建資源存取權的 AWS Identity and Access Management (IAM) 角色和許可

- 使用 Amazon 簡單存儲服務（Amazon S3）和亞馬遜雅典娜創建的亞馬遜 QuickSight 帳戶
- 已安裝 AWS CDK 版本 2.55.1 或更新版本
- 已 Python 裝 3.9 版或更新版本

限制

- 此解決方案部署到單一 AWS 帳戶。
- 除非 AWS CloudTrail 已設定並將資料存放在 S3 儲存貯體，否則解決方案不會追蹤部署之前發生的事件。

產品版本

- AWS CDK 版本 2.55.1 或更新版本
- 版 Python 3.9 或更高版本

架構

目標技術堆疊

- Amazon Athena
- AWS CloudTrail
- AWS Glue
- AWS Lambda
- Amazon QuickSight
- Amazon S3

目標架構

AWS CDK 程式碼會部署在 AWS 帳戶中設定資源評估功能所需的所有資源。下圖顯示將 CloudTrail 日誌傳送到 AWS Glue、Amazon Athena 和 QuickSight.

1. CloudTrail 將日誌發送到 S3 存儲桶以進行存儲。
2. 事件通知會叫用 Lambda 函數來處理記錄並產生篩選的資料。

3. 篩選後的資料會存放在另一個 S3 儲存貯體中。
4. 在 S3 儲存貯體中的篩選資料上設定 AWS Glue 爬蟲程式，以便在 AWS Glue 資料型錄表格中建立結構描述。
5. 篩選後的資料已準備就緒，供 Amazon Athena 查詢。
6. 查詢的資料由存取以進行 QuickSight 視覺化。

自動化和規模

- 如果 AWS 組織中有整個組織的 CloudTrail 追蹤，則可將此解決方案從一個 AWS 帳戶擴展到多個 AWS Organizations 帳戶。透過 CloudTrail 在組織層級部署，您也可以使用此解決方案擷取所有必要資源的資源稽核詳細資料。
- 此模式使用 AWS 無伺服器資源部署解決方案。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和 AWS 區域的整個生命週期進行管理。
- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。此模式使用 AWS Glue 爬行程式和 AWS Glue 資料型錄表格。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon QuickSight 是雲端規模商業智慧 (BI) 服務，可協助您在單一儀表板中視覺化、分析和報告資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

代碼存儲庫

此模式的代碼可在 GitHub infrastructure-assessment-iac-automation存儲庫中找到。

代碼存儲庫包含以下文件和文件夾：

- `lib`資料夾 — AWS CDK 會建構用來建立 AWS 資源的 Python 檔案
- `src/lambda_code`— 在 Lambda 函數中運行的 Python 代碼
- `requirements.txt`-必須安裝的所有 Python 依賴項的列表
- `cdk.json`— 用於提供旋轉資源所需的值的輸入文件

最佳實務

設定 Lambda 函數的監控和警示。如需詳細資訊，請參閱監控 Lambda 函數和疑難排解。如需使用 Lambda 函數時的一般最佳實務，請參閱 AWS 文件。

史诗

設定您的環境

任務	描述	所需技能
克隆本地計算機上的存儲庫。	若要複製儲存庫，請執行 `git clone https://github.com/aws-samples/infrastructure-assessment-iac-automation.git` 命令。	AWS DevOps、 DevOps 工程師
設定 Python 虛擬環境並安裝必要的相依性。	若要設定 Python 虛擬環境，請執行下列命令。 `cd infrastructure-assessment-iac-automation` `python3 -m venv .venv` `source .venv/bin/activate`	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	若要設定所需的相依性，請執行命令`pip install -r requirements.txt` 。	
設定 AWS CDK 環境並合成 AWS CDK 程式碼。	1. 若要在您的 AWS 帳戶中設定 AWS CDK 環境，請執行命令`cdk bootstrap aws://ACCOUNT-NUMBER/REGION` 。 2. 若要將程式碼轉換為 AWS CloudFormation 堆疊組態，請執行命令`cdk synth`。	AWS DevOps、 DevOps 工程師

在本機電腦上設定 AWS 登入資料

任務	描述	所需技能
匯出要部署堆疊的帳戶和區域的變數。	若要使用環境變數為 AWS CDK 提供 AWS 登入資料，請執行下列命令。 `export CDK_DEFAULT_ACCOUNT=<12 Digit AWS Account Number>` `export CDK_DEFAULT_REGION=<region>`	AWS DevOps、 DevOps 工程師
設定 AWS CLI 設定檔。	若要為帳戶設定 AWS CLI 設定檔，請遵循 AWS 文件中的指示。	AWS DevOps、 DevOps 工程師

設定和部署資源評估工具

任務	描述	所需技能
在帳戶中部署資源。	若要使用 AWS CDK 在 AWS 帳戶中部署資源，請執行下列動作： 1. 在複製存放庫的根目錄中，在cdk.json檔案中提供下列參數的輸入： • s3_context • ct_context • kms_context • lambda_context • glue_context • qs_context 這些值定義資源配置和命名法。預設值已設定，並可視需要變更。 注意：若要避免發生錯誤，指出 S3 儲存貯體已經存在，請務必在ct和output區段s3_context 中提供唯一的名稱。 2. 若要部署資源，請執行命令cdk deploy。 該cdk deploy命令會建立一個 CloudTrail 資源來記錄事件，並將日誌檔儲存在輸入 S3 儲存貯體中。Lambda 函數會處理追蹤的記錄檔。篩選後的結果會存放在輸	AWS DevOps

任務	描述	所需技能
	出 S3 儲存貯體中，並可供 Amazon Athena 和 Amazon 使用 QuickSight。	
執行 AWS Glue 爬行者程式並建立資料目錄表格。	AWS Glue 爬行程式是用來維持資料結構描述的動態。此解決方案會按照 AWS Glue 爬行程式排程器所定義，定期執行爬蟲程式，在 AWS Glue 資料型錄表格中建立和更新分割區。在輸出 S3 儲存貯體中可用資料之後，請使用下列步驟執行 AWS Glue 爬行者程式，並建立資料目錄表格結構描述以進行測試： 1. 登入 AWS 管理主控台並導覽至 AWS Glue 主控台。 2. 在導覽窗格的 [資料目錄] 下，選擇 [爬行者程式]。 3. 選取`iac-tool-qa-resource-iac-json-crawler` 爬行者程式。 4. 執行爬行者程式。 5. 爬行者程式成功執行之後，便會建立 AWS Glue 資料型錄表格。AWS QuickSight 將使用表格來視覺化資料。 注意：AWS CDK 程式碼會設定 AWS Glue 爬行程式在特定時間執行，但您也可以視需要執行。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
部署建 QuickSight 構。	1. 若要部署 QuickSight 建構，請取消註解#QuickSight setup - start 和 #QuickSight setup – ends in resource_iac_tool_stack.py 之間的程式碼。 2. 取消註釋後，運行命cdk deploy令以創建QuickSight DataSource 並QuickSight DataSet在 QuickSight 帳戶中。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
建立 QuickSight 儀表板。	欲建立範例 QuickSight 儀表板和分析，請執行下列操作： 1. 導覽至 QuickSight 主控台，然後選取要在其中部署資源的 AWS 區域。 2. 在導覽窗格中，選擇 [資料集]，然後驗證`ct-operations-iac-ds` 已在 Amazon 資料集中建立名為的 QuickSight 資料集。 如果您沒有看到資料集，請重新部署 QuickSight 建構。 3. 選取資`ct-operations-iac-ds` 料集，然後選擇「用於分析」。 4. 選取預設工作表。 5. 從左側的字段列表中選擇相應的列。 6. 選取所需欄之後，請選取適當的視覺類型以檢視資料。 如需詳細資訊，請參閱在 Amazon 中開始分析 QuickSight和 Amazon 中的視覺類型 QuickSight。	AWS DevOps、 DevOps 工程師

清理解決方案中的所有 AWS 資源

任務	描述	所需技能
移除 AWS 資源。	1. 若要移除解決方案部署的 AWS 資源，請執行命令`cdk destroy`。 2. 刪除兩個 S3 儲存貯體中的所有物件，然後移除值區。 如需詳細資訊，請參閱刪除值區。	AWS DevOps、 DevOps 工程師

在 AWS 資源評估工具自動化之上設定其他功能

任務	描述	所需技能
監控並清理手動建立的資源。	(選擇性) 如果您的組織有使用 IaC 工具建立資源的合規要求，您可以使用 AWS 資源評估工具自動化擷取手動佈建的資源，以達成合規。您也可以使用該工具將資源導入 IaC 工具或重新創建它們。若要監視手動佈建的資源，請執行下列高階工作： 1. 部署 AWS 資源評估工具自動化。 2. 設定 Lambda 函數，每天查詢 Athena 表格、尋找手動佈建資源的相關資料，並將其匯出至逗號分隔值 (CSV) 檔案。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	3. 執行 Lambda 函數之後，可以將包含所需資料的通知傳送給各自的利益相關者。 4. 如需更長的保留時間，.csv 檔案可以存放在 S3 儲存貯體中。 5. 根據 .csv 檔案中的資訊，刪除手動建立的資源，或將其匯入至現有的 IaC 解決方案。	

故障診斷

問題	解決方案
AWS CDK 會傳回錯誤。	如需 AWS CDK 問題的相關說明，請參閱疑難排解常見 AWS CDK 問題。

相關資源

- 使用 Python 構建函數
- 開始使用 AWS CDK
- 在 Python 中使用 AWS CDK
- 建立記 CloudTrail 錄追蹤
- 開始使用 Amazon QuickSight

其他資訊

多個帳戶

若要為多個帳戶設定 AWS CLI 登入資料，請使用 AWS 設定檔。如需詳細資訊，請參閱設定 AWS CLI 中的設定多個設定檔一節。

AWS CDK 命令

使用 AWS CDK 時，請記住下列有用的命令：

- 列出應用程式中的所有堆疊

```
cdk ls
```

- 發出合成的 AWS 範本 CloudFormation

```
cdk synth
```

- 將堆疊部署到您的預設 AWS 帳戶和區域

```
cdk deploy
```

- 將已部署的堆疊與目前狀態進行比較

```
cdk diff
```

- 開啟 AWS CDK 文件

```
cdk docs
```

使用開放原始碼工具自動安裝 SAP 系統

由吉列爾梅·塞斯特海姆（AWS）創建

代碼存儲庫：主存儲庫	環境：生產	技術： DevOps
工作負載：SAP	AWS 服務：Amazon EC2; Amazon S3	

Summary

此模式顯示如何使用開放原始碼工具建立下列資源，將 SAP 系統安裝自動化：

- 一個數據庫
- 一個 SAP ABAP 中央服務 (ASCS) 執行個體
- SAP 主要應用程式伺服器 (PAS) 執行處理

HashiCorp 地形創建 SAP 系統的基礎設施和 Ansible 配置操作系統（OS）和安裝 SAP 應用程序。詹金斯運行安裝。

此設定可將 SAP 系統安裝轉變為可重複的程序，有助於提高部署效率和品質。

注意：此模式中提供的範例程式碼適用於高可用性 (HA) 系統和非 HA 系統。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 包含所有 SAP 媒體檔案的亞馬遜簡易儲存服務 (Amazon S3) 儲存貯體
- 具有存取金鑰和秘密金鑰且具有下列許可的 AWS 身分與存取管理 (IAM) 主體：
  - 唯讀許可：Amazon Route 53，AWS Key Management Service (AWS KMS)
  - 讀取和寫入許可：Amazon S3，Amazon Elastic Compute Cloud (Amazon EC2)，Amazon Elastic File System（Amazon EFS），IAM，Amazon CloudWatch，亞馬遜 DynamoDB
- 一 Route 53 私人託管區域

- 在 Amazon Marketplace 場訂閱紅帽企業 Linux 適用於 SAP 的 HA 和更新服務 8.2 Amazon 機器映像（AMI）
- AWS KMS 客戶受管金鑰
- 安全殼層 (SSH) key pair
- 一個 Amazon EC2 安全群組，允許從安裝 Jenkins 的主機名稱在連接埠 22 上進行 SSH 連線 (主機名稱很可能是本機主機)
- 流浪者通過 HashiCorp 安裝和配置
- VirtualBox由甲骨文安裝和配置
- 熟悉 Git，地形，Ansible 和詹金斯

限制

- 只有 SAP S/4HANA 1909 已針對此特定案例進行完整測試。如果您使用另一個版本的 SAP HANA，則此模式中的 Ansible 程式碼範例需要修改。
- 此模式中的範例程序適用於 Mac 作業系統和 Linux 作業系統。某些指令只能在基於 UNIX 的終端中執行。但是，您可以通過使用不同的命令和 Windows 操作系統來實現類似的結果。

產品版本

- 汁液
- RHEL 企業版 (RHEL) 8.2 或更高版本

架構

下圖顯示使用開放原始碼工具自動化 AWS 帳戶中 SAP 系統安裝的範例工作流程：

該圖顯示以下工作流程：

1. 詹金斯通過運行地形和 Ansible 代碼協調運行 SAP 系統安裝。
2. 地形代碼構建 SAP 系統的基礎設施。
3. 可用的代碼可配置操作系統並安裝 SAP 應用程序。
4. 一個 SAP S/4HANA 1909 資料庫、ASCS 執行個體以及包含所有已定義先決條件的 PAS 執行個體都安裝在 Amazon EC2 執行個體上。

注意：此模式中的範例設定會在您的 AWS 帳戶中自動建立 Amazon S3 儲存貯體，以存放 Terraform 狀態檔案。

技術堆疊

- 地形
- Ansible
- Jenkins
- 一個數據庫
- 一個 SAP ASCS 執行個體
- 一個 SAP PAS 執行個體
- Amazon EC2

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以根據需要啟動任意數量的虛擬伺服器，並快速擴展或縮減它們。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以保護資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他工具

- HashiCorp Terraform 是一個命令列介面應用程式，可協助您使用程式碼來佈建和管理雲端基礎結構和資源。
- Ansible 是一種開源配置代碼（CAC）工具，可幫助自動化應用程序，配置和 IT 基礎架構。
- Jenkins 是一個開源的自動化服務器，使開發人員能夠構建，測試和部署他們的軟件。

Code

此模式的代碼可在 GitHub aws-install-sap-with-jenkin s-ansible 存儲庫中找到。

史诗

設定必要條件

任務	描述	所需技能
將您的 SAP 媒體文件添加到 Amazon S3 存儲桶。	建立包含所有 SAP 媒體檔案的 Amazon S3 儲存貯體。 重要：請確定您遵循 Launch Wizard 文件中 AWS 啟動精靈 S/4HANA 的資料夾階層。	雲端管理員
安裝 VirtualBox。	VirtualBox由甲骨文安裝和配置。	DevOps 工程師
安裝流浪漢。	安裝和配置流浪者 HashiCorp。	DevOps 工程師
設定您的 AWS 帳戶。	1. 確認您擁有具有存取金鑰和秘密金鑰的 IAM 主體，且具有下列權限： • 唯讀許可：Amazon Route 53，AWS Key Management Service (AWS KMS) • 讀取和寫入許可：Amazon S3，Amazon Elastic Compute Cloud (Amazon EC2)，Amazon Elastic File System（Amazon EFS），IAM，Amazon CloudWatch，亞馬遜 DynamoDB	一般 AWS

任務	描述	所需技能
	2. 儲存 IAM 主體的存取金鑰和秘密金鑰以供日後參考。 3. 如果您還沒有 Route 53 私人託管區域，請創建一個 Route 53 私人託管區域。儲存區域名稱 (例如，sapteam.net) 以供稍後參考。 4. 在 Amazon Marketplace 訂閱 RHEL 適用於 SAP 的 HA 和更新服務 8.2 AMI。儲存 AMI 識別碼 (例如，阿密 0 萬) 以供稍後參考。 5. 建立 AWS KMS 客戶受管金鑰。保存 KMS 密鑰的 Amazon 資源名稱（ARN）以供以後參考。 注意：以下是 AWS KMS 客戶受管金鑰 ARN 的範例：arn: AW: KMS: 美國東部 1:12341234: 金鑰 6. 建立 SSH key pair。保存密鑰對的名稱和 .pem 文件以供日後參考。 7. 建立一個 Amazon EC2 安全群組，該群組允許從安裝 Jenkins 的主機名稱在連接埠 22 上進行 SSH 連線。儲存安全群組 ID 以供日後參考。 注意：主機名稱最有可能是本地主機。	

建置並執行 SAP 安裝

任務	描述	所需技能
從複製程式碼儲存庫 GitHub。	克隆 aws-install-sap-with-詹金斯安裝的存儲庫。 GitHub	DevOps 工程師
啟動詹金斯服務。	開啟終端機。然後，瀏覽至包含複製程式碼儲存庫資料夾的本機資料夾，並執行下列命令： `sudo vagrant up` 注意：詹金斯啟動大約需要 20 分鐘。該命令返回服務已啟動並在成功時正在運行的消息。	DevOps 工程師
在網頁瀏覽器中打開詹金斯並登錄。	1. 在網頁瀏覽器中，輸入 http://localhost:5555。詹金斯打開。 2. 登錄到詹金斯通過使用管理員的用戶名和密碼我的秘密密碼通過密碼。	DevOps 工程師
設定 SAP 系統安裝參數。	1. 在詹金斯，選擇管理詹金斯。然後，選擇「管理認證」。您可以設定的認證變數清單隨即顯示。 2. 設定下列所有認證變數： • 對於 AWS_ACCOUNT_憑證，請輸入您的 IAM 主體的存取金鑰 ID 和秘密存取金鑰識別碼。 • 針對 AMI_ID，請輸入適用於 SAP 的紅帽企業 Linux，以	AWS 系統管理員、 DevOps 工程師

任務	描述	所需技能
	及更新服務 8.2 AMI 的 AMI 識別碼。 • 如果是 K MS 客戶受管金鑰的 ARN，請輸入您的 AWS KMS 客戶受管金鑰的 ARN。 • 對於 SSH_KEYAI R_NAME，請輸入安全殼層 key pair 的名稱，而不輸入 .pem 檔案類型。 • 對於 SSH_KEYAIR_FILE，請輸入金鑰組的 .pem 檔案的完整名稱 (例如，我的鍵盤 .pem)。確保您還將密鑰對的 .pem 文件上傳到詹金斯。 • 對於 S3_ROOT_ 資料夾 _ 安裝檔案，請輸入包含 SAP 媒體檔案的 Amazon S3 儲存貯體和資料夾的名稱 (如果適用的話) (例如，s3:///S4H 1909)。my-media-bucket • 針對私有 DNS_ZONE_ NAME，請輸入您的 Route 53 私人託管區域的名稱 (例如，我的公司網路)。 • 針對 VPC_ID，請輸入您要在其中建立 SAP 資源之 Amazon 虛擬私人雲端的虛擬私人雲端識別碼 (例如 vpc-12345)。 • 對於 SUBNET_IDS，如果您在測試環境中工作 (針對	

任務	描述	所需技能
	future HA 功能)，請輸入兩個公用子網路識別碼。如果您在生產環境中工作，最佳做法是將兩個私有子網路與防禦主機搭配使用。 • 對於安裝詹金斯的主機名稱，請輸入允許在連接埠 22 上進行 SSH 連線的 Amazon EC2 安全群組的識別碼。 附註：您可以根據您的使用案例，視需要設定其他非必要參數。例如，您可以變更 SAP 系統識別碼 (SID) 的執行個體、預設密碼、名稱和 SAP 系統的標籤。 所有必需的變量在其名稱的開頭都有（必需的）。	

任務	描述	所需技能
執行 SAP 系統安裝。	1. 在詹金斯，選擇詹金斯首頁。然後，選擇 SAP 解決方案 + ASCS + 步驟 3 實例。 2. 選擇旋轉並安裝。然後，選擇主要。 3. 選擇 [立即建置]。 如需管道步驟的相關資訊，請參閱 AWS 部落格上使用開放原始碼工具自動化 SAP 安裝的瞭解管道步驟一節。 注意：如果發生錯誤，請將游標移到出現的紅色錯誤方塊上，然後選擇記錄檔。出現錯誤之管線步驟的記錄檔。發生大多數錯誤的原因是參數設定不正確。	DevOps 工程師, AWS 系統管理員

相關資源

- DevOps 適用於 SAP — SAP 安裝：從 2 個月到 2 小時 (DevOps 企業高峰會影片庫)

使用 AWS CDK 自動化 AWS 服務目錄產品組合和產品部署

由桑迪蓋萬德 (AWS)、拉傑尼許泰亞吉 (AWS) 和維約瑪·薩克德瓦 (AWS) 所建立

代碼存儲庫：aws-cdk-servicecatalog-automation	環境：PoC 或試點	技術: DevOps; 基礎設施; 管理與治理
工作負載：開源	AWS 服務：AWS Service Catalog；AWS CDK	

Summary

AWS Service Catalog 可協助您集中管理已核准用於組織 AWS 環境的 IT 服務或產品目錄。產品集合稱為產品組合，而產品組合也包含組態資訊。使用 AWS Service Catalog，您可以為組織中的每種使用者類型建立自訂的產品組合，然後授與適當產品組合的存取權。然後，這些用戶可以從產品組合中快速部署他們需要的任何產品。

如果您有複雜的網路基礎結構，例如多區域和多帳戶架構，建議您在單一的集中帳戶中建立及管理 Service Catalog 產品組合。此模式說明如何使用 AWS Cloud Development Kit (AWS CDK) 在中央帳戶中自動建立 Service Catalog 產品組合、授予最終使用者存取權限，然後選擇性地在一或多個目標 AWS 帳戶中佈建產品。此 ready-to-use 解決方案會在來源帳戶中建立 Service Catalog 產品組合。此外，還可以選擇使用 AWS CloudFormation 堆疊在目標帳戶中佈建產品，並協助您 TagOptions 設定產品：

- AWS CloudFormation StackSets — 您可以使用 StackSets 跨多個 AWS 區域和帳戶啟動 Service Catalog 產品。在此解決方案中，您可以選擇在部署此解決方案時自動佈建產品。如需詳細資訊，請參閱使用 AWS CloudFormation StackSets (Service Catalog 文件) 和StackSets 概念 (CloudFormation 文件)。
- TagOption 程式庫 — 您可以使用物件 TagOption 庫來管理已佈建產品的標籤。A TagOption是 AWS Service Catalog 中管理的金鑰值組。它不是 AWS 標籤，但它可以做為基礎建立 AWS 標籤的範本 TagOption。如需詳細資訊，請參閱TagOption 程式庫 (Service Catalog 文件)。

先決條件和限制

先決條件

- 您想要用來作為管理 Service Catalog 產品組合的來源帳戶的有效 AWS 帳戶。
- 如果您使用此解決方案在一或多個目標帳戶中佈建產品，則目標帳戶必須已存在且處於作用中狀態。
- 用於存取 AWS 服務目錄、AWS 和 AWS IAM 的 AWS Identity and Access Management (IAM) 許可。 CloudFormation

產品版本

- AWS CDK 版本 2.27.0

架構

目標技術堆疊

- 集中式 AWS 帳戶中的 Service Catalog 產品組合
- 部署在目標帳戶中的 Service Catalog 產品

目標架構

1. 在產品組合 (或來源) 帳戶中，您可以使用 AWS 帳戶、AWS 區域、IAM 角色、產品組合和使用案例的產品資訊來更新 config.json 檔案。
2. 您可以部署 AWS CDK 應用程式。
3. AWS CDK 應用程式會擔任部署 IAM 角色，並建立 config.json 檔案中定義的 Service Catalog 產品組合和產品。

   如果您設定 StackSets 為在目標帳戶中部署產品，則程序會繼續進行。如果您未設定 StackSets 佈建任何產品，則程序已完成。
4. AWS CDK 應用程式擔任StackSet 管理員角色，並部署您在 config. json 檔案中定義的 AWS CloudFormation 堆疊集。
5. 在目標帳戶中， StackSets 承擔StackSet 執行角色並佈建產品。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CDK 工具組是命令列雲端開發套件，可協助您與 AWS CDK 應用程式互動。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Service Catalog 可協助您集中管理 AWS 核准的 IT 服務目錄。最終使用者可在機構所設的限制範圍內，迅速地只部署自己需要且經核准的 IT 服務。

代碼存儲庫

此模式的程式碼可在 GitHubaws-cdk-servicecatalog-automation存放庫中取得。代碼存儲庫包含以下文件和文件夾：

- cdk-sevicecatalog-app— 此資料夾包含此解決方案的 AWS CDK 應用程式。
- config — 此資料夾包含 config.json 檔案，以及用於在 Service Catalog 組合中部署產品的 CloudFormation 範本。
- 配置/config.json — 此文件包含所有配置信息。您可以更新此檔案，以針對您的使用案例自訂此解決方案。
- 配置/模板 — 此文件夾包含服務中心產品的 CloudFormation 模板。
- setup.sh — 此指令碼會部署解決方案。
- uninstall.sh — 此指令碼會刪除堆疊和部署此解決方案時建立的所有 AWS 資源。

若要使用範例程式碼，請依照 Epics 一節中的指示操作。

最佳實務

- 用於部署此解決方案的 IAM 角色應遵循最低權限 (IAM 文件) 的原則。
- 遵守使用 AWS CDK 開發雲端應用程式的最佳實務 (AWS 部落格文章)。
- 遵守 AWS 最 CloudFormation 佳實務 (CloudFormation 文件)。

史诗

設定您的環境

任務	描述	所需技能
安裝 AWS CDK 工具組。	確定您已安裝 AWS CDK 工具組。輸入以下命令以確認是否已安裝並檢查版本。 `cdk --version` 如果未安裝 AWS CDK 工具組，請輸入以下命令進行安裝。 `npm install -g aws-cdk@2.27.0` 如果 AWS CDK 工具組版本早於 2.27.0，則輸入下列命令將其更新為 2.27.0 版。 `npm install -g aws-cdk@2.27.0 --force`	AWS DevOps、 DevOps 工程師
複製儲存庫。	輸入以下命令。在 [其他資訊] 區段的 [複製存放庫] 中，您可以複製包含存放庫 URL 的完整指令。這將從 GitHub中克隆存aws-cdk-servicecatalog-automation儲庫。 `git clone <repository-URL>.git`	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	這將在目標目錄中創建一個cd aws-cdk-servicecatalog-automation 文件夾。輸入以下命令以導航到此文件夾。 `cd aws-cdk-servicecatalog-automation`	
設定 AWS 登入資料。	輸入下列命令：這些變數會匯出下列變數，這些變數定義了您要部署堆疊的 AWS 帳戶和區域。 `export CDK_DEFAULT_ACCOUNT=<12-digit AWS account number>` `export CDK_DEFAULT_REGION=<AWS Region>` AWS CDK 的 AWS 登入資料是透過環境變數提供的。	AWS DevOps、 DevOps 工程師
設定最終使用者 IAM 角色的許可。	如果您打算使用 IAM 角色來授與產品組合及其中產品的存取權，則這些角色必須具有由 servicecatalog.amazonaws.com 服務主體承擔的許可。如需有關如何授予這些許可的指示，請參閱使用 Service Catalog 啟用受信任存取 (AWS Organizations 文件)。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
設定所需的 IAM 角色 StackSets。	如果您使 StackSets 用在目標帳戶中自動佈建產品，則需要設定管理和執行堆疊集的 IAM 角色。 1. 在來源帳戶中，確認是否`AWSCloudFormationStackSetAdministrationRole` 已存在。在目標帳戶中，確認是否`AWSCloudFormationStackSetExecutionRole` 已存在。如果這些角色已經存在，您可以跳到下一個史詩。 2. 遵循授與自我管理許可 (IAM 文件) 中的指示，在產品組合帳戶中建立堆疊集管理角色，並在每個目標帳戶中建立執行角色。	AWS DevOps、 DevOps 工程師

自訂及部署解決方案

任務	描述	所需技能
建立 CloudFormation 範本。	在`config/templates` 資料夾中，為您要包含在產品組合中的任何產品建立 CloudFormation 範本。如需詳細資訊，請參閱使用 AWS CloudFormation 範本 (CloudFormation 文件)。	應用開發人員、AWS DevOps、 DevOps 工程師

任務	描述	所需技能
自訂組態檔案。	在config資料夾中，開啟 config.json 檔案，並根據您的使用案例定義參數。除非另有說明，否則需要以下參數： • 在此portfolios 段落中，定義下列參數以建立一或多個 Service Catalog 產品組合： • portfolioName -投資組合的名稱。 • providerName — 管理學檔之人員、專案團隊或組織的名稱。 • description -投資組合的簡要說明。 • roles— (選用) 應該可存取此產品組合的任何 IAM 角色的名稱。具有此角色的使用者可以存取此產品組合中的產品。 • users— (選用) 應該存取此產品組合及其產品的任何 IAM 使用者的名稱。 • groups— (選用) 任何應該可存取此產品組合及其產品的 IAM 使用者群組的名稱。 警告：IAM 使用者擁有長期登入資料，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在	AWS 應用程式開發人員、DevOps 工程師 DevOps

任務	描述	所需技能
	不再需要這些使用者時移除這些使用者。 重要事項：`rolesusers`、和`groups`都是可選參數，但如果您未定義其中一個參數，則沒有人可以在 Service Catalog 主控台中檢視產品組合產品。至少定義其中一個參數。如需詳細資訊，請參閱授與 Service Catalog 使用者的權限 (Service Catalog 文件)。 • (選擇性) 在`tagOption` 區段中， TagOptions 為產品定義： • `key`— TagOption 金鑰名稱 • `value`— 允許的字串值 TagOption 如需詳細資訊，請參閱TagOption 程式庫 (Service Catalog 文件)。 • 在此**`products`**區段中，定義產品的下列參數： • `portfolioName` — 您要新增產品的產品組合名稱。您只能指定一個投資組合。 • `productName` — 產品名稱。 • `owner`— 產品的所有者。	

任務	描述	所需技能
	• productVersionName — 以字串值表示的產品版本名稱，例如v1。 • templatePath — 產品 CloudFormation 範本的檔案路徑。 • deployWithStackSets — (選擇性) 指定一或多個您要用 StackSets 於在產品組合中自動佈建產品的帳戶和區域。如果您使用此部署選項，則需要此段落中的所有下列參數： • accounts— 目標帳戶。 • regions— 目標區域。 • stackSetAdministrationRoleName — 用來管理 StackSets 組態的 IAM 角色名稱。請不要變更此值。此角色必須具有此完全相同的名稱。 • stackSetExecutionRoleName — 部署堆疊執行個體之目標帳戶中 IAM 角色的名稱。請不要變更此值。此角色必須具有此完全相同的名稱。	

任務	描述	所需技能
	如需已完成組態檔案的範例，請參閱其他資訊一節中的範例組態檔案。	
部署解決方案。	輸入以下命令。這會部署 AWS CDK 應用程式，並依照 config.json 檔案中指定的方式佈建 Service Catalog 產品組合和產品。 `sh +x setup.sh`	AWS 應用程式開發人員、DevOps 工程師 DevOps

任務	描述	所需技能
驗證部署。	請執行下列動作，確認部署成功： 1. 使用可存取您在設定檔中定義的一或多個產品組合的登入資料登入 AWS 管理主控台。 2. 在 https://console.aws.amazon.com/servicecatalog/ 開啟 Service Catalog 主控台。 3. 在功能窗格的 [佈建] 下，選擇 [產品]。確認您看到您為產品組合指定的產品清單。 4. 遵循啟動產品 (Service Catalog 文件) 中的指示，啟動其中一個可用產品。確認可用的產品版本和標籤符合您在組態檔案中提供的值。 5. 如果您選擇使用在一或多個目標帳戶中自動佈建產品 StackSets，請執行下列動作： a. 使用認證登入，授與您檢視其中一個目標帳戶中佈建產品的權限。 b. 在 Service Catalog 主控台的導覽窗格的佈建下，選擇已佈建的產品。 c. 確認預期的產品出現在清單中。	一般 AWS

任務	描述	所需技能
（可選）更新產品組合和產品。	如果您想要使用此解決方案來更新產品組合或產品，或提供新產品： 1. 在 config.json 檔案中進行必要的變更。 2. 視需要在`config/template` 資料夾中加入或修改任何 CloudFormation 樣板。 3. 重新部署解決方案。 例如，您可以新增其他產品組合或佈建更多資源。AWS CDK 應用程式只會實作變更。如果先前部署的產品組合或產品沒有任何變更，則重新部署不會影響它們。	一般 AWS 應用 DevOps 程式開發人員、工程師

清理解決方案

任務	描述	所需技能
(選擇性) 移除此解決方案部署的 AWS 資源。	如果您想要刪除已佈建的產品，請遵循刪除佈建的產品 (Service Catalog 文件) 中的指示進行。 如果您要刪除此解決方案所建立的所有資源，請輸入下列命令。 `sh uninstall.sh`	AWS DevOps、 DevOps 工程師、應用程式開發者

相關資源

- AWS Service Catalog 建構程式庫 (AWS API 參考)
- StackSets 概念 (CloudFormation 文件)
- AWS Service Catalog (AWS 行銷)
- 搭配 AWS CDK 使用 Service Catalog (AWS 研討會)

其他資訊

其他資訊

克隆存儲庫

輸入下列指令以從中複製儲存庫 GitHub。

```
git clone https://github.com/aws-samples/aws-cdk-servicecatalog-automation.git
```

範例設定檔

以下是含有範例值的範例 config.json 檔案。

```
{
    "portfolios": [
        {
            "displayName": "EC2 Product Portfolio",
            "providerName": "User1",
            "description": "Test1",
            "roles": [
                "<Names of IAM roles that can access the products>"
            ],
            "users": [
                "<Names of IAM users who can access the products>"
            ],
            "groups": [
                "<Names of IAM user groups that can access the products>"
            ]
        },
        {
            "displayName": "Autoscaling Product Portfolio",
            "providerName": "User2",
            "description": "Test2",
```

```
            "roles": [
                "<Name of IAM role>"
            ]
        }
    ],
    "tagOption": [
        {
            "key": "Group",
            "value": [
                "finance",
                "engineering",
                "marketing",
                "research"
            ]
        },
        {
            "key": "CostCenter",
            "value": [
                "01",
                "02",
                "03",
                "04"
            ]
        },
        {
            "key": "Environment",
            "value": [
                "dev",
                "prod",
                "stage"
            ]
        }
    ],
    "products": [
        {
            "portfolioName": "EC2 Product Profile",
            "productName": "Ec2",
            "owner": "owner1",
            "productVersionName": "v1",
            "templatePath": "../../config/templates/template1.json"
        },
        {
            "portfolioName": "Autoscaling Product Profile",
            "productName": "autoscaling",
```

```
            "owner": "owner1",
            "productVersionName": "v1",
            "templatePath": "../../config/templates/template2.json",
            "deployWithStackSets": {
                "accounts": [
                    "012345678901",
                ],
                "regions": [
                    "us-west-2"
                ],
                "stackSetAdministrationRoleName":
 "AWSCloudFormationStackSetAdministrationRole",
                "stackSetExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"
            }
        }
    ]
}
```

使用和事件將事件驅動的備份自動化 CodeCommit 到 Amazon S3 CodeBuild CloudWatch

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：生產	技術： DevOps；儲存與備份	工作負載：所有其他工作
AWS 服務：Amazon S3; Amazon CloudWatch; AWS CodeBuild; AWS CodeCommit		

Summary

在 Amazon Web Services (AWS) 雲端上，您可以使用 AWS CodeCommit 託管安全的 Git 儲存庫。CodeCommit 是完全受管的原始檔控制服務。但是，如果意外刪除了 CodeCommit 存放庫，其內容也會被刪除且無法還原。

此模式說明如何在對儲存 CodeCommit 庫進行變更後，自動將存放庫備份到 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體。如果稍後刪除 CodeCommit 儲存區域，此備份策略會提供 point-in-time 復原選項。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 根據您的需求配置了用戶訪問的現有 CodeCommit 存儲庫。如需詳細資訊，請參閱文件 CodeCommit中的設定 CodeCommit AWS。
- 用於上傳 CodeCommit 備份的 S3 儲存貯體。

限制

- 此模式會自動備份您的所有存 CodeCommit 儲庫。如果要備份個別 CodeCommit 儲存庫，則必須修改 Amazon CloudWatch 活動規則。

架構

下圖說明此模式的工作流程。

工作流程由以下步驟組成：

1. 代碼被推送到一個 CodeCommit 存儲庫。
2. CodeCommit 存放庫會通知有關存放庫變更的 CloudWatch 事件 (例如，`git push`指令)。
3. CloudWatch 事件會叫用 AWS， CodeBuild 並將 CodeCommit 儲存庫資訊傳送給它。
4. CodeBuild 克隆整個 CodeCommit 存儲庫並將其打包到 .zip 文件中。
5. CodeBuild 將 .zip 檔案上傳到 S3 儲存貯體。

技術, 堆

- CloudWatch 活動
- CodeBuild
- CodeCommit
- Amazon S3

工具

- Amazon CloudWatch 活動 — CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS CodeBuild — CodeBuild 是全受管持續整合服務，可編譯原始程式碼、執行測試，以及產生可立即部署的軟體套件。
- AWS CodeCommit — CodeCommit 是一種完全受管的原始檔控制服務，可託管安全的 Git 儲存庫。
- AWS Identity and Access Management (IAM) — IAM 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。

史诗

創建一個 CodeBuild 項目

任務	描述	所需技能
建立 CodeBuild 服務角色。	登入 AWS 管理主控台，並開啟 IAM 主控台。選擇角色，然後選擇建立角色。建立服務角色 CodeBuild 以複製 CodeCommit 存放庫、將檔案上傳到 S3 儲存貯體，然後將日誌傳送到 Amazon CloudWatch。如需詳細資訊，請參閱 CodeBuild 文件中的建立 CodeBuild 服務角色。	雲端管理員
創建一個 CodeBuild 項目。	在 CodeBuild 主控台上，選擇 [建立 CodeBuild 專案]。使用 [其他資訊] 區段中的`buildspec.yml` 範本建立 CodeBuild 專案。如需此故事的說明，請參閱 CodeBuild 文件中的建立組建專案。	雲端管理員

建立和設定 CloudWatch 事件規則

任務	描述	所需技能
為 CloudWatch 事件建立 IAM 角色。	在 IAM 主控台上，選擇「角色」，然後為 CloudWatch 事件建立 IAM 角色。如需有關這方面的詳細資訊，請參閱 IAM 文件中的CloudWatch 活動 IAM 角色。	雲端管理員

任務	描述	所需技能
	重要：您必須為 CloudWatch 事件的 IAM 角色新增`codebuild:StartBuild` 許可。	

任務	描述	所需技能
建立 CloudWatch 事件規則。	1. 在 CloudWatch 主控台上，選擇 [事件]，然後選擇 [規則]。選擇 [建立規則]，然後使用 [其他資訊] 區段中的 [CloudWatch 事件] 規則。這會建立監聽 CodeCommit 儲存庫中事件變更 (例如，`git push`或`git commit`指令) 的規則。如需詳細資訊，請參閱 AWS CodePipeline 文件中的為 CodeCommit 來源建立 CloudWatch 事件規則。 2. 選擇 [目標]，選擇 [主題]，然後選擇 [設定輸入]。選擇 [輸入變壓器]，然後使用 [其他資訊] 區段中的輸入路徑和輸入範本。這樣可以確保您的 CodeCommit 存儲庫詳細信息被解析並作為環境變量發送到 CodeBuild 項目。若要取得更多資訊，請參閱 CloudWatch 文件中的輸入轉換器自學課程。 3. 選擇設定詳細資料，然後輸入規則的名稱和說明。選擇 Create rule (建立規則)。 重要事項：此 CloudWatch 事件規則說明所有 CodeCommit 儲存庫中的變更。如果您想要備份個別 CodeCommit 存放庫或針對不同的存放庫備份使用	雲端管理員

任務	描述	所需技能
	不同的 S3 儲存貯體，則必須修改 CloudWatch 事件規則。	

相關資源

建立 CodeBuild 專案

- 建立 CodeBuild 服務角色
- 創建一個 CodeBuild 項目
- Git 客戶端命令所需的權限

建立和設定 CloudWatch 事件規則

- 為 CodeCommit 來源建立 CloudWatch 事件規則
- 使用輸入轉換器自訂傳遞至事件目標的內容
- 建立在 CloudWatch 事件上起始的事件規則
- 建立 CloudWatch 活動 IAM 角色

其他資訊

CodeBuild 構建規格的 .yml 模板

```
version: 0.2
phases:
  install:
    commands:
      - pip install git-remote-codecommit
  build:
    commands:
       - env
       - git clone -b $REFERENCE_NAME codecommit::$REPO_REGION://$REPOSITORY_NAME
       - dt=$(date '+%d-%m-%Y-%H:%M:%S');
       - echo "$dt"
       - zip -yr $dt-$REPOSITORY_NAME-backup.zip ./
```

```
      - aws s3 cp $dt-$REPOSITORY_NAME-backup.zip s3:// #substitute a valid S3 Bucket
 Name here
```

CloudWatch 事件規則

```
{
  "source": [
    "aws.codecommit"
  ],
  "detail-type": [
    "CodeCommit Repository State Change"
  ],
  "detail": {
    "event": [
      "referenceCreated",
      "referenceUpdated"
    ]
  }
}
```

CloudWatch 事件規則目標的輸入轉換器範例

輸入路徑：

```
{"referenceType":"$.detail.referenceType","region":"$.region","repositoryName":"$.detail.reposi
```

輸入範本 (請填寫適當的數值):

```
{
    "environmentVariablesOverride": [
        {
            "name": "REFERENCE_NAME",
            "value": ""
        },
        {
            "name": "REFERENCE_TYPE",
            "value": ""
        },
        {
            "name": "REPOSITORY_NAME",
            "value": ""
        },
```

```
        {
            "name": "REPO_REGION",
            "value": ""
        },
        {
            "name": "ACCOUNT_ID",
            "value": ""
        }
    ]
}
```

使用 AWS CodePipeline 和 AWS 自動化堆疊集部署 CodeBuild

創建者：蒂亞加拉揚瑪尼 (AWS)、米希爾博卡 (AWS) 和拉格戈達 (AWS)

代碼存儲庫：automated-code-pipeline-stackset-部署	環境：生產	技術： DevOps; 軟件開發和測試
AWS 服務：AWS CodeBuild ；AWS ；AWS CodeCommit ；AWS CodePipeline ；AWS Organizations ；AWS CloudFormation		

Summary

在持續整合和持續交付 (CI/CD) 程序中，您可能想要將應用程式自動部署到所有現有 AWS 帳戶，以及在 AWS Organization 中新增至組織的新帳戶。當您針對此需求架構 CI/CD 解決方案時，AWS CloudFormation 委派的堆疊集管理員功能非常有用，因為它會限制對管理帳戶的存取，以提供一層安全性。不過，AWS CodePipeline 使用服務管理許可模型，將應用程式部署到多個帳戶和區域。您必須使用 AWS Organizations 管理帳戶來部署堆疊集，因為 AWS CodePipeline 不支援委派的堆疊集管理員功能。

此模式描述了如何解決此限制。該模式使用 AWS CodeBuild 和自訂指令碼，透過 AWS 自動化堆疊集部署 CodePipeline。它會自動執行下列應用程式部署活動：

- 將應用程式作為堆疊集部署到現有的組織單位 (OU)
- 將應用程式的部署延伸至其他 OU 和區域
- 從所有或特定 OU 或區域移除已部署的應用程式

先決條件和限制

先決條件

在您遵循此模式中的步驟之前：

- 在您的 AWS Organizations 管理帳戶中建立組織。如需指示，請參閱 AWS Organizations 文件。

- 啟用 AWS Organizations 之間的受信任存取，並 CloudFormation 使用服務管理許可。如需指示，請參閱 CloudFormation 文件中的啟用 AWS Organizations 的受信任存取。

限制

此模式提供的程式碼有下列限制：

- 您只能為應用程式部署單一 CloudFormation 範本；目前不支援多個範本部署。
- 自訂目前的實作需要 DevOps 專業知識。
- 此模式不使用 AWS 金鑰管理系統 (AWS KMS) 金鑰。但是，您可以透過重新配置此模式所包含的 CloudFormation 範本來啟用此功能。

架構

CI/CD 部署管線的此架構可處理下列項目：

- 將堆疊集部署責任委派給專用的 CI/CD 帳戶，做為應用程式部署的堆疊集管理員，以限制直接存取管理帳戶。
- 每當在 OU 下建立並對應新帳戶時，都會使用服務管理的權限模型自動部署應用程式。
- 確保環境層級所有帳戶的應用程式版本一致性。
- 在儲存庫和管道層級使用多個核准階段，為已部署的應用程式提供額外的安全性和控管層級。
- 在 CodeBuild 中使用自訂建置的部署指令碼自動部署或移除堆疊集和堆疊執行個體，克服目前的 CodePipeline 限制。如需自訂指令碼所實作之流程控制與 API 呼叫階層的圖解，請參閱其他資訊一節。
- 為開發、測試和生產環境建立個別的堆疊集。此外，您還可以建立在每個階段結合多個 OU 和區域的堆疊集。例如，您可以在開發部署階段中結合沙箱和開發 OU。
- 支援在帳戶子集或 OU 清單中部署應用程式，或從中排除應用程式。

自動化和規模

您可以使用此模式提供的程式碼為應用程式建立 AWS CodeCommit 儲存庫和程式碼管道。然後，您可以將這些作為堆疊集部署到 OU 層級的多個帳戶。此程式碼也會自動化像是 Amazon Simple Notification Service (Amazon SNS) 主題等元件，以通知核准者、必要的 AWS Identity and Access Management (IAM) 角色，以及要套用在管理帳戶中的服務控制政策 (SCP)。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 將部署 CodeDeploy自動化到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。

代碼存儲庫

此模式的代碼可在 GitHub automated-code-pipeline-stackset-deployment 存儲庫中找到。如需資料夾結構和其他詳細資訊，請參閱存放庫的 Readme 檔案。

最佳實務

在 OU 層級部署應用程式時，此模式會限制直接存取管理帳戶。在管線和儲存庫程序中新增多個核准階段，有助於為您使用此方法部署的應用程式和元件提供額外的安全性和控管。

史诗

在 AWS Organizations 中設定帳戶

任務	描述	所需技能
啟用管理帳戶中的所有功能。	遵循 AWS Organizations 文件中的指示，為您的組織啟用管理帳戶中的所有功能。	AWS 管理員、平台管理員
建立一個 CI/CD 帳戶。	在 AWS Organizations 中，在您的組織中建立專用的 CI/CD 帳戶，並指派團隊來擁有和控制帳戶的存取權限。	AWS 管理員
新增委派的管理員。	在管理帳戶中，將您在上一個步驟中建立的 CI/CD 帳戶註冊為委派的堆疊集管理員。如需指示，請參閱 AWS CloudFormation 文件。	AWS 管理員、平台管理員

建立應用程式儲存庫和 CI/CD 管線

任務	描述	所需技能
克隆代碼存儲庫。	1. 將此模式提供的程式碼儲存庫複製到您的電腦上： `git clone https://github.com/aws-samples/automated-code-pipeline-stackset-deployment.git` 2. 檢閱 Readme 檔案以瞭解目錄結構和其他詳細資料。	AWS DevOps

任務	描述	所需技能
建立 SNS 主題。	您可以使用 GitHub 存放庫中提供的`sns-template.yaml` 範本來建立 SNS 主題並設定核准請求的訂閱。 1. 在 AWS 主控台上，登入 CI/CD 帳戶。 2. 請在以下位置開啟 CloudFormation 主控台：https://console.aws.amazon.com/cloudformation。 3. 使用新資源創建一個新堆棧（標準選項）。 4. 在「指定範本」中，選擇「上傳範本檔案」、「選擇檔案」，然後從複製 GitHub 儲存庫的`templates` 資料夾中選取`sns-template.yaml` 檔案。選擇下一步。 5. 提供有意義的應用程式堆疊名稱。 6. 指定資源的前置詞。 7. 選擇「下一步」、「下一步」和「提交 8. 成功建立堆疊後，請選擇「輸出」索引標籤，並記下提取請求、測試環境和生產環境 SNS 主題的 Amazon 資源名稱 (ARN)。您將在後續步驟中使用此資訊。	AWS DevOps

任務	描述	所需技能
為 CI/CD 元件建立 IAM 角色。	您可以使用 GitHub 存放庫中提供的`cicd-role-template.yaml` 範本來建立 CI/CD 元件所需的 IAM 角色和政策。 1. 在 AWS 主控台上，登入 CI/CD 帳戶。 2. 請在以下位置開啟 CloudFormation 主控台：https://console.aws.amazon.com/cloudformation。 3. 使用新資源創建一個新堆棧（標準選項）。 4. 在「指定範本」中，選擇「上傳範本檔案」、「選擇檔案」，然後從複製 GitHub 儲存庫的`templates` 資料夾中選取`cicd-role-template.yaml` 檔案。選擇下一步。 5. 提供有意義的應用程式堆疊名稱。 6. 輸入下列參數的值： • 權限邊界原則的 ARN。您可以從 IAM 主控台上權限界限政策的 [政策詳細資料] 區段取得此 ARN。 • 您先前記下之 SNS 生產核准主題的 ARN。 • 您先前記下之 SNS 測試核准主題的 ARN。 • 範本所建立之資源的前置詞。	AWS DevOps

任務	描述	所需技能
	7. 選擇「下一步」、「下一步」和「提交 8. 成功建立堆疊後，請選擇 [輸出] 索引標籤，並記下已建立的 IAM 角色的 ARN。您將在後續步驟中使用此資訊。	

任務	描述	所需技能
為您的應用程式建立 CodeCommit 儲存庫和程式碼管道。	您可以使用 GitHub 存放庫中提供的`cicd-pipeline-template.yaml` 範本，為應用程式建立 CodeCommit 存放庫和程式碼管道。 1. 在 AWS 主控台上，登入 CI/CD 帳戶。 2. 請在以下位置開啟 CloudFormation 主控台：https://console.aws.amazon.com/cloudformation。 3. 使用新資源創建一個新堆棧（標準選項）。 4. 在「指定範本」中，選擇「上傳範本檔案」、「選擇檔案」，然後從複製 GitHub 儲存庫的`templates` 資料夾中選取`cicd-pipeline-template.yaml` 檔案。選擇下一步。 5. 提供有意義的應用程式堆疊名稱。 6. 輸入下列參數的值： • AppRepositoryName— 將為應用程式建立的 CodeCommit 存放庫名稱。 • AppRepositoryDescription— 將針對應用程式建立的 CodeCommit 存放庫的簡短描述。	AWS DevOps

任務	描述	所需技能
	• ApplicationName— 您的應用程序的名稱。此字串用作 CodeCommit 儲存庫的名稱和 CI/CD 管線的前置詞。 • CloudWatchEventRoleARN — 先前工作之 CloudWatch 事件角色的 ARN。 • CodeBuildProjectRoleARN — 先前作業中 CodeBuild 專案角色的 ARN。 • CodePipelineRoleARN — 先前作業中 CodePipeline 角色的 ARN。 • DeploymentConfigBucket— 將存放部署組態檔案和指令碼 .zip 檔案的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體名稱。 • DeploymentConfigKey-路徑和 .zip 文件名（Amazon S3 密鑰）。 • 提取要求通知 — 提取要求通知之 SNS 主題的 ARN。 • ProdApprovalSNSARN — 適用於生產核准之 SNS 主題的 ARN。 • 測試核准權限 — 測試核准之 SNS 主題的 ARN。	

任務	描述	所需技能
	• TemplateBucket— CI/CD 帳戶中將存放 CI/CD 管線建立範本的 S3 儲存貯體名稱。 7. 選擇「下一步」、「下一步」和「提交 8. 當堆疊順利完成時，會建立具有指定名稱和預設目錄結構的 CodeCommit 儲存庫、部署組態檔、指令碼，以及儲存庫的程式碼管線。	

部署堆疊集

任務	描述	所需技能
複製應用程式存放庫。	您先前使用的 CI/CD 管線範本會建立範例應用程式存放庫和程式碼管線。若要複製並驗證儲存庫： 1. 登入 CI/CD 帳戶。 2. 找到您在上一個史詩中創建的應用程序存儲庫和 CI/CD 管道。 3. 複製儲存庫的 URL，並使用 git clone 指令在本機電腦上複製儲存庫。 4. 確認目錄結構和檔案是否符合下列項目： `root` `  |- deploy_configs`	應用程式開發人員、資料

任務	描述	所需技能
	<pre>| |- deployment_config.json |- parameters | |- template-parameter-dev.json | |- template-parameter-test.json | |- template-parameter-prod.json |- templates | |- template.yml |- buildspec.yml</pre> 其中資料deploy_configs 夾包含部署設定檔，templates 和parameters 資料夾包含預設檔案，您將以自己的 CloudFormation 範本和參數檔取代這些檔案。 重要：請勿自訂資料夾結構。 5. 建立特徵分支。	

任務	描述	所需技能
新增應用程式成品。	使用 CloudFormation 範本更新應用程式儲存庫。 注意：此解決方案僅支持單個 CloudFormation 模板的部署。 1. 建立用於部署應用程式程式碼變更的 CloudFormation 範本，並將其命名`<application-name>.yaml` 。 2. 將應用程式儲存庫`templates` 資料夾中的`template.yml` 檔案取代為您在步驟 1 中建立的 CloudFormation 範本。 3. 為每個環境 (開發、測試和生產) 準備參數檔案。 4. 使用格式命名參數檔案`<cloudformation-template-name>-parameter-<environment-name>.json` 。 5. 將`parameters` 資料夾中的預設參數檔案取代為步驟 4 中的檔案。	應用程式開發人員、資料

任務	描述	所需技能
更新部署規劃檔。	更新deployment_config.json 檔案： 1. 在應用程式存放庫中，導覽至資deploy_configs 料夾。 2. 開啟檔案deploymen t_config.json ： { "deployme nt_action": "<deploy/delete>", "stack_set_name": "<stack set name>", "stack_se t_desciption": "<stack set descripti on>", "deployme nt_targets": { "dev": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"], "filter_type":	應用程式開發人員、資料

任務	描述	所需技能
	"<DIFFERENCE/INTERSECTION/UNION>" }, "test": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"], "filter_type": "<DIFFERENCE/INTERSECTION/UNION>" }, "prod": { "org_units": ["list of OUs"], "regions": ["list of regions"], "filter_accounts": ["list of accounts"],	

任務	描述	所需技能
	<pre>"filter_type": "<DIFFERENCE/INTERSECTION/UNION>" } }, "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"], "auto_deployement": "<True/False>", "retain_stacks_on_account_removal": "<True/False>", "region_deployment_concurrency": "<SEQUENTIAL/PARALLEL>" }</pre> 3. 更新部署動作、堆疊集名稱、堆疊集說明及部署目標的值。 例如，您可以設定`deployment_action delete`為刪除整個堆疊集及其關聯的堆疊執行個體。用`deploy`來建立新的堆疊集、更新現有的堆疊集，或新增或移除其他 OU 或區域的堆疊執行	

任務	描述	所需技能
	個體。如需更多範例，請參閱其他資訊一節。 此模式會將環境名稱新增至您在部署組態檔中提供的堆疊集名稱，以便為每個環境建立個別的堆疊集。	

任務	描述	所需技能
提交更改並部署堆棧集。	提交您在應用程式範本中指定的變更，並逐階段將堆疊集合併並部署到多個環境中： 1. 保存所有文件並將更改提交到本地應用程序存儲庫的功能分支。 2. 將功能分支推送到遠程存儲庫。 3. 創建一個提取請求以將更改合併到主分支。 當提取請求獲得核准且變更已合併到主分支時，將啟動 CI/CD 管線。 4. 成功完成開發部署階段後，請檢查 CloudFormation 主控台「服務管理」索引標籤。StackSets 您將看到一個帶有後綴的新堆棧集`dev`。 5. 檢查開發部署階段的 CodeBuild 記錄檔是否有任何問題。 6. 請核准者核准這些階段的部署，並重複步驟 5 和 6，將堆疊集部署到測試和生產環境中。用於測試和生產環境的堆棧集具有後綴`test`和`prod`	應用程式開發人員、資料

故障診斷

問題	解決方案
部署失敗，但發生例外狀況： 將模板參數文件的名稱更改為-parameter-.json 使用，默認名稱不允許 <application name><evn >	CloudFormation 範本參數檔案必須遵循指定的命名慣例。請更新參數檔案名稱，然後再試一次。
部署失敗，但發生例外狀況： 將 CloudFormation 模板的名稱更改為 .yml，默認模板 .yml 或模板 .yaml 是不允許的 <application name>	CloudFormation 範本名稱必須遵循指定的命名慣例。請更新檔案名稱，然後再試一次。
部署失敗，但發生例外狀況： 找不到 {環境名稱} 環境的有效 CloudFormation 範本及其參數檔	檢查 CloudFormation 樣板的檔案命名慣例及其指定環境的參數檔案。
部署失敗，但發生例外狀況： 部署設定檔中提供的部署動作無效。有效的選項是「部署」和「刪除」。	您在部署規劃檔中為`deployment_action` 參數指定了無效的值。參數有兩個有效值：`deploy`和`delete`。用`deploy`於建立和更新堆疊集及其關聯的堆疊執行個體。`delete`只有當您想要移除整個堆疊集和關聯的堆疊執行個體時才使用。

相關資源

- GitHub automated-code-pipeline-stackset-部署存儲庫
- 啟用組織中的所有功能 (AWS Organizations 文件)
- 註冊委派的管理員 (AWS CloudFormation 文件)
- 服務管理堆疊集的帳戶層級目標 (AWS CloudFormation 文件)

其他資訊

流程圖

下列流程圖說明由自訂指令碼實作以自動化堆疊集部署的流程控制和 API 呼叫階層。

範例部署規劃檔

創建一個新的堆棧集

下列部署設定檔會在三個 OU 中建立`sample-stack-set`在 AWS 區域`us-east-1`中呼叫的新堆疊集。

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                          },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
```

```
 }
```

將現有的堆疊集部署到另一個 OU

如果您部署上一個範例中顯示的組態，並且想要將堆疊集部署到開發環境dev-org-unit-2中呼叫的其他 OU，則部署組態檔案可能如下所示。

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                              "dev": {
                                          "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                          "regions": ["us-east-1"],
                                          "filter_accounts": [],
                                          "filter_type": ""
                                      },
                              "test": {
                                          "org_units": ["test-org-unit-1"],
                                          "regions": ["us-east-1"],
                                          "filter_accounts": [],
                                          "filter_type": ""
                                      },
                              "prod": {
                                          "org_units": ["prod-org-unit-1"],
                                          "regions": ["us-east-1"],
                                          "filter_accounts": [],
                                          "filter_type": ""
                                      }
                            },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

將現有堆疊集部署到另一個 AWS 區域

如果您部署上個範例中顯示的組態，並且想要針對兩個 OU (和us-east-2) 將堆疊集部署到開發環境中的其他 AWS 區域 (dev-org-unit-1dev-org-unit-2)，則部署組態檔可能如下所示。

附註： CloudFormation 範本中的資源必須是有效且特定於區域的資源。

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                              "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-1", "us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                              "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                              "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                            },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

從 OU 或 AWS 區域移除堆疊執行個體

假設上一個範例中顯示的部署組態已經部署。下列設定檔會從 OU 的兩個區域移除堆疊執行個體dev-org-unit-2。

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
```

```
                            "dev": {
                                        "org_units": ["dev-org-unit-1"],
                                        "regions": ["us-east-1", "us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                           },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
 }
```

下列組態檔會us-east-1針對開發環境中的兩個 OU 從 AWS 區域移除堆疊執行個體。

```
{
     "deployment_action": "deploy",
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
```

```
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                                "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                            },
     "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
     "auto_deployement": "True",
     "retain_stacks_on_account_removal": "True",
     "region_deployment_concurrency": "PARALLEL"
 }
```

刪除整個堆疊集

下列部署設定檔會刪除整個堆疊集及其所有相關聯的堆疊執行個體。

```
{
     "deployment_action": “delete”,
     "stack_set_name": "sample-stack-set",
     "stack_set_desciption": "this is a sample stack set",
    "deployment_targets": {
                                "dev": {
                                        "org_units": ["dev-org-unit-1", "dev-org-
unit-2"],
                                        "regions": ["us-east-2"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                                "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                                "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
```

```
                        },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

從部署中排除帳戶

下列部署規劃檔將屬於 OU dev-org-unit-1 一部分的帳戶111122223333從部署中排除。

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
                        "dev": {
                                    "org_units": ["dev-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": ["111122223333"],
                                    "filter_type": "DIFFERENCE"
                                },
                        "test": {
                                    "org_units": ["test-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                },
                        "prod": {
                                    "org_units": ["prod-org-unit-1"],
                                    "regions": ["us-east-1"],
                                    "filter_accounts": [],
                                    "filter_type": ""
                                }
                       },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

將應用程式部署到 OU 中的帳戶子集

下列部署組態檔案只會將應用程式部署到 OU dev-org-unit-1 中的三個帳戶 (111122223333444455556666、和777788889999)。

```
{
    "deployment_action": "deploy",
    "stack_set_name": "sample-stack-set",
    "stack_set_desciption": "this is a sample stack set",
   "deployment_targets": {
                            "dev": {
                                        "org_units": ["dev-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": ["111122223333",
"444455556666", "777788889999"],
                                        "filter_type": "INTERSECTION"
                                    },
                            "test": {
                                        "org_units": ["test-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    },
                            "prod": {
                                        "org_units": ["prod-org-unit-1"],
                                        "regions": ["us-east-1"],
                                        "filter_accounts": [],
                                        "filter_type": ""
                                    }
                          },
    "cft_capabilities": ["CAPABILITY_IAM", "CAPABILITY_NAMED_IAM"],
    "auto_deployement": "True",
    "retain_stacks_on_account_removal": "True",
    "region_deployment_concurrency": "PARALLEL"
}
```

使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔

由阿里·阿斯福爾（AWS）和亞倫列儂（AWS）創建

環境：PoC 或試點	技術： DevOps；軟體開發與測試；管理與治理；安全性、身分識別、合規性；基礎架構	工作負載：開源
AWS 服務：Amazon SNS; Amazon SQS; AWS CodeBuild; AWS CodePipeline; AWS Systems Manager; AWS CodeCommit		

Summary

您可以將 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體與 AWS Systems Manager 整合，以自動化操作任務並提供更多可見度和控制權。若要與 Systems Manager 整合，EC2 執行個體必須已安裝 AWS Systems Manager 代理程式 (SSM 代理程式) 和 `AmazonSSMManagedInstanceCore` AWS Identity and Access Management (IAM) 政策附加至其執行個體設定檔。

但是，如果您想確保所有 EC2 執行個體設定檔都附加了`AmazonSSMManagedInstanceCore`政策，則更新沒有執行個體設定檔的新 EC2 執行個體或具有執行個體設定檔但沒有該`AmazonSSMManagedInstanceCore`政策的 EC2 執行個體可能會面臨挑戰。在多個 Amazon Web Services (AWS) 帳戶和 AWS 區域之間新增此政策也可能很困難。

透過在 AWS 帳戶中部署三個雲端託管政策，此模式有助於解決這些挑戰：

- 第一個雲端託管人政策會檢查具有執行個體設定檔但沒有`AmazonSSMManagedInstanceCore`政策的現有 EC2 執行個體。然後會附加`AmazonSSMManagedInstanceCore`原則。
- 第二個 Cloud 託管人政策會檢查沒有執行個體設定檔的現有 EC2 執行個體，並新增已附加`AmazonSSMManagedInstanceCore`政策的預設執行個體設定檔。
- 第三個雲端託管人政策會在您的帳戶中建立 AWS Lambda 函數，以監控 EC2 執行個體和執行個體設定檔的建立。這可確`AmazonSSMManagedInstanceCore`保在建立 EC2 執行個體時自動附加政策。

此模式使用 AWS DevOps 工具將雲端託管人政策的持續大規模部署到多帳戶環境，而無需佈建單獨的運算環境。

先決條件和限制

先決條件

- 兩個或多個作用中的 AWS 帳戶。一個帳戶是安全帳戶，其他帳戶是成員帳戶。
- 在安全帳戶中佈建 AWS 資源的許可。此模式使用管理員權限，但您應該根據組織的需求和策略授與權限。
- 能夠從安全帳戶擔任 IAM 角色到成員帳戶，並建立必要的 IAM 角色。如需詳細資訊，請參閱 IAM 文件中的使用 IAM 角色在 AWS 帳戶之間委派存取權。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。基於測試目的，您可以使用`aws configure`命令或設定環境變數來設定 AWS CLI。重要事項：不建議在生產環境中使用此功能，因此我們建議僅授予此帳戶的最低權限存取權。有關此方面的詳細資訊，請參閱 IAM 文件中的授予最低權限。
- 該`devops-cdk-cloudcustodian.zip`文件（附件），下載到您的本地計算機。
- 熟悉 Python。
- 安裝和設定所需的工具 (Node.js、AWS Cloud Development Kit (AWS CDK) 和 Git)。您可以使用`install-prerequisites.sh`檔案中的`devops-cdk-cloudcustodian.zip`檔案來安裝這些工具。 確保您以 root 權限運行此文件。

限制

- 雖然此模式可用於生產環境，但請確保所有 IAM 角色和政策都符合組織的需求和政策。

Package 版本

- 雲端託管人 0.9 版或更新版本
- TypeScript 版本 3.9.7 或更新版本
- Node.js 版本 14.15.4 或更新版本
- npm版本 7.6.1 或更新版本
- AWS CDK 版本 1.96.0 或更新版本

架構

該圖顯示以下工作流程：

1. 雲端託管政策會推送至安全帳戶中的 AWS CodeCommit 儲存庫。Amazon CloudWatch 事件規則會自動啟動 AWS CodePipeline 管道。
2. 管道會從中擷取最新的程式碼， CodeCommit 並將其傳送至 AWS 處理之持續整合和持續交付 (CI/CD) 管道的持續整合部分。 CodeBuild
3. CodeBuild 執行完整 DevSecOps 動作 (包括 Cloud Managed 政策上的政策語法驗證)，並以`--dryrun`模式執行這些政策以檢查識別的資源。
4. 如果沒有錯誤，下一個工作會警示管理員檢閱變更並核准部署至成員帳戶。

技術堆疊

- AWS CDK
- CodeBuild
- CodeCommit
- CodePipeline
- IAM
- Cloud Custodian

自動化和規模

AWS CDK 管道模組佈建一個 CI/CD 管道，除了使 CodePipeline 用 AWS 堆疊部署 AWS 資源之外 CodeBuild，還可用來協調原始程式碼的建置和測試。 CloudFormation 您可以將此模式用於組織中的所有成員帳戶和區域。您也可以擴充`Roles creation`堆疊，以在成員帳戶中部署其他 IAM 角色。

工具

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可在程式碼中定義雲端基礎設施，並透過 AWS CloudFormation 佈建雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- AWS CodeBuild 是雲端中的全受管建置服務。

- AWS CodeCommit 是一種版本控制服務，可用於私有存放和管理資產。
- AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。
- AWS Identity and Access Management 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。
- Cloud Bookdian 是一種工具，可將大多數組織用於管理其公共雲帳戶的數十種工具和腳本統一為一個開源工具。
- Node.js 是建立在谷歌瀏覽器的 V8 JavaScript 引擎 JavaScript 運行時。

Code

有關此模式中使用的模塊，帳戶函數，文件和部署命令的詳細列表，請參閱README文件中的devops-cdk-cloudcustodian.zip文件（附件）。

史诗

使用 AWS CDK 設定管道

任務	描述	所需技能
設定 CodeCommit 儲存庫。	1. 解壓縮本機電腦上工作目錄中的devops-cdk-cloudcustodian.zip 檔案 (附加)。 2. 登入安全帳戶的 AWS 管理主控台，開啟 CodeCommit 主控台，然後建立新的存放devops-cdk-cloudcustodian 庫。 3. 切換到項目目錄並將 CodeCommit 存儲庫設置為原點，提交更改，然後通過運行以下命令將其推送到 origin 分支： • cd devops-cdk-cloudcustodian	開發人員

任務	描述	所需技能
	• `git init --initial-branch=main` • `git add . git commit -m 'initial commit'` • `git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/devops-cdk-cloudcustodian` • `git push origin main` 如需詳細資訊，請參閱 AWS CodeCommit 文件中的建立 CodeCommit 儲存庫。	
安裝所需的工具。	使用該`install-prerequisites.sh` 文件在 Amazon Linux 上安裝所有必需的工具。這不包括 AWS CLI，因為它已預先安裝。 如需相關詳細資訊，請參閱 AWS CDK 文件中 AWS CDK 入門的先決條件一節。	開發人員

任務	描述	所需技能
安裝所需的 AWS CDK 套件。	1. 在 AWS CLI 中執行以下命令來設定虛擬環境：`$ python3 -m venv .env` 2. 執行下列命令以啟動您的虛擬環境：`$ source .env/bin/activate` 3. 啟動虛擬環境之後，請執行下列命令來安裝所需的相依性：`$ pip install -r requirements.txt` 4. 若要新增其他相依性 (例如其他 AWS CDK 程式庫)，請將它們新增至`requirements.txt` 檔案，然後執行下列命令：`pip install -r requirements.txt` AWS CDK 需要下列套件，並包含在`requirements.txt` 檔案中： • `aws-cdk.aws-cloudwatch` • `aws-cdk.aws-codebuild` • `aws-cdk.aws-codecommit` • `aws-cdk.aws-codedeploy` • `aws-cdk.aws-codepipeline` • `aws-cdk.aws-codepipeline-actions`	開發人員

任務	描述	所需技能
	• aws-cdk.aws-events • aws-cdk.aws-events-targets • aws-cdk.aws-iam • aws-cdk.aws-logs • aws-cdk.aws-s3 • aws-cdk.aws-sns • aws-cdk.aws-sns-subscriptions • aws-cdk.aws-sqs • aws-cdk.core	

設定您的環境

任務	描述	所需技能
更新必要的變數。	開啟 CodeCommit 儲存庫根資料夾中的vars.py檔案，並更新下列變數： • var_deploy_region = 'us-east-1' 使用您希望部署管道的 AWS 區域進行更新。 • var_codecommit_repo_name = "cdk-cloudcustodian" 使用 CodeCommit 儲存庫的名稱進行更新。 • var_codecommit_branch_name = "main" 使	開發人員

任務	描述	所需技能
	用 CodeCommit 分支的名稱更新。 • `var_adminEmail=notifyadmin@email.com` ’ 使用核准變更之管理員的電子郵件地址進行更新。 • 使用 Slack Webhook 更新 `var_slackWebHookUrl = https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXX ',` 用於在進行變更時傳送雲端託管人通知。 • `var_orgId = ‘o-yyyyyyyyyy’` 使用您的組織 ID 更新。 • `security_account = ‘123456789011’` 使用部署管道的帳戶的 AWS 帳戶 ID 進行更新。 • 使`member_accounts = [‘111111111111’,’111111111112’,’111111111113’]` 用您想要啟動 AWS CDK 堆疊並部署必要 IAM 角色的成員帳戶進行更新。 • `True`如果您希望管道自動將 AWS CDK 引導至您的成員帳戶，請設`cdk_bootstrap_member_accounts = True` 定為。 如果	

任務	描述	所需技能
	設True為，則還需要成員帳戶中現有 IAM 角色的名稱，該角色可以從安全帳戶承擔。此 IAM 角色也必須具有啟動 AWS CDK 的必要許可。 • 使cdk_bootstrap_role = ‘AWSControlTowerExecution’ 用成員帳戶中現有的 IAM 角色進行更新，這些角色可以從安全帳戶承擔。此角色也必須獲得啟動 AWS CDK 的權限。注意：這僅在設定cdk_bootstrap_member_accounts 為時適用True。	

任務	描述	所需技能
使用成員帳戶資訊更新帳戶 .yml 檔案。	若要針對多個帳戶執行 c7n-org 雲端託管工具，您必須將`accounts.yml` 設定檔放在儲存庫的根目錄中。以下是 AWS 雲端託管人設定檔範例： `accounts:` `- account_id: '123123123123'` `  name: account-1` `  regions:` `  - us-east-1` `  - us-west-2` `  role: arn:aws:iam::123123123123:role/CloudCustodian` `  vars:` `  charge_code: xyz` `  tags:` `  - type:prod` `  - division:some division` `  - partition:us` `  - scope:pci`	開發人員

啟動 AWS 帳戶

任務	描述	所需技能
自舉安全帳戶。	透過執行`deploy_account` 下列命令來啟動`cloudcustodian_stack` 應用程式： `cdk bootstrap -a 'python3`	開發人員

任務	描述	所需技能
	`cloudcustodian/cloudcustodian_stack.py`	
選項 1-自動引導成員帳戶。	如果在`vars.py`檔案`True`中將`cdk_bootstrap_member_accounts` 變數設定為，則管線會自動啟動載入`member_accounts` 變數中指定的帳戶。 如果需要，您可以使`*cdk_bootstrap_role*` 用可以從安全帳戶承擔的 IAM 角色進行更新，並具有啟動 AWS CDK 所需許可的 IAM 角色。 新增至`member_accounts` 變數的新帳戶會由管線自動啟動載入，以便部署必要的角色。	開發人員

任務	描述	所需技能
選項 2-手動引導成員帳戶。	雖然我們不建議使用此方法，但是您可以將值設定`cdk_bootstrap_member_accounts` 為，`False`並透過執行下列命令手動執行此步驟： `$ cdk bootstrap -a 'python3 cloudcustodian/member_account_roles_stack.py' \ --trust {security_account_id} \ --context assume-role-credentials:writeIamRoleName={role_name} \ --context assume-role-credentials:readIamRoleName={role_name} \ --mode=ForWriting \ --context bootstrap=true \ --cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess`  重要事項：請務必使用可從安全帳戶承擔的 IAM 角色名稱更新和`{role_name}` 值，並	開發人員

任務	描述	所需技能
	具有啟動 AWS CDK 所需的許可。{security_account_id} 您也可以使用其他方法來引導成員帳戶，例如使用 AWS CloudFormation。如需有關此功能的詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。	

部署 AWS CDK 堆疊

任務	描述	所需技能
在成員帳戶中建立 IAM 角色。	執行下列命令以部署 member_account_roles_stack 堆疊，並在成員帳戶中建立 IAM 角色： `cdk deploy --all -a 'python3 cloudcustodian/member_account_roles_stack.py' --require-approval never`	開發人員
部署雲端託管管線堆疊。	執行下列命令，以建立部署到安全性帳戶的雲端託cloudcustodian_stack.py 管管線： `cdk deploy -a 'python3 cloudcustodian/cloudcustodian_stack.py'`	開發人員

相關資源

- 開始使用 AWS CDK

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS CDK 為微型服務自動建置 CI/CD 管道和 Amazon ECS 叢集

創建者瓦爾沙拉朱 (AWS)

環境：PoC 或試點	技術： DevOps；容器與微服務；現代化；基礎架構	AWS 服務：AWS CodeBuild；AWS CodeCommit；AWS CodePipeline；Amazon ECS；AWS CDK

Summary

此模式說明如何自動建立持續整合和持續交付 (CI/CD) 管道和基礎基礎設施，以便在 Amazon Elastic Container Service (Amazon ECS) 上建立和部署微服務。如果您想要設定 proof-of-concept CI/CD 管線，向您的組織顯示 CI/CD、微服務和. DevOps 您也可以使用此方法建立初始 CI/CD 管線，然後根據組織的需求自訂或變更這些管道。

該模式的方法可建立生產環境和非生產環境，每個環境都有虛擬私有雲 (VPC) 和一個設定為在兩個可用區域中執行的 Amazon ECS 叢集。這些環境由您的所有微服務共用，然後您為每個微服務建立 CI/CD 管道。這些 CI/CD 管道從 AWS 的來源儲存庫提取變更 CodeCommit、自動建立變更，然後將其部署到您的生產和非生產環境中。當管線成功完成其所有階段時，您可以使用 URL 存取生產環境和非生產環境中的微服務。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶。
- 包含`starter-code.zip`檔案 (附加) 的現有 Amazon 簡易儲存服務 (Amazon S3) 儲存貯體。
- AWS Cloud Development Kit (AWS CDK) 已在您的帳戶中安裝和設定。如需這方面的詳細資訊，請參閱 AWS CDK 文件中的 AWS CDK 入門。
- Python 3 和`pip`, 安裝和配置. 有關這方面的更多信息，請參閱 Python 文檔。
- 熟悉 AWS CDK，AWS，AWS CodePipeline CodeBuild CodeCommit，Amazon Elastic Container Registry (Amazon ECR)，Amazon ECS 和 AWS Fargate。
- 熟悉碼頭工人。

- 對 CI/CD 的了解和. DevOps

限制

- 一般 AWS 帳戶限制適用。如需詳細資訊，請參閱 AWS 一般參考文件中的 AWS 服務配額。

產品版本

- 該代碼已使用 Node.js 版本 16.13.0 和 AWS CDK 版本 1.132.0 進行了測試。

架構

該圖顯示以下工作流程：

1. 應用程式開發人員將程式碼提交至 CodeCommit 儲存庫。
2. 配管已啟動。
3. CodeBuild 構建碼頭映像並將其推送到 Amazon ECR 存儲庫
4. CodePipeline 將新映像部署到非生產 Amazon ECS 叢集中的現有 Fargate 服務。
5. Amazon ECS 將映像從 Amazon ECR 存儲庫提取到非生產的 Fargate 服務中。
6. 使用非生產 URL 執行測試。
7. 發行管理員會核准生產部署。
8. CodePipeline 將新映像部署到生產 Amazon ECS 叢集中的現有 Fargate 服務
9. Amazon ECS 將圖像從 Amazon ECR 存儲庫提取到生產 Fargate 服務中。
10生產使用者可以使用生產 URL 存取您的功能。

技術, 堆

- AWS CDK
- CodeBuild
- CodeCommit
- CodePipeline
- Amazon ECR

- Amazon ECS
- Amazon VPC

自動化和規模

您可以使用此模式的方法為共用 AWS CloudFormation 堆疊中部署的微型服務建立管道。自動化可以在每個 VPC 中建立多個 Amazon ECS 叢集，也可以為部署在共用 Amazon ECS 叢集中的微型服務建立管道。但是，這需要您提供新的資源資訊做為管線堆疊的輸入。

工具

- AWS CDK — AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可在程式碼中定義雲端基礎設施，並透過 AWS 佈建雲端基礎設施。 CloudFormation
- AWS CodeBuild — AWS CodeBuild 是雲端中的全受管建置服務。 CodeBuild 編譯您的原始程式碼、執行單元測試，並產生準備好部署的成品。
- AWS CodeCommit — AWS CodeCommit 是一種版本控制服務，可讓您在 AWS 雲端私有存放和管理 Git 儲存庫。 CodeCommit 您無需管理自己的原始檔控制系統，也不必擔心擴充其基礎架構。
- AWS CodePipeline — AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。您可以快速建模並設定軟體發行程序的不同階段。 CodePipeline 自動執行持續發行軟體變更所需的步驟。
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) 是可高度擴展、快速的容器管理服務，可用於在叢集上執行、停止和管理容器。您可以在 AWS Fargate 管理的無伺服器基礎設施上執行任務和服務。或者，若要進一步控制基礎設施，您可以在您管理的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體叢集上執行任務和服務。
- Docker — Doc ker 幫助開發人員將任何應用程序打包，運送和運行作為一個輕量級，便攜和自給自足的容器。

Code

此模式的代碼可在`cicdstarter.zip`和`starter-code.zip`文件（附加）中找到。

史诗

設定您的環境

任務	描述	所需技能
設定 AWS CDK 的工作目錄。	1. 建立在本機電腦cicdproject 上命名的目錄。 2. 將cicdstarter.zip 文件（附件）下載到目cicdproject 錄中並解壓縮。這將創建一個名為的文件夾cicdstarter 。 3. 執行 cd <user-home>/cicdproject/cicdstarter 命令。 4. 透過執行python3 -m venv .venv指令來設定 Python 虛擬環境。 5. 執行 source ./.venv/bin/activate 命令。 6. 透過執行aws configure 命令或使用下列環境變數來設定 AWS 環境： • AWS_ACCESS_KEY_ID • AWS_SECRET_ACCESS_KEY • AWS_DEFAULT_REGION	AWS DevOps、雲端基礎設施

建立共用基礎結構

任務	描述	所需技能
建立共用基礎結構。	1. 在您的工作目錄中，執行cd cicdvpcecs 命令。 2. 運行命pip3 install -r requirements.txt 令以安裝所有必需的 Python 依賴關係 3. 執行cdk bootstrap command以設定適用於 AWS CDK 的 AWS 環境。 4. 執行 cdk synth --context aws_account=<aws_account_ID> --context aws_region=<aws-region> 命令。 5. 執行 cdk deploy --context aws_account=<aws_account_ID> --context aws_region=<aws-region> 命令。 6. AWS CloudFormation 堆疊會建立下列基礎設施： • 名為的非生產 VPC cicd-vpc-ecs/cicd-vpc-nonprod • 名為的生產 VPC cicd-vpc-ecs/cicd-vpc-prod	AWS DevOps、雲端基礎設施

任務	描述	所需技能
	• 名為的非生產 Amazon ECS 叢集 `cicd-ecs-nonprod` • 命名為的生產 Amazon ECS 叢集 `cicd-ecs-prod`	
監控 AWS CloudFormation 堆疊。	1. 登入 AWS 管理主控台，開啟 AWS CloudFormation 主控台，然後從清單中選擇`cicd-vpc-ecs` 堆疊。 2. 在堆疊詳細資料窗格中，選擇 [事件] 索引標籤，然後監視堆疊建立的進度。	AWS DevOps、雲端基礎設施
測試 AWS CloudFormation 堆疊。	1. 建立 `cicd-vpc-ecs` AWS CloudFormation 堆疊之後，請確定已建立`cicd-vpc-ecs/cicd-vpc-nonprod` 和 `cicd-vpc-ecs/cicd-vpc-prod` VPC。 2. 確保已建立`cicd-ecs-nonprod` 和 `cicd-ecs-prod` Amazon ECS 叢集。 重要：請確定您已記錄兩個 VPC 的 ID，以及兩個 VPC 中預設安全性群組的安全性群組識別碼。	AWS DevOps、雲端基礎設施

為微服務建立 CI/CD 管線

任務	描述	所需技能
建立微服務的基礎結構。	1. 命名您的微服務。例如，此模式使用myservice1 作為微服務的名稱。 2. 在您的工作目錄中運行cd <working-directory>/cdkpipeline 命令。 3. 執行 pip3 install -r requirements.txt 命令。 4. 執行此模式的 [其他資訊] 區段中提供的完整cdk synth命令。 5. 執行此模式的 [其他資訊] 區段中提供的完整cdk deploy命令。 注意：您也可以使用目錄中的cdk.json檔案來提供這兩個命令的值。	AWS DevOps、雲端基礎設施
監控 AWS CloudFormation 堆疊。	開啟 AWS CloudFormation 主控台並監控myservice1-cicd-stack 堆疊的進度。最後，狀態會變更為CREATE_COMPLETE 。	AWS DevOps、雲端基礎設施
測試 AWS CloudFormation 堆疊。	1. 在 AWS 主 CodeCommit 控台上，確認名為的存放庫myservice1 存在並包含入門程式碼。	

任務	描述	所需技能
	2. 在 AWS 主 CodeBuild 控台上，確認名為的建置專案myservice1 存在。 3. 在 Amazon ECR 主控台上，確認名為myservice 1 的 Amazon ECR 儲存庫是否存在。 4. 在 Amazon ECS 主控台上，確認名為的 Fargate 服務同時myservice1 存在於非生產和生產 Amazon ECS 叢集中。 5. 在 Amazon Elastic Compute Cloud (Amazon EC2) 主控台上，確認已建立非生產和生產應用程式負載平衡器。記錄 ALB 的 DNS 名稱。 6. 在 AWS 主 CodePipel ine 控台上，確認名為的管道myservice 1 存在。它必須具有SourceBuild、Deploy-NonProd 、和Deploy-Pr od 階段。管線也應具有in progress狀態。 7. 監視管線，直到所有階段都完成為止。 8. 手動核准它進行生產。 9. 在瀏覽器視窗中，輸入 ALB 的 DNS 名稱。 10應用程式應顯示Hello World在非生產和生產 URL 中。	

任務	描述	所需技能
使用管道。	1. 開啟您先前建立的存放 CodeCommit 庫，然後開啟`index.js`檔案。 2. 使用 `Hello CI/CD` 取代 `Hello World`。 3. 保存並提交更改到主分支。 4. 驗證管線是否啟動，且變更會經過`BuildDeploy-NonProd` 、和`Deploy-Prod` 階段。 5. 手動核准生產。 6. 現在應該會顯示*`Hello CICD`*生產和非生產 URL。	AWS DevOps、雲端基礎設施
為每個微服務重複此史詩。	重複此史詩中的工作，為您的每個微服務建立 CI/CD 管道。	AWS DevOps、雲端基礎設施

相關資源

- 搭配 AWS 用 Python
- AWS Python 參考
- 使用 AWS CDK 建立 AWS Fargate 服務

其他資訊

`cdk synth` 命令

```
cdk synth --context aws_account=<aws_account_number> --context
 aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production
 VPC> --context vpc_prod_id=<id_of_production_VPC> --context
 ecssg_nonprod_id=< default_security_group_id_of_non-production_VPC>
 --context ecssg_prod_id=<default_security_group_id_of_production_VPC>
 --context code_commit_s3_bucket_for_code=<S3 bucket name> --context
```

```
 code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
 microservice_name=<name_of_microservice>
```

cdk deploy command

```
cdk deploy --context aws_account=<aws_account_number> --context
 aws_region=<aws_region> --context vpc_nonprod_id=<id_of_non_production_VPC>
 --context vpc_prod_id=<id_of_production_VPC> --context ecssg_nonprod_id=<
 default_security_group_id_of_non-production_VPC> --context
 ecssg_prod_id=<default_security_group_id_of_production_VPC> --
context code_commit_s3_bucket_for_code=<S3 bucket name> --context
 code_commit_s3_object_key_for_code=<Object_key_of_starter_code> --context
 microservice_name=<name_of_microservice>
```

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 DevOps 實務和 AWS Cloud9 建立鬆散結合的架構與微型服務

創建者亞歷山大·納迪（AWS）

環境：PoC 或試點	技術： DevOps; 無服務器; Web 和移動應用程序; 數據庫	AWS 服務：AWS Cloud9; AWS; AWS CloudFormation CodePipeline; Amazon DynamoDB; AWS CodeCommit

Summary

此模式示範如何針對開始在 Amazon Web Services (AWS) 上測試 DevOps 實務的開發人員和開發主管，在無伺服器架構中開發典型 Web 應用程式。它構建了一個示例應用程序，該應用程序創建用於瀏覽和購買書籍的店面和後端，並提供可以獨立開發的微服務。該模式使用 AWS Cloud9 做為開發環境、Amazon DynamoDB 資料庫做為資料存放區，以及 AWS 等 AWS 服務來提 CodeBuild 供持續整合 CodePipeline 和持續部署 (CI/CD) 功能。

此模式會引導您完成下列開發活動：

- 建立標準 AWS Cloud9 開發環境
- 使用 AWS CloudFormation 範本建立書籍的 Web 應用程式和微服務
- 使用 AWS Cloud9 修改前端、提交變更和測試變更
- 建立並測試微服務的 CI/CD 管線
- 自動化單元測試

此模式的程式碼在 GitHub AWS DevOps 端對端研討會儲存庫中提供。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS DevOps 端對端研討會的檔案下載到您的電腦

重要事項：在您的 AWS 帳戶中建立此示範應用程式會建立和消耗 AWS 資源。您必須負責建立和執行應用程式所使用的 AWS 服務和資源成本。完成工作後，請務必移除所有資源，以避免持續收費。如需清理指示，請參閱《史詩》一節。

限制

本逐步解說僅用於示範和開發目的。若要在生產環境中使用它，請參閱 AWS Identity and Access Management (IAM) 文件中的安全最佳實務，並對 IAM 角色、Amazon DynamoDB 和其他使用的服務進行必要的變更。Web 應用程式衍生自 AWS 書店示範應用程式；如需其他考量事項，請參閱 README 檔案的已知限制一節。

架構

書店應用程式的架構會在 AWS 書店示範應用程式的 README 檔案的架構區段中說明。

從部署的角度來看，書店演示應用程序使用單個 CloudFormation 模板將所有服務和對象部署在一個堆棧中。此模式會進行一些變更，以示範特定開發人員或團隊如何在特定產品 (Books) 中運作，並獨立於應用程式的其餘部分進行更新。因此，此模式的程式碼會將 Books 微服務的 AWS Lambda 函數和相關物件分隔為第二個 CloudFormation 範本，以建立 Books 堆疊。這使得可以通過使用 CI/CD 實踐來查看微服務正在更新。在下圖中，虛線邊框標識 Books 微服務。

工具

工具

- 開玩笑框架進行測試 JavaScript
- Python 3.9

Code

此模式的原始程式碼和範本可在 GitHub AWS DevOps 端對端研討會儲存庫中取得。在按照 Epics 部分中的步驟進行操作之前，請將存儲庫中的所有文件下載到您的計算機。

注意：「Epics」段落提供此逐步解說的高階步驟，以提供有關程序的一般資訊。若要完成每個步驟，請參閱 AWS DevOps 端對端研討會儲存庫中的 README 檔案，以取得詳細指示。

AWS DevOps 端對端研討會儲存庫可擴充 AWS 書店示範應用程式儲存庫，並使用修改過的 AWS Cloud9 啟動載入程式碼版本來建立 AWS Cloud9 IDE。

最佳實務

使用書店應用程序很簡單。以下是一些建議的最佳作法：

- 安裝應用程序時，您可以使用您選擇的項目名稱或使用默認名稱（`demobookstore`）以方便起見。
- 應用程式啟動並執行之後，如果您想繼續測試一天，最好關閉 Amazon Neptune 資料庫，因為資料庫執行個體可能會產生額外費用。不過，請注意，資料庫會在七天後自動啟動。
- 如需程式碼詳細資訊，請參閱 AWS 書店示範應用程式儲存庫的文件。它描述了每個微服務和表。
- 有關其他最佳實踐，請參閱如果您有時間的某些挑戰... AWS DevOps 端對端研討會儲存庫中的讀我檔案區段。我們建議您檢閱這些資訊，深入了解安全性的其他功能，並練習解耦服務。

史诗

下載源代碼

任務	描述	所需技能
從下載源代碼 GitHub。	此模式的原始程式碼和範本可在 GitHub AWS DevOps 端對端研討會儲存庫中取得。在您執行「Epics」區段中的後續步驟之前，請先將儲存庫中的所有檔案下載到您的電腦。 注意：「Epics」段落提供此逐步解說的高階步驟，以提供有關程序的一般資訊。若要完成每個步驟，請參閱 AWS DevOps 端對端研討會儲存庫中的 README 檔案，以取得詳細指示。 AWS DevOps 端對端研討會儲存庫可擴充 AWS 書店示範應用程式儲存庫，並使用修改過的 AWS Cloud9 啟動載入程	應用程式開發人員

任務	描述	所需技能
	式碼版本來建立 AWS Cloud9 IDE。	

建置書店網頁應用程式和 Books 微服務

任務	描述	所需技能
建立書店應用程式的前端和 Lambda 函數。	1. 登入CloudFormation 主控台，然後部署`DemoBookstoreMainTemplate.yml` 範本以建立 DemoBookStoreStack 堆疊。這會建立 Books 微服務之外的前端和 Lambda 函數。 2. 在堆疊的 [輸出] 索引標籤中，記下WebApplication標籤下的網站 URL。	開發人員
建立「書籍」微服務。	在主CloudFormation 控台上，部署`DemoBookstoreBooksServiceTemplate.yml` 範本以建立 DemoBooksServiceStack 堆疊。	開發人員
測試您的應用程式。	使用 DemoBookStoreStack 堆疊中的網站 URL 存取書店應用程式。	開發人員

使用 Cloud9 環境來維護您的應用程式

任務	描述	所需技能
建立一個 AWS Cloud9 IDE。	在主CloudFormation 控台上，部署C9EnvironmentTemplate.yml 範本以建立 AWS Cloud9 環境。	開發人員，開發人員
建立 CodeCommit 儲存庫。	1. 登入 AWS CodeCommit 主控台，並確認您有一個demobookstore-WebAssets 儲存庫，其中包含前端應用程式的程式碼。 2. 為demobookstore-BooksService 所謂的 Books 微服務創建一個存儲庫。 3. 使用git clone命令複製 AWS Cloud9 (demobookstore-WebAssets 和demobookstore-BooksService) 中的兩個儲存庫。	開發人員
更改前端中的代碼並檢查管道。	1. 使用 AWS Cloud9 在網頁上進行一些程式碼變更。這將更新存demobookstore-WebAssets 儲庫。 2. 在 AWS CodePipeline 主控台上，確認示範儲存資產管道正在執行。 3. 通過從瀏覽器中刷新它來測試您的 Web 應用程序（Firefox 上的 Ctrl+F5）。	開發人員

為叢書微服務實作 CI/CD 管線

任務	描述	所需技能
新增用於組建和服務更新的 YAML 檔案。	1. 在 AWS Cloud9 中，上傳`buildspec.yml` 和`DemoBookstoreBooksServiceUpdateTemplate.yml` 檔案。 • `buildspec.yml` 具有構建說明，還包括自動化測試的測試說明。他們在這一點上被評論，並將在以後使用。 • `DemoBookstoreBooksServiceUpdateTemplate.yml` 是的更新版本`DemoBookstoreBooksServiceTemplate.yml` ，將用於管線的部署階段。 2. 提交並推送文件。	開發人員
為建置管道建立 S3 儲存貯體。	若要建立 S3 儲存貯體，請按照 Amazon S3 文件中的指示進行操作。 • 值區名稱必須是全域唯一的，例如`demobookstore-books-service-pipeline-bucket-<YYYYMMDDHHMM>` 。	開發人員

任務	描述	所需技能
	• 清除 [封鎖所有公用存取] 核取方塊，然後選取 [我確認...] 核取方塊。	
使用 IAM 建立 CloudFormation 部署角色。	建立`demobookstore-CloudFormation-role` 角色並附加`AdministratorAccess` 原則。在下一個史詩中，您可以重新配置此角色以獲得最低權限。	開發人員
建立新管道以自動化 Books 微服務的建置和部署。	使用「提交」、「建置」和「部署」階段建立BooksService管道 (例如，demo書店-Pipeline)，如讀我檔案中所述。	開發人員
在 AWS Cloud9 中測試您的微服務。	在ListBooks功能中進行更改並查看管道工作。	開發人員
自動執行 ListBooks Lambda 函數的單元測試。	在 AWS Cloud9 IDE 中，啟用組建以執行單元測試，並檢查測試結果。如需指示，請參閱讀我檔案。	開發人員

(選擇性) 實作其他功能

任務	描述	所需技能
確保您的解決方案安全。	配置`demobookstore-CloudFormation-role` 為具有最低權限，並檢查其他使用的角色。	開發人員
消除 CloudFormation 模板中的依賴關係。	`DemoBookstoreMainTemplate.yml` 模板和模	開發人員

任務	描述	所需技能
	板之間交換信息的方法基於輸出和導入。DemoBooks toreBooksServiceTe mplate.yml 在這兩個模板之間傳遞值會增加依賴關係。若要消除相依性，請考慮使用 AWS Systems Manager Parameter Store。	
建立購物車微服務。	使用 Books 微服務做為範例，將購物車相關功能從範 DemoBookstoreMainT emplate.yml 本中取出，並建立購物車微服務。	開發人員

清除

任務	描述	所需技能
刪除 S3 儲存貯體。	在 Amazon S3 主控台上，刪除下列與範例 Web 應用程式相關聯的儲存貯體： • 為 AWS 書店示範應用程式建立的兩個值區。值區名稱以您在建立前端 CloudForm ation 時為 AWS 提供的堆疊名稱開頭，DemoBookS toreStack例如。 • <YYYYMMDDHHMM>一個用於構建管道的存儲桶; 例如，demobookstore-books-service-pipeline-bucket-。	開發人員

任務	描述	所需技能
刪除堆疊。	在CloudFormation 主控台上，刪除與範例 Web 應用程式相關聯的堆疊： • DemoBooksServiceStack • DemoBookStoreStack 移除作業可能需要超過 90 分鐘。如果移除失敗，請再次刪除它們，並根據通知刪除任何手動資源 (例如，VPC 或網路介面)。	開發人員
刪除 IAM 角色。	在 IAM 主控台上，刪除下列角色： • `demobookstore-Cloudformation-role` • `demobookstore-BooksService-BuildProject-service-role` 如需 step-by-step 指示，請參閱 IAM 文件。	開發人員

相關資源

- AWS 書店演示應用程序
- AWS Cloud9 啟動載入範例
- 在 AWS CloudFormation 主控台建立堆疊 (AWS CloudFormation 文件)
- 建立儲存貯體 (Amazon S3 文件)

其他資訊

如需詳細 step-by-step 指示，請參閱 AWS DevOps 端對端研討會 GitHub 儲存庫中的 README 檔案。

關於 2023 年 5 月的更新：此模式已更新為使用較新版本的節點和 Python。我們更新了原始碼中的許多套件，並移除了 Glyphicon，因為它不再是免費的。我們也移除了 AWS 書店示範應用程式儲存庫上的所有相依性，因此這兩個儲存庫現在可以獨立進化。

使用 GitHub 動作和地形表單建置碼頭映像並將其推送到 Amazon ECR

創建者魯奇卡莫迪（AWS）

代碼存儲庫：docker-ecr-actions-workflow	環境：生產	技術： DevOps；容器與微服務；基礎架構
工作負載：所有其他工作	AWS 服務：Amazon ECR	

Summary

此模式說明如何建立可重複使用的 GitHub 工作流程以建立 Docerfile，並將產生的映像推送至 Amazon Elastic Container Registry (Amazon ECR)。該模式通過使用地形和操作自動化您的 Docker 文件的構建過程。 GitHub 如此可將人為錯誤的可能性降至最低，並大幅縮短部署時間。

GitHub 存放庫主要分支的 GitHub 推送動作會啟動資源的部署。工作流程會根據組織和存放庫名稱的 GitHub 組合，建立唯一的 Amazon ECR 儲存庫。然後，它將碼頭文件映像推送到 Amazon ECR 存儲庫。

先決條件和限制

前提

- 作用中的 AWS 帳戶
- 一個活躍的 GitHub 帳戶。
- 一個GitHub 存儲庫。
- 已安裝並設定地形版本 1 或更新版本。
- 用於 Terraform 後端的 Amazon 簡單存儲服務（亞馬遜 S3）存儲桶。
- 用於地形狀態鎖定和一致性的 Amazon DynamoDB 表。資料表必須具有以類型命名`LockID`的分割索引鍵`String`。如果未設定此選項，則會停用狀態鎖定。
- 一種 AWS Identity and Access Management (IAM) 角色，具有為 Terraform 設定 Amazon S3 後端的許可。如需設定指示，請參閱地形文件。

限制

此可重複使用的代碼僅通過操作進 GitHub 行了測試。

架構

目標技術堆疊

- Amazon ECR 儲存庫
- GitHub 動作
- 地形

目標架構

此圖展示了以下要點：

1. 用戶將 Docker 文件和地形模板添加到存儲庫中。 GitHub

2. 這些新增會啟 GitHub 動作業工作流程。

3. 工作流程會檢查 Amazon ECR 儲存庫是否存在。如果沒有，它會根據 GitHub 組織和存放庫名稱建立存放庫。

4. 工作流程會建立 Docker 檔案，並將映像推送至 Amazon ECR 儲存庫。

工具

Amazon 服務

- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器登錄服務。

其他工具

- GitHub 動作已整合到 GitHub 平台中，以協助您在 GitHub 儲存庫中建立、共用和執行工作流程。您可以使用 GitHub 動作來自動化工作，例如建置、測試和部署程式碼。
- Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署基礎結構。 HashiCorp

代碼存儲庫

此模式的程式碼可在 GitHub Docker ECR 動作工作流程存放庫中找到。

- 當您建立 GitHub 動作時，Docker 工作流程檔案會儲存在此儲存庫的`/.github/workflows/`資料夾中。此解決方案的工作流程位於工作流程. yaml 檔案中。
- 該文件`e2e-test`夾提供了一個示例 Docker 文件供參考和測試。

最佳實務

- 如需撰寫 Docker 檔案的最佳作法，請參閱 Doc ker 文件。
- 使用適用於 Amazon ECR 的 VPC 端點。VPC 私人雲端節點由 AWS 提供支援 PrivateLink，這項技術可讓您透過私有 IP 地址私有存取 Amazon ECR API。對於使用 Fargate 啟動類型的 Amazon ECS 任務，VPC 端點可讓任務從 Amazon ECR 提取私有映像，而無需為任務指派公用 IP 地址。

史诗

設定 OIDC 提供者和儲存庫 GitHub

任務	描述	所需技能
配置 OpenID Connect。	創建一個 OpenID Connect（OIDC）提供程序。您將在此動作中使用 IAM 角色的信任政策中使用提供者。有關說明，請參閱 GitHub 文檔中的在 Amazon Web Services 中配置 OpenID Connect。	AWS 管理員、AWS DevOps、一般 AWS
克隆存 GitHub 儲庫。	將 GitHub Docker ECR 動作工作流程存放庫複製到您的本機資料夾中： `$git clone https://github.com/aws-samples/docker-ecr-actions-workflow`	DevOps 工程師

自訂 GitHub 可重複使用的工作流程並部署 Docker 映像

任務	描述	所需技能
自訂啟動 Docker 工作流程的事件。	此解決方案的工作流程位於工作流程 .yaml 中。此指令碼目前設定為在收到`workflow_dispatch` 事件時部署資源。您可以將事件變更為，`workflow_call` 並從其他父工作流程呼叫工作流程來自訂此組態。	DevOps 工程師
自訂工作流程。	工作流程 .yaml 檔案設定為建立動態、可重複使用的工作流程。 GitHub 您可以編輯此檔案以自訂預設組態，或者如果您使用`workflow_dispatch` 事件手動起始部署，則可以從動 GitHub 作主控台傳遞輸入值。 • 請務必指定正確的 AWS 帳戶 ID 和目標區域。 • 建立 Amazon ECR 生命週期政策 (請參閱範例政策)，並據此更新預設路徑 (`e2e-test/policy.json`)。 • 工作流程檔案需要兩個 IAM 角色作為輸入： • 具有為 Terraform 設定 Amazon S3 後端之權限的 IAM 角色 (請參閱先決條件一節)。您可以在中更新預設角`workload-assumable-role` 色名稱。 `yaml`相應的文件。	DevOps 工程師

任務	描述	所需技能
	• 具有存取權限的 IAM 角色 GitHub。此角色也用於 Amazon ECR 政策中，以限制 Amazon ECR 操作。如需詳細資訊，請參閱 data.tf 檔案。	
部署地形範本。	工作流程會根據您設定的事件，自動部署建立 Amazon ECR 儲存庫的 Terraform 範本。 GitHub 這些範本可在 Github 儲存庫的根目錄中以 `.tf`檔案的形式使用。	AWS DevOps、 DevOps 工程師

故障診斷

問題	解決方案
當您將 Amazon S3 和 DynamoDB 設定為地形遠端後端時發生問題或錯誤。	請遵循 Terraform 文件中的指示，在 Amazon S3 和 DynamoDB 資源上為遠端後端組態設定所需的許可。
無法使用事件執行或啟`workflow_dispatch` 動工作流程。	設定為從`workflow_dispatch` 事件部署的工作流程只有在主分支上設定工作流程時才會運作。

相關資源

- 重複使用工作流程 (GitHub 文件)
- 觸發工作流程 (GitHub 文件)

使用 AWS CodeCommit、AWS 和 AWS Device Farm 建置和測試 iOS 應用程式 CodePipeline

創建者：阿卜杜拉希·奧拉耶 (AWS)

R 類型：不適用	來源：內部部署 DevOps 程序	目標：適用於 AWS 上開發 iOS 應用程式的 CI/CD 管道
創建者：AWS	環境：PoC 或試點	技術：網絡和移動應用程序; DevOps
AWS 服務：AWS CodeCommit；AWS CodePipeline；AWS Device Farm		

Summary

此模式概述了建立持續整合和持續交付 (CI/CD) 管道的步驟，該管道使用 AWS 在 AWS CodePipeline 上的實際裝置上建立和測試 iOS 應用程式。該模式使用 AWS 存放應 CodeCommit 用程式程式碼、用於建置 iOS 應用程式的 Jenkins 開放原始碼工具，以及使用 AWS Device Farm 在實際裝置上測試建置的應用程式。這三個階段是使用 AWS CodePipeline 在管道中協調的。

這個模式是基於在 AWS DevOps 部落格上使用 AWS 和行動服務建置和測試 iOS DevOps 和 iPadOS 應用程式後的文章。如需詳細指示，請參閱部落格文章。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 蘋果開發者帳戶
- 建置伺服器
- Xcode 版本 11.3（在構建服務器上安裝並設置）
- 在工作站上安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)
- Git 的基本知識

限制

- 應用程式建置伺服器必須執行 macOS。
- 組建伺服器必須具有公用 IP 位址，因此 CodePipeline 可以遠端連線到它以啟動組建。

架構

源, 技術, 堆棧

- 涉及在實體裝置上使用模擬器或手動測試的內部部署 iOS 應用程式建置程序

目標技術堆疊

- 用於存 CodeCommit 放應用程式原始碼的 AWS 儲存庫
- 使用 Xcode 構建的應用程序的詹金斯服務器
- AWS Device Farm 裝置集區，用於在實際裝置上測試應用程式

目標架構

當使用者對來源儲存庫提交變更時，管道 (AWS CodePipeline) 會從來源儲存庫擷取程式碼、啟動 Jenkins 組建，然後將應用程式程式碼傳遞給 Jenkins。建置完成後，管道會擷取建置成品，並啟動 AWS Device Farm 任務，以針對裝置集區測試應用程式。

工具

- AWS CodePipeline 是全受管的持續交付服務，可協助您自動化發行管道，以快速可靠地更新應用程式和基礎設施。 CodePipeline 每次發生程式碼變更時，都會根據您定義的發行模型，自動執行發行程序的建置、測試和部署階段。
- AWS CodeCommit 是全受管的原始檔控制服務，可託管安全的 Git 儲存庫。它可讓團隊在安全且可高度擴展的生態系統中輕鬆協作程式碼。 CodeCommit 無需操作您自己的原始檔控制系統，也不必擔心擴展其基礎架構。
- AWS Device Farm 是一種應用程式測試服務，可讓您透過廣泛的桌面瀏覽器和實際行動裝置對 Web 和行動應用程式進行測試，進而提高 Web 和行動應用程式的品質，而無需佈建和管理任何測試基礎設施。
- Jenkins 是一個開源的自動化服務器，使開發人員能夠構建，測試和部署他們的軟件。

史诗

設定建置環境

任務	描述	所需技能
在運行 macOS 的構建服務器上安裝詹金斯。	詹金斯將用於構建應用程序，所以你必須首先在構建服務器上安裝它。若要取得此項工作和後續任務的詳細指示，請參閱 AWS 部落格文章：使用 AWS 和行動服務和其他資源建置 DevOps 和測試 iOS 和 iPadOS 應用程式，請參閱此模式結尾的「相關資源」一節。	DevOps
配置詹金斯。	按照屏幕上的說明配置詹金斯。	DevOps
安裝詹金斯的 AWS CodePipeline 插件。	這個插件必須安裝在詹金斯服務器上，以便詹金斯與 AWS CodePipeline 服務進行交互。	DevOps
創建一個詹金斯自由泳項目。	在詹金斯，創建一個自由式項目。設定專案以指定觸發程序和其他組建設定選項。	DevOps

設定 AWS Device Farm

任務	描述	所需技能
建立 Device Farm 專案。	開啟 AWS 裝 Device Farm 主控台。創建一個項目和一個設備池進行測試。如需指示，請參閱部落格文章。	開發人員

設定來源儲存庫

任務	描述	所需技能
創建一個 CodeCommit 存儲庫。	創建一個存儲庫，其中源代碼將被存儲。	DevOps
將您的應用程式程式碼提交至儲存庫。	Connect 至您建立的 CodeCommit 存放庫。將代碼從本地計算機推送到存儲庫。	DevOps

配置管道

任務	描述	所需技能
在 AWS 中建立管道 CodePipeline。	開啟 AWS CodePipeline 主控台並建立管道。管線會協調 CI/CD 程序的所有階段。如需指示，請參閱 AWS 部落格文章：使用 AWS 和行動服務建置和測試 iOS DevOps 和 iPadOS 應用程式。	DevOps
將測試階段新增至管線。	若要新增測試階段並將其與 AWS Device Farm 整合，請編輯管道。	DevOps
啟動管線。	若要啟動管線和 CI/CD 程序，請選擇「發行變更」。	DevOps

檢視應用測試結果

任務	描述	所需技能
檢閱測試結果。	在 AWS Device Farm 主控台中，選取您建立的專案，然後	開發人員

任務	描述	所需技能
	檢閱測試結果。控制台將顯示每個測試的詳細信息。	

相關資源

S 此模式的tep-by-step 說明

- 使用 AWS 和行動服務建置和測試 iOS DevOps 和 iPadOS 應用程式 (AWS DevOps 部落格文章)

設定 AWS Device Farm

- AWS Device Farm 主控台

設定來源儲存庫

- 建立 AWS CodeCommit 儲存庫
- Connect 到 AWS CodeCommit 儲存庫

配置管道

- AWS CodePipeline 主控台

其他資源

- AWS CodePipeline 文件
- AWS CodeCommit 文件
- AWS Device Farm 文件
- 詹金斯文檔
- 詹金斯安 macOS
- 詹金斯的 AWS CodePipeline 插件
- 安裝
- AWS CLI 安裝和組態
- Git 文件

使用 cdk-nag 規則套件檢查 AWS CDK 應用程式或 CloudFormation 範本以取得最佳實務

創建者阿倫·唐蒂

環境：生產	技術： DevOps; 安全性，身份，合規	工作負載：開源
AWS 服務：AWS CDK		

Summary

此模式說明如何使用 cdk-nag 公用程式，透過組合使用規則套件來檢查 AWS Cloud Development Kit (AWS CDK) 應用程式的最佳實務。 cdk-nag 是一個由 cfn_nag 啟發的開源項目。它使用 AWS CDK 方面，在 AWS 解決方案程式庫、Health 保險可攜性與責任法案 (HIPAA) 和國家標準技術研究所 (NIST) 800-53 等評估套件中實作規則。您可以使用這些套件中的規則來檢查 AWS CDK 應用程式的最佳實務、根據最佳實務偵測和修復程式碼，以及隱藏您不想在評估中使用的規則。

您也可以使用 cdk-nag 使用雲端格式化包含模組來檢查 AWS CloudFormation 範本。

如需所有可用套件的相關資訊，請參閱 cdk-na g 儲存庫的「規則」一節。評估套件適用於：

- AWS 解決方案庫
- HIPAA 安全性
- 第四版
- 第五版
- 支付卡產業資料安全標準 (PCI DSS) 3.2.1

先決條件和限制

先決條件

- 使用 AWS CD K 的應用程式

工具

- AWS CDK — Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可在程式碼中定義雲端基礎設施，並透過 AWS 佈建雲端基礎設施。 CloudFormation
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，也可以一起啟動並設定它們為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。

史诗

將 CDK 與您的 AWS CDK 應用程式整合

任務	描述	所需技能
了解有關 CDG 的信息。	導航到 cdk-nag GitHub 存儲庫並通讀文檔。	應用程式開發人員
在您的 AWS CDK 應用程式中安裝 CDK 套件。	若要在 AWS CDK 應用程式中使用 cdk-nag，您必須先安裝它。 CDG-嘮叨可以從 PyPI，故宮，和阿帕奇的 Maven 下載。 NuGet如需有關可用版本和下載位置的最新資訊，請參閱存放庫中的讀我檔案。	應用程式開發人員
選擇您的 NagPacks.	cdk-nag 有不同的規則包稱為。NagPacks每個都 NagPack 包含符合特定標準的規則。例如，AWS 解決方案 NagPack 包含一般最佳實務，而 NIST 800-53 第 5 版 NagPack 可協助遵循法規。您可以將多個套用 NagPacks 至您的應用程式，也可以視需要新增和移除套件。如需可用套件的清單，請參閱 GitHub 存放庫中的讀我	應用程式開發人員

任務	描述	所需技能
	檔案。如需有關每個套件中個別規則的資訊，請參閱 GitHub 存放庫的「規則」一節。	
將 CDK 中整合到您的 AWS CDK 應用程式中。	您可以將 cdk-nag 集成到應用程序範圍內的應用程序中，或將其集成到應用程序中的單個階段或堆棧中。例如，若要將 AWS 解決方案和 HIPAA 安全性整合 NagPacks 到整個應用程式層級的 AWS CDK v2 TypeScript 應用程式，您可以使用下列程式碼： `import { App, Aspects } from 'aws-cdk-lib';` `import { CdkTestStack } from '../lib/cdk-test-stack';` `import { AwsSolutionsChecks, HIPAASecurityChecks } from 'cdk-nag';` `const app = new App();` `new CdkTestStack(app, 'CdkNagDemo');` `// Simple rule informational messages` `Aspects.of(app).add(new AwsSolutionsChecks());` `// Additional explanations on the purpose of triggered rules` `Aspects.of(app).add(new HIPAASecurityChecks({ verbose: true }));`	應用程式開發人員

相關資源

- cdk-nag 代碼存儲庫
- 建造樞紐中的 CDG-中

設定對 Amazon DynamoDB 的跨帳戶存取權

由沙希達爾米亞（AWS）和傑伊恩加莫里（AWS）創建

環境：生產	技術： DevOps; 數據庫; 安全性，身份，合規	AWS 服務：Amazon DynamoDB 支援；AWS Identity and Access Management；AWS Lambda

Summary

此模式說明設定跨帳戶存取 Amazon DynamoDB 的步驟。如果服務具有在資料庫中設定適當的 AWS 身分和存取管理 (IAM) 許可，則 Amazon Web 服務 (AWS) 服務可以存取位於相同 AWS 帳戶中的 DynamoDB 表。不過，從不同 AWS 帳戶存取需要設定 IAM 許可，並在兩個帳戶之間建立信任關係。

此模式提供步驟和範例程式碼，以示範如何在一個帳戶中設定 AWS Lambda 函數，以便在不同帳戶中讀取和寫入 DynamoDB 表格。

先決條件和限制

- 兩個作用中的 AWS 帳戶。此模式將這些帳戶稱為「帳戶 A」和「帳戶 B」。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 已安裝並設定為存取帳戶 A，以建立 DynamoDB 資料庫。此模式中的其他步驟提供如何使用 IAM、DynamoDB 和 Lambda 主控台的指示。如果您打算改用 AWS CLI，請將其設定為存取這兩個帳戶。

架構

在下圖中，AWS Lambda、Amazon EC2 和 DynamoDB 都在同一個帳戶中。在這個案例中，Lambda 函數和亞馬遜彈性運算雲端 (Amazon EC2) 執行個體可以存取 DynamoDB。

如果不同 AWS 帳戶中的資源嘗試存取 DynamoDB，則需要設定跨帳戶存取和信任關係。例如，在下圖中，若要啟用帳戶 A 中的 DynamoDB 與帳戶 B 中 Lambda 函數之間的存取權，您必須在帳戶之間建立信任關係，並將適當的存取權授予 Lambda 服務和使用者，如 E pics 一節所述。

工具

AWS 服務

- Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

Code

此模式包含「其他資訊」一節中的範例程式碼，以說明如何在帳戶 B 中設定 Lambda 函數，以便從帳戶 A 的 DynamoDB 表中寫入和讀取該程式碼。此程式碼僅供說明和測試用途。如果您要在生產環境中實作此模式，請使用程式碼做為參考，並針對您自己的環境進行自訂。

此模式說明透過 Lambda 和 DynamoDB 進行跨帳戶存取。您也可以對其他 AWS 服務使用相同的步驟，但請確保在兩個帳戶中授予和設定適當的許可。例如，如果您想要授與帳戶 A 中 Amazon Relational Database Service (Amazon RDS) 資料庫的存取權，請為該資料庫建立角色，並將其與信任關係繫結。在帳戶 B 中，如果您想要使用 Amazon EC2 而非 AWS Lambda，請建立個別的 IAM 政策和角色，然後將它們附加到 EC2 執行個體。

史诗

在帳戶 A 中建立動態資料表

任務	描述	所需技能
在帳戶 A 中建立 DynamoDB 資料表。	為帳戶 A 設定 AWS CLI 之後，請使用下列 AWS CLI 命令建立 DynamoDB 表格： `aws dynamodb create-table \` `--table-name Table-Acccount-A \`	AWS DevOps

任務	描述	所需技能
	--attribute-definitions \ Attribute Name=category,AttributeType=S \ Attribute Name=item,AttributeType=S \ --key-schema \ Attribute Name=category,KeyType=HASH \ Attribute Name=item,KeyType=RANGE \ --provisioned-throughput \ ReadCapacityUnits=5,WriteCapacityUnits=5 如需有關建立資料表的詳細資訊，請參閱 DynamoDB 文件。	

在帳戶 A 中建立角色

任務	描述	所需技能
在帳戶 A 中建立角色。	帳戶 B 將使用此角色來取得存取帳戶 A 的權限。若要建立角色，請執行下列動作： 1. 登入帳戶 A，於 https://<account-ID-for-Account-A>.signin.aws.amazon.com/console 。	AWS DevOps

任務	描述	所需技能
	2. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。 3. 在主控台的導覽窗格中，選擇 [角色]，然後選擇 [建立角色]。 4. 對於選取受信任的實體，請選擇 AWS 帳戶，然後在「AWS 帳戶」區段中選擇「其他 AWS 帳戶」。 5. 對於帳戶 ID，請輸入帳戶 B 的 ID。 6. 選擇 Next: Permissions (下一步：許可)。 7. 在 [篩選器原則] 方塊中，輸入 DynamoDB。 8. 在 DynamoDB 原則清單中，選取AmazonDynamo資料庫。FullAccess 附註：此原則允許對 DynamoDB 執行所有動作。作為安全性最佳做法，您應該永遠只授與必要的權限。如需可改為選擇的其他政策清單，請參閱 IAM 文件中的範例政策。 9. 選擇下一步：命名、複查和建立。 10.在角色名稱中，輸入角色的唯一名稱 (例如，DynamoDB FullAcces	

任務	描述	所需技能
	s-用於帳戶 B)，然後新增選用的角色描述。 11檢閱所有區段，並 (選擇性) 將標籤附加為索引鍵值配對，將中繼資料新增至角色。 12選擇建立角色。 如需建立角色的詳細資訊，請參閱 IAM 文件。	
請注意帳戶 A 中角色的 ARN。	1. 在 IAM 主控台的導覽窗格中，選擇 [角色]。 2. 在搜尋方塊中，輸入 DynamoDB FullAccess-對帳戶 B (或您在上一個內文中建立的角色名稱)，然後選擇角色。 3. 在角色的摘要頁面中，複製 Amazon 資源名稱 (ARN)。在帳戶 B 中設定 Lambda 程式碼時，您將使用 ARN。	AWS DevOps

設定從帳戶 B 存取帳戶 A

任務	描述	所需技能
建立存取帳戶 A 的策略。	1. 在下列位置登入帳戶 B `https://<account-ID-for-Account-B>.signin.aws.amazon.com/console` 。	AWS DevOps

任務	描述	所需技能
	2. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。 3. 在主控台的瀏覽窗格中，選擇 [原則]，然後選擇 [建立原則]。 4. 選擇 JSON 標籤。 5. 輸入或貼上下列 JSON 文件： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<Account-A-ID>:role/DynamoDB-FullAccess-For-Account-B" } ] }` 其中`Resource`屬性包含您在帳戶 A 的上一個故事中建立的角色的 ARN。 6. 選擇下一步：標籤。 7. (選用) 藉由連接標籤作為鍵值組，將中繼資料新增至政策。	

任務	描述	所需技能
	8. 選擇 下一步：檢閱。 9. 針對原則名稱，輸入原則的唯一名稱 (例如，DynamoDB FullAccess-帳戶內原則 A)，並新增選擇性原則說明。 10選擇建立政策。 如需建立政策的詳細資訊，請參閱 IAM 文件。	

任務	描述	所需技能
根據策略建立角色。	帳戶 B 中的 Lambda 函數會使用此角色來讀取和寫入帳戶 A 中的 DynamoDB 表格。 1. 在帳戶 B 的 IAM 主控台的導覽窗格中，選擇 [角色]，然後選擇 [建立角色]。 2. 對於 Select type of trusted entity (選取信任的實體類型)，選擇 AWS service (AWS 服務)。 3. 對於使用案例，請選擇 Lambda。 4. 選擇 Next: Permissions (下一步：許可)。 5. 在 [篩選器原則] 方塊中，輸入 DynamoDB。 6. 在 DynamoDB FullAccess原則清單中，選取您在上一個故事中建立的 DynamoDB-帳戶內政策 A。 7. 選擇下一步：命名、複查和建立。 8. 在角色名稱中，輸入角色的唯一名稱 (例如，DynamoDB FullAccess-帳戶內 A)，然後新增選用的角色描述。 9. 檢閱所有區段，並 (選擇性) 將標籤附加為索引鍵值配對，將中繼資料新增至角色。 10選擇建立角色。	AWS DevOps

任務	描述	所需技能
	您現在可以在下一個史詩中將此角色附加到 Lambda 函數。 如需建立角色的詳細資訊，請參閱 IAM 文件。	

在帳戶 B 中建立 Lambda 函數

任務	描述	所需技能
建立 Lambda 函數以將資料寫入 DynamoDB 資料。	1. 在下列位置登入帳戶 B `https://<account-ID-for-Account-B>.signin.aws.amazon.com/console` 。 2. 開啟 Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。 3. 在主控台的導覽窗格中，選擇 [函數]，然後選擇 [建立函數]。 4. 在「名稱」中輸入「寫入函數」。 5. 在「執行階段」中，選擇 Python 3.8 或更新版本。 6. 對於 [權限]，[變更預設執行角色]，選擇 [使用現有角色]。 7. 對於現有角色，請選擇帳戶內 A FullAccess。 8. 選擇 建立函式。 9. 在「程式碼」索引標籤中，貼上此模式「其他資訊」	AWS DevOps

任務	描述	所需技能
	區段中提供的 Lambda 寫入函數範例程式碼。請務必為`RoleArn`欄位提供正確的角色 ARN (從史詩般的 [在帳戶 A 中建立角色])，並變更`region_name` 為帳戶 A 中建立 DynamoDB 表格的位置 (從 [帳戶 A] 中的史詩建立 DynamoDB 表格)。如果不這樣做會導致`ResourceNotFoundException` 錯誤。 10.若要部署程式碼，請選擇 [部署]。 11.選擇 [測試] 來執行函式。這會提示您設定測試事件。使用您偏好的名稱 (例如) 建立新事件 MyTestEventForWrite，然後儲存組態。 12.通過選擇測試再次運行該功能。這將使用您提供的事件名稱運行代碼。 13.檢查函數的輸出。它應該類似於其他信息的 Lambda 寫入函數部分中顯示的輸出。此輸出表示函數存取了帳戶 A 中的 DynamoDB 表格，並且能夠向其寫入資料。 如需建立 Lambda 函數的詳細資訊，請參閱 Lambda 文件。	

任務	描述	所需技能
建立 Lambda 函數以從 DynamoDB 資料讀取資料。	1. 在 Lambda 主控台的導覽窗格中，選擇 [函數]，然後選擇 [建立函數]。 2. 在「名稱」中，輸入讀取函數。 3. 在「執行階段」中，選擇 Python 3.8 或更新版本。 4. 對於 [權限]，[變更預設執行角色]，選擇 [使用現有角色]。 5. 對於現有角色，請選擇帳戶內 A FullAccess。 6. 選擇 建立函式。 7. 在 [程式碼] 索引標籤中，貼上此模式的其他資訊區段中提供的 Lambda 讀取函數範例程式碼。請務必為`RoleArn`欄位提供正確的角色 ARN (從史詩般的 [在帳戶 A 中建立角色])，並變更`region_name` 為帳戶 A 中建立 DynamoDB 表格的位置 (從 [帳戶 A] 中的史詩建立 DynamoDB 表格)。如果不這樣做會導致`ResourceNotFoundException` 錯誤。 8. 若要部署程式碼，請選擇 [部署]。 9. 選擇 [測試] 來執行函式。這會提示您設定測試事件。使用您偏好的名稱 (例如) 建立新事件 MyTestEve	AWS DevOps

任務	描述	所需技能
	ntForRead，然後儲存組態。 10通過選擇測試再次運行該功能。這將使用您提供的事件名稱運行代碼。 11檢查函數的輸出。它應該類似於其他信息的 Lambda 讀取函數部分中顯示的輸出。此輸出表示函數存取了帳戶 A 中的 DynamoDB 表，並且能夠讀取您新增至表格的資料。 如需建立 Lambda 函數的詳細資訊，請參閱 Lambda 文件。	

清除資源

任務	描述	所需技能
刪除您建立的資源。	如果您在測試或概念驗證 (PoC) 環境中執行此模式，請刪除您建立的資源以避免產生成本。 1. 在帳戶 B 中，刪除您為連線至 DynamoDB 而建立的兩個 Lambda 函數和其他資源。 2. 在帳戶 A 中，刪除您建立的 DynamoDB 表格。 3. IAM 政策不會花費任何費用，因此您可以保持原樣。	AWS DevOps

任務	描述	所需技能
	但是，為了安全起見，我們建議您刪除為此模式建立的下列角色和策略： • 帳戶 A：對帳戶的完全訪問權限 -一個角色 • 帳戶 DynamoDB 帳戶內角色 FullAccess • 帳戶 DynamoDB 帳戶內政策 A 政策 FullAccess	

相關資源

- 開始使用 AWS CLI (AWS CLI 文件)
- 設定 AWS CLI (AWS CLI 文件)
- 開始使用 Dynamo DB 文件
- 開始使用 Lambda (AWS Lambda 文件)
- 建立角色以將許可委派給 IAM 使用者 (IAM 文件)
- 建立 IAM 政策 (IAM 文件)
- 跨帳戶政策評估邏輯 (IAM 文件)
- IAM JSON 政策元素參考資料 (IAM 文件)

其他資訊

本節中的代碼僅用於說明和測試目的。如果您要在生產環境中實作此模式，請使用程式碼做為參考，並針對您自己的環境進行自訂。

Lambda 函數

範例程式碼

```
import boto3
from datetime import datetime
```

```
sts_client = boto3.client('sts')
sts_session = sts_client.assume_role(RoleArn='arn:aws:iam::<Account-A ID>:role/
DynamoDB-FullAccess-For-Account-B', RoleSessionName='test-dynamodb-session')


KEY_ID = sts_session['Credentials']['AccessKeyId']
ACCESS_KEY = sts_session['Credentials']['SecretAccessKey']
TOKEN = sts_session['Credentials']['SessionToken']


dynamodb_client = boto3.client('dynamodb',
                               region_name='<DynamoDB-table-region-in-account-A',
                               aws_access_key_id=KEY_ID,
                               aws_secret_access_key=ACCESS_KEY,
                               aws_session_token=TOKEN)

def lambda_handler(event, context):
     now = datetime.now()
     date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
     data = dynamodb_client.put_item(TableName='Table-Acccount-A', Item={"category":
 {"S": "Fruit"},"item": {"S": "Apple"},"time": {"S": date_time}})
     return data
```

範例輸出

Lambda 取函數

範例程式碼

```
import boto3
from datetime import datetime


sts_client = boto3.client('sts')
sts_session = sts_client.assume_role(RoleArn='arn:aws:iam::<Account-A ID>:role/
DynamoDB-FullAccess-For-Account-B', RoleSessionName='test-dynamodb-session')


KEY_ID = sts_session['Credentials']['AccessKeyId']
ACCESS_KEY = sts_session['Credentials']['SecretAccessKey']
TOKEN = sts_session['Credentials']['SessionToken']
```

```
dynamodb_client = boto3.client('dynamodb',
                               region_name='<DynamoDB-table-region-in-account-A>',
                               aws_access_key_id=KEY_ID,
                               aws_secret_access_key=ACCESS_KEY,
                               aws_session_token=TOKEN)


def lambda_handler(event, context):
     response = dynamodb_client.get_item(TableName='Table-Acccount-A', Key={'category':
{'S':'Fruit'}, 'item':{'S':'Apple'}})
     return response
```

範例輸出

為在 Amazon EKS 上執行的應用程式設定相互 TLS 身份驗證

創建者：馬亨德拉·西達帕 (AWS)

環境：PoC 或試點	技術： DevOps; 安全性，身份，合規性	AWS 服務：Amazon EKS；Amazon Route 53

Summary

憑證型相互傳輸層安全性 (TLS) 是選用的 TLS 元件，可在伺服器與用戶端之間提供雙向對等驗證。使用相互 TLS 時，用戶端必須在工作階段交涉程序期間提供 X.509 憑證。伺服器會使用此憑證來識別和驗證用戶端。

相互 TLS 是物聯網（IoT）應用程序的常見要求，可用於開放銀行等 business-to-business 應用程序或標準。

此模式說明如何使用 NGINX 輸入控制器，為在 Amazon Elastic Kubernetes Service (Amazon EKS) 叢集上執行的應用程式設定相互 TLS。您可以透過註解輸入資源，為 NGINX 輸入控制器啟用內建的相互 TLS 功能。如需有關 NGINX 控制器上相互 TLS 註解的詳細資訊，請參閱 Kubernetes 文件中的用戶端憑證驗證。

重要：此模式使用自我簽署憑證。我們建議您僅在測試叢集中使用此模式，而不要在生產環境中使用。如果您想要在生產環境中使用此模式，可以使用 AWS 私有憑證授權單位 (AWS Private CA) 或現有的公開金鑰基礎設施 (PKI) 標準來發行私有憑證。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶。
- 現有 Amazon EKS 叢集。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 1.7 版或更新版本，可在 macOS、Linux 或視窗上安裝和設定。
- kubectl 命令列公用程式已安裝並設定為存取 Amazon EKS 叢集。如需這方面的詳細資訊，請參閱 Amazon EKS 文件中的安裝 kubectl。
- 用於測試應用程式的現有網域名稱系統 (DNS) 名稱。

限制

- 此模式使用自我簽署憑證。我們建議您僅在測試叢集中使用此模式，而不要在生產環境中使用。

架構

技術堆疊

- Amazon EKS
- Amazon Route 53
- 庫貝克特爾

工具

- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。
- Kubectl 是您用來與 Amazon EKS 叢集互動的命令列公用程式。

史诗

產生自我簽署憑證

任務	描述	所需技能
產生 CA 金鑰和憑證。	執行下列命令，產生憑證授權單位 (CA) 金鑰和憑證。 `openssl req -x509 -sha256 -newkey rsa:4096 -keyout ca.key -out ca.crt -days 356 -nodes -subj '/CN=Test Cert Authority'`	DevOps 工程師

任務	描述	所需技能
產生伺服器金鑰和憑證，並使用 CA 憑證簽署。	產生伺服器金鑰和憑證，並執行下列命令以 CA 憑證簽署。 `openssl req -new -newkey rsa:4096 -keyout server.key -out server.csr -nodes -subj '/CN= <your_domain_name> ' && openssl x509 -req -sha256 -days 365 -in server.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt` 重要事項：請務必以現<your_domain_name>有的網域名稱取代。	DevOps 工程師
產生用戶端金鑰和憑證，並使用 CA 憑證簽署。	產生用戶端金鑰和憑證，並執行下列命令以 CA 憑證簽署。 `openssl req -new -newkey rsa:4096 -keyout client.key -out client.csr -nodes -subj '/CN=Test' && openssl x509 -req -sha256 -days 365 -in client.csr -CA ca.crt -CAkey ca.key -set_serial 02 -out client.crt`	DevOps 工程師

部署 NGINX 入口控制器

任務	描述	所需技能
在您的 Amazon EKS 叢集中部署 NGINX 輸入控制器。	使用下列指令部署 NGINX 入口控制器。 `kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.7.0/deploy/static/provider/aws/deploy.yaml`	DevOps 工程師
確認 NGINX 輸入控制器服務正在執行。	使用下列命令確認 NGINX 入口控制器服務是否正在執行。 `kubectl get svc -n ingress-nginx` 重要：請確定服務位址欄位包含網路負載平衡器的網域名稱。	DevOps 工程師

在 Amazon EKS 叢集中建立命名空間以測試相互 TLS

任務	描述	所需技能
在 Amazon EKS 叢集中建立命名空間。	執行下列命令，建立`mtls`在 Amazon EKS 叢集中呼叫的命名空間。 `kubectl create ns mtls`	DevOps 工程師

任務	描述	所需技能
	這會部署範例應用程式以測試相互 TLS。	

建立範例應用程式的部署和服務

任務	描述	所需技能
在 mtls 命名空間中建立 Kubernetes 部署和服務。	建立名為 `mtls.yaml` 的檔案。將以下程式碼貼到檔案。 `kind: Deployment` `apiVersion: apps/v1` `metadata:` `  name: mtls-app` `  labels:` `  app: mtls` `spec:` `  replicas: 1` `  selector:` `  matchLabels:` `  app: mtls` `  template:` `  metadata:` `  labels:` `  app: mtls` `  spec:` `  containers:` `  - name: mtls-app` `  image: hashicorp/http-echo` `  args:` `  - "-text=mTLS is working"` `---` `kind: Service` `apiVersion: v1`	DevOps 工程師

任務	描述	所需技能
	`metadata:` `  name: mtls-service` `spec:` `  selector:` `  app: mtls` `  ports:` `  - port: 5678 #` `  Default port for image` 執行下列命令，在mtls命名空間中建立 Kubernetes 部署和服務。 `kubectl create -f` `  mtls.yaml -n mtls`	
確認已建立 Kubernetes 部署。	執行下列命令以確認已建立部署，並且有一個網繭處於可用狀態。 `kubectl get deploy -n` `  mtls`	DevOps 工程師
確認已建立 Kubernetes 服務。	請執行下列命令，確認是否已建立 Kubernetes 服務。 `kubectl get service -n` `  mtls`	DevOps 工程師

在 mtls 命名空間中創建一個密鑰

任務	描述	所需技能
為輸入資源建立密碼。	執行下列命令，使用您先前建立的憑證，為 NGINX 輸入控制器建立密碼。	DevOps 工程師

任務	描述	所需技能
	`kubectl create secret generic mtls-certs --from-file=tls.crt=server.crt --from-file=tls.key=server.key --from-file=ca.crt=ca.crt -n mtls` 您的密碼具有用於用戶端識別伺服器的伺服器憑證，以及伺服器用來驗證用戶端憑證的 CA 憑證。	

在 mtls 命名空間中創建輸入資源

任務	描述	所需技能
在 mtls 命名空間中建立輸入資源。	建立名為 `ingress.yaml` 的檔案。將以下代碼粘貼到文件中（替換為您現<your_domain_name> 有的域名）。 `apiVersion: networking.k8s.io/v1` `kind: Ingress` `metadata:` `  annotations:` `  nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"` `  nginx.ingress.kubernetes.io/auth-tls-secret: mtls/mtls-certs` `  name: mtls-ingress` `spec:` `  ingressClassName: nginx`	DevOps 工程師

任務	描述	所需技能
	<pre> rules: - host: "*.<your_ domain_name>" http: paths: - path: / pathType: Prefix backend: service: name: mtls- service port: number: 5678 tls: - hosts: - "*.<your_ domain_name>" secretName: mtls- certs</pre> 執行下列命令，在命mtls名空間中建立輸入資源。 <pre>kubectl create -f ingress.yaml -n mtls</pre> 這表示 NGINX 入口控制器可以將流量路由到您的範例應用程式。	

任務	描述	所需技能
確認已建立輸入資源。	執行下列命令，確認輸入資源是否已建立。 `kubectl get ing -n mtls` 重要：請確定輸入資源的位址顯示為 NGINX 輸入控制器建立的負載平衡器。	DevOps 工程師

設定 DNS 以將主機名稱指向負載平衡器

任務	描述	所需技能
建立指向 NGINX 輸入控制器之負載平衡器的 CNAME 記錄。	登入 AWS 管理主控台，開啟 Amazon Route 53 主控台，然後建立指`mtls.<your_domain_name>` 向 NGINX 輸入控制器的負載平衡器的標準名稱 (CNAME) 記錄。 如需詳細資訊，請參閱 Route 53 說明文件中的使用 Route 53 主控台建立記錄。	DevOps 工程師

測試應用程式。

任務	描述	所需技能
測試不使用憑證的相互 TLS 設定。	執行下列命令。 `curl -k https://mtls.<your_domain_name>`	DevOps 工程師

任務	描述	所需技能
	您應該會收到「400 未傳送必要的 SSL 憑證」錯誤回應。	
使用憑證測試相互 TLS 設定。	執行下列命令。 `curl -k https://mtls.<your_domain_name> --cert client.crt --key client.key` 您應該收到「MTL 正在工作」的響應。	DevOps 工程師

相關資源

- 使用 Amazon Route 53 控制台創建記錄
- 在 Amazon EKS 上使用帶有 NGINX 入口控制器的 Network Load Balancer 器
- 用戶端憑證驗證

使用 Firelens 日誌路由器為 Amazon ECS 創建自定義日誌解析器

由瓦倫·夏爾馬（AWS）創建

環境：生產	技術： DevOps；容器與微服務	工作負載：所有其他工作
AWS 服務：Amazon ECS		

Summary

火鏡是 Amazon Elastic Container Service (Amazon ECS) 和 AWS Fargate 的日誌路由器。您可以使用 Firelens 將容器日誌從 Amazon ECS 路由到 Amazon CloudWatch 和其他目的地（例如，Splunk 或相撲邏輯）。Firelens 可與 Fluentd 或 Flu ent Bit 作為記錄代理程式搭配使用，這意味著您可以使用 Amazon ECS 任務定義參數來路由日誌。

藉由選擇在來源層級剖析記錄檔，您可以分析記錄資料並執行查詢，以更有效率且有效率地回應作業問題。由於不同的應用程式具有不同的記錄模式，因此您需要使用自訂剖析器來建構記錄檔，並讓您的終端目的地的搜尋更容易。

此模式使用帶有自定義解析器的 Firelens 日誌路由器，將日誌 CloudWatch 從運行在 Amazon ECS 上的示例 Spring Boot 應用程序中推送日誌。然後，您可以使用 Amazon CloudWatch 日誌洞察，根據自訂剖析器產生的自訂欄位篩選日誌。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶。
- AWS Command Line Interface (AWS CLI) (AWS CLI)，在您的本機電腦上安裝和設定。
- Docker，在本地計算機上安裝和配置。
- Amazon Elastic Container Registry (Amazon ECR) 上現有的基於春季啟動的容器化應用程序。

架構

技術堆疊

- CloudWatch
- Amazon ECR
- Amazon ECS
- Fargate
- Docker
- Fluent Bit

工具

- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種 AWS 受管容器映像登錄服務，安全、可擴展且可靠。
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) 是可高度擴展、快速的容器管理服務，可讓您輕鬆執行、停止和管理叢集上的容器。
- AWS Identity and Access Management (IAM) — IAM 是一種用於安全控制 AWS 服務存取的 Web 服務。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- Docker — Doc ker 是用於開發，運輸和運行應用程序的開放平台。

Code

下列檔案會附加至此病毒碼：

- `customFluentBit.zip`— 包含要新增自訂剖析和組態的檔案。
- `firelens_policy.json`— 包含用於建立 IAM 政策的政策文件。
- `Task.json`— 包含 Amazon ECS 的範例任務定義。

史诗

創建自定義流利位圖像

任務	描述	所需技能
建立 Amazon ECR 儲存庫。	登入 AWS 管理主控台、開啟 Amazon ECR 主控台，然後建立名為`fluentbit_custom` 的儲存庫。 如需這方面的詳細資訊，請參閱 Amazon ECR 文件中的建立儲存庫。	系統管理員，開發者
解壓縮 customFluentBit .zip 套件。	1. 將`customFluentBit.zip` 套件（附件）下載到您的本機電腦。 2. 執行下列命令以解壓縮至`customFluentBit` 目錄：`unzip -d customFluentBit.zip` 3. 該目錄包含以下添加自定義解析和配置所需的文件： • `parsers/springboot_parser.conf` — 包含剖析器指令，並定義自訂剖析器的規則運算式 (regex) 模式。您可以為特定解析器添加正則表達式模式。 • `conf/parse_springb`	

任務	描述	所需技能
	oot.conf — 包含篩選器和服務指示詞。 • 碼頭文件	
建立自訂泊塢視窗映像檔。	1. 將目錄切換至 customFluentBit 。 2. 開啟 Amazon ECR 主控台，選擇fluentbit_custom 儲存庫，然後選擇 [檢視推送命令]。 3. 上傳您的專案。 4. 上傳完成後，複製組建的 URL。當您在 Amazon ECS 中創建一個容器時，此 URL 是必需的 如需這方面的詳細資訊，請參閱 Amazon ECR 文件中的推送碼頭映像。	系統管理員，開發者

設定 Amazon ECS 叢集

任務	描述	所需技能
建立 Amazon ECS 叢集	按照 Amazon ECS 文件中「建立叢集」中「僅限聯網範本」一節的指示，建立 Amazon ECS 叢集。 注意：請務必選擇建立 VPC 為您的 Amazon ECS 叢集建立新的虛擬私有雲端 (VPC)。	系統管理員，開發者

設定 Amazon ECS 任務

任務	描述	所需技能
設定 Amazon ECS 任務執行 IAM 角色。	使用AmazonECSTaskExecutionRolePolicy 受管政策建立 Amazon ECS 任務執行 IAM 角色。如需這方面的詳細資訊，請參閱 Amazon ECS 文件中的 Amazon ECS 任務執行 IAM 角色。 附註：請務必記錄 IAM 角色的 Amazon 資源名稱 (ARN)。	系統管理員，開發者
將身分與存取權管理政策附加到 Amazon ECS 任務執行 IAM 角色。	1. 使用 firelens_policy.json (附加) 政策文件建立 IAM 政策。如需詳細資訊，請參閱 IAM 說明文件中的「在 JSON」索引標籤上建立政策。 2. 將此政策附加到您先前建立的 Amazon ECS 任務執行 IAM 角色。如需這方面的詳細資訊，請參閱 IAM 文件中的新增 IAM 政策 (AWS CLI)。	系統管理員，開發者
設定 Amazon ECS 任務定義。	1. 更新Task.json 範例任務定義 (附加) 中的下列各節： • 更新executionRoleArn 和使taskRoleArn 用任務執行 IAM 角色的 ARN • containerDefinitions 使用您之前創建的自	系統管理員，開發者

任務	描述	所需技能
	定義 Fluent 位 Docker 映像更新中的映像 • `containerDefinitions` 使用應用程式映像檔的名稱更新中的映像 2. 開啟 Amazon ECS 主控台，選擇「任務定義」，選擇「建立新任務定義」，然後在「選取相容性」 頁面上選擇「Fargate」。 3. 選擇 [透過 Json 設定]，將更新的`Task.json` 檔案貼到文字區域，然後選擇 [儲存]。 4. 建立任務定義。 如需這方面的詳細資訊，請參閱 Amazon ECS 文件中的建立任務定義。	

執行 Amazon ECS 任務

任務	描述	所需技能
運行 Amazon ECS 任務。	在 Amazon ECS 主控台上，選擇叢集，選擇您先前建立的叢集，然後執行獨立任務。 如需有關此項目的詳細資訊，請參閱 Amazon ECS 文件中的執行獨立任務。	系統管理員，開發者

驗證記 CloudWatch 錄

任務	描述	所需技能
驗證記錄檔。	1. 開啟主 CloudWatch 控台，選擇 [記錄群組]，然後選擇`/aws/ecs/containerinsights/{{cluster_ARN}}/firelens/application` 。 2. 驗證日誌，特別是由自定義解析器添加的自定義字段。 3. 用 CloudWatch 於根據自訂欄位篩選記錄檔。	系統管理員，開發者

相關資源

- Amazon ECS 碼頭基礎知識
- AWS Fargate 上的 Amazon ECS
- 設定基本服務參數

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用和 HashiCorp 打包器創建管道 CodePipeline 和 AMI

由阿卡什·庫馬爾（AWS）創建

環境：PoC 或試點	來源： DevOps	目標：Amazon 機器映像（AMI）
R 類型：重新主機	工作負載：所有其他工作	技術: DevOps; 現代化; Web 和移動應用程序

Summary

此模式提供程式碼範例和步驟，讓您使用 AWS 在 Amazon Web Services (AWS) 雲端建立管道，以 CodePipeline 及使用 HashiCorp 封包器建立 Amazon 機器映像 (AMI)。該模式基於持續集成實踐，該實踐使用基於 Git 的版本控制系統自動構建和測試代碼。在此模式中，您可以使用 AWS 建立和複製程式碼儲存庫 CodeCommit。然後，使用 AWS 建立專案並設定原始程式碼 CodeBuild。最後，創建一個提交到您的存儲庫的 AMI。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 用於啟動 Amazon 彈性運算雲（Amazon EC2）實例的亞馬遜 Linux AMI
- HashiCorp 封裝程式 0.12.3 或更新版本
- Amazon CloudWatch 活動（可選）
- Amazon CloudWatch 日誌（可選）

架構

下圖顯示了一個應用程序代碼的示例，該代碼通過使用此模式的體系結構自動創建 AMI。

該圖顯示以下工作流程：

1. 開發人員將程式碼變更提交至私有 CodeCommit Git 儲存庫。然後， CodePipeline 用 CodeBuild 於啟動組建，並將準備好部署的新成品新增到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。
2. CodeBuild 使用封包器根據 JSON 模板捆綁和打包 AMI。如果啟用， CloudWatch 事件可以在原始程式碼中發生變更時自動啟動管線。

技術堆疊

- CodeBuild
- CodeCommit
- CodePipeline
- CloudWatch 活動（可選）

工具

- AWS CodeBuild — AWS CodeBuild 是雲端中的全受管建置服務。 CodeBuild 編譯您的原始程式碼、執行單元測試，並產生準備好部署的成品。
- AWS CodeCommit — AWS CodeCommit 是一種版本控制服務，可讓您在 AWS 雲端私有存放和管理 Git 儲存庫。 CodeCommit 您無需管理自己的原始檔控制系統，也不必擔心擴充其基礎架構。
- AWS CodePipeline — AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。
- HashiCorp Packer — HashiCorp Packer 是一種開源工具，用於從單一源配置自動創建相同的機器映像。Packer 是輕量級的，可在每個主要操作系統上運 parallel，並為多個平台並行創建機器映像。

Code

此模式包括下列附件：

- `buildspec.yml`— 此檔案用 CodeBuild 來建置和建立用於部署的成品。
- `amazon-linux_packer-template.json`— 這個文件使用打包器來創建一個 Amazon Linux AMI。

史诗

設定程式碼儲存庫

任務	描述	所需技能
建立存放庫。	創建一個 CodeCommit 存儲庫。	AWS 系統管理員
複製儲存庫。	透過複製存 CodeCommit 放庫連 Connect 至存放庫。	應用程式開發人員
將源代碼推送到遠程存儲庫。	1. 建立提交以將`buildspec.yml` 和`amazon-linux_packer-template.json` 檔案新增至您的本機儲存庫。 2. 將提交從本地存儲庫推送到遠程 CodeCommit 存儲庫。	應用程式開發人員

為應用程式建立 CodeBuild 專案

任務	描述	所需技能
建立建置專案。	1. 登入 AWS 管理主控台，開啟 AWS 主 CodeBuild 控台，然後選擇 [建立建立專案]。 2. 在「專案名稱」中，輸入專案的名稱。 3. 針對來源供應商，選擇 AWS CodeCommit。 4. 在「存放庫」中，選擇您要在其中建立程式碼管線的存放庫。	應用程式開發人員、AWS 系統管

任務	描述	所需技能
	5. 針對環境映像，選擇受管理的映像或自訂映像。 6. 針對 Operating system (作業系統)，選擇 Ubuntu。 7. 對於 RunTime（S），選擇標準。 8. 針對 Image (映像)，選擇 aws/codebuild/standard:4.0。 9. 對於映像版本，請選擇「永遠使用此執行階段版本的最新映像檔」。 10對於「環境」，請選擇 Linux。 11選擇「已授權」核取方塊。 12對於服務角色，請選擇 [新增服務角色] 或 [現有服務角色] 13對於「建置規格」，請選擇「使用 Buildspec 檔案」或「插入建置指令」。 14(選擇性) 針對「人工因素」區段中的類型，選擇無人工因素。 15(建議) 若要將組建輸出記錄檔上傳至記 CloudWatch 錄檔，請選擇CloudWatch 記錄檔。 16(選擇性) 若要將建置輸出日誌上傳到 Amazon S3，請選擇 S3 日誌核取方塊。	

任務	描述	所需技能
	17選擇 Create build project (建立建置專案)。	

設置管道

任務	描述	所需技能
管道名稱	1. 登入 AWS 管理主控台，開啟 AWS 主 CodePipeline 控台，然後選擇 [建立管道]。 2. 對於配管名稱，輸入配管的名稱。 3. 對於服務角色，請選擇 [新增服務角色] 或 [現有服務角色] 4. 針對 Role name (角色名稱)，輸入您的角色名稱。 5. 如果您希望 Amazon S3 建立儲存貯體並將成品存放在儲存貯體中，請在「進階設定」區段中選擇「預設位置」。若要使用現有的 S3 儲存貯體，請選擇 [自訂位置]。選擇下一步。 6. 針對來源供應商，選擇 AWS CodeCommit。 7. 對於存放庫名稱，請選擇您先前複製的存放庫。對於分支名稱，請選擇您的源代碼分支。 8. 對於變更偵測選項，請選擇 Amazon CloudWatch 事件 (建議使用) 以啟動管道，或	應用程式開發人員、AWS 系統管

任務	描述	所需技能
	選擇 AWS CodePipeline 定期檢查是否有變更。選擇下一步。 9. 對於建置供應商，請選擇 AWS CodeBuild。 10對於「項目名稱」，選擇您在「為應用程序創建 CodeBuild 項目」中創建的構建項目史詩。 11選擇建置選項，然後選擇 [下一步]。 12選擇 [略過部署階段]。 13選擇 Create pipeline (建立管道)。	

相關資源

- 使用 AWS 中的儲存庫 CodeCommit
- 使用組建專案
- 使用中的管線 CodePipeline

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用建立管道並將成品更新部署到現場部署 EC2 執行個體 CodePipeline

由阿卡什·庫馬爾（AWS）創建

環境：PoC 或試點	來源： DevOps	目標：亞馬遜 EC2/ 現場部署
R 類型：重新主機	技術: DevOps; 現代化; Web 和移動應用程序	AWS 服務：AWS CodeBuild；AWS CodeCommit；AWS CodeDeploy；AWS CodePipeline

Summary

此模式提供在 Amazon Web Ser vices (AWS) 雲端建立管道的程式碼範例和步驟，並將更新的成品部署到 AWS 中的現場部署 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體 CodePipeline。該模式基於持續集成實踐。這種做法會使用 Git 版本控制系統自動建置和測試程式碼。在此模式中，您可以使用 AWS 建立和複製程式碼儲存庫 CodeCommit。然後，您可以使用 AWS 建立專案並設定原始程式碼 CodeBuild。最後，您可以使用 AWS 為現場部署 EC2 執行個體建立應用程式並設定其目標環境 CodeDeploy。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 用於在部署期間識別 EC2 執行個體的使用
- CodeDeploy 代理程式, 安裝在 EC2 執行個體
- 您所需的執行階段軟體，安裝在 EC2 執行個體
- Amazon Corretto 8 的 Java 開發工具包
- 阿帕奇 Tomcat 網絡服務器，已安裝
- Amazon CloudWatch 活動（可選）
- 用於登錄 Web 服務器的 key pair（可選）
- 一個 Web 應用程序的阿帕奇 Maven 應用程序項目

架構

下圖顯示使用此模式的架構部署到現場部署 EC2 執行個體的 Java Web 應用程式範例。

該圖顯示以下工作流程：

1. 開發人員將程式碼變更提交至私有 CodeCommit Git 儲存庫。
2. CodePipeline 用 CodeBuild 於啟動建置，並新增準備在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中部署的新成品。
3. CodePipeline 使用 CodeDeploy 代理程式預先安裝部署人工因素變更所需的任何相依性。
4. CodePipeline 使用 CodeDeploy 代理程式將成品從 S3 儲存貯體部署到目標 EC2 執行個體。如果啟用 ， CloudWatch 事件可以在原始程式碼中發生變更時自動啟動管線。

技術, 堆

- CodeBuild
- CodeCommit
- CodeDeploy
- CodePipeline
- CloudWatch 活動（ 可選 ）

工具

- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。 CodeBuild 編譯您的原始程式碼、執行單元測試，並產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 將部署 CodeDeploy自動化到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。

Code

此模式包括下列附件：

- `buildspec.yml`— 此檔案指定建置和建立部署加工品所 CodeBuild 需的動作。
- `appspec.yml`— 此檔案指定為現場部署 EC2 執行個體建立應用程式和設定目標環境所 CodeDeploy 需的動作。
- `install_dependencies.sh`— 這個檔案會為 Apache 的 Tomcat 網頁伺服器安裝相依性。
- `start_server.sh`— 此檔案會啟動 Apache 的 Tomcat 網頁伺服器。
- `stop_server.sh`— 這個檔案會停止阿帕奇 Tomcat 網頁伺服器。

史诗

設定程式碼儲存庫

任務	描述	所需技能
建立存放庫。	創建一個 CodeCommit 存儲庫。	AWS 系統管理員
複製儲存庫。	透過複製存 CodeCommit 放庫連 Connect 至存放庫。	應用程式開發人員
將源代碼推送到遠程存儲庫。	1. 建立提交以將`buildspec.yml` 和`appspec.yml` 檔案新增至您的本機儲存庫。 2. 將提交從本地存儲庫推送到遠程 CodeCommit 存儲庫。	應用程式開發人員

為應用程式建立 CodeBuild 專案

任務	描述	所需技能
建立建置專案。	1. 登入 AWS 管理主控台，開啟 AWS 主 CodeBuild 控台，然後選擇 [建立建立專案]。	AWS 管理員、應用程式開發

任務	描述	所需技能
	2. 在「專案名稱」中，輸入專案的名稱。 3. 對於來源供應商，請選擇 AWS CodeCommit。 4. 在「存放庫」中，選擇您要在其中建立程式碼管線的存放庫。 5. 針對環境映像，選擇受管理的映像或自訂映像。 6. 針對 Operating system (作業系統)，請選擇 Amazon Linux 2。 7. 對於 RunTime（S），選擇標準。 8. 對於圖像，請選擇 AWS/代碼生成器/亞馬遜鏈 2-aarch64 標準：2.0。 9. 對於映像版本，請選擇「永遠使用此執行階段版本的最新映像檔」。 10對於服務角色，請選擇 [新增服務角色] 或 [現有服務角色] 11對於「建置規格」，請選擇「使用 Buildspec 檔案」或「插入建置指令」。 12(選擇性) 選擇新增人工因素以設定人工因素。 13(選擇性) 若要將建置輸出日誌上傳到 Amazon CloudWatch，請選擇CloudWatch 記錄。	

任務	描述	所需技能
	14選擇 Create build project (建立建置專案)。	

針對現場部署 EC2 執行個體設定成品

任務	描述	所需技能
建立應用程式。	1. 登入 AWS 管理主控台，開啟 AWS 主控 CodeDeploy 台，然後選擇 [建立應用程式]。 2. 在應用程式名稱中，輸入應用程式的名稱。 3. 對於運算平台，請選擇 EC2/內部部署。 4. 選擇建立應用程式，然後選擇建立部署群組。 5. 在部署群組名稱中，輸入名稱。 6. 建立的服務角色 CodeDeploy。附註：服務角色必須具有授與目標環境 CodeDeploy 存取權的權限。 7. 對於服務角色，請選擇您在步驟 6 中建立的服務角色。 8. 對於部署類型，請根據您的業務需求選擇就地或藍/綠。 9. 對於環境組態，請選擇符合您業務需求的選項。 10(選擇性) 在 Amazon EC2 主控台中分別為負載平衡器建立目標群組，然後返回	AWS 系統管理員、應用程式開發

任務	描述	所需技能
	AWS 主控 CodeDeploy 台的建立部署群組頁面以選擇負載平衡器和目標群組。 11選擇 Create deployment group (建立部署群組)。	

設置管道

任務	描述	所需技能
建立管線。	1. 登入 AWS 管理主控台，開啟 AWS 主 CodePipeline 控台，然後選擇 [建立管道]。 2. 對於配管名稱，輸入配管的名稱。 3. 對於服務角色，請選擇 [新增服務角色] 或 [現有服務角色] 4. 針對 Role name (角色名稱)，輸入您的角色名稱。 5. 如果您希望 Amazon S3 建立儲存貯體並將成品存放在儲存貯體中，請在「進階設定」區段中選擇「預設位置」。若要使用現有的 S3 儲存貯體，請選擇 [自訂位置]。選擇下一步。 6. 對於來源供應商，請選擇 AWS CodeCommit。 7. 對於存放庫名稱，請選擇您先前複製的存放庫。對於分支名稱，請選擇您的源代碼分支。	AWS 系統管理員、應用程式開發

任務	描述	所需技能
	8. 對於變更偵測選項，請選擇 Amazon CloudWatch 事件 (建議使用) 或 AWS CodePipeline。選擇下一步。 9. 對於建置供應商，請選擇 AWS CodeBuild。 10.在 [專案名稱] 中，選擇您在 [為此模式的應用程式建立 CodeBuild 專案] 區段中建立的建置專案。 11選擇您的建置選項，然後選擇 [下一步]。 12對於部署供應商，請選擇 AWS CodeDeploy。 13選擇應用程式名稱和部署群組，然後選擇 [下一步]。 14選擇 Create pipeline (建立管道)。	

相關資源

- 使用 AWS 中的儲存庫 CodeCommit
- 使用組建專案
- 使用中的應用程式 CodeDeploy
- 使用中的管線 CodePipeline

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

自動為 Java 和 Python 項目創建動態 CI 管道

創建者：芳香拉吉·傑亞拉揚（AWS），阿馬納特·雷迪（AWS），馬赫什·拉格南（AWS）和維傑亞庫馬蘭奈爾（AWS）

代碼存儲庫：automated-ci-pipeline-creation	環境：PoC 或試點	技術： DevOps; 基礎架構; 無伺服器; 雲端原生
工作負載：所有其他工作	AWS 服務：AWS CodeBuild；AWS CodePipeline；AWS Lambda；AWS Step Functions；AWS CodeCommit	

Summary

此模式示範如何使用 AWS 開發人員工具自動為 Java 和 Python 專案建立動態持續整合 (CI) 管道。

隨著技術堆棧多樣化和開發活動的增加，創建和維護整個組織中一致的 CI 管道可能變得很困難。透過自動化 AWS Step Functions 中的程序，您可以確保 CI 管道的用法和方法保持一致。

為了自動創建動態 CI 管道，此模式使用以下變量輸入：

- 程式語言 (僅適用於 Java 或 Python)
- 管道名稱
- 所需的管道階段

注意：Step Functions 使用多個 AWS 服務協調管道建立。如需此解決方案中使用之 AWS 服務的詳細資訊，請參閱此模式的「工具」一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 位於部署此解決方案的相同 AWS 區域中的 Amazon S3 儲存貯體
- AWS Identity and Access Management (IAM) 主體，具有建立此解決方案所需資源所需的 AWS CloudFormation 許可

限制

- 此模式僅支持 Java 和 Python 項目。
- 以此模式佈建的 IAM 角色遵循最低權限原則。IAM 角色的許可必須根據 CI 管道需要建立的特定資源進行更新。

架構

目標技術堆疊

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- IAM
- Amazon Simple Storage Service (Amazon S3)
- AWS Systems Manager
- AWS Step Functions
- AWS Lambda
- Amazon DynamoDB

目標架構

下圖顯示使用 AWS 開發人員工具自動為 Java 和 Python 專案建立動態 CI 管道的範例工作流程。

該圖顯示以下工作流程：

1. AWS 使用者以 JSON 格式提供 CI 管道建立的輸入參數。此輸入會啟動 Step Functions 工作流程 (狀態機器)，該工作流程會使用 AWS 開發人員工具建立 CI 管道。

2. Lambda 函數會讀取名為輸入參考的資料夾，該資料夾存放在 Amazon S3 儲存貯體中，然後產生建置規格 .yml 檔案。此產生的檔案定義了 CI 管道階段，並存放回存放參數參考的同一個 Amazon S3 儲存貯體中。
3. Step Functions 會檢查 CI 管道建立工作流程的相依性是否有任何變更，並視需要更新相依性堆疊。
4. Step Functions 在 CloudFormation 堆棧中創建 CI 管道資源，包括 CodeCommit 存儲庫，CodeBuild 項目和 CodePipeline 管道。
5. 該 CloudFormation 堆棧將所選技術堆棧（Java 或 Python）的示例源代碼和構建規格 .yml 文件複製到存儲庫中。 CodeCommit
6. CI 管線執行階段詳細資料會儲存在 DynamoDB 資料表中。

自動化和規模

- 此模式僅適用於單一開發環境。在多個開發環境中使用需要變更組態。
- 若要新增對多個 CloudFormation 堆疊的支援，您可以建立其他 CloudFormation 範本。如需詳細資訊，請參閱 CloudFormation 文件 CloudFormation中的 AWS 入門。

工具

工具

- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Systems Manager Parameter Store 為組態資料管理和機密管理提供安全的階層式儲存。

Code

此模式的代碼可在 GitHub automated-ci-pipeline-creation存儲庫中找到。存放庫包含建立此 CloudFormation 模式概述之目標架構所需的範本。

最佳實務

- 請勿將憑證 (密碼) (例如權杖或密碼) 直接輸入 CloudFormation 範本或 Step Functions 動作設定中。如果這樣做，資訊將顯示在 DynamoDB 記錄中。而是使用 AWS Secrets Manager 來設定和存放機密。然後，視需要參考儲存在 Secrets Manager 中的密碼 CloudFormation 範本和 Step Functions 式動作組態中。如需詳細資訊，請參閱秘密管理員文件中的 AWS 秘密管理員是什麼。
- 為存放在 Amazon S3 中的 CodePipeline 成品設定伺服器端加密。如需詳細資訊，請參閱 CodePipeline 文件中的為 Amazon S3 中存放的成品設定 CodePipeline伺服器端加密。
- 設定 IAM 角色時套用最低權限許可。如需詳細資訊，請參閱 IAM 文件中的套用最低權限許可。
- 請確定您的 Amazon S3 儲存貯體無法公開存取。如需詳細資訊，請參閱 Amazon S3 文件中的設定 S3 儲存貯體的區塊公共存取設定。
- 請確定您已啟用 Amazon S3 儲存貯體的版本控制。如需詳細資訊，請參閱 Amazon S3 文件中的在 S3 儲存貯體中使用版本控制。
- 設定 IAM 政策時，請使用 IAM 存取分析器。該工具提供可操作的建議，以幫助您編寫安全且功能齊全的 IAM 政策。如需詳細資訊，請參閱 IAM 文件中的使用 AWS Identity and Access Management 存取分析器。
- 如果可能，請在設定 IAM 政策時定義特定的存取條件。
- 啟用 Amazon CloudWatch 日誌記錄以進行監控和稽核。如需詳細資訊，請參閱什麼是 Amazon CloudWatch 日誌？ 在文 CloudWatch 檔中。

史诗

設定必要條件

任務	描述	所需技能
建立 Amazon S3 儲存貯體。	建立 Amazon S3 儲存貯體 (或使用現有儲存貯體) 來存放解決方案所需的 CloudFormation 範本、原始程式碼和輸入檔案。 如需詳細資訊，請參閱 Amazon S3 文件中的步驟 1：建立您的第一個 S3 儲存貯體。 注意：Amazon S3 儲存貯體必須位於部署解決方案的相同 AWS 區域。	AWS DevOps
克隆存 GitHub 儲庫。	在終端機視窗中執行下列命令來複製 GitHub automated-ci-pipeline-creation儲存庫： `git clone https://github.com/aws-samples/automated-ci-pipeline-creation.git` 如需詳細資訊，請參閱 GitHub 文件中的複製存放庫。	AWS DevOps
將解決方案範本資料夾從複製的 GitHub 儲存庫上傳到 Amazon S3 儲存貯體。	從複製的解決方案範本資料夾複製內容，並將其上傳到您建立的 Amazon S3 儲存貯體。 如需詳細資訊，請參閱 Amazon S3 文件中的上傳物件。	AWS DevOps

任務	描述	所需技能
	注意：請確定您只上傳解決方案範本資料夾的內容。您只能在 Amazon S3 儲存貯體的根層級上傳檔案。	

部署解決方案

任務	描述	所需技能
使用複製的存放庫中的 template.yml 檔案，建立 CloudFormation 堆疊以部署解決方案。 GitHub	1. 登入 AWS 管理主控台，然後開啟 AWS CloudFormation 主控台。 2. 選擇建立堆疊。會出現一個下拉式清單。 3. 在下拉式清單中，選取 [使用新資源 (標準)]。建立堆疊」頁面隨即開啟。 4. 在「指定範本」區段中，選取「上載範本檔案」旁邊的核取方塊。 5. 選取 Choose file (選擇檔案)。然後，導覽至複製的 GitHub 存放庫的根資料夾，並選取 template. yml 檔案。然後選擇 Open (開啟)。 6. 選擇下一步。隨即開啟 [指定堆疊詳細資料] 頁 7. 在「參數」段落中，指定下列參數： • 對於 S3 TemplateBucketName，請輸入您先前建立的 Amazon S3 儲存貯體名稱，其中包含此	AWS 管理員，AWS DevOps

任務	描述	所需技能
	解決方案的原始程式碼和參考資料。請確定值區名稱參數是小寫的。 • 在動 DynamoDBTable 中，輸入堆疊所建立的 DynamoDB 表格名稱。CloudFormation • 在中 StateMachineName，輸入 CloudFormation 堆疊所建立之「Step Functions」狀態機器的名稱。 8. 選擇下一步。[設定堆疊選項] 頁面隨即開啟。 9. 在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。請勿變更任何預設值。「複查」頁面隨即開啟。 10. 檢閱堆疊建立設定。然後，選擇「創建堆棧」以啟動堆棧。 注意：建立堆疊時，堆疊會列在「堆疊」頁面上，狀態為「建立 _ IN_PROGRESS」。在完成此模式中的剩餘步驟之前，請確定您等待堆疊的狀態變更為 CREATE_COMPLETE。	

測試設定

任務	描述	所需技能
執行您建立的步驟函式。	1. 登入 AWS 管理主控台，然後開啟 Step Functions 主控台。 2. 開啟您建立的步驟函數。 3. 選擇 Start execution (開始執行)。然後，以 JSON 格式輸入工作流程的輸入值 (請參閱下列範例輸入)。 4. 選擇 Start execution (開始執行)。 格式化 `{ "details": { "tech_stack": "Name of the Tech Stack (python/java)", "project_name": "Name of the Project that you want to create with", "pre_build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "build": "Choose the step if it required in the buildspec.yml file i.e., yes/no", "post_build": "Choose the step if it required in the buildspec.yml file i.e., yes/no",`	AWS 管理員，AWS DevOps

任務	描述	所需技能

```
    "reports": "Choose
 the step if it required
 in the buildspec.yml
 file i.e., yes/no",
  }
}
```

輸入範例

```
{
  "details": {
    "tech_stack":
 "java",
    "project_name":
 "pipeline-java-pjt",
    "pre_build": "yes",
    "build": "yes",
    "post_build":
 "yes",
    "reports": "yes"
  }
}
```

Python 入示例

```
{
  "details": {
    "tech_stack":
 "python",
    "project_name":
 "pipeline-python-p
jt",
    "pre_build": "yes",
    "build": "yes",
    "post_build":
 "yes",
    "reports": "yes"
  }
}
```

任務	描述	所需技能
確認已建立 CI 管線的 CodeCommit 存放庫。	1. 登入 AWS 管理主控台，然後開啟主CodeCommit 控台。 2. 在「存放庫」頁面上，確認您建立的 CodeCommit 存放庫名稱是否顯示在儲存庫清單中。存儲庫的名稱附加以下內容：pipeline-java-pjt- Repo 3. 打開 CodeCommit 存儲庫並驗證示例源代碼以及 buildspec.yml 文件是否被推送到主分支。	AWS DevOps
檢查項 CodeBuild 目資源。	1. 登入 AWS 管理主控台，然後開啟主CodeBuild 控台。 2. 在 [建置專案] 頁面上，確認您建立的 CodeBuild 專案名稱是否顯示在專案清單中。該項目的名稱附加以下內容：pipeline-java-pjt- Build 3. 選取 CodeBuild 專案名稱以開啟專案。然後，檢閱並驗證下列組態： • 專案組態 • 來源 • Environment (環境) • 建置規格 • Batch 設定 • 文物	AWS DevOps

任務	描述	所需技能
驗證階 CodePipeline 段。	1. 登入 AWS 管理主控台，然後開啟主CodePipeline 控台。 2. 在「配管」(Pipeline) 頁面上，確認您建立的配管名稱是否出現在配管清單中。管道的名稱附加以下內容：pipeline-java-pjt- 管線 3. 選取要開啟配管的配管名稱。然後，檢閱並驗證管道的每個階段，包括提交和部署。	AWS DevOps
確認 CI 管線已成功執行。	1. 在CodePipeline 主控台的「配管」頁面上，選取要檢視管線狀態的管線名稱。 2. 確認管線的每個階段都有「成功」狀態。	AWS DevOps

清除您的資源

任務	描述	所需技能
刪除中的資源堆疊 CloudFormation。	在中刪除 CI 管道的資源堆棧 CloudFormation。 如需詳細資訊，請參閱 CloudFormation 文件中的刪除 AWS CloudFormation 主控台上的堆疊。 注意：請確保刪除名為 -stack 的堆棧<project_name>。	AWS DevOps

任務	描述	所需技能
在 Amazon S3 和 CloudFormation. 中刪除 CI 管道的依賴關係。	1. 清空名為的 Amazon S3 存儲桶DeploymentArtifactBucket。如需詳細資訊，請參閱 Amazon S3 文件中的清空儲存貯體。 2. 在中刪除 CI 管道的依賴關係堆棧 CloudFormation。如需詳細資訊，請參閱 CloudFormation 文件中的刪除 AWS CloudFormation 主控台上的堆疊。 注意：請務必刪除名為的堆疊pipeline-creation-dependencies-stack。	AWS DevOps
刪除 Amazon S3 範本儲存貯體。	刪除您在此模式的 [設定必要條件] 區段中建立的 Amazon s3 儲存貯體，該模式會儲存此解決方案的範本。 如需詳細資訊，請參閱 Amazon S3 文件中的刪除儲存貯體。	AWS DevOps

相關資源

- 建立使用 Lambda 的 Step Functions 狀態機器 (AWS Step Functions 文件)
- AWS Step Functions WorkFlow 工作室 (AWS Step Functions 文件)
- DevOps 和 AWS
- AWS 如何 CloudFormation 運作？ (AWS CloudFormation 文件)
- 使用 AWS CodeCommit、AWS、AWS 和 AWS 完整的 CI/CD CodePipeline (AWS 部落格文章) CodeBuild CodeDeploy

- IAM 和 AWS STS 配額、名稱要求和字元限制 (IAM 文件)

使用地形部署 CloudWatch Synthetics 金絲雀

由德魯巴約提穆克吉 (AWS) 和讓·弗朗索瓦·蘭德羅 (AWS) 創建

代碼存儲庫：使用地形部署 CloudWatch Synthetics 金絲雀	環境：生產	技術： DevOps; 業務生產力; 軟件開發和測試; 基礎架構; Web 和移動應用
AWS 服務：Amazon CloudWatch; Amazon S3; Amazon SNS; Amazon VPC; AWS Identity and Access Management		

Summary

重要的是要從客戶的角度驗證系統的健康狀態，並確認客戶能夠連接。當客戶不經常呼叫端點時，這會更加困難。Amazon CloudWatch Synthetics 支援建立金絲雀，可同時測試公有端點和私有端點。通過使用 Canary，即使系統不在使用中，您也可以知道系統的狀態。這些金絲雀是要么 Node.js 木偶腳本或 Python 硒腳本。

此病毒碼描述如何使用 HashiCorp Terraform 部署測試私有端點的金絲雀。它嵌入了測試 URL 是否返回的木偶腳本。200-OK然後，Terraform 指令碼可與部署私有端點的指令碼整合。您也可以修改解決方案以監控公用端點。

先決條件和限制

前提

- 具有虛擬私有雲 (VPC) 和私有子網路的有效亞馬遜網路服務 (AWS) 帳戶
- 可從私有子網路連線的端點 URL
- 安裝在部署環境中的地形

限制

當前的解決方案適用於以下 CloudWatch Synthetics 運行時版本：

- syn-nodejs-puppeteer-3.4
- syn-nodejs-puppeteer-3.5
- syn-nodejs-puppeteer-3.6
- syn-nodejs-puppeteer-3.7

隨著新的執行階段版本發佈，您可能需要更新目前的解決方案。您還需要修改解決方案以跟上安全性更新。

產品版本

- 地形

架構

Amazon CloudWatch Synthetics 是基於 CloudWatch Lambda 和 Amazon Simple Storage Service (Amazon S3)。Amazon CloudWatch 提供了一個嚮導來創建金絲雀和一個顯示初期測試運行狀態的儀表板。Lambda 函數會執行指令碼。Amazon S3 會存放來自初期測試執行的日誌和螢幕擷取畫面。

此模式透過目標子網路中部署的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體模擬私有端點。Lambda 函數需要在部署私有端點的 VPC 中彈性網路介面。

上圖顯示以下項目：

1. Synthetics 金絲雀啟動金絲雀 Lambda 函數。
2. 金絲雀 Lambda 函數連接到 elastic network interface。
3. 初期測試 Lambda 函數會監控端點的狀態。
4. Synthetics 金絲雀將運行數據推送到 S3 存儲桶和 CloudWatch 指標。
5. 系統會根據指標啟動 CloudWatch 警示。
6. CloudWatch 警示會啟動 Amazon Simple Notification Service (Amazon SNS) 主題。

工具

AWS 服務

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。

- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。此病毒碼使用 VPC 端點和彈性網路介面。

其他服務

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。此模式使用 Terraform 來部署基礎結構。
- 木偶是一個 Node.js 文件庫。S CloudWatch ynthetics 運行時使用木偶框架。

Code

該解決方案可在 GitHub 雲 watch-synthetics-canary-terraform存儲庫中找到。如需詳細資訊，請參閱其他資訊一節。

史诗

實作監控私人 URL 的解決方案

任務	描述	所需技能
收集監視私人 URL 的需求。	收集完整的 URL 定義：網域、參數和標頭。若要私下與 Amazon S3 和 Amazon 通訊 CloudWatch，請使用 VPC 端點。請注意端點如何存取 VPC 和子網路。考慮金絲雀運行的頻率。	雲端架構師、網路管理員
修改現有解決方案以監視私人 URL。	修改`terraform.tfvars` 檔案：	雲端架構師

任務	描述	所需技能
	• name-你的金絲雀的名字。 • runtime_version — 金絲雀的執行階段版本。我們建議使用 syn-nodejs-puppeteer -3.7。 • take_screenshot — 是否應該截取屏幕截圖。 • api_hostname — 受監控端點的主機名稱。 • api_path— 受監控端點的路徑。 • vpc_id— 初期測試 Lambda 函數所使用的 VPC ID。 • subnet_ids — 初期測試 Lambda 函數所使用的子網路 ID。 • frequency — 金絲雀的運行頻率（以分鐘為單位）。 • alert_sns_topic — CloudWatch 警示通知傳送至的 SNS 主題。	

任務	描述	所需技能
部署和操作解決方案。	若要部署解決方案，請執行下列動作： 1. 從開發環境中的`cloudwatch-synthetics-canary-terraform` 目錄中，初始化 Terraform。 `terraform init` 2. 規劃並檢閱變更。 `terraform plan` 3. 部署解決方案。 `terraform apply`	雲端架構師、 DevOps 工程師

故障診斷

問題	解決方案
已佈建資源的刪除會卡住。	以該順序手動刪除初期測試 Lambda 函數、對應的 elastic network interface 和安全群組。

相關資源

- 使用綜合監測
- 使用 Amazon CloudWatch Synthetics 監控 API Gateway 端點 (部落格文章)

其他資訊

儲存庫成品

存放庫人工因素的結構如下。

```
.
### README.md
### main.tf
### modules
#   ### canary
#   ### canary-infra
### terraform.tfvars
### tf.plan
### variable.tf
```

該main.tf文件包含核心模塊，並部署兩個子模塊：

- canary-infra部署金絲雀所需的基礎架構。
- canary部署金絲雀。

解決方案的輸入參數位於terraform.tfvars檔案中。您可以使用下面的代碼示例來創建一個 Canary。

```
module "canary" {
    source = "./modules/canary"
    name   = var.name
    runtime_version = var.runtime_version
    take_screenshot = var.take_screenshot
    api_hostname = var.api_hostname
    api_path = var.api_path
    reports-bucket = module.canary_infra.reports-bucket
    role = module.canary_infra.role
    security_group_id = module.canary_infra.security_group_id
    subnet_ids = var.subnet_ids
    frequency = var.frequency
    alert_sns_topic = var.alert_sns_topic
}
```

相應的 .var 檔案如下。

```
name   = "my-canary"
runtime_version = "syn-nodejs-puppeteer-3.7"
take_screenshot = false
api_hostname = "mydomain.internal"
```

```
api_path = "/path?param=value"
vpc_id = "vpc_id"
subnet_ids = ["subnet_id1"]
frequency = 5
alert_sns_topic = "arn:aws:sns:eu-central-1:111111111111:yyyyy"
```

清理解決方案

如果您要在開發環境中測試此問題，則可以清理解決方案以避免產生成本。

1. 在 AWS 管理主控台上，導覽至 Amazon S3 主控台。清空解決方案建立的 Amazon S3 儲存貯體。如果需要，請確保備份數據。
2. 在您的開發環境中，從`cloudwatch-synthetics-canary-terraform`目錄中執行`destroy`命令。

```
terraform destroy
```

在 Amazon ECS 上部署適用於 Java 微服務的 CI/CD 管道

由維傑·湯普森（AWS）和桑卡爾桑格博特拉（AWS）創建

環境：PoC 或試點	技術： DevOps；容器與微服務	AWS 服務：AWS CodeBuild；Amazon EC2 容器註冊表；Amazon ECS；AWS Fargate；AWS CodePipeline

Summary

此模式會引導您完成使用 AWS 在現有 Amazon 彈性容器服務 (Amazon ECS) 叢集上為 Java 微服務部署持續整合和持續交付 (CI/CD) 管道的步驟。 CodeBuild當開發人員提交更改時，會啟動 CI/CD 管道，並在中啟動構建過程。 CodeBuild構建完成後，成品將被推送到 Amazon Elastic Container Registry (Amazon ECR)，並從 Amazon ECR 獲取最新構建並推送到 Amazon ECS 服務。

先決條件和限制

先決條件

- 在 Amazon ECS 上執行的現有 Java 微服務應用程式
- 熟悉 AWS CodeBuild 和 AWS CodePipeline

架構

源, 技術, 堆棧

- 在 Amazon ECS 上運行的 Java 微服務
- Amazon ECR 中的代碼存儲庫
- AWS Fargate

來源架構

目標技術堆疊

- Amazon ECR
- Amazon ECS
- AWS Fargate
- AWS CodePipeline
- AWS CodeBuild

目標架構

自動化和規模

CodeBuild `buildspec.yml`檔案:

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws --version
      - $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
      - REPOSITORY_URI=$AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO
      - COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
      - IMAGE_TAG=build-$(echo $CODEBUILD_BUILD_ID | awk -F":" '{print $2}')
  build:
    commands:
      - echo Build started on `date`
      - echo building the Jar file
      - mvn clean install
      - echo Building the Docker image...
      - docker build -t $REPOSITORY_URI:$BUILD_TAG .
      - docker tag $REPOSITORY_URI:$BUILD_TAG $REPOSITORY_URI:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker images...
      - docker push $REPOSITORY_URI:$BUILD_TAG
```

```
      - docker push $REPOSITORY_URI:$IMAGE_TAG
      - echo Writing image definitions file...
      - printf '[{"name":"%s","imageUri":"%s"}]' $DOCKER_CONTAINER_NAME
 $REPOSITORY_URI:$IMAGE_TAG > imagedefinitions.json
      - cat imagedefinitions.json
artifacts:
    files:
     - imagedefinitions.json
     - target/DockerDemo.jar
```

工具

AWS 服務

- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。AWS 可持續 CodeBuild 擴展並同時處理多個組建，因此您的組建不會留在佇列中。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。您可以將 AWS CodePipeline 與第三方服務整合 GitHub，或使用 AWS 服務 (例如 AWS CodeCommit 或 Amazon ECR)。
- Amazon Elastic Container Registry (Amazon ECR) 是全受管的登錄，可讓開發人員輕鬆存放、管理和部署 Docker 容器映像。Amazon ECR 與 Amazon ECS 整合，以簡化您的 development-to-production 工作流程。Amazon ECR 在高可用性和可擴展的架構中託管您的映像，因此您可以可靠地為應用程式部署容器。與 AWS Identity and Access Management (IAM) 整合可提供每個儲存庫的資源層級控制。
- Amazon Elastic Container Service (Amazon ECS) 可高度擴展、高效能的容器協調服務，支援 Docker 容器，並可讓您在 AWS 上輕鬆執行和擴展容器化應用程式。Amazon ECS 無需安裝和操作自己的容器協調軟體、管理和擴展虛擬機器叢集，或在這些虛擬機器上排程容器。
- AWS Fargate 是 Amazon ECS 的運算引擎，可讓您執行容器，而不必管理伺服器或叢集。使用 AWS Fargate，您不再需要佈建、設定和擴展虛擬機器叢集來執行容器。這樣一來即無須選擇伺服器類型、決定何時擴展叢集，或最佳化叢集壓縮。

其他工具

- Docker 是一個平台，可讓您在稱為容器的套件中建置、測試和交付應用程式。
- Git 是一個分佈式版本控制系統，用於在軟件開發過程中跟踪源代碼的變化。它是專為協調程序員之間的工作，但它可以用來跟踪任何一組文件的變化。其目標包括速度、資料完整性，以及對分散式非線性工作流程的支援。您也可以使用 AWS CodeCommit 作為 Git 的替代方案。

史诗

在 AWS 中設定建置專案 CodeBuild

任務	描述	所需技能
建立 CodeBuild 建置專案。	在 AWS 主 CodeBuild 控台中，建立建置專案並指定其名稱。	應用程式開發人員、AWS 系統管
選取來源。	此模式使用 Git 作為代碼存儲庫，因此請GitHub 從可用選項列表中進行選擇。選擇一個公共存儲庫或從您的 GitHub 帳戶。	應用程式開發人員、AWS 系統管
選取儲存庫。	選取您要從中建置程式碼的儲存庫。	應用程式開發人員、AWS 系統管
選取環境。	您可以從受管理映像清單中選取，或使用 Docker 選擇自訂映像檔。此模式使用下列受管理的映像檔： • Amazon Linux 2 • 執行階段：標準 • 圖片版本 1.0	應用程式開發人員、AWS 系統管
選擇服務角色。	您可以建立服務角色，或從現有角色清單中選取。	應用程式開發人員、AWS 系統管
新增環境變數。	在「其他組態」區段中，設定下列環境變數： • 預設 AWS 區域的 AW_ 預設區域 • 使用者帳號的 AW_ 帳戶識別碼	應用程式開發人員、AWS 系統管

任務	描述	所需技能
	• 適用於 Amazon ECR 私有儲存庫 • 構建版本的 BUILD_TAG（最新版本是此變量的值） • 工作中容器名稱的「碼頭 _ 容器名稱」 這些變量是buildspec .yml 文件中的佔位符，並將與它們各自的值替換。	
創建一個構建規格文件。	您可以在pom.xml與此模式中提供的配置相同的位置創建一個buildspec.yml 文件，或者使用在線 buildspec 編輯器並添加配置。依照提供的步驟，以適當的值設定環境變數。	應用程式開發人員、AWS 系統管
設定專案的人工因素。	(選擇性) 視需要設定成品的建置專案。	應用程式開發人員、AWS 系統管
設定 Amazon CloudWatch 日誌。	(選擇性) 視需要為建置專案設定 Amazon CloudWatch 日誌。此步驟為選用步驟，但建議使用。	應用程式開發人員、AWS 系統管
設定 Amazon S3 日誌。	(選用) 如果您想要存放日誌，請為建置專案設定 Amazon 簡單儲存服務 (Amazon S3) 日誌。	應用程式開發人員、AWS 系統管

在 AWS 中設定管道 CodePipeline

任務	描述	所需技能
建立管道。	在 AWS 主 CodePipeline 控台上，建立管道並指定其名稱。如需建立管道的詳細資訊，請參閱 AWS CodePipeline 文件。	應用程式開發人員、AWS 系統管
選取服務角色。	建立服務角色，或從現有服務角色清單中選取。如果您 CodePipeline 要建立服務角色，請提供角色的名稱，然後選取建立角色的選項。	應用程式開發人員、AWS 系統管
選擇人工因素商店。	在進階設定中，如果您希望 Amazon S3 建立儲存貯體並將成品存放在儲存貯體中，請使用成品存放區的預設位置。或者，選取自訂位置並指定現有值區。您也可以選擇使用加密金鑰來加密成品。	應用程式開發人員、AWS 系統管
指定來源提供者。	對於來源提供者，請選擇 GitHub (版本 2)。	應用程式開發人員、AWS 系統管
選取程式碼的儲存庫和分支。	如果您尚未登入，請提供要連線的連線詳細資訊 GitHub，然後選取存放庫名稱和分支名稱。	應用程式開發人員、AWS 系統管
變更偵測選項。	選擇 [在原始程式碼變更時啟動管線]，然後移至下一頁。	應用程式開發人員、AWS 系統管
選取組建提供者。	對於建置供應商，請選擇 AWS CodeBuild，然後提供建置專案	應用程式開發人員、AWS 系統管

任務	描述	所需技能
	的 AWS 區域和專案名稱詳細資訊。 針對 [建置類型] 選擇 [單一組建]。	
選擇部署提供者。	對於部署供應商，請選擇 Amazon ECS。如果需要，請選擇叢集名稱、服務名稱、映像定義檔 (如果有的話) 以及部署逾時值。選擇 Create pipeline (建立管道)。	應用程式開發人員、AWS 系統管

相關資源

- AWS ECS 文件
- AWS ECR 文件
- AWS CodeBuild 文件
- AWS CodeCommit 文件
- AWS CodePipeline 文件
- 使用 Amazon ECR 做為來源，為您的容器映像建立持續交付管道 (部落格文章)

使用 AWS CodeCommit 和 AWS 在多 CodePipeline 個 AWS 帳戶中部署 CI/CD 管道

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：PoC 或試點	技術： DevOps	工作負載：所有其他工作
AWS 服務：AWS CodeCommit；AWS CodePipeline		

Summary

此模式說明如何在不同的 Amazon Web Services (AWS) 帳戶 DevOps、開發人員、測試和生產工作流程中，為應用程式程式碼工作負載部署持續整合和持續交付 (CI/CD) 管道。

您可以使用多個 AWS 帳戶策略來提供高層級的資源或安全隔離、優化成本，以及區分生產工作流程。

您應用程式的程式碼在所有這些獨立 AWS 帳 DevOps 戶中保持相同，並且會在您的帳戶託管的中央 AWS CodeCommit 儲存庫中進行維護。您的開發人員、預備帳戶和生產帳戶在此 CodeCommit 儲存庫中有個別的 Git 分支。

例如，當程式碼提交到中央 CodeCommit 儲存庫中的開發人員 Git 分支時，您 DevOps 帳戶中的 Amazon 會 EventBridge 在您的開發人員帳戶中通知 EventBridge 儲存庫變更。在您的開發人員帳戶中，AWS CodePipeline 和來源階段會進入`InProgress`狀態。來源階段是從中央 CodeCommit 存放庫中的開發人員 Git 分支進行設定，並 CodePipeline 假設該 DevOps 帳戶的服務角色。

開發人員分支中的儲存 CodeCommit 庫內容會上傳到 Amazon Simple Storage Service (Amazon S3) 儲存貯體中的成品存放區，並使用 AWS Key Management Service (AWS KMS) 金鑰加密。在來源階段的狀態變更為 `Succeeded` in 之後 CodePipeline，程式碼會轉換至管線執行的下一個階段。

先決條件和限制

先決條件

- 每個所需環境 (DevOps、開發人員、測試和生產) 的現有 AWS 帳戶。這些帳戶可由 AWS Organizations 託管。

- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。

架構

技術堆疊

- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon EventBridge
- AWS Identity and Access Management (IAM)
- AWS KMS
- AWS Organizations
- Amazon S3

工具

- AWS CodeBuild — CodeBuild 是全受管持續整合服務，可編譯原始程式碼、執行測試，以及產生可立即部署的軟體套件。
- AWS CodeCommit — CodeCommit 是一種完全受管的原始程式碼控制服務，可託管安全的 Git 儲存庫
- AWS CodePipeline — CodePipeline 是全受管的持續交付服務，可協助您將發行管道自動化，以便快速可靠地更新應用程式和基礎設施。
- Amazon EventBridge — EventBridge 是一種無伺服器事件匯流排服務，可將您的應用程式與來自各種來源的資料連接起來。
- AWS Identity and Access Management (IAM) — IAM 可協助您安全地管理 AWS 服務和資源的存取。
- AWS KMS — AWS Key Management Service (AWS KMS) 可協助您建立和管理加密金鑰，並控制其在各種 AWS 服務和應用程式中的使用。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。

史诗

在您的 DevOps AWS 帳戶中建立資源

任務	描述	所需技能
創建一個 CodeCommit 存儲庫。	登入您 DevOps 帳戶的 AWS 管理主控台，然後開啟主 CodeCommit 控台。建立儲存庫，並為您的開發人員、預備和生產 AWS 帳戶設定所有必要的 Git 分支。如需此和其他故事的說明，請參閱「相關資源」一節。	DevOps 工程師
建立存 CodeCommit 放庫的存取認證。	在 IAM 主控台上，建立存取登入資料，以允許應用程式開發人員從儲存庫推送和提取應用程式的程式碼 CodeCommit 庫。	DevOps 工程師
為 CodePipeline 服務角色建立 IAM 角色。	在 IAM 主控台上，建立可供所有 CodePipeline 服務角色使用的 IAM 角色來存取中央 CodeCommit 存放庫。	雲端管理員
為您的其他 AWS 帳戶設定 EventBridge 規則。	在 Amazon EventBridge 主控台上，設定規則，將有關相關 CodeCommit 儲存庫變更的通知傳送至 EventBridge 個別開發人員、預備和生產 AWS 帳戶。	雲端管理員
建立 AWS KMS 金鑰。	在 AWS KMS 主控台上，建立 KMS 金鑰，讓您 CodePipeline 的個別開發人員、測試和生產 AWS 帳戶加密和解密成品。	雲端管理員

在其他 AWS 帳戶中建立資源

任務	描述	所需技能
設定 EventBridge 以接收來自 DevOps AWS 帳戶的事件。	針對其中一個個別 AWS 帳戶 (開發人員、預備或生產) 登入 AWS 管理主控台。在 Amazon 主 EventBridge 控台上，設定 EventBridge 為從您的 DevOps 帳戶接收 CodeCommit 儲存庫變更事件。	雲端管理員
建立 S3 儲存貯體。	在 Amazon S3 主控台上，建立 S3 儲存貯體來存放 CodePipeline 成品。	雲端管理員
為 CodePipeline 階段建立所有必要的 AWS 資源。	建立 CodePipeline 階段所需的所有其他 AWS 資源。這些資源會根據 CI/CD 管道中每個 AWS 帳戶的角色而有所不同。	雲端管理員
建立 IAM 角色。	在 IAM 主控台上，為 CodePipeline 服務角色建立 IAM 角色。此服務角色必須能夠在 DevOps 帳戶中擔任 IAM 角色，才能 CodeCommit 存取存放庫。	雲端管理員
在中建立管線 CodePipeline。	在主 CodePipeline 控台上，建立管線。然後建立一個來源階段，指向其個別 Git 分支 DevOps 帳戶中的 CodeCommit 儲存庫。	雲端管理員
對所有 AWS 帳戶重複上述步驟。	針對 CI/CD 策略所需的所有 AWS 帳戶重複這些步驟。	雲端管理員

相關資源

在您的 DevOps AWS 帳戶中建立資源

- 建立儲 CodeCommit 存庫
- 設定 CodeCommit 儲存庫
- 在 CodeCommit 存儲庫中創建並共享分支
- 建立 CodeCommit 儲存庫的存取認證
- 為 CodePipeline 服務角色建立 IAM 角色
- 設定規則 EventBridge
- 建立 AWS KMS 金鑰
- 設定下列項目的帳號策略和角色 CodePipeline

在其他 AWS 帳戶中建立資源

- 開啟 EventBridge 以接收來自 DevOps AWS 帳戶的事件
- 為 CodePipeline 成品建立 S3 儲存貯體
- 為 CodePipeline 階段建立所有其他必要的 AWS 資源
- 為 CodePipeline 服務角色建立 IAM 角色
- 在中建立管線 CodePipeline
- 在中 CodePipeline 建立使用其他 AWS 帳戶資源的管道

其他資源

- 建立您的最佳實務 AWS 環境
- 驗證和存取控制 CodeCommit

使用 AWS Network Firewall 和 AWS Transit Gateway 部署防火牆

由希利康帕蒂爾（AWS）創建

代碼庫：aws-network-firewall-deployment-with-transit-gateway	環境：PoC 或試點	技術： DevOps; 網絡; 安全性，身份，合規
AWS 服務：AWS Network Firewall；AWS Transit Gateway；Amazon VPC；Amazon CloudWatch		

Summary

此模式說明如何使用 AWS Network Firewall 和 AWS Transit Gateway 部署防火牆。Network Firewall 資源是使用 AWS CloudFormation 範本進行部署。Network Firewall 會隨著您的網路流量自動擴充，並可支援數十萬個連線，因此您不必擔心建立和維護自己的網路安全性基礎架構。傳輸閘道是網路傳輸中樞，您可將其用來互相連線 Virtual Private Cloud (VPC) 和內部部署網路。

在此模式中，您還將學習在網絡架構中包含檢查 VPC。最後，此模式說明如何使用 Amazon CloudWatch 為防火牆提供即時活動監控。

提示：最佳做法是避免使用 Network Firewall 子網路部署其他 AWS 服務。這是因為 Network Firewall 無法檢查來自防火牆子網路內來源或目的地的流量。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Identity and Access Management (IAM) 角色和政策許可
- CloudFormation 範本權限

限制

您可能會遇到網域篩選的問題，而且可能需要使用其他類型的設定。如需詳細資訊，請參閱 Network Firewall 文件中的 AWS Network Firewall 中的可設定狀態網域清單規則群組。

架構

技術, 堆

- Amazon CloudWatch 日誌
- Amazon VPC
- AWS Network Firewall
- AWS Transit Gateway

目標架構

下圖顯示如何使用「Network Firewall」和「Transit Gateway」來檢查流量：

該架構包括以下組件：

- 您的應用程式託管在兩個支點 VPC 中。VPC 由 Network Firewall 監控。
- 輸出 VPC 可直接存取網際網路閘道，但不受 Network Firewall 保護。
- 檢查 VPC 是部署 Network Firewall 的位置。

自動化和規模

您可以使CloudFormation用基礎結構作為程式碼來建立此模式。

工具

AWS 服務

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

- AWS Network Firewall 是適用於 AWS 雲端中 VPC 的可設定狀態、受管網路防火牆以及入侵偵測與防護服務。
- AWS Transit Gateway 是連接 VPC 和現場部署網路的中央中樞。

Code

此模式的程式碼可在具有 Transit Gateway 儲存庫的 GitHub AWS Network Firewall 部署中取得。您可以使用此存放庫中的 CloudFormation 範本來部署使用 Network Firewall 的單一檢查 VPC。

史诗

建立支點虛擬私人雲端和檢查 VPC

任務	描述	所需技能
準備和部署 CloudFormation 範本。	1. 從 GitHub 存放庫下載`cloudformation/aws_nw_fw.yml` 範本。 2. 使用您的值更新模板。 3. 部署範本。	AWS DevOps

建立交通閘道和路線

任務	描述	所需技能
建立傳輸閘道。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在瀏覽窗格中，選擇「運輸閘道」。 3. 選擇 Create transit gateway (建立傳輸閘道)。 4. 在「名稱」標籤中，輸入傳輸閘道的名稱。 5. 在「說明」中，輸入傳輸閘道的說明。	AWS DevOps

任務	描述	所需技能
	6. 對於 Amazon 端自主系統編號 (ASN)，請保留預設的 ASN 值。 7. 選取 DNS 支援選項。 8. 選擇 VPN ECMP 支援選項。 9. 選取預設路由表關聯選項。此選項會自動將傳輸閘道附件與傳輸閘道的預設路由表建立關聯。 10 選取預設路由表格傳輸選項。此選項會自動將傳輸閘道附件傳送到傳輸閘道的預設路由表。 11 選擇 Create transit gateway (建立傳輸閘道)。	
建立交通閘道附件。	為下列項目建立傳輸閘道附件： • 檢查 VPC 和 Transit Gateway 子網路中的檢查附件 • 網輻式 VPCA 和私有子網路中的 SspokeVPC A 附件 • 支點式 VPCB 和私有子網路中的 SPOkeVPCB 附件 • 輸出 VPC 和私有子網路中的輸出 SVPC 附件	AWS DevOps

任務	描述	所需技能
建立交通閘道路由表。	1. 為網輻 VPC 建立傳輸閘道路由表。此路由表必須與檢查 VPC 以外的所有 VPC 相關聯。 2. 建立防火牆的傳輸閘道路由表。此路由表必須僅與檢查 VPC 相關聯。 3. 為防火牆新增路由至傳輸閘道路由表： • 對於`0.0.0/0`，請使用電子郵件附件。 • 對於 S spokeVPC A CIDR 區塊，請使用附件。 • 對於 S spokeVPCB CIDR 區塊，請使用附件。 4. 將路由新增至網輻 VPC 的傳輸閘道路由表。對於`0.0.0/0`，請使用檢查 VPC 附件。	AWS DevOps

建立防火牆和路由

任務	描述	所需技能
在檢查 VPC 中建立防火牆。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在功能窗格的 [Network Firewall] 下，選擇 [防火牆]。 3. 選擇建立防火牆。	AWS DevOps

任務	描述	所需技能
	4. 在 [名稱] 中，輸入您要用來識別此防火牆的名稱。您無法在建立防火牆之後變更其名稱。 5. 對於 VPC，請選擇您的檢測 VPC。 6. 對於可用區域和子網路，請選取您識別的區域和防火牆子網路。 7. 在 [關聯的防火牆策略] 區段中，選擇 [關聯現有的防火牆策略]，然後選取您先前建立的防火牆策略。 8. 選擇建立防火牆。	

任務	描述	所需技能
建立防火牆策略。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在瀏覽窗格的 [Network Firewall] 下，選擇 [防火牆策略]。 3. 在 [說明防火牆策略] 頁面上，選擇 [建立防火牆策略] 4. 在 [名稱] 中，輸入您要用於防火牆策略的名稱。稍後在此病毒碼中將策略與防火牆建立關聯時，您將使用該名稱來識別策略。您無法在建立防火牆政策之後變更其名稱。 5. 選擇下一步。 6. 在 [新增規則群組] 頁面的 [無狀態規則群組] 區段中，選擇 [新增無狀態規則群組]。 7. 在 [從現有規則群組新增] 對話方塊中，選取您先前建立之無狀態規則群組的核取方塊。選擇 [新增規則群組]。注意：在頁面底部，防火牆策略的容量計數器會顯示在防火牆策略允許的最大容量旁邊新增此規則群組所消耗的容量。 8. 將無狀態的預設動作設定為「轉寄至可設定狀態規則」。 9. 在「可設定狀態規則群組」區段中，選擇「新增可設定	AWS DevOps

任務	描述	所需技能
	狀態規則群組」，然後選取您先前建立之可設定狀態規則群組的核取方塊。選擇 [新增規則群組]。 10選擇 [下一步] 以逐步完成安裝精靈的其餘部分，然後選擇 [建立防火牆原則]。	

任務	描述	所需技能
更新您的 VPC 路由表。	檢驗 VPC 路由表 1. 在ANF子網路路由表格 (`Inspection-ANFRT`) 中，新增`0.0.0/0`至 Transit Gateway ID。 2. 在 Transit Gateway 子網路路由表 (`Inspection-TGWRT`) 中，新增`0.0.0/0`至 eGressVPN。 主要航空路由表 在私人路由表格中，新增`0.0.0.0/0` 至傳 Transit Gateway ID。 輪輻式路由表 在私人路由表格中，新增`0.0.0.0/0` 至傳 Transit Gateway ID。 出口 VPC 路由表 在出口公用路由表格中，將 SspokeVPCA 和網輻式 VPCB CIDR 區塊新增至 Transit Gateway 識別碼。對私有子網路重複相同的步驟。	AWS DevOps

設定 CloudWatch 以執行即時網路檢測

任務	描述	所需技能
更新防火牆的記錄設定。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在功能窗格的 [Network Firewall] 下，選擇 [防火牆]。 3. 在「防火牆」頁面中，選擇您要編輯的防火牆名稱。 4. 選擇防火牆詳細資料標籤。在「記錄日誌」段落中，選擇編輯。 5. 視需要調整 [防護記錄類型] 選項。您可以設定警示和流程記錄的記錄。 • 警示 — 傳送符合任何可設定狀態規則 (其中動作設定為「警示」或「中斷」) 的流量記錄。如需有狀態規則和規則群組的詳細資訊，請參閱 AWS Network Firewall 中的規則群組。 • 流量 — 傳送無狀態引擎轉寄至可設定狀態規則引擎的所有網路流量記錄。 6. 針對每個選取的記錄檔類型，選擇目的地類型，然後提供記錄目的地的資訊。如需詳細資訊，請參閱 Network Firewall 文件中的 AWS Network Firewall 記錄目的地。	AWS DevOps

任務	描述	所需技能
	7. 選擇儲存。	

驗證設定

任務	描述	所需技能
啟動 EC2 執行個體以測試設定。	在支點 VPC 中啟動兩個 Amazon 彈性運算雲端 (Amazon EC2) 執行個體：一個用於 Jumpbox，另一個用於測試連線。	AWS DevOps
檢查指標。	測量結果會先依服務命名空間分組，然後依每個命名空間內的各種維度組合分組。Network Firewall 的 CloudWatch 命名空間為`AWS/NetworkFirewall`。 1. 登入 AWS 管理主控台，並開啟 CloudWatch 主控台。 2. 在導覽窗格中，選擇 指標。 3. 在 [所有量度] 索引標籤上，選擇 [區域]，然後選擇 [AWS/ NetworkFirewall]。	AWS DevOps

相關資源

- 簡單的單一區域架構搭配網際網路閘道
- 具備網際網路閘道的多區域架構
- 具有網際網路閘道和 NAT 閘道的架構

使用 AWS CodePipeline CI/CD 管道部署 AWS 膠合任務

創建者：布魯諾·克萊因（AWS）和路易斯·亨里克山田莊（AWS）

環境：生產	技術: DevOps; 大數據	AWS 服務：AWS AWS Glue CodeCommit；AWS CodePipeline；AWS；AWS

Summary

此模式示範如何將 Amazon Web Services (AWS) CodeCommit 和 AWS CodePipeline 與 AWS Glue 整合，並在開發人員將變更推送至遠端 AWS CodeCommit 儲存庫時立即使用 AWS Lambda 啟動任務。

當開發人員提交對擷取、轉換和載入 (ETL) 存放庫的變更，並將變更推送到 AWS 時 CodeCommit，會叫用新管道。管道會啟動 Lambda 函數，透過這些變更啟動 AWS Glue 任務。AWS AWS Glue 任務會執行 ETL 任務。

在企業、開發人員和資料工程師希望在提交變更並推送至目標儲存庫後立即啟動工作時，此解決方案非常有用。它有助於實現更高級別的自動化和重現性，從而避免任務啟動和生命週期期間發生錯誤。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Git 安裝在本地機器上
- Amazon Cloud Development Kit（Amazon CDK）安裝在本地計算機上
- Python 安裝在本地計算機上
- 附件部分中的代碼

限制

- AWS Glue 任務成功啟動後，管道就會完成。它不會等待工作的結束。

- 附件中提供的代碼僅用於演示目的。

架構

目標技術堆疊

- AWS Glue
- AWS Lambda
- AWS CodePipeline
- AWS CodeCommit

目標架構

該過程包括以下步驟：

1. 開發人員或資料工程師對 ETL 程式碼進行修改、提交，並將變更推送至 AWS。 CodeCommit
2. 推送會啟動管線。
3. 管道會啟動 Lambda 函數，該函數會在儲存庫`codecommit:GetFile`上呼叫，並將檔案上傳到亞馬遜簡單儲存服務 (Amazon S3)。
4. Lambda 函數會使用 ETL 程式碼啟動新的 AWS AWS Glue 任務。
5. Lambda 函數完成管道。

自動化和規模

範例附件示範如何將 AWS Glue 與 AWS 整合 CodePipeline。它提供了一個基準示例，您可以自定義或擴展以供自己使用。有關詳細信息，請參見史詩部分。

工具

- AWS CodePipeline — AWS CodePipeline 是全受管的持續交付服務，可協助您自動化發行管道，以快速可靠地更新應用程式和基礎設施。
- AWS CodeCommit — AWS CodeCommit 是一種全受管的原始檔控制服務，可託管安全的 Git 型存放庫。
- AWS Lambda — AWS Lambda 是一種無伺服器運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。

- AWS Glue — AWS Glue 是一種無伺服器資料整合服務，可讓您輕鬆探索、準備和合併資料，以進行分析、機器學習和應用程式開發。
- Git 客戶端 — Git 提供 GUI 工具，或者您可以使用命令行或桌面工具從 GitHub.
- AWS CDK — AWS CDK 是開放原始碼軟體開發架構，可協助您使用熟悉的程式設計語言定義雲端應用程式資源。

史诗

部署範例程式碼

任務	描述	所需技能
設定 AWS CLI。	將 AWS Command Line Interface (AWS CLI) (AWS CLI) 設定為目標並使用您目前的 AWS 帳戶進行驗證。如需相關指示，請參閱 AWS CLI 文件。	開發者、 DevOps 工程師
解壓縮範例專案檔案。	從附件中解壓縮檔案，以建立包含範例專案檔案的資料夾。	開發者、 DevOps 工程師
部署範例程式碼。	解壓縮檔案後，請從擷取位置執行以下命令以建立基準範例： `cdk bootstrap` `cdk deploy` `git init` `git remote add origin <code-commit-repository-url>` `git stage .` `git commit -m "adds sample code"` `git push --set-upstream origin main`	開發者、 DevOps 工程師

任務	描述	所需技能
	執行最後一個命令後，您可以監控管道和 AWS Glue 任務的狀態。	
自訂程式碼。	根據您的業務需求自訂 etl.py 檔案的程式碼。您可以修改 ETL 代碼，修改管道階段或擴展解決方案。	數據工程師

相關資源

- 開始使用 AWS CDK
- 在 AWS AWS Glue 中新增任務
- 源動作集成 CodePipeline
- 在管道中叫用 AWS Lambda 函數 CodePipeline
- AWS AWS Glue 編程
- AWS CodeCommit GetFile API

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 EC2 執行個體設定檔從 AWS Cloud9 部署 Amazon EKS 叢集

由薩加爾·帕尼格拉希（AWS）創建

環境：生產	技術： DevOps；容器與微服務	工作負載：所有其他工作
AWS 服務：Amazon EKS；AWS Cloud9；AWS Identity and Access Management；AWS CloudFormation		

Summary

此模式說明如何使用 AWS Cloud9 和 AWS CloudFormation 建立 Amazon Elastic Kubernetes Service (Amazon EKS) 叢集，無需為 Amazon Web Services (AWS) 帳戶中的使用者啟用程式設計存取即可操作該叢集。

AWS Cloud9 是雲端整合式開發環境 (IDE)，可協助您使用瀏覽器撰寫、執行和偵錯程式碼。AWS Cloud9 可做為控制中心，透過使用亞馬遜彈性運算雲端 (Amazon EC2) 執行個體設定檔和 AWS CloudFormation 範本佈建 Amazon EKS 叢集。

如果您不想建立 AWS Identity and Access Management (IAM) 使用者，而且想要改用 IAM 角色，則可以使用此模式。以角色為基礎的存取控制 (RBAC) 會根據個別使用者的角色來規範資源的存取。此模式示範如何更新 Amazon EKS 叢集中的 RBAC，以允許存取特定 IAM 角色。

該模式的設定也可協助您的 DevOps 團隊使用 AWS Cloud9 功能來維護和開發基礎設施即程式碼 (IaC) 資源，以建立 Amazon EKS 基礎設施。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 為帳戶建立 IAM 角色和政策的許可。使用者的 IAM 角色必須包含`AWSCloud9Administrator`政策。還必須建立`AWSServiceRoleForAmazonEKS`和`eksNodeRoles`角色，因為必須建立 Amazon EKS 叢集。

- 庫伯尼特人概念的知識。

限制

- 此模式說明如何建立基本的 Amazon EKS 叢集。對於生產叢集，您必須更新 AWS CloudFormation 範本。
- 此病毒碼不會部署其他 Kubernetes 元件 (例如，流量、輸入控制器或儲存控制器)。

架構

技術, 堆

- AWS Cloud9
- AWS CloudFormation
- Amazon EKS
- IAM

自動化和規模

您可以擴展此模式並將其整合到持續整合和持續部署 (CI/CD) 管道中，以自動化 Amazon EKS 的完整佈建。

工具

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定，以減少管理這些資源的時間，將更多時間專注於應用程式。
- AWS Cloud9 — AWS Cloud9 提供豐富的程式碼編輯體驗，並支援多種程式設計語言和執行階段除錯器，以及內建終端機。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- Kubectl — `kubectl` 命令列公用程式可用來與 Amazon EKS 叢集互動。

史诗

為 EC2 執行個體設定檔建立 IAM 角色

任務	描述	所需技能
建立 IAM 政策。	登入 AWS 管理主控台，開啟 IAM 主控台，選擇 [政策]，然後選擇 [建立政策]。選擇 JSON 選項卡並粘貼 policy-role-eks-instance-profile-for-cloud 9.json 文件中的內容（附件）。 解決原則驗證期間產生的任何安全性警告、錯誤或一般警告，然後選擇 [檢閱原則]。輸入政策的名稱。我們建議您使`eks-instance-profile-for-cloud9` 用策略名稱。 檢閱政策 Summary (摘要) 來查看您的政策所授予的許可。然後選擇 Create policy (建立政策)。	雲端管理員
使用政策建立 IAM 角色。	在 IAM 主控台上，選擇 [角色]，然後選擇 [建立角色]。選擇 AWS 服務，然後從清單中選擇 EC2。 選擇下一步：許可並搜尋您先前建立的 IAM 政策。根據您的需求選擇適當的標籤。 在「複查」區段中，輸入角色的名稱。我們建議您	雲端管理員

任務	描述	所需技能
	使`role-eks-instance-profile-for-cloud9` 用角色名稱。然後選擇 Create role (建立角色)。	

為 Amazon EKS RBAC 建立 IAM 政策和角色

任務	描述	所需技能
建立 IAM 政策。	在 IAM 主控台上，選擇 [政策]，然後選擇 [建立政策]。選擇 JSON 選項卡，然後粘貼 policy-for-eks-rbac .json 文件中的內容（附件）。 解決原則驗證期間產生的任何安全性警告、錯誤或一般警告，然後選擇 [檢閱原則]。輸入政策的名稱。我們建議您使`policy-for-eks-rbac` 用策略名稱。檢閱政策 Summary (摘要) 來查看您的政策所授予的許可。然後選擇 Create policy (建立政策)。	雲端管理員
使用政策建立 IAM 角色。	在 IAM 主控台上，選擇 [角色]，然後選擇 [建立角色]。選擇 AWS 服務，然後從清單中選擇 EC2。選擇下一步：許可並搜尋您先前建立的 IAM 政策。根據您的需求選擇適當的標籤。 在「複查」區段中，輸入角色的名稱。我們建議您使`role-`	雲端管理員

任務	描述	所需技能
	`eks-admin-for-rbac` 用角色名稱。然後選擇 Create role (建立角色)。	

建立 AWS Cloud9 環境

任務	描述	所需技能
建立 AWS Cloud9 環境。	開啟 AWS Cloud9 主控台，然後選擇建立環境。在 [名稱環境] 頁面上，輸入環境的名稱。我們建議您使`eks-management-env` 用環境名稱。根據您的需求設定其餘設定，然後選擇 [下一步]。 在 Review (檢閱) 頁面上，選擇 Create environment (建立環境)。等待 AWS Cloud9 建立您的環境。這可能需要幾分鐘的時間。 如需有關可用組態選項的詳細資訊，請參閱 AWS Cloud9 文件中的建立 EC2 環境。	雲端管理員
移除 AWS Cloud9 的臨時身分與存取權管理登入資料。	佈建 AWS Cloud9 環境後，在齒輪圖示中選擇「設定」。在喜好設定下，選擇 AWS 設定，然後選擇登入資料。 關閉 AWS 受管的臨時登入資料，然後關閉索引標籤。	雲端管理員
將 EC2 執行個體設定檔連接至基礎 EC2 執行個體。	開啟 Amazon EC2 主控台，然後選擇符合 AWS Cloud9 環境	雲端管理員

任務	描述	所需技能
	的 EC2 執行個體。如果您使用我們建議的名稱，則會呼叫 EC2 執行個體`aws-cloud9-eks-management-env` 。 選擇 EC2 執行個體，選擇 [動作]，然後選擇 [執行個體設定]。選擇 [附加/取代 IAM 角色]。搜尋`role-eks-instance-profile-for-cloud9` 或先前建立的 IAM 角色名稱，然後選擇 [套用]。	

建立 Amazon EKS 叢集

任務	描述	所需技能
建立 Amazon EKS 叢集。	下載並開啟適用於 AWS 的 eks-cfn.yaml (隨附) 範本。CloudFormation根據您的需求編輯範本。 開啟 AWS Cloud9 環境，然後選擇 [新增檔案]。將您先前建立的 AWS CloudFormation 範本貼到欄位中。我們建議您使用 eks-cfn.yaml 作為範本名稱。 在 AWS Cloud9 終端機中，執行下列命令以建立 Amazon EKS 叢集： `aws cloudformation create-stack --`	雲端管理員

任務	描述	所需技能
	`stack-name eks-cluster --template-body file://eks-cfn.yaml --region <your_AWS_Region>` 如果 AWS CloudFormation 呼叫成功，您會在輸出中收到 AWS CloudFormation 堆疊的 Amazon 資源名稱 (ARN)。堆疊建立可能需要 10 到 20 分鐘。	
驗證 Amazon EKS 叢集的狀態。	在 AWS 主 CloudFormation 控台上，開啟「堆疊」頁面，然後選擇堆疊名稱。 堆疊狀態碼顯示時會建立堆疊`CREATE_COMPLETE` 。如需詳細資訊，請參閱 AWS CloudFormation 文件中的檢視 AWS CloudFormation 堆疊資料和資源。	雲端管理員

存取 Amazon EKS 叢集中的 Kubernetes 資源

任務	描述	所需技能
在 AWS Cloud9 環境中安裝庫貝克特爾。	按照 Amazon EKS 文件`kubectl`中的安裝 kubectl 中的指示，在您的 AWS Cloud9 環境中進行安裝。	雲端管理員
在 AWS Cloud9 中更新新的 Amazon EKS 組態。	在 AWS Cloud9 終端機中執行下列命令，以便`kubeconfi`	雲端管理員

任務	描述	所需技能
	g 從 Amazon EKS 叢集更新到 AWS Cloud9 環境： `aws eks update-kubeconfig --name EKS-DEV2 --region <your_AWS_Region>` 重要事項：這`EKS-DEV2`是您用來建立叢集的 AWS CloudFormation 範本中 Amazon EKS 叢集的名稱。 執行命`kubectl get all -A`令以檢視所有 Kubernetes 資源。	

任務	描述	所需技能
將管理員身分與存取權管理員角色新增至 Kubernetes RBAC。	在 AWS Cloud9 終端機中執行下列命令，以編輯模式開啟適用於 Amazon EKS 的 RBAC 組態對應： `kubectl edit cm/aws-auth -n kube-system` 在`mapRoles`區段下方附加下列幾行： `- groups:` `- system:masters` `rolearn: <ARN_of_IAM_role _from_second_epic>` `username: eksadmin` Lint YAML 格式的檔案，以避免語法錯誤。使用`vi`指令儲存檔案，然後結束檔案。 附註：透過新增本節，您可以通知 Kubernetes RBAC 將在 Amazon EKS 叢集上接收完整的管理員存取權。`<ARN_of_IAM_role _from_second_epic>` 這表示識別的身分與存取權管理角色可以在 Kubernetes 叢集上執行管理動作。AWS 會在佈建 Amazon EKS 叢集`mapRoles`時在下方新增現有區段。	雲端管理員

相關資源

參考

- 模組化且可擴展的 Amazon EKS 架構 (快速入門)
- 管理 Amazon EKS 叢集的使用者或 IAM 角色
- 用於建立新的 Amazon EKS 控制平面的 AWS CloudFormation 範本

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS CodePipeline、AWS 和 AWS 在多個 AWS 區域部署程式碼 CodeCommit CodeBuild

創建者拉瑪·阿南德·克里希納·瓦拉納西（AWS）

創建者：AWS	環境：PoC 或試點	技術：管理與治理； DevOps
AWS 服務：AWS CodeCommit；AWS CodePipeline；AWS CodeBuild		

Summary

此模式示範如何使用 AWS 跨多個 Amazon Web Services (AWS) 區域建立基礎設施或架構 CloudFormation。其中包括跨多個 AWS 區域的持續整合 (CI)/持續部署 (CD)，以加快部署速度。 此模式中的步驟已經過測試，建立 AWS CodePipeline 任務以部署到三個 AWS 區域為例。您可以根據使用案例變更「區域」數目。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 適用於 AWS 和 AWS 的兩個 AWS Identity CodeBuild and Access Management (IAM) 角色，其中包 CloudFormation 含適當的政策，用於 CodeBuild 執行測試、捆綁、封裝成品以及 parallel 行部署到多個 AWS 區域的 CI 任務。 注意：交叉檢查由建立的政策， CodePipeline 以確認 AWS 在 CI CodeBuild 和 CD 階段是否 CloudFormation 具有適當的許可。
  - 與亞馬遜 S3 FullAccess 和CloudWatchFullAccess政策的 CodeBuild 角色。這些政策可讓您 CodeCommit 透過 Amazon 存 CodeBuild 取 AWS 的觀看事件，以 CloudWatch 及使用亞馬遜簡單儲存服務 (Amazon S3) 做為成品存放區。
  - 具有下列政策的 AWS CloudFormation 角色，讓 AWS CloudFormation 在最後的建置階段能夠建立或更新 AWS Lambda 函數、推送或觀看 Amazon CloudWatch 日誌，以及建立和更新變更集。
    - AWSLambdaFullAccess
    - AWSCodeDeployFullAccess

- CloudWatchFullAccess
- AWSCloudFormationFullAccess
- AWSCodePipelineFullAccess

架構

此模式的多區域架構和工作流程包括以下步驟。

1. 您將程式碼傳送至儲 CodeCommit 存庫。
2. 在收到任何代碼更新或提交後， CodeCommit 調用 CloudWatch 事件，這反過來又啟動 CodePipeline 工作。
3. CodePipeline 參與由 CodeBuild處理的 CI。會執行下列工作。
   - 測試 AWS CloudFormation 範本 (選用)
   - 包含在部署中的每個區域的 AWS CloudFormation 範本封裝。例如，此模式會 parallel 部署到三個 AWS 區域，因此將 AWS CloudFormation 範本 CodeBuild 封裝到三個 S3 儲存貯體中，每個指定區域各一個。S3 儲存貯體僅供 CodeBuild 成品儲存庫使用。
4. CodeBuild 將成品封裝為下一個部署階段的輸入，該階段會在三個 AWS 區域 parallel 執行。如果您指定不同數量的區域， CodePipeline 將會部署到這些區域。

工具

工具

- AWS CodePipeline — CodePipeline 是一種持續交付服務，可用於建立模型、視覺化和自動化持續發行軟體變更所需的步驟。
- AWS CodeBuild — CodeBuild 是全受管的建置服務，可編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit — CodeCommit 是由 Amazon Web Services 託管的版本控制服務，您可以使用它在雲端私有存放和管理資產 (例如原始碼和二進位檔案)。
- AWS CloudFormation — AWS CloudFormation 是一項服務，可協助您建立 Amazon Web Services 資源模型和設定，以減少管理這些資源的時間，將更多時間專注於在 AWS 中執行的應用程式。
- AWS Identity and Access Management — AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。

- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。此服務旨在降低開發人員進行網路規模運算的難度。

Code

以下示例代碼適用於文BuildSpec.yaml件（構建階段）。

```
---
artifacts:
discard-paths: true
files:
- packaged-first-region.yaml
- packaged-second-region.yaml
- packaged-third-region.yaml
phases:
build:
commands:
- echo "********BUILD PHASE - CF PACKAGING**********"
- "aws cloudformation package --template-file sam-template.yaml --s3-bucket
 $S3_FIRST_REGION --output-template-file packaged-first-region.yaml --region
 $FIRST_REGION"
- "aws cloudformation package --template-file sam-template.yaml --s3-bucket
 $S3_SECOND_REGION --output-template-file packaged-second-region.yaml --region
 $SECOND_REGION"
- "aws cloudformation package --template-file sam-template-anand.yaml --s3-bucket
 $S3_THIRD_REGION --output-template-file packaged-third-region.yaml --region
 $THIRD_REGION"
install:
commands:
- echo "********BUILD PHASE - PYTHON SETUP**********"
runtime-versions:
python: 3.8
post_build:
commands:
- echo "********BUILD PHASE - PACKAGING COMPLETION**********"
pre_build:
commands:
- echo "********BUILD PHASE - DEPENDENCY SETUP**********"
- "npm install --silent --no-progress"
- echo "********BUILD PHASE - DEPENDENCY SETUP DONE**********"
version: 0.2
```

史诗

準備代碼和 CodeCommit 存儲庫

任務	描述	所需技能
選取用於部署的主要 AWS 區域。	登入您的 AWS 帳戶並選擇部署的主要區域。存 CodeCommit 放庫將位於主要區域中。	DevOps
建立存 CodeCommit 放庫。	創建存 CodeCommit 儲庫，並將所需的代碼推入其中。程式碼通常包括 AWS CloudFormation 或 AWS SAM 範本、Lambda 程式碼 (如果有的話)，以及做為輸入到 AWS 的 CodeBuild `buildspec.yaml` 檔案 CodePipeline。	DevOps
將代碼推入存 CodeCommit 儲庫。	在「附件」區段中，下載此範例的程式碼，然後將所需的程式碼推入其中。一般而言，這些程式碼可以包含 AWS CloudFormation 或 AWS SAM 範本、Lambda 程式碼和 CodeBuild `buildspec.yaml` 檔案做為輸入到管道中。	DevOps

來源階段：建立管線

任務	描述	所需技能
建立工 CodePipeline 作。	在 CodePipeline 主控台上，選擇 [建立管線]。	DevOps

任務	描述	所需技能
命名工 CodePipeline 作並選擇服務角色設定。	輸入工作的名稱，並保留預設服務角色設定，以便 CodePipeline 建立附加必要原則的角色。	DevOps
指定人工因素存放區的位置。	在 [進階設定] 下，保留預設選項，以 CodePipeline 建立用於程式碼成品儲存的 S3 儲存貯體。如果您改用現有的 S3 儲存貯體，則儲存貯體必須位於您在第一個史詩中指定的主要區域。	DevOps
指定加密金鑰。	保留預設選項：預設 AWS 受管金鑰，或選擇使用您自己的 AWS Key Management Service (AWS KMS) 客戶受管金鑰。	DevOps
指定來源提供者。	在「來源供應商」下，選擇 AWS CodeCommit。	DevOps
指定存放庫。	選擇您在第一個史詩中創建的 CodeCommit 存儲庫。如果您將程式碼放在分支中，請選擇分支。	DevOps
指定偵測程式碼變更的方式。	保留預設的 Amazon CloudWatch 事件，CodePipeline 作為啟動任務的 CodeCommit 變更觸發器。	DevOps

建置階段：設定管道

任務	描述	所需技能
指定組建提供者。	對於建置供應商，請選擇 AWS CodeBuild。	DevOps
指定 AWS 區域。	選擇您在第一個史詩中指定的主要區域。	DevOps

構建階段：創建和配置項目

任務	描述	所需技能
建立專案	選擇 [建立專案]，然後輸入專案的名稱。	DevOps
指定環境影像。	對於此模式示範，請使用預設的 CodeBuild 受管理映像檔。您還可以選擇使用自定義 Docker 映像（如果有的話）。	DevOps
指定作業系統。	選擇任一 Amazon Linux 2 或 Ubuntu。	DevOps
指定服務角色。	在開始建立 CodePipeline 工作 CodeBuild 之前，請選擇您為其建立的角色。(請參閱先決條件一節)。	DevOps
設定其他選項。	對於「逾時」和「佇列」逾時，請保留預設值。若為憑證，請保留預設設定，除非您擁有要使用的自訂憑證。	DevOps
建立環境變數。	針對您要部署的每個 AWS 區域，透過提供 S3 儲存貯體名	DevOps

任務	描述	所需技能
	稱和區域名稱 (例如 us-east-1) 來建立環境變數。	
提供構建規格文件名，如果它不是構建規格 .yml。	如果檔案名稱為預設值，請將此欄位保持空白`buildspec.yaml` 。如果您重新命名了 Buildspec 檔案，請在此輸入名稱。確保它與 CodeCommit 存儲庫中的文件的名稱匹配。	DevOps
指定記錄。	若要查看 Amazon CloudWatch 事件的日誌，請保留預設設定。或者，您可以定義任何特定的組或記錄器名稱。	DevOps

略過部署階段

任務	描述	所需技能
略過部署階段並完成管道的建立。	設定管線時， CodePipeline 可讓您在「部署」階段中僅建立一個階段。若要部署到多個 AWS 區域，請跳過此階段。建立管道之後，您可以新增多個部署階段。	DevOps

部署階段：設定要部署至第一個區域的管道

任務	描述	所需技能
將階段新增至部署階段。	編輯管線，然後在「部署」階段中選擇「新增」階段。第一階段適用於主要區域。	DevOps

任務	描述	所需技能
提供階段的動作名稱。	輸入反映第一個 (主要) 階段和「區域」的唯一名稱。<region>例如，輸入主 _ _ 部署。	DevOps
指定動作提供者。	針對動作供應商，選擇 AWS CloudFormation。	DevOps
設定第一階段的「區域」。	選擇第一個（主要）區域，即設置 CodePipeline 和 CodeBuild 的相同「區域」。這是您要部署堆疊的主要區域。	DevOps
指定輸入人工因素。	選擇BuildArtifact。這是構建階段的輸出。	DevOps
指定要採取的動作。	在「動作」模式中，選擇「建立或更新堆疊」。	DevOps
輸入 CloudFormation 堆疊的名稱。		DevOps
指定第一個「區域」的範本。	針對第一個 (主要) 區域，選取由封裝 CodeBuild 並傾印至 S3 儲存貯體的區域特定套件名稱。	DevOps
指定權能。	如果堆疊範本包含 IAM 資源，或者您直接從包含巨集的範本建立堆疊，則需要功能。對於此模式，請使用能力 _IAM，功能名稱 _IAM，功能 _ 自動擴展。	DevOps

部署階段：設定管線以部署至第二個區域

任務	描述	所需技能
將第二個階段新增至部署階段。	若要為第二個區域新增階段，請編輯管線，然後在「部署」階段中選擇「新增」階段。重要事項：建立第二個「區域」的程序與第一個「區域」的程序相同，但下列值除外。	DevOps
提供第二個階段的動作名稱。	輸入反映第二個階段和第二個「區域」的唯一名稱。	DevOps
設定第二個階段的「區域」。	選擇您要部署堆疊的第二個區域。	DevOps
指定第二個「區域」的樣板。	選取由第二個區域封裝 CodeBuild 並傾印至 S3 儲存貯體的區域特定套件名稱。	DevOps

部署階段：設定要部署至第三個區域的管道

任務	描述	所需技能
將第三個階段新增至部署階段。	若要新增第三個區域的階段，請編輯管線，然後在「部署」階段中選擇「新增階段」。重要事項：建立第二個「區域」的程序與前兩個「區域」的程序相同，但下列值除外。	DevOps
提供第三個階段的動作名稱。	輸入反映第三個階段和第三個區域的唯一名稱。	DevOps

任務	描述	所需技能
設定第三階段的「區域」。	選擇您要部署堆疊的第三個區域。	DevOps
指定第三個「區域」的樣板。	選取由第三個區域封裝 CodeBuild 並傾印至 S3 儲存貯體的區域特定套件名稱。	DevOps

清理部署

任務	描述	所需技能
刪除 AWS 資源。	若要清理部署，請刪除每個區域中的 CloudFormation 堆疊。然後從主要「 CodeCommit 區域」刪除 CodeBuild、與 CodePipeline 資源。	DevOps

相關資源

- 什麼是 AWS CodePipeline？
- AWS 無伺服器應用程式模型
- AWS CloudFormation
- 適用於 AWS 的 AWS CloudFormation 架構結構參考 CodePipeline

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

將 AWS Organizations 中各個組織的 AWS Backup 報告匯出為 CSV 檔案

由芳香拉傑傑亞拉揚（AWS）和普魯舒坦 G K（AWS）創建

代碼存儲庫：aws-backup-report-generator	環境：PoC 或試點	技術: DevOps; 基礎設施
工作負載：所有其他工作	AWS 服務：AWS Backup；AWS Identity and Access Management；AWS Lambda；Amazon S3；Amazon EventBridge	

Summary

此模式顯示如何將 AWS Organizations 中各個組織的 AWS Backup 任務報告匯出為 CSV 檔案。該解決方案使用 AWS Lambda 和 Amazon 根據其狀態對 AWS Backup 任務報告進行分類，這有助於設定以狀態 EventBridge 為基礎的自動化。

AWS Backup 可協助組織集中管理和自動化 AWS 服務、雲端和內部部署的資料保護。不過，對於 AWS Organizations 內設定的 AWS Backup 任務，合併報告只能在每個組織管理帳戶的 AWS 管理主控台中使用。將此報告引入管理帳戶之外，可以減少稽核所需的工作量，並增加自動化、通知和警示的範圍。

先決條件和限制

前提

- 有效的 AWS 帳戶
- AWS Organizations 中的作用中組織，至少包含一個管理帳戶和一個成員帳戶
- AWS Backup 在 AWS Organizations 中在組織層級設定 (如需詳細資訊，請參閱使用 AWS 部落格上的 AWS Backup 在 AWS 服務之間進行大規模集中式備份)
- Git，在本地計算機上安裝和配置

限制

此模式提供的解決方案僅識別針對 AWS Backup 任務設定的 AWS 資源。報告無法識別未透過 AWS Backup 設定備份的 AWS Backup 資源。

架構

目標技術堆疊

- AWS Backup
- AWS CloudFormation
- Amazon EventBridge
- AWS Lambda
- AWS Security Token Service (AWS STS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Identity and Access Management (IAM)

目標架構

下圖顯示了將 AWS Organizations 中各個組織的 AWS Backup 任務報告匯出為 CSV 檔案的範例工作流程。

該圖顯示以下工作流程：

1. 排程的 EventBridge 事件規則會叫用成員 (報告) AWS 帳戶中的 Lambda 函數。
2. 然後，Lambda 函數會使用 AWS STS 來擔任具有連線到管理帳戶所需許可的 IAM 角色。
3. 然後，Lambda 函數會執行下列動作：

- 向 AWS Backup 服務請求合併的 AWS Backup 任務報告
- 根據 AWS Backup 任務狀態對結果進行分類
- 將回應轉換為 CSV 檔案
- 將結果上傳到報告帳戶中的 Amazon S3 儲存貯體，該資料夾根據其建立日期標記

工具

工具

- AWS Backup 是一種全受管服務，可協助您集中和自動化 AWS 服務、雲端和內部部署的資料保護。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

代碼

此模式的代碼可在 GitHub aws-backup-report-generator存儲庫中找到。

最佳實務

- Amazon S3 的安全最佳實務 (Amazon S3 使用者指南)
- 使用 AWS Lambda 函數的最佳實務 (AWS Lambda 開發人員指南)
- 管理帳戶的最佳實務 (AWS Organizations 使用者指南)

史诗

部署解決方案元件

任務	描述	所需技能
克隆存 GitHub 儲庫。	在終端機視窗中執行下列命令來複製 GitHub aws-backup-report-generator儲存庫：	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	`git clone https://github.com/aws-samples/aws-backup-report-generator.git` 如需詳細資訊，請參閱在 GitHub 文件中複製儲存庫。	

任務	描述	所需技能
在成員 (報告) AWS 帳戶中部署解決方案元件。	1. 在成員 (報告) 帳戶中，登入 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 選擇 Create stack (建立堆疊)，然後選擇 With new resources (standard) (使用新資源 (標準))。 3. 在 [建立堆疊] 頁面的 [指定範本] 區段中，選擇 [上傳範本檔案]。 4. 選取 Choose file (選擇檔案)。然後，瀏覽至本機工作站上複製 GitHub 儲存庫的根資料夾，並選擇範本報告 .yaml。 5. 選擇 [開啟]，然後選擇 [下一步]。 6. 在 [指定堆疊詳細資料] 頁面上，對於 [堆疊名稱]，輸入 CloudFormation 堆疊的名稱。 7. 對於 ManagementAccountID，請在 AWS 組織中輸入組織管理帳戶的 AWS Organizations 帳戶 ID。 8. 選擇下一步。 9. 在 [設定堆疊選項] 頁面上，選擇 [下一步]。 10. 在「複查」頁面上，選取核取方塊，以確認您已檢閱組態。	DevOps 工程師, AWS DevOps

任務	描述	所需技能
	11選擇建立堆疊。當解決方案元件部署在成員 (報告) 帳戶中時，堆疊會顯示 CREATE_CO MPLETE 狀態。	

測試解決方案

任務	描述	所需技能
在測試之前，請確定 EventBridge 規則已執行。	請等待至少 24 小時，或增加範本之範本 report .yml 檔案中的報告頻率，以確保 EventBridge 規則執行。 CloudFormation 若要增加報告頻率 1. 在複製的存放庫中開啟範本報告 .yml 檔案。 2. 在具有邏輯識別碼 'LambdaSchedule' 的事件規則中，尋找 'Schedule Expression'。 3. 編輯 'ScheduleExpression' 鍵，使其包含有效的 cron 運算式。例如，下列 cron 運算式會將事件規則排程為每五分鐘執行一次：“`cron (* /5 * * * *)`”	AWS DevOps、 DevOps 工程師
檢查 Amazon S3 儲存貯體以取得產生的報告。	1. 在成員 (報告) 帳戶中，登入 AWS 管理主控台，然後開啟主CloudFormation 控台。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	2. 在「堆疊」窗格中，選取您建立的堆疊名稱。然後，選擇資源標籤。 3. 在「資源」窗格的「邏輯 ID」欄中，找到「BackupReportS3 Bucket」。然後，在新索引標籤中開啟關聯的 Amazon S3 儲存貯體，方法是選取該邏輯 ID 旁邊的實體 ID 欄中的連結。 4. 請確定值區包含以下格式產生的報告：BackupReports///BackupReport------.<yyyy><mm><dd><BACKUP JOB STATUS><dd><Mon><yyyy>csv	

清除您的資源

任務	描述	所需技能
從成員 (報告) 帳戶刪除解決方案元件。	1. 在成員 (報告) 帳戶中，開啟解決方案的 Amazon S3 儲存貯體。如需指示，請參閱檢查 S3 儲存貯體中的步驟 2-4，瞭解此模式的「測試解決方案」一節的產生報告故事。 2. 刪除值區的內容並清空值區。如需指示，請參閱 Amazon S3 使用者指南中的清空儲存貯體。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	3. 在成員 (報告) 帳戶中，登入 AWS 管理主控台，然後開啟主CloudFormation 控台。 4. 在「堆疊」窗格中，選取您建立的堆疊名稱旁邊的核取方塊。再選擇 Delete (刪除)。	
從管理帳戶刪除解決方案元件。	1. 在管理帳戶中，登入 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 在「堆疊」窗格中，選取您建立的堆疊名稱旁邊的核取方塊。再選擇 Delete (刪除)。	AWS DevOps、 DevOps 工程師

相關資源

- 教學：將 AWS Lambda 與排程事件搭配使用 (AWS Lambda 文件)
- 建立排程事件以執行 AWS Lambda 函數 (適用於文件的 AWS 開發套 JavaScript 件)
- IAM 教學課程：使用 IAM 角色 (IAM 文件) 在 AWS 帳戶之間委派存取權
- AWS Organizations 術語和概念 (AWS Organizations 文件)
- 使用 AWS Backup 主控台建立報告計劃 (AWS Backup 文件)
- 建立稽核報告 (AWS Backup 文件)
- 建立隨需報告 (AWS Backup 文件)
- 什麼是 AWS Backup？ (AWS Backup 文件)
- 使用 AWS Backup 自動化跨 AWS 服務大規模集中 AWS Backup (AWS 部落格文章)

將 Amazon EC2 執行個體清單的標籤匯出為 CSV 檔案

由思達菊 (AWS) 和派俊賢 (AWS) 創建

程式碼儲存庫：搜尋和匯出 EC2 標籤	環境：生產	技術： DevOps
AWS 服務：Amazon EC2		

Summary

此模式顯示如何以程式設計方式將 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體清單的標籤匯出至 CSV 檔案。

透過使用提供的 Python 指令碼範例，您可以縮短依特定標籤檢閱和分類 Amazon EC2 執行個體所需的時間。例如，您可以使用指令碼快速識別和分類安全性團隊已標記進行軟體更新的執行個體清單。

先決條件和限制

先決條件

- 安裝和配置 Python 3
- 安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)

限制

此模式中提供的 Python 指令碼範例只能根據下列屬性搜尋 Amazon EC2 執行個體：

- 執行個體 ID
- 私有 IPv4 地址
- 公有 IPv4 地址

工具

- Python 是一種通用的計算機編程語言。
- 虛擬環境可以幫助您創建孤立的 Python 環境。

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

代碼存儲庫

此模式的示例 Python 腳本可在 GitHub 搜索 ec2-instances-export-tags 存儲庫中找到。

史诗

安裝和設定必要條件

任務	描述	所需技能
克隆存 GitHub 儲庫。	注意：如果您在執行 AWS CLI 命令時收到錯誤訊息，請確定您使用的是最新的 AWS CLI 版本。 通過在終端窗 GitHub 口中運行以下 Git 命令克隆搜索 ec2-instances-export-tags 存儲庫： `git clone https://github.com/aws-samples/search-ec2-instances-export-tags.git`	DevOps 工程師
安裝並啟動虛擬環境。	1. 通過運行以下命令來安裝虛擬環境： `python3 -m pip install virtualenv` 2. 執行下列命令來建立新的虛擬環境： `python3 -m venv env`	DevOps 工程師

任務	描述	所需技能
	3. 執行下列命令以啟動新的虛擬環境： `source env/bin/activate` 如需詳細資訊，請參閱虛擬使用者指南。	
安裝依存項目。	1. 通過在終端中運行以下命令打開代碼目錄： `cd search-ec2-instances-export-tags` 2. 通過運行以下 pip 命令來安裝該`requirements.txt` 文件： `pip3 install -r requirements.txt`	DevOps 工程師
設定 AWS 命名的設定檔。	如果您還沒有，請設定包含執行指令碼所需登入資料的 AWS 命名設定檔。要創建一個命名的配置文件，請運行 aws 配置命令。 如需詳細資訊，請參閱 AWS CLI 文件中的使用命名設定檔。	DevOps 工程師

設定並執行 Python 指令碼

任務	描述	所需技能
建立輸入檔案。	建立輸入檔案，其中包含您希望指令碼搜尋和匯出標籤的 Amazon EC2 執行個體清單。您可以列出執行個體 ID、私人 IPv4 位址或公用 IPv4 位址。 重要事項：請確定每個 Amazon EC2 執行個體都列在輸入檔案中各自的行上。 輸入檔案範例 `1 i-0547c351bdfe85b9f` `2 54.157.194.156` `3 172.31.85.33` `4 54.165.198.144` `5 i-0b6223b5914111a4b` `6 172.31.85.44` `7 54.165.198.145` `8 172.31.80.219` `9 172.31.94.199`	DevOps 工程師
執行 Python 指令碼。	通過在終端中運行以下命令來運行腳本： `python search_instances.py -i INPUTFILE -o OUTPUTFILE -r REGION [-p PROFILE]` 注意：`INPUTFILE` 替換為輸入文件的名稱。`OUTPUTFILE` 替換為您要為 CSV 輸出文	DevOps 工程師

任務	描述	所需技能
	件提供的名稱。REGION以您的 Amazon EC2 資源所在的 AWS 區域取代。如果您使用的是 AWS 命名設定檔，請以您正在使用的具名設定檔取PROFILE代。 若要取得支援參數及其說明的清單，請執行下列命令： `python search_instances.py -h` 有關更多信息並查看輸出文件示例，請參閱 GitHub search-ec2-instances-export-tags 存儲庫中的README.md 文件。	

相關資源

- 設定 AWS CLI (AWS CLI 使用者指南)

使用對流圈產生包含 AWS 組態受管規則的 AWS CloudFormation 範本

由盧卡斯國家（AWS）和弗雷迪·威爾遜（AWS）創建

環境：生產	技術：管理與治理 DevOps；安全性、身分識別、合規	工作負載：Microsoft；開源
AWS 服務：AWS Config；AWS CloudFormation		

Summary

許多組織使用 AWS Config 受管規則，根據常見的最佳實務來評估其 Amazon Web 服務 (AWS) 資源的合規性。不過，這些規則可能很耗時維護，而且此模式可協助您利用 Python 程式庫 Troposphere 來產生和管理 AWS Config 受管規則。

此模式可協助您管理 AWS Config 受管規則，方法是使用 Python 指令碼將包含 AWS 受管規則的 Microsoft Excel 試算表轉換為 AWS CloudFormation 範本。對流圈充當基礎結構代碼（IAC），這意味著您可以使用託管規則更新 Excel 電子表格，而不是使用 JSON 或 YAML 格式的文件。然後，您可以使用範本啟動 AWS CloudFormation 堆疊，在 AWS 帳戶中建立和更新受管規則。

AWS CloudFormation 範本使用 Excel 試算表定義每個 AWS Config 受管規則，並協助您避免在 AWS 管理主控台中手動建立個別規則。指令碼會將每個受管理規則的參數預設為空白字典，以及範圍的`ComplianceResourceTypes`預設值`THE_RULE_IDENTIFIER.template file`。 如需規則識別碼的詳細資訊，請參閱 AWS Config 文件中的使用 AWS CloudFormation 範本建立 AWS Config 受管規則。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 熟悉使用 AWS CloudFormation 範本建立 AWS Config 受管規則。如需詳細資訊，請參閱 AWS Config 文件中的使用 AWS CloudFormation 範本建立 AWS Config 受管規則。
- Python 3，安裝和配置。有關這方面的更多信息，請參閱 Python 文檔。

- 現有的整合式開發環境 (IDE)，例如 AWS Cloud9。如需這方面的詳細資訊，請參閱什麼是 AWS Cloud9？ 在 AWS Cloud9 文件中。
- 在範例 `excel_config_rules.xlsx` Excel 試算表 (附加) 的欄中識別您的組織單位 (OU)。

史诗

自訂和設定 AWS Config 受管規則

任務	描述	所需技能
更新範例 Excel 試算表。	下載範例 `excel_config_rules.xlsx` Excel 試算表 (隨附) 和標籤做為`Implemented` 您要使用的 AWS Config 受管規則。 標記為的規則`Implemented` 將新增至 AWS CloudFormation 範本。	開發人員
(選擇性) 使用 AWS 組態規則參數更新組態規則檔案。	某些 AWS Config 受管規則需要參數，而且應該使用`--param-file` 選項以 JSON 檔案的形式傳遞至 Python 指令碼。例如，`access-keys-rotated` 受管理規則使用下列`maxAccessKeyAge` 參數： `{ "access-keys-rotated": { "InputParameters": { "maxAccessKeyAge": 90 } } }`	開發人員

任務	描述	所需技能
	在此範例參數中maxAccess KeyAge ，設定為 90 天。指令碼會讀取參數檔案，並新增找到InputParameters 的任何參數檔案。	
(選擇性) 使用 AWS Config 組態更新組態規則 _ 參數 .json 檔案。 ComplianceResource Types	在預設情況下，Python 指令碼會ComplianceResource Types 從 AWS 定義的範本擷取。如果您想要覆寫特定 AWS Config 受管規則的範圍，則需要使用--param-file 選項將其作為 JSON 檔案傳遞至 Python 指令碼。 例如，下列範例程式碼顯示如何將 for 設定ComplianceResourceTypes ec2-volume-inuse-check 為["AWS::EC2::Volume"] 清單： `{ "ec2-volume-inuse-check": { "Scope": { "ComplianceResourceTypes": [ "AWS::EC2::Volume" ] } } }`	開發人員

運行 Python 本

任務	描述	所需技能
從 requirements.txt 檔案安裝點子套件。	下載該`requirements.txt` 文件（附件）並在 IDE 中運行以下命令以安裝 Python 軟件包： `pip3 install -r requirements.txt`	開發人員
執行 Python 指令碼。	1. 將`aws_config_rules.py` 文件（附件）下載到本地計算機。 2. 執行 - `python3 aws_config_rules.py --ou <OU_NAME>` 命令。附註：`--ou`定義要在 Excel 試算表中選擇的 OU 欄。 您也可以新增下列選用參數： • `--config-rule-option` — 定義要從 Excel 試算表中選擇的規則。預設值為`Implemented` 參數。 • `--excel-file` — Excel 試算表的路徑。預設值為 `aws_config_rules.xlsx` 。 • `--param-file` — 參數 JSON 檔案的路徑。預設值為 `config_rules_params.json` 。	開發人員

任務	描述	所需技能
	• `--max-execution-frequency` — 定義 AWS Config 受管規則的評估頻率。選項包括`One_HourThree_Hours` 、`Six_Hours` 、`Twelve_Hours` 、或`TwentyFour_Hours` 。預設值為 `TwentyFour_Hours` 。	

部署 AWS Config 受管規則

任務	描述	所需技能
啟動 AWS CloudFormation 堆疊。	1. 登入 AWS 管理主控台，開啟 AWS 主 CloudFormation 控台，然後選擇 [建立堆疊]。 2. 在 [指定範本] 頁面上，選擇 [上傳範本檔案]，然後上傳 AWS CloudFormation 範本。 3. 指定堆疊名稱，然後選擇 [下一步]。 4. 指定標籤，然後選擇 [下一步]。 5. 選擇建立堆疊。	開發人員

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

讓 SageMaker 筆記本執行個體暫時存取另一個 AWS 帳戶中的 CodeCommit 儲存庫

創建者：海爾格奧夫德海德（AWS）

環境：生產	技術： DevOps; 分析; 機器學習和人工智能; 管理與治理	AWS 服務：AWS CodeCommit；AWS Identity and Access Management；Amazon SageMaker

Summary

此模式顯示如何授予 Amazon SageMaker 筆記本執行個體和使用者臨時 CodeCommit 存取另一個 AWS 帳戶中的 AWS 儲存庫。此模式還顯示了如何為每個實體可以在每個存儲庫上執行的特定操作授予細微的權限。

Organizations 通常會將存 CodeCommit 放庫存放在與託管其開發環境的帳戶不同的 AWS 帳戶中。此多帳戶設定有助於控制存放庫的存取，並降低意外刪除的風險。若要授予這些跨帳戶許可，最佳實務是使用 AWS Identity and Access Management (IAM) 角色。然後，每個 AWS 帳戶中預先定義的 IAM 身分可以暫時扮演角色，跨帳戶建立受控的信任鏈。

附註：您可以套用類似的程序來授與其他 IAM 身分對 CodeCommit 儲存庫的跨帳戶存取權限。如需詳細資訊，請參閱 AWS 使用 CodeCommit 者指南中的使用角色設定對 AWS CodeCommit 儲存庫的跨帳戶存取。

先決條件和限制

先決條件

- 具有 CodeCommit 儲存庫 (帳戶 A) 的作用中 AWS 帳戶
- 具有 SageMaker 筆記本執行個體 (帳戶 B) 的第二個有效 AWS 帳戶
- 具有足夠許可的 AWS 使用者，可在帳戶 A 中建立和修改 IAM 角色
- 具有足夠許可的第二個 AWS 使用者，可在帳戶 B 中建立和修改 IAM 角色

架構

下圖顯示授予 SageMaker 筆記本執行個體和一個 AWS 帳戶跨帳戶存取 CodeCommit 存放庫中的使用者的工作流程範例：

該圖顯示以下工作流程：

1. 帳戶 B 中的 AWS 使用者角色和 SageMaker 筆記本執行個體角色採用具名設定檔。
2. 具名設定檔的權限原則會在帳戶 A 中指定設定檔隨後假設的 CodeCommit 存取角色。
3. 帳戶 A 中 CodeCommit 存取角色的信任策略允許帳戶 B 中具名的設定檔擔任 CodeCommit 存取角色。
4. 帳戶 A 中 CodeCommit 存放庫的 IAM 許可政策允許存 CodeCommit 取角色 CodeCommit 存取存放庫。

技術, 堆

- CodeCommit
- Git
- IAM
- pip
- SageMaker

工具

- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Git 是一個分佈式版本控制系統，用於在軟件開發過程中跟踪源代碼的變化。
- git-remote-codecommit是一個實用程序，可以幫助您通過擴展 Git 從 CodeCommit 存儲庫中推送和提取代碼。
- 點子是 Python 的軟件包安裝程序。您可以使用 pip 從 Python 軟件 Package 索引和其他索引安裝軟件包。

最佳實務

使用 IAM 政策設定許可時，請務必僅授與執行工作所需的權限。如需詳細資訊，請參閱 IAM 文件中的套用最低權限許可。

實施此模式時，請確保執行以下操作：

- 確認 IAM 原則僅具有在每個存放庫中執行特定必要動作所需的許可。例如，建議允許已核准的 IAM 原則將變更推送和合併到特定的儲存庫分支，但只要求合併到受保護的分支。
- 確認 IAM 原則會根據各自的角色和每個專案的職責，指派不同的 IAM 角色。例如，開發人員擁有與發行管理員或 AWS 管理員不同的存取權限。

史诗

設定 IAM 角色

任務	描述	所需技能
設定 CodeCommit 存取角色和權限原則。	注意：若要自動化此史詩中記錄的手動設定程序，您可以使用 AWS CloudFormation 範本。 在包含 CodeCommit 存放庫 (帳戶 A) 的帳戶中，執行下列動作： 1. 建立可由帳戶 B 中的 SageMaker 筆記本執行個體角色扮演的 IAM 角色。 2. 建立可授與存放庫存取權的 IAM 政策，並將政策附加至該角色。僅用於測試目的，請選擇 AWSCodeCommitPowerUserAWS 受管政策。除了刪除資源的能力外，此原則會授與所有CodeCommit 權限。	一般 AWS、AWS DevOps

任務	描述	所需技能
	3. 修改角色的信任原則，讓帳戶 B 列為受信任的實體。 重要事項：在將此設定移至生產環境之前，最佳做法是撰寫自己的 IAM 政策以套用最低權限許可。如需詳細資訊，請參閱此模式的其他資訊一節。	

任務	描述	所需技能
授與 SageMaker 記事本執行個體在帳戶 B 中的角色權限，以在帳戶 A 中擔任 CodeCommit 存取角色。	在包含 SageMaker 筆記本執行個體 IAM 角色 (帳戶 B) 的帳戶中，執行下列動作： 1. 建立 IAM 政策，允許 IAM 角色或使用者在帳戶 A 中擔任 CodeCommit 存取角色。 允許 IAM 角色或使用者擔任跨帳戶角色的 IAM 許可政策範例 `{ "Version": "2012-10-17", "Statement": [ { "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam:::accountA_ID:role/accountArole_ID" } }` 2. 將原則附加至帳戶 B 中 SageMaker 記事本執行個體的角色。 3. 讓 SageMaker 筆記本執行個體在帳戶 B 中的角色扮演帳戶 A 中的 CodeCommit 存取角色。	一般 AWS、AWS DevOps

任務	描述	所需技能
	注意：若要檢視儲存庫的 Amazon 資源名稱 (ARN)，請參閱 AWS CodeCommit 使用者指南中的檢視 CodeCommit 儲存庫詳細資訊。	

在帳戶 B 中設定您的 SageMaker 記事本執行個體

任務	描述	所需技能
在 AWS SageMaker 筆記本執行個體上設定使用者設定檔以擔任帳戶 A 中的角色。	重要：請確定您已安裝最新版本的 AWS Command Line Interface (AWS CLI) (AWS CLI)。 在包含 SageMaker 記事本執行個體 (帳戶 B) 的帳戶中，執行下列動作： 1. 登入 AWS 管理主控台，並開啟 SageMaker 主控台。 2. 存取 SageMaker 筆記本執行個體。隨即開啟 Jupyter 介面。 3. 選擇 [新增]，然後選擇 [終端機]。Jupyter 環境中會開啟一個新的終端機視窗。 4. 瀏覽至 SageMaker 筆記本執行個體的 ~/. aws/config 檔案。然後，輸入下列陳述式，將使用者設定檔新增至檔案：	一般 AWS、AWS DevOps

任務	描述	所需技能
	`` ------.aws/config- ------------- [profile remoterep ouser] role_arn = arn:aws:i am::<ID of Account A>:role/<rolename> role_session_name = remoteaccesssession region = eu-west-1 credential_source = Ec2InstanceMetadata ---------------- ----------------- ```	
安裝 git-remote-codecommit 公用程式。	按照 AWS CodeCommit 使用者指南中的步驟 2：安裝 git-remote-codecommit中的指示進行操作。	資料科學家

存取儲存庫

任務	描述	所需技能
使用 Git 命令或 CodeCommit 存取儲存庫 SageMaker。	若要使用 Git 假設帳戶 B 中的 SageMaker 筆記本執行個體角色的 IAM 主體現在可以執行 Git 命令來 CodeCommit 存取帳戶 A 中的儲存庫。例如，使用者可以執行`git clone`、`git pull`和`git push`等命令。 如需指示，請參閱 AWS CodeCommit 使用者指南中的	Git，bash 控制台

任務	描述	所需技能
	Connect 到 AWS CodeCommit 儲存庫。 如需如何搭配使用 Git 的詳細資訊 CodeCommit，請參閱 AWS 使用 CodeCommit 者指南 CodeCommit中的 AWS 入門。 若要使用 SageMaker 若要從 SageMaker 主控台使用 Git，您必須允許 Git 從您的 CodeCommit 儲存庫擷取憑證。如需指示，請參閱 SageMaker 文件中的將不同 AWS 帳戶中的 CodeCommit 儲存庫與筆記本執行個體建立關聯。	

相關資源

- 使用角色設定對 AWS CodeCommit 儲存庫的跨帳戶存取 (AWS CodeCommit 文件)
- IAM 教學課程：使用 IAM 角色 (IAM 文件) 在 AWS 帳戶之間委派存取權

其他資訊

將權限 CodeCommit 限制為特定動作

若要限制 IAM 主體可在 CodeCommit 存放庫中執行的動作，請修改存 CodeCommit 取政策中允許的動作。

如需 CodeCommit API 操作的詳細資訊，請參閱 CodeCommit AWS CodeCommit 使用者指南中的許可參考。

注意：您也可以編輯 AWSCodeCommitPowerUserAWS 受管政策以符合您的使用案例。

限制特定儲 CodeCommit 存庫的權限

若要建立只有特定使用者可存取多個程式碼存放庫的多租戶環境，請執行下列動作：

1. 在帳戶 A 中建立多個 CodeCommit 存取角色，然後設定每個存取角色的信任原則，以允許帳戶 B 中的特定使用者擔任該角色。
2. 將 「Resource」 條件新增至每個存 CodeCommit 取角色的政策，以限制每個角色可以承擔的程式碼儲存庫。

限制 IAM 主體 CodeCommit 存取特定儲存庫的「資源」條件範例

```
"Resource" : [<REPOSITORY_ARN>,<REPOSITORY_ARN> ]
```

注意：為了協助識別和區分同一 AWS 帳戶中的多個程式碼儲存庫，您可以為存放庫的名稱指派不同的前置詞。例如，您可以使用符合不同開發人員群組的前綴來命名程式碼儲存庫，例如我的專案子專案 1-repo1 和我的專案子專案 2-repo1。然後，您可以根據每個開發人員群組指派的前置詞建立 IAM 角色。例如，您可以建立名為我的專案-子專案 1 重新存取的角色，並將其存取權授與所有包含前置詞 myproject subprojec t1 的程式碼儲存庫。

參照包含特定前綴的代碼存儲庫 ARN 的示例「Resource」條件

```
"Resource" : arn:aws:codecommit:<region>:<account-id>:myproject-subproject1-*
```

為多帳戶環境實作 GitHub Flow 分支 DevOps 策略

由邁克·斯蒂芬斯（AWS）和阿布拉什·維諾德（AWS）創建

代碼存儲庫：git-branching-strategies-for-多帳戶開發	環境：生產	技術: DevOps; 軟件開發和測試; 多賬戶策略
AWS 服務：AWS CodeArtifact；AWS CodeBuild；AWS；AWS CodeCommit；AWS CodeDeploy；AWS CodePipeline		

Summary

管理原始程式碼儲存庫時，不同的分支策略會影響開發團隊所使用的軟體開發和發行程序。常見分支策略的例子包括幹線， GitHub 流量和 Gitflow。這些策略使用不同的分支，並且在每個環境中執行的活動都不同。正在實施 DevOps 流程的 Organizations 將受益於視覺化指南，以幫助他們了解這些分支策略之間的差異。在您的組織中使用此視覺效果可協助開發團隊調整工作並遵循組織標準。此模式提供此視覺效果，並說明在組織中實作 GitHub Flow 分支策略的程序。

此模式是有關為多 AWS 帳戶個組織選擇和實施分 DevOps 支策略的文件系列的一部分。本系列旨在幫助您從一開始就應用正確的策略和最佳實務，以簡化您在雲端中的體驗。 GitHub Flow 只是組織可以使用的其中一種可能的分支策略。本文檔系列還涵蓋了幹線和 Gitflow 分支模型。如果您尚未這麼做，建議您先檢閱選擇多帳戶 DevOps 環境的 Git 分支策略，然後再實作此模式的指引。請使用盡職調查為您的組織選擇正確的分支策略。

本指南提供了一個圖表，顯示組織如何實施 GitHub Flow 策略。建議您檢閱 AWS Well-Architected 的 DevOps 指引，以檢閱最佳做法。此模式包含 DevOps 程序中每個步驟的建議工作、步驟和限制。

先決條件和限制

先決條件

- Git，已安裝。這被用作源代碼存儲庫工具。
- 繪圖 IO，已安裝。此應用程序用於查看和編輯圖表。

架構

目標架構

下圖可以像一個普內特廣場（維基百科）使用。您可以將垂直軸上的分支與水平軸上的 AWS 環境對齊，以決定在每個案例中要執行的動作。數字表示工作流程中動作的順序。此範例會引導您從分`feature`支到生產環境中的部署。

如需 GitHub Flow 方法中的 AWS 帳戶、環境和分支的詳細資訊，請參閱選擇多帳戶 DevOps 環境的 Git 分支策略。

自動化和規模

持續整合與持續交付 (CI/CD) 是自動化軟體發行生命週期的程序。它會自動執行傳統上從初始提交到生產環境中取得新程式碼所需的大部分或全部手動程序。CI/CD 管線包含沙箱、開發、測試、測試和生產環境。在每個環境中，CI/CD 管線會佈建部署或測試程式碼所需的任何基礎結構。 通過使用 CI/CD，開發團隊可以對代碼進行更改，然後自動測試和部署。CI/CD 管道還透過強制執行功能接受和部署的一致性、標準、最佳實務和最低接受程度，為開發團隊提供治理和防護。 如需詳細資訊，請參閱 < 在上實踐持續整合和持續交付 > AWS。

AWS 提供一套開發人員服務，可協助您建置 CI/CD 管線。例如，這AWS CodePipeline是一項全受管的持續交付服務，可協助您將發行管道自動化，進行快速可靠的應用程式和基礎結構更新。 AWS CodeCommit旨在安全地託管可擴展的 Git 存儲庫，並AWS CodeBuild編譯源代碼，運行測試以及生成 ready-to-deploy 軟件包。如需詳細資訊，請參閱上的開發人員工具 AWS。

工具

AWS 服務和工具

AWS 提供了一套開發人員服務，您可以用來實現此模式：

- AWS CodeArtifact是可高度擴充的受管理成品儲存庫服務，可協助您儲存和共用應用程式開發的軟體套件。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit是一項版本控制服務，可協助您私下儲存和管理 Git 儲存庫，而無需管理您自己的原始檔控制系統。

- AWS CodeDeploy自動部署到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、 AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。

其他工具

- Draw.io 桌面是用於製作流程圖和圖表的應用程序。程式碼儲存庫包含適用於 Draw.io 的 .drawio 格式的範本。
- Figma 是一個專為協作而設計的在線設計工具。該代碼庫包含用於 Figma 的 .fig 格式的模板。

代碼存儲庫

此模式中圖表的原始檔案位於「 GitHub Git GitHub 流程分支策略」儲存庫中。它包括 PNG，繪製 .io 和 Figma 格式的文件。您可以修改這些圖表以支援組織的處理。

最佳實務

請遵循 AWS Well-Architected 的 DevOps 指導和選擇適用於多帳戶環境的 Git 分支策略中的最佳做法和建議。 DevOps 這些可協助您有效地實作 GitHub 以流程為基礎的開發、促進協同合作、改善程式碼品質，以及簡化開發程序。

史诗

檢閱 GitHub 流程工作流程

任務	描述	所需技能
複查標準「 GitHub 流程」處理。	1. 在沙箱環境中，開發人員從`feature`分支創建一個`main`分支並使用命名模式 `feature/<ticket>_<initials>_<short description>` 。 2. 開發人員向`feature`分支添加一個或多個提交，每個提	DevOps 工程師

任務	描述	所需技能
	交代表一個離散的更改或改進。 3. 開發人員打開合併請求（MR）以將更改合併到`main`分支中。這會啟動檢閱程序。 4. 在審查過程中，開發人員會討論代碼更改並提供反饋。我們的目標是確保這些變化具有高質量並符合項目的標準。 5. 開發人員建立合併要求之後，自動建置程序便會啟動，並將`feature`分支中的變更部署至開發環境。 6. 自動化測試可驗證合併請求中封裝的更改的完整性和質量。要完成合併請求，需要成功的構建，成功部署和成功的測試。 7. 當審核程序完成時，變更會合併到`main`分支中。 8. 核准人手動核准發行成品部署至測試環境。 9. 核准者會手動核准發行人工因素部署至測試環境。 10核准人手動核准發行人工因素部署至生產環境。	

任務	描述	所需技能
檢閱錯誤修正 GitHub 流程程序。	1. 開發人員從分bugfix支創建一個main分支並使用命名模式bugfix/<ticket number>_<developer initials>_<descriptor> 。 2. 開發人員修復了問題，提交修復並構建bugfix分支。 3. 開發人員打開合併請求以將bugfix分支合併到main分支中。這會啟動檢閱程序。 4. 在審查過程中，開發人員會討論代碼更改並提供反饋。 5. 審查完成和批准後，開發人員將完成分支機bugfix構的合併請求到分main支機構。 6. 核准人手動核准發行人工因素部署至較高環境。	DevOps 工程師

任務	描述	所需技能
檢閱修補程式 GitHub 流程程序。	GitHub Flow 旨在實現持續交付，其中代碼更改頻繁且可靠地部署到更高的環境。關鍵是每個`feature`分支都可以隨時部署。 `Hotfix`類似於`feature`或分支的`bugfix`分支，可以遵循與這些其他分支相同的過程。但是，鑑於它們的緊迫性，hotfix 通常具有更高的優先級。根據團隊的政策和情況的即時性，該過程中的某些步驟可以加快。例如，hotfix 的程式碼檢閱可能會快速追蹤。因此，雖然 hotfix 程序並行的功能或錯誤修正程序，周圍 hotfix 的緊急程度可能會保證在程序遵守的修改。它是至關重要的建立有關管理 hotfix，以確保他們處理有效和安全的指導方針。	DevOps 工程師

故障診斷

問題	解決方案
分支衝突	GitHub 流程模型可能發生的一個常見問題是需要在生產環境中發生 hotfix，但需要在修改相同資源的`featurebugfix`、或`hotfix`分支中發生對應的變更。我們建議您經常將變更合併`main`到較低分支中，以避免在合併到時發生重大衝突`main`。

問題	解決方案
團隊成熟度	GitHub Flow 鼓勵日常部署到更高的環境，實現真正的持續集成和持續交付（CI/CD）。團隊必須具備工程成熟度，才能為其建立功能並建立自動化測試。在核准變更之前，小組必須執行詳盡的合併請求審核。這促進了強大的工程文化，在開發過程中提高質量，責任性和效率。

相關資源

本指南不包含 Git 的訓練；不過，如果您需要這項訓練，網際網路上有許多高品質的資源可供使用。我們建議您從 Git 文件網站開始。

下列資源可協助您 GitHub 在 AWS 雲端.

AWS DevOps 指導

- AWS DevOps 指引
- AWS 部署管線參考架構
- 什麼是 DevOps？
- DevOps 資源

GitHub 流程引導

- GitHub 流程快速入門教學課程 () GitHub
- 為什麼要 GitHub 流動？

其他資源

- 十二因素應用程序方法（12 因素 .net）

為多帳戶環境實施 Gitflow 分支策略 DevOps

由邁克·斯蒂芬斯（AWS），斯蒂芬 DiCato （AWS），蒂姆·萬德海姆（AWS）和阿布拉什維諾德（AWS）創建

代碼存儲庫：git-branching-strategies-for-多帳戶開發	環境：生產	技術: DevOps; 軟件開發和測試; 多賬戶策略
AWS 服務：AWS CodeArtifact；AWS CodeBuild；AWS；AWS CodeCommit；AWS CodeDeploy；AWS CodePipeline		

Summary

管理原始程式碼儲存庫時，不同的分支策略會影響開發團隊所使用的軟體開發和發行程序。常見分支策略的例子包括幹線，Gitflow 和流量。 GitHub 這些策略使用不同的分支，並且在每個環境中執行的活動都不同。正在實施 DevOps 流程的 Organizations 將受益於視覺化指南，以幫助他們了解這些分支策略之間的差異。在您的組織中使用此視覺效果可協助開發團隊調整工作並遵循組織標準。此模式提供了此視覺效果，並描述了在組織中實施 Gitflow 分支策略的過程。

此模式是有關為多 AWS 帳戶個組織選擇和實施分 DevOps 支策略的文件系列的一部分。本系列旨在幫助您從一開始就應用正確的策略和最佳實務，以簡化您在雲端中的體驗。Gitflow 只是您的組織可以使用的一種可能的分支策略。本文件系列也涵蓋主幹和GitHub 流量分支模型。如果您尚未這麼做，建議您先檢閱選擇多帳戶 DevOps 環境的 Git 分支策略，然後再實作此模式的指引。請使用盡職調查為您的組織選擇正確的分支策略。

本指南提供了一個圖表，顯示組織如何實施 Gitflow 策略。建議您檢閱 AWS Well-Architected 的 DevOps 指引，以檢閱最佳做法。此模式包含 DevOps 程序中每個步驟的建議工作、步驟和限制。

先決條件和限制

先決條件

- Git，已安裝。這被用作源代碼存儲庫工具。
- 繪圖 IO，已安裝。此應用程序用於查看和編輯圖表。

- （可選）Gitflow 插件，已安裝。

架構

目標架構

下圖可以像一個普內特廣場（維基百科）使用。您可以將垂直軸上的分支與水平軸上的 AWS 環境對齊，以決定在每個案例中要執行的動作。數字表示工作流程中動作的順序。此範例會引導您從功能分支到生產環境中的部署。

有關 Gitflow 方法中的 AWS 帳戶，環境和分支的更多信息，請參閱為多帳戶環境選擇 Git 分支策略。
DevOps

自動化和規模

持續整合與持續交付 (CI/CD) 是自動化軟體發行生命週期的程序。它會自動執行傳統上從初始提交到生產環境中取得新程式碼所需的大部分或全部手動程序。CI/CD 管線包含沙箱、開發、測試、測試和生產環境。在每個環境中，CI/CD 管線會佈建部署或測試程式碼所需的任何基礎結構。 通過使用 CI/CD，開發團隊可以對代碼進行更改，然後自動測試和部署。CI/CD 管道還透過強制執行功能接受和部署的一致性、標準、最佳實務和最低接受程度，為開發團隊提供治理和防護。 如需詳細資訊，請參閱 < 在上實踐持續整合和持續交付 > AWS。

AWS 提供一套開發人員服務，可協助您建置 CI/CD 管線。例如，這AWS CodePipeline是一項全受管的持續交付服務，可協助您將發行管道自動化，進行快速可靠的應用程式和基礎結構更新。 AWS CodeCommit旨在安全地託管可擴展的 Git 存儲庫，並AWS CodeBuild編譯源代碼，運行測試以及生成 ready-to-deploy 軟件包。如需詳細資訊，請參閱上的開發人員工具 AWS。

工具

AWS 服務和工具

AWS 提供了一套開發人員服務，您可以用來實現此模式：

- AWS CodeArtifact是可高度擴充的受管理成品儲存庫服務，可協助您儲存和共用應用程式開發的軟體套件。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。

- AWS CodeCommit是一項版本控制服務，可協助您私下儲存和管理 Git 儲存庫，而無需管理您自己的原始檔控制系統。
- AWS CodeDeploy自動部署到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、 AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。

其他工具

- Draw.io 桌面是用於製作流程圖和圖表的應用程序。程式碼儲存庫包含適用於 Draw.io 的 .drawio 格式的範本。
- Figma 是一個專為協作而設計的在線設計工具。該代碼庫包含用於 Figma 的 .fig 格式的模板。
- （可選）Gitflow 插件是 Git 擴展的集合，可為 Gitflow 分支模型提供高級存儲庫操作。

代碼存儲庫

此模式中圖表的原始檔案可在 GitFlow儲存庫的 GitHub Git 分支策略中找到。它包括 PNG，繪製 .io 和 Figma 格式的文件。您可以修改這些圖表以支援組織的處理。

最佳實務

請遵循 AWS Well-Architected 的 DevOps 指導和選擇適用於多帳戶環境的 Git 分支策略中的最佳做法和建議。 DevOps 這些幫助您有效地實施基於 GITFlow 的開發，促進協作，提高代碼質量並簡化開發過程。

史诗

檢閱 Gitflow 工作流程

任務	描述	所需技能
檢閱標準的 Gitflow 程序。	1. 在沙箱環境中，開發人員從`feature`分支創建一個`develop`分支並使用命名模式`feature/<ticket>_<initials>`	DevOps 工程師

任務	描述	所需技能
	_<short description> 。 2. 開發人員開發程式碼，並以反覆方式將程式碼部署至沙箱環境，以完成票證。 注意：開發人員可以選擇性地建立sandbox分支，以便在沙箱環境中執行自動化建置或部署管道。 3. 開發人員通過使用壁球合併創建從develop分支到分支的合併請求。feature 4. 持續整合和持續交付 (CI/CD) 管線會自動建置並部署develop分支至開發環境。 5. （可選）開發人員在繼續發布活動之前將其他feature分支集成到開發分支中。 6. 當您準備好釋放develop分支中的功能時，開發人員會創建一個release/v <number> 從release分支命名的develop分支。 7. 開發人員構建發布分支，該分支發布成品以便在其他環境中重複使用。 8. 核准人手動核准發行成品部署至測試環境。 9. 核准者會手動核准發行人工因素部署至測試環境。	

任務	描述	所需技能
	10.核准人手動核准發行人工因素部署至生產環境。 11.開發人員將release分支合併到分main支中。理想情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。 12.開發人員將release分支合併到分develop支中。理想情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。	

任務	描述	所需技能
檢閱修補程式 Gitflow 程序。	1. 開發人員從分hotfix支創建一個main分支並使用命名模式hotfix/<ticket>_<initials>_<short description> 。 2. 開發人員從分release支創建一個main分支並命名它release/v<number> 。 3. 開發人員修復了問題，提交修復並構建hotfix分支。 4. 開發人員通過使用壁球合併創建從release/v<number> 分支到分支的合併請求。hotfix 5. 開發人員建置release分支，該分支會發佈成品以便在其他環境中重複使用。 6. 核准人手動核准發行成品部署至測試環境。 7. 核准者會手動核准發行人工因素部署至測試環境。 8. 核准人手動核准發行人工因素部署至生產環境。 9. 開發人員將release分支合併到分main支中。理想情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。 10.開發人員將release分支合併到分develop支中。理想	DevOps 工程師

任務	描述	所需技能
	情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。 11 如果偵測到衝突，開發人員會收到警示，並透過合併要求解決衝突。	

任務	描述	所需技能
查看錯誤修正 Gitflow 過程。	1. 開發人員從當前bugfix分支創建一個release/v<number> 分支並使用命名模式bugfix/<ticket number>_<developer initials>_<descriptor> 。 2. 開發人員修復了問題，提交修復並構建bugfix分支。 3. 開發人員通過使用壁球合併創建從release/v<number> 分支到分支的合併請求。bugfix 4. 開發人員建置release分支，該分支會發佈成品以便在其他環境中重複使用。 5. 核准人手動核准發行人工因素部署至測試環境。 6. 核准者會手動核准將發行人工因素部署至階段環境。 7. 核准人手動核准發行人工因素部署至生產環境。 8. 開發人員將release分支合併到分main支中。理想情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。 9. 開發人員將release分支合併到分develop支中。理想情況下，開發人員使用自動化腳本來執行快進合併。不要使用壁球合併。	DevOps 工程師

任務	描述	所需技能
	10如果偵測到衝突，開發人員會收到警示，並透過合併要求解決衝突。	

故障診斷

問題	解決方案
分支衝突	Gitflow 模型可能發生的一個常見問題是需要在生產環境中發生 hotfix，但需要在較低的環境中發生相應的變更，其中另一個分支正在修改相同的資源。我們建議您一次只有一個發行分支處於作用中狀態。如果您一次有多個作用中，環境中的變更可能會發生衝突，而且您可能無法將分支移至生產環境。
合併	發行版本應該合併回 main 並儘快開發，以將工作合併回主要分支。
壁球合併	只有在從分支合併到`feature`分支時才使用壁球合`develop`併。在較高的分支中使用壁球合併會導致將更改合併回較低分支時遇到困難。

相關資源

本指南不包含 Git 的訓練；不過，如果您需要這項訓練，網際網路上有許多高品質的資源可供使用。我們建議您從 Git 文件網站開始。

以下資源可以幫助您進行 Gitflow 分支旅程。 AWS 雲端

AWS DevOps 指導

- AWS DevOps 指引
- AWS 部署管線參考架構
- 什麼是 DevOps？

- DevOps 資源

吉流指引

- 原來的 Gitflow 博客（文森特·德森博客文章）
- Gitflow 工作流程 (大地圖)
- 啟用 Gitflow GitHub：如何使用 Git 流程工作流程搭配 GitHub 基礎存放庫 (影片) YouTube
- Git 流程初始化示例（YouTube 視頻）
- 從開始到結束的 Gitflow 發布分支（YouTube 視頻）

其他資源

十二因素應用程序方法（12 因素 .net）

為多帳戶環境實作幹線分支 DevOps 策略

由邁克·史蒂芬斯（AWS）和瑞揚威爾遜（AWS）創建

代碼存儲庫：git-branching-strategies-for-多帳戶開發	環境：生產	技術: DevOps; 軟件開發和測試; 多賬戶策略
AWS 服務：AWS CodeArtifact；AWS CodeBuild；AWS；AWS CodeCommit；AWS CodeDeploy；AWS CodePipeline		

Summary

管理原始程式碼儲存庫時，不同的分支策略會影響開發團隊所使用的軟體開發和發行程序。常見分支策略的例子包括幹線， GitHub 流量和 Gitflow。這些策略使用不同的分支，並且在每個環境中執行的活動都不同。正在實施 DevOps 流程的 Organizations 將受益於視覺化指南，以幫助他們了解這些分支策略之間的差異。在您的組織中使用此視覺效果可協助開發團隊調整工作並遵循組織標準。此模式提供此視覺效果，並說明在組織中實作 Trunk 分支策略的程序。

此模式是有關為多 AWS 帳戶個組織選擇和實施分 DevOps 支策略的文件系列的一部分。本系列旨在幫助您從一開始就應用正確的策略和最佳實務，以簡化您在雲端中的體驗。Trunk 只是您的組織可以使用的一種可能的分支策略。本文檔系列還涵蓋了 GitHub Flow 和 Gitflow 分支模型。如果您尚未這麼做，建議您先檢閱選擇多帳戶 DevOps 環境的 Git 分支策略，然後再實作此模式的指引。請使用盡職調查為您的組織選擇正確的分支策略。

本指南提供了一個圖表，顯示組織如何實施 Trunk 策略。建議您查看官方的 AWS Well-Architected DevOps 指南，以檢閱最佳做法。此模式包含 DevOps 程序中每個步驟的建議工作、步驟和限制。

先決條件和限制

先決條件

- Git，已安裝。這被用作源代碼存儲庫工具。
- 繪製 .io，已安裝。此應用程序用於查看和編輯圖表。

架構

目標架構

下圖可以像一個普內特廣場（維基百科）使用。您可以將垂直軸上的分支與水平軸上的 AWS 環境對齊，以決定在每個案例中要執行的動作。數字表示工作流程中動作的順序。此範例會引導您從分`feature`支到生產環境中的部署。

如需 Trunk 方法中的 AWS 帳戶、環境和分支的詳細資訊，請參閱選擇多帳戶 DevOps 環境的 Git 分支策略。

自動化和規模

持續整合與持續交付 (CI/CD) 是自動化軟體發行生命週期的程序。它會自動執行傳統上從初始提交到生產環境中取得新程式碼所需的大部分或全部手動程序。CI/CD 管線包含沙箱、開發、測試、測試和生產環境。在每個環境中，CI/CD 管線會佈建部署或測試程式碼所需的任何基礎結構。 通過使用 CI/CD，開發團隊可以對代碼進行更改，然後自動測試和部署。CI/CD 管道還透過強制執行功能接受和部署的一致性、標準、最佳實務和最低接受程度，為開發團隊提供治理和防護。 如需詳細資訊，請參閱 < 在上實踐持續整合和持續交付 > AWS。

AWS 提供一套開發人員服務，可協助您建置 CI/CD 管線。例如，這AWS CodePipeline是一項全受管的持續交付服務，可協助您將發行管道自動化，進行快速可靠的應用程式和基礎結構更新。 AWS CodeCommit旨在安全地託管可擴展的 Git 存儲庫，並AWS CodeBuild編譯源代碼，運行測試以及生成 ready-to-deploy 軟件包。如需詳細資訊，請參閱上的開發人員工具 AWS。

工具

AWS 服務和工具

AWS 提供了一套開發人員服務，您可以用來實現此模式：

- AWS CodeArtifact是可高度擴充的受管理成品儲存庫服務，可協助您儲存和共用應用程式開發的軟體套件。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit是一項版本控制服務，可協助您私下儲存和管理 Git 儲存庫，而無需管理您自己的原始檔控制系統。

- AWS CodeDeploy自動部署到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、 AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。

其他工具

- Draw.io 桌面 — 用於製作流程圖和圖表的應用程序。
- Figma 是一個專為協作而設計的在線設計工具。該代碼庫包含用於 Figma 的 .fig 格式的模板。

代碼存儲庫

此模式中圖表的原始檔案可在中繼儲存庫的 GitHub Git 分支策略中找到。它包括 PNG，繪製 .io 和 Figma 格式的文件。您可以修改這些圖表以支援組織的處理。

最佳實務

請遵循 AWS Well-Architected 的 DevOps 指導和選擇適用於多帳戶環境的 Git 分支策略中的最佳做法和建議。 DevOps 這些可協助您有效地實作以 Trunk 為基礎的開發、促進協同合作、改善程式碼品質，以及簡化開發流程。

史诗

複查幹線工作流程

任務	描述	所需技能
檢閱標準幹線程序。	1. 在沙箱環境中，開發人員從`feature`分支創建一個`main`分支並使用命名模式 `feature/<ticket>_<initials>_<short description>` 。 2. 開發人員開發程式碼，並以反覆方式將程式碼部署至沙箱環境，以完成票證。	DevOps 工程師

任務	描述	所需技能
	注意：開發人員可以選擇性地建立sandbox分支，在沙箱環境中執行自動化建置或部署管道。 3. 開發人員通過使用壁球合併創建從main分支到分支的合併請求。feature 4. 持續整合與持續交付 (CI/CD) 管線會自動建置成品，並將其從main分支發佈至開發環境。 5. 核准人手動核准發行成品部署至開發環境。 6. 核准人手動核准發行成品部署至測試環境。 7. 核准者會手動核准發行人工因素部署至測試環境。 8. 核准人手動核准發行人工因素部署至生產環境。	

故障診斷

問題	解決方案
分支衝突	Trunk 模型可能發生的一個常見問題是，其中一個 hotfix 需要發生在生產環境中，但相應的變更需要發生在feature分支，其中相同的資源正在修改。我們建議您經常將變更合併main到較低分支中，以避免在合併到時發生重大衝突main。

相關資源

本指南不包含 Git 的訓練；不過，如果您需要這項訓練，網際網路上有許多高品質的資源可供使用。我們建議您從 Git 文件網站開始。

下列資源可協助您在 AWS 雲端.

AWS DevOps 指導

- AWS DevOps 指引
- AWS 部署管線參考架構
- 什麼是 DevOps？
- DevOps 資源

行李箱指引

- 基於幹線的開發

其他資源

- 十二因素應用程序方法

自動檢測更改並為中的龍斷啟動不同的 CodePipeline 管道 CodeCommit

創建者：赫爾頓里貝羅 (AWS)、巴塔利亞彼得洛 (AWS) 和里卡多莫拉斯 (AWS)

程式碼儲存庫：AWS CodeCommit 單回購多管道觸發程序	環境：PoC 或試點	技術： DevOps; 基礎架構; 無伺服器
AWS 服務：AWS CodeCommit；AWS CodePipeline；AWS		

Summary

此模式可協助您自動偵測中以龍斷為基礎之應用程式之原始程式碼的變更， AWS CodeCommit 然後啟動管線，在其中執行每 AWS CodePipeline 個微服務的持續整合和持續傳遞 (CI/CD) 自動化。這種方法意味著您的龍斷式應用程式中的每個微服務都可以擁有專用的 CI/CD 管道，以確保更好的可見性、更輕鬆的程式碼共用，以及改善協同作業、標準化和可探索性。

此模式中描述的解決方案不會在 monorepo 內部的微服務之間執行任何依賴性分析。它只檢測源代碼中的更改，並啟動匹配的 CI/CD 管道。

該模式使用 AWS Cloud9 作為整合式開發環境 (IDE)，並 AWS Cloud Development Kit (AWS CDK) 使用兩個 AWS CloudFormation 堆疊來定義基礎結構：`MonoRepoStack`和`PipelinesStack`. `MonoRepoStack`堆疊會在中建立單一回購，以 AWS CodeCommit 及啟動 CI/CD 管線的 AWS Lambda 函數。`PipelinesStack`堆疊會定義您的管線基礎結構。

重要事項：此模式的工作流程是概念證明 (PoC)。我們建議您僅在測試環境中使用它。如果您想要在生產環境中使用此模式的方法，請參閱 AWS Identity and Access Management (IAM) 文件中的 IAM 中的安全性最佳實務，並對 IAM 角色和進行必要的變更 AWS 服務。

先決條件和限制

先決條件

- 一個活躍的 AWS 帳戶。

- AWS Command Line Interface (AWS CLI)，已安裝並設定。如需詳細資訊，請參閱 AWS CLI 說明文件 AWS CLI中的〈安裝、更新和解除安裝〉。
- Python 3 和pip, 安裝在您的本地計算機上. 如需詳細資訊，請參閱 Python 文件。
- AWS CDK，已安裝並配置。如需詳細資訊，請參閱 AWS CDK 文件 AWS CDK中的入門。
- 一個 AWS Cloud9 IDE，已安裝和配置。如需詳細資訊，請參閱 AWS Cloud9 文件 AWS Cloud9中的設定。
- GitHub AWS CodeCommit monorepo 多管道觸發存儲庫，克隆到您的本地計算機上。
- 包含您要建置和部署之應用程式程式碼的現有目錄 CodePipeline。
- 熟悉和 DevOps 最佳做法的 AWS 雲端經驗。為了提高您的熟悉度 DevOps，您可以使用模式使用做法和 AWS 規範指導網站 AWS Cloud9上的微服 DevOps 務建立鬆散結合的架構。

架構

下圖顯示如何使用定 AWS CDK 義具有兩個 AWS CloudFormation 堆疊的基礎結構：MonoRepoStack和PipelinesStack。

該圖顯示以下工作流程：

1. 啟動程序會使 AWS CDK 用建立 AWS CloudFormation 堆疊MonoRepoStack和PipelinesStack.
2. MonoRepoStack堆疊會為您的應用程式建立 CodeCommit 儲存庫，以及在每次提交後啟動的 monorepo-event-handler Lambda 函數。
3. PipelinesStack堆疊會在中 CodePipeline 建立由 Lambda 函數初始化的管道。每個微服務都必須具有已定義的基礎結構管線。
4. 的管道由 Lambda 函數啟動，並根據中的原始程式碼啟動其隔離的 CI/CD 階段。microservice-n CodeCommit
5. 的管道由 Lambda 函數啟動，並根據中的原始程式碼啟動其隔離的 CI/CD 階段。microservice-1 CodeCommit

下圖顯示 AWS CloudFormation 堆疊MonoRepoStack和帳戶PipelinesStack中的部署。

1. 使用者變更應用程式其中一個微服務中的程式碼。
2. 用戶將更改從本地存儲庫推送到存 CodeCommit 儲庫。
3. 推送活動會啟動 Lambda 函數，該函數會接收到存放庫的所有推送。 CodeCommit
4. Lambda 函數會讀取參數存放區中的參數 AWS Systems Manager，此功能可擷取最新的提交 ID。參數具有命名格式：`/MonoRepoTrigger/{repository}/{branch_name}/LastCommit`。如果找不到參數，Lambda 函數會從儲存庫讀取最後一個提交 ID，並將傳回的值 CodeCommit 儲存在參數存放區中。
5. 識別提交 ID 和變更的檔案後，Lambda 函數會識別每個微服務目錄的管道，並啟動所需 CodePipeline 的管線。

工具

- AWS Cloud Development Kit (AWS CDK)是一個軟件開發框架，用於在代碼中定義雲基礎架構並通過 AWS CloudFormation.
- Python 是一種編程語言，可以讓您快速工作並更有效地整合系統。

Code

此模式的源代碼和模板可在 GitHub AWS CodeCommit monorepo 多管道觸發器存儲庫中找到。

最佳實務

- 此範例架構不包含已部署基礎結構的監視解決方案。如果您想要在生產環境中部署此解決方案，建議您啟用監視。如需詳細資訊，請參閱 AWS Serverless Application Model (AWS SAM) 文件中的使用應用程式深入解析監控無伺服器應用 CloudWatch 程式。
- 當您編輯此模式提供的範例程式碼時，請遵循 AWS CDK 文件中開發和部署雲端基礎結構的最佳做法。
- 定義微服務管道時，請檢閱 AWS CodePipeline 文件中的安全性最佳做法。
- 您也可以使用 cdk-na g 公用程式來檢查您的 AWS CDK 程式碼以取得最佳作法。此工具使用一組依套件分組的規則來評估您的程式碼。可用的包裝包括：
  - AWS 解決方案庫
  - Health 保險可攜性與責任法案 (HIPAA) 安全
  - 美國國家標準與技術研究院
  - 第五版

- 支付卡產業資料安全標準 (PCI DSS) 3.2.1

史诗

設定環境

任務	描述	所需技能
創建一個虛擬的 Python 環境。	在 AWS Cloud9 IDE 中，透過執行下列命令建立虛擬 Python 環境並安裝所需的相依性： `make install`	開發人員
引導 AWS 帳戶 和 AWS 區域的 AWS CDK.	通過運行以下命令引導所需的 AWS 帳戶 和區域： `make bootstrap account-id=<your-AWS-account-ID> region=<required-region>`	開發人員

為微服務新增管線

任務	描述	所需技能
將範例程式碼新增至應用程式目錄。	將包含範例應用程式程式碼的目`monorepo-sample` 錄新增至複製的 GitHub AWS CodeCommit monorepo 多管道觸發程序儲存庫中的目錄。	開發人員
編輯 `monorepo-main.json` 檔案。	將應用程序代碼的目錄名稱和管道的名稱添加到克隆存儲庫中的`monorepo-main.json` 文件中。	開發人員

任務	描述	所需技能
建立管線。	在存放庫的Pipelines 目錄中，為您的應用程式新增管線class。目錄包含兩個範例檔案，pipeline_hotsite.py 和pipeline_demo.py 。每個檔案都有三個階段：來源、建置和部署。 您可以複製其中一個檔案，並根據應用程式的需求對其進行變更。	開發人員

任務	描述	所需技能
編輯 monorepo_config.py 檔案。	在中service_map ，新增應用程式的目錄名稱以及您為管線建立的類別。 例如，下列程式碼顯示目錄中的管線定義，該Pipelines 目錄使用以MySamplePipeline 類別命名pipeline_mysample.py 的檔案： `...` `# Pipeline definition imports` `from pipelines.pipeline_demo import DemoPipeline` `from pipelines.pipeline_hotsite import HotsitePipeline` `from pipelines.pipeline_mysample import MySamplePipeline` `### Add your pipeline configuration here` `service_map: Dict[str, ServicePipeline] = {` `# folder-name -> pipeline-class` `'demo': DemoPipeline(),` `'hotsite': HotsitePipeline(),` `'mysample': MySamplePipeline()` `}`	開發人員

部署 MonoRepoStack 堆疊

任務	描述	所需技能
部署 AWS CloudFormation 堆疊。	執行`make deploy-core` 命令，在複製的存放庫的根目錄中以預設參數值部署 AWS CloudFormation MonoRepoStack 堆疊。 您可以透過執行`make deploy-core monorepo-name=<repo_name>` 指令來變更儲存庫的名稱。 附註：您可以使用`make deploy monorepo-name=<repo_name>` 指令同時部署這兩個管線。	開發人員
驗證存 CodeCommit 放庫。	透過執行`aws codecommit get-repository --repository-name <repo_name>` 命令驗證您的資源是否已建立。 重要提示：由於 AWS CloudFormation 堆棧創建了存儲 monorepo 的存儲 CodeCommit 庫，因此如果您已經開始將修改推送到其中，請不要運行該`cdk destroy MonoRepoStack` 命令。	開發人員
驗證 AWS CloudFormation 堆疊結果。	執行下列命令，驗證 AWS CloudFormation MonoRepoStack 堆疊是否已正確建立及設定：	開發人員

任務	描述	所需技能
	`aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query 'StackSummaries[?StackName == 'MonoRepoStack']'`	

部署 PipelinesStack 堆疊

任務	描述	所需技能
部署 AWS CloudFormation 堆疊。	您必須在部署 AWS CloudFormation `PipelinesStack` 堆疊之後部署`MonoRepoStack` 堆疊。當新的微服務添加到 monorepo 的代碼庫中時，堆棧的大小會增加，並在新的微服務上架時重新部署。 執行`make deploy-pipelines` 命令來部署 PipelinesStack 堆疊。 附註：您也可以執行`make deploy monorepo-name=<repo_name>` 命令，同時部署這兩個管道。 下列範例輸出顯示`PipelinesStacks` 部署如何在實作結束時列印微服務的 URL： `Outputs:`	開發人員

任務	描述	所需技能
	`PipelinesStack.dem ourl = .cloudfront.net PipelinesStack.hotsi teurl = .cloudfro nt.net`	
驗證 AWS CloudFormation 堆疊結果。	執行下列命令，驗證 AWS CloudFormation `Pipelines Stacks` 堆疊是否已正確建立及設定： `aws cloudformation list-stacks --stack-s tatus-filter CREATE_CO MPLETE UPDATE_COMPLETE --query 'StackSum maries[?StackName == 'PipelinesStack']'`	開發人員

清除資源

任務	描述	所需技能
刪除 AWS CloudFormation 堆疊。	執行 `make destroy` 命令。	開發人員
刪除管道的 S3 儲存貯體。	1. 登入AWS Management Console並開啟亞馬遜簡易儲存服務 (Amazon S3) 主控台。 2. 刪除與管道相關聯的 S3 儲存貯體，並使用下列名稱：`pipelinesstack-codepipeline*`	開發人員

故障診斷

問題	解決方案
我遇到了 AWS CDK 問題。	請參閱 AWS CDK 文件中的常見 AWS CDK 問題疑難排解。
我推送了我的微服務代碼，但微服務管道沒有運行。	設定驗證 驗證分支配置： • 確保您將代碼推送到正確的分支。此管線配置為僅在對分支進行變更時執`main`行。除非特別配置，否則推送到其他分支不會啟動管道。 • 推送程式碼之後，請檢查中是否顯示提交，AWS CodeCommit 以確定推送成功，以及本機環境與儲存庫之間的連線是否完整無損。如果推送程式碼發生問題，請重新整理您的認證 驗證組態檔案： • 確認中的`service_map` 變數`monorepo_config.py` 正確反映微服務的目前目錄結構。該變量在將代碼推送映射到相應的管道中起著至關重要的作用。 • 請確定`monorepo-main.json` 已更新，以包含微服務的新對應。此檔案對於管線識別並正確處理微服務的變更至關重要。 主控台上的疑難排解 AWS CodePipeline 檢查： • 在上 AWS Management Console，確認您位於管道託管的 AWS 區域 位置。開啟主CodePipeline 控台，並檢查是否已啟動與您的微服務對應的管線。

問題	解決方案
	錯誤分析：如果管道已啟動但失敗，請檢閱所提供的任何錯誤訊息或記錄檔， CodePipeline 以瞭解發生了什麼問題。 AWS Lambda 疑難排解： • 在AWS Lambda 主控台上，開啟 L `monorepo-event-handler` ambda 函數。驗證函數是否已啟動以回應程式碼推送。 日誌分析：檢查 Lambda 函數的日誌是否有任何問題。日誌可以提供有關函數運行時發生的情況的詳細見解，並幫助確定函數是否按預期處理事件。

問題	解決方案
我需要重新部署我所有的微服務。	有兩種方法可強制重新部署所有微服務。選擇符合您需求的選項。 方法 1：刪除參數存儲中的參數 此方法涉及刪除 Systems Manager 參數存放區中的特定參數，該參數會追蹤用於部署的上次提交 ID。當您移除此參數時，系統會強制在下一個觸發程序時重新部署所有微服務，因為它會將其視為全新狀態。 步驟： 1. 找出包含您 monorepo 的提交 ID 或相關部署標記的特定參數存放區項目。參數名稱的格式如下：`"/MonoRepoTrigger/{repository}/{branch_name}/LastCommit"` 2. 如果參數值很重要，或者您希望在重設之前保留部署狀態的記錄，請考慮備份參數值。 3. 使用 AWS Management Console AWS CLI、或 SDK 刪除已識別的參數。此動作會重設部署標記。 4. 刪除之後，下次推送至儲存庫時，應該會導致系統部署所有微服務，因為它會尋找最新的認可以考慮進行部署。 優點： • 以最少的步驟簡單快速實現。 • 不需要進行任意程式碼變更即可啟動部署。 缺點： • 對部署程序的精細控制較少。

問題	解決方案
	• 如果使用「參數存放區」管理其他重要組態，則可能有風險。 方法 2：在每個 monorepo 子文件夾中推送一個提交 此方法涉及進行一些小的更改，並將其推送到 monorepo 中的每個微服務子文件夾中以啟動其單獨的管道。 步驟： 1. 列出所有需要重新部署的 monorepo 中的微服務。 2. 對於每個微服務，請在其子資料夾中進行最小且不具影響力的變更。這可能是更新 README檔案、在設定檔中新增註解，或是任何不影響服務功能的變更。 3. 以清楚的訊息提交這些變更 (例如「啟動微服務的重新部署」)，並將其推送至儲存庫。確保將更改推送到啟動部署的分支。 4. 監視每個微服務的管道，以確認它們已成功啟動並完成。 優點： • 提供重新部署哪些微服務的精細控制。 • 更安全，因為它不涉及刪除可能用於其他目的的的配置參數。 缺點： • 更耗時，尤其是對於大量微服務而言。

問題	解決方案
	• 需要進行不必要的代碼更改，以使提交歷史混亂。

相關資源

- 使用 CDK Pipelines 的持續整合與交付 (CI/CD) (說明文件)AWS CDK
- aws-cd/管道模塊（API 參考）AWS CDK

使用 AWS 整合比特儲存庫與 AWS Amplify CloudFormation

創建者奧爾文亞伯拉罕（AWS）

環境：生產	技術： DevOps	AWS 服務：AWS Amplify；AWS CloudFormation

Summary

AWS Amplify 可協助您快速部署和測試靜態網站，而不必設定通常需要的基礎設施。如果您的組織想要使用 Bitbucket 進行原始檔控制，無論是移轉現有的應用程式程式碼或建置新的應用程式，您都可以部署此模式的方法。透過使 CloudFormation 用 AWS 自動設定 Amplify，您可以查看您使用的組態。

此模式說明如何使用 AWS 將 Bitbucket 儲存庫與 AWS CloudFormation Amplify 整合，以建立前端持續整合和持續部署 (CI/CD) 管道和部署環境。該模式的方法意味著您可以為可重複部署建立 Amplify 前端管道。

先決條件和限制

前提

- 有效的 Amazon Web Services (AWS) 帳戶
- 具有管理員訪問權限的活動 Bitbucket 帳戶
- 訪問使用 cURL 或郵差應用程序的終端
- 熟悉 Amplify
- 熟悉 AWS CloudFormation
- 熟悉 YAML 格式的文件

架構

技術, 堆

- Amplify

- AWS CloudFormation
- Bitbucket

工具

- AWS Amplify — Amplify 可協助開發人員開發和部署支援雲端的行動和 Web 應用程式。
- AWS CloudFormation — AWS CloudFormation 是一項可協助您建立 AWS 資源模型和設定 AWS 資源的服務，以減少管理這些資源的時間，將更多時間專注於在 AWS 中執行的應用程式。
- 比特桶-Bitbucket 是專為專業團隊設計的 Git 存儲庫管理解決方案。它為您提供了一個集中的位置來管理 Git 存儲庫，協作您的源代碼，並引導您完成開發流程。

Code

該`bitbucket-amplify.yml`文件（附件）包含此 CloudFormation 模式的 AWS 模板。

史诗

配置比特桶存儲庫

任務	描述	所需技能
（可選）創建一個比特桶存儲庫。	1. 登錄到您的 Bitbucket 帳戶並創建一個新的存儲庫。如需這方面的詳細資訊，請參閱 Bitbucket 文件中的建立 Git 儲存庫。 2. 記錄工作區的名稱。 注意：您也可以使用現有的 Bitbucket 儲存庫。	DevOps 工程師
開啟工作區設定。	1. 開啟工作區，然後選擇「存放庫」頁籤。 2. 選擇您要與擴大整合的存 Amplify 庫。	DevOps 工程師

任務	描述	所需技能
	3. 選擇存放庫名稱上方的工作區名稱。 4. 在側邊欄上，選擇「設定」。	
建立 OAuth 取用者。	1. 在 [應用程式和功能] 區段中，選擇 [OAuth 取用者]，然後選擇 [新增消費者]。 2. 輸入消費者的名稱，例如`Amplify Integration` 。 3. 輸入回呼網址。雖然此欄位是必要的輸入，但它不會用來完成整合，因此值可能是`http://localhost:3000` 4. 勾選 [這是私人消費者] 核取方塊。 5. 選擇下列權限： • 項目 — `Read` • 儲存庫 — `Admin` • 提取請求 — `Read` • 網絡掛鉤 — `Read` 和 `Write` 6. 保留所有其他欄位的預設選項，然後選擇「提交」。 7. 記錄產生的金鑰和密碼。	DevOps 工程師

任務	描述	所需技能
獲取 OAuth 訪問令牌。	1. 開啟終端機視窗並執行下列命令： `curl -X POST -u "KEY:SECRET" https://bitbucket.org/site/oauth2/access_token -d grant_type=client_credentials` 重要:將KEY之SECRET前記錄的金鑰和密碼取代為和。 2. 記錄訪問令牌，而不使用引號。令牌僅在有限的時間內有效，默認時間為兩小時。您必須在此時間 CloudFormation 範圍內執行 AWS 範本。	DevOps 工程師

建立和部署 AWS CloudFormation 堆疊

任務	描述	所需技能
下載 AWS CloudFormation 範本。	下載 `bitbucket-amplify.yml` AWS CloudFormation 範本 (隨附)。除了 Amplify 專案和分支之外，此範本還會在 Amplify 中建立 CI/CD 管線。	
建立和部署 AWS CloudFormation 堆疊。	1. 在您要部署的 AWS 區域中登入 AWS 管理主控台，然後開啟 AWS 主 CloudFormation 控台。	DevOps 工程師

任務	描述	所需技能
	2. 選擇 [建立堆疊 (含新資源)]，然後選擇 [上傳範本檔案]。 3. 上傳 `bitbucket-amplify.yml` 檔案。 4. 選擇 [下一步]，輸入堆疊名稱，然後輸入下列參數： • 訪問令牌：粘貼您之前創建的 OAuth 訪問令牌。 • 存儲庫 URL：添加比特桶項目存儲庫的 URL。URL 通常採用以下格式：`https://bitbucket.org/<WORKSPACE_NAME>/<REPO_NAME>` • 分支名稱：這必須與您的 Bitbucket 存儲庫中的分支名稱匹配。當您執行 AWS CloudFormation 堆疊時，此分支不需要存在，但在將程式碼部署到環境時需要這個分支。 • 專案名稱：這是要與「Amplify」專案相關聯的名稱。 5. 選擇下一步，然後選擇建立堆疊。	

測試 CI/CD 管線

任務	描述	所需技能
將程式碼部署到儲存庫中的分支。	1. 通過運行以下命令克隆您的 Bitbucket 存儲庫：`git clone https://bitbucket.org/<WORKSPACE_NAME>/<REPO_NAME>` 2. 查看執行 AWS CloudFormation 指令碼時使用的分支名稱。若要建立並簽出新分支，請執行`git checkout -b <BRANCH_NAME>` 指令。若要簽出現有分支，請執行指`git checkout <BRANCH_NAME>` 令 3. 將代碼提交到分支中，並通過運行和`git push`命令將其推送到`git commit`遠程分支。 4. 然後 Amplify 建置和部署應用程式。 如需這方面的詳細資訊，請參閱 Bitbucket 文件中的基本 Git 命令。	應用程式開發人員

相關資源

身份驗證方法（特拉西文檔）

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Step Functions 函數和 Lambda 代理函數在 AWS 帳戶之間啟動 CodeBuild 專案

由理查德·米爾納瓦特（AWS）和阿米特·安哈勒卡（AWS）創建

程式碼儲存庫：跨帳戶 CodeBuild 代理	環境：生產	技術： DevOps；管理與治理；營運；無伺服器
AWS 服務：AWS CodeBuild；AWS Lambda；AWS Step Functions；AWS X-Ray；AWS CloudFormation		

Summary

此模式示範如何使用 AWS 步驟函數和 AWS Lambda 代理函數，跨多個 AWS 帳戶非同步啟動 AWS CodeBuild 專案。您可以使用模式的示例 Step Functions 狀態機來測試 CodeBuild 項目的成功性。

CodeBuild 協助您從完全受管的執行階段環境使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 啟動操作任務。您可以透過覆寫環境變數，在執行階段變更 CodeBuild 專案的行為。此外，您還可以使用 CodeBuild 來管理工作流程。如需詳細資訊，請參閱 AWS 工作坊網站上的 Ser vice Catalog 工具，以及在 AWS 資料庫部落格 EventBridge上使用 AWS CodeBuild 和 Amazon 在 Amazon RDS for PostgreSQL 中安排任務。

先決條件和限制

先決條件

- 兩個作用中的 AWS 帳戶：使用 Step Functions 函數叫用 Lambda 代理函數的來源帳戶，以及用於建立遠端 CodeBuild 範例專案的目標帳戶

限制

- 此模式無法用於在帳戶之間複製人工因素。

架構

下圖顯示了此模式構建的體系結構。

該圖顯示以下工作流程：

1. Step Functions 數狀態機器會剖析提供的輸入對應，並針對您定義的每個帳戶、區域和專案叫用 Lambda Proxy 函數 (`codebuild-proxy-lambda`)。
2. Lambda 代理函數使用 AWS Security Token Service (AWS STS) 假設 IAM 代理角色 (`codebuild-proxy-role`)，該角色與目標帳戶中的 IAM 政策 (`codebuild-proxy-policy`) 相關聯。
3. Lambda 函數會使用假定的角色啟動 CodeBuild 專案並傳回 CodeBuild 工作 ID。Step Functions 狀態機器會迴圈並輪詢 CodeBuild 工作，直到收到成功或失敗狀態為止。

狀態機邏輯如下圖所示。

技術堆疊

- AWS CloudFormation
- CodeBuild
- IAM
- Lambda
- Step Functions
- X-Ray

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS CloudFormation 設計師提供整合的 JSON 和 YAML 編輯器，可協助您檢視和編輯 CloudFormation 範本。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。
- AWS X-Ray 可協助您收集應用程式所提供請求的相關資料，並提供工具供您檢視、篩選和深入瞭解該資料，以識別問題和優化機會。

Code

此模式的範例程式碼可在 GitHub跨帳戶 CodeBuild Proxy 儲存庫中取得。此模式使用適用於 Python 程式庫的 AWS Lambda Powertools 來提供記錄和追蹤功能。如需此程式庫及其公用程式的詳細資訊，請參閱適用於 AWS Lambda (Python) 的動力工具。

最佳實務

1. 調整「步驟功能」狀態機器中的等待時間值，以盡量減少工作狀態的輪詢請求。使用 CodeBuild 專案的預期執行時間。
2. 在「Step Functions」中調整地圖的`MaxConcurrency`屬性，以控制可以 parallel 執行的 CodeBuild 專案數目。
3. 如有需要，請檢閱生產準備就緒的範例程式碼。考慮解決方案可能會記錄哪些資料，以及預設 Amazon CloudWatch 加密是否足夠。

史诗

在來源帳戶中建立 Lambda 代理函數和相關聯的 IAM 角色

任務	描述	所需技能
記錄 AWS 帳戶 ID。	需要 AWS 帳戶 ID 才能設定跨帳戶的存取權限。 記錄來源和目標帳戶的 AWS 帳戶 ID。如需詳細資訊，請參	AWS DevOps

任務	描述	所需技能
	閱 IAM 文件中的尋找 AWS 帳戶 ID。	
下載 AWS CloudFormation 範本。	1. 從GitHub 存放庫下載此 CloudFormation 模式的 `sample_target_codebuild_template.yaml` AWS 範本。 2. 從GitHub 存放庫下載此 CloudFormation 模式的 `codebuild_lambda_proxy_template.yaml` AWS 範本。 注意：在 AWS CloudFormation 範本中，`<SourceAccountId>` 是來源帳戶的 AWS 帳戶 ID，`<TargetAccountId>` 也是目標帳戶的 AWS 帳戶 ID。	AWS DevOps

任務	描述	所需技能
建立和部署 AWS CloudFormation 堆疊。	1. 登入來源帳戶的 AWS 管理主控台，開啟 AWS CloudFormation 主控台，然後選擇「堆疊」。 2. 選擇 Create stack (建立堆疊)，然後選擇 With new resources (standard) (使用新資源 (標準))。 3. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。 4. 在 [上傳範本檔案] 中，選擇 [檔案]，然後選擇您下載的`codebuild_lambda_proxy_template.yaml` 檔案。選擇下一步。 5. 對於堆疊名稱，輸入堆疊的名稱 (例如，`codebuild-lambda-proxy`)。 6. 將`crossAccountTargetRoleArn` 參數取代為您的 <TargetAccountId> (例如，<arn:aws:iam::123456789012:role/proxy-lambda-codebuild-role>)。附註：您不需要更新**`targetCodeBuildProject`** 參數的預設值。 7. 選擇 [下一步]，接受預設堆疊建立選項，然後選擇 [下一步]。	AWS DevOps

任務	描述	所需技能
	8. 選擇 [我確認 AWS CloudFormation 可能會使用自訂名稱建立 IAM 資源] 核取方塊，然後選擇 [建立堆疊]。 注意：您必須先為代理 Lambda 函數建立 AWS CloudFormation 堆疊，才能在目標帳戶中建立任何資源。當您在目標帳戶中建立信任政策時，IAM 角色會從角色名稱轉換為內部識別碼。這就是 IAM 角色必須已經存在的原因。	
確認代理功能和狀態機的創建。	1. 等待 AWS CloudFormation 堆疊達到建立 _ 完成狀態。這應該需要不到一分鐘的時間。 2. 開啟 AWS Lambda 主控台，選擇「函數」，然後尋找`lambda-proxy-Proxy Lambda-<GUID>` 函數。 3. 開啟 AWS Step Functions 主控台，選擇狀態機器，然後尋找狀`sample-crossaccount-codebuild-state-machine` 態機器。	AWS DevOps

在目標帳戶中建立 IAM 角色並啟動範例 CodeBuild 專案

任務	描述	所需技能
建立和部署 AWS CloudFormation 堆疊。	1. 登入目標帳戶的 AWS 管理主控台，開啟 AWS CloudFormation 主控台，然後選擇「堆疊」。 2. 選擇 [建立堆疊]，然後選擇 [使用新資源 (標準)]。 3. 針對 Template source (範本來源)，選擇 Upload a template file (上傳範本檔案)。 4. 在 [上傳範本檔案] 中，選擇 [選擇檔案]，然後選擇`sample_target_codebuild_template.yaml` 檔案。選擇下一步。 5. 在堆疊名稱中，輸入堆疊的名稱 (例如:`sample-codebuild-stack`)。 6. 將`crossAccountSourceRoleArn` 參數取代為您的 <SourceAccountId> (例如，<arn:aws:iam::123456789012:role/codebuild-proxy-lambda-role>)。 7. 選擇 [下一步]，接受預設堆疊建立選項，然後選擇 [下一步]。 8. 選擇 [我確認 AWS CloudFormation 可能會使用自訂名稱建立 IAM 資源] 核	AWS DevOps

任務	描述	所需技能
	取方塊，然後選擇 [建立堆疊]。	
確認範例 CodeBuild 專案的建立。	1. 等待 AWS CloudFormation 堆疊達到建立 _ 完成狀態。這應該需要不到一分鐘的時間。 2. 開啟 AWS CodeBuild 主控台，然後尋找`sample-codebuild-project` 專案。	AWS DevOps

測試跨帳戶 Lambda 代理函數

任務	描述	所需技能
啟動狀態機。	1. 登入來源帳戶的 AWS 管理主控台，開啟 AWS Step Functions 主控台，然後選擇狀態機器。 2. 選擇狀`sample-crossaccount-codebuild-state-machine` 態機，然後選擇開始執行。 3. 在輸入編輯器中，輸入下列 JSON，並以`<TargetAccountID>` 包含 CodeBuild 專案的帳戶的 AWS 帳戶 ID 取代。 `{ "crossAccountTargetRoleArns": [ {`	AWS DevOps

任務	描述	所需技能
	<pre>"arn": "arn:aws:iam::<TargetAccountID>:role/proxy-lambda-codebuild-role", "region": "eu-west-1", "codeBuildProject": "sample-codebuild-project", "SampleValue1": "Value1", "SampleValue2": "Value2" }] }</pre> 附註：索引鍵值配對會作為環境變數從來源帳戶中的函數傳遞至目標帳戶中的 CodeBuild 專案。 4. 選擇 Start execution (開始執行)。 5. 在狀態機器頁面的 [詳細資料] 索引標籤上，檢查 [執行狀態] 是否設定為 [成功]。這會確認您的狀態機器正在執行。備註：狀態機器可能需要 30 秒左右的時間才能達到「成功」狀態。 6. 若要查看狀態機器中某個步驟的輸出和輸入，請在 [執行事件歷程記錄] 區段中展開該步驟。例如，展開 Lambda- CodeBuild 代理-開始步驟。輸出包含有關已	

任務	描述	所需技能
	覆寫環境變數、原始裝載和 CodeBuild 工作 ID 的詳細資訊。	
驗證環境變數。	1. 登入目標帳戶的 AWS 管理主控台。 2. 開啟 AWS 主 CodeBuild 控台，展開 [建置]，然後選擇 [建立專案]。 3. 選擇`sample-codebuild-project` 專案，然後選擇 [檢視詳細資料]。 4. 在 [建置歷程記錄] 索引標籤上，選擇專案的最新組建，然後選擇 [檢視記錄]。 5. 在 log 輸出中，確認列印到 STDOUT 的環境變數是否與 Step Functions 樣本狀態機器中的環境變數相符。	AWS DevOps

故障診斷

問題	解決方案
Step Functions 執行所花費的時間比預期更長。	在 Step Function 狀態機器中調整地圖的`MaxConcurrency` 屬性，以控制可以 parallel 執行的 CodeBuild 專案數目。
CodeBuild 工作的執行所花費的時間比預期更長。	1. 調整「Step Functions」狀態機器中的等待時間值，以盡量減少工作狀態的輪詢請求。使用 CodeBuild 專案的預期執行時間。

問題	解決方案
	2. 考慮 CodeBuild 是否適合使用的工具。例如，初始化任務所需的 CodeBuild 時間可能會比 AWS Lambda 長得多。如果需要高輸送量和快速完成時間，請考慮將商業邏輯遷移到 AWS Lambda 並使用散發架構。

使用 AWS 程式碼服務和 AWS KMS 多區域金鑰，管理對多個帳戶和區域的微型服務的藍/綠部署

創建者：巴拉吉維達吉 (AWS)、阿什庫馬爾 (AWS)、費薩爾沙赫達 (AWS)、阿南德·克里希納瓦拉納西 (AWS)、瓦尼莎·唐蒂雷迪 (AWS) 和維韋克唐穆圖 (AWS)

代碼庫：ecs-blue-green-global-代碼管deployment-with-multiregion-cmk道	環境：PoC 或試點	技術： DevOps；容器與微服務
AWS 服務：AWS CloudFormation; AWS CodeBuild; AWS CodeDeploy; AWS CodePipeline; Amazon ECS		

Summary

此模式說明如何根據藍/綠部署策略，將全球微服務應用程式從中央 AWS 帳戶部署到多個工作負載帳戶和區域。該模式支持以下內容：

- 軟體是在中央帳戶中開發的，而工作負載和應用程式則分散在多個帳戶和 AWS 區域。
- 單一 AWS 金鑰管理系統 (AWS KMS) 多區域金鑰可用於加密和解密，以涵蓋災難復原。
- KMS 金鑰是區域特定的，必須在三個不同的區域中維護或建立管線加工品。KMS 多區域金鑰有助於跨區域保留相同的金鑰 ID。
- Git 工作流程分支模型是使用兩個分支（開發和 main）實現的，並通過使用提取請求（PR）合併代碼。從此堆疊部署的 AWS Lambda 函數會建立從開發分支到主分支的 PR。將 PR 合併到主分支會啟動 AWS CodePipeline 管道，該管道協調持續整合和持續交付 (CI/CD) 流程，並跨帳戶部署堆疊。

此模式透過 AWS CloudFormation 堆疊提供範例基礎設施即程式碼 (IaC) 設定，以示範此使用案例。微型服務的藍/綠部署是使用 AWS 來實作。 CodeDeploy

先決條件和限制

先決條件

- 四個作用中的 AWS 帳戶：
  - 用於管理程式碼管道和維護 AWS CodeCommit 儲存庫的工具帳戶。
  - 三個工作負載 (測試) 帳戶用於部署微服務工作負載。
- 此樣式使用下列「區域」。如果您想要使用其他區域，則必須對 AWS CodeDeploy 和 AWS KMS 多區域堆疊進行適當的修改。
  - 工具 (AWS CodeCommit) 帳戶：`ap-south-1`
  - 工作負載 (測試) 帳戶 1：`ap-south-1`
  - 工作負載 (測試) 帳戶 2：`eu-central-1`
  - 工作負載 (測試) 帳戶 3：`us-east-1`
- 三個 Amazon Simple Storage Service (Amazon S3) 儲存貯體，適用於每個工作負載帳戶中的部署區域。（這些被稱為`S3BUCKETNAMETESTACCOUNT1`，`S3BUCKETNAMETESTACCOUNT2` `S3BUCKETNAMETESTACCOUNT3` 後來在這種模式中。）

  例如，您可以在特定帳戶和區域中建立這些值區，如下所示 (以隨機數字取代 xxxx)：

```
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-xxxx-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-xxxx-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-xxxx-us-east-1 --region us-east-1

#Example
##In Test Account 1
aws s3 mb s3://ecs-codepipeline-18903-ap-south-1 --region ap-south-1
##In Test Account 2
aws s3 mb s3://ecs-codepipeline-18903-eu-central-1 --region eu-central-1
##In Test Account 3
aws s3 mb s3://ecs-codepipeline-18903-us-east-1 --region us-east-1
```

限制

該模式使用 AWS CodeBuild 和其他組態檔來部署範例微服務。如果您有不同的工作負載類型 (例如，無伺服器)，則必須更新所有相關組態。

架構

目標技術堆疊

- AWS CloudFormation
- AWS CodeCommit
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

目標架構

自動化和規模

使用 AWS CloudFormation 堆疊範本 (IaC) 自動化設定。它可以輕鬆擴展到多個環境和帳戶。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 將部署 CodeDeploy自動化到亞馬遜彈性運算雲端 (Amazon EC2) 或現場部署執行個體、AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

- Git 是開放原始碼的分散式版本控制系統，可與 AWS CodeCommit 儲存庫搭配使用。
- Docker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中提供軟體。此模式使用 Docker 在本地構建和測試容器映像。
- cfn-lint 和 cfn-nag 是開源工具，可幫助您查看 CloudFormation 堆棧中的任何錯誤和安全問題。

代碼存儲庫

此模式的程式碼可在多個區域和帳戶儲存庫的「 GitHub 全域藍/綠」部署中使用。

史诗

設定環境變數

任務	描述	所需技能
匯出用於 CloudFormation 堆疊部署的環境變數。	定義環境變量，這些變量將在此模式稍後用作 CloudFormation 堆棧的輸入。 1. 更新您在三個帳戶與區域中建立的值區名稱，如先前 < 先決條件 > 一節所述： `export S3BUCKETNAMETESTACCOUNT1=<S3BUCKETACCOUNT1>` `export S3BUCKETNAMETESTACCOUNT2=<S3BUCKETACCOUNT2>` `export S3BUCKETNAMETESTACCOUNT3=<S3BUCKETACCOUNT3>` 2. 定義隨機字串以建立成品值區，因為值區名稱必須是全域唯一的：	AWS DevOps

任務	描述	所需技能
	`` export BUCKETSTA RTNAME=ecs-codepip eline-artifacts-19 992 ``` 3. 定義並匯出帳號 ID 和區域： ``` export TOOLSACCO UNT=<TOOLSACCOUNT> export CODECOMMI TACCOUNT=<CODECOMM ITACCOUNT> export CODECOMMI TREGION=ap-south-1 export CODECOMMI TREPONAME=Poc export TESTACCOU NT1=<TESTACCOUNT1> export TESTACCOU NT2=<TESTACCOUNT2> export TESTACCOU NT3=<TESTACCOUNT3> export TESTACCOU NT1REGION=ap-south -1 export TESTACCOU NT2REGION=eu-centr al-1 export TESTACCOU NT3REGION=us-east-1 export TOOLSACCO UNTREGION=ap-south -1 export ECRREPOSI TORYNAME=web ```	

Package 和部署基礎結構的 CloudFormation 堆疊

任務	描述	所需技能
複製儲存庫。	將範例儲存庫複製到工作位置的新儲存庫中： `##In work location` `git clone https://github.com/aws-samples/ecs-blue-green-global-deployment-with-multiregion-cmk-codepipeline.git`	AWS DevOps
Package 雲形資源。	在此步驟中，您可以封裝 CloudFormation 範本參考的本機成品，以建立服務所需的基礎設施資源，例如 Amazon Virtual Private Cloud (Amazon VPC) 和 Application Load Balancer。 範本位於程式碼儲存庫的`Infra`資料夾中。 `##In TestAccount1##` `aws cloudformation package \` `--template-file mainInfraStack.yaml \` `--s3-bucket $S3BUCKETNAMETESTACCOUNT1 \` `--s3-prefix infraStack \` `--region $TESTACCOUNT1REGION \` `--output-template-file infrastructure_`	AWS DevOps

任務	描述	所需技能
	`${TESTACCOUNT1}.template` `##In TestAccount2##` `aws cloudformation package \` `--template-file mainInfraStack.yaml \` `--s3-bucket $S3BUCKETNAMETESTACCOUNT2 \` `--s3-prefix infraStack \` `--region $TESTACCOUNT2REGION \` `--output-template-file infrastructure_${TESTACCOUNT2}.template` `##In TestAccount3##` `aws cloudformation package \` `--template-file mainInfraStack.yaml \` `--s3-bucket $S3BUCKETNAMETESTACCOUNT3 \` `--s3-prefix infraStack \` `--region $TESTACCOUNT3REGION \` `--output-template-file infrastructure_${TESTACCOUNT3}.template`	

任務	描述	所需技能
驗證封裝範本。	驗證封裝範本： `aws cloudformation validate-template \ --template-body file://infrastructure_${TESTACCOUNT1}.template` `aws cloudformation validate-template \ --template-body file://infrastructure_${TESTACCOUNT2}.template` `aws cloudformation validate-template \ --template-body file://infrastructure_${TESTACCOUNT3}.template`	AWS DevOps

任務	描述	所需技能
將套件檔案部署到工作負載帳戶中，	1. 根據您的設定，更新 infraParameters.json 指令碼中的預留位置值和帳戶名稱。 2. 將套件範本部署到三個工作負載帳戶。 `##In TestAccount1##` `aws cloudformation deploy \` `--template-file infrastructure_${TESTACCOUNT1}.template \` `--stack-name mainInfrastack \` `--parameter-overrides file://infraParameters.json \` `--region $TESTACCOUNT1REGION \` `--capabilities CAPABILITY_IAM CAPABILITY_NAMED_IAM` `##In TestAccount2##` `aws cloudformation deploy \` `--template-file infrastructure_${TESTACCOUNT2}.template \` `--stack-name mainInfrastack \` `--parameter-overrides file://infraParameters.json \` `--region $TESTACCOUNT2REGION \`	AWS DevOps

任務	描述	所需技能
	`` --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM ##In TestAccount3## aws cloudformation deploy \ --template-file infrastructure_${T ESTACCOUNT3}.templ ate \ --stack-name mainInfrastack \ --parameter- overrides file://in fraParameters.json \ --region $TESTACCO UNT3REGION \ --capabilities CAPABILITY_IAM CAPABILITY_NAMED_I AM ```	

推送樣本圖像並擴展 Amazon ECS

任務	描述	所需技能
將範例映像推送至 Amazon ECR 儲存庫。	將範例 (NGINX) 映像推送至亞馬遜彈性容器登錄 (Amazon ECR) 儲存庫，名為 web (如參數中所設定)。您可以根據需要自訂影像。 若要登入並設定將映像推送到 Amazon ECR 的登入資料，請遵循 Amazon ECR 文件中的指示。	AWS DevOps

任務	描述	所需技能
	這些命令是： `docker pull nginx` `docker images` `docker tag <imageid> aws_account_id.dkr.ecr.region.amazonaws.com/<web>:latest` `docker push <aws_account_id>.dkr.ecr.<region>.amazonaws.com/<web>:tag`	
擴展 Amazon ECS 並驗證存取權限。	1. 擴展 Amazon ECS 以建立兩個複本： `aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 2` 其中`Poc-Service` 指的是您的範例應用程式。 2. 使用瀏覽器的完整網域名稱 (FQDN) 或 DNS，或使用 curl 命令，確認服務可從應用 Application Load Balancer 器存取。	AWS DevOps

設定程式碼服務和資源

任務	描述	所需技能
在工具帳戶中創建一個 CodeCommit 存儲庫。	使用 CodeCommit 存放庫`code`資料夾中	AWS DevOps

任務	描述	所需技能
	的codecommit.yaml 範本，在工具帳戶中建立 GitHub 存放庫。您只能在計劃開發程式碼的單一區域中建立此儲存庫。 `aws cloudformation deploy --stack-name codecommitrepoStack --parameter-overrides CodeCommitReponame=$CODECOMMITREPONAME \ ToolsAccount=$TOOLSACCOUNT --template-file codecommit.yaml --region $TOOLSACCOUNTREGION \ --capabilities CAPABILITY_NAMED_IAM`	

任務	描述	所需技能
建立用於管理由產生的成品的 S3 儲存貯體 CodePipeline。	建立 S3 儲存貯體，以管理使 CodePipeline 用 GitHub 儲存庫code資料夾中的pre-reqs-bucket.yaml 範本所產生的成品。堆疊必須部署在所有三個工作負載 (測試) 和工具帳戶和區域中。 `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TESTACCOUNT1REGION --capabilities CAPABILITY_NAMED_IAM`  `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \`	AWS DevOps

任務	描述	所需技能
	TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=$BUCKETSTAR TNAME \ TestAccount1=$TE STACCOUNT1 TestAccou nt2=$TESTACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=$BUCKETSTAR TNAME \	

任務	描述	所需技能
	TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file prereqs_bucket.yaml --region $TOOLSACCOUNTREGION --capabilities CAPABILITY_NAMED_IAM	

任務	描述	所需技能
設定多區域 KMS 金鑰。	1. 使用 CodePipeline 將使用的主要金鑰和複本金鑰建立多區域 KMS 金鑰。在我們的例子中，`ToolsAccount1region - ap-south-1` 將是主要區域。 `aws cloudformation deploy --stack-name ecs-codepipeline-pre-reqs-KMS \ --template-file pre-reqs_KMS.yaml --parameter-overrides \ TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT --region $TOOLSACCOUNTREGION` 2. 設定要傳遞給專案的 CMKARN 變數。 CodeBuild 這些值在 ecs-codepipeline-pre-reqs-KMS 模板堆棧的輸出中可用（密鑰 ID 在所有區域中都相同，並開頭為`mrk-`）。或者，您可以從工具帳戶中獲取 CMKARN 值。在所有帳戶會話中導出它們：	AWS DevOps

任務	描述	所需技能
	`export CMKARN1=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx` `export CMKARN2=arn:aws:kms:eu-central-1:<TOOLSACCOUNTID>:key/mrk-xxx` `export CMKARN3=arn:aws:kms:us-east-1:<TOOLSACCOUNTID>:key/mrk-xxx` `export CMARNTOOLS=arn:aws:kms:ap-south-1:<TOOLSACCOUNTID>:key/mrk-xxx`	

任務	描述	所需技能
在工具帳戶中設置 CodeBuild 項目。	1. 使用 GitHub 儲存庫code資料夾中的codebuild_IAM.yaml 範本，在工具帳戶的單一區域中為 AWS 設定 AWS CodeBuild 的 AWS Identity and Access Management (IAM)： `#In ToolsAccount` `aws cloudformation deploy --stack-name ecs-codebuild-iam \` `--template-file codebuild_IAM.yaml --region $TOOLSACCOUNTREGION \` `--capabilities CAPABILITY_NAMED_IAM` 2. 使用codebuild.yaml 範本為您的建置 CodeBuild 專案進行設定。在所有三個區域中部署此範本，如下所示： `aws cloudformation deploy --stack-name ecscodebuildstack --parameter-overrides ToolsAccount=$TOOLSACCOUNT \` `CodeCommitRepoName=$CODECOMMITREPONAME ECRRepositoryName=$ECRREPOSITORYNAME APPACCOUNTID=$TESTACCOUNT1 \`	AWS DevOps

任務	描述	所需技能
	TestAccount3=$TE STACCOUNT3 CodeCommi tRegion=$CODECOMMI TREGION CMKARN=$C MKARN1 \ --template-file codebuild.yaml --region $TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=$TOOL SACCOUNT \ CodeCommitRepoName= $CODECOMMITREPONAME ECRRepositoryName= $ECRREPOSITORYNAME APPACCOUNTID=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tRegion=$CODECOMMI TREGION CMKARN=$C MKARN2 \ --template-file codebuild.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name ecscodebuildstack -- parameter-overrides ToolsAccount=$TOOL SACCOUNT \	

任務	描述	所需技能
	CodeCommitRepoName= $CODECOMMITREPONAME ECRRepositoryName= $ECRREPOSITORYNAME APPACCOUNTID=$TEST ACCOUNT3 \ CodeCommitRegion= $CODECOMMITREGION CMKARN=$CMKARN3 \ --template-file codebuild.yaml --region $TESTACCO UNT3REGION --capabil ities CAPABILIT Y_NAMED_IAM	

任務	描述	所需技能
CodeDeploy 在工作負載帳戶中設定。	使用 GitHub 儲存區域`code`資料夾中的`codedeploy.yaml` 樣板，CodeDeploy 在所有三個工作負載帳戶中進行設定。的輸出`mainInfraStack` 包括 Amazon ECS 叢集的 Amazon 資源名稱 (ARN) 和應用程式負載平衡器接聽程式。 附註：基礎結構堆疊中的值已經匯出，因此 CodeDeploy 堆疊範本會匯入這些值。 `##WorkloadAccount1##` `aws cloudformation deploy --stack-name ecscodedeploystack \` `--parameter-overrides ToolsAccount=$TOOLSACCOUNT mainInfrastackname=mainInfrastack \` `--template-file codedeploy.yaml --region $TESTACCOUNT1REGION --capabilities CAPABILITY_NAMED_IAM` `##WorkloadAccount2##` `aws cloudformation deploy --stack-name ecscodedeploystack \` `--parameter-overrides ToolsAccount=$TOOLSACCOUNT mainInfrastackname=mainInfrastack \`	AWS DevOps

任務	描述	所需技能
	`--template-file codedeploy.yaml --region $TESTACCOUNT2REGION --capabilities CAPABILITY_NAMED_IAM` `##WorkloadAccount3##` `aws cloudformation deploy --stack-name ecscodedeploystack \` `--parameter-overrides ToolsAccount=$TOOLSACCOUNT mainInfrastackname=mainInfrastack \` `--template-file codedeploy.yaml --region $TESTACCOUNT3REGION --capabilities CAPABILITY_NAMED_IAM`	

在工具帳戶 CodePipeline 中設置

任務	描述	所需技能
在工具帳戶中建立程式碼管線。	在工具帳戶中，執行以下命令： `aws cloudformation deploy --stack-name ecscodepipelinestack --parameter-overrides \` `TestAccount1=$TESTACCOUNT1 TestAccount1Region=$TESTACCOUNT1REGION \`	AWS DevOps

任務	描述	所需技能
	TestAccount2=$TE STACCOUNT2 TestAccou nt2Region=$TESTACC OUNT2REGION \ TestAccount3=$TE STACCOUNT3 TestAccou nt3Region=$TESTACC OUNT3REGION \ CMKARNTools=$CMK TROOLSARN CMKARN1= $CMKARN1 CMKARN2=$ CMKARN2 CMKARN3=$ CMKARN3 \ CodeCommitRepoName= $CODECOMMITREPONAME BucketStartName=$B UCKETSTARTNAME \ --template-file codepipeline.yaml -- capabilities CAPABILIT Y_NAMED_IAM	

任務	描述	所需技能
在 AWS KMS 金鑰政策 CodePipeline 和 S3 儲存貯體政策中提供存取權和 CodeBuild 角色。	1. 在 AWS KMS 金鑰政策中提供存取權 CodePipeline 和 CodeBuild 角色： `aws cloudformation deploy --stack-name ecs-codepipeline-pre-reqs-KMS \ --template-file pre-reqs_KMS.yaml --parameter-overrides \ CodeBuildCondition=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT --region $TOOLSACCOUNTREGION` 2. 更新 S3 儲存貯體政策以允許存取 CodePipeline 和 CodeDeploy 角色： `aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccount1=$TESTACCOUNT1`	AWS DevOps

任務	描述	所需技能
	TestAccount2=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT1REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter- overrides BucketSta rtName=$BUCKETSTAR TNAME \ PutS3BucketPolic y=true TestAccou nt1=$TESTACCOUNT1 TestAccount2=$TEST ACCOUNT2 \ TestAccount3=$TE STACCOUNT3 CodeCommi tAccount=$CODECOMM ITACCOUNT ToolsAcco unt=$TOOLSACCOUNT \ --template-file pre- reqs_bucket.yaml --region $TESTACCO UNT2REGION --capabil ities CAPABILIT Y_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts -bucket --parameter-	

任務	描述	所需技能
	overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TESTACCOUNT3REGION --capabilities CAPABILITY_NAMED_IAM aws cloudformation deploy --stack-name pre-reqs-artifacts-bucket --parameter-overrides BucketStartName=$BUCKETSTARTNAME \ PutS3BucketPolicy=true TestAccount1=$TESTACCOUNT1 TestAccount2=$TESTACCOUNT2 \ TestAccount3=$TESTACCOUNT3 CodeCommitAccount=$CODECOMMITACCOUNT ToolsAccount=$TOOLSACCOUNT \ --template-file pre-reqs_bucket.yaml --region $TOOLSACCOUNTREGION --capabil	

任務	描述	所需技能
	ities CAPABILIT Y_NAMED_IAM	

呼叫並測試管道

任務	描述	所需技能
將變更推送至 CodeCommit 儲存庫。	1. 如 AWS CodeCommit 文件所述，使codecommi trepoStack 用git clone命令複製在中建立的 CodeCommit 存放庫。 2. 使用必要的詳細資料更新輸入人工因素： • JSON 檔案：AccountID 在此檔案的三個位置更新檔案。重新命名三個檔案以包含帳號 ID。 • YAML 檔案：更新工作定義 ARN 和版本。重新命名三個檔案以包含帳號 ID。 3. 修改index.html 檔案以對首頁進行一些小的變更。 4. 將以下文件複製到存儲庫並提交： index.html Dockerfile buildspec.yaml appspec_<accountid >.yaml (3 files - one per account)	

任務	描述	所需技能
	taskdef<accountid>.json (3 files - one per account) 5. 啟動或重新啟動管線並驗證結果。 6. 使用 FQDN 或 DNS 從應用程式負載平衡器存取服務，並確認已部署更新。	

清除

任務	描述	所需技能
清理所有部署的資源。	1. 將 Amazon ECS 縮小到零執行個體： `aws ecs update-service --cluster QA-Cluster --service Poc-Service --desired-count 0` 2. 刪除每個帳號和區域中的 CloudFormation 堆疊： `##In Tools Account##` `aws cloudformation delete-stack --stack-name ecscodepipelinestack --region $TOOLSACCOUNTREGION` `aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT1REGION`	

任務	描述	所需技能
	aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT2REGION aws cloudformation delete-stack --stack-name ecscodebuildstack --region $TESTACCOUNT3REGION aws cloudformation delete-stack --stack-name ecs-codepipeline-pre-reqs-KMS --region $TOOLSACCOUNTREGION aws cloudformation delete-stack --stack-name codecommitrepoStack --region $TOOLSACCOUNTREGION aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TESTACCOUNT1REGION aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TESTACCOUNT2REGION aws cloudformation delete-stack --stack-name pre-reqs-artifacts-bucket --region $TESTACCOUNT3REGION aws cloudformation delete-stack --stack-name pre-reqs-	

任務	描述	所需技能
	artifacts-bucket --region $TOOLSACC OUNTREGION aws cloudformation delete-stack -- stack-name ecs-codeb uild-iam --region $TOOLSACCOUNTREGION ##NOTE: Artifact buckets will not get deleted if there are artifacts so it has to be emptied manually before deleting.##	
	##In Workload / Test Accounts## ##Account:1## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region $TESTACCOUNT1REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region $TESTACCOUNT1REGION ##Account:2## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region $TESTACCOUNT2REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region $TESTACCOUNT2REGION	

任務	描述	所需技能
	`` ##Account:3## aws cloudformation delete-stack -- stack-name ecscodede ploystack --region $TESTACCOUNT3REGION aws cloudformation delete-stack -- stack-name mainInfra stack --region $TESTACCOUNT3REGION ##NOTE: Amazon ECR (web) will not get deleted if the registry still includes images. It can be manually cleaned up if not required. ```	

故障診斷

問題	解決方案
您提交到存儲庫的更改不會被部署。	• 檢查 Docker 構建操作中的 CodeBuild 日誌是否存在錯誤。如需詳細資訊，請參閱CodeBuild 文件。 • 檢查部 CodeDeploy 署是否有任何 Amazon ECS 部署問題。

相關資源

- 推送碼頭圖像（Amazon ECR 文檔）
- Connect 到 AWS CodeCommit 儲存庫 (AWS CodeCommit 文件)
- AWS 疑難排解 CodeBuild (AWS CodeBuild 文件)

使用 AWS CloudFormation 和 AWS Config 監控 Amazon ECR 儲存庫是否有萬用字元許可

由維克蘭特特爾卡（AWS），薩吉德莫明（AWS）和瓦西姆·本哈拉姆（AWS）創建

環境：生產	技術： DevOps；容器與微服務	AWS 服務：AWS CloudFormation；AWS Config；Amazon ECR；Amazon SNS；AWS Lambda

Summary

在 Amazon Web Services (AWS) 雲端上，Amazon Elastic Container Registry (Amazon ECR) 是一種受管容器映像登錄服務，可使用 AWS Identity and Access Management (IAM) 支援具有以資源為基礎的許可的私有存放庫。

IAM 同時支援資源和動作屬性中的「*」萬用字元，這可讓您更輕鬆地自動選擇多個相符項目。在測試環境中，您可以在儲存庫政策聲明的主要元素中使用ecr:*萬用字元權限，允許所有經過驗證的 AWS 使用者存取 Amazon ECR 儲存庫。在無法存取生產資料的開發帳戶中開發和測試時，ecr:*萬用字元權限非常有用。

不過，您必須確定生產環境中未使用ecr:*萬用字元權限，因為這可能會造成嚴重的安全性弱點。此模式的方法可協助您識別儲存庫政策陳述式中包含ecr:*萬用字元權限的 Amazon ECR 儲存庫。 該 CloudFormation 模式提供了在 AWS Config 中建立自訂規則的步驟和 AWS 範本。然後，AWS Lambda 函數會監控您的 Amazon ECR 儲存庫政策陳述式是否有ecr:*萬用字元許可。如果發現不合規的儲存庫政策陳述式，Lambda 會通知 AWS Config 將事件傳送到 Amazon， EventBridge EventBridge 然後啟動亞馬遜簡單通知服務 (Amazon SNS) 主題。SNS 主題會透過電子郵件通知您有關不符合規範的儲存庫原則陳述式。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需有關這方面的詳細資訊，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI。

- 現有的 Amazon ECR 儲存庫，其中包含附加的政策聲明，可在您的測試環境中安裝和設定。如需這方面的詳細資訊，請參閱 Amazon ECR 文件中的建立私有存放庫和設定儲存庫政策聲明。
- AWS Config，在您偏好的 AWS 區域中設定。如需有關這方面的詳細資訊，請參閱 AWS Config 文件中的 AWS Config 入門。
- 該`aws-config-cloudformation.template`文件（附件），下載到您的本地計算機。

限制

- 此模式的解決方案是地區，您的資源必須在相同的區域中創建。

架構

下圖顯示 AWS Config 如何評估 Amazon ECR 儲存庫政策陳述式。

該圖顯示以下工作流程：

1. AWS Config 會啟動自訂規則。
2. 自訂規則會叫用 Lambda 函數來評估 Amazon ECR 儲存庫政策陳述式的合規性。然後 Lambda 函數會識別不相容的儲存庫原則陳述式。
3. Lambda 函數會將不合規狀態傳送至 AWS Config。
4. AWS Config 會將事件傳送到 EventBridge。
5. EventBridge 將不符合性通知發佈至 SNS 主題。
6. Amazon SNS 會傳送電子郵件警示給您或授權使用者。

自動化和規模

此模式的解決方案可監控任意數量的 Amazon ECR 儲存庫政策陳述式，但是您要評估的所有資源都必須在相同區域中建立。

工具

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們

一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。

- AWS 組態 — AWS Config 提供 AWS 帳戶中 AWS 資源組態的詳細檢視。這包含資源彼此之間的關係和之前的組態方式，所以您可以看到一段時間中組態和關係的變化。
- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種 AWS 受管容器映像登錄服務，安全、可擴展且可靠。Amazon ECR 支援私有儲存庫，其具有使用 IAM 的資源型許可。

- Amazon EventBridge — Amazon EventBridge 是一種無伺服器事件匯流排服務，您可以使用它將應用程式與來自各種來源的資料連接起來。 EventBridge 將來自應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務的即時資料串流傳遞至 AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點或其他帳戶中的事件匯流排等目標。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

此模式的代碼在`aws-config-cloudformation.template`文件中可用（附加）。

史诗

建立 AWS CloudFormation 堆疊

任務	描述	所需技能
建立 AWS CloudFormation 堆疊。	透過在 AWS CLI 中執行下列命令來建立 AWS CloudFormation 堆疊： `$ aws cloudformation create-stack --stack-name=AWSConfigECR \ --template-body file://aws-config-`	AWS DevOps

任務	描述	所需技能
	`cloudformation.template \ --parameters ParameterKey=<email>,ParameterValue=<myemail@example.com> \ --capabilities CAPABILITY_NAMED_IAM`	

測試 AWS Config 自訂規則

任務	描述	所需技能
測試 AWS Config 自訂規則。	1. 登入 AWS 管理主控台，開啟 AWS Config 主控台，然後選擇 [資源]。 2. 在 [資源清查] 頁面上，您可以依資源類別、資源類型和符合性狀態進行篩選。 3. 包含`ecr:*`是的 Amazon ECR 儲存庫`NON-COMPLIANT?` 和不包含的 Amazon ECR 儲存庫是。`ecr:* COMPLIANT` 4. 如果 Amazon ECR 儲存庫包含不合規的政策陳述式，訂閱 SNS 主題的電子郵件地址會收到通知。	AWS DevOps

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

從 AWS CodeCommit 事件執行自訂動作

創建者：阿卜杜拉希·奧拉耶 (AWS)

環境：PoC 或試點	技術: DevOps; 管理與治理	AWS 服務：AWS CodeCommit；Amazon SNS

Summary

當您使用 AWS CodeCommit 儲存庫存放程式碼時，您可能想要監控儲存庫，並在特定事件發生時啟動動作工作流程。例如，您可能想要在使用者在提交中對某行程式碼發表註解時傳送電子郵件通知，或啟動 AWS Lambda 函數，在提交後對儲存庫內容執行安全掃描。此模式概述了為自訂動作配置 CodeCommit 存放庫的步驟。該模式使用 AWS CodeCommit 通知規則擷取感興趣的事件，然後將這些事件傳送到設定的目標。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 熟悉 Git 命令。
- AWS CodeCommit，設定完成。如需指示，請參閱設定 AWS CodeCommit。
- (建議使用) 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需指示，請參閱 AWS CLI 入門。

架構

工具

AWS 服務

- AWS CodeCommit 是全受管的原始檔控制服務，可託管安全的 Git 儲存庫。它可讓團隊在安全且可高度擴展的生態系統中輕鬆協作程式碼。 CodeCommit 無需操作您自己的原始檔控制系統，也不必擔心擴充其基礎架構

- Amazon Simple Notification Service (Amazon SNS) 是一種 Web 服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。Amazon SNS 針對高輸送量、以推送為基礎的簡訊提供主題 (通訊管道)。 many-to-many 使用 Amazon SNS 主題，發佈者可以將訊息分發給大量訂閱者以進行 parallel 處理，包括 Amazon Simple Queue Service (Amazon SQS) 佇列、AWS Lambda 函數和 HTTP/S 網路掛鉤。您也可以使用 Amazon SNS 透過行動推送、簡訊和電子郵件傳送通知給最終使用者。

史诗

設定 CodeCommit 儲存庫

任務	描述	所需技能
創建一個 CodeCommit 存儲庫。	使用 CodeCommit 主控台或 AWS CLI 建立 CodeCommit 存放庫。如需指示，請參閱建立 CodeCommit 存放庫。	DevOps 工程師
將內容推送至 CodeCommit 儲存庫。	建立儲存庫之後，請使用 Git 命令將內容新增至其中。您可以從電腦移轉現有 Git 儲存庫的內容或本機、未建立版本控制的內容。如需指示，請參閱將檔案新增至儲存庫或遷移到 AWS CodeCommit。	DevOps 工程師

設定 Amazon SNS

任務	描述	所需技能
建立 SNS 主題。	此 SNS 主題會從中接收事件 CodeCommit。如需指示，請參閱建立 Amazon SNS 主題。	雲端架構師、 DevOps 工程師
建立自訂動作的資源。	對於要執行的自訂動作，您必須建立對應的資源。例如，如果您的自訂動作是執	雲端架構師、 DevOps 工程師

任務	描述	所需技能
	行 Lambda 程式碼並將訊息傳送至 SQS 佇列，您必須建立 Lambda 函數和 SQS 佇列。電子郵件和 SMS 通知等操作不需要資源。如需詳細資訊，請參閱 AWS 文件，了解您所建立的資源類型。	
訂閱 SNS 主題的自訂動作資源。	根據自訂動作，您可以建立適當通訊協定的訂閱。例如，您可以訂閱電子郵件通知的電子郵件地址、訂閱 Lambda 函數來執行自訂程式碼，或訂閱 SQS 佇列以傳送事件到 Amazon SQS。對於電子郵件和 SMS 之類的訂閱協議，您需要分別從發送到電子郵件或電話號碼的鏈接確認訂閱。如需指示，請參閱訂閱 Amazon SNS 主題。	雲端架構師、 DevOps 工程師

設定通知規則

任務	描述	所需技能
建立存 CodeCommit 放庫的通知規則。	建立通知規則時，請選取應起始通知的 Git 事件，選取 SNS 主題做為目標類型，然後選取您先前建立的 SNS 主題。您也可以為存放庫設定多個目標。如需指示，請參閱建立通知規則。	DevOps 工程師
測試自訂動作。	執行其中一個設定為起始通知的事件。例如，如果您選取該	DevOps 工程師

任務	描述	所需技能
	事件作為觸發程序，請建立提取要求。您應該會看到自訂動作正在執行中。例如，如果您訂閱 SNS 主題的電子郵件地址，您應該會收到電子郵件通知。	

相關資源

- AWS CodeCommit 文件
- Amazon SNS 文件
- Git 文件

將 Amazon CloudWatch 指標發佈到 CSV 檔案

創建者：阿卜杜拉希·奧拉耶 (AWS)

環境：PoC 或試點	技術： DevOps	AWS 服務：Amazon CloudWatch

Summary

此模式使用 Python 指令碼擷取 Amazon CloudWatch 指標，並將指標資訊轉換為逗號分隔值 (CSV) 檔案以提高可讀性。指令碼會採用應擷取其指標的 AWS 服務做為必要引數。您可以將 AWS 區域和 AWS 登入資料設定檔指定為選用引數。如果您未指定這些引數，指令碼會使用針對執行指令碼之工作站所設定的預設區域和設定檔。指令碼執行後，會在相同的目錄中產生並儲存 CSV 檔案。

有關此模式提供的腳本和相關文件，請參閱附件部分。

先決條件和限制

先決條件

- Python 3. x
- AWS 命令列界面 (AWS CLI)

限制

指令碼目前支援下列 AWS 服務：

- AWS Lambda
- Amazon Elastic Compute Cloud (Amazon EC2)
  - 根據預設，指令碼不會收集亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區指標。若要收集 Amazon EBS 指標，您必須修改附加的`metrics.yaml`檔案。
- Amazon Relational Database Service (Amazon RDS)
  - 但是，該腳本不支持 Amazon Aurora。
- Application Load Balancer
- Network Load Balancer

- Amazon API Gateway

工具

- Amazon CloudWatch 是專為 DevOps 工程師、開發人員、網站可靠性工程師 (SRE) 和 IT 經理所打造的監控服務。 CloudWatch 提供資料和可行的見解，協助您監控應用程式、回應整個系統的效能變更、最佳化資源使用率，以及取得營運狀態的統一檢視。 CloudWatch 以日誌、指標和事件的形式收集監控和操作資料，並提供在 AWS 和現場部署伺服器上執行的 AWS 資源、應用程式和服務的統一檢視。

史诗

安裝和設定必要條件

任務	描述	所需技能
安裝必要條件。	執行以下命令： `$ pip3 install -r requirements.txt`	開發人員
設定 AWS CLI。	執行以下命令： `$ aws configure`	開發人員

配置 Python 本

任務	描述	所需技能
開啟指令碼。	若要變更指令碼的預設組態，請開啟`metrics.yaml` 。	開發人員
設定指令碼的期間。	這是獲取的時間段。預設期間為 5 分鐘 (300 秒)。您可以變更時段，但請注意下列限制：	開發人員

任務	描述	所需技能
	• 如果您指定的小時數值介於 3 小時到 15 天之前，請使用 60 秒 (1 分鐘) 的倍數作為期間。 • 如果您指定的時數值介於 15 小時到 63 天之前，請使用 300 秒 (5 分鐘) 的倍數作為期間。 • 如果您指定的小時值大於 63 天之前，請使用 3,600 秒 (1 小時) 的倍數作為期間。 否則，API 操作將不會返回任何數據點。	
設定指令碼的時間。	此值指定您要擷取的量度小時數。預設值為 1 小時。若要擷取多天的量度，請提供以小時為單位的值。例如，對於 2 天，請指定 48。	開發人員
變更指令集的統計值。	(選擇性) 全域統計值為`Average`，擷取未指派特定統計值的測量結果時會使用此值。此指令集支援統計值`MaximumSampleCount` 、和`Sum`。	開發人員

運行 Python 本

任務	描述	所需技能
執行指令碼。	使用下列命令：	開發人員

任務	描述	所需技能
	`$ python3 cwreport.py <service>` 若要查看服務值以及選用性`region` 和`profile` 參數的清單，請執行下列命令： `$ python3 cwreport.py -h` 如需有關選用參數的詳細資訊，請參閱其他資訊一節。	

相關資源

- 設定 AWS CLI
- 使用 Amazon CloudWatch 指標
- Amazon CloudWatch 文檔
- EC2 CloudWatch 指標
- AWS Lambda 指標
- Amazon RDS 指標
- Application Load Balancer
- Network Load Balancer 度量
- Amazon API Gateway 指標

其他資訊

腳本用法

```
$ python3 cwreport.py -h
```

語法範例

```
python3 cwreport.py <service> <--region=Optional Region> <--profile=Optional credential
 profile>
```

參數

- 服務 (必要)-您要執行指令碼的服務。指令碼目前支援下列服務：AWS Lambda、Amazon EC2、Amazon RDS、Application Load Balancer、Network Load Balancer 和 API Gateway。
- 區域 (選用)-要從中擷取指標的 AWS 區域。預設「區域」為ap-southeast-1。
- 設定檔 (選用)-要使用的 AWS CLI 命名的設定檔。如果未指定此參數，則會使用預設設定的認證設定檔。

範例

- 若要使用預設區域ap-southeast-1和預設設定的登入資料來擷取 Amazon EC2 指標：$ python3 cwreport.py ec2
- 若要指定「區域」並擷取 API Gateway 指標：$ python3 cwreport.py apigateway --region us-east-1
- 若要指定 AWS 設定檔並擷取 Amazon EC2 指標：$ python3 cwreport.py ec2 --profile testprofile
- 若要同時指定區域和設定檔以擷取 Amazon EC2 指標：$ python3 cwreport.py ec2 --region us-east-1 --profile testprofile

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用最新的框架在 AWS Glue 中對 Python ETL 任務執行單元測試

代碼存儲庫：aw s-膠/作業單元測試	環境：生產	技術： DevOps; 大數據; 軟件開發和測試
AWS 服務：AWS CloudFormation; AWS; AWS CodeBuild; AWS CodeCommit; AWS CodePipeline; AWS AWS Glue		

Summary

您可以在本機開發環境中針對 AWS Glue 執行 Python 擷取、轉換和載入 (ETL) 任務的單元測試，但在 DevOps 管道中複寫這些測試可能很困難且耗時。當您在 AWS 技術堆疊上將大型主機 ETL 程序現代化時，單元測試特別具有挑戰性。此模式向您展示如何簡化單元測試，同時保持現有功能不變，避免在您發布新功能並維護高品質軟件時對關鍵應用程序功能造成干擾。您可以使用此模式中的步驟和程式碼範例，透過使用 AWS 中的 pytest 架構，在 AWS Glue 中針對 Python ETL 任務執行單元測試。CodePipeline您也可以使用此模式來測試和部署多個 AWS Glue 任務。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Glue 程式庫的 Amazon 彈性容器註冊表 (Amazon ECR) 映像 URI，可從亞馬遜 ECR 公共圖庫下載
- Bash 終端機 (在任何作業系統上)，包含目標 AWS 帳戶和 AWS 區域的設定檔
- Python 3.10 或更高版本
- 最火焰的
- 用於測試 AWS 服務的摩托 Python 程式庫

架構

技術堆疊

- Amazon Elastic Container Registry (Amazon ECR)
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- AWS Glue
- Pytest
- Python
- 適用於 AWS AWS Glue 的 Python ETL 程式庫

目標架構

下圖說明如何將以 Python 為基礎的 AWS Glue ETL 程序的單元測試整合到典型的企業級 AWS DevOps 管道中。

該圖顯示以下工作流程：

1. 在原始碼階段， CodePipeline 使用原始程式碼的 CodeCommit 儲存庫，包括範例 Python ETL 任務 (`sample.py`)、單元測試檔案 (`test_sample.py`) 和 AWS CloudFormation 範本。然後，將最新的代碼從主分支 CodePipeline 轉移到 CodeBuild 項目以進行進一步處理。
2. 在建置和發佈階段，先前來源階段的最新程式碼會透過 AWS Glue 公開 Amazon ECR 映像的協助進行單元測試。然後，測試報告會發行至 CodeBuild 報表群組。AWS Glue 程式庫的公用 Amazon ECR 儲存庫中的容器映像檔包含在 AWS Glue 本機執行和單元測試PySpark型 ETL 任務所需的所有二進位檔案。公用容器儲存庫有三個映像標籤，AWS Glue 支援的每個版本各一個。為了演示目的，此模式使用`glue_libs_4.0.0_image_01`圖像標籤。若要使用此容器映像檔做為中的執行階段影像 CodeBuild，請複製與您要使用之映像標籤對應的映像 URI，然後更新`TestBuild`資源 GitHub 儲存庫中的`pipeline.yml`檔案。
3. 在部署階段， CodeBuild 專案會啟動，如果所有測試通過，它會將程式碼發佈到 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體。
4. 使用者使用`deploy`資料夾中的 CloudFormation 範本部署 AWS Glue 任務。

工具

AWS 工具

- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Glue 是全受管的 ETL 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。

其他工具

- Python 是一種高級別的，解釋的通用編程語言。
- 摩托車是用於測試 AWS 服務的 Python 庫。
- Pytest 是一個用於編寫小單元測試的框架，可擴展以支持應用程序和庫的複雜功能測試。
- 適用於 AWS Glue 的 Python ETL 程式庫是用於本機開發 AWS Glue PySpark 批次任務的 Python 程式庫的儲存庫。

Code

此模式的代碼可在 GitHub aws-glu e-jobs-單元測試存儲庫中找到。存放庫包含下列資源：

- 資料夾中以 Python 為基礎的 AWS Glue 任務範例 `src`
- 文件夾中關聯的單元測試用例（使用 pytest 框架構建）`tests`
- 資`deploy`料夾中的 CloudFormation 範本 (以 YAML 撰寫)

最佳實務

CodePipeline 資源安全

最佳做法是為連接到中的管道的來源存放庫使用加密和驗證 CodePipeline。如需詳細資訊，請參閱 CodePipeline 文件中的安全性最佳做法。

監控和記錄資 CodePipeline 源

最佳做法是使用 AWS 記錄功能來判斷使用者在您的帳戶中執行的動作以及使用的資源。記錄檔會顯示下列內容：

- 動作的時間和日期
- 動作的來源 IP 位址
- 哪些動作因權限不足而失敗

AWS CloudTrail 和 Amazon CloudWatch 活動中提供了日誌記錄功能。您可以使 CloudTrail 用記錄 AWS 帳戶或代表 AWS 帳戶發出的 AWS API 呼叫和相關事件。如需詳細資訊，請參閱 CodePipeline 文件 CloudTrail中的使用 AWS 記錄 CodePipeline API 呼叫。

您可以使用 CloudWatch 事件監控 AWS 雲端資源和在 AWS 上執行的應用程式。您也可以在 CloudWatch 事件中建立警示。如需詳細資訊，請參閱文 CodePipeline 件中的監視事 CodePipeline 件。

史诗

部署原始程式碼

任務	描述	所需技能
準備要部署的程式碼歸檔。	1. `code.zip`從 GitHub aws-glue-jobs-單元測試存儲庫下載，或使用命令列工具自行建立 .zip 檔案。例如，您可以在終端機中執行下列命令，在 Linux 或 Mac 上建立 .zip 檔案： `git clone https://github.com/aws-samples/aws-glue-jobs-unit-testing.git` `cd aws-glue-jobs-unit-testing` `git checkout master` `zip -r code.zip src/ tests/ deploy/`	DevOps 工程師

任務	描述	所需技能
	2. 登入 AWS 管理主控台，然後選擇您選擇的 AWS 區域。 3. 建立 S3 儲存貯體，然後將 .zip 套code.zip件和檔案 (先前下載) 上傳到您建立的 S3 儲存貯體。	

任務	描述	所需技能
建立 CloudFormation 堆疊。	1. 登入 AWS 管理主控台，然後開啟主CloudFormation 控台。 2. 選擇 [建立堆疊]，然後選擇 [使用現有資源 (匯入資源)]。 3. 在「建立堆疊」頁面的「指定樣板」段落中，選擇上傳樣板檔案，然後選擇管線 .yml 樣板 (從儲存區域下載)。 GitHub 然後選擇下一步。 4. 針對堆疊名稱，輸入膠合單元測試管線，或選擇您選擇的堆疊名稱。 5. 對於「ApplicationStack 名稱」，請使用預先填入的膠合程式碼管線應用程式名稱。這是管線所建立的 CloudFormation 堆疊名稱。 6. 對於 BranchName，使用預先填入的主要名稱。這是在存放庫中建立的分支名稱，用於從 S3 CodeCommit 儲存貯體的 .zip 檔案簽入程式碼。 7. 對於 BucketName，請使用預先填入的 aws-膠水神器-us-east-1 儲存貯體名稱。這是包含 .zip 檔案的 S3 儲存貯體名稱，管道會用來存放程式碼成品。 8. 對於CodeZip檔案，請使用預先填入的 code.zip 值。這	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	是範例程式碼 S3 物件的金鑰名稱。物件應該是 .zip 檔案。 9. 對於 RepositoryName，請使用預先填入的 aws- 膠合單元測試名稱。這是由堆棧創建的 CodeCommit 存儲庫的名稱。 10對於 TestReportGroupName，請使用預先填入的膠水單位-報告名稱。這是為了存儲單元 CodeBuild 測試報告而創建的測試報告組的名稱。 11選擇 [下一步]，然後在 [設定堆疊選項] 頁面上再選擇 [下一步]。 12.在 [檢閱] 頁面的 [功能] 下，選擇 [我確認 CloudFormation 可能建立具有自訂名稱的 IAM 資源] 選項。 13選擇提交。堆疊建立完成後，您可以在 [資源] 索引標籤上看到已建立的資源。堆棧創建大約需要 5-7 分鐘。 堆疊會使用從 .zip 檔案簽入並上傳到 S3 CodeCommit 儲存貯體的初始程式碼自動建立儲存庫。此外，堆棧創建使用 CodeCommit 存儲庫作為源 CodePipeline 視圖。在上面的步驟中， CodeCommit 存儲庫	

任務	描述	所需技能
	是 aws-胶-单元测试，管道是 aws-胶合单元-测试管道。	
清理環境中的資源。	若要避免額外的基礎架構成本，請務必在試驗此模式中提供的範例之後刪除堆疊。 1. 開啟主CloudFormation 控台，然後選取您建立的堆疊。 2. 選擇刪除。這會刪除堆疊建立的所有資源，包括 CodeCommit 儲存庫、AWS Identity and Access Management (IAM) 角色或政策，以及 CodeBuild 專案。	AWS DevOps、 DevOps 工程師

運行單元測試

任務	描述	所需技能
在管道中運行單元測試。	1. 若要測試已部署的管道，請登入 AWS 管理主控台，然後開啟主CodePipeline 控台。 2. 選取 CloudFormation 堆疊建立的管線，然後選擇 [釋放變更]。管道開始運行（使用 CodeCommit 存儲庫中的最新代碼）。 3. TEST_AND_build 階段完成後，請選擇 [詳細資料] 索引標籤，然後檢查記錄檔。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	4. 選擇報告選項卡，然後從報告歷史記錄中選擇測試報告以查看單元測試結果。 5. 部署階段完成後，請在 AWS Glue 主控台上執行並監控已部署的 AWS Glue 任務。如需詳細資訊，請參閱 AWS Glue 文件中的監控 AWS AWS Glue。	

故障診斷

問題	解決方案
具有 Amazon S3、Amazon ECR 或 CodeCommit 來源的管道不再自動啟動	如果您變更使用 Amazon 中事件規則 EventBridge 或 CloudWatch 事件進行變更偵測的動作的任何組態設定，AWS 管理主控台可能無法偵測到來源識別碼相似且具有相同初始字元的變更。由於新的事件規則不是由控制台建立的，因此管線不會再自動啟動。 例如，將 CodeCommit 分支名稱從更改`MyTestBranch-1` 為`MyTestBranch-2` 是一個小改變。由於變更位於分支名稱的末尾，因此來源動作的事件規則可能不會更新或建立新來源設定的規則。 這適用於下列使用事件中 CloudWatch 事件進行變更偵測的來源動作： • 來源動作位於 Amazon S3 時，S3 儲存貯體名稱和 S3 物件金鑰參數或主控台識別碼 • 來源動作在 Amazon ECR 中時，存放庫名稱和映像標記參數或主控台識別碼

問題	解決方案
	• 來源動作位於中時的存放庫名稱和分支名稱參數或主控台識別碼 CodeCommit 若要解決此問題，請執行下列其中一個動作： • 變更 Amazon S3、Amazon ECR 或中的組態設定 CodeCommit，以便對參數值的起始部分進行變更。例如，將您的分支名稱從更改`release-branch` 為`2nd-release-branch` 。避免在名稱的末尾進行更改，例如`release-branch-2` 。 • 變更 Amazon S3、Amazon ECR 或 CodeCommit 每個管道的組態設定。例如，將您的分支名稱從更改`myRepo/myBranch` 為`myDeployRepo/myDeployBranch` 。避免在名稱的末尾進行更改，例如`myRepo/myBranch2` 。 • 不 CloudFormation 要使用 AWS 管理主控台，而是使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 或 AWS 建立和更新您的變更偵測事件規則。如需針對 Amazon S3 來源動作建立事件規則的指示，請參閱 Amazon S3 來源動作和 CloudWatch 事件。如需針對 Amazon ECR 動作建立事件規則的指示，請參閱 Amazon ECR 來源動作和 CloudWatch 事件。如需為動作建立事件規則的指示，請參閱CodeCommit 來源 CodeCommit 動作與 CloudWatch 事件。在主控台中編輯動作設定之後，請接受由主控台建立的更新變更偵測資源。

相關資源

- AWS Glue

- 在本機開發和測試 AWS Glue 任務
- CloudFormation 適用於 AWS AWS Glue 的 AWS

其他資訊

此外，您可以使用 AWS CLI 部署 AWS CloudFormation 範本。如需詳細資訊，請參閱 CloudFormation 文件中的使用轉換快速部署範本。

在 Amazon S3 中設置頭盔 v3 圖表存儲庫

環境：PoC 或試點	技術： DevOps；容器與微服務；現代化	工作負載：所有其他工作
AWS 服務：Amazon S3		

Summary

這種模式通過將 Helm v3 存儲庫集成到 Amazon Amazon Web Services 服務（AWS）雲上的亞馬遜 Simple Storage Service (Amazon S3)，可以幫助您有效地管理 Helm v3 圖表。若要使用此模式，您必須熟悉 Kubernetes 和掌舵，這是 Kubernetes 套件管理員。使用 Helm 存儲庫存儲圖表和管制圖版本可以改善中斷期間的平均還原時間（MTTR）。

此模式使用 AWS CodeCommit 進行 Helm 存放庫建立，並使用 S3 儲存貯體做為 Helm 圖表儲存庫，因此組織內的開發人員可以集中管理和存取圖表。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Python 版本 2.7.12 或更高版本
- pip
- 具有子網路和 Amazon 彈性運算雲端 (Amazon EC2) 執行個體的虛擬私有雲 (VPC)
- 安裝在 EC2 實例上的 Git
- 用於建立 S3 儲存貯體的 AWS Identity and Access Management (IAM) 存取權
- 從用戶端機器存取 Amazon S3 的 IAM (程式設計或角色)
- AWS CodeCommit 儲存庫
- AWS 命令列界面 (AWS CLI)

產品版本

- 頭盔第 3 版

- Python 版本 2.7.12 或更高版本

架構

目標技術堆疊

- Amazon S3
- AWS CodeCommit
- Helm
- 庫貝克特爾
- Python 和點子
- Git
- 幫助 -3 插件

目標架構

自動化和規模

- 您可以將Helm 整合到現有的持續整合/持續交付 (CI/CD) 自動化工具中，以自動化 Helm 圖表的封裝和版本控制 (此模式超出範圍)。
- GitVersion 或者 Jenkins 內建編號可用於自動化圖表的版本控制。

工具

- 掌舵 — Helm 是 Kubernetes 的套件管理員，可協助您在 Kubernetes 叢集上安裝及管理應用程式。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Helm-S3 插件 — 幫助 -3 插件支持與 Amazon S3 的交互。它可以與頭盔 v2 或頭盔 v3 一起使用。

史诗

安裝和驗證頭盔 v3

任務	描述	所需技能
安裝頭盔 v3 客戶端。	若要在本機系統上下載並安裝 Helm 用戶端，請執行下列命令：`sudo curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash`	雲端管理員、 DevOps 工程師
驗證頭盔的安裝。	若要驗證 Helm 用戶端，請執行下列命令：`helm version --short`	雲端管理員、 DevOps 工程師

將 S3 儲存貯體初始化為 Helm 儲存庫

任務	描述	所需技能
為舵圖表創建 S3 存儲桶。	建立唯一的 S3 儲存貯體。在值區中，建立名為的資料夾 `stable/myapp` 。此模式中的範例使用`s3://my-helm-charts/stable/myapp` 作為目標圖表儲存庫。	雲端管理員、 DevOps 工程師
安裝 Amazon S3 的幫助 -3 插件。	若要在用戶端電腦上安裝 helm-s3 外掛程式，請執行下列命令：`helm plugin install https://github.com/hypnoglow/helm-s3.git`	雲端管理員、 DevOps 工程師

任務	描述	所需技能
初始化 Amazon S3 掌舵存儲庫。	要將目標文件夾初始化為 Helm 存儲庫，請使用以下命令：`helm s3 init s3://my-helm-charts/stable/myapp` 命令會在目標中建立 `index.yaml` 檔案，以追蹤儲存在該位置的所有圖表資訊。	雲端管理員、 DevOps 工程師
驗證新創建的 Helm 存儲庫。	若要確認`index.yaml` 檔案是否已建立，請執行下列命令：`aws s3 ls s3://my-helm-charts/stable/myapp/`	雲端管理員、 DevOps 工程師
將 Amazon S3 存儲庫添加到客戶端機器上的掌舵。	若要將目標存放庫別名新增至 Helm 用戶端機器，請使用下列命令：`helm repo add stable-myapp s3://my-helm-charts/stable/myapp/`	雲端管理員、 DevOps 工程師

在 Amazon S3 掌舵存儲庫中打 Package 和發布圖表

任務	描述	所需技能
克隆你的頭盔圖表。	如果您的 CodeCommit 存儲庫中沒有本地 Helm 圖表，請運行以下命令從存儲 GitHub 庫克隆它們：`git clone <url_of_your_helm_source_code>.git`	雲端管理員、 DevOps 工程師

任務	描述	所需技能
打 Package 當地的頭盔圖表。	若要封裝您建立或複製的圖表，請使用下列命令：`helm package ./my-app` 例如，此模式使用圖`my-app`表。此命令會將`my-app`圖表資料夾的所有內容封裝到封存檔案中，該檔案會使用檔`Chart.yaml` 案中提到的版本號碼來命名。	雲端管理員、 DevOps 工程師
將本機套件存放在 Amazon S3 掌舵儲存庫中。	若要將本機套件上傳到 Amazon S3 中的 Helm 儲存庫，請執行下列命令：`helm s3 push ./my-app-0.1.0.tgz stable-myapp` 在命令中，`my-app`是您的圖表資料夾名稱，`0.1.0`是中提到的圖表版本`Chart.yaml` ，並且`stable-myapp` 是目標儲存庫別名。	雲端管理員、 DevOps 工程師
搜尋「頭盔」圖表。	若要確認圖表同時出現在本機和 Amazon S3 Helm 儲存庫中，請執行下列命令：`helm search repo stable-myapp`	雲端管理員、 DevOps 工程師

升級你的頭盔倉庫

任務	描述	所需技能
修改並封裝圖表。	在中values.yaml ，將replicaCount 值設定為1，然後封裝圖表，這次Chart.yaml 將中的版本變更為0.1.1。版本控制理想地通過自動化通過使用 CI/CD 管道中的工具 GitVersion 或 Jenkins 構建編號來實現。自動化版本號超出此模式的範圍。若要封裝圖表，請執行下列命令：helm package ./my-app/	雲端管理員、 DevOps 工程師
將新版本推送到 Amazon S3 中的頭盔存儲庫。	要推送新的軟件包，版本 0.1.1，到 Amazon S3 中的我的頭盔圖管理庫，運行以下命令：helm s3 push ./my-app-0.1.1.tgz stable-myapp	雲端管理員、 DevOps 工程師
驗證更新的頭盔圖表。	若要確認更新的圖表同時出現在本機和 Amazon S3 Helm 儲存庫中，請執行下列命令。 helm repo update helm search repo stable-myapp	雲端管理員、 DevOps 工程師

從 Amazon S3 掌舵存儲庫搜索並安裝圖表

任務	描述	所需技能
搜索我的應用程序圖表的所有版本。	若要檢視圖表的所有可用版本，請使用`--versions` 旗標執行下列命令：`helm search repo my-app --versions` 如果沒有旗標，Helm 預設會顯示圖表的最新上傳版本。	DevOps 工程師
從 Amazon S3 掌舵存儲庫安裝圖表。	自動安裝超出此模式的範圍，但您可以手動安裝。先前工作的搜尋結果會顯示`my-app`圖表的多個版本。若要從 Amazon S3 掌舵儲存庫安裝新版本 (0.1.1)，請使用下列命令：`helm upgrade --install my-app-release stable-myapp/my-app --version 0.1.1 --namespace dev`	DevOps 工程師

使用 Helm 回滾到以前的版本

任務	描述	所需技能
檢閱特定修訂的詳細資訊。	自動復原超出此病毒碼的範圍，但您可以手動復原至較早的版本。在您切換或復原至作業中版本，以及在安裝修訂版本之前進行額外的驗證層之前，請使用下列指令檢視哪些值已傳遞給每個修訂版：`helm`	DevOps 工程師

任務	描述	所需技能
	`get values --revisio n=2 my-app-release`	
回滾到以前的版本。	自動復原超出此模式的範圍。若要手動復原至先前的修訂版，請使用下列指令：`helm rollback my-app-re lease 1` 此範例將復原至修訂編號 1。	DevOps 工程師

相關資源

- 頭盔文件
- 幫助 -3 插件（麻省理工學院許可證）
- Amazon Simple Storage Service (Amazon S3)

使用 AWS CodePipeline 和 AWS CDK 設定 CI/CD 管道

程式碼儲存庫：AWS CodePipeline 搭配 CI/CD	環境：PoC 或試點	技術： DevOps
工作負載：開源	AWS 服務：AWS CodePipeline	

首頁

透過持續整合和持續交付 (CI/CD) 自動化您的軟體建置和發行程序，可支援可重複的建置，並快速交付新功能給使用者。您可以快速輕鬆地測試每個代碼更改，並且可以在發布軟件之前 catch 並修復錯誤。透過測試和發行程序執行每項變更，您可以驗證應用程式或基礎結構程式碼的品質。CI/CD 體現了一套文化、一組作業原則和實務集合，協助應用程式開發團隊更頻繁且可靠地提供程式碼變更。此實作也稱為 CI/CD 管線。

此模式定義了 Amazon Web Services (AWS) 上可重複使用的持續整合和持續交付 (CI/CD) 管道。AWS CodePipeline 管道是使用 AWS Cloud Development Kit (AWS CDK) v2 編寫的。

使用時 CodePipeline，您可以透過 AWS 管理主控台界面、AWS Command Line Interface (AWS CLI) (AWS CLI)、AWS 或 AWS 開發套件，為軟體發行程序的不同階段建模。 CloudFormation此模式示範使用 AWS CDK 實作 CodePipeline 及其元件。除了建構程式庫之外，AWS CDK 還包含一個工具組 (CLI 命令cdk)，這是與 AWS CDK 應用程式互動的主要工具。除其他功能外，此工具組還提供將一或多個堆疊轉換為 CloudFormation 範本，並將其部署到 AWS 帳戶的功能。

該管道包括用於驗證第三方庫安全性的測試，並有助於確保在指定的環境中快速自動發布。您可以通過驗證過程來提高應用程序的整體安全性。

此模式的目的是加速 CI/CD 管線的使用，以部署程式碼，同時確保您部署的資源遵循 DevOps 最佳實務。實作範例程式碼之後，您將擁有包 CodePipeline含 linting、測試、安全檢查、部署和部署後程序的 AWS。此模式還包括生成文件的步驟。開發人員可以使用 Makefile 在本機重現 CI/CD 步驟，並提高開發程序的速度。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 在下面的一個基本的了解：
  - AWS CDK
  - AWS CloudFormation
  - AWS CodePipeline
  - TypeScript

限制

此模式 TypeScript 僅會將 AWS CDK 用於使用。它不涵蓋 AWS CDK 支援的其他語言。

產品版本

使用下列工具的最新版本：

- AWS 命令列界面 (AWS CLI)
- CFN_NAG
- git 遠程代碼提交
- Node.js

架構

目標技術堆疊

- AWS CDK
- AWS CloudFormation
- AWS CodeCommit
- AWS CodePipeline

目標架構

管道是由 AWS CodeCommit 儲存庫 (`SampleRepository`) 中的變更觸發。在開始時，會 CodePipeline 建置構件、自行更新，並啟動部署程序。產生的管道會將解決方案部署到三個獨立的環境：

- Dev — 在使用中開發環境中進行三步程式碼檢查
- 測試 — 集成和回歸測試環境

- 產品 — 生產環境

包括在開發階段的三個步驟是短信，安全性和單元測試。這些步驟並行運行以加快該過程。為了確保管線僅提供工作中的加工品，只要處理程序中的某個步驟失敗，它就會停止執行。開發階段部署之後，管道會執行驗證測試以驗證結果。在成功的情況下，管道會接著將成品部署到測試環境，其中包含部署後驗證。最後一步是將成品部署到 Prod 環境。

下圖顯示從 CodeCommit 存放庫到執行的建置與更新程序的工作流程 CodePipeline、三個 Dev 環境步驟，以及三個環境中每個環境中的後續部署與驗證。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。在此模式中， CloudFormation 模板可用於創建 CodeCommit 存儲庫和 CodePipeline CI/CD 管道。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS CodePipeline 是 CI/CD 服務，可協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

其他工具

- cfn_nag 是一種開放原始碼工具，可在範本中尋找 CloudFormation 模式，以識別潛在的安全性問題。
- git 遠程代碼提交是一種通過擴展 Git 從 CodeCommit 存儲庫中推送和提取代碼的實用程序。
- Node.js 是一個事件驅動的 JavaScript 運行時環境，旨在構建可擴展的網絡應用程序。

Code

此模式的程式碼可在 CodePipeline 具有 CI/CD 實務儲存庫的 GitHub AWS 中取得。

最佳實務

檢閱 AWS Identity and Access Management (IAM) 政策等資源，以確認這些資源符合您的組織最佳實務。

史诗

安裝工具

任務	描述	所需技能
在 macOS 或 Linux 上安裝工具。	如果您使用的是 MacOS 或 Linux，則可以通過在首選終端機中運行以下命令或使用 Linux 版本的自製軟件來安裝工具。 `brew install` `brew install git-remote-codecommit` `brew install ruby brew-gem` `brew-gem install cfn-nag`	DevOps 工程師
使用 AWS Cloud9 安裝工具。	如果您使用的是 AWS Cloud9，請執行下列命令來安裝工具。 `gem install cfn-nag` 注意：AWS Cloud9 應該已安裝 Node.js 和 npm。若要檢查安裝或版本，請執行下列命令。 `node -v`	DevOps 工程師

任務	描述	所需技能
	`npm -v`	
設定 AWS CLI。	若要設定 AWS CLI，請使用適用於您作業系統的指示： • 視窗：透過 AWS CLI 登入資料協助程式，在 Windows 上透過 HTTPS 連線至 AWS CodeCommit 儲存庫的設定步驟 • Linux、macOS、Unix：透過 AWS CLI 認證協助程式，在 Linux、macOS 或 Unix 上建立 HTTPS 連線至 AWS CodeCommit 儲存庫的設定步驟	DevOps 工程師

設定初始部署

任務	描述	所需技能
下載或克隆代碼。	若要取得此模式所使用的程式碼，請執行下列其中一個動作： • 從軟件 GitHub 庫中的版本下載最新的源代碼，然後將下載的文件解壓縮到一個文件夾中。 • 通過運行以下命令克隆項目。 `git clone --depth 1 https://github.com`	DevOps 工程師

任務	描述	所需技能
	`/aws-samples/aws-codepipeline-cicd.git` 從複製的`.git`存放庫中移除目錄。 `cd ./aws-codepipeline-cicd` `rm -rf ./.git` 稍後，您將使用新建立的 AWS CodeCommit 儲存庫做為遠端原始伺服器。	
Connect 到 AWS 帳戶。	您可以使用臨時安全令牌或 landing zone 身份驗證進行連接。若要確認您使用的是正確的帳戶和 AWS 區域，請執行下列命令。 `AWS_REGION="eu-west-1"` `ACCOUNT_NUMBER=$(aws sts get-caller-identity --query Account --output text)` `echo "${ACCOUNT_NUMBER}"`	DevOps 工程師

任務	描述	所需技能
引導環境。	若要啟動 AWS CDK 環境，請執行下列命令。 `npm install` `npm run cdk bootstrap "aws://${ACCOUNT_NUMBER}/${AWS_REGION}"` 成功啟動環境後，應顯示以下輸出。 `# Bootstrapping environment aws://{account}/{region}...` `# Environment aws://{account}/{region} bootstrapped` 如需 AWS CDK 啟動安裝的詳細資訊，請參閱 AWS CDK 文件。	DevOps 工程師

任務	描述	所需技能
合成模板。	若要合成 AWS CDK 應用程式，請使用指cdk synth令。 `npm run cdk synth` 您應該會看到下列輸出。 `Successfully synthesized to <path-to-directory>/aws-codepipeline-cicd/cdk.out` `Supply a stack id (CodePipeline, Dev-MainStack) to display its template.`	DevOps 工程師

任務	描述	所需技能
部署 CodePipeline 堆疊。	現在您已啟動並合成 CloudFormation 範本，您可以進行部署。部署將建立 CodePipeline 管線和 CodeCommit 儲存庫，這將成為管線的來源和觸發程序。 `npm run cdk -- deploy CodePipeline --require-approval never` 執行命令之後，您應該會看到成功部署 CodePipeline 堆疊和輸出資訊。會`CodePipeline.RepositoryName` 為您提供 AWS 帳戶中 CodeCommit 儲存庫的名稱。 `CodePipeline: deploying...` `CodePipeline: creating CloudFormation changeset...` `# CodePipeline` `Outputs:` `CodePipeline.RepositoryName = SampleRepository` `Stack ARN:` `arn:aws:cloudformation:REGION:ACCOUNT-ID:stack/CodePipeline/STACK-ID`	DevOps 工程師

任務	描述	所需技能
設置遠程 CodeCommit 存儲庫和分支。	成功部署後， CodePipeline 將啟動管道的第一次執行，您可以在 AWS CodePipeline 主控台中找到該管道。由於 AWS CDK 且 CodeCommit 未啟動預設分支，因此此初始管道執行將會失敗，並傳回下列錯誤訊息。 `The action failed because no branch named main was found in the selected AWS CodeCommit repository SampleRepository. Make sure you are using the correct branch name, and then try again. Error: null` 要修復此錯誤，請將遠程原點設置為`SampleRepository` ，然後創建所需的`main`分支。 `RepoName=$(aws cloudformation describe-stacks --stack-name CodePipeline --query "Stacks[0].Outputs[?OutputKey=='RepositoryName'].OutputValue" --output text)` `echo "${RepoName}"` `#` `git init` `git branch -m master main`	DevOps 工程師

任務	描述	所需技能
	`git remote add origin codecommit://${RepoName}` `git add .` `git commit -m "Initial commit"` `git push -u origin main`	

測試已部署的 CodePipeline 管線

任務	描述	所需技能
提交變更以啟動管線。	成功的初始部署之後，您應該擁有一個完整的 CI/CD 管線，並將分`main`支`SampleRepository` 做為來源分支。一旦您將變更提交至`main`分支，管線就會啟動並執行下列動作順序： 1. 從 CodeCommit 存儲庫中獲取代碼。 2. 建置您的程式碼。 3. 更新管道本身（`UpdatePipeline` ）。 4. 運行三個並行作業進行 linting，安全性和單元測試檢查。 5. 在成功的情況下，管道會將`Main`堆疊部署`./lib/main-stack.ts` 至 Dev 環境。 6. 針對已部署的資源執行部署後檢查。您可以在	DevOps 工程師

任務	描述	所需技能
	CodePipeline 控制台中按照所有 CodePipeline 步驟和結果進行操作。 7. 在成功的情況下，管道將重複測試和 Prod 環境的部署和驗證。	

使用生成檔在本機進行測試

任務	描述	所需技能
使用生成文件運行開發過程。	您可以使用`make`命令在本機執行整個管線，也可以執行個別步驟 (例如`make linting`)。 若要測試使用`make`，請執行下列動作： • 實施本地管道：`make` • 僅運行單元測試：`make unittest` • 部署到當前帳戶：`make deploy` • 清理環境：`make clean`	應用 DevOps 程式開發人員、

清除資源

任務	描述	所需技能
刪除 AWS CDK 應用程式資源。	若要清理 AWS CDK 應用程式，請執行下列命令。 `cdk destroy --all`	DevOps 工程師

任務	描述	所需技能
	請注意，在啟動載入期間建立的 Amazon 簡易儲存服務 (Amazon S3) 儲存貯體不會自動刪除。他們需要允許刪除的保留政策，或者您需要在 AWS 帳戶中手動刪除它們。	

故障診斷

問題	解決方案
範本未如預期般運作。	如果出現問題並且模板不起作用，請確保您具有以下內容： • 工具的正確版本。 • 存取目標 AWS 帳戶 (網路連線)。 • 目標 AWS 帳戶擁有足夠的許可。

相關資源

- 開始使用 IAM 身分中心的常見任務
- AWS CodePipeline 文件集
- AWS CDK

使用憑證管理員和讓我們 end-to-end 加密為 Amazon EKS 上的應用程式設定加密

由馬亨德拉·西達帕（AWS）和瓦森斯傑亞拉伊（AWS）創建

代碼存儲庫：Amazon E nd-to-end KS 上的 E 加密	環境：PoC 或試點	技術： DevOps；容器與微服務；安全性、身分識別、合規性
工作負載：所有其他工作	AWS 服務：Amazon EKS；Amazon Route 53	

Summary

實作 end-to-end 加密可能很複雜，而且您需要管理微服務架構中每個資產的憑證。雖然您可以使用 Network Load Balancer 或 Amazon Amazon API Gateway 在 Amazon Web Services (AWS) 網路邊緣終止傳輸層安全性 (TLS) 連線，但有些組織需要 end-to-end 加密。

此模式使用 NGINX 入口控制器進行入口。這是因為當您建立 Kubernetes 輸入時，輸入資源會使用 Network Load Balancer。Network Load Balancer 不允許上傳用戶端憑證。因此，您無法透過 Kubernetes 輸入來達成相互 TLS。

此模式適用於需要在其應用程式中所有微服務之間進行相互驗證的組織。相互 TLS 減少了維護使用者名稱或密碼的負擔，也可以使用統包安全性架構。如果您的組織有大量連接的設備或必須遵守嚴格的安全性準則，則此模式的方法是兼容的。

此模式透過在 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 上執行的應用程式實施 end-to-end 加密，有助於提高組織的安全狀態。此模式在 Amazon EKS 儲存庫的 GitHub E nd-to-end 加密中提供範例應用程式和程式碼，以顯示微服務如何在 Amazon EKS 上以 end-to-end 加密方式執行。該模式的方法使用證書管理器，這是 Kubernetes 的附加組件，並將「讓我們加密」作為證書頒發機構（CA）。Let's Encrypt 是一種經濟高效的解決方案，用於管理證書並提供有效期為 90 天的免費證書。在 Amazon EKS 上部署新的微服務時，CERT-Manager 會自動執行憑證的隨選佈建和輪換。

目標受眾

對於具有 Kubernetes、TLS、Amazon Route 53 和網域名稱系統 (DNS) 相關經驗的使用者，建議使用此模式。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有 Amazon EKS 叢集。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 1.7 版或更新版本，可在 macOS、Linux 或視窗上安裝和設定。
- 命`kubectl`令列公用程式已安裝並設定為存取 Amazon EKS 叢集。如需這方面的詳細資訊，請參閱 Amazon EKS 文件中的安裝 kubectl。
- 用來測試應用程式的現有 DNS 名稱。如需這方面的詳細資訊，請參閱 Amazon Route 53 說明文件中的使用 Amazon Route 53 註冊網域名稱。
- 最新的 Helm 版本，安裝在您的本地計算機上。如需有關這方面的詳細資訊，請參閱 Amazon EKS 文件中的搭配使用 Hel m 和 GitHub 掌舵儲存庫。
- Amazon EKS 儲存庫上的 GitHub E nd-to-end 加密已複製到您的本機電腦。
- 從 Amazon EKS 儲存庫上複製的 GitHub E nd-to-end 加密取代`policy.json`和`trustpolicy.json`檔案中的下列值：
  - `<account number>`— 替換為您要在其中部署解決方案的帳戶的 AWS 帳戶 ID。
  - `<zone id>`— 替換為域名的 Route 53 區域 ID.
  - `<node_group_role>`— 取代為與 Amazon EKS 節點關聯的 AWS Identity and Access Management (IAM) 角色的名稱。
  - `<namespace>`— 取代為您部署 NGINX 入口控制器和範例應用程式的 Kubernetes 命名空間。
  - `<application-domain-name>`— 替換為從 Route 53 的 DNS 域名.

限制

- 此模式不會說明如何輪換憑證，而只示範如何在 Amazon EKS 上搭配微型服務使用憑證。

架構

下圖顯示此模式的工作流程和架構元件。

該圖顯示以下工作流程：

1. 客戶端發送訪問應用程序到 DNS 名稱的請求。
2. 路 Route 53 記錄是 Network Load Balancer 的 CNAME。
3. Network Load Balancer 會將要求轉送至使用 TLS 接聽程式設定的 NGINX 輸入控制器。NGINX 入口控制器和 Network Load Balancer 之間的通訊遵循 HTTPS 通訊協定。
4. NGINX Ingress 控制器會根據用戶端對應用程式服務的要求執行路由。
5. 應用程式服務會將要求轉送至應用程式網繭。該應用程序旨在通過調用密碼來使用相同的證書。
6. 網繭會使用憑證管理員憑證執行範例應用程式。NGINX 入口控制器與網繭之間的通訊使用 HTTPS。

注意：CERT-管理員會在自己的命名空間中執行。它使用 Kubernetes 叢集角色，在特定命名空間中將憑證佈建為密碼。您可以將這些命名空間附加到應用程式網繭和 NGINX 入口控制器。

工具

AWS 服務

- Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，您可以使用它在 AWS 上執行 Kubernetes，而無需安裝、操作和維護自己的 Kubernetes 控制平面或節點。
- E@@ lastic Load Balancing 會自動將傳入流量分配到多個目標、容器和 IP 位址。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

其他工具

- 憑證管理員是 Kubernetes 的附加元件，可要求憑證、將憑證散佈至 Kubernetes 容器，並自動執行憑證續約。
- NGINX 入口控制器是適用於 Kubernetes 和容器化環境中雲端原生應用程式的流量管理解決方案。

史诗

使用 Route 53 建立和設定公用託管區域

任務	描述	所需技能
在 Route 53 中創建一個公共託管區域。	登入 AWS 管理主控台，開啟 Amazon Route 53 主控台，選擇託管區域，然後選擇 [建立託管區域]。建立公用託管區域並記錄區域 ID。如需這方面的詳細資訊，請參閱 Amazon Route 53 說明文件中的建立公用託管區域。 注意：ACME DNS01 會使用 DNS 提供者來張貼認證管理員發出憑證的挑戰。這項挑戰要求您透過將特定值放在該網域名稱下的 TXT 記錄中，以證明您可以控制網域名稱的 DNS。在 Let's Encrypt 為您的 ACME 客戶端提供一個令牌後，您的客戶端會創建一個從該令牌和您的帳戶密鑰派生的 TXT 記錄，並將該記錄放在`_acme-challenge.<YOURDOMAIN>` 。然後讓我們加密查詢該記錄的 DNS。如果找到相符項目，您可以繼續發行憑證。	AWS DevOps

設定 IAM 角色以允許憑證管理員存取公有託管區域

任務	描述	所需技能
建立認證管理員的 IAM 政策。	需要 IAM 政策，才能向憑證管理員提供驗證您是否擁有 Route 53 網域的權限。`policy.json` 範例 IAM 政策是在 Amazon EKS 儲存庫上複製的 GitHub E nd-to-end 加密中的`1-IAMRole` 目錄中提供的。 在 AWS CLI 中輸入以下命令以建立 IAM 政策。 `aws iam create-policy \ --policy-name PolicyForCertManager \ --policy-document file://policy.json`	AWS DevOps
為憑證管理員建立 IAM 角色。	建立 IAM 政策之後，您必須建立 IAM 角色。`1-IAMRole` 目錄中提供了 IAM 角色`trustpolicy.json` 範例。 在 AWS CLI 中輸入以下命令以建立 IAM 角色。 `aws iam create-role \ --role-name RoleForCertManager \ --assume-role-policy-document file://trustpolicy.json`	AWS DevOps

任務	描述	所需技能
將政策連接到角色。	在 AWS CLI 中輸入以下命令，將 IAM 政策附加到 IAM 角色。`AWS_ACCOUNT_ID` 以 AWS 帳戶的 ID 取代。 `aws iam attach-role-policy \ --policy-arn arn:aws:iam::AWS_ACCOUNT_ID:policy/PolicyForCertManager \ --role-name RoleForCertManager`	AWS DevOps

在 Amazon EKS 中設置 NGINX 入口控制器

任務	描述	所需技能
部署 NGINX 入口控制器。	安裝使`nginx-ingress` 用 Helm 的最新版本。在部署之前，您可以根據自己的需求修改`nginx-ingress` 配置。此模式使用帶註釋的面向內部的 Network Load Balancer，該模式可在目錄中使用。`5-Nginx-Ingress-Controller` 從目錄執行下列 Helm 指令，以安裝 NGINX 入口控制器。`5-Nginx-Ingress-Controller` `helm install test-nginx nginx-stable/`	AWS DevOps

任務	描述	所需技能
	`nginx-ingress -f 5-Nginx-Ingress-Controller/values_internal_nlb.yaml`	
確認已安裝 NGINX 入口控制器。	輸入 `helm list` 命令。輸出應顯示 NGINX 入口控制器已安裝。	AWS DevOps

任務	描述	所需技能
建立 Route 53 A 記錄。	A 記錄指向 NGINX 入口控制器建立的 Network Load Balancer。 1. 取得 Network Load Balancer 的 DNS 名稱。如需指示，請參閱取得 ELB 負載平衡器的 DNS 名稱。 2. 在 Amazon Route 53 主控台上，選擇託管區域。 3. 選取您要在其中建立記錄的公用託管區域，然後選擇 [建立記錄]。 4. 輸入記錄的名稱。 5. 在 [記錄類型] 中，選擇 [A]- 將流量路由到 IPv4 和部分 AWS 資源。 6. 啟用別名。 7. 在將流量路由至中，執行下列操作： a. 選擇 Network Load Balancer 的別名。 b. 選擇部署 Network Load Balancer 的 AWS 區域。 c. 輸入 Network Load Balancer 的 DNS 名稱。 8. 選擇建立記錄。	AWS DevOps

VirtualServer 在 Amazon EKS 上設置 NGINX

任務	描述	所需技能
部署 NGINX VirtualServer。	NGINX VirtualServer 資源是負載平衡配置，是輸入資源的替代方案。建立 NGINX VirtualServer 資源的組態位於目錄中的`nginx_virtualserver.yaml` 檔案中`6-Nginx-Virtual-Server` 。在中輸入下列指令`kubectl`以建立 NGINX VirtualServer 資源。 `kubectl apply -f nginx_virtualserver.yaml` 重要：請確定您已更新`nginx_virtualserver.yaml` 檔案中的應用程式網域名稱、憑證密碼和應用程式服務名稱。	AWS DevOps
驗證 VirtualServer 是否已建立 NGINX。	在中輸入下列指令，`kubectl`以確認 NGINX VirtualServer 資源是否已成功建立。 `kubectl get virtualserver` 注意：請確認`Host`資料欄與應用程式的網域名稱相符。	AWS DevOps
在啟用 TLS 的情況下部署 NGINX 網頁伺服器。	此病毒碼使用已啟用 TLS 的 NGINX 網頁伺服器作為測	AWS DevOps

任務	描述	所需技能
	試 end-to-end 加密的應用程式。部署測試應用程式所需的組態檔案位於目demo-webserver 錄中。 在中輸入下列指令kubectl以部署測試應用程式。 `kubectl apply -f nginx-tls-ap.yaml`	
確認已建立測試應用程式資源。	在中輸入下列指令，kubectl以確認是否已為測試應用程式建立必要的資源： • `kubectl get deployments` 備註：驗證Ready欄與Available 欄。 • `kubectl get pods | grep -i example-deploy` 注意：網繭應running處於狀態。 • `kubectl get configmap` • `kubectl get svc`	AWS DevOps

任務	描述	所需技能
驗證應用程式。	1. 輸入下列命令，方法是以<application-domain-name> 您先前建立的 Route53 DNS 名稱取代。 `curl --verbose https://<application-domain-name>` 2. 確認您可以存取應用程式。	AWS DevOps

相關資源

AWS 資源

- 使用 Amazon Route 53 控制台創建記錄（Amazon Route 53 文檔）
- 在 Amazon EKS 上搭配 NGINX 輸入控制器使用 Network Load Balancer (AWS 部落格文章)

其他資源

- Route 53（證書管理器文檔）
- 設定 DNS01 挑戰提供者 (憑證管理員文件)
- 讓我們加密 DNS 挑戰（讓我們加密文檔）

使用 Flux 簡化 Amazon EKS 多租戶應用程式部署

由納迪姆·拉哈曼（AWS），阿迪亞·阿姆巴蒂（AWS），安妮特（AWS）和伯利康帕蒂爾（AWS）創建

代碼存儲庫：aws-eks-multitenancy-deployment	環境：PoC 或試點	技術： DevOps；容器與微服務
AWS 服務：AWS CodeBuild ; AWS CodeCommit; AWS CodePipeline; Amazon EKS; Amazon VPC		

Summary

許多提供產品和服務的公司都是受數據管制的行業，這些行業必須在其內部業務功能之間保持數據障礙。此模式說明如何使用 Amazon 彈性 Kubernetes 服務 (Amazon EKS) 中的多租戶功能建立資料平台，以在共用單一 Amazon EKS 叢集的租用戶或使用者之間實現邏輯和實體隔離。該模式通過以下方法提供隔離：

- 庫伯尼特命名空間隔離
- 角色型存取控制 (RBAC)
- 網路政策
- 資源配額
- AWS Identity and Access Management 服務帳戶 (IRSA) 的 (IAM) 角色

此外，此解決方案使用 Flux 在部署應用程式時保持租用戶組態不可變。您可以在組態中指定包含 Flux `kustomization.yaml` 檔案的租用戶存放庫，以部署租用戶應用程式。

此模式會實作下列項目：

- 透過手動部署 Terraform 指令碼建立的 AWS CodeCommit 儲存庫、 AWS CodeBuild 專案和 AWS CodePipeline 管線。
- 託管租用戶所需的網路和運算元件。這些是通過 CodePipeline 並通 CodeBuild 過使用地形創建的。
- 租用戶命名空間、網路原則和資源配額 (透過 Helm 圖表設定)。

- 屬於不同租用戶的應用程式，使用 Flux 部署。

我們建議您根據您獨特的需求和安全性考量，仔細規劃和建置自己的多租戶架構。此模式為您的實施提供了一個起點。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) 2.11.4 版或更新版本，已安裝並設定
- 在本機電腦上安裝的地形版本 0.12 或更新版本
- 地形 AWS 供應商 3.0.0 版或更新版本
- 庫伯內提供者 2.10 版或更新版本
- 頭盔提供者 2.8.0 版或更新版本
- Kubectl 提供者版本 1.14 或更新版本

限制

- 依賴於 Terraform 手動部署：工作流程的初始設置（包括創建 CodeCommit 存儲庫， CodeBuild 項目和 CodePipeline 管道）依賴於手動 Terraform 部署。這在自動化和可擴展性方面引入了潛在的限制，因為它需要對基礎結構更改進行手動介入。
- CodeCommit 存儲庫依賴：工作流程依賴存 CodeCommit 儲庫作為源代碼管理解決方案，並與 AWS 服務緊密結合。

架構

目標架構

此模式會部署三個模組，為資料平台建置管線、網路和運算基礎結構，如下圖所示。

管道架構：

網路架構：

運算架構：

工具

AWS 服務

- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit是一項版本控制服務，可協助您私下儲存和管理 Git 儲存庫，而無需管理您自己的原始檔控制系統。
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。
- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在上執行 Kubernetes， AWS 而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- AWS Transit Gateway 是連接虛擬私有雲端 (VPC) 和內部部署網路的中央中樞。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。此虛擬網路與您在自己的資料中心中操作的傳統網路相似，且具備使用 AWS 可擴展基礎設施的優勢。

其他工具

- 纖毛網路原則支援庫伯尼特 L3 和 L4 網路原則。它們可以使用 L7 原則進行擴充，以針對 HTTP、卡夫卡和 gRPC 以及其他類似通訊協定提供 API 層級的安全性。
- Flux 是以 Git 為基礎的持續交付 (CD) 工具，可自動在 Kubernetes 上進行應用程式部署。
- Helm 是 Kubernetes 的開放原始碼套件管理員，可協助您在 Kubernetes 叢集上安裝和管理應用程式。
- Terraform 是一種基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署資源。HashiCorp

代碼存儲庫

此模式的程式碼可在 GitHub EKS 多租戶 Terra form 解決方案存放庫中取得。

最佳實務

如需使用此實作的指導方針和最佳作法，請參閱下列內容：

- Amazon EKS 多租戶最佳實務
- 助焊劑文件

史诗

為 Terraform 建置、測試和部署階段建立管道

任務	描述	所需技能
克隆項目存儲庫。	在終端機視窗中執行下列命令，以複製 GitHub EKS 多租戶 Terraform 解決方案儲存庫： `git clone https://github.com/aws-samples/aws-eks-multitenancy-deployment.git`	AWS DevOps
引導地形 S3 儲存貯體和 Amazon DynamoDB。	1. 在`bootstrap` 資料夾中，開啟`bootstrap.sh` 檔案並更新 S3 儲存貯體名稱、DynamoDB 資料表名稱和的變數值： AWS 區域 `S3_BUCKET_NAME="<S3_BUCKET_NAME>" DYNAMODB_TABLE_NAME="<DYNAMODB_NAME>" REGION="<AWS_REGION>"` 2. 執行 `bootstrap.sh` 指令碼。此指令碼需要 AWS CLI您安裝為必要條件的一部分。	AWS DevOps

任務	描述	所需技能
	`cd bootstrap` `./bootstrap.sh`	

任務	描述	所需技能
更新run.sh和locals.tf 檔案。	1. 啟動程序順利完成後，請從指令碼variables 區段複製 S3 儲存貯體和 DynamoDB 表格名稱：bootstrap.sh `# Variables S3_BUCKET_NAME="< S3_BUCKET_NAME>" DYNAMODB_TABLE_NAME ="<DYNAMODB_NAME"` 2. 將這些值粘貼到run.sh腳本中，該腳本位於項目的根目錄中： `BACKEND_BUCKET_ID= "<SAME_NAME_AS_S3_ BUCKET_NAME>" DYNAMODB_ID=" <SAME_NAME_AS_DYNA MODB_NAME>"` 3. 將專案程式碼上傳至儲 CodeCommit 存庫。您可以true在檔案中將下列變數設定為，透過 Terraform 自動建立此儲存庫：demo/pipeline/locals.tf `create_new_repo = true` 4. 根據您的需求更新locals.tf 檔案以建立管線資源。	AWS DevOps

任務	描述	所需技能
部署管線模組。	若要建立管線資源，請手動執行下列 Terraform 指令。沒有自動執行這些命令的協調流程。 `./run.sh -m pipeline -e demo -r <AWS_REGION> -t init` `./run.sh -m pipeline -e demo -r <AWS_REGION> -t plan` `./run.sh -m pipeline -e demo -r <AWS_REGION> -t apply`	AWS DevOps

建立網路基礎架構

任務	描述	所需技能
啟動配管。	1. 在資`templates` 料夾中，請確定`buildspec` 檔案已將下列變數設定為`network`： `TF_MODULE_TO_BUILD: "network"` 2. 在CodePipeline 主控台的管線詳細資訊頁面上，選擇 [發行變更] 來啟動管線。 在第一次執行之後，每當您對 CodeCommit 儲存庫主分支提交變更時，管線就會自動啟動。	AWS DevOps

任務	描述	所需技能
	管道包括以下階段： • validate初始化 Terraform、使用檢測和 tfsec 工具執行 T erraform 安全性掃描，並將掃描報告上傳到 S3 儲存貯體。 • plan 顯示 Terraform 計劃並將計劃上傳到 S3 存儲桶。 • apply套用 S3 儲存貯體的 Terraform 計劃輸出並建立 AWS 資源。 • destroy移除apply階段期間建立的 AWS 資源。若要啟用此選擇性階段，請在demo/pipeline/locals.tf 檔案true中將下列變數設定為： `enable_destroy_stage = true`	

任務	描述	所需技能
驗證透過網路模組建立的資源。	確認管線部署成功後，已建立下列 AWS 資源： • 具有三個公用和三個私有子網路、網際網路閘道和 NAT 閘道的輸出 VPC。 • 具有三個私有子網路的 Amazon EKS VPC。 • 租用戶 1 和租用戶 2 VPC 各有三個私有子網路。 • 包含所有 VPC 附件和路由至每個私有子網路的傳輸閘道。 • 具有目標 CIDR 區塊的 Amazon EKS 出口 VPC 的靜態傳輸閘道路由。`0.0.0.0/0` 若要讓所有 VPC 都能透過 Amazon EKS 輸出 VPC 進行輸出網際網路存取，這是必要的。	AWS DevOps

建立運算基礎架構

任務	描述	所需技能
更新`locals.tf` 以啟用 CodeBuild 專案對 VPC 的存取權。	若要為 Amazon EKS 私有叢集部署附加元件，該 CodeBuild 專案必須附加至 Amazon EKS VPC。 1. 在資`demo/pipeline` 料夾中，開啟`locals.tf` 檔案，並將`vpc_enabled` 變數設定為`true`。	AWS DevOps

任務	描述	所需技能
	2. 執行指run.sh令碼，將變更套用至管線模組： `demo/pipeline/locals.tf` `./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd init` `./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd plan` `./run.sh -m pipeline -env demo -region <AWS_REGION> -tfcmd apply`	
更新buildspec 檔案以建置運算模組。	在資templates 料夾中的所有 buildspec YAML 檔案中，將TF_MODULE_TO_BUILD 變數的值從設定network為compute： `TF_MODULE_TO_BUILD: "compute"`	AWS DevOps

任務	描述	所需技能
更新租用戶管理「頭盔」圖表的values檔案。	1. 在下列位置開啟values.yaml 檔案： `cd cfg-terraform/demo/compute/cfg-tenant-mgmt` 該文件如下所示： `---` `global:` `  clusterRoles:` `  operator: platform-tenant` `  flux: flux-tenant-applier` `  flux:` `  tenantCloneBaseUrl: ${TEANT_BASE_URL}` `  repoSecret: ${TENANT_REPO_SECRET}` `tenants:` `  tenant-1:` `  quotas:` `  limits:` `  cpu: 1` `  memory: 1Gi` `  flux:` `  path: overlays/tenant-1` `  tenant-2:` `  quotas:` `  limits:` `  cpu: 1` `  memory: 2Gi` `  flux:`	AWS DevOps

任務	描述	所需技能
	<pre>path: overlays/ tenant-2</pre> 2. 在global和tenants區段中，根據您的需求更新組態： • tenantCloneBaseUrl — 託管所有租戶代碼的存儲庫路徑（我們為所有租戶使用相同的 Git 存儲庫） • repoSecret — Kubernetes 密鑰，用於保存 SSH 密鑰和已知主機以向全局租用戶 Git 存儲庫進行身份驗證 • quotas— 您想要為每個租用戶套用的 Kubernetes 資源配額 • flux path— 全域承租人存放庫中租用戶應用程式 YAML 檔案的路徑	

任務	描述	所需技能
驗證運算資源。	在上述步驟中更新檔案後，會自動 CodePipeline 啟動。確認它已為運算基礎結構建立下列 AWS 資源： • 具有私有端點的 Amazon EKS 集群 • Amazon EKS 工作者節點 • Amazon EKS 附加組件：外部秘密`aws-loadbalancer-controller`，和 `metrics-server` • GitOps 模塊，助焊劑頭盔圖，纖毛頭盔圖和租戶管理頭盔圖	AWS DevOps

檢查租戶管理和其他資源

任務	描述	所需技能
驗證 Kubernetes 中的租用戶管理資源。	執行下列命令，以檢查是否已在 Helm 的協助下成功建立租用戶管理資源。 1. 已建立租用戶命名空間，如下`values.yaml` 所示： `kubectl get ns -A` 2. 配額會指派給每個租用戶命名空間，如下所示`values.yaml` ：	AWS DevOps

任務	描述	所需技能
	`kubectl get quota --namespace=<tenant_namespace>` 3. 每個承租人命名空間的配額詳細資料都是正確的 `kubectl describe quota cpu-memory-resource-quota-limit -n <tenant_namespace>` 4. Cilium 網路原則已套用至每個租用戶命名空間： `kubectl get CiliumNetworkPolicy -A`	

任務	描述	所需技能
驗證租戶應用程式部署。	執行下列命令以確認承租人應用程式是否已部署。 1. Flux 能夠連接到 GitOps 模塊中指定的 CodeCommit 存儲庫： `kubectl get gitrepositories -A` 2. Flux 系統化控制器已在儲存庫中部署 YAML 檔案：CodeCommit `kubectl get kustomizations -A` 3. 所有應用程式資源都會部署在其租用戶命名空間中： `kubectl get all -n <tenant_namespace>` 4. 已為每個租用戶建立輸入： `kubectl get ingress -n <tenant_namespace>`	

故障診斷

問題	解決方案
您遇到類似下列內容的錯誤訊息： `Failed to checkout and determine revision: unable to clone unknown`	請依照下列步驟疑難排解問題：

問題	解決方案
`error: You have successfully authenticated over SSH. You can use Git to interact with AWS CodeCommit.`	1. 驗證承租人應用程式儲存區域：空白或設定錯誤的儲存區域可能造成錯誤。確定承租人應用程式存放庫包含所需的程式碼。 2. 重新部署`tenant_mgmt` 模組：在`tenant_mgmt` 模組設定檔中，找出`app`區塊，然後將`deploy`參數設定為`0`： `deploy = 0` 執行 Terraform `apply` 指令之後，請將`deploy`參數值變回：`1` `deploy = 1` 3. 重新檢查狀態：執行上述步驟之後，請使用下列命令來檢查問題是否仍然存在： `kubectl get gitrepositories -A` 如果仍然存在，請考慮深入了解 Flux 日誌以獲取更多詳細信息，或參閱 Flux 一般故障排除指南。

相關資源

- 地形的 Amazon EKS 藍圖
- Amazon EKS 最佳實務指南，多租戶部分
- 通量網站
- 頭盔網站

其他資訊

以下是部署租用戶應用程式的存放庫結構範例：

```
applications
sample_tenant_app
### README.md
### base
#   ### configmap.yaml
#   ### deployment.yaml
#   ### ingress.yaml
#   ### kustomization.yaml
#   ### service.yaml
### overlays
    ### tenant-1
    #   ### configmap.yaml
    #   ### deployment.yaml
    #   ### kustomization.yaml
    ### tenant-2
        ### configmap.yaml
        ### kustomization.yaml
```

使用自訂資源，將多個電子郵件端點訂閱至 SNS 主題

由里卡多·莫拉斯（AWS）創建

環境：生產	技術： DevOps	AWS 服務：Amazon SNS；AWS CloudFormation；AWS Lambda

Summary

注意，2022 年 8 月：AWS CloudFormation 現在支援透過AWS::SNS::Topic物件及其訂閱屬性訂閱多個資源。

此模式說明如何訂閱多個電子郵件地址，以接收來自 Amazon Simple Notification Service (Amazon SNS) 主題的通知。它使用 AWS Lambda 函數做為 AWS CloudFormation 範本中的自訂資源。Lambda 函數與輸入參數相關聯，該參數可指定 SNS 主題的電子郵件端點。

目前，您可以使用 AWS CloudFormation 範本物件，AWS::SNS::Topic並訂AWS::SNS::Subscription閱 SNS 主題的單一端點。若要訂閱多個端點，您必須多次呼叫物件。透過使用 Lambda 函數做為自訂資源，您可以透過輸入參數訂閱多個端點。您可以在任何 AWS CloudFormation 範本中使用此 Lambda 函數做為自訂資源。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在您的本機環境中使用存取金鑰和秘密金鑰設定的 AWS 設定檔。您也可以從 AWS Cloud9 執行此程式碼。
- 下列項目的權限：
  - AWS Identity and Access Management (IAM) 角色和政策
  - AWS Lambda 功能
  - Amazon Simple Storage Service (Amazon S3)，用於上傳 Lambda 函數
  - Amazon SNS 主題和政策
  - AWS CloudFormation 堆疊

限制

- 該代碼支持 Linux 和 macOS 工作站。

產品版本

- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版或更新版本。

架構

目標技術堆疊

- AWS CloudFormation
- Amazon SNS
- AWS Lambda

工具

工具

- AWS CLI 第 2 版

Code

附件包含下列檔案：

- Lambda 函數：`lambda_function.py`
- AWS CloudFormation 範本：`template.yaml`
- 用於處理多個或單個電子郵件端點訂閱的兩個參數文件：`parameters-multiple-values.json`（用作默認值）和 `parameters-one-value.json`

若要部署堆疊，您可以使用任一參數檔案。若要指定多個電子郵件端點：

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION>
```

若要指定單一電子郵件端點：

```
./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json
```

史诗

選項 1-使用一個電子郵件訂閱部署 SNS 主題

任務	描述	所需技能
設定 SNS 主題訂閱的電子郵件端點。	編輯檔案 parameters-one-value.json （附加），並變更pSNSNotificationsEmail 參數值以反映您要使用的電子郵件地址，例如someone@example.com 。	
部署建立資源和訂閱的 AWS CloudFormation 堆疊。	使用您的 AWS 設定檔名稱、AWS 區域和parameters-one-value.json 檔案執行 deploy.sh 命令。 `./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION> -f parameters-one-value.json`	具有適當許可的 IAM 角色

選項 2-部署具有兩個或多個電子郵件訂閱的 SNS 主題

任務	描述	所需技能
設定 SNS 主題訂閱的電子郵件端點。	編輯檔案 parameters-multiple-values.json (附加)，然後變更pSNSNotificationsEmail 參數值	

任務	描述	所需技能
	以反映您要使用的電子郵件地址，並以逗號分隔，如下所示：someone1@example.com, someone2@example.com 。	
部署建立資源和訂閱的 AWS CloudFormation 堆疊。	使用您的 AWS 設定檔名稱和 AWS 區域執行 deploy.sh 命令。您不必指定parameters-multiple-values.json 檔案，因為預設會使用該檔案。 `./deploy.sh -p <YOUR_AWS_PROFILE_NAME> -r <YOUR_AWS_PROFILE_REGION>`	具有適當許可的 IAM 角色

選項 3-透過 AWS CloudFormation 範本部署 SNS 主題

任務	描述	所需技能
建立 SNS 主題。	透過 AWS CloudFormation 範本建立 SNS 主題，而不需在範本物件中指定訂閱端點。AWS::SNS::Topic 您可以template.yaml 在附件中作為起點使用。	具有適當許可的 IAM 角色
建立 SNS 主題原則。	在 AWS CloudFormation 範本中建立 SNS 主題政策。	具有適當許可的 IAM 角色
訂閱 SNS 主題的電子郵件端點清單。	根據電子郵件端點清單 (一或多個)，為端點訂閱您建立的 SNS 主題。	具有適當許可的 IAM 角色

相關資源

參考

- AWS CloudFormation 自訂資源 (AWS 文件)
- 使用 Python、AWS Lambda 和小幫手建立 AWS CloudFormation 自訂資源 (部落格文章)

所需工具

- AWS CLI 第 2 版

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Serverspec 進行基礎架構程式碼的測試驅動開發

由蘇珊特杰格代爾（AWS）創建

環境：PoC 或試點	技術： DevOps; 基礎架構; 混合雲	AWS 服務：Amazon EC2; AWS CodeBuild; AWS CodeDeploy

Summary

此模式向您展示在 Amazon Web Services（AWS）雲上編寫基礎設施代碼時，如何使用 Serverspec 使用測試驅動開發（TDD）。該模式也涵蓋 AWS 的自動化 CodePipeline。TDD 將注意力集中在基礎設施代碼必須做什麼，並設置一個清晰的完成定義。您可以使用伺服器規格來測試 AWS CloudFormation、Terraform 依據和 Ansible 等工具建立的基礎設施。 HashiCorp

伺服器規格可協助重構基礎架構程式碼。使用 Serverspec，您可以編寫 RSpec 測試來檢查各種軟件包和軟件的安裝，運行命令，檢查正在運行的進程和端口，檢查文件權限設置，等等。伺服器規格會檢查您的伺服器是否設定正確。您只能在伺服器上安裝 Ruby。您不需要安裝任何代理程式軟體。

測試驅動的基礎架構具有以下優點：

- 跨平台測試
- 驗證期望
- 對自動化充滿信心
- 基礎架構一致性與穩
- 提早失敗

您可以使用此模式為 Apache 軟體執行伺服器規格單元測試，並在建立 Amazon 機器映像 (AMI) 期間檢查檔案權限設定。只有在所有測試用例通過時，才會創建 AMI。伺服器規格將執行下列測試：

- 阿帕奇進程正在運行。
- 阿帕奇端口正在運行。
- Apache 配置文件和目錄存在於某些位置，等等。
- 檔案權限已正確設定。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- 具有公有子網路的虛擬私有雲 (VPC)
- AWS Command Line Interface (AWS CLI) (AWS CLI) 和 Git 的安裝

產品版本

- HashiCorp 封裝程式版本：1.6.6
- 紅寶石版本：2.5.1 及更高版本
- AWS CLI 版本：

架構

目標架構

1. 當您將程式碼推送至 CodeCommit 儲存庫時，Amazon CloudWatch 事件會參與. CodePipeline 在管道的第一階段，代碼是從 CodeCommit中獲取。
2. 第二個管線階段會執行 CodeBuild，此階段會驗證並建置 Packer 範本。
3. 作為打包器構建佈建程序的一部分，打包器安裝 Apache 和 Ruby 軟件。然後佈建程式會呼叫使用 Serverspec 來單元測試 Apache 處理序、連接埠、檔案和目錄的殼層指令碼。封裝程式後處理器會寫入 JavaScript 物件標記法 (JSON) 檔案，其中包含 Packer 在執行期間產生的所有成品清單
4. 最後，亞馬遜彈性運算雲端 (Amazon EC2) 執行個體是使用封包器產生的 AMI ID 建立的。

工具

- AWS CLI — Amazon Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可使用命令列殼層中的命令與 AWS 服務互動。

- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供了一系統事件 near-real-time 流，這些事件描述了 Amazon Web Services（AWS）資源的變化。
- AWS CodeBuild — AWS CodeBuild 是雲端中的全受管建置服務。 CodeBuild 編譯您的原始程式碼、執行單元測試，並產生準備好部署的成品。
- AWS CodeCommit — AWS CodeCommit 是由亞馬遜網路服務託管的版本控制服務。您可以使 CodeCommit 用在雲端中私有儲存和管理資產 (例如文件、原始程式碼和二進位檔案)。
- AWS CodePipeline — AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。您可以使用快速模型化和設定軟體發行程序的不同階段。
- HashiCorp Packer — P HashiCorp acker 是一種工具，用於從單一來源配置自動創建相同的機器映像。
- 伺服器規格 — 伺服器規格執行 RSpec 測試以檢查伺服器組態。伺服器規格使用 Ruby，而且您不需要安裝代理程式軟體。

Code

代碼已附上。該代碼使用以下結構，具有三個目錄和八個文件。

```
### amazon-linux_packer-template.json (Packer template)
### buildspec.yaml (CodeBuild .yaml file)
### pipeline.yaml (AWS CloudFormation template to automate CodePipeline)
### rspec_tests (RSpec required files and spec)
#   ### Gem-file
#   ### Rakefile
#   ### spec
#       ### apache_spec.rb
#       ### spec_helper.rb
### scripts
    ### rspec.sh (Installation of Ruby and initiation of RSpec)
```

史诗

設定 AWS 登入資料

任務	描述	所需技能
建立 IAM 使用者。	建立具有程式設計和主控台存取權的 AWS Identity and	開發者, 系統管理員, DevOps 工程師

任務	描述	所需技能
	Access Management (IAM) 使用者。如需詳細資訊，請參閱 AWS 文件。	
設定 AWS 登入資料。	在本機電腦或環境中，為 IAM 使用者設定 AWS 登入資料。如需指示，請參閱 AWS 文件。	開發者, 系統管理員, DevOps 工程師
測試您的認證。	若要驗證設定的認證，請執行下列命令。 `aws sts get-caller-identity --profile <profile>`	開發者, 系統管理員, DevOps 工程師

AWS CodePipeline

任務	描述	所需技能
創建一個 CodeCommit 存儲庫。	若要建立 CodeCommit 存放庫，請執行下列命令。 `aws codecommit create-repository --repository-name "<provide repository-name>" --repository-description "repository to unit test the infrastructure code"`	開發者, 系統管理員, DevOps 工程師
編寫 RSpec 的測試。	為您的基礎結構建立 RSpec 測試案例。如需詳細資訊，請參閱其他資訊一節。	開發者、 DevOps 工程師

任務	描述	所需技能
將代碼推送到 CodeCommit 存儲庫。	若要將附加的程式碼推送至 CodeCommit 儲存庫，請執行下列命令。 `git clone <repository url>` `cp -R /tmp/<code folder>/ <repository_folder>/` `git add .` `git commit -m"initial commit"` `git push`	開發者, 系統管理員, DevOps 工程師
建立管線。	若要建立管道，請執行其他資訊一節中的 AWS CLI 命令。	開發者, 系統管理員, DevOps 工程師
啟動配管。	將程式碼提交至 CodeCommit 儲存庫。對存儲庫的任何提交都將啟動管道。	開發者, 系統管理員, DevOps 工程師
測試阿帕奇網址。	若要測試 AMI 安裝，請使用下列 URL。 `http://<your instance public ip>/hello.html` 該頁面將顯示一個「你好從阿帕奇」消息。	開發者, 系統管理員, DevOps 工程師

相關資源

- HashiCorp
- HashiCorp 打包機
- 伺服器規格

- 簡介 ServerSpec：什麼是伺服器規格以及我們如何在 Stelligent 中使用它？（外部博客文章）
- 基礎架構程式碼的測試驅動開發 (外部部落格文章)
- 使用 HashiCorp Packer 和 ServerSpec（外部文章）創建和測試映像

其他資訊

編寫 RSpec 的測試

此模式的 RSpec 測試位於<repository folder>/rspec_tests/spec/apache_spec.rb。

```
require 'spec_helper'


describe service('httpd') do
  it { should be_enabled }
  it { should be_running }
end


describe port(80) do
  it { should be_listening }
end


describe file('/etc/httpd/conf/httpd.conf') do
  it { should exist }
  it { should be_owned_by 'root' }
  it { should contain 'ServerName www.example.com' }
end


describe file('/etc/httpd/conf/httpd.conf') do
  its(:content) { should match /ServerName www.example.com/ }
end


describe file('/var/www/html/hello.html') do
  it { should exist }
  it { should be_owned_by 'ec2-user' }
end
```

```
describe file('/var/log/httpd') do
  it { should be_directory }
end


describe file('/etc/sudoers') do
  it { should be_mode 440 }
end


describe group('root') do
  it { should have_gid 0 }
end
```

您可以將自己的測試添加到目/spec錄中。

建立管線

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
pipeline.yaml --parameters ParameterKey=RepositoryName,ParameterValue=<provide
 repository-name> ParameterKey=ApplicationName,ParameterValue=<provide
 application-name> ParameterKey=SecurityGroupId,ParameterValue=<provide
 SecurityGroupId> ParameterKey=VpcId,ParameterValue=<provide VpcId>
 ParameterKey=SubnetId,ParameterValue=<provide SubnetId> ParameterKey=Region,ParameterValue=<pr
 AccountId> --capabilities CAPABILITY_NAMED_IAM
```

參數詳情

repository-name— AWS CodeCommit 儲存庫的名稱

application-name— Amazon 資源名稱 (ARN) 與鏈接ApplicationName; 提供任何名稱

SecurityGroupId— AWS 帳戶中已開啟連接埠 80 的任何安全群組 ID

VpcId— 您的虛擬私人 VPC 的識別碼

SubnetId— VPC 中公有子網路的識別碼

Region— 您執行此模式的 AWS 區域

Keypair— 用於登入 EC2 執行個體的安全殼層 (SSH) 金鑰名稱

AccountId— 您的 AWS 帳戶識別碼

您也可以使用 AWS 管理主控台並傳遞上一 CodePipeline 個命令列中的相同參數來建立管道。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在 AWS 中使用第三方 Git 來源儲存庫 CodePipeline

環境：PoC 或試點	技術： DevOps	工作負載：開源
AWS 服務：AWS CodeBuild ；AWS CodePipeline ；AWS		

Summary

此模式說明如何搭 CodePipeline 配第三方 Git 來源儲存庫使用 AWS。

AWS CodePipeline 是一種持續交付服務，可自動執行建置、測試和部署軟體的任務。該服務目前支援由 GitHub AWS CodeCommit 和阿特拉西亞比特桶管理的 Git 儲存庫。不過，有些企業會使用與其單一登入 (SSO) 服務和 Microsoft Active Directory 整合的協力廠商 Git 儲存庫進行驗證。您可以 CodePipeline 透過建立自訂動作和 Webhook 來使用這些第三方 Git 儲存庫做為來源。

webhook 是一種 HTTP 通知，用於檢測另一個工具（例如 GitHub 存儲庫）中的事件，並將這些外部事件連接到管道。當您在中建立 Webhook 時 CodePipeline，服務會傳回您可以在 Git 儲存庫 Webhook 中使用的網址。如果您將程式碼推送到 Git 儲存庫的特定分支，Git Webhook 會透過此 URL 啟動 CodePipeline Webhook，並將管線的來源階段設定為「進行中」。當管線處於此狀態時，工作者會輪詢 CodePipeline 自訂工作、執行工作，然後將成功或失敗狀態傳送至 CodePipeline。在這種情況下，由於管道處於來源階段，因此工作者會使用輪詢任務提供的物件金鑰，取得 Git 儲存庫的內容、壓縮內容，然後將其上傳到 Amazon Simple Storage Service (Amazon S3) 儲存貯體，在此儲存管道的成品。您也可以將自訂動作的轉換與 Amazon 中的事件相關聯 CloudWatch，並根據事件啟動工作者。此設定可讓您使用服務本身不支援的第三方 Git 儲存庫做為 CodePipeline來源。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 一個 Git 存儲庫，支持網絡掛鉤並可以通過互聯網連接到一個 CodePipeline 網絡掛鉤網址
- AWS Command Line Interface (AWS CLI) (AWS CLI) 已安裝並設定為與 AWS 帳戶搭配使用

架構

該模式包括以下步驟：

1. 使用者將程式碼提交至 Git 儲存庫。
2. Git 網絡掛鉤被調用。
3. CodePipeline 網絡掛鉤被調用。
4. 管線會設定為「進行中」，且來源階段會設定為「進行中」狀態。
5. 來源階段動作會啟動「 CloudWatch 事件」規則，表示該規則已啟動。
6. 此 CloudWatch 事件會啟動一個 Lambda 函數。
7. Lambda 函數會取得自訂動作工作的詳細資料。
8. Lambda 函數會啟動 AWS， CodeBuild 並傳遞所有與工作相關的資訊。
9. CodeBuild 從 Secrets Manager 獲取公開安全殼層金鑰或 HTTPS Git 存取的使用者認證。
10. CodeBuild 克隆特定分支的 Git 存儲庫。
11. CodeBuild 壓縮存檔並將其上傳到作為 CodePipeline 成品存放區的 S3 儲存貯體。

工具

- AWS CodePipeline — AWS CodePipeline 是全受管的持續交付服務，可協助您自動化發行管道，以快速可靠地更新應用程式和基礎設施。 CodePipeline 根據您定義的發行模型，針對每個程式碼變更，自動執行發行程序的建置、測試和部署階段。這使您能夠快速可靠地提供功能和更新。您可以將 AWS CodePipeline 與第三方服務 (例如， GitHub 或使用您自己的自訂外掛程式) 整合。
- AWS Lambda — AWS Lambda 可讓您執行程式碼，而無需佈建或管理伺服器。使用 Lambda，您可以為幾乎任何類型的應用程式或後端服務執行程式碼，無需管理。您可以上傳程式碼，Lambda 會處理以高可用性執行和擴展程式碼所需的一切。您可以將程式碼設定為從其他 AWS 服務自動啟動，或直接從任何 Web 或行動應用程式呼叫程式碼。
- AWS CodeBuild — AWS CodeBuild 是全受管的持續整合服務，可編譯原始程式碼、執行測試，以及產生可立即部署的軟體套件。有了 CodeBuild，您不需要佈建、管理和擴展自己的組建伺服器。CodeBuild 持續擴展並同時處理多個構建，因此您的構建不會留在隊列中等待。您可以利用預先封裝好的組建環境立即開始使用，或是建立自訂的組建環境來使用您自己的組建工具。
- AWS Secrets Manager — AWS Secrets Manager 可協助您保護存取應用程式、服務和 IT 資源所需的機密。此服務可讓您輪換、管理和擷取資料庫登入資料、API 金鑰和其他機密的整個生命週期。使

用者和應用程式可透過呼叫 Secrets Manager API 擷取密碼，而不必以純文字格式對敏感資訊進行硬式編碼。Secrets Manager 提供秘密輪換與 Amazon 關係數據庫服務（亞馬遜 RDS），Amazon Redshift 和 Amazon DocumentDB 的內置集成。該服務可以擴展以支持其他類型的密鑰，包括 API 密鑰和 OAuth 令牌。此外，Secrets Manager 可讓您使用精細的許可來控制密碼的存取，以及針對 AWS 雲端、第三方服務和現場部署環境中的資源集中稽核機密輪替。

- Amazon CloudWatch — Amazon CloudWatch 是專為 DevOps 工程師、開發人員、網站可靠性工程師 (SRE) 和 IT 經理打造的監控和觀察服務。 CloudWatch 為您提供資料和可行的見解，以監控您的應用程式、回應整個系統的效能變化、最佳化資源使用率，以及取得營運狀態的統一檢視。CloudWatch 以日誌、指標和事件的形式收集監控和操作資料，為您提供在 AWS 和現場部署伺服器上執行的 AWS 資源、應用程式和服務的統一檢視。您可以使用 CloudWatch 來偵測環境中的異常行為、設定警示、並列視覺化記錄和指標、採取自動化動作、疑難排解問題，以及探索洞察，以確保應用程式順暢執行。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，可讓您針對各種使用案例存放和保護任意數量的資料，例如網站、行動應用程式、備份和還原、存檔、企業應用程式、IoT 裝置和大數據分析。Amazon S3 提供 easy-to-use 管理功能來協助您組織資料，並設定精細調整的存取控制，以符合您的特定業務、組織和合規要求。

史诗

在中建立自訂動作 CodePipeline

任務	描述	所需技能
使用 AWS CLI 或 AWS 建立自訂動作 CloudFormation。	此步驟涉及建立可用於特定區域 AWS 帳戶中管道的來源階段的自訂來源動作。您必須使用 AWS CLI 或 AWS CloudFormation (而非主控台) 來建立自訂來源動作。有關本和其他史詩中描述的命令和步驟的更多信息，請參閱此模式末尾的「相關資源」一節。在 AWS CLI 中，使用 create-custom-action-type 命令。使用--Configuration-屬性來提供工作者輪 CodePipeline 詢工	一般 AWS

任務	描述	所需技能
	作時所需的所有參數，以供工作者處理。請務必記下提供給--provider 和-action 版本選項的值，以便您可以在使用此自訂來源階段建立管線時使用相同的值。您也可以使用資源類型類 AWS::CodePipeline::CustomAction型在 AWS CloudFormation 中建立自訂來源動作。	

設定驗證

任務	描述	所需技能
建立 SSH 金鑰對。	建立安全殼層 (SSH) key pair。如需指示，請參閱 GitHub 文件。	系統/工程師 DevOps
在 AWS 秘密管理員中建立密碼。	從安全殼層 key pair 複製私密金鑰的內容，並在 AWS Secrets Manager 中建立密碼。訪問 Git 存儲庫時，此密鑰用於身份驗證。	一般 AWS
將公鑰添加到 Git 存儲庫。	將 SSH key pair 中的公開金鑰新增至 Git 儲存庫帳戶設定，以便針對私密金鑰進行驗證。	系統/工程師 DevOps

建立管線和網路掛鉤

任務	描述	所需技能
建立包含自訂來源動作的管道。	在中建立管線 CodePipeline。當您設定來源階段時，請選擇您先前建立的自訂來源動作。您可以在 AWS CodePipeline 主控台或 AWS CLI 中執行此操作。 CodePipeline 會提示您輸入在自訂動作上設定的組態屬性。工作 Worker 需要此資訊才能處理自訂動作的工作。遵循精靈並建立管線的下一個階段。	一般 AWS
創建一個 CodePipeline 網絡掛鉤。	為您使用自訂來源動作建立的管線建立 Webhook。您必須使用 AWS CLI 或 AWS CloudFormation （不是控制台）來創建網絡掛鉤。在 AWS CLI 中，執行推桿命令，並為網路掛接選項提供適當的值。記下命令返回的網絡掛鉤 URL。如果您使用 AWS CloudFormation 建立 Webhook，請使用資源類型 AWS::CodePipeline::Webhook。確保從創建的資源中輸出 webhook URL，並記下它。	一般 AWS
建立 Lambda 函數和 CodeBuild 專案。	在此步驟中，您可以使用 Lambda 並 CodeBuild 建立工作工作者，以輪詢 CodePipeline 自訂動作的工作請求、執行工作，並將狀態結果傳回	一般 AWS、程式碼開發人員

任務	描述	所需技能
	至 CodePipeline。當管道的自訂來源動作階段轉換為「進行中」時，建立由 Amazon CloudWatch 事件規則啟動的 Lambda 函數。啟動 Lambda 函數時，應透過輪詢工作來取得自訂動作工作詳細資料。您可以使用 PollForJobs API 傳回此資訊。取得輪詢的工作資訊之後，Lambda 函數應傳回確認，然後處理資訊及其從自訂動作的組態屬性中取得的資料。當 Worker 準備好與 Git 儲存庫通訊時，您可以啟動 CodeBuild 專案，因為使用 SSH 用戶端來處理 Git 工作很方便。	

在中建立活動 CloudWatch

任務	描述	所需技能
建立 CloudWatch 事件規則。	建立一個 CloudWatch 事件規則，在管道的自訂動作階段轉換為「進行中」時，啟動 Lambda 函數作為目標。	一般 AWS

相關資源

在中建立自訂動作 CodePipeline

- 在中建立並新增自訂動作 CodePipeline
- AWS::CodePipeline::CustomAction類型資源

設定驗證

- 使用 AWS Secrets Manager 建立和管理機密

建立管線和網路掛鉤

- 在中建立配管 CodePipeline
- 置入網路掛接指令參考
- AWS::CodePipeline::Webhook 資源
- PollForJobs API 參考
- 在中建立並新增自訂動作 CodePipeline
- 在 AWS 中建立建置專案 CodeBuild

建立事件

- 使用 Amazon CloudWatch 事件偵測管道狀態的變更並做出回應

其他參考

- 使用中的管線 CodePipeline
- AWS Lambda 開發人員指南

使用 AWS 建立 CI/CD 管道以驗證地形組態 CodePipeline

創建者：芳香拉吉傑亞拉揚（AWS）和維傑亞庫馬蘭奈爾（AWS）

代碼存儲庫：aws-codepipeline-terraform-cicd-示例	環境：PoC 或試點	技術： DevOps
工作負載：所有其他工作	AWS 服務：AWS CodeBuild；AWS CodeCommit；AWS；Amazon S3 CodePipeline；AWS Identity and Access Management	

Summary

此模式顯示如何使用 AWS 部署的持續整合和持續交付 (CI/CD) 管道來測試 HashiCorp Terraform 組態。 CodePipeline

Terraform 是一個命令列介面應用程式，可協助您使用程式碼來佈建和管理雲端基礎結構和資源。此模式中提供的解決方案會建立 CI/CD 管線，藉由執行五個階段來協助您驗證 Terraform 組態的完整性：CodePipeline

1. “checkout”從 AWS CodeCommit 儲存庫中提取您正在測試的 Terraform 組態。
2. “validate”運行 infrastructure-as-cod （ IaC ）驗證工具，包括 tfsec，火石和切科夫。該階段還運行以下地形 IaC 驗證命令：和。`terraform validate` `terraform fmt`
3. “plan”顯示套用 Terraform 組態時，會將哪些變更套用至基礎結構。
4. “apply”使用產生的計畫，在測試環境中佈建所需的基礎結構。
5. “destroy”移除“apply”階段期間建立的測試基礎結構。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- Git，在本地計算機上安裝和配置
- 地形，在本地計算機上安裝和配置

限制

- 此模式的方法僅 CodePipeline 將 AWS 部署到一個 AWS 帳戶和 AWS 區域中。多帳戶和多區域部署需要變更組態。
- 此模式佈建 (程式碼管線 _iam_role) 的 AWS Identity and Access Management (IAM) 角色遵循最低權限原則。此 IAM 角色的許可必須根據管道需要建立的特定資源進行更新。

產品版本

- AWS CLI 版本 2.9.15 或更新版本
- 地形版本 1.3.7 或更高版本

架構

目標技術堆疊

- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- AWS IAM
- Amazon Simple Storage Service (Amazon S3)
- AWS Key Management Service (AWS KMS)
- 地形

目標架構

下圖顯示用於在中測試 Terraform 組態的 CI/CD 管線工作流程範例。 CodePipeline

該圖顯示以下工作流程：

1. 在中 CodePipeline，AWS 使用者透過在 AWS CLI 中執行`terraform apply`命令來啟動 Terraform 計劃中提出的動作。
2. AWS CodePipeline 擔任 IAM 服務角色，其中包含存取 CodeCommit CodeBuild、AWS KMS 和 Amazon S3 所需的政策。
3. CodePipeline 執行“`checkout`”管道階段，從 AWS CodeCommit 儲存庫提取 Terraform 組態以進行測試。
4. CodePipeline 運行“`validate`”階段通過運行 IaC 驗證工具，並在項目中運行 Terraform IaC 驗證命令來測試 Terraform 配置。 CodeBuild
5. CodePipeline 執行“`plan`”階段，以根據 Terraform 組態在 CodeBuild 專案中建立計劃。AWS 使用者可以在將變更套用至測試環境之前檢閱此計劃。
6. Code Pipeline 會使用 CodeBuild 專案在測試環境中佈建所需的基礎結構，來執行“`apply`”階段來實作計劃。
7. CodePipeline 執行“`destroy`”階段，此階段會用 CodeBuild 來移除“`apply`”階段期間建立的測試基礎結構。
8. Amazon S3 儲存貯體存放管道成品，這些成品透過使用 AWS KMS 客戶受管金鑰進行加密和解密。

工具

工具

AWS 服務

- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他服務

- HashiCorp Terraform 是一個命令列介面應用程式，可協助您使用程式碼來佈建和管理雲端基礎結構和資源。

Code

此模式的代碼可在 GitHub aws-codepipeline-terraform-cicdsamples存儲庫中找到。存儲庫包含創建此模式中概述的目標體系結構所需的 Terraform 配置。

史诗

佈建解決方案元件

任務	描述	所需技能
克隆存 GitHub 儲庫。	在終端機視窗中執行下列命令來複製 GitHubaws-codepipeline-terraform-cicdsamples儲存庫： `git clone https://github.com/aws-samples/aws-codepipeline-terraform-cicd-samples.git` 如需詳細資訊，請參閱 GitHub 文件中的複製存放庫。	DevOps 工程師
創建一個地形變量定義文件。	根據您的使用`terraform.tfvars`案例需求建立檔案。您可以更新複製儲存庫中`examples/terraform.tfvars`檔案中的變數。 如需詳細資訊，請參閱 Terraform 文件中的指派根模組變數的值。	DevOps 工程師

任務	描述	所需技能
	注意：儲存庫的Readme.md 檔案包含有關所需變數的詳細資訊。	
將 AWS 設定為地形表單供應商。	1. 在程式碼編輯器中，開啟複製的儲存庫main.tf檔案。 2. 新增必要的組態，以建立目標 AWS 帳戶的連線。 如需詳細資訊，請參閱 Terraform 文件中的 AWS 供應商。	DevOps 工程師

任務	描述	所需技能
更新用於建立 Amazon S3 複寫儲存貯體的 Terraform 提供者組態。	1. 執行下列命令以開啟儲存庫的S3目錄： `cd ./modules/s3` 2. 透過更新檔案中的region值，更新用於建立 Amazon S3 複寫儲存貯體的 Terraform 提供者組態。tf確保您進入了希望 Amazon S3 複寫物件的目標區域。 3. (選擇性) 依預設，Terraform 會使用本機狀態檔案進行狀態管理。如果您想要將 Amazon S3 新增為遠端後端，則必須更新 Terraform 組態。如需詳細資訊，請參閱 Terraform 文件中的後端設定。 注意：複寫會啟用 Amazon S3 儲存貯體中物件的自動非同步複製。	DevOps 工程師
初始化地形組態。	若要初始化包含 Terraform 組態檔的工作目錄，請在複製的儲存庫的根資料夾中執行下列命令： `terraform init`	DevOps 工程師

任務	描述	所需技能
建立地形計劃。	若要建立 Terraform 計劃，請在複製的儲存庫的根資料夾中執行下列命令： `terraform plan --var-file=terraform.tfvars -out=tfplan` 注意：Terraform 會評估組態檔案，以判斷已宣告資源的目標狀態。然後將目標狀態與目前狀態進行比較，並建立計劃。	DevOps 工程師
驗證地形計劃。	檢閱 Terraform 計劃，並確認其在目標 AWS 帳戶中設定所需的架構。	DevOps 工程師
部署解決方案。	1. 若要套用 Terraform 計劃，請在複製的儲存庫的根資料夾中執行下列命令： `terraform apply "tfplan"` 2. 輸入 yes 以確認您要部署資源。 注意：Terraform 會建立、更新或銷毀基礎結構，以達到組態檔案中宣告的目標狀態。	DevOps 工程師

透過執行管線來驗證地形組態

任務	描述	所需技能
設定原始程式碼儲存庫。	1. 從 Terraform 輸出中，取得儲存庫的來源儲存庫詳細資料，其中包含您要驗證的 Terraform 組態。 2. 登入 AWS 管理主控台。然後，開啟主CodeCommit 控台。 3. 在名為的源存儲庫中創建一個新分支`main`。如需指示，請參閱 CodeCommit 文件中的在 AWS CodeCommit 中建立分支。 4. 將來源儲存庫的`main`分支複製到本機工作站。如需指示，請參閱 CodeCommit 文件中的使用 AWS CLI 登入資料協助程式在 Windows 上進行 HTTPS 連線至 AWS CodeCommit 儲存庫的設定步驟。 5. 執行下列命令，從 GitHubaws-codepipeline-terraform-cicdsamples儲存庫複製**`templates`** 資料夾： `cp -r templates $YOUR_CODECOMMIT_REPO_ROOT` 附註：此資`templates` 料夾包含組建規格檔案和來源	DevOps 工程師

任務	描述	所需技能
	儲存庫根目錄的驗證指令集。 6. 將所需的 Terraform IaC 配置添加到源存儲庫的根文件夾中。 7. 在項目的 Terraform 配置中添加遠程後端的詳細信息。如需詳細資訊，請參閱地形文件中的 S3。 8. (選擇性) 更新`templates`資料夾中的變數，以啟動或停用預先設定的掃描、工具變更版本，以及在自訂指令碼檔案中指定您的目錄。如需詳細資訊，請參閱此模式的其他資訊一節。 9. 將更改推送到源存儲庫的`main`分支。	

任務	描述	所需技能
驗證管道階段。	1. 登入 AWS 管理主控台，並開啟 CodePipeline 主控台。 2. 在上一個 Epic 部分的`terraform apply "tfplan"`命令生成的輸出中，找到生成的名稱CodePipeline。 3. 在 CodePipeline 主控台中開啟管道，然後選擇 [釋出變更]。 4. 檢閱每個管道階段，並確認其正常運作。 如需詳細資訊，請參閱 AWS CodePipeline 使用者指南中的檢視管道詳細資訊和歷史記錄 (主控台)。 重要事項：將變更提交至來源儲存庫的主分支時，會自動啟動測試管線。	DevOps 工程師

任務	描述	所需技能
驗證報告輸出。	1. 在CodePipeline 主控台的左側導覽窗格中，選擇 [建置]。然後，選擇報告歷史記錄。 2. 檢閱管線產生的 tfsec 和檢查掃描報告。這些報告可協助您透過視覺化和圖形表示來識別問題。 注意：`<project_name>-validate` CodeBuild 專案會在“validate” 階段期間為您的程式碼產生弱點報告。	DevOps 工程師

清除您的資源

任務	描述	所需技能
清理管道和相關資源。	若要從 AWS 帳戶刪除測試資源，請在複製的儲存庫的根資料夾中執行下列命令： `terraform destroy --var-file=terraform.tfvars`	DevOps 工程師

故障診斷

問題	解決方案
您在“apply”階段期間收到AccessDenied 錯誤訊息。	1. 檢閱與“apply”階段相關聯之 CodeBuild 專案的執行記錄，以識別任何遺失的 IAM 許可。如需詳細資訊，請參閱 AWS CodeBuild

問題	解決方案
	使用者指南中的檢視 AWS CodeBuild 中的組建詳細資訊。 2. 在程式碼編輯器中，開啟複製的儲存庫`modules`資料夾。然後，導航到該文件`iam-role`夾並打開該`main.tf`文件夾中的文件。 3. 在`codepipeline_policy`　聲明中，新增 AWS 帳戶中佈建資源所需的 IAM 政策。

相關資源

- 模塊塊（地形文檔）
- 如何使用 CI/CD 透過 Terraform 部署和設定 AWS 安全服務 (AWS 部落格文章)
- 使用服務連結角色 (IAM 文件)
- 建立管線 (AWS CLI 文件)
- 為 Amazon S3 中存放的成品設定伺服器端加密 CodePipeline (AWS CodePipeline 文件)
- AWS 的配額 CodeBuild (AWS CodeBuild 文件)
- AWS 中的資料保護 CodePipeline (AWS CodePipeline 文件)

其他資訊

自定義地形模塊

以下是在此模式中使用的自定義 Terraform 模塊的列表：

- `codebuild_terraform`創建形 CodeBuild 成管道的每個階段的項目。
- `codecommit_infrastructure_source_repo`擷取並建立來源 CodeCommit 儲存庫。
- `codepipeline_iam_role`為管道建立必要的 IAM 角色。
- `codepipeline_kms`為 Amazon S3 物件加密和解密建立必要的 AWS KMS 金鑰。
- `codepipeline_terraform`建立來源 CodeCommit 儲存庫的測試管線。
- `s3_artifacts_bucket`建立 Amazon S3 儲存貯體來管理管道成品。

建置規格檔案

以下是此模式用於運行每個管道階段的構建規範（buildspec）文件的列表：

- buildspec_validate.yml執行“validate”階段。
- buildspec_plan.yml執行“plan”階段。
- buildspec_apply.yml執行“apply”階段。
- buildspec_destroy.yml執行“destroy”階段。

建立規格檔案變數

每個 buildspec 文件都使用以下變量來激活不同的構建特定設置：

變數	預設值	描述
CODE_SRC_DIR	"."	定義源 CodeCommit 目錄
TF_VERSION	「1.3.7」	定義構建環境的地形版本

該buildspec_validate.yml文件還支持以下變量來激活不同的構建特定的設置：

變數	預設值	描述
SCRIPT_DIR	「。 /模板/腳本」	定義腳本目錄
ENVIRONMENT	「開發」	定義環境名稱
SKIPVALIDATIONFAILURE	「Y」	跳過失敗驗證
ENABLE_TFVALIDATE	「Y」	啟動地形驗證
ENABLE_TFFORMAT	「Y」	啟動地形格式
ENABLE_TFCHECKOV	「Y」	啟動切科夫掃描
ENABLE_TFSEC	「Y」	啟動 tfsec 掃描
TFSEC_VERSION	「一二八」	定義了 TFSec 版本

更多模式

- ???
- 將一個 AWS 帳戶中的 AWS CodeCommit 儲存庫與另一個帳戶中的 SageMaker 工作室建立關聯
- 使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目
- 自動執行 Amazon Lookout for Vision 訓練和部署，以進行異常偵測
- 使用 AWS Batch 為 Amazon RDS for PostgreSQL 資料庫執行個體自動備份
- 使用 AWS SAM 自動部署巢狀應用程式
- 使用 CI/CD 管道，在 Amazon EKS 中自動部署節點終止處理程式
- ???
- 使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation
- 自動化跨 AWS 帳戶複寫 Amazon RDS 執行個體
- 使用 CI/CD 管道自動建置 Java 應用程式並將其部署到 Amazon EKS
- 使用 Python 應用程式為亞馬遜動態 B 自動產生模型和 CRUD 函數
- 使用 IAM 存取分析器和 AWS CloudFormation 巨集 CodePipeline，在 AWS 帳戶中自動驗證和部署 IAM 政策和角色
- 在 AWS 雲端上的 Sun SPARC 伺服器備份 Sun 字元 SSP 模擬器
- 使用 AWS DataOps 開發套件建立資料管道以擷取、轉換和分析 Google 分析資料
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建置微焦點企業伺服器 PAC
- 使用 EC2 Image Builder 和 Terraform 為強化的容器映像建立管道
- 使用 Amazon SageMaker 和 Azure 建置 MLOP 工作流程 DevOps
- ???
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 使用 NLog 在 Amazon CloudWatch 日誌中設定 .NET 應用程式的記錄
- 從 AWS 儲存庫持續部署現代 AWS Amplify Web 應用程式 CodeCommit
- 為其建立自訂 Docker 容器映像， SageMaker 並將其用於 AWS Step Functions 中的模型培訓
- 在不支援 AWS 的 AWS 區域建立管道 CodePipeline
- 使用 Amazon CloudWatch 異常偵測為自訂指標建立警示
- 部署管道，同時偵測多個程式碼交付項目中的安全性問題
- 使用基礎設施即程式碼在 AWS 雲端部署和管理無伺服器資料湖
- 使用 Amazon EKS 和 Amazon S3 中的頭盔圖儲存庫來部署 Kubernetes 資源和套件

- 使用 AWS CDK 來部署多堆疊應用程式 TypeScript
- 使用地形表單部署 AWS WAF 解決方案的安全自動化
- 使用 RAG 和提示，開發先進的生成式 AI 聊天助理 ReAct
- ???
- 使用 Amazon Personalize 個人化產生個人化和重新排名的建議
- 當 AWS KMS 金鑰的金鑰狀態變更時，取得 Amazon SNS 通知
- 透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能
- 使用 Kubernetes 在 Amazon EKS 工作者節點上安裝 SSM 代理程式 DaemonSet
- 整合石分支通用控制器與 AWS 大型主機現代化
- 大型主機現代化： DevOps 在具有微焦點的 AWS 上
- 使用 AWS 以程式碼形式管理 AWS IAM 身分中心許可集 CodePipeline
- 使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式
- 將 DNS 記錄批量遷移到 Amazon Route 53 私有託管區域
- SageMaker 使用 AWS 開發人員工具將 ML 建置、訓練和部署工作負載遷移到 Amazon
- 監控跨多個 AWS 帳戶共用 Amazon 機器映像的使用
- 優化 AWS 應用程序容器生成的碼頭映像
- 使用 AWS Step Functions 透過驗證、轉換和分割協調 ETL 管道
- 在多帳戶 VPC 設計中保留非工作負載子網路的可路由 IP 空間
- 使用程式碼儲存庫在 AWS Service Catalog 中佈建 Terraform 產品
- ???
- 輪換資料庫認證而不重新啟動
- 從 AWS Step Functions 同步執行 AWS Systems Manager Automation 任務
- 使用 AWS CDK 和在 Amazon ECS Anywhere 為混合式工作負載設定 CI/CD 管道 GitLab
- 使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施
- 使用 AWS 在 Amazon EC2 上自動設定 UiPath RPA 機器人 CloudFormation
- 在 SaaS 架構中使用 C# 和 AWS CDK 進行筒倉模型的租用戶上線
- 使用地形表單為組織自動啟 GuardDuty 用 Amazon
- 在本機驗證地形表單 (AFT) 程式碼的 Account Factory
- ???

使用者運算

主題

- 使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation
- 更多模式

使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation

由拉姆·坎達斯瓦米（AWS）和阮宗（AWS）創建

環境：生產	技術：終端使用者運算、雲端原生、成本管理 DevOps、SaaS	工作負載：Microsoft
AWS 服務：Amazon AppStream 2.0；AWS CloudFormation		

Summary

此模式提供程式碼範例和步驟，以使用 AWS CloudFormation 範本在亞 Amazon Web Services (AWS) 雲端中自動建立 Amazon AppStream 2.0 資源。該模式向您展示如何使用 AWS CloudFormation 堆疊自動化 AppStream 2.0 應用程式資源的建立，包括映像產生器、映像、叢集執行個體和堆疊。您可以使用桌面平台或應用程式傳遞模式，在符合 HTML5 規範的瀏覽器上將 AppStream 2.0 應用程式串流給使用者。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 接受 AppStream 2.0 條款和條件
- AppStream 資源的基本知識，例如堆疊、叢集和映像建立工具

限制

- 建立執行個體後，您無法修改與 AppStream 2.0 執行個體相關聯的 AWS Identity and Access Management (IAM) 角色。
- 建立映像產生器之後，您就無法在 AppStream 2.0 映像產生器執行個體上修改內容 (例如子網路或安全性群組)。

架構

下圖顯示如何使用 AWS CloudFormation 範本自動建立 AppStream 2.0 資源。

該圖顯示以下工作流程：

1. 您可以根據此 CloudFormation 模式的其他資訊部分中的 YAML 程式碼建立 AWS 範本。
2. AWS CloudFormation 範本會建立 AWS CloudFormation 測試堆疊。
   a. (選擇性) 您可以使用 AppStream 2.0 建立映像產生器執行個體。
   b. (選擇性) 您可以使用自訂軟體建立 Windows 映像檔。
3. AWS CloudFormation 堆疊會建立 AppStream 2.0 叢集執行個體和堆疊。
4. 您可以在符合 HTML5 標準的瀏覽器上將 AppStream 2.0 資源部署給使用者。

技術, 堆

- Amazon AppStream 2.0
- AWS CloudFormation

工具

- Amazon AppStream 2.0 — Amazon AppStream 2.0 是全受管的應用程式串流服務，可讓您從任何地方立即存取桌面應用程式。 AppStream 2.0 管理託管和執行應用程式所需的 AWS 資源、自動擴展，以及視需求為使用者提供存取權限。
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。

史诗

(選擇性) 建立 AppStream 2.0 映像

任務	描述	所需技能
安裝自訂軟體並建立映像檔。	1. 安裝您計劃部署給使用者的 AppStream 2.0 應用程式。 2. 使用光子建立映像代理程式或 PowerShell 指令碼，為您的自訂軟體建立新的 Windows 映像檔。 注意：請考慮使用 Windows AppLocker 功能進一步鎖定映像檔。	AWS DevOps、雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
更新 AWS CloudFormation 範本。	1. 將此模式的「其他資訊」區段中的程式碼儲存為 YAML 檔案。 2. 使用環境中參數所需的值來更新 YAML 檔案。	AWS 系統管理員、雲端管理員、雲端架構師、一般 AWS、AWS 管理員
使用範本建立 AWS CloudFormation 堆疊。	1. 登入 AWS 管理主控台並開啟 AWS 主 CloudFormation 控台。 2. 在導覽窗格中，選擇 [堆疊]。 3. 選擇 Create stack (建立堆疊)，然後選擇 With new	應用程式擁有者、AWS 系統管理員、Windows 工程師

任務	描述	所需技能
	resources (standard) (使用新資源 (標準))。 4. 在 [先決條件-準備範本] 區段中，選擇 [範本已就緒]。 5. 在「指定範本」區段中，選擇「上傳範本檔案」。 6. 選擇 [選擇檔案]，然後選擇更新的 AWS CloudFormation 範本。 7. 完成精靈中的其餘步驟以建立堆疊。	

相關資源

參考

- 開始使用 Amazon AppStream 2.0：使用範例應用程式進行設定
- 建立 AppStream 2.0 叢集與堆疊

教學課程和影片

- Amazon AppStream 2.0 用戶工作流
- 如何將舊版視窗表單應用程式遷移到 Amazon AppStream 2.0
- AWS RE: 發明 2018 年：使用 Amazon AppStream 2.0 安全地交付桌面應用程式 (BAP201)

其他資訊

下列程式碼是 AWS 範 CloudFormation 本的範例，可讓您自動建立 AppStream 2.0 個資源。

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  SubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
  testSecurityGroup:
```

```
    Type: 'AWS::EC2::SecurityGroup::Id'
  ImageName:
    Type: String
Resources:

  AppStreamFleet:
    Type: 'AWS::AppStream::Fleet'
    Properties:
      ComputeCapacity:
        DesiredInstances: 5
      InstanceType: stream.standard.medium
      Name: appstream-test-fleet
      DisconnectTimeoutInSeconds: 1200
      FleetType: ON_DEMAND
      IdleDisconnectTimeoutInSeconds: 1200
      ImageName: !Ref ImageName
      MaxUserDurationInSeconds: 345600
      VpcConfig:
        SecurityGroupIds:
          - !Ref testSecurityGroup
        SubnetIds: !Ref SubnetIds
  AppStreamStack:
    Type: 'AWS::AppStream::Stack'
    Properties:
      Description: AppStream stack for test
      DisplayName: AppStream test Stack
      Name: appstream-test-stack
      StorageConnectors:
        - ConnectorType: HOMEFOLDERS
      UserSettings:
        - Action: CLIPBOARD_COPY_FROM_LOCAL_DEVICE
          Permission: ENABLED
        - Action: CLIPBOARD_COPY_TO_LOCAL_DEVICE
          Permission: ENABLED
        - Action: FILE_DOWNLOAD
          Permission: ENABLED
        - Action: PRINTING_TO_LOCAL_DEVICE
          Permission: ENABLED
  AppStreamFleetAssociation:
    Type: 'AWS::AppStream::StackFleetAssociation'
    Properties:
      FleetName: appstream-test-fleet
      StackName: appstream-test-stack
    DependsOn:
```

```
      - AppStreamFleet
      - AppStreamStack
```

更多模式

- 使用工作階段管理員 Connect 到 Amazon EC2 執行個體
- 改善 Amazon Connect 聯絡中心代理工作站的通話品質
- 從 AWS Step Functions 同步執行 AWS Systems Manager Automation 任務

高效能運算

主題

- 為 AWS 設定 Grafana 監控儀表板 ParallelCluster
- 使用 NICE EnginFrame 和 NICE DCV 工作階段管理員設定 auto 調整規模的虛擬桌面基礎架構 (VDI)

為 AWS 設定 Grafana 監控儀表板 ParallelCluster

由達里奧·拉波塔（AWS）和威廉·盧（AWS）創建

代碼存儲庫：parallelcluster-monitoring-dashboard	環境：PoC 或試點	技術：高效能運算、分析、管理與治理
工作負載：開源	AWS 服務：AWS ParallelCluster	

Summary

AWS 可 ParallelCluster 協助您部署和管理高效能運算 (HPC) 叢集。它支援 AWS Batch 和 Slurm 開放原始碼任務排程器。雖然 AWS ParallelCluster 已與 Amazon 整合以 CloudWatch 進行記錄和指標，但不會為工作負載提供監控儀表板。

適用於 AWS 的 Grafana 儀表板 ParallelCluster (GitHub) 是 AWS 的監控儀表板。 ParallelCluster它在作業系統 (OS) 層級提供工作排程器見解和詳細的監控指標。有關此解決方案中包含的儀表板的詳細資訊，請參閱 GitHub 存放庫中的範例儀表板。這些指標可協助您進一步瞭解 HPC 工作負載及其效能。不過，儀表板程式碼不會針對最新版本的 AWS ParallelCluster 或解決方案中使用的開放原始碼套件更新。此模式可增強解決方案，以提供下列優點：

- 支援 ParallelCluster AWS
- 使用最新版本的開源軟件包，包括 Prometheus，Grafana，Prometheus 思流出口商和 NVIDIA DCGM 出口商
- 增加 Slurm 工作所使用的 CPU 核心和 GPU 數目
- 新增工作監視儀表板
- 針對具有 4 或 8 個圖形處理單元 (GPU) 的節點增強 GPU 節點監控儀表板

此版本的增強型解決方案已在 AWS 客戶的 HPC 生產環境中實作和驗證。

先決條件和限制

先決條件

- 已安裝和設定 AWS ParallelCluster CLI。
- AWS 支援的網路組態 ParallelCluster。此模式使 ParallelCluster 用 AWS 使用兩個子網路組態，這需要公有子網路、私有子網路、網際網路閘道和 NAT 閘道。
- 所有 AWS ParallelCluster 叢集節點都必須能夠存取網際網路。這是必要的，以便安裝腳本可以下載開源軟件和 Docker 映像。
- Amazon Elastic Compute Cloud (Amazon EC2) 中的 key pair。具有此 key pair 的資源具有頭節點的安全殼層 (SSH) 存取權。

限制

- 這種模式旨在支持 Ubuntu 20.04 LTS。如果您使用的是不同版本的 Ubuntu，或者您使用 Amazon Linux 或 CentOS，那麼您需要修改此解決方案提供的腳本。這些修改不包括在此模式中。

產品版本

- Ubuntu 20.04 LTS
- ParallelCluster 3. X

帳單和成本考量

- 以此模式部署的解決方案不在免費方案的涵蓋範圍內。費用適用於 Amazon EC2，亞馬遜 FSx 的盧斯特，亞馬遜 VPC 中的 NAT 網關和 Amazon Route 53。

架構

目標架構

下圖顯示使用者如何在頭節點 ParallelCluster 上存取 AWS 的監控儀表板。頭節點運行 NICE DCV，Prometheus，Grafana，Prometheus 漿流出口商，Prometheus Node Exporter 和 NGINX 開源。運算節點會執行 Prometheus Node Exporter 程式，如果節點包含 GPU，它們也會執行 NVIDIA DCGM 匯出程式。頭節點從計算節點檢索信息，並在 Grafana 儀表板中顯示該數據。

在大多數情況下，由於工作排程器不需要大量的 CPU 或記憶體，因此頭節點的負載不會過重。使用者在連接埠 443 上使用 SSL 存取頭節點上的儀表板。

所有獲得授權的檢視者都可以匿名檢視監控儀表板。只有 Grafana 管理員可以修改儀表板。您可以在檔案中設定 Grafana 管理員的`aws-parallelcluster-monitoring/docker-compose/docker-compose.head.yml`密碼。

工具

AWS 服務

- NICE DCV 是一種高性能遠端顯示協議，可幫助您在不同的網絡條件下從任何雲或數據中心將遠程桌面和應用程序流傳輸到任何設備。
- AWS 可 ParallelCluster協助您部署和管理高效能運算 (HPC) 叢集。它支援 AWS Batch 和 Slurm 開放原始碼任務排程器。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。

其他工具

- Docker 是一組平台即服務 (PaaS) 產品，它們在作業系統層級使用虛擬化，在容器中提供軟體。
- Grafana 是一個開源軟件，可幫助您查詢，可視化，警報和探索指標，日誌和跟踪。
- NGINX 開源是一個開源的 Web 服務器和反向代理。
- NVIDIA 資料中心 GPU 管理器 (DCGM) 是一套工具，可在叢集環境中管理和監控 NVIDIA 資料中心圖形處理單元 (GPU)。在這種模式中，您可以使用 DCGM 出口商，它可以幫助您從 Prometheus 導出 GPU 指標。
- Prometheus 是一個開放原始碼系統監視工具組，可收集並將其指標儲存為時間序列資料，其中包含關聯的索引鍵值配對 (稱為標籤)。在此模式中，您也可以使用 Prometheus Slurm 匯出程式來收集和匯出量度，並使用 Prometheus Node Exporter 匯出程式從運算節點匯出指標。
- Ubuntu 是一個開放原始碼、以 Linux 為基礎的作業系統，專為企業伺服器、桌上型電腦、雲端環境和 IoT 而設計。

代碼存儲庫

此模式的代碼可在 GitHub pcluster-monitoring-dashboard存儲庫中找到。

史诗

建立所需的資源

任務	描述	所需技能
建立 S3 儲存貯體。	建立 Amazon S3 儲存貯體。您可以使用此值區來儲存組態指令碼。如需指示，請參閱 Amazon S3 文件中的建立儲存貯體。	一般 AWS
複製儲存庫。	通過運行以下命令克隆 GitHub pcluster-monitoring-dashboard回購。 `git clone https://github.com/aws-samples/parallelcluster-monitoring-dashboard.git`	DevOps 工程師
建立管理員密碼。	1. 選擇`aws-parallelcluster-monitoring` 文件夾，選擇文`docker-compose` 件夾，然後打開碼頭組成 .head.yml 文件。 2. 在`GF_SECURITY_ADMIN_PASSWORD` 變數中，`Grafana4PC!` 以您選擇的密碼取代。這是您用來管理 Grafana 帳戶的管理密碼。 3. 保存並關閉碼頭構成 .head.yml 文件。	命令介面指令碼

任務	描述	所需技能
將必要的檔案複製到 S3 儲存貯體。	將 post_install.sh 指令碼和aws-parallelcluster-monitoring資料夾複製到您建立的 S3 儲存貯體中。如需指示，請參閱 Amazon S3 文件中的上傳物件。	一般 AWS
為頭節點設定其他安全群組。	1. 建立頭節點的安全性群組。此安全性群組將允許將輸入流量傳送至頭節點上的監視儀表板。如需指示，請參閱 Amazon VPC 文件中的建立安全群組。 2. 將輸入規則新增至安全性群組。如需指示，請參閱 Amazon VPC 文件中的將規則新增至安全群組。對規則使用下列參數： • 類型 — HTTPS • 通訊協定 — TCP • 連接埠範圍 — 443 • 來源 — 輸入您的 IP 位址 • 說明 — 允許使用者存取監控儀表板	AWS 管理員
設定頭節點的 IAM 政策。	為頭節點建立以身分識別為基礎的原則。此政策允許節點從 Amazon 擷取指標資料 CloudWatch。 GitHub 存放庫包含一個示例策略。如需指示，請參閱 AWS Identity and Access Management (IAM) 文件中的建立 IAM 政策。	AWS 管理員

任務	描述	所需技能
設定運算節點的 IAM 政策。	為運算節點建立以身分識別為基礎的原則。此原則可讓節點建立包含工作 ID 和工作擁有者的標籤。 GitHub 存放庫包含一個示例策略。如需指示，請參閱 IAM 文件中的建立 IAM 政策。 如果您使用提供的範例檔案，請取代下列值： • <REGION>— 託管叢集的 AWS 區域 • <ACCOUNT_ID>— AWS 帳戶識別碼	AWS 管理員

建立叢集

任務	描述	所需技能
修改提供的叢集範本檔案。	建立 AWS ParallelCluster 叢集。使用提供的叢集 .yaml AWS CloudFormation 範本檔案作為建立叢集的起點。取代提供的範本中的下列值： • <REGION>— 託管叢集的 AWS 區域。 • <HEADNODE_SUBNET>— VPC 的公用子網路。 • <ADDITIONAL_HEAD_NODE_SG>— 您為頭節點建立的安全群組名稱。	AWS 管理員

任務	描述	所需技能
	• <KEY_NAME>— 輸入現有 Amazon EC2 key pair 的名稱。具有此 key pair 的資源具有頭節點的安全殼層 (SSH) 存取權。 • <ALLOWED_IPS>-輸入 CIDR 格式的 IP 位址範圍，此範圍允許與頭節點建立 SSH 連線。 • <ADDITIONAL_HEAD_NODE_POLICY>— 輸入您為標頭節點建立的 IAM 政策名稱。 • <BUCKET_NAME>— 輸入您建立的 S3 儲存貯體的名稱。 • <COMPUTE_SUBNET>— 輸入 VPC 中私有子網路的名稱。 • <ADDITIONAL_COMPUTE_NODE_POLICY>— 輸入您為運算節點建立的 IAM 政策名稱。	

任務	描述	所需技能
建立 叢集	在 AWS ParallelCluster CLI 中，輸入以下命令。這會部署 CloudFormation 範本並建立叢集。如需有關此命令的詳細資訊，請參閱 AWS 文件中的 pcluster 建立叢集。 ParallelCluster `pcluster create-cluster -n <cluster_name> -c cluster.yaml`	AWS 管理員
監視叢集建立。	輸入下列命令以監視叢集建立。如需有關此命令的詳細資訊，請參閱 AWS 文件中的 pcluster 描述叢集。 ParallelCluster `pcluster describe-cluster -n <cluster_name>`	AWS 管理員

使用 Grafana 儀表板

任務	描述	所需技能
存取 Grafana 入口網站。	1. 輸入下列命令以擷取頭節點的公用 IP 位址。 `pcluster describe-cluster -n <cluster_name> --query headNode.publicIpAddress`	AWS 管理員

任務	描述	所需技能
	2. 在網頁瀏覽器中，瀏覽至下列 URL 以存取 Grafana 儀表板。 HTTPS：//<head_node_public_ip_address> 3. 在 Grafana 首頁上，選擇左側功能表上的 4 方形儀表板圖示，然後選擇 [一般]。這會顯示已設定儀表板的清單。以下儀表板在 Grafana 可用: • 叢集成本 — 包含叢集成本的相關資訊 • 叢集記錄 — 包含叢集記錄的相關資訊 • 計算節點詳細資訊 — 包含運算節點使用量統計資料的相關資訊 • 計算節點清單 — 包含叢集的運算節點清單 • GPU 節點 — 包含 GPU 節點使用率統計資料的相關資訊 • 工作詳細資訊 — 包含工作資源使用率的相關資訊 • 標頭節點詳細資訊 — 包含頭節點使用狀況統計資料的相關資訊 • ParallelCluster 摘要 — 包含叢集使用量的相關資訊	

清理解決方案以避免產生相關成本

任務	描述	所需技能
刪除叢集。	輸入下列指令以刪除叢集。如需有關此命令的詳細資訊，請參閱 AWS 文件中的 pcluster 刪除叢集。 ParallelCluster `pcluster delete-cluster -n <cluster_name>`	AWS 管理員
刪除 IAM 政策。	刪除您為頭節點和計算節點建立的原則。如需刪除政策的詳細資訊，請參閱 IAM 說明文件中的刪除 IAM 政策。	AWS 管理員
刪除安全性群組和規則。	刪除您為頭節點建立的安全性群組。如需詳細資訊，請參閱 Amazon VPC 文件中的刪除安全群組規則和刪除安全群組。	AWS 管理員
刪除 S3 儲存貯體。	刪除您建立用來存放組態指令碼的 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的刪除儲存貯體。	一般 AWS

故障診斷

問題	解決方案
在瀏覽器中無法存取頭節點。	檢查安全性群組，並確認輸入連接埠 443 已開啟。
Grafana 不打開。	在頭節點上，檢查容器日誌`docker logs Grafana`。

問題	解決方案
某些量度沒有資料。	在頭節點上，檢查所有容器的容器日誌。

相關資源

AWS 文件

- 適用於 Amazon EC2 的 IAM 政策

其他 AWS 資源

- AWS ParallelCluster
- AWS 的監控儀表板 ParallelCluster (AWS 部落格文章)

其他資源

- Prometheus 監測系統
- Grafana

使用 NICE EnginFrame 和 NICE DCV 工作階段管理員設定 auto 調整規模的虛擬桌面基礎架構 (VDI)

創建者：達里奧拉波塔和薩爾瓦多·馬卡羅內（AWS）

代碼存儲庫：elastic-vdi-infrastructure	環境：PoC 或試點	技術：高效能運算；基礎架構
AWS 服務：AWS CDK；AWS；Amazon EC2 Auto Scaling CloudFormation；Elastic Load Balancing (ELB)		

Summary

NICE DCV 是一種高性能的遠端顯示協議，可幫助您在不同的網絡條件下將遠程桌面和應用程序從任何雲或數據中心流式傳輸到任何設備。使用 NICE DCV 和 Amazon 彈性運算雲端 (Amazon EC2)，您可以在 EC2 執行個體上遠端執行圖形密集型應用程式，並將其使用者界面串流至更簡單的遠端用戶端機器。這樣就不需要昂貴的專用工作站，並且無需在雲端和用戶端機器之間傳輸大量資料。

這種模式設置了一個功能齊全的 auto 擴展 Linux 和 Windows 虛擬桌面基礎結構（VDI），可通過基於 Web 的用戶界面訪問。VDI 解決方案為研發 (R&D) 使用者提供可存取且高效能的使用者介面，可提交圖形密集型分析要求，並從遠端檢閱結果。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 管理員權限和一組存取金鑰。
- 已安裝和設定的 AWS Cloud Development Kit (AWS CDK) 工具組。如需詳細資訊，請參閱安裝 AWS CDK。
- 為您的 AWS 帳戶安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI。
- Python, 安裝和配置. 如需詳細資訊，請參閱原始碼發行版本 (Python 網站)。

- 提供一或多個虛擬私有雲 (VPC)。
- 可使用兩個或多個彈性 IP 位址。如需有關預設限制的詳細資訊，請參閱彈性 IP 位址限制。
- 對於 Linux EC2 執行個體，請設定安全殼層 (SSH) key pair。如需詳細資訊，請參閱金鑰配對和 Linux 執行個體。

產品版本

- AWS CDK 版本 2.26.0 或更新版本
- Python 版本 3.8 或更新版本

架構

目標架構

下圖顯示此 VDI 解決方案的不同元件。使用者會根據適用於視窗和 Linux NICE EnginFrame DCV 執行個體的 Amazon EC2 自動擴展群組，與 NICE 互動以啟動 Amazon EC2 執行個體。

自動化和規模

此模式隨附的程式碼會建立自訂 VPC、公用和私有子網路、網際網路閘道、NAT 閘道、Application Load Balancer、安全群組和 IAM 政策。AWS CloudFormation 還用於創建 Linux 和視窗漂亮的 DCV 服務器的車隊。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- NICE DCV 是一種高性能遠端顯示協議，可幫助您在不同的網絡條件下從任何雲或數據中心將遠程桌面和應用程序流傳輸到任何設備。在這種模式中，它提供了高頻寬效率的體驗，可遠端串流高效能運算 (HPC) 3D 圖形。
- NICE DCV 工作階段管理員可協助您在 NICE DCV 伺服器叢集中建立和管理 NICE DCV 工作階段的生命週期。

- NICE EnginFrame 是一個先進的前端 Web 界面，用於訪問雲中的技術和科學應用程序。

代碼存儲庫

此模式的代碼可在帶有 NICE EnginFrame 和 NICE DCV 會話管理器存儲庫的自動縮放 VDI 解決方案中找到。

史诗

部署虛擬桌面基礎結構

任務	描述	所需技能
複製儲存庫。	克隆包含代碼的存儲庫。 `git clone https://github.com/aws-samples/elastic-vdi-infrastructure.git`	雲端架構師
安裝所需的 AWS CDK 程式庫。	安裝 AWS CDK 程式庫。 `cd elastic-vdi-infrastructure` `python3 -m venv .venv` `source .venv/bin/activate` `pip3 install -r requirements.txt`	雲端架構師
更新參數。	1. 在您選擇的文字編輯器中開啟 app.py 檔案。 2. 取代下列必要參數的 `CHANGE_ME` 值： • `region`— 目標 AWS 區域。如需完整清單，請參閱 AWS 區域。	雲端架構師

任務	描述	所需技能
	• `account`— 目標 AWS 帳戶的識別碼。如需詳細資訊，請參閱尋找您的 AWS 帳戶 ID。 • `key_name`— 用於存取 Linux EC2 執行個體的 key pair。 3. (選擇性) 修改下列參數的值，為您的環境自訂解決方案： • `ec2_type_enginframe` — EnginFrame 執行個體類型 • `ec2_type_broker` — 工作階段管理員代理實例類型 • `ebs_enginframe_size` — 執行個體的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區的 EnginFrame 大小 • `ebs_broker_size` — 工作階段管理員代理人執行個體的 EBS 磁碟區大小 • `TagName and TagValue`— 資源的計費標籤 • `efadmin_uid` — 管理 EnginFrame 員 (efadmin) 使用者的唯一識別碼 • `linux_shared_storage_size` — 以千兆位元	

任務	描述	所需技能
	組為單位的 OpenZF 大小 (GiB) • `Shared_Storage_Linux` — 共享存儲的掛載點 • `Enginframe_installer` — 下載鏈接 EnginFrame • `Session_Manager_Broker_Installer` — 工作階段管理員代理人的下載連結 4. 儲存並關閉 app.py 檔案。	

任務	描述	所需技能
部署解決方案。	依序執行下列命令。 `cdk bootstrap` `cdk deploy Assets-Stack Parameters-Stack` `cdk deploy Elastic-Vdi-Infrastructure` 部署完成時，會傳回下列兩個輸出： • `Elastic-Vdi-Infrastructure.EnginFrameURL` — 入 EnginFrame 口網站的 HTTPS 位址 • `Elastic-Vdi-InfrastruSecretEFadminPassword` — 包含 ef admin 使用者密碼的秘密的 Amazon 資源名稱 (ARN) 請記下這些值。您稍後在此模式中使用它們。	雲端架構師

任務	描述	所需技能
部署 Linux 伺服器的叢集。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 選擇 [建立堆疊]，然後選擇 [使用新資源]。 3. 在雲格式檔案資料夾中，選取 .yaml 檔案。dcv-linux-fleet 4. 在 [指定堆疊詳細資訊] 頁面上，定義下列參數： • 堆疊名稱 — 堆疊的名稱。 • DcvFleet— NICE DCV DCV 艦隊的名稱。請勿將此值留空或使用空格。 • InstanceType— 叢集的執行個體類型。 • RootVolumeSize— Linux EC2 執行個體的根磁碟區大小。 • MinSize— 應該可用且未執行任何 DCV 工作階段的節點數目下限。例如，如果您輸入2，則解決方案從 2 個節點開始。當使用者建立工作階段時，可用節點的數目會減少到1，而解決方案會建立另一個節點來維持最小值。 • MaxSize— 叢集中節點的最大數目。如果已達到上	雲端架構師

任務	描述	所需技能
	限，使用者將無法啟動新的工作階段。 • BillingTagName— 用於計費的標籤名稱。此標籤名稱必須與用於 Windows 堆疊的名稱不同。 • BillingTagValue— 用於計費的標籤值。 5. 完成堆疊建立精靈，然後選擇 [送出] 開始建立堆疊。	

任務	描述	所需技能
部署視窗伺服器群。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 選擇 [建立堆疊]，然後選擇 [使用新資源]。 3. 在雲格式檔案資料夾中，選取 .yaml 檔案。dcv-windows-fleet 4. 在 [指定堆疊詳細資訊] 頁面上，定義下列參數： • 堆疊名稱 — 堆疊的名稱。 • DcvFleet— NICE DCV DCV 艦隊的名稱。請勿將此值留空或使用空格。 • InstanceType— 叢集的執行個體類型。 • RootVolumeSize— 視窗 EC2 執行個體的根磁碟區大小。 • MinSize— 應該可用且未執行任何 DCV 工作階段的節點數目下限。 • MaxSize— 叢集中節點的最大數目。 • BillingTagName— 用於計費的標籤名稱。此標籤名稱必須與用於 Linux 堆疊的名稱不同。 • BillingTagValue— 用於計費的標籤值。	雲端架構師

任務	描述	所需技能
	5. 完成堆疊建立精靈，然後選擇 [送出] 開始建立堆疊。	

存取已部署的環境

任務	描述	所需技能
擷取管 EnginFrame 理員密碼。	EnginFrame 管理帳戶名為 efadmin，密碼會以密碼的形式存放在 AWS Secrets Manager 員中。密碼的 ARN 是動態產生的，而且會顯示在 AWS CDK 部署的輸出中。 1. 在之前的史詩中，在部署解決方案故事中，在`Elastic-Vdi-Infrastructure.SecretEFadminPassword` 輸出之下，找到生成的秘密的 ARN。 2. 執行下列其中一項動作以擷取密碼： • 使用 Sec rets Manager 主控台。如需詳細資訊，請參閱擷取密碼。 • 輸入 get-secret-value 命令。 `aws secretsmanager get-secret-value \ --secret-id <secret_arn> \ --query SecretString \`	雲端架構師

任務	描述	所需技能
	`--output text`	
存取入 EnginFrame 口網站。	1. 在之前的史詩中，在部署解決方案故事中，在`Elastic-Vdi-Infrastructure.EnginFrameURL` 輸出之下，找到 EnginFrame 入口網站的 HTTPS 位址。 2. 在網頁瀏覽器中，輸入入口網站的 HTTPS 位址。 3. 輸入 efadmin 使用者的認證。	雲端架構師
啟動視窗工作階段。	1. 在入 EnginFrame 口網站的功能表中，選擇 [Windows 桌面]。 2. 當系統提示您以 Windows 系統管理員身分登入時，請輸入與 efadmin 使用者相同的密碼。 3. 確認 Windows 工作階段已成功啟動。	雲端架構師
啟動一個工作階段。	1. 在入 EnginFrame 口網站的功能表中，選擇 Linux 桌面平台。 2. 當系統提示您登入時，請輸入 efadmin 使用者的認證。 3. 確認 Linux 工作階段已成功啟動。	雲端架構師

清除

任務	描述	所需技能
刪除堆疊。	在 AWS 主 CloudFormation 控台中，刪除 Windows 和 Linux 伺服器叢集的堆疊。如需詳細資訊，請參閱刪除堆疊。	雲端架構師
刪除基礎結構。	使用下列 AWS CDK 命令刪除已部署的基礎設施。 `cdk destroy --all`	雲端架構師

故障診斷

問題	解決方案
部署未完成，因為它已中斷。	按照清理史詩中的說明進行操作，然後重複此模式以再次部署環境。

相關資源

- NICE DCV
- 尼斯 EnginFrame

混合式雲端

主題

- 使用混合式連結模式將資料中心擴充功能設定為 VMware Cloud on AWS
- 設定 VMware 自動化以在 VMware Cloud on AWS 佈建虛擬機器
- 在 AWS 上使用 VMware 雲端部署軟體定義的資料中心
- 將 VMware 網路洞察與 VMware Cloud on AWS 整合
- 使用 HCX 作業系統協助移轉，將虛擬機器移轉至 VMware Cloud on AWS
- 使用 VMware 詠嘆調操作的日誌，將日誌從 AWS 雲端傳送到潑濺
- 使用 AWS CDK 和在 Amazon ECS Anywhere 為混合式工作負載設定 CI/CD 管道 GitLab
- 更多模式

使用混合式連結模式將資料中心擴充功能設定為 VMware Cloud on AWS

由迪帕克庫馬爾（AWS）創建

環境：生產	技術：混合雲；基礎架構；移轉	工作負載：所有其他工作
AWS 服務：AWS Direct Connect		

Summary

注意：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。 AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何使用混合式連結模式，透過使用單一 VMware vSphere 用戶端介面來檢視和管理現場部署資料中心和 VMware Cloud on AWS 軟體定義的資料中心 (SDDC) 中的庫存。

透過設定混合式連結模式，您可以將內部部署虛擬機器 (VM) 和應用程式移轉至雲端 SDDC。然後，您的 IT 團隊可以使用熟悉的 VMware 工具來管理雲端資源，而且不需要任何新工具。您也可以使用 VMware 雲端閘道裝置，確保作業一致並簡化管理作業。

此模式提供兩個設定混合式連結模式的選項，但您一次只能使用一個選項。第一個選項會安裝雲端閘道應用裝置，並使用它從內部部署 vCenter Server 連結至雲端軟體定義的資料中心。第二個選項會從雲端 SDDC 設定「混合式連結模式」。

先決條件和限制

先決條件 (兩個選項)

- 現有的內部部署資料中心和雲端 SDDC。
- 現場部署資料中心與雲端軟體定義的資料中心之間的現有連線 (使用 AWS Direct Connect、VPN 或兩者)。

- 內部部署資料中心和雲端 SDDC 會與網路時間通訊協定 (NTP) 或其他授權時間來源同步。
- 內部部署資料中心與雲端 SDDC 之間往返時間的最大延遲時間不超過 100 毫秒。
- 具備內部部署環境存取權的雲端管理員。
- vCenter 伺服器的完整網域名稱 (FQDN) 必須解析為私人 IP 位址。

選項 1 的先決條件

- 內部部署環境應在 vSphere 6.5.0d 或更新版本上執行。
- 雲端閘道設備和 vCenter 伺服器可以透過 AWS Direct Connect、VPN 或兩者進行通訊。
- 雲端閘道裝置符合硬體需求。
- 防火牆連接埠已開啟。

選項 2 的先決條件

- 內部部署 vCenter 伺服器在 vSphere 6.0 更新 3 或更新版本上執行，或在 vSphere 6.5.0d 或更新版本上執行。
- 登入認證可用於內部部署 vSphere 單一登入 (SSO) 網域。
- 內部部署環境中的使用者對基本辨別名稱 (基本 DN) 具有唯讀存取權。
- 內部部署網域名稱系統 (DNS) 伺服器已針對 VMware 管理閘道設定。
- 使用 VMware 連線驗證器實作網路連線測試。
- 防火牆連接埠已開啟。

限制

- 混合式連結模式只能連線一個內部部署 vCenter 伺服器增強型連結模式網域。
- 混合式連結模式僅支援執行 6.7 版或更新版本的內部部署 vCenter Server。

架構

下圖顯示設定混合式連結模式的兩個選項。

使用混合式連結模式移轉不同工作負載

混合式連結模式支援透過使用冷移轉或透過 VMware vSphere vMotion 進行即時移轉，在內部部署資料中心與雲端軟體定義的資料中心之間移轉工作負載。選擇移轉方法時必須考量的因素包括虛擬交換器類型和版本、雲端 SDDC 的連線類型，以及虛擬硬體版本。

冷移轉適用於遭遇停機時間的虛擬機器。您可以關閉虛擬機器、移轉它們，然後重新開啟它們。移轉時間較快，因為不需要複製使用中記憶體。對於接受停機時間的應用程式 (例如，第 3 層應用程式或開發和測試工作負載)，建議您使用冷移轉。如果您的虛擬機器無法停機，您應該考慮針對關鍵任務應用程式使用 vMotion 進行即時移轉。

下圖提供使用「混合式連結模式」之不同工作負載移轉類型的概觀。

工具

- VMware 雲端服務是由 AWS 和 VMware 共同開發的整合式雲端產品。
- VMware 雲端閘道裝置可提供多種內部部署資源連線至雲端資源的混合雲使用案例。
- VMware vSphere 是 VMware 的虛擬化平台，可將資料中心轉型為包含 CPU、儲存和網路資源的彙總運算基礎架構。

史诗

選項 1-搭配雲端閘道裝置使用混合式連結模式

任務	描述	所需技能
設定雲端閘道裝置。	1. 登入 VMware Cloud on AWS 主控台並下載雲端閘道設備。 2. 執行下列兩個步驟，在您的內部部署環境中安裝 Cloud Gateway 裝置： • 選擇 [開始] 以設定並部署雲端閘道裝置。 • 設定混合連結模式。	雲端管理員

任務	描述	所需技能
	如需詳細資訊和詳細步驟，請參閱 VMware 說明文件中的使用 vCenter 雲端閘道應用裝置設定混合式連結模式。	

選項 2-從雲端軟體定義的資料中心使用混合式連結模式

任務	描述	所需技能
從雲端 SDDC 設定混合式連結模式。	1. 登入 VMware Cloud on AWS 主控台，並使用連線驗證器檢查所有必要的網路連線能力。如需此相關資訊，請參閱 VMware 說明文件中的驗證混合式連結模式的網路連線。 2. 登入雲端軟體定義的 vSphere Client，選擇 [功能表]，選擇 [管理]，然後選擇 [網域]。 3. 在「混合雲端」區段中，選擇「連結的網域」，然後連線到您的內部部署 vCenter Server。 4. 將身分識別來源新增至雲端 SDDC 輕量型目錄存取通訊協定 (LDAP) 網域。如需此相關資訊，請參閱 VMware 說明文件中的將身分識別來源新增至軟體定義的資料中心 LDAP 網域。	雲端管理員

相關資源

- 設定混合式連結模式
- 在 AWS 上設定適用於 VMware 雲端的混合式連結模式

設定 VMware 自動化以在 VMware Cloud on AWS 佈建虛擬機器

由迪帕克庫馬爾（AWS）創建

環境：生產	技術：混合雲；基礎架構	工作負載：所有其他工作
AWS 服務：AWS Direct Connect；AWS Site-to-Site VPN		

Summary

注意事項：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

VMware vRealize 自動化是您可以用來要求和管理 IT 資源的自動化軟體。透過選擇使用 VMware Cloud on AWS 設定 vRealize 自動化，您可以在多個資料中心和雲端環境中自動交付虛擬機器 (VM)、應用程式和 IT 服務。

然後，您的 IT 團隊可以建立目錄項目，以設定服務佈建和操作功能，您的使用者可以要求並搭配其現有 vRealize Automation 工具使用這些功能。您也可以整合 AWS 上的 VMware 雲端與 vRealize 自動化雲端組件，以提高 IT 靈活性和效率。

此模式說明如何設定 VMware vRealize 自動化，以便在 AWS 上自動建置虛擬機器或應用程式功能。

先決條件和限制

先決條件

- 現有的現場部署資料中心和 VMware Cloud on AWS 軟體定義資料中心 (SDDC)。如需雲端 SDCC 的詳細資訊，請參閱 VMware 說明文件中的關於軟體定義的資料中心。
- 現場部署資料中心與雲端軟體定義的資料中心之間的現有連線，使用 AWS Direct Connect、VPN (路由或原則型)，或兩者兼而有之。
- 內部部署資料中心和雲端 SDDC 會與網路時間通訊協定 (NTP) 或其他授權時間來源同步。

- 內部部署資料中心與雲端 SDDC 之間往返時間的最大延遲時間不超過 100 毫秒。
- vCenter 伺服器的完整網域名稱 (FQDN) 必須解析為私人 IP 位址。
- 具有內部部署環境存取權的雲端 SDDC 使用者。
- vRealize 自動化雲端組件服務角色中的組織擁有者存取權。
- 具有 vRealize 自動化服務代理人使用服務之權限的使用者。
- 現場部署資料中心的無類別網域間路由 (CIDR) 範圍必須開放，才能從 AWS 主控台上的 VMware Cloud 產生 API 權杖。下列清單提供產生 API 權杖所需的最低角色：
  - 組織成員
  - 組織擁有者
  - 服務角色 VMware Cloud on AWS
  - 管理員
  - NSX 雲端系統管理員
  - NSX Cloud 稽核員

如需相關詳細資訊，請參閱 AWS 合作夥伴網路部落格提供的 VMware Cloud on AWS 軟體定義的軟體定義定義的連線選項。

限制

- 您只能在一個 vRealize 自動化中設定 20 個具有公有端點的 VMware 雲端帳戶。如需此相關資訊，請參閱 VMware 說明文件中的延展性和並行最大值。

產品版本

- vRealize 自動化版本 8.x 或更新版本
- 身分識別管理員 3.x 版或更新版本
- VMware 套件生命週期管理員 8.x 版或更新版本

架構

下圖顯示了 vRealize 自動化服務，這些服務可以同時使用現場部署和 VMware Cloud on AWS 環境的基礎設施。

VMware 雲端組裝元件

VMware 雲端組件是 vRealize 自動化的核心元件，您可以使用它來部署和佈建虛擬機器和計算資源。下表說明必須針對在 AWS 上佈建虛擬機器設定的 VMware 雲端組件元件。

零組件	定義
雲端帳戶	Cloud 帳戶提供連接詳細信息（例如，服務器名稱，用戶名和密碼，訪問密鑰和 API 令牌）。VMware 雲端組件會使用雲端帳戶來收集資源的詳細目錄。
雲端區域	雲端區域可識別雲端帳戶中的資源界限 (例如 AWS 區域和雲端 SDDC)。雲端區域會將計算資源與雲端組件專案建立關聯。
項目	專案是由使用者和資源 (例如雲端區域) 組成的邏輯實體。它也包含建置虛擬機器時所使用的資源配額和虛擬機器命名原則。
風味映射	風格映射提供有關雲模板中使用的虛擬機器容量（例如 CPU 數量和內存量）的信息。
影像映射	映像對應會對應雲端範本中使用的 VMware vSphere 虛擬機器範本和 Amazon Web Services (AWS) 映像。如需此相關資訊，請參閱 VMware 說明文件中的 vRealize 自動化中的進一步了解映像對應。
網路設定檔	網路設定檔控制在 VM 佈建期間選擇網路的放置決策。
儲存設定檔	儲存裝置設定檔可控制在 VM 佈建期間選擇儲存區的放置決策。
雲端範本	VMware 雲端範本是 vRealize 自動化的重要元件，因為它們定義了雲端基礎架構佈建和協調。雲模板是資源的規範，包括資源類型，資源屬性以及要從用戶那裡收集的輸入。

工具

- VMware vRealize 自動化 — vRealize 自動化是具有事件導向狀態管理和合規性的基礎架構自動化平台。它旨在協助組織控制和保護自助式雲端、具有治理的多雲端自動化，以及 DevOps基於基礎架構的交付。
- VMware Cloud on AWS — VMware 雲端是由 AWS 和 VMware 共同開發的整合式雲端產品。

史诗

生成 API 令牌

任務	描述	所需技能
從您的 VMware Cloud on AWS 帳戶產生 API 權杖。	1. 登入 VMware 雲端主控台。 2. 在 VMware 雲端服務工具列上，選擇我的帳戶，然後選擇 API 權杖。 3. 輸入 API 令牌的名稱，提供所需的壽命，並定義令牌的範圍。 4. 選擇「開啟 ID」核取方塊，然後選擇「產生」。 5. 記錄 API 令牌的憑據。 如需相關資訊，請參閱 VMware 說明文件中的如何產生 API 權杖。	雲端管理員

在內部部署資料中心安裝 vRealize 自動化

任務	描述	所需技能
下載所需的軟體。	從我的 VMware 入口網站下載 VMware vRealize 套件 ISO 檔	雲端管理員

任務	描述	所需技能
	案。此套件包含 vRealize 套件生命週期管理員、VMware 身分識別管理員和 vRealize 自動化。	
安裝 軟體。	安裝軟體並連線至您的雲端 SDCC，方法是遵循 VMware 說明文件中的使用簡易安裝程式安裝 vRealize Suite 生命週期管理員和 VMware 身分識別管理員中的指示。 重要：請確定您的安裝可以使用下列項目： • 內部部署 VMware vCenter 伺服器設定和登入認證 • vRealize 自動化 IP 和子網路的網路詳細資料 • vRealize 自動化授權金鑰	雲端管理員、雲端架構師

透過 VMware 雲端組件 Connect AWS 上的 VMware 雲端

任務	描述	所需技能
設定您的雲端帳戶。	1. 在 VMware Cloud 主控台上，開啟 [基礎結構] 索引標籤，選擇 [管理 — 雲端帳戶]，然後選擇 [新增雲端帳戶]。 2. 選擇 VMware Cloud on AWS 作為類型。 3. 粘貼您之前記錄的 API 令牌信息。這會填入 AWS 上	雲端架構師、雲端管理員

任務	描述	所需技能
	VMware 雲端組織中所有可用的雲端軟體定義資料中心。 4. 選擇所需的雲端 SDCC，然後提供軟體定義的資料中心的 vCenter 使用者名稱和密碼。 5. 成功驗證後，您可以檢視整合式 VMware Cloud on AWS 帳戶，其狀態為「正常」。 如需相關資訊，請參閱 VMware 說明文件中的 vRealize 自動化中的建立 AWS 雲端帳戶。	
配置項目。	1. 在 VMware GCP 主控台上，開啟 [專案] 索引標籤，然後選擇 [新增專案]。 2. 輸入您的專案名稱。 3. 開啟「雲端區域」索引標籤，然後選擇預設 VMware Cloud on AWS 雲端帳戶。	雲端管理員

任務	描述	所需技能
設定雲端區域。	1. 在 VMware 雲端主控台上，開啟雲端區域，然後為您的軟體定義的資料中心選擇雲端區域。 2. 依預設，`cloudadmin@vmc.local` (這是雲端軟體定義的 vCenter 的預設本機使用者識別碼) 只有在中佈建的存取權。`Compute-ResourcePool` 3. 打開雲區域下的計算選項卡，然後選擇計算-ResourcePool。	雲端管理員
配置風味映射。	1. 開啟「風味對應」標籤，然後建立新的風味對應。 2. 輸入風格名稱，選擇 VMware Cloud on AWS 帳戶，然後提供 vCPUs 的數量和記憶體容量。	雲端管理員
設定映像對應。	1. 開啟「影像對映」並建立新的影像對應。 2. 輸入影像名稱。 3. 選擇 VMware Cloud on AWS 帳戶，並提供所需的雲端帳戶範本。	雲端管理員

任務	描述	所需技能
設定網路設定檔。	1. 開啟網路設定檔並建立新的網路設定檔。 2. 輸入網路設定檔名稱。 3. 開啟 [網路] 索引標籤，然後選擇要用於佈建的現有網路。	雲端管理員
設定儲存區設定檔。	1. 開啟 [儲存裝置設定檔]，然後選擇 [新增 2. 輸入儲存裝置設定檔的名稱。 3. 在 [原則] 區段中，建立新原則。 4. 選擇工作負載資料存依預設，`cloudadmin@vmc.local` 只有在工作負載資料存放區中進行佈建的存取權。	雲端管理員

任務	描述	所需技能
建立雲端範本。	1. 開啟 [設計] 索引標籤，選擇 [雲端範本]，然後選擇 [從新增] 和 [空白畫布]。 2. 提供雲端範本的名稱和說明。 3. 選擇您之前創建的項目。 4. 從雲模板資源設計頁面中，根據您的需求將組件拖到空白畫布中。 5. 選擇「測試」以測試範本並修正任何問題。 6. 選擇部署並提供部署名稱以部署虛擬機器。 如需相關資訊，請參閱 VMware 說明文件中的建立基本雲端範本。	雲端管理員

相關資源

- 將 vRealize 自動化 8.x 版 Connect 到您的軟體定義的資料中心：
- 從 AWS 主控台上的 VMware 雲端部署軟體定義的資料中心
- AWS 與 VMware Cloud on AWS 直接連接整合

在 AWS 上使用 VMware 雲端部署軟體定義的資料中心

由迪帕克庫馬爾（AWS）和德里克·考克斯（AWS）創建

環境：生產	技術：混合雲；基礎架構	工作負載：所有其他工作
AWS 服務：Amazon VPC		

Summary

注意：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。 AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何建立託管於 Amazon Web Services (AWS) 雲端的 VMware 軟體定義資料中心 (SDDC)。您可以部署軟體定義的資料中心，將您的 VMware vSphere 工作負載遷移到 AWS 雲端，並在使用現有 VMware 工具和技能的同時利用 AWS 服務。您可以使用此軟體定義的資料中心，在 VMware 以 vSphere 為基礎的私有雲、公有雲和混合雲環境中執行生產應用程式，並優化 AWS 服務存取權。例如，您可以使用 SDDC 作為災難復原的次要站台，或將資料中心擴展到不同的地理位置。

VMware Cloud on AWS 是一項 pay-as-you-go (隨需) 服務，可讓各種規模的企業使用各種 AWS 服務，在 VMware 以 vSphere 為基礎的雲端環境中執行工作負載。您可以從每個 SDDC 叢集至少 2 台主機開始，並在生產環境中，每個叢集最多可擴充 16 台主機。如需詳細資訊，請參閱 VMware Cloud on AWS 的網站。若要進一步了解 SDDC，請參閱 VMware 說明文件中的關於軟體定義的資料中心。

先決條件和限制

先決條件

- 註冊 MyVMware 帳戶並填寫所有欄位。
- 註冊 AWS 帳戶。如需指示，請參閱 AWS 知識中心。
- 註冊一個我的雲端 AWS 帳戶。 啟用連結會傳送至您註冊時指定的電子郵件地址。

限制

- 請參閱 VMware 網站上的 VMware 雲端組態限制頁面。

產品版本

- 請參閱 VMware 說明文件中的 VMware 雲端版本注意事項。

架構

目標技術堆疊

下圖顯示在 AWS 裸機專用基礎設施上執行的 VMware 軟體堆疊，包括 vSphere、vCenter、vSAN 和 NSX-T。您可以透過與其他 AWS 服務 (例如亞馬遜彈性運算雲端 (Amazon EC2)、亞馬遜簡單儲存服務 (Amazon S3)、Amazon Redshift、AWS 直接連接、Amazon Relational Database Service 服務 (Amazon RDS) 和亞馬遜 Amazon DynamoDB 等無縫整合，在 AWS 上管理以 VMware 為基礎的資源和工具。

VMware Cloud on AWS 的基本實體是軟體定義的資料中心，其中包含下列元件：

- 運算：運算元件是 AWS 軟體定義的 VMware 雲端軟體定義的最低層。VMware Cloud on AWS Amazon EC2 裸機執行個體類型上執行。其中包括`i3.metal`、和 `i3en.metali4i.metal`，並提供對實體資源 (例如處理器和記憶體) 的直接存取。

  重要事項：VMware Cloud on AWS 的`i3.metal`執行個體類型 (包括一年期和三年期限的隨需和訂閱選項) 將於 2026 年 12 月 31 日結束使用壽命並終止支援。此外，新客戶目前無法要求`i3.metal`執行個體。如需詳細資訊，請參閱 VMware 雲端部落格上的公告。
- 儲存：軟體定義的資料中心叢集使用非揮發性記憶體快閃記憶體 (NVMe) 快閃記憶體儲存裝置，支援 VMware vSAN 的全快閃記憶體組態，提供快速且高效能的儲存裝置。 從軟體定義資料中心 1.20 版開始，VMware Cloud on AWS 提供兩種類型的外部儲存支援：適用於 NetApp ONTAP 的 Amazon FSX 和 VMware 雲端彈性儲存。
- 網路功能：網路功能和原則是使用軟體定義的資料中心叢集中的 VMware NSX-T 來管理。在 SDDC 叢集中建立多層虛擬網路，以將網路資源與實體設備分開。如此一來，VMware 雲端使用者就能建立邏輯的軟體定義網路。

工具

- VMware 雲端服務是由 AWS 和 VMware 共同開發的整合式雲端產品。

史诗

在您的 AWS 帳戶中建立 VPC 和子網路

任務	描述	所需技能
登入 AWS 帳戶。	使用具有管理員許可的登入資料登入 AWS 帳戶。	雲端管理員
建立新 VPC	在此步驟中，您將定義連結至 SDDC 的虛擬私有雲 (VPC)。如果您已有要用於軟體定義資料中心的 VPC，請略過此步驟。 1. 選擇要在 AWS 軟體定義的資料中心部署您的 VMware Cloud on AWS 區域。 2. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。 3. 在導覽窗格中，選擇 Your VPCs (您的 VPC)。 4. 選擇建立 VPC。 5. 指定 VPC 設定，例如 VPC 名稱標籤、IPv4 CIDR 區塊、租用 (保留為預設值)，然後選擇 [建立 VPC]。 6. 建立 VPC 後，選擇 [關閉]。 如需詳細資訊，請參閱 AWS 文件中的建立和設定 VPC。	雲端管理員
建立私有子網路。	您現在將為每個可用區域建立 elastic network interface (ENI) 的私有子網路。建議您使用沒	雲端管理員

任務	描述	所需技能
	有連接網際網路閘道的子網路。 1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。 2. 在導覽窗格中，選擇 Subnets (子網)。 3. 選擇 Create Subnet (建立子網路)。 4. 在 [建立子網路] 頁面上，選擇您先前建立的 VPC。 5. 完成子網路的設定，包括子網路名稱、可用區域和 IPv4 CIDR 區塊。 6. 選擇 Create Subnet (建立子網路)。 重複這些步驟，為區域中的每個可用區域建立子網路。	

在 AWS 上啟用雲端

任務	描述	所需技能
啟動服務。	當您註冊 MyVMware 帳戶時，VMware 會向您傳送歡迎電子郵件和啟用連結至您指定的電子郵件地址。 1. 在瀏覽器中開啟歡迎電子郵件中的「啟用服務」連結。	雲端管理員

任務	描述	所需技能
	2. 使用我的 VMware 認證登入。 3. 檢閱並接受使用服務的條款與條件。 4. 完成帳戶激活過程。系統會將您重新導向至 AWS 主控台上的 VMware 雲端。(注意：VMware Cloud on AWS 帳戶是以組織為基礎，該組織代表訂閱該帳戶的群組或企業單位。 此組織與 AWS Organizations 沒有任何關係。) 5. 在 [選取或建立組織] 頁面上，建立連結至 MyVMware 帳戶的組織。 6. 輸入「組織名稱」與「地址」以進行邏輯區分。 7. 選取「建立組織」以完成處理。 如需有關此程序的詳細資訊，請參閱 AWS 文件中的 AWS 上的軟體定義資料中心部署和最佳實務指南。	

任務	描述	所需技能
指派 IAM 角色。	建立組織後，將特權存取指派給特定使用者，以存取雲端服務和 SDDC 主控台、SDDC 和 NCX 元件。如需指示，請參閱 VMware 說明文件中的指派 VMC 服務角色給組織成員。 組織角色有兩種類型： • 組織擁有者可以新增、移除和修改使用者，以及存取所有雲端資源。 • 組織成員只能存取雲端資源。	雲端管理員

部署軟體定義的資料中心

任務	描述	所需技能
在您的 VMware 雲端 AWS 帳戶中部署軟體定義的資料中心。	重要事項：當 AWS 帳戶與 VMware 組織建立關聯作為記錄賣家後，就無法更新 AWS 帳戶編號。每個 VMware 組織只能有一位 AWS 銷售商的記錄。 若要部署軟體定義的資料中心： 1. 登入 VMC 主控台，網址為 https://vmc.vmware.com。 2. 從可用的服務中選擇 VMware Cloud on AWS 服務。 3. 選擇 [建立 SDDC]。	雲端管理員、雲端架構師

任務	描述	所需技能
	4. 輸入 SDDC 屬性，例如 AWS 區域、部署 (單一主機、多主機或延伸叢集)、主機類型、SDDC 名稱、主機數目、主機容量和總容量，然後選擇下一步。 5. Connect 到您的 AWS 帳戶，然後選擇 [下一步]。 6. 選取先前設定的 VPC 和子網路，然後選擇 [下一步]。 7. 輸入軟體定義的資料中心的管理子網路 CIDR 區塊，然後選擇 [下一步]。如需詳細資訊，請參閱 VMware 雲端部落格上的為軟體定義的資料中心選取 IP 子網路和連線。 8. 選取這兩個核取方塊，以確認您負責建置 SDDC 的成本，然後選擇「部署 SDDC」。 當您選擇部署軟體定義的資料中心時，我們會向您收費。您將無法暫停或取消部署程序，這需要一些時間才能完成。 如需有關建立軟體定義的資訊，請參閱 VMware 說明文件中的從 VMC 主控台部署軟體定義的資料中心。	

相關資源

- 部署和管理軟體定義的資料中心 (VMware 說明文件)
- VMware Cloud on AWS 的功能 (AWS 網站)
- 使用 VMware 雲端在 AWS 上加速雲端移轉和現代化 (影片)

將 VMware 網路洞察與 VMware Cloud on AWS 整合

由迪帕克庫馬爾（AWS），皮特拉（AWS）和薩欽特里韋迪（AWS）創建

環境：PoC 或試點	資料來源：VMware 網絡洞察	目標：VMware Cloud on AWS
R 類型：搬遷	工作負載：所有其他工作	技術：混合雲；基礎架構；移轉
AWS 服務：VMware Cloud on AWS		

Summary

注意：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。 AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何在虛擬機器上整合 VMware vRealize 網路洞察與 VMware 雲端，以 AWS 及如何檢查來自虛擬機器的流量。此整合也可協助您規劃將應用程式移轉至 VMware 雲端 AWS。

vRealize 網路洞察可讓您掌握網路基礎架構。它提供網路監控和分析功能，以改善安全性、降低移轉風險並最佳化效能。您可以使用此工具監視虛擬機器的流量流量，並根據觀察到的流量檢視建議的安全規則。如需 vRealize 網路鑑識的相關資訊，請參閱 VMware 說明文件。

VMware Cloud on AWS 是一項 pay-as-you-go (隨選) 服務，可讓各種規模的企業使用各式各樣的雲端環境，在 VMware 以 vSphere 為基礎的雲端環境中執行工作負載。 AWS 服務您可以從每個 SDDC 叢集至少 2 台主機開始，並在生產環境中，每個叢集最多可擴充 16 台主機。如需詳細資訊，請參閱 VMware 雲端 AWS網站。若要進一步了解 SDDC，請參閱 VMware 說明文件中的關於軟體定義的資料中心。

先決條件和限制

先決條件

- VMware 雲端在 AWS 軟體定義的資料中心，已部署

限制

- 如需已知限制，請參閱 VMware 說明文件。

產品版本

- 網路洞察版本 5.0.0
- VMware Cloud on AWS 軟體定義的資料中心 1.24 版

架構

源, 技術, 堆棧

- 網路洞察

目標技術堆疊

- 在 VMware 雲端 AWS

目標架構

下圖顯示內部部署 VMware 雲端 AWS 與 vRealize 網路洞察之間的連線能力。

工具

- VMware Cloud on AWS 是由 AWS VMware 與 VMware 共同開發的整合式雲端服務。
- VMware vRealize 網路洞察是一種監控與分析工具，可提供網路基礎架構的可見度，以進行安全性規劃和疑難排解。

史诗

為 vRealize 網路洞察設定您的環境

任務	描述	所需技能
建立 VMware 使用者帳戶。	建立 VMware 使用者帳戶或登入您現有的 VMware 帳戶。 若要開立新帳戶： 1. 填寫註冊表單即可註冊 VMware 客戶 Connect 帳戶。 新用戶將收到一封電子郵件以激活其帳戶。 2. 輸入電子郵件中的驗證碼。 3. 登入「客戶 Connect」。	雲端管理員
下載 vRealize 網路鑑識的 OVA 檔案。	下載 vRealize 網路鑑識的 OVA 檔案:. 1. 瀏覽至 VMware 產品下載頁面，網址為 https://my.vmware.com/group/vmware/home。 2. 搜尋 vRealize 網路洞察。 3. 下載最新的 vRealize 網路洞察 5.0.0 版平台和收集器 OVA 檔案。	雲端管理員
部署網路洞察。	如需部署指示，請參閱 VMware 說明文件。	雲端管理員

新增資料來源和收集器

任務	描述	所需技能
新增資料來源。	1. 登入 vRealize 網路洞察。 2. 選擇「設置」，「帳戶和數據源」，「添加源」。 3. 對於「類型」，請選擇內部部署 vCenter 伺服器。 如需詳細資訊，請參閱 VMware 說明文件。	雲端管理員
設定資料來源的收集器。	如需相關指示，請參閱 VMware 說明文件。	雲端管理員

分析應用相依性

任務	描述	所需技能
建立 應用程式。	如果 vRealize 網路鑑識中沒有現有的應用程式，請依照 VMware 說明文件中的步驟建立應用程式。	雲端管理員
探索並分析您的應用程式。	1. 使用 vRealize 網路洞察探索您的應用程式。如需相關指示，請參閱 VMware 說明文件。 2. 分析您的應用程式。如需相關指示，請參閱 VMware 說明文件。	雲端管理員

相關資源

- 使用 VMware 雲端 AWS(AWS 規範性指引) 在 AWS 上部署 VMware 軟體定義的資料中心
- AWS 使用混合式連結模式設定 VMware Cloud 的資料中心延伸模組 (AWS 規範指引)
- 將 VMware 軟體定義的軟體定義資料中心移轉至 VMware 雲端 AWS 使用 VMware 硬體驗 (AWS 規範性指引)
- VMware 網路洞察說明文件 (VMware 網站)

使用 HCX 作業系統協助移轉，將虛擬機器移轉至 VMware Cloud on AWS

由迪帕克庫馬爾（AWS）和希曼州古普塔（AWS）創建

環境：PoC 或試點	資料來源：非 vSphere 環境	目標：VMware 雲端在 AWS 軟體定義的資料中心
R 類型：搬遷	工作負載：所有其他工作	技術：混合雲；移轉

Summary

注意事項：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何使用作業系統協助移轉 (OSAM)，將虛擬機器 (VM) 從非 vSphere 環境移轉至 VMware 亞馬遜雲端網路服務 (AWS)。

OSAM 屬於 VMware 混合雲延伸功能 (HCX) 的一部分，該延伸功能包含 VMware Cloud on AWS 中。您可以使用 OSAM 將非 vSphere 環境 (例如 VMware KVM 或 Hyper-V) 移轉至 VMware Cloud on AWS。OSAM 使用安裝在 Windows 或 Linux 來賓虛擬機器上的 Sentinel 軟體，以協助將虛擬機器從現場部署環境複製到 VMware Cloud on AWS 的軟體定義資料中心 (SDDC)。

此模式說明如何啟用 OSAM、在 Windows 虛擬機器上安裝 Sentinel 軟體、在來源站台上連接 HCX 哨兵閘道 (SGW) 設備並註冊，以及在目的地站點與 HCX 哨兵資料接收器 (SDR) 設備建立轉送連線以啟動移轉。

如需有關 OSAM 的詳細資訊，請參閱 VMware 說明文件。

先決條件和限制

先決條件

- 在來源和目標環境中安裝 HCX。如需 HCX 先決條件，請參閱 AWS Prescriptive Guidance 文件中的使用 VMware HCX 將 VMware 軟體定義的資料定義中心移轉至 AWS 上的 VMware 雲端。

- 如需 OSAM 先決條件，請參閱 VMware 說明文件中的安裝檢查清單。
- 如需 OSAM 連接埠的資訊，請參閱 VMware 連接埠和通訊協定網站上的 VMware HCX 連接埠需求。

限制

- VMware HCX 4.2.0 組態限制
- OSAM 部署的考量
- 支援的客體作業系統
- 客體作業系統考量

產品版本

- VMware 恒生國際校驗 4.2.0
- VMware 軟體定義的資料中心 1.12

架構

下圖顯示 HCX OSAM 如何與 Sentinel 軟體搭配使用，將非 vSphere 虛擬機器從現場部署環境複寫到 VMware Cloud on AWS。

OSAM 由三個元件組成：

- Sentinel 閘道 (SGW) 應用裝置，用於在來源 VMware 型環境中連接和轉送工作負載和應用程式
- Sentinel 資料接收器 (SDR)，用於目的地 VMware 雲端對 AWS 環境，用於從來源接收移轉的工作負載
- Sentinel 軟體，必須安裝在您要移轉的每個客體虛擬機器上

OSAM 使用安裝在 Windows 或 Linux 客體虛擬機器上的定點軟體，協助將虛擬機器從內部部署複製到 VMware 軟體定義的資料中心。您在客體虛擬機器上安裝的 Sentinel 軟體會從客體虛擬機器收集系統組態，並協助進行資料複寫。此資訊也可用來建立用於移轉的客體虛擬機器詳細目錄，並協助準備複本虛擬機器上的磁碟以供複寫和移轉之用。

工具

- VMware 恒生國際校驗 4.2.0
- VMware 雲端在 AWS 軟體定義的資料中心

史诗

配置 HCX

任務	描述	所需技能
部署 HCX 雲端和 HCX 連接器。	請依照 VMware 說明文件中的 HCX 連接器和 HCX 雲端安裝中的指示進行。	雲端管理員、系統管理員

設定 OSAM 並移轉虛擬機器

任務	描述	所需技能
安裝 HCX 哨兵。	若要在 Linux 上安裝哨兵： 1. 在用於 HCX 連接器的 vCenter 伺服器中，選擇互連、多站台服務網格、哨兵管理。 2. 選擇下載 Linux 套裝軟體。 3. 在 Linux 機器上安裝定點代理程式。 如需詳細資訊，請參閱 VMware 說明文件中的下載和安裝 HCX Sentinel 代理程式軟體。	雲端管理員

任務	描述	所需技能
移轉虛擬機器。	若要在群組中移轉 VM (稱為行動群組)，請依照下列步驟執行： 1. 在 vSphere 用戶端中，從 HCX 外掛程式中，選擇服務，移轉。 2. 選擇 Migrate (遷移)。 3. 選擇非 vSphere 詳細目錄，遠端連線。這會顯示您在其上安裝 HCX 哨兵的虛擬機器清單。 4. 在群組名稱中，輸入您要為虛擬機器建立的行動群組名稱。 5. 選擇您要移轉的虛擬機器，然後選擇 [新增] 將它們新增至行動群組。 6. 針對每個虛擬機器： a. 選取目的地運算容器。 b. 選取目的地儲存裝置。 c. 選取移轉設定檔。 d. 選取目標資料夾。 7. 若要啟動移轉程序，請選擇 [執行]。 HCX 會在移轉開始前驗證您的虛擬機器選擇。 如需詳細資訊，請參閱 VMware 說明文件中的移轉具有行動群組的虛擬機器和監	雲端管理員

任務	描述	所需技能
	視和預估使用行動群組進行移轉。	

相關資源

VMware 說明文件：

- VMware HCX 使用者指南
- 安裝檢查清單 B-使用 VMC 軟體定義的資料中心目的地環境的 HCX
- 在 AWS 上的 VMware 雲端中使用 VMware 環保總體驗
- 適用於 VMware Cloud on AWS 的 HCX 作業系統協助移轉
- VMware 恒生校驗 4.2.1 版本資訊

使用 VMware 詠嘆調操作的日誌，將日誌從 AWS 雲端傳送到潑濺

由迪帕克庫馬爾（AWS）和皮特拉皮特拉（AWS）創建

環境：生產	資料來源：VMware Cloud on AWS 的日誌和事件	目標：Splunk 內部部署端點
R 類型：搬遷	工作負載：所有其他工作	技術：混合雲；基礎架構；移轉
AWS 服務：VMware Cloud on AWS		

Summary

注意事項：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此病毒碼說明如何使用 VMware 核取記錄作業，將 AWS 事件或記錄檔上的 VMware 雲端轉寄至系統記錄檔或 HTTP 端點 (例如 Splunk)。

VMware Aria 日誌操作是一種日誌分析工具，可在 VMware 雲端 AWS 環境中提供增強的可見性和加速故障排除。您可以將此工具設定為將 VMware Cloud 中的全部或部分記錄檔或事件傳送 AWS 至系統日誌或 HTTP 端點。端點可以是軟體即服務 (SaaS) 端點，也可以是內部部署端點，例如 Splunk。（此模式提供了 Splunk 的說明。）若要進一步了解適用於記錄的 VMware 詠嘆調作業，請參閱 VMware 說明文件。

VMware Cloud on AWS 是一項 pay-as-you-go (隨選) 服務，可讓各種規模的企業使用各式各樣的雲端環境，在 VMware 以 vSphere 為基礎的雲端環境中執行工作負載。 AWS 服務您可以從每個軟體定義的資料中心 (SDDC) 叢集至少 2 台主機開始，在生產環境中，每個叢集最多可擴充 16 台主機。如需詳細資訊，請參閱 VMware 雲端 AWS網站。若要進一步了解 SDDC，請參閱 VMware 說明文件中的關於軟體定義的資料中心。

先決條件和限制

前提

- 潑濺，在內部部署上設定

限制

您可以註冊 VMware Aria 操作記錄的免費試用訂閱。此訂閱的有效期為 30 天，並具有以下限制：

- 可轉寄的記錄檔大小上限：每天 50 GB 記錄
- 您可以建立的記錄檔轉送設定數目上限：10
- 您可以啟動的記錄轉送設定數目上限：5

要訪問所有服務功能，您必須升級到高級訂閱。

如需有關試用和進階訂閱的詳細資訊，請參閱 VMware 說明文件中的 VMware Aria 日誌作業 (SaaS) 訂閱和計費。如需有關使用限制的詳細資訊，請參閱 VMware 說明文件中的功能使用限制。

產品版本

- 軟 AWS 體定義資料中心上的 VMware 雲端 1.24 版
- 適用於記錄的 VMware 詠嘆調作業 8.10 版
- 內部部署潑濺版 9.x

架構

源, 技術, 堆棧

- 在 VMware 雲端 AWS
- VMware Aria Operations for Logs

目標技術堆疊

- 內部部署溢出

目標架構

下圖顯示企業資料中心與 VMware 雲端中記錄的 VMware Aria 作業之間的連線能力 AWS。

工具

- VMware 雲上 AWS是與 VMware 共同開發的整合式雲端產品。 AWS
- VMware 的詠嘆調操作日誌是 VMware 雲上的日誌分析和故障排除工具 AWS。

史诗

部署軟體定義的資料中心並針對記錄啟用 VMware 詠嘆調作業

任務	描述	所需技能
在軟 AWS 體定義的資料中心上部署 VMware 雲端。	依照 AWS 規範指引中的使用 VMware 雲端部署 VMware 軟體定義 AWS 的資料中心 AWS 中的指示進行。	雲端架構師、雲端管理員
註冊適用於記錄的 VMware 詠嘆調作業。	如需相關指示，請參閱 VMware 說明文件。	雲端架構師

部署雲端代理

任務	描述	所需技能
部署雲端代理伺服器。	若要將記錄檔轉寄至 Splunk 的內部部署執行個體，您必須為記錄的 VMware Aria 作業新增雲端代理。此 Proxy 會從內部部署資料中心接收資訊，並將其傳送至 VMware Aria 作業進行記錄以進行分析。 如果要下載並安裝雲端代理伺服器： 1. 請確定連接埠 443、22 和 514 在您的內部部署環境與	雲端管理員、雲端架構師

任務	描述	所需技能
	VMware 雲端之間已開啟。AWS對於其他連接埠，您可以使用 1514/TCP 或 6514/TCP。如需有關連接埠的詳細資訊，請參閱 VMware 說明文件中的 VMware Aria 作業記錄防火牆建議。 2. 登入 VMware 詠嘆調作業以進行記錄。 3. 在首頁上，選擇 Widget 中的「新增收集器」。 4. 在雲端 Proxy 虛擬應用裝置畫面上，複製權杖金鑰。您必須在 24 小時內使用此金鑰才能完成下列步驟。 5. 選擇 OVA 檔案的下載連結。 6. 導覽至 VMware vSphere 網路用戶端，選擇您的叢集，然後選取部署 OVF 範本。 7. 當系統提示您輸入金鑰時，請貼上您在步驟 4 中複製的權杖金鑰。 8. 選擇 [完成] 以安裝雲端代理伺服器。	

將記錄檔轉寄至內部部署 Splunk 端點

任務	描述	所需技能
設定記錄檔轉送。	如果要將記錄檔轉寄至 Splunk 端點：	

任務	描述	所需技能
	1. 登入 VMware 詠嘆調作業以進行記錄。 2. 瀏覽至「記錄檔管理」。 3. 選擇記錄檔轉送。 4. 選擇「新增組態」，然後完成下列設定： • 提供記錄檔轉送組態的名稱。 • 針對「目的地」，選擇「內部部署 • 針對雲端代理伺服器，請選取您先前安裝的雲端代理伺服器。 • 針對「端點類型」，選擇「TCP」。 • 對於端點 URL，請以下列格式提供您的內部部署 Splunk URL： `tcp://x.x.x.x (your Splunk IP address): 514` • (選擇性) 對於標籤，您可以指定標籤名稱和值以便於查詢。 • 選擇「套用至所有記錄」或「套用至特定記錄檔」。如果您想要將 AWS 上的所有 VMware 雲端記錄檔傳送至 Splunk，請選擇 [套用至所有記錄]。 5. 選擇 Verify (驗證)。	

任務	描述	所需技能
	6. 選擇儲存。 如需詳細資訊，請參閱 VMware 說明文件中的從 VMware Aria 作業轉寄記錄檔。	

相關資源

- AWS 網站上的 VMware 雲端服務
- 關於軟體定義的資料中心 (VMware 說明文件)
- 使用 VMware 雲端 AWS(AWS 規範性指 AWS 引) 在上部署 VMware 軟體定義的資料中心
- 使用 VMware HCX (AWS 規範性指引) 將工作負載移轉至 VMware 雲端 AWS
- AWS 使用混合式連結模式設定 VMware Cloud 的資料中心延伸模組 (AWS 規範指引)

使用 AWS CDK 和在 Amazon ECS Anywhere 為混合式工作負載設定 CI/CD 管道 GitLab

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）創建

代碼庫： amazon-ecs-anywhere-cicd-pipeline-cdk-sample	環境：PoC 或試點	技術：混合雲；容器與微服務；基礎架構； DevOps
工作負載：開源	AWS 服務：AWS CDK；AWS CodePipeline；Amazon ECS；AWS Systems Manager；AWS CodeCommit	

Summary

Amazon ECS Anywhere 是 Amazon Elastic Container Service (Amazon ECS) 的擴展。它支援將外部執行個體 (例如現場部署伺服器或虛擬機器 (VM)) 註冊到 Amazon ECS 叢集。is 功能有助於降低成本並減輕複雜的本機容器協調和操作。您可以使用 ECS Anywhere 在內部部署和雲端環境中部署和執行容器應用程式。您的團隊不需要學習多個領域和技能，或者自行管理複雜的軟體。

此模式描述使用亞馬遜網路服務 (AWS) Cloud Development Kit (AWS CDK) 堆疊，透過 Amazon ECS 任何地 step-by-step 方執行個體佈建 Amazon ECS 叢集的方法。然後，您可以使 CodePipeline 用 AWS 設定持續整合和持續部署 (CI/CD) 管道。然後，您將程式 GitLab 碼儲存庫複寫到 AWS，CodeCommit 並在 Amazon ECS 叢集上部署容器化應用程式。

此模式旨在協助那些使用內部部署基礎結構執行容器應用程式並用 GitLab 來管理應用程式程式碼庫的使用者。您可以使用 AWS 雲端服務來管理這些工作負載，而不會干擾現有的現場部署基礎設施。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在內部部署基礎結構上執行的容器應用
- 您可以在其中管理應用程式程式碼 GitLab 庫的儲存庫。如需詳細資訊，請參閱存放庫 (GitLab)。

- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI (AWS CLI 文件)。
- AWS CDK 工具組，可在全球範圍內安裝和設定。如需詳細資訊，請參閱安裝 AWS CDK (AWS CDK 文件)。
- npm，安裝和配置的 AWS CDK 在 TypeScript. 如需詳細資訊，請參閱下載和安裝 Node.js 和 npm (npm 文件)。

限制

- 如需限制和考量事項，請參閱 Amazon ECS 文件中的外部執行個體 (Amazon ECS 無處不在)。

產品版本

- AWS CDK 工具組版本 2.27.0 或更新版本
- 故宮版本 7.20.3 或更新版本
- Node.js 版本 16.6.1 或更新版本

架構

目標技術堆疊

- AWS CDK
- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon ECS Anywhere
- Amazon Elastic Container Registry (Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS 系統管理員
- GitLab 儲存庫

目標架構

此圖表示此模式中描述的兩個主要工作流程：佈建 Amazon ECS 叢集，以及設定用於設定和部署 CI/CD 管道的 CI/CD 管道，如下所示：

1. 佈建 Amazon ECS 叢集
    a. 當您部署第一個 AWS CDK 堆疊時，它會在 AWS 上建立 CloudFormation 堆疊。
    b. 此 CloudFormation 堆疊佈建了一個 Amazon ECS 叢集和相關的 AWS 資源。
    c. 若要向 Amazon ECS 叢集註冊外部執行個體，您必須在虛擬機器上安裝 AWS Systems Manager 代理程式 (SSM 代理程式)，並將該虛擬機器註冊為 AWS Systems Manager 受管執行個體。
    d. 您也必須在虛擬機器上安裝 Amazon ECS 容器代理程式和 Docker，才能將其註冊為 Amazon ECS 叢集的外部執行個體。
    e. 使用 Amazon ECS 叢集註冊和設定外部執行個體後，它可以在已註冊為外部執行個體的 VM 上執行多個容器。
    f. Amazon ECS 叢集處於作用中狀態，可透過容器執行應用程式工作負載。Amazon ECS Anywhere 不在容器執行個體在現場部署環境中執行，但與雲端中的 Amazon ECS 叢集相關聯。
2. 設定和部署 CI/CD 管線
    a. 當您部署第二個 AWS CDK 堆疊時，它會在 AWS 上建立另一個 CloudFormation 堆疊。
    b. 此 CloudFormation 堆疊在 CodePipeline 和相關 AWS 資源中佈建管道。
    c. 您將應用程式程式碼變更推送並合併到內部部署 GitLab 存放庫。
    d. GitLab 儲存庫會自動複製到 CodeCommit 儲存庫。
    e. CodeCommit 存放庫的更新會自動啟動 CodePipeline。
    f. CodePipeline 從中 CodeBuild複製代碼， CodeCommit 並在中創建可部署的應用程序構建。
    g. CodePipeline 建立 CodeBuild 建置環境的 Docker 映像檔，並將其推送至 Amazon ECR 存放庫。
    h. CodePipeline 啟 CodeDeploy 動從 Amazon ECR 存放庫提取容器映像的動作。
    i. CodePipeline 在 Amazon ECS 叢集上部署容器映像。

自動化和規模

此模式使用 AWS CDK 做為基礎設施即程式碼 (IaC) 工具來設定和部署此架構。AWS CDK 可協助您協調 AWS 資源，並在任何地方和 CI/CD 管道設定 Amazon ECS。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon Elastic Container Registry (Amazon ECR) 是一種安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。此模式也使用 Amazon ECS Anywhere 不在，提供將現場部署伺服器或虛擬機器註冊到 Amazon ECS 叢集的支援。

其他工具

- Node.js 是一個事件驅動的 JavaScript 運行時環境，旨在構建可擴展的網絡應用程序。
- npm 是一個在 Node.js 環境中運行的軟件註冊表，用於共享或借用軟件包以及管理私有軟件包的部署。
- Vagrant 是用於構建和維護便攜式虛擬軟件開發環境的開源實用程序。為了演示目的，此模式使用 Vagrant 創建一個內部部署虛擬機。

代碼存儲庫

此模式的程式碼可在使用 AWS CDK 儲存庫的 Amazon ECS Anywhere 不在的 GitHub CI/CD 管道中取得。

最佳實務

部署此模式時，請考慮下列最佳作法：

- 使用 AWS CDK 開發和部署雲端基礎設施的最佳實務

- 使用 AWS CDK 開發雲端應用程式的最佳實務 (AWS 部落格文章)

史诗

驗證 AWS CDK 組態

任務	描述	所需技能
驗證 AWS CDK 版本。	輸入下列命令，以驗證 AWS CDK 工具組的版本。 `cdk --version` 此病毒碼需要 2.27.0 版或更新版本。如果您使用的是舊版，請按照 AWS CDK 文件中的指示進行更新。	DevOps 工程師
驗證 NPM 版本。	通過輸入以下命令驗證 NPM 的版本。 `npm --version` 此病毒碼需要 7.20.3 版或更新版本。如果您有較早的版本，請按照 npm 文檔中的說明進行更新。	DevOps 工程師
設定 AWS 登入資料。	輸入`aws configure` 命令並按照提示設定 AWS 登入資料。 `$aws configure` `AWS Access Key ID [None]: <your-access-key-ID>`	DevOps 工程師

任務	描述	所需技能
	`AWS Secret Access Key [None]: <your-secret-access-key>` `Default region name [None]: <your-Region-name>` `Default output format [None]:`	

啟動 AWS CDK 環境

任務	描述	所需技能
複製 AWS CDK 程式碼儲存庫。	1. 輸入下列命令，使用 AWS CDK 儲存庫為此模式複製適用於 Amazon ECS Anywhere 的 CI/CD 管道。 `git clone https://github.com/aws-samples/amazon-ecs-anywhere-cicd-pipeline-cdk-sample.git` 2. 輸入下列命令，導覽至複製的目錄。 `cd amazon-ecs-anywhere-cicd-pipeline-cdk-sample`	DevOps 工程師
引導環境。	輸入以下命令，將 CloudFormation 範本部署到您要使用的帳戶和 AWS 區域。	DevOps 工程師

任務	描述	所需技能
	`cdk bootstrap <account-number>/<Region>` 如需詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。	

為任何地方的 Amazon ECS 建置和部署基礎設施

任務	描述	所需技能
安裝軟件包依賴關係並編譯 TypeScript 文件。	安裝套件相依性，並輸入下列指令來編譯 TypeScript 檔案。 `$cd EcsAnywhereCdk` `$npm install` `$npm fund` 這些指令會安裝範例存放庫中的所有套件。如需詳細資訊，請參閱 npm 文件中的 npm ci 和 npm 安裝。如果您在輸入這些命令時收到遺失套件的任何錯誤，請參閱此模式的疑難排解一節。	DevOps 工程師
建置專案。	若要建立專案程式碼，請輸入下列命令。 `npm run build` 如需有關建置和部署專案的詳細資訊，請參閱 AWS CDK 文件中的第一個 AWS CDK 應用程式。	DevOps 工程師

任務	描述	所需技能
部署 Amazon ECS Anywhere 不在的基礎設施堆疊。	1. 輸入以下命令列出堆疊。 `$cdk list` 2. 確認輸出會傳回`EcsAnywhereInfraStack` 和`ECSAnywherePipelineStack` 堆疊。 3. 輸入以下命令來部署`EcsAnywhereInfraStack` 堆疊。 `$cdk deploy EcsAnywhereInfraStack`	DevOps 工程師
驗證堆棧的創建和輸出。	1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/cloudformation/ 開啟 CloudFormation 主控台。 2. 在「堆疊」頁面上，選取`EcsAnywhereInfraStack` 堆疊。 3. 確認堆疊狀態為`CREATE_IN_PROGRESS` 或`CREATE_COMPLETE` 。 設定 Amazon ECS 叢集可能需要一些時間。在堆疊建立完成之前，請勿繼續進行。	DevOps 工程師

設定內部部署 VM

任務	描述	所需技能
設定您的虛擬機器。	通過從 Vagrantfile 所在的根目錄中輸入`vagrant up`命令來創建一個流浪虛擬機。有關更多信息，請參閱流浪文檔。	DevOps 工程師
將您的 VM 註冊為外部執行個體。	1. 使用`vagrant ssh`命令登錄到流浪虛擬機。有關更多信息，請參閱流浪文檔。 2. 按照 AWS CLI 安裝說明並輸入以下命令，在虛擬機器上安裝 AWS CLI。 `$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" \ > -o "awscliv2.zip" $sudo apt install unzip $unzip awscliv2.zip $sudo ./aws/install $aws configure AWS Access Key ID [None]: <your-access-key-ID> AWS Secret Access Key [None]: <your-secret-access-key> Default region name [None]: <your-Region-name> Default output format [None]:`	DevOps 工程師

任務	描述	所需技能
	1. 建立啟動碼和 ID，您可以用來向 AWS Systems Manager 註冊您的虛擬機器，以及啟用外部執行個體。此命令的輸出包括啟動 ID 和啟動碼值。 `aws ssm create-activation \ > --iam-role EcsAnywhereInstanceRole \ > | tee ssm-activation.json` 如果您在執行此命令時收到錯誤訊息，請參閱疑難排解一節。 2. 匯出啟動 ID 和代碼值。 `export ACTIVATION_ID=<activation-ID> export ACTIVATION_CODE=<activation-code>` 3. 將安裝指令碼下載至您的虛擬機器。 `curl --proto "https" -o "ecs-anywhere-install.sh" \ > "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh"`	

任務	描述	所需技能
	4. 在虛擬機器上執行安裝指令碼。 `sudo bash ecs-anywhere-install.sh \ --cluster EcsAnywhereCluster \ --activation-id $ACTIVATION_ID \ --activation-code $ACTIVATION_CODE \ --region <region-name>` 這會將您的虛擬機器設定為 Amazon ECS Anywhere 外部執行個體，並在 Amazon ECS 叢集中註冊該執行個體。如需詳細資訊，請參閱 Amazon ECS 文件中的將外部執行個體註冊到叢集。如果您遇到任何問題，請參閱疑難排解一節。	
驗證 Amazon ECS Anywhere 不在和外部虛擬機器的狀態。	若要確認您的虛擬機器是否已連線至 Amazon ECS 控制平面並執行，請使用下列命令。 `$aws ssm describe-instance-information $aws ecs list-container-instances --cluster $CLUSTER_NAME`	DevOps 工程師

部署 CI/CD 管線

任務	描述	所需技能
在 CodeCommit 回購中創建一個分支。	通過創建存儲庫的第一個提交來創建一個在 CodeCommit repo `main` 中命名的分支。您可以依照 AWS 文件在中建立提交 CodeCommit。下列是範例命令。 `aws codecommit put-file \ --repository-name EcsAnywhereRepo \ --branch-name main \ --file-path README.md \ --file-content "Test" \ --name "Dev Ops" \ --email "devops@example.com" \ --commit-message "Adding README."`	DevOps 工程師
設置回購鏡像。	您可以在外部來源之間鏡像 GitLab 存放庫。您可以選取作為來源的儲存庫。分支、標記和提交會自動同步。在託管應用程式和 GitLab 儲存區域的儲存庫之間設定推入鏡像。CodeCommit 如需指示，請參閱從設定推送鏡像 GitLab 到 CodeCommit (說明GitLab 文件)。 附註：依預設，鏡像會自動同步存放庫。如果您想要手動	DevOps 工程師

任務	描述	所需技能
	更新儲存庫，請參閱更新鏡像 (GitLab 說明文件)。	
部署 CI/CD 管線堆疊。	輸入以下命令來部署 `EcsAnywherePipelineStack` 堆疊。 `$cdk deploy EcsAnywherePipelineStack`	DevOps 工程師

任務	描述	所需技能
測試 CI/CD 管線。	1. 進行應用程式程式碼變更，並將其推送至來源、內部部署 GitLab 存放庫。如需詳細資訊，請參閱推送選項 (GitLab 文件)。例如，編輯 `../application/index.html` 檔案以更新應用程式版本值。 2. 當代碼被複製到 CodeCommit 存儲庫時，這將啟動 CI/CD 管道。執行以下任意一項： • 如果您使用自動鏡像來同步 GitLab 存放庫與存 CodeCommit 放庫，請繼續執行下一個步驟。 • 如果您使用手動鏡像，請依照更新鏡像 (說明 GitLab 文件) 中的指示，將應用程式程式碼變更推送至 CodeCommit 存放庫。 3. 在本機電腦的網頁瀏覽器中，輸入 http://localhost:80。這將打開 NGINX 網頁，因為端口 80 被轉發到流浪文件中的本地主機。確認您可以檢視更新的應用程式版本值。這會驗證管線和映像部署。 4. (選擇性) 如果您想要在 AWS 管理主控台驗證部署，請執行下列動作：	DevOps 工程師

任務	描述	所需技能
	a. 開啟 Amazon ECS 主控台，網址為 https://console.aws.amazon.com/ecs/。 b. 從導覽列中選取要使用的「區域」。 c. 在導覽窗格中，選擇叢集。 d. 在 [叢集] 頁面上，選取EcsAnywhereCluster叢集。 e. 選擇「作業定義」。 f. 確認容器正在執行。	

清除

任務	描述	所需技能
清理並刪除資源。	逐步完成此模式之後，您應該移除您建立的 proof-of-concept 資源。若要清理，請輸入下列指令。 `$cdk destroy EcsAnywherePipelineStack` `$cdk destroy EcsAnywhereInfraStack`	DevOps 工程師

故障診斷

問題	解決方案
安裝套件相依性時遺失套件的錯誤。	輸入下列其中一個指令來解決遺失的套件。 `$npm ci` 或 `$npm install -g @aws-cdk/<package_name>`
當您在虛擬機器上執行`aws ssm create-activation` 命令時，您會收到下列錯誤。 `An error occurred (ValidationException) when calling the CreateActivation operation: Nonexistent role or missing ssm service principal in trust policy: arn:aws:iam::000000000000:role/EcsAnywhereInstanceRole`	`EcsAnywhereInfraStack` 堆疊尚未完全部署，且尚未建立執行此命令所需的 IAM 角色。在 CloudFormation 主控台中檢查堆疊狀態。狀態變更為後重試指令`CREATE_COMPLETE` 。
Amazon ECS 運作狀態檢查會傳回`UNHEALTHY` ，而您在 Amazon ECS 主控台中叢集的「服務」區段中看到下列錯誤。 `service EcsAnywhereService was unable to place a task because no container instance met all of its requirements. Reason: No Container Instances were found in your cluster.`	輸入以下命令，在您的流浪虛擬機器上重新啟動 Amazon ECS 代理程式。 `$vagrant ssh` `$sudo systemctl restart ecs` `$sudo systemctl status ecs`

相關資源

- Amazon ECS Anywhere 營銷頁面
- Amazon ECS Anywhere 文檔
- Amazon ECS Anywhere 演示（視頻）
- Amazon ECS Anywhere 研討會樣品 () GitHub
- 儲存庫鏡像 (GitLab 說明文件)

更多模式

- 使用 AWS Transit Gateway 自動化區域間對等互連的設定
- 使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式
- 使用萬 LiveData 迪斯科遷移器將 Hadoop 資料遷移到 Amazon S3
- 使用 HCX 自動化功能移轉 VMware 虛擬機器
- 使用 VMware HCX 將工作負載遷移到 AWS 上的 VMware 雲端
- 在 AWS 上從 F5 遷移到 Application Load Balancer 時修改 HTTP 標頭
- ???
- 使用 BMC 探索查詢擷取移轉資料以進行移轉規劃
- 使用 Serverspec 進行基礎架構程式碼的測試驅動開發

基礎設施

主題

使用工作階段管理員和 Amazon EC2 執行個體 Connect 存取防禦主機

由皮奧特·喬特科夫斯基（AWS）和維托爾德·科瓦利克（AWS）創建

程式碼儲存庫：使用工作階段管理員和 Amazon EC2 執行個體 Connect 存取防禦主機	環境：PoC 或試點	技術：基礎架構；雲端原生；安全性、身分識別、合規性；網路
AWS 服務：Amazon EC2; AWS Systems Manager; Amazon VPC		

Summary

防禦主機 (有時稱為 Jumpbox) 是一種伺服器，可提供從外部網路到私人網路中資源的單一存取點。暴露於外部公用網路 (例如網際網路) 的伺服器會對未經授權的存取造成潛在的安全風險。保護和控制對這些伺服器的存取非常重要。

此模式說明如何使用工作階段管理員和 Amazon EC2 執行個體 Connect 安全地連接到 AWS 帳戶中部署的 Amazon 彈性運算雲端 (Amazon EC2) 堡壘主機。工作階段管理員是 AWS Systems Manager 的一項功能。這種模式的好處包括：

- 部署的堡壘主機沒有任何公開的入站連接埠暴露在公用網際網路上。這會減少潛在的攻擊面。
- 您不需要在 AWS 帳戶中存放和維護長期安全殼層 (SSH) 金鑰。相反地，每個使用者每次連線到防禦主機時，都會產生一個新的安全殼層 key pair。附加至使用者 AWS 登入資料的 AWS Identity and Access Management (IAM) 政策可控制對防禦主機的存取。

目標受眾

此模式適用於對亞馬遜 EC2，Amazon Virtual Private Cloud (VPC) 和哈希科普地形有基本了解經驗的讀者。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已安裝和設定
- 已安裝 AWS CLI 的工作階段管理員外掛程式
- 地形 CLI，已安裝
- Terraform 狀態的儲存，例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體，以及做為存放 Terraform 狀態的遠端後端的 Amazon DynamoDB 表格。如需使用 Terraform 狀態的遠端後端的詳細資訊，請參閱 S3 後端 (Terraform 文件)。如需使用 S3 後端設定遠端狀態管理的程式碼範例，請參閱 remote-state-s3 後端 (Terraform 登錄)。請注意以下要求：
  - S3 儲存貯體和 DynamoDB 資料表必須位於相同的 AWS 區域。
  - 建立 DynamoDB 表時，分割區索引鍵必須是 `LockID` (區分大小寫)，且分區索引鍵類型必須是。`String`所有其他表格設定必須為其預設值。如需詳細資訊，請參閱 DynamoDB 文件中的關於主索引鍵和建立表格。
- 安裝的 SSH 用戶端

限制

- 這種模式旨在作為概念證明（PoC）或作為進一步發展的基礎。在生產環境中不應以目前的形式使用它。部署之前，請調整儲存庫中的範例程式碼，以符合您的需求和使用案例。
- 此模式假設目標防禦主機使用 Amazon Linux 2 做為其作業系統。雖然可以使用其他 Amazon 機器映像 (AMI)，但其他作業系統超出此模式的範圍。
- 在此模式中，防禦主機位於沒有 NAT 閘道和網際網路閘道的私有子網路中。這種設計將 EC2 實例與公共互聯網隔離。您可以新增特定的網路組態，以便它與網際網路通訊。如需詳細資訊，請參閱 Amazon V PC 文件中的將虛擬私有雲端 (VPC) Connect 到其他網路。同樣地，除非您明確授予許可，否則堡壘主機無法存取 AWS 帳戶中的任何其他資源。如需詳細資訊，請參閱 IAM 文件中的以資源為基礎的政策。

產品版本

- AWS CLI 第 2 版
- 地形版本 1.3.9 版本

架構

目標技術堆疊

- 具有單一私有子網路的 VPC
- 下列介面 VPC 端點：
  - `amazonaws.<region>.ssm` – Systems Manager 服務的端點。
  - `amazonaws.<region>.ec2messages`— Systems Manager 使用此端點從 SSM 代理程式呼叫 Systems Manager 服務。
  - `amazonaws.<region>.ssmmessages`— 工作階段管理員使用此端點透過安全資料通道連接到 EC2 執行個體。
- 執行 Amazon 2 的 `t3.nano` EC2 執行個體
- IAM 角色和執行個體設定檔
- 適用於端點和 EC2 執行個體的 Amazon VPC 安全群組和安全群組規則

目標架構

該圖顯示了以下過程：

1. 使用者假設具有執行下列動作的權限的 IAM 角色：
   - 驗證、授權並連線至 EC2 執行個體
   - 使用階段作業管理員啟動階段
2. 使用者透過工作階段管理員啟動 SSH 工作階段。
3. 工作階段管理員會驗證使用者、驗證相關 IAM 政策中的許可、檢查組態設定，並傳送訊息給 SSM 代理程式以開啟雙向連線。
4. 使用者透過 Amazon EC2 中繼資料將安全殼層公開金鑰推送到防禦主機。這必須在每次連接之前完成。SSH 公開金鑰會維持 60 秒的可用狀態。
5. 防禦主機會與 Systems Manager 和 Amazon EC2 的介面 VPC 端點進行通訊。
6. 使用者使用 TLS 1.2 加密的雙向通訊通道，透過工作階段管理員存取防禦主機。

自動化和規模

下列選項可用於自動化部署或擴充此架構：

- 您可以透過持續整合和持續交付 (CI/CD) 管道部署架構。
- 您可以修改程式碼以變更防禦主機的執行個體類型。

- 您可以修改程式碼以部署多個防禦主機。在bastion-host/main.tf檔案中的aws_instance資源區塊中，新增中count繼引數。如需詳細資訊，請參閱地形文件。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。此模式使用「會話管理器」，這是 Systems Manager 器的一種功能。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他工具

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。此模式使用地形 CLI。

代碼存儲庫

使用工作階段管理員和 Amazon EC2 執行個體 Connect 存放庫，可在存 GitHub 取防禦主機中取得此模式的程式碼。

最佳實務

- 我們建議您使用自動化程式碼掃描工具來改善程式碼的安全性和品質。此模式是使用 Cch kov (IaC 的靜態程式碼分析工具) 進行掃描。建議您至少使用和 `terraform fmt -check -recursive` Terraform 指令來執行基本的驗證`terraform validate`和格式化檢查。

- 為 IaC 添加自動化測試是一個很好的做法。如需有關測試 Terraform 程式碼之不同方法的詳細資訊，請參閱測試地形 (HashiCorp Terraform 部落格文章)。
- 在部署期間，每次偵測到新版本的 Amazon Linux 2 AMI 時，Terraform 都會使用取代 EC2 執行個體。這會部署新版作業系統，包括修補程式和升級。如果部署排程很少發生，這可能會造成安全性風險，因為執行個體沒有最新的修補程式。經常更新並將安全修補程式套用至部署的 EC2 執行個體非常重要。如需詳細資訊，請參閱 Amazon EC2 中的更新管理。
- 由於此模式是概念證明，因此它使用 AWS 受管政策，例如`AmazonSSMManagedInstanceCore`. AWS 受管政策涵蓋常見使用案例，但不授與最低權限許可。根據您的使用案例需要，建議您建立自訂原則，以針對此架構中部署的資源授與最低權限權限。如需詳細資訊，請參閱開始使用 AWS 受管政策並轉向最低權限許可。
- 使用密碼來保護對 SSH 金鑰的存取，並將金鑰儲存在安全的位置。
- 設定防禦主機的記錄和監視。從操作和安全性的角度來看，記錄和監控是維護系統的重要組成部分。有多種方法可以監視防禦主機中的連線和活動。如需詳細資訊，請參閱「Systems Manager」文件中的下列主題：
  - 監控 AWS Systems Manager
  - AWS Systems Manager 中的記錄和監控
  - 稽核會話活動
  - 記錄會話活動

史诗

部署資源

任務	描述	所需技能
克隆代碼存儲庫。	1. 在指令行介面中，將工作目錄變更為要儲存範例檔案的位置。 2. 輸入以下命令。 `git clone https://github.com/aws-samples/secured-bastion-host-terraform.git`	DevOps 工程師, 開發者

任務	描述	所需技能
初始化地形工作目錄。	此步驟僅對於第一個部署而言是必要的。如果您要重新部署病毒碼，請跳至下一個步驟。 在複製存放庫的根目錄中，輸入下列命令，其中： • $S3_STATE_BUCKET 是包含地形狀態的 S3 儲存貯體的名稱 • $PATH_TO_STATE_FILE 是地形狀態文件的關鍵，例如 infra/bastion-host/tetfstate • $AWS_REGION 是部署 S3 儲存貯體的區域 `terraform init \` `    -backend-config="bucket=$S3_STATE_BUCKET" \` `    -backend-config="key=$PATH_TO_STATE_FILE" \` `    -backend-config="region=$AWS_REGION` 注意：或者，您可以開啟 config.tf 檔案，並在此terraform 段落中手動提供這些值。	DevOps 工程師, 開發人員, 地形

任務	描述	所需技能
部署資源。	1. 在複製的存放庫的根目錄中，輸入下列命令。 `terraform apply -var-file="dev.tfvars"` 2. 檢閱將套用至 AWS 帳戶的所有變更清單，然後確認部署。 3. 等到所有資源都部署完畢。	DevOps 工程師, 開發人員, 地形

設定本機環境

任務	描述	所需技能
設定 SSH 連線。	更新 SSH 設定檔以允許透過工作階段管理員進行 SSH 連線。如需指示，請參閱允許工作階段管理員的 SSH 連線。這允許授權的使用者輸入 Proxy 命令，該命令會啟動工作階段管理員工作階段，並透過雙向連線傳輸所有資料。	DevOps 工程師
產生安全殼層金鑰。	輸入下列命令以產生本機私密和公開安全殼層 key pair。您可以使用此 key pair 來連線到防禦主機。 `ssh-keygen -t rsa -f my_key`	DevOps 工程師, 開發者

使用工作階段管理員 Connect 線到堡壘主機

任務	描述	所需技能
取得執行個體 ID。	1. 若要連線到已部署的防禦主機，您需要 EC2 執行個體的 ID。執行下列其中一項動作來找出 ID： • 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。在導覽窗格中，選擇執行個體。找到防禦主機執行個體。 • 在 AWS CLI 中，輸入以下命令。 `aws ec2 describe-instances` 若要篩選結果，請輸入以下指令，其中$BASTION_HOST_TAG 是指派給防禦主機的標籤。此標籤的預設值為sandbox-dev-bastion-host 。 `aws ec2 describe-instances \ --filters "Name=tag:Name,Values=$BASTION_HOST_TAG" \ --output text \ --query 'Reservations[*].Instances[*].InstanceId' \`	一般 AWS

任務	描述	所需技能
	--output text 2. 複製 EC2 執行個體的識別碼。您稍後使用此 ID。	

任務	描述	所需技能
傳送安全殼層公開金鑰。	注意：在此段落中，您會將公開金鑰上傳至防禦主機的執行個體中繼資料。上傳金鑰之後，您有 60 秒的時間可以開始與防禦主機的連線。60 秒後，會移除公開金鑰。如需詳細資訊，請參閱此病毒碼的疑難排解一節。快速完成後續步驟，以防止在連線到防禦主機之前移除金鑰。 1. 使用 EC2 執行個體連線，將安全殼層金鑰傳送至防禦主機。輸入以下命令，其中： • $INSTANCE_ID 是 EC2 執行個體的識別碼 • $PUBLIC_KEY_FILE 是公開金鑰檔案的路徑，例如 my_key.pub 重要事項：請務必使用公開金鑰，而非私密金鑰。 `aws ec2-instance-connect send-ssh-public-key \ --instance-id $INSTANCE_ID \ --instance-os-user ec2-user \ --ssh-public-key file://$PUBLIC_KEY_FILE`	一般 AWS

任務	描述	所需技能
	2. 請等待，直到您收到指出金鑰已成功上傳的訊息。立即繼續下一個步驟。	
Connect 到堡壘主機。	1. 輸入下列命令，透過工作階段管理員連線到堡壘主機，其中： • `$PRIVATE_KEY_FILE` 是您私鑰的路徑，例如 `my_key` • `$INSTANCE_ID` 是 EC2 執行個體的識別碼 `ssh -i $PRIVATE_KEY_FILE ec2-user@$INSTANCE_ID` 2. 輸入以確認連線`yes`。這會使用工作階段管理員開啟 SSH 連線。 注意：還有其他選項可用來開啟與防禦主機的 SSH 連線。如需詳細資訊，請參閱此模式的其他資訊一節中的與防禦主機建立 SSH 連線的替代方法。	一般 AWS

(選用) 清除

任務	描述	所需技能
移除部署的資源。	1. 若要移除所有已部署的資源，請從複製的存放庫的根目錄執行下列命令。	DevOps 工程師, 開發人員, 地形

任務	描述	所需技能
	`terraform destroy -var-file="dev.tfvars"` 2. 確認移除資源。	

故障診斷

問題	解決方案
`TargetNotConnected` 嘗試連線到防禦主機時發生錯誤	1. 根據 Amazon EC2 文件中的執行個體重新開機中的指示，將防禦主機重新開機。 2. 執行個體成功重新啟動後，請將公開金鑰重新傳送到防禦主機，然後重新嘗試連線。
`Permission denied` 嘗試連線到防禦主機時發生錯誤	將公開金鑰上傳到防禦主機之後，您只有 60 秒的時間可以啟動連線。60 秒後，金鑰會自動移除，而且您無法使用金鑰連線至執行個體。如果發生這種情況，您可以重複步驟將金鑰重新傳送至執行個體。

相關資源

AWS 文件

- AWS Systems Manager 會話管理器 (Systems Manager 文件)
- 安裝 AWS CLI 的工作階段管理員外掛程式 (Systems Manager 文件)
- 允許工作階段管理員的 SSH 連線 (Systems Manager 文件)
- 關於使用 EC2 執行個體 Connect (Amazon EC2 文件)
- 使用 EC2 執行個體 Connect 進行連線 (Amazon EC2 文件)
- Amazon EC2 的身分和存取管理 (Amazon EC2 文件)
- 使用 IAM 角色將許可授與在 Amazon EC2 執行個體上執行的應用程式 (IAM 文件)

- IAM 中的安全最佳做法 (IAM 文件)
- 使用安全群組控制資源流量 (Amazon VPC 文件)

其他資源

- 地形開發者網頁
- 命令：驗證（地形文檔）
- 命令：fmt（地形文檔）
- 測試 HashiCorp 地形（HashiCorp 博客文章）
- 切科夫網頁

其他資訊

與防禦主機建立 SSH 連線的替代方法

網路埠轉遞

您可以使用`-D 8888`此選項來開啟具有動態連接埠轉送的 SSH 連線。有關更多信息，請參閱以下說明。以下是使用連接埠轉送來開啟 SSH 連線的命令範例。

```
ssh -i $PRIVATE_KEY_FILE -D 8888 ec2-user@$INSTANCE_ID
```

這是一種連接打開 SOCKS 代理，該代理可以通過堡壘主機從本地瀏覽器轉發流量。如果您使用的是 Linux 或 MacOS，若要查看所有選項，請輸入`man ssh`。這會顯示 SSH 參考手冊。

使用提供的腳本

您可以使用程式碼儲存庫中包含的 connect.sh 指令碼，而不是手動執行使用工作階段管理員 Connect 到防禦主機中所述的步驟。此指令碼會產生 SSH key pair、將公開金鑰推送至 EC2 執行個體，並啟動與防禦主機的連線。當您執行指令碼時，您可以將標籤和金鑰名稱當做引數傳遞。以下是執行指令碼的命令範例。

```
./connect.sh sandbox-dev-bastion-host my_key
```

使用 AWS 受管 Microsoft AD 和現場部署 Microsoft 活動目錄集中 DNS 解析

由布萊恩·威斯特摩蘭（AWS）創建

環境：生產	技術：基礎設施; 網絡 DevOps; 安全性，身份，合規性; 操作系統	工作量：Microsoft
AWS 服務：AWS 管理 Microsoft AD；Amazon Route 53；AWS RAM；AWS Directory Service；AWS Organizations；AWS Direct Connect；AWS CLI		

Summary

此模式透過使用適用於 Microsoft 活動目錄的 AWS 目錄服務 (AWS 受管 Microsoft AD)，提供在 AWS 多帳戶環境中集中網域名稱系統 (DNS) 解析的指導。在此模式中，AWS DNS 命名空間是現場部署 DNS 命名空間的子網域。此模式也提供有關如何設定現場部署 DNS 伺服器，以便在現場部署 DNS 解決方案使用 Microsoft Active Directory 時將查詢轉寄給 AWS 的指導。

先決條件和限制

先決條件

- 使用 AWS 組織設定的 AWS 多帳戶環境。
- AWS 帳戶之間建立的網路連線。
- 在 AWS 與現場部署環境之間建立的網路連線 (使用 AWS Direct Connect 或任何類型的 VPN 連線)。
- 在本機工作站上設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- AWS Resource Access Manager (AWS RAM) 用於在帳戶之間共用 Amazon Route 53 規則。因此，必須在 AWS Organizations 環境中啟用共用功能，如史詩一節所述。

限制

- AWS 受管 Microsoft AD 標準版有 5 個股票的限制。
- AWS 受管 Microsoft AD 企業版的股票限制為 125 個。
- 此模式的解決方案僅限於支援透過 AWS RAM 共用的 AWS 區域。

產品版本

- Microsoft 活動目錄上運行視窗服務器

架構

目標架構

在這種設計中，AWS 受管 Microsoft AD 安裝在共享服務 AWS 帳戶中。儘管這不是必需的，但此模式會假設此配置。如果您在不同的 AWS 帳戶中設定 AWS 受管 Microsoft AD，則可能需要相應地修改史詩部分中的步驟。

此設計使用 Route 53 解析器，透過使用 Route 53 規則來支援名稱解析。如果現場部署 DNS 解決方案使用 Microsoft DNS，則為 AWS 命名空間 (aws.company.com) 建立條件式轉送規則，這是公司 DNS 命名空間 (company.com) 的子網域，並不簡單。如果您嘗試創建一個傳統的條件轉發器，它會導致錯誤。這是因為 Microsoft 活動目錄已被認為是授權的任何子域的. company.com 要解決此錯誤，您必須首先創建一個委託aws.company.com以委託該命名空間的權限。然後，您可以建立條件式轉寄站。

根據根 AWS 命名空間，每個支點帳戶的虛擬私有雲 (VPC) 都可以擁有自己的唯一 DNS 命名空間。在此設計中，每個支點帳戶都會在基礎 AWS 命名空間中附加帳戶名稱的縮寫。在支點帳戶中建立私有託管區域之後，這些區域會與支點帳戶中的 VPC 以及中央 AWS 網路帳戶中的 VPC 相關聯。這可讓中央 AWS 網路帳戶回答與支點帳戶相關的 DNS 查詢。

自動化和規模

此設計利用 Route 53 解析器端點在 AWS 和您的現場部署環境之間擴展 DNS 查詢。 每個 Route 53 解析器端點包含多個彈性網路介面 (分散在多個可用區域)，而每個網路介面每秒最多可處理 10,000 個查詢。 Route 53 解析器每個端點最多支援 6 個 IP 位址，因此此設計總共支援每秒高達 60,000 個 DNS 查詢，分散在多個可用區域，以實現高可用性。

此外，此模式會自動考慮 AWS 內部的 future 成長。在內部部署設定的 DNS 轉送規則不需要修改，以支援新增至 AWS 的新 VPC 及其相關聯的私有託管區域。

工具

AWS 服務

- 適用於 Microsoft 活動目錄的 AWS Directory Service 可讓您的目錄感知工作負載和 AWS 資源在 AWS 雲端中使用 Microsoft 活動目錄。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Resource Access Manager (AWS RAM) 可協助您在 AWS 帳戶之間安全地共用資源，以減少營運開銷，並提供可見性和可稽核性。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。在此模式中，AWS CLI 用於設定 Route 53 授權。

史诗

建立和共用 AWS 受管 Microsoft AD 目錄

任務	描述	所需技能
部署 AWS 受管 Microsoft AD。	1. 建立並設定新目錄。如需詳細步驟，請參閱 AWS 目錄服務管理指南中的建立 AWS 受管 Microsoft AD 目錄。 2. 記錄 AWS 受管 Microsoft AD 網域控制站的 IP 位址。 這些將在稍後的步驟中引用。	AWS 管理員

任務	描述	所需技能
共享目錄。	建立目錄之後，與 AWS 組織中的其他 AWS 帳戶共用該目錄。如需指示，請參閱《AWS Directory Service 管理指南》中的共用目錄。 注意：AWS 受管 Microsoft AD 標準版的共用限制為 5 個。 企業版的股份限制為 125 股。	AWS 管理員

設定 Route 53

任務	描述	所需技能
建立 Route 53 解析器。	Route 53 解析器可促進 AWS 與現場部署資料中心之間的 DNS 查詢解析。 1. 按照 Route 53 開發人員指南中的 i 結構安裝 Rou te 53 解析器。 2. 在中央 AWS 網路帳戶 VPC 內至少兩個可用區域的私有子網路中設定 Route 53 解析器，以獲得高可用性。 注意：雖然不需要使用中央 AWS 網路帳戶 VPC，但其餘步驟會採用此組態。	AWS 管理員
建立 Route 53 規則。	您的特定使用案例可能需要大量的 Route 53 規則，但您需要將下列規則設定為基準：	AWS 管理員

任務	描述	所需技能
	• 內部部署命名空間的外寄規則 (company.com) 使用輸出 Route 53 解析器。 • 與分支 AWS 帳戶共用此規則。 • 將此規則與支點帳戶 VPC 產生關聯。 • AWS 命名空間 (aws.company.com) 的傳出規則，指向中央網路帳戶 Route 53 傳入解析器。 • 與分支 AWS 帳戶共用此規則。 • 將規則與支點帳戶 VPC 產生關聯。 • 請勿將此規則與中央 AWS 網路帳戶 VPC (其中包含 Route 53 解析器) 建立關聯。 • AWS 命名空間 (aws.company.com) 的第二個傳出規則，指向 AWS 受管 Microsoft AD 網域控制站 (使用上一個史詩中的 IP)。 • 將此規則與中央 AWS 網路帳戶 VPC (其中包含 Route 53 解析器) 建立關聯。 • 請勿將此規則與其他 AWS 帳戶共用或關聯。	

任務	描述	所需技能
	如需詳細資訊，請參閱 Route 53 開發人員指南中的管理轉送規則。	

設定內部部署活動目錄 DNS

任務	描述	所需技能
建立委派。	使用 Microsoft DNS 嵌入式管理單元 (dnsmgmt.msc) 為活動目錄內的company.com 命名空間建立新委派。委派網域的名稱應為aws。這會產生委派的完整網域名稱 (FQDN)。aws.company.com 對於名稱伺服器，請使用中央 DNS AWS 帳戶中 AWS 輸入 Route 53 解析器的 IP 位址來取得 IP 值，並用server.aws.company.com 於名稱。	Active Directory
建立條件式轉寄站。	使用 Microsoft DNS 嵌入式管理單元 (dnsmgmt.msc) 來建立新的條件式轉寄站。aws.company.com 使用 AWS 受管 Microsoft AD 網域控制站的 IP 位址做為條件式轉寄站的目標。	Active Directory

為分支 AWS 帳戶建立 Route 53 私有託管區域

任務	描述	所需技能
建立 Route 53 私人託管區域。	在每個支點帳戶中建立 Route 53 私人託管區域。 將此私有託管區域與支點帳戶 VPC 建立關聯。 如需詳細步驟，請參閱 Route 53 開發人員指南中的建立私有託管區域。	AWS 管理員
建立授權。	使用 AWS CLI 為中央 AWS 網路帳戶 VPC 建立授權。從每個分支 AWS 帳戶的內容執行此命令： `aws route53 create-vpc-association-authorization --hosted-zone-id <hosted-zone-id> \ --vpc VPCRegion=<region>,VPCId=<vpc-id>` 其中： • `<hosted-zone-id>` 是支點帳戶中的 Route 53 私人託管區域。 • `<region>`並且`<vpc-id>`是中央 AWS 網路帳戶 VPC 人雲端的 AWS 區域和 VPC 識別碼。	AWS 管理員
建立關聯。	使用 AWS CLI 為中央 AWS 網路帳戶 VPC 建立 Route 53 私有託管區域關聯。從中央 AWS 網路帳戶的內容執行此命令：	AWS 管理員

任務	描述	所需技能
	`aws route53 associate-vpc-with-hosted-zone --hosted-zone-id <hosted-zone-id> \ --vpc VPCRegion=<region>,VPCId=<vpc-id>` 其中： • `<hosted-zone-id>` 是支點帳戶中的 Route 53 私人託管區域。 • `<region>`並且`<vpc-id>`是中央 AWS 網路帳戶的 AWS 區域和 VPC 人雲端 ID。	

相關資源

- 使用 Route 53 解析器 (馬哈茂德·馬圖克提供的 AWS 部落格文章) 簡化多帳戶環境中的 DNS 管理
- 使用 AWS 受管 Microsoft AD 建立目錄 (AWS Directory Service 文件)
- 共用 AWS 受管 Microsoft AD 目錄 (AWS Directory Service 文件)
- 安裝 Route 53 解析器（Amazon R oute 53 文檔）
- 創建 Route 53 私有託管區域（Amazon Route 53 文檔）

使用 Amazon CloudWatch 觀察性存取管理員集中監控

由阿南德·克里希納瓦拉納西（AWS），吉米·摩根（AWS），阿希什·庫馬爾（AWS），巴拉吉吠陀（AWS），賈格迪什科莫庫拉（AWS），莎拉·錢德拉·波圖拉（AWS）和維韋克唐穆圖（AWS）創建

代碼存儲庫:cloudwatch-observability-access-manager-地形	環境：生產	技術：基礎設施、多帳戶策略、營運
AWS 服務：Amazon CloudWatch；Amazon CloudWatch 日誌		

Summary

觀察性對於監控、瞭解和疑難排解應用程式至關重要。跨多個帳戶的應用程式與 AWS Control Tower 或 landing zone 實作一樣，會產生大量的日誌和追蹤資料。若要快速疑難排解問題或瞭解使用者分析或商業分析，您需要跨所有帳戶的通用可觀察性平台。Amazon 可 CloudWatch 觀察性存取管理員可讓您從中央位置存取和控制多個帳戶日誌。

您可以使用「觀察性存取管理員」來檢視和管理來源帳戶所產生的可觀測性資料記錄。來源帳戶是為其資源產生可觀察性資料的個別 AWS 帳戶。可觀察性數據在源帳戶和監視帳戶之間共享。共用的可觀察性資料可以包含 Amazon 中的指標 CloudWatch、Amazon 日誌中的 CloudWatch 日誌以及 AWS X-Ray 中的追蹤。如需詳細資訊，請參閱可觀測性存取管理員文件。

此模式適用於擁有在多個 AWS 帳戶中執行的應用程式或基礎設施，且需要共同位置檢視日誌的使用者。它說明如何使用 Terraform 來設定可觀察性存取管理員，以監視這些應用程式或基礎結構的狀態和健康狀態。您可以透過多種方式安裝此解決方案：

- 作為一個獨立的 Terraform 模塊，您可以手動設置
- 使用持續整合與持續交付 (CI/CD) 管線
- 透過與其他解決方案整合，例如地形的 AWS Control Tower Account Factory (AFT)

Epics 一節中的說明涵蓋了手動實施。如需 AFT 安裝步驟，請參閱 Readme 檔案以瞭解 GitHub 可觀測性存取管理員存放庫。

先決條件和限制

先決條件

- 在您的系統或自動化管道中安裝或參考的 Terraform。（我們建議您使用最新版本。）
- 您可以用作集中監控帳戶的帳戶。其他帳戶會建立中央監控帳戶的連結，以便檢視記錄。
- (選用) 原始程式碼儲存庫，例如 AWS GitHub CodeCommit、Atlassian Bitbucket 或類似系統。如果您使用自動化的 CI/CD 管道，則不需要源代碼存儲庫。
- (選用) 建立提取要求 (PR) 以進行程式碼檢閱和程式碼協同作業的 GitHub權限。

限制

觀察性訪問管理器具有以下服務配額，無法更改。部署此功能之前，請考慮這些配額。如需詳細資訊，請參閱 CloudWatch 文件中的CloudWatch 服務配額。

- 來源帳戶連結：您最多可以將每個來源帳戶連結至五個監視帳戶。
- 接收器：每個帳戶只能使用一個接收器。

除此之外：

- 接收器和連結必須在相同的 AWS 區域中建立；它們不能是跨區域。
- 對於跨區域、跨帳戶監控，您可以針對警示和指標建立跨帳戶和跨區域 CloudWatch 儀表板，但記錄和追蹤除外。另一個選項是使用 Amazon OpenSearch 服務創建集中日誌記錄。

架構

零組件

Amazon CloudWatch 可觀察性存取管理器由兩個主要元件組成，可實現跨帳戶觀察性：

- 接收器可讓來源帳戶將可觀察性資料傳送至中央監視帳戶。接收器基本上提供了一個閘道結合，供來源帳戶連接。只能有一個接收器閘道或連接，多個帳戶可以連接到它。
- 每個源帳戶都有一個鏈接到接收器網關連接，並通過此鏈接發送觀察性數據。您必須先建立接收器，才能從每個來源帳戶建立連結。

架構

下圖說明了可觀察性訪問管理器及其組成部分。

工具

AWS 服務

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

工具

- Terraform 是一種基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署資源。HashiCorp
- 適用於地形的 AWS Control Tower Account Factory (AFT) 會設定 Terraform 管道，以協助您在 AWS Control Tower 中佈建和自訂帳戶。您可以選擇性地使用 AFT 在多個帳戶之間大規模設置可觀察性訪問管理器。

代碼存儲庫

此模式的代碼可在 GitHub 觀察性訪問管理器存儲庫中找到。

最佳實務

- 在 AWS Control Tower 環境中，將記錄帳戶標記為中央監控帳戶 (接收器)。
- 如果您有多個組織在 AWS Organizations Organization 中擁有多個帳戶，建議您在組態政策中加入組織而非個別帳戶。如果您的帳號數量很少，或者這些帳戶不屬於接收器組態策略中的組織，您可能會決定改為包含個別帳戶。

史诗

設定水槽模組

任務	描述	所需技能
複製儲存庫。	克隆 GitHub 觀察性訪問管理器存儲庫： `git clone https://github.com/aws-samples/cloudwatch-observability-access-manager-terraform`	AWS DevOps、雲端管理員、AWS 管理員
指定接收器模組的性質值。	在`main.tf`檔案 (位於存放庫的`deployments/aft-account-customizations/LOGGING/terraform/` 資料夾中) 中，指定下列屬性的值： • `sink_name` ：Amazon CloudWatch 水槽的名字。 • `allowed_oam_resource_types` ：觀察性存取管理員目前支援 CloudWatch 指標、日誌群組和 AWS X-Ray 追蹤。 • `allowed_source_accounts` ：允許將記錄檔傳送至中央接 CloudWatch 收器帳戶的來源帳戶。 • `allowed_source_organizations` ：允許將記錄檔傳送至中央接	AWS DevOps、雲端管理員、AWS 管理員

任務	描述	所需技能
	CloudWatch 收器帳戶的來源 Control Tower 組織。 如需詳細資訊，請參閱 AWS CloudFormation 文件AWS::Oam::Sink中的。	
安裝水槽模塊。	匯出您選擇作為監控帳戶的 AWS 帳戶登入資料，然後安裝可觀察性存取管理員接收器模組： `Terraform Init` `Terrafom Plan` `Terraform Apply`	AWS DevOps、雲端管理員、AWS 管理員

設定連結模組

任務	描述	所需技能
指定連結模組的性質值。	在`main.tf` 檔案 (位於存放庫的`deployments/aft-account-customizations/LOGGING/terraform/` 資料夾中) 中，指定下列屬性的值： • `account_label` ：使用下列其中一個值： • `$AccountName` ：帳戶的名稱。 • `$AccountEmail` ：全域唯一的電子郵件地址，其中包括電子郵件	AWS DevOps、雲端管理員、雲端架構師

任務	描述	所需技能
	網域 (例如hello@example.com) • $AccountEmailNoDomain ：沒有網域名稱的電子郵件地址。 • allowed_oam_resource_types ：觀察性存取管理員目前支援 CloudWatch 指標、日誌群組和 AWS X-Ray 追蹤。 如需詳細資訊，請參閱 AWS CloudFormation 文件AWS::Oam::Link中的。	
為個別帳戶安裝連結模組。	導出個人帳戶的憑據並安裝觀察性訪問管理器鏈接模塊： `Terraform Plan` `Terraform Apply` 您可以為每個帳戶單獨設置鏈接模塊，也可以使用 AFT 在大量帳戶中自動安裝此模塊。	AWS DevOps、雲端管理員、雲端架構師

核准 sink-to-link 連線

任務	描述	所需技能
檢查狀態訊息。	1. 登入至監控帳戶。 2. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/	

任務	描述	所需技能
	3. 在左側的導覽窗格中，選擇設定。 在右側，您應該會看到狀態訊息「監控帳戶已啟用」，並顯示綠色核取記號。這意味著監視帳戶具有可觀察性訪問管理器接收器，其他帳戶的鏈接將連接到該接收器。	

任務	描述	所需技能
核准 link-to-sink 連線。	1. 選擇狀態訊息下方的連結帳號的資源選項。這些資訊會確認這是監視帳戶、列出從租用戶來源帳戶 (記錄檔、指標、追蹤) 共用的資料，並將帳戶標籤顯示為 $AccountName。 此畫面提供兩種將租用戶帳戶連結至監控帳戶的選項：組織層級核准或帳戶層級核准。對於每個選項，您可以選擇下載 AWS CloudFormation 範本以進行核准，或個別核准每個帳戶。 2. 為了簡單起見，請選擇任何帳戶以在每個帳戶級別進行批准。此選項提供帳戶的核准連結。 3. 選擇「複製 URL」以複製連結。 4. 登入每個來源帳戶。 5. 在瀏覽器視窗中，貼上連結，然後選擇 [核准連結連線至接收器]。 6. 對其他來源帳戶重複此步驟。 如需詳細資訊，請參閱 Amazon CloudWatch 文件中的連結監控帳戶與來源帳戶。	AWS DevOps、雲端管理員、雲端架構師

驗證跨帳戶可觀察性資料

任務	描述	所需技能
檢視跨帳戶資料。	1. 登入中央監控帳戶。 2. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/ 3. 在左側導覽窗格中，選擇檢視跨帳戶記錄、指標和追蹤的選項。	AWS DevOps、雲端管理員、雲端架構師

(選擇性) 讓來源帳戶信任監視帳戶

任務	描述	所需技能
檢視來自其他帳戶的指標、儀表板、記錄、Widget 和警示。	作為附加功能，您可以與其他帳戶共享 CloudWatch 指標，儀表板，日誌，小部件和警報。每個帳戶都使用名為 CloudWatch-CrossAccountSharingRole 的 IAM 角色來存取此資料。 與中央監視帳戶具有信任關係的來源帳戶可以擔任此角色，並從監視帳戶檢視資料。 CloudWatch 提供建立角色的範例 CloudFormation 指令碼。選擇在 IAM 中管理角色，然後在您要查看數據的帳戶中運行此腳本。 {	AWS DevOps、雲端管理員、雲端架構師

任務	描述	所需技能
	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principa l": { "AWS": ["arn:aws:iam::XXXX XXXXX:root", "arn:aws:iam::XXXX XXXXX:root", "arn:aws:iam::XXXX XXXXX:root", "arn:aws:iam::XXXX XXXXX:root"] }, "Action": "sts:AssumeRole" }] }</pre> 如需詳細資訊，請參閱文件 CloudWatch 中 CloudWatch的啟用跨帳戶功能	

(選擇性) 從監控帳戶檢視跨帳戶跨區域

任務	描述	所需技能
設定跨帳戶、跨區域存取。	在中央監控帳戶中，您可以選擇添加帳戶選擇器，以便在	AWS DevOps、雲端管理員、雲端架構師

任務	描述	所需技能
	帳戶之間輕鬆切換並查看其數據，而無需進行身份驗證。 1. 登入中央監控帳戶。 2. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/ 3. 在左側導覽窗格中，選擇 [設定]。 4. 在 [檢視跨帳戶跨區域] 區段中，選擇 [設定]。 5. 選擇 [啟用]，然後選取 [在主控台中顯示選取器] 核取方塊。 6. 選擇這些選項的其中之一： • 帳戶 ID 輸入：每當您想要更改帳戶以查看跨帳戶數據時，此選項都會提示您手動輸入帳戶 ID。 • AWS 組織帳戶選擇器：如果您已 CloudWatch 與 AWS Organizations 整合，此選項會提供下拉式選擇器，其中包含組織中帳戶的完整清單。 • 自訂帳戶選擇器：此選項可讓您手動輸入帳戶 ID 清單以填入選取器。 7. 選擇儲存變更。	

任務	描述	所需技能
	如需詳細資訊，請參閱文件 CloudWatch 中的跨帳戶跨區域 CloudWatch 主控台。	

相關資源

- CloudWatch 跨帳戶可觀察性（Amazon CloudWatch 文檔）
- Amazon CloudWatch 觀察性訪問管理器 API 參考（Amazon CloudWatch 文檔）
- 資源：aws_oam_ 接收器（地形文檔）
- 資料來源：aws_oam_ 連結 (地形文件)
- CloudWatchObservabilityAccessManager(AWS 文件)

啟動時檢查 EC2 執行個體是否有強制標籤

環境：生產	技術：基礎架構；管理與治理；安全性、身分識別、合規性；雲端原生	AWS 服務：Amazon EC2; AWS CloudTrail; Amazon CloudWatch; Amazon SNS

Summary

Amazon Elastic Compute Cloud (Amazon EC2) 在 Amazon Web Services (AWS) Cloud 提供可擴展的運算容量。使用 Amazon EC2 可減少前期所需的硬體投資，讓您更快速開發並部署應用程式。

您可以使用標記以不同的方式對 AWS 資源進行分類。當您的帳戶中有許多資源，並且想要根據標籤快速識別特定資源時，EC2 執行個體標記非常有用。您可以使用標籤將自訂中繼資料指派給 EC2 執行個體。標籤由使用者定義的索引鍵和值組成。我們建議您建立一組一致的標籤，以符合組織的需求。

此模式提供 AWS CloudFormation 範本，可協助您監控 EC2 執行個體的特定標籤。範本會建立監控 AWS CloudTrail TagResource或 CloudWatch 事件的 Amazon 活UntagResource動事件，以偵測新的 EC2 執行個體標記或標籤移除。如果缺少預先定義的標籤，它會呼叫 AWS Lambda 函數，該函數會使用 Amazon Simple Notification Service (Amazon SNS) 將違規訊息傳送到您提供的電子郵件地址。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 一個 Amazon Simple Storage Service (Amazon S3) 存儲桶，用於上傳提供的 Lambda 代碼。
- 您希望接收違規通知的電子郵件地址。

限制

- 此解決方案支持 CloudTrail TagResource或UntagResource事件。它不會為任何其他事件建立通知。
- 此解決方案僅檢查標籤鍵。它不監視關鍵值。

架構

工作流架構

自動化和規模

- 您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需在每個區域或帳戶中執行一次範本。

工具

AWS 服務

- Amazon EC2 — 亞馬遜彈性運算雲（Amazon EC2）是一種 Web 服務，可在雲中提供安全、可調整大小的運算容量。它旨在讓開發人員更輕鬆地進行 Web 規模的雲端運算。
- AWS CloudTrail — CloudTrail 是一項 AWS 服務，可協助您對 AWS 帳戶進行管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。 CloudWatch 事件會在發生作業變更時知道，並視需要採取更正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。
- AWS Lambda — Lambda 是一種運算服務，可支援執行程式碼，而不需要佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種網路服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。

Code

該模式包括一個包含兩個文件的附件：

- `index.zip`是包含此模式之 Lambda 程式碼的壓縮檔案。
- `ec2-require-tags.yaml`是部署 Lambda 程式碼的 CloudFormation 範本。

有關如何使用這些文件的信息，請參見 Epics 部分。

史诗

部署 Lambda 程式碼

任務	描述	所需技能
將代碼上傳到 S3 存儲桶。	建立新的 S3 儲存貯體，或使用現有的 S3 儲存貯體上傳附加的`index.zip` 檔案 (Lambda 程式碼)。此儲存貯體必須與您要監控的資源 (EC2 執行個體) 位於相同的 AWS 區域。	雲端架構師
部署 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 Cloudformation 主控台，然後部署附件中提供的`ec2-require-tags.yaml` 檔案。在下一個史詩中，提供模板參數的值。	雲端架構師

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選取的 S3 儲存貯體的名稱。此 S3 儲存貯體包含 Lambda 程式碼的 .zip 檔案，且必須與 CloudFormation 範本和您要監控的 EC2 執行個體位於相同的 AWS 區域。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如，`index.zip`	雲端架構師

任務	描述	所需技能
	或controls/index.zip)。	
提供電子郵件地址。	提供您要接收違規通知的作用中電子郵件地址。	雲端架構師
定義記錄層級。	指定記錄日誌層級和詳細資訊。 Info指定應用程式進度的詳細資訊訊息，應僅用於偵錯。 Error指定仍然允許應用程式繼續執行的錯誤事件。 Warning指定潛在的有害情況。	雲端架構師
輸入所需的標籤關鍵字。	輸入您要檢查的標籤鍵。如果要指定多個鍵，請用逗號分隔它們，不帶空格。(例如，ApplicationId,CreatedBy,Environment, Organization 搜尋四個機碼。) E CloudWatch vents 事件會搜尋這些標籤鍵，並在找不到時傳送通知。	雲端架構師

確認訂閱

任務	描述	所需技能
確認電子郵件訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件訊息傳送至您提供的電子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 建立儲存貯體 (Amazon S3 文件)
- 上傳物件 (Amazon S3 文件)
- 標記您的 Amazon EC2 資源 (Amazon EC2 文件)
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail (Amazon CloudWatch 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用工作階段管理員 Connect 到 Amazon EC2 執行個體

由傑森·科尼克（AWS），阿布舍克巴斯基科帕（AWS）和亞尼夫·羅恩（AWS）創建

環境：生產	技術：基礎架構、雲端原生、終端使用者運算、營運	AWS 服務：Amazon CloudWatch 日誌；AWS Systems Manager；Amazon EC2

Summary

此模式說明如何使用工作階段管理器 (AWS Systems Manager 的功能) 連接到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。使用此模式，您可以通過 Web 瀏覽器在 EC2 實例上運行 bash 命令。工作階段管理員不需要開啟輸入連接埠，也不需要 EC2 執行個體的公有 IP 地址。此外，也不需要使用不同的安全殼層 (SSH) 金鑰維護防禦主機。您可以使用 AWS Identity and Access Management (IAM) 政策管理工作階段管理員的存取，並設定記錄記錄功能，以記錄重要資訊，例如執行個體存取和動作。

在此模式中，您可以設定 IAM 角色，並將其與使用 Amazon 機器映像 (AMI) 佈建的 Linux EC2 執行個體相關聯。然後，您可以在 Amazon Lo CloudWatch gs 中設定記錄，並使用工作階段管理員啟動執行個體的工作階段。

雖然此模式連線到 Amazon Web Services (AWS) 雲端中的 Linux EC2 執行個體，但您可以使用此方法將工作階段管理員用於與其他伺服器 (例如現場部署伺服器或其他虛擬機器) 的連線。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 存取受管理節點的權限。如需指示，請參閱控制使用者工作階段存取受管理節點。
- `ssm`、`ec2`、和`s3`的 VPC 端點。`ec2messages` `ssmmessages`如需指示，請參閱 Systems Manager 說明文件中的建立 VPC 端點。

架構

目標技術堆疊

- 工作階段管理員
- Amazon EC2
- CloudWatch 日誌

目標架構

1. 使用者透過 IAM 驗證其身分和登入資料。
2. 使用者透過工作階段管理員啟動 SSH 工作階段，並將 API 呼叫傳送至 EC2 執行個體。
3. 安裝在 EC2 執行個體上的 AWS Systems Manager SSM 代理程式會連線到工作階段管理員並執行命令。
4. 為了稽核和監視目的，工作階段管理員會將記錄資料傳送至記 CloudWatch 錄檔。或者，您可以將日誌資料傳送到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。如需詳細資訊，請參閱使用 Amazon S3 記錄工作階段資料 (Systems Manager 文件)。

工具

AWS 服務

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。此模式使用 Amazon 機器映像（AMI）來佈建 Linux EC2 實例。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。這種模式使用會話管理器，Systems Manager 的功能。

最佳實務

我們建議您閱讀有關 AWS Well-Architected Framework 的安全性支柱的更多資訊，並探索加密選項，並在設定工作階段管理員 (Systems Manager 管理員文件) 中套用安全建議。

史诗

設定基礎架構

任務	描述	所需技能
建立 IAM 角色。	建立 SSM 代理程式的 IAM 角色。請遵循為 AWS 服務建立角色 (IAM 文件) 中的指示，並注意以下事項： 1. 對於 AWS 服務，請選擇 EC2。 2. 對於「權限原則」，請選擇`AmazonSSMManagedInstanceCore` 。 3. 在角色名稱中，輸入`EC2_SSM_Role` 。	AWS 系統管理員
建立 EC2 執行個體。	1. 建立 EC2 執行個體。請遵循啟動執行個體 (Amazon EC2 文件) 中的指示，並注意下列事項： a. 在「名稱和標籤」區段中，選擇「新增其他標籤」。在 Key (金鑰) 中，輸入 `Name`，並且在 Value (值) 中，輸入 `Production_Server_One` 。 b. 選擇已預先安裝 SSM 代理程式的 Amazon Linux AMI。如需完整清單，請參閱預先安裝 SSM 代理程式的 AMI (Systems Manager 說明文件)。	AWS 系統管理員

任務	描述	所需技能
	c. 在「進階詳細資料」區段的 IAM 執行個體設定檔中，選擇 EC2_SSM_Role。 2. 開啟 Systems Manager 主控台，網址為 https://console.aws.amazon.com/systems-manager/。 3. 在導覽窗格中，選擇 Fleet Manager。 4. 確認執行個體是否出現在受管節點清單中。	
設定記錄。	1. 在記錄檔中建立 CloudWatch 記錄群組。依照建立記錄群組 (CloudWatch 記錄檔說明文件) 中的指示進行。命名新的記錄群組`SessionManager` 。 2. 設定工作階段管理員的記錄。請遵循使用 Amazon CloudWatch 日誌記錄工作階段資料 (Systems Manager 文件) 中的指示，並注意下列事項： a. 請勿選取 [僅允許加密的 CloudWatch 記錄群組]。 b. 在 [從清單中選擇記錄群組] 中，選擇SessionManager。	AWS 系統管理員

連線到執行個體

任務	描述	所需技能
Connect 至 EC2 執行個體。	1. 在 Systems Manager 主控台中啟動工作階段。如需指示，請參閱啟動工作階段 (Systems Manager 說明文件)。對於目標執行個體，請選擇生產伺服器 _ One 執行個體左側的選項按鈕。 2. 建立連接後，運行幾個 bash 命令。 3. 在 Systems Manager 主控台中，結束工作階段。如需指示，請參閱結束工作階段 (Systems Manager 說明文件)。	AWS 系統管理員
驗證記錄。	1. 在 CloudWatch 記錄檔中，開啟記錄群組的記錄資料流。如需指示，請參閱檢視記錄資料 (CloudWatch 記錄檔文件)。 2. 在記錄資料中，確認已列出您在上一個故事中執行的指令。	AWS 系統管理員

故障診斷

問題	解決方案
IAM 問題	如需支援，請參閱疑難排解 (IAM 文件)。

相關資源

- 完成工作階段管理員先決條件 (Systems Manager
- 使用 Amazon 設計和實作日誌記錄和監控 CloudWatch (AWS Prescriptive Guidance)

在不支援 AWS 的 AWS 區域建立管道 CodePipeline

創建者阿南德·克里希納·瓦拉納西（AWS）

代碼存儲庫：invisible-codepipeline-unsupported-regions	環境：PoC 或試點	技術：基礎設施; DevOps
AWS 服務：AWS CodeBuild；AWS CodeCommit；AWS CodeDeploy；AWS CodePipeline		

Summary

AWS CodePipeline 是持續交付 (CD) 協調服務，屬於來自 Amazon Web Services (AWS) 的一組 DevOps 工具。它整合了各種來源 (例如版本控制系統和儲存解決方案)、AWS 和 AWS 合作夥伴的持續整合 (CI) 產品和服務，以及開放原始碼產品，為快速的應用程式和基礎設施部署提供 end-to-end 工作流程服務。

CodePipeline 不過，並非所有 AWS 區域都支援，而且擁有連接 AWS CI/CD 服務的隱形協調器非常有用。此模式描述如何在尚 CodePipeline 未透過 AWS CodeCommit、AWS 和 AWS 等 AWS CI/CD 服務支援的 AWS 區域實 end-to-end 作工作流程管道。 CodeBuild CodeDeploy

先決條件和限制

前提

- 有效的 AWS 帳戶
- AWS Cloud Development Kit (AWS CDK) CLI 2.28 版或更新版本

架構

目標技術堆疊

下圖顯示在不支援的區域中建立的管道 CodePipeline，例如非洲 (開普敦) 區域。開發人員將 CodeDeploy 配置文件（也稱為部署生命週期鉤子腳本）推送到託管的 Git 存儲庫。 CodeCommit（請參閱此模式提供的GitHub 存儲庫。） Amazon EventBridge 規則會自動啟動。 CodeBuild

CodeDeploy 組態檔案會從中擷取， CodeCommit 做為管線來源階段的一部分，然後傳輸到 CodeBuild。

在下一個階段， CodeBuild 執行下列工作：

1. 下載應用程式原始程式碼 TAR 檔案。您可以使用 AWS Systems Manager 的功能參數存放區來設定此檔案的名稱。
2. 下載組 CodeDeploy 態檔案。
3. 建立應用程式原始程式碼和特定於應用程式類型的 CodeDeploy 組合歸檔。
4. 使用合併的存檔，啟動 CodeDeploy 部署到 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。

工具

AWS 服務

- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodeDeploy自動部署到 Amazon EC2 或現場部署執行個體、AWS Lambda 函數或亞馬遜彈性容器服務 (Amazon ECS) 服務。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。

Code

此模式的程式碼可在「 GitHub CodePipeline 不支援的地區」儲存庫中取得。

史诗

設定開發人員工作站

任務	描述	所需技能
安裝 AWS CDK CLI。	如需相關指示，請參閱 AWS CDK 文件。	AWS DevOps
安裝 Git 用戶端。	要創建提交，您可以使用安裝在本地計算機上的 Git 客戶端，然後將提交推送到 CodeCommit 存儲庫。若要 CodeCommit 使用 Git 用戶端進行設定，請參閱CodeCommit 文件。	AWS DevOps
安裝 npm。	安裝 npm 軟件包管理器。如需詳細資訊，請參閱 npm 文件。	AWS DevOps

設置管道

任務	描述	所需技能
克隆代碼存儲庫。	執行下列命令，將「 GitHub CodePipeline 不支援的區域」儲存庫複製到您的本機電腦。 `git clone https://github.com/aws-samples/invisible-codepipeline-unsupported-regions`	DevOps 工程師
在中設定參數。	開啟`cdk.json`檔案並提供下列參數的值：	AWS DevOps

任務	描述	所需技能
	<pre>"pipeline_account" :"XXXXXXXXXXXX", "pipeline_region":" us-west-2", "repo_name": "app-dev-repo", "ec2_tag_key": "test-vm", "configName" : "cbdeployconfig", "deploymentGroupNa me": "cbdeploygroup", "applicationName" : "cbdeployapplicati on", "projectName" : "CodeBuildProject"</pre> 其中： • pipeline_account 是要建立管道的 AWS 帳戶。 • pipeline_region 是要建立管道的 AWS 區域。 • repo_name 是存 CodeCommit 放庫的名稱。 • ec2_tag_key 是附加到要部署代碼的 EC2 實例的標籤。 • configName 是組 CodeDeploy 態檔案的名稱。 • deploymentGroupNam e 是部 CodeDeploy 署群組的名稱。	

任務	描述	所需技能
	• `applicationName` 是 CodeDeploy 應用程式名稱。 • `projectName` 是 CodeBuild 專案名稱。	
設定 AWS CDK 建構程式庫。	在複製的 GitHub 存放庫中，使用下列命令安裝 AWS CDK 建構程式庫、建立應用程式，以及進行合成以產生應用程式的 AWS CloudFormation 範本。 `npm i aws-cdk-lib` `npm run build` `cdk synth`	AWS DevOps
部署範例 AWS CDK 應用程式。	在不支援的區域 (例如`af-south-1`) 中執行下列命令來部署程式碼。 `cdk deploy`	AWS DevOps

設定下列項目的 CodeCommit 儲存庫 CodeDeploy

任務	描述	所需技能
設定應用程式的 CI/CD。	複製您在`cdk.json`檔案中指定的 CodeCommit 存放庫 (`app-dev-repo` 依預設會呼叫此儲存庫)，以設定應用程式的 CI/CD 管線。 `git clone https://g` `it-codecommit.us-w`	AWS DevOps

任務	描述	所需技能
	`est-2.amazonaws.com/v1/repos/app-dev-repo` 其中存放庫名稱和區域取決於您在`cdk.json`檔案中提供的值。	

測試管道

任務	描述	所需技能
使用部署指示測試管線。	「GitHub CodePipeline 不支援的區域」儲存庫的`CodeDeploy_Files` 資料夾包含指示部署應用程式 CodeDeploy 的範例檔案。該`appspec.yml` 文件是一個 CodeDeploy 配置文件，其中包含用於控制應用程序部署流程的掛接。您可以使用範例檔案`index.html` `start_server.sh` 、`stop_server.sh` 、和`install_dependencies.sh` 來更新託管在 Apache 上的網站。以下是範例 — 您可以使用 GitHub 儲存庫中的程式碼來部署任何類型的應用程式。將檔案推送至 CodeCommit 存放庫時，會自動啟動不可見的管線。如需部署結果，請檢查 CodeBuild 和 CodeDeploy 主控台中個別階段的結果。	AWS DevOps

相關資源

- 開始使用 (AWS CDK 文件)
- Cloud Development Kit (CDK) 簡介 (AWS 工作坊工作室)
- AWS CDK 工作坊

使用私有靜態 IP 在 Amazon EC2 上部署卡桑德拉集群，以避免重新平衡

創建者迪平耆那教（AWS）

環境：PoC 或試點	來源：內部部署 VM	目標：Amazon EC2
R 類型：重新主機	工作負載：開源	技術：基礎架構；資料庫；移轉
AWS 服務：Amazon EC2		

Summary

Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的私有 IP 會在其整個生命週期中保留。不過，私有 IP 可能會在計劃或意外的系統當機期間變更；例如，在 Amazon 機器映像 (AMI) 升級期間。在某些情況下，保留私有靜態 IP 可以增強工作負載的效能和復原時間。例如，對 Apache Cassandra 種子節點使用靜態 IP 可防止叢集產生重新平衡的負荷。

此模式描述如何將次要 elastic network interface 至 EC2 執行個體，以在重新託管期間保持 IP 靜態。該模式側重於 Cassandra 集群，但是您可以將此實現用於受益於私有靜態 IP 的任何體系結構。

先決條件和限制

先決條件

- 有效的 Amazon 網路服務 (AWS) 帳戶

產品版本

- DataStax 版本
- 操作系統：Ubuntu 16.04.6 LTS

架構

來源架構

來源可以是現場部署虛擬機器 (VM) 或 AWS 雲端中 EC2 執行個體上的 Cassandra 叢集。下圖說明第二個案例。此範例包括四個叢集節點：三個種子節點和一個管理節點。在來源架構中，每個節點都附加了單一網路介面。

目標架構

目的地叢集託管在 EC2 執行個體上，每個節點都附加了次要 elastic network interface，如下圖所示。

自動化和規模

您也可以自動將第二個 elastic network interface 到 EC2 Auto Scaling 群組，如 AWS 知識中心影片中所述。

史诗

在 Amazon EC2 上配置卡桑德拉集群

任務	描述	所需技能
啟動 EC2 節點以託管卡桑德拉叢集。	在 Amazon EC2 主控台上，為您的 AWS 帳戶中的 Ubuntu 節點啟動四個 EC2 執行個體。三個（種子）節點用於 Cassandra 集群，第四個節點充當集群管理節點，您將在其中安裝 DataStax 企業（DSE）。 OpsCenter如需指示，請參閱 Amazon EC2 文件。	雲端工程師
確認節點通訊。	請確定四個節點可透過資料庫和叢集管理連接埠彼此通訊。	網絡工程師
在管理節點 OpsCenter 上安裝 DSE。	從管理節點上的 Debian 軟件包中安裝 DSE OpsCenter	DBA

任務	描述	所需技能
	6.1。如需指示，請參閱DataStax 文件。	

任務	描述	所需技能
建立次要網路介面。	卡桑德拉根據該節點的 EC2 實例的 IP 地址為每個節點生成一個通用唯一標識符（UUID）。此 UUID 用於在環上分配虛擬節點（vnode）。當 Cassandra 部署在 EC2 執行個體上時，IP 地址會在建立執行個體時自動指派給執行個體。 如果發生計劃或意外中斷，新 EC2 執行個體的 IP 位址會變更、資料分發變更，而且必須重新平衡整個環。這是不可取的。若要保留指派的 IP 位址，請使用具有固定 IP 位址的次要 elastic network interface。 1. 在 Amazon EC2 主控台上，選擇「網路界面」、「建立網路界面」。 2. 對於子網路，選取您在其中建立 EC2 執行個體的子網路。 3. 針對私人 IPv4 位址，選擇「自動指派」。 4. 在 [安全性群組] 中，選取安全性群組，然後選擇 [建立網路介面]。 如需有關建立網路界面的詳細資訊，請參閱 Amazon EC2 文件。	雲端工程師

任務	描述	所需技能
將次要網路介面連接至叢集節點。	1. 在 Amazon EC2 主控台上，選擇執行個體。 2. 選取您先前建立的 EC2 執行個體的核取方塊。 3. 選擇 Actions (動作)、Networking (網路)、Attach network interface (連接網路介面)。 4. 選取您在上一個步驟中建立的網路介面，然後選擇 [連接]。 如需有關連接網路界面的詳細資訊，請參閱 Amazon EC2 文件。	雲端工程師
在 Amazon EC2 中新增路由以解決非對稱路由問題。	當您連接第二個網路介面時，網路很可能會執行非對稱路由。為了避免這種情況，您可以為新的網路介面新增路由。 如需非對稱路由的深入解釋和修復，請參閱 AWS 知識中心影片或克 McManus服多家用伺服器上的非對稱路由 (2004 年 4 月 5 日在《Linux 日誌》中撰寫的文章)。	網絡工程師
更新 DNS 項目以指向次要網路介面 IP。	將節點的完整網域名稱 (FQDN) 指向次要網路介面的 IP。	網絡工程師

任務	描述	所需技能
安裝和使用 DSE 配置卡桑德拉集群。 OpsCenter	當叢集節點已準備好使用次要網路介面時，您可以安裝和配置 Cassandra 叢集。	DBA

從節點故障中復原叢集

任務	描述	所需技能
為叢集種子節點建立 AMI。	備份節點，以便在節點故障時使用數據庫二進製文件還原它們。如需指示，請參閱 Amazon EC2 文件中的建立 AMI。	備份管理員
從節點故障中恢復。	將故障節點取代為從 AMI 啟動的新 EC2 執行個體，並連接故障節點的次要網路介面。	備份管理員
驗證卡桑德拉集群是健康的。	取代節點啟動時，請驗證 DSE OpsCenter 中的叢集健全狀況。	DBA

相關資源

- 從 Debian 軟件包中安裝 DSE OpsCenter 6.1（DataStax 文檔）
- 如何讓次要網路界面在 Ubuntu EC2 執行個體中運作 (AWS 知識中心影片)
- 在 Amazon EC2 上運行阿帕奇卡桑德拉的最佳實踐（AWS 博客文章）

使用 AWS 傳輸閘道 Connect 將 VRF 延伸至 AWS

環境：PoC 或試點	技術：基礎架構；網路	AWS 服務：AWS Direct Connect；AWS Transit Gateway

Summary

虛擬路由和轉發（VRF）是傳統網絡的一項功能。它使用隔離的邏輯路由網域 (以路由表的形式) 來分隔相同實體基礎結構內的網路流量。當您將現場部署網路連接到 AWS 時，您可以設定 AWS 傳輸閘道以支援 VRF 隔離。此模式使用範例架構，將內部部署 VRF 連接到不同的傳輸閘道路由表。

此模式在 AWS Direct Connect 和傳輸閘道 Connect 附件中使用傳輸虛擬界面 (VIF) 來擴展 VRF。傳輸 VIF 用於存取與直接 Connect 閘道相關聯的一或多個 Amazon VPC 傳輸閘道。傳輸閘道 Connect 附件會將傳輸閘道與在 VPC 中執行的第三方虛擬應用裝置連接。傳輸閘道 Connect 附件支援通用路由封裝 (GRE) 通道通訊協定以達到高效能，並支援邊界閘道通訊協定 (BGP) 進行動態路由。

此模式中描述的方法具有以下優點：

- 使用「Transit Gateway Connect」，您最多可以通告 1,000 個路由到「Transit Gateway Connect」對等，並從中接收多達 5,000 條路由。不使用「傳輸閘道 Connect」的「直 Connect」傳輸 VIF 功能，每個傳輸閘道限制為 20 個字首。
- 您可以保持流量隔離，並使用 Transit Gateway Connect 在 AWS 上提供託管服務，無論客戶使用的 IP 地址結構描述為何。
- VRF 流量不需要周遊公共虛擬界面。這使得在許多組織中更容易遵守法規遵循和安全性要求。
- 每個 GRE 通道最多支援 5 Gbps，每個交通閘道 Connect 附件最多可以有四個 GRE 通道。這比許多其他連線類型快，例如支援高達 1.25 Gbps 的 AWS Site-to-Site VPN 連線。

先決條件和限制

先決條件

- 已建立必要的 AWS 帳戶 (如需詳細資訊，請參閱架構)
- 在每個帳戶中擔任 AWS Identity and Access Management (IAM) 角色的許可。

- 每個帳戶中的 IAM 角色都必須具有佈建 AWS Transit Gateway 和 AWS Direct Connect 資源的許可。如需詳細資訊，請參閱傳輸閘道的驗證和存取控制，並參閱 Direct Connect 的身分識別與存取管理。
- 「直 Connect 線」連線已成功建立。如需詳細資訊，請參閱使用連線精靈建立連線。

限制

- 生產、QA 和開發帳戶中 VPC 的傳輸閘道附件有限制。如需詳細資訊，請參閱將閘道附件傳輸至 VPC。
- 建立與使用 Direct Connect 閘道均設有限制。如需詳細資訊，請參閱 AWS Direct Connect 配額。

架構

目標架構

下列範例架構提供可重複使用的解決方案，以部署具有傳輸閘道 Connect 附件的傳輸 VIF。這種架構通過使用多個直接 Connect 位置提供恢復性。如需詳細資訊，請參閱直 Connect 說明文件中的最大恢復能力。現場部署網路具有生產、QA 和開發 VRF，這些 VRF 延伸到 AWS，並透過使用專用路由表隔離。

在 AWS 環境中，有兩個帳戶專用於擴充 VRF：直 Connect 帳戶和網路中樞帳戶。直 Connect 線帳戶包含每個路由器的連線和傳輸 VIF。您可以從直 Connect 帳戶建立傳輸 VIF，但將它們部署到網路中樞帳戶，以便您可以將它們與網路中樞帳戶中的直 Connect 閘道建立關聯。網路中樞帳戶包含直接 Connect 閘道和傳輸閘道。AWS 資源的連接方式如下：

1. 傳輸 VIF 使用直接 Connect 帳戶中的 AWS Direct Connect 來 Connect 位置的路由器。
2. 傳輸 VIF 會 Connect「直接連線」與網路中樞帳戶中的「直接連 Connect」閘道連線。
3. 傳輸閘道關聯會將 Direct Connect 閘道與網路中樞帳戶中的傳輸閘道連接起來。
4. 傳輸閘道 Connect 附件將傳輸閘道與生產、QA 和開發帳戶中的 VPC 連接起來。

交通 VIF 架構

下圖顯示傳輸 VIF 的組態詳細資料。此範例架構使用 VLAN 做為通道來源，但您也可以使用迴路。

以下是傳輸 VIF 的組態詳細資料，例如自主系統編號 (ASN)。

資源	項目	Detail
路由器 -01	ASN	65534
路由器 -02	ASN	65534
路由器 -03	ASN	65534
路由器 -04	ASN	65534
Direct Connect 閘道	ASN	64601
Transit Gateway	ASN	64600
	CIDR 區塊	10.100.254.0/24

交通閘道 Connect 架構

下圖和表格說明如何透過傳輸閘道 Connect 附件設定單一 VRF。對於其他 VRF，請在 CIDR 區塊內指派唯一的通道識別碼、傳輸閘道 GRE IP 位址和 BGP。對等 GRE IP 位址與傳輸 VIF 中的路由器對等 IP 位址相符。

下表包含路由器設定詳細資料。

路由器	通道	IP 地址	來源	目的地
路由器 -01	第一隧道	169.254.101.17	VLAN 60	10.100.254.1
			169.254.100.1	
路由器 -02	十一號隧道	169.254.101.81	VLAN 61	10.100.254.11
			169.254.100.5	
路由器 -03	第 21 號隧道	169.254.101.145	VLAN 62	10.100.254.21
			169.254.100.9	

路由器 -04	三十號隧道	169.254.101.209	VLAN 63 169.254.100.13	10.100.254.31

下表包含傳輸閘道組態詳細資訊。

通道	交通閘道 GRE IP 位址	對等 GRE IP 位址	CIDR 區塊內的 BGP
第一隧道	10.100.254.1	VLAN 60 169.254.100.1	169.254.101.16/29
十一號隧道	10.100.254.11	VLAN 61 169.254.100.5	169.254.101.80/29
第 21 號隧道	10.100.254.21	VLAN 62 169.254.100.9	169.254.101.144/29
三十號隧道	10.100.254.31	VLAN 63 169.254.100.13	169.254.101.208/29

部署

Epics 一節說明如何在多個客戶路由器上部署單一 VRF 的範例組態。步驟 1-5 完成後，您可以針對要擴充到 AWS 的每個新 VRF，使用步驟 6-7 建立新的傳輸閘道 Connect 附件：

1. 建立傳輸閘道。
2. 為每個 VRF 建立 Transit Gateway 路由表。
3. 建立傳輸虛擬介面。
4. 建立直 Connect 閘道。
5. 建立 Direct Connect 閘道虛擬介面，並使用允許的前置字元建立閘道關聯。
6. 建立傳輸閘道 Connect 附件。
7. 建立傳 Transit Gateway Connect 對等。
8. 建立傳輸閘道 Connect 線附件與路由表的關聯。

9. 通告路由到路由器。

工具

AWS 服務

- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- AWS Transit Gateway 是連接虛擬私有雲 (VPC) 和現場部署網路的中央中樞。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

史诗

規劃架構

任務	描述	所需技能
建立自訂架構圖。	1. 在「附件」區段中，下載圖表範本。 2. 在 Microsoft 辦公室中打開附加的圖表 PowerPoint。 3. 在架構概觀投影片上，自訂您環境的架構圖。識別需要延伸到 AWS 環境的現場部署 VRF。 4. 在「交通 VIF」幻燈片上，自訂架構圖。識別路由器、直 Connect 閘道和傳輸閘道的 AS 號碼。識別傳輸 VIF 每一端的 IP 位址。 5. 在「Transit Gateway Connect」投影片上，自訂每個 VRF 的架構圖。	雲端架構師、網路管理員

任務	描述	所需技能
	識別設定路由器和 Transit Gateway Connect 對等所需的所有必要 IP 位址。	

建立傳 Transit Gateway 資源

任務	描述	所需技能
建立傳輸閘道。	1. 登入網路中樞帳戶。 2. 依照建立公共交通閘道中的指示操作。請注意此模式的下列事項： • 對於 Amazon 端自主系統編號 (ASN)，請輸入唯一的 ASN。就此範例而言，出貨預先通知是64600。 • 選取 [DNS 支援]。 • 對於此範例架構，不需要 VPN ECMP 支援、預設路由表關聯、預設路由表定期和多點傳送支援。 • 對於傳輸閘道 CIDR 區塊，請輸入傳輸閘道的 IPv4 CIDR 區塊。就此範例而言，CIDR 區塊為10.100.254.0/24 。	網路管理員、雲端架構師
建立交通閘道路由表。	請依照建立交通閘道路由表中的指示進行。請注意此模式的下列事項： • 在「名稱」標籤中，提供傳輸閘道路由表的名稱。我們	雲端架構師、網路管理員

任務	描述	所需技能
	建議使用與 VRF 相對應的名稱，例如`routetable-dev-vrf` 。 • 對於傳輸閘道 ID，請選擇您先前建立的傳輸閘道。	

建立傳輸虛擬介面

任務	描述	所需技能
建立傳輸虛擬介面。	1. 登入直接 Connect 帳戶。 2. 依照建立直 Connect 閘道的傳輸虛擬介面中的指示進行。請注意此模式的下列事項： • 在虛擬介面名稱中，輸入傳輸 VIF 的名稱。我們建議使用與路由器對應的名稱，例如`transit-vif-router01` 。 • 對於「連線」，請選取路由器，例如`router-01` 。 • 若為虛擬介面擁有者，請輸入網路中樞帳戶的帳戶 ID。如需指示，請參閱檢視您的 AWS 帳戶 ID。 • 對於直 Connect 閘道，請勿進行任何選取。您可以在後續步驟中 Connect「直接連線」閘道。	雲端架構師、網路管理員

任務	描述	所需技能
	• 對於 VLAN，請輸入路由器的 VLAN，例如。60 • 對於 BGP ASN，請輸入路由器的出貨預先通知，例如。65534 • 在 Additional settings (其他設定) 之下，執行下列動作： • 選擇 IPv4。 • 對於您的路由器對等 IP，請輸入路由器對等 IP 地址，例如169.254.100.1 。 • 對於 Amazon 路由器對等 IP。輸入 Amazon 路由器對等 IP，如169.254.100.2 。 • 對於 BGP 驗證金鑰，需要密碼。如果保留空白，AWS 會建立只能在此帳戶中存取的金鑰。 3. 重複這些指示，為 VRF 建立所有傳輸 VIF。	

建立直接 Connect 資源

任務	描述	所需技能
建立一個 Direct Connect 閘道。	1. 登入網路中樞帳戶。	雲端架構師、網路管理員

任務	描述	所需技能
	2. 依照建立直 Connect 閘道中的指示進行。請注意此模式的下列事項： • 對於 Amazon 端 ASN，請輸入直 Connect 閘道的 ASN，例如。64601 • 請勿選擇虛擬私有閘道。	
將直接連線閘道 Connect 至傳輸 VIF。	1. 在網路中樞帳戶中，開啟 AWS Direct Connect 主控台，網址為 https://console.aws.amazon.com/directconnect/v2/。 2. 在導覽窗格中，選擇 Virtual Interfaces (虛擬介面)。 3. 選取新的公共交通 VIF，然後選擇「接受」。 4. 選擇您建立的直 Connect 閘道。 5. 對每個運輸 VIF 重複這些指示。	雲端架構師、網路管理員

任務	描述	所需技能
使用允許的前置詞建立「直 Connect」閘道關聯。	在網路中樞帳戶中，遵循關聯傳輸閘道中的指示。請注意此模式的下列事項： • 對於「閘道」，請選擇您先前建立的傳輸閘道。 • 針對「允許的首碼」，輸入指派給傳輸閘道的 CIDR 區塊，例如。**`10.100.254.0/24`** 建立此關聯會自動建立具有「直 Connect 閘道」資源類型的傳輸閘道附件。此附件不需要與運輸閘道路由表相關聯。	雲端架構師、網路管理員

任務	描述	所需技能
建立傳輸閘道 Connect 附件。	1. 在網路中樞帳戶中，開啟 Amazon VPC 主控台，網址為 https://console.aws.amazon.com/vpc/。 2. 在導覽窗格中，選擇傳輸閘道連接。 3. 選擇 Create transit gateway attachment (建立傳輸閘道連接)。 4. 在「名稱」標籤中，輸入附件的名稱。我們建議使用與 VRF 相對應的名稱，例如 PROD-VRF。 5. 對於傳輸閘道 ID，請選擇您先前建立的傳輸閘道。 6. 在 Attachment type (連接類型) 中，選擇 Connect (連線)。 7. 針對傳輸附件 ID，選擇您先前建立的直 Connect 閘道。 8. 選擇 Create transit gateway attachment (建立傳輸閘道連接)。 9. 對要擴展的每個 VRF 重複此步驟。	雲端架構師、網路管理員

任務	描述	所需技能
建立傳 Transit Gateway Connect 對等。	1. 在網路中樞帳戶中，依照建立傳輸閘道 Connect 對等 (GRE 通道) 中的指示進行。請注意此模式的下列事項： • 在「名稱」標籤中，輸入「Transit Gateway Connect」對等的名稱。我們建議使用與路由器對應的名稱，例如`connectpeer-router01` 。 • 對於傳輸閘道 GRE 位址，請從傳輸閘道 CIDR 區塊輸入指派的 IP 位址，例如`10.100.254.1` 。 • 對於對等 GRE 位址，請輸入指派給在路由器上為傳輸 VIF 建立的 VLAN 的 IP 位址，例如。`169.254.100.1` 只要 AWS 可以連線到 IP 位址，您就可以針對對等 GRE 位址使用任何介面，例如 VLAN 或迴路。 • 對於 CIDR 區塊內部的 BGP，請在 CIDR 區塊 IP 位址內輸入 BGP，例如。`169.254.101.16/29` • 針對對等出貨預先通知，輸入路由器的 ASN，例如。`65534`	

任務	描述	所需技能
	2. 重複這些指示，為每個路由器建立 GRE 通道。	

宣傳路由器的路由

任務	描述	所需技能
廣告的路線。	將新的傳輸閘道 Connect 附件與您先前為此 VRF 建立的路由表建立關聯。例如，將生產傳輸閘道 Connect 附件與`Production-VRF` 路由表相關聯。 為通告給路由器的前綴創建靜態路由。 1. 登入網路中樞帳戶。 2. 前往 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。 3. 在導覽窗格的「運輸閘道」下，選擇「運輸閘道」路由表。 4. 選取 `Production-VRF` 路由表。 5. 在 [動作] 功能表上，選擇 [建立靜態路由]。 6. 針對 CIDR，輸入目標 VPC 中傳輸閘道附件的通告路由的 CIDR 區塊，例如。`10.100.1.0/24`	網路管理員、雲端架構師

任務	描述	所需技能
	7. 在 「選擇附件」 中，選擇相關的傳輸閘道「Connect」附件。 8. 選擇 Create static route (建立靜態路由)。	

相關資源

AWS 文件

- 直 Connect 文件
  - 使用直 Connect 閘道
  - 傳輸閘道關聯
  - AWS Direct Connect 虛擬界面
- Transit Gateway 文件
  - 使用運輸閘道
  - 將閘道附件傳輸至直 Connect 閘道
  - Transit Gateway Connect 附件和傳輸閘道 Connect 對等
  - 建立傳輸閘道 Connect 附件

AWS 部落格文章

- 使用 AWS Transit Gateway 連線分段混合網路
- 使用 AWS Transit Gateway 連線來擴充 VRF 並增加 IP 前置詞廣告

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

當 AWS KMS 金鑰的金鑰狀態變更時，取得 Amazon SNS 通知

創建者：舒伯漢姆哈索拉 (AWS)、芳香拉吉傑亞拉揚 (AWS) 和納瓦迪帕雷克 (AWS)

代碼存儲庫：aws-kms-deletion-notification	環境：PoC 或試點	技術：基礎架構；雲端原生 DevOps；安全性、身分識別、合規性
工作負載：所有其他工作	AWS 服務：Amazon EventBridge；AWS KMS；Amazon SNS	

Summary

刪除該金鑰時，與 AWS Key Management Service (AWS KMS) 金鑰相關聯的資料和中繼資料會遺失。刪除是不可逆的，您無法恢復丟失的數據（包括加密的數據）。您可以設定通知系統來提醒您 AWS KMS 金鑰的金鑰狀態變更，以防止資料遺失。

此模式說明如何監控 AWS KMS 金鑰的狀態變更，方法是在 AWS KMS 金鑰的金鑰狀態變更為`Disabled`或時，使用 Amazon EventBridge 和 Amazon 簡單通知服務 (Amazon SNS) 發出自動通知`PendingDeletion`。例如，如果使用者嘗試停用或刪除 AWS KMS 金鑰，您將收到一封電子郵件通知，其中包含有關嘗試狀態變更的詳細資訊。您也可以使用此模式來排程刪除 AWS KMS 金鑰。

先決條件和限制

先決條件

- 具有 AWS 身分和存取管理 (IAM) 使用者的有效 AWS 帳戶
- 一個 AWS KMS 金鑰

架構

技術, 堆

- Amazon EventBridge
- AWS Key Management Service (AWS KMS)

- Amazon Simple Notification Service (Amazon SNS)

目標架構

下圖顯示用於建立自動監控和通知程序的架構，以偵測 AWS KMS 金鑰狀態的任何變更。

該圖顯示以下工作流程：

1. 使用者停用或排程刪除 AWS KMS 金鑰。
2. EventBridge 規則會評估排程`Disabled`或`PendingDeletion`事件。
3. 此 EventBridge 規則會叫用 Amazon SNS 主題。
4. Amazon SNS 會傳送電子郵件通知訊息給使用者。

附註：您可以自訂電子郵件訊息以符合組織的需求。我們建議包括使用 AWS KMS 金鑰之實體的相關資訊。這可協助使用者瞭解刪除 AWS KMS 金鑰的影響。您也可以排程在刪除 AWS KMS 金鑰前一到兩天傳送的提醒電子郵件通知。

自動化和規模

AWS CloudFormation 堆疊會部署所有必要的資源和服務，讓此模式正常運作。您可以在單一帳戶中獨立實作模式，或在 AWS 組織中將 AWS 用 CloudFormation StackSets於多個獨立帳戶或組織單位。

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和 AWS 區域的整個生命週期進行管理。此 CloudFormation 模式的範本描述您想要的所有 AWS 資源，並為您 CloudFormation 佈建和設定這些資源。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。 EventBridge 從您自己的應用程式和 AWS 服務交付即時資料串流，並將該資料路由到 AWS Lambda 等目標。 EventBridge 簡化了構建事件驅動架構的過程。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。

Code

GitHub 監控 AWS KMS 金鑰停用和排程刪除存放庫中提供此模式的程式碼。

史诗

部署 CloudFormation 範本

任務	描述	所需技能
複製儲存庫。	執行下列命令，將 GitHub Monitor AWS KMS 金鑰停用和排程刪除儲存庫複製到本機機器： `git clone https://github.com/aws-samples/aws-kms-deletion-notification`	AWS 管理員、雲端架構師
更新範本的參數。	在程式碼編輯器中，開啟您從儲存庫複製的`Alerting-KMS-Events.yaml` CloudFormation 範本，然後更新下列參數： • 在中`DestinationEmailAddress`，輸入您計劃用來接收 SNS 通知的作用中電子郵件地址。 • 在中`SNSTopicName`，輸入 SNS 主題的名稱。	AWS 管理員、雲端架構師
部署 CloudFormation 範本。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 在瀏覽窗格中，選擇 [建立堆疊]，然後選擇 [使用新資源 (標準)]。	AWS 管理員、雲端架構師

任務	描述	所需技能
	3. 在 [識別資源] 頁面上，選擇 [下一步]。 4. 在 [指定範本] 頁面上，對於 [範本來源]，選取 [上傳範本檔案]。 5. 選擇 [選擇檔案]，從複製的 GitHub 儲存庫中選取`Alerting-KMS-Events.yaml` 檔案，然後選擇 [下一步]。 6. 在堆疊名稱中，輸入您的堆疊名稱。 7. 選擇提交。	

確認訂閱

任務	描述	所需技能
確認訂閱電子郵件。	CloudFormation 範本成功部署後，Amazon SNS 會傳送訂閱確認訊息到您在 CloudFormation 範本中提供的電子郵件地址。 若要接收通知，您必須確認此電子郵件訂閱。如需詳細資訊，請參閱 Amazon SNS 開發人員指南中的確認訂閱。	AWS 管理員、雲端架構師

測試訂閱通知

任務	描述	所需技能
停用 AWS KMS 金鑰。	1. 登入 AWS 管理主控台並開啟 AWS KMS 主控台。 2. 若要變更「地區」，請選擇目前顯示的「區域」名稱，然後選擇您要切換的「區域」。 3. 在導覽窗格中，選擇 Customer managed keys (客戶受管金鑰)。 4. 選取要啟用或停用之 AWS KMS 金鑰的核取方塊。 5. 若要停用 AWS KMS 金鑰，請選擇 [金鑰動作]，然後選擇 [停用]。	AWS 管理員
驗證訂閱。	確認您已收到 Amazon SNS 通知電子郵件。	AWS 管理員

清除資源

任務	描述	所需技能
刪除 CloudFormation 堆疊。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 在導覽窗格中，選擇 Stacks (堆疊)。 3. 選取您先前建立的堆疊，然後選擇 [刪除]。	AWS 管理員

相關資源

- AWS CloudFormation (AWS 文件)
- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- 在 AWS 上建立事件驅動架構 (AWS 研討會工作室文件)
- AWS Key Management Service 最佳實務 (AWS 白皮書)
- AWS Key Management Service 的安全最佳實務 (AWS KMS 開發人員指南)

其他資訊

Amazon SNS 預設會提供傳輸中加密。若要符合安全最佳實務，您也可以使用 AWS KMS 客戶受管金鑰為 Amazon SNS 啟用伺服器端加密。

大型主機現代化： DevOps 在具有微焦點的 AWS 上

創建者容凱文 (AWS)

資料來源：IBM z/OS 大型主機	目標：AWS	R 類型：不適用
環境：PoC 或試點	技術: DevOps; 基礎設施	AWS 服務：Amazon EC2; AWS CloudFormation; AWS CodeBuild; AWS; AWS CodeCommit; AWS CodeDeploy; AWS Systems Manager; AWS CodePipeline

Summary

客戶挑戰

在大型主機硬體上執行核心應用程式的 Organizations，在硬體需要擴充以滿足數位創新的需求時，通常會遇到一些挑戰。這些挑戰包括以下限制。

- 大型主機開發與測試環境無法擴充，因為大型主機硬體元件的不靈活性，以及變更的成本高昂。
- 大型主機開發正面臨技能短缺，因為新開發人員並不熟悉，也對傳統大型主機開發工具不感興趣。現代化技術，例如容器、持續整合/持續交付 (CI/CD) 管線，以及現代化的測試架構並不適用於大型主機開發。

模式結果

為了解決這些挑戰，Amazon Web Services (AWS) 和 AWS 合作夥伴網路 (APN) 合作夥伴 Micro Focus 已經合作建立了這種模式。該解決方案旨在幫助您實現以下結果。

- 改善開發人員生產力 開發人員可以在幾分鐘內獲得新的大型主機開發實例。
- 使用 AWS 雲端建立具有幾乎無限容量的新大型主機測試環境。
- 快速佈建新的大型主機 CI/CD 基礎架構。使用 AWS 和 AWS 系統管理員可在一小時內完 CloudFormation 成 AWS 上的佈建。
- AWS DevOps 工具的原生用於大型主機開發，包括 AWS CodeBuild、AWS CodeCommit CodePipeline CodeDeploy、AWS 和亞馬遜彈性容器登錄 (Amazon ECR)。

- 將傳統的瀑布式開發轉變為大型主機專案的敏捷開發。

技術摘要

在這種模式中，目標堆棧包含以下組件。

邏輯元件	實作方案	描述
源代碼存儲庫	微焦點 AccuRev 服務器 CodeCommit, Amazon ECR	源代碼管理-該解決方案使用兩種類型的源代碼。 • 大型機源代碼，例如 COBOL，JCL 等 • AWS 基礎設施範本和自動化腳本 這兩種類型的源代碼都需要版本控制，但它們在不同的 SCM 中進行管理。部署到大型主機或 Micro Focus 企業伺服器的原始程式碼是在 Micro Focus 伺服器中進行管理。 AccuRev AWS 範本和自動化指令碼在中管理 CodeCommit。Amazon ECR 用於碼頭映像存儲庫。
企業開發者實例	Amazon Elastic Compute Cloud (Amazon EC2)，微焦點企業 Eclipse 開發人員	大型主機開發人員可以使用適用於 Eclipse 的微焦點企業開發人員，在 Amazon EC2 中開發程式碼。這樣就不需要依賴大型主機硬體來撰寫和測試程式碼。
微焦點授權管理	微焦點 License Manager	針對集中式 Micro Focus 授權管理和管理，此解決方案會使用 Micro Focus License Manager 來代管所需的授權。

CI/CD 管道	CodePipeline,, CodeBuild CodeDeploy, 微焦點企業開發人員在一個容器, 微焦點企業測試服務器在一個容器, 微焦點企業服務器	大型主機開發團隊需要 CI/CD 管線來執行程式碼編譯、整合測試和回歸測試。在 AWS 中， CodePipeline 並且 CodeBuild 可以在容器中與 Micro Focus 企業開發人員和企業測試服務器原生合作。

先決條件和限制

先決條件

名稱	描述
py3270	是一個 Python 接口，它是一個 IBM 3270 終端仿真器。它為 x3270 或 s3270 子進程提供了一個 API。
X3270	x3270 是一款適用於 X 視窗系統和視窗的 IBM 3270 終端模擬器。 這可以由開發人員在本地進行單元測試。
機器人框架-大型主機 -3270 圖書館	大型機 3270 是一個基於 py3270 項目的機器人框架庫。
微焦點攝影	Micro Focus Verastream 是一個整合平台，可讓您以行動應用程式、網路應用程式和 SOA 網路服務的測試方式來測試大型主機資產。
微焦點統一功能測試（UFT）安裝程序和許可證	Micro Focus 統一功能測試是為軟體應用程式和環境提供功能和回歸測試自動化的軟體。
微焦點企業伺服器安裝程式和授權	企業服務器為大型主機應用程序提供了運行時環境。
微焦點企業測試伺服器安裝程式和授權	微焦點企業測試伺服器是 IBM 大型主機應用程式測試環境

適用於 Windows 和 Linux 作業系統的伺服器的微焦點 AccuRev 安裝程式和授權，以及微焦點安裝程式和授權 AccuRev	AccuRev 提供源代碼管理（SCM）。該 AccuRev 系統是專為開發一組文件的團隊使用而設計的。
微焦點企業開發 Eclipse 安裝程式、修補程式和授權	企業開發人員為大型主機開發人員提供一個平台，以開發和維護核心大型主機線上和批次應用程式。

限制

- 在 CodeBuild中不支援建置視窗泊塢視窗映像檔。這個回報的問題需要 Windows 核心 /HC 和碼頭群組的支援。解決方法是通過使用 Systems Manager 創建 Docker 映像構建手冊。此模式使用因應措施來構建 Eclipse 和微焦點企業測試服務器容器映像的微焦點企業開發人員。
- Windows 尚未支援虛擬私人雲端 (VPC) 連線，因此該病毒碼不會使用 Micro Focus License Manager 來管理 Micro Focus 企業開發人員和 Micro Focus 企業測試伺服器容器中的授權。CodeBuild

產品版本

- 微焦點企業開發者 5.5 或更新版本
- 微焦點企業測試伺服器 5.5 或更新版本
- 微焦點企業伺服器 5.5 或更新版本
- 微對焦 AccuRev 7.x 或更新版本
- 微焦點企業開發人員和企業測試服務器的 Windows 碼頭基礎映像：微軟/點網絡框架 -4.7.2 運行時
- AccuRev 客戶端的 Linux 碼頭基礎映像：亞馬遜：2

架構

大型主機環境

在傳統的大型主機開發中，開發人員需要使用大型主機硬體來開發和測試程式。它們面臨容量限制，例如，開發/測試環境的每秒限制百萬個指令 (MIPS)，而且必須仰賴大型主機電腦上可用的工具。

在許多組織中，大型主機開發遵循瀑布式開發方法，團隊仰賴長週期來發佈變更。這些發行週期通常比數位產品開發更長。

下圖顯示共用大型主機硬體進行開發的多個大型主機專案。在大型主機硬體中，向外擴充開發和測試環境以進行更多專案的成本很高。

AWS 架構

此模式將大型主機開發延伸到 AWS 雲端。首先，它使用 Micro Focus AccuRev SCM 在 AWS 上託管大型主機原始程式碼。然後，Micro Focus 企業開發人員和 Micro Focus 企業測試伺服器可用於在 AWS 上建立和測試大型主機程式碼。

下列各節說明樣式的三個主要元件。

1。供應鏈

在 AWS 中，該模式使用 Micro Focus AccuRev 為大型主機原始程式碼建立一組 SCM 工作區和版本控制。其以串流為基礎的架構可讓多個團隊進行 parallel 大型主機開發。若要合併變更，請 AccuRev 使用推進概念。若要將該變更新增至其他工作區，請 AccuRev 使用更新概念。

在專案層級，每個團隊都可以在中建立一或多個串流， AccuRev 以追蹤專案層級的變更。這些稱為專案串流。這些專案串流是從相同的父資料流繼承而來。父流用於合併來自不同項目流的更改。

每個專案串流都可以將程式碼提升至 AccuRev，而且會設定促銷後觸發器來啟動 AWS CI/CD 管道。項目流更改的成功構建可以提升到其父流以進行更多回歸測試。

通常情況下，父流被稱為系統集成流。當從專案串流升級至系統整合串流時，促銷後觸發程序會啟動另一個 CI/CD 管線以執行回歸測試。

除了大型主機程式碼之外，此模式還包括 AWS CloudFormation 範本、Systems Manager 自動化文件和指令碼。遵循 infrastructure-as-code 最佳實務，它們在 AWS 中受到版本控制。 CodeCommit

如果您需要將大型主機程式碼同步回大型主機環境以進行部署，Micro Focus 提供企業同步解決方案，該解決方案可將 AccuRev SCM 的程式碼同步回大型主機供應鏈管理。

2. 開發人員和測試環境

在大型組織中，擴充超過一百個甚至超過一千個大型主機開發人員具有挑戰性。為了解決此限制，該模式使用 Amazon EC2 Windows 執行個體進行開發。在實例上，微焦點企業開發人員 Eclipse 工具已安裝。開發人員可以在執行個體本機上執行所有大型主機程式碼測試和偵錯。

AWS Systems Manager 狀態管理員和自動化文件可用來自動化開發人員執行個體佈建。建立開發人員執行個體的平均時間在 15 分鐘內。準備好下列軟體和組態。

- AccuRev 用於簽出並提交源代碼的 Windows 客戶端 AccuRev
- 適用於 Eclipse 的微焦點企業開發人員工具，可在本機撰寫、測試和偵錯大型主機程式碼
- 開源測試框架 Python 行為驅動開發（BDD）測試框架行為，py3270 和 x3270 模擬器，用於創建腳本來測試應用程序
- Docker 開發人員工具，用於構建企業測試服務器 Docker 映像並在企業測試服務器 Docker 容器中測試應用程序

在開發週期中，開發人員使用 EC2 執行個體在本機開發和測試大型主機程式碼。成功測試本地更改後，開發人員將更改升級到 AccuRev 服務器中。

3. CI/CD 管道

在該模式中，CI/CD 管道在部署到生產環境之前用於集成測試和回歸測試。

如 SCM 一節所述， AccuRev 使用兩種類型的串流：專案串流和整合串流。每個流都與 CI/CD 管道連接起來。為了執行 AccuRev 伺服器和 AWS 之間的整合 CodePipeline，該模式使用推廣 AccuRev 後指令碼建立事件以啟動 CI/CD。

例如，當開發人員提升對專案資料流的變更時 AccuRev，它會啟動推廣後指令碼，以便在 AccuRev Server 中執行。然後指令碼會將變更的中繼資料上傳至 Amazon Simple Storage Service (Amazon S3) 儲存貯體，以建立 Amazon S3 事件。此事件將啟動要執行的 CodePipeline 已設定管線。

整合串流及其相關聯管線會使用相同的事件起始機制。

在 CI/CD 管道中，搭配 Micro Focus AccuRev Linux CodePipeline CodeBuild 用戶端容器搭配使用，從串流中檢出最新的 AccuRev 程式碼。然後管道開始使 CodeBuild 用 Micro Focus 企業開發人員 Windows 容器來編譯原始程式碼，並使用 Micro Focus 企業測試伺服器 Windows 容器中 CodeBuild 測試大型主機應用程式。

CI/CD 管道是使用 AWS CloudFormation 範本建立的，而藍圖將用於新專案。透過使用範本，專案在 AWS 中建立新的 CI/CD 管道需要不到一個小時的時間。

為了在 AWS 上擴展您的大型主機測試能力，該模式建立了 Micro Focus DevOps 測試套件、微焦點遠端流和 Micro Focus UFT 伺服器。透過使用現代 DevOps 工具，您可以根據需要在 AWS 上執行任意數量的測試。

下圖顯示了 AWS 上採用 Micro Focus 的大型主機開發環境範例。

目標技術堆疊

本節將詳細介紹樣式中每個元件的架構。

1。源代碼存儲庫 — AccuRev 供應鏈多

微焦點 AccuRev 供應鏈管理設置為管理大型主機源代碼版本。為了獲得高可用性， AccuRev 支援主要和複本模式。在主要節點上執行維護時，操作員可容錯移轉至複本。

為了加快 CI/CD 管道的回應速度，該模式會使用 Amazon E CloudWatch vents 偵測原始程式碼變更並啟動管道。

1. CodePipeline 已設定為使用 Amazon S3 來源。
2. CloudWatch 事件規則設定為從來源 S3 儲存貯體擷取 S3 事件。
3. 「 CloudWatch 事件」規則會將目標設定至管線。
4. AccuRev SCM 設定為在推廣完成後在本機執行推廣後指令碼。
5. AccuRev SCM 會產生包含促銷中繼資料的 XML 檔案，而指令碼會將 XML 檔案上傳至來源 S3 儲存貯體。
6. 上傳之後，來源 S3 儲存貯體會傳送符合 CloudWatch 事件規則的事件，而 CloudWatch 事件規則會啟動執 CodePipeline 行。

當管道運行時，它會啟動一個 CodeBuild 項目，使用 AccuRev Linux 客戶端容器從關聯的流中檢出最新的大型主機代碼。 AccuRev

下圖顯示 AccuRev 伺服器設定。

2. 企業開發者範本

該模式使用 Amazon EC2 範本來簡化開發人員執行個體的建立。通過使用狀態管理器，它可以將軟件和許可證設置一致地應用於 EC2 實例。

Amazon EC2 範本根據其 VPC 內容設定和預設執行個體設定建立，並遵循企業標記要求。透過使用範本，專案團隊可以建立自己的新開發實例。

當開發人員執行個體啟動時，Systems Manager 員會透過與標籤建立關聯，使用狀態管理員來套用自動化。自動化包括下列一般步驟。

1. 安裝 Micro Focus 企業開發者軟體並安裝修補程式。
2. 安裝微焦點 AccuRev 用戶端。
3. 安裝預先設定的指令碼，供開發人員加入 AccuRev 串流。初始化日食工作區。
4. 安裝開發工具，包括 x3270、3270 和泊塢視窗。
5. 設定授權設定以指向 Micro Focus License Manager 負載平衡器。

下圖顯示 Amazon EC2 範本建立的企業開發人員執行個體，其中包含由狀態管理員套用到執行個體的軟體和組態。企業開發人員執行個體會連線到 Micro Focus License Manager 以啟用授權。

3. CI/CD 管道

如 AWS 架構一節所述，在模式中，有專案層級的 CI/CD 管道和系統整合管道。每個大型主機專案團隊都會建立管線或多個 CI/CD 管線，以建置他們在專案中開發的程式。這些項目 CI/CD 管道從關聯 AccuRev 的流檢出源代碼。

在專案團隊中，開發人員會在相關 AccuRev 資料流中宣傳其程式碼。然後，促銷活動會啟動專案管道以建置程式碼並執行和整合測試。

每個 CodeBuild 項目 CI/CD 管道都使用微焦點企業開發人員工具 Amazon ECR 圖像和微焦點企業測試服務器工具 Amazon ECR 圖像的項目。

CodePipeline 並 CodeBuild 用於建立 CI/CD 管線。因為 CodeBuild 沒 CodePipeline 有預付費用或承諾，您只需按實際用量付費。相較於大型主機硬體，AWS 解決方案可大幅縮短硬體佈建前置時間，並降低測試環境的成本。

在現代開發中，使用多種測試方法。例如，測試驅動開發（TDD），BDD 和機器人框架。透過這種模式，開發人員可以使用這些現代工具進行大型主機測試。例如，通過使用 x3270，py3270 和行為 python 測試工具，您可以定義在線應用程序的行為。您也可以在這些 CI/CD 管線中使用建置大型主機 3270 機器人架構。

下圖顯示了團隊流 CI/CD 管道。

下圖顯示了 CodePipeline 在大型機器人 3270 機器人框架中生成的項目 CI/CD 測試報告。

下圖顯示了由 CodePipeline 在 Py3270 和行為 BDD 生成的項目 CI/CD 測試報告。

成功通過項目級測試後，測試的代碼被手動升級到 AccuRev SCM 中的集成流。在團隊對其項目管道的測試覆蓋範圍有信心之後，您可以自動執行此步驟。

當程式碼升級時，系統整合 CI/CD 管線會檢查合併的程式碼，並執行回歸測試。合併的程式碼會從所有 parallel 專案串流推進。

根據測試環境所需的精細晶粒度，客戶可以在不同的環境中擁有更多系統整合的 CI/CD 管線，例如 UAT、預生產。

在模式中，在系統集成管道中使用的工具是微焦點企業測試服務器，微聚焦 UFT 服務器和微焦點 Verastream。所有這些工具都可以部署到 Docker 容器中，並與 CodeBuild.

成功測試大型主機程式之後，成品會透過版本控制存放在 S3 儲存貯體中。

下圖顯示了一個系統集成的 CI/CD 管道。

在系統整合 CI/CD 管線中成功測試成品之後，即可將其升級以進行生產部署。

如果您需要將原始程式碼部署回大型主機，Micro Focus 提供企業同步解決方案，讓原始程式碼從 AccuRev 回到大型主機努力同步。

下圖顯示將成品部署到 Micro Focus 企業伺服器的生產 CI/CD 管線。在此範例中， CodeDeploy 協調已測試的大型主機成品部署到 Micro Focus 企業伺服器中。

除了 CI/CD 管道的架構說明之外，您還可以閱讀 AWS DevOps 部落格文章使用 Micro Focus 企業套件在 AWS 上自動執行數千個大型主機測試，以取得有關在和中測試大型主機應用程式的詳細資訊。CodeBuild CodePipeline如需在 AWS 上進行大型主機測試的最佳實務和詳細資訊，請參閱部落格文章。

工具

工具

AWS 自動化工具

- AWS CloudFormation
- Amazon CloudWatch 活動
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline
- Amazon ECR
- Amazon Simple Storage Service (Amazon S3)
- AWS Secrets Manager
- AWS Systems Manager

Micro Focus 工具

- 日蝕微焦點企業開發人員
- 微焦點企業測試伺服器
- 微焦點企業伺服器（生產部署）
- Micro Focus AccuRev
- 微焦點 License Manager
- 微焦點終端主機整合商
- 微聚焦 UFT 一號

其他工具

- X3270

- py3270
- 機器人框架-大型主機 -3270 圖書館

史诗

建立 AccuRev SCM 基礎架構

任務	描述	所需技能
使用 AWS 部署主要 AccuRev SCM 伺服器 CloudFormation。		AWS CloudFormation
建立 AccuRev 管理員使用者。	登入 AccuRev SCM 伺服器，然後執行 CLI 命令以建立系統管理員使用者。	AccuRev SCM 伺服器管理
建立 AccuRev 串流。	創建 AccuRev 按順序從上層流繼承的流：生產，系統集成，團隊流。	AccuRev 供應鏈管理
建立開發人員 AccuRev 登入帳戶。	使用 AccuRev SCM CLI 命令為大型主機開發人員建立使用 AccuRev 者登入帳戶。	AccuRev 供應鏈管理

建立企業開發人員 Amazon EC2 啟動範本

任務	描述	所需技能
使用 AWS 部署 Amazon EC2 啟動範本 CloudFormation。	使用 AWS CloudFormation 為微型焦點企業開發人員執行個體部署 Amazon EC2 啟動範本。此範本包含 Micro Focus 企業開發人員執行個體的 Systems Manager 員自動化文件。	AWS CloudFormation

任務	描述	所需技能
從 Amazon EC2 範本建立企業開發人員執行個體。		AWS 主控台登入和大型主機開發人員技能

創建微焦點企業開發者工具 Docker 圖像

任務	描述	所需技能
創建微焦點企業開發人員工具 Docker 映像。	使用 Docker 命令和微焦點企業開發人員工具 Docker 文件來創建 Docker 映像。	Docker
在 Amazon ECR 中創建碼頭存儲庫。	在 Amazon ECR 主控台上，為微型焦點企業開發人員泊塢視窗映像建立存放庫。	Amazon ECR
將微焦點企業開發人員工具 Docker 映像推送到 Amazon ECR。	執行 Docker 推送命令，以推送企業開發人員工具 Docker 映像檔，將其儲存在 Amazon ECR 的 Docker 儲存庫中。	Docker

建立微焦點企業測試伺服器 Docker 影像

任務	描述	所需技能
建立微焦點企業測試伺服器 Docker 影像。	使用 Docker 命令和微焦點企業測試伺服器 Docker 檔案來建立 Docker 映像檔。	Docker
在 Amazon ECR 中創建碼頭存儲庫。	在 Amazon ECR 主控台上，為微焦點企業測試伺服器泊塢視窗映像建立 Amazon ECR 儲存庫。	Amazon ECR

任務	描述	所需技能
將微焦點企業測試伺服器泊塢視窗映像推送至 Amazon ECR。	執行 Docker 推送命令，以在 Amazon ECR 中推送並儲存企業測試伺服器泊塢視窗映像檔。	Docker

建立團隊串流 CI/CD 管線

任務	描述	所需技能
建立 AWS CodeCommit 儲存庫。	在 CodeCommit 主控台上，為基礎設施和 AWS CloudFormation 程式碼建立以 Git IT 為基礎的儲存庫。	AWS CodeCommit
將 AWS CloudFormation 範本和自動化程式碼上傳到 CodeCommit 存放庫。	執行 Git push 命令，將 AWS CloudFormation 範本和自動化程式碼上傳到儲存庫。	Git
透過以下方式部署團隊串流 CI/CD 管線。 CloudFormation	使用準備好的 AWS CloudFormation 範本部署團隊串流 CI/CD 管道。	AWS CloudFormation

建立系統整合 CI/CD 管線

任務	描述	所需技能
創建微型聚焦 UFT 泊塢視窗圖像。	使用碼頭命令和微焦點 UFT 碼頭文件來創建微焦點泊塢窗圖像。	Docker
在 Amazon ECR 中為微焦點 UFT 映像創建碼頭存儲庫。	在 Amazon ECR 主控台上，為微型聚焦 UFT 映像檔建立碼頭儲存庫。	Amazon ECR

任務	描述	所需技能
將微焦 UFT 泊塢視窗圖像推送到 Amazon ECR。	執行 Docker 推送命令，以在 Amazon ECR 中推送並儲存企業測試伺服器泊塢視窗映像檔。	Docker
創建微焦點真正的泊塢視窗圖像。	使用碼頭命令和微焦點真正碼頭文件來創建碼頭圖像。	Docker
在 Amazon ECR 中為微焦點真正流圖像創建碼頭存儲庫。	在 Amazon ECR 主控台上，為微型焦點真人影像建立碼頭儲存庫。	Amazon ECR
透過以下方式部署系統整合 CI/CD 管線。 CloudFormation	使用準備好的 AWS CloudFormation 範本部署系統整合 CI/CD 管道。	AWS CloudFormation

建立生產部署 CI/CD 管線

任務	描述	所需技能
使用 AWS 快速入門部署微焦點企業伺服器。	若要使用 AWS 部署 Micro Focus 企業伺服器 CloudFormation，請在 AWS 快速入門啟動 Micro Focus 企業伺服器。	AWS CloudFormation
部署生產部署 CI/CD 管線。	在 AWS 主 CloudFormation 控台上，使用 AWS CloudFormation 範本部署生產部署 CI/CD 管道。	AWS CloudFormation

相關資源

參考

- AWS DevOps 部落格-使用微焦企業套件在 AWS 上自動執行數千個大型主機測試

- 存儲庫 GitHub
- 高傳統 PT-GDC /機器人框架大型主機-3270-庫存儲庫 GitHub
- 歡迎您的行為！
- APN 合作夥伴部落格-標籤：微焦點
- 從啟動範本啟動執行個體

AWS Marketplace

- 微聚焦 UFT 一號

AWS 快速入門

- AWS 上的微焦點企業伺服器

在多帳戶 VPC 設計中保留非工作負載子網路的可路由 IP 空間

由亞當·斯派塞（AWS）創建

代碼存儲庫：不可路由的輔助 CIDR 模式	環境：生產	技術：基礎設施 DevOps; 管理與治理; 網絡
AWS 服務：AWS Transit Gateway；Amazon VPC；Elastic Load Balancing (ELB)		

Summary

Amazon Web Services (AWS) 已發佈最佳實務，建議在虛擬私有雲端 (VPC) 中針對傳輸閘道附件和閘道 Load Balancer 端點 (以支援 AWS Network Firewall 或第三方設備) 使用專用子網路。這些子網路是用來包含這些服務的彈性網路介面。如果您同時使用 AWS Transit Gateway 和閘道 Load Balancer，則會在 VPC 的每個可用區域中建立兩個子網路。由於 VPC 的設計方式，這些額外的子網路不能小於 /28 遮罩，而且可能會消耗寶貴的可路由 IP 空間，否則可用於路由工作負載。此模式示範如何為這些專用子網路使用次要、不可路由的無類別網域間路由 (CIDR) 範圍，以協助保留可路由 IP 空間。

先決條件和限制

前提

- 可路由 IP 空間的多 VPC 策略
- 您使用之服務的不可路由 CIDR 範圍 (傳輸閘道附件和閘道 Load Balancer 或 Network Firewall 端點)

架構

目標架構

此模式包括兩個參考架構：一個架構具有傳輸閘道 (TGW) 附件的子網路和閘道 Load Balancer 端點 (GWLBE)，第二個架構只有 TGW 附件的子網路。

架構 1 – 連接 TGW 的 VPC，具有輸入路由至設備

下圖表示跨越兩個可用區域的 VPC 參考架構。在輸入時，VPC 會使用輸入路由模式，將目的地為公有子網路的流量導向至應用bump-in-the-wire 裝置以進行防火牆檢查。TGW 附件支援從私有子網路輸出至個別 VPC。

此模式對 TGW 附件子網路和 GWLbe 子網路使用不可路由的 CIDR 範圍。在 TGW 路由表中，此不可路由的 CIDR 通過使用一組更具體的路由配置為黑洞（靜態）路由。如果路由要傳播到 TGW 路由表，這些更具體的黑洞路由將適用。

在此範例中，將 /23 可路由 CIDR 分割並完全配置給可傳遞子網路。

架構 2 — 連接 TGW 的 VPC

下圖顯示跨越兩個可用區域的 VPC 的另一個參考架構。TGW 附件支援從私有子網路到個別 VPC 的輸出流量 (輸出)。它僅針對 TGW 附件子網路使用不可路由的 CIDR 範圍。在 TGW 路由表中，此不可路由的 CIDR 使用一組更具體的路由配置為黑洞路由。如果路由要傳播到 TGW 路由表，這些更具體的黑洞路由將適用。

在此範例中，將 /23 可路由 CIDR 分割並完全配置給可傳遞子網路。

工具

AWS 服務和資源

- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。在此模式中，VPC 次要 CIDR 用於保留工作負載 CIDR 中的可路由 IP 空間。
- 網際網路閘道輸入路由 (Edge 關聯) 可與閘道 Load Balancer 端點搭配使用，用於專用的非路由子網路。
- AWS Transit Gateway 是連接 VPC 和現場部署網路的中央中樞。在此模式中，VPC 會集中連接至傳輸閘道，而傳輸閘道附件位於專用的非路由式子網路中。
- Gateway Load Balancer 可讓您部署、擴展和管理虛擬設備，如防火牆、入侵偵測與預防系統，以及深層封包檢查系統。閘道充當所有流量的單一入口和出口點。在此病毒碼中，閘道 Load Balancer 的端點可用於專用的非路由子網路中。
- AWS Network Firewall 是適用於 AWS 雲端中 VPC 的可設定狀態、受管網路防火牆以及入侵偵測與防護服務。在此病毒碼中，防火牆的端點可用於專用的非路由子網路中。

代碼存儲庫

GitHub 不可路由的次要 CIDR CloudFormation 模式存放庫中提供此模式的 Runbook 和 AWS 範本。您可以使用範例檔案在您的環境中設定工作實驗室。

最佳實務

AWS Transit Gateway

- 為每個傳輸閘道 VPC 連接使用個別子網路。
- 從傳輸閘道附件子網路的次要非路由 CIDR 範圍配置 /28 子網路。
- 在每個傳輸閘道路由表中，為不可路由的 CIDR 範圍新增一個靜態、更具體的路由作為黑洞。

閘道 Load Balancer 和輸入路由

- 使用輸入路由將流量從網際網路導向至閘道 Load Balancer 端點。
- 為每個閘道 Load Balancer 端點使用不同的子網路。
- 針對閘道 Load Balancer 端點子網路，從次要非路由 CIDR 範圍配置 /28 子網路。

史诗

建立 VPC

任務	描述	所需技能
決定不可路由的 CIDR 範圍。	判斷將用於傳輸閘道附件子網路和任何閘道 Load Balancer 或 Network Firewall 端點子網路 (選擇性) 的不可路由 CIDR 範圍。此 CIDR 範圍將用作 VPC 的次要 CIDR。它不得從 VPC 的主要 CIDR 範圍或較大的網路進行路由。	雲端架構師
決定 VPC 的可路由 CIDR 範圍。	決定將用於 VPC 的一組可路由 CIDR 範圍。此 CIDR 範圍	雲端架構師

任務	描述	所需技能
	將用作虛擬私人雲端的主要 CIDR。	
建立虛擬私人雲端。	建立 VPC 並將其附加至傳輸閘道。根據您在前兩個步驟中決定的範圍，每個 VPC 都應具有可路由的主要 CIDR 範圍和不可路由的次要 CIDR 範圍。	雲端架構師

設定 Transit Gateway 黑洞路由

任務	描述	所需技能
將更特定的不可路由的 CIDR 建立為黑洞。	每個傳輸閘道路由表都必須為不可路由的 CIDR 建立一組黑洞路由。這些設定是為了確保來自次要 VPC CIDR 的任何流量都保持不可路由，並且不會洩漏到較大的網路中。這些路由應該比 VPC 上設定為次要 CIDR 的不可路由 CIDR 更具體。例如，如果次要不可路由的 CIDR 是 100.64.0.0/26，則傳輸閘道路由表中的黑洞路由應該是 100.64.0.0/27 和 100.64.0.32/27。	雲端架構師

相關資源

- 部署閘道 Load Balancer 的最佳作法
- 閘道 Load Balancer 的分散式檢測架構
- 網絡沉浸日-互聯網到 VPC 防火牆實驗室
- 傳輸閘道設計最佳做法

其他資訊

在處理需要大量 IP 位址的大型擴充容器部署時，不可路由的次要 CIDR 範圍也很有用。您可以將此模式與私有 NAT 閘道搭配使用，以使用不可路由的子網路來裝載容器部署。如需詳細資訊，請參閱部落格文章如何使用私有 NAT 解決方案解決私有 IP 耗盡問題。

使用程式碼儲存庫在 AWS Service Catalog 中佈建 Terraform 產品

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）和泰米爾塞爾文 P（AWS）創建

環境：PoC 或試點	技術：基礎設施; DevOps	工作負載：所有其他工作
AWS 服務：AWS Service Catalog；Amazon EC2		

Summary

AWS Service Catalog 支援針對您的 HashiCorp Terraform 組態進行控管的自助佈建。如果您使用 Terraform，您可以使用 Service Catalog 做為單一工具，在 AWS 中大規模組織、管理和分發您的 Terraform 組態。您可以存取 Service Catalog 主要功能，包括編目標準化和預先核准的基礎設施即程式碼 (IaC) 範本、存取控制、具有最低權限存取的雲端資源佈建、版本控制、共用至數千個 AWS 帳戶以及標記。一般使用者 (例如工程師、資料庫管理員和資料科學家) 會查看他們有權存取的產品和版本清單，而且他們可以透過單一動作進行部署。

此模式可協助您使用 Terraform 程式碼部署 AWS 資源。儲存庫中的 Terraform 程式碼可透過 Service Catalog GitHub 存取。使用這種方法，您可以將產品與現有的 Terraform 工作流程整合。管理員可以使用 Terraform 建立 Service Catalog 產品組合，並將 AWS Launch Wizard 產品新增至其中。

以下是此解決方案的優點：

- 由於 Service Catalog 中具有復原功能，因此如果在部署期間發生任何問題，您可以將產品還原為先前的版本。
- 您可以輕鬆識別產品版本之間的差異。這可協助您解決部署期間的問題。
- 您可以在服務目錄中設定儲存庫連線 GitHub，例如 To GitLab、或 AWS CodeCommit。您可以直接透過儲存庫進行產品變更。

如需 AWS Service Catalog 整體優勢的相關資訊，請參閱什麼是 Service Catalog。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶

- 包含 ZIP 格式的地形表單組態檔案的 BitBucket、或其他儲存庫。 GitHub
- 已安裝 AWS 無伺服器應用程式模型命令列界面 (AWS SAM CLI)。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 去，安裝。
- Python 本 3.9, 安裝. AWS 山姆 CLI 需要這個版本的 Python。
- 寫入和執行 AWS Lambda 函數和許可以存取和管理 Service Catalog 產品和產品組合的許可。

架構

目標技術堆疊

- AWS Service Catalog
- AWS Lambda

目標架構

該圖顯示以下工作流程：

1. 當 Terraform 配置準備就緒時，開發人員會創建一個包含所有地形代碼的 .zip 文件。開發人員會將 .zip 檔案上傳至連線至 Service Catalog 的程式碼儲存庫。
2. 管理員會將 Terraform 產品與 Service Catalog 中的產品組合相關聯。管理員也會建立允許使用者佈建產品的啟動條件約束。
3. 在 Service Catalog 中，最終使用者使用 Terraform 組態啟動 AWS 資源。他們可以選擇要部署的產品版本。

工具

AWS 服務和工具

- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Service Catalog 可協助您集中管理 AWS 核准的 IT 服務目錄。最終使用者可在機構所設的限制範圍內，迅速地只部署自己需要且經核准的 IT 服務。

其他服務

- Go 是 Google 支持的開源編程語言。
- Python 是一種通用的計算機編程語言。

代碼存儲庫

如果您需要可透過 Service Catalog 部署的範例 Terraform 組態，您可以使用 GitHub Amazon Macie 組織安裝使用 Terraform 儲存庫中的組態。不需要使用此儲存庫中的程式碼範例。

最佳實務

- 透過 Service Catalog 啟動產品時，請設定變數值，而不是在 Terraform 組態檔案 (`terraform.tfvars`) 中提供變數值。
- 僅將產品組合的存取權授與特定使用者或管理員。
- 遵循最低權限原則，並授予執行任務所需的最低權限。如需詳細資訊，請參閱 IAM 文件中的授與最低權限和安全性最佳實務。

史诗

設定您的本機工作站

任務	描述	所需技能
(選擇性) 安裝泊塢視窗。	如果您想要在開發環境中執行 AWS Lambda 函數，請安裝 Docker。如需相關說明，請參閱 Docker 文件中的安裝 Docker 引擎。	DevOps 工程師
安裝適用於地形表單的 AWS Service Catalog 引擎。	1. 輸入以下命令以複製適用於 Terraform 儲存庫的 AWS Service Catalog 引擎。 `git clone https://github.com/aws-samples/service-catalo`	DevOps 工程師, AWS 管理員

任務	描述	所需技能
	g-engine-for-terra form-os.git 2. 導覽至複製存放庫的根目錄。 3. 輸入以下命令。這將安裝引擎。 run ./bin/bash/ deploy-tre.sh -r 在自動安裝期間，不會使用在預設設定檔中設定的 AWS 區域。而是在執行此命令時提供「區域」。	

Connect 存 GitHub 儲庫

任務	描述	所需技能
建立與 GitHub 存放庫的連線。	1. 登入 AWS 管理主控台，然後開啟開發人員工具主控台。您可以選擇 AWS CodePipeline、AWS 或 AWS 等服務來存取開發人員工具主控台 CodeDeploy。 CodeCommit 2. 在左側導覽窗格中，選擇 [設定]，然後選擇 [連線]。 3. 選擇建立連線。 4. 選取維護地形原始程式碼的儲存庫。例如，您可以選擇 Bitbucket GitHub、或GitHub 企業伺服器。	AWS 管理員

任務	描述	所需技能
	5. 輸入連線的名稱，然後選擇 [Connect 線]。 6. 出現提示時，請驗證存放庫。 驗證完成後，會建立連線，且狀態會變更為 Active。	

在 Service Catalog 中建立地形產品

任務	描述	所需技能
建立 Service Catalog 產品。	1. 開啟 AWS Service Catalog 主控台。 2. 瀏覽至 [管理] 區段，然後選擇 [產品清單]。 3. 選擇「建立產品」。 4. 在「產品詳細資訊」區段的「建立產品」頁面上，選擇「外部產品類型」。Service Catalog 使用此產品類型來支援 Terraform 社群版產品。 5. 輸入 Service Catalog 產品的名稱和擁有者。 6. 選取 [使用 CodeStar 提供者指定程式碼儲存庫]。 7. 為您的儲存庫輸入下列資訊： • 使用 Connect 線至您的提供者 AWS CodeConnections — 選取您先前建立的連線。	AWS 管理員

任務	描述	所需技能
	• 存放庫 — 選取存放庫。 • 分支 — 選取分支。 • 範本檔案路徑 — 選擇儲存程式碼範本檔案的路徑。檔案名稱的結尾應為`tar.gz`。 8. 在「版本名稱和說明」下，提供有關產品版本的資訊。 9. 選擇「建立產品」。	
建立組合。	1. 開啟 AWS Service Catalog 主控台。 2. 瀏覽至 [管理] 區段，然後選擇 [學檔]。 3. 選擇建立作品集 4. 輸入下列值： • 產品組合名 – `Sample terraform` • 投資組合說明-`Sample portfolio for Terraform configurations` • 擁有者 — 您的聯絡資訊，例如電子郵件 5. 選擇建立。	AWS 管理員

任務	描述	所需技能
將地形產品添加到產品組合中。	1. 開啟 AWS Service Catalog 主控台。 2. 瀏覽至 [管理] 區段，然後選擇 [產品清單]。 3. 選取您先前建立的地形產品。 4. 選擇 [動作]，然後選擇 [新增產品至產品組合]。 5. 選擇`Sample terraform` 投資組合。 6. 選擇新增產品至產品組合。	AWS 管理員
建立存取政策。	1. 開啟 AWS Identity and Access Management (IAM) 主控台。 2. 在導覽窗格上選擇 Policies (政策)。 3. 在內容窗格中，選擇 Create policy (建立政策)。 4. 選擇「JSON」選項。 5. 在此模式的 [其他資訊] 區段中的 [存取] 原則中輸入範例 JSON 原則。 6. 選擇下一步。 7. 在 [檢閱並建立] 頁面的 [原則名稱] 方塊中，輸入`TerraformResourceCreationAndArtifactAccessPolicy` 。 8. 選擇建立政策。	AWS 管理員

任務	描述	所需技能
建立自訂信任原則。	1. 開啟 AWS Identity and Access Management (IAM) 主控台。 2. 在導覽窗格中，選擇 Roles (角色)。 3. 選擇 Create Role (建立角色)。 4. 在 [信任的實體類型] 下，選擇 [自訂信任原則 5. 在 JSON 政策編輯器中，在此模式的其他資訊區段的信任原則中輸入範例 JSON 政策。 6. 選擇下一步。 7. 在 [權限原則] 底下，選擇`TerraformResourceCreationAndArtifactAccessPolicy` 您先前建立的原則。 8. 選擇下一步。 9. 在 [角色詳細資料] 底下的 [角色名稱] 方塊中，輸入`SCLaunch-product` 。 重要事項：角色名稱必須以開頭`SCLaunch`。 10選擇建立角色。	AWS 管理員

任務	描述	所需技能
將啟動限制新增至 Service Catalog 產品。	1. 以具有管理許可的使用者身分登入 AWS 管理主控台。 2. 開啟 AWS Service Catalog 主控台。 3. 在導覽窗格中，選擇「學檔」。 4. 選擇您先前建立的學檔。 5. 在學檔詳細資訊頁面上，選擇「條件約束」標籤，然後選擇「建立條件約束」。 6. 對於「產品」，請選取您先前建立的 Terraform 產品。 7. 在「啟動限制」下，對於「方法」，選擇輸入角色名稱。 8. 在 [角色名稱] 方塊中輸入`SCLaunch-product` 。 9. 選擇建立。	AWS 管理員
授與產品的存取權。	1. 開啟 AWS Service Catalog 主控台。 2. 在導覽窗格中，選擇「學檔」。 3. 選擇您先前建立的學檔。 4. 選擇存取權標籤，然後選擇 [授與存取權]。 5. 選擇 [角色] 索引標籤，然後選取應具有部署此產品存取權的角色。 6. 選擇 Grant access (授與存取權)。	AWS 管理員

任務	描述	所需技能
啟動產品。	1. 以具有部署 Service Catalog 產品許可的使用者身分登入 AWS 管理主控台。 2. 開啟 AWS Service Catalog 主控台。 3. 在導覽窗格中，選擇 [產品]。 4. 選擇您先前建立的產品，然後選擇 [啟動產品]。 5. 輸入產品名稱並定義任何必要的參數。 6. 選擇「啟動產品」。	DevOps 工程師

驗證部署

任務	描述	所需技能
驗證部署。	Service Catalog 佈建工作流程有兩個 AWS Step Functions 狀態機器： • `ManageProvisionedProductStateMachine` — 在佈建新的 Terraform 產品以及更新現有 Terraform 佈建的產品時，Service Catalog 會叫用此狀態機器。 • `TerminateProvisionedProductStateMachine` — Service Catalog 在終止現有 Terraform 佈建的產品時叫用此狀態機器。	DevOps 工程師

任務	描述	所需技能
	您可以檢查ManagePro visionedProductSta teMachine 狀態機器的記錄檔，以確認產品已佈建。 1. 登入 AWS 管理主控台，然後開啟 AWS Step Functions 主控台。 2. 在左側導覽窗格中，選擇 [狀態機器]。 3. 選擇ManagePro visionedProductSta teMachine 。 4. 在「執行」清單中，輸入佈建的產品 ID 以尋找執行項目。 注意：狀態檔案後端值區名稱開頭為sc-terraform-engine-state- 。 5. 驗證帳號中是否已建立所有必要的資源。	

清理基礎架構

任務	描述	所需技能
刪除佈建的產品。	1. 以具有部署 Service Catalog 產品許可的使用者身分登入 AWS 管理主控台。 2. 開啟 AWS Service Catalog 主控台。 3. 在左側導覽中，選擇已佈建的產品。	DevOps 工程師

任務	描述	所需技能
	4. 選取您建立的產品。 5. 在「動作」 清單中，選擇「終止」。 6. 在確認文字方塊中，輸入`terminate` ，然後選擇「終止已佈建的產品」。 7. 重複這些步驟以終止所有已佈建的產品。	

任務	描述	所需技能
移除地形表單的 AWS Service Catalog 引擎。	1. 以具有管理許可的使用者身分登入 AWS 管理主控台。 2. 開啟 Amazon S3 主控台。 3. 在導覽窗格中，選擇 儲存貯體。 4. 選取`sc-terraform-engine-logging-XXXX`值區。 5. 選擇 [清空]。 6. 針對下列值區重複步驟 4 到 5： • `sc-terraform-engine-state-XXXX` • `terraform-engine-bootstrap-XXXX` 7. 開啟 AWS CloudFormation 主控台，然後驗證您位於正確的 AWS 區域。 8. 在左側導覽列中，選擇「堆疊」。 9. 選取`SAM-TRE`，然後選擇 [刪除]。等待，直到堆棧被刪除。 10選取`Bootstrap-TRE` ，然後選擇 [刪除]。等待，直到堆棧被刪除。	AWS 管理員

相關資源

AWS 文件

- 開始使用地形產品

地形文件

- Terraform 安裝
- 地形後端配置
- 地形 AWS 供應商文件

其他資訊

存取政策

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        },
        {
            "Action": [
                "s3:CreateBucket*",
                "s3:DeleteBucket*",
                "s3:Get*",
                "s3:List*",
                "s3:PutBucketTagging"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:ListGroupResources",
                "resource-groups:DeleteGroup",
                "resource-groups:Tag"
            ],
```

```
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

信任政策

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GivePermissionsToServiceCatalog",
            "Effect": "Allow",
            "Principal": {
                "Service": "servicecatalog.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_id:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
TerraformExecutionRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogExternalParameterParserRole*",
```

```
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogTerraformOSParameterParserRole*"
                    ]
                }
            }
        }
    ]
}
```

使用 Amazon SES 使用單一電子郵件地址註冊多個 AWS 帳戶

由喬·沃茲尼亞克（AWS）和舒邦吉維什瓦卡瑪（AWS）創建

代碼存儲庫：GitHub aws-account-factory-email	環境：PoC 或試點	技術：基礎架構、管理與治理、訊息與通訊
AWS 服務：AWS Lambda；Amazon SES；亞馬遜		

Summary

此模式說明如何將真實電子郵件地址與 AWS 帳戶相關聯的電子郵件地址分離。AWS 帳戶需要在帳戶建立時提供唯一的電子郵件地址。在某些組織中，管理 AWS 帳戶的團隊必須承擔與其簡訊團隊管理許多唯一電子郵件地址的負擔。對於管理許多 AWS 帳戶的大型組織來說，這可能很困難。

此模式提供獨特的電子郵件地址自動售貨解決方案，可讓 AWS 帳戶擁有者將一個電子郵件地址與多個 AWS 帳戶建立關 然後，AWS 帳戶擁有者的真實電子郵件地址會與表格中產生的這些電子郵件地址相關聯。該解決方案處理唯一電子郵件帳戶的所有傳入電子郵件，查找每個帳戶的所有者，然後將任何收到的消息轉發給所有者。

先決條件和限制

先決條件

- AWS 帳戶的管理存取權。
- 存取開發環境。我們建議您使用 AWS Cloud9，以免必須自行設定所需的工具和存取金鑰。
- (選用) 熟悉 AWS Cloud Development Kit (AWS CDK) 工作流程和 Python 程式設計語言，可協助您對任何問題進行疑難排解或進行修改。

限制

- 64 個字元的整體出售電子郵件地址長度。如需詳細資訊，請參閱 AWS Organizations API 參考資料CreateAccount中的。

產品版本

- Node.js 版本 12.7.0 或更新版本
- Python 3.9 或更高版本
- Python 軟件包點和虛擬環境
- AWS CDK 版本 2.23.0 或更新版本
- 泊塢視窗 20.10.x 或更新版本

架構

目標技術堆疊

- AWS CloudFormation 堆疊
- AWS Lambda 函數
- 亞馬遜簡單電子郵件地址 (Amazon SES) 規則和規則集
- AWS Identity and Access Management (IAM) 角色和政策
- Amazon Simple Storage Service (Amazon S3) 存儲桶和存儲桶政策
- AWS Key Management Service (AWS KMS) 金鑰和金鑰政策
- Amazon Simple Notification Service (Amazon SNS) 主題和主題政策
- Amazon DynamoDB 資料表

目標架構

此圖顯示了兩個流程：

- 電子郵件地址自動售貨流程：在圖中，電子郵件地址自動售貨流程（下部）通常以帳戶自動售貨解決方案或外部自動化開始，或者是手動調用。在要求中，會呼叫 Lambda 函數，其中包含所需的中繼資料的承載。函數使用此資訊產生唯一的帳戶名稱和電子郵件地址、將其儲存在 DynamoDB 資料庫中，然後將值傳回給呼叫者。然後，這些值就可以用來建立新的 AWS 帳戶 (通常是使用 AWS Organizations)。
- 電子郵件轉發流程：此流程在上圖的上部分中說明。使用電子郵件地址自動售貨流程產生的帳戶電子郵件建立 AWS 帳戶時，AWS 會將各種電子郵件 (例如帳戶註冊確認和定期通知) 傳送到該電子郵件地址。按照此模式中的步驟操作，您可以使用 Amazon SES 設定 AWS 帳戶以接收整個網域的電子郵件。此解決方案設定轉寄規則，允許 Lambda 處理所有內送電子郵件、檢查T0地址是否在

DynamoDB 表格中，然後將訊息轉寄至帳戶擁有者的電子郵件地址。使用此程序可讓帳戶擁有者將多個帳戶與一個電子郵件地址建立關聯。

自動化和規模

此模式使用 AWS CDK 完全自動化部署。該解決方案使用 AWS 受管服務，這些服務可以自動 (或可以設定) 擴展以滿足您的需求。Lambda 函數可能需要額外的組態來滿足您的擴展需求。如需詳細資訊，請參閱 Lambda 文件中的 Lambda 函數擴展。

工具

AWS 服務

- AWS Cloud9 是整合式開發環境 (IDE)，可協助您撰寫程式碼、建置、執行、測試和偵錯軟體。它也可協助您將軟體發行到 AWS 雲端。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Email Service (Amazon SES) 可協助您使用自己的電子郵件地址和網域來傳送和接收電子郵件。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

部署所需的工具

- 透過 AWS CLI 和 IAM 存取您的 AWS 帳戶進行開發環境。 如需詳細資訊，請參閱「相關資源」區段中的連結。我們建議您使用 AWS Cloud9 來簡化設定程序。

- 如果您使用 AWS Cloud9，系統會為您設定下列項目。 如果您選擇不使用 AWS Cloud9，則需要安裝下列項目：
  - 用於設定 AWS CDK 存取登入資料的 AWS CLI。如需詳細資訊，請參閱 AWS CLI 文件。
  - 版 Python 3.9 或更高版本
  - Python 軟件包點和虛擬環境
  - Node.js 版本 12.7.0 或更新版本
  - AWS CDK 版本 2.23.0 或更新版本
  - 泊塢視窗 20.10.x 版或更新版本

Code

此模式的程式碼可在 GitHub AWS 帳戶工廠電子郵件存放庫中取得。

史诗

配置目標部署環境

任務	描述	所需技能
識別或建立 AWS 帳戶。	識別您擁有完整管理存取權限的現有或新 AWS 帳戶，以部署電子郵件解決方案。	AWS 管理員、雲端管理員
設定部署環境。	依照下列步驟設定易於使用的部署環境並設定相依性： 1. 將 AWS Cloud9 的執行個體部署為專用部署環境。如需相關指示，請參閱開始使用 AWS Cloud9。 2. 使用以下命令將 GitHub AWS 帳戶工廠電子郵件存放庫程式碼庫複製到 AWS Cloud9 執行個體： `git clone https://github.com/aws-samp`	AWS 應用程 DevOps式開發人員

任務	描述	所需技能
	`les/aws-account-fa ctory-email` 3. 在`requirements.txt` 檔案中 (位於存放庫的根目錄) 中，更新開頭`aws-cdk-l ib==` 為的行，以符合您環境中執行的 AWS CDK 版本。若要識別版本，請使用`cdk --version` 指令。	

設定已驗證的網域

任務	描述	所需技能
識別和配置網域。	電子郵件轉寄功能需要專用網域。識別並分配您可以透過 Amazon SES 驗證的網域或子網域。在部署電子郵件轉寄解決方案的 AWS 帳戶內，應該可以使用此網域接收傳入的電子郵件。 網域要求： • 網域應該是標準網域或子網域。 • 該域應該是外部 DNS 解析，因為它將用於接收來自組織外部的電子郵件。	雲端管理員、網路管理員、DNS 管理員
驗證網域。	確認識別的網域可用於接受內送電子郵件。 請完成 Amazon SES 文件中的驗證您的網域以接收 Amazon	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	SES 電子郵件中的指示。這需要與負責網域 DNS 記錄的人員或團隊進行協調。	
設定 MX 記錄。	使用指向 AWS 帳戶和區域中 Amazon SES 端點的 MX 記錄來設定您的網域。如需詳細資訊，請參閱 Amazon SES 文件中的發佈接收 Amazon SES 電子郵件的 MX 記錄。	雲端管理員、網路管理員、DNS 管理員

部署電子郵件自動販賣和轉寄解決

任務	描述	所需技能
修改預設值。	編輯cdk.json檔案中的某些預設值 (位於存放庫的根目錄中)，以便解決方案在部署後正確運作。 1. 修改SES_DOMAIN_NAME 值以符合您先前驗證的網域名稱。 2. 修改ADDRESS_FROM 值以包含中的相同網域SES_DOMAIN_NAME 。地址的本機部分應由您的雲端團隊決定。此地址會成為透過解決方案轉寄之每封電子郵件的FROM地址。 3. 修改ADDRESS_ADMIN 值以符合任何不相符之內送郵件將轉寄至的電子郵件地址。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	此值必須是有效且可操作的電子郵件地址。	

任務	描述	所需技能
部署電子郵件自動販賣和轉寄解決方案	1. 創建一 Python 虛擬環境： `python -m venv .venv` 2. 啟動虛 Python 環境： `source .venv/bin/activate` 或者，在視窗平台上，使用： `% .venv\Scripts\activate.bat` 3. 安裝所有 Python 的要求，沒有錯誤： `pip install -r requirements.txt` 4. 合成 CloudFormation 模板： `cdk synth` 確認沒有錯誤，且完整 CloudFormation 範本包含預期的輸出。 5. (選擇性) 如果您是第一次將 AWS CDK 程式碼部署到目前的 AWS 帳戶或區域，請啟動環境。如需詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	`cdk bootstrap aws://AWS-ACCOUNT-NUMBER/REGION` 替換AWS-ACCOUNT-NUMBER REGION和實際值。 6. 部署解決方案： `cdk bootstrap` `cdk deploy` 這些命令應該沒有錯誤地完成。	

任務	描述	所需技能
確認已部署解決方案。	在開始測試之前，請先確認解決方案已成功部署： 1. 開啟 AWS CloudFormation 主控台並尋找包含該名稱的 CloudFormation 堆疊`AwsMailFwdStack` 。 2. 確認此`AwsMailFwdStack` 堆疊具有下列資源： • Lambda 函數 • Amazon SES 規則和規則集 • (IAM) 角色和政策 • Amazon S3 存儲桶和存儲桶政策 • AWS KMS 金鑰和金鑰政策 • Amazon SNS 主題和主題政策 • DynamoDB 表	AWS 應用程式開發人員 DevOps

驗證電子郵件自動販賣和轉發功能如預期般

任務	描述	所需技能
驗證 API 是否正常運作。	在此步驟中，您將測試資料提交至解決方案的 API，並確認解決方案產生預期的輸出，並確認後端作業已如預期般執行。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	使用測試輸入以手動方式執行銷售電子郵件 Lambda 函數。如需範例，請參閱範例檔案。) 對於`OwnerAddress` ，請使用有效的電子郵件地址。API 應傳回帳戶名稱和帳戶電子郵件，其值會如預期般。	
確認電子郵件正在轉寄。	在此步驟中，您會透過系統傳送測試電子郵件，並驗證電子郵件是否已轉寄給預期的收件者。 1. 從最後一步獲取帳戶電子郵件。 2. 將包含測試主題和正文文本的電子郵件發送到此地址。 3. 確認您已透過帳戶擁有者的電子郵件地址收到電子郵件。 4. 請確認您收到的電子郵件`FROM`地址與中的`ADDRESS_FROM` 設定相符`cdk.json`。 5. 確認收到的電子郵件的主旨和內文與原始傳送的郵件相同。	AWS 應用程式開發人員 DevOps

故障診斷

問題	解決方案
系統未如預期轉寄電子郵件。	確認您的設定是否正確：

問題	解決方案
	1. 您應該已經完成網域的 Amazon SES 驗證程序。 2. 您的網域應該使用指向 AWS 帳戶和區域中 Amazon SES 端點的 MX 記錄進行正確設定。如需詳細資訊，請參閱 Amazon SES 文件中的發佈接收 Amazon SES 電子郵件的 MX 記錄。 驗證網域設定後，請依照下列步驟執行： 1. 針對您部署解決方案的帳戶和區域開啟 AWS CloudWatch 主控台，然後在導覽窗格中導覽至 CloudWatch 日誌群組。 2. 搜尋記錄群組清單`SesMailForwardLogGroup` 。 3. 調查此群組中的記錄檔，以查看電子郵件自動販賣和轉寄程序期間是否產生任何錯誤。

問題	解決方案
當您嘗試部署 AWS CDK 堆疊時，您會收到類似下列內容的錯誤： 「模板格式錯誤：無法識別的資源類型」	在大多數情況下，此錯誤訊息表示您目標的區域沒有所有可用的 AWS 服務。如果您使用 AWS Cloud9 部署解決方案，則可能會鎖定與執行 AWS Cloud9 執行個體的區域不同的區域。 注意：根據預設，AWS CDK 會部署到您在 AWS CLI 中設定的區域和帳戶。 可能的解決方案： 1. 透過按區域檢閱 AWS 服務，調查此解決方案所需的所有服務 (請參閱此模式稍早的 Target 技術堆疊部分) 是否位於您目標的 AWS 區域中。 2. 如果您使用 AWS Cloud9 且鎖定與執行 AWS Cloud9 執行個體所在區域不同的區域，請務必在部署解決方案之前設定AWS_DEFAULT_REGION 環境變數或使用 AWS CLI 設定區域。如需詳細資訊，請參閱 AWS CLI 文件中用來設定 AWS CLI 的環境變數。或者，您可以按照 AWS CDK 文件中的環境說明修改儲存庫根目錄中的app.py檔案，以包含硬式編碼的帳戶 ID 和區域。

問題	解決方案
當您部署解決方案時，您會收到錯誤訊息： 「部署失敗:錯誤: AwsMailFwdStack: SSM 參數 /cdk 引導程序 /hnb659f/ 版本未找到。環境是否已被引導？ 請運行 'cdk 引導程序 '」	如果您從未將任何 AWS CDK 資源部署到目標的 AWS 帳戶和區域，則必須先按照錯誤指示執行`cdk bootstrap` 命令。如果您在執行啟動載入命令後繼續收到此錯誤，您可能會嘗試將解決方案部署到與執行 AWS Cloud9 執行個體所在區域不同的區域。 若要解決此問題，請先設定`AWS_DEFAULT_REGION` 環境變數或使用 AWS CLI 設定區域，然後再部署解決方案。或者，您可以按照 AWS CDK 文件中的環境說明修改儲存庫根目錄中的`app.py`檔案，以包含硬式編碼的帳戶 ID 和區域。

相關資源

- 如需安裝 AWS CLI 的說明，請參閱安裝或更新最新版本的 AWS CLI。
- 如需使用 IAM 存取登入資料設定 AWS CLI 的說明，請參閱設定 AWS CLI。
- 如需 AWS CDK 的相關說明，請參閱開始使用 AWS CD K。

其他資訊

成本

部署此解決方案時，AWS 帳戶持有人可能會產生與使用下列服務相關的費用。 了解這些服務的計費方式對您而言非常重要，以便您了解任何潛在的費用。如需定價資訊，請參閱下列頁面：

- Amazon SES 定價
- Amazon S3 定價
- AWS Cloud9 價
- AWS KMS 司定價
- AWS Lambda 定價
- Amazon DynamoDB 定價

在多帳戶 AWS 環境中為混合網路設定 DNS 解析

創建者阿米爾·杜拉尼

環境：生產	技術：基礎架構；網路	AWS 服務：AWS 記憶體；Amazon Route 53；AWS Control Tower

Summary

此模式說明如何使用現場部署網域名稱系統 (DNS) 服務搭配 Amazon Route 53 Resolver 規則和輸出解析器端點進行名稱解析。

DNS 是建立和維護跨網路環境通訊的基礎。 如果您有混合式網路連線環境，您可以共用 DNS 和 Active Directory 等重要網路服務，而不必擔心跨帳戶和虛擬私人雲端 (VPC) 管理分散式環境的操作負擔。此方法可協助您建置及支援跨越大量帳戶的應用程式。例如，如果您有數百或數千個具有混合式連線需求的多區域帳戶，則可以在 AWS 組織內的所有連線環境中安全有效地共用 DNS 服務。

DNS 對於應用程式的所有層級 (Web、應用程式和資料庫) 之間的 IP 網路至關重要。最佳做法是僅授予 DNS 專家團隊完整存取權，以便設定、操作和支援此資源。在混合式連線環境中，您可以使用條件式轉送，繼續將內部部署 DNS 用於源自不同帳戶資源的名稱解析要求。

此模式涵蓋 AWS 多帳戶環境中的混合式 DNS 解析。對於單一帳戶，請參閱在單一帳戶 AWS 環境中為混合網路設定 DNS 解析模式。

先決條件和限制

先決條件

- 以最佳實務為基礎，並使用 AWS 控制塔建置的 AWS 多帳戶環境。下一節中的圖表顯示了這種環境的典型架構。
- 使用 AWS Transit Gateway，可擴展帳戶和 VPC 之間的路由基礎設施。
- 使用 Amazon Route 53 傳出解析器端點和解析器規則。
- 使用 AWS 資源存取管理員 (AWS RAM)，輸出解析器規則的資源共用率。

架構

AWS 多帳戶架構

目標技術堆疊

- 現有的現場部署 DNS 基礎設施，可跨大量 AWS 主體進行輸出名稱解析
- Route 53 解析器規則和輸出解析器端點
- AWS RAM 可與 AWS 組織內外的其他 AWS 主體共用 Route 53 解析器規則

目標架構

下圖說明設定 end-to-end 混合式 DNS 解析的步驟。AWS RAM 可用來共用 Route 53 解析器規則和解析器端點，這些端點是從中央共用服務帳戶設定和管理的。Route 53 解析器端點會針對每個可用區域設定為接收位於內部部署資料中心之資源的輸出名稱解析要求，然後將這些要求轉寄至內部部署 DNS 解析器。內部部署 DNS 解析器會將名稱解析回應傳送至輸出端點，然後將回應轉寄給 VPC 解析器。這些步驟使用主機名稱而非 IP 位址來建立 end-to-end 通訊。

下圖顯示了更詳細的體系結構。

自動化和規模

您可以使用 AWS 範本，透過 AWS 記憶體設定和共用 Route 53 解析器規則。 CloudFormation

工具

AWS 服務

- AWS Control Tower 可協助您按照規範的最佳實務來設定和管理 AWS 多帳戶環境。
- AWS Resource Access Manager (AWS RAM) 可協助您在 AWS 帳戶之間安全地共用資源，以減少營運開銷，並提供可見性和可稽核性。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

其他工具

- 查詢和挖掘是用於查詢 DNS 記錄的實用程序.

史诗

設定解析器端點和規則

任務	描述	所需技能
設定 Route 53 輸出解析程式端點和規則。	1. 針對您要設定的 AWS 帳戶登入 AWS 管理主控台，並從中共用 Route 53 對外解析器規則。 2. 請在 https://console.aws.amazon.com/route53/ 開啟 Route 53 主控台。 3. 在導覽列上，選擇您要設定解析器端點的區域。 4. 在瀏覽窗格中，選擇「輸出端點」，然後選擇「設定端點」。 5. 提供一般設定、IP 位址和選用的標籤資訊，然後選擇 [下一步]。 6. 建立一或多個規則以指定要轉寄至網路之 DNS 查詢的網域名稱，然後選擇 [儲存]。 如需詳細資訊，請參閱 Route 53 說明文件中的將輸出 DNS 查詢轉寄至您的網路。	一般 AWS
建立 Route 53 對外解析程式規則並與 AWS 主體共用。	1. 開啟 AWS 記憶體主控台，網址為 https://c	一般 AWS

任務	描述	所需技能
	onsole.aws.amazon.com/ram/。 2. 在瀏覽窗格中，選擇 [資源共用]，然後選擇 [建立資源共用]。 3. 提供共用名稱。 4. 對於資源類型，請選擇「解析器規則」。 5. 選擇您要共用的 [解析器] 規則，提供選擇性的標籤索引鍵和值資訊，然後選擇 [下一步]。 6. 選擇您要與其共用解析器規則資源的主參與者。主體可以是 AWS 組織的內部或外部。例如，您可以選擇 AWS 組織、組織內的特定組織單位 (OU) 或特定帳戶。 7. 檢閱並建立資源共用。 建立並共用資源之後，資源會顯示在與其共用主參與者之導覽窗格的 [與我共用] 區段中。 8. 將 (主體) 帳戶中的 VPC 與共用服務或網路帳戶共用的解析器規則建立關聯。 如需詳細資訊，請參閱 AWS RAM 文件中的共用 AWS 資源。	

任務	描述	所需技能
測試輸出 DNS 名稱解析。	在您與之共用解析器規則的帳戶中 VPC 中的執行個體上使用 nslookup 或 dig 公用程式來測試名稱解析。 查詢應解析為位於內部部署資料中心內之資源的 IP 位址。	一般 AWS

相關資源

- 解析混合式環境中的內部部署 DNS (影片)
- 將輸出 DNS 查詢轉送至您的網路 (Route 53 說明文件)
- 共用您的 AWS 資源 (AWS 記憶體文件)

在單一帳戶 AWS 環境中為混合網路設定 DNS 解析

創建者：阿卜杜拉希·奧拉耶 (AWS)

環境：生產	技術：基礎設施	AWS 服務：Amazon Route 53；Amazon VPC

Summary

此模式說明如何設定完全混合式網域名稱系統 (DNS) 架構，以便對內部部署資源、AWS 資源和網際網路 DNS 查詢進行 DNS 解析，而不需要管理負擔。 end-to-end 該模式描述如何設定 Amazon Route 53 Resolver 轉送規則，以根據網域名稱決定應將來自 AWS 的 DNS 查詢傳送到何處。內部部署資源的 DNS 查詢會轉寄至內部部署 DNS 解析器。AWS 資源和網際網路 DNS 查詢的 DNS 查詢可透過路由 53 解析器解析。

此模式涵蓋 AWS 單一帳戶環境中的混合式 DNS 解析。如需在 AWS 多帳戶環境中設定輸出 DNS 查詢的相關資訊，請參閱在多帳戶 AWS 環境中為混合網路設定 DNS 解析模式。

先決條件和限制

先決條件

- 一個 AWS 帳戶
- AWS 帳戶中的虛擬私有雲 (VPC)
- 透過 AWS 虛擬私有網路 (AWS VPN) 或 AWS Direct Connect，在現場部署環境與 VPC 之間建立網路連線
- 內部部署 DNS 解析器的 IP 位址 (可從您的 VPC 存取)
- 轉發至內部部署解析器的網域/子網域名稱 (例如，onprem.mydc.com)
- AWS 私有託管區域的網域/子網域名稱 (例如，myvpc.cloud.com)

架構

目標技術堆疊

- Amazon Route 53 私人託管區域
- Amazon Route 53 Resolver
- Amazon VPC
- AWS VPN 或直接 Connect

目標架構

工具

- Amazon Route 53 Resolver 透過在整個混合雲中實現無縫 DNS 查詢解析，讓企業客戶更輕鬆地進行混合雲。您可以建立 DNS 端點和條件式轉送規則，以解析內部部署資料中心與 VPC 之間的 DNS 命名空間。
- Amazon Route 53 私有託管區域是一個容器，其中包含有關您希望 Route 53 如何在您使用 Amazon VPC 服務建立的一或多個 VPC 內回應網域及其子網域的 DNS 查詢的相關資訊。

史诗

設定私有託管區域

任務	描述	所需技能
為 AWS 保留的網域名稱 (例如 myvpc.cloud.com) 建立 Route 53 私有託管區域。	此區域保留應從現場部署環境解析的 AWS 資源的 DNS 記錄。如需指示，請參閱 Route 53 說明文件中的建立私有託管區域。	網路管理員, 系統管理員
將私有託管區域與您的 VPC 建立關聯。	若要讓 VPC 中的資源能夠解析此私有託管區域中的 DNS 記錄，您必須將 VPC 與託管區域建立關聯。如需指示，請參閱 Route 53 說明文件中的建立私有託管區域。	網路管理員, 系統管理員

設定 Route 53 解析器端點

任務	描述	所需技能
建立入站端點。	Route 53 解析器使用輸入端點接收來自內部部署 DNS 解析器的 DNS 查詢。如需指示，請參閱 Route 53 說明文件中的將輸入 DNS 查詢轉寄至您的 VPC。記下輸入端點 IP 位址。	網路管理員, 系統管理員
建立輸出端點。	Route 53 解析程式會使用輸出端點將 DNS 查詢傳送至內部部署 DNS 解析器。如需指示，請參閱 Route 53 說明文件中的將輸出 DNS 查詢轉寄至您的網路。記下輸出端點 ID。	網路管理員, 系統管理員

設定轉寄規則並將其與您的 VPC 建立關聯

任務	描述	所需技能
建立內部部署網域的轉寄規則。	此規則會指示 Route 53 解析器將內部部署網域 (例如 onprem.mydc.com) 的任何 DNS 查詢轉送至內部部署 DNS 解析器。若要建立此規則，您需要內部部署 DNS 解析器的 IP 位址，以及 Route 53 解析器的輸出端點識別碼。如需指示，請參閱 Route 53 說明文件中的管理轉送規則。	網路管理員, 系統管理員
將轉送規則與您的 VPC 建立關聯。	若要讓轉送規則生效，您必須將規則與 VPC 產生關聯。然後，Route 53 解析器在解析域時考慮規則。如需指示，請參	網路管理員, 系統管理員

任務	描述	所需技能
	閱 Route 53 說明文件中的管理轉送規則。	

設定內部部署 DNS 解析器

任務	描述	所需技能
在內部部署 DNS 解析器中設定條件式轉送。	若要從內部部署環境傳送至 Route 53 私人託管區域的 DNS 查詢，您必須在內部部署 DNS 解析器中設定條件式轉送。這會指示 DNS 解析器將 AWS 網域的所有 DNS 查詢 (例如，針對 myvpc.cloud.com) 轉寄至 Route 53 解析器的入埠端點 IP 位址。	網路管理員, 系統管理員

測試 end-to-end DNS 解析度

任務	描述	所需技能
測試從 AWS 到現場部署環境的 DNS 解析。	從 VPC 中的伺服器，針對內部部署網域 (例如伺服器 1.onprem.mydc.com) 執行 DNS 查詢。	網路管理員, 系統管理員
測試從現場部署環境到 AWS 的 DNS 解析。	從現場部署伺服器執行 AWS 網域的 DNS 解析 (例如伺服器 1.myvpc.cloud.com)。	網路管理員, 系統管理員

相關資源

- 使用 Amazon Route 53 和 AWS 傳輸閘道 (AWS 聯網和內容交付部落格) 進行混合雲的集中式 DNS 管理
- 使用 Route 53 解析器 (AWS 安全部落格)，在多帳戶環境中簡化 DNS 管理
- 使用私有託管區域 (Route 53 文件)
- 開始使用 Route 53 解析器 (R oute 53 文件)

使用 AWS 在 Amazon EC2 上自動設定 UiPath RPA 機器人 CloudFormation

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）和泰米爾塞爾文 P（AWS）創建

環境：PoC 或試點	技術：基礎設施; DevOps	工作負載：所有其他工作
AWS 服務：Amazon CloudWatch；Amazon EC2 Image Builder；AWS Systems Manager；AWS CloudFormation		

Summary

此模式說明如何在 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上部署機器人程序自動化 (RPA) 機器人。它使用 EC2 Image Builder 管道來創建自定義 Amazon 機器映像（AMI）。AMI 是預先設定的虛擬機器 (VM) 映像，其中包含用於部署 EC2 執行個體的作業系統 (OS) 和預先安裝的軟體。此模式使用 AWS CloudFormation 範本在自訂 AMI 上安裝UiPath 工作室社群版本。 UiPath 是一種 RPA 工具，可幫助您設置機器人以自動化您的任務。

作為此解決方案的一部分，EC2 Windows 執行個體是透過使用基礎 AMI 啟動，而 UiPath Studio 應用程式則安裝在執行個體上。此病毒碼會使用 Microsoft 系統準備工具 (Sysprep) 工具來複製自訂的 Windows 安裝。之後，它會移除主機資訊，並從執行個體建立最終 AMI。然後，您可以使用最終 AMI 搭配您自己的命名慣例和監視設定，視需求啟動執行個體。

注意：此模式不提供有關使用 RPA 機器人的任何信息。如需相關資訊，請參閱UiPath 文件。您也可以根據您的需求自訂安裝步驟，使用此模式來設定其他 RPA 機器人應用程式。

此模式提供下列自動化功能和優點：

- 應用程式部署和共用：您可以建立用於應用程式部署的 Amazon EC2 AMI，並透過 EC2 Image Builder 管道在多個帳戶之間共用這些管道，該管道使用 AWS CloudFormation 範本做為基礎設施即程式碼 (IaC) 指令碼。

- Amazon EC2 佈建和擴展： CloudFormation iAC 範本提供自訂電腦名稱順序和使用中目錄加入自動化。
- 可觀察性和監控：該模式設定了 Amazon CloudWatch 儀表板以協助您監控 Amazon EC2 指標 (例如 CPU 和磁碟使用量)。
- RPA 為您的業務帶來的好處：RPA 提高了準確性，因為機器人可以自動且一致地執行指派的任務。RPA 還可以提高速度和生產力，因為它消除了不會增加價值的操作並處理重複的活動。

先決條件和限制

前提

- 有效的 AWS 帳戶
- 用於部署 CloudFormation 範本的 AWS Identity and Access Management (IAM) 許可
- 使用 EC2 Image Builder 設定跨帳戶 AMI 分發的 IAM 政策

架構

1. 系統管理員會在`ec2-image-builder.yaml`檔案中提供基礎 Windows AMI，並在 CloudFormation 主控台中部署堆疊。
2. CloudFormation 堆疊會部署 EC2 Image Builder 管道，其中包括下列資源：
   - `Ec2ImageInfraConfiguration`
   - `Ec2ImageComponent`
   - `Ec2ImageRecipe`
   - `Ec2AMI`
3. EC2 Image Builder 管道會使用基礎 AMI 啟動暫時的 Windows EC2 執行個體，並安裝必要的元件 (在本例中為 UiPath Studio)。
4. EC2 Image Builder 會移除所有主機資訊，並從 Windows 伺服器建立 AMI。
5. 您可以使用自訂 AMI 更新`ec2-provisioning yaml`檔案，並根據您的需求啟動多個 EC2 執行個體。
6. 您可以使用 CloudFormation 範本部署計數巨集。此巨集為 CloudFormation 資源提供 Count 屬性，因此您可以輕鬆地指定相同類型的多個資源。
7. 您可以更新 CloudFormation `ec2-provisioning.yaml`檔案中巨集的名稱並部署堆疊。

8. 系統管理員會根據需求更新`ec2-provisioning.yaml`檔案並啟動堆疊。
9. 範本會透過 UiPath Studio 應用程式部署 EC2 執行個體。

工具

AWS 服務

- AWS 可 CloudFormation協助您以自動化且安全的方式建模和管理基礎設施資源。
- Amazon 可 CloudWatch協助您在 AWS、現場部署和其他雲端上觀察和監控資源和應用程式。
- 亞馬遜彈性運算雲端 (Amazon EC2) 在 AWS 雲端提供安全且可調整大小的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- EC2 Image Builder 可簡化虛擬機器和容器映像的建置、測試和部署，以便在 AWS 或現場部署使用。
- Amazon 可 EventBridge協助您跨 AWS、現有系統或軟體即服務 (SaaS) 應用程式大規模建置事件驅動的應用程式。
- AWS Identity and Access Management (IAM) 可協助您安全地控制 AWS 資源的存取。使用 IAM，您可以集中管理許可，以控制使用者可以存取的 AWS 資源。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- AWS Lambda 是一種無伺服器、事件驅動的運算服務，可讓您針對幾乎任何類型的應用程式或後端服務執行程式碼，而無需佈建或管理伺服器。您可以從 200 多個 AWS 服務和 SaaS 應用程式呼叫 Lambda 函數，而且只需按使用量付費。
- Amazon Simple Storage Service (Amazon S3) 是雲端物件儲存服務，可協助您存放、保護和擷取任意數量的資料。
- AWS Systems Manager 代理程式 (SSM 代理程式) 可協助系統管理員更新、管理和設定 EC2 執行個體、邊緣裝置、現場部署伺服器和虛擬機器 (VM)。

代碼存儲庫

此模式的程式碼可在 GitHub UiPath RPA 機器人設定中使用 CloudFormation儲存庫。該模式還使用 AWS CloudFormation Macros 存儲庫中提供的宏。

最佳實務

- AWS 每個月都會推出新的視窗 AMI。其中包含最新的 OS 修補程式、驅動程式和啟動代理程式。建議您在啟動新執行個體或建立自訂映像檔時使用最新的 AMI。

- 在映像檔建置期間套用所有可用的 Windows 或 Linux 安全性修補程式

史诗

為基礎映像部署映像管道

任務	描述	所需技能
設定 EC2 Image Builder 管道。	1. 使用 CloudFormation儲存庫複製 UiPath RPA 機器人設定，或從存放庫下載`ec2-image-builder.yaml` 範本。 2. 登入 AWS 管理主控台，然後開啟 AWS 主 CloudFormation 控台。 3. 選擇建立堆疊。 4. 在 Specify template (指定範本) 區段中，選擇 Upload a template file (上傳範本檔案)。 5. 從您的電腦找出並上傳`ec2-image-builder.yaml` 範本，然後選擇 [下一步]。 6. 為您的堆棧提供輸入參數或接受默認值。選擇下一步。 注意：參數的數量和值可能會根據您的輸入值而有所不同。 7. 選擇性地設定堆疊選項，然後選擇 [下一步]。 8. 檢閱您的堆疊詳細資料。	AWS DevOps

任務	描述	所需技能
	9. 在畫面結尾，選取確認權能的核取方塊，然後選擇 [提交]。 10監視堆疊的進度。狀態為時CREATE_COMPLETE ，表示部署就緒。	
檢視 EC2 Image Builder 設定。	EC2 Image Builder 設定包括基礎設施組態、分發設定和安全掃描設定。若要檢視設定： 1. 開啟 EC2 Image Builder 主控台。 2. 從導覽窗格中，導覽至各種 Image Builder 設定。 注意：最佳做法是，您應該只透過 CloudFormation 範本對 EC2 Image Builder 進行任何更新。	AWS DevOps
檢視影像管線。	若要檢視已部署的映像管線： 1. 在 EC2 Image Builder 主控台上，從導覽窗格中選擇映像管道。 2. 選取您建立的影像管線。 3. 檢視輸出影像、映像配方、基礎設施組態、分發設定、Amazon EventBridge 規則和標籤的組態詳細資料。	AWS DevOps

任務	描述	所需技能
檢視 Image Builder 記錄。	EC2 Image Builder 日誌會彙總在日 CloudWatch 誌群組中。若要檢視中的記錄檔 CloudWatch： 1. 開啟 CloudWatch 主控台。 2. 在導覽窗格中依序選擇 Logs (日誌)、Log groups (日誌群組)。 3. 選擇記錄群組名稱。EC2 Image Builder 日誌會彙總到日誌群組中`/aws/imagebuilder/XXX` 。 4. 檢查個別記錄資料流中的最新記錄檔，以瞭解執行映像管線時遇到的任何錯誤。 EC2 Image Builder 日誌也存放在 S3 儲存貯體中。若要檢視值區中的記錄： 1. 開啟 Amazon S3 主控台。 2. 在 Buckets (儲存貯體) 清單中，選擇您的儲存貯體名稱。日誌會彙總到 S3 儲存貯體中`<stack-name>-XXXXXX` 。	AWS DevOps

任務	描述	所需技能
將 UiPath 檔案上傳到 S3 儲存貯體。	1. 從以下位置下載工 UiPath 作室的.msi檔案 https://download.uipath.com/UiPathStudioCommunity .msi。 2. 上傳 至 S3 儲存貯體。 3. 在使用者資料區段的行號 310 中，更新ec2-image-builder.yaml 範本中的值區名稱和檔案金鑰。	AWS DevOps

部署和測試計數巨集

任務	描述	所需技能
部署計數巨集。	1. 克隆或下載計數 CloudFormation 宏。 2. 導覽至 Count 資料夾。 3. 您將需要 S3 儲存貯體來存放 CloudFormation 成品。如果您還沒有 S3 儲存貯體，請使用名稱建立一個儲存貯體aws s3 mb s3://<bucket name> 。 4. Package「計數」巨集範本。範本使用 AWS 無伺服器應用程式模型 (SAM)，因此必須先對其進行轉換，然後才能部署。 `aws cloudformation package \ --template-file template.yaml \`	DevOps 工程師

任務	描述	所需技能
	--s3-bucket <your bucket name here> \ --output-template-file packaged.yaml	
	例如： aws cloudformation package \ --template-file template.yaml \ --s3-bucket count-macro-ec2 \ --output-template-file packaged.yaml	
	5. 部署封裝範本以建立 CloudFormation 堆疊。 aws cloudformation deploy \ --stack-name Count-macro \ --template-file packaged.yaml \ --capabilities CAPABILITY_IAM	
	如果您想使用控制台，請按照上一篇史詩或CloudFormation 文檔中的說明進行操作。	

任務	描述	所需技能
測試「計數」巨集。	若要測試巨集的功能，請嘗試啟動巨集隨附的範例範本。 `aws cloudformation deploy \ --stack-name Count-test \ --template-file test.yaml \ --capabilities CAPABILITY_IAM`	DevOps 工程師

部署 CloudFormation 堆疊以使用自訂映像檔佈建執行個體

任務	描述	所需技能
部署 Amazon EC2 佈建範本。	若要使用下列方式部署 EC2 映像管道 CloudFormation： 1. 從GitHub 存放庫下載`ec2-provisioning.yaml` 範本，或在複製存放庫時在您的電腦上尋找範本。 2. 開啟 CloudFormation 主控台。 3. 重複第一個 Epic 中的步驟（或按照CloudFormation 文檔中的說明）進行部署`ec2-provisioning.yaml` 。	AWS DevOps
查看 Amazon EC2 設置。	Amazon EC2 設定包括安全性、聯網、儲存、狀態檢查、監控和標記組態。若要檢視這些組態：	AWS DevOps

任務	描述	所需技能
	1. 開啟 Amazon EC2 主控台。 2. 在導覽窗格中，選擇「執行個體」，然後選取由 Amazon EC2 佈建範本建立的 EC2 執行個體。 3. 在執行個體摘要中，選取索引標籤以檢視對應的 Amazon EC2 設定。	
檢視 CloudWatch 儀表板。	1. 開啟 CloudWatch 主控台。 2. 在導覽窗格中，選擇 Dashboards (儀表板)。 3. 選擇具有您的堆疊名稱的儀表板。 附註：佈建堆疊之後，需要花費一些時間將指標填入儀表板。 儀表板提供以下指標：`CPUUtilization` `DiskUtilization` 、`MemoryUtilization` 、`NetworkIn` 、、`NetworkOut` 、`StatusCheckFailed` 。	AWS DevOps

任務	描述	所需技能
檢視記憶體和磁碟使用量的自訂指標。	1. 在CloudWatch 主控台上，選擇 [儀表板]。 2. 在導覽窗格中，選擇 Metrics (指標)、All metrics (所有指標)。 3. 選擇 [自訂命名空間]、[CW Agent]。	AWS DevOps
檢視記憶體和磁碟使用狀況的警示。	1. 在CloudWatch 主控台的導覽窗格中，選擇 [儀表板]。 2. 選擇 所有警示 。	AWS DevOps
驗證快照生命週期規則。	1. 開啟 Amazon EC2 主控台。 2. 在導覽窗格中，選擇 Lifecycle Manager (生命週期管理器)。 3. 驗證 AMI 生命週期的設定。	AWS DevOps

刪除環境（可選）

任務	描述	所需技能
刪除堆疊。	當 PoC 或試驗專案完成時，我們建議您刪除建立的堆疊，以確保您不需要支付這些資源的費用。 1. 開啟 AWS CloudFormation 主控台。 2. 在瀏覽窗格中，選擇「堆疊」，然後選取您先前建立的一個或兩個您要刪除的堆疊。此堆疊目前必須正在執行。	AWS DevOps

任務	描述	所需技能
	3. 在 stack details (堆疊詳細資訊) 窗格中，選擇 Delete (刪除)。 4. 出現提示時，選擇 Delete stack (刪除堆疊)。 重要事項：堆疊刪除作業在開始後無法停止。堆疊繼續進行到 DELETE_IN_PROGRESS (正在刪除) 狀態。 如果刪除失敗，堆棧將處於狀DELETE_FAILED 態。如需解決方案，請參閱 AWS CloudFormation 疑難排解文件中的刪除堆疊失敗。 如需有關防止堆疊遭到意外刪除的資訊，請參閱 AWS CloudFormation 文件中的防止堆疊遭到刪除。	

故障診斷

問題	解決方案
部署 Amazon EC2 佈建範本時，您會收到錯誤訊息：從轉換 123xxxx:: 計數收到格式錯誤的回應。	這是已知問題。請參閱 AWS CloudFormation 巨集儲存庫中的自訂解決方案和 PR。) 若要修正此問題，請開啟 AWS Lambda 主控台並index.py使用GitHub 儲存庫中的內容進行更新。

相關資源

GitHub 儲存庫

- UiPath RPA 機器人設定使用 CloudFormation
- 計數 CloudFormation 巨集

AWS 參考資料

- 在 AWS CloudFormation 主控台上建立堆疊 (CloudFormation 文件)
- 疑難排解 CloudFormation (CloudFormation 文件)
- 監控 Amazon EC2 執行個體的記憶體和磁碟指標 (亞馬遜 EC2 文件)
- 如何使用 CloudWatch 代理程式來檢視 Windows 伺服器上效能監視器的度量？ （AWS RE：發布文章）

其他參考

- UiPath 文件
- 在 SysPreped AMI 中設置主機名（布萊恩海灘的博客文章）
- 如何在參數更改時使用宏重新處理模板？ (堆疊溢位)

EnterpriseOne 使用 AWS 彈性災難復原為 Oracle JD 愛德華設定災難復原

創建者：坦尼蓋維爾蒂魯馬萊 (AWS)

環境：生產	技術：基礎架構、移轉、網路	工作量：甲骨文
AWS 服務：AWS 彈性災難復原；Amazon EC2		

Summary

自然災難、應用程式故障或服務中斷所引發的災難會損害收入，並導致企業應用程式停機。為了減少此類事件的影響，規劃災難復原 (DR) 對於採用 JD Edwards EnterpriseOne 企業資源規劃 (ERP) 系統和其他關鍵任務和關鍵業務軟體的公司來說至關重要。

此模式說明企業如何將 AWS 彈性災難復原作為 JD Edwards EnterpriseOne 應用程式的 DR 選項。同時也概述使用彈性災難復原容錯移轉和容錯回復，為 AWS 雲端中 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上託管的資料庫建構跨區域 DR 策略的步驟。

注意：此模式要求跨區域 DR 實作的主要和次要區域必須託管在 AWS 上。

Oracle JD 愛德華 EnterpriseOne是一個集成的 ERP 軟件解決方案，適用於各行各業的中型到大型公司。

AWS Elastic 災難復原使用經濟實惠的儲存裝置、最少運算和復原，快速可靠地 point-in-time 復原現場部署和雲端應用程式，將停機時間和資料遺失降到最低

AWS 提供四種核心 DR 架構模式。本文件著重於使用導光策略的設定、配置和最佳化。此策略可協助您建立成本較低的 DR 環境，在此環境中，您一開始佈建複製伺服器以從來源資料庫複製資料，而且只有在啟動 DR 鑽研與復原時，才佈建實際的資料庫伺服器。此策略可免除在 DR 區域中維護資料庫伺服器的費用。相反，您需要支付用作複寫伺服器的較小 EC2 執行個體。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在 Oracle 資料庫或 Microsoft SQL Server 上執行的 JD 愛德華 EnterpriseOne 應用程式，且受管 EC2 執行個體上處於執行狀態的受支援資料庫。此應用程式應包含安裝在一個 AWS 區域中的所有 JD Edwards EnterpriseOne 基礎元件 (企業伺服器、HTML 伺服器和資料庫伺服器)。
- 用於設定彈性災難復原服務的 AWS Identity and Access Management (IAM) 角色。
- 執行彈性災難復原的網路，根據所需的連線設定進行設定。

限制

- 除非資料庫託管在 Amazon 關聯式資料庫服務 (Amazon RDS) 上，否則您可以使用此模式複寫所有層，在這種情況下，我們建議您使用 Amazon RDS 的跨區域複製功能。
- 彈性災難復原與 CloudEndure 災難復原不相容，但您可以從 CloudEndure 災難復原升級。如需詳細資訊，請參閱彈性災難復原文件中的常見問題集。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 會限制您拍攝快照的速率。您可以使用彈性災難復原，在單一 AWS 帳戶中複寫最多 300 部伺服器。若要複寫更多伺服器，您可以使用多個 AWS 帳戶或多個目標 AWS 區域。您必須為每個帳戶和區域分別設定彈性災難復原。) 如需詳細資訊，請參閱彈性災難復原文件中的最佳做法。
- 來源工作負載 (JD Edwards EnterpriseOne 應用程式和資料庫) 必須託管在 EC2 執行個體上。此模式不支援內部部署或其他雲端環境中的工作負載。
- 這種模式著重於 JD 愛德華茲 EnterpriseOne 組件。完整的 DR 和業務連續性計劃 (BCP) 應包括其他核心服務，包括：
  - 網路 (虛擬私有雲、子網路和安全性群組)
  - Active Directory
  - Amazon WorkSpaces
  - Elastic Load Balancing
  - 受管資料庫服務，例如 Amazon Relational Database Service 服務 (Amazon RDS)

如需有關必要條件、組態和限制的其他資訊，請參閱彈性災難復原說明文件。

產品版本

- 甲骨文 JD 愛德華 EnterpriseOne (Oracle 與 SQL 伺服器支援的版本是以 Oracle 最低技術需求為基礎)

架構

目標技術堆疊

- 適用於生產和非生產環境的單一區域和單一虛擬私有雲 (VPC)，第二個區域用於 DR
- 單一可用區域，確保伺服器之間的低延遲
- 可分配網路流量的應用 Application Load Balancer，以提升應用程式跨多個可用區域的延展性和可用性
- Amazon Route 53 提供域名系統（DNS）配置
- Amazon WorkSpaces 為用戶提供雲端桌面體驗
- 亞馬遜簡易儲存服務 (Amazon S3)，用於存放備份、檔案和物件
- Amazon CloudWatch 適用於應用程式記錄、監控和警示
- 適用於災難復原的 Amazon 彈性災難復原

目標架構

下圖顯示 EnterpriseOne 使用彈性災難復原的 JD Edwards 跨區域災難復原架構。

程序

以下是該過程的高級審查。有關詳細信息，請參見史詩部分。

- 彈性災難復原複寫從初始同步開始。在初始同步期間，AWS 複寫代理程式會將來源磁碟中的所有資料複寫到暫存區子網路中的適當資源。
- 在初始同步完成之後，連續複寫會無限期地繼續進行。
- 在安裝代理程式並開始複寫之後，您可以檢閱啟動參數，其中包括服務特定組態和 Amazon EC2 啟動範本。當來源伺服器指示為可進行復原時，您可以啟動執行處理。
- 當彈性災難復原發出一系列 API 呼叫以開始啟動作業時，會根據您的啟動設定立即在 AWS 上啟動復原執行個體。該服務會在啟動期間自動啟動轉換服務器。
- 轉換完成後，新執行個體就會在 AWS 上執行，並可供使用。啟動時的來源伺服器狀態由與啟動執行個體相關聯的磁碟區表示。轉換程序涉及變更驅動程式、網路和作業系統授權，以確保執行個體在 AWS 上以原生方式開機。
- 啟動後，新建立的磁碟區將不再與來源伺服器保持同步。AWS 複寫代理程式會持續定期將對來源伺服器所做的變更複寫到暫存區域磁碟區，但啟動的執行個體不會反映這些變更。

- 當您啟動新的鑽研或復原執行個體時，資料一律會反映在從來源伺服器複寫到暫存區子網路的最新狀態中。
- 當來源伺服器標示為準備復原時，您可以啟動執行處理。

備註：此程序的運作方式有兩種：用於從主要 AWS 區域容錯移轉到 DR 區域，以及在復原時容錯回到主要站台。您可以透過以完全協調的方式將資料複製的方向從目標機器反轉回來源機器，以準備容錯回復。

此模式中描述的此過程的好處包括：

- 彈性：複寫伺服器會根據資料集和複寫時間向外擴充並擴充，因此您可以執行 DR 測試，而不會中斷來源工作負載或複寫。
- 可靠性：複製功能強大、不中斷且持續。
- 自動化：此解決方案為測試、復原和容錯回復提供統一的自動化程序。
- 成本最佳化：您只能複製所需的磁碟區並為其付費，並且只有在啟動這些資源後，才能為 DR 位置的運算資源付費。您可以針對多個來源或具有大量 EBS 磁碟區的單一來源使用成本最佳化的複製執行個體 (我們建議您使用運算最佳化執行個體類型)。

自動化和規模

當您大規模執行災難復原時，JD Edwards EnterpriseOne 伺服器會與環境中的其他伺服器具有相依性。例如：

- 在開機時連線到 JD Edwards EnterpriseOne 支援的資料庫的 JD Edwards EnterpriseOne 應用程式伺服器對該資料庫具有相依性。
- 需要驗證且需要在開機時連線至網域控制站以啟動服務的 JD Edwards EnterpriseOne 伺服器具有網域控制站的相依性。

因此，我們建議您將容錯移轉工作自動化。例如，您可以使用 AWS Lambda 或 AWS Step Functions 自動執行 JD Edwards EnterpriseOne 啟動指令碼和負載平衡器變更，以自動化 end-to-end 容錯移轉程序。如需詳細資訊，請參閱部落格文章使用 AWS 彈性災難復原建立可擴展的災難復原計劃。

工具

AWS 服務

- Amazon Elastic Block Store (Amazon EBS) 提供區塊層級儲存體磁碟區，可與 EC2 執行個體搭配使用。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Elastic 災難復原使用經濟實惠的儲存裝置、最少運算和復原功能，快速可靠地 point-in-time 復原現場部署和雲端應用程式，將停機時間和資料遺失
- Amazon Virtual Private Cloud (Amazon VPC) 可讓您完全控制虛擬聯網環境，包括資源配置、連線和安全性。

最佳實務

一般最佳做法

- 有一個書面計劃，說明在發生真正的恢復事件時該怎麼做。
- 正確設定彈性災難復原之後，請建立 AWS CloudFormation 範本，以便在需要時隨需建立組態。決定伺服器和應用程式的啟動順序，並將其記錄在復原計畫中。
- 執行定期鑽研 (適用標準 Amazon EC2 費率)。
- 使用彈性災難復原主控台或以程式設計方式監視進行中複寫的健全狀況。
- 保護 point-in-time 快照並在終止執行個體之前進行確認。
- 為 AWS 複寫代理程式安裝建立 IAM 角色。
- 在真實 DR 案例中為復原執行個體啟用終止保護。
- 針對您啟動復原執行個體的伺服器，請勿在彈性災難復原主控台中使用 AWS 中斷連線動作，即使發生真實復原事件也是如此。執行中斷連線會終止與這些來源伺服器相關的所有複製資源，包括您的 point-in-time (PIT) 復原點。
- 變更 PIT 原則以變更快照保留的天數。
- 在彈性災難復原啟動設定中編輯啟動範本，為目標伺服器設定正確的子網路、安全群組和執行個體類型。
- 使用 Lambda 或 Step Functions 自動化 JD Edwards EnterpriseOne 啟動指令碼和負載平衡器變更，將 end-to-end 容錯移轉程序自動化。

JD 愛德華茲 EnterpriseOne 優化和注意事項

- 移PrintQueue至資料庫。
- 移MediaObjects至資料庫。

- 從批處理和邏輯服務器中排除日誌和臨時文件夾。
- 從甲骨文中排除臨時文件夾 WebLogic。
- 建立用於容錯移轉之後啟動的指令碼。
- 排除 SQL 伺服器的臨時資料庫。
- 排除甲骨文的臨時文件。

史诗

執行初始任務和配置

任務	描述	所需技能
設定複製網路。	在主要 AWS 區域中實作您的 JD Edwards EnterpriseOne 系統，並識別 DR 的 AWS 區域，請按照彈性災難復原文件中複寫網路需求一節中的步驟來規劃和設定複寫和 DR 網路。	AWS 管理員
決定 RPO 和 RTO。	識別應用程式伺服器和資料庫的復原時間目標 (RTO) 和復原點目標 (RPO)。	雲端架構師、DR 架構師
啟用 Amazon EFS 的複寫功能。	如果適用，請使用 AWS、rsync 或其他適當的工具，針對共用檔案系統 (Amazon EFS) 等共用檔案系統啟用從 AWS DataSync 主要到 DR 區域的複寫功能。	雲端管理員
在 DR 的情況下管理 DNS。	識別在 DR 鑽研或實際 DR 期間更新網域名稱系統 (DNS) 的程序	雲端管理員
建立用於設定的 IAM 角色。	遵循彈性災難復原文件中彈性災難復原初始化和許可一節中	雲端管理員

任務	描述	所需技能
	的指示，建立 IAM 角色以初始化和管理 AWS 服務。	
設定 VPC 對等互連。	請確定來源和目標 VPC 是對等的，並且彼此可以存取。如需組態指示，請參閱 Amazon VPC 說明文件。	AWS 管理員

設定彈性災難復原複寫設定

任務	描述	所需技能
初始化彈性災難復原。	開啟彈性災難復原主控台，選擇目標 AWS 區域 (您將在其中複寫資料並啟動復原執行個體)，然後選擇 [設定預設複寫設定]。	AWS 管理員
設定複製伺服器。	1. 在 [設定複製伺服器] 窗格中，輸入暫存區域子網路和複製伺服器執行個體類型。預設選取 `t3.small` 執行個體類型。根據您的需求設定此設定，並記得考量執行個體定價。如需詳細資訊，請參閱 Amazon EC2 定價。 2. 在「服務存取」段落中，選擇檢視詳細資訊，以複查服務連結的角色和服務初始化期間建立的其他原則。 3. 選擇下一步。	AWS 管理員
設定磁碟區和安全群組。	1. 在 [磁碟區和安全群組] 窗格中，選取複寫伺服器的 EBS 磁碟區類型，然後將	AWS 管理員

任務	描述	所需技能
	Amazon EBS 加密設定為預設。 2. 選取 [永遠使用 AWS 彈性災難復原安全群組]，讓彈性災難復原自動附加並監控預設安全群組。 3. 選擇下一步。	
設定其他設定。	1. 在 [其他設定] 窗格中，設定資料路由和節流、PIT 原則和標籤。 • 資料路由和節流控制資料從外部伺服器流向複寫伺服器的方式。選擇 [使用私有 IP 進行資料複製]。否則，複寫伺服器會自動指派公用 IP，而資料將透過公用網際網路流動。 • 在「時間點 (PIT)」原則區段中，設定保留原則，以決定不需要快照的持續時間。預設保留期間為七天。 • 在「標籤」區段中，將自訂標籤新增至 AWS 帳戶中由彈性災難復原建立的資源。 2. 選擇 [下一步]，檢閱下一個窗格中的設定，然後選擇 [建立預設值] 以建立預設範本。	AWS 管理員

安裝 AWS 複寫代理程式

任務	描述	所需技能
建立 IAM 角色。	建立包含`AWSElasticDisasterRecoveryAgentInstallationPolicy` 政策的 IAM 角色。在 [選取 AWS 存取類型] 區段中，啟用程式設計存取。請注意存取金鑰 ID 和秘密存取金鑰。在安裝 AWS 複寫代理程式期間，您將需要此資訊。	AWS 管理員
檢查要求。	在彈性災難復原文件中查看並完成安裝 AWS 複寫代理程式的先決條件。	AWS 管理員
安裝 AWS 複寫代理程式。	遵循作業系統的安裝指示，並安裝 AWS 複寫代理程式。 • 對於 Microsoft 視窗：下載安裝文件並以管理員身份運行 .exe 文件。 回應提示以完成安裝。 • 對於 Linux：複製以下命令（按照顯示的順序）並將其粘貼到您的安全殼層（SSH）會話中。第一個命令會下載安裝程式，第二個命令會執行它。 `wget -O ./aws-replication-installer-init.py https://aws-elastic-disaster-recovery-us-west-2.s3.amazonaws.co`	AWS 管理員

任務	描述	所需技能
	m/latest/linux/aws -replication-insta ller-init.py 注意：變更 URL 以反映您的地區。 sudo python3 aws-repli cation-installer-i nit.py 回應提示以完成安裝。 對其餘伺服器重複這些步驟。	
監視複寫。	返回彈性災難復原來源伺服器窗格以監視複寫狀態。初始同步將花費一些時間，具體取決於數據傳輸的大小。 當來源伺服器完全同步時，伺服器狀態會更新為「就緒」。這表示已在暫存區中建立複製伺服器，且 EBS 磁碟區已從來源伺服器複製到暫存區。	AWS 管理員

設定啟動設定

任務	描述	所需技能
編輯啟動設定。	若要更新鑽研與復原執行個體的啟動設定，請在彈性災難修復主控台上選取來源伺服器，然後選擇 [動作] > [編輯啟動設定]。或者，您可以從 [來源	AWS 管理員

任務	描述	所需技能
	伺服器] 頁面選擇複製的來源機器，然後選擇 [啟動設定] 索引標籤。此索引標籤有兩個部分：一般啟動設定和 EC2 啟動範本。	
設定一般啟動設定。	根據您的需求修改一般啟動設定。 • 執行個體類型正確調整大小：如果選擇「基本」，彈性災難復原會略過您在 Amazon EC2 啟動範本中選取的執行個體類型，並根據來源伺服器的作業系統、CPU 和 RAM 自動選擇執行個體類型。 • 複製私有 IP：選擇是否要彈性災難復原，以確保鑽研或復原執行個體使用的私有 IP 與來源伺服器使用的私有 IP 相符。如果選擇是，請確保您在 Amazon EC2 啟動範本中設定的子網路 IP 範圍包含私有 IP 位址。 如需詳細資訊，請參閱彈性災難復原文件中的一般啟動設定。	AWS 管理員

任務	描述	所需技能
設定 Amazon EC2 啟動範本。	彈性災難復原使用 Amazon EC2 啟動範本為每個來源伺服器啟動鑽研和復原執行個體。安裝 AWS 複寫代理程式後，會為您新增至彈性災難復原的每個來源伺服器自動建立啟動範本。 如果要將 Amazon EC2 啟動範本與彈性災難復原搭配使用，則必須將該範本設定為預設啟動範本。 如需詳細資訊，請參閱彈性災難復原文件中的 EC2 啟動範本。	AWS 管理員

起始 DR 鑽研和容錯移轉

任務	描述	所需技能
啟動鑽研	1. 在 [彈性災難復原] 主控台上，開啟 [來源伺服器] 頁面，並確認來源伺服器的狀態為 [就緒]。 2. 選取您要執行 DR 鑽研的所有來源伺服器。 3. 從「起始復原工作」功能表中，選擇「起始追溯」，然後選取適當的 point-in-time 快照。這會啟動所選來源伺服器的復原工作。您可以在 [復原工作歷程記錄] 索引標籤上監視工作的狀態。	AWS 管理員

任務	描述	所需技能
	注意：來源伺服器的進一步變更將會同步至複製伺服器，而非鑽研執行處理。 啟動的鑽研執行處理也會顯示在「復原執行處理」頁面上 4. 測試並驗證 DR 鑽研執行個體。 5. 在復原執行個體頁面上，選取鑽研執行個體，然後選擇動作 > 中斷與 AWS 的連線。這會從復原執行個體中刪除 AWS 複寫代理程式，並從彈性災難復原中移除與復原執行個體相關聯的所有資源。 6. 選擇刪除復原執行個體。這會從彈性災難復原主控台刪除執行個體的表示法，並完全取消執行個體與彈性災難復原服務的關聯性。它不會刪除基礎 EC2 實例。 7. 從 Amazon EC2 主控台終止 DR 鑽研執行個體。 如需詳細資訊，請參閱彈性災難復原說明文件中的準備容錯移轉。	

任務	描述	所需技能
驗證鑽床。	在上一個步驟中，您已在 DR 區域中啟動新的目標執行個體。目標執行處理是根據您啟動啟動時所建立的快照來源伺服器的複本。 在此程序中，您可以連接到 Amazon EC2 目標機器，以確認它們正在如預期般執行。 1. 開啟 Amazon EC2 主控台。 2. 選擇執行個體 (執行中)。 3. 選取目標執行個體並記下其私有 IPv4 位址。 4. 請確定您可以連線至 EC2 執行個體，而且 JD Edwards EnterpriseOne 和相關元件會如預期般複寫。	

任務	描述	所需技能
起始容錯移轉。	容錯移轉是將流量從主要系統重新導向至次要系統。彈性災難復原可協助您透過在 AWS 上啟動復原執行個體來執行容錯移轉。啟動復原執行個體後，您可以將主要系統的流量重新導向至這些執行個體。 1. 在彈性災難復原主控台上，開啟 [來源伺服器] 頁面，並確認來源伺服器的 [準備復原] 欄顯示 [就緒]，且 [資料複寫狀態] 欄顯示 [狀況良好]。 2. 選取來源伺服器。從 [開始復原工作] 功能表中，選擇 [開始復原]。 3. 選取要從中啟動復原執行處理的 point-in-time 快照，然後選擇 [開始復原]。 這會啟動復原工作。您可以在「復原執行處理」頁面監督工作的狀態。 4. 測試並驗證復原執行個體。如果需要，請調整 DNS 組態，並將您的 JD Edwards EnterpriseOne 應用程式連接到資料庫。 5. 您現在可以中斷連線和解除委任來源 JD Edwards EnterpriseOne 伺服器，因為所有變更都已寫入新的復原執行個體。	AWS 管理員

任務	描述	所需技能
	6. 遵循安裝 AWS 複寫代理程式史詩中所述的程序，將復原執行個體註冊為 DR 區域中的來源伺服器。 如需詳細資訊，請參閱彈性災難復原說明文件中的執行容錯移轉。	

任務	描述	所需技能
啟動容錯回復。	啟動容錯回復的程序與啟動容錯移轉的程序類似。 1. 在主要區域中開啟彈性災難復原主控台。導覽至復原執行個體頁面，選取鑽研執行個體，然後選擇 [動作]、[中斷與 AWS 的連線]、[刪除復原執行個體]。 2. 在 DR 區域中開啟彈性災難復原主控台。透過安裝 AWS 複寫代理程式，將新的 JD Edwards EnterpriseOne 伺服器註冊為 DR 區域中的來源伺服器。資料將與在新暫存子網路中佈建的新複製伺服器同步。 附註：當新的 JD Edwards 伺 EnterpriseOne 服器註冊為來源伺服器時，您可能會在彈性災難復原主控台中看到兩個來源伺服器：一個從主 EC2 執行個體建立的伺服器，以及從復原執行個體建立的新伺服器。我們建議您正確標記伺服器以避免混淆，最好將新伺服器新增至啟動範本。 3. 若要從主要區域重新啟動 DR 複寫，請取消已啟動復原執行個體與 DR 區域中的彈性災難復原主控台的關聯，並將主機註冊為主要區域中的來源伺服器。	AWS 管理員

任務	描述	所需技能
	如需詳細資訊，請參閱彈性災難復原說明文件中的執行容錯回復。	

任務	描述	所需技能
啟動 JD 愛德華茲 EnterpriseOne 組件。	1. 通過登錄到 EnterpriseOne 數據庫服務器啟動 JD 愛德華數據庫。 2. 當數據庫運行時，啟動 JD Edwards EnterpriseOne 邏輯和批處理服務器。 3. 在網頁伺服器 WebLogic 上啟動，然後在 JAS 伺服器上啟動 JAS 執行個體。 4. WebLogic 在佈建伺服器和 SM 主控台的伺服器上啟動。 5. 在伺服器上啟動 SM 代理程式。 6. 確認登入 JD 愛德華茲 EnterpriseOne 正常運作。 您需要將 Route 53 和 Application Load Balancer 中的變更納入 JD Edwards EnterpriseOne 連結才能運作。 您可以使用 Lambda、步驟函數和 Systems Manager (執行命令) 來自動執行這些步驟。 備註：彈性災難復原會對託管作業系統和檔案系統的來源 EC2 執行個體 EBS 磁碟區執行區塊層級複寫。使用 Amazon EFS 建立的共用檔案系統不屬於此複寫的一部分。如第一篇史詩所述 DataSync，您可以使用 AWS 將共用檔案	JD 愛德華 EnterpriseOne 數控

任務	描述	所需技能
	系統複寫到 DR 區域，然後在 DR 系統中掛載這些複寫的檔案系統。	

故障診斷

問題	解決方案
來源伺服器資料複寫狀態停止且複寫延遲。如果您檢查詳細資料，資料複製狀態會顯示未看到代理程式。	檢查以確認停止的來源伺服器正在執行中。 備註：如果來源伺服器當機，複製伺服器會自動終止。 如需延遲問題的詳細資訊，請參閱彈性災難復原說明文件中的複寫延遲問題。
在 RHEL 8.2 中掃描磁碟後，在來源 EC2 執行個體中安裝 AWS 複寫代理程式會失敗。 `aws_replication_agent_installer.log` 顯示內核頭文件丟失。	在 RHEL 8、CentOS 8 或 Oracle 8 上安裝 AWS 複寫代理程式之前，請先執行： `sudo yum install elfutils-libelf-devel` 如需詳細資訊，請參閱彈性災難復原說明文件中的 Linux 安裝需求。
在彈性災難復原主控台上，您會看到來源伺服器為 [就緒]，且延遲且資料複寫狀態為 [停止]。 根據 AWS 複寫代理程式無法使用的時間長度，狀態可能表示出現高延遲，但問題仍然不變。	使用作業系統命令確認 AWS 複寫代理程式正在來源 EC2 執行個體中執行，或確認執行個體正在執行。 修正任何問題後，彈性災難復原將重新啟動掃描。請等到所有資料均已同步，且複寫狀態為 [正常]，然後再啟動 DR 鑽研。
具有高延遲的初始複寫。在彈性災難復原主控台上，您可以看到來源伺服器的初始同步狀態非常慢。	檢查彈性災難復原說明文件的複寫延遲問題一節中所述的複寫延遲問題。

問題	解決方案
	複寫伺服器可能因為內建運算作業而無法處理負載。在這種情況下，請在諮詢 AWS 技術 Support 團隊後嘗試升級執行個體類型。

相關資源

- AWS 彈性災難復原使用者指南
- 使用 AWS 彈性災難復原建立可擴展的災難復原計劃 (AWS 部落格文章)
- AWS 彈性災難復原-技術簡介 (AWS 技能建置課程；需要登入)
- AWS 彈性災難復原快速入門指南

使用 AWS 在不同 AWS 區域的 Amazon EFS 檔案系統之間同步資料 DataSync

由莎拉特·錢德拉·波圖拉（AWS）和阿迪亞·安巴蒂（AWS）創建

代碼存儲庫：aws-efs-crossregion-datasync	環境：PoC 或試點	技術：基礎架構、儲存與備份
AWS 服務：AWS CDK；AWS DataSync；Amazon EFS		

Summary

此解決方案為不同 AWS 區域中的 Amazon 彈性檔案系統 (Amazon EFS) 執行個體之間提供有效且安全的資料同步功能強大的架構。此方法具有可擴充性，並提供受控的跨區域資料複寫。此解決方案可以增強您的災難復原和資料備援策略。

透過使用 AWS Cloud Development Kit (AWS CDK)，此模式會使用做為基礎設施即程式碼 (IaC) 方法來部署解決方案資源。AWS CDK 應用程式部署了必要的 AWS DataSync、Amazon EFS、Amazon Virtual Private Cloud (Amazon VPC) 和亞馬遜彈性運算雲端 (Amazon EC2) 資源。此 IaC 提供可重複且受版本控制的部署程序，與 AWS 最佳實務完全一致。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 2.9.11 版或更新版本，已安裝和設定
- AWS CDK 版本 2.114.1 或更新版本，已安裝並啟動載入
- 已安裝 NodeJS 版本 20.8.0 或更新版本

限制

- 該解決方案繼承了 Amazon EFS DataSync 的限制，例如資料傳輸速率、大小限制和區域可用性。如需詳細資訊，請參閱 AWS DataSync 配額和 Amazon EFS 配額。

- 此解決方案僅支援 Amazon EFS。 DataSync 支持其他 AWS 服務，例如 Amazon 簡單存儲服務（亞馬遜 S3）和 Amazon FSx for Lustre。不過，此解決方案需要修改，才能將資料與這些其他服務同步。

架構

此解決方案會部署下列 AWS CDK 堆疊：

- Amazon VPC 堆疊 — 此堆疊可在主要和次要 AWS 區域中設定虛擬私有雲端 (VPC) 資源，包括子網路、網際網路閘道和 NAT 閘道。
- Amazon EFS 堆疊 — 此堆疊可將 Amazon EFS 檔案系統部署到主要和次要區域，並將它們連接到各自的 VPC。
- Amazon EC2 堆疊 — 此堆疊會在主要和次要區域啟動 EC2 執行個體。這些執行個體設定為掛接 Amazon EFS 檔案系統，以便存取共用儲存。
- DataSync 位置堆棧-該堆棧使用稱為在主要和次`DataSyncLocationConstruct`要區域中創建 DataSync 位置資源的自定義構造。這些資源會定義資料同步處理的端點。
- DataSync 任務堆棧-該堆棧使用稱為`DataSyncTaskConstruct`在主區域中創建 DataSync 任務的自定義構造。此工作設定為使用 DataSync 來源和目的地位置，在主要和次要區域之間同步處理資料。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS DataSync 是一種線上資料傳輸和探索服務，可協助您在 AWS 儲存服務之間移動檔案或物件資料。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

代碼存儲庫

此模式的程式碼可在 GitHub Amazon EFS 跨區域 DataSync 專案儲存庫中取得。

最佳實務

請遵循在中使用 AWS CDK 的最佳實務中所述的最佳實務 TypeScript 來建立 IaC 專案。

史诗

部署 AWS CDK 應用程式

任務	描述	所需技能
克隆項目存儲庫。	輸入以下命令以複製 Amazon EFS 跨區域 DataSync 專案儲存庫。 `git clone https://github.com/aws-samples/aws-efs-crossregion-datasync.git`	AWS DevOps
安裝 npm 依賴關係。	輸入以下命令。 `npm ci`	AWS DevOps
選擇主要和次要區域。	在複製的存放庫中，導覽至目`src/infa`錄。在`Launcher.ts` 檔案中，更新`PRIMARY_AWS_REGION` 和`SECONDARY_AWS_REGION` 值。使用對應的區域代碼。 `const primaryRegion = { account: account, region: '<PRIMARY_AWS_REGION>' };`	AWS DevOps

任務	描述	所需技能
	`const secondaryRegion = { account: account, region: '<SECONDARY_AWS_REGION>' };`	
引導環境。	輸入以下命令以啟動您要使用的 AWS 帳戶和 AWS 區域。 `cdk bootstrap <aws_account>/<aws_region>` 如需詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。	AWS DevOps
列出 AWS CDK 堆疊。	輸入以下命令以檢視應用程式中 AWS CDK 堆疊的清單。 `cdk ls`	AWS DevOps
合成 AWS CDK 堆疊。	輸入以下命令，為 AWS CDK 應用程式中定義的每個堆疊產生 AWS CloudFormation 範本。 `cdk synth`	AWS DevOps
部署 AWS CDK 應用程式。	輸入以下命令將所有堆疊部署到您的 AWS 帳戶，而無需手動核准任何變更。 `cdk deploy --all --require-approval never`	AWS DevOps

驗證部署

任務	描述	所需技能
登入主要區域中的 EC2 執行個體。	1. 使用 AWS Systems Manager 的工作階段管理員，登入主要區域中的 EC2 執行個體。如需指示，請參閱使用 AWS Systems Manager 工作階段管理員 Connect 到 Linux 執行個體。 2. 將目錄變更為 Amazon EFS 掛接路徑。 `cd /mnt/efs`	AWS DevOps
創建一個臨時文件。	輸入下列命令，在 Amazon EFS 掛載路徑中建立暫存檔案。 `sudo dd if=/dev/zero \` `of=tmptst.dat \` `bs=1G \` `seek=5 \` `count=0` `ls -lrt tmptst.dat`	AWS DevOps
開始工 DataSync 作。	輸入以下命令，將暫存檔案從主要區域複製到次要區域，其中<ARN-task> 是 DataSync 任務的 Amazon 資源名稱 (ARN)。 `aws datasync start-tas` `k-execution \`	AWS DevOps

任務	描述	所需技能
	`--task-arn <ARN-task>` 命令會以下列格式傳回工作執行的 ARN。 `arn:aws:datasync:<region>:<account-ID>:task/task-execution/<exec-ID>`	
檢查資料傳輸的狀態。	輸入下列命令來描述 DataSync 執行工作，其中`<ARN-task-execution>` 是工作執行的 ARN。 `aws datasync describe-task-execution \ --task-execution-arn <ARN-task-execution>` 當、且`VerifyStatus` 全部都具有值時 `PrepareStatus` `TransferStatus` ，DataSync 任務即完成`SUCCESS`。	AWS DevOps

任務	描述	所需技能
登入次要區域中的 EC2 執行個體。	1. 使用 AWS Systems Manager 的工作階段管理員，登入次要區域中的 EC2 執行個體。如需指示，請參閱使用 AWS Systems Manager 工作階段管理員 Connect 到 Linux 執行個體。 2. 將目錄變更為 Amazon EFS 掛接路徑。 `cd /mnt/efs`	AWS DevOps
驗證複寫。	輸入以下命令以確認該暫存檔案存在於 Amazon EFS 檔案系統中。 `ls -lrt` `tmptst.dat`	AWS DevOps

相關資源

AWS 文件

- AWS CDK 應用程式介面參考
- 使用 Amazon EFS 設定 AWS DataSync 傳輸
- AWS DataSync 傳輸的疑難排解問題

其他 AWS 資源

- AWS DataSync 常見問題

將心臟起搏器叢集從 ENSA1 升級至 ENSA2

由格格利塞爾迪（AWS）和巴拉茲·桑多爾斯庫布利克（AWS）創建

環境：生產	來源：基於 ENSA1 的起搏器群集	目標：基於 ENSA2 的起搏器群集
R 型：重新建築	工作負載：SAP	技術：基礎設施；現代化
AWS 服務：Amazon EC2		

Summary

此模式說明將以獨立排入佇列伺服器 (ENSA1) 為基礎的 SAP 起搏器叢集升級為 ENSA2 的步驟和考量事項。此模式中的資訊適用於 SUSE Linux 企業伺服器 (SLES) 和 RHEL (RHEL) 作業系統。

起搏器叢集 NetWeaver 在 ENSA1 架構上執行，並專門針對 ENSA1 進行設定。如果您在 Amazon Web Services (AWS) 上執行 SAP 工作負載，但您有興趣轉移到 ENSA2，您可能會發現 SAP、SUSE 和 RHEL 文件無法提供全面的資訊。此模式描述了重新配置 SAP 參數和起搏器叢集以從 ENSA1 升級到 ENSA2 所需的技術步驟。它提供了 SUSE 系統的範例，但 RHEL 叢集的概念是相同的。

附註：ENSA1 和 ENSA2 是僅與 SAP 應用程式有關的概念，因此此模式中的資訊不適用於 SAP HANA 或其他類型的叢集。

從技術上講，ENSA2 可搭配或不搭配排入佇列複製器使用 2。不過，高可用性 (HA) 和容錯移轉自動化 (透過叢集解決方案) 需要排入佇列複製器 2。此模式使用術語 ENSA2 叢集來表示具有獨立排入佇列伺服器 2 和排入佇列複製器 2 的叢集。

先決條件和限制

前提

- 在 SLES 或 RHEL 上使用起搏器和群組同步的工作型 ENSA 型叢集。
- 至少兩個亞馬遜彈性運算雲端 (Amazon EC2) 執行個體正在執行中 (ABAP) SAP 中央服務 (ASCS/SCS) 和排入佇列複寫伺服器 (ERS) 執行個體。

- 管理 SAP 應用程式和叢集的知識。
- 以根使用者身分存取 Linux 環境。

限制

- 以 ENSA1 為基礎的叢集僅支援雙節點架構。
- 以 ENSA2 為基礎的叢集無法部署至 7.52 之前的 SAP NetWeaver 版本。
- 叢集中的 EC2 執行個體應位於不同的 AWS 可用區域。

產品版本

- SAP NetWeaver 版本 7.52 或更新版本
- 從 2020 年開始，僅支援 ENSA2 叢集
- 支援 ENSA2 和排入佇列複製器 2 的核心 7.53 或更新版本
- 適用於 SAP 應用程式的 SLE 版本 12 或更新版本
- 適用於具有高可用性 (HA) 之 SAP 的 RHEL 7.9 版或更新版本

架構

源, 技術, 堆棧

- SAP NetWeaver 7.52 與 SAP 核心 7.53 或更新版本
- SLES 或 RHEL 作業系統

目標技術堆疊

- SAP NetWeaver 7.52 與 SAP 核心 7.53 或更新版本，包括具有 ABAP 平台的 2020 年 S/4HANA
- SLES 或 RHEL 作業系統

目標架構

下圖顯示以 ENSA2 叢集為基礎的 ASCS/SCS 和 ERS 執行個體的 HA 組態。

ENSA1 和 ENSA2 叢集的比較

SAP 引入了 ENSA2 作為 ENSA1 的繼任者。以 ENSA1 為基礎的叢集支援雙節點架構，其中 ASCS/SCS 執行個體會在發生錯誤時容錯移轉至 ERS。此限制源於 ASCS/SCS 執行個體在容錯移轉後，如何從 ERS 節點的共用記憶體重新取得鎖定資料表資訊。具有排入佇列複製器 2 的 ENSA2 型叢集可消除此限制，因為 ASCS/SCS 執行個體可透過網路從 ERS 執行個體收集鎖定資訊。以 ENSA2 為基礎的叢集可以有兩個以上的節點，因為 ASCS/SCS 執行個體不再需要容錯移轉至 ERS 節點。不過，在雙節點的 ENSA2 叢集環境中，ASCS/SCS 執行個體仍會容錯移轉至 ERS 節點，因為叢集中沒有其他節點可容錯移轉至。) ENSA2 從 SAP 核心 7.50 開始受到支援，但有一些限制。對於支援排入佇列複製器 2 的 HA 設定，最低需求為 NetWeaver 7.52 (請參閱 SAP OSS 注意事項 2630416)。預設情況下建議使用 ENSA2 架構，而 S/4HANA 僅支援從 2020 年版本開始的 ENSA2。

自動化和規模

目標架構中的 HA 叢集會讓 ASCS 自動容錯移轉至其他節點。

移至以 ENSA2 為基礎之叢集的案例

升級為以 ENSA2 為基礎的叢集有兩種主要案例：

- 案例 1：假設您的 SAP 發行版本和核心版本支援 ENSA2，則您選擇升級至 ENSA2，而不需要隨附的 SAP 升級或 S/4HANA 轉換。
- 案例 2：您使用 SUM 移至 ENSA2 作為升級或轉換的一部分 (例如，移至 S/4HANA 1809 或更新版本)。

史詩部分涵蓋了這兩種情況的步驟。第一個案例會要求您先手動設定 SAP 相關參數，然後才能變更 ENSA2 的叢集配置。在第二個案例中，SUM 會部署二進位檔案和 SAP 相關的參數，而您唯一剩下的工作就是更新 HA 的叢集組態。我們仍建議您在使用 SUM 之後驗證 SAP 參數。在大多數情況下，S/4HANA 轉換是叢集升級的主要原因。

工具

- 對於作業系統套件管理員，我們建議使用 Zypper (適用於 SLES) 或 YUM (適用於 RHEL) 工具。
- 對於叢集管理，我們建議使用 crm (適用於 SLES) 或個人電腦 (適用於 RHEL) 殼層。
- SAP 執行個體管理工具，例如 SAP 控制。
- （可選）用於 S/4HANA 轉換升級的 SUM 工具。

最佳實務

- 如需在 AWS 上使用 SAP 工作負載的最佳實務，請參閱 AWS Well-Architected Framework SAP 鏡頭。
- 考慮 ENSA2 多節點架構中的叢集節點數目 (奇數或偶數)。
- 為 SLE15 設定 ENSA2 叢集，以符合 SAP S/4-HA-CLU 1.0 認證標準。
- 在升級到 ENSA2 之前，請務必先儲存或備份現有的叢集和應用程式狀態。

史诗

針對 ENSA2 手動設定 SAP 參數 (僅適用於案例 1)

任務	描述	所需技能
在預設設定檔中設定參數。	如果您想要在保留相同 SAP 版本的同時升級至 ENSA2，或者您的目標版本預設為 ENSA1，請將預設設定檔 (DEFAULT.PFL 檔案) 中的參數設定為下列值。 `enq/enable=TRUE` `enq/serverhost=sapascsvirt` `enq/serverinst=10` `(instance number of ASCS/SCS instance)` `enque/process_location=REMOTESA` `enq/replicatorhost=sapersvirt` `enq/replicatorinst=11` `(instance number of ERS instance)` 其中`sapascsvirt` 是 ASCS 執行個體的虛擬主機名	SAP

任務	描述	所需技能
	稱，`sapersvirt` 也是 ERS 執行個體的虛擬主機名稱。您可以變更這些項目以符合您的目標環境。 注意：若要使用此升級選項，您的 SAP 版本和核心版本必須支援 ENSA2 和排入佇列複製器 2。	

任務	描述	所需技能
設定 ASCS/SCS 執行個體設定檔。	如果您想要在保留相同 SAP 版本的同時升級至 ENSA2，或者您的目標版本預設為 ENSA1，請在 ASCS/SCS 執行個體設定檔中設定下列參數。 定義 ENSA1 的輪廓截面看起來如下所示。 `#----------------- ----------------- ----------------- --------- Start SAP enqueue server #------------------ ----------------- ----------------- ------- _EN = en.sap$(S APSYSTEMNAME)$(INS TANCE_NAME) Execute_04 = local rm - f $(_EN) Execute_05 = local ln - s -f $(DIR_EXECUTABLE)/ enserver$(FT_EXE) $(_EN) Start_Program_01 = local $(_EN) pf=$(_PF)` 若要針對 ENSA2 重新設定此區段，請執行下 1. _ENQ根據來自 SAP 的最新資訊，將_EN程式前置詞變更為 (OSS 注意事項 2501860；需要 SAP ONE	SAP

任務	描述	所需技能
	Support 啟動平台使用者帳戶)。 2. enserver將enq_server 排入佇列伺服器的二進位檔從變更為。 3. 將新參數設定enq/server/replication/enable 為TRUE。 4. 確保這一點Autostart = 0。 變更後，此設定檔區段看起來會如下所示。 `#------------------ ------------------ ------------------ --------- Start SAP enqueue server #------------------ ------------------ ------------------ ------- _ENQ = enq.sap$(SAPSYSTEMNAME)$(IN STANCE_NAME) Execute_04 = local rm - f $(_ENQ) Execute_05 = local ln - s -f $(DIR_EXECUTABLE)/ enq_server$(FT_EXE) $(_ENQ) Start_Program_01 = local $(_ENQ) pf= $(_PF) ... enq/server/replic ation/enable = TRUE`	

任務	描述	所需技能
	`Autostart = 0` 重要：不_ENQ得啟用重新啟動選項。如果RestartProgram_01 設定為_ENQ，請將其變更為StartProgram_01 。這可防止 SAP 重新啟動服務或干擾叢集管理的資源。	

任務	描述	所需技能
設定 ERS 設定檔。	如果您想要在保留相同 SAP 版本的同時升級至 ENSA2，或者您的目標版本預設為 ENSA1，請在 ERS 執行個體設定檔中設定下列參數。 尋找定義排入佇列複製器的區段。它將類似於以下內容。 `#----------` `----------` `----------` `Start enqueue replicati` `on server` `#----------` `----------` `----------` `_ER = er.sap$(S` `APSYSTEMNAME)$(INS` `TANCE_NAME)` `Execute_03 = local rm -` `f $(_ER)` `Execute_04 = local ln -` `s -f $(DIR_EXECUTABLE)/` `enrepserver$(FT_EXE)` ` $(_ER)` `Start_Program_00 =` ` local $(_ER) pf=$(_PF)` ` NR=$(SCSID)` 重新設定此段落的排入佇列複製器 2： 1. 將_ER程式首碼變更為_ENQR根據 SAP 的最新記事 (OSS 注意事項 2501860；需要 SAP ONE	SAP

任務	描述	所需技能
	Support 啟動平台使用者帳戶)。 2. 將排入佇列複製器的二進位檔變更為enq_replicator 而非。enrepserver 3. 確保這一點Autostart = 0。 變更後，此設定檔區段看起來應如下所示。 `#------------------ ------------------ ------------------- Start enqueue replication server #------------------- ------------------ ------------------ _ENQR = enqr.sap$(SAPSYSTEMNAME)$(INSTANCE_NAME) Execute_01 = local rm -f $(_ENQR) Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enq_replicator$(FT_EXE) $(_ENQR) Start_Program_00 = local $(_ENQR) pf=$(_PF) NR=$(SCSID) … Autostart = 0` 重要：不_ENQR得啟用重新啟動選項。如果RestartPr	

任務	描述	所需技能
	ogram_01 設定為_ENQR，請將其變更為StartProgram_01 。這可防止 SAP 重新啟動服務或干擾叢集管理的服務。	
重新啟動 SAP 啟動服務。	變更此史詩中先前描述的設定檔之後，請重新啟動 ASCS/SCS 和 ERS 的 SAP 啟動服務。 `sapcontrol -nr 10 -function RestartService SCT` `sapcontrol -nr 11 -function RestartService SCT` 其中SCT指的是 SAP 系統識別碼，並假設 10 和 11 分別為 ASCS/SCS 與 ERS 執行處理的執行個體編號。	SAP

重新設定 ENSA2 的叢集 (兩種情況都需要)

任務	描述	所需技能
驗證 SAP 資源代理程式中的版本號碼。	當您使用 SUM 將 SAP 升級至 S/4HANA 1809 或更新版本時，SUM 會處理 SAP 設定檔中的參數變更。只有叢集需要手動調整。不過，建議您先驗證參數設定，然後再對叢集進行任何變更。	AWS 系統管理員

任務	描述	所需技能
	注意：本史詩中的範例假設您使用的是 SUSE 作業系統。如果您使用的是 RHEL，您將需要使用諸如 YUM 和電腦外殼之類的工具，而不是 Zypper 和 crm。 檢查架構中的兩個節點，以確認`resource-agents` 套件符合 SAP 建議的最低版本。針對 SLE，請查看 SAP 作業系統附註 2641019。若為 RHEL，請查看 SAP 作業系統註釋 2641322。SAP 備註需要 SAP ONE Support 啟動台使用者帳戶。） `sapers:sctadm 23> zypper search -s -i resource-agents` `Loading repository data...` `Reading installed packages...` `S | Name | Type | Version | Arch | Repository` `--+-----------------+---------+-----------------+---------------+------+-----------------` `i | resource-agents | package | 4.8.0+git30.d0077df0-150300.8.28.1 | x86_64 |`	

任務	描述	所需技能
	`SLE-Product-HA15-SP3-Updates` 如有必要，請更新resource-agents 版本。	
備份叢集配置。	備份 CRM 叢集配置，如下所示。 `crm configure show > /tmp/cluster_config_backup.txt`	AWS 系統管理員
設定維護模式。	將叢集設定為維護模式。 `crm configure property maintenance-mode="true"`	AWS 系統管理員

任務	描述	所需技能
檢查叢集配置。	檢查目前的叢集配置。 `crm configure show` 以下是完整輸出的摘錄： `node 1: sapascs` `node 2: sapers` `...` `primitive rsc_sap_S` `CT_ASCS10 SAPInstance` ` \` `operations $id=rsc_s` `ap_SCT_ASCS10-oper` `ations \` `op monitor interval=120` ` timeout=60 on-fail=r` `estart \` `params InstanceN` `ame=SCT_ASCS10_sap` `ascsvirt START_PRO` `FILE="/sapmnt/SCT/` `profile/SCT_ASCS10` `_sapascsvirt" \` `  AUTOMATIC_RECOVER=` `false \` `meta resource-stickines` `s=5000 failure-t` `imeout=60 migration-` `threshold=1 priority=` `10` `primitive rsc_sap_S` `CT_ERS11 SAPInstance \` `operations $id=rsc_s` `ap_SCT_ERS11-opera` `tions \` `op monitor interval=120` ` timeout=60 on-fail=r` `estart \` `params InstanceN` `ame=SCT_ERS11_sape`	AWS 系統管理員

任務	描述	所需技能
	`` rsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ERS11_ sapersvirt" \ AUTOMATIC_RECOVER= false IS_ERS=true \ meta priority=1000 ... colocation col_sap_S CT_no_both -5000: grp_SCT_ERS11 grp_SCT_ASCS10 location loc_sap_S CT_failover_to_ers rsc_sap_SCT_ASCS10 \ rule 2000: runs_ers_SCT eq 1 order ord_sap_S CT_first_start_asc s Optional: rsc_sap_S CT_ASCS10:start rsc_sap_SCT_ERS11: stop symmetrical=false ... ``` 其中sapascsvirt 指的是 ASCS 執行個體的虛擬主機名稱，sapersvirt 是指 ERS 執行個體的虛擬主機名稱，並SCT參照 SAP 系統識別碼。	

任務	描述	所需技能
移除容錯移轉主機託管限制。	在前面的範例中，位置限制會`loc_sap_SCT_failover_to_ers` 指定 ASCS 的 ENSA1 功能在容錯移轉時一律遵循 ERS 執行個體。使用 ENSA2 時，ASCS 應該能夠自由容錯移轉到任何參與的節點，因此您可以移除此限制。 `crm configure delete loc_sap_SCT_failover_to_ers`	AWS 系統管理員

任務	描述	所需技能
調整基元。	您還需要對 ASCS 和 ERS SAP 實例原語進行較小的更改。 以下是針對 ENSA1 設定的 ASCS SAP 執行個體原始元件的範例。 `primitive rsc_sap_SCT_ASCS10 SAPInstance \` `operations $id=rsc_sap_SCT_ASCS10-operations \` `op monitor interval=120 timeout=60 on-fail=restart \` `params InstanceName=SCT_ASCS10_sapascsvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ASCS10_sapascsvirt" \` `AUTOMATIC_RECOVER=false \` `meta resource-stickiness=5000 failure-timeout=60 migration-threshold=1 priority=10` 若要升級至 ENSA2，請將此組態變更為下列設定。 `primitive rsc_sap_SCT_ASCS10 SAPInstance \`	AWS 系統管理員

任務	描述	所需技能
	operations $id=rsc_s ap_SCT_ASCS10-oper ations \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceN ame=SCT_ASCS10_sap ascsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ASCS10 _sapascsvirt" \ AUTOMATIC_RECOVER= false \ meta resource-stickines s=3000 這是一個針對 ENSA1 配置的 ERS SAP 實例原始的示例。 primitive rsc_sap_S CT_ERS11 SAPInstance \ operations $id=rsc_s ap_SCT_ERS11-opera tions \ op monitor interval=120 timeout=60 on-fail=r estart \ params InstanceN ame=SCT_ERS11_sape rsvirt START_PRO FILE="/sapmnt/SCT/ profile/SCT_ERS11_ sapersvirt" \ AUTOMATIC_RECOVER= false IS_ERS=true \ meta priority=1000 若要升級至 ENSA2，請將此組態變更為下列設定。	

任務	描述	所需技能
	`primitive rsc_sap_SCT_ERS11 SAPInstance \ operations $id=rsc_sap_SCT_ERS11-operations \ op monitor interval=120 timeout=60 on-fail=restart \ params InstanceName=SCT_ERS11_sapersvirt START_PROFILE="/sapmnt/SCT/profile/SCT_ERS11_sapersvirt" \ AUTOMATIC_RECOVER=false IS_ERS=true` 您可以通過各種方式更改基元。例如，您可以在編輯器 (例如 vi) 中修改它們，如下列範例所示。 `crm configure edit rsc_sap_SCT_ERS11`	
停用維護模式。	停用叢集上的維護模式。 `crm configure property maintenance-mode="false"` 當叢集不在維護模式時，它會嘗試使用新的 ENSA2 設定使 ASCS 和 ERS 執行個體上線。	AWS 系統管理員

(選擇性) 新增叢集節點

任務	描述	所需技能
檢閱最佳做法。	在新增更多節點之前，請務必瞭解最佳作法，例如使用奇數或偶數節點。	AWS 系統管理員
加入節點。	新增更多節點需要執行一系列工作，例如更新作業系統、安裝符合現有節點的軟體套件，以及讓裝載可用。您可以使用 SAP 軟體啟動設定管理員 (SWPM) 中的 [準備其他主機] 選項來建立主機的 SAP 特定基準。如需詳細資訊，請參閱下一節所列的 SAP 指南。	AWS 系統管理員

相關資源

SAP 和 SUSE 參考資料

若要存取 SAP 備註，您必須擁有 SAP ONE Support 啟動台使用者帳戶。如需詳細資訊，請參閱 SAP 支援網站。

- ABAP NetWeaver 應用程式伺服器的說明文件
- 在瑞士醫管局環境中安裝 ENSA2 及由 ENSA1 更新至 ENSA2
- 使用紅帽 HA 解決方案時，安裝 ENSA2 並從 ENSA1 更新至 ENSA2
- SAP 注意事項 2711036-在高階管理局環境中使用獨立排入佇列伺服器 2
- 獨立式排入佇列伺服器 2 (SAP 文件)
- SAP S/4 HANA-排入佇列複寫 2 高可用性叢集-安裝指南 (SUSE 文件)

AWS 參考資料

- SAP HANA on AWS：適用於 SLES 和 RHEL 的高可用性組態指南
- SAP 鏡頭-AWS Well-Architected Framework

在不同 AWS 帳戶的 VPC 中使用一致的可用區域

由亞當·斯派塞（AWS）創建

程式碼儲存庫：多帳戶可用區域對應	環境：生產	技術：基礎設施
AWS 服務：AWS CloudFormation；Amazon VPC；AWS Lambda		

Summary

在 Amazon Web Services (AWS) 雲端上，可用區域的名稱可能會因 AWS 帳戶而異，而可用區域 ID (AZ ID) 則可識別其位置。如果您使用 AWS CloudFormation 建立虛擬私有雲端 (VPC)，則必須在建立子網路時指定可用區域的名稱或 ID。如果您在多個帳戶中建立 VPC，則可用區域名稱會隨機分配，這表示子網路在每個帳戶中使用不同的可用區域。

若要跨帳戶使用相同的可用區域，您必須將每個帳戶中的可用區域名稱對應至相同的 AZ ID。例如，下圖顯示 `use1-az6` AZ ID `us-east-1a` 在 AWS 帳戶 A 和 AWS 帳戶 Z `us-east-1c` 中命名。

此模式提供跨帳戶、可擴充的解決方案，以便在子網路中使用相同的可用區域，協助確保區域一致性。區域一致性可確保您的跨帳戶網路流量避免跨可用區域網路路徑，這有助於降低資料傳輸成本並降低工作負載之間的網路延遲。

此模式是 AWS CloudFormation AvailabilityZoneId 屬性的替代方法。

先決條件和限制

先決條件

- 在相同 AWS 區域中至少有兩個作用中的 AWS 帳戶。
- 評估需要多少可用區域才能支援該區域中的 VPC 需求。
- 識別並記錄您需要支援的每個可用區域的 AZ ID。如需詳細資訊，請參閱 AWS 資源存取管理員文件中 AWS 資源的可用區域 ID。

- AZ ID 的有序、逗號分隔清單。例如，清單上的第一個可用區域對應為`az1`，第二個可用性區域對應為，而此對應結構會繼續執行`az2`，直到您的逗號分隔清單完全對應為止。沒有可對應的 AZ ID 數目上限。
- 來自 GitHub 多帳戶可用區域對應存放庫的`az-mapping.yaml`檔案，已複製到您的本機電腦

架構

下圖顯示部署在帳戶中以及建立 AWS Systems Manager Parameter Store 值的架構。當您在帳戶中建立 VPC 時，會使用這些參數存放區值。

該圖顯示以下工作流程：

1. 此模式的解決方案部署到需要 VPC 區域一致性的所有帳戶。
2. 解決方案會為每個 AZ ID 建立參數存放區值，並儲存新的可用區域名稱。
3. AWS CloudFormation 範本使用存放在每個參數存放區值中的可用區域名稱，這可確保區域一致性。

下圖顯示了使用此模式的解決方案創建 VPC 的工作流程。

該圖顯示以下工作流程：

1. 向 AWS CloudFormation 提交用於建立 VPC 的範本。
2. AWS 會 CloudFormation 解析每個可用區域的參數存放區值，並傳回每個 AZ ID 的可用區域名稱。
3. VPC 會使用區域一致性所需的正確 AZ ID 來建立。

部署此病毒碼的解決方案之後，您可以建立參照參考參數存放區值的子網路。如果您使用 AWS CloudFormation，則可以參考下列 YAML 格式的範例程式碼中的可用區域對應參數值：

```
Resources:
    PrivateSubnet1AZ1:
        Type: AWS::EC2::Subnet
        Properties:
            VpcId: !Ref VPC
```

```
            CidrBlock: !Ref PrivateSubnetAZ1CIDR
            AvailabilityZone:
                !Join
                    - ''
                    - - '{{resolve:ssm:/az-mapping/az1:1}}'
```

此範例程式碼包含在來自 GitHub 多帳戶可用區域對應儲存庫的`vpc-example.yaml` 檔案中。它說明如何建立與參數存放區值對齊的 VPC 和子網路，以達到區域一致性。

技術, 堆

- AWS CloudFormation
- AWS Lambda
- AWS Systems Manager 參數存放區

自動化和規模

您可以使用 AWS CloudFormation StackSets 或 AWS 控制塔解決方案的自訂，將此模式部署到所有 AWS 帳戶。如需詳細資訊，請參閱 AWS Cloudformation 文件 CloudFormation StackSets中的使用 AWS 和 AWS 解決方案程式庫中的 AWS Control Tower 自訂。

部署 AWS CloudFormation 範本後，您可以更新範本以使用參數存放區值，並在管道或根據您的需求部署 VPC。

工具

AWS 服務

- AWS 可 CloudFormation協助您建立 AWS 資源的模型和設定、快速且一致地佈建它們，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- AWS Systems Manager Parameter Store 是 AWS Systems Manager 的一項功能。它為組態資料管理和機密管理提供安全的階層式儲存。

Code

此模式的程式碼在 GitHub 多帳戶可用區域對應儲存庫中提供。

史诗

部署 AZ 對應的 .yaml 檔案

任務	描述	所需技能
決定區域所需的可用區域。	1. 確定必須在您的區域中一致使用的 AZ ID。 2. 將這些 AZ ID 記錄在逗號分隔的清單中，並依照您要套用的順序來記錄。例如，清單上的第一個可用區域對應為`az1`，第二個區域對應為`az2`。沒有可對應的 AZ ID 數目上限。	雲端架構師
部署 AZ 對應的 .yaml 檔案。	使用`az-mapping.yaml` 檔案在所有必要的 AWS 帳戶中建立 AWS CloudFormation 堆疊。在`AZIds`參數中，使用您先前建立的逗號分隔清單。 我們建議您使用 AWS CloudFormation StackSets 或 AWS Control Tower 解決方案的自訂項目。	雲端架構師

在您的帳戶中部署 VPC

任務	描述	所需技能
自訂 AWS CloudFormation 範本。	使用 AWS 建立子網路時 CloudFormation，請自訂範本以使用您先前建立的參數存放區值。	雲端架構師

任務	描述	所需技能
	如需範例範本，請參閱 GitHub 多帳戶可用區域對應存放庫中的`vpc-example.yaml` 檔案。	
部署虛擬私人雲端。	將自訂的 AWS CloudFormation 範本部署到您的帳戶中。然後，區域中的每個 VPC 在用於子網路的可用區域中都具有區域一致性	雲端架構師

相關資源

- AWS 資源的可用區域 ID (AWS 資源存取管理員文件)
- AWS::EC2::Subnet(AWS CloudFormation 文件)

在本機驗證地形表單 (AFT) 程式碼的 Account Factory

由亞歷山大流行（AWS）和米哈爾·戈爾尼亞克（AWS）創建

環境：生產	技術：基礎設施 DevOps; 現代化; 軟件開發和測試	工作負載：開源
AWS 服務：AWS Control Tower		

Summary

此模式顯示如何在本機測試 HashiCorp 由 AWS Control Tower Account Factory 管理的地形表單程式碼 (AFT)。Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。AFT 設定 Terraform 管道，協助您在 AWS 控制塔中佈建和自訂多個 AWS 帳戶。

在程式碼開發期間，在 AFT 管線之外，在本機測試 Terraform 基礎結構做為程式碼 (IaC) 會很有幫助。此模式顯示了如何執行以下操作：

- 擷取存放在 AFT 管理帳戶中 AWS 儲存 CodeCommit 庫中的 Terraform 程式碼的本機副本。
- 使用擷取的程式碼在本機模擬 AFT 管線。

此程序也可以用來執行不屬於一般 AFT 管線一部分的 Terraform 命令。例如，您可以使用此方法執行命令`terraform validate`，例如`terraform plan`、`terraform destroy`、和`terraform import`。

先決條件和限制

先決條件

- 使用 AWS 控制塔的有效 AWS 多帳戶環境
- 完全部署的 AFT 環境
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 適用於程式碼提交、安裝和設定的 AWS CLI 登入資料助手

- Python 3. x
- Git，在本地計算機上安裝和配置
- git-remote-commit 公用程式, 已安裝和設定
- 地形，安裝和配置（本地 Terraform 軟件包版本必須與 AFT 部署中使用的版本匹配）

限制

- 此模式不涵蓋 AWS Control Tower、AFT 或任何特定 Terraform 模組所需的部署步驟。
- 在此程序期間本機產生的輸出不會儲存在 AFT 管線執行階段記錄中。

架構

目標技術堆疊

- 在 AWS Control Tower 部署中部署的 AFT 基礎設施
- 地形
- Git
- AWS CLI 第 2 版

自動化和規模

此模式顯示如何在單一 AFT 受管 AWS 帳戶中針對 AFT 全球帳戶自訂呼叫 Terraform 程式碼。驗證 Terraform 代碼後，您可以將其套用至多帳戶環境中的剩餘帳戶。如需詳細資訊，請參閱 AWS Control Tower 文件中的重新叫用自訂。

您也可以使用類似的程序，在本機終端機中執行 AFT 帳戶自訂。若要從 AFT 帳戶自訂本機叫用 Terraform 程式碼，請從 AFT 管理帳戶複製aft-global-account-customizations儲存aft-account-customizations 庫而非儲存庫。 CodeCommit

工具

AWS 服務

- AWS Control Tower 可協助您按照規範的最佳實務來設定和管理 AWS 多帳戶環境。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

其他服務

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。
- Git 是一個開放原始碼的分散式版本控制系統。

Code

以下是一個可用於本地運行由 AFT 管理的 Terraform 代碼的示例 bash 腳本。若要使用指令碼，請遵循此模式「Epics」一節中的指示。

```
#! /bin/bash
# Version: 1.1 2022-06-24 Unsetting AWS_PROFILE since, when set, it interferes with
 script operation
#          1.0 2022-02-02 Initial Version
#
# Purpose: For use with AFT: This script runs the local copy of TF code as if it were
 running within AFT pipeline.
#        * Facilitates testing of what the AFT pipline will do
#           * Provides the ability to run terraform with custom arguments (like 'plan'
 or 'move') which are currently not supported within the pipeline.
#
# © 2021 Amazon Web Services, Inc. or its affiliates. All Rights Reserved.
# This AWS Content is provided subject to the terms of the AWS Customer Agreement
# available at http://aws.amazon.com/agreement or other written agreement between
# Customer and either Amazon Web Services, Inc. or Amazon Web Services EMEA SARL or
 both.
#
# Note: Arguments to this script are passed directly to 'terraform' without parsing nor
 validation by this script.
#
# Prerequisites:
#    1. local copy of ct GIT repositories
#    2. local backend.tf and aft-providers.tf filled with data for the target account
 on which terraform is to be run
#       Hint: The contents of above files can be obtain from the logs of a previous
 execution of the AFT pipeline for the target account.
#    3. 'terraform' binary is available in local PATH
#    4. Recommended: .gitignore file containing 'backend.tf', 'aft_providers.tf' so the
 local copy of these files are not pushed back to git

readonly credentials=$(aws sts assume-role \
```

```
    --role-arn arn:aws:iam::$(aws sts get-caller-identity --query "Account" --output
 text ):role/AWSAFTAdmin \
    --role-session-name AWSAFT-Session \
    --query Credentials )

unset AWS_PROFILE
export AWS_ACCESS_KEY_ID=$(echo $credentials | jq -r '.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo $credentials | jq -r '.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo $credentials | jq -r '.SessionToken')
terraform "$@"
```

史诗

將範例程式碼儲存為本機檔案

任務	描述	所需技能
將範例程式碼儲存為本機檔案。	1. 複製此模式「代碼」部分中的示例 bash 腳本，然後將其粘貼到代碼編輯器中。 2. 將檔案命名為 `ct_terraform.sh` 。然後，將檔案儲存在本機專用資料夾中，例如`~/scripts` 或`~/bin`。	AWS 管理員
使範例程式碼可執行。	執行下列其中一個動作，開啟終端機視窗並在 AWS AFT 管理帳戶中進行驗證： • 使用已設定具有存取 AFT 管理帳戶所需許可的現有 AWS CLI 設定檔。若要使用設定檔，您可以執行下列命令： `export AWS_PROFILE=<aft account profile name>`	AWS 管理員

任務	描述	所需技能
	• 如果您的組織使用 SSO 存取 AWS，請在組織的 SSO 頁面上輸入 AFT 管理帳戶的登入資料。 注意：您的組織可能也有自訂工具，可為 AWS 環境提供身份驗證登入資料。	
在正確的 AWS 區域中驗證對 AFT 管理帳戶的存取權。	重要事項：請確定您使用的終端機工作階段與驗證 AFT 管理帳戶相同。 1. 執行下列命令，導覽至 AFT 部署的 AWS 區域： `export AWS_REGION=<aft_region>` 2. 請執行下列動作，確認您使用的帳戶正確無誤： • 執行以下命令： `aws code-commit list-repositories` • 然後，確認輸出中列出的存放庫與 AFT 管理帳戶中的存放庫名稱相符。	AWS 管理員
建立新的本機目錄以儲存 AFT 儲存庫程式碼。	在相同的終端工作階段中，執行下列命令： `mkdir my_aft` `cd my_aft`	AWS 管理員

任務	描述	所需技能
克隆遠程 AFT 存儲庫代碼。	1. 在本機終端機中，執行下列命令： `git clone codecommit::$AWS_REGION://aft-global-customizations` 注意：為了簡單起見，此過程和 AFT 僅使用主代碼分支。若要使用程式碼分支，您也可以在此處輸入程式碼分支指令。 但是，當 AFT 自動化應用來自主分支的代碼時，來自非主分支的任何應用更改都將被回滾。 2. 然後，執行下列命令，瀏覽至複製的目錄： `cd aft-global-customizations/terraform`	AWS 管理員

建立 AFT 管線在本機執行所需的 Terraform 組態檔案

任務	描述	所需技能
開啟先前執行的 AFT 管線，並將 Terraform 組態檔案複製到本機資料夾。	注意：AFT 管道在本地運行需要在此史詩中創建的後端 .tf 和後提供程序 .tf 配置文件。這些檔案會在雲端式 AFT 管線中自動建立，但必須手動建立，管線才能在本機執行。在本機執行 AFT 管道需要一組檔案，代	AWS 管理員

任務	描述	所需技能
	表在單一 AWS 帳戶內執行管道。 1. 使用您的 AWS Control Tower 管理帳戶登入資料登入 AWS 管理主控台。然後開啟 AWS CodePipeline 主控台。請確定您位於部署 AFT 的相同 AWS 區域。 2. 在左側導覽窗格中，選擇 Pipelines (管道)。 3. 選擇 ##########-定制管道。(########## 是您用來在本機執行 Terraform 程式碼的 AWS 帳戶識別碼)。 4. 確定「最近執行已標示」顯示「成功」值。如果值不同，您必須在 AFT 管線中重新叫用自訂。如需詳細資訊，請參閱 AWS Control Tower 文件中的重新叫用自訂。 5. 選擇最新的執行階段以顯示其詳細資料。 6. 在「應用-AFT-全局-自定義」部分中，找到「應用-地形」階段。 7. 選取「應用程式-地形」階段的「詳細資訊」區段。 8. 尋找「應用程式-地形」階段的執行階段記錄。	

任務	描述	所需技能
	9. 在運行時日誌中，查找以下行開頭和結尾的部分：「\ n \ n aft-Providers.tf...」\ n\ n 後端 .tf」 10. 複製這兩個標籤之間的輸出，並將其另存為本地 Terraform 文件夾（終端會話的當前工作目錄）`aft-providers.tf` 中命名的本地文件。 示例 auto 生成的提供程序 .tf 語句 `## Autogenerated providers.tf ## ## Updated on: 2022-05-31 16:27:45 ## provider "aws" { region = "us-east-2" assume_role { role_arn = "arn:aws:iam::######## :role/AWSAFTExecution" } default_tags { tags = { managed_by = "AFT" } } }` 11. 在執行階段記錄中，尋找以下列幾行開頭和結尾	

任務	描述	所需技能
	的區段：「\ n\ n tf...」\ n backup.tf」 12.複製這兩個標籤之間的輸出，並將其另存為本地 Terraform 文件夾（終端會話的當前工作目錄）tf中命名的本地文件。 自動生成的後端 .tf 語句示例 `## Autogenerated backend.tf ## ## Updated on: 2022-05-31 16:27:45 ## terraform { required_version = ">= 0.15.0" backend "s3" { region = "us-east-2" bucket = "aft-backend-###########-primary-region" key = "############-aft-global-customizations/terraform.tfstate" dynamodb_table = "aft-backend-#######" encrypt = "true" kms_key_id = "cbdc21d6-e04d-4c37-854f-51e199cfcb7c"`	

任務	描述	所需技能
	kms_key_id = "########-####-####-####-############" role_arn = "arn:aws:iam::#########:role/AWSAFTExecution" } } 注意：backend.tf 和aft-providers.tf 檔案會繫結至特定 AWS 帳戶、AFT 部署和資料夾。這些文件也有所不同，具體取決於它們是否位於同一 AFT 部署中的aft-account-customizations存儲aft-global-customizations庫和存儲庫中。請確定您從相同的執行階段清單產生這兩個檔案。	

使用範例 bash 指令碼在本機執行 AFT 管線

任務	描述	所需技能
實作您要驗證的 Terraform 組態變更。	1. 執行下列命令，導覽至複製的aft-global-customizations儲存庫： cd aft-global-customizations/terraform 注意：文件backend.tf 和aft-providers.tf 位於此目錄中。該目錄還包含來自存儲庫的	AWS 管理員

任務	描述	所需技能
	地形文件aft-global-customizations 。 2. 將您要在本機測試的 Terraform 程式碼變更合併到組態檔案中。	

任務	描述	所需技能
執行 ct_terraform.sh 指令碼並檢閱輸出。	1. 瀏覽至包含 sh 指令碼的本機資料夾。 2. 若要驗證修改過的 Terraform 程式碼，請執行下列`ct_terraform.sh` 命令來執行指令碼： `~/scripts/ct_terraform.sh apply` 注意：您可以在此步驟中運行任何 Terraform 命令。若要查看 Terraform 命令的完整清單，請執行下列命令： `terraform --help` 3. 檢閱命令輸出。然後，在提交並將更改推送回 AFT 存儲庫之前，在本地調試代碼更改。 重要： • 在本地進行且未推回到遠程存儲庫的任何更改都是臨時的，並且可以通過運行的 AFT 管道自動化隨時撤消。 • AFT 自動化可以隨時運行，因為它可以由其他用戶和 AFT 自動化觸發器調用。	AWS 管理員

任務	描述	所需技能
	• AFT 將始終從存儲庫的主分支應用代碼，撤消任何未提交的更改。	

提交並推送您的本地代碼更改回 AFT 存儲庫

任務	描述	所需技能
將後端 .tf 和後端提供者 .tf 檔案的參考資料新增到 .gitignore 檔案中。	執行下列指令，將您建立的`backend.tf` 和`aft-providers.tf` `.gitignore` 檔案新增至檔案： `echo backend.tf >> .gitignore` `echo aft-providers.tf >>.gitignore` 注意：將文件移動到文`.gitignore` 件可確保它們不會被提交並推回遠程 AFT 存儲庫。	AWS 管理員
將您的代碼更改提交並推送到遠程 AFT 存儲庫。	1. 若要將任何新的 Terraform 組態檔案新增至儲存庫，請執行下列命令： `git add <filename>` 2. 若要提交變更並將其推送至 AWS 中的遠端 AFT 儲存庫 CodeCommit，請執行下列命令： `git commit -a`	AWS 管理員

任務	描述	所需技能
	`git push` 重要事項：您遵循此程序所引入的程式碼變更，直到此時間僅套用至一個 AWS 帳戶為止。	

將變更推出至由 AFT 管理的多個帳戶

任務	描述	所需技能
對 AFT 管理的所有帳戶推出更改。	若要對 AFT 管理的多個 AWS 帳戶進行變更，請按照 AWS Control Tower 文件中重新叫用自訂項目中的指示進行操作。	AWS 管理員

更多模式

- 使用僅供讀取複本將 HA 新增至 Oracle PeopleSoft Amazon 自訂
- 使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目
- 自動化 AWS 資源評估
- 使用 AWS CDK 自動化 AWS 服務目錄產品組合和產品部署
- 使用 DR 協調器架構自動化跨區域容錯移轉和容錯回復
- ???
- 自動化跨 AWS 帳戶複寫 Amazon RDS 執行個體
- 使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔
- 使用 AWS CDK 為微型服務自動建置 CI/CD 管道和 Amazon ECS 叢集
- 自動檢測更改並為中的壟斷啟動不同的 CodePipeline 管道 CodeCommit
- ???
- 使用 AWS DataOps 開發套件建立資料管道以擷取、轉換和分析 Google 分析資料
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建置微焦點企業伺服器 PAC
- 使用 GitHub 動作和地形表單建置碼頭映像並將其推送到 Amazon ECR
- 使用 Terraform 在 AWS Organizations 中集中 IAM 存取金鑰管理
- 使用 Terraform 在 AWS Organizations 中集中軟體套件分發
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 使用混合式連結模式將資料中心擴充功能設定為 VMware Cloud on AWS
- 在 AWS 上的 SQL Server 中的「永遠開啟」可用性群組中設定唯讀路由
- ???
- 自動為 Java 和 Python 項目創建動態 CI 管道
- 在 AWS 上使用 VMware 雲端部署軟體定義的資料中心
- 使用私有端點和應用 Application Load Balancer 在內部網站上部署 Amazon API Gateway API
- 部署和偵錯 Amazon EKS 叢集
- 使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制 CloudFormation
- 使用地形表單部署和管理 AWS Control Tower 控制
- 使用地形部署 CloudWatch Synthetics 金絲雀
- 使用地形表單部署 AWS WAF 解決方案的安全自動化

- 記錄您的 AWS landing zone 設計
- 確保 IAM 設定檔與 EC2 執行個體相關聯
- 將 AWS Organizations 中各個組織的 AWS Backup 報告匯出為 CSV 檔案
- 使用 Amazon Personalize 個人化產生個人化和重新排名的建議
- 在未使用 AWS KMS 金鑰加密 Amazon 資料 Firehose 資源時識別並發出警示
- 使用啟動程序管道實作地形 (AFT) 的 Account Factory
- 使用 Kubernetes 在 Amazon EKS 工作者節點上安裝 SSM 代理程式 DaemonSet
- 在 Amazon EKS 工作者節點上安裝 SSM CloudWatch 代理程式和代理程式 preBootstrapCommands
- 將 VMware 網路洞察與 VMware Cloud on AWS 整合
- 管理多個 AWS 帳戶和 AWS 區域的 AWS 服務目錄產品
- 使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式
- 將 DNS 記錄批量遷移到 Amazon Route 53 私有託管區域
- 將 Oracle 電子商務套件遷移到 Amazon RDS 定制
- 將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制
- 使用 AWS MGN 將 RHEL BYOL 系統遷移到 AWS 包含授權的執行個體
- 使用 VMware 硬體校驗，將 VMware 軟體定義的軟體定義資料中心遷移至 VMware 雲端
- 監控 Amazon ElastiCache 叢集以進行靜態加密
- 監控安全群組的 ElastiCache 叢集
- 使用 AWS 服務監控 SAP RHEL 起搏器叢集
- 從多個 VPC 私有存取中央 AWS 服務端點
- 輪換資料庫認證而不重新啟動
- 建立 IAM 使用者時傳送通知
- 使用 VMware 詠嘆調操作的日誌，將日誌從 AWS 雲端傳送到潑濺
- 使用 AWS CDK 和在 Amazon ECS Anywhere 為混合式工作負載設定 CI/CD 管道 GitLab
- 在 AWS 上設定高可用性 PeopleSoft 架構
- ???
- 使用 NICE EnginFrame 和 NICE DCV 工作階段管理員設定 auto 調整規模的虛擬桌面基礎架構 (VDI)
- 在 Amazon RDS 上為甲骨文電子商務套件設置 HA/DR 架構，並使用活動備用數據庫
- 在多區域、多帳戶組織中設定 AWS CloudFormation 漂移偵測

- 使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施
- 在 Aurora 相容上設定甲骨文 UTL_FILE 功能
- 使用 AWS 私有 CA 和 AWS 記憶體簡化私有憑證管理
- 使用 AWS Organizations 自動標記 Transit Gateway 附件
- 甲骨文 PeopleSoft 應用程式在 Amazon RDS 自訂的轉換角色
- 使用 Serverspec 進行基礎架構程式碼的測試驅動開發

IoT

主題

- 針對 AWS IoT 環境中的安全事件設定記錄和監控
- 擷取和查詢資料湖中的 AWS IoT 中 SiteWise 繼資料屬性
- 使用用戶端裝置設定 AWS IoT Greengrass 並進行疑難排解
- 更多模式

針對 AWS IoT 環境中的安全事件設定記錄和監控

創建者普拉特克普拉卡什 (AWS)

環境：生產	技術：IoT；安全性、身分識別、合規性；營運	工作負載：所有其他工作
AWS 服務：Amazon CloudWatch; Amazon OpenSearch 服務; Amazon GuardDuty; AWS IoT Core; AWS IoT Device Defender; AWS IoT Device Managemen t; Amazon CloudWatch 日誌		

Summary

確保您的物聯網 (IoT) 環境安全是重要的優先事項，尤其是因為組織正在將數十億台裝置連接到其 IT 環境。此模式提供了一個參考架構，您可以用來在 Amazon Web Services (AWS) 雲端上實作 IoT 環境中的安全事件記錄和監控。一般而言，AWS 雲端上的 IoT 環境具有以下三層：

- 產生相關遙測資料的 IoT 裝置。
- 將您的 IoT 裝置連接到其他裝置和 AWS 服務的 AWS IoT 服務 (例如 AWS IoT AWS IoT Core、AWS IoT 裝置管理或 AWS IoT 裝置防禦者)。
- 後端 AWS 服務可協助處理遙測資料，並為您的不同商業使用案例提供有用的見解。

AWS IoT 鏡頭提供的最佳實務-AWS Well-Architected Framework 白皮書可協助您檢閱和改善雲端架構，並深入了解設計決策對業務的影響。重要的建議是分析裝置和 AWS 雲端上的應用程式日誌和指標。您可以利用不同的方法和技巧 (例如威脅模型) 來識別必須監控以偵測潛在安全性問題的指標和事件，藉此達成此目標。

此模式說明如何使用 AWS IoT 和安全服務，為 AWS 雲端上的 IoT 環境設計和實作安全記錄和監控參考架構。此架構建立在現有的 AWS 安全最佳實務之上，並將其套用至您的 IoT 環境。

先決條件和限制

先決條件

- 既有 landing zone 環境。如需詳細資訊，請參閱 AWS 規範指導網站上設定安全且可擴展的多帳戶 AWS 環境指南。
- 您的 landing zone 必須提供以下帳戶：
  - 記錄封存帳戶 — 此帳戶適用於需要存取登陸區域組織單位 (OU) 中帳戶記錄資訊的使用者。如需詳細資訊，請參閱 AWS AWS Prescriptive Guidance 網站上 AWS 安全參考架構指南的安全 OU — 日誌存檔帳戶部分。
  - 安全性帳戶 — 您的安全性與法規遵循團隊會使用此帳戶進行稽核或執行緊急安全性作業。這個帳戶也被指定為 Amazon 的管理員帳戶 GuardDuty。管理員帳戶的使用者除了檢視和管理自己帳戶和所有成員帳戶的 GuardDuty 發現項目之外 GuardDuty，還可以設定。如需有關此項目的詳細資訊，請參閱 Amazon GuardDuty 文件 GuardDuty中的管理多個帳戶。
  - IoT 帳戶 — 此帳戶適用於您的 IoT 環境。

架構

此模式可擴充 AWS 解決方案庫的集中式記錄解決方案，以收集和處理與安全相關的 IoT 事件。集中式記錄解決方案部署在安全帳戶中，可在單一儀表板中協助收集、分析和顯示 Amazon CloudWatch 日誌。此解決方案可整合、管理和分析來自多個來源的記錄檔。最後，集中式記錄解決方案也使用 Amazon OpenSearch 服務和OpenSearch 儀表板來顯示所有日誌事件的統一檢視。

下列架構圖顯示 AWS 雲端上 IoT 安全記錄和參考架構的關鍵元件。

該圖顯示以下工作流程：

1. IoT 物件是必須監控異常安全事件的裝置。這些裝置會執行代理程式，將安全事件或指標發佈到 AWS IoT Core 和 AWS IoT Device Defender。
2. 啟用 AWS IoT 記錄後，AWS IoT 會在每個訊息透過訊息代理程式和規則引擎從裝置傳送到 Amazon CloudWatch 日誌時，傳送有關每個訊息的進度事件。您可以使用記 CloudWatch 錄訂閱將事件推送至集中式記錄解決方案。如需詳細資訊，請參閱 AWS IoT 核心文件中的 AWS IoT 指標和維度。
3. AWS IoT Device Defender 可協助監控 IoT 裝置的不安全組態和安全指標。偵測到異常時，警示會通知亞馬遜簡單通知服務 (Amazon SNS)，該服務具有 AWS Lambda 函數做為訂閱者。Lambda 函數會將警示做為訊息傳送至 CloudWatch 記錄檔。您可以使用記 CloudWatch 錄訂閱將事件推送至

集中式記錄解決方案。如需詳細資訊，請參閱 AWS IoT Core 文件中的稽核檢查、裝置端指標和雲端指標。

4. AWS CloudTrail 記錄進行變更的 AWS IoT 核心控制平面動作 (例如，建立、更新或附加 API)。當設定 CloudTrail 為 landing zone 實作的一部分時，它會將事件傳送至記錄 CloudWatch 檔，而且您可以使用訂閱將事件推送至集中式記錄解決方案
5. AWS Config 受管規則或自訂規則會評估屬於您 IoT 環境的資源。使用「 CloudWatch 記錄檔」做為目標的「 CloudWatch 事件」來監視符合性變更通知。將合規性變更通知傳送至記錄 CloudWatch 檔後，您可以使用訂閱將事件推送至您的集中式記錄解決方案。
6. Amazon 會 GuardDuty 持續分析 CloudTrail 管理事件，並協助識別來自已知惡意 IP 地址、異常地理位置或匿名代理伺服器對 AWS IoT Core 端點進行的 API 呼叫。使用 Amazon CloudWatch 事件監控 GuardDuty 通知，並將日誌群組中的 CloudWatch 日誌群組做為目標。當 GuardDuty 通知傳送至 CloudWatch 記錄檔時，您可以使用訂閱將事件推送至您的集中式監控解決方案，或使用安全性帳戶中的 GuardDuty 主控台來檢視通知。
7. AWS Security Hub 使用安全最佳實務來監控您的 IoT 帳戶。使用 [記錄檔] 中的 CloudWatch 記錄群組做為目標的 CloudWatch 事件來監視 Security Hub 通知。當 Security Hub 通知傳送至 CloudWatch 記錄檔時，請使用訂閱將事件推送至您的集中式監控解決方案，或使用安全性帳戶中的 Security Hub 主控台來檢視通知。
8. Amazon Detective ess 會評估和分析資訊，以隔離根本原因，並針對異常呼叫 AWS IoT 端點或 IoT 架構中其他服務的安全發現採取行動。
9. Amazon Athena 會查詢存放在日誌存檔帳戶中的日誌，以加強您對安全發現結果的瞭解，並識別趨勢和惡意活動。

工具

- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中輕鬆分析資料。
- AWS 可 CloudTrail協助您啟用 AWS 帳戶的管控、合規以及操作和風險稽核。
- Amazon 會即時 CloudWatch監控您的 AWS 資源和您在 AWS 上執行的應用程式。 您可以用 CloudWatch 來收集和追蹤指標，這些指標是您可以針對資源和應用程式測量的變數。
- Amazon CloudWatch Logs 會集中您使用的所有系統、應用程式和 AWS 服務的日誌。您可以檢視和監視記錄、搜尋特定錯誤代碼或模式、根據特定欄位對其進行篩選，或安全地將其封存以供日 future 分析。
- AWS Config 在您 AWS 帳戶內提供 AWS 資源組態的詳細檢視。
- Amazon Detective tor 可讓您輕鬆分析、調查並快速識別安全發現結果或可疑活動的根本原因。

- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務，可讓您以簡單且經濟實惠的方式將資料分類、清理資料、豐富資料，以及在各種資料存放區和資料串流之間可靠地移動資料。
- Amazon GuardDuty 是一種持續的安全監控服務。
- AWS IoT Core 為連線網際網路的裝置 (例如感應器、致動器、嵌入式裝置、無線裝置和智慧設備) 提供安全的雙向通訊，以透過 MQTT、HTTPS 和 WAN 連線到 AWS 雲端。 LoRa
- AWS IoT Device Defender 是一項安全服務，可讓您稽核裝置的組態、監控連線裝置以偵測異常行為，並降低安全風險。
- Amazon Ser OpenSearch vice 是一種受管服務，可讓您在 AWS 雲端輕鬆部署、操作和擴展 OpenSearch 叢集。
- AWS Organizations Organization 是一種帳戶管理服務，可讓您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Security Hub 為您提供 AWS 安全狀態的全面檢視，並協助您根據安全產業標準和最佳實務檢查環境。
- Amazon Virtual Private Cloud (Amazon VPC) 佈建 AWS 雲端的邏輯隔離部分，您可以在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 AWS 可擴展基礎設施的優勢。

史诗

在您的 landing zone 環境中設定 IoT 帳戶

任務	描述	所需技能
驗證 IoT 帳戶中的安全護欄。	驗證您的 IoT 帳戶中已啟用 AWS Config 和安全中心的護欄。 CloudTrail GuardDuty	AWS 管理員
驗證您的 IoT 帳戶設定為安全性帳戶的成員帳戶。	驗證您的 IoT 帳戶已設定並關聯為您安全性帳戶中 GuardDuty 的安全性中心的成員帳戶。 如需詳細資訊，請參閱 Amazon GuardDuty 文件中的使用 AWS Organizations	AWS 管理員

任務	描述	所需技能
	管理 GuardDuty 帳戶和 AWS Security Hub 文件中的管理管理員和成員帳戶。	
驗證記錄封存。	驗證 CloudTrail AWS Config 和 VPC 流程日誌是否存放在日誌存檔帳戶中。	AWS 管理員

設定集中式記錄解決方案

任務	描述	所需技能
在您的安全性帳戶中設定集中式記錄解決方案。	登入安全帳戶的 AWS 管理主控台，並從 AWS 解決方案程式庫設定集中式記錄解決方案，以收集、分析和顯示 Amazon OpenSearch 服務和 OpenSearch 儀表板中的 CloudWatch 日誌。 如需詳細資訊，請參閱 AWS 解決方案程式庫中的集中 CloudWatch 日誌記錄實作指南，在單一儀表板中收集、分析和顯示 Amazon 日誌。	AWS 管理員

在您的 IoT 帳戶中設定和設定 AWS 資源

任務	描述	所需技能
設定 AWS IoT 記錄。	為您的 IoT 帳戶登入 AWS 管理主控台。設定和設定 AWS IoT Core，以便將日誌傳送到 CloudWatch 日誌。	AWS 管理員

任務	描述	所需技能
	如需詳細資訊，請參閱 AWS IoT Core 文件中的設定 AWS IoT 記錄和使用 CloudWatch 日誌監控 AWS IoT。	
設定 AWS IoT Device Defender。	設定 AWS IoT Device Defender 來稽核您的 IoT 資源並偵測異常情況。 如需有關這方面的詳細資訊，請參閱 AWS IoT 核心文件中的 AWS IoT 裝置防禦者入門。	AWS 管理員
設定 CloudTrail。	設定 CloudTrail 將事件傳送至 CloudWatch 記錄檔。 如需詳細資訊，請參閱 AWS CloudTrail 文件中的將事件傳送至 CloudWatch 日誌。	AWS 管理員
設定 AWS Config 和 AWS Config 規則。	設定 AWS Config 和所需的 AWS Config 規則。如需詳細資訊，請參閱 AWS Config 文件中的使用主控台設定 AWS Config 和使用主控台設定 AWS Config 規則。	AWS 管理員

任務	描述	所需技能
設定 GuardDuty。	設定並設定 GuardDuty 以將發現項目傳送至 Amazon CloudWatch 事件，並將日誌群組中的 CloudWatch 日誌群組做為目標。 如需這方面的詳細資訊，請參閱 Amazon GuardDuty 文件中的使用 Amazon E CloudWatch vents 建立對 GuardDuty 發現項目的自訂回應。	AWS 管理員
設定 Security Hub。	設定 Security Hub 並啟用 CIS AWS 基礎基準測試和 AWS 基礎安全最佳實務標準。 如需相關資訊，請參閱 AWS Security Hub 文件中的自動回應和修復。	AWS 管理員
成立 Amazon Detective。	設立 Detective 以協助分析保安結果 有關這方面的更多信息，請參閱 Amazon Detective 文檔中的設置 Amazon Detective。	AWS 管理員
設置 Amazon Athena 和 AWS AWS Glue。	設定 Athena 和 AWS Glue 以查詢執行安全事件調查的 AWS 服務日誌。 如需詳細資訊，請參閱 Amazon Athena 文件中的查詢 AWS 服務日誌。	AWS 管理員

相關資源

- 什麼是 landing zone？

擷取和查詢資料湖中的 AWS IoT 中 SiteWise 繼資料屬性

創建者：安巴里斯東加卡爾 (AWS)

環境：生產	技術：IoT；分析；大數據	AWS 服務：AWS IoT SiteWise；AWS Lambda；AWS AWS Glue

Summary

AWS IoT SiteWise 使用資產模型和階層來代表您的工業設備、流程和設施。每個模型或資產都可以有多個特定於您環境的屬性。中繼資料屬性範例包括資產的場地或實體位置、工廠詳細資料以及設備識別碼。這些屬性值可補充資產測量資料，以最大化商業價值。機器學習 (ML) 可提供此中繼資料的其他見解，並簡化工程工作。

不過，無法直接從 AWS IoT SiteWise 服務查詢中繼資料屬性。若要使屬性可查詢，您必須擷取屬性並將其擷取到資料湖中。此模式使用 Python 指令碼擷取所有 AWS IoT SiteWise 資產的屬性，並將它們導入 Amazon Simple Storage Service (Amazon S3) 貯體中的資料湖。完成此程序後，您可以使用 Amazon Athena 中的 SQL 查詢來存取 AWS IoT 中 SiteWise 繼資料屬性和其他資料集，例如測量資料集。使用 AWS IoT SiteWise 監視器或儀表板時，中繼資料屬性資訊也很有用。您也可以使用 S3 儲存貯體中的擷取屬性來建立 AWS QuickSight 儀表板。

該模式具有參考代碼，您可以使用適合您使用案例的最佳運算服務來實作程式碼，例如 AWS Lambda 或 AWS Glue。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 設定 AWS Lambda 函數或 AWS AWS Glue 任務的許可。
- Amazon S3 儲存貯體。
- 資產模型和階層是在 AWS IoT 中設定的 SiteWise。如需詳細資訊，請參閱建立資產模型 (AWS IoT SiteWise 文件)。

架構

您可以使用 Lambda 函數或 AWS Glue 任務來完成此程序。如果您的模型少於 100 個，且每個模型的屬性平均為 15 或更少，我們建議您使用 Lambda。對於所有其他使用案例，我們建議使用 AWS Glue。

解決方案架構和工作流程如下圖所示。

1. 排程的 AWS AWS Glue 任務或 Lambda 函數會執行。它會從 AWS IoT 擷取資產中繼資料屬性，SiteWise 並將其導入 S3 儲存貯體。
2. AWS Glue 爬行者程式會在 S3 儲存貯體中檢索擷取的資料，並在 AWS Glue 資料型錄中建立表格。
3. Amazon Athena 使用標準 SQL 查詢 AWS Glue 資料型錄中的表格。

自動化和規模

您可以根據 AWS IoT SiteWise 資產組態的更新頻率，排程 Lambda 函數或 AWS Glue 任務每天或每週執行。

範例程式碼可處理的 AWS IoT SiteWise 資產數量沒有限制，但是大量資產可能會增加完成程序所需的時間。

工具

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中分析資料。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS IoT 可 SiteWise協助您大規模收集、建立模型、分析和視覺化來自工業設備的資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

- 適用於 Python 的 AWS 開發套件 (Boto3) 是一套軟體開發套件，可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。

史诗

設定工作或功能

任務	描述	所需技能
在 IAM 中設定許可。	在 IAM 主控台中，將許可授與 Lambda 函數或 AWS Glue 任務假定的 IAM 角色，以執行下列動作： • 讀取 AWS IoT SiteWise 服務的內容 • 寫入 S3 儲存貯體 如需詳細資訊，請參閱為 AWS 服務建立角色 (IAM 文件)。	一般 AWS
建立 Lambda 函數或 AWS AWS Glue 任務。	如果您使用的是 Lambda，請建立新的 Lambda 函數。在「執行階段」中，選擇 Python。如需詳細資訊，請參閱使用 Python 建置 Lambda 函數 (Lambda 說明文件)。 如果您使用 AWS Glue，請在 AWS Glue 主控台中建立新的 Python 殼層任務。如需詳細資訊，請參閱新增 Python 殼層任務 (AWS Glue 文件)。	一般 AWS
更新 Lambda 函數或 AWS AWS Glue 任務。	修改新的 Lambda 函數或 AWS Glue 任務，然後在其他資訊區段中輸入程式碼範例。	一般 AWS

任務	描述	所需技能
	根據您的使用案例的需要修改程式碼。如需詳細資訊，請參閱使用主控台編輯器編輯程式碼 (Lambda 文件) 和使用指令碼 (AWS Glue 文件)。	

執行工作或功能

任務	描述	所需技能
執行 Lambda 函數或 AWS AWS Glue 任務。	執行 Lambda 函數或 AWS AWS Glue 任務。如需詳細資訊，請參閱叫用 Lambda 函數 (Lambda 文件) 或使用觸發器啟動任務 (AWS Glue 文件)。這會擷取 AWS IoT SiteWise 階層中資產和模型的中繼資料屬性，並將其存放在指定的 S3 儲存貯體中。	一般 AWS
設定 AWS Glue 合爬蟲程式。	使用 CSV 格式檔案所需的格式分類器設定 AWS Glue 爬蟲程式。使用 Lambda 函數或 AWS Glue 任務中使用的 S3 儲存貯體和前置詞詳細資訊。如需詳細資訊，請參閱定義爬蟲程式 (AWS Glue 文件)。	一般 AWS
執行 AWS Glue 爬蟲程式。	執行搜尋器以處理 Lambda 函數或 AWS Glue 任務所建立的資料檔案。爬行者程式會在指定的 AWS Glue 資料型錄中建立資料表。如需詳細資訊，請參閱或使用觸發器啟動檢索器 (AWS Glue 文件)。	一般 AWS

任務	描述	所需技能
查詢中繼資料屬性。	使用 Amazon Athena，根據您的使用案例的需要使用標準 SQL 查詢 AWS Glue 資料型錄。您可以將中繼資料屬性表格與其他資料庫和資料表聯結。如需詳細資訊，請參閱入門 (Amazon Athena 文件)。	一般 AWS

相關資源

- Amazon Athena 文
- AWS AWS Glue 文件
- AWS IoT SiteWise API 參考
- AWS IoT SiteWise 使用者指南
  - 入門
  - 建模工業資產
  - 定義資產模型 (階層) 之間的關係
  - 關聯和取消關聯資產
  - 建立 AWS IoT SiteWise 示範
- 物聯網 SiteWise (開發套件適用於 Python 文件)
- Lambda 文件

其他資訊

Code

提供的範例程式碼僅供參考，您可以根據使用案例的需要自訂此程式碼。

```
# Following code can be used in an AWS Lambda function or in an AWS Glue Python shell
 job.
# IAM roles used for this job need read access to the AWS IoT SiteWise service and
 write access to the S3 bucket.
sw_client = boto3.client('iotsitewise')
```

```
s3_client = boto3.client('s3')
output = io.StringIO()

attribute_list=[]
bucket = '{3_bucket name}'
prefix = '{s3_bucket prefix}'
output.write("model_id,model_name,asset_id,asset_name,attribuet_id,attribute_name,attribute_val
\n")

m_resp = sw_client.list_asset_models()
for m_rec in m_resp['assetModelSummaries']:
     model_id = m_rec['id']
     model_name = m_rec['name']

     attribute_list.clear()
     dam_response = sw_client.describe_asset_model(assetModelId=model_id)
     for rec in dam_response['assetModelProperties']:
         if 'attribute' in rec['type']:
            attribute_list.append(rec['name'])

     response = sw_client.list_assets(assetModelId=model_id, filter='ALL')
     for asset in response['assetSummaries']:
         asset_id = asset['id']
         asset_name = asset['name']
         resp = sw_client.describe_asset(assetId=asset_id)
         for rec in resp['assetProperties']:
            if rec['name'] in attribute_list:
                p_resp = sw_client.get_asset_property_value(assetId=asset_id,
 propertyId=rec['id'])
                if 'propertyValue' in p_resp:
                    if p_resp['propertyValue']['value']:
                        if 'stringValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['stringValue']) + "\n")

                        if 'doubleValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['doubleValue']) + "\n")
                        if 'integerValue' in p_resp['propertyValue']['value']:
                             output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['integerValue']) + "\n")
```

```
                        if 'booleanValue' in p_resp['propertyValue']['value']:
                            output.write(model_id + "," + model_name + ","
 + asset_id + "," + asset_name + "," + rec['id'] + "," + rec['name'] + "," +
 str(p_resp['propertyValue']['value']['booleanValue']) + "\n")

output.seek(0)
s3_client.put_object(Bucket=bucket, Key= prefix + '/data.csv', Body=output.getvalue())
output.close()
```

使用用戶端裝置設定 AWS IoT Greengrass 並進行疑難排解

由馬魯安·塞菲亞尼和阿卡蘭卡·德席爾瓦（AWS）創建

環境：PoC 或試點	技術：IoT	AWS 服務：AWS IoT Core；AWS 物聯網核心

Summary

AWS IoT Greengrass 是一種開放原始碼邊緣執行階段和雲端服務，用於在邊緣裝置上建置、部署和管理物聯網 (IoT) 軟體。AWS 物聯網網路環境的使用案例包括：

- 使用 AWS IoT Greengrass 閘道作為家庭自動化中樞的智慧家庭
- AWS IoT Greengrass 可協助從現場擷取和本機處理資料的智慧工廠

AWS IoT Greengrass 可做為其他邊緣裝置 (也稱為用戶端裝置) 的安全、經驗證的 MQTT 連線端點，否則通常會直接連線到 AWS IoT Core。當用戶端裝置無法直接存取 AWS IoT Core 端點的網路時，此功能非常有用。

您可以針對下列使用案例設定 AWS IoT Greengrass 以搭配用戶端裝置使用：

- 對於用戶端裝置將資料傳送到 AWS IoT Greengrass
- 讓 AWS IoT Greengrass 路轉寄資料到 AWS IoT Core
- 利用進階 AWS IoT Core 規則引擎功能

這些功能需要在 AWS IoT Greengrass 裝置上安裝和設定下列元件：

- MQTT 經紀商
- MQTT 大橋
- 用戶端裝置驗證
- IP 偵測器

此外，來自用戶端裝置的已發佈訊息必須採用 JSON 格式或通訊協定緩衝區 (protobuf) 格式。

此模式說明如何安裝和設定這些必要元件，並提供疑難排解提示和最佳做法。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版
- 兩個執行 Python 3.7 或更新版本的用戶端裝置
- 一個運行 Java 運行時環境（JRE）版本 8 或更高版本的核心設備，以及 Amazon 線 11 或 OpenJDK 11

限制

- 您必須選擇可使用 AWS IoT 核心的 AWS 區域。如需目前 AWS IoT Core 的區域清單，請參閱 AWS 服務 (按區域分類)。
- 核心裝置必須至少有 172 MB 記憶體和 512 MB 的磁碟空間。

架構

下圖顯示此模式的解決方案架構。

該架構包括：

- 兩個用戶端裝置。每個裝置都包含私密金鑰、裝置憑證和根憑證授權單位 (CA) 憑證。包含 MQTT 用戶端的 AWS IoT 裝置開發套件也會安裝在每個用戶端裝置上。
- 已部署 AWS IoT Greengrass 的核心裝置，其中包含下列元件：
  - MQTT 經紀商
  - MQTT 大橋
  - 用戶端裝置驗證
  - IP 偵測器

此架構支援下列案例：

- 用戶端裝置可以使用其 MQTT 用戶端，透過核心裝置的 MQTT 代理程式彼此通訊。
- 用戶端裝置也可以透過核心裝置的 MQTT 代理程式和 MQTT 橋接器與雲端中的 AWS IoT Core 通訊。
- 雲端中的 AWS IoT Core 可透過 MQTT 測試用戶端以及核心裝置的 MQTT 橋接器和 MQTT 代理程式，將訊息傳送到用戶端裝置。

如需有關用戶端裝置與核心裝置之間通訊的詳細資訊，請參閱其他資訊一節。

工具

AWS 服務

- AWS IoT Greengrass 是開放原始碼物聯網 (IoT) 邊緣執行階段和雲端服務，可協助您在裝置上建置、部署和管理 IoT 應用程式。
- AWS IoT Core 為連線到網際網路的裝置提供安全的雙向通訊，以連線到 AWS 雲端。
- AWS IoT Device SDK 是一套軟體開發套件，其中包含開放原始碼程式庫、含範例的開發人員指南，以及移植指南，讓您可以在自己選擇的硬體平台上建置創新的 IoT 產品或解決方案。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

最佳實務

- 來自用戶端裝置的訊息承載應採用 JSON 或 Protobuf 格式，才能利用 AWS IoT Core 規則引擎的進階功能，例如轉換和條件式動作。
- 設定 MQTT 橋接器以允許雙向通訊。
- 在 AWS IoT Greengrass 中設定和部署 IP 偵測器元件，以確保核心裝置的 IP 位址包含在 MQTT 代理程式憑證的主體替代名稱 (SAN) 欄位中。

史诗

設定核心裝置

任務	描述	所需技能
在您的核心裝置上設 AWS IoT Greengrass 境。	遵循開發人員指南中的指示，安裝 AWS IoT Greengrass 核心軟體。	AWS IoT Greengrass
檢查您的安裝狀態。	使用下列命令檢查核心裝置上 AWS IoT Greengrass 服務的狀態： `sudo systemctl status greengrass.service` 該命令的預期輸出是： `Launched Nucleus successfully`	一般 AWS
設定 IAM 政策並將其附加到 Greengrass 服務角色。	1. 建立 IAM 政策以允許進出 MQTT 橋接器的通訊。以下是一個示例策略： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iot:*" ],`	一般 AWS

任務	描述	所需技能

```
            "Resource
": "*"
        },
        {
            "Sid":
 "GreengrassActions
",
            "Effect":
 "Allow",
            "Action":
 [

 "greengrass:*"
            ],
            "Resource
": "*"
        }
    ]
}
```

2. 將原則附加至 Greengrass 服務角色。若要取得服務角色，請使用下列命令：

```
aws greengrassv2
 get-service-role-f
or-account --region
 <region>
```

其中<region>指的是您的 AWS 區域。

任務	描述	所需技能
在 AWS IoT Greengrass 核心裝置中設定和部署必要的元件。	設定及部署下列元件： • `greengrass.clientdevices.mqtt.Moquette`（請參閱組態詳細資料） • `greengrass.clientdevices.mqtt.Bridge`（請參閱配置詳細信息和下一個任務） • `greengrass.clientdevices.Auth`（請參閱配置詳細信息和下一個任務之後的任務） • `aws.greengrass.clientdevices.IPDetector`（請參閱組態詳細資料）	AWS IoT Greengrass

任務	描述	所需技能
確認 MQTT 橋接器允許雙向通訊。	若要在用戶端裝置和 AWS IoT Core 之間轉送 MQTT 訊息，請設定和部署 MQTT 橋接器元件，並指定要轉送的主題。範例如下： `{ "mqttTopicMapping": { "ClientDevicesToCloud": { "topic": "dt/#", "source": "LocalMqtt", "target": "IotCore" }, "CloudToClientDevices": { "topic": "cmd/#", "source": "IotCore", "target": "LocalMqtt" } } }`	AWS IoT Greengrass

任務	描述	所需技能
確認 auth 組件允許客戶端設備連接和發布或訂閱主題。	下列`aws.greengrass.clientdevices.Auth` 組態可讓所有用戶端裝置連線、發佈訊息和訂閱所有主題。 `{ "deviceGroups": { "formatVersion": "2021-03-05", "definitions": { "MyPermissiveDeviceGroup": { "selectionRule": "thingName: *", "policyName": "MyPermissivePolicy" } }, "policies": { "MyPermissivePolicy": { "AllowAll": { "statementDescription": "Allow client devices to perform all actions.", "operations": ["*"], "resources": ["*"] } } } } }`	AWS IoT Greengrass

任務	描述	所需技能
	`}`	

設定用戶端裝置

任務	描述	所需技能
安裝 AWS IoT 裝置開發套件。	在用戶端裝置上安裝 AWS IoT 裝置開發套件。如需支援語言和相關開發套件的完整清單，請參閱 AWS IoT Core 文件。 例如，適用於 Python 開發套件的 AWS IoT 裝置開發套件位於上 GitHub。若要安裝此 SDK： 1. 確認已安裝 Python 3.7 或更新版本，按照 GitHub 存放庫的 [必要條件] 頁面上的指示。 2. 使用 pip 命令來安裝 SDK。 對於 MacOS 系統和 Linux: `python3 -m pip install awsiotsdk` 針對 Windows： `python -m pip install awsiotsdk` 或者，您也可以從來源儲存庫安裝 SDK：	一般 AWS IoT

任務	描述	所需技能
	`# Create a workspace directory to hold all the SDK files` `mkdir sdk-workspace` `cd sdk-workspace` `# Clone the repository` `git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git` `# Install using Pip (use 'python' instead of 'python3' on Windows)` `python3 -m pip install ./aws-iot-device-sdk-python-v2`	

任務	描述	所需技能
創建一個東西。	1. 在 AWS IoT 主控台中，如果出現 [開始使用] 按鈕，請選擇該按鈕。否則，請在功能窗格中選擇 [安全性]、[原則]。 2. 如果 [您還沒有任何策略] 對話方塊出現，請選擇 [建立策略]。否則，請選擇 Create (建立)。 3. 輸入 AWS IoT 政策的名稱 (例如，`ClientDevicePolicy`)。 4. 在 [新增陳述式] 區段中，以下列 JSON 程式碼取代現有原則。`<account>` 以您`<region>`的 AWS 區域和 AWS 帳戶號碼取代和。 `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "iot:Connect", "Resource": "arn:aws:iot:region:account:client/*" }, { "Effect": "Allow", "Action": "iot:Publish", "Resource": "*"`	AWS IoT Core

任務	描述	所需技能
	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "iot:Connect", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Receive", "Resource": "*" }, { "Effect": "Allow", "Action": "iot:Subscribe", "Resource": "*" }, { "Effect": "Allow", "Action": ["iot:GetThingShadow", "iot:UpdateThingShadow", "iot:DeleteThingShadow"], "Resource": "arn:aws:iot:region:account:thing/*" }] }</pre> 5. 選擇建立。 6. 在 AWS IoT 主控台的導覽窗格中，選擇管理，物件。 7. 如果顯示 [您尚未擁有任何物件] 對話方塊，請選擇	

任務	描述	所需技能
	[註冊物件]。否則，請選擇 Create (建立)。 8. 在 Creating AWS IoT things (建立 AWS IoT 物件) 頁面上，選擇 Create a single thing (建立單一物件)。 9. 在 Add your device to the device registry (將裝置新增至裝置登錄檔) 頁面上，輸入您 IoT 物件的名稱 (例如 `ClientDevice1`)，然後選擇 Next (下一步)。 附註：您無法在建立物件之後變更該物件的名稱。若要變更名稱，您必須建立新物件，為其指定新名稱，然後刪除舊物件。 10.在 Add a certificate for your thing (新增物件的憑證) 頁面上，選擇 Create certificate (建立憑證)。 11.選擇 Download (下載) 連結來下載憑證、私有金鑰和根憑證授權機構憑證。 重要事項：這是您下載憑證和私密金鑰的唯一機會。 12.若要啟用憑證，請選擇 Activate (啟用)。憑證必須處於作用中狀態，裝置才能連線到 AWS IoT。 13.選擇 Attach a policy (連接政策)。	

任務	描述	所需技能
	14針對 [新增物件的原則] ClientDevicePolicy，選擇 [註冊物件]。	
從核心裝置下載 CA 憑證。	如果您希望 Greengrass 核心裝置可以在離線環境中運作，則必須將 Greengrass 核心 CA 憑證提供給用戶端裝置，以便它可以驗證 MQTT 代理程式的憑證 (由 Greengrass 核心 CA 核心 CA 核心)。因此，取得此憑證的副本非常重要。使用下列其中一種方法來下載 CA 憑證： • 如果您可以從電腦存取 AWS IoT Greengrass 裝置，請`https://<device IP>:8883`在網頁瀏覽器中輸入並檢視 MQTT 代理程式憑證和 CA 憑證。您也可以將 CA 憑證儲存到用戶端裝置。 • 或者，您也可以使用 OpenSSL 命令列： `openssl s_client -showcerts -connect <device IP>:8883`	一般 AWS
複製用戶端裝置中的認證。	複製 Greengrass 核心 CA 憑證、裝置憑證和用戶端裝置中的私密金鑰。	一般 AWS

任務	描述	所需技能
建立用戶端裝置與核心裝置的關聯。	將用戶端裝置與核心裝置建立關聯，以便他們可以探索核心裝置。然後，用戶端裝置可以使用 Greengrass 探索 API 擷取其關聯核心裝置的連線資訊和憑證。如需詳細資訊，請參閱 AWS IoT Greengrass 文件中的關聯用戶端裝置。 1. 在 AWS IoT Greengrass 主控台上，選擇核心裝置。 2. 選擇要管理的核心裝置。 3. 在核心裝置的詳細資料頁面上，選擇 [用戶端裝置] 索引標籤。 4. 在關聯的用戶端裝置區段中，選擇關聯用戶端裝置。 5. 在「將用戶端裝置與核心裝置建立關聯」模式中，針對要關聯的每個用戶端裝置執行下列動作 a. 輸入要建立為用戶端裝置關聯的 AWS IoT 物件的名稱。 b. 選擇新增。 6. 選擇 Associate (關聯)。 您關聯的用戶端裝置現在可以使用 Greengrass 探索 API 來探索此核心裝置。	AWS IoT Greengrass

傳送和接收資料

任務	描述	所需技能
將資料從一個用戶端裝置傳送到另一個用戶端裝置。	使用裝置中的 MQTT 用戶端發佈有關該主題的`dt/client 1/sensor` 訊息。	一般 AWS
將資料從用戶端裝置傳送到 AWS IoT Core。	使用裝置中的 MQTT 用戶端發佈有關該主題的`dt/client 1/sensor` 訊息。 在 MQTT 測試用戶端中，訂閱裝置傳送訊息的主題，或訂閱 `#` 以取得所有主題 (請參閱詳細資訊)。	一般 AWS
將訊息從 AWS IoT Core 傳送到用戶端裝置。	在 MQTT 測試用戶端頁面的 [發佈至主題] 索引標籤的 [主題名稱] 欄位中，輸入郵件的主題名稱。在此範例中，請用`cmd/client1` 於主題。	一般 AWS

故障診斷

問題	解決方案
無法驗證伺服器憑證錯誤	當 MQTT 用戶端無法在 TLS 交涉期間驗證 MQTT 代理程式提供的憑證時，就會發生這個錯誤。最常見的原因是 MQTT 用戶端沒有 CA 憑證。請依照下列步驟確定已將 CA 憑證提供給 MQTT 用戶端。 1. 如果您可以從電腦存取 AWS IoT Greengrass 裝置，請`https://<device IP>:8883` 在瀏覽器視窗中輸入以檢視 MQTT 代理程式憑

問題	解決方案
	證和 CA 憑證。您也可以將 CA 憑證儲存到用戶端裝置。 或者，您也可以使用 OpenSSL 命令列： `openssl s_client -showcerts -connect <device IP>:8883` 2. 將 Moquette CA 和 Greengrass 核心 CA 證書的內容保存到文件中，然後使用以下命令查看解碼的內容： `openssl x509 -in <Name of CA>.pem -text` 模型 CA 憑證應該會顯示 SAN 欄位，如下列範例所示： `X509v3 Subject Alternative Name: IP Address:XXX.XXX.XXX.XXX, IP Address:127.0.0.1, DNS:localhost`
無法驗證伺服器名稱錯誤	當 MQTT 用戶端無法驗證它是否連線到正確的伺服器時，就會發生這個錯誤。最常見的原因是 Greengrass 裝置的 IP 位址未列在憑證的 SAN 欄位中。 依照上一個解決方案中的指示取得 MQTT 代理程式憑證，並確認 SAN 欄位包含 AWS IoT Greengrass 裝置的 IP 位址，如其他資訊一節所述。如果沒有，請確認 IP 偵測器元件已正確安裝，然後重新啟動核心裝置。

問題	解決方案
僅在從內嵌式用戶端裝置連線時無法驗證伺服器名稱	Mmbed TLS 是內嵌裝置中常用的 TLS 程式庫，目前僅在憑證的 SAN 欄位中支援 DNS 名稱驗證，如 Mmbed TLS 程式庫程式碼所示。由於核心裝置沒有自己的網域名稱，而且取決於 IP 位址，因此使用 Mmbed TLS 的 TLS 用戶端在 TLS 交涉期間會失敗伺服器名稱驗證，進而導致連線失敗。我們建議您在 x50 9_crt_check_san 功能中，將 SAN IP 位址驗證新增至您的 MBed TLS 程式庫中。

相關資源

- AWS 物聯網管理說明文件
- AWS IoT Core 文件
- MQTT 代理程式元件
- MQTT 橋接器元件
- 用戶端裝置驗證元件
- IP 偵測器元件
- AWS IoT 裝置開發套件
- 使用 AWS IoT 網站實作本機用戶端裝置 (AWS 部落格文章)
- RFC 5280 — 網際網路 X.509 公開金鑰基礎結構憑證和憑證撤銷清單 (CRL) 設定檔

其他資訊

本節提供有關用戶端裝置與核心裝置之間通訊的其他資訊。

MQTT 代理程式會接聽核心裝置中的連接埠 8883，以進行 TLS 用戶端連線嘗試。下圖顯示 MQTT 代理程式的伺服器憑證範例。

範例憑證會顯示下列詳細資料：

- 此憑證是由 AWS IoT Greengrass 核心 CA 核心 CA 所發行，該 CA 是本機且特定於核心裝置的；也就是說，它充當本機 CA。
- 此證書每週由客戶端 auth 組件自動輪換，如下圖所示。您可以在客戶端身份驗證組件配置中設置此間隔。

- 主體別名 (SAN) 在 TLS 用戶端的伺服器名稱驗證中扮演重要角色。它可協助 TLS 用戶端確保其連線到正確的伺服器，並協助避免 TLS 工作階段設定期間的 man-in-the-middle 攻擊。在示例證書中，SAN 字段表示該服務器正在本地主機上監聽（本地 Unix 域套接字），並且網絡接口具有 IP 地址 192.168.1.12。

TLS 用戶端會使用憑證中的 SAN 欄位來驗證在伺服器驗證期間是否連線到合法伺服器。相反地，在 HTTP 伺服器與瀏覽器之間進行一般 TLS 交換期間，在伺服器驗證程序期間，會使用一般名稱 (CN) 欄位或 SAN 欄位中的網域名稱來交叉檢查瀏覽器實際連線的網域。如果核心裝置沒有網域名稱，SAN 欄位中包含的 IP 位址就會有相同的用途。如需詳細資訊，請參閱 RFC 5280 — 網際網路 X.509 公開金鑰基礎結構憑證和憑證撤銷清單 (CRL) 設定檔的主體替代名稱一節。

AWS IoT Greengrass 中的第個 IP 偵測器元件可確保憑證的 SAN 欄位中包含正確的 IP 位址。

範例中的憑證是由做為本機 CA 的 AWS IoT Greengrass 裝置簽署。TLS 用戶端 (MQTT 用戶端) 不知道此 CA，因此我們必須提供如下所示的 CA 憑證。

更多模式

- 使用 AWS IoT 資料以符合成本效益的方式，將物聯網資料直接導入 Amazon S3

機器學習與人工智慧

主題

彙總 Amazon DynamoDB 中的資料，用於 Athena 的機器學習預測

由薩欽多志（AWS）和彼得·莫納（AWS）創建

程式碼儲存庫：搭配亞馬遜雅典娜 ML 使用機器學習預測，透過 Amazon DynamoDB 資料	環境：生產	技術：機器學習與人工智慧、資料庫、無伺服器
工作負載：開源	AWS 服務：Amazon Athena；Amazon DynamoDB；AWS Lambda；Amazon；Amazon SageMaker QuickSight	

Summary

此模式說明如何使用 Amazon Athena 在 Amazon DynamoDB 表格中建立複雜的物聯網 (IoT) 資料彙總。您也會學到如何使用 Amazon 透過機器學習 (ML) 推論來豐富資料，以 SageMaker 及如何使用 Athena 查詢地理空間資料。您可以使用此模式作為建立符合組織需求之 ML 預測解決方案的基礎。

為了演示目的，此模式使用了經營踏板車共享的企業的示例場景，並希望預測必須為不同城市社區的客戶部署的最佳踏板車數量。該企業使用預先訓練的 ML 模型，根據過去四個小時預測客戶下一小時的需求。該方案使用路易斯維爾地鐵政府公民創新與技術辦公室提供的公共數據集。此案例的資源可在 GitHub 存放庫中使用。

先決條件和限制

- 有效的 AWS 帳戶
- 使用下列 AWS 身分和存取管理 (IAM) 角色建立 AWS CloudFormation 堆疊的許可：
  - Amazon Simple Storage Service (Amazon S3) 儲存貯體
  - Athena
  - DynamoDB
  - SageMaker
  - AWS Lambda

架構

技術, 堆

- Amazon QuickSight
- Amazon S3
- Athena
- DynamoDB
- Lambda
- SageMaker

目標架構

下圖顯示了使用 Athena 的查詢功能、Lambda 函數、Amazon S3 儲存、 SageMaker 端點和儀表板，在 DynamoDB 中建立複雜資料彙總的架構。 QuickSight

該圖顯示以下工作流程：

1. DynamoDB 資料表會擷取從一群踏板車傳輸的 IoT 資料。
2. Lambda 函數會以擷取的資料載入 DynamoDB 資料表。
3. Athena 查詢會為代表城市社區的地理空間資料建立新的 DynamoDB 表格。
4. 查詢位置會儲存在 S3 儲存貯體中。
5. Athena 函數會從裝載預先訓練的 ML 模型的 SageMaker 端點查詢 ML 推論。
6. Athena 會直接從 DynamoDB 資料表查詢資料，並彙總資料以進行分析。
7. 使用者在 QuickSight 儀表板中檢視分析資料的輸出。

工具

AWS 工具

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在 Amazon S3 中分析資料。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon SageMaker 是一種受管的 ML 服務，可協助您建置和訓練機器學習模型，然後將其部署到生產就緒的託管環境中。

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon QuickSight 是雲端規模商業智慧 (BI) 服務，可協助您在單一儀表板中視覺化、分析和報告資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

Code

此模式的程式碼可在 GitHub 使用 Amazon DynamoDB 資料的 ML 預測與亞馬 Amazon Athena ML 儲存庫中取得。您可以使用存放庫中的範 CloudFormation 本來建立範例案例中使用的下列資源：

- DynamoDB 資料表
- 一個 Lambda 函數，用於加載具有相關數據的表
- 推論請求的 SageMaker 端點，其中包含存放在 Amazon S3 中的預先訓練 XGBoost 模型
- 名為的 Athena 工作組 `V2EngineWorkGroup`
- 命名 Athena 查詢查詢地理空間形狀文件並預測踏板車需求
- 預先建置的 Amazon Athena DynamoDB 連接器，可讓 Athena 與 DynamoDB 通訊，並使用 AWS 無伺服器應用程式模型 (AWS SAM) 建立參照 DynamoDB 連接器的應用程式

史诗

取得範例資料集

任務	描述	所需技能
下載資料集和資源。	1. 下載無碼頭車輛租賃的公共數據集。基於示範目的，此資料會作為使用案例的一部分預先填入 DynamoDB 中，但在生產環境中，您可以透過各種機制 (例如 IoT 裝置或 Amazon Kinesis 取用者) 將此資料傳送到 DynamoDB。這些機制使	應用程式開發人員、資料

任務	描述	所需技能
	用 Lambda 將資料插入 DynamoDB。 2. 下載代表肯塔基州路易斯維爾市內歷史和文化社區邊界的 GIS shapefile。公共數據集由肯塔基州路易斯維爾和傑佛遜縣提供。原始的形狀文件已經轉換為文本文件，您可以使用 Athena 進行查詢，但是您可以在 Amazon Athena 的 GIS 形狀文件的地理空間處理中找到用於在 Jupyter 筆記本中轉換 shape 文件的 Python 代碼。 GitHub 3. 下載預先訓練的 Python 程式碼，透過使用 SageMaker 和 Athena 來訓練 ML 模型以進行每小時預測。 4. 在 Athena 取得 SQL 查詢，將 DynamoDB 中儲存的資料所有內容整合在一起，以進行即時預測。 5. (可選) 用 QuickSight 於在肯塔基州路易斯維爾地圖上視覺化地理空間資料。	

使用 CloudFormation 範本部署必要的資源

任務	描述	所需技能
建立 CloudFormation 堆疊。	1. 從 GitHub 存放庫下載 CloudFormation 範本。	AWS DevOps

任務	描述	所需技能
	2. 登入 AWS 管理主控台，然後選擇us-east-1 。注意：ML 模型存放在 us-east-1 AWS 區域的亞馬遜彈性容器登錄 (Amazon ECR) 中，但模式不受區域限制。您可以在支援此模式中使用的 AWS 服務的任何區域複製該模式。 3. 開啟主 CloudFormation 控台，然後在功能窗格上選擇 [堆疊]。 4. 選擇 [建立堆疊]，然後選擇 [使用現有資源 (匯入資源)]。 5. 在 [識別資源] 頁面上，選擇 [下一步]。 6. 在「指定範本」區段中，選取「上載範本檔案」做為「範本來源」。 7. 選擇 [檔案]，然後選擇您先前下載的 CloudFormation 範本。 8. 選擇 [下一步]，接受預設參數值，然後選擇 [下一步] 以逐步完成其餘的設定精靈。 9. 選取 [我確認 AWS CloudFormation 可能會使用自訂名稱建立 IAM 資源] 核取方塊。 10選擇建立堆疊。	

任務	描述	所需技能
	注意： CloudFormation 堆疊可能需要 15-20 分鐘才能建立這些資源。	
驗證部 CloudFormation 署。	若要確認範本中的範例資料是否已載入 DynamoDB，請執行下列動作： CloudFormation 1. 開啟 DynamoDB 主控台，然後從導覽窗格中選擇 [表格]。 2. 在「表格」區段中，檢查`DynamoDBTableDocklessVehicles` 表格。 3. 資源建立完成後，開啟 Athena 主控台，然後從導覽窗格中選擇「工作群組」。 4. 選擇`V2EngineWorkGroup` 工作群組，然後選擇 [切換工作群組]。 5. 如果您收到儲存查詢結果位置的提示，請選擇具有寫入許可的 Amazon S3 位置。 6. 選擇儲存。 7. 在瀏覽窗格中，選擇 [查詢編輯器]，然後選取`athena-ml-db-<your-AWS-account-number>` 資料庫。	應用程式開發人員

將地理位置文件加載到 Athena

任務	描述	所需技能
使用地理空間資料建立 Athena 表格。	若要將地理位置檔案載入 Athena，請執行下列動作： 1. 開啟 A thena 主控台，然後從導覽窗格中選擇 [查詢編輯器]。 2. 選擇「已儲存的查詢」標籤。 3. 搜尋並選取 Q1: 志趣相投。 4. 若要返回查詢編輯器，請選擇「編輯器」頁籤。 5. 選擇執行。這將`louisville_ky_neighborhoods` 在您的數據庫中創建一個名為的表。請確定資料表是在`athena-ml-db-<your-AWS-account-number>` 資料庫中建立的。 該查詢為表示城市社區的地理空間數據創建一個新表格。數據表是從 GIS 形狀文件創建的。`CREATE EXTERNAL TABLE`陳述式會定義資料表的結構定義，以及基礎資料檔案的位置和格式。 有關處理形狀文件並生成此表的 Python 代碼，請參閱 AWS 示例中的 Amazon Athena GIS 形狀文件的地理空間處理。如需詳細的 SQL 程式碼，請參	數據工程師

任務	描述	所需技能
	閱中的 create_neighborhood_table.sql GitHub。	

從彙總的 DynamoDB 資料中，依鄰近地區預測機車的需求

任務	描述	所需技能
在 Athena 宣告要查詢的函數 SageMaker。	1. 開啟 Athena 主控台，從功能窗格中選擇 [查詢編輯器]，然後選擇 [編輯器] 索引標籤。 2. 將下列 SQL 陳述式複製並貼到查詢編輯器中： `USING EXTERNAL FUNCTION predict_demand ( location_id BIGINT, hr BIGINT , dow BIGINT, n_pickup_1 BIGINT, n_pickup_2 BIGINT, n_pickup_3 BIGINT, n_pickup_4 BIGINT, n_dropoff_1 BIGINT, n_dropoff_2 BIGINT, n_dropoff_3 BIGINT, n_dropoff_4 BIGINT ) RETURNS DOUBLE SAGEMAKER '<Your SageMaker endpoint>'` SQL 陳述式的第一部分宣告外部函數，以便從裝載預先訓練模型的 SageMaker 端點查詢 ML 推論。	資料科學家、資料工程師

任務	描述	所需技能
	然後，執行下列動作： 1. 定義輸入參數的順序和類型以及傳回值的類型。 2. 選擇執行。	

任務	描述	所需技能
從彙總的 DynamoDB 資料中，依鄰近地區預測機車的需求。	現在，您可以使用 Athena 直接從 DynamoDB 查詢交易資料，然後彙總資料以進行分析和預測。透過直接查詢 DynamoDB 資料庫的 NoSQL 資料庫並不容易達到這個目標。 1. 開啟 A thena 主控台，然後從導覽窗格中選擇 [查詢] 編輯器。 2. 選擇「已儲存的查詢」標籤。 3. 搜尋並選取第 2 季：動態 ScooterPredict浴室。 4. 若要返回查詢編輯器，請選擇「編輯器」頁籤。 5. 選擇執行。 SQL 陳述式會執行下列動作： • 使用 A thena 聯合查詢來查詢含有原始行程資料的 DynamoDB 表 • 使用 Athena 的地理空間功能將地理坐標放置到社區 • 透過使用 ML 推論來豐富資料 SageMaker 如需有關在 Athena 使用 SQL 彙總 DynamoDB 資料和 SageMaker 推論資料的詳細資	應用程式開發人員、資料

任務	描述	所需技能
	訊，請參閱. GitHub athena_long.sql	
驗證輸出。	輸出表格包括鄰近區域質心的鄰域、經度和緯度。它還包括預計下一個小時的車輛數量。 查詢會產生所選時間點的預測。您可以透過變更陳述式中任何位置的運算式`TIMESTAMP '2019-09-07 15:00'` ，進行其他任何時間的預測。 如果 DynamoDB 表格中有即時資料饋送，請將時間戳記變更為。`NOW()`	應用程式開發人員、資料

清理環境

任務	描述	所需技能
刪除資源。	1. 開啟 A thena 主控台並清空您建立為 CloudFormation 堆疊一部分的值區。 2. 開啟主CloudFormation 控台，然後刪除名為的堆疊`bdb-1462-athena-dynamodb-ml-stack` 。 3. 開啟 Amazon CloudWatch 主控台，然後刪除名為的日誌群組`/aws/sagemaker/Endpoints/Sg-athena-ml-dynamodb-model-endpoint` 。	AWS 應用程式開發人員 DevOps

相關資源

- Amazon Athena 查詢聯盟 SDK (GitHub)
- 查詢地理空間資料 (Amazon Athena 用戶指南
- 透過亞馬遜雅典娜 ML (AWS 大數據部落格) 使用針對亞馬遜 DynamoDB 資料進行 ML 預測
- Amazon ElastiCache 適用於 Redis 的（AWS 文檔）
- Amazon Neptune（AWS 文檔）

將一個 AWS 帳戶中的 AWS CodeCommit 儲存庫與另一個帳戶中的 SageMaker 工作室建立關聯

由勞倫斯·范德馬斯（AWS）和奧布里·奧斯楚伊森（AWS）創建

環境：生產	技術：機器學習與人工智慧 DevOps；安全性、身分識別、合規性；雲端原生	AWS 服務：AWS CodeCommit；Amazon SageMaker；AWS Identity and Access Management

Summary

此模式提供有關如何將一個 AWS 帳戶 (帳戶 A) 中的 AWS CodeCommit 儲存庫與另一個 AWS 帳戶 (帳戶 B) 中的 Amazon SageMaker 工作室建立關聯的指示和程式碼。若要設定關聯，您必須在帳戶 A 中建立 AWS Identity and Access Management (IAM) 政策和角色，並在帳戶 B 中建立 IAM 內嵌政策，然後使用殼層指令碼將存放 CodeCommit 庫從帳戶 A 複製到帳戶 B 中的 SageMaker Studio。

先決條件和限制

前提

- 兩個 AWS 帳戶，一個包含 CodeCommit 存放庫，另一個包含具有使用者的 SageMaker 網域
- 透過虛擬私有網路 (VPC) 端點存取或存取 AWS Security Token Service (AWS STS) 的佈建網SageMaker 域 CodeCommit 和使用者
- 對 IAM 的基本了解
- 對SageMaker 工作室的基本了解
- 對 Git 的一個基本的了解和 CodeCommit

限制

此模式僅適用於 SageMaker 工作室，而不適用於 Amazon SageMaker 上的 RStudio。

架構

技術, 堆棧

- Amazon SageMaker
- Amazon SageMaker 一室
- AWS CodeCommit
- AWS Identity and Access Management (IAM)
- Git

目標架構

下圖顯示將存放 CodeCommit 庫從帳戶 A 與帳戶 B 中 SageMaker Studio 產生關聯的架構。

該圖顯示以下工作流程：

1. 使用者在帳戶 B 中使用 SageMaker Studio 中的 SageMaker 執行`sts:AssumeRole`角色時，透過該角色在帳戶 A 中擔任角色，假定的角色包括複製指定存放庫並與之互動的 CodeCommit 權限。`MyCrossAccountRepositoryContributorRole`
2. 用戶從 SageMaker Studio 中的系統終端執行 Git 命令。

自動化和規模

此模式包含可使用 AWS Cloud Development Kit (AWS CDK)、AWS CloudFormation 或地形自動化的手動步驟所組成。

工具

AWS 工具

- Amazon SageMaker 是一種受管機器學習 (ML) 服務，可協助您建立和訓練機器學習模型，然後將其部署到生產就緒的託管環境中。
- Amazon SageMaker Studio 是適用於機器學習的 Web 型整合式開發環境 (IDE)，可讓您建置、訓練、偵錯、部署和監控機器學習模型。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

其他工具

- Git 是一個分佈式版本控制系統，用於在軟件開發過程中跟踪源代碼的變化。

史诗

在帳戶 A 中建立 IAM 政策和 IAM 角色

任務	描述	所需技能
在帳戶 A 中建立存放庫存取的 IAM 政策。	1. 登入 AWS 管理主控台，並開啟 IAM 主控台。 2. 在導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。 3. 選擇 JSON 標籤。 4. 從此模式的其他資訊區段中的 IAM 政策範例複製政策陳述式，然後將陳述式貼到 JSON 編輯器中。請務必取代策略中的所有預留位置值。 5. 選擇「下一步:標籤」，然後選擇「下一步:檢閱」。 6. 針對 Name (名稱)，輸入政策的名稱。注意：在此模式中，IAM 政策稱為`CrossAccountAccessForMySharedDemoRepo`，但您可以選擇您喜歡的任何策略名稱。 7. 選擇建立政策。	AWS DevOps

任務	描述	所需技能
	提示：最佳做法是將 IAM 政策的範圍限制為使用案例所需的最低許可。	
在帳戶 A 中建立存放庫存取權的 IAM 角色。	1. 在 IAM 主控台的導覽窗格中，選擇 [角色]，然後選擇 [建立角色]。 2. 對於受信任的實體類型，選取 AWS 帳戶。 3. 在 AWS 帳戶區段中，選取「其他 AWS 帳戶」。 4. 在帳戶 ID 中，輸入帳戶 B 的帳號 ID。 5. 在 [新增權限] 頁面上，搜尋並選擇您先前建立的`CrossAccountAccessForMySharedDemoRepo` 原則。 6. 選擇下一步。 7. 在 Role name (角色名稱) 中，輸入名稱。附註：在此模式中，IAM 角色名稱稱稱為`MyCrossAccountRepositoryContributorRole` ，但您可以選擇喜歡的任何角色名稱。 8. 選擇 [建立角色]，然後複製新角色的 Amazon 資源名稱 (ARN)。	AWS DevOps

在帳戶 B 中建立 IAM 內嵌政策

任務	描述	所需技能
將內嵌原則附加至帳戶 B 中附加至 SageMaker 網域使用者的執行角色。	1. 在 IAM 主控台的導覽窗格中，選擇 [角色]。 2. 在帳戶 B 中搜尋並選擇附加至 SageMaker 網域使用者的執行角色。 3. 選擇 [新增權限]，然後選擇 [建立內嵌原則]。 4. 選擇 JSON 標籤。 5. 複製下列原則陳述式，然後將其貼到 JSON 編輯器中。 `{ "Version": "2012-10-17", "Statement": [ { "Sid": "VisualEditor0", "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>" } ] }` 6. <Account_A_ID> 以帳戶 A 的帳戶 ID 取代<Account_A_Role_Na	AWS DevOps

任務	描述	所需技能
	me> 為您先前建立的 IAM 角色名稱。 7. 選擇檢閱政策。 8. 在名稱中，輸入內嵌政策的名稱。 9. 選擇建立政策。	

為帳戶 B 複製 SageMaker Studio 中的儲存庫

任務	描述	所需技能
在帳戶 B 的 SageMaker Studio 中建立殼層指令碼。	1. 在SageMaker 主控台的導覽窗格中，選擇 [Studio]。 2. 選取您的使用者設定檔，然後選擇開啟工作室。 3. 在「首頁」區段中，選擇「開啟啟動器」。 4. 在「公用程式和檔案」區段中，選擇「文字檔案」。 5. 從此模式的其他資訊區段中的範例 SageMaker shell 指令碼複製指令碼，然後將陳述式貼到新檔案中。請務必取代指令碼中的所有預留位置值。 6. 以滑鼠右鍵按一下新檔案的 untitled.txt 索引標籤，然後選擇 [重新命名文字]。在「新名稱」中輸入 cross_account_git_clone.sh，然後選擇「重新命名」。	AWS DevOps

任務	描述	所需技能
從系統終端調用 shell 腳本。	1. 在SageMaker 主機的 [首頁] 區段中，選擇 [開啟啟動器]。 2. 在公用程式和檔案區段中，選擇系統終端機。 3. 在終端機中，執行下列命令： `chmod u+x ./cross_account_git_clone.sh && ./cross_account_git_clone.sh` 您已在 SageMaker Studio 跨帳戶中克隆 CodeCommit 存儲庫。您現在可以從系統終端執行所有 Git 命令。	AWS DevOps

其他資訊

IAM 政策範例

如果您使用此範例原則，請執行下列動作：

- <CodeCommit_Repository_Region>以存放庫的 AWS 區域取代。
- <Account_A_ID>以帳戶 A 的帳號 ID 取代。
- <CodeCommit_Repository_Name>以帳戶 A 中的 CodeCommit 儲存庫名稱取代。

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
```

```
            "codecommit:BatchGet*",
            "codecommit:Create*",
            "codecommit:DeleteBranch",
            "codecommit:Get*",
            "codecommit:List*",
            "codecommit:Describe*",
            "codecommit:Put*",
            "codecommit:Post*",
            "codecommit:Merge*",
            "codecommit:Test*",
            "codecommit:Update*",
            "codecommit:GitPull",
            "codecommit:GitPush"
        ],
        "Resource": [

 "arn:aws:codecommit:<CodeCommit_Repository_Region>:<Account_A_ID>:<CodeCommit_Repository_Name>
        ]
    }
]
}
```

SageMaker 殼層指令碼範例

如果您使用此範例指令碼，請執行下列動作：

- <Account_A_ID>以帳戶 A 的帳號 ID 取代。
- 以您先前建立的 IAM 角色名稱取<Account_A_Role_Name>代。
- <CodeCommit_Repository_Region>以存放庫的 AWS 區域取代。
- <CodeCommit_Repository_Name>以帳戶 A 中的 CodeCommit 儲存庫名稱取代。

```
#!/usr/bin/env bash
#Launch from system terminal
pip install --quiet git-remote-codecommit

mkdir -p ~/.aws
touch ~/.aws/config

echo "[profile CrossAccountAccessProfile]
region = <CodeCommit_Repository_Region>
credential_source=EcsContainer
```

```
role_arn = arn:aws:iam::<Account_A_ID>:role/<Account_A_Role_Name>
output = json" > ~/.aws/config

echo '[credential "https://git-
codecommit.<CodeCommit_Repository_Region>.amazonaws.com"]
        helper = !aws codecommit credential-helper $@ --profile
 CrossAccountAccessProfile
        UseHttpPath = true' > ~/.gitconfig

git clone codecommit::<CodeCommit_Repository_Region>://
CrossAccountAccessProfile@<CodeCommit_Repository_Name>
```

自動執行 Amazon Lookout for Vision 訓練和部署，以進行異常偵測

由邁克爾·沃爾納（AWS），加布里埃爾·羅德里格斯·加西亞（AWS），王康康（AWS），舒克拉特霍傑夫（AWS），桑傑·阿肖克（AWS），亞辛·扎弗里（AWS）和加布里埃爾·齊爾卡（AWS）創建

代碼庫：automated-silicon-wafer-anomaly--換視detection-using-amazon-lookout覺	環境：生產	技術：機器學習和人工智能; 雲原生; DevOps
AWS 服務：AWS CloudFormation；AWS；AWS CodeBuild；AWS CodeCommit；AWS CodePipeline；AWS Lambda；Amazon Lookout for Vision		

Summary

此模式可協助您將 Amazon Lookout for Vision 機器學習模型的訓練和部署自動化，以進行視覺檢查。雖然此模式專注於矽晶圓的異常偵測，但是您可以調整解決方案，以應用於各種產品和產業。

2020 年，全球最大的半導體製造商之一的年產能突破 1200 萬片等效 12 英寸晶圓。為了確保這些晶圓的品質和可靠性，目視檢測是生產過程中不可或缺的一個步驟。傳統的目視檢查方法，例如人工採樣或使用依賴統計措施的過時舊式工具，可能會耗時且效率低下。鑑於此過程的規模及其對於更廣泛的半導體產業的重要性，使用先進的人工智慧 (AI) 技術將視覺檢測最佳化和自動化，有很大的機會。

Lookout for Vision 有助於簡化圖像和物體檢測過程，減少了對昂貴且不一致的人工檢測的需求。該解決方案改善了質量控制，促進了準確的缺陷和損壞評估，並確保符合行業標準。此外，您無需專門的機器學習專業知識，就可以自動執行 Lookout for Vision 檢測流程。

使用此解決方案，您可以將計算機視覺模型集成到任何系統中。例如，您可以將模型集成到一個網站中，用戶可以在其中上傳圖像並對其進行分析是否存在缺陷。下圖顯示了化學機械拋光 (CMP) 製程中具有刮傷缺陷的矽晶圓範例。您可以使用「Lookout for Vision」來偵測這些異常。例如，「檢 Lookout for Vision」以 99.04% 的信心偵測到此影像中的異常。

此解決方案是根據使用 Amazon Lookout for Vision 部落格文章建立以事件為基礎的追蹤解決方案中所述的程式碼和使用案例為基礎。此解決方案會修改原始程式碼，以啟用 CI/CD 管道自動化，並整合開放原始碼 Amazon Lookout for Vision Python SDK () GitHub。如需有關 Python 開發套件的詳細資訊，請參閱使用 Python SDK 部落格文章建置、訓練和部署適用於視覺模型的 Amazon 瞭望工具。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS 帳戶中的管理許可
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝和設定的 AWS CDK
- Python 版本 3.10，已安裝

架構

目標架構

此架構說明透過 CI/CD 管道的 Amazon Lookout for Vision 模型的建置、訓練和部署的自動化作業。該圖顯示以下工作流程：

1. 該代碼存儲在 Amazon 存儲 CodeCommit 庫中。開發人員可以修改代碼，更改輸入圖像或向自動化管道添加其他步驟。
2. 在部署解決方案或更新 CodeCommit 存儲庫的主分支後，Amazon CodePipeline 會自動將代碼推送到 Amazon CodeBuild。
3. CodeBuild 使用 Lookout for Vision Python SDK 來訓練和部署映像分類模型。用於訓練的映像存放在亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體中。 CodeBuild 自動下載這些圖像並存儲它們。要根據您的需求自定義解決方案，您可以導入自己的圖像。
4. Lookout for Vision 模型會透過 AWS Lambda 向最終使用者公開。但是，您不僅限於這種方法。您也可以在 IoT 裝置的邊緣部署 Lookout for Vision，也可以按排程以批次處理程序的形式執行，以產生預測。

工具

AWS 服務

- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Lookout for Vision 使用電腦視覺，準確且大規模地在工業產品中尋找視覺偵測。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

代碼存儲庫

此模式的程式碼可在 GitHub 自動化 Amazon Lookout for Vision 訓練和矽晶圓異常偵測儲存庫的部署中找到。

最佳實務

當執行程式碼做為實驗時，請務必停止您的 Amazon Lookout for Vision 端點。

史诗

部署解決方案

任務	描述	所需技能
克隆存 GitHub 儲庫。	將針對矽晶圓異常偵測儲存庫的 GitHub 自動化 Amazon Lookout 視覺訓練和部署複製到您的本機工作站。	Bash

任務	描述	所需技能
	`git clone https://github.com/aws-samples/automated-silicon-wafer-anomaly-detection-using-amazon-lookout-for-vision.git`	
建立虛擬環境。	輸入以下指令，在本機工作站上建立虛擬環境。 `python3 -m venv .venv`	Python
安裝依存項目。	建立虛擬環境之後，輸入下列指令以安裝所需的相依性。 `pip install -r requirements.txt`	Python
(僅限 Linux 使用者) 啟動虛擬環境。	初始化完成並建立虛擬環境之後，請使用下列指令來啟動虛擬環境。 `source .venv/bin/activate`	Bash
(僅限 Windows 使用者) 啟動虛擬環境。	初始化完成並建立虛擬環境之後，請使用下列指令來啟動虛擬環境。 `.venv\Scripts\activate.bat`	PowerShell

任務	描述	所需技能
部署堆疊。	1. 在 AWS CDK CLI 中，輸入以下命令以合成 AWS CloudFormation 範本。 `cdk synth` 2. 輸入以下命令以部署 CloudFormation 堆疊。 `cdk deploy --all --require-approval never` `--all` flag可確保一次安裝所有元件。 `--require-approval` 永遠不需要核准每個元件部署。	AWS 管理員

測試解決方案

任務	描述	所需技能
輸入範例測試事件。	1. 開啟 Lambda 主控台中的函數頁面。 2. 選擇`amazon-lookout-for-vision-project-lambda` 功能。 3. 選擇測試標籤。 4. 在 [測試事件] 下，選擇 [建立新事件]。 5. 輸入以下內容。 6. 選擇 測試。 `{`	一般 AWS

任務	描述	所需技能
	<pre> "tbd": "tbd" }</pre> 7. 若要檢閱測試結果，在 Execution result (執行結果) 下，展開 Details (詳細資訊)。	

相關資源

AWS 文件

- 開始使用 Amazon Lookout for Vision
- 開始使用 AWS CDK

AWS 部落格文章

- 使用 Python SDK 建置、訓練和部署適用於視覺模型的 Amazon 瞭望
- 使用 Amazon 視覺瞭望建置以事件為基礎的追蹤解決方案
- Amazon Lookout for Vision Python 開發套件：交叉驗證和與其他 AWS 服務整合

使用亞馬遜文本提取自動從 PDF 文件中提取內容

創建者：賈天下 (AWS)

環境：生產	技術：機器學習與人工智慧、分析、大數據	AWS 服務：Amazon S3; Amazon Textract Amazon SageMaker

Summary

許多組織需要從上傳到其業務應用程式的 PDF 檔案中擷取資訊。例如，組織可能需要準確地從稅務或醫療 PDF 文件中提取信息，以進行稅務分析或醫療索賠處理。

在 Amazon Web Services (AWS) 雲端上，Amazon Textract 會自動從 PDF 檔案擷取資訊 (例如，列印的文字、表單和表格)，並產生 JSON 格式的檔案，其中包含原始 PDF 檔案中的資訊。您可以在 AWS 管理主控台中使用 Amazon Textract，或透過實作 API 呼叫來使用。我們建議您使用程式化 API 呼叫來擴展和自動處理大量 PDF 檔案。

Amazon Textract 處理檔案時，會建立下列`Block`物件清單：頁面、行和文字單字、表單 (鍵值組)、表格和儲存格，以及選取元素。其他物件資訊也包括在內，例如邊界方框、置信區間、ID 和關係。Amazon Textract 提取的內容信息作為字符串。需要正確識別和轉換的資料值，因為下游應用程式可以更輕鬆地使用這些值。

此模式描述了使用 Amazon Textract 自動從 PDF 檔案擷取內容並將其處理為乾淨輸出的 step-by-step 工作流程。此樣式使用範本比對技術來正確識別必要欄位、索引鍵名稱和表格，然後將後處理校正套用至每個資料類型。您可以使用此模式來處理不同類型的 PDF 檔案，然後您可以縮放和自動化此工作流程，以處理具有相同格式的 PDF 檔案。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體，用於在 PDF 檔案轉換為 JPEG 格式以供 Amazon Textract 處理之後存放這些檔案。如需 S3 儲存貯體的詳細資訊，請參閱 Amazon S3 文件中的儲存貯體概觀。

- `Textract_PostProcessing.ipynb`Jupyter 筆記本（隨附），已安裝和配置。如需 Jupyter 筆記本的詳細資訊，請參閱 Amazon 文件中的建立 Jupyter 筆記本。 SageMaker
- 具有相同格式的現有 PDF 文件。
- 對 Python 的理解。

限制

- 您的 PDF 文件必須具有良好的質量並且清晰可讀。建議使用原生 PDF 檔案，但如果所有個別文字都清楚，您可以使用轉換為 PDF 格式的掃描文件。如需詳細資訊，請參閱 AWS Machine Learning 部落格上的使用 Amazon Textract 進行預先處理的 PDF 文件：視覺效果偵測和移除。
- 對於多頁檔案，您可以使用非同步作業或將 PDF 檔案分割為單一頁面，然後使用同步作業。如需有關這兩個選項的詳細資訊，請參閱 Amazon Textract 文件中的偵測和分析多頁文件中的文字和偵測和分析單頁文件中的文字。

架構

此模式的工作流程首先在範例 PDF 檔案上執行 Amazon Textract (第一次執行)，然後在與第一個 PDF 具有相同格式 (重複執行) 的 PDF 檔案上執行。下圖顯示組合的「首次執行」和「重複執行」工作流程，該工作流程會自動並重複從 PDF 檔案中擷取相同格式的內容。

圖表顯示此模式的下列工作流程：

1. 將 PDF 文件轉換為 JPEG 格式並將其存儲在 S3 存儲桶中。
2. 調用 Amazon Textract 取 API 並解析 Amazon Textract 取響應 JSON 文件。
3. 通過為每個必填字段添加正確的`KeyName:DataType`配對來編輯 JSON 文件。為「重複」執行階段建立`TemplateJSON`檔案。
4. 定義每個資料類型的後處理校正函數 (例如，浮點數、整數和日期)。
5. 準備與您的第一個 PDF 文件具有相同格式的 PDF 文件。
6. 調用 Amazon Textract 取 API 並解析 Amazon Textract 取響應 JSON。
7. 將剖析的 JSON 檔案與檔案相符。`TemplateJSON`
8. 導入後處理更正。

最終的 JSON 輸出文件具有正確的`KeyName`和`Value`每個必填字段。

目標技術堆疊

- Amazon SageMaker
- Amazon S3
- Amazon Textract

自動化和規模

您可以使用 AWS Lambda 函數在將新的 PDF 檔案新增至 Amazon S3 時，啟動 Amazon Textract 函數來自動執行重複執行工作流程。然後，Amazon Textract 會執行處理指令碼，最終輸出可以儲存到儲存位置。如需這方面的詳細資訊，請參閱 Lambda 文件中的使用 Amazon S3 觸發器叫用 Lambda 函數。

工具

- Amazon SageMaker 是全受管的 ML 服務，可協助您快速輕鬆地建立和訓練機器學習模型，然後將它們直接部署到生產就緒的託管環境中。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Textract 可讓您輕鬆地將文件文字偵測和分析新增至您的應用程式。

史诗

第一次運行

任務	描述	所需技能
轉換 PDF 檔案。	將 PDF 檔案分割為單一頁面，然後將 PDF 檔案轉換為 JPEG 格式，以便進行 Amazon Textract 同步操作 (Syn API)，為您的首次執行做好準備。 附註：您也可以針對多頁 PDF 檔案使用 Amazon Textract 非同步作業 (Asyn API)。	資料科學家、開發人

任務	描述	所需技能
解析 Amazon Textract 塊響應 JSON。	開啟 `Textract_PostProcessing.ipynb` Jupyter 筆記本 (隨附)，然後使用下列程式碼呼叫 Amazon Textract API： `response = textract.analyze_document( Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename) } }, FeatureTypes=["TABLES", "FORMS"])` 通過使用下面的代碼解析響應 JSON 成一個表單和表： `parseformKV=form_kv_from_JSON(response) parseformTables=get_tables_fromJSON(response)`	資料科學家、開發人

任務	描述	所需技能
編輯範本 JSON 檔案。	編輯每個KeyName和對應的解析 JSONDataType（例如字符串，浮點數，整數或日期）和表頭（例如，ColumnNames 和RowNames）。 此範本適用於每個個別的 PDF 檔案類型，這意味著可以將範本重複使用於具有相同格式的 PDF 檔案。	資料科學家、開發人
定義後處理校正功能。	Amazon Textract 對TemplateJSON 檔案的回應中的值為字串。日期、浮點數、整數或貨幣沒有差異。這些值必須針對您的下游使用案例轉換為正確的資料類型。 使用以下代碼根據TemplateJSON 文件更正每種數據類型： `finalJSON=postprocessingCorrection(parsedJSON,templateJSON)`	資料科學家、開發人

重複執行

任務	描述	所需技能
準備好 PDF 檔案。	準備 PDF 檔案，方法是將這些檔案分割為單一頁面，然後將它們轉換為 JPEG 格式以進行 Amazon Textract 同步操作 (Syn API)。	資料科學家、開發人

任務	描述	所需技能
	附註：您也可以針對多頁 PDF 檔案使用 Amazon Textract 非同步作業 (Asyn API)。	
調用 Amazon Textract 取 API。	使用下列程式碼呼叫 Amazon Textract 取 API： `response = textract.analyze_document( Document={ 'S3Object': { 'Bucket': BUCKET, 'Name': '{}'.format(filename) } }, FeatureTypes=["TABLES", "FORMS"])`	資料科學家、開發人
解析 Amazon Textract 塊響應 JSON。	通過使用下面的代碼解析響應 JSON 成一個表單和表： `parseformKV=form_kv_from_JSON(response) parseformTables=get_tables_fromJSON(response)`	資料科學家、開發人

任務	描述	所需技能
加載模板 JSON 文件並將其與解析的 JSON 匹配。	使用TemplateJSON 檔案，透過下列指令擷取正確的鍵值配對和表格： `form_kv_corrected= form_kv_correction (parseformKV,templ ateJSON) form_table_correct ed=form_Table_corr ection(parseformTa bles, templateJSON) form_kv_table_correcte d_final={**form_kv _corrected , **form_ta ble_corrected}`	資料科學家、開發人
後處理校正。	DataType在TemplateJ SON 文件和後處理函數中使用，通過使用以下代碼更正數據： `finalJSON=postproc essingCorrection(f orm_kv_table_corre cted_final,templat eJSON)`	資料科學家、開發人

相關資源

- 使用 Amazon Textract 自動從文件擷取文字和結構化資料
- 使用 Amazon Textract 文本提取文本和結構化數據
- Amazon Textract 資源

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon SageMaker 和 Azure 建置 MLOP 工作流程 DevOps

創建者迪皮卡庫馬爾（AWS）和薩拉·范德穆斯代克（AWS）

環境：生產	技術：機器學習和人工智能 DevOps; 運營	工作量：Microsoft
AWS 服務：Amazon API Gateway；Amazon ECR；Amazon EventBridge；AWS Lambda；Amazon SageMaker		

Summary

機器學習作業 (MLOP) 是一組實務，可自動化並簡化機器學習 (ML) 工作流程和部署。MLOP 著重於自動化機器學習生命週期。它有助於確保不只是開發模型，而且系統和重複部署、監控和重新訓練模型。它為 ML 帶來了 DevOps 原則。MLOP 可加快機器學習模型的部署速度，隨著時間的推移提高準確性，並且更有力地確保它們能夠提供真正的商業價值。

在開始 MLOP 旅程之前，組 Organizations 通常會擁有現有的 DevOps 工具和資料儲存解決方案。這種模式展示了如何利用 Microsoft Azure 和 AWS 的優勢。它可以幫助您將 Azure DevOps 與 Amazon 集成 SageMaker 以創建 MLOP 工作流程。

該解決方案簡化了 Azure 和 AWS 之間的工作。您可以使用 Azure 進行開發，並使用 AWS 進行機器學習。它促進了從頭到尾製作機器學習模型的有效程序，包括 AWS 上的資料處理、培訓和部署。為了提高效率，您可以透過 Azure 管 DevOps 線管理這些處理序。

先決條件和限制

先決條件

- Azure 訂用帳戶 — 可存取 Azure 服務，例如 Azure DevOps，以設定持續整合和持續部署 (CI/CD) 管線。
- 有效 AWS 帳戶 — 使用此模式中使用之 AWS 服務的許可。
- 資料 — 存取歷史資料以訓練機器學習模型。

- 熟悉機器學習概念 — 了解 Python，Jupyter 筆記本和機器學習模型開發。
- 安全組態 — 在 Azure 和 AWS 之間正確設定角色、政策和權限，以確保資料傳輸和存取的安全性。

限制

- 本指引不提供有關安全跨雲端資料傳輸的指導。如需跨雲端資料傳輸的詳細資訊，請參閱適用於混合雲和多雲端的 AWS 解決方案。
- 多雲端解決方案可能會增加即時資料處理和模型推論的延遲時間。
- 本指南提供了一個多帳戶 MLOP 架構的範例。根據您的機器學習和 AWS 策略進行調整是必要的。

架構

目標架構

目標架構將 Azure DevOps 與 Amazon 整合 SageMaker，以建立跨雲端機器學習工作流程。它使用 Azure 進行 CI/CD 處理程序，以及 SageMaker 機器學習模型訓練和部署。其中概述透過模型建置和部署取得資料的程序 (從 Amazon S3、雪花和 Azure 資料湖等來源)。關鍵元件包括用於模型建置和部署的 CI/CD 管道、資料準備、基礎設施管理，以及 SageMaker 用於培訓、評估和部署機器學習模型的 Amazon。此架構旨在跨雲端平台提供高效率、自動化且可擴充的機器學習工作流程。

該架構由以下組件組成：

1. 資料科學家在開發帳戶中執行 ML 實驗，透過使用各種資料來源探索 ML 使用案例的不同方法。數據科學家執行單元測試和試驗。下面的模型評估，數據科學家推送和合併的代碼模型生成存儲庫，這是託管在 Azure DevOps。此儲存庫包含多步驟模型建置管線的程式碼。
2. 在 Azure 上 DevOps，提供持續整合 (CI) 的模型建置管線可在程式碼合併至主分支時自動或手動啟動。在「自動化」帳戶中，這會啟動資料預先處理、模型訓練和評估，以及根據準確性條件模型註冊的 SageMaker 管道。
3. 自動化帳戶是託管 ML 環境 (Amazon ECR)、模型 (Amazon S3)、模型中繼資料 (模型登錄)、功能 (SageMaker SageMaker 功能存放區)、自動化管道 (管道) 和 ML 日誌深入解析 (SageMaker 以 CloudWatch 及 OpenSearch 服務) 的 ML 平台的中央帳戶。此帳戶可重複使用 ML 資產，並強制執行最佳實務以加速 ML 使用案例的交付。
4. 最新的模型版本會新增至 SageMaker 模型登錄以供檢閱。它會追蹤模型版本和各自的成品 (歷程和中繼資料)。它也會管理模型的狀態 (核准、拒絕或擱置中)，並管理下游部署的版本。

5. 在模型登錄檔中訓練過的模型透過 Studio 介面或 API 呼叫核准之後，就可以將事件傳送至 Amazon EventBridge。 EventBridge 在 Azure 上啟動模型部署管線 DevOps。
6. 提供持續部署 (CD) 的「模型部署」管線會從「模型部署」儲存庫出庫來源。原始碼包含程式碼、模型部署的組態，以及品質基準測試的測試指令碼。「模型部署」管線可根據您的推論類型量身打造。
7. 品質控制檢查之後，「模型部署」管線會將模型部署到「預備」帳戶。 測試帳戶是生產帳戶的副本，用於整合測試和評估。對於批次轉換，Model Deploy 管線可以自動更新批次推論程序，以使用最新的核准模型版本。對於即時、無伺服器或非同步推論，它會設定或更新個別模型端點。
8. 在測試帳戶中成功測試之後，可以透過 Model Deploy 管道手動核准，將模型部署到生產帳戶。此管線會在「部署至生產」步驟中佈建生產端點，包括模型監控和資料回饋機制。
9. 模型生產後，請使用「模 SageMaker 型監視器」和「 SageMaker 澄清」等工具來識別偏差、偵測漂移，並持續監控模型的效能。

自動化和規模

使用基礎結構即程式碼 (IaC) 自動部署到多個帳戶和環境。藉由自動化設定 MLOps 工作流程的程序，可以分隔處理不同專案的 ML 團隊所使用的環境。AWS 透過將基礎設施視為程式碼， CloudFormation 協助您建立模型、佈建和管理 AWS 資源。

工具

AWS 服務

- Amazon SageMaker 是一種受管機器學習服務，可協助您建立和訓練機器學習模型，然後將其部署到生產就緒的託管環境中。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。在此模式中，Amazon S3 用於資料儲存，並與整合以 SageMaker 進行模型訓練和模型物件。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。在這種模式中，Lamdba 用於數據預處理和後處理任務。
- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。在這種模式中，它存儲 SageMaker 用作培訓和部署環境的 Docker 容器。

- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。在此模式中，可 EventBridge 協調啟動自動模型重新訓練或部署的事件驅動或基於時間的工作流程。
- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。 在此模式中，它用於為 Amazon SageMaker 端點建立面向外部的單一入口點。

其他工具

- Azure 可 DevOps協助您管理 CI/CD 管線，並促進程式碼建置、測試和部署。
- Azure 資料湖儲存體或雪花型可能是 ML 模型訓練資料的協力廠商來源。

最佳實務

在實作此多雲端 MLOP 工作流程的任何元件之前，請先完成下列活動：

- 定義並瞭解機器學習工作流程及支援工作流程所需的工具。不同的用例需要不同的工作流程和組件。例如，個人化使用案例中的功能重複使用和低延遲推論可能需要功能 feature store 放區，但其他使用案例可能不需要此功能存放區。要成功地自訂架構，需要瞭解資料科學團隊的目標工作流程、使用案例需求和偏好的協同合作方法。
- 為架構的每個元件建立明確的責任分離。將資料儲存分散到 Azure 資料湖儲存體、雪花式和 Amazon S3 之間，可能會增加複雜性和成本。如果可能，請選擇一致的儲存機制。同樣地，請避免使用 Azure 和 AWS DevOps 服務的組合，或結合使用 Azure 和 AWS 機器學習服務。
- 選擇一或多個現有模型和資料集，以執行 MLOP 工作流程的 end-to-end 測試。測試成品應反映資料科學團隊在平台進入生產環境時所開發的實際使用案例。

史诗

設計您的 MLOP 架構

任務	描述	所需技能
識別資料來源。	根據目前和 future 的使用案例、可用的資料來源以及資料類型 (例如機密資料)，記錄需要與 MLOP 平台整合的資料來源。資料可以存放在 Amazon	資料工程師、資料科學家、雲端架構師

任務	描述	所需技能
	S3、Azure 資料湖儲存體、雪花式儲存體或其他來源。建立計劃，將這些來源與您的平台整合，並確保存取正確資源的安全。	
選擇適用的服務。	根據資料科學團隊所需的工作流程、適用的資料來源和現有雲端架構，新增或移除服務來自訂架構。例如，資料工程師和資料科學家可以在 AWS Glue 或 Amazon EMR 中 SageMaker執行資料預處理和功能工程。這是不太可能需要所有這三種服務。	AWS 管理員、資料工程師、資料科學家、ML 工程師
分析安全性需求。	收集和記錄安全要求. 這包括確定： • 哪些團隊或工程師可以存取特定資料來源 • 是否允許團隊訪問其他團隊的代碼和模型 • 團隊成員對非開發帳戶應擁有哪些權限 (如果有的話) • 跨雲端資料傳輸需要採取哪些安全措施	AWS 管理員、雲端架構師

設定 AWS Organizations

任務	描述	所需技能
設定 AWS Organizations。	在根 AWS 帳戶上設定 AWS Organizations。這可協助您管理作為多帳戶 MLOP 策略	AWS 管理員

任務	描述	所需技能
	一部分所建立的後續帳戶。如需詳細資訊，請參閱 AWS Organizations 文件。	

設置開發環境和版本控制

任務	描述	所需技能
建立 AWS 開發帳戶。	建立 AWS 帳戶，讓資料工程師和資料科學家有權實驗和建立機器學習模型。如需指示，請參閱 AWS Organizations 文件中的在組織中建立成員帳戶。	AWS 管理員
建立 `Model Build` 儲存庫。	在 Azure 中建立 Git 存放庫，資料科學家可以在實驗階段完成後推送其模型建置和部署程式碼。如需指示，請參閱 Azure 說明 DevOps 文件中的設定 Git 存放庫。	DevOps 工程師，ML 工程師
建立 `Model Deploy` 儲存庫。	在 Azure 中建立儲存標準部署程式碼和範本的 Git 存放庫。它應該包含組織所使用的每個部署選項的程式碼，如設計階段所識別。例如，它應包括即時端點、非同步端點、無伺服器推論或批次轉換。如需指示，請參閱 Azure 說明 DevOps 文件中的設定 Git 存放庫。	DevOps 工程師，ML 工程師
建立 Amazon ECR 儲存庫。	設定 Amazon ECR 儲存庫，將核准的機器學習環境存放	ML 工程師

任務	描述	所需技能
	為 Docker 映像檔。允許資料科學家和機器學習工程師定義新的環境。如需指示，請參閱 Amazon ECR 文件中的建立私有存放庫。	
設置 SageMaker 工作室。	根據先前定義的安全性需求和偏好的資料科學工具，例如您選擇的整合 SageMaker 式開發環境 (IDE)，在開發帳戶上設定 Studio。使用生命週期組態自動化重要功能的安裝，並為資料科學家建立統一的開發環境。如需詳細資訊，請參閱 SageMaker 文件中的 Amazon SageMaker 工作室。	ML 工程師，數據科學家

整合 CI/CD 管線

任務	描述	所需技能
建立自動化帳戶。	建立執行自動化管道和任務的 AWS 帳戶。您可以授予資料科學團隊對此帳戶的讀取權限。如需指示，請參閱 AWS Organizations 文件中的在組織中建立成員帳戶。	AWS 管理員
設定模型登錄。	在自動化帳戶中設定 SageMaker 模型登錄。此登錄會儲存 ML 模型的中繼資料，並協助特定資料科學家或團隊主管核准或拒絕模型。如需詳細資訊，請參閱 SageMaker 文	ML 工程師

任務	描述	所需技能
	件中的使用模型登錄註冊和部署模型。	
建立Model Build管線。	在 Azure 中建立 CI/CD 管線，在程式碼推送至存放庫時手動或自動啟動Model Build。管道應該簽出原始程式碼，並在自動化帳戶中建立或更新 SageMaker 管道。管線應該將新模型新增至模型登錄。如需建立管線的詳細資訊，請參閱 Azure 管道文件。	DevOps 工程師，ML 工程師

建置部署堆疊

任務	描述	所需技能
建立 AWS 預備和部署帳戶。	建立 AWS 帳戶以進行 ML 模型的暫存和部署。這些帳戶應該是相同的，以便在移至生產環境之前準確測試階段中的模型。您可以授予資料科學團隊對預備帳戶的讀取權限。如需指示，請參閱 AWS Organizations 文件中的在組織中建立成員帳戶。	AWS 管理員
設定 S3 儲存貯體以進行模型監控。	如果您要為Model Deploy管線建立的已部署模型啟用模型監視，請完成此步驟。建立 Amazon S3 儲存貯體以存放輸入和輸出資料。如需建立 S3 儲存貯體的詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。設定跨帳戶權限，以便	ML 工程師

任務	描述	所需技能
	在自動化帳戶中執行自動化模型監視工作。如需詳細資訊，請參閱 SageMaker 文件中的監控資料和模型品質。	
建立Model Deploy管線。	在 Azure 中建立 CI/CD 管線，該管線會在模型登錄中核准模型時啟動。管道應該簽出原始程式碼和模型人工因素、建置基礎結構範本以在測試和生產帳戶中部署模型、在測試帳戶中部署模型、執行自動化測試、等待手動核准，以及將核准的模型部署到生產帳戶中。如需建立管線的詳細資訊，請參閱 Azure 管道文件。	DevOps 工程師，ML 工程師

(選擇性) 自動化 ML 環境基礎架構

任務	描述	所需技能
建置 AWS CDK 或 CloudFormation 範本。	為需要自動部署的所有環境定義 AWS Cloud Development Kit (AWS CDK) 或 AWS CloudFormation 範本。這可能包括開發環境、自動化環境，以及測試和部署環境。如需詳細資訊，請參閱 AWS CDK 和CloudFormation文件。	AWS DevOps
建立Infrastructure 管線。	在 Azure 中建立用於基礎結構部署的 CI/CD 管線。管理員可以啟動此管道以建立新的 AWS 帳戶，並設定 ML 團隊所需的環境。	DevOps 工程師

故障診斷

問題	解決方案
監控和漂移偵測不足 — 監控不當可能導致錯過模型效能問題或資料漂移的偵測。	使用 Amazon CloudWatch、 SageMaker 模型監視器和 SageMaker 澄清等工具強化監控架構。設定警示，針對已識別的問題立即採取行動
CI 管線觸發錯誤 — Azure 中的 CI 管線 DevOps 可能不會在程式碼合併時觸發，因為設定錯誤。	檢查 Azure DevOps 專案設定，以確保 Webhook 已正確設定並指向正確的 SageMaker 端點。
治理 — 中央自動化帳戶可能無法在 ML 平台上強制執行最佳實務，導致工作流程不一致。	稽核自動化帳戶設定，確保所有 ML 環境和模型都符合預先定義的最佳作法和原則。
模型登錄核准延遲 — 這會發生在檢查和核准模型的延遲時，可能是因為人們需要時間檢閱模型，或是因為技術問題。	實施通知系統，以通知利益相關者正在等待批准的模型，並簡化審核流程。
模型部署事件失敗 — 傳送至啟動模型部署管線的事件可能會失敗，進而造成部署延遲。	確認 Amazon EventBridge 具有正確的權限和事件模式，可成功叫用 Azure DevOps 管道。
生產部署瓶頸 — 手動核准程序可能會產生瓶頸，延遲模型的生產部署。	優化模型部署管道內的審批工作流程，促進及時審查和清晰的溝通渠道。

相關資源

AWS 文件

- Amazon SageMaker 文檔
- Machine Learning 鏡頭 (AWS 架構良好的架構)
- 規劃成功的 MLOP (AWS Prescriptive Guidance)

其他 AWS 資源

- 適用於 Amazon 企業的 MLOP 基礎藍圖 SageMaker (AWS 部落格文章)
- AWS 高峰會 2022 年澳新銀行-建築師的電nd-to-end 子商務部門 (YouTube 影片)

Azure 文件

- Azure DevOps 文件
- Azure 管道文件

為其建立自訂 Docker 容器映像， SageMaker 並將其用於 AWS Step Functions 中的模型培訓

由朱莉婭·布魯斯茲茲（AWS），妮哈·沙爾馬（AWS），奧布里·奧斯楚森（AWS），莫罕·戈爾達·普魯肖塔瑪（AWS）和馬特烏斯扎倫巴（AWS）創建

環境：生產	技術：機器學習和人工智能; DevOps	AWS 服務：Amazon ECR; Amazon SageMaker; AWS Step Functions

Summary

此模式示範如何為 Amazon 建立 Docker 容器映像， SageMaker並將其用於 AWS Step Functions 中的培訓模型。透過將自訂演算法封裝在容器中，您幾乎可以在 SageMaker 環境中執行任何程式碼，而不論程式設計語言、架構或相依性為何。

在提供的範例SageMaker 筆記本中，自訂 Docker 容器映像儲存在 Amazon Elastic Container Registry (Amazon ECR) 中。然後，Step Functions 會使用儲存在 Amazon ECR 中的容器來執行 Python 處理指令碼。 SageMaker然後，容器將模型導出到 Amazon Simple Storage Service (Amazon S3)。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- SageMaker具有 Amazon S3 許可的 AWS Identity and Access Management (IAM) 角色
- Step Functions 的 IAM 角色
- 熟悉 Python
- 熟悉 Amazon SageMaker Python 開發套件
- 熟悉 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 熟悉適用於 Python 的 AWS 開發套件 (博多 3)
- 熟悉 Amazon ECR
- 熟悉碼頭工人

產品版本

- AWS Step Functions 數資料科學開發套件 2.3.0 版
- Amazon SageMaker Python 開發套件版本 2.78.0

架構

下圖顯示建立 Docker 容器映像 SageMaker，然後在 Step Functions 中將其用於訓練模型的範例工作流程：

該圖顯示以下工作流程：

1. 資料科學家或 DevOps 工程師使用 Amazon SageMaker 筆記本建立自訂的 Docker 容器映像。
2. 資料科學家或 DevOps 工程師將 Docker 容器映像儲存在私有登錄中的 Amazon ECR 私有儲存庫中。
3. 資料科學家或工 DevOps 程師使用 Docker 容器，在 Step Functions 數工作流程中執行 Python SageMaker 處理工作。

自動化和規模

此樣式中的範例 SageMaker 筆記本使用`ml.m5.xlarge`筆記本實例類型。您可以變更執行個體類型以符合您的使用案例。如需 SageMaker 筆記本執行個體類型的詳細資訊，請參閱 Amazon SageMaker 定價。

工具

- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon SageMaker 是一種受管機器學習 (ML) 服務，可協助您建立和訓練機器學習模型，然後將其部署到生產就緒的託管環境中。
- Amazon SageMaker Python 開發套件是用於訓練和部署機器學習模型的開放原始碼程式庫。SageMaker
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。

- AWS Step Functions 數資料科學 Python 開發套件是一個開放原始碼程式庫，可協助您建立可處理和發佈機器學習模型的 Step Functions 數工作流程。

史诗

創建一個自定義碼頭容器映像並將其存儲在 Amazon ECR 中

任務	描述	所需技能
設置 Amazon ECR 並創建一個新的私有註冊表。	如果您尚未設定 Amazon ECR，請按照 Amazon ECR 使用者指南中的使用 Amazon ECR 設定中的說明進行設定。每個 AWS 帳戶都提供一個預設的私有 Amazon ECR 登錄。	DevOps 工程師
建立 Amazon ECR 私有儲存庫。	請遵循 Amazon ECR 使用者指南中建立私有存放庫中的指示進行。 注意：您創建的存儲庫是您將存儲自定義 Docker 容器映像的位置。	DevOps 工程師
建立 Docker 檔案，其中包含執行 SageMaker 處理工作所需的規格。	通過配置 Docker 文件創建一個 Docker 文件，其中包含運行 SageMaker 處理任務所需的規格。如需指示，請參閱 Amazon SageMaker 開發人員指南中的調整您自己的訓練容器。 如需 Docker 檔案的詳細資訊，請參閱 Docker 文件中的 Docker 檔案參考。 用於創建 Docker 文件的示例 Jupyter 筆記本代碼單元格	DevOps 工程師

任務	描述	所需技能
	儲存格 1 `# Make docker folder` `!mkdir -p docker` 儲存格 2 `%%writefile docker/Dockerfile` `FROM python:3.7-slim-buster` `RUN pip3 install pandas==0.25.3 scikit-learn==0.21.3` `ENV PYTHONUNBUFFERED=TRUE` `ENTRYPOINT ["python3"]`	

任務	描述	所需技能
建立您的碼頭容器映像檔，並將其推送至 Amazon ECR。	1. 使用您在 AWS CLI 中執行`docker build`命令建立的 Docerfile 來建立容器映像。 2. 透過執行`docker push`命令，將容器映像檔推送至 Amazon ECR。 如需詳細資訊，請參閱在建置您自己的演算法容器中的 < 建置和註冊容器 > GitHub。 用於構建和註冊 Docker 映像的示例 Jupyter 筆記本代碼單元格 重要事項：在執行下列儲存格之前，請確定您已建立 Docker 檔案並將其儲存在名為的目錄中。`docker`此外，請確定您已建立 Amazon ECR 儲存庫，並將第一個儲存格中的`ecr_repository`值取代為儲存庫的名稱。 儲存格 1 `import boto3` `tag = ':latest'` `account_id = boto3.cli` `ent('sts').get_cal` `ler_identity().get` `('Account')` `region = boto3.Ses` `sion().region_name` `ecr_repository = 'byoc'`	DevOps 工程師

任務	描述	所需技能
	`image_uri = '{}.dkr.ecr.{}.amazonaws.com/{}'.format(account_id, region, ecr_repository + tag)` 儲存格 2 `# Build docker image` `!docker build -t $image_uri docker` 儲存格 3 `# Authenticate to ECR` `!aws ecr get-login -password --region {region} | docker login --username AWS --password-stdin {account_id}.dkr.ecr.{region}.amazonaws.com` 儲存格 4 `# Push docker image` `!docker push $image_uri` 注意：[您必須向私人登錄驗證 Docker 用戶端](#)，才能使用`docker push`和命`docker pull`令。這些命令會在登錄中的儲存庫中推送和提取映像檔。	

建立使用自訂 Docker 容器映像檔的 Step Functions 工作流程

任務	描述	所需技能
建立包含自訂處理和模型訓練邏輯的 Python 指令碼。	撰寫要在資料處理指令碼中執行的自訂處理邏輯。然後，將其保存 Python 名為`training.py` 。 如需詳細資訊，請參閱開啟 SageMaker 指令碼模式時使用自己的模型 GitHub。 包含自訂處理和模型訓練邏輯的 Python 指令碼範例 `%%writefile training.py` `from numpy import empty` `import pandas as pd` `import os` `from sklearn import datasets, svm` `from joblib import dump, load` `if __name__ == '__main__':` `digits = datasets.load_digits()` `#create classifier object` `clf = svm.SVC(gamma=0.001, C=100.)` `#fit the model` `clf.fit(digits.data[:-1], digits.target[:-1])`	資料科學家

任務	描述	所需技能
	`#model output in binary format` `output_path = os.path.join('/opt/ml/processing/model', "model.joblib")` `dump(clf, output_path)`	

任務	描述	所需技能
建立「Step Functions」工作流程，將您的「 SageMaker 處理」工作列為其中一個步驟。	安裝和匯入 AWS Step Functions 數資料科學開發套件，並將 training.py 檔案上傳到 Amazon S3。然後，使用 Amazon SageMaker Python 開發套件在步驟函式中定義處理步驟。 重要：請確定您已在 AWS 帳戶中為 Step Functions 建立 IAM 執行角色。 範例環境設定和要上傳到 Amazon S3 的自訂訓練指令碼 `!pip install stepfunct ions` `import boto3` `import stepfunctions` `import sagemaker` `import datetime` `from stepfunctions import steps` `from stepfunct ions.inputs import ExecutionInput` `from stepfunct ions.steps import ( Chain )` `from stepfunct ions.workflow import Workflow` `from sagemaker .processing import ScriptProcessor,`	資料科學家

任務	描述	所需技能
	`` ProcessingInput, ProcessingOutput sagemaker_session = sagemaker.Session() bucket = sagemaker _session.default_b ucket() role = sagemaker .get_execution_role() prefix = 'byoc-tra ining-model' # See prerequisites section to create this role workflow_execution_rol e = f"arn:aws:iam:: {account_id}:role/Ama zonSageMaker-StepF unctionsWorkflowEx ecutionRole" execution_input = ExecutionInput(schema={ "Preproce ssingJobName": str}) input_code = sagemaker _session.upload_data("training.py", bucket=bucket, key_prefix="prepro cessing.py",) ``` 使用自訂 Amazon ECR 映像檔和 Python 指令碼的 SageMaker 處理步驟定義範例	

任務	描述	所需技能
	注意：請務必使用execution_input 參數來指定工作名稱。每次執行工作時，參數的值必須是唯一的。此外，training.py 文件的代碼作為input參數傳遞給ProcessingStep ，這意味著它將被複製到容器內。ProcessingInput 程式碼的目的地與中的第二個引數相同container_entrypoint 。 `script_processor = ScriptProcessor(command=['python3'], image_uri=image_uri, role=role, instance_count=1, instance_type='ml.m5.xlarge')` `processing_step = steps.ProcessingStep( "training-step", processor=script_processor, job_name=execution_input["PreprocessingJobName"], inputs=[ ProcessingInput( source=input_code,`	

任務	描述	所需技能
	<pre>destinati on="/opt/ml/proces sing/input/code", input_nam e="code",),], outputs=[ProcessingOutput(source='/ opt/ml/processing/ model', destinati on="s3://{}/{}".fo rmat(bucket, prefix), output_na me='byoc-example')], container_entrypoi nt=["python3", "/opt/ ml/processing/input/c ode/training.py"],)</pre> 執行處理工作的範例 Step Functions SageMaker 工作流程 備註：此工作流程範例僅包含 SageMaker 處理工作步驟，而不包含完整的「Step Functions」工作流程。如需完整的工作流程範例，請參閱 AWS Step Functions 數資料科學開發套件文件 SageMaker中的範例筆記本。	

任務	描述	所需技能
	`workflow_graph = Chain([processing_step])` `workflow = Workflow(` `  name="ProcessingWorkflow",` `  definition=workflow_graph,` `  role=workflow_execution_role` `)` `workflow.create()` `# Execute workflow execution = workflow.execute(` `  inputs={` `  "PreprocessingJobName": str(datetime.datetime.now().strftime("%Y%m%d%H%M-%SS")),` `  # Each pre processing job (SageMaker processing job) requires a unique name,` `  }` `)` `execution_output = execution.get_output(wait=True)`	

相關資源

- 處理資料 (Amazon SageMaker 開發人員指南)
- 調整您自己的訓練容器 (Amazon SageMaker 開發人員指南)

使用 Amazon 中的推論管道將預處理邏輯部署到單一端點中的 ML 模型 SageMaker

創建者莫罕·戈瓦達普魯肖塔瑪（AWS），加布里埃爾·羅德里格斯·加西亞（AWS）和馬特烏斯扎倫巴（AWS）

環境：生產	技術：機器學習與人工智慧、容器與微服務	AWS 服務：Amazon SageMaker；Amazon ECR

Summary

此模式說明如何使用 Amazon SageMaker 中的推論管道在單一端點中部署多個管道模型物件。管線模型物件代表不同的機器學習 (ML) 工作流程階段，例如前置處理、模型推論和後處理。為了說明序列連接的管線模型物件的部署，此模式會示範如何部署以內建的線性學習器演算法為基礎的 SciKit-learn 容器和回歸模型。 SageMaker部署託管在中的單一端點後面 SageMaker。

注意：此病毒碼中的部署會使用 ml.m4.2xlarge 執行個體類型。我們建議使用符合您資料大小需求和工作流程複雜性的執行個體類型。如需詳細資訊，請參閱 Amazon SageMaker 定價。此模式使用預先建置的 Docker 映像檔進行 Scikit 學習，但您可以使用自己的 Docker 容器，並將它們整合到您的工作流程中。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Python 3.9
- Amazon SageMaker Python 開發套件和博托 3 庫
- 具有基本 SageMaker 許可和 Amazon 簡單儲存服務 (Amazon S3) 許可的 AWS Identity and Access Management (AWS IAM) 角色

產品版本

- Amazon SageMaker Python 開發套件 2.49.2

架構

目標技術堆疊

- Amazon Elastic Container Registry (Amazon ECR)
- Amazon SageMaker
- Amazon SageMaker 一室
- Amazon Simple Storage Service (Amazon S3)
- 適用於 Amazon 的即時推論端點 SageMaker

目標架構

下圖顯示 Amazon SageMaker 管道模型物件部署的架構。

該圖顯示以下工作流程：

1. SageMaker 筆記型電腦會部署管線模型。
2. S3 儲存貯體存放模型成品。
3. Amazon ECR 從 S3 存儲桶獲取源容器映像。

工具

AWS 工具

- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon SageMaker 是一種受管的 ML 服務，可協助您建置和訓練機器學習模型，然後將其部署到生產就緒的託管環境中。
- Amazon SageMaker Studio 是適用於機器學習的網頁式整合式開發環境 (IDE)，可讓您建置、訓練、偵錯、部署和監控機器學習模型。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

此模式的代碼可在具有 Scikit 學習和線 GitHub 性學習器存放庫的推論管道中找到。

史诗

準備資料集

任務	描述	所需技能
為您的回歸工作準備資料集。	在 Amazon SageMaker 工作室打開筆記本。 若要匯入所有必要的程式庫並初始化您的工作環境，請在筆記本中使用下列範例程式碼： `import sagemaker` `from sagemaker import get_execution_role` `sagemaker_session = sagemaker.Session()` `# Get a SageMaker-compatible role used by this Notebook Instance.` `role = get_execution_role()` `# S3 prefix` `bucket = sagemaker_session.default_bucket()` `prefix = "Scikit-LinearLearner-pipeline-abalone-example"` 若要下載範例資料集，請將下列程式碼新增至您的記事本： `! mkdir abalone_data`	資料科學家

任務	描述	所需技能
	`! aws s3 cp s3://sagemaker-sample-files/datasets/tabular/uci_abalone/abalone.csv ./abalone_data` 注意：此模式中的範例使用 UCI Machine Learning 儲存庫中的鮑魚資料集。	
將資料集上傳到 S3 儲存貯體。	在您先前準備好資料集的筆記本中，新增下列程式碼，將範例資料上傳至 S3 儲存貯體： `WORK_DIRECTORY = "abalone_data"` `train_input = sagemaker_session.upload_data( path="{}/{}".format(WORK_DIRECTORY, "abalone.csv"), bucket=bucket, key_prefix="{}/{}".format(prefix, "train"), )`	資料科學家

使用 SKLearn 創建數據預處理器

任務	描述	所需技能
準備 preprocessor.py 指令碼。	1. 從 GitHub sklearn_abalone_featurizer.py 存儲庫中的 Python 文件複製預處理邏輯，然後將代碼粘貼到一個名為`sklearn_a`	資料科學家

任務	描述	所需技能
	balone_featurizer.py 的單獨 Python 文件中。您可以修改程式碼以符合自訂資料集和自訂工作流程。 2. 將sklearn_abalone_featurizer.py 檔案儲存在專案的根目錄中 (亦即，在執行 SageMaker 筆記本的相同位置)。	

任務	描述	所需技能
創建 SKLearn 預處理器對象。	若要建立可併入最終推論管道的 SKLearn 預處理器物件 (稱為 SKLearn 估算器)，請在筆記本中執行下列程式碼：SageMaker `from sagemaker.sklearn.estimator import SKLearn` `FRAMEWORK_VERSION = "0.23-1"` `script_path = "sklearn_abalone_featurizer.py"` `sklearn_preprocessor = SKLearn(` `entry_point=script_path,` `role=role,` `framework_version=FRAMEWORK_VERSION,` `instance_type="ml.c4.xlarge",` `sagemaker_session=sagemaker_session,` `)` `sklearn_preprocessor.fit({"train": train_input})`	資料科學家

任務	描述	所需技能
測試預處理器的推論。	若要確認您的預處理器已正確定義，請在 SageMaker 筆記本中輸入下列程式碼，以啟動批次轉換工作： `# Define a SKLearn Transformer from the trained SKLearn Estimator` `transformer = sklearn_preprocessor.transformer(` `instance_count=1, instance_type="ml.m5.xlarge", assemble_with="Line", accept="text/csv"` `)` `# Preprocess training input` `transformer.transform(train_input, content_type="text/csv")` `print("Waiting for transform job: " + transformer.latest_transform_job.job_name)` `transformer.wait()` `preprocessed_train = transformer.output_path`	

建立機器學習模型

任務	描述	所需技能
創建一個模型對象。	若要根據線性學員演算法建立模型物件，請在 SageMaker 記事本中輸入下列程式碼： `import boto3 from sagemaker .image_uris import retrieve` `ll_image = retrieve( "linear-learner", boto3.Session().region_name) s3_ll_output_key_prefix = "ll_training_output" s3_ll_output_location = "s3://{}/{}/{}/{}" .format( bucket, prefix, s3_ll_output_key_prefix, "ll_model" )` `ll_estimator = sagemaker.estimator.Estimator( ll_image, role, instance_count=1, instance_type="ml.m4.2xlarge", volume_size=20, max_run=3600, input_mode="File", output_path=s3_ll_output_location,`	資料科學家

任務	描述	所需技能
	<pre> sagemaker_session= sagemaker_session,) ll_estimator.s et_hyperparameters (feature_dim=10, predictor_type="re gressor", mini_batc h_size=32) ll_train_data = sagemaker.inputs.T rainingInput(preprocessed_train , distribution="Full yReplicated", content_type="text /csv", s3_data_type="S3Pr efix",) data_channels = {"train": ll_train_ data} ll_estimator.fit(inpu ts=data_channels, logs=True)</pre> 上述程式碼會從公用 Amazon ECR 登錄擷取模型的相關 Amazon ECR Docker 映像，建立估算器物件，然後使用該物件來訓練回歸模型。	

部署最終管道

任務	描述	所需技能
部署管線模型。	若要建立管線模型物件 (也就是預處理器物件) 並部署物件，請在 SageMaker 筆記本中輸入下列程式碼： `from sagemaker.model import Model` `from sagemaker.pipeline import PipelineModel` `import boto3` `from time import gmtime, strftime` `timestamp_prefix = strftime("%Y-%m-%d-%H-%M-%S", gmtime())` `scikit_learn_inferencee_model = sklearn_preprocessor.create_model()` `linear_learner_model = ll_estimator.create_model()` `model_name = "inference-pipeline-" + timestamp_prefix` `endpoint_name = "inference-pipeline-ep-" + timestamp_prefix` `sm_model = PipelineModel(` `name=model_name, role=role, models=[scikit_learn_infe`	資料科學家

任務	描述	所需技能
	rencee_model, linear_learner_model]) sm_model.deploy(init ial_instance_count =1, instance_type="ml. c4.xlarge", endpoint_ name=endpoint_name) 注意：您可以調整模型物件中使用的例證類型，以符合您的需求。	
測試推論。	若要確認端點是否正常運作，請在 SageMaker 筆記本中執行下列範例推論程式碼： from sagemaker.predicto r import Predictor from sagemaker.serializ ers import CSVSerial izer payload = "M, 0.44, 0.365, 0.125, 0.516, 0.2155, 0.114, 0.155" actual_rings = 10 predictor = Predictor(endpoint_name=endp oint_name, sagemaker _session=sagemaker _session, serialize r=CSVSerializer()) print(predictor .predict(payload))	資料科學家

相關資源

- 使用 Amazon SageMaker 推論管道和 Scikit-學習 (AWS Machine Learning 部落格) 在進行預測之前預先處理輸入資料
- 使用 Amazon 的端對端 Machine Learning SageMaker (GitHub)

使用 RAG 和提示，開發先進的生成式 AI 聊天助理 ReAct

創建者：庫馬爾傑亞拉揚（AWS），橋俊東（AWS），卡拉楊（AWS），基奧瓦·傑克遜（AWS），諾亞·漢密爾頓（AWS）和曹操（AWS）

代碼存儲庫：genai-bedrock-chatbot	環境：PoC 或試點	技術：機器學習和人工智能; 數據庫 DevOps; 無服務器
AWS 服務：Amazon 基岩；Amazon ECS；Amazon Kendra；AWS Lambda		

Summary

一家典型的公司有 70% 的數據被困在孤立的系統中。您可以使用生成式人工智慧型聊天助理，透過自然語言互動，獲得這些資料孤島之間的洞察和關係。為了充分利用生成式 AI，輸出必須值得信賴、準確且包含可用的企業資料。以聊天為基礎的助理成功取決於下列因素：

- 生成 AI 模型（例如人為克勞德 2）
- 資料來源向量化
- 用於提示模型的進階推理技巧，例如ReAct 框架

此模式提供資料擷取方法，例如 Amazon Simple Storage Service (Amazon S3) 貯體、AWS Glue 和 Amazon Relational Database Service 服務 (Amazon RDS) 等資料來源。值是通過交錯檢索增強生成（RAG）與 chain-of-thought 方法從該數據中獲得的。結果支援複雜的聊天型助理對話，這些對話會利用您公司儲存的整個資料。

此模式使用 Amazon SageMaker 手冊和定價資料表做為範例，探索生成式 AI 聊天型助理的功能。您將建立聊天型助理，透過回答有關定價和 SageMaker 服務功能的問題，協助客戶評估服務。該解決方案使用 Streamlit 庫來構建前端應用程序，並使用 LangChain 框架來開發由大型語言模型（LLM）提供支持的應用程序後端。

對聊天型助理進行查詢時，系統會以初始意圖分類來傳送至三種可能的工作流程之一。最複雜的工作流程結合了一般諮詢指導和複雜的定價分析。您可以調整模式以適應企業、企業和工業使用案例。

先決條件和限制

先決條件

- 安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)
- AWS Cloud Development Kit (AWS CDK) 工具組 2.114.1 或更新版本已安裝並設定
- 對 Python 和 AWS CDK 的基本熟悉程度
- 安裝的 Git
- 碼頭工人已安裝
- 安裝和設定 Python 3.11 或更新版本 (如需詳細資訊，請參閱「工具」一節)
- 使用 AWS CDK 啟動載入的作用中 AWS 帳戶
- Amazon 基岩服務中啟用了 Amazon 泰坦和人為克勞德模型訪問
- AWS 安全登入資料`AWS_ACCESS_KEY_ID`，包括在終端機環境中正確設定

限制

- LangChain 不支持每個 LLM 進行流媒體。支持人工克勞德模型，但來自 AI21 實驗室的模型不是。
- 此解決方案部署到單一 AWS 帳戶。
- 此解決方案只能部署在提供 Amazon 基岩和 Amazon Kendra 的 AWS 區域。如需可用性的相關資訊，請參閱 Amazon 基岩和 Amazon Kendra 的文件。

產品版本

- Python 版本 3.11 或更高版本
- 流媒體版本 1.30.0 或更高版本
- 串流聊天版本 0.1.1 或更新版本
- LangChain 版本 0.1.12 或更新版本
- AWS CDK 版本 2.132.1 或更新版本

架構

目標技術堆疊

- Amazon Athena

- Amazon Bedrock
- Amazon Elastic Container Service (Amazon ECS)
- AWS Glue
- AWS Lambda
- Amazon S3
- Amazon Kendra
- Elastic Load Balancing

目標架構

AWS CDK 程式碼會部署在 AWS 帳戶中設定聊天型助理應用程式所需的所有資源。下圖中顯示的基於聊天的助理應用程序旨在回答用戶的 SageMaker 相關查詢。使用者透過應用程式負載平衡器連線到包含託管 Streamlit 應用程式之 Amazon ECS 叢集的 VPC。協調流程 Lambda 函數會連接至應用程式。S3 儲存貯體資料來源會透過 Amazon Kendra 和 AWS Glue 將資料提供給 Lambda 函數。Lambda 函數連接至 Amazon 基岩，以回答聊天型助理使用者的查詢 (問題)。

1. 協調流程 Lambda 函數會將 LLM 提示請求傳送至 Amazon 基岩模型 (克勞德 2)。
2. Amazon 基岩將 LLM 響應發送回協調流程 Lambda 函數。

協調流程 Lambda 函數內的邏輯流程

當使用者透過 Streamlit 應用程式提出問題時，會直接叫用協調流程 Lambda 函數。下圖顯示叫用 Lambda 函數時的邏輯流程。

- 步驟 1-輸入`query`（問題）被分為三個意圖之一：
  - 一般 SageMaker 指引問題
  - 一般 SageMaker 定價 (訓練/推論) 問題
  - 與定價相關的 SageMaker 複雜問題
- 第 2 步-輸入`query`啟動三個服務之一：
  - `RAG Retrieval service`，從 Amazon Kendra 向量資料庫擷取相關內容，並透過 Amazon 基岩呼叫 LLM，將擷取的內容匯總為回應。

  - `Database Query service`，它使用-LLM，數據庫元數據和相關表中的示例行將輸query入轉換為 SQL 查詢。資料庫查詢服務會透過 Amazon Athena 針對 SageMaker 定價資料庫執行 SQL 查詢，並將查詢結果摘要為回應。
  - `In-context ReACT Agent service`，在提供響應之前query將輸入分解為多個步驟。代理在推理過程中使用`RAG Retrieval service`和`Database Query service`作為工具來檢索相關信息。推理和操作過程完成後，代理程序生成最終答案作為響應。
- 步驟 3 — 來自協調流程 Lambda 函數的回應會傳送至 Streamlit 應用程式做為輸出。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中分析資料。
- Amazon 基岩是一項全受管服務，可透過統一的 API，讓領先的 AI 新創公司和 Amazon 提供的高效能基礎模型 (FM) 供您使用。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。此模式使用 AWS Glue 爬行程式和 AWS Glue 資料型錄表格。
- Amazon Kendra 是一種智慧型搜尋服務，它使用自然語言處理和進階機器學習演算法，傳回資料中搜尋問題的特定答案。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。

代碼存儲庫

此模式的代碼可在 GitHub genai-bedrock-chatbot存儲庫中找到。

代碼存儲庫包含以下文件和文件夾：

- `assets`文件夾-靜態資產架構圖和公共數據集
- `code/lambda-container`資料夾 — 在 Lambda 函數中執行的 Python 程式碼
- `code/streamlit-app`資料夾 — 在 Amazon ECS 中作為容器映像執行的 Python 程式碼
- `tests`資料夾 — 為了單元測試 AWS CDK 建構而執行的 Python 檔案
- `code/code_stack.py`-AWS CDK 構建用於創建 AWS 資源的 Python 文件
- `app.py`— 用於在目標 AWS 帳戶中部署 AWS 資源的 AWS CDK 堆疊 Python 檔案
- `requirements.txt`— 必須為 AWS CDK 安裝的所有 Python 相依性清單
- `requirements-dev.txt`— 必須為 AWS CDK 安裝的所有 Python 相依性清單，才能執行單元測試套件
- `cdk.json`— 用於提供旋轉資源所需的值的輸入文件

注意：AWS CDK 程式碼使用 AWS 管理的 L3 (第 3 層) 建構和 AWS Identity and Access Management (IAM) 政策來部署解決方案。

最佳實務

- 此處提供的程式碼範例僅適用於 proof-of-concept (PoC) 或試驗示範。如果您想要將程式碼帶到生產環境中，請務必使用下列最佳作法：
  - Amazon S3 存取日誌已啟用。
  - 已啟用 VPC 流程記錄。
  - Amazon Kendra 企業版索引已啟用。
- 設定 Lambda 函數的監控和警示。如需詳細資訊，請參閱監控 Lambda 函數和疑難排解。如需使用 Lambda 函數時的一般最佳實務，請參閱 AWS 文件。

史诗

在本機電腦上設定 AWS 登入資料

任務	描述	所需技能
將部署堆疊的帳戶和 AWS 區域匯出變數。	若要使用環境變數為 AWS CDK 提供 AWS 登入資料，請執行下列命令。 `export CDK_DEFAU LT_ACCOUNT=<12 Digit AWS Account Number> export CDK_DEFAU LT_REGION=<region>`	DevOps 工程師, AWS DevOps
設定 AWS CLI 設定檔。	若要為帳戶設定 AWS CLI 設定檔，請遵循 AWS 文件中的指示。	DevOps 工程師, AWS DevOps

設定您的環境

任務	描述	所需技能
克隆本地計算機上的存儲庫。	要克隆存儲庫，請在終端中運行以下命令。 `git clone https://g ithub.com/awslabs/ genai-bedrock-chat bot.git`	DevOps 工程師, AWS DevOps
設定 Python 虛擬環境並安裝必要的相依性。	若要設定 Python 虛擬環境，請執行下列命令。 `cd genai-bedrock-chat bot python3 -m venv .venv`	DevOps 工程師, AWS DevOps

任務	描述	所需技能
	`source .venv/bin/activate` 若要設定所需的相依性，請執行下列命令。 `pip3 install -r requirements.txt`	
設定 AWS CDK 環境並合成 AWS CDK 程式碼。	1. 若要在您的 AWS 帳戶中設定 AWS CDK 環境，請執行下列命令。 `cdk bootstrap aws://ACCOUNT-NUMBER/REGION` 2. 若要將程式碼轉換為 AWS CloudFormation 堆疊組態，請執行命令`cdk synth`。	DevOps 工程師, AWS DevOps

設定和部署聊天型助理應用程式

任務	描述	所需技能
提供克勞德模型存取權限。	若要為您的 AWS 帳戶啟用人性克勞德模型存取權限，請按照 Amazon 基岩文件中的說明進行操作。	AWS DevOps
在帳戶中部署資源。	若要使用 AWS CDK 在 AWS 帳戶中部署資源，請執行下列動作： 1. 在複製存放庫的根目錄中，在`cdk.json`檔	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	案中提供logging參數的輸入。範例值為INFODEBUG、WARN、和ERROR。 這些值定義 Lambda 函數和流光應用程式的記錄層級訊息。 2. 複製儲存庫根目錄中的app.py檔案包含用於部署的 AWS CloudFormation 堆疊名稱。預設堆疊名稱為chatbot-stack 。 3. 若要部署資源，請執行命令cdk deploy。 此命cdk deploy令使用 L3 建構建立多個 Lambda 函數，將文件和 CSV 資料集檔案複製到 S3 儲存貯體。 4. 命令完成後，登入 AWS 管理主控台、開啟主控 CloudFormation 台，然後檢閱堆疊是否成功部署。 成功部署後，您可以使用 [CloudFormation 輸出] 區段中提供的 URL 存取以聊天為基礎的助理應用程式。	

任務	描述	所需技能
執行 AWS Glue 爬行者程式並建立資料目錄表格。	AWS Glue 爬行程式是用來維持資料結構描述的動態。此解決方案會視需求執行爬蟲，在 AWS Glue 資料型錄表格中建立和更新分割區。將 CSV 資料集檔案複製到 S3 儲存貯體之後，執行 AWS Glue 爬蟲程式並建立資料目錄表格結構描述以進行測試： 1. 瀏覽至 AWS Glue 主控台。 2. 在導覽窗格的 [資料目錄] 下，選擇 [爬行者程式]。 3. 選取具有尾碼`sagemaker-pricing-crawler` 的爬行者程式。 4. 執行爬行者程式。 5. 爬行者程式成功執行之後，便會建立 AWS Glue 資料型錄表格。 注意：AWS CDK 程式碼會設定 AWS Glue 爬行程式視需求執行，但您也可以排程定期執行。	DevOps 工程師, AWS DevOps

任務	描述	所需技能
啟動文件索引。	將檔案複製到 S3 儲存貯體之後，請使用 Amazon Kendra 對其進行編目並編製索引： 1. 瀏覽至 Amazon Kendra 主控台。 2. 選取帶有尾碼的索引 `chatbot-index` 。 3. 在導覽窗格中，選擇 [資料來源]，然後選取具有尾碼的資料來源連接器`chatbot-index` 。 4. 選擇「立即同步」以啟動索引程序。 注意：AWS CDK 程式碼會將 Amazon Kendra 索引同步設定為隨需執行，但您也可以使用排程參數定期執行。	AWS DevOps、 DevOps 工程師

清理解決方案中的所有 AWS 資源

任務	描述	所需技能
移除 AWS 資源。	測試解決方案之後，請清理資源： 1. 若要移除解決方案部署的 AWS 資源，請執行命令`cdk destroy`。 2. 刪除兩個 S3 儲存貯體中的所有物件，然後移除值區。	DevOps 工程師, AWS DevOps

任務	描述	所需技能
	如需詳細資訊，請參閱刪除值區。	

故障診斷

問題	解決方案
AWS CDK 會傳回錯誤。	如需 AWS CDK 問題的相關說明，請參閱疑難排解常見 AWS CDK 問題。

相關資源

- Amazon 基岩：
  - 模型存取
  - 基礎模型的推論參數
- 使用 Python 構建函數
- 開始使用 AWS CDK
- 在 Python 中使用 AWS CDK
- AWS 上的生成 AI 應用程式建置器
- LangChain 文件
- 流文檔

其他資訊

AWS CDK 命令

使用 AWS CDK 時，請記住下列有用的命令：

- 列出應用程式中的所有堆疊

```
cdk ls
```

- 發出合成的 AWS 範本 CloudFormation

```
cdk synth
```

- 將堆疊部署到您的預設 AWS 帳戶和區域

```
cdk deploy
```

- 將已部署的堆疊與目前狀態進行比較

```
cdk diff
```

- 開啟 AWS CDK 文件

```
cdk docs
```

- 刪除 CloudFormation 堆疊並移除 AWS 部署的資源

```
cdk destroy
```

使用 Amazon 基岩代理程式和知識庫，開發以聊天為基礎的全自動化助理

由喬俊東（AWS），卡拉楊（AWS），基奧瓦·傑克遜（AWS），諾亞漢密爾頓（AWS），庫馬爾·傑亞拉揚（AWS）和曹帥（AWS）創建

代碼存儲庫：genai-bedrock-agent-chatbot	環境：PoC 或試點	技術：機器學習與 AI；無伺服器
AWS 服務：Amazon 基岩；AWS CDK；AWS Lambda		

Summary

許多組織在建立能夠協調各種資料來源以提供全方位解答的聊天型助理時面臨挑戰。這種模式提供了一種解決方案，用於開發基於聊天的助理，該助理能夠通過簡單的部署來回答文檔和數據庫中的查詢。

從 Amazon Bdrock 開始，這項全受管的生成人工智慧 (AI) 服務提供各式各樣的進階基礎模型 (FMs)。這有助於高效地建立生成式 AI 應用程式，並專注於隱私權和安全性。在文件擷取的內容中，擷取增強一代 (RAG) 是一項關鍵功能。它使用知識庫來使用外部來源的上下文相關信息來增強 FM 提示。Amazon OpenSearch 無伺服器索引可作為 Amazon 基岩知識庫背後的向量資料庫。透過仔細迅速的工程來增強這項整合，以最大程度地減少不準確性，並確保回應已錨定在事實文件中。對於資料庫查詢，Amazon 基岩的 FM 會將文字查詢轉換為結構化 SQL 查詢，並納入特定參數。這可讓您從 AWS Glue 資料庫管理的資料庫精確擷取資料。Amazon Athena 用於這些查詢。

為了處理更複雜的查詢，獲得全面的答案需要來自文檔和數據庫的信息。Amazon Bedrock 的代理程式是一項生成式 AI 功能，可協助您建立自主代理程式，以了解複雜的任務，並將其分解為更簡單的協調任務。Amazon Bdrock 自主代理程式協助從簡化任務中擷取的洞察結合，可增強資訊的合成，從而獲得更全面和詳盡的答案。此模式示範如何使用 Amazon Bdrock 以及自動化解決方案中的相關生成 AI 服務和功能來建立聊天型助理。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 泊塢視窗，已安裝
- 已安裝並啟動至或 AWS 區域的 AWS Cloud Development Kit (AWS CDK) `us-east-1` `us-west-2`
- 已安裝 AWS CDK 工具組 2.114.1 版或更新版本
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝版本 3.11 或更新版本
- 在 Amazon 基岩中，可以訪問克勞德 2，克勞德 2.1，克勞德即時和泰坦嵌入 G1-文本

限制

- 此解決方案部署到單一 AWS 帳戶。
- 此解決方案只能部署在支援 Amazon 基岩和 Amazon OpenSearch 無伺服器的 AWS 區域。如需詳細資訊，請參閱 Amazon 基岩和 Amazon OpenSearch 無伺服器相關文件。

產品版本

- 羅馬索引版本 0.10.6 或更新版本
- 方塊煉金術版本 2.0.23 或更高版本
- 開放搜索-PY 版本 2.4.2 或更高版本
- 請求版本 1.2.3 或更新版本
- 適用於蟒蛇的 AWS 開發套件 (博圖 3) 1.34.57 版或更新版本

架構

目標技術堆疊

AWS Cloud Development Kit (AWS CDK) 是一種開放原始碼軟體開發架構，可在程式碼中定義雲端基礎設施，並透過 AWS CloudFormation 佈建雲端基礎設施。此模式中使用的 AWS CDK 堆疊會部署下列 AWS 資源：

- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Glue 資料型錄，適用於 AWS Glue 資料庫元件
- AWS Lambda
- AWS Identity and Access Management (IAM)

- Amazon OpenSearch 無服務器
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- AWS Fargate
- Amazon Virtual Private Cloud (Amazon VPC)
- Application Load Balancer

目標架構

圖表顯示在單一 AWS 區域內使用多個 AWS 服務的全方位 AWS 雲端原生設定。聊天型助理的主要介面是 Amazon ECS 叢集上託管的 Streamlit 應用程式。應用程式負載平衡器管理可存取性 透過此介面進行的查詢會啟動 `Invocation` Lambda 函數，然後與 Amazon 基岩的代理程式進行介面。此代理程式透過諮詢 Amazon 基岩的知識庫或叫用 Lambda 函數來回應使用者查詢。`Agent executor`此函數會依照預先定義的 API 結構描述觸發一組與代理程式相關聯的動作。Amazon 基岩的知識庫使用 OpenSearch 無伺服器索引做為其向量資料庫基礎。此外，此`Agent executor`函數會產生 SQL 查詢，這些查詢會透過 Amazon Athena 針對 AWS Glue 資料庫執行。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中分析資料。
- Amazon 基岩是一項全受管服務，可透過統一的 API，讓領先的 AI 新創公司和 Amazon 提供的高效能基礎模型 (FM) 供您使用。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。

- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。此模式使用 AWS Glue 爬行程式和 AWS Glue 資料型錄表格。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon OpenSearch 無伺服器是 Amazon OpenSearch 服務的隨需無伺服器組態。在此模式中，OpenSearch 無伺服器索引可做為 Amazon 基岩知識庫的向量資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

- 流光是一個用於創建數據應用程序的開源 Python 框架。

代碼存儲庫

此模式的代碼可在 GitHub genai-bedrock-agent-chatbot存儲庫中找到。代碼存儲庫包含以下文件和文件夾：

- `assets`folder — 靜態資產，例如架構圖和公開資料集。
- `code/lambdas/action-lambda`資料夾 — Lambda 函數的 Python 程式碼，可做為 Amazon 基岩代理程式的動作。
- `code/lambdas/create-index-lambda`資料夾 — 建立 OpenSearch 無伺服器索引之 Lambda 函數的 Python 程式碼。
- `code/lambdas/invoke-lambda`資料夾 — 叫用 Amazon 基岩代理程式的 Lambda 函數的 Python 程式碼，該代理程式會直接從流光應用程式呼叫。
- `code/lambdas/update-lambda`資料夾 — Lambda 函數的 Python 程式碼，可在透過 AWS CDK 部署 AWS 資源後更新或刪除資源。
- `code/layers/boto3_layer`資料夾 — 建立可在所有 Lambda 函數共用的 Boto3 層的 AWS CDK 堆疊。
- `code/layers/opensearch_layer`資料夾 — 建立 OpenSearch 無伺服器層的 AWS CDK 堆疊，用於安裝所有相依性以建立索引。
- `code/streamlit-app`資料夾 — 在 Amazon ECS 中作為容器映像執行的 Python 程式碼
- `code/code_stack.py`— AWS CDK 會建構建建立 AWS 資源的 Python 檔案。
- `app.py`— 在目標 AWS 帳戶中部署 AWS 資源的 AWS CDK 堆疊 Python 檔案。

- `requirements.txt`— 必須為 AWS CDK 安裝的所有 Python 相依性清單。
- `cdk.json`— 用來提供建立資源所需值的輸入檔案。另外，在`context/config`字段中，您可以相應地自定義解決方案。如需有關自訂的詳細資訊，請參閱其他資訊一節。

最佳實務

- 此處提供的程式碼範例僅用於 proof-of-concept (PoC) 或試驗目的。如果您想要將程式碼帶到生產環境中，請務必使用下列最佳作法：
  - 啟用 Amazon S3 存取記錄
  - 啟用 VPC 流程記錄檔
- 設定 Lambda 函數的監控和警示。如需詳細資訊，請參閱監控 Lambda 函數和疑難排解。如需最佳實務，請參閱使用 AWS Lambda 函數的最佳實務。

史诗

在本機工作站上設定 AWS 登入資料

任務	描述	所需技能
匯出帳戶和區域的變數。	若要使用環境變數為 AWS CDK 提供 AWS 登入資料，請執行下列命令。 `export CDK_DEFAU LT_ACCOUNT=<12-digit AWS account number> export CDK_DEFAU LT_REGION=<Region>`	AWS DevOps、 DevOps 工程師
設定 AWS CLI 命名的設定檔。	若要為帳戶設定 AWS CLI 命名的設定檔，請按照組態和登入資料檔案設定中的指示進行操作。	AWS DevOps、 DevOps 工程師

設定您的環境

任務	描述	所需技能
將存放庫克隆到您的本地工作站。	要克隆存儲庫，請在終端中運行以下命令。 `git clone https://github.com/awslabs/genai-bedrock-agent-chatbot.git`	DevOps 工程師, AWS DevOps
設定虛 Python 環境。	若要設定 Python 虛擬環境，請執行下列命令。 `cd genai-bedrock-agent-chatbot` `python3 -m venv .venv` `source .venv/bin/activate` 若要設定所需的相依性，請執行下列命令。 `pip3 install -r requirements.txt`	DevOps 工程師, AWS DevOps
設定 AWS CDK 環境。	若要將程式碼轉換為 AWS CloudFormation 範本，請執行命令`cdk synth`。	AWS DevOps、 DevOps 工程師

設定和部署應用程式

任務	描述	所需技能
在帳戶中部署資源。	若要使用 AWS CDK 在 AWS 帳戶中部署資源，請執行下列動作： 1. 在複製存放庫的根目錄中，在`cdk.json`檔案中提供記錄參數的輸入。範例值為`INFO`、`DEBUG`、`WARN`、和`ERROR`。 這些值定義 Lambda 函數和 Streamlit 應用程式的記錄層級訊息。 2. 複製儲存庫根目錄中的`cdk.json`檔案包含用於部署的 AWS CloudFormation 堆疊名稱。預設堆疊名稱為`chatbot-stack` 。預設的 Amazon 基岩代理程式名稱為`ChatbotBedrockAgent` ，預設的 Amazon 基岩代理程式別名為。`Chatbot_Agent` 3. 若要部署資源，請執行命令`cdk deploy`。 此命`cdk deploy`令使用第 3 層建構建立多個 Lambda 函數，將文件和 CSV 資料集檔案複製到 S3 儲存貯體。它還為 Amazon 基岩代理程式部署 Amazon 基岩代	DevOps 工程師, AWS DevOps

任務	描述	所需技能
	理程式、知識庫和 Action group Lambda 函數。 4. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/cloudformation/ 開啟 CloudFormation 主控台。 5. 確認堆疊已成功部署。如需指示，請參閱在 AWS CloudFormation 主控台上檢閱您的堆疊。 成功部署之後，您可以使用主控台中 [輸出] 索引標籤上提供的 URL，存取以聊天為基礎的助理應用程式 CloudFormation 。	

清理解決方案中的所有 AWS 資源

任務	描述	所需技能
移除 AWS 資源。	測試解決方案之後，若要清理資源，請執行命令cdk destroy。	AWS DevOps、 DevOps 工程師

相關資源

AWS 文件

- Amazon 基岩資源：
  - 模型存取
  - 基礎模型的推論參數

  - Amazon 基岩代理
  - Amazon 基岩知識庫
- 使用 Python 構建函數
- AWS CDK 資源：
  - 開始使用 AWS CDK
  - 疑難排解常見的 AWS CDK 問題
  - 在 Python 中使用 AWS CDK
- AWS 上的生成 AI 應用程式建置器

其他 AWS 資源

- Amazon OpenSearch 無伺服器的向量引擎

其他資源

- LlamaIndex 文件
- 流文檔

其他資訊

使用您自己的資料自訂聊天型助理

若要整合您的自訂資料以部署解決方案，請遵循下列結構化準則。這些步驟旨在確保順暢且有效率的整合程序，讓您能夠使用自訂資料有效地部署解決方案。

用於知識庫資料整合

資料準備

1. 找到`assets/knowledgebase_data_source/`目錄。
2. 將資料集置於此資料夾中。

組態調整

1. 開啟 `cdk.json` 檔案。

2. 導覽至context/configure/paths/knowledgebase_file_name欄位，然後對其進行相應更新。
3. 瀏覽至bedrock_instructions/knowledgebase_instruction欄位，然後更新欄位，以準確反映新資料集的細微差別和內容。

用於結構化資料整合

資料組織

1. 在目assets/data_query_data_source/錄中，建立子目錄，例如tabular_data。
2. 將您的結構化數據集（可接受的格式包括 CSV，JSON，ORC 和實木複合地板）放入此新創建的子文件夾中。
3. 如果您要連線至現有的資料庫，請更新create_sql_engine()中的函數code/lambda/action-lambda/build_query_engine.py以連線至您的資料庫。

組態和程式碼更新

1. 在cdk.json檔案中，更新context/configure/paths/athena_table_data_prefix欄位以與新資料路徑對齊。
2. code/lambda/action-lambda/dynamic_examples.csv透過整合與資料集對應的新文字轉 SQL 範例來進行修訂。
3. 修改code/lambda/action-lambda/prompt_templates.py以鏡像結構化資料集的屬性。
4. 在cdk.json檔案中，更新context/configure/bedrock_instructions/action_group_description欄位以說明 Action group Lambda 函數的用途和功能。
5. 在assets/agent_api_schema/artifacts_schema.json檔案中，說明 L Action group ambda 函數的新功能。

一般更新

在cdk.json檔案的本context/configure/bedrock_instructions/agent_instruction節中，考慮到新整合的資料，提供 Amazon Bdrock 代理程式預期功能和設計目的的的完整說明。

使用 Amazon 基岩和 Amazon Transcribe 來記錄語音輸入的機構知識

由庫馬爾傑亞拉揚（AWS），橋俊東（AWS），吳梅根（AWS）和拉吉夫·庫帕迪伊（AWS）創建

代碼存儲庫：genai-knowledge-capture	環境：PoC 或試點	技術：機器學習與人工智慧、企業生產力、雲端原生
AWS 服務：Amazon 基岩；AWS CDK；AWS Lambda；Amazon SNS；AWS Step Functions；Amazon Transcribe		

Summary

捕捉機構知識對於確保組織的成功和韌性至關重要。機構知識代表了員工隨著時間的推移積累的集體智慧，見解和經驗，通常在自然界中默契並非正式地傳下來。這種豐富的信息包括獨特的方法，最佳實踐以及解決複雜問題的解決方案，這些問題可能無法在其他地方記錄。通過正式化和記錄這些知識，公司可以保留機構記憶，促進創新，增強決策過程，並加快新員工的學習曲線。此外，它還促進協作，賦予個人權力，並培養持續改進的文化。最終，利用機構知識可以幫助公司利用其最有價值的資產（勞動力的集體智能）在動態的業務環境中應對挑戰，推動增長並保持競爭優勢。

這種模式解釋瞭如何通過高級員工的錄音來捕獲機構知識。它使用 Amazon Transcribe 和 Amazon 基岩進行系統的文檔和驗證。通過記錄這些非正式知識，您可以保留它，並與後續的員工群體共享。這項努力通過整合通過直接經驗獲得的實踐知識來支持卓越運營並提高培訓計劃的有效性。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 泊塢視窗，已安裝
- AWS Cloud Development Kit (AWS CDK) 2.114.1 版或更新版本，已安裝並啟動載入或 AWS 區域 `us-east-1` `us-west-2`

- 已安裝 AWS CDK 工具組 2.114.1 版或更新版本
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝版本 3.12 或更新版本
- 建立 Amazon Transcribe、Amazon 基岩、Amazon Simple Storage Service (Amazon S3) 和 AWS Lambda 資源的許可

限制

- 此解決方案部署到單一 AWS 帳戶。
- 此解決方案只能在提供 Amazon 基岩和 Amazon Transcribe 的 AWS 區域部署。如需可用性的相關資訊，請參閱 Amazon 基岩和 Amazon 轉錄的文件。
- 音頻文件必須是 Amazon Transcribe 支持的格式。如需支援的格式清單，請參閱 Transcribe 文件中的媒體格式。

產品版本

- 適用於蟒蛇的 AWS 開發套件 (博圖 3) 1.34.57 版或更新版本
- LangChain 版本 0.1.12 或更新版本

架構

該架構代表 AWS 上的無伺服器工作流程。AWS Step Functions 可協調 Lambda 函數，以進行音訊處理、文字分析和文件產生。下圖顯示了 Step Functions 工作流程，也稱為狀態機。

狀態機中的每個步驟都由不同的 Lambda 函數處理。以下是文件產生程序中的步驟：

1. `preprocess`Lambda 函數會驗證傳遞至 Step Functions 數的輸入，並列出提供的 Amazon S3 URI 資料夾路徑中存在的所有音訊檔案。工作流程中的下游 Lambda 函數會使用檔案清單來驗證、摘要和產生文件。
2. `transcribe`Lambda 函數使用 Amazon Transcribe 將音頻文件轉換為文本記錄。此 Lambda 函數負責啟動轉錄程序，並準確地將語音轉換為文字，然後儲存以供後續處理。
3. L `validate` ambda 函數會分析文字記錄，判斷回應初始問題的相關性。透過 Amazon 基岩使用大型語言模型 (LLM)，它可以識別主題相關答案，並將主題答案與離題回應分開。
4. L `summarize` ambda 函數使用 Amazon 基岩來生成一致且簡潔的主題答案摘要。

5. generateLambda 函數將摘要彙編成一個結構良好的文檔。它可以根據預定義的模板格式化文檔，並包括任何其他必要的內容或數據。
6. 如果任何 Lambda 函數失敗，您會透過亞馬遜簡單通知服務 (Amazon SNS) 收到電子郵件通知。

在整個過程中，AWS Step Functions 數可確保以正確的順序啟動每個 Lambda 函數。該狀態機具有 parallel 處理的能力，以提高效率。Amazon S3 儲存貯體充當中央儲存庫，透過管理涉及的各種媒體和文件格式來支援工作流程。

工具

AWS 服務

- Amazon 基岩是一項全受管服務，可透過統一的 API，讓領先的 AI 新創公司和 Amazon 提供的高效能基礎模型 (FM) 供您使用。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。
- Amazon Transcribe 是一種自動語音辨識服務，使用機器學習模型將音訊轉換為文字。

其他工具

- LangChain是用於開發由大型語言模型（LLM）提供支持的應用程序的框架。

代碼存儲庫

此模式的代碼可在 GitHub genai-knowledge-capture存儲庫中找到。

代碼存儲庫包含以下文件和文件夾：

- assets資料夾 — 解決方案的靜態資產，例如架構圖和公開資料集
- code/lambdas資料夾 — 所有 Lambda 函數的 Python 程式碼

  - `code/lambdas/generate`文件夾-從 S3 存儲桶中的摘要數據生成文檔的 Python 代碼
  - `code/lambdas/preprocess`文件夾-處理 Step Functions 狀態機輸入的 Python 代碼
  - `code/lambdas/summarize`文件夾-使用 Amazon 基岩服務總結轉錄數據的 Python 代碼
  - `code/lambdas/transcribe`文件夾-使用 Amazon Transcribe 將語音數據（音頻文件）轉換為文本的 Python 代碼
  - `code/lambdas/validate`文件夾-驗證所有答案是否與同一主題有關的 Python 代碼
- `code/code_stack.py`— 用於建立 AWS 資源的 AWS CDK 建構 Python 檔案
- `app.py`— 用於在目標 AWS 帳戶中部署 AWS 資源的 AWS CDK 應用程式 Python 檔案
- `requirements.txt`— 必須為 AWS CDK 安裝的所有 Python 相依性清單
- `cdk.json`— 輸入檔案，以提供建立資源所需的值

最佳實務

提供的程式碼範例僅用於 proof-of-concept (PoC) 或試驗目的。如果您想要將解決方案推向生產環境，請使用下列最佳作法：

- 啟用 Amazon S3 存取記錄
- 啟用 VPC 流程記錄

史诗

在本機工作站上設定 AWS 登入資料

任務	描述	所需技能
匯出帳戶和 AWS 區域的變數。	若要使用環境變數為 AWS CDK 提供 AWS 登入資料，請執行下列命令。 `export CDK_DEFAULT_ACCOUNT=<12-digit AWS account number>` `export CDK_DEFAULT_REGION=<Region>`	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
設定 AWS CLI 命名的設定檔。	若要為帳戶設定 AWS CLI 命名的設定檔，請按照組態和登入資料檔案設定中的指示進行操作。	AWS DevOps、 DevOps 工程師

設定您的環境

任務	描述	所需技能
將存放庫克隆到您的本地工作站。	要克隆genai-knowledge-capture存儲庫，請在終端中運行以下命令。 `git clone https://github.com/aws-samples/genai-knowledge-capture`	AWS DevOps、 DevOps 工程師
(選擇性) 取代音訊檔案。	若要自訂範例應用程式以合併您自己的資料，請執行下列動作： 1. 導覽至複製的存放庫中的`assets/audio_samples` 資料夾。 2. 刪除包含範例音訊檔案的資料夾。 3. 為您要分析的每個主題建立一個資料夾。 4. 將音頻文件傳輸到各自的文件夾。	AWS DevOps、 DevOps 工程師
設定虛 Python 環境。	若要設定 Python 虛擬環境，請執行下列命令。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	`cd genai-knowledge-capture` `python3 -m venv .venv` `source .venv/bin/activate` `pip install -r requirements.txt`	
合成 AWS CDK 程式碼。	若要將程式碼轉換為 AWS CloudFormation 堆疊組態，請執行下列命令。 `cdk synth`	AWS DevOps、 DevOps 工程師

設定和部署解決方案

任務	描述	所需技能
提供基礎模型存取權。	為您的 AWS 帳戶啟用對人類克勞德 3 十四行詩模型的存取權。如需指示，請參閱基岩文件中的新增模型存取權限。	AWS DevOps
在帳戶中部署資源。	若要使用 AWS CDK 在 AWS 帳戶中部署資源，請執行下列動作： 1. (選擇性) 在複製存放庫的根目錄中，於`app.py`檔案中更新 AWS CloudFormation 堆疊名稱。預設堆疊名稱為`genai-knowledge-capture-stack` 。 2. 若要部署資源，請執行命令`cdk deploy`。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	此命`cdk deploy`令使用第 3 層建構來建立一組 Lambda 函數、S3 儲存貯體、Amazon SNS 主題和 Step Functions 狀態機器。在部署期間，資`assets/audio_samples` 料夾中的音訊檔案會複製到 S3 儲存貯體。 3. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/cloudformation/ 開啟 CloudFormation 主控台。 4. 確認堆疊已成功部署。如需指示，請參閱在 AWS CloudFormation 主控台上檢閱您的堆疊。	

任務	描述	所需技能
訂閱 Amazon SNS 主題。	若要訂閱 Amazon SNS 主題以取得通知，請執行下列動作： 1. 在 CloudFormation 主控台的導覽窗格中，選擇 [堆疊]。 2. 選擇`genai-knowledge-capture-stack` 堆疊。 3. 選擇 Output (輸出) 索引標籤。 4. 使用金鑰尋找 Amazon SNS 主題名稱`SNSTopicName` 。 5. 按照訂閱電子郵件地址到 Amazon SNS 主題中的指示，設定要接收通知的電子郵件地址。	一般 AWS

測試解決方案

任務	描述	所需技能
運行狀態機。	1. 開啟「Step Functions」主控台。 2. 在 [狀態電腦] 頁面上，選擇 [genai-knowledge-capture-stack狀態機器]。 3. 選擇 Start execution (開始執行)。 4. (選擇性) 在「名稱」方塊中，輸入執行項目的名稱。	一般 AWS 應用程式開發人員

任務	描述	所需技能
	5. 在「輸入」區域中，透過取代預留位置文字來輸入下列 JSON 物件，其中： • <Name>是您想要命名文檔的名稱。 • <S3 bucket name>是包含音訊檔案的 Amazon S3 儲存貯體的名稱。 • <Folder path>是包含音訊檔案的目錄。 { "documentName": "<Name>", "audioFileFolderUri": "s3://<S3 bucket name>/<Folder path>" } 6. 選擇 Start Execution (開始執行)。 7. 在執行詳細資訊頁面上，檢閱結果並等待執行完成。	

清理解決方案中的所有 AWS 資源

任務	描述	所需技能
移除 AWS 資源。	測試解決方案之後，請清理資源： 1. 刪除 S3 儲存貯體中的所有物件，然後刪除儲存貯體。如需詳細資訊，請參閱刪除值區。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	2. 從複製的儲存庫中，執行命令`cdk destroy`。	

相關資源

AWS 文件

- Amazon 基岩資源：
  - 模型存取
  - 基礎模型的推論參數
- AWS CDK 資源：
  - 開始使用 AWS CDK
  - 在 Python 中使用 AWS CDK
  - 疑難排解常見的 AWS CDK 問題
  - 工具包命令
- AWS Step Functions 資源：
  - 開始使用 AWS Step Functions
  - 疑難排解
- 使用 Python 構建函數
- AWS 上的生成 AI 應用程式建置器

其他資源

- LangChain 文件

使用 Amazon Personalize 個人化產生個人化和重新排名的建議

由梅森·卡希爾（AWS），馬修·沙塞（AWS）和塔約奧拉吉德（AWS）創建

代碼存儲庫：personalize-pet-recommendations	環境：PoC 或試點	技術：機器學習和 AI；雲端原生；基礎架構 DevOps；無伺服器
工作負載：開源	AWS 服務：AWS CloudFormation；Amazon Kinesis Data Firehose；AWS Lambda；Amazon Personalize；AWS Step Functions 數	

Summary

此模式說明如何使用 Amazon Personalize 根據從這些使用者擷取的即時使用者互動資料，為您的使用者產生個人化的建議 (包括重新排名的建議)。此模式中使用的示例場景基於寵物收養網站，該網站根據用戶的互動（例如，用戶訪問的寵物）為其用戶生成建議。透過遵循範例案例，您將學習如何使用 Amazon Kinesis Data Streams 擷取互動資料，AWS Lambda 產生建議並重新排名建議，以及使用 Amazon 資料 Firehose 將資料存放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。您也會學習如何使用 AWS Step Functions 建立可管理解決方案版本 (也就是訓練有素的模型) 的狀態機器，以產生您的建議。

先決條件和限制

先決條件

- 具有已啟動載入的 AWS Cloud Development Kit (AWS CDK) 的有效 AWS 帳戶
- 具有已設定登入資料的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- Python 3.9

產品版本

- Python 3.9

- AWS CDK 2.23.0 或更新版本
- AWS CLI 2.7.27 或更新版本

架構

技術堆疊

- Amazon 數據 Firehose
- Amazon Kinesis Data Streams
- Amazon Personalize
- Amazon Simple Storage Service (Amazon S3)
- AWS Cloud Development Kit (AWS CDK)
- AWS 命令列界面 (AWS CLI)
- AWS Lambda
- AWS Step Functions

目標架構

下圖說明將即時資料導入 Amazon Personalize 的管道。然後，管道會使用該資料為使用者產生個人化和重新排名的建議。

該圖顯示以下工作流程：

1. Kinesis Data Streams 會擷取即時使用者資料 (例如，造訪過的寵物等事件)，以供 Lambda 和 Firehose 處理。
2. Lambda 函數會處理 Kinesis Data Streams 中的記錄，並進行 API 呼叫，將記錄中的使用者互動新增至 Amazon Personalize 中的事件追蹤器。
3. 以時間為基礎的規則會叫用 Step Functions 狀態機器，並使用 Amazon Personalize 中事件追蹤器中的事件，為建議和重新排名模型產生新的解決方案版本。
4. 狀態機器會更新 Amazon Personalize 行銷活動，以使用新的解決方案版本。
5. Lambda 透過呼叫 Amazon Personalize 重新排名行銷活動，重新排名推薦項目清單。
6. Lambda 會呼叫 Amazon Personalize 建議行銷活動，擷取建議項目的清單。
7. Firehose 會將事件儲存到 S3 儲存貯體，在該儲存貯體中可做為歷史資料存取。

工具

AWS 工具

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon 資料 Firehose 可協助您將即時串流資料交付到其他 AWS 服務、自訂 HTTP 端點和受支援的第三方服務供應商擁有的 HTTP 端點。
- Amazon Kinesis Data Streams 可協助您即時收集和處理大型資料串流。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Personalize 是全受管的機器學習 (ML) 服務，可協助您根據資料為使用者產生項目建議。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 Lambda 函數和其他 AWS 服務，以建立關鍵業務應用程式。

其他工具

- pytest 是一個用於編寫小型可讀測試的 Python 框架。
- Python 是一種通用的計算機編程語言。

Code

此模式的代碼可在 GitHub 動物推薦程序存儲庫中找到。您可以使用此儲存庫中的 AWS CloudFormation 範本部署範例解決方案的資源。

注意：Amazon Personalize 解決方案版本、事件追蹤器和行銷活動由可擴充原生 CloudFormation 資源的自訂資源 (在基礎設施內) 提供支援。

史诗

建立基礎結構

任務	描述	所需技能
創建一個獨立的 Python 環境。	Mac 電腦設定	DevOps 工程師

任務	描述	所需技能
	1. 若要手動建立虛擬環境，請從終端機執行`$ python3 -m venv .venv`指令。 2. 初始化過程完成後，運行`$ source .venv/bin/activate` 命令以激活虛擬環境。 視窗設定 若要手動建立虛擬環境，請從終端機執行`% .venv\Scripts\activate.bat` 指令。	

任務	描述	所需技能
合成 CloudFormation 模板。	1. 要安裝所需的依賴關係，請從終端運行`$ pip install -r requirements.txt` 命令。 2. 在 AWS CLI 中，設定下列環境變數： • `export ACCOUNT_ID=123456789` • `export CDK_DEPLOY_REGION=us-east-1` • `export CDK_ENVIRONMENT=dev` 3. 在`config/{env}.yml` 檔案中，更新`vpcId`以符合您的虛擬私有雲 (VPC) ID。 4. 若要合成此程式碼的 CloudFormation 範本，請執行命`$ cdk synth`令。 注意：在步驟 2 中，`CDK_ENVIRONMENT` 指的是`config/{env}.yml` 文件。	DevOps 工程師

任務	描述	所需技能
部署資源並建立基礎結構。	若要部署解決方案資源，請從終端機執行`./deploy.sh` 命令。 這個命令會安裝所需的 Python 相依性。Python 指令碼會建立 S3 儲存貯體和 AWS Key Management Service (AWS KMS) 金鑰，然後新增初始模型建立的種子資料。最後，指令碼會執行`cdk deploy`以建立剩餘的基礎結構。 注意：初始模型訓練會在堆疊建立期間進行。堆疊最多可能需要兩個小時才能完成建立。	DevOps 工程師

相關資源

- 動物推薦 () GitHub
- AWS CDK 參考文件
- 肉毒桿菌 3 文件
- 使用 Amazon 個人化 (AWS Machine Learning 部落格) 針對您選擇的商業指標優化個人化建議

其他資訊

有效載荷和回應範例

推薦 Lambda 函數

若要擷取建議，請使用下列格式的承載向建議 Lambda 函數提交請求：

```
{
  "userId": "3578196281679609099",
  "limit": 6
```

```
}
```

下列範例回應包含動物群組清單：

```
[{"id": "1-domestic short hair-1-1"},
{"id": "1-domestic short hair-3-3"},
{"id": "1-domestic short hair-3-2"},
{"id": "1-domestic short hair-1-2"},
{"id": "1-domestic short hair-3-1"},
{"id": "2-beagle-3-3"},
```

如果您省略userId欄位，函式會傳回一般建議。

重新排名 Lambda 函數

若要使用重新排名，請向重新排名的 Lambda 函數提交請求。裝載包含要重新排名userId的所有項目 ID 及其中繼資料。下面的示例數據使用牛津寵物類animal_species_id（1 = 貓，2 = 狗）和整數 1-5 用於和：animal_age_idanimal_size_id

```
{
   "userId":"12345",
   "itemMetadataList":[
      {
         "itemId":"1",
         "animalMetadata":{
            "animal_species_id":"2",
            "animal_primary_breed_id":"Saint_Bernard",
            "animal_size_id":"3",
            "animal_age_id":"2"
         }
      },
      {
         "itemId":"2",
         "animalMetadata":{
            "animal_species_id":"1",
            "animal_primary_breed_id":"Egyptian_Mau",
            "animal_size_id":"1",
            "animal_age_id":"1"
         }
      },
      {
         "itemId":"3",
```

```
         "animalMetadata":{
            "animal_species_id":"2",
            "animal_primary_breed_id":"Saint_Bernard",
            "animal_size_id":"3",
            "animal_age_id":"2"
         }
      }
   ]
}
```

Lambda 函數會重新排列這些項目，然後傳回包含項目 ID 和 Amazon Personalize 直接回應的訂購清單。這是物品所在的動物群體和分數的排名列表。Amazon Personalize 使用使用者個人化和個人化排名配方，在建議中包含每個項目的分數。這些分數代表 Amazon Personalize 對使用者接下來要選擇哪個項目的相對確定性。分數越高代表確定性越高。

```
{
   "ranking":[
      "1",
      "3",
      "2"
   ],
   "personalizeResponse":{
      "ResponseMetadata":{
         "RequestId":"a2ec0417-9dcd-4986-8341-a3b3d26cd694",
         "HTTPStatusCode":200,
         "HTTPHeaders":{
            "date":"Thu, 16 Jun 2022 22:23:33 GMT",
            "content-type":"application/json",
            "content-length":"243",
            "connection":"keep-alive",
            "x-amzn-requestid":"a2ec0417-9dcd-4986-8341-a3b3d26cd694"
         },
         "RetryAttempts":0
      },
      "personalizedRanking":[
         {
            "itemId":"2-Saint_Bernard-3-2",
            "score":0.8947961
         },
         {
            "itemId":"1-Siamese-1-1",
            "score":0.105204
         }
```

```
      ],
      "recommendationId":"RID-d97c7a87-bd4e-47b5-a89b-ac1d19386aec"
   }
}
```

Amazon Kinesis

傳送至 Amazon Kinesis 的承載具有下列格式：

```
{
    "Partitionkey": "randomstring",
    "Data": {
        "userId": "12345",
        "sessionId": "sessionId4545454",
        "eventType": "DetailView",
        "animalMetadata": {
            "animal_species_id": "1",
            "animal_primary_breed_id": "Russian_Blue",
            "animal_size_id": "1",
            "animal_age_id": "2"
        },
        "animal_id": "98765"

    }
}
```

附註：系統會針對未驗證的使用者移除此`userId`欄位。

在 Amazon 上訓練和部署支援 GPU 的自訂機器學習模型 SageMaker

環境：PoC 或試點	技術：機器學習與人工智慧；容器與微服務	AWS 服務：Amazon ECS; Amazon SageMaker

Summary

訓練和部署支援圖形處理單元 (GPU) 的機器學習 (ML) 模型需要初始設定和初始化特定環境變數，才能完全發揮 NVIDIA GPU 的優點。但是，設置環境並使其與 Amazon 網絡服務（AWS）雲上的亞馬遜 SageMaker 架構兼容可能非常耗時。

此模式可協助您使用 Amazon 訓練和建置支援 GPU 的自訂機器學習模型。 SageMaker它提供了訓練和部署在開放原始碼 Amazon 評論資料集上建置的自訂 CatBoost 模型的步驟。然後，您可以在`p3.16xlarge`亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上對其效能進行基準測試。

如果您的組織想要在上部署現有 GPU 支援的 ML 模型，則此模式非常有用。 SageMaker您的資料科學家可以依照此模式中的步驟建立 NVIDIA GPU 支援的容器，並在這些容器上部署機器學習模型。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- Amazon Simple Storage Service (Amazon S3) 來源儲存貯體，用於存放模型成品和預測。
- 了解 SageMaker 筆記本實例和 Jupyter 筆記本。
- 瞭解如何建立具有基本 SageMaker 角色許可、S3 儲存貯體存取和更新許可，以及 Amazon 彈性容器登錄 (Amazon ECR) 的其他許可的 AWS Identity and Access Management (IAM) 角色。

限制

- 此模式適用於使用以 Python 撰寫的訓練和部署程式碼的受監督 ML 工作負載。

架構

技術, 堆

- SageMaker
- Amazon ECR

工具

工具

- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種 AWS 受管容器映像登錄服務，安全、可擴展且可靠。
- Amazon SageMaker — SageMaker 是一個全受管的 ML 服務。
- 碼頭工人 — D ocker 是用於快速構建，測試和部署應用程序的軟件平台。
- Python-Python 是一種編程語言。

Code

此模式的程式碼可在使用 Catboost 和 SageMaker存放庫 GitHub 實作檢閱分類模型中找到。

史诗

準備資料

任務	描述	所需技能
建立 IAM 角色並附加必要政策。	登入 AWS 管理主控台、開啟 IAM 主控台，然後建立新的 IAM 角色。將下列內嵌政策連接到角色： • `AmazonEC2ContainerRegistryFullAccess` • `AmazonS3FullAccess` • `AmazonSageMakerFullAccess`	資料科學家

任務	描述	所需技能
	如需有關此項目的詳細資訊，請參閱 Amazon SageMaker 文件中的建立筆記本執行個體。	
建立 SageMaker 筆記本執行個體。	開啟 SageMaker 主控台，選擇 [記事本執行個體]，然後選擇 [建立筆記本執行個 對於 IAM 角色，請選擇您先前建立的 IAM 角色。根據需求設定筆記本執行個體，然後選擇 [建立筆記本執行個體]。 如需詳細步驟和指示，請參閱 Amazon SageMaker 文件中的建立筆記本執行個體。	資料科學家
複製儲存庫。	在 SageMaker 筆記本執行個體中開啟終端機，然後執行下列命令，使用 Catboost 和 SageMaker存放庫複製 GitHub 實作檢閱分類模型： `git clone https://github.com/aws-samples/review-classification-using-catboost-sagemaker.git`	
啟動 Jupyter 筆記本。	啟動 `Review classification model with Catboost and SageMaker.ipynb` Jupyter 筆記本，其中包含預先定義的步驟。	資料科學家

特徵工程

任務	描述	所需技能
在 Jupyter 筆記本中執行命令。	開啟 Jupyter 筆記本並執行下列故事中的命令，準備資料以訓練 ML 模型。	資料科學家
從 S3 儲存貯體讀取資料。	`import pandas as pd` `import csv` `fname = 's3://ama` `zon-reviews-pds/ts` `v/amazon_reviews_u` `s_Digital_Video_Do` `wnload_v1_00.tsv.gz'` `df = pd.read_csv(fname,` `sep='\t',delimiter` `='\t',error_bad_li` `nes=False)`	資料科學家
預先處理資料。	`import numpy as np` `def pre_process(df):` `  df.fillna(value={'` `review_body': '',` `'review_headline':` `''}, inplace=True)` `df.fillna( value={'v` `erified_purchase':` `'Unk'}, inplace=True)` `df.fillna(0,` `inplace=True)` `return df` `df = pre_process(df)` `df.review_date =` `pd.to_datetime(df.` `review_date)` `df['target'] =` `np.where(df['star_` `rating']>=4,1,0)`	資料科學家

任務	描述	所需技能
	附註：此程式碼會以空白字串取代中'review_body' 的 null 值'Unk'，並將資'verified_purchase' 料行取代為 (表示「unknown」)。	

任務	描述	所需技能
將資料分割為訓練、驗證和測試資料集。	為了使目標標籤在分割集中的分佈保持相同，您必須使用 scikit-learn 庫對取樣進行分層。 `from sklearn.model_selection import StratifiedShuffleSplit` `sss = StratifiedShuffleSplit(n_splits=2, test_size=0.10, random_state=0)` `sss.get_n_splits(df, df['target'])` `for train_index, test_index in sss.split(df, df['target']):` `X_train_vallid , X_test = df.iloc[train_index], df.iloc[test_index]` `sss.get_n_splits(X_train_vallid, X_train_vallid['target'])` `for train_index, test_index in sss.split(X_train_vallid, X_train_vallid['target']):` `X_train , X_valid = X_train_vallid.iloc[train_index], X_train_vallid.iloc[test_index]`	資料科學家

建置、執行 Docker 映像檔並將其推送至 Amazon ECR

任務	描述	所需技能
準備並推送泊塢視窗映像。	在 Jupyter 筆記本中，執行下列故事中的命令以準備 Docker 映像並將其推送至 Amazon ECR。	ML 工程師
在 Amazon ECR 中創建一個存儲庫。	`%%sh` `algorithm_name=s agemaker-catboost- github-gpu-img` `chmod +x code/train chmod +x code/serve` `account=$(aws sts get- caller-identity -- query Account --output text)` `# Get the region defined in the current configuration (default to us-west-2 if none defined) region=$(aws configure get region) region=${region:-us- east-1}` `fullname="${accou nt}.dkr.ecr.${regi on}.amazonaws.com/ ${algorithm_name}: latest"` `aws ecr create-re pository --repository-`	ML 工程師

任務	描述	所需技能
	`name "${algorithm_name} " > /dev/nul`	
在本地構建碼頭映像。	`docker build -t "${algorithm_name}" . docker tag ${algorithm_name} ${fullname}`	ML 工程師
運行碼頭映像並將其推送到 Amazon ECR。	`docker push ${fullname}`	ML 工程師

培訓

任務	描述	所需技能
建立 SageMaker 超參數調整工作。	在 Jupyter 筆記本中，執行下列故事中的命令，以使用 Docker 映像建立 SageMaker 超參數調整工作。	資料科學家
創建一個 SageMaker 估計器。	使用 Docker 映像檔的名稱建立SageMaker 估算器。 `import sagemaker as sage from time import gmtime, strftime sess = sage.Session() from sagemaker.tuner import IntegerParameter, CategoricalParameter, ContinuousParameter, HyperparameterTuner  account = sess.boto_session.client('s`	資料科學家

任務	描述	所需技能
	ts').get_caller_id entity()['Account'] region = sess.boto _session.region_name image = '{}.dkr.e cr.{}.amazonaws.co m/sagemaker-catboo st-github-gpu-img: latest'.format(acc ount, region) tree_hpo = sage.esti mator.Estimator(im age, role, 1, 'ml.p3.16xlarge', train_volume_size = 100, output_path="s3:// {}/sagemaker/DEMO- GPU-Catboost/outpu t".format(bucket), sagemaker_session= sess)	

任務	描述	所需技能
建立 HPO 工作。	使用參數範圍建立超參數最佳化 (HPO) 調整工作，並將列車和驗證集作為參數傳遞至函數。 `hyperparameter_ranges = {'iterations': IntegerParameter(80000, 130000),` `'max_depth': IntegerParameter(6, 10),` `'max_ctr_complexity': IntegerParameter(4, 10),` `'learning_rate': ContinuousParameter(0.01, 0.5)}` `objective_metric_name = 'auc'` `metric_definitions = [{'Name': 'auc',` `'Regex': 'auc: ([0-9\.]+)'}]` `tuner = HyperparameterTuner(tree_hpo,` `objective_metric_name,` `hyperparameter_ranges,` `metric_definitions,`	資料科學家

任務	描述	所需技能
	`objective_type='Maximize',` `max_jobs=50,` `max_parallel_jobs=2)`	
執行 HPO 工作。	`train_location = 's3://'+bucket+'/sagemaker/DEMO-GPU-Catboost/data/train/'` `valid_location = 's3://'+bucket+'/sagemaker/DEMO-GPU-Catboost/data/valid/'` `tuner.fit({'train': train_location, 'validation': valid_location })`	資料科學家
獲得表現最佳的培訓工作。	`import sagemaker as sage` `from time import gmtime, strftime` `sess = sage.Session()` `best_job =tuner.best_training_job()`	資料科學家

批次轉換

任務	描述	所需技能
在測試資料上建立 SageMaker 批次轉換工作以進行模型預測。	在 Jupyter 筆記本中，執行下列故事中的命令，從 SageMaker 超參數調整工作建立模型，並在測試資料上提交 SageMaker 批次轉換工作以進行模型預測。	資料科學家
建立 SageMaker 模型。	使用最好的訓練工作在 SageMaker 模型中建立模型。 `attached_estimator = sage.estimator.Estimator.attach(best_job)` `output_path ='s3://'+bucket+'/sagemaker/DEMO-GPU-Catboost/data/test-predictions/'` `input_path ='s3://'+bucket+'/sagemaker/DEMO-GPU-Catboost/data/test/'` `transformer = attached_estimator.transformer(instance_count=1,` `instance_type='ml.p3.16xlarge',` `assemble_with='Line',`	資料科學家

任務	描述	所需技能
	accept='text/csv', max_payload=1, output_path=output_path, env = {'SAGEMAKER_MODEL_SERVER_TIMEOUT' : '3600' })	
建立批次轉換工作。	在測試資料集上建立批次轉換工作。 transformer.transform(input_path, content_type='text/csv', split_type='Line')	資料科學家

分析結果

任務	描述	所需技能
閱讀結果並評估模型的效能。	在 Jupyter 筆記本中，執行下列故事中的命令，以讀取結果，並評估「中華民國曲線下區域」(ROC-AUC) 和「精確回復曲線下的區域」(PR-AUC) 模型量度上的模型效能。	資料科學家

任務	描述	所需技能
	如需相關資訊，請參閱 Amazon Machine Learning (Amazon ML) 文件中的 Amazon 機器學習關鍵概念。	
讀取批次轉換工作結果。	將批次轉換工作結果讀取為資料框。 `file_name = 's3://'+bucket+'/sagemaker/DEMO-GPU-Catboost/data/test-predictions/file_1.out'` `results = pd.read_csv(file_name, names=['review_id','target','score'] ,sep='\t',escapechar ='\\' , quoting=csv.QUOTE_NONE,` `lineterminator='\n',quotechar='"').dropna()`	資料科學家

任務	描述	所需技能
評估效能指標。	在 ROC-AUC 和 PR-AUC 上評估模型的效能。 `from sklearn import metrics` `import matplotlib` `import pandas as pd` `matplotlib.use('agg', warn=False, force=True)` `from matplotlib import pyplot as plt` `%matplotlib inline` `def analyze_results(labels, predictions):` `precision, recall, thresholds = metrics.precision_recall_curve(labels, predictions)` `auc = metrics.auc(recall, precision)` `fpr, tpr, _ = metrics.roc_curve(labels, predictions)` `roc_auc_score = metrics.roc_auc_score(labels, predictions)` `print('Neural-Nets: ROC auc=%.3f' % ( roc_auc_score))` `plt.plot(fpr, tpr, label="data 1, auc=" + str(roc_auc_score))`	資料科學家

任務	描述	所需技能
	<pre> plt.xlabel('1-Specificity') plt.ylabel('Sensitivity') plt.legend(loc=4) plt.show() lr_precision, lr_recall, _ = metrics.precision_recall_curve(labels, predictions) lr_auc = metrics.auc(lr_recall, lr_precision) # summarize scores print('Neural-Nets: PR auc=%.3f' % (lr_auc)) # plot the precision-recall curves no_skill = len(labels[labels==1.0]) / len(labels) plt.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill') plt.plot(lr_recall, lr_precision, marker='.', label='Neural-Nets') # axis labels plt.xlabel('Recall') plt.ylabel('Precision') # show the legend plt.legend() # show the plot</pre>	

任務	描述	所需技能
	plt.show() return auc analyze_results(results['target'].values,results['score'].values)	

相關資源

- SageMaker 通過構建 Scikit 碼頭集裝箱在 Amazon 培訓和託管 Scikit 學習模型

其他資訊

下面的列表顯示了在構建，運行，並推送 Docker 映像到 Amazon ECR 史詩 Docer 文件的不同元素。

使用 AWS-安裝 Python。

```
FROM amazonlinux:1

 RUN yum update -y && yum install -y python36 python36-devel python36-libs python36-
tools python36-pip && \
 yum install gcc tar make wget util-linux kmod man sudo git -y && \
 yum install wget -y && \
 yum install aws-cli -y && \
 yum install nginx -y && \
yum install gcc-c++.noarch -y && yum clean all
```

安裝 Python 件

```
RUN pip-3.6 install --no-cache-dir --upgrade pip && \pip3 install --no-cache-dir --
upgrade setuptools && \
pip3 install Cython && \
```

```
pip3 install --no-cache-dir numpy==1.16.0 scipy==1.4.1 scikit-learn==0.20.3
 pandas==0.24.2 \
flask gevent gunicorn boto3 s3fs  matplotlib joblib catboost==0.20.2
```

安裝 CUDA 和 CuDNN

```
RUN wget https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/
cuda_9.0.176_384.81_linux-run \
&& chmod u+x cuda_9.0.176_384.81_linux-run \
&& ./cuda_9.0.176_384.81_linux-run --tmpdir=/data --silent --toolkit --override \
&& wget https://custom-gpu-sagemaker-image.s3.amazonaws.com/installation/cudnn-9.0-
linux-x64-v7.tgz \
&& tar -xvzf cudnn-9.0-linux-x64-v7.tgz \
&& cp /data/cuda/include/cudnn.h /usr/local/cuda/include \
&& cp /data/cuda/lib64/libcudnn* /usr/local/cuda/lib64 \

&& chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn* \
&&  rm -rf /data/*
```

建立所需的目錄結構 SageMaker

```
RUN mkdir /opt/ml /opt/ml/input /opt/ml/input/config /opt/ml/input/data /opt/ml/input/
data/training /opt/ml/model /opt/ml/output /opt/program
```

設定環境變數

```
ENV PYTHONPATH=/opt/program
ENV PYTHONUNBUFFERED=TRUE
ENV PYTHONDONTWRITEBYTECODE=TRUE
ENV PATH="/opt/program:${PATH}"

# Set NVIDIA mount environments
ENV LD_LIBRARY_PATH=/usr/local/nvidia/lib:/usr/local/nvidia/lib64:$LD_LIBRARY_PATH
ENV NVIDIA_VISIBLE_DEVICES="all"
ENV NVIDIA_DRIVER_CAPABILITIES="compute,utility"
ENV NVIDIA_REQUIRE_CUDA "cuda>=9.0"
```

將訓練和推論檔案複製到 Docker 映像檔

```
COPY code/* /opt/program/
WORKDIR /opt/program
```

針對 TB 級 SageMaker ML 資料集的分散式特徵工程使用處理

由克里斯·布姆豪爾（AWS）創建

環境：生產	技術：機器學習和人工智能; 大數據	AWS 服務：Amazon SageMaker

Summary

許多 TB 級或更大的資料集通常由階層式資料夾結構組成，而資料集中的檔案有時會共用相互依存性。因此，機器學習 (ML) 工程師和資料科學家必須做出周到的設計決策，準備這類資料以進行模型訓練和推論。此模式示範如何使用手動巨集分割和微分片技術，結合 Amazon SageMaker 處理和虛擬 CPU (vCPU) 平行化，為複雜的大數據 ML 資料集有效地擴展功能工程程序。

此模式將巨集分割定義為跨多台機器分割資料目錄以進行處理，而將每台機器上的資料分割為跨多個處理執行緒的資料進行微分片。該模式通過使用 Amazon SageMaker 與 PhysioNet M IMIC-III 數據集中的樣本時間序列波形記錄來示範這些技術。透過在此模式中實作技術，您可以將特徵工程的處理時間和成本降至最低，同時將資源使用率和輸送量效率最大化。無論資料類型為何，這些優化都仰賴 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的分散式 SageMaker 處理，以及針對類似大型資料集的 vCPUs。

先決條件和限制

先決條件

- 如果您想要為自己的資料集實作此模式，請存取 SageMaker 筆記本執行個體或 SageMaker Studio。如果您是第一 SageMaker 次使用 Amazon，請參閱 AWS 文件 SageMaker中的 Amazon 入門。
- SageMaker 工作室，如果你想用 PhysioNet MIMIC III 樣本數據來實現這種模式。
- 該模式使用 SageMaker 處理，但不需要任何運行 SageMaker 處理作業的經驗。

限制

- 此模式非常適合包含相互依存檔案的 ML 資料集。手動巨集分割和並行執行多個單一執 parallel 個體處理工作，這些相互依存性最大的好 SageMaker 處。對於不存在此類相依性的資料集，Procade 中的`ShardedByS3Key`功能可能是巨集分割的更好替代方案，因為它會將資料分割傳送至由相同

SageMaker 處理工作管理的多個執行個體。不過，您可以在這兩種情況下實作此模式的微分片策略，以充分利用執行個體 vCPUs。

產品版本

- Amazon SageMaker Python 開發套件第 2 版

架構

目標技術堆疊

- Amazon Simple Storage Service (Amazon S3)
- Amazon SageMaker

目標架構

宏分片和分散式 EC2 執行個體

此架構中所表示的 10 個 parallel 程序反映 MIMIC-III 資料集的結構。（流程由橢圓表示，用於簡化圖表。）當您使用手動巨集分割時，類似的架構也適用於任何資料集。對於 MIMIC-III，您可以通過單獨處理每個患者組文件夾，以最小的努力使用數據集的原始結構來發揮您的優勢。在下圖中，記錄群組區塊會顯示在左側 (1)。鑑於數據的分佈式性質，由患者組分片是有意義的。

但是，按患者組進行手動分片意味著每個患者組文件夾都需要單獨的處理任務，如圖（2）的中間部分所示，而不是具有多個 EC2 實例的單個處理任務。由於 MIMIC III 的資料包含二進位波形檔案和相符的文字標頭檔案，而且必須依賴 wfdb 程式庫進行二進位資料擷取，因此特定患者的所有記錄都必須在同一個執行個體上提供。確定每個二進位波形檔案的相關標頭檔案也存在的唯一方法是實作手動分片以在其自己的處理工作中執行每個碎片，並指定何`s3_data_distribution_type='FullyReplicated'`時定義處理工作輸入。或者，如果所有資料都在單一目錄中可用，且檔案之間沒有相依性，則更合適的選項可能是啟動具有多個 EC2 執行個體並`s3_data_distribution_type='ShardedByS3Key'`指定的單一處理任務。指定`ShardedByS3Key` 為 Amazon S3 資料分發類型，可指示 SageMaker 跨執行個體自動管理資料分片。

為每個資料夾啟動「處理」工作是預先處理資料的一種具成本效益的方式，因為同時執行多個執行個體可節省時間。為了節省額外的成本和時間，您可以在每個處理任務中使用微分片。

微分片和 parallel vCPUs

在每個處理任務中，分組的資料會進一步劃分，以充分利用 SageMaker 全受控 EC2 執行個體上所有可用的 vCPUs。圖表 (2) 中間部分的區塊描述了每個主要處理工作中發生的情況。病患記錄資料夾的內容會根據執行個體上可用的 vCPUs 數目平面化並平均分割。分割資料夾內容後，大小均勻的檔案集會分散到所有 vCPUs 以進行處理。處理完成後，每個 vCPU 的結果會合併為每個「處理」工作的單一資料檔案。

在附加的代碼中，這些概念在`src/feature-engineering-pass1/preprocessing.py`文件的以下部分中表示。

```
def chunks(lst, n):
    """
    Yield successive n-sized chunks from lst.

    :param lst: list of elements to be divided
    :param n: number of elements per chunk
    :type lst: list
    :type n: int
    :return: generator comprising evenly sized chunks
    :rtype: class 'generator'
    """
    for i in range(0, len(lst), n):
        yield lst[i:i + n]


# Generate list of data files on machine
data_dir = input_dir
d_subs = next(os.walk(os.path.join(data_dir, '.')))[1]
file_list = []
for ds in d_subs:
    file_list.extend(os.listdir(os.path.join(data_dir, ds, '.')))
dat_list = [os.path.join(re.split('_|\.', f)[0].replace('n', ''), f[:-4]) for f in
 file_list if f[-4:] == '.dat']

# Split list of files into sub-lists
cpu_count = multiprocessing.cpu_count()
splits = int(len(dat_list) / cpu_count)
if splits == 0: splits = 1
dat_chunks = list(chunks(dat_list, splits))

# Parallelize processing of sub-lists across CPUs
```

```
ws_df_list = Parallel(n_jobs=-1, verbose=0)(delayed(run_process)(dc) for dc in
 dat_chunks)

# Compile and pickle patient group dataframe
ws_df_group = pd.concat(ws_df_list)
ws_df_group = ws_df_group.reset_index().rename(columns={'index': 'signal'})
ws_df_group.to_json(os.path.join(output_dir, group_data_out))
```

函數chunks，首先定義為通過將其分成均勻大小的長度塊n 並將這些結果作為生成器返回來消耗給定列表。接下來，通過編譯存在的所有二進制波形文件的列表，將數據扁平化到患者文件夾中。完成此操作後，將獲得 EC2 執行個體上可用的 vCPUs 數量。透過呼叫chunks，將二進位波形檔案清單平均分為這些 vCPUs，然後使用 jobli b 的 Parallel 類別，在自己的 vCPU 上處理每個波形子清單。處理任務會自動將結果合併為單一資料框清單， SageMaker 然後在任務完成時將結果寫入 Amazon S3 之前進行進一步處理。在此範例中，處理任務將 10 個檔案寫入 Amazon S3 (每個任務一個)。

完成所有初始處理任務後，第二個處理任務 (顯示在圖表右側的區塊中) 會結合每個主要處理任務產生的輸出檔案，並將合併的輸出寫入 Amazon S3 (4)。

工具

工具

- Python-用於此模式的示例代碼是 Python（版本 3）。
- SageMaker Studio — Amazon SageMaker Studio 是適用於機器學習的網頁型整合式開發環境 (IDE)，可讓您建置、訓練、偵錯、部署和監控機器學習模型。您可以在 Studio 內 SageMaker 使用 Jupyter 筆記本來執行 SageMaker 處理工作。
- SageMaker 處理 — Amazon SageMaker 處理提供了一種簡化的方式來執行資料處理工作負載。在這種模式中，特徵工程代碼是通過使用 SageMaker 處理作業大規模實現的。

Code

附加的 .zip 檔案提供此模式的完整程式碼。下節說明為此模式建置架構的步驟。每個步驟都以附件中的範例程式碼說明。

史诗

設定您的 SageMaker 工作室環境

任務	描述	所需技能
訪問 Amazon SageMaker 工作室。	按照 Amazon SageMaker 文件中提供的指示，在您的 AWS 帳戶上線進入 SageMaker 工作室。	資料科學家、ML 工程師
安裝 wget 公用程式。	如果您使用新的 SageMaker Studio 配置登錄，或者您以前從未在 SageMaker Studio 中使用過這些實用程序，請安裝 w get。 若要安裝，請在 SageMaker Studio 主控台中開啟終端機視窗，然後執行下列命令： `sudo yum install wget`	資料科學家、ML 工程師
下載並解壓縮範例程式碼。	在「附件」區段中下載`attachments.zip` 檔案。在終端機視窗中，導覽至下載檔案的資料夾並解壓縮其內容： `unzip attachment.zip` 導覽至您解壓縮 .zip 檔案的資料夾，然後解壓縮`Scaled-Processing.zip` 檔案的內容。	資料科學家、ML 工程師

任務	描述	所需技能
	`unzip Scaled-Processing.zip`	
從中下載範例資料集，然後將其上傳到 Amazon S3。	在包含檔案的資料夾中執行 `get_data.ipynb` Jupyter 筆記本。`Scaled-Processing` 此筆記本會從 physionet.org 下載一個範例 MIMIC III 資料集，並將其上傳到 Amazon S3 中的 SageMaker 工作室工作階段儲存貯體。	資料科學家、ML 工程師

設定第一個預先處理指令碼

任務	描述	所需技能
平面化所有子目錄的檔案階層。	在大型資料集 (例如 MIMIC-III) 中，檔案通常會分散在多個子目錄中，即使在邏輯父群組中也是如此。您的指令碼應設定為展平所有子目錄中的所有群組檔案，如下列程式碼所示。 `# Generate list of .dat files on machine` `data_dir = input_dir` `d_subs = next(os.walk(os.path.join(data_dir, '.')))[1]` `file_list = []` `for ds in d_subs:` `  file_list.extend(os.listdir(os.path.join(data_dir, ds, '.')))`	資料科學家、ML 工程師

任務	描述	所需技能
	`dat_list = [os.path.join(re.split('_|\.', f)[0].replace('n', ''), f[:-4]) for f in file_list if f[-4:] == '.dat']` 注意此 Epic 中的範例程式碼片段來自附件中提供的`src/feature-engineering-pass1/preprocessing.py` 檔案。	
根據 vCPU 計數將檔案分割為子群組。	視執行指令碼的執行個體上存在的 vCPUs 數目而定，檔案應分為大小均勻的子群組或區塊。在此步驟中，您可以實作類似下列的程式碼。 `# Split list of files into sub-lists` `cpu_count = multiprocessing.cpu_count()` `splits = int(len(dat_list) / cpu_count)` `if splits == 0: splits = 1` `dat_chunks = list(chunks(dat_list, splits))`	資料科學家、ML 工程師

任務	描述	所需技能
平行處理跨 vCPUs 的子群組。	指令碼邏輯應該設定為 parallel 處理所有子群組。要做到這一點，使用 Joblib 庫的`Parallel` 類和`delayed` 方法如下。 `# Parallelize processing of sub-lists across CPUs` `ws_df_list = Parallel(n_jobs=-1, verbose=0)(delayed(run_process)(dc) for dc in dat_chunks)`	資料科學家、ML 工程師
將單一檔案群組輸出儲存到 Amazon S3。	parallel vCPU 處理完成後，應合併每個 vCPU 的結果，並上傳至檔案群組的 S3 儲存貯體路徑。在此步驟中，您可以使用類似下列內容的程式碼。 `# Compile and pickle patient group dataframe` `ws_df_group = pd.concat(ws_df_list)` `ws_df_group = ws_df_group.reset_index().rename(columns={'index': 'signal'})` `ws_df_group.to_json(os.path.join(output_dir, group_data_out))`	資料科學家、ML 工程師

設定第二個預先處理指令碼

任務	描述	所需技能
合併執行第一個指令碼的所有處理工作所產生的資料檔案。	先前的指令碼會針對每個 SageMaker 處理資料集中檔案群組的處理工作輸出單一檔案。 接下來，您需要將這些輸出檔案合併為單一物件，然後將單一輸出資料集寫入 Amazon S3。這在文件中演示，該`src/feature-engineering-pass1p5/preprocessing.py` 文件在附件中提供，如下所示。 `def write_parquet(wavs_df, path):` `"""` `Write waveform summary dataframe to S3 in parquet format.` `:param wavs_df: waveform summary dataframe` `:param path: S3 directory prefix` `:type wavs_df: pandas dataframe` `:type path: str` `:return: None` `"""` `extra_args = {"ServerSideEncryption": "aws:kms"}` `wr.s3.to_parquet(` `df=wavs_df,` `path=path,` `compression='snappy',`	資料科學家、ML 工程師

任務	描述	所需技能
	<pre>s3_additi onal_kwargs=extra_ args) def combine_data(): """ Get combined data and write to parquet. :return: waveform summary dataframe :rtype: pandas dataframe """ wavs_df = get_data( ) wavs_df = normalize _signal_names(wavs _df) write_parquet(wavs _df, "s3://{}/{}/ {}".format(buck et_xform, dataset_p refix, pass1p5ou t_data)) return wavs_df wavs_df = combine_d ata()</pre>	

執行處理工作

任務	描述	所需技能
執行第一個處理工作。	若要執行巨集分割，請為每個檔案群組執行個別的處理工	資料科學家、ML 工程師

任務	描述	所需技能
	作。微分片會在每個處理工作內執行，因為每個工作都會執行您的第一個指令碼。下列程式碼會示範如何針對下列程式碼片段 (包含在中notebooks/FeatExtract_Pass1.ipynb) 中的每個檔案群組目錄啟動處理工作。 `pat_groups = list(range(30,40))` `ts = str(int(time.time()))` `for group in pat_groups:` `  sklearn_processor = SKLearnProcessor(framework_version='0.20.0',` `  role=role,` `  instance_type='ml.m5.4xlarge',` `  instance_count=1,` `  volume_size_in_gb=5)` `  sklearn_processor.run(` `  code='../src/feature-engineering-pass1/preprocessing.py',` `  job_name='-'.join(['scaled-`	

任務	描述	所需技能
	processing-p1', str(group), ts]), arguments=["input_path", "/opt/ml/processing/input", "output_path", "/opt/ml/processing/output", "group_data_out", "ws_df_group.json"], inputs= [ProcessingInput(source=f's3://{sess.default_bucket()}/data_inputs/{group}', destination='/opt/ml/processing/input', s3_data_distribution_type='FullyReplicated')], outputs= [ProcessingOutput(source='/opt/ml/processing/output', destination=f's3://{sess.default_bucket()}/data_outputs/{group}'	

任務	描述	所需技能
	)], wait=False)	

任務	描述	所需技能
執行第二個處理工作。	若要合併第一組處理工作所產生的輸出，並執行任何額外的預處理計算，請使用單 SageMaker 一處理工作來執行第二個指令碼。下面的代碼演示了這一點（包括在中notebooks/FeatExtract_Pass1p5.ipynb ）。 `ts = str(int(time.time()))` `bucket = sess.default_bucket()` `sklearn_processor = SKLearnProcessor(framework_version='0.20.0',` `role=role,` `instance_type='ml.t3.2xlarge',` `instance_count=1,` `volume_size_in_gb=5)` `sklearn_processor.run(` `code='../src/feature-engineering-pass1p5/preprocessing.py',` `job_name='-'.join(['scaled-processing', 'p1p5', ts]),` `arguments=['bucket', bucket,`	資料科學家、ML 工程師

任務	描述	所需技能
	`'pass1out_prefix', 'data_outputs', 'pass1out_data', 'ws_df_group.json', 'pass1p5out_data', 'waveform_summary.parquet', 'statsdata_name', 'signal_stats.csv'], wait=True )`	

相關資源

- 使用快速入門到 Amazon SageMaker 工作室（SageMaker 文檔）
- 處理資料 (SageMaker 文件)
- 數據處理與學習（文檔）SageMaker
- 工作平行文件
- 穆迪, B., 穆迪, G., 比利亞羅爾, M., 克利福德, G., & 席爾瓦, I. (2020). 模擬三波形數據庫（1.0 版）。PhysioNet。
- 約翰遜, A. E. W., 波拉德, T. J., 沈, L., 雷曼, L., 馮, M., 加塞米, M., 穆迪, B., 斯佐洛維茨, P., 塞利, L., & 馬克, R. G. (2016). MIMIC-III，一個可自由訪問的重症監護數據庫。「科學資料」，第三頁、一六零三五年。
- 模擬 III 波形資料庫授權

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用燒瓶和 AWS Elastic Beanstalk 將 AI/ML 模型結果視覺化

創建者：克里斯·科迪爾 (AWS) 和杜爾加蘇里

環境：PoC 或試點	技術：機器學習和人工智能; 分析 DevOps; Web 和移動應用	工作負載：開源
AWS 服務：Amazon Comprehend；AWS Elastic Beanstalk		

Summary

將人工智慧和機器學習 (AI/ML) 服務的輸出視覺化通常需要複雜的 API 呼叫，這些 API 呼叫必須由開發人員和工程師自訂。如果您的分析師想要快速探索新的資料集，這可能是一個缺點。

您可以使用 Web 式使用者介面 (UI) 來增強服務的可存取性，並提供更具互動性的資料分析形式，該介面可讓使用者上傳自己的資料並在儀表板中視覺化模型結果。

此模式使用 Flask 和 Plotly 將 Amazon Comprehend 與自訂 Web 應用程式整合，並從使用者提供的資料中視覺化情緒和實體。此模式也提供使用 AWS Elastic Beanstalk 部署應用程式的步驟。您可以使用 Amazon 網路服務 (AWS) AI 服務或在端點上託管的自訂訓練模型 (例如 Amazon SageMaker 端點) 來調整應用程式。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI)，在您的本機電腦上安裝和設定。如需這方面的詳細資訊，請參閱 AWS CLI 文件中的組態基礎知識。您也可以使用 AWS Cloud9 整合式開發環境 (IDE)；如需有關這方面的詳細資訊，請參閱 AWS Cloud9 文件中的 Python 教學和在 AWS Cloud9 IDE 中預覽執行中的應用程式。
- 瓶的 Web 應用程序框架的理解。如需有關 Flask 的詳細資訊，請參閱 Flask 文件中的快速入門導覽課程。
- 安裝和配置的 Python 3.6 或更高版本。您可以按照 AWS Elastic Beanstalk 文件中的設定 Python 開發環境中的指示來安裝 Python。

- 已安裝並設定 Elastic Beanstalk 命令列介面 (EB CLI)。如需有關這方面的詳細資訊，請參閱安裝 EB CLI 和 AWS Elastic Beanstalk 說明文件中的設定 EB CLI。

限制

- 此模式的 Flask 應用程式是設計用來處理使用單一文字資料行且限制為 200 列的 .csv 檔案。應用程序代碼可以調整以處理其他文件類型和數據分區。
- 該應用程序不會考慮數據保留，並繼續彙總上傳的用戶文件，直到它們被手動刪除。您可以將應用程式與 Amazon Simple Storage Service (Amazon S3) 整合以取得永久物件儲存，或使用 Amazon DynamoDB 等資料庫進行無伺服器金鑰值儲存。
- 該應用程序僅考慮英語文檔。不過，您可以使用 Amazon Comprehend 來偵測文件的主要語言。如需每個動作所支援語言的詳細資訊，請參閱 Amazon Comprehend 文件中的 API 參考資料。
- 其他資訊一節提供包含常見錯誤及其解決方案的疑難排解清單。

架構

燒瓶應用程式架構

瓶是在 Python 開發 Web 應用程序的輕量級框架。它旨在將 Python 強大的數據處理與豐富的 Web UI 相結合。此模式的 Flask 應用程式會示範如何建立可讓使用者上傳資料、將資料傳送至 Amazon Comprehend 進行推論的 Web 應用程式，然後將結果視覺化。 該應用程序具有以下結構：

- `static`— 包含支援 Web UI 的所有靜態檔案 (例如 JavaScript，CSS 和影像)
- `templates`-包含應用程序的所有 HTML 頁面
- `userData`— 儲存上傳的使用者資料
- `application.py`-瓶應用程序文件
- `comprehend_helper.py`— 使 API 調用到 Amazon Comprehend 函數
- `config.py`-應用程序配置文件
- `requirements.txt`— 應用程式所需的 Python 相依性

該`application.py`腳本包含 Web 應用程序的核心功能，它由四個瓶路由。下圖顯示了這些瓶路線。

- `/`是應用程式的根目錄，並將使用者引導至`upload.html`頁面 (儲存在`templates`目錄中)。

- `/saveFile`是用戶上傳文件後調用的路由。這條路由通過 HTML 表單，其中包含用戶上傳的文件接收`POST`請求。該文件被保存在目`userData`錄和路由重定向用戶的`/dashboard`路由。
- `/dashboard`將使用者傳送至`dashboard.html`頁面。在此頁面的 HTML 中，它會執行從`/data`路由讀取資料`static/js/core.js`的 JavaScript 程式碼，然後為頁面建立視覺效果。
- `/data`是一個 JSON API，用於在儀表板中顯示要視覺化的資料。此路由會讀取使用者提供的資料，並使用中的函數將使用者資料傳送`comprehend_helper.py`至 Amazon Comprehend 進行情緒分析和具名實體辨識 (NER)。Amazon 的響應被格式化並作為 JSON 對象返回。

部署架構

如需有關在 AWS 雲端使用 Elastic Beanstalk 部署之應用程式的設計考量的詳細資訊，請參閱 AWS Elastic Beanstalk 文件中的。

設計考量

技術堆疊

- Amazon Comprehend
- Elastic Beanstalk
- Flask

自動化和規模

Elastic Beanstalk 部署會使用負載平衡器和自動擴展群組 auto 動設定。如需更多組態選項，請參閱 AWS Elastic Beanstalk 文件中的設定彈性豆莖環境。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種統一的工具，可為與 AWS 的所有部分進行互動提供一致的界面。
- Amazon Comprehend 使用自然語言處理 (NLP) 擷取有關文件內容的見解，而不需要特殊的預先處理。
- AWS Elastic Beanstalk 可協助您在 AWS 雲端快速部署和管理應用程式，而不必了解執行這些應用程式的基礎設施。
- E@@ lastic Beanstalk CLI (EB CLI) 是 AWS Elastic Beanstalk 的命令列界面，提供互動式命令，可簡化從本機儲存庫建立、更新和監控環境的過程。

- 瓶框架使用 Python 執行數據處理和 API 調用，並提供與 Plotly 交互式網絡可視化。

Code

此模式的程式碼可在使用燒瓶和 AWS Elastic Beanstalk 儲存庫的 GitHub 視覺化 AI/ML 模型結果中取得。

史诗

設定燒瓶應用程式

任務	描述	所需技能
克隆存 GitHub 儲庫。	透過執行下列命令，使用 Flask 和 AWS Elastic Beanstalk 儲存庫，從 GitHub 視覺化 AI/ML 模型結果中提取應用程式程式碼： `git clone git@github.com:aws-samples/aws-comprehend-elasticbeanstalk-for-flask.git` 注意：請確定您使用 GitHub.	開發人員
安裝 Python 塊。	複製儲存庫之後，會建立新的本機`aws-comprehend-elasticbeanstalk-for-flask` 目錄。在該目錄中，該`requirements.txt` 文件包含運行應用程序的 Python 模塊和版本。使用以下命令來安裝模塊： `cd aws-comprehend-elasticbeanstalk-for-flask`	Python 開發者

任務	描述	所需技能
	`pip install -r requirements.txt`	
在本機測試應用程式。	通過運行以下命令啟動瓶服務器： `python application.py` 這會傳回執行中伺服器的相關資訊。您應該可以通過打開瀏覽器並訪問該應用程序 http://localhost:5000 注意：如果您在 AWS Cloud9 IDE 中執行應用程式，則需要使用下列行取代`application.py` 檔案中的`application.run()` 命令： `application.run(host=os.getenv('IP', '0.0.0.0'),port=int(os.getenv('PORT', 8080)))` 您必須在部署之前還原此變更。	Python 開發者

將應用程式部署到 Elastic Beanstalk

任務	描述	所需技能
啟動 Elastic Beanstalk 應用程式。	若要以 Elastic Beanstalk 應用程式的形式啟動專案，請從應用程式的根目錄執行下列命令：	建築師、開發者

任務	描述	所需技能
	`eb init -p python-3.6 comprehend_flask --region us-east-1` 重要： • `comprehend_flask` 是 Elastic Beanstalk 應用程序的名稱，可以根據您的要求進行更改。 • 您可以使用自己選擇的區域取代 AWS 區域。如果您未指定區域，則會使用 AWS CLI 中的預設區域。 • 該應用程序是使用 Python 3.6 版本構建的。如果您使用其他 Python 版本，可能會遇到錯誤。 執行命`eb init -i`令以取得更多部署組態選項。	
部署 Elastic Beanstalk 環境。	從應用程式的根目錄執行下列命令： `eb create comprehend-flask-env` 注意：`comprehend-flask-env` 是 Elastic Beanstalk 環境的名稱，可以根據您的要求進行更改。名稱只能包含字母、數字和破折號。	建築師、開發者

任務	描述	所需技能
授權您的部署以使用 Amazon Comprehend。	雖然您的應用程式可能已成功部署，但您也應該為您的部署提供對 Amazon Comprehend 的存取權。 `ComprehendFullAccess` 這是一項 AWS 受管政策，為部署的應用程式提供許可，以便對 Amazon Comprehend 進行 API 呼叫。 執行下列命令，將`ComprehendFullAccess` 政策附加到 `aws-elasticbeanstalk-ec2-role` (此角色是為您部署的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體自動建立的)： `aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/ComprehendFullAccess --role-name aws-elasticbeanstalk-ec2-role` 重要：`aws-elasticbeanstalk-ec2-role` 在您的應用程式部署時建立。您必須先完成部署程序，才能附加 AWS Identity and Access Management (IAM) 政策。	安全架構師開發人員

任務	描述	所需技能
造訪已部署的應用程式。	成功部署應用程式之後，您可以執行`eb open`命令來造訪應用程式。 您也可以執行命`eb status`令以接收有關部署的詳細資料。部署 URL 會列在下`CNAME`。	建築師、開發者

(選擇性) 根據您的 ML 模型自訂應用程式

任務	描述	所需技能
授權 Elastic Beanstalk 存取新模型。	請確定 Elastic Beanstalk 具有新模型端點所需的存取權限。例如，如果您使用 Amazon SageMaker 端點，則您的部署需要具有叫用端點的權限。 如需有關此項目的詳細資訊，請參閱 Amazon SageMaker 文件InvokeEndpoint中的。	安全架構師開發人員
將使用者資料傳送至新模型。	若要變更此應用程式中的基礎 ML 模型，您必須變更下列檔案： • `comprehend_helper.py` — 這是與 Amazon Comprehend 連接、處理回應並將最終結果傳回應用程式的 Python 指令碼。在此指令碼中，您可以將資料路由到 AWS 雲端上的其他 AI 服務，也可以將資料傳送到自訂模型端點。我們建議您也	資料科學家

任務	描述	所需技能
	格式化此指令碼中的結果，以便進行邏輯分離和此模式的可重複使用性。 • application.py — 如果您變更comprehend_helper.py 指令碼或函數的名稱，則需要更新應用程式application.py 指令碼以反映這些變更。	
更新儀表板視覺效果。	通常，合併新的 ML 模型意味著必須更新視覺效果以反映新結果。這些變更會在下列檔案中進行： • templates/dashboard.html — 預建的應用程式僅佔兩個基本視覺效果。頁面的整個佈局可以在此文件中進行調整。 • static/js/core.js — 此指令碼會擷取 Flask 伺服器/data路線的格式化輸出，並使用 Plotly 建立視覺效果。您可以新增或更新頁面的圖表。	网页开发者

(選擇性) 部署更新的應用程式

任務	描述	所需技能
更新應用程式的需求檔案。	在將變更傳送至 Elastic Beanstalk 之前，請在應用程式	Python 開發者

任務	描述	所需技能
	的根目錄中執行下列命令來更新`requirements.txt` 檔案以反映任何新的 Python 模組： `pip freeze > requirements.txt`	
重新部署 Elastic Beanstalk 環境。	若要確保您的應用程式變更會反映在 Elastic Beanstalk 部署中，請瀏覽至應用程式的根目錄並執行下列命令： `eb deploy` 這會將最新版本的應用程式程式碼傳送至您現有的 Elastic Beanstalk 部署。	系統管理員、建築師

相關資源

- 使用 Amazon API 閘道和 AWS Lambda 呼叫亞馬遜 SageMaker 模型端點
- 將燒瓶應用程序部署到 Elastic Beanstalk
- EB CLI 命令參考資料
- 設定您的 Python 發環境

其他資訊

故障排除列

以下是六個常見錯誤及其解決方案。

錯誤 1

```
Unable to assume role "arn:aws:iam::xxxxxxxxxx:role/aws-elasticbeanstalk-ec2-role".
 Verify that the role exists and is configured correctly.
```

解決方案：如果執行時發生此錯誤`eb create`，請在 Elastic Beanstalk 主控台上建立範例應用程式以建立預設執行個體設定檔。如需有關這方面的詳細資訊，請參閱 AWS Elastic Beanstalk 文件中的建立彈性豆莖環境。

錯誤二

```
Your WSGIPath refers to a file that does not exist.
```

解決方案：部署記錄檔中會發生此錯誤，因為 Elastic Beanstalk 預期會命名燒瓶程式碼。`application.py`如果您選擇不同的名稱，請執行`eb config`並編輯 WSGIPath，如下列程式碼範例所示：

```
aws:elasticbeanstalk:container:python:
     NumProcesses: '1'
     NumThreads: '15'
     StaticFiles: /static/=static/
     WSGIPath: application.py
```

確保用文件`application.py`名替換。

您還可以利用古尼科恩和處理檔。如需有關此方法的詳細資訊，請參閱 AWS Elastic Beanstalk 文件中的使用處理檔來設定 WSGI 伺服器。

錯誤三

```
Target WSGI script '/opt/python/current/app/application.py' does not contain WSGI
 application 'application'.
```

解決方案：Elastic Beanstalk 希望命名代表燒瓶應用程序的變量。`application`請確定`application.py`檔案使用做`application`為變數名稱：

```
application = Flask(__name__)
```

錯誤四

```
The EB CLI cannot find your SSH key file for keyname
```

解決方案：使用 EB CLI 指定要使用的金鑰對，或為部署的 EC2 執行個體建立 key pair。要解決錯誤，請運行`eb init -i`並且其中一個選項將詢問：

```
Do you want to set up SSH for your instances?
```

回應建立 key pair 或指定現有 key pair。Y

錯誤 5

我已經更新了我的代碼並重新部署，但我的部署沒有反映我的更改。

解決方案：如果您在部署中使用 Git 儲存庫，請務必在重新部署之前新增並提交變更。

錯誤 6

您正在從 AWS Cloud9 IDE 預覽燒瓶應用程式並遇到錯誤。

解決方案：如需有關此問題的詳細資訊，請參閱 AWS Cloud9 文件中的在 AWS Cloud9 IDE 中預覽執行中的應用程式。

使用 Amazon Comprehend 的自然語言處理

透過選擇使用 Amazon Comprehend，您可以執行即時分析或非同步批次任務，在個別文字文件中偵測自訂實體。Amazon Comprehend 還可讓您訓練可透過建立端點即時使用的自訂實體辨識和文字分類模型。

此模式使用非同步批次工作，從包含多個文件的輸入檔案中偵測情緒和實體。此模式提供的示例應用程序旨在為用戶上傳包含單列的 .csv 文件，每行一個文本文檔。使用燒瓶和 AWS Elastic Beanstalk 儲存庫將 GitHub 視覺化 AI/ML 模型中的`comprehend_helper.py`檔案讀取輸入檔案，並將輸入傳送至 Amazon Comprehend 進行處理。

BatchDetect實體

Amazon Comprehend 會檢查具名實體的批次文件文字，並傳回偵測到的實體、位置、實體類型，以及指示 Amazon Comprehend 信賴程度的分數。一次 API 呼叫最多可傳送 25 份文件，每份文件大小小小於 5,000 位元組。您可以根據使用案例篩選結果，以僅顯示某些實體。例如，您可以略過‘quantity’實體類型，並為偵測到的實體設定臨界值評分 (例如 0.75)。我們建議您先探索特定使用案例的結果，然後再選擇閾值。如需有關此項目的詳細資訊，請參閱 Amazon Comprehend 文件中的BatchDetect實體。

BatchDetect情緒

Amazon Comprehend 會檢查一批傳入的文件，並傳回每份文件 (POSITIVE、NEUTRAL、MIXED或) 的普遍情緒。NEGATIVE一次 API 呼叫最多可傳送 25 份文件，每份文件大小小小於 5,000 位元組。分析情緒非常簡單，您可以選擇要在最終結果中顯示得分最高的情緒。如需有關此項目的詳細資訊，請參閱 Amazon Comprehend 文件中的BatchDetect情緒。

瓶配置處理

Flask 服務器使用一系列配置變量來控制服務器的運行方式。這些變數可以包含除錯輸出、工作階段 Token 或其他應用程式設定。您也可以定義可在應用程式執行時存取的自訂變數。設定組態變數有多種方法。

在此模式中，組態會在中定義，config.py並在中繼承application.py。

- config.py包含在應用程式啟動時設定的組態變數。在這個應用程序中，一個DEBUG變量被定義為告訴應用程序在調試模式下運行服務器。注意：在生產環境中執行應用程式時，不應使用偵錯模式。UPLOAD_FOLDER是定義為稍後在應用程式中參考的自訂變數，並通知它應該儲存上傳的使用者資料的位置。
- application.py啟動 Flask 應用程式，並繼承中定義的組態設定。config.py這是由下面的代碼執行：

```
application = Flask(__name__)
application.config.from_pyfile('config.py')
```

更多模式

- 在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight
- 讓 SageMaker 筆記本執行個體暫時存取另一個 AWS 帳戶中的 CodeCommit 儲存庫
- SageMaker 使用 AWS 開發人員工具將 ML 建置、訓練和部署工作負載遷移到 Amazon
- 使用 Amazon Redshift ML 執行進階分析

大型主机

主題

- 使用 BMC AMI 雲端資料將大型主機資料備份並存檔到 Amazon S3
- 在 AWS 雲端建置進階大型主機檔案檢視器
- 容器化已由 Blu Age 現代化的大型主機工作負載
- 使用 Python 在 AWS 上將 EBCDIC 資料轉換並解壓縮為 ASCII
- 使用 AWS Lambda 在 Amazon S3 中將大型主機檔案從 EBCDIC 格式轉換為以字元分隔的 ASCII 格式
- 使用 Micro Focus 轉換具有複雜記錄佈局的大型主機資料檔案
- 使用 Terraform 為容器化的藍光時代應用程式部署環境
- 在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight
- 整合石分支通用控制器與 AWS 大型主機現代化
- 使用精確 Connect 將 VSAM 文件遷移和複寫到 Amazon RDS 或 Amazon MSK
- 使用 OpenText 微焦點企業伺服器和 LRS X 在 AWS 上現代化大型主機輸出管理 PageCenter
- 使用微焦點企業伺服器和 LRS VPSX/MFI，在 AWS 上現代化大型主機批次列印工作負載
- 使用微焦企業伺服器和 LRS VPSX/MFI，在 AWS 上現代化大型主機線上列印工作負載
- 使用 Transfer Family 列將大型主機檔案直接移至 Amazon S3
- 以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3
- 更多模式

使用 BMC AMI 雲端資料將大型主機資料備份並存檔到 Amazon S3

由桑托什庫馬爾辛格（AWS），米凱爾·利伯曼（Model 9 大型主機軟件），吉爾伯托·比昂多（AWS）和李美琪（AWS）創建

環境：PoC 或試點	來源：大型機	目標：Amazon S3
R 類型：不適用	技術：大型主機、儲存與備份、現代化	AWS 服務：Amazon EC2; Amazon EFS; Amazon S3; AWS Direct Connect

Summary

此模式示範如何將大型主機資料直接備份和存檔至 Amazon Simple Storage Service (Amazon S3)，然後使用 BMC AMI 雲端資料 (先前稱為 Model9 管理員) 將該資料還原到大型主機。如果您正在尋找一種方法，將備份與封存解決方案現代化，做為大型主機現代化專案的一部分，或是要符合法規要求，此模式可協助達成這些目標。

一般而言，在大型主機上執行核心商務應用程式的組織會使用虛擬磁帶櫃 (VTL) 來備份檔案和記錄等資料存放區。這個方法可能很昂貴，因為它會耗用可計費的 MIPS，而且儲存在大型主機外部磁帶上的資料無法存取。若要避免這些問題，您可以使用 BMC AMI 雲端資料，快速且符合成本效益的方式，將操作和歷史大型主機資料直接傳輸到 Amazon S3。您可以使用 BMC AMI 雲端資料，透過 TCP/IP 備份和封存資料， AWS 同時利用 IBM z 整合資訊處理器 (ZiIP) 引擎來降低成本、平行處理和傳輸時間。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有有效授權金鑰的 BMC AMI 雲端資料
- 大型主機與 AWS 之間的 TCP/IP 連線
- 讀/寫存取S3 儲存貯體的 AWS Identity and Access Management (IAM) 角色
- 存取大型主機安全性產品 (RACF) 以便執行 BMC AMI 雲端處理程序
- 具有可用網路連接埠的 BMC AMI 雲端 z/OS 代理程式 (Java 版本 8 64 位元 SR5 FP16 或更新版本)、允許存取 S3 儲存貯體的防火牆規則，以及專用 z/FS 檔案系統

- 滿足 BMC AMI 雲端管理伺服器的需求

限制

- BMC AMI 雲端資料會將其作業資料儲存在 PostgreSQL 資料庫中，該資料庫與管理伺服器在相同的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上以碼頭容器的形式執行。目前不支援 Amazon Relational Database Service 服務 (Amazon RDS) 做為 BMC AMI 雲端資料的後端。如需有關最新產品更新的詳細資訊，請參閱新增功能？ 在 BMC 文檔中。
- 此病毒碼只會備份及封存 z/OS 大型主機資料。BMC AMI 雲端資料僅備份和封存大型主機檔案。
- 此模式不會將資料轉換為標準的開放格式，例如 JSON 或 CSV。使用額外的轉換服務，例如 BMC AMI 雲端分析 (以前稱為 Model9 重力) 將資料轉換成標準的開放格式。雲端原生應用程式和資料分析工具可在資料寫入雲端後存取資料。

產品版本

- BMC AMI 雲端資料版本 2.x

架構

源, 技術, 堆棧

- 執行 z/OS 的大型主機
- 大型主機檔案，例如資料集和 z/OS UNIX 系統服務 (USS) 檔案
- 大型主機磁碟，例如直接存取儲存裝置 (DASD)
- 大型主機磁帶 (虛擬或實體磁帶櫃)

目標技術堆疊

- Amazon S3
- 虛擬私有雲 (VPC) 中的 Amazon EC2 執行個體
- AWS Direct Connect
- Amazon Elastic File System (Amazon EFS)

目標架構

下圖顯示了一個參考架構，其中大型主機上的 BMC AMI 雲端資料軟體代理程式驅動將資料存放在 Amazon S3 中的傳統資料備份和存檔程序。

該圖顯示以下工作流程：

1. BMC AMI 雲端資料軟體代理程式會在大型主機邏輯磁碟分割 (LPAR) 上執行。軟體代理程式會透過 TCP/IP 從 DASD 或磁帶直接讀取和寫入大型主機資料至 Amazon S3。
2. AWS Direct Connect 在內部部署網路與 AWS. 為了增強安全性，請執行 site-to-site VPN AWS Direct Connect 來加密傳輸中的資料。
3. S3 儲存貯體會將大型主機檔案存放為物件儲存資料，BMC AMI 雲端資料代理程式會直接與 S3 儲存貯體通訊。憑證用於代理程式與 Amazon S3 之間所有通訊的 HTTPS 加密。Amazon S3 資料加密用於加密和保護靜態資料。
4. BMC AMI 雲端資料管理伺服器會在 EC2 執行個體上以 Docker 容器的形式執行。這些執行個體會與在大型主機 LPAR 和 S3 儲存貯體上執行的代理程式通訊。
5. Amazon EFS 會掛接在主動和被動 EC2 執行個體上，以共用網路檔案系統 (NFS) 儲存。這是為了確保在發生容錯移轉時，與管理伺服器上建立的原則相關的中繼資料不會遺失。如果主動伺服器發生容錯移轉，則可以存取被動伺服器而不會造成任何資料遺失。如果被動伺服器發生故障，可以存取主動伺服器而不會造成任何資料遺失。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在中提供可擴展的運算容量 AWS 雲端。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Elastic File System (Amazon EFS) 可協助您在中建立和設定共用檔案系統 AWS 雲端。
- Amazon Simple Storage Service (Amazon S3) 是雲端物件儲存服務，可協助您存放、保護和擷取幾乎任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。此虛擬網路與您在自己的資料中心中操作的傳統網路相似，且具備使用 AWS 可擴展基礎設施的優勢。
- AWS Direct Connect透過標準乙太網路光纖纜線，將您的內部網路連結至某個 AWS Direct Connect 位置。透過此連線，您可以直接建立公用 AWS 服務的虛擬介面，同時繞過網路路徑中的網際網路服務提供者。

- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。

BMC 工具

- BMC AMI 雲管理服務器是一個 GUI 應用程序，它在 Amazon Amazon Amazon EC2 亞馬遜機器映像（AMI）上作為碼頭容器運行。管理伺服器提供管理 BMC AMI Cloud 活動的功能，例如報告、建立和管理原則、執行封存，以及執行備份、召回和還原。
- BMC AMI 雲端代理程式會在內部部署大型主機 LPAR 上執行，該大型主機 LPAR 會使用 TCP/IP 直接讀取和寫入物件儲存裝置。已啟動的任務在大型主機 LPAR 上執行，負責在 Amazon S3 和從 Amazon S3 讀取和寫入備份和存檔資料。
- BMC AMI 雲端大型主機命令列介面 (M9CLI) 提供一組指令，可直接從 TSO/E 或批次作業執行 BMC AMI 雲端動作，而不需要依賴管理伺服器。

史诗

建立 S3 儲存貯體和 IAM 政策

任務	描述	所需技能
建立 S3 儲存貯體。	建立 S3 儲存貯體以存放您要從大型主機環境備份和存檔的檔案和磁碟區。	一般 AWS
建立 IAM 政策。	所有 BMC AMI 雲端管理伺服器和代理程式都需要存取您在上一步中建立的 S3 儲存貯體。 若要授予必要的存取權，請建立下列 IAM 政策： `{ "Version": "2012-10-17", "Statement": [ {`	一般 AWS

任務	描述	所需技能
	<pre>"Sid": "Listfolder", "Action": ["s3:ListBucket", "s3:GetBucketLocation", "s3:ListBucketVersions"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<Bucket Name>"] }, { "Sid": "Objectaccess", "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVersion", "s3:DeleteObject", "s3:PutObjectAcl", "s3:GetObjectVersion"</pre>	

任務	描述	所需技能
	<pre>], "Resource": ["arn:aws:s3:::<Bucket Name>/*"] }] }</pre>	

獲取 BMC AMI 雲軟件許可證並下載軟件

任務	描述	所需技能
取得 BMC AMI 雲端軟體授權。	若要取得軟體授權金鑰，請聯絡 BMC AMI 雲端團隊。產生授權時需要 z/OS D M=CPU 指令的輸出。	建立領導
下載 BMC AMI 雲端軟體和授權金鑰。	依照 BMC 說明文件中的指示取得安裝檔案和授權金鑰。	大型主機基礎架構管理員

在大型主機上安裝 BMC AMI 雲端軟體代理程式

任務	描述	所需技能
安裝 BMC AMI 雲端軟體代理程式。	1. 開始安裝程序之前，請確認已符合代理程式的最低軟體和硬體需求。 2. 若要安裝代理程式，請遵循 BMC 說明文件中的指示。 3. 代理程式在大型主機 LPAR 上開始執行之後，請檢查佇列中是否有ZM91000I	大型主機基礎架構管理員

任務	描述	所需技能
	MODEL9 BACKUP AGENT INITIALIZED 訊息。透過在代理程式的 STDOUT 中尋找Object store connectivity has been established successfully 訊息，確認代理程式與 S3 儲存貯體之間已成功建立連線。	

在 EC2 執行個體上設定 BMC AMI 雲端管理伺服器

任務	描述	所需技能
創建 Amazon EC2 2 個實例。	遵循 Amazon EC2 文件中的步驟 1：啟動執行個體的指示，在不同的可用區域啟動兩個 Amazon EC2 Linux 2 執行個體。 執行個體必須符合下列建議的硬體和軟體需求： • 處理器 — 最少 4 個核心 • 記憶體 — 至少 8 GB • 磁碟機 — 40 GB • 推薦的 EC2 執行個體 — 大型 • 作業系統 — Linux • 軟體 — 泊塢視窗、解壓縮、Vim • 網路頻寬 — 最低 1 GB	雲端架構師、雲端管理員

任務	描述	所需技能
	如需詳細資訊，請參閱 BMC 文件。	
建立一個 Amazon EFS 檔案系統。	依照 Amazon EFS 文件中的步驟 1：建立您的 Amazon EFS 檔案系統中的指示，建立 Amazon EFS 檔案系統。 建立檔案系統時，請執行下列動作： • 選擇標準儲存空間類別。 • 選擇您用來啟動 EC2 執行個體的相同 VPC。	雲端管理員、雲端架構師

任務	描述	所需技能
安裝 Docker 並設定管理伺服器。	Connect 至您的 EC2 執行個體： 按照 Amazon EC2 文件中的連線到 Linux 執行個體的指示，Connect 到您的 EC2 執行個體。 設定您的 EC2 執行個體： 對於每個 EC2 執行個體，請執行下列動作： 1. 要安裝 Docker，請運行以下命令： `sudo yum install docker` 2. 要啟動 Docker，請運行以下命令： `sudo service docker start` 3. 若要驗證 Docker 的狀態，請執行下列命令： `sudo service docker status` 4. 在`/etc/selinux` 資料夾中，將`config`檔案變更為`SELINUX=permissive`。 5. 將`model9-v2.x.y_build_build-id-server.`	雲端架構師、雲端管理員

任務	描述	所需技能
	zip 和VerificationScripts.zip 檔案 (您之前下載的檔案) 上傳到其中一個 EC2 執行個體中的暫存資料夾 (例如，到執行個體中的/var/tmp資料夾)。 6. 要轉到該tmp文件夾，請運行以下命令： `cd/var/tmp` 7. 若要解壓縮驗證指令碼，請執行以下指令： `unzip VerificationScripts.zip` 8. 若要變更目錄，請執行以下指令： `cd /var/tmp/sysutils/PrereqsScripts` 9. 若要執行驗證指令碼，請執行下列命令： `./M9VerifyPrereqs.sh` 10驗證指令碼提示輸入後，輸入 Amazon S3 URL 和連接埠號碼。然後，輸入 z/OS IP/DNS 和連接埠號碼。 備註：指令碼會執行檢查，以確認 EC2 執行個體可以與	

任務	描述	所需技能
	大型主機上執行的 S3 儲存貯體和代理程式連線。如果已建立連線，則會顯示成功訊息。	

任務	描述	所需技能
安裝管理伺服器軟體。	1. 在您打算建立作用中伺服器的 EC2 執行個體中的根目錄 (例如/data/model9) 中建立資料夾和子資料夾。 2. 若要安裝amazon-efs-utils套件並掛接先前建立的 Amazon EFS 檔案系統，請執行下列命令： `sudo yum install -y amazon-efs-utils` `sudo mount -t efs -o tls <File System ID>:/ /data/model9` 3. 若要使用 Amazon EFS /etc/fstab 檔案系統的項目來更新 EC2 執行個體的檔案 (以便在 Amazon EC2 重新啟動時自動重新掛載 Amazon EFS)，請執行以下命令： `<Amazon-EFS-file-system-id>:/ /data/model9 efs defaults,_netdev 0 0` 4. 若要定義 BMC AMI Cloud 安裝檔案的路徑和目標安裝位置，請執行下列命令以匯出變數： `export MODEL9_HOME=/data/model9` `export M9INSTALL=/var/tmp`	雲端架構師、雲端管理員

任務	描述	所需技能
	注意：我們建議您將這些 EXPORT 命令新增至指令.bashrc碼。 5. 若要變更目錄，請執行cd $MODEL9_HOME 命令，然後執行命mkdir diag令來建立另一個子目錄。 6. 若要解壓縮安裝檔案，請執行以下指令： `unzip $M9INSTALL/model9-<v2.x.y>_build_<build-id>-server.zip` 注意：替換x.y（版本）和您build-id的值。 7. 若要部署應用程式，請執行下列命令： `docker load -i $MODEL9_HOME/model9-<v2.x.y>_build_<build-id>.docker` `docker load -i $MODEL9_HOME/postgres-12.10-x86.docker.gz` 注意：替換v2.x.y（版本）和您build-id的值。 8. 在$MODEL9_HOME/conf資料夾中，更新model9-local.yml 檔案。	

任務	描述	所需技能
	注意：某些參數具有默認值，其他參數可以根據需要進行更新。如需詳細資訊，請參閱model9-local.yml 檔案中的指示。 9. 建立名為的檔案 **$MODEL9_HOME/conf** ，然後將下列參數新增至檔案： `TZ=America/New_York` `EXTRA_JVM_ARGS=-Xmx2048m` 10.若要建立 Docker 網路橋接器，請執行以下指令： `docker network create -d bridge model9network` 11.若要啟動 BMC AMI 雲端的 PostgreSQL 資料庫容器，請執行下列命令： `docker run -p 127.0.0.1:5432:5432 \` `-v $MODEL9_HOME/db/data:/var/lib/postgresql/data:z \` `--name model9db --restart unless-stopped \` `--network model9network \` `-e POSTGRES_PASSWORD=model9 -e POSTGRES_`	

任務	描述	所需技能
	DB=model9 -d postgres:12.10	
	12.PostgreSQL 容器開始執行之後，請執行下列命令以啟動應用程式伺服器： docker run -d -p 0.0.0.0:443:443 -p 0.0.0.0:80:80 \ --sysctl net.ipv4. tcp_keepalive_time =600 \ --sysctl net.ipv4. tcp_keepalive_intv l=30 \ --sysctl net.ipv4. tcp_keepalive_prob es=10 \ -v $MODEL9_HOME:/mode l9:z -h $(hostname) --restart unless-st opped \ --env-file $MODEL9_H OME/conf/model9.env \ --network model9net work \ --name model9-v2.x.y model9:<v2.x.y>.<b uild-id> 注意：替換v2.x.y（版本）和您build-id的值。 13.若要檢查兩個容器的健全狀況狀態，請執行以下命令： docker ps -a	

任務	描述	所需技能
	14.若要在被動 EC2 執行個體上安裝管理伺服器，請重複步驟 1—4、7 和 10-13。 注意：若要疑難排解問題，請移至資/data/model9/logs/ 料夾中儲存的記錄檔。如需詳細資訊，請參閱 BMC 文件。	

在 BMC AMI 雲端管理伺服器上新增代理程式並定義備份或封存原則

任務	描述	所需技能
新增代理程式。	新增代理程式之前，請先確認下列事項： • BMC AMI 雲端代理程式正在大型主機 LPAR 上執行，且已完全初始化。在多工緩衝處理中尋找ZM91000I MODEL9 BACKUP AGENT INITIALIZED 初始化訊息來識別代理程式。 • 管理伺服器的 Docker 容器已完全初始化並執行。 您必須先在管理伺服器上建立代理程式，才能定義任何備份和封存原則。如果要建立代理程式，請執行下列動作： 1. 使用網頁瀏覽器存取部署在 Amazon EC2 機器上的管理	大型主機儲存管理員或開發人員

任務	描述	所需技能
	伺服器，然後使用大型主機登入資料登入。 2. 選擇代理程式索引標籤，然後選擇 [新增代理程式]。 3. 在名稱中，輸入代理程式名稱。 4. 對於主機名稱 /IP 位址，請輸入大型主機的主機名稱或 IP 位址。 5. 在連接埠中，輸入您的連接埠號碼。 6. 選擇 [測試連線]。如果成功建立連線，您可以看到成功訊息。 7. 選擇 CREATE (建立)。 建立代理程式之後，您會在表格中顯示的新視窗中看到物件儲存和大型主機代理程式的連線狀態。	
建立備份或封存原則。	1. 選擇 [策略]。 2. 選擇 [建立原則]。 3. 在 [建立新原則] 頁面上，輸入您的原則規格。 注意：如需可用規格的詳細資訊，請參閱 BMC 說明文件中的建立新原則。 4. 選擇 Finish (完成)。 5. 新政策現在會以表格形式列出。若要查看此表格，請選擇策略標籤。	大型主機儲存管理員或開發人員

從管理伺服器執行備份或封存原則

任務	描述	所需技能
執行備份或封存原則。	手動或自動執行您先前從管理伺服器建立的資料備份或封存原則 (根據排程)。若要手動執行原則： 1. 從導覽功能表中選擇 [策略] 索引標籤。 2. 在您要執行之原則的表格右側，選擇三點功能表。 3. 選擇「立即執行」。 4. 在彈出的確認視窗中，選擇是，立即執行原則。 5. 策略執行後，請在策略活動區段中確認執行狀態。 6. 針對執行的原則，請選擇三點功能表，然後選擇 [檢視執行記錄檔] 以查看記錄檔。 7. 若要確認已建立備份，請檢查 S3 儲存貯體。	大型主機儲存管理員或開發人員
還原備份或封存原則。	1. 在導覽功能表上，選擇 [策略] 索引標籤。 2. 選擇執行還原程序的原則。這會列出過去針對該特定原則執行的所有備份或封存活動。 3. 若要選取要還原的備份，請選擇 [日期時間] 資料行。群`file/Volume/Storage` 組名稱會顯示原則的執行詳細資料。	大型主機儲存管理員或開發人員

任務	描述	所需技能
	4. 在表格右側，選擇三點功能表，然後選擇 [還原]。 5. 在快顯視窗中，輸入目標名稱、磁碟區和儲存群組，然後選擇 [還原]。 6. 輸入您的大型主機認證，然後再次選擇 [還原]。 7. 若要確認還原是否成功，請檢查記錄檔或大型主機。	

從大型主機執行備份或封存原則

任務	描述	所需技能
使用 M9CLI 執行備份或封存原則。	使用 M9CLI 從 TSO/E、REXX 或透過 JCLS 執行備份與還原程序，而無需在 BMC AMI 雲端管理伺服器上設定規則。 使用 TSO/E： 如果您使用 TSO/E，請確定已`M9CLI REXX`連結至。`TSO`若要透過 TSO/E 備份資料集，請使用指令。`TSO M9CLI BACKDSN <DSNAME>` 注意：如需有關 M9CLI 命令的詳細資訊，請參閱 BMC 說明文件中的 CLI 參考資料。 使用 JCLs： 若要使用 JCL 執行備份和封存原則，請執行命令。`M9CLI`	大型主機儲存管理員或開發人員

任務	描述	所需技能
	使用批處理操作： 下列範例說明如何透過批次執行M9CLI命令來封存資料集： `//JOBNAME JOB …` `//M9CLI EXEC PGM=IKJEF` `T01` `//STEPLIB DD DISP=SHR,` `DSN=<MODEL9 LOADLIB>` `//SYSEXEC DD DISP=SHR,` `DSN=<MODEL9 EXEC LIB>` `//SYSTSPRT DD SYSOUT=*` `//SYSPRINT DD SYSOUT=*` `//SYSTSIN DD TSO M9CLI` ` ARCHIVE` ` M9CLI ARCHIVE <DSNNAME` ` OR DSN PATTERN>` `/`	
在 JCL 批次中執行備份或封存原則。	BMC AMI 雲提供了一個名為 M9SAPIJ 的例程 JCL 例程。您可以自訂 M9SAPIJ 來執行使用 JCL 在管理伺服器上建立的特定原則。此工作也可以是批次排程器的一部分，用於自動執行備份和還原程序。 批次工作需要下列必要值： • 管理伺服器IP位址/主機名稱 • 連接埠號碼 • 策略 ID 或策略名稱 (在管理伺服器上建立) 附註：您也可以依照範例工作的指示來變更其他值。	大型主機儲存管理員或開發人員

相關資源

- 使用 AWS 進行大型主機現代化 (AWS 文件)
- 大型主機的雲端 Backup 如何使用 Model9 和 AWS 降低成本 (AWS 合作夥伴網路部落格)
- 如何使用 Model9 在 AWS 上啟用大型主機資料分析 (AWS 合作夥伴網路部落格)
- AWS Direct Connect 彈性建議 (AWS 文件)
- BMC AMI 雲端文件 (BMC 網站)

在 AWS 雲端建置進階大型主機檔案檢視器

創建者：戈帕爾薩米（AWS）和耶利米奧康納（AWS）

環境：PoC 或試點	技術：大型主機、移轉、無伺服器	工作負載：IBM
AWS 服務：Amazon Athena；AWS Lambda；Amazon OpenSearch 服務；AWS Step Functions		

Summary

此模式提供程式碼範例和步驟，協助您建立進階工具，以使用 AWS 無伺服器服務瀏覽和檢閱大型主機固定格式檔案。此模式提供如何將大型主機輸入檔案轉換為 Amazon Ser OpenSearch vice 文件以進行瀏覽和搜尋的範例。文件查看器工具可以幫助您實現以下目標：

- 保留相同的大型主機檔案結構和配置，以確保 AWS 目標移轉環境的一致性 (例如，您可以在將檔案傳輸至外部對象的批次應用程式中維持相同的檔案配置)
- 在大型主機移轉期間加速開發與測試
- Support 移轉後的維護活動

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有可供您舊式平台存取之子網路的虛擬私有雲 (VPC)
- 輸入檔案及其對應的通用面向業務語言 (COBOL) 字帖本 (注意：如需輸入檔案和 COBOL 字帖的範例，請參閱儲存庫中的內容。gfs-mainframe-solutions GitHub 如需 COBOL 撰寫本的詳細資訊，請參閱 IBM 網站上的《z/OS 6.3 版企業 COBOL 程式設計指南》。)

限制

- 字帖解析僅限於不超過兩個嵌套級別（發生）

架構

源, 技術, 堆棧

- FB（固定阻止）格式的輸入文件
- 字帖書佈局

目標技術堆疊

- Amazon Athena
- Amazon OpenSearch 服務
- Amazon Simple Storage Service (Amazon S3)
- AWS Lambda
- AWS Step Functions

目標架構

下圖顯示剖析大型主機輸入檔案，並將其轉換為 OpenSearch Service 文件以供瀏覽和搜尋的程序。

該圖顯示以下工作流程：

1. 管理員使用者或應用程式會將輸入檔案推送到一個 S3 儲存貯體，將 COBOL 撰寫本推送到另一個 S3 儲存貯體。
2. 具有輸入檔案的 S3 儲存貯體會叫用 Lambda 函數，以啟動無伺服器 Step Functions 數工作流程。注意：在此模式中使用 S3 事件觸發器和 Lambda 函數來驅動 Step Functions 數工作流程是可選的。此模式中的程式 GitHub 碼範例不包括使用這些服務，但您可以根據需求使用這些服務。
3. 「Step Functions」工作流程會協調下列 Lambda 函數中的所有批次處理：
   - 該`s3copybookparser.py`函數解析字帖本佈局並提取字段屬性，數據類型和偏移量（輸入數據處理所需）。
   - 該`s3toathena.py`函數創建一個 Athena 表格佈局。Athena 會剖析`s3toathena.py`函數處理的輸入資料，並將資料轉換為 CSV 檔案。

- 此`s3toelasticsearch.py`函數會從 S3 儲存貯體擷取結果檔案，並將檔案推送至 OpenSearch「服務」。

4. 使用者存取具有 OpenSearch 服務的 OpenSearch 儀表板，以擷取各種資料表和資料行格式的資料，然後針對索引資料執行查詢。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中分析資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。在此模式中，您可以使用 Lambda 實作核心邏輯，例如剖析檔案、轉換資料，以及將資料載入 OpenSearch Service 以進行互動式檔案存取。
- Amazon OpenSearch 服務是一種受管服務，可協助您在 AWS 雲端部署、操作和擴展 OpenSearch 服務叢集。在此模式中，您可以使用 OpenSearch Service 來索引轉換後的檔案，並為使用者提供互動式搜尋功能。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 Lambda 函數和其他 AWS 服務，以建立關鍵業務應用程式。在此模式中，您可以使用 Step Functions 來協調 Lambda 函數。

其他工具

- GitHub是一種代碼託管服務，提供協作工具和版本控制。
- Python 是一種高級別的編程語言。

Code

此模式的代碼可在 GitHub gfs-mainframe-patterns存儲庫中找到。

史诗

準備目標環境

任務	描述	所需技能
建立 S3 儲存貯體。	建立 S3 儲存貯體，用於存放撰稿本、輸入檔案和輸出檔案。我們建議您的 S3 儲存貯體採用下列資料夾結構： • `copybook/` • `input/` • `output/` • `query/` • `results/`	一般 AWS
創建 s3copybook 解析器函數。	1. 建立名為的 Lambda 函數，`s3copybookparser` 並從GitHub儲存庫上傳原始程式碼 (`s3copybookparser.py` 和`copybook.py`)。 2. 將 IAM 政策附加`S3ReadOnly` 至 Lambda 函數。	一般 AWS
建立伺服器函數。	1. 建立名為的 Lambda 函數，`s3toathena` 並從GitHub儲存庫上傳原始程式碼 (`s3toathena.py`)。將 Lambda 逾時設定為超過 60 秒。 2. 若要提供所需資源的存取權，請將 IAM 政策`AmazonAthenaFullAc`	一般 AWS

任務	描述	所需技能
	cess 附加S3FullAcc ess 至 Lambda 函數。	
創建 s3 彈性搜索功能。	1. 將 Python 相依性新增至您的 Lambda 環境。重要事項：若要使用s3toelast icsearch 函數，您必須新增 Python 相依性，因為 Lambda 函數使用 Python 彈性搜尋用戶端相依性 (Elasticsearch==7.9 .0 和requests_ aws4auth)。 2. 建立名為的 Lambda 函數，s3toelast icsearch 並從GitHub儲存庫上傳原始程式碼 (s3toelasticsearch. py)。 3. 將 Python 相依性匯入為 Lambda 層。 4. 將身分S3ReadOnl y 與存取權管理政策附加AmazonOpenSearchSe rviceReadOnlyAcces s 至 Lambda 函數。	一般 AWS

任務	描述	所需技能
建立服 OpenSearch 務叢集。	建立叢集 1. 建立OpenSearch 服務叢集。建立叢集時，請執行下列動作： • 為可用於登入OpenSearch 儀表板的叢集建立主要使用者和密碼。注意：如果您透過 Amazon Cognito 使用身份驗證，則不需要執行此步驟。 • 選擇精細的存取控制。這為您提供了控制 OpenSearch 服務中數據訪問的其他方法。 2. 複製網域 URL，並將其做為環境變數 「HOST」 傳遞至 Lambda 函數`s3toelasticsearch` 。 授予 IAM 角色的存取權 若要提供對 Lambda 函數 IAM 角色 (`arn:aws:iam::**:role/service-role/s3toelasticsearch-role-**`) 的細微存取權限，請執行下列動作： 1. 以主要使用者身分登入 OpenSearch 儀表板。 2. 選擇 [安全性] 索引標籤，然後選擇 [角色]、[all_acces	一般 AWS

任務	描述	所需技能
	s]、[對應使用者]、[後端角色] 3. 新增 Lambda 函數 IAM 角色的 Amazon 資源名稱 (ARN)，然後選擇 [儲存]。如需詳細資訊，請參閱 OpenSearch Service 說明文件中的將角色對應至使用者。	
建立用於協調流程的 Step Functions。	1. 建立具有標準流程的 Step Functions 狀態機。定義包含在GitHub 存放庫中。 2. 在 JSON 指令碼中，將 Lambda 函數的 ARN 取代為環境中 Lambda 函數的 ARN。	一般 AWS

部署和執行

任務	描述	所需技能
將輸入檔案和抄本上傳到 S3 儲存貯體。	從GitHub 存放庫範例資料夾下載範例檔案，並將檔案上傳到您先前建立的 S3 儲存貯體。 1. 上傳`Mockedcopy.cpy` 並`acctix.cpy` 到文`<S3_Bucket>/copybook` 件夾。 2. 將`Modedupdate.txt` 和範`acctindex.cpy` 例輸入檔案上傳至資`<S3_Bucket>/input` 料夾。	一般 AWS

任務	描述	所需技能
呼叫 Step Functions。	1. 登入 AWS 管理主控台並開啟 Step Functions 主控台。 2. 在瀏覽窗格中，選擇 [狀態機器]。 3. 選擇您的狀態機器，然後選擇 [開始執行]。 4. 在 [輸入] 方塊中，輸入下列抄本/檔案路徑做為 S3 儲存貯體的 JSON 變數，然後選擇 [開始執行]。 `{ "s3_copybook_bucket_name": "<BUCKET NAME>", "s3_copybook_bucket_key": "<COPYBOOK PATH>", "s3_source_bucket_name": "<BUCKET NAME", "s3_source_bucket_key": "INPUT FILE PATH" }` 例如： `{ "s3_copybook_bucket_name": "fileaidtest", "s3_copybook_bucket_key": "copybook/acctix.cpy", "s3_source_bucket_name": "fileaidtest",`	一般 AWS

任務	描述	所需技能
	<pre>"s3_source_bucket_ key": "input/ac ctindex" }</pre>	
驗證 Step Functions 中的工作流程執行。	在「Step Functions」主控台中，檢閱「圖形」檢視窗中的工作流程執行。執行執行狀態會以顏色編碼來代表執行狀態。例如，藍色表示進行中，綠色表示成功，紅色表示失敗。您也可以複查「執行事件歷史記錄」段落中的表格，以取得有關執行事件的詳細資訊。 如需圖形化工作流程執行的範例，請參閱此模式的其他資訊一節中的 Step Functions 圖表。	一般 AWS
在 Amazon 驗證交付日誌 CloudWatch。	1. 登入 AWS 管理主控台，並開啟 CloudWatch 主控台。 2. 在瀏覽窗格中，展開 [記錄檔]，然後選擇 [記錄群組]。 3. 在搜尋方塊中，搜尋 `s3toelasticsearch` 函數的記錄群組。 如需成功傳送記錄檔的範例，請參閱此病毒碼的其他資訊一節中的CloudWatch 傳送記錄檔。	一般 AWS

任務	描述	所需技能
驗證 OpenSearch 儀表板中的格式化檔案並執行檔案作業。	1. 登入 AWS 管理主控台。在分析下，選擇 Amazon OpenSearch 服務。 2. 在導覽窗格中，選擇 [網域]。 3. 在搜尋方塊中，在OpenSearch 儀表板中輸入網域的 URL。 4. 選擇儀表板，然後以主要使用者身分登入。 5. 以表格格式瀏覽索引資料。 6. 將輸入檔案與 OpenSearch 儀表板中格式化的輸出檔案 (索引文件) 進行比較。儀表板檢視會顯示格式化檔案新增的欄標題。確認來自未格式化輸入檔案的來源資料與儀表板檢視中的目標資料相符。 7. 針對索引的檔案執行搜尋 (例如，使用欄位名稱、值或運算式)、篩選器和 DQL (儀表板查詢語言) 等動作。	一般 AWS

相關資源

參考

- 示例 COBOL 字帖（IBM 文檔）
- BMC 電腦軟體檔案援助 (BMC 文件)

教學課程

- 教學課程：使用 Amazon S3 觸發程序叫用 Lambda 函數 (AWS Lambda 文件)
- 如何使用 AWS 步驟函數和 AWS Lambda (AWS 文件) 建立無伺服器工作流程
- 將 OpenSearch 儀表板與 Amazon OpenSearch 服務搭配使用 (AWS 文件)

其他資訊

Step Functions 圖

下列範例顯示 Step Functions 圖形。此圖形顯示此模式中使用的 Lambda 函數的執行執行狀態。

CloudWatch 傳送記錄

下列範例顯示執行作業的成功傳送記錄檔。`s3toelasticsearch`

2022-08-10 嘸 15:53:33.033-05:00	處理文件數量：100	
	2022-08-10 嘸 15:53：33.171-05:00	[資訊] 2022-08-10T20：33:33.171 Z A1B2C3D4-5678-90B-常見問題的例子：https://search-essearch-3h4uqclifeqaj2vg4mphe7ffle.us-east-2.es.amazonaws.com:443/_bulk [狀態:200 請求:0.100 秒]
	2022-08-10 嘸 15:53：33.172-05:00	大量寫入成功：100 份文件

容器化已由 Blu Age 現代化的大型主機工作負載

由理查德·米爾納瓦特（AWS）創建

代碼庫：藍光時代應用程序容器示例	環境：生產	來源：大型主機工作負載
目標：容器	R 型：重新建築	工作負載：IBM；所有其他工作負載
技術：大型主機；容器與微服務；移轉；現代化	AWS 服務：Amazon ECS; Amazon ECR	

Summary

此模式為執行使用 Blu Age 工具現代化的大型主機工作負載提供範例容器環境。Blu Age 可將舊式大型主機工作負載轉換為現代 Java 程式碼。此模式提供了圍繞 Java 應用程序的包裝，因此您可以使用容器協調服務，如 Amazon Elastic Container Service (Amazon ECS) 或 Amazon E lastic Kubernetes Service（Amazon EKS）來運行它。

如需使用 Blu Age 和 AWS 服務將工作負載現代化的詳細資訊，請參閱以下 AWS Prescriptive Guidance 出版物：

- 在無伺服器 AWS 基礎設施上執行現代化的 Blu Age 大型主機工作負載
- 使用 Terraform 為容器化的藍光時代應用程式部署環境

如需使用 Blu Age 現代化大型主機工作負載的協助，請在 Blu Age 網站上選擇聯絡我們的專家，聯絡 Blu Age 團隊。如需將現代化工作負載遷移到 AWS、將它們與 AWS 服務整合以及將它們移入生產環境的協助，請聯絡您的 AWS 客戶經理或填寫 AWS Professional Services 表單。

先決條件和限制

先決條件

- 由藍光時代創建的現代化 Java 應用程序。出於測試目的，此模式提供了一個示例 Java 應用程序，您可以將其用作概念證明。
- 您可以用來建置容器的 Docker 環境。

限制

視您使用的容器協調流程平台而定，可供容器使用的資源 (例如 CPU、RAM 和儲存區) 可能會受到限制。例如，如果您將 Amazon ECS 搭配 AWS Fargate 使用，請參閱 Amazon ECS 文件以瞭解限制和考量事項。

架構

源, 技術, 堆棧

- 藍色時代
- Java

目標技術堆疊

- Docker

目標架構

下圖顯示了 Docker 容器中的藍光時代應用程序的體系結構。

1. 容器的進入點是包裝函式指令碼。此 bash 腳本負責為 Blu Age 應用程序準備運行時環境並處理輸出。
2. 容器內的環境變數用於設定包裝器指令碼中的變數，例如 Amazon Simple Storage Service (Amazon S3) 貯體名稱和資料庫登入資料。環境變數由 AWS Secrets Manager 或參數存放區 (AWS Systems Manager 的一項功能) 提供。如果您使用 Amazon ECS 做為容器協調服務，也可以在 Amazon ECS 任務定義中對環境變數進行硬式編碼。
3. 包裝腳本負責在運行 Blu Age 應用程序之前，將 S3 存儲桶中的任何輸入文件拉入容器。AWS Command Line Interface (AWS CLI) (AWS CLI) 已安裝在容器內。這提供了一種機制，可讓您透過閘道虛擬私有雲端 (VPC) 端點存取存放在 Amazon S3 中的物件。
4. 藍光時代應用程式的 Java 封存檔 (JAR) 檔案可能需要與其他資料來源 (例如 Amazon Aurora) 通訊。
5. 完成後，包裝器指令碼會將產生的輸出檔案交付到 S3 儲存貯體，以供進一步處理 (例如，透過 Amazon CloudWatch 記錄服務)。如果您使用的是標準 CloudWatch 記錄的替代方案，則此模式也支援將壓縮的日誌檔傳遞到 Amazon S3。

工具

AWS 服務

- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。

工具

- Docker 是用於構建，測試和部署應用程序的軟件平台。Docker 將軟體封裝成稱為容器的標準化單元，這些單元包含軟體所需的一切，包括程式庫、系統工具、程式碼和執行階段。您可以使用 Docker 將應用程式部署和擴展到任何環境中。
- Bash 是 GNU 操作系統的命令語言界面（外殼）。
- Java 是在這種模式中使用的編程語言和開發環境。
- Blu Age 是 AWS 大型主機現代化工具，可將舊式大型主機工作負載 (包括應用程式程式碼、相依性和基礎設施) 轉換為雲端的現代工作負載。

代碼存儲庫

此模式的代碼可在 GitHub Blu Age 示例容器存儲庫中找到。

最佳實務

- 外部化的變量，通過使用環境變量改變你的應用程序的行為。這些變數可讓容器協調流程解決方案在不重建容器的情況下變更執行階段環境。此模式包括可用於 Blu Age 應用程序的環境變量示例。
- 在運行 Blu Age 應用程序之前驗證所有應用程序依賴關係 例如，請確認資料庫可供使用，而且認證是否有效。在包裝腳本中編寫測試以驗證依賴關係，如果不滿足，則提前失敗。
- 在包裝器腳本中使用詳細日誌記錄。直接與執行中的容器互動可能具有挑戰性，具體取決於協調流程平台以及工作所需的時間。確保寫入有用的輸出以幫STDOUT助診斷任何問題。例如，輸出可能會在您執行應用程式之前和之後包含應用程式的工作目錄內容。

史诗

獲取藍色時代應用程序 JAR 文件

任務	描述	所需技能
選項 1-與藍色時代合作以獲取應用程序的 JAR 文件。	此模式中的容器需要 Blu Age 應用程序。或者，您也可以使用此模式提供的範例 Java 應用程式來建立原型。 與 Blu Age 團隊合作，為您的應用程序獲取可以烘烤到容器中的 JAR 文件。如果 JAR 檔案無法使用，請參閱下一個使用範例應用程式的工作。	雲端架構師
選項 2-建置或使用提供的範例應用程式 JAR 檔案。	此模式提供預先建置的範例 JAR 檔案。該文件將應用程序的環境變量輸出到睡眠 30 秒並退出STDOUT之前。 該文件被命名為`bluAgeSample.jar` ，位於 GitHub 存儲庫的 docker 文件夾中。 如果您想要變更程式碼並建置您自己的 JAR 檔案版本，請使用位於. /java_sample/src/sample_java_app.java 在 GitHub 存儲庫中。您可以在中使用構建腳本。 /java_sample/build.sh 編譯 Java 源代碼並構建一個新的 JAR。	應用程式開發人員

建立藍色時代容器

任務	描述	所需技能
克隆存 GitHub 儲庫。	使用以下命令克隆示例代碼存儲庫： `git clone https://github.com/aws-samples/aws-blu-age-sample-container`	AWS DevOps
使用泊塢窗來構建容器。	在將容器推送到 Docker 註冊表（例如 Amazon ECR）之前，請使用 Docker 來構建容器： 1. 從您選擇的終端中，導航到本地 GitHub 存儲庫中的`docker`文件夾。 2. 使用此命令來構建容器： `docker build -t <tag> .` 其中`<tag>` 是您要使用的容器名稱。	AWS DevOps
測試藍光時代容器。	(選擇性) 如有必要，請使用以下指令在本機測試容器： `docker run -it <tag> /bin/bash`	AWS DevOps
驗證到您的碼頭存儲庫。	如果您打算使用 Amazon ECR，請按照 Amazon ECR 文件中的指示安裝和設定 AWS	AWS DevOps

任務	描述	所需技能
	CLI，並向您的預設登錄驗證 Docker CLI。 我們建議您使用命get-login-password 令進行驗證。 注意：如果您使用「檢視」按鈕，Amazon ECR 主控台會提供此命令的預先填入版本。如需詳細資訊，請參閱 Amazon ECR 文件。 `aws ecr get-login -password --region <region> | docker login --username AWS --password-stdin <account>.dkr.ecr.<region>.amazonaws.com` 如果您不打算使用 Amazon ECR，請依照容器登錄系統提供的指示操作。	
建立容器存放庫。	在 Amazon ECR 中創建一個存儲庫。如需指示，請參閱模式使用 T erraform 為容器化的 Blu Age 應用程式部署環境。 如果您正在使用其他容器登錄系統，請遵循該系統提供的指示。	AWS DevOps

任務	描述	所需技能
標記容器並將其推送到目標存儲庫。	如果您使用的是 Amazon ECR： 1. 使用 Amazon ECR 登錄和儲存庫標記本機 Docker 映像，以便將其推送到遠端儲存庫： `docker tag <tag>:latest <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber>` 2. 將映像推送到遠程存儲庫： `docker push <account>.dkr.ecr.<region>.amazonaws.com/<repository>:<versionNumber>` 如需詳細資訊，請參閱 Amazon ECR 使用者指南中的推送泊塢視窗映像。	AWS DevOps

相關資源

AWS 資源

- AWS 藍光時代範例容器儲存庫
- 在無伺服器 AWS 基礎設施上執行現代化的 Blu Age 大型主機工作負載
- 使用 Terraform 為容器化的藍光時代應用程式部署環境
- 搭配 AWS CLI 使用 Amazon ECR (Amazon ECR 使用者指南)

- 私有登錄身份驗證 (Amazon ECR 使用者指南)
- Amazon ECS 文件
- Amazon EKS 文檔

其他資源

- 藍色時代網站
- 碼頭網站

使用 Python 在 AWS 上將 EBCDIC 資料轉換並解壓縮為 ASCII

創建者：路易斯古斯塔沃丹達斯 (AWS)

程式碼儲存庫:大型主機資料公用程式	環境：PoC 或試點	資料來源：大型主機 EBCDIC 資料
目標：分散式或雲端現代化 ASCII 資料	R 類型：重新平台	工作負載：IBM
技術：大型主機、資料庫、儲存與備份、現代化	AWS 服務：Amazon EBS; Amazon EC2	

Summary

由於大型主機通常託管關鍵業務資料，因此將資料移轉到 Amazon Web Services (AWS) 雲端或其他美國標準資訊交換代碼 (ASCII) 環境時，資料現代化是最重要的任務之一。在大型主機上，資料通常會以延伸的二進位編碼十進位交換碼 (EBCDIC) 格式編碼。匯出資料庫、虛擬儲存存取方法 (VSAM) 或平面檔案通常會產生封裝的二進位 EBCDIC 檔案，而這些檔案移轉較為複雜。最常用的資料庫遷移解決方案是變更資料擷取 (CDC)，在大多數情況下，它會自動轉換資料編碼。不過，CDC 機制可能不適用於這些資料庫、VSAM 或平面檔案。對於這些文件，需要一種替代方法來實現數據現代化。

此模式描述了如何通過將 EBCDIC 數據轉換為 ASCII 格式來實現現代化。轉換後，您可以將數據加載到分佈式數據庫中，或者讓雲中的應用程序直接處理數據。該模式使用存mainframe-data-utilities GitHub 儲庫中的轉換腳本和示例文件。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- EBCDIC 輸入文件及其相應的面向業務的通用語言（COBOL）字帖本。存儲庫中包含了一個示例 EBCDIC 文件和 COBOL 字帖本。mainframe-data-utilities GitHub 如需 COBOL 撰寫本的詳細資訊，請參閱 IBM 網站上的企業 COBOL 適用於 z/OS 6.4 的程式設計指南。

限制

- 不支援 COBOL 程式內定義的檔案配置。它們必須單獨提供。

產品版本

- Python 版本 3.8 或更新版本

架構

源, 技術, 堆棧

- 在大型主機上的 EBCDIC 數據
- 聯邦字帖

目標技術堆疊

- 虛擬私有雲 (VPC) 中的亞馬遜彈性運算雲 (Amazon EC2) 執行個體
- Amazon Elastic Block Store (Amazon EBS)
- Python 及其所需的軟件包， JavaScript 對象符號（JSON），系統和日期時間
- ASCII 平面檔案可供現代應用程式讀取或載入關聯式資料庫表格

目標架構

架構圖顯示了在 EC2 實例上將 EBCDIC 文件轉換為 ASCII 文件的過程：

1. 使用 parse_copybook_to_json.py 指令碼，您可以將 COBOL 字帖轉換為 JSON 檔案。
2. 您可以使用 JSON 檔案和 extract_ebcdic_to_ascii.py 指令碼，將 EBCDIC 資料轉換為 ASCII 檔案。

自動化和規模

完成第一次手動檔案轉換所需的資源之後，您就可以自動化檔案轉換。此模式不包含自動化指示。有多種方法可以自動化轉換。以下是一種可能方法的概述：

1. 將 AWS Command Line Interface (AWS CLI) (AWS CLI) 和 Python 指令碼命令封裝到殼層指令碼中。

2. 建立 AWS Lambda 函數，以非同步方式將殼層指令碼任務提交至 EC2 執行個體。如需詳細資訊，請參閱使用 AWS Lambda 排定安全殼層任務。
3. 建立 Amazon Simple Storage Service (Amazon S3) 觸發程序，以便在每次上傳舊版檔案時呼叫 Lambda 函數。如需詳細資訊，請參閱使用 Amazon S3 觸發器叫用 Lambda 函數。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以根據需要啟動任意數量的虛擬伺服器，並快速擴展或縮減它們。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

其他工具

- GitHub是一種代碼託管服務，提供協作工具和版本控制。
- Python 是一種高級別的編程語言。

代碼存儲庫

此模式的代碼可在mainframe-data-utilities GitHub 存儲庫中找到。

史诗

準備 EC2 執行個體

任務	描述	所需技能
啟動 EC2 執行個體。	EC2 執行個體必須具有輸出網際網路存取權。這可讓執行個體存取上可用的 Python 原始程	一般 AWS

任務	描述	所需技能
	式碼 GitHub。若要建立執行個體： 1. 在以下位置打開 Amazon EC2 控制台 https://console.aws.amazon.com/ec2。 2. 啟動一個 EC2 執行個體。使用公用 IP 位址，並允許透過連接埠 22 進行輸入存取。請確定執行個體的儲存大小至少是 EBCDIC 資料檔案大小的兩倍。如需指示，請參閱 Amazon EC2 文件。	
安裝 Git。	1. 使用安全殼層 (SSH) 用戶端，連接到剛啟動的 EC2 執行個體。如需詳細資訊，請參閱 Connect 到 Linux 執行個體。 2. 在 Amazon EC2 主控台中，執行下列命令。這會在 EC2 執行個體上安裝 Git。 `sudo yum install git` 3. 運行以下命令並確認 Git 已成功安裝。 `git --version`	一般 AWS

任務	描述	所需技能
安裝 Python。	1. 在 Amazon EC2 主控台中，執行下列命令。這會在 EC2 執行個體上安裝 Python。 `sudo yum install python3` 2. 在 Amazon EC2 主控台中，執行下列命令。這會在 EC2 執行個體上安裝 Pipp3。 `sudo yum install python3-pip` 3. 在 Amazon EC2 主控台中，執行下列命令。這會在 EC2 執行個體上安裝適用於 Python 的 AWS 開發套件 (Boto3)。 `sudo pip3 install boto3` 4. 在 Amazon EC2 主控台中，執行下列命令，AWS 區域的程式碼在哪裡<us-east-1> 。如需區域代碼的完整清單，請參閱 Amazon EC2 文件中的可用區域。 `export AWS_DEFAULT_REGION=<us-east-1>`	一般 AWS

任務	描述	所需技能
克隆存 GitHub 儲庫。	1. 在 Amazon EC2 主控台中，執行下列命令。這會複製mainframe-data-utilities儲存庫， GitHub 並開啟預設的複製位置，即home資料夾。 `git clone https://github.com/aws-samples/mainframe-data-utilities.git` 2. 在資home料夾中，確認資`mainframe-data-utilities` 料夾存在。	一般 AWS、 GitHub

從電子數據創建 ASCII 文件

任務	描述	所需技能
將 COBOL 字帖解析為 JSON 佈局文件。	在`mainframe-data-utilities` 資料夾內，執行 parse_copybook_to_json.py 指令碼。該自動化模塊從 COBOL 字帖讀取文件佈局並創建一個 JSON 文件。JSON 文件包含解釋和從源文件中提取數據所需的信息。這將創建從 COBOL 字帖的 JSON 元數據。 下面的命令 COBOL 字帖轉換為 JSON 文件。 `python3 parse_copybook_to_json.py \`	一般 AWS

任務	描述	所需技能

```
-copybook LegacyRef
erence/COBPACK2.cpy \
-output sample-data/
cobpack2-list.json \
-dict sample-data/
cobpack2-dict.json \
-ebcdic sample-data/
COBPACK.OUTFILE.txt \
-ascii sample-data/
COBPACK.ASCII.txt \
-print 10000
```

該腳本打印接收到的參數。

```
------------------
------------------
------------------
-----------------
Copybook file.....
.........| LegacyRef
erence/COBPACK2.cpy
Parsed copybook (JSON
 List).| sample-data/
cobpack2-list.json
JSON Dict (document
ation)...| sample-da
ta/cobpack2-dict.json
ASCII file.....
............| sample-
data/COBPACK.ASCII.t
xt
EBCDIC file.....
...........| sample-
data/COBPACK.OUTFILE
.txt
Print each.....
............| 10000
---------------------
------------------
```

任務	描述	所需技能
	`` ------------------ -------------- ``` 如需有關引數的詳細資訊，請參閱 GitHub 存放庫中的 README 檔案。	

任務	描述	所需技能
檢查 JSON 版面配置檔案。	1. 導覽至 parse_copybook_to_json.py 指令碼中定義的輸出路徑。 2. 檢查範例資料/cobpack 2-清單 .json 檔案的建立時間，以確認您已選取適當的 JSON 版面配置檔案。 3. 檢查 JSON 檔案，並確認內容與下列內容類似。 `"input": "extract-ebcdic-to-ascii/COBPACK.OUTFILE.txt", "output": "extract-ebcdic-to-ascii/COBPACK.ASCII.txt", "max": 0, "skip": 0, "print": 10000, "lrecl": 150, "rem-low-values": true, "separator": "|", "transf": [ { "type": "ch", "bytes": 19, "name": "OUTFILE-TEXT" }` JSON 佈局文件最重要的屬性是： • input— 包含要轉換的 EBCDIC 檔案的路徑	一般 AWS、JSON

任務	描述	所需技能
	• `output`— 定義要產生 ASCII 檔案的路徑 • `lrecl`— 以位元組為單位指定邏輯記錄長度的大小 • `transf`-列出所有字段及其大小（以字節為單位） 如需 JSON 版面配置檔案的詳細資訊，請參閱 GitHub 存放庫中的 README 檔案。	

任務	描述	所需技能
建立 ASCII 檔案。	執行 extract_ebcdic_to_ascii.py 指令碼，該指令碼包含在複製的 GitHub 儲存庫中。此指令碼會讀取 EBCDIC 檔案，並寫入已轉換且可讀的 ASCII 檔案。 `python3 extract_ebcdic_to_ascii.py -local-json sample-data/cobpack2-list.json` 當腳本處理 EBCDIC 數據時，它會為每批 10,000 條記錄打印一條消息。請參閱以下範例。 `---------- 2023-05-15 21:21:46.322253 | Local Json file | -local-json | sample-data/cobpack2-list.json 2023-05-15 21:21:47.034556 | Records processed | 10000 2023-05-15 21:21:47.736434 | Records processed | 20000 2023-05-15 21:21:48.441696 | Records processed | 30000 2023-05-15 21:21:49.173781 | Records processed | 40000`	一般 AWS

任務	描述	所需技能
	2023-05-15 21:21:49. 874779 | Records processed | 50000 2023-05-15 21:21:50. 705873 | Records processed | 60000 2023-05-15 21:21:51. 609335 | Records processed | 70000 2023-05-15 21:21:52. 292989 | Records processed | 80000 2023-05-15 21:21:52. 938366 | Records processed | 89280 2023-05-15 21:21:52. 938448 Seconds 6.616232 有關如何變更列印頻率的詳細資訊，請參閱 GitHub 存放庫中的 README 檔案。	

任務	描述	所需技能
檢查 ASCII 檔案。	1. 檢查 extract-ebcdic-to-ascii/COBPack.ascii.txt 檔案的建立時間，以確認該檔案是否最近建立。 2. 在 Amazon EC2 控制台中，輸入以下命令。這將打開 ASCII 文件的第一條記錄。 `head sample-data/COBPACK.ASCII.txt -n 1| xxd` 3. 檢查第一條記錄的內容。由於 EBCDIC 文件通常是二進制文件，因此它們沒有回車和換行符（CRLF）特殊字符。extract_ebcdic_to_ascii.py 指令碼會新增直線字元做為資料行分隔符號，這是在指令碼參數中定義的。 如果您使用提供的範例 EBCDIC 檔案，則以下是 ASCII 檔案中的第一條記錄。 `00000000: 2d30 3030 3030 3030 3030 3130 3030 3030  -000000000100000` `00000010: 3030 307c 3030 3030 3030 3030 3031 3030  000|0000000100` `00000020: 3030 3030 3030 7c2d 3030 3030 3030 3030  000000|-0000000`	一般 AWS

任務	描述	所需技能
	00000030: 3031 3030 3030 3030 3030 7c30 7c30 7c31 0100000000| 0|0|1 00000040: 3030 3030 3030 3030 7c2d 3130 3030 3030 00000000| -100000 00000050: 3030 307c 3130 3030 3030 3030 307c 2d31 000|10000 0000|-1 00000060: 3030 3030 3030 3030 7c30 3030 3030 7c30 00000000| 00000|0 00000070: 3030 3030 7c31 3030 3030 3030 3030 7c2d 0000|1000 00000|- 00000080: 3130 3030 3030 3030 307c 3030 3030 3030 100000000| 000000 00000090: 3030 3030 3130 3030 3030 3030 307c 2d30 000010000 0000|-0 000000a0: 3030 3030 3030 3030 3031 3030 3030 3030 000000000 1000000 000000b0: 3030 7c41 7c41 7c0a 00|A|A|.	

任務	描述	所需技能
評估 EBCDIC 檔案。	在 Amazon EC2 控制台中，輸入以下命令。這會開啟 EBCDIC 檔案的第一筆記錄。 `head sample-data/COBPACK.OUTFILE.txt -c 150 | xxd` 如果您使用範例 EBCDIC 檔案，則結果如下。 `00000000: 60f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 f0f0 `........ .... 00000010: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 f0f0 .... 00000020: f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f0f0 f1f0 .... 00000030: f0f0 f0f0 f0f0 d000 0000 0005 f5e1 00fa .... 00000040: 0a1f 0000 0000 0005 f5e1 00ff ffff fffa .... 00000050: 0a1f 0000 000f 0000 0c10 0000 000f 1000 .... 00000060: 0000 0d00 0000 0000 1000 0000 0f00 0000 ....`	一般 AWS, Linux, 電子證券

任務	描述	所需技能
	00000070: 0000 1000 0000 0dc1 c100 0000 0000 0000 .... 00000080: 0000 0000 0000 0000 0000 0000 0000 0000 .... 00000090: 0000 0000 0000	
	要評估源文件和目標文件之間的等價性，需要有關 EBCDIC 的全面知識。例如，範例 EBCDIC 檔案的第一個字元是連字號 ()。-在 EBCDIC 文件的十六進制表示法中，此字符由 60 ASCII 文件的十六進制表示法表示，此字符由 表示。2D如需電子碼轉換為 ASCII 的轉換表格，請參閱 IBM 網站上的 EBCDIC 轉換為 ASCII 碼。	

相關資源

參考

- 光電中心字元集 (IBM 說明文件)
- 電子證券轉換為 ASCII 碼 (IBM 文件)
- 聯盟 (IBM 文件集)
- 基本的 JCL 概念 (IBM 文檔)
- Connect 到您的 Linux 執行個體 (Amazon EC2 文件)

教學課程

- 使用 AWS Lambda 排程安全殼層任務 (AWS 部落格文章)
- 使用 Amazon S3 觸發程序來叫用 Lambda 函數 (AWS Lambda 文件)

使用 AWS Lambda 在 Amazon S3 中將大型主機檔案從 EBCDIC 格式轉換為以字元分隔的 ASCII 格式

創建者：路易斯古斯塔沃丹達斯 (AWS)

程式碼儲存庫:大型主機資料公用程式	環境：PoC 或試點	資料來源：IBM 光電中心檔案
目標：分隔的 ASCII 檔案	R 類型：重新平台	工作負載：IBM
技術：大型主機	AWS 服務：AWS CloudShell; AWS Lambda; Amazon S3; Amazon CloudWatch	

Summary

此模式說明如何啟動 AWS Lambda 函數，該函數會自動將大型主機 EBCDIC (延伸二進位編碼的十進位交換碼) 檔案轉換為以字元分隔的 ASCII (美國資訊交換標準碼) 檔案。Lambda 函數會在 ASCII 檔案上傳至 Amazon 簡單儲存服務 (亞馬遜 S3) 儲存貯體之後執行。檔案轉換後，您可以讀取 x86 工作負載上的 ASCII 檔案，或將檔案載入到現代資料庫中。

此模式中展示的檔案轉換方法可協助您克服在現代環境中使用 EBCDIC 檔案的挑戰。以 EBCDIC 編碼的文件通常包含以二進制或打包十進制格式表示的數據，並且字段是固定長度。這些特性會造成障礙，因為現代 x86 型工作負載或分散式環境通常會使用 ASCII 編碼的資料，而且無法處理 EBCDIC 檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- S3 儲存貯體
- 具有管理許可的 AWS Identity and Access Management (IAM) 使用者
- AWS CloudShell
- Python 3.8.0 或更高版本
- 以 EBCDIC 編碼的平面檔案及其對應的資料結構，採用一般業務導向語言 (COBOL) 字帖

注意：此模式使用範例 EBCDIC 檔案 (用戶端 .txt) 及其對應的 COBOL 字帖 (COBKS05 .c py)。這兩個檔案都可以在 GitHub mainframe-data-utilities儲存庫中使用。

限制

- COBOL 字帖通常保存多個佈局定義。該mainframe-data-utilities項目可以解析這種字帖，但無法推斷數據轉換時要考慮哪種佈局。這是因為抄本不持有這種邏輯（而是保留在 COBOL 程序上）。因此，您必須在剖析文字本之後，手動設定選取版面的規則。
- 此模式受 Lambda 配額限制。

架構

源, 技術, 堆棧

- IBM z/OS、IBM i 和其他光電控系統
- 含有以 EBCDIC 編碼資料的循序檔案 (例如 IBM Db2 卸載)
- 聯邦字帖

目標技術堆疊

- Amazon S3
- Amazon S3 事件通知
- IAM
- Lambda 函數
- Python 3.8 或更高版本
- 大型主機資料公用程式
- 中繼資料
- 以字元分隔的 ASCII 檔案

目標架構

下圖顯示了將大型主機 EBCDIC 檔案轉換為 ASCII 檔案的架構。

該圖顯示以下工作流程：

1. 用戶運行字帖解析器腳本將 COBOL 字帖轉換為 JSON 文件。
2. 使用者將 JSON 中繼資料上傳到 S3 儲存貯體。這可讓資料轉換 Lambda 函數讀取中繼資料。
3. 使用者或自動化程序會將 EBCDIC 檔案上傳到 S3 儲存貯體。
4. S3 通知事件會觸發資料轉換 Lambda 函數。
5. AWS 會驗證 Lambda 函數的 S3 儲存貯體讀寫許可。
6. Lambda 會從 S3 儲存貯體讀取檔案，並在本機將檔案從 EBCDIC 轉換為 ASCII 碼。
7. Lambda 在 Amazon 中記錄流程狀態 CloudWatch。
8. Lambda 將 ASCII 文件寫回 Amazon S3。

注意：在將中繼資料轉換為 JSON，然後將該資料上傳到 S3 儲存貯體之後，字帖剖析器指令碼只會執行一次。初始轉換之後，任何使用上傳至 S3 儲存貯體的相同 JSON 檔案的 EBCDIC 檔案將使用相同的中繼資料。

工具

AWS 工具

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS CloudShell 是以瀏覽器為基礎的殼層，您可以使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 和一系列預先安裝的開發工具來管理 AWS 服務。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。Lambda 只會在需要時執行程式碼並自動調整規模，因此您只需按使用的運算時間付費。

其他工具

- GitHub是一種代碼託管服務，提供協作工具和版本控制。
- Python 是一種高級別的編程語言。

Code

此模式的代碼可在 GitHub mainframe-data-utilities存儲庫中找到。

最佳實務

請考慮下列最佳作法：

- 在 Amazon 資源名稱 (ARN) 級別設置所需的許可。
- 一律授與 IAM 政策的最低權限許可。如需詳細資訊，請參閱 IAM 文件中的 IAM 中的安全性最佳實務。

史诗

建立環境變數和工作資料夾

任務	描述	所需技能
建立環境變數。	將下列環境變數複製到文字編輯器，然後將<placeholder>下列範例中的值取代為您的資源值： `bucket=<your_bucket_name>` `account=<your_account_number>` `region=<your_region_code>` 注意：稍後您將建立 S3 儲存貯體、AWS 帳戶和 AWS 區域的參考。 若要定義環境變數，請開啟CloudShell 主控台，然後將更新的環境變數複製並貼到命令列上。 注意：每次 CloudShell 工作階段重新啟動時，您都必須重複此步驟。	一般 AWS

任務	描述	所需技能
建立工作資料夾。	若要簡化稍後的資源清理程序，請執行下列指令 CloudShell 來在中建立工作資料夾： `mkdir workdir; cd workdir` 附註：每次失去工作 CloudShell 階段的連線時，都必須將目錄變更為工作目錄 (`workdir`)。	一般 AWS

定義 IAM 角色和政策

任務	描述	所需技能
為 Lambda 函數建立信任政策。	EBCDIC 轉換器在一個 Lambda 函數中運行。該函數必須具有 IAM 角色。在建立 IAM 角色之前，您必須定義可讓資源採用該政策的信任政策文件。 從 CloudShell 工作資料夾中，執行以下指令來建立原則文件： `E2ATrustPol=$(cat <<EOF` `{` `  "Version": "2012-10-17",` `  "Statement": [` `  {` `  "Effect": "Allow",`	一般 AWS

任務	描述	所需技能
	<pre>"Principa l": { "Service": "lambda.a mazonaws.com" }, "Action": "sts:AssumeRole" }] } EOF) printf "$E2ATrustPol" > E2ATrustPol.json</pre>	
建立用於 Lambda 轉換的 IAM 角色。	若要建立 IAM 角色，請從 CloudShell 工作資料夾執行以下 AWS CLI 命令： <pre>aws iam create-role --role-name E2AConvLa mbdaRole --assume- role-policy-docume nt file://E2ATrustPol .json</pre>	一般 AWS

任務	描述	所需技能
為 Lambda 函數建立 IAM 政策文件。	Lambda 函數必須具有 S3 儲存貯體的讀寫存取權限，以及 Amazon CloudWatch 日誌的寫入許可。 若要建立 IAM 政策，請從 CloudShell 工作資料夾執行下列命令： `E2APolicy=$(cat <<EOF` `{` `"Version": "2012-10-17",` `"Statement": [` `{` `"Sid": "Logs",` `"Effect": "Allow",` `"Action": [` `"logs:PutLogEvents",` `"logs:CreateLogStream",` `"logs:CreateLogGroup"` `],` `"Resource": [` `"arn:aws:logs:*:*:log-group:*",` `"arn:aws:logs:*:*:log-group:*:log-stream:*"` `]` `},` `{`	一般 AWS

任務	描述	所需技能
	<pre>"Sid": "S3", "Effect": "Allow", "Action": ["s3:GetObject", "s3:PutObject", "s3:GetObjectVersion"], "Resource": ["arn:aws:s3:::%s/*", "arn:aws:s3:::%s"] }] } EOF) printf "$E2APolicy" "$bucket" "$bucket" > E2AConvLambdaPolic y.json</pre>	
將 IAM 政策文件附加到 IAM 角色。	若要將 IAM 政策附加到 IAM 角色，請從 CloudShell 工作資料夾執行下列命令： <pre>aws iam put-role-policy --role-name E2AConvLa mbdaRole --policy-name E2AConvLambdaPolic y --policy-document file://E2AConvLamb daPolicy.json</pre>	一般 AWS

建立用於 EBCDIC 轉換的 Lambda 函數

任務	描述	所需技能
下載 EBCDIC 轉換源代碼。	從 CloudShell 工作資料夾中，執行下列指令以從中下載 mainframe-data-utilities 原始程式碼 GitHub： `git clone https://github.com/aws-samples/mainframe-data-utilities.git mdu`	一般 AWS
建立 ZIP 封裝。	從 CloudShell 工作資料夾中，執行下列命令以建立用於 EBCDIC 轉換的 Lambda 函數的 ZIP 套件： `cd mdu; zip ../mdu.zip *.py; cd ..`	一般 AWS
建立 Lambda 函數。	從 CloudShell 工作資料夾中，執行下列命令以建立用於 EBCDIC 轉換的 Lambda 函數： `aws lambda create-function \` `--function-name E2A \` `--runtime python3.9 \` `--zip-file fileb://mdu.zip \` `--handler extract_ebcdic_to_ascii.lambda_handler \` `--role arn:aws:iam::$account:role/E2AConvLambdaRole \`	一般 AWS

任務	描述	所需技能
	`--timeout 10 \` `--environment "Variable s={layout=$bucket/ layout/}"` 附註：環境變數配置會告知 Lambda 函數 JSON 中繼資料所在的位置。	
為 Lambda 函數建立以資源為基礎的政策。	從 CloudShell 工作資料夾執行下列命令，以允許 Amazon S3 事件通知觸發 Lambda 函數進行 EBCDIC 轉換： `aws lambda add-permi ssion \` `--function-name E2A \` `--action lambda:In vokeFunction \` `--principal s3.amazon aws.com \` `--source-arn arn:aws:s 3:::$bucket \` `--source-account $account \` `--statement-id 1`	一般 AWS

建立 Amazon S3 事件通知

任務	描述	所需技能
建立 Amazon S3 事件通知的組態文件。	當檔案放置在輸入資料夾中時，Amazon S3 事件通知會啟動 EBCDIC 轉換 Lambda 函數。	一般 AWS

任務	描述	所需技能
	在工 CloudShell 作資料夾中，執行下列命令，為 Amazon S3 事件通知建立 JSON 文件： `{ "LambdaFunctionConfigurations": [ { "Id": "E2A", "LambdaFunctionArn": "arn:aws:lambda:%s:%s:function:E2A", "Events": [ "s3:ObjectCreated:Put" ], "Filter": { "Key": { "FilterRules": [ { "Name": "prefix", "Value": "input/" } ] } } } ] } EOF )` `printf "$S3E2AEvent" "$region" "$account" > S3E2AEvent.json`	

任務	描述	所需技能
建立 Amazon S3 事件通知。	從 CloudShell 工作資料夾執行下列命令以建立 Amazon S3 事件通知： `aws s3api put-bucket-notification-configuration --bucket $bucket --notification-configuration file://S3E2AEvent.json`	一般 AWS

建立並上傳 JSON 中繼資料

任務	描述	所需技能
解析 COBOL 字帖。	從 CloudShell 工作資料夾執行下列命令，將 COBOL 範例字稿剖析為 JSON 檔案 (定義如何正確讀取和分割資料檔案)： `python3 mdu/parse_copybook_to_json.py \` `-copybook mdu/LegacyReference/COBKS05.cpy \` `-output CLIENT.json \` `-output-s3key CLIENT.ASCII.txt \` `-output-s3bkt $bucket \` `-output-type s3 \` `-print 25`	一般 AWS
新增轉換規則。	樣本數據文件及其相應的 COBOL 字帖是一個多佈局文	一般 AWS、IBM 大型主機、科博爾

任務	描述	所需技能
	件。這意味著轉換必須根據特定規則切片數據。在這種情況下，每行中位置 3 和 4 上的字節定義佈局。 從 CloudShell 工作資料夾中，編輯CLIENT.json 檔案並將內容從變更"transf-rule": [], 為以下內容： `"transf-rule": [` `{` `"offset": 4,` `"size": 2,` `"hex": "0002",` `"transf": "transf1"` `},` `{` `"offset": 4,` `"size": 2,` `"hex": "0000",` `"transf": "transf2"` `}` `],`	
將 JSON 中繼資料上傳至 S3 儲存貯體。	從 CloudShell 工作資料夾執行下列 AWS CLI 命令，將 JSON 中繼資料上傳到 S3 儲存貯體： `aws s3 cp CLIENT.json` `s3://$bucket/layout/` `CLIENT.json`	一般 AWS

轉換文件

任務	描述	所需技能
將 EBCDIC 檔案傳送到 S3 儲存貯體。	從 CloudShell 工作資料夾執行下列命令，將 EBCDIC 檔案傳送至 S3 儲存貯體： `aws s3 cp mdu/sample-data/CLIENT.EBCDIC.txt s3://$bucket/input/` 備註：建議您為輸入 (EBCDIC) 和輸出 (ASCII) 檔案設定不同的資料夾，以避免在 ASCII 檔案上傳至 S3 儲存貯體時再次呼叫 Lambda 轉換函數。	一般 AWS
檢查輸出。	從 CloudShell 工作資料夾中，執行下列命令以檢查是否在 S3 儲存貯體中產生 ASCII 檔案： `awss3 ls s3://$bucket/` 備註：數據轉換可能需要幾秒鐘的時間才能發生。我們建議您多次檢查 ASCII 檔案。 ASCII 檔案可用之後，執行下列命令，將檔案從 S3 儲存貯體下載到目前的資料夾： `aws s3 cp s3://$bucket/CLIENT.ASCII.txt .` 檢查 ASCII 文件內容：	一般 AWS

任務	描述	所需技能
	`head CLIENT.ASCII.txt`	

清潔環境

任務	描述	所需技能
(選擇性) 準備變數和資料夾。	如果失去與的連線 CloudShell，請重新連線，然後執行以下指令，將目錄變更為工作資料夾： `cd workdir` 請確定已定義環境變數： `bucket=<your_bucket_name>` `account=<your_account_number>` `region=<your_region_code>`	一般 AWS
移除值區的通知設定。	從 CloudShell 工作資料夾執行下列命令以移除 Amazon S3 事件通知組態： `aws s3api put-bucket-notification-configuration \` `--bucket=$bucket \` `--notification-configuration="{}"`	一般 AWS

任務	描述	所需技能
刪除 Lambda 函數。	從 CloudShell 工作資料夾中，執行下列命令以刪除 EBCDIC 轉換器的 Lambda 函數： `awslambdadelete-function--function-nameE2A`	一般 AWS
刪除 IAM 角色和政策。	從 CloudShell 工作資料夾中，執行下列命令以移除 EBCDIC 轉換器角色和原則： `aws iam delete-role-policy --role-name E2AConvLambdaRole --policy-name E2AConvLambdaPolicy` `aws iam delete-role --role-name E2AConvLambdaRole`	一般 AWS
刪除 S3 儲存貯體中產生的檔案。	從 CloudShell 工作資料夾中，執行下列命令以刪除 S3 儲存貯體中產生的檔案： `aws s3 rm s3://$bucket/layout --recursive` `aws s3 rm s3://$bucket/input --recursive` `aws s3 rm s3://$bucket/CLIENT.ASCII.txt`	一般 AWS

任務	描述	所需技能
刪除工作資料夾。	從 CloudShell 工作資料夾中，執行以下指令以移除workdir及其內容： `cd ..; rm -Rf workdir`	一般 AWS

相關資源

- 大型主機數據實用程序自述文件 ( ) GitHub
- 光電中心字元集 (IBM 說明文件)
- 電子證券轉換為 ASCII 碼 (IBM 文件)
- 聯盟 (IBM 文件集)
- 使用 Amazon S3 觸發程序來叫用 Lambda 函數 (AWS Lambda 文件)

使用 Micro Focus 轉換具有複雜記錄佈局的大型主機資料檔案

由彼得·韋斯特創作

環境：生產	來源：大型主機 EBCDIC 數據文件	目標：微焦點 ASCII 數據文件
R 類型：重新主機	工作負載：所有其他工作	技術：大型主機；現代化
AWS 服務：AWS 大型主機現代化		

Summary

此模式說明如何使用 Micro Focus 結構檔案，將包含非文字資料和複雜記錄配置的大型主機資料檔案從 EBCDIC (延伸二進位編碼的十進位交換碼) 字元編碼轉換為 ASCII (美國資訊交換標準碼) 字元編碼。若要完成檔案轉換，您必須執行下列動作：

1. 準備單一來源檔案，以說明大型主機環境中的所有資料項目和記錄配置。
2. 使用 Micro Focus 資料檔案編輯器作為 Micro Focus 經典資料檔案工具或資料檔案工具的一部分，建立包含資料記錄配置的結構檔案。結構檔案會識別非文字資料，以便您可以正確地將大型主機檔案從 EBCDIC 轉換為 ASCII。
3. 使用「傳統資料檔工具」或「資料檔案工具」來測試結構檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 適用於 Windows 的微型焦點企業開發人員，可透過 AWS 大型主機現代化取得

產品版本

- 微焦點企業伺服器 7.0 及更新版本

工具

- Micro Focus 企業開發人員為使用企業開發人員的任何整合式開發環境 (IDE) 變體所建立的應用程式提供執行環境。
- Micro Focus 經典資料檔案工具可協助您轉換、瀏覽、編輯和建立資料檔案。經典數據文件工具包括數據文件轉換器，記錄佈局編輯器和數據文件編輯器。
- 微型焦點資料檔案工具可協助您建立、編輯和移動資料檔案。數據文件工具包括數據文件編輯器，文件轉換實用程序和數據文件結構命令行實用程序。

史诗

準備來源檔

任務	描述	所需技能
識別來源元件。	識別檔案的所有可能記錄配置，包括任何包含非文字資料的重新定義。 如果您有包含重新定義的佈局，則必須將這些佈局分解為描述數據結構的每個可能排列的唯一佈局。一般而言，資料檔案的記錄版面配置可以用下列原型描述： • 僅使用文本數據記錄佈局 • 使用非文本數據記錄佈局 • 具有從屬於 RELIDISE 子句的非文本數據的記錄佈局 如需針對包含複雜記錄配置的檔案建立扁平化記錄配置的詳細資訊，請參閱針對大型主機移轉的 ASCII 環境重新裝載 EBCDIC 應用程式。	應用程式開發人員

任務	描述	所需技能
識別記錄版面配置條件。	對於具有多個記錄版面配置的檔案或包含具有 REDELIDISE 子句之複雜配置的檔案，識別記錄內的資料和條件，您可以使用這些資料和條件來定義轉換期間要使用的配置。我們建議您與了解處理這些檔案之程式的主題專家 (SME) 討論此工作。 例如，檔案可能包含兩種包含非文字資料的記錄類型。您可以檢查源代碼，並可能找到類似以下內容的代碼： `MOVE "M" TO PART-TYPE` `MOVE "MAIN ASSEMBLY" TO` `PART-NAME` `MOVE "S" TO PART-TYPE` `MOVE "SUB ASSEMBLY 1"` `TO PART-NAME` 該代碼可幫助您識別以下內容： • 「部分類型」字段用於確定記錄類型 • 值「M」用於「M 部分記錄」 • 值「S」用於「S-部分記錄」 您可以記錄此欄位所使用的值，以將記錄版面配置與檔案中正確的資料記錄相關聯。	應用程式開發人員

任務	描述	所需技能
建置來源檔案。	如果在多個來源檔案上描述檔案，或者如果記錄配置包含從屬於 REDELIDES 子句的非文字資料，則建立包含記錄版面配置的新來源檔案。新的程序並不需要使用 SELECT 和 FD 語句來描述文件。該程序可以簡單地將記錄描述包含為 01 工作-存儲中的級別。 附註：您可以為每個資料檔建立來源檔案，或建立描述所有資料檔的主要來源檔案。	應用程式開發人員
編譯源文件。	編譯來源檔案以建置資料字典。我們建議您使用 EBCDIC 字元集來編譯來源檔案。如果正在使用 IBMCOMP 指令或 ODOSLIDE 指令，那麼您也必須在源文件中使用這些指令。 注意：IBMCOMP 影響 COMP 字段的字節存儲和 ODOSLIDE 影響填充發生變化的結構。如果這些指令設置不正確，則轉換工具將無法正確讀取數據記錄。這會導致轉換後的文件中的數據不正確。	應用程式開發人員

（選項 A）使用經典數據文件工具創建結構文件

任務	描述	所需技能
啟動工具並載入字典。	1. 選擇 Windows [開始] 功能表圖示，搜尋並選擇 Micro	應用程式開發人員

任務	描述	所需技能
	Focus 企業開發人員，然後選擇 [傳統資料檔案工具]。 2. 選擇檔案，然後選擇錄製版面配置。 3. 在「選取要從中建構配置圖的檔案」對話方塊中，對於「檔案名稱」，選取您先前編譯來源檔案時所建立的 IDY (`.idy`) 檔案。然後選擇 Open (開啟)。 4. 若要確認「傳統資料檔工具」使用 EBCDIC，請在「資料檔工具」對話方塊中，如果 IDY 檔案設定為 EBCDIC 且「資料圖」設定為 AN SI，請在「資料檔工具」對話方塊中選擇「是」。	

任務	描述	所需技能
建立預設記錄版面配置。	對不符合任何條件式版面配置的所有記錄使用預設記錄版面配置。 1. 在「版面配置」視窗中，展開資料結構，然後找出用於預設版面配置的 01 層級。 2. 在 01 項目上按一下滑鼠右鍵，然後選擇「新配置」。 3. 在 [新增記錄配置精靈] 對話方塊中，選擇 [預設配置]，然後選擇 [下一步]。 4. 選擇 Finish (完成)。 默認布局顯示在「布局」窗格中，您可以通過紅色文件夾圖標標識。	應用程式開發人員

任務	描述	所需技能
建立條件式記錄版面配置。	當檔案中有多個記錄配置時，請使用條件式記錄配置。 1. 在「版面配置」窗格中，展開資料結構，然後找出用於條件式版面配置的 01 層級。 2. 在 01 項目上按一下滑鼠右鍵，然後選擇「新配置」。 3. 在 [新增記錄配置精靈] 對話方塊中，選擇 [條件式配置]，然後選擇 [下一步]。 4. 選擇 Finish (完成)。條件版面配置會顯示在「版面配置」窗格中，並且可以透過黃色資料夾圖示來識別。 5. 展開條件版面配置，在必須放置條件的欄位上按一下滑鼠右鍵，然後選擇「內容」。 6. 在「欄位性質」對話方塊中，輸入條件。確認字元集已設定為 EBCDIC，然後選擇 [確定]。 已設定條件的欄位旁邊會出現核取記號。 7. 對於需要此版面配置條件的任何其他欄位重複步驟 5 到 6。 8. 對必須新增的任何其他條件式版面重複步驟 1—6。 9. 選擇 [檔案]，選擇 [另存新檔]，然後將結構檔案儲存至磁碟。	應用程式開發人員

（選項 B）使用數據文件工具創建結構文件

任務	描述	所需技能
啟動工具並載入字典。	1. 選擇 Windows [開始] 功能表圖示，搜尋並選擇 Micro Focus 企業開發人員，然後選擇 [資料檔案工具]。 2. 選擇檔案、新增、結構檔案。 3. 在「開啟」對話方塊中，對於「檔案名稱」，選取先前編譯來源檔案時所建立的 IDY (`.idy`) 檔案。然後選擇 Open (開啟)。 4. 若要確認資料檔工具使用 EBCDIC，請確認 [除錯檔案] 區段中的下拉式功能表已設定為 EBCDIC。	應用程式開發人員
建立預設記錄版面配置。	對於不符合任何條件式版面配置的所有記錄，請使用預設記錄版面配置。 1. 在左窗格的 [可用配置圖] 區段中，展開資料結構，然後找出用於預設配置圖的 01 層級。 2. 在 01 項目上按一下右鍵，然後選擇「建立預設配置」。 預設版面配置會顯示在「版面配置」窗格中，並可透過藍色的「D」圖示來識別。	應用程式開發人員

任務	描述	所需技能
建立條件式記錄版面配置。	當檔案中有多個記錄配置時，請使用條件式記錄配置。 1. 在右窗格的「選取的配置」區段中，展開資料結構，然後找出用於條件式版面配置的 01 層級。 2. 在 01 項目上按一下滑鼠右鍵，然後選擇「建立條件配置」 條件式版面配置會顯示在右側的「版面配置」窗格中，並可透過綠色的「C」圖示來識別。 3. 展開條件版面配置，在必須放置條件的欄位上按一下滑鼠右鍵，然後選擇「內容」。 4. 在「欄位性質」對話方塊中，輸入條件。確認字元集已設定為 EBCDIC，然後選擇 [確定]。 具有條件設定的欄位旁邊會出現紅色的「IF」圖示。 5. 對於需要此版面配置條件的任何其他欄位重複步驟 3 到 4。 6. 對必須新增的任何其他條件式版面重複步驟 1—4。 7. 選擇 [檔案]，選擇 [另存新檔]，然後將結構檔案儲存至磁碟。	應用程式開發人員

（選項 A）使用傳統數據文件工具測試結構文件

任務	描述	所需技能
測試 EBCDIC 資料檔案。	確認您可以使用結構檔正確檢視 EBCDIC 測試資料檔案。 1. 選擇 Windows [開始] 功能表圖示，尋找並選擇 Micro Focus 企業開發人員，然後選擇 [傳統資料工具]。 2. 選擇 [檔案]，然後選擇 [開啟]。 3. 在「開啟」對話方塊中，選取 EBCDIC 資料集做為「檔案名稱」，然後選擇「開啟」。 4. 選擇檔案、資料檔案編輯器、載入記錄配置。 5. 在「開啟舊檔」對話方塊中，選取結構檔案做為「檔案名稱」，然後選擇「開啟」。 6. 若要確認字元集模式設定為 EBCDIC，請確認下拉式功能表已設定為 EBCDIC。您可以在左窗格中看到原始記錄數據，在右窗格中看到格式化的數據。 7. 選擇各種記錄，以確保所有格式都以正確的佈局呈現。	應用程式開發人員

（選項 B）使用數據文件工具測試結構文件

任務	描述	所需技能
測試 EBCDIC 資料檔案。	確認您可以使用結構檔正確檢視 EBCDIC 測試資料檔案。 1. 選擇 Windows [開始] 功能表圖示，尋找並選取 Micro Focus 企業開發人員，然後選擇 [資料檔案工具]。 2. 選擇檔案、開啟舊檔、資料檔案。 3. 在「開啟資料檔」對話方塊的「本機」頁籤上，對於「檔案名稱」，選擇「瀏覽」以尋找 EBCDIC 測試檔案的位置。 4. 對於「結構檔案」(可選)，請選擇「瀏覽」以尋找結構檔案的位置。 5. 在「檔案詳細資訊」區段中，輸入檔案的詳細資訊，並確認「編碼」設定為 EBCD IC。 6. 根據您的需求，選擇「開放共用」或「開放獨佔」模式。 7. 確認工具列「外觀」區段中的下拉式功能表已設定為 EBCD IC。您將在左窗格中看到原始記錄數據，並在右窗格中看到格式化的數據。 8. 選擇各種記錄，以確保所有格式都以正確的佈局呈現。	應用程式開發人員

測試資料檔案轉換

任務	描述	所需技能
測試 EBCDIC 檔案的轉換。	1. 選擇 Windows [開始] 功能表圖示，尋找並選取 Micro Focus 企業開發人員，然後選擇 [傳統資料工具]。 2. 選擇 [工具]，然後選擇 [轉換]。 3. 在「資料檔轉換」對話方塊的「輸入檔案」區段中，對於「檔案名稱」，選擇「瀏覽」以尋找並選取 EBCDIC 輸入檔案。確認字元集設定為 EBCDIC。 4. 在「字元集轉換」區段中，選取「轉換字元集」和「包含非文字資料項目的記錄」核取方塊。選擇 「選取要轉換的版面配置」，然後選擇「瀏覽」以尋找並選取結構檔案。 5. 在「新檔案」區段中，對於「檔案名稱」，輸入您要建立的 ASCII 輸出檔案的路徑和檔名。依預設，轉換工具預設為與輸入檔案相同的格式。若要進行測試，請將選項設定保留為其預設值。 6. 選擇「轉換」。 7. 遵循 (選項 A) 使用傳統資料檔工具測試結構檔案或 (選項 B) 使用資料檔工具測試結構檔案一節中的步驟，但	應用程式開發人員

任務	描述	所需技能
	載入 ASCII 輸出檔案而不是 EBCDIC 檔案。 8. 將 EBCDIC 和 ASCII 檔案都載入資料檔案編輯器中，然後並排比較檔案以檢查轉換的準確性。	

相關資源

- 微焦點（微焦點文件）
- 大型主機和舊版程式碼 (AWS 部落格文章)
- AWS Prescriptive Guidance (AWS 文件)
- AWS 文件 (AWS 文件)
- AWS 一般參考資料 (AWS 文件)
- AWS 詞彙表 (AWS 文件)

使用 Terraform 為容器化的藍光時代應用程式部署環境

由理查德·米爾納瓦特（AWS）創建

代碼存儲庫：藍光時代示例 ECS 基礎設施（地形）	環境：生產	資料來源：大型機
目標：容器	R 類型：重新平台	工作負載：IBM；所有其他工作負載
技術：大型主機；容器與微服務	AWS 服務：Amazon ECS; AWS Step Functions; Amazon VPC; Amazon Aurora	

Summary

將舊式大型主機工作負載移轉至現代雲端架構可免除維護大型主機的成本 — 只會隨著環境老化而增加的成本。不過，從大型主機移轉工作可能會帶來獨特的挑戰。內部資源可能不熟悉工作邏輯，而且與商品、一般化 CPU 相比，在這些特殊工作中，大型主機的高效能可能很難複寫。重寫這些工作可能是一項龐大的任務，需要大量的努力。

Blu Age 將舊式大型主機工作負載轉換為現代 Java 程式碼，然後您可以將其當作容器執行。

此模式為執行已透過 Blu Age 工具現代化的容器化應用程式提供無伺服器架構範例。隨附的 HashiCorp Terraform 文件將為 Blu Age 容器的協調構建安全架構，同時支持批處理任務和實時服務。

如需使用 Blu Age 和 AWS 服務將工作負載現代化的詳細資訊，請參閱以下 AWS Prescriptive Guidance 出版物：

- 在 AWS 無伺服器基礎設施上執行已透過 Blu Age 現代化的大型主機工作負載
- 容器化已由 Blu Age 現代化的大型主機工作負載

如需使用 Blu Age 現代化大型主機工作負載的協助，請在 Blu Age 網站上選擇聯絡我們的專家，聯絡 Blu Age 團隊。如需將現代化工作負載遷移到 AWS、將它們與 AWS 服務整合以及將它們移入生產環境的協助，請聯絡您的 AWS 客戶經理或填寫 AWS Professional Services 表單。

先決條件和限制

先決條件

- 由容器化大型主機工作負載提供的容器化 Blu Age 應用程式範例，這些工作負載已透過 Blu Age 模式進行現代化。範例應用程式提供邏輯來處理現代化應用程式的輸入和輸出，並且可以與此架構整合。
- 部署這些資源需要地形。

限制

- 亞馬遜彈性容器服務 (Amazon ECS) 限制了可供容器使用的任務資源。這些資源包括 CPU、記憶體和儲存空間。例如，使用 Amazon ECS 搭配 AWS Fargate 時，會套用任務資源限制。

產品版本

此解決方案已通過以下版本進行測試：

- 地形
- 地形化 AWS 供應商 4.46.0

架構

源, 技術, 堆棧

- 藍色時代
- 地形

目標技術堆疊

- Amazon Aurora PostgreSQL-Compatible Edition
- AWS Backup
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon ECS
- AWS Identity and Access Management 服務 (IAM)
- AWS AWS KMS 鑰管理伺服器

- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Step Functions
- AWS Systems Manager

目標架構

下圖顯示了解決方案架構。

1. 此解決方案會部署下列 IAM 角色：
   - Batch 工作角色
   - Batch 工作執行角色
   - 服務工作角色
   - 服務作業執行角色
   - Step Functions 角色
   - AWS Backup 角色
   - RDS 增強型監控角色。

   這些角色符合最低權限的存取原則。
2. Amazon ECR 用於存放由此模式編排的容器映像。
3. AWS Systems Manager Parameter Store 會在執行時期將每個環境的組態資料提供給 Amazon ECS 任務定義。
4. AWS Secrets Manager 會在執行時期向 Amazon ECS 任務定義提供有關環境的敏感組態資料。資料已由 AWS KMS 加密。
5. Terraform 模組會針對所有即時和批次任務建立 Amazon ECS 任務定義。
6. Amazon ECS 使用 AWS Fargate 作為運算引擎來執行批次任務。這是一項短暫的任務，由 AWS Step Functions 的要求啟動。
7. 與 Amazon Aurora PostgreSQL 相容提供支援現代化應用程式的資料庫。這會取代大型主機資料庫，例如 IBM Db2 或 IBM IMS 資料庫。
8. Amazon ECS 執行長期服務，以提供現代化的即時工作負載。這些無狀態應用程式會在跨可用區域的容器中永久執行。

9. Network Load Balancer 可用來授與即時工作負載的存取權。Network Load Balancer 支援舊版通訊協定，例如 IBM CICS。或者，您可以將應用程式負載平衡器與 HTTP 型工作負載搭配使用。
10. Amazon S3 為任務輸入和輸出提供物件儲存。容器應處理 Amazon S3 的提取和推送操作，以準備 Blu Age 應用程式的工作目錄。
11. AWS Step Functions 服務用於協調執行 Amazon ECS 任務以處理批次工作負載。
12. 每個批次工作負載的 SNS 主題可用來整合現代化應用程式與其他系統 (例如電子郵件)，或啟動其他動作，例如將輸出物件從 Amazon S3 傳送到 FTP。

注意：根據預設，解決方案無法存取網際網路。此模式假設虛擬私有雲端 (VPC) 將使用 AWS Transit Gateway 等服務連接到其他網路。因此，會部署多個介面 VPC 端點，以授予解決方案所使用 AWS 服務的存取權。若要開啟直接網際網路存取，您可以使用 Terraform 模組中的切換開關，將 VPC 端點取代為網際網路閘道和關聯的資源。

自動化和規模

在此模式中使用無伺服器資源，有助於確保透過向外擴充，此設計的規模幾乎沒有限制。如此可減少吵雜的鄰居擔憂，例如在原始大型主機上可能會遇到的運算資源競爭。可以將 Batch 工作排程為視需要同時執行。

單個容器受 Fargate 支持的最大尺寸的限制。如需詳細資訊，請參閱 Amazon ECS 文件中的任務 CPU 和記憶體一節。

若要水平擴展即時工作負載，您可以新增容器。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- AWS Backup 是一種全受管服務，可協助您集中和自動化 AWS 服務、雲端和內部部署的資料保護。
- Amazon Elastic Container Registry (Amazon ECR) 是安全、可擴展且可靠的受管容器映像登錄服務。
- Amazon Elastic Container Service (Amazon ECS) 是快速、可擴展的容器管理服務，可協助您執行、停止和管理叢集上的容器。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。
- AWS Systems Manager Parameter Store 為組態資料管理和機密管理提供安全的階層式儲存。

其他服務

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。此模式使用 Terraform 來創建示例體系結構。

代碼存儲庫

此模式的源代碼可在 GitHub 藍光時代示例 ECS 基礎架構（Terraform）存儲庫中找到。

最佳實務

- 對於測試環境，請使用設定現代化應用程式的`forceDate`選項等功能，以便始終執行一段已知的時間段，以產生一致的測試結果。
- 單獨調整每個任務以消耗最佳資源量。您可以使用 Amazon CloudWatch 容器深入解析取得潛在瓶頸的指導。

史诗

準備要部署的環境

任務	描述	所需技能
複製解決方案原始程式碼。	從GitHub 專案複製解決方案程式碼。	DevOps 工程師

任務	描述	所需技能
透過部署資源來儲存 Terraform 狀態來啟動環境。	1. 開啟終端機視窗，並確認已安裝 Terraform，並確認 AWS 登入資料可供使用。 2. 導覽至 `bootstrap-terraform` 資料夾。 3. `main.tf`如果您想要變更 S3 儲存貯體 (`<accountId>-terraform-backend`) 和 Amazon DynamoDB 表格 (`terraform-lock`) 的名稱，請編輯檔案。 4. 執行`terraform apply`命令以部署資源。記下 S3 儲存貯體和 DynamoDB 料表名稱。	DevOps 工程師

部署解決方案基礎結

任務	描述	所需技能
檢閱並更新地形組態。	在根目錄中，開啟檔案`main.tf`,檢閱內容，並考慮進行下列更新： 1. 搜尋字串並將其取代為您要使用`eu-west-1` 的所需區域，以更新 AWS 區域。 2. 如果預設值在之前的 Epic 中已更改，請更新`Terraform Backend`區塊中的值區名稱。	DevOps 工程師

任務	描述	所需技能
	3. 如果在之前的 Epic 中更改了默認dynamodb_table 值，請更新該值。 4. 將stack_prefix 變數的值更新為您想要的字串。此字串會附加在此模式建立的所有資源名稱之前。 5. 更新vpc_cidr這應該至少是一個/24地址範圍的值。 6. 檢閱Locals區段。這是用來定義將部署的藍光時代任務。該解決方案將迭代列表對象bluage_batch_modules ，為列表中的每個元素創建關聯的資源（Step Functions 狀態機，任務定義和 SNS 主題）。在某些情況下，您可能想要針對不同的環境調整變數。例如，若要在測試環境中強制執行階段，您可以變更force_execution_time 變數的值。 7. 若要開啟網際網路存取，請將的值direct_internet_access_required 從變更false為true。這將部署網際網路閘道，以及開啟基礎結構的公用網際網路存取的 NAT 閘道和路由表。依預設，該解決方案會將介面虛擬私人雲端端點部署到	

任務	描述	所需技能
	VPC 中，而無需直接存取網際網路。 8. 若要授`additional_nlb_igress_cidrs` 與透過 Elastic Load Balancing 提供服務的任何用戶端-伺服器工作負載的存取權，請使用應允許的 CIDR 網路更新值。	
部署地形文件。	從終端機執行`terraform apply`命令以部署所有資源。複查 Terraform 產生的變更，然後輸入 yes 以啟動建置。 請注意，部署此基礎結構可能需要 15 分鐘以上的時間。	DevOps 工程師

(選擇性) 部署有效的 Blu Age 容器化應用程式

任務	描述	所需技能
將藍光時代容器映像推送到 Amazon ECR。	將容器推入您在之前史詩中建立的 Amazon ECR 儲存庫。如需指示，請參閱 Amazon ECR 文件。 記下容器映像 URI。	DevOps 工程師
更新地形以引用藍光時代容器映像。	更新檔案`main.tf`以參照您上傳的容器映像檔。	DevOps 工程師
重新部署地形文件。	從終端機執行`terraform apply`以部署所有資源。檢閱	DevOps 工程師

任務	描述	所需技能
	Terraform 的建議更新，然後輸入 yes 以繼續部署。	

相關資源

- 藍色時代
- 在 AWS 無伺服器基礎設施上執行已透過 Blu Age 現代化的大型主機工作負載
- 容器化已由 Blu Age 現代化的大型主機工作負載

在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight

環境：PoC 或試點	技術：大型主機；分析；移轉；現代化；機器學習與人工智慧	工作負載：IBM
AWS 服務：AWS Lambda；AWS 大型主機現代化；Amazon；Amazon QuickSight S3		

Summary

如果您的組織在大型主機環境中託管業務關鍵資料，那麼從該資料中獲得深入解析對於推動成長和創新至關重要。透過解除鎖定大型主機資料，您可以建置更快、安全且可擴展的商業智慧，以加速 Amazon Web Services (AWS) 雲端中資料導向的決策、成長和創新。

此模式提供了一種解決方案，可透過將AWS Mainframe Modernization 檔案傳輸與 BMC 和 Amazon Q 配合使用，從大型主機資料建立可共用敘述。 QuickSight透過使用 BMC 的 AWS 大型主機現代化檔案傳輸，可將大型主機資料集傳輸至 Amazon 簡單儲存服務 (Amazon S3)。 AWS Lambda 函數會格式化並準備大型主機資料檔案，以便載入 Amazon。 QuickSight

在 Amazon 提供資料之後 QuickSight，您可以在 Amazon Q 中使用自然語言提示 QuickSight 來建立資料摘要、提出問題以及產生資料故事。您不必撰寫 SQL 查詢或學習商業智慧 (BI) 工具。

業務背景

此模式為大型主機資料分析和資料洞察使用案例提供解決方案。使用該模式，您可以為公司的數據構建可視化儀表板。為了展示解決方案，這種模式使用了一家醫療保健公司，該公司為其在美國的成員提供醫療，牙科和視力計劃。在此範例中，成員人口統計資料和計劃資訊儲存在大型主機資料集中。視覺化儀表板顯示以下內容：

- 按地區分的成員分佈
- 按性別劃分的成員分
- 按年齡分類的成員分

- 按計劃類型分類的成員分
- 未完成預防免疫接種的會員

建立儀表板後，您會產生一個資料故事，說明先前分析的深入解析。數據故事提供了增加已完成預防免疫接種的成員數量的建議。

先決條件和限制

先決條件

- 一個活躍的 AWS 帳戶
- 含有業務資料的大型主機資料集
- 在大型主機上安裝檔案傳輸代理程式的存取權

限制

- 您的大型主機資料檔案應採用 Amazon 支援的其中一種檔案格式。 QuickSight如需支援的檔案格式清單，請參閱 Amazon QuickSight 文件。

  此模式使用 Lambda 函數將大型主機檔案轉換成 Amazon 支援的格式。 QuickSight

架構

下圖顯示透過使用 AWS Mainframe Modernization 檔案傳輸搭配 BMC 和 Amazon Q，從大型主機資料產生商業見解的架構。 QuickSight

該圖顯示以下工作流程：

1. 包含商業資料的大型主機資料集會透過使用 BMC 的 AWS Mainframe Modernization 檔案傳輸到 Amazon S3。
2. Lambda 函數會將檔案傳輸目標 S3 儲存貯體中的檔案轉換為逗號分隔值 (CSV) 格式。
3. Lambda 函數會將轉換後的檔案傳送至來源資料集 S3 儲存貯體。
4. 檔案中的資料是由 Amazon QuickSight 擷取。
5. 用戶在 Amazon 訪問數據 QuickSight。您可以在中使用 Amazon Q QuickSight ，透過自然語言提示與資料互動。

工具

AWS 服務

- AWS Lambda 是一項運算服務，可協助您執行程式碼，無需佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Mainframe Modernization 使用 BMC 進行檔案傳輸會將大型主機資料集轉換並傳輸到 Amazon S3，以進行大型主機現代化、移轉和擴充使用案例。
- Amazon QuickSight 是雲端規模的 BI 服務，可協助您在單一儀表板中視覺化、分析和報告資料。這種模式使用 Amazon Q 的生成 BI 功能 QuickSight。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

最佳實務

- 當您使用 BMC 和 Lambda 函數建立 AWS Mainframe Modernization 檔案傳輸的 AWS Identity and Access Management (IAM) 角色時，請遵循最低權限原則。
- 確保您的來源資料集具有支援 Amazon 的資料類型 QuickSight。如果來源資料集包含不支援的資料類型，請將其轉換為支援的資料類型。如需有關不受支援的大型主機資料類型，以及如何將其轉換為中 Amazon Q 支援的資料類型的詳細資訊 QuickSight，請參閱相關資源一節。

史诗

使用 BMC 設定 AWS Mainframe Modernization 檔案傳輸

任務	描述	所需技能
安裝檔案傳輸代理程式。	若要在大型主機上安裝 AWS Mainframe Modernization 檔案傳輸代理程式，請遵循文件中的指示。AWS	大型主機系統管理員
為大型主機檔案傳輸建立 S3 儲存貯體。	使用 BMC 建立 S3 儲存貯體以儲存「檔案傳輸」的輸出 AWS Mainframe Modernization 檔	移民工程師

任務	描述	所需技能
	案。在架構圖中，這是文件傳輸目的地存儲桶。	
建立資料傳輸端點。	1. 建立 S3 儲存貯體以暫存輸入大型主機檔案，以便透過 BMC 進行 AWS Mainframe Modernization 檔案傳輸。 2. 若要建立大型主機資料傳輸端點，請遵循文件中的指示。AWS	AWS 大型主機現代化專家

將大型主機檔案副檔名轉換為 Amazon 整合 QuickSight

任務	描述	所需技能
建立 S3 儲存貯體。	為 Lambda 函數建立 S3 儲存貯體，將轉換後的大型主機檔案從來源複製到最終目的地儲存貯體。	移民工程師
建立 Lambda 函數。	若要建立 Lambda 函數來變更副檔名，並將大型主機檔案複製到目的地儲存貯體，請執行下列動作： 1. 登入 AWS Management Console，然後瀏覽至主 AWS Lambda 控台。 2. 選擇 [建立函式]，然後選擇 [從頭開始作者]。 3. 在函數名稱中，輸入函數的名稱。 4. 在「執行階段」下拉式清單中，選擇「Python .3.x」。	移民工程師

任務	描述	所需技能
	5. 展開 [變更預設執行角色]，然後選擇 [使用基本 Lambda 權限建立新角色]。 6. 選擇建立函數。 7. 選擇 [程式碼] 索引標籤，然後貼上 [其他資訊] 區段中提供的 `S3CopyLambda.py` Python 程式碼。Python 代碼是通過使用 Amazon Q 開發人員在 Microsoft 視覺工作室集成開發環境（IDE）生成的。 8. 編輯您先前建立的 S3 儲存貯體名稱，然後編輯`change destination_file_key` 到大型主機檔案名稱。`destination_bucket_name` 9. 部署 Lambda 函數。	

任務	描述	所需技能
建立 Amazon S3 觸發程序以叫用 Lambda 函數。	若要設定叫用 Lambda 函數的觸發器，請執行下列動作： 1. 在 Lambda 主控台上，開啟「函數」頁面。 2. 選擇 Lambda 函數。 3. 在函數概覽中，選擇新增觸發器。 4. 在 [觸發器組態] 下拉式清單中，選擇 S3。 5. 在「時段」欄位中，輸入來源值區的名稱。 6. 在 [事件類型] 下拉式清單中，選擇 [所有物件建立事件]。 7. 選取 [我確認不建議針對輸入和輸出使用相同的 S3 儲存貯體] 核取方塊，然後選擇 [新增]。 如需詳細資訊，請參閱教學課程：使用 Amazon S3 觸發條件叫用 Lambda 函數。	遷移, 領導

任務	描述	所需技能
為 Lambda 函數提供 IAM 許可。	Lambda 函數需要 IAM 許可才能存取檔案傳輸目的地和來源資料集 S3 儲存貯體。透過允許`s3:GetObject` 和許可檔案傳輸目的地 S3 儲存貯體以及來源資料集 S3 儲存貯體的`s3:PutObject` 存取`s3:DeleteObject` 權，來更新與 Lambda 函數執行角色相關聯的政策。 如需詳細資訊，請參閱教學課程：使用 Amazon S3 觸發器叫用 Lambda 函數中的建立許可政策一節。	遷移, 領導

定義大型主機資料傳輸工作

任務	描述	所需技能
建立傳輸任務，將大型主機檔案複製到 S3 儲存貯體。	若要建立大型主機檔案傳輸工作，請遵循文件中的指示。AWS Mainframe Modernization 注意：指定原始碼頁編碼為 IBM1047，將目標字碼頁編碼指定為 UTF-8。	移民工程師
驗證傳輸工作。	若要驗證資料傳輸是否成功，請遵循AWS Mainframe Modernization 文件中的指示。確認大型主機檔案位於檔案傳輸目的地 S3 儲存貯體中。	遷移, 領導

任務	描述	所需技能
驗證 Lambda 複製函數。	確認已啟動 Lambda 函數，並將檔案以 .csv 副檔名複製到來源資料集 S3 儲存貯體。 由 Lambda 函數創建的 .csv 文件是 Amazon QuickSight 的輸入數據文件。如需資料範例，請參閱「附件」區段中的`Sample-data-member-healthcare-APG` 檔案。	遷移, 領導

將 Amazon Connect QuickSight 到大型主機數據

任務	描述	所需技能
設置 Amazon QuickSight。	要設置 Amazon QuickSight，請按照AWS 文檔中的說明進行操作。	遷移, 領導
為 Amazon 創建一個數據集 QuickSight。	要為 Amazon 創建數據集 QuickSight，請按照AWS 文檔中的說明進行操作。輸入資料檔案是您定義大型主機資料傳輸工作時所建立的轉換後的大型主機檔案。	遷移, 領導

在中使用 Amazon Q，從大型主機資料中取得商業洞見 QuickSight

任務	描述	所需技能
設置 Amazon Q QuickSight.	此功能需要企業版。若要在中設定 Amazon Q QuickSight，請執行下列動作：	遷移, 領導

任務	描述	所需技能
	1. 若要取得 Amazon Q 附加元件，請遵循說明文件中的步驟 1：取得 Q 附加元AWS件。 2. 若要在 Amazon Q 中使用生成式 BI 功能，請升級使用者的帳戶。按照AWS 文檔中的說明進行操作。 3. 使用您先前建立的資料集建立 Amazon Q 主題。按照AWS 文檔中的說明進行操作。 4. 若要設定主題中繼資料，使其適合自然語言，請遵循文件中的指示。AWS	

任務	描述	所需技能
分析大型主機資料並建置視覺化儀表板。	若要在 Amazon 中分析和視覺化您的資料 QuickSight，請執行以下操作： 1. 若要建立大型主機資料分析，請遵循文件中的指示。AWS針對資料集，選擇在上一個步驟中建立的資料集。 2. 在分析頁面上，選擇構建可視化。 3. 在「建立分析」主題視窗中，選擇「更新現有主題」。 4. 在 [選取主題] 下拉式清單中，選擇您先前建立的主題。 5. 選擇 [主題連結]。 6. 連結主題後，選擇建立視覺效果以開啟 Amazon Q 建立視覺化視覺化視窗。 7. 在提示欄中。寫你的分析問題。用於此模式的示例問題如下： • 按地區顯示成員分佈 • 按年齡顯示成員分佈 • 按性別顯示成員分佈 • 按計劃類型顯示成員分佈 • 顯示會員未完成預防免疫 輸入問題後，請選擇 [建置]。Amazon Q 在	移民工程師

任務	描述	所需技能
	QuickSight 創建的視覺效果。 8. 若要將視覺效果新增至視覺化儀表板，請選擇 [新增至分析]。 完成後，您可以發佈儀表板，以便與組織中的其他人共用。如需範例，請參閱其他資訊一節中的大型主機視覺化儀表板。	

使用 Amazon Q QuickSight 從大型主機資料建立資料故事

任務	描述	所需技能
建立資料故事。	創建數據故事以解釋先前分析的見解，並生成建議以增加會員的預防免疫接種： 1. 若要建立資料故事，請遵循AWS 文件中的指示。 2. 對於資料故事提示，請使用下列命令： `Build a data story about Region with most numbers of members. Also show the member distribution by medical plan, vision plan, dental plan. Recommend how to`	移民工程師

任務	描述	所需技能
	`motivate members to complete immunization. Include 4 points of supporting data for this pattern.` 您也可以建立自己的提示，以產生其他業務見解的資料故事。 3. 選擇 [新增視覺效果]，然後新增與資料故事相關的視覺效果。對於此模式，請使用您先前建立的視覺效果。 4. 選擇 Build (建置)。 5. 如需資料內文輸出的範例，請參閱其他資訊一節中的資料故事輸出。	
檢視產生的資料故事。	若要檢視產生的資料故事，請遵循AWS 文件中的指示。	遷移, 領導
編輯產生的資料故事。	若要變更資料內文中的格式、版面配置或視覺效果，請遵循AWS 文件中的指示。	遷移, 領導
分享資料故事。	若要共用資料故事，請遵循AWS 文件中的指示。	移民工程師

故障診斷

問題	解決方案
無法找到在使用 BMC 的檔案傳輸中建立傳輸任務的資料集搜尋條件中輸入的大型主機 AWS Mainframe Modernization 檔案或資料集。	1. 首先，請選擇「使用 BMC 傳輸」主控台上的「資料 AWS Mainframe Modernization 傳輸端點」來檢查連線。如果上次活動訊號時間超過兩分鐘，表示尚未建立檔案傳輸的連線。如果在大型主機上執行的代理程式的上次活動訊號時間少於 2 分鐘，表示代理程式的連線成功。繼續執行步驟 2。 2. 檢查設 AWS Secrets Manager 置。秘密金鑰必須在 Secrets Manager 中設定，其中包含大型主機使用者識別碼值的金鑰 **`userId`** (大寫 I)，以及**`password`**具有大型主機密碼值的金鑰。`userId`和`password`秘密金鑰區分大小寫，必須按原樣輸入。

相關資源

若要將封包十進位 (COMP-3) 或二進位 (COMP 或 COMP-4) 等大型主機資料類型轉換為 Amazon 支援的資料類型 QuickSight，請參閱下列模式：

- 通過使用 Python 將 EBCDIC 數據轉換並解壓縮為 ASCII AWS
- 在 Amazon S3 中，使用將大型主機檔案從 EBCDIC 格式轉換為字元分隔的 ASCII 格式 AWS Lambda

其他資訊

三CopyLambda年

下面的 Python 代碼是通過使用與 Amazon Q 開發人員在 IDE 中的提示生成：

```
#Create a lambda function triggered by S3. display the S3 bucket name and key
import boto3
s3 = boto3.client('s3')
```

```
def lambda_handler(event, context):
print(event)
bucket = event['Records'][0]['s3']['bucket']['name']
key = event['Records'][0]['s3']['object']['key']
print(bucket, key)
#If key starts with object_created, skip copy, print "copy skipped". Return lambda with
 key value.
if key.startswith('object_created'):
print("copy skipped")
return {
'statusCode': 200,
'body': key
}
# Copy the file from the source bucket to the destination bucket.
 Destination_bucket_name = 'm2-filetransfer-final-opt-bkt'. Destination_file_key =
 'healthdata.csv'
copy_source = {'Bucket': bucket, 'Key': key}
s3.copy_object(Bucket='m2-filetransfer-final-opt-bkt', Key='healthdata.csv',
 CopySource=copy_source)
print("file copied")
#Delete the file from the source bucket.
s3.delete_object(Bucket=bucket, Key=key)
return {
'statusCode': 200,
'body': 'Copy Successful'
}
```

大型主機視覺化儀表板

下面的數據視覺是由 Amazon Q 中 QuickSight 的分析問題創建的show member distribution by region。

以下數據可視化是由 Amazon Q 在 QuickSight 為這個問題創建的show member distribution by Region who have not completed preventive immunization, in pie chart。

資料故事輸出

下列螢幕擷取畫面顯示 Amazon Q 在中 QuickSight 針對提示建立的資料故事部分 Build a data story about Region with most numbers of members. Also show the member

```
distribution by medical plan, vision plan, dental plan. Recommend how to
motivate members to complete immunization. Include 4 points of supporting
data.
```

在介紹中，數據故事建議選擇成員最多的地區，以從免疫工作中獲得最大影響。

數據故事提供了前三個地區的會員人數的分析，並將西南地區命名為專注於免疫接種工作的主要地區。

注意：西南和東北區域各有八個成員。但是，西南地區有更多的成員沒有完全接種疫苗，因此從提高免疫率的舉措中受益更大的潛力。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

整合石分支通用控制器與 AWS 大型主機現代化

程式碼儲存庫：AWS 大型主機現代化石分公司整合	環境：PoC 或試點	技術：大型主機；現代化 DevOps；營運；SaaS
工作負載：開放原始碼 Microsoft	AWS 服務：AWS 大型主機現代化；Amazon RDS；Amazon S3	

Summary

此模式說明如何將 Stonebranch 通用自動化中心 (UAC) 工作負載協調與 Amazon Web Services (AWS) 大型主機現代化服務整合。AWS 大型主機現代化服務可將大型主機應用程式移轉並現代化到 AWS 雲端。它提供兩種模式：採用 Micro Focus 企業技術的 AWS 大型主機現代化重新平台，以及 AWS Blu Age 的 AWS 大型主機現代化自動重構。

石分支 UAC 是一個實時的 IT 自動化和協調平台。UAC 旨在跨混合 IT 系統（從現場部署到 AWS）自動化和協調任務、活動和工作流程。使用大型主機系統的企業用戶端正在轉換為以雲端為中心的現代化基礎架構和應用程式。Stonebranch 的工具和專業服務有助於將現有的排程器和自動化功能遷移到 AWS 雲端。

使用 AWS 大型主機現代化服務將大型主機計劃遷移或現代化到 AWS 雲端時，您可以使用此整合自動化批次排程、提高靈活性、改善維護並降低成本。

此模式提供了將 Stonebranch 排程器與遷移到 AWS 大型主機現代化服務 Micro Focus 企業執行階段的大型主機應用程式整合的說明。此模式適用於解決方案架構師、開發人員、顧問、移轉專家，以及從事移轉、現代化、作業或 DevOps工作的其他人員。

目標成果

此模式著重於提供以下目標結果：

- 能夠從 Stonbranch 通用控制器排程、自動化和執行在 AWS 大型主機現代化服務 (微焦點執行階段) 中執行的大型主機批次任務。
- 從 Stonebranch 通用控制器監控應用程式的批次處理程序。
- 從 Stonebranch 通用控制器自動或手動啟動/重新執行/停止批次處理。

- 擷取 AWS 大型主機現代化批次程序的結果。
- 在 Stone 分支通用控制器中擷取批次任務的 AWS CloudWatch 日誌。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有 Job 控制語言 (JCL) 檔案的 Micro Focus Bankdemo 應用程式，以及部署在 AWS 大型主機現代化服務 (Micro Focus 執行階段) 環境中的批次程序
- 有關如何建置和部署在 Micro Focus 企業伺服器上執行的大型主機應用程式的基本知識
- 石分支通用控制器的基礎知識
- 石支試用許可證（聯繫石支）
- 具有至少四個核心、8 GB 記憶體和 2 GB 磁碟空間的 Windows 或 Linux Amazon 彈性運算雲端 (亞馬遜 EC2) 執行個體 (例如 xlarge) 執行個體
- 阿帕奇雄貓版本 8.5.x 或 9.0.x 版
- 甲骨文 Java 運行時環境（JRE）或 OpenJDK 版本 8 或 11
- Amazon Aurora MySQL 兼容版
- 用於匯出儲存庫的亞馬遜簡易儲存服務 (Amazon S3) 儲存貯體
- Amazon Elastic File System (Amaon EFS)，適用於代理程式石分支通用訊息服務 (OMS) 連線，提供高可用性 (HA)
- 石科通用控制器 7.2 通用代理 7.2 安裝檔
- AWS 大型主機現代化任務排程範本 (.zip 檔案的最新發行版本)

限制

- 該產品和解決方案已通過 OpenJDK 8 和 11 的測試和兼容性驗證。
- AWS 大型主機現代化石分支整合任務排程範本僅適用於 AWS 大型主機現代化服務。
- 這個任務調度模板將只能在 Unix，Linux 或 Windows 版本的石分支代理工作。

架構

目標狀態架構

下圖顯示此試行方案所需的 AWS 環境範例。

1. Stonebranch 通用自動化中心 (UAC) 包括兩個主要元件：通用控制器和通用代理程式。石分支 OMS 被用作控制器和各個代理之間的消息總線。
2. 石分支 UAC 數據庫由通用控制器使用。該數據庫可以是 MySQL，Microsoft SQL 服務器，甲骨文，或 Aurora MySQL 的兼容。
3. AWS 大型主機現代化服務 — 部署應用程式的 Micro Focus 執行階段環境。BankDemo BankDemo 應用程式檔案將存放在 S3 儲存貯體中。此存儲桶還包含大型主機 JCL 文件。
4. 石分支 UAC 可以為批量運行運行以下功能：
   a. 使用連結至 AWS 大型主機現代化服務之 S3 儲存貯體中的 JCL 檔案名稱啟動批次任務。
   b. 取得批次工作執行的狀態。
   c. 等待批次工作執行完成。
   d. 擷取批次工作執行的記錄。
   e. 重新執行失敗的批次工作。
   f. 在工作執行時取消批次處理工作。
5. 石科 UAC 可以為應用程序運行以下功能：
   a. 開始申請
   b. 獲取應用程序的狀態
   c. 等到應用程式啟動或停止
   d. 停止申請
   e. 擷取應用程式作業的記錄

石分支工作轉換

下圖顯示了 Stonbranch 在現代化過程中的工作轉換過程。它說明任務排程和任務定義如何轉換成可執行 AWS 大型主機現代化批次任務的相容格式。

1. 對於轉換程序，工作定義會從現有的大型主機系統匯出。
2. 您可以將 JCL 檔案上傳到大型主機現代化應用程式的 S3 儲存貯體，以便 AWS 大型主機現代化服務部署這些 JCL 檔案。
3. 轉換工具會將匯出的工作定義轉換為 UAC 工作。

4. 建立所有作業定義和工作排程之後，這些物件會匯入至通用控制器。然後，轉換後的任務會在 AWS 大型主機現代化服務中執行程序，而不是在大型主機上執行這些程序。

石支 UAC 架構

下列架構圖表示高可用性 (HA) 通用控制器的 active-active-passive 模型。Stonebranch UAC 部署在多個可用區域中，以提供高可用性並支援災難復原 (DR)。

通用控制器

兩個 Linux 伺服器會佈建為通用控制器。兩者都連接到相同的數據庫端點。每個伺服器都有通用控制器應用程式和 OMS。最新版本的通用控制器會在佈建時使用。

通用控制器部署在 Tomcat Web 應用程序中作為文檔根目錄，並在端口 80 上提供服務。此部署可簡化前端負載平衡器的組態。

使用石分支萬用字元憑證 (例如`https://customer.stonebranch.cloud`) 啟用透過 TLS 或 HTTPS 的 HTTP。這樣可以保護瀏覽器和應用程序之間的通信。

OMS

通用代理程式和 OMS (Opswise 訊息服務) 位於每個通用控制器伺服器上。從客戶端部署的所有通用代理程式都會設定為連線到兩個 OMS 服務。OMS 充當通用代理程式和通用控制器之間的通用訊息傳送服務。

Amazon EFS 會在每部伺服器上掛接多工緩衝處理目錄。OMS 會使用此共用多工緩衝處理目錄來保留來自控制器和代理程式的連線和工作資訊。OMS 在高可用性模式下運作。如果作用中 OMS 停止運作，則被動 OMS 可以存取所有資料，而且會自動繼續作用中作業。通用代理程式會偵測此變更，並自動連線至新的作用中 OMS。

資料庫

Amazon Relational Database Service 服務 (Amazon RDS) 內建 UAC 資料庫，其引擎與 Amazon Aurora MySQL 相容。Amazon RDS 有助於定期管理和提供定期備份。兩個通用控制器執行個體都會連線到相同的資料庫端

負載平衡器

為每個執行個體設定 Application Load Balancer。負載平衡器會在任何給定時刻將流量導向至作用中控制器。您的執行個體網域名稱會指向個別的負載平衡器端點。

网址

每個執行個體都有一個 URL，如下列範例所示。

Environment (環境)	執行個體
生產	客戶. 石分支雲
開發 (非生產)	客戶開發. 石分支.
測試（非生產）	客戶最好的. 石分支.

注意：可以根據您的需求設定非生產執行個體名稱。

高可用性

高可用性 (HA) 是指系統在指定時間內持續運作而不失敗的能力。此類失敗包括但不限於儲存裝置、CPU 或記憶體問題所造成的伺服器通訊回應延遲，以及網路連線。

若要符合 HA 的要求：

- 所有 EC2 執行個體、資料庫和其他組態都會鏡像到相同 AWS 區域內的兩個獨立可用區域。
- 控制器是透過 Amazon 機器映像 (AMI) 在兩個可用區域中的兩部 Linux 伺服器上佈建的。例如，如果您在歐洲歐洲西部 -1 區域佈建，則可用區域中有一個通用控制器，歐盟西部-1a 和歐洲西部 -1c 可用區域。
- 不允許任何作業直接在應用程式伺服器上執行，也不允許在這些伺服器上儲存任何資料。
- 應用程式負載平衡器會在每個通用控制器上執行健康狀態檢查，以識別作用中的控制器，並將流量導向 如果一部伺服器發生問題，負載平衡器會自動將被動通用控制器提升為主動狀態。然後，負載平衡器會從健康狀態檢查中識別新的作用中通用控制器執行個體，並開始導向流量 容錯移轉會在四分鐘內發生，不會遺失任務，且前端 URL 保持不變。
- Aurora 與 MySQL 相容的資料庫服務會儲存通用控制器資料。對於生產環境，資料庫叢集是以單一 AWS 區域內兩個不同可用區域中的兩個資料庫執行個體建立的。這兩個通用控制器都使用指向單一資料庫叢集端點的 Java 資料庫連線 (JDBC) 介面。如果有一個資料庫執行個體發生問題，資料庫叢集端點會動態指向運作良好的執行個體。不需要手動介入。

Backup 和清除

Stonebranch 通用控制器被設置為備份和清除以下表中顯示的時間表中的舊數據。

類型	排程
活動	7 天
審計	90 天
歷程記錄	六十天

超過顯示日期的 Backup 資料會匯出為 .xml 格式，並儲存在檔案系統中。備份程序完成後，系統會從資料庫中清除較舊的資料，並在 S3 儲存貯體中存檔最多一年的生產執行個體。

您可以在通用控制器介面中調整此排程。但是，增加這些時間範圍可能會導致維護期間更長的停機時間。

工具

AWS 服務

- AWS 大型主機現代化是 AWS 雲端原生平台，可協助您將大型主機應用程式現代化為 AWS 受管執行階段環境。它提供工具和資源來協助您規劃和實作遷移和現代化。
- Amazon Elastic Block Store (Amazon EBS) 提供區塊層級儲存體磁碟區，可搭配使用 Amazon EC2 執行個體。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。此模式使用 Amazon Aurora MySQL 兼容版本。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon EC2 執行個體、容器和 IP 地址。此模式使用應用程式負載平衡器。

石枝

- 通用自動化中心 (UAC) 是企業工作負載自動化產品的系統。此模式使用下列 UAC 元件：
  - 通用控制器，在 Tomcat Web 容器中運行的 Java Web 應用程序，是通用自動化中心的企業作業調度程序和工作負載自動化代理解決方案。控制器提供使用者介面，可用來建立、監視和設定控制

器資訊、處理排程邏輯、處理往來通用代理程式的所有訊息；以及同步通用自動化中心的大部分高可用性作業。

  - Universal Agent 是獨立於廠商的排程代理程式，可與所有主要運算平台 (舊版和分散式) 上的現有工作排程器協同作業。支援在 Z 系列、I 系列、Unix、Linux 或視窗上執行的所有排程器。
- Universal Agent 是獨立於廠商的排程代理程式，可與所有主要運算平台 (舊版和分散式) 上的現有工作排程器協同作業。支援在 Z 系列、I 系列、Unix、Linux 或視窗上執行的所有排程器。
- Stonebranch aws-mainframe-modernization-stonebranch 整合 AWS 大型主機現代化通用擴充功能是在 AWS 大型主機現代化平台中執行、監控和重新執行批次任務的整合範本。

Code

此模式的代碼可在 AWS 大型 GitHub 主機現代化石分支集成存儲庫中找到。

史诗

在 Amazon EC2 上安裝通用控制器和通用代理

任務	描述	所需技能
下載安裝檔案。	從石分支服務器下載安裝。要獲取安裝文件，請與石分支聯繫。	雲端架構師
啟動 EC2 執行個體。	安裝通用控制器和通用代理程式需要約 3 GB 的額外空間。因此，請為執行個體提供至少 30 GB 的磁碟空間。 將連接埠 8080 新增至安全性群組，以便可存取。	雲端架構師
檢查先決條件。	在安裝之前，請執行下列動作： 1. 依照下載 Java 執行階段環境中所述安裝 Java。 `$ sudo yum -y update`	雲端管理員、Linux 管理員

任務	描述	所需技能
	`` $ sudo yum install java-11-amazon-cor retto ``` 請務必使用其中一個支援的 JAVA 版本。上一個命令應該安裝 java 11。請檢查 Java 版本，並確保您使用的是版本 11，然後再繼續。 2. 如安裝 Apache Tomcat 文件中所述，執行下列命令。 ``` $ sudo yum install tomcat tomcat-admin- webapps $ sudo systemctl enable tomcat $ sudo systemctl start tomcat ``` 3. 如建立 Aurora MySQL 資料庫叢集並連線至叢集中所述，建立 Amazon Aurora 資料庫。使用 Amazon Aurora 與 MySQL 相容的版本。 選擇一個主用戶名和主密碼。保留其餘設定的預設值。	

任務	描述	所需技能
安裝通用控制器。	1. 將universal-controller-7.2.0.0.tar 安裝檔案上傳至 EC2 執行個體。 2. 將安裝檔案解除封存至temp資料夾。 `$ tar -xvf universal-controller-7.2.0.0.tar` 3. 授予安裝指令碼執行權限。 `$ chmod a+x install-controller.sh` 4. 安裝控制器。此範例使用下列命令，在 /usr/share/tomca t 下安裝通用控制器。使用您在先前步驟中建立的 Amazon Aurora 資料庫。 `$ sudo ./install-controller.sh --tomcat-dir /usr/share/tomcat/ --controller-file universal-controller-7.2.0.0-build.145.war --dbuser admin --dbpass “****” --dbname uc --rdbms mysql --dburl jdbc:mysql://database-2-instance-1.cih63miincgy.us-east-1.rds.amazonaws.com:3306/`	雲端架構師、Linux 管理員

任務	描述	所需技能
	腳本輸出的最後一行應該是「安裝完成」。 5. 導覽至 EC2 執行個體中的下列 URL。 `http://<public_ip>:8080/uc` 6. 在登入畫面上，在「使用者名稱」區段中輸入 ops.admin，並將「密碼」欄位保持空白。 7. 為使用者設定新密ops.admin 碼。	

任務	描述	所需技能
安裝通用代理程式。	1. 將sb-7.2.0.1-linux-3.10-x86_64.tar.Z 安裝檔案上傳至 EC2 執行個體。 2. 登入 EC2 執行個體。 3. 取消封存通用代理程式安裝套件。 `$ zcat sb-7.2.0.1-linux-3.10-x86_64.tar.Z | tar xvf -` 4. 執行下列命令。 `$ sudo ./unvinst --oms_servers 7878@localhost --oms_autostart yes --python yes` 5. 建立 PAM 檔案。 `$ cp /etc/pam.d/login /etc/pam.d/ucmd` 6. 啟用通用代理程式的自動啟動。 `$ /sbin/restorecon -v /etc/rc.d/init.d/ubrokerd`	雲端管理員、Linux 管理員

任務	描述	所需技能
將 OMS 新增至通用控制器。	1. 與使用者一起登入通 `ops.admin` 用控制器。 2. 選擇畫面左上角的「服務」功能表，然後選擇「系統」中的「OMS 伺服器」功能表 3. 在 OMS 伺服器位址欄位中，輸入 localhost，然後儲存。 4. 您會看到 OMS 伺服器的狀態為 [已連線] 和 [工作階段狀態] 為 [作業中]。	通用控制器管理

匯入 AWS 大型主機現代化通用延伸模組並建立任務

任務	描述	所需技能
匯入整合範本。	在此步驟中，您需要 AWS 大型主機現代化通用擴充功能。確定已下載 .zip 檔案的最新發行版本。 1. 與使用者一起登入通 `ops.admin` 用控制器。 2. 瀏覽至「服務」，「匯入整合範本」。 3. 選取整合範本 .zip 檔案 (`aws_mainframe_modernization_stonebranch_extension.zip`)，然後選擇「匯入」。	通用控制器管理

任務	描述	所需技能
	匯入整合範本之後，您會在「可用服務」下看到 AWS 大型主機現代化任務。	

任務	描述	所需技能
啟用可解析的認證。	1. 導覽至服務、AWS 大型主機現代化任務。 2. 在右側面板中，填寫必填欄位： • 名稱：新大型主機現代化工作 • 代理程式：選取唯一的代理程式 (AGNT0001)。 在 AWS 大型主機現代化詳細資訊下： • 動作：列出環境 • AWS 登入資料：如果您已將 AWS Identity and Access Management (IAM) 角色新增至 EC2 執行個體，則可以將此欄位保留空白。如果要使用`AWSAccessKeyID` 和`AWSSecretKey` ，請選擇欄位旁邊的圖示 ()。 在開啟的「身份證明詳細資料」視窗中，輸入下列資訊，然後儲存。 • 名稱：AWS 大型主機現代化登入資料 • 執行階段使用者：在此欄位中寫入 AWS 存取金鑰 ID。 • 執行階段密碼：在此欄位中寫入 AWS 秘密金鑰。	通用控制器管理

任務	描述	所需技能
	• 端點：確定端點具有正確的 AWS 區域。預設值為 https://m2.us-east-1.amazonaws.com。 • 區域：輸入 AWS 大型主機現代化服務的區域。預設值為 `us-east-1` 。 3. 將預設值保留在其餘欄位中，並儲存工作。	

任務	描述	所需技能
啟動工作。	1. 在右側面板的頂部，選擇啟動任務。 2. 在「確認」視窗中，選擇「啟動」。之後，通用控制器主控台將顯示類似下列訊息的訊息。 2022-08-24 上午十一時四十九分 使用任務實例 sys_id 166129149363414631 3NC8E38DB8OZJY 成功啟動通用任務「新大型主機現代化任務」。 3. 導覽至 [執行個體] 如果您看不到 [執行個體] 索引標籤，請選擇向右箭頭以向右捲動。 4. 開啟清單中任務實例的前後關聯 (按一下滑鼠右鍵) 功能表，選擇「擷取輸出」，然後在「擷取輸出」中選擇「提交」 5. 在「檢索輸出」窗口中，您將看到 STDOUT 中的環境列表。	通用控制器管理

測試啟動批次工作

任務	描述	所需技能
建立批次工作的工作。	1. 導覽至服務、AWS 大型主機現代化任務。	通用控制器管理

任務	描述	所需技能
	2. 在右側面板中，填寫必填欄位： • 名稱：新大型主機現代化工作 • 代理程式：選取唯一的代理程式 (AGNT0001)。 在 AWS 大型主機現代化詳細資訊下： • 動作：啟動 Batch (或啟動 Batch 並等待執行批次任務，然後等待 AWS 中的任務完成) • AWS 登入資料：如果您已將 IAM 角色新增至 EC2 執行個體，則可以將此欄位保持空白。如果要使用AWSAccess KeyID 和AWSSecret Key ，請選擇欄位旁邊的圖示 ()。 • 端點：確定端點具有正確的 AWS 區域。預設值為 https://m2.us-east-1.amazonaws.com。 • 區域：輸入 AWS 大型主機現代化服務的區域。預設值為 us-east-1 。 • 應用模組：選擇欄位 () 旁邊的圖示，然後在「重新整理應用模組選擇」中選擇「提交」。這將連接到 AWS 大型主機現代化服務，並傳回應用程式清	

任務	描述	所需技能
	單。現在，您可以從下拉列表中選擇應用程序。選取要執行批次工作的應用程式。 • JCL 檔案名稱：RUNHELLO.jcl • 等待成功或失敗：如果選取此選項，工作將會等到批次工作的狀態為成功或失敗。 • 輪詢間隔：這是每個輪詢之間的時間量。 • 擷取執行記錄：如果選取此選項，則會在批次工作完成時自動擷取記錄檔。 • 日誌格式：這是要打印出來的日誌格式。它可以是文本或 JSON 格式。 3. 將預設值保留在其餘欄位中，並儲存工作。	

任務	描述	所需技能
啟動工作。	1. 在右側面板的頂部，選擇啟動任務。 2. 在「確認」視窗中，選擇「啟動」。之後，通用控制器主控台將顯示類似下列訊息的訊息。 上午十一時五十九分 使用任務實例 sys_id 成功啟動通用任務「大型主機現代化開始 Batch」。 <sys id> 3. 導覽至 [執行個體] 如果您看不到 [執行個體] 索引標籤，請選擇向右箭頭以向右捲動。 4. 開啟清單中任務實例的前後關聯 (按一下滑鼠右鍵) 功能表，選擇「擷取輸出」，然後在「擷取輸出」中選擇「提交」 5. 在「檢索輸出」窗口中，您將看到 STDOUT 中的環境列表。	通用控制器管理

建立多個任務的工作流程

任務	描述	所需技能
複製任務。	1. 開啟您要建立複本之工作的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「複製」。	通用控制器管理

任務	描述	所需技能
	2. 在「複製 AWS 大型主機現代化任務」視窗中，為新任務輸入以下新名稱：大型主機現代化開始 Batch-RUN AWS2。 3. 使用下列名稱再次複製工作：大型主機現代化開始 Batch-RUNAWS3。 4. 使用以下名稱再次複製任務：大型主機現代化開始 Batch-RUNAWS4。 5. 使用下列名稱複製工作的最後一次：大型主機現代化開始 Batch-FOOBAR。	

任務	描述	所需技能
更新工作。	1. 開啟 (連按兩下)「大型主機現代化開始 Batch-RUNAWS2」工作，將 `RUNAWS2.jcl` 「JCL 檔案名稱」欄位變更為，然後儲存。 2. 開啟 (連按兩下)「大型主機現代化開始 Batch-RUNAWS3」工作，將 *`RUNAWS3.jcl`* 「JCL 檔案名稱」欄位變更為，然後儲存。 3. 開啟 (連按兩下)「大型主機現代化開始 Batch-RUNAWS4」工作，將 *`RUNAWS4.jcl`* 「JCL 檔案名稱」欄位變更為，然後儲存。 4. 開啟 (按兩下)「大型主機現代化開始 Batch-FOO BAR」工作，將「JCL 檔案名稱」欄位變更為，然後儲存。*`MISSING.jcl`* 此工作將失敗，因為 JCL 檔案名稱值不正確。	通用控制器管理

任務	描述	所需技能
建立工作流程。	1. 導航到服務，工作流程。 2. 在右側面板的「名稱」欄位中輸入大型主機現代化工作流程，然後儲存。 3. 在右側面板中，選擇「編輯工作流程」。 4. 在「工作流程編輯器」標籤上，「加入工作」按鈕 (+)。 5. 在「工作搜尋」視窗中，選擇「搜尋」以查看「通用控制器」中的所有作業。 6. 按一下大型主機現代化開始 Batch 工作旁邊的圖示，然後將圖示拖曳至工作流程編輯器中的空白位置。 7. 對其他大型主機現代化工作重複相同的動作，並按照其他資訊一節中所示的方式放置它們。 8. 選擇「連線」按鈕 ()，然後將工作 Connect 在一起。若要將工作與另一個工作連線，請按一下工作中間的，然後將其拖曳至目標工作。 9. 依照「其他資訊」區段中所示 Connect 工作，然後儲存工作流程。 10.在 「工作流程編輯器」中的空白處按一下滑鼠右鍵，選擇「啟動工作流程」，然後選	通用控制器管理

任務	描述	所需技能
檢查工作流程的狀態。	1. 在左側菜單中，選擇「活動」 2. 在視窗中間，選擇 [開始]。 您將在列表中看到任務實例的列表。 3. 開啟 (按兩下) 清單中的大型主機現代化工作流程，或開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇「工作流程工作流程」、「檢視工作流程 您會看到「其他資訊」區段中所示的工作。第二個任務預計會失敗，因為你使用了一個缺少的 JCL 文件。	通用控制器管理員

疑難排解失敗的批次工作並重新執

任務	描述	所需技能
修復失敗的工作並重新執行。	1. 開啟 (連按兩下) 失敗的工作，以查看工作的錯誤。 2. 您有兩個選項可以修復失敗的任務。 • 修正 JCL 檔案名稱，並將其設定為`FOOBAR.jcl` 。 • 將正確的 JCL 檔案名稱新增至 JCL 檔案名稱 (暫存)。此欄位將覆寫「JCL 檔案名稱」欄位。	通用控制器管理

任務	描述	所需技能
	對於此試驗，請選擇第二個選項，然後儲存工作實例。 3. 在「工作流程監視器」中，開啟失敗工作的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇指令，重新執行。 4. 之後，所有任務將成功完成。	

建立啟動應用程式和停止應用程式

任務	描述	所需技能
建立「啟動應用程式」動作。	1. 導覽至服務、AWS 大型主機現代化任務。 2. 在右側面板中，填寫必填欄位。 • 名稱：大型主機現代化啟動應用程式 • 代理程式：選取唯一的代理程式 (AGNT0001) 在 AWS 大型主機現代化詳細資訊下： • 動作：啟動應用程式 • AWS 登入資料：如果您已將 IAM 角色新增至 EC2 執行個體，則可以將此欄位保持空白。如果要使用AWSAccessKeyID 和AWSSecret	通用控制器管理

任務	描述	所需技能
	Key ，請選取之前建立的認證。 • 終點：確定端點具有正確的「區域」。預設值為 https://m2.us-east-1.amazonaws.com。 • 區域：輸入 AWS 大型主機現代化服務的區域。預設值為 us-east-1 。 • 應用模組：選擇欄位 () 旁邊的圖示，然後在「重新整理應用模組選擇」中選擇「提交」。這將連接到 AWS 大型主機現代化服務，並傳回應用程式清單。現在，您可以從下拉列表中選擇應用程序。選取要執行批次工作的應用程式。 • 等待成功或失敗：如果選取此選項，工作將會等到批次工作的狀態為成功或失敗。 • 輪詢間隔：這是每個輪詢之間的時間量。 • 擷取執行記錄：如果選取此選項，則會在批次工作完成時自動擷取記錄檔。 • 日誌格式：這是要打印出來的日誌格式。它可以是文本或 JSON 格式。 3. 將預設值保留在其餘欄位中，並儲存工作。	

任務	描述	所需技能
	4. 現在複製此任務並為停止應用程序創建一個任務。將名稱變更為「大型主機現代化停止應用程式」，然後將動作變更為「停止應用程式」。	

建立取消 Batch 執行作業

任務	描述	所需技能
建立「取消 Batch」動作。	1. 導覽至服務、AWS 大型主機現代化任務。 2. 在右側面板中，填寫必填欄位。 • 名稱：大型主機現代化取消 Batch 執行 • 代理程式：選取唯一的代理程式 (AGNT0001) 在 AWS 大型主機現代化詳細資訊下： • 動作：取消 Batch 執行 • AWS 登入資料：如果您已將 IAM 角色新增至 EC2 執行個體，則可以將此欄位保持空白。如果要使用`AWSAccessKeyID` 和`AWSSecretKey` ，請選取之前建立的認證。 • 終點：確定端點具有正確的「區域」。預設	

任務	描述	所需技能
	值為 https://m2.us-east-1.amazonaws.com。 • 區域：輸入 AWS 大型主機現代化服務的區域。預設值為 us-east-1 。 • 應用模組：選擇欄位 () 旁邊的圖示，然後在「重新整理應用模組選擇」中選擇「提交」。這將連接到 AWS 大型主機現代化服務，並傳回應用程式清單。現在，您可以從下拉列表中選擇應用程序。選取要執行批次工作的應用程式。 • 等待成功或失敗：如果選取此選項，工作將會等到批次工作的狀態為成功或失敗。 • 輪詢間隔：這是每個輪詢之間的時間量。 • 擷取執行記錄：如果選取此選項，則會在批次工作完成時自動擷取記錄檔。 • 日誌格式：這是要打印出來的日誌格式。它可以是文本或 JSON 格式。 3. 將預設值保留在其餘欄位中，並儲存工作。	

相關資源

- 通用控制器

- 通用代理
- LDAP 設定
- 單一登入設定
- 高可用性
- 快速轉換工具

其他資訊

工作流程編輯器中的圖示

連接的所有任務

工作流狀態

使用精確 Connect 將 VSAM 文件遷移和複寫到 Amazon RDS 或 Amazon MSK

由普拉奇汗娜（AWS）和波療法戈帕爾薩米（AWS）創建

環境：PoC 或試點	資料來源:VSAM	目標：資料庫
R 型：重新建築	工作負載：IBM	技術：大型主機；現代化
AWS 服務：Amazon MSK；Amazon RDS；AWS 大型主機現代化		

Summary

此模式說明如何使用「精確 Connec t」，將虛擬儲存存取方法 (VSAM) 檔案從大型主機移轉和複寫到 AWS 雲端中的目標環境。此模式涵蓋的目標環境包括 Amazon Relational Database Service 服務 (Amazon RDS) 和 Amazon 阿帕奇卡夫卡 (Amazon MSK) 的亞馬遜受管串流。Connect 使用變更資料擷取 (CDC) 持續監控來源 VSAM 檔案的更新，然後將這些更新傳輸到一或多個 AWS 目標環境。您可以使用此模式來滿足應用程式現代化或資料分析目標。例如，您可以使用 Connect 將 VSAM 應用程式檔案以低延遲遷移到 AWS 雲端，或將 VSAM 資料遷移到 AWS 資料倉儲或資料湖，以進行可容忍應用程式現代化所需要高於同步延遲的分析。

先決條件和限制

先決條件

- IBM z/OS 第一版或更新版本
- 適用於 z/OS (CICS TS) 第 5.1 版或更新版本的 CICS 交易伺服器 (CIC/VS AM 資料擷取)
- IBM MQ 8.0 或更新版本
- 符合 z/OS 安全性需求 (例如，SQData 載入程式庫的 APF 授權)
- 已開啟 VSAM 復原記錄
- (選用) CICS VSAM 復原版本 (CICS 虛擬實境) 可自動擷取疾病控制中心記錄
- 有效的 AWS 帳戶

- Amazon Virtual Private Cloud (VPC)，其中包含可由您的舊式平台存取的子網路
- 從精確的 VSAM Connect 許可證

限制

- Connect 不支援根據來源 VSAM 結構描述或撰稿自動建立目標資料表。您必須第一次定義目標資料表結構。
- 對於非串流目標 (例如 Amazon RDS)，您必須在套用引擎組態指令碼中指定轉換來源為目標對應。
- 記錄、監控和警示功能是透過 API 實作，且需要外部元件 (例如 Amazon CloudWatch) 才能完全運作。

產品版本

- 適用於 z/OS 的 SQ資料 40134
- SQDATA 4.0.43 適用於 Amazon Amazon 彈性計算雲（Amazon EC2）上的亞馬遜亞馬遜機器映像（AMI）

架構

源, 技術, 堆棧

- Job 控制語言 (JCL)
- z/OS Unix 外殼和互動式系統生產力設備 (ISPF)
- VSAM 公用程式

目標技術堆疊

- Amazon EC2
- Amazon MSK
- Amazon RDS
- Amazon VPC

目標架構

將 VSAM 文件遷移到 Amazon RDS

下圖顯示如何透過在來源環境 (現場部署大型主機) 和目標環境 (AWS 雲端) 中使用 CDC 代理程式/發行者，即時或近乎即時地將 VSAM 檔案遷移到關聯式資料庫，例如 Amazon RDS。

圖表顯示下列批次工作流程：

1. Connect 通過比較備份文件中的 VSAM 文件以識別更改來捕獲文件的更改，然後將更改發送到日誌流。
2. 發行者會使用系統記錄串流中的資料。
3. 發行者會透過 TCP/IP 將擷取的資料變更通訊至目標引擎。「控制器協助程式」會驗證來源與目標環境之間的通訊。
4. 目標環境中的套用引擎會接收來自發行者代理程式的變更，並將其套用至關聯式或非關聯式資料庫。

圖表顯示下列線上工作流程：

1. Connect 會使用記錄複寫來擷取線上檔案中的變更，然後將擷取的變更串流至記錄串流。
2. 發行者會使用系統記錄串流中的資料。
3. 發行者會透過 TCP/IP 將擷取的資料變更通訊至目標引擎。「控制器協助程式」會驗證來源與目標環境之間的通訊。
4. 目標環境中的套用引擎會從發行者代理程式接收變更，然後將變更套用至關聯式或非關聯式資料庫。

將 VSAM 文件遷移到 Amazon MSK

下圖顯示如何以高效能模式將 VSAM 資料結構從大型主機串流到 Amazon MSK，以及如何自動產生與 Amazon MSK 整合的 JSON 或 AVRO 結構描述轉換。

圖表顯示下列批次工作流程：

1. Connect 捕獲更改通過使用 CICS VR 或通過比較 VSAM 文件從備份文件來識別變化的文件。擷取的變更會傳送至記錄串流。
2. 發行者會使用系統記錄串流中的資料。
3. 發行者會透過 TCP/IP 將擷取的資料變更通訊至目標引擎。「控制器協助程式」會驗證來源與目標環境之間的通訊。

4. 以 parallel 處理模式操作的複製器引擎會將資料分割至工作快取單位。
5. 工作者執行緒會擷取快取中的資料。
6. 資料會從工作者執行緒發佈至 Amazon MSK 主題。
7. 使用者透過使用連接器，將 Amazon MSK 的變更套用至 Amazon DynamoDB、Amazon Simple Storage Service (Amazon S3) 或 Amazon OpenSearch 服務等目標。

圖表顯示下列線上工作流程：

1. 使用記錄複製來擷取線上檔案中的變更。擷取的變更會串流至記錄串流。
2. 發行者會使用系統記錄串流中的資料。
3. 發行者會透過 TCP/IP 將擷取的資料變更通訊至目標引擎。「控制器協助程式」會驗證來源與目標環境之間的通訊。
4. 以 parallel 處理模式操作的複製器引擎會將資料分割至工作快取單位。
5. 工作者執行緒會擷取快取中的資料。
6. 資料會從工作者執行緒發佈至 Amazon MSK 主題。
7. 使用者透過使用連接器，將 Amazon MSK 的變更套用至 DynamoDB、Amazon S3 或 OpenSearch 服務等目標。

工具

- 適用 Managed Streaming for Apache Kafka (Amazon MSK) 是一項全受管服務，可協助您建置和執行使用 Apache Kafka 處理串流資料的應用程式。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。

史诗

準備來源環境 (大型主機)

任務	描述	所需技能
安裝 Connect 疾病控制中心 4.1.	1. 請聯絡精確 Support 團隊以取得授權與安裝套件。	IBM 大型主機開發人員/管理員

任務	描述	所需技能
	2. 使用範例 JCL 來安裝 Connect 疾控中心 4.1。如需指示，請參閱精確說明文件中的使用 JCL 安裝 Connect CDC (SQData)。 3. 執行此命SETPROG APF令以授權「Connect 載入程式庫」。	
設定 ZF 目錄。	若要設定 ZF 目錄，請遵循 Exact 文件中 ZF 變數目錄中的指示。 注意：控制器協助程式和擷取/發行者代理程式組態儲存在 z/OS UNIX 系統服務檔案系統 (稱為 ZF) 中。控制器協助程式、擷取、儲存和發行者代理程式需要預先定義的 ZF 目錄結構，才能儲存少量檔案。	IBM 大型主機開發人員/管理員
設定 TCP/IP 連接埠。	若要設定 TCP/IP 連接埠，請遵循精確說明文件中 TCP/IP 連接埠的指示。 注意：「控制器協助程式」需要來源系統上的 TCP/IP 連接埠。目標系統上的引擎會參考連接埠 (其中會處理擷取的變更資料)。	IBM 大型主機開發人員/管理員

任務	描述	所需技能
建立 z/OS 記錄串流。	若要建立 z/OS 記錄串流，請遵循精確說明文件中建立 z/OS 系統記錄串流中的指示。 備註：Connect 會使用記錄串流在移轉期間擷取來源環境和目標環境之間的資料並串流。 如需建立 z/OS 的 JCL 範例 LogStream，請參閱精確說明文件中的建立 z/OS 系統記錄串流。	IBM 大型主機開發人員
識別並授權 ZF 使用者的 ID 以及已開始的工作。	使用 RACF 來授與 OMVS ZF 檔案系統的存取權。如需 JCL 範例，請參閱精確說明文件中的識別和授權 ZF 使用者和啟動工作 ID。	IBM 大型主機開發人員/管理員
產生 z/OS 公開/私密金鑰和授權金鑰檔案。	運行 JCL 以生成 key pair。如需範例，請參閱此模式的其他資訊一節中的金鑰配對範例。 如需指示，請參閱 Exact 文件中的產生 z/OS 公開金鑰和私密金鑰以及授權金鑰檔案。	IBM 大型主機開發人員/管理員

任務	描述	所需技能
激活 CICS VSAM 日誌複製並將其附加到日誌流。	執行下列 JCL 指令碼： `//STEP1 EXEC PGM=IDCAMS` `//SYSPRINT DD SYSOUT=*` `//SYSIN DD *` `ALTER SQDATA.CICS.FILEA -` `LOGSTREAMID(SQDATA.VSAMCDC.LOG1) -` `LOGREPLICATE`	IBM 大型主機開發人員/管理員
通過 FCT 激活 VSAM 文件恢復日誌。	修改檔案控制表 (FCT) 以反映下列參數變更： `Configure FCT Parms` `CEDA ALT FILE(name) GROUP(groupname)` `DSNAME(data set name)` `RECOVERY(NONE|BACKOUTONLY|ALL)` `FWDRECOVLOG(NO|1-99)` `BACKUPTYPE(STATIC|DYNAMIC)` `RECOVERY PARAMETERS` `RECOVery : None | Backoutonly | All` `Fwdrecovlog : No | 1-99` `BAckuptype : Static | Dynamic`	IBM 大型主機開發人員/管理員

任務	描述	所需技能
設定發行者代理程式的 CD CzLog 。	1. 建立 CD CzLog 發行者 CAB 檔案。 2. 加密已發佈的資料。 3. 準備 CD 發行CzLog 者執行階段 JCL。	IBM 大型主機開發人員/管理員
啟動控制器協助程式。	1. 開啟 ISPF 面板並執行下列命令以開啟精確選單：`EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. 若要設定「控制器協助程式」，請從功能表中選擇選項 2。	IBM 大型主機開發人員/管理員
啟用發行者。	1. 開啟 ISPF 面板並執行下列命令以開啟精確選單：`EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. 要設置發布者，請從菜單中選擇選項 3，然後選擇 I 進行插入。	IBM 大型主機開發人員/管理員
啟用記錄串流。	1. 開啟 ISPF 面板並執行下列命令以開啟精確選單：`EXEC 'SQDATA.V4nnnnn.ISPFLIB(SQDC$STA)' 'SQDATA.V4nnnnn'` 2. 要設置日誌流，請從菜單中選擇選項 4，然後選擇 I 進行插入。然後，輸入在上述步驟中建立的記錄資料流的名稱。	IBM 大型主機開發人員/管理員

準備目標環境 (AWS)

任務	描述	所需技能
在 EC2 執行個體上精確安裝。	要在 Amazon Amazon EC2 的亞馬遜 Linux AMI 上安裝精確 Connect，請按照在 UNIX 上安裝 Connect CDC（SQ數據）的說明進行操作。	一般 AWS
開啟 TCP/IP 連接埠。	若要修改安全性群組以包含輸入和輸出存取的控制器協助程式連接埠，請遵循 Expection 說明文件中 TCP/IP 的指示。	一般 AWS
創建文件目錄。	若要建立檔案目錄，請遵循 Exact 文件中準備目標套用環境中的指示。	一般 AWS
建立套用引擎組態檔案。	在「套用引擎」的工作目錄中建立「套用引擎」組態檔案。下面的示例配置文件顯示阿帕奇卡夫卡作為目標： `builtin.features=SASL_SCRAM` `security.protocol=SASL_SSL` `sasl.mechanism=SCRAM-SHA-512` `sasl.username=` `sasl.password=` `metadata.broker.list=` 注意：如需詳細資訊，請參閱 Apache Kafka 文件中的安全性。	一般 AWS

任務	描述	所需技能
建立用於套用引擎處理的指令碼。	建立套用引擎的指令碼，以處理來源資料並將來源資料複製到目標。如需詳細資訊，請參閱 Exact 文件中的建立套用引擎指令碼。	一般 AWS
執行指令碼。	使用SQDPARSE和SQDENG命令執行指令碼。如需詳細資訊，請參閱 Exact 文件中的剖析 ZoS 指令碼。	一般 AWS

驗證環境

任務	描述	所需技能
驗證 VSAM 文件和 CDC 處理目標表的列表。	1. 驗證 VSAM 檔案，包括複寫記錄、復原記錄檔、FCT 參數和記錄資料流。 2. 驗證目標數據庫表，包括是否根據所需的模式定義，表訪問和其他條件創建表。	一般 AWS、大型主機
確認 Connect CDC SQ資料產品已連結。	執行測試工作，並確認此工作的傳回碼為 0 (成功)。 注意：Connect 線 CDC SQData 套用引擎狀態訊息應該會顯示作用中的連線訊息。	一般 AWS、大型主機

運行和驗證測試用例（Batch）

任務	描述	所需技能
在大型主機中執行批次工作。	使用修改後的 JCL 執行批次應用程式工作。在修改後的 JCL 中包含執行以下操作的步驟： 1. 備份數據文件。 2. 將備份文件與修改後的數據文件進行比較，生成增量文件，然後記下消息中的增量記錄計數。 3. 將差異檔案推送至 z/OS 記錄串流。 4. 運行 JCL。如需 JCL 範例，請參閱精確文件中的準備檔案比較擷取 JCL。	一般 AWS、大型主機
檢查日誌流。	檢查記錄流，確認您可以看到已完成大型主機批次工作的變更資料。	一般 AWS、大型主機
驗證源增量變化和目標表的計數。	若要確認記錄已結算，請執行下列動作： 1. 從批次 JCL 訊息收集來源差異計數。 2. 監控在 VSAM 文件中插入，更新或刪除的記錄數的記錄級別計數應用引擎。 3. 查詢目標資料表的記錄計數。 4. 比較並計算所有不同的記錄計數。	一般 AWS、大型主機

運行和驗證測試用例（在線）

任務	描述	所需技能
在 CICS 區域中執行線上交易。	1. 運行在線事務以驗證測試用例。 2. 驗證交易執行程式碼 (RC=0 — 成功)。	IBM 大型主機開發人員
檢查日誌流。	確認記錄資料流已填入特定的記錄層級變更。	AWS 大型主機開發人員
驗證目標資料庫中的計數。	監視「套用引擎」的記錄層級計數。	恰恰，Linux
驗證目標數據庫中的記錄計數和數據記錄。	查詢目標資料庫以驗證記錄計數和資料記錄。	一般 AWS

相關資源

- VSAM z/OS (精確說明文件)
- 應用引擎（精確的文檔）
- 複製器引擎 (精確說明文件)
- 記錄資料流 (IBM 說明文件)

其他資訊

組態檔案範例

這是日誌串流的範例組態檔案，其中來源環境是大型主機，目標環境為 Amazon MSK：

```
 -- JOBNAME -- PASS THE SUBSCRIBER NAME
 -- REPORT  progress report will be produced after "n" (number) of Source records
processed.

 JOBNAME VSMTOKFK;
 --REPORT EVERY 100;
```

```
 -- Change Op has been ‘I’ for insert, ‘D’ for delete , and ‘R’ for Replace. For RDS
it is 'U' for update
 -- Character Encoding on z/OS is Code Page 1047, on Linux and UNIX it is Code Page
819 and on Windows, Code Page 1252
 OPTIONS
 CDCOP('I', 'U', 'D'),
 PSEUDO NULL = NO,
 USE AVRO COMPATIBLE NAMES,
 APPLICATION ENCODING SCHEME = 1208;

 --       SOURCE DESCRIPTIONS

 BEGIN GROUP VSAM_SRC;
 DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
 END GROUP;

 --       TARGET DESCRIPTIONS

 BEGIN GROUP VSAM_TGT;
 DESCRIPTION COBOL ../copybk/ACCOUNT AS account_file;
 END GROUP;

 --       SOURCE DATASTORE (IP & Publisher name)

 DATASTORE cdc://10.81.148.4:2626/vsmcdct/VSMTOKFK
 OF VSAMCDC
 AS CDCIN
 DESCRIBED BY GROUP VSAM_SRC ACCEPT ALL;

 --       TARGET DATASTORE(s) - Kafka and topic name

 DATASTORE 'kafka:///MSKTutorialTopic/key'
 OF JSON
 AS CDCOUT
 DESCRIBED BY GROUP VSAM_TGT FOR INSERT;

 --       MAIN SECTION

 PROCESS INTO
 CDCOUT
 SELECT
 {
 SETURL(CDCOUT, 'kafka:///MSKTutorialTopic/key')
```

```
  REMAP(CDCIN, account_file, GET_RAW_RECORD(CDCIN, AFTER), GET_RAW_RECORD(CDCIN,
 BEFORE))
  REPLICATE(CDCOUT, account_file)
  }
  FROM CDCIN;
```

金鑰配對範例

這是如何運行 JCL 來生成 key pair 的例子：

```
//SQDUTIL EXEC PGM=SQDUTIL //SQDPUBL DD DSN=&USER..NACL.PUBLIC, //
 DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //
 SPACE=(TRK,(1,1)) //SQDPKEY DD DSN=&USER..NACL.PRIVATE, //
 DCB=(RECFM=FB,LRECL=80,BLKSIZE=21200), // DISP=(,CATLG,DELETE),UNIT=SYSDA, //
 SPACE=(TRK,(1,1)) //SQDPARMS DD  keygen //SYSPRINT DD SYSOUT= //SYSOUT DD SYSOUT=* //
SQDLOG DD SYSOUT=* //*SQDLOG8 DD DUMMY
```

使用 OpenText 微焦點企業伺服器和 LRS X 在 AWS 上現代化大型主機輸出管理 PageCenter

創建者：舒本羅伊（AWS），亞伯拉罕·朗登（微焦點）和蓋伊·塔克（列維，雷和壽普公司）

環境：PoC 或試點	資料來源：IBM 大型主機	目標：AWS
R 類型：重新平台	工作負載：IBM	技術：大型主機；移轉；現代化
AWS 服務：AWS 管理 Microsoft AD；Amazon EC2；Amazon FSx for Windows File Server；Amazon RDS；AWS 大型主機現代化		

Summary

透過現代化您的大型主機輸出管理，您可以透過 Amazon Web Services (AWS) 雲端原生技術來節省成本、減輕維護舊有系統的技術債務，以 DevOps 及改善彈性和靈活性。此模式說明如何在 AWS 雲端上將關鍵業務大型主機輸出管理工作負載現代化。該模式使用OpenText 微焦點企業服務器作為一個現代化的大型主機應用程序的運行時，利維，雷和壽普公司（LRS）VPSX/MFI（微焦點接口）作為打印服務器和 LRS X 作為歸檔服務器。 PageCenterLRS PageCenter X 提供輸出管理解決方案，可用於檢視、索引、搜尋、封存和保護企業輸出的存取。

該模式以重新平台大型主機現代化方法為基礎。大型主機應用程式是透過 AWS 大型主機現代化在 Amazon Elastic Compute Cloud (Amazon EC2) 上進行遷移。大型主機輸出管理工作負載會遷移至 Amazon EC2，而大型主機資料庫 (例如 IBM Db2 for z/OS) 則會遷移至 Amazon Relational Database Service 服務 (Amazon RDS)。LRS 目錄整合伺服器 (LRS/DIS) 與適用於 Microsoft Active Directory 的 AWS Directory Service 搭配使用，以進行輸出管理工作流程身份驗證和授權。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 大型主機輸出管理工作負載。

- 有關如何重建和交付在 OpenText Micro Focus 企業伺服器上執行的大型主機應用程式的基本知識。如需詳細資訊，請參閱 OpenText Micro Focus 文件中的企業伺服器資料表。
- LRS 雲打印解決方案和概念的基礎知識。如需詳細資訊，請參閱 LRS 文件中的輸出現代化。
- 微焦點企業服務器軟件和許可證。如需詳細資訊，請聯絡 OpenText 微焦點銷售人員。
- LRS VPSX/MFI，LRS PageCenter X，LRS /列和LRS/DIS 軟件和許可證。更多產品資訊請洽 LRS。您必須提供將安裝 LRS 產品的 EC2 執行個體的主機名稱。

備註：如需有關大型主機輸出管理工作負載之組態考量的詳細資訊，請參閱此模式的其他資訊一節中的考量事項。

產品版本

- OpenText 微焦點企業伺服器 8.0 或更新版本
- LRS VPSX/MFI
- LRS PageCenter X V1R3 或更高版本

架構

源, 技術, 堆棧

- 作業系統 —
- 程式語言 — 通用面向業務的語言 (COBOL)、Job 控制語言 (JCL) 和客戶資訊控制系統 (CICS)
- 資料庫 — IBM Db2 (適用於 z/OS)、IBM 資訊管理系統 (IMS) 資料庫，以及虛擬儲存存取方法 (VSAM)
- 安全性 — 資源存取控制設施 (RACF)、z/OS 的 CA 最高機密，以及存取控制設施 2 (ACF2)
- 列印與封存解決方案 — IBM 大型主機 z/OS 輸出與列印產品 (IBM Inforint 伺服器，適用於 z/OS、LRS 和 CA 交付) 和封存解決方案 (CA 遞送、ASG 莫比烏斯或 CA 服務包)

來源架構

下圖顯示大型主機輸出管理工作負載的典型目前狀態架構。

該圖顯示以下工作流程：

1. 使用者在以 COBOL 撰寫之 IBM CICS 應用程式建置的參與系統 (SoE) 上執行商業交易。
2. SoE 會叫用大型主機服務，該服務會將商業交易資料記錄在 system-of-records (sR) 資料庫中，例如 IBM Db2 for z/OS。
3. SoR 保留了來自 SoE 的業務數據。
4. 批次工作排程器會啟動批次工作以產生列印輸出。
5. 批次工作會從資料庫中擷取資料。它根據業務需求格式化數據，然後產生業務輸出，如帳單，身份證，或貸款報表。最後，批次工作會根據業務需求，將輸出路由傳送至輸出管理，以進行輸出格式、發佈和儲存。
6. 輸出管理從批次工作接收輸出。輸出管理索引、排列及發佈輸出至輸出管理系統中的指定目的地，例如 LRS PageCenter X 解決方案 (如此模式所示) 或 CA View。
7. 用戶可以查看，搜索和檢索輸出。

目標技術堆疊

- 操作系統 — 在 Amazon EC2 上運行的視窗服務器
- 運算 — Amazon EC2
- 儲存 — 適用於 Windows 檔案伺服器的 Amazon 彈性區塊存放區 (Amazon EBS) 和亞馬遜 FSx
- 編程語言-COBOL，JCL 和 CICS
- 數據庫 — Amazon RDS
- 安全性 — AWS 管理 Microsoft AD
- 列印和存檔 — AWS 上的 LRS 列印 (VPSX) 和存檔 (PageCenterX) 解決方案
- 大型主機執行階段環境 — OpenText 微焦點企業伺服器

目標架構

下圖顯示部署在 AWS 雲端的大型主機輸出管理工作負載的架構。

該圖顯示以下工作流程：

1. 批次工作排程器會啟動批次工作以建立輸出，例如帳單對帳單、ID 卡或貸款對帳單。
2. 大型主機批次任務 (重新組成 Amazon EC2) 會使用 OpenText Micro Focus 企業伺服器執行階段從應用程式資料庫擷取資料、將商業邏輯套用至資料，以及格式化資料。然後，它會使用 OpenText Micro Focus 印表機結束模組 (OpenText Micro Focus 文件) 將資料傳送至輸出目的地。

3. 應用程式資料庫 (在 Amazon RDS 上執行的 SoR) 會保留列印輸出的資料。
4. LRS VPSX/MFI 列印解決方案部署在 Amazon EC2 上，其操作數據存儲在 Amazon EBS 中。LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /隊列傳輸代理程式，透過微焦點 JES 列印結束 API 收集輸出資料。 OpenText

   LRS VPSX/MFI 不進行數據預處理，如 EBCDIC 轉換為 ASCII 轉換。它還可以執行更複雜的工作，包括將大型主機專用的資料串流 (例如 IBM 進階功能簡報 (AFP) 和施樂線條條件資料流 (LCDS) 轉換為更常見的檢視和列印資料串流，例如印表機命令語言 (PCL) 和 PDF。

   在 LRS PageCenter X 的維護期間，LRS VPSX/MFI 會保留輸出佇列並做為輸出佇列的備份。LRS VPSX/MFI 使用 LRS /列協議連接並將輸出發送到LRS PageCenter X。LRS /queue 會執行工作的就緒和完成交換，以協助確保資料傳輸發生。

   備註：

   如需有關從 OpenText Micro Focus 列印結束傳送至 LRS /列和LRS VPSX/MFI 支援大型主機批次機制的列印資料的詳細資訊，請參閱其他資訊一節中的列印資料擷取。

   LRS VPSX/MFI 可以在印表機-機隊層級執行健康狀態檢查。如需詳細資訊，請參閱此模式的其他資訊一節中的印表機-機隊健康狀態檢查。
5. LRS PageCenter X 輸出管理解決方案部署在 Amazon EC2 上，其操作資料會儲存在適用於 Windows 檔案伺服器的 Amazon FSx 中。LRS PageCenter X 提供了一個集中的報表管理系統，其中包含匯入到 LRS PageCenter X 的所有檔案，以及所有能夠存取檔案的使用者。使用者可以檢視特定的檔案內容，或在多個檔案中搜尋符合條件。

   LRS/NETX 元件是一種多執行緒 Web 應用程式伺服器，可為 LRS PageCenter X 應用程式和其他 LRS 應用程式提供通用的執行階段環境。LRS /網頁連接元件安裝在您的web 伺服器上，並提供從 Web 伺服器到 LRS/Netx Web 應用程式伺服器的 Connect 器。
6. LRS PageCenter X 提供檔案系統物件的儲存空間。LRS PageCenter X 的操作數據存儲在 Amazon FSx 中，適用於 Windows 文件服務器。
7. 輸出管理身份驗證和授權由 AWS 管理 Microsoft AD 與 LRS/DIS 執行。

注意：目標解決方案通常不需要變更應用程式即可配合大型主機格式化語言，例如 IBM AFP 或 Xerox LCDS。

AWS 基礎設施架構

下圖顯示適用於大型主機輸出管理工作負載的高可用性和安全 AWS 基礎設施架構。

該圖顯示以下工作流程：

1. 批次排程器會啟動批次程序，並跨多個可用區域部署在 Amazon EC2 上，以實現高可用性 (HA)。

   注意：此模式不涵蓋批次排程器的實作。如需有關實作的詳細資訊，請參閱排程器的軟體廠商說明文件。

2. 大型主機批次工作 (以 JCL 或 COBOL 等程式設計語言撰寫) 使用核心業務邏輯來處理並產生列印輸出，例如帳單、ID 卡和貸款陳述式。批次任務部署在跨 HA 的兩個可用區域的 Amazon EC2 上。它使用 OpenText 微型聚焦打印退出 API 將打印輸出路由到 LRS VPSX/MFI 進行數據預處理。
3. LRS VPSX/MFI 列印伺服器部署在 Amazon EC2 上，跨兩個 HA 可用區域 (作用中-待命備援配對)。它使用 Amazon EBS 作為操作數據存放區。Network Load Balancer 會在 LRS VPSX/MFI EC2 執行個體上執行健康狀態檢查。如果作用中執行個體處於狀態不良，負載平衡器會將流量路由至其他可用區域中的熱待命執行個體。列印要求會在每個 EC2 執行個體的本機 LRS Job 佇列中保留。發生故障時，必須重新啟動失敗的執行個體，LRS 服務才能繼續處理列印要求。

   注意：LRS VPSX/MFI 也可以在印表機-機隊層級執行健康狀態檢查。如需詳細資訊，請參閱此模式的其他資訊一節中的印表機-機隊健康狀態檢查。

4. LRS PageCenter X 輸出管理是在 Amazon EC2 上部署在 HA 的兩個可用區域 (主動-待命備援配對) 的兩個可用區域。它使用 Amazon FSx for Windows File Server 作為操作數據存儲。如果作用中執行個體處於狀態不良，負載平衡器會對 LRS PageCenter X EC2 執行個體執行健康狀態檢查，並將流量路由到其他可用區域中的待命執行個體。
5. Network Load Balancer 會提供 DNS 名稱，以將 LRS VPSX/MFI 伺服器與 LRS X 整合。PageCenter

   注意：LRS PageCenter X 支援第 4 層負載平衡器。

6. LRS PageCenter X 使用適用於 Windows 檔案伺服器的 Amazon FSx 做為跨 HA 兩個可用區域部署的操作資料存放區。LRS PageCenter X 只能理解檔案共用中的檔案，而不是外部資料庫中的檔案。
7. AWS 受管 Microsoft AD 可與 LRS/DIS 搭配使用，以執行輸出管理工作流程身份驗證和授權。如需詳細資訊，請參閱其他資訊一節中的列印輸出驗證和授權。

工具

AWS 服務

- 適用於 Microsoft 活動目錄的 AWS Directory Service 可讓您的目錄感知工作負載和 AWS 資源在 AWS 雲端中使用 Microsoft 活動目錄。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon EC2 執行個體、容器和 IP 地址。此模式使用 Network Load Balancer。
- Amazon FSx 提供支援業界標準連線協定的檔案系統，並在 AWS 區域提供高可用性和複寫。這種模式使用 Amazon FSx for Windows File Server。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。

其他工具

- LRS PageCenter X 軟體提供可擴充的文件與報表內容管理解決方案，透過自動化索引、加密和進階搜尋功能，協助使用者從資訊中獲得最大價值。
- LRS VPSX/MFI (微型對焦介面) 由 LRS 和微焦點共同開發，可擷取 OpenText 微焦點企業伺服器 JES 捲軸的輸出，並可靠地將其傳送至指定的列印目的地。 OpenText
- LRS /列是以TCP/IP 為基礎的傳輸代理程式。LRS VPSX/MFI 透過微聚焦 JES 列印結束程式設計介面 OpenText 採集或擷取列印資料。
- LRS 目錄整合伺服器 (LRS/DIS) 用於列印工作流程期間的驗證和授權。
- OpenText Micro Focus 企業伺服器是適用於大型主機應用程式的應用程式部署環境。它為移轉或使用任何版本的 OpenText Micro Focus 企業開發人員建立的大型主機應用程式提供執行階段環境。

史诗

設定 OpenText Micro Focus 執行階段並部署大型主機批次應用程式

任務	描述	所需技能
設定執行階段並部署示範應用程式。	若要在 Amazon EC2 上設定 OpenText Micro Focus 企業伺服器並部署 OpenText Micro Focus BankDemo 示範應用程式，請遵循 AWS 大型主機現代化使用者指南中的指示。 此 BankDemo 應用程式是建立並啟動列印輸出的大型主機批次應用程式。	雲端架構師

在 Amazon EC2 上設置 LRS 列印伺服器

任務	描述	所需技能
創建一個 Amazon EC2 窗口實例。	若要啟動 Amazon EC2 Windows 執行個體，請遵循 Amazon EC2 文件中的步驟 1：啟動執行個體中的指示。使用與 LRS 產品授權相同的主機名稱。 您的執行個體必須符合下列 LRS VPSX/MFI 的硬體和軟體需求： • CPU — 雙核心 • 公羊-16 GB • 磁碟機 — 500 GB • 最低 EC2 執行個體數量 `m5.xlarge`	雲端架構師

任務	描述	所需技能
	• 操作系統 — 視窗 • 軟體 — 網際網路資訊服務 (IIS) 或 Apache 注意：先前的硬體和軟體需求適用於小型印表機機群 (約 500-1000)。若要取得完整要求，請諮詢您的 LRS 和 AWS 聯絡人。 1. 建立 Windows 執行個體時，請確認 EC2 主機名稱與 LRS 產品授權所使用的主機名稱相同。 2. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。 3. 在 Windows「開始」功能表上，找到並開啟「伺服器管理員」。 4. 在 [伺服器管理員] 中，選擇 [儀表板]、[快速入門]、[新增角色和功能]，然後選擇 [伺服 5. 在 [伺服器角色] 中，選擇 [WebServer (IIS)]，然後選擇 [應用程式開發]。 6. 在 [應用程式開發] 中，選取 [CGI] 核取方塊。 7. 若要安裝 CGI，請遵循 Windows 伺服器管理員新增角色和功能精靈中的指示。	

任務	描述	所需技能
	8. 在 EC2 執行個體的 Windows 防火牆中開啟連接埠 5500，以進行 LRS /列通訊。	
在 EC2 執行個體上安裝 LRS VPSX/MFI。	1. 連線至 EC2 執行個體。 2. 從您應該收到的 LRS 電子郵件中開啟產品下載頁面的連結。 注意：LRS 產品通過電子文件傳輸（EFT）進行分發。 3. 下載 LRS VPSX/MFI，並解壓縮該文件（默認文件夾： ）。c:\LRS 4. 若要安裝 LRS VPSX/MFI，請從解壓縮的資料夾啟動 LRS 產品安裝程式。 5. 在 [選取功能] 功能表上，選取 [VPSX® 伺服器]，然後選擇 [下一步] 以開始安裝程序。安裝完成後，您將收到成功訊息。	雲端架構師

任務	描述	所需技能
安裝 LRS /列。	1. Connect 到您的 OpenText 微焦點企業伺服器 EC2 執行個體。 2. 從您應該收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS / Queue，然後解壓縮檔案。 3. 導覽至您下載檔案的位置，然後啟動 LRS 產品安裝程式以安裝 LRS /列。 4. 遵循 LRS 產品安裝程式中的指示完成安裝程序。	雲端架構師

任務	描述	所需技能
安裝 LRS/DIS。	LRS/DIS 產品通常包含在 LRS VPSX 安裝中。但是，如果 LRS/DIS 未與 LRS VPSX 一起安裝，請使用以下步驟進行安裝： 1. Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS/DIS，然後解壓縮檔案。 3. 導覽至您下載檔案的位置，然後啟動 LRS 產品安裝程式。 4. 在「LRS 產品安裝程式」中，展開「LRS 雜項工具」，選取「LRS DIS」，然後選擇「下一步」。 5. 遵循 LRS 產品安裝程式中的其餘指示，以完成安裝程序。	雲端架構師

任務	描述	所需技能
建立目標群組。	依照為 Network Load Balancer 建立目標群組中的指示，建立目標群組。建立目標群組時，請將 LRS VPSX/MFI EC2 執行個體註冊為目標： 1. 在 [指定群組詳細資訊] 頁面上，選擇執行處理做為 [選擇目標類型]。 2. 針對通訊協定，選擇 TCP。 3. 在「連接埠」中，選擇 5500。 4. 在 [註冊目標] 頁面的 [可用執行個體] 區段中，選取 LRS VPSX/MFI EC2 執行個體。	雲端架構師

任務	描述	所需技能
建立 Network Load Balancer。	若要建立 Network Load Balancer，請遵循 E lastic Load Balancing 說明文件中的指示。您的 Network Load Balancer 會將流量從 OpenText 微焦點企業伺服器路由到 LRS VPSX/MFI EC2 執行個體。 建立 Network Load Balancer 時，請在「監聽器和路由」頁面選擇下列值： 1. 針對 Protocol (通訊協定)，選擇 TCP。 2. 在「連接埠」中，選擇 5500。 3. 對於「預設」動作，請為您先前建立的目標群組選擇「轉寄至」。	雲端架構師

整合 OpenText 微焦企業伺服器與LRS /隊列和LRS VPSX/MFI

任務	描述	所需技能
設定微焦點企業伺服器以進行 LRS /列整合。	1. 按照 Amazon EC2 文件中的指示，Connect 到您的 OpenText 微焦點企業伺服器 EC2 執行個體。 2. 在 Windows [開始] 功能表上，開啟 OpenText Micro Focus 企業伺服器管理使用者介面。	雲端架構師

任務	描述	所需技能
	3. 在選單列中，選擇「原生」。 4. 在瀏覽窗格中，選擇 [目錄伺服器]，然後為您的 [企業伺服器] 區域選擇 BANKDEMO。 5. 從左側導覽窗格的 [一般] 中，向下捲動至 [其他] 區段，以設定環境變數 (LRSQ_ADDRESS LRSQ_PORT 、、LRSQ_COMMAND) 以指向 LRSQ。 • 對於 LRSQ_ADDRESS，請輸入您先前建立之 Network Load Balancer 的 IP 位址或 DNS 名稱。 • 針對連接埠，請輸入 VPSX L RSQ 接聽程式連接埠 (5500)。 • 對於 LRSQ_ 指令，請輸入 LRSQ 可執行檔的路徑位置。 注意：LRS 目前支援的 DNS 名稱上限為 50 個字元。如果您的 DNS 名稱長度超過 50 個字元，則可以使用 Network Load Balancer 的 IP 位址作為替代方案。	

任務	描述	所需技能
為 LRS VPSX/MFI 整合設定 OpenText 微型焦點企業伺服器。	1. 將`VPSX_MFI_R2` 資料夾從 LRS VPSX/MFI 安裝程式複製到微焦點企業伺服器位置 (位於)。`C\BANKDEMO\print` 2. 按照 Amazon EC2 文件中的指示，Connect 到您的微焦點企業伺服器 EC2 執行個體。 3. 在 Windows [開始] 功能表上，開啟 Micro Focus 企業伺服器管理使用者介面。 4. 在功能表列上，選擇 [原生]。 5. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO]。 6. 在「銀行示範」下，選擇 JES。 7. 在「JES 程式路徑」下，從中`C\BANKDEMO\print` 加入`DLL(VPSX_MFI_R2)` 路徑。	雲端架構師

設定列印佇列和列印使用者

任務	描述	所需技能
將 OpenText Micro Focus 列印結束模組與 Micro Focus 企業伺服器批次印表機伺服器執行程序產生關聯。	1. 按照 Amazon EC2 文件中的指示，Connect 到您的 OpenText 微焦點企業伺服器 EC2 執行個體。	雲端架構師

任務	描述	所需技能
	2. 在 Windows [開始] 功能表上，開啟 OpenText Micro Focus 企業伺服器管理使用者介面。 3. 在功能表列上，選擇 [原生]。 4. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO]。 5. 在「銀行示範」下，選擇 JES，然後向下捲動至「印表機」。 6. 在印表機中，將 OpenText 微型聚焦列印結束模組 (LRSPRTE6 用於 Batch) 與 OpenText Micro Focus 企業伺服器批次印表機伺服器執行程序 (SEP) 產生關聯。這樣可以將列印輸出路由傳送至 LRS VPSX/MFI。 如需有關組態的詳細資訊，請參閱 OpenText Micro Focus 文件中的使用結束。	

任務	描述	所需技能
在 LRS VPSX/MFI 中建立列印輸出佇列，並將其與 LRS X 整合。 PageCenter	1. Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗開始功能表上，開啟 VPSX 網頁介面。 3. 在導覽窗格中，選擇 [印表機]。 4. 選擇 [新增]，然後選擇 [新增印表機]。 5. 在 [印表機規劃] 頁面上，輸入 L oc al 做為「印表機名稱」。 6. 針對 VPSX 識別碼，請輸入 VPS1。 7. 對於 CommType，請選取「TCPIP /LR SQ」。 8. 對於主機 /IP 位址，請輸入 LRS PageCenter X EC2 執行個體前方的 Network Load Balancer IP 位址。 9. 針對「遠端連接埠」，輸入 5800。 10對於遠端佇列，請輸入將儲存輸出的 LRS PageCenter X 文件資料夾名稱。 11選擇新增。	雲端架構師

任務	描述	所需技能
在 LRS VPSX/MFI 中創建一個打印用戶。	1. Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗開始功能表上，開啟 VPSX 網頁介面。 3. 在功能窗格中，選擇 [安全性]，然後選擇 [使用者]。 4. 在 [使用者名稱] 欄中，選擇 [管理員]，然後選擇 [複製]。 5. 在「使用者設定檔維護」視窗中，對於「使用者名稱」，輸入使用者名稱 (例如，PrintUser)。 6. 在「說明」中，輸入簡短描述 (例如，測試列印的使用者)。 7. 選擇更新。這將創建一個打印用戶（例如，PrintUser）。 8. 在導覽窗格的 [使用者] 下，選擇您建立的新使用者。 9. 在 [命令] 功能表上選擇 [安全性]。 10. 在 [安全性規則] 頁面上，選擇所有適用的印表機安全性和工作安全性選項，然後選擇 [儲存]。 11. 若要將新的列印使用者新增至 [系統管理員] 群組，請在功能窗格上選擇 [安全性]，然後選擇 [設定]。	雲端架構師

任務	描述	所需技能
	12.在「安全性」組態視窗中，將新的列印使用者新增至「管理員」欄。	

在 Amazon EC2 上設置 LRS PageCenter X 服務器

任務	描述	所需技能
創建一個 Amazon EC2 窗口實例。	依照步驟 1 中的指示啟動 Amazon EC2 Windows 執行個體：啟動 Amazon EC2 文件中的執行個體。使用與 LRS 產品授權相同的主機名稱。 您的執行個體必須符合 LRS PageCenter X 的下列硬體和軟體需求： • CPU — 雙核心 • 公羊-16 GB • 磁碟機 — 500 GB • 最低 EC2 執行個體數量 `m5.xlarge` • 操作系統 — 視窗 • 軟件-IIS 或阿帕奇 注意：先前的硬體和軟體需求適用於小型印表機群 (約 500—1000)。若要取得完整要求，請諮詢您的 LRS 和 AWS 聯絡人。 1. 建立 Windows 執行個體時，請確認 EC2 主機名稱與	雲端架構師

任務	描述	所需技能
	LRS 產品授權所使用的主機名稱相同。 2. 按照 Amazon EC2 文件中的指示 Connect 到您的 EC2 執行個體。 3. 在 Windows「開始」功能表上，找到並開啟「伺服器管理員」。 4. 在伺服器管理員中，選擇儀表板、快速入門、新增角色和功能，然後選擇伺服器角色。 5. 在 [伺服器角色] 中，選擇 [WebServer (IIS)]，然後選擇 [應用程式開發]。 6. 在 [應用程式開發] 中，選取 [CGI] 核取方塊。 7. 若要安裝 CGI，請遵循 Windows 伺服器管理員新增角色和功能精靈中的指示。 8. 在 EC2 執行個體的 Windows 防火牆中，為輸入 TCP/IP 流量開啟連接埠 5800。LRS VPSX 使用 5800 連接埠上的 TCPIP 通訊協定與 LRS X 通訊協定進行通訊。 PageCenter	

任務	描述	所需技能
在 EC2 執行個體上安裝 LRS PageCenter X。	1. 連線至 EC2 執行個體。 2. 從您應該收到的 LRS 電子郵件中開啟產品下載頁面的連結。 注意：LRS 產品通過電子文件傳輸（EFT）進行分發。 3. 下載 LRS PageCenter X，然後解壓縮檔案（預設資料夾：`c:\LRS`）。 4. 若要安裝 LRS PageCenter X，請從解壓縮的資料夾啟動 LRS 產品安裝程式。 5. 在 [選取功能] 功能表上，選取 [PageCenterX]，然後選擇 [下一步] 以開始安裝程序。安裝完成後，您將收到成功訊息。	雲端架構師

任務	描述	所需技能
安裝 LRS/DIS。	LRS/DIS 產品通常包含在 LRS VPSX 安裝中。但是，如果 LRS/DIS 未與 LRS VPSX 一起安裝，請使用以下步驟進行安裝： 1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS/DIS，然後解壓縮檔案。 3. 導覽至您下載檔案的位置，然後啟動 LRS 產品安裝程式。 4. 在「LRS 產品安裝程式」中，展開「LRS 雜項工具」，選取「LRS DIS」，然後選擇「下一步」。 5. 遵循 LRS 產品安裝程式中的其餘指示，以完成安裝程序。	雲端架構師

任務	描述	所需技能
建立目標群組。	依照為 Network Load Balancer 建立目標群組中的指示，建立目標群組。建立目標群組時，請將 LRS PageCenter X EC2 執行個體註冊為目標： 1. 在 [指定群組詳細資訊] 頁面上，選擇執行處理做為 [選擇目標類型]。 2. 針對通訊協定，選擇 TCP。 3. 對於「連接埠」，請選擇 5800。 4. 在 [註冊目標] 頁面的 [可用執行個體] 區段中，選取 LRS PageCenter X EC2 執行個體。	雲端架構師

任務	描述	所需技能
建立 Network Load Balancer。	若要建立 Network Load Balancer，請遵循 E lastic Load Balancing 說明文件中的指示。您的 Network Load Balancer 會將流量從 LRS VPSX/MFI 路由傳送至 LRS X EC2 執行個體。 PageCenter 建立 Network Load Balancer 時，請在「監聽器和路由」頁面選擇下列值： 1. 針對 Protocol (通訊協定)，選擇 TCP。 2. 對於「連接埠」，請選擇 5800。 3. 對於「預設」動作，請為您先前建立的目標群組選擇「轉寄至」。	雲端架構師

在 LRS X 中設定輸出管理功能 PageCenter

任務	描述	所需技能
在 LRS PageCenter X 中啟用匯入功能。	您可以使用 LRS PageCente r X 匯入功能，透過 Job 名稱或表單 ID 等準則識別 LRS PageCenter X 上的輸出。然後，您可以將輸出路由到 LRS PageCenter X 中的特定資料夾。 若要啟用「匯入」功能，請執行下列動作：	雲端架構師

任務	描述	所需技能
	1. 按照 Amazon EC2 文件中的指示，Connect 到您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員。 4. 在 [組態] 頁面上，選擇 [進階] > [匯入參數]。 5. 在「匯入參數」區段中，選取「進階匯入」核取方塊。 6. 若要確認變更，請選擇「更新」。	

任務	描述	所需技能
設定文件保留原則。	LRS PageCenter X 使用文件保留政策來決定在 L PageCenter RS X 中保留文件的時間長度。 若要設定文件保留原則，請執行下列動作： 1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員。 4. 在 [管理員] 頁面上，選擇 [封存群組清單/一般管理員]，然後選擇 [保留原則]。 5. 在 [保留原則] 區段中，選擇 [新增] 以建立保留原則。 6. 在 [保留原則資訊] 頁面上，輸入 [保留原則] 名稱、[說明] 和 [文件保留期間]。 7. 若要儲存變更並建立原則，請選擇 [確定]。	雲端架構師

任務	描述	所需技能
建立規則，將輸出文件路由至 LRS PageCenter X 中的特定資料夾。	在 LRS PageCenter X 中，「目的地」會決定「報表定義」呼叫此目的地時，要傳送輸出的資料夾路徑。在此範例中，根據報表定義中的 [表單 ID] 資料夾建立資料夾，然後將輸出儲存至該資料夾。 1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員、進階匯入、目的地。 4. 在「目的地」區段中，選擇「新增」以開啟「目的地維護」表單。 5. 在「目的地維護」 表單中，輸入下列值： • 目的地名稱 — 表單 • 描述 — 目的地的描述，例如表單式資料夾結構 • 目標類型 — 資料夾 • 資料夾參數 — 匯入資料夾路徑 (當文件到達時，將在 PageCenter X 中建立的資料夾路徑；例如，路徑**`/Test/&FORM/&IMPORTDATE/&IMPORTTIME`** 將建立基本**`Test`**資料夾、以表單 ID 為基礎的子資料	雲端架構師

任務	描述	所需技能
	夾**STD**、以匯入日期為基礎的子資料夾，然後根據匯入時間建立子資料夾) • 文件名稱 — 當文件儲存在資料夾中時，指派給文件的動態名稱。 6. 在下拉式清單中，選擇保留原則。例如，選擇「年度 1」可保留文件 1 年。 7. 若要儲存變更，請選擇 [確定]。	

任務	描述	所需技能
建立報表定義。	1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇 [管理員]、[進階匯入]、[報表定義]，然後選擇 [新增 4. 在 [報表定義維護] 頁面的 [一般] 索引標籤上，輸入報表定義名稱。 5. 在「一般」標籤的「欄位」下，您可以指定選取條件，例如「Job 名稱」、「表單」、「類別」和「作者」。例如，您可以輸入 MFI DEMO 的「Job 名稱」。「工作名稱」(Job Name) 值將是將產生列印輸出的批次工作的名稱。 6. 在「目的地」頁籤的「可用目的地」下，選擇先前建立的目的地 (表單)。 7. 選擇新增，將表單目的地新增為指派的目的地。 備註：此範例包含一個報表定義，其中 MFIDEMO 產生並遞送至 LRS PageCenter X 的輸出會儲存在目標定義中定義的資料夾結構中。	雲端架構師

設定輸出管理的驗證和授權

任務	描述	所需技能
使用使用者和群組建立 AWS 受管 Microsoft AD 網域。	1. 若要在 AWS 受管 Microsoft AD 上建立目錄，請按照建立您的 AWS 受管 Microsoft AD 目錄中的指示操作。 2. 若要部署 EC2 執行個體 (使用中目錄管理員) 並安裝使用中目錄工具來管理您的 AWS 受管 Microsoft AD，請遵循步驟 3：部署 EC2 執行個體以管理您的 AWS 受管 Microsoft AD 中的指示。 3. 若要連接到您的 EC2 執行個體，請遵循 Amazon EC2 文件中的指示進行。 注意：連線至 EC2 執行個體時，在 Windows 安全性視窗中，輸入您在步驟 1 中建立之目錄的管理員登入資料。 4. 在 Windows「開始」功能表的「Windows 系統管理工具」下，選擇「作用中目錄使用者和電腦」。 5. 若要在 Active Directory 網域中建立列印使用者，請遵循建立使用者中的指示。	雲端架構師
將 EC2 執行個體加入 AWS 受管 Microsoft AD 網域。	將 LRS VPSX/MFI 和 LRS PageCenter X EC2 執行個體自動加入您的 AWS 受管 Microsoft AD 網域 (AWS 知	雲端架構師

任務	描述	所需技能
	識中心文件) 或手動 (AWS Directory Service 文件)。	
為 LRS PageCenter X EC2 執行個體設定和整合 AWS 受管 Microsoft AD。	1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員。 4. 在「組態」頁面的「安全參數」段落中，選取 LR S/DIS 做為「安全類型」。 5. 在「安全參數」區段中輸入其餘選項的偏好設定。 6. 在 Windows [開始] 功能表上，開啟 [PageCenterX] 資料夾，選擇 [伺服器啟動]，然後選擇 [伺服器停止]。 7. 使用您的使用中目錄使用者名稱和密碼登入 LRS PageCenter X。	雲端架構師

任務	描述	所需技能
設定匯入群組，以將輸出從 LRS VPSX 匯入至 LRS X。PageCenter	1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員、安全性管理員、群組。 4. 在「群組」區段中，選擇「新增」以開啟「群組」偏好設定表單。 5. 在「群組」偏好設定表單中，輸入「群組名稱」與「描述」的值。 6. 展開 [一般] 選項，然後選取 [匯入] 核取方塊。 7. 若要儲存變更，請選擇 [確定]。	雲端架構師
將安全性規則新增至「匯入」群組。	1. 開啟「匯入」群組的內容 (按一下右鍵) 功能表。 2. 選擇 [進階]，然後選擇 [安全性]。 3. 在「安全性」區段中，選擇「匯入」，然後選取「子資料夾」核取方塊。 4. 若要儲存變更，請選擇「套用」。	雲端架構師

任務	描述	所需技能
在 LRS PageCenter X 中創建一個用戶以執行從 LRS VPSX/MFI 的輸出導入。	當您在 LRS PageCenter X 中建立使用者以執行輸出匯入時，使用者名稱應與 LRS VPSX/MFI 中列印輸出佇列的 VPSX ID 相同。在此範例中，VPSX 識別碼為 VPS1。 1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員、安全管理員、使用者。 4. 選擇 [新增] 開啟 [使用者設定檔維護] 表單。 5. 在使用者設定檔維護中，對於使用者名稱，輸入 VPS1。	雲端架構師

任務	描述	所需技能
將 LRS PageCenter X 匯入使用者新增至僅匯入群組。	若要提供從 LRS VPSX 匯入文件至 LRS X 的必要權限，請執行下列 PageCenter操作： 1. Connect 至您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員、安全性管理員、群組。 4. 在 [群組] 區段中，開啟 [僅匯入] 群組的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [進階]、[安全性]。 5. 在「資料夾安全性記錄」(ImportOnly) 頁面上，選擇「使用者」頁籤。 6. 在 [使用者] 索引標籤的 [名稱] 下，從下拉式清單中選取 [使用者 VPS1]，然後選擇 [套用]。	雲端架構師

任務	描述	所需技能
透過 AWS 管理的 Microsoft AD 為 LRS VPSX/MFI EC2 執行個體設定 LRS/DIS。	1. Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗開始功能表上，開啟 VPSX 網頁介面。 3. 在瀏覽窗格中，選擇 [安全性]，然後選擇 [設定]。 4. 在「安全組態」頁面的「安全參數」段落中，選取 LRS/DIS (外部) 做為「安全類型」。 5. 在「安全參數」區段中輸入其餘選項的偏好設定。 6. 在 Windows [開始] 功能表上，開啟 [LRS 輸出管理] 資料夾，選擇 [伺服器啟動]，然後選擇 [伺服器停止]。 7. 使用您的活動目錄用戶名和密碼登錄到 LRS VPSX/MFI。	雲端架構師

將適用於 Windows 檔案伺服器的 Amazon FSx 設定為 L PageCenter RS X 的操作資料存放區

任務	描述	所需技能
建立 LRS PageCenter X 的檔案系統。	若要使用適用於 Windows 檔案伺服器的 Amazon FSx 做為異地同步備份環境中 LRS PageCenter X 的操作資料存放區，請遵循步驟 1：建立檔案系統中的指示。	雲端架構師

任務	描述	所需技能
將檔案共用對應至 LRS PageCenter X EC2 執行個體。	若要將在上一步中建立的檔案共用對應至 LRS PageCenter X EC2 執行個體，請遵循步驟 2：將檔案共用對應至執行 Windows Server 的 EC2 執行個體中的指示。	雲端架構師
將 LRS PageCenter X 控制目錄和主資料夾目錄對應至 Amazon FSx 網路共用磁碟機。	1. 按照 Amazon EC2 文件中的指示，Connect 到您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在資料夾總管中，選擇管理員，設定。 4. 在 [組態] 頁面上，選擇 [目錄]，然後選擇 [控制目錄]。 5. 在控制目錄中，輸入`\\FSx file share DNS name\share\cntl` 。 6. 在主資料夾目錄中，輸入`\\FSx file share DNS name\share\mstr` 。	雲端架構師

測試輸出管理工作流程

任務	描述	所需技能
從 OpenText Micro Focus BankDemo 應用程式啟動批次列印要求。	1. 在您的 OpenText 微焦點企業伺服器 EC2 執行個體中開啟 3270 終端機模擬器。 2. 透過執行命令 Connect 線至 BankDemo 應用程	測試工程師

任務	描述	所需技能
	式connect 127.0.0.1 :9278 。 3. 在 BankDemo 指令行介面上，對於「使用者識別碼」，輸入 B0001。在密碼中，輸入非空白金鑰。 4. 針對「請求列印對帳單」選項，在空白行中輸入 X。 5. 在 [依據傳送陳述式] 區段中，針對 [郵件] 輸入 Y，然後按 F10。	

任務	描述	所需技能
檢查 LRS PageCenter X 中的列印輸出。	1. 按照 Amazon EC2 文件中的指示，Connect 到您的 LRS PageCenter X EC2 執行個體。 2. 在視窗的「開始」功能表上，開啟 PCX 網頁介面。 3. 在功能窗格中，開啟 [測試] 資料夾，開啟 STD 資料夾，然後開啟具有工作執行日期的資料夾，例如 `08-03-2023` (MM-DD-YYYY)。 附註：這與內文中定義的資料夾結構相同建立規則，將輸出文件路由到 LRS PageCenter X 中的特定資料夾。 4. 開啟 `formtest-STD.txt` 檔案。 現在，您可以看到帳戶對帳單的打印輸出，其中包含「帳戶號碼」列。 、摘要、日期、金額及餘額。如需範例，請參閱此樣式的`batch_print_output` 附件。	測試工程師

相關資源

- LRS
- 進階功能簡報資料串流 (IBM 文件)

- 線路調節數據流（LCDS）（堆肥文檔）
- AWS 上的微焦點企業伺服器 (AWS 快速入門)
- 運用 Micro Focus 強化 AWS 上的企業大型主機工作負載 (部落格文章)
- 在 AWS 上將大型主機線上列印工作負載現代化 (AWS Prescriptive Guidance)
- 在 AWS 上將大型主機批次列印工作負載現代化 (AWS Prescriptive Guidance)

其他資訊

考量

在您的現代化過程中，您可以考慮針對大型主機批次和線上程序及其產生的輸出的各種組態。大型主機平台已由每位使用該平台的客戶和廠商自訂，這些平台符合直接影響列印的特定需求。例如，您目前的平台可能會將 IBM AFP 資料串流或施樂液晶顯示器整合到目前的工作流程中。此外，大型主機托架控制字元和通道命令字可能會影響列印頁面的外觀，並且可能需要特殊處理。作為現代化規劃程序的一部分，我們建議您評估並瞭解特定列印環境中的組態。

列印資料擷取

OpenText 微聚焦列印結束會傳遞必要的資訊，讓 LRS VPSX/MFI 有效處理捲軸檔案。該信息由相關控制塊中傳遞的字段組成，如下所示：

- 工作名稱
- 擁有者 (使用者 ID)
- 目的地
- 形式
- 檔案名稱
- 作家

LRS VPSX/MFI 支援下列大型主機批次機制，用於從微焦點企業伺服器擷取資料： OpenText

- 使用標準 z/OS JCL SYSOUT DD/輸出語句批次 COBOL 打印/線軸處理。
- 使用標準 z/OS JCL CA-佇列子系統 DD 陳述式的批次 COBOL 列印/捲軸處理。
- 使用 CBLTDLI 介面進行 IMS/COBOL 列印/捲軸處理。如需支援方法和程式設計範例的完整清單，請參閱產品授權隨附的 LRS 文件。

印表機-車隊健康檢查

LRS VPSX/MFI (LRS LoadX) 可以執行深入的運作狀態檢查，包括裝置管理和作業最佳化。裝置管理可偵測印表機裝置中的故障，並將列印要求路由至健康狀態良好的印表機。如需深入瞭解印表機叢集執行狀態檢查的詳細資訊，請參閱產品授權隨附的 LRS 文件。

列印驗證和授權

LRS/DIS 使 LRS 應用程序能夠通過使用 Microsoft 活動目錄或輕量級目錄訪問協議（LDAP）服務器來驗證用戶 ID 和密碼。除了基本的列印授權之外，LRS/DIS 也可以在下列使用案例中套用粒度層級的列印安全性控制：

- 管理誰可以瀏覽印表機工作。
- 管理其他使用者工作的瀏覽層級。
- 管理操作任務 — 例如，保留或釋放、清除、修改、複製和重定路由等命令層級安全性。安全性可以透過使用者識別碼或群組來設定，類似於使用中目錄安全性群組或 LDAP 群組。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用微焦點企業伺服器和 LRS VPSX/MFI，在 AWS 上現代化大型主機批次列印工作負載

創建者：舒本羅伊（AWS），亞伯拉罕·朗登（微焦點），蓋伊·塔克（列維，雷和壽普公司）和凱文容（AWS）

環境：PoC 或試點	資料來源：IBM 大型主機	目標：AWS
R 類型：重新平台	工作負載：IBM	技術：大型主機；現代化
AWS 服務：AWS 管理 Microsoft AD; Amazon EC2; Amazon S3; Amazon EBS		

Summary

此模式說明如何使用 Micro Focus 企業伺服器做為現代化大型主機應用程式的執行時間，將您的業務關鍵大型主機批次列印工作負載現代化，並將 LRS VPSX/MFI (Micro Focus 界面) 作為列印伺服器，在亞馬遜網路服務 (AWS) 雲端上將業務關鍵大型主機批次列印工作負載現代化。該模式以重新平台大型主機現代化方法為基礎。在這種方法中，您可以將大型主機批次任務遷移到 Amazon Elastic Compute Cloud (Amazon EC2)，然後將大型主機資料庫 (例如 IBM DB2 for z/OS) 遷移到 Amazon Relational Database Service 服務 (Amazon RDS)。現代化列印工作流程的身份驗證和授權是由適用於 Microsoft 活動目錄的 AWS Directory Service 執行，也稱為 AWS 託管 Microsoft AD。LRS 目錄資訊伺服器 (LRS/DIS) 與 AWS 受管 Microsoft AD 整合。透過將批次列印工作負載現代化，您可以降低 IT 基礎設施成本、減輕維護舊有系統的技術債務、移除資料孤島、使用 DevOps 模型提高靈活性和效率，以及利用 AWS 雲端中的隨需資源和自動化功能。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 大型主機列印或輸出管理工作負載
- 有關如何重建和交付在 Micro Focus 企業伺服器上執行的大型主機應用程式的基本知識 (如需詳細資訊，請參閱 Micro Focus 文件中的企業伺服器資料表)。
- LRS 雲端列印解決方案和概念的基本知識 (如需詳細資訊，請參閱 LRS 文件中的輸出現代化)。

- Micro Focus 企業伺服器軟體和使用授權 (如需詳細資訊，請洽 Micro Focus 銷售人員。)
- LRS VPSX/MFI、LRS /隊列和LRS/DIS 軟體和授權 (如需詳細資訊，請聯絡 LRS 銷售人員。)

備註：如需有關大型主機批次列印工作負載之組態考量的詳細資訊，請參閱此模式的其他資訊一節中的考量事項。

產品版本

- 微焦點企業伺服器 6.0 (產品更新 7)
- LRS VPSX/MFI V1R 3 或更高版本

架構

源, 技術, 堆棧

- 作業系統 —
- 程式語言 — 通用面向業務的語言 (COBOL)、Job 控制語言 (JCL) 和客戶資訊控制系統 (CICS)
- 資料庫 — IBM DB2 適用於 z/OS 與虛擬儲存存取方法 (VSAM)
- 安全性 — 資源存取控制設施 (RACF)、z/OS 的 CA 最高機密，以及存取控制設施 2 (ACF2)
- 列印與輸出管理 — IBM 大型主機 z/OS 列印產品 (IBM Tivoli 輸出管理程式，適用於 z/OS、LRS 和 CA 檢視)

目標技術堆疊

- 操作系統 — 在 Amazon EC2 上運行的 Microsoft 視窗服務器
- 運算 — Amazon EC2
- 編程語言-COBOL，JCL 和 CICS
- 數據庫 — Amazon RDS
- 安全性 — AWS 管理 Microsoft AD
- 列印和輸出管理 — AWS 上的 LRS 列印解決方案
- 大型主機執行階段環境 — 微焦點企業伺服器

來源架構

下圖顯示大型主機批次列印工作負載的典型目前狀態架構：

該圖顯示以下工作流程：

1. 使用者在以 COBOL 撰寫之 IBM CICS 應用程式建置的參與系統 (SoE) 上執行商業交易。
2. SoE 會叫用大型主機服務，該服務會將商業交易資料記錄在 system-of-records (sR) 資料庫中，例如 IBM DB2 for z/OS。
3. SoR 保留了來自 SoE 的業務數據。
4. 批次工作排程器會啟動批次工作以產生列印輸出。
5. 批次工作會從資料庫擷取資料、根據業務需求格式化資料，然後產生業務輸出，例如帳單、ID 卡或貸款對帳單。最後，批次工作會根據業務需求，將輸出路由至列印輸出管理，以進行處理和輸出傳送。
6. 列印輸出管理會從批次工作接收列印輸出，然後將輸出傳送至指定的目的地，例如電子郵件、使用安全 FTP 的檔案共用、使用 LRS 列印解決方案的實體印表機 (如此模式所示) 或 IBM Tivoli。

目標架構

下圖顯示部署在 AWS 雲端的大型主機批次列印工作負載的架構：

該圖顯示以下工作流程：

1. 批次工作排程器會啟動批次工作以建立列印輸出，例如帳單對帳單、ID 卡或貸款對帳單。
2. 大型主機批次任務 (重新組成 Amazon EC2) 會使用 Micro Focus 企業伺服器執行階段從應用程式資料庫擷取資料、將商業邏輯套用至資料、格式化資料，然後使用 Micro Focus 列印結束 (Micro Focus 文件) 將資料傳送至列印目的地。
3. 應用程式資料庫 (在 Amazon RDS 上執行的 SoR) 會保留列印輸出的資料。
4. LRS VPSX/MFI 列印解決方案部署在 Amazon EC2 上，其操作資料儲存在 Amazon Elastic Block Store (Amazon EBS) 中。LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /queue 傳輸代理程式，透過微焦點 JES 列印結束 API 收集列印資料，並將資料傳送至指定的印表機目的地。

附註：目標解決方案通常不需要變更應用程式以適應大型主機格式化語言，例如 IBM 進階功能簡報 (AFP) 或施樂生產線狀況資料串流 (LCDS)。如需有關在 AWS 上使用 Micro Focus 進行大型主機應用程式遷移和現代化的詳細資訊，請參閱 AWS 文件中的使用 Micro Focus 為 AWS 上的企業大型主機工作負載提供支援。

AWS 基礎設施架構

下圖顯示適用於大型主機批次列印工作負載的高可用性和安全 AWS 基礎設施架構：

該圖顯示以下工作流程：

1. 批次排程器會啟動批次程序，並跨多個可用區域部署在 Amazon EC2 上，以實現高可用性 (HA)。注意：此模式不涵蓋批次排程器的實作。如需有關實作的詳細資訊，請參閱排程器的軟體廠商說明文件。
2. 大型主機批次工作 (以 JCL 或 COBOL 等程式設計語言撰寫) 使用核心業務邏輯來處理並產生列印輸出，例如帳單、ID 卡和貸款陳述式。該任務部署在 HA 的兩個可用區域的 Amazon EC2 上，並使用微焦點列印結束將列印輸出路由到 LRS VPSX/MFI 進行最終使用者列印。
3. LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /隊列傳輸代理程式從微焦點JES 列印結束程式設計介面收集或擷取列印資料。「列印結束」會傳遞必要的資訊，以啟用 LRS VPSX/MFI，以有效地處理多工緩衝處理檔案並動態建置 LRS /列命令。接著會使用 Micro Focus 的標準內建函數來執行這些指令。備註：如需有關從 Micro Focus 列印結束傳送至 LRS /queue 以及 LRS VPSX/MFI 支援大型主機批次機制的列印資料的詳細資訊，請參閱此模式的其他資訊一節中的列印資料擷取。
4. Network Load Balancer 提供 DNS 名稱，以整合微焦點企業伺服器與 LRS VPSX/MFI。注意：LRS VPSX/MFI 支援第 4 層負載平衡器。Network Load Balancer 也會對 LRS VPSX/MFI 執行基本健康狀態檢查，並將流量路由到狀態良好的已註冊目標。
5. LRS VPSX/MFI 列印伺服器部署在 HA 的兩個可用區域的 Amazon EC2 上，並使用 Amazon EBS 做為操作資料存放區。LRS VPSX/MFI 支援主動-主動和主動-被動服務模式。此架構使用主動-被動配對中的多個 AZ 作為主動和熱待命。Network Load Balancer 會對 LRS VPSX/MFI EC2 執行個體執行健康狀態檢查，如果作用中執行個體處於運作狀態不良，則網路負載平衡器會將流量路由至其他 AZ 中的熱待命執行個體。列印要求會在每個 EC2 執行個體的本機 LRS Job 佇列中保留。在復原的情況下，必須重新啟動失敗的執行個體，LRS 服務才能繼續處理列印要求。注意：LRS VPSX/MFI 也可以在印表機機群層級執行健康狀態檢查。如需詳細資訊，請參閱此模式的其他資訊一節中的印表機群健康狀態檢查。
6. AWS 受管 Microsoft AD 與 LRS/DIS 整合，以執行列印工作流程身份驗證和授權。如需詳細資訊，請參閱此模式的其他資訊一節中的列印驗證和授權。
7. LRS VPSX/MFI 使用 Amazon EBS 進行區塊儲存。您可以將 Amazon EBS 資料從作用中 EC2 執行個體備份到 Amazon S3 做為 point-in-time 快照，並將其還原到熱待命 EBS 磁碟區。若要自動建立、保留和刪除 Amazon EBS 磁碟區快照，您可以使用 Amazon Data Lifecycle Manager 設定自動快照的頻率，並根據您的 RTO/R PO 要求進行還原。

工具

AWS 服務

- Amazon EBS — 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，以便與 EC2 執行個體搭配使用。EBS 磁碟區的行為與未格式化的原始區塊型儲存設備相似。您可以將這些磁碟區做為裝置，掛載在您的執行個體上。
- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。它為關聯式資料庫提供符合成本效益且可調整大小的容量，並管理常見的資料庫管理工作。
- AWS 受管 Microsoft AD — 適用於 Microsoft 活動目錄的 AWS Directory Service，也稱為 AWS 受管 Microsoft 活動目錄，可讓您的目錄感知工作負載和 AWS 資源使用 AWS 中的受管活動目錄。

其他工具

- LRS VPSX/MFI（微對焦接口）— VPSX/MFI 由 LRS 和微聚焦共同開發，可捕獲微焦點企業服務器 JES 線軸的輸出，並可靠地將其提供到指定的打印目的地。
- LRS 目錄資訊伺服器 (LRS/DIS) — LRS/DIS 用於列印工作流程期間的驗證和授權。
- LRS /隊列 — LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /隊列傳輸代理程式，透過微焦點 JES 列印結束程式設計介面收集或擷取列印資料。
- Micro Focus 企業伺服器 — Micro Focus 企業伺服器是大型主機應用程式的應用程式部署環境。它為使用任何版本的 Micro Focus 企業開發人員遷移或建立的大型主機應用程式提供執行環境。

史诗

在 Amazon EC2 上設定微焦點企業伺服器，並部署大型主機批次應用程式

任務	描述	所需技能
設定 Micro Focus 企業伺服器並部署示範應用程式。	在 Amazon EC2 上設定微焦點企業伺服器，然後依照 AWS 上的微焦點企業伺服器快速入門部署指南中的指 BankDemo	雲端架構師

任務	描述	所需技能
	示，在 Amazon EC2 上部署 Micro Focus 示範應用程式。 此 BankDemo 應用程式是建立並啟動列印輸出的大型主機批次應用程式。	

在 Amazon EC2 上設置 LRS 列印伺服器

任務	描述	所需技能
取得用於列印的 LRS 產品授權。	若要取得 LRS VPSX/MFI、LRS /列和LRS/DIS 的 LRS 產品授權，請聯絡 LRS 輸出管理團隊。您必須提供將安裝 LRS 產品之 EC2 執行個體的主機名稱。	建立領導
建立一個 Amazon EC2 視窗執行個體以安裝 LRS VPSX/MFI。	依照步驟 1 中的指示啟動 Amazon EC2 Windows 執行個體：啟動 Amazon EC2 文件中的執行個體。您的執行個體必須符合下列 LRS VPSX/MFI 的硬體和軟體需求： • 中央處理器 — 雙核心 • 公羊-16 GB • 磁碟機 — 500 GB • 最低 EC2 執行個體 — 大型 • 作業系統 — 視窗/linux • 軟體 — 網際網路資訊服務 (IIS) 或 Apache	雲端架構師

任務	描述	所需技能
	注意：先前的硬體和軟體需求適用於小型印表機群 (約 500—1000)。若要取得完整要求，請諮詢您的 LRS 和 AWS 聯絡人。 當您建立 Windows 執行個體時，請執行下列動作： 1. 確認 EC2 主機名稱與用於 LRS 產品授權的主機名稱相同。 2. 完成以下操作，在 Amazon EC2 中啟用 CGI： a. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。 b. 在 Windows「開始」功能表中，找到並開啟「伺服器管理員」。 c. 在伺服器管理員中，選擇儀表板、快速入門、新增角色和功能。然後，選擇 [伺服器角色]。 d. 在 [伺服器角色] 中，選擇 [WebServer (IIS)]，然後選擇 [應用程式開發]。 e. 在 [應用程式開發] 中，選取 [CGI] 核取方塊。 f. 請依照 Windows 伺服器管理員新增角色和功能精靈上的指示來安裝 CGI。	

任務	描述	所需技能
	g. 在 EC2 執行個體的 Windows 防火牆中開啟連接埠 5500，以進行 LRS /列通訊。	
在 EC2 執行個體上安裝 LRS VPSX/MFI。	1. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟產品下載頁面的連結。注意：LRS 產品通過電子文件傳輸（EFT）進行分發。 3. 下載 LRS VPSX/MFI 並解壓縮該文件（默認文件夾： ）。`c:\LRS` 4. 從解壓縮的資料夾啟動 LRS 產品安裝程式，以安裝 LRS VPSX/MFI。 5. 在「選取功能」功能表中，選取「VPSX® 伺服器 (V1R3.022)」，然後選擇「下一步」以開始安裝程序。安裝完成後，您將收到成功訊息。	雲端架構師

任務	描述	所需技能
安裝 LRS /列。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS /queue，然後解壓縮檔案。 3. 移至您下載檔案的位置，然後啟動 LRS 產品安裝程式以安裝 LRS /列。	雲端架構師
安裝 LRS/DIS。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS/DIS，然後解壓縮檔案。 3. 移至您下載檔案的位置，然後啟動 LRS 產品安裝程式。 4. 在「LRS 產品安裝程式」中，展開「LRS 雜項工具」，選取「LRS DIS」，然後選擇「下一步」。 5. 遵循 LRS 產品安裝程式中的其餘指示，以完成安裝程序。	雲端架構師

任務	描述	所需技能
建立目標群組並將 LRS VPSX/MFI EC2 註冊為目標。	依照 Elastic Load Balancing 說明文件中的為您的 Network Load Balancer 建立目標群組中的指示，建立目標群組。 建立目標群組時，請執行下列動作： 1. 在 [指定群組詳細資訊] 頁面上，選擇執行處理做為 [選擇目標類型]。 2. 針對通訊協定，選擇 TCP。 3. 在「連接埠」中，選擇 5500。 4. 在 [註冊目標] 頁面的 [可用執行個體] 區段中，選取 LRS VPSX/MFI EC2 執行個體。	雲端架構師

任務	描述	所需技能
建立 Network Load Balancer。	依照 Elastic Load Balancing 說明文件中建立 Network Load Balancer 的指示進行。您的 Network Load Balancer 會將流量從微焦點企業伺服器路由到 LRS VPSX/MFI EC2。 建立 Network Load Balancer 時，請在「監聽器和路由」頁面執行下列動作： 1. 針對 Protocol (通訊協定)，選擇 TCP。 2. 在「連接埠」中，選擇 5500。 3. 對於「預設」動作，請為您先前建立的目標群組選擇「轉寄至」。	雲端架構師

將微焦企業伺服器與 LRS VPSX/MFI 和 LRS /隊列整合

任務	描述	所需技能
設定微焦點企業伺服器以進行 LRS /列整合。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 2. 在 Windows「開始」功能表中，開啟「微焦點企業伺服器管理」使用者介面。 3. 在選單列中，選擇「原生」。	雲端架構師

任務	描述	所需技能
	4. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO]。 5. 從左側導覽窗格的 [一般] 中，向下捲動至 [其他] 區段，以設定環境變數 (LRSQ_ADDRESS、LRSQ_PORT、LRSQ_命令) 以指向 LRSQ。 6. 對於 LRSQ_ADDRESS，請輸入您先前建立之 Network Load Balancer 的 IP 位址或 DNS 名稱。 7. 針對連接埠，請輸入 VPSX L RSQ 接聽程式連接埠 (5500)。 8. 對於 LRSQ_ 指令，請輸入 LRSQ 可執行檔的路徑位置。 注意：LRS 目前支援的 DNS 名稱最大字元限制為 50，但 future 可能會有所變更。如果您的 DNS 名稱大於 50，則可以使用 Network Load Balancer 的 IP 位址作為替代方案。	

任務	描述	所需技能
為 LRS VPSX/MFI 整合設定微型焦點企業伺服器。	1. 將VPSX_MFI_R2 資料夾從 LRS VPSX/MFI 安裝程式複製到微焦點企業伺服器位置 (位於)。C\BANKDEMO\print 2. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 3. 在 Windows「開始」功能表中，開啟「微焦點企業伺服器管理」使用者介面。 4. 在選單列中，選擇「原生」。 5. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO]。 6. 在「銀行示範」下，選擇 JES。 7. 在「JES 程式路徑」下，從該C\BANKDEMO\print 位置加入DLL(VPSX_MFI_R2) 路徑。	雲端架構師

在微焦點企業服務器和 LRS VPSX/MFI 中設置打印機和打印用戶

任務	描述	所需技能
將 Micro Focus 列印結束模組與 Micro Focus 企業伺服器批次印表機伺服器執行程序產生關聯。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您	雲端架構師

任務	描述	所需技能
	的 Micro Focus 企業伺服器 EC2 執行個體。 2. 在 Windows「開始」功能表中，開啟「微焦點企業伺服器管理」使用者介面。 3. 在選單列中，選擇「原生」。 4. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO]。 5. 在「銀行示範」下，選擇 JES，然後向下捲動至「印表機」。 6. 在「印表機」中，將「微焦點列印結束」模組 (LRSPRTE6 用於 Batch) 與 Micro Focus 企業伺服器批次印表機伺服器執行程序 (SEP) 產生關聯。這樣可以將列印輸出路由傳送至 LRS VPSX/MFI。 7. 登入企業伺服器管理 UI。 如需有關組態的詳細資訊，請參閱 Micro Focus 文件中的使用結束。	

任務	描述	所需技能
在 LRS VPSX/MFI 中添加打印機。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從視窗的「開始」功能表開啟 VPSX 網頁介面。 3. 在導覽窗格中，選擇 [印表機]。 4. 選擇 [新增]，然後選擇 [新增印表機]。 5. 在 [印表機規劃] 頁面上，輸入 L oc al 做為 [印表機名稱]。 6. 針對 VPSX 識別碼，請輸入 VPS1。 7. 對於 CommType，請選取「TCPIP /LR SQ」。 8. 對於主機 /IP 位址，輸入要新增之實體印表機的 IP 位址。 9. 在「裝置」中，輸入裝置的名稱。 10選擇任一視窗驅動程序或鏈接 /Mac 驅動程序。 11選擇新增。	雲端架構師

任務	描述	所需技能
在 LRS VPSX/MFI 中創建一個打印用戶。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從視窗的「開始」功能表開啟 VPSX 網頁介面。 3. 在功能窗格中，選擇 [安全性]，然後選擇 [使用者]。 4. 在 [使用者名稱] 欄中，選擇 [管理員]，然後選擇 [複製]。 5. 在「使用者設定檔維護」視窗中，對於「使用者名稱」，輸入使用者名稱 (例如，PrintUser)。 6. 在「說明」中，輸入簡短描述 (例如，測試列印的使用者)。 7. 選擇更新。這將創建一個打印用戶（例如，PrintUser）。 8. 在導覽窗格的 [使用者] 下，選擇您建立的新使用者。 9. 從「命令」功能表中選擇「安全性」。 10. 在 [安全性規則] 頁面上，選擇所有適用的印表機安全性和工作安全性選項，然後選擇 [儲存]。 11. 若要將新的列印使用者新增至 [系統管理員] 群組，請	雲端架構師

任務	描述	所需技能
	移至功能窗格，選擇 [安全性]，然後選擇 [設定]。 12.在「安全性」組態視窗中，將新的列印使用者新增至「管理員」欄。	

設定列印驗證和授權

任務	描述	所需技能
使用使用者和群組建立 AWS 受管 Microsoft AD 網域。	1. 按照 AWS 目錄服務文件中的建立您的 AWS 受管 Microsoft AD 目錄中的指示，在 AWS 上建立活動目錄。 2. 依照 AWS 目錄服務文件中的步驟 3 中的指示，部署 EC2 執行個體 (使用中目錄管理員) 並安裝 Active Directory 工具來管理您的 AWS 受管 Microsoft AD：部署 EC2 執行個體以管理您的 AWS 受管 Microsoft AD。 3. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。注意：連線至 EC2 執行個體時，請在 Windows 安全性視窗中輸入管理員登入資料 (針對您在步驟 1 中建立的目錄)。	雲端架構師

任務	描述	所需技能
	4. 在 Windows「開始」功能表的「Windows 系統管理工具」下，選擇「作用中目錄使用者和電腦」。 5. 按照 AWS 目錄服務文件中的建立使用者中的步驟，在 Active Directory 網域中建立列印使用者。	
將 LRS VPSX/MFI EC2 加入 AWS 受管 Microsoft AD 網域。	自動將 LRS VPSX/MFI EC2 加入您的 AWS 受管 Microsoft AD 網域 (AWS 知識中心文件) 或手動 (AWS Directory Service 文件)。	雲端架構師

任務	描述	所需技能
設定並整合 LRS/DIS 與 AWS 受管 Microsoft AD。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗「開始」功能表中，開啟 VPSX 網頁介面。 3. 在瀏覽窗格中，選擇 [安全性]，然後選擇 [設定]。 4. 在「安全組態」頁面的「安全參數」段落中，選取「內部」做為「安全類型」。 5. 在「安全性參數」區段中輸入其餘選項的偏好設定。 6. 從 Microsoft Windows 的「開始」功能表開啟「LRS 輸出管理」資料夾，選擇「伺服器啟動」，然後選擇「伺服器停止」。 7. 使用您的活動目錄用戶名和密碼登錄到 LRS VPSX/MFI。	雲端架構師

測試列印工作流程

任務	描述	所需技能
從 Micro Focus BankDemo 應用程式啟動批次列印要求。	1. 在您的微焦點企業伺服器 EC2 執行個體中開啟 3270 終端機模擬器。 2. 執行下列命令以 Connect 至 BankDemo 應用程	測試工程師

任務	描述	所需技能
	式：connect 127.0.0.1 :9278 3. 在 BankDemo 指令行介面上，對於「使用者識別碼」，輸入 B0001。在「密碼」中，輸入非空白金鑰。 4. 針對「請求列印對帳單」選項，在空白行中輸入 X。 5. 在 [依據傳送陳述式] 區段中，針對 [郵件] 輸入 Y，然後按 F10。	
在 LRS VPSX/MFI 中檢查打印輸出。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗開始功能表中，開啟 VPSX 網頁介面。 3. 在功能窗格中，選擇 [印表機]，然後選擇 [輸出佇列]。 4. 在 [多工緩衝處理 ID] 欄中，選擇印表機佇列中要求的多工緩衝處理 ID。 5. 在 [動作] 索引標籤的 [命令] 欄中，選擇 [瀏覽]。 現在，您可以看到帳戶對帳單的打印輸出，其中包含「帳戶號碼」列。 、摘要、日期、金額及餘額。有關示例，請參閱此模式的批次列印輸出附件。	測試工程師

相關資源

- LRS 輸出現代化 (LRS 文件)
- ANSI 和機器托架控制 (IBM 說明文件)
- 頻道命令詞 (IBM 文件)
- 運用 Micro Focus 強化 AWS 上的企業大型主機工作負載 (AWS 合作夥伴網路部落格)
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建立微焦點企業伺服器 PAC (AWS Prescriptive Guidance 文件)
- 進階功能簡報 (AFP) 資料串流 (IBM 說明文件)
- 線路調節數據流（LCDS）（堆肥文檔）
- AWS 上的微焦點企業伺服器 (AWS 快速入門)

其他資訊

考量

在您的現代化過程中，您可以考慮為大型主機批次處理程序及其產生的輸出提供各式各樣的組態。大型主機平台已由每位使用該平台的客戶和廠商自訂，這些平台符合直接影響列印的特定需求。例如，您目前的平台可能會將 IBM 進階功能簡報 (AFP) 或施樂生產線狀況資料串流 (LCDS) 納入目前的工作流程。此外，大型主機托架控制字元和 channel 指令字可能會影響列印頁面的外觀，並且可能需要特殊處理。作為現代化規劃程序的一部分，我們建議您評估並瞭解特定列印環境中的組態。

列印資料擷取

微聚焦列印結束會傳遞必要的資訊，以使 LRS VPSX/MFI 能夠有效處理多工緩衝處理檔案。該信息由相關控制塊中傳遞的字段組成，例如：

- 工作名稱
- 擁有者 (使用者 ID)
- 目的地
- 形式
- 檔案名稱
- 作家

LRS VPSX/MFI 支援下列大型主機批次處理機制，用於從微焦點企業伺服器擷取資料。

- 使用標準z/OS JCL SYSOUT DD/輸出語句批次 COBOL 列印/線軸處理
- 使用標準z/OS JCL CA-線軸子SYS DD 陳述式的批次 COBOL 列印/捲軸處理
- 使用CBLTDLI 介面的 IMS/COBOL 列印/線路處理 (如需支援方法和程式設計範例的完整清單，請參閱產品授權隨附的 LRS 文件)。

印表機機群健康檢查

LRS VPSX/MFI (LRS LoadX) 可以執行深入的運作狀態檢查，包括裝置管理和作業最佳化。裝置管理可偵測印表機裝置中的故障，並將列印要求路由至健康狀態良好的印表機。如需深入瞭解印表機叢集執行狀態檢查的詳細資訊，請參閱產品授權隨附的 LRS 文件。

列印驗證和授權

LRS/DIS 使 LRS 應用程序能夠通過使用 Microsoft 活動目錄或 LDAP 服務器來驗證用戶 ID 和密碼。除了基本的列印授權之外，LRS/DIS 也可以在下列使用案例中套用粒度層級的列印安全性控制：

- 管理誰可以瀏覽印表機工作。
- 管理其他使用者工作的瀏覽層級。
- 管理操作任務。例如，指令層級安全性，例如保留/釋放、清除、修改、複製和重定路由。安全性可以透過使用者識別碼或群組 (類似於 AD 群組或 LDAP 群組) 來設定。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用微焦企業伺服器和 LRS VPSX/MFI，在 AWS 上現代化大型主機線上列印工作負載

創建者：舒本羅伊（AWS），亞伯拉罕·朗登（微焦點），蓋伊·塔克（列維，雷和壽普公司）和凱文容（AWS）

環境：PoC 或試點	資料來源：大型機	目標：AWS
R 類型：重新平台	工作負載：IBM	技術：大型主機；移轉；現代化
AWS 服務：AWS 管理 Microsoft AD; Amazon EC2; Amazon RDS; Amazon EBS		

Summary

此模式說明如何使用 Micro Focus 企業伺服器做為現代化大型主機應用程式的執行時間，將您的業務關鍵大型主機線上列印工作負載現代化，並將 LRS VPSX/MFI (Micro Focus 界面) 作為列印伺服器，在亞馬遜網路服務 (AWS) 雲端上現代化業務關鍵大型主機線上列印工作負載。該模式以重新平台大型主機現代化方法為基礎。在這種方法中，您可以將大型主機線上應用程式遷移到 Amazon Elastic Compute Cloud (Amazon EC2)，然後將大型主機資料庫 (例如 IBM DB2 for z/OS) 遷移到 Amazon Relational Database Service 服務 (Amazon RDS)。現代化列印工作流程的身份驗證和授權是由適用於 Microsoft 活動目錄的 AWS Directory Service 執行，也稱為 AWS 託管 Microsoft AD。LRS 目錄資訊伺服器 (LRS/DIS) 與 AWS 受管 Microsoft AD 整合，以進行列印工作流程身份驗證和授權。透過將線上列印工作負載現代化，您可以降低 IT 基礎設施成本、減輕維護舊有系統的技術債務、移除資料孤島、使用 DevOps 模型提高靈活性和效率，以及利用 AWS 雲端中的隨需資源和自動化功能。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 大型主機線上列印或輸出管理工作負載
- 有關如何重建和交付在 Micro Focus 企業伺服器上執行的大型主機應用程式的基本知識 (如需詳細資訊，請參閱 Micro Focus 文件中的企業伺服器資料表)。

- LRS 雲端列印解決方案和概念的基本知識 (如需詳細資訊，請參閱 LRS 文件中的輸出現代化)。
- Micro Focus 企業伺服器軟體與授權 (如需詳細資訊，請洽 Micro Focus 銷售人員。)
- LRS VPSX/MFI、LRS /隊列和LRS/DIS 軟體和授權 (如需詳細資訊，請聯絡 LRS 銷售人員。)

備註：如需有關大型主機線上列印工作負載之組態考量的詳細資訊，請參閱此模式的其他資訊一節中的考量事項。

產品版本

- 微焦點企業伺服器 8.0 或更新版本
- LRS VPSX/MFI V1R 3 或更高版本

架構

源, 技術, 堆棧

- 作業系統 —
- 程式語言 — 通用面向業務的語言 (COBOL) 和客戶資訊控制系統 (CICS)
- 資料庫 — IBM DB2 適用於 z/OS 的 IBM 資訊管理系統 (IMS) 與虛擬儲存存取方法 (VSAM)
- 安全性 — 資源存取控制設施 (RACF)、z/OS 的 CA 最高機密，以及存取控制設施 2 (ACF2)
- 列印與輸出管理 — IBM 大型主機 z/OS 列印產品 (適用於 z/OS、LRS 和 CA 檢視的 IBM 資訊印刷伺服器)

目標技術堆疊

- 操作系統 — 在 Amazon EC2 上運行的 Microsoft 視窗服務器
- 運算 — Amazon EC2
- 編程語言-COBOL 和 CICS
- 數據庫 — Amazon RDS
- 安全性 — AWS 管理的 Microsoft AD
- 列印和輸出管理 — AWS 上的 LRS 列印解決方案
- 大型主機執行階段環境 — 微焦點企業伺服器

來源架構

下圖顯示大型主機線上列印工作負載的典型目前狀態架構。

該圖顯示以下工作流程：

1. 使用者在以 COBOL 撰寫之 IBM CICS 應用程式建置的參與系統 (SoE) 上執行商業交易。
2. SoE 會叫用大型主機服務，該服務會將商業交易資料記錄在 system-of-records (sR) 資料庫中，例如 IBM DB2 for z/OS。
3. SoR 保留了來自 SoE 的業務數據。
4. 使用者啟動從 CICS SoE 產生列印輸出的請求，該請求會起始列印異動應用模組來處理列印請求。
5. 列印交易應用程式 (例如 CICS 和 COBOL 程式) 會從資料庫擷取資料、根據業務需求格式化資料，並產生業務輸出 (列印資料)，例如帳單、ID 卡或貸款對帳單。然後，應用程式會使用虛擬電信存取方法 (VTAM) 傳送列印要求。z/OS 列印伺服器 (例如 IBM Inforint 伺服器) 會使用 NetSpool 或類似的 VTAM 元件攔截列印要求，然後使用 JES 輸出參數在 JES 多工緩衝處理上建立列印輸出資料集。JES 輸出參數會指定列印伺服器用來將輸出傳送至特定網路印表機的路由資訊。VTAM 一詞是指 z/OS 通訊伺服器和 z/OS 的系統網路架構 (SNA) 服務元素。
6. 列印輸出傳輸元件會將輸出列印資料集從 JES 多工緩衝處理傳輸到遠端印表機或列印伺服器，例如 LRS (如此模式所示)、IBM Inforint 伺服器或電子郵件目的地。

目標架構

下圖顯示部署在 AWS 雲端的大型主機線上列印工作負載的架構：

該圖顯示以下工作流程：

1. 使用者從線上 (CICS) 使用者介面啟動列印要求，以建立列印輸出，例如帳單、身份證或貸款對帳單。
2. 大型主機線上應用程式 (轉換為 Amazon EC2) 會使用 Micro Focus 企業伺服器執行階段從應用程式資料庫擷取資料、將商業邏輯套用至資料、格式化資料，然後使用 Micro Focus CICS 列印結束 (DFHUPRNT) 將資料傳送至列印目的地。
3. 應用程式資料庫 (在 Amazon RDS 上執行的 SoR) 會保留列印輸出的資料。
4. LRS VPSX/MFI 列印解決方案部署在 Amazon EC2 上，其操作資料儲存在 Amazon Elastic Block Store (Amazon EBS) 中。LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /queue 傳輸代理程式，透

過微焦點 CICS 列印結束 API (DFHUPRNT) 收集列印資料，並將資料傳送至指定的印表機目的地。在現代化的 CICS 應用程序中使用的原始 TERMID（術語）被用作 VPSX/MFI 隊列名稱。

附註：目標解決方案通常不需要變更應用程式以適應大型主機格式化語言，例如 IBM 進階功能簡報 (AFP) 或施樂生產線狀況資料串流 (LCDS)。如需有關在 AWS 上使用 Micro Focus 進行大型主機應用程式遷移和現代化的詳細資訊，請參閱 AWS 文件中的使用 Micro Focus 為 AWS 上的企業大型主機工作負載提供支援。

AWS 基礎設施架構

下圖顯示適用於大型主機線上列印工作負載的高可用性和安全 AWS 基礎設施架構：

該圖顯示以下工作流程：

1. 大型主機線上應用程式 (以 CICS 或 COBOL 等程式設計語言撰寫) 使用核心業務邏輯來處理和產生列印輸出，例如帳單、身分證和貸款對帳單。線上應用程式部署在跨兩個可用區域 (AZ) 的 Amazon EC2 上以實現高可用性 (HA)，並使用微焦 CICS 列印結束將列印輸出路由到 LRS VPSX/MFI 以進行最終使用者列印。
2. LRS VPSX/MFI 使用以 TCP/IP 為基礎的 LRS /隊列傳輸代理程式，從微焦點線上列印結束程式設計介面收集或擷取列印資料。「線上列印結束」會傳遞必要的資訊，以啟用 LRS VPSX/MFI 有效處理列印檔案並動態建置 LRS /queue 指令。

   注意：如需有關各種列印 CICS 應用程式設計方法的詳細資訊，以及 Micro Focus 企業伺服器和 LRS VPSX/MFI 中如何支援這些程式的資訊，請參閱此模式的其他資訊一節中的列印資料擷取。
3. Network Load Balancer 提供 DNS 名稱，以整合微焦點企業伺服器與 LRS VPSX/MFI。注意：LRS VPSX/MFI 支援第 4 層負載平衡器。Network Load Balancer 也會對 LRS VPSX/MFI 執行基本健康狀態檢查，並將流量路由到狀態良好的已註冊目標。
4. LRS VPSX/MFI 列印伺服器部署在 HA 的兩個可用區域的 Amazon EC2 上，並使用 Amazon EBS 做為操作資料存放區。LRS VPSX/MFI 支援主動-主動和主動-被動服務模式。此架構使用主動-被動配對中的多個可用區域作為主動和熱待命。Network Load Balancer 會對 LRS VPSX/MFI EC2 執行個體執行健康狀態檢查，如果作用中執行個體處於運作狀態不良，則網路負載平衡器會將流量路由到另一個可用區域中的熱待命執行個體。列印要求會在每個 EC2 執行個體的本機 LRS Job 佇列中保留。在復原的情況下，必須重新啟動失敗的執行個體，LRS 服務才能繼續處理列印要求。

   注意：LRS VPSX/MFI 也可以在印表機機群層級執行健康狀態檢查。如需詳細資訊，請參閱此模式的其他資訊一節中的印表機群健康狀態檢查。

5. AWS 受管 Microsoft AD 與 LRS/DIS 整合，以執行列印工作流程身份驗證和授權。如需詳細資訊，請參閱此模式的其他資訊一節中的列印驗證和授權。
6. LRS VPSX/MFI 使用 Amazon EBS 進行區塊儲存。您可以將 Amazon EBS 資料從作用中 EC2 執行個體備份到 Amazon S3 做為 point-in-time 快照，並將其還原到熱待命 EBS 磁碟區。若要自動建立、保留和刪除 Amazon EBS 磁碟區快照，您可以使用 Amazon Data Lifecycle Manager 設定自動快照的頻率，並根據您的 RTO/R PO 要求進行還原。

工具

AWS 服務

- Amazon Elastic Block Store (Amazon EBS) 提供區塊層級儲存體磁碟區，可搭配使用 Amazon EC2 執行個體。EBS 磁碟區的行為與未格式化的原始區塊型儲存設備相似。您可以將這些磁碟區做為裝置，掛載在您的執行個體上。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- 適用於 Microsoft 活動目錄 (AD) 的 AWS Directory Service，也稱為 AWS 受管 Microsoft 活動目錄，可讓您的目錄感知工作負載和 AWS 資源在 AWS 中使用受管的活動目錄。

其他工具

- 由 LRS 和微焦點共同開發的 LRS VPSX/MFI（微型對焦接口）可捕獲微焦點企業服務器 JES 線軸的輸出，並可靠地將其輸送到指定的打印目的地。
- LRS 目錄資訊伺服器 (LRS/DIS) 用於列印工作流程期間的驗證和授權。
- LRS /queue 是一種以 TCP/IP 為基礎的 LRS /隊列傳輸代理程式，由 LRS VPSX/MFI 使用，透過微焦點線上列印結束程式設計介面收集或擷取列印資料。
- Micro Focus 企業伺服器是適用於大型主機應用程式的應用程式部署環境。它為使用任何版本的 Micro Focus 企業開發人員移轉或建立的大型主機應用程式提供執行環境。

史诗

在 Amazon EC2 上設定微焦點企業伺服器，並部署大型主機線上應用程式

任務	描述	所需技能
設定 Micro Focus 企業伺服器並部署示範線上應用程式。	在 Amazon EC2 上設定微焦點企業伺服器，然後依照微焦點文件中的教學：CICS Support 中的指示，在 Amazon EC2 上部署微焦點帳戶示範應用程式 (ACCT 示範)。 ACCT 示範應用程式是一個大型主機線上 (CICS) 應用程式，可建立並啟動列印輸出。	雲端架構師

在 Amazon EC2 上設置 LRS 列印伺服器

任務	描述	所需技能
取得用於列印的 LRS 產品授權。	若要取得 LRS VPSX/MFI、LRS /列和LRS/DIS 的 LRS 產品授權，請聯絡 LRS 輸出管理團隊。您必須提供將安裝 LRS 產品之 EC2 執行個體的主機名稱。	建立領導
建立一個 Amazon EC2 視窗執行個體以安裝 LRS VPSX/MFI。	依照步驟 1 中的指示啟動 Amazon EC2 Windows 執行個體：啟動 Amazon EC2 文件中的執行個體。您的執行個體必須符合下列 LRS VPSX/MFI 的硬體和軟體需求： • 中央處理器 — 雙核心 • 公羊-16 GB	雲端架構師

任務	描述	所需技能
	• 磁碟機 — 500 GB • 最低 EC2 執行個體 — 大型 • 作業系統 — 視窗/linux • 軟體 — 網際網路資訊服務 (IIS) 或 Apache 注意：先前的硬體和軟體需求適用於小型印表機群 (約 500—1000)。若要取得完整要求，請諮詢您的 LRS 和 AWS 聯絡人。 當您建立 Windows 執行個體時，請執行下列動作： 1. 確認 EC2 主機名稱與用於 LRS 產品授權的主機名稱相同。 2. 完成以下操作，在 Amazon EC2 中啟用 CGI： a. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。 b. 在 Windows「開始」功能表中，找到並開啟「伺服器管理員」。 c. 在伺服器管理員中，選擇儀表板、快速入門、新增角色和功能。然後，選擇 [伺服器角色]。	

任務	描述	所需技能
	d. 在 [伺服器角色] 中，選擇 [WebServer (IIS)]，然後選擇 [應用程式開發]。 e. 在 [應用程式開發] 中，選取 [CGI] 核取方塊。 f. 請依照 Windows 伺服器管理員新增角色和功能精靈上的指示來安裝 CGI。 g. 在 EC2 執行個體的 Windows 防火牆中開啟連接埠 5500，以進行 LRS /列通訊。	

任務	描述	所需技能
在 EC2 執行個體上安裝 LRS VPSX/MFI。	1. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟產品下載頁面的連結。注意：LRS 產品通過電子文件傳輸（EFT）進行分發。 3. 下載 LRS VPSX/MFI 並解壓縮該文件（默認文件夾：）。`c:\LRS` 4. 從解壓縮的資料夾啟動 LRS 產品安裝程式，以安裝 LRS VPSX/MFI。 5. 在「選取功能」功能表中，選取「VPSX® 伺服器 (V1R3.022)」，然後選擇「下一步」以開始安裝程序。安裝完成後，您將收到成功訊息。	雲端架構師

任務	描述	所需技能
安裝 LRS /列。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS /queue，然後解壓縮檔案。 3. 移至您下載檔案的位置，然後啟動 LRS 產品安裝程式以安裝 LRS /queue。	雲端架構師
安裝 LRS/DIS。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從您應收到的 LRS 電子郵件中開啟 LRS 產品下載頁面的連結，下載 LRS/DIS，然後解壓縮檔案。 3. 移至您下載檔案的位置，然後啟動 LRS 產品安裝程式。 4. 在「LRS 產品安裝程式」中，展開「LRS 雜項工具」，選取「LRS DIS」，然後選擇「下一步」。 5. 依照 LRS 產品安裝程式中的其餘指示完成安裝程序。	雲端架構師

任務	描述	所需技能
建立目標群組並將 LRS VPSX/MFI EC2 註冊為目標。	依照 Elastic Load Balancing 說明文件中的為您的 Network Load Balancer 建立目標群組中的指示，建立目標群組。 建立目標群組時，請執行下列動作： 1. 在 [指定群組詳細資訊] 頁面上，選擇執行處理做為 [選擇目標類型]。 2. 針對通訊協定，選擇 TCP。 3. 在「連接埠」中，選擇 5500。 4. 在 [註冊目標] 頁面的 [可用執行個體] 區段中，選取 LRS VPSX/MFI EC2 執行個體。	雲端架構師

任務	描述	所需技能
建立 Network Load Balancer。	遵循 Elastic Load Balancing 說明文件中建立 Network Load Balancer 的指示。您的 Network Load Balancer 會將流量從微焦點企業伺服器路由到 LRS VPSX/MFI EC2。 建立 Network Load Balancer 時，請在「監聽器和路由」頁面執行下列動作： 1. 針對 Protocol (通訊協定)，選擇 TCP。 2. 在「連接埠」中，選擇 5500。 3. 對於「預設」動作，請為您先前建立的目標群組選擇「轉寄至」。	雲端架構師

將微焦企業伺服器與 LRS VPSX/MFI 和 LRS /隊列整合

任務	描述	所需技能
設定微焦點企業伺服器以進行 LRS /列整合。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 2. 在 Windows「開始」功能表中，開啟「微焦點企業伺服器管理」使用者介面。 3. 在選單列中，選擇「原生」。	雲端架構師

任務	描述	所需技能
	4. 在瀏覽窗格中，選擇 [目錄伺服器]，然後選擇 [BANKDEMO] 或 [企業伺服器] 區域。 5. 從左側導覽窗格的 [一般] 中，向下捲動至 [其他] 區段，以設定環境變數 (LRSQ_ADDRESS、LRSQ_PORT、LRSQ_命令) 以指向 LRSQ。 6. 對於 LRSQ_ADDRESS，請輸入您先前建立之 Network Load Balancer 的 IP 位址或 DNS 名稱。 7. 針對連接埠，請輸入 VPSXL RSQ 接聽程式連接埠 (5500)。 8. 對於 LRSQ_ 指令，請輸入 LRSQ 可執行檔的路徑位置。 9. 注意：LRS 目前支援的 DNS 名稱最大字元限制為 50，但 future 可能會有所變更。如果您的 DNS 名稱大於 50，則可以使用 Network Load Balancer 的 IP 位址作為替代方案。	

任務	描述	所需技能
讓 CICS 列印結束 (DFHUPRNT) 可供微焦點企業伺服器初始化使用。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 Micro Focus 企業伺服器 EC2 執行個體。 2. 將 CICS 列印結束 (DFHAPRNT) 從 LRS VPSX/MFI 可執行檔資料夾 (命名為VPSX_MFI_R2) 複製到微焦點企業伺服器 EC2 執行個體位置。對於 32 位系統，位置是C:\Program Files (x86) \Micro Focus \Enterprise Server \bin 。對於 64 位元系統，位置為C:\Program Files (x86) \Micro Focus\Enterprise Server\bin64 。注意：複製DFHUPRNT_64.dll 檔案DFHUPRNT.dll 時必須重新命名為。 驗證微焦點企業伺服器是否偵測到 CICS 列印結束 (DFHUPRNT) 1. 停止並啟動微焦點企業伺服器。 2. 在 Micro Focus 企業伺服器的「管理」面板中，開啟監視器、記錄檔和主控台記錄。	雲端架構師

任務	描述	所需技能
	3. 檢查控制台日誌是否有以下消息：「3270 打印機用戶退出 DFHUPRNT 安裝成功。」	

任務	描述	所需技能
將 CICS 印表機的終端機 ID (TERMID) 定義為微焦點企業伺服器。	啟用微焦點企業伺服器中的 3270 列印 1. 在 Micro Focus 企業伺服器的「管理」面板中，開啟 CICS、資源、按群組。 2. 在左側導覽面板中，選擇 SIT (系統初始化表格)，然後選擇 BN KICV。 3. 在 [一般] 區段中，向下捲動至 3270，然後選取 [3270 列印] 核取方塊。 在微焦點企業伺服器中定義 CICS 印表機的終端 1. 在 Micro Focus 企業伺服器的「管理」面板中，開啟 CICS、資源、按類型。 2. 從左側導覽面板中選擇「字詞」，然後選擇「新增」。建立終端機資源」表單隨即開啟。 3. 在名稱中，輸入 LRS 列印佇列的名稱。（注意：此模式使用 「P275」 作為 CICS 打印機的終端機 ID 和 LRS VPSX 打印隊列。） 4. 在「群組」中，輸入「銀行項」。 5. 對於自動安裝 — 型號，輸入 NO。	雲端架構師

任務	描述	所需技能
	6. 對於「端子識別碼-端子類型」，請輸入 DFHPRT32。 7. 針對「網路名稱」，輸入 VTAMP275。 8. 對於「終端機使用」，請選取「服務中」核取方塊。 9. 捲動頁面頂端，然後選擇 [儲存]。 10選擇 Install (安裝)。快顯訊息會顯示成功的安裝訊息。	

在微焦點企業服務器和 LRS VPSX/MFI 中設置打印機和打印用戶

任務	描述	所需技能
在 LRS VPSX 中建立列印佇列。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從視窗的「開始」功能表開啟 VPSX 網頁介面。 3. 在導覽窗格中，選擇 [印表機]。 4. 選擇 [新增]，然後選擇 [新增印表機]。 5. 在「印表機規劃」頁面上，對於「印表機名稱」，輸入 P275。 6. 針對 VPSX 識別碼，請輸入 VPS1。	雲端架構師

任務	描述	所需技能
	7. 對於 CommType，請選取「TCPIP /LR SQ」。 8. 對於主機 /IP 位址，輸入要新增之實體印表機的 IP 位址。 9. 在「裝置」中，輸入裝置的名稱。 10選擇任一視窗驅動程序或鏈接 /Mac 驅動程序。 11選擇新增。 注意：列印佇列必須等同於 Micro Focus 企業伺服器中建立的列印 TERMID。	

任務	描述	所需技能
在 LRS VPSX/MFI 中創建一個打印用戶。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 從視窗的「開始」功能表開啟 VPSX 網頁介面。 3. 在功能窗格中，選擇 [安全性]，然後選擇 [使用者]。 4. 在 [使用者名稱] 欄中，選擇 [管理員]，然後選擇 [複製]。 5. 在「使用者設定檔維護」視窗中，對於「使用者名稱」，輸入使用者名稱 (例如，PrintUser)。 6. 在「說明」中，輸入簡短描述 (例如，測試列印的使用者)。 7. 選擇更新。這將創建一個打印用戶（例如，PrintUser）。 8. 在導覽窗格的 [使用者] 下，選擇您建立的新使用者。 9. 從「命令」功能表中選擇「安全性」。 10. 在 [安全性規則] 頁面上，選擇所有適用的印表機安全性和工作安全性選項，然後選擇 [儲存]。 11. 若要將新的列印使用者新增至 [系統管理員] 群組，請	雲端架構師

任務	描述	所需技能
	移至功能窗格，選擇 [安全性]，然後選擇 [設定]。 12.在「安全性」組態視窗中，將新的列印使用者新增至「管理員」欄。	

設定列印驗證和授權

任務	描述	所需技能
使用使用者和群組建立 AWS 受管 Microsoft AD 網域。	1. 按照 AWS 目錄服務文件中的建立您的 AWS 受管 Microsoft AD 目錄中的指示，在 AWS 上建立活動目錄。 2. 依照 AWS 目錄服務文件中的步驟 3 中的指示，部署 EC2 執行個體 (使用中目錄管理員) 並安裝 Active Directory 工具來管理您的 AWS 受管 Microsoft AD：部署 EC2 執行個體以管理您的 AWS 受管 Microsoft AD。 3. 按照步驟 2：Connect Amazon EC2 文件中的執行個體的說明 Connect 到 EC2 執行個體。注意：連線至 EC2 執行個體時，請在 Windows 安全性視窗中輸入管理員登入資料 (針對您在步驟 1 中建立的目錄)。	雲端架構師

任務	描述	所需技能
	4. 在 Windows「開始」功能表的「Windows 系統管理工具」下，選擇「作用中目錄使用者和電腦」。 5. 按照 AWS 目錄服務文件中的建立使用者中的步驟，在 Active Directory 網域中建立列印使用者。	
將 LRS VPSX/MFI EC2 加入 AWS 受管 Microsoft AD 網域。	自動將 LRS VPSX/MFI EC2 加入您的 AWS 受管 Microsoft AD 網域 (AWS 知識中心文件) 或手動 (AWS Directory Service 文件)。	雲端架構師

任務	描述	所需技能
設定並整合 LRS/DIS 與 AWS 受管 Microsoft AD。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗「開始」功能表中，開啟 VPSX 網頁介面。 3. 在瀏覽窗格中，選擇 [安全性]，然後選擇 [設定]。 4. 在「安全組態」頁面的「安全參數」段落中，選取「內部」做為「安全類型」。 5. 在「安全性參數」區段中輸入其餘選項的偏好設定。 6. 從 Microsoft Windows 的「開始」功能表開啟「LRS 輸出管理」資料夾，選擇「伺服器啟動」，然後選擇「伺服器停止」。 7. 使用您的活動目錄用戶名和密碼登錄到 LRS VPSX/MFI。	雲端架構師

測試線上列印工作流程

任務	描述	所需技能
從 Micro Focus ACCT 示範應用程式啟動線上列印要求。	1. 在您的微焦點企業伺服器 EC2 執行個體中開啟 TN3270 終端機模擬器。（注意：此模式使用 3270 終端仿真器。）	雲端架構師

任務	描述	所需技能
	2. Connect 到 TN3270 終端機模擬器（倫巴）。對於主機名稱地址，請使用 127.0.0.1。對於遠端連接埠，請使用 9 270。 3. 連接到 3270 螢幕後，按下左右鍵以清除螢幕。 4. 若要啟動 ACCT 示範應用程式，請在清除畫面中輸入 ACCT。這會開啟 ACCT 線上示範 (CICS) 應用程式主畫面。注意：主屏幕包括菜單選項，例如帳戶文件，要按名稱搜索，請輸入，請求類型，帳戶和打印機。 5. 若要從 ACCT 線上示範 (CICS) 應用程式提交列印請求，請在請求類型欄位中輸入 P，在帳戶欄位中輸入 11111，在印表機欄位中輸入 P275。請務必將印表機欄位中的值設定為 CICS 印表機終端機 ID 的值。 6. 按 Enter。 畫面底部會出現「列印要求已排程」訊息。如此可確認 ACCT 示範應用程式產生線上列印要求，並傳送至 LRS VPS/MFI 進行列印處理。	

任務	描述	所需技能
在 LRS VPSX/MFI 中檢查打印輸出。	1. 按照步驟 2：Connect 到 Amazon EC2 文件中的執行個體的說明，Connect 到您的 LRS VPSX/MFI EC2 執行個體。 2. 在視窗「開始」功能表中，開啟 VPSX 網頁介面。 3. 在功能窗格中，選擇 [印表機]，然後選擇 [輸出佇列]。尋找您之前為線上列印建立的 P275 列印佇列。 4. 對於列印佇列 (P275)，在「多工緩衝處理 ID」欄中，為印表機佇列中的請求選擇多工緩衝處理 ID。 5. 在 [動作] 索引標籤的 [命令] 欄中，選擇 [瀏覽]。 現在，您可以看到帳戶對帳單的打印輸出，其中包含帳戶號碼，姓氏，第一，地址，電話，號碼的列。 信用卡發行、發行日期、金額及餘額。 如需範例，請參閱此樣式的線上列印輸出附件。	測試工程師

相關資源

- LRS 輸出現代化 (LRS 文件)
- VTAM 網路概念 (IBM 說明文件)
- 邏輯單位 (LU) 類型摘要 (IBM 說明文件)

- ANSI 和機器托架控制 (IBM 說明文件)
- 運用 Micro Focus 強化 AWS 上的企業大型主機工作負載 (AWS 合作夥伴網路部落格)
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建立微焦點企業伺服器 PAC (AWS Prescriptive Guidance 文件)
- 進階功能簡報 (AFP) 資料串流 (IBM 說明文件)
- 線路調節數據流（LCDS）（堆肥文檔）

其他資訊

考量

在您的現代化過程中，您可以考慮針對大型主機線上程序的各種組態及其產生的輸出。大型主機平台已由每位使用該平台的客戶和廠商自訂，這些平台符合直接影響列印的特定需求。例如，您目前的平台可能會將 IBM 進階功能簡報 (AFP) 或施樂生產線狀況資料串流 (LCDS) 納入目前的工作流程。此外，大型主機托架控制字元和 channel 指令字可能會影響列印頁面的外觀，並且可能需要特殊處理。作為現代化規劃程序的一部分，我們建議您評估並瞭解特定列印環境中的組態。

列印資料擷取

本節摘要列印 IBM 大型主機環境中可使用的 CICS 應用程式程式設計方法。LRS VPSX/MFI 組件提供的技術，以允許相同的應用程序以相同的方式創建數據。下表說明在 AWS 和 Micro Focus 企業伺服器 (搭配 LRS VPSX/MFI 列印伺服器) 中執行的現代化 CICS 應用程式中，如何支援每種應用程式程式設計方法。

方法	描述	Support 現代化環境中的方法
執行 CICS 發送文本.. 或者執行 CICS 發送地圖..	這些 CICS 和 VTAM 方法負責建立並提供 3270/SCS 列印資料串流至 LUTYPE0、LUTYPE1 和 LUTYPE3 列印裝置。	微焦點線上列印結束 (DFHUPRNT) 應用程式介面 (API) 可讓 VPSX/MFI 透過這些方法建立 3270/SCS 列印資料串流時，處理列印資料。
執行 CICS 發送文本.. 或者執行 CICS 發送地圖.. (使用協力廠商 IBM 大型主機軟體)	該 CICS 和 VTAM 方法負責創建和提供 3270/SCS 打印數據流到 LUTYPE0，LUTYPE1 和 LUTYPE3 打印設備。協力廠商軟體產品會攔截列印資料，	當使用以下任一方法創建 3270/SCS 打印數據流時，微型聚焦在線打印退出 API 使用 VPSX/MFI 處理打印數據。

	使用 ASA/MCH 控制字元將資料轉換為標準列印格式資料，並將資料放在使用 JES 的大型主機列印系統處理的 JES 捲軸上。	
執行 CICS 線軸線軸	這種方法是由 CICS 應用程序用於將數據直接寫入 JES 線軸。然後，使用 JES 的大型主機列印系統即可處理資料。	微焦點企業伺服器將資料多工緩衝處理到企業伺服器多工緩衝處理，VPSX/MFI Batch 列印結束 (LRSPRTE6) 可將資料多工緩衝處理至 VPSX。
DR/API	LRS 提供的程式化介面用於將列印資料寫入 JES。	VPSX/MFI 提供替換介面，可將列印資料直接捲動至 VPSX。

印表機機群健康檢查

LRS VPSX/MFI (LRS LoadX) 可以執行深入的運作狀態檢查，包括裝置管理和作業最佳化。裝置管理可偵測印表機裝置中的故障，並將列印要求路由至健康狀態良好的印表機。如需深入瞭解印表機叢集執行狀態檢查的詳細資訊，請參閱產品授權隨附的 LRS 文件。

列印驗證和授權

LRS/DIS 使 LRS 應用程序能夠通過使用 Microsoft 活動目錄或 LDAP 服務器來驗證用戶 ID 和密碼。除了基本的列印授權之外，LRS/DIS 也可以在下列使用案例中套用粒度層級的列印安全性控制：

- 管理誰可以瀏覽印表機工作。
- 管理其他使用者工作的瀏覽層級。
- 管理操作任務。例如，指令層級安全性，例如保留/釋放、清除、修改、複製和重定路由。安全性可以透過使用者識別碼或群組 (類似於 AD 群組或 LDAP 群組) 來設定。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Transfer Family 列將大型主機檔案直接移至 Amazon S3

創建者：路易斯古斯塔沃丹達斯 (AWS)

環境：生產	資料來源：大型機	目標：Amazon S3
R 類型：不適用	工作負載：IBM	技術：大型主機、儲存與備份、現代化
AWS 服務：AWS Transfer Family；Amazon S3		

Summary

作為現代化旅程的一部分，您可能會面臨在現場部署伺服器和 Amazon Web Services (AWS) 雲端之間傳輸檔案的挑戰。從大型主機傳輸資料可能是一項重大挑戰，因為大型主機通常無法存取現代資料存放區，例如 Amazon 簡單儲存服務 (Amazon S3)、Amazon Elastic Block Store (Amazon EBS) 或 Amazon Elastic File System (Amazon EFS)。

許多客戶使用中繼預備資源 (例如現場部署 Linux、Unix 或 Windows 伺服器) 將檔案傳輸到 AWS 雲端。您可以使用 AWS Transfer Family 搭配安全殼層 (SSH) 檔案傳輸協定 (SFTP)，將大型主機檔案直接上傳到 Amazon S3，以避免這種間接方法。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有可供您舊式平台存取之子網路的虛擬私有雲 (VPC)
- 適用於 VPC 的 Transfer Family 端點
- 大型主機虛擬儲存存取方法 (VSAM) 檔案轉換為連續的固定長度檔案 (IBM 說明文件)

限制

- SFTP 預設會以二進位模式傳輸檔案，這表示檔案會上傳至 Amazon S3，並保留 EBCDIC 編碼。如果您的檔案不包含二進位或封裝的資料，您可以在傳輸期間使用 sftp ascii 子指令 (IBM 文件) 將檔案轉換為文字。

- 您必須解壓縮包含已封裝和二進位內容的大型主機檔案 (AWS Prescriptive Guidance)，才能在目標環境中使用這些檔案。
- Amazon S3 物件的大小範圍從最少 0 個位元組到最大 5 TB。如需 Amazon S3 功能的詳細資訊，請參閱 Amazon S3 常見問答集。

架構

源, 技術, 堆棧

- Job 控制語言 (JCL)
- z/OS Unix 外殼與 ISPF
- SFTP
- VSAM 和平面文件

目標技術堆疊

- Transfer Family
- Amazon S3
- Amazon Virtual Private Cloud (Amazon VPC)

目標架構

下圖顯示使用傳輸系列搭配 SFTP 將大型主機檔案直接上傳到 S3 儲存貯體的參考架構。

該圖顯示以下工作流程：

1. 您可以使用 JCL 任務，透過直接 Connect 將大型主機檔案從舊式大型主機傳輸到 AWS 雲端。
2. 直接 Connect 可讓您的網路流量保留在 AWS 全球網路上，並略過公用網際網路。直接 Connect 還可以提高網絡速度，從 50 Mbps 開始，最高可擴展到 100 Gbps。
3. VPC 端點可讓您的 VPC 資源與受支援的服務之間進行連線，而無需使用公用網際網路。對 Transfer Family 和 Amazon S3 的存取透過位於兩個私有子網路和可用區域中的彈性網路界面來實現高可用性。
4. Transfer Family 會驗證使用者，並使用 SFTP 從舊版環境接收檔案，並將其移至 S3 儲存貯體。

自動化和規模

Transfer Family 服務就緒後，您可以使用 JCL 任務做為 SFTP 用戶端，將無限數量的檔案從大型主機傳輸到 Amazon S3。您也可以在準備好傳輸大型主機檔案時，使用大型主機批次工作排程器來執行 SFTP 工作，藉此自動化檔案傳輸。

工具

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。
- AWS Transfer Family 可讓您使用 SFTP、FTPS 和 FTP 協定，安全地將週期性 business-to-business 檔案傳輸擴展到 Amazon S3 和 Amazon EFS。

史诗

建立 S3 儲存貯體和存取政策

任務	描述	所需技能
建立 S3 儲存貯體。	建立 S3 儲存貯體以託管您從舊版環境傳輸的檔案。	一般 AWS
建立 IAM 角色和政策。	Transfer Family 使用您的 AWS Identity and Access Management (IAM) 角色來授予您先前建立之 S3 儲存貯體的存取權。 建立包含下列 IAM 政策的 IAM 角色： `{ "Version": "2012-10-17", "Statement": [`	一般 AWS

任務	描述	所需技能
	{ "Sid": "UserFolderListing", "Action": ["s3:ListBucket", "s3:GetBucketLocation"], "Effect": "Allow", "Resource": ["arn:aws:s3:::<your-bucket-name>"] }, { "Sid": "HomeDirObjectAccess", "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObjectAcl", "s3:GetObject", "s3:DeleteObjectVersion", "s3:DeleteObject", "s3:PutObjectAcl", "s3:GetObjectVersion"],	

任務	描述	所需技能
	<pre>"Resource": "arn:aws:s3:::<your-bucket-name>/*" }] }</pre> 附註：建立 IAM 角色時，必須選擇「移轉」使用案例。	

定義轉移服務

任務	描述	所需技能
建立 SFTP 伺服器。	1. 登入 AWS 管理主控台，開啟 Transfer Family 主控台，然後選擇 [建立伺服器]。 2. 只選擇 SFTP (SSH 檔案傳輸通訊協定)-透過安全殼層通訊協定進行檔案傳輸，然後選擇 [下一步]。 3. 對於身分識別提供者，請選擇「服務管理」，然後選擇「 4. 針對「端點類型」，選擇「VPC 託管」。 5. 針對 [存取] 選擇 [內部]。 6. 在 VPC 中，選擇您的 VPC。 7. 在 [可用區域] 區段中，選擇您的可用區域和子網路。 8. 在 [安全性群組] 區段中，選擇您的安全性群組，然後選擇 [下一步]。	一般 AWS

任務	描述	所需技能
	9. 對於網域，選擇 Amazon S3，然後選擇下一步。 10.保留 [設定其他詳細資料] 頁面上的預設選項，然後選擇 [下一步]。 11選擇 Create server (建立伺服器)。 注意：如需如何設定 SFTP 伺服器的詳細資訊，請參閱建立啟用 SFTP 的伺服器 (AWS Transfer Family 使用者指南)。	
取得伺服器位址。	1. 開啟「Transfer Family」主控台，然後在「伺服器 ID」欄中選擇您的伺服器 ID。 2. 在「端點詳細資料」區段中，對於「端點類型」，選擇端點 ID。這將帶您進入 Amazon VPC 控制台。 3. 在 Amazon VPC 主控台的 [詳細資料] 索引標籤上，找到 DNS 名稱旁邊的 DNS 名稱。	一般 AWS
建立 SFTP 用戶端 key pair。	為 Microsoft 視窗或澳門/ Linux /Unix 創建一個安全殼層 key pair。	一般 AWS、SSH

任務	描述	所需技能
建立 SFTP 使用者。	1. 開啟「Transfer Family」主控台，從導覽窗格中選擇「伺服器」，然後選取您的伺服器。 2. 在「伺服器 ID」欄中，選擇伺服器的伺服器 ID，然後選擇「新增使用者」。 3. 在使用者名稱中，輸入與您的 SSH key pair 使用者名稱相符的使用者名稱。 4. 針對「角色」，選擇您先前建立的 IAM 角色。 5. 對於主目錄，請選擇您先前建立的 S3 儲存貯體。 6. 對於 SSH 公開金鑰，請輸入您先前建立的 key pair。 7. 選擇新增。	一般 AWS

傳輸大型主機檔案

任務	描述	所需技能
將 SSH 私密金鑰傳送至大型主機。	使用 SFTP 或 SCP 將安全殼層私密金鑰傳送至舊版環境。 SFTP 範例： `sftp [USERNAME@mainframeIP]` `[password]` `cd [/u/USERNAME]` `put [your-key-pair-file]`	大型主機、z/OS Unix 外殼、FTP、SCP

任務	描述	所需技能
	SCP 範例： `scp [your-key-pair-file] [USERNAME@MainframeIP]:/[u/USERNAME]` 接下來，將 SSH 金鑰儲存在 z/OS Unix 檔案系統的使用者名稱下，以便稍後執行檔案傳輸批次工作 (例如`/u/CONTROLM`)。 注意：如需 z/OS Unix 殼層的詳細資訊，請參閱 z/OS 殼層簡介 (IBM 說明文件)。	

任務	描述	所需技能
創建 JCL SFTP 客戶端。	因為大型主機沒有原生 SFTP 用戶端，因此您必須使用 BPXBATCH 公用程式，從 z/OS Unix 殼層執行 SFTP 用戶端。 在 ISPF 編輯器中，建立 JCL SFTP 用戶端。例如： `//JOBNAM JOB ...` `//************` `******************` `******************` `******************` `****` `//SFTP EXEC PGM=BPXBA` `TCH,REGION=0M` `//STDPARM DD *` `SH cp "//'MAINF` `RAME.FILE.NAME'"` ` filename.txt;` `echo 'put filename.txt'` ` > uplcmd;` `sftp -b uplcmd -i` ` ssh_private_key_fi` `le ssh_username@<tran` `sfer service ip or` ` DNS>;` `//SYSPRINT DD SYSOUT=*` `//STDOUT DD SYSOUT=*` `//STDENV DD *` `//STDERR DD SYSOUT=*` 注意：如需如何在 z/OS Unix 殼層中執行命令的詳細資訊，請參閱 BPXBATCH 公用程式 (IBM 說明文件)。如需如何在 z/OS 中建立或編輯 JCL 工作	JCL，大型機，z/OS Unix 外殼

任務	描述	所需技能
	的詳細資訊，請參閱什麼是 ISPF？ 以及 ISPF 編輯器 (IBM 說明文件)。	
運行 JCL SFTP 客戶端。	1. 在 ISPF 編輯器中，輸入 SUB，然後在建立 JCL 工作之後按 ENTER 鍵。 2. 監視 SDSF 中大型主機的檔案傳輸批次工作活動。 附註：如需如何檢查批次任務活動的詳細資訊，請參閱 z/OS SDSF 使用者指南 (IBM 說明文件)。	大型主機, JCL, ISPF
驗證檔案傳輸。	1. 登入 AWS 管理主控台，開啟 Amazon S3 主控台，然後從導覽窗格中選擇「儲存貯體」。 2. 選擇與您的 Transfer Family 相關聯的值區。 3. 在 [物件] 索引標籤的 [物件] 區段中，尋找您從大型主機傳輸的檔案。	一般 AWS
自動執行 JCL SFTP 用戶端。	使用作業排程器自動觸發 JCL SFTP 用戶端。 備註：您可以使用大型主機作業排程器 (例如 BMC Controll-M 或 CA 工作負載自動化)，根據時間和其他批次工作相依性自動化檔案傳輸的批次工作。	Job 排程器

相關資源

- AWS Transfer Family 如何運作
- 使用 AWS 進行大型主機現代化

以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3

創建者：布魯諾·薩希諾格魯 (AWS)、伊萬舒斯特 (AWS) 和阿比希爾薩加爾 (AWS)

程式碼儲存庫：將 DB2 z/OS 卸載至 S3	環境：生產	資料來源:Db2
目標：Amazon S3	R 類型：重新平台	工作負載：IBM
技術：大型主機；資料湖；資料庫；軟體開發與測試；移轉	AWS 服務：Amazon Aurora; AWS AWS Glue; Amazon S3; AWS Transfer Family; Amazon Athena	

Summary

在許多企業中，大型主機仍然是一個記錄系統，其中包含大量數據，包括具有當前記錄以及歷史業務交易記錄的主數據實體。它通常是孤立的，並且不容易被同一企業內的分佈式系統訪問。隨著雲端技術的出現和大數據民主化，企業有興趣使用隱藏在大型主機資料中的洞察力來開發新的業務能力。

有了這個目標，企業希望開放他們的大型主機 Db2 資料到他們的 Amazon Web Services (AWS) 雲端環境。業務原因有幾種，並且轉移方法因情況而異。您可能希望將應用程式直接連接到大型主機，或者您可能希望以近乎即時的方式複製資料。如果使用案例是饋送資料倉儲或資料湖，則不再需要考慮 up-to-date 複本，而且此模式中描述的程序可能就足夠了，特別是如果您想要避免任何協力廠商產品授權成本。另一個使用案例可能是移轉專案的大型主機資料傳輸。在移轉案例中，執行功能對等測試需要資料。本文中描述的方法是將 Db2 資料傳輸到 AWS 雲端環境的具有成本效益的方法。

由於 Amazon 簡單儲存服務 (Amazon S3) 是整合度最高的 AWS 服務之一，因此您可以使用其他 AWS 服務 (例如亞馬遜雅典娜、AWS Lambda 函數或亞馬遜) 存取資料並直接收集見解 QuickSight 。您也可以使用 AWS Glue 或 AWS Database Migration Service (AWS DMS)，將資料載入 Amazon Aurora 或亞馬遜動態資料庫。考慮到這一目標，這說明如何在大型主機上以 ASCII 格式卸載 CSV 檔案中的 Db2 資料，並將檔案傳輸到 Amazon S3。

為了這個目的，大型主機腳本已經開發，以幫助生成作業控制語言（JCLs）卸載和傳輸盡可能多的 Db2 表，因為你需要。

先決條件和限制

先決條件

- 具有執行重新結構化延伸執行程式 (REXX) 和 JCL 指令碼之授權的 IBM z/OS 作業系統使用者。
- 存取 z/OS Unix 系統服務 (USS) 以產生 SSH (安全殼層) 私密金鑰和公開金鑰。
- 可寫入的 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的建立您的第一個 S3 儲存貯體。
- 支援 AWS Transfer Family SSH 檔案傳輸協定 (SFTP) 的伺服器，使用以身分供應商身分管理的服務，Amazon S3 做為 AWS 儲存服務。如需詳細資訊，請參閱 AWS Transfer Family 文件中的建立啟用 SFTP 的伺服器。

限制

- 此方法不適用於近乎即時或即時的資料同步。
- 資料只能從 Db2 z/OS 移動到 Amazon S3，而不能相反地移動。

架構

源, 技術, 堆棧

- 在 z/OS 上執行 Db2 的大型主機

目標技術堆疊

- AWS Transfer 系列
- Amazon S3
- Amazon Athena
- Amazon QuickSight
- AWS Glue
- Amazon Relational Database Service (Amazon RDS)
- Amazon Aurora
- Amazon Redshift

來源與目標架構

下圖顯示產生、擷取及傳輸 ASCII CSV 格式的 Db2 z/OS 資料到 S3 儲存貯體的程序。

1. 從 Db2 目錄中選取要進行資料移轉的表格清單。
2. 此清單用於驅動卸載工作的產生，其中包含外部格式的數值和資料欄。
3. 然後，使用 AWS 傳輸系列將資料傳輸到 Amazon S3。
4. AWS Glue 擷取、轉換和載入 (ETL) 任務可以轉換資料，並以指定格式將資料載入已處理的儲存貯體，或者 AWS Glue 可以將資料直接輸入資料庫。
5. Amazon Athena 和 Amazon QuickSight 可用於查詢和呈現數據以推動分析。

下圖顯示了整個過程的邏輯流程。

1. 第一個 JCL，稱為 TABNAME，將使用 Db2 實用程序 DSNTIAUL 提取並生成您打算從 Db2 卸載的表的列表。若要選擇表格，您必須手動調整 SQL 輸入，以選取並新增篩選準則，以包含一或多個 Db2 綱要。
2. 第二 JCL，稱為 REXXEXEC，將使用一個 JCL 骨架和提供用於處理由 JCL TABNAME 創建的表列表中的 REXX 程序，並生成每個表名一個 JCL。每個 JCL 將包含一個用於卸載表的步驟，並使用 SFTP 協議將文件發送到 S3 存儲桶的另一個步驟。
3. 最後一步包括運行 JCL 卸載表並將文件傳輸到 AWS。您可以使用內部部署或 AWS 上的排程器自動化整個程序。

工具

AWS 服務

- Amazon Athena 是一種互動式查詢服務，可協助您使用標準 SQL 直接在亞馬遜簡單儲存服務 (Amazon S3) 中分析資料。
- Amazon Aurora 是全受管的關聯式資料庫引擎，專為雲端建置，並與 MySQL 和 PostgreSQL 相容。
- AWS Glue 是全受管的擷取、轉換和載入 (ETL) 服務。它可協助您在資料存放區和資料串流之間可靠地分類、清理、擴充和移動資料。
- Amazon QuickSight 是雲端規模商業智慧 (BI) 服務，可協助您在單一儀表板中視覺化、分析和報告資料。

- Amazon Redshift 是 AWS 雲端中的受管 PB 級資料倉儲服務。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Transfer Family 是一種安全的傳輸服務，可讓您將檔案傳入和傳出 AWS 儲存服務。

大型主機工具

- SSH 檔案傳輸通訊協定 (SFTP) 是一種安全的檔案傳輸通訊協定，可在伺服器之間進行遠端登入和傳輸檔案。SSH 通過加密所有流量來提供安全性。
- DSNTIAUL 是 IBM 提供的用於卸載數據的示例程序。
- DSNUTILB 是 IBM 提供的公用事業批處理程序，用於從 DSNTIAUL 不同的選項卸載數據。
- z/OS OpenSSH 是在 IBM 作業系統 z/OS 下，在 Unix 系統服務上執行的開放原始碼軟體 SSH 連接埠。SSH 是在 TCP/IP 網絡上運行的兩台計算機之間的安全，加密的連接程序。它提供了多種實用程序，包括 SSH 凱基。
- REXX（重組擴展執行程序）腳本用於使用 Db2 卸載和 SFTP 步驟自動化 JCL 生成。

Code

此模式的代碼可在 GitHub unloaddb 2 存儲庫中找到。

最佳實務

對於第一次卸載，生成的 JCL 應卸載整個表數據。

在第一次完全卸載之後，執行增量卸載以獲得更好的性能和成本節省。pdate 模板 JCL 組合中的 SQL 查詢以適應卸載過程的任何更改。

您可以手動轉換結構定義，也可以使用 Lambda 上的指令碼，並將 Db2 SYSPINCH 作為輸入來轉換結構定義。對於工業程序，AWS Schema Conversion Tool (SCT) 是偏好的選項。

最後，使用大型主機型排程器或 AWS 上的排程器，搭配大型主機上的代理程式，協助管理和自動化整個程序。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	如需指示，請參閱建立您的第一個 S3 儲存貯體。	一般 AWS

設定 Transfer Family 伺服器

任務	描述	所需技能
建立啟用 SFTP 的伺服器。	若要在 AWS Transfer Family 列主控台上開啟並建立 SFTP 伺服器，請執行以下操作： 1. 在 [選擇通訊協定] 頁面上，選取 [SFTP (SSH 檔案傳輸通訊協定)-透過安全殼層傳輸檔案] 核取方塊。 2. 針對身分識別提供者，選擇「服務管理」。 3. 對於端點，選擇「可公開存取」。 4. 對於域，選擇 Amazon S3。 5. 在 [設定其他詳細資料] 頁面上，保留預設設定。 6. 建立伺服器。	一般 AWS
為 Transfer Family 建立 IAM 角色。	若要為 Transfer Family 列建立 AWS Identity and Access Management (IAM) 角色以存取 Amazon S3，請按照建立 IAM 角色和政策中的指示進行操作。	AWS 管理員

任務	描述	所需技能
新增 Amazon S3 服務受管使用者。	若要新增 Amazon S3 服務受管使用者，請遵循 AWS 文件中的指示，並使用您的大型主機使用者 ID。	一般 AWS

保護通訊協定

任務	描述	所需技能
建立安全殼層金鑰。	在您的大型主機 USS 環境下，執行下列命令。 `ssh-keygen -t rsa` 注意：當系統提示輸入密碼時，請將其保持空白。	大型主機開發人員
為 SSH 資料夾和金鑰檔提供正確的授權等級。	默認情況下，公鑰和私鑰將存儲在用戶目錄中`/u/home/username/.ssh` 。 您必須將授權 644 授權給密鑰文件，並將 700 授予該文件夾。 `chmod 644 .ssh/id_rsa` `chmod 700 .ssh`	大型主機開發人員
將公開金鑰內容複製到您的 Amazon S3 服務受管使用者。	若要複製美國產生的公開金鑰內容，請開啟 AWS Transfer Family 主控台。 1. 在導覽窗格中，選擇 Servers (伺服器)。	大型主機開發人員

任務	描述	所需技能
	2. 在「伺服器 ID」欄中選擇識別碼，以查看伺服器詳細資料 3. 在 [使用者] 下，選擇使用者名稱以查看使用者詳細資料 4. 在 [SSH 公開金鑰] 下，選擇 [新增安全殼層公開金鑰]，將公開金鑰新增至使用者。對於 SSH 公鑰，請輸入您的公鑰。您的密鑰已通過服務驗證，然後您可以添加新用戶。 5. 選擇 Add key (新增金鑰)。	

產生 JCL

任務	描述	所需技能
產生範圍內的 Db2 資料表清單。	提供輸入 SQL 以建立範圍用於資料移轉的資料表清單。此步驟需要您使用 SQL WHERE 子句來指定決定 Db2 目錄表格 SYSIB.SYSTABLE 的選取準則。您可以自訂篩選器，以包含以特定前置字元開頭的特定結構描述或資料表名稱，或以遞增卸載的時間戳記為基礎。輸出會擷取在大型主機上的實體順序 (PS) 資料集中。該數據集將作為輸入 JCL 生成的下一階段。	大型主機開發人員

任務	描述	所需技能
	在使用 JCL TABNAME（如有必要，您可以重命名它），進行以下更改： 1. <Jobcard>以工作類別和授權執行 Db2 公用程式的使用者取代。 2. 替代<HLQ1>或自訂輸出資料集名稱，以符合您的網站標準。 3. 根據您的網站標準，更新 PDSEs（分區數據集擴展）的 STEPLIB 堆棧。此模式中的範例使用 IBM 預設值。 4. 使用特定於安裝的值取代 PLAN 名稱和 LIB。 5. 取代 <Schema><Prefix>Db2 目錄的選取條件。 6. 將生成的 JCL 保存在 PDS（分區數據集）庫中。 7. 提交 JCL。 Db2 表格清單萃取工作 `<Jobcard>` `//*` `//* UNLOAD ALL THE TABLE` `NAMES FOR A PARTICULAR` `SCHEMA` `//*` `//STEP01 EXEC PGM=IEFBR` `14` `//*` `//DD1 DD DISP=(MOD` `,DELETE,DELETE),` `// UNIT=SYSDA,`	

任務	描述	所需技能
	`//  SPACE=(1000,` `(1,1)),` `//  DSN=<HLQ1` `>.DSN81210.TABLIST` `//*` `//DD2  DD  DISP=(MOD` `,DELETE,DELETE),` `//  UNIT=SYSDA,` `//  SPACE=(1000,` `(1,1)),` `//  DSN=<HLQ1` `>.DSN81210.SYSPUNCH` `//*` `//UNLOAD  EXEC PGM=IKJEF` `T01,DYNAMNBR=20` `//SYSTSPRT DD` ` SYSOUT=*` `//STEPLIB  DD` ` DISP=SHR,DSN=DSNC1` `0.DBCG.SDSNEXIT` `//  DD  DISP=SHR,` `DSN=DSNC10.SDSNLOAD` `//  DD  DISP=SHR,` `DSN=CEE.SCEERUN` `//  DD  DISP=SHR,` `DSN=DSNC10.DBCG.RU` `NLIB.LOAD` `//SYSTSIN  DD  *` `  DSN SYSTEM(DBCG)` `  RUN  PROGRAM(D` `SNTIAUL) PLAN(DSNT` `IB12) PARMS('SQL') -` `  LIB('DSNC` `10.DBCG.RUNLIB.LOAD')` `  END` `//SYSPRINT DD SYSOUT=*` `//*` `//SYSUDUMP DD SYSOUT=*` `//*` `//SYSREC00 DD DISP=(NEW` `,CATLG,DELETE),`	

任務	描述	所需技能
	`//  UNIT=SYSDA,SPACE=(32760,(1000,500)),` `//  DSN=<HLQ1>.DSN81210.TABLIST` `//*` `//SYSPUNCH DD DISP=(NEW,CATLG,DELETE),` `//  UNIT=SYSDA,SPACE=(32760,(1000,500)),` `//  VOL=SER=SCR03,RECFM=FB,LRECL=120,BLKSIZE=12` `//  DSN=<HLQ1>.DSN81210.SYSPUNCH` `//*` `//SYSIN  DD *` `  SELECT CHAR(CREATOR), CHAR(NAME)` `  FROM SYSIBM.SYSTABLES` `  WHERE OWNER = '<Schema>'` `  AND NAME LIKE '<Prefix>%'` `  AND TYPE = 'T';` `/*`	

任務	描述	所需技能
修改 JCL 樣板。	所提供的這種模式的 JCL 模板包含一個通用的作業卡和庫名稱。不過，大多數大型主機網站都會有自己的資料集名稱、程式庫名稱和工作卡的命名標準。例如，執行 Db2 作業可能需要特定的工作類別。Job 項目子系統實作 JES2 和 JES3 可以強制執行其他變更。標準載入程式庫可能具有與 IBM 預設值不同的第一個限定詞。SYS1因此，在執行範本之前，請先自訂樣板以說明網站特有的標準。 在 JCL 的骨架中進行以下更改： 1. 使用授權執行 Db2 公用程式的工作類別和使用者修改工作卡。 2. 自訂輸出資料集名稱以符合您的網站標準。 3. 根據您的站點標準更新 PDS 的步驟庫堆棧。此模式中的範例使用 IBM 預設值。 4. <DSN>以您的 Db2 子系統名稱和關聯識別碼取代。 5. 將產生的 JCL 儲存在屬於 ISPSLIB 堆疊 (ISPF 的標準骨架範本程式庫) 的一部分的 PDS 程式庫中。 卸載和 SFTP JCL 骨架	大型主機開發人員

任務	描述	所需技能
	`//&USRPFX.U JOB (DB2UNLOAD),'JOB', CLASS=A,MSGCLASS=A, // TIME=1440 ,NOTIFY=&USRPFX //* DELETE DATASETS //STEP01 EXEC PGM=IEFBR14 //DD01 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSD A, // SPACE=(TR K,(1,1)), // DSN=&USRPFX..DB2.P UNCH.&JOBNAME //DD02 DD DISP=(MOD ,DELETE,DELETE), // UNIT=SYSD A, // SPACE=(TR K,(1,1)), // DSN=&USRPFX..DB2.U NLOAD.&JOBNAME //* //* RUNNING DB2 EXTRACTION BATCH JOB FOR AWS DEMO //* //UNLD01 EXEC PGM=DSNUTILB,REGIO N=0M, // PARM='<DSN>,UNLOAD' //STEPLIB DD DISP=SHR,DSN=DSNC1 0.DBCG.SDSNEXIT // DD DISP=SHR, DSN=DSNC10.SDSNLOAD //SYSPRINT DD SYSOUT=* //UTPRINT DD SYSOUT=* //SYSOUT DD SYSOUT=*`	

任務	描述	所需技能
	`//SYSPUN01 DD` ` DISP=(NEW,CATLG,DE` `LETE),` `//  SPACE=(CY` `L,(1,1),RLSE),` `// DSN=&USRPFX..DB2.P` `UNCH.&JOBNAME` `//SYSREC01 DD` ` DISP=(NEW,CATLG,DE` `LETE),` `//  SPACE=(CY` `L,(10,50),RLSE),` `// DSN=&USRPFX..DB2.U` `NLOAD.&JOBNAME` `//SYSPRINT DD SYSOUT=*` `//SYSIN  DD *` `  UNLOAD` `  DELIMITED COLDEL ','` `  FROM TABLE &TABNAME` `  UNLDDN SYSREC01` `  PUNCHDDN SYSPUN01` `  SHRLEVEL CHANGE` ` ISOLATION UR;` ` /*` `//*` `//* FTP TO AMAZON S3` ` BACKED FTP SERVER IF` ` UNLOAD WAS SUCCESSFUL` `//*` `//SFTP EXEC PGM=BPXBA` `TCH,COND=(4,LE),RE` `GION=0M` `//STDPARM DD *` ` SH cp "//'&USRP` `FX..DB2.UNLOAD.&JO` `BNAME'"` `  &TABNAME..csv;` ` echo "ascii` ` " >> uplcmd;` ` echo "PUT &TABNAME.` `.csv " >>>> uplcmd;`	

任務	描述	所需技能
	sftp -b uplcmd -i .ssh/id_rsa &FTPUSER.@&FTPSITE; rm &TABNAME..csv; //SYSPRINT DD SYSOUT=* //STDOUT DD SYSOUT=* //STDENV DD * //STDERR DD SYSOUT=*	

任務	描述	所需技能
產生「整批卸載 JCL」。	這一步涉及通過使用 JCL 的 ISPF 環境下運行一個 REXX 腳本。提供在第一個步驟中建立的範圍內表格清單，做為針對名稱產生大量 JCL 的輸入。TABLIST DD該 JCL 將生成針對名稱指定的用戶指定的分區數據集每個表名一個新的 JCL。ISPFILE DD事先分配此圖書館。每個新的 JCL 將有兩個步驟：一個步驟將 Db2 表卸載到一個文件中，並將文件發送到 S3 存儲桶的一個步驟。 在 JCL REXXEXEC 中進行下列變更 (您可以變更名稱)： 1. Job card user ID以對資料表具有卸載權限的大型主機使用者識別碼來取代。替代SYSPROC、ISPPLIB、ISP和ISPTLIB<HLQ1>值或自訂DSN以符合您的場地標準。若要找出特定於安裝的值，請使用指令。TSO ISRDDN 2. 使用在安裝中具有工作執行權限的使用者 ID 來取代<MFUSER>。 3. <FTPUSER> 用在安裝中具有 USS 和 FTP 權限的用戶 ID 替代。假設此使用者識別碼及其 SSH 安全金鑰位於	大型主機開發人員

任務	描述	所需技能
	大型主機上適當的 Unix 系統服務目錄中。 4. 替代 <AWS TransferFamily IP> AWS Transfer Family IP 地址或網域名稱。此位址將用於 SFTP 步驟。 5. 申請網站的標準住宿和更新 REXX 程序如下所述後提交 JCL。 大量 JCL 產生工作 `//RUNREXX JOB (CREATEJCL),'RUNS ISPF TABLIST',` `CLASS=A,MSGCLASS=A,` `//         TIME=1440` `,NOTIFY=&SYSUID` `//* Most of the values` `  required can be updated` `  to your site specific` `//* values using the` `  command 'TSO ISRDDN' in` `  your ISPF session.` `//* Update all the lines` `  tagged with //update` `  marker to desired` `//* site specific` `  values.` `//ISPF EXEC PGM=IKJEFT01,REGION=2048K,DYNAMNBR=25` `//SYSPROC   DD` `  DISP=SHR,DSN=USER.Z23D.CLIST`	

任務	描述	所需技能
	`//SYSEXEC DD DISP=SHR,DSN=<HLQ1>.TEST.REXXLIB` `//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU` `//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU` `// DD DISP=SHR,DSN=<HLQ1>.TEST.ISPSLIB` `//ISPMLIB DD DSN=ISP.SISPMENU,DISP=SHR` `//ISPTLIB DD DDNAME=ISPTABL` `// DD DSN=ISP.SISPTENU,DISP=SHR` `//ISPTABL DD LIKE=ISP.SISPTENU,UNIT=VIO` `//ISPPROF DD LIKE=ISP.SISPTENU,UNIT=VIO` `//ISPLOG DD SYSOUT=*,RECFM=VA,LRECL=125` `//SYSPRINT DD SYSOUT=*` `//SYSTSPRT DD SYSOUT=*` `//SYSUDUMP DD SYSOUT=*` `//SYSDBOUT DD SYSOUT=*` `//SYSTSPRT DD SYSOUT=*` `//SYSUDUMP DD SYSOUT=*` `//SYSDBOUT DD SYSOUT=*`	

任務	描述	所需技能
	`` //SYSHELP DD DSN=SYS1.HELP,DISP =SHR //SYSOUT DD SYSOUT=* //* Input list of tablenames //TABLIST DD DISP=SHR,DSN=<HLQ1 >.DSN81210.TABLIST //* Output pds //ISPFILE DD DISP=SHR,DSN=<HLQ1 >.TEST.JOBGEN //SYSTSIN DD * ISPSTART CMD(ZSTEPS <MFUSER> <FTPUSER> <AWS TransferFamily IP>) /* ``` 在您使用 REXX 指令碼之前，請進行下列變更： 1. 將 REXX 腳本保存在SYSEXEC堆棧下定義的 PDS 庫中，在上一個步驟中以 ZSTEPS 作為成員名稱編輯的 JCL REXXEXEC 中。如果你想重命名它，你應該更新 JCL 以滿足您的需求。 2. 此指令碼會使用追蹤選項來列印其他資訊，以防發生錯誤。您可以改為在EXECIO、ISPEXEC和TSO陳述式之後加入錯誤處理程式碼，然後移除追蹤行。 3. 此指令碼使用LODnnnnn命名慣例產生成員名稱，該慣例最多可支援 100,000 個成	

任務	描述	所需技能
	員。如果您有超過 100,000 個表格，請使用較短的前置字元，並調整tempjob陳述式中的數字。 步驟腳本	

```
/*REXX - - - - - - - - - -
 - - - - - - - - - - - - - -
 - - - - - - - - - - - */
/* 10/27/2021 - added
 new parms to accommoda
te ftp */
Trace "o"
    parse arg usrpfx
 ftpuser ftpsite
    Say "Start"
    Say "Ftpuser: "
 ftpuser "Ftpsite:"
 ftpsite
    Say "Reading table
 name list"
    "EXECIO * DISKR
 TABLIST (STEM LINE.
 FINIS"
    DO I = 1 TO LINE.0
      Say I
      suffix = I
      Say LINE.i
      Parse var LINE.i
 schema table rest
      tabname =
 schema !! "." !! table
      Say tabname
      tempjob= "LOD" !!
 RIGHT("0000" !! i, 5)
      jobname=tempjob
      Say tempjob
      ADDRESS ISPEXEC
 "FTOPEN "
```

任務	描述	所需技能
	<pre>ADDRESS ISPEXEC "FTINCL UNLDSKEL" /* member will be saved in ISPDSN library allocated in JCL */ ADDRESS ISPEXEC "FTCLOSE NAME("tem pjob")" END ADDRESS TSO "FREE F(TABLIST) " ADDRESS TSO "FREE F(ISPFILE) " exit 0</pre>	

運行 JCL

任務	描述	所需技能
執行「Db2 卸載」步驟。	JCL 生成後，你將有盡可能多的 JCLs，因為你有需要卸載的表。 這個故事使用 JCL 生成的例子來解釋結構和最重要的步驟。 您不需要執行任何操作。以下資料只供參考。如果您打算提交您在上一個步驟中產生的 JCLs，請跳至提交 LODNNNNN JCLs 工作。 當使用 JCL 與 IBM 提供的 DSNUTILB Db2 實用程序卸載 Db2 數據時，必須確保卸	大型主機開發人員、系統工程師

任務	描述	所需技能
	載的數據不包含壓縮的數字數據。若要完成此操作，請使用 DSNUTILB DELIMITED 參數。 此DELIMITED 參數支援卸載 CSV 格式的資料，方法是將字元新增為文字欄位的分隔符號和雙引號、移除 VARCHAR 欄中的間距，並將所有數值欄位轉換為外部 FORMAT FORMAT (包括日期欄位)。 下面的例子顯示了在生成的 JCL 卸載步驟是什麼樣子，使用逗號字符作為分隔符。 `UNLOAD DELIMITED COLDEL ',' FROM TABLE SCHEMA_NAME.TBNAME UNLDDN SYSREC01 PUNCHDDN SYSPUN01 SHRLEVEL CHANGE ISOLATION UR;`	

任務	描述	所需技能
執行 SFTP 步驟。	若要使用來自 JCL 的 SFTP 通訊協定，請使用 BPXBATCH 公用程式。 SFTP 公用程式無法直接存取 MVS 資料集。您可以使用 copy 指令 (cp) 將順序檔案複製&USRPFX..DB2.UNLOAD.&JOBNAME 到 USS 目錄中，它會變成此目錄&TABNAME..csv 。 使用私密金鑰 (id_rsa) 執行sftp命令，並使用 RACF 使用者 ID 做為使用者名稱，以連接到 AWS Transfer Family IP 地址。 `SH cp "//'&USRPFX..DB2.UNLOAD.&JOBNAME'" &TABNAME..csv; echo "ascii " >> uplcmd; echo "PUT &TABNAME..csv " >>>> uplcmd; sftp -b uplcmd -i .ssh/id_rsa &FTPUSER.@&FTP_TF_SITE; rm &TABNAME..csv;`	大型主機開發人員、系統工程師

任務	描述	所需技能
提交 LODNNNN JCL。	先前的 JCL 已經生成了所有需要卸載，轉換為 CSV，並轉移到 S3 存儲桶的 LoDNNNN JCL 表。 對已產生的所有 JCL 執行`submit`命令。	大型主機開發人員、系統工程師

相關資源

如需有關本文件所使用之不同工具和解決方案的詳細資訊，請參閱下列內容：

- OpenSSH 使用者指南
- Db2 z/OS — 卸載控制陳述式範例
- Db2 z/OS — 卸載分隔檔案
- Transfer Family — 建立啟用 SFTP 的伺服器
- Transfer Family — 與服務管理的使用者合作

其他資訊

在 Amazon S3 上取得 Db2 資料之後，您有許多方法可以開發新的見解。由於 Amazon S3 與 AWS 資料分析服務整合，因此您可以在分散式端自由使用或公開這些資料。例如，您可以執行下列動作：

- 在 Amazon S3 上建置資料湖，並透過使用 query-in-place、分析和機器學習工具擷取有價值的見解，而無需移動資料。
- 透過設定與 AWS Transfer Family 列整合的上傳後處理工作流程來啟動 Lambda 函數。
- 使用 AWS Glue 開發新的微服務，以存取 Amazon S3 或全受管資料庫中的資料，這是一種無伺服器資料整合服務，可讓您輕鬆探索、準備和合併用於分析、機器學習和應用程式開發的資料。

在遷移使用案例中，因為您可以將任何資料從大型主機傳輸到 S3，因此您可以執行下列動作：

- 使用 Amazon S3 Glacier 和 S3 Glacier 深度存檔，淘汰實體基礎設施，並建立符合成本效益的資料存檔策略。

- 使用 Amazon S3 和其他 AWS 服務 (例如 S3 Glacier 和 Amazon EFS) 建置可擴展、耐用且安全的備份和還原解決方案，以增強或取代現有的現場部署功能。

更多模式

- 使用精確 Connect 將大型主機資料庫複寫到 AWS

管理與治理

主題

在未使用 AWS KMS 金鑰加密 Amazon 資料 Firehose 資源時識別並發出警示

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：管理與治理；分析；大數據；雲端原生；基礎架構；安全性、身分識別、合規性	AWS 服務：AWS CloudTrail；Amazon CloudWatch；AWS Identity and Access Management；Amazon Kinesis；AWS Lambda；Amazon SNS

Summary

為了達到合規性，某些組織必須在資料交付資源 (例如 Amazon Data Firehose) 上啟用加密功能。此模式顯示了一種在資源不符合性時監視、偵測和通知的方法。

為了維護加密需求，此模式可在 Amazon Web Services (AWS) 上使用，以提供未使用 AWS Key Management Service (AWS KMS) 金鑰加密的 Firehose 交付資源的自動監控和偵測。解決方案會傳送警示通知，並可延伸以執行自動修復。此解決方案可套用至個別帳戶或多帳戶環境，例如使用 AWS 登陸區域或 AWS Control Tower 的環境。

先決條件和限制

先決條件

- Firehose 交付串流
- AWS 有足夠的許可和熟悉度 CloudFormation，這些 AWS 用於此基礎設施自動化

限制

解決方案不是即時的，因為它使用 AWS CloudTrail 事件進行偵測，而且在建立未加密的資源和傳送通知之間存在延遲。

架構

目標技術堆疊

此解決方案使用無伺服器技術和下列服務：

- AWS CloudTrail
- Amazon CloudWatch
- AWS 命令列界面 (AWS CLI)
- AWS Identity and Access Management (IAM)
- Amazon 數據 Firehose
- AWS Lambda
- Amazon Simple Notification Service (Amazon SNS)

目標架構

1. 使用者建立或修改 Firehose。
2. 檢測到並匹配 CloudTrail 事件。
3. Lambda 被調用。
4. 識別不符合標準的資源。
5. 已傳送電子郵件通知。

自動化和規模

使用 AWS CloudFormation StackSets，您可以使用單一命令將此解決方案套用到多個 AWS 區域或帳戶。

工具

- AWS CloudTrail — AWS CloudTrail 是一項 AWS 服務，可協助您啟用 AWS 帳戶的管控、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。事件包括在 AWS 管理主控台中採取的動作、AWS 命令列界面，以及 AWS 開發套件和 API 操作。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供一系統事件 near-real-time 串流，用於描述 AWS 資源的變更。

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- IAM — AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- Amazon 數據防火軟管 — Amazon 數據 Firehose 是一種全受管服務，用於交付實時流數據。使用 Firehose，您不需要撰寫應用程式或管理資源。您可以將資料生產者設定為將資料傳送至 Firehose，它會自動將資料傳送到您指定的目的地。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。

史诗

強制執行合規性加密

任務	描述	所需技能
部署 AWS CloudFormation StackSets。	在 AWS CLI 中，使用`firehose-encryption-checker.yaml` 範本 (附加) 執行下列命令來建立堆疊集。 為參數提供有效的 Amazon SNS 主題 Amazon 資源名稱 (ARN)。部署應成功建立 CloudWatch 事件規則、Lambda 函數，以及具有必要許可的 IAM 角色，如範本中所述。 `aws cloudformation create-stack-set --stack-set-name my-stack-set --`	雲端架構師、系統管理員

任務	描述	所需技能
	`template-body file://firehose-encryption-checker.yaml`	
建立堆疊執行個體。	堆疊必須在您選擇的 AWS 區域以及一或多個帳戶中建立。 若要建立堆疊執行個體，請執行下列命令，將堆疊名稱、帳戶號碼和 Region 取代為您自己的執行個體。 `aws cloudformation create-stack-instances --stack-set-name my-stack-set --accounts 123456789012 223456789012 --regions us-east-1 us-east-2 us-west-1 us-west-2 --operation-preferences FailureToleranceCount=1`	雲端架構師、系統管理員

相關資源

- 使用 AWS CloudFormation StackSets
- 什麼是 Amazon CloudWatch 活動？

其他資訊

AWS Config 不支援 Firehose 交付串流資源類型，因此無法在解決方案中使用 AWS Config 規則。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目

創建者：巴加利（AWS）

創建者：AWS	環境：PoC 或試點	技術：雲端原生；基礎架構 DevOps；現代化；安全性、身分識別、合規性；管理與治理
工作量：Microsoft	AWS 服務：AWS Systems Manager	

Summary

AWS 系統管理器是亞馬遜彈性運算雲端 (Amazon EC2) 執行個體的遠端管理工具。Systems Manager 可讓您在 Amazon Web Services 上掌握和控制基礎設施。這個多功能工具可用來修正安全性弱點掃描報告識別為弱點的 Windows 登錄變更。

此模式涵蓋透過自動化登錄變更以確保環境安全而建議的登錄變更，以確保執行 Windows 作業系統的 EC2 執行個體安全的步驟。該模式使用運行命令來運行命令文檔。代碼已附加，其中一部分包含在「代碼」部分中。

先決條件和限制

- 有效的 AWS 帳戶
- 存取 EC2 執行個體和 Systems Manager 的權限

架構

目標技術堆疊

- 具有兩個子網路和一個網路位址轉譯 (NAT) 閘道的虛擬私有雲 (VPC)
- Systems Manager 命令文檔添加或更新註冊表名稱和值
- Systems Manager 運行命令在指定的 EC2 實例上運行命令文檔

目標架構

工具

工具

- IAM 政策和角色 — AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- Amazon 簡單存儲服務 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。此服務旨在降低開發人員進行網路規模運算的難度。在這種模式中，S3 存儲桶用於存儲 Systems Manager 日誌。
- AWS Systems Manager — AWS Systems Manager 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。Systems Manager 會掃描您的代管執行個體，並報告偵測到的任何政策違規情況 (或採取修正措施)，協助您維護安全性與合規性。
- AWS Systems Manager 命令文件 — 執行命令使用 AWS Systems Manager 命令文件。系統 Systems Manager 理員支援的所有 Linux 和 Windows 伺服器作業系統都支援大多數指令文件。
- AWS Systems Manager 執行命令 — AWS Systems Manager 執行命令可讓您從遠端安全地管理受管執行個體的組態。使用 Run Command，您可以自動執行一般管理工作，並大規模執行一次性的組態變更。

Code

您可以使用下列範例程式碼，將 Microsoft Windows 登錄名稱`Version`、登錄路徑和值新增至`2`。`HKCU:\Software\ScriptingGuys\Scripts`

```
#Windows registry path which needs to add/update
$registryPath ='HKCU:\\Software\\ScriptingGuys\\Scripts'
#Windows registry Name  which needs to add/update
$Name = 'Version'
#Windows registry value  which needs to add/update
$value = 2
# Test-Path cmdlet to see if the registry key exists.
IF(!(Test-Path $registryPath))
        {
           New-Item -Path $registryPath -Force | Out-Null
           New-ItemProperty -Path $registryPath -Name $name -Value     $value ` -
PropertyType DWORD -                 Force | Out-        Null
```

```
        } ELSE {
                        New-ItemProperty -Path $registryPath -Name $name -Value $value `
 -PropertyType            DWORD        -Force | Out-Null
            }
echo 'Registry Path:'$registryPath
 echo 'Registry Name:'$registryPath
 echo 'Registry Value:'(Get-ItemProperty -Path $registryPath -Name $Name).version
```

附上完整的 Systems Manager 命令文件 JavaScript 物件符號 (JSON) 程式碼範例。

史诗

設定 VPC

任務	描述	所需技能
建立 VPC。	在 AWS 管理主控台上，建立具有公有和私有子網路的 VPC 以及 NAT 閘道。如需詳細資訊，請參閱 AWS 文件。	雲端管理員
建立安全性群組。	確定每個安全性群組都允許從來源 IP 位址存取遠端桌面通訊協定 (RDP)。	雲端管理員

建立 IAM 政策和 IAM 角色

任務	描述	所需技能
建立 IAM 政策。	建立可讓您存取 Amazon S3、Amazon EC2 和 Systems Manager 的 IAM 政策。	雲端管理員
建立 IAM 角色。	建立 IAM 角色，並附加可讓您存取 Amazon S3、Amazon EC2 和 Systems Manager 的 IAM 政策。	雲端管理員

執行自動化

任務	描述	所需技能
建立系 Systems Manager 指令文件。	創建一個 Systems Manager 命令文檔，將部署 Microsoft Windows 註冊表更改添加或更新。	雲端管理員
運行系 Systems Manager 運行命令。	運行 Systems Manager 運行命令，選擇命令文檔和 Systems Manager 目標實例。這會將所選命令文件中的 Microsoft Windows 登錄變更推送至目標執行個體。	雲端管理員

相關資源

- AWS Systems Manager
- AWS Systems Manager 文件
- AWS Systems Manager 運行命令

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Systems Manager 維護視窗自動停止和啟動 Amazon RDS 資料庫執行個體

由阿希塔·德席爾瓦（AWS）創建

環境：生產	技術：管理與治理、成本管理、資料庫、雲端原生	AWS 服務：AWS Systems Manager；Amazon RDS

Summary

此模式示範如何使用 AWS Systems Manager 維護 Windows，按特定排程自動停止和啟動 Amazon Relational Database Service (Amazon RDS) 資料庫執行個體 (例如，在工作時間以外關閉資料庫執行個體以降低成本)。

AWS Systems Manager Automation 提供停止`AWS-StopRdsInstance`和啟動 Amazon RDS 資料庫執行個體的操作手`AWS-StartRdsInstance`冊。這表示您不需要使用 AWS Lambda 函數撰寫自訂邏輯，也不需要建立 Amazon CloudWatch 事件規則。

AWS Systems Manager 提供兩種排程任務的功能：狀態管理員和維護視窗。狀態管理器為您的 Amazon Web Services (AWS) 帳戶中的資源設定和維護所需的狀態組態一次或按特定排程。維護 Windows 會在特定時間範圍內，針對您帳戶中的資源執行工作。雖然您可以在狀態管理員或維護 Windows 中使用此模式的方法，但我們建議您使用維護 Windows，因為它可以根據指派的優先順序執行一或多個任務，也可以執行 AWS Lambda 函數和 AWS Step Functions 任務。如需狀態管理員和維護視窗的詳細資訊，請參閱 AWS Systems Manager 文件中的狀態管理員和維護視窗之間的選擇。

此模式提供詳細步驟，以設定兩個使用 cron 運算式停止並啟動 Amazon RDS 資料庫執行個體的獨立維護時段。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 您想要按特定排程停止和啟動的現有 Amazon RDS 資料庫執行個體。
- Cron 表達式為您所需的時間表。例如，*(0 9 * * 1-5)*cron 表示式會在星期一至星期五的早上 09:00 執行。

- 熟悉 Systems Manager。

限制

- Amazon RDS 資料庫執行個體一次最多可停止七天。7 天後，資料庫執行個體會自動重新啟動，以確保收到任何必要的維護更新。
- 您無法停止僅供讀取複本或具有僅供讀取複本的資料庫執行個體。
- 您無法在異地同步備份組態中停止 Amazon RDS for SQL Server 資料庫執行個體。
- 服務配額適用於維護 Windows 和 Systems Manager 自動化。如需有關服務配額的詳細資訊，請參閱 AWS 一般參考文件中的 AWS Systems Manager 端點和配額。

架構

下圖顯示自動停止和啟動 Amazon RDS 資料庫執行個體的工作流程。

工作流程包含下列步驟：

1. 建立維護時段，並使用 cron 運算式定義 Amazon RDS 資料庫執行個體的停止和開始排程。

2. 使用`AWS-StopRdsInstance`或 `AWS-StartRdsInstance` runbook 將 Systems Manager 自動化工作註冊到維護視窗。

3. 為 Amazon RDS 資料庫執行個體使用以標籤為基礎的資源群組，在維護時段註冊目標。

技術, 堆

- AWS CloudFormation
- AWS Identity and Access Management (IAM)
- Amazon RDS
- Systems Manager

自動化和規模

您可以標記必要的 Amazon RDS 資料庫執行個體、建立包含所有標記資料庫執行個體的資源群組，並將此資源群組註冊為維護時段的目標，同時停止和啟動多個 Amazon RDS 資料庫執行個體。

工具

- AWS CloudFormation 是一項可協助您建立 AWS 資源模型和設定 AWS 資源的服務。
- AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。
- Amazon Relational Database Service (Amazon RDS) 是一種 Web 服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- AWS Resource Groups 可協助您將 AWS 資源組織為群組、標記資源，以及管理、監控和自動化分組資源上的任務。
- AWS Systems Manager 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。
- AWS Systems Manager Automation 可簡化 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和其他 AWS 資源的常見維護和部署任務。
- AWS Systems Manager 維護 Windows 可協助您定義何時在執行個體上執行可能中斷性動作的排程。

史诗

建立和設定 Systems Manager 自動化的 IAM 服務角色

任務	描述	所需技能
設定系 Systems Manager 自動化的 IAM 服務角色。	登入 AWS 管理主控台，並為系統管理員自動化建立服務角色。您可以使用下列兩種方法之一來建立此服務角色： • 使用 AWS 設 CloudFormation 定 Systems Manager 自動化的服務角色 • 使用 IAM 設定 Systems Manager 自動化的角色 系 Systems Manager 自動化工作流程會使用服務角色在 Amazon RDS 資料庫執行個體	AWS 管理員

任務	描述	所需技能
	上執行啟動和停止動作來呼叫 Amazon RDS。 服務角色必須設定下列內嵌政策，該政策具有啟動和停止 Amazon RDS 資料庫執行個體的許可： `{ "Version": "2012-10-17", "Statement": [ { "Sid": "RdsStartStop", "Effect": "Allow", "Action": [ "rds:StopDBInstance", "rds:StartDBInstance" ], "Resource": "<RDS_Instance_ARN>" }, { "Sid": "RdsDescribe", "Effect": "Allow", "Action": "rds:DescribeDBInstances", "Resource": "*" } ] }`	

任務	描述	所需技能
	請務必以 <RDS_Instance_ARN> Amazon RDS 資料庫執行個體的 Amazon 資源名稱 (ARN) 取代。 重要：請確定您已記錄服務角色的 ARN。	

建立資源群組

任務	描述	所需技能
標記 Amazon RDS 資料庫執行個體。	開啟 Amazon RDS 主控台，並標記要新增至資源群組的 Amazon RDS 資料庫執行個體。標籤是指派給 AWS 資源的中繼資料，由鍵值配對組成。我們建議您使用 Action 作為標籤鍵，並將StartStop其用作值。 如需有關此項目的詳細資訊，請參閱 Amazon RDS 文件中的新增、列出和移除標籤。	AWS 管理員
為已標記的 Amazon RDS 資料庫執行個體建立資源群組。	開啟 AWS Resource Groups 主控台，並根據您為 Amazon RDS 資料庫執行個體建立的標籤建立資源群組。 在「分組準則」下，確定您選擇 AWS:: RDS:: DBInstance 做為資源類型，然後提供標籤的鍵值對 (例如，「動作-」)。StartStop這可確保服務僅檢查 Amazon RDS 資料庫執行個	AWS 管理員

任務	描述	所需技能
	體，而不會檢查具有此標籤的其他資源。 請確定您已記錄資源群組的名稱。 如需詳細資訊和詳細步驟，請參閱 AWS Resource Groups 文件中的建立以標籤為基礎的查詢和建立群組。	

設定維護時段以停止 Amazon RDS 資料庫執行個體

任務	描述	所需技能
建立維護時段。	1. 開啟 AWS Systems Manager 主控台，選擇維護視窗，然後選擇 [建立維護時段]。提供維護時段的名稱 (例如「StopRds執行處理」)，輸入說明，然後取消核取「允許未註冊的目標」。 2. 選擇 CRON/ 速率運算式並提供排程運算式，以定義何時應停止 Amazon RDS 資料庫執行個體。輸入 1 做為「持續時間」，輸入 0 表示「停止啟動工作」。依預設，時區會顯示 UTC。您可以根據 cron 運算式中定義的時間戳記，變更時區以起始維護時段。 3. 選擇 Create maintenance window (建立維護時段)。系統會返回維護時段頁面，	AWS 管理員

任務	描述	所需技能
	且維護時段的狀態為「已啟用」。 重要事項：停止資料庫執行個體的工作幾乎會在啟動時立即執行，而且不會跨越整個維護時段的持續時間。此模式提供「持續時間」和「停止啟動」任務的最小值，因為它們是維護時段的必要參數。 如需詳細資訊和詳細步驟，請參閱 AWS Systems Manager 文件中的建立維護時段 (主控台)。	
將目標指派給維護時段。	1. 在 AWS Systems Manager 主控台上，選擇維護 Windows，選擇 [動作]，然後選擇 [註冊目標]。 2. 在 [目標] 區域中，指定 [選擇資源群組]，然後選擇帳號中現有資源群組的名稱。 3. 對於資源類型，請選擇 AWS:: RDS:: 資料庫執行個體，然後選擇註冊目標。 如需詳細資訊和詳細步驟，請參閱 AWS Systems Manager 文件中的將目標指派到維護時段 (主控台)。	AWS 管理員

任務	描述	所需技能
將工作指派給維護時段。	1. 在 AWS Systems Manager 主控台上，選擇維護時段，然後選擇您的維護時段。選擇 [動作]，然後選擇 [註冊自動化工作 2. 針對「文件」，選擇 AWS StopRds 執行個體。 3. 在「目標」段落中，選擇選取註冊的目標群組，然後選擇您在目前的維護時段註冊的維護時段目標。 4. 針對「匯率」控制，指定 100% 做為「並行」與「錯誤」臨界值。 您可以根據工作並行和錯誤臨界值的需求，變更「比率」控制值。如需詳細資訊，請參閱 AWS Systems Manager 文件中的關於並行和錯誤閾值。 5. 在 IAM 服務角色區段中，對於服務角色，請將此方塊保留空白或建立您自己的自訂角色。如果您將方塊保留空白，Systems Manager 會自動建立服務連結角色，AWSServiceRoleForAmazonSSM 然後將角色與工作產生關聯。若要建立您自己的自訂角色，請參閱為維護時段建立自訂服務角色 (主控台)，然後將該自訂角色與工作產生關聯。	AWS 管理員

任務	描述	所需技能
	6. 在「輸入參數」區段中，為 Runbook 指定下列參數： • InstanceId: {{RESOURCE_ID}} • AutomationAssume角色：提供您為 Systems Manager 自動化建立之服務角色的 ARN。 • 備註：對於 InstanceId，虛擬參數是用來從 ARN 擷取 Amazon RDS 資料庫資源識別碼。若要進一步了解虛擬參數，請參閱 AWS Systems Manager 文件中的關於虛擬參數。 7. 選擇註冊自動化工作。 重要事項：「服務角色」選項會定義維護時段執行工作所需的服務角色。不過，此角色與您先前針對 Systems Manager 自動化建立的服務角色並不相同。 如需詳細資訊和詳細步驟，請參閱 AWS Systems Manager 文件中的將任務指派到維護時段 (主控台)。	

設定維護時段以啟動 Amazon RDS 資料庫執行個體

任務	描述	所需技能
設定維護時段以啟動 Amazon RDS 資料庫執行個體。	重複設定維護時段中的步驟，停止 Amazon RDS 資料庫執行個體史詩，以設定另一個維護時段，以便在排定的時間啟動 Amazon RDS 資料庫執行個體。 重要事項：設定維護時段以啟動資料庫執行個體時，必須進行下列變更： • 為維護時段使用新名稱 (例如，「StartRds執行個體」)。 • 將 cron 運算式取代為您要用來啟動資料庫執行個體的 cron 運算式。 • `AWS-StartRdsInstance` 在任務中`AWS-StopRdsInstance` 用替換工作手冊。	AWS 管理員

相關資源

- 使用 Systems Manager 自動化文件來管理執行個體並降低停工時間成本 (AWS 部落格文章)

使用 Terraform 在 AWS Organizations 中集中軟體套件分發

由普拉迪普庫馬爾潘迪伊（AWS），阿爾蒂·拉吉普特（AWS），金塔馬尼阿佛萊（AWS），T.V.R.L.L.Phani 庫馬爾達迪（AWS），馬尤里·新德（AWS）和普拉塔普庫馬爾南達（AWS）創建

環境：生產	技術：管理與治理；基礎設施	AWS 服務：AWS Organizations；AWS Systems Manager

Summary

企業通常會維護分散AWS 帳戶到多個多個元件，以便AWS 區域在工作負載之間建立強大的隔離屏障。為了保持安全性和合規性，他們的管理團隊會安裝用於安全掃描的代理程式TrendMicro工具 (例如CrowdStrikeSentinelOne、或工具)，以及 Amazon CloudWatch 代理程式、Datadog AAppDynamics gent 或代理程式進行監控。當這些團隊想要在這個大型環境中集中自動化軟體套件管理和發佈時，通常會面臨挑戰。

代理商是一項功能 AWS Systems Manager，可透過單一簡化介面，將軟體封裝及發佈到管理的 Microsoft Windows 和 Linux 執行個體，跨雲端和內部部署伺服器自動化。此模式示範如何使用 Terraform 進一步簡化管理軟體安裝的程序，以及在大量執行個體和成員帳戶中AWS Organizations以最小的努力執行指令碼。

此解決方案適用於由系統管理器管理的 Amazon、Linux 和 Windows 執行個體。

先決條件和限制

- 具有要安裝軟體的代理商套件
- 地形版本 0.15. 0 或更新版本
- 亞馬遜彈性運算雲端 (Amazon EC2) 執行個體由系統管理器管理，並具有在目標帳戶中存取亞馬遜簡單儲存服務 (Amazon S3) 的基本許可
- 使用以下方式設定組織的 landing zone AWS Control Tower
- (選擇性) 地形表單 (AFT) 的 Account Factory

架構

資源詳情

此模式會使用 Terraform 的 Account Factory (AFT) 建立所有必要的AWS資源，並使用程式碼管線來部署部署部署部署部署帳戶中的資源。代碼管道在兩個存儲庫中運行：

- 全局定制包含 Terraform 代碼，該代碼將在 AFT 註冊的所有帳戶中運行。
- 帳戶自訂包含將在部署帳戶中執行的 Terraform 程式碼。

您也可以在帳戶自訂資料夾中執行 Terraform 命令，在不使用 AFT 的情況下部署此解決方案。

地形代碼部署以下資源：

- AWS Identity and Access Management(IAM) 角色和政策
  - SystemsManager-AutomationExecutionRole 授與使用者在目標帳戶中執行自動化的權限。
  - SystemsManager-AutomationAdministrationRole 授與使用者在多個帳戶和組織單位 (OU) 中執行自動化的權限。
- 包的壓縮文件和清單 .json
  - 在 Systems Manager 中，套件包含至少一個軟體或可安裝資產的 .zip 檔案。
  - JSON 資訊清單包含指向套件程式碼檔案的指標。
- S3 儲存貯體
  - 整個組織共用的分散式套件會安全地存放在 Amazon S3 儲存貯體中。
- AWS Systems Manager文件 (SSM 文件)
  - `DistributeSoftwarePackage`包含將軟體套件散發到成員帳戶中每個目標執行個體的邏輯。
  - `AddSoftwarePackageToDistributor`包含封裝可安裝軟體資產並將其新增至「自動化」功能的AWS Systems Manager邏輯。
- Systems Manager 關聯
  - Systems Manager 關聯是用來部署解決方案。

架構和工作流程

此圖說明了下列步驟：

1. 若要從集中式帳戶執行解決方案，請將套件或軟體以及部署步驟上傳到 S3 儲存貯體。
2. 您的自訂套件會出現在 Systems Manager 主控台的 [文件] 區段的 [我擁有] 索引標籤中。

3. 狀態管理器，Systems Manager 的能力，創建，調度，並運行在整個組織的包的關聯。該關聯指定軟體套件必須先在受管理的節點上安裝並執行，然後才能將其安裝在目標節點上。
4. 此關聯會指示 Systems Manager 在目標節點上安裝套件。
5. 對於任何後續的安裝或變更，使用者可以定期或手動從單一位置執行相同的關聯，以跨帳戶執行部署。
6. 在成員帳戶中，自動化會將部署命令傳送給代理商。
7. 代理商會跨執行個體散佈軟體套件。

此解決方案使用中的管理帳戶AWS Organizations，但您也可以指定帳戶 (委派的系統管理員) 代表組織管理此帳戶。

工具

AWS 服務

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。此模式使用 Amazon S3 集中並安全地存放分散式套件。
- AWS Systems Manager協助您管理在AWS 雲端. 它可簡化應用程式和資源管理、縮短偵測和解決作業問題的時間，並協助您大規模安全地管理AWS資源。此模式使用下列 Systems Manager 功能：
  - 代理商可協助您將軟體封裝並發佈至 Systems Manager 受管理的執行個體
  - 自動化可簡化許多AWS服務的常見維護、部署和補救工作。
  - 文件會在整個組織和帳戶的系統管理員受管理執行個體上執行動作。
- AWS Organizations是一項帳戶管理服務，可協助您將多個AWS帳戶整合到您建立並集中管理的組織中。

其他工具

- Terraform 是一種基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署資源。HashiCorp

代碼存儲庫

此模式的指示和程式碼可在 GitHub 集中式套件發佈儲存庫中取得。

最佳實務

- 若要將標籤指派給關聯，請使用 AWS Command Line Interface(AWS CLI) 或AWS Tools for PowerShell。不支援使用 Systems Manager 主控台將標籤新增至關聯。如需詳細資訊，請參閱「Systems Manager」文件中的標籤 Systems Manager 資源。
- 若要使用從另一個帳戶共用的文件的新版本來執行關聯，請將文件版本設定為`default`。
- 若只要標記目標節點，請使用一個標籤鍵。如果您想要使用多個標籤鍵來鎖定節點，請使用資源群組選項。

史诗

設定來源檔案和帳號

任務	描述	所需技能
複製儲存庫。	1. 複製 GitHub 集中式套件發佈儲存庫： `git clone https://github.com/aws-samples/aws-organization-centralised-package-distribution` 2. Terraform 程式碼儲存庫需要兩個由 AFT 管理的自訂資料夾。確認存放庫的本機副本包含下列資料夾： `$ cd centralised-package-distribution` `$ ls` `global-customization account-customization`	DevOps 工程師
更新全局變量。	更新檔案中的下列輸入參`global-customizati`	DevOps 工程師

任務	描述	所需技能
	on/variables.tf 數。這些變數適用於由 AFT 建立和管理的所有帳戶。 • account_id ：要部署代理商解決方案的帳戶 ID。 • aws_region ：關聯AWS區域的部署位置。	
更新帳戶變數。	更新檔案中的下列輸入參account-customization/variables.tf 數。這些變數僅適用於由 AFT 建立和管理的特定帳戶。 • package_bucket_name ：包含套件散發檔案的 S3 儲存貯體的名稱。 • package_name ：套件發佈檔案的名稱。 • package_version ：安裝程式的套件版本。	DevOps 工程師

自訂參數和部署檔案

任務	描述	所需技能
更新狀態管理員關聯的輸入參數。	更新account-customization/association.tf 檔案中的下列輸入參數，以定義要在執行個體上維護的狀態。如果預設參數值支援您的使用案例，您可以使用預設參數值。	DevOps 工程師

任務	描述	所需技能
	• `targetAccounts` ：AWS Organizations 內的組織單位 (OU) ID，代表具有要分發之目標執行個體的帳戶。OU 識別碼以「ou」開頭。 • `targetRegions` ：目標執行個體正在執行的位置 AWS 區域 (例如，「US-東 1″ 或」ap-東部-2」)。 • `action`：指定是否要安裝或解除安裝套件。 • `installationType` ：下列其中一種安裝類型： • `uninstall` ：已解除安裝套件。 • `reinstall` ：應用程式會離線，直到重新安裝程序完成為止。 • `In-place update`：當新的或更新的文件添加到安裝時，該應用程序可用。 • `name`：要安裝或解除安裝的套件名稱。 • `version`：要安裝或解除安裝的套件版本。如果沒有安裝套件版本，系統會傳回錯誤訊息。 • `bucketName` ：套件已部署到的 S3 儲存貯體名稱。此存儲桶應僅包含軟件包和清單文件。	

任務	描述	所需技能
	• bucketPrefix ：存放套件資產的 S3 前綴。 • AutomationAssumeRole ：的 Amazon 資源名稱（ARN）。SystemsManager-AutomationAdministrationRole	
準備封裝的壓縮manifest.json 檔案和檔案。	此病毒碼提供資料夾中包含安 PowerShell 裝和解除安裝指令碼的範例可安裝檔案 (適用於 Windows 的 .msi 和 Linux 的 .rpm)。account-customization/package 1. 以您自己的檔案取代 PowerShell 可安裝的檔案，或提供可安裝的檔案、安裝和解除安裝指令碼，以及資訊清單檔案，以便在您帳戶中的account-customization 資料夾中建立套件。 2. 根據您的需求，自訂 Terraform 在account-customization 資料夾中產生的預設manifest.json 檔案。	DevOps 工程師

執行地形命令來佈建資源

任務	描述	所需技能
初始化地形組態。	若要使用 AFT 自動部署解決方案，請將程式碼推送至AWS CodeCommit： `$ git add *` `$ git commit -m "message"` `$ git push` 您也可以透過從資料夾執行 Terraform 命令，在不使用 AFT 的情況下部署此解決方案。`account-customization` 要初始化包含 Terraform 文件的工作目錄，請運行： `$ terraform init`	DevOps 工程師
預覽變更。	若要預覽 Terraform 將對基礎結構所做的變更，請執行以下命令： `$ terraform plan` 此命令會評估 Terraform 組態，以判斷已宣告之資源的所需狀態。它還將所需的狀態與實際基礎結構進行比較，以便在工作區內佈建。	DevOps 工程師
套用變更。	執行下列命令以實作您對`variables.tf` 檔案所做的變更：	DevOps 工程師

任務	描述	所需技能
	`$ terraform apply`	

驗證資源

任務	描述	所需技能
驗證 SSM 文件的建立。	1. 在 [系 Systems Manager] 主控台的左側導覽窗格中，選擇 [文件]。 2. 選擇 Owned by me (我所擁有) 索引標籤。 您應該會看到`DistributeSoftwarePackage` 和`AddSoftwarePackageToDistributor` 套件。	DevOps 工程師
驗證自動化的成功部署。	1. 在 [系 Systems Manager] 主控台的左側導覽窗格中，選擇 [自動化]。 2. 在 [自動化執行] 清單中，您應該會看到最新的`DistributeSoftwarePackage` 和`AddSoftwarePackageToDistributor` 部署。 3. 選擇「執行 ID」以驗證它們是否已成功完成。	DevOps 工程師
驗證套件是否已部署到目標成員帳戶執行個體。	1. 在 [系 Systems Manager] 主控台的瀏覽窗格中，選擇 [執行命令]。	DevOps 工程師

任務	描述	所需技能
	2. 在命令歷史記錄中，您將看到每次調用及其狀態。 3. 選擇任何命令 ID 以查看每個目標執行處理的部署歷史記錄。 4. 選擇實例 ID，然後檢查分發的「輸出」部分。	

故障診斷

問題	解決方案
狀態管理員關聯失敗或處於待處理狀態。	請參閱AWS知識中心中的疑難排解資訊。
排程的關聯無法執行。	您的排程規格可能無效。狀態管理器目前不支持在 cron 表達式中為關聯指定月份。使用 Cron 或速率運算式來確認排程。

相關資源

- 集中式套件發佈 (GitHub 儲存庫)
- 地形形式的 Account Factory (AFT)
- 使用案例和最佳做法 (AWS Systems Manager文件)

設定 VPC 流程日誌以便跨 AWS 帳戶集中管理

由本傑明·莫里斯（AWS）和阿曼考爾甘地（AWS）創建

環境：生產	技術：管理與治理	AWS 服務：Amazon VPC; Amazon S3

Summary

在 Amazon Web Services (AWS) 虛擬私有雲端 (VPC) 中，VPC 流程日誌功能可以為操作和安全疑難排解提供有用的資料。不過，在多帳戶環境中使用 VPC 流程記錄會有限制。具體來說，不支援來自 Amazon 日誌的跨帳戶流程 CloudWatch 日誌。相反地，您可以透過使用適當的儲存貯體政策設定 Amazon Simple Storage Service (Amazon S3) 儲存貯體來集中日誌。

附註：此模式討論將流程記錄傳送到集中位置的需求。但是，如果您還希望日誌在成員帳戶中可用，則可以為每個 VPC 創建多個流程日誌。無法存取記錄封存帳戶的使用者可以查看流量記錄以進行疑難排解。或者，您也可以為每個將記錄檔傳送至記錄的 VPC 設定單一流程記 CloudWatch 錄。然後，您可以使用 Amazon 資料 Firehose 訂閱篩選器將日誌轉寄到 S3 儲存貯體。如需詳細資訊，請參閱相關資源一節。

先決條件和限制

前提

- 有效的 AWS 帳戶
- 擁有用於集中化日誌 (例如日誌存檔) 的帳戶的 AWS Organizations 組織

限制

如果您使用 AWS Key Management Service (AWS KMS) 受管金鑰`aws/s3`來加密中央儲存貯體，則不會接收來自其他帳戶的日誌。相反，您會看到如下所示的錯誤。

```
"Unsuccessful": [
        {
            "Error": {
                "Code": "400",
                "Message": "LogDestination: <bucketName> is undeliverable"
```

```
            },
            "ResourceId": "vpc-1234567890123456"
        }
    ]
```

這是因為帳戶的 AWS 受管金鑰無法跨帳戶共用。

解決方案是使用 Amazon S3 受管加密 (SSE-S3) 或可與成員帳戶共用的 AWS KMS 客戶受管金鑰。

架構

目標技術堆疊

在下圖中，針對每個 VPC 部署兩個流程記錄檔。一個人將日誌發送到本地 CloudWatch 日誌組。另一個會將日誌傳送到集中式記錄帳戶中的 S3 儲存貯體。值區政策允許記錄傳遞服務將記錄檔寫入值區。

重要事項：瞭解與此解決方案所需的儲存貯體政策相關的風險。由於寫入此儲存貯體的主體是服務主體，而不是 AWS Identity and Access Management (IAM) 主體，因此該`aws:PrincipalOrgID`條件將不是有效的條件。這表示目前無法根據帳戶的上階組織來限制寫入。

若要保護值區的安全，請使用 hard-to-guess 值區名稱，並將值區名稱視為不應在組織外顯示的敏感值。請確定您在儲存貯體政策中使用的是最低權限，授予的權限不超過`s3:putObject`和`s3:GetBucketAcl`權限。如果您在具有靜態帳戶集的環境中工作，則可以使用「拒絕」效果來封鎖特定帳戶以外的存取，儘管這對於大多數組織而言在作業上並不可行。

目標架構

自動化和規模

每個 VPC 都設定為將日誌傳送到中央記錄帳戶中的 S3 儲存貯體。使用下列其中一種自動化解決方案，協助確保流程記錄已正確設定：

- AWS CloudFormation StackSets
- 適用於地形 (AFT) 的 AWS Control Tower Account Factory
- 具有修復功能的 AWS Config 規則

工具

工具

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。此病毒碼使用 VPC 流程記錄功能來擷取 VPC 中進出網路介面之 IP 流量的相關資訊。

最佳實務

使用基礎結構即程式碼 (IaC) 可大幅簡化 VPC 流程記錄的部署程序。抽象化您的 VPC 部署定義以包含流程記錄資源建構，將會自動將 VPC 與流程記錄一起部署。這將在下一節中進行演示。

集中式流程記錄

將集中式流程記錄新增至 Terraform 中的 VPC 模組的範例語法 HashiCorp

此程式碼會建立流程日誌，將日誌從 VPC 傳送到集中式 S3 儲存貯體。請注意，此模式不涵蓋 S3 存儲桶的創建。

如需建議的儲存貯體政策陳述式，請參閱其他資訊一節。

```
variable "vpc_id" {
  type        = string
  description = "ID of the VPC for which you want to create a Flow Log"
}

locals {
  # For more details: https://docs.aws.amazon.com/vpc/latest/userguide/flow-
logs.html#flow-logs-custom
  custom_log_format_v5 = "$${version} $${account-id} $${interface-id} $${srcaddr} $
${dstaddr} $${srcport} $${dstport} $${protocol} $${packets} $${bytes} $${start} $${end}
 $${action} $${log-status} $${vpc-id} $${subnet-id} $${instance-id} $${tcp-flags} $
${type} $${pkt-srcaddr} $${pkt-dstaddr} $${region} $${az-id} $${sublocation-type} $
${sublocation-id} $${pkt-src-aws-service} $${pkt-dst-aws-service} $${flow-direction} $
${traffic-path}"
}

resource "aws_flow_log" "centralized" {
  log_destination      = "arn:aws:s3:::centralized-vpc-flow-logs-
<log_archive_account_id>" # Optionally, a prefix can be added after the ARN.
```

```
  log_destination_type = "s3"
  traffic_type         = "ALL"
  vpc_id               = var.vpc_id
  log_format           = local.custom_log_format_v5 # If you want fields from VPC Flow
 Logs v3+, you will need to create a custom log format.
  tags                 = {
    Name = "centralized_flow_log"
  }
}
```

本機流程記錄

在具有所需權限的 Terraform 中將本機流程記錄新增至 VPC 模組的範例語法

此程式碼會建立流程記錄，將記錄從 VPC 傳送到本機 CloudWatch 記錄群組。

```
data "aws_region" "current" {}

variable "vpc_id" {
  type        = string
  description = "ID of the VPC for which you want to create a Flow Log"
}

resource "aws_iam_role" "local_flow_log_role" {
  name = "flow-logs-policy-${var.vpc_id
}"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
}
```

```
resource "aws_iam_role_policy" "logs_permissions" {
  name = "flow-logs-policy-${var.vpc_id}"
  role = aws_iam_role.local_flow_log_role.id

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:CreateLogDelivery",
        "logs:DeleteLogDelivery"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:${data.aws_region.current.name}:*:log-group:vpc-flow-
logs*"
    }
  ]
}
EOF
}

resource "aws_cloudwatch_log_group" "local_flow_logs" {
  # checkov:skip=CKV_AWS_338:local retention is set to 30, centralized S3 bucket can
 retain for long-term
  name              = "vpc-flow-logs/${var.vpc_id}"
  retention_in_days = 30
}

resource "aws_flow_log" "local" {
  iam_role_arn    = aws_iam_role.local_flow_log_role.arn
  log_destination = aws_cloudwatch_log_group.local_flow_logs.arn
  traffic_type    = "ALL"
  vpc_id          = var.vpc_id
  tags            = {
    Name = "local_flow_log"
  }
}
```

史诗

部署 VPC 流程記錄基礎結構

任務	描述	所需技能
決定加密策略並建立中央 S3 儲存貯體的政策。	中央儲存貯體不支援 aws/s3 AWS KMS 金鑰，因此您必須使用 SSE-S3 或 AWS KMS 客戶受管金鑰。如果您使用 AWS KMS 金鑰，金鑰政策必須允許成員帳戶使用金鑰。	合規
建立中央流程記錄值區。	建立要傳送流程記錄的中央儲存貯體，並套用您在上一步中選擇的加密策略。這應該在日誌存檔或類似用途的帳戶中。 從 [其他資訊] 區段取得儲存貯體政策，並在使用您的環境特定值更新預留位置之後，將其套用至中央儲存貯體。	一般 AWS
將 VPC 流程記錄設定為將記錄檔傳送至中央流程記錄值區。	將流程記錄新增至您要從中收集資料的每個 VPC。最可擴展的方法是使用 IaC 工具，例如 AFT 或 AWS Cloud Development Kit (AWS CDK)。例如，您可以建立一個 Terraform 模組，將 VPC 與流程記錄一起部署。如有必要，您可以手動新增流程記錄。	網路管理員
將 VPC 流程記錄設定為傳送至本機 CloudWatch 記錄。	（選擇性）如果您希望在產生記錄檔的帳戶中可見流程記錄，請建立另一個流程記錄檔以將資料傳送至本機帳戶中的 CloudWatch 記錄。或者，您	一般 AWS

任務	描述	所需技能
	可以將資料傳送到本機帳戶中帳戶特定的 S3 儲存貯體。	

相關資源

- 如何使用集中式流程記錄資料來促進資料分析並滿足安全性需求 (部落格文章)
- 如何使用 AWS Config 規則自動啟用 VPC 流程日誌 (部落格文章)

其他資訊

桶政策

新增預留位置名稱值之後，此儲存貯體政策範例可套用至流程日誌的中央 S3 儲存貯體。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<BUCKET_NAME>/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
```

```
            "Resource": "arn:aws:s3:::<BUCKET_NAME>"
        },
        {
            "Sid": "DenyUnencryptedTraffic",
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::<BUCKET_NAME>/*",
                "arn:aws:s3:::<BUCKET_NAME>"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

如果您有帳戶的靜態清單，您可以新增下列陳述式來拒絕該清單以外的任何帳戶。

```
{
    "Sid": "AccountDenyList",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "NotResource": [
        "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID1>/*",
        "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID2>/*",
        "arn:aws:s3:::<BUCKET_NAME>/<OPTIONAL_PREFIX>/AWSLogs/<ACCOUNT_ID3>/*",
    ]
}
```

作為先前 NotResource-Deny 模式的替代方法，您可以改為將條件新增至每個對帳Allow單，以指定已核准的科目。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": [
            "111111111111",
```

```
            "222222222222"
        ]
    }
}
```

添加前綴

如果您擔心儲存貯體名稱公開公開的情況下不必要的外部寫入，也可以限制儲存貯體內已知前置詞的寫入。如果您實作此操作，請更新aws_flow_log資源log_destination中的，以包含儲存貯體 Amazon 資源名稱 (ARN) 後面的前置詞。例如，下列陳述式會將寫入限制為特定前置詞。

```
{
    "Sid": "PrefixAllowList",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "NotResource": [
        "arn:aws:s3:::<BUCKET_NAME>/<PREFIX>/*"
    ]
}
```

使用 NLog 在 Amazon CloudWatch 日誌中設定 .NET 應用程式的記錄

由比布蒂·薩胡（AWS）和羅布希爾（AWS）（AWS）創建

環境：生產	技術：管理與治理 DevOps; Web 和移動應用程序	工作量：Microsoft
AWS 服務：Amazon CloudWatch 日誌		

Summary

此模式說明如何使用 NLog 開放原始碼記錄架構在 Amazon Lo CloudWatch gs 中記錄 .NET 應用程式使用情況和事件。在 CloudWatch 主控台中，您可以近乎即時地檢視應用程式的記錄訊息。您也可以設定測量結果並設定警示，以便在超過測量結果臨界值時通知您。使用 CloudWatch 應用程式見解，您可以檢視自動化或自訂儀表板，以顯示受監控應用程式的潛在問題。 CloudWatch 應用程式深入解析可協助您快速找出應用程式和基礎架構持續發生的問題。

若要將記錄訊息寫入 CloudWatch 記錄檔，請將`AWS.Logger.NLog` NuGet 套件加入至 .NET 專案。然後，您將`NLog.config`檔案更新為使用 CloudWatch 記錄檔做為目標。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- .NET Web 或主控台應用程式，可：
  - 使用支持的 .NET 框架或 .NET 核心版本。如需詳細資訊，請參閱產品版本。
  - 使用 nLog 將日誌數據發送到應用程序洞察。
- 為 AWS 服務建立 IAM 角色的許可。如需詳細資訊，請參閱服務角色權限。
- 將角色傳遞給 AWS 服務的許可。如需詳細資訊，請參閱授予使用者將角色傳遞至 AWS 服務。

產品版本

- .NET 架構 3.5 版或更新版本
- . NET 核心版本 1.0.1、2.0.0 或更新版本

架構

目標技術堆疊

- nLog
- Amazon CloudWatch 日誌

目標架構

1. .NET 應用程序將日誌數據寫入 NLog 日誌框架。
2. nLog 會將記錄檔資料寫入 CloudWatch 防護記錄。
3. 您可以使用 CloudWatch 警示和自訂儀表板來監視 .NET 應用程式。

工具

AWS 服務

- Amazon CloudWatch 應用程式深入解析可協助您觀察應用程式和基礎 AWS 資源的運作狀態。
- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- 的 AWS 工具 PowerShell是一 PowerShell 組模組，可協助您從命令列對 AWS 資源執行操作 PowerShell 指令碼。

其他工具

- 日誌 .nlog 是一個 NLog 目標，用於將日誌數據記錄到日誌中。 CloudWatch
- nLog 是 .NET 平台的開放原始碼記錄架構，可協助您將記錄資料寫入目標，例如資料庫、記錄檔或主控台。

- PowerShell是一個 Microsoft 的自動化和配置管理程序，可以在 Windows，Linux 和 macOS 上運行。
- Visual Studio 是整合式開發環境 (IDE)，其中包含編譯器、程式碼完成工具、圖形設計工具，以及其他支援軟體開發的功能。

最佳實務

- 設定目標記錄群組的保留原則。這必須在 NLog 配置之外完成。根據預設，記錄資料會無限期地儲存在 CloudWatch 防護記錄中。
- 遵守管理 AWS 存取金鑰的最佳實務。

史诗

設定存取權和工具

任務	描述	所需技能
建立 IAM 政策。	請遵循 IAM 說明文件中使用 JSON 編輯器建立政策中的指示。輸入下列 JSON 原則，此原則具有允許 CloudWatch 記錄讀取和寫入記錄檔所需的最低權限。 { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["logs:CreateLogGroup", "logs:CreateLogStream",	AWS 管理員 DevOps

任務	描述	所需技能
	"logs:GetLogEvents", "logs:PutLogEvents", "logs:DescribeLogGroups", "logs:DescribeLogStreams", "logs:PutRetentionPolicy"], "Resource": ["*"] }] }	
建立 IAM 角色。	遵循 IAM 文件中建立角色以將許可委派給 AWS 服務中的說明進行操作。選取您先前建立的策略。這是記錄 CloudWatch 檔假設執行記錄動作的角色。	AWS 管理員 DevOps

任務	描述	所需技能
為. 設定的 AWS 工具 PowerShell。	1. 請按照安裝 AWS 工具中針對您的作業系統的說明進行操作 PowerShell。 2. 使用適用於 PowerShell 指令程式的 AWS 工具，將存取金鑰和秘密金鑰存放在設定檔中。如需指示，請參閱 AWS 工具中的管理設定檔以取得 PowerShell 文件。	一般 AWS

設定 nLog

任務	描述	所需技能
安裝 NuGet 套件。	1. 在 Visual Studio 中，選擇 [檔案]，然後選擇 [開啟專案或解決方案]。 2. 選擇要在其中安裝 NLog 的項目。 3. 在 Visual Studio 中，選擇工具、P NuGet ackage 管理員、P ackage 管理員主控台。 4. 輸入以下指令來安裝 `AWS.Logger.NLog` NuGet 套件。 `Install-Package AWS.Logger.NLog -Version 3.1.0`	應用程式開發人員

任務	描述	所需技能
設定記錄目標。	1. 開啟 NLog.config 檔案。 2. 對於目標type，輸入AWSTarget 。 3. 針對目標logGroup，輸入您要使用的記錄群組名稱。如果記錄群組尚不存在，則會自動建立具有提供名稱的新記錄群組。 4. 針對目標region，輸入設定 CloudWatch 日誌的 AWS 區域。 5. 針對目標profile，輸入您先前建立的設定檔名稱來儲存存取金鑰和秘密金鑰。 6. 儲存並關閉 NLog.config 檔案。 如需範例組態檔案，請參閱此模式的其他資訊一節。當您運行應用程序時，NLog 將寫入日誌消息並將其發送到 CloudWatch 日誌。	應用程式開發人員

驗證和監控記錄

任務	描述	所需技能
驗證記錄。	依照記錄檔文件中檢視傳送至 CloudWatch 記錄檔的記 CloudWatch 錄檔資料中的指示進行。驗證是否正在記錄 .NET 應用程式的記錄事件	一般 AWS

任務	描述	所需技能
	。如果未記錄日誌事件，請參閱此病毒碼中的疑難排解一節。	
監視 .NET 應用程式堆疊。	根據您的使用案例的需要 CloudWatch ，在中設定監視。您可以使用CloudWatch 日誌深入解析、CloudWatch 指標深入解析和CloudWatch 應用程式洞察來監視您的 .NET 工作負載。您也可以設定警示，以便接收警示，並且可以建立自訂儀表板以從單一檢視監視工作負載。	一般 AWS

故障診斷

問題	解決方案
記錄檔資料不會顯示在 CloudWatch 記錄中。	確保 IAM 政策已附加到 CloudWatch 日誌假定的 IAM 角色。如需指示，請參閱「Epics」一節中的「設定存取權限和工具」一節。

相關資源

- 使用記錄群組和記錄串流 (CloudWatch 記錄檔文件)
- Amazon CloudWatch 日誌和 .NET 日誌框架 (AWS 部落格文章)

其他資訊

以下是範例NLog.config檔案。

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
  <configSections>
    <section name="nlog" type="NLog.Config.ConfigSectionHandler, NLog" />
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <nlog>
    <extensions>
      <add assembly="NLog.AWS.Logger" />
    </extensions>
    <targets>
      <target name="aws" type="AWSTarget" logGroup="NLog.TestGroup" region="us-east-1"
 profile="demo"/>
    </targets>
    <rules>
      <logger name="*" minlevel="Info" writeTo="aws" />
    </rules>
  </nlog>
</configuration>
```

跨不同 AWS 帳戶和 AWS 區域複製 AWS Service Catalog 產品

由薩欽維格（AWS）和桑托什羽衣甘藍（AWS）創建

環境：生產	技術：管理與治理；無伺服器	工作負載：所有其他工作
AWS 服務：AWS Service Catalog；AWS Lambda		

Summary

AWS Service Catalog 是一項區域服務，這表示 AWS Service Catalog 產品組合和產品只能在建立這些產品組合和產品的 AWS 區域中顯示。如果您在新區域中設定 AWS Service Catalog 中樞，則必須重新建立現有產品，這可能是一個耗時的程序。

此模式的方法說明如何將產品從來源 AWS 帳戶或區域中的 AWS Service Catalog 中樞複製到目的地帳戶或區域中的新中樞，以協助簡化此程序。如需 AWS 服務目錄中樞和支點模型的詳細資訊，請參閱 AWS Service Catalog 中樞和支點模型：如何在 AWS 管理和控管部落格上自動化 AWS Service Catalog 對多個帳戶的部署和管理。

該模式也提供跨帳戶或其他區域複製 AWS Service Catalog 產品所需的個別程式碼套件。透過使用此模式，您的組織可以節省時間、在新的 AWS Service Catalog 中樞提供現有和先前的產品版本、將手動錯誤的風險降到最低，並在多個帳戶或區域擴展該方法。

注意：此圖案的「史詩」區段提供兩個複製產品的選項。您可以使用選項 1 跨帳戶複製產品，或選擇選項 2 以跨區域複製產品。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 來源帳戶或區域中的現有 AWS Service Catalog 產品。
- 目的地帳戶或區域中的現有 AWS 服務目錄中心。
- 如果您想要跨帳戶複製產品，則必須共用包含產品的 AWS Service Catalog 產品組合，然後將其匯入目的地帳戶。如需詳細資訊，請參閱 AWS 服務目錄文件中的共用和匯入產品組合。

限制

- 您想要跨區域或帳戶複製的 AWS Service Catalog 產品不能屬於多個產品組合。

架構

下圖顯示將 AWS Service Catalog 產品從來源帳戶複製到目的地帳戶。

下圖顯示將 AWS Service Catalog 產品從來源區域複製到目的地區域。

技術, 堆

- Amazon CloudWatch
- AWS Identity and Access Management (IAM)
- AWS Lambda
- AWS Service Catalog

自動化和規模

您可以使用 Lambda 函數擴展此模式的方法，該函數可根據收到的請求數量或需要複製的 AWS Service Catalog 產品數量進行擴展。如需這方面的詳細資訊，請參閱 AWS Lambda 文件中的 Lambda 函數擴展。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Service Catalog 可協助您集中管理 AWS 核准的 IT 服務目錄。最終使用者可在機構所設的限制範圍內，迅速地只部署自己需要且經核准的 IT 服務。

Code

您可以使用 `cross-account-copy`套件 (附加) 跨帳戶複製 AWS Service Catalog 產品，或使用`cross-region-copy`套件 (附加) 複製跨區域的產品。

此`cross-account-copy`套件包含下列檔案：

- `copyconf.properties`— 包含跨帳戶複製產品的區域和 AWS 帳戶 ID 參數的組態檔。
- `scProductCopyLambda.py`— 用於跨帳戶複製產品的 Python 函數。
- `createDestAccountRole.sh`— 在目標帳戶中建立 IAM 角色的指令碼。
- `createSrcAccountRole.sh`— 在來源帳戶中建立 IAM 角色的指令碼。
- `copyProduct.sh`— 建立和叫用 Lambda 函數以跨帳戶複製產品的指令碼。

此`cross-region-copy`套件包含下列檔案：

- `copyconf.properties`— 包含跨區域複製產品的區域和 AWS 帳戶 ID 參數的組態檔。
- `scProductCopyLambda.py`— 用於跨區域複製產品的 Python 函數。
- `copyProduct.sh`— 用於建立 IAM 角色以及建立和叫用 Lambda 函數以跨區域複製產品的指令碼。

史诗

選項 1 — 跨帳戶複製 AWS Service Catalog 產品

任務	描述	所需技能
更新組態檔案。	1. 將`cross-account-copy` 套件（附件）下載到您的本機電腦。 2. 使用下列值更新`copyconf.properties` 組態檔案： • `srcRegion` — 提供包含產品的來源地區。 • `destRegion` — 提供產品的目的地區域。	AWS 管理員、AWS 系統管理員、雲端管理員

任務	描述	所需技能
	• `sourceAccountId` — 為您的來源帳戶提供 AWS 帳戶 ID。 • `destAccountId` — 提供目的地帳戶的 AWS 帳戶 ID。	
在目的地帳戶中設定 AWS CLI 的登入資料。	透過執行`aws configure` 命令並提供下列值，設定您的登入資料以存取目的地帳戶中的 AWS CLI： `$aws configure` `AWS Access Key ID [None]: <your_access_key_id>` `AWS Secret Access Key [None]: <your_secret_access_key>` `Default region name [None]: Region` `Default output format [None]:` 如需這方面的詳細資訊，請參閱 AWS 命令列界面文件中的組態基礎知識。	AWS 管理員、AWS 系統管理員、雲端管理員

任務	描述	所需技能
在來源帳戶中設定 AWS CLI 的登入資料。	透過執行`aws configure` 命令並提供下列值，設定您的登入資料以存取來源帳戶中的 AWS CLI： `$aws configure` `AWS Access Key ID` `[None]: <your_access_key_id>` `AWS Secret Access Key` `[None]: <your_secret_access_key>` `Default region name` `[None]: Region` `Default output format` `[None]:` 如需這方面的詳細資訊，請參閱 AWS 命令列界面文件中的組態基礎知識。	AWS 管理員、AWS 系統管理員、雲端管理員
在目的地帳戶中建立 Lambda 執行角色。	在您的目標帳戶中執行`createDestAccountRole.sh` 指令碼。此指令碼會實作下列動作： • 在目的地帳戶中建立 Lambda 執行角色 • 建立並附加 Lambda 執行角色的 IAM 政策	AWS 管理員、AWS 系統管理員、雲端管理員

任務	描述	所需技能
在來源帳戶中建立跨帳戶 IAM 角色。	在您的來源帳戶中執行`createSrcAccountRole.sh` 指令碼。此指令碼會實作下列動作： • 在來源帳戶中建立跨帳戶 IAM 角色，該角色由目標帳戶中的 Lambda 執行角色假設，以複製產品 • 為來源帳戶中的跨帳戶角色建立並附加 IAM 政策	AWS 管理員、AWS 系統管理員、雲端管理員
在目標帳戶中執行複製產品指令碼。	在您的目標帳戶中執行`copyProduct.sh` 指令碼。此指令碼會實作下列動作： • 建立並叫用 Lambda 函數，將產品從來源帳戶複製到目的地帳戶	AWS 管理員、AWS 系統管理員、雲端管理員

選項 2 — 將 AWS Service Catalog 產品從來源區域複製到目的地區域

任務	描述	所需技能
更新組態檔案。	1. 將`cross-region-copy`套件（附件）下載到您的本機電腦。 2. 使用下列值更新`copyconf.properties` 組態檔案： • `srcRegion` — 提供包含產品的來源地區。 • `destRegion` — 提供產品的目的地區域。	AWS 系統管理員、雲端管理員、AWS 管理員

任務	描述	所需技能
	• accountId — 提供您的 AWS 帳戶 ID。	
為 AWS CLI 設定您的登入資料。	透過執行aws configure 命令並提供下列值，設定您的登入資料以存取環境中的 AWS CLI： `$aws configure` `AWS Access Key ID` `[None]: <your_access_key_id>` `AWS Secret Access Key` `[None]: <your_secret_access_key>` `Default region name` `[None]: Region` `Default output format` `[None]:` 如需這方面的詳細資訊，請參閱 AWS 命令列界面文件中的組態基礎知識。	AWS 管理員、AWS 系統管理員、雲端管理員
執行複製產品指令碼。	在您的目的地區域中執行copyProduct.sh 指令碼。此指令碼會實作下列動作： • 建立 Lambda 執行角色 • 建立並附加 Lambda 執行角色的 IAM 政策 • 建立並呼叫 Lambda 函數，將產品從來源區域複製到目的地區域	AWS 管理員、AWS 系統管理員、雲端管理員

相關資源

- 建立 Lambda 執行角色 (AWS Lambda 文件)
- 建立 Lambda 函數 (AWS Lambda 文件)
- AWS Service Catalog API 參考
- AWS Service Catalog 文件

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon CloudWatch 異常偵測為自訂指標建立警示

由拉姆·坎達斯瓦米（AWS）和拉希姆吉瓦尼（AWS）創建

環境：生產	技術：管理與治理；營運 DevOps；雲端原生	AWS 服務：Amazon CloudWatch

Summary

在 Amazon Web Services (AWS) 雲端上，您可以使用 Amazon 建立警示 CloudWatch 來監控指標並傳送通知，或在超出閾值時自動進行變更。

為了避免受到靜態臨界值的限制，您可以根據過去的模式建立警示，並在特定測量結果在正常作業時段之外時通知您。例如，您可以從 Amazon API Gateway 監控 API 的回應時間，並接收有關異常情況的通知，以防止您達成服務等級協議 (SLA)。

此模式說明如何針對自訂量度使用 CloudWatch 異常偵測。該模式向您展示如何在 Amazon CloudWatch 日誌洞見中建立自訂指標，或使用 AWS Lambda 函數發佈自訂指標，然後使用 Amazon 簡單通知服務 (Amazon SNS) 設定異常偵測並建立通知。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 設定為傳送電子郵件通知的現有 SNS 主題。如需這方面的詳細資訊，請參閱 Amazon SNS 文件中的開始使用 Amazon SNS。
- 使用CloudWatch 記錄設定的現有應用程式。

限制

- CloudWatch 度量不支援毫秒的時間間隔。如需有關一般和自訂指標精細度的詳細資訊，請參閱 Amazon CloudWatch 常見問答集。

架構

該圖顯示以下工作流程：

1. 使用由記錄建立和更新之指標的 CloudWatch 記錄會串流至 CloudWatch。
2. 警示會根據閾值啟動，並將警示傳送至 SNS 主題。
3. Amazon SNS 會傳送電子郵件通知給您。

技術, 堆

- 雲觀察
- AWS Lambda
- Amazon SNS

工具

- Amazon Cloudwatch — CloudWatch 提供可靠、可擴展且靈活的監控解決方案。
- AWS Lambda — L ambda 是一種運算服務，可協助您執行程式碼，而無需佈建或管理伺服器。
- Amazon SNS — 亞馬遜簡單通知服務 (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者的訊息傳遞。

史诗

設定自訂量度的異常偵測

任務	描述	所需技能
選項 1-使用 Lambda 函數建立自訂指標。	下載`lambda_function.py`檔案 (附加)，然後取代 AWS 文`lambda_function.py`件中aws-lambda-developer-guide儲存庫中的範例檔案 GitHub。這提供了一個範例 Lambda 函數，可將自訂	DevOps 工程師, AWS DevOps

任務	描述	所需技能
	指標傳送至 CloudWatch 日誌。Lambda 函數使用 Boto3 API 與之整合。 CloudWatch 執行 Lambda 函數之後，您可以登入 AWS 管理主控台、開啟主控 CloudWatch 台，並在已發佈的命名空間下提供已發佈的指標。	
選項 2 — 從 CloudWatch 日誌組創建自定義指標。	登入 AWS 管理主控台，開啟主 CloudWatch 控台，然後選擇日誌群組。選擇您要為其建立測量結果的日誌群組。 選擇「動作」，然後選擇「建立量度篩選」 在「篩選模式」中，輸入您要使用的濾鏡模式。如需詳細資訊，請參閱 CloudWatch 文件中的篩選器和模式語法。 若要測試篩選器模式，請在「測試模式」下輸入一或多個記錄事件。每個日誌事件都必須在一行內，因為 Log event messages (日誌事件訊息) 方塊中使用換行來分隔日誌事件。測試模式之後，您可以在「度量詳細資料」下輸入指標的名稱和值。 如需建立自訂指標的詳細資訊和步驟，請參閱 CloudWatch 文件中的為記錄群組建立指標篩選器。	DevOps 工程師, AWS DevOps

任務	描述	所需技能
為您的自定義指標創建警報。	在 CloudWatch 主控台上，選擇 [警示]，然後選擇 [建立鬧鐘]。選擇選取測量結果，然後在搜尋方塊中輸入您先前建立的測量結果名稱。選擇圖形指標索引標籤，並根據您的需求設定選項。 在「條件」下，選擇「異常偵測」而非「靜態閾值」。這會根據兩個標準預設偏差向您展示區帶。您可以設定閾值並根據您的需求進行調整。 選擇下一步。 注意：頻帶是動態的，取決於資料點的品質。當您開始彙總更多資料時，區帶和臨界值會自動更新。	DevOps 工程師, AWS DevOps

任務	描述	所需技能
設定 SNS 通知。	在 [通知] 下，選擇警示處於狀態、OK狀態或ALARMINSUFFICIENT_DATA 狀態時要通知的 SNS 主題。 若要讓警示針對相同的警示狀態或不同警示狀態傳送多個通知，請選擇 Add notification (新增通知)。選擇下一步。輸入警示的名稱與說明。名稱只能包含 ASCII 字元。然後選擇下一步。 在 [預覽並建立] 底下，確認資訊和條件正確無誤，然後選擇 [建立鬧鐘]。	DevOps 工程師, AWS DevOps

相關資源

- 將自訂量度發佈至 CloudWatch
- 使用 CloudWatch 異常偵測
- 警報事件和 Amazon EventBridge
- 將自訂指標推送至 Cloud Watch 時，應遵循哪些最佳做法？ (影片)
- CloudWatch 應用程式洞察簡介 (影片)
- 使用 CloudWatch (影片) 偵測異常

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

記錄您的 AWS landing zone 設計

創建者：邁克爾·戴赫內特（AWS），弗洛里安蘭格（AWS）和邁克爾·洛德曼（AWS）

環境：生產	技術：管理與治理；基礎設施；安全性、身分識別、合規性	AWS 服務：AWS Control Tower

Summary

landing zone 是一個架構良好的多帳戶環境，基於安全性和合規性最佳做法。它是整個企業的容器，可容納您所有的組織單位 (OU) AWS 帳戶、使用者和其他資源。landing zone 可以擴展以滿足任何規模的企業的需求。 AWS 有兩個選項可用於建立您的 landing zone：使用的服務型 landing zone AWS Control Tower或您建立的自訂 landing zone。每個選項都需要不同程度的 AWS 知識。

AWS 創建 AWS Control Tower 通過自動化 landing zone 的設置來幫助您節省時間。 AWS Control Tower 由管理 AWS 並使用最佳做法和準則來協助您建立基礎環境。 AWS Control Tower 使用整合式服務，例如AWS Service Catalog和 AWS Organizations，在您的 landing zone 佈建帳戶，並管理這些帳戶的存取權限。

AWS landing zone 專案的需求、實作詳細資料和營運行動項目有所不同。有需要與每個 landing zone 實施來處理的自定義方面。這包括 (但不限於) 如何處理存取管理、使用哪種技術堆疊，以及對於卓越營運的監控要求。此樣式提供可協助您記錄 landing zone 域專案的範本。透過使用範本，您可以更快地記錄專案，並協助開發和營運團隊瞭解您的 landing zone。

先決條件和限制

限制

這種模式不會描述什麼是 landing zone 或如何實施一個。如需這些主題的詳細資訊，請參閱相關資源一節。

史诗

建立設計文件

任務	描述	所需技能
識別關鍵利益相關者	識別連結到您的 landing zone 的關鍵服務和團隊經理。	專案經理
自訂範本。	在「附件」區段中下載範本，然後依照下列步驟更新範本： 1. 移除任何不適用於組織登陸區域或程序的區段。 2. 新增組織唯一的任何區段。	專案經理
完成範本。	在與利益相關者會面或使用 write-and-review 流程時，完成範本，如下所示： 1. 使用藍色方塊中的指引和資訊來完成每個區段。 2. 使用組織的自訂值取代或移除任何黃色欄位。 3. 使用您的自訂架構或流程圖取代或移除任何影像欄位。 4. 完成範本的修訂歷史記錄和貢獻者部分。	專案經理
共用設計文件。	完成 landing zone 設計文件後，請將其儲存在共用儲存庫或中央位置，供所有利益相關者存取。建議您使用標準文件控制程序來記錄並核准設計文件的修訂版本。	專案經理

相關資源

- AWS Control Tower 文件
  - 規劃您的 AWS Control Tower landing zone
  - AWS 為您的 AWS Control Tower landing zone 提供多帳戶策略
  - landing zone 域設置的管理秘訣
  - 對 landing zone 配置的期望
- AWS Control Tower(AWS 解決方案程式庫) 的自訂
- 設定安全且可擴充的多帳戶 AWS 環境 (AWS 規範指引)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在多區域、多帳戶組織中設定 AWS CloudFormation 漂移偵測

環境：生產	技術：管理與治理、雲端原生、基礎架構、營運、現代化	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS Config；AWS Lambda；AWS CloudFormation		

Summary

Amazon Web Services (AWS) 上的客戶通常都在尋找一種有效的方法來偵測資源組態不相符，包括 AWS CloudFormation 堆疊中的漂移，並儘快修復它們。使用 AWS Control Tower 或 AWS 登陸區域解決方案時，尤其是這種情況。

此模式提供了規範性的解決方案，藉由使用合併的資源組態變更並採取這些變更來產生結果，有效地解決問題。此解決方案是針對在多個區域或多個帳戶或兩者的組合中建立多個 CloudFormation 堆疊的案例而設計。該解決方案的目標如下：

- 簡化漂移檢測過程
- 設定通知和警示
- 設定合併報告

先決條件和限制

先決條件

- 在必須監控的所有區域和帳戶中啟用 AWS Config

限制

- 產生的報告僅支援 .csv 或 .json 輸出格式。

架構

目標技術堆疊

目前的指引將協助組織使用下列服務的組合來達成目標：

- AWS Config 規則
- Amazon CloudWatch 法則
- AWS Identity and Access Management (IAM)
- AWS Lambda
- Amazon Simple Notification Service (Amazon SNS)

1. AWS 組態規則可偵測漂移。
2. 其他帳戶中的漂移偵測結果會傳送至管理帳戶。
3. 該 CloudWatch 規則會呼叫 Lambda。
4. Lambda 會查詢 AWS Config 規則以取得彙總結果。
5. Lambda 通知 Amazon SNS，該 SNS 會發送有關漂移的電子郵件通知。

自動化和規模

此處介紹的解決方案可以針對其他區域和帳戶進行擴展。

工具

AWS 組態 — AWS Config 提供 AWS 帳戶中 AWS 資源組態的詳細檢視。這包含資源彼此之間的關係和之前的組態方式，所以您可以看到一段時間中組態和關係的變化。使用 AWS Config，您可以評估、稽核和評估 AWS 資源的組態。

Amazon CloudWatch — Amazon 即時 CloudWatch 監控您的 AWS 資源和您在 AWS 上執行的應用程式。您可以用 CloudWatch 來收集和追蹤指標，這些指標是您可以針對資源和應用程式測量的變數。

AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。

Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。

史诗

自動偵測漂移 CloudFormation

任務	描述	所需技能
建立彙總器。	在 AWS Config 主控台上，在管理帳戶中建立彙總工具。確定已開啟資料複寫，以便 AWS Config 可以從來源帳戶擷取資料。此外，請選取所有適用的地區和帳戶。您可以根據組織選取帳戶。這是建議使用的方法，因為組織中的新帳戶會自動成為彙總器的一部分。	雲端架構師
建立 AWS 受管規則。	新增 `cloudformation-stack-drift-detection-check` AWS 受管規則。規則需要一個參數值：`cloudformationArn` 。輸入具有偵測堆疊漂移許可的 IAM 角色 Amazon 資源名稱 (ARN)。此外，該角色必須具有可讓 AWS Config 擔任該角色的信任政策。	雲端架構師
建立彙總器的進階查詢區段。	若要從多個來源擷取漂移的堆疊，請建立下列查詢： `SELECT resourceId, configuration.driftInformation.stackDriftStatus WHERE resourceType =`	雲架構師、開發人員

任務	描述	所需技能
	`'AWS::CloudFormation::Stack' AND configuration.driftInformation.stackDriftStatus IN ('DRIFTED')`	
自動執行查詢並發佈。	使用所附加的程式碼建立 Lambda 函數。Lambda 會將結果發佈到 Amazon SNS 主題，該主題在 Lambda 函數中以環境變數形式提供。此外，若要接收警示，請建立現有 Amazon SNS 主題的電子郵件訂閱。	雲架構師、開發人員
建立 CloudWatch 規則。	建立以排程為基礎的 CloudWatch 規則，以呼叫 Lambda 函數，此函數負責警示。	雲端架構師

相關資源

資源

- 什麼是 AWS Config？
- 概念：多帳戶多區域資料彙總
- 多帳戶多區域資料彙總
- 偵測堆疊和資源的未受管理組態變更
- IAM：將 IAM 角色傳遞給特定的 AWS 服務
- 什麼是 Amazon SNS？

其他資訊

考量

使用涉及特定時間間隔的 API 呼叫的自訂解決方案，在每個 CloudFormation 堆疊或堆疊集上啟動漂移偵測並不是最佳選擇。它會導致大量 API 調用並影響性能。由於 API 呼叫的數目，可能會發生節流。另一個潛在問題是，如果僅根據排程識別資源變更，則偵測延遲。

常見問答集

問：是否應該搭配 AWS 著陸區使用附加元件型解決方案？

答：由於 AWS Config 中提供進階查詢功能以及彙總器，建議您使用 AWS Config 而不是附加元件。

問：此解決方案如何解決 CloudFormation StackSets？

答：由於堆棧集是由堆棧組成的，因此您可以使用此解決方案。堆疊執行個體詳細資料也可做為解決方案的一部分使用。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能

由拉胡爾·蓋克瓦德博士（AWS）創建

代碼存儲庫：Amazon DevOps 大師示例代碼	環境：PoC 或試點	技術：管理與治理；雲端原生；營運 DevOps；安全性、身分識別、合規性；無伺服器
AWS 服務：Amazon API Gateway；AWS CDK；Amazon DevOps 大師；Amazon DynamoDB；AWS Organizations		

Summary

此模式示範了在 TypeScript中使用 AWS Cloud Development Kit (AWS CDK) 在多個 Amazon Web Services (AWS) 區域、帳戶和組織單位 (OU) 之間啟用 Amazon DevOps Guru 服務的步驟。您可以使用 AWS CDK 堆疊 CloudFormation StackSets 從管理員 (主要) AWS 帳戶部署 AWS，以跨多個帳戶啟用 Amazon DevOps Guru，而不必登入每個帳戶，並為每個帳戶個別啟用 DevOps Guru。

Amazon DevOps Guru 提供人工智慧操作 (AIOps) 功能，協助您提高應用程式的可用性並更快地解決操作問題。 DevOps Guru 通過應用機器學習（ML）驅動的建議來減少您的手動工作量，而無需任何 ML 專業知識。 DevOps Guru 分析您的資源和運營數據。如果偵測到任何異常，就會提供量度、事件和建議來協助您解決問題。

此模式說明啟用 Amazon DevOps 大師的三個部署選項：

- 適用於跨多個帳戶和區域的所有堆疊資源
- 適用於 OU 中的所有堆疊資源
- 適用於跨多個帳戶和區域的特定堆疊資源

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI。)
- 已安裝和設定的 AWS CDK 工具組。請參閱 AWS CDK 文件中的 AWS CDK 工具組。)
- 節點 Package 管理員 (npm)，已在 TypeScript.（請參閱 npm 文檔中的下載和安裝 Node.js 和 npm。）
- 已安裝並設定 Python3，用於執行 Python 指令碼，將流量插入範例無伺服器應用程式。（請參閱 Python 文檔中的 Python 設置和使用方法。）
- 點子，安裝和配置為安裝 Python 請求庫。（請參閱 PyPI 網站上的 pip 安裝說明。）

產品版本

- AWS CDK 工具組 1.107.0 版或更新版本
- 故宮版本 7.9.0 或更新版本
- Node.js 版本 15.3.0 或更新版本

架構

技術

此模式的架構包括下列服務：

- Amazon DevOps 大師
- AWS CloudFormation
- Amazon API Gateway
- AWS Lambda
- Amazon DynamoDB
- Amazon CloudWatch
- AWS CloudTrail

AWS CDK 堆疊

該模式使用下列 AWS CDK 堆疊：

- `CdkStackSetAdminRole`— 建立 AWS Identity 和存取管理 (IAM) 管理員角色，以建立管理員和目標帳戶之間的信任關係。
- `CdkStackSetExecRole`— 建立 IAM 角色以信任管理員帳戶。
- `CdkDevopsGuruStackMultiAccReg`— 在多個 AWS 區域和帳戶中啟用所有堆疊的 DevOps 專家，並設定亞馬遜簡單通知服務 (Amazon SNS) 通知。
- `CdkDevopsGuruStackMultiAccRegSpecStacks`— 啟用跨多個 AWS 區域和帳戶的專 DevOps 家來處理特定堆疊，並設定 Amazon SNS 通知。
- `CdkDevopsguruStackOrgUnit`— 跨 O DevOps U 啟用大師，並設定 Amazon SNS 通知。
- `CdkInfrastructureStack`— 在管理員帳戶中部署範例無伺服器應用程式元件，例如 API Gateway、Lambda 和 DynamoDB，以示範故障注入和深入解析產生。

應用程式架構範

下圖說明已在多個帳戶和區域中部署的範例無伺服器應用程式的架構。該模式使用管理員帳戶部署所有 AWS CDK 堆疊。它還使用管理員帳戶作為設置 DevOps Guru 的目標帳戶之一。

1. 啟用 DevOps Guru 後，它會首先基準每個資源的行為，然後從供應指標中擷取操作 CloudWatch 資料。
2. 如果偵測到異常，就會將其與來自的事件建立關聯 CloudTrail，並產生深入分析。
3. 洞察力提供了一系列相關的事件以及規定的建議，以使操作員能夠識別罪魁禍首資源。
4. Amazon SNS 會將通知訊息傳送給操作員。

自動化和規模

此模式提供的GitHub 存放庫使用 AWS CDK 做為基礎設施即程式碼 (IaC) 工具來建立此架構的組態。AWS CDK 可協助您協調資源，並啟用跨多個 AWS 帳戶、區域和 OU 的 DevOps 大師。

工具

AWS 服務

- AWS CDK — AWS Cloud Development Kit (AWS CDK) 可協助您使用以下五種支援的程式設計語言之一，將雲端基礎設施定義為程式碼： TypeScript JavaScript、Python、Java 和 C#。

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種統一的工具，可為與 AWS 服務和資源互動提供一致的命令列界面。

Code

此模式的原始程式碼可在 GitHub Amazon DevOps Guru CDK 範例儲存庫中取得。AWS CDK 程式碼是用來撰寫的 TypeScript。若要複製並使用儲存庫，請遵循下一節中的指示。

重要：此模式中的一些故事包括針對 Unix、Linux 和 macOS 格式化的 AWS CDK 和 AWS CLI 命令範例。對於 Windows，請使用脫字符號 (^) 取代每一行結尾的反斜線 (\) 接續字元。

史诗

準備 AWS 資源以進行部署

任務	描述	所需技能
設定 AWS 命名的設定檔。	如下所示設定 AWS 命名的設定檔，以在多帳戶環境中部署堆疊。 對於管理員帳戶： `$aws configure --profile administrator` `AWS Access Key ID [****]: <your-administrator-access-key-ID>` `AWS Secret Access Key [****]: <your-administrator-secret-access-key>` `Default region name [None]: <your-administrator-region>` `Default output format [None]: json` 對於目標帳戶：	DevOps 工程師

任務	描述	所需技能
	`` $aws configure --profile target AWS Access Key ID [****: <your-target- access-key-ID> AWS Secret Access Key [****]: <your-target- secret-access-key> Default region name [None]: <your-target- region> Default output format [None]: json ``` 如需詳細資訊，請參閱 AWS CLI 文件中的使用命名設定檔。	
驗證 AWS 設定檔組態。	(選用) 您可以按照 AWS CLI 文件中的設定`credentials` 和`config`檢視組態設定中的指示，驗證和檔案中的 AWS 設定檔組態。	DevOps 工程師
驗證 AWS CDK 版本。	執行下列命令來驗證 AWS CDK 工具組的版本： ``` $cdk --version ``` 此病毒碼需要版本 1.107.0 或更新版本。如果您使用的是舊版 AWS CDK，請按照 AWS CDK 文件中的指示進行更新。	DevOps 工程師

任務	描述	所需技能
克隆項目代碼。	使用以下命令克隆此模式的 GitHub 存儲庫： `$git clone https://github.com/aws-samples/amazon-devopsguru-cdk-samples.git`	DevOps 工程師

任務	描述	所需技能
安裝軟件包依賴關係並編譯 TypeScript 文件。	安裝套件相依性，並執行下列命令來編譯 TypeScript 檔案： `$cd amazon-devopsguru-cdk-samples` `$npm install` `$npm fund` 這些指令會安裝範例存放庫中的所有套件。 重要事項：如果您收到有關遺失套件的任何錯誤，請使用下列其中一個指令： `$npm ci` —或— `$npm install -g @aws-cdk/<package-name>` 您可以在檔案的Dependencies 區段中找到套/amazon-devopsguru-cdk-samples/package.json 件名稱和版本的清單。如需詳細資訊，請參閱 npm 文件中的 npm ci 和 npm 安裝。	DevOps 工程師

建置 (合成) AWS CDK 堆疊

任務	描述	所需技能
設定 Amazon SNS 通知的電子郵件地址。	請依照下列步驟提供 Amazon SNS 通知的電子郵件地址： 1. 編輯文件`/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-stack.ts` 和`/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-org-uni-stack.ts` 。 2. 在「`DevOpsGuruTopic Subscription` 」區段中，使用您的電子郵件地址更新`Endpoint`參數。 3. 儲存並關閉檔案。	DevOps 工程師
建置專案程式碼。	通過運行命令來構建項目代碼並合成堆棧： `npm run build && cdk synth` 您應該會看到類似下列的輸出： `$npm run build && cdk synth` `> cdk-devopsguru@0.1.0 build` `> tsc`	DevOps 工程師

任務	描述	所需技能
	`Successfully synthesized to ~/amazon-devopsguru-cdk-samples/cdk.out` `Supply a stack id (CdkDevopsGuruStackMultiAccReg,CdkDevopsGuruStackMultiAccRegSpecStacks, CdkDevopsguruStackOrgUnit, CdkInfrastructureStack, CdkStackSetAdminRole, CdkStackSetExecRole) to display its template.` 如需詳細資訊和步驟，請參閱 AWS CDK 文件中的第一個 AWS CDK 應用程式。	
列出 AWS CDK 堆疊。	執行下列命令以列出所有 AWS CDK 堆疊： `$cdk list` 此指令會顯示下列清單： `CdkDevopsGuruStackMultiAccReg` `CdkDevopsGuruStackMultiAccRegSpecStacks` `CdkDevopsguruStackOrgUnit` `CdkInfrastructureStack` `CdkStackSetAdminRole` `CdkStackSetExecRole`	DevOps 工程師

選項 1-為多個帳戶的所有堆疊資源啟用 DevOps Guru

任務	描述	所需技能
部署 AWS CDK 堆疊以建立 IAM 角色。	此模式使用 AWS CloudFormation StackSets 跨多個帳戶執行堆疊操作。如果您要建立第一個堆疊集，則必須建立下列 IAM 角色，才能在 AWS 帳戶中設定所需的許可： • `AWSCloudFormationStackSetAdministrationRole` • `AWSCloudFormationStackSetExecutionRole` 附註：角色必須具有這些完全相同的名稱。 1. 執行下列 CLI 命令，`AWSCloudFormationStackSetAdministrationRole` 在管理員 (主要) 帳戶中建立 IAM 角色： `$cdk deploy CdkStackSetAdminRole --profile administrator` 2. 在您要執行堆疊執行個體的所有目標帳戶中建立 IAM `AWSCloudFormationStackSetExecutionRo`	DevOps 工程師

任務	描述	所需技能
	le 角色。若要建立此角色，請執行下列 CLI 命令： `$cdk deploy CdkStackSetExecRole \` `  --parameters AdministratorAccountId=<administrator-account-ID> \` `  --profile administrator` `$cdk deploy CdkStackSetExecRole \` `  --parameters AdministratorAccountId=<administrator-account-ID> \` `  --profile target` 如需詳細資訊，請參閱 AWS CloudFormation 文件中的授予自我管理許可。	

任務	描述	所需技能
部署 AWS CDK 堆疊，以跨多個帳戶啟用 DevOps 大師。	AWS CDK CdkDevopsGuruStackMultiAccReg 堆疊可建立堆疊集，以跨多個帳戶和區域部署堆疊執行個體。若要部署堆疊，請使用指定的參數執行下列 CLI 命令： `$cdk deploy CdkDevopsGuruStackMultiAccReg \ --profile administrator \ --parameters AdministratorAccountId=<administrator-account-ID> \ --parameters TargetAccountId=<target-account-ID> \ --parameters RegionIds="<region-1>,<region-2>"` 目前 Amazon DevOps 大師可在大DevOps 師常見問答集中列出的 AWS 區域使用。	DevOps 工程師

選項 2-為整個 O DevOps U 的所有堆疊資源啟用 Guru

任務	描述	所需技能
擷取 OU 識別碼。	在 AWS Organizations 主控台上，識別您要啟用 DevOps Guru 的組織單位 ID。	DevOps 工程師

任務	描述	所需技能
啟用 OU 的服務管理權限。	如果您使用 AWS Organizations 進行帳戶管理，則必須授予服務管理許可才能啟用 DevOps Guru。而不是手動建立 IAM 角色，而是使用組織型受信任存取和服務連結角色 (SLR)。	DevOps 工程師
部署 AWS CDK 堆疊以啟用跨 OU 的 DevOps 大師。	AWS CDK `CdkDevopsguruStackOrgUnit` 堆疊可在整個 OU 中提供 DevOps 大師服務。若要部署堆疊，請使用指定的參數執行下列命令： `$cdk deploy CdkDevopsguruStackOrgUnit \ --profile administrator \ --parameters RegionIds="<region-1>,<region-2>" \ --parameters OrganizationalUnitIds="<OU-1>,<OU-2>"`	DevOps 工程師

選項 3-為跨多個帳戶的特定堆疊資源啟用 DevOps Guru

任務	描述	所需技能
部署 AWS CDK 堆疊以建立 IAM 角色。	如果您尚未建立第一個選項中顯示的必要 IAM 角色，請先執行下列動作： 1. 執行下列 CLI 命令，`AWSCloudF`	DevOps 工程師

任務	描述	所需技能
	ormationStackSetAdministrationRole 在管理員 (主要) 帳戶中建立 IAM 角色： `$cdk deploy CdkStackSetAdminRole --profile administrator` 2. 在您要執行堆疊執行個體的所有目標帳戶中建立 IAM AWSCloudFormationStackSetExecutionRole 角色。若要建立此角色，請執行 CLI 命令： `$cdk deploy CdkStackSetExecRole \` `--parameters AdministratorAccountId=<administrator-account-ID> \` `--profile administrator` `$cdk deploy CdkStackSetExecRole \` `--parameters AdministratorAccountId=<administrator-account-ID> \` `--profile target` 如需詳細資訊，請參閱 AWS CloudFormation 文件中的授予自我管理許可。	

任務	描述	所需技能
刪除現有堆疊。	如果您已經使用第一個選項來啟用所有堆疊資源的 DevOps Guru，您可以使用下列命令刪除舊堆疊： `$cdk destroy CdkDevopsGuruStackMultiAccReg --profile administrator` 或者，您可以在重新部署堆疊時變更 `RegionIds` 參數，以避免堆疊已存在的錯誤。	DevOps 工程師
使用堆疊清單更新 AWS CDK 堆疊。	1. 編輯 `/amazon-devopsguru-cdk-samples/lib/cdk-devopsguru-multi-acc-reg-spec-stack.ts` 檔案。 2. 在`Resources`、`CloudFormation`、下`StackNames`，列出您要啟用 DevOps Guru 的堆疊。為了進行示範，參數會指定`CdkInfrastructureStack` 堆疊，但您可以根據需求編輯此項目。 3. 儲存並關閉檔案。 4. 要合成和更新堆棧模板，請運行： `$cdk synth`	數據工程師

任務	描述	所需技能
部署 AWS CDK 堆疊，以針對多個帳戶的特定堆疊資源啟用 DevOps Guru。	AWS CDK CdkDevopsGuruStackMultiAccRegSpecStacks 堆疊可讓 DevOps 大師處理多個帳戶的特定堆疊資源。若要部署堆疊，請執行下列命令： `$cdk deploy CdkDevopsGuruStackMultiAccRegSpecStacks \ --profile administrator \ --parameters AdministratorAccountId=<administrator-account-ID> \ --parameters TargetAccountId=<target-account-ID> \ --parameters RegionIds="<region-1>,<region-2>"` 注意：如果您先前為選項 1 部署了此堆疊，請變更RegionIds 參數 (確定從可用的區域中選擇)，以避免堆疊已存在的錯誤。	DevOps 工程師

部署 AWS CDK 基礎設施堆疊

任務	描述	所需技能
部署範例無伺服器基礎架構堆疊。	AWS CDK CdkInfrastructureStack 堆疊可部署無伺服器元件，例如	DevOps 工程師

任務	描述	所需技能
	API Gateway、Lambda 和 DynamoDB 表格，以展示大師深入分析資訊。 DevOps 若要部署堆疊，請執行下列命令： `$cdk deploy CdkInfrastructureStack --profile administrator`	
在 DynamoDB 支援中插入範例記錄。	執行下列命令，將範例記錄填入 DynamoDB 表格。提供指令碼的正確路`populate-shops-dynamodb-table.json` 徑。 `$aws dynamodb batch-write-item \ --request-items file://scripts/populate-shops-dynamodb-table.json \ --profile administrator` 該命令會顯示下列輸出： `{ "UnprocessedItems": {} }`	DevOps 工程師

任務	描述	所需技能
驗證在 DynamoDB 中插入的記錄。	若要驗證 DynamoDB 表格是否包含`populate-shops-dynamodb-table.json` 檔案中的範例記錄，請存取 `ListRestApiEndpointMonitorOperator` API 的 URL，該 URL 會以 AWS CDK 堆疊的輸出形式發佈。您也可以在`CdkInfrastructureStack` 堆疊的 AWS CloudFormation 主控台的 [輸出] 索引標籤中找到此 URL。AWS CDK 輸出看起來會類似下列內容： `CdkInfrastructureStack.CreateRestApiMonitorOperatorEndpointD1D00045 = https://oure17c5vob.execute-api.<your-region>.amazonaws.com/prod/` `CdkInfrastructureStack.ListRestApiMonitorOperatorEndpointABBDB8D8 = https://cdff8icfrn4.execute-api.<your-region>.amazonaws.com/prod/`	DevOps 工程師

任務	描述	所需技能
等待資源完成基準。	此無伺服器堆疊具有一些資源。我們建議您等待 2 個小時，然後再執行後續步驟。如果您在生產環境中部署此堆疊，則最多可能需要 24 小時才能完成基準，視您在 DevOps Guru 中選取要監視的資源數量而定。	DevOps 工程師

產生 DevOps 大師見解

任務	描述	所需技能
更新 AWS CDK 基礎設施堆疊。	要嘗試 DevOps Guru 見解，您可以進行一些配置更改以重現典型的操作問題。 1. 編輯 `/amazon-devopsguru-cdk-samples/lib/infrastructure-stack.ts` 檔案。 2. 在此`DDB Table`區段中，將 DynamoDB 表的讀取容量從 5 變更為 1。 3. 儲存並關閉檔案。 4. 執行下列命令來合成和部署更新的 AWS CDK 基礎設施堆疊： `$cdk synth` `$cdk deploy CdkInfrastructureStack --profile administrator`	DevOps 工程師

任務	描述	所需技能
在 API 上插入 HTTP 要求。	在 `ListRestApiMonitorOperatorEndpointxxxx` API 上以 HTTP 要求的形式插入輸入流量： 1. 編輯 Python 腳本`/amazon-devopsguru-cdk-samples/scripts/sendAPIRequest.py` 。 2. 使用的 API 連結更新`url`變數`ListRestApiMonitorOperatorEndpointxxxx` 。您可以在 AWS CDK 部署命令的輸出中，或在 AWS Cloudform 主控台的堆疊的「輸出」索引標籤中找到此 URL。 3. 儲存並關閉檔案。 4. 使用以下命令來執行 Python 指令碼： `$python sendAPIRequest.py` 5. 確保您獲得 200 狀態碼。 6. 您可能需要透過多個 (最好是四個) 終端機執行指令碼，以高速插入流量。 7. 指令碼在一個迴圈中執行大約 10 分鐘後，您可以在 DevOps Guru 主控台上看到操作深入分析。	DevOps 工程師

任務	描述	所需技能
檢閱 DevOps 大師見解。	在標準條件下， DevOps Guru 儀表板會在進行中的見解計數器中顯示零。如果偵測到異常，就會以深入解析的形式提出警示。在導覽窗格中，選擇「見解」以查看異常的詳細資料，包括概觀、彙總量度、相關事件和建議。如需有關檢閱見解的詳細資訊，請參閱使用 Amazon DevOps Guru 透過 AIOps 取得營運深入解析部落格文章。	DevOps 工程師

清除

任務	描述	所需技能
清理和刪除資源。	逐步完成此模式之後，您應該移除建立的資源，以避免產生任何進一步的費用。執行這些命令： `$cdk destroy CdkDevopsGuruStackMultiAccReg --profile administrator` `$cdk destroy CdkDevopsguruStackOrgUnit --profile administrator` `$cdk destroy CdkDevopsGuruStackMultiAccRegSpecStacks --profile administrator` `$cdk destroy CdkInfrastructureStack --profile administrator`	DevOps 工程師

任務	描述	所需技能
	`$cdk destroy CdkStackSetAdminRole --profile administrator` `$cdk destroy CdkStackSetExecRole --profile administrator` `$cdk destroy CdkStackSetExecRole --profile target`	

相關資源

- 使用 Amazon DevOps 大師使用 AIOps 獲得營運見解
- 使用 AWS 輕鬆設定跨多個帳戶和區域的 Amazon DevOps 大師 CloudFormation StackSets
- DevOps 大師工作坊

使用啟動程序管道實作地形 (AFT) 的 Account Factory

由維尼修斯埃利亞斯（AWS）和埃德加科斯塔菲略（AWS）創建

代碼存儲庫：aft-bootstrap-pipeline	環境：生產	技術：管理與治理；基礎設施
工作負載：開源	AWS 服務：AWS CodeBuild；AWS；AWS CodeCommit CodePipeline；AWS Control Tower；AWS Organizations	

Summary

此模式提供了一種簡單且安全的方法，可從的管理 AWS Control Tower 帳戶部署 Terraform (AFT) 的 Account Factory。 AWS Organizations解決方案的核心是一個 AWS CloudFormation 範本，可透過建立 Terraform 管線來自動執行 AFT 組態，該管線的結構可輕鬆適應初始部署或後續更新。

安全性和資料完整性是首要任務 AWS，因此 Terraform 狀態檔案是追蹤受管基礎設施和組態狀態的關鍵元件，安全地存放在 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體中。該存儲桶配置了多種安全措施，包括服務器端加密和阻止公共訪問的策略，以幫助確保您的 Terraform 狀態受到保護，防止未經授權的訪問和數據洩露。

管理帳戶可協調和監督整個環境，因此它是中的重要資源。 AWS Control Tower此模式遵循 AWS 最佳實務，並確保部署程序不僅有效率，而且還符合安全性和治理標準，以提供全面、安全且有效率的方式來在您的 AWS 環境中部署 AFT。

如需 AFT 的詳細資訊，請參閱AWS Control Tower 文件。

先決條件和限制

先決條件

- 一個基本的 AWS 多帳戶環境，至少具有以下帳戶：管理帳戶，日誌存檔帳戶，審計帳戶以及一個額外的 AFT 管理帳戶。
- 一個既定的 AWS Control Tower 環境。管理帳戶應該正確設定，因為 CloudFormation 範本會在其中部署。

- AWS 管理帳戶中的必要權限。您需要足夠的許可來建立和管理資源，例如 S3 儲存貯體、 AWS Lambda 函數、 AWS Identity and Access Management (IAM) 角色和 AWS CodePipeline 專案。
- 熟悉地形。了解 Terraform 的核心概念和工作流程非常重要，因為部署涉及產生和管理 Terraform 組態。

限制

- 請注意您帳戶中的AWS 資源配額。部署可能會建立多個資源，且遇到服務配額可能會阻礙部署程序。
- 該模板是專為特定版本的地形和。 AWS 服務升級或變更版本可能需要修改範本。

產品版本

- 地形版本 1.5.7 或更新版本
- 船尾服務版本 1.11.1 或更新版本

架構

目標技術堆疊

- AWS CloudFormation
- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon EventBridge
- IAM
- AWS Lambda
- Amazon S3

目標架構

下圖說明了此模式中討論的實現。

工作流程包含三個主要任務：建立資源、產生內容和執行管道。

建立資源

此CloudFormation 模式提供的範本會根據您在部署範本時選取的參數，建立並設定所有必要的資源。範本至少會建立下列資源：

- CodeCommit 存儲 AFT 地形引導程序代碼的存儲庫
- 用於存放與 AFT 實作相關聯的 Terraform 狀態檔案的 S3 儲存貯體
- 一 CodePipeline 條管道
- 兩個 CodeBuild 項目實施 Terraform 計劃並在管道的不同階段應用命令
- 適用於 CodeBuild 和 CodePipeline 服務的 IAM 角色
- 用於存放管道執行階段成品的第二個 S3 儲存
- 擷取`main`分支上 CodeCommit 儲存庫變更的 EventBridge 規則
- EventBridge 規則的另一個 IAM 角色

此外，如果您將 CloudFormation 範本中的`Generate AFT Files`參數設定為`true`，範本會建立下列額外資源來產生內容：

- 用於存放產生的內容並用作儲存庫來源的 S3 CodeCommit 儲存貯體
- Lambda 函數，用於處理給定的參數並生成適當的內容
- 用於執行 Lambda 函數的 IAM 函數
- 部署範本時執行 Lambda 函數的 CloudFormation 自訂資源

生成內容

為了產生 AFT 啟動程序檔案及其內容，此解決方案使用 Lambda 函數和 S3 儲存貯體。函數會在值區中建立資料夾，然後在資料夾內建立兩個檔案：`main.tf`和`backend.tf`。該函數還處理提供的 CloudFormation 參數，並用預定義的代碼填充這些文件，替換相應的參數值。

若要檢視做為範本來產生檔案的程式碼，請參閱解決方案的GitHub 存放庫。基本上，文件生成如下。

主 .tf

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory?
ref=<aft_version>"
```

```
  # Required variables
  ct_management_account_id  = "<ct_management_account_id>"
  log_archive_account_id    = "<log_archive_account_id>"
  audit_account_id          = "<audit_account_id>"
  aft_management_account_id = "<aft_management_account_id>"
  ct_home_region            = "<ct_home_region>"

  # Optional variables
  tf_backend_secondary_region = "<tf_backend_secondary_region>"
  aft_metrics_reporting       = "<false|true>"

  # AFT Feature flags
  aft_feature_cloudtrail_data_events      = "<false|true>"
  aft_feature_enterprise_support          = "<false|true>"
  aft_feature_delete_default_vpcs_enabled = "<false|true>"

  # Terraform variables
  terraform_version      = "<terraform_version>"
  terraform_distribution = "<terraform_distribution>"

}
```

後端 .tf

```
terraform {
  backend "s3" {
    region = "<aft-main-region>"
    bucket = "<s3-bucket-name>"
    key    = "aft-setup.tfstate"
  }
}
```

在 CodeCommit 存放庫建立期間，如果您將`Generate AFT Files`參數設定為`true`，範本會使用 S3 儲存貯體及產生的內容做為`main`分支的來源，以自動填入存放庫。

運行管道

建立資源並設定啟動程序檔案之後，管線便會執行。第一階段（源）從存儲庫的主分支獲取源代碼，第二階段（構建）運行 Terraform 計劃命令並生成要審查的結果。在第三階段 (核准) 中，管線會等待手動動作核准或拒絕最後階段 (部署)。在最後階段，管線會使用前一個 Terraform `apply` 命令的結果作為輸入來執行 Terraform `plan` 命令。最後，會使用跨帳戶角色和管理帳戶中的權限，在 AFT 管理帳戶中建立 AFT 資源。

工具

AWS 服務

- AWS CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit是一種版本控制服務，可以幫助您私下存儲和管理 Git 存儲庫，而無需管理自己的源代碼控制系統。
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS Lambda是一種運算服務，可執行程式碼以回應事件並自動管理運算資源，提供快速建立適用於生產環境的現代化無伺服器應用程式。
- AWS SDK for Python (Boto3)是一個軟體開發套件，可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。

其他工具

- Terraform 是一種基礎架構即程式碼 (IaC) 工具，可讓您安全有效地建置、變更和版本基礎架構。這包括運算執行個體、儲存體和網路等低階元件，以及 DNS 項目和 SaaS 功能等高階元件。
- Python 是一種易於學習，功能強大的編程語言。它具有高效的高級數據結構，並提供了一種簡單而有效的方法來面向對象的編程。

代碼存儲庫

此模式的代碼可在 GitHub AFT 啟動程序管道存儲庫中找到。

如需官方 AFT 儲存庫，請參閱中的地形 AWS Control Tower Account Factory。 GitHub

最佳實務

當您使用提供的 CloudFormation 範本部署 AFT 時，我們建議您遵循最佳做法，以協助確保安全、有效率且成功的實作。推行和操作船尾服務的主要指引和建議包括以下內容。

- 徹底檢閱參數：仔細檢閱並瞭解 CloudFormation 範本中的每個參數。準確的參數配置對於 AFT 的正確設置和功能至關重要。

- 定期模板更新：保持模板與最新的 AWS 功能和 Terraform 版本更新。定期更新可幫助您充分利用新功能並維護安全性。
- 版本控制：固定您的 AFT 模塊版本，並在可能的情況下使用單獨的 AFT 部署進行測試。
- 範圍：僅使用 AFT 來部署基礎結構護欄和自訂項目。請勿使用它來部署應用程式。
- 絨毛和驗證：AFT 管道需要一個連接和經過驗證的 Terraform 配置。在將組態推送至 AFT 儲存庫之前，請先執行 lint、驗證和測試。
- Terraform 模塊：將可重複使用的 Terraform 代碼構建為模塊，並始終指定 Terraform 和 AWS 提供程序版本以符合您組織的需求。

史诗

設定和設定 AWS 環境

任務	描述	所需技能
準備 AWS Control Tower 環境。	AWS Control Tower 在您的 AWS 環境中進行設定和配置，以確保您的 AWS 帳戶. 如需詳細資訊，請參閱 AWS Control Tower 文件 AWS Control Tower中的入門。	雲端管理員
啟動 AFT 管理帳戶。	使用 AWS Control Tower Account Factory 啟動新的作 AWS 帳戶 為您的 AFT 管理帳戶。如需詳細資訊，請參閱 AWS Control Tower 文件中的使用 AWS Service Catalog Account Factory 佈建帳戶。	雲端管理員

在管理帳戶中部署 CloudFormation 範本

任務	描述	所需技能
啟動 CloudFormation 範本。	在這個史詩中，您可以部署此解決方案提供的 CloudFormation 範本，以便在 AWS 管理帳戶中設定 AFT 啟動程序管道。管道會在您先前史詩中設定的 AFT 管理帳戶中部署 AFT 解決方案。 步驟 1：開啟主 AWS CloudFormation 控台 • 登入 AWS Management Console 並開啟AWS CloudFormation 主控台。確保您在正確的 AWS Control Tower 主要區域內操作。 步驟 2：建立新堆疊 1. 選擇建立新堆疊。 2. 選取上傳範本檔案的選項，然後上傳此CloudFormation模式提供的範本。 步驟 3：設定堆疊參數 • `Repository Name` ：指定儲存 AFT 啟動程序模組的存放庫名稱。 • `Branch Name`：指定來源儲存庫分支。 • `CodeBuild Docker Image`：選擇要用作	雲端管理員

任務	描述	所需技能
	CodeBuild Docker 基本映像檔的檔案。 步驟 4：決定檔案產生 • 此`Generate AFT Files`參數可控制是否產生預設 AFT 部署檔案。將此參數設定為： • `true`，自動建立並將 AFT 部署檔案儲存在指定的儲存庫中。 • `false`如果你想手動處理文件創建或已經有了文件。 如果您選取`false`，請跳至步驟 8；否則，請先執行步驟 5-7。 步驟 5：填寫 AWS Control Tower 和 AFT 帳戶詳細信息 • 輸入 AWS Control Tower 和 AFT 帳戶特定資訊： • `Log Archive Account ID`：中日誌存檔帳戶 ID 的 ID AWS Control Tower。 • `Audit Account ID`：中稽核帳戶的識別碼 AWS Control Tower。 • `AFT Management Account ID`：您在第一	

任務	描述	所需技能
	部史詩中創建的 AFT 管理帳戶的 ID。 • `AFT Main Region`和`AFT Secondary Region`：AFT 部署的主要和次 AWS 區域 要。 步驟 6：設定 AFT 選項 • 設定量度報告： • `AFT Enable Metrics Reporting`：啟用或停用 AFT 量度報告。如需詳細資訊，請參閱 AWS Control Tower 文件中的操作指標。 • 設定 AFT 功能選項： • `Enable AFT CloudTrail Data Events`：在所有 AFT 管理帳戶中啟用 CloudTrail 資料事件。如需詳細資訊，請參閱文件中的資AWS CloudTrail 料事件 AWS Control Tower。 • `Enable AFT Enterprise Support`：在所有 AFT 管理帳戶中啟用企業 Support。如需詳細資訊，請參閱 AWS Control	

任務	描述	所需技能
	Tower 文件中的AWS 企業 Support 計劃。 • `Enable AFT Delete Default VPC`：僅刪除 AFT 管理帳戶中的所有 VPC。如需詳細資訊，請參閱 AWS Control Tower 文件中的刪除 AWS 預設 VPC。 步驟 7：指定版本 • `AFT Terraform Version`：選擇要在 AFT 管線中使用的地形版本。 • `AFT Version`：定義要部署的 AFT 版本。保留預設設定 (`latest`) 以使用最新的 AFT 版本。 步驟 8：檢閱並建立堆疊 • 檢閱所有參數和設定。如果一切正常，請繼續創建堆棧。 步驟 9：監視堆疊建立 • AWS CloudFormation 佈建和配置您定義的資源。在 CloudFormation 主控台上監視堆疊建立程序。此過程可能需要幾分鐘的時間。	

任務	描述	所需技能
	步驟 10：驗證部署 • 當堆疊狀態顯示「建立 _ 完成」時，請確認已正確建立所有資源。 • 在「輸出」區段中，記下`TerraformBackendBucketName` 值。	

填入並驗證 AFT 啟動程序儲存庫和管線

任務	描述	所需技能
填入 AFT 啟動程序儲存庫。	(選擇性) 部署 CloudFormation 範本後，您可以在新建立的 AFT 啟動程序存放庫中填入或驗證內容，並測試管線是否已成功執行。 如果將`Generate AFT Files`參數設定為`true`，請跳至下一個內文 (驗證管線)。 步驟 1：填入儲存庫 1. 開啟主AWS CodeCommit 控台並選取新建立的存放庫。如果您保留預設名稱，則會呼叫儲存庫`aft-setup`。 2. 使用安全殼層、HTTPS 或 HTTPS (GRC) 將存放庫複製到您的本機電腦，然後在編輯器中開啟它。	雲端管理員

任務	描述	所需技能
	3. 創建一個名為的文件夾，terraform 並在其中創建兩個空文件：backend.tf 和main.tf。 4. 打開文backend.tf 件並添加以下代碼片段： `terraform { backend "s3" { region = "<aft-main-region>" bucket = "<s3-bucket-name>" key = "aft-setup" } }` 在檔案中： • 以主要<aft-main-region>的船尾區域取代。這應該符合 AWS Control Tower 主要區域。 • <s3-bucket-name> 以地形後端儲存貯體的名稱取代。您可以在先前部署的 CloudFormation 範本所產生的Terraform BackendBucketName 輸出中找到此項目。 5. 開啟main.tf檔案並使用AFT 儲存庫中可用的其中	

任務	描述	所需技能
	一個範例來部署 AFT。例如，您可以使用偏好的版本控制系統 (VCS) 提供者 (CodeCommit GitHub、或 Bitbucket)，或自訂 AFT VPC。如需更多 AFT 輸入選項，請參閱 AFT 儲存庫中的 README 檔案。 步驟 2：提交並推送您的更改 • 建立並填入資料夾和檔案之後，請確認您的變更，並將程式碼上傳至儲存庫。管線會自動啟動，在 [來源] 和 [建置] 階段執行，然後在 [部署] 階段之前等待核准動作。	

任務	描述	所需技能
驗證 AFT 啟動程序管線。	步驟 1：檢視管道 • 開啟主CodePipeline 控台並檢查aft-bootstrap-pipeline 管線是否已成功啟動。它應該運行 Terraform 計劃或等待手動批准操作。 步驟 2：核准地形計劃結果 • 您可以檢閱 Terraform 計劃的結果，方法是查看「建置」階段的執行記錄，然後在「核准」階段核准或拒絕執行。如果您核准，管線會開始在提供的 AFT 管理帳戶中部署 AFT 資源。 步驟 3：等待部署 • 等待管線成功執行。這應該需要大約 30 分鐘。您可能遇到的任何失敗通常是 API 配額造成的。在這些情況下，您可以重新執行管道以繼續部署。 步驟 4：檢查創建的資源 • 存取 AFT 管理帳戶，並確認已建立資源。	雲端管理員

故障診斷

問題	解決方案
CloudFormation 範本中包含的自訂 Lambda 函數會在部署期間失敗。	檢查 Amazon CloudWatch 日誌中的 Lambda 函數以識別錯誤。記錄檔提供詳細資訊，可協助您找出特定問題。確認 Lambda 函數具有必要的權限，並確認環境變數已正確設定。
您在資源建立或管理中遇到因權限不當而造成的失敗。	檢閱附加至 Lambda 函數的 IAM 角色和政策 CodeBuild，以及部署中涉及的其他服務。確認他們具有必要的權限。如果存在權限問題，請調整 IAM 政策以授予所需的存取權。
您正在使用具有較新 AWS 服務 或 Terraform 版本的過時版本的 CloudFormation 模板版本。	定期更新 CloudFormation 模板，以與最新的 AWS 和 Terraform 版本兼容。請查看版本說明或文件，瞭解任何版本特定的變更或需求。
您在部署期間達到 AWS 服務 配額。	部署管道之前，請檢查 S3 儲存貯體、IAM 角色和 Lambda 函數等資源的 AWS 服務 配額。如有必要，請求增加。如需詳細資訊，請參閱 AWS 網站AWS 服務 配額。
由於 CloudFormation 範本中輸入參數不正確，導致您遇到錯誤。	仔細檢查所有輸入參數是否有錯別字或不正確的值。確認資源識別碼 (例如帳號 ID 和區域名稱) 正確無誤。

相關資源

若要成功實作此模式，請檢閱下列資源。這些資源提供了額外的資訊和指導，這些資訊和指導在使用設定和管理 AFT 方面非常寶貴。 AWS CloudFormation

AWS文件：

- AWS Control Tower 用戶指南提供有關設置和管理的詳細信息 AWS Control Tower。
- AWS CloudFormation 文件提供 CloudFormation 範本、堆疊和資源管理的深入解析。

IAM 政策和最佳做法：

- IAM 中的安全最佳實務說明如何使用 IAM 角色和政策來協助保護 AWS 資源安全。

地形上： AWS

- 地形表單提 AWS 供者文件提供有關使用 Terra form 與. AWS

AWS 服務 配額：

- AWS 服務 配@@ 額提供有關如何查看 AWS 服務 配額以及如何增加請求的信息。

管理多個 AWS 帳戶和 AWS 區域的 AWS 服務目錄產品

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：管理與治理、雲端原生、基礎架構、現代化	工作負載：所有其他工作
AWS 服務：AWS Service Catalog；AWS CloudFormation		

Summary

Amazon Web Services (AWS) Service Catalog 可簡化並加速企業的基礎設施即程式碼 (IaC) 範本的管理和分發。您可以使用 AWS CloudFormation 範本定義產品所需的 AWS 資源 (堆疊) 集合。AWS 可讓您透過單一操作跨多個帳戶和 AWS 區域建立、更新或刪除堆疊，藉此 CloudFormation StackSets 擴充此功能。

AWS Service Catalog 管理員使用開發人員撰寫的 CloudFormation 範本來建立產品，然後發佈產品。然後，這些產品會與產品組合相關聯，並將限制套用於治理。為了讓其他 AWS 帳戶或組織單位 (OU) 的使用者可以使用您的產品，您通常會與他們共用您的產品組合。此模式描述管理以 AWS 為基礎的 AWS Service Catalog 產品供應項目的替代方法 CloudFormation StackSets。您可以使用堆疊集合約束來設定 AWS 區域和帳戶，而不是共用產品組合，以便部署和使用產品。使用此方法，您可以在多個帳戶、OU 和 AWS 區域佈建 AWS Service Catalog 產品，並從中央位置進行管理，同時滿足您的管理要求。

這種方法的好處：

- 產品是從主要帳戶佈建和管理，而不會與其他帳戶共用。
- 此方法提供以特定產品為基礎的所有佈建產品 (堆疊) 的整合檢視。
- 使用 AWS 服務管理連接器進行設定比較容易，因為它只針對一個帳戶。
- 從 AWS Service Catalog 查詢和使用產品變得更加容易。

先決條件和限制

先決條件

- 適用於 IaC 和 CloudFormation 版本控制的 AWS 範本
- 用於佈建和管理 AWS 資源的多帳戶設定和 AWS 服務目錄

限制

- 此方法使用 AWS CloudFormation StackSets，以及 StackSets 適用的限制：
  - StackSets 不支援透過巨集進行 CloudFormation 範本部署。如果您使用巨集來預先處理範本，您將無法使用 StackSets基於的部署。
  - StackSets 提供了取消堆棧與堆棧集的關聯的功能，因此您可以定位特定堆棧以解決問題。但是，取消關聯的堆棧不能與堆棧集重新關聯。
- AWS Service Catalog 會自動產生 StackSet 名稱。目前不支援自訂。

架構

目標架構

1. 使用者建立 AWS CloudFormation 範本以 JSON 或 YAML 格式佈建 AWS 資源。
2. CloudFormation 範本會在 AWS Service Catalog 中建立產品，並將其新增至產品組合。
3. 使用者會建立已佈建的產品，以便在目標帳戶中建立 CloudFormation 堆疊。
4. 每個堆疊都會佈 CloudFormation 建範本中指定的資源。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

- AWS Service Catalog 可協助您集中管理 AWS 核准的 IT 服務目錄。最終使用者可在機構所設的限制範圍內，迅速地只部署自己需要且經核准的 IT 服務。

史诗

跨帳戶佈建產品

任務	描述	所需技能
建立組合。	產品組合是一個容器，其中包含一或多個根據特定條件分組在一起的產品。使用產品組合可協助您在整個產品組合中套用共同的限制條件。 若要建立產品組合，請遵循 AWS Service Catalog 文件中的指示。如果您使用的是 AWS CLI，以下是一個示例命令： `aws servicecatalog create-portfolio --provider-name my-provider --display-name my-portfolio` 如需詳細資訊，請參閱 AWS CLI 文件。	AWS Service Catalog
建立 CloudFormation 範本。	建立描述資源的 CloudFormation 範本。資源屬性值應在適當的情況下進行參數化。	AWS CloudFormation, JSON / 亞姆
使用版本資訊建立產品。	當您在 AWS Service Catalog 中發佈 CloudFormation 範本時，範本會變成產品。提供選擇性版本詳細資料參數的值，	AWS Service Catalog

任務	描述	所需技能
	例如版本標題和說明；這將有助於稍後查詢產品。 若要建立產品，請遵循 AWS Service Catalog 文件中的指示。如果您使用的是 AWS CLI，則範例命令為： `aws servicecatalog create-product --cli-input-json file://create-product-input.json` 其中`create-product-input.json` 是傳遞產品參數的檔案。如需此檔案的範例，請參閱「其他資訊」一節。如需詳細資訊，請參閱 AWS CLI 文件。	
套用約束。	將堆疊集限制套用至產品組合，以設定產品部署選項，例如多個 AWS 帳戶、區域和許可。如需指示，請參閱 AWS Service Catalog 文件。	AWS Service Catalog

任務	描述	所需技能
新增 許可。	為使用者提供權限，以便他們可以啟動產品組合中的產品。如需主控台指示，請參閱 AWS Service Catalog 文件。如果您使用的是 AWS CLI，以下是一個示例命令： `aws servicecatalog associate-principal-with-portfolio \ --portfolio-id port-2s6abcdefwdh4 \ --principal-arn arn:aws:iam::444455556666:role/Admin \ --principal-type IAM` 如需詳細資訊，請參閱 AWS CLI 文件。	AWS Service Catalog

任務	描述	所需技能
佈建產品。	佈建的產品是產品的資源執行個體。根據 CloudFormation 範本佈建產品會啟動 CloudFormation 堆疊及其基礎資源。 根據堆疊集限制，以適用的 AWS 區域和帳戶為目標佈建產品。在 AWS CLI 中，以下是一個示例命令： `aws servicecatalog provision-product \ --product-id prod-abcdfz3syn2rg \ --provisioning-artifact-id pa-abc347 pcsccfm \ --provisioned-product-name "mytestpp name3"` 如需詳細資訊，請參閱 AWS CLI 文件。	AWS Service Catalog

相關資源

參考

- AWS Service Catalog 概觀
- 使用 AWS CloudFormation StackSets

教學課程和影片

- AWS RE: 創新 2019：自動化一切：選項和最佳實務 (影片)

其他資訊

使用指`create-product`令時，`cli-input-json`參數會指向指定資訊的檔案，例如產品擁有者、支援電子郵件和 CloudFormation 範本詳細資料。下面是這樣一個文件的例子：

```
{
   "Owner": "Test admin",
      "SupportDescription": "Testing",
         "Name": "SNS",
            "SupportEmail": "example@example.com",
            "ProductType": "CLOUD_FORMATION_TEMPLATE",
               "AcceptLanguage": "en",
                  "ProvisioningArtifactParameters": {
                     "Description": "SNS product",
                        "DisableTemplateValidation": true,
                           "Info": {
                              "LoadTemplateFromURL": "<url>"
                     },
                           "Name": "version 1"
}
```

將 AWS 成員帳戶從 AWS Organizations 遷移到 AWS Control Tower

由小魯道夫創作 塞拉達 (AWS)

環境：生產	技術：管理與治理；現代化	AWS 服務：AWS Organizations；AWS Control Tower

Summary

此模式說明如何將 Amazon Web Services (AWS) 帳戶從 AWS Organizations 遷移到 AWS 組織 (由管理帳戶管理的成員帳戶) 到 AWS Control Tower。透過在 AWS Control Tower 註冊帳戶，您可以利用預防性和偵探護欄以及簡化帳戶管理的功能。如果您的 AWS Organization 管理帳戶遭到入侵，而您想要將成員帳戶移至受 AWS Organizations Control Tower 管理的新組織，您也可能想要遷移您的成員帳戶。

AWS Control Tower 提供的架構可結合並整合其他多個 AWS 服務 (包括 AWS Organizations) 的功能，並確保跨多帳戶環境一致的合規和管控。使用 AWS Control Tower，您可以遵循一組規定的規則和定義，以擴展 AWS Organizations 的功能。例如，您可以使用護欄來確保已建立安全性記錄檔和必要的跨帳戶存取權限，而且不會變更。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 AWS Organizations 的目標組織中設定 AWS Control Tower (如需指示，請參閱 AWS Control Tower 文件中的設定)
- AWS Control Tower (AWSControlTowerAdmins群組成員) 的管理員登入資料
- 來源 AWS 帳戶的管理員登入資料

限制

- AWS Organizations 中的來源管理帳戶必須與 AWS Control Tower 中的目標管理帳戶不同。

產品版本

- AWS Control Tower 2.3 版 (2020 年 2 月) 或更新版本 (請參閱版本說明)

架構

下圖說明移轉程序和參考架構。此模式會將 AWS 帳戶從來源組織遷移到受 AWS Control Tower 管理的目標組織。

註冊程序包含下列步驟：

1. 該帳戶離開 AWS Organizations 中的來源組織。
2. 該帳戶將成為獨立帳戶。這意味著它不屬於任何組織，因此管理和帳單由帳戶管理員獨立管理。
3. 目標組織會傳送帳戶加入組織的邀請。
4. 獨立帳戶接受邀請，並成為目標組織的成員。
5. 該帳戶已註冊 AWS Control Tower，並移至已註冊的組織單位 (OU)。(我們建議您查看 AWS Control Tower 儀表板以確認註冊。) 此時，已註冊 OU 中啟用的所有護欄都會生效。

工具

AWS 服務

- AWS Organizations Organization 是一種帳戶管理服務，可讓您將多個 AWS 帳戶合併到您建立並集中管理的單一實體 (一個組織) 中。
- AWS Control Tower 整合了其他服務的功能，包括 AWS Organizations、AWS IAM 身分中心 (AWS Single Sign-On 的後續產品) 和 AWS Service Catalog，可協助您在 AWS 雲端中對所有組織和帳戶大規模執行和管理安全、操作和合規的管控規則。

史诗

從來源組織移除成員帳戶

任務	描述	所需技能
確認成員帳戶可以作為獨立帳戶執行。	確認將離開來源組織的成員帳戶具有作為獨立帳戶作業所需的資訊。例如，如果成員帳戶沒有帳單資訊，則無法以獨立帳戶的形式運作，因為 AWS 會使用付款資訊對帳戶未連結至組織時發生的任何可計費 AWS 活動收取費用。 通常，如果您使用 AWS Organizations 主控台、API 或 AWS 命令列界面 (CLI) 命令建立成員帳戶，則不會自動收集獨立帳戶所需的資訊。若要新增此資訊，請登入帳戶，然後指定支援方案、聯絡資訊和付款方式。 如需從組織中移除帳戶之前需要瞭解的詳細資訊，請參閱 AWS Organizations Organization 文件中的「從組織移除帳戶之前」。	帳戶管理員
從其來源組織中移除成員帳戶。	按照 AWS Organizations 文件中的指示從組織中移除成員帳戶。您可以登入組織的管理帳戶並移除成員帳戶，或登入成員帳戶並離開組織。	管理帳戶管理員或帳戶管理員

任務	描述	所需技能
	如果您沒有管理員層級認證可移除或離開帳戶，請向組織的管理員尋求協助。 如果會員帳戶缺少支援方案、聯絡資訊或付款資訊，系統會提示您提供並確認該資訊。 當您離開組織時，系統會將您重新導向至 AWS Organization 主控台的 [入門] 頁面，您可以在其中檢視帳戶加入其他組織的邀請。 重要提示：此時，您的帳戶是獨立帳戶。如果您執行的工作負載不在 AWS 免費方案涵蓋範圍內，我們會根據您為該帳戶提供的付款和帳單資訊向您收費。	
確認成員帳戶不再是來源組織的一部分。	在 AWS Organizations 主控台中，您應該不會再看到 [離開組織] 按鈕。反之，您應該會看到來自其他組織的待處理邀請（如果有的話）。	帳戶管理員

任務	描述	所需技能
從您離開的組織中移除授予帳戶存取權的 IAM 角色。	從來源組織移除帳戶時，AWS Organizations 或管理員建立的 AWS Identity and Access Management (IAM) 角色不會自動刪除。若要終止來源組織管理帳戶的存取權，您必須手動刪除 IAM 角色。如需詳細資訊，請參閱 IAM 文件中的刪除角色或執行個體設定檔。 當成員帳戶離開組織時，所有附加至該帳戶的標籤都會遭到刪除。獨立帳戶不支援標籤。	帳戶管理員

透過 AWS Control Tower 邀請帳戶加入新組織

任務	描述	所需技能
登入 AWS Control Tower。	以管理員身分登入 AWS Control Tower 主控台。 目前沒有直接的方法可以將 AWS 帳戶從來源組織移至由 AWS Control Tower 管理的 OU 中的組織。不過，當您將 AWS Control Tower 管理註冊到已由 AWS 控制塔管理的 OU 時，您可以將 AWS Control Tower 管理延伸到現有的 AWS 帳戶。這就是為什麼您必須登入 AWS Control Tower 才能執行此步驟。	AWS Control Tower 管理員

任務	描述	所需技能
邀請會員帳戶。	1. 登入 AWS Organizations 主控台，然後導覽至 AWS 帳戶頁面。 2. 在 [新增 AWS 帳戶] 頁面上，選擇 [邀請現有 AWS 帳戶]。 3. 填寫帳戶資訊，包括 12 位數的帳號 (不含破折號) 以及選用的說明和標籤，然後選擇 [傳送邀請]。 重要事項：確認帳戶轉移不會影響任何應用程式或網路連線。 此動作會傳送邀請電子郵件，其中包含成員帳戶的連結。當帳戶管理員跟隨連結並接受邀請時，成員帳戶就會出現在 AWS 帳戶頁面中。如需詳細資訊，請參閱 AWS 組織文件中的邀請 AWS 帳戶加入您的組織。	AWS Control Tower 管理員

任務	描述	所需技能
測試應用程序和連接性。	當成員帳戶已註冊到新組織時，它會出現在根目錄內的 OU 中。它也會出現在 AWS Control Tower 主控台中，標記為尚未註冊帳戶，因為尚未在 AWS Control Tower 註冊的 OU 中註冊該帳戶。 請確認下列內容： • 查看 AWS Control Tower 儀表板，瞭解是否有任何防護欄違規情況。 • 請檢查網路連線 (VPN 或 AWS Direct Connect 線)，確定不受傳輸影響。 • (應用程式擁有者) 測試與此帳戶相關聯的應用程式，以確認它們如預期般執行，而且相依性不受帳戶移轉的影響。	AWS Control Tower 管理員、成員帳戶管理員、應用程式擁有者

準備註冊的帳戶

任務	描述	所需技能
檢閱護欄並修復所有違犯。	檢閱在目標 OU 中定義的護欄，尤其是預防護欄，並修正任何違規。 當您設定 AWS Control 塔 landing zone 時，預設會啟用一些強制性的預防性護欄。這些無法停用。在註冊帳戶之前，您必須檢閱這些強制性護	AWS Control Tower 管理員，成員帳戶管理員

任務	描述	所需技能
	欄並修復成員帳戶 (手動或使用指令碼)。 注意：預防性護欄可確保 AWS Control Tower 註冊帳戶合規，並防止違反政策。任何違反預防性護欄的行為都可能會影響註冊。成功註冊後，Detective 護欄違規行為會出現在 AWS Control Tower 儀表板中 (如果偵測到)。它們不會影響註冊程序。如需詳細資訊，請參閱 AWS 文件中的 AWS Control Tower 中的護欄。	
在修復護欄違規後檢查連線問題。	在某些情況下，您可能必須關閉特定連接埠或停用服務，才能修正護欄違規問題。在註冊帳戶之前，請確定已修正使用這些連接埠和服務的應用程式。	應用程式擁

將帳戶註冊到 AWS Control Tower

任務	描述	所需技能
登入 AWS Control Tower 主控台。	使用具有 AWS Control Tower 管理許可的登入登入資料。請勿使用根使用者 (管理帳戶) 登入資料來註冊 AWS Organizations 帳戶。這將顯示一條錯誤消息。	AWS Control Tower 管理員

任務	描述	所需技能
註冊帳戶。	1. 在 AWS Control Tower 的 Account Factory 頁面中，選擇註冊帳戶。 2. 填寫詳細資訊，包括與您要註冊之帳戶相關聯的電子郵件地址、將出現在 AWS Control Tower 中的顯示名稱、IAM 身分中心電子郵件地址、帳戶擁有者的名字和姓氏，以及您要註冊帳戶的 OU。IAM 身分中心電子郵件地址是您偏好的使用者電子郵件地址。您可以使用與帳戶電子郵件相同的電子郵件地址。 3. 選擇 Enroll acount (註冊帳戶)。 如需詳細資訊，請參閱 AWS Control Tower 文件中的註冊現有帳戶。	AWS Control Tower 管理員

註冊後驗證帳戶

任務	描述	所需技能
驗證帳戶。	在 AWS Control Tower 中，選擇帳戶。您剛剛註冊的帳戶的初始狀態為註冊。註冊完成後，其狀態會變更為 [已註冊]。	AWS Control Tower 管理員，成員帳戶管理員

任務	描述	所需技能
檢查護欄是否違犯。	OU 中定義的護欄將自動套用至註冊的成員帳戶。監控 AWS Control 塔儀表板是否有違規情況，並相應地修復它們。如需詳細資訊，請參閱 AWS 文件中的 AWS Control Tower 中的護欄。	AWS Control Tower 管理員，成員帳戶管理員

故障診斷

問題	解決方案
您收到錯誤訊息：發生未知的錯誤。請稍後再試一次，或聯絡 AWS Support。	當您在 AWS Control Tower 中使用 root 使用者登入資料 (管理帳戶) 註冊新帳戶時，就會發生此錯誤。AWS Service Catalog 無法將 Account Factory 產品組合或產品對應至根使用者，因此會產生錯誤訊息。若要修正此錯誤，請使用非根、完整存取權的使用者 (管理員) 認證來註冊新帳戶。如需如何將管理存取權指派給管理使用者的詳細資訊，請參閱 AWS IAM 身分中心入門 (AWS Single Sign-On 的後續任務) 文件。
AWS Control Tower 活動頁面會顯示「取得災難性漂移」動作。	此動作會反映服務的漂移檢查，並不表示 AWS Control Tower 設定有任何問題。無需採取任何動作。

相關資源

文件

- AWS Organizations 術語和概念 (AWS Organizations 文件)
- 什麼是 AWS Control Tower？ (AWS Control Tower 文件)
- 從組織移除成員帳戶 (AWS Organizations 文件)

- 在 AWS Control Tower 中建立管理員帳戶 (AWS Control Tower 文件)

教學課程和影片

- AWS Control Tower 工作坊 (自主進度研討會)
- 什麼是 AWS Control Tower？ (影片)
- 在 AWS Control Tower 中佈建使用者 (影片)
- 為現有組織啟用 AWS Control Tower (影片)

監控跨多個 AWS 帳戶共用 Amazon 機器映像的使用

由納文·薩塔爾（AWS）和桑迪普·蓋萬德（AWS）創建

代碼存儲庫：cross-account-ami-auditing-地形樣本	環境：PoC 或試點	技術：管理與治理 DevOps；無伺服器；營運
AWS 服務：Amazon DynamoDB；AWS Lambda；Amazon EventBridge		

Summary

Amazon 機器映像（AMI）用於在 Amazon 網絡服務（AWS）環境中創建亞馬遜彈性計算雲（亞馬遜 EC2）實例。您可以在單獨的集中式 AWS 帳戶中建立 AMI，此帳戶在此模式中稱為建立者帳戶。然後，您可以在同一個 AWS 區域中的多個 AWS 帳戶共用 AMI，這些帳戶在此模式中稱為消費者帳戶。從單一帳戶管理 AMI 可提供擴充性並簡化控管。在消費者帳戶中，您可以參考 Amazon EC2 自動擴展啟動範本和 Amazon Elastic Kubernetes Service (Amazon EKS) 節點群組中的共用 AMI。

當共用 AMI 遭到淘汰、取消註冊或取消共用時，在消費者帳戶中參考 AMI 的 AWS 服務將無法使用此 AMI 啟動新執行個體。相同執行個體的任何 auto 縮放事件或重新啟動都會失敗。這可能會導致生產環境中的問題，例如應用程式停機或效能降低。當 AMI 共用和使用事件發生在多個 AWS 帳戶中時，可能很難監控此活動。

此模式可協助您監控相同區域中帳戶之間的共用 AMI 使用情況和狀態。它使用無伺服器 AWS 服務，例如 Amazon EventBridge、亞 Amazon DynamoDB、AWS Lambda 和 Amazon Simple Email Service (Amazon SES)。您可以使用 HashiCorp Terraform 將基礎架構佈建為程式碼 (IaC)。當消費者帳戶中的服務參考已取消註冊或未共用的 AMI 時，此解決方案會提供警示。

先決條件和限制

先決條件

- 兩個或多個有效 AWS 帳戶：一個創作者帳戶和一個或多個消費者帳戶
- 從建立者帳戶共用至消費者帳戶的一或多個 AMI
- 地形 CLI，已安裝（地形文檔）
- 地形 AWS 供應商，已設定 (地形文件)

- （可選，但建議使用）地形後端，已配置（地形文檔）
- Git，已安裝

限制

- 此病毒碼會使用帳戶 ID 監控已共用至特定帳戶的 AMI。此模式不會使用組織 ID 監視已共用給組織的 AMI。
- AMI 只能共用至相同 AWS 區域內的帳戶。此病毒碼會監控單一目標區域內的 AMI。若要監控多個區域中 AMI 的使用情況，請在每個區域部署此解決方案。
- 此病毒碼不會監控在部署此解決方案之前共用的任何 AMI。如果您想要監視先前共用的 AMI，您可以取消共用 AMI，然後與消費者帳戶重新共用。

產品版本

- 地形版本 1.2.0 或更高版本
- 地形 AWS 供應商 4.20 版或更新版本

架構

目標技術堆疊

以下資源被佈建為 IaC 通過地形：

- Amazon DynamoDB 資料表
- Amazon EventBridge 規則
- AWS Identity and Access Management (IAM) 角色
- AWS Lambda 函數
- Amazon SES

目標架構

該圖顯示以下工作流程：

1. 建立者帳戶中的 AMI 會與相同 AWS 區域中的消費者帳戶共用。

2. 共用 AMI 時，建立者帳戶中的 Amazon EventBridge 規則會擷取`ModifyImageAttribute`事件，並在建立者帳戶中啟動 Lambda 函數。
3. Lambda 函數會將與 AMI 相關的資料儲存在建立者帳戶的 DynamoDB 表格中。
4. 當消費者帳戶中的 AWS 服務使用共用 AMI 啟動 Amazon EC2 執行個體，或者當共用 AMI 與啟動範本相關聯時，消費者帳戶中的 EventBridge 規則會擷取共用 AMI 的使用情況。
5. 此 EventBridge 規則會啟動消費者帳戶中的 Lambda 函數。Lambda 函數會執行下列動作：
   a. Lambda 函數會更新消費者帳戶中 DynamoDB 表格中與 AMI 相關的資料。
   b. Lambda 函數會在建立者帳戶中擔任 IAM 角色，並更新建立者帳戶中的 DynamoDB 表格。在`Mapping`表格中，它會建立將執行個體 ID 或啟動範本 ID 對應至其個別 AMI ID 的項目。
6. 在創建者帳戶中集中管理的 AMI 已被棄用，取消註冊或取消共享。
7. 建立者帳戶中的 EventBridge 規則會使用動作擷取`ModifyImageAttribute`或`DeregisterImage`事件，並啟`remove`動 Lambda 函數。
8. Lambda 函數會檢查 DynamoDB 資料表，以判斷是否在任何取用者帳戶中使用 AMI。如果`Mapping`表格中沒有與 AMI 相關聯的執行個體 ID 或啟動範本 ID，則程序已完成。
9. 如果任何執行個體 ID 或啟動範本 ID 與`Mapping`表格中的 AMI 相關聯，則 Lambda 函數會使用 Amazon SES 傳送電子郵件通知給已設定的訂閱者。

工具

AWS 服務

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Email Service (Amazon SES) 可協助您使用自己的電子郵件地址和網域來傳送和接收電子郵件。

其他工具

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。
- Python 是一種通用的計算機編程語言。

代碼存儲庫

此模式的代碼可在 GitHub cross-account-ami-monitoring-terra form 樣本存儲庫中找到。

最佳實務

- 請遵循使用 AWS Lambda 函數的最佳實務。
- 請遵循建置 AMI 的最佳做法。
- 建立 IAM 角色時，請遵循最低權限原則，並授予執行任務所需的最低權限。如需詳細資訊，請參閱 IAM 文件中的授與最低權限和安全性最佳實務。
- 設定 AWS Lambda 函數的監控和警示。如需詳細資訊，請參閱監控 Lambda 函數和疑難排解。

史诗

自訂地形組態檔案

任務	描述	所需技能
建立 AWS CLI 命名的設定檔。	對於建立者帳戶和每個消費者帳戶，請建立一個名為 AWS Command Line Interface (AWS CLI) (AWS CLI) 的設定檔。如需指示，請參閱 AWS 入門資源中心中的設定 AWS CLI。	DevOps 工程師
複製儲存庫。	輸入以下命令。這通過使用 SSH 克隆 cross-account-ami-monitoring-terraform 樣本存儲庫。 GitHub `git clone git@githu b.com:aws-samples/`	DevOps 工程師

任務	描述	所需技能
	`cross-account-ami-monitoring-terraform-samples.git`	
更新提供程序 .tf 文件。	1. 輸入下列命令以導覽至複製儲存庫中的`terraform` 資料夾。 `cd cross-account-ami-monitoring/terrafo rm` 2. 開啟 `provider.tf` 檔案。 3. 更新建立者帳戶和消費者帳戶的 Terraform AWS 供應商組態，如下所示： • 在中`alias`，輸入提供者組態的名稱。 • 對於`region`，輸入您要在其中部署此解決方案的目標 AWS 區域。 • 對於`profile`，輸入用於存取帳戶的 AWS CLI 命名設定檔。 4. 如果您要設定一個以上的消費者帳戶，請為每個額外的消費者帳戶建立設定檔。 5. 儲存並關閉 `provider.tf` 檔案。 如需有關設定提供者的詳細資訊，請參閱 Terraform 文件中的多個提供者組態。	DevOps 工程師

任務	描述	所需技能
更新地界形 .tfvars 檔案。	1. 開啟 `terraform.tfvars` 檔案。 2. 在`account_email_mapping` 參數中，設定建立者帳戶和消費者帳戶的警示，如下所示： • 對於`account`，輸入帳戶 ID。 • 對於`email`，輸入您要傳送警示的電子郵件地址。每個帳戶只能輸入一個電子郵件地址。 3. 如果您要設定多個消費者帳戶，請為每個額外的消費者帳戶輸入一個帳戶和電子郵件地址。 4. 儲存並關閉 `terraform.tfvars` 檔案。	DevOps 工程師

任務	描述	所需技能
更新主要 .tf 檔案。	只有在將此解決方案部署到多個用戶帳戶時，才完成這些步驟。如果您只將此解決方案部署到一個消費者帳戶，則不需要修改此檔案。 1. 開啟 `main.tf` 檔案。 2. 對於每個額外的消費者帳戶，請根據模板中的`consumer_account_A` 模塊創建一個新模塊。對於每個消費者帳戶而言`provider`，值應與您在`provider.tf` 檔案中輸入的別名相符。 3. 儲存並關閉 `main.tf` 檔案。	DevOps 工程師

使用地形部署解決方案

任務	描述	所需技能
部署解決方案。	在 Terraform CLI 中，輸入下列命令，在建立者和消費者帳戶中部署 AWS 資源： 1. 輸入以下命令來初始化地形。 `terraform init` 2. 輸入下列指令以驗證地形組態。	DevOps 工程師

任務	描述	所需技能
	`terraform validate` 3. 輸入下列命令以建立 Terraform 執行計畫。 `terraform plan` 4. 檢閱 Terraform 計劃中的組態變更，並確認您要實作這些變更。 5. 輸入以下命令以部署資源。 `terraform apply`	
驗證電子郵件地址身份。	部署 Terraform 方案時，Terraform 會在 Amazon SES 中為每個消費者帳戶建立電子郵件地址身分。您必須先驗證電子郵件地址，才能將通知傳送到該電子郵件地址。如需指示，請參閱 Amazon SES 文件中的驗證電子郵件地址身分。	一般 AWS

驗證資源部署

任務	描述	所需技能
驗證建立者帳戶中的部署。	1. 登入建立者帳戶。 2. 在導覽列中，確認是否正在檢視目標「區域」。如果您位於不同的區域，請選擇目前顯示的區域名稱，然後選擇目標地區。	DevOps 工程師

任務	描述	所需技能
	3. 請在 https://console.aws.amazon.com/dynamodb/ 開啟 DynamoDB 主控台。 4. 在導覽窗格中，選擇 Tables (資料表)。 5. 在表格清單中，驗證資料`AmiShare`表是否存在。 6. 開啟 Lambda 主控台，網址為 https://console.aws.amazon.com/lambda。 7. 在導覽視窗中，選擇函數。 8. 在函數清單中，驗證`ami-share` 函數是否存在。 9. 在 https://console.aws.amazon.com/iamv2/ 開啟身分與存取管理主控台。 10.在導覽窗格中，選擇角色。 11.在角色清單中，驗證角色是否存在。`external-ddb-role` 12.請在以下位置開啟 EventBridge 主控台。 https://console.aws.amazon.com/events/ 13.在導覽窗格中，選擇 Rules (規則)。 14.在規則清單中，驗證規`modify_image_attribute_event` 則是否存在。 15.在以下位置打開 Amazon SES 控制台 https:/co	

任務	描述	所需技能
	nsole.aws.amazon.com/ses/。 16.在功能窗格中，選擇 [已驗證身分]。 17.在身分清單中，驗證每個消費者帳戶的電子郵件地址身份是否已註冊並驗證。	

任務	描述	所需技能
驗證消費者帳戶中的部署。	1. 登入消費者帳戶。 2. 在導覽列中，確認是否正在檢視目標「區域」。如果您位於不同的區域，請選擇目前顯示的區域名稱，然後選擇目標地區。 3. 請在 https://console.aws.amazon.com/dynamodb/ 開啟 DynamoDB 主控台。 4. 在導覽窗格中，選擇 Tables (資料表)。 5. 在表格清單中，驗證資料`Mapping`表是否存在。 6. 開啟 Lambda 主控台，網址為 https://console.aws.amazon.com/lambda。 7. 在導覽視窗中，選擇函數。 8. 在函數清單中，驗證 `ami-usage-function` 和`ami-deregister-function` 函數是否存在。 9. 請在以下位置開啟 EventBridge 主控台。 https://console.aws.amazon.com/events/ 10.在導覽窗格中，選擇 Rules (規則)。 11.在規則清單中，驗證`ami_usage_events` 和`ami_deregister_events` 規則是否存在。	DevOps 工程師

驗證監控

任務	描述	所需技能
在創建者帳戶中創建 AMI。	1. 在創建者帳戶中，創建一個私人 AMI。如需指示，請參閱從 Amazon EC2 執行個體建立 AMI。 2. 與其中一個消費者帳戶共享新的 AMI。如需指示，請參閱與特定 AWS 帳戶共用 AMI。	DevOps 工程師
在消費者帳戶中使用 AMI。	在消費者帳戶中，使用共用 AMI 建立 EC2 執行個體或啟動範本。如需指示，請參閱如何從自訂 AMI 啟動 EC2 執行個體 (AWS RE: POST 知識中心) 或如何建立啟動範本 (Amazon EC2 Auto Scaling 文件)。	DevOps 工程師
驗證監控和警示。	1. 登入建立者帳戶。 2. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。 3. 在導覽窗格中，選擇 AMIs (AMI)。 4. 選取清單中的 AMI，然後選擇 [動作]、[編輯 AMI 權限]。 5. 在 [共用帳戶] 區段中，選取消費者帳戶，然後選擇 [移除選取項目]。 6. 選擇儲存變更。	DevOps 工程師

任務	描述	所需技能
	7. 驗證您為消費者帳戶定義的目標電子郵件地址是否會收到 AMI 已取消共用的通知。	

(選擇性) 停止監視共用 AMI

任務	描述	所需技能
刪除資源。	1. 輸入下列命令以移除此病毒碼部署的資源，並停止監視共用 AMI。 `terraform destroy` 2. 輸入以確認`destroy`指令`yes`。	DevOps 工程師

故障診斷

問題	解決方案
我沒有收到電子郵件提示。	未傳送 Amazon SES 電子郵件的原因可能有多種。請檢查以下內容： 1. 在 Epics 區段中，使用驗證資源部署史詩來確認基礎設施已在所有 AWS 帳戶中正確佈建。 2. 驗證 Amazon CloudWatch 日誌中的 Lambda 函數事件。如需指示，請參閱 Lambda 文件中的使用 CloudWatch 主控台。確認沒有權限問題，例如在任何以身分識別為基礎或以資源為基礎的策略中明確拒絕。如需詳細資訊，請參閱 IAM 文件中的政策評估邏輯。

問題	解決方案
	3. 在 Amazon SES 中，驗證電子郵件地址身分的狀態是否為「已驗證」。如需詳細資訊，請參閱驗證電子郵件地址身分。

相關資源

AWS 文件

- 使用 Python 建置 Lambda 函數 (Lambda 文件)
- 創建一個 AMI (Amazon EC2 文檔)
- 與特定 AWS 帳戶共用 AMI (Amazon EC2 文件)
- 取消註冊您的 AMI (Amazon EC2 文件)

地形文件

- 安裝地形
- 地形後端配置
- 地形 AWS 供應商
- 地形二進制下載

在 AWS Organizations 中設定程式化帳戶關閉的提醒

創建者：理查德·米爾納瓦特 (AWS)、德波吉巴德拉 (AWS) 和馬納夫·亞達夫 (AWS)

程式碼儲存庫：AWS 帳戶關閉通知程式	環境：生產	技術：管理與治理
AWS 服務：AWS CloudTrail；Amazon EventBridge；AWS Lambda；AWS Organizations；Amazon SNS		

Summary

適用於 AWS Organizations 的 CloseAccount API 可讓您以程式設計方式關閉組織內的成員帳戶，而無需使用根登入資料登入帳戶。RemoveAccountFromOrganization API 會從 AWS Organizations 中的組織中提取帳戶，因此它會成為獨立帳戶。

這些 API 可能會增加可關閉或移除 AWS 帳戶的操作員數量。在 AWS Organizations 管理帳戶中透過 AWS Identity and Access Management (IAM) 存取組織的所有使用者都可以呼叫這些 API，因此不僅限於具有任何關聯多因素身份驗證 (MFA) 裝置的帳戶根電子郵件擁有者存取權限。

此模式會在呼叫`CloseAccount`和 `RemoveAccountFromOrganization` API 時實作警示，因此您可以監視這些活動。對於提醒，它使用 Amazon 簡單通知服務（Amazon SNS）主題。您也可以透過網路掛接設定 Slack 通知。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Organizations 中的組織
- 在組織根目錄下存取組織管理帳戶，以建立必要的資源

限制

- 如 AWS Organizations API 參考資料所述，CloseAccountAPI 只允許在滾動 30 天內關閉 10% 的作用中成員帳戶。
- 關閉 AWS 帳戶時，其狀態會變更為「已暫停」。在此狀態轉換後的 90 天內，AWS Support 可以重新開啟帳戶。90 天後，帳戶將被永久刪除。
- 有權存取 AWS Organizations 管理帳戶和 API 的使用者也可能擁有停用這些提醒的許可。如果主要考慮的是惡意行為，而不是意外刪除，請考慮使用 IAM 許可邊界來保護此模式所建立的資源。
- API 會在美國東部 (維吉尼亞北部) 區域 (us-east-1) 呼叫CloseAccount並RemoveAccountFromOrganization處理。因此，您必須部署此解決us-east-1方案才能觀察事件。

架構

目標技術堆疊

- AWS Organizations
- AWS CloudTrail
- Amazon EventBridge
- AWS Lambda
- Amazon SNS

目標架構

下圖顯示此模式的解決方案架構。

1. AWS Organizations 會處理CloseAccount或RemoveAccountFromOrganization請求。
2. Amazon EventBridge 與 AWS 整合，可 CloudTrail 將這些事件傳送到預設事件匯流排。
3. 自訂 Amazon EventBridge 規則與 AWS Organizations 請求相符，並呼叫 AWS Lambda 函數。
4. Lambda 函數會向 SNS 主題傳送訊息，使用者可以訂閱該 SNS 主題以接收電子郵件警示或進一步處理。
5. 如果啟用了 Slack 通知，Lambda 函數會將訊息傳送至 Slack 網路掛鉤。

工具

AWS 服務

- AWS 透過將基礎設施視為程式碼來 CloudFormation提供一系列相關 AWS 和第三方資源的模型、快速且一致地佈建它們，以及在整個生命週期中管理這些資源的方法。
- Amazon EventBridge 是一種無伺服器事件匯流排服務，可用來連接應用程式與來自各種來源的資料。 EventBridge 接收事件、環境變更的指示器，並套用規則以將事件路由至目標。規則會根據事件結構 (稱為事件模式) 或排程，將事件與目標進行比對。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只會在需要時執行您的程式碼，並自動擴展，每天從幾個請求擴展到每秒數千個。您只需為使用的運算時間支付費用。程式碼未執行時無須付費。
- AWS Organizations 可協助您集中管理和控管您的環境，同時擴展和擴展 AWS 資源。使用 AWS Organizations Organization，您可以透過程式設計方式建立新的 AWS 帳戶和分配資源、將帳戶分組以組織工作流程、將政策套用至帳戶或群組以進行管理，以及針對所有帳戶使用單一付款方式簡化帳單。
- AWS CloudTrail 監控和記錄整個 AWS 基礎設施的帳戶活動，並讓您控制儲存、分析和修復動作。
- 亞馬遜簡單通知服務 (Amazon SNS) 是用於 application-to-application (A2A) 和 application-to-person (A2P) 通訊的全受管簡訊服務。

其他工具

- 適用於 Python 程式庫的 AWS Lambda Powertools 是一組公用程式，可為 Lambda 函數提供追蹤、記錄、指標和事件處理功能。

Code

此模式的程式碼位於 GitHub AWS 帳戶更緊密的通知程式儲存庫中。

該解決方案包括為此 CloudFormation 模式部署架構的模板。它使用適用於 Python 程式庫的 AWS Lambda 工具來提供記錄和追蹤。

史诗

部署架構

任務	描述	所需技能
啟動解決方案堆疊的 CloudFormation 範本。	此 CloudFormation 模式的模板位於GitHub 存儲庫的主分支中。它會部署 IAM 角色、EventBridge 規則、Lambda 函數和 SNS 主題。 若要啟動範本： 1. 複製存GitHub 放庫以取得解決方案程式碼的副本。 2. 開啟 AWS 組織管理帳戶的 AWS 管理主控台。 3. 選擇美國東部 (維吉尼亞北部) 區域 (`us-east-1`)，然後開啟主CloudFormation 控台。 4. 使用`account-closure-notifier.yml` 範本並指定下列值來建立堆疊： • 堆疊名稱：`aws-account-closure-notifier-stack` • `ResourcePrefix` 參數：`aws-account-closure-notifier` • `SlackNotification` 參數：如果需要 Slack 通知，請將此設定變更為`true`。	AWS 管理員

任務	描述	所需技能
	• `SlackWebhookEndpoint` 參數：如果需要 Slack 通知，請指定網路掛接網址。 如需啟動 CloudFormation 堆疊的詳細資訊，請參閱 AWS 文件。	
確認解決方案已成功啟動。	1. 等待 CloudFormation 堆疊達到「建立 _ 完成」狀態。 2. 在中開啟EventBridge 主控台`us-east-1` 。 3. 確認已使用名稱建立新規則`aws-account-closure-notifier-event-rule` 。	AWS 管理員

任務	描述	所需技能
訂閱 SNS 主題。	(選擇性) 如果您要訂閱 SNS 主題： 1. 在中開啟 Amazon SNS 主控台us-east-1 ，然後找到名為的主題aws-account-closure-notifier-sns-topic 。 2. 選擇主題名稱，然後選擇 [建立訂閱]。 3. 對於通訊協定，選擇電子郵件。 4. 針對 Endpoint，指定應接收通知的電子郵件地址，然後選擇 [建立訂閱]。 5. 檢查您的電子郵件收件匣是否有來自 AWS 通知的訊息。使用此電子郵件中的連結確認訂閱。 如需有關設定 SNS 通知的詳細資訊，請參閱 Amazon SNS 文件。	AWS 管理員

驗證解決方案

任務	描述	所需技能
將測試事件傳送至預設事件匯流排。	GitHub 存放庫提供範例事件，您可以傳送至 EventBridge 預設事件匯流排以進行測試。此 EventBridge 規則也會對使用自訂事件來源account.c	AWS 管理員

任務	描述	所需技能
	losure.notifier 的事件做出反應。 注意：您無法使用 CloudTrail 事件來源傳送此事件，因為無法以 AWS 服務的形式傳送事件。 若要傳送測試事件： 1. 在中開啟EventBridge 主控台us-east-1 。 2. 在導覽窗格的 [匯流排] 下，選擇 [事件匯流排]，然後選取預設事件匯流排。 3. 選擇 [傳送事件]。 4. 針對事件來源，輸入account.closure.notifier 。 5. 對於詳細資訊類型，請輸入 AWS API Call via CloudTrail 。 6. 對於事件詳細資訊，請將 GitHub 儲存庫tests/dummy-event.json 中的內容複製並貼到文字方塊中。 7. 選擇「傳送」以啟動通知工作流程。	
確認已收到電子郵件通知。	檢查訂閱 SNS 主題的信箱是否有通知。您應該會收到一封電子郵件，其中包含已關閉帳戶的詳細資訊，以及執行 API 呼叫的主體。	AWS 管理員

任務	描述	所需技能
確認已收到 Slack 通知。	(選擇性) 如果您在部署 CloudFormation 範本時為`SlackWebhookEndpoint` 參數指定了 Webhook URL，請檢查對應至 Webhook 的 Slack 通道。它應該顯示一條消息，其中包含已關閉的帳戶和執行 API 調用的主體的詳細信息。	AWS 管理員

相關資源

- CloseAccount 動作 (AWS Organizations API 參考)
- RemoveAccountFromOrganization 動作 (AWS Organizations API 參考)
- AWS Lambda 電動工 Python

更多模式

- 自動化 AWS 資源評估
- 使用 AWS CDK 自動化 AWS 服務目錄產品組合和產品部署
- 使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔
- 自動加密現有和新的 Amazon EBS 磁碟區
- 集中式記錄和多帳戶安全防護
- 啟動時檢查 EC2 執行個體是否有強制標籤
- 為雲端作業模式建立 RACI 或 RASCI 矩陣
- 使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統
- 使用 AWS CloudFormation 安全防護政策建立 AWS 組態自訂規則
- 自動建立基於標籤的 Amazon CloudWatch 儀表板
- 使用 AWS Config 和 AWS Systems Manager 刪除未使用的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區
- 使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制 CloudFormation
- 使用地形表單部署和管理 AWS Control Tower 控制
- 使用 AWS CodePipeline、AWS 和 AWS 在多個 AWS 區域部署程式碼 CodeCommit CodeBuild
- 使用以下方式匯出 AWS IAM 身分中心身分及其指派的報告 PowerShell
- 使用對流圈產生包含 AWS 組態受管規則的 AWS CloudFormation 範本
- 讓 SageMaker 筆記本執行個體暫時存取另一個 AWS 帳戶中的 CodeCommit 儲存庫
- 使用 Step Functions 函數和 Lambda 代理函數在 AWS 帳戶之間啟動 CodeBuild 專案
- 使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器
- 監控 IAM 根使用者活動
- ???
- 在多帳戶 VPC 設計中保留非工作負載子網路的可路由 IP 空間
- 使用 Amazon SES 使用單一電子郵件地址註冊多個 AWS 帳戶
- 輪換資料庫認證而不重新啟動
- 使用現場部署 SMTP 伺服器和資料庫郵件傳送 Amazon RDS for SQL Server 服器資料庫執行個體的通知
- 為 AWS 設定 Grafana 監控儀表板 ParallelCluster

- 使用 AWS Organizations 自動標記 Transit Gateway 附件
- 使用 BMC 探索查詢擷取移轉資料以進行移轉規劃
- 使用 Amazon 將所有 AWS 帳戶的 IAM 登入資料報告視覺化 QuickSight

訊息與通訊

主題

- 在 Amazon MQ 中自動化 RabbitMQ 組態
- 改善 Amazon Connect 聯絡中心代理工作站的通話品質
- 更多模式

在 Amazon MQ 中自動化 RabbitMQ 組態

創建者：瑜伽什·巴蒂亞（AWS）和阿弗羅茲汗（AWS）

環境：PoC 或試點	技術：訊息與通訊 DevOps；基礎架構	AWS 服務：Amazon MQ；AWS CloudFormation

Summary

Amazon MQ 是受管訊息代理程式服務，可與許多熱門訊息代理程式提供相容性。將 Amazon MQ 與 RabbitMQ 搭配使用，可提供在 Amazon Web Services (AWS) 雲端管理的強大 RabbitMQ 叢集，並提供多個代理程式和組態選項。Amazon MQ 提供高可用性、安全且可擴展的基礎設施，而且每秒可輕鬆處理大量訊息。多個應用程式可以將基礎結構與不同的虛擬主機、佇列和交換器搭配使用。不過，管理這些組態選項或手動建立基礎結構可能需要時間和精力。此模式描述了一種通過單個文件管理 RabbitMQ 配置的方法。您可以在任何持續集成（CI）工具（如詹金斯或 Bamboo）中嵌入此模式提供的代碼。

您可以使用此模式來配置任何 RabbitMQ 叢集。它所需要的只是連線到叢集。雖然還有許多其他方法可以管理 RabbitMQ 配置，但此解決方案只需一個步驟即可創建整個應用程序配置，因此您可以輕鬆管理隊列和其他詳細信息。

先決條件和限制

先決條件

- 已安裝並設定 AWS Command Line Interface (AWS CLI) (AWS CLI) 以指向您的 AWS 帳戶 (如需指示，請參閱 AWS CLI 文件)
- 安裝了 Ansible，所以你可以運行教戰手冊來創建配置
- 安裝的兔子管理員（有關說明，請參閱 RabbitMQ 文檔）
- Amazon MQ 中的 RabbitMQ 叢集，使用健康的 Amazon 指標建立 CloudWatch

其他需求

- 確保分別為虛擬主機和用戶創建配置，而不是 JSON 的一部分。
- 請確定組態 JSON 是存放庫的一部分，並且是版本控制的。

- 兔子管理員 CLI 的版本必須與 RabbitMQ 服務器的版本相同，因此最好的選擇是從 RabbitMQ 控制台下載 CLI。
- 作為管道的一部分，請確保在每次運行之前驗證 JSON 語法。

產品版本

- AWS CLI 2.0 版
- 安智版本
- 兔子管理員版本 3.9.13 (必須與 RabbitMQ 伺服器版本相同)

架構

源, 技術, 堆棧

- 在現有內部部署虛擬機器 (VM) 或 Kubernetes 叢集 (在內部部署或雲端中) 上執行的 RabbitMQ 叢集

目標技術堆疊

- 適用於兔子 MQ 的 Amazon MQ 上的自動兔子 MQ 組態

目標架構

有很多方法可以配置 RabbitMQ。此模式使用匯入組態功能，其中單一 JSON 檔案包含所有組態。此文件應用所有設置，並且可以通過版本控制系統（例如 Bitbucket 或 Git）進行管理。這種模式使用 Ansible 通過兔子管理員 CLI 來實現配置。

工具

工具

- Rabbitmqadmin 是一個適用於基於 HTTP 的 API 的命令列工具。它用於管理和監視 RabbitMQ 節點和集群。
- Ansible 是用於自動化應用程式和 IT 基礎架構的開放原始碼工具。
- AWS CLI 可讓您使用命令列殼層中的命令與 AWS 服務互動。

AWS 服務

- Amazon MQ 是一種受管訊息代理程式服務，可讓您輕鬆在雲端中設定和操作訊息代理程式。
- AWS 可 CloudFormation協助您設定 AWS 基礎設施，並使用基礎設施即程式碼加速雲端佈建。

Code

附件中提供了此模式中使用的 JSON 配置文件和示例 Ansible 教戰手冊。

史诗

建立您的 AWS 基礎設施

任務	描述	所需技能
在 AWS 上建立 RabbitMQ 叢集。	如果您還沒有 RabbitMQ 叢集，可以使用 AWS 在 AWS 上 CloudFormation建立堆疊。或者，您可以使用 Ansible 中的雲形模塊來創建堆棧。使用後一種方法，您可以將 Ansible 用於兩項任務：創建 RabbitMQ 基礎結構並管理配置。	AWS CloudFormation，安易

建立適用於兔子 MQ 組態的 Amazon MQ

任務	描述	所需技能
建立屬性檔案。	下載附件中的 JSON 配置文件（`rabbitmqconfig.json` ），或從 RabbitMQ 控制台導出它。 修改它以設定佇列、交換和繫結。此配置文件演示了以下內容：	JSON

任務	描述	所需技能
	-建立兩個佇列：`sample-queue1` 和 `sample-queue2` -創建兩個交易所：`sample-exchange1` 和 `sample-exchange2` -實現隊列和交換之間的綁定 這些設定會根據 rabbitmqadmin 的要求，在根 (/) 虛擬主機下執行。	
擷取適用於 RabbitMQ 基礎設施的 Amazon MQ 詳細資訊。	在 AWS 上擷取 RabbitMQ 基礎設施的下列詳細資訊： • 代理程式名稱 • 兔子 MQ 主機 • RabbitMQ 使用者名稱 (叢集建立期間建立的管理員使用者) • 兔子 MQ 密碼 您可以使用 AWS 管理主控台或 AWS CLI 擷取此資訊。這些詳細資料可讓 Ansible 教戰手冊連線到您的 AWS 帳戶，並使用 RabbitMQ 叢集執行命令。 重要事項：執行 Ansible 教戰手冊的電腦必須能夠存取您的 AWS 帳戶，而且必須已經設定 AWS CLI，如先決條件一節所述。	AWS CLI，Amazon MQ

任務	描述	所需技能
建立主機檔案。	為 Ansible 創建hosts_var 文件，並確保所有變量都在文件中定義。考慮使用 Ansible 保管庫來存儲密碼。您可以按照以下方式配置hosts_var 文件（用您的信息替換星號）： `RABBITMQ_HOST: "**********.mq.us-east-2.amazonaws.com"` `RABBITMQ_VHOST: "/"` `RABBITMQ_USERNAME: "admin"` `RABBITMQ_PASSWORD: "*******"`	Ansible
創建一個安全的教戰手冊。	如需教戰手冊範例，請參閱附件ansible-rabbit-config.yaml 中的。下載並儲存此檔案。Ansible 教戰手冊導入和管理應用程序需要的所有 RabbitMQ 配置，例如隊列，交換和綁定。 請遵循 Ansible 教戰手冊的最佳做法，例如保護密碼。使用 Ansible 保險庫進行密碼加密，並從加密文件中檢索 RabbitMQ 密碼。	Ansible

部署組態

任務	描述	所需技能
執行教戰手冊。	運行您在上一個史詩中創建的 Ansible 劇本。 `ansible-playbook ansible-rabbit-config.yaml` 您可以在 RabbitMQ 控制台上驗證新的配置。	兔子 MQ, Amazon MQ, 安易

相關資源

- 從兔子 MQ 遷移到 Amazon MQ (AWS 部落格文章)
- 管理命令列工具 (RabbitMQ 文件)
- 建立或刪除 AWS CloudFormation 堆疊 (Ansible 文件)
- 將訊息驅動的應用程式遷移到適用於 RabbitMQ 的 Amazon MQ (AWS 部落格文章)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

改善 Amazon Connect 聯絡中心代理工作站的通話品質

由歐尼斯特·奧茲多巴 (AWS) 創建

環境：生產	技術：訊息與通訊；終端使用者運算	AWS 服務：Amazon Connect

Summary

通話品質問題是客服中心疑難排解的一些最困難的問題。若要避免語音品質問題和複雜的疑難排解程序，您必須最佳化專員的工作環境和工作站設定。此模式描述 Amazon Connect 聯絡中心中代理程式工作站的語音品質最佳化技術。它提供了以下幾個方面的建議：

- 工作環境調整。專員的周圍環境不會影響通過網絡傳輸語音的方式，但它們確實會影響通話質量。
- 代理程式工作站設定 客服中心工作站的硬件和網絡配置對通話質量有顯著的影響。
- 瀏覽器設定。客服人員使用網頁瀏覽器存取 Amazon Connect 聯絡人控制台 (CCP) 網站並與客戶通訊，因此瀏覽器設定可能會影響通話品質。

下列元件也可能會影響通話品質，但是它們不在工作站的範圍內，而且不包含在此模式中：

- 流量透過 AWS 直接連接、全通道 VPN 或分割通道 VPN 流向 Amazon Web Services 務 (AWS) 雲端
- 在公司辦公室內外工作時的網絡狀況
- 公共交換電話網絡（PSTN）連接
- 客戶的設備和電話運營商
- 虛擬桌面基礎架構 (VDI) 設定

如需這些區域的詳細資訊，請參閱 Amazon Connect 文件中的常見聯絡人控制台 (CCP) 問題和使用端點測試公用程式。

先決條件和限制

先決條件

- 頭戴式裝置和工作站必須符合 Amazon Connect 管理員指南中指定的要求。

限制

- 此模式中的最佳化技術適用於柔和的電話語音品質。當您在桌上型電話模式下設定 Amazon Connect CCP 時，它們不適用。不過，如果您的軟體電話設定無法為通話提供可接受的語音品質，則可以使用桌面電話模式。

產品版本

- 如需支援的瀏覽器和版本，請參閱 Amazon Connect 管理員指南。

架構

此模式與架構無關，因為它以代理程式工作站設定為目標。如下圖所示，從代理程式到客戶的語音路徑會受到客戶的頭戴式裝置、瀏覽器、作業系統、工作站硬體和網路的影響。

在 Amazon Connect 聯絡中心，使用者的音訊連接是透過 WebRTC 建立的。語音使用 Opus 交互式音頻編解碼器進行編碼，並使用傳輸過程中的安全實時傳輸協議（SRTP）進行加密。其他網路架構也是可行的，包括 VPN、私有 WAN/LAN 和 ISP 網路。

工具

- Amazon Connect 端點測試公用程式 — 此公用程式會檢查網路連線和瀏覽器設定。
- WebRTC 技術設定的瀏覽器設定編輯器：
  - 對於火狐：關於：配置
  - 對於鉻：鉻：//標誌
- CCP 記錄剖析器 — 此工具可協助您分析 CCP 記錄，以便進行疑難排解。

史诗

調整工作環境

任務	描述	所需技能
減少背景噪音。	避免嘈雜的環境。如果無法做到這一點，請使用以下隔音提示最佳化環境： • 通過使用消散聲音的表面，例如窗簾，地毯和柔軟的家具來吸收噪音。 • 通過在桌子之間放置屏障來阻止噪音。 • 考慮使用主動降噪（ANC）解決方案，例如白噪發生器，以幫助集中並確保隱私，或使用降噪耳機。 • 防止通話中的回音。大的空白空間可能會產生迴聲效果或放大噪音。覆蓋可以反彈聲音的表面將有助於減少回音。	代理人，經理

優化代理程式工作站

任務	描述	所需技能
選擇合適的耳機。	• 如果環境嘈雜，請選擇立體聲耳機。將聲音導向雙耳，有助於專員更好地集中並聽到客戶的聲音，並減少代理商提高聲音的可能性，從而降低整體噪音。	代理人，經理

任務	描述	所需技能
	• 避免使用大聲的揚聲器或內置電腦音頻。為了獲得最佳品質，請使用客服中心專用的有線耳機。無線耳機很方便，但由於無線電干擾和轉碼，它們可能是額外的音頻延遲和降低音頻質量的來源。	
請依預期使用耳機。	• 啟用頭戴式裝置的主動降噪功能和語音增強功能 (如果可用)。查找諸如 ANC 或 ANR 之類的設置。如需啟用這些設定的指示，請參閱頭戴式裝置的使用者手冊。 • 調整您的麥克風，以便您可以直接對其說話。麥克風的最佳位置就在下巴下方。正確的放置可以使 10 分貝（dB）的差異在聲級。大多數耳機都允許您旋轉或彎曲麥克風臂（懸臂），因此在說話時將其保持在正確的位置非常重要。 • 有些耳機配備了多個麥克風和進階功能，例如語音波束成形，有助於在沒有動臂的情況下擷取語音。若要確定您是依照製造商的預期使用主麥克風，請參閱裝置的使用者手冊。	代理程式

任務	描述	所需技能
檢查工作站資源。	請確定您的代理程式的電腦效能良好。如果他們使用耗用資源的協力廠商應用程式，他們的電腦可能不符合執行 CCP 的最低硬體需求。如果代理程式遇到通話品質問題，請確定他們有足夠的處理能力 (CPU)、磁碟空間、網路頻寬和可供 CCP 使用的記憶體。代理程式應關閉所有不必要的應用程式和標籤，以改善 CCP 效能和通話品質。	管理員

任務	描述	所需技能
設定作業系統的音效設定。	麥克風電平和提升的默認設置通常可以正常工作。如果您發現輸出語音安靜或麥克風接收太多，調整這些設定可能會有所幫助。可以在計算機的系統聲音配置中找到麥克風設置（聲音，MacOS 上的輸入，Windows 中的麥克風屬性）。您可以透過系統工具或第三方應用程式存取可能會影響語音品質的進階設定。以下是您可以檢查的一些設置： • 取樣率 — 此值決定每秒偵測聲音的次數。預設設定通常為 44 或 48 kHz 茲。Amazon Connect 的最佳值是 48 kHz。您可以使用瀏覽器設定來覆寫預設值。如需詳細資訊，請參閱 Amazon Connect 管理員指南中的疑難排解一節。 • 增益 — 此值決定麥克風放大聲音的程度。如果您調高增益，您的麥克風可能會發出更多的背景噪音。 • 位元深度 — 此數位解析度值描述辨識的聲音振幅等級數。位元深度越高，聲音聽起來就越平滑。但是，許多傳統電話語音網路使用脈衝編碼調變 (PCM) 標準，該標準僅支援 8 位元解析度。	代理人，管理員

任務	描述	所需技能
	• 開放閾值-這是麥克風拾取的最小聲音振幅。 如果您遇到語音品質問題，請先嘗試將這些值還原為預設設定，然後再進一步調查。 如需有關這些設定和其他可調整設定的詳細資訊，請參閱裝置手冊。	

任務	描述	所需技能
使用有線網路。	通常，有線乙太網路的延遲時間較低，因此更容易提供語音資料傳輸所需的一致傳輸品質。 我們建議每次通話至少有 100 KB 頻寬。 • 如果客服人員在家工作，我們建議使用無線連線進行有線連線。聽到客戶的聲音不應超過 150 毫秒。您可以從 Amazon Connect 端點測試公用程式存取 Amazon Connect 的延遲測試。但是，此公用程式會測量從瀏覽器到 Amazon Connect 區域的延遲，而非客戶的延遲。150 毫秒的單向延遲建議可防止代理程式與客戶互相交談。該值是從端到端測量的，每個元素都會增加延遲，包括 Amazon Connect 區域和客戶之間的呼叫部分。 • 如果客服人員在辦公室工作，只要參數在建議的範圍內，即可接受企業 Wi-Fi，並優先處理即時傳輸通訊協定 (RTP) 流量。	網路管理員、代理

任務	描述	所需技能
更新硬體驅動程式。	當您使用具有韌體的 USB 或其他類型的頭戴式裝置時，我們建議您將耳機保持最新版本更新。使用輔助連接埠的簡單耳機會使用電腦的內建音訊裝置，因此請確定作業系統硬體驅動程式為最新版本。在極少數情況下，音訊驅動程式更新可能會造成音訊問題，因此您可能需要將其復原。如需有關變更韌體和驅動程式版本的詳細資訊，請參閱裝置手冊。	管理員
避免使用 USB 集線器和加密狗。	連接頭戴式裝置時，請避免使用其他裝置，例如硬體鎖、連接埠類型轉換器、集線器和延長纜線。 這些裝置可能會影響通話品質。將設備直接 Connect 到計算機中的端口。	代理程式

任務	描述	所需技能
檢查 CCP 記錄檔。	CCP 記錄剖析器提供檢查應用程式記錄檔的簡易方法。 1. 通話後下載 CCP 記錄。 2. 開啟 CCP 記錄剖析器。 3. 拖放記錄檔以上傳記錄以供分析。 4. 分析記錄後，預設會選取 [快照與記錄] 索引標籤。選擇其旁邊的「指標」選項卡以查看見解。 5. 在 WebRTC 測量結果-音訊輸入區段中，檢查下列項目： • 音頻電平圖表，以查看您收到的音頻電平是否高於 0。這表示音訊是從您的來電者接收到的。 • 任何遺失封包的封包圖形。如果此圖表顯示大幅增加，請聯絡您的 IT 支援團隊。 6. 在 WebRTC 測量結果-音訊輸出區段中，檢查下列項目： • 「音訊電平」圖表，以確認音訊已從您的裝置傳送出去。 • 「封包」圖形。如果您看到封包遺失尖峰，請向您的 IT 支援團隊報告。 • 抖動緩衝區和 RTT 圖形。往返時間（RTT）值超過	特工（高級技能）

任務	描述	所需技能
	300 將影響通話體驗。將這些報告給您的 IT 支援團隊。	

優化瀏覽器設置

任務	描述	所需技能
恢復默認 WebRTC 技術設置。	WebRTC 技術必須啟用，才能使用 CCP 撥打軟電話。我們建議您保留 WebRTC 相關功能的預設設定。 • 在 Chrome 瀏覽器中，您可以通過導航到網址瀏覽器：// 標誌來設置標誌。在搜尋方塊中輸入 WebRTC 以尋找可能干擾 CCP 的設定，並將這些設定設定為 [預設值]。 • 在 Firefox 中，在網址列中輸入關於:config，然後在設定頁面的搜尋方塊中輸入 WebRTC。非預設設定會以粗體文字顯示，且可變更為「預設」。	管理員
故障排除時禁用瀏覽器擴展。	某些瀏覽器擴充功能可能會影響通話品質，甚至阻止呼叫正確連線。在瀏覽器中使用隱身視窗或私人模式，並停用所有擴充功能。如果這樣可以解決問題，請查看您的瀏覽器擴展程序並查找可疑的附加組件，或單獨禁用它們。	代理人，管理員

任務	描述	所需技能
檢查瀏覽器採樣率。	確認您的麥克風輸入設定為最佳 48 kHz 取樣率。如需指示，請參閱 Amazon Connect 管理員指南。	代理人，管理員

相關資源

如果您已按照此模式中的步驟操作，但仍遇到通話品質問題，請參閱下列資源以取得疑難排解提示。

- 檢閱常見的聯絡人控制台 (CCP) 問題。
- 檢查與端點測試公用程式的連線。
- 請遵循疑難排解指南以解決任何其他問題。

如果您的疑難排解和調整無法解決通話品質問題，則根本原因可能是工作站的外部原因。如需進一步疑難排解，請聯絡您的 IT 支援團隊。

更多模式

- 通過使用 CQRS 和事件採購將巨石分解為微服務
- 將 Amazon API Gateway 與 Amazon SQS 整合以處理異步 REST API
- 使用 Amazon SES 使用單一電子郵件地址註冊多個 AWS 帳戶
- 使用 AWS Fargate 大規模執行訊息導向工作負載

遷移

主題

使用自動化遷移策略識別和規劃 AppScore

環境：生產	來源：所有工作負載	目標：AWS 雲端
R 類型：不適用	工作負載：所有其他工作	技術：遷移；現代化；Web 和移動應用程序；SaaS
AWS 服務：AWS Application Discovery Service；AWS Migration Hub		

Summary

現場部署應用程式需要轉型方法來協助發揮 Amazon Web Services (AWS) 雲端的優勢。七種常見的移轉策略 (7 Rs) 為您提供轉換選項，這些選項不同於在內部部署資料庫伺服器中進行技術變更，以及使用雲端原生微服務架構重建應用程式。

選擇使用完整的 7 Rs 模型意味著您在應用程式和業務層級進行操作，而不是僅評估和準備要移轉的伺服器。雖然您可以使用 AWS Migration Evaluator 等工具取得伺服器資料，但通常不會記錄其他應用程式資訊 (例如藍圖狀態、所需的復原時間目標 (RTO) 和復原點目標 (RPO) 或資料隱私權要求)。

此模式說明如何使用以應用程式為中心的產品組合檢視AppScore來避免這些挑戰。這包括針對完整 7 Rs 模型的每個應用程式建議的 AWS 雲端轉換路徑。 AppScore 協助您擷取應用程式資訊、判斷理想的轉換路徑、識別採用雲端所帶來的風險、複雜性和優點，以及快速定義移轉範圍、移動群組和排程。

此模式是由 AWS 合作夥伴 AWS 和AppScore 技術有限公司所建立。

先決條件和限制

先決條件

- 您想要遷移到 AWS 雲端的現有應用程式。
- 來自 AWS Migration Evaluator 等工具的現有伺服器庫存資訊。您也可以在移轉的稍後階段匯入此資料。
- 具有超級使用者權限的現有 AppScore 帳戶。如需 AppScore 使用者帳戶的詳細資訊，請參閱如何將角色型存取控制 (RBAC) 指派給使用者？ 在 AppScore 文檔中

- 瞭解如何在中指派 RBAC 角色。 AppScore AppScore 提供三個主題專家 (SME) 角色，符合評分階段提出的問題。這意味著中小企業只回答與其專業知識和角色相關的問題。如需相關資訊，請參閱如何將角色型存取控制 (RBAC) 指派給使用者？ 在文 AppScore 檔中。
- 瞭解的 AppScore建議，這些建議是以下列三種應用程式屬性類別為基礎：
  - 風險 — 應用程式的業務重要性，無論是包含機密資料、資料主權要求，以及應用程式使用者或介面的數量
  - 複雜性 — 應用程式的開發語言 (例如，COBOL 的分數高於 .NET 或 PHP)、年齡、使用者介面或介面數量
  - 效益 — 批次處理需求、應用程式設定檔、災難復原模式、開發與測試環境的使用
- 了解迭代數據捕獲的四個階段： AppScore
  - 路標 — 結合伺服器資料以產生 7 Rs 評估的問題。如需詳細資訊，請參閱 AppScore 文件中的如何路標和評分應用程式。
  - 評分 — 為風險、利益和複雜性產生分數的問題。
  - 目前狀態評估 — 提供應用程式目前狀態評估的問題。
  - 轉型 — 全面評估 future 狀態設計應用的問題。

重要提示：只有路標和評分階段才能接收申請分數，7 Rs 評估並啟用小組規劃。將應用程式和表單範圍分組之後，您可以完成「目前狀態評估」和「轉換」階段，以建立更詳細的應用程式概觀。

架構

下圖顯示使用應用 AppScore 程式和伺服器資料建立移轉策略和轉換計劃建議的工作流程。

工具

- AppScore— AppScore 透過針對完整的 7 Rs 模型，為每個應用程式提供以應用程式為中心的產品組合檢視以應用程式為中心的檢視，並針對每個應用程式提供到雲端的建議路徑，協助您彌合
- AWS Migration Evaluator — AWS Migration Evaluator 是一種遷移評估服務，可協助您建立規劃和移轉的方向性商業案例。

史诗

建立並載入初始應用程式清單

任務	描述	所需技能
準備應用程序列表。	使用您的使用者認證登 AppScore 入入口網站。Import Template從「應用程式」頁面下載，然後Import Template使用應用程式的非技術屬性 (例如，資料分類或可自訂的屬性清單) 更新。 如需這方面的詳細資訊，請參閱如何變更文件中的 AppScore 應用程式和商業問卷。 AppScore 注意：您也可以在「應用程式」頁面中選擇新增應用程式，手動新增應用程式。然後，您可以輸入應用程式的非技術屬性。	移民工程師
匯入應用程式資料。	在 [應用程式] 頁面上，選擇 [匯入應用程式] 以匯入應用程式資	移民工程師

擷取應用程式和業務資料

任務	描述	所需技能
查看並回答路標和評分問題。	開啟伺服器頁面，然後選擇匯入伺服器。選擇包含伺服器資料的 .csv 檔案。	應用所有者

任務	描述	所需技能
	該文件可以包括屬性，如名稱，數據中心，操作系統，虛擬或物理，應用程序名稱，角色，數據庫技術，環境，CPU 核心數和利用率，RAM 大小和利用率，磁盤大小和利用率，匹配的機器類型，以及當前和預計每月成本。 確認欄對應，然後選擇 [確認並匯入]。匯入資料中缺少的資訊會在 [伺服器] 頁面上反白顯示。您可以在此頁面上或使用「大量編輯」 選項來解決這些間隙。服務器與相關應用程序相關聯。但是，如果中不存在應用程式 AppScore，則會自動建立這些應用程式，然後將伺服器建立關聯。 您也可以使用 API 連線透過 AWS 遷移中心擷取資料。如需有關這方面的詳細資訊，請參閱如何透過 API 從 AWS Migration Hub 匯入伺服器？在文 AppScore 檔中。 附註：如果您使用探索工具 (例如 AWS Migration Evaluator) 擷取一段時間內的效能，則必須儘快載入伺服器資料的早期擷取，並在完全擷取效能指標時重新整理資料。 AppScore 使用伺服器名稱、作業系統和	

任務	描述	所需技能
	資料庫版本、資料中心和環境來提供分數和 7 Rs 建議。	
檢查應用程序分數。	開啟 [應用程式] 頁面以查看應用程式的分數和 7 Rs 評量。系統也會計算您目前的執行成本。當新資訊匯入「應用程式」或「伺服器」頁面時，會更新這些計算。	應用所有者
分析個別應用程式。	在「應用程式」頁面上選擇應用程式，以檢閱詳細建議。您可以選擇「應用程式評估報告」來產生 .pdf 或 .docx 檔案，其中包含特定應用程式的詳細評估資料。	應用所有者

建立移轉排程

任務	描述	所需技能
選擇移動群組的應用程式。	開啟「計劃」頁面，選擇「群組建構器」，然後根據您的需求建立應用模組移轉群組。 您可以在「欄」區段的應用程式清單中新增或移除屬性。您也可以使用 [篩選器] 區段中的應用程式屬性來選擇特定的應用程式，其中包括篩選掉已經屬於現有移動群組一部分的所有應用程式。	移民工程師
建立移動群組。	選擇 [選取的群組]、輸入移動群組的名稱、選擇要包含在移	移民工程師

任務	描述	所需技能
	動群組中的應用程式，然後選擇 [新增至群組]。	
排程移轉。	在 [轉換排程] 頁面上，AppScore 提供移動群組的預估轉換持續時間、工時和成本。移動群組會自動新增至整體轉換排程中。 備註：您可以在「計劃設定值」頁面中自訂工作量估算背後的假設。這有助於使它們符合您組織的需求。如需相關資訊，請參閱 AppScore 文件中的如何設定規劃設定。	移民工程師
產生完整的轉換報告。	開啟群組管理員頁面，然後選擇建立應用程式轉換報表文件。選擇移動群組，然後選擇 [匯出]。這會產生一個 .docx 檔案，該檔案總結了轉換，包括每個移動群組的詳細資訊。 如需應用程式轉換報表範例，請參閱 AppScore 網站上的範例應用程式轉換報表。	移民工程師

相關資源

- 什麼是應用程序遷移的 7 盧比？
- 仔細看看 AppScore
- AppScore 在 AWS Marketplace 中

使用 Microsoft Excel 和 Python 為 AWS DMS 任務建立 AWS CloudFormation 範本

創建者：文卡塔納溫·科普拉 (AWS)

環境：PoC 或試點	來源：自動化	目標：AWS 雲端中的資料庫
R 類型：不適用	工作量：Microsoft	技術：移轉；資料庫

Summary

此 CloudFormation 模式概述了使用 Microsoft Excel 和 Python 為 AWS 資料庫遷移服務 (AWS DMS) 自動建立 AWS 範本的步驟。

使用 AWS DMS 遷移資料庫通常需要建立 AWS CloudFormation 範本以佈建 AWS DMS 任務。之前，建立 AWS CloudFormation 範本需要具備 JSON 或 YAML 程式設計語言的知識。使用此工具，您只需要 Excel 的基本知識以及如何使用終端或命令窗口運行 Python 腳本。

作為輸入，該工具會採用 Excel 活頁簿，其中包含要遷移的表格名稱、AWS DMS 端點的 Amazon 資源名稱 (ARN)，以及 AWS DMS 複寫執行個體。然後，該工具會針對所需的 AWS DMS 任務產生 AWS CloudFormation 範本。

如需詳細步驟和背景資訊，請參閱 AWS 資料庫部落格中的部落格文章使用 Microsoft Excel 為 AWS DMS 任務建立 AWS CloudFormation 範本。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Microsoft Excel 版本
- Python 版本 2.7 或更高版本
- xlrd Python 模塊（使用命令安裝在命令提示符下：點子安裝 XLRD）
- AWS DMS 來源和目標端點以及 AWS DMS 複寫執行個體

限制

- 結構定義、資料表和相關資料行的名稱會在目的地端點轉換成小寫字元。
- 此工具無法解決 AWS DMS 端點和複寫執行個體的建立問題。
- 目前，該工具僅支援每個 AWS DMS 任務的一個結構描述。

架構

源, 技術, 堆棧

- 內部部署資料庫
- Microsoft Excel

目標技術堆疊

- AWS CloudFormation 範本
- AWS 雲端中的資料庫

架構

工具

- 或任何支援 3.6 版本的整合式開發環境
- Microsoft 辦公室 2016（適用於 Microsoft Excel）

史诗

設定網路、AWS DMS 複寫執行個體和端點

任務	描述	所需技能
如有必要，請求增加服務配額。	如有需要，請求增加 AWS DMS 任務的服務配額。	一般 AWS

任務	描述	所需技能
設定 AWS 區域、虛擬私有雲端 (VPC)、CIDR 範圍、可用區域和子網路。		一般 AWS
設定 AWS DMS 複寫執行個體。	AWS DMS 複寫執行個體可以連接到現場部署和 AWS 資料庫。	一般 AWS
設定 AWS DMS 端點。	設定來源和目標資料庫的端點。	一般 AWS

準備適用於 AWS DMS 任務和標籤的工作表

任務	描述	所需技能
設定表格清單。	列出與移轉相關的所有資料表。	資料庫
準備任務工作表。	使用您設定的表格清單準備 Excel 工作表。	一般 AWS、Microsoft Excel
準備標籤工作表。	詳細說明要連接到 AWS DMS 任務的 AWS 資源標籤。	一般 AWS、Microsoft Excel

下載並執行工具

任務	描述	所需技能
從 GitHub 存放庫下載並擷取範本產生工具。	GitHub 資料庫dms-cloudformation-templates-generator:https://github.com/aws-samples/	

任務	描述	所需技能
運行該工具。	請依照「參考與說明」下方所列部落格文章中的詳細指示進行。	

相關資源

- 使用 Microsoft Excel 建立適用於 AWS DMS 任務的 AWS CloudFormation 範本 (部落格文章)
- DMS CloudFormation 範本產生器 (GitHub 儲存庫)
- Python 文檔
- XLRD 說明和下載
- AWS DMS 說明文件
- AWS CloudFormation 文件

開始使用自動化產品組合探索

創建者：清瓦拉普拉蒂克（AWS）和小魯道夫 塞拉達 (AWS)

環境：生產	來源：內部部署	目標：內部部署
R 類型：不適用	工作負載：所有其他工作	技術：遷移

Summary

將應用程式和伺服器遷移到 Amazon Web Services (AWS) 雲端時，評估產品組合和收集中繼資料是一項重大挑戰，特別是對於擁有 300 部以上伺服器的大型遷移而言。使用自動化產品組合探索工具可協助您收集應用程式的相關資訊，例如使用者數量、使用頻率、相依性，以及應用程式基礎結構的相關資訊。在規劃移轉波時，此資訊非常重要，因此您可以適當地排定優先順序並將具有類似特徵的應用程式使用探索工具可簡化產品組合團隊與應用程式擁有者之間的溝通，因為組合團隊可以驗證探索工具的結果，而不必手動收集中繼資料。此模式討論選取自動化探索工具的關鍵考量事項，以及如何在環境中部署和測試自動化探索工具的相關資訊。

此模式包括一個模板，這是構建自己的高級活動檢查清單的起點。清單旁邊是一個負責任的，負責的，諮詢，知情的（RACI）矩陣的模板。您可以使用此 RACI 矩陣來決定檢查清單中每項任務的負責人。

史诗

選取探索工具

任務	描述	所需技能
判斷探索工具是否適合您的使用案例。	探索工具可能不是您使用案例的最佳解決方案。考慮選取、採購、準備和部署探查工具所需的時間量。在您的環境中為無代理程式探索工具設定掃描應用裝置，或將代理程式安裝到所有範圍內的工作負載，可能需要 4 到 8 週的時間。部署	遷移負責人，移民工程師

任務	描述	所需技能
	之後，您必須等待 4 到 12 週的時間，探索工具透過掃描應用程式工作負載並執行應用程式堆疊分析來收集中繼資料。如果您移轉的伺服器少於 100 部，您或許可以手動收集中繼資料並分析相依性，比使用自動探索工具部署和收集中繼資料所需的時間更快。	
選取探索工具。	在「其他資訊」一節中檢閱選取自動化探索工具的考量事項。決定針對您的使用案例選取探查工具的適當條件，然後根據這些準則評估每個工具。如需自動化探索工具的完整清單，請參閱探索、規劃和建議移轉工具。	遷移負責人，移民工程師

準備安裝

任務	描述	所需技能
準備部署前檢查清單。	建立部署工具前必須完成之工作的檢查清單。如需範例，請參閱 Flexera 文件網站上的部署前檢查清單。	構建負責人，遷移工程師，遷移負責人，網絡管理員
準備網路需求。	佈建工具執行和存取目標伺服器所需的連接埠、通訊協定、IP 位址和路由。如需詳細資訊，請參閱探索工具的安裝指南。如需範例，請參閱 Flexera 文件網站上的部署需求。	移轉工程師、網路管理員、雲端架構師

任務	描述	所需技能
準備帳戶和憑證要求。	識別存取目標伺服器及安裝工具所有元件所需的認證。	雲端管理員, 一般 AWS, 遷移工程師, 遷移負責人, 網路管理員, AWS 管理員
準備要在其上安裝工具的設備。	確定要安裝工具元件的設備符合工具的規格和平台需求。	遷移工程師，遷移負責人，網絡管理員
準備變更命令。	根據組織中的變更管理流程，準備所需的任何變更命令，並確保核准這些變更命令。	建立領導者，遷移領導者
向利益相關者發送要求。	將部署前檢查清單和網路需求傳送給利害關係人。利益相關者應該在繼續部署之前審查，評估和準備必要的要求。	建立領導者，遷移領導者

部署工具

任務	描述	所需技能
下載安裝程式。	下載安裝程式或虛擬機器映像。虛擬機器映像通常採用開放虛擬化格式 (OVF)。	建立領導者，遷移領導者
將檔案解壓縮。	如果您使用的是安裝程式，則必須在內部部署伺服器上下載並執行安裝程式。	建立領導者，遷移領導者
在伺服器上部署工具。	在目標、內部部署伺服器上部署探查工具，如下所示： • 如果來源檔案是虛擬機器映像，請將其部署到虛擬機器環境中，例如 VMware。	構建負責人，遷移負責人，網絡管理員

任務	描述	所需技能
	• 如果您的來源檔案是安裝程式，請執行安裝程式以安裝並設定工具。	
登入探索工具。	按照屏幕上的提示進行操作，然後登錄以開始使用該工具。	遷移領導者，建立領導者
啟用產品。	輸入您的授權金鑰。	建立領導者，遷移領導者
設定工具。	輸入存取目標伺服器所需的任何證明資料，例如 Windows、VMware、簡易網路管理通訊協定 (SNMP) 和安全殼層通訊協定 (SSH) 或資料庫的認證。	建立領導者，遷移領導者

測試工具

任務	描述	所需技能
選取測試伺服器。	識別可用來測試探索工具的一小組非生產子網路或 IP 位址。這有助於您快速驗證掃描，快速識別和疑難排解任何錯誤，並將測試與生產環境隔離。	構建負責人，遷移負責人，網絡管理員
開始掃描選取的測試伺服器。	對於無代理程式探查工具，請在探查工具主控台中輸入所選測試伺服器的子網路或 IP 位址，然後開始掃描。 對於以代理程式為基礎的探查工具，請在所選測試伺服器上安裝代理程式。	構建負責人，遷移負責人，網絡管理員

任務	描述	所需技能
檢閱掃描結果。	檢閱測試伺服器的掃描結果。如果發現任何錯誤，請進行故障排除並修復錯誤。記錄錯誤和解決方案。您在 future 參考此資訊，您可以將此資訊新增至您的投資組合 Runbook。	構建負責人，遷移負責人，網絡管理員
重新掃描測試伺服器。	重新掃描完成後，請重複掃描，直到沒有錯誤為止。	構建負責人，遷移負責人，網絡管理員

相關資源

AWS 資源

- AWS 雲端移轉的應用程式組合評估指南
- 探索、規劃和建議移轉工具

常用探索工具的部署指南

- 部署 RN150 虛擬設備 (文件集)
- 採集者安裝（模型。。它的文檔）
- 內部部署分析伺服器安裝 (模型清單說明文件)

其他資訊

選取自動化探索工具的考量

每個探索工具都有優點和限制。為您的使用案例選取適當的工具時，請考慮下列事項：

- 選擇一個探索工具，該工具可以收集您需要的大部分（如果不是全部）中繼資料，以實現您的投資組合評估目標。
- 識別您需要手動收集的任何中繼資料，因為該工具不支援它。
- 向利害關係人提供探索工具需求，以便他們可以根據其內部安全性和合規性要求 (例如伺服器、網路和認證需求) 來審查和評估工具。

  - 此工具是否需要您在範圍內的工作負載中安裝代理程式？
  - 此工具是否需要您在環境中設定虛擬應用裝置？
- 決定您的資料存放需求。有些組織不想將資料儲存在環境之外。若要解決這個問題，您可能需要在內部部署環境中安裝工具的某些元件。
- 請確定此工具支援範圍內工作負載的作業系統 (OS) 和作業系統版本。
- 判斷您的產品組合是否包含大型主機、中階伺服器和舊版伺服器。大多數探查工具都可以將這些工作負載偵測為相依性，但某些工具可能無法取得裝置詳細資料，例如使用率和伺服器相依性。Device42 和現代化 IT 探索工具都支援大型主機和中階伺服器。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

將現場部署工作負載遷移到 AWS 上的 Cloudera 資料平台

環境：PoC 或試點	來源：雲端工作負載	目標：雲達資料平台 (CDP) 公有雲
R 類型：不適用	工作負載：所有其他工作	技術：移民、大數據、資料庫、分析
AWS 服務：Amazon EC2; Amazon EKS; AWS Identity and Access Management; Amazon S3; Amazon RDS		

Summary

此模式描述了將現場部署 Cloudera 分散式 Hadoop (CDH)、Hortonworks 資料平台 (HDP) 和 Cloudera 資料平台 (CDP) 工作負載遷移到 AWS 上的 CDP 公有雲的高階步驟。我們建議您與 Cloudera 專業服務和系統整合商 (SI) 合作，以實施這些步驟。

Cloudera 客戶希望將其內部部署 CDH、HDP 和 CDP 工作負載移至雲端的原因有很多。一些典型的原因包括：

- 簡化新資料平台範例的採用，例如資料湖房或資料網格
- 提高企業靈活性，使現有資料資產的存取和推論民主化
- 降低總擁有成本 (TCO)
- 增強工作量彈性
- 提供更高的延展性；與舊版的內部部署安裝基礎相比，大幅縮短佈建資料服務的時間
- 淘汰舊版硬體；大幅減少硬體重新整理週期
- 利用 Cloudera 授權模式 (CCU) 將 pay-as-you-go 定價延伸至 AWS 上的 Cloudera 工作負載
- 利用更快的部署和持續整合與持續整合 (CI/CD) 平台
- 針對多個工作負載使用單一整合平台 (CDP)

Cloudera 支援所有主要工作負載，包括 Machine Learning、資料工程、資料倉儲、作業資料庫、串流處理 (CSP) 以及資料安全與控管。Cloudera 已在內部部署提供這些工作負載多年，您可以使用 CDP 公有雲端搭配工作負載管理員和複寫管理員，將這些工作負載遷移到 AWS 雲端。

Cloudera 共用資料體驗 (SDX) 在這些工作負載中提供共用的中繼資料目錄，以促進一致的資料管理和作業。SDX 還包括全面、精細的安全性以防範威脅，以及統一治理，可用於稽核和搜尋功能，以符合支付卡產業資料安全標準 (PCI DSS) 和 GDPR 等標準。

CDP 遷移一目了然

工作量	來源工作量	CDH、亞盤和 CDP 私有雲
	來源環境	• Windows，Linux • 現場部署、主機託管或任何非 AWS 環境
	目標工作量	AWS 上的 CDP 公有雲
	目的地環境	• 部署模式：客戶帳戶 • 經營模式：客戶/雲德拉控制平面
	移轉策略 (7 秒)	重新裝載、重新平台或重構
	這是工作負載版本中的升級嗎？	是
移民	移轉期間	• 部署：大約 1 週的時間來創建客戶帳戶，虛擬私有雲 (VPC) 和 CDP 公共雲客戶管理環境。 • 移轉持續時間：1-4 個月，視工作負載的複雜性和大小而定。
成本	在 AWS 上執行工作負載的成本	• 將 CDH 工作負載遷移到 AWS 的成本較高層次會假設您將在 AWS 上建立新環境。它包括人員的時間和精

		力會計，以及為新環境佈建計算資源和授權軟件。 • Cloudera 雲端消費型定價模式可讓您靈活運用爆量和自動擴充功能。如需詳細資訊，請參閱 Cloudera 網站上的 CDP 公有雲服務費率。 • Cloudera 企業資料中心以亞馬遜彈性運算雲端 (Amazon EC2) 為基礎，並密切模型化傳統叢集。您可以自訂資料中樞，但這會影響成本。 • CDP 公共雲數據倉庫，Cloudera Machine Learning 和 Cloudera 數據工程（CDE）是基於容器的，可以配置為自動擴展。
基礎設施協議和框架	系統要求	請參閱先決條件一節。
	SLA	請參閱 CDP 公有雲服務級別協議。
	博士	請參閱 Cloudera 文件中的災難復原。
	授權和操作模式 (適用於目標 AWS 帳戶)	使用您自己的授權 (BYOL) 模式
合規	安全要求	請參閱 Cloudera 文檔中的 Cloudera 安全概述。
	其他合規性認證	請參閱 Cloudera 網站上有關一般資料保護法規 (GDPR) 合規性和 CDP 信任中心的相關資訊。

先決條件和限制

先決條件

- AWS 帳戶要求，包括帳戶、資源、服務和許可，例如 AWS Identity and Access Management (IAM) 角色和政策設定
- 從 Clou@@ dera 網站部署 CDP 的先決條件

移轉需要下列角色和專業知識：

Role	技能和責任
遷移, 領導	確保執行支持，團隊協作，規劃，實施和評估
雲德中小企業	CDH，HDP 和 CDP 管理，系統管理和架構方面的專業技能
AWS 架構師	AWS 服務、聯網、安全和架構方面的技能

架構

建置適當的架構是確保移轉與效能符合您期望的關鍵步驟。為了符合此教戰手冊假設的遷移工作，AWS 雲端中的目標資料環境 (無論是在虛擬私有雲端 (VPC) 託管的執行個體或 CDP 上，在作業系統和軟體版本以及主要機器規格方面，都必須與來源環境相同。

下圖 (在 Cloudera 共用資料體驗資料表的權限下複製) 顯示 CDP 環境的基礎架構元件，以及層級或基礎架構元件的互動方式。

該架構包括以下 CDP 組件：

- 資料中樞是一項服務，用於啟動和管理由 Cloudera 執行階段支援的工作負載叢集。您可以使用 Data Hub 中的叢集定義佈建和存取自訂使用案例的工作負載叢集，以及定義自訂叢集配置。如需詳細資訊，請參閱 C loudera 網站。
- 資料流和串流可解決企業在動態資料方面面臨的關鍵挑戰。它管理以下內容：
  - 處理大量和大規模的即時資料串流
  - 追蹤串流資料的資料來源和歷程

- 管理和監控邊緣應用程式和串流來源

有關更多信息，請參閱 Cloudera 網站上的 Cloudera DataFlow 和 CSP。

- 資料工程包括資料整合、資料品質和資料控管，可協助組織建置和維護資料管線和工作流程。如需詳細資訊，請參閱 C loudera 網站。了解對 Spot 執行個體的支援，以便在 AWS 上為 Cloudera 資料工程工作負載節省成本。
- 資料倉儲可讓您建立獨立的資料倉儲和資料集區，以自動調整以符合工作負載需求。此服務為每個資料倉儲和資料超市提供隔離的運算執行個體和自動化最佳化，並協助您在符合 SLA 的同時節省成本。如需詳細資訊，請參閱 C loudera 網站。了解如何在 AWS 上管理 Cloudera 資料倉儲的成本和 auto-scaling。
- CDP 中的操作數據庫為可擴展的高性能應用程序提供了可靠和靈活的基礎。它提供即時、隨時可用、可擴充的資料庫，在統一的作業和倉儲平台中，為傳統結構化資料與新的非結構化資料提供服務。如需詳細資訊，請參閱 C loudera 網站。
- Machine Learning 是一種雲端原生機器學習平台，可將自助式資料科學和資料工程功能合併為企業資料雲端中的單一可攜式服務。它可讓您在任何地方的資料上進行機器學習和人工智慧 (AI) 的可擴充部署。如需詳細資訊，請參閱 C loudera 網站。

AWS 上的 CDP

下圖 (經 Cloudera 網站的許可改編) 顯示 AWS 上 CDP 的高階架構。CDP 實現了自己的安全模型來管理帳戶和數據流。這些都是透過使用跨帳戶角色與 IAM 整合。

CDP 控制平面駐留在自己的 VPC 中的 Cloudera 主帳戶中。每個客戶帳戶都有自己的子帳戶和唯一的 VPC。跨帳戶 IAM 角色和 SSL 技術會將管理流量路由傳送至控制平面，往返於每個客戶 VPC 內的網際網路可路由公用子網路上的客戶服務。在客戶的 VPC 上，Cloudera 共用資料體驗 (SDX) 提供企業級安全性，並具有統一的治理和合規性，因此您可以更快地從資料中獲得見解。SDX 是一個融入所有 Cloudera 產品的設計理念。如需適用於 AWS 的 SDX 和 CDP 公有雲網路架構的詳細資訊，請參閱 Cloudera 文件。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

- Amazon Elastic Kubernetes Service (Amazon EKS) 可協助您在 AWS 上執行 Kubernetes，而無需安裝或維護自己的 Kubernetes 控制平面或節點。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

自動化和工具

- 對於其他工具，您可以使用 Cloudera Backup 資料復原 (BDR)、AWS Snowball 和 AWS Snowmobile，協助將資料從現場部署 CDH、HDP 和 CDP 遷移到 AWS 託管的 CDP。
- 對於新部署，建議您使用適用於 CDP 的 AWS 合作夥伴解決方案。

史诗

準備移轉

任務	描述	所需技能
參與克洛德拉團隊。	Cloudera 與客戶一起追求標準化的參與模式，並可與您的系統整合商 (SI) 合作推廣相同的方法。請聯絡 Cloudera 客戶團隊，以便他們提供指導和必要的技術資源以開始專案。聯繫 Cloudera 團隊可確保所有必要的團隊都可以在日期臨近時為遷移做好準備。 您可以聯絡 Cloudera 專業服務部門，以更低的成本和最佳效能，快速將 Cloudera 部署從試驗轉移到生產環境。有關產品	遷移, 領導

任務	描述	所需技能
	的完整列表，請參閱 C loudera 網站。	
在 AWS 上為您的 VPC 擬私人雲端建立 CDP 公有雲環境。	與 Cloudera 專業服務或您的 SI 合作，在 AWS 上規劃 CDP 公有雲端並將其部署到 VPC 中。	雲端架構師, 中小企業
排定優先順序並評估要移轉的工作	評估所有內部部署工作負載，以確定最容易移轉的工作負載。不是關鍵任務的應用程式最好先移動，因為這些應用程式對客戶的影響最小。在成功移轉其他工作負載後，最後儲存關鍵任務工作負載。 備註：暫時性 (CDP 資料工程) 工作負載比持續性 (CDP 資料倉儲) 工作負載更容易移轉。移轉時，請務必考慮資料量和位置。挑戰包括將資料從內部部署環境持續複寫到雲端，以及變更資料擷取管道以將資料直接匯入雲端。	遷移, 領導
討論 CDH、HDP、CDP 和舊版應用程式遷移活動。	考慮並開始使用 Cloudera 工作負載管理員規劃下列活動： • 要複製到 AWS 環境的資料和工作負載 • 雲端就緒資料 • 嘈雜的鄰居，這消耗了資源，並為其他租戶創造問題 • 彈性工作量 • 具有高營運開銷的小型叢集	遷移, 領導

任務	描述	所需技能
完成 Cloudera 複製管理員的需求和建議。	與 Cloudera 專業服務和您的 SI 合作，準備將工作負載遷移到 AWS 上的 CDP 公有雲環境。 瞭解下列需求和建議可協助您避免在安裝複寫管理員服務期間和之後的常見問題。 • 檢閱複製管理員支援文件，以確認您符合環境和系統需求。如需詳細資訊，請參 Support Cloudera 網站上 CDP 公有雲複寫管理員的支援對照表。 • 您不需要對將安裝複寫管理員應用程式和資料生命週期管理員 (DLM) 引擎的節點具有 root 存取權。 • 在複寫管理員初始安裝期間安裝 Apache Hive，除非您確定 future 不會使用 Hive 複寫。如果您決定在複製管理器中創建 HDFS 複製策略後安裝蜂巢, 你必須刪除，然後重新創建所有 HDFS 複製策略添加蜂巢後. • 複寫管理員中使用的叢集必須具有對稱的組態。複寫關係中的每個叢集必須設定完全相同的安全性 (Kerberos)、使用者管理 (LDAP/AD) 和 Knox 代理伺服器。集群服務，如 Hadoop 的分佈式文件系統（HDFS），阿帕奇蜂巢，阿帕奇諾克斯，阿	遷移, 領導

任務	描述	所需技能
	帕奇遊俠和阿帕奇阿特拉斯可以有高可用性（HA）不同的配置。例如，來源和目標叢集可能有個別的 HA 和非 HA 組態。	

將 CDP 遷移到 AWS

任務	描述	所需技能
使用 Cloudera 工作負載管理員移轉開發/測試環境的第一個工作負載。	SI 可協助您將第一個工作負載遷移到 AWS 雲端。這應該是一個不面向客戶或關鍵任務的應用程序。開發/測試移轉的理想選擇是具有雲端可輕鬆擷取資料的應用程式，例如 CDP 資料工程工作負載。與可能有許多需要不間斷存取的使用者 (例如 CDP Data Warehouse 工作負載) 相比，這是暫時性工作負載，存取該工作負載的使用者通常較少。數據工程工作負載不是持久性的，如果出現問題，可最大限度地減少業務影響。但是，這些工作對於生產報告來說可能很重要，因此請先排定低影響力的資料工程工作負載。	遷移, 領導
視需要重複移轉步驟。	Cloudera 工作負載管理員可協助識別最適合雲端的工作負載。它提供了諸如雲端效能分級、目標環境的大小/容量計劃，以及複寫計劃等指標。移轉的最佳候選項包括季節性工	雲德中小企業

任務	描述	所需技能
	作負載、臨機操作報告，以及不會耗用大量資源的間歇性工作。 Cloudera 複寫管理員會將資料從內部部署移至雲端，並從雲端移至內部部署。 使用工作負載管理員，主動最佳化資料倉儲、資料工程和機器學習的工作負載、應用程式、效能和基礎架構容量。如需有關如何現代化資料倉儲的完整指南，請參閱 Cloudera 網站。	

相關資源

克魯德拉文檔：

- 使用 CDP、Cloudera 管理員和複寫管理員註冊傳統叢集：
  - 管理主控台
  - 複寫管理員蜂巢複寫
- 哨兵複製
- 哨兵權限
- 資料中樞叢集規劃清單
- 工作負載管理器
- 複寫管理員需求
- 雲端數據平台可觀測性
- AWS 要求

AWS 文件：

- 雲端資料遷移

重新啟動 RHEL 來源伺服器後，自動重新啟動 AWS 複寫代理程式而不停用 SELinux

由阿尼爾·庫納帕雷迪（AWS），香穆甘·香克（AWS）和文卡特拉瑪納奇塔（AWS）創建

環境：生產	技術：移轉；作業系統	工作負載：開源
AWS 服務：AWS 應用程式遷移服務		

Summary

AWS 應用程式遷移服務有助於簡化、加速和自動化您的 RHEL (RHEL) 工作負載遷移到 Amazon Web Services (AWS) 雲端。若要將來源伺服器新增至應用程式遷移服務，請在伺服器上安裝 AWS 複寫代理程式。

應用程式移轉服務提供即時、非同步的區塊層級複製。這表示您可以在整個複寫程序期間繼續正常的 IT 作業。 這些 IT 作業可能需要您在移轉期間重新啟動或重新啟動 RHEL 來源伺服器。如果發生這種情況，AWS 複寫代理程式將不會自動重新啟動，而且您的資料複寫也會停止。一般而言，您可以將安全性增強型 Linux (SELinux) 設定為停用或寬鬆模式，以自動重新啟動 AWS 複寫代理程式。但是，您組織的安全性原則可能會禁止停用 SELinux，而且您可能還必須重新標記檔案。

此模式描述了如何在 RHEL 來源伺服器在遷移期間重新啟動或重新啟動時自動重新啟動 AWS 複寫代理程式，而不關閉 SELinux。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 您想要遷移到 AWS 雲端的現場部署 RHEL 工作負載。
- 從應用程式移轉服務主控台初始化應用程式移轉服務 僅在您第一次使用此服務時才需要初始化。如需指示，請參閱應用程式移轉服務說明文件。
- 適用於應用程式遷移服務的現有 AWS Identity and Access Management (IAM) 政策。如需詳細資訊，請參閱應用程式移轉服務說明文件。

版本

- RHEL 版本 7 或更新版本

工具

AWS 服務

- AWS 應用程式遷移服務是一種高度自動化 lift-and-shift (重新託管) 解決方案，可簡化、加速並降低將應用程式遷移到 AWS 的成本。

命令

下表提供您將在 RHEL 來源伺服器上執行的 Linux 指令清單。這些也在這種模式的史詩和故事中進行了描述。

命令	Description
`#systemctl –version`	識別系統版本。
`#systemctl list-units --type=service`	列出 RHEL 伺服器上所有可用的作用中服務。
`#systemctl list-units --type=service | grep running`	列出 RHEL 伺服器上目前執行的所有服務。
`#systemctl list-units --type=service | grep failed`	列出 RHEL 伺服器重新啟動或重新啟動後無法載入的所有服務。
`restorecon -Rv /etc/rc.d/init.d/aws-replication-service`	將前後關聯變更為`aws-replication-service` 。
`yum install policycoreutils*`	安裝 SELinux 系統作業所需的原則核心公用程式。
`ausearch -c "insmod" --raw | audit2allow -M my-modprobe`	搜尋稽核記錄檔並建立策略的模組。
`semodule -i my-modprobe.pp`	啟動策略。

`cat my-modprobe.te`	顯示my-modprobe.te 檔案的內容。
`semodule -l | grep my-modprobe`	檢查政策是否已經載入 SELinux 模組。

史诗

安裝 AWS 複寫代理程式並重新啟動 RHEL 來源伺服器

任務	描述	所需技能
使用存取金鑰和秘密存取金鑰建立「應用程式移轉服務」使用者。	若要安裝 AWS 複寫代理程式，您必須使用必要的 AWS 登入資料建立應用程式遷移服務使用者。如需指示，請參閱應用程式移轉服務說明文件。	移民工程師
安裝 AWS 複寫代理程式。	1. 登入 AWS 管理主控台，然後開啟 AWS Migration Service 主控台，網址為 https://console.aws.amazon.com/mgn/home。 2. 依照應用程式移轉服務說明文件中的指示設定複製設定。 3. 按照應用程式遷移服務文件中的指示安裝 AWS 複寫代理程式。 4. 在 [來源伺服器] 頁面上，選擇 RHEL 來源伺服器，然後選擇 [複製] 以啟動初始複製。如需詳細資訊，請參閱應用程式移轉服務說明文件。	移民工程師

任務	描述	所需技能
重新啟動或重新啟動 RHEL 來源伺服器。	當 RHEL 來源伺服器的「資料複製」狀態在「移轉」儀表板上顯示為「正常」時，請重新啟動	移民工程師
檢查資料複製狀態。	等待一小時，然後在 [遷移] 儀表板上再次檢查資料複寫狀態。它應該處於停滯狀態。	移民工程師

檢查 RHEL 來源伺服器上的 AWS 複寫代理程式狀態

任務	描述	所需技能
識別系統版本。	開啟 RHEL 來源伺服器的命令列介面，然後執行下列指令來識別系統版本： `#systemctl –version`	移民工程師
列出所有作用中的服務。	若要列出 RHEL 伺服器上所有可用的作用中服務，請執行以下命令： `#systemctl list-units --type=service`	移民工程師
列出所有執行中的服務。	若要列出 RHEL 伺服器上目前執行的所有服務，請使用以下指令： `#systemctl list-units --type=service | grep running`	移民工程師

任務	描述	所需技能
列出所有載入失敗的服務。	若要列出 RHEL 伺服器重新啟動或重新啟動後無法載入的所有服務，請執行以下命令： `#systemctl list-units --type=service | grep failed`	移民工程師

建立並執行 SELinux 模組

任務	描述	所需技能
變更安全性內容。	在 RHEL 來源伺服器的命令列介面中，執行下列命令，將安全內容變更為 AWS 複寫服務： `restorecon -Rv /etc/rc.d/init.d/aws-replication-service`	移民工程師
安裝核心公用程式。	要安裝 SELinux 系統及其策略操作所需的核心實用程序，請運行以下命令： `yum install policycoreutils*`	移民工程師
搜尋稽核記錄檔並建立策略的模組。	執行 命令： `ausearch -c "insmod" --raw | audit2allow -M my-modprobe`	移民工程師
顯示 my-modprobe-te 檔案的內容。	`my-modprobe.te` 檔案由稽核 2 允許指令產生。它包括	移民工程師

任務	描述	所需技能
	SELinux 網域、原則來源目錄和子目錄，並指定與網域相關聯的存取向量規則和轉換。若要顯示檔案的內容，請執行以下指令： `cat my modprobe.te`	
啟動策略。	若要插入模組並使原則套件處於作用中狀態，請執行下列命令： `semodule -i my-modprobe.pp`	移民工程師
檢查模塊是否已加載。	執行 命令： `semodule -l | grep my-modprobe` SELinux 模組載入之後，您將不再需要在移轉期間將 SELinux 設定為停用或寬容模式。	移民工程師
重新啟動或重新啟動 RHEL 來源伺服器，並確認資料複製狀態。	開啟 AWS Migration Service 主控台，瀏覽至資料複寫進度，然後重新啟動或重新啟動 RHEL 來源伺服器。資料複製現在應該會在 RHEL 來源伺服器重新啟動後自動繼續。	移民工程師

相關資源

- 應用程式遷移服務說
- 技術培訓教材

- 解決 AWS 複寫代理程式問題
- 應用程式移轉服務原

重新建築師

主題

- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 在與 PostgreSQL 相容的 Aurora 中建立應用程式使用者和角色
- 使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR
- 使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS
- 在 Aurora 相容中使用檔案編碼將 BLOB 檔案載入至文字
- 使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS SCT 和 AWS DMS 將適用於甲骨文的亞馬遜 RDS 遷移到適用於 PostgreSQL 的 CLI 馬遜 RDS CloudFormation
- 將甲骨文序列遷移 _ 可重複使用的編譯包到 PostgreSQL
- 將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容
- 將基於函數的索引從甲骨文遷移到 PostgreSQL
- 使用擴充功能將甲骨文原生函數遷移至 PostgreSQL
- 使用 AWS DMS 將 Db2 資料庫從 Amazon EC2 遷移到與 MySQL 相容的 Aurora
- 通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB
- 將現場部署 ThoughtSpot 獵鷹資料庫遷移到 Amazon Redshift
- 使用 AWS DMS 將甲骨文資料庫遷移到 Amazon DynamoDB 資料庫
- 使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL
- 從 Amazon RDS for Oracle 遷移到 Amazon RDS for MySQL
- 使用 AWS DMS 和 AWS SCT，從 Amazon EC2 上的 IBM Db2 遷移到 Aurora 與 PostgreSQL 相容
- 使用和 AWS DMS 從甲骨文 8i 或 9i 遷移到亞馬遜 RDS SharePlex
- 使用具體化視圖和 AWS DMS，從甲骨文 8i 或 9i 遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 從亞馬遜上的甲骨文遷移到 Amazon RDS for MySQL
- 使用 AWS DMS 從甲骨文遷移到 Amazon DocumentDB
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將現場部署甲骨文資料庫遷移到適用於 MySQL 的 Amazon RDS for MySQL

- 使用甲骨文旁觀者和 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS
- 使用甲骨文從甲骨文數據庫遷移到亞馬遜 RDS GoldenGate
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Aurora
- 將資料從內部部署 Oracle 資料庫遷 PostgreSQL 至 Aurora
- 使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器
- 使用 AWS DMS 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將現場部署 Microsoft SQL 伺服器資料庫遷移至 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將現場部署 Vertica 資料庫遷移到 Amazon Redshift 移
- 將舊有應用程式從 Oracle Pro*C 移轉至 ECPG
- 將虛擬生成的列從甲骨文遷移到 PostgreSQL
- 在 Aurora 相容上設定甲骨文 UTL_FILE 功能
- 從甲骨文遷移到 Amazon Aurora PostgreSQL 後驗證數據庫對象

將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型

由納雷什·達默拉（AWS）創建

環境：PoC 或試點	來源：甲骨文	目標：Amazon Aurora
R 型：重新建築	工作量：甲骨文	技術：移轉、軟體開發與測試、儲存與備份、資料庫
AWS 服務：Amazon Aurora; AWS DMS; Amazon RDS; AWS SCT		

Summary

從適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 遷移到 Amazon Aurora PostgreSQL 相容版本期間，在亞馬遜網路服務 (AWS) Database Migration Service (AWS DMS) 中驗證遷移時，可能會遇到資料不匹配。為了防止這種不匹配，您可以將 VARCHAR2（1）數據類型轉換為布爾數據類型。

VARCHAR2 資料類型會儲存可變長度的文字字串，而 VARCHAR2 (1) 則表示字串長度為 1 個字元或 1 個位元組。如需 VARCHAR2 的詳細資訊，請參閱 Oracle 內建資料類型 (Oracle 說明文件)。

在此模式中，在範例來源資料表資料行中，VARCHAR2 (1) 資料可以是 Y，代表「是」或「N」，表示「否」。 此模式包括使用 AWS DMS 和 AWS Schema Conversion Tool (AWS SCT) 將此資料類型從 VARCHAR2 (1) 中的 Y 和 N 值轉換為布林值的真值或假值的說明。

目標受眾

對於那些有使用 AWS DMS 將甲骨文資料庫遷移到與 Aurora PostgreSQL 相容的經驗的人士，建議使用此模式。當您完成移轉時，請遵循將甲骨文轉換為 Amazon RDS for PostgreSQL 或 Amazon Aurora PostgreSQL (AWS SCT 文件) 中的建議。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 確認您的環境已為 Aurora 做好準備，包括設定認證、權限和安全群組。如需詳細資訊，請參閱為 Amazon Aurora 設定環境 (Aurora 文件)。
- 適用於甲骨文資料庫的來源亞馬遜 RDS，其中包含一個包含 VARCHAR2 (1) 資料的表格欄。
- 目標 Amazon Aurora 與 PostgreSQL 相容的資料庫執行個體。如需詳細資訊，請參閱建立資料庫叢集和連線至 Aurora PostgreSQL 資料庫叢集上的資料庫 (Aurora 說明文件)。

產品版本

- Amazon RDS for Oracle 文 12.1.0.2 或更高版本。
- AWS DMS 版本 3.1.4 或更新版本。如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源和使用 PostgreSQL 資料庫做為 AWS DMS 的目標 (AWS DMS 文件)。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。
- AWS Schema Conversion Tool (AWS SCT) 1.0.632 版或更新版本。我們建議您使用最新版本的 AWS SCT 以獲得最全面的版本和功能支援。
- Aurora 支援適用於 Aurora 與 PostgreSQL 相容的資料庫引擎版本 (Aurora 文件) 中列出的 PostgreSQL 版本。

架構

源, 技術, 堆棧

Amazon RDS for Oracle 數據庫實例

目標技術堆疊

Amazon Aurora 兼容資料庫執行個體

來源與目標架構

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

其他服務

- Oracle SQL 開發人員是一個整合式開發環境，可簡化傳統和雲端部署中 Oracle 資料庫的開發與管理。在此模式中，您可以使用此工具連線到適用於 Oracle 資料庫執行個體的 Amazon RDS，並查詢資料。
- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。在此模式中，您可以使用此工具連接至 Aurora 資料庫執行個體並查詢資料。

史诗

準備移轉

任務	描述	所需技能
建立資料庫移轉報告。	1. 在 AWS SCT 中，建立資料庫遷移評估報告。如需詳細資訊，請參閱建立移轉評估報告。 2. 檢閱並執行移轉評估報告中的行動項目。如需詳細資訊，請參閱評估報告行動項目。	DBA, 開發人員
停用目標資料庫上的外來索引鍵條件約束。	在 PostgreSQL 中，外鍵是通過使用觸發器來實現的。在滿載階段，AWS DMS 每次載入一個表格。強烈建議您使用下列其中一種方法，在完全載	DBA, 開發人員

任務	描述	所需技能
	入期間停用外來索引鍵條件約束： • 暫時停用執行個體的所有觸發，並完成完全載入。 • 在 PostgreSQL 中使用 `session_replication_role` 參數。 如果停用外部索引鍵條件約束是不可行的，請為父資料表和子表格特定的主要資料建立 AWS DMS 遷移任務。	
停用目標資料庫上的主索引鍵和唯一索引鍵。	使用下列命令，停用目標資料庫上的主索引鍵和條件約束。這有助於改善初始載入工作的效能。 `ALTER TABLE <table> DISABLE PRIMARY KEY;` `ALTER TABLE <table> DISABLE CONSTRAINT <constraint_name>;`	DBA, 開發人員
建立初始載入工作。	在 AWS DMS 中，為初始負載建立遷移任務。如需指示，請參閱建立工作。對於移轉方法，請選擇「移轉現有資料」。此遷移方法在 API Full Load 中被調用。不要開始這項工作尚未。	DBA, 開發人員

任務	描述	所需技能
編輯初始載入工作的工作設定。	編輯工作設定以新增資料驗證。這些驗證設定必須在 JSON 檔案中建立。如需指示和範例，請參閱指定工作設定。新增下列驗證： • 若要驗證 VARCHAR2 (1) 資料是否正確地轉換為目標資料庫中的布林值，請在此模式的「其他資訊」區段的「資料驗證指令碼」中新增程式碼。驗證指令碼會將目標資料表中的布林值 1 轉換為 Y，0 轉換成 N，然後將目標資料表中的值與來源資料表進行比較。 若要驗證其餘的資料移轉，請在工作中啟用資料驗證。如需詳細資訊，請參閱資料驗證工作設定。	AWS 管理員，DBA
建立進行中的複寫工作。	在 AWS DMS 中，建立使目標資料庫與來源資料庫保持同步的遷移任務。如需指示，請參閱建立工作。對於移轉方法，請選擇 [僅複製資料變更]。不要開始這項工作尚未。	DBA

測試移轉工作

任務	描述	所需技能
建立用於測試的範例資料。	在來源資料庫中，建立含有資料的範例資料表以供測試之用。	開發人員
確認沒有衝突的活動。	使用檢`pg_stat_activity` 查伺服器上是否有任何可能影響移轉的活動。如需詳細資訊，請參閱統計資料收集器 (PostgreSQL 文件集)。	AWS 管理員
啟動 AWS DMS 遷移任務。	在 AWS DMS 主控台的 [儀表板] 頁面上，啟動您在先前史詩中建立的初始載入和進行中的複寫任務。	AWS 管理員
監視工作和表格載入狀態。	在移轉期間，請監視工作狀態和表格狀態。當初始載入工作完成時，在 [表格統計資料] 索引標籤上： • 「載入」狀態應為「表格已完成」。 • 驗證狀態應該被驗證。	AWS 管理員
驗證移轉結果。	使用 pgAdmin，查詢目標資料庫上的資料表。 成功的查詢表示資料已成功移轉。	開發人員
在目標數據庫上添加主鍵和外鍵。	在目標數據庫上創建主鍵和外鍵。如需詳細資訊，請參閱改變資料表 (PostgreSQL 網站)。	DBA
清理測試數據。	在來源和目標資料庫上，清理為單元測試所建立的資料。	開發人員

切過

任務	描述	所需技能
完成移轉。	重複以前的史詩，測試遷移任務，使用真實的源數據。 這會將資料從來源移轉至目標資料庫。	開發人員
驗證來源和目標資料庫是否同步。	驗證來源和目標資料庫是否同步。如需詳細資訊和指示，請參閱 AWS DMS 資料驗證。	開發人員
停止來源資料庫。	停止 Amazon RDS for Oracle 數據庫。如需指示，請參閱暫時停止 Amazon RDS 資料庫執行個體。當您停止來源資料庫時，AWS DMS 中的初始負載和進行中的複寫任務會自動停止。停止這些工作不需要其他動作。	開發人員

相關資源

AWS 參考資料

- 使用 AWS DMS 和 AWS SCT (AWS Prescriptive Guidance) 將甲骨文資料庫遷移至 Aurora
- 將甲骨文轉換 Amazon RDS for PostgreSQL PostgreSQL 或 Amazon Aurora (AWS SCT 文件)
- AWS DMS 的運作方式 (AWS DMS 文件)

其他參考

- 布林型資料類 PostgreSQL 件集)
- Oracle 內建資料類型 (Oracle 文件集)
- pgAdmin (pgAdmin 網站)
- SQL 開發人員 (甲骨文網站)

教程和視頻

- 開始使用 AWS DMS
- 開始使用 Amazon RDS
- AWS DMS 簡介 (影片)
- 了解 Amazon RDS（視頻）

其他資訊

數據驗證腳本

以下數據驗證腳本將 1 轉換為 Y，將 0 轉換為 N。 這有助於成功完成 AWS DMS 任務並通過表格驗證。

```
{
"rule-type": "validation",
"rule-id": "5",
"rule-name": "5",
"rule-target": "column",
"object-locator": {
"schema-name": "ADMIN",
"table-name": "TEMP_CHRA_BOOL",
"column-name": "GRADE"
},
"rule-action": "override-validation-function",
"target-function": "case grade when '1' then 'Y' else 'N' end"
        }
```

指令碼中的case陳述式會執行驗證。如果驗證失敗，AWS DMS 會在目標資料庫執行個體的公開 .awsdms_validation_ v1 資料表中插入一筆記錄。此記錄包括資料表名稱、錯誤時間，以及來源和目標資料表中不相符值的詳細資訊。

如果您未將此資料驗證指令碼新增至 AWS DMS 任務，且資料已插入目標資料表中，則 AWS DMS 任務會將驗證狀態顯示為不匹配記錄。

在 AWS SCT 轉換期間，AWS DMS 遷移任務會將 VARCHAR2 (1) 資料類型的資料類型變更為布林值，並在資料行上新增主索引鍵限制。"N0"

在與 PostgreSQL 相容的 Aurora 中建立應用程式使用者和角色

創建者：阿布舍克維爾瑪（AWS）

環境：PoC 或試點	來源：任何資料庫	目標 PostgreSQL 庫
R 型：重新建築	工作負載：開源	技術：移轉；資料庫
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

當您移轉至 Amazon Aurora PostgreSQL 相容版本時，必須在與 Aurora PostgreSQL 相容的資料庫中建立存在於來源資料庫上的資料庫使用者和角色。您可以使用兩種不同的方法，在相容於 Aurora PostgreSQL 中建立使用者和角色：

- 在目標中使用與來源資料庫中類似的使用者和角色。在這種方法中，會從來源資料庫擷取使用者和角色的資料定義語言 (DDL)。然後，它們被轉換並應用到目標 Aurora PostgreSQL 兼容的數據庫。例如，使用 SQL 將使用者、角色和授權從 Oracle 對應到 PostgreSQL 的部落格文章涵蓋了使用從 Oracle 來源資料庫引擎擷取的方式。
- 使用在開發、管理期間常用的標準化使用者和角色，以及在資料庫中執行其他相關作業。這包括唯讀、讀取/寫入、開發、管理和由個別使用者執行的部署作業。

此模式包含在與 Aurora PostgreSQL 相容的使用者和角色建立所需的授權，以符合標準化使用者和角色方法的需求。使用者和角色建立步驟與授與資料庫使用者最少權限的安全性原則一致。下表列出使用者、他們對應的角色，以及他們在資料庫上的詳細資訊。

使用者	Roles (角色)	用途
`APP_read`	`APP_RO`	用於架構的唯讀存取 APP
`APP_WRITE`	`APP_RW`	用於架構上的寫入和讀取操作 APP

APP_dev_user	APP_DEV	用於模式上的開發目的APP_DEV，在模式上具有只讀訪問權限 APP
Admin_User	rds_superuser	用於對數據庫執行管理員操作
APP	APP_DEP	用於在模式下創建對象，並在APP模式中的對象的部署 APP

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶
- 適用於 PostgreSQL 資料庫、Amazon Aurora 與 PostgreSQL 資料庫相容的版本資料庫或 Amazon Relational Database Service 服務 (Amazon RDS)

產品版本

- 所有 PostgreSQL

架構

源, 技術, 堆棧

- 任何資料庫

目標技術堆疊

- Amazon Aurora 郵政兼容

目標架構

下圖顯示了與 Aurora PostgreSQL 相容的資料庫中的使用者角色和結構描述架構。

自動化和規模

此模式包含使用者、角色和結構描述建立命令檔，您可以多次執行這些命令檔，而不會對來源或目標資料庫的現有使用者造成任何影響。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。

其他服務

- psql 是一個以終端為基礎的前端工具，會隨每個 PostgreSQL 資料庫安裝一起安裝。它具有用於運行 SQL，PL-PGSQL 和操作系統命令的命令行界面。
- pgAdmin 是一個開放源代碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

建立使用者和角色

任務	描述	所需技能
建立部署使用者。	部署使用者APP將用來在部署期間建立和修改資料庫物件。使用下列指令碼在結構描述APP_DEP中建立部署使用者角色APP。驗證存取權限，以確保此使用者只有在所需綱要中建立物件的權限APP。 1. Connect 至管理員使用者，並建立結構描述。	DBA

任務	描述	所需技能
	`` CREATE SCHEMA APP; ``` 2. 建立使用者。 ``` CREATE USER APP WITH PASSWORD <password > ; ``` 3. 建立角色。 ``` CREATE ROLE APP_DEP ; GRANT all on schema APP to APP_DEP ; GRANT USAGE ON SCHEMA APP to APP_DEP ; GRANT connect on database <db_name> to APP_DEP ; GRANT APP_DEP to APP; ``` 4. 若要測試權限，請連線至表格APP並建立表格。 ``` set search_path to APP; SET CREATE TABLE test(id integer) ; CREATE TABLE ``` 5. 檢查權限。 ``` select schemaname , tablename , tableowne r from pg_tables where tablename like 'test' ; schemaname 	tablename | tableowner ```	

任務	描述	所需技能
	APP | test | APP	

任務	描述	所需技能
建立唯讀使用者。	唯讀使用者APP_read將用於在結構描述中執行唯讀作業APP。使用下列指令碼建立唯讀使用者。驗證存取權限，以確定此使用者只有讀取綱要中物件的權限，以APP及自動授與在結構描述中建立之任何新物件的讀取存取權APP。 1. 建立使用者APP_read。 `create user APP_read ;` `alter user APP_read` `with password` `'your_password' ;` 2. 建立角色。 `CREATE ROLE APP_ro ;` `GRANT SELECT ON ALL` `TABLES IN SCHEMA APP` `TO APP_RO ;` `GRANT USAGE ON SCHEMA` `APP TO APP_RO` `GRANT CONNECT ON` `DATABASE testdb TO` `APP_RO ;` `GRANT APP_RO TO` `APP_read;` 3. 若要測試權限，請使用使用APP_read者登入。 `set search_path to` `APP ;` `create table test1( id` `integer) ;` `ERROR: permission` `denied for schema APP`	DBA

任務	描述	所需技能
	LINE 1: create table test1(id integer) ; insert into test values (34) ; ERROR: permission denied for table test SQL state: 42501 select from test no rows selected	

任務	描述	所需技能
建立讀取/寫入使用者。	讀取/寫入使用者APP_WRITE 將用來對結構描述執行讀取和寫入作業APP。使用下列指令碼來建立讀取/寫入使用者，並將APP_RW角色授與該使用者。驗證存取權限，以確定此使用者只對結構描述中的物件具有讀取和寫入權限，以APP及自動授與在結構描述中建立之任何新物件的讀取和寫入權限APP。 1. 建立使用者。 `CREATE USER APP_WRITE ; alter user APP_WRITE with password 'your_password' ;` 2. 建立角色。 `CREATE ROLE APP_RW; GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA APP TO APP_RW ; GRANT CONNECT ON DATABASE postgres to APP_RW ; GRANT USAGE ON SCHEMA APP to APP_RW ; ALTER DEFAULT PRIVILEGES IN SCHEMA APP GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO APP_RW ;`	

任務	描述	所需技能
	GRANT APP_RW to APP_WRITE 3. 若要測試權限，請使用使用APP_WRITE 者登入。 SET SEARCH_PATH to APP; CREATE TABLE test1(id integer) ; ERROR: permission denied for schema APP LINE 1: create table test1(id integer) ; SELECT * FROM test ; id ---- 12 INSERT INTO test values (31) ; INSERT 0 1	

任務	描述	所需技能
建立管理員使用者。	admin 使用者Admin_User 將用來在資料庫上執行管理員作業。這些作業的範例為CREATE ROLE和CREATE DATABASE。Admin_User 使用內建角色rds_superuser 在資料庫上執行管理作業。使用下列指令碼建立和測試資料庫中 admin 使用者Admin_User 的權限。 1. 建立使用者並授與角色。 `create user Admin_Use r WITH PASSWORD ‘Your password’ ALTER user Admin_user CREATEDB; ALTER user Admin_user CREATEROLE;` 2. 若要測試權限，請從 Admin_User 使用者登入。 `SELECT * FROM APP.test ; id ---- 31 CREATE ROLE TEST ; CREATE DATABASE test123 ;`	DBA

任務	描述	所需技能
建立開發使用者。	開發使用者APP_dev_user 將有權在其本機結構描述中建立物件，APP_DEV並在結構描述中建立讀取權限APP。使用下列指令碼來建立和測試資料庫APP_dev_user 中使用者的權限。 1. 建立使用者。 `CREATE USER APP1_dev_user with password ‘your password’;` 2. 建立的APP_DEV綱要App_dev_user 。 `CREATE SCHEMA APP1_DEV ;` 3. 建立 APP_DEV 角色。 `CREATE ROLE APP1_DEV ;` `GRANT APP1_RO to APP1_DEV ;` `GRANT SELECT ON ALL TABLES IN SCHEMA APP1_DEV to APP1_dev_user` `GRANT USAGE, CREATE ON SCHEMA APP1_DEV to APP1_DEV_USER` `GRANT APP1_DEV to APP1_DEV_USER ;` 4. 若要測試權限，請從中登入APP_dev_user 。	DBA

任務	描述	所需技能
	`CREATE TABLE APP1_dev.test1( id integer ) ;` `CREATE TABLE` `INSERT into APP1_dev.test1 ( select * from APP1.test );` `INSERT 0 1` `CREATE TABLE APP1.test4 ( id int) ;` `ERROR: permissio n denied for schema APP1` `LINE 1: create table APP1.test4 ( id int) ;`	

相關資源

PostgreSQL 件

- 建立角色
- 建立使用者
- 預定義角色

其他資訊

PostgreSQL 14 增強功能

PostgreSQL 14 提供了一組預先定義的角色，可以訪問某些常用的特權功能和信息。管理員（包括具有CREATE ROLE權限的角色）可以將這些角色或其他角色授與使用者，讓他們能夠存取指定的權能和資訊。

管理員可以使用GRANT命令將這些角色的存取權授與使用者。例如，若要將pg_signal_backend角色授與Admin_User，您可以執行下列命令。

```
GRANT pg_signal_backend TO Admin_User;
```

該pg_signal_backend角色旨在允許管理員啟用受信任的非超級用戶角色將信號發送到其他後端。如需詳細資訊，請參閱 PostgreSQL 14 增強功能。

微調存取

在某些情況下，可能需要為使用者提供更精細的存取權 (例如，以表格為基礎的存取或以資料行為基礎的存取)。在這種情況下，您可以建立其他角色，將這些權限授與使用者。如需詳細資訊，請參閱 PostgreSQL 撥款。

使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR

由 HariKrishna 博加達 (AWS) 創建

環境：PoC 或試點	來源：甲骨文	目標：Aurora
R 型：重新建築	工作量：甲骨文	技術：移民、現代化、資料庫
AWS 服務：Amazon Aurora		

Summary

企業災難復原 (DR) 的最佳做法基本上包括設計和實作容錯的硬體和軟體系統，以便在災難中存活 (業務持續性) 和恢復正常作業 (恢復業務)，只需最少的干預，理想情況下也不會遺失資料。建置容錯環境以滿足企業災難復原目標既昂貴又耗時，而且需要企業的堅定承諾。

與任何其他保護 Oracle 資料的方法相比，Oracle 資料庫提供了三種不同的 DR 方法，可提供最高層級的資料保護和可用性。

- Oracle 零資料遺失復原應用裝置
- Oracle Active Data Guard
- 甲骨文 GoldenGate

此模式提供了一種使用 Amazon Aurora 全球數據庫模擬甲骨 GoldenGate 文 DR 的方法。參考架構在三個 AWS 區域中使 GoldenGate 用 Oracle 進行災難復原。該模式會逐步將來源架構重新平台轉換為以 Amazon Aurora PostgreSQL 相容版本為基礎的雲端原生 Aurora 全球資料庫。

Aurora 全域資料庫專為具有全球覆蓋量的應用程式所設計 單一 Aurora 資料庫可跨越多個 AWS 區域，最多具有五個次要區域。Aurora 全域資料庫提供下列功能：

- 實體儲存層級複寫
- 低延遲全域讀取
- 從全區域停機時快速進行災難復原
- 快速跨區域遷移
- 跨區域的低複寫延遲
- L ittle-to-no 效能對資料庫的影響

如需 Aurora 全球資料庫功能和優點的詳細資訊，請參閱使用 Amazon Aurora 全域資料庫。如需意外和受管容錯移轉的詳細資訊，請參閱在 Amazon Aurora 全球資料庫中使用容錯移轉。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 適用於應用程式連線的 Java 資料庫連線能力 (JDBC) PostgreSQL 驅動程式
- 以亞馬遜 Aurora PostgreSQL 相容版本為基礎的極光全球資料庫
- 甲骨文實際應用程式叢集 (RAC) 資料庫移轉至 Aurora 全球資料庫，以相容 Aurora PostgreSQL 為基礎

Aurora 全球資料庫的局限性

- 並非所有 AWS 區域都提供 Aurora 全球資料庫。如需支援的區域清單，請參閱使用 Aurora PostgreSQL 的 Aurora 全域資料庫。
- 如需 Aurora 全域資料庫不受支援的功能以及其他限制的相關資訊，請參閱 Amazon Aurora 全域資料庫的限制。

產品版本

- Amazon Aurora 版 — 兼容版本 10.14 或更高版本

架構

源, 技術, 堆棧

- 四節點資料庫
- 甲骨文 GoldenGate

來源架構

下圖顯示三個叢集，其中包含四個節點 Oracle RAC，位於使用 Oracle 複寫的不同 AWS 區域。GoldenGate

目標技術堆疊

- 以 Aurora PostgreSQL 為基礎的三個叢集 Amazon Aurora 全球資料庫，主要區域中有一個叢集，在不同的次要區域有兩個叢集

目標架構

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- Amazon Aurora 全球資料庫橫跨多個 AWS 區域，提供低延遲的全球讀取，並從可能影響整個 AWS 區域的罕見中斷中斷提供快速復原。

史诗

新增含讀取器資料庫執行個體

任務	描述	所需技能
連接一或多個次要 Aurora 叢集。	在 AWS 管理主控台上，選擇 Amazon Aurora。選取主要叢集，選擇「動作」，然後從下拉式清單中選擇「新增區域」。	DBA
選取執行個體類別。	您可以變更次要叢集的執行個體類別。不過，我們建議保持它與主要叢集執行個體類別相同。	DBA
新增第三個區域。	重複此史詩中的步驟，在第三個區域中新增叢集。	DBA

容錯移轉 Aurora 全域資料庫

任務	描述	所需技能
從 Aurora 全域資料庫移除主要叢集。	1. 在「資料庫」頁面上，選擇主要叢集。 2. 選擇「從全域移除」以容錯移轉至次要叢集。	DBA
重新設定您的應用程式，將寫入流量轉向新提升的叢集。	使用新升級叢集的端點修改應用程式中的端點。	DBA
停止向不可用的叢集發出任何寫入作業。	停止對您移除之叢集的應用程式和任何資料處理語言 (DML) 活動。	DBA
建立新的 Aurora 全域資料庫。	現在，您可以使用新升級的叢集作為主要叢集來建立 Aurora 全域資料庫。	DBA

啟動主要叢集

任務	描述	所需技能
從全域資料庫選取要啟動的主要叢集。	在 Amazon Aurora 主控台的全域資料庫設定中，選擇主要叢集。	DBA
啟動叢集。	在 [動作] 下拉式清單中，選擇 [開始]。此過程可能需要一些時間。重新整理畫面以查看狀態，或在作業完成後檢查叢集目前的狀態資料欄。	DBA

清理資源

任務	描述	所需技能
刪除剩餘的次要叢集。	完成容錯移轉試驗之後，請從全域資料庫移除次要叢集。	DBA
刪除主要叢集。	移除叢集。	DBA

相關資源

- 使用 Amazon Aurora 全球數據
- 使用亞馬遜 Aurora 全球資料庫的極光 PostgreSQL 災難復原解決方案 (部落格文章

使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS
R 型：重新建築	工作負載：甲骨文; 開源	技術：移民、資料庫、現代化
AWS 服務：Amazon EC2; Amazon RDS		

Summary

許多移轉策略和方法會分多個階段執行，這些階段可持續數週到數個月。在此期間，您可能會因為要移轉至 PostgreSQL 資料庫執行個體的來源 Oracle 資料庫執行個體進行修補或升級而發生延遲。若要避免這種情況，建議您將剩餘的 Oracle 資料庫程式碼以遞增方式移轉至 PostgreSQL 資料庫程式碼。

此模式為多 TB 的 Oracle 資料庫執行個體提供不停機的增量移轉策略，該執行個體在初始移轉後執行了大量交易，且必須移轉至 PostgreSQL 資料庫。您可以使用此模式的 step-by-step 方法，將適用於 Oracle 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS) 逐步遷移到 Amazon RDS for PostgreSQL 資料庫執行個體，而無需登入 Amazon Web Services (AWS) 管理主控台。

此模式使用「Oracle SQL 開發人員」來尋找來源 Oracle 資料庫中兩個結構描述之間的差異。然後，您可以使用 AWS 結構描述轉換工具 (AWS SCT)，將 Amazon RDS for Oracle 文資料庫結構描述物件轉換為亞馬遜 RDS 資料庫結構描述物件。然後，您可以在 Windows 命令提示字元中執行 Python 指令碼，為來源資料庫物件的累加式變更建立 AWS SCT 物件。

附註：在移轉生產工作負載之前，建議您在測試或非生產環境中針對此模式的方法執行概念驗證 (PoC)。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 適用於甲骨文資料庫執行個體的現有 Amazon RDS。

- 現有的亞馬遜 RDS 資料庫執行個體。
- 使用適用於甲骨文和 PostgreSQL 資料庫引擎的 JDBC 驅動程式安裝和設定 AWS SCT。如需這方面的詳細資訊，請參閱 AWS SCT 文件中的安裝 AWS SCT 和安裝所需的資料庫驅動程式。
- 甲骨文 SQL 開發人員，安裝和配置。如需這方面的詳細資訊，請參閱 Oracle SQL 開發人員說明文件。
- 該`incremental-migration-sct-sql.zip`文件（附件），下載到您的本地計算機。

限制

- 適用於 Oracle 資料庫的來源 Amazon RDS 執行個體的最低需求為：
  - 甲骨文 10.2 及更新版本（適用於 10.x 版）、11g（版本 11.2.0.3.v1 及更高版本）和最高 12.2 版本，以及 18c（適用於企業版、標準版、標準版和標準二版）
- 適用於 PostgreSQL 資料庫執行個體的目標 Amazon RDS 資料庫執行個體的最低需求為：
  - 版本 9.4 及更新版本 (適用於版本 9.x)、10.x 及 11.x 版
- 此模式使用 Oracle SQL 開發人員。如果您使用其他工具尋找和匯出結構描述差異，結果可能會有所不同。
- Oracle SQL 開發人員產生的 SQL 指令碼可能會引發轉換錯誤，這表示您必須執行手動移轉。

- 如果 AWS SCT 來源和目標測試連線失敗，請務必為虛擬私有雲端 (VPC) 安全群組設定 JDBC 驅動程式版本和輸入規則，以接受傳入流量。

產品版本

- 適用於甲骨文資料庫執行個體的亞馬遜 RDS 版本 12.1.0.2（版本 10.2 及更新版本）
- Amazon RDS for PostgreSQL 行個體 11.5 版 (9.4 版及更新版本)
- 甲骨文 SQL 開發人員 19.1 版及更新版本
- AWS SCT 版本 1.0.632 及更新版本

架構

源, 技術, 堆棧

- Amazon RDS for Oracle 數據庫

目標技術堆疊

- Amazon RDS for PostgreSQL 行個體

來源與目標架構

下圖顯示將 Amazon RDS for Oracle 文資料庫執行個體遷移到 Amazon RDS for PostgreSQL 的資料庫執行個體。

此圖表顯示下列移轉工作流程：

1. 開啟「Oracle SQL 開發人員」，並連線至來源和目標資料庫。
2. 產生差異報告，然後產生結構描述差異物件的 SQL 指令碼檔案。如需有關差異報告的詳細資訊，請參閱 Oracle 文件中的詳細差異報告。
3. 設定 AWS SCT 並執行 Python 程式碼。
4. SQL 腳本文件從甲骨文轉換為 PostgreSQL。
5. 在目標 PostgreSQL 資料庫執行個體上執行 SQL 指令碼檔案。

自動化和規模

您可以通過在單個程序中向 Python 腳本中添加多個功能的其他參數和安全相關更改來自動執行此遷移。

工具

- AWS SCT — AWS Schema Conversion Tool (AWS SCT) 可將現有的資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎。
- Oracle SQL 開發人員 — Oracle SQL 開發人員是一個整合式開發環境 (IDE)，可簡化傳統與雲端式部署中 Oracle 資料庫的開發與管理。

Code

該incremental-migration-sct-sql.zip文件（附件）包含此模式的完整源代碼。

史诗

針對來源資料庫結構描述差異建立 SQL 指令集檔案

任務	描述	所需技能
在 Oracle SQL 開發人員中執行資料庫差異。	1. 登入來源 Oracle 資料庫執行個體，選擇「工具」，然後選擇「資料庫差異」。 2. 在「來源連線」中選擇來源資料庫。 3. 在「目的地連線」中選擇更新或已修補的來源資料庫。 4. 根據您的需求設定其餘選項，選擇 [下一步]，然後選擇 [完成] 以產生差異報告。	DBA
產生 SQL 指令集檔案。	選擇「產生命令檔」以產生 SQL 檔案的差異。 這會產生 AWS SCT 用來將資料庫從甲骨文轉換為 PostgreSQL 的 SQL 指令碼檔案。	DBA

使用 Python 指令碼在 AWS SCT 中建立目標資料庫物件

任務	描述	所需技能
使用視窗命令提示字元設定 AWS SCT。	1. 從預先安裝的 AWS SCT 資料夾複製AWSSchema ConversionToolBatc	DBA

任務	描述	所需技能
	h.jar 檔案，並將其貼到您的工作目錄中。 2. 從文件incremental-migration-sct-sql.zip 夾中的run_aws_sct_sql.py 文件部署 Python 代碼（附加）。這會在目錄中建立 .xml 檔案和 .sct 檔案，其中包含您的來源和projects目標資料庫環境組態詳細資訊。它也會讀取您在「Oracle SQL 開發人員」中產生的 SQL 指令碼檔案。最後，它在output目錄中創建 .sql 文件對象。 3. 使用下列格式，在database_migration.txt 檔案中設定來源和目標環境組態詳細資訊： #source_vendor,source_hostname,source_dbname,source_user,source_pwd,source_schema,source_port,source_sid,target_vendor,target_hostname,target_user,target_pwd,target_dbname,target_port ORACLE,myoracledb.cokmvis0v46q.us-east-1.rds.amazonaws.com	

任務	描述	所需技能
	`` ,ORCL,orcl,orcl123 4,orcl,1521,ORCL,P OSTGRESQL,mypgdbin stance.cokmvis0v46 q.us-east-1.rds.am azonaws.com,pguser ,pgpassword,pgdb,5432 ``` 4. 根據您的需求修改 AWS SCT 組態參數，然後將 SQL 指令碼檔案複製到`input`子目錄中的工作目錄中。	
執行 Python 指令碼。	1. 使用下列命令執行 Python 指令碼：`$ python run_aws_sct_sql.py database_migration.txt` 2. 這將創建數據庫對象的 SQL 文件。可以手動轉換具有轉換錯誤的未轉換代碼。	DBA
在亞馬遜 RDS 中創建對象	在您的 Amazon RDS 資料庫執行個體中執行 SQL 檔案並建立物件。	DBA

相關資源

- Amazon RDS 上的甲骨文
- Amazon RDS 上的 PostgreSQL
- 使用 AWS SCT 使用者界面
- 使用甲骨文作為 AWS SCT 的來源

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在 Aurora 相容中使用檔案編碼將 BLOB 檔案載入至文字

由古迪瓦達 (AWS) 和吉萬·雪蒂 (AWS) 創建

環境：生產	來源：內部部署 Oracle 資料庫	目標：Aurora 郵政兼容
R 型：重新建築	工作負載：甲骨文; 開源	技術：移轉；資料庫
AWS 服務：Amazon Aurora		

Summary

通常在移轉期間，在某些情況下，您必須處理從本機檔案系統上的檔案載入的非結構化和結構化資料。資料也可能位於不同於資料庫字元集的字元集中。

這些文件包含以下類型的數據：

- 中繼資料 — 此資料描述檔案結構。
- 半結構化資料 — 這些是特定格式的文字字串，例如 JSON 或 XML。您可能能夠對此類數據進行斷言，例如「將始終以 '<' 開頭」或「不包含任何換行符」。
- 全文 — 此資料通常包含所有類型的字元，包括換行符和引號字元。它也可能由 UTF-8 中的多字節字符組成。
- 二進制數據-此數據可能包含字節或字節組合，包括空值和 end-of-file 標記。

加載這些類型的數據混合物可能是一個挑戰。

該模式可與現場部署 Oracle 資料庫、亞馬遜 Amazon Web Services (AWS) 雲端上的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上的 Oracle 資料庫，以及適用於 Oracle 資料庫的 Amazon Relational Database Service 服務 (Amazon RDS) 一起使用。作為一個例子，這種模式使用 Amazon Aurora PostgreSQL 兼容版本。

在 Oracle 數據庫中，借助`BFILE`（二進制文件）指針，`DBMS_LOB`包和 Oracle 系統函數，您可以從文件中加載並使用字符編碼轉換為 CLOB。由於 PostgreSQL 在遷移至 Amazon Aurora PostgreSQL 相容版本資料庫時不支援 BLOB 資料類型，因此這些函數必須轉換為與 PostgreSQL 相容的指令碼。

此模式提供兩種方法，可將檔案載入 Amazon Aurora PostgreSQL 相容資料庫中的單一資料庫資料行：

- 方法 1 — 您可以使用具有編碼選項的aws_s3擴充table_import_from_s3功能，從 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體匯入資料。
- 方法 2 — 您在數據庫外部編碼為十六進制，然後解碼以查看數據庫TEXT內部。

我們建議您使用方法 1，因為與 Aurora PostgreSQL 相容已與擴充功能直接整合。aws_s3

此模式使用範例，將包含具有多位元組字元和不同格式的電子郵件範本的平面檔案載入 Amazon Aurora PostgreSQL 相容資料庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon RDS 執行個體或 Aurora 與 PostgreSQL 相容的執行個體
- SQL 和關係數據庫管理系統（RDBMS）的基本了解
- Amazon Simple Storage Service (Amazon S3) 存儲桶。
- 甲骨文和 PostgreSQL 中的系統功能知識
- 轉速 Pack HexDump age-XXD-0.1.1（包含在 Amazon 2 中）

限制

- 對於TEXT數據類型，可以存儲的最長字符串約為 1 GB。

產品版本

- Aurora 支援 Amazon Aurora PostgreSQL 更新中列出的 Postgre SQL 版本。

架構

目標技術堆疊

- Aurora 郵政兼容

目標架構

方法 1 — 使用從 S3 匯入

從現場部署伺服器，會將包含多位元組字元和自訂格式的電子郵件範本的檔案傳輸到 Amazon S3。此模式提供的自訂資料庫函數使用的aws_s3.table_import_from_s3函數將檔案載入資料庫，並將查詢結果作為TEXT資料類型傳回。file_encoding

1. 檔案會傳輸到暫存 S3 儲存貯體。
2. 檔案會上傳到 Amazon Aurora PostgreSQL 相容資料庫。
3. 使用 PpgAdmin 用戶端，自訂功能load_file_into_clob會部署至 Aurora 資料庫。
4. 自定義函數內部使table_import_from_s3用文件編碼。從函數的輸出是通過使用array_to_string和array_agg作為TEXT輸出獲得的。

方法 2 — 將數據庫外部的十六進制編碼並解碼以查看數據庫內的 TEXT

來自內部部署伺服器或本機檔案系統的檔案會轉換為十六進位傾印。然後將文件作為字段TEXT導入到 PostgreSQL 中。

1. 使用xxd -p選項，在命令列中將檔案轉換為十六進位傾印。
2. 使用\copy選項將十六進位傾印檔案上傳至 Aurora PostgreSQL 相容，然後將十六進位傾印檔案解碼為二進位檔案。
3. 編碼要返回的二進制數據TEXT。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

其他工具

- PGadmin4 是 PostgreSQL 的開放原始碼管理與開發平台。pGadmin4 可以在 Linux、Unix、Mac 作業系統和視窗上使用來管理 PostgreSQL。

史诗

方法 1：將資料從 Amazon S3 匯入至 Aurora 與 PostgreSQL 相容

任務	描述	所需技能
啟動 EC2 執行個體。	如需啟動執行個體的指示，請參閱啟動執行個體。	DBA
安裝客戶端 pgAdmin 工具。	下載並安裝 pgAdmin。	DBA
建立 IAM 政策。	建立名為的 AWS Identity and Access Management (IAM) 政策`aurora-s3-access-pol` ，以授予存放檔案之 S3 儲存貯體的存取權。使用下面的代碼，替換`<bucket-name>` 為您的 S3 存儲桶的名稱。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "s3:GetObject", "s3:AbortMultipartUpload", "s3:DeleteObject", "s3:ListMultipartUploadParts", "s3:PutObject",`	DBA

任務	描述	所需技能
	<pre>"s3:ListBucket"], "Resource": ["arn:aws:s3:::<buc ket-name>/*", "arn:aws:s3:::<buc ket-name>"] }] }</pre>	
建立 IAM 角色，以將物件從 Amazon S3 匯入至相容 Aurora。	使用下列程式碼建立以AssumeRole信任關係命名`aurora-s3-import-role` 的 IAM 角色。`AssumeRole` 允許 Aurora 代表您存取其他 AWS 服務。 <pre>{ "Version": "2012-10- 17", "Statement": [{ "Effect": "Allow","Principal": { "Service": "rds.amazonaws.com" },"Action": "sts:AssumeRole" }] }</pre>	DBA

任務	描述	所需技能
將 IAM 角色與叢集建立關聯。	若要將 IAM 角色與 Aurora PostgreSQL 相容的資料庫叢集建立關聯，請執行下列 AWS CLI 命令。變更<Account-ID> 為託管 Aurora PostgreSQL 相容資料庫的 AWS 帳戶 ID。這可讓 Aurora 與 PostgreSQL 相容的資料庫存取 S3 儲存貯體。 `aws rds add-role-to-db-cluster --db-cluster-identifier aurora-postgres-cl --feature-name s3Import --role-arn arn:aws:iam::<Account-ID>:role/aurora-s3-import-role`	DBA
將範例上傳至 Amazon S3。	1. 在此模式的「其他資訊」區段中，將電子郵件範本程式碼複製到名為的檔案中`salary.event.notification.email.vm` 。 2. 將檔案上傳到 S3 儲存貯體。	DBA，應用程式擁有者

任務	描述	所需技能
部署自訂功能。	1. 從 [其他資訊] 區段中，將自訂函數 `load_file_into_clob` SQL 檔案內容複製到暫存資料表中。 2. 登入 Aurora PostgreSQL 相容的資料庫，並使用 pgAdmin 用戶端將其部署到資料庫結構描述。	應用程式擁有者，DBA
執行將資料匯入資料庫的自訂函數。	執行下列 SQL 命令，以適當的值取代尖括號中的項目。 `select load_file_into_clob('aws-s3-import-test'::text,'us-west-1'::text,'employee.salary.event.notification.email.vm'::text);` 在執行指令之前，請使用適當的值取代尖括號中的項目，如下列範例所示。 `Select load_file_into_clob('aws-s3-import-test'::text,'us-west-1'::text,'employee.salary.event.notification.email.vm'::text);` 該命令會從 Amazon S3 載入檔案，並將輸出傳回為TEXT。	應用程式擁有者，DBA

方法 2：在本地 Linux 系統中將模板文件轉換為十六進制轉儲

任務	描述	所需技能
將範本檔案轉換為十六進位傾印。	Hexdump 公用程式會以十六進位、十進位、八進位或 ASCII 顯示二進位檔案的內容。該hexdump命令是軟件util-linux 包的一部分，並預先安裝在 Linux 發行版中。六進制轉儲包也是 Amazon Linux 2 的一部分。 若要將檔案內容轉換為十六進位傾印，請執行下列 shell 命令。 `xxd -p </path/file.vm>  | tr -d '\n' > </path/file.hex>` 以適當的值取代路徑和檔案，如下列範例所示。 `xxd -p employee.salary.event.notification.email.vm |  tr -d '\n' > employee.salary.event.notification.email.vm.hex`	DBA
將十六進制轉儲文件加載到數據庫模式中。	使用下列命令，將十六進位傾印檔案載入 Aurora PostgreSQL 相容的資料庫。 1. 登入至 Aurora PostgreSQL 資料庫，並建立名為的	DBA

任務	描述	所需技能
	新資料表。email_template_hex `CREATE TABLE email_template_hex(hex_data TEXT);` 2. 通過使用以下命令將文件從本地文件系統加載到數據庫模式中。 `\copy email_template_hex FROM '/path/file.hex';` 將路徑取代為本機檔案系統上的位置。 `\copy email_template_hex FROM '/tmp/employee.salary.event.notification.email.vm.hex';` 3. 創建一個名為表email_template_bytea 。 `CREATE TABLE email_template_bytea(hex_data bytea);` 4. 將資料從插email_template_hex 入email_template_bytea 。	

任務	描述	所需技能
	`` INSERT INTO email_tem plate_bytea (hex_data) (SELECT decode(he x_data, 'hex') FROM email_template_hex limit 1); ``` 5. 要將十六進制 bytea 代碼作為TEXT數據返回，請運行以下命令。 ``` SELECT encode(he x_data::bytea, 'escape') FROM email_template_byt ea; ```	

相關資源

參考

- 使用 PostgreSQL 資料庫做為 AWS Database Migration Service 的目標
- 甲骨文資料庫 19c 到 Amazon Aurora 與 PostgreSQL 相容性 (12.4) 移轉教戰手冊
- 建立 IAM 政策
- 將 IAM 角色與 Amazon Aurora MySQL 資料庫叢集建立關聯
- pgAdmin

教學課程

- Amazon RDS 入門
- 從甲骨文遷移到 Amazon Aurora

其他資訊

自定義函數

```
CREATE OR REPLACE FUNCTION load_file_into_clob(
    s3_bucket_name text,
    s3_bucket_region text,
    file_name text,
    file_delimiter character DEFAULT '&'::bpchar,
    file_encoding text DEFAULT 'UTF8'::text)
    RETURNS text
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
DECLARE
    blob_data BYTEA;
    clob_data TEXT;
    l_table_name CHARACTER VARYING(50) := 'file_upload_hex';
    l_column_name CHARACTER VARYING(50) := 'template';
    l_return_text TEXT;
    l_option_text CHARACTER VARYING(150);
    l_sql_stmt CHARACTER VARYING(500);

BEGIN

    EXECUTE format ('CREATE TEMPORARY TABLE %I (%I text, id_serial serial)',
 l_table_name, l_column_name);

    l_sql_stmt := 'select ''(format text, delimiter ''''''' || file_delimiter || ''''''',
 encoding ''''''' || file_encoding ||  ''''''')'' ';

    EXECUTE FORMAT(l_sql_stmt)
    INTO l_option_text;

    EXECUTE FORMAT('SELECT aws_s3.table_import_from_s3($1,$2,$6,
 aws_commons.create_s3_uri($3,$4,$5))')
    INTO l_return_text
    USING l_table_name, l_column_name, s3_bucket_name,
 file_name,s3_bucket_region,l_option_text;

    EXECUTE format('select array_to_string(array_agg(%I order by id_serial),E''\n'')
 from %I', l_column_name, l_table_name)
    INTO clob_data;
```

```
    drop table file_upload_hex;

    RETURN clob_data;
END;
$BODY$;
```

電郵範本

```
##############################################################################################
##
 ##
##    johndoe Template Type: email
 ##
##    File: johndoe.salary.event.notification.email.vm
 ##
##    Author: Aimée Étienne    Date 1/10/2021
      ##
##  Purpose: Email template used by EmplmanagerEJB to inform a johndoe they         ##
##        have been given access to a salary event
 ##
##    Template Attributes:
 ##
##        invitedUser - PersonDetails object for the invited user
  ##
##        salaryEvent - OfferDetails object for the event the user was given access
 ##
##        buyercollege - CompDetails object for the college owning the salary event
 ##
##        salaryCoordinator - PersonDetails of the salary coordinator for the event
 ##
##        idp - Identity Provider of the email recipient
 ##
##        httpWebRoot - HTTP address of the server
 ##
##
 ##
##############################################################################################

$!invitedUser.firstname $!invitedUser.lastname,
```

```
Ce courriel confirme que vous avez ete invite par $!salaryCoordinator.firstname $!
salaryCoordinator.lastname de $buyercollege.collegeName a participer a l'evenement
 "$salaryEvent.offeringtitle" sur johndoeMaster Sourcing Intelligence.

Votre nom d'utilisateur est $!invitedUser.username

Veuillez suivre le lien ci-dessous pour acceder a l'evenement.

${httpWebRoot}/myDashboard.do?idp=$!{idp}

Si vous avez oublie votre mot de passe, utilisez le lien "Mot de passe oublie" situe
 sur l'ecran de connexion et entrez votre nom d'utilisateur ci-dessus.

Si vous avez des questions ou des preoccupations, nous vous invitons a
 communiquer avec le coordonnateur de l'evenement $!salaryCoordinator.firstname $!
salaryCoordinator.lastname au ${salaryCoordinator.workphone}.

*******

johndoeMaster Sourcing Intelligence est une plateforme de soumission en ligne pour les
 equipements, les materiaux et les services.

Si vous avez des difficultes ou des questions, envoyez un courriel a
 support@johndoeMaster.com pour obtenir de l'aide.
```

使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	資料來源：Amazon RDS for Oracle	目標：Amazon RDS
R 型：重新建築	工作負載：甲骨文; 開源	技術：移轉；安全性、身分識別、合規性；資料庫
AWS 服務：AWS DMS; Amazon RDS		

Summary

此模式提供有關將甲骨文資料庫執行個體的 Amazon Relational Database Service 服務 (Amazon RDS) 遷移到亞馬遜網路服務 (AWS) 雲端上的亞馬遜 RDS for PostgreSQL 資料庫的指導。若要加密資料庫之間的連線，該模式會在 Amazon RDS 和 AWS Database Migration Service (AWS DMS) 中使用憑證授權單位 (CA) 和 SSL 模式。

該模式描述具有大量交易的多 TB Oracle 來源資料庫的停機時間很少或沒有停機時間的線上移轉策略。為了資料安全起見，模式會在傳輸資料時使用 SSL。

此模式使用 AWS Schema Conversion Tool (AWS SCT) 將 Amazon RDS for Oracle 文資料庫結構描述轉換 Amazon RDS for PostgreSQL 架構。然後，該模式會使用 AWS DMS 將資料從 Amazon RDS for Oracle 文資料庫遷移到 Amazon RDS for PostgreSQL 資料庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon RDS 資料庫憑證授權單位 (CA) 僅設定使用 rds-ca-2019 (證書已於 2020 年 3 月 5 日到期)
- AWS SCT
- AWS DMS
- pgAdmin

- SQL 工具 (例如，SQL 開發人員或 SQL* 加號)

限制

- Amazon RDS for Oracle 文資料庫 — 最低要求是適用於甲骨文 19c 版的企業版和標準二版。
- Amazon RDS for PostgreSQL 資料庫的最低要求為 PostgreSQL 版本 12 及更新版本 (適用於 9.x 及更新版本)。

產品版本

- Amazon RDS for Oracle 文數據庫 12.1.0.2 版實例
- 亞馬遜 RDS 資料庫 11.5 版執行個體

架構

源, 技術, 堆棧

- 一個適用於甲骨文資料庫執行個體的亞馬遜 RDS，其版本為 12.1.0.2.v18。

目標技術堆疊

- AWS DMS
- 使用 11.5 版 Amazon RDS for PostgreSQL 資料庫執行個體。

目標架構

下圖顯示甲骨文（源）和 PostgreSQL（目標）數據庫之間的數據遷移架構的體系結構。該架構包括以下內容：

- 虛擬私有雲 (VPC)
- 可用區域
- 私有子網路
- 甲骨文數據庫的亞馬遜 RDS
- AWS DMS 複寫執行個體
- 適用於 PostgreSQL 資料庫的 RDS

若要加密來源和目標資料庫的連線，必須在 Amazon RDS 和 AWS DMS 中啟用 CA 和 SSL 模式。

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

其他服務

- pgAdmin 是一個開放原始碼管理 PostgreSQL。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

為甲骨文執行個體設定亞馬遜 RDS

任務	描述	所需技能
建立 Oracle 資料庫執行處理。	登入您的 AWS 帳戶，開啟 AWS 管理主控台，然後導覽至 Amazon RDS 主控台。在主控台上，選擇 [建立資料庫]，然後選擇 [Oracle]。	一般 AWS、DBA
設定安全性群組。	設定輸入和輸出安全性群組。	一般 AWS
建立選項群組。	在與 Amazon RDS for Oracle 資料庫相同的 VPC 人雲端和	一般 AWS

任務	描述	所需技能
	安全群組中建立選項群組。針對 [選項]，選擇 [SSL]。在 [連接埠] 中，選擇 2484 (適用於 SSL 連線)。	
設定選項設定。	請使用下列設定： • `SQLNET.CIPHER_SUITE`：`SSL_RSA_WITH_AES_256_CBC_SHA` • `SQLNET.SSL_VERSION`：`1.2 or 1.0`	一般 AWS
修改適用於 Oracle 資料庫執行個體的 RDS。	將 CA 憑證設定為 RDS-CA-2019。在「選項群組」下，貼附先前建立的選項群組。	DBA，一般 AWS

任務	描述	所需技能
確認適用於 Oracle 資料庫執行個體的 RDS 可用。	請確定適用於 Oracle 資料庫的 Amazon RDS 資料庫執行個體已啟動並且可以存取資料庫結構描述。 若要連接至 Oracle 資料庫的 RDS，請使用sqlplus命令列中的命令。 `$ sqlplus orcl/**** @myoracledb.cokmvis0v46q.us-east-1.rds.amazonaws.com:1521/ORCL` `SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 15 18:11:07 2019` `Copyright (c) 1982, 2016, Oracle. All rights reserved.` `Last Successful login time: Mon Dec 16 2019 23:17:31 +05:30` `Connected to:` `Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production` `With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options` `SQL>`	DBA
在適用於 Oracle 資料庫的 RDS 中建立物件和資料。	建立物件並在結構描述中插入資料。	DBA

設定適用於 PostgreSQL 的亞馬遜 RDS 執行個體

任務	描述	所需技能
建立適用於 PostgreSQL 的資料庫。	在 Amazon RDS 主控台建立資料庫頁面上，選擇 PostgreSQL 以建立 Amazon RDS for PostgreSQL 的資料庫執行個體。	DBA，一般 AWS
設定安全性群組。	設定輸入和輸出安全性群組。	一般 AWS
建立參數群組。	如果您使用的是 PostgreSQL 11.x 版，請建立參數群組以設定 SSL 參數。在 PostgreSQL 中，依預設會啟用 SSL 參數群組。	一般 AWS
編輯參數。	將`rds.force_ssl` 參數變更為`1`（開啟）。 依預設，`ssl`參數為 1 (開啟)。將`rds.force_ssl` 參數設定為`1`，您可以強制所有連線僅透過 SSL 模式進行連線。	一般 AWS
修改 RDS 資料庫執行個體。	將 CA 憑證設定為 RDS-CA-2019。根據您的 PostgreSQL 版本，附加預設參數群組或先前建立的參數群組。	DBA，一般 AWS
確認適用於 PostgreSQL 的資料庫執行個體可用。	請確定 Amazon RDS for PostgreSQL 的資料庫已啟動並執行。 該`psql`命令建立了從命令行`sslmode`設置的 SSL 連接。	DBA

任務	描述	所需技能
	一個選項是在參數群組sslmode=1 中進行設定，並使用psql連接，而不將sslmode參數包括在指令中。 下面的輸出顯示 SSL 連接已建立。 `$ psql -h mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com -p 5432 "dbname=pgdb user=pguser"` `Password for user pguser:` `psql (11.3, server 11.5)` `SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)` `Type "help" for help.` `pgdb=>` 第二個選項是sslmode=1 在參數群組中設定，並將sslmode參數包括在指psql令中。 下面的輸出顯示 SSL 連接已建立。 `$ psql -h mypgdbinstance.cokmvis0v46q.us-east-1.rds.ama`	

任務	描述	所需技能
	zonaws.com -p 5432 "dbname=pgdb user=pgus er sslmode=require" Password for user pguser: psql (11.3, server 11.5) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA- AES256-GCM-SHA384, bits: 256, compressi on: off) Type "help" for help. pgdb=>	

設定和執行 AWS SCT

任務	描述	所需技能
安裝 AWS SCT。	安裝最新版本的 AWS SCT 應用程式。	一般 AWS
使用 JDBC 驅動程式設定 AWS SCT。	下載適用於甲骨 PostgreSQL 和驅動程序的 Java 數據庫連接性（JDBC）驅動程序。 ojdbc8.jar 若要在 AWS SCT 中設定驅動程式，請選擇設定、全域設定、驅動程式。	一般 AWS
建立 AWS SCT 專案。	使用 Oracle 做為來源資料庫引擎，使用亞馬遜 RDS 作為目標資料庫引擎，建立 AWS SCT 專案和報告：	一般 AWS

任務	描述	所需技能
	1. 透過提供連線詳細資料，測試與來源 Oracle 資料庫的連線，以及以 Amazon RDS for PostgreSQL 資料庫的目標。 對於來源 Oracle 資料庫，需要下列權限或權限： • `CONNECT` • `SELECT_CATALOG_ROLE` • `SELECT ANY DICTIONARY` • `SELECT on SYS.USER$ TO <sct_user>` 如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS SCT 的來源。 來源和目標連線都必須成功，AWS SCT 才能啟動遷移報告。 2. 在報表之後，輸入要轉換的結構描述，然後選擇「完成」。	

任務	描述	所需技能
驗證數據庫對象。	1. 選擇 [載入綱要]。 AWS SCT 會顯示來源和轉換後的目標物件，包括有錯誤的物件。更新目標資料庫上任何不正確的物件。 2. 檢閱錯誤，並使用手動介入來清除錯誤。 3. 清除所有錯誤之後，再次選擇「載入結構描述」。 4. 選擇「套用至資料庫」。 5. Connect 到 PGAdmin，或任何支持 PostgreSQL 數據庫連接的工具，並檢查模式和對象。	DBA，一般 AWS

設定和執行 AWS DMS

任務	描述	所需技能
建立複寫執行個體。	1. 登入您的帳戶、開啟 AWS 管理主控台，然後瀏覽至 AWS DMS 主控台。 2. 使用 VPC、安全群組、可用區域和額外連線屬性的有效設定建立複寫執行個體。	一般 AWS
匯入憑證。	1. 下載 RDS-鈣 -2019 根證書. 2. 在 [憑證] 頁面上，將憑證匯入為`rds-ca-2019-root`。	一般 AWS
建立來源端點。	1. 選擇選取 RDS 資料庫執行個體，然後選取您建立的	一般 AWS

任務	描述	所需技能
	RDS 適用於 Oracle 資料庫執行個體，建立適用於 Oracle 的來源端點。端點組態詳細資料將會自動填入。 2. 選擇「手動提供存取資訊」。對於「連接埠」，請確定您輸入 2484。 3. 在安全通訊端層 (SSL) 模式下，選擇驗證 CA，，然後選擇您先前建立的 CA 憑證。 4. 在「端點設定」 下，新增額外的連線屬性`NumberDataTypeScale=-2` 以支援無大小的`NUMBER`資料類型。 如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS Database Migration Service 的來源。	

任務	描述	所需技能
建立目標端點。	1. 選擇選取 RDS 資料庫執行個體，然後選取您的 RDS 資料庫執行個體，建立適用於 PostgreSQL 的目標端點。端點組態詳細資料將會自動填入。 2. 選擇「手動提供存取資訊」。對於「連接埠」，請確定您輸入 2484。 如需詳細資訊，請參閱使用 PostgreSQL 資料庫做為 AWS Database Migration Service 的目標。	一般 AWS
測試端點。	1. 測試來源端點和目標端點，以確認兩者都成功且可用。 2. 如果測試失敗，請確定安全群組輸入規則有效。	一般 AWS

任務	描述	所需技能
建立移轉工作。	若要建立完整載入和變更資料擷取 (CDC) 或進行資料驗證的移轉任務，請執行下列動作： 1. 若要建立資料庫移轉工作，請選擇複寫執行個體、來源資料庫端點、目標資料庫端點。將移轉類型指定為下列其中一項： • 移轉現有資料 (滿載) • 僅複製資料變更 (CDC) • 遷移現有資料並複寫持續中的變更 (滿載和 CDC) 2. 在 [資料表對應] 底下，您可以設定 GUI 或 JSON 格式的選取規則和轉換規則： • 在「選取規則」下，選取結構描述、輸入表格名稱，然後選取要設定的動作 (包括或排除)，例如「綱要 ORCL」、「表格名稱%」、「動作包含」。 • 在「轉換規則」下，執行下列其中一項作業： • 選取結構描述並選擇動作 (大小寫、首碼、尾碼)；例如，「目標綱要 ORCL」、「動作」「使用小寫」。 • 選取結構描述，輸入表格名稱，然後選擇動作 (大小寫、首碼、尾	一般 AWS

任務	描述	所需技能
	碼)；例如，「目標結構描述 ORCL」、「表格 %」、「動作」「使用小寫」。 3. 開啟 Amazon CloudWatch 日誌監控。 4. 對於映射規則，請添加以下 JSON 代碼。 `{ "rules": [ { "rule-type": "transformation", "rule-id": "1", "rule-name": "1", "rule-target": "table", "object-locator": { "schema-name": "%", "table-name": "%" }, "rule-action": "convert-lowercase", "value": null, "old-value": null }, { "rule-type": "transformation",`	

任務	描述	所需技能
	<pre>"rule-id" : "2", "rule-name": "2", "rule-target": "schema", "object-locator": { "schema-name": "ORCL", "table-name": "%" }, "rule-action": "convert-lowercase", "value": null, "old-value": null }, { "rule-type": "selection", "rule-id" : "3", "rule-name": "3", "object-locator": { "schema-name": "ORCL", "table-name": "DEPT" }, "rule-action": "include", "filters" : [] }</pre>	

任務	描述	所需技能
	] }	
計劃生產執行。	與應用程式擁有者等利益相關者確認停機時間，以便在生產系統中執行 AWS DMS。	遷移, 領導

任務	描述	所需技能
執行 遷移任務。	1. 啟動狀態 CloudWatch 為「就緒」的 AWS DMS 任務，並在 Amazon 中監控遷移任務日誌是否有任何錯誤。 如果您選擇移轉現有資料並複寫進行中的變更做為移轉類型，且狀態為 [載入完成持續複寫]，則 CDC 資料移轉的完整負載會完成，並持續進行驗證。 2. 開始移轉之後，您可以在中取得其他 SSL 連線資訊。CloudWatch對於 Oracle，CloudWatch 顯示以下連接字符串。 `2019-12-17T09:15:11 [SOURCE_UNLOAD ]I: Connecting to Oracle: Beginning session (oracle_endpoint_conn.c:834)` PostgreSQL 的連接字符串將類似於下面的例子。 `2019-12-17T09:15:11 [TARGET_LOAD ]I: Going to connect to ODBC connection string: PROTOCOL=7.4-0;DRIVER={Post`	一般 AWS

任務	描述	所需技能
	greSQL};SERVER=mypgdbinstance.cokmvis0v46q.us-east-1.rds.amazonaws.com;DATABASE=pgdb;PORT=5432;sslmode=require;UID=pguser;(odbc_endpoint_imp.c:2218)	
驗證資料。	檢閱來源 Oracle 和目標 PostgreSQL 資料庫中的移轉任務結果和資料： 1. Connect 到 pgAdmin，並使用模式檢查 PostgreSQL 數據庫中的數據。ORCL 2. 若是 CDC，請在來源 Oracle 資料庫中插入或更新資料，以檢查進行中的變更。	DBA
停止移轉工作。	成功完成資料驗證後，請停止移轉工作。	一般 AWS

清理資源

任務	描述	所需技能
刪除 AWS DMS 任務。	1. 在 AWS DMS 主控台上，導覽至資料庫遷移任務，並停止任何進行中或執行中的 AWS DMS 任務。	一般 AWS

任務	描述	所需技能
	2. 選取一或多個工作，選擇「動作」，然後選擇「刪除」。	
刪除 AWS DMS 端點。	選取您建立的來源和目標端點，然後選擇「動作」，然後選擇「刪除」。	一般 AWS
刪除 AWS DMS 複寫執行個體。	選擇複製執行個體，選擇 [動作]，然後選擇 [刪除]。	一般 AWS
刪除 PostgreSQL 庫。	1. 在 Amazon RDS 主控台上，選擇資料庫。 2. 選取您建立的 PostgreSQL 資料庫執行個體，選擇 [動作]，然後選擇 [刪除]。	一般 AWS
刪除 Oracle 資料庫。	在 Amazon RDS 主控台上，選取 Oracle 資料庫執行個體，選擇「動作」，然後選擇「刪除」。	一般 AWS

故障診斷

問題	解決方案
AWS SCT 來源和目標測試連線失敗。	設定 JDBC 驅動程式版本和 VPC 安全群組輸入規則，以接受傳入流量。
Oracle 來源端點測試執行失敗。	檢查端點設定以及複製執行個體是否可用。
AWS DMS 任務完全負載執行失敗。	檢查來源和目標資料庫是否具有相符的資料類型和大小。
AWS DMS 驗證移轉任務會傳回錯誤。	1. 檢查表是否有主鍵。沒有主鍵的表不會被驗證。

問題	解決方案
	2. 如果表具有主鍵但返回錯誤，請檢查源端點中的額外連接屬性。額外的連接屬性必 numberDataTypeScale=-2 須支持沒有大小動態基於表中可用的數據的數據類型。NUMBER

相關資源

資料庫

- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL

SSL 數據庫連接

- 使用 SSL/TLS 加密與資料庫執行個體的連線
  - 將 SSL 與 Oracle 資料庫執行個體的 RDS 搭配使用
  - 使用 SSL/TLS 保護連線安全
  - 下載 CA-2019 根憑證
- 使用選項群組
  - 新增選項至 Oracle 資料庫執行個體
  - Oracle 安全通訊端層
- 使用參數群組
- PostgreSQL 式連接參數
- 使用來自 JDBC 的 SSL

AWS

- AWS Schema Conversion Tool
- AWS Schema Conversion Tool 使用者指南
- 使用 AWS SCT 使用者界面
- 使用 Oracle 資料庫做為 AWS SCT 的來源

AWS DMS

- AWS Database Migration Service
- AWS Database Migration Service 使用者指南
  - 使用 Oracle 資料庫做為 AWS DMS 的來源
  - 使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標
- 搭配 AWS Database Migration Service 使用 SSL
- 移轉執行關聯式資料庫的應用程 AWS

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS SCT 和 AWS DMS 將適用於甲骨文的亞馬遜 RDS 遷移到適用於 PostgreSQL 的 CLI 馬遜 RDS CloudFormation

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	資料來源：Amazon RDS for Oracle	目標：Amazon RDS for PostgreSQL
R 型：重新建築	工作負載：甲骨文; 開源	技術：移轉；資料庫
AWS 服務：AWS DMS; Amazon RDS; AWS SCT		

Summary

此模式顯示如何使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 將適用於甲骨文資料庫執行個體的多 TB Amazon Relational Database Service 服務 (Amazon RDS) 遷移到適用於 PostgreSQL 的資料庫執行個體。此方法提供最短的停機時間，而且不需要登入 AWS 管理主控台。

使用 AWS Schema Conversion Tool (AWS SCT) 和 AWS Database Migration Service (AWS DMS) 主控台，此模式有助於避免手動設定和個別遷移。此解決方案會為多個資料庫設定一次性組態，並在 AWS CLI 上使用 AWS SCT 和 AWS DMS 來執行遷移。

該模式使用 AWS SCT 將資料庫架構物件從亞馬遜 RDS 轉換為適用 Amazon RDS for PostgreSQL，然後使用 AWS DMS 遷移資料。使用 AWS CLI 中的 Python 指令碼，您可以使用 AWS 範本建立 AWS SCT 物件和 AWS DMS 任務。 CloudFormation

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 適用於甲骨文資料庫執行個體的現有 Amazon RDS。
- 現有的亞馬遜 RDS 資料庫執行個體。
- 具有 Windows 或 Linux 作業系統的 Amazon EC2 執行個體或本機電腦，用於執行指令碼。
- 瞭解下列 AWS DMS 移轉任務類型：`full-load`、`cdc`、`full-load-and-cdc`。 如需詳細資訊，請參閱 AWS DMS 文件中的建立任務。

- 使用適用於甲骨文和 PostgreSQL 資料庫引擎的 Java 資料庫連接 (JDBC) 驅動程式進行安裝和設定的 AWS SCT。如需詳細資訊，請參閱 AWS SCT 文件中的安裝 AWS SCT 和安裝所需的資料庫驅動程式。
- 已安裝 AWS SCT 資料夾中的`AWSSchemaConversionToolBatch.jar`檔案，已複製到您的工作目錄。

- `cli-sct-dms-cft.zip`文件（附加），下載並提取到您的工作目錄中。

- 最新的 AWS DMS 複寫執行個體引擎版本。如需詳細資訊，請參閱 AWS Support 文件中的如何建立 AWS DMS 複寫執行個體，以及 AWS DMS 文件中的 AWS DMS 3.4.4 版本說明。
- AWS CLI 第 2 版，使用您的存取金鑰 ID、秘密存取金鑰和執行指令碼的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體或作業系統 (OS) 的預設 AWS 區域名稱進行安裝和設定。如需詳細資訊，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI 第 2 版和設定 AWS CLI。
- 熟悉 AWS CloudFormation 範本。如需詳細資訊，請參閱 AWS CloudFormation 文件中的 AWS CloudFormation 概念。
- Python 版本 3，在執行指令碼的亞馬遜 EC2 執行個體或作業系統上安裝和設定。如需詳細資訊，請參閱 Python 文件。

限制

- 適用於 Oracle 資料庫的來源 Amazon RDS 執行個體的最低需求為：
  - 甲骨文版本 12c（v12.1.0.2，版 12.2.0.1），18c（v18.0.0.0）和 19c（版 19.0.0.0），適用於企業版，標準版，標準版和標準二版。
  - 雖然 Amazon RDS 支援甲骨文 18c (v18.0.0.0)，但此版本處於淘汰的路徑上，因為甲骨文在該日期之後不再提供 18c 的修補程式。 end-of-support 如需詳細資訊，請參閱 Amazon RDS 文件中的甲骨文。
  - Amazon RDS for Oracle 11g 不再受支持。
- 適用於 PostgreSQL 資料庫執行個體的目標 Amazon RDS 資料庫執行個體的最低需求為：
  - 第 9 版 PostgreSQL 版本 9.5 和 9.6 版)、第 10.x 版、11.x 版、第 12 版和第 13.x 版

產品版本

- Amazon RDS for Oracle 文資料庫執行個體 12.1.0.2 及更新版本
- Amazon RDS for PostgreSQL 庫執行個體 11.5 版及更新版本

- AWS CLI 第 2 版
- 最新版本的 AWS SCT
- Python 3 的最新版本

架構

源, 技術, 堆棧

- Amazon RDS for Oracle

目標技術堆疊

- Amazon RDS for PostgreSQL

來源與目標架構

下圖顯示使用 AWS DMS 和 Python 指令碼將 Amazon RDS for Oracle 文資料庫執行個體遷移到 Amazon RDS 適用於 PostgreSQL 的資料庫執行個體。

此圖表顯示下列移轉工作流程：

1. Python 指令碼使用 AWS SCT 連線到來源和目標資料庫執行個體。
2. 使用者使用 Python 指令碼啟動 AWS SCT，將甲骨文程式碼轉換為 PostgreSQL 程式碼，然後在目標資料庫執行個體上執行該程式碼。
3. Python 指令碼會為來源和目標資料庫執行個體建立 AWS DMS 複寫任務。
4. 使用者部署 Python 指令碼以啟動 AWS DMS 任務，然後在資料遷移完成後停止任務。

自動化和規模

您可以通過在單個程序中向 Python 腳本中添加多個功能的其他參數和安全相關更改來自動執行此遷移。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。這種模式使用 Python 腳本將 .csv 輸入文件轉換為 .json 輸入文件。在 AWS CLI 命令中使用 .json 檔案來建立 AWS 堆疊，該 CloudFormation 堆疊會使用 Amazon 資源名稱 (ARN)、遷移類型、任務設定和表格對映建立多個 AWS DMS 複寫任務。
- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。此模式使用 AWS DMS 透過命令列執行的 Python 指令碼建立、啟動和停止任務，並建立 AWS CloudFormation 範本。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。此模式需要已安裝 AWS SCT 目錄中的`AWSSchemaConversionToolBatch.jar`檔案。

Code

該`cli-sct-dms-cft.zip`文件（附件）包含此模式的完整源代碼。

史诗

在 AWS CLI 中設定 AWS SCT 並建立資料庫物件

任務	描述	所需技能
將 AWS SCT 設定為從 AWS CLI 執行。	1. 使用下列格式，在`database_migration.txt` 檔案中設定來源和目標環境組態詳細資訊： `#source_vendor,source_hostname,source_dbname,source_user,source_pwd,source_schema,source_port,source_sid,target_vendor,target_hostname,target_user,target_pwd,target_dbname,target_port` `ORACLE,myoracledb.cokmvis0v46q.us-east-1.rds.amazo`	DBA

任務	描述	所需技能
	naws.com,ORCL,orcl ,orcl1234,orcl,152 1,ORCL,POSTGRESQL, mypgdbinstance.cok mvis0v46q.us-east- 1.rds.amazonaws.co m,pguser,pgpasswor d,pgdb,5432 2. 根據下列檔案中的需求修改 AWS SCT 組態參數：project_settings.xml Oracle_PG_Test_Batch.xml 、和ORACLE-orcl-to-POSTGRESQL.xml 。	
執行 run_aws_sct.py Python 本。	使用下列命令執行 run_aws_sct.py Python 指令碼： $ python run_aws_sct.py database_migration.txt Python 腳本將數據庫對象從甲骨文轉換為 PostgreSQL，並創建在 PostgreSQL 格式的 SQL 文件。此指令碼也會建立 Database migration assessment report .pdf 檔案，以提供資料庫物件的詳細建議和轉換統計資料。	DBA

任務	描述	所需技能
在 Amazon RDS for PostgreSQL。	1. 必要時，手動修改 AWS SCT 產生的 SQL 檔案。 2. 在您的 Amazon RDS 資料庫執行個體中執行 SQL 檔案並建立物件。	DBA

使用 AWS CLI 和 AWS 設定和建立 AWS DMS 任務 CloudFormation

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。	登入 AWS 管理主控台，開啟 AWS DMS 主控台，然後建立根據您的需求設定的複寫執行個體。 如需詳細資訊，請參閱 AWS DMS 文件中的建立複寫執行個體和 AWS Support 文件中的如何建立 AWS DMS 複寫執行個體。	DBA
建立來源端點。	在 AWS DMS 主控台上，選擇端點，然後根據您的需求為 Oracle 資料庫建立來源端點。 注意：額外的連接屬性必須是`numberDataTypeScale` 一個`-2`值。 如需詳細資訊，請參閱 AWS DMS 文件中的建立來源和目標端點。	DBA

任務	描述	所需技能
建立目標端點。	在 AWS DMS 主控台上，選擇端點，然後根據您的需求為 PostgreSQL 資料庫建立目標端點。 如需詳細資訊，請參閱 AWS DMS 文件中的建立來源和目標端點。	DevOps 工程師
設定要從 AWS CLI 執行的 AWS DMS 複寫詳細資訊。	使用下列格式，使用來源端點 ARN、目標端點 ARN 和複寫執行個體 ARN，在`dms-arn-list.txt` 檔案中設定 AWS DMS 來源和目標端點以及複寫詳細資訊： `#sourceARN,targetARN,repARN` `arn:aws:dms:us-east-1:123456789012:endpoint:EH7AINRUDZ5GOYIY6HVMXECMCQ` `arn:aws:dms:us-east-1:123456789012:endpoint:HHJVUV57N7O3CQF4PJZKGIOYY5` `arn:aws:dms:us-east-1:123456789012:rep:LL57N77AQQAHHJF4PJFHNEDZ5G`	DBA

任務	描述	所需技能
執行 dms-create-task .py 指 Python 碼以建立 AWS DMS 任務。	1. 使用下列命令執行 dms-create-task.py Python 指令碼： `$ python dms-create-task.py database_migration.txt dms-arn-list.txt <cft-stack-name> <migration-type>` • `database_migration.txt` 是資料庫移轉文字檔 • `dms-arn-list.txt` 是 AWS DMS 的 ARN 清單 • `<cft-stack-name>` 是使用者定義的 AWS CloudFormation 堆疊名稱 • `<migration-type>` 是移轉類型 (全負載、cdc 或) full-load-and-cdc 2. 視您的遷移類型而定，您可以使用下列命令建立三種類型的 AWS DMS 任務： • `$ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack full-load` • `$ python dms-create-task.py database_`	DBA

任務	描述	所需技能
	migration.txt dms-arn-list.txt dms-cli-cft-stack cdc • $ python dms-create-task.py database_migration.txt dms-arn-list.txt dms-cli-cft-stack full-load-and-cdc 3. 已建立 AWS CloudFormation 堆疊和 AWS DMS 任務	
檢查 AWS DMS 任務是否已準備就緒。	在 AWS 主控台中，在「狀態」區段中檢查 AWS DMS 任務是否處於Ready狀態。	DBA

使用 AWS CLI 啟動和停止 AWS DMS 任務

任務	描述	所需技能
啟動 AWS DMS 任務。	使用下列命令執行 dms-start-task.py Python 指令碼： $ python dms-start-task.py start '<cdc-start-datetime>' 附註：開始日期和時間必須為'DD-MON-YYYY' 或時間'YYYY-MM-DDTHH:MI:SS' 戳記資料類型格式 (例如，'01-Dec-2	DBA

任務	描述	所需技能
	019’ 或‘2018-03-08T12:12:12’) 您可以在 AWS DMS 主控台的「任務」頁面上，在遷移任務的「表格統計資料」索引標籤中查看 AWS DMS 任務狀態。	
驗證資料。	1. 完成全負載移轉之後，工作會持續執行，以進行持續的資料變更 (CDC)。 2. 當 CDC 完成或不需要移轉其他變更時，請檢閱並驗證 Oracle 和 PostgreSQL 資料庫中的移轉任務結果和資料。 3. 您可以在 AWS DMS 主控台的任務頁面上 **Validation state Validation pending Validation failed Validation suspended** ，在資料庫遷移任務的 [表格統計資料] 索引標籤中檢查狀態和計數欄 (、、、和**Validation details**) 來驗證資料。 如需詳細資訊，請參閱 AWS DMS 文件中的 AWS DMS 資料驗證。	DBA

任務	描述	所需技能
停止 AWS DMS 任務。	使用下列命令執行 Python 指令碼： `$ python dms-start-task.py stop` 注意：根據驗證`failed`狀態，AWS DMS 任務可能會以狀態停止。如需詳細資訊，請參閱 < 其他資訊 > 一節中的疑難排解表格。	DBA

故障診斷

問題	解決方案
AWS SCT 來源和目標測試連線失敗	設定 JDBC 驅動程式版本和 VPC 安全群組輸入規則，以接受傳入流量。
來源或目標端點測試回合失敗	檢查端點設定和複製執行個體是否處於`Available` 狀態。檢查端點連線狀態是否為`Successful` 。 如需詳細資訊，請參閱 AWS Support 文件中的如何對 AWS DMS 端點連線故障進行疑難排解。
完全負載運行失敗	檢查來源和目標資料庫是否具有相符的資料類型和大小。 如需詳細資訊，請參閱 AWS DMS 文件中的 AWS DMS 中的移轉任務疑難排解。
驗證執行錯誤	檢查表是否有一個主鍵，因為非主鍵表沒有被驗證。

問題	解決方案
	如果資料表有主索引鍵和錯誤，請檢查來源端點中的額外連線屬性是否具有numberDataTypeScale=-2 。 如需詳細資訊，請參閱使用 Oracle 做為 AWS DMS 來源時的額外連線屬性 OracleSettings，以及 AWS DMS 文件中的疑難排解。

相關資源

- 安裝 AWS SCT
- AWS DMS 簡介 (影片)
- 在 AWS 中使用 AWS CLI CloudFormation
- 使用 AWS SCT 使用者界面
- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 使用甲骨文作為 AWS SCT 的來源
- 使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標
- AWS DMS 中資料遷移的來源
- AWS DMS 中資料遷移的目標
- 雲形 (AWS CLI 文件)
- 雲形建立堆疊 (AWS CLI 文件)
- dms (AWS CLI 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

將甲骨文序列遷移 _ 可重複使用的編譯包到 PostgreSQL

創建者：維奈帕拉迪 (AWS)

環境：PoC 或試點	來源：甲骨文數據庫	目標：PostgreSQL
R 型：重新建築	工作負載：甲骨文; 開源	技術：移轉；資料庫
AWS 服務：AWS SCT；Amazon Aurora		

Summary

這種模式提供了一 step-by-step 種方法，用於將定義為可重複使用編譯的甲骨文軟件包遷移到 Amazon Web Services（AWS）上的 PostgreSQL。這種方法維護了 SERIALLY_ 可重複使用編譯的功能。

PostgreSQL 不支持軟件包的概念和可重複使用的編譯。若要在 PostgreSQL 中取得類似的功能，您可以建立套件的結構描述，並在結構描述內部署所有相關物件 (例如函數、程序和類型)。為了實現 SERIALLY_RELEY 編譯指令碼的功能，此模式中提供的範例包裝函數指令碼使用 AWS Schema Conversion Tool (AWS SCT) 擴充套件。

有關更多信息，請參閱 Oracle 文檔中的「可重複使用編譯」。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 最新版本的 AWS SCT 和所需的驅動程式
- 適用於 PostgreSQL 資料庫的 Amazon Aurora 版本資料庫或 Amazon Relational Database Service 服務 (Amazon RDS)

產品版本

- Oracle 資料庫版本 10g 及更新版本

架構

源, 技術, 堆棧

- 內部部署的 Oracle 資料庫

目標技術堆疊

- Aurora 兼容或 Amazon RDS for PostgreSQL
- AWS SCT

移轉架構

工具

AWS 服務

- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。
- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。

其他工具

- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。

史诗

使用 AWS SCT 遷移甲骨文套件

任務	描述	所需技能
設定 AWS SCT。	設定 AWS SCT 連線至來源資料庫。如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS SCT 的來源。	DBA, 開發人員
轉換腳本。	使用 AWS SCT 將目標資料庫選取為與 Aurora PostgreSQL 相容的方式來轉換 Oracle 套件。	DBA, 開發人員
保存 .sql 文件。	儲存 .sql 檔案之前，請將 AWS SCT 中的專案設定選項修改為每個階段的單一檔案。AWS SCT 會根據物件類型將 .sql 檔案分割成多個 .sql 檔案。	DBA, 開發人員
更改代碼。	開啟 AWS SCT 產生的`init`函數，然後變更它，如其他資訊一節中的範例所示。它會添加一個變量來實現的功能`pg_serialize = 0` 。	DBA, 開發人員
測試轉換。	將`init`函數部署到與 Aurora PostgreSQL 相容的資料庫，並測試結果。	DBA, 開發人員

相關資源

- AWS Schema Conversion Tool
- Amazon RDS
- Amazon Aurora 功能

- 序列 _ 可重複使用的編譯指令

其他資訊

```
Source Oracle Code:

CREATE OR REPLACE PACKAGE test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
PROCEDURE function_1
 (test_id number);
PROCEDURE function_2
 (test_id number
 );
END;

CREATE OR REPLACE PACKAGE BODY test_pkg_var
IS
PRAGMA SERIALLY_REUSABLE;
v_char VARCHAR2(20) := 'shared.airline';
v_num number := 123;

PROCEDURE function_1(test_id number)
IS
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
v_char:='test1';
function_2(0);
END;

PROCEDURE function_2(test_id number)
is
begin
dbms_output.put_line( 'v_char-'|| v_char);
dbms_output.put_line( 'v_num-'||v_num);
END;
END test_pkg_var;

Calling the above functions

set serveroutput on
```

```
EXEC test_pkg_var.function_1(1);


EXEC test_pkg_var.function_2(1);


Target Postgresql Code:


CREATE SCHEMA test_pkg_var;

CREATE OR REPLACE FUNCTION test_pkg_var.init(pg_serialize IN INTEGER DEFAULT 0)

RETURNS void
AS
$BODY$

DECLARE

BEGIN

if aws_oracle_ext.is_package_initialized( 'test_pkg_var' ) AND pg_serialize = 0

then

return;

end if;

PERFORM aws_oracle_ext.set_package_initialized( 'test_pkg_var' );

PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',
 'shared.airline.basecurrency'::CHARACTER

VARYING(100));

PERFORM aws_oracle_ext.set_package_variable('test_pkg_var', 'v_num', 123::integer);

END;

$BODY$

LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION test_pkg_var.function_1(pg_serialize int default 1)

RETURNS void
AS

$BODY$
DECLARE

BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');

raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

PERFORM aws_oracle_ext.set_package_variable( 'test_pkg_var', 'v_char',
 'test1'::varchar);

PERFORM test_pkg_var.function_2(0);
END;

$BODY$
LANGUAGE plpgsql;


CREATE OR REPLACE FUNCTION test_pkg_var.function_2(IN pg_serialize integer default 1)

RETURNS void

AS

$BODY$

DECLARE

BEGIN

PERFORM test_pkg_var.init(pg_serialize);

raise notice 'v_char%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_char');
```

```
raise notice 'v_num%',aws_oracle_ext.get_package_variable( 'test_pkg_var', 'v_num');

END;
$BODY$
LANGUAGE plpgsql;


Calling the above functions

select test_pkg_var.function_1()

 select test_pkg_var.function_2()
```

將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容

由阿努拉達奇塔（AWS）和拉格什拉格夫（AWS）創建

環境：PoC 或試點	來源：甲骨文	目標：Aurora
R 型：重新建築	工作負載：開源	技術：移民、資料庫、現代化
AWS 服務：AWS Identity and Access Managemen t；AWS Lambda；Amazon S3；Amazon SNS；Amazon Aurora		

Summary

外部資料表可讓 Oracle 查詢儲存在純資料庫外部的資料。您可以使用 ORACLE_LOADER 驅動程式存取 SQL*Loader 公用程式可載入的任何格式儲存的任何資料。您無法在外部資料表上使用資料操縱語言 (DML)，但可以使用外部資料表進行查詢、聯結和排序作業。

Amazon Aurora PostgreSQL 兼容版本不提供與甲骨文中的外部表類似的功能。相反地，您必須使用現代化來開發符合功能需求且節儉的可擴充解決方案。

此模式提供了使用擴展程序將不同類型的甲骨文外部表遷移到 Amazon Web Services（AWS）雲上的 Aurora PostgreSQL 兼容版本的步驟。aws_s3

我們建議您在生產環境中實作此解決方案之前，先徹底測試此解決

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS 命令列界面 (AWS CLI)
- 可用的 Aurora 與 PostgreSQL 相容的資料庫執行個體。
- 具有外部表格的內部部署 Oracle 資料庫
- 客戶端 API

- 資料檔案

限制

- 此模式不提供作為 Oracle 外部表格取代的功能。但是，可以進一步增強步驟和示例代碼，以實現數據庫現代化目標。
- 文件不應包含在導aws_s3出和導入功能中作為分隔符傳遞的字符。

產品版本

- 若要從 Amazon S3 匯入 RDS 版 PostgreSQL 資料庫必須執行 PostgreSQL 或更新版本。

架構

源, 技術, 堆棧

- Oracle

來源架構

目標技術堆疊

- Amazon Aurora 郵政兼容
- Amazon CloudWatch
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)

目標架構

下圖顯示解決方案的高階表示法。

1. 檔案會上傳至 S3 儲存貯體。
2. 啟動 Lambda 函數。
3. Lambda 函數啟動數據庫函數調用。
4. Secrets Manager 提供資料庫存取的認證。
5. 根據 DB 功能，會建立 SNS 警示。

自動化和規模

對外部表格的任何新增或變更都可以使用中繼資料維護來處理。

工具

- 與亞馬遜 Aurora PostgreSQL 相容 — 與 Amazon Aurora PostgreSQL 相容的版本是完全受管、與 PostgreSQL 相容且符合 ACID 標準的關聯式資料庫引擎，結合了高階商業資料庫的速度和可靠性，以及開放原始碼資料庫的成本效益。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是管理 AWS 服務的統一工具。只要下載和設定一個工具，您就可以從命令列控制多個 AWS 服務，並透過指令碼自動化這些服務。
- Amazon CloudWatch — Amazon CloudWatch 監控 Amazon S3 資源和利用率。
- AWS Lambda — AWS Lambda 是一種無伺服器運算服務，可支援執行程式碼而無需佈建或管理伺服器、建立工作負載感知叢集擴展邏輯、維護事件整合或管理執行階段。在此模式中，Lambda 會在檔案上傳到 Amazon S3 時執行資料庫函數。
- AWS Secrets Manager — AWS Secrets Manager 是一種用於登入資料儲存和擷取的服務。使用 Secrets Manager，您可以透過 API 呼叫 Secret Secrets Manager 來取代程式碼中的硬式編碼認證 (包括密碼)，以程式設計方式擷取密碼。
- Amazon S3 — 亞馬遜簡單儲存服務 (Amazon S3) 提供了一個儲存層，用於接收和存放檔案，以便在與 Aurora PostgreSQL 相容的叢集之間進行取用和傳輸。
- aws_s3 — 此`aws_s3`擴充功能整合了與 Amazon S3 和 Aurora 相容的功能。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和用戶端之間訊息的交付或傳送。在這種模式中，Amazon SNS 用於發送通知。

Code

無論何時將檔案放置在 S3 儲存貯體中，都必須從處理應用程式或 Lambda 函數建立和呼叫資料庫函數。有關詳細信息，請參閱代碼（附件）。

史诗

建立外部檔案

任務	描述	所需技能
將外部檔案新增至來源資料庫。	創建一個外部文件，並將其移動到`oracle`目錄。	DBA

設定目標 (Aurora 與 PostgreSQL 相容)

任務	描述	所需技能
建立 Aurora 資料庫。	在與 Amazon Aurora PostgreSQL 相容的叢集中建立資料庫執行個體。	DBA
建立結構定義、aws_s3 延伸模組和資料表。	使用 [其他資訊] 區段`ext_tbl_scripts` 中的下方的程式碼。這些資料表包括實際資料表、臨時資料表、錯誤和記錄資料表，以及中繼資料表。	DBA, 開發人員
創建數據庫函數。	若要建立 DB 函數，請使用 [其他資訊] 區段中的`load_external_table_latest` 函式下的程式碼。	DBA, 開發人員

建立和設定 Lambda 函數

任務	描述	所需技能
建立角色。	建立具有存取 Amazon S3 和 Amazon Relational Database Service 服務 (Amazon RDS)	DBA

任務	描述	所需技能
	權限的角色。此角色將被指派給 Lambda 以執行模式。	
建立 Lambda 函數。	建立 Lambda 函數，該函數會從 Amazon S3 讀取檔案名稱 (例如，`file_key = info.get('object', {}).get('key')`)，並以檔案名稱做為輸入參數呼叫資料庫函數 (例如`curs.callproc("load_external_tables", [file_key])`)。 根據函數呼叫結果而定，將啟動 SNS 通知 (例如`client.publish(TopicArn='arn:',Message='fileloadsuccess',Subject='fileloadsuccess')`)。 根據您的業務需求，您可以視需要建立含有額外程式碼的 Lambda 函數。如需詳細資訊，請參閱 L ambda 文件。	DBA
設定 S3 儲存貯體事件觸發器。	設定機制，針對 S3 儲存貯體中的所有物件建立事件呼叫 Lambda 函數。	DBA
創建一個密碼。	使用秘密管理員為資料庫認證建立密碼名稱。在 Lambda 函數中傳遞秘密。	DBA

任務	描述	所需技能
上傳支 Lambda 檔案。	上傳包含 Lambda 支援套件和附加的 Python 指令碼的 .zip 檔案，以連線至 Aurora PostgreSQL 相容。Python 程式碼會呼叫您在資料庫中建立的函數。	DBA
建立 SNS 主題。	建立 SNS 主題以傳送資料載入成功或失敗的郵件。	DBA

添加與 Amazon S3 的集成

任務	描述	所需技能
建立 S3 儲存貯體。	在 Amazon S3 主控台上，使用不包含前導斜線的唯一名稱建立 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。	DBA
建立 IAM 政策。	若要建立 AWS Identity and Access Management (IAM) 政策，請使用其他資訊一節`s3bucketpolicy_for_import` 中下方的程式碼。	DBA
建立角色。	為 Aurora PostgreSQL 相容建立兩個角色，一個用於匯入角色，另一個角色用於匯出。將對應的策略指派給角色。	DBA
將角色附加到與 Aurora PostgreSQL 相容的叢集。	在管理角色下，將匯入和匯出角色附加至 Aurora PostgreSQL 叢集。	DBA

任務	描述	所需技能
為 Aurora 相容建立支援物件。	對於表格指令碼，請使用「其他資訊」一節`ext_tbl_scripts` 中下的程式碼。 對於自訂函數，請使用 [其他資訊] 區段`load_external_Table_latest` 中的程式碼。	DBA

處理測試檔

任務	描述	所需技能
將檔案上傳到 S3 儲存貯體。	若要將測試檔案上傳到 S3 儲存貯體，請使用主控台或 AWS CLI 中的下列命令。 `aws s3 cp /Users/Desktop/ukpost/exttbl/"testing files"/aps s3://s3importtest/inputext/aps` 一旦檔案上傳完畢，值區事件就會啟動 Lambda 函數，該函數會執行與 Aurora PostgreSQL 相容的函數。	DBA
檢查數據以及日誌和錯誤文件。	與 Aurora PostgreSQL 相容的函數會將檔案載入主資料表，並在 S3 儲存貯體中建立`.log`和`.bad`檔案。	DBA
監控解決方案。	在 Amazon 主 CloudWatch 控台中，監控 Lambda 函數。	DBA

相關資源

- Amazon S3 整合
- Amazon Simple Storage Service (Amazon S3)
- 使用 Amazon Aurora PostgreSQL 兼容版
- AWS Lambda
- Amazon CloudWatch
- AWS Secrets Manager
- 設定 Amazon SNS 通知

其他資訊

表格腳本

```
CREATE EXTENSION aws_s3 CASCADE;
CREATE TABLE IF NOT EXISTS meta_EXTERNAL_TABLE
(
    table_name_stg character varying(100) ,
    table_name character varying(100)  ,
    col_list character varying(1000)  ,
    data_type character varying(100)  ,
    col_order numeric,
    start_pos numeric,
    end_pos numeric,
    no_position character varying(100)  ,
    date_mask character varying(100)  ,
    delimeter character(1)  ,
    directory character varying(100)  ,
    file_name character varying(100)  ,
    header_exist character varying(5)
);
CREATE TABLE IF NOT EXISTS ext_tbl_stg
(
    col1 text
);
CREATE TABLE IF NOT EXISTS error_table
(
    error_details text,
    file_name character varying(100),
    processed_time timestamp without time zone
```

```
);
CREATE TABLE IF NOT EXISTS log_table
(
    file_name character varying(50) COLLATE pg_catalog."default",
    processed_date timestamp without time zone,
    tot_rec_count numeric,
    proc_rec_count numeric,
    error_rec_count numeric
);
sample insert scripts of meta data:
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'source_filename', 'character varying', 2, 8, 27, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'record_type_identifier', 'character varying', 3, 28, 30, NULL, NULL, NULL,
 'databasedev', 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'fad_code', 'numeric', 4, 31, 36, NULL, NULL, NULL, 'databasedev', 'externalinterface/
loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'session_sequence_number', 'numeric', 5, 37, 42, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
INSERT INTO meta_EXTERNAL_TABLE (table_name_stg, table_name, col_list, data_type,
 col_order, start_pos, end_pos, no_position, date_mask, delimeter, directory,
 file_name, header_exist) VALUES ('F_EX_APS_TRANSACTIONS_STG', 'F_EX_APS_TRANSACTIONS',
 'transaction_sequence_number', 'numeric', 6, 43, 48, NULL, NULL, NULL, 'databasedev',
 'externalinterface/loaddir/APS', 'NO');
```

s3 儲存格政策 _ 用於匯入

```
---Import role policy
--Create an IAM policy to allow, Get,  and list actions on S3 bucket
 {
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "s3import",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::s3importtest",
                "arn:aws:s3:::s3importtest/*"
            ]
        }
    ]
}
--Export Role policy
--Create an IAM policy to allow, put,  and list actions on S3 bucket
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3export",
            "Action": [
                "S3:PutObject",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::s3importtest/*"
            ]
        }
    ]
}
```

示例數據庫函數加載外部表最新

```
CREATE OR REPLACE FUNCTION public.load_external_tables(pi_filename text)
 RETURNS character varying
 LANGUAGE plpgsql
AS $function$
/* Loading data from S3 bucket into a APG table */
DECLARE
 v_final_sql TEXT;
 pi_ext_table TEXT;
```

```
r refCURSOR;
v_sqlerrm text;
v_chunk numeric;
i integer;
v_col_list TEXT;
v_postion_list CHARACTER VARYING(1000);
v_len  integer;
v_delim varchar;
v_file_name CHARACTER VARYING(1000);
v_directory CHARACTER VARYING(1000);
v_table_name_stg CHARACTER VARYING(1000);
v_sql_col TEXT;
v_sql TEXT;
v_sql1 TEXT;
v_sql2 TEXT;
v_sql3 TEXT;
v_cnt integer;
v_sql_dynamic TEXT;
v_sql_ins TEXT;
proc_rec_COUNT integer;
error_rec_COUNT integer;
tot_rec_COUNT integer;
v_rec_val integer;
rec record;
v_col_cnt integer;
kv record;
v_val text;
v_header text;
j integer;
ERCODE VARCHAR(5);
v_region text;
cr CURSOR FOR
SELECT distinct DELIMETER,
  FILE_NAME,
  DIRECTORY
FROM  meta_EXTERNAL_TABLE
WHERE table_name = pi_ext_table
  AND DELIMETER IS NOT NULL;


cr1 CURSOR FOR
  SELECT   col_list,
  data_type,
  start_pos,
```

```
   END_pos,
   concat_ws('',' ',TABLE_NAME_STG) as TABLE_NAME_STG,
   no_position,date_mask
 FROM  meta_EXTERNAL_TABLE
 WHERE table_name = pi_ext_table
 order by col_order asc;
cr2 cursor FOR
SELECT  distinct table_name,table_name_stg
   FROM  meta_EXTERNAL_TABLE
   WHERE upper(file_name) = upper(pi_filename);


BEGIN
 -- PERFORM utl_file_utility.init();
   v_region := 'us-east-1';
   /* find tab details from file name */


   --DELETE FROM  ERROR_TABLE WHERE file_name= pi_filename;
  -- DELETE FROM  log_table WHERE file_name= pi_filename;


 BEGIN


   SELECT distinct table_name,table_name_stg INTO strict pi_ext_table,v_table_name_stg
   FROM  meta_EXTERNAL_TABLE
   WHERE upper(file_name) = upper(pi_filename);
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
    raise notice 'error 1,%',sqlerrm;
    pi_ext_table := null;
    v_table_name_stg := null;
      RAISE USING errcode = 'NTFIP' ;
    when others then
        raise notice 'error others,%',sqlerrm;
 END;
 j :=1 ;

for rec in  cr2
 LOOP
```

```
 pi_ext_table     := rec.table_name;
 v_table_name_stg := rec.table_name_stg;
 v_col_list := null;


IF pi_ext_table IS NOT NULL
 THEN
   --EXECUTE concat_ws('','truncate table  ' ,pi_ext_table) ;
  EXECUTE concat_ws('','truncate table  ' ,v_table_name_stg) ;




      SELECT distinct DELIMETER INTO STRICT v_delim
      FROM  meta_EXTERNAL_TABLE
      WHERE table_name = pi_ext_table;


      IF v_delim IS NOT NULL THEN
    SELECT distinct DELIMETER,
      FILE_NAME,
      DIRECTORY ,
      concat_ws('',' ',table_name_stg),
      case  header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
    INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
    FROM  meta_EXTERNAL_TABLE
    WHERE table_name = pi_ext_table
      AND DELIMETER IS NOT NULL;


    IF    upper(v_delim) = 'CSV'
    THEN
      v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3 ( ''',
      v_table_name_stg,''','''',
      ''DELIMITER '''','''' CSV HEADER QUOTE '''''''''', aws_commons.create_s3_uri
( ''',
      v_directory,''','',v_file_name,''', ''',v_region,'''))');
      ELSE
      v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(''',
          v_table_name_stg, ''','''', ''DELIMITER AS ''''^''''',''',','
         aws_commons.create_s3_uri
          ( ''',v_directory, ''','',
          v_file_name, ''',',
```

```
            ''',v_region,''')
         )');
         raise notice 'v_sql , %',v_sql;
      begin
       EXECUTE  v_sql;
      EXCEPTION
        WHEN OTHERS THEN
          raise notice 'error 1';
        RAISE USING errcode = 'S3IMP' ;
      END;


      select count(col_list) INTO v_col_cnt
      from  meta_EXTERNAL_TABLE where table_name = pi_ext_table;







       -- raise notice 'v_sql 2, %',concat_ws('','update ',v_table_name_stg, ' set
col1 = col1||''',v_delim,''''');


      execute concat_ws('','update ',v_table_name_stg, ' set col1 =
col1||''',v_delim,''''');




      i :=1;
      FOR rec in cr1
      loop
      v_sql1 := concat_ws('',v_sql1,'split_part(col1,''',v_delim,''',', i,')',' as
',rec.col_list,',');
      v_sql2 := concat_ws('',v_sql2,rec.col_list,',');
  --    v_sql3 := concat_ws('',v_sql3,'rec.',rec.col_list,'::',rec.data_type,',');


      case
        WHEN upper(rec.data_type) = 'NUMERIC'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
```

```
                ELSE
                coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))):NUMERIC,0)::',rec.data_type,' END as ',rec.col_list,',') ;
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''99990101''),''YYYYMMDD''):',rec.data_type,' END as ',rec.col_list,',');
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
 'MM/DD/YYYY hh24:mi:ss'
        THEN v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''01/01/9999 0024:00:00''),''MM/DD/YYYY hh24:mi:ss''):',rec.data_type,' END as
',rec.col_list,',');
         ELSE
       v_sql3 := concat_ws('',v_sql3,' case WHEN
length(trim(split_part(col1,''',v_delim,''',', i,'))) =0
               THEN null
                ELSE
                 coalesce((trim(split_part(col1,''',v_delim,''',',
i,'))),''''):',rec.data_type,' END as ',rec.col_list,',') ;
      END case;


      i :=i+1;
      end loop;


        -- raise notice 'v_sql 3, %',v_sql3;


      SELECT trim(trailing ' ' FROM v_sql1) INTO v_sql1;
      SELECT trim(trailing ',' FROM v_sql1) INTO v_sql1;


      SELECT trim(trailing ' ' FROM v_sql2) INTO v_sql2;
      SELECT trim(trailing ',' FROM v_sql2) INTO v_sql2;
```

```
       SELECT trim(trailing ' ' FROM v_sql3) INTO v_sql3;
       SELECT trim(trailing ',' FROM v_sql3) INTO v_sql3;


       END IF;
      raise notice 'v_delim , %',v_delim;


     EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg)  INTO v_cnt;


    raise notice 'stg cnt , %',v_cnt;


    /* if upper(v_delim) = 'CSV' then
       v_sql_ins := concat_ws('', ' SELECT * from ' ,v_table_name_stg );
     else
      -- v_sql_ins := concat_ws('',' SELECT ',v_sql1,'  from (select col1 from
 ' ,v_table_name_stg , ')sub ');
       v_sql_ins := concat_ws('',' SELECT ',v_sql3,'  from (select col1 from
 ' ,v_table_name_stg , ')sub ');
       END IF;*/


v_chunk := v_cnt/100;




for i in 1..101
loop
     BEGIN
    -- raise notice 'v_sql , %',v_sql;
       -- raise notice 'Chunk number , %',i;
       v_sql_ins := concat_ws('',' SELECT ',v_sql3,'  from (select col1 from
 ' ,v_table_name_stg , ' offset ',v_chunk*(i-1), ' limit ',v_chunk,') sub ');


     v_sql := concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins);
     -- raise notice 'select statement , %',v_sql_ins;
          -- v_sql := null;
     -- EXECUTE concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins, 'offset
 ',v_chunk*(i-1), ' limit ',v_chunk );
```

```
    --v_sql := concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins );


    -- raise notice 'insert statement , %',v_sql;


   raise NOTICE 'CHUNK START %',v_chunk*(i-1);
  raise NOTICE 'CHUNK END %',v_chunk;


    EXECUTE v_sql;


 EXCEPTION
      WHEN OTHERS THEN
      -- v_sql_ins := concat_ws('',' SELECT ',v_sql1, '  from (select col1 from
' ,v_table_name_stg , ' )sub ');
        -- raise notice 'Chunk number for cursor , %',i;


   raise NOTICE 'Cursor - CHUNK START %',v_chunk*(i-1);
  raise NOTICE 'Cursor -  CHUNK END %',v_chunk;
        v_sql_ins := concat_ws('',' SELECT ',v_sql3, '  from (select col1 from
' ,v_table_name_stg , ' )sub ');


        v_final_sql := REPLACE (v_sql_ins, ''''::text, ''''''::text);
       -- raise notice 'v_final_sql %',v_final_sql;
        v_sql :=concat_ws('','do $a$ declare  r refcursor;v_sql text; i
numeric;v_conname text;  v_typ  ',pi_ext_table,'[]; v_rec  ','record',';
          begin







          open r for execute ''select col1 from ',v_table_name_stg ,'  offset
',v_chunk*(i-1), ' limit ',v_chunk,''';
          loop
          begin
          fetch r into v_rec;
          EXIT WHEN NOT FOUND;
```

```
          v_sql := concat_ws('''','insert into  ',pi_ext_table,' SELECT ',REPLACE
 (v_sql3, ''''::text, ''''''::text) , '  from ( select ''''''',v_rec.col1,''''''' as
 col1) v'');
           execute v_sql;


          exception
           when others then
         v_sql := ''INSERT INTO  ERROR_TABLE VALUES (concat_ws(''''''''','''''Error
 Name: ''''',$$''||SQLERRM||''$$,'''''Error State: ''''','''''''||
SQLSTATE||''''''','''''record : ''''',$$''||v_rec.col1||''$$),'''''||
pi_filename||''''',now())'';


             execute v_sql;
            continue;
          end ;
          end loop;
          close r;
          exception
          when others then
        raise;
          end ; $a$');
     -- raise notice ' inside excp v_sql %',v_sql;
         execute v_sql;
     --  raise notice 'v_sql %',v_sql;
      END;
  END LOOP;
     ELSE


     SELECT distinct DELIMETER,FILE_NAME,DIRECTORY ,concat_ws('',' ',table_name_stg),
       case  header_exist when 'YES' then 'CSV HEADER' else 'CSV' end as header_exist
       INTO STRICT v_delim,v_file_name,v_directory,v_table_name_stg,v_header
     FROM  meta_EXTERNAL_TABLE
     WHERE table_name = pi_ext_table                  ;
     v_sql := concat_ws('','SELECT aws_s3.table_import_FROM_s3(''',
       v_table_name_stg, ''','''', ''DELIMITER AS ''''#'''' ',v_header,' '','','
      aws_commons.create_s3_uri
       ( ''',v_directory, ''','''',
       v_file_name, ''',',
```

```
        ''''',v_region,''')
      )');
         EXECUTE  v_sql;


    FOR rec in cr1
    LOOP


     IF rec.start_pos IS NULL AND rec.END_pos IS NULL AND rec.no_position = 'recnum'
     THEN
       v_rec_val := 1;
     ELSE


      case
        WHEN upper(rec.data_type) = 'NUMERIC'
        THEN v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1)))::NUMERIC,0)::',rec.data_type,' END as
',rec.col_list,',') ;
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDD'
        THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''99990101''),''YYYYMMDD''):',rec.data_type,'
END as ',rec.col_list,',');
        WHEN UPPER(rec.data_type) = 'TIMESTAMP WITHOUT TIME ZONE' AND rec.date_mask =
'YYYYMMDDHH24MISS'
        THEN v_sql1 := concat_ws('','case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                to_date(coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''9999010100240000''),''YYYYMMDDHH24MISS''):',rec.data_
END as ',rec.col_list,',');
         ELSE
```

```
       v_sql1 := concat_ws('',' case WHEN length(trim(substring(COL1,
',rec.start_pos ,',', rec.END_pos,'-',rec.start_pos ,'+1))) =0
               THEN null
                ELSE
                 coalesce((trim(substring(COL1, ',rec.start_pos ,',',
rec.END_pos,'-',rec.start_pos ,'+1))),''''):',rec.data_type,' END as
',rec.col_list,',') ;
      END case;


     END IF;
     v_col_list := concat_ws('',v_col_list ,v_sql1);
    END LOOP;




          SELECT trim(trailing ' ' FROM v_col_list) INTO v_col_list;
          SELECT trim(trailing ',' FROM v_col_list) INTO v_col_list;


          v_sql_col   :=  concat_ws('',trim(trailing ',' FROM v_col_list) , ' FROM
 ',v_table_name_stg,' WHERE col1 IS NOT NULL AND length(col1)>0 ');




          v_sql_dynamic := v_sql_col;


          EXECUTE  concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg) INTO v_cnt;




        IF v_rec_val = 1 THEN
            v_sql_ins := concat_ws('',' select row_number() over(order by ctid) as
line_number ,' ,v_sql_dynamic) ;


        ELSE
              v_sql_ins := concat_ws('',' SELECT' ,v_sql_dynamic) ;
          END IF;
```

```
    BEGIN
      EXECUTE concat_ws('','insert into  ', pi_ext_table ,' ', v_sql_ins);
          EXCEPTION
             WHEN OTHERS THEN
         IF v_rec_val = 1 THEN
                v_final_sql := ' select row_number() over(order by ctid) as
line_number ,col1 from ';
               ELSE
                v_final_sql := ' SELECT col1 from';
              END IF;
      v_sql :=concat_ws('','do $a$ declare  r refcursor;v_rec_val numeric :=
',coalesce(v_rec_val,0),';line_number numeric; col1 text; v_typ  ',pi_ext_table,'[];
v_rec  ',pi_ext_table,';
            begin
            open r for execute ''',v_final_sql, ' ',v_table_name_stg,' WHERE col1 IS
NOT NULL AND length(col1)>0 '' ;
            loop
            begin
            if   v_rec_val = 1 then
            fetch r into line_number,col1;
            else
            fetch r into col1;
            end if;


            EXIT WHEN NOT FOUND;
             if v_rec_val = 1 then
             select line_number,',trim(trailing ',' FROM v_col_list) ,' into v_rec;
             else
               select ',trim(trailing ',' FROM v_col_list) ,' into v_rec;
             end if;


            insert into  ',pi_ext_table,' select v_rec.*;
             exception
             when others then
              INSERT INTO  ERROR_TABLE VALUES (concat_ws('''',''Error Name:
'',SQLERRM,''Error State: '',SQLSTATE,''record : '',v_rec),''',pi_filename,''',now());
              continue;
             end ;
              end loop;
            close r;
             exception
```

```
            when others then
            raise;
            end ; $a$');
        execute v_sql;


    END;


        END IF;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM  ' ,pi_ext_table)   INTO proc_rec_COUNT;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM  error_table WHERE file_name
='''',pi_filename,''' and processed_time::date = clock_timestamp()::date')  INTO
error_rec_COUNT;


  EXECUTE concat_ws('','SELECT COUNT(*) FROM ',v_table_name_stg)   INTO tot_rec_COUNT;


  INSERT INTO  log_table values(pi_filename,now(),tot_rec_COUNT,proc_rec_COUNT,
error_rec_COUNT);


  raise notice 'v_directory, %',v_directory;


  raise notice 'pi_filename, %',pi_filename;


  raise notice 'v_region, %',v_region;


 perform aws_s3.query_export_to_s3('SELECT
replace(trim(substring(error_details,position(''('' in
error_details)+1),'')''),'','','';''),file_name,processed_time FROM  error_table WHERE
file_name = '''||pi_filename||'''',
  aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
  options :='FORmat csv, header, delimiter $$,$$'
  );
```

```
raise notice 'v_directory, %',v_directory;


   raise notice 'pi_filename, %',pi_filename;


   raise notice 'v_region, %',v_region;


  perform aws_s3.query_export_to_s3('SELECT * FROM  log_table WHERE file_name = '''||
pi_filename||'''',
   aws_commons.create_s3_uri(v_directory, pi_filename||'.log', v_region),
   options :='FORmat csv, header, delimiter $$,$$'
   );




   END IF;
 j := j+1;
 END LOOP;


       RETURN 'OK';
EXCEPTION
    WHEN  OTHERS THEN
  raise notice 'error %',sqlerrm;
   ERCODE=SQLSTATE;
   IF ERCODE = 'NTFIP' THEN
     v_sqlerrm := concat_Ws('',sqlerrm,'No data for the filename');
   ELSIF ERCODE = 'S3IMP' THEN
    v_sqlerrm := concat_Ws('',sqlerrm,'Error While exporting the file from S3');
   ELSE
      v_sqlerrm := sqlerrm;
   END IF;


 select distinct directory into v_directory from  meta_EXTERNAL_TABLE;




 raise notice 'exc v_directory, %',v_directory;
```

```
   raise notice 'exc pi_filename, %',pi_filename;


   raise notice 'exc v_region, %',v_region;


  perform aws_s3.query_export_to_s3('SELECT * FROM  error_table WHERE file_name = '''||
pi_filename||'''',
   aws_commons.create_s3_uri(v_directory, pi_filename||'.bad', v_region),
   options :='FORmat csv, header, delimiter $$,$$'
   );
    RETURN null;
END;
$function$
```

將基於函數的索引從甲骨文遷移到 PostgreSQL

由韋蘭賈納魯格蘭希（AWS）和納瓦卡坎塔魯（AWS）創建

環境：生產	來源：甲骨文	目標：PostgreSQL
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫

Summary

索引是提高數據庫性能的常見方法。索引允許數據庫服務器比沒有索引更快地查找和檢索特定行。但是索引也會增加整個數據庫系統的開銷，因此應該合理地使用它們。以函數為基礎的索引 (以函數或運算式為基礎) 可能涉及多個資料行和數學運算式。以函數為基礎的索引可改善使用索引運算式之查詢的效能。

從本身上講，PostgreSQL 不支持使用將波動性定義為穩定的函數創建基於函數的索引。但是，您可以創建具有波動性的類似函數，`IMMUTABLE`並在索引創建中使用它們。

`IMMUTABLE`函數不能修改數據庫，並且保證在給定相同參數的情況下永遠返回相同的結果。此類別可讓最佳化工具在查詢使用常數引數呼叫函式時預先評估函式。

這種模式有助於遷移基於甲骨文函數的索引時，功能`to_char`，如`to_date`，和 `to_number` PostgreSQL 等效。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶
- 設定並執行監聽器服務的來源 Oracle 資料庫執行處理
- 熟悉 PostgreSQL 庫

限制

- 資料庫大小限制為 64 TB。
- 在索引創建中使用的函數必須是不可變的。

產品版本

- 版本 11g (11.2.0.3.v1 及更新版本) 以及最高至 12.2 和 18c 的所有甲骨文資料庫版本
- PostgreSQL 及更高版本

架構

源, 技術, 堆棧

- 內部部署或亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上的 Oracle 資料庫，或適用於 Oracle 資料庫執行個體的 Amazon RDS

目標技術堆疊

- 任 PostgreSQL 擎

工具

- pgAdmin 4 是一個開源管理工具，適用於 Postgres。pgAdmin 4 工具提供了用於創建、維護和使用數據庫對象的圖形界面。
- Oracle SQL 開發人員是一個整合式開發環境 (IDE)，用於在傳統和雲端部署中開發和管理 Oracle 資料庫。

史诗

使用預設函數建立以函數為基礎的索引

任務	描述	所需技能
使用 to_char 函數在資料行上建立以函數為基礎的索引。	使用下面的代碼來創建基於函數的索引。 `postgres=# create table funcindex( col1 timestamp without time zone);` `CREATE TABLE`	DBA, 應用程式開發人員

任務	描述	所需技能
	`postgres=# insert into funcindex values (now()); INSERT 0 1 postgres=# select * from funcindex; col1 ---------- 2022-08-09 16:00:57.77414 (1 rows) postgres=# create index funcindex_idx on funcindex(to_char(col1,'DD-MM-YYYY HH24:MI:SS')); ERROR: functions in index expression must be marked IMMUTABLE` 注意：PostgreSQL 不允許在沒有子句的情況下創建基於函數的索引。IMMUTABLE	
檢查功能的波動性。	要檢查功能波動性，請使用其他信息部分中的代碼。	DBA

使用包裝函數創建基於函數的索引

任務	描述	所需技能
創建一個包裝函數。	若要建立包裝函式，請使用 [其他資訊] 區段中的程式碼。	PostgreSQL 员

任務	描述	所需技能
使用包裝函數建立索引。	使用 [其他資訊] 區段中的程式碼，建立使用者定義函數，其`IMMUTABLE` 中的關鍵字與應用程式位於相同的結構描述中，並在索引建立指令碼中參照該函數。 如果使用者定義函數是在通用結構描述中建立的 (從上一個範例中)，請`search_path` 如下所示更新。 `ALTER ROLE <ROLENAME> set search_path=$user, COMMON;`	開 PostgreSQL 員

驗證索引建立

任務	描述	所需技能
驗證索引建立。	根據查詢存取模式，驗證是否需要建立索引。	DBA
驗證是否可以使用索引。	若要檢查 PostgreSQL 最佳化程式是否拾取以函數為基礎的索引，請使用「解釋」或「解釋分析」來執行 SQL 陳述式。使用 [其他資訊] 區段中的程式碼。如果可能的話，也收集表格統計資料。 注意：如果您注意到說明計劃，PostgreSQL 優化器因為謂詞條件而選擇了基於函數的索引。	DBA

相關資源

- 以函數為基礎的索引 (Oracle 文件集)
- 運算式上的索 PostgreSQL 件集)
- PostgreSQL 性 PostgreSQL)
- PostgreSQL 路徑 (Postgre SQL)
- 甲骨 PostgreSQL 資料庫 19c 到 Amazon Aurora

其他資訊

創建一個包裝函數

```
CREATE OR REPLACE FUNCTION myschema.to_char(var1 timestamp without time zone, var2
 varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
 sql IMMUTABLE;
```

使用包裝函數創建索引

```
postgres=# create function common.to_char(var1 timestamp without time zone, var2
 varchar) RETURNS varchar AS $BODY$ select to_char(var1, 'YYYYMMDD'); $BODY$ LANGUAGE
 sql IMMUTABLE;
CREATE FUNCTION
postgres=# create index funcindex_idx on funcindex(common.to_char(col1,'DD-MM-YYYY
 HH24:MI:SS'));
CREATE INDEX
```

檢查功能的波動性

```
SELECT DISTINCT p.proname as "Name",p.provolatile as "volatility" FROM
 pg_catalog.pg_proc p
 LEFT JOIN pg_catalog.pg_namespace n ON n.oid = p.pronamespace
 LEFT JOIN pg_catalog.pg_language l ON l.oid = p.prolang
 WHERE n.nspname OPERATOR(pg_catalog.~) '^(pg_catalog)$' COLLATE pg_catalog.default AND
 p.proname='to_char'GROUP BY p.proname,p.provolatile
ORDER BY 1;
```

驗證索引是否可以使用

```
explain analyze <SQL>
```

```
postgres=# explain select col1 from funcindex where common.to_char(col1,'DD-MM-YYYY
 HH24:MI:SS') = '09-08-2022 16:00:57';
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------
 Index Scan using funcindex_idx on funcindex  (cost=0.42..8.44 rows=1 width=8)
   Index Cond: ((common.to_char(col1, 'DD-MM-YYYY HH24:MI:SS'::character
 varying))::text = '09-08-2022 16:00:57'::text)
(2 rows)
```

使用擴充功能將甲骨文原生函數遷移至 PostgreSQL

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS
R 型：重新建築	工作負載：甲骨文; 開源	技術：移轉；資料庫
AWS 服務：Amazon EC2; Amazon RDS		

Summary

此遷移模式提供 step-by-step 指導，指引將適用於甲骨文資料庫執行個體的 Amazon Relational Database Service 服務 (Amazon RDS) 移轉到 Amazon RDS 版或 Amazon Aurora PostgreSQL 相容版本資料庫，方法是將`aws_oracle_ext`和`orafce`擴充修改為 PostgreSQL () 原生內建程式碼。`psql`這樣可以節省處理時間。

此模式說明具有大量交易的多 TB Oracle 來源資料庫不會停機的離線手動移轉策略。

遷移程序使用 AWS Schema Conversion Tool (AWS SCT) 搭配`aws_oracle_ext`和`orafce`擴充功能，將 Amazon RDS for Oracle 文資料庫架構的資料庫結構描述轉換為 Amazon RDS 或與 Aurora PostgreSQL 相容的資料庫結構描述。然後將代碼手動更改為 PostgreSQL 支持的本機`psql`內置代碼。這是因為擴充功能呼叫會影響 PostgreSQL 資料庫伺服器上的程式碼處理，而且並非所有的擴充程式碼都完全投訴或與 PostgreSQL 程式碼相容。

此模式主要著重於使用 AWS SCT 和擴充功能`aws_oracle_ext`和`orafce`. 您可以將已使用的擴充功能轉換為原生 PostgreSQL (`psql`) 內建插件。然後，您刪除對擴展名的所有引用並相應地轉換代碼。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 作業系統 (視窗或 Mac) 或亞馬遜 EC2 執行個體 (啟動並執行)
- 奧拉夫

限制

並非所有使用aws_oracle_ext或orafce擴展的甲骨文函數都可以轉換為本地 PostgreSQL 函數。它可能需要手動返工，以便使用 PostgreSQL 庫進行編譯。

使用 AWS SCT 擴充功能的一個缺點是它在執行和擷取結果時效能降低。它的成本可以從簡單的 PostgreSQL 解釋計劃（語句的執行計劃）中了解甲骨文SYSDATE函數遷移到所有三個代碼（aws_oracle_ext，和psql默認）之間的 PostgreSQL NOW() 函數，如附加文檔中的性能比較檢查部分中所述。orafce

產品版本

- 資料來源：Amazon RDS for Oracle 文資料庫 10.2 及更新版本 (適用於 10.x)、11 公克 (11.2.0.3.v1 及更高版本) 及最高 12.2、18 c 和 19c (及更高版本)，適用於企業版、標準版 1 和標準版 2
- 目標：Amazon RDS for PostgreSQL 或 Aurora 與 PostgreSQL 相容的資料庫 9.4 及更新版本 (適用於 9.x)、10.x、11.x、12 倍、13.x 和 14.x (及更新版本)
- AWS SCT：最新版本 (此模式已使用 1.0.632 進行測試)
- 歐拉菲：最新版本（這種模式與 3.9.0 測試）

架構

源, 技術, 堆棧

- 一個適用於甲骨文資料庫執行個體的亞馬遜 RDS 版本

目標技術堆疊

- 具有 11.5 版 Amazon RDS for PostgreSQL 或 Aurora 兼容資料庫執行個體

資料庫遷移架構

下圖表示源甲骨文和目標 PostgreSQL 數據庫之間的數據庫遷移架構。此架構包括 AWS 雲端、虛擬私有雲端 (VPC)、可用區域、私有子網路、Oracle 資料庫的 Amazon RDS、AWS SCT、Amazon RDS 與 PostgreSQL 相容或 Aurora PostgreSQL 相容的資料庫、甲骨文 (aws_oracle_ext和) 的擴充功能，以及結構化查詢語言 (SQLorafce) 檔案。

1. 啟動適用於 Oracle 資料庫執行個體 (來源資料庫) 的 Amazon RDS。

2. 將 AWS SCT 與aws_oracle_ext和orafce擴充套件搭配使用，將原始程式碼從甲骨文轉換為 PostreSQL。
3. 此轉換會產生支援 PostgreSQL 的移轉檔案。
4. 手動將未轉換的甲骨文擴展代碼轉換為 PostgreSQL () psql 代碼。
5. 手動轉換會產生支援 PostgreSQL 的轉換後的 .sql 檔案。
6. 在您的 Amazon RDS 資料庫執行個體 (目標資料庫) 上執行這些 .sql 檔案。

工具

工具

AWS 服務

- AWS SCT-AWS Schema Conversion Tool (AWS SCT) 會將您現有的資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎。您可以轉換關聯式線上交易處理 (OLTP) 結構描述或資料倉儲結構描述。轉換後的結構描述適用於 Amazon RDS for MySQL 資料庫執行個體、Amazon Aurora 資料庫叢集、Amazon RDS for PostgreSQL SQL 資料庫執行個體或 Amazon Redshift 叢集。轉換後的結構描述也可以與 Amazon EC2 執行個體上的資料庫搭配使用，或以資料形式存放在 Amazon S3 儲存貯體中。

  AWS SCT 提供以專案為基礎的使用者界面，可自動將來源資料庫的資料庫結構描述轉換為與目標 Amazon RDS 執行個體相容的格式。

  您可以使用 AWS SCT 從 Oracle 來源資料庫遷移到前面列出的任何目標。您可以使用 AWS SCT 匯出來源資料庫物件定義，例如結構描述、檢視、預存程序和函數。

  您可以使用 AWS SCT 將資料從甲骨文轉換為 Amazon RDS for PostgreSQL) 或 Amazon Aurora PostgreSQL 相容版本。

  在這種模式中，您可以使用 AWS SCT 使用擴充功能和將 Oracle 程式碼轉換並遷移到 PostgreSQLorafce，aws_oracle_ext然後手動將擴充功能代碼遷移到psql預設或原生內建程式碼。
- AWS SCT 擴充套件是一種附加模組，可模擬將物件轉換為目標資料庫時所需的來源資料庫中存在的函數。您必須先轉換資料庫結構描述，才能安裝 AWS SCT 延伸套件。

  轉換資料庫或資料倉儲結構描述時，AWS SCT 會將額外的結構描述新增至目標資料庫。此結構描述會實作來源資料庫的 SQL 系統功能，當您將已轉換的結構描述寫入至目標資料庫時需要這些功能。這個額外的結構描述稱為延伸套件結構描述。

OLTP 資料庫的擴充套件結構描述會根據來源資料庫來命名。對於 Oracle 資料庫，擴充套件結構描述為AWS_ORACLE_EXT。

其他工具

- Orafce-Orafce 是實現甲骨文兼容功能，數據類型和包的模塊。這是一個具有伯克利源分發（BSD）許可證的開源工具，因此任何人都可以使用它。該orafce模塊對於從甲骨文遷移到 PostgreSQL 非常有用，因為它在 PostgreSQL 中實現了許多甲骨文函數。

Code

有關從 Oracle 到 PostgreSQL 的所有常用和遷移代碼的清單，以避免使用 AWS SCT 擴展程式碼，請參閱隨附的文件。

史诗

設定 Amazon RDS for Oracle 來源資料庫

任務	描述	所需技能
建立 Oracle 資料庫執行處理。	從 Amazon RDS 主控台建立適用於甲骨文或 Aurora PostgreSQL 相容的資料庫執行個體。	一般 AWS、DBA
設定安全性群組。	設定輸入和輸出安全性群組。	一般 AWS
建立資料庫。	建立包含所需使用者和綱要的 Oracle 資料庫。	一般 AWS、DBA
創建對象。	創建對象並在模式中插入數據。	DBA

設定 Amazon RDS for PostgreSQL 的目標資料庫

任務	描述	所需技能
建立資 PostgreSQL 行個體。	從 Amazon RDS 主控台建立適用於 PostgreSQL 的亞馬遜 RDS 或 Amazon Aurora 資料庫執行個體。	一般 AWS、DBA
設定安全性群組。	設定輸入和輸出安全性群組。	一般 AWS
建立資料庫。	使用所需的使用者和結構描述建立 PostgreSQL 資料庫。	一般 AWS、DBA
驗證擴充功能。	請確定 PostgreSQL 資料庫中已正確安裝和設定和設定。`aws_oracle_extorafce`	DBA
請確認 PostgreSQL 資料庫可供使用。	請確定 PostgreSQL 資料庫已啟動並執行中。	DBA

使用 AWS SCT 和擴充功能將甲骨文架構遷移到 PostgreSQL

任務	描述	所需技能
安裝 AWS SCT。	安裝最新版本的 AWS SCT。	DBA
設定 AWS SCT。	使用適用於甲骨文 () 和 PostgreSQL () 的 Java 資料庫連線 (`ojdbc8.jar` JDBC) 驅動程式來設定 AWS SCT。`postgresql-42.2.5.jar`	DBA
啟用 AWS SCT 擴充套件或範本。	在 AWS SCT 專案設定下，透過 Oracle 資料庫結構描述的`aws_oracl`	DBA

任務	描述	所需技能
	e_ext 和orafce擴充功能啟用內建函數實作。	
轉換結構描述。	在 AWS SCT 中，選擇「轉換結構描述」，將結構描述從甲骨文轉換為 PostgreSQL，然後產生 .sql 檔案。	DBA

將 AWS SCT 擴充程式碼轉換為 psql 程式碼

任務	描述	所需技能
手動轉換代碼。	手動將支援擴充功能的每一行程式碼轉換為psql預設的內建程式碼，如附件文件所述。例如，ORACLE.SYSDATE() 將AWS_ORACLE_EXT.SYSDATE() 或變更為NOW()。	DBA
驗證代碼	(選擇性) 透過在 PostgreSQL 資料庫中暫時執行程式碼來驗證每一行程式碼。	DBA
在 PostgreSQL 資料庫中建立物件。	若要在 PostgreSQL 資料庫中建立物件，請執行 AWS SCT 產生並在前兩個步驟中修改的 .sql 檔案。	DBA

相關資源

- 資料庫
  - Amazon RDS 上的甲骨文
  - Amazon RDS 上的

  - 與 Amazon Aurora 合作
  - PostgreSQL 计划
- AWS SCT
  - AWS Schema Conversion Tool 概觀
  - 使用者指南
  - 使用 AWS SCT 使用者界面
  - 使用 Oracle 資料庫做為 AWS SCT 的來源
- 適用於 AWS SCT 的擴充功能
  - 使用 AWS SCT 擴充套件
  - 甲骨文功能
  - PGXN 神諭
  - GitHub 機會

其他資訊

如需詳細資訊，請遵循詳細指令 (含語法和範例)，以手動轉換附加文件中的程式碼。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS DMS 將 Db2 資料庫從 Amazon EC2 遷移到與 MySQL 相容的 Aurora

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	資料來源：Amazon EC2 上的 IBM Db2	目標：Amazon Aurora MySQL 兼容版
R 型：重新建築	工作負載：IBM	技術：移轉；資料庫
AWS 服務：AWS DMS；Amazon EC2；AWS SCT；Amazon Aurora		

Summary

將適用於 LUW 的 IBM Db2 資料庫遷移到亞馬遜彈性運算雲端 (Amazon EC2) 之後，請考慮移至亞 Amazon Web Services (AWS) 雲端原生資料庫來重新架構資料庫。此模式涵蓋將在 Amazon EC2 執行個體上執行的 IBM Db2 for LUW 資料庫遷移到 AWS 上與 Amazon Aurora MySQL 相容的版本資料庫。

該模式描述了具有大量交易的多 TB Db2 來源資料庫的停機時間最短的線上移轉策略。

此模式使用 AWS Schema Conversion Tool (AWS SCT) 將 Db2 資料庫結構描述轉換為 Aurora 與 MySQL 相容的結構描述。然後，該模式使用 AWS Database Migration Service (AWS DMS) 將資料從 Db2 資料庫遷移到 Aurora 與 MySQL 相容的資料庫。AWS SCT 未轉換的程式碼將需要手動轉換。

先決條件和限制

先決條件

- 具有虛擬私有雲 (VPC) 的有效 AWS 帳戶
- AWS SCT
- AWS DMS

產品版本

- 最新版本
- 適用於 Linux 版本 11.1.4.4 及更新版本的 Db2

架構

源, 技術, 堆棧

- 安裝在 EC2 執行個體上的資料庫

目標技術堆疊

- Amazon Aurora 與 MySQL 相容的版本資料庫執行個體

來源與目標架構

下圖顯示來源 Db2 和目標 Aurora MySQL 相容資料庫之間的資料移轉架構。AWS 雲端上的架構包括虛擬私有雲 (VPC) (虛擬私有雲)、可用區域、Db2 執行個體的公有子網路和 AWS DMS 複寫執行個體，以及 Aurora MySQL 相容資料庫的私有子網路。

工具

AWS 服務

- Amazon Aurora 是全受管的關聯式資料庫引擎，專為雲端建置，並與 MySQL 和 PostgreSQL 相容。
- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。AWS SCT 支援做為 LUW 版本 9.1、9.5、9.7、10.1、11.1 和 11.5 版的來源 IBM Db2。

最佳實務

如需最佳實務，請參閱 AWS Database Migration Service 的最佳實務。

史诗

設定來源 IBM Db2 資料庫

任務	描述	所需技能
在 Amazon EC2 上建立 IBM Db2 資料庫。	您可以使用 AWS Marketplace 的 Amazon 機器映像 (AMI) 或在 EC2 執行個體上安裝 Db2 軟體，在 EC2 執行個體上建立 IBM Db2 資料庫。 透過選取 IBM Db2 的 AMI 來啟動 EC2 執行個體 (例如 IBM Db2 v11.5.7 RHEL 7.9)，這與內部部署資料庫類似。	DBA，一般 AWS
設定安全性群組。	分別使用連接埠 22 和 50000 設定 SSH (安全殼層) 和 TCP 的 VPC 安全性群組輸入規則。	一般 AWS
建立資料庫執行個體。	建立新的執行個體 (使用者) 和資料庫 (結構描述)，或使用預設`db2inst1`執行個體和範例資料庫。 1. 使用終端機 Connect 至 Db2 資料庫，連接至 EC2 執行個體。或者，您也可以安裝任何將連線至 Db2 資料庫的資料庫用戶端軟體。 2. 若要設定 db2inst1 使用者的密碼，請執行命令。`sudo passwd db2inst1` 3. 若要連線至 db2inst1 執行個體，請執行指令。`sudo su - db2inst1`	DBA

任務	描述	所需技能
	4. 若要連線至 Db2 資料庫，請執行指令`db2`。 5. 若要連線至範例資料庫，請使用指令`connect to sample`。或者，連線到您建立的資料庫。 6. 連線到資料庫執行個體之後，請使用 Db2 SQL 陳述式建立物件並將資料插入這些物件中。	
確認 Db2 資料庫執行個體可用。	若要確認 Db2 資料庫執行處理已啟動並執行，請使用指令`Db2pd -`。	DBA

設定目標 Aurora 與 MySQL 相容的資料庫

任務	描述	所需技能
建立與 MySQL 相容的 Aurora 資料庫。	使用 AWS RDS 服務建立具有 MySQL 相容性資料庫的 Amazon Aurora • 在 Amazon Aurora 上創建具有 MySQL 兼容性和您選擇的版本的數據庫，例如 Aurora（MySQL）—5.6.10a • 安裝 MySQL 工作台應用程序或您首選的數據庫客戶端軟件，它允許您連接到 MySQL 數據庫	DBA，一般 AWS
設定安全性群組。	針對 SSH 和 TCP 連線設定 VPC 安全群組輸入規則。	一般 AWS

任務	描述	所需技能
確認 Aurora 資料庫可供使用。	若要確定 Aurora 與 MySQL 相容的資料庫已啟動並執行，請執行下列動作： 1. 透過安全殼層 Connect 線至 EC2 執行個體。 2. 從 MySQL 工作台設定並連線至與 MySQL 相容的執行個體。使用端點做為主機名稱，如下列範例所示。 `mysql-cluster-instance-1.cokmvis0v46q.us-east-1.rds.amazonaws.com` 3. 建立並連線至新結構描述 (例如，`mysql-sample-db2`)。 4. 運行 MySQL 語句來檢查數據庫中的模式和對象。	DBA

設定和執行 AWS SCT

任務	描述	所需技能
安裝 AWS SCT。	下載並安裝最新版本的 AWS SCT (目前的最新版本 1.0.628)。	一般 AWS
設定 AWS SCT。	1. 下載適用於 IBM Db2 (4.22.X 版) 和 MySQL (8.x) 的 Java 資料庫連線能力 (JDBC) 驅動程式。	一般 AWS

任務	描述	所需技能
	2. 若要在 AWS SCT 中設定驅動程式，請選擇設定、全域設定、驅動程式。	
建立 AWS SCT 專案。	建立 AWS SCT 專案和報告，該專案和報告使用適用於 LUW 的 Db2 做為來源資料庫引擎，而且目標資料庫引擎使用 Aurora MySQL 相容。 若要識別連線到 LUW 資料庫的 Db2 所需的權限，請參閱使用 Db2 LUW 做為 AWS SCT 的來源。	一般 AWS

任務	描述	所需技能
驗證物件。	選擇載入綱要，驗證物件。更新目標資料庫上任何不正確的物件： 1. 透過提供連線詳細資料連線至 Amazon Aurora 與 MySQL 相容的伺服器，然後選擇 [測試連線]。 來源和目標連線都必須成功，AWS SCT 才能啟動遷移報告。 2. 報表完成後，輸入要轉換的結構描述，然後選擇 [完成]。 AWS SCT 會列出任何已轉換且有錯誤的來源和目標物件。 3. 檢閱錯誤，然後手動清除錯誤。 4. 清除所有錯誤之後，請開啟綱要的內容 (按一下滑鼠右鍵) 功能表，然後選擇「載入結構描述」。 5. 選擇「套用至資料庫」。 6. 在 MySQL 工作台中，連接到 Aurora MySQL 相容的資料庫，並檢查結構描述和物件。	DBA，一般 AWS

設定和執行 AWS DMS

任務	描述	所需技能
建立複寫執行個體。	登入 AWS 管理主控台，導覽至 AWS DMS 服務，並針對您為來源和目標資料庫設定的 VPC 安全群組建立有效設定的複寫執行個體。	一般 AWS
建立端點。	建立 Db2 資料庫的來源端點，並為 Aurora MySQL 相容資料庫建立目標端點： 1. 選擇選取 RDS 資料庫執行個體，然後選擇您建立的 Db2 執行個體，以建立 IBM Db2 作為來源的端點。端點組態詳細資料將會自動填入。 2. 在端點特定的設定中，新增下列額外的連接屬性。 `CurrentLSN=<scan>; MaxKBytesPerRead=64; SetDataCaptureChanges=true` 如果您沒有提及這些屬性，則來源端點測試連線將無法成功。如需詳細資訊，請參閱使用 IBM Db2 LUW 做為 AWS DMS 的來源。 3. 選擇選取 RDS 資料庫執行個體，然後選擇您建立的 Aurora MySQL 相容執行個體，以建立 Aurora MySQL	一般 AWS

任務	描述	所需技能
	相容的端點做為目標。端點組態詳細資料將會自動填入。如需詳細資訊，請參閱使用與 MySQL 相容的資料庫做為 AWS Database Migration Service 的目標。 4. 測試來源端點和目標端點。確認兩者都成功且可用 5. 如果測試失敗，請確定安全群組輸入規則有效。	

任務	描述	所需技能
建立移轉工作。	建立單一移轉任務或多個移轉任務以進行全負載和 CDC 或資料驗證： 1. 若要建立資料庫移轉工作，請選擇複寫執行個體、來源資料庫端點、目標資料庫端點。將移轉類型指定為「移轉現有資料 (滿載)」、「僅複製資料變更 (CDC)」或「移轉現有資料並複寫進行中的變更 (滿載和 CDC)」。 2. 在 [資料表對應] 底下，您可以設定 GUI 或 JSON 格式的選取規則和轉換規則。 3. 在選取規則下，選取綱要、輸入表格名稱，然後選取要設定的動作 (包含/排除) (例如，綱要:SAMPLE；表格名稱:%，動作：包含)。 4. 在轉換規則下，選取目標 (綱要、表格或資料欄)。選取綱要名稱，然後選擇動作 (大小寫、首碼、尾碼)；例如，目標：綱要`mysql-sample-db` ；動作：設為小寫。 5. 開啟 Amazon CloudWatch 日誌監控。	一般 AWS
計劃生產執行。	與應用程式擁有者等利益相關者確認停機時間，以便在生產系統中執行 AWS DMS。	遷移, 領導

任務	描述	所需技能
執行移轉工作。	1. 啟動狀態為就緒的 AWS DMS 任務。 2. 監控 Amazon 日誌中的遷移任務 CloudWatch 日誌是否有任何錯誤。	一般 AWS
驗證資料。	檢閱來源 Db2 和目標 MySQL 資料庫中的移轉工作結果和資料： 1. 如果狀態為 [載入完成進行中複寫]，則會完成 CDC 資料移轉的完整負載，且驗證正在進行中。 2. Connect 至 Aurora 與 MySQL 相容的資料庫，並檢查資料。 3. 通過在 Db2 數據庫中插入或更新數據來檢查正在進行的更改。	DBA
停止移轉工作。	成功完成資料驗證後，請停止驗證移轉工作。	一般 AWS

故障診斷

問題	解決方案
AWS SCT 來源和目標測試連線失敗。	設定 JDBC 驅動程式版本和 VPC 安全群組輸入規則，以接受傳入流量。
Db2 來源端點測試執行失敗。	設定額外連線設定`CurrentLSN=<scan>;` 。
工 AWSDMS 作無法連線至 Db2 來源，並傳回下列錯誤。	若要避免發生錯誤，請執行下列命令：

問題	解決方案
database is recoverable if either or both of the database configuration parameters LOGARCHMETH1 and LOGARCHMETH2 are set to ON	1. $ db2 update db cfg for sample using LOGARCHMETH1 DISK:/home/db2inst1/logs 2. $ db2stop 3. $ db2start 4. $ db2 connect to sample `SQL1116N A connection to or activation of database "SAMPLE" cannot be made because of BACKUP PENDING. SQLSTATE=57019` 5. $ db2 backup database sample to ../logs `SQL2036N The path for the file or device "../logs" is not valid` 6. $ cd 7. $ pwd `/home/db2inst1` 8. $ mkdir /tmp/backup 9. $ db2 backup database sample to /tmp/backup `Backup successful. The timestamp for this backup image is : 20190530084921` 10$ db2 connect to sample `Database Connection Information` `Database server = DB2/LINUX 9.7.1` `SQL authorization ID = DB2INST1`

問題	解決方案
	`Local database alias = SAMPLE`

相關資源

Amazon EC2

- Amazon EC2
- Amazon EC2 用戶指南

資料庫

- 數據庫
- Amazon Aurora
- 使用 Amazon Aurora MySQL

AWS

- AWS DMS 結構描述轉換
- AWS Schema Conversion Tool 使用者指南
- 使用 AWS SCT 使用者界面
- 使用 IBM Db2 LUW 做為 AWS SCT 的來源

AWS DMS

- AWS Database Migration Service
- AWS Database Migration Service 使用者指南
- 資料移轉的來源
- 資料移轉的目標
- AWS Database Migration Service 和 AWS Schema Conversion Tool 現在支援 IBM Db2 LUW 做為來源 (部落格文章)
- 移轉執行關聯式資料庫的應用程 AWS

通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB

資料來源：Amazon EC2 上的 Microsoft SQL 服務器	目標：Amazon DocumentDB	R 型：重新建築
環境：PoC 或試點	技術：雲端原生；資料庫；移轉	工作量：Microsoft
AWS 服務：Amazon EC2; Amazon DocumentDB		

Summary

此模式說明如何使用 AWS Database Migration Service (AWS DMS) 將託管在 Amazon 彈性運算雲端 (亞馬遜 EC2) 執行個體上的 Microsoft SQL 伺服器資料庫遷移到 Amazon DocumentDB 資料庫 (具有 MongoDB 相容性) 資料庫。

AWS DMS 複寫任務會讀取 SQL Server 資料庫的資料表結構、在 Amazon DocumentDB 中建立對應的集合，以及執行全負載移轉。

您也可以使用此模式將現場部署 SQL 伺服器或適用於 SQL 伺服器資料庫執行個體的 Amazon Relational Database Service 服務 (Amazon RDS) 遷移到 Amazon DocumentDB。如需詳細資訊，請參閱 AWS 規範指導網站上的將 Microsoft SQL Server 資料庫移轉至 AWS 雲端的指南。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- EC2 執行個體上現有的 SQL 伺服器資料庫。
- 在 SQL 伺服器資料庫中指派給 AWS DMS 的固定資料庫 (db_owner) 角色。如需詳細資訊，請參閱 SQL Server 文件中的資料庫層級角色。
- 熟悉使用`mongodump`、`mongorestore`、`mongoexport`和`mongoimport`公用程式將資料移入和移出 Amazon Document DB 叢集。
- Microsoft SQL 服務器管理工作室，安裝和配置。

限制

- Amazon DocumentDB 中的群集大小限制為 64 TB。如需詳細資訊，請參閱 Amazon DocumentDB 中的叢集限制。
- AWS DMS 不支援將多個來源資料表合併為單一 Amazon DocumentDB 集合。
- 如果 AWS DMS 在沒有主索引鍵的情況下處理來源資料表的任何變更，則會忽略來源資料表中的大型物件 (LOB) 資料行。

架構

源, 技術, 堆棧

- Amazon EC2

目標架構

目標技術堆疊

- Amazon DocumentDB

工具

- AWS DMS — AWS Database Migration Service (AWS DMS) 可協助您輕鬆安全地遷移資料庫。
- Amazon DocumentDB — Amazon DocumentDB（與 MongoDB 兼容性）是一種快速，可靠且全受管的數據庫服務。
- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。
- Microsoft SQL 服務器-SQL 服務器是一個關係數據庫管理系統。
- SQL 伺服器管理工作室 (SSMS) — SSMS 是用於管理 SQL 伺服器，包括存取、設定和管理 SQL 伺服器元件的工具。

史诗

建立和設定虛擬私人雲端

任務	描述	所需技能
建立 VPC。	登入 AWS 管理主控台並開啟 Amazon VPC 主控台。建立具有 IPv4 CIDR 區塊範圍的虛擬私有雲 (VPC)。	系統管理員
建立安全性群組和網路 ACL。	在 Amazon VPC 主控台上，根據您的需求為 VPC 建立安全群組和網路存取控制清單 (網路 ACL)。您也可以使用這些組態的預設設定。有關此和其他故事的更多信息，請參閱「相關資源」部分。	系統管理員

建立和設定 Amazon DocumentDB 叢集

任務	描述	所需技能
創建一個 Amazon DocumentDB 集群。	開啟 Amazon DocumentDB 主控台，然後選擇「叢集」。選擇「建立」，然後使用一個執行個體建立 Amazon DocumentDB 叢集。重要：請務必使用 VPC 的安全性群組來設定此叢集。	系統管理員
安裝蒙戈外殼。	蒙戈殼層是一個命令列公用程式，可用來連接和查詢 Amazon DocumentDB 叢集。要安裝它，請運行「/等/yum.repos.d/mongodb-org-3.6.repo」命令來創建存儲庫文	系統管理員

任務	描述	所需技能
	件。執行「須藤 yum 安裝-y mongodb-org-shell」指令來安裝蒙戈殼層。若要加密傳輸中的資料，請下載 Amazon DocumentDB 的公開金鑰，然後連線到您的 Amazon DocumentDB 執行個體。如需這些步驟的詳細資訊，請參閱 < 相關資源 > 一節。	
在 Amazon DocumentDB 集群中創建一個數據庫。	使用您的資料庫名稱執行「使用」命令，以在 Amazon DocumentDB 叢集中建立資料庫。	系統管理員

建立和設定 AWS DMS 複寫執行個體

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。	開啟 AWS DMS 主控台，然後選擇「建立複寫執行個體」。輸入複製工作的名稱和說明。選擇執行個體類別、引擎版本、儲存體、VPC、異地同步備份，並使其公開存取。選擇「高級」選項卡以設置網絡和加密設置。指定維護設定，然後選擇 [建立複製執行個體]。	系統管理員
設定 SQL 伺服器資料庫。	登入 Microsoft SQL Server，並為來源端點和 AWS DMS 複寫執行個體之間的通訊新增輸入規則。使用複寫執行個體的私有 IP 位址做為來源。重要事項：複寫執行個體和目標端點	系統管理員

任務	描述	所需技能
	應位於相同的 VPC 上。如果來源和複寫執行個體的 VPC 不同，請在安全性群組中使用替代來源。	

在 AWS DMS 中建立和測試來源和目標端點

任務	描述	所需技能
建立來源和目標資料庫端點。	開啟 AWS DMS 主控台，然後選擇「Connect 來源和目標資料庫端點」。指定來源和目標資料庫的連線資訊。如果需要，請選擇「進階」索引標籤來設定「額外連線屬性」的值。在端點組態中下載並使用憑證套件。	系統管理員
測試端點連線。	選擇「運行測試」以測試連接。透過驗證安全群組設定以及來自來源和目標資料庫執行個體的 AWS DMS 複寫執行個體的連線，對任何錯誤訊息進行疑難排解。	系統管理員

移轉資料

任務	描述	所需技能
建立 AWS DMS 遷移任務。	在 AWS DMS 主控台上，選擇「任務」、「建立任務」。指定工作選項，包括來源和目的地端點名稱，以及複製執行個體名稱。在「遷移類型」下，	系統管理員

任務	描述	所需技能
	選擇「遷移現有數據」和「僅複製數據更改」。選擇「開始任務」。	
執行 AWS DMS 移轉任務。	在 [工作設定] 底下，指定資料表準備模式的設定，例如 [不執行任何動作]、[刪除目標上的資料表]、[截斷] 和 [在複寫中包含 LOB 資料行]。設定 AWS DMS 將接受的最大 LOB 大小，然後選擇「啟用記錄」。將「高級設置」保留為默認值，然後選擇「創建任務」。	系統管理員
監控移轉。	在 AWS DMS 主控台上，選擇「任務」，然後選擇遷移任務。選擇「任務監視」以監視您的任務。當完整載入移轉完成並套用快取的變更時，工作就會停止。	系統管理員

測試並驗證移轉

任務	描述	所需技能
通過使用蒙戈外殼 Connect 到 Amazon DocumentDB 集群。	開啟 Amazon DocumentDB 主控台，在「叢集」下選擇您的叢集。在「Connect 和安全性」選項卡中，選擇「使用 mongo 外殼連接到此集群」。	系統管理員
驗證移轉的結果。	使用數據庫名稱運行「use」命令，然後運行「顯示集合」命令。運行「db。.count ( ) ;」命令與您的數據庫的名稱。如	系統管理員

任務	描述	所需技能
	果結果與來源資料庫相符，表示移轉成功。	

相關資源

建立和設定虛擬私人雲端

- 為您的 VPC 建立安全群組
- 建立網路 ACL

建立和設定 Amazon DocumentDB 叢集

- 創建一個 Amazon DocumentDB 集群
- 安裝蒙戈外殼 Amazon DocumentDB
- Connect 到您的 Amazon DocumentDB 集群

建立和設定 AWS DMS 複寫執行個體

- 使用公有和私有複寫執行個體

在 AWS DMS 中建立和測試來源和目標端點

- 使用 Amazon DocumentDB 作為 AWS DMS 的目標
- 使用 SQL 伺服器資料庫做為 AWS DMS 的來源
- 使用 AWS DMS 端點

移轉資料

- 遷移到 Amazon DocumentDB

其他資源

- 使用 SQL 伺服器做為 AWS DMS 來源的限制
- 如何使用 Amazon DocumentDB 大規模建立和管理應用程序

將現場部署 ThoughtSpot 獵鷹資料庫遷移到 Amazon Redshift

創建者：巴圖爾加·普雷瓦拉查 (AWS) 和安東尼·普拉薩德·泰瓦拉治 (AWS)

環境：PoC 或試點	來源：本地 ThoughtSpot 獵鷹數據庫	目標：Amazon Redshift
R 型：重新建築	工作負載：所有其他工作	技術：移轉；資料庫
AWS 服務：AWS DMS；Amazon Redshift		

Summary

內部部署資料倉儲需要大量的管理時間和資源，尤其是大型資料集。建造，維護和增長這些倉庫的財務成本也非常高。為了協助管理成本、降低擷取、轉換和載入 (ETL) 複雜性，並在資料成長時提供效能，您必須不斷選擇要載入的資料以及要封存的資料。

透過將現場部署 ThoughtSpot Falcon 資料庫遷移到 Amazon Web Services (AWS) 雲端，您可以存取雲端資料湖和資料倉儲，從而提高業務靈活性、安全性和應用程式可靠性，同時降低整體基礎設施成本。Amazon Redshift 有助於大幅降低資料倉儲的成本和營運開銷。您也可以使用 Amazon Redshift Spectrum 以原生格式分析大量資料，而無需載入資料。

此模式描述了將 ThoughtSpot Falcon 資料庫從現場部署資料中心遷移到 AWS 雲端上的 Amazon Redshift 資料庫的步驟和程序。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在內部部署資料中心託管的 ThoughtSpot Falcon 資料庫

產品版本

- ThoughtSpot 7.0.1 版本

架構

該圖顯示以下工作流程：

1. 資料託管在內部部署關聯式資料庫中。
2. AWS Schema Conversion Tool (AWS SCT) 會轉換與 Amazon Redshift 相容的資料定義語言 (DDL)。
3. 建立表格之後，您可以使用 AWS Database Migration Service (AWS DMS) 遷移資料。
4. 該數據被加載到 Amazon Redshift。
5. 如果您使用 Redshift 頻譜或已經在 Amazon S3 中託管資料，則資料會存放在亞馬遜簡單儲存服務 (Amazon S3) 中。

工具

- AWS DMS — AWS 資料遷移服務 (AWS DMS) 可協助您快速安全地將資料庫遷移到 AWS。
- Amazon Redshift — Amazon Redshift 是一種快速、全受管的 PB 級資料倉儲服務，可讓您使用現有的商業智慧工具，以簡單且經濟實惠的方式，有效率地分析所有資料。
- AWS SCT — AWS Schema Conversion Tool (AWS SCT) 可將現有的資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎。

史诗

準備移轉

任務	描述	所需技能
識別適當的 Amazon Redshift 組態。	根據您的需求和資料量，識別適當的 Amazon Redshift 叢集組態。 如需詳細資訊，請參閱 Amazon Redshift 文件中的 Amazon Redshift 叢集。	DBA

任務	描述	所需技能
研究 Amazon Redshift 以評估它是否符合您的要求。	使用 Amazon Redshift 常見問答集來了解和評估 Amazon Redshift 是否符合您的需求。	DBA

準備目標 Amazon Redshift 集群

任務	描述	所需技能
創建一個 Amazon Redshift 集群。	登入 AWS 管理主控台，開啟 Amazon Redshift 主控台，然後在虛擬私有雲端 (VPC) 中建立 Amazon Redshift 叢集。 如需詳細資訊，請參閱 Amazon Redshift 文件中的在 VPC 中建立叢集。	DBA
為您的 Amazon Redshift 資料庫設計執行 PoC。	透過為資料庫設計執行概念驗證 (PoC)，以遵循 Amazon Redshift 最佳實務。 如需詳細資訊，請參閱亞馬遜紅移文件中的對亞馬 Amazon Redshift 進行概念驗證。	DBA
建立資料庫使用者。	在 Amazon Redshift 資料庫中建立使用者，並授與適當的角色以存取結構描述和表格。 如需詳細資訊，請參閱 Amazon Redshift 文件中的授予使用者或使用者群組的存取權限。	DBA

任務	描述	所需技能
將組態設定套用至目標資料庫。	根據您的需求，將組態設定套用至 Amazon Redshift 資料庫。 如需有關啟用資料庫、工作階段和伺服器層級參數的詳細資訊，請參閱 Amazon Redshift 文件中的組態參考。	DBA

在 Amazon Redshift 群集中創建對象

任務	描述	所需技能
在 Amazon Redshift 中使用 DDL 手動創建表。	(選擇性) 如果您使用 AWS SCT，則會自動建立資料表。不過，如果複寫 DDL 時發生失敗，您必須手動建立資料表	DBA
建立 Redshift 頻譜的外部表格。	使用 Amazon Redshift Spectrum 的外部結構描述建立外部資料表。若要建立外部表格，您必須是外部結構描述的擁有者或資料庫超級使用者。 如需詳細資訊，請參閱亞馬遜紅移文件中的為 Amazon Redshift Spectrum 建立外部表格。	DBA

使用 AWS DMS 遷移資料

任務	描述	所需技能
使用 AWS DMS 來遷移資料。	在 Amazon Redshift 資料庫中建立表格的 DDL 之後，請使用 AWS DMS 將資料遷移到 Amazon Redshift 移。 如需詳細步驟和指示，請參閱 AWS DMS 文件中的使用 Amazon Redshift 資料庫做為 AWS DMS 的目標。	DBA
使用 COPY 指令載入資料。	使用 Amazon Redshift COPY 命令將數據從 Amazon S3 加載到 Amazon Redshift。 如需詳細資訊，請參閱亞馬遜紅移文件中的使用 COPY 命令從 Amazon S3 載入。	DBA

驗證 Amazon Redshift 集群

任務	描述	所需技能
驗證來源和目標記錄。	驗證從來源系統載入的來源和目標記錄的表格計數。	DBA
實作 Amazon Redshift 最佳實務以進行效能調整。	實作表格和資料庫設計的 Amazon Redshift 最佳實務。 如需詳細資訊，請參閱部落格文章 Amazon Redshift 的十大效能調整技術。	DBA
最佳化查詢效能。	Amazon Redshift 使用 SQL 查詢與系統中的資料和物件	DBA

任務	描述	所需技能
	互動。資料操作語言 (DML) 是 SQL 的子集，您可以用來檢視、新增、變更和刪除資料。DDL 是用來新增、變更及刪除資料庫物件 (例如資料表和檢視) 的 SQL 子集。 如需詳細資訊，請參閱 Amazon Redshift 文件中的調整查詢效能。	
實作 WLM。	您可以使用工作負載管理 (WLM) 定義多個查詢佇列，並在執行時期將查詢路由至適當的佇列。 如需詳細資訊，請參閱 Amazon Redshift 文件中的實作工作負載管理。	DBA
使用並行縮放。	透過使用並行擴展功能，您可以支援幾乎無限制的並行使用者和並行查詢，並具有一致的快速查詢效能。 如需詳細資訊，請參閱 Amazon Redshift 文件中的使用並行擴展。	DBA

任務	描述	所需技能
使用 Amazon Redshift 最佳實務進行表格設計。	規劃資料庫時，某些重要的資料表設計決策可能會對整體查詢效能產生極大的影響。 如需有關選擇最合適資料表設計選項的詳細資訊，請參閱 Amazon Redshift 說明文件中的 Amazon Redshift 設計資料表的最佳實務。	DBA
在 Amazon Redshift 中創建具體化視圖。	具體化視觀表包含根據一或多個基礎表格上 SQL 查詢的預先計算結果集。您可以使用與查詢資料庫中其他表格或視觀表相同的方式，發出SELECT敘述句來查詢具體化視觀表。 如需詳細資訊，請參閱 Amazon Redshift 文件中的在 Amazon Redshift 中建立具體化視圖。	DBA
定義表之間的連接。	若要在中同時搜尋一個以上的資料表 ThoughtSpot，您必須指定包含跨兩個資料表相符資料的欄，以定義資料表之間的聯結。這些欄代表聯結foreign key的primary key和。 您可以通過使用 Amazon Redshift 或ALTER TABLE ThoughtSpot命令來定義它們。如需詳細資訊，請參閱 Amazon Redshift 文件中的 ALTER 表。	DBA

設置 ThoughtSpot 與 Amazon Redshift 的連接

任務	描述	所需技能
添加一個 Amazon Redshift 連接。	將 Amazon Redshift 連接添加到您的現場部署 ThoughtSpot Falcon 數據庫。 如需詳細資訊，請參閱 ThoughtSpot 文件中的新增 Amazon Redshift 連線。	DBA
編輯 Amazon Redshift 連接。	您可以編輯 Amazon Redshift 連接以添加表格和列。 如需詳細資訊，請參閱 ThoughtSpot 文件中的編輯 Amazon Redshift 連線。	DBA
重新映射 Amazon Redshift 連接。	編輯新增 Amazon Redshift 連線時建立的來源對應 .yaml 檔案，以修改連線參數。 例如，您可以將現有的資料表或資料行重新對應至現有資料庫連線中的其他資料表或資料行。 ThoughtSpot 建議您在重新對映連線中的表格或欄之前和之後檢查相依性，以確保它們依需要顯示。 如需詳細資訊，請參閱文件中的重新對應 Amazon Redshift 連線。 ThoughtSpot	DBA
從 Amazon Redshift 連接中刪除一個表。	(選擇性) 如果您嘗試移除 Amazon Redshift 連線中的表格，請 ThoughtSpot 檢查相依性並顯示相依物件的清單。您	DBA

任務	描述	所需技能
	可以選擇列出的物件來刪除它們或移除相依性。然後，您可以移除表格。 如需詳細資訊，請參閱 ThoughtSpot 文件中的從 Amazon Redshift 連線刪除表格。	
從 Amazon Redshift 連線中刪除含有相依物件的表格。	(選擇性) 如果您嘗試刪除具有相依物件的資料表，則會封鎖該作業。會出現一個`Cannot delete`視窗，其中包含相依物件的連結清單。刪除所有依賴關係後，您可以刪除該表 如需詳細資訊，請參閱 ThoughtSpot 文件中的從 Amazon Redshift 連線刪除包含相依物件的表格。	DBA
刪除亞 Amazon Redshift 連接。	(選擇性) 由於連線可用於多個資料來源或視覺效果，因此您必須先刪除使用該連線的所有來源和任務，才能刪除 Amazon Redshift 連線。 如需詳細資訊，請參閱 ThoughtSpot 文件中的刪除 Amazon Redshift 連線。	DBA
檢查 Amazon Redshift 的連接參考。	請確定您使用 ThoughtSpot 文件中的連線參考提供 Amazon Redshift 連線的必要資訊。	DBA

其他資訊

- 利用 ThoughtSpot 和 Amazon Redshift，任何規模的 AI 驅動分析
- Amazon Redshift 定價
- 開始使用 AWS SCT
- 開始使用 Amazon Redshift
- 使用資料擷取代理
- 小雞 FIL-提高了與 AWS 洞察的速度 ThoughtSpot

使用 AWS DMS 將甲骨文資料庫遷移到 Amazon DynamoDB 資料庫

創建者：蘭巴布卡內納（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon DynamoDB
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon DynamoDB		

Summary

此模式會引導您完成使用 AWS Database Migration Service (AWS D MS) 將甲骨文資料庫遷移到 Amazon DynamoDB 的步驟。它涵蓋了三種類型的源數據庫：

- 內部部署甲骨文
- Amazon 彈性運算雲上的甲骨文數據庫（Amazon EC2）
- 適用於甲骨文資料庫執行個體的 Amazon Relational Database Service 服務

在此概念證明中，此模式著重於從適用於 Oracle 資料庫執行個體的 Amazon RDS 進行遷移。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 連線到適用於甲骨文資料庫的應用程式
- 在來源 Amazon RDS for Oracle 文資料庫中建立的資料表，其中包含主索引鍵和範例資料

限制

- 由於 Amazon DynamoDB 不支援這些資料庫物件，因此不會將 Oracle 資料庫物件 (例如程序、函數、套件和觸發器) 納入移轉考量。

產品版本

- 此模式適用於 AWS DMS 支援的所有 Oracle 資料庫版本和版本。如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源和使用 Amazon DynamoDB 資料庫做為 AWS DMS 的目標。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。

架構

源, 技術, 堆棧

- Amazon RDS for Oracle 文數據庫執行個體、Amazon EC2 上的甲骨文或現場部署 Oracle

目標技術堆疊

- Amazon DynamoDB

AWS 資料遷移架構

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。這種模式使用 Amazon RDS for Oracle。

史诗

規劃移轉

任務	描述	所需技能
建立 VPC。	在您的 AWS 帳戶中，建立虛擬私有雲 (VPC) 和私有子網路。	系統管理員

任務	描述	所需技能
建立安全性群組和網路存取控制清單。	如需詳細資訊，請參閱 AWS 文件。	系統管理員
設定並啟動適用於甲骨文資料庫執行個體的 Amazon RDS。	如需詳細資訊，請參閱 AWS 文件。	DBA, 系統管理員

移轉資料

任務	描述	所需技能
建立身分與存取權管理角色以存取 DynamoDB。	在 AWS Identity and Access Management (IAM) 主控台中，建立角色、附加政策`AmazonDynamoDBFullAccess to it`，然後選取 AWS DMS 做為服務。	系統管理員
建立用於移轉的 AWS DMS 複寫執行個體。	複寫執行個體應與來源資料庫位於相同的可用區域和 VPC 中。	系統管理員
在 AWS DMS 中建立來源和目標端點。	若要建立來源資料庫端點，您有兩個選項： • 在 Amazon RDS 主控台上，選擇資料庫、資料庫識別碼、連線和安全性，然後選擇端點。 • 在 AWS DMS 主控台上，選擇選取 RDS 資料庫執行個體。 若要建立目標資料庫端點，請從上一個任務中選擇 Amazon	系統管理員

任務	描述	所需技能
	資源名稱 (ARN) 角色以存取 DynamoDB。	
建立 AWS DMS 任務，將來源 Oracle 資料庫表格載入 DynamoDB。	從先前步驟中選擇來源和目標端點名稱以及複製執行個體。該類型可以是滿載。選擇 Oracle 綱要並指定% 以選取所有表格。	系統管理員
驗證動態資料表中的資料表。	若要檢視移轉結果，請從 Dynam oDB 主控台的左側導覽窗格中選擇 [表格]。	DBA

移轉應用程式

任務	描述	所需技能
修改應用程式程式碼。	若要連線至 DynamoDB 並從中擷取資料，請更新應用程式程式碼。	應用程式擁有者、DBA、系統管理員

切過

任務	描述	所需技能
將應用程式用戶端切換為使用 DynamoDB。		DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉 AWS 資源。	例如，關閉 Amazon RDS 適用於甲骨文執行個體、DynamoD	DBA, 系統管理員

任務	描述	所需技能
	B 和 AWS DMS 複寫執行個體。	
收集指標。	指標包括移轉時間、工具所執行的手動作業和工作的百分比，以及節省成本。	DBA、應用程式擁有者、系統管理員

相關資源

- AWS Database Migration Service 和 Amazon DynamoDB：您需要知道的事項 (部落格文章)
- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 使用 Amazon DynamoDB 庫做為 AWS Database Migration Service 的目標
- 從資料庫管理系統移轉至 Amazon DynamoDB 最佳實務 (白皮書)

使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL

創建者：索拉夫密什拉（AWS）和愛德華多·瓦倫蒂姆（AWS）

環境：PoC 或試點	來源：甲骨文數據庫	目標：PostgreSQL
R 型：重新建築	工作量：甲骨文	技術：移轉、資料庫、儲存與備份
AWS 服務：AWS DMS		

Summary

此模式說明如何使用不支援原生分割的 AWS Database Migration Service (AWS DMS)，加快將分區資料表從 Oracle 載入 PostgreSQL 的速度。目標 PostgreSQL 資料庫可以安裝在亞馬遜彈性運算雲端 (Amazon EC2) 上，也可以是適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 或與 Amazon Aurora PostgreSQL 相容的版本資料庫執行個體。

上傳分區資料表包括下列步驟：

1. 建立類似 Oracle 分割區表格的父項表格，但不要包含任何分割區。
2. 建立將繼承您在步驟 1 中建立的父資料表的子資料表。
3. 建立程序函數和觸發程序，以處理父資料表中的插入。

但是，由於每次插入都會觸發觸發，因此使用 AWS DMS 的初始負載可能會非常緩慢。

為了加快從 Oracle 到 PostgreSQL 9.0 的初始載入速度，此模式會為每個分割區建立單獨的 AWS DMS 任務，並載入對應的子資料表。然後，您可以在切換期間建立觸發器。

第 10 PostgreSQL 援原生磁碟分割。但是，在某些情況下，您可能會決定使用繼承的分區。如需詳細資訊，請參閱其他資訊一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- 具有分區資料表的來源 Oracle 資料庫
- AWS PostgreSQL 資料庫

產品版本

- PostgreSQL

架構

源, 技術, 堆棧

- 甲骨文中的分區資料表

目標技術堆疊

- PostgreSQL 中的分區資料表 (在亞馬 Amazon EC2、亞馬遜 RDS 版或 Aurora)

目標架構

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。

史诗

設定 AWS DMS

任務	描述	所需技能
在 PostgreSQL 中創建表。	使用分區所需的檢查條件在 PostgreSQL 中創建父表和對應的子表。	DBA
為每個分割區建立 AWS DMS 任務。	在 AWS DMS 任務中包含分區的篩選條件。將磁碟分割對應	DBA

任務	描述	所需技能
	至對應的 PostgreSQL 子資料表。	
使用全負載和變更資料擷取 (CDC) 執行 AWS DMS 任務。	請確定`StopTaskCachedChangesApplied` 參數已設定為，`true`且`StopTaskCachedChangesNotApplied` 參數設定為`false`。	DBA

切過

任務	描述	所需技能
停止複寫工作。	停止工作之前，請確認來源和目的地是否同步。	DBA
在父資料表上建立觸發程式。	由於父資料表會接收所有的插入和更新命令，因此請建立觸發程序，根據分割條件將這些命令路由到相應的子資料表。	DBA

相關資源

- AWS DMS
- 資料表分 PostgreSQL 件集)

其他資訊

雖然 PostgreSQL 版本 10 支援原生磁碟分割，但您可能會決定在下列使用案例中使用繼承的磁碟分割：

- 分區強制執行一項規則，即所有分區必須具有與父級相同的一組列，但是表繼承支持具有額外列的子級。

- 資料表繼承支援多重繼承。
- 宣告式磁碟分割僅支援清單和範圍分割。使用表繼承，您可以根據需要劃分數據。但是，如果條件約束排除無法有效地修剪分割區，則查詢效能將受到影響。
- 使用宣告式磁碟分割時，某些作業需要比使用資料表繼承時更強的鎖定。例如，在分區資料表中新增或移除分區資料表時，需要ACCESS EXCLUSIVE鎖定父資料表，而SHARE UPDATE EXCLUSIVE鎖定則足以進行一般繼承。

當您使用不同的任務分割區時，如果發生任何 AWS DMS 驗證問題，也可以重新載入分區。為了獲得更好的效能和複寫控制，請在不同的複寫執行個體上執行

從 Amazon RDS for Oracle 遷移到 Amazon RDS for MySQL

由吉泰爾庫馬爾（AWS），妮哈·沙爾馬（AWS）和斯里尼·拉馬斯瓦米（AWS）創建

環境：PoC 或試點	資料來源：Amazon RDS for Oracle	目標：適用於 MySQL 的 Amazon RDS for MySQL
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式提供有關將適用於甲骨文資料庫執行個體的 Amazon Relational Database Service 服務 (Amazon RDS) 遷移到亞馬遜網路服務 (AWS) 上的 Amazon RDS for MySQL for MySQL 資料庫執行個體的指導。該模式使用 AWS Database Migration Service (AWS DMS) 和 AWS Schema Conversion Tool (AWS SCT)。

該模式提供了處理存儲過程遷移的最佳實踐。它還涵蓋和代碼更改以支持應用層。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 一個 Amazon RDS for Oracle 源數據庫。
- 一個適用於 MySQL 的 Amazon RDS for MySQL 目標資料庫。來源和目標資料庫應位於相同的虛擬私有雲 (VPC) 中。如果您使用多個 VPC，或者您必須擁有必要的存取權限。
- 允許來源和目標資料庫、AWS SCT、應用程式伺服器和 AWS DMS 之間連線的安全群組。
- 具有在來源資料庫上執行 AWS SCT 所需權限的使用者帳戶。
- 在來源資料庫上執行 AWS DMS 時啟用補充記錄。

限制

- 來源和目標 Amazon RDS 資料庫大小限制為 64 TB。如需 Amazon RDS 大小的資訊，請參閱 AWS 文件。

- 甲骨文對數據庫對象不區分大小寫，但 MySQL 不區分大小寫。AWS SCT 可以在建立物件時處理此問題。但是，需要一些手動工作才能支持完全不區分大小寫。
- 此遷移不會使用 MySQL 擴充功能來啟用 Oracle 原生函式。AWS SCT 可處理大部分的轉換，但需要一些工作才能手動變更程式碼。
- Java 資料庫連線 (JDBC) 驅動程式需要在應用程式中變更。

產品版本

- Amazon RDS for Oracle 12.2.0.1 及更高版本。如需 Oracle 版本目前支援的 RDS 資訊，請參閱 AWS 文件。
- Amazon RDS for MySQL 適用於 MySQL 8.0.15 及更高版本。如需目前支援的適用於 MySQL 的 RDS 版本，請參閱 AWS 文件。
- AWS DMS 版本 3.3.0 及更新版本。如需 AWS DMS 支援的來源端點和目標端點的詳細資訊，請參閱 AWS 文件。
- AWS SCT 版本 1.0.628 及更新版本。 請參閱 AWS 文件中的 AWS SCT 來源和目標端點支援對照表。

架構

源, 技術, 堆棧

- Amazon RDS for Oracle。如需詳細資訊，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源。

目標技術堆疊

- 適用於 MySQL 的 Amazon RDS for MySQL。如需詳細資訊，請參閱使用與 MySQL 相容的資料庫做為 AWS DMS 的目標。

移轉架構

在下圖中，AWS SCT 會從 Amazon RDS for Oracle 來源資料庫複製和轉換結構描述物件，然後將物件傳送到 Amazon RDS for MySQL 版 MySQL 目標資料庫。AWS DMS 會從來源資料庫複寫資料，並將其傳送到 Amazon RDS for MySQL 執行個體。

工具

- AWS 資料遷移服務可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。這種模式使用 Amazon RDS for Oracle 和 Amazon RDS for MySQL 的 MySQL。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

史诗

準備移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA SysAdmin
選擇適當的執行個體類型 (容量、儲存空間功能、網路功能)。		DBA SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA SysAdmin
選擇應用程式移轉策略。	考慮是否要完全停機時間或切換活動的部分停機時間。	DBA,, 應用程式 SysAdmin擁有者

設定基礎結

任務	描述	所需技能
建立 VPC 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動適用於甲骨文執行個體的亞馬遜 RDS。		DBA SysAdmin
設定並啟動適用於 MySQL 的 Amazon RDS for MySQL 執行個體。		DBA SysAdmin
準備一個測試用例以驗證代碼轉換。	這將有助於單元測試轉換後的代碼。	DBA, 開發人員
設定 AWS DMS 執行個體。		
在 AWS DMS 中設定來源端點和目標端點。		

移轉資料

任務	描述	所需技能
使用 AWS SCT 產生目標資料庫指令碼。	檢查 AWS SCT 轉換的程式碼的準確性。將需要一些手動工作。	DBA, 開發人員
在 AWS SCT 中，選擇「不區分大小寫」設定。	在 AWS SCT 中，選擇「專案設定」、「目標區分大小寫」、「不區分大小寫」。	DBA, 開發人員

任務	描述	所需技能
在 AWS SCT 中，選擇不使用 Oracle 原生函數。	在「專案設定」中，勾選「_CHAR/到_編號/到_DATE」的功能。	DBA, 開發人員
對「sql%找不到」程式碼進行變更。	您可能必須手動轉換代碼。	
查詢預存程序中的資料表和物件 (使用小寫查詢)。		DBA, 開發人員
完成所有變更後建立主要指令碼，然後在目標資料庫上部署主要指令碼。		DBA, 開發人員
使用範例資料進行單元測試預存程序和應用程式呼叫。		
清理單元測試期間創建的數據。		DBA, 開發人員
卸除目標資料庫上的外來索引鍵條件約束。	此步驟是載入初始資料所需的。如果您不想刪除外部索引鍵條件約束，則必須針對主要和次要資料表的特定資料建立移轉工作。	DBA, 開發人員
在目標資料庫上放置主索引鍵和唯一索引鍵。	此步驟會為初始負載帶來更好的效能。	DBA, 開發人員
啟用來源資料庫的補充記錄日誌。		DBA
在 AWS DMS 中為初始負載建立遷移任務，然後執行它。	選擇移轉現有資料的選項。	DBA
將主鍵和外鍵添加到目標數據庫。	需要在初始負載之後添加約束。	DBA, 開發人員

任務	描述	所需技能
建立進行中複寫的移轉工作。	進行中的複寫會使目標資料庫與來源資料庫保持同步。	DBA

遷移應用

任務	描述	所需技能
將甲骨文本地函數替換為 MySQL 本地函數。		應用所有者
請確定 SQL 查詢中的資料庫物件僅使用小寫名稱。		DBA,, 應用程式 SysAdmin擁有者

切換到目標數據庫

任務	描述	所需技能
關閉應用程式伺服器。		應用所有者
驗證來源和目標資料庫是否同步。		DBA，應用程式擁有者
停止適用於 Oracle 資料庫執行個體的亞馬遜 RDS。		DBA
停止移轉工作。	這將在您完成上一步後自動停止。	DBA
改變 JDBC 連接從甲骨文到 MySQL。		應用程式擁有者，DBA
啟動應用程式。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
審核並驗證專案文件。		DBA SysAdmin
收集有關遷移時間、手動與工具任務的百分比、節省成本等指標。		DBA SysAdmin
停止和刪除 AWS DMS 執行個體。		DBA
移除來源端點和目標端點。		DBA
移除移轉工作。		DBA
拍攝 Amazon RDS for Oracle 資料庫執行個體的快照。		DBA
刪除適用於 Oracle 資料庫執行個體的亞馬遜 RDS。		DBA
關閉並刪除您使用的任何其他臨時 AWS 資源。		DBA SysAdmin
關閉專案並提供任何意見反應。		DBA

相關資源

- AWS DMS
- AWS
- Amazon RDS 定價
- 開始使用 AWS DMS
- Amazon RDS 入門

使用 AWS DMS 和 AWS SCT，從 Amazon EC2 上的 IBM Db2 遷移到 Aurora 與 PostgreSQL 相容

創建者：西爾森杜哈爾德（AWS）和薩欽科德沃爾（AWS）

環境：PoC 或試點	資料來源:	目標：Aurora 郵政兼容
R 型：重新建築	工作負載：IBM	技術：移轉；資料庫
AWS 服務：Amazon Aurora；AWS DMS；AWS SCT		

Summary

此模式提供將 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上的 IBM Db2 資料庫移轉至 Amazon Aurora PostgreSQL 相容版本資料庫執行個體的指導。此模式使用 AWS Database Migration Service (AWS DMS) 和 AWS Schema Conversion Tool (AWS SCT) 進行資料遷移和結構描述轉換。

該模式針對具有大量交易的多 TB IBM Db2 資料庫的停機時間很少或沒有停機時間的線上移轉策略。我們建議您將具有資料類型的主索引鍵 (PKs) 和外鍵 (FK) 中的資料行轉換為 PostgreSQL `INT` 或`BIGINT`在 PostgreSQL 中，`NUMERIC`以獲得更好的效能。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- EC2 執行個體上的來源 IBM Db2 資料庫

產品版本

- DB2/LINUXX64 版本 11.1.4.4 及更新版本

架構

源, 技術, 堆棧

- EC2 執行個體上的 Db2 資料庫

目標技術堆疊

- 相容於 Aurora PostSQL 版本 10.18 或更新版本的資料庫執行個體

資料庫遷移架構

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料庫遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。您可以使用 AWS DMS 在使用最廣泛的商業和開放原始碼資料庫之間移轉資料。AWS DMS 支援不同資料庫平台之間的異質移轉，例如 IBM Db2 到 Aurora PostgreSQL 相容版本 10.18 或更高版本。如需詳細資訊，請參閱 AWS DMS 文件中的資料遷移來源和資料遷移的目標。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分資料庫程式碼物件 (包括檢視、預存程序和函數) 轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。任何未自動轉換的物件都會清楚標示，以便手動轉換它們以完成移轉。AWS SCT 也可以掃描應用程式原始程式碼以尋找內嵌 SQL 陳述式並進行轉換。

史诗

設定環境

任務	描述	所需技能
建立與 PoAurora greSQL 相容的資料庫執行個體。	若要建立資料庫執行個體，請遵循 AWS 文件中的指示。對於引擎類型，請選擇 Amazon Aurora。對於版本，請選擇與 Amazon Aurora PostgreSQL 兼容的版本。	Amazon RDS

任務	描述	所需技能
	與 Aurora PostgreSQL 相容版本 10.18 或更新版本的資料庫執行個體應與來源 IBM Db2 資料庫位於相同的虛擬私有雲端 (VPC) 中。	

轉換您的數據庫模式

任務	描述	所需技能
安裝並驗證 AWS SCT。	1. 按照 AWS SCT 文件中的步驟安裝 AWS SCT。 2. 按照 AWS SCT 文件中的程序驗證安裝。	AWS 管理員、DBA、移轉工程師
啟動 AWS SCT 並建立專案。	若要啟動 AWS SCT 工具並建立新專案以執行資料庫遷移評估報告，請遵循 AWS SCT 文件中的指示。	移民工程師
新增資料庫伺服器並建立對應規則。	1. 按照 AWS SCT 文件中的指示，新增來源和目標資料庫伺服器。 2. 建立對應規則以定義來源資料庫的目標資料庫平台。如需相關指示，請參閱 AWS SCT 文件。	移民工程師
建立資料庫移轉評估報告。	按照 AWS SCT 文件中的步驟建立資料庫遷移評估報告。	移民工程師
檢視評估報告。	使用資料庫移轉評估報告的 [摘要] 索引標籤來檢視報表並分析資料。此分析可協助您判斷移	移民工程師

任務	描述	所需技能
	轉的複雜性。如需詳細資訊，請參閱 AWS SCT 文件。	
轉換結構描述。	若要轉換來源資料庫結構描述： 1. 在 AWS SCT 主控台上，選擇檢視，然後選擇主視圖。 2. 從來源結構描述中選取物件或父節點，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇 [轉換結構描述]。 如需詳細資訊，請參閱 AWS SCT 文件。	移民工程師
將轉換的資料庫結構描述套用至目標資料庫執行個體	1. 在顯示您目標資料庫執行個體之計劃結構描述的專案右側面板中，選擇結構描述元素。 2. 開啟結構描述元素的內容 (按一下右鍵) 選單，然後選擇 Apply to database (套用至資料庫)。 如需詳細資訊，請參閱 AWS SCT 文件。	移民工程師

移轉您的資料

任務	描述	所需技能
設定 VPC 和資料庫參數群組。	設定 VPC 和 DB 參數群組，並設定移轉所需的輸入規則和	移民工程師

任務	描述	所需技能
	參數。如需相關指示，請參閱 AWS DMS 文件。 對於 VPC 安全群組，請為 Db2 選取 EC2 執行個體，以及與 Aurora PostgreSQL 相容的資料庫執行個體。此複寫執行個體必須與來源和目標資料庫執行個體位於相同的 VPC 中。	
準備來源和目標資料庫執行個體。	準備要移轉的來源和目標資料庫執行個體。在生產環境中，來源資料庫將已存在。 對於來源資料庫，伺服器名稱必須是執行 Db2 所在 EC2 執行個體的公用網域名稱系統 (DNS)。對於用戶名，您可以使用`db2inst1`後面的端口，這將是 5000 的 IBM Db2。	移民工程師

任務	描述	所需技能
建立 Amazon EC2 用戶端和端點。	1. 創建一個 Amazon EC2 客戶端。您可以使用此用戶端將要複寫的資料填入來源資料庫。您也可以使用此用戶端，透過在目標資料庫上執行查詢來驗證複寫。 2. 為來源資料庫和目標資料庫執行個體建立端點，以用於後續步驟。如需相關指示，請參閱 AWS DMS 文件。您必須為來源和目標資料庫建立個別的端點。對於與 Aurora PostgreSQL 相容的 10.18 版或更新版本，連接埠將是 5432，而且您可以從資料庫執行個體的端點取得伺服器名稱。	移民工程師
建立複寫執行個體。	使用 AWS DMS 主控台建立複寫執行個體，並指定來源和目標端點。複製執行個體會在端點之間執行資料移轉。如需詳細資訊，請參閱 AWS DMS 文件。	移民工程師

任務	描述	所需技能
建立 AWS DMS 任務以遷移資料。	依照 AWS DMS 說明文件中的步驟，建立任務，將來源 IBM Db2 表格載入目標 PostgreSQL 資料庫執行個體。 • 對於來源和目標，請使用來源和目標端點名稱。 • 遷移類型可以是全負載。 • 對於結構描述規則，您可以使用 Db2 資料庫中的`inst1`結構描述。 • 對於表格名稱，請指定`%`轉所有表格。當載入完成時，您會看到`inst1`結構描述的 Db2 資料表出現在 Aurora PostgreSQL 相容資料庫中。	移民工程師

相關資源

參考

- Amazon Aurora 文檔
- PostgreSQL 料包裝 (FDW) 文件
- 匯入外部結構描述文件
- AWS DMS 說明文件
- AWS SCT 文件

教學課程和影片

- 開始使用 AWS DMS (逐步解說)
- Amazon EC2 簡介-彈性雲端伺服器和 AWS 託管 (影片)

使用和 AWS DMS 從甲骨文 8i 或 9i 遷移到亞馬遜 RDS SharePlex

創建者：庫馬爾·巴布 P G（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：亞馬遜 RDS 後服務/ Amazon Aurora
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

此模式說明如何將現場部署 Oracle 8i 或 9i 資料庫遷移到適用於 PostgreSQL 或 Amazon 極光的 Amazon Relational Database Service 服務 (亞馬遜 RDS)。AWS Database Migration Service (AWS DMS) 不支援 Oracle 8i 或 9i 做為來源，因此 Quest 會將資料從現場部署 8i 或 9i 資料庫 SharePlex 複製到與 AWS DMS 相容的中繼 Oracle 資料庫 (甲骨文 10g 或 11g)。

透過使用 AWS 結構描述轉換工具 (AWS SCT) 和 AWS DMS，將結構描述和資料從中繼 Oracle 執行個體遷移到 AWS 上的 PostgreSQL 資料庫。此方法有助於以最小的複寫延遲，將資料從來源 Oracle 資料庫連續串流至目標 PostgreSQL 資料庫執行個體。在此實作中，停機時間限制為在目標 PostgreSQL 資料庫上建立或驗證所有外部索引鍵、觸發程序和序列所需的時間長度。

遷移過程使用安裝 Oracle 10g 或 11g 的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體來託管來自來源 Oracle 資料庫的變更。AWS DMS 使用此中繼 Oracle 執行個體做為來源，將資料串流至 Amazon RDS for PostgreSQL) 或 Aurora PostgreSQL。您可以從內部部署 Oracle 資料庫暫停和繼續資料複寫至中繼 Oracle 執行個體。它也可以從中繼 Oracle 執行個體暫停和恢復到目標 PostgreSQL 資料庫，以便您可以使用 AWS DMS 資料驗證或自訂資料驗證工具來驗證資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Oracle 8i 或 9i 資料庫
- 在現場部署資料中心和 AWS 之間設定 AWS Direct Connect

- 適用於 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式，可安裝在本機電腦或已安裝 AWS SCT 的 EC2 執行個體上
- 熟悉使用 Oracle 資料庫做為 AWS DMS 來源
- 熟悉使用 PostgreSQL 資料庫做為 AWS DMS 目標
- 熟悉任務數 SharePlex 據複製

限制

- 資料庫大小限制為 64 TB
- 內部部署 Oracle 資料庫必須是企業版

產品版本

- 來源資料庫的甲骨文 8i 或 9i
- 中繼資料庫的甲骨文 10 克或 11 克
- PostgreSQL 9.6 或更高版本

架構

源, 技術, 堆棧

- 甲骨文 8i 或 9i 数据库
- 追求 SharePlex

目標技術堆疊

- Amazon RDS for PostgreSQL

來源與目標架構

工具

- AWS DMS — AWS Database Migration Service (AWS DMS) 可協助您快速安全地遷移資料庫。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。AWS DMS 可以在使用最廣泛的商業和開放原始碼資料庫之間移轉您的資料。
- AWS SCT — AWS Schema Conversion Tool (AWS SCT) 透過自動將來源資料庫結構描述和大多數資料庫程式碼物件 (包括視圖、存放程序和函數) 轉換為與目標資料庫相容的格式，進而使異質資料庫遷移可預測。無法自動轉換的物件會清楚標示，以便可以手動轉換它們以完成移轉。AWS SCT 也可以掃描您的應用程式原始程式碼是否有內嵌的 SQL 陳述式，並將其轉換為資料庫結構描述轉換專案的一部分。在此過程中，AWS SCT 會將舊版 Oracle 和 SQL Server 函數轉換為其 AWS 等效函數，以執行雲端原生程式碼優化，協助您在遷移資料庫時將應用程式現代化。結構描述轉換完成後，AWS SCT 可以使用內建的資料遷移代理程式，協助將資料從一系列資料倉儲遷移到 Amazon Redshift。
- 任務 SharePlex — Qu est SharePlex 是 Oracle 對 Oracle 的資料複製工具，可在停機時間降至最低且不會遺失資料的情況下移動資料。

史诗

建立 EC2 執行個體並安裝甲骨文

任務	描述	所需技能
為 Amazon EC2 設置網絡。	建立虛擬私有雲 (VPC)、子網路、網際網路閘道、路由表和安全群組。	AWS SysAdmin
建立新的 EC2 執行個體。	選取 EC2 執行個體的 Amazon 機器映像 (AMI)。選擇執行個體大小並設定執行個體詳細資訊：執行個體數量 (1)、上一步中的 VPC 和子網路、自動指派公用 IP 以及其他選項。新增儲存區、設定安全群組，以及啟動執行個體。出現提示時，建立並儲存 key pair 以供下一個步驟使用。	AWS SysAdmin

任務	描述	所需技能
在 EC2 實例上安裝甲骨文。	取得授權和所需的 Oracle 二進位檔案，並在 EC2 執行個體上安裝 Oracle 10g 或 11g。	DBA

在 EC2 執行個體 SharePlex 上設定並設定資料複寫

任務	描述	所需技能
設定 SharePlex。	建立 Amazon EC2 執行個體，並安裝與甲骨文 8i 或 9i 相容的 SharePlex 二進位檔案。	AWS SysAdmin、DBA
設定資料複製。	請遵循 SharePlex 最佳做法，設定從內部部署 Oracle 8i/9i 資料庫複寫至 Oracle 10g/11g 執行個體的資料複寫。	DBA

將甲骨文數據庫模式轉換為

任務	描述	所需技能
設定 AWS SCT。	創建一個新的報告，然後連接到甲骨文作為源和 PostgreSQL 作為目標。在專案設定中，開啟「SQL 指令碼」索引標籤，然後將目標 SQL 指令碼變更為「多個檔案」。	DBA
轉換 Oracle 資料庫結構描述。	在「動作」標籤中，選擇「產生報表」、「轉換結構描述」，然後選擇「另存為 SQL」	DBA

任務	描述	所需技能
修改 AWS SCT 產生的 SQL 指令碼。		DBA

建立和設定 Amazon RDS 資料庫執行個體

任務	描述	所需技能
建立 Amazon RDS 資料庫執行個體。	在 Amazon RDS 主控台中，建立新的 PostgreSQL 資料庫執行個體。	AWS SysAdmin、DBA
設定資料庫執行個體。	指定資料庫引擎版本、資料庫執行個體類別、異地同步備份部署、儲存類型和配置的儲存體。輸入資料庫執行個體識別碼、主要使用者名稱和主要密碼。	AWS SysAdmin、DBA
設定網路和安全性。	指定 VPC、子網路群組、公用存取性、可用區域偏好設定和安全性群組。	AWS SysAdmin、DBA
設定資料庫選項。	指定資料庫名稱、連接埠、參數群組、加密和主要金鑰。	AWS SysAdmin、DBA
設定備份。	指定備份保留期間、備份時段、開始時間、持續時間，以及是否要將標記複製到快照。	AWS SysAdmin、DBA
設定監視選項。	啟用或停用增強型監控和效能洞察。	AWS SysAdmin、DBA
設定維護選項。	指定 auto 次要版本升級、維護時段，以及開始日期、時間和持續時間。	AWS SysAdmin、DBA

任務	描述	所需技能
從 AWS SCT 執行移轉前指令碼。	在 Amazon RDS 執行個體上，執行下列指令碼：create_database.sql、create_sequence.sql、create_table.sql、create_view.sql 和完成。create_function.sql	AWS SysAdmin、DBA

使用 AWS DMS 遷移資料

任務	描述	所需技能
在 AWS DMS 中建立複寫執行個體。	填寫名稱、執行個體類別、VPC (與 EC2 執行個體相同)、異地同步備份和公共協助工具的欄位。在進階組態區段中，指定已配置的儲存體、子網路群組、可用區域、VPC 安全群組和 AWS Key Management Service (AWS KMS) 根金鑰。	AWS SysAdmin、DBA
建立來源資料庫端點。	指定端點名稱、類型、來源引擎 (Oracle)、伺服器名稱 (Amazon EC2 私有 DNS 名稱)、連接埠、SSL 模式、使用者名稱、密碼、SID、VPC (指定具有複寫執行個體的 VPC) 以及複寫執行個體。若要測試連線，請選擇 [執行測試]，然後建立端點。您也可以設定下列進階設定： maxFileSize 和「 numberDataType縮放」。	AWS SysAdmin、DBA
建立 AWS DMS 複寫任務。	指定工作名稱、複製執行個體、來源和目標端點，以及複	AWS SysAdmin、DBA

任務	描述	所需技能
	製執行個體。對於移轉類型，請選擇「移轉現有資料並複寫進行中的變更」。清除 [建立時啟動工作] 核取方塊。	
設定 AWS DMS 複寫任務設定。	對於目標資料表準備模式，請選擇「不執行任何動作」。完全載入完成後停止工作，以建立主索引鍵。指定受限或完整 LOB 模式，並啟用控制表。或者，您可以設定 CommitRate 進階設定。	DBA
設定表格對映。	在表格對應段落中，為移轉中包含的所有綱要中的所有表格建立「包括」規則，然後建立「排除」規則。新增三個轉換規則，將結構描述、資料表和資料行名稱轉換為小寫，並新增此特定移轉所需的任何其他規則。	DBA
開始工作。	啟動複寫工作。確保滿載正在運行。在主要 Oracle 資料庫上執行 ALTER 系統交換器記錄檔，以啟動作業。	DBA
從 AWS SCT 執行中間移轉指令碼。	在 Amazon RDS for PostgreSQL 中，執行以下指令碼：create_index.sql 和 create_constraint.sql。	DBA

任務	描述	所需技能
重新啟動工作以繼續變更資料擷取 (CDC)。	在適用 Amazon RDS for PostgreSQL 資料庫執行個體中，執行真空，然後重新啟動 AWS DMS 任務以套用快取的疾病控制中心變更。	DBA

切換至 PostgreSQL 資料庫

任務	描述	所需技能
檢查 AWS DMS 日誌和中繼資料表。	驗證任何錯誤並在需要時修復。	DBA
停止所有 Oracle 相依性。	關閉 Oracle 資料庫上的監聽器，然後執行 ALTER 系統交換器日誌檔案。如果 AWS DMS 任務沒有顯示任何活動，請停止該任務。	DBA
從 AWS SCT 執行移轉後指令碼。	在 Amazon RDS for PostgreSQL 中，執行以下指令碼：create_foreign_key_constraint.sql 和 create_triggers.sql。	DBA
完成任何其他 Amazon RDS for PostgreSQL 的步驟。	如果需要，增加序列以匹配 Oracle，運行真空和分析，並拍攝快照以實現合規性。	DBA
開啟與 Amazon RDS for PostgreSQL。	從 Amazon RDS for PostgreSQL 移除 AWS DMS 安全群組、新增生產安全群組，並將您的應用程式指向新的資料庫。	DBA

任務	描述	所需技能
清理 AWS DMS 資源。	移除端點、複寫任務、複寫執行個體和 EC2 執行個體。	SysAdmin, DBA

相關資源

- AWS DMS 說明文件
- AWS SCT 文件
- Amazon RDS for PostgreSQL 價
- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標
- 任務 SharePlex 文件

使用具體化視圖和 AWS DMS，從甲骨文 8i 或 9i 遷移到亞馬遜 RDS

由庫馬爾巴布 P G（AWS）和普利納斯·帕特爾（AWS）創建

環境：PoC 或試點	資料來源：甲骨文 8i 或 9i	目標：Amazon RDS for PostgreSQL 或 Aurora 兼容
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS; Amazon Aurora		

Summary

此模式說明如何將現場部署舊版 Oracle 8i 或 9i 資料庫遷移到 Amazon Relational Database Service 服務 (Amazon RDS)，適用於 PostgreSQL 或 Amazon Aurora PostgreSQL 相容版本。

AWS Database Migration Service (AWS DMS) 不支援 Oracle 8i 或 9i 做為來源，因此此模式使用與 AWS DMS 相容的中繼 Oracle 資料庫執行個體，例如甲骨文 10g 或 11g。它也會使用具體化視觀表功能，將資料從來源 Oracle 8i/9i 執行處理移轉至中繼 Oracle 10g/11g 執行處理。

AWS Schema Conversion Tool (AWS SCT) 會轉換資料庫結構描述，而 AWS DMS 會將資料遷移到目標 PostgreSQL 資料庫。

此模式可協助想要在最短的資料庫停機時間內從舊版 Oracle 資料庫移轉的使用者。在此實作中，停機時間會限制在建立或驗證目標資料庫上所有外部索引鍵、觸發程序和序列所需的時間長度。

該模式使用亞馬遜彈性運算雲端 (Amazon EC2) 執行個體，並安裝了 Oracle 10g/11g 資料庫，以協助 AWS DMS 串流資料。您可以暫時暫停從現場部署 Oracle 資料庫到中繼 Oracle 執行個體的串流複寫，讓 AWS DMS 能夠 catch 資料驗證或使用其他資料驗證工具。當 AWS DMS 完成移轉目前的變更時，PostgreSQL 資料庫執行個體和中繼 Oracle 資料庫將擁有相同的資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Oracle 8i 或 9i 資料庫
- 在現場部署資料中心和 AWS 之間設定 AWS Direct Connect

- 適用於 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式，可安裝在本機電腦或已安裝 AWS SCT 的 EC2 執行個體上
- 熟悉使用 Oracle 資料庫做為 AWS DMS 來源
- 熟悉使用 PostgreSQL 資料庫做為 AWS DMS 目標

限制

- 資料庫大小限制為 64 TB

產品版本

- 來源資料庫的甲骨文 8i 或 9i
- 中繼資料庫的甲骨文 10 克或 11 克
- PostgreSQL 10.17 或更高版本

架構

源, 技術, 堆棧

- 甲骨文 8i 或 9i 数据库

目標技術堆疊

- Amazon RDS for PostgreSQL SQL 或 Aurora 兼容

目標架構

工具

- AWS DMS 可協助您快速安全地遷移資料庫。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。AWS DMS 可以在使用最廣泛的商業和開放原始碼資料庫之間移轉您的資料。
- AWS SCT 會自動將來源資料庫結構描述和大多數資料庫程式碼物件 (包括檢視、預存程序和函數) 轉換為與目標資料庫相容的格式。無法自動轉換的物件會清楚標示，以便可以手動轉換它們以完成移轉。AWS SCT 也可以掃描您的應用程式原始程式碼是否有內嵌的 SQL 陳述式，並將其轉換為資料

庫結構描述轉換專案的一部分。在此過程中，AWS SCT 會將舊版 Oracle 和 SQL Server 函數轉換為其 AWS 等效函數，以執行雲端原生程式碼優化，協助您在遷移資料庫時將應用程式現代化。結構描述轉換完成後，AWS SCT 可以使用內建的資料遷移代理程式，協助將資料從一系列資料倉儲遷移到 Amazon Redshift。

最佳實務

如需重新整理具體化視觀表的最佳作法，請參閱下列 Oracle 文件：

- 重新整理具體化
- 快速重新整理具體化視觀

史诗

在 EC2 執行個體上安裝 Oracle 並建立具體化視圖

任務	描述	所需技能
設定 EC2 執行個體的網路。	建立虛擬私有雲 (VPC)、子網路、網際網路閘道、路由表和安全群組。	AWS SysAdmin
建立 EC2 執行個體。	選取 EC2 執行個體的 Amazon 機器映像 (AMI)。選擇執行個體大小並設定執行個體詳細資訊：執行個體數量 (1)、上一步中的 VPC 和子網路、自動指派公用 IP 以及其他選項。新增儲存區、設定安全群組，以及啟動執行個體。出現提示時，建立並儲存 key pair 以供下一個步驟使用。	AWS SysAdmin
在 EC2 執行個體上安裝甲骨文。	取得授權和所需的 Oracle 二進位檔案，並在 EC2 執行個體上安裝 Oracle 10g 或 11g。	DBA

任務	描述	所需技能
設定甲骨文網路。	在中修改或新增項目 `listener.ora` 以連線到內部部署來源 Oracle 8i/9i 資料庫，然後建立資料庫連結。	DBA
建立具體化視觀表。	識別要在來源 Oracle 8i/9i 資料庫中複製的資料庫物件，然後使用資料庫連結建立所有物件的具體化視觀表。	DBA
建置命令檔以依需要的間隔重新整理具體化視觀	開發和部署指令碼，以便在 Amazon EC2 Oracle 10g/11g 執行個體上按需要的間隔重新整理具體化視圖。使用累加式重新整理選項重新整理具體化視觀表	DBA

將甲骨文數據庫模式轉換為

任務	描述	所需技能
設定 AWS SCT。	創建一個新的報告，然後連接到甲骨文作為源和 PostgreSQL 作為目標。在專案設定中，開啟 [SQL 指令碼] 索引標籤。將目標 SQL 腳本更改為「多個文件」。(AWS SCT 不支援 Oracle 8i/9i 資料庫，因此您必須在中繼 Oracle 10g/11g 執行個體上還原僅結構描述的傾印，並將其用作 AWS SCT 的來源。)	DBA

任務	描述	所需技能
轉換 Oracle 資料庫結構描述。	在 [動作] 索引標籤上，選擇 [產生報告]、[轉換結構描述]，然後選取 [另存為	DBA
修改 SQL 指令碼。	根據最佳做法進行修改。例如，切換至適當的資料類型，並針對 Oracle 特定函數開發 PostgreSQL 對等項目。	DBA, 開發人員

建立和設定 Amazon RDS 資料庫執行個體以託管轉換後的資料庫

任務	描述	所需技能
建立 Amazon RDS 資料庫執行個體。	在 Amazon RDS 主控台中，建立新的 PostgreSQL 資料庫執行個體。	AWS SysAdmin、DBA
設定資料庫執行個體。	指定資料庫引擎版本、資料庫執行個體類別、異地同步備份部署、儲存類型和配置的儲存體。輸入資料庫執行個體識別碼、主要使用者名稱和主要密碼。	AWS SysAdmin、DBA
設定網路和安全性。	指定 VPC、子網路群組、公用存取性、可用區域偏好設定和安全性群組。	DBA, SysAdmin
設定資料庫選項。	指定資料庫名稱、連接埠、參數群組、加密和主要金鑰。	資料庫管理 (AWS) SysAdmin
設定備份。	指定備份保留期間、備份時段、開始時間、持續時間，以及是否要將標記複製到快照。	AWS SysAdmin、DBA

任務	描述	所需技能
設定監視選項。	啟用或停用增強型監控和效能洞察。	AWS SysAdmin、DBA
設定維護選項。	指定 auto 次要版本升級、維護時段，以及開始日期、時間和持續時間。	AWS SysAdmin、DBA
從 AWS SCT 執行移轉前指令碼。	在目標 Amazon RDS for PostgreSQL SQL 執行個體上，透過使用 AWS SCT 中的 SQL 指令碼和其他修改來建立資料庫結構描述。這些可能包括運行多個腳本，包括用戶創建，數據庫創建，模式創建，表，視圖，函數和其他代碼對象。	AWS SysAdmin、DBA

使用 AWS DMS 遷移資料

任務	描述	所需技能
在 AWS DMS 中建立複寫執行個體。	填寫名稱、執行個體類別、VPC (與 EC2 執行個體相同)、異地同步備份和公共協助工具的欄位。在進階組態區段中，指定已配置的儲存體、子網路群組、可用區域、VPC 安全群組和 AWS Key Management Service (AWS KMS) 金鑰。	AWS SysAdmin、DBA
建立來源資料庫端點。	指定端點名稱、類型、來源引擎 (Oracle)、伺服器名稱 (EC2 執行個體的私有 DNS 名稱)、連接埠、SSL 模式、使用者	AWS SysAdmin、DBA

任務	描述	所需技能
	名稱、密碼、SID、VPC (指定具有複寫執行個體的 VPC) 以及複寫執行個體。若要測試連線，請選擇 [執行測試]，然後建立端點。您也可以設定下列進階設定：maxFileSize和「numberDataType縮放」。	
將 AWS DMS Connect 到 Amazon RDS for PostgreSQL	如果您的 PostgreSQL 資料庫位於另一個 VPC 中，則為跨 VPC 的連線建立移轉安全群組。	AWS SysAdmin、DBA
建立目標資料庫端點。	指定端點名稱、類型、來源引擎 (PostgreSQL)、伺服器名稱 (Amazon RDS 端點)、連接埠、SSL 模式、使用者名稱、密碼、資料庫名稱、VPC (指定具有複寫執行個體的 VPC) 和複寫執行個體。若要測試連線，請選擇 [執行測試]，然後建立端點。您也可以設定下列進階設定：maxFileSize和「numberDataType縮放」。	AWS SysAdmin、DBA
建立 AWS DMS 複寫任務。	指定工作名稱、複製執行個體、來源和目標端點，以及複製執行個體。對於移轉類型，請選擇移轉現有資料並複寫進行中的變更。清除 [建立時啟動工作] 核取方塊。	AWS SysAdmin、DBA

任務	描述	所需技能
設定 AWS DMS 複寫任務設定。	針對目標資料表準備模式，選擇「不執行任何動 完全負載完成後停止工作 (以建立主索引鍵)。指定受限或完整 LOB 模式，並啟用控制表。或者，您可以設定CommitRate進階設定。	DBA
設定表格對映。	在「表格對應」段落中，為移轉中包含的所有綱要中的所有表格建立「包括」規則，然後建立「排除」規則。新增三個轉換規則，將結構描述、資料表和資料行名稱轉換為小寫，並新增此特定移轉所需的任何其他規則。	DBA
開始工作。	啟動複寫工作。確保滿載正在運行。`ALTER SYSTEM SWITCH LOGFILE`在主要 Oracle 資料庫上執行以啟動作業。	DBA
從 AWS SCT 執行中間移轉指令碼。	在 Amazon RDS for PostgreSQL 中，執行下列指令碼：`create_index.sql` 和 `create_constraint.sql` (如果最初未建立完整的結構描述)。	DBA
繼續工作以繼續變更資料擷取 (CDC)。	`VACUUM`在適用 Amazon RDS for PostgreSQL 的資料庫執行個體上執行，然後重新啟動 AWS DMS 任務以套用快取的 CDC 變更。	DBA

切換至 PostgreSQL 資料庫

任務	描述	所需技能
檢查 AWS DMS 日誌和驗證表格。	檢查並修正任何複寫或驗證錯誤。	DBA
停止使用內部部署 Oracle 資料庫及其相依性。	停止所有 Oracle 相依性、關閉 Oracle 資料庫上的監聽器，然後執行`ALTER SYSTEM SWITCH LOGFILE`。如果 AWS DMS 任務沒有顯示任何活動，請停止該任務。	DBA
從 AWS SCT 執行移轉後指令碼。	在 Amazon RDS for PostgreSQL 行下列指令碼：`create_foreign_key_constraint.sql` and `create_triggers.sql` 確保序列是最新的。	DBA
完成其他 Amazon RDS for PostgreSQL 的步驟。	如果需要，請增加序列以符合 Oracle，執行`VACUUM`和`ANALYZE`，並建立快照以符合性。	DBA
開啟與 Amazon RDS for PostgreSQL。	從 Amazon RDS for PostgreSQL 移除 AWS DMS 安全群組、新增生產安全群組，並將您的應用程式指向新的資料庫。	DBA
清理 AWS DMS 物件。	移除端點、複寫任務、複寫執行個體和 EC2 執行個體。	SysAdmin, DBA

相關資源

- AWS DMS 說明文件

- AWS SCT 文件
- Amazon RDS for PostgreSQL 價
- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標

使用 AWS DMS 和 AWS SCT 從亞馬遜上的甲骨文遷移到 Amazon RDS for MySQL

由阿尼爾·庫納帕雷迪（AWS）和哈沙德戈希爾創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for MySQL
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上管理 Oracle 資料庫需要資源，而且成本高昂。將這些資料庫移至適用於 MySQL 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS)，可將整體 IT 預算最佳化，以簡化您的工作。Amazon RDS for MySQL 也提供異地同步備份、可擴展性和自動備份等功能。

此模式會引導您完成將 Amazon EC2 上的來源 Oracle 資料庫遷移到目標 Amazon RDS for MySQL 資料庫執行個體的過程。它使用 AWS Database Migration Service (AWS DMS) 來遷移資料，而 AWS 結構描述轉換工具 (AWS SCT) 將來源資料庫結構描述和物件轉換為與 Amazon RDS for MySQL 相容的格式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 執行處理和監聽器服務的來源資料庫，在「存檔日誌」模式下執行
- 目標 Amazon RDS for MySQL 適用於 MySQL 資料庫，具有足夠的儲存空間進行資料移轉

限制

- AWS DMS 不會在目標資料庫上建立結構描述；您必須這麼做。目標的綱要名稱必須已存在。來源結構描述中的表格會匯入使用者/結構描述，AWS DMS 用來連接到目標執行個體。如果您必須遷移多個結構描述，即必須建立多項複寫任務。

產品版本

- 版本 10.2 及更新版本、11 克和最大至 12.2 和 18c 的所有甲骨文資料庫版本。如需支援版本的最新清單，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源和使用與 MySQL 相容的資料庫做為 AWS DMS 的目標。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。如需 AWS SCT 支援的 Oracle 資料庫版本的相關資訊，請參閱 AWS SCT 文件。
- AWS 資料管理系統支援的版本 5.5、5.6 和 5.7 版的 MySQL。

架構

源, 技術, 堆棧

- EC2 執行個體上的甲骨文資料庫

目標技術堆疊

- Amazon RDS for MySQL 資料庫執行個體

資料移轉架構

來源與目標架構

工具

- AWS DMS-AWS Database Migration Service (AWS DMS) 是一種網路服務，可用來將資料從現場部署的資料庫、Amazon RDS 資料庫執行個體或 EC2 執行個體的資料庫遷移到 AWS 服務上的資料庫，例如 Amazon RDS for MySQL 或 EC2 執行個體。您也可以將資料庫從 AWS 服務遷移到現場部署資料庫。您可以在異質或同質資料庫引擎之間移轉資料。
- AWS SCT-AWS Schema Conversion Tool (AWS SCT) 可將來源資料庫結構描述和大部分資料庫程式碼物件 (包括檢視、預存程序和函數) 自動轉換為與目標資料庫相容的格式，進而使異質資料庫遷移可預測。使用 AWS SCT 轉換資料庫結構描述和程式碼物件後，您可以使用 AWS DMS 將資料從來源資料庫遷移到目標資料庫，以完成遷移專案。

史诗

規劃移轉

任務	描述	所需技能
識別來源和目標資料庫版本和引擎。		DBA/開發人員
識別 DMS 複製執行個體。		DBA/開發人員
識別儲存需求，例如儲存類型和容量。		DBA/開發人員
識別網路需求，例如延遲和頻寬。		DBA/開發人員
識別來源和目標伺服器執行處理的硬體需求 (根據 Oracle 相容性清單和容量需求)。		DBA/開發人員
識別來源和目標資料庫的網路存取安全性需求。		DBA/開發人員
安裝 AWS SCT 和甲骨文驅動程式。		DBA/開發人員
決定備份策略。		DBA/開發人員
決定可用性需求。		DBA/開發人員
識別應用程式遷移和切換策略。		DBA/開發人員
根據容量、儲存和網路功能，選擇適當的資料庫執行個體類型。		DBA/開發人員

設定環境

任務	描述	所需技能
建立 Virtual Private Cloud (VPC) 來源、目標和複寫執行個體應位於相同的 VPC 中。將它們放在相同的可用區域中也很好。		開發人員
建立資料庫存取所需的安全性群組。		開發人員
產生並設定 key pair。		開發人員
設定子網路、可用區域和 CIDR 區塊。		開發人員

設定來源：EC2 執行個體上的 Oracle 資料庫

任務	描述	所需技能
在具有必要使用者和角色的 Amazon EC2 上安裝 Oracle 資料庫。		DBA
執行下一欄中的三個步驟，以便從 EC2 執行個體外部存取 Oracle。	1. 將本機主機變更`tnsnames`為 Amazon EC2 公有 DNS。 2. 將本機主機變更`listener`為 Amazon EC2 公有 DNS。 3. 停止並重新啟動監聽器。	DBA
當 Amazon EC2 重新啟動時，公共 DNS 會發生變化。確保在「tnsnames」和「監聽		DBA/開發人員

任務	描述	所需技能
器」中更新 Amazon EC2 公共 DNS，或使用彈性 IP 地址。		
設定 EC2 執行個體安全群組，以便複寫執行個體和必要的用戶端可以存取來源資料庫。		DBA/開發人員

設定目標：適用於 MySQL 的 Amazon RDS for MySQL

任務	描述	所需技能
設定並啟動 Amazon RDS for MySQL 的資料庫執行個體。		開發人員
在 Amazon RDS for MySQL 版 MySQL 資料庫執行個體中建立必要的表格空間。		DBA
設定安全群組，讓複寫執行個體和必要的用戶端可以存取目標資料庫。		開發人員

設定 AWS SCT 並在目標資料庫中建立結構描述

任務	描述	所需技能
安裝 AWS SCT 和甲骨文驅動程式。		開發人員
輸入適當的參數並連接到來源和目標。		開發人員
產生結構描述轉換報告。		開發人員

任務	描述	所需技能
視需要更正程式碼和結構描述，尤其是表格空間和引號，並在目標資料庫上執行。		開發人員
移轉資料之前，請先驗證來源與目標上的結構描述。		開發人員

使用 AWS DMS 遷移資料

任務	描述	所需技能
對於全負載和變更資料擷取 (CDC) 或只是 CDC，您必須設定額外的連線屬性。		開發人員
在 AWS DMS 來源 Oracle 資料庫定義中指定的使用者必須獲得所有必要的權限。如需完整清單，請參閱 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.Oracle.html#CHAP_Source.Oracle.Self-Managed。		DBA/開發人員
啟用來源資料庫中的補充記錄日誌。		DBA/開發人員
對於全載和變更資料擷取 (CDC) 或只是 CDC，請在來源資料庫中啟用「存檔日誌」模式。		DBA
建立來源和目標端點，並測試連線。		開發人員

任務	描述	所需技能
端點連線成功後，請建立複寫工作。		開發人員
在工作中選取僅 CDC (或) 滿載加上 CDC，分別擷取僅連續複寫 (或) 滿載加上進行中變更的變更。		開發人員
執行複寫任務並監控 Amazon CloudWatch 日誌。		開發人員
驗證來源和目標資料庫中的資料。		開發人員

遷移您的應用程序並切除

任務	描述	所需技能
遵循應用程式遷移策略的步驟。		DBA、開發人員、應用程式擁有者
請遵循應用程式切換/切換策略的步驟。		DBA、開發人員、應用程式擁有者

關閉專案

任務	描述	所需技能
驗證來源資料庫與目標資料庫中的結構描述和資料。		DBA/開發人員
收集移轉時間的指標、手動與工具的百分比、節省成本等。		DBA/ 開發人員/AppOwner
審核專案文件和人工因素。		DBA/ 開發人員/AppOwner

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA/開發人員
關閉專案並提供意見反應。		DBA/ 開發人員/AppOwner

相關資源

- AWS DMS 說明文件
- AWS 管理系統網站
- AWS DMS 部落格文章
- 將 Oracle Database 遷移到 AWS 的策略
- Amazon RDS for Oracle 常見問題
- Oracle 常見問題
- Amazon EC2
- Amazon EC2 常見問
- 在雲端運算環境中授權 Oracle 軟體

使用 AWS DMS 從甲骨文遷移到 Amazon DocumentDB

R 型：重新建築	來源：數據庫：關係	目標：Amazon DocumentDB
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作量：甲骨文	AWS 服務：Amazon DocumentDB	

Summary

此模式提供使用 AWS Database Migration Service (AWS DMS) 將甲骨文資料庫移轉至 Amazon 文件資料庫 (具有 MongoDB 相容性) 資料庫的指導。此方法可套用至現場部署 Oracle 來源資料庫，以及適用於 Oracle 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS)。此模式使用 Amazon RDS Oracle 資料庫來源執行個體做為範例。

Amazon DocumentDB 資料庫 (與 MongoDB 相容性) 是一種完全受管、與 MongoDB 相容的文件資料庫服務，可讓您輕鬆地儲存、查詢和索引 JSON 資料。

此模式的使用案例是將 Oracle 資料庫表格 one-to-one 複寫到 Amazon DocumentDB 集合。該模式使用 AWS DMS 複寫任務來讀取 Oracle 資料庫的表格結構、在 Amazon DocumentDB 中建立對應的集合，以及執行全負載遷移。您可以查看和查詢您的數據在 Amazon DocumentDB，就像你會在 MongoDB 中。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 熟悉 Oracle 資料庫的使用
- 熟悉使用 Amazon DocumentDB
- 對於 Oracle 用戶，請選擇任何表格權限
- 對於 Amazon DocumentDB 的使用，轉儲數據所需的特權

限制

使用 Amazon DocumentDB 做為 AWS DMS 的目標時，會有下列限制：

- 在 Amazon DocumentDB 中，集合名稱不能包含金錢符號 ($)。此外，資料庫名稱不能包含任何 Unicode 字元。
- AWS DMS 不支援將多個來源資料表合併為單一 Amazon DocumentDB 集合。
- 當 AWS DMS 從沒有主索引鍵的來源資料表處理變更時，該資料表中的任何大型二進位物件 (LOB) 資料行都會遭到忽略。
- 如果啟用「變更資料表」選項，且 AWS DMS 遇到名為「_id」的來源資料行，則該欄會在變更表格中顯示為「__id」(兩個底線)。
- 如果您選擇 Oracle 作為來源端點，則 Oracle 來源必須啟用完整的補充記錄日誌。否則，如果來源中有未變更的資料行，資料會以空值的形式載入 Amazon DocumentDB。

產品版本

- Amazon RDS for Oracle 文 11.2.0.3 或更高版本
- AWS DMS 版本 3.1.3 或更新版本 (如需最新版本資訊，請參閱 AWS DMS 文件中的使用亞馬遜文件資料庫做為 AWS DMS 的目標)

架構

源, 技術, 堆棧

- Amazon RDS for Oracle 數據庫

目標技術堆疊

- Amazon DocumentDB

來源與目標架構

工具

- AWS DMS — AWS Database Migration Service (AWS DMS) 是一種 Web 服務，可用來將資料從來源資料存放區遷移到目標資料存放區。AWS DMS 使用者指南指定支援與 AWS DMS 搭配使用的

Oracle 來源資料庫版本和版本。如需與此模式相關的其他資訊，請參閱使用 Amazon DocumentDB 做為 AWS DMS 的目標。

- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您的 Amazon DocumentDB 叢集應該在預設虛擬私有雲端 (VPC) 中執行。若要與 Amazon 文件資料庫叢集互動，您必須在建立 Amazon DocumentDB 叢集的相同 AWS 區域中，將 EC2 執行個體啟動到預設 VPC。如需詳細資訊，請參閱 Amazon 文件中的啟動 Amazon EC2 執行個體。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		AWS 管理員
選擇適當的執行個體類型 (容量、儲存空間功能、網路功能)。		AWS 管理員
識別來源和目標資料庫的網路/主機存取安全性需求。		AWS 管理員
建立來源和目標資料庫的輸出安全群組。		AWS 管理員
為 Amazon DocumentDB 創建和設定 EC2 實例。		AWS 管理員

設定基礎結

任務	描述	所需技能
建立 VPC 和子網路。		AWS 管理員
建立安全性群組和網路存取控制清單 (ACL)。		AWS 管理員

任務	描述	所需技能
設定並啟動來源 Amazon RDS for Oracle 執行個體。		AWS 管理員
設定並啟動 Amazon DocumentDB 執行個體。		AWS 管理員

準備來源資料庫

任務	描述	所需技能
確認可以使用連線詳細資訊連線 Oracle 資料庫。		AWS 管理員
確認 Oracle 使用者具有「選取任何表格」權限。		AWS 管理員

準備目標資料庫

任務	描述	所需技能
選擇適當的執行個體類別和執行個體數目，以建立 Amazon DocumentDB 叢集。		AWS 管理員

設定 Amazon EC2

任務	描述	所需技能
設定 EC2 執行個體。	若要與 Amazon 文件資料庫叢集互動，您必須在建立 Amazon DocumentDB 叢集的相同 AWS 區域中，將 EC2 執行個體啟動到預設 VPC。設	AWS 管理員

任務	描述	所需技能
	定 EC2 執行個體的 AWS 區域、VPC、可用區域和子網路。	
設定 key pair。	公開/私密 key pair 可讓您在 EC2 執行個體啟動後安全地連線。	AWS 管理員
設定防禦主機 CIDR 範圍 (選擇性)。	設定允許外部安全殼層 (SSH) 存取防禦主機執行個體的 CIDR IP 範圍。	AWS 管理員

移轉資料 — 滿載

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。		AWS 管理員
建立來源端點和目標端點。		AWS 管理員
建立完整負載的 AWS DMS 複寫任務。		AWS 管理員

測試遷移

任務	描述	所需技能
透過 EC2 執行個體 Connect 線到 Amazon DocumentDB 叢集。		AWS 管理員
Connect 到使用蒙戈殼集群。	如需指示，請參閱 < 參考和說明 > 一節中的 Amazon 文件資料庫連結。	AWS 管理員

任務	描述	所需技能
驗證移轉的結果。		AWS 管理員

相關資源

- AWS DMS 的運作方式
- 遷移到 Amazon DocumentDB
- 使用 Amazon DocumentDB 作為 AWS DMS 的目標
- Amazon DocumentDB 概述
- 訪問和使用您的 Amazon DocumentDB 集群使用蒙戈殼
- 使用離線方法從 MongoDB 遷移到 Amazon DocumentDB（博客文章）
- 如何使用 Amazon DocumentDB (與 MongoDB 相容性) 大規模建置和管理應用程式 (部落格文章)

使用 AWS DMS 和 AWS SCT 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS

創建者：維拉賈納魯格蘭希（AWS）和維諾德庫馬爾（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for MariaDB
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式會引導您完成將 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 Oracle 資料庫遷移到適用於 MariaDB 資料庫執行個體的 Amazon Relational Database Service 服務 (Amazon RDS) 的步驟。該模式使用 AWS 資料遷移服務 (AWS DMS) 進行資料遷移，並使用 AWS Schema Conversion Tool (AWS SCT) 進行結構描述轉換。

在 EC2 執行個體上管理 Oracle 資料庫需要更多資源，而且比在 Amazon RDS 上使用資料庫更昂貴。Amazon RDS 可讓您輕鬆地在雲端中設定、操作和擴展關聯式資料庫。Amazon RDS 提供符合成本效益且可調整大小的容量，同時自動執行硬體佈建、資料庫設定、修補和備份等耗時的管理任務。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 啟動並執行執行處理和監聽器服務的來源 Oracle 資料庫。該數据庫應處于「存檔日志」模式下。
- 熟悉使用 Oracle 資料庫做為 AWS DMS 的來源。
- 熟悉使用甲骨文作為 AWS SCT 的來源。

限制

- 資料庫大小限制：64 TB

產品版本

- 版本 10.2 及更新版本、11 克和最大至 12.2 和 18c 的所有甲骨文資料庫版本。如需支援版本的最新清單，請參閱 AWS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源和 AWS SCT 版本表格。
- Amazon RDS 支援 MariaDB 伺服器社群伺服器版本 10.3、10.4、10.5 和 10.6 版。如需支援版本的最新清單，請參閱 Amazon RDS 文件。

架構

源, 技術, 堆棧

- EC2 執行個體上的甲骨文資料庫

目標技術堆疊

- Amazon RDS for MariaDB

資料移轉架構

目標架構

工具

- AWS Schema Conversion Tool (AWS SCT) 可將來源資料庫結構描述和大多數資料庫程式碼物件 (包括檢視、預存程序和函數) 自動轉換為與目標資料庫相容的格式，讓異質資料庫遷移可預測。使用 AWS SCT 轉換資料庫結構描述和程式碼物件後，您可以使用 AWS DMS 將資料從來源資料庫遷移到目標資料庫，以完成遷移專案。如需詳細資訊，請參閱 AWS SCT 文件中的使用 Oracle 做為 AWS SCT 的來源。
- AWS Database Migration Service (AWS DMS) 可協助您快速安全地將資料庫遷移到 AWS。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。AWS DMS 可以在使用最廣泛的商業和開放原始碼資料庫之間移轉您的資料。AWS DMS 支援甲骨文到甲骨文等同質遷移，以及在不同的資料庫平台 (例如甲骨文或 Microsoft SQL 伺服器到 Amazon Aurora) 之間進行異質遷移。若要進一步了解如何遷移 Oracle 資料庫，請參閱 AWS DMS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源。

史诗

規劃移轉

任務	描述	所需技能
識別版本和資料庫引擎。	識別來源和目標資料庫版本和引擎。	DBA, 開發人員
識別複製執行個體。	識別 AWS DMS 複寫執行個體。	DBA, 開發人員
識別儲存需求。	識別儲存類型和容量。	DBA, 開發人員
識別網路需求。	識別網路延遲和頻寬。	DBA, 開發人員
識別硬體需求。	識別來源和目標伺服器執行處理的硬體需求 (根據 Oracle 相容性清單和容量需求)。	DBA, 開發人員
識別安全性需求。	識別來源和目標資料庫的網路存取安全性需求。	DBA, 開發人員
安裝驅動程式。	安裝最新的 AWS SCT 和甲骨文驅動程式。	DBA, 開發人員
決定備份策略。		DBA, 開發人員
決定可用性需求。		DBA, 開發人員
選擇應用程式遷移/切換策略。		DBA, 開發人員
選取 執行個體類型。	根據容量、儲存空間和網路功能選取適當的執行個體類型。	DBA, 開發人員

設定環境

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)	來源、目標和複寫執行個體應位於相同的 VPC 且位於相同的可用區域 (建議使用)。	開發人員
建立安全性群組。	建立資料庫存取所需的安全性群組。	開發人員
產生金鑰對。	產生並設定 key pair。	開發人員
設定其他資源。	設定子網路、可用區域和 CIDR 區塊。	開發人員

設定來源

任務	描述	所需技能
啟動 EC2 執行個體。	如需指示，請參閱 Amazon EC2 文件。	開發人員
安裝 Oracle 資料庫。	使用必要的使用者和角色，在 EC2 執行個體上安裝 Oracle 資料庫。	DBA
請遵循任務說明中的步驟，從 EC2 執行個體外部存取 Oracle。	1. 將本機主機變更`tnsnames`為 Amazon EC2 公有 DNS。 2. 將本機主機變更`listener`為 Amazon EC2 公有 DNS。 3. 停止並重新啟動監聽器。	DBA
更新 Amazon EC2 公共 DNS。	EC2 執行個體重新啟動後，公用 DNS 會變更。確保在和中更新 Amazon EC2 公共 DNS	DBA, 開發人員

任務	描述	所需技能
	`tnsnameslistener`，或使用彈性 IP 地址。	
設定 EC2 執行個體安全群組。	設定 EC2 執行個體安全群組，讓複寫執行個體和所需的用戶端可以存取來源資料庫。	DBA, 開發人員

設定適用於馬里亞 DB 環境的目標亞馬遜 RDS

任務	描述	所需技能
啟動 RDS 資料庫執行個體。	設定並啟動適用於 MariaDB 資料庫執行個體的 Amazon RDS。	開發人員
建立表格空間。	在 Amazon RDS MariaDB 資料庫中建立任何必要的表格空間。	DBA
設定安全性群組。	設定安全群組，讓複寫執行個體和必要的用戶端可以存取目標資料庫。	開發人員

設定 AWS SCT

任務	描述	所需技能
安裝驅動程式。	安裝最新的 AWS SCT 和甲骨文驅動程式。	開發人員
連接。	輸入適當的參數，然後連接到來源和目標。	開發人員
產生結構描述轉換報告。	產生 AWS SCT 結構描述轉換報告。	開發人員

任務	描述	所需技能
視需要更正程式碼和結構描述。	對程式碼和結構描述 (特別是表格空間和引號) 進行任何必要的更正。	DBA, 開發人員
驗證結構描述。	載入資料之前，請先驗證來源與目標上的結構描述。	開發人員

使用 AWS DMS 遷移資料

任務	描述	所需技能
設定連線屬性。	對於全負載和變更資料擷取 (CDC)，或僅針對 CDC，請設定額外的連線屬性。如需詳細資訊，請參閱 Amazon RDS 文件。	開發人員
啟用補充記錄日誌。	啟用來源資料庫的補充記錄日誌。	DBA, 開發人員
啟用封存記錄模式。	對於完整負載和 CDC (或只適用於 CDC)，請在來源資料庫上啟用封存記錄模式。	DBA
建立和測試端點。	建立來源和目標端點並測試連線。如需詳細資訊，請參閱 Amazon DMS 文件。	開發人員
建立複製工作。	端點連線成功後，請建立複寫工作。如需詳細資訊，請參閱 Amazon DMS 文件。	開發人員
選擇複製類型。	在工作中選擇 [僅限 CDC] 或 [全負載加上 CDC]，以僅擷取	開發人員

任務	描述	所需技能
	連續複寫的變更，或分別擷取完整負載和進行中的變更。	
啟動並監視工作。	啟動複寫任務並監控 Amazon CloudWatch 日誌。如需詳細資訊，請參閱 Amazon DMS 文件。	開發人員
驗證資料。	驗證來源和目標資料庫中的資料。	開發人員

移轉應用程式並切換至目標資料庫

任務	描述	所需技能
遵循選擇的應用程式遷移策略。		DBA、應用程式擁有者、開發人員
遵循所選的應用程式切換/切換策略。		DBA、應用程式擁有者、開發人員

關閉專案

任務	描述	所需技能
驗證結構描述和資料。	在專案關閉之前，請確保在來源與目標中成功驗證結構描述和資料。	DBA, 開發人員
收集指標。	收集移轉時間的指標、手動與工具作業的百分比、節省成本，以及類似準則。	DBA、應用程式擁有者、開發人員
檢閱文件。	審核專案文件和人工因素。	DBA、應用程式擁有者、開發人員

任務	描述	所需技能
關閉資源。	關閉臨時 AWS 資源。	DBA, 開發人員
關閉專案。	關閉移轉專案並提供任何意見反應。	DBA、應用程式擁有者、開發人員

相關資源

- MariaDB Amazon RDS 概述
- 亞馬遜 RDS 系列產品詳細資訊
- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 將甲骨文資料庫遷移到 AWS 的策略
- 在雲端運算環境中授權 Oracle 軟體
- Amazon RDS for Oracle 常見問題
- AWS DMS 概觀
- AWS DMS 部落格文章
- Amazon EC2 概述
- Amazon EC2 常見問
- AWS SCT 文件

使用 AWS DMS 和 AWS SCT 將現場部署甲骨文資料庫遷移到適用於 MySQL 的 Amazon RDS for MySQL

R 型：重新建築	來源：數據庫：關係	目標：適用於 MySQL 的 Amazon RDS for MySQL
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作量：甲骨文	AWS 服務：Amazon RDS	

Summary

此模式會逐步引導您將現場部署 Oracle 資料庫遷移到適用於 MySQL 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS)。它使用 AWS Database Migration Service (AWS DMS) 來遷移資料，而 AWS 結構描述轉換工具 (AWS SCT) 將來源資料庫結構描述和物件轉換為與 Amazon RDS for MySQL 相容的格式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Oracle 資料庫

限制

- 資料庫大小限制：64 TB

產品版本

- 版本 11g (11.2.0.3.v1 及更新版本) 以及最高至 12.2 和 18c 的所有 Oracle 資料庫版本。如需支援版本的最新清單，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。如需 AWS SCT 支援的 Oracle 資料庫版本的相關資訊，請參閱 AWS SCT 文件。
- AWS 資料管理系統目前支援 MySQL 版本 5.5、5.6 和 5.7 版。如需支援版本的最新清單，請參閱 AWS 文件中的使用與 MySQL 相容的資料庫做為 AWS DMS 的目標。

架構

源, 技術, 堆棧

- 內部部署甲骨文

目標技術堆疊

- Amazon RDS for MySQL 資料庫執行個體

資料移轉架構

工具

- AWS DMS-AWS Database Migration 服務 (AWS DMS) 可協助您遷移關聯式資料庫、資料倉儲、NoSQL 資料庫和其他類型的資料存放區。您可以使用 AWS DMS 將資料遷移至 AWS 雲端，可在現場部署執行個體 (透過 AWS 雲端設定) 或在雲端和現場部署設定之間進行。
- AWS SCT-AWS S chema Conversion Tool (AWS SCT) 可用來將資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎。該工具轉換的自定義代碼包括視圖，存儲過程和函數。該工具無法自動轉換的任何代碼都會清楚標記，以便您可以自行轉換。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA, SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin

任務	描述	所需技能
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA, SysAdmin
識別應用程式移轉策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結構

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 Amazon RDS 資料庫執行個體。		DBA, SysAdmin

移轉資料

任務	描述	所需技能
使用 AWS SCT 遷移資料庫結構描述。		DBA
使用 AWS DMS 遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
使用 AWS SCT 分析和轉換應用程式程式碼中的 SQL 程式碼。	如需詳細資訊，請參閱 https://docs.aws.amazon.com/SchemaConversion 工具/最新/使用者指南/章節轉換。	應用所有者
遵循應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, SysAdmin
審核並驗證專案文件。		DBA, SysAdmin
收集移轉時間的指標、手動與工具的百分比、節省成本等。		DBA, SysAdmin
關閉專案並提供意見反應。		

相關資源

參考

- AWS DMS 說明文件

- AWS SCT 文件
- Amazon RDS 定價

教程和視頻

- 開始使用 AWS DMS
- Amazon RDS 入門
- AWS DMS (影片)
- Amazon RDS (視頻)

使用甲骨文旁觀者和 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS

由凱迪·莫蒂卡（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：亞馬遜 RDS 後服務/ Amazon Aurora
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式說明如何在最短的停機時間內將現場部署 Oracle 資料庫遷移至下列任一 PostgreSQL 相容 AWS 資料庫服務：

- Amazon Relational Database Service 服務 (Amazon RDS)
- Amazon Aurora PostgreSQL-Compatible Edition

該解決方案使用 AWS Database Migration Service (AWS DMS) 來遷移資料、AWS Schema Conversion Tool (AWS SCT) 轉換資料庫結構描述，以及 Oracle 旁觀者資料庫來協助管理遷移。在此實作中，停機時間會限制在建立或驗證資料庫上所有外部索引鍵所需的長時間。

此解決方案也使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配 Oracle 旁觀者資料庫，協助透過 AWS DMS 控制資料串流。您可以暫時暫停從現場部署 Oracle 資料庫到 Oracle 旁觀者的串流複寫，以啟用 AWS DMS 以 catch 資料驗證，或使用其他資料驗證工具。AWS DMS 完成移轉目前變更 Amazon RDS for PostgreSQL 資料庫執行個體或 Aurora PostgreSQL 相容的資料庫執行個體和旁觀者資料庫將擁有相同的資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中已設定作用中資料保全待命資料庫的來源 Oracle 資料庫
- AWS Direct Connect 在現場部署資料中心和 AWS Secrets Manager 之間設定，用於存放資料庫機密

- 適用於 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式，可安裝在本機電腦或已安裝 AWS SCT 的 EC2 執行個體上
- 熟悉使用 Oracle 資料庫做為 AWS DMS 的來源
- 熟悉使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標

限制

- 資料庫大小限制：64 TB

產品版本

- AWS DMS 支援 10.2 及更新版本 (適用於版本 10.x)、11 克以及最高 12.2、18c 和 19 c 的所有甲骨文資料庫版本。如需支援版本的最新清單，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。如需 AWS SCT 支援的 Oracle 資料庫版本的相關資訊，請參閱 AWS SCT 文件。
- AWS DMS 支援 PostgreSQL 9.4 及更新版本 (適用於版本 9.x)、10.x、11.x、12 倍和 13.x 版。如需最新資訊，請參閱 AWS 文件中的使用 PostgreSQL 資料庫做為 AWS DMS 的目標。

架構

源, 技術, 堆棧

- 內部部署 Oracle 資料庫
- 持有 Oracle 資料庫旁觀者的 EC2 執行個體

目標技術堆疊

- Amazon RDS for PostgreSQL PostgreSQL 或 Aurora 執行個體

目標架構

下圖顯示使用 AWS DMS 和甲骨文旁觀者將 Oracle 資料庫遷移到與 PostgreSQL 相容的 AWS 資料庫的工作流程範例：

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。
- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。

史诗

將甲骨文數據庫模式轉換為 PostgreSQL

任務	描述	所需技能
設定 AWS SCT。	創建一個新的報告，並連接到甲骨文作為源和 PostgreSQL 作為目標。在 [專案設定] 中，移至 [SQL 指令碼] 索引標籤。將「目標 SQL 命令檔」變更為「多個檔案」。這些文件將在以後使用並命名如下： • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql • create_function.sql	DBA
轉換 Oracle 數據庫模式。	在「動作」標籤中，選擇「產生報告」。然後，選擇「轉換結構描述」並選擇「另存為 SQL」。	DBA
修改指令碼。	例如，如果來源結構描述中的數字已在 PostgreSQL 中轉換為數字格式，但您想要改用	DBA

任務	描述	所需技能
	BIGINT 來取得更好的效能，您可能會想要修改指令碼。	

建立和設定 Amazon RDS 資料庫執行個體

任務	描述	所需技能
建立 Amazon RDS 資料庫執行個體。	在正確的 AWS 區域中，建立新的 PostgreSQL 資料庫執行個體。如需詳細資訊，請參閱 Amazon RDS 說明文件中的建立 PostgreSQL 資料庫執行個體和連線到 PostgreSQL 資料庫執行個體上的資料庫。	AWS SysAdmin、DBA
設定資料庫執行個體規格	指定資料庫引擎版本、資料庫執行個體類別、異地同步備份部署、儲存類型和配置的儲存。輸入資料庫執行個體識別碼、主要使用者名稱和主要密碼。	AWS SysAdmin、DBA
設定網路和安全性。	指定虛擬私人雲端 (VPC)、子網路群組、公用存取性、可用區域偏好設定和安全性群組。	DBA, SysAdmin
設定資料庫選項。	指定資料庫名稱、連接埠、參數群組、加密和 KMS 金鑰。	AWS SysAdmin、DBA
設定備份。	指定備份保留期間、備份時段、開始時間、持續時間，以及是否要將標記複製到快照。	AWS SysAdmin、DBA
設定監視選項。	啟用或停用增強型監控和效能洞察。	AWS SysAdmin、DBA

任務	描述	所需技能
設定維護選項。	指定 auto 次要版本升級、維護時段，以及開始日期、時間和持續時間。	AWS SysAdmin、DBA
從 AWS SCT 執行移轉前指令碼。	在 Amazon RDS 執行個體上，執行 AWS SCT 產生的下列指令碼： • create_database.sql • create_sequence.sql • create_table.sql • create_view.sql • create_function.sql	AWS SysAdmin、DBA

在 Amazon EC2 中配置甲骨文旁觀者

任務	描述	所需技能
為 Amazon EC2 設置網絡。	建立新的 VPC、子網路、網際網路閘道、路由表和安全群組。	AWS SysAdmin
建立 EC2 執行個體。	在適當的 AWS 區域中，建立新的 EC2 執行個體。選取 Amazon Machine Image (AMI)、選擇執行個體大小，然後設定執行個體詳細資訊：執行個體數目 (1)、您在上一個任務中建立的 VPC 和子網路、自動指派公用 IP 以及其他選項。新增儲存區、設定安全群組，然後啟動。出現提示時，建立並儲存 key pair 以供下一個步驟使用。	AWS SysAdmin

任務	描述	所需技能
將 Oracle 來源資料庫 Connect 至 EC2 執行個體。	IPv4 的公共 IP 地址和 DNS 複製到一個文本文件，並通過使用 SSH 連接，如下所示:ssh-我「你的文件. address-or-public	AWS SysAdmin
在 Amazon EC2 中為旁觀者設定初始主機。	設置 SSH 密鑰，bash 配置文件，ORATAB 和符號鏈接。建立甲骨文目錄。	AWS SysAdmin 管理員
在 Amazon EC2 中為旁觀者設定資料庫副本	使用 RMAN 建立資料庫副本、啟用補充記錄日誌，以及建立待命控制檔。複製完成後，請將資料庫置於復原模式。	AWS SysAdmin、DBA
設定「Oracle 資料保全」。	修改您的偵聽器 .ora 文件並啟動監聽器。設定新的歸檔目的地。將旁觀者置於恢復模式，替換臨時文件以避免 future 的損壞，必要時安裝 crontab 以防止存檔目錄空間不足，並編輯源和待命文件的 manage-trclog-files-oracle.cfg 文件。	AWS SysAdmin、DBA
準備 Oracle 資料庫以同步出貨。	新增待命記錄檔並變更復原模式。在來源主要和來源待命狀態上，將記錄傳送變更為 SYNC AFFIRM。切換主日誌、透過 Amazon EC2 旁觀者警示日誌確認您正在使用待命日誌檔，並確認重做串流以 SYNC 方式流動。	AWS SysAdmin、DBA

使用 AWS DMS 遷移資料

任務	描述	所需技能
在 AWS DMS 中建立複寫執行個體。	填寫名稱、執行個體類別、VPC (與 Amazon EC2 執行個體相同)、異地同步備份和公共協助工具的欄位。在進階下，指定已配置的儲存體、子網路群組、可用區域、VPC 安全群組和 AWS Key Management Service (AWS KMS) 金鑰。	AWS SysAdmin、DBA
建立來源資料庫端點。	指定端點名稱、類型、來源引擎 (Oracle)、伺服器名稱 (Amazon EC2 私有 DNS 名稱)、連接埠、SSL 模式、使用者名稱、密碼、SID、VPC (指定具有複寫執行個體的 VPC) 以及複寫執行個體。若要測試連線，請選擇 [執行測試]，然後建立端點。您也可以設定下列進階設定：maxFileSize和「numberDataType縮放」。	AWS SysAdmin、DBA
將 AWS DMS Connect 到 Amazon RDS for PostgreSQL。	為跨 VPC 的連線建立移轉安全性群組。	AWS SysAdmin、DBA
建立目標資料庫端點。	指定端點名稱、類型、來源引擎 (PostgreSQL)、伺服器名稱 (Amazon RDS 端點)、連接埠、SSL 模式、使用者名稱、密碼、資料庫名稱、VPC (指定具有複寫執行個體的 VPC) 和複寫執行個體。若要測試連線，請選擇 [執行測試]，然	AWS SysAdmin、DBA

任務	描述	所需技能
	後建立端點。您也可以設定下列進階設定：maxFileSize 和「numberDataType縮放」。	
建立 AWS DMS 複寫任務。	指定工作名稱、複製執行個體、來源和目標端點，以及複製執行個體。對於移轉類型，請選擇移轉現有資料並複寫進行中的變更。清除「建立時啟動工作」核取方塊。	AWS SysAdmin、DBA
設定 AWS DMS 複寫任務設定。	針對目標資料表準備模式，選擇「不執行任何動 完全負載完成後停止任務（創建主鍵）。指定受限或完整 LOB 模式，並啟動控制表。或者，您可以設定CommitRate進階設定。	DBA
設定表格對映。	在「表格對應」段落中，為移轉中包含的所有綱要中的所有表格建立「包括」規則，然後建立「排除」規則。新增三個轉換規則，將結構描述、資料表和資料行名稱轉換為小寫，並新增此特定移轉所需的任何其他規則。	DBA
開始工作。	啟動複寫工作。確保滿載正在運行。在主要 Oracle 資料庫上執行 ALTER 系統交換器記錄檔，以啟動作業。	DBA

任務	描述	所需技能
從 AWS SCT 執行中間移轉指令碼。	在 Amazon RDS for PostgreSQL 中，執行以下由 AWS SCT 產生的指令碼： • create_index.sql • create_constraint.sql	DBA
重新啟動工作以繼續變更資料擷取 (CDC)。	在適用 Amazon RDS for PostgreSQL 的資料庫執行個體上執行真空，然後重新啟動 AWS DMS 任務以套用快取的疾病控制中心變更。	DBA

切換至 PostgreSQL 資料庫

任務	描述	所需技能
檢閱 AWS DMS 日誌和驗證表格是否有任何錯誤。	檢查並修正任何複寫或驗證錯誤。	DBA
停止所有 Oracle 相依性。	停止所有的 Oracle 相依性、關閉 Oracle 資料庫上的監聽器，然後執行 ALTER 系統切換日誌檔案。如果沒有顯示任何活動，請停止 AWS DMS 任務。	DBA
從 AWS SCT 執行移轉後指令碼。	在 Amazon RDS for PostgreSQL 中，執行以下由 AWS SCT 產生的指令碼： • create_foreign_key_constraint.sql • create_triggers.sql	DBA

任務	描述	所需技能
完成其他 Amazon RDS for PostgreSQL 驟。	如果需要，增加序列以匹配 Oracle，運行真空和分析，並拍攝快照以確保合規性。	DBA
開啟與 Amazon RDS for PostgreSQL。	從 Amazon RDS for PostgreSQL 移除 AWS DMS 安全群組、新增生產安全群組，並將您的應用程式指向新的資料庫。	DBA
清理 AWS DMS 物件。	移除端點、複寫任務、複寫執行個體和 EC2 執行個體。	SysAdmin, DBA

相關資源

- AWS DMS 文件集
- AWS SCT 文件
- Amazon RDS for PostgreSQL 價

使用甲骨文從甲骨文數據庫遷移到亞馬遜 RDS GoldenGate

由神達尼 (AWS)、拉傑什庫瑪爾薩班卡 (AWS) 和新杜莎帕特魯 (AWS) 所建立

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for PostgreSQL
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式顯示如何使用 Oracle 雲端基礎設施 (OCI) 將甲骨文資料庫遷移到適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS)。 GoldenGate

透過使用 Oracle GoldenGate，您可以在來源資料庫與一或多個目的地資料庫之間複製資料，而且停機時間最短。

備註：來源 Oracle 資料庫可以位於現場部署，也可以位於 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上。您可以在使用內部部署複寫工具時使用類似的程序。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 甲骨文 GoldenGate 許可證
- 用於連線至 PostgreSQL 資料庫的 Java 資料庫連線能力 (JDBC) 驅動程式
- 使用 AWS 結構描述轉換工具 (AWS SCT) 建立的結構描述和表格，在目標 Amazon RDS for PostgreSQL 資料庫

限制

- Oracle 只 GoldenGate 能複製現有的資料表資料 (初始載入) 和持續變更 (變更資料擷取)

產品版本

- Oracle 資料庫企業版 10g 或更新版本
- 甲骨文 GoldenGate 12.2.0.1.1 適用於甲骨文或更新版本
- 甲骨文 GoldenGate 12.2.0.1.1 適用於 PostgreSQL 或更新版本

架構

下圖顯示使用甲骨文將甲骨文資料庫遷移到亞馬遜 RDS 的工作流程範例： GoldenGate

該圖顯示以下工作流程：

1. 「Oracle GoldenGate 擷取」處理作業會針對來源資料庫執行，以擷取資料。
2. Oracle GoldenGate 複本程序會將擷取的資料傳送到目標 Amazon RDS for PostgreSQL 料庫。

工具

- Oracle 可協 GoldenGate助您在 Oracle 雲端基礎架構中設計、執行、協調和監控資料複製和串流資料處理解決方案。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。

史诗

下載並安裝甲骨文 GoldenGate

任務	描述	所需技能
下載甲骨文 GoldenGate。	請下載以下版本的甲骨文 GoldenGate： • 甲骨文 GoldenGate 12.2.0.1.1 適用於甲骨文或更新版本 • 甲骨文 GoldenGat e 12.2.0.1.1 適用於 PostgreSQL 或更新版本	DBA

任務	描述	所需技能
	若要下載軟體，請參閱 Oracle 網站上的「Oracle 下 GoldenGate 載管理系統」。	
在來源「Oracle GoldenGate 資料庫」伺服器上安裝適用於 Oracle 的 Oracle。	如需相關指示，請參閱 Oracle GoldenGate 文件集。	DBA
在亞馬遜 EC2 執行個體上安裝甲骨 GoldenGate PostgreSQL 資料庫。	如需相關指示，請參閱 Oracle GoldenGate 文件集。	DBA

GoldenGate 在來源和目標資料庫上設定 Oracle

任務	描述	所需技能
在來源資料庫上設定 Oracle 資料庫的 Oracle。 GoldenGate	如需相關指示，請參閱 Oracle GoldenGate 文件集。 請確定您已設定下列項目： • 補充記錄 • 甲骨文 GoldenGate 用戶 • 任何必要的授權和權限 • 參數檔 • 經理流程 • 目錄 • 全局文件 • 甲骨文錢包	DBA
在目標資料庫上設定 GoldenGate 適用於 PostgreSQL 的甲骨文。	如需相關指示，請參閱甲骨 GoldenGate 文網站上的第六部分。 請確定您已設定下列項目：	DBA

任務	描述	所需技能
	• 經理流程 • 全局文件 • 甲骨文錢包	

設定資料擷取

任務	描述	所需技能
在來源資料庫中設定「擷取」處理作業。	在來源 Oracle 資料庫中，建立擷取檔案以擷取資料。 如需指示，請參閱 Oracle 說明文件中的新增擷取。 附註：擷取檔案包括擷取參數檔案和軌跡檔案目錄的建立。	DBA
設置資料汲取，以將軌跡檔案從來源傳輸到目標資料庫。	依照 Oracle 網站上「資料庫公用程式」中 PARFILE 中的指示，建立 EXTRACT 參數檔案和軌跡檔案目錄。 如需詳細資訊，請參閱什麼是軌跡？ in 融合中間件了解甲骨文 GoldenGate在甲骨文網站上。	DBA
在亞馬遜 EC2 執行個體上設定複寫。	建立複製參數檔案和軌跡檔案目錄。 如需有關建立複製參數檔案的詳細資訊，請參閱 Oracle 資料庫說明文件中的 3.5 節驗證參數檔案。	DBA

任務	描述	所需技能
	如需有關建立追蹤檔案目錄的詳細資訊，請參閱 Oracle Cloud 說明文件中的建立追蹤。 重要事項：請確定您在目標的 GLOBALS 檔案中新增檢查點資料表項目。 如需詳細資訊，請參閱什麼是複本？ in 融合中間件了解甲骨文 GoldenGate在甲骨文網站上。	

設定資料複製

任務	描述	所需技能
在來源資料庫中，建立參數檔案以擷取初始載入的資料。	請遵循 Oracle 雲端文件中 GGSCI 中建立參數檔案中的指示進行。 重要:確定管理員正在目標上執行。	DBA
在目標資料庫中，建立參數檔案以複製初始載入的資料。	請遵循 Oracle 雲端文件中 GGSCI 中建立參數檔案中的指示進行。 重要:請確定您已新增並啟動複本程序。	DBA

切換到 Amazon RDS for PostgreSQL 庫

任務	描述	所需技能
停止複本處理作業，並確定來源和目標資料庫同步。	比較來源和目標資料庫之間的資料列計數，以確定資料複製成功。	DBA
設定資料定義語言 (DDL) 支援。	執行 DDL 指令碼，以便在 PostgreSQL 上建立觸發程序、序列、同義字和參考索引鍵。 注意：您可以使用任何標準 SQL 用戶端應用程式連線到資料庫叢集中的資料庫。例如，您可以使用 pgAdmin 連線到資料庫執行個體。	DBA

相關資源

- Amazon RDS for PostgreSQL（Amazon RDS 用戶指南）
- Amazon EC2 文件
- 甲骨文 GoldenGate 支持的處理方法和數據庫（Oracle 文檔）

使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移

來源：甲骨文	目標：Redshift	R 型：重新建築
環境：生產	技術：移轉、分析、資料庫	工作量：甲骨文
AWS 服務：Amazon Redshift；AWS DMS		

Summary

此模式為使用 AWS 資料庫遷移服務 (AWS DMS) 和 AWS Schema Conversion Tool (AWS SCT) 將 Oracle 資料庫遷移到 Amazon Web Services 務 (AWS) 雲端中的 Amazon Redshift 雲端資料倉儲提供指導。該模式涵蓋內部部署或安裝在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的來源 Oracle 資料庫。它還涵蓋適用於甲骨文數據庫的 Amazon 關係數據庫服務（亞馬遜 RDS）。

先決條件和限制

先決條件

- 在現場部署資料中心或 AWS 雲端中執行的 Oracle 資料庫
- 有效的 AWS 帳戶
- 熟悉使用 Oracle 資料庫做為 AWS DMS 的來源
- 熟悉使用 Amazon Redshift 資料庫做為 AWS DMS 的目標
- Amazon RDS，亞 Amazon Redshift，適用的數據庫技術和 SQL 的知識
- 適用於已安裝 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式

產品版本

- 對於自我管理的 Oracle 資料庫，AWS DMS 支援 10.2 及更新版本的所有 Oracle 資料庫版本 (適用於版本 10。 x），11 克和最多 12.2，18 分和 19 克。對於 AWS 管理的亞馬遜 RDS 適用於甲骨文資料庫，AWS DMS 支援 11g 版本 (11.2.0.4 及更新版本) 和最高 12.2、18c 和 19c 的所有 Oracle 資料庫版本。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。

架構

源, 技術, 堆棧

下列其中一項：

- 內部部署 Oracle 資料庫
- EC2 執行個體上的甲骨文資料庫
- Amazon RDS for Oracle 數據庫實例

目標技術堆疊

- Amazon Redshift

目標架構

從在 AWS 雲端執行的 Oracle 資料庫到 Amazon Redshift：

從現場部署資料中心執行的 Oracle 資料庫到 Amazon Redshift：

工具

- AWS DMS-AWS 資料遷移服務 (AWS DMS) 可協助您快速安全地將資料庫遷移到 AWS。來源資料庫會在移轉期間保持完全運作，將依賴資料庫之應用程式的停機時間降至最低。AWS DMS 可以在最廣泛使用的商業和開放原始碼資料庫之間移轉您的資料。
- AWS SCT-AWS 結構描述轉換工具 (AWS SCT) 可用於將現有的資料庫結構描述從一個資料庫引擎轉換為另一個資料庫引擎。它支持各種數據庫引擎，包括甲骨文，SQL 服務器和 PostgresSQL，作為源。

史诗

準備移轉

任務	描述	所需技能
驗證資料庫版本。	驗證來源和目標資料庫版本，並確定 AWS DMS 支援這些版本。如需有關支援 Oracle 資料庫版本的詳細資訊，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源。如需使用 Amazon Redshift 做為目標的相關資訊，請參閱使用 Amazon Redshift 資料庫做為 AWS DMS 的目標。	DBA
建立 VPC 和安全性群組。	如果虛擬私有雲 (VPC) 不存在，請在您的 AWS 帳戶中建立虛擬私有雲 (VPC)。為來源和目標資料庫的輸出流量建立安全群組。如需詳細資訊，請參閱 Amazon Virtual Private Cloud 端 (Amazon VPC) 文件。	系統管理員
安裝 AWS SCT。	下載並安裝最新版本的 AWS SCT 及其對應的驅動程式。如需詳細資訊，請參閱安裝、驗證和更新 AWS SCT。	DBA
為 AWS DMS 任務建立使用者。	在來源資料庫中建立 AWS DMS 使用者，並授與其讀取權限。AWS SCT 和 AWS DMS 都會使用此使用者。	DBA
測試資料庫連線能力。	測試與 Oracle 資料庫執行個體的連線。	DBA

任務	描述	所需技能
在 AWS SCT 中建立新專案。	開啟 AWS SCT 工具並建立新專案。	DBA
分析要移轉的 Oracle 綱要。	使用 AWS SCT 分析要移轉的結構描述，並產生資料庫遷移評估報告。如需詳細資訊，請參閱 AWS SCT 文件中的建立資料庫遷移評估報告。	DBA
檢閱評估報告。	檢閱有關遷移可行性的報告。某些 DB 物件可能需要手動轉換。如需有關報告的詳細資訊，請參閱 AWS SCT 文件中的檢視評估報告。	DBA

準備目標資料庫

任務	描述	所需技能
創建一個 Amazon Redshift 集群。	在您先前建立的 VPC 中建立一個 Amazon Redshift 叢集。如需詳細資訊，請參閱 Amazon Redshift 文件中的 Amazon Redshift 叢集。	DBA
建立資料庫使用者。	從 Oracle 來源資料庫擷取使用者、角色和授權的清單。在目標 Amazon Redshift 資料庫中建立使用者，並套用上一個步驟中的角色。	DBA
評估資料庫參數。	複查 Oracle 來源資料庫中的資料庫選項、參數、網路檔案和資料庫連結，並評估它們對目標的適用性。	DBA

任務	描述	所需技能
將任何相關設定套用至目標。	如需有關此步驟的詳細資訊，請參閱 Amazon Redshift 文件中的組態參考資料。	DBA

在目標資料庫中建立物件

任務	描述	所需技能
在目標資料庫中建立 AWS DMS 使用者。	在目標資料庫中建立 AWS DMS 使用者，並授與其讀取和寫入權限。驗證來自 AWS SCT 的連線能力。	DBA
轉換結構描述、檢閱 SQL 報告，並儲存任何錯誤或警告。	如需詳細資訊，請參閱 AWS SCT 文件中的使用 AWS SCT 轉換資料庫結構描述。	DBA
將結構描述變更套用至目標資料庫，或將其儲存為 .sql 檔案。	如需指示，請參閱 AWS SCT 文件中的 AWS SCT 中儲存和套用轉換後的結構描述。	DBA
驗證目標資料庫中的物件。	驗證在目標資料庫的上一個步驟中建立的物件。重新撰寫或重新設計任何未成功轉換的物件。	DBA
禁用外鍵和觸發器。	禁用任何外鍵和觸發器。執行 AWS DMS 時，這些可能會導致在滿載程序期間發生資料載入問題。	DBA

使用 AWS DMS 遷移資料

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。	登入 AWS 管理主控台，然後開啟 AWS DMS 主控台。在瀏覽窗格中，選擇 [複製執行個體] > [建立複製執行個體 如需詳細指示，請參閱 AWS DMS 文件中的 AWS DMS 入門中的步驟 1。	DBA
建立來源端點和目標端點。	建立來源和目標端點，測試從複寫執行個體到來源端點和目標端點的連線。如需詳細指示，請參閱 AWS DMS 文件中的 AWS DMS 入門中的步驟 2。	DBA
建立複製工作。	建立複寫工作並選取適當的移轉方法。如需詳細指示，請參閱 AWS DMS 文件中的 AWS DMS 入門中的步驟 3。	DBA
開始資料複製。	啟動複寫工作，並監視記錄檔是否有任何錯誤。	DBA

遷移應用程式

任務	描述	所需技能
建立應用程式伺服器	在 AWS 上建立新的應用程式伺服器。	應用程式擁
遷移應用程序代碼。	將應用程式程式碼移轉至新伺服器。	應用程式擁

任務	描述	所需技能
設定應用程式伺服器。	設定目標資料庫和驅動程式的應用程式伺服器。	應用程式擁
優化應用程序代碼。	最佳化目標引擎的應用程式程式碼。	應用程式擁

切換到目標數據庫

任務	描述	所需技能
驗證使用者。	在目標 Amazon Redshift 資料庫中，驗證使用者並授予他們角色和權限。	DBA
驗證應用程式是否已鎖定。	確保應用程序已鎖定，以防止進一步的更改。	應用程式擁
驗證資料。	驗證目標 Amazon Redshift 資料庫中的資料。	DBA
啟用外鍵和觸發器。	在目標 Amazon Redshift 資料庫中啟用外部索引鍵和觸發器。	DBA
Connect 到新的資料庫。	將應用程式設定為連線到新的 Amazon Redshift 資料庫。	應用程式擁
執行最終檢查。	在上線之前，請執行最後的全面系統檢查。	DBA，應用程式擁有者
去直播吧。	使用目標 Amazon Redshift 資料庫上線。	DBA

關閉移轉專案

任務	描述	所需技能
關閉臨時 AWS 資源。	關閉臨時 AWS 資源，例如 AWS DMS 複寫執行個體和用於 AWS SCT 的 EC2 執行個體。	DBA, 系統管理員
檢閱文件。	檢閱和驗證移轉專案文件。	DBA, 系統管理員
收集指標。	收集有關移轉專案的資訊，例如移轉時間、手動與工具工作的百分比，以及節省總成本。	DBA, 系統管理員
關閉專案。	關閉專案並提供意見反應。	DBA, 系統管理員

相關資源

參考

- AWS DMS 使用者指南
- 使用者指南
- Amazon Redshift 入門指南

教學課程和影片

- 深入瞭解 AWS SCT 和 AWS DMS (來自 AWS RE: 發明 2019 的簡報)
- 開始使用 AWS Database Migration Service

使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Aurora

由桑西爾·拉馬薩米（AWS）創建

環境：PoC 或試點	來源：甲骨文數據庫	目標：Amazon Aurora PostgreSQL 兼容
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon Aurora		

Summary

此模式說明如何使用 AWS 資料遷移服務 (AWS DMS) 和 AWS Schema Conversion Tool (AWS SCT)，將甲骨文資料庫遷移到 Amazon Aurora PostgreSQL 相容版本。

該模式涵蓋內部部署的來源 Oracle 資料庫、安裝在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 Oracle 資料庫，以及適用於 Oracle 資料庫的 Amazon Relational Database Service 服務 (Amazon RDS)。該模式將這些數據庫轉換為 Aurora PostgreSQL 兼容。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 位於現場部署資料中心或 AWS 雲端中的 Oracle 資料庫。
- 安裝在本機電腦或 EC2 執行個體上的 SQL 用戶端。
- 適用於 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式，安裝在本機電腦或已安裝 AWS SCT 的 EC2 執行個體上。

限制

- 資料庫大小限制：128 TB
- 如果來源資料庫支援商業 off-the-shelf (COTS) 應用程式，或是特定於廠商，您可能無法將其轉換為其他資料庫引擎。在使用此模式之前，請確認應用程式支援 Aurora PostgreSQL 相容。

產品版本

- 對於自我管理的 Oracle 資料庫，AWS DMS 支援 10.2 及更新版本 (適用於版本 10.x)、11g 以及最高 12.2、18c 和 19c 的所有 Oracle 資料庫版本。如需受支援的 Oracle 資料庫版本 (包括自我管理和 Amazon RDS for Oracle 版本) 的最新清單，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源和使用 PostgreSQL 資料庫做為 AWS DMS 的目標。
- 我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。如需 AWS SCT 支援的 Oracle 資料庫版本的相關資訊，請參閱 AWS SCT 文件。
- Aurora 支援 Amazon Aurora PostgreSQL 版本和引擎版本中列出的 PostgreSQL 版本。

架構

源, 技術, 堆棧

下列其中一項：

- 內部部署 Oracle 資料庫
- EC2 執行個體上的甲骨文資料庫
- Amazon RDS for Oracle 數據庫實例

目標技術堆疊

- Aurora 郵政兼容

目標架構

資料移轉架構

- 從在 AWS 雲端中執行的 Oracle 資料庫

- 從內部部署資料中心執行的 Oracle 資料庫

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

史诗

準備移轉

任務	描述	所需技能
準備來源資料庫。	若要準備來源資料庫，請參閱 AWS SCT 文件中的使用 Oracle 資料庫做為 AWS SCT 的來源。	DBA
為 AWS SCT 建立 EC2 執行個體。	視需要建立和設定 AWS SCT 的 EC2 執行個體。	DBA
下載 AWS SCT。	下載最新版本的 AWS SCT 和相關驅動程式。如需詳細資訊，請參閱 AWS SCT 文件中的安裝、驗證和更新 AWS SCT。	DBA
新增使用者和權限。	在來源資料庫中新增及驗證必要條件使用者和權限。	DBA
建立 AWS SCT 專案。	為工作負載建立 AWS SCT 專案，並連線到來源資料庫。如需指示，請參閱 AWS SCT 文件中的建立 AWS SCT 專案和新增資料庫伺服器。	DBA
評估可行性。	產生評估報告，摘要說明無法自動轉換之結構描述的行動項	DBA

任務	描述	所需技能
	目，並提供手動轉換工作的預估值。如需詳細資訊，請參閱 AWS SCT 文件中的建立和檢閱資料庫遷移評估報告。	

準備目標資料庫

任務	描述	所需技能
建立目標 Amazon RDS 資料庫執行個體。	使用 Amazon Aurora 做為資料庫引擎，建立目標 Amazon RDS 資料庫執行個體。如需指示，請參閱 Amazon RDS 文件中的建立 Amazon RDS 資料庫執行個體。	DBA
擷取使用者、角色和權限。	從來源資料庫擷取使用者、角色和權限的清單。	DBA
地圖使用者。	將現有的資料庫使用者對應至新的資料庫使用者。	應用所有者
建立使用者。	在目標資料庫中建立使用者。	DBA，應用程式擁有者
套用角色。	將上一個步驟中的角色套用至目標資料庫。	DBA
檢查選項、參數、網路檔案和資料庫連結。	複查來源資料庫中的選項、參數、網路檔案和資料庫連結，然後評估它們對目標資料庫的適用性。	DBA
套用設定。	將任何相關設定套用至目標資料庫。	DBA

傳送物件

任務	描述	所需技能
設定 AWS SCT 連線能力。	設定與目標資料庫的 AWS SCT 連線。	DBA
使用 AWS SCT 轉換結構描述。	AWS SCT 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式。該工具無法自動轉換的任何代碼都會清楚標記，以便您可以手動轉換它。	DBA
檢閱報告。	複查產生的 SQL 報告，並儲存任何錯誤和警告。	DBA
套用自動結構描述變更。	將自動化結構描述變更套用至目標資料庫，或將其儲存為 .sql 檔案。	DBA
驗證物件。	驗證 AWS SCT 是否在目標上建立了物件。	DBA
處理未轉換的項目。	手動重寫、拒絕或重新設計任何無法自動轉換的項目。	DBA，應用程式擁有者
套用角色和使用者權限。	套用產生的角色和使用者權限，並檢閱任何例外狀況。	DBA

遷移數據

任務	描述	所需技能
確定方法。	決定移轉資料的方法。	DBA
建立複寫執行個體。	從 AWS DMS 主控台建立複寫執行個體。如需詳細資訊，請	DBA

任務	描述	所需技能
	參閱 AWS DMS 文件中的使用 AWS DMS 複寫執行個體。	
建立來源端點和目標端點。	若要建立端點，請遵循 AWS DMS 文件中建立來源和目標端點中的指示。	DBA
建立複製工作。	若要建立任務，請參閱 AWS DMS 文件中的使用 AWS DMS 任務。	DBA
啟動複寫工作並監視記錄檔。	如需有關此步驟的詳細資訊，請參閱 AWS DMS 文件中的監控 AWS DMS 任務。	DBA

移轉應用程式

任務	描述	所需技能
分析並轉換應用程式程式碼中的 SQL 項目。	使用 AWS SCT 分析和轉換應用程式程式碼中的 SQL 項目。當您將資料庫結構描述從一個引擎轉換到另一個引擎，您也需更新應用程式中的 SQL 程式碼，以便與新的資料庫引擎互動，取代舊引擎。您可以檢視、分析、編輯和儲存轉換後的 SQL 程式碼。	應用所有者
建立應用程式伺服器	在 AWS 上建立新的應用程式伺服器。	應用所有者
遷移應用程序代碼。	將應用程式程式碼移轉至新伺服器。	應用所有者

任務	描述	所需技能
設定應用程式伺服器。	設定目標資料庫和驅動程式的應用程式伺服器。	應用所有者
修正程式碼。	修正應用程式中原始碼資料庫引擎專屬的任何程式碼。	應用所有者
優化代碼。	針對目標資料庫引擎優化您的應用程式程式碼。	應用所有者

切過

任務	描述	所需技能
切換到目標數據庫。	執行新資料庫的切換。	DBA
鎖定應用程式。	鎖定應用程式以防止任何進一步變更。	應用所有者
驗證變更。	驗證是否已將所有變更傳播至目標資料庫。	DBA
重定向到目標數據庫。	將新的應用程式伺服器指向目標資料庫。	應用所有者
檢查一切。	執行最後的全面性系統檢查。	應用所有者
去直播吧。	完成最終切換任務。	應用所有者

關閉專案

任務	描述	所需技能
關閉臨時資源。	關閉臨時 AWS 資源，例如 AWS DMS 複寫執行個體和用	DBA，應用程式擁有者

任務	描述	所需技能
	於 AWS SCT 的 EC2 執行個體。	
更新意見反應。	針對內部團隊更新 AWS DMS 程序的意見反應。	DBA，應用程式擁有者
修訂流程和範本。	修改 AWS DMS 程序並視需要改善範本。	DBA，應用程式擁有者
驗證文件。	審核並驗證專案文件。	DBA，應用程式擁有者
收集指標。	收集指標以評估遷移時間、手動與工具成本節約的百分比等。	DBA，應用程式擁有者
關閉專案。	關閉遷移專案並向利益相關者提供意見反應。	DBA，應用程式擁有者

相關資源

參考

- 使用 Oracle 資料庫做為 AWS DMS 的來源
- 使用 PostgreSQL 資料庫做為 AWS Database Migration Service 的目標
- 甲骨文數據庫 11g/12c 到 Amazon Aurora 與 PostgreSQL 兼容性 (9.6.x) 遷移教戰手冊
- 甲骨文資料庫 19c 到 Amazon Aurora 與 PostgreSQL 相容性 (12.4) 移轉教戰手冊
- 將適用於甲骨文數據庫的亞馬遜 RDS 遷移到 Amazon Aurora PostgreSQL 兼容版
- AWS 資料遷移服務
- AWS Schema Conversion Tool
- 從甲骨文遷移到 Amazon Aurora
- Amazon RDS 定價

教學課程和影片

- 資料庫移轉逐步解說

- 開始使用 AWS DMS
- Amazon RDS 入門
- AWS 資料遷移服務 (影片)
- 將甲骨文數據庫遷移到 PostgreSQL（視頻）

其他資訊

.

將資料從內部部署 Oracle 資料庫遷 PostgreSQL 至 Aurora

由鄧美雪（AWS）和蜀南祥（AWS）創建

環境：PoC 或試點	來源：甲骨文	目標：Aurora 郵政兼容
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon Aurora；AWS DMS；AWS SCT		

Summary

此模式提供從現場部署 Oracle 資料庫移轉至 Amazon Aurora PostgreSQL 相容版本的資料指引。它針對包含具有高資料處理語言 (DML) 活動的大型表格的多 TB Oracle 資料庫停機時間最少的線上資料移轉策略。「Oracle 作用中資料保全」待命資料庫是用來卸載主要資料庫之資料移轉的來源。在完全負載期間，可以暫停從 Oracle 主要資料庫到待命資料庫的複寫，以避免發生 ORA-01555 錯誤。

主鍵（PK）或具有數據類型 NUMBER 的外鍵（FK）中的表列通常用於在 Oracle 中存儲整數。我們建議您在 PostgreSQL 中將它們轉換為 INT 或大 INT，以獲得更好的性能。您可以使用 AWS Schema Conversion Tool (AWS SCT) 變更 PK 和 FK 欄的預設資料類型對應。如需詳細資訊，請參閱 AWS 部落格文章將 NUMBER 資料類型從甲骨文轉換為 PostgreSQL。) 此模式中的資料遷移使用 AWS Database Migration Service (AWS DMS) 進行全負載和變更資料擷取 (CDC)。

您也可以使用此模式將現場部署 Oracle 資料庫遷移到適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 或亞馬遜彈性運算雲端 (亞馬遜 EC2) 上託管的甲骨文資料庫遷移到 Amazon RDS (適用於 PostgreSQL) 或 Aurora PostgreSQL 相容。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中已設定作用中資料保全待命的 Oracle 來源資料庫
- 在現場部署資料中心和 AWS 雲端之間設定 AWS Direct Connect
- 熟悉使用 Oracle 資料庫做為 AWS DMS 的來源
- 熟悉使用 PostgreSQL 資料庫做為 AWS 資料庫管理系統的目標

限制

- Amazon Aurora 資料庫叢集可以建立最多 128 TiB 的儲存體。Amazon RDS for PostgreSQL 的資料庫執行個體可以建立最多 64 TiB 的儲存體。如需最新儲存資訊，請參閱 AWS 文件中的 Amazon Aurora 儲存和可靠性以及 Amazon RDS 資料庫執行個體儲存。

產品版本

- AWS DMS 支援 10.2 及更新版本 (適用於版本 10.x)、11 克以及最高 12.2、18c 和 19 c 的所有甲骨文資料庫版本。如需支援版本的最新清單，請參閱 AWS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源。

架構

源, 技術, 堆棧

- 設定 Oracle 作用中資料保全待命的內部部署 Oracle 資料庫

目標技術堆疊

- Aurora 郵政兼容

資料移轉架構

工具

- AWS DMS-AWS Database Migration Service (AWS DMS) 支援多個來源和目標資料庫。如需支援的 Oracle 來源和目標資料庫版本和版本清單，請參閱 AWS DMS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源。如果 AWS DMS 不支援來源資料庫，您必須選取另一種移轉階段 6 中的資料的方法 (在「史詩」區段中)。 重要注意事項：由於這是異質移轉，因此您必須先檢查資料庫是否支援商業 off-the-shelf (COTS) 應用程式。如果應用程式為 COTS，請在繼續之前諮詢廠商以確認是否支援 Aurora PostgreSQL 相容。如需詳細資訊，請參閱 AWS 文件中的 AWS DMS 逐步移轉逐步解說。
- AWS SCT-AWS S chema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，以促進異質資料庫遷移。該工具轉換的自定義代碼包括視圖，存儲過程和函數。該工具無法自動轉換的任何代碼都會清楚標記，以便您可以自行轉換。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA
安裝 AWS SCT 和驅動程式。		DBA
新增和驗證 AWS SCT 先決條件使用者和補助金來源資料庫。		DBA
為工作負載建立 AWS SCT 專案，並連線到來源資料庫。		DBA
產生評估報告並評估可行性。		DBA，應用程式擁有者

準備目標資料庫

任務	描述	所需技能
建立與 Aurora PostgreSQL 相容的目標資料庫。		DBA
從來源資料庫擷取使用者、角色和授權清單。		DBA
將現有的資料庫使用者對應至新的資料庫使用者。		應用所有者
在目標資料庫中建立使用者。		DBA
將上一個步驟中的角色套用至目標 Aurora PostgreSQL 相容資料庫。		DBA

任務	描述	所需技能
複查來源資料庫中的資料庫選項、參數、網路檔案和資料庫連結，並評估它們對目標資料庫的適用性。		DBA，應用程式擁有者
將任何相關設定套用至目標資料庫。		DBA

準備資料庫物件程式碼轉換

任務	描述	所需技能
設定與目標資料庫的 AWS SCT 連線。		DBA
在 AWS SCT 中轉換結構描述，並將轉換後的程式碼儲存為 .sql 檔案。		DBA，應用程式擁有者
手動轉換任何無法自動轉換的資料庫物件。		DBA，應用程式擁有者
優化數據庫代碼轉換。		DBA，應用程式擁有者
根據物件類型，將 .sql 檔案分隔成多個 .sql 檔案。		DBA，應用程式擁有者
驗證目標資料庫中的 SQL 指令碼。		DBA，應用程式擁有者

準備資料移轉

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。		DBA
建立來源端點和目標端點。	如果 PKS 和 FK 的資料類型是從 Oracle 中的數字轉換為 PostgreSQL 中的 BIGINT，請考慮在建立來源端點`numberDataTypeScale=-2` 時指定連線屬性。	DBA

移轉資料 — 滿載

任務	描述	所需技能
在目標資料庫中建立結構描述和表格。		DBA
透過將資料表分組或根據資料表大小分割大型資料表，來建立 AWS DMS 全負載任務。		DBA
短時間停止來源 Oracle 資料庫上的應用程式。		應用所有者
確認 Oracle 待命資料庫與主要資料庫是否同步，並停止從主要資料庫複製到待命資料庫。		DBA，應用程式擁有者
在來源 Oracle 資料庫上啟動應用程式。		應用所有者
從 Oracle 待命資料庫到 Aurora PostgreSQL 相容的		DBA

任務	描述	所需技能
資料庫，以 parallel 方式啟動 AWS DMS 全負載任務。		
在滿載完成後建立 PKs 和次要索引。		DBA
驗證資料。		DBA

遷移資料 — CDC

任務	描述	所需技能
透過指定自訂 CDC 開始時間或系統變更編號 (SCN)，以建立 AWS DMS 進行中的複寫任務，方法是在 Oracle 待命與主資料庫同步化時，以及在先前的任務中重新啟動應用程式之前。		DBA
parallel 啟動 AWS DMS 任務，將進行中的變更從 Oracle 待命資料庫複寫到 Aurora PostgreSQL 相容的資料庫。		DBA
重新建立從 Oracle 主要資料庫到待命資料庫的複製。		DBA
當目標 Aurora PostgreSQL 相容資料庫幾乎與來源 Oracle 資料庫同步時，請監視記錄檔並停止 Oracle 資料庫上的應用程式。		DBA，應用程式擁有者

任務	描述	所需技能
當目標與來源 Oracle 資料庫完全同步時，請停止 AWS DMS 任務。		DBA
建立 SDK 並驗證目標資料庫中的資料。		DBA
在目標資料庫中建立函數、檢視、觸發程序、序列及其他物件類型。		DBA
在目標資料庫中套用角色授與。		DBA

移轉應用程式

任務	描述	所需技能
使用 AWS SCT 分析和轉換應用程式程式碼中的 SQL 陳述式。		應用所有者
在 AWS 上建立新的應用程式伺服器。		應用所有者
將應用程式程式碼移轉至新伺服器。		應用所有者
設定目標資料庫和驅動程式的應用程式伺服器。		應用所有者
修正應用程式中原始碼資料庫引擎專屬的任何程式碼。		應用所有者
最佳化目標資料庫的應用程式程式碼。		應用所有者

切過

任務	描述	所需技能
將新的應用程式伺服器指向目標資料庫。		DBA，應用程式擁有者
執行完整性檢查。		DBA，應用程式擁有者
去直播吧。		DBA，應用程式擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, 系統管理員
審核並驗證專案文件。		DBA，應用程式擁有者
收集移轉時間的指標、手動與工具使用的百分比、節省成本，以及類似資料。		DBA，應用程式擁有者
關閉專案並提供意見反應。		DBA，應用程式擁有者

相關資源

參考

- 甲骨文資料庫與 Aurora PostgreSQL 相容：移轉教戰手冊
- 將 Amazon RDS for Oracle Database 遷移到 Amazon Aurora MySQL
- AWS 管理系統網站
- AWS DMS 說明文件
- 網站
- AWS SCT 文件
- 從甲骨文遷移到 Amazon Aurora

教學課程

- 開始使用 AWS DMS
- Amazon RDS 入門
- AWS Database Migration Service 逐步解說

使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器

由阿米特·庫馬爾（AWS）創建

環境：PoC 或試點	資料來源：SAP 日月光	目標：Amazon RDS for SQL Server
R 型：重新建築	工作負載：SAP	技術：移民、資料庫、現代化
AWS 服務：Amazon RDS; AWS DMS		

Summary

此模式提供有關將 SAP 調適性伺服器企業 (ASE) 資料庫遷移到執行 Microsoft SQL Server 的亞馬遜關聯式資料庫服務 (Amazon RDS) 資料庫執行個體的指導。來源資料庫可以位於現場部署資料中心或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上。該模式使用 AWS Database Migration Service (AWS DMS) 遷移資料，以及 (選擇性) 電腦輔助軟體工程 (CASE) 工具來轉換資料庫結構描述。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心或 EC2 執行個體中的 SAP ASE 資料庫
- 已啟動並執行的目標 Amazon RDS for SQL Server 的資料庫

限制

- 資料庫大小限制：64 TB

產品版本

- 僅適用於版本 15.7 或 16 倍。如需最新資訊，請參閱使用 SAP 資料庫做為 AWS DMS 的來源。

- 對於 Amazon RDS 目標資料庫，AWS DMS 支援適用於企業版、標準、網頁和快速版的 Amazon RDS 上的 Microsoft SQL 伺服器版本。如需支援版本的最新資訊，請參閱 AWS DMS 文件。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。

架構

源, 技術, 堆棧

- 位於內部部署或 Amazon EC2 執行個體上的 SAP ASE 資料庫

目標技術堆疊

- Amazon RDS for SQL Server 資料庫執行個體

來源與目標架構

從 Amazon EC2 上的 SAP ASE 資料庫到 Amazon RDS for SQL Server 料庫執行個體：

從現場部署 SAP ASE 資料庫到 Amazon RDS for SQL Server 資料庫執行個體：

工具

- AWS Database Migration Service (AWS DMS) 是一種網路服務，可用來將資料從現場部署的資料庫、Amazon RDS 資料庫執行個體或 EC2 執行個體的資料庫遷移到 AWS 服務上的資料庫，例如 Amazon RDS for SQL Server 或 EC2 執行個體。您也可以將資料庫從 AWS 服務遷移到現場部署資料庫。您可以在異質或同質資料庫引擎之間移轉資料。
- 對於結構描述轉換，您可以選擇性地使用 erwin 資料建模工具或 SAP。 PowerDesigner

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA

任務	描述	所需技能
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA, SysAdmin
識別應用程式移轉策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結構

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 Amazon RDS 資料庫執行個體。		SysAdmin

移轉資料-選項 1

任務	描述	所需技能
手動移轉資料庫結構描述，或使用 CASE 工具，例如 Erwin 資料建模工具或 SAP。PowerDesigner		DBA

移轉資料-選項 2

任務	描述	所需技能
使用 AWS DMS 遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集指標，例如移轉時間、手動與自動化工作的百分比，以及節省成本。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供意見反應。		DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- AWS 管理系統網站
- Amazon RDS 定價
- 使用 SAP ASE 資料庫做為 AWS DMS 的來源
- 適用於 SQL 伺服器的 RDS 自訂限制

教學課程和影片

- 開始使用 AWS DMS
- Amazon RDS 入門
- AWS DMS (影片)
- Amazon RDS（視頻）

使用 AWS DMS 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移

創建者馬塞洛·費爾南德斯（AWS）

環境：PoC 或試點	來源：Microsoft SQL 服務器	目標：Amazon Redshift
R 型：重新建築	工作量：Microsoft	技術：移轉；資料庫
AWS 服務：Amazon Redshift		

Summary

此模式提供使用 AWS 資料遷移服務 (AWS DMS) 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移的指導。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Microsoft SQL 伺服器資料庫
- 已完成使用 Amazon Redshift 資料庫做為 AWS DMS 目標的先決條件，如 AWS DMS 文件所討論

產品版本

- SQL 伺服器 2005-2019 年，企業版，標準版，工作組，開發人員和網頁版。如需支援版本的最新清單，請參閱 AWS 文件中的使用 Microsoft SQL Server 資料庫做為 AWS DMS 的來源。

架構

源, 技術, 堆棧

- 內部部署 Microsoft SQL 伺服器資料庫

目標技術堆疊

- Amazon Redshift

資料移轉架構

工具

- AWS DMS 是一種資料遷移服務，支援多種類型的來源和目標資料庫。如需有關支援與 AWS DMS 搭配使用的 Microsoft SQL Server 資料庫版本和版本的資訊，請參閱 AWS DMS 文件中的使用 Microsoft SQL 伺服器資料庫做為 AWS DMS 的來源。如果 AWS DMS 不支援您的來源資料庫，則必須為資料遷移選取替代方法。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA, 系統管理員
識別儲存需求 (儲存類型和容量)。		DBA, 系統管理員
根據容量、儲存空間功能和網路功能，選擇適當的執行個體類型。		DBA, 系統管理員
識別來源和目標資料庫的網路存取安全性需求。		DBA, 系統管理員
識別應用程式移轉策略。		DBA、應用程式擁有者、系統管理員

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)	如需詳細資訊，請參閱 AWS 文件中的使用 VPC 中的資料庫執行個體。	系統管理員
建立安全性群組。		系統管理員
設定並啟動 Amazon Redshift 叢集。	如需詳細資訊，請參閱 Amazon Redshift 文件中的建立範例 Amazon Redshift 叢集。	DBA, 系統管理員

移轉資料

任務	描述	所需技能
使用 AWS DMS 從 Microsoft SQL 伺服器資料庫遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA、應用程式擁有者、系統管理員

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉臨時資源。		DBA, 系統管理員
審核並驗證專案文件。		DBA、應用程式擁有者、系統管理員
收集指標，例如移轉時間、手動與自動化工作的百分比，以及節省成本。		DBA、應用程式擁有者、系統管理員
關閉專案並提供意見反應。		DBA、應用程式擁有者、系統管理員

相關資源

參考

- AWS DMS 說明文件
- Amazon Redshift 文檔
- Amazon Redshift 定價

教學課程和影片

- 開始使用 AWS DMS
- 開始使用 Amazon RedShift
- 使用 Amazon Redshift 資料庫做為 AWS Database Migration Service 的目標
- AWS DMS (影片)

使用 AWS SCT 資料擷取代理程式將現場部署 Microsoft SQL 伺服器資料庫遷移至 Amazon Redshift 移

由妮哈·塔庫爾（AWS）創建

環境：PoC 或試點	來源：Microsoft SQL 服務器	目標：Amazon Redshift
R 型：重新建築	工作負載：Microsoft	技術：移轉；資料庫
AWS 服務：Amazon Redshift；AWS SCT		

Summary

此模式概述了使用 AWS Schema Conversion Tool (AWS SCT) 資料擷取代理程式，將現場部署 Microsoft SQL Server 來源資料庫遷移到 Amazon Redshift 目標資料庫的步驟。代理程式是與 AWS SCT 整合的外部程式，但在其他地方執行資料轉換，並代表您與其他 AWS 服務互動。

先決條件和限制

先決條件

- 用於內部部署資料中心中資料倉儲工作負載的 Microsoft SQL Server 來源資料庫
- 有效的 AWS 帳戶

產品版本

- Microsoft SQL 伺服器 2008 年或更高版本。如需支援版本的最新清單，請參閱 AWS SCT 文件。

架構

技術堆疊來源

- 內部部署 Microsoft SQL 伺服器資料庫

技術, 堆, 目標

- Amazon Redshift

資料移轉架構

工具

- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，以處理異質資料庫遷移。當來源和目標資料庫截然不同時，您可以使用 AWS SCT 代理程式執行其他資料轉換。如需詳細資訊，請參閱 AWS 文件中的將資料從現場部署資料倉儲遷移到 Amazon Redshift。

最佳實務

- 適用於 AWS SCT 的最佳實務
- Amazon Redshift 的最佳實踐

史诗

準備移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA SysAdmin
選擇適當的執行個體類型 (容量、儲存功能、網路功能)。		DBA SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA SysAdmin

任務	描述	所需技能
選擇應用程式移轉策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組。		SysAdmin
設定並啟動 Amazon Redshift 叢集。		SysAdmin

移轉資料

任務	描述	所需技能
使用 AWS SCT 資料擷取代理程式移轉資料。		DBA

遷移應用

任務	描述	所需技能
遵循選擇的應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切換到目標數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集指標，例如移轉時間、手動與自動化工作的百分比，以及節省成本。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供任何意見反應。		DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- 使用者指南
- 使用資料擷取代理
- Amazon Redshift 定價

教學課程和影片

- 開始使用 AWS 結 Schema Conversion Tool
- 開始使用 Amazon RedShift

使用 AWS SCT 資料擷取代理程式將 Teradata 資料庫遷移到 Amazon Redshift 移

R 型：重新建築	來源：數據庫：關係	目標：Amazon Redshift
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
AWS 服務：Amazon Redshift		

Summary

此模式會引導您完成將 Teradata 資料庫 (用作現場部署資料中心中的資料倉儲使用) 遷移至 Amazon Redshift 資料庫的步驟。該模式使用 AWS Schema Conversion Tool (AWS SCT) 資料擷取代理程式。代理程式是與 AWS SCT 整合的外部程式，但在其他地方執行資料轉換，並代表您與其他 AWS 服務互動。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 位於內部部署資料中心的 Teradata 來源資料庫

產品版本

- 第 13 版及更高版本。如需支援版本的最新清單，請參閱 AWS SCT 文件。

架構

源, 技術, 堆棧

- 內部部署資料庫

目標技術堆疊

- Amazon Redshift 叢集

資料移轉架構

工具

- AWS SCT — AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，以處理異質資料庫遷移。當來源和目標資料庫彼此之間截然不同時，您可以使用 AWS SCT 代理程式執行其他資料轉換。如需詳細資訊，請參閱 AWS 文件中的將資料從現場部署資料倉儲遷移到 Amazon Redshift。

史诗

準備移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA, SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
選擇適當的執行個體類型 (容量、儲存功能、網路功能)。		DBA, SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA, SysAdmin
選擇應用程式移轉策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組。		SysAdmin
設定並啟動 Amazon Redshift 叢集。		SysAdmin

移轉資料

任務	描述	所需技能
使用 AWS SCT 資料擷取代理程式遷移資料。	如需使用 AWS SCT 資料擷取代理程式的詳細資訊，請參閱參考和說明一節中的連結。	DBA

遷移應用

任務	描述	所需技能
遵循選擇的應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切換到目標 Amazon Redshift 數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集有關遷移時間、手動與工具任務的百分比、節省成本等指標。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供任何意見反應。		

相關資源

參考

- 使用者指南
- 使用資料擷取代理
- Amazon Redshift 定價
- 將功能重設時間轉換為 Amazon Redshift SQL (AWS Prescriptive Guidance)
- 將太數據標準化時間功能轉換為 Amazon Redshift SQL (AWS Prescriptive Guidance)

教學課程

- 開始使用 AWS 結 Schema Conversion Tool
- 開始使用 Amazon RedShift

使用 AWS SCT 資料擷取代理程式將現場部署 Vertica 資料庫遷移到 Amazon Redshift 移

R 型：重新建築	來源：數據庫：關係	目標：Amazon Redshift
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
AWS 服務：Amazon Redshift		

Summary

此模式提供使用 AWS 結構描述轉換工具 (AWS SCT) 資料擷取代理程式將現場部署 Vertica 資料庫移轉至 Amazon Redshift 叢集的指導。代理程式是與 AWS SCT 整合的外部程式，但在其他地方執行資料轉換，並代表您與其他 AWS 服務互動。

先決條件和限制

前提

- 有效的 AWS 帳戶
- 用於內部部署資料中心中資料倉儲工作負載的 Vertica 來源資料庫
- 亞 Amazon Redshift 目標叢集

產品版本

- Vertica 版本 7.2.2 及更高版本。如需支援版本的最新清單，請參閱 AWS SCT 文件。

架構

源, 技術, 堆棧

- 內部部署 Vertica 資料庫

目標技術堆疊

- 亞 Amazon Redshift 集群

資料移轉架構

工具

- AWS SCT-AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，以處理異質資料庫遷移。當來源和目標資料庫彼此之間截然不同時，您可以使用 AWS SCT 代理程式執行其他資料轉換。如需詳細資訊，請參閱 AWS 文件中的將資料從現場部署資料倉儲遷移到 Amazon Redshift。

史诗

準備移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
選擇適當的執行個體類型 (容量、儲存功能、網路功能)。		DBA, SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA, SysAdmin
選擇應用程式移轉策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin

任務	描述	所需技能
建立安全性群組。		SysAdmin
設定並啟動 Amazon Redshift 叢集。		SysAdmin

移轉資料

任務	描述	所需技能
使用 AWS SCT 資料擷取代理程式移轉資料。	如需使用 AWS SCT 資料擷取代理程式的詳細資訊，請參閱參考和說明一節中的連結。	DBA

遷移應用

任務	描述	所需技能
遵循選擇的應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切換到目標數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, SysAdmin

任務	描述	所需技能
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集有關遷移時間、手動與工具任務的百分比、節省成本等指標。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供任何意見反應。		

相關資源

參考

- 使用者指南
- 使用資料擷取代理
- Amazon Redshift 定價

教學課程和影片

- 開始使用 AWS 結 Schema Conversion Tool
- 開始使用 Amazon RedShift

將舊有應用程式從 Oracle Pro*C 移轉至 ECPG

創建者：西帕薩拉迪（AWS）和馬赫斯巴魯穆里（AWS）

環境：PoC 或試點	來源：甲骨文	目標：PostgreSQL
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫

Summary

大多數已內嵌 SQL 程式碼的舊版應用程式都使用 Oracle Pro*C 預先編譯器來存取資料庫。當您將這些 Oracle 資料庫遷移到適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 或 Amazon Aurora PostgreSQL 相容版本時，您必須將應用程式程式碼轉換為與 PostgreSQL 中的預編譯器相容的格式 (稱為 ECPG)。這個模式描述了如何在 PostgreSQL ECPG 中將甲骨文專業 * C 代碼轉換為相應的代碼。

如需 Pro*C 的詳細資訊，請參閱 Oracle 文件集。如需 ECPG 的簡要介紹，請參閱其他資訊一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon RDS for PostgreSQL 或 Aurora 相容資料庫
- 在內部部署執行的 Oracle 資料庫

工具

- 下一節中列出的 PostgreSQL 套件。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可透過命令列殼層中的命令與 AWS 服務互動。只要使用最少的組態，您就可以執行 AWS CLI 命令，從命令提示字元實作與以瀏覽器為基礎的 AWS 管理主控台所提供的功能相同。

史诗

在 CentOS 或 RHEL 上設定構建環境

任務	描述	所需技能
安 PostgreSQL 件。	使用下列命令來安裝必要的 PostgreSQL 套件。 `yum update -y` `yum install -y yum-utils` `rpm -ivh https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm` `dnf -qy module disable postgresql`	應用 DevOps 程式開發人員、
安裝頭文件和庫。	使用下列指令來安裝包含標頭檔案和程式庫的`postgresql12-devel` 套件。在開發環境和執行階段環境中安裝套件，以避免執行階段環境發生錯誤。 `dnf -y install postgresql12-devel` `yum install ncompress zip ghostscript jq unzip wget git -y` 僅針對開發環境，同時執行下列命令。 `yum install zlib-devel make -y`	應用 DevOps 程式開發人員、

任務	描述	所需技能
	`ln -s /usr/pgsql-12/bin/ecpg /usr/bin/`	
設定環境路徑變數。	設定 PostgreSQL 用戶端程式庫的環境路徑。 `export PATH=$PATH:/usr/pgsql-12/bin`	應用 DevOps 程式開發人員、
視需要安裝其他軟體。	如果需要，請在甲骨文中安裝 pGloader 作為替代 SQL * 載入器。 `wget -O /etc/yum.repos.d/pgloader-ccl.repo https://dl.packager.io/srv/opf/pgloader-ccl/master/installer/el/7.repo` `yum install pgloader-ccl -y` `ln -s /opt/pgloader-ccl/bin/pgloader /usr/bin/` 如果您要從 Pro*C 模組呼叫任何 Java 應用程式，請安裝 Java。 `yum install java -y` 安裝螞蟻來編譯 Java 代碼。 `yum install ant -y`	應用 DevOps 程式開發人員、

任務	描述	所需技能
安裝 AWS CLI。	安裝 AWS CLI 以執行命令，以便從您的應用程式與 AWS 秘密管理員和亞馬遜簡單儲存服務 (Amazon S3) 等 AWS 服務互動。 `cd /tmp/` `curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"` `unzip awscliv2.zip` `./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update`	應用 DevOps 程式開發人員、
確定要轉換的程序。	識別您要從 Pro*C 轉換為 ECPG 的應用程式。	應用程序開發人員，應用

將 Pro*C 程式碼轉換為 ECPG

任務	描述	所需技能
刪除不需要的標題。	移除 PostgreSQL 中不需要的`include` 標頭，例如`oci.h`、`oratypes`、和。`sqlda`	應用程式擁有者、應用程
更新變數宣告。	針對用作主機變數的所有變數宣告新增`EXEC SQL`陳述式。 從應用程式中移除下列`EXEC SQL VAR`宣告。	應用程序開發人員，應用

任務	描述	所需技能
	`EXEC SQL VAR query IS STRING(2048);`	

任務	描述	所需技能
更新功能。	該ROWNUM功能在 PostgreSQL 中不可用。將其替換為 SQL 查詢中的ROW_NUMBER 窗口函數。 專業版 * C 代碼： `SELECT SUBSTR(RTRIM(FILE_NAME,'.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME FROM DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2 WHERE ROWNUM <=1 ORDER BY ROWNUM;` 華訊產品代碼： `SELECT SUBSTR(RTRIM(FILE_NAME,'.txt'),12) INTO :gcpclFileseq FROM (SELECT FILE_NAME , ROW_NUMBER() OVER (ORDER BY FILE_NAME DESC) AS ROWNUM FROM demo_schema.DEMO_FILES_TABLE WHERE FILE_NAME LIKE '%POC%' ORDER BY FILE_NAME DESC) FL2`	應用程序開發人員，應用

任務	描述	所需技能
	`WHERE ROWNUM <=1 ORDER BY ROWNUM;`	
更新函數參數以使用別名變量。	在 PostgreSQL 中，函數參數不能用作主機變量。使用別名變數覆寫它們。 專業版 * C 代碼： `int processData(int referenceId){ EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }` 華訊產品代碼： `int processData(int referenceIdParam){ EXEC SQL int referenceId = referenceIdParam; EXEC SQL char col_val[100]; EXEC SQL select column_name INTO :col_val from table_name where col=:referenceId; }`	應用程序開發人員，應用

任務	描述	所需技能
更新結構類型。	typedef如果使用struct類struct型變量作為主機變量，則在EXEC SQL BEGIN和END塊中定義類型。如果struct類型是在 header (.h) 檔案中定義的，請包含 EXEC SQL include 陳述式的檔案。 專業版 * C 代碼： 標頭檔案 (demo.h) `struct s_partiti on_ranges { char sc_table_ group[31]; char sc_table_ name[31]; char sc_range_ value[10]; }; struct s_partiti on_ranges_ind { short ss_table_ group; short ss_table_ name; short ss_range_ value; };` 華訊產品代碼： 標頭檔案 (demo.h)	應用程序開發人員，應用

任務	描述	所需技能
	`EXEC SQL BEGIN DECLARE SECTION;` `typedef struct` `{` `  char  sc_table_group[31];` `  char  sc_table_name[31];` `  char  sc_range_value[10];` `} s_partition_ranges;` `typedef struct` `{` `  short  ss_table_group;` `  short  ss_table_name;` `  short  ss_range_value;` `} s_partition_ranges_ind;` `EXEC SQL END DECLARE SECTION;` **專業版 * C 檔案 () demo.pc** `#include "demo.h"` `struct s_partition_ranges gc_partition_data[MAX_PART_TABLE] ;` `struct s_partition_ranges_ind gc_partition_data_ind[MAX_PART_TABLE] ;` **華訊產品檔案 () demo.pc** `exec sql include "demo.h"`	

任務	描述	所需技能
	`EXEC SQL BEGIN DECLARE SECTION;` `s_partition_ranges gc_partition_data[MAX_PART_TABLE] ;` `s_partition_ranges_ind gc_partition_data_ind[MAX_PART_TABLE] ;` `EXEC SQL END DECLARE SECTION;`	
修改邏輯以從游標擷取。	若要使用陣列變數從游標擷取多個資料列，請變更要使用FETCH FORWARD的程式碼。 專業版 * C 代碼： `EXEC SQL char aPoeFiles[MAX_FILES][FILENAME_LENGTH];` `EXEC SQL FETCH filename_cursor into :aPoeFiles;` 華訊產品代碼： `EXEC SQL char aPoeFiles[MAX_FILES][FILENAME_LENGTH];` `EXEC SQL int fetchSize = MAX_FILES;` `EXEC SQL FETCH FORWARD :fetchSize filename_cursor into :aPoeFiles;`	應用程序開發人員，應用

任務	描述	所需技能
修改沒有傳回值的套件呼叫。	沒有返回值的 Oracle 包函數應該使用指標變量調用。如果您的應用程式包含多個具有相同名稱的函數，或者如果未知型別函數產生執行階段錯誤，請將值類型轉換為資料類型。 專業版 * C 代碼： `void ProcessData (char *data , int id) { EXEC SQL EXECUTE BEGIN pkg_demo. process_data (:data, :id); END; END-EXEC; }` 華訊產品代碼： `void ProcessData (char *dataParam, int idParam ) { EXEC SQL char *data = dataParam; EXEC SQL int id = idParam; EXEC SQL short rowInd; EXEC SQL short rowInd = 0;`	應用程序開發人員，應用

任務	描述	所需技能
	EXEC SQL SELECT pkg_demo.process_data (inp_data => :data::te xt, inp_id => :id) INTO :rowInd; }	

任務	描述	所需技能
重寫游標變數。	重寫SQL_CURSOR 變量及其實現。 專業版 * C 代碼： `/* SQL Cursor */` `SQL_CU` `RSOR demo_cursor;` `EXEC SQL` ` ALLOCATE :demo_cursor;` `EXEC SQL EXECUTE` `  BEGIN` `  pkg_demo.` `get_cursor(` `  demo_cur=` `>:demo_cursor` `  );` `  END;` `END-EXEC;` 華訊產品代碼： `EXEC SQL DECLARE` ` demo_cursor CURSOR FOR` ` SELECT` `  * from` `  pkg_demo.open_file` `name_rc(` `  demo_cur=` `>refcursor` `  ) ;` `EXEC SQL char open_file` `name_rcInd[100];` `# As the below function` ` returns cursor_name` ` as` `# return we need to` ` use char[] type as` ` indicator.`	應用程序開發人員，應用

任務	描述	所需技能
	`EXEC SQL SELECT pkg_demo.get_cursor ( demo_cur=>'demo_cursor' ) INTO :open_filename_rcInd;`	
套用常見的移轉模式。	• 變更 SQL 查詢，使其與 PostgreSQL 相容。 • 在 ECPG 中不支援匿名區塊時，將它們移至資料庫。 • 移除不受 PostgreSQL 支援的`dbms_application_info` 邏輯。 • 在游標關閉之後移動`EXEC SQL COMMIT`語句。如果您在循環中提交查詢以從光標中獲取記錄，則光標被關閉，並顯示光標不存在錯誤。 • 如需有關處理 ECPG 中的例外狀況和錯誤碼的詳細資訊，請參閱 PostgreSQL 文件中的錯誤處理。	應用程序開發人員，應用
視需要啟用偵錯。	要在調試模式下運行 ECPG 程序，請在主功能塊內添加以下命令。 `ECPGdebug(1, stderr);`	應用程序開發人員，應用

編譯 ECPG 程式

任務	描述	所需技能
建立 ECPG 的可執行檔。	如果您有一個名為嵌入式 SQL C 源文件`prog1.pgc` ，則可以通過使用以下命令序列創建一個可執行程序。 `ecpg prog1.pgc` `cc -I/usr/local/pgsql/include -c prog1.c` `cc -o prog1 prog1.o -L/usr/local/pgsql/lib -lecpg`	應用程序開發人員，應用
創建一個用於編譯的 make 文件。	建立 make 檔案以編譯 ECPG 程式，如下列範例檔案所示。 `CFLAGS ::= $(CFLAGS) -I/usr/pgsql-12/include -g -Wall` `LDFLAGS ::= $(LDFLAGS) -L/usr/pgsql-12/lib -Wl,-rpath,/usr/pgsql-12/lib` `LDLIBS ::= $(LDLIBS) -lecpg` `PROGRAMS = test` `.PHONY: all clean` `%.c: %.pgc` `  ecpg $<` `all: $(PROGRAMS)` `clean:` `  rm -f $(PROGRAMS) $(PROGRAMS:%=%.c) $(PROGRAMS:%=%.o)`	應用程序開發人員，應用

測試應用程式。

任務	描述	所需技能
測試代碼。	測試轉換後的應用程式程式碼，以確定其運作正常。	應用程式開發人員、應用程式擁有者、

相關資源

- ECPG-C 語言中的嵌入式 PostgreSQL 文件集)
- 錯誤處理 (文件集)
- 為什麼要使用甲骨文專業版 *C/C ++ 預編譯器（Oracle 文檔）

其他資訊

PostgreSQL 有一個嵌入式 SQL 預編譯器，ECPG，它相當於甲骨文 Pro*C 預編譯器。ECPG 會以特殊函式呼叫取代 SQL 呼叫，將內嵌 SQL 陳述式的 C 程式轉換為標準 C 程式碼。然後可以使用任何 C 編譯器工具鏈處理輸出文件。

輸入和輸出文件

ECPG 會將您在命令列上指定的每個輸入檔案轉換為對應的 C 輸出檔案。如果輸入檔案名稱沒有副檔名，則假設為 .pgc。檔案的副檔名會被取代，`.c`以建構輸出檔案名稱。但是，您可以使用`-o`選項來取代預設輸出檔案名稱。

如果您使用破折號 (-) 作為輸入檔案名稱，ECPG 會從標準輸入讀取程式並寫入標準輸出，除非您使用`-o`選項覆寫該檔案名稱。

頭文件

當 PostgreSQL 編譯器編譯預先處理的 C 程式碼檔案時，它會在 PostgreSQL 目錄中尋找 ECPG 標頭檔案。`include`因此，您可能必須使用`-I`選項將編譯器指向正確的目錄 (例如`-I/usr/local/pgsql/include`)。

Libraries (程式庫)

使用 C 代碼和嵌入式 SQL 的程序必須與`libecpg`庫鏈接。例如，您可以使用連結器選項 `-L/usr/local/pgsql/lib -lecpg`。

轉換後的 ECPG 應用程式會透過內嵌式 SQL 程式`libpq`庫 (`ecpglib`) 呼叫程式庫中的函數，並使用標準的前端/後端通訊協定與 PostgreSQL 伺服器通訊。

將虛擬生成的列從甲骨文遷移到 PostgreSQL

創建者：韋蘭賈納魯格蘭希（AWS），拉傑什·馬蒂瓦勒（AWS）和拉梅什帕瑟里（AWS）

環境：生產	來源：甲骨文數據庫	目標：Amazon RDS for PostgreSQL 或 Aurora 兼容
R 型：重新建築	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon Aurora; Amazon RDS; AWS DMS		

Summary

在版本 11 及更早版本中，PostgreSQL 不提供直接等同於甲骨文虛擬列的功能。從 Oracle 資料庫遷移到 PostgreSQL 版本 11 或更早版本時，處理虛擬產生的資料行很困難，原因有兩個：

- 移轉期間不會顯示虛擬資料行。
- 版本 12 之前的運算式不支援此`generate`運算式。

但是，有一些解決方法可以模擬類似的功能。當您使用 AWS Database Migration Service (AWS DMS) 將資料從 Oracle 資料庫遷移到 PostgreSQL 版本 11 及更早版本時，您可以使用觸發函數將值填入虛擬產生的資料欄中。這種模式提供了甲骨文數據庫和 PostgreSQL 代碼的例子，你可以使用這個目的。在 AWS 上，您可以使用 Amazon Relational Database Service 服務 (Amazon RDS) 適用於 PostgreSQL 資料庫或 Amazon Aurora PostgreSQL 相容版本。

從第 12 版開始，支援產生的資料行。產生的欄可以從其他欄值即時計算，也可以計算和儲存。PostgreSQL 產生的資料行類似於甲骨文虛擬資料行。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 來源甲骨文資料庫
- 目標資料庫 (在亞馬遜 RDS 上適用於 PostgreSQL 或 Aurora 相容)
- 編碼專業知識

限制

- 僅適用於 PostgreSQL 之前的版本。
- 適用於「Oracle 資料庫」版本 11g 或更新版本。
- 資料移轉工具不支援虛擬欄。
- 僅適用於在同一個表格中定義的欄。
- 如果虛擬產生的資料行參照具決定性的使用者定義函數，則無法將其用作分割索引鍵資料欄。
- 運算式的輸出必須是純量值。它不能返回一個 Oracle 提供的數據類型，一個用戶定義的類型LOB，或。LONG RAW
- 針對虛擬資料行定義的索引等同於 PostgreSQL 中以函數為基礎的索引。
- 必須收集表統計信息。

工具

- pgAdmin 4 是一個適用於 PostgreSQL 的開源管理工具。此工具提供圖形介面，可簡化資料庫物件的建立、維護和使用。
- Oracle SQL 開發人員是一個免費的整合式開發環境，可在傳統和雲端部署中使用 Oracle 資料庫中的 SQL。

史诗

建立來源與目標資料庫表格

任務	描述	所需技能
建立來源「Oracle 資料庫」表格。	在「Oracle 資料庫」中，使用下列陳述式建立含有虛擬產生資料行的資料表。 `CREATE TABLE test.generated_column ( CODE NUMBER, STATUS VARCHAR2(12) DEFAULT 'PreOpen', FLAG CHAR(1) GENERATED ALWAYS AS (CASE`	DBA, 應用程式開發人員

任務	描述	所需技能
	<pre>UPPER(STATUS) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) VIRTUAL VISIBLE);</pre> 在此來源表格中，資料行中的資STATUS料會透過 AWS DMS 遷移到目標資料庫。不過，FLAG資料行是透過使用generate by功能填入的，因此 AWS DMS 在遷移期間看不到此欄。若要實作的功能generated by，您必須使用目標資料庫中的觸發程序和函式來填入資料行中的FLAG值，如下一個史詩所示。	
在 AWS 上建立目標資料表。	使用下列陳述式在 AWS 上建立 PostgreSQL 資料表。 <pre>CREATE TABLE test.generated_column (code integer not null, status character varying(12) not null , flag character(1));</pre> 在此表中，該status列是一個標準列。該flag列將根據列中的數據生成的status列。	DBA, 應用程式開發人員

創建一個觸發函數來處理 PostgreSQL 中的虛擬列

任務	描述	所需技能
建立一個 PostgreSQL 程序。	在 PostgreSQL 中，創建一個觸發器。 `CREATE TRIGGER tgr_gen_column AFTER INSERT OR UPDATE OF status ON test.generated_column FOR EACH ROW EXECUTE FUNCTION test.tgf_gen_column();`	DBA, 應用程式開發人員
創建一個觸 PostgreSQL 函數。	在 PostgreSQL 中，為觸發器創建一個函數。此函數會填入由應用程式或 AWS DMS 插入或更新的虛擬資料欄，並驗證資料。 `CREATE OR REPLACE FUNCTION test.tgf_gen_column() RETURNS trigger AS $VIRTUAL_COL$ BEGIN IF (TG_OP = 'INSERT') THEN IF (NEW.flag IS NOT NULL) THEN RAISE EXCEPTION 'ERROR: cannot insert into column "flag"' USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF;`	DBA, 應用程式開發人員

任務	描述	所需技能
	IF (TG_OP = 'UPDATE') THEN IF (NEW.flag::VARCHAR ! = OLD.flag::varchar) THEN RAISE EXCEPTION 'ERROR: cannot update column "flag"' USING DETAIL = 'Column "flag" is a generated column.'; END IF; END IF; IF TG_OP IN ('INSERT' ,'UPDATE') THEN IF (old.flag is NULL) OR (coalesce(old.stat us,'') != coalesce(new.status,'')) THEN UPDATE test.gene rated_column SET flag = (CASE UPPER(status) WHEN 'OPEN' THEN 'N' ELSE 'Y' END) WHERE code = new.code; END IF; END IF; RETURN NEW; END $VIRTUAL_COL$ LANGUAGE plpgsql;	

使用 AWS DMS 測試資料遷移

任務	描述	所需技能
建立複寫執行個體。	若要建立複寫執行個體，請遵循 AWS DMS 文件中的指示。複寫執行個體應與來源和目標	DBA, 應用程式開發人員

任務	描述	所需技能
	資料庫位於相同的虛擬私有雲 (VPC) 中。	
建立來源端點和目標端點。	若要建立端點，請遵循 AWS DMS 文件中的指示。	DBA, 應用程式開發人員
測試端點連線。	您可以指定 VPC 和複寫執行個體，然後選擇 [執行測試] 來測試端點連線。	DBA, 應用程式開發人員
建立並啟動完整載入工作。	如需指示，請參閱 AWS DMS 文件中的建立任務和全載任務設定。	DBA, 應用程式開發人員
驗證虛擬資料行的資料。	比較來源和目標資料庫中虛擬資料行中的資料。您可以手動驗證資料，或為此步驟撰寫指令碼。	DBA, 應用程式開發人員

相關資源

- 開始使用 AWS Database Migration Service (AWS DMS 文件)
- 使用 Oracle 資料庫做為 AWS DMS 的來源 (AWS DMS 文件)
- 使用 PostgreSQL 資料庫做為 AWS DMS 的目標 (AWS DMS 文件)
- 在 PostgreSQL 中產生的資料欄 (文件)
- 觸發函數 (PostgreSQL 集)
- Oracle 資料庫中的虛擬資料欄 (Oracle 文件集)

在 Aurora 相容上設定甲骨文 UTL_FILE 功能

由拉凱什拉格夫 (AWS) 和阿努拉達 (AWS) 創建

環境：PoC 或試點	來源：甲骨文	目標：Aurora
R 型：重新建築	工作量：甲骨文	技術：移轉；基礎架構；資料庫
AWS 服務：Amazon S3；Amazon Aurora		

Summary

在亞馬遜 Amazon Web Services (AWS) 雲端上從甲骨文到 Amazon Aurora PostgreSQL 相容版本的遷移過程中，您可能會遇到多個挑戰。例如，移轉依賴 Oracle UTL_FILE 公用程式的程式碼永遠是一項挑戰。在 Oracle PL/SQL 中，UTL_FILE套裝程式會與相關作業系統一起用於檔案作業 (例如讀取和寫入)。該UTL_FILE實用程序適用於服務器和客戶端計算機系統。

Amazon Aurora PostgreSQL 相容於受管資料庫產品。因此，無法訪問數據庫服務器上的文件。此模式會引導您完成整合亞馬遜簡單儲存服務 (Amazon S3) 和 Amazon Aurora PostgreSQL 相容，以實現一部分功能。UTL_FILE使用此集成，我們可以創建和使用文件，而無需使用第三方提取，轉換和加載（ETL）工具或服務。

您也可以選擇設定 Amazon CloudWatch 監控和 Amazon SNS 通知。

我們建議您在生產環境中實作此解決方案之前，先徹底測試此解決

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Database Migration Service (AWS DMS) 專業知識
- PL/PGSQL 編碼方面的專業知識
- Amazon Aurora 與 PostgreSQL 相容的叢集

- S3 儲存貯體

限制

此模式不提供用來替代 Oracle UTL_FILE 公用程式的功能。但是，可以進一步增強步驟和示例代碼，以實現數據庫現代化目標。

產品版本

- Amazon Aurora 郵政兼容版 11.9

架構

目標技術堆疊

- Amazon Aurora 郵政兼容
- Amazon CloudWatch
- Amazon Simple Notification Service (Amazon SNS)
- Amazon S3

目標架構

下圖顯示解決方案的高階表示法。

1. 檔案會從應用程式上傳到 S3 儲存貯體。
2. 該aws_s3擴展訪問數據，使用 PL/pgSQL，並將數據上傳到 Aurora PostgreSQL 兼容。

工具

- 與 Amazon Aurora PostgreSQL 相容 — 與 Amazon Aurora PostgreSQL 相容的版本是完全受管、與 PostgreSQL 相容且符合 ACID 標準的關聯式資料庫引擎。它結合了高階商業資料庫的速度和可靠性，以及開放原始碼資料庫的成本效益。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是管理 AWS 服務的統一工具。只要下載和設定一個工具，您就可以從命令列控制多個 AWS 服務，並透過指令碼自動化這些服務。

- Amazon CloudWatch — Amazon CloudWatch 監控 Amazon S3 資源和使用。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。在這種模式中，Amazon S3 提供了一個儲存層，用於接收和存放檔案，以便在與 Aurora PostgreSQL 相容的叢集之間進行使用和傳輸。
- aws_s3 — 此aws_s3擴充功能整合了與 Amazon S3 和 Aurora 相容的功能。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和用戶端之間訊息的交付或傳送。在這種模式中，Amazon SNS 用於發送通知。
- PGAdmin — PGAdmin 是 Postgres 的開放原始碼管理工具。pgAdmin 4 提供了一個圖形介面，用於建立、維護和使用資料庫物件。

Code

為了實現所需的功能，該模式創建與命名類似的多個函數UTL_FILE。其他資訊區段包含這些函式的程式碼庫。

在代碼中，testaurorabucket用測試 S3 存儲桶的名稱替換。us-east-1以測試 S3 儲存貯體所在的 AWS 區域取代。

史诗

整合 Amazon S3 和 Aurora 兼容

任務	描述	所需技能
設定 IAM 政策。	建立 AWS Identity and Access Management (IAM) 政策，以授與 S3 儲存貯體及其中物件的存取權。如需程式碼，請參閱「其他資訊」一節。	AWS 管理員，DBA
將 Amazon S3 存取角色 PostgreSQL 至 Aurora	建立兩個 IAM 角色：一個角色用於讀取，另一個角色用於寫入 Amazon S3。將這兩個角色連接到與 Aurora PostgreSQL 相容的叢集： • S3 匯出功能的一個角色	AWS 管理員，DBA

任務	描述	所需技能
	• S3 匯入功能的一個角色 如需詳細資訊，請參閱 Aurora PostgreSQL 相容文件，了解如何將資料匯入和匯出到 Amazon S3。	

在相容 Aurora 中設定擴充功能

任務	描述	所需技能
創建 aws_commons 擴展。	`aws_commons` 擴充功能是擴充功能的`aws_s3`相依性。	DBA, 開發人員
建立 aws_s3 延伸模組。	該`aws_s3`擴展與 Amazon S3 交互。	DBA, 開發人員

驗證 Amazon S3 和 Aurora 相容於 PostgreSQL 的整合

任務	描述	所需技能
測試將檔案從 Amazon S3 匯入 Aurora。	若要測試將檔案匯入 Aurora PostgreSQL 相容，請建立範例 CSV 檔案並將其上傳到 S3 儲存貯體。根據 CSV 檔案建立資料表定義，並使用`aws_s3.table_import_from_s3` 函數將檔案載入資料表。	DBA, 開發人員
測試將檔案從 Aurora 匯出到 Amazon S3。	若要測試從 Aurora PostgreSQL 相容匯出檔案，請建立測試資料表，將資料填入資料，然後使用函數匯出資	DBA, 開發人員

任務	描述	所需技能
	料。aws_s3.query_export_to_s3	

若要模擬 UTL_FILE 公用程式，請建立包裝函式

任務	描述	所需技能
建立檔案公用程式結構描述。	該模式將包裝函數保持在一起。若要建立結構描述，請執行下列命令。 `CREATE SCHEMA utl_file_utility;`	DBA, 開發人員
創建文件類型類型。	若要建立file_type 類型，請使用下列程式碼。 `CREATE TYPE utl_file_utility.file_type AS ( p_path character varying(30), p_file_name character varying );`	DBA/開發人員
創建初始化函數。	該init函數初始化常見的變量，如bucket或region。如需程式碼，請參閱「其他資訊」一節。	DBA/開發人員
創建包裝函數。	建立包裝函數fopen、put_line、和fclose。如需程式碼，請參閱「其他資訊」一節。	DBA, 開發人員

測試包裝函數

任務	描述	所需技能
在寫入模式下測試包裝函數。	若要在寫入模式中測試包裝函式，請使用 [其他資訊] 區段中提供的程式碼。	DBA, 開發人員
在追加模式下測試包裝函數。	若要在附加模式中測試包裝函式，請使用其他資訊一節中提供的程式碼。	DBA, 開發人員

相關資源

- Amazon S3 整合
- Amazon Simple Storage Service (Amazon S3)
- Aurora
- Amazon CloudWatch
- Amazon SNS

其他資訊

設定 IAM 政策

建立下列原則。

策略名稱

S3 IntRead

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3integrationtest
",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
```

```
                "s3:ListBucket"
            ],
            "Resource": [
         "arn:aws:s3:::testaurorabuc
ket/*",
         "arn:aws:s3:::testaurorabuc
ket"
            ]
        }
    ]
}
```

S3 IntWrite

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3integrationtest
",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",

                "s3:ListBucket"
            ],
            "Resource": [
    "arn:aws:s3:::testaurorabucket/
*",                "arn:aws:s3:::test
aurorabucket"
            ]
        }
    ]
}
```

創建初始化函數

若要初始化常用變數，例如bucket或region，請使用下列程式碼建立init函式。

```
CREATE OR REPLACE FUNCTION utl_file_utility.init(
    )
    RETURNS void
    LANGUAGE 'plpgsql'
```

```
    COST 100
    VOLATILE
AS $BODY$
BEGIN
      perform set_config
      ( format( '%s.%s','UTL_FILE_UTILITY', 'region' )
      , 'us-east-1'::text
      , false );

      perform set_config
      ( format( '%s.%s','UTL_FILE_UTILITY', 's3bucket' )
      , 'testaurorabucket'::text
      , false );
END;
$BODY$;
```

創建包裝函數

建立fopenput_line、和fclose包裝函式。

打開

```
CREATE OR REPLACE FUNCTION utl_file_utility.fopen(
    p_file_name character varying,
    p_path character varying,
    p_mode character DEFAULT 'W'::bpchar,
    OUT p_file_type utl_file_utility.file_type)
    RETURNS utl_file_utility.file_type
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
declare
    v_sql character varying;
    v_cnt_stat integer;
    v_cnt integer;
    v_tabname character varying;
    v_filewithpath character varying;
    v_region character varying;
    v_bucket character varying;

BEGIN
```

```
   /*initialize common variable */
   PERFORM utl_file_utility.init();
   v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' ) );
   v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' ) );

   /* set tabname*/
   v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
length(p_file_name) else strpos(p_file_name,'.') - 1 end );
   v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
concat_ws('/',p_path,p_file_name) end ;
   raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
v_region;

   /* APPEND MODE HANDLING; RETURN EXISTING FILE DETAILS IF PRESENT ELSE CREATE AN
EMPTY FILE */
   IF p_mode = 'A' THEN
       v_sql := concat_ws('','create temp table if not exists ', v_tabname,' (col1
text)');
       execute v_sql;

       begin
       PERFORM aws_s3.table_import_from_s3
           ( v_tabname,
           '',
           'DELIMITER AS ''#''',
           aws_commons.create_s3_uri
           (     v_bucket,
               v_filewithpath ,
               v_region)
           );
       exception
           when others then
            raise notice 'File load issue ,%',sqlerrm;
            raise;
       end;
       execute concat_ws('','select count(*) from ',v_tabname) into v_cnt;

       IF v_cnt > 0
       then
           p_file_type.p_path := p_path;
           p_file_type.p_file_name := p_file_name;
       else
           PERFORM aws_s3.query_export_to_s3('select ''''',
```

```
                            aws_commons.create_s3_uri(v_bucket, v_filewithpath,
 v_region)
                              );

            p_file_type.p_path := p_path;
            p_file_type.p_file_name := p_file_name;
        end if;
        v_sql := concat_ws('','drop table ', v_tabname);
        execute v_sql;
    ELSEIF p_mode = 'W' THEN
            PERFORM aws_s3.query_export_to_s3('select ''''',
                            aws_commons.create_s3_uri(v_bucket, v_filewithpath,
 v_region)
                              );
            p_file_type.p_path := p_path;
            p_file_type.p_file_name := p_file_name;
    END IF;

EXCEPTION
        when others then
            p_file_type.p_path := p_path;
            p_file_type.p_file_name := p_file_name;
            raise notice 'fopenerror,%',sqlerrm;
            raise;
END;
$BODY$;
```

推桿線

```
CREATE OR REPLACE FUNCTION utl_file_utility.put_line(
    p_file_name character varying,
    p_path character varying,
    p_line text,
    p_flag character DEFAULT 'W'::bpchar)
    RETURNS boolean
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
/*****************************************************************************
* Write line, p_line in windows format to file, p_fp - with carriage return
* added before new line.
```

```
**************************************************************************/
declare
    v_sql varchar;
    v_ins_sql varchar;
    v_cnt INTEGER;
    v_filewithpath character varying;
    v_tabname  character varying;
    v_bucket character varying;
    v_region character varying;

BEGIN
 PERFORM utl_file_utility.init();

/* check if temp table already exist */

 v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
 length(p_file_name) else strpos(p_file_name,'.') - 1 end );

 v_sql := concat_ws('','select count(1) FROM pg_catalog.pg_class c LEFT JOIN
 pg_catalog.pg_namespace n ON n.oid = c.relnamespace where n.nspname like ''pg_temp_
%'''
                         ,' AND pg_catalog.pg_table_is_visible(c.oid) AND
 Upper(relname) = Upper( '''
                         ,  v_tabname ,''' ) ');

 execute v_sql into v_cnt;

  IF v_cnt = 0 THEN
         v_sql := concat_ws('','create temp table ',v_tabname,' (col text)');
        execute v_sql;
        /* CHECK IF APPEND MODE */
        IF upper(p_flag) = 'A' THEN
            PERFORM utl_file_utility.init();
            v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
 'region' ) );
            v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY',
 's3bucket' ) );

            /* set tabname*/
            v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
 concat_ws('/',p_path,p_file_name) end ;

            begin
               PERFORM aws_s3.table_import_from_s3
```

```
                    ( v_tabname,
                         '',
                      'DELIMITER AS ''#''',
                       aws_commons.create_s3_uri
                          ( v_bucket,
                              v_filewithpath,
                              v_region    )
                    );
            exception
                when others then
                    raise notice  'Error Message : %',sqlerrm;
                    raise;
            end;
        END IF;
    END IF;
    /* INSERT INTO TEMP TABLE */
    v_ins_sql := concat_ws('','insert into ',v_tabname,' values(''',p_line,''')');
    execute v_ins_sql;
    RETURN TRUE;
    exception
            when others then
                raise notice  'Error Message : %',sqlerrm;
                raise;
END;
$BODY$;
```

fclose

```
CREATE OR REPLACE FUNCTION utl_file_utility.fclose(
    p_file_name character varying,
    p_path character varying)
    RETURNS boolean
    LANGUAGE 'plpgsql'

    COST 100
    VOLATILE
AS $BODY$
DECLARE
    v_filewithpath character varying;
    v_bucket character varying;
    v_region character varying;
    v_tabname character varying;
    v_sql character varying;
```

```
BEGIN
      PERFORM utl_file_utility.init();

    v_region := current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 'region' ) );
    v_bucket :=  current_setting( format( '%s.%s', 'UTL_FILE_UTILITY', 's3bucket' ) );

    v_tabname := substring(p_file_name,1,case when strpos(p_file_name,'.') = 0 then
 length(p_file_name) else strpos(p_file_name,'.') - 1 end );
    v_filewithpath := case when NULLif(p_path,'') is null then p_file_name else
 concat_ws('/',p_path,p_file_name) end ;

    raise notice 'v_bucket %, v_filewithpath % , v_region %', v_bucket,v_filewithpath,
 v_region ;

    /* exporting to s3 */
    perform aws_s3.query_export_to_s3
        (concat_ws('','select * from ',v_tabname,'  order by ctid asc'),
            aws_commons.create_s3_uri(v_bucket, v_filewithpath, v_region)
        );
    v_sql := concat_ws('','drop table ', v_tabname);
    execute v_sql;
    RETURN TRUE;
EXCEPTION
       when others then
     raise notice 'error fclose %',sqlerrm;
     RAISE;
END;
$BODY$;
```

測試您的設置和包裝函數

使用下列匿名程式碼區塊來測試您的設定。

測試寫入模式

下列程式碼會寫入 S3 儲存貯體`s3inttest`中名為的檔案。

```
do $$
declare
l_file_name varchar := 's3inttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'W';
l_fs utl_file_utility.file_type ;
l_status boolean;
```

```
begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;

select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket: for
 test purpose', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;

end;
$$
```

測試追加模式

下列程式碼會將行附加到先前測試中建立的s3inttest檔案。

```
do $$
declare
l_file_name varchar := 's3inttest' ;
l_path varchar := 'integration_test' ;
l_mode char(1) := 'A';
l_fs utl_file_utility.file_type ;
l_status boolean;

begin
select * from
utl_file_utility.fopen( l_file_name, l_path , l_mode ) into l_fs ;
raise notice 'fopen : l_fs : %', l_fs;


select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket: for
 test purpose : append 1', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;

select * from
utl_file_utility.put_line( l_file_name, l_path ,'this is test file:in s3bucket : for
 test purpose : append 2', l_mode ) into l_status ;
raise notice 'put_line : l_status %', l_status;
```

```
select * from utl_file_utility.fclose( l_file_name , l_path ) into l_status ;
raise notice 'fclose : l_status %', l_status;

end;
$$
```

Amazon SNS 通知

或者，您可以在 S3 儲存貯體上設定亞馬遜 CloudWatch 監控和 Amazon SNS 通知。如需詳細資訊，請參閱監控 Amazon S3 和設定 Amazon SNS 通知。

從甲骨文遷移到 Amazon Aurora PostgreSQL 後驗證數據庫對象

創建者：文卡特拉瑪納奇塔（AWS）和愛德華多·瓦倫蒂姆（AWS）

R 型：重新建築	來源：關係	目標：Amazon Aurora，亞馬遜 RDS
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作量：甲骨文	AWS 服務：Amazon Aurora	

Summary

此模式描述了將 Oracle 資料庫遷移到 Amazon Aurora PostgreSQL 相容版本後驗證物件的 step-by-step 方法。

此模式概述了資料庫物件驗證的使用情境和步驟；如需詳細資訊，請參閱 AWS 資料庫部落格上的使用 AWS SCT 和 AWS DMS 遷移後驗證資料庫物件。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已移轉至 Aurora PostgreSQL 相容資料庫的內部部署 Oracle 資料庫。
- 已套用 AmazonRDS DataFullAccess 原則的登入認證，適用於與 Aurora PostgreSQL 相容的資料庫。
- 此模式使用 Aurora 無伺服器資料庫叢集的查詢編輯器，該叢集可在 Amazon Relational Database Service (Amazon RDS) 主控台取得。但是，您可以將此模式與任何其他查詢編輯器搭配使用。

限制

- 甲骨文同義詞對象在 PostgreSQL 中不可用，但可以通過視圖或設置搜索路徑查詢進行部分驗證。
- Amazon RDS 查詢編輯器僅適用於特定 AWS 區域以及某些 MySQL 和 Postgre SQL 版本。

架構

工具

工具

- 與 Amazon Aurora PostgreSQL 相容的版本 — Aurora PostgreSQL 相容於完全受管、與 PostgreSQL 相容且符合 ACID 標準的關聯式資料庫引擎，結合了高階商業資料庫的速度和可靠性，以及開放原始碼資料庫的簡易性和成本效益。
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。其能為產業標準的關聯式資料庫提供具成本效益、可調整大小的容量，並管理常見的資料庫管理任務。
- Aurora 查詢編輯器分別版 — 查詢編輯器可協助您在 Amazon RDS 主控台中執行 SQL 查詢。您可以在 Aurora 無伺服器資料庫叢集上執行任何有效的 SQL 陳述式，包括資料操作和資料定義陳述式。

要驗證對象，請使用「附件」部分中「對象驗證腳本」文件中的完整腳本。請使用下表作為參考。

甲骨文對象	要使用的腳本
套件	查詢 1
資料表	查詢 3
檢視	查詢條件 5
序列	查詢條件 7
觸發	查詢條件 9
主索引鍵	查詢條件 11
索引	查詢條件 13
檢查限制	查詢十五
外部索引鍵	查詢條件 17

PostgreSQL 对象	要使用的腳本

套件	查詢 2
資料表	查詢條件 4
檢視	查詢條件 6
序列	查詢條件 8
觸發	查詢條件十
主索引鍵	查詢條件 12
索引	查詢條件 14
檢查限制	查詢條款 16
外部索引鍵	查詢條件 18

史诗

驗證來源 Oracle 資料庫中的物件

任務	描述	所需技能
在來源 Oracle 資料庫中執行「套件」驗證查詢。	從「附件」部分下載並打開「對象驗證腳本」文件。透過用戶端程式 Connect 線至來源 Oracle 資料庫。從「對象驗證腳本」文件運行「查詢 1」驗證腳本。重要事項：在查詢中輸入您的 Oracle 使用者名稱，而非「your_schema」。請務必記錄您的查詢結果。	開發人員, DBA
運行「表」驗證查詢。	從「對象驗證腳本」文件運行「查詢 3」腳本。請務必記錄您的查詢結果。	開發人員, DBA

任務	描述	所需技能
運行「視圖」驗證查詢。	從「對象驗證腳本」文件運行「查詢 5」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「序列」計數驗證。	從「對象驗證腳本」文件運行「查詢 7」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「觸發器」驗證查詢。	從「對象驗證腳本」文件運行「查詢 9」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「主鍵」驗證查詢。	從「對象驗證腳本」文件運行「查詢 11」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「索引」驗證查詢。	從「對象驗證腳本」文件運行「查詢 13」驗證腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「檢查約束」驗證查詢。	從「對象驗證腳本」文件運行「查詢 15」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「外鍵」驗證查詢。	從「對象驗證腳本」文件運行「查詢 17」驗證腳本。請務必記錄您的查詢結果。	開發人員, DBA

驗證目標 Aurora 與 PostgreSQL 相容資料庫中的物件

任務	描述	所需技能
使用查詢編輯器 Connect 至目標 Aurora PostgreSQL 相容資料庫。	登入 AWS 管理主控台並開啟 Amazon RDS 主控台。在右上角，選擇您在其中建立與 Aurora PostgreSQL 相容資料	開發人員, DBA

任務	描述	所需技能
	庫的 AWS 區域。在導覽窗格中，選擇「資料庫」，然後選擇與 Aurora PostgreSQL 相容的目標資料庫。在「動作」中，選擇「查詢」。重要事項：如果您之前沒有連線到資料庫，則會開啟「Connect 到資料庫」頁面。然後，您需要輸入資料庫資訊，例如使用者名稱和密碼。	
運行「包」驗證查詢。	從「附件」部分中的「對象驗證腳本」文件運行「查詢 2」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「表」驗證查詢。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 4」指令碼。請務必記錄您的查詢結果。	開發人員, DBA
運行「視圖」驗證查詢。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 6」指令碼。請務必記錄您的查詢結果。	開發人員, DBA
運行「序列」計數驗證。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 8」指令碼。請務必記錄您的查詢結果。	開發人員, DBA

任務	描述	所需技能
運行「觸發器」驗證查詢。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 10」指令碼。請務必記錄您的查詢結果。	開發人員, DBA
運行「主鍵」驗證查詢。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 12」指令碼。請務必記錄您的查詢結果。	開發人員, DBA
運行「索引」驗證查詢。	返回至 Aurora PostgreSQL 相容資料庫的查詢編輯器，並從「物件驗證指令碼」檔案執行「查詢 14」指令碼。請務必記錄您的查詢結果。	開發人員, DBA
運行「檢查約束」驗證查詢。	從「對象驗證腳本」文件運行「查詢 16」腳本。請務必記錄您的查詢結果。	開發人員, DBA
運行「外鍵」驗證查詢。	從「對象驗證腳本」文件運行「查詢 18」驗證腳本。請務必記錄您的查詢結果。	開發人員, DBA

比較來源和目標資料庫驗證記錄

任務	描述	所需技能
比較並驗證兩個查詢結果。	比較甲骨文和 Aurora PostgreSQL 相容資料庫的查詢結果，以驗證所有物件。如果它們都匹配，則所有對象都已成功驗證。	開發人員, DBA

相關資源

- 使用 AWS SCT 和 AWS DMS 在遷移後驗證資料庫物件
- Amazon Aurora 功能：兼容 PostgreSQL 的版本

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

重新主持

主題

- 加速 Microsoft 工作負載探索和移轉到 AWS
- 在 Windows 上自動執行 AWS Managed Services 的工作負載前擷取活動
- 在將主機移轉至 AWS 期間建立防火牆請求的核准程序
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 使用日誌傳送將 LUW 的 Db2 移轉至 Amazon EC2，以減少中斷時間
- 透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2
- 使用 HCX 自動化功能移轉 VMware 虛擬機器
- 將 F5 大 IP 工作負載遷移到 AWS 雲端上的 F5 大 IP VE
- 使用二進位方法將現場部署 Go Web 應用程式遷移到 AWS Elastic Beanstalk
- 使用適用於 SFTP 的 AWS 轉移，將現場部署 SFTP 伺服器遷移到 AWS
- 使用 AWS 應用程式遷移服務將現場部署虛擬機器遷移到 Amazon EC2
- 使用 AWS SFTP 將小型資料集從現場部署遷移到 Amazon S3
- 從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk
- 將現場部署 Oracle 資料庫遷移到 Amazon EC2 上的甲骨文
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到 Amazon EC2
- 將現場部署 SAP ASE 資料庫遷移至 Amazon EC2
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon EC2
- 將現場部署 MySQL 資料庫遷移到 Amazon EC2
- 使用應用程式遷移服務減少同質 SAP 移轉切換時間
- 在 AWS 雲端重新託管現場部署工作負載：移轉檢查清單
- 使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施
- 使用 BMC 探索查詢擷取移轉資料以進行移轉規劃

加速 Microsoft 工作負載探索和移轉到 AWS

創建者阿里·阿爾贊德

環境：生產	來源：執行內部部署或其他雲端服務提供者的 Microsoft 工作負載	目標：Amazon EC2 窗口
R 類型：重新主機	工作量：Microsoft	技術：遷移
AWS 服務：Amazon EC2		

Summary

此模式說明如何使用移轉驗證器工具 PowerShell 組模組探索 Microsoft 工作負載並將其遷移到 AWS。此模組的運作方式是針對與任何 Microsoft 工作負載相關聯的一般任務，執行多次檢查和驗證。例如，模組會檢查可能有多個磁碟連接的執行個體，或是使用多個 IP 位址的執行個體。如需模組可執行之檢查的完整清單，請參閱模組 GitHub 頁面上的檢查一節。

移轉驗證器工具組 PowerShell 模組可協助您的組織減少探索 Microsoft 工作負載上正在執行哪些應用程式和服務所需的時間和精力。此模組也可助您識別工作負載的組態，以便瞭解 AWS 是否支援您的組態。此模組也會提供後續步驟和緩解動作的建議，以避免在遷移之前、期間或之後出現任何組態錯誤。

先決條件和限制

先決條件

- 本機管理員帳戶
- PowerShell 4.0

限制

- 僅適用於 Microsoft 視窗伺服器 2012 R2 或更高版本

工具

工具

- PowerShell 4.0

代碼存儲庫

此模式的移轉驗證器工具 PowerShell 組模組可在 GitHub migration-validator-toolkit-for-micro soft 工作負載儲存庫中取得。

史诗

在單一目標上執行移轉驗證程式工具 PowerShell 組模組

任務	描述	所需技能
下載、擷取、匯入和叫用模組。	選擇下列其中一種方法來下載並部署模組： • 執行指 PowerShell 令碼 • 下載並解壓縮 .zip 檔案 • 克隆存 GitHub 儲庫 執行指 PowerShell 令碼 在中 PowerShell，執行下列範例程式碼： `#MigrationValidatorToolkit` `$uri = 'https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads/archive/refs/heads/main.zip'` `$destination = (Get-Location).Path` `if ((Test-Path -Path "$destination\MigrationValidatorToolkit.zip" -PathType`	系統管理員

任務	描述	所需技能
	Leaf) -or (Test-Path -Path "$destination\MigrationValidatorToolkit")) { write-host "File $destination\MigrationValidatorToolkit.zip or folder $destination\MigrationValidatorToolkit found, exiting" }else { Write-host "Enable TLS 1.2 for this PowerShell session only." [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12 $webClient = New-Object System.Net.WebClient Write-host "Downloading MigrationValidatorToolkit.zip" $webClient.DownloadFile($uri, "$destination\MigrationValidatorToolkit.zip") Write-host "MigrationValidatorToolkit.zip download successfully" Add-Type -Assembly "system.io.compression.filesystem" [System.IO.Compression.ZipFile]::ExtractToDirectory("$destination\Migrat	

任務	描述	所需技能
	ionValidatorToolki t.zip","$destinati on\MigrationValida torToolkit") Write-host "Extracting Migration ValidatorToolkit.zip complete successfully" Import-Module "$destination\Migr ationValidatorToolkit \migration-validator- toolkit-for-microsoft -workloads-main\Mi grationValidatorTo olkit.psm1"; Invoke- MigrationValidatorTo olkit } 程式碼會從 .zip 檔案下載模組。然後，程式碼會擷取、匯入和叫用模組。 下載並解壓縮 .zip 檔案 1. 下載 .zip 檔案 (下載)。 2. 解壓縮 .zip 檔案。 3. 請按照本指南的手動調用模塊故事中的步驟進行操作。 克隆存 GitHub 儲庫 1. 若要複製 GitHub migration-validator-toolkit-for-microsoft 工作負載儲存庫，請在終端機視窗中執行下列 Git 命令：	

任務	描述	所需技能
	`` git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git ``` 2. 請按照本指南的手動調用模塊故事中的步驟進行操作。	

任務	描述	所需技能
手動呼叫模組。	1. 轉到存儲下載模塊的目錄。 2. 若要產生您選擇的輸出，請在中以管理員身分執行下列其中一個指令 PowerShell： 格式表格式： `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit` 格式列表格式： `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -List` 輸出GridView格式： `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -GridView` ConvertTo-CSV 格式： `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-MigrationValidatorToolkit -csv`	系統管理員

在多個目標上執行移轉驗證程式工具 PowerShell 組模組

任務	描述	所需技能
下載 .zip 檔案或複製 GitHub 存放庫。	請選擇下列其中一個選項： • 下載壓縮文件。 (下載). • 若要複製 GitHub migration-validator-toolkit-for-microsoft 工作負載儲存庫，請在終端機視窗中執行下列 Git 命令： `git clone https://github.com/aws-samples/migration-validator-toolkit-for-microsoft-workloads.git`	系統管理員
更新 server.csv 清單。	如果您已下載 .zip 檔案，請依照下列步驟執行： 1. 解壓縮 .zip 檔案。 2. 前往 `Migration ValidatorToolkit\Inputs\` 目錄。 3. `serverlist.csv` 使用目標電腦的主機名稱進行更新。	系統管理員
調用模塊。	您可以使用網域中使用具有目標電腦管理員存取權的網域使用者的任何電腦。 1. 將原始程式碼下載為 .zip 檔案並解壓縮檔案。	系統管理員

任務	描述	所需技能
	2. 以中 PowerShell的管理員身分執行下列命令： `Import-Module .\MigrationValidatorToolkit.psm1;Invoke-DomainComputers` 輸出的 .csv 檔案會`MigrationValidatorToolkit\Outputs\folder` 以字首名稱`DomainComputers_MigrationAutomations_YYYY-MM-DDTHH-MM-SS` 儲存在中。	

故障診斷

問題	解決方案
`MigrationValidatorToolkit` 將有關執行、命令和錯誤的資訊寫入執行中主機上的記錄檔。	您可以在下列位置手動檢視記錄檔： 1. 前往 `MigrationValidatorToolkit\logs\` 目錄。 2. 找到記錄檔。記錄檔名稱的格式為：`ComputerName_MigrationValidatorToolkit_YYYY-MM-SSTHH-MM-SS.log`

相關資源

- 將 Microsoft 工作負載遷移到 AWS 的選項、工具和最佳實務 (AWS Prescriptive Guidance)
- Microsoft 遷移模式 (AWS Prescriptive Guidance)

- AWS 上的免費雲端遷移服務 (AWS 文件)
- 預先定義的啟動後動作 (應用程式行銷文件

其他資訊

常見問答集

我在哪裡可以運行遷移驗證器工具包 PowerShell 模塊？

您可以在 Microsoft 視窗伺服器 2012 R2 或更高版本上運行該模組。

我什麼時候運行這個模塊？

建議您在移轉過程的評估階段執行模組。

模組是否會修改我現有的伺服器？

沒有 此模組中的所有動作都是唯讀的。

運行該模塊需要多長時間？

運行該模塊通常需要 1-5 分鐘，但這取決於服務器的資源分配。

模組需要執行哪些權限？

您必須從本機管理員帳戶執行模組。

我可以在實體伺服器上執行模組嗎？

是的，只要操作系統是 Microsoft 視窗服務器 2012 R2 或更高版本。

如何針對多部伺服器大規模執行模組？

若要在多部加入網域的電腦上大規模執行模組，請依照本指南在多個目標上執行移轉驗證程式工具 PowerShell 組模組中的步驟執行。對於未加入網域的電腦，請使用遠端叫用或在本指南的單一目標上執行移轉驗證工具 PowerShell 組模組中的步驟，在本機執行模組。

在 Windows 上自動執行 AWS Managed Services 的工作負載前擷取活動

由張雅各 (AWS)、葉嘉文 (AWS) 和德韋恩·波德隆 (AWS) 創建

代碼存儲庫：GitHub	環境：生產	來源：視窗服務器
目標：AWS Managed Services	R 類型：重新主機	技術：遷移
AWS 服務：AWS CloudFormation；AWS Managed Services；AWS Systems Manager；Amazon S3		

Summary

在 Amazon Web Services (AWS) 雲端上，AWS Managed Services (AMS) 使用 AMS 工作負載擷取 (WIGS) 將現有工作負載移至 AMS 受管 VPC。此模式描述了一種解決方案，可將常見的工作負載前擷取活動自動化，例如升級 .NET 和 Windows，以 PowerShell 及執行由 AMS 維護的 Windows WIGS 預先擷取驗證。該模式還為運行結果提供了統一的用戶界面。它會將執行預先擷取活動的 AWS Systems Manager 命令文件封裝到 AWS CloudFormation 範本中。範本可以重複部署，而不需要存取 Systems Manager 本身或與 AMS 的自動化產生衝突。

事務, 背景

移轉至 AMS 需要使用包含 AMS 元件的 AMS 受管亞馬遜機器映像 (AMI) 佈建新的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。在現有資料中心執行的任何工作負載或應用程式都必須重新部署到從這些 AMS AMI 啟動的全新 EC2 執行個體。為了避免在此過程中潛在的大量手動工作，AMS 團隊建立了 AMS 工作負載擷取 (WIGS) 工作流程，將您的自訂映像檔上載至 AMS。

Windows 執行個體必須滿足幾個先決條件，才能進行 WIGS 程序。Windows PowerShell 指令碼通常用於執行必要的準備工作 (WIGS 準備)，並檢查執行個體是否已準備好進行 WIG (WIGS 預先擷取驗證)。準備和驗證程序需要工程師在每部伺服器上花費 15-30 分鐘的時間，逐一手動登入並執行指令碼。

業務司機

傳統上，使用 Systems Manager，您可以自動執行 Windows PowerShell 指令碼等作業工作。但是，由於 AMS 與用戶的自動化之間的風險較高和頻繁的衝突，AMS 通常不會授予其用戶對 Systems Manager 的訪問權限。

對於使用 AWS 應用程式遷移服務 (AWS MGN) 的大規模移轉，`C:\Program Files (x86)\AWS Replication Agent\post_launch folder`通常會在啟動測試或切換執行個體時自動執行。PowerShell 不過，如果在執行個體啟動期間立即執行這些指令碼，通常會與 AMS 的自動化衝突。因此，啟動可能會失敗，但未提供疑難排解失敗所需的執行結果。

這種模式解決了這些問題，並提供了工作的自動化解決方案。

先決條件和限制

先決條件

- 已完成具有 AMS 上線的有效 AWS 帳戶。
- AWS 帳戶中的 Amazon 簡單存儲服務（亞馬遜 S3）存儲桶。如果您在帳戶中沒有可控制的 S3 儲存貯體，請使用變更請求 (RFC) 來建立一個儲存貯體。
- 從儲存庫下載的先行檔案。ams-auto-prewigs-windows
- 套用此模式的伺服器必須符合下列需求：
  - 執行視窗伺服器 2012 或更新版本。
  - 在沙箱 VPC 遷移子網路中啟動或準備啟動。
  - 安裝 AWS Systems Manager 代理程式 (SSM 代理程式)。
  - 附加 AWS Identity and Access Management (IAM) 執行個體設定檔。執行個體設定檔必須具有從同一 AWS 帳戶中的 S3 儲存貯體下載檔案的許可。在先前的移轉設定期間，通常已建立符合上述需求的執行個體設定檔。
  - 可從 AWS Systems Manager 車隊管理員檢視。

限制

- 假髮前的活動會根據您的環境和業務需求而有所不同。您可能需要對此模式進行輕微修改，以滿足您的特定需求。

產品版本

- 此病毒碼已透過視窗伺服器 2012、2012 年 R2、2016 年和 2019 年進行測試。它理論上適用於更高版本的 Windows。它不適用於早期的 Windows 版本。

架構

架構圖顯示以下內容：

1. 具有遷移子網路的沙箱 VPC，其中包含尚未準備好的伺服器。
2. 存放 CloudFormation 範本所使用指令碼的 S3 儲存貯體。
3. CloudFormation 範本會部署「Systems Manager 命令」文件。程序會重複執行，直到步驟完成為止。
4. 例證已準備好，並製作 WIGS 的 RFC。
5. 在 AMS 受管理的 VPC 中，AMS 受管理子網路會在工作負載擷取後包含伺服器。

運作方式

- 此模式封裝在 AWS CloudFormation 範本中，允許基礎設施即程式碼 (IaC) 可重複部署。您只需為每個需要此自動化的 AWS 帳戶部署一次此範本。
- 自動化應用於部署此模式的 AWS 帳戶中具有標籤金鑰 AutoPreWIG 的所有 EC2 執行個體。第一次啟動具有標籤金鑰 AutoPreWIG 的 Amazon EC2 Windows 執行個體時，自動化會執行下列任務。
  1. PowerShell 將視窗升級至 5.1 版和 .NET 至 4.5.2 版本。執行個體可能會重新啟動多次，視其現有的 Windows PowerShell 和 .NET 版本而定。每次重新啟動後，升級都會繼續進行，直到完成為止。此步驟會在從 Windows 指令碼修改的 CloudFormation 範本中使用內嵌程式 PowerShell 碼，以及伺服器重新開機的特定 Systems Manager 指引。
  2. 從 Amazon S3 下載並執行您已自訂的視窗 PowerShell 指令碼，以準備適用於 WIG 的 Amazon EC2 Windows 執行個體。如需詳細資訊，請參閱《史詩》一節。
  3. 從 AWS 安裝視窗 WIGS 預先擷取驗證 PowerShell 模組。
  4. 執行 Windows WIGS 預先擷取驗證，並在系統管理員狀態管理員中檢視結果。

工具

- AWS CloudFormation — AWS CloudFormation 是一項可協助您建立 AWS 資源模型和設定 AWS 資源的服務。您可以使用描述所需的所有 AWS 資源及其相依性，以便您可以啟動和設定這些資源為堆疊。，此模式使用 CloudFormation 範本自動化此模式中的資源部署。
- AWS Managed Services — AWS Managed Services (AMS) 是一種企業服務，可為您的 AWS 基礎設施提供持續管理。在 AMS 環境中對基礎結構所做的變更必須透過 RFC 進行。

- AWS Systems Manager — AWS Systems Manager (先前稱為 SSM) 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。您可以使用 Systems Manager 主控台檢視來自多個 AWS 服務的操作資料，並自動化 AWS 資源的操作任務。此模式使用「Systems Manager」來執行和檢視假髮前活動的執行結果。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。此模式使用 Amazon S3 存儲 CloudFormation 模板和下載的 Windows PowerShell 腳本。

史诗

建立自訂 Windows PowerShell 指令碼以自動執行其他工作

任務	描述	所需技能
根據業務需求對伺服器執行必要的變更。	如果您需要在擷取伺服器之前自動套用變更，請建立名為 `ingestion-prep.ps1` 的 Windows PowerShell 指令碼。 重要事項：指令碼不得包含重新啟動伺服器的指示，也不得要求管理員權限。	PowerShell 腳本
移除 AMS 不支援的軟體。	AMS 需要特定的軟體，例如防毒應用程式和 VMware 工具，才能執行 WIGS。在 `ingestion-prep.ps1` 指令碼中包含解除安裝。如需有關不受支援軟體的詳細資訊，請參閱 AWS 文件。	PowerShell 腳本

將 CloudFormation 範本和可選的視窗 PowerShell 指令碼上傳到 Amazon S3

任務	描述	所需技能
在 S3 中創建一個文件夾。	在部署此模式的相同 AWS 帳戶中的 S3 儲存貯體中，建立資料夾。	一般 AWS
上傳腳本。	將`PreWIGs_CFN.json` CloudFormation 範本和 `ingestion-prep.ps1` Windows 指PowerShell 令碼 (您在上一個史詩中建立) 上傳到 Amazon S3 資料夾。	一般 AWS

部署 CloudFormation 堆疊

任務	描述	所需技能
選取變更類型。	瀏覽至 AMS 主控台以建立 RFC。使用「從 CloudFormation (CFN) 範本建立堆疊」變更類型。	一般 AMS
設定 CloudFormation 範本路徑的執行參數。	在 [執行組態] 區段中，展開 [其他組態]。在CloudFormation 範本 S3 端點方塊中，將 URL 貼至 CloudFormation 範本。	一般 AMS
指定 Amazon S3 資料夾的路徑。	在「參數」下，用ScriptSource作「名稱」。在「值」中，輸入包含 Windows PowerShell 指令碼之 S3 資料夾的路徑。確保您使用 `https://xxx` URL 而不是 `s3://xxx` URI，並/在末尾包含。	一般 AMS

任務	描述	所需技能
部署堆疊。	若要部署堆疊，請選擇 [建立]。	一般 AMS
將 RFC 升級為 AMS 行動。	RFC 必須由 AMS Ops 小組手動實作，因為它使用 Systems Manager 來部署資源，且需要安全性檢閱。一旦您創建 RFC，它將被系統自動拒絕。選擇 RFC，並向 RFC 添加一個對應關係，說明請手動執行。請記下 RFC ID，並使用服務要求進行升級。	一般 AMS

將自動化應用於實例

任務	描述	所需技能
將 AutoPre WIGs 標籤新增至實例。	請注意您要套用此自動化操作的所有執行個體的 ID，並等待至少 30 分鐘，讓執行個體完成 AMS 實作的自動化作業。提交自動 RFC 以將 AutoPreWIGs 作為金鑰新增標籤，並將任何字串 (例如 1) 新增為值。 自動化操作將在您新增標籤後幾分鐘內套用。	一般 AMS
驗證自動化結果。	開啟 Systems Manager 主控台，然後選擇 [狀態管理員]。選擇名稱為 AMS 預先佈置和驗證關聯的關聯 ID。在 [執行歷史記錄] 索引標籤上，您可以看到自動化的結果。	一般 AMS

任務	描述	所需技能
修正所有錯誤。	如果自動化失敗，請選擇其執行 ID。您可以查看每個 EC2 執行個體的執行結果。若要查看自動化操作中每個步驟的詳細資訊，請選擇「輸出」。如果特定步驟失敗，請使用「輸出」和「錯誤」段落中的資訊來診斷問題。	移民工程師
移除 AutoPre假髮標籤。	重要事項：修正錯誤後 (如果有的話)，請提交自動 RFC 以移除 AutoPreWIGs 標籤。如果您不移除標籤，WIGS 將會失敗。	一般 AMS

擷取準備好的執行個體

任務	描述	所需技能
提交假髮的 RFC。	現在執行個體已準備好進行工作負載擷取，請提交 WIG 的 RFC。	一般 AMS

相關資源

- AMS 工作負載擷取 (假髮)
- 移轉工作負載:Windows 預先擷取驗證
- AWS 應用程式遷移服務快速入門指南
- 開始使用 AWS CloudFormation
- 設定 AWS Systems Manager

在將主機移轉至 AWS 期間建立防火牆請求的核准程序

創建者：斯里坎斯朗格瓦哈拉 (AWS)

R 類型：重新主機	環境：生產	技術：遷移
資料來源：內部部署	目標：AWS 雲端	

Summary

如果您想要使用 AWS 應用程式遷移服務或 AWS 上的雲端遷移工廠重新託管到 Amazon Web Services (AWS) 雲端，其中一個先決條件是您必須保持 TCP 連接埠 443 和 1500 的開啟狀態。一般而言，開啟這些防火牆連接埠需要您的資訊安全 (InfoSec) 團隊核准。

此模式概述了在重新託管遷移到 AWS 雲端期間從 InfoSec 團隊取得防火牆請求核准的程序。您可以使用此程序來避免 InfoSec 小組拒絕您的防火牆要求，這可能會變得昂貴且耗時。防火牆申請程序在 AWS 遷移顧問和領導人之間有兩個審核和核准步驟，他們與您 InfoSec 和應用程式團隊合作開啟防火牆連接埠。

此模式假設您正在規劃與 AWS 顧問或組織的遷移專家進行重新託管遷移。如果您的組織沒有防火牆核准程序或防火牆要求全面核准表單，則可以使用此模式。如需有關此功能的詳細資訊，請參閱此模式的 < 限制 > 一節。如需應用程式移轉服務之網路需求的詳細資訊，請參閱應用程式移轉服務說明文件中的網路需求。

先決條件和限制

先決條件

- 與您組織的 AWS 顧問或遷移專家進行計劃的重新託管遷移
- 遷移堆疊所需的連接埠和 IP 資訊
- 現有和 future 的狀態體系結構圖
- 內部部署和目的地基礎結構、連接埠和 zone-to-zone 流量的防火牆資訊
- 防火牆要求檢閱檢查清單 (隨附)
- 根據您組織的需求設定的防火牆要求文件
- 防火牆審核者和核准者的連絡人清單，包括下列角色：

- 防火牆請求提交者 — AWS 遷移專家或顧問。防火牆要求提交者也可以是組織的移轉專家。
- 防火牆請求審核者 — 一般而言，這是 AWS 的單一聯絡窗口 (SPOC)。
- 防火牆請求核准者 — 專 InfoSec 案團隊成員。

限制

- 此病毒碼描述一般防火牆要求核准程序。個別組織的需求可能會有所不同。
- 請確定您已追蹤防火牆要求文件的變更。

下表顯示此模式的使用案例。

您的組織是否有現有的防火牆核准程序？	您的組織是否有現有的防火牆要求表單？	建議的動作
是	是	與 AWS 顧問或遷移專家協同合作，以實作您組織的程序。
否	是	使用此病毒碼的防火牆核准程序。請使用 AWS 顧問或組織的移轉專家來提交防火牆要求全面核准表單。
否	否	使用此病毒碼的防火牆核准程序。請使用 AWS 顧問或組織的移轉專家來提交防火牆要求全面核准表單。

架構

下圖顯示防火牆要求核准程序的步驟。

工具

您可以使用掃描器工具，如帕洛阿爾托網絡或SolarWinds分析和驗證防火牆和 IP 地址。

史诗

分析防火牆要求

任務	描述	所需技能
分析連接埠和 IP 位址。	防火牆要求提交者會完成初始分析，以瞭解所需的防火牆連接埠和 IP 位址。完成此操作後，他們會要求您的 InfoSec 團隊打開所需的端口並映射 IP 地址。	AWS 雲端工程師、移轉專家

驗證防火牆要求

任務	描述	所需技能
驗證防火牆資訊。	AWS 雲端工程師會安排與您的 InfoSec 團隊進行會議。在此會議期間，工程師會檢查並驗證防火牆要求資訊。 一般而言，防火牆要求提交者與防火牆要求者是相同的人員。如果觀察到或建議任何內容，則此驗證階段可以根據核准者給出的反饋進行迭代。	AWS 雲端工程師、移轉專家
更新防火牆要求文件。	在 InfoSec 團隊分享他們的意見反應之後，就會編輯、儲存並重新上傳防火牆要求文件。此文件會在每次版序之後更新。 建議您將此文件儲存在版本控制的儲存資料夾中。這表示所有變更都會追蹤並正確套用。	AWS 雲端工程師、移轉專家

提交防火牆要求

任務	描述	所需技能
提交防火牆要求。	防火牆請求核准者核准防火牆全面核准請求後，AWS 雲端工程師會提交防火牆要求。請求指定必須開啟的連接埠，以及映射和更新 AWS 帳戶所需的 IP 地址。 您可以在提交防火牆要求後提出建議或提供意見反應。我們建議您將此意見反應程序自動化，並透過定義的工作流程機制傳送任何編輯。	AWS 雲端工程師、移轉專家

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶

由阿尼爾·庫納帕雷迪（AWS）和文卡特拉瑪納奇塔（AWS）創建

環境：生產	資料來源：AWS 雲端中的 VPC	目標：由 AWS Managed Services 管理的 VPC
R 類型：重新主機	工作負載：Microsoft	技術：移轉；作業；安全性、身分識別、合規性；雲端原生
AWS 服務：AWS Managed Services		

Summary

此模式說明將 Amazon Elastic Compute Cloud (Amazon EC2) Windows 執行個體遷移和導入 Amazon Web Services (AWS) Managed Services (AMS) 帳戶的 step-by-step 程序。AMS 可協助您更有效率且安全地管理執行個體。AMS 提供營運彈性、增強安全性與合規性，並協助您最佳化容量並降低成本。

此模式從 EC2 Windows 執行個體開始，該執行個體已移轉至 AMS 帳戶中的暫存子網路。您可以使用各種遷移服務和工具來執行此任務，例如 AWS 應用程式遷移服務。

若要對 AMS 管理的環境進行變更，請針對特定作業或動作建立並提交變更請求 (RFC)。使用 AMS 工作負載擷取 (WIGS) RFC，您可以將執行個體內嵌到 AMS 帳戶中，並建立自訂的 Amazon 機器映像 (AMI)。然後，您可以透過提交另一個 RFC 以建立 EC2 堆疊來建立 AMS 管理的 EC2 執行個體。如需詳細資訊，請參閱 AMS 文件中的 AMS 工作負載擷取。

先決條件和限制

先決條件

- 由 AMS 管理的作用中 AWS 帳戶
- 現有的 landing zone
- 在 AMS 管理的 VPC 中進行變更的權限
- AMS 帳戶中暫存子網路中的 Amazon EC2 Windows 執行個體
- 完成使用 AMS WIGS 移轉工作負載的一般先決條件

- 完成使用 AMS 假髮移轉工作負載的 Windows 先決條件

限制

- 此模式適用於操作視窗伺服器的 EC2 執行個體。此模式不適用於執行其他作業系統 (例如 Linux) 的執行個體。

架構

源, 技術, 堆棧

AMS 帳戶中暫存子網路中的 Amazon EC2 視窗執行個體

目標技術堆疊

由 AWS Managed Services (AMS) 管理的 Amazon EC2 視窗執行個體

目標架構

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Managed Services (AMS) 提供 AWS 基礎設施的持續管理，包括 AWS 工作負載的監控、事件管理、安全指導、修補程式支援和備份，協助您更有效率且安全地營運。

其他服務

- PowerShell是一個 Microsoft 的自動化和配置管理程序，可以在 Windows，Linux 和 macOS 上運行。

史诗

設定執行個體的設定

任務	描述	所需技能
變更 DNS 用戶端設定。	1. 在來源 EC2 執行個體上，以系統管理員身分開啟命令提示字元`gpedit.msc` ，輸入，然後按 Enter。 2. 在本機群組原則編輯器中，瀏覽至 [電腦設定]、[系統管理範本]、[網路]、[DNS 用戶端 3. 針對 [主要 DNS 尾碼]，選擇 [未設定]。 4. 對於 [主要 DNS 尾碼遞增]，請選擇 [未設定]。	移民工程師
更改視窗更新設置。	1. 在本機群組原則編輯器中，瀏覽至 [電腦設定]、[系統管理範本]、[Windows 元件]、[Windows 更新]。 2. 針對 [指定內部網路 Microsoft 更新服務位置]，選擇 [未設定] 3. 對於 [設定自動更新]，選擇 [未設定]。 4. 針對 [自動更新] 偵測頻率，選擇 [未設定]。 5. 關閉本機群組原則編輯器。	移民工程師
啟動防火牆。	1. 在來源 EC2 執行個體上，以系統管理員身分開啟命令提示字元`services.msc` ，輸入，然後按 Enter。	移民工程師

任務	描述	所需技能
	2. 在視窗服務中，啟動防火牆。 3. 關閉視窗服務。	

準備 AMS WIGS 的執行個體

任務	描述	所需技能
清理並準備執行個體。	1. 使用防禦主機和本機登入資料，建立與暫存子網路中 EC2 執行個體的遠端桌面通訊協定 (RDP) 連線。 2. 移除 AMS 中不需要的所有舊版軟體、防毒軟體和備份解決方案。	移民工程師
修復 sppnp.dll 檔案。	1. 前往 `C:\Windows\System32\sppnp.dll` 。 2. `sppnp.dll` 將重新命名為 `sppnp_old.dll` 。 3. 使用 PowerShell 和管理員認證，輸入下列命令： `dism /online /cleanup-image /restorehealth` `sfc /scannnow` 4. 重新啟動 EC2 視窗執行個體。	移民工程師
執行假設前驗證指令碼。	1. 從 AMS 說明文件中的移轉工作負載：Windows 預先擷取驗證，下載 Windows WIGS 預先擷取驗證壓縮檔	移民工程師

任務	描述	所需技能
	案 (`windows-prewings-valication.zip`)。 2. 執行 Windows 假設前驗證指令碼，並驗證結果。 3. 如果驗證失敗，請修正問題，然後重新執行驗證指令碼，直到驗證成功為止。	
建立故障安全防護 AMI。	在假髮前驗證通過後，建立預先擷取 AMI，如下所示： 1. 選擇部署，高級堆棧組件，AMI，創建。 2. 在建立期間，新增標籤`Key=Name, Value=APPLICATION-ID_Ingest Ready` 。 3. 在繼續之前，請等待 AMI 創建。 如需詳細資訊，請參閱 AMI | 在 AMS 文件中建立。	移民工程師

擷取和驗證執行個體

任務	描述	所需技能
提交 RFC 以建立工作負載擷取堆疊。	提交變更請求 (RFC) 以啟動 AMS 假髮。如需指示，請參閱 AMS 說明文件中的工作負載擷取堆疊：建立。這會啟動工作負載擷取並安裝 AMS 所需的所有軟體，包括備份工	移民工程師

任務	描述	所需技能
	具、Amazon EC2 管理軟體和防毒軟體。	
驗證成功移轉。	工作負載擷取完成後，您可以看到 AMS 代管的執行個體和 AMS 導入的 AMI。 1. 使用網域認證登入 AMS 代管的執行個體。 2. 驗證網域加入的方式如下： a. 在 [Windows 檔案總管] 中，以滑鼠右鍵按一下 [本機]，然後選擇 b. 在「裝置規格」區段中，確認網域出現在「完整裝置名稱」中。 3. 驗證來源和目標磁碟機。	移民工程師

在目標 AMS 帳戶中啟動執行個體

任務	描述	所需技能
提交 RFC 以建立 EC2 堆疊。	1. 使用 Windows 執行個體的 AMS 擷取 AMI，根據 AMS 文件中建立 EC2 堆疊執行個體中的指示，為 EC2 堆疊準備 RFC。在 EC2 堆疊 RFC 中，提供所有參數，包括伺服器名稱、標籤、目標 VPC、目標子網路、執行個體類型、目標安全群組、擷取 AMI 和角色。 2. 提交 EC2 堆疊的 RFC，然後等待執行個體成功建立。	移民工程師

相關資源

AWS 方案指引

- 在 Windows 上自動執行 AWS Managed Services 的工作負載前擷取活動
- 使用 Python 在 AMS 中自動創建一個 RFC

AMS 文件

- AMS 工作負載擷取
- 遷移如何改變您的資源
- 移轉工作負載：標準程序

行銷資源

- AWS Managed Services
- AWS Managed Services 常見問題
- AWS Managed Services 資源
- AWS Managed Services 功能

使用日誌傳送將 LUW 的 Db2 移轉至 Amazon EC2，以減少中斷時間

由蔡豐 (AWS)、安巴里什薩塔卡 (AWS) 和索拉福·夏爾馬 (AWS) 創建

環境：生產	來源：適用於 Linux 的內部部署 Db2	目標：Amazon EC2 上的 Db2
R 類型：重新主機	工作負載：IBM	技術：移轉；資料庫
AWS 服務：AWS Direct Connect；Amazon EBS；Amazon EC2；Amazon S3；AWS Site-to-Site VPN		

Summary

當客戶將其 IBM Db2 用於 LUW (Linux、UNIX 和 Windows) 工作負載遷移到 Amazon Web Services (AWS) 時，使用亞馬遜彈性運算雲端 (Amazon EC2) 搭配自攜授權 (BYOL) 模型是最快速的方法。不過，將大量資料從現場部署 Db2 遷移到 AWS 可能是一項挑戰，尤其是當中斷時間很短時。許多客戶嘗試將中斷時間設定為少於 30 分鐘，因此資料庫本身的時間很少。

此模式涵蓋如何使用交易記錄傳送，以短暫中斷時間完成 Db2 移轉。這種方法適用於小端 Linux 平台上的 Db2。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 EC2 執行個體上執行的 Db2 執行個體，符合現場部署檔案系統配置
- EC2 執行個體可存取的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體
- AWS Identity and Access Management (IAM) 政策和角色，可以透過程式設計方式向 Amazon S3 進行呼叫
- Amazon EC2 和現場部署伺服器上的同步時區和系統時鐘
- 透過 AWS Site-to-Site VPN 或 AWS 直接連接至 AWS 的現場部署網路

限制

- Db2 現場部署執行個體和 Amazon EC2 必須位於相同的平台系列上。
- 必須記錄 Db2 內部部署工作負載。若要封鎖任何未記錄的交易，請`blocknonlogged=yes`在資料庫組態中設定。

產品版本

- 適用於 LUW 11.5.9 版及更新版本的 Db2

架構

源, 技術, 堆棧

- DB2 在 Linux 上

目標技術堆疊

- Amazon EBS
- Amazon EC2
- AWS Identity and Access Management (IAM)
- Amazon S3
- AWS Site-to-Site VPN 或直接 Connect

目標架構

下圖顯示一個 Db2 執行個體以虛擬私人網路 (VPN) 連接到 Amazon EC2 上的 Db2 連線在現場部署執行。虛線代表資料中心和 AWS 雲端之間的 VPN 通道。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Site-to-Site VPN 可協助您在 AWS 上啟動的執行個體和自己的遠端網路之間傳遞流量。

其他工具

- db2cli 是 Db2 互動式 CLI 指令。

最佳實務

- 在目標資料庫上，使用 Amazon S3 的閘道端點存取 Amazon S3 中的資料庫備份映像和日誌檔。
- 在來源資料庫上，使用適用 PrivateLink 於 Amazon S3 的 AWS 將資料庫備份映像和日誌檔案傳送到 Amazon S3。

史诗

設定環境變數

任務	描述	所需技能
設定環境變數。	此模式使用以下名稱： • 執行個體名稱：`db2inst1` • 資料庫名稱：`SAMPLE` 您可以變更它們以符合您的環境。	DBA

設定內部部署 Db2 伺服器

任務	描述	所需技能
設定 AWS CLI。	若要下載並安裝最新版本的 AWS CLI，請執行下列命令： `$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"` `unzip awscliv2.zip` `sudo ./aws/install`	管理員
設定 Db2 封存記錄的本機目的地。	為了使 Amazon EC2 上的目標資料庫與現場部署來源資料庫保持同步，需要從來源擷取最新的交易日誌。 在此設置中，`/db2logs`由源`LOGARCHMETH2` 上設置為臨時區域。此目錄中的存檔日誌將同步到 Amazon S3，並可由亞馬 Amazon EC2 上的 Db2 存取。該模式使用，`LOGARCHMETH2` 因為`LOGARCHMETH1` 可能已配置為使用 AWS CLI 命令無法訪問的第三方供應商工具。若要擷取記錄檔，請執行下列命令： `db2 connect to sample` `db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs`	DBA

任務	描述	所需技能
執行線上資料庫備份。	執行線上資料庫備份，並將其儲存至本機備份檔案系統： `db2 backup db sample online to /backup`	DBA

設定 S3 儲存貯體和 IAM 政策

任務	描述	所需技能
建立 S3 儲存貯體。	為現場部署伺服器建立 S3 儲存貯體，以便將備份 Db2 映像和日誌檔傳送到 AWS 上。該桶也將由 Amazon EC2 訪問： `aws s3api create-bucket --bucket logshipmig-db2 --region us-east-1`	AWS 系統管理員
建立 IAM 政策。	該`db2bucket.json` 檔案包含存取 Amazon S3 儲存貯體的 IAM 政策： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "kms:GenerateDataKey", "kms:Decrypt",`	AWS 管理員、AWS 系統管理員

任務	描述	所需技能
	<pre>"s3:PutObject", "s3:GetObject", "s3:AbortMultipart Upload", "s3:ListBucket", "s3:DeleteObject", "s3:GetObjectVersi on", "s3:ListMultipartU ploadParts"], "Resource": ["arn:aws:s3:::logs hipmig-db2/*", "arn:aws:s3:::logs hipmig-db2"] }] }</pre> 若要建立政策，請使用下列 AWS CLI 命令： <pre>aws iam create-policy \ --policy-name db2s3policy \ --policy-document file://db2bucket.j son</pre>	

任務	描述	所需技能
	JSON 輸出會顯示政策的 Amazon 資源名稱 (ARN)，其中`aws_account_id` 代表您的帳戶 ID： `"Arn": "arn:aws:iam::aws_account_id:policy/db2s3policy"`	
將 IAM 政策附加到 EC2 執行個體使用的 IAM 角色。	在大多數 AWS 環境中，執行中的 EC2 執行個體具有由系統管理員設定的 IAM 角色。如果未設定 IAM 角色，請建立角色並選擇 EC2 主控台上的修改 IAM 角色，將角色與託管 Db2 資料庫的 EC2 執行個體建立關聯。使用政策 ARN 將 IAM 政策附加到 IAM 角色： `aws iam attach-role-policy \` `--policy-arn "arn:aws:iam::aws_account_id:policy/db2s3policy" \` `--role-name db2s3role` 附加政策後，任何與 IAM 角色關聯的 EC2 執行個體都可以存取 S3 儲存貯體。	AWS 管理員、AWS 系統管理員

將來源資料庫備份映像和日誌檔案傳送到 Amazon S3

任務	描述	所需技能
在現場部署 Db2 伺服器上設定 AWS CLI。	使用先前步驟中Secret Access Key產生的Access Key ID和產生的 AWS CLI： `$ aws configure` `AWS Access Key ID` `[None]: **********` `AWS Secret Access Key` `[None]: **********` `Default region name` `[None]: us-east-1` `Default output format` `[None]: json`	AWS 管理員、AWS 系統管理員
將備份映像傳送到 Amazon S3。	之前，線上資料庫備份已儲存至/backup本機目錄。若要將該備份映像傳送至 S3 儲存貯體，請執行下列命令： `aws s3 sync /backup` `s3://logshipmig-db2/` `SAMPLE_backup`	AWS 管理員、移轉工程師
將 Db2 存檔日誌傳送到 Amazon S3。	將現場部署 Db2 存檔日誌與 Amazon EC2 上的目標 Db2 執行個體可存取的 S3 儲存貯體同步： `aws s3 sync /db2logs` `s3://logshipmig-db2/` `SAMPLE_LOG`	AWS 管理員、移轉工程師

任務	描述	所需技能
	使用 cron 或其他排程工具定期執行此命令。頻率取決於來源資料庫封存交易記錄檔的頻率。	

將 Amazon EC2 上的 Db2 Connect 到 Amazon S3 並開始數據庫同步

任務	描述	所需技能
創建一個 PKCS12 密鑰庫。	Db2 使用公開金鑰加密標準 (PKCS) 加密金鑰存放區來確保 AWS 存取金鑰的安全。建立金鑰儲存庫並設定來源 Db2 執行個體以使用它： `gsk8capicmd_64 -keydb -create -db "/home/db2inst1/.keystore/db2s3.p12" -pw "<password>" -type pkcs12 -stash` `db2 "update dbm cfg using keystore_location /home/db2inst1/.keystore/db2s3.p12 keystore_type pkcs12"`	DBA
建立 Db2 儲存存取權別名。	若要建立儲存區存取別名，請使用下列指令碼語法： `db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint>`	DBA

任務	描述	所需技能
	container '<bucket_name>'" 例如，您的指令碼可能如下所示： db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazonaws.com container 'logshipmig-db2'"	

任務	描述	所需技能
設定暫存區域。	根據預設，Db2 會使用DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH 做為上傳和下載檔案到 Amazon S3 和從 Amazon S3 上傳和下載檔案的暫存區域。預設路徑位於執行個體主目錄sqllib/tmp/RemoteStorage.xxxx 下，xxxx參照 Db2 分割區編號。請注意，暫存區域必須有足夠的容量來容納備份映像檔和記錄檔。您可以使用登錄將暫存區指向不同的目錄。 我們也建議使用DB2_ENABLE_COS_SDK=ON DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore 、和連結至程式awssdk庫來略過 Amazon S3 暫存區進行資料庫備份和還原： `#By root:` `cp -rp /home/db2inst1/sqllib/lib64/awssdk/RHEL/7.6/* /home/db2inst1/sqllib/lib64/` `#By db2 instance owner:` `db2set DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH=/db2stage` `db2set DB2_ENABLE_COS_SDK=ON`	DBA

任務	描述	所需技能
	`Db2set DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore` `db2stop` `db2start`	
從備份映像還原資料庫。	從 S3 儲存貯體中的備份映像還原 Amazon EC2 上的目標資料庫： `db2 restore db sample from DB2REMOTE://DB2AWSS3/logshipmig-db2/SAMPLE_backup replace existing`	DBA

任務	描述	所需技能
向前捲動資料庫。	還原完成之後，目標資料庫會進入向前復原擱置狀態。設定LOGARCHMETH1 並LOGARCHMETH2 讓 Db2 知道從何處取得交易記錄檔： `db2 update db cfg for SAMPLE using LOGARCHMETH1 'DB2REMOTE://DB2AWSS3//SAMPLE_LOGS/'` `db2 update db cfg for SAMPLE using LOGARCHMETH2 OFF` 啟動資料庫向前復原： `db2 ROLLFORWARD DATABASE sample to END OF LOGS` 此命令會處理已傳輸到 S3 儲存貯體的所有記錄檔。根據內部部署 Db2 伺服器上s3 sync命令的頻率定期執行它。例如，如果每小時s3 sync執行一次，且同步處理所有記錄檔需要 10 分鐘，請將命令設定為在每小時後 10 分鐘執行。	DBA

在切換視窗期間將 Db2 上的 Amazon EC2 上線上

任務	描述	所需技能
使目標資料庫上線。	在切換視窗期間，請執行下列其中一項作業： • 將內部部署資料庫放入ADMIN MODE，然後執行s3 sync命令以強制封存最後一個交易記錄檔。 • 關閉資料庫。 將最後一個交易日誌同步到 Amazon S3 之後，請執行最後一次ROLLFORWARD 命令： `db2 rollforward DB sample to END OF LOGS` `db2 rollforward DB sample complete` `Rollforward Status` `....` `Rollforward status = not pending` `....` `DB20000I The ROLLFORWARD command completed successfully.` `db2 activate db sample` `DB20000I The ACTIVATE DATABASE command completed successfully.`	DBA

任務	描述	所需技能
	使目標資料庫上線，並將應用程式連線指向 Amazon EC2 上的 Db2。	

故障診斷

問題	解決方案
如果多個資料庫在不同的主機 (DEV、QA、PROD) 上具有相同的執行個體名稱和資料庫名稱，則備份和記錄檔可能會移至相同的子目錄。	對 DEV、QA 和 PROD 使用不同的 S3 儲存貯體，並將主機名稱新增為子目錄前綴以避免混淆。
如果同一位置有多個備份映像，則在還原時會出現以下錯誤： `SQL2522N More than one backup file matches the time stamp value provided for the backed up database image.`	在命`restore`令中，添加備份的時間戳： `db2 restore db sample from DB2REMOTE://DB2AWSS3/logshipmig-db2/SAMPLE_backup taken at 20230628164042 replace existing`

相關資源

- Db2 備份和還原不同作業系統和硬體平台之間的作業
- 設定資料庫存取權限別名和 DB2REMOTE
- Db2 向前滾動命令
- Db2 次要記錄封存方法

透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2

創建者：馮蔡 (AWS)、阿魯娜甘格雷迪 (AWS) 和文卡特山戈文丹 (AWS)

環境：生產	資料來源：IBM Db2 適用於內部部署的 LUW	目標：Amazon EC2 上的 Db2
R 類型：重新主機	工作負載：IBM	技術：移轉、資料庫、作業系統
AWS 服務：AWS Direct Connect；Amazon EC2；Amazon S3；AWS Site-to-Site VPN		

Summary

當客戶將他們的 IBM Db2 LUW (Linux、UNIX 和 Windows) 工作負載遷移到 Amazon Web Services (AWS) 時，使用亞馬遜彈性運算雲端 (Amazon EC2) 搭配使用自有授權 (BYOL) 模型是最好的方法。不過，將大量資料從現場部署 Db2 遷移到 AWS 可能是一項挑戰，尤其是當中斷時間很短時。許多客戶嘗試將中斷時間設定為少於 30 分鐘，因此資料庫本身的時間很少。

此模式涵蓋如何使用 Db2 高可用性災難復原 (HADR)，以短暫中斷時間完成 Db2 移轉。此方法適用於小端 Linux 平台上且未使用資料分割功能 (DPF) 的 Db2 資料庫。

先決條件和限制

前提

- 有效的 AWS 帳戶
- 在 Amazon EC2 執行個體上執行的 Db2 執行個體，符合現場部署檔案系統配置
- EC2 執行個體可存取的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體
- AWS Identity and Access Management (IAM) 政策和角色，可以透過程式設計方式向 Amazon S3 進行呼叫
- Amazon EC2 和現場部署伺服器上的同步時區和系統時鐘
- 透過 AWS Site-to-Site VPN 或 AWS 直接連接至 AWS 的現場部署網路
- 現場部署伺服器與 HADR 連接埠上的 Amazon EC2 之間的通訊

限制

- Db2 現場部署執行個體和 Amazon EC2 必須位於相同的平台系列上。
- 在資料分割資料庫環境中不支援 HADR。
- HADR 不支援將原始 I/O (直接磁碟存取) 用於資料庫記錄檔。
- HADR 不支援無限日誌記錄。
- LOGINDEXBUILD必須設定為YES，這會增加重建索引的記錄使用量。
- 必須記錄 Db2 內部部署工作負載。blocknonlogged=yes在數據庫配置中設置以阻止任何未記錄的事務。

產品版本

- 適用於 LUW 11.5.9 版及更新版本的 Db2

架構

源, 技術, 堆棧

- DB2 在 Linux 上

目標技術堆疊

- Amazon EC2
- AWS Identity and Access Management (IAM)
- Amazon S3
- AWS Site-to-Site VPN

目標架構

在下圖中，Db2 內部部署正在db2-server1作為主運行。它有兩個 HADR 待命目標。一個待命目標位於內部部署，並且是可選的。另一個待命目db2-ec2標是在 Amazon EC2 上。資料庫切斷到 AWS 之後，db2-ec2成為主要資料庫。

1. 記錄會從主要內部部署資料庫串流至待命內部部署資料庫。

2. 使用 Db2 HADR，日誌會透過 Site-to-Site VPN 從主要現場部署資料庫串流到 Amazon EC2 上的 Db2。
3. Db2 備份和存檔日誌會從主要現場部署資料庫傳送到 AWS 上的 S3 儲存貯體。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Site-to-Site VPN 可協助您在 AWS 上啟動的執行個體和自己的遠端網路之間傳遞流量。

其他工具

- db2cli 是 Db2 互動式 CLI 指令。

最佳實務

- 在目標資料庫上，使用 Amazon S3 的閘道端點存取 Amazon S3 中的資料庫備份映像和日誌檔。
- 在來源資料庫上，使用適用 PrivateLink 於 Amazon S3 的 AWS 將資料庫備份映像和日誌檔案傳送到 Amazon S3。

史诗

設定環境變數

任務	描述	所需技能
設定環境變數。	此病毒碼使用下列名稱和連接埠： 1. Db2 內部部署主機名稱：`db2-server1` 2. HADR 待命主機名稱：`db2-server2` （如果 HADR 目前正在內部部署執行） 3. Amazon EC2 主機名稱：`db2-ec2` 4. 執行個體名稱：`db2inst1` 5. 資料庫名稱：`SAMPLE` 6. 哈德爾連接埠： • `db2-server1: 50010` • `db2-server2: 50011` • `db2-ec2: 50012` 您可以變更它們以符合您的環境。	DBA

設定內部部署 Db2 伺服器

任務	描述	所需技能
設定 AWS CLI。	若要下載並安裝最新版本的 AWS CLI，請執行下列命令：	管理員

任務	描述	所需技能
	`$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"` `unzip awscliv2.zip` `sudo ./aws/install`	

任務	描述	所需技能
設定 Db2 封存記錄的本機目的地。	大量更新批次工作和網路速度變慢等情況可能會導致 HADR 待命伺服器出現延遲。為了 catch，待命伺服器需要來自主要伺服器的交易記錄檔。請求日誌的位置順序如下： • 主要伺服器上的使用中記錄目錄 • 待命伺服器上的LOGARCHMETH1 或LOGARCHMETH2 位置 • 主要伺服器上的LOGARCHMETH1 或LOGARCHMETH2 位置 在此設置中，/db2logs由源LOGARCHMETH2 上設置為臨時區域。此目錄中的存檔日誌將同步到 Amazon S3，並可由亞馬 Amazon EC2 上的 Db2 存取。該模式使用的LOGARCHMETH2 原因是LOGARCHMETH1 可能已配置為使用 AWS CLI 命令無法訪問的第三方供應商工具： `db2 connect to sample` `db2 update db cfg for SAMPLE using LOGARCHMETH2 disk:/db2logs`	DBA

任務	描述	所需技能
執行線上資料庫備份。	執行線上資料庫備份，並將其儲存至本機備份檔案系統： `db2 backup db sample online to /backup`	DBA

設定 S3 儲存貯體和 IAM 政策

任務	描述	所需技能
建立 S3 儲存貯體。	為現場部署伺服器建立 S3 儲存貯體，以便將備份 Db2 映像和日誌檔傳送到 AWS 上。該桶將由 Amazon EC2 訪問： `aws s3api create-bucket --bucket hadrmig-db2 --region us-east-1`	AWS 管理員
建立 IAM 政策。	該`db2bucket.json` 檔案包含用於存取 S3 儲存貯體的 IAM 政策： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "kms:GenerateDataKey", "kms:Decrypt",`	AWS 管理員、AWS 系統管理員

任務	描述	所需技能
	<pre>"s3:PutObject", "s3:GetObject", "s3:AbortMultipart Upload", "s3:ListBucket", "s3:DeleteObject", "s3:GetObjectVersi on", "s3:ListMultipartU ploadParts"], "Resource": ["arn:aws:s3:::hadr mig-db2/*", "arn:aws:s3:::hadr mig-db2"] }] }</pre> 若要建立政策，請使用下列 AWS CLI 命令： <pre>aws iam create-policy \ --policy-name db2s3hapolicy \ --policy-document file://db2bucket.j son</pre>	

任務	描述	所需技能
	JSON 輸出會顯示政策的 Amazon 資源名稱 (ARN)，其中`aws_account_id` 代表您的帳戶 ID： `"Arn": "arn:aws:iam::aws_account_id:policy/db2s3hapolicy"`	
將 IAM 政策連接至 IAM 角色。	通常，執行 Db2 的 EC2 執行個體會具有系統管理員指派的 IAM 角色。如果未指派任何 IAM 角色，您可以在 Amazon EC2 主控台上選擇修改 IAM 角色。 將 IAM 政策附加到與 EC2 執行個體相關聯的 IAM 角色。附加政策後，EC2 執行個體可以存取 S3 儲存貯體： `aws iam attach-role-policy --policy-arn "arn:aws:iam::aws_account_id:policy/db2s3hapolicy" --role-name db2s3harole`	

將來源資料庫備份映像和日誌檔案傳送到 Amazon S3

任務	描述	所需技能
在現場部署 Db2 伺服器上設定 AWS CLI。	使用您先前產生的`Access Key ID`和`Secret Access Key`來設定 AWS CLI：	AWS 管理員、AWS 系統管理員

任務	描述	所需技能
	`$ aws configure` `AWS Access Key ID` ` [None]: **********` `AWS Secret Access Key` ` [None]: ********` `**********` `Default region name` ` [None]: us-east-1` `Default output format` ` [None]: json`	
將備份映像傳送到 Amazon S3。	之前，線上資料庫備份已儲存至/backup本機目錄。若要將該備份映像傳送至 S3 儲存貯體，請執行下列命令： `aws s3 sync /backup` ` s3://hadrmig-db2/S` `AMPLE_backup`	AWS 管理員、AWS 系統管理員
將 Db2 存檔日誌傳送到 Amazon S3。	將現場部署 Db2 存檔日誌與 Amazon S3 儲存貯體同步，該儲存貯體可由目標 Db2 執行個體在 Amazon EC2 上存取此儲存貯體： `aws s3 sync /db2logs` ` s3://hadrmig-db2/S` `AMPLE_LOGS` 使用 cron 或其他排程工具定期執行此命令。頻率取決於來源資料庫封存交易記錄檔的頻率。	

將 Amazon EC2 上的 Db2 Connect 到 Amazon S3 並開始初始數據庫同步

任務	描述	所需技能
創建一個 PKCS12 密鑰庫。	Db2 使用公開金鑰加密標準 (PKCS) 加密金鑰存放區來確保 AWS 存取金鑰的安全。創建一個密鑰庫，並配置源 Db2 以使用它： `gsk8capicmd_64 -keydb -create -db "/home/db2inst1/.keystore/db2s3.p12" -pw "<password>" -type pkcs12 -stash` `db2 "update dbm cfg using keystore_location /home/db2inst1/.keystore/db2s3.p12 keystore_type pkcs12"`	DBA
建立 Db2 儲存存取權別名。	Db2 使用儲存存取別名，透過INGEST、LOAD或RESTORE DATABASE命令直接存取 Amazon S3。BACKUP DATABASE 因為您已將 IAM 角色指派給 EC2 執行個體，而USER且不PASSWORD是必要的： `db2 "catalog storage access alias <alias_name> vendor S3 server <S3 endpoint>`	DBA

任務	描述	所需技能
	container '<bucket_name>'" 例如，您的指令碼可能如下所示： db2 "catalog storage access alias DB2AWSS3 vendor S3 server s3.us-east-1.amazonaws.com container 'hadrmig-db2'"	

任務	描述	所需技能
設定暫存區域。	我們建議使用DB2_ENABLE_COS_SDK=ON DB2_OBJECT_STORAGE_SETTINGS=EnableStreamingRestore 、和連結至程式awssdk庫來略過 Amazon S3 暫存區進行資料庫備份和還原： #By root: cp -rp /home/db2inst1/sqllib/lib64/awssdk/RHEL/7.6/* /home/db2inst1/sqllib/lib64/ #By db2 instance owner: db2set DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH=/db2stage db2set DB2_ENABLE_COS_SDK=ON db2set DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH=/db2stage db2stop db2start	DBA

任務	描述	所需技能
從備份映像還原資料庫。	從 S3 儲存貯體中的備份映像還原 Amazon EC2 上的目標資料庫： `db2 create db sample on /data1` `db2 restore db sample from DB2REMOTE://DB2AWSS3/hadrmig-db2/SAMPLE_backup replace existing`	DBA

在內部部署沒有 HADR 的情況下設置 HADR

任務	描述	所需技能
將內部部署 Db2 伺服器設定為主要伺服器。	將 HADR db2-server1 (內部部署來源) 的資料庫組態設定更新為主要項目。設置HADR_SYNCMODE 為 SUPERASYNC mode，它具有最短的事務響應時間： `db2 update db cfg for sample using HADR_LOCAL_HOST db2-server1 HADR_LOCAL_SVC 50010 HADR_REMOTE_HOST db2-ec2 HADR_REMOTE_SVC 50012 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000I The UPDATE DATABASE CONFIGURATION command`	DBA

任務	描述	所需技能
	`completed successfully` 預計現場部署資料中心和 AWS 之間的網路延遲。(您可以根據網路可靠性設定不同的HADR_SYNCMODE 值。 如需詳細資訊，請參閱「相關資源」一節)。	
變更目標資料庫記錄檔存檔目的地。	變更目標資料庫日誌存檔目的地以符合 Amazon EC2 環境： `db2 update db cfg for SAMPLE using LOGARCHMETH1 'DB2REMOTE://DB2AWSS3//SAMPLE_LOGS/' LOGARCHMETH2 OFF` `DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully`	DBA

任務	描述	所需技能
在 Amazon EC2 伺服器上為 Db2 設定 HADR。	將 HADR 的資料庫組態更新db2-ec2為待命狀態： `db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCAL_SVC 50012 HADR_REMOTE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully`	DBA

任務	描述	所需技能
驗證哈德爾設置。	驗證來源和目標 Db2 伺服器上的 HADR 參數。 若要驗證安裝程式db2-server1 ，請執行下列命令： `db2 get db cfg for sample|grep HADR` `HADR database role = PRIMARY` `HADR local host name (HADR_LOCAL_HOST) = db2-server1` `HADR local service name (HADR_LOCAL_SVC) = 50010` `HADR remote host name (HADR_REMOTE_HOST) = db2-ec2` `HADR remote service name (HADR_REMOTE_SVC) = 50012` `HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1` `HADR timeout value (HADR_TIMEOUT) = 120` `HADR target list (HADR_TARGET_LIST) =` `HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC` `HADR spool log data limit (4KB)`	DBA

任務	描述	所需技能
	(HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF 若要驗證安裝程式db2-ec2，請執行下列命令： db2 get db cfg for sample|grep HADR HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-ec2 HADR local service name (HADR_LOCAL_SVC) = 50012 HADR remote host name (HADR_REMOTE_HOST) = db2-server1 HADR remote service name (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server	

任務	描述	所需技能
	(HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF HADR_LOCAL_HOST 、HADR_LOCAL_SVC HADR_REMOTE_HOST 、和HADR_REMOTE_SVC 參數表示一個主要和一個待命 HADR 設定。	

任務	描述	所需技能
啟動 Db2 哈德爾執行個體。	請先在待命伺服器db2-ec2上啟動 Db2 HADR 執行個體： `db2 start hadr on db sample as standby` `DB20000I The START HADR ON DATABASE command completed successfully.` 在主要 (來源) 伺服器上啟動 Db2 HADR：db2-server1 `db2 start hadr on db sample as primary` `DB20000I The START HADR ON DATABASE command completed successfully.` 內部部署 Db2 與 Amazon EC2 之間的 HADR 連線現已成功建立。Db2 主要伺服器會即時db2-server1 啟動串流交易db2-ec2記錄檔記錄。	DBA

當 HADR 存在於內部部署時設置 HADR

任務	描述	所需技能
在 Amazon EC2 上新增 Db2 做為輔助待命。	如果 HADR 在現場部署 Db2 執行個體上執行，您可以使用HADR_TARGET_LIST 在上執行下列命令，將 Amazon EC2 上的 Db2 新增為輔助待命：db2-ec2	DBA

任務	描述	所需技能
	db2 update db cfg for sample using HADR_LOCAL_HOST db2-ec2 HADR_LOCAL_SVC 50012 HADR_REMOTE_HOST db2-server1 HADR_REMOTE_SVC 50010 HADR_REMOTE_INST db2inst1 HADR_SYNCMODE SUPERASYNC DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully. db2 update db cfg for sample using HADR_TARGET_LIST "db2-server1:50010|db2-server2:50011" DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.	

任務	描述	所需技能
將輔助待命資訊新增至內部部署伺服器。	HADR_TARGET_LIST 在兩個內部部署伺服器 (主要伺服器和待命) 上的更新。 在上db2-server1 ，執行下列程式碼： db2 update db cfg for sample using HADR_TARGET_LIST "db2-server2:50011|db2-ec2:50012" DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully. SQL1363W One or more of the parameters submitted for immediate modification were not changed dynamically. For these configuration parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective. 在上db2-server2 ，執行下列程式碼： db2 update db cfg for sample using HADR_TARGET_LIST "db2-serv	DBA

任務	描述	所需技能
	er1:50010|db2-ec2: 50012" DB20000I The UPDATE DATABASE CONFIGURATION command completed successfu lly. SQL1363W One or more of the parameter s submitted for immediate modificat ion were not changed dynamically. For these configura tion parameters, the database must be shutdown and reactivated before the configuration parameter changes become effective.	

任務	描述	所需技能
驗證哈德爾設置。	驗證來源和目標 Db2 伺服器上的 HADR 參數。 在上db2-server1 ，執行下列程式碼： `db2 get db cfg for sample|grep HADR` `HADR database role = PRIMARY` `HADR local host name (HADR_LOCAL_HOST) = db2-server1` `HADR local service name (HADR_LOCAL_SVC) = 50010` `HADR remote host name (HADR_REMOTE_HOST) = db2-server2` `HADR remote service name (HADR_REMOTE_SVC) = 50011` `HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1` `HADR timeout value (HADR_TIMEOUT) = 120` `HADR target list (HADR_TARGET_LIST) = db2-server2:50011|db2-ec2:50012` `HADR log write synchronization mode`	

任務	描述	所需技能
	(HADR_SYNCMODE) = NEARSYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF 在上db2-server2 ，執行下列程式碼： db2 get db cfg for sample|grep HADR HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-server2 HADR local service name (HADR_LOCAL_SVC) = 50011 HADR remote host name (HADR_REMOTE_HOST) = db2-server1 HADR remote service name	

任務	描述	所需技能
	(HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server1:50010|db2-ec2:50012 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0 HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF 在上db2-ec2，執行下列程式碼： db2 get db cfg for sample|grep HADR	

任務	描述	所需技能
	HADR database role = STANDBY HADR local host name (HADR_LOCAL_HOST) = db2-ec2 HADR local service name (HADR_LOCAL_SVC) = 50012 HADR remote host name (HADR_REMOTE_HOST) = db2-server1 HADR remote service name (HADR_REMOTE_SVC) = 50010 HADR instance name of remote server (HADR_REMOTE_INST) = db2inst1 HADR timeout value (HADR_TIMEOUT) = 120 HADR target list (HADR_TARGET_LIST) = db2-server1:50010|db2-server2:50011 HADR log write synchronization mode (HADR_SYNCMODE) = SUPERASYNC HADR spool log data limit (4KB) (HADR_SPOOL_LIMIT) = AUTOMATIC(52000) HADR log replay delay (seconds) (HADR_REPLAY_DELAY) = 0	

任務	描述	所需技能
	HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0 HADR SSL certificate label (HADR_SSL_LABEL) = HADR SSL Hostname Validation (HADR_SSL_HOST_VAL) = OFF HADR_LOCAL_HOST 、HADR_LOCAL_SVC HADR_REMOTE_HOST HADR_REMOTE_SVC 、和HADR_TARGET_LIST 參數表示一個主要和兩個待命 HADR 設定。	

任務	描述	所需技能
停止並啟動 Db2 哈德爾。	HADR_TARGET_LIST 現在已在所有三台伺服器上設定。每個 Db2 服務器都知道另外兩個。停止並重新啟動 HADR (短暫中斷) 以利用新的組態。 在上db2-server1 ，執行下列命令： `db2 stop hadr on db sample` `db2 deactivate db sample` `db2 activate db sample` 在上db2-server2 ，執行下列命令： `db2 deactivate db sample` `db2 start hadr on db sample as standby` `SQL1766W The command completed successfully` 在上db2-ec2，執行下列命令： `db2 start hadr on db sample as standby` `SQL1766W The command completed successfully` 在上db2-server1 ，執行下列命令：	DBA

任務	描述	所需技能
	`db2 start hadr on db sample as primary` `SQL1766W The command completed successfully` 內部部署 Db2 與 Amazon EC2 之間的 HADR 連線現已成功建立。Db2 主要伺服器會db2-server1 啟動即時串流交易記錄檔記錄。db2-server2 db2-ec2	

在切換窗口期間將 Amazon EC2 上的 Db2 作為主要

任務	描述	所需技能
請確定待命伺服器上沒有 HADR 延遲。	從主要伺服器db2-server1 檢查 HADR 狀態。當處於REMOTE_CATCHUP 狀態時不要驚慌，HADR_STATE 這在設置為SUPERASYNC 時HADR_SYNCMODE 是正常的。PRIMARY_LOG_TIME 並顯STANDBY_REPLAY_LOG_TIME 示它們是同步的： `db2pd -hadr -db sample` `HADR_ROLE = PRIMARY` `REPLAY_TYPE = PHYSICAL` `HADR_SYNCMODE = SUPERASYNC`	DBA

任務	描述	所需技能
	STANDBY_ID = 2 LOG_STREAM_ID = 0 HADR_STATE = REMOTE_CATCHUP ..... PRIMARY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292) STANDBY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292) STANDBY_REPLAY_LOG_TIME = 10/26/2022 02:11:32.000000 (1666750292)	

任務	描述	所需技能
運行哈德爾接管。	若要完成移轉，請db2-ec2執行 HADR 接管命令來建立主要資料庫。使用指令db2pd來驗證HADR_ROLE 值： `db2 TAKEOVER HADR ON DATABASE sample` `DB20000I The TAKEOVER HADR ON DATABASE command completed successfully.` `db2pd -hadr -db sample` `Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:03:25 -- Date 2022-10-26-02.46.45.048988` `HADR_ROLE = PRIMARY` `REPLAY_TYPE = PHYSICAL` 若要完成 AWS 的移轉作業，請將應用程式連線指向 Amazon EC2 上的 Db2。	

故障診斷

問題	解決方案
如果出於防火牆和安全原因使用 NAT，則主機可能有兩個 IP 位址 (一個內部和一個外部)，這	若要在 NAT 環境中支援 HADR，您可以同時HADR_LOCAL_HOST 使用內部和外部位址進行設定。例如，如果 Db2 服務器具有內部名

問題	解決方案
可能會導致 HADR IP 位址檢查失敗。該START HADR ON DATABASE命令將返回以下消息： HADR_LOCAL_HOST:HADR_LOCAL_SVC (-xx-xx-xx-xx.:50011 (xx.xx.xx.xx:50011)) on remote database is different from HADR_REMOTE_HOST:HADR_REMOTE_SVC (xx-xx-xx-xx.:50011 (x.x.x.x:50011)) on local database.	稱host1和外部名稱host1E，則HADR_LOCAL_HOST 可HADR_LOCAL_HOST: "host1 | host1E"以。

相關資源

- Db2 備份和還原不同作業系統和硬體平台之間的作業
- 設定資料庫存取權限別名和 DB2REMOTE
- Db2 高可用性災難復原
- hadr_syncmode-在對等狀態配置參數中日誌寫入的 HADR 同步模式

使用 HCX 自動化功能移轉 VMware 虛擬機器

創建者：吉里納迪明提（AWS），哈桑·阿德科亞（AWS）和納文·德希瓦爾

環境：生產	資料來源：內部部署或雲端 VMware vCenter 或軟體定義的資料中心	目標：VMware Cloud on AWS
R 類型：重新主機	工作負載：所有其他工作	技術：移轉；混合雲
AWS 服務：VMware Cloud on AWS		

Summary

注意：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。 AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何使用由 VMware PowerCLI 指令碼提供支援的 VMware 混合雲延伸功能 (HCX) 自動化，將 VMware 內部部署虛擬機器 (VM) 移轉至 AWS 上的 VMware 雲端。PowerCLI 是一個建立在視窗 PowerShell上的命令列工具。它可協助您管理 VMware 軟體，並將基礎架構和移轉工作自動化。

您可以調整此模式，以便在 vCenter、軟體定義資料中心 (SDDC) 和雲端環境的任何組合之間進行移轉。此模式隨附的 PowerCLI 指令碼會針對所有 VM 組態和排程工作使用自動化，而非滑鼠按一下，因此可節省移轉活動的時間，並協助降低人為錯誤的風險。

先決條件和限制

先決條件

- 具有軟體定義資料中心的 VMware 雲端 AWS 雲端帳戶
- 現有的內部部署或雲端型 vCenter 或軟體定義的資料中心
- 具有來源和目標 vCenter 或 SDDC 必要權限的使用者帳戶
- HCX 網站配對與來源與目的地 vCenter 或 SDDC 之間設定的 HCX 網路延伸 (HCX-NE)
- 在您選擇的伺服器上安裝

限制

- 如果來源 vCenter 使用跨 vCenter NSX，PowerCLI 模組將無法運作。使用指令碼方法 (例如 Python) 搭配 HCX API 而非 PowerCLI。
- 如果移轉的虛擬機器需要新的名稱或 IP 位址，請使用指令碼方法 (例如 Python) 搭配 HCX API。
- 此模式不會填入必要的 .csv 檔案。您可以使用 VMware vRealize 網路鑑識 (VRNi) 或其他方法來填入檔案。

產品版本

- 第 5 vSphere 或更新版本
- VMware 國際貨運中心 4.4 版或更新版本
- VMware PowerCLI 版本 12.7 或更新版本

架構

源, 技術, 堆棧

- 內部部署或雲端型 VMware

目標技術堆疊

- VMware Cloud on AWS

目標架構

工具

AWS 服務

- VMware Cloud on AWS 服務是由 AWS 和 VMware 共同設計的服務，可協助您將現場部署 VMware 虛擬主機環境遷移並擴展到 AWS 雲端。

其他工具

- VMware 混合雲擴充功能 (HCX) 是一種公用程式，可將工作負載從現場部署 VMware 環境移轉至 VMware Cloud on AWS，而無需變更基礎平台。注意：本產品先前稱為「混合雲延伸功能」和「NSX 混合式 Connect」。此模式會使用 HCX 進行虛擬機器移轉。
- VMware 威力 CLI 是一種用於自動化 VMware vSphere 和虛擬雲端管理的命令列工具。您可以使用指令 PowerShell 程式在 Windows PowerShell 中執行 PowerCLI 命令。這個病毒碼會使用 PowerCLI 來執行移轉命令。

代碼

簡單、獨立的指令碼

我們建議您使用這個單機指令碼進行初始測試，以確認組態選項是否已接受並如預期般運作。如需指示，請參閱《史詩》一節。

```
<# Manual Variables #>
$HcxServer = "[enterValue]"
$SrcNetworkName = "[enterValue]"
$DstNetworkName = "[enterValue]"
$DstComputeName = "[enterValue]"
$DstDSName = "[enterValue]"
$DstFolderName = "[enterValue]"
$vmName = "[enterValue]"

<# Environment Setup #>
Connect-HCXServer -Server $HcxServer
$HcxDstSite = Get-HCXSite -Destination
$HcxSrcSite = Get-HCXSite -Source
$SrcNetwork = Get-HCXNetwork -Name $SrcNetworkName -Type VirtualWire -Site $HcxSrcSite
$DstNetwork = Get-HCXNetwork -Name $DstNetworkName -Type NsxtSegment -Site $HcxDstSite
$DstCompute = Get-HCXContainer -Name $DstComputeName -Site $HcxDstSite
$DstDS = Get-HCXDatastore -Name $DstDSName -Site $HcxDstSite
$DstFolder = Get-HCXContainer -name $DstFolderName -Site $HcxDstSite
$vm = Get-HCXVM -Name $vmName

<# Migration #>
$NetworkMapping = New-HCXNetworkMapping -SourceNetwork $SrcNetwork -DestinationNetwork
 $DstNetwork
$NewMigration = New-HCXMigration -VM $vm -MigrationType vMotion -SourceSite $HcxSrcSite
 -DestinationSite $HcxDstSite -Folder $DstFolder -TargetComputeContainer $DstCompute
 -TargetDatastore $DstDS -NetworkMapping $NetworkMapping -DiskProvisionType Thin
```

```
 -UpgradeVMTools $True -RemoveISOs $True -ForcePowerOffVm $True -RetainMac $True -
UpgradeHardware $True -RemoveSnapshots $True
```

全功能、以 .csv 為基礎的指令碼

測試完成後，您可以在生產環境中使用以下腳本。如需指示，請參閱《史詩》一節。

```
<# Schedule #>
write-host("Getting Time for Scheduling")
$startTime = [DateTime]::Now.AddDays(12)
$endTime = [DateTime]::Now.AddDays(15)

<# Migration #>
Connect-HCXServer -Server [enterValue]
write-host("Getting Source Site")
$HcxSrcSite = Get-HCXSite
write-host("Getting Target Site")
$HcxDstSite = Get-HCXSite -Destination
$HCXVMS = Import-CSV .\Import_VM_list.csv
ForEach ($HCXVM in $HCXVMS) {
    $DstFolder = Get-HCXContainer $HCXVM.DESTINATION_VM_FOLDER -Site $HcxDstSite
    $DstCompute = Get-HCXContainer $HCXVM.DESTINATION_COMPUTE  -Site $HcxDstSite
    $DstDatastore = Get-HCXDatastore $HCXVM.DESTINATION_DATASTORE -Site $HcxDstSite
    $SrcNetwork = Get-HCXNetwork $HCXVM.SOURCE_NETWORK -Type VirtualWire -Site
 $HcxSrcSite
    $DstNetwork = Get-HCXNetwork $HCXVM.DESTINATION_NETWORK -Type NsxtSegment -Site
 $HcxDstSite
    $NetworkMapping = New-HCXNetworkMapping -SourceNetwork $SrcNetwork -
DestinationNetwork $DstNetwork
    $NewMigration = New-HCXMigration -VM (Get-HCXVM $HCXVM.VM_NAME) -MigrationType
 Bulk -SourceSite $HcxSrcSite -DestinationSite $HcxDstSite -Folder $DstFolder -
TargetComputeContainer $DstCompute -TargetDatastore $DstDatastore -NetworkMapping
 $NetworkMapping -DiskProvisionType Thin -UpgradeVMTools $True -RemoveISOs $True -
ForcePowerOffVm $True -RetainMac $True -UpgradeHardware $True -RemoveSnapshots $True -
ScheduleStartTime $startTime -ScheduleEndTime $endTime
    Start-HCXMigration -Migration $NewMigration -Confirm:$false
}
```

史诗

收集手動變數的資訊

任務	描述	所需技能
尋找來源和目的地 vCenter 和軟體定義的資料中心伺服器名稱。	PowerCLI 指令碼需要這個史詩中描述的變數。您可以事先收集這些資訊，以便於使用指令碼。 在 vSphere 主控台的 HCX 區段中，選擇基礎結構，網站配對。記下顯示的來源和目的地伺服器名稱。	雲端架構師
尋找來源和目的地 HCX 名稱。	在 vSphere 主控台的 HCX 區段中，選擇系統，管理。記下顯示的來源和目的地 HCX 名稱。	雲端架構師
尋找來源和目的地網路名稱。	在 vSphere 主控台的 HCX 區段中，選擇系統，網路延伸。記下來源和目標網路名稱。 備註：或者，您可以在連線至 HCX 伺服器後，使用 PowerCLI Get-HCx 網路和取得 HCX 網路-目的地命令來取得來源和目的地網路名稱。	雲端架構師
從 vSphere 主控台收集其他資訊。	在 vSphere 主控台上，收集下列資訊： • 您要移轉的虛擬機器名稱 • 目的地運算環境 (叢集/主機) • 目的地儲存 • 目的地 VM 資料夾名稱	雲端架構師

做出移轉決策

任務	描述	所需技能
決定移轉選項。	決定下列項目： • MigrationType — HCX 輔助移轉類型為 vMotion、大量、冷移轉和 RAV。您的選擇取決於停機時間需求、網路頻寬、移轉時間範圍和工作負載類型。如需詳細資訊，請參閱 AWS 部落格文章，將工作負載移轉至具有混合雲延伸功能 (HCX) 的 AWS 上的 VMware 雲端。 • DiskProvisionType (Thin, Thick) • UpgradeVMTools ($True, $False) • RemoveISOs ($True, $False) • ForcePowerOffVm ($True, $False) • RetainMac ($True, $False) • UpgradeHardware ($True, $False) • RemoveSnapshots ($True, $False) 如需每個選項的詳細資訊，請參閱 VMware 開發人員說明文件。	雲端架構師

運行簡單腳本進行初始測試

任務	描述	所需技能
複製指令碼。	腳本的簡單版本在單個文件中是獨立的。您可以使用它來測試單一機器的移轉。 從此病毒碼的「程式碼」區段複製第一個指令碼，並將其儲存在已安裝 VMware PowerCLI 模組的電腦上。若要安裝 PowerCLI，請依照 VMware 說明文件中的指示進行。)	雲端架構師
設置腳本變量。	設置腳本`Manual Variables`部分中的所有變量。	雲端架構師
設定移轉變數。	`New-HCXMigration`設置腳本`Migration`部分中的所有設置。	雲端架構師
指定工址。	(選擇性) 如果來源或目的地有多個網站，請在指令碼的`Environment Setup`區段中手動指定網站。 如果源和目的地具有單個站點，腳本將自動查找信息。	雲端架構師
執行指令碼。	在安裝 PowerCLI 的伺服器上，從提高權限的 PowerShell 視窗執行指令碼，並在出現提示時輸入認證。	雲端架構師
驗證指令碼。	確認已啟動虛擬機器移轉。	雲端架構師

執行功能完整的指令碼以移轉多個 VM

任務	描述	所需技能
建立並填入 .csv 檔案。	建立在電腦Import_VM _list.csv 上呼叫的 .csv 檔案，並在其中填入下列範例內容： `VM_NAME,DESTINATIO N_VM_FOLDER,DESTIN ATION_COMPUTE,DEST INATION_DATASTORE, SOURCE_NETWORK,DES TINATION_NETWORK [enterValue],[enter Value],[enterValue ],[enterValue],[en terValue],[enterVa lue]` 將 .csv 檔案[enterVal ue] 中的每個檔案取代為您先前收集的資訊。 備註：您可以使用 VMware vRealize 網路鑑識 (VRNi) 或其他方法來填入 .csv 檔案。	雲端架構師
複製指令碼。	功能完整的指令碼版本會使用外部 .csv 檔案中的資訊來自動移轉多個虛擬機器。 從此病毒碼的「程式碼」區段複製第二個指令碼，並將其儲存在已安裝 VMware PowerCLI 模組的電腦上，與 .csv 檔案位於相同的資料夾中。	雲端架構師
修改指令碼。	編輯指令碼以進行下列變更：	雲端架構師

任務	描述	所需技能
	• 第 7 行：設定 HCX 伺服器變數 (`Connect-HCXServer`)。 • 第 12 行：（可選）如果您以不同的方式設定 .csv 檔案名稱，請更新它。 • 明細行 3-4：(選擇性) 設定排程。 • 第 20 行：（選擇性）在`Migration` 區段中指定`New-HCXMigration` 設定。 • 第 9 行與第 11 行：(選擇性) 如果來源或目的地包含多個網站，請手動指定所需的地點。	
執行指令碼。	在安裝 PowerCLI 的伺服器上，從提高權限的 PowerShell 視窗執行指令碼，並在出現提示時輸入認證。	雲端架構師
驗證指令碼。	確認已啟動虛擬機器移轉。	雲端架構師

故障診斷

問題	解決方案
指令碼失敗並顯示錯誤訊息： 「所有源網絡未映射到目標！」	如果來源 vCenter 使用跨 vCenter NSX，PowerCLI 模組將無法運作。使用指令碼方法 (例如 Python) 搭配 HCX API 而非 PowerCLI。這是 PowerCLI 指令碼的已知限制。
指令碼失敗並顯示錯誤訊息：	您輸入的認證不會提供必要的權限。

問題	解決方案
「連接-HCX 服務器錯誤：未經授權」	

相關資源

- 使用混合雲擴充功能 (HCX) 將工作負載移轉至 AWS 上的 VMware 雲端 (AWS 部落格文章)
- 選擇移轉方法，將 VMware 應用程式和工作負載重新定位到 AWS 雲端 (AWS Prescriptive Guidance)
- 使用 VMware HCX (AWS 規範指引)，將 VMware 軟體定義的資料定義中心遷移至 AWS 上的 VMware 雲端
- 開始使用 HCX 模組 (VMware 部落格文章)

將 F5 大 IP 工作負載遷移到 AWS 雲端上的 F5 大 IP VE

創建者威爾·鮑爾（AWS）

資料來源：F5 大 IP 轉運管理系統 13.1 及更新版本	目標：AWS 上的 F5 大 IP VE	R 類型：重新主機
環境：生產	技術：移轉；安全性、身分識別、合規性；網路	工作負載：所有其他工作
AWS 服務：Amazon EC2；Amazon VPC；AWS Transit Gateway；Amazon；Amazon CloudFront CloudWatch；AWS Global Accelerator；AWS CloudFormation		

Summary

Organizations 希望遷移到 Amazon Web Services (AWS) 雲端，以提高靈活性和彈性。將 F5 BIG-IP 安全和流量管理解決方案遷移到 AWS 雲端後，您可以專注於在整個企業架構中高價值操作模型的靈活性和採用。

此模式說明如何將 F5 BIG-IP 工作負載移轉至 AWS 雲端上的 F5 BIG-IP 虛擬版本 (VE) 工作負載。透過重新裝載現有環境並部署重新平台化的層面，例如服務探索和 API 整合，來移轉工作負載。AWS CloudFormation 範本可加速您的工作負載移轉至 AWS 雲端。

此模式適用於遷移 F5 安全和流量管理解決方案的技術工程和架構團隊，並隨附 AWS 規範指導網站上 AWS 雲端上的 F5 BIG-IP 移轉至 F5 BIG-IP VE 的指南。

先決條件和限制

先決條件

- 現有的內部部署 F5 大 IP 工作負載。
- 適用於大 IP VE 版本的現有 F5 授權。
- 作用中的 AWS 帳戶

- 現有的虛擬私有雲端 (VPC) 透過 NAT 閘道或彈性 IP 地址設定出口，並可存取下列端點進行設定：Amazon 簡單儲存服務 (Amazon S3)、亞馬遜彈性運算雲端 (Amazon EC2)、AWS Security Token Service (AWS STS) 和亞馬遜。 CloudWatch您也可以將模組化且可擴充的 VPC 架構快速入門修改為部署的建置區塊。
- 一或兩個現有的可用區域，視您的需求而定。
- 每個可用區域中有三個現有的私有子網路。
- AWS CloudFormation 範本，可在 F5 GitHub 儲存庫中使用。

在移轉期間，視需求而定，您也可以使用下列項目：

- F5 雲端容錯移轉延伸模組，用於管理彈性 IP 位址對應、次要 IP 對應和路由表變更。
- 如果您使用多個可用區域，則需要使用 F5 雲端容錯移轉延伸模組來處理虛擬伺服器的彈性 IP 對應。
- 您應該考慮使用 F5 應用程式服務 3 (AS3)、F5 應用程式服務範本 (FAST) 或其他基礎結構作為程式碼 (IaC) 模型來管理組態。準備 IaC 模型中的配置並使用代碼存儲庫將有助於遷移和持續的管理工作。

专业

- 此模式需要熟悉如何將一個或多個 VPC 連接到現有的資料中心。如需有關此功能的詳細資訊，請參閱 Amazon VPC 文件中的網路到 Amazon 虛擬私人雲端連線選項。
- F5 產品和模組也需要熟悉，包括流量管理作業系統 (TMOS)、本機流量管理員 (LTM)、全球流量管理員 (GTM)、存取原則管理員 (APM)、應用程式安全管理員 (ASM)、進階 Firewall Manager 員 (AFM) 和 BIG-IQ。

產品版本

- 雖然此模式支援 F5 大 IP 12.1 版或更新版本，但我們建議您使用 F5 大 IP 13.1 版或更新版本。

架構

源, 技術, 堆棧

- F5 大 IP 工作負載

目標技術堆疊

- Amazon CloudFront
- Amazon CloudWatch
- Amazon EC2
- Amazon S3
- Amazon VPC
- AWS Global Accelerator
- AWS STS
- AWS Transit Gateway
- F5 大 IP

目標架構

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon CloudFront 透過全球資料中心網路提供您的 Web 內容，加快 Web 內容的分發速度，進而降低延遲並提升效能。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Security Token Service (AWS STS) 可協助您為使用者申請臨時、有限權限的登入資料。
- AWS Transit Gateway 是連接虛擬私有雲 (VPC) 和現場部署網路的中央中樞。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

史诗

探索與評估

任務	描述	所需技能
評估 F5 大 IP 的性能。	收集並記錄虛擬伺服器上應用程式的效能測量結果，以及將要移轉的系統指標。這將有助於正確調整目標 AWS 基礎設施的大小，以實現更好的成本優化。	F5 建築師，工程師和網絡架構師，工程師
評估 F5 大 IP 作業系統和組態。	評估要移轉的物件，以及是否需要維護網路結構 (例如 VLAN)。	F5 建築師、工程師
評估 F5 授權選項。	評估您需要的授權和消費模式。此評估應根據您對 F5 BIG-IP 作業系統和組態的評估進行評估。	F5 建築師、工程師
評估公共應用程序。	判斷哪些應用程式需要公用 IP 位址。將這些應用程式與所需的執行個體和叢集配對，以符合效能和服務等級協定 (SLA) 需求。	F5 架構師、雲端架構師、網路架構師、工程師、應用程式團隊
評估內部應用程式。	評估內部使用者將使用哪些應用程式。確保您知道這些內部使用者在組織中的位置，以及這些環境如何連接到 AWS 雲端。您也應該確定這些應用程式可以使用網域名稱系統 (DNS) 做為預設網域的一部分。	F5 架構師、雲端架構師、網路架構師、工程師、應用程式團隊

任務	描述	所需技能
最終確定 AMI。	並非所有 F5 BIG-IP 版本都會建立為 Amazon 機器映像 (AMI)。如果您有特定的必要快速修復工程 (QFE) 版本，可以使用 F5 BIG-IP 影像產生器工具。如需有關此工具的詳細資訊，請參閱 < 相關資源 > 一節。	F5 架構師、雲端架構師、工程師
完成執行個體類型和架構。	決定執行個體類型、VPC 架構和互連架構。	F5 架構師、雲端架構師、網路架構師、工程師

完成安全性與法規遵循相關活動

任務	描述	所需技能
記錄現有的 F5 安全性原則。	收集並記錄現有的 F5 安全性原則。確保您在安全的代碼存儲庫中創建它們的副本。	F5 建築師、工程師
對 AMI 進行加密。	(選擇性) 您的組織可能需要對靜態資料進行加密。如需有關建立自訂使用授權 (BYOL) 映像的詳細資訊，請參閱 < 相關資源 > 一節。	F5 架構師、工程師雲端架構師、工程師
硬化設備。	這將有助於防止潛在的漏洞。	F5 建築師、工程師

設定您的新 AWS 環境

任務	描述	所需技能
建立邊緣和安全性帳戶。	登入 AWS 管理主控台並建立可提供和操作邊緣和安全服務	雲端架構師、工程師

任務	描述	所需技能
	的 AWS 帳戶。這些帳戶可能與為共用服務和應用程式操作 VPC 的帳戶不同。此步驟可作為 landing zone 的一部分完成。	
部署邊緣和安全 VPC。	設定和設定提供邊緣和安全性服務所需的 VPC。	雲端架構師、工程師
Connect 至來源資料中心。	Connect 至託管 F5 BIG-IP 工作負載的來源資料中心。	雲端架構師、網路架構師、工程師
部署 VPC 連線。	將邊緣和安全服務 VPC Connect 到應用程式 VPC。	網路架構師、工程師
部署執行個體。	使用「相關資源」部分中的 AWS CloudFormation 範本部署執行個體。	F5 建築師、工程師
測試和設定執行個體容錯移轉	請確定已設定 AWS 進階 HA iApp 範本或 F5 雲端容錯移轉延伸模組並正常運作。	F5 建築師、工程師

設定聯網

任務	描述	所需技能
準備 VPC 拓撲。	開啟 Amazon VPC 主控台，並確定您的 VPC 人雲端具有 F5 BIG-IP VE 部署的所有必要子網路和防護。	網路架構師、F5 架構師、雲端架構師、工程師
準備您的 VPC 端點。	如果 F5 大 IP 工作負載無法存取 TMM 界面上的 NAT 閘道或彈性 IP 地址，請準備適用於	雲端架構師、工程師

任務	描述	所需技能
	Amazon EC2、Amazon S3 和 AWS STS 的 VPC 端點。	

移轉資料

任務	描述	所需技能
移轉組態。	將 F5 大 IP 組態遷移到 AWS 雲端上的 F5 大 IP VE。	F5 建築師、工程師
關聯次要 IP。	虛擬伺服器 IP 位址與指派給執行個體的次要 IP 位址有關係。指派次要 IP 位址，並確定已選取「允許重新對應/重新指派」。	F5 建築師、工程師

測試配置

任務	描述	所需技能
驗證虛擬伺服器組態。	測試虛擬伺服器。	F5 架構師、應用程式團隊

完成作業

任務	描述	所需技能
建立備份策略。	必須關閉系統才能建立完整快照。如需詳細資訊，請參閱 < 相關資源 > 一節中的 < 更新 F5 BIG-IP 虛擬機器 > 一節。	F5 架構師、雲端架構師、工程師
建立叢集容錯移轉執行簿。	請確定容錯移轉 Runbook 程序已完成。	F5 建築師、工程師

任務	描述	所需技能
設定和驗證記錄。	設定 F5 遙測串流，將記錄檔傳送至所需的目的地。	F5 建築師、工程師

完成切換

任務	描述	所需技能
切換到新的部署。		F5 架構師、雲端架構師、網路架構師、工程師 AppTeams

相關資源

移轉指南

- 在 AWS 雲端上從 F5 大 IP 遷移到 F5 大 IP VE

中五資源

- F5 GitHub 儲存庫中的 AWS CloudFormation 範本
- AWS Marketplace 中的 F5
- F5 大 IP VE 概觀
- 快速入門示例-使用 WAF（LTM + ASM）的 BIG-IP 虛擬版
- AWS 上的 F5 應用程式服務：概觀 (影片)
- F5 應用服務 3 擴充功能使用者指南
- F5 雲端文件
- F5 圖標控制休息維基
- F5 單一組態檔案的概觀 (11.x-15.x)
- F5 拓撲實驗室
- F5 白皮書
- F5 大 IP 圖像生成器工具
- 更新 F5 大 IP VE 虛擬機器

- UCS 歸檔「平台移轉」選項概述

使用二進位方法將現場部署 Go Web 應用程式遷移到 AWS Elastic Beanstalk

由蘇哈斯巴薩瓦拉（AWS）和舒馬茲穆赫塔爾卡濟（AWS）創建

環境：PoC 或試點	資料來源：應用	目標：Elastic Beanstalk
R 類型：重新主機	技術：遷移；Web 和移動應用程序	AWS 服務：AWS Elastic Beanstalk

Summary

此模式說明如何將現場部署 Go Web 應用程式遷移到 AWS Elastic Beanstalk。應用程式遷移後，Elastic Beanstalk 會為來源服務包建立二進位檔，並將其部署到 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。

作為轉載遷移策略，此模式的方法很快，不需要更改代碼，這意味著更少的測試和遷移時間。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 內部部署 Go Web 應用程式。
- 包含 Go 應用程序源代碼的 GitHub 存儲庫。如果您不使用 GitHub，還有其他方法可以為 Elastic Beanstalk 建立應用程式來源套裝軟體。

產品版本

- Elastic Beanstalk 支持的最新 Go 版本。如需詳細資訊，請參閱 Elastic Beanstalk 文件。

架構

源, 技術, 堆棧

- 內部部署 Go Web 應用程式

目標技術堆疊

- AWS Elastic Beanstalk
- Amazon CloudWatch

目標架構

工具

- AWS Elastic Beanstalk 可在 AWS 雲端快速部署和管理應用程式，使用者無須了解執行這些應用程式的基礎設施。Elastic Beanstalk 可降低管理複雜性而不會限制選擇或控制。
- GitHub是一個開源的分佈式版本控制系統。

史诗

建立 Go Web 應用程式來源套件 .zip 檔案

任務	描述	所需技能
建立 Go 應用程式的來源套裝軟體。	打開包含 Go 應用程序源代碼的 GitHub 存儲庫並準備源包。來源套裝軟體在根目錄中包含一個`application.go` 來源檔案，該檔案託管 Go 應用程式的主要套件。如果您未使用 GitHub，請參閱此模式稍早的「必要條件」一節，瞭解建立應用程式來源套裝軟體的其他方法。	系統管理、應用程式開發
建立一個程式組態檔案。	在源包中創建一個文件`.ebextensions` 夾，然後在此`options.config` 文件夾中創建一個文件。如需詳細資訊，請參閱 Elastic Beanstalk 文件。	系統管理、應用程式開發
建立來源服務包 .zip 檔案。	執行下列命令。	系統管理、應用程式開發

任務	描述	所需技能
	`git archive -o ../godemoapp.zip HEAD` 這會建立來源套件 .zip 檔案。下載 .zip 檔案並儲存為本機檔案。 重要事項：.zip 檔案不能超過 512 MB，且無法包含父資料夾或頂層目錄。	

將 Go Web 應用程序遷移到 Elastic Beanstalk

任務	描述	所需技能
選擇 Elastic Beanstalk 應用程序。	1. 登入 AWS 管理主控台並開啟 Elastic Beanstalk 主控台。 2. 從區域清單中選擇您的 AWS 區域。 3. 在瀏覽窗格中，選擇「應用程式」，然後選擇現有的 Elastic Beanstalk 應用程式或建立應用程式。 有關如何創建 Elastic Beanstalk 應用程序的說明，請參閱 E lastic Beanstalk 文檔。	系統管理、應用程式開發
啟動 Elastic Beanstalk 網頁伺服器環境。	1. 在 [應用程式概觀] 頁面上，選擇 [建立新環境]，然後選擇 [Web 伺服器環境]。 2. 填寫「環境名稱」和「網域名稱」欄位。	系統管理、應用程式開發

任務	描述	所需技能
	3. 選擇平台版本，然後選擇轉到作為您的平台。	
將源包 .zip 文件上傳到 Elastic Beanstalk。	1. 在 [應用程式碼] 中，選擇 [上傳程式碼]，然後選擇 [本機檔案]。 2. 選擇包含來源套件的 .zip 檔案。 3. 在 [版本] 標籤中，為檔案指定唯一的名稱，然後選擇 [建立環境]。	系統管理、應用程式開發
測試已部署的 Go Web 應用程式。	您將被重定向到 Elastic Beanstalk 應用程序的概述頁面。在概觀頂端的「環境 ID」旁邊，選擇結尾的 URL 以導覽`elasticbeanstalk.com` 至您的應用程式。您的應用程式必須在其組態檔案中使用此名稱作為環境變數，並將其顯示在網頁上。	系統管理、應用程式開發

故障診斷

問題	解決方案
無法透過應用程式 Application Load Balancer。	檢查包含 Elastic Beanstalk 應用程序的目標組。如果狀況不佳，請登入 Elastic Beanstalk 執行個體並檢查`nginx.conf` 檔案組態，以確認其路由到正確的健康狀態 URL。您可能需要變更目標群組健全狀況檢查 URL。

相關資源

- Elastic Beanstalk 支持的 Go 平台版本
- 使用配置文件與 Elastic Beanstalk
- 在 Elastic Beanstalk 中創建示例應用程序

使用適用於 SFTP 的 AWS 轉移，將現場部署 SFTP 伺服器遷移到 AWS

由阿卡什·庫馬爾（AWS）創建

環境：生產	來源：存儲	目標：Amazon S3
R 類型：重新主機	技術：移轉、儲存與備份、Web 與行動應用程式	AWS 服務：Amazon S3；AWS Transfer Family；Amazon CloudWatch 日誌

Summary

此模式說明如何使用 AWS 傳輸 SFTP 服務，將使用安全殼層 (SSH) 檔案傳輸協定 (SFTP) 的現場部署檔案傳輸解決方案遷移到 Amazon Web Services 務 (AWS) 雲端。使用者通常透過其網域名稱或固定 IP 連線至 SFTP 伺服器。這種圖案涵蓋了兩種情況。

AWS Transfer for SFTP 是 AWS Transfer Family 的成員。這是一種安全的傳輸服務，可用來透過 SFTP 將檔案傳入和傳出 AWS 儲存服務。您可以將 AWS 轉移用於 SFTP 與亞馬遜簡單儲存服務 (Amazon S3) 或亞馬 Amazon Elastic File System (Amazon EFS) 搭配使用。此模式使用 Amazon S3 進行存儲。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的 SFTP 網域名稱或固定的 SFTP 網域名稱。

限制

- 您可以在一個請求中傳輸的最大對象是目前 5 GiB。對於大於 100 MiB 的檔案，請考慮使用 Amazon S3 分段上傳。

架構

源, 技術, 堆棧

- 內部部署平面檔案或資料庫傾印檔案。

目標技術堆疊

- AWS Transfer for SFTP
- Amazon S3
- Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM) 角色和政策
- 彈性 IP 地址
- 安全群組
- Amazon CloudWatch 日誌（可選）

目標架構

自動化和規模

若要自動化此模式的目標架構，請使用附加的 AWS CloudFormation 範本：

- `amazon-vpc-subnets.yml`佈建具有兩個公有子網路和兩個私有子網路的虛擬私有雲 (VPC)。
- `amazon-sftp-server.yml`提供 SFTP 伺服器。
- `amazon-sftp-customer.yml`添加用戶。

工具

AWS 服務

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。此模式使用 Amazon S3 做為檔案傳輸的儲存系統。
- 適用於 SFTP 的 AWS 傳輸可協助您透過 SFTP 協定將檔案傳入和傳出 AWS 儲存服務。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

史诗

建立 VPC

任務	描述	所需技能
使用子網路建立 VPC。	前往 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。建立具有兩個公有子網路的虛擬私有雲 (VPC)。(第二個子網路提供高可用性。) —或— 您可以在CloudFormation 控制台中部署附加的 CloudFormation 模板`amazon-vpc-subnets.yml` ，以自動執行此史詩中的任務。	系統管理員開發人員
新增網際網路閘道。	佈建網際網路閘道並將其連接到 VPC。	系統管理員開發人員
遷移現有的 IP。	將現有 IP 附加至彈性 IP 位址。您可以從地址池創建彈性 IP 地址並使用它。	系統管理員開發人員

佈建 SFTP 伺服器

任務	描述	所需技能
建立一個 SFTP 伺服器。	在 https://console.aws.amazon.com/transfer/ 開啟 AWS Transfer Family 主控台。遵循 AWS Transfer Family 文件中為您的伺服器建立網際網路對向端點中的指示，建立具有	系統管理員開發人員

任務	描述	所需技能
	面向網際網路的端點的 SFTP 伺服器。針對「端點類型」，選擇「VPC 託管」。若要存取，請選擇 [網際網路面對 對於 VPC，請選擇您在之前史詩中創建的 VPC。 —或— 您可以在CloudFormation 控制台中部署附加的 CloudFormation 模板`amazon-sftp-server.yml` ，以自動執行此史詩中的任務。	
移轉網域名稱。	將現有網域名稱附加到自訂主機名稱。如果您使用新的網域名稱，請使用 Amazon Route 53 DNS 別名。對於現有的網域名稱，請選擇「其他 DNS」。如需詳細資訊，請參閱 AWS Transfer Family 文件中的使用自訂主機名稱。	系統管理員開發人員
新增記 CloudWatch 錄角色。	（可選）如果要啟用 CloudWatch 記錄，請使用 Lo CloudWatch gs API 操作`logs:CreateLogGroup` 、`logs:CreateLogStream` `logs:DescribeLogStreams` 、和建立`Transfer` 角色`logs:PutLogEvents` 。如需詳細資訊，請參閱 AWS Transfer Family 文件 CloudWatch中的日誌活動。	開發人員，系統管理

任務	描述	所需技能
儲存並提交。	選擇儲存。對於「動作」，請選擇「啟動」，然後等待 SFTP 伺服器建立狀態為「線上」。	系統管理員開發人員

將彈性 IP 位址對應至 SFTP 伺服器

任務	描述	所需技能
停止伺服器，以便您可以修改設定。	在 AWS Transfer Family 主控台上，選擇 [伺服器]，然後選取您建立的 SFTP 伺服器。針對 Actions (動作)，選擇 Stop (停止)。當伺服器離線時，請選擇 [編輯] 修改其設定。	開發人員，系統管理
選擇可用區域和子網路。	在 [可用區域] 區段中，選擇 VPC 的可用區域和子網路。	系統管理員開發人員
新增彈性 IP 位址。	針對 IPv4 位址，請為每個子網路選擇一個彈性 IP 位址，然後選擇 [儲存]。	系統管理員開發人員

新增使用者

任務	描述	所需技能
為使用者建立 IAM 角色以存取 S3 儲存貯體。	為 S3 儲存貯體名稱建立 IAM 角色 `s3:ListBucket` `s3:GetBucketLocation` ，並`s3:PutObject` 將其新增為資源。`Transfer`如需詳細資訊，請參閱 AWS Transfer	系統管理員開發人員

任務	描述	所需技能
	Family 文件中的建立 IAM 角色和政策。 —或— 您可以在CloudFormation 控制台中部署附加的 CloudFormation 模板`amazon-sftp-customer.yml` ，以自動執行此史詩中的任務。	
建立 S3 儲存貯體。	為應用程式建立 S3 儲存貯體。	系統管理員開發人員
建立選擇性資料夾。	(選擇性) 如果您想要為使用者個別存放檔案，請在特定的 Amazon S3 資料夾中，視需要新增資料夾。	系統管理員開發人員
建立安全殼層公開金鑰。	若要建立 SSH key pair，請參閱 AWS Transfer Family 文件中的產生 SSH 金鑰。	系統管理員開發人員
新增使用者。	在 AWS Transfer Family 主控台上，選擇 [伺服器]，選取您建立的 SFTP 伺服器，然後選擇 [新增使用者]。對於主目錄，選擇您建立的 S3 儲存貯體。對於 SSH 公開金鑰，請指定 SSH key pair 的公開金鑰部分。新增 SFTP 伺服器的使用者，然後選擇 [新增]。	系統管理員開發人員

測試 SFTP 伺服器

任務	描述	所需技能
更新安全性群組。	在 SFTP 伺服器的「安全群組」區段中，新增測試機器的 IP 以取得 SFTP 存取權。	開發人員
使用 SFTP 用戶端公用程式來測試伺服器。	使用任何 SFTP 用戶端公用程式測試檔案傳輸。如需用戶端清單和指示，請參閱 AWS 傳輸系列文件中的使用用戶端傳輸檔案。	開發人員

相關資源

- AWS Transfer Family 使用者指南
- Amazon S3 用戶指南
- Amazon EC2 文件中的彈性 IP 地址

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS 應用程式遷移服務將現場部署虛擬機器遷移到 Amazon EC2

創建者：阮清（AWS）

環境：生產	來源：內部部署虛擬機	目標：Amazon EC2
R 類型：重新主機	技術：遷移	AWS 服務：AWS 應用程式遷移服務；Amazon EC2；Amazon EBS

Summary

在應用程式遷移方面，組織可以採取不同的方法將應用程式的伺服器從現場部署環境重新託管 (升降和轉移) 到 Amazon Web Services (AWS) 雲端。一種方法是佈建新的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，然後從頭開始安裝和設定應用程式。另一種方法是使用第三方或 AWS 原生遷移服務同時遷移多個伺服器。

此模式概述了使用 AWS 應用程式遷移服務將支援的虛擬機器 (VM) 遷移到 AWS 雲端上的 Amazon EC2 執行個體的步驟。您可以使用此模式中的方法，逐一移轉一或多部虛擬機器，或根據概述的步驟建立適當的自動化指令碼來自動移轉一或多部虛擬機器。

先決條件和限制

先決條件

- 其中一個支援應用程式遷移服務的 AWS 區域中的有效 AWS 帳戶
- 透過私有網路，使用 AWS Direct Connect 或虛擬私有網路 (VPN)，或透過網際網路，來源伺服器與目標 EC2 伺服器之間的網路連線

限制

- 如需支援區域的最新清單，請參閱支援的 AWS 區域。
- 如需支援的作業系統清單，請參閱支援的作業系統和 Amazon EC2 常見問答集的一般部分。

架構

源, 技術, 堆棧

- 執行 Amazon EC2 支援之作業系統的實體、虛擬或雲端託管伺服器

目標技術堆疊

- 執行與來源虛擬機器相同作業系統的 Amazon EC2 執行個體
- Amazon Elastic Block Store (Amazon EBS)

來源與目標架構

下圖顯示了解決方案的高級架構和主要組件。在內部部署資料中心中，有具有本機磁碟的虛擬機器。在 AWS 上，有一個包含複寫伺服器的暫存區，以及具有 EC2 執行個體的遷移資源區域，用於測試和切換。這兩個子網路都包含 EBS 磁碟區。

1. 初始化 AWS 應用程式遷移服務。
2. 設定暫存區伺服器組態和報表，包括暫存區資源。
3. 在來源伺服器上安裝代理程式，並使用連續區塊層級資料複寫（壓縮和加密）。
4. 自動化協調流程和系統轉換，以縮短切換時間。

網路架構

下圖顯示從網路角度來看解決方案的高階架構和主要元件，包括現場部署資料中心和 AWS 上主要元件之間通訊所需的通訊協定和連接埠。

工具

- AWS 應用程式遷移服務可協助您將應用程式重新託管 (提升和轉移) 到 AWS 雲端，而且停機時間最短。

最佳實務

- 切換至目標 EC2 執行個體之前，請勿將來源伺服器離線或重新開機。
- 為用戶提供充足的機會，在目標服務器上執行用戶接受度測試（UAT），以識別和解決任何問題。理想情況下，此測試應在切換前至少兩週開始。

- 經常在應用程式移轉服務主控台上監視伺服器複寫狀態，以及早發現問題。
- 使用臨時的 AWS Identity and Access Management (IAM) 登入資料進行代理程式安裝，而非永久 IAM 使用者登入資料。

史诗

產生 AWS 登入資料

任務	描述	所需技能
建立 AWS 複寫代理程式 IAM 角色。	使用 AWS 帳戶的管理許可登入。 在 AWS Identity and Access Management (IAM) 主控台上，建立 IAM 角色： 1. 在 IAM 主控台上，選擇 [角色]。 2. 選擇建立角色。 3. 在 [選取受信任的實體] 頁面的 [受信任的實體類型] 區段中，選取 AWS 帳戶。 4. 在「AWS 帳戶」區段中，選取「這個帳戶 (帳戶 ID >)」。 5. 選擇下一步。 6. 在 [新增權限] 頁面上，搜尋`AWSApplicationMigrationAgentInstallationPolicy` 原則，選取原則名稱旁邊的核取方塊。 7. 選擇下一步。 8. 在角色詳細資料頁面上，輸入 MGN_Agent 安裝角色作為角色名稱。	AWS 管理員、移轉工程師

任務	描述	所需技能
	9. 確認欄位是否正確，然後選擇 [建立角色]。	
產生臨時安全登入資料。	在已安裝 AWS Command Line Interface (AWS CLI) (AWS CLI) 的機器上，使用管理許可登入。或者 (在受支援的 AWS 區域內)，在 AWS 管理主控台使用管理許可登入 AWS 帳戶，然後開啟 AWS CloudShell。 使用下列命令產生臨時登入資料，並<account-id> 以 AWS 帳戶 ID 取代。 `aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/MGN_Agent_Installation_Role --role-session-name mgn_installation_session_role` 從命令的輸出中，複製AccessKeyId **SecretAccessKey** 、和的值**SessionToken** 。 將它們存放在安全的位置以備以後使用。 重要事項：這些臨時登入資料將在一小時後過期。如果您在一小時後需要認證，請重複上述步驟。	AWS 管理員、移轉工程師

初始化應用程式移轉服務並建立複製設定範本

任務	描述	所需技能
初始化服務。	在主控台上，以管理許可登入 AWS 帳戶。 選擇應用程式遷移服務，然後選擇 [開始使用]。	AWS 管理員、移轉工程師
建立並設定複製設定範本。	1. 提供下列組態詳細資料： a. 選取暫存區子網路。 b. 選取複製伺服器執行個體類型 (`t3.small`依預設)。 c. 選取 EBS 磁碟區類型 (預設為 gp3)。 d. 選取 EBS 加密選項。 e. 確定已選取 [一律使用應用程式移轉服務安全性群組] 核取方塊。 f. 如果您在現場部署環境和 AWS 之間使用私有網路連線 DirectConnect，請選取 [使用私有 IP 進行資料複寫 (VPN、VPC 對等互連)] 核取方塊。 g. 如果您要限制應用程式移轉服務的網路頻寬，請選取「節流網路頻寬 (每部伺服器-以 Mbps 為單位)」核取方塊。 2. 選擇 Create template (建立範本)。	AWS 管理員、移轉工程師

任務	描述	所需技能
	應用程式遷移服務會自動建立所有必要的 IAM 角色，以促進資料複寫和啟動遷移的伺服器。	

在來源機器上安裝 AWS 複寫代理程式

任務	描述	所需技能
準備好所需的 AWS 登入資料。	當您在來源伺服器上執行安裝程式檔案時，您將需要輸入先前產生的暫時認證`AccessKeyId` ，包括`SecretAccessKey` 、和`SessionToken` 。	遷移工程師，AWS 管理員
對於 Linux 伺服器，請安裝代理程式。	複製安裝程式命令，登入來源伺服器，然後執行安裝程式。如需詳細指示，請參閱 AWS 文件。	AWS 管理員、移轉工程師
如果是 Windows 伺服器，請安裝代理程式。	將安裝程式檔案下載到每部伺服器，然後執行安裝程式指令。如需詳細指示，請參閱 AWS 文件。	AWS 管理員、移轉工程師
等待初始資料複製完成。	安裝代理程式之後，來源伺服器會顯示在「應用程式移轉服務」主控台的「來源伺服器」區段中。等待伺服器進行初始資料複寫。	AWS 管理員、移轉工程師

設定啟動設定

任務	描述	所需技能
指定伺服器詳細資訊。	在「應用程式移轉服務」主控台上，選擇「來源伺服器」區段，然後從清單中選擇伺服器名稱以存取伺服器詳細資訊。	AWS 管理員、移轉工程師
設定啟動設定。	選擇 [啟動設定] 索引標籤。您可以設定各種設定，包括一般啟動設定和 EC2 啟動範本設定。如需詳細指示，請參閱 AWS 文件。	AWS 管理員、移轉工程師

執行測試

任務	描述	所需技能
測試來源伺服器。	1. 在 [應用程式移轉服務] 主控台的 [來源伺服器] 區段中，確定來源伺服器的移轉生命週期已準備就緒可供測試，且資料複寫狀態為 [狀況良好]。 2. 選取每個來源伺服器左側的核取方塊。 3. 選擇 [測試和切換]，然後選擇 [啟動測試執行個體]。 4. 出現提示時，選擇啟動。 伺服器將會啟動。	AWS 管理員、移轉工程師

任務	描述	所需技能
確認測試已成功完成。	測試伺服器完全啟動之後，頁面上的 [警示] 狀態會顯示每個伺服器的 [已啟動]。	AWS 管理員、移轉工程師
測試伺服器。	針對測試伺服器執行測試，以確保其正常運作。	AWS 管理員、移轉工程師

排程並執行切換

任務	描述	所需技能
排程切換視窗。	與相關團隊安排適當的切換時間表。	AWS 管理員、移轉工程師
執行切換。	1. 在「應用程式移轉」主控台的「來源伺服器」頁面上，選取每個來源伺服器左側的核取方塊。 2. 選擇測試和切換，然後選擇標記為「準備切換」。 3. 確認每個來源伺服器的移轉生命週期都已準備就緒，可進行切換。 4. 選擇 [測試和切換]，然後選取 [啟動切換實例]。 5. 出現提示時，選擇啟動。伺服器將會啟動。 來源伺服器的移轉生命週期將變更為正在進行中的切換。	AWS 管理員、移轉工程師
確認切換已成功完成。	切換伺服器完全啟動之後，「來源伺服器」頁面上的	AWS 管理員、移轉工程師

任務	描述	所需技能
	「警示」狀態會顯示每個伺服器的「已啟動」。	
測試伺服器。	針對切換伺服器執行測試，以確保其正常運作。	AWS 管理員、移轉工程師
完成切換。	選擇 [測試和切換]，然後選取 [完成切換] 以完成移轉程序。	AWS 管理員、移轉工程師

相關資源

- AWS Application Migration Service
- AWS 應用程式遷移服務使用指南

使用 AWS SFTP 將小型資料集從現場部署遷移到 Amazon S3

R 類型：重新主機	來源：存儲	目標：Amazon S3
創建者：AWS	環境：生產	技術：儲存與備份；移轉
AWS 服務：Amazon S3		

Summary

此模式說明如何使用適用於 SFTP 的 AWS 傳輸 (AWS SFTP)，將小型資料集 (5 TB 或更少) 從現場部署資料中心遷移到 Amazon 簡單儲存服務 (Amazon S3)。 資料可以是資料庫傾印或平面檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在您的資料中心和 AWS 之間建立的 AWS Direct Connect 連結

限制

- 資料檔案必須小於 5 TB。對於超過 5 TB 的檔案，您可以執行多部分上傳到 Amazon S3，或選擇其他資料傳輸方法。

架構

源, 技術, 堆棧

- 內部部署平面檔案或資料庫傾印

目標技術堆疊

- Amazon S3

來源與目標架構

工具

- AWS SFTP — 允許使用安全檔案傳輸通訊協定 (SFTP) 將檔案直接傳入和傳出 Amazon S3。
- AWS Direct Connect — 建立從現場部署資料中心到 AWS 的專用網路連線。
- VPC 端點 — 可讓您將 VPC 以私有方式連接到由 AWS 提供支援的 AWS 服務和 VPC 端點服務，PrivateLink 而無需網際網路閘道、網路位址轉譯 (NAT) 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公用 IP 位址即可與服務中的資源通訊。

史诗

準備移轉

任務	描述	所需技能
記錄目前的 SFTP 需求。		應用程式擁有者，SA
識別驗證需求。	需求可能包括以金鑰為基礎的驗證、使用者名稱或密碼或身分識別提供者 (IdP)。	應用程式擁有者，SA
識別應用程式整合需求。		應用程式擁
識別需要服務的使用者。		應用程式擁
判斷 SFTP 伺服器端點的 DNS 名稱。		聯網
決定備份策略。		SA、DBA (如果資料已傳輸)
識別應用程式移轉或切換策略。		應用程式擁有者、SA、DBA

設定基礎結構

任務	描述	所需技能
在您的 AWS 帳戶中建立一或多個虛擬私有雲端 (VPC) 和子網路。		應用程式擁有者，AMS
建立安全性群組和網路存取控制清單 (ACL)。		安全性、網路、AMS
建立 S3 儲存貯體。		應用程式擁有者，AMS
建立身分識別和存取管理 (IAM) 角色。	建立包含許可的 IAM 政策，讓 AWS SFTP 存取您的 S3 儲存貯體。此 IAM 政策決定您提供 SFTP 使用者的存取層級。建立另一個 IAM 政策以建立與 AWS SFTP 的信任關係。	安全性、AMS
建立註冊網域的關聯 (選用)。	如果您有自己的註冊網域，則可以將其與 SFTP 伺服器建立關聯。您可以從網域或子網域將 SFTP 流量路由到 SFTP 伺服器端點。	網路、AMS
建立一個 SFTP 伺服器。	指定服務用來驗證使用者的身分識別提供者類型。	應用程式擁有者，AMS
開啟一個 SFTP 用戶端。	開啟 SFTP 用戶端並設定連線以使用 SFTP 端點主機。AWS SFTP 支援任何標準的 SFTP 用戶端。常用的 SFTP 用戶端包括 OpenSSH、WinSCP、數碼鴨和. FileZilla 您可以從 AWS SFTP 主控台取得 SFTP 伺服器主機名稱。	應用程式擁有者，AMS

計劃和測試

任務	描述	所需技能
規劃應用程式移轉。	規劃所需的任何應用程式組態變更、設定移轉日期，並決定測試排程。	應用程式擁有者，AMS
測試基礎結構。	在非生產環境中進行測試。	應用程式擁有者，AMS

相關資源

參考

- 適用 AWS Transfer for SFTP 使用者指南
- AWS Direct Connect 資源
- VPC 端點

教學課程和影片

- AWS Transfer for SFTP (影片)
- 適用 AWS Transfer for SFTP 使用者指南
- AWS SA 白板-直 Connect (影片)

從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk

R 類型：重新主機	資源：應用程序開發	目標：AWS Elastic Beanstalk
創建者：AWS	環境：PoC 或試點	技術：容器與微服務；Web 和行動應用程式；移轉
工作負載：開放原始碼；	AWS 服務：AWS Elastic Beanstalk	

Summary

此模式說明如何將在現場部署 Oracle GlassFish 伺服器上執行的 Java 應用程式遷移到 AWS 雲端中的 AWS Elastic Beanstalk。

在 AWS 上，Java 應用程式部署在具有 AWS Elastic Beanstalk 的 Docker GlassFish 伺服器上，該伺服器會在亞馬遜彈性運算雲端 (Amazon EC2) Auto Scaling 群組中執行。

附加功能：

- Amazon Elastic Beanstalk 充當幾個基礎資源的包裝。它會設定 Elastic Load Balancing (處理來自 Amazon Route 53 的傳入流量)、將流量分散到一個或多個 EC2 執行個體，也可做為部署工具使用。
- 若要將現場部署資料庫遷移到 Amazon Relational Database Service (Amazon RDS)，請更新資料庫連線詳細資訊。在後端資料庫中，您可以設定 Amazon RDS 異地同步備份部署並選擇資料庫引擎類型。
- 您可以使用異地同步備份部署來實現高可用性，以及 Auto Scaling 群組和擴展政策，以提高恢復能力。
- 您可以根據 Amazon CloudWatch 指標設定擴展政策。
- 在 AWS Elastic Beanstalk 中，您可以設定基礎 Elastic Load Balancing 設定和 Amazon EC2 Auto Scaling。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 執行於的內部部署 Java 應用程式 GlassFish
- 一個 Java 網絡應用程序資源（WAR）文件

產品版本

- 甲骨文玻璃魚 4.1.2 和 5.0
- 爪哇 GlassFish
- 爪哇 8 GlassFish 4.1 或更新版本

架構

源, 技術, 堆棧

- 應用程式開發於 GlassFish

目標技術堆疊

- Elastic Beanstalk

目標架構

部署工作流

工具

- Amazon Elastic Beanstalk — 用於在包括阿帕奇，NGINX，乘客和 IIS 在服務器上部署和擴展使用 Java，淨，PHP，Node.js，Python，紅寶石，圍棋和碼頭開發的 Web 應用程序和服務的服務的服務。
- Amazon CloudWatch — 提供資料和可操作的見解，以監控應用程式、回應整個系統的效能變化、最佳化資源使用率，並提供營運狀態的統一檢視。
- Docker — 將軟體封裝成標準化單位的平台，以便快速建置、測試及部署應用程式。
- Java-一種通用的編程語言。Java 是基於類的，面向對象的，旨在具有較少的實現依賴關係。

史诗

設定 VPC

任務	描述	所需技能
使用必要資訊建立虛擬私有雲 (VPC) 執行個體。		SysAdmin
在 VPC 內建立至少兩個子網路。		SysAdmin
根據需求創建一個路由表。		SysAdmin

設置 Amazon S3

任務	描述	所需技能
建立 Amazon Simple Storage Service (Amazon S3) 儲存貯體。		SysAdmin
將 WAR 檔案複製到 S3 儲存貯體，然後上傳應用程式程式碼。		SysAdmin

建立 IAM 角色

任務	描述	所需技能
建立 AWS Identity and Access Management (IAM) 角色。	您可以使用默認的「aws-elasticbeanstalk-ec2 角色」配置文件，或讓 Elastic Beanstalk 自動創建它。	SysAdmin

設置 Elastic Beanstalk

任務	描述	所需技能
開啟 Elastic Beanstalk 儀表板。		SysAdmin
創建一個新的應用程序並選擇 Web 服務器環境。		SysAdmin
選擇 GlassFish 泊塢視窗作為預先設定的平台。		SysAdmin
上傳代碼。	從本機系統檔案提供 S3 儲存貯體檔案 URL 或 ZIP 檔案。	SysAdmin
選擇環境類型。	在組態容量設定中，選擇單一執行個體或 Load Balancer。	SysAdmin
設定 Load Balancer。	如果您在上一個步驟中選擇了 Load Balancer，請設定異地同步備份部署。	SysAdmin
在「組態安全性」設定中，選擇先前建立的 IAM 角色。		SysAdmin
在組態安全設定中，如果您有現有的 key pair，請使用它或建立新的 Amazon EC2 key pair。		SysAdmin
在組態監控設定中，設定 Amazon CloudWatch。		SysAdmin
在 [組態安全性] 設定中，選擇先前建立的 VPC。		SysAdmin
選擇「建立環境」。		SysAdmin

測試應用程式。

任務	描述	所需技能
使用建立環境中提供的 URL 測試應用程式。		
在 Amazon Route 53 中應用域名服務（DNS）更改。		

相關資源

- 甲骨 GlassFish 文文件
- GlassFish 開放原始碼 Java EE 參考實作
- AWS Elastic Beanstalk 文件
- 使用 Elastic Beanstalk 與 Amazon CloudWatch
- AWS Elastic Beanstalk 定價
- EC2 Auto Scaling 組
- 縮放 Auto Scaling 組的大小
- Amazon RDS 異地備份部署

將現場部署 Oracle 資料庫遷移到 Amazon EC2 上的甲骨文

由白芝夏克 (AWS) 和潘卡·舒達里 (AWS) 創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon EC2 上的甲骨文
R 類型：重新主機	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon EC2		

Summary

此模式會引導您完成在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上將現場部署 Oracle 資料庫遷移到 Oracle 的步驟。它描述了兩個遷移選項：使用 AWS 資料遷移服務 (AWS DMS) 或使用原生 Oracle 工具，例如 RMAN、資料泵匯入/匯出、可傳輸的表格空間和 Oracle。 GoldenGate

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Oracle 資料庫

限制

- 目標作業系統 (OS) 必須受到 Amazon EC2 的支援。如需支援系統的完整清單，請參閱 Amazon EC2 常見問答集。

產品版本

- 適用於 Enterprise、Standard、Standard One 和 Standard Two 等版本的 Oracle 10.2 版和更新版本 (適用於 10.x 版)、11g 版且最高可達 12.2 版，以及 18c 版。如需 AWS DMS 支援的最新版本清單，請參閱 AWS DMS 文件中的「資料遷移來源」中的「現場部署和 Amazon EC2 執行個體資料庫」。

架構

源, 技術, 堆棧

- 內部部署 Oracle 資料庫

目標技術堆疊

- Amazon EC2 上的甲骨文數據庫實例

目標架構

資料移轉架構

使用 AWS DMS：

使用原生的甲骨文工具：

工具

- AWS DMS- AWS Database Migration 服務 (AWS DMS) 支援多種類型的來源和目標資料庫。如需支援的資料庫版本和版本的相關資訊，請參閱使用 Oracle 資料庫做為 AWS DMS 的來源。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。
- 原生 Oracle 工具- RMAN、資料汲取匯入/匯出、可傳輸表格空間、Oracle GoldenGate

史诗

規劃移轉

任務	描述	所需技能
驗證來源資料庫和目標資料庫的版本。		DBA
識別目標作業系統的版本。		DBA, SysAdmin
根據 Oracle 相容性清單和容量需求，識別目標伺服器執行處理的硬體需求。		DBA, SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
識別網路需求 (延遲和頻寬)。		DBA, SysAdmin
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路/主機存取安全性需求。		DBA, SysAdmin
識別安裝 Oracle 軟體所需的作業系統使用者清單。		DBA, SysAdmin
下載 AWS Schema Conversion Tool (AWS SCT) 和驅動程式。		DBA
為工作負載建立 AWS SCT 專案，並連線到來源資料庫。		DBA
生成用於創建對象（表，索引，序列等）的 SQL 文件。		DBA

任務	描述	所需技能
決定備份策略。		DBA, SysAdmin
決定可用性需求。		DBA
識別應用程式遷移/切換策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結構

任務	描述	所需技能
在您的 AWS 帳戶中建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 EC2 執行個體。		SysAdmin

安裝甲骨文軟件

任務	描述	所需技能
建立 Oracle 軟體所需的作業系統使用者和群組。		DBA, SysAdmin
下載所需的 Oracle 軟體版本。		
在 EC2 執行個體上安裝甲骨文軟體。		DBA, SysAdmin
使用 AWS SCT 產生的指令碼，建立資料表、主索引鍵、檢視和序列等物件。		DBA

移轉資料-選項 1

任務	描述	所需技能
使用原生 Oracle 工具或協力廠商工具來移轉資料庫物件和資料。	Oracle 工具包括「資料汲取」匯入/匯出、RMAN、可傳輸表格空間和。 GoldenGate	DBA

移轉資料-選項 2

任務	描述	所需技能
決定移轉方法。		DBA
在 AWS DMS 主控台中建立複寫執行個體。		DBA
建立來源端點和目標端點。		DBA
建立複製工作。		DBA
啟用變更資料擷取 (CDC) 以擷取連續複寫的變更。		DBA
執行複寫工作並監視記錄。		DBA
當滿載完成時，創建輔助對象，如索引和外鍵。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切過

任務	描述	所需技能
遵循應用程式切換/切換策略。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時的 AWS Secrets Manager 資源。		DBA, SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集移轉時間的指標、手動與工具的百分比、節省成本等。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供意見反應。		

相關資源

參考

- 將甲骨文資料庫遷移到 AWS 的策略
- 將甲骨文資料庫遷移到 AWS 雲端
- Amazon EC2 網站
- AWS 管理系統網站
- AWS DMS 部落格文章
- Amazon EC2 定價
- 在雲端運算環境中授權 Oracle 軟體

教學課程和影片

- Amazon EC2 入門
- 開始使用 AWS DMS
- Amazon EC2 簡介-彈性雲端伺服器和 AWS 託管 (影片)

使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到 Amazon EC2

由納瓦坎塔盧裡（AWS）創建

環境：PoC 或試點	來源：內部部署 Oracle 資料庫	目標：Amazon EC2 上的甲骨文數據庫
R 類型：重新主機	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon EC2；AWS Direct Connect		

Summary

移轉資料庫時，您必須考慮諸如來源和目標資料庫引擎和版本、移轉工具和服務，以及可接受的停機期間等因素。如果您要將現場部署 Oracle 資料庫遷移到亞馬遜彈性運算雲端 (Amazon EC2)，您可以使用 Oracle 工具，例如 Oracle 資料泵和 Oracle 復原管理器 (RMAN)。如需策略的詳細資訊，請參閱將 Oracle 資料庫遷移到 AWS 雲端。

Oracle 資料泵浦可協助您擷取資料庫的邏輯一致備份，並將其還原到目標 EC2 執行個體。此模式說明如何使用 Oracle 資料泵浦和NETWORK_LINK參數，將現場部署 Oracle 資料庫遷移至 EC2 執行個體，將停機時間降至最低。NETWORK_LINK參數透過資料庫連結啟動匯入。目標 EC2 執行個體上的 Oracle 資料汲取匯入 (impdp) 用戶端會連線到來源資料庫、從中擷取資料，然後將資料直接寫入目標執行個體上的資料庫。此解決方案中沒有使用任何備份或傾印檔案。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 一個內部部署 Oracle 資料庫，可：
  - 不是甲骨文實際應用程式叢集 (RAC) 資料庫
  - 不是 Oracle 自動儲存體管理 (Oracle ASM) 資料庫
  - 處於讀寫模式。
- 您已在現場部署資料中心和 AWS 之間建立 AWS Direct Connect 連結。如需詳細資訊，請參閱建立連線 (直 Connect 線說明文件)。

產品版本

- Oracle 資料庫 10g 發行版本 1 (10.1) 及更新版本

架構

源, 技術, 堆棧

- 內部部署資料中心中的獨立式 (非 RAC 和非 ASM) Oracle 資料庫伺服器

目標技術堆疊

- 在 Amazon EC2 上運行的甲骨文數據庫

目標架構

AWS 架構良好的架構的可靠性支柱建議建立資料備份，以協助提供高可用性和彈性。如需詳細資訊，請參閱在 AWS 上執行 Oracle 資料庫的最佳實務中建構高可用性。此模式使用 Oracle 主動資料保全在 EC2 執行個體上設定主資料庫和待命資料庫。為了獲得高可用性，EC2 執行個體應位於不同的可用區域。不過，可用區域可以位於相同的 AWS 區域或不同的 AWS 區域。

作用中資料保全提供實體待命資料庫的唯讀存取權，並從主要資料庫持續套用重做變更。根據復原點目標 (RPO) 和復原時間目標 (RTO)，您可以選擇同步和非同步重做傳輸選項。

下圖顯示主要和待命 EC2 執行個體位於不同 AWS 區域時的目標架構。

資料移轉架構

完成設定目標架構之後，您可以使用 Oracle 資料泵將現場部署資料和結構描述遷移到主要 EC2 執行個體。在切換期間，應用程式無法存取內部部署資料庫或目標資料庫。您必須關閉這些應用程式，直到它們可以連線到主要 EC2 執行個體上的新目標資料庫為止。

下圖顯示了資料移轉期間的架構。在此範例架構中，主要和待命 EC2 執行個體位於不同的 AWS 區域。

工具

AWS 服務

- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

其他工具和服務

- Oracle 作用中資料保全可協助您建立、維護、管理及監督待命資料庫。
- Oracle 資料泵取可協助您以高速將資料和中繼資料從一個資料庫移至另一個資料庫。

最佳實務

- 在 AWS 上執行 Oracle Database 的最佳實務
- 使用網路連結匯入資料

史诗

在 AWS 上設定 EC2 執行個體

任務	描述	所需技能
識別內部部署主機的來源硬體組態和核心參數。	驗證內部部署組態，包括儲存區大小、每秒輸入/輸出作業 (IOPS) 和 CPU。這對於基於 CPU 核心的 Oracle 授權而言非常重要。	DBA SysAdmin
在 AWS 上建立基礎設施。	建立虛擬私人雲端 (VPC)、私有子網路、安全群組、網路存取控制清單 (ACL)、路由表和網際網路閘道。如需詳細資訊，請參閱下列內容：	資料庫管理員、AWS 系統管理員

任務	描述	所需技能
	• 虛擬私人雲端和子網路 • 教學課程：建立 VPC 以搭配資料庫執行個體使用	
使用作用中資料保全設定 EC2 執行個體。	使用主動資料保全組態設定來設定 AWS EC2 執行個體，如 AWS Well-Architected Framework 中所述。EC2 執行個體上的 Oracle 資料庫版本可能與現場部署版本不同，因為此模式使用邏輯備份。注意下列事項： • 將目標資料庫置於讀寫模式。 • 在目標資料庫上，提供來源資料庫的透明網路基板 (TNS) 詳細資料。 如需詳細資訊，請參閱： • 啟動資料庫 (Oracle 文件集) • 建立與設定 Oracle 資料庫 (Oracle 說明文件)	資料庫管理員、AWS 系統管理員

將資料庫遷移到 Amazon EC2

任務	描述	所需技能
從 EC2 執行個體建立內部部署資料庫的資料庫連結。	在 EC2 執行個體上的 Oracle 資料庫與現場部署 Oracle 資料庫之間建立資料庫連結 (dblink)。如需詳細資訊，請參	DBA

任務	描述	所需技能
	閱使用網路連結匯入來移動資料 (Oracle 說明文件)。	
驗證 EC2 執行個體和現場部署主機之間的連線。	使用 dblink 確認 EC2 執行個體與現場部署資料庫之間的連線正常運作。如需指示，請參閱建立資料庫連結 (Oracle 說明文件)。	DBA
停止連線到內部部署資料庫的所有應用程式	核准資料庫停機時間之後，請關閉任何與內部部署資料庫連結的應用程式和相依工作。您可以直接從應用程序執行此操作，也可以使用 cron 從數據庫中執行此操作。如需詳細資訊，請參閱使用 Crontab 公用程式在 Oracle Linux 上排程作業。	DBA, 應用程式開發人員
排程資料移轉工作。	在目標主機上，使用指令`impdb`來排程「資料汲取」匯入。這會將目標資料庫連線到內部部署主機，並開始資料移轉。如需詳細資訊，請參閱資料汲取匯入和網路 _LINK (Oracle 文件集)。	DBA
驗證資料移轉。	數據驗證是一個關鍵步驟。對於數據驗證，您可以使用自定義工具或 Oracle 工具，例如數據庫鏈接和 SQL 查詢的組合。	DBA

切過

任務	描述	所需技能
將來源資料庫置於唯讀模式。	確認應用程式已關閉，且未對來源資料庫進行任何變更。以唯讀模式開啟來源資料庫。這有助於您避免任何未結交易。如需詳細資訊，請參閱 SQL 敘述句ALTER DATABASE中的 (Oracle 說明文件)。	DBA、 DevOps 工程師、應用程式開發人員
驗證對象計數和數據。	若要驗證資料和物件，請使用自訂工具或 Oracle 工具，例如資料庫連結和 SQL 查詢的組合。	DBA, 應用程式開發人員
將應用程式 Connect 到主要 EC2 執行個體上的資料庫。	將應用程式的連線屬性變更為指向您在主要 EC2 執行個體上建立的新資料庫。	DBA, 應用程式開發人員
驗證應用程式效能。	啟動應用程式。使用「自動工作負載儲存區域」來驗證應用程式的功能和效能 (Oracle 說明文件)。	應用程式開發人員、 DevOps 工程師、DBA

相關資源

AWS 參考資料

- 將甲骨文資料庫遷移到 AWS 雲端
- Amazon EC2 甲骨文
- 將龐大的 Oracle 資料庫遷移到 AWS 以適用於跨平台
- 虛擬私人雲端和子網路
- 教學課程：建立 VPC 以搭配資料庫執行個體使用

甲骨文參考

- Oracle 資料保全組態
- 資料汲取匯入

將現場部署 SAP ASE 資料庫遷移至 Amazon EC2

R 類型：重新主機	來源：數據庫：關係	目標：Amazon EC2 上的 SAP 調適性伺服器企業
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作負載：SAP	AWS 服務：Amazon EC2	

Summary

此模式說明如何將 SAP 自適應伺服器企業 (ASE) 資料庫從現場部署主機遷移到 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。該模式涵蓋了 AWS Database Migration Service (AWS DMS) 或 SAP ASE 原生工具的使用，例如日月光駕駛艙、適用於日月光的 Sybase 中央，以及用於移轉的 DBA 駕駛艙。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的 SAP ASE 來源資料庫

限制

- 來源資料庫必須小於 64 TB

產品版本

- SAP 日月光第 15.x 版及 16.x 版或更新版本

架構

源, 技術, 堆棧

- 內部部署 SAP ASE 資料庫

目標技術堆疊

- EC2 執行個體上的 SAP ASE 資料庫

資料庫遷移架構

使用 AWS DMS：

使用原生 SAP ASE 工具：

工具

- AWS DMS-AWS 資料遷移服務 (AWS DMS) 支援多種不同的來源和目標資料庫。如需詳細資訊，請參閱資料移轉來源和資料移轉的目標。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。
- SAP ASE-原生工具包括日月光駕駛艙、適用於日月光的 Sybase 中央和 DBA 駕駛艙。

史诗

分析移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA
識別目標作業系統版本。		DBA, SysAdmin
根據 SAP ASE 相容性清單和容量需求，識別目標伺服器執行個體的硬體需求。		DBA, SysAdmin
識別儲存類型和容量的需求。		DBA, SysAdmin
識別網路需求，包括延遲和頻寬。		DBA, SysAdmin

任務	描述	所需技能
選擇適當的執行個體類型、容量、儲存功能和網路功能。		DBA, SysAdmin
識別來源和目標資料庫的網路和主機存取安全性需求。		DBA, SysAdmin
識別 SAP ASE 軟體安裝所需的作業系統使用者清單。		DBA, SysAdmin
決定備份策略。		DBA
決定可用性需求。		DBA
識別應用程式遷移和轉換策略。		DBA,, 應用程式 SysAdmin擁有者

設定基礎結構

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 EC2 執行個體。		SysAdmin

安裝軟體

任務	描述	所需技能
建立 SAP ASE 軟體運作所需的作業系統使用者和群組。		DBA, SysAdmin

任務	描述	所需技能
下載所需版本的 SAP ASE 軟體。		DBA, SysAdmin
在 EC2 執行個體上安裝 SAP ASE 資料庫、備份伺服器軟體和複寫伺服器軟體，然後設定伺服器。		DBA, SysAdmin

移轉資料-選項 1

任務	描述	所需技能
使用原生 SAP ASE 工具或協力廠商工具移轉資料庫物件和資料。	請參閱 SAP ASE 或協力廠商工具的說明文件。其中包括日月光駕駛艙、適用於日月光的 Sybase 中央和 DBA 駕駛艙。	DBA

遷移數據-選項 2

任務	描述	所需技能
使用 AWS DMS 遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切過

任務	描述	所需技能
遵循應用程式切換或轉換策略。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, SysAdmin
驗證和審核專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集移轉時間的指標、手動與工具成本節約的百分比等。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供任何意見反應。		DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- Amazon EC2
- AWS DMS
- Amazon EC2 定價

教學課程和影片

- Amazon EC2 入門
- 開始使用 AWS Database Migration Service
- AWS 資料遷移服務 (影片)

- Amazon EC2 簡介-彈性雲端伺服器和 AWS 託管 (影片)

將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon EC2

R 類型：重新主機	來源：數據庫：關係	目標：Amazon EC2 上的 Microsoft SQL 服務器
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作負載：Microsoft	AWS 服務：Amazon EC2	

Summary

此模式說明如何將現場部署 Microsoft SQL Server 資料庫遷移到 Amazon 彈性運算雲端 (亞馬遜 EC2) 執行個體上的 Microsoft SQL Server。它涵蓋兩個遷移選項：使用 AWS 資料遷移服務 (AWS DMS) 或使用原生 Microsoft SQL Server 工具，例如備份和還原、複製資料庫精靈，或複製和附加資料庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon EC2 支援的作業系統 (如需受支援作業系統版本的完整清單，請參閱 Amazon EC2 常見問答集)
- 內部部署資料中心中的 Microsoft SQL 伺服器來源資料庫

產品版本

- Microsoft SQL 伺服器版本適用於企業版、標準、工作群組和開發人員版本 (如果您使用的是 AWS DMS)。2008R2 若要遷移 Microsoft SQL 伺服器網頁版或快速版，請使用原生或協力廠商工具。如需支援版本的最新清單，請參閱使用 Microsoft SQL 伺服器資料庫做為 AWS DMS 的目標。

架構

源, 技術, 堆棧

- 內部部署 Microsoft SQL Server 資料庫

目標技術堆疊

- EC2 實例上的 Microsoft SQL 服務器數據庫

目標架構

資料移轉架構

- 使用 AWS DMS

- 使用原生 SQL 伺服器工具

工具

- AWS DMS-AWS 資料遷移服務 (AWS DMS) 可協助您在廣泛使用的商業和開放原始碼資料庫 (包括甲骨文、SQL 伺服器、MySQL 和 PostgreSQL) 之間移轉資料，或從廣泛使用的商業和開放原始碼資料庫遷移 您可以使用 AWS DMS 將資料遷移至 AWS 雲端，可在現場部署執行個體 (透過 AWS 雲端設定) 或在雲端和現場部署設定之間進行。
- 原生 Microsoft SQL Server 工具-這些工具包括備份和恢復，複製數據庫嚮導，以及複製和附加數據庫。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA
識別目標作業系統版本。		DBA, SysAdmin
根據 Microsoft SQL Server 相容性清單和容量需求，識別目		DBA, SysAdmin

任務	描述	所需技能
標伺服器執行個體的硬體需求。		
識別類型和容量的儲存需求。		DBA, SysAdmin
識別網路需求，包括延遲和頻寬。		DBA, SysAdmin
根據容量、儲存功能和網路功能選擇 EC2 執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路和主機存取安全性需求。		DBA, SysAdmin
識別安裝 Microsoft SQL Server 軟體所需的使用者清單。		DBA, SysAdmin
決定備份策略。		DBA
決定可用性需求。		DBA
識別應用程式移轉和切換策略。		DBA, SysAdmin

設定基礎結構

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 EC2 執行個體。		SysAdmin

安裝軟體

任務	描述	所需技能
建立 Microsoft SQL 伺服器軟體所需的使用者和群組。		DBA, SysAdmin
下載 Microsoft SQL 服務器軟件。		DBA, SysAdmin
在 EC2 執行個體上安裝 Microsoft SQL 伺服器軟體並設定伺服器。		DBA, SysAdmin

移轉資料-選項 1

任務	描述	所需技能
使用原生 Microsoft SQL Server 工具或協力廠商工具來移轉資料庫物件和資料。	工具包括備份和還原、複製資料庫精靈，以及複製和附加資料庫。	DBA

遷移數據-選項 2

任務	描述	所需技能
使用 AWS DMS 遷移資料。	如需使用 AWS DMS 的詳細資訊，請參閱參考和說明一節中的連結。	DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。	使用 AWS Schema Conversion Tool (AWS SCT) 來分析和修	DBA，應用程式擁有者

任務	描述	所需技能
	改內嵌在應用程式原始程式碼中的 SQL 程式碼。	

切過

任務	描述	所需技能
遵循應用程式切換策略。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉所有臨時 AWS 資源。	臨時資源包括 AWS DMS 複寫執行個體和適用於 AWS SCT 的 EC2 執行個體。	DBA, SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集移轉時間的指標、手動與工具成本節約的百分比等。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供意見反應。		DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- 在 Amazon Web Services 上部署 Microsoft SQL 服務器
- Amazon EC2
- Amazon EC2 常見問

- AWS Database Migration Service
- Amazon EC2 定價
- AWS 上的 Microsoft 產品
- AWS 上的 Microsoft 授權
- AWS 上的 Microsoft SQL 服務器

教學課程和影片

- Amazon EC2 入門
- 開始使用 AWS Database Migration Service
- 將 Amazon EC2 實例添加到您的目錄（Simple AD 和 Microsoft AD）
- AWS Database Migration Service (影片)
- Amazon EC2 簡介-彈性雲端伺服器和 AWS 託管 (影片)

將現場部署 MySQL 資料庫遷移到 Amazon EC2

R 類型：重新主機	來源：數據庫：關係	目標：Amazon EC2 上的 MySQL
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作負載：開源		

Summary

此模式提供將現場部署 MySQL 資料庫遷移至 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上的 MySQL 資料庫的指導。該模式討論如何使用 AWS Database Migration Service (AWS DMS) 或原生 MySQL 工具，例如用於遷移的 mysqldbcopy 和 mysqldump。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的 MySQL 來源資料庫

產品版本

- MySQL 5.5、5.6 和 5.7 版
- 如需 Amazon EC2 支援的目標作業系統清單，請參閱 Amazon EC2 常見問答集

架構

源, 技術, 堆棧

- 內部部署 MySQL 資料庫

目標技術堆疊

- Amazon EC2 上的 MySQL 數據庫實例

AWS 資料遷移方法

- AWS DMS
- 本地 MySQL 工具（神秘數據庫複製，神秘轉儲）

目標架構

AWS 資料遷移架構

使用 AWS DMS：

使用本地 MySQL 工具：

工具

- AWS DMS-AWS Database Migration Service (AWS DMS) 支援多個來源和目標資料庫。如需 AWS DMS 支援的 MySQL 來源和目標資料庫的詳細資訊，請參閱將與 MySQL 相容的資料庫遷移到 AWS。如果 AWS DMS 不支援您的來源資料庫，則必須選擇其他方法來遷移資料。
- 本地 MySQL 工具 -神秘的數據庫複製和神秘的轉儲

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。		DBA
識別目標作業系統版本。		DBA, SysAdmin
根據 MySQL 相容性清單和容量需求，識別目標伺服器執行個體的硬體需求。		DBA, SysAdmin

任務	描述	所需技能
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
識別網路需求，例如延遲和頻寬。		DBA, SysAdmin
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路或主機存取安全性需求。		DBA, SysAdmin
識別安裝 MySQL 軟體所需的作業系統使用者清單。		DBA, SysAdmin
決定備份策略。		DBA
決定可用性需求。		DBA
識別應用程式移轉或轉換策略。		DBA, SysAdmin

設定基礎結構

任務	描述	所需技能
建立虛擬私有雲 (VPC) 和子網路。		SysAdmin
建立安全性群組和網路存取控制清單 (ACL)。		SysAdmin
設定並啟動 EC2 執行個體。		SysAdmin

安裝 MySQL 件

任務	描述	所需技能
建立 MySQL 軟體運作所需的作業系統使用者和群組。		DBA, SysAdmin
下載所需版本的 MySQL 軟體。		DBA, SysAdmin
在 EC2 執行個體上安裝 MySQL 軟體並設定伺服器。		DBA, SysAdmin

移轉資料-選項 1

任務	描述	所需技能
使用原生 MySQL 工具或協力廠商工具來移轉資料庫物件和資料。	這些工具包括 mysqldbcopy 和 mysqldump。	DBA

移轉資料-選項 2

任務	描述	所需技能
使用 AWS DMS 遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA,, 應用程式 SysAdmin擁有者

切過

任務	描述	所需技能
遵循應用程式切換或轉換策略。		DBA,, 應用程式 SysAdmin擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。	關閉 AWS DMS 複寫執行個體。	DBA, SysAdmin
審核並驗證專案文件。		DBA,, 應用程式 SysAdmin擁有者
收集移轉時間的指標、手動與工具的百分比、節省成本等。		DBA,, 應用程式 SysAdmin擁有者
關閉專案並提供意見反應。		DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- Amazon EC2 網站
- 數據管理系統網站
- Amazon EC2 定價
- AWS DMS 逐步解說逐步解說

教學課程和影片

- 開始使用 AWS DMS
- Amazon EC2 簡介-彈性雲端伺服器和 AWS 託管 (影片)

使用應用程式遷移服務減少同質 SAP 移轉切換時間

由帕維爾·魯賓（AWS），迭戈瓦爾韋德（AWS）和蘇尼爾·亞達夫（AWS）創建

環境：生產	來源：內部部署 SAP ASE 資料庫	目標：Amazon EC2 上的 SAP 資料庫
R 類型：重新主機	工作負載：SAP	技術：移轉；資料庫
AWS 服務：AWS 應用程式遷移服務；Amazon EBS		

Summary

此模式概述了使用 AWS 應用程式遷移服務遷移 SAP 工作負載的步驟。應用程式移轉服務透過使用區塊層級複製來維護從其來源持續同步的複製磁碟區，以便進行切換。

SAP 工作負載包括應用程式 SAP 客戶關係管理 (SAP CRM)、SAP 企業資源規劃 (ERP) 和 SAP 企業倉儲 (SAP BW)。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶，在來源 SAP 伺服器與 AWS 上的目的地虛擬私有雲端 (VPC) 之間具有穩定的網路連線
- 適用於內部部署資料中心的 Linux 或 Windows 的 SAP 調適性伺服器企業 (ASE) 來源資料庫

限制

- 目標作業系統必須受到亞馬遜彈性運算雲 (Amazon EC2) 的支援。如需詳細資訊，請參閱 Amazon EC2 常見問答集。

架構

源, 技術, 堆棧

- 一個 SAP 日月光資料庫

目標技術堆疊

- Amazon EC2
- Amazon Elastic Block Store (Amazon EBS)

來源與目標架構

下圖顯示透過複寫代理程式從內部部署伺服器移轉至應用程式移轉服務端點。Amazon Simple Storage Service (Amazon S3) 端點可用來存取安裝和組態檔案。暫存區域和遷移資源的子網路包含 EC2 執行個體，在 EBS 磁碟區上具有資料儲存。連接埠 TCP 443 用於將來源機器網路連接到應用程式遷移服務，以及將暫存區子網路連接到應用程式遷移服務、Amazon EC2 和 Amazon S3 區域端點。連接埠 TCP 1500 用於區域網路和暫存區域之間的資料複寫。

工具

- AWS 應用程式遷移服務可協助您將 (lift-and-shift) 應用程式重新託管到 AWS 雲端，而且不需要變更，停機時間也最小。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Security Token Service (AWS STS) 可協助您為使用者申請臨時、有限權限的登入資料。

史诗

初始化應用程式遷移

任務	描述	所需技能
初始化應用程式移轉服務	在您要部署 SAP ASE 資料庫的 AWS 區域中初始化應用程	AWS 管理員

任務	描述	所需技能
	式遷移服務。當您第一次瀏覽至每個區域的「應用程式遷移服務」頁面時，AWS 會提供自動化設定。	
手動建立服務角色。	(選擇性) 如果您想要使用自動化 (例如 AWS Control Tower) 來設定帳戶，可以手動建立安裝、複寫和啟動所需的六個 AWS Identity and Access Management (IAM) 角色。如需指示，請參閱 AWS 文件。	AWS 管理員
建立複製設定範本。	複寫設定範本定義子網路、執行個體類型、Amazon EBS 加密以及資料路由的方式。如需詳細設定資訊，請參閱 AWS 文件。	一般 AWS

產生代理程式安裝的認證

任務	描述	所需技能
建立新的 IAM 角色。	在 IAM 主控台上，導覽至 [角色]，然後選擇 [建立角色]。 對於受信任的實體類型，請選擇 AWS 帳戶，然後選擇下一步。	AWS 系統管理員
附加 AWSApplicationMigrationAgentPolicy 至 IAM 角色。	AWS 受管`AWSApplicationMigrationAgentPolicy` 政策包含執行應用程式遷移服務代理程式安裝的必要許可。	AWS 系統管理員

任務	描述	所需技能
	附加原則之後，請選擇 [下一步]。	
完成角色建立。	指定好記名稱，然後選擇 [建立角色]。	AWS 系統管理員
產生臨時認證。	若要產生存取金鑰 ID、秘密存取金鑰和工作階段權杖，請遵循 AWS STS 文件中的指示進行。這些認證會在代理程式安裝期間使用。	AWS 系統管理員

在 SAP 來源機器上安裝應用程式移轉服務代理程式

任務	描述	所需技能
在 SAP 來源機器上下載代理程式安裝程式。	下載適用於您來源作業系統的代理程式安裝程式：Windows 或 Linux。	應用所有者
安裝 AWS 複寫代理程式。	當您在來源機器上執行代理程式安裝程式檔案時，系統會先要求您輸入存取金鑰、秘密存取金鑰、工作階段 Token 以及要複寫目標的地區。使用先前建立的 IAM 角色中的臨時登入資料，以及初始化期間設定的相同區域。	應用所有者
等待初始資料複製。	安裝代理程式之後，來源機器會顯示在應用程式移轉服務主控台的電腦索引標籤上。	應用所有者

設定目標電腦的啟動範本

任務	描述	所需技能
更新來源伺服器的 Launch 範本。	每個來源伺服器都使用唯一的 EC2 Launch 範本，通知目標 EC2 伺服器的組態。如果您想要自訂遷移伺服器的 Amazon EC2 組態，可以編輯此範本。	一般 AWS
設定預設的啟動範本版本。	在您對 Launch 範本進行必要的變更之後，請指定使用此更新版本做為預設 Launch 範本。如需詳細資訊，請參閱 AWS 文件。	一般 AWS
關閉 [執行個體類型正確大小]。	(選擇性) 執行個體類型大小適當大小會根據來源 SAP 伺服器的組態提供自動執行個體類型建議。建議您關閉此設定，以便在 Launch 範本中指定自訂執行個體類型。	一般 AWS

執行測試

任務	描述	所需技能
啟動測試啟動。	在 [應用程式移轉服務] 主控台上，選取一或多個伺服器，然後選取 [測試和切換] 下的 [啟動測試執行個體]。	一般 AWS、移轉工程師、移轉主管
等待轉換和啟動過程完成。	您可以在 [啟動歷史記錄] 索引標籤上檢閱啟動程序。機器成功啟動為 EC2 執行個體後，[警示] 索引標籤將更新為 [已啟動]。	

任務	描述	所需技能
驗證測試是否已成功完成。	透過遠端桌面通訊協定 (RDP) 或 SSH (安全殼層) Connect 線至已啟動的執行個體，並執行適當的應用程式檢查。例如，登入 SAP 介面並驗證功能。	遷移工程師，應用程序所
更新來源生命週期。	如果測試成功，請在測試和切換索引標籤上將來源機器生命週期更新為「準備切換」 標記為「準備切換」。	移民工程師，移民負責人

排程並執行對 Amazon EC2 目標的切換

任務	描述	所需技能
排程切換視窗。		切換領導者，遷移領導者，應用程序所有者
啟動切換啟動。	選取一或多個伺服器。在「測試和切換」索引標籤上，選取「應用程式移轉服務」主控台的「測試和切換」下的「啟動切換執行個體」。	移民工程師
等待轉換和啟動過程完成。	您可以在 [啟動歷史記錄] 索引標籤上檢閱啟動程序。機器成功啟動為 EC2 執行個體後，[警示] 索引標籤將更新為 [已啟動]。	
確認切換已成功完成。	透過 RDP 或 SSH Connect 至已啟動的執行個體，並執行適當的應用程式檢查。	應用所有者，遷移工程師

任務	描述	所需技能
更新來源生命週期。	如果切換成功，請透過選取「測試與切換」頁籤上的「完成切換」來更新來源電腦生命週期。	移民工程師

相關資源

參考

- AWS Application Migration Service
- AWS 應用程式遷移常見

影片

- AWS 應用程式遷移服務架構

在 AWS 雲端重新託管現場部署工作負載：移轉檢查清單

創建者：斯里坎斯朗格瓦哈拉 (AWS)

環境：PoC 或試點	來源：本地工作負載	目標：AWS 雲端
R 類型：重新主機	工作量：Microsoft	技術：移轉、混合雲、作業系統
AWS 服務：AWS 應用程式遷移服務；Amazon EC2；Amazon Connect		

Summary

在 Amazon Web Services (AWS) 雲端中重新託管現場部署工作負載涉及下列遷移階段：規劃、預先探索、探索、建置、測試和切換。此模式概述了階段及其相關任務。這些工作會以較高層級的方式進行描述，並支援約 75% 的所有應用程式工作負載。您可以在敏捷的衝刺週期中，在兩到三週內實施這些任務。

您應該與遷移團隊和顧問一起檢閱和審核這些任務。審核之後，您可以收集輸入資料、根據需求消除或重新評估工作，以及修改其他工作以支援產品組合中至少 75% 的應用程式工作負載。然後，您可以使用敏捷的項目管理工具，如 Atlassian Jira 或 Rally 軟件來導入任務，將任務分配給資源，並跟踪您的遷移活動。

該模式假設您正在使用 AWS 雲端遷移工廠重新託管工作負載，但您可以使用選擇的遷移工具。

Macie 可協助識別以資料來源形式儲存的知識庫中的敏感資料、模型叫用日誌，以及 S3 儲存貯體中的提示存放。 如需 Macie 安全性最佳做法，請參閱本指南中先前的 Macie 章節。

先決條件和限制

先決條件

- 用於跟踪遷移任務的項目管理工具（例如，阿特拉西亞 Jira 或 Rally 軟件）
- 用於在 AWS 上重新託管工作負載的遷移工具 (例如雲端移轉工廠)

架構

來源平台

- 內部部署來源堆疊 (包括技術、應用程式、資料庫和基礎結構)

目標平台

- AWS 雲端目標堆疊 (包括技術、應用程式、資料庫和基礎設施)

架構

下圖說明使用雲端遷移工廠和 AWS 應用程式遷移服務來重新託管 (探索伺服器從現場部署來源環境並將其移轉至 AWS)。

工具

- 您可以使用自己選擇的移轉和專案管理工具。

史诗

規劃階段

任務	描述	所需技能
新郎預發現積壓。	與部門主管和應用程序所有者進行預發現積壓修飾工作會議。	項目經理，敏捷的 Scrum 領導者
進行衝刺計劃工作會議。	作為範圍設定練習，請在衝刺和波之間分發您要移轉的應用程式。	項目經理，敏捷的 Scrum 領導者

發現前階段

任務	描述	所需技能
確認應用知識。	確認並記錄應用程序所有者及其對應用程序的知識。判斷技術問題是否有另一個要點的人。	移民專家（面試官）
判斷應用程式合規需求。	請與應用程式擁有者確認應用程式不必符合支付卡產業資料安全標準 (PCI DSS)、薩班斯-奧克斯利法案 (SOX)、個人識別資訊 (PII) 或其他標準的要求。如果符合性需求存在，小組必須在將要移轉的伺服器上完成符合性檢查。	移民專家（面試官）
確認生產發行需求。	向應用程式擁有者或技術連絡人確認將移轉應用程式發行至生產環境的需求 (例如發行日期和停機時間)。	移民專家（面試官）
取得伺服器清單。	取得與目標應用程式相關聯的伺服器清單。	移民專家（面試官）
獲取顯示當前狀態的邏輯圖。	向企業架構設計人員或應用程式擁有者取得應用程式的目前狀態圖。	移民專家（面試官）
建立顯示目標狀態的邏輯圖表。	建立顯示 AWS 上目標架構的應用程式邏輯圖。此圖表應說明伺服器、連線能力和對應因素。	企業建築師，企業主
取得伺服器資訊。	收集與應用程式相關聯之伺服器的相關資訊，包括其組態詳細資料。	移民專家（面試官）

任務	描述	所需技能
將伺服器資訊新增至探索範本。	將詳細的服務器信息添加到應用程序發現模板mobilize-application-questionnaire.xlsx 中（有關此模式，請參閱附件中的）。此範本包含所有與應用程式相關的安全性、基礎結構、作業系統和網路詳細資料。	移民專家（面試官）
發佈應用程式探索範本。	與應用程式擁有者和遷移團隊共用應用程式探索範本，以便進行一般存取和使用。	移民專家（面試官）

探索階段

任務	描述	所需技能
確認伺服器清單。	請與應用程式擁有者或技術主管確認伺服器清單以及每個伺服器的用途。	移民專家
識別並新增伺服器群組。	識別伺服器群組 (例如 Web 伺服器或應用程式伺服器)，並將此資訊新增至應用程式探索範本。選取每個伺服器所屬的應用程式層 (Web、應用程式、資料庫)。	移民專家
填寫應用程式探索範本。	在遷移團隊、應用程式團隊和 AWS 的協助下完成應用程式探索範本的詳細資料。	移民專家
添加缺少的服務器詳細信息（中間件和操作系統團隊）。	請中介軟體和作業系統 (OS) 團隊檢閱應用程式探索範本，並	移民專家

任務	描述	所需技能
	新增任何遺失的伺服器詳細資料，包括資料庫資訊。	
取得入站/出站流量規則 (網路團隊)。	要求網路小組取得來源和目的地伺服器的入站/出站流量規則。網路小組也應該新增現有的防火牆規則、將這些規則匯出為安全群組格式，並將現有的負載平衡器新增至應用程式探索範本。	移民專家
識別所需的標記。	決定應用程式的標籤需求。	移民專家
建立防火牆要求詳細資料	擷取並篩選與應用程式通訊所需的防火牆規則。	移轉專家、解決方案架構師、網路主管
更新 EC2 執行個體類型。	根據基礎設施和伺服器需求，更新要在目標環境中使用的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體類型。	移轉專家、解決方案架構師、網路主管
識別目前的狀態圖。	識別或建立顯示應用程式目前狀態的圖表。該圖將在信息安全（InfoSec）請求中使用。	解決方案架構師移轉專家
完成 future 狀態圖。	完成顯示應用程式 future (目標) 狀態的圖表。該圖也將在 InfoSec 請求中使用。	解決方案架構師移轉專家
建立防火牆或安全群組服務要求。	建立防火牆或安全群組服務要求 (適用於開發 /QA、生產前和生產)。如果您使用的是雲端移轉工廠，請加入複製特定連接埠 (如果尚未開啟)。	移轉專家、解決方案架構師、網路主管

任務	描述	所需技能
檢閱防火牆或安全群組要求 (InfoSec 團隊)。	在此步驟中，小 InfoSec 組會檢閱並核准在上一個步驟中建立的防火牆或安全性群組要求。	InfoSec 工程師，移民專家
實作防火牆安全群組要求 (網路群組)。	小 InfoSec 組核准防火牆要求之後，網路小組會實作必要的入站/輸出防火牆規則。	移轉專家、解決方案架構師、網路主管

建置階段 (針對開發/品質保證、生產前和生產環境重複)

任務	描述	所需技能
匯入應用程式和伺服器資料。	1. 確認您是以具有範圍內來源伺服器本機系統管理員權限的網域使用者身分登入移轉執行伺服器。 2. 使用移轉接入表單來匯入範圍內來源伺服器的屬性。如需其他資訊，請參閱雲端移轉工廠實作指南。 如果您沒有使用雲端移轉工廠，請遵循設定移轉工具的指示。	遷移專家，雲端管理員
檢查來源伺服器的必要條件。	與範圍內的來源伺服器連線，以驗證必要條件，例如 TCP Connect 埠 1500、TCP 連接埠 443、根磁碟區可用空間、.NET 架構版本和其他參數。這些都是複製所需的。如需其他資訊，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員

任務	描述	所需技能
建立服務要求以安裝複寫代理程式。	建立服務要求，在範圍內的伺服器上安裝複寫代理程式，以供開發 /QA、生產前或生產使用。	遷移專家，雲端管理員
安裝複寫代理程式。	在開發 /QA、生產前或生產機器上的範圍內來源伺服器上安裝複寫代理程式。如需其他資訊，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員
推送啟動後的指令碼。	應用程式遷移服務支援啟動後指令碼，協助您自動執行作業系統層級的活動，例如在啟動目標執行個體後安裝或解除安裝軟體 此步驟會根據識別要移轉的伺服器，將啟動後指令碼推送至 Windows 或 Linux 電腦。如需相關指示，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員
驗證複製狀態。	使用提供的指令碼，自動確認範圍內來源伺服器的複寫狀態。指令碼每五分鐘重複一次，直到指定波形中所有來源伺服器的狀態變更為 [正常] 為止。如需相關指示，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員

任務	描述	所需技能
建立管理員使用者。	從範圍內的來源伺服器移轉到 AWS 之後，可能需要來源機器上的本機管理員或 sudo 使用者，才能對任何問題進行疑難排解。當驗證伺服器 (例如 DC 或 LDAP 伺服器) 無法連線時，移轉小組會使用此使用者登入目標伺服器。如需此步驟的指示，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員
驗證啟動範本。	驗證伺服器中繼資料，以確保其成功運作，並且沒有無效的資料。此步驟會驗證測試和切換中繼資料。如需相關指示，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員

測試階段 (針對開發 /QA、生產前和生產環境重複)

任務	描述	所需技能
建立服務要求。	為基礎結構團隊和其他團隊建立服務請求，以執行應用程式切換至開發 /QA、生產前或生產執行個體。	遷移專家，雲端管理員
設定負載平衡器 (選用)。	使用 IRULES 設定必要的負載平衡器，例如應用程式負載平衡器或 F5 負載平衡器。	遷移專家，雲端管理員
啟動執行個體進行測試。	在測試模式下，在應用程式移轉服務中針對特定波形啟動所有目標電腦。如需其他資訊，	遷移專家，雲端管理員

任務	描述	所需技能
	請參閱雲端移轉工廠實作指南。	
驗證目標執行個體狀態。	檢查相同波形中所有範圍內來源伺服器的啟動程序，以驗證目標執行個體的狀態。目標執行個體最多可能需要 30 分鐘才能啟動。您可以登入 Amazon EC2 主控台、搜尋來源伺服器名稱並檢閱狀態檢查欄，以手動檢查狀態。通過的狀態 2/2 檢查表示從基礎架構的角度來看，執行個體狀況良好。	遷移專家，雲端管理員
修改 DNS 項目。	修改網域名稱系統 (DNS) 項目。（使用`resolv.conf` 或用`host.conf` 於 Microsoft 視窗環境。）將每個 EC2 執行個體設定為指向此主機的新 IP 位址。 注意：請確定現場部署和 AWS 雲端伺服器之間沒有 DNS 衝突。根據託管伺服器的環境而定，此步驟和下列步驟是可選的。	遷移專家，雲端管理員
測試 EC2 執行個體與後端主機的連線。	使用已移轉伺服器的網域認證來檢查登入。	遷移專家，雲端管理員
更新 DNS A 記錄。	更新每台主機的 DNS A 記錄，以指向新的 Amazon EC2 私有 IP 地址。	遷移專家，雲端管理員

任務	描述	所需技能
更新 DNS CNAME 記錄。	更新虛擬 IP (負載平衡器名稱) 的 DNS CNAME 記錄，以指向 Web 和應用程式伺服器的叢集。	遷移專家，雲端管理員
在適用的環境中測試應用程式。	登入新的 EC2 執行個體，並在開發 /QA、生產前和生產環境中測試應用程式。	遷移專家，雲端管理員
標記為準備切換。	測試完成時，請變更來源伺服器的狀態，以指出其已準備好進行切換，以便使用者可以啟動切換執行個體。如需相關指示，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員

切換階段

任務	描述	所需技能
建立生產部署計劃。	建立生產部署計劃 (包括取消計劃)。	遷移專家，雲端管理員
通知運營團隊停機時間。	通知營運團隊伺服器的停機時間排程。有些團隊可能需要變更請求或服務請求 (CR/SR) 票證才能進行此通知。	遷移專家，雲端管理員
複製生產機器。	使用應用程式移轉服務或其他移轉工具來複製生產機器。	遷移專家，雲端管理員
關閉範圍內的來源伺服器。	驗證來源伺服器的複寫狀態之後，您可以關閉來源伺服器，以停止從用戶端應用程式到伺服器的交易。您可以在切換視	雲端管理員

任務	描述	所需技能
	窗中關閉來源伺服器。如需詳細資訊，請參閱雲端移轉工廠實作指南。	
啟動用於切換的執行個體。	在切換模式下，在應用程式移轉服務中針對特定波形啟動所有目標電腦。如需詳細資訊，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員
擷取目標執行個體 IP。	擷取目標執行個體的 IP。如果 DNS 更新是您環境中的手動程序，您必須取得所有目標執行個體的新 IP 位址。如需詳細資訊，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員
驗證目標伺服器連線。	更新 DNS 記錄之後，請使用主機名稱連線至目標執行個體以驗證連線。如需詳細資訊，請參閱雲端移轉工廠實作指南。	遷移專家，雲端管理員

相關資源

- 如何移轉
- AWS 雲端移轉工廠實作指南
- 透過雲端移轉工廠自動化大規模伺服器移轉
- AWS 應用程式遷移服務使用指南
- Migration Acceleration Program (MAP)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施

由馬尼什·加格 (AWS)、法尼庫馬爾·達迪 (AWS)、尼沙德曼卡 (AWS) 和拉傑尼什泰亞吉 (AWS) 創建

代碼存儲庫：aws-windows-failover-cluster-自動化	環境：PoC 或試點	來源：本地 SQL 伺服器資料庫
目標：EC2 上的 Microsoft SQL 服務器	R 類型：重新主機	工作量：Microsoft
技術：遷移; 基礎設施; DevOps	AWS 服務：AWS 管理 Microsoft AD; Amazon EC2; Amazon FSx; AWS Systems Manager	

Summary

如果您需要快速移轉大量 Microsoft SQL Server 永遠在容錯移轉叢集執行個體 (FCI)，這個模式可以協助您將佈建時間降到最低。藉由使用 Windows 檔案伺服器的自動化和 Amazon FSx，可減少人工作業、人為錯誤，以及部署大量叢集所需的時間。

此模式在 Amazon Web Services（AWS）上的多可用區域（異地同步備份）部署中為 SQL Server FCI 設置基礎設施。此基礎設施所需的 AWS 服務佈建可使用 AWS CloudFormation 範本自動化。在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上的 SQL Server 安裝和叢集節點建立是透過使用 PowerShell 命令執行的。

此解決方案使用高可用性異地同步備份 Amazon FSx (適用於 Windows) 檔案系統，做為儲存 SQL Server 資料庫檔案的共用見證。託管 SQL 伺服器的 Amazon FSx 檔案系統和 EC2 視窗執行個體會加入 Microsoft 活動目錄 (AWS 受管 Microsoft AD) 網域的相同 AWS Directory Service。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有足夠許可以使用 AWS CloudFormation 範本佈建資源的 AWS 使用者
- 適用於 Microsoft Active Directory 的 AWS Directory Service
- AWS Secrets Manager 中的登入資料，可在金鑰值組中向 AWS 受管 Microsoft AD 進行驗證：

- ADDomainName: <Domain Name>
- ADDomainJoinUserName: <Domain Username>
- ADDomainJoinPassword: <Domain User Password>
- Target0U: <Target OU Value>

注意：您將在 AWS Systems Manager 自動化中使用相同的金鑰名稱來進行 AWS 受管 Microsoft AD 加入活動。

- SQL Server 媒體檔案適用於 SQL Server 安裝和 Windows 服務或建立的網域帳戶，這些檔案將在叢集建立期間使用
- 虛擬私有雲端 (VPC)，在不同的可用區域中具有兩個公用子網路、可用區域中有兩個私有子網路、一個網際網路閘道、NAT 閘道、路由表關聯，以及一個跳躍伺服器

產品版本

- 視窗服務器 2012 R2 和 Microsoft SQL 服務器 2016

架構

源, 技術, 堆棧

- 使用共用磁碟機搭配 FCI 的內部部署 SQL 伺服器

目標技術堆疊

- 執行個體
- Amazon FSx for Windows File Server
- AWS Systems Manager Automation 手冊
- 網路組態 (VPC、子網路、網際網路閘道、NAT 閘道、跳躍伺服器、安全群組)
- AWS Secrets Manager
- AWS 受管 Microsoft AD
- Amazon EventBridge
- AWS Identity and Access Management (IAM)

目標架構

下圖顯示單一 AWS 區域中的 AWS 帳戶，其 VPC 包含兩個可用區域、兩個具有 NAT 閘道的公有子網路、第一個公有子網路中的跳躍伺服器、兩個私有子網路，每個子網路都有節點安全群組中 SQL Server 節點的 EC2 執行個體，以及連接到每個 SQL Server 節點的 Amazon FSx 檔案系統。AWS Directory Service、Amazon EventBridge、AWS Secrets Manager 和 AWS Systems Manager 也包含在內。

自動化和規模

- 您可以使用 AWS Systems Manager 加入 AWS 受管 Microsoft AD 並執行 SQL 伺服器安裝。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Directory Service 提供多種方式，可將 Microsoft 活動目錄 (AD) 與其他 AWS 服務搭配使用，例如 Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Relational Database Service 服務 (Amazon RDS) 適用於 SQL 伺服器，以及適用於 Windows 檔案伺服器的 Amazon FSx。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。

其他工具

- PowerShell是一個 Microsoft 的自動化和配置管理程序，可以在 Windows，Linux 和 macOS 上運行。此模式使用 PowerShell 腳本。

代碼存儲庫

此模式的代碼可在 GitHub aws-windows-failover-cluster-自動化存儲庫中使用。

最佳實務

- 用來部署此解決方案的 IAM 角色應遵循最低權限原則。如需詳細資訊，請參閱 IAM 文件。
- 遵循 AWS 最 CloudFormation 佳實務。

史诗

部署基礎架構

任務	描述	所需技能
部署系 Systems Manager CloudFormation 堆疊。	1. 登入您的 AWS 帳戶，然後開啟 AWS 管理主控台。 2. 導覽至主 CloudFormation 控台，然後上傳`ssm.yaml`範本以建立 Systems Manager CloudFormation 堆疊。提供下列參數的值： • StateUnJoinAssociationLoggingBucketName— 為範本將為記錄目的而建立的 S3 儲存貯體提供名稱。 • SSM 關聯 UnjoinName — 提供資源的名稱。`AWS::SSM::Association` • SSM AutomationDocumentName — 提供 Systems Manager 自動化手冊的名稱。	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	• EventBridgeName— 提供 EventBridge 事件匯流排的名稱。 3. 啟動`ssm.yaml` CloudFormation 範本以部署 Systems Manager CloudFormation 堆疊。範本會建立 Systems Manager Atomation runboook，該執行個體會在具有標籤的新 EC2 執行個體啟動時啟動。`ADJoined:FSXADD`自動化執行手冊會將執行個體新增至 AWS 受管 Microsoft AD 目錄。	

任務	描述	所需技能
部署基礎架構堆疊。	成功部署 Systems Manager 堆疊後，建立包含 EC2 執行個體節點、安全群組、Windows 檔案伺服器檔案系統的 Amazon FSx 和 IAM 角色的`infra`堆疊。 1. 導覽至主 CloudFormation 控台並啟動`infra-cf.yaml` 範本。若要部署此堆疊，需要下列參數： • `ActiveDirectoryId` — AWS 管理 Microsoft AD 的 ID • `ADDnsIpAddresses1` — AWS 管理 Microsoft AD 的主要 DNS IP 地址 • `ADDnsIpAddresses2` — AWS Microsoft AD 管理的次要 DNS IP 地址 • `FSxSecurityGroupName` — Amazon FSx 安全組的名稱 • `FSxWindowsFileSystemName` — Amazon FSx 驅動器的名稱 • `ImageID`— 用於建立 SQL 伺服器執行個體節點的基礎視窗 2012 R2 映像或 Amazon 機器映像 (AMI) 的識別碼	AWS DevOps、 DevOps 工程師

任務	描述	所需技能
	• KeyPairName — 要連接到 EC2 執行個體節點以進行存取的鍵值對 • Node1SecurityGroupName — 第一個節點安全性群組的名稱 • Node2SecurityGroupName — 第二個節點安全性群組的名稱 • OUSecretName — 包含 AWS 受管 Microsoft AD 資訊的秘密名稱 • PrivateSubnet1 — 第一個私有子網路的識別碼 • PrivateSubnet2 — 第二個私人子網路的識別碼 • SqlFSxFCIName — 套用至主節點和次要節點以及 Amazon FSx 的標籤名稱。 • SqlFSxServerNetBIOSName1 — 主 EC2 執行個體節點的名稱 (最多 15 個字元) • SqlFSxServerNetBIOSName2 — 次要 EC2 執行個體節點的名稱 (最多 15 個字元) • VPC— VPC 識別碼 • WorkloadInstanceType — EC2 執行個體的類型	

任務	描述	所需技能
	部署infra堆疊。堆疊會建立設定視窗 SQL 伺服器 FCI 所需的所有基礎結構元件。 2. EC2 執行個體節點啟動後，系統會叫用 Systems Manager 自動化文件，將這些執行個體加入 AWS 受管 Microsoft AD。您可以在 Systems Manager 主控台的 [自動化] 頁面上追蹤進度。	

設置視窗 SQL 服務器始終在 FCI

任務	描述	所需技能
安裝視窗工具。	1. 登入主要 EC2 執行個體，也就是節點 1。要安裝 Windows 功能（活動目錄和 FCI 工具），請運行以下 PowerShell 腳本。 `Install-WindowsFeature -Name RSAT-AD-Powershell,Failover-Clustering -IncludeManagementTools` `Install-WindowsFeature -Name RSAT-Clustering,RSAT-ADDS-Tools,RSAT-AD-Powershell,RSAT-DHCP,RSAT-DNS-Server`	AWS DevOps、 DevOps 工程師、DBA

任務	描述	所需技能
	2. 登入次要 EC2 執行個體 (節點 2)，然後執行相同的指令碼以啟用節點 2 上的功能。	
在使用中目錄網域服務中預先安裝叢集電腦物件。	若要在使用中目錄網域服務 (AD DS) 中預先安裝叢集名稱物件 (CNO)，並預先安裝叢集角色的虛擬電腦物件 (VCO)，請遵循 Windows Server 說明文件中的指示。	AWS DevOps、DBA、工程師 DevOps

任務	描述	所需技能
建立 WSFC。	若要建立 Windows 伺服器容錯移轉叢集 (WSFC) 叢集，請執行下列動作： 1. 登入主要 EC2 執行個體，也就是節點 1。若要建立 Amazon FSx 檔案共用並授與所列 AD 服務帳戶的完整存取權，請執行下列程式碼。 `Invoke-Command -ComputerName "<FSx Windows Remote PowerShell Endpoint>" -ConfigurationName FSxRemoteAdmin -scriptblock { New-FSxSmbShare -Name "SQLDB" -Path "D:\share" -Description "SQL Databases Share" -ContinuouslyAvailable $true -FolderEnumerationMode AccessBased -EncryptData $true grant-fsxsmbshareaccess -name SQLDB -AccountName "<domain\user>" -accessRight Full }` 這個命令也會建立持續可用的 (CA) 檔案共用，這是最佳化以供 Microsoft SQL Server 使用。	AWS DevOps、DBA、工程師 DevOps

任務	描述	所需技能
	2. 若要在主要執行個體 (節點 1) 上建立容錯移轉叢集，請執行下列命令。 `New-Cluster -Name <CNO Name> -Node <Node1 Name>, <Node2 Name> -StaticAddress <Node1 Secondary Private IP>, <Node2 Secondary Private IP>` 該命令需要以下參數： • Name— 叢集的名稱 (CNO) • Node-主節點和次要節點的名稱，分別 • StaticAddress — 主要節點和次要節點的次要 IP 位址 重要事項：網域管理員或一般使用者必須擁有兩個節點的系統管理員權限，才能建立 Windows Server 容錯移轉叢集 (WSFC) 叢集。否則，上一個命令將失敗並返回消息，`You do not have administrator privilege on servers`。 3. 建立叢集之後，執行下列命令以附加檔案共用見證。 `Set-ClusterQuorum -FileShareWitness \`	

任務	描述	所需技能
	`\<FSx Windows Remote PowerShell Endpoint>\share\witness`	

任務	描述	所需技能
安裝 SQL 伺服器容錯移轉叢集。	WSFC 叢集設定完成後，請在主要執行個體 (節點 1) 上安裝 SQL Server 叢集。 1. 在兩個節點的 T 驅動器中，創建`tempdb`和`log`文件夾。這些文件夾在 PowerShell 命令中使用。 2. 在兩個節點上複製 SQL Server 安裝的 SQL Server 媒體檔案之後，請在節點 1 上執行下列 PowerShell 命令，以在節點 1 上安裝 SQL Server。 `D:\setup.exe /Q ` `/ACTION=InstallFailoverCluster ` `/IACCEPTSQLSERVERLICENSETERMS ` `/FEATURES="SQL,IS,BC,Conn" ` `/INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server” ` `/INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server" ` `/RSINSTALLMODE="FilesOnlyMode" ` `/INSTANCEID="MSSQLSERVER" ` `/INSTANCENAME="MSSQLSERVER" ` `/FAILOVERCLUSTERGROUP="SQL Server (MSSQLSERVER)" `	AWS DevOps、DBA、工程師 DevOps

任務	描述	所需技能
	/FAILOVERCLUSTER IPADDRESSES="IPv4; <2nd Sec Private Ip node1>;Cluster Network 1;<subnet mask>" ` /FAILOVERCLUSTER NETWORKNAME="<Fail over cluster Network Name>" ` /INSTANCEDIR="C: \Program Files\Mic rosoft SQL Server" ` /ENU="True" ` /ERRORREPORTING=0 ` /SQMREPORTING=0 ` /SAPWD="<Domain User password>" ` /SQLCOLLATION="S QL_Latin1_General_ CP1_CI_AS" ` /SQLSYSADMINACCO UNTS="<domain\user name>" ` /SQLSVCACCOUNT=" <domain\username>" /SQLSVCPASSWORD="< Domain User password>" ` /AGTSVCACCOUNT=" <domain\username>" /AGTSVCPASSWORD="< Domain User password>" ` /ISSVCACCOUNT="<domain \username>" /ISSVCPAS SWORD="<Domain User password>" ` /FTSVCACCOUNT="NT Service\MSSQLFDLau ncher" ` /INSTALLSQLDATADIR="\ \<FSX DNS name>\sha	

任務	描述	所需技能
	re\Program Files\Mic rosoft SQL Server" ` /SQLUSERDBDIR="\\<FSX DNS name>\share\data" ` /SQLUSERDBLOGDIR="\ \<FSX DNS name>\share \log" ` /SQLTEMPDBDIR="T: \tempdb" ` /SQLTEMPDBLOGDIR="T: \log" ` /SQLBACKUPDIR="\\<FSX DNS name>\share\SQLBac kup" ` /SkipRules=Clust er_VerifyForErrors ` /INDICATEPROGRESS	

任務	描述	所需技能
將次要節點新增至叢集。	若要將 SQL Server 新增至次要節點 (節點 2)，請執行下列PowerShell 命令。 `D:\setup.exe /Q ` `/ACTION=AddNode ` `/IACCEPTSQLSERVERLICENSETERMS ` `/INSTANCENAME="MSSQLSERVER" ` `/FAILOVERCLUSTERGROUP="SQL Server (MSSQLSERVER)" ` `/FAILOVERCLUSTERIPADDRESSES="IPv4;<2nd Sec Private Ip node2>;Cluster Network 2;<subnet mask>" ` `/FAILOVERCLUSTERNETWORKNAME="<Fail over cluster Network Name>" ` `/CONFIRMIPDEPENDENCYCHANGE=1 ` `/SQLSVCACCOUNT="<domain\username>" /SQLSVCPASSWORD="<Domain User password>" ` `/AGTSVCACCOUNT="domain\username>" /AGTSVCPASSWORD="<Domain User password>" ` `/FTSVCACCOUNT="NT Service\MSSQLFDLauncher" ` `/SkipRules=Cluster_VerifyForErrors ` `/INDICATEPROGRESS`	AWS DevOps、DBA、工程師 DevOps

任務	描述	所需技能
測試 SQL 伺服器 FCI。	1. 在其中一個節點的 Windows 執行個體上，在 [系統管理工具] 中，啟動容錯移轉叢集管理員。 2. 瀏覽至「節點」，並確認節點狀態為「狀態執行中」。 3. 選取角色，開啟 SQL Server (MSSQLSERVER) 的內容 (按一下滑鼠右鍵) 功能表，然後選取 [移動並選取節點]。 4. 選取節點之後，SQL Server 應該會在另一個節點上執行。	DBA，工程師 DevOps

清除資源

任務	描述	所需技能
清理資源。	若要清理資源，請使用 AWS CloudFormation 堆疊刪除程序： 1. 開啟 AWS CloudFormation 主控台。 2. 在「堆疊」頁面上，選取`infra`堆疊。此堆疊目前必須正在執行。 3. 在 stack details (堆疊詳細資訊) 窗格中，選擇 Delete (刪除)。 4. 當系統提示時，選取 Delete stack (刪除堆疊)。	AWS DevOps、DBA、工程師 DevOps

任務	描述	所需技能
	5. 對ssm堆疊重複步驟 2-4。 堆疊刪除完成後，堆疊就會處於狀DELETE_COMPLETE 態。DELETE_COMPLETE 狀態中的堆疊預設不會顯示在 CloudFormation 主控台中。若要顯示已刪除的堆疊，您必須變更堆疊檢視篩選器，如在 AWS CloudFormation 主控台上檢視已刪除的堆疊中所述。 如果刪除失敗，堆疊將處於狀DELETE_FAILED 態。如需解決方案，請參閱 CloudFormation 文件中的刪除堆疊失敗。	

故障診斷

問題	解決方案
AWS CloudFormation 範本失敗	如果 CloudFormation 範本在部署期間失敗，請執行下列動作： 1. 開啟 AWS CloudFormation 主控台。 2. 在 CloudFormation 主控台的 [堆疊] 頁面上，選取堆疊。 3. 選擇事件，並檢查堆疊狀態。
AWS 受管 Microsoft AD 加入失敗	若要疑難排解聯結問題，請依照下列步驟執行： 1. 開啟 Systems Manager 主控台。 2. 選取部署區域。

問題	解決方案
	3. 在左窗格中，選擇 [自動化]，然後找出故障的自動化工作流程簿。 4. 打開自動化手冊，並檢查執行狀態和執行步驟。 5. 調查失敗步驟的詳細資訊，以查看確切的錯誤或失敗。

相關資源

- 使用適用於 Windows 檔案伺服器的 Amazon FSx 簡化您的 Microsoft SQL 伺服器高可用性部署
- 使用 FSx 的 FSx for Windows File Server 與 Microsoft SQL 伺服器

使用 BMC 探索查詢擷取移轉資料以進行移轉規劃

由本·泰勒-漢布林（AWS），西蒙·坎寧安（AWS），艾瑪·鮑德里（AWS）和夏普南·汗（AWS）創建

環境：生產	資料來源：BMC 探索	目標：移轉計劃
R 類型：重新主機	工作負載：所有其他工作	技術：移轉、管理與治理、網路、混合雲
AWS 服務：AWS Migration Hub		

Summary

本指南提供查詢範例和步驟，協助您使用 BMC Discovery 從內部部署基礎結構和應用程式擷取資料。此病毒碼顯示如何使用 BMC 探索查詢掃描基礎結構，並擷取軟體、服務和相依性資訊。對於大規模遷移到 Amazon Web Services (AWS) 雲端的評估和動員階段，需要擷取的資料。您可以使用此資料做出重要決策，決定要在移轉計劃中一起移轉哪些應用程式。

先決條件和限制

先決條件

- BMC 探索 (以前稱 BMC ADDM) 或軟體即服務 (SaaS) 的 BMC 螺旋探索版本的授權
- 已安裝內部部署或 SaaS 版本的 BMC Discovery (注意：對於內部部署版本的 BMC Discovery，您必須在具有存取範圍內的所有網路和伺服器裝置的用戶端網路上安裝應用程式，以便跨多個資料中心進行移轉。 必須根據應用程式安裝說明提供對用戶端網路的存取。 如果需要掃描 Windows 伺服器資訊，則您必須在網路中設定 Windows 代理管理員裝置。)
- 如果您使用的是 BMC Helix 探索，網路存取可讓應用程式掃描跨資料中心的裝置

產品版本

- BMC 探索車 22.2 (12.5)
- BMC 的探索結果 22.1 (12.4)
- 小輪車的發現 21.3 (12.3)
- BMC 探索小輪車

- 小輪車探索計劃 20.08 (12.1)
- BMC 探索車 (12.0)
- 小輪車發現 11.3
- BMC 發現 11.2
- BMC 發現 11.1
- BMC 發現 11.0
- BMC 探索中庭酒店 10.2
- BMC 探索中庭酒店 10.1
- BMC 探索中庭酒店 10.0

架構

下圖顯示資產管理者如何使用 BMC 探索查詢來掃描 SaaS 和內部部署環境中的 BMC 模型應用程式。

該圖顯示了以下工作流程：資產管理員使用 BMC 探索或 BMC Helix 探索來掃描在託管在多個實體伺服器上的虛擬伺服器上執行的資料庫和軟體執行個體。該工具可以使用跨越多個虛擬和實體伺服器的元件建模應用程式。

技術, 堆

- BMC 探索
- BMC 螺旋探索

工具

- BMC 探索是一種資料中心探索工具，可協助您自動探索資料中心。
- BMC Helix 探索是以 SaaS 為基礎的探索與相依性建模系統，可協助您動態建立資料資產及其相依性的模型。

最佳實務

移轉至雲端時，最佳做法是對應應用程式、相依性和基礎結構資料。對應可協助您瞭解目前環境的複雜性，以及各種元件之間的相依性。

這些查詢提供的資產資訊非常重要，原因如下：

1. 規劃 — 瞭解元件之間的相依性可協助您更有效地規劃移轉程序。例如，您可能需要先移轉某些元件，以確保其他元件可以成功移轉。
2. 風險評估 — 對應元件之間的相依性可協助您識別移轉程序期間可能發生的任何潛在風險或問題。例如，您可能會發現某些元件依賴過時或不受支援的技術，這些技術可能會導致雲端發生問題。
3. 雲端架構 — 對應您的應用程式和基礎架構資料也可以協助您設計符合您組織需求的合適雲端架構。例如，您可能需要設計多層架構來支援高可用性或延展性需求。

整體而言，對應應用程式、相依性和基礎架構資料是雲端移轉程序中的關鍵步驟。對應練習可協助您更瞭解目前的環境、識別任何潛在問題或風險，以及設計合適的雲端架構。

史诗

識別和評估探索工具

任務	描述	所需技能
識別 ITSM 擁有者。	識別 IT 服務管理 (ITSM) 擁有者 (通常透過聯絡營運支援團隊)。	遷移, 領導
檢查 CMDB。	識別包含資產資訊的組態管理資料庫 (CMDB) 數目，然後識別該資訊的來源。	遷移, 領導
識別探索工具並檢查 BMC 探索的使用情況。	如果您的組織使用 BMC 探索將您環境的相關資料傳送至 CMDB 工具，請檢查其掃描的範圍和涵蓋範圍。例如，檢查 BMC 探索是否正在掃描所有資料中心，以及存取伺服器是否位於周邊區域。	遷移, 領導
檢查應用程序建模的級別。	檢查應用程式是否在「BMC 探索」中建模。如果不是，建議您使用 BMC 探索工具來建模	移民工程師，移民負責人

任務	描述	所需技能
	執行中的軟體執行個體提供應用程式和商業服務。	

擷取基礎架構資

任務	描述	所需技能
在物理和虛擬服務器上提取數據。	若要擷取 BMC 探索所掃描之實體與虛擬伺服器上的資料，請使用「查詢建構器」執行下列查詢： `search Host show key as 'Serverid', virtual, name as 'HOSTNAME', os_type as 'osName', os_version as 'OS Version', num_logical_processors as 'Logical Processor Counts', cores_per_processor as 'Cores per Processor', logical_ram as 'Logical RAM', #Consumer:StorageUse:Provider:DiskDrive.size as 'Size'` 附註：您可以使用萃取資料來確定移轉的適當執行個體大小。	移民工程師，移民負責人
擷取建模應用程式上的資料。	如果您的應用程式是在 BMC 探索中建模，您可以擷取執行應用程式軟體之伺服器的相關資料。若要取得伺服器名稱，	BMC 探索應用程式擁有者

任務	描述	所需技能
	請使用「查詢建構器」執行下列查詢： `search SoftwareInstance show key as 'ApplicationID', #RunningSoftware:HostedSoftware:Host:Host.key as 'ReferenceID', type, name` 注意：「BMC 探索」中的應用程式是由一組執行中的軟體實例來建模。該應用程序依賴於運行應用程序軟件的所有服務器。	

任務	描述	所需技能
提取數據庫上的數據。	若要取得所有已掃描資料庫和執行這些資料庫的伺服器清單，請使用 Query Builder 執行下列查詢： `search Database show key as 'Key', name, type as 'Source Engine Type', #Detail:Detail:ElementWithDetail:SoftwareInstance.name as 'Software Instance', #Detail:Detail:ElementWithDetail:SoftwareInstance.product_version as 'Product Version', #Detail:Detail:ElementWithDetail:SoftwareInstance.edition as 'Edition', #Detail:Detail:ElementWithDetail:SoftwareInstance.#RunningSoftware:HostedSoftware:Host:Host.key as 'ServerID'`	應用所有者

任務	描述	所需技能
擷取伺服器通訊上的資料。	若要取得 BMC Discovery 從歷史網路通訊記錄所收集之伺服器之間所有網路通訊的相關資訊，請使用「查詢建構器」來執行下列查詢： `search Host TRAVERSE InferredElement:Inference:Associate:DiscoveryAccess TRAVERSE DiscoveryAccess:DiscoveryAccessResult:DiscoveryResult:NetworkConnectionList TRAVERSE List:List:Member:DiscoveredNetworkConnection PROCESS WITH networkConnectionInfo`	BMC 探索應用程式擁有者
擷取應用程式探索的資料。	若要取得應用程式相依性的資訊，請使用「查詢建構器」執行下列查詢： `search SoftwareInstance show key as 'SRC App ID', #Dependant:Dependency:DependedUpon:SoftwareInstance.key as 'DEST App ID'`	BMC 探索應用程式擁有者

任務	描述	所需技能
提取業務服務的數據。	若要擷取主機所提供之業務服務的資料，請使用「查詢建構器」執行下列查詢： `search Host show name, #Host:HostedSoftware:AggregateSoftware:BusinessService.name as 'Name'`	BMC 探索應用程式擁有者

故障診斷

問題	解決方案
查詢無法執行或包含未填入的資料行。	檢閱「BMC 探索」中的資產記錄，並決定您需要哪些欄位。然後，使用「查詢建置器」取代查詢中的這些欄位。
相依資產的詳細資訊不會填入。	這可能是由於訪問權限或網絡連接。探索工具可能沒有存取特定資產的必要權限，特別是在不同網路或不同環境中的情況下。 我們建議您與探索主題專家密切合作，以確保識別所有相關資產。

相關資源

參考

- BMC 探查授權權利 (BMC 文件)
- BMC 探索功能與元件 (BMC 文件)
- BMC 探索使用者指南 (BMC 文件)
- 搜尋資料 (關於 BMC 探索) (BMC 文件)
- 移轉的產品組合探索和分析 (AWS Prescriptive Guidance)

教學課程和影片

- BMC 探索：網路研討會-報告查詢最佳做法 (第 1 部分) (YouTube)

搬遷

主題

- 使用 AWS DMS 將適用於甲骨文資料庫的 Amazon RDS 遷移到另一個 AWS 帳戶和 AWS 區域以進行持續複寫
- 使用 VMware 硬體校驗，將 VMware 軟體定義的軟體定義資料中心遷移至 VMware 雲端
- 將 Amazon RDS 資料庫執行個體遷移到另一個 VPC 端或帳戶
- 將 Amazon RDS for Oracle 執行個體遷移到另一個 VPC
- 將 Amazon Redshift 叢集遷移到中國的 AWS 區域
- 使用 VMware HCX 將工作負載遷移到 AWS 上的 VMware 雲端
- 使用傳輸在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫

使用 AWS DMS 將適用於甲骨文資料庫的 Amazon RDS 遷移到另一個 AWS 帳戶和 AWS 區域以進行持續複寫

由杜爾加普拉薩德奇普里（AWS）和愛德華多·瓦倫蒂姆（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for Oracle
R 類型：搬遷	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

警告：IAM 使用者擁有長期登入資料，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。

此模式會引導您完成將 Oracle 來源 Amazon Relational Database Service (Amazon RDS) 遷移到不同的 AWS 帳戶 和的步驟 AWS 區域。該模式使用資料庫快照集進行一次性的完整資料載入，並啟用 AWS Database Migration Service (AWS DMS) 進行中的複寫。

先決條件和限制

先決條件

- 包 AWS 帳戶 含來源 Amazon RDS for Oracle 資料庫的作用中，該資料庫已使用非預設 AWS Key Management Service (AWS KMS) 金鑰加密
- 一個活動 AWS 帳戶 在不同於 AWS 區域 來源資料庫，用於目標 Amazon RDS for Oracle 文資料庫
- 來源和目標 VPC 之間的虛擬私有雲 (VPC) 對等互連
- 熟悉使用 Oracle 資料庫作為下列項目的來源： AWS DMS
- 熟悉使用 Oracle 資料庫作為下列項目的目標： AWS DMS

產品版本

- 甲骨文版本 11 克（版本 11.2.0.3.v1 及更高版本）和最高 12.2 和 18c。如需支援版本和版本的最新清單，請參閱使用 Oracle 資料庫作為來源，以 AWS DMS及使用 Oracle 資料庫作為 AWS 文件 AWS DMS中的目標。如需 Amazon RDS 支援的甲骨文版本，請參閱 Amazon RDS 上的甲骨文。

架構

來源與目標技術堆疊

- Amazon RDS for Oracle 數據庫

持續複寫架構

工具

用於一次性完整資料載入的工具

- Amazon Relational Database Service (Amazon RDS) 會建立資料庫執行個體的儲存磁碟區快照，備份整個資料庫執行個體，而不僅備份個別資料庫。建立資料庫快照時，您必須找出要進行備份的資料庫執行個體，並為該資料庫快照命名，使得您稍後可透過它進行還原。建立快照所需的時間量因資料庫的大小而異。由於快照包括整個儲存體磁碟區，檔案大小，例如暫存檔案，也會影響建立快照所需的時間量。如需使用資料庫快照的詳細資訊，請參閱 Amazon RDS 文件中的建立資料庫快照。
- AWS Key Management Service （ AWS KMS ）為 Amazon RDS 加密創建密鑰。 建立加密的資料庫執行個體時，您也可以提供加密AWS KMS金鑰的金鑰識別碼。如果您未指定AWS KMS金鑰識別碼，Amazon RDS 會使用新資料庫執行個體的預設加密金鑰。 AWS KMS會建立您的預設加密金鑰 AWS 帳戶。您的每個密鑰都 AWS 帳戶 有不同的默認加密密鑰 AWS 區域。對於此模式，應使用非預設AWS KMS金鑰加密 Amazon RDS 資料庫執行個體。如需使用AWS KMS金鑰進行 Amazon RDS 加密的詳細資訊，請參閱 Amazon RDS 文件中的加密 Amazon RDS 資源。

用於進行中複寫的工具

- AWS Database Migration Service (AWS DMS) 用於複寫進行中的變更，以及保持來源和目標資料庫同步。如需有關使用進行中複寫的 AWS DMS 詳細資訊，請參閱 AWS DMS 文件中的使用 AWS DMS 複寫執行個體。

史诗

設定您的來源 AWS 帳戶

任務	描述	所需技能
準備來源 Oracle 資料庫執行個體。	讓適用於 Oracle 資料庫的 Amazon RDS 執行個體以存檔日誌模式執行，並設定保留期。如需詳細資訊，請參閱使用 AWS 受管理的 Oracle 資料庫作為來源 AWS DMS。	DBA
設定來源 Oracle 資料庫執行個體的補充記錄日誌。	為適用於 Oracle 資料庫執行個體的 Amazon RDS 設定資料庫層級和表格層級補充記錄。如需詳細資訊，請參閱使用 AWS 受管理的 Oracle 資料庫作為來源 AWS DMS。	DBA
更新來源帳戶中的 AWS KMS 金鑰策略。	更新來源中的 AWS KMS 金鑰政策， AWS 帳戶 以允許目標 AWS 帳戶 使用加密的 Amazon RDS AWS KMS 金鑰。如需詳細資訊，請參閱AWS KMS 文件。	SysAdmin
建立來源資料庫執行個體的手動 Amazon RDS 資料庫快照。		AWS IAM 使用者
將手動加密的 Amazon RDS 快照與目標共用 AWS 帳戶。	如需詳細資訊，請參閱共用資料庫快照。	AWS IAM 使用者

設定您的目標 AWS 帳戶

任務	描述	所需技能
附加策略。	在目標中 AWS 帳戶，將 AWS Identity and Access Management (IAM) 政策附加到根 IAM 使用者，以允許 IAM 使用者使用共用 AWS KMS 金鑰複製加密的資料庫快照。	SysAdmin
切換到源 AWS 區域。		AWS IAM 使用者
複製共用快照。	在 Amazon RDS 主控台的「快照」窗格中，選擇「與我共用」，然後選取共用快照。使用來源資料庫所使用 AWS KMS 金鑰的 Amazon 資源名稱 (ARN)，將快照複製到與 AWS 區域 來源資料庫相同的快照。如需詳細資訊，請參閱複製資料庫快照。	AWS IAM 使用者
切換到目標 AWS 區域，並創建一個新的 AWS KMS 密鑰。		AWS IAM 使用者
複製快照。	切換到源 AWS 區域。在 Amazon RDS 主控台的「快照」窗格中，選擇「我擁有」，然後選取複製的快照。使用新目標 AWS 區域 的 AWS KMS 金鑰將快照複製到目標 AWS 區域。	AWS IAM 使用者
還原快照。	切換到目標 AWS 區域。在 Amazon RDS 主控台的「快照」窗格中，選擇「我擁有」。選取複製的快照，然後	AWS IAM 使用者

任務	描述	所需技能
	將其還原到適用於 Oracle 資料庫的 Amazon RDS 執行個體。如需詳細資訊，請參閱從資料庫快照還原。	

準備來源資料庫以進行中的複寫

任務	描述	所需技能
建立具有適當權限的 Oracle 使用者。	建立具有 Oracle 所需權限的 Oracle 使用者，作為的來源 AWS DMS。如需詳細資訊，請參閱AWS DMS 文件。	DBA
設定 Oracle LogMiner 或 Oracle 二進位讀取器的來源資料庫。		DBA

準備目標資料庫以進行持續複寫

任務	描述	所需技能
建立具有適當權限的 Oracle 使用者。	建立具有 Oracle 所需權限的 Oracle 使用者，作為的目標 AWS DMS。如需詳細資訊，請參閱AWS DMS 文件。	DBA

建立 AWS DMS 元件

任務	描述	所需技能
在目標中建立複寫執行個體 AWS 區域。	在目 AWS 區域標的 VPC 中建立複寫執行個體。如需詳細資訊，請參閱AWS DMS 文件。	AWS IAM 使用者
使用必要的加密建立來源和目標端點，並測試連線。	如需詳細資訊，請參閱AWS DMS 文件。	DBA
建立複寫工作。	1. 針對移轉類型，選擇進行中的複寫。 2. 對於變更資料擷取 (CDC) 的起點，請在拍攝 Amazon RDS 快照進行全負載時使用 Oracle 系統變更編號 (SCN)，或使用滿載時的時間戳記。 3. 對於`TargetTablePrepMode`，請選擇「不做 _」。如果工作具有大型二進位物件 (LOB) 資料表，請選擇受限 LOB 模式，然後將 LOB 大小上限設定為表格中 LOB 資料的大小上限。 4. 啟用記錄。 5. 將透過索引鍵相關的資料表群組成單一工作。如果有包含大量 LOB 資料的表格，而且表格與其他表格沒有任何關係，請使用先前描述的 LOB 設定值來為其建立個別的工作。	IAM 使用者

任務	描述	所需技能
	如需詳細資訊，請參閱AWS DMS 文件。	
啟動任務並對其進行監視。	如需詳細資訊，請參閱AWS DMS 文件。	AWS IAM 使用者
視需要啟用工作的驗證。	請注意，啟用驗證確實會影響複寫的效能。如需詳細資訊，請參閱AWS DMS 文件。	AWS IAM 使用者

相關資源

- 變更金鑰政策
- 建立手動 Amazon RDS 資料庫快照
- 共用手動 Amazon RDS 資料庫快照
- 複製快照
- 從 Amazon RDS 數據庫快照恢復
- 開始使用 AWS DMS
- 使用 Oracle 資料庫作為下列項目的來源 AWS DMS
- 使用 Oracle 資料庫作為下列項目的目標 AWS DMS
- AWS DMS 使用 VPC 對等互連進行設定
- 如何與另一個人共用手動 Amazon RDS 資料庫快照或資料庫叢集快照 AWS 帳戶？ (AWS 知識中心文章)

使用 VMware 硬體校驗，將 VMware 軟體定義的軟體定義資料中心遷移至 VMware 雲端

由迪帕克庫馬爾（AWS）創建

環境：PoC 或試點	來源：網絡	目標：VMware Cloud on AWS
R 類型：搬遷	技術：移轉；基礎架構	

Summary

注意：自 2024 年 4 月 30 日起，VMware 雲端服務不再由 AWS 或其通路合作夥伴轉售。 AWS 該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何使用 VMware 混合雲擴充功能 (HCX) 將您的現場部署虛擬機器 (VM) 和應用程式遷移到 VMware 雲端 Amazon Web Services (AWS)。遷移作業使用 AWS 雲端上的 VMware 企業級軟體定義資料中心 (SDDC) 軟體，為 AWS 服務提供最佳化存取權。

VMware Cloud on AWS 將運算、儲存和網路虛擬化產品 (vSphere、vSAN 和 VMware NSX) 與 VMware vCenter 伺服器管理整合，這項管理經過最佳化，可在專用、彈性的裸機 AWS 基礎設施上執行。所產生的基礎架構是低維護、簡化且超融合的。

透過這項服務，IT 團隊可以使用熟悉的 VMware 工具來管理雲端資源。如需詳細資訊，請參閱 VMware 網站上的 VMware 雲端服務。

VMware HCX 支援三種類型的雲端移轉：

- 混合性 (資料中心延伸模組)：將現有的現場部署 VMware 軟體定義的資料中心延伸至 AWS，以提供足跡擴充、隨需容量、測試/開發環境以及虛擬桌面。
- 雲端疏散 (整個資料中心的基礎設施重新整理)：整合資料中心並完全移轉至 AWS 雲端 (包括處理資料中心主機代管或租用結束)。
- 應用程式特定遷移：將個別應用程式移至 AWS 雲端以滿足特定業務需求。

先決條件和限制

先決條件

- 註冊 AWS 帳戶 (建立 VMware 雲端軟體定義的資料中心時需要使用)。
- 註冊我的 VMware 帳戶。在 https://my.vmware.com/web/vmware/ 註冊並填寫所有字段。
- 檢查 vCenter 和主機的版本，並收集虛擬機器的數目。如果可能，請要求 RVTools 匯出以顯示有關虛擬環境的資訊。我們建議使用 vCenter 6.0 或更高版本。
- 如果您想要擴充資料中心網路 (L2)、使用 HCX 測試 vMotion，或使用 vRealize 網路洞察來分析應用程式相依性，則必須部署分散式虛擬交換器。
- 挑選不衝突的現場部署目前管理子網路，以在 VMware Cloud on AWS 建立軟體定義的資料中心。
- 檢閱《VMware HCX 使用者指南》中提供的先決條件，以驗證 HCX 需求。
- 識別虛擬機器並分組以進行移轉浪潮。檢查可用於測試的虛擬機器。
- 收集有關相對頻寬消耗、WAN 壓縮和資料傳輸速度的任何資料。

備註

- 不需要在內部部署使用 VMware NSX-V 或 NSX-T。
- HCX 無需額外費用 (包含在 AWS 上的 VMware 雲端中)。

架構

下圖顯示建立在多元件服務上的 HCX 解決方案。每個元件都支援 HCX 解決方案中的特定功能。如需有關每個 HCX 元件的詳細資訊，請參閱將工作負載移轉至具有混合雲延伸功能 (HCX) 的 AWS 上的 VMware 雲端部落格文章。

源, 技術, 堆棧

- 由 VMware vSphere 管理的內部部署虛擬機器和應用

目標技術堆疊

- VMware Cloud on AWS

工具

- VMware HCX — VMware HCX 是一種工具，可讓您在資料中心和雲端環境之間移轉應用程式和工作負載。它包含在 AWS 上的 VMware 雲端中。

史诗

規劃移轉

任務	描述	所需技能
選擇移轉策略。	決定是要擴充資料中心 (混合性)、移動所有資料中心 (雲端疏散)，還是要將特定應用程式移至 AWS。	SysAdmin，應用所有者
驗證 HCX 要求。	如需移轉資訊，請參閱 VMware HCX 使用者指南。	SysAdmin，應用所有者

在 AWS 上遷移至 VMware 雲端

任務	描述	所需技能
遷移您的 VM 或應用程式。	如需詳細資訊，請參閱 VMware 說明文件中的使用 VMware HCX 進行混合式移轉。	SysAdmin，應用所有者

相關資源

- VMware Cloud on AWS：開始使用
- 使用 VMware HCX 進行混合式移轉
- VMware HCX 使用者指南
- VMware Cloud on AWS 定價
- VMware Cloud on AWS 發展藍圖

將 Amazon RDS 資料庫執行個體遷移到另一個 VPC 端或帳戶

創建者：德魯巴約提穆克吉 (AWS)

環境：PoC 或試點	資料來源：Amazon RDS	目標：Amazon RDS
R 類型：搬遷	技術：移轉；資料庫	AWS 服務：Amazon RDS; Amazon VPC

Summary

此模式提供將 Amazon 關聯式資料庫服務 (Amazon RDS) 資料庫執行個體從一個虛擬私有雲端 (VPC) 遷移到同一 AWS 帳戶中的另一個虛擬私有雲端 (VPC)，或從一個 AWS 帳戶遷移到另一個 AWS 帳戶的指導。

如果出於分離或安全原因 (例如，當您想要將應用程式堆疊和資料庫放置在不同的 VPC 中) 而想要將 Amazon RDS 資料庫執行個體遷移到另一個 VPC 或帳戶時，此模式非常有用。

將資料庫執行個體遷移到另一個 AWS 帳戶涉及的步驟，例如拍攝手動快照、共用快照，以及還原目標帳戶中的快照。此過程可能很耗時，具體取決於數據庫更改和交易率。這也會導致資料庫停機，因此請事先規劃移轉。考慮採用藍/綠部署策略，將停機時間降至最低 或者，您可以評估 AWS 資料遷移服務 (AWS DMS)，將變更的停機時間降到最低。但是，此模式不涵蓋此選項。若要進一步了解，請參閱 AWS DMS 文件。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- VPC、子網路和 Amazon RDS 主控台所需的 AWS Identity and Access Management (IAM) 許可

限制

- 變更 VPC 會導致資料庫重新開機，導致應用程式中斷。我們建議您在低尖峰時段移轉。
- 將 Amazon RDS 遷移到其他 VPC 時的限制：
  - 您要遷移的資料庫執行個體必須是沒有待命狀態的單一執行個體。它不能是叢集的成員。
  - Amazon RDS 不得位於多個可用區域中。

  - Amazon RDS 不得有任何僅供讀取複本。
  - 在目標 VPC 中建立的子網路群組必須具有來自執行來源資料庫之可用區域的子網路。
- 將 Amazon RDS 遷移到另一個 AWS 帳戶時的限制：
  - 目前不支援使用 Amazon RDS 的預設服務金鑰加密的共用快照。

架構

遷移至相同 AWS 帳戶中的 VPC 人雲端

下圖顯示將 Amazon RDS 資料庫執行個體遷移到同一 AWS 帳戶中不同 VPC 的工作流程。

這些步驟包括以下內容。有關詳細說明，請參閱史詩部分。

1. 在目標 VPC 中建立資料庫子網路群組。資料庫子網路群組是一組子網路，您可以在建立資料庫執行個體時用來指定特定 VPC。
2. 在來源 VPC 中設定 Amazon RDS 資料庫執行個體，以使用新的資料庫子網路群組。
3. 套用變更以將 Amazon RDS 資料庫遷移到目標 VPC。

遷移到不同的 AWS 帳戶

下圖顯示將 Amazon RDS 資料庫執行個體遷移到其他 AWS 帳戶的工作流程。

這些步驟包括以下內容。有關詳細說明，請參閱史詩部分。

1. 在來源 AWS 帳戶中存取 Amazon RDS 資料庫執行個體。
2. 在來源 AWS 帳戶中建立 Amazon RDS 快照。
3. 與目標 AWS 帳戶共享 Amazon RDS 快照。
4. 在目標 AWS 帳戶中存取 Amazon RDS 快照。
5. 在目標 AWS 帳戶中建立 Amazon RDS 資料庫執行個體。

工具

AWS 服務

- Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展關聯式資料庫。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

最佳實務

- 如果將 Amazon RDS 資料庫執行個體遷移到另一個帳戶時需要考慮資料庫停機時間，建議您使用 AWS DMS。此服務提供資料複寫，導致中斷時間少於五分鐘。

史诗

遷移到相同 AWS 帳戶中的不同 VPC

任務	描述	所需技能
建立新 VPC	在 Amazon VPC 主控台上，建立具有所需屬性和 IP 位址範圍的新 VPC 和子網路。如需詳細指示，請參閱 Amazon VPC 文件。	管理員
建立資料庫子網路群組。	在 Amazon RDS 控制台上： 1. 選擇子網路群組、建立資料庫子網路群組。 2. 輸入子網路群組名稱、說明和 VPC 識別碼。 3. 新增屬於子網路群組的子網路。新增子網路以涵蓋至少兩個可用區域。 4. 選擇建立。 如需其他資訊，請參閱 Amazon RDS 文件。	管理員

任務	描述	所需技能
修改 Amazon RDS 資料庫執行個體以使用新的子網路群組。	在 Amazon RDS 控制台上： 1. 在導覽窗格中，選擇「資料庫」，然後選擇要遷移的 Amazon RDS 資料庫執行個體。 2. 在 [連線] 區段中，選擇與目標 VPC 相關聯的子網路群組。 3. 在「排程修改」段落中，選擇「立即套用」。 移轉到目標 VPC 完成後，會將目標 VPC 的預設安全群組指派給 Amazon RDS 資料庫執行個體。您可以使用資料庫執行個體所需的輸入和輸出規則，為該 VPC 設定新的安全群組。 或者，透過明確提供新的 VPC 安全群組 ID，使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 執行到目標 VPC 的遷移。例如： `aws rds modify-db-instance \` `--db-instance-identifier testrds \` `--db-subnet-group-name new-vpc-subnet-group \` `--vpc-security-group-ids sg-idxxxx \`	管理員

任務	描述	所需技能
	`--apply-immediatel y`	

遷移到不同的 AWS 帳戶

任務	描述	所需技能
在目標 AWS 帳戶中建立新的 VPC 和子網路群組。	1. 在 Amazon VPC 主控台上，建立具有所需屬性和 IP 位址範圍的新 VPC。如需詳細指示，請參閱 Amazon VPC 文件。 2. 依照 Amazon VPC 文件中的指示，為新的 VPC 建立子網路。 3. 在 Amazon RDS 主控台上，建立資料庫子網路群組。如需指示，請參閱 Amazon RDS 文件。	管理員
共用資料庫的手動快照，並與目標帳戶共用。	1. 按照 Amazon RDS 說明文件中的指示拍攝來源資料庫的手動快照。 2. 透過提供目標帳戶 ID 與目標 AWS 帳戶共用快照。如需指示，請參閱有關與其他帳戶共用資料庫快照的 Re:post 文章。	管理員
啟動新的 Amazon RDS 資料庫執行個體。	從目標 AWS 帳戶的共用快照啟動新的 Amazon RDS 資料庫執行個體。如需指示，請參閱 Amazon RDS 文件。	管理員

相關資源

- Amazon VPC 文件
- Amazon RDS 文件
- 如何變更 RDS 資料庫執行個體的 VPC？（AWS RE：發布文章）
- 如何將 Amazon RDS 資源的擁有權轉移到不同的 AWS 帳戶？（AWS RE：發布文章）
- 如何與另一個 AWS 帳戶共用手動 Amazon RDS 資料庫快照或 Aurora 資料庫叢集快照？（AWS RE：發布文章）
- AWS DMS 說明文件

將 Amazon RDS for Oracle 執行個體遷移到另一個 VPC

創建者：皮尼什辛格爾（AWS）

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for Oracle
R 類型：搬遷	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此遷移模式提供 step-by-step 指導，說明將 Oracle 資料庫 (資料庫) 執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS) 從一個虛擬私有雲端 (VPC) 遷移到同一 Amazon Web Services (AWS) 帳戶中的另一個 VPC。例如，如果您的企業要求資料庫和 Amazon Elastic Compute Cloud (Amazon EC2) 應用程式伺服器位於相同的 VPC 中，則可以使用此模式。

該模式描述具有大量交易的多 TB Oracle 來源資料庫幾乎沒有停機時間的線上移轉策略。

若要將適用於 Oracle 資料庫的 Amazon RDS 執行個體移至另一個 VPC 人雲端，您必須變更 Amazon RDS 子網路群組。此子網路群組必須使用新的 VPC 和必要的子網路進行預先設定。在 VPC 從一個網路變更到另一個網路期間，Amazon RDS 執行個體會重新啟動，因此在移動過程中無法存取資料庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 兩個具有私有子網路的 VPC
- 適用於 Oracle 的 Amazon RDS 資料庫執行個體 (已啟動並執行)，已設定輸入和輸出安全群組

限制

- 不支援跨多個可用區域 (異地同步備份) 的資料庫執行個體。但是，此模式提供了一種解決此限制的方法。

- 僅供讀取複本開啟時，無法移轉資料庫執行個體。
- 新 VPC 中的子網路群組應與資料庫位於相同的可用區域中。
- 移轉應該在排程的維護期間或低流量時間進行，因為將資料庫移至另一個 VPC 會導致資料庫重新開機，導致應用程式中斷數分鐘。

產品版本

- Amazon RDS for Oracle 資料庫執行個體，12.1.0.2 及更新版

架構

源, 技術, 堆棧

- VPC 中適用於甲骨文 12.1.0.v22 資料庫執行個體的亞馬遜 RDS
- 在單獨的路由表中配置的 VPC
- 在 VPC 中設定的 Amazon RDS 子網路群組
- Amazon RDS 選項組（如果需要）

目標技術堆疊

- 適用於甲骨文資料庫執行個體的亞馬遜 RDS，在另一個 VPC 中使用版本 12.1.0.2.v22
- Amazon VPC 在單獨的路由中配置
- 在新的 VPC 中設定的 Amazon RDS 子網路群組
- Amazon RDS 選項組（如果需要）

來源與目標架構

下圖顯示如何使用主控台將 Amazon RDS for Oracle 資料庫從一個 VPC 中的私有子網路移至不同 VPC 中的私有子網路。

1. 使用主控台修改適用於 Oracle 資料庫執行個體的來源 Amazon RDS。
2. 在目標 VPC 中，修改子網路群組，並修改選項群組 (如果使用)。

工具

- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。它為關聯式資料庫提供符合成本效益且可調整大小的容量，並管理常見的資料庫管理工作。這種模式使用 Amazon RDS for Oracle。

史诗

變更現有 VPC 中 Amazon RDS for Oracle 文資料庫的組態

任務	描述	所需技能
建立子網路群組。	在 Amazon RDS 中設定子網路群組。	一般 AWS
建立選項群組。	(選擇性) 在 Amazon RDS 中設定選項群組。	一般 AWS
修改適用於 Oracle 資料庫執行個體的亞馬遜 RDS。	使用子網路群組和選項群組修改資料庫。	一般 AWS、DBA
如有必要，請更新 Oracle 資料庫。	若要移轉來源 Amazon RDS for Oracle 資料庫，請進行下列變更： • 移除僅供讀取複本 (如果存在)。 • 關閉異地同步備份功能 (如果已開啟)。	一般 AWS

在目標虛擬私人雲端中設定亞馬遜 RDS

任務	描述	所需技能
建立子網路群組。	在 Amazon RDS 中，使用新 VPC 的子網路和資料庫的可用區域來設定子網路群組。	一般 AWS

任務	描述	所需技能
建立選項群組。	(選擇性) 在 Amazon RDS 中設定選項群組。	一般 AWS
修改 Amazon RDS for Oracle 數據庫。	使用新的子網路群組和新 VPC 的選項群組來修改資料庫。您可以立即套用這些變更，也可以在維護時段中套用。 修改可能需要幾分鐘的時間才能完成。在修改期間，您會看到下列狀態變更： • moving-to-vpc • C onfiguring-enhanced-monitoring • Modifying (正在修改) • 可用性 修改將附加新 VPC 的預設安全性群組。根據 Amazon RDS for Oracle 的需要附加新的安全群組。	一般 AWS、DBA
如有必要，請更新 Amazon RDS for Oracle 數據庫。	在新 VPC 中移轉至目標 Amazon RDS for Oracle 資料庫之後，請視需要進行下列修改： • 如果僅供讀取複本存在於來源資料庫中，請開啟僅供讀取複本。 • 如果已在來源資料庫中開啟異地同步備份功能，請開啟此功能。	一般 AWS

任務	描述	所需技能
測試應用程式連線。	從任何應用程式執行資料庫連線測試。確認新虛擬私人雲端中已修改的 Amazon RDS 資料庫已連線，且可從應用程式存取。	應用所有者

相關資源

- Amazon VPC 文件
- 虛擬私人雲端和子網路
- 在 VPC 中使用資料庫執行個體
- Amazon RDS 文件
- Amazon RDS 上的甲骨文
- Amazon RDS 控制台
- 如何變更 Amazon RDS 資料庫執行個體的 VPC？

將 Amazon Redshift 叢集遷移到中國的 AWS 區域

由晶燕 (AWS) 創建

R 類型：搬遷	環境：生產	技術：資料庫；移轉
工作負載：所有其他工作	AWS 服務：Amazon Redshift	資料來源：AWS Redshift
目標：AWS Redshift		

Summary

此模式提供了一 step-by-step 種將 Amazon Redshift 叢集從另一個 AWS 區域遷移到中國的 AWS 區域的方法。

此模式使用 SQL 命令重新建立所有資料庫物件，並使用 UNLOAD 命令將此資料從 Amazon Redshift 移至來源區域中的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。然後，資料會遷移到位於中國 AWS 區域的 S3 儲存貯體。COPY 命令用於從 S3 儲存貯體載入資料，並將其傳輸到目標 Amazon Redshift 叢集。

Amazon Redshift 目前不支援跨區域功能，例如快照複製到中國的 AWS 區域。此模式提供了一種解決該限制的方法。您也可以反轉此模式中的步驟，將資料從中國的 AWS 區域遷移到另一個 AWS 區域。

先決條件和限制

先決條件

- 中國區域和中國以外 AWS 區域的有效 AWS 帳戶
- 中國區域和中國以外的 AWS 區域中的現有 Amazon Redshift 叢集

限制

- 這是離線遷移，這表示來源 Amazon Redshift 叢集無法在遷移期間執行寫入操作。

架構

源, 技術, 堆棧

- 亞馬遜紅移叢集位於中國以外的 AWS 區域

目標技術堆疊

- Amazon Redshift 叢集位於中國的 AWS 區域

目標架構

工具

工具

- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，可提供可擴展性、資料可用性、安全性和效能。您可以使用 Amazon S3 存放來自 Amazon Redshift 的資料，也可以將資料從 S3 儲存貯體複製到 Amazon Redshift。
- Amazon Redshift — Amazon Redshift 是雲端中的全受管 PB 級資料倉儲服務。
- psql — psql 是一個基於終端的前端到 PostgreSQL。

史诗

準備來源區域中的移轉

任務	描述	所需技能
在來源區域中啟動和設定 EC2 執行個體。	登入 AWS 管理主控台並開啟亞馬遜彈性運算雲端 (Amazon EC2) 主控台。您目前的區域會顯示在畫面頂端的導覽列中。此區域不能是中國的 AWS 區域。從 Amazon EC2 主控台儀表板選擇「啟動執行個體」，然後建立和設定 EC2 執行個體。重要事項：確保輸入規則的 EC2 安全群組允許從來源機器不受限制地存取 TCP 連接埠	DBA, 開發人員

任務	描述	所需技能
	22。如需如何啟動和設定 EC2 執行個體的指示，請參閱「相關資源」一節。	
安裝 psql 工具。	下載 PostgreSQL 裝 Amazon Redshift 不提供 psql 工具，它是與 PostgreSQL 一起安裝的。如需有關使用 psql 和安裝 PostgreSQL 工具的詳細資訊，請參閱 < 相關資源 > 一節。	DBA
記錄 Amazon Redshift 叢集詳細資訊。	開啟 Amazon Redshift 主控台，然後在導覽窗格中選擇「叢集」。然後從清單中選擇 Amazon Redshift 群集名稱。在「屬性」選項卡的「數據庫配置」部分中，記錄「數據庫名稱」和「端口」。打開「連接詳細信息」部分並記錄「端點」，這是「端點：<port>/<databasename>」格式。重要事項：確保針對輸入規則的 Amazon Redshift 安全群組允許從 EC2 執行個體不受限制地存取 TCP 連接埠 5439。	DBA

任務	描述	所需技能
將 psql Connect 到 Amazon Redshift 集群。	<databasename><port>在命令提示字元中，執行「psql-h-<endpoint>U-<userid>d-p」命令來指定連線資訊。在 psql 密碼提示下，輸入「<userid>」使用者的密碼。然後，您就會連線到 Amazon Redshift 叢集，並可以互動方式輸入命令。	DBA
建立 S3 儲存貯體。	開啟 Amazon S3 主控台，然後建立 S3 儲存貯體來存放從 Amazon Redshift 匯出的檔案。如需如何建立 S3 儲存貯體的指示，請參閱「相關資源」一節。	DBA，AWS 一般資訊
建立支援卸載資料的 IAM 政策。	開啟 AWS Identity and Access Management (IAM) 主控台，然後選擇「政策」。選擇「創建策略」，然後選擇「JSON」選項卡。從「其他資訊」部分複製並貼上用於卸載資料的 IAM 政策。重要事項：將「s3_bucket_name」取代為您的 S3 儲存貯體的名稱。選擇 [檢閱原則]，然後輸入原則的名稱和說明。選擇「創建策略」。	DBA

任務	描述	所需技能
建立 IAM 角色以允許 Amazon Redshift 執行卸載操作。	開啟 IAM 主控台，然後選擇「角色」。選擇「建立角色」，然後在「選取受信任實體類型」中選擇「AWS 服務」。為服務選擇「Redshift」，選擇「Redshift-可自定義」，然後選擇「下一步」。選擇您之前創建的「卸載」策略，然後選擇「下一步」。輸入「角色名稱」，然後選擇「建立角色」。	DBA
將 IAM 角色與 Amazon Redshift 叢集建立關聯。	開啟 Amazon Redshift 主控台，然後選擇「管理 IAM 角色」。從下拉菜單中選擇「可用角色」，然後選擇您之前創建的角色。選擇「應用更改」。當「管理 IAM 角色」上 IAM 角色的「狀態」顯示為「同步」時，您可以執行 UNLOAD 命令。	DBA
停止對 Amazon Redshift 叢集的寫入操作。	您必須記住停止對來源 Amazon Redshift 叢集的所有寫入操作，直到遷移完成為止。	DBA

準備目標區域中的移轉

任務	描述	所需技能
在目標區域中啟動和設定 EC2 執行個體。	登入中國北京或寧夏區域的 AWS 管理主控台。從 Amazon EC2 主控台選擇「啟動執行個體」，然後建立和設定 EC2	DBA

任務	描述	所需技能
	執行個體。重要事項：請確保 Amazon EC2 輸入規則的安全群組允許從來源機器不受限制地存取 TCP 連接埠 22。如需如何啟動和設定 EC2 執行個體的進一步指示，請參閱「相關資源」一節。	
記錄 Amazon Redshift 叢集詳細資訊。	開啟 Amazon Redshift 主控台，然後在導覽窗格中選擇「叢集」。然後從清單中選擇 Amazon Redshift 群集名稱。在「屬性」選項卡的「數據庫配置」部分中，記錄「數據庫名稱」和「端口」。打開「連接詳細信息」部分並記錄「端點」，這是「端點：<port>/<databasename>」格式。重要事項：請確定傳入規則的 Amazon Redshift 安全群組允許從 EC2 執行個體不受限制地存取 TCP 連接埠 5439。	DBA
將 psql Connect 到 Amazon Redshift 集群。	<databasename><port>在命令提示字元中，執行「psql-h-<endpoint>U-<userid>d-p」命令來指定連線資訊。在 psql 密碼提示下，輸入「<userid>」使用者的密碼。然後，您就會連線到 Amazon Redshift 叢集，並可以互動方式輸入命令。	DBA

任務	描述	所需技能
建立 S3 儲存貯體。	開啟 Amazon S3 主控台，然後建立 S3 儲存貯體以保存從 Amazon Redshift 匯出的檔案。如需此和其他故事的說明，請參閱「相關資源」一節。	DBA
建立支援複製資料的 IAM 政策。	開啟 IAM 主控台，然後選擇「政策」。選擇「創建策略」，然後選擇「JSON」選項卡。從「其他資訊」區段複製並貼上用於複製資料的 IAM 政策。重要事項：將「s3_bucket_name」取代為您的 S3 儲存貯體的名稱。選擇 [檢閱策略]，輸入策略的名稱和說明。選擇「創建策略」。	DBA
建立 IAM 角色以允許 Amazon Redshift 執行複製操作。	開啟 IAM 主控台，然後選擇「角色」。選擇「建立角色」，然後在「選取受信任實體類型」中選擇「AWS 服務」。為服務選擇「Redshift」，選擇「Redshift-可自定義」，然後選擇「下一步」。選擇您之前創建的「複製」策略，然後選擇「下一步」。輸入「角色名稱」，然後選擇「建立角色」。	DBA

任務	描述	所需技能
將 IAM 角色與 Amazon Redshift 叢集建立關聯。	開啟 Amazon Redshift 主控台，然後選擇「管理 IAM 角色」。從下拉菜單中選擇「可用角色」，然後選擇您之前創建的角色。選擇「應用更改」。當「管理 IAM 角色」上 IAM 角色的「狀態」顯示為「同步」時，您可以執行「COPY」命令。	DBA

在開始移轉之前驗證來源資料和物件資訊

任務	描述	所需技能
驗證來源 Amazon Redshift 資料表中的資料列。	使用「其他資訊」區段中的指令碼來驗證和記錄來源 Amazon Redshift 表格中的列數。請記住要平均分割卸載和 COPY 腳本的數據。這將提高數據卸載和加載效率，因為每個腳本覆蓋的數據量將是平衡的。	DBA
驗證來源 Amazon Redshift 叢集中的資料庫物件數目。	使用「其他資訊」區段中的指令碼來驗證和記錄來源 Amazon Redshift 叢集中的資料庫、使用者、結構描述、表格、檢視和使用者定義函數 (UDF) 的數量。	DBA
在移轉之前驗證 SQL 陳述式結果。	一些用於數據驗證的 SQL 語句應根據實際的業務和數據情況進行排序。這是為了驗證導入	DBA

任務	描述	所需技能
	的數據，以確保其一致並正確顯示。	

將資料和物件移轉至目標區域

任務	描述	所需技能
產生 Amazon Redshift DDL 指令碼。	使用「其他資訊」一節中「查詢 Amazon Redshift 的 SQL 陳述式」區段中的連結，產生資料定義語言 (DDL) 指令碼。這些 DDL 指令碼應包括「建立使用者」、「建立結構描述」、「使用者的結構描述權限」、「建立資料表/檢視」、「使用者的物件權限」和「建立函數」查詢。	DBA
在 Amazon Redshift 叢集中為目標區域建立物件。	在中國的 AWS 區域使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 執行 DDL 指令碼。這些指令碼會在目標區域的 Amazon Redshift 叢集中建立物件。	DBA
將來源 Amazon Redshift 叢集資料卸載到 S3 儲存貯體。	執行 UNLOAD 命令，將來源區域中的 Amazon Redshift 叢集中的資料卸載到 S3 儲存貯體。	DBA, 開發人員
將來源區域 S3 儲存貯體資料傳輸到目標區域 S3 儲存貯體。	將資料從來源區域 S3 儲存貯體傳輸到目標 S3 儲存貯體。由於無法使用「$ aws s3 sync」命令，請確保您使用「相關資源」一節中「將	開發人員

任務	描述	所需技能
	Amazon S3 資料從 AWS 區域傳輸到中國的 AWS 區域」一文中概述的程序。	
將資料載入目標 Amazon Redshift 叢集。	在目標區域的 psql 工具中，執行 COPY 命令，將資料從 S3 儲存貯體載入目標 Amazon Redshift 叢集。	DBA

在移轉後驗證來源和目標區域中的資料

任務	描述	所需技能
驗證並比較來源和目標資料表中的資料列數目。	驗證並比較來源和目標區域中的表格資料列數，以確保所有資料列都已移轉。	DBA
驗證並比較來源和目標資料庫物件的數目。	驗證並比較來源和目標區域中的所有資料庫物件，以確保所有資料庫物件都已移轉。	DBA
驗證並比較來源和目標區域中的 SQL 命令檔結果。	執行移轉前準備好的 SQL 指令碼。驗證並比較資料，以確保 SQL 結果正確無誤。	DBA
重設目標 Amazon Redshift 叢集中所有使用者的密碼。	遷移完成並驗證所有資料後，您應該重設中國 AWS 區域中 Amazon Redshift 叢集的所有使用者密碼。	DBA

相關資源

- 將 Amazon S3 資料從 AWS 區域傳輸到中國的 AWS 區域
- 建立 S3 儲存貯體
- 重置亞 Amazon Redshift 用戶密碼

- psql 文件

其他資訊

卸載資料的 IAM 政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

複製資料的 IAM 政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::s3_bucket_name"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::s3_bucket_name/*"]
    }
  ]
}
```

用於查詢 Amazon Redshift 的 SQL 語句

```
##Database

select * from pg_database where datdba>1;

##User

select * from pg_user where usesysid>1;

##Schema

SELECT n.nspname AS "Name",

  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"

FROM pg_catalog.pg_namespace n

WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'

ORDER BY 1;

##Table

select count(*) from pg_tables where schemaname not in
 ('pg_catalog','information_schema');

select schemaname,count(*) from pg_tables where schemaname not in
 ('pg_catalog','information_schema') group by schemaname order by 1;

##View

SELECT

    n.nspname AS schemaname,c.relname AS
 viewname,pg_catalog.pg_get_userbyid(c.relowner) as "Owner"

FROM

    pg_catalog.pg_class AS c

INNER JOIN

    pg_catalog.pg_namespace AS n
```

```
    ON c.relnamespace = n.oid

WHERE relkind = 'v' and n.nspname not in ('information_schema','pg_catalog');

##UDF

SELECT

   n.nspname AS schemaname,

   p.proname AS proname,

   pg_catalog.pg_get_userbyid(p.proowner) as "Owner"

FROM pg_proc p

LEFT JOIN pg_namespace n on n.oid = p.pronamespace

WHERE p.proowner != 1;
```

用來產生 DDL 陳述式的 SQL 指令碼

- 由用戶獲取模式私有腳本
- 產生 DDL 指令碼
- 生成 _ 視圖 _ DDL
- 產生使用者授權撤銷 _DDL
- 產生 _UDF _ DDL

使用 VMware HCX 將工作負載遷移到 AWS 上的 VMware 雲端

由迪帕克庫馬爾（AWS），德里克·考克斯（AWS）和希曼舒古普塔（AWS）創建

環境：生產	來源：內部部署 VMware 工作負	目標：VMware Cloud on AWS
R 類型：搬遷	工作負載：所有其他工作	技術：移轉；混合雲
AWS 服務：VMware Cloud on AWS；Amazon VPC		

Summary

注意事項：自 2024 年 4 月 30 日起，VMware 雲端服務 AWS 已不再由 AWS 或其通路合作夥伴轉售。該服務將繼續通過博通提供。我們鼓勵您與您的 AWS 代表聯繫以獲取詳細信息。

此模式說明如何使用 VMware 混合雲擴充功能 (HCX) 將工作負載從現場部署 VMware 環境移轉至 AWS 上的 VMware 雲端，而不需變更基礎平台。VMware HCX 可簡化移轉作業、協助重新平衡工作負載、協助保護資料，以及最佳化內部部署資料中心與雲端伺服器的災難回復程序。該模式討論了安裝，配置，升級和卸載 HCX 的步驟。

恒生國際假期支援以下項目：

- 舊版 VMware vSphere — HCX 可協助您將虛擬機器 (VM) 從舊版的 vSphere 移轉至 VMware Cloud on AWS。主機會自動更新和修復，以消除耗時的更新，以準備移轉。
- 大量移轉 — 您可以將 HCX 與 WAN 最佳化服務搭配使用，只要一個步驟即可移轉大量 VM，而不會停機，將內部部署網路擴展到雲端。
- 異質網路環境 — 您目前的網路 (例如 vSphere、NSX、VXLAN 或 NSX-T) 會決定移轉的複雜性。HCX 會擷取您網路應用程式的基礎知識，並將您目前的網路延伸至雲端，而不需要任何複雜的程序。
- 網路速度較慢 — 移轉通常需要連線速度超過 250 Mbps。HCX 可以以更低的速度遷移您的工作負載，大約 100 Mbps。

HCX 支援三種類型的雲端移轉：

- 混合性 (資料中心延伸模組) — 將現有的現場部署 VMware 軟體定義資料中心 (SDDC) 延伸至 AWS，以提供覆蓋區擴充、隨需容量、測試/開發環境和虛擬桌面。
- 雲端疏散 (整個資料中心的基礎設施重新整理) — 整合資料中心並完全移轉至 AWS 雲端 (包括處理資料中心主機代管或租用結束)。
- 應用程式特定遷移 — 將個別應用程式移至 AWS 雲端以滿足特定業務需求。

您可以使用 HCX 在現場部署環境和 AWS 上的 VMware 雲端之間雙向遷移工作負載。HCX 提供多種方式在來源和目標位置之間移轉工作負載：

- HCX 冷移轉可移轉離線的虛擬機器。此方法適用於已關閉電源的虛擬機器，因為它需要大量停機時間。
- HCX vMotion 使用 VMware vMotion 通訊協定來移動虛擬機器。HCX vMotion 提供零停機時間遷移，但一次只能遷移一台虛擬機器。
- HCX 大量移轉使用 VMware vSphere 複寫通訊協定將虛擬機器移至目的地。您可以 parallel 移轉多個虛擬機器，並排程轉換。停機時間等同於伺服器重新開機，而且所有 VM 的切換都是 parallel 進行的。
- HCX 複寫輔助 vMotion (RAV) 是 HCX 批量遷移和 HCX vMotion 的組合。它提供 parallel 遷移，調度和零停機時間。
- 當您在內部部署使用多個虛擬機器管理程式和非 vSphere 虛擬機器時，HCX 作業系統協助移轉可協助您大量移轉多個虛擬機器。當您使用 HCX 作業系統協助遷移從現場部署遷移至 VMware Cloud on AWS 時，即可免費使用 HCX 作業系統協助遷移，但如果您想要在兩個現場部署環境之間遷移，或從內部部署遷移至其他雲端供應商，則需要額外授權

先決條件和限制

先決條件

- 用於存取 VMware 主控台的 VMware 帳戶。
  - HCX 需要下列防火牆連接埠。

來源	目的地	連線埠
HCX 經理和內部設備的 IP	HCX 管理員和應用裝置在 VMware Cloud on AWS 的 IP	UDP 500、UDP 4500 和 ICMP

HCX 經理和內部設備的 IP	连接网络微软件 | 混合动力软件下载	TCP 443
HCX 經理和內部設備的 IP	HCX 雲端網址	TCP 443

如果內部部署網路具有內部防火牆，則必須在資料中心內允許本機更多連接埠。如需 HCX 的連接埠需求完整清單，請參閱 VMware HCX 說明文件。

- 若要設定 HCX，您需要網域名稱系統 (DNS) IP、vCenter 完整網域名稱 (FQDN)、NTP 伺服器 FQDN、單一登入 (SSO) 使用者以及類似資訊。事先收集這些詳細資料，以避免部署發生任何延遲。

限制

您可以使用網路延伸設備，在現場部署環境和 AWS 上的 VMware Cloud 之間擴充最多八個網路。如需 HCX 服務限制的完整清單，請參閱 VMware HCX 說明文件。

架構

源, 技術, 堆棧

- 內部部署 VMWARE

目標技術堆疊

- VMware Cloud on AWS

工具

工具

- VMware Cloud on AWS 服務是由 AWS 和 VMware 共同設計的服務，可協助您將現場部署 VMware 虛擬主機環境遷移並擴展到 AWS 雲端。
- VMware 混合雲擴充功能 (HCX) 是 VMware 公用程式，可將工作負載從現場部署 VMware 環境移轉至 VMware Cloud on AWS，而無需變更基礎平台。

史诗

部署恒生校驗

任務	描述	所需技能
在 VMware Cloud on AWS 中啟用 HCX 服務	1. 登入 AWS 主控台的 VMware 雲端。 2. 瀏覽至您的 SDCC，然後選擇 [檢視詳細資料]。 3. 選擇「附加組件」標籤。 4. 選擇「開啟 HCX」。 5. 選擇部署 HCX 並確認。HCX 部署即將開始。	雲端管理員、系統管理員
產生 HCX 啟動金鑰。	1. 在 AWS 主控台上使用 VMware 雲端。 2. 瀏覽至您的 SDCC，然後選擇 [檢視詳細資料]。 3. 選擇「附加組件」標籤。 4. 選擇 [開啟 HCX]，然後選擇 [啟用金鑰]。 5. 選擇建立啟動金鑰並複製金鑰。	雲端管理員、系統管理員
在雲端軟體定義的資料中心上新增 HDCX 的防火牆規則。	部署 HCX Manager 之後，您需要設定防火牆規則，以啟用內部部署環境與軟體定義的資料中心之間的通訊。您需要建立兩個防火牆規則：一個用於輸入，另一個用於輸出通訊。 1. 在 AWS 主控台上的 VMware 雲端上，選取您的軟體定義的資料中心，然後瀏覽至「聯網與安全」。	雲端管理員、系統管理員

任務	描述	所需技能
	2. 選擇 [閘道防火牆]，然後選擇 [管理閘道] 索引標籤。 3. 選擇新增規則並建立輸出規則： a. 提供規則名稱。 b. 編輯來源並選取 HCX。 c. 編輯目的地，並提供可存取 HCX 的內部部署 IP 和子網路。 d. 對於「服務」，請選擇任何。 e. 針對 [動作] 選擇 [允許]。 f. 選擇 Publish (發佈)。 4. 選擇新增規則並建立輸入規則： a. 提供規則名稱。 b. 編輯來源並提供可存取 HCX 的內部部署 IP 和子網路。 c. 編輯目的地並選擇 HCX。 d. 對於「服務」，請選擇「安全殼層」、「HTTPS」、「TCP (9443)」和「ICMP」。 e. 針對 [動作] 選擇 [允許]。 f. 選擇 Publish (發佈)。	

任務	描述	所需技能
在內部部署安裝 HCX 管理員。	1. 登入雲端 vCenter，然後從功能表導覽至 HCX。 2. 在 HCX 儀表板上，選擇管理，系統更新。 3. 要求 VMware HCX 連接器的下載連結，然後下載內部部署 OVA 檔案。 4. 登入您的內部部署 vCenter，並使用下載的 OVA 檔案部署 OVF 範本。 5. 在範本部署期間，請在出現提示時提供靜態 IP、NTP、DNS、DNS 搜尋清單及其他詳細資料。 6. 驗證所有詳細資料以完成 HCX 管理員部署。	雲端管理員、系統管理員

任務	描述	所需技能
在內部部署配置 HCX 經理。	1. 在瀏覽器中打開 HCX 管理器：`https://<HCX_Manager_IP>:9433` 。 2. 使用部署期間提供的使用者名稱和密碼登入。 3. 輸入您之前創建的激活密鑰，然後選擇激活以激活 HCX 實例。 4. 選擇「確認」以進行下一個步驟。 5. 選取內部部署資料中心的位置，然後選擇 [繼續]。 6. 在 [系統名稱] 中，輸入主機名稱，然後選擇 [繼續] 以完成啟動。 7. 輸入資訊以設定您的 vCenter 連線。 8. 輸入資訊以設定 SSO/PSC 詳細資料。 9. 選擇 [重新啟動] 讓變更生效。	雲端管理員、系統管理員

任務	描述	所需技能
設定網站配對。	在雲端和內部部署設定 HCX 之後，請依照下列步驟設定它們之間的站台配對。 1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 在左側導覽窗格中，選擇 [站台配對]，然後選擇 [Connect 至遠端站台]。 3. 在 [Connect 至遠端網站] 對話方塊中，新增 HCX 雲端 URL 和認證，然後選擇 [Connect]。 站台配對完成後，站台配對儀表板會顯示已連接的內部部署和雲端 SDDC。	雲端管理員、系統管理員

任務	描述	所需技能
建立網路設定檔。	網路設定檔是網路第 3 層元件的抽象化。此設定檔是建立運算設定檔的先決條件。 1. 登入您的雲端 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇 [互連]，選擇 [網路設定檔] 索引標籤，然後選擇 [建立網路設定檔]。 3. 設定網路設定檔： a. 選擇 vCenter 伺服器。 b. 選擇網路。 c. 新增設定檔的名稱。 d. 提供 IP 集區、首碼長度、閘道器、DND 和 MTU。 e. 選擇建立。 4. 依照相同的程序在內部部署建立網路設定檔。	雲端管理員、系統管理員

任務	描述	所需技能
建立運算設定檔。	運算設定檔包含 HCX 的網路、儲存和運算詳細資料。 HCX 在建立服務網格期間建立 HCX 設備時，會使用這些設定。 1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇 [互連]，選擇 [運算設定檔] 索引標籤，然後選擇 [建立運算設定檔]。 3. 指定運算設定檔的名稱。 4. 選取您要啟用的 HCX 服務，然後選擇「繼續」。 5. 選取服務資源。如果有多個叢集，請選取要為其啟動 HCX 服務的每個叢集，然後選擇 [繼續]。 6. 選取用於部署 HCX 應用裝置的運算和儲存資源，然後選擇 [繼續]。 7. 選取可用於連線 vCenter 和 ESXi 主機管理介面的管理網路設定檔，然後選擇繼續。 8. 選取可用於連線遠端站台上的互連設備的上行網路設定檔，以及遠端站台應用裝置可用於連線本機互連設備的上行網路設定檔，然後選擇 [繼續]。 9. 選取 vMotion 網路設定檔，然後選擇 [繼續]。	雲端管理員、系統管理員

任務	描述	所需技能
	10.選取 vSphere 複寫網路設定檔，然後選擇 [繼續]。 11.針對網路延伸選取適當的分散式交換器，然後選擇 [繼續]。 12.檢閱所有需要在 WAN 和 LAN 連線中開啟的連接埠，然後選擇 [繼續]。 13.若要建立運算設定檔，請選擇 [完成]。 14.依照相同的步驟，在雲端網站上建立運算設定檔。	

任務	描述	所需技能
建立服務網格。	服務網格為內部部署網站和雲端站台提供 HCX 服務組態。建立服務網格會啟動 HCX 互連虛擬應用裝置在兩個站台上的部署。必須在來源站台上建立互連服務。 1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇 [互連]，選擇 [服務網格] 索引標籤，然後選擇 [建立服務網格]。 3. 選取要在其中建立服務網格的來源和目的地站台，然後選擇 [繼續]。 4. 選取來源和您之前建立的目的地網站的計算設定檔，然後選擇 [繼續]。 5. 選擇您要啟用的 HCX 服務，然後選擇繼續。 6. 選取來源和目標站台的上行設定檔，然後選擇繼續。 7. 檢閱資源和網路，然後選擇 [繼續]。 8. 提供服務網格的名稱，然後選擇 [完成]。 服務網格部署將啟動。您可以在服務網格的 [工作] 索引標籤中追蹤進度。部署完成時，會顯示您為服務網格啟用的所有 HCX 服務的狀態。	雲端管理員、系統管理員

使用 HCX 擴充網路

任務	描述	所需技能
建立網路擴充功能。	您可以使用 HCX 網路延伸功能，在雲端 SDDC HCX 站台建立 L2 網路延伸模組，並橋接遠端和來源網路。 這可讓您將伺服器從現場部署遷移到 VMware Cloud on AWS，同時保留相同的 IP 位址。 1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇服務，網絡擴展。 3. 選擇「延伸網路」或「建立網路分機」。 4. 選取適當的服務網格、分散式連接埠群組或 NSX 邏輯交換器。 5. 提供閘道 IP 位址，然後選擇 [提交]。 當網路擴充功能完成時，系統會顯示擴充功能已完成。	雲端管理員、系統管理員

使用 HCX 設定備份工作

任務	描述	所需技能
設定複製。	若要使用 HCX 複寫虛擬機器：	雲端管理員、系統管理員

任務	描述	所需技能
	1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇 [移轉]，然後選擇 [移轉] 索引標籤。 3. 提供行動群組名稱，選取要移轉的虛擬機器，然後選擇 [新增]。 4. 選擇目標運算容器、儲存區資料夾、移轉類型 (冷、大量、RAV、vMotion) 和轉換排程。 5. 選擇 [驗證]，等待驗證完成，然後選擇執行以開始複製。	

升級 HCX

任務	描述	所需技能
檢閱建議和步驟。	大型移轉專案的持續時間可達六到八個月，有時更長，VMware 會定期發佈包含軟體修正、安全性更新和錯誤修正的 HCX 更新。我們建議您將 HCX 和設備保持在最新狀態，以消除任何安全漏洞並利用新功能。 注意：如果您目前的 HCX 版本落後於最新版本或更早版本的三個版本，則無法升級 HCX，並且必須重新部署。	雲端管理員、系統管理員

任務	描述	所需技能
	HCX 升級包括三個步驟： 1. 在內部部署和雲端備份 HCX 經理。 2. 在內部部署和雲端升級 HCX 管理員。 3. 升級內部部署和雲端中的服務網狀設備。 以下故事將更詳細地討論這些步驟。	

任務	描述	所需技能
備份 HCX 雲端管理員。	適用於 VMware 雲端的 HCX 雲端管理器由 VMware 管理，因此您無法拍攝快照。要備份 HCX Cloud Manager，您必須從 HCX 控制台下載備份，並使用此備份來恢復 HCX 配置，以防升級失敗或必須復原到上一個階段。 1. 登入 HCX 雲端管理員，請於。`https://<HCX_cloudmanager_ip_or_fqdn>:9433` 2. 導航到管理，故障排除，Backup 和還原。 3. 在「Backup」區段中，選擇「產生」以建立備份檔案。 4. 選擇下載資源保存備份文件。 HCX-IX、HCX-NE 和 HCX-WO 等 HCX 服務設備不需要個別備份。	雲端管理員、系統管理員

任務	描述	所需技能
在現場備份 HCX 經理。	您可以透過兩種方式在內部部署備份 HCX Manager：擷取虛擬機器快照或備份組態檔。 若要建立虛擬機器快照： 1. 登入您的內部部署 vCenter。 2. 前往虛擬機器和範本，然後導覽至 HCX 管理員虛擬機器。 3. 選擇「動作」、「快照」、「建立快照 若要備份組態檔案： 1. 登入 HCX 雲端管理員，請於。`https://<HCX_cloudmanager_ip_or_fqdn>:9433` 2. 導航到管理，故障排除，Backup 和還原。 3. 在「Backup」區段中，選擇「產生」以建立備份檔案。 4. 選擇下載資源保存備份文件。 HCX-IX、HCX-NE 和 HCX-WO 等 HCX 服務設備不需要個別備份。	雲端管理員、系統管理員

任務	描述	所需技能
在內部部署和雲端升級 HCX 管理員。	您必須先在內部部署升級 HCX 管理員，然後再升級 HCX 雲端管理員。 若要升級內部部署的 HCX 經理： 1. 登入 vCenter 並瀏覽至 HCX 儀表板。 2. 選擇系統，管理。 3. 在 [管理] 頁面上，選擇 [系統更新] 索引標籤。「可用的服務更新版本」欄顯示擱置的更新。 4. 選擇 [選取服務更新]、[下載] 以下載更新以供稍後升級使用，或選擇 [下載與升級] 以立即下載並部署更新。如果您選取 [下載]，請選擇 [升級]，並在準備就緒時確認啟動升級。 5. 升級完成時： • 在 HCX 管理員管理頁面上，驗證是否顯示最新的 HCX 版本。 • 在 HCX 儀表板上，檢查以確認站點配對為啟動。 • 選擇「基礎架構」、「服務網格」，並確認所有 HCX 服務正常運作。 按照相同的步驟升級 HCX 雲端管理程式。	雲端管理員、系統管理員

任務	描述	所需技能
升級服務網格設備。	服務網格獨立於來源站點的 HCX 經理進行更新。目標網站上的服務網格設備會自動更新。 若要在來源站台升級服務網格設備： 1. 登入 vCenter，然後瀏覽至 HCX 儀表板。 2. 選擇基礎結構，然後選擇服務網格索引標籤。 3. 如果您看到橫幅「服務網格設備的新版本可用。按一下「更新設備」以升級到最新版本，」選擇「更新設備」。 4. 在顯示設備的對話方塊中，選擇一個或多個設備，然後選擇確定以開始升級程序。(我們建議您更新所有服務網格設備。) 5. 選擇檢視每個服務網格的工作以監視升級。 6. 升級完成後，每個應用裝置和服務都會顯示一個橫幅，以確認成功完成。 7. 在升級後驗證通道狀態： • 選擇基礎結構、服務網格、檢視應用裝置。 • 通道狀態欄應顯示為啟動，且畫面不應指示設備的任何其他可用版本。	雲端管理員、系統管理員

刪除 HCX 網絡擴展

任務	描述	所需技能
取消延伸網路。	先前的步驟說明如何使用 HCX 網路延伸功能來建立 L2 網路延伸模組，並在從現場部署遷移到 AWS 上的 VMware 雲端期間保留現有 IP。當特定 VLAN 中的所有虛擬機器都移至 VMware Cloud on AWS 時，您必須取消延伸現場部署網站和雲端軟體定義的資料中心之間的網路，並在軟體定義的資料中心中使網路可路由。 我們建議您在將所有虛擬機器從內部部署遷移到 AWS 上的 VMware Cloud 之後，立即移除延伸網路，以避免延遲。 1. 登入您的內部部署 vCenter，然後瀏覽至 HCX 儀表板。 2. 在 HCX 儀表板上，選擇服務，網路延伸。 3. 選取您要取消擴充的網路，然後選擇 [取消延伸網路]。 4. 選取 [解除延伸後將雲端網路 Connect 至雲端邊緣閘道]。這將激活雲端上的網絡。	雲端管理員、系統管理員
在雲 SDDC 中路由移動的網絡。	1. 登入 VMC 入口網站。 2. 瀏覽至 SDCC，然後選擇 [檢視詳細資料]。	雲端管理員、系統管理員

任務	描述	所需技能
	3. 選擇 [網路與安全性] 索引標籤。 4. 在 [網路與安全性] 頁面上： • 選擇 [網路]、[區段]，並確認最近未延伸的子網路顯示為 [可路由]。 • 選擇 [詳細目錄]、[群組]，然後將該子網路新增至群組。 • 選擇 [安全性]、[分散式防火牆]，並確認群組是預定防火牆規則的一部分。	

解除安裝 HCX

任務	描述	所需技能
檢查先決條件。	在資料中心結束的情況下，建議您在遷移專案結束時解除安裝 HCX 並移除其元件。但是，如果您仍然保留內部部署足跡，則可能需要保持 HCX 運作。 解除安裝 HCX 之前，請確定： • 沒有作用中的移轉。 • 已移除所有網路擴充功能。	雲端管理員、系統管理員
在內部部署解除安裝 HCX。	1. 登入您的內部部署 vCenter 並導覽至 HCX 主控台。 2. 選擇 [服務]、[移轉]，並確認您沒有作用中的移轉。	雲端管理員、系統管理員

任務	描述	所需技能
	3. 選擇 [服務]、[網路延伸]，並確認沒有延伸網路。 4. 選擇基礎結構、站台配對、服務網格。 5. 識別服務網格，然後選擇 [刪除]。 6. 在確認提示中，再次選擇「刪除」。「移除服務網格」橫幅會出現在服務網格畫面上。 7. 對您擁有的任何其他服務網格重複步驟 5-6。 8. 若要移除站台配對，請選擇 [基礎結構]、[站台配對]，然後中斷所有配對的站台 9. 移除 HCX 管理員應用裝置： a. 登入您的內部部署 vCenter，然後導覽至 HCX 管理員應用裝置。 b. 選擇 [動作]、[電源]、[關機]。 c. 選擇「動作」，「從磁碟刪除」。	

任務	描述	所需技能
從內部部署 vCenter 伺服器取消註冊 HCX 外掛程式。	1. 登入至 vCenter 暴民使用者介面，位於`https://<vc_fqdn>/mob` 。 2. 在「內容」區段中，選擇「值」欄中的內容。 3. 在內容頁面上，選擇 ExtensionManager查看所有已註冊的外掛程式。 4. 請注意以`com.vmware.hybridity` 、`com.vmware.hcsp.alarm` 和開頭的副檔名`com.vmware.vca.marketing.ngc.ui` 。 5. 刪除擴展名： • 在「方法」區段中，選擇UnregisterExtension。 • 輸入步驟 4 中註明的延伸索引鍵，然後選擇叫用方法以移除擴充功能。 移除所有延伸模組後，HCX 外掛程式將從 vSphere 網頁用戶端中消失。	雲端管理員、系統管理員

任務	描述	所需技能
在雲端中解除安裝 HCX。	若要移除雲端中的 HCX 服務網格和站台配對，請重複先前在內部部署解除安裝 HCX 中所述的步驟。在 VMware Cloud on AWS 中，HCX 管理員是由 VMware 管理。您無法將其從 vCenter 刪除，但可以從 VMC 管理介面取消部署。 要取消部署「HCX 管理員」： 1. 登入 VMC 管理介面。 2. 選擇您的組織和軟體定義的資料中心。 3. 選擇「附加項目」以顯示所有已部署 HCX 的 SDDC。 4. 選擇「取消部署 HCX」。	雲端管理員、系統管理員

故障診斷

問題	解決方案
設定 HCX 大量遷移時，您無法選取要移轉的伺服器。	原因:這些伺服器的移轉已取消，但在清理期間未更新 HCX 資料庫。HCX 將資料庫移轉視為仍在進行中，因此已將狀態鎖定在「轉換進行中」。 解決方案：聯絡 VMware 支援團隊以清理 HCX 資料庫。
切換失敗，但可與「強制關閉電源」選項搭配使用。	原因:VMware Tools 版本不符合 HCX 批量遷移的先決條件，因此 HCX 無法關閉源虛擬機器。

問題	解決方案
	解決方案：將 VMware 工具更新為適用於您的移轉類型的建議版本。
進行移轉時，HCX 站台配對裝置升級失敗，並顯示錯誤「不允許進行大量移轉進行中的作業」。	原因:切換後 HCX 數據庫未更新。 解決方案：確定沒有進行中的移轉。升級站台配對應用裝置時，選擇強制升級。
切換失敗，並出現錯誤「低資源可用性」。	原因:主機虛擬機上的存儲空間不足。 解決方案：移轉前檢查儲存和運算資源。

相關資源

參考

- VMware Cloud on AWS 功能
- VMware Cloud on AWS 概觀和作業模式 (AWS Prescriptive Guidance 方針)
- 使用 VMware HCX (AWS 規範指引)，將 VMware 軟體定義的資料定義中心遷移至 AWS 上的 VMware 雲端
- VMware 雲端上的 VMware 總體驗 (VMware 說明文件)
- 恒生國際校驗集團版本資訊 (VMware 說明文件)
- AWS 上的軟體定義資料中心部署和最佳實務指南 (AWS 白皮書)

工具

- 使用 VMware 雲端技術區域 (VMware 雲端技術區) 在 AWS 上進行自動化

合作夥伴

- VMware Cloud on AWS 合作夥伴計劃

影片

- VMware Cloud on AWS (YouTube 影片)

使用傳輸在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫

由勞納克里沙巴（AWS）和吉泰爾庫馬爾（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for PostgreSQL
R 類型：搬遷	工作負載：開源	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式說明在兩個適用於 PostgreSQL 資料庫執行個體的 Amazon Relational Database Service (Amazon RDS) 之間移轉極大型資料庫的步驟，方法是使用 pg _port 延伸模組。此擴充套件提供實體的傳輸機制來移動每個資料庫。透過以最少的處理流式傳輸資料庫檔案，它提供了一種極快速的方法，在資料庫執行個體之間移轉大型資料庫，而且停機 此擴充功能使用提取模型，其中目標資料庫執行個體從來源資料庫執行個體匯入資料庫。

先決條件和限制

先決條件

- 兩個資料庫執行個體都必須執行相同的 PostgreSQL 主要版本。
- 資料庫不得存在於目標上。否則，傳輸會失敗。
- 除了 pg_ port 之外，不得在來源資料庫中啟用任何擴充功能。
- 所有來源資料庫物件都必須位於預設 pg_default 表格空間中。
- 來源資料庫執行個體的安全群組應允許來自目標資料庫執行個體的流量。
- 安裝 PostgreSQL 用戶端 (例如 psql)，或PgAdmin與 Amazon RDS PostgreSQL 資料庫執行個體搭配使用。您可以在本機系統中安裝用戶端，也可以使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。在這種模式中，我們在 EC2 實例上使用 psql。

限制

- 您無法在不同主要版本的 Amazon RDS 之間傳輸資料庫。

- 來源資料庫的存取權限和擁有權不會轉移至目標資料庫。
- 您無法在僅供讀取複本上或僅供讀取複本的父項執行個體上傳輸資料庫。
- 您不能在計劃使用此方法傳輸的任何資料庫資料表中使用 reg 資料類型。
- 您可以在資料庫執行個體上同時執行多達 32 個總傳輸 (包括匯入和匯出)。
- 您無法重新命名或包含/排除表格。一切都按原樣遷移。

小心

- 在移除擴充功能之前先進行備份，因為移除擴充功能也會移除相依物件以及對資料庫作業至關重要的一些資料。
- 當您決定 pg_ port 的工作程式數目和`work_mem`值時，請考慮在來源執行個體上其他資料庫上執行的執行個體類別和處理序。
- 傳輸開始時，來源資料庫上的所有連線都會結束，並將資料庫置於唯讀模式。

附註：傳輸在一個資料庫上執行時，不會影響相同伺服器上的其他資料庫。

產品版本

- Amazon RDS for PostgreSQL 10.10 及更新版本，以及適用於 PostgreSQL 11.5 及更新版本的亞馬遜 RDS。如需最新版本資訊，請參閱 Amazon RDS 說明文件中的在資料庫執行個體之間傳輸 PostgreSQL 資料庫。

架構

工具

- pg_port 提供了一個物理傳輸機制來移動每個數據庫。藉由以最少的處理方式串流資料庫檔案，實體傳輸會比傳統傾印和載入程序更快地移動資料，而且停機時間最少。PostgreSQL 可傳輸的資料庫使用提取模式，也就是目的地的資料庫執行個體從來源資料庫執行個體輸入資料庫。您可以在準備來源和目標環境時在資料庫執行個體上安裝此延伸模組，如此模式中所述。
- psql 可讓您連接至您的 PostgreSQL 資料庫執行個體，並使用這些執行個體。若要在您的系統上安裝 psql，請參閱下載頁面。

史诗

建立目標參數群組

任務	描述	所需技能
為目標系統建立參數群組。	指定可將其識別為目標參數群組的群組名稱；例如，pgtarget-param-group 如需指示，請參閱 Amazon RDS 文件。	DBA
修改參數群組的參數。	設定下列參數： 1. 加入pg_transport 至shared_preload_libraries 參數。 `shared_preload_libraries = pg_stat_statements, pg_transport` 2. 設定 pg_transport.num_workers 參數。選擇您要用來執行傳輸的工作者數目。您設定的值會決定將在來源中建立的transport.send_file Worker 數目。 3. 將的值增加max_worker_processes 到的值的三倍以上pg_transport.num_workers 。例如，如果您將值設定pg_transport.num_workers 為 4，	DBA

任務	描述	所需技能
	則max_worker_processes 值應至少為 13。如果失敗，pg_port 會建議使用最小值。 4. 設定pg_transport.timing 為 1。此設定可讓您在傳輸期間報告計時資訊。 5. 設定 pg_transport.work_mem 參數。此參數指定要配置給每個 Worker 的最大記憶體。預設值為 128 MB。 如需這些參數的詳細資訊，請參閱 Amazon RDS 文件。	

建立來源參數群組

任務	描述	所需技能
建立來源系統的參數群組。	指定可將其識別為來源參數群組的群組名稱；例如，pgsource-param-group 如需指示，請參閱 Amazon RDS 文件。	DBA
修改參數群組的參數。	設定下列參數： 1. 加入pg_transport 至shared_preload_libraries 參數。	DBA

任務	描述	所需技能
	`shared_preload_libraries = pg_stat_statements, pg_transport` 2. 設定 `pg_transport.num_workers` 參數。此參數在目標中定義的值會決定要使用的 `transport.send_file` Worker 數目。如果您有在此執行個體上執行的匯入，請增加此值，但請考慮已在執行的 Worker 數目。 3. `max_worker_processes` 將的值增加到目標上的值的三倍以`pg_transport.num_workers` 上。例如，如果您將目標上`pg_transport.num_workers` 的值設定為 4，則來源上的`max_worker_processes` 值應至少為 13。如果失敗，pg_port 會建議使用最小值。 4. 設定 `pg_transport.work_mem` 參數。此參數指定要配置給每個 Worker 的最大記憶體。預設值為 128 MB。 如需這些參數的詳細資訊，請參閱 Amazon RDS 文件。	

準備目標環境

任務	描述	所需技能
建立新的 Amazon RDS 資料庫執行個體，以將您的來源資料庫傳輸到該執行個體。	根據您的業務需求判斷執行個體類別和 PostgreSQL 版本。	DBA, 系統管理員, 資料庫架構師
修改目標的安全群組，以允許從 EC2 執行個體在資料庫執行個體連接埠上進行連線。	根據預設，PostgreSQL 執行個體的連接埠是 5432。如果您使用其他連接埠，則必須為 EC2 執行個體開啟與該連接埠的連線。	DBA, 系統管理員
修改例證，並指定新目標參數群組。	例如 `pgtarget-param-group` 。	DBA
重新啟動目標 Amazon RDS 資料庫執行個體。	參數`shared_preload_libraries` 和`max_worker_processes` 是靜態參數，需要重新啟動實例。	DBA, 系統管理員
使用 psql 從 EC2 實例 Connect 到數據庫。	使用 命令： `psql -h <rds_end_point> -p PORT -U username -d database -W`	DBA
建立傳輸延伸模組。	以具有該`rds_superuser` 角色的使用者身分執行下列查詢： `create extension pg_transport;`	DBA

準備來源環境

任務	描述	所需技能
修改來源的安全群組，以允許來自 Amazon EC2 執行個體和目標資料庫執行個體的資料庫執行個體連接埠	根據預設，PostgreSQL 執行個體的連接埠是 5432。如果您使用其他連接埠，則必須為 EC2 執行個體開啟與該連接埠的連線。	DBA, 系統管理員
修改例證並指定新來源參數群組。	例如 `pgsource-param-group` 。	DBA
重新啟動來源 Amazon RDS 資料庫執行個體。	參數`shared_preload_libraries` 和`max_worker_processes` 是靜態參數，需要重新啟動實例。	DBA
使用 psql 從 EC2 實例 Connect 到數據庫。	使用 命令： `psql -h <rds_end_point> -p PORT -U username -d database -W`	DBA
建立 pg_port 延伸模組，並從要傳輸的資料庫中移除所有其他擴充功能。	如果來源資料庫上安裝了 pg_port 以外的任何副檔名，傳輸將會失敗。此命令必須由具有該`rds_superuser` 角色的使用者執行。	DBA

執行運輸

任務	描述	所需技能
執行乾運行。	首先使用該`transport.import_from_server` 功能執行乾運行：	DBA

任務	描述	所需技能
	`SELECT transport .import_from_serve r( 'source-db-instanc e-endpoint', source- db-instance-port, 'source-db-instance- user', 'source-user- password', 'source- database-name', 'destination-user- password', 'true');` 此函數的最後一個參數（設置為`true`）定義了空運行。 此函數會顯示執行主要傳輸時會看到的任何錯誤。請先解決錯誤，然後再執行主要傳輸。	
如果無法執行成功，請初始化資料庫傳輸。	執行`transport.import_from_server` 函數以執行傳輸。它連接到源並導入數據。 `SELECT transport .import_from_serve r( 'source-db-instanc e-endpoint', source- db-instance-port, 'source-db-instance- user', 'source-user- password', 'source- database-name', 'destination-user- password', false);` 此函數的最後一個參數（設置為`false`）表示這不是乾運行。	DBA

任務	描述	所需技能
執行傳輸後的步驟。	資料庫傳輸完成之後： • 驗證目標環境中的資料。 • 將所有角色和權限新增至目標。 • 視需要啟用目標和來源中所有必要的擴充功能。 • 還原`max_worker_processes` 參數的值。	DBA

相關資源

- Amazon RDS 文件
- 運輸文件
- 使用 RDS PostgreSQL 可傳輸資料庫移轉資料庫 (部落格文章)
- PostgreSQL 載
- psql 實用程序
- 建立資料庫參數群組
- 修改資料庫參數群組中的參數
- PostgreSQL 載

平台重建

主題

- 設定 Oracle 資料庫與 Aurora 相容之間的連結
- 使用 AWS DMS 將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3
- SageMaker 使用 AWS 開發人員工具將 ML 建置、訓練和部署工作負載遷移到 Amazon
- 將 OpenText TeamSite 工作負載遷移到 AWS 雲端
- 將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列
- 透過資料庫連結使用直接的 Oracle 資料汲取匯入，將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon RDS
- 將 Oracle 電子商務套件遷移到 Amazon RDS 定制
- 將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制
- 將甲骨文功能遷移到 AWS 上的 PostgreSQL
- 將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫
- 將 Redis 工作負載遷移到 AWS 上的 Redis 企業雲端
- 使用 AWS SCT 和 AWS DMS 將亞馬 Amazon EC2 上的 SAP ASE 遷移到與 Amazon Aurora PostgreSQL 相容
- 使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器
- 將簡訊佇列從 Microsoft Azure 服務匯流排遷移到 Amazon SQS
- 使用 Oracle 資料泵和 AWS DMS 將甲骨文 JD 愛德華資料 EnterpriseOne 庫遷移到 AWS
- 使用 AWS DMS 將甲骨文 PeopleSoft 資料庫遷移到 AWS
- 將現場部署 MySQL 資料庫遷移到 Amazon RDS for MySQL
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server
- 使用複製將資料從 Microsoft Azure Blob 遷移到 Amazon S3
- 在 AWS 上從 Couchbase 伺服器遷移到卡佩拉
- 從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的阿帕奇 Tomcat
- 使用 Auto Scaling 能從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的 Apache Tomcat
- 將 .NET 應用程式從 Microsoft Azure 應用程式服務遷移到 AWS Elastic Beanstalk
- 將自我託管的 MongoDB 環境遷移到 AWS 雲端上的 MongoDB 地圖集
- 在 Amazon ECS 上從甲骨文遷移 WebLogic 到阿帕奇湯姆貓（TomEE）
- 使用 AWS DMS 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS

- 使用 Logstash 將現場部署 Oracle 資料庫遷移至 Amazon OpenSearch 服務
- 將現場部署甲骨文資料庫遷移到 Amazon RDS for Oracle
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到亞馬遜 RDS
- 使用合格邏輯從 Amazon EC2 上的 PostgreSQL 遷移到亞馬遜 RDS
- 將內部 PostgreSQL 料庫遷移至 Aurora
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到執行 Linux 的 Amazon EC2 上的 Microsoft SQL 伺服器
- 使用連結伺服器將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server 服器
- 使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將 Microsoft SQL 伺服器資料庫遷移到 Aurora MySQL
- 使用原生工具將現場部署 MariaDB 資料庫遷移到適用於 MariaDB 的 Amazon RDS
- 將內部部署 MySQL 資料庫遷移至 Aurora MySQL
- 使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL
- 使用 AWS 應用程式容器將現場部署 Java 應用程式遷移到 AWS
- 在 AWS 大型遷移中遷移共用檔案系統
- 使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS
- 更改 Python 和 Perl 應用程序以支持從 Microsoft SQL 服務器遷移到 Amazon Aurora PostgreSQL 兼容版本的數據庫

設定 Oracle 資料庫與 Aurora 相容之間的連結

由吉萬·謝蒂 (AWS)、巴努古迪瓦達 (AWS)、舒尚德希穆克 (AWS)、烏提亞古普塔 (AWS) 和維卡斯古普塔 (AWS) 所建立

環境：PoC 或試點	來源：甲骨文數據庫	目標：Aurora 郵政兼容
R 類型：重新平台	工作負載：甲骨文; 開源	技術：移轉；資料庫
AWS 服務：Amazon Aurora；Amazon EC2 Auto Scaling；Amazon Route 53		

Summary

在遷移到 Amazon Web Services (AWS) 雲端的過程中，您可以將應用程式現代化以使用雲端原生資料庫。從甲骨文數據庫遷移到 Amazon Aurora PostgreSQL 相容版本是邁向現代化的一個步驟。作為移轉的一部分，原生 Oracle 資料庫連結也需要轉換。

使用資料庫連結，一個資料庫可以存取另一個資料庫中的物件。從 Oracle 資料庫移轉至 Aurora PostgreSQL 相容之後，必須將 Oracle 資料庫伺服器與其他 Oracle 資料庫伺服器的資料庫連結轉換為 PostgreSQL 與 Oracle 資料庫連結。

此模式顯示如何設定從 Oracle 資料庫伺服器到 Aurora PostgreSQL 相容資料庫的資料庫連結。由於資料庫連結是單向的，所以模式也涵蓋了將資料庫連結從 PostgreSQL 資料庫轉換為 Oracle 資料庫。

從 Oracle 資料庫移轉並轉換至 Aurora PostgreSQL 相容資料庫之後，需要執行下列步驟才能設定資料庫之間的資料庫連結：

- 若要將「Oracle 資料庫」連結設定為來源，並將與 Aurora PostgreSQL 相容設定為目標的資料庫連結，必須設定「Oracle 資料庫閘道」，以便在異質資料庫之間進行通訊。
- 如果您將 Aurora PostgreSQL 相容版本 12.6 及更早版本之間的資料庫連結設定為來源資料庫，並將「Oracle 資料庫」設定為目標，則**`oracle_fdw`**擴充功能在本機上無法使用。相反地，您可以使用 Aurora PostgreSQL 相容資料庫中的`postgres_fdw`擴充功能，並`oracle_fdw`在亞馬遜彈性運算雲端 (Amazon EC2) 上建立的 PostgreSQL 資料庫中進行設定。此資料庫充當 Aurora PostgreSQL 相容資料庫和 Oracle 資料庫之間的中介。此模式包含兩個選項，可用來設定使用 Aurora PostgreSQL 12.6 及更早版本的資料庫連結：

- 使用 Amazon EC2 啟動指令碼在 Amazon EC2 Auto Scaling 群組中設定 EC2 執行個體，該指令碼會更新亞馬 Amazon Route 53 中的內部網域名稱系統 (DNS) 項目。
- 使用 Network Load Balancer 以取得高可用性 (HA)，在 Amazon EC2 Auto Scaling 群組中設定 EC2 執行個體。

如果您要在與 Aurora PostgreSQL 相容版本 12.7 及更新版本之間設定資料庫連結，則可以使用擴充功能。`oracle_fdw`

先決條件和限制

前提

- 虛擬私有雲 (VPC) 中的 Amazon Aurora PostgreSQL 相容資料庫
- 甲骨文和 Aurora 兼容數據庫之間的網絡連接

限制

- 目前，無法將資料庫連結設定為 Oracle 的 Amazon Relational Database Service 服務 (Amazon RDS) 做為來源資料庫，而 Aurora PostgreSQL 則與目標資料庫相容。

產品版本

- 甲骨文資料庫 11g 及更新版本
- Aurora 郵政兼容 11 及更高版本

架構

源, 技術, 堆棧

移轉之前，來源 Oracle 資料庫可以使用資料庫連結存取其他 Oracle 資料庫中的物件。這可以在內部部署或 AWS 雲端的 Oracle 資料庫之間原生運作。

目標技術堆疊

選項 1

- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon EC2 執行個體上的 PostgreSQL 資料庫

- Amazon EC2 Auto Scaling 群組
- Amazon Route 53
- Amazon Simple Notification Service (Amazon SNS)
- AWS Identity and Access Management (IAM)
- AWS Direct Connect

選項 2

- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon EC2 執行個體上的 PostgreSQL 資料庫
- Amazon EC2 Auto Scaling 群組
- Network Load Balancer
- Amazon SNS
- Direct Connect

選項三

- Amazon Aurora PostgreSQL-Compatible Edition
- Direct Connect

目標架構

選項 1

下圖顯示使用`oracle_fdw`和`postgres_fdw`擴充功能的資料庫連結設定，以及由 Amazon EC2 Auto Scaling 展群組和 Route 53 所提供的 HA。

1. 具有擴充功能的 Aurora PostgreSQL 相容執行個體會連`postgres_fdw`線至 Amazon EC2 上的 PostgreSQL 資料庫。
2. 具有`oracle_fdw`副檔名的 PostgreSQL 資料庫位於「Auto Scaling」群組中。
3. Amazon EC2 上的 PostgreSQL 資料庫使用直接 Connect 來連接到現場部署的 Oracle 資料庫。
4. Oracle 資料庫設定了 Oracle 資料庫閘道，以便從 Oracle 資料庫連線到 AWS 上的 PostgreSQL 資料庫。

5. IAM 授予 Amazon EC2 的許可，以更新 Route 53 記錄。
6. Amazon SNS 會針對自動擴展動作傳送警示。
7. Route 53 中設定的網域名稱會指向 PostgreSQL 亞馬遜 EC2 執行個體 IP 地址。

選項 2

下圖顯示使用oracle_fdw和postgres_fdw擴充功能 (由 Auto Scaling 群組和 Network Load Balancer 提供的 HA) 的資料庫連結設定。

1. 具有擴充功能的 Aurora PostgreSQL 相容執行個體會連postgres_fdw線到 Network Load Balancer。
2. Network Load Balancer 會將連線從 Aurora PostgreSQL 相容資料庫分配到 Amazon EC2 上的 PostgreSQL 資料庫。
3. 具有oracle_fdw副檔名的 PostgreSQL 資料庫位於「Auto Scaling」群組中。
4. Amazon EC2 上的 PostgreSQL 資料庫使用直接 Connect 來連接到現場部署的 Oracle 資料庫。
5. Oracle 資料庫設定了 Oracle 資料庫閘道，以便從 Oracle 資料庫連線到 AWS 上的 PostgreSQL 資料庫。
6. Amazon SNS 會針對自動擴展動作傳送警示。

選項三

下圖顯示使用 Aurora PostgreSQL 相容資料庫中的oracle_fdw擴充功能設定資料庫連結。

1. 具有oracle_fdw擴充功能的 Aurora PostgreSQL 相容執行個體會使用「直 Connect」來連線至 Oracle 資料庫。
2. 在 Oracle 伺服器上設定的「Oracle 資料庫閘道」可透過「直 Connect 線」與 Aurora PostgreSQL 相容的資料庫進行連線。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。在此模式中，選項 1 和 2 使用 EC2 執行個體來託管 PostgreSQL 資料庫。
- Amazon EC2 Auto Scaling 可協助您維持應用程式的可用性，並允許您根據定義的條件自動新增或移除 Amazon EC2 執行個體。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。此模式使用 Network Load Balancer。

其他服務

- 「Oracle 資料庫閘道」可讓 Oracle 資料庫存取非 Oracle 系統中的資料。

史诗

選項 1 與選項 2 的一般設定工作

任務	描述	所需技能
建立一個 EC2 執行個體，並設定 PostgreSQL 能。	1. 使用 Amazon Linux 2 作業系統建立 EC2 執行個體。 2. 若要安裝 PostgreSQL，請以 ec2 使用者身分登入 EC2 執行個體，然後執行下列命令。	雲端管理員，DBA

任務	描述	所需技能
	`` sudo su - root sudo tee /etc/yum. repos.d/pgdg.repo< <EOF [pgdg12] name=PostgreSQL 12 for RHEL/CentOS 7 - x86_64 baseurl=https://down load.postgresql.or g/pub/repos/yum/12/ redhat/rhel-7-x86_64 enabled=1 gpgcheck=0 EOF sudo yum install -y postgresql12-server sudo yum install postgresql12-devel sudo /usr/pgsql-12/ bin/postgresql-12- setup initdb sudo systemctl enable postgresql-12 sudo systemctl start postgresql-12 ```	
	3. 從下載`oracle_fdw` 源代碼 GitHub。 ``` mkdir -p /var/lib/ pgsql/oracle_fdw/ cd /var/lib/pgsql/ oracle_fdw/ wget https://g ithub.com/laurenz/ ```	

任務	描述	所需技能
	oracle_fdw/archive/refs/heads/master.zip unzip master.zip 4. 安裝 Oracle 即時用戶端並設定 Oracle 環境變數。 yum install https://download.oracle.com/otn_software/linux/instantclient/1912000/oracle-instantclient19.12-basic-19.12.0.0.0-1.x86_64.rpm yum install https://download.oracle.com/otn_software/linux/instantclient/1912000/oracle-instantclient19.12-devel-19.12.0.0.0-1.x86_64.rpm export ORACLE_HOME=/usr/lib/oracle/19.12/client64export LD_LIBRARY_PATH=/usr/lib/oracle/19.12/client64/lib:$LD_LIBRARY_PATH 5. 確保指的pg_config 是正確的版本。 which pg_config	

任務	描述	所需技能
	6. 編譯oracle_fdw 。 `cd /var/lib/pgsql/oracle_fdw/oracle_fdw-master` `make` `make install` 注意：如果您收到錯誤消失oci.h，請在中添加以下內容Makefile： • 至PG_CPPFLAGS ，新增 -I/usr/include/oracle/19.12/client64 • 至SHLIB_LINK ，新增 -L/usr/lib/oracle/19.12/client64/lib 如需詳細資訊，請參閱 Oracle _fdw 儲存庫。 7. 登入 PostgreSQL 資料庫並建立擴充功能。oracle_fdw `sudo su - postgres` `psql postgres` `create extension oracle_fdw;` 8. 建立將擁有外部資料表的 PostgreSQL 使用者。 `CREATE USER pguser WITH PASSWORD '<password>';`	

任務	描述	所需技能
	GRANT CONNECT ON DATABASE postgres TO pguser; 9. 建立外部資料包裝函式。以您的 Oracle 資料庫伺服器詳細資訊取代下列值： • <Oracle DB Server IP> • <Oracle DB Port> • <Oracle_SID> create server oradb foreign data wrapper oracle_fdw options (dbserver '//<Oracle DB Server IP>:<Oracle DB Port>/<Oracle_SID>'); GRANT USAGE ON FOREIGN SERVER oradb TO pguser; 10.若要建立使用者對應和對應至 Oracle 資料表的外部資料表，請以下列方式連線到 PostgreSQL 資料庫pguser，然後執行下列命令。請注意，在示例代碼中，DMS_SAMPLE 用作包含該NAME_DATA 表的 Oracle 模式，並且dms_sample 是其密碼。根據需要更換它們。 create user mapping for pguser server	

任務	描述	所需技能
	oradb options (user 'DMS_SAMPLE', password 'dms_sample'); 注意：下列範例會在 PostgreSQL 中為 Oracle 資料庫中的資料表建立外部資料表。必須為每個需要從 PostgreSQL 執行個體存取的 Oracle 資料表建立類似的外部資料表。 CREATE FOREIGN TABLE name_data(name_type CHARACTER VARYING(15) NOT NULL, name CHARACTER VARYING(45) NOT NULL) SERVER oradb OPTIONS (schema 'DMS_SAMPLE', table 'NAME_DATA'); select count(*) from name_data; 11.在 EC2 執行個體上設定 PostgreSQL 資料庫，以便在 PostgreSQL 資料庫啟動期間找到 Oracle 程式庫。這是oracle_fdw 擴展所需的。 sudo systemctl stop postgresql-12	

任務	描述	所需技能
	注意：編輯/usr/lib/systemd/system/postgresql-12.service 檔案以包含環境變數，以便systemctl 啟動可以找到所需的 Oracle 程式庫oracle_fdw 。 `# Oracle Environment Variables` `Environment=ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/db_1` `Environment=LD_LIBRARY_PATH=/u01/app/oracle/product/12.2.0.1/db_1/lib:/lib:/usr/lib` `sudo systemctl start postgresql-12`	

選項 1：設定具有 Oracle_fdw 和 postgres_fdw 副檔名、Auto Scaling 群組和 Route 53 的資料庫連結

任務	描述	所需技能
在 Amazon Route 53 中設置一個私有託管區域。	1. 在 Amazon Route 53 中創建一個私有託管區域。記下將與 EC2 執行個體相關聯的網域名稱。 2. 使用可解析為 EC2 執行個體 IP 地址 (包含 oracle_fdw PostgreSQL 擴充功能)	DBA、雲端系統管理員

任務	描述	所需技能
	的簡單路由政策新增「A」記錄。 3. 儲存「A」記錄後，請記下步驟 1 中網域名稱的託管區域 ID。這將用於建立適當的 IAM 政策。	

任務	描述	所需技能
建立將連接至 EC2 執行個體的 IAM 角色。	若要建立將附加至 EC2 執行個體的 IAM 角色，請使用下列政策。替換<Hosted zone ID>為上一個故事中捕獲的信息。 { "Version": "2012-10-17", "Statement": [{ "Sid": "VisualEditor0", "Effect": "Allow", "Action": "route53:ChangeResourceRecordSets", "Resource": "arn:aws:route53:: :hostedzone/<Hosted zone ID>" }, { "Sid": "VisualEditor1", "Effect": "Allow", "Action": "route53:ListHostedZones", "Resource": "*" }] }	雲端管理員，DBA

任務	描述	所需技能
建立 EC2 啟動範本。	1. 建立包含 `oracle_fdw` PostgreSQL 擴充功能的 EC2 執行個體的 AMI。 2. 使用 AMI 建立 EC2 啟動範本。 3. 若要允許從 Aurora PostgreSQL 相容執行個體連線到 EC2 執行個體上的 PostgreSQL 資料庫，請將您先前建立的 IAM 角色建立關聯，然後連接安全群組。 4. 在「使用者資料」區段中，加入下列指令、變`Domain Name`更`Hosted zone ID`和適當的值。然後選擇 [建立啟動範本]。 `#!/bin/bash` `v_zone_id='Hosted zone ID'` `v_domain_name='Domain Name'` `v_local_ipv4=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)` `aws route53 change-resource-record-sets --hosted-zone-id $v_zone_id --change-batch '{"Changes":[{"Action":"UPSERT","ResourceRecordSet":{"Name":"'$`	雲端管理員，DBA

任務	描述	所需技能
	v_domain_name'","Type":"A","TTL":10,"ResourceRecords":[{"Value":"'$v_local_ipv4'"}]}}]}'	

任務	描述	所需技能
設定「Auto Scaling」群組。	1. 若要設定「Auto Scaling」群組，請使用您在上一個步驟中建立的啟動範本。 2. 設定用於啟動 EC2 執行個體的適當 VPC 和子網路。選項 1 安裝程式不使用 Load Balancer。 3. 在 Scaling 政策下，將 [所需]、[最小] 和 [最大容量] 設定為 1。 4. 若要傳送警示給作業團隊，請新增事件的通知，例如「啟動」或「終止」。 5. 檢閱組態，然後選擇「建立 Auto Scaling」群組。 完成後，Auto Scaling 展組會啟動包含 `oracle_fdw` PostgreSQL 擴展的 EC2 實例，該擴展程序連接到 Oracle 數據庫。 注意：當您需要存取新的 Oracle 資料表或變更 Oracle 資料表的結構時，這些變更必須反映在 PostgreSQL 外部資料表中。實作變更之後，您必須建立 EC2 執行個體的新 AMI，並使用它來設定啟動範本。	雲端管理員，DBA

任務	描述	所需技能
在相容 Aurora 的執行個體中設定 Postgre_fdw 延伸功能。	1. `postgres_fdw` 在與 Aurora PostgreSQL 相容的執行個體中進行設定。這會連線到 Amazon EC2 上的 PostgreSQL 資料庫，該資料庫充當 Aurora PostgreSQL 相容執行個體和 Oracle 資料庫之間的中繼節點。 2. Connect 至與 Aurora PostgreSQL 相容的執行個體，並執行下列命令。 `create extension postgres_fdw;` `CREATE SERVER pgoradb FOREIGN DATA WRAPPER postgres_fdw OPTIONS (dbname 'postgres', host 'Domain Name', port '5432');` `CREATE USER MAPPING for postgres SERVER pgoradb OPTIONS (user 'pguser', password '<password>');` `CREATE FOREIGN TABLE data_mart.name_data( name_type CHARACTER VARYING(15) NOT NULL, name CHARACTER VARYING(45) NOT NULL ) SERVER pgoradb OPTIONS (schema_name`	雲端管理員，DBA

任務	描述	所需技能
	'public', table_name 'name_data'); select count(*) from data_mart.name_dat a; 這樣就完成了從 Aurora PostgreSQL 兼容到 Oracle 數據庫的數據庫鏈接的設置。 此解決方案提供災難復原 (DR) 策略，以防託管 PostgreSQL 資料庫的 EC2 執行個體發生故障。Auto Scaling 群組會啟動新的 EC2 執行個體，並使用新 EC2 執行個體的 IP 位址更新 DNS。如此可確保 Aurora PostgreSQL 相容執行個體中的外部資料表可以存取 Oracle 資料表，而無需手動介入。	

選項 2：設定具有 oracle_fdw 和 postgres_fdw 延伸模組、Auto Scaling 群組和 Network Load Balancer 的資料庫連結

任務	描述	所需技能
建立 EC2 啟動範本。	1. 建立包含 `oracle_fdw` PostgreSQL 擴充功能的 EC2 執行個體的 AMI。 2. 使用 AMI 建立 EC2 啟動範本。	雲端管理員，DBA

任務	描述	所需技能
設定目標群組、Network Load Balancer 和 Auto Scaling 群組。	1. 若要建立目標群組，請選擇「執行處理」作為目標類型。在 [通訊協定] 中選擇 [TCP]，然後針對 [連接埠] 選擇 5432。然後選擇要放置目標群組的 VPC，並選取適當的 Health 檢查。 2. 在 VPC 中建立內部 Network Load Balancer。將負載平衡器設定為監聽通訊協定:連接埠 TCP: 5432。將「預設」動作設定為「轉寄至」，然後選擇您建立的目標群組。 3. 使用您建立的啟動範本設定「Auto Scaling」群組。 4. 使用將用於啟動 EC2 執行個體的適當 VPC 和子網路來設定 Auto Scaling 群組。 5. 針對「負載平衡」選項，選擇「連結至現有的負載平衡器」，然後選取您建立的「目標群組」。對於 Health 態檢查，請選取 ELB。 6. 在 Scaling 政策下，將 [所需的容量] 和 [最小容量] 設定為 2，並視需要將 [最大容量] 設定為較高的數目以支援 HA 負載。 7. 若要傳送警示給作業團隊，請新增事件的通知，例如「啟動」或「終止」。	雲端管理員，DBA

任務	描述	所需技能
	8. 檢閱組態，然後選擇「建立 Auto Scaling」群組。 完成時，Auto Scaling 展組會啟動所需數量的 EC2 執行個體，其中包含連接到 Oracle 資料庫的 `oracle_fdw` PostgreSQL 擴充功能。 注意：當您需要存取新的 Oracle 資料表或變更 Oracle 資料表的結構時，這些變更必須反映在 PostgreSQL 外部資料表中。實作變更之後，您必須建立 EC2 執行個體的新 AMI，並使用它來設定啟動範本。	

任務	描述	所需技能
在相容 Aurora 的執行個體中設定 Postgre_fdw 延伸功能。	postgres_fdw 在與 Aurora PostgreSQL 相容的執行個體中進行設定。這會透過 Network Load Balancer 連線至 EC2 上的 PostgreSQL 資料庫。EC2 上的 PostgreSQL 執行個體充當 Aurora 與 PostgreSQL 相容執行個體和 Oracle 資料庫之間的中繼節點。 Connect 至與 Aurora PostgreSQL 相容的執行個體，並執行下列命令。 `create extension postgres_fdw; CREATE SERVER pgoradb FOREIGN DATA WRAPPER postgres_fdw OPTIONS (dbname 'postgres', host 'DNS name of Network Load Balancer', port '5432'); CREATE USER MAPPING for postgres SERVER pgoradb OPTIONS (user 'pguser', password '<password>'); CREATE FOREIGN TABLE data_mart.name_data(name_type CHARACTER VARYING(15) NOT NULL, name CHARACTER VARYING(45) NOT NULL`	雲端管理員，DBA

任務	描述	所需技能
	) SERVER pgoradb OPTIONS (schema_name 'public', table_name 'name_dat a'); select count(*) from data_mart.name_data; 這樣就完成了從 Aurora PostgreSQL 兼容到甲骨文數據庫的數據庫鏈接的設置。 如果 EC2 託管 PostgreSQL 資料庫發生故障，Network Load Balancer 會識別故障並停止傳送到失敗的 EC2 執行個體的流量。Auto Scaling 群組會啟動新的 EC2 執行個體，並向負載平衡器註冊該執行個體。這可確保在原始 EC2 執行個體發生故障後，Aurora PostgreSQL 相容執行個體中的外部表可以存取 Oracle 資料表，而無需手動介入。	

選項 3：在與 Aurora PostgreSQL 相容的資料庫中設定具有 Oracle_fdw 副檔名的資料庫連結

任務	描述	所需技能
在相容 Aurora 的執行個體中設定 Oracle_fdw 延伸模組。	對於 Aurora PostgreSQL 相容的資料庫版本 12.7 及更新版本，`oracle_fdw` 延伸功能可以原生使用。這樣就不需要在 EC2 執行個體上建立中繼 PostgreSQL 資料庫。與 Aurora PostgreSQL 相容的執	雲端管理員，DBA

任務	描述	所需技能
	行個體可以直接連線到 Oracle 資料庫。 1. 若要建立oracle_fdw 擴充功能，請登入與 Aurora PostgreSQL 相容的執行個體，然後執行下列命令。 `create extension oracle_fdw;` 2. 建立外部資料包裝函式。以您的 Oracle 資料庫伺服器詳細資訊取代下列值： • <Oracle DB Server IP> • <Oracle DB Port> • <Oracle_SID> `create server oradb foreign data wrapper oracle_fdw options (dbserver '//<Oracle DB Server IP>:<Oracle DB Port>/<Oracle_SID>');` 3. 若要建立使用者對應和對應至 Oracle 表格的外部表格，請執行下列命令。請注意，在示例代碼中，DMS_SAMPLE 用作包含該NAME_DATA 表的 Oracle 模式，並且dms_sample 是其密碼。根據需要更換它們。 此外，必須在與 Aurora PostgreSQL 相容的	

任務	描述	所需技能
	執行個體中建立外部資料表，才能存取其他 Oracle 資料表。 `create user mapping for postgres server oradb options (user 'DMS_SAMPLE', password 'dms_sample');` `CREATE FOREIGN TABLE name_data( name_type character varying(15) OPTIONS (key 'true') NOT NULL, name character varying(45) OPTIONS (key 'true') NOT NULL )SERVER oradb OPTIONS (schema 'DMS_SAMPLE', table 'NAME_DATA');` 必須為每個需要從 PostgreSQL 執行個體存取的 Oracle 資料表建立類似的外部資料表。	

設定 Oracle 資料庫閘道，以便從內部部署 Oracle 資料庫連線到 Aurora PostgreSQL 相容

任務	描述	所需技能
在內部部署 Oracle 資料庫伺服器中設定閘道。	1. 以 root 使用者身分，安裝最新的 UnixODBC 驅動程式管理員。	DBA

任務	描述	所需技能
	`sudo yum install unixODBC*`	
	2. 安 PostgreSQL 程式 ()。`psqlODBC` `sudo wget https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm` `sudo yum install pgdg-redhat-repo-latest.noarch.rpm` `sudo yum install postgresql12-odbc`	
	3. 為驅動程式建立 ODBC 資料來源名稱 (DSN)。 UnixODBC 驅動程式管理員提供`odbcinst`、和`isql`命令列公用程式`odbc_config` ，用於設定和測試驅動程式。使用`odbcinst`或`odbc_config` 實用程序，您可以找到 UnixODBC 驅動程序管理器文件以傳遞驅動程序信息以創建 DSN。 `odbcinst -j` 下面的代碼顯示示例輸出。	

任務	描述	所需技能
	<pre>unixODBC 2.3.1 DRIVERS..... ....: /etc/odbc inst.ini SYSTEM DATA SOURCES: /etc/odbc .ini FILE DATA SOURCES.. : /etc/ODBCDataSourc es USER DATA SOURCES.. : /root/.odbc.ini SQLULEN Size.......: 8 SQLLEN Size........: 8 SQLSETPOSIROW Size.: 8 odbc_config --odbcini --odbcinstini /etc/odbc.ini /etc/odbcinst.ini</pre> 從示例輸出中，您可以看到`odbcinst.ini` 和`odbc.ini`文件。基本上，`odbcinst.ini` 是 ODBC 驅動程序在環境中的註冊表和配置文件，同時`odbc.ini`是 ODBC DSN 的註冊表和配置文件。要啟用驅動程序，你需要修改這兩個文件。 4. 在 ODBC `psqlODBC` 驅動程式檔案中設定驅動程式庫`/etc/odbc`	

任務	描述	所需技能
	inst.ini ，並將以下幾行新增至檔案結尾。這些行為驅動程序創建一個條目。 [PostgreSQL] Description = ODBC for PostgreSQL Driver = /usr/lib/psqlodbcw.so Setup = /usr/lib/libodbcpsqlS.so Driver64 = /usr/lib64/psqlodbcw.so Setup64 = /usr/lib64/libodbcpsqlS.so FileUsage = 1 5. 在/etc/odbc.ini 檔案中建立 DSN。驅動程式管理員讀取此檔案，以判斷如何使用中指定的驅動程式詳細資訊連線至資料庫odbcinst.ini 。以實際值取代下列參數： • <PostgreSQL Port> • <PostgreSQL Database Name> • <Aurora PostgreSQL Endpoint> • <PostgreSQL username> • <PostgreSQL password>	

任務	描述	所需技能
	<pre>[pgdsn] Driver=/usr/pgsql-12/lib/psqlodbc.so Description=PostgreSQL ODBC Driver Database=<PostgreSQL Database Name> Servername=<Aurora PostgreSQL Endpoint> Username=<PostgreSQL username> Password=<PostgreSQL password> Port=<PostgreSQL Port> UseDeclareFetch=1 CommLog=/tmp/pgodbclink.log Debug=1 LowerCaseIdentifier=1</pre>	
	6. 使用此isql公用程式，測試您所建立之 PostgreSQL 資料庫 DSN 的 ODBC 連線 (psqlODBC)。 <pre>isql -v pgdsn</pre> 下面的代碼顯示示例輸出。 <pre>+------------------ ------------------ ----+ | Connected! | |</pre>	

任務	描述	所需技能
	<pre>| sql-statement | | help [tablename] | | quit | | | +---------- ---------- ------+ quit</pre>	
	7. 使用 DSN，建立 ODBC (HS) 服務處理常式的閘道。 作為用oracle戶，initDSN.ora 在位置創建一個文件$ORACLE_HOME/hs/admin 。在這種情況下，pgdsn是 DSN，所以你需要創建一個名為initpgdsn.ora 的文件。 <pre>more initpgdsn.ora</pre> 下面的代碼顯示示例輸出。 <pre># This is a sample agent init file that contains the HS parameters that are # needed for the Database Gateway for ODBC # # HS init parameters</pre>	

任務	描述	所需技能
	<pre># HS_FDS_CONNEC T_INFO=pgdsn HS_FDS_TRACE_L EVEL=OFF HS_FDS_TRACE_FILE_ NAME=/tmp/ora_hs_t race.log HS_FDS_SHAREABLE_N AME=/usr/lib64/lib odbc.so HS_NLS_NCHAR=UCS2 HS_LANGUAGE=AMERICA N_AMERICA.AL32UTF8 # # ODBC specific environment variables # set ODBCINI=/etc/ odbc.ini</pre>	
	8. 透過在SID_LIST_ LISTENER 中加入 DSN 項目來調整偵聽程式 ($ORACLE_HOME/netwo rk/admin/listener. ora)。 <pre>more $ORACLE_HOME/ network/admin/ listener.ora</pre> 下面的代碼顯示示例輸出。 <pre>SID_LIST_LISTENER =</pre>	

任務	描述	所需技能
	(SID_LIST = (SID_DESC= (SID_NAME = pgdsn) (ORACLE_HOME = /u01/app/oracle/product/12.2.0.1/db_1) (ENVS="LD_LIBRARY_PATH=/lib64:/usr/lib:/usr/lib64:/u01/app/oracle/product/12.2.0.1/db_1") (PROGRAM=dg4odbc))) 9. 透過新增 DSN 項目來調整 tnsname ($ORACLE_HOME/network/admin/tnsnames.ora)。 `more $ORACLE_HOME/network/admin/tnsnames.ora` 下面的代碼顯示示例輸出。 `pgdsn=(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SID=pgdsn))(HS=OK))` 10. 重新啟動 Oracle 監聽器，以便對網路檔案建立的 DSN 相關項目生效，並以	

任務	描述	所需技能
	適當<Listener Name>的 Oracle 監聽器名稱變更。 `lsnrctl stop <Listener Name>` `lsnrctl start <Listener Name>` 重新啟動甲骨文監聽器後，它將創建一個 DSN 名稱甲骨文 HS 處理程序（pgdsn）。 11.使用 DSN 建立 Oracle 資料庫連結，藉由登入「Oracle 資料庫」來存取 PostgreSQL 資料庫。 `create public database link pgdb connect to "postgres" identified by "postgres" using 'pgdsn';` 12.使用建立的 Oracle 資料庫連結存取 PostgreSQL 資料。 `select count(*) from "pg_tables"@pgdb;`	

相關資源

- Amazon Aurora PostgreSQL
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Identity and Access Management (IAM)

- 從啟動範本啟動執行個體
- Auto Scaling 群組
- Amazon Route 53
- Amazon Simple Notification Service (SNS)
- AWS Network Load Balancer
- Oracle 資料庫閘道

其他資訊

雖然此`oracle_fdw`擴充功能適用於與 Aurora PostgreSQL 相容的 12.7 版及更新版本，但此模式包含適用於舊版 Aurora PostgreSQL 相容資料庫的解決方案，因為許多客戶支援舊版 Aurora PostgreSQL 相容資料庫，而升級資料庫則涉及多個層級的應用程式和效能測試。此外，資料庫連結功能也廣泛使用，本文的目標是為所有版本的 Aurora PostgreSQL 相容版本提供選項。

使用 AWS DMS 將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3

由瑞典克里希納（AWS）創建

環境：PoC 或試點	來源：Microsoft SQL 服務器	目標：Amazon S3
R 類型：重新平台	工作量：Microsoft	技術：移轉；資料庫
AWS 服務：AWS DMS；Amazon S3		

Summary

Organizations 通常需要將資料庫複製到 Amazon Simple Storage Service (Amazon S3) 以進行資料庫遷移、備份和還原、資料存檔和資料分析。此模式說明如何將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3。來源資料庫可以託管在現場部署或 Amazon Elastic Compute Cloud (Amazon EC2) 或 Amazon Relational Database Service 服務 (Amazon RDS) 亞馬遜關聯式資 Microsoft 庫服務 (亞馬遜 RDS) Amazon Web Services (AWS) 雲端上。

使用 AWS Database Migration Service (AWS DMS) 匯出資料。根據預設，AWS DMS 會以逗號分隔值 (.csv) 格式寫入完整負載和變更資料擷取 (CDC) 資料。為了獲得更緊湊的存儲和更快的查詢選項，此模式使用 Apache 實木複合地板（.quet）格式選項。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 該帳戶的 AWS Identity and Access Management (IAM) 角色，可對目標 S3 儲存貯體進行寫入、刪除和標記存取權，以及作為受信任實體新增至此 IAM 角色的 AWS DMS (`dms.amazonaws.com`)
- 現場部署 Microsoft SQL 伺服器資料庫 (或 EC2 執行個體或亞馬遜 RDS SQL 伺服器資料庫上的 Microsoft SQL 伺服器)
- AWS 上的虛擬私有雲端 (VPC) 與 AWS Direct Connect 或虛擬私人網路 (VPN) 提供的現場部署網路之間的網路連線

限制

- 舊於 3.4.7 的 AWS DMS 版本目前不支援啟用虛擬私人雲端 (閘道虛擬私人雲端) S3 儲存貯體。
- 不支援在完全載入階段變更來源資料表結構。
- 不支援 AWS DMS 完整大型二進位物件 (LOB) 模式。

產品版本

- 適用於企業、標準、工作群組和開發人員版本的 Microsoft SQL Server 2005 或更新版本。
- 在 AWS DMS 版本 3.3.2 及更新版本中提供對 Microsoft SQL 伺服器 2019 版本作為來源的 Support 援。

架構

源, 技術, 堆棧

- 現場部署 Microsoft SQL 伺服器資料庫 (或 EC2 執行個體或亞馬遜 RDS SQL 伺服器資料庫上的 Microsoft SQL 伺服器)

目標技術堆疊

- AWS Direct Connect
- AWS DMS
- Amazon S3

目標架構

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

史诗

準備移轉

任務	描述	所需技能
驗證資料庫版本。	驗證來源資料庫版本，並確定 AWS DMS 支援該版本。如需有關支援 SQL 伺服器資料庫版本的詳細資訊，請參閱使用 Microsoft SQL 伺服器資料庫做為 AWS DMS 的來源。	DBA
建立 VPC 和安全性群組。	在您的 AWS 帳戶中，建立 VPC 和安全群組。如需詳細資訊，請參閱 Amazon VPC 文件。	系統管理員
為 AWS DMS 任務建立使用者。	在來源資料庫中建立 AWS DMS 使用者，並授予其讀取許可。AWS DMS 將會使用此使用者。	DBA
測試資料庫連線能力。	從 AWS DMS 使用者測試與 SQL 伺服器資料庫執行個體的連線。	DBA
建立 S3 儲存貯體。	建立目標 S3 儲存貯體。此儲存貯體會保留已移轉的資料表資料。	系統管理員
建立 IAM 政策和角色。	1. 若要建立具有值區許可的 IAM 政策，請使用其他資訊區段中的程式碼。 2. 為 AWS DMS 建立角色，並將政策附加到該角色。	系統管理員

使用 AWS DMS 遷移資料

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。	登入 AWS 管理主控台，然後開啟 AWS DMS 主控台。在瀏覽窗格中，選擇 [複製執行個體] > [建立複製執行個體 如需指示，請參閱 AWS DMS 文件中的步驟 1。	DBA
建立來源端點和目標端點。	建立來源端點和目標端點。測試從複製執行個體到來源端點和目標端點的連線。如需指示，請參閱 AWS DMS 文件中的步驟 2。	DBA
建立複製工作。	建立複寫任務，然後選取具有變更資料擷取 (CDC) 的全負載或全負載，將資料從 SQL Server 遷移到 S3 儲存貯體。如需指示，請參閱 AWS DMS 文件中的步驟 3。	DBA
開始資料複製。	啟動複寫工作，並監視記錄檔是否有任何錯誤。	DBA

驗證數據

任務	描述	所需技能
驗證移轉的資料。	在主控台上，導覽至目標 S3 儲存貯體。開啟與來源資料庫名稱相同的子資料夾。確認資料夾包含從來源資料庫移轉的所有表格。	DBA

清除資源

任務	描述	所需技能
關閉並刪除臨時 AWS 資源。	關閉您為資料遷移建立的臨時 AWS 資源 (例如 AWS DMS 複寫執行個體)，並在驗證匯出後將其刪除。	DBA

相關資源

- AWS Database Migration Service 使用者指南
- 使用 Microsoft SQL 伺服器資料庫作為 AWS DMS 的來源
- 使用 Amazon S3 作為 AWS Database Migration Service 的目標
- 使用 S3 儲存貯體做為 AWS DMS 目標 (AWS RE: 貼文)

其他資訊

使用下列程式碼為 AWS DMS 角色新增具有 S3 儲存貯體許可的 IAM 政策。用您的儲存貯體名稱取代 `bucketname`。

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "s3:PutObject",
                 "s3:DeleteObject"
             ],
             "Resource": [
                 "arn:aws:s3:::bucketname*"
             ]
         },
         {
             "Effect": "Allow",
             "Action": [
                 "s3:ListBucket"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::bucketname*"
            ]
        }
    ]
}
```

SageMaker 使用 AWS 開發人員工具將 ML 建置、訓練和部署工作負載遷移到 Amazon

創建者：蘇格蘭·馬文 (AWS)

R 類型：重新平台	資源：M achine Learning	目標：Amazon SageMaker
創建者：AWS	環境：PoC 或試點	技術：機器學習和人工智能 DevOps; 遷移
AWS 服務：Amazon SageMaker		

Summary

此模式提供指導，讓您遷移在 Unix 或 Linux 伺服器上執行的現場部署機器學習 (ML) 應用程式，以便使用 Amazon 在 AWS 上進行訓練和部署 SageMaker。此部署使用持續整合和持續部署 (CI/CD) 管線。遷移模式是使用 AWS CloudFormation 堆疊部署的。

先決條件和限制

先決條件

- 使用 AWS 登陸區域的有效 AWS 帳戶
- 在您的 Unix 或 Linux 伺服器上安裝和設定 AWS Command Line Interface (AWS CLI)
- AWS CodeCommit 或亞馬遜簡單儲存服務 (Amazon S3) 中的 ML 原始程式碼儲存庫 GitHub

限制

- 一個 AWS 區域只能部署 300 個個別管道。
- 此模式適用於使用 Python 程式 train-and-deploy 碼的受監督機器學習工作負載。

產品版本

- 碼頭工人版本 19.03.5, 建立 633a0ea, 使用 Python 3.6 倍

架構

源, 技術, 堆棧

- 內部部署 Linux 計算執行個體，其中包含本機檔案系統或關聯式資料庫中的資料

來源架構

目標技術堆疊

- 使用 Amazon S3 CodePipeline 部署的 AWS 用於資料儲存，而 Amazon DynamoDB 則做為中繼資料存放區，用於追蹤或記錄管道執行

目標架構

應用移轉架構

- 原生 Python 套件和 AWS CodeCommit 儲存庫 (以及 SQL 用戶端，適用於資料庫執行個體上的現場部署資料集)

工具

- Python
- Git
- AWS CLI — AWS CLI 會部署 AWS CloudFormation 堆疊，並將資料移至 S3 儲存貯體。S3 存儲桶，反過來，導致目標。

史诗

規劃移轉

任務	描述	所需技能
驗證原始程式碼和資料集。		資料科學家
識別目標建置、訓練和部署執行個體類型和大小。		資料工程師、資料科學家
建立能力清單和容量需求。		
識別網路需求。		DBA, 系統管理員
識別來源和目標應用程式的網路或主機存取安全性需求。		數據工程師，ML 工程師，系統管理員
決定備份策略。		ML 工程師，系統管理員
決定可用性需求。		ML 工程師，系統管理員
識別應用程式移轉或轉換策略。		資料科學家、ML 工程師

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		ML 工程師，系統管理員
建立安全性群組。		ML 工程師，系統管理員
針對 ML 程式碼設定 Amazon S3 儲存貯體和 AWS CodeCommit 儲存庫分支。		ML 工程師

上傳數據和代碼

任務	描述	所需技能
使用原生 MySQL 工具或第三方工具，將訓練、驗證和測試資料集遷移到佈建的 S3 儲存貯體。	這是 AWS CloudFormation 堆疊部署所必需的。	數據工程師，ML 工程師
將 ML 訓練和託管程式碼 Package 為 Python 套件，然後推送至 AWS CodeCommit 或中佈建的儲存庫 GitHub。	您需要存放庫的分支名稱，才能部署 AWS CloudFormation 範本以進行移轉。	資料科學家、ML 工程師

移轉應用程式

任務	描述	所需技能
遵循 ML 工作負載移轉策略。		應用程式擁有者、ML 工程
部署 AWS CloudFormation 堆疊。	使用 AWS CLI 建立在此解決方案隨附的 YAML 範本中宣告的堆疊。	資料科學家、ML 工程師

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		應用程式擁有者、資料科學家、ML 工程

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。	關閉 AWS CloudFormation 範本中的任何自訂資源 (例如，任何未使用的 AWS Lambda 函數)。	資料科學家、ML 工程師
審核並驗證專案文件。		應用程式擁有者、資料科
使用運算子驗證結果和 ML 模型評估量度。	請確定模型效能符合應用程式使用者的期望，且與內部部署狀態相當。	應用程式擁有者、資料科
關閉專案並提供意見反應。		應用程式擁有者、ML 工程

相關資源

- AWS CodePipeline
- AWS CodeBuild
- Amazon SageMaker
- Amazon Simple Storage Service (Amazon S3)
- Amazon DynamoDB
- AWS Lambda

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

將 OpenText TeamSite 工作負載遷移到 AWS 雲端

創建者：巴圖爾加·普雷夫拉查（AWS），邁克爾·斯圖爾特和卡洛斯·馬魯安達·莫利納

環境：生產	資料來源：內部部署	目標：AWS
R 類型：重新平台	工作負載：所有其他工作	技術：遷移；Web 和移動應用程序
AWS 服務：Amazon EC2; Amazon RDS		

Summary

警告：此案例需要具有程式設計存取權限和長期登入資料的 IAM 使用者，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。如有必要，可更新存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的更新存取金鑰。

許多 OpenText Experience Platform 執行個體都託管在內部部署或傳統託管解決方案上，具有固定容量和傳統成本模 將您的體 OpenText 驗平台工作負載遷移到 Amazon Web Services (AWS) 雲端，除了可以降低整體擁有成本之外，還可提高業務靈活性和整合機會，從而提供額外的功能和價值。

此模式提供將OpenText TeamSite工作負載遷移到 AWS 雲端的步驟和範本。此模式提供詳細的 Epics 區段，引導您完成移轉程序，協助您瞭解如何規劃 OpenText TeamSite 移轉專案的範圍和編列預算。

此模式由 AWS 和 AWS 合作夥伴 TBSCG 開發，並隨附 AWS 規範指導網站將工作負載移轉 OpenText TeamSite 和媒體管理工作負載至 AWS 雲端的指南。

先決條件和限制

先決條件

- 至少一個有效的 AWS 帳戶
- 託管於內部部署資料中心或其他雲端供應商的 OpenText 工作負載

- 作用中 OpenText 授權

移轉程序也需要下表所述的角色和職責。

Role	責任
贊助	內部贊助
送貨經理	遷移交付
方案架構師	定義當前和新的架構
DevOps 工程師	DevOps 活動
QA 測試儀	系統級測試
產品擁有者	根據業務需求排定任務優先順序
TeamSite 作者	移轉使用者驗收測試 (UAT)
TeamSite 管理員	移民 UAT
OpenText 鉛	OpenText 產品專家
OpenText 開發者	OpenText 產品專家
定價專家	AWS 和 OpenText 授權
IT 安全性	IT 安全性基準
第三方整合開發	重製現有的整合
前端開發者	變更已移轉的前端程式碼
資料庫管理員	資料庫組態

限制

- 確保與目標作業系統 (OS) 的相容性。您可以使用您要移轉之產品版本之產 OpenText 品版本說明中的相容性對照表。

架構

源, 技術, 堆棧

- OpenText 在內部部署或其他雲端供應商代管的客戶體驗解決方案：
  - OpenText TeamSite
  - OpenText LiveSite
  - OpenText 媒體管理
  - OpenText MediaBin

目標技術堆疊

- 在 AWS 雲端託管且使用下列 AWS 服務的 OpenText 客戶體驗平台：
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon Elastic Container Service (Amazon ECS)
  - Amazon OpenSearch 服務
  - Elastic Load Balancing
  - AWS Lambda
  - Amazon API Gateway
  - Amazon Relational Database Service (Amazon RDS)
  - Amazon Elastic Block Store (Amazon EBS)
  - Amazon Simple Storage Service (Amazon S3)

目標架構

工具

- AWS Database Migration Service (AWS DMS) 是一種雲端服務，可讓您輕鬆移轉關聯式資料庫、資料倉儲、NoSQL 資料庫和其他類型的資料存放區。
- AWS 應用程式遷移服務會自動轉換來源伺服器，以便在 AWS 上以原生方式執行。它還會透過內建和自訂的最佳化選項，簡化應用程式的現代化程序。

史诗

探索與評估

任務	描述	所需技能
舉辦有關發現要求的研討會。	與業務和技術團隊舉辦研討會，探索目前的情況、收集需求並驗證遷移策略。視您移轉的複雜性和範圍而定，您的組織可能需要數個研討會。 持續時間：兩週	贊助商（可選），交付經理，解決方案架構師，負 OpenText 責人，產品負責人
分析解決方案和移轉需求。	分析並記錄影響規劃解決方案和移轉程序設計的業務、功能和技術需求。 持續時間：一周	解決方案架構師， OpenText 鉛，產品負責人
記錄您現有的 OpenText 架構。	記錄您現有的 OpenText 架構，包括核心元件和所有相關的應用程式和服務。 持續時間：一周	解決方案架構師， OpenText 鉛，產品負責人
定義規劃的 AWS 架構。	根據識別的元件、需求並使用 OpenText 相容性矩陣來定義規劃的 AWS 架構。您可以在版本的 OpenText TeamSite 版本說明中找到 OpenText 相容性矩陣。 持續時間：一周	解決方案架構師, OpenText 鉛, 產品負責人, IT 安全
評估計劃 AWS 架構的大小。	根據工作負載和其他非功能性需求，不同架構元件的大小需求會有所不同。	解決方案架構師， OpenText 鉛

任務	描述	所需技能
	持續時間：兩天	
計算總體擁有成本。	計算您提議的解決方案的總持有成本 (TCO)。 持續時間：兩天	解決方案架構師、定價專
定義每個元件的移轉策略。	定義並記錄七種常見遷移策略 (7 Rs) 中哪一種，用於必須遷移到 AWS 雲端的每個核心或其他元件。 持續時間：一周	解決方案架構師， OpenText 鉛，產品負責人
定義元件的移轉程序。	定義每個工作負載元件的詳細移轉程序。 持續時間：一周	解決方案架構師, OpenText 鉛, 產品負責人, IT 安全
定義全域移轉程序和相依性。	建立全域移轉程序和行事曆，其中包含元件、相依性和業務持續性的移轉詳細資料。 持續時間：三天	解決方案架構師, OpenText 鉛, 產品負責人, IT 安全

安全性與合規性活動

任務	描述	所需技能
建立安全性原則。	在 AWS 帳戶中設定客戶受管安全政策。除了自動關閉未使用的帳戶之外，這些應包括密碼複雜性和輪換。 如需有關客戶受管政策的詳細資訊，請參閱 AWS Identity and Access Management	方案架構師

任務	描述	所需技能
	(IAM) 文件中的客戶受管政策。	
建立 IAM 使用者。	建立需要存取 AWS 管理主控台、AWS Command Line Interface (AWS CLI) (AWS CLI) 和 AWS 開發套件的 IAM 使用者。 有關建立 IAM 使用者的詳細資訊，請參閱 IAM 文件中的在 AWS 帳戶中建立 IAM 使用者。	方案架構師
建立 IAM 群組。	建立所需的 IAM 使用者群組 (例如管理員或開發人員群組)，並將 IAM 使用者新增至這些群組。 如需 IAM 使用者群組的詳細資訊，請參閱 IAM 說明文件中的 IAM 使用者群組。	方案架構師
附加安全性原則。	將安全政策附加到 IAM 群組或角色。 如需詳細資訊，請參閱 IAM 說明文件中的將政策附加到 IAM 使用者群組。	方案架構師
開啟詳細帳單功能。	如需帳單的詳細資訊，請參閱 AWS Billing and Cost Management 文件中的監控用量和成本。	方案架構師

任務	描述	所需技能
檢查您帳戶的聯絡詳細資料。	請確定您帳戶的連絡人詳細資料是最新的，並對應至組織中的多個人。 如需詳細資訊，請參閱 AWS Billing and Cost Management 文件中的管理 AWS 帳戶。	解決方案架構師、產品所
新增安全性連絡人資訊。	使用您的安全聯絡資訊設定您的聯絡資訊。 如需詳細資訊，請參閱 AWS Billing and Cost Management 文件中的管理 AWS 帳戶。	解決方案架構師、IT 安全
為 EC2 執行個體設定 IAM 角色。	設定 EC2 執行個體的 IAM 角色。 如需這方面的詳細資訊，請參閱 Amazon EC2 文件中的適用於 Amazon EC2 的 IAM 角色。	方案架構師
設定對 AWS Support 的存取權限。	將 IAM 政策附加到需要存取 AWS Support 中心並建立 Support 案例的 IAM 使用者。 如需詳細資訊，請參閱 AWS Support 文件中的 AWS Support 存取許可。	方案架構師

任務	描述	所需技能
啟用 CloudTrail。	CloudTrail 在所有 AWS 區域自動啟用 AWS。 如需這方面的詳細資訊，請參閱 AWS CloudTrail 文件`create-trail` 中的使用。	方案架構師
啟用 CloudTrail 記錄檔驗證。	啟用 CloudTrail 記錄檔的驗證。 如需詳細資訊，請參閱 AWS CloudTrail 文件 CloudTrail中的啟用記錄檔完整性驗證。	方案架構師
限制存取任何包含 CloudTrail 日誌的 S3 儲存貯體。	套用儲存貯體政策，限制存取包含 CloudTrail 日誌檔的 S3 儲存貯體。 如需有關這方面的詳細資訊，請參閱 AWS CloudTrail 文件 CloudTrail中的 Amazon S3 儲存貯體政策。	方案架構師
CloudTrail 與 CloudWatch 記錄檔整合	將由 Amazon CloudWatch 日誌產生 CloudTrail 的追蹤整合。 如需詳細資訊，請參閱 AWS CloudTrail 文件中的將事件傳送至 CloudWatch 日誌	方案架構師

任務	描述	所需技能
在所有必要區域中啟用 AWS Config。	在所有必要區域中自動啟用 AWS Config。 您可以使用 AWS CLI 設定 AWS 組態。如需詳細資訊，請參閱 AWS Config 文件中的使用 AWS CLI 設定 AWS Config。	方案架構師
啟用 S3 儲存貯體存取的記錄。	使用自動化 S3 儲存貯體存取記錄 CloudTrail。 如需此相關資訊，請參閱 Amazon S3 文件中的啟用 S3 儲存貯體和物件的 CloudTrail 事件記錄。	方案架構師
為 CloudTrail. 設定的 AWS KMS 金鑰政策	自動化的 AWS Key Management Service (AWS KMS) 金鑰政策的組態 CloudTrail。 如需有關這方面的詳細資訊，請參閱 AWS CloudTrail 文件 CloudTrail中的設定 AWS KMS 金鑰政策。	方案架構師
靜態加密 CloudTrail 日誌。	使用 AWS KMS 中保留的客戶受管金鑰設定伺服器端加密 CloudTrail 日誌。 如需詳細資訊，請參閱 AWS 文件中的使用 AWS KMS 受管金鑰 (SSE-KMS) 加密 CloudTrail 日誌檔案。CloudTrail	方案架構師

任務	描述	所需技能
自動旋轉 KMS 金鑰。	設定 AWS KMS 金鑰的輪替。 如需詳細資訊，請參閱 AWS KMS 文件中的如何啟用和停用自動金鑰輪替。	方案架構師
設定 CloudWatch 警報。	設定由特定事件啟動的 Amazon CloudWatch 警示。例如，對 API 的未經授權請求或根帳戶的使用。 如需詳細資訊，請參閱 AWS 安全部落格使用 AWS 帳戶的根存取金鑰時，如何接收通知。	方案架構師
設定安全群組。	設定安全群組，以確保連接埠 22 和 3389 上不允許不受限制的輸入流量。	方案架構師
開啟 VPC 流程記錄。	擷取虛擬私有雲 (VPC) 中網路介面的拒絕 IP 流量，並進行設定 CloudWatch 以擷取。 如需有關此項目的詳細資訊，請參閱 Amazon VPC 文件中的建立流程日誌。	方案架構師
修改預設安全性群組以限制所有流量。	修改每個 VPC 的預設安全群組，以便預設拒絕流量，並透過安全性群組明確授與存取權限。 如需有關此功能的詳細資訊，請參閱 Amazon VPC 說明文件中的 VPC 安全群組。	方案架構師

任務	描述	所需技能
設定 VPC 之間的路由表。	使用必要的最少存取權限設定 VPC 對等互連的路由表。 如需此相關資訊，請參閱 Amazon V PC 說明文件中的更新 VPC 對等連線的路由表。	方案架構師

新 AWS 基礎設施的設定活動

任務	描述	所需技能
佈建 AWS 基礎設施。	建立 AWS 帳戶和資源。 持續時間：兩週	DevOps 工程師, 方案架構師
設定 DevOps 工具和程序。	設定 DevOps 工具和程序，例如持續整合和持續交付 (CI/CD) 管線，以及自動化測試架構。	DevOps 工程師, 方案架構師
自動化核心元件的移轉。	使用現有範本或指令碼自動化 OpenText 產品的安裝和設定 TeamSite，包括 LiveSite、OpenDeploy 和 MediaBin。 持續時間：一周	DevOps 工程師, 解決方案架構師, OpenText 鉛
自動化其他元件的移轉。	分析並自動化與 OpenText 核心元件整合的其他應用程式 (例如，其他資料庫、通訊、監視或快取元件) 的移轉作業。 持續時間：兩週	DevOps 工程師, 解決方案架構師, OpenText 鉛
調整核心元件。	對 OpenText 核心元件的自訂進行任何必要的變更 (例如整合)。	解決方案架構師、 OpenText 潛在客戶、 OpenText 開發人員、第三方整合開發人員

任務	描述	所需技能
實作及設定其他服務。	佈建、設定和實作任何新的 AWS 服務，例如 AWS Lambda 函數或 Amazon API Gateway。	DevOps 工程師, 解決方案架構師, 協力廠商整合開發人員,
移轉或重構其他元件。	移轉其他元件，包括任何必要的重構。這包括外部應用程式，例如自訂報告入口網站或現有的 API 整合層。	DevOps 工程師, 解決方案架構師, 協力廠商整合開發人員,
在開發環境中執行遷移。	適用於開發環境的自動化移轉活動，包括系統佈建、資料移轉、應用程式移轉、安裝和組態。	DevOps 工程師
在生產環境中執行移轉。	生產環境的自動化移轉活動，包括系統佈建、資料移轉、應用程式移轉、安裝和組態。	DevOps 工程師

網路活動

任務	描述	所需技能
為每個 VPC 定義 CIDR 區塊。	為每個非預設 VPC 定義無類別網域間路由 (CIDR) 區塊 (IP 範圍和遮罩)。 持續時間：少於一周	DevOps 工程師, 方案架構師
定義子網路和可用區域。	定義每個非預設 VPC 中使用的子網路和可用區域。 持續時間：少於一周	DevOps 工程師, 方案架構師

任務	描述	所需技能
定義安全性群組。	定義用於控制 AWS 資源安全性的安全群組和安全群組規則。 持續時間：少於一周	DevOps 工程師, 方案架構師
定義網路 ACL。	定義網路存取控制清單 (ACL) 以控制子網路邊界的安全性。 持續時間：少於一周	DevOps 工程師, 方案架構師

移轉資料庫

任務	描述	所需技能
準備來源資料庫。	使用 AWS DMS 準備每個來源資料庫，以便持續複寫到 AWS 雲端。	DevOps 工程師, 方案架構師
建立 OpenText 核心元件的資料庫。	建立「開啟文字」 TeamSite、LiveSite和「 MediaBin 元件」所需的資料庫。請確定已根據 OpenText 安裝文件正確設定使用者和存取權限。	解決方案架構師、 OpenText 領導、 OpenText 開發
從來源資料庫伺服器複製資料。	自動化將 OpenText 核心元件的資料從來源資料庫伺服器複製到目標資料庫伺服器的程序。	解決方案架構師、 OpenText 領導、 OpenText 開發
從資料庫伺服器同步資料。	自動執行從來源資料庫到目標資料庫的定期資料同步處理程序。	OpenText 開發者

內容移轉活動

任務	描述	所需技能
複製內 OpenText TeamSite 容存放區。	將內容存放區從來源 OpenText TeamSite 伺服器複製到目標 OpenText TeamSite 伺服器的程序自動化。	解決方案架構師、 OpenText 領導、 OpenText 開發
對應使用者和群組。	內部 OpenText TeamSite 使用者 ID 與目標系統 ID 的內部對應。	OpenText 鉛
同步處理 OpenText TeamSite 內容存放區。	自動執行來源和目標內容存放區的定期同步處理程序。這是作為遷移和 QA 過程的一部分實施。	OpenText 開發者
從 Web 伺服器複製資料。	將資料從來源網頁伺服器複製到目標網頁伺服器的程序自動化。	解決方案架構師、 OpenText 領導、 OpenText 開發
同步 Web 伺服器資料。	自動執行來源和目標 Web 伺服器資料的定期同步處理程序。	OpenText 開發者
從 Web 伺服器檔案系統複製資料。	將內容和其他 Web 資產從來源 Web 伺服器檔案系統複製到目標網頁伺服器的程序自動化。	解決方案架構師、 OpenText 領導、 OpenText 開發
同步 Web 伺服器檔案系統。	自動執行從來源 Web 伺服器檔案系統到目標網頁伺服器的內容和其他 Web 資產的定期同步處理程序。	OpenText 開發者
產生摘要和索引。	自動執行任何可產生摘要或使 OpenText TeamSite 用 Web 伺服器內容做為資料來源的其	解決方案架構師、 OpenText 領導、 OpenText 開發

任務	描述	所需技能
	他索引 (例如 Web 搜尋) 的處理程序。	
同步饋送和索引的生成。	在資料同步後，自動執行定期重新產生摘要和索引的程序。	OpenText 開發者

測試和品質保證活動

任務	描述	所需技能
執行移轉 QA。	測試目標 AWS 環境、應用程式和服務，以確保正確建置和設定自動化遷移程序。	DevOps 工程師， OpenText 領導，QA 測試儀
進行性能測試。	在特定工作負載下測試響應性和穩定性方面的性能。調查、測量、驗證或驗證目的地系統的其他品質屬性，例如可擴充性和可靠性。 為了使此測試很有用，您必須擁有與生產環境相同大小的測試環境。 持續時間：一至兩週	DevOps 工程師， OpenText 領導
安全性測試。	漏洞掃描和滲透測試，以揭示應用程序安全機制中的潛在缺陷，以保護數據並根據需要維護功能。 為了使此測試非常有用，您必須擁有相當於生產環境的網路和安全性方面的測試環境。 持續時間：一至兩週	DevOps 工程師， OpenText 領導

營運整合活動

任務	描述	所需技能
檢查操作準備程度。	了解您目前如何執行 IT 操作，以及在 AWS 雲端中的操作方式。您可以通過定義雲操作模式來實現這一業務成果。 持續時間：一周	DevOps 工程師，OpenText 領導，服務交付經理
投資於營運自動化。	投資於自動化以交付 AWS 操作模型。	DevOps 工程師，OpenText 領導，服務交付經理
整合作業。	繼續使用目前的 IT 工具，並透過整合到 AWS 雲端來擴充這些工具。	DevOps 工程師，OpenText 領導，服務交付經理

切換活動

任務	描述	所需技能
切換 DNS 服務。	將網域名稱系統 (DNS) 從現有主機手動切換到以 AWS 雲端為基礎的主機。 持續時間：一小時	DevOps 工程師，OpenText 領導
測試災難恢復。	測試災難復原、備份還原，並執行自動化測試。 持續時間：一天	DevOps 工程師，OpenText 領導，QA 測試儀
驗證監控和分析。	驗證監視和分析是否正常運作。 持續時間：兩小時	DevOps 工程師，OpenText 領導

任務	描述	所需技能
關閉舊環境並要求關閉伺服器。	持續時間：三天	DevOps 工程師，OpenText 領導

相關資源

- 客戶受管政策
- 在您的 AWS 帳戶中建立 IAM 使用者
- IAM 使用者群組
- 將政策附加到 IAM 使用者群組
- 監控您的使用情況和成本
- 管理 AWS 帳戶
- Amazon EC2 的 IAM 角色
- AWS Support 的存取許可
- 使用創建軌跡
- 啟用記錄檔完整性驗證 CloudTrail
- Amazon S3 存儲桶政策 CloudTrail
- 將事件傳送至 CloudWatch 記錄檔
- 使用 AWS CLI 設定 AWS Config
- 啟用 S3 儲存貯體和物件的 CloudTrail 事件記錄
- 設定 AWS KMS 金鑰政策 CloudTrail
- 使用 AWS KMS 受管金鑰 (SSE-KMS) 加密 CloudTrail 日誌檔
- 如何啟用和停用自動按鍵旋轉
- 如何在使用 AWS 帳戶的根存取金鑰時接收通知
- 建立流程記錄
- 您的 VPC 的安全群組
- 更新 VPC 對等連線的路由表

將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列

創建者：賽克里希納南布魯（AWS）和新德莎帕特魯（AWS）

環境：PoC 或試點	來源：甲骨文數據庫	目標：Aurora 兼容或 Amazon RDS for PostgreSQL
R 類型：重新平台	工作負載：甲骨文; 開源	技術：移轉、儲存與備份、資料庫
AWS 服務：Amazon Aurora; AWS DMS; Amazon S3; Amazon RDS		

Summary

此模式說明如何將甲骨文字元大型物件 (CLOB) 值分割為個別的資料列，以及適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS)。PostgreSQL 不支援 CLOB 資料類型。

具有間隔分割區的表格會在來源 Oracle 資料庫中識別，而且會擷取表格名稱、分割區類型、分割區間隔以及其他描述資料，並將其載入到目標資料庫中。您可以使用 AWS Database Migration Service (AWS DMS) 將小於 1 GB 的 CLOB 資料以文字形式載入目標資料表，或者可以以 CSV 格式匯出資料，將資料載入 Amazon Simple Storage Service (Amazon S3) 儲存貯體，然後將其遷移到目標 PostgreSQL 資料庫。

移轉之後，您可以使用此模式提供的自訂 PostgreSQL 程式碼，根據新行字元識別碼 (`CHR(10)`) 將 CLOB 資料分割成個別資料列，並填入目標資料表。

先決條件和限制

先決條件

- 具有間隔分割區和具有 CLOB 資料類型之記錄的 Oracle 資料庫表格。
- 與 Aurora PostgreSQL 相容或 Amazon RDS 版 PostgreSQL 資料庫，其資料表結構類似於來源資料表 (相同的欄和資料類型)。

限制

- CLOB 值不得超過 1 GB。
- 目標資料表中的每一列都必須有新的行字元識別碼。

產品版本

- Oracle 12c
- Aurora 郵政 11.6

架構

下圖顯示了一個源甲骨文表與 CLOB 數據，以及 Aurora PostgreSQL 兼容版本 11.6 中的等效 PostgreSQL 表。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

您可以使用下列用戶端工具來連線、存取和管理與 Aurora PostgreSQL 相容和 Amazon RDS for PostgreSQL 資料庫。（這些工具不在此模式中使用。）

- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。
- DBeaver 是面向開發人員和數據庫管理員的開源數據庫工具。您可以使用此工具來操作、監視、分析、管理和移轉資料。

最佳實務

如需將資料庫從甲骨文遷移到 PostgreSQL 的最佳實務，請參閱 AWS 部落格文章將甲骨文資料庫遷移到 Amazon RDS PostgreSQL 或 Amazon Aurora PostgreSQL：遷移程序和基礎設施考量的最佳實務。

如需設定用於遷移大型二進位物件的 AWS DMS 任務的最佳實務，請參閱 AWS DMS 文件中的遷移大型二進位物件 (LOB)。

史诗

識別 CLOB 資料

任務	描述	所需技能
分析 CLOB 資料。	在來源 Oracle 資料庫中，分析 CLOB 資料以查看其是否包含欄標題，以便決定將資料載入目標資料表的方法。 要分析輸入數據，請使用以下查詢。 `SELECT * FROM clobdata_or;`	開發人員
將 CLOB 資料載入目標資料庫。	將具有 CLOB 資料的表格移轉到 Aurora 或 Amazon RDS 目標資料庫中的臨時 (暫存) 資料表。您可以使用 AWS DMS 或將資料以 CSV 檔案形式上傳到 Amazon S3 儲存貯體。 如需使用 AWS DMS 進行此任務的相關資訊，請參閱 AWS DMS 文件中的使用 Oracle 資料庫做為來源和使用 PostgreSQL 資料庫做為目標。	遷移工程師，DBA

任務	描述	所需技能
	如需使用 Amazon S3 進行此任務的相關資訊，請參閱 AWS DMS 文件中的使用 Amazon S3 做為目標。	
驗證目 PostgreSQL 表。	使用目標資料庫中的下列查詢，針對來源資料驗證目標資料 (包括標頭)。 `SELECT * FROM clobdata_pg;` `SELECT * FROM clobdatatarget;` 將結果與來源資料庫的查詢結果進行比較 (從第一個步驟開始)。	開發人員
將 CLOB 數據拆分為單獨的行。	執行 [其他資訊] 區段中提供的自訂 PostgreSQL 程式碼，以分割 CLOB 資料，並將其插入目標 PostgreSQL 資料表中的個別資料列。	開發人員

驗證資料。

任務	描述	所需技能
驗證目標表中的數據。	驗證通過使用以下查詢插入到目標表中的數據。 `SELECT * FROM clobdata_pg;` `SELECT * FROM clobdatatarget;`	開發人員

相關資源

- CLOB 資料類型 (Oracle 文件集)
- 資料類型 (文件集)

其他資訊

用於拆分 CLOB 數據的 PostgreSQL 函數

```
do
$$
declare
totalstr varchar;
str1 varchar;
str2 varchar;
pos1 integer := 1;
pos2 integer ;
len integer;


begin
        select rawdata||chr(10) into totalstr from clobdata_pg;
        len :=  length(totalstr) ;
        raise notice 'Total length : %',len;
        raise notice 'totalstr : %',totalstr;
        raise notice 'Before while loop';


        while pos1 < len  loop


                  select position (chr(10) in totalstr) into pos2;
                 raise notice '1st position of new line : %',pos2;


                str1 := substring (totalstr,pos1,pos2-1);
                raise notice 'str1 : %',str1;


                   insert into clobdatatarget(data) values (str1);
                   totalstr := substring(totalstr,pos2+1,len);
```

```
                    raise notice 'new totalstr :%',totalstr;
                len :=  length(totalstr) ;


        end loop;
end
$$
LANGUAGE 'plpgsql' ;
```

輸入和輸出範例

在遷移資料之前，您可以使用下列範例來試用 PostgreSQL 程式碼。

建立含有三個輸入行的 Oracle 資料庫。

```
CREATE TABLE clobdata_or (
id INTEGER GENERATED ALWAYS AS IDENTITY,
rawdata clob  );


insert into clobdata_or(rawdata) values (to_clob('test line 1') || chr(10) ||
 to_clob('test line 2') || chr(10) || to_clob('test line 3') || chr(10));
COMMIT;


SELECT * FROM clobdata_or;
```

這將顯示以下輸出。

id	原始數據
1	測試線 1 測試線 2 測試線 3

將來源資料載入 PostgreSQL 暫存資料表 (clobdata_pg) 中進行處理。

```
SELECT * FROM clobdata_pg;

CREATE TEMP TABLE clobdatatarget (id1 SERIAL,data VARCHAR );

<Run the code in the additional information section.>
```

```
SELECT * FROM clobdatatarget;
```

這將顯示以下輸出。

ID1	資料
1	測試線 1
2	測試線 2
3	測試線 3

透過資料庫連結使用直接的 Oracle 資料汲取匯入，將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon RDS

創建者：里茲旺德 (AWS)

環境：生產	來源：內部部署 Oracle 資料庫	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：AWS DMS；AWS Direct Connect；Amazon RDS		

Summary

許多模式涵蓋了使用 Oracle 資料泵 (一種原生 Oracle 公用程式)，將現場部署 Oracle 資料庫遷移到亞馬遜 RDS for Oracle 工作負載，這是遷移大型 Oracle 工作負載的首選方式。這些模式通常涉及將應用程式結構描述或表格匯出到傾印檔案，將傾印檔案傳輸到 Amazon RDS for Oracle 上的資料庫目錄，然後從傾印檔案匯入應用程式結構描述和資料。

使用這種方法，移轉可能需要更長的時間，具體取決於資料大小以及將傾印檔案傳輸到 Amazon RDS 執行個體所需的時間。此外，傾印檔案位於 Amazon RDS 執行個體的 Amazon Elastic Block Store (Amazon EBS) 磁碟區上，該磁碟區必須足夠容納資料庫和傾印檔案。匯入後刪除傾印檔案時，無法擷取空白空間，因此您會繼續支付未使用的空間費用。

此模式透過資料庫連結使用 Oracle 資料泵 API (DBMS_DATAPUMP)，在 Amazon RDS 執行個體上執行直接匯入，以減輕這些問題。此模式會在來源與目標資料庫之間同時啟動匯出和匯入管線。此模式不需要調整傾印檔案的 EBS 磁碟區大小，因為磁碟區上不會建立或儲存傾印檔案。此方法可節省未使用磁碟空間的每月成本。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶。
- 透過至少兩個可用區域的私有子網路設定的虛擬私有雲端 (VPC)，以提供 Amazon RDS 執行個體的網路基礎設施。

- 內部部署資料中心中的 Oracle 資料庫。
- 單一可用區域中的現有 Amazon RDS Oracle 執行個體。使用單一可用區域可改善移轉期間的寫入效能。異地同步備份部署可在切換前 24—48 小時啟用。
- AWS Direct Connect (建議用於大型資料庫)。
- 內部部署的網路連線和防火牆規則，設定為允許從 Amazon RDS 執行個體到現場部署 Oracle 資料庫的輸入連線。

限制

- Amazon RDS for Oracle 文的數據庫大小限制為 64 TiB（截至 2022 年 12 月）。

產品版本

- 來源資料庫：Oracle 資料庫版本 10g 版本 1 及更新版本。
- 目標資料庫：如需 Amazon RDS 上支援的最新版本和版本清單，請參閱 AWS 文件中的 Amazon RDS for Oracle 文件。

架構

源, 技術, 堆棧

- 內部部署或雲端中的自我管理 Oracle 資料庫

目標技術堆疊

- Amazon RDS for Oracle

目標架構

下圖顯示在單一可用區域環境中從現場部署 Oracle 資料庫遷移到 Amazon RDS for Oracle 的架構。箭頭方向描繪了架構中的數據流。圖表不會顯示啟動連線的元件。

1. 適用於 Oracle 執行個體的 Amazon RDS 會連接到現場部署來源 Oracle 資料庫，以透過資料庫連結執行全負載遷移。
2. AWS DMS 會連接到現場部署來源 Oracle 資料庫，以使用變更資料擷取 (CDC) 執行持續複寫。

3. CDC 變更會套用至 Amazon RDS for Oracle 資料庫。

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。此病毒碼使用 CDC 和僅複製資料變更設定。
- AWS Direct Connect 會透過標準乙太網路光纖纜線將您的內部網路連結到直接 Connect 位置。透過此連線，您可以直接建立公有 AWS 服務的虛擬界面，同時略過網路路徑中的網際網路服務供應商。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。

其他工具

- 「Oracle 資料汲取」可協助您以高速將資料和中繼資料從一個資料庫移至另一個資料庫。
- Oracle 即時客戶端或 SQL 開發人員等客戶端工具用於在數據庫上連接和運行 SQL 查詢。

最佳實務

雖然 AWS Direct Connect 在現場部署網路和 AWS 之間使用專用的私有網路連線，但是請考慮下列選項，為傳輸中的資料提供額外的安全性和資料加密：

- 使用 Amazon 網站對站點 VPN 或從現場部署網路到 AWS 網路的 IPSec VPN 連線的虛擬私人網路 (VPN)
- 在內部部署 Oracle 資料庫上設定的 Oracle 資料庫原生網路加密
- 使用 TLS 加密

史诗

準備內部部署來源 Oracle 資料庫

任務	描述	所需技能
設定從目標資料庫到來源資料庫的網路連線。	設定現場部署網路和防火牆，以允許從目標 Amazon RDS	網路管理員、安全工程師

任務	描述	所需技能
	執行個體連入到現場部署來源 Oracle 資料庫的連入連線。	
建立具有適當權限的資料庫使用者。	在內部部署來源 Oracle 資料庫中建立資料庫使用者，並具有使用「Oracle 資料汲取」在來源和目標之間移轉資料的權限。 `GRANT CONNECT to <migration_user>;` `GRANT DATAPUMP_EXP_FULL_DATABASE to <migration_user>;` `GRANT SELECT ANY TABLE to <migration_user>;`	DBA

任務	描述	所需技能
準備用於 AWS DMS CDC 移轉的現場部署來源資料庫。	(選擇性) 在完成 Oracle 資料泵完整負載之後，準備用於 AWS DMS CDC 移轉的現場部署來源 Oracle 資料庫： 1. 設定「Oracle 資料汲取」移轉期間管理「倒溯」所需的其他權限。 `GRANT FLASHBACK ANY TABLE to <migration_user>;` `GRANT FLASHBACK ARCHIVE ADMINISTER to <migration_user>;` 2. 若要在 AWS DMS 的自我管理 Oracle 來源上設定所需的使用者帳戶權限，請參閱 AWS D MS 文件。 3. 若要使用 AWS DMS 為 CDC 準備 Oracle 自我管理來源資料庫，請參閱 AWS DMS 文件。	DBA
安裝和設定 SQL 開發人員。	安裝並設定 SQL 開發人員，以便在來源和目標資料庫上連線並執行 SQL 查詢。	DBA，移民工程師

任務	描述	所需技能
產生命令檔以建立表格空間。	使用下列範例 SQL 查詢，在來源資料庫上產生指令碼。 `SELECT 'CREATE TABLESPACE ' tablespace_name ' DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE UNLIMITED;' from dba_tablespaces where tablespace_name not in ('SYSTEM', 'SYSAUX','TEMP','UNDOTBS1') order by 1;` 該腳本將應用於目標數據庫。	DBA
產生指令碼以建立使用者、設定檔、角色和權限。	若要產生命令檔來建立資料庫使用者、設定檔、角色和權限，請使用「Oracle 客戶 Support 務部」文件「如何擷取使用者的 DDL」(包括使用 dbms_metadata.get_ddl (文件識別碼 2739952.1) 中的指令集 (需要 Oracle 帳戶)。 該腳本將應用於目標數據庫。	DBA

為甲骨文執行個體準備目標亞馬遜 RDS

任務	描述	所需技能
建立來源資料庫的資料庫連結，並驗證連線能力。	若要建立內部部署來源資料庫的資料庫連結，您可以使用下列範例命令。	DBA

任務	描述	所需技能
	`CREATE DATABASE LINK link2src CONNECT TO <migration_user_account> IDENTIFIED BY <password> USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>) (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';` 若要驗證連線能力，請執行下列 SQL 命令。 `select * from dual@link2src;` 如果響應是成功的連接X。	
執行指令碼以準備目標執行個體。	執行先前產生的指令碼以準備適用於 Oracle 執行個體的目標 Amazon RDS： 1. 資料表空間 2. 描述檔 3. 角色 這有助於確保「Oracle 資料汲取」移轉可以建立綱要及其物件。	DBA，移民工程師

透過資料庫連結使用「Oracle 資料汲取匯入」來執行完整載入移轉

任務	描述	所需技能
移轉必要的結構描述。	若要將必要的結構描述從來源現場部署資料庫遷移到目標 Amazon RDS 執行個體，請使用其他資訊一節中的程式碼： • 若要移轉單一結構描述，請從 [其他資訊] 區段執行程式碼 1。 • 若要移轉多個結構描述，請從 [其他資訊] 區段執行程式碼 2。 若要調整移轉的效能，您可以執行下列命令來調整 parallel 處理程序的數目。 `DBMS_DATAPUMP.SET_ PARALLEL (handle => v_hdnl, degree => 4);`	DBA
收集綱要統計資料以改善效能。	「收集綱要統計資料」命令會傳回為資料庫物件收集的 Oracle 查詢最佳化處理程 使用此資訊，最佳化處理程式可以針對這些物件選取任何查詢的最佳執行計劃。 `EXECUTE DBMS_STAT S.GATHER_SCHEMA_ST ATS(ownname => '<schema_name>');`	DBA

使用 Oracle 資料泵和 AWS DMS 執行全負載移轉和 CDC 複寫

任務	描述	所需技能
擷取來源內部部署 Oracle 資料庫上的 SCN。	擷取來源內部部署 Oracle 資料庫上的系統變更編號 (SCN)。您將使用 SCN 進行全負載匯入，並作為 CDC 複寫的起點。 若要在來源資料庫上產生目前的 SCN，請執行下列 SQL 敘述句。 `SELECT current_scn FROM V$DATABASE;`	DBA
執行結構描述的完整負載移轉。	若要將所需的結構描述 (FULL LOAD) 從來源現場部署資料庫遷移到目標 Amazon RDS 執行個體，請執行以下操作： • 若要移轉單一結構描述，請從 [其他資訊] 區段執行程式碼 3。 • 若要移轉多個結構描述，請從 [其他資訊] 區段執行程式碼 4。 在程式碼中，取代<CURRENT_SCN_VALUE_IN_SOURCE_DATABASE> 為從來源資料庫擷取的 SCN。 `DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value`	DBA

任務	描述	所需技能
	=> <CURRENT_SCN_VALUE _IN_SOURCE_DATABAS E>); 若要調整移轉的效能，您可以調整 parallel 程序的數目。 DBMS_DATAPUMP.SET_ PARALLEL (handle => v_hdnl, degree => 4);	
停用已移轉結構描述下的觸發器。	在您開始執行 AWS DMS 僅限 CDC 的任務之前，請先停用遷移的結構描述TRIGGERS下的。	DBA
收集綱要統計資料以改善效能。	「收集綱要統計資料」命令會傳回為資料庫物件收集的 Oracle 查詢最佳化處理程 使用此資訊，最佳化處理程式可以針對這些物件選取任何查詢的最佳執行計劃。 EXECUTE DBMS_STAT S.GATHER_SCHEMA_ST ATS(ownname => '<schema_name>');	DBA

任務	描述	所需技能
使用 AWS DMS 執行從來源到目標的持續複寫。	使用 AWS DMS 執行從來源 Oracle 資料庫到目標 Amazon RDS (適用於甲骨文執行個體) 的持續複寫。 如需詳細資訊，請參閱使用 AWS DMS 建立持續複寫的任務和部落格文章如何在 AWS DMS 中使用原生 CDC 支援。	DBA，移民工程師

切換到 Amazon RDS for Oracle

任務	描述	所需技能
在切換前 48 小時在執行個體上啟用異地同步備份。	如果這是生產執行個體，建議您在 Amazon RDS 執行個體上啟用異地同步備份部署，以提供高可用性 (HA) 和災難復原 (DR) 的優點。	DBA，移民工程師
停止僅限 AWS DMS 光碟的任務 (如果 CDC 已開啟)。	1. 確保 AWS DMS 任務的 Amazon CloudWatch 指標上的來源延遲和目標延遲顯示為 0 秒。 2. 停止僅限 AWS DMS 光碟的工作。	DBA
啟用觸發器。	啟用您在 CDC 工作建立之前停用的觸發器。	DBA

相關資源

AWS

- 使用 AWS DMS 為 CDC 準備 Oracle 自我管理的來源資料庫

- 使用 AWS DMS 建立進行中複寫的任務
- 異地同步備份部署提供高可用
- 如何在 AWS DMS 中使用原生 CDC 支援 (部落格文章)

甲骨文文件

- 數據管理系統

其他資訊

程式碼 1：僅限全負載移轉，單一應用程式結構描述

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''<schema_name>'')');  --
 To migrate one selected schema
    DBMS_DATAPUMP.METADATA_FILTER (hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')'); --
 To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

程式碼 2：僅限全負載移轉、多個應用程式結構描述

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'SCHEMA_LIST',
 '''<SCHEMA_1>'',''<SCHEMA_2>'', ''<SCHEMA_3>'''); -- To migrate multiple schemas
```

```
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

程式碼 3：僅限 CDC 工作之前的全負載移轉，單一應用程式結構描述

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE( handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''<schema_name>'')');  --
 To migrate one selected schema
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
    DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value =>
 <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

代碼 4：僅限 CDC 任務之前的全負載遷移，多個應用程序模式

```
DECLARE
    v_hdnl NUMBER;
BEGIN
    v_hdnl := DBMS_DATAPUMP.OPEN (operation => 'IMPORT', job_mode  => 'SCHEMA',
 remote_link => '<DB LINK Name to Source Database>', job_name  => null);
    DBMS_DATAPUMP.ADD_FILE (handle => v_hdnl, filename  => 'import_01.log',  directory
 => 'DATA_PUMP_DIR', filetype  => dbms_datapump.ku$_file_type_log_file);
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'SCHEMA_LIST',
 '''<SCHEMA_1>'',''<SCHEMA_2>'', ''<SCHEMA_3>'''); -- To migrate multiple schemas
    DBMS_DATAPUMP.METADATA_FILTER (v_hdnl, 'EXCLUDE_PATH_EXPR','IN (''STATISTICS'')');
 -- To prevent gathering Statistics during the import
```

```
    DBMS_DATAPUMP.SET_PARAMETER (handle => v_hdnl, name => 'FLASHBACK_SCN', value =>
 <CURRENT_SCN_VALUE_IN_SOURCE_DATABASE>); -- SCN required for AWS DMS CDC only task.
    DBMS_DATAPUMP.SET_PARALLEL (handle => v_hdnl, degree => 4);  -- Number of parallel
 processes performing export and import
    DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

混合移轉方法可以更好地運作的案例

在極少數情況下，來源資料庫包含具有數百萬個資料列的資料表和非常大型的 LOBLOSEGION 資料行，此模式會拖慢移轉速度。Oracle 會透過網路連結，一次移轉一個大區段。它會從來源表格擷取單一資料列 (連同 LOB 資料欄資料)，然後將資料列插入目標資料表中，重複此程序直到所有資料列都移轉為止。透過資料庫連結的 Oracle 資料汲取不支援 LOB 區段的大量載入或直接路徑載入機制。

在這種情況下，我們建議您執行以下操作：

- 新增下列中繼資料篩選器，可在「Oracle 資料汲取」移轉期間略過已識別的表格。

```
dbms_datapump.metadata_filter(handle =>h1, name=>'NAME_EXPR', value => 'NOT IN
 (''TABLE_1'',''TABLE_2'')');
```

- 使用 AWS DMS 任務 (全負載遷移，必要時搭配 CDC 複寫) 遷移已識別的資料表。AWS DMS 會從來源 Oracle 資料庫擷取多個資料列，然後將它們以批次形式插入目標 Amazon RDS 執行個體，進而提升效能。

將 Oracle 電子商務套件遷移到 Amazon RDS 定制

由西蒙·坎寧安（AWS），傑迪普南迪（AWS），尼廷薩克森納（AWS）和毗濕奴文納科塔（AWS）創建

環境：生產	來源：Amazon EC2 或內部部署	目標：Amazon RDS 自定義
R 類型：重新平台	工作量：甲骨文	技術：移轉、資料庫、基礎架構
AWS 服務：Amazon EFS; Amazon RDS; AWS Secrets Manager		

Summary

Oracle 電子商務套件是一種企業資源計劃 (ERP) 解決方案，用於自動化整個企業的處理，例如財務、人力資源、供應鏈和製造。它具有三層架構：客戶端，應用程序和數據庫。以前，您必須在自我管理的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上執行 Oracle 電子商務套件資料庫，但現在您可以從 Amazon Relational Database Service 服務 (Amazon RDS) 自訂中受益。

Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。它可以自動執行資料庫管理工作和作業，同時讓身為資料庫管理員的您可以存取和自訂資料庫環境和作業系統。當您將 Oracle 資料庫遷移到 Amazon RDS 客製化時，Amazon Web Services (AWS) 會處理繁重的工作，例如備份任務並確保高可用性，同時您可以專注於維護 Oracle 電子商務套件應用程式和功能。如需移轉時要考慮的關鍵因素，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

此模式著重於使用 Oracle 復原管理器 (RMAN) 備份和 Amazon 執行個體之間的 Amazon 彈性檔案系統 (Amazon EFS) 共用檔案系統，將 Amazon EC2 上的獨立 Oracle 資料庫遷移到 Amazon RDS 自訂的步驟。病毒碼使用 RMAN 完整備份（有時也稱為層級 0 備份）。 為了簡單起見，它使用冷備份，其中應用程序關閉，並且數據庫被掛載而不是打開。(您也可以使用「Oracle 資料保全」或「RMAN」複製來進行備份。 但是，此模式不包括這些選項。)

如需在 AWS 上架構 Oracle 電子商務套件以提供高可用性和災難復原的相關資訊，請參閱使用作用中待命資料庫在 Amazon RDS 自訂上為 Oracle 電子商務套件設定 HA/DR 架構的模式。

備註：此樣式提供 Oracle 客戶服務備註的連結。您需要「Oracle 客戶 Support 部」帳戶才能存取這些文件。

先決條件和限制

先決條件

- 甲骨文 12.1.0.2 或 19c 版 (最低 19.3) 來源資料庫，在 Amazon EC2 上執行，並搭配甲骨文 7 或紅帽企業 Linux (RHEL) 7.x 版。此模式假設來源資料庫名稱為，VIS且 Oracle 19c 的其他容器資料庫名稱為VISCDB，但您可以使用其他名稱。

  附註：只要您在現場部署網路和 Amazon 虛擬私人雲端 (Amazon VPC) 之間具有適當的網路連線，您也可以將此模式用於現場部署 Oracle 來源資料庫。
- 甲骨文電子商務套件 12.2.x 版應用程序（視覺實例）。此程序已在 12.2.11 版上進行測試。
- 單一 Oracle 電子商務套件應用模組層。不過，您可以調整此模式來處理多個應用程式層級。
- 對於甲骨文 12.1.0.2，Amazon RDS 自定義設置了至少 16 GB 的交換空間。否則，12c 範例光碟會顯示警告。（甲骨文 19c 不需要示例光盤，如本文件後面提到的那樣。）

開始移轉之前，請先完成下列步驟：

1. 在 Amazon RDS 主控台上，使用資料庫名稱 VIS (或您的來源資料庫名稱) 建立適用於 Oracle 資料庫的 Amazon RDS 自訂執行個體。如需指示，請參閱 AWS 文件中的使用 Amazon RDS 自訂和 Oracle 專用的 Amazon RDS 自訂 — 資料庫環境中的新控制功能部落格文章。這可確保將資料庫名稱設定為與來源資料庫相同的名稱。（如果保留空白，EC2 實例和數據庫名稱將設置為ORCL。） 確保您至少使用已套用至來源的修補程式建立自訂引擎版本 (CEV)。如需詳細資訊，請參閱 Amazon RDS 文件中的準備建立 CEV。

   甲骨文 19c 注意事項：目前，對於甲骨文 19c，Amazon RDS 容器數據庫名稱可以自定義。預設值為 RDSCDB。請務必使用與來源 EC2 執行個體相同的系統識別碼 (SID) 來建立 RDS 自訂 Oracle 執行個體。例如，在此模式中，會假設 Oracle 19c SID 位於來源執行環境VISCDB上。因此，Amazon RDS 自定義目標甲骨文 19c SID 也應該是。VISCDB
2. 使用足夠的儲存空間、vCPU 和記憶體來配置 Amazon RDS 自訂資料庫執行個體，以符合 Amazon EC2 來源資料庫。 為此，您可以根據 vCPU 和記憶體比對 Amazon EC2 執行個體類型。
3. 建立一個 Amazon EFS 檔案系統，並將其掛接到 Amazon EC2 和 Amazon RDS 自訂執行個體。 如需指示，請參閱整合 Amazon RDS 自訂適用於甲骨文與 Amazon EFS 的部落格文章。此模式假設您已在來源 Amazon Amazon EC2 和目標 Amazon RDS 自訂資料庫執行個體/RMAN上掛接

Amazon EFS 磁碟區，並且可以在來源和目標之間進行網路連線。您也可以使用 Amazon FSx 或任何共用磁碟機來使用相同的方法。

假設

此模式假設您的應用程式和資料庫使用邏輯主機名稱，這會減少移轉步驟的數目。您可以調整這些步驟以使用實體主機名稱，但邏輯主機名稱可降低移轉程序的複雜性。如需使用邏輯主機名稱優點的相關資訊，請參閱下列支援注意事項：

- 對於 12 分，甲骨文客戶 Support 注意事項 2246690.1
- 對於 19c，甲骨文客戶 Support 注意事項 2617788.1

此模式不涵蓋甲骨文 12c 到 19c 的升級案例，並著重於將 Amazon EC2 上運行的相同版本甲骨文數據庫遷移到 Amazon RDS 自定義為甲骨文。

Amazon RDS 自訂適用於 Oracle 本位目錄支援自訂。(「Oracle 本位目錄」會儲存 Oracle 二進位檔案。) 您可以將指定路徑/rdsdbbin/oracle的預設路徑變更為，例如/d01/oracle/VIS/19c。為了簡單起見，此模式中的指示採用預設路徑/rdsdbbin/oracle。

限制

此模式不支持以下功能和配置：

- 將資料庫ARCHIVE_LAG_TARGET參數設定為 60—7200 範圍以外的值
- 停用資料庫執行個體記錄模式 (NOARCHIVELOG)
- 關閉 EC2 實例的EBS-optimized屬性
- 修改連接到 EC2 執行個體的原始亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區
- 新增 EBS 磁碟區或將磁碟區類型從gp2變更為 gp3
- 對於 TNS 文件的 Support
- 更改位control_file置和名稱（必須是/rdsdbdata/db/VISCDB_A/controlfile/control-01.ctl，其中VISCDB是 CDB 名稱）

如需有關這些組態和其他不受支援組態的其他資訊，請參閱 Amazon RDS 文件中的修正不支援的組態。

產品版本

如需 Amazon RDS 自訂支援的 Oracle 資料庫版本和執行個體類別，請參閱 Amazon RDS 自訂適用於 Oracle 的可用性和要求。

架構

下列架構圖表示在 AWS 上的單一可用區域中執行的 Oracle 電子商務套件系統。應用程式層可透過應用程式負載平衡器存取，應用程式和資料庫都位於私有子網路中，而 Amazon RDS 自訂和 Amazon EC2 資料庫層使用 Amazon EFS 共用檔案系統來存放和存取 RMAN 備份檔案。

工具

AWS 服務

- Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。它可以自動執行資料庫管理工作和作業，同時讓身為資料庫管理員的您可以存取和自訂資料庫環境和作業系統。
- Amazon Elastic File System (Amazon EFS) 是一種簡單的無伺服器彈性檔案系統，無需管理或佈建即可新增和移除檔案。此模式使用 Amazon EFS 共用檔案系統來存放和存取 RMAN 備份檔案。
- AWS Secrets Manager 是 AWS 受管服務，可讓您輕鬆輪換、管理和擷取資料庫登入資料、API 金鑰和其他機密資訊。Amazon RDS 自訂會在建立資料庫時，將 key pair 和資料庫使用者登入資料存放在 Secrets Manager 在此模式中，您可以從 Secrets Manager 擷取資料庫使`ADMIN`用者密碼，以建立`RDSADMIN`和使用者，以及變更 sys 和系統密碼。

其他工具

- RMAN 是為 Oracle 資料庫提供備份和復原支援的工具。此模式會使用 RMAN 在 Amazon RDS 自訂上還原的 Amazon EC2 上執行來源 Oracle 資料庫的冷備份。

最佳實務

- 使用邏輯主機名稱。這會大幅減少您必須執行的複製後指令碼數目。如需詳細資訊，請參閱 Oracle 客戶 Support 部注意事項 2246690.1。
- Amazon RDS 自定義默認使用 Oracle 自動內存管理（AMM）。 如果您想要使用巨型核心，可以將 Amazon RDS 自訂設定為改用自動共用記憶體管理 (ASMM)。
- 依預設，保持`memory_max_target`參數為啟用狀態。架構會在背景中使用此參數來建立僅供讀取複本。

- 啟用「Oracle 倒溯資料庫」。此功能在容錯移轉 (非切換) 測試案例中非常有用，以恢復待命狀態。
- 對於資料庫初始化參數，請自訂 Amazon RDS 自訂資料庫執行個體提供的標準 PFILE，而不是使用 Oracle 來源資料庫中的 SPFILE。這是因為在 Amazon RDS 自訂中建立僅供讀取複本時，空格和註解會導致問題。如需資料庫初始化參數的詳細資訊，請參閱 Oracle 客戶 Support 注意事項 396009.1。

在下面的史詩部分中，我們為 Oracle 12.1.0.2 和 19c 提供了單獨的說明，其中細節不同。

史诗

關閉來源應用程式

任務	描述	所需技能
關閉應用程式。	若要關閉來源應用程式，請使用下列指令： `$ su - applmgr` `$ cd $INST_TOP/admin/scripts` `$ ./adstpall.sh`	DBA
建立 .zip 檔案。	在來源應用程式層上建立`appsutil.zip` 檔案。您稍後將使用此檔案來設定 Amazon RDS 自訂資料庫節點。 `$ perl $AD_TOP/bin/admkappsutil.pl`	DBA
將 .zip 檔案複製到 Amazon EFS。	`appsutil.zip` 從複製`$INST_TOP/admin/out` 到您共用的 Amazon EFS 磁碟區 (`/RMAN/appsutil`)。您可以使用安全複製 (SCP) 或其他傳輸機制手動傳輸檔案。	DBA

預先複製來源資料庫

任務	描述	所需技能
在 Amazon EC2 上預先複製資料庫層。	以 Oracle 使用者身分登入並執行： `$ cd $ORACLE_HOME/appsutil/scripts/$CONTEXT_NAME` `$ perl adpreclone.pl dbTier` 檢查產生的記錄檔，以確認作業已順利完成。	DBA
將 appsutil.zip 複製到共用的 Amazon EFS 檔案系統。	建立 tar 備份並將其複製`$ORACLE_HOME/appsutil` 到共用的 Amazon EFS 檔案系統 (例如，`/RMAN/appsutil`)： `$ cd $ORACLE_HOME` `$ tar cvf sourceappsutil.tar appsutil` `$ cp sourceappsutil.tar /RMAN/appsutil`	DBA

對來源 Amazon EC2 資料庫執行冷 RMAN 完整備份

任務	描述	所需技能
建立備份指令碼。	將來源資料庫的 RMAN 完整備份至共用的 Amazon EFS 檔案系統。	DBA

任務	描述	所需技能
	為了簡單起見，此模式會執行冷 RMAN 備份。不過，您可以修改這些步驟，使用「Oracle 資料保全」執行熱 RMAN 備份，以減少停機時間。 1. 在掛接模式下啟動來源 Amazon EC2 資料庫： `$ sqlplus / as sysdba` `$ SQL> shutdown immediate` `$ SQL> startup mount` 2. 建立 RMAN 備份指令碼 (視您的 Oracle 版本而定，使用下列其中一個範例，或執行其中一個現有的 RMAN 指令碼)，將資料庫備份到您所掛載的 Amazon EFS 檔案系統 (/RMAN在本範例中)。 對於甲骨文 12.1.0.2： `$ vi FullRMANColdBackup.sh` `#!/bin/bash` `. /home/oracle/.bash_profile` `export ORACLE_SID=VIS` `export ORACLE_HOME=/d01/oracle/VIS/12.1.0` `export DATE=$(date +%y-%m-%d_%H%M%S)`	

任務	描述	所需技能
	rman target / log=/RMAN /VISDB_${DATE}.log << EOF run { allocate channel ch1 device type disk format '/RMAN/visdb_full_ bkp_%u'; allocate channel ch2 device type disk format '/RMAN/visdb_full_ bkp_%u'; crosscheck backup; delete noprompt obsolete; BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG; backup archivelog all; release channel ch1; release channel ch2; } EOF 對於甲骨文 19c： $ vi FullRMANColdBackup .sh #!/bin/bash . /home/oracle/.bash _profile export ORACLE_SI D=VISCDB export ORACLE_HOME=/ d01/oracle/VIS/19c export DATE=$(date + %y-%m-%d_%H%M%S)	

任務	描述	所需技能
	rman target / log=/RMAN /VISDB_${DATE}.log << EOF run { allocate channel ch1 device type disk format '/RMAN/visdb_full_ bkp_%u'; allocate channel ch2 device type disk format '/RMAN/visdb_full_ bkp_%u'; crosscheck backup; delete noprompt obsolete; BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG; backup archivelog all; backup current controlfile format '/ RMAN/cntrl.bak'; release channel ch1; release channel ch2; } EOF	
執行備份指令碼。	變更權限、以 Oracle 使用者身分登入，然後執行指令碼： $ chmod 755 FullRMANC oldBackup.sh $ ./FullRMANColdBack up.sh	DBA

任務	描述	所需技能
檢查錯誤並記下備份檔案的名稱。	檢查 RMAN 記錄檔是否有錯誤。如果一切正常，列出控制文件的備份。請注意輸出檔案的名稱。 對於甲骨文 12.1.0.2： `RMAN> connect target /` `RMAN> list backup of controlfile;` `BS Key Type LV Size Device Type Elapsed Time Completion Time` `------- ---- -- ---------- --------- ---------- ---------------` `9 Full 1.11M DISK 00:00:04 23-APR-22` `BP Key: 9 Status: AVAILABLE Compressed: YES Tag: TAG20220423T121011` `Piece Name: /RMAN/visdb_full_bkp_100rlsbt` `Control File Included: Ckp SCN: 122045953 96727 Ckp time: 23-APR-22` `/RMAN/visdb_full_bkp_100rlsbt` 稍後當您在 Amazon RDS 自訂上還原資料庫時，您將使用備份檔案。	DBA

任務	描述	所需技能
	對於甲骨文 19c： `RMAN> connect target /` `RMAN> list backup of controlfile;` `BS Key Type LV Size Device Type Elapsed Time Completion Time` `------- ---- -- ---------- ----------- ------------ ---------- -- ------------ ---------- 38 Full 17.92M DISK 00:00:01 25-NOV-22 BP Key: 38 Status: AVAILABLE Compressed: NO Tag: TAG20221125T095014 Piece Name: /RMAN/cntrl.bak Control File Included: Ckp SCN: 122046201 88873 Ckp time: 23-NOV-22` `/RMAN/cntrl.bak` 稍後當您在 Amazon RDS 自訂上還原資料庫時，您將使用備份檔案。	

設定目標 Amazon RDS 自訂資料庫

任務	描述	所需技能
更改 hosts 文件並設置主機名。	注意：本節中的命令必須以 root 使用者身分執行。	DBA

任務	描述	所需技能
	1. 在 Amazon RDS 自訂資料庫執行個體上編輯/etc/hosts 檔案。執行此操作的簡單方法是從來源 Amazon EC2 資料庫主機檔案複製資料庫和應用程式主機項目。 `<IP-address> OEBS-app01.localdomain OEBS-app01 OEBS-app01log.localdomain OEBS-app01log` `<IP-address> OEBS-db01.localdomain OEBS-db01 OEBS-db01log.localdomain OEBS-db01log` 其中<IP-address> 是資料庫節點 IP 位址，您應該將其取代為 Amazon RDS 自訂 IP 位址。邏輯主機名稱會附加*log。 2. 執行以下hostnamectl 指令來變更資料庫主機名稱： `$ sudo hostnamectl set-hostname --static persistent-hostname` 例如： `$ sudo hostnamectl set-hostname --static OEBS-db01log`	

任務	描述	所需技能
	如需其他資訊，請參閱有關指派靜態主機名稱的知識中心文章。 3. 重新啟動 Amazon RDS 自訂資料庫執行個體。 不要擔心關閉數據庫，因為您將在後面的步驟中刪除它。 `$ reboot` 4. Amazon RDS 自訂資料庫執行個體備份時，請登入並確認主機名稱已變更： `$ hostname` `oebs-db01`	

任務	描述	所需技能
安裝「Oracle 電子商務套件」軟體。	將 Oracle 電子商務套件建議的 RPM 安裝到 Amazon RDS 自訂資料庫執行個體上的 Oracle 本位目錄位置。如需詳細資訊，請參閱 Oracle 客戶 Support 部備註 #1330701 .1。以下是部分清單。每個版本的 RPM 清單都會變更，因此請檢查以確定已安裝所有必要的 RPM。 以 root 使用者身分執行： `$ sudo yum -y update` `$ sudo yum install -y elfutils-libelf-devel*` `$ sudo yum install -y libXp-1.0.2-2.1*.i686` `$ sudo yum install -y libXp-1.0.2-2.1*` `$ sudo yum install -y compat-libstdc++-*` 請先確認已安裝所有必要的修補程式，然後再繼續執行下一個步驟。	DBA

任務	描述	所需技能
安裝 VNC 伺服器。	注意：您可以針對 Oracle 19c 省略此步驟，因為不再需要使用「範例光碟」；請參閱 Oracle 客戶 Support 中心注意事項 2782085.1。 對於甲骨文 12.1.0.2： 安裝 VNC 伺服器及其相依桌上型電腦套件。這是在下一個步驟中安裝 12c 範例 CD 的要求。 1. 以 root 使用者身分執行： `$ sudo yum install -y tigervnc-server` `$ sudo yum install -y *kde*` `$ sudo yum install -y *xorg*` 2. 啟動使`rdsdb`用者的 VNC 伺服器，並設定 VNC 的密碼： `$ su - rdsdb` `$ vncserver :1` `$ vncpassword`	DBA

任務	描述	所需技能
安裝 12c 範例光碟。	注意：您可以針對 Oracle 19c 省略此步驟，因為不再需要使用「範例光碟」；請參閱 Oracle 客戶 Support 中心注意事項 2782085.1。 對於甲骨文 12.1.0.2： 1. 請從以下位置下載安裝檔案：https://edelivery.oracle.com/。 如果是甲骨文電子商務套件 12.2.11 — 甲骨文資料庫 12c 版本 1 (12.1.0.2)，請尋找適用於 Linux x86-64 V100102-01.zip 的範例。 2. 創建一個目錄來存儲示例 CD： `$ mkdir /RMAN/12c examples` 3. 使用您選擇的傳輸機制 (例如 SCP)，將範例 CD .zip 檔案複製到此目錄： `V100102-01.zip` 4. 將所有權變更為`rdsdb`： `$ chown -R rdsdb:rdsdb /RMAN/12cexamples` 5. 作為用`rdsdb`戶，解壓縮文件：	DBA

任務	描述	所需技能
	`$ unzip V10010201.zip` 6. 從可以存取 VNC 用戶端和 Amazon RDS 自訂功能的用戶端進行 Connect 線。請確定您已開啟必要的網路連線和防火牆連接埠，以允許 VNC 存取。例如，執行的 VNC 伺服器需`display :1`要在與 Amazon RDS 自訂 EC2 主機相關聯的安全群組上開啟連接埠 5901。 7. 切換到您複製示例 CD 的目錄： `$ cd /RMAN/12cexamples/examples` 8. 執行安裝程式。 確保驗證文`oraInst.loc` 件的位置。 `./runInstaller -invPtrLoc /rdsdbbin/oracle.12.1.custom.r1.EE.1/oraInst.loc` 9. 在安裝「範例光碟」期間，請使用下列參數： `Skip Software Update` `Downloads` `Select Oracle Home` `12.1.0.2` `(Oracle Base = /rdsdbbin)`	

任務	描述	所需技能
	(Software Location = /rdsdbbin/oracle/12.1.custom.r1.EE.1) 10. 安裝程式包含五個步驟，並附有提示。 請按照以下步驟操作，直到安裝完成。	

卸除入門資料庫並建立儲存資料庫檔案的目錄

任務	描述	所需技能
暫停自動化模式。	在繼續執行後續步驟之前，您必須暫停 Amazon RDS 自訂資料庫執行個體上的自動化模式，以確保自動化不會干擾 RMAN 活動。 使用以下 AWS Command Line Interface (AWS CLI) (AWS CLI) 命令暫停自動化。(請確定您已先設定 AWS CLI。) aws rds modify-db-instance \ --db-instance-identifier VIS \ --automation-mode all-paused \ --resume-full-automation-mode-minute 360 \ --region eu-west-1 當您指定暫停的持續時間時，請確定您預留足夠的時間進行	DBA

任務	描述	所需技能
	RMAN 還原。這取決於來源資料庫的大小，因此請相應地修改 360 值。	
卸除入門資料庫。	刪除現有的 Amazon RDS 自定義數據庫。 以 Oracle 本位目錄使用者的身分執行下列命令。（除非您對其進行了自定義`rdsdb`，否則默認用戶是。） `$ sqlplus / as sysdba` `SQL> shutdown immediate ;` `SQL> startup nomount restrict;` `SQL> alter database mount;` `SQL> drop database;` `SQL> exit`	DBA

任務	描述	所需技能
建立儲存資料庫檔案的目錄。	對於甲骨文 12.1.0.2： 建立資料庫、控制檔、資料檔和線上記錄的目錄。在上一個命令中使用control_files 參數的父目錄 (在本例中為VIS_A)。以 Oracle 本位目錄使用者身分執行下列命令 (預設為rdsdb)。 `$ mkdir -p /rdsdbdata/db/VIS_A/controlfile` `$ mkdir -p /rdsdbdata/db/VIS_A/datafile` `$ mkdir -p /rdsdbdata/db/VIS_A/onlinelog` 對於甲骨文 19c： 建立資料庫、控制檔、資料檔和線上記錄的目錄。在上一個命令中使用control_files 參數的父目錄 (在本例中為VISCDB_A)。以 Oracle 本位目錄使用者身分執行下列命令 (預設為rdsdb)。 `$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/controlfile` `$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/datafile` `$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/onlinelog`	DBA

任務	描述	所需技能
	`$ mkdir -p /rdsdbdata/db/cdb/VISCDB_A/onlinelog/arch` `$ mkdir /rdsdbdata/db/pdb/VISCDB_A`	

任務	描述	所需技能
建立並修改 Oracle 電子商務套件的參數檔案。	在此步驟中，您不會從來源資料庫複製伺服器參數檔 (SPFILE)。相反地，您需要使用透過 Amazon RDS 自訂資料庫執行個體建立的標準參數檔案 (PFILE)，並新增 Oracle 電子商務套件所需的參數。 當您刪除資料庫時，Amazon RDS 自動化會建立`init.ora`檔案的備份，該備份會與 Amazon RDS 自訂資料庫相關聯。此檔案稱為`oracle_pfile` ，位於中`/rdsdbdata/config` 。 對於甲骨文 12.1.0.2： 1. 將 `/rdsdbdata/config/oracle_pfile` 複製至 `$ORACLE_HOME` 。 `$ cp /rdsdbdata/config/oracle_pfile $ORACLE_HOME/dbs/initVIS.ora` 2. 在 Amazon RDS 自訂資料庫執行個體上編輯`initVIS.ora` 檔案。驗證源上的所有參數，並根據需要添加任何參數。如需詳細資訊，請參閱甲骨文客戶 Support 備註 396009.1。 重要:請確定您新增的參數中沒有註解。註解會造成自動化	DBA

任務	描述	所需技能
	問題，例如建立僅供讀取複本和發出 point-in-time 復原 (PITR)。 3. 根據您的需求，將類似下列內容的參數新增至`initVIS.ora` 檔案： `*.workarea_size_policy='AUTO'` `*.plsql_code_type='INTERPRETED'` `*.cursor_sharing='EXACT'` `*._b_tree_bitmap_plans=FALSE` `*.session_cached_cursors=500` `*.optimizer_adaptive_features=false` `*.optimizer_secure_view_merging=false` `*.SQL92_SECURITY=TRUE` `*.temp_undo_enabled=true` `_system_trig_enabled = TRUE` `nls_language = american` `nls_territory = america` `nls_numeric_characters = "."` `nls_comp = binary` `nls_sort = binary` `nls_date_format = DD-MON-RR` `nls_length_semantics = BYTE` `aq_tm_processes = 1` `_sort_elimination_cost_ratio =5`	

任務	描述	所需技能
	_like_with_bind _as_equality = TRUE _fast_full_scan_enable d = FALSE _b_tree_bitmap_plans = FALSE optimizer_secure_view _merging = FALSE _optimizer_autostats_ job = FALSE parallel_max_servers = 8 parallel_min_servers = 0 parallel_degree_policy = MANUAL sec_case_sensitive_l ogon = FALSE compatible = 12.1.0 o7_dictionary_access ibility = FALSE utl_file_dir =/tmp 4. 修改以下內容。這些值將取決於您的來源系統，因此請根據您目前的設定對其進行修改。 *.open_cursors=500 *.undo_tablespace ='APPS_UNDOTS1 5. 移除 SPFILE 參考。 *.spfile='/rdsdbbi n/oracle/dbs/spfil eVIS.ora' 備註：	

任務	描述	所需技能
	• 不要改變 Amazon RDS 自訂 PFILE 為control_files 和db_unique_name 提供的值。Amazon RDS 期望這些值。如果您 future 嘗試建立僅供讀取複本，則偏離它們會造成問題。 • 根據預設，Amazon RDS 自訂使用自動記憶體管理 (AMM)。如果您想要使用巨型記憶體，可以將 Amazon RDS 自訂設定為使用自動共用記憶體管理 (ASMM)。 • 依預設，保持memory_max_target 參數為啟用狀態。Amazon RDS 架構會在背景中使用此功能來建立僅供讀取複本。 6. 執行下startup nomount列命令，確認initVIS.ora 檔案沒有問題： `SQL> startup nomount pfile=/rdsdbbin/oracle/dbs/initVIS.ora;` `SQL> create spfile='/rdsdbdata/admin/VIS/pfile/spfileVIS.ora' from pfile;` `SQL> exit` 7. 為 SPFILE 建立符號連結。	

任務	描述	所需技能
	`$ ln -s /rdsdbdata/admin/VIS/pfile/spfileVIS.ora $ORACLE_HOME/dbs/` 對於甲骨文 19c： 1. 將 `/rdsdbdata/config/oracle_pfile` 複製至 `$ORACLE_HOME` 。 `$ cp /rdsdbdata/config/oracle_pfile $ORACLE_HOME/dbs/initVISCDB.ora` 2. 在 Amazon RDS 自訂資料庫執行個體上編輯`initVISCDB.ora` 檔案。驗證源上的所有參數，並根據需要添加任何參數。如需詳細資訊，請參閱甲骨文客戶 Support 備註 396009.1。 重要:請確定您新增的參數中沒有註解。 如果有意見，則會造成自動化問題，例如建立僅供讀取複本和發出 point-in-time 復原 (PITR)。 3. 根據您的需求，將類似下列內容的參數新增至`initVISCDB.ora` 檔案。 `*.instance_name=VISCDB`	

任務	描述	所需技能
	*.sec_case_sensit ive_logon= FALSE *.result_cache_ma x_size = 600M *.optimizer_adaptive_p lans =TRUE *.optimizer_adaptive_ statistics = FALSE *.pga_aggregate_limit = 0 *.temp_undo_enabled = FALSE *._pdb_name_case_sens itive = TRUE *.event='10946 trace name context forever, level 8454144' *.workarea_size_p olicy='AUTO' *.plsql_code_t ype='INTERPRETED' *.cursor_sharing=' EXACT' *._b_tree_bitmap_pla ns=FALSE *.session_cached_c ursors=500 *.optimizer_secu re_view_merging=false *.SQL92_SECURITY=TRUE _system_trig_enabled = TRUE nls_language = american nls_territory = america nls_numeric_charact ers = "." nls_comp = binary nls_sort = binary nls_date_format = DD- MON-RR	

任務	描述	所需技能
	nls_length_semantics = BYTE aq_tm_processes = 1 _sort_eliminatio n_cost_ratio =5 _like_with_bind _as_equality = TRUE _fast_full_scan_enable d = FALSE _b_tree_bitmap_plans = FALSE optimizer_secure_view _merging = FALSE _optimizer_autostats_ job = FALSE parallel_max_servers = 8 parallel_min_servers = 0 parallel_degree_policy = MANUAL 4. 修改以下內容。這些值將取決於您的來源系統，因此請根據您目前的設定對其進行修改。 *.open_cursors=500 *.undo_tablespace ='UNDOTBS1' 5. 移除 SPFILE 參照： *.spfile='/rdsdbbi n/oracle/dbs/spfil eVISCDB.ora' 備註：	

任務	描述	所需技能
	• 不要改變 Amazon RDS 自訂 PFILE 為control_files 和db_unique_name 提供的值。Amazon RDS 期望這些值。如果您 future 嘗試建立僅供讀取複本，則偏離它們會造成問題。 • 根據預設，Amazon RDS 自訂使用自動記憶體管理 (AMM)。如果您想要使用巨型記憶體，可以將 Amazon RDS 自訂設定為使用自動共用記憶體管理 (ASMM)。 • 依預設，保持memory_max_target 參數為啟用狀態。Amazon RDS 架構會在背景中使用此功能來建立僅供讀取複本。 6. 執行下startup nomount列命令，確認initVISCDB.ora 檔案沒有問題： `SQL> startup nomount pfile=/rdsdbbin/oracle/dbs/initVISCDB.ora;` `SQL> create spfile='/rdsdbdata/admin/VISCDB/pfile/spfileVISCDB.ora' from pfile;` `SQL> exit`	

任務	描述	所需技能
	7. 為 SPFILE 建立符號連結。 `$ ln -s /rdsdbdata/admin/VISCDB/pfile/spfileVISCDB.ora $ORACLE_HOME/dbs/`	

任務	描述	所需技能
從備份還原 Amazon RDS 自訂資料庫。	對於甲骨文 12.1.0.2： 1. 使用您先前在來源上擷取的備份檔案來還原控制檔案： `RMAN> connect target /` `RMAN> RESTORE CONTROLFILE FROM '/RMAN/visdb_full_bkp_100rlsbt';` `Starting restore at 10-APR-22` `using target database control file instead of recovery catalog` `allocated channel: ORA_DISK_1` `channel ORA_DISK_1: SID=201 device type=DISK` `channel ORA_DISK_1: restoring control file` `channel ORA_DISK_1: restore complete, elapsed time: 00:00:01` `output file name=/rdsdbdata/db/VIS_A/controlfile/control-01.ctl` `Finished restore at 10-APR-22` 2. 編目備份片段，以便您可以發出RMAN `restore`： `RMAN> alter database mount;`	DBA

任務	描述	所需技能
	RMAN> catalog start with '/RMAN/visdb'; 3. 創建一個腳本來恢復數據庫： $ vi restore.sh rman target / log=/home /rdsdb/rman.log << EOF run { set newname for database to '/rdsdbdata/db/VIS _A/datafile/%b'; restore database; switch datafile all; switch tempfile all; } EOF 4. 將來源還原到目標 Amazon RDS 自訂資料庫。您必須變更指令碼的權限以允許執行它，然後執行指restore.sh 令碼來還原資料庫。 $ chmod 755 restore.sh $ nohup ./restore.sh & 對於甲骨文 19c： 1. 使用您先前在來源上擷取的備份檔案來還原控制檔案： RMAN> connect target / RMAN> RESTORE CONTROLFI LE FROM '/RMAN/cn trl.bak';	

任務	描述	所需技能
	Starting restore at 07-JUN-23 using target database control file instead of recovery catalog allocated channel: ORA_DISK_1 channel ORA_DISK_1: SID=201 device type=DISK channel ORA_DISK_1: restoring control file channel ORA_DISK_1: restore complete, elapsed time: 00:00:01 output file name=/rdsdbdata/db/cdb/VISCDB_A/controlfile/control-01.ctl Finished restore at 07-JUN-23	
	2. 編目備份片段，以便您可以發出RMAN `restore`： `RMAN> alter database mount;` `RMAN> catalog start with '/RMAN/visdb';`	
	如果您在使用`start with`指令時遇到問題，您可以個別新增備份部分；例如： `RMAN> catalog backuppiece '/RMAN/visdb_full_bkp_1d1e507m';`	

任務	描述	所需技能
	然後為每個備份部分重複該命令。 3. 建立還原資料庫的指令碼。根據您的要求修改可插拔的數據庫名稱。根據可用的 vCPUs 數量配置 parallel 通道，以加速還原程序。 `$ vi restore.sh` `rman target / log=/home /rdsdb/rmancdb.log <<  EOF` `run {` `allocate channel c1  type disk;` `allocate channel c2  type disk;` `.... .... ....` `allocate channel c<N>  type disk;` `set newname for database  to '/rdsdbdata/db/cdb /VISCDB_A/datafile/ %b';` `set newname for database  root to '/rdsdbda ta/db/cdb/VISCDB_A/ datafile/%f_%b';` `set newname for  database "PDB$SEED" to  '/rdsdbdata/db/cdb/ pdbseed/%f_%b';` `set newname for  pluggable database VIS  to '/rdsdbdata/db/pdb /VISCDB_A/%f_%b';` `restore database;` `switch datafile all;` `switch tempfile all;`	

任務	描述	所需技能
	release channel c1; release channel c2; release channel c3; .... release channel c<N>; } EOF 4. 將來源還原到目標 Amazon RDS 自訂資料庫。您必須變更指令碼的權限以允許執行它，然後執行指令restore.sh 來還原資料庫。 $ chmod 755 restore.sh $ nohup ./restore.sh &	

任務	描述	所需技能
檢查記錄檔是否有問題。	對於甲骨文 12.1.0.2： 1. 檢閱檔案以確認沒有問rman.log題： `$ cat /home/rdsdb/rman.log` 2. 確認在控制檔中註冊的記錄檔路徑： `SQL> select member from v$logfile;` `MEMBER` `----------` `/d01/oracle/VIS/data/log1.dbf` `/d01/oracle/VIS/data/log2.dbf` `/d01/oracle/VIS/data/log3.dbf` 3. 重新命名記錄檔，以符合目標的檔案路徑。 替換路徑以匹配上一步的輸出： `SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log1.dbf' TO '/rdsdbdata/db/VIS_A/onlinelog/log1.dbf';` `SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log2.`	DBA

任務	描述	所需技能
	dbf' TO '/rdsdbdata/db/VIS_A/onlinelog/log2.dbf'; SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/data/log3.dbf' TO '/rdsdbdata/db/VIS_A/onlinelog/log3.dbf'; 對於甲骨文 19c： 1. 檢閱檔案以確認沒有問rmancdb.log 題： $ cat /home/rdsdb/rmancdb.log 2. 確認在控制檔中註冊的記錄檔路徑： SQL> select member from v$logfile; MEMBER ---------- /d01/oracle/VIS/oradata/VISCDB/redo03.log /d01/oracle/VIS/oradata/VISCDB/redo02.log /d01/oracle/VIS/oradata/VISCDB/redo01.log	

任務	描述	所需技能
	3. 重新命名記錄檔，以符合目標的檔案路徑。 替換路徑以匹配上一步的輸出： `SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VISCDB/redo01.log' TO '/rdsdbdata/db/cdb/VISCDB_A/onlinelog/log1.dbf';` `SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VISCDB/redo02.log' TO '/rdsdbdata/db/cdb/VISCDB_A/onlinelog/log2.dbf';` `SQL> ALTER DATABASE RENAME FILE '/d01/oracle/VIS/oradata/VISCDB/redo03.log' TO '/rdsdbdata/db/cdb/VISCDB_A/onlinelog/log3.dbf';` 4. 確認控制檔中註冊的路徑、記錄檔狀態和群組編號： `SQL> column REDOLOG_FILE_NAME format a50` `SQL> SELECT a.GROUP#, a.status, b.MEMBER AS REDOLOG_FILE_NAME, (a.BYTES/1024/1024) AS SIZE_MB` `FROM v$log a` `JOIN v$logfile b ON a.Group#=b.Group#` `ORDER BY a.GROUP#;`	

任務	描述	所需技能
	GROUP# STATUS REDOLOG_FILE_NAME SIZE_MB 1 CURRENT /rdsdbdata/db/cdb/VISCDB_A/onlinelog/log1.dbf 512 2 INACTIVE /rdsdbdata/db/cdb/VISCDB_A/onlinelog/log2.dbf 512 3 INACTIVE /rdsdbdata/db/cdb/VISCDB_A/onlinelog/log3.dbf 512	

任務	描述	所需技能
確認您可以開啟 Amazon RDS 自訂資料庫，並建立 OMF 記錄檔。	Amazon RDS 自訂軟體使用 Oracle 受管檔案 (OMF) 來簡化操作。您可以將僅供讀取複本升級為獨立執行個體，但必須先使用 OMF 建立記錄檔。這是為了確保在升級執行個體時使用正確的路徑。如需如何提升僅供讀取複本的詳細資訊，請參閱 Amazon RDS 文件。當您嘗試升級僅供讀取複本時，無法使用 OMF 檔案可能會造成問題。 1. 用以下命令打開數據庫`resetlogs` ： `SQL> alter database open resetlogs;` 注意：如果您收到錯誤訊息 ORA-00392：執行緒 1 的 log xx 已清除，則不允許作業，請依照 ORA-00392 疑難排解一節中的步驟執行。 2. 確認資料庫已開啟： `SQL> select open_mode from v$database;` `OPEN_MODE` `----------` `READ WRITE` 3. 建立 OMF 記錄檔。使用先前日誌檔查詢的輸出，根據您的需求變更群組編號、群組數	DBA

任務	描述	所需技能
	目和大小。下面的例子從組 4 開始，並為了簡單起見，添加了三個組。 `SQL> alter database add logfile group 4 size 512M;` `Database altered.` `SQL> alter database add logfile group 5 size 512M;` `Database altered.` `SQL> alter database add logfile group 6 size 512M;` `Database altered.` 4. 刪除以前的非 OMF 檔案。以下是您可以根據自己的需求和先前步驟中查詢輸出進行自訂的範例： `SQL> alter database drop logfile group 1;` `System altered.` `SQL> alter database drop logfile group 2;` `System altered.` `SQL> alter database drop logfile group 3;` `System altered.` 注意：如果您在嘗試刪除記錄檔時收到 ORA-01624 錯誤訊息，請參閱疑難排解一節。 5. 確認您可以看到已建立的 OMF 檔案。（甲骨文 12.1.0.2	

任務	描述	所需技能
	和 19c 的目錄路徑各不相同，但概念是相同的。) `SQL> select member from v$logfile;` `MEMBER` `----------` `/rdsdbdata/db/cdb/VISCDB_A/onlinelog/o1_mf_4_ksrbslny_.log` `/rdsdbdata/db/cdb/VISCDB_A/onlinelog/o1_mf_5_ksrchw0k_.log` `/rdsdbdata/db/cdb/VISCDB_A/onlinelog/o1_mf_6_ksrcnl9v_.log` 6. 重新啟動資料庫，並確認執行處理正在使用 SPFILE： `SQL> shutdown immediate` `SQL> startup` `SQL> show parameter spfile` 對於甲骨文 12.1.0.2，此查詢返回： `spfile /rdsdbbin/oracle/dbs/spfileVIS.ora` 對於甲骨文 19c，查詢返回：	

任務	描述	所需技能
	`spfile /rdsdbbin /oracle/dbs/spfile VISCDB.ora`	
	7. 僅適用於 Oracle 19c，請檢查容器資料庫的狀態，並視需要開啟它： `SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED ---------- --------- ---------- --------- - 2 PDB$SEED READ ONLY NO 3 VIS MOUNTED NO` `SQL> alter session set container=VIS; Session altered.` `SQL> alter database open; Database altered.` `SQL> alter database save state; Database altered.` `SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED ---------- --------- ---------- ----------`	

任務	描述	所需技能
	3 VIS READ WRITE NO SQL> exit 8. 從中刪除該init.ora文件$ORACLE_HOME/dbs ，因為您沒有使用 PFILE： $ cd $ORACLE_HOME/dbs 對於甲骨文 12.1.0.2，請使用以下命令： $ pwd /rdsdbbin/oracle/dbs $ rm initVIS.ora 對於甲骨文 19c，請使用以下命令： $ pwd /rdsdbbin/oracle/dbs $ rm initVISCDB.ora	

從密碼 Secrets Manager 擷取密碼、建立使用者和變更密碼

任務	描述	所需技能
從密碼 Secrets Manager 擷取密碼。	您可以在主控台或使用 AWS CLI 執行這些步驟。下列步驟提供主控台的指示。 1. 登入 AWS 管理主控台，開啟位於 https://console.a	DBA

任務	描述	所需技能
	ws.amazon.com/rds/ 的 Amazon RDS 主控台。 2. 在導覽窗格中，選擇「資料庫」，然後選取 Amazon RDS 資料庫。 3. 選擇「組態」，並記下執行處理的資源 ID (格式為:db-WZ4WLCK6AOQ6TJGZKMGRCDCI3Y)。 4. 開啟 AWS Secrets Manager 主控台，網址為 https://console.aws.amazon.com/secretsmanager/。 5. 選擇與名稱相同的密碼do-not-delete-custom-<resource_id> ，其中resource-id 指的是您在步驟 3 中記下的執行個體 ID。 6. 選擇 Retrieve secret value (擷取秘密值)。	

任務	描述	所需技能
建立 RDSADMIN 使用者。	RDSADMIN是 Amazon RDS 自訂資料庫執行個體中的監控和協調器資料庫使用者。由於入門資料庫已卸除，且目標資料庫已使用 RMAN 從來源還原，因此您必須在還原操作後重新建立此使用者，以確保 Amazon RDS 自訂監控能如預期般運作。您也必須為RDSADMIN使用者建立個別的設定檔和表格空間。甲骨文 12.1.0.2 和 19c 的說明略有不同。 對於甲骨文 12.1.0.2： 1. 在 SQL 提示字元中輸入下列命令： `SQL> set echo on feedback on serverout on` `SQL> @?/rdbms/admin/utlpwdmg.sql` `SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_FUNCTION NULL;` 2. 建立設定檔RDSADMIN：	DBA

任務	描述	所需技能
	SQL> create profile RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER_CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTEMPTS 10 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; 3. 將SYS、SYSTEM和DBSNMP使用者設定檔設定為RDSADMIN： SQL> set echo on feedback on serverout on	

任務	描述	所需技能
	SQL> alter user SYS profile RDSADMIN; SQL> alter user SYSTEM profile RDSADMIN; SQL> alter user DBSNMP profile RDSADMIN; 4. 建立表RDSADMIN格空間： SQL> create bigfile tablespace rdsadmin datafile size 7M autoextend on next 1m Logging online permanent blocksize 8192 extent managemen t local autoallocate default nocompress segment space managemen t auto; 5. 建立使RDSADMIN用者。將RDSADMIN密碼取代為您先前從密碼管理員取得的密碼： SQL> create user rdsadmin identified by xxxxxxxxxx Default tablespace rdsadmin Temporary tablespace temp profile rdsadmin ; 6. 授與權限給RDSADMIN： SQL> grant select on sys.v_$instance to rdsadmin;	

任務	描述	所需技能
	SQL> grant select on sys.v_$archived_log to rdsadmin; SQL> grant select on sys.v_$database to rdsadmin; SQL> grant select on sys.v_$database_incarnation to rdsadmin; SQL> grant select on dba_users to rdsadmin; SQL> grant alter system to rdsadmin; SQL> grant alter database to rdsadmin; SQL> grant connect to rdsadmin with admin option; SQL> grant resource to rdsadmin with admin option; SQL> alter user rdsadmin account unlock identified by xxxxxxxxxxx; SQL> @?/rdbms/admin/userlock.sql SQL> @?/rdbms/admin/utlrp.sql 對於甲骨文 19c： 1. 在 SQL 提示字元中輸入下列命令： SQL> set echo on feedback on serverout on SQL> @?/rdbms/admin/utlpwdmg.sql	

任務	描述	所需技能
	SQL> alter profile default LIMIT FAILED_LOGIN_ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL; 2. 建立設定檔RDSADMIN。 注意：RDSADMIN在甲骨文 19c C## 中有一個前綴。 這是因為資料庫參common_user_prefix 數設定為C##。RDSADMIN在甲骨文 12.1.0.2 中沒有前綴。 SQL> create profile C##RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER_CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTEMPTS 10 PASSWORD_LIFE_TIME UNLIMITED	

任務	描述	所需技能
	PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; 3. 將SYS、SYSTEM和DBSNMP使用者設定檔設定為 RDSADMIN： SQL> alter user SYS profile C##RDSADMIN; SQL> alter user SYSTEM profile C##RDSADMIN; SQL> alter user DBSNMP profile C##RDSADMIN; 4. 建立表RDSADMIN格空間： SQL> create bigfile tablespace rdsadmin datafile size 7M autoextend on next 1m Logging online permanent blocksize 8192 extent managemen t local autoallocate default nocompress segment space managemen t auto;	

任務	描述	所需技能
	5. 建立使RDSADMIN用者。將RDSADMIN密碼取代為您先前從密碼管理員取得的密碼。 `SQL> create user C##rdsadmin identified by xxxxxxxxxx profile C##rdsadmin container=all;` 6. 授與權限給RDSADMIN： `SQL> grant select on sys.v_$instance to c##rdsadmin;` `SQL> grant select on sys.v_$archived_log to c##rdsadmin;` `SQL> grant select on sys.v_$database to c##rdsadmin;` `SQL> grant select on sys.v_$database_incarnation to c##rdsadmin;` `SQL> grant select on dba_users to c##rdsadmin;` `SQL> grant alter system to C##rdsadmin;` `SQL> grant alter database to C##rdsadmin;` `SQL> grant connect to C##rdsadmin with admin option;` `SQL> grant resource to C##rdsadmin with admin option;`	

任務	描述	所需技能
	SQL> alter user C##rdsadmin account unlock identified by xxxxxxxxxxx; SQL> @?/rdbms/admin/use rlock.sql SQL> @?/rdbms/admin/utl rp.sql	

任務	描述	所需技能
建立主要使用者。	因為啟動器資料庫已經卸除，而且目標資料庫是使用 RMAN 從來源還原的，所以您必須重新建立 master 使用者。在此範例中，主要使用者名稱為 admin。 對於甲骨文 12.1.0.2： `SQL> create user admin identified by <password>; SQL> grant dba to admin` 對於甲骨文 19c： `SQL> alter session set container=VIS;` `Session altered.` `SQL> create user admin identified by <password>;` `User created.` `SQL> grant dba to admin;` `Grant succeeded.`	DBA

任務	描述	所需技能
變更超級使用者密碼。	1. 使用您從密碼管理員擷取的密碼來變更系統密碼。 對於甲骨文 12.1.0.2： `SQL> alter user sys identified by xxxxxxxxxx; SQL> alter user system identified by xxxxxxxxxx;` 對於甲骨文 19c： `SQL> alter user sys identified by xxxxxxxxxx container =all; SQL> alter user system identified by xxxxxxxxxx container =all;` 1. 變更EBS_SYSTEM 密碼。 對於甲骨文 12.1.0.2： `SQL> alter user ebs_system identified by xxxxxxxxxx;` 對於甲骨文 19c： 對於此版本，您還必須連接到容器數據庫，以在那裡更新EBS_SYSTEM 密碼。	DBA

任務	描述	所需技能
	`SQL> alter session set container=vis;` `SQL> alter user ebs_system identified by xxxxxxxxxx;` `SQL> exit;` 如果您未變更這些密碼，Amazon RDS Custom 會顯示錯誤訊息：資料庫監控使用者或使用者登入資料已變更。	

建立 Oracle 電子商務套件的目錄，安裝 ETCC，並執行自動設定

任務	描述	所需技能
建立 Oracle 電子商務套件所需的目錄。	1. 在 Amazon RDS 自訂 Oracle 資料庫上，以 Oracle 本位目錄使用者身分執行下列指令碼，以在中建立9idata目錄$ORACLE_HOME/nls/data/9idata 。此目錄是必需的 Oracle 電子商務套件。 `perl $ORACLE_HOME/nls/data/old/cr9idata.pl` 忽略此ORA_NLS10 訊息，因為您將在稍後的步驟中建立啟用前後關聯的環境。 2. 複製您先前從共用 Amazon EFS 檔案系統建立的檔案	

任務	描述	所需技能
	，然後在 Amazon RDS 自訂 Oracle 本位目錄中將其解壓。appsutil.tar 這會在目appsutil錄中建立目$ORACLE_HOME 錄。 `$ cd /RMAN/appsutil` `$ cp sourceappsutil.tar $ORACLE_HOME` `$ cd $ORACLE_HOME` `$ tar xvf sourceappsutil.tar appsutil` 3. 複製您先前儲存在 Amazon EFS 共用檔案系統上的檔案。appsutil.zip 這是您在應用程式層上建立的檔案。 以 Amazon RDS 自訂資料庫執行個體上的rdsdb使用者身分： `$ cp /RMAN/appsutil/appsutil.zip $ORACLE_HOME` `$ cd $ORACLE_HOME` 4. 解壓縮appsutil.zip 檔案，在 Oracle 本位目appsutil錄中建立目錄和子目錄： `$ unzip -o appsutil.zip` 該-o選項意味著某些文件將被覆蓋。	

任務	描述	所需技能
設定暴拉檔案和 SQLnet.ora 檔案。	您必須設定tnsnames.ora 檔案，才能使用「自動設定」工具連線到資料庫。在下面的例子中，你可以看到該 tnsnames.ora 文件是軟鏈接的，但默認情況下文件是空的。 `$ cd $ORACLE_HOME/network/admin` `$ ls -ltr` `-rw-r--r-- 1 rdsdb database 373 Oct 31 2013 shrept.lst` `lrwxrwxrwx 1 rdsdb database 30 Feb 9 17:17 listener.ora -> /rdsdbdata/config/listener.ora` `lrwxrwxrwx 1 rdsdb database 28 Feb 9 17:17 sqlnet.ora -> /rdsdbdata/config/sqlnet.ora` `lrwxrwxrwx 1 rdsdb database 30 Feb 9 17:17 tnsnames.ora -> /rdsdbdata/config/tnsnames.ora` 1. 建立項tnsnames.ora 目。由於 Amazon RDS 自動化剖析檔案的方式，因此您必須確定項目不包含任何空格、註解或額外的行。否則，使用某些 API（例如 create-db-instance-read-replica）時可能	DBA

任務	描述	所需技能
	會遇到問題。使用以下內容作為範例。 2. 根據您的要求更換端口，主機和 SID： `$ vi tnsnames.ora` `VIS=(DESCRIPTION=` `(ADDRESS_LIST=(ADD` `RESS=(PROTOCOL=TCP` `)(PORT=1521)(HOST=` `xx.xx.xx.xx)))(CON` `NECT_DATA=(SID=VIS)` `(SERVER=DEDICATED)))` 注意：文件中不應該有多餘的行。如果您不移除這些行，將 future 建立僅供讀取複本時可能會遇到問題。僅供讀取複本的建立可能會失敗，並顯示錯誤訊息：活動擲回例外狀況: HostManagerException: 無法成功呼叫任何主機上的限制應用程式。 3. 確認可以連線到資料庫： `$ tnsping vis` `OK (0 msec)` 4. 僅適用於甲骨文 19c，請更新`sqlnet.ora` 檔案。如果不這樣做，將導致錯誤 ORA-01017：無效的使用者名稱/密碼；當您嘗試連線到資料庫時，登入遭拒。`sqlnet.ora` 在中編輯`$ORACLE_HOME/`	

任務	描述	所需技能
	network/admin 以符合下列項目： `NAMES.DIRECTORY_PATH=(TNSNAMES, ONAMES, HOSTNAME)` `SQLNET.EXPIRE_TIME= 10` `SQLNET.INBOUND_CONNECT_TIMEOUT =60` `SQLNET.ALLOWED_LOGON_VERSION_SERVER=10` `HTTPS_SSL_VERSION=undetermined` 5. 測試連線能力： `$ sqlplus apps/****@vis`	

任務	描述	所需技能
設定資料庫。	現在您已經測試了與資料庫的連線，您可以使用 appsutil 公用程式來設定資料庫，以建立啟用內容的環境。 對於甲骨文 12.1.0.2： 1. 執行下列命令： `$ cd $ORACLE_HOME/appsutil/bin` `$ perl adbldxml.pl appsuser=apps` `Enter Hostname of Database server: oebs-db01` `Enter Port of Database server: 1521` `Enter SID of Database server: VIS` `Enter Database Service Name: VIS` `Enter the value for Display Variable: :1` `The context file has been created at: /rdsdbbin/oracle/appsutil/VIS_oebs-db01.xml` 2. `oraInst.loc` 從根使用者建立： `$ vi /etc/oraInst.loc` `inventory_loc=/rdsdbbin/oracle.12.1.custom.r1.EE.1/oraInventory` `inst_group=database`	DBA

任務	描述	所需技能
	3. 複製內容檔案以使用您在上一個步驟中建立的內容檔案來設定邏輯主機名稱。作為用rdsdb戶，運行： `$ cd $ORACLE_HOME/appsutil/clone/bin` `$ perl adclonectx.pl \` `contextfile=[ORACLE_HOME]/appsutil/[current context file] \` `template=[ORACLE_HOME]/appsutil/template/adxdbctx.tmp` 其中oebs-db01log 指的是邏輯主機名稱。例如： `$ perl adclonectx.pl \` `contextfile=/rdsdbbin/oracle.12.1.custom.r1.EE.1/appsutil/VIS_oebs-db01.xml \` `template=/rdsdbbin/oracle/appsutil/template/adxdbctx.tmp` `Target System Hostname (virtual or normal) [oebs-db01] : oebs-db01log` `Target System Base Directory : /rdsdbbin/oracle` `Target Instance is RAC (y/n) [n] : n` `Target System Database SID : VIS`	

任務	描述	所需技能
	Oracle OS User [rdsdb] : Oracle OS Group [rdsdb] : database Role separation is supported y/n [n] ? : n Target System utl_file_ dir Directory List : / tmp Number of DATA_TOP's on the Target System [1] : Target System DATA_TOP Directory 1 [/rdsdbbi n/oracle/data] : / rdsdbdata/db/VIS_A/ datafile/ Target System RDBMS ORACLE_HOME Directory [/rdsdbbin/oracle/ 12.1.0] : /rdsdbbin/ oracle Do you want to preserve the Display [:1] (y/n) : y Do you want the target system to have the same port values as the source system (y/n) [y] ? : y The new database context file has been created : /rdsdbbin/oracle.1 2.1.custom.r1.EE.1/ appsutil/clone/bin/ VIS_oebs-db01log.xml contextfile=/rdsdbbin/ oracle.12.1.custom	

任務	描述	所需技能
	.r1.EE.1/appsutil/clone/bin/VIS_oebs-db01log.xml 對於甲骨文 19c： 1. 執行下列命令： $ cd $ORACLE_HOME/appsutil/bin $ perl adbldxml.pl appsuser=apps Enter Hostname of Database server: oebs-db01 Enter Port of Database server: 1521 Enter SID of Database server: VIS Enter the database listener name:L_VISCDB_001 Enter the value for Display Variable: :1 The context file has been created at: /rdsdbbin/oracle/appsutil/VIS_oebs-db01.xml 2. oraInst.loc 從根使用者建立： $ vi /etc/oraInst.loc inventory_loc=/rdsdbbin/oracle/oraInventory inst_group=database	

任務	描述	所需技能
	3. 複製內容檔案以使用您在上一個步驟中建立的內容檔案來設定邏輯主機名稱。作為用rdsdb戶，運行： `$ cd $ORACLE_HOME/appsutil/clone/bin` `$ perl adclonectx.pl \` `contextfile=[ORACLE_HOME]/appsutil/[current context file] \` `template=[ORACLE_HOME]/appsutil/template/adxdbctx.tmp` 其中oebs-db01log 指的是邏輯主機名稱。例如： `$ perl adclonectx.pl \` `contextfile=/rdsdbbin/oracle/appsutil/VIS_oebs-db01.xml \` `template=/rdsdbbin/oracle/appsutil/template/adxdbctx.tmp` `Target System Hostname (virtual or normal) [oebs-db01] : oebs-db01log` `Target System Base Directory : /rdsdbbin/oracle` `Target Instance is RAC (y/n) [n] : n` `Target System CDB Name : VISCDB` `Target System PDB Name : VIS` `Oracle OS User [oracle] : rdsdb`	

任務	描述	所需技能
	Oracle OS Group [dba] : database Role separation is supported y/n [n] ? : n Number of DATA_TOP's on the Target System [2] : Target System DATA_TOP Directory 1 [/d01/oracle/VISCDB] : /rdsdbdata/db/pdb/VISCDB_A Target System DATA_TOP Directory 2 [/d01/oracle/data] : /rdsdbdata/db/pdb/VISCDB_A/datafile Specify value for OSBACKUPDBA group [database] : Specify value for OSDGDBA group [database] : Specify value for OSKMDBA group [database] : Specify value for OSRACDBA group [database] : Target System RDBMS ORACLE_HOME Directory [/d01/oracle/19.0.0] : /rdsdbbin/oracle Do you want to preserve the Display [:1] (y/n) : y Do you want the target system to have the same port values as the source system (y/n) [y] ? : y	

任務	描述	所需技能
	Validating if the source port numbers are available on the target system.. Complete port information available at /rdsdbbin/oracle/appsutil/clone/bin/out/VIS_oebs-db01log/portpool.lst New context path and file name [VIS_oebs-db01log.xml] : /rdsdbbin/oracle/appsutil/VIS_oebs-db01log.xml Do you want to overwrite it (y/n) [n] ? : y Replacing /rdsdbbin/oracle/appsutil/VIS_oebs-db01log.xml file. The new database context file has been created : contextfile=/rdsdbbin/oracle/appsutil/VIS_oebs-db01log.xml Check Clone Context logfile /rdsdbbin/oracle/appsutil/clone/bin/CloneContext_0609141428.log for details.	

任務	描述	所需技能
安裝 ETCC 並執行自動設定。	1. 安裝 Oracle 電子商務套件技術核心檢查程式 (ETCC)。 從我的甲骨文 Support 中心下載修補程式 17537119，並依照中的指示操作。README.txt 您將建立目錄etcc中名為的$ORACLE_HOME 目錄，解壓縮修補程式以建立名為的命令檔checkMTpatch.sh ，然後執行命令檔來檢查修補程式版本。 2. 執行 Autoconfig 公用程式，並傳遞新的邏輯主機名稱內容檔案。 對於甲骨文 12.1.0.2： `cd $ORACLE_HOME/appsutil/bin` `$ ./adconfig.sh contextfile=/rdsdbbin/oracle.12.1.customstom.r1.EE.1/appsutil/clone/bin/VIS_oebs-db01log.xml` 對於甲骨文 19c： 自動配置期望監聽器名稱匹CDBNAME配。因此，備份的原始監聽器組態檔將會L_<CDBNAME>_001 暫時使用。	DBA

任務	描述	所需技能
	$ lsnrctl stop L_VISCDB_ 001 $ cp -rp /rdsdbdata/ config/listener.ora / rdsdbdata/config/ listener.ora_orig $ vi /rdsdbdata/ config/listener.ora :%s/L_VISCDB_001/ VISCDB/g $ lsnrctl start VISCDB $ cd /rdsdbbin/oracle/a ppsutil $ . ./txkSetCfgCDB.env dboraclehome=/rdsd bbin/oracle.19.cus tom.r1.EE-CDB.1 Oracle Home being passed: /rdsdbbin/ oracle $ echo $ORACLE_HOME /rdsdbbin/orac le.19.custom.r1.EE- CDB.1 $ export ORACLE_SI D=VISCDB $ cd $ORACLE_HOME/ appsutil/bin $ perl $ORACLE_H OME/appsutil/bin/t xkPostPDBCreationT asks.pl -dboraclehome= $ORACLE_HOME -outdir= $ORACLE_HOME/appsut il/log -cdbsid=VISCDB -pdbsid=VIS -appsuser =apps -dbport=1521 - servicetype=onpremise	

任務	描述	所需技能
	`Enter the APPS Password: <apps password>` `Enter the CDB SYSTEM Password:<password from secrets manager>` 注意：如果您的資料庫目錄有所變更，請依照 Oracle 客戶 Support 注意事項 2525754.1 中的指示進行。	

為 Amazon RDS 自定義和甲骨文電子商務套件配置 TNS 項目

任務	描述	所需技能
為 Amazon RDS 自定義和甲骨文電子商務套件配置 TNS 條目。	自動設定會在預設位置產生 TNS ifile。對於 Oracle 12.1.0.2 (這是一個非國家開發資料庫) 和 Oracle19c PDB 的預設位置為。`$ORACLE_HOME/network/admin/$<CONTEXT_NAME>` Oracle 19c 的 CDB 會使用預設值`$ORACLE_HOME/network/admin/`，如同您在先前步驟`$TNS_ADMIN` 中執行自動設定時所產生的環境檔案中所定義。 對於 Oracle 12.1.0.2 和 19c CDB，您不會使用這些功能，因為自動設定所產生的`tnsnames.`	DBA

任務	描述	所需技能
	ora 和listener.ora 檔案不符合 Amazon RDS 要求，例如沒有空格或註解。而是使用 Amazon RDS 自訂資料庫隨附的一般檔案，以確保符合系統預期的內容，並減少錯誤範圍。 例如，Amazon RDS 自訂需要以下命名格式： `L_<INSTANCE_NAME>_001` 對於甲骨文 12.1.0.2，這將是： `L_VIS_001` 對於甲骨文 19c，這將是： `L_VISCDB_001` 以下是您將使用的listener.ora 文件的示例。這是在您建立 Amazon RDS 自訂資料庫時產生的。此時，您尚未對此文件進行任何更改，並將其保留為默認文件。 對於甲骨文 12.1.0.2： `$ cd $ORACLE_HOME/network/admin` `$ cat listener.ora` `ADR_BASE_L_VIS_001=/rdsdbdata/log/`	

任務	描述	所需技能
	SID_LIST_L_VIS_ 001=(SID_LIST = (SID_DESC = (SID_NAME = VIS)(GLOBAL_DBNAME = VIS) (ORACLE_HOME = / rdsdbbin/oracle))) L_VIS_001=(DESCR IPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521) (HOST = xx.xx.xx. xx))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = 127.0.0.1)))) SUBSCRIBE_FOR_NODE_DOW N_EVENT_L_VIS_001=OFF 對於 Oracle 19c：使用監聽器名稱L_<INSTANCE_NAME>_001 還原原始listener.ora 檔案。 $ cd $ORACLE_HOME/netwo rk/admin $ cp -rp /rdsdbdata/ config/listener.ora / rdsdbdata/config/ listener.ora_autoc onfig $ cp -rp /rdsdbdat a/config/listener. ora_orig /rdsdbdata/ config/listener.ora $ cat listener.ora	

任務	描述	所需技能
	SUBSCRIBE_FOR_ NODE_DOWN_EVENT_L_ VISCDB_001=OFF ADR_BASE_L_VISCDB_001 =/rdsdbdata/log/ USE_SID_AS_SERVICE_ L_VISCDB_001=ON L_VISCDB_001=(DESCRI PTION_LIST = (DESCRIPT ION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = xx.xx.xx. xx))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = 1521)(HOST = 127.0.0.1)))) SID_LIST_L_VISCDB_001= (SID_LIST = (SID_DESC = (SID_NAME = VISCDB)(G LOBAL_DBNAME = VISCDB) (ORACLE_HOME = / rdsdbbin/oracle))) 啟動標準 Amazon RDS 操作的接聽程式L_<INSTANCE_NAME>_001 ： $ lsnrctl stop $ lsnrctl start L_VISCDB_001 對於甲骨文 12.1.0.2： 編輯甲骨文電子商務套件環境文件以更改使用 Amazon RDS 自定義通用 TNS 文件的$TNS_ADMIN 路徑。環境檔案是在您之前執行自動設定時建立的。透過移除	

任務	描述	所需技能
	<CONTEXT_NAME> 後綴來編輯TNS_ADMIN 變數。 注意：您應該只在 Oracle 12.1.0.2 中編輯環境檔案，因為 19c 的預設本位目錄與 Amazon RDS 自訂的預設值相同。$ORACLE_HOME/network/admin 例如，在甲骨文 12.1.0.2 中，編輯以下文件： `$ vi $ORACLE_HOME/VIS_oebs-db01log.env` 從以下位置更改路徑： `TNS_ADMIN="/rdsdbbin/oracle/network/admin/VIS_oebs-db01log"` `export TNS_ADMIN` 至: `TNS_ADMIN="/rdsdbbin/oracle/network/admin"` `export TNS_ADMIN` 注意：每次執行自動設定時，您都必須重複此步驟，以確定使用正確的 TNS ifile。(僅限 12.1.0.2)。 對於甲骨文 19c：	

任務	描述	所需技能
	1. s_cdb_tnsadmin 將資料庫層內容變數的值變更為，<ORACLE_HOME>/network/admin 而非<ORACLE_HOME>/network/admin/<CONTEXT_NAME> 。 注意：請勿更新s_db_tnsadmin 上下文變數。保留為<ORACLE_HOME>/network/admin/<CONTEXT_NAME> 。 `$ . $ORACLE_HOME/VIS_oebs-db01log.env` `$ vi $CONTEXT_FILE` 2. 儲存您對的值所做的變更s_cdb_tnsadmin 。 s_db_tnsadmin 和的值看起來s_cdb_tnsadmin 應類似下列內容，PDB 名稱為VIS，資料庫節點邏輯名稱為oebs-db01log 。 `$ grep -i tns_admin $CONTEXT_FILE` `<TNS_ADMIN oa_var="s_db_tnsadmin">/rdsdbbin/oracle/network/admin/VIS_oebs-db01log</TNS_ADMIN>` `<CDB_TNS_ADMIN oa_var="s_cdb_tnsa`	

任務	描述	所需技能
	dmin">/rdsdbbin/or acle/network/admin</ CDB_TNS_ADMIN> 3. 在資料庫層上執行自動設定： $. $ORACLE_HOME/VISCD B_oebs-db01log.env $ export ORACLE_PD B_SID=VIS $ sqlplus "/ as sysdba" @$ORACLE_HOME/apps util/admin/adgrant s.sql APPS $ sqlplus "/ as sysdba" @$ORACLE_HOME/rdbms/ admin/utlrp.sql $. $ORACLE_HOME/VIS_o ebs-db01log.env $ echo $ORACLE_SID VIS $ cd $ORACLE_HOME/appsu til/scripts/$CONTE XT_NAME $ ./adautocfg.sh	

任務	描述	所需技能
設定 rdsdb 使用者的環境。	對甲骨文 19c 跳過此步驟。 對於甲骨文 12.1.0.2： 現在您已完成自動設定和 TNS 項目，您需要在`rdsdb`使用者設定檔中設定環境檔案來載入環境檔案。 更新`.bash_profile` 以呼叫 Oracle 電子商務套`.env`件資料庫檔案。您需要更新設定檔以確保已載入環境。此環境檔案是在您之前執行自動設定時建立的。 當您執行 Autoconfig 時，會建立下列範例環境檔案： `. /rdsdbbin/oracle/VIS_oebs-db01log.env` 作為用`rdsdb`戶： `cd $HOME` `vi .bash_profile` `export LD_LIBRARY_PATH=${ORACLE_HOME}/lib:${ORACLE_HOME}/ctx/lib` `export SHLIB_PATH=${ORACLE_HOME}/lib` `export PATH=$PATH:${ORACLE_HOME}/bin` `alias sql='rlwrap -c sqlplus / as sysdba'` `. ${ORACLE_HOME}/VIS_oebs-db01log.env`	DBA

任務	描述	所需技能
	注意：對於 Oracle 19c，您不需要在中載入 CDB 環境。`.bash_profile` 這是因為預設值設定`ORACLE_HOME` 為預設路徑`$ORACLE_HOME/network/admin` ，也就是 `rdsdb` (Oracle 本位目錄) 使用者的預設本位目錄。	

任務	描述	所需技能
為 Amazon RDS 自訂設定應用程式和資料庫。	完成甲骨文 12.1.0.2 和 19c 的前兩個步驟。每個版本的後續步驟都有所不同。 1. 在應用程式層上`/etc/hosts`，編輯資料庫的 IP 位址，並將其變更為 Amazon RDS 自訂 IP 位址： `xx.xx.xx.xx OEBS-db01.localdomain OEBS-db01 OEBS-db01log.localdomain OEBS-db01log` 因為您使用的是邏輯主機名稱，因此幾乎可以無縫地取代資料庫節點。 2. 在 Amazon RDS 自訂資料庫執行個體上，新增或修改指派給來源 EC2 執行個體的安全群組，以反映 Amazon RDS 自訂資料庫執行個體，以確保應用程式可存取節點。 對於甲骨文 12.1.0.2： 3. 執行自動設定。身為應用程式擁有者 (例如，`applmgr`)，執行： `$ cd $INST_TOP/admin/scripts` `$ ./adautocfg.sh` `AutoConfig completed successfully.`	DBA

任務	描述	所需技能
	4. 驗證項fnd_nodes 目： `SQL> select node_name from apps.fnd_nodes NODE_NAME ---------- ---------- ---------- ---------- ------ AUTHENTICATION OEBS-APP01LOG OEBS-DB01LOG` 5. 確認您可以登入並啟動應用程式： `$ ./adstrtal.sh` 對於甲骨文 19c： 1. 檢查 PDB 是否打開，並在需要時打開它： `SQL> show pdbs CON_ID CON_NAME OPEN MODE RESTRICTED ---------- --------- ---------- ---------- 2 PDB$SEED READ ONLY NO 3 VIS MOUNTED SQL> alter session set container=vis;`	

任務	描述	所需技能
	`` SQL> alter database open; SQL> alter database save state; ``` 2. 將連線測試為apps： ``` SQL> sqlplus apps/**** @vis ``` 3. 在資料庫層上執行自動設定： ``` $. $ORACLE_HOME/VIS_o ebs-db01log.env $ echo $ORACLE_SID VIS $ cd $ORACLE_HOME/appsu til/scripts/$CONTE XT_NAME $ ./adautocfg.sh ``` 4. 以應用程式擁有者身分在應用程式層上執行自動設定 (例如，applmgr)： ``` $ cd $INST_TOP/admin/sc ripts $ ./adautocfg.sh AutoConfig completed successfully. ``` 5. 驗證項fnd_nodes 目： ``` SQL> select node_name from apps.fnd_nodes	

任務	描述	所需技能
	NODE_NAME ---------- ---------- ---------- ---------- ------ AUTHENTICATION OEBS-APP01LOG OEBS-DB01LOG 6. 啟動應用程式： $ ./adstrtal.sh	

執行移轉後步驟

任務	描述	所需技能
繼續自動化以確認其有效。	使用下列 AWS CLI 命令恢復自動化： aws rds modify-db-instance \ --db-instance-identifier vis \ --automation-mode full \ 該數據庫現在由 Amazon RDS 自定義管理。 例如，如果接聽器或資料庫故障，Amazon RDS 自訂代理程式將重新啟動它們。若要測試這個問題，請執行下列命令。 停止監聽程式範例：	DBA

任務	描述	所需技能
	`-bash-4.2$ lsnrctl stop vis` 關閉資料庫範例： `SQL> shutdown immediate ;`	
驗證結構描述、連線和維護工作。	若要完成移轉，您必須至少執行下列工作。 • 執行FS_CLONE以同步修補程式檔案系統。 • 收集綱要統計資料。 • 確保外部界面和系統可以連接到新的 Amazon RDS 自訂資料庫。 • 設定備份和維護排程。 • 藉由發出切換來切換檔案系統，確認 AD 線上修補 (ADOP) 是否如預期般運作。	DBA

故障診斷

問題	解決方案
當您嘗試卸除記錄檔時，您收到 ORA-01624 錯誤。	如果您在嘗試卸除記錄檔時收到 ORA-01624 錯誤，請依照下列步驟執行。 發出下列命令，並等待您要刪除的記錄檔狀態為止INACTIVE。如需中狀態碼的詳細資訊V $log，請參閱 Oracle 說明文件。下面是一個示例命令及其輸出：

問題	解決方案
	<pre>SQL> select group#, status from v$log; GROUP# STATUS ---------- ---------------- 1 ACTIVE 2 CURRENT 3 UNUSED 4 UNUSED 5 UNUSED 6 UNUSED 6 rows selected.</pre> 在此範例中，記錄檔 1 為ACTIVE，因此您必須強制執行記錄檔切換三次，以確保您先前新增的第一個新記錄檔的狀態為CURRENT： <pre>SQL> alter system switch logfile; System altered. SQL> alter system switch logfile; System altered. SQL> alter system switch logfile; System altered.</pre> 請等待您要刪除的所有記錄檔 (如下列範例所示)，然後執行DROP LOGFILE命令。INACTIVE <pre>SQL> select group#, status from v$log; GROUP# STATUS ---------- ---------------- 1 INACTIVE 2 INACTIVE 3 INACTIVE 4 CURRENT 5 UNUSED 6 UNUSED 6 rows selected.</pre>

問題	解決方案
當您使用開啟資料庫時，您收到 ORA-00392 錯誤訊息`resetlogs` 。	如果您收到錯誤 ORA-00392: log xx 的執行緒 1 被清除，不允許作業，請執行下列命令 (以記錄檔編號取代`xx`)，然後重新執行 open `resetlogs` 命令： `SQL> alter database clear logfile group xx;` `SQL> alter database open resetlogs;`

問題	解決方案
您無法使用系統管理員或應用程式使用者連線至應用程式。	若要確認問題，請執行下列 SQL 查詢： `SQL> select dbms_java.get_jdk_ version() from dual; select dbms_java.get_jdk_version() from dual ERROR at line 1: ORA-29548: Java system class reported: release of Java system classes in the database (19.0.0.0.220719 1.8) does not match that of the oracle executabl e (19.0.0.0.0 1.8)` 根本原因：來源資料庫已套用多個修補程式，但 Amazon RDS 自訂DB_HOME是新安裝，或 CEV 未包含所有修補程式，因為您在建立 CEV 時並未使用必要的 RSU 修補程式，例如 OJVM。 若要驗證這一點，請檢查、和中是否列出來源修補程式詳細資料`opatch -lsinventory` 。`$ORACLE_HOME/sqlpath $ORACLE_HOME/.patch_storage` 參考：數據跟踪-詳細失敗，並顯示錯誤：」補丁 xxxxxx：存檔的修補程序目錄為空」（文檔 ID 2235541.1） 修正：將遺失的修補程式相關檔案從來源 (`$ORACLE_HOME/sqlpatch/`) 複製到 Amazon RDS 自訂 (`$ORACLE_HOME/sqlpatch/`)，然後重新執行。`./datapatch -verbose` 例如： `-bash-4.2$ cp -rp 18793246 20204035 20887355 22098146 22731026 $ORACLE_H OME/sqlpatch/`

問題	解決方案
	或者，您可以在 CDB 和 PDB 上執行下列命令來使用因應措施： `@?/javavm/install/update_javavm_db.sql` 然後在 PDB 上執行下列命令： `sql> alter session set container=vis;` `@?/javavm/install/update_javavm_db.sql` 現在再次運行測試： `SQL> select dbms_java.get_jdk_version() from dual;`

相關資源

- 使用 Amazon RDS 自定義（Amazon RDS 文檔）
- 適用於甲骨文的 Amazon RDS 自訂 — 資料庫環境中的新控制功能 (AWS 新聞部落格)
- 整合 Amazon RDS 自訂與 Amazon EFS (AWS 資料庫部落格)
- 遷移 AWS 上的甲骨文電子商務套件 (AWS 白皮書)
- AWS 上的甲骨文電子商務套件架構 (AWS 白皮書)
- 在 Amazon RDS 上為甲骨文電子商務套件設置 HA/DR 架構，並使用有效的待命數據庫（AWS Prescriptive Guidance）

其他資訊

維護作業

使用新修正程式修正 Oracle 電子商務套件資料庫

由於 bin 磁碟區 (`/rdsdbbin`) 是 out-of-place 升級，因此在 CEV 升級期間會捨棄 bin 磁碟區的內容。因此，您必須先建立`appsutil`目錄副本，才能使用 CEV 執行任何升級。

在來源 Amazon RDS 自訂執行個體上，在升級 CEV 之前，請先備份。$ORACLE_HOME/appsutil

注意：此範例使用 NFS 磁碟區。但是，您可以使用 Amazon Simple Storage Service (Amazon S3) 的副本。

1. 建立一個目錄以將應用程式存放在來源 Amazon RDS 自訂執行個體上：

```
$ mkdir /RMAN/appsutil.preupgrade
```

2. 焦油並複製到 Amazon EFS 卷：

```
$ tar cvf /RMAN/appsutil.preupgrade appsutil
```

3. 請確認 tar 檔案是否存在：

```
$ bash-4.2$ ls -l /RMAN/appsutil.preupgrade
-rw-rw-r-- 1 rdsdb rdsdb 622981120 Feb  8 20:16 appsutil.tar
```

4. 請按照 Amazon RDS 說明文件中升級 RDS 自訂資料庫執行個體中的指示，升級至最新的 CEV (已建立先決條件 CEV)。

您也可以使用 OPATCH 直接進行修補。請參閱 Amazon RDS 文件中適用於 Oracle 升級的 RDS 自訂需求和注意事項一節。

注意：在 CEV 修正處理作業期間，主機機器的 IP 位址不會變更。這個程序會執行 out-of-place 升級，而且在啟動期間，新的 bin 磁碟區會附加到相同的執行個體上。

將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制

創建者：古普塔高拉夫 (AWS)

環境：生產	資料來源：Amazon EC2	目標：Amazon RDS 自定義
R 類型：重新平台	工作量：甲骨文	技術：移轉；基礎架構；資料庫
AWS 服務：Amazon RDS; Amazon S3; AWS Secrets Manager; Amazon EFS		

Summary

Oracle PeopleSoft 是針對整個企業流程的企業資源規劃 (ERP) 解決方案。 PeopleSoft 具有三層架構：客戶端，應用程序和數據庫。 PeopleSoft 可以在 Amazon 關係數據庫服務（亞馬遜 RDS）上運行。現在，您也可以在 Amazon RDS 自訂 PeopleSoft 上執行，這可讓您存取基礎作業系統。

Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。當您將 Oracle 資料庫遷移到 Amazon RDS 客製化時，Amazon Web Services (AWS) 可以管理備份任務和高可用性，同時您可以專注於維護 PeopleSoft 應用程式和功能。如需移轉時要考慮的關鍵因素，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

此模式著重於使用 Oracle 復原管理器 (RMAN) 備份將亞馬遜彈性運算雲端 (Amazon EC2) 上的 PeopleSoft 資料庫遷移到 Amazon RDS 自訂的步驟。它在 EC2 執行個體和 Amazon RDS 自訂之間使用 Amazon 彈性檔案系統 (Amazon EFS) 共用檔案系統，不過您也可以使用 Amazon FSx 或任何共用磁碟機。病毒碼使用 RMAN 完整備份（有時稱為層級 0 備份）。

先決條件和限制

先決條件

- 甲骨文 19C 版來源資料庫，使用甲骨文 7、甲骨文 8、紅帽企業 Linux 7 或 RHEL 8 在 Amazon EC2 上執行。在此模式的範例中，來源資料庫名稱為`FSDM092`，但這不是必要條件。

  備註：您也可以將此模式與內部部署 Oracle 來源資料庫搭配使用。您必須在內部部署網路和虛擬私人雲端 (VPC) 之間具有適當的網路連線能力。

- PeopleSoft 9.2 示範執行個體。
- 單一 PeopleSoft 應用程式層。不過，您可以調整此模式來處理多個應用程式層級。
- Amazon RDS 自訂已設定至少 8 GB 的交換空間。

限制

此模式不支持以下配置：

- 將資料庫`ARCHIVE_LAG_TARGET`參數設定為 60—7200 範圍以外的值
- 停用資料庫執行個體記錄模式 (`NOARCHIVELOG`)
- 關閉 EC2 執行個體的 Amazon 彈性區塊存放區 (亞馬遜 EBS) 優化屬性
- 修改連接到 EC2 執行個體的原始 EBS 磁碟區
- 新增 EBS 磁碟區或將磁碟區類型從 gp2 變更為 gp3
- 變更`LOG_ARCHIVE_FORMAT`參數的副檔名格式 (需要`*.arc`)
- 多工或更改控制文件的位置和名稱（必須是）`/rdsdbdata/db/*DBNAME*/controlfile/control-01.ctl`

如需有關這些組態和其他不受支援組態的其他資訊，請參閱 Amazon RDS 文件。

產品版本

如需 Amazon RDS 自訂支援的 Oracle 資料庫版本和執行個體類別，請參閱適用於 Oracle 的 Amazon RDS 自訂需求和限制。

架構

目標技術堆疊

- Application Load Balancer
- Amazon EFS
- Amazon RDS Custom for Oracle
- AWS Secrets Manager
- Amazon Simple Storage Service (Amazon S3)

目標架構

下列架構圖表示在 AWS 的單一可用區域中執行的 PeopleSoft 系統。應用程式層可透過應用程式負載平衡器存取。應用程式和資料庫都位於私有子網路中，而 Amazon RDS 自訂和 Amazon EC2 資料庫執行個體則使用 Amazon EFS 共用檔案系統來存放和存取 RMAN 備份檔案。Amazon S3 用於創建自定義 RDS Oracle 引擎和存儲重做日誌元數據。

工具

工具

AWS 服務

- Amazon RDS Custom for Oracle 是一種受管資料庫服務，適用於需要存取基礎作業系統和資料庫環境的舊式、自訂和封裝應用程式。它會自動執行資料庫管理工作，例如備份和高可用性。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。此模式使用 Amazon EFS 共用檔案系統來存放和存取 RMAN 備份檔案。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。在此模式中，您可以從 Secrets Manager 擷取資料庫使`ADMIN`用者密碼，以建立`RDSADMIN`和使用者，以及變更`sys`和`system`密碼。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。此模式使用應用程式負載平衡器。

其他工具

- 「Oracle 復原管理程式 (RMAN)」提供 Oracle 資料庫的備份與復原支援。此模式使用 RMAN 在 Amazon RDS 自訂上還原的 Amazon EC2 上執行來源 Oracle 資料庫的熱備份。

最佳實務

- 對於資料庫初始化參數，請自訂 Amazon RDS 自訂資料庫執行個體提供的標準 pfile， PeopleSoft 而不是使用 Oracle 來源資料庫中的 spfile。這是因為在 Amazon RDS 自訂中建立僅供讀取複本時，空格和註解會導致問題。如需有關資料庫初始化參數的詳細資訊，請參閱 Oracle 客戶 Support 中心注意事項 1100831.1 (需要 Oracle 客戶 Support 部帳戶)。

- Amazon RDS 自定義默認使用 Oracle 自動內存管理。如果您想要使用 Hugemem 核心，可以將 Amazon RDS 自訂設定為改用自動共用記憶體管理。
- 依預設，保持`memory_max_target`參數為啟用狀態。架構會在背景中使用此功能來建立僅供讀取複本。
- 啟用「Oracle 倒溯資料庫」。在容錯移轉 (非切換) 測試案例中恢復待命時，此功能非常有用。

史诗

設定資料庫執行個體和檔案系統

任務	描述	所需技能
建立資料庫執行個體。	在 Amazon RDS 主控台中，使用名為 FSDMO92 的資料庫名稱 (或您的來源資料庫名稱) 建立適用於 Oracle 資料庫的 Amazon RDS 自訂執行個體。 如需指示，請參閱 AWS 文件中的使用 Amazon RDS 自訂和 Oracle 專用的 Amazon RDS 自訂 — 資料庫環境中的新控制功能部落格文章。這可確保將資料庫名稱設定為與來源資料庫相同的名稱。（如果保持空白，EC2 實例和數據庫名稱將設置為`ORCL`。）	DBA

對來源 Amazon EC2 資料庫執行 RMAN 完整備份

任務	描述	所需技能
建立備份指令碼。	建立 RMAN 備份指令碼，將資料庫備份到您所掛載的 Amazon EFS 檔案系統 (`/efs`在以下範例中)。您可以使	DBA

任務	描述	所需技能
	用範例程式碼，或執行其中一個現有的 RMAN 指令碼。 `#!/bin/bash` `Dt=`date +'%Y%m%d-%H%M'` `BACKUP_LOG="rman-${ORACLE_SID}-$Dt"` `export TAGDATE=`date +%Y%m%d%H%M`;` `LOGPATH=/u01/scripts/logs` `rman target / >> $LOGPATH/rman-${ORACLE_SID}-$Dt << EOF` `SQL "ALTER SYSTEM SWITCH LOGFILE";` `SQL "ALTER SESSION SET NLS_DATE_FORMAT="DD.MM.YYYY HH24:MI:SS"";` `RUN` `{` `ALLOCATE CHANNEL ch11 TYPE DISK MAXPIECESIZE 5G;` `ALLOCATE CHANNEL ch12 TYPE DISK MAXPIECESIZE 5G;` `BACKUP AS COMPRESSED BACKUPSET FULL DATABASE FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_FULL' ;` `SQL "ALTER SYSTEM ARCHIVE LOG CURRENT";` `BACKUP FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_ARCHIVE' ARCHIVELOG ALL DELETE ALL INPUT ;`	

任務	描述	所需技能
	BACKUP CURRENT CONTROLFILE FORMAT '/efs/rman_backup/FSCM/%d_%T_%s_%p_CONTROL'; } EXIT; EOF	
執行備份指令碼。	若要執行 RMAN 備份命令檔，請以「Oracle 本位目錄使用者」的身分登入，然後執行命令檔。 $ chmod a+x rman_backup.sh $ ./rman_backup.sh &	DBA

<table>
<tr><th>任務</th><th>描述</th><th>所需技能</th></tr>
<tr><td>檢查錯誤並記下備份檔案的名稱。</td><td>檢查 RMAN 記錄檔是否有錯誤。如果一切正常，請運行以下命令列出控制文件的備份。

<pre>
RMAN> list backup of
 controlfile;

using target database
 control file instead of
 recovery catalog
</pre>

請注意輸出檔案的名稱。

<pre>
List of Backup Sets
===================

BS Key Type LV Size
     Device Type Elapsed
 Time Completion Time
------- ---- --
 ---------- ---------
-- ------------
 ----
12   Full   21.58M
   DISK   00:00:01
    13-JUL-22
        BP Key: 12
  Status: AVAILABLE
 Compressed: NO Tag:
 TAG20220713T150155
        Piece Name: /
efs/rman_backup/F
SCM/FSDM092_202207
13_12_1_CONTROL
 Control File Included:
 Ckp SCN: 165591599
85898   Ckp time: 13-
JUL-22
</pre></td><td>DBA</td></tr>
</table>

任務	描述	所需技能
	當您在 Amazon RDS 自訂上還原資料庫`/efs/rman_backup/FSCM/FSDMO92_20220713_12_1_CONTROL` 時，將使用備份控制檔案。	

關閉來源應用程式層

任務	描述	所需技能
關閉應用程式。	若要關閉來源應用程式層，請使用`psadmin`公用程式或命`psadmin`令列公用程式。 1. 要關閉 Web 服務器，請運行以下命令。 `psadmin -w shutdown -d "webserver domain name"` 2. 若要關閉應用程式伺服器，請執行下列命令。 `psadmin -c shutdown -d "application server domain name"` 3. 若要關閉處理程序排程器，請執行下列命令。 `psadmin -p stop -d "process scheduler domain name"`	DBA，PeopleSoft 系統管理員

設定目標 Amazon RDS 自訂資料庫

任務	描述	所需技能
安裝 nFS 公用程式 rpm 套件。	若要安裝nfs-utils rpm套件，請執行下列命令。 `$ yum install -y nfs-utils`	DBA
掛載 EFS 儲存區。	從 Amazon EFS 主控台頁面取得 Amazon EFS 掛載命令。使用網路檔案系統 (NFS) 用戶端，在 Amazon RDS 執行個體上掛載 EFS 檔案系統。 `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-xxxxxxxxxx.efs.eu-west-1.amazonaws.com:/ /efs` `sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-xxxxxxxxxx.efs.eu-west-1.amazonaws.com:/ /efs`	DBA

卸除入門資料庫並建立儲存資料庫檔案的目錄

任務	描述	所需技能
暫停自動化模式。	在繼續執行後續步驟之前，您必須暫停 Amazon RDS 自訂資料庫執行個體上的自動化模式，以確保自動化不會干擾 RMAN 還原活動。 您可以使用 AWS 主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 命令暫停自動化 (請確定您已先設定 AWS CLI)。 `aws rds modify-db-instance \ --db-instance-id entifier peoplesoft-fscm-92 \ --automation-mode all-paused \ --resume-full-au tomation-mode-minute 360 \ --region eu-west-1` 當您指定暫停的持續時間時，請確定您預留足夠的時間進行 RMAN 還原。這取決於來源資料庫的大小，因此請相應地修改 360 值。 此外，請確定暫停自動化的總時間不會與資料庫的備份或維護時段重疊。	DBA

任務	描述	所需技能
建立和修改參數檔案 PeopleSoft	若要為其建立和修改 pfile PeopleSoft，請使用以 Amazon RDS 自訂資料庫執行個體建立的標準 pfile。加入您需要的參數 PeopleSoft。 1. 執行下列命令切換至`rds user rdsdb`。 `$ sudo su - rdsdb` 2. 登入起始資料庫上的 SQL*Plus，然後執行下列命令來建立 pfile。 `SQL> create pfile from spfile;` 這會在`$ORACLE_HOME/dbs`中建立 pfile。 3. 做一個這個 pfile 的備份。 4. 編輯 pfile 以新增或更新 PeopleSoft參數。 `*._gby_hash_aggregation_enabled=false` `*._unnest_subquery=false` `*.nls_language='AMERICAN'` `*.nls_length_semantics='CHAR'`	DBA

任務	描述	所需技能
	`` *.nls_territ ory='AMERICA' *.open_cursors=1000 *.db_files=1200 *.undo_tablespace=' UNDOTBS1' ``` PeopleSoft 您可以在 Oracle 客戶 Support 中心注意事項 1100831.1 中找到相關參數。 5. 從 pfile 中移除 spfile 參考。 ``` *.spfile='/rdsdbbi n/oracle/dbs/spfil eFSDMO92.ora' ```	
卸除入門資料庫。	若要刪除現有的 Amazon RDS 自訂資料庫，請使用下列程式碼。 ``` $ sqlplus / as sysdba SQL> shutdown immediate ; SQL> startup mount exclusive restrict; SQL> drop database; SQL> exit ```	

任務	描述	所需技能
從備份還原 Amazon RDS 自訂資料庫。	使用下列指令碼還原資料庫。指令碼會先還原控制檔，然後從儲存在 EFS 掛載上的備份片段還原整個資料庫。 `#!/bin/bash` `Dt=`date +'%Y%m%d-%H%M'` `BACKUP_LOG="rman-${ORACLE_SID}-$Dt"` `export TAGDATE=`date +%Y%m%d%H%M`;` `LOGPATH=/rdsdbdata/scripts/logs` `rman target / >> $LOGPATH/rman-${ORACLE_SID}-$Dt << EOF` `restore controlfile from "/efs/rman_backup/FSCM/FSDMO92_20220713_12_1_CONTROL";` `alter database mount;` `run` `{` `set newname for database to '/rdsdbdata/db/FSDMO92_A/datafile/%f_%b';` `SET NEWNAME FOR TEMPFILE 1 TO '/rdsdbdata/db/FSDMO92_A/datafile/%f_%b';` `RESTORE DATABASE;` `SWITCH DATAFILE ALL;` `SWITCH TEMPFILE ALL;` `RECOVER DATABASE;` `}` `EOF`	DBA

任務	描述	所需技能
	sqlplus / as sysdba >> $LOGPATH/rman-${ORACLE_SID}-$Dt<<-EOF ALTER DATABASE RENAME FILE '/u01/psoft/db/oradata/FSDMO92/redo01.log' TO '/rdsdbdata/db/FSDMO92_A/onlinelog/redo01.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/oradata/FSDMO92/redo02.log' TO '/rdsdbdata/db/FSDMO92_A/onlinelog/redo02.log'; ALTER DATABASE RENAME FILE '/u01/psoft/db/oradata/FSDMO92/redo03.log' TO '/rdsdbdata/db/FSDMO92_A/onlinelog/redo03.log'; alter database clear unarchived logfile group 1; alter database clear unarchived logfile group 2; alter database clear unarchived logfile group 3; alter database open resetlogs; EXIT EOF	

從密碼 Secrets Manager 擷取密碼、建立使用者和變更密碼

任務	描述	所需技能
從密碼管理員擷取密碼。	您可以使用 AWS 主控台或 AWS CLI 執行此步驟。下列步驟顯示主控台的指示。 1. 登入 AWS 管理主控台並開啟 Amazon RDS 主控台。 2. 在導覽窗格中，選擇「資料庫」，然後選取 Amazon RDS 資料庫。 3. 選擇「組態」頁籤，並記下執行處理的資源 ID。它將採用格式db-<ID>（例如，db-73GJNHLGDNZNDOXNWXSECUW6LE ）。 4. 開啟 Secrets Manager 主控台。 5. 選擇與名稱相同的密碼do-not-delete-custom-<resource_id> ，其中resource-id 指的是您在步驟 3 中記下的資源 ID。 6. 選擇 Retrieve secret value (擷取秘密值)。 sys、system、和admin使用者的密碼將相同。rdsadmin	DBA
建立 RDSADMIN 使用者。	RDSADMIN是用於監控和協調 Amazon RDS 自訂資料庫執行個體的資料庫使用者。由於入門資料庫已卸除，且目	DBA

任務	描述	所需技能
	標資料庫已使用 RMAN 從來源還原，因此您必須在還原操作後重新建立此使用者，以確保 Amazon RDS 自訂監控能如預期般運作。您也必須為RDSADMIN使用者建立個別的設定檔和表格空間。 1. 在 SQL 提示字元中輸入下列命令。 `SQL> set echo on feedback on serverout on` `SQL> @?/rdbms/admin/utlpwdmg.sql` `SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ ATTEMPTS UNLIMITED PASSWORD_LIFE_TIME UNLIMITED PASSWORD_VERIFY_F UNCTION NULL;` 2. 建立設定檔RDSADMIN。 `SQL> set echo on feedback on serverout on` `SQL> alter session set "_oracle_script"=t rue;` `SQL> CREATE PROFILE RDSADMIN LIMIT COMPOSITE_LIMIT UNLIMITED`	

任務	描述	所需技能
	SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION UNLIMITED CPU_PER_CALL UNLIMITED LOGICAL_READS_PER _SESSION UNLIMITED LOGICAL_READS_PER _CALL UNLIMITED IDLE_TIME UNLIMITED CONNECT_TIME UNLIMITED PRIVATE_SGA UNLIMITED FAILED_LOGIN_ATTE MPTS 10 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_REUSE_TIME UNLIMITED PASSWORD_REUSE_MAX UNLIMITED PASSWORD_VERIFY_F UNCTION NULL PASSWORD_LOCK_TIME 86400/86400 PASSWORD_GRACE_TIME 604800/86400; 3. 建立表RDSADMIN格空間。 SQL> CREATE BIGFILE TABLESPACE rdsadmin '/rdsdbdata/db/FSD MO92_A/datafile/rd sadmin.dbf' DATAFILE SIZE 7M AUTOEXTEND ON NEXT 1m LOGGING ONLINE PERMANENT BLOCKSIZE 8192 EXTENT MANAGEMEN	

任務	描述	所需技能
	T LOCAL AUTOALLOCATE DEFAULT NOCOMPRES S SEGMENT SPACE MANAGEMENT AUTO; 4. 建立使RDSADMIN用者。將RDSADMIN密碼取代為您先前從密碼管理員取得的密碼。 SQL> CREATE USER rdsadmin IDENTIFIED BY xxxxxxxxxx DEFAULT TABLESPACE rdsadmin TEMPORARY TABLESPACE TEMP profile rdsadmin ; 5. 授與權限給 RDSADMIN。 SQL> GRANT "CONNECT" TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT "RESOURCE " TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT "DBA" TO RDSADMIN; SQL> GRANT "SELECT_C ATALOG_ROLE" TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT ALTER SYSTEM TO RDSADMIN; SQL> GRANT UNLIMITED TABLESPACE TO RDSADMIN; SQL> GRANT SELECT ANY TABLE TO RDSADMIN;	

任務	描述	所需技能
	SQL> GRANT ALTER DATABASE TO RDSADMIN; SQL> GRANT ADMINISTER DATABASE TRIGGER TO RDSADMIN; SQL> GRANT ANY OBJECT PRIVILEGE TO RDSADMIN WITH ADMIN OPTION; SQL> GRANT INHERIT ANY PRIVILEGES TO RDSADMIN; SQL> ALTER USER RDSADMIN DEFAULT ROLE ALL; 6. Set the SYS, SYSTEM, and DBSNMP user profiles to RDSADMIN. SQL> set echo on feedback on serverout on SQL> alter user SYS profile RDSADMIN; SQL> alter user SYSTEM profile RDSADMIN; SQL> alter user DBSNMP profile RDSADMIN;	

任務	描述	所需技能
建立主要使用者。	因為啟動器資料庫已經卸除，而且目標資料庫是使用 RMAN 從來源還原的，所以您必須重新建立 master 使用者。在此範例中，主要使用者名稱為 admin。 `SQL> create user admin identified by <password>;` `SQL> grant dba to admin`	DBA
變更系統密碼。	使用您從密碼管理員擷取的密碼來變更系統密碼。 `SQL> alter user sys identified by xxxxxxxxxx;` `SQL> alter user system identified by xxxxxxxxxx;` 如果您未變更這些密碼，Amazon RDS Custom 會顯示錯誤訊息：「資料庫監控使用者或使用者登入資料已變更」。	DBA

為 Amazon RDS 自定義和配置 TNS 條目 PeopleSoft

任務	描述	所需技能
設定 tnsname 檔案。	若要從應用程式層連線到資料庫，請設定tnsnames.ora 檔案，以便從應用程式層	DBA

任務	描述	所需技能
	連線到資料庫。在下面的例子中，你可以看到有一個軟鏈接到該tnsnames.ora 文件，但默認情況下文件是空的。 `$ cd /rdsdbbin/oracle/network/admin` `$ ls -ltr` `-rw-r--r-- 1 rdsdb database 1536 Feb 14 2018 shrept.lst` `lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 listener.ora -> /rdsdbdata/config/listener.ora` `lrwxrwxrwx 1 rdsdb database 28 Apr 5 13:19 sqlnet.ora -> /rdsdbdata/config/sqlnet.ora` `lrwxrwxrwx 1 rdsdb database 30 Apr 5 13:19 tnsnames.ora -> /rdsdbdata/config/tnsnames.ora` 1. 建立項tsnames.ora 目。由於 Amazon RDS 自動化剖析檔案的方式，您必須確定項目不包含任何空格、註解或額外的行。否則，使用某些 API（例如 create-db-instance-read-replica）時可能會遇到問題。 2. 根據您的 PeopleSoft 資料庫需求取代連接埠、主機和	

任務	描述	所需技能
	SID。使用下列程式碼做為範例。 `$ vi tnsnames.ora` `FSDMO92=(DESCRIP` `TION = (ADDRESS_` `LIST = (ADDRESS =` `(PROTOCOL = TCP)(HOST` `= x.x.x.x)(PORT =` `1521))) (CONNECT_` `DATA = (SERVER =` `DEDICATED) (SID =` `FSDMO92)))` 3. 若要確認可以連線到 PeopleSoft 資料庫，請執行下列命令。 `$ tnsping FSDMO92` `TNS Ping Utility` `for Linux: Version` `19.0.0.0.0 -` `Production on 14-` `JUL-2022 10:16:45` `Copyright (c) 1997,` `2021, Oracle. All` `rights reserved.` `Used parameter files:` `/rdsdbbin/oracle/net` `work/admin/sqlnet.` `ora` `Used TNSNAMES adapter` `to resolve the alias` `Attempting to` `contact (DESCRIPT` `ION = (ADDRESS_`	

任務	描述	所需技能
	LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = x.x.x.x)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SID = FSDMO92))) OK (0 msec)	

建立 spfile 軟連結

任務	描述	所需技能
建立 spfile 軟體連結。	1. 若要在該位置建立 spfile/rdsdbdata/admin/FSDMO92/pfile ，請執行下列命令。 `SQL> create spfile='/rdsdbdata/admin/FSDMO92/pfile/spfileFSDMO92.ora' from pfile;` 2. 瀏覽$ORACLE_HOME/dbs並建立 spfile 的軟連結。 `ln -s '/rdsdbdata/admin/FSDMO92/pfile/spfileFSDMO92.ora' spfileFSDMO92.ora` 3. 建立此檔案之後，您可以使用 spfile 關閉並啟動資料庫。	DBA

執行移轉後步驟

任務	描述	所需技能
驗證結構描述、連線和維護工作。	若要完成移轉，請執行下列工作。 • 收集綱要統計資料。 • 確保 PeopleSoft 應用程式層可以連接到新的 Amazon RDS 自訂資料庫。 • 設定備份與維護排程。	DBA

相關資源

- 與 Amazon RDS 自定義工作
- 適用於甲骨文的 Amazon RDS 自訂 — 資料庫環境中的新控制功能 (部落格文章)
- 整合 Amazon RDS 自訂甲骨文與 Amazon EFS (部落格文章)
- 將 Amazon RDS 設定為甲骨文 PeopleSoft 資料庫 (AWS 白皮書)

將甲骨文功能遷移到 AWS 上的 PostgreSQL

由拉凱什拉格夫 (AWS) 和拉麥絲帕瑟里 (AWS) 創建

環境：PoC 或試點	來源：甲骨文數據庫	目標：AWS 上的資料庫
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon Aurora; Amazon RDS; AWS SCT; AWS CLI		

Summary

此模式描述了將 Oracle 資料庫中的ROWID虛擬資料欄功能遷移到 Amazon 關聯式資料庫服務 (Amazon RDS) 中的 PostgreSQL 資料庫的選項，適用於 PostgreSQL、Amazon Aurora PostgreSQL 相容版本或亞馬遜 Elastic Compute Cloud (Amazon EC2)。

在 Oracle 資料庫中，ROWID虛擬資料欄是資料表中資料列的實體位址。此偽列用於唯一標識行，即使主鍵不存在於表中。PostgreSQL 有一個類似的偽列ctid，但它不能用作. ROWID 如 PostgreSQL 文檔中所述，如果更新或每個VACUUM進程之後，ctid可能會更改。

有三種方法可以在 PostgreSQL 中創建ROWID虛擬列功能：

- 使用主索引鍵欄，而不是識別資料表中的資料列。ROWID
- 在表中使用邏輯主/唯一鍵（可能是複合鍵）。
- 添加一個包含自動生成值的列，並使其成為模擬的主/唯一鍵。ROWID

此模式會引導您完成所有三種實作，並說明每個選項的優缺點。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 程序語 PostgreSQL 編碼專業知識
- 來源 Oracle 資料庫

- Amazon RDS for PostgreSQL 或 Aurora 與 PostgreSQL 相容的叢集，或用於託管 PostgreSQL 資料庫的 EC2 執行個體

限制

- 此模式提供了功能的解決方法。ROWIDPostgreSQL 不提供與甲骨文數據庫ROWID中的相同內容。

產品版本

- PostgreSQL 11.9 或更高版本

架構

源, 技術, 堆棧

- Oracle Database

目標技術堆疊

- Aurora PostgreSQL 相容的亞馬遜 RDS，或具有 PostgreSQL 資料庫的 EC2 執行個體

實作選項

有三個選項可以解決 PostgreSQL 缺乏ROWID支援的問題，這取決於您的資料表是否具有主索引鍵或唯一索引、邏輯主索引鍵或身分識別屬性。您的選擇取決於您的專案時間表、目前的移轉階段，以及應用程式和資料庫程式碼的相依性。

選項	Description	優點	缺點
主鍵或唯一索引	如果你的 Oracle 表有一個主鍵，你可以使用這個鍵的屬性來唯一標識一行。	• 不依賴於專有的數據庫功能。 • 對效能的影響最小，因為主索引鍵欄位已編製索引。	• 需要變更應用程式和資料庫程式碼，這些程式碼必ROWID須依賴於切換至主索引鍵欄位

邏輯主/唯一鍵	如果您的 Oracle 表具有邏輯主鍵，則可以使用此鍵的屬性唯一標識行。邏輯主索引鍵是由可唯一識別資料列的屬性或一組屬性所組成，但不會透過條件約束在資料庫上強制執行。	• 不依賴於專有的數據庫功能。	• 需要變更應用程式和資料庫程式碼，這些程式碼必ROWID須依賴於切換至主索引鍵欄位 • 如果邏輯主索引鍵的屬性未編製索引，則會對效能造成重大影響。不過，您可以新增唯一索引來防止效能問題。
識別屬性	如果您的 Oracle 表沒有主鍵，則可以將其他字段創建為GENERATED ALWAYS AS IDENTITY。每當資料插入資料表時，這個屬性就會產生唯一的值，因此它可用來唯一識別資料處理語言 (DML) 作業的資料列。	• 不依賴於專有的數據庫功能。 • PostgreSQL 資料庫會填入屬性並維護其唯一性。	• 需要變更依賴於切換到識別屬性的應用程式和資料庫程式碼。ROWID • 如果其他欄位未編製索引，則會對效能產生重大影響。但是，您可以添加索引以防止性能問題。

工具

- 適用於 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 PostgreSQL 關聯式資料庫。
- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。在這種模式中，您可以使用 AWS CLI 透過 PGAdmin 執行 SQL 命令。
- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。

史诗

識別來源表格

任務	描述	所需技能
識別使用ROWID屬性的 Oracle 表格。	使用 AWS 結 Schema Conversion Tool (AWS SCT) 來識別具有ROWID功能的 Oracle 表格。如需詳細資訊，請參閱 AWS SCT 文件。 —或— 在 Oracle 中，使用檢DBA_TAB_COLUMNS 視表來識別具有ROWID屬性的表格。這些欄位可用來儲存 10 位元組英數字元。確定用法並將其轉換為適當的VARCHAR字段。	DBA 或開發人員
識別參考這些資料表的程式碼。	使用 AWS SCT 產生遷移評估報告，以識別受影響的程序。ROWID如需詳細資訊，請參閱 AWS SCT 文件。 —或— 在來源 Oracle 資料庫中，使用dba_source 表格的文字欄	DBA 或開發人員

任務	描述	所需技能
	位來識別使用ROWID功能的物件。	

確定主鍵用法

任務	描述	所需技能
識別沒有主鍵的表。	在來源 Oracle 資料庫中，用DBA_CONSTRAINTS 來識別沒有主索引鍵的資料表。此資訊將協助您決定每個表格的策略。例如： `select dt.* from dba_tables dt where not exists (select 1 from all_constraints ct where ct.owner = Dt.owner and ct.table_name = Dt.table_name and ct.constraint_type = 'P' ) and dt.owner = '{schema}'`	DBA 或開發人員

識別並套用解決方案

任務	描述	所需技能
對具有已定義或邏輯主索引鍵的資料表套用變更。	變更 [其他資訊] 區段中顯示的應用程式和資料庫程式碼，以使用唯一的主索引鍵或邏輯主索引鍵來識別資料表中的資料列。	DBA 或開發人員
向沒有已定義或邏輯主索引鍵的資料表新增其他欄位。	新增類型的屬性`GENERATED ALWAYS AS IDENTITY`。變更 [其他資訊] 區段中顯示的應用程式和資料庫程式碼。	DBA 或開發人員
如有必要，請新增索引。	將索引添加到其他字段或邏輯主鍵以提高 SQL 性能。	DBA 或開發人員

相關資源

- PostgreSQL 民共和国 PostgreSQL 文件
- 產生的資料行 PostgreSQL 件集)
- 偽資料欄 (Oracle 文件集)

其他資訊

以下各節提供甲骨文和 PostgreSQL 的代碼示例來說明這三種方法。

案例 1：使用主唯一索引鍵

在下列範例中，您會建立以`emp_id`做為主索引鍵的資料表`testrowid_s1`。

甲骨文代碼：

```
create table testrowid_s1 (emp_id integer, name varchar2(10), CONSTRAINT testrowid_pk
 PRIMARY KEY (emp_id));
INSERT INTO testrowid_s1(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s1(emp_id,name) values (2,'empname2');
```

```
INSERT INTO testrowid_s1(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s1(emp_id,name) values (4,'empname4');
commit;

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3pAAAAAAAMOAAA          1 empname1
AAAF3pAAAAAAAMOAAB          2 empname2
AAAF3pAAAAAAAMOAAC          3 empname3
AAAF3pAAAAAAAMOAAD          4 empname4

UPDATE testrowid_s1 SET name = 'Ramesh' WHERE rowid = 'AAAF3pAAAAAAAMOAAB' ;
commit;

SELECT rowid,emp_id,name FROM testrowid_s1;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3pAAAAAAAMOAAA          1 empname1
AAAF3pAAAAAAAMOAAB          2 Ramesh
AAAF3pAAAAAAAMOAAC          3 empname3
AAAF3pAAAAAAAMOAAD          4 empname4
```

PostgreSQL：

```
CREATE TABLE public.testrowid_s1
(
    emp_id integer,
    name character varying,
    primary key (emp_id)
);

insert into public.testrowid_s1 (emp_id,name) values
(1,'empname1'),(2,'empname2'),(3,'empname3'),(4,'empname4');

select emp_id,name from testrowid_s1;
 emp_id |   name
--------+----------
      1 | empname1
      2 | empname2
      3 | empname3
      4 | empname4
```

```
update testrowid_s1 set name = 'Ramesh' where emp_id = 2 ;

select emp_id,name from testrowid_s1;
 emp_id |   name
--------+----------
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

案例 2：使用邏輯主鍵

在下列範例中，您建立的資料表testrowid_s2emp_id做為邏輯主索引鍵。

甲骨文代碼：

```
create table testrowid_s2 (emp_id integer, name varchar2(10) );
INSERT INTO testrowid_s2(emp_id,name) values (1,'empname1');
INSERT INTO testrowid_s2(emp_id,name) values (2,'empname2');
INSERT INTO testrowid_s2(emp_id,name) values (3,'empname3');
INSERT INTO testrowid_s2(emp_id,name) values (4,'empname4');
commit;

SELECT rowid,emp_id,name FROM testrowid_s2;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3rAAAAAAAMeAAA          1 empname1
AAAF3rAAAAAAAMeAAB          2 empname2
AAAF3rAAAAAAAMeAAC          3 empname3
AAAF3rAAAAAAAMeAAD          4 empname4

UPDATE testrowid_s2 SET name = 'Ramesh' WHERE rowid = 'AAAF3rAAAAAAAMeAAB' ;
commit;

SELECT rowid,emp_id,name FROM testrowid_s2;
ROWID                  EMP_ID NAME
------------------ ---------- ----------
AAAF3rAAAAAAAMeAAA          1 empname1
AAAF3rAAAAAAAMeAAB          2 Ramesh
AAAF3rAAAAAAAMeAAC          3 empname3
AAAF3rAAAAAAAMeAAD          4 empname4
```

PostgreSQL：

```
CREATE TABLE public.testrowid_s2
(
    emp_id integer,
    name character varying
);

insert into public.testrowid_s2 (emp_id,name) values
(1,'empname1'),(2,'empname2'),(3,'empname3'),(4,'empname4');

select emp_id,name from testrowid_s2;
 emp_id |   name
--------+----------
      1 | empname1
      2 | empname2
      3 | empname3
      4 | empname4

update testrowid_s2 set name = 'Ramesh' where emp_id = 2 ;

select emp_id,name from testrowid_s2;
 emp_id |   name
--------+----------
      1 | empname1
      3 | empname3
      4 | empname4
      2 | Ramesh
```

案例 3：使用身份識別屬性

在下列範例中，您可以使用識別屬性來建立沒有主索引鍵的資料表testrowid_s3。

甲骨文代碼：

```
create table testrowid_s3 (name varchar2(10));
INSERT INTO testrowid_s3(name) values ('empname1');
INSERT INTO testrowid_s3(name) values ('empname2');
INSERT INTO testrowid_s3(name) values ('empname3');
INSERT INTO testrowid_s3(name) values ('empname4');
commit;

SELECT rowid,name FROM testrowid_s3;
ROWID              NAME
------------------ ----------
```

```
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB empname2
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4

UPDATE testrowid_s3 SET name = 'Ramesh' WHERE rowid = 'AAAF3sAAAAAAAMmAAB' ;
commit;

SELECT rowid,name FROM testrowid_s3;
ROWID              NAME
------------------ ----------
AAAF3sAAAAAAAMmAAA empname1
AAAF3sAAAAAAAMmAAB Ramesh
AAAF3sAAAAAAAMmAAC empname3
AAAF3sAAAAAAAMmAAD empname4
```

PostgreSQL：

```
CREATE TABLE public.testrowid_s3
(
    rowid_seq bigint generated always as identity,
    name character varying
);

insert into public.testrowid_s3 (name) values
('empname1'),('empname2'),('empname3'),('empname4');

select rowid_seq,name from testrowid_s3;
 rowid_seq |   name
-----------+----------
         1 | empname1
         2 | empname2
         3 | empname3
         4 | empname4

update testrowid_s3 set name = 'Ramesh' where rowid_seq = 2 ;

select rowid_seq,name from testrowid_s3;
 rowid_seq |   name
-----------+----------
         1 | empname1
         3 | empname3
         4 | empname4
```

```
2 | Ramesh
```

將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫

創建者：西帕薩拉迪（AWS）和維拉賈納魯魯格蘭希（AWS）

環境：PoC 或試點	來源：甲骨文	目標：PostgreSQL
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon Aurora		

Summary

此模式顯示如何使用預先定義的中繼資料表，將 Oracle 資料庫錯誤碼移轉至 Amazon Aurora PostgreSQL 相容版本資料庫。

甲骨文數據庫錯誤代碼並不總是有相應的 PostgreSQL 錯誤代碼。這種錯誤代碼的差異可能會使得在目標 PostgreSQL 架構中配置程序或函數的處理邏輯變得困難。

您可以將對 PL/pgSQL 程式有意義的來源和目標資料庫錯誤碼儲存在中繼資料表中，以簡化程序。然後，設定表格以標記有效的 Oracle 資料庫錯誤代碼，並將它們對應至其 PostgreSQL 對等項目，然後再繼續其餘的處理序邏輯。如果 Oracle 資料庫錯誤碼不在中繼資料表中，處理序會以例外狀況結束。然後，您可以手動檢閱錯誤詳細資訊，並在程式需要時將新的錯誤代碼新增至表格。

透過使用此組態，您的 Amazon Aurora PostgreSQL 相容資料庫可以像來源 Oracle 資料庫一樣處理錯誤。

注意：設定 PostgreSQL 資料庫以正確處理 Oracle 資料庫錯誤碼，通常需要變更資料庫和應用程式程式碼。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 啟動並執行執行處理和監聽器服務的來源 Oracle 資料庫
- 已啟動並執行的 Amazon Aurora PostgreSQL 相容叢集
- 熟悉甲骨文數據庫

- 熟悉 PostgreSQL 庫

架構

下圖顯示 Amazon Aurora PostgreSQL 相容資料庫工作流程的範例，用於驗證和處理資料錯誤碼：

該圖顯示以下工作流程：

1. 一個表格包含 Oracle 數據庫錯誤代碼和分類及其等效的 PostgreSQL 錯誤代碼和分類。該表包括 valid_error 列，如果特定的，預定義的錯誤代碼是否有效，則對其進行分類。
2. 當 PL/PgSQL 函數（函數處理數據）拋出異常時，它會調用第二個 PL/PgSQL 函數（錯誤驗證）。
3. 錯誤驗證函數接受甲骨文數據庫錯誤代碼作為輸入參數。然後，函數會根據資料表檢查傳入的錯誤代碼，以查看錯誤是否包含在資料表中。
4. 如果甲骨文數據庫錯誤代碼包含在表中，則錯誤驗證函數返回一個 TRUE 值，並繼續進程邏輯。如果錯誤碼未包含在資料表中，則函數會傳回 FALSE 值，且處理序邏輯會以例外狀況結束。
5. 當函數傳回 FALSE 值時，應用程式的功能潛在客戶會手動檢閱錯誤詳細資訊，以判斷其有效性。
6. 然後，新的錯誤代碼是否手動添加到表中。如果錯誤代碼有效並添加到表中，則 error_validation 函數返回 TRUE 值下次發生異常時。如果錯誤碼無效，且處理序必須在例外狀況發生時失敗，則不會將錯誤碼新增至資料表。

技術堆疊

- Amazon Aurora PostgreSQL
- pgAdmin
- Oracle SQL Developer

工具

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- pgAdmin 是一個開放原始碼的管理和開發工 PostgreSQL。它提供了一個圖形界面，可簡化數據庫對象的創建，維護和使用。
- Oracle SQL 開發人員是一個免費的整合式開發環境，可簡化傳統與雲端部署中 Oracle 資料庫的開發與管理。

史诗

將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫

任務	描述	所需技能
在 Amazon Aurora PostgreSQL 相容資料庫中建立資料表。	執行下列 PostgreSQL 料表命令： `( source_error_code numeric NOT NULL, target_error_code character varying NOT NULL, valid_error character varying(1) NOT NULL );`	PostgreSQL 開發人員, 甲骨 PostgreSQL, 光源/Aurora
將 PostgreSQL 錯誤代碼及其對應的甲骨文數據庫錯誤代碼添加到表中。	執行 PostgreSQL 插入命令，將必要的錯誤碼值新增至錯誤碼資料表。 PostgreSQL 錯誤碼必須使用不同的字元資料類型 (SQL STATE 值)。甲骨文錯誤代碼必須使用數字數據類型（SQLCODE 值）。 示例插入語句： `insert into error_codes values (-1817,'2007','Y');`	PostgreSQL 開發人員, 甲骨 PostgreSQL, 光源/Aurora

任務	描述	所需技能
	`insert into error_codes values (-1816,'22007','Y');` `insert into error_codes values (-3114,'08006','N');` 注意：如果您要捕獲 Oracle 特定的 Java 數據庫連接（JDBC）異常，則必須將這些異常替換為通用跨數據庫異常或切換到 PostgreSQL 特定的異常。	
創建一個 PL/pgSQL 函數來驗證錯誤代碼。	通過運行創建函數命令創建一個 PL/PGSQL 函數。確保該函數執行以下操作： • 接受由程序拋出的 Oracle 錯誤代碼。 • 檢查錯誤代碼是否存在於錯誤代碼表。 • 返回 TRUE 或 FALSE 值，根據錯誤代碼是否存在於元數據表或不。	PostgreSQL 開發人員, 甲骨 PostgreSQL, 光源/Aurora

任務	描述	所需技能
手動檢閱 PL/pgSQL 函式所記錄的新錯誤碼。	手動檢閱新的錯誤碼。 如果新的錯誤碼對您的使用案例有效，請執行 PostgreSQL INSERT 命令，將其新增至錯誤碼資料表。 -或- 如果新的錯誤碼對您的使用案例無效，請勿將其新增至資料表。進程邏輯將繼續失敗，並在發生錯誤時以異常狀況退出。	PostgreSQL 開發人員, 甲骨 PostgreSQL, 光源/Aurora

相關資源

附錄 A 錯誤碼 (PostgreSQL 說明文件)

資料庫錯誤訊息 (Oracle 資料庫文件)

將 Redis 工作負載遷移到 AWS 上的 Redis 企業雲端

創建者：安東尼·普拉薩德·特瓦拉治（AWS）和斯里尼瓦斯彭迪亞拉（Redis）

環境：生產	來源：內部部署 (Redis 或其他) 資料庫	目標：AWS 上的企業雲端
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫
AWS 服務：AWS DMS；Amazon S3		

Summary

此模式討論將 Redis 工作負載遷移至 Amazon Web Services (AWS) 上 Redis 企業雲端的高階程序。其中說明移轉步驟、提供可用工具選取的相關資訊，以及討論使用每個工具的優點、缺點和步驟。或者，如果您在從 Redis 移轉工作負載時需要其他協助，您可以參與 Redis 專業服務。

如果您在內部部署執行 Redis OSS 或 Redis 企業軟體，您就熟悉在資料中心維護 Redis 資料庫所帶來的巨大管理開銷和作業複雜性。藉由將工作負載移轉至雲端，您可以大幅減輕這項操作負擔，並利用 Redis 企業雲端，這是 Redis 提供的完全託管資料庫即服務 (DBaaS) 產品。此遷移有助於提高業務靈活性、改善應用程式可靠性並降低整體成本，同時您可以存取最新的 Redis Enterprise Cloud on AWS 功能，例如 99.999% 的可用性、架構簡易性和擴展性。

Redis Enterprise Cloud 在金融服務、零售、醫療保健和遊戲領域，以及需要詐欺偵測、即時庫存、理賠處理和工作階段管理解決方案的使用案例中，都有潛在的應用程式。您可以使用 Redis 企業雲端連線到 AWS 資源，例如，連線到在 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上執行的應用程式伺服器，或連線到部署為 AWS Lambda 服務的微服務。

先決條件和限制

假設

- 您目前正在操作要移轉至雲端的內部部署資料庫系統。
- 您已確定工作負載的移轉需求，包括：
  - 資料一致性需求

  - 基礎架構和系統環境需求
  - 資料對應與轉換需求
  - 功能測試要求
  - 性能測試要求
  - 驗證要求
  - 定義切換策略
- 您已評估遷移所需的時間表和成本估算。
- 您的需求會將工作範圍以及您識別為移轉一部分的系統和資料庫納入考量。
- 您已經確定了利益相關者以及他們的角色和責任在一個負責任的，負責的，諮詢，知情的（RACI）矩陣。
- 您已收到所有利益相關者的必要協議和批准。

成本

根據現有來源資料庫的技術規格 (例如記憶體大小、輸送量和總資料大小)，Redis 解決方案架構設計人員可以在 Redis 企業雲上調整目標系統的大小。 如需一般定價資訊，請參閱 Redis 網站上的 Redis 定價。

人才與技能

移轉程序包含下列角色和職責。

Role	Description	所需技能
移轉解決方案架	具備定義、規劃和實作移轉策略專業知識的技術架構師	對來源和目標系統的技術和應用程式層級了解；將工作負載移轉至雲端的經驗
數據架構師	技術架構師，在定義、實作和提供各種資料庫的資料解決方案方面擁有豐富經驗	結構化和非結構化數據的數據建模，對企業實施數據庫有深刻的理解和經驗
Redis 解決方案架構師	技術架構師，可協助針對適當使用案例建構最佳大小的 Redis 叢集	針對各種使用案例建構和部署 Redis 解決方案的專業知識

雲端方案架構師	對雲端解決方案有深入瞭解的技術架構師，尤其是在 AWS 上	具備雲端架構解決方案的專業知識；工作負載移轉與應用程式現代化經驗
企業建築師	一位技術架構師，對貴組織的技術格局有全面了解，他對 future 藍圖有共同的願景，以及誰在組織中的所有團隊中實踐和建立標準化的架構最佳實踐	軟體架構認證，例如 TOGAF、基礎軟體工程技能、解決方案架構和企業架構專業知識
IT 或 DevOps 工程師	負責建立和維護基礎結構的工程師，包括監視基礎結構的問題、執行維護工作，以及視需要進行更新。	深入了解各種技術，包括操作系統，網絡和雲計算; 熟悉 Python，Bash 和 Ruby 等編程語言以及碼頭，Kubernetes 和 Ansible 等工具

架構

移轉選項

下圖顯示將現場部署 (以 Redis 為基礎或其他) 資料來源遷移到 AWS 的選項。它顯示了數種您可以選擇的遷移工具，例如將 Redis 資料庫 (RDB) 檔案匯出到亞馬遜簡單儲存服務 (Amazon S3)、使用 Redis 複寫功能或使用 AWS DMS。

1. 內部部署資料來源：不是以 Redis 為基礎的資料庫，例如 MySQL、PostgreSQL、甲骨文、SQL 伺服器或 MariaDB。
2. 內部部署資料來源：以 Redis 通訊協定為基礎的資料庫，例如 Redis OSS 和 Redis 企業軟體。
3. 從基於 Redis 的數據庫遷移數據最簡單的方法是導出 RDB 文件並將其導入到 AWS 上的目標 Redis 企業雲中。
4. 或者，您可以使用 Redis 中的複寫功能 (`Replica0f`) 將資料從來源移轉到目標。
5. 如果您的數據遷移要求包括數據轉換，則可以使用 Redis 輸入/輸出工具（RIOT）遷移數據。
6. 或者，您也可以使用 AWS 資料遷移服務 (AWS DMS) 從 SQL 資料庫遷移資料。

7. 您必須為 AWS DMS 使用虛擬私有雲端 (VPC) 對等互連，才能將資料成功遷移到 AWS 上的目標 Redis 企業雲端。

目標架構

下圖顯示 AWS 上 Redis 企業雲端的典型部署架構，並說明如何將其與重要 AWS 服務搭配使用。

1. 您可以連線到 AWS 上由 Redis 企業雲端支援的商業應用程式。
2. 您可以在自己的 AWS 帳戶中，在該帳戶的 VPC 中執行商業應用程式。
3. 您可以使用 Redis 企業雲端資料庫端點連線到應用程式。範例包括在 EC2 執行個體上執行的應用程式伺服器、部署為 AWS Lambda 服務的微服務、亞馬遜彈性容器服務 (Amazon ECS) 應用程式或 Amazon Elastic Kubernetes Service (Amazon EKS) 應用程式。
4. 在 VPC 中執行的商務應用程式需要與 Redis 企業雲端 VPC 擬私人雲端 VPC 人雲端的對等連線。這使業務應用程序可以通過私有端點安全地連接。
5. AWS 上的 Redis 企業雲端是在 AWS 上部署為 DBaaS 的記憶體內部 NoSQL 資料庫平台，並由 Redis 完全管理。
6. Redis 企業雲端部署在由 Redis 建立的標準 AWS 帳戶中的 VPC 內。
7. 基於安全理由，Redis 企業雲部署在私有子網路中，可在私有端點和公用端點存取。建議您將用戶端應用程式連線到私有端點上的 Redis。如果您打算使用公用端點，強烈建議您啟用 TLS 來加密用戶端應用程式和 Redis 企業雲之間的資料。

Redis 遷移方法與 AWS 遷移方法一致，該方法在 AWS Prescriptive Guidance 網站上動員您的組織以加速大規模遷移。

自動化和規模

移轉的環境設定任務可透過 AWS 登陸區域和基礎設施即程式碼 (IAC) 範本自動化，以進行自動化和擴展。這些都在這種模式的史詩部分討論。

工具

根據您的資料遷移需求，您可以從一系列技術選項中選擇將資料遷移到 AWS 上的 Redis 企業雲端。下表說明並比較這些工具。

工具	Description	優點	缺點

RDB 導出和導入	您可以從來源匯出資料 (例如 Redis 的 OSS 或 Redis 企業軟體) 資料庫中的 RDB 檔案的形式。如果您的資料庫是透過 Redis OSS 叢集提供，則會將每個主要碎片匯出至 RDB。 然後，您可以在一個步驟中匯入所有 RDB 檔案。如果您的來源資料庫是以 OSS 叢集為基礎，但目標資料庫並未使用 OSS 叢集 API，您必須變更應用程式原始程式碼以使用標準 Redis 用戶端程式庫。 資料轉換需求或邏輯資料庫合併需要較複雜的程序，此程序會在此表格稍後的「邏輯資料庫合併」中說明。	• 簡單。 • 適用於任何基於 Redis 的解決方案，該解決方案可以將 RDB 格式的數據導出為源（包括 Redis OSS 和 Redis 企業軟件）。 • 通過簡單的過程實現數據一致性。	• 不滿足資料轉換需求或支援邏輯資料庫合併。 • 耗時較大的資料集。 • 沒有增量遷移支持可以導致更長的停機時間。

Redis 複寫功能 (主動-被動)	您可以持續將資料從 Redis OSS、企業軟體或企業雲端資料庫複寫到 Redis 企業雲端資料庫。初始同步處理之後，Redis 複寫功能 (`ReplicaOf `) 會執行增量移轉，這表示幾乎沒有觀察到的應用程式停機時間。 Redis 複寫功能旨在以主動-被動方式使用。假設目標是被動的，並取得完全重新同步處理 (清除並從來源資料庫同步處理)。因此，在來源和目標之間切換稍微複雜一些。 您可以將 OSS 叢集的所有主要碎片指定為來源，從 Redis OSS 叢集複寫到標準叢集 Redis 企業雲端資料庫。不過，Redis 複寫功能最多允許 32 個來源資料庫。	• 支援連續複寫 (初始資料載入後跟差異)。 • 幾乎沒有停機時間 (取決於複寫延遲)。 • 實現資料一致性。	• 只有一個站點打算處於活動狀態，因此在站點之間切換更加複雜。 • 從 OSS 叢集移轉時，最多支援 32 個主要碎片。

AWS DMS	您可以使用 AWS DMS 將資料從任何支援的來源資料庫遷移到目標 Redis 資料存放區，而且停機時間最短。如需詳細資訊，請參閱 AWS DMS 文件中的使用 Redis 做為 AWS DMS 的目標。	• 支援 NoSQL 和 SQL 資料來源的移轉。 • 與其他 AWS 服務搭配良好運作。 • 支援即時移轉和變更資料擷取 (CDC) 使用案例。	• Redis 的鍵值不能包含特殊字符，如 %。 • 不支援移轉列或欄位名稱中包含特殊字元的資料。 • 不支援完整的大型二進位物件 (LOB) 模式。
邏輯資料庫合併	特殊的資料庫合併需求可能需要自訂資料移轉解決方案。例如，Redis OSS 中可能有四個邏輯資料庫 (`SELECT 0..3`)，但您可能想要使用單一資料庫端點，而不是將資料移至多個 Redis 企業雲端資料庫。Redis Enterprise 不支援可選取的邏輯資料庫，因此您必須轉換來源資料庫的實體資料模型。例如，您可以將每個資料庫索引對應`0`至前置詞 (1到`usrcmp`、目標等)，然後使用移轉指令碼或擷取、轉換和載入 (ETL) 工具輸出 RDB 檔案，然後您可以將其匯入目標資料庫。	• 使用自訂指令碼，在移轉至目標系統期間對整形資料進行精細控制。	• 如果您決定不完成移轉，復原可能會非常具有挑戰性，尤其是在必須將較新的資料復原至來源系統時。 • 如果目標是為一次性移轉建置一次性解決方案，則建置成本可能會很高。 • 如果移轉需求頻繁變更，程式碼、基礎結構、開發時間和其他區域的維護成本可能會很高。

此外，您還可以使用 AWS 提供的下列工具和服務。

評估和探索工具：

- AWS Application Discovery Service
- 移轉評估員

應用程式和伺服器移轉工具：

- AWS Application Migration Service

資料庫遷移工具：

- AWS Schema Conversion Tool
- AWS Database Migration Service

資料移轉工具：

- AWS Storage Gateway
- AWS DataSync
- AWS Direct Connect
- AWS Snowball
- Amazon 數據 Firehose

移轉管理：

- AWS Migration Hub

AWS 合作夥伴解決方案

- AWS 遷移能力合作夥伴

史诗

完成探索和評估工作

任務	描述	所需技能
識別工作負載。	識別您要移轉的適當候選工作負載。選擇要移轉的工作負載之前，請考慮下列事項： • 移轉或不移轉此工作負載的商業價值為何？ • 如果此工作負載未成功移轉至目標系統，是否有應變計劃？ 理想情況下，選擇具有最大業務影響且風險最小的工作負載。保持整個過程迭代和以小的增量遷移。	資料架構師、商業領袖、移轉專案贊助商
識別資料來源和需求；設計資料模型。	Redis 舉辦研討會，以加速探索並定義專案的移轉規劃。作為本研討會的一部分，Redis 團隊會識別資料來源和來源資料模型需求，並分析如何在 Redis 企業雲中重新整理這些需求。 Redis 移轉小組 (專業服務) 會與您的組織執行詳細的資料模型設計練習。作為本練習的一部分，Redis 團隊： • 標識目標 Redis 的數據結構。 • 定義資料對映策略。	Redis 解決方案架構師

任務	描述	所需技能
	• 記錄遷移方法和建議。 • 與利益相關者審查並完成數據模型。	
識別來源資料庫的特性。	識別在來源和目標環境中使用的 Redis 產品。例如： • 來源資料庫是 OSS 叢集資料庫、獨立 Redis 資料庫還是 Redis 企業資料庫？ • 目標資料庫是 Redis 企業標準資料庫還是 OSS 叢集相容資料庫？ • 應用程序源代碼有什麼影響？	資料架構師
收集目前的系統 SLA 和其他大小指標。	決定以輸送量 (每秒作業數)、延遲、每個資料庫的整體記憶體大小和高可用性 (HA) 需求來表示的目前服務等級協定 (SLA)。	資料架構師

任務	描述	所需技能
識別目標系統的特性。	確定這些問題的答案： • 需要遷移多少資料？ • 遷移給定數量的數據需要多長時間？ • 遷移的停機時間需求為何？您的服務或應用程序在特定時間內不可用是否可以接受？ 如果是這樣，多長時間？ • 移轉的資料應該如何一致？目標數據庫是否可以處於稍微不一致（過時）的狀態？ • 數據是否必須在加載到目標數據庫之前進行轉換？（例如，您可能希望在遷移之前將可選擇的數據庫索引轉換為前綴。） • 來源資料庫是否可從目標資料庫的主機存取 (例如，從對等 VPC 或使用加密的公用端點)？ • 使用 Redis 技術架構師完成資料大小調整和 Redis 叢集大小調整練習。 • 識別網路需求、基礎架構需求、軟體版本和軟體授權，並在移轉前採購任何元件。 • 這些數據的傳輸是否存在任何安全問題？	資料架構師、Redis 解決方案架構師 (選用)

任務	描述	所需技能
識別依賴關係。	識別要移轉之目前系統的上游與下游相依性。確定移轉工作與其他相依系統移轉保持一致。例如，如果您打算將其他商業應用程式從現場部署遷移到 AWS 雲端，請識別這些應用程式，並根據專案目標、時間表和利益相關者進行調整。	資料架構師、企業架構師
識別移轉工具。	根據您的資料移轉需求 (例如來源資料或停機時間需求)，您可以使用上述「工具」一節中所述的任何工具。此外，您還可以使用： • 使用 CRDB 部署進行雙向 (主動-主動) 複寫。 • 自定義導出/導入腳本（例如，通過使用DUMP/RESTORE 命令）。 • 其他導出/導入工具和輔助工具，例如 RIOT，ECSTATS 2 或 ETL 工具。 • IaC 工具，例如地形或 AWS 模板。 CloudFormation	移轉解決方案架構師、Redis 解決方案架
建立應變計劃。	建立應變計劃，以便在移轉期間遇到問題時復原。	項目管理，技術團隊，包括建築師

完成安全性與合規性工作

任務	描述	所需技能
保護 Redis 管理主控台的安全。	若要保護管理主控台，請遵循 Redis 文件中的指示。	IT 基礎架構管理
保護 Redis 的資料庫。	請參閱 Redis 文檔中的以下頁面： • 定義角色型存取控制。 • 定義網路安全性。 • 啟用 TLS。	
安全的雲端 API。	當您啟用 API 時，您可以為 Redis 雲端帳戶的所有擁有者管理 API 金鑰。如需 API 安全性功能的概觀，請參閱 Redis 網站上的 API 驗證文件。	IT 基礎架構管理

設定新環境

任務	描述	所需技能
在 AWS 上設定新環境。	此工作包括： • AWS 登陸區域設定活動。landing zone 支持： • 多帳戶部署 • 最低安全性基準 • 以安全性基準和 ISV 先決條件 (網路、安全性組態等) 佈建新帳戶的自動化方式 • 通知、集中記錄和監控	IT 或 DevOps 工程師

任務	描述	所需技能
	• ISV 軟體組態活動。這包括需要包含在移轉中的組態，例如產品和工作負載設定以及變更。 • IaC 活動，例如設定或自訂 AWS CloudFormation 或地形範本。	
部署移轉架構。	1. 在 AWS 上設定 Redis 企業雲端。 2. 安裝 RIOT 或 AWS DMS 等遷移工具。如需可用工具的清單，請參閱「工具」一節。 3. 在應用程式、移轉和資料庫層之間建立連線。 4. 建立一個範例工作負載，該工作負載可以流經每個層並移轉一小組範例資料。 您現在已準備好執行實際的資料遷移管道並對其進行測試。	IT 或 DevOps 工程師

設定網路

任務	描述	所需技能
建立連線能力。	在現場部署基礎設施和 AWS 雲端資源之間建立連線。使用安全群組、AWS Direct Connect 和其他資源來實現此功能。如需詳細資訊，請參	IT 或 DevOps 工程師

任務	描述	所需技能
	閱 AWS 網站上的將資料中心 Connect 到 AWS。	
設定 VPC 對等互連。	在執行商業應用程式的 VPC (或執行移轉工具的 EC2 執行個體或 AWS DMS 複寫伺服器) 與執行 Redis 企業雲端的 VPC 之間建立 VPC 對等。如需指示，請參閱 Amazon VPC 文件中的開始使用 Amazon VPC，以及 Redis 文件中的啟用 VPC 對等。	IT 或 DevOps 工程師

移轉資料

任務	描述	所需技能
選擇資料移轉工具。	檢閱「工具」區段中的表格，以查看這些工具的說明、優點和缺點： • RDS 匯出和匯入 • Redis 的複製功能 ( ) `ReplicaOf` • AWS DMS • 邏輯資料庫合併 下列列說明與每個工具相關聯的資料移轉工作。	移轉解決方案架
選項 1：使用 RDB 匯出和匯入。	1. 中斷來源連線：停止來源資料庫上的流量 (例如，中斷商務應用程式的連線)。	移轉解決方案架構師、Redis 解決方案架

任務	描述	所需技能
	2. 匯出：將來源資料庫的資料匯出為 RDB 檔案。 3. 階段：將資料上傳到 AWS 上 Redis 企業雲端執行個體可存取的位置 (例如，您可以將資料上傳到 S3 儲存貯體或 FTP 伺服器)。 4. 導入：將 RDB 文件（通過在一個導入步驟中列出所有文件）導入到 Redis 企業雲目標數據庫。 5. 切除：移至目標資料庫 (例如，透過將應用程式連線到該資料庫)。 如需詳細資訊，請參閱 Redis 文件。	

任務	描述	所需技能
選項 2：使用 Redis 複寫功能 (主動-被動)。	1. Connect 資料庫：在來源資料庫和目標資料庫之間建立連`ReplicaOf` 結。 2. 執行初始同步：等到來源資料庫和目標資料庫之間的初始同步處理完成。 3. 中斷來源連線：停止來源資料庫上的流量 (例如，中斷應用程式的連線)。 4. 執行增量複寫：等到目標資料庫上複寫增量。 5. 切斷：移至目標資料庫 (例如，透過將應用程式連接至該資料庫)。 6. 刪除：移除來源資料庫和目標資料庫之間的`ReplicaOf` 連結。 如需詳細資訊，請參閱 Redis 文件。	移轉解決方案架構師、Redis 解決方案架

任務	描述	所需技能
選項 3：使用 AWS DMS。	1. 設定 AWS DMS 複寫執行個體：此執行個體會執行所有遷移程序。如需指示：在 AWS DMS 文件中使用 AWS DMS 複寫執行個體。 2. 定義來源資料庫：定義來源端點。測試來源端點與 AWS DMS 複寫伺服器之間的連線。如需指示：在 AWS DMS 文件中建立來源端點和目標端點。 3. 設定目標資料庫：在 AWS 上設定 Redis 企業雲端，並設定要遷移到的資料庫。 4. 定義目標資料庫：定義目標端點。請確定已在執行 AWS DMS 的 VPC 人雲端與在 AWS 上託管 Redis 企業雲端的虛擬私人雲端之間建立 VPC 對等。測試 AWS DMS 複寫伺服器與目標資料庫之間的連線。 5. 建立 AWS DMS 任務：建立一個或一組任務，以定義要用來遷移資料的表格和複寫程序。如需指示：在 AWS DMS 文件中使用 AWS DMS 任務。 6. 遷移：透過執行 AWS DMS 任務來遷移資料。 7. 切斷：移至目標資料庫 (例如，透過將應用程式連接至該資料庫)。	移轉解決方案架構師、Redis 解決方案架

任務	描述	所需技能
選項 4：使用邏輯資料庫合併。	此選項涉及使用遷移腳本或 ETL 工具，該工具可以轉換源數據庫的物理數據模型並生成 RDB 文件。如有需要，Redis 專業服務可協助完成此步驟。	移轉解決方案架構師、Redis 解決方案架

遷移您的應用

任務	描述	所需技能
調整專案管理時間表和目標。	將應用程式層的遷移專案目標、里程碑和時間表與 Redis 資料遷移專案的目標、里程碑和時間表一致。	專案管理
調整測試活動。	在 AWS 雲端中遷移和現代化應用程式層之後，請將應用程式層指向 AWS 上新遷移的 Redis 企業雲端進行測試。	測試

測試

任務	描述	所需技能
實施測試計劃。	在您的網站的測試環境中，根據測試需求，執行實作階段期間開發的資料移轉常式和指令碼。	測試
測試資料品質。	在移轉資料後測試資料品質。	測試
測試功能。	測試資料查詢和應用程式層，以確保應用程式的執行層級與來源系統相同。	測試

切過

任務	描述	所需技能
做出切換決定。	完成所有應用程式層級和資料庫層級測試之後，執行領導團隊和利益相關者就是否根據測試團隊確認的最終結果切換到 AWS 上的新環境做出最終決定。	項目管理，商業冠軍
切換到 AWS 雲端。	確認所有項目都已就緒後，請將應用程式層指向新移轉的資料，並將用戶端指向以 AWS 上新 Redis Enterprise Cloud 系統為基礎執行的新應用程式層。	IT 或 DevOps 工程師、資料架構師、移轉解決方案架構師、Redis 解決方案架構師

相關資源

雷迪斯資源

- Redis 企業雲端文件
- 防暴工具（GitHub 存儲庫）
- 地形表單提供者 (下載)

AWS 資源

- 示範移轉
- AWS 夥伴解決方案
- 文件
- 部落格文章
- 白皮书
- 教學課程和影片
- AWS 雲端移轉

- AWS 方案指引

其他資訊

如需將 Redis 工作負載遷移到 AWS 雲端的標準安全要求，請參閱 AWS 網站上的安全、身分和合規的最佳實務，以及 Redis 網站上的 Redis 信任中心。

使用 AWS SCT 和 AWS DMS 將亞馬 Amazon EC2 上的 SAP ASE 遷移到與 Amazon Aurora PostgreSQL 相容

由阿米特·庫馬爾（AWS）和安吉·古普塔創建

環境：PoC 或試點	資料來源：SAP 日月光	目標：Aurora 郵政兼容
R 類型：重新平台	工作負載：SAP	技術：移轉；資料庫
AWS 服務：AWS DMS；AWS SCT		

Summary

此模式說明如何使用 AWS Schema Conversion Tool (AWS SCT) 和 AWS Database Migration Service (AWS DMS)，將託管在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上的 SAP 自適應伺服器企業 (SAP ASE) 資料庫遷移到亞馬遜 Aurora PostgreSQL 相容版本。該模式著重於存儲對象的數據定義語言（DDL）轉換和數據遷移。

Aurora PostgreSQL 相容支援線上交易處理 (OLTP) 工作負載。此受管理服務提供可依需求自動調整規模的組態。它可以根據應用程序的需求自動啟動，關閉，擴展或縮小數據庫。您可以在雲端執行資料庫，而無需管理任何資料庫執行個體。Aurora PostgreSQL 相容為罕見、間歇性或無法預測的工作負載提供了符合成本效益的選項。

遷移過程包括兩個主要階段：

- 使用 AWS SCT 轉換資料庫結構描述
- 使用 AWS DMS 遷移資料

有關這兩個階段的詳細說明，請參閱《史詩》一節。如需將 AWS DMS 與 SAP ASE 資料庫搭配使用特定問題的疑難排解相關資訊，請參閱 AWS DMS 文件中的 SAP ASE 疑難排解問題。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- EC2 執行個體上的來源 SAP ASE 資料庫，伺服器、資料庫和接聽程式服務已啟動並執行
- 目標 Aurora 與 PostgreSQL 相容的資料庫

限制

- 連線的連接埠號碼必須是 5432。
- 預設情況下，huge_pages 功能處於開啟狀態，但可以修改。
- P oint-in-time 復原 (PITR) 粒度為 5 分鐘。
- 目前無法使用跨區域複寫。
- Aurora 資料庫的最大儲存體大小為 128 TiB。
- 您最多可以建立 15 個僅供讀取複本。
- 資料表大小限制僅受限於 Aurora 叢集磁碟區的大小，因此 Aurora PostgreSQL 相容資料庫叢集的資料表大小上限為 32 TiB。我們建議您遵循資料表設計的最佳作法，例如對大型資料表進行分割。

產品版本

- 來源資料庫：AWS DMS 目前支援日月光 15、15.5、15.7 和 16.x。如需有關 SAP ASE 版本支援的最新資訊，請參閱 AWS DMS 使用者指南。
- 目標資料庫：PostgreSQL 9.4 及更新版本 (適用於版本 9.x)、10.x、11.x、12 倍、13.x 和 14 倍。如需最新支援的 PostgreSQL 版本，請參閱 AWS DMS 使用者指南。
- Amazon Aurora 1.x 或更高版本。如需最新資訊，請參閱 Aurora 文件中與 Aurora PostgreSQL 相容的發行版本和引擎版本。

架構

源, 技術, 堆棧

- 在 Amazon EC2 上執行的 SAP ASE 資料庫

目標技術堆疊

- Aurora 兼容數據庫

移轉架構

工具

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- AWS Schema Conversion Tool (AWS SCT) 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，藉此支援異質資料庫遷移。
- AWS DMS 支援數個不同的來源和目標資料庫。如需詳細資訊，請參閱 AWS DMS 文件中的資料遷移來源和資料遷移的目標。如需最全面的版本和功能支援，我們建議您使用最新版本的 AWS DMS。

史诗

設定環境

任務	描述	所需技能
在來源 EC2 執行個體中設定網路存取。	在託管來源 SAP ASE 資料庫的 EC2 執行個體中設定安全群組。 如需指示，請參閱 Amazon EC2 文件中適用於 Linux 執行個體的 Amazon EC2 安全群組。	系統管理員
建立您的目標 Aurora 與 PostgreSQL 相容的資料庫叢集。	為目標資料庫安裝、設定和啟動與 Aurora PostgreSQL 相容的叢集。 如需詳細資訊，請參閱 Aurora 文件中的建立 Amazon Aurora 資料庫叢集。	DBA
設定目標資料庫叢集的授權。	設定目標資料庫的安全群組和防火牆。	DBA, 系統管理員

任務	描述	所需技能
	如需指示，請參閱 Aurora 文件中的建立 Amazon Aurora 資料庫叢集。	

使用 AWS SCT 轉換您的資料庫結構描述

任務	描述	所需技能
啟動 AWS SCT。	按照 AWS SCT 文件中的指示啟動 AWS SCT。 AWS SCT 提供以專案為基礎的使用者界面，可自動將 SAP ASE 來源資料庫的資料庫結構描述轉換為與目標 Aurora PostgreSQL 相容資料庫執行個體的格式。	DBA
建立 AWS SCT 端點。	為來源 SAP 日月光和目標 PostgreSQL 資料庫建立端點。 如需相關指示，請參閱 AWS SCT 文件。	DBA
建立評估報告。	建立資料庫移轉評估報告以評估移轉並偵測任何不相容的物件和功能。 如需相關指示，請參閱 AWS SCT 文件。	DBA
轉換結構描述。	依照 AWS SCT 文件中的指示轉換資料庫結構描述。	DBA

任務	描述	所需技能
驗證數據庫對象。	如果 AWS SCT 無法轉換資料庫物件，則會識別其名稱和其他詳細資訊。您必須手動轉換這些物件。 若要識別這些不相符項目，請遵循 AWS 部落格文章中的指示，在從 SAP ASE 移轉至亞馬遜 RDS 版 PostgreSQL 或 Amazon Aurora PostgreSQL 後驗證資料庫物件。	DBA

分析 AWS DMS 遷移

任務	描述	所需技能
驗證來源和目標資料庫版本。	檢查 SAP ASE 資料庫版本是否與 AWS DMS 的相容性。 如需詳細資訊，請參閱 AWS DMS 文件中的 AWS DMS 來源和 AWS DMS 的目標。	DBA
識別儲存類型和容量的需求。	根據來源資料庫的大小，為目標資料庫選擇適當的儲存容量。	DBA, 系統管理員
選擇複製執行個體的執行個體類型、容量和其他功能。	選擇符合您需求的執行個體類型、容量、儲存功能和網路功能。 如需指引，請參閱 AWS DMS 文件中為您的遷移選擇正確的 AWS DMS 複寫執行個體。	DBA, 系統管理員

任務	描述	所需技能
識別網路存取安全性需求。	識別來源和目標資料庫的網路存取安全性需求。 請遵循 AWS DMS 文件中的為複寫執行個體設定網路中的指導。	DBA, 系統管理員

遷移數據

任務	描述	所需技能
透過在 AWS DMS 中建立遷移任務來遷移資料。	若要遷移資料，請建立任務並遵循 AWS DMS 文件中的指示進行操作。 我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。	DBA
驗證資料。	若要驗證您的資料是否已準確地從來源資料庫遷移到目標資料庫，請遵循 AWS DMS 文件中提供的資料驗證準則。	DBA

移轉應用程式

任務	描述	所需技能
識別應用程式移轉策略。	選擇七種策略之一 (7Rs)，將應用程式移轉至雲端。	DBA、應用程式擁有者、系統管理員
遵循應用程式遷移策略。	完成應用程式小組所識別的資料庫工作，包括更新目標資料	DBA、應用程式擁有者、系統管理員

任務	描述	所需技能
	庫的 DNS 連線詳細資料，以及更新動態查詢。	

切換到目標數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。	將來源資料庫的連線切換至目標資料庫。 如需詳細資訊，請參閱關聯式資料庫移轉策略的「切換」一節。	DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。	終止所有遷移任務、複寫執行個體、端點以及其他 AWS SCT 和 AWS DMS 資源。 如需詳細資訊，請參閱 AWS DMS 文件。	DBA, 系統管理員
審核並驗證專案文件。	驗證專案文件中的所有步驟，以確保所有工作都已順利完成。	DBA、應用程式擁有者、系統管理員
關閉專案。	關閉移轉專案並提供任何意見反應。	DBA、應用程式擁有者、系統管理員

相關資源

參考

- 在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線 (AWS Prescriptive Guidance)
- 使用 pg_port 在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫 (AWS Prescriptive Guidance)
- Amazon Aurora 定價
- 與 Amazon Aurora PostgreSQL 相容版本的最佳實務 (Amazon Aurora 文件)
- AWS SCT 文件
- AWS DMS 說明文件
- 使用 SAP ASE 資料庫做為 AWS DMS 的來源

教學課程和影片

- 開始使用 AWS Database Migration Service
- AWS Database Migration Service (影片)

使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器

創建者：茜卓·謝哈爾雅拉塔（AWS）和伊戈爾·科瓦爾丘克（AWS）

環境：生產	來源：視窗網絡應用程序	目標：AWS 上的 Application Load Balancer
R 類型：重新平台	工作量：Microsoft	技術：移轉、管理與治理、Web 和行動應用程式
AWS 服務：Elastic Load Balancing (ELB)；AWS Certificate Manager (ACM)		

Summary

該模式提供了使用 AWS Certificate Manager (ACM) 從現場部署伺服器上託管的網站或 Microsoft 網際網路資訊服務 (IIS) 上的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體遷移現有的安全通訊端層 (SSL) 憑證的指導。然後，SSL 憑證便可與 AWS 上的 Elastic Load Balancing 搭配使用。

SSL 保護您的數據，確認您的身份，提供更好的搜索引擎排名，有助於滿足支付卡行業數據安全標準（PCI DSS）要求，並提高客戶信任度。管理這些工作負載的開發人員和 IT 團隊希望他們的 Web 應用程式和基礎結構 (包括 IIS 伺服器和 Windows Server) 保持符合其基準原則。

此模式涵蓋從 Microsoft IIS 手動匯出現有的 SSL 憑證，將它們從個人資訊交換 (PFX) 格式轉換為 ACM 支援的私人增強型郵件 (PEM) 格式，然後將它們匯入 AWS 帳戶中的 ACM。它也說明如何為您的應用程式建立 Application Load Balancer，並將應用 Application Load Balancer 設定為使用匯入的憑證。然後在 Application Load Balancer 器上終止 HTTPS 連線，您不需要 Web 伺服器上的進一步設定額外負荷。如需詳細資訊，請參閱為應用程式負載平衡器建立 HTTPS 接聽程式。

Windows 伺服器會使用 .pfx 或 .p12 檔案來包含公開金鑰檔案 (SSL 憑證) 及其唯一的私密金鑰檔案。憑證授權單位 (CA) 會提供您的公開金鑰檔案。您可以使用伺服器來產生建立憑證簽署要求 (CSR) 的相關私密金鑰檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS 上的虛擬私有雲端 (VPC)，在目標使用的每個可用區域中至少有一個私有子網路和一個公有子網路
- 在視窗伺服器 2012 或更新版本上執行的 IIS 8.0 或更新版本
- 在 IIS 上執行的網路應用程式
- IIS 伺服器的管理員存取權

架構

源, 技術, 堆棧

- 使用 SSL 的 IIS 網頁伺服器實作，以確保資料在加密連線中安全傳輸 (HTTPS)

來源架構

目標技術堆疊

- 您的 AWS 帳戶中的 ACM 憑證
- 設定為使用匯入憑證的應用 Application Load Balancer
- 私有子網路中的 Windows 伺服器執行個體

目標架構

工具

- AWS Certificate Manager (ACM) 可協助您建立、存放和更新公有和私有 SSL/TLS X.509 憑證和金鑰，以保護您的 AWS 網站和應用程式。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 EC2 執行個體、容器和 IP 地址。

最佳實務

- 強制執行從 HTTP 到 HTTPS 的流量重新導向。

- 正確設定 Application Load Balancer 的安全群組，以僅允許傳入流量傳輸至特定連接埠。
- 在不同的可用區域啟動 EC2 執行個體，以確保高可用性。
- 將應用程式的網域設定為指向應用程式負載平衡器的 DNS 名稱，而非其 IP 位址。
- 確定應用 Application Load Balancer 已設定應用程式層健全狀況檢查。
- 設定健全狀況檢查的臨界值。
- 使用 Amazon CloudWatch 監控應用 Application Load Balancer。

史诗

匯出 .pfx 檔案

任務	描述	所需技能
從視窗伺服器匯出 .pfx 檔案。	若要從 Windows 伺服器中的內部部署 IIS 管理員將 SSL 憑證匯出為 .pfx 檔案，請執行下列動作： 1. 選擇開始、系統管理、網際網路資訊服務 (IIS) 管理員。 2. 選取伺服器名稱，然後在「安全性」下按兩下「伺服器憑證」 3. 選擇您要匯出的憑證，然後選擇 [匯出]。 4. 在「匯出憑證」方塊中，選擇 .pfx 檔案的位置、路徑和名稱。 5. 指定並確認 .pfx 檔案的密碼。 注意：安裝 .pfx 檔案時需要此密碼。 6. 選擇確定。	系統管理員

任務	描述	所需技能
	您的 .pfx 檔案現在應該儲存到您指定的位置和路徑。	

將 PFX 編碼的憑證轉換為 PEM 格式

任務	描述	所需技能
下載並安裝 OpenSSL 工具包。	1. 下載並安裝 Win32/64 從閃亮的光線製作網站。 2. 將 OpenSSL 二進位檔案的位置新增至系統PATH變數，以便可以使用二進位檔案供命令列使用。	系統管理員
將 PFX 編碼的憑證轉換為 PEM 格式。	下列步驟會將 PFX 編碼的已簽署憑證檔案轉換為三個 PEM 格式的檔案： • `cert-file.pem` 包含資源的 SSL/TLS 憑證。 • `privatekey.pem` 包含沒有密碼保護的憑證私密金鑰。 • `ca-chain.pem` 包含 CA 的根憑證。 若要轉換 PFX 編碼憑證： 1. 執行視窗 PowerShell。 2. 使用下列命令從 PFX 檔案擷取憑證的私密金鑰。出現提示時輸入憑證密碼。	系統管理員

任務	描述	所需技能
	`openssl pkcs12 -in <filename>.pfx -nocerts -out withpw-privatekey.pem` 此命令會產生名為的 PEM 編碼私密金鑰檔案。`privatekey.pem` 出現提示時，請輸入密碼片語以保護私密金鑰檔案。 3. 執行下列命令以移除密碼。出現提示時，請提供您在步驟 2 中建立的複雜密碼。 `openssl rsa -in withpw-privatekey.pem -out privatekey.pem` 如果命令成功，它會顯示訊息「寫入 RSA 金鑰」。 4. 使用下列命令將憑證從 PFX 檔案傳輸至 PEM 檔案。 `openssl pkcs12 -in <file_name>.pfx -clcerts -nokeys -out cert-file.pem` 這會建立名為的 PEM 編碼憑證檔案。`cert-file.pem` 如果命令成功，它會顯示消息「MAC 驗證確定。」	

任務	描述	所需技能
	5. 從 PFX 檔案建立 CA 鏈結檔案。下列命令會建立名為的 CA 鏈結檔案`ca-chain.pem` 。 `openssl pkcs12 -in <file_name>.pfx -cacerts -nokeys -chain -out ca-chain.pem` 如果命令成功，它會顯示消息「MAC 驗證確定。」	

將憑證匯入 ACM

任務	描述	所需技能
準備匯入憑證。	在 ACM 主控台上，選擇 [匯入憑證]。	雲端管理員
提供憑證主體。	針對憑證主體，貼上您要匯入的 PEM 編碼憑證。 如需此史詩中所描述之命令和步驟的詳細資訊，請參閱 ACM 說明文件中的匯入憑證。	雲端管理員
提供憑證私密金鑰。	對於 Certificate private key (憑證私有金鑰)，貼上與憑證公有金鑰相符的 PEM 編碼、未加密私有金鑰。	雲端管理員
提供憑證鏈結。	對於憑證鏈結，貼上儲存在檔案中的 PEM 編碼憑證鏈結。`CertificateChain.pem`	雲端管理員

任務	描述	所需技能
匯入憑證。	選擇 Review and import (檢閱和匯入)。確認憑證的相關資訊正確無誤，然後選擇 [匯入]。	雲端管理員

建立 Application Load Balancer

任務	描述	所需技能
建立和設定負載平衡器和監聽器。	請遵循 E lastic Load Balancing 說明文件中的指示來設定目標群組、註冊目標，以及建立「Application Load Balancer」和監聽器。為連接埠 443 新增第二個接聽程式 (HTTPS)。	雲端管理員

故障診斷

問題	解決方案
即使將 OpenSSL 命令添加到系統路徑中，Windows 也 PowerShell 無法識別它。	檢`$env:path` 查並確定其中包含 OpenSSL 二進位檔案的位置。 如果沒有，請在中執行下列命令 PowerShell： `$env:path = $env:path + ";C:\OpenSSL-Win64\bin"`

相關資源

將憑證匯入 ACM

- ACM 控制台
- 用於匯入的憑證和金鑰格式

- 匯入憑證
- AWS Certificate Manager 使用者指南

建立應用程式負載平衡器

- 建立應用程式負載平衡器
- Application Load Balancer 使用指南

將簡訊佇列從 Microsoft Azure 服務匯流排遷移到 Amazon SQS

R 類型：重新平台	來源：使用 Azure 服務總線隊列的應用程序	目標：Amazon SQS
創建者：AWS	環境：PoC 或試點	技術：Web 和移動應用程序; 遷移
工作量：Microsoft	AWS 服務：Amazon SQS	

Summary

此模式描述如何將 .NET 框架或 .NET 核心 Web 或主控台應用程式從使用 Microsoft Azure 服務匯流排佇列簡訊平台遷移到 Amazon Simple Queue Service (Amazon SQS)。

應用程式會使用訊息傳送服務，將資料傳送至其他應用程式，以及從中接收資料 這些服務有助於在雲端中建置分離、可高度擴充的微服務、分散式系統和無伺服器應用程式。

Azure 服務匯流排佇列是支援佇列和發佈/訂閱訊息的更廣泛 Azure 通訊基礎結構的一部分。

Amazon SQS 是全受管訊息佇列服務，可讓您分離和擴展微型服務、分散式系統和無伺服器應用程式。Amazon SQS 消除了與管理和操作訊息導向中介軟體相關的複雜性和開銷，並讓開發人員能夠專注於差異化工作。使用 Amazon SQS，您可以在任何磁碟區的軟體元件之間傳送、存放和接收訊息，而不會遺失訊息或需要其他服務可用。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 使用 Azure 服務匯流排佇列的 .NET 架構或 .NET 核心 Web 或主控台應用程式 (附加的範例程式碼)

產品版本

- . NET 框架 3.5 或更高版本，或. NET 核心 1.0.1，2.0.0 或更高版本

架構

源, 技術, 堆棧

- 使用 Azure 服務匯流排佇列傳送訊息的 .NET (核心或架構) Web 或主控台應用程式

目標技術堆疊

- Amazon SQS

工具

工具

- Microsoft Visual Studio

Code

若要為 Amazon SQS 建立 AWS Identity 和存取管理 (IAM) 政策：

1. 登入 AWS 管理主控台，然後前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。

2. 在左邊的導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。

3. 選擇 JSON 索引標籤，然後貼上下列程式碼：

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "sqs:DeleteMessage",
            "sqs:GetQueueUrl",
            "sqs:ChangeMessageVisibility",
            "sqs:SendMessageBatch",
            "sqs:ReceiveMessage",
            "sqs:SendMessage",
            "sqs:GetQueueAttributes",
            "sqs:ListQueueTags",
```

```
            "sqs:ListDeadLetterSourceQueues",
            "sqs:DeleteMessageBatch",
            "sqs:PurgeQueue",
            "sqs:DeleteQueue",
            "sqs:CreateQueue",
            "sqs:ChangeMessageVisibilityBatch",
            "sqs:SetQueueAttributes"
         ],
         "Resource": "arn:aws:sqs:*:<AccountId>:*"
      },
      {
         "Sid": "VisualEditor1",
         "Effect": "Allow",
         "Action": "sqs:ListQueues",
         "Resource": "*"
      }
   ]
}
```

4. 選擇 [檢閱原則]，輸入名稱，然後選擇 [建立原則]。

5. 將新建立的政策附加到現有的 IAM 角色，或建立新角色。

史诗

在 AWS 中設定 Amazon SQS

任務	描述	所需技能
為 Amazon SQS 建立 IAM 政策。	建立可提供 Amazon SQS 存取權的 IAM 政策。如需原則範例，請參閱程式碼一節。	系統工程師
建立 AWS 設定檔。	執行適用於 PowerShell 命令集的 AWS 工具來建立新的設定檔AWSCredential。此命令會將您的存取金鑰和密鑰儲存在您指定的設定檔名稱下的預設認證檔案中。將您先前建立的 Amazon SQS 政策與此帳戶連結。保留 AWS 存取金鑰 ID 和	系統工程師

任務	描述	所需技能
	秘密存取金鑰。在接下來的步驟中將需要這些步驟。	
建立 SQS 佇列。	您可以建立標準佇列或先進先出 (FIFO) 佇列。如需指示，請參閱「參照」一節中的連結。	系統工程師

修改 .NET 應用程式程式碼

任務	描述	所需技能
為視覺工作室安裝 AWS 工具組。	這個工具組是 Microsoft Visual Studio 的擴充功能，可讓您更輕鬆地在 AWS 中建置和部署 .NET 應用程式。如需安裝和使用說明，請參閱「參考資料」一節中的連結。	应用开发人
安裝 AWSSDK .SQS NuGet 套件。	您可以通過在 Visual Studio 中選擇「管理 Package」或通過運行命令「安裝 NuGet 包 AWSSDK .SQS」來安裝 AWSSDK .SQS。	应用开发人
在 .NET 應用程式中建立 AWSCredentials 物件。	附件中的範例應用程式顯示如何建立繼承自 AWSCredentials 的 Basic AWSCredentials 物件。您可以使用先前的存取金鑰 ID 和秘密存取金鑰，或讓物件在執行階段從 .aws 資料夾中挑選這些金鑰做為使用者設定檔的一部分。	应用开发人
建立 SQS 用戶端物件。	為 .NET 框架創建一個 SQS 客戶端對象（亞馬遜	应用开发人

任務	描述	所需技能
	SQLENT）。這是亞馬遜 .SQS 命名空間的一部分。這個對象是必需的，而不是我Queue Client，這是微軟的一部分。ServiceBus 命名空間。	
呼叫方 SendMessageAsync 法，將訊息傳送至 SQS 佇列。	變更將訊息傳送至佇列的程式碼以使用 amazonSqsClient. SendMessageAsync 方法。如需詳細資訊，請參閱隨附的程式碼範例。	应用开发人
呼叫方 ReceiveMessageAsync 法以接收來自 SQS 佇列的訊息。	變更接收訊息的程式碼以使用 amazonSqsClient. ReceiveMessageAsync 方法。如需詳細資訊，請參閱隨附的程式碼範例。	应用开发人
呼叫方 DeleteMessageAsync 法以刪除 SQS 佇列中的訊息。	若要刪除訊息，請變更佇列中的程式碼。 CompleteAsync 方法的 amazonSqsClient. DeleteMessageAsync 方法。如需詳細資訊，請參閱隨附的程式碼範例。	应用开发人

相關資源

- 適用於 .NET 開發人員的 AWS 開發套件
- 使用 Amazon SQS 進行簡訊
- 使用適用於 .NET 的 AWS 開發套件建立和使用 Amazon SQS 佇列
- 發送 Amazon SQS 消息
- 從 Amazon SQS 隊列接收消息
- 從 Amazon SQS 佇列中刪除訊息
- AWS Toolkit for Visual Studio

其他資訊

此模式包含兩個範例應用程式 (請參閱附件一節)：

- AzureSbTestApp包含使用 Azure 服務匯流排佇列的程式碼。
- AmazonSqsTestApp使用 Amazon SQS。這是一個使用 .NET Core 2.2 的控制台應用程序，包括用於發送和接收消息的示例。

備註：

- 隊列是我的一個對象QueueClient，這是微軟的 Azure 的一部分。 ServiceBus 命名空間 (包含在微軟. Azure 中。 ServiceBus NuGet 包）。
- amazonSqsClient 是 AmazonSQSClient 的一個對象，它是亞馬遜 .SQS 命名空間的一部分（包含在 .SQS 包中）。 AWSSDK NuGet
- 根據代碼運行的位置，例如，如果它在 EC2 上運行，角色需要具有寫入 SQS 隊列的權限。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Oracle 資料泵和 AWS DMS 將甲骨文 JD 愛德華資料 EnterpriseOne 庫遷移到 AWS

創建者：塔尼蓋威爾·特魯馬萊（AWS）

環境：生產	資料來源：甲骨文 JD 愛德華茲 EnterpriseOne	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS; AWS DMS		

Summary

您可以在 Amazon Relational EnterpriseOne Database Ser vice 服務 (Amazon RDS) 上遷移和執行 JD 愛德華資料庫。將資料庫遷移到 Amazon RDS 時，AWS 可以處理備份任務和高可用性設定，因此您可以專注於維護 EnterpriseOne 應用程式及其功能。有關遷移過程中要考慮的關鍵因素的完整清單，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

遷移 EnterpriseOne 資料庫的方式有多種，包括：

- 使用 Oracle 通用 Batch 引擎 (UBE) R98403 建立結構描述和表格，並使用 AWS Database Migration Service (AWS DMS) 進行遷移
- 使用資料庫原生工具建立結構描述和表格，並使用 AWS DMS 進行遷移
- 使用資料庫原生工具移轉現有資料 (滿載)，並使用 AWS DMS 進行變更資料擷取 (CDC) 任務

此模式涵蓋了第三個選項。本文說明如何透過搭配 AWS DMS 及其 CDC 功能使用 Oracle 資料泵，將您的現場部署資料 EnterpriseOne 庫遷移到 Amazon RDS for Oracle。

Oracle JD Edwards EnterpriseOne 是一種企業資源規劃 (ERP) 解決方案，適用於製造、建構、配送、服務或管理產品或實體資產的組織。JD Edwards EnterpriseOne 支援各種硬體、作業系統和資料庫平台。

當您遷移 JD Edwards 等關鍵 ERP 應用程式時 EnterpriseOne，將停機時間降至最低是關鍵。AWS DMS 透過支援從來源資料庫到目標資料庫的全負載和連續複寫，將停機時間降至最低。AWS DMS 也提供移轉的即時監控和記錄功能，協助您識別並解決任何可能導致停機的問題。

使用 AWS DMS 複寫變更時，必須指定時間或系統變更編號 (SCN) 做為從資料庫日誌讀取變更的起點。確保 AWS DMS 能夠存取這些日誌在指定的時間長度 (我們建議使用 15 天)，以確保 AWS DMS 能夠存取這些變更是至關重要的。

先決條件和限制

先決條件

- 在 AWS 雲端環境中佈建的適用於 Oracle 的亞馬遜 RDS 資料庫作為目標資料庫。如需指示，請參閱 Amazon RDS 文件。
- 在現場部署或 AWS 上 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上執行的 EnterpriseOne 資料庫。

  注意：此模式是專為從現場部署遷移到 AWS 而設計，但是已在 EC2 執行個體上使用 EnterpriseOne 資料庫進行測試。如果您打算從內部部署環境移轉，則必須設定適當的網路連線能力。
- 綱要詳細資料。識別您打算移轉的 Oracle 資料庫結構描述 (例如 DV920) EnterpriseOne。在開始移轉程序之前，請先收集有關結構描述的下列詳細資訊：
  - 綱要大小
  - 每個物件類型的物件數目
  - 無效物件的數目

限制

- 您必須在適用於 Oracle 資料庫的目標 Amazon RDS 上建立任何所需的結構描述，AWS DMS 不會為您建立這些結構描述。(Epics 一節說明如何使用「資料汲取」來匯出和匯入資料架構。) 目標 Oracle 資料庫的綱要名稱必須已存在。來源結構描述中的表格會匯入使用者或結構描述，而 AWS DMS 則使用管理員或系統帳戶連線到目標執行個體。若要遷移多個結構描述，您可以建立多個複寫任務。您也可以將資料移轉至目標執行個體上的不同結構描述。若要這麼做，請在 AWS DMS 表格對映上使用結構描述轉換規則。
- 這種模式已通過演示數據集進行了測試。建議您驗證資料集和自訂的相容性。
- 此模式會使用在 Microsoft 視窗上執行的 EnterpriseOne 資料庫。不過，您可以在 AWS DMS 支援的其他作業系統上使用相同的程序。

架構

下圖顯示在 Oracle 資料庫 EnterpriseOne 上執行的系統做為來源資料庫，以及將 Amazon RDS for Oracle 資料庫做為目標資料庫執行。資料會從來源 Oracle 資料庫匯出，然後使用 Oracle 資料泵將資料匯入目標 Amazon RDS for Oracle 資料庫，然後使用 AWS DMS 複寫以進行 CDC 更新。

1. Oracle 資料泵浦會從來源資料庫擷取資料，然後將資料傳送至 Amazon RDS for Oracle 資料庫目標。
2. CDC 資料會從來源資料庫傳送到 AWS DMS 中的來源端點。
3. 從來源端點，資料會傳送至執行複寫任務的 AWS DMS 複寫執行個體。
4. 複寫任務完成後，資料會傳送到 AWS DMS 中的目標端點。
5. 資料會從目標端點傳送至適用於 Oracle 資料庫執行個體的 Amazon RDS。

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。

其他服務

- 「Oracle 資料汲取」可協助您將資料和中繼資料從一個資料庫高速移動到另一個資料庫。

最佳實務

移轉 LOB

如果來源資料庫包含需要移轉至目標資料庫的大型二進位物件 (LOB)，AWS DMS 會提供下列選項：

- 完整 LOB 模式 — AWS DMS 會將所有 LOB 從來源移轉到目標資料庫，無論其大小為何。雖然遷移速度比其他模式慢，但優點是數據不會被截斷。為了獲得更好的效能，您可以在新的複寫執行個體上建立個別工作，以移轉具有 LOB 大於幾 MB 的資料表。

- 受限 LOB 模式 — 您可以指定 LOB 資料行資料的大小上限，這可讓 AWS DMS 預先配置資源並大量套用 LOB。如果 LOB 資料行的大小超過任務中指定的大小，AWS DMS 會截斷資料並將警告傳送到 AWS DMS 日誌檔。如果 LOB 資料大小在有限的 LOB 大小內，您可以使用有限的 LOB 模式來改善效能。
- 內嵌 LOB 模式 — 您可以透過複寫小型和大型 LOB 來移轉 LOB，而不會截斷資料或降低工作效能。首先，指定`InlineLobMaxSize`參數值，此值只有在完整 LOB 模式設定為時才可用`true`。AWS DMS 任務會以內嵌方式傳輸小型 LOB，這樣更有效率。然後，AWS DMS 會透過從來源資料表執行查閱來移轉大型 LOB。不過，內嵌 LOB 模式只能在滿載階段運作。

生成序列值

在 AWS DMS CDC 程序期間，不會從來源資料庫複寫增量序號。為避免序列值出現差異，您必須從所有序列的來源產生最新的序列值，並將其套用至目標 Amazon RDS for Oracle 資料庫。

AWS Secrets Manager

為了協助管理您的登入資料，我們建議您遵循部落格文章中的指示，使用 AWS Secrets Manager 管理您的 AWS DMS 端點登入資料。

效能

- 複寫執行個體-如需選擇最佳執行個體大小的指引，請參閱 AWS DMS 文件中的為複寫執行個體選取最佳大小。
- 連接選項-為了避免延遲問題，我們建議您選擇正確的連接選項。AWS Direct Connect 提供 AWS 資源的最短路徑，因為它是企業資料中心和 AWS 之間的專用連線。在傳輸過程中，您的網路流量會保留在 AWS 全球網路上，而且永遠不會透過網際網路。與使用 VPN 或公共互聯網相比，這可以減少遇到瓶頸或意外增加延遲的機會。
- 網路頻寬-若要最佳化效能，請確認您的網路輸送量是否快速。如果您在現場部署來源資料庫和 AWS DMS 之間使用 VPN 通道，請確保頻寬足以滿足您的工作負載。
- 任務 parallel 性-您可以通過在滿載期間並行加載多個表來加速數據複製。此病毒碼使用 RDBMS 端點，因此此選項僅適用於滿載程序。工作 parallel 處理原則由參數控制，該`MaxFullLoadSubTasks`參數決定平行執行的完整載入子工作數目。依預設，此參數設定為 8，這表示八個表格 (如果在表格對映中選取) 會在完整模式下一起載入。您可以在工作的 JSON 指令碼的完整載入工作設定區段中調整此參數。
- 資料表 parallel 處理原則 —AWS DMS 也可讓您使用多個平行執行緒來載入單一大型資料表。這對於具有數十億筆記錄的 Oracle 來源表格，以及多個分割區和子分割區特別有用。如果來源資料表未分割，您可以使用資料行界限進行 parallel 載入。

- 分割負載-當您將負載分割到多個任務或 AWS DMS 執行個體時，請在擷取變更時記住交易界限。

史诗

使用「Oracle 資料汲取管理系統」匯出 EnterpriseOne 綱要

任務	描述	所需技能
產生 SCN。	當來源資料庫處於作用中狀態並由 EnterpriseOne 應用程式正在使用時，請使用「Oracle 資料汲取」啟動資料匯出。您必須先從來源資料庫產生系統變更編號 (SCN)，以確保在使用 Oracle 資料泵匯出期間的資料一致性，以及做為 AWS DMS 中 CDC 的起點。 若要從來源資料庫產生目前的 SCN，請使用下列 SQL 陳述式： `SQL> select current_scn from v$database;` `CURRENT_SCN` `-----------` `30009727` 儲存產生的 SCN。匯出資料和建立 AWS DMS 複寫任務時，將會使用 SCN。	DBA
建立參數檔案。	若要建立匯出結構描述的參數檔案，您可以使用下列程式碼。	DBA

任務	描述	所需技能
	`directory=DMS_DATA_PUMP_DIR logfile=export_dms.log dumpfile=export_dms_data.dmp schemas=<schema name> flashback_scn=<SCN from previous command>` 附註：您也可以根據自己的需求，使用下列指令DATA_PUMP_DIR 來定義自己的指令。 `SQL> CREATE OR REPLACE DIRECTORY DMS_DATA_PUMP_DIR AS '<Directory for dump>'; Directory created.` `SQL> GRANT READ, WRITE ON DIRECTORY DMS_DATA_PUMP_DIR TO SYSTEM; Grant succeeded.`	

任務	描述	所需技能
匯出結構描述。	欲執行匯出，請使用公用expdp程式，如下所示： `C:\Users\Administrator>expdp system/********@<DB Name> PARFILE='<Path to PAR file create above>'` `Export: Release 19.0.0.0.0 - Production on *** *** ** **:**:** ****` `Version 19.3.0.0.0` `Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.` `Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production` `Starting "SYSTEM". "SYS_EXPORT_SCHEMA_02":  system/******@<DB Name>PARFILE='E:\exp_dms_datapump.par'` `Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA` `Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS` `Processing object type SCHEMA_EXPORT/TABL`	DBA

任務	描述	所需技能
	E/STATISTICS/TABLE _STATISTICS Processing object type SCHEMA_EXPORT/STAT ISTICS/MARKER Processing object type SCHEMA_EXPORT/USER Processing object type SCHEMA_EXPORT/ROLE _GRANT Processing object type SCHEMA_EXPORT/DEFA ULT_ROLE Processing object type SCHEMA_EXPORT/TABL ESPACE_QUOTA Processing object type SCHEMA_EXPORT/PRE_ SCHEMA/PROCACT_SCHEMA Processing object type SCHEMA_EXPORT/TABLE/ TABLE Processing object type SCHEMA_EXPORT/TABL E/GRANT/OWNER_GRANT/ OBJECT_GRANT Processing object type SCHEMA_EXPORT/TABLE/ INDEX/INDEX Processing object type SCHEMA_EXPORT/TABLE/ CONSTRAINT/CONSTRAINT . . exported "<Schema Name>"."<Table Name>" 228.9 MB 496397 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _02" successfully loaded/unloaded	

任務	描述	所需技能
	`` ********** ********** ********** ********** **** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_02 is: E:\DMSDUMP\EXPORT_ DMS_DATA.DMP Job "SYSTEM"."SYS_EXPO RT_SCHEMA_02" successfully completed at *** *** ** **:**:** **** elapsed 0 00:01:57 ```	

使用「Oracle 資料汲取」來匯入 EnterpriseOne 綱要

任務	描述	所需技能
將傾印檔案傳輸到目標執行個體。	若要使用公用DBMS_FILE_TRANSFER 程式傳輸檔案，您需要建立從來源資料庫到 Amazon RDS for Oracle 執行個體的資料庫連結。建立連結後，您可以使用公用程式將資料泵檔案直接傳輸到 Amazon RDS 執行個體。 或者，您可以將資料泵檔案傳輸到亞馬遜簡單儲存服務 (Amazon S3)，然後將它們匯入 Amazon RDS for Oracle 執行個體。如需有關此選項的詳細資訊，請參閱其他資訊一節。	DBA

任務	描述	所需技能
	若要建立連線至目標資料庫執行個體上 Amazon RDS 主要使用者的資料庫連結ORARDSDB，請在來源資料庫上執行下列命令： `sqlplus / as sysdba` `SQL*Plus: Release 19.0.0.0.0 on *** *** ** **:**:** **** Version 19.3.0.0.0` `Copyright (c) 1982, 2019, Oracle. All rights reserved.` `Connected to: Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 Version 19.3.0.0.0` `SQL> create database link orardsdb connect to admin identifie d by "******" using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = orcl.******.us-east-1.rds.amazonaws.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED ) (SERVICE_NAME = orcl)))';` `Database link created.` `SQL>`	

任務	描述	所需技能
測試資料庫連結。	測試資料庫連結，以確保您可以使用連線到 Amazon RDS 適用於 Oracle 目標資料庫`sqlplus`。 `SQL> select name from v$database@orardsdb;` `NAME` `---------` `ORCL`	DBA

任務	描述	所需技能
將傾印檔案傳輸到目標資料庫。	若要將傾印檔案複製到 Amazon RDS for Oracle 資料庫，您可以使用預設DATA_PUMP_DIR 目錄，也可以使用下列必須在目標 Amazon RDS 執行個體上執行的程式碼來建立自己的目錄： `exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'DMS_TARGET_PUMP_DIR');` `PL/SQL procedure successfully completed.` 下列指令碼會使用名 EXPORT_DMS_DATA.DMP 為的資料庫連結，將來源執行個體命名的傾印檔案複製到目標 Amazon RDS for Oracle 資料庫orardsdb。您必須在來源資料庫執行處理上執行指令集。 `BEGIN DBMS_FILE_TRANSFER.PUT_FILE( source_directory_object => 'DMS_DATA_PUMP_DIR', source_file_name => 'EXPORT_DMS_DATA.DMP',`	DBA

任務	描述	所需技能
	destination_directory_object => 'DMS_TARGET_PUMP_DIR', destination_file_name => 'EXPORT_DMS_DATA.DMP', destination_database => 'orardsdb'); END; PL/SQL procedure successfully completed.	
列出目標資料庫中的傾印檔案。	PL/SQL 程序完成後，您可以使用下列程式碼在 Amazon RDS 適用於甲骨文資料庫中列出資料傾印檔案： select * from table (rdsadmin.rds_file_util.listdir(p_directory => 'DMS_TARGET_PUMP_DIR'));	DBA

任務	描述	所需技能
在目標執行處理中建立 JDE 特定的使用者。	在目標執行個體中使用下列命令，建立 JD Edwards 設定檔和角色： `SQL> CREATE PROFILE "JDEPROFILE" LIMIT IDLE_TIME 15;` `Profile created.` `SQL> CREATE ROLE "JDE_ROLE";` `Role created.` `SQL> CREATE ROLE "JDEADMIN";` `CREATE ROLE "JDEUSER";` `Role created.` `Role created.` 將必要權限授與角色： `SQL> GRANT CREATE ANY SEQUENCE TO JDE_ROLE;` `GRANT DROP ANY SEQUENCE TO JDE_ROLE;` `GRANT CREATE ANY TRIGGER TO JDE_ROLE;` `GRANT DROP ANY TRIGGER TO JDE_ROLE;`	DBA, 傑德數控

任務	描述	所需技能
在目標執行處理中建立表格空間。	針對此移轉涉及的綱要使用下列命令，在目標執行處理中建立必要的表格空間： `SQL> CREATE TABLESPACE <Tablespace Name for Tables>;` `Tablespace created.` `SQL> CREATE TABLESPACE <Tablespace Name for Indexes>;` `Tablespace created.`	DBA, 傑德數控

任務	描述	所需技能
在目標資料庫上起始匯入。	在開始匯入程序之前，請使用資料傾印檔案在目標 Amazon RDS for Oracle 資料庫上設定角色、結構描述和表格空間。 若要執行匯入，請使用 Amazon RDS 主要使用者帳戶存取目標資料庫，然後使用`tnsnames.ora` 檔案中的連接字串名稱，其中包括 Amazon RDS for Oracle Database `tns-entry` 。如有必要，您可以加入重新對應選項，將資料傾印檔案匯入不同的表格空間或不同的綱要名稱。 若要開始匯入，請使用下列程式碼： `impdp admin@orardsdb directory=DMS_TARGET_PUMP_DIR logfile=import.log dumpfile=EXPORT_DMS_DATA.DMP` 若要確保匯入成功，請檢查匯入記錄檔案中是否有任何錯誤，並檢閱詳細資訊，例如物件計數、列計數和無效物件。如果有任何無效的物件，請重新編譯它們。此外，請比較來源和目標資料庫物件，以確認它們相符。	DBA

使用來源端點和目標端點佈建 AWS DMS 複寫執行個體

任務	描述	所需技能
下載 範本。	下載 AWS CloudFormation DMS 複寫執行個體及其來源和目標端點佈建 AWS DMS 複寫執行個體。	雲端管理員，DBA
開始建立堆疊。	1. 登入 AWS 管理主控台，然後開啟 AWS 主 CloudFormation 控台，網址為 https://console.aws.amazon.com/cloudformation。 2. 選擇建立堆疊。 3. 對於 Specify template (指定範本)，選擇 Upload a template file (上傳範本檔案)。 4. 選擇 [選擇檔案]。 5. 選擇`DMS_instance.yaml` 檔案。 6. 選擇下一步。	雲端管理員，DBA
指定參數。	1. 在堆疊名稱中，輸入您的堆疊名稱。 2. 對於 AWS DMS 執行個體參數，請輸入下列參數： • DMS InstanceType — 根據您的業務需求，為 AWS DMS 複寫執行個體選擇所需的執行個體。 • DMS StorageSize — 根據遷移的大小，輸入 AWS DMS 執行個體的儲存大小。	雲端管理員，DBA

任務	描述	所需技能
	3. 針對來源 Oracle 資料庫組態，輸入下列參數： • SourceOracleEndpoint識別碼 — 來源 Oracle 資料庫伺服器名稱 • SourceOracleDatabaseName— 來源資料庫服務名稱或階段作業 ID (SID) (如適用) • SourceOracleUserName— 來源資料庫使用者名稱 (預設值為`system`) • SourceOracle資料庫密碼 — 來源資料庫使用者名稱的密碼 • SourceOracle資料庫連接埠 — 來源資料庫連接埠 4. 針對 Oracle 資料庫組態的目標 RDS，輸入下列參數： • 目標OracleEndpoint識別碼 — 目標 RDS 資料庫端點 • 目標資料庫 OracleDatabaseName — 目標 RDS 資料庫名稱 • 目標使用者名稱 OracleUserName — 目標 RDS 使用者名稱 • 目標排序 B 密碼 — 目標 RDS 密碼 • TargetOracle資料庫連接埠 — 目標 RDS 資料庫連接埠	

任務	描述	所需技能
	5. 對於 VPC、子網路和安全群組組態，請輸入下列參數： • VPCID — 複寫執行個體的虛擬私人雲端 • VPC SecurityGroupId — 複寫執行個體的 VPC 安全性群組 • DMS 子網路 1 — 可用區域 1 的子網路 • DMS 子網路 2 — 可用區域 2 的子網路 6. 選擇下一步。	
建立堆疊。	1. 在 [設定堆疊選項] 頁面上，對於標籤，輸入任何選用值。 2. 選擇下一步。 3. 在 [複查] 頁面上，確認詳細資料，然後選擇 [送出]。 佈建應在大約 5-10 分鐘內完成。當 AWS CloudFormation 堆疊頁面顯示「建立 _ 完成」時，就會完成此操作。	雲端管理員，DBA
設定端點。	1. 開啟 AWS DMS 主控台，網址為 https://console.aws.amazon.com/dms/v2/。 2. 對於資源管理，請選擇複製執行個體，然後檢閱複製執行個體。 3. 對於資源管理，請選擇「端點」，然後檢閱端點。	雲端管理員，DBA

任務	描述	所需技能
測試連線能力。	在來源端點和目標端點顯示狀態為「作用中」之後，請測試連線。選擇為每個端點（來源和目標）執行測試，以確保狀態顯示為成功。	雲端管理員，DBA

為即時複寫建立 AWS DMS 複寫任務

任務	描述	所需技能
建立複寫工作。	使用下列步驟建立 AWS DMS 複寫任務： 1. 開啟 AWS DMS 主控台，網址為 https://console.aws.amazon.com/dms/v2/。 2. 在導覽窗格的「移轉資料」下，選擇「資料庫移轉工作」。 3. 在 [工作組態] 方塊中，對於 [工作識別碼]，輸入您的工作識別元。 4. 針對複寫執行個體，選擇您建立的 DMS 複製執行個體。 5. 針對來源資料庫端點，選擇您的來源端點。 6. 針對目標資料庫端點，選擇您的目標 Amazon RDS for Oracle 資料庫。 7. 針對 [移轉類型]，選擇 [僅複製資料變更]。如果您收到需要開啟補充記錄的訊息，請	雲端管理員，DBA

任務	描述	所需技能
	依照疑難排解一節中的指示進行。 8. 在 [工作設定] 方塊中，選擇 [指定記錄序號]。 9. 針對「系統變更編號」，輸入您從來源 Oracle 資料庫產生的 Oracle 資料庫 SCN。 10選擇「啟用驗證」。 11選擇「啟動 CloudWatch 記錄」。 啟用此功能後，您可以驗證資料和 Amazon CloudWatch 日誌以檢閱 AWS DMS 複寫執行個體日誌。 12在「選取規則」下，完成下列操作： • 對於綱要，請選擇輸入綱要。 • 在「綱要名稱」中，輸入 JDE 綱要名稱 (例如：DV920)。 • 在「表格名稱」中，輸入%。 • 在「動作」中選擇「包含」。 13選擇 Create task (建立任務)。 建立任務之後，AWS DMS 會將持續的變更從您在 CDC 啟動模式下提供的 SCN 遷移到 Amazon RDS for Oracle 資料	

任務	描述	所需技能
	庫執行個體。您也可以檢閱 CloudWatch 記錄檔來驗證移轉。	
重複複寫工作。	重複上述步驟，為屬於移轉一部分的其他 JD Edwards 結構描述建立複寫工作。	雲端管理員、DBA、JDE 數控管理員

驗證目標 Amazon RDS for Oracle 資料庫的資料庫結構描述

任務	描述	所需技能
驗證資料傳輸。	AWS DMS 任務開始後，您可以查看任務頁面上的 [表格統計資料] 索引標籤，以查看對資料所做的變更。 您可以在 [資料庫移轉工作] 頁面的主控台中監視進行中複寫的狀態。 如需詳細資訊，請參閱 AWS DMS 資料驗證。	雲端管理員，DBA

切過

任務	描述	所需技能
停止複寫。	中止複寫程序並停止來源應用程式服務。	雲端管理員，DBA
啟動 JD 愛德華茲應用程序。	在 AWS 上啟動目標 JD 愛德華簡報和邏輯層應用程式，並將其導向至適用於 Oracle 資料庫的 Amazon RDS。	DBA、JDE 數控管理員

任務	描述	所需技能
	當您存取應用程式時，您應該注意到所有連線現在都已透過 Amazon RDS for Oracle 資料庫建立。	
關閉來源資料庫。	確認沒有其他連線之後，您可以關閉來源資料庫。	DBA

故障診斷

問題	解決方案
您會收到警告訊息，啟用來源資料庫中的補充記錄日誌以進行中的複製	輸入下列命令以啟用補充記錄日誌： `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;` `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;` `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;` `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;` `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;` `SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;`
AWS DMS 已關閉補充記錄功能。	AWS DMS 中預設會關閉補充記錄。若要為來源 Oracle 端點開啟它，請執行下列動作： 1. 登入 AWS 管理主控台，然後開啟 AWS DMS 主控台，網址為 https://console.aws.amazon.com/dms/v2/。 2. 選擇端點。 3. 選擇要新增補充記錄的 Oracle 來源端點。 4. 選擇 Modify (修改)。

問題	解決方案
	5. 選擇進階，然後將下列程式碼新增至額外連線屬性文字方塊： `addSupplementalLogging=Y` 6. 選擇 Modify (修改)。
在 CDB 層級未啟用補充記錄。	1. 輸入此命令： `SQL> alter session set container = CDB$ROOT;` `Session altered.` 2. 重複這些步驟來啟用補充記錄日誌。
您收到錯誤訊息：「測試端點失敗：應用程式狀態：1020912，應用程式訊息：Oracle PDB 環境中 LogMiner 不支援端點初始化失敗」。	如果您遇到此錯誤訊息，您可以使用二進位讀取器來取代 LogMiner。 在「端點設定」 下，將此行新增至來源資料庫的額外連線屬性： `useLogMinerReader=N;useBfile=Y;`

相關資源

- 開始使用 AWS Database Migration Service
- AWS Database Migration Service 的最佳實務
- 將甲骨文資料庫遷移到 AWS 雲端
- 適用於 AWS 的 AWS Database Migration Service 資源類型參考 CloudFormation
- 使用 AWS 秘密管理員管理您的 AWS DMS 端點登入資料
- AWS Database Migration Service 中的移轉任務疑難排
- AWS Database Migration Service 的最佳實務

其他資訊

使用 Amazon S3 傳輸文件

若要將檔案傳輸到 Amazon S3，您可以使用 AWS CLI 或 Amazon S3 主控台。將檔案傳輸到 Amazon S3 之後，您可以使用 Amazon RDS for Oracle 文執行個體從 Amazon S3 匯入資料泵檔案。

如果您選擇使用 Amazon S3 整合作為替代方法傳輸傾印檔案，請執行下列步驟：

1. 建立 S3 儲存貯體。
2. 使用「Oracle 資料汲取」，從來源資料庫匯出資料。
3. 將資料汲取檔案上傳到 S3 儲存貯體。
4. 將資料汲取檔案從 S3 儲存貯體下載到目標 Amazon RDS for Oracle 文資料庫。
5. 使用「資料汲取」檔案執行匯入。

備註：若要在 S3 和 RDS 執行個體之間傳輸大型資料檔案，建議您使用 Amazon S3 Transfer Acceleration 功能。

使用 AWS DMS 將甲骨文 PeopleSoft 資料庫遷移到 AWS

環境：生產	來源：甲骨文 PeopleSoft	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：AWS DMS; Amazon RDS		

Summary

Oracle PeopleSoft 是針對整個企業流程的企業資源規劃 (ERP) 解決方案。 PeopleSoft 具有三層架構：客戶端，應用程序和數據庫。 PeopleSoft 可以在 Amazon 關係數據庫服務（亞馬遜 RDS）上運行。

如果您將 Oracle 資料庫遷移到 Amazon RDS，Amazon Web Services (AWS) 可以處理備份任務和高可用性，讓您可以自由地專注於維護 PeopleSoft 應用程式及其功能。有關遷移過程中要考慮的關鍵因素的完整清單，請參閱 AWS Prescriptive Guidance 中的 Oracle 資料庫遷移策略。

此模式提供了一個解決方案，可讓您使用 Oracle 資料泵搭配 AWS Database Migration Service (AWS DMS) 及其變更資料擷取 (CDC) 功能，將現場部署 Oracle 資料庫遷移至 Amazon RDS for Oracle。

移轉重要的 ERP 應用程式 (例如 Oracle) 時 PeopleSoft，將停機時間降至最低是關鍵。AWS DMS 透過支援全負載和連續複寫，將停機時間降至最低。從來源資料庫到目標資料庫。AWS DMS 還提供移轉的即時監控和記錄功能，協助您識別並解決任何可能導致停機的問題。

使用 AWS DMS 複寫變更時，您必須指定時間或系統變更編號 (SCN) 做為 AWS DMS 讀取資料庫日誌變更的起點。確保 AWS DMS 能夠存取這些日誌在伺服器上存取一段指定的時間至關重要。

先決條件和限制

先決條件

- 在您的 AWS 雲端環境中佈建適用於 Oracle 資料庫的亞馬遜 RDS 做為目標資料庫。
- 在現場部署或 AWS 雲端的亞馬遜彈性運算雲端 (Amazon EC2) 上執行的 Oracle PeopleSoft 資料庫。

注意：此模式是專為從現場部署遷移到 AWS 而設計的，但已在 Amazon EC2 執行個體上使用 Oracle 資料庫進行測試。若要從內部部署移轉，您需要設定適當的網路連線。

- 綱要詳細資料。將 Oracle PeopleSoft 應用程式遷移到亞馬遜 RDS 適用於甲骨文時，必須確定要遷移的 Oracle 資料庫結構描述 (例如SYSADM)。在開始移轉程序之前，請先收集有關結構描述的下列詳細資訊：
  - 大小
  - 每個物件類型的物件數目
  - 無效物件的數目。

  此資訊將有助於移轉程序。

限制

- 這種情況已經過測試只與 PeopleSoft DEMO 數據庫。它尚未通過大型數據集進行測試。

架構

下圖顯示將 Oracle 資料庫做為來源資料庫執行的執行個體，以及將 Amazon RDS for Oracle 資料庫作為目標資料庫執行。系統會使用 Oracle 資料泵將資料從來源 Oracle 資料庫匯出並匯入目標 Amazon RDS for Oracle 資料庫，然後使用 AWS DMS 複寫以進行 CDC 變更。

1. 初始步驟涉及使用 Oracle 資料泵從來源資料庫擷取資料，然後將資料傳送到 Amazon RDS for Oracle 資料庫目標。
2. 資料會從來源資料庫傳送到 AWS DMS 中的來源端點。
3. 從來源端點，資料會傳送至執行複寫任務的 AWS DMS 複寫執行個體。
4. 複寫任務完成後，資料會傳送到 AWS DMS 中的目標端點。
5. 資料會從目標端點傳送至適用於 Oracle 資料庫執行個體的 Amazon RDS。

工具

AWS 服務

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。

- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。

其他服務

- Oracle 資料泵取可協助您以高速將資料和中繼資料從一個資料庫移至另一個資料庫。

最佳實務

移轉 LOB

如果來源資料庫包含需要移轉至目標資料庫的大型二進位物件 (LOB)，AWS DMS 會提供下列選項：

- 完整 LOB 模式 — AWS DMS 會將所有 LOB 從來源移轉到目標資料庫，無論其大小為何。雖然遷移速度較慢，但優點是數據不會被截斷。為了獲得更好的效能，您可以在新的複寫執行個體上建立個別工作，以移轉 LOB 大於幾 MB 的表格。
- 受限 LOB 模式 — 您可以指定 LOB 資料行資料的大小上限，這可讓 AWS DMS 預先配置資源並大量套用 LOB。如果 LOB 資料行的大小超過任務中指定的大小，AWS DMS 會截斷資料並將警告傳送到 AWS DMS 日誌檔。如果 LOB 資料大小在有限 LOB 大小內，您可以使用受限 LOB 模式來改善效能。
- 內嵌 LOB 模式 — 您可以透過複寫小型和大型 LOB 來移轉 LOB，而不會截斷資料或降低工作效能。首先，指定 InlineLobMaxSize 參數的值，此值只有在「完整 LOB」模式設定為 true 時才可用。AWS DMS 任務會以內嵌方式傳輸小型 LOB，這樣更有效率。然後，AWS DMS 會透過從來源資料表執行查閱來移轉大型 LOB。不過，內嵌 LOB 模式只能在滿載階段運作。

生成序列值

請記住，在使用 AWS DMS 進行變更資料擷取程序期間，不會從來源資料庫複寫增量序號。為避免序列值出現差異，您必須從所有序列的來源產生最新的序列值，並將其套用至目標 Amazon RDS for Oracle 資料庫。

憑證管理

為了協助保護您的 AWS 資源，我們建議遵循 AWS Identity and Access Management (IAM) 的最佳實務。

史诗

使用來源端點和目標端點佈建 AWS DMS 複寫執行個體

任務	描述	所需技能
下載 範本。	下載 DMS_Instance AWS CloudFormation 範本以佈建 AWS DMS 複寫執行個體及其來源和目標端點。	雲端管理員，DBA
開始建立堆疊。	1. 在 AWS 管理主控台上，選擇CloudFormation。 2. 選擇建立堆疊。 3. 對於 Specify template (指定範本)，選擇 Upload a template file (上傳範本檔案)。 4. 選擇 [選擇檔案]。 5. 選擇`DMS_instance.yaml` 檔案。 6. 選擇下一步。	雲端管理員，DBA
指定參數。	1. 在堆疊名稱中，輸入您的堆疊名稱。 2. 在 AWS DMS 執行個體參數下，輸入下列參數： • DMS InstanceType — 根據您的業務需求，為 AWS DMS 複寫執行個體選擇所需的執行個體。 • DMS StorageSize — 根據遷移的大小，輸入 AWS DMS 執行個體的儲存大小。	雲端管理員，DBA

任務	描述	所需技能
	3. 在來源 Oracle 資料庫組態下，輸入下列參數： • SourceOracle端點 ID — 來源 Oracle 資料庫伺服器名稱 • SourceOracleDatabaseName— 來源資料庫服務名稱或階段作業 ID (SID) (如適用) • SourceOracleUserName— 來源資料庫使用者名稱 (預設為系統) • SourceOracle資料庫密碼 — 來源資料庫使用者名稱的密碼 • SourceOracle資料庫連接埠 — 來源資料庫連接埠 4. 在 Oracle 資料庫組態的目標 RDS 下，輸入下列參數： • 目標OracleEndpoint識別碼 — 目標 RDS 資料庫端點 • 目標資料庫OracleDatabase名稱 — 目標 RDS 資料庫名稱 • 目標OracleUser名稱 — 目標 RDS 使用者名稱 • 目標排序 B 密碼 — 目標 RDS 密碼 • TargetOracle資料庫連接埠 — 目標 RDS 資料庫連接埠	

任務	描述	所需技能
	5. 在 VPC、子網路和安全群組組態下，輸入下列參數： • VPCID — 複寫執行個體的虛擬私人雲端 • VPC SecurityGroup ID — 複寫執行個體的 VPC 安全性群組 • DMS 子網路 1 — 可用區域 1 的子網路 • DMS 子網路 2 — 可用區域 2 的子網路 6. 選擇下一步。	
建立堆疊。	1. 在 [設定堆疊選項] 頁面上，對於標籤，輸入任何選用值。 2. 選擇下一步。 3. 在 [複查] 頁面上，確認詳細資料，然後選擇 [送出]。 佈建應在大約 5-10 分鐘內完成。當 AWS CloudFormation 堆疊頁面顯示「建立 _ 完成」時，就會完成此操作。	雲端管理員，DBA
設定端點。	1. 在 AWS 管理主控台中，選擇資料庫遷移服務。 2. 在資源管理下，選擇複製執行個體。 3. 在資源管理下，選擇端點。	雲端管理員，DBA

任務	描述	所需技能
測試連線能力。	在來源端點和目標端點顯示狀態為「作用中」之後，請測試連線。選擇為每個端點（來源和目標）執行測試，以確保狀態顯示為成功。	雲端管理員，DBA

使用 Oracle 資料汲取，從內部部署 Oracle 資料庫匯出 PeopleSoft 結構描述

任務	描述	所需技能
產生 SCN。	當來源資料庫處於作用中狀態且應用程式正在使用時，請使用「Oracle 資料汲取」啟動資料匯出。您必須先從來源資料庫產生系統變更編號 (SCN)，以確保在 Oracle 資料泵匯出期間的資料一致性，以及做為 AWS DMS 中擷取變更資料的起點。 若要從來源資料庫產生目前的 SCN，請輸入下列 SQL 敘述句。 `SQL> select name from v$database;` `SQL> select name from v$database;` `NAME` `---------` `PSFTDMO` `SQL> SELECT current_scn FROM v$database;` `CURRENT_SCN` `-----------` `23792008`	DBA

任務	描述	所需技能
	儲存產生的 SCN，以便在匯出資料和建立 AWS DMS 複寫任務時使用。	

任務	描述	所需技能
建立參數檔案。	若要建立匯出結構描述的參數檔案，您可以使用下列程式碼。 `$ cat exp_datapmp.par` `userid=system/*******` `directory=DATA_P` `UMP_DIR` `logfile=export_dms_` `sample_user.log` `dumpfile=export_dms_` `sample_data_%U.dmp` `schemas=SYSADM` `flashback_scn=237920` `08` 附註：您也可以根據自己的需求，使用下列指令DATA_PUMP_DIR 來定義自己的指令。 `SQL> CREATE OR REPLACE` ` DIRECTORY DATA_PUMP` `_DIR AS '/opt/oracle/` `product/19c/dbhome_1/` `dmsdump/';` `Directory created.` `SQL> GRANT READ, WRITE` ` ON DIRECTORY DATA_PUMP` `_DIR TO system;` `Grant succeeded.` `SQL>` `SQL> SELECT owner,` ` directory_name,` ` directory_path FROM` ` dba_directories WHERE` ` directory_name='DA` `TA_PUMP_DIR';` `OWNER DIRECTORY_NAME` ` DIRECTORY_PATH`	DBA

任務	描述	所需技能
	`` ---------- ---------- ---------- ---------- ---------- ---------- SYS DATA_PUMP_DIR /opt/ oracle/product/19c/dbh ome_1/dmsdump/ ```	

任務	描述	所需技能
匯出結構描述。	若要執行匯出，請使用expdp公用程式。 `$ expdp parfile=exp_datapmp.par` `.....` `Transferring the dump file with DBMS_FILE_TRANSFER to Target:` `. . exported "SYSADM"."PS_XML_TEMPLT_LNG" 6.320 KB 0 rows` `. . exported "SYSADM"."PS_XML_TEMPLT_LNK" 6.328 KB 0 rows` `. . exported "SYSADM"."PS_XML_XLATDEF_LNG" 6.320 KB 0 rows` `. . exported "SYSADM"."PS_XML_XLATITM_LNG" 7.171 KB 0 rows` `. . exported "SYSADM"."PS_XPQRYRUNCNTL" 7.601 KB 0 rows` `. . exported "SYSADM"."PS_XPQRYRUNPARM" 7.210 KB 0 rows` `. . exported "SYSADM"."PS_YE_AMOUNTS" 9.351 KB 0 rows` `. . exported "SYSADM"."PS_YE_DATA" 16.58 KB 0 rows` `. . exported "SYSADM"."PS_YE_EE" 6.75 KB 0 rows` `. . exported "SYSADM"."PS_YE_W2CP_AMOUNTS" 9.414 KB 0 rows`	DBA

任務	描述	所需技能
	. . exported "SYSADM". "PS_YE_W2CP_DATA" 20.94 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_AMOUNTS" 10.27 KB 0 rows . . exported "SYSADM". "PS_YE_W2C_DATA" 20.95 KB 0 rows . . exported "SYSADM". "PS_ZBD_JOBCODE_TBL" 14.60 KB 0 rows . . exported "SYSADM". "PTGRANTTBL" 5.468 KB 0 rows Master table "SYSTEM". "SYS_EXPORT_SCHEMA _01" successfully loaded/unloaded ** Dump file set for SYSTEM.SYS_EXPORT_ SCHEMA_01 is: /opt/oracle/pr oduct/19c/dbhome_1 /dmsdump/export_dm s_sample_data_01.dmp Job "SYSTEM"."SYS_EXPO RT_SCHEMA_01" successfully completed at Mon Dec 19 20:13:57 2022 elapsed 0 00:38:22	

使用 Oracle 資料泵將 PeopleSoft 架構匯入亞馬遜 RDS 資料庫

任務	描述	所需技能
將傾印檔案傳輸到目標執行個體。	若要使用傳輸檔案DBMS_FILE_TRANSFER ，您需要建立	DBA

任務	描述	所需技能
	從來源資料庫到 Amazon RDS for Oracle 執行個體的資料庫連結。建立連結後，您可以使用公用程式將「資料汲取」檔案直接傳輸到 RDS 執行個體。 或者，您可以將資料泵檔案傳輸到亞馬遜簡單儲存服務 (Amazon S3)，然後將它們匯入 Amazon RDS for Oracle 執行個體。如需有關此選項的詳細資訊，請參閱其他資訊一節。 若要建立連線至目標資料庫執行個體上 Amazon RDS 主要使用者的資料庫連結ORARDSDB，請在來源資料庫上執行下列命令。 `$sqlplus / as sysdba` `$ SQL> create database link orardsdb connect to admin identified by "*****" using '(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = testpsft.*******.us-west-2.rds.amazonaws.com)(PORT = 1521))(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl)))';` `Database link created.`	

任務	描述	所需技能
測試資料庫連結。	測試資料庫連結，以確保您可以使用 sqlplus 連線到適用於 Oracle 的亞馬遜 RDS 目標資料庫。 `SQL>` `SQL> select name from v$database@orardsdb;` `NAME` `---------` `ORCL` `SQL>`	DBA

任務	描述	所需技能
將傾印檔案傳輸到目標資料庫。	若要將傾印檔案複製到 Amazon RDS for Oracle 資料庫，您可以使用預設DATA_PUMP_DIR 目錄，也可以使用下列程式碼建立自己的目錄。 `exec rdsadmin.rdsadmin_util.create_directory(p_directory_name => 'TARGET_PUMP_DIR') ;` 下列指令碼會使用名 `export_dms_sample_data_01.dmp` 為的資料庫連結，將來源執行個體命名的傾印檔案複製到目標 Amazon RDS for Oracle 資料庫orardsdb。 `$ sqlplus / as sysdba SQL> BEGIN DBMS_FILE_TRANSFER.PUT_FILE( source_directory_object => 'DATA_PUMP_DIR', source_file_name => 'export_dms_sample_data_01.dmp', destination_directory_object => 'TARGET_PUMP_DIR'', destination_file_name => 'export_dms_sample_data_01.dmp',`	DBA

任務	描述	所需技能
	destination_database => 'orardsdb'); END; / PL/SQL procedure successfully completed .	
列出目標資料庫中的傾印檔案。	PL/SQL 程序完成後，您可以使用下列程式碼在 Amazon RDS 適用於甲骨文資料庫中列出資料傾印檔案。 SQL> select * from table (rdsadmin.rds_file _util.listdir(p_di rectory => 'TARGET_P UMP_DIR'));	DBA

任務	描述	所需技能
在目標資料庫上起始匯入。	在開始匯入程序之前，請使用資料傾印檔案在目標 Amazon RDS for Oracle 資料庫上設定角色、結構描述和表格空間。 若要執行匯入，請使用 Amazon RDS 主要使用者帳戶存取目標資料庫，然後使用`tnsnames.ora` 檔案中的連接字串名稱，其中包括 Amazon RDS for Oracle Database `tns-entry` 。如有必要，您可以加入重新對應選項，將資料傾印檔案匯入不同的表格空間或不同的綱要名稱。 若要開始匯入，請使用下列程式碼。 `impdp admin@orardsdb directory=TARGET_PUMP_DIR logfile=import.log dumpfile=export_dms_sample_data_01.dmp` 若要確保匯入成功，請檢查匯入記錄檔是否有任何錯誤，並檢閱詳細資訊，例如物件計數、列計數和無效物件。如果有任何無效的物件，請重新編譯它們。此外，請比較來源和目標資料庫物件，以確認它們相符。	DBA

使用 CDC 建立 AWS DMS 複寫任務以執行即時複寫

任務	描述	所需技能
建立複寫工作。	使用下列步驟建立 AWS DMS 複寫任務： 1. 在 AWS DMS 主控台的 [轉換和遷移] 下，選擇 [資料庫遷移任務]。 2. 在「工作組態」下，針對「工作識別元」輸入您的工作識別元 3. 針對複寫執行個體，選擇您建立的 DMS 複製執行個體。 4. 針對來源資料庫端點，選擇您的來源端點。 5. 針對目標資料庫端點，選擇您的目標 Amazon RDS for Oracle 資料庫。 6. 針對 [移轉類型]，選擇 [僅複製資料變更]。如果您收到需要開啟補充記錄的訊息，請遵循其他資訊一節中的指示。 7. 在 [工作設定] 下，選取 [指定記錄序號]。 8. 針對「系統變更編號」，輸入您從來源 Oracle 資料庫產生的 Oracle 資料庫 SCN。 9. 選擇「啟用驗證」。 10選擇「啟動 CloudWatch 記錄」。	雲端管理員，DBA

任務	描述	所需技能
	啟用此功能後，您可以驗證資料和 Amazon CloudWatch 日誌以檢閱 AWS DMS 複寫執行個體日誌。 11.在「選取規則」下，完成下列操作： • 對於綱要，請選擇輸入綱要。 • 針對綱要名稱，輸入 SYSADM。 • 在「表格名稱」中，輸入%。 • 在「動作」中選擇「包含」。 12.在轉換規則下，完成下列操作： • 選擇「表格」做為「目標」。 • 選擇輸入結構描述做為配置名稱。 • 針對綱要名稱，輸入 SYSADM。 • 在「動作」中選擇「重新命名為」 13選擇 Create task (建立任務)。 建立任務之後，它會將 CDC 從您在 CDC 啟動模式下提供的 SCN 移轉到適用於 Oracle 的 Amazon RDS 資料庫執行個	

任務	描述	所需技能
	體。您也可以檢閱 CloudWatch 記錄檔來進行驗證。	

驗證目標 Amazon RDS for Oracle 資料庫的資料庫結構描述

任務	描述	所需技能
驗證資料傳輸。	AWS DMS 任務開始後，您可以查看任務頁面上的 [表格統計資料] 索引標籤，以查看對資料所做的變更。 您可以在 [資料庫移轉工作] 頁面的主控台中監視進行中複寫的狀態。 如需詳細資訊，請參閱 AWS DMS 資料驗證。	雲端管理員，DBA

切過

任務	描述	所需技能
停止複寫。	停止複寫程序並停止來源應用程式服務。	雲端管理員，DBA
啟動 PeopleSoft 中間層。	在 AWS 中啟動目標 PeopleSoft 中間層應用程式，並將其導向至最近遷移的 Amazon RDS for Oracle 料庫。 當您存取應用程式時，您應該注意到所有應用程式連線現	DBA， PeopleSoft 系統管理員

任務	描述	所需技能
	在都是透過 Amazon RDS for Oracle 資料庫建立的。	
關閉來源資料庫。	確認來源資料庫沒有其他連線之後，就可以將其關閉。	DBA

相關資源

- 開始使用 AWS Database Migration Service
- AWS Database Migration Service 的最佳實務
- 將甲骨文資料庫遷移到 AWS 雲端

其他資訊

使用 Amazon S3 傳輸文件

若要將檔案傳輸到 Amazon S3，您可以使用 AWS CLI 或 Amazon S3 主控台。將檔案傳輸到 Amazon S3 之後，您可以使用 Amazon RDS for Oracle 執行個體從 Amazon S3 匯入資料泵檔案。

如果您選擇使用 Amazon S3 整合作為替代方法傳輸傾印檔案，請執行下列步驟：

1. 建立 S3 儲存貯體。
2. 使用「Oracle 資料汲取」，從來源資料庫匯出資料。
3. 將資料汲取檔案上傳到 S3 儲存貯體。
4. 將資料汲取檔案從 S3 儲存貯體下載到目標 Amazon RDS for Oracle 文資料庫。
5. 使用「資料汲取」檔案執行匯入。

注意：若要在 S3 和 RDS 執行個體之間傳輸大型資料檔案，建議使用 Amazon S3 Transfer Acceleration 功能。

啟動補充記錄

如果您收到警告訊息，要求在來源資料庫中啟用補充記錄日誌以進行中的複製，請使用下列步驟。

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
```

將現場部署 MySQL 資料庫遷移到 Amazon RDS for MySQL

創建者：洛倫佐·莫塔（AWS）

環境：PoC 或試點	來源：內部部署 MySQL 數據庫	目標：適用於 MySQL 的 Amazon RDS for MySQL
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式提供將現場部署 MySQL 資料庫遷移至 MySQL 專用 Amazon Relational Database Service 服務 (Amazon RDS) 的指引。該模式討論如何使用 AWS Database Migration Service (AWS DMS) 或原生 MySQL 工具 (例如 mysqldbcopy 和 mysq l dump) 進行完整的資料庫遷移。此模式主要適用於 DBA 和解決方案架構師。它可以在小型或大型專案中用作測試程序 (我們建議至少一個測試週期) 或作為最終移轉程序。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的 MySQL 來源資料庫

限制

- 資料庫大小限制：64 TB

產品版本

- MySQL 版本 5.5, 5.7, 8.0. 如需支援版本的最新清單，請參閱 AWS 文件中的 Amazon RDS 上的 MySQL。如果您使用的是 AWS DMS，另請參閱使用與 MySQL 相容的資料庫做為 AWS DMS 目前支援的 MySQL 版本的 AWS DMS 的目標。

架構

源, 技術, 堆棧

- 內部部署 MySQL 資料庫

目標技術堆疊

- 執行 MySQL 的 Amazon RDS 資料庫執行個體

目標架構

下圖顯示遷移後實作的目標 Amazon RDS for MySQL 版。

AWS 資料遷移架構

使用 AWS DMS：

下圖顯示當您使用 AWS DMS 傳送完整變更和增量變更直到切換時的資料遷移架構。從內部部署到 AWS 的網路連線取決於您的需求，並且超出此模式的範圍。

使用本地 MySQL 工具：

下圖顯示了當您使用本地 MySQL 工具的數據遷移體系結構。匯出傾印檔案會複製到亞馬遜簡單儲存服務 (Amazon S3)，並在切換之前匯入 AWS 中的 Amazon RDS for MySQL 資料庫。從內部部署到 AWS 的網路連線取決於您的需求，並且超出此模式的範圍。

備註：

- 根據停機時間需求和資料庫的大小，使用 AWS DMS 或變更資料擷取 (CDC) 工具可將切換時間降至最低。AWS DMS 可協助將新目標的切換時間縮短到最少 (通常是分鐘)。 如果數據庫的大小和網絡延遲允許一個短窗口，那麼使用 mysqldump 或 mysqldbcopy 的離線策略就足夠了。（我們建議進行測試以獲得大約的時間。）
- 與離線選項相比，AWS DMS 之類的 CDC 策略通常需要更多的監控和複雜性。

工具

- AWS 服務：AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。如需 AWS DMS 支援的 MySQL 來源和目標資料庫的詳細資訊，請參閱將與 MySQL 相容的資料庫遷移到 AWS。如果 AWS DMS 不支援您的來源資料庫，則必須選擇其他方法來遷移資料。
- 本地 MySQL 工具：神秘的數據庫複製和神秘的轉儲
- 第三方工具：佩爾科納 XtraBackup

史诗

規劃移轉

任務	描述	所需技能
驗證資料庫版本。	驗證來源和目標資料庫版本。	DBA
識別硬體需求。	識別目標伺服器的硬體需求。	DBA, 系統管理員
識別儲存需求。	識別目標資料庫的儲存需求 (例如儲存類型和容量)。	DBA, 系統管理員
選擇執行個體類型。	根據容量、儲存區功能和網路功能選擇目標執行個體類型。	DBA, 系統管理員
識別網路存取需求。	識別來源和目標資料庫的網路存取安全需求。	DBA, 系統管理員
識別不支援的物件。	識別不受支援的物件 (如果有的話)，並決定移轉時間。	DBA
識別依賴關係。	識別遠端資料庫上的任何相依性。	DBA
決定應用程式移轉策略。	決定移轉用戶端應用程式的策略。	DBA、應用程式擁有者、系統管理員

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)	設定路由表、網際網路閘道、NAT 閘道和子網路。如需詳細資訊，請參閱 Amazon RDS 文件中的 VPC 和 Amazon RDS。	系統管理員
建立安全性群組。	根據您的需求設定連接埠和 CIDR 範圍或特定 IP。MySQL 的預設連接埠為 3306。如需詳細資訊，請參閱 Amazon RDS 說明文件中的使用安全群組控制存取。	系統管理員
設定並啟動 Amazon RDS for MySQL 版資料庫執行個體。	如需指示，請參閱 Amazon RDS 文件中的建立 Amazon RDS 資料庫執行個體。檢查支援的版本。	系統管理員

移轉資料-選項 1 (使用原生工具)

任務	描述	所需技能
使用原生 MySQL 工具或協力廠商工具來移轉資料庫物件和資料。	有關說明，請參閱 MySQL 工具的文檔，例如 mysqldbcopy，mysqldump 和佩爾科納（用於物理遷移）。XtraBackup 如需有關選項的詳細資訊，請參閱部落格文章將 MySQL 移轉到 Amazon RDS for MySQL (MySQL 版) 或 Amazon Aurora MySQL 的移轉選項。	DBA

遷移資料-選項 2 (使用 AWS DMS)

任務	描述	所需技能
使用 AWS DMS 遷移資料。	如需相關指示，請參閱 AWS DMS 文件。	DBA

切換前執行初步工作

任務	描述	所需技能
修正物件計數差異。	從來源資料庫和新的目標資料庫收集物件計數。修正目標資料庫中的差異。	DBA
檢查依賴關係。	檢查與其他數據庫之間的依賴關係（鏈接）是否有效並按預期工作。	DBA
執行測試。	如果這是測試週期，請執行查詢測試、收集指標並修正問題。	DBA

切過

任務	描述	所需技能
切換到目標資料庫。	將用戶端應用程式切換至新基礎結構。	DBA、應用程式擁有者、系統管理員
提供測試支持。	為功能應用程序測試提供支持。	DBA

關閉專案

任務	描述	所需技能
關閉資源。	關閉您為遷移建立的臨時 AWS 資源。	DBA, 系統管理員
驗證專案文件。	審核並驗證專案文件。	DBA、應用程式擁有者、系統管理員
收集指標。	收集指標，例如移轉時間、手動與自動化工作的百分比、節省成本等。	DBA、應用程式擁有者、系統管理員
關閉專案。	關閉專案並提供意見反應。	DBA、應用程式擁有者、系統管理員
解除委任來源資料庫。	完成所有移轉和切換工作後，請解除內部部署資料庫的委任。	DBA, 系統管理員

相關資源

參考

- 關聯式資料庫的遷移策略
- 數據管理系統網站
- AWS DMS 文件集
- Amazon RDS 文件
- Amazon RDS 定價
- VPC 和 Amazon RDS
- Amazon RDS 異地備份部署
- 使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL

教程

- 開始使用 AWS DMS
- Amazon RDS 入門

將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server

創建者：恩里克·洛保（AWS），喬納森·佩雷拉·克魯茲（AWS）和維沙爾辛格（AWS）

環境：PoC 或試點	來源：Microsoft SQL 服務器	目標：Amazon RDS for SQL Server
R 類型：重新平台	工作量：Microsoft	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式提供從現場部署 Microsoft SQL 伺服器資料庫遷移到適用於 SQL 伺服器的 Amazon Relational Database Service 服務 (Amazon RDS) 的指引。它描述了兩個遷移選項：使用 AWS 資料遷移服務 (AWS DMS) 或使用原生 Microsoft SQL Server 工具，例如複製資料庫精靈。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 Microsoft SQL 伺服器資料庫

限制

- 資料庫大小限制：16 TB

產品版本

- 企業版、標準版、工作群組和開發人員版本。如需支援的版本和功能的最新清單，請參閱 AWS 文件中的 Amazon RDS 上的 Microsoft SQL 伺服器。如果您使用的是 AWS DMS，另請參閱使用 Microsoft SQL 伺服器資料庫作為 AWS DMS 目標 (適用於 AWS DMS 支援的 SQL 伺服器版本)。

架構

源, 技術, 堆棧

- 內部部署 Microsoft SQL 伺服器資料庫

目標技術堆疊

- Amazon RDS for SQL Server 資料庫執行個體

來源與目標架構

使用 AWS DMS：

使用原生 SQL 伺服器工具：

工具

- AWS DMS 支援數種類型的來源和目標資料庫。如需詳細資訊，請參閱 AWS DMS 逐步解說。如果 AWS DMS 不支援來源資料庫，請選取其他移轉資料的方法。
- 本機 Microsoft SQL Server 工具包括備份和恢復，複製數據庫嚮導，複製和附加數據庫。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA, 系統管理員

任務	描述	所需技能
識別儲存需求 (儲存類型和容量)。		DBA, 系統管理員
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, 系統管理員
識別來源和目標資料庫的網路存取安全性需求。		DBA, 系統管理員
識別應用程式移轉策略。		DBA, 系統管理員

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		系統管理員
建立安全性群組。		系統管理員
設定並啟動 Amazon RDS 資料庫執行個體。		DBA, 系統管理員

移轉資料-選項 1

任務	描述	所需技能
使用原生 SQL Server 工具或協力廠商工具來移轉資料庫物件和資料。		DBA

移轉資料-選項 2

任務	描述	所需技能
使用 AWS DMS 遷移資料。		DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA、應用程式擁有者、系統管理員

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, 系統管理員
審核並驗證專案文件。		DBA、應用程式擁有者、系統管理員
收集指標，例如移轉時間、手動與自動化工作的百分比，以及節省成本。		DBA、應用程式擁有者、系統管理員
關閉專案並提供意見反應。		DBA、應用程式擁有者、系統管理員

相關資源

參考

- 在 Amazon Web Services 上部署 Microsoft SQL 服務器
- AWS 管理系統網站
- Amazon RDS 定價
- AWS 上的 Microsoft 產品
- AWS 上的 Microsoft 授權
- AWS 上的 Microsoft SQL 服務器
- 搭配 Microsoft SQL 伺服器資料庫執行個體使用視窗驗證
- Amazon RDS 異地備份部署

教學課程和影片

- 開始使用 AWS DMS
- Amazon RDS 入門
- AWS DMS (影片)
- Amazon RDS（視頻）

使用複製將資料從 Microsoft Azure Blob 遷移到 Amazon S3

由蘇哈斯巴薩瓦拉（AWS），艾丹·基恩（AWS）和寇里巷（AWS）創建

環境：PoC 或試點	來源：Microsoft Azure 存儲容器	目標：Amazon S3 桶
R 類型：重新平台	工作量：Microsoft	技術：移轉、儲存與備份
AWS 服務：Amazon S3		

Summary

此模式說明如何使用 Rclone 將資料從 Microsoft Azure Blob 物件儲存體遷移到亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。您可以使用此模式來執行一次性移轉或持續同步處理資料。Rclone 是用 Go 編寫的命令行程序，用於跨雲提供商的各種存儲技術移動數據。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 儲存在 Azure Blob 容器服務中的資料

架構

源, 技術, 堆棧

- 蔚藍的 Blob 存儲容器

目標技術堆疊

- Amazon S3 儲存貯體
- Amazon Elastic Compute Cloud (Amazon EC2) Linux 實例

架構

工具

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Rclone 是一個開源的命令行程序，靈感來自 rsync。它用於跨許多雲存儲平台管理文件。

最佳實務

將資料從 Azure 移轉到 Amazon S3 時，請注意以下考量事項，以避免不必要的成本或降低傳輸速度：

- 在與 Azure 儲存體帳戶和 Blob 容器相同的地理區域中建立 AWS 基礎設施，例如 AWS 區域 `us-east-1` (維吉尼亞北部) 和 Azure 區域。`East US`
- 如果可能，請避免使用 NAT 閘道，因為它會產生輸入和輸出頻寬的資料傳輸費用。
- 使用適用於 Amazon S3 的 VPC 人雲端閘道端點來提高效能。
- 考慮使用以 AWS 重力 2 (ARM) 處理器為基礎的 EC2 執行個體，以獲得比 Intel x86 執行個體更低的成本和更高的效能。Rclone 是大量交叉編譯的，並提供了一個預編譯的 ARM 二進製文件。

史诗

準備 AWS 和 Azure 雲端資源

任務	描述	所需技能
準備目的地 S3 儲存貯體。	在適當的 AWS 區域建立新的 S3 儲存貯體，或選擇現有儲存貯體做為要遷移之資料的目的地。	AWS 管理員
為 Amazon EC2 建立 IAM 執行個體角色。	為 Amazon EC2 建立新的 AWS Identity and Access Management (IAM) 角色。此角色可讓您的 EC2 執行個體寫入目的地 S3 儲存貯體的存取權。	AWS 管理員

任務	描述	所需技能
將政策附加到 IAM 執行個體角色。	使用 IAM 主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 為 EC2 執行個體角色建立內嵌政策，以允許對目的地 S3 儲存貯體寫入存取許可。如需範例原則，請參閱其他資訊一節。	AWS 管理員
啟動 EC2 執行個體。	啟動設定為使用新建立的 IAM 服務角色的 Amazon Linux 2 EC2 執行個體。此執行個體也需要透過網際網路存取 Azure 公用 API 端點。 注意：請考慮使用以 AWS 引力為基礎的 EC2 執行個體來降低成本。Rclone 提供 ARM 編譯的二進製文件。	AWS 管理員
建立 Azure 的 AD 服務主體。	使用 Azure CLI 建立具有來源 Azure Blob 儲存體容器唯讀存取權的 Azure 作用中目錄 (Azure AD) 服務主體。如需指示，請參閱其他資訊一節。將這些登入資料存放在 EC2 執行個體上`~/azure-principal.json` 。	雲端系統管理員，Azure

安裝和配置複製

任務	描述	所需技能
下載並安裝複製。	下載並安裝 Rclone 命令列程式。如需安裝指示，請參閱 Rclone 安裝說明文件。	一般 AWS，雲端管理員
設定複製。	複製下列`rclone.conf` 範例檔案。以您`AZStorage Account` 的 Azure 儲存體帳戶名稱`us-east-1` 和 S3 儲存貯體所在的 AWS 區域取代。將此檔案儲存到 EC2 執行個體`~/.config/rclone/rclone.conf` 上的位置。 `[AZStorageAccount]` `type = azureblob` `account = AZStorage Account` `service_principal_file = azure-principal.json` `[s3]` `type = s3` `provider = AWS` `env_auth = true` `region = us-east-1`	一般 AWS，雲端管理員
驗證複製組態。	若要確認 Rclone 已設定且權限運作正常，請確認 Rclone 可以剖析您的組態檔，而且 Azure Blob 容器和 S3 儲存貯體內的物件是否可存取。如需驗證命令範例，請參閱下列內容。	一般 AWS，雲端管理員

任務	描述	所需技能
	• 在配置文件中列出配置的遙控器。這將確保您的配置文件被正確解析。檢閱輸出以確定其與您的`rclone.conf` 檔案相符。 `rclone listremotes` `AZStorageAccount:` `s3:` • 列出已設定帳戶中的 Azure Blob 容器。以您在`rclone.conf` 檔案中使用的儲存體帳戶名稱取`AZStorageAccount` 代。 `rclone lsd AZStorage` `Account:` `2020-04-29 08:29:26` `docs` • 列出 Azure Blob 容器中的檔案。以 Azure 儲存體帳戶中的實際 Blob 容器名稱取代此命令中的文件。 `rclone ls AZStorage` `Account:docs` `824884 administr` `ator-en.a4.pdf` • 列出 AWS 帳戶中的儲存貯體。 `[root@ip-10-0-20-157` `~]# rclone lsd s3:`	

任務	描述	所需技能
	2022-03-07 01:44:40 examplebucket-01 2022-03-07 01:45:16 examplebucket-02 2022-03-07 02:12:07 examplebucket-03 • 列出 S3 儲存貯體中的檔案。 [root@ip-10-0-20-157 ~]# rclone ls s3:examplebucket-01 template0.yaml template1.yaml	

使用複製移轉資料

任務	描述	所需技能
從您的容器遷移資料。	執行複製複製或同步命令。 範例：複製 此命令會將資料從來源 Azure Blob 容器複製到目的地 S3 儲存貯體。 rclone copy AZStorageAccount:blob-container s3:examplebucket-01 範例：同步	一般 AWS，雲端管理員

任務	描述	所需技能
	此命令會同步處理來源 Azure Blob 容器和目的地 S3 儲存貯體之間的資料。 `rclone sync AZStorageAccount:blob-container s3:examplebucket-01` 重要事項：使用 sync 命令時，來源容器中不存在的資料將從目的地 S3 儲存貯體中刪除。	
同步您的容器。	初始複製完成後，針對進行中的移轉執行 Rclone sync 命令，以便只複製目標 S3 儲存貯體遺失的新檔案。	一般 AWS，雲端管理員
確認資料已成功移轉。	若要檢查資料是否已成功複製到目的地 S3 儲存貯體，請執行 Rclone lsd 和 ls 命令。	一般 AWS，雲端管理員

相關資源

- Amazon S3 使用者指南 (AWS 文件)
- 適用於 Amazon EC2 的 IAM 角色 (AWS 文件)
- 創建一個 Microsoft 的天藍色斑點容器（Microsoft Azure 文檔）
- 複製命令 (複製文件)

其他資訊

EC2 執行個體的範例角色政策

此政策為您的 EC2 執行個體提供對帳戶中特定儲存貯體的讀寫存取權。如果您的儲存貯體使用客戶受管金鑰進行伺服器端加密，則該政策可能需要額外存取 AWS Key Management Service (AWS KMS)。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
              "arn:aws:s3:::BUCKET_NAME/*",
              "arn:aws:s3:::BUCKET_NAME"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

建立唯讀 Azure AD 服務主體

Azure 服務主體是客戶應用程式、服務和自動化工具用來存取特定 Azure 資源的安全性識別。將其視為具有特定角色和嚴格控制權限的用戶身份（登錄名和密碼或證書）來訪問您的資源。若要建立唯讀服務主體以遵循最低權限權限，並保護 Azure 中的資料免於意外刪除，請依照下列步驟執行：

1. 登入您的 Microsoft Azure 雲端帳戶入口網站，然後在中啟動雲殼層， PowerShell 或使用工作站上的 Azure 命令列介面 (CLI)。
2. 建立服務主體，並將其設定為 Azure Blob 儲存體帳戶的唯讀存取權。將此命令的 JSON 輸出保存到名為的本地文件中azure-principal.json。該文件將上傳到您的 EC2 實例。將顯示在大括號 ({和}) 中的預留位置變數取代為您的 Azure 訂用帳戶識別碼、資源群組名稱和儲存體帳戶名稱。

```
az ad sp create-for-rbac `
--name AWS-Rclone-Reader `
--role "Storage Blob Data Reader" `
```

```
--scopes /subscriptions/{Subscription ID}/resourceGroups/{Resource Group Name}/
providers/Microsoft.Storage/storageAccounts/{Storage Account Name}
```

在 AWS 上從 Couchbase 伺服器遷移到卡佩拉

創建者：巴圖爾加·普雷夫拉查 (AWS)、馬克賭博和索拉伯尚巴格 (AWS)

環境：生產	資料來源：轉換基礎伺服器	目標：卡佩拉
R 類型：重新平台	工作負載：所有其他工作	技術：移轉、分析、資料庫

Summary

Couchbase 卡佩拉是全受管的 NoSQL 資料庫即服務 (DBaaS)，適用於關鍵任務應用程式 (例如，使用者設定檔或線上目錄和庫存管理)。Couchbase 卡佩拉在一個由 CouchBase 管理的 Amazon Web Services (AWS) 帳戶中管理您的 DBaaS 工作負載。Capella 可讓您在單一介面中輕鬆執行和管理多叢集、多重 AWS 區域、多雲端和混合雲複寫。

Couchbase 卡佩拉幫助您立即擴展 Couchbase 服務器應用程序，幫助您在幾分鐘內創建多節點集群。Couchbase 卡佩樂支持所有 Couchbase 服務器功能，包括 SQL++，全文搜索，事件服務和分析服務。它也不需要管理安裝、升級、備份和一般資料庫維護。

此模式說明將自我管理的 Couchbase 伺服器環境遷移到 AWS 雲端的步驟和最佳實務。該模式提供了一個可重複的過程，用於從 Couchbase 服務器集群遷移數據和索引，無論是在內部部署或在雲中運行，到 Couchbase 卡佩拉。使用這些步驟可協助您避免移轉期間發生問題，並加速整體移轉程序。

此模式提供下列兩個移轉選項：

- 如果您要移轉的索引少於 50 個，則選項 1 適用。
- 如果您要移轉 50 個以上的索引，則選項 2 適用。

您還可以在自我管理的 Couchbase 服務器上設置示例數據，以便跟隨遷移指南進行操作。

如果您選擇遷移選項 2，或者如果您使用的範圍或集合不是預設值，則必須使用範例設定檔 (位於「其他資訊」區段中)。

先決條件和限制

先決條件

- 現有的卡佩樂付費帳戶。您也可以在 AWS 上建立 Couchbase 卡佩拉帳戶並使用 Couchbase 卡佩拉免費試用版，然後升級到付費帳戶以設定叢集以進行遷移。要從試用版開始，請按照 Couchbase 卡佩拉入門中的說明進行操作。
- 現有的自我管理 Couchbase 伺服器環境，可在內部部署或部署在雲端服務供應商上。
- 對於移轉選項 2，轉接器基底殼層和組態檔案。若要建立組態檔案，您可以使用 [其他資訊] 區段中的範例檔案。
- 熟悉管理沙發基地服務器和 Couchbase 卡佩拉。
- 熟悉在命令行界面（CLI）中打開 TCP 端口和運行命令。

移轉程序也需要下表所述的角色和專業知識。

Role	专业	責任
沙發庫管理員	• 熟悉沙發基地服務器和卡佩拉 • 基本的命令行知識很有幫助，但不是必需的	• Couchbase 伺服器與卡佩爾拉特定工作
系統管理員，IT 管理員	• 熟悉自我管理的 Couchbase 伺服器系統環境與管理	• 在自我管理的 Couchbase 伺服器叢集節點上開啟連接埠並確定 IP 位址

限制

- 這種模式用於將數據，索引和 Couchbase 全文搜索索引從 Couchbase 服務器遷移到 AWS 上的 Couchbase 卡佩拉。該模式不適用於遷移 Couchbase 事件服務或 Couchbase 分析。
- Couchbase 卡佩樂可在多個 AWS 區域使用。有 up-to-date 關卡佩拉支持的區域的信息，請參閱 Couchbase 文檔中的 Amazon Web Services。

產品版本

- Couchbase 服務器（社區或企業）版本 5.x 或更高版本

架構

源, 技術, 堆棧

- 轉換基礎伺服器

目標技術堆疊

- 卡佩拉

目標架構

1. 您可以使用卡佩拉控制平面訪問 Couchbase 卡佩拉。您可以使用卡佩拉控制平面來執行以下操作：
   - 控制和監控您的帳戶。
   - 管理叢集和資料、索引、使用者和群組、存取權限、監控和事件。
2. 已建立叢集。
3. 嘉佩樂資料平面位於 CouchBase 管理的 AWS 帳戶中。建立新叢集之後，Couchbase 卡佩拉會在所選 AWS 區域的多個可用區域中部署該叢集。
4. 您可以在 AWS 帳戶的 VPC 中開發和部署 Couchbase 應用程式。一般而言，此 VPC 會透過 V PC 對等存取卡佩拉資料平面。

工具

- Couchbase 跨資料中心複製 (XDCR) 有助於在不同雲端供應商和不同資料中心的叢集之間複製資料。它是用來從自我管理的 Couchbase 服務器集群數據遷移到 Couchbase 卡佩拉。

  注意：XDCR 不能與 Couchbase 服務器社區版一起使用以遷移到 Couchbase 卡佩拉。相反，您可以使用 cbexport。如需詳細資訊，請參閱從社群版遷移資料史詩。
- C@@ ouchbase 外殼是 Couchbase 服務器和 Couchbase 卡佩拉訪問本地和遠程 Couchbase 集群的命令行外殼。在這種模式中，Couchbase 外殼用於遷移索引。
- cbexport 是一個用於從 Couchbase 集群導出數據的 Couchbase 實用程序。包含在伺服器 CLI 工具中。

史诗

準備移轉

任務	描述	所需技能
評估自我管理的 Couchbase 伺服器叢集的大小。	登入 Couchbase 伺服器的 Couchbase 網頁主控台，並評估您自我管理叢集的節點和儲存貯體。 1. 若要顯示叢集節點清單，請選擇導覽列中的 [伺服器] 索引標籤。 2. 記錄節點數目，然後選擇清單上的每個節點以顯示其屬性。 3. 記錄每個節點的記憶體和儲存空間。 4. 選擇導覽列中的「值區」標籤，然後選擇清單中的每個值區以顯示其屬性。記錄每個值區的 RAM 配額和衝突解決方案設定。 您將使用自我管理的 Couchbase 服務器集群配置作為調整大小和配置 Couchbase 卡佩拉目標集群的一般指南。 有關更詳細的 Couchbase 卡佩拉尺寸練習的幫助，請聯繫 Couchbase。	沙發庫管理員
在自我管理的 Couchbase 伺服器叢集上記錄 Couchbase 服務分佈。	1. 在 Couchbase Web 主控台上，選擇「伺服器」索引標籤以顯示叢集節點清單。	沙發庫管理員

任務	描述	所需技能
	2. 選擇每個節點以顯示其屬性，然後記錄每個節點（數據服務，查詢服務，索引服務，搜索服務，分析服務和事件服務）的 Couchbase 服務分佈。	
記錄自我管理的 Couchbase 伺服器叢集節點的 IP 位址。	如果您使用的是社群版，請忽略此步驟。）記錄叢集中每個節點的 IP 位址。稍後它們將被添加到您的 Couchbase 卡佩拉集群中的允許列表中。	轉換器管理員，系統管理員

在卡佩拉上部署和配置資源

任務	描述	所需技能
選擇一個範本。	1. 登入您的 Couchbase Capella 控制平面，在主導覽中選擇 [儀表板] 索引標籤或 [叢集] 索引標籤，然後選擇 [建立叢集]。 2. 使用您評估自我管理的 Couchbase Server 叢集所記錄的資訊，選擇符合組態需求的叢集範本。如果找不到適當的範本，請在 [叢集大小編輯器] 中選擇 [自訂範本]。	沙發庫管理員
選擇並設定節點。	選擇並設定節點以符合您自我管理的 Couchbase Server 叢集環境，包括節點數量、服務分配、計算或 RAM 以及儲存體。	沙發庫管理員

任務	描述	所需技能
	Couchbase 卡佩拉使用多維縮放的最佳實踐。只能根據部署最佳實務來選擇服務和節點。這可能意味著您無法完全匹配自我管理的 Couchbase 服務器集群的配置。	
部署叢集。	選擇支援區域和支援套件，然後部署叢集。如需詳細步驟和指示，請參閱 Couchbase 文件中的建立叢集。 重要提示：如果您使用的是 Couchbase 卡佩拉免費試用版，則必須在開始遷移之前將其轉換為付費帳戶。要轉換您的帳戶，請打開 Couchbase 卡佩拉控制平面的「計費」部分，然後選擇「添加激活 ID」。在您與 Couchbase 銷售完成購買協議之後，或在您透過 AWS Marketplace 購買後，系統會將啟用 ID 傳送至您的帳單聯絡電子郵件地址。	沙發庫管理員

任務	描述	所需技能
建立資料庫認證使用者。	資料庫認證使用者是叢集特有的，由使用者名稱、密碼和一組值區權限組成。建立值區和存取值區資料時，需要此使用者。 在 Couchbase 卡佩拉控制平面中，按照 Couchbase 卡佩拉文檔中配置數據庫憑據中的說明創建新集群的數據庫憑據。 注意：如果組織使用者想要遠端或透過 Couchbase Capella UI 存取特定叢集上的值區資料，則組織使用者需要指派給他們的組織角色認證。這與數據庫憑據分開，這些憑據通常由應用程序和集成使用。創建組織用戶允許您創建和管理 Couchbase 卡佩拉集群上的目標存儲桶。	沙發庫管理員
如果使用移轉選項 2，請安裝轉換器殼層。	您可以在任何具有網絡訪問自我管理的 Couchbase 服務器和 Couchbase 卡佩拉集群的系統上安裝 Couchbase 外殼。如需詳細資訊，請參閱 Couchbase 殼層文件中的安裝核心殼層版本 1.0.0-beta .5。 透過在命令列終端機中測試與自我管理叢集的連線，確認已安裝 Couchbase Shell。	轉換器管理員，系統管理員

任務	描述	所需技能
允許 IP 位址。	1. 在 Couchbase 卡佩拉控制平面中，選擇叢集，然後選擇您的目標叢集。 2. 選擇叢集的 [連線] 索引標籤，並記錄叢集的 [管理允許的 IP] 下所列的連線端點。 3. 若要將安裝 Couchbase 殼層的系統 IP 位址以及自我管理的 Couchbase 伺服器叢集執行個體的 IP 位址新增為允許的 IP 位址，請執行下列動作： a. 在 [廣域網路] 下方，選擇 [管理允許的 IP]。 b. 選擇 [新增允許的 IP]，輸入您安裝 Couchbase 命令介面的系統的 IP 位址，然後選擇 [新增 IP]。 c. 重複上一步，以新增自我管理的 Couchbase 伺服器叢集執行個體的 IP 位址。 如需有關允許的 IP 位址的詳細資訊，請參閱 Couchbase 文件中的設定允許的 IP 位址。	轉換器管理員，系統管理員

任務	描述	所需技能
設定憑證。	1. 若要下載叢集的根憑證，請在「根憑證」下選擇「下載」。 2. 使用 .pem 副檔名將根憑證儲存在系統上執行 Couchbase 命令介面的資料夾中。 3. 接下來，登入您的自我管理 Couchbase 伺服器 Web 主控台，選擇左側導覽列中的 [安全性]，然後選擇 [憑證] 索引標籤。 4. 複製自我管理的 Couchbase 伺服器叢集的根憑證，並將其儲存為 .pem 檔案到您儲存 Couchbase Capella 叢集的根憑證檔案的相同資料夾中。如需有關根憑證的詳細資訊，請參閱 Couchbase 伺服器文件中的根憑證。	轉換器管理員，系統管理員

任務	描述	所需技能
建立用於轉接器基底殼的組態檔案。	在 Couchbase 命令介面安裝的主目錄中建立一個設定點檔案 (例如)。`/<HOME_DIRECTORY>/.cbsh/config` 有關更多信息，請參閱 Couchbase 文檔中的 Config 點文件。 將來源和目標叢集的連線特性新增至組態檔。您可以使用 [其他資訊] 區段中的範例組態檔案，並編輯叢集的設定。 將具有更新設定的組態檔案儲存到`.cbsh`資料夾中 (例如，`/<HOME_DIRECTORY>/.cbsh/config`)。	轉換器管理員，系統管理員
建立目標值區。	對於每個源存儲桶，請按照 Couchbase 文檔中的創建存儲桶中的說明在 Couchbase 集群中創建一個目標存儲桶。 您的目標存儲桶配置必須與自我管理的 Couchbase 服務器集群中存儲桶的存儲桶名稱，內存設置和衝突解決設置匹配。	沙發庫管理員

任務	描述	所需技能
建立範圍和集合。	每個存儲桶都包含一個默認範圍和帶有密鑰空間_default._default 的集合。 如果您在範圍和集合中使用任何其他密鑰空間，則必須在目標 Capella 集群中創建相同的密鑰空間。 1. 在您安裝了 Couchbase 外殼的系統上打開命令行終端。 2. 若要啟動轉接器基底殼，請執行下列命令。 `./cbsh` 3. 針對您要移轉的每個值區，請執行下列命令，在 Capella 叢集中建立範圍和集合。請務必以您<BUCKET_NAME>要移轉的值區名稱取代。 `scopes --clusters "On-Prem-Cluster" --bucket <BUCKET_NAME> | select scope | where scope != "_default" | each { |it| scopes create $it.scope --clusters "Capella-Cluster" } collections --clusters "On-Prem-Cluster" --bucket <BUCKET_NAME> | select scope collection | where $it.scope != "_default" | where $it.colle`	沙發庫管理員

任務	描述	所需技能
	`ction != "_default" | each { |it| collectio ns create $it.colle ction --clusters "Capella-Cluster" -- bucket <BUCKET_NAME> -- scope $it.scope }`	

從企業版遷移數據

任務	描述	所需技能
在自我管理的 Couchbase 伺服器叢集節點上開啟 TCP 連接埠。	請確定已開啟適當的連接埠，以便在自我管理的 Couchbase 伺服器叢集的節點上進行 XDCR 通訊。如需詳細資訊，請參閱 Couchbase 伺服器連接埠說明文件。	轉換器管理員，系統管理員
如果您正在使用 Couchbase 服務器企業版，請設置 Couchbase XDCR。	1. 在 Couchbase 卡佩拉控制平面主導覽中，選擇 [叢集]，然後選擇要移轉的目標叢集。 2. 在「根憑證」下，選擇「複製」。 3. 登入您自我管理的 Couchbase 伺服器網頁主控台，然後在主導覽中選擇 XDCR。然後選擇添加遙控器。 4. 輸入以下設定： • 叢集名稱 — 卡佩拉叢集連線的名稱	沙發庫管理員

任務	描述	所需技能
	• IP/主機名 — 您的 Couchbase 卡佩拉群集的連接端點 • 遠端叢集的使用者名稱 — 您 Couchbase 卡佩拉叢集的資料庫使用者 • 密碼 — 您的 Couchbase 卡佩拉叢集的資料庫使用者密碼 • 啟用安全連線 — 已選取 • 完整 (TLS 加密密碼和資料) — 已選取 5. 貼上您先前複製的 Capella 叢集根憑證，然後選擇 [儲存]。	
啟動沙發基座 XDCR。	1. 在您自我管理的 Couchbase 伺服器 Web 主控台中，在主導覽中選擇 [XDCR]，然後選擇 [新增複寫]。 2. 輸入以下設定： • 從值區複製 — 選取要移轉的來源值區。 • 遠端儲存貯體 — 輸入目標值區名稱。 • 遠端叢集 — 選取您先前建立的目標叢集。 3. 選擇儲存複製。複寫程序應該會在幾秒鐘內開始。	沙發庫管理員

使用選項 1 遷移索引

任務	描述	所需技能
將自我管理叢集索引遷移到 Couchbase 卡佩拉。	重要事項：如果您要移轉的索引少於 50 個，我們建議您執行此程序。如果要移轉的索引超過 50 個，建議您使用移轉選項 2。 1. 在「轉換庫」Web 主控台上，選擇「索引」。 2. 在索引清單中，選擇您要移轉的第一個索引。然後會顯示索引定義。 3. 使用CREATE陳述式複製索引定義，但不要複製`WITH { "defer_build":true }` 。 例如，從下列範例索引定義中，您只能複製`CREATE INDEX `cityindex` ON `travel-sample`(`city`)` 。 `CREATE INDEX `cityindex` ON `travel-sample`(`city`) WITH { "defer_build":true }` 4. 在 Couchbase 卡佩拉控制平面中，選擇叢集，然後選擇目標叢集。 5. 在「工具」下拉式清單中，選擇 「查詢維護作業 將您先前複製的CREATE陳述式	轉換器管理員，系統管理員

任務	描述	所需技能
	貼到 [查詢編輯器] 中，然後選擇 [執行]。這將創建並構建索引。 6. 若要確認已建立索引，請從 [工具] 下拉式清單中選擇 [索引]。該列表顯示索引已創建並構建。 7. 針對必須移轉的每個索引重複此程序。	

使用選項 2 遷移索引

任務	描述	所需技能
移轉索引定義。	重要事項：如果您要移轉 50 個以上的索引，我們建議您執行此程序。如果要移轉的索引少於 50 個，建議您使用移轉選項 1。 1. 在您安裝了 Couchbase 外殼的系統上打開命令行終端。 2. 若要啟動轉接器基底殼，請執行下列命令。 `./cbsh` 3. 若要連線到自我管理的 Couchbase 伺服器叢集，請執行下列命令。 `cb-env cluster On-Prem-Cluster`	轉換器管理員，系統管理員

任務	描述	所需技能
	4. 若要將索引定義從自我管理的 Couchbase 伺服器叢集遷移到 Couchbase 卡佩拉叢集，請針對您要移轉的每個值區執行下列命令。請務必使用<BUCKET_NAME> 與您要移轉之索引對應的值區名稱來取代。此遷移選項要求您的目標值區名稱與來源值區名稱相同。 `query indexes --definitions | where bucket =~ <BUCKET_NAME>| get definition | each { |it| query $it --clusters Capella-Cluster }`	

任務	描述	所需技能
建立索引定義。	1. 若要將內容切換至 Couchbase 卡佩拉叢集，請執行下列命令： `cb-env cluster Capella-Cluster` 2. 若要建立移轉至 Couchbase Capella 叢集的索引定義，請執行下列指令，並以 <BUCKET_NAME> 與您要建立之索引對應的值區名稱取代。 `query 'SELECT RAW CONCAT("BUILD INDEX ON ", k , "(['", CONCAT2 ("','", inames), "']);") FROM system:indexes AS s LET bid = CONCAT("` ",s.bucket_id, "`"), sid = CONCAT("`", s.scope_id, "`"), kid = CONCAT("` ", s.keyspace_id, "`"), k = NVL2(bid, CONCAT2(".", bid, sid, kid), kid) WHERE s.namespa ce_id = "default" AND s.bucket_id = "" GROUP BY k LETTING inames = ARRAY_AGG (s.name) FILTER (WHERE s.state = 'deferred') HAVING ARRAY_LENGTH(iname`	轉換器管理員，系統管理員

任務	描述	所需技能
	s) > 0;' | each { | it| query $it } 3. 對每個桶重複此步驟。	

移轉全文檢索搜尋索引

任務	描述	所需技能
將自我管理的叢集全文檢索搜尋索引遷移到 Couchbase 卡佩拉。	1. 在轉換器網頁主控台中，選擇「搜尋」。 2. 在全文檢索搜尋 (FTS) 索引清單中，選擇要移轉的第一個 FTS 索引，然後選擇 [顯示索引定義 JSON]，然後選擇 [複製到剪貼簿]。 記下索引名稱及其所屬值區。 3. 在 Couchbase 卡佩拉控制平面中，選擇叢集，然後選擇目標叢集。 4. 在 [工具] 下拉式清單中，選擇 [全文搜尋]。 5. 選擇「匯入索引」，然後貼上 FTS 索引定義。 6. 輸入索引名稱，選取正確的值區 (如自我管理叢集上所述)，然後選擇 [建立]。 7. 針對必須移轉的每個 FTS 索引重複此程序。	沙發庫管理員

從社區版遷移數據

任務	描述	所需技能
從自我管理的 Couchbase 服務器社區版導出數據。	加密的 XDCR 不適用於 Couchbase 社區版。您可以從 Couchbase 社區版導出數據，然後手動將數據導入到 Couchbase 卡佩拉。 若要從來源值區匯出資料，請cbexport在命令列中使用。 以下命令作為示例提供。 `cbexport json \` `--cluster localhost \` `--bucket <SOURCE BUCKET NAME> \` `--format lines \` `--username <USERNAME> \` `--password <PASSWORD> \` `--include-key cbkey \` `--scope-field cbscope \` `--collection-field cbcoll \` `--output cbexported_data.json` 請注意cbkey，cbscope、cbcoll、和cbexported_data.json 是任意標籤。它們將在稍後的過程中被引用，因此，如果您選擇以不同的方式命名它們，請記下它們。	沙發庫管理員

任務	描述	所需技能
將數據導入卡佩拉。	1. 在 Couchbase 卡佩拉控制平面中，選擇叢集，然後選擇目標叢集。 2. 在 [工具] 下拉式清單中，選擇 [匯入]。這將打開一個嚮導，其中包含以下六個步驟： a. 值區 — 選擇目標值區。 b. 檔案 — 選擇 [JSON]，選擇 [行]，然後選擇 [使用網頁瀏覽器]。如果您有大量數據，則可以瀏覽「手動」選項。選取建立者的檔案`cbexport`。 c. 集合 — 選擇 [自訂集合對應]。 如果您的社群版資料庫不使用範圍或集合，或僅使用 _default，您可以改為選擇「選取單一集合」選項。 對於「集合對映表示式」，輸入`%cbscope% .%cbcoll%`。若要驗證此表示式是否正常運作，您可以貼上範例資料，如下所示。 `{ "cbscope" :"inventory", "cbcoll":"landmark", "cbkey":" landmark_3991" }`	沙發庫管理員

任務	描述	所需技能
	d. 密鑰-選擇客戶代。（如果您不關心保留正在導入的數據的密鑰，則可以選擇「自動生成的 UUID」，然後繼續進行步驟 5。）輸入 %cb key % 做為「關鍵名稱產生器表示式」。若要驗證此運算式是否正常運作，請貼上一些範例資料。 e. 組態 — 選擇忽略欄位，然後輸入 CBscope、CBkey。這些欄位包含在匯入後不需要位於目標值區中的暫時性資訊。將其他設定保留為各自的預設設定。 f. 導入-查看，並在準備好時選擇導入。等待上傳和數據導入。 對於大文件，Couchbase 卡佩拉支持使用 cURL 命令行導入。您可以在 Couchbase 卡佩拉文檔中的導入數據中更詳細地了解導入選項。	

測試並驗證移轉

任務	描述	所需技能
驗證資料移轉。	1. 在 Couchbase 卡佩拉控制平面中，選擇叢集，然後在叢集清單中選擇目標叢集。 2. 選擇目標叢集的「值區」索引標籤。確認目標值區中的項目數目 (文件) 與來源值區中的項目數量相符。 3. 在目標叢集的 [工具] 下拉式清單中，選擇 [文件]。確認所有文件均已移轉。 4. (選擇性) 移轉所有資料之後，您可以透過刪除複寫來關閉複寫。如需詳細資訊，請參閱 Couchbase 文件中的刪除複寫。	沙發庫管理員
驗證索引移轉。	在 Couchbase 卡佩拉控制面的目標集群的工具下拉列表中，選擇索引。確認已移轉並建立索引。	沙發庫管理員
驗證查詢結果。	1. 在 Couchbase 卡佩拉控制平面的目標叢集的「工具」下拉式清單中，選擇「查詢工作台」。 2. 執行範例 N1QL 查詢或應用程式中使用的查詢。請確定您收到的結果與在自我管理的 Couchbase 伺服器叢集中執行查詢時相同。	沙發庫管理員

任務	描述	所需技能
驗證全文檢索搜尋結果 (適用於移轉 FTS 索引時)。	1. 在 Couchbase 卡佩拉控制面的目標集群的工具下拉列表中，選擇全文搜索。 2. 透過選擇 FTS 索引名稱來選取其名稱。 3. 選擇 Search (搜尋)。 4. 輸入搜尋查詢範例，然後選擇「搜尋」。 5. 確認結果與在自我管理叢集上執行搜尋時的結果相同。	沙發庫管理員

相關資源

準備移轉

- 開始使用卡佩拉免費試用
- 庫奇基地卡佩樂的雲端供應商需求
- 沙巴卡佩拉尺寸指南

遷移數據和索引

- XDCR 核磁碟機基座
- 沙發基殼說明文件

庫奇基地卡佩拉 SLA 和支持

- 卡佩拉服務水平協議 (SLA)
- 卡佩樂服務支持政策

其他資訊

下面的代碼是用於 Couchbase 外殼的示例配置文件。

```
Version = 1

[[clusters]]
identifier = "On-Prem-Cluster"
hostnames = ["<SELF_MANAGED_COUCHBASE_CLUSTER>"]
default-bucket = "travel-sample"
username = "<SELF_MANAGED_ADMIN>"
password = "<SELF_MANAGED_ADMIN_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"

[[clusters]]
identifier = "Capella-Cluster"
hostnames = ["<COUCHBASE_CAPELLA_ENDPOINT>"]
default-bucket = "travel-sample"
username = "<CAPELLA_DATABASE_USER>"
password = "<CAPELLA_DATABASE_USER_PWD>"
tls-cert-path = "/<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>"
data-timeout = "2500ms"
connect-timeout = "7500ms"
query-timeout = "75s"
```

儲存組態檔之前，請使用下表確定您已新增自己的來源和目標叢集資訊。

<SELF_MANAGED_COUCHBASE_CLUSTER>	使用您自我管理的 Couchbase 伺服器叢集的 IP 位址。
<SELF_MANAGED_ADMIN>	使用自我管理的 Couchbase 伺服器叢集的系統管理員使用者。
<ABSOLUTE_PATH_TO_SELF_MANAGED_ROOT_CERT>	為您的自我管理的 Couchbase 服務器群集使用保存的根證書文件的絕對路徑。
<COUCHBASE_CAPELLA_ENDPOINT>	使用您的卡佩拉群集的連接端點。
<CAPELLA_DATABASE_USER>	使用您的 Couchbase 卡佩拉叢集的資料庫使用者。

<CAPELLA_DATABASE_USER_PWD>	使用您的 Couchbase 卡佩拉叢集的資料庫使用者密碼。
<ABSOLUTE_PATH_TO_COUCHBASE_CAPELLA_ROOT_CERT>	為您的 Couchbase 卡佩拉群集使用保存的根證書文件的絕對路徑。

從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的阿帕奇 Tomcat

由尼爾·阿德良（AWS）和阿弗羅茲汗（AWS）創建

環境：生產	資料來源：應用	目標：Amazon EC2 實例上的阿帕奇湯姆貓
R 類型：重新平台	工作負載：IBM；開放原始	技術：遷移；Web 和移動應用程序
AWS 服務：Amazon EC2		

Summary

此模式會引導您完成從現場部署紅帽企業 Linux (RHEL) 6.9 或更新版本執行 IBM WebSphere 應用程式伺服器 (WAS) 的系統遷移到在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上執行 Apache Tomcat 的 RHEL 8 的步驟。

此模式可套用至下列來源版本和目標版本：

- WebSphere 應用程序服務器 7.x 到阿帕奇湯姆貓 8（使用 Java 7 或更高版本）
- WebSphere 應用程序服務器 8.x 到阿帕奇湯姆貓 8（使用 Java 7 或更高版本）
- WebSphere 應用程序服務器 8.5.5.x 到阿帕奇湯姆貓 9（使用 Java 8 或更高版本）
- WebSphere 應用程序服務器 8.5.5.x 到阿帕奇湯姆貓 10（使用 Java 8 或更高版本）

先決條件和限制

前提

- 有效的 AWS 帳戶
- 源代碼 Java 代碼，具有以下假設：
  - 使用 Java 7 或更新版本的 Java 開發套件 (JDK) 版本
  - 使用春天或阿帕奇支柱框架
  - 不使用企業 Java 豆類（EJB）框架或任何其他 Tomcat 無法使用的 WebSphere 服務器功能

  - 主要使用小服務程序或 Java 服務器頁面（JSP）
  - 使用 Java 資料庫連線 (JDBC) 連接器連線至資料庫
- 來源 IBM WebSphere 應用程式伺服器 7.x 版或更新版本
- 目標阿帕奇雄貓 8.5 或更高版本

架構

源, 技術, 堆棧

- 使用 Apache 的 Struts 模型-視圖-控制器（MVC）框架構建的 Web 應用程序
- 在 IBM 應用程式伺服器 7.x 版或 8.x 版上執行的網頁 WebSphere 應用程式
- 使用輕量型目錄存取通訊協定 (LDAP) 連接器連線至 LDAP 目錄 (iPlane /eTrust) 的 Web 應用程式
- 使用 IBM Tivoli 存取管理員 (TAM) 連線能力來更新 TAM 使用者密碼的應用程式 (在目前的實作中，應用程式使用 PD.jar)

本地資料庫

- Oracle Database 21c (21.0.0.0)
- Oracle 資料庫 19c (19.0.0.0)
- Oracle 資料庫 12c 版本 2 (12.2.0.1)
- Oracle 資料庫 12c 版本 1 (12.1.0.2)

目標技術堆疊

- 在 EC2 執行個體上執行 RHEL 上執行的 Apache Tomcat 版本 8 (或更新版本)
- Amazon Relational Database Service（Amazon RDS）甲骨文

如需有關 Amazon RDS 支援的甲骨文版本的詳細資訊，請參閱 Amazon RDS for Oracle 文網站。

目標架構

工具

- 應用程式層：將 Java 應用程式重建為 WAR 檔案。

- 資料庫層：Oracle 原生備份與還原。
- 阿帕奇 Tomcat 遷移工具雅加達 EE. 這個工具需要為 Java EE 8 編寫的 Web 應用程序，該應用程序在 Apache Tomcat 9 上運行並自動將其轉換為在阿帕奇 Tomcat 10 上運行，該應用程序實現雅加達 EE 9。

史诗

規劃移轉

任務	描述	所需技能
完成應用程式探索、目前狀態佔用空間和效能基準。		BA，遷移領導
驗證來源和目標資料庫版本。		DBA
識別目標伺服器 EC2 執行個體的硬體需求。		DBA, SysAdmin
識別儲存需求 (儲存類型和容量)。		DBA, SysAdmin
根據容量、儲存功能和網路功能選擇適當的 EC2 執行個體類型。		DBA, SysAdmin
識別來源和目標資料庫的網路存取安全性需求。		DBA, SysAdmin
識別應用程式遷移策略和工具。		DBA，遷移領導
完成應用程式的移轉設計和移轉指南。		建立領導者，遷移領導者
完成應用程式移轉手冊。		構建鉛，切換引線，測試導線，遷移領導

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		SysAdmin
建立安全性群組。		SysAdmin
設定並啟動 Amazon RDS for Oracle。		DBA, SysAdmin

移轉資料

任務	描述	所需技能
建立或取得端點的存取權，以擷取資料庫備份檔案。		DBA
使用原生資料庫引擎或協力廠商工具來移轉資料庫物件和資料。	如需詳細資訊，請參閱「其他資訊」一節中的「移轉資料庫物件和資料」。	DBA

移轉應用程式

任務	描述	所需技能
提出要移轉的變更請求 (CR)。		切換引線
取得移轉的 CR 核准。		切換引線
依照應用程式移轉手冊的應用程式移轉策略執行。	如需詳細資訊，請參閱其他資訊一節中的「設定應用程式層」。	DBA、移轉工程師、應用程式擁有者
升級應用程式 (如有必要)。		DBA、移轉工程師、應用程式擁有者

任務	描述	所需技能
完成功能性、非功能性、資料驗證、SLA 和效能測試。		測試潛在客戶，應用程序所有者，應用

切過

任務	描述	所需技能
取得應用程式擁有者或企業擁有者的簽署。		切換引線
將應用程式用戶端切換到新的基礎結構。		DBA、移轉工程師、應用程式擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA，移民工程師，SysAdmin
審核並驗證專案文件。		遷移, 領導
收集指標，例如移轉時間、手動與自動化工作的百分比，以及節省成本。		遷移, 領導
關閉專案並提供意見反應。		應用程式擁有者移轉主管

相關資源

參考

- 阿帕奇湯姆貓 10.0 文檔
- 阿帕奇湯姆貓 9.0 文檔

- 阿帕奇湯姆貓 8.0 文檔
- 阿帕奇湯姆貓 8.0 安裝指南
- 阿帕奇雄貓 JNDI 文檔
- Amazon RDS for Oracle 網站
- Amazon RDS 定價
- 甲骨文和 Amazon Web Services
- Amazon RDS 上的甲骨文
- Amazon RDS 異地備份部署

教學課程和影片

- Amazon RDS 入門

其他資訊

移轉資料庫物件和資料

例如，如果您使用的是原生 Oracle 備份/還原公用程式：

1. 為資料庫備份檔案建立 Amazon 簡易儲存服務 (Amazon S3) 備份 (選用)。
2. 將 Oracle 資料庫資料備份至網路共用資料夾。
3. 登入移轉暫存伺服器以對應網路共用資料夾。
4. 將資料從網路共用資料夾複製到 S3 儲存貯體。
5. 為甲骨文請求 Amazon RDS 異地同步備份部署。
6. 將現場部署資料庫備份還原至 Amazon RDS for Oracle

設定應用程式層

1. 從阿帕奇湯姆貓網站安裝湯姆貓 8（或 9 月 10 日）。
2. 將應用 Package 式和共用程式庫封裝成 WAR 檔案。
3. 在 Tomcat 中部署 WAR 檔案。
4. 監視`Linux cat`任何缺少共享庫的開始日誌 WebSphere。
5. 觀看`Linux cat`任何 WebSphere特定部署描述元延伸模組的開始記錄。
6. 從伺服器收集任何遺失的相依 Java 程式 WebSphere 庫。

7. 使用與 TomCAT 相容的對等項目修改 WebSphere特定的部署描述元素。
8. 使用相依 Java 程式庫和更新的部署描述元重建 WAR 檔案。
9. 更新 LDAP 組態、資料庫組態和測試連線 (請參閱 Apache Tomcat 說明文件中的範圍組態操作方法和 JNDI 資料來源的操作方法)。

10對照還原的亞馬遜 RDS 適用於 Oracle 資料庫測試已安裝的應用程式。

11從 EC2 執行個體為 Linux 建立 Amazon 機器映像 (AMI)。

12使用「應 Application Load Balancer」和「自動擴展」群組啟動完成的架構

13更新 URL（使用 WebSeal 結合）以指向應用 Application Load Balancer。

14更新組態管理資料庫 (CMDB)。

使用 Auto Scaling 能從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的 Apache Tomcat

R 類型：重新平台	資料來源：應用	目標：啟用 Auto Scaling 功能的 Amazon EC2 執行個體上的 Apache Tomcat
創建者：AWS	環境：PoC 或試點	技術：Web 和移動應用程序; 遷移
工作負載：開放原始碼；	AWS 服務：Amazon EC2	

Summary

此模式提供指導，讓您在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上將 Java WebSphere 應用程式從 IBM 應用程式伺服器遷移到 Apache Tomcat，且啟用了 Amazon EC2 Auto Scaling 功能。

通過使用這種模式，您可以實現：

- 降低 IBM 授權成本
- 使用異地同步備份部署提供高
- 使用 Amazon EC2 Auto Scaling 改善應用程式彈性

先決條件和限制

先決條件

- Java 應用程式 (版本 7. x 或 8. x) 應該在 LAMP 堆棧中開發。
- 目標狀態是在 Linux 主機上託管 Java 應用程序。這個模式已經在 RHEL 7 環境中成功實作。其他 Linux 發行版本可以遵循這種模式，但應該引用 Apache Tomcat 發行版的配置。
- 您應該了解 Java 應用程序的依賴關係。
- 您必須擁有 Java 應用程式原始程式碼的存取權，才能進行變更。

限制和重新平台變更

- 您應該瞭解企業封存 (EAR) 元件，並確認所有程式庫都封裝在 Web 元件 WAR 檔案中。您需要配置阿帕奇 Maven 的 WAR 插件，並產生 WAR 文件工件。
- 當使用 Apache 的 Tomcat 8，有 servlet-api.jar 和應用程序包內置的 jar 文件之間的一個已知的衝突。若要解決此問題，請從應用程式套件中刪除 servlet-api.jar。
- 您必須配置位於 Apache Tomcat 配置的類路徑中的網絡信息/資源。依預設，JAR 程式庫不會載入目錄中。或者，您可以在 src/main/資源下部署所有資源。
- 檢查 Java 應用程序中是否有任何硬編碼的上下文根目錄，並更新 Apache Tomcat 的新上下文根目錄。
- 要設置 JVM 運行時選項，您可以在 Apache 的湯姆貓箱文件夾中創建配置文件 setenv.sh; 例如，JAVA_OPTS，JAVA_HOME 等。
- 驗證是在容器層級配置的，並在 Apache Tomcat 配置中設置為領域。已針對下列三個領域中的任何一個建立驗證：
  - JDBC 資料庫範圍會在 JDBC 驅動程式存取的關聯式資料庫中查詢使用者。
  - DataSource 「資料庫範圍」會查詢 JNDI 存取之資料庫中的使用者。
  - JNDI 目錄範圍會查詢 JNDI 提供者存取之輕量型目錄存取協定 (LDAP) 目錄中的使用者。 查找需要：
    - LDAP 連線詳細資料：使用者搜尋依據、搜尋篩選器、角色庫、角色篩選
    - 主要 JNDI 目錄範圍：連線至 LDAP、認證使用者，以及擷取使用者為成員的所有群組
- 授權：如果容器具有以角色為基礎的授權來檢查 web.xml 中的授權條件約束，則必須定義 Web 資源，並與條件約束中定義的角色進行比較。如果 LDAP 沒有群組角色對應，您必須<security-role-ref>在 web.xml 中設定屬性，才能達到群組角色對應。若要查看組態文件的範例，請參閱 Oracle 說明文件。
- 資料庫連線：使用 Amazon Relational Database Service 服務 (Amazon RDS) 端點網址和連線詳細資料，在 Apache Tomcat 中建立資源定義。使用 JNDI 查詢來更新應用程式程式碼以參照 a DataSource 。中定義的現有資料庫連線 WebSphere 將無法運作，因為它使用 WebSphere的 JNDI 名稱。您可以<resource-ref>在 web.xml 中新增具有 JNDI 名稱和 DataSource 類型定義的項目。若要查看範例組態文件，請參閱 Apache Tomcat 說明文件。
- 記錄：根據預設，Apache Tomcat 會記錄到主控台或記錄檔。您可以透過更新記錄檔屬性來啟用範圍層級追蹤 (請參閱 Tomcat 中的記錄)。如果您使用的是阿帕奇 Log4j 的日誌追加到一個文件中，你必須下載 tomcat-juli 並將其添加到類路徑。
- 工作階段管理：如果您保留 IBM WebSeal 以進行應用程式負載平衡和工作階段管理，則不需要變更。如果您在 AWS 上使用 Application Load Balancer 或 Network Load Balancer 來取代 IBM WebSeal 元件，則必須使用具有 Memcached 叢集的 Amazon ElastiCache 執行個體來設定工作階段管理，並將 Apache Tomcat 設定為使用開放原始碼工作階段管理。

- 如果您使用的是 IBM WebSeal 正向代理，則必須在 AWS 上設定新的 Network Load Balancer。使用 Network Load Balancer 提供的 IP 進行 WebSeal 結合配置。
- SSL 組態：建議您使用安全通訊端層 (SSL) 進行 end-to-end 通訊。若要在 Apache Tomcat 中設定 SSL 伺服器設定，請依照 Apache Tomcat 文件中的指示進行。

架構

源, 技術, 堆棧

- WebSphere 應用程式伺服器

目標技術堆疊

- 該架構使用 E lastic Load Balancing (版本 2) 。如果您使用 IBM WebSeal 進行識別管理和負載平衡，您可以選取 AWS 上的 Network Load Balancer 來與 IBM WebSeal 反向代理整合。
- Java 應用程式會部署到 Apache Tomcat 應用程式伺服器，該伺服器會在 Amazon EC2 Auto Scaling 群組的 EC2 執行個體上執行。您可以根據 Amazon CloudWatch 指標 (例如 CPU 使用率) 設定擴展政策。
- 如果您要淘汰使用 IBM WebSeal 進行負載平衡，您可以使用 Amazon 的 Memcached ElastiCache 進行工作階段管理。
- 對於後端資料庫，您可以為 Amazon RDS 部署高可用性 (異地同步備份) 並選取資料庫引擎類型。

目標架構

工具

- AWS CloudFormation
- AWS Command Line Interface (AWS CLI)
- 阿帕奇雄貓 (版本 7。 x 或 8. x)
- 瑞爾 7 號或 CENTOS 7
- Amazon RDS 異地備份部署
- Amazon 內 ElastiCache 存緩存 (可選)

史诗

設定 VPC

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		
建立子網路。		
如有必要，請建立路由表。		
建立網路存取控制清單 (ACL)。		
設定 AWS Direct Connect 線或企業 VPN 連線。		

重新平台應用程式

任務	描述	所需技能
重構應用程序構建 Maven 配置以生成 WAR 工件。		
重構 Apache Tomcat 中的應用程式相依性資料來源。		
重構應用程序源代碼以在 Apache Tomcat 中使用 JNDI 名稱。		
部署戰爭成品到阿帕奇湯姆貓。		
完成應用程式驗證和測試。		

設定網路

任務	描述	所需技能
設定企業防火牆以允許連線至相依性服務。		
設定企業防火牆以允許終端使用者存取 AWS 上的 Elastic Load Balancing。		

建立應用程式基礎結

任務	描述	所需技能
在 EC2 執行個體上建立和部署應用程式。		
建立 ElastiCache 適用於 Memcached 的 Amazon 叢集以進行工作階段管理。		
為後端資料庫建立 Amazon RDS 異地同步備份執行個體。		
建立 SSL 憑證並將其匯入 AWS Certificate Manager (ACM)。		
在負載平衡器上安裝 SSL 憑證。		
為阿帕奇 Tomcat 服務器安裝 SSL 證書。		
完成應用程式驗證和測試。		

切過

任務	描述	所需技能
關閉現有的基礎結構。		
將資料庫從生產環境還原到 Amazon RDS。		
通過進行 DNS 更改切斷應用程序。		

相關資源

參考

- 阿帕奇湯姆貓 7.0 文檔
- 阿帕奇湯姆貓 7.0 安裝指南
- 阿帕奇雄貓 JNDI 文檔
- Amazon RDS 異地備份部署
- Amazon ElastiCache

教學課程和影片

- Amazon RDS 入門

將 .NET 應用程式從 Microsoft Azure 應用程式服務遷移到 AWS Elastic Beanstalk

由拉格文德馬達姆希提 (AWS) 創建

環境：PoC 或試點	來源：應用程序	目標：AWS Elastic Beanstalk
R 類型：重新平台	工作量：Microsoft	技術：遷移；Web 和移動應用程序

Summary

此模式描述如何將託管在 Microsoft Azure 應用程序服務上的 .NET Web 應用程序遷移到 AWS Elastic Beanstalk。有兩種方法可以將應用程序遷移到 Elastic Beanstalk：

- 使用適用 AWS Toolkit for Visual Studio-此外掛程式提供了將自訂 .NET 應用程式部署到 AWS 的最簡單、最直接的方法。您可以使用此方法將 .NET 程式碼直接部署到 AWS，並直接從 Visual Studio 建立支援資源，例如 SQL 伺服器資料庫的 Amazon Relational Database Service 服務 (Amazon RDS)。
- 上傳並部署至 Elastic Beanstalk-每個 Azure 應用程式服務都包含名為 Kudu 的背景服務，可用於擷取記憶體傾印和部署記錄、檢視組態參數，以及存取部署套件。您可以使用 Kudu 主控台存取 Azure 應用程式服務內容、擷取部署套件，然後使用 Elastic Beanstalk 主控台中的上傳和部署選項，將套件上傳至 Elastic Beanstalk。

此模式描述了第二種方法（通過 Kudu 將應用程序上傳到 Elastic Beanstalk）。該模式還使用以下 AWS 服務：AWS Elastic Beanstalk，Amazon Virtual Private Cloud（Amazon VPC），Amazon CloudWatch，Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling，Amazon Simple Storage Service (Amazon S3) 和 Amazon Route 53。

.NET Web 應用程式部署到 AWS Elastic Beanstalk，該應用程式在 Amazon EC2 Auto Scaling 群組中執行。您可以根據 Amazon CloudWatch 指標 (例如 CPU 使用率) 設定擴展政策。對於資料庫，您可以在異地同步備份環境或 Amazon DynamoDB 中使用 Amazon RDS，視您的應用程式和業務需求而定。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 Azure 應用程式服務中執行的 .NET Web 應用程式
- 使用 Azure 應用程式服務同時控制台的權限

產品版本

- . NET 核心 (x64) 1.0.1, 2.0.0, 或更高版本, 或. NET 框架 4.x, 3.5 (見. NET 在視窗服務器平台歷史記錄)
- 網際網路資訊服務 (IIS) 8.0 版或更新版本，在 Windows 伺服器 2012 年或更新版本上執行
- 2.0 或 4.0 執行階段。

架構

源, 技術, 堆棧

- 使用 .NET 框架 3.5 或更高版本，或 .NET 核心 1.0.1，2.0.0 或更高版本開發，並託管在 Azure 應用程序服務（Web 應用程序或 API 應用程序）上的應用程序

目標技術堆疊

- AWS Elastic Beanstalk 在 Amazon EC2 Auto Scaling 組中運行

移轉架構

部署工作流

工具

工具

- . NET 核心或. NET 框架
- C#
- IIS
- 酷渡控制台

AWS 服務和功能

- AWS Elastic Beanstalk — Elastic Beanstalk 是一種用於部署和擴展 .NET Web 應用程式的 easy-to-use 服務。Elastic Beanstalk 可自動管理容量佈建、負載平衡和自 auto 調整規模。
- 亞馬遜 EC2 Auto Scaling 組 — Elastic Beanstalk 包括一個 Auto Scaling 組，用於管理環境中的 Amazon EC2 執行個體。在單一執行個體環境中，Auto Scaling 群組可確保隨時都有一個執行個體正在執行。在負載平衡環境中，您可以使用一系列要執行的執行個體設定群組，而 Amazon EC2 Auto Scaling 會視需要根據負載新增或移除執行個體。
- E@@ lastic Load Balancing — 當您在 AWS Elastic Beanstalk 中啟用負載平衡時，它會建立負載平衡器，將流量分配到環境中的 EC2 執行個體之間。
- Amazon CloudWatch — Elastic Beanstalk 會自動使用 Amazon CloudWatch 提供有關您的應用程序和環境資源的信息。Amazon CloudWatch 支援標準指標、自訂指標和警示。
- Amazon Route 53 — Amazon Route 53 是一種高可用性和可擴展的雲域名系統（DNS）網絡服務。您可以使用 Route 53 別名記錄將自訂網域名稱對應至 AWS Elastic Beanstalk 環境。

史诗

設定 VPC

任務	描述	所需技能
設定虛擬私有雲 (VPC)。	在您的 AWS 帳戶中，建立包含必要資訊的 VPC。	系統管理員
建立子網路。	在 VPC 中建立兩個或多個子網路。	系統管理員
建立路由表格。	根據您的需求建立路由表。	系統管理員

設置 Elastic Beanstalk

任務	描述	所需技能
存取 Azure 應用程式服務執行主控台。	瀏覽至應用程式服務儀表板，然後選擇 [進階工具]、[Go]，透過 Azure 入口網站	應用程式開發人員, 系統

任務	描述	所需技能
	存取 Kudu。或者，您可以修改 Azure 應用程式服務 URL，如下所示：`https://<appservicename>.scm.azurewebsites.net` 。	
從 Kudu 下載部署包。	選擇選項以切換作業選DebugConsole項至「視窗 PowerShell 」。這將打開工藤控制台。轉到`wwwroot`文件夾並下載。這將下載 Azure 應用程序服務部署包作為 zip 文件。有關示例，請參閱附件。	應用程式開發人員, 系統
為 Elastic Beanstalk 創建一個包。	解壓縮您從 Azure 應用程式服務下載的部署套件。創建一個名為 JSON 文件`aws-windows-deployment-manifest.json` （此文件僅適用於 .NET 核心應用程序）。建立包含 `aws-windows-deployment-manifest.json` Azure 應用程式服務部署套件檔案的 zip 檔案。有關示例，請參閱附件。	應用程式開發人員, 系統
創建一個新的 Elastic Beanstalk 應用程序。	開啟 Elastic Beanstalk 主控台。選擇現有的應用程式或建立新的應用程式。	應用程式開發人員, 系統

任務	描述	所需技能
建立環境。	在 Elastic Beanstalk 控制台的「操作」菜單中，選擇「創建環境」。選取網頁伺服器環境和 .NET/IIS 平台。對於應用程式代碼，請選擇上傳。上傳您為 Elastic Beanstalk 準備的 zip 文件，然後選擇創建環境。	應用程式開發人員, 系統
配置 Amazon CloudWatch。	依預設，會啟用基本 CloudWatch 監視。如果您想要變更組態，請在 Elastic Beanstalk 精靈中選擇已發佈的應用程式，然後選擇 [監視]。	系統管理員
確認部署套件位於 Amazon S3 中。	建立應用程式環境後，您可以在 S3 儲存貯體中找到部署套件。	應用程式開發人員, 系統
測試應用程式。	建立環境後，請使用 Elastic Beanstalk 主控台中提供的 URL 來測試應用程式。	系統管理員

相關資源

- AWS 彈性豆堆疊概念 (Elastic Beanstalk 文件)
- 在 Elastic Beanstalk 上開始使用 .NET (Elastic Beanstalk 文檔)
- 酷控制台 ( ) GitHub
- 使用「刪除」來管理 Azure Web 應用程式 (GS 實驗室文章)
- 自訂 ASP.NET 核心 Elastic Beanstalk 部署 (適用 AWS Toolkit for Visual Studio 使用者指南)
- Elastic Load Balancing 文件
- AWS Elastic Beanstalk 支援的平台 (Elastic Beanstalk 文件)
- 將 Web 應用程式部署到 AWS (C# 角落文章)
- 擴展 Auto Scaling 群組的大小 (Amazon EC2 文件)

- Amazon RDS 的高可用性 (異地同步備份) (Amazon RDS 文件)

其他資訊

備註

- 如果您要將內部部署或 Azure SQL 伺服器資料庫移轉至 Amazon RDS，您也必須更新資料庫連線詳細資料。
- 為了測試目的，附上示例演示應用程序。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

將自我託管的 MongoDB 環境遷移到 AWS 雲端上的 MongoDB 地圖集

來源:MongoDB	目標：AWS 上的 MongoDB 地圖集	R 類型：重新平台
環境：生產	技術：移轉、分析、資料庫	工作負載：所有其他工作
AWS 服務：Amazon EC2; Amazon VPC		

Summary

此模式說明從自我管理的 MongoDB 環境 (包括 MongoDB 社群伺服器、企業伺服器、企業進階、MLAB 或任何受管 MongoDB 叢集) 遷移到 Amazon Web Services (AWS) 雲端上的 MongoDB 地圖集的步驟。它使用阿特拉斯即時遷移服務來協助加速從 MongoDB 到 MongoDB 地圖集的資料遷移。

該模式隨附於 AWS 規範指導網站上 AWS 雲端上從 MongoDB 遷移到 MongoDB 地圖集的指南。它提供了遷移的實施步驟。

此模式適用於 AWS 服務整合商合作夥伴 (SI 合作夥伴) 和 AWS 使用者。

先決條件和限制

先決條件

- 一個源的 MongoDB 環境遷移到 MongoDB 地圖集

专业

- 這種模式需要熟悉 MongoDB、MongoDB 地圖集和 AWS 服務。如需詳細資訊，請參閱 AWS 規範指導網站上的 AWS 雲端上從 MongoDB 遷移到 MongoDB 地圖集指南中的角色和職責。

產品版本

- 版本 MongoDB 或更新版本

架構

對於支援不同使用情境的 MongoDB 地圖集參考架構，請參閱 AWS 上的 MongoDB 地圖集參考架構，在 AWS AWS Prescriptive Guidance 指南網站上從 AWS 雲端遷移到 AWS 雲端上的 MongoDB 地圖集。

工具

- 阿特拉斯即時遷移服務 — 免費的 MongoDB 公用程式，可協助將資料庫遷移至 Atlas 此服務會使來源資料庫與目的地資料庫保持同步，直到切換為止。當您準備好切斷時，您可以停止應用程式執行個體，將它們指向目的地 Atlas 叢集，然後重新啟動它們。

史诗

探索與評估

任務	描述	所需技能
決定叢集大小。	使用 db.stats () 中的資訊作為總索引空間，估計工作集的大小。假設將經常存取一定百分比的資料空間。或者，您可以根據自己的假設估計記憶體需求。此任務應該需要大約一周的時間。有關此和本史詩中其他故事的更多信息和示例，請參閱「相關資源」部分中的鏈接。	MongoDB DBA，應用程序架構師
估計網路頻寬需求。	若要估算您的網路頻寬需求，請將平均文件大小乘以每秒提供的文件數目。請考慮叢集上任何節點的最大流量作為基礎。若要計算叢集到用戶端應用程式的下游資料傳輸速率，請使用一段時間內傳回的總文件總和。如果您的應用程式從次要節點讀取，請將這個文件	MongoDB

任務	描述	所需技能
	總數除以可用於讀取作業的節點數目。若要尋找資料庫的平均文件大小，請使用 db.stats ()。 avgObjSize 指令。此任務通常需要一天的時間。	
選擇阿特拉斯等級。	請依照 MongoDB 文件中的指示，選取正確的 Atlas 叢集層。	MongoDB
計劃應用程式切換。		MongoDB DBA，應用程序架構師

在 AWS 上設定新的 MongoDB 地圖集環境

任務	描述	所需技能
在 AWS 上建立新的 MongoDB 地圖集叢集。	在 MongoDB 地圖集中，選擇「建立叢集」以顯示「建立新叢集」對話方塊。選取 AWS 做為雲端供應商。	MongoDB
選取區域和全域叢集配置。	從 Atlas 叢集的可用 AWS 區域清單中選取。視需要設定全域叢集。	MongoDB
選取叢集層。	選取您偏好的叢集層。您的階層選擇會決定記憶體、儲存裝置和 IOPS 規格等因素。	MongoDB
設定其他叢集設定。	設定其他叢集設定，例如 MongoDB 版本、備份和加密選項。如需這些選項的詳細資訊，請參閱 < 相關資源 > 一節中的連結。	MongoDB

設定安全性與合規性

任務	描述	所需技能
設定存取清單。	若要連線至 Atlas 叢集，您必須在專案的存取權清單中新增項目。Atlas 使用傳輸層安全性 (TLS)/安全通訊端層 (SSL) 來加密資料庫連線至虛擬私有雲 (VPC) 的連線。要設置項目的訪問列表以及有關此史詩故事的更多信息，請參閱「相關資源」部分中的鏈接。	MongoDB
驗證和授權使用者。	您必須建立並驗證將存取 MongoDB 地圖集叢集的資料庫使用者。若要存取專案中的叢集，使用者必須屬於該專案，而且他們可以屬於多個專案。	MongoDB
建立自訂角色。	(選擇性) 如果內建 Atlas 資料庫使用者權限未涵蓋您想要的權限集，Atlas 支援建立自訂角色。	MongoDB
設定 VPC 對等互連。	(選用) 阿特拉斯支援 VPC 與其他 AWS、Azure 或谷歌雲端平台 (GCP) 虛擬私人雲端平台 (GCP) 虛擬私人雲端互連。	MongoDB
設定 AWS PrivateLink 端點。	(選用) 您可以使用 AWS 在 AWS 上設定私有端點 PrivateLink。	MongoDB
啟用雙因素驗證。	(選用) Atlas 支援雙重認證 (2FA)，協助使用者控制其 Atlas 帳戶的存取權。	MongoDB

任務	描述	所需技能
透過 LDAP 設定使用者驗證和授權。	(選擇性) Atlas 支援透過輕量型目錄存取通訊協定 (LDAP) 執行使用者驗證和授權。	MongoDB
設定統一的 AWS 存取權限。	(選用) 某些 Atlas 功能，包括 Atlas 資料湖和使用客戶金鑰管理進行靜態加密，使用 AWS Identity and Access Management (AWS IAM) 角色進行身份驗證。	MongoDB
使用 AWS KMS 設定靜態加密。	(選用) Atlas 支援使用 AWS 金鑰管理系統 (AWS KMS) 來加密儲存引擎和雲端供應商備份。	MongoDB
設定用戶端欄位層級加密。	(選用) Atlas 支援用戶端欄位層級加密，包括欄位的自動加密。	MongoDB

移轉資料

任務	描述	所需技能
在 MongoDB 地圖集中啟動您的目標副本集。	在 MongoDB 地圖集中啟動您的目標副本集。在 Atlas 即時移轉服務中，選擇「我已準備好遷移」。	MongoDB
將 Atlas 即時遷移服務新增至 AWS 來源叢集的存取清單。	這有助於準備來源環境以連接到目標 Atlas 叢集。	MongoDB
使用 Atlas 即時遷移服務驗證您的 AWS 登入資料。	選擇「開始遷移」。當「準備切換」按鈕變為綠色時，執行	MongoDB

任務	描述	所需技能
	切換。檢閱阿特拉斯叢集效能指標。	

設定作業整合

任務	描述	所需技能
Connect 至 MongoDB 地圖集群集。		应用开发人
與叢集資料互動。		应用开发人
監視叢集。		MongoDB
備份和還原叢集資料。		MongoDB

相關資源

移轉指南

- 在 AWS 雲端上從 MongoDB 移轉至蒙古資料庫地圖集

探索與評估

- 記憶體
- 使用 Atlas 樣本數據集調整大小示
- 行動應用程式的大小範例
- 網路流量
- 叢集自動調整
- 地圖集尺寸模板

設定安全性與合規性

- 設定 IP 存取清單項目

- 配置數據庫用戶
- 使用者存取
- 設定自訂角色
- 數據庫用戶權限
- 設定網路對等連線
- 設定私有端點
- 雙因素身份驗證
- 使用 LDAP 設定使用者驗證和授權
- 阿特拉斯資料湖
- 使用客戶金鑰管理進行靜態加密
- 使用 IAM 角色
- 用戶端欄位層級加密
- 自動用戶端欄位層級加密
- MongoDB 阿特拉斯安全
- MongoDB 中心
- 安全功能和設置

在 AWS 上設定新的 MongoDB 地圖集環境

- 雲端供應商和區域
- 全域叢集
- 叢集層
- 其他叢集設定
- 開始使用阿特拉斯
- 使用者存取
- 叢集

移轉資料

- 監控您的叢集

整合作業

- Connect 至叢集
- 在阿特拉斯執行 CRUD 作業
- 監控您的叢集
- Backup 和還原叢集資料

在 Amazon ECS 上從甲骨文遷移 WebLogic 到阿帕奇湯姆貓（TomEE）

R 類型：重新平台	來源：容器	目標：阿帕奇湯姆貓（托米）在 Amazon ECS
創建者：AWS	環境：PoC 或試點	技術：容器與微服務；遷移
工作量：甲骨文	AWS 服務：Amazon ECS	

Summary

此模式討論了將運行甲骨文的現場部署 Oracle Solaris SPARC 系統遷移 WebLogic 到使用 Amazon 彈性容器服務（亞馬遜 ECS）運行 Apache ToMee（帶有添加容器支持的 Apache Tomcat）的基於碼頭的容器安裝的步驟。

如需移轉與您要從 Oracle 移轉 WebLogic 至 Tomcat 之應用程式相關聯之資料庫的相關資訊，請參閱此目錄中的資料庫移轉模式。

最佳實務

移轉 Java 和 Java 企業版 (Java EE) Web 應用程式的步驟會因應用程式使用的容器特定資源數目而有所不同。基於 Spring 的應用程序通常更容易遷移，因為它們在部署容器上具有少量的依賴關係。相較之下，使用企業 JavaBeans (EJB) 和受管理容器資源 (例如執行緒集區、Java 驗證和授權服務 (JAAS) 和容器管理持續性 (CMP) 的 Java EE 應用程式需要更多的努力。

為「Oracle 應用程式伺服器」開發的應用程式經常使用「Oracle 識別管理」 移轉至開放原始碼應用程式伺服器的客戶經常選擇使用 SAML 型聯盟重新實作身分識別與存取管理。其他人使用 Oracle HTTP 伺服器網路閘門來處理從 Oracle 識別管理套件進行移轉時不是一個選項。

Java 和 Java EE Web 應用程式非常適合在以碼頭為基礎的 AWS 服務上進行部署，例如 AWS Fargate 和 Amazon ECS。客戶經常選擇已預先安裝最新版本的目標應用程式伺服器 (例如 TomEE) 和 Java 開發套件 (JDK) 的 Docker 映像檔。他們將應用程式安裝在基本 Docker 映像之上，將其發佈到 Amazon Elastic Container Registry (Amazon ECR) 登錄中，並將其用於 AWS Fargate 或 Amazon ECS 上的應用程式可擴展部署。

理想情況下，應用程式部署是彈性的；也就是說，應用程式執行個體的數量會根據流量或工作負載擴展或擴展。這表示應用程式執行個體必須上線或終止，才能根據需求調整容量。

將 Java 應用程式移至 AWS 時，請考慮將其設為無狀態。這是 AWS 架構良好架構架構的關鍵架構原則，可使用容器化實現水平擴展。例如，大多數基於 Java 的 Web 應用程序在本地存儲用戶會話信息。為了在 Amazon Elastic Compute Cloud (Amazon EC2) 中自動擴展或出於其他原因導致應用程式執行個體終止，使用者工作階段資訊應儲存在全域，以便 Web 應用程式使用者可以繼續無縫且透明地工作，而無需重新連線或重新登入 Web 應用程式。這種方法有多種架構選項，包括 Amazon ElastiCache for Redis，或在全域資料庫中存放工作階段狀態。應用程式伺服器 (例如 TomEE) 具有外掛程式，可透過 Redis、資料庫和其他全域資料存放區啟用工作階段儲存和管理。

使用可輕鬆與 Amazon 和 AWS X-Ray 整合的通用集中式記錄 CloudWatch 和偵錯工具。移轉提供改善應用程式生命週期功能的機會。例如，您可能想要將建置程序自動化，以便使用持續整合和持續傳遞 (CI/CD) 管線輕鬆進行變更。這可能需要對應用程序進行更改，以便可以在不停機的情況下部署應用程序。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 源代碼 Java 代碼和 JDK
- 以 Oracle 建置的來源應用程式 WebLogic
- 定義的身分識別與存取管理解決方案 (SAML 或 Oracle Webgate)
- 定義應用程式工作階段管理的解決方案 (移動 like-for-like 或使用 Amazon ElastiCache，或視需要使應用程式無狀態)
- 了解團隊是否需要重構特定於 J2EE 的庫以便移植到 Apache TomeE（請參閱 Apache 網站上的 Java EE 7 實現狀態）
- 根據您的安全需求強化 ToMee 映像檔
- 含預先安裝目標 ToMee 的容器影像
- 如果需要，同意並實施應用程序修復（例如，記錄調試構建，身份驗證）

產品版本

- 甲骨文 WebLogic OC4J, 9i, 10 克
- 托姆貓 7（使用 Java 1.6 或更高版本）

架構

源, 技術, 堆棧

- 使用 Oracle 建置的 Web 應用程式 WebLogic
- 使用甲骨文網閘或 SAML 驗證的 Web 應用程式
- 連線至 Oracle 資料庫版本 10g 及更新版本的 Web 應用程式

目標技術堆疊

- 在 Amazon ECS 上運行的 Tomme（具有添加容器支持的 Apache Tomcat）（另請參閱在 Amazon ECS 上部署 Java Web 應用程序和 Java 微服務）
- 適用於甲骨文的亞馬 Amazon Relational Database Service（Amazon RDS）；對於 Amazon RDS 支持的甲骨文版本，請參閱 Amazon RDS for Oracle

目標架構

工具

若要在 ToMee 上作業，Java 應用程式必須重建成 .war 檔案。在某些情況下，可能需要變更應用程式才能在 ToMee 上操作應用程式；您應該檢查以確定必要的組態選項和環境屬性已正確定義。

此外，應正確定義 Java 命名和目錄介面 (JNDI) 查詢和 JavaServer 頁面 (JSP) 命名空間。 考慮檢查應用程序使用的文件名，以避免與內置 T 庫的命名衝突。例如，persistence.xml 是由 Apache 的 OpenJPA 框架（這是捆綁在東米的 OpenEJB）用於配置目的的的文件名。在 PUI 中的 persistence.xml 文件包含春季框架豆聲明。

TomEe 7.0.3 及更新版本 (Tomcat 8.5.7 及更新版本) 會針對具有特殊字元的原始 (未編碼) 網址傳回 HTTP 400 回應 (錯誤要求)。伺服器回應會顯示為一般使用者的空白頁面。早期版本的 TomEe 和 Tomcat 允許在 URL 中使用某些未編碼的特殊字符; 但是，它被認為是不安全的，如 CVE-2016-6816 網站上所述。要解決 URL 編碼問題，直接通過傳遞給瀏覽器的 URL JavaScript 必須使用 encodeURI () 方法進行編碼，而不是用作原始字符串。

在 ToMee 中部署 .war 檔案之後，請監視 Linux cat 中是否有任何遺失的共用程式庫和 Oracle 特定延伸模組的開始記錄，以便從 Tomcat 程式庫新增遺失的元件。

一般程序

- 在 TomEE 上設定應用程式。
- 識別並重新設定應用程式伺服器特定的組態檔案和資源，從來源到目標格式。
- 識別並重新設定 JNDI 資源。
- 將 EJB 命名空間和查詢調整為目標應用程式伺服器所需的格式 (如果適用)。
- 重新設定 JAAS 應用程式容器特定的資訊安全角色和原則對應 (如果適用)。
- 將應用程式和共用程式庫 Package 到 .war 檔案中。
- 使用提供的碼頭容器，在 TomEe 中部署 .war 檔案。
- 監視開始記錄檔，以識別任何遺失的共用程式庫和部署描述元延伸 如果找到任何東西，請返回到第一個任務。
- 對照還原的 Amazon RDS 資料庫測試已安裝的應用程式。
- 遵循部署 D ocker 容器中的指示，透過負載平衡器和 Amazon ECS 叢集啟動完整架構。
- 更新 URL 以指向負載平衡器。
- 更新組態管理資料庫 (CMDB)。

史诗

規劃移轉

任務	描述	所需技能
執行應用程式探索 (目前的狀態佔用空間和效能基準)。		BA，遷移領導
驗證來源和目標資料庫版本和引擎。		DBA
驗證來源和目標應用程式設計 (身分識別和工作階段管理)。		DBA、移轉工程師、應用程式擁有者
識別目標伺服器執行個體的硬體和儲存需求。		DBA SysAdmin
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA SysAdmin

任務	描述	所需技能
識別來源和目標資料庫的網路存取安全性需求。		DBA SysAdmin
識別應用程式遷移策略和工具。		DBA，遷移領導
完成應用程式的移轉設計和移轉指南。		建立領導者，遷移領導者
完成應用程式移轉手冊。		構建領導，切換引線，測試導線，遷移領導

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		SysAdmin
建立安全性群組。		SysAdmin
設定並啟動 Amazon RDS 資料庫執行個體。		DBA SysAdmin
設定 Amazon ECS 部署。		SysAdmin
將您的應用程式 Package 為 Docker 映像檔。		SysAdmin
將映像推送到 Amazon ECR 登錄 (或跳過此步驟並將其推送到 Amazon ECS 叢集)。		SysAdmin
設定應用程式和 Amazon ECS 服務選項的任務定義。		SysAdmin

任務	描述	所需技能
設定叢集、檢閱安全設定，以及設定 AWS Identity and Access Management (IAM) 角色。		SysAdmin
啟動您的設置並根據您的應用程序遷移 runbook 運行測試。		SysAdmin

移轉資料

任務	描述	所需技能
取得安全保證團隊的許可，將生產資料移至 AWS。		DBA、移轉工程師、應用程式擁有者
建立並取得端點的存取權，以擷取資料庫備份檔案。		DBA
使用原生資料庫引擎或協力廠商工具來移轉資料庫物件和資料。		DBA
從應用程式移轉 Runbook 執行必要的測試，以確認資料移轉成功。		DBA、移轉工程師、應用程式擁有者

移轉應用程式

任務	描述	所需技能
建立要移轉的變更請求 (CR)。		切換引線
取得移轉的 CR 核准。		切換引線

任務	描述	所需技能
遵循應用程式遷移手冊中的應用程式遷移策略。		DBA、移轉工程師、應用程式擁有者
升級應用程序（如果需要）。		DBA、移轉工程師、應用程式擁有者
完整的功能性、非功能性、資料驗證、SLA 和效能測試。		測試潛在客戶，應用程序所有者，應用

切過

任務	描述	所需技能
取得應用程式或企業擁有者的簽署。		切換引線
執行表格主題練習，逐步執行切換工作簿的所有步驟。		DBA、移轉工程師、應用程式擁有者
將應用程式用戶端切換至新基礎結構。		DBA、移轉工程師、應用程式擁有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA，移民工程師，SysAdmin
審核並驗證專案文件。		遷移, 領導
收集移轉時間的指標、手動與工具的百分比、節省成本等。		遷移, 領導
關閉專案並提供意見反應。		應用程式擁有者移轉主管

相關資源

參考

- 阿帕奇湯姆貓 7.0 文檔
- 阿帕奇湯姆貓 7.0 安裝指南
- 阿帕奇雄貓 JNDI 文檔
- 阿帕奇托米文檔
- Amazon RDS for Oracle
- Amazon RDS 定價
- 甲骨文和 AWS
- Amazon 上的甲骨文 RDS 文檔
- Amazon RDS 異地備份部署
- 開始使用 Amazon ECS
- Amazon RDS 入門

教學課程和影片

- 在 Amazon RDS 上運行甲骨文數據庫的最佳實踐（RE：創新 2018 演示文稿）

使用 AWS DMS 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS

R 類型：重新平台	來源：數據庫：關係	目標：Amazon RDS for Oracle
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作量：甲骨文	AWS 服務：Amazon EC2; Amazon RDS	

Summary

此模式描述了使用 AWS Database Migration Service (AWS DMS) 將亞馬遜彈性運算雲端 (Amazon EC2) 上的 Oracle 資料庫遷移到適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 的步驟。該模式還使用 Oracle SQL 開發人員或 SQL *Plus 連接到 Oracle 資料庫執行個體，並包含可自動執行某些任務的 AWS CloudFormation 範本。

遷移到 Amazon RDS for Oracle 可讓您專注於業務和應用程式，同時 Amazon RDS 負責佈建資料庫、備份和復原、安全修補程式、版本升級和儲存管理等資料庫管理任務。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon EC2 上甲骨文數據庫的亞馬遜機器映像（AMI）

產品版本

- AWS DMS 支援適用於企業版、標準版、標準版和標準二版本的 Amazon RDS 執行個體資料庫的甲骨文 11g (版本 11.2.0.3.v1 及更新版本)、12c 和 18c 版本。如需支援版本的最新資訊，請參閱 AWS 文件中的使用 Oracle 資料庫做為 AWS DMS 的目標。(附加的 AWS CloudFormation 範本使用 Oracle 12c 版作為來源資料庫。)
- 甲骨文 SQL 開發人員 4.0.3

架構

來源架構

- Amazon EC2 上的 Oracle 數據庫

目標架構

- Amazon RDS for Oracle

移轉架構

工具

- AWS DMS — AWS Database Migration Service (AWS DMS) 可協助您快速安全地將資料庫遷移到 AWS。它支援同質和異質移轉。如需有關受支援的 Oracle 資料庫版本和版本的資訊，請參閱 AWS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源和使用 Oracle 資料庫做為 AWS DMS 的目標。
- Oracle SQL 開發人員或 SQL *Plus — 這些工具可讓您連線到適用於 Oracle 資料庫的 Amazon RDS 執行個體。

史诗

設定您的目標資料庫

任務	描述	所需技能
建立適用於 Oracle 資料庫執行個體的亞馬遜 RDS。	登入 AWS 管理主控台，開啟位於 https://console.aws.amazon.com/rds/ 的 Amazon RDS 主控台。透過選取適當的引擎、範本、資料庫證明資料設定、執行個體類型、儲存體、異地同步備份設定、虛擬私人雲端 (VPC) 和組態、登入證明資料以及 Oracle 資料庫的	開發人員

任務	描述	所需技能
	其他設定，來建立 Oracle 資料庫執行個體。如需相關指示，請檢視「相關資源」一節中的連結。或者使用附件中的 AWS CloudFormation 範本 (建立 RDS.YAML) 建立適用於甲骨文資料庫執行個體的 Amazon RDS。	
Connect 到 Amazon RDS 並授予權限給甲骨文用戶。	修改安全群組以開啟適當的連接埠，以從本機機器和 AWS DMS 複寫執行個體連接。設定連線時，請確定已選取「可公開存取」選項，以便您可以從 VPC 外部連線到資料庫。使用 Oracle SQL 開發人員或 SQL *Plus Connect 到 Amazon RDS，方法是使用登入登入資料、建立 AWS DMS 使用者，並提供必要的權限給 AWS DMS 使用者以修改資料庫。	開發人員

設定來源 EC2 執行個體的安全群組

任務	描述	所需技能
檢查 Oracle 資料庫是否已啟動且正在執行。	使用安全殼層 (SSH) 連線至 EC2 執行個體，並嘗試使用 SQL *Plus 連線至 Oracle 資料庫。	開發人員
修改安全性群組。	修改 EC2 執行個體的安全群組以開啟適當的連接埠，以便從本機機器和 AWS DMS 複寫執行個體進行連線。	開發人員

設定 AWS DMS

任務	描述	所需技能
建立 AWS DMS 複寫執行個體。	在 AWS DMS 中，在與您的 Amazon RDS for Oracle 資料庫執行個體相同的 VPC 中建立複寫執行個體。指定複寫執行個體的名稱和說明、選擇執行個體類別和複寫引擎版本 (使用預設值)、選擇建立 Amazon RDS 資料庫執行個體的 VPC、視需要設定異地同步備份設定、分配儲存、指定可用區域，以及設定其他設定。或者，您也可以使用附件中的 AWS CloudFormation 範本 (DMS.YAML) 來實作此步驟。	DBA
Connect 至來源和目標資料庫端點。	透過指定端點識別碼、引擎、伺服器、連接埠、登入認證和額外連線屬性來建立來源和目標資料庫端點。對於來源伺服器，請使用託管 Oracle 資料庫之 EC2 執行個體的公有 DNS。對於目標伺服器，請使用適用於甲骨文的亞馬遜 RDS 端點。執行測試以確認來源和目標連線是否正常運作。或者，您也可以使用附件中的 AWS CloudFormation 範本 (DMS.YAML) 來實作此步驟。	DBA
建立 AWS DMS 任務。	建立 AWS DMS 任務以將資料從來源端點遷移到目標端點，以設定來源端點和/或目標	DBA

任務	描述	所需技能
	端點之間的複寫。建立 AWS DMS 任務時，請指定複寫執行個體、來源端點、目標端點、遷移類型 (僅限資料、僅複寫或兩者)、表格對應和篩選器。執行 AWS DMS 任務、監控任務、查看表格統計資料，以及在 Amazon CloudWatch 中檢查日誌。或者，您也可以使用附件中的 AWS CloudFormation 範本 (DMS.YAML) 來實作此步驟。	

相關資源

- 建立 Amazon RDS 資料庫執行個體
- 連接至執行 Oracle 資料庫引擎的資料庫執行個體
- AWS DMS 說明文件
- AWS DMS 逐步解說逐步解說
- 將甲骨文資料庫遷移到 AWS 雲端

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Logstash 將現場部署 Oracle 資料庫遷移至 Amazon OpenSearch 服務

由阿迪亞·戈特蒂（AWS）創建

環境：PoC 或試點	來源：甲骨文數據庫	目標：Amazon OpenSearch 服務
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon OpenSearch 服務		

Summary

此模式說明如何使用 Logstash 將資料從現場部署 Oracle 資料庫移至 Amazon OpenSearch 服務。它包括架構考量，以及一些必要的技能和建議。資料可以來自單一資料表，也可以來自需要執行全文檢索搜尋的多個資料表。

OpenSearch 服務可以在虛擬私有雲（VPC）中進行配置，也可以在基於 IP 的限制下公開放置。此模式描述了 VPC 中配置 OpenSearch 服務的案例。Logstash 用於從 Oracle 數據庫中收集數據，將其解析為 JSON 格式，然後將數據饋送到 OpenSearch 服務中。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Java 8（由登錄信息 6.4.3 所需）
- 使用 AWS 虛擬私有網路 (AWS VPN) 建立的 VPC 中的現場部署資料庫伺服器與 Amazon 彈性運算雲端 (Amazon EC2) 執行個體之間的連線
- 用於從資料庫擷取要推送至 OpenSearch 服務的必要資料的查詢
- 甲骨文 Java 數據庫連接（JDBC）驅動程序

限制

- Logstash 無法識別從資料庫中硬刪除的記錄

產品版本

- Oracle 資料庫 12c
- OpenSearch 客戶服務
- 洛格斯塔什 6.4.3

架構

源, 技術, 堆棧

- 內部部署甲骨文
- 現場部署 AWS VPN

目標技術堆疊

- VPC
- EC2 執行個體
- OpenSearch 服務
- Logstash
- NAT 閘道 (適用於 EC2 執行個體上的作業系統更新，以及安裝 Java 8、記錄檔和外掛程式)

資料移轉架構

工具

- 洛格斯塔什 6.4.3
- JDBC 輸入插件（下載和更多信息）
- 落格輸出插件（_es）logstash-output-amazon
- 甲骨文 JDBC 驅動程序

史诗

規劃移轉

任務	描述	所需技能
識別來源資料的大小。	來源資料的大小是您用來決定索引中要設定之碎片數目的其中一個參數。	DBA, 資料庫開發人員
分析每列的數據類型和相應的數據。	OpenSearch 當文檔中找到以前看不到的字段時，Service 動態映射數據類型。如果有任何特定的資料類型或格式 (例如，日期欄位) 需要明確宣告，請識別欄位並在建立索引期間定義這些欄位的對應。	應用程式擁有者、開發人員、資料
確定是否有任何具有主鍵或唯一鍵的列。	若要避免在更新或插入期間重複 Amazon Ser OpenSearc h vice 中的記錄，您需要在 `amazon_es` 外掛程式的輸出區段中進行`document_id` 設定 (例如，`document_id => "%{customer_id}"` 其中`customer_id` 是主索引鍵)。	應用程式擁有者、開
分析新增記錄的數量和頻率；檢查刪除記錄的頻率。	需要這項工作才能瞭解來源資料的成長速度。如果數據集中讀取並且插入很少見，則可以使用單個索引。如果經常插入新記錄且沒有刪除，則碎片大小可以輕鬆超過建議的 50 GB 大小上限。在這種情況下，您可以通過在 Logstash 和可以使	應用程式擁有者、開

任務	描述	所需技能
	用別名訪問它的代碼中配置索引模式來動態創建索引。	
決定需要多少複本。		應用程式擁有者、開
決定要在索引上設定的碎片數目。		應用程式擁有者、開
識別專用主節點、資料節點和 EC2 執行個體的執行個體類型。	如需詳細資訊，請參閱相關資源一節。	應用程式擁有者、開
確定所需的專用主節點和數據節點的數量。	如需詳細資訊，請參閱相關資源一節。	

移轉資料

任務	描述	所需技能
啟動 EC2 執行個體。	在連接 AWS VPN 的虛擬私人 VPC 內啟動 EC2 執行個體。	Amazon VPC 構建，AWS VPN
在 EC2 執行個體上安裝記錄檔。		開發人員
安裝 Logstash 插件。	安裝所需的 Logstash 外掛程式`jdbc-input` 和 `logstash-output-amazon_es`	開發人員
配置記錄信息。	建立 Logstash 金鑰儲存庫來儲存敏感資訊，例如 AWS Secrets Manager 金鑰和資料庫登入資料，然後將參考資料放在 Logstash 組態檔中。	開發人員

任務	描述	所需技能
設定無效字母佇列和永久佇列。	根據預設，當 Logstash 遇到因為資料包含對應錯誤或其他問題而無法處理的事件時，Logstash 管線會掛起或捨棄失敗的事件。若要在此情況下防止資料遺失，您可以設定 Logstash 將不成功的事件寫入無效字母佇列，而不是捨棄它們。為了在異常終止期間防止資料遺失，Logstash 具有永久佇列功能，可將訊息佇列儲存在磁碟上。持續性佇列提供 Logstash 中的資料耐久性。	開發人員
創建 Amazon OpenSearch 服務域。	使用不需要使用 AWS Identity and Access Managemen t (IAM) 登入資料簽署請求的存取政策建立 Amazon OpenSearch 服務網域。Amaz on OpenSearch 服務域必須在同一個 VPC 中創建。您也應該選取例證類型，並根據您的分析設定專用節點和主節點的數目。	開發人員
設定所需的 Amazon OpenSearch 服務日誌。	如需詳細資訊，請參閱OpenSearch 服務文件。	
建立索引。		開發人員

任務	描述	所需技能
啟動記錄儲存。	以背景服務的形式執行記錄檔。Logstash 運行配置的 SQL 查詢，提取數據，將其轉換為 JSON 格式，並將其饋送到 OpenSearch 服務。對於初始載入，請勿在 Logstash 組態檔中設定排程器。	開發人員
檢查文件。	檢查索引上的文檔數量，以及是否所有文檔都存在於源數據庫中。在初始加載期間，它們被添加到索引中，並用於停止 Logstash。 變更 Logstash 組態以新增根據用戶端需求以固定間隔執行的排程器，然後重新啟動 Logstash。Logstash 只會挑選上次執行之後更新或新增的記錄，而上次執行時間戳記會儲存在使用 Logstash 組態檔案中屬性設定的檔案`last_run_metadata_path => “/usr/share/logstash/.logstash_jdbc_last_run”` 中。	開發人員

相關資源

- 建議的 CloudWatch 鬧鐘
- 專用 Amazon OpenSearch 服務主節點
- 調整 Amazon OpenSearch 服務域
- 日誌記錄文檔
- JDBC 輸入插件

- 落格輸出插件
- Amazon OpenSearch 服務網站

將現場部署甲骨文資料庫遷移到 Amazon RDS for Oracle

由白芝謝克（AWS）和帕萬普蘇魯裡（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS; AWS DMS		

Summary

此模式說明將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 的步驟。在移轉過程中，您可以建立移轉計劃，並根據來源資料庫考慮有關目標資料庫基礎結構的重要因素。您可以根據業務需求和使用案例，選擇以下兩種移轉選項之一：

1. AWS Database Migration Service (AWS DMS) — 您可以使用 AWS DMS 快速安全地將資料庫遷移到 AWS 雲端。您的來源資料庫會在移轉期間保持完整運作，如此可將依賴資料庫的應用程式停機時間降到最低。您可以使用 AWS DMS 建立任務，在透過名為變更資料擷取 (CDC) 的程序完成初始完全負載遷移後擷取持續變更，以縮短移轉時間。如需詳細資訊，請參閱 AWS 文件中的使用 AWS DMS 從甲骨文遷移到 Amazon RDS。
2. 原生 Oracle 工具 — 您可以使用原生的 Oracle 工具來移轉資料庫，例如 Oracle 與資料汲取匯出，以及 Oracle 適用 GoldenGate於 CDC 的 「資料汲取匯入」。您也可以使用原始的 Oracle 工具，例如原始的「匯出」公用程式和原始的「匯入」公用程式，以減少滿載時間。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署 Oracle 資料庫
- 一個 Amazon RDS 甲骨文數據庫（數據庫）實例

限制

- 資料庫大小限制：64 TB

產品版本

- 甲骨文版本 11 克（版本 11.2.0.3.v1 及更高版本）以及最高 12.2 和 18c。如需最新的受支援版本和版本清單，請參閱 AWS 文件中的亞馬遜 RDS 版。如需 AWS DMS 支援的 Oracle 版本，請參閱 AWS DMS 文件中的使用 Oracle 資料庫做為 AWS DMS 的來源。

架構

源, 技術, 堆棧

- 內部部署甲骨文

目標技術堆疊

- Amazon RDS for Oracle

來源與目標架構

下圖顯示如何使用 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS

該圖顯示以下工作流程：

1. 建立或使用現有的資料庫使用者、將所需的 AWS DMS 許可授與該使用者、開啟 ARCHIVELOG 模式，然後設定補充記錄。
2. 設定現場部署和 AWS 網路之間的網際網路閘道。
3. 設定 AWS DMS 的來源和目標端點。
4. 設定 AWS DMS 複寫任務，將資料從來源資料庫遷移到目標資料庫。
5. 完成目標資料庫上的移轉後活動。

下圖顯示如何使用原生 Oracle 工具將現場部署 Oracle 資料庫遷移至 Amazon RDS for Oracle 版甲骨文。

該圖顯示以下工作流程：

1. 建立或使用現有的資料庫使用者，並使用 Oracle 匯出 () 和 Import (expimp) 公用程式授與備份 Oracle 資料庫的必要權限。
2. 設定現場部署和 AWS 網路之間的網際網路閘道。
3. 在防禦主機上設定 Oracle 用戶端以取得備份資料庫。
4. 將備份資料庫上傳到亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。
5. 將資料庫備份從 Amazon S3 還原到 Amazon RDS for Oracle 資料庫。
6. 設定 GoldenGate 適用於 CDC 的甲骨文。
7. 完成目標資料庫上的移轉後活動。

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署設定的組合之間遷移資料存放區。
- Oracle 原生工具可協助您執行同質移轉。您可以使用「Oracle 資料汲取」在來源資料庫和目標資料庫之間移轉資料。此病毒碼會使用「Oracle 資料汲取」來執行從來源資料庫到目標資料庫的完整載入。
- Oracle 可 GoldenGate協助您在兩個或多個資料庫之間執行邏輯複寫。此模式用 GoldenGate 於使用「Oracle 資料汲取」，在初始載入之後複製差異變更。

史诗

規劃移轉

任務	描述	所需技能
建立專案文件並記錄資料庫詳細資料。	1. 記錄您的移轉目標、移轉需求、主要專案利益相關者、專案里程碑、專案截止日期、關鍵指標、移轉風險和風險緩解計畫。 2. 記錄有關源數據庫的重要信息，包括 RAM，IOPS 和 CPU。您稍後將使用此資訊	DBA

任務	描述	所需技能
	來決定適當的目標資料庫執行個體。 3. 驗證來源資料庫和目標資料庫的版本。	
識別儲存需求。	識別並記錄您的儲存需求，包括下列項目： 1. 計算配置給來源資料庫執行個體的儲存體。 2. 從來源資料庫執行個體收集歷史成長指標。 3. Forecast 目標資料庫執行個體的 future 成長。 注意：針對一般用途 (gp2) 固態硬碟磁碟區，每 1 GB 的儲存空間可獲得三個 IOPS。 透過計算來源資料庫上的讀取和寫入 IOPS 總數來配置儲存體。	DBA, SysAdmin
根據運算需求選擇適當的執行個體類型。	1. 判斷目標資料庫執行個體的運算需求。 2. 識別效能問題。 3. 考慮決定適當執行個體類型的因素： • 來源資料庫執行個體的 CPU 使用率 • 來源資料庫執行個體的 IOPS (讀取和寫入) • 來源資料庫執行個體的記憶體佔用	SysAdmin

任務	描述	所需技能
識別網路存取安全性需求。	1. 識別並記錄來源和目標資料庫的網路存取安全性需求。 2. 設定適當的安全群組，讓應用程式能夠與資料庫通訊。	DBA, SysAdmin
識別應用程式移轉策略。	1. 決定並記錄移轉切換策略。 2. 決定並記錄應用程式的復原時間目標 (RTO) 與復原點目標 (RPO)，然後據此規劃切換。	DBA,, 應用程式 SysAdmin擁有者
識別移轉風險。	評估資料庫和記錄移轉的特定風險和緩解措施。例如： • 識別無記錄表，並突出顯示恢復時數據丟失的風險。 • 擷取來源資料庫使用者和權限，並反白顯示與 Amazon RDS 權限的衝突。 • 複查警示日誌中是否有任何 Oracle 特定的錯誤和警告。 • 識別目標資料庫執行個體受支援和不受支援的功能。 • 檢閱目標資料庫版本引擎的已停用功能。	DBA

設定基礎結構

任務	描述	所需技能
建立 VPC。	為目標資料庫執行個體建立新的 Amazon Virtual Private Cloud (Amazon VPC)。	SysAdmin

任務	描述	所需技能
建立安全性群組。	在新 VPC 中建立安全群組，以允許資料庫執行個體的輸入連線。	SysAdmin
建立適用於 Oracle 資料庫執行個體的亞馬遜 RDS。	使用新的 VPC 和安全群組建立目標資料庫執行個體，然後啟動執行個體。	SysAdmin

(選項 1) 使用原生 Oracle 或協力廠商工具來移轉資料

任務	描述	所需技能
準備來源資料庫。	1. 建立「資料汲取」目錄或使用既有目錄。 2. 建立移轉使用者並授與執行資料汲取的權限。 3. 將來源資料庫中的角色、使用者和表格空間擷取為 SQL 命令檔。 4. 將擷取的資料汲取傾印傳輸到目標資料庫執行個體`data pump`目錄。	DBA, SysAdmin
準備目標資料庫。	1. 確認已在目標 Amazon RDS 適用於 Oracle 資料庫執行個體上安裝或啟用所有資料庫選項 (例如文字和 Java)。 2. 建立「資料汲取」目錄或使用既有目錄。 3. 建立移轉使用者並授與執行資料汲取匯入的權限。	DBA, SysAdmin

任務	描述	所需技能
	4. 在目標資料庫執行個體上建立必要的表格空間、使用者和角色。 5. 將傳輸的「資料汲取」匯出傾印匯入目標資料庫。 6. 建立在匯入或建立物件期間排除的任何索引。 7. 建立在匯入期間排除的任何限制。 8. 驗證或重新編譯無效的物件。 9. 重建無效的索引。 10驗證來源與目標資料庫之間的資料庫物件計數。 11解決物件計數之間發現的任何差異。	

(選項 2) 使用 AWS DMS 遷移資料

任務	描述	所需技能
準備資料。	1. 清除來源資料庫中的資料。 2. 建立複寫執行個體。 3. 建立來源端點和目標端點。 4. 識別要移轉的表格和物件數目。	DBA
移轉資料。	1. 刪除目標資料庫的外部索引鍵限制和觸發。 2. 卸除目標資料庫上的次要索引。	DBA

任務	描述	所需技能
	3. 設定從來源資料庫到目標資料庫的 AWS DMS 全載任務設定。 4. 啟用外鍵。 5. 讓 AWS DMS CDC 能夠複寫進行中的變更。 6. 啟用觸發器。 7. 更新序列。 8. 驗證源和目標數據。	

切換到目標數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。	1. 停止所有指向 Oracle 的應用程式服務和從屬端連線。 2. 執行 AWS DMS 任務。 3. 設定復原任務 (例如，將 CDC 從 Amazon RDS 資料庫反轉至現場部署 Oracle 資料庫)。 4. 驗證資料。 5. 將 Amazon 路線 53 設定為適用於 Oracle 資料庫的新 Amazon RDS 執行個體，在新的目標資料庫上啟動應用程式服務。 6. 將 Amazon CloudWatch 監控新增到新的 Amazon RDS for Oracle 資料庫執行個體。	DBA,, 應用程式 SysAdmin擁有者

任務	描述	所需技能
實施您的回滾計劃。	1. 停止所有指向適用 Oracle 資料庫執行個體之 Amazon RDS 的應用程式服務。 2. 使用 AWS DMS 任務將變更還原至來源現場部署 Oracle 資料庫。 3. 停止從現場部署 Oracle 資料庫執行的 AWS DMS 任務到適用於甲骨文資料庫的 Amazon RDS。 4. 重新設定來源 Oracle 資料庫上的應用程式。 5. 確認復原部署已完成。	DBA，應用程式擁有者

關閉移轉專案

任務	描述	所需技能
清理資源。	關閉或移除臨時 AWS 資源，例如 AWS DMS 複寫執行個體和 S3 儲存貯體。	DBA, SysAdmin
檢閱專案文件。	檢閱您的移轉規劃文件和目標，然後確認您已完成所有必要的移轉步驟。	DBA,, 應用程式 SysAdmin擁有者
收集指標。	記錄重要的移轉指標，包括完成移轉所花費的時間、手動與工具型作業的百分比、節省成本，以及其他相關指標。	DBA,, 應用程式 SysAdmin擁有者
關閉專案。	關閉遷移專案並擷取有關工作量的意見反應。	DBA,, 應用程式 SysAdmin擁有者

相關資源

參考

- 將 Oracle 資料庫遷移到 AWS 的策略 (AWS 白皮書)
- AWS Database Migration Service (AWS DMS 文件)
- Amazon RDS 定價 (Amazon RDS 文檔)

教學課程和影片

- 開始使用 AWS Database Migration Service (AWS DMS 文件)
- Amazon RDS 資源 (Amazon RDS 文檔)
- AWS Database Migration Service (DMS) (YouTube)

使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到亞馬遜 RDS

由莫漢安南（AWS）和布萊恩莫特澤（AWS）創建

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式說明如何使用 Oracle 資料泵，將 Oracle 資料庫從現場部署資料中心遷移到適用於 Oracle 資料庫執行個體的 Amazon 關聯式資料庫服務 (Amazon RDS)。

該模式涉及從來源資料庫建立資料傾印檔案，將檔案存放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中，然後將資料還原到適用於 Oracle 資料庫執行個體的 Amazon RDS。當您在使用 AWS Database Migration Service (AWS DMS) 進行遷移時遇到限制時，此模式非常有用。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 在 AWS Identity and Access Management (IAM) 和 Amazon S3 多部分上傳中建立角色所需的許可
- 從來源資料庫匯出資料所需的權限
- 安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)

產品版本

- 「Oracle 資料汲取管理系統」僅適用於 Oracle 資料庫 10g 發行版本 1 (10.1) 及更新版本。

架構

源, 技術, 堆棧

- 內部部署甲骨文

目標技術堆疊

- Amazon RDS for Oracle
- SQL 從屬端 (Oracle SQL 開發人員)
- S3 儲存貯體

來源與目標架構

工具

AWS 服務

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。在這種模式中，IAM 用於建立將資料從 Amazon S3 遷移到亞馬遜 RDS 版甲骨文所需的角色和政策。
- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

- Oracle 資料汲取可協助您以高速將資料和中繼資料從一個資料庫移至另一個資料庫。在此模式中，Oracle 資料泵用於將資料傾印 (.dmp) 檔案匯出到 Oracle 伺服器，並將其匯入 Amazon RDS for Oracle 文。如需詳細資訊，請參閱 Amazon RDS 文件中的將資料匯入 Oracle。
- Oracle SQL 開發人員是一個整合式開發環境，可簡化傳統與雲端式部署中 Oracle 資料庫的開發與管理。它會與現場部署 Oracle 資料庫和 Amazon RDS for Oracle 互動，以執行匯出和匯入資料所需的 SQL 命令。

史诗

建立 S3 儲存貯體

任務	描述	所需技能
建立儲存貯體。	若要建立 S3 儲存貯體，請遵循 AWS 文件中的指示。	AWS 系統管理員

建立 IAM 角色並指派政策

任務	描述	所需技能
設定 IAM 權限。	若要設定許可，請遵循 AWS 文件中的指示。	AWS 系統管理員

建立適用於 Oracle 資料庫執行個體的目標 Amazon RDS，並將 Amazon S3 整合角色建立關聯

任務	描述	所需技能
建立適用於甲骨文資料庫執行個體的目標 Amazon RDS。	若要建立 Amazon RDS for Oracle 執行個體，請遵循 AWS 文件中的指示進行。	AWS 系統管理員
將角色與資料庫執行個體建立關聯。	若要將角色與執行個體建立關聯，請遵循 AWS 文件中的指示。	DBA

在目標資料庫上建立資料庫使用者

任務	描述	所需技能
建立使用者。	從 Oracle SQL 開發人員或 SQL*Plus Connect 到目標亞馬遜 RDS 適用於 Oracle 資料庫，然後執行下列 SQL 命令來	DBA

任務	描述	所需技能
	建立要將結構描述匯入的使用者。 `create user SAMPLE_SCHEMA identified by <PASSWORD>;` `grant create session, resource to <USER NAME>;` `alter user <USER NAME> quota 100M on users;`	

從來源 Oracle 資料庫建立匯出檔案

任務	描述	所需技能
建立資料傾印檔案。	若要在匯出使用SAMPLE_SCHEMA 者的DATA_PUMP_DIR 目錄sample.dmp 中建立名為的傾印檔案，請使用下列指令碼。 `DECLARE` `hdnl NUMBER;` `BEGIN` `hdnl := dbms_datapump.open(operation => 'EXPORT',` `job_mode => 'SCHEMA',` `job_name => NULL);` `dbms_datapump.add_file( handle => hdnl,`	DBA

任務	描述	所需技能
	filename => 'sample.dmp', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file); dbms_datapump.add_file(handle => hdnl, filename => 'export.log', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file); dbms_datapump.metadata_filter(hdnl, 'SCHEMA_EXPR', 'IN ('''SAMPLE_SCHEMA''')'); dbms_datapump.start_job(hdnl); END; / 檢閱本機DATA_PUMP_DIR 目錄中的export.log 檔案，以檢閱匯出詳細資訊。	

將轉儲文件上傳到 S3 存儲桶

任務	描述	所需技能
將資料傾印檔案從來源上傳至 S3 儲存貯體。	使用 AWS CLI 執行下列命令。 `aws s3 cp sample.dmp s3://<bucket_created_epic_1>/`	DBA

將匯出檔案從 S3 儲存貯體下載到 RDS 執行個體

任務	描述	所需技能
將數據轉儲文件下載到 Amazon RDS	若要將傾印檔案`sample.dmp` 從 S3 儲存貯體複製到 Amazon RDS for Oracle 資料庫，請執行下列 SQL 命令。在此範例中，`sample.dmp` 檔案會從 S3 儲存貯體下載`my-s3-integration1` 至 Oracle 目錄`DATA_PUMP_DIR` 。請確定您有足夠的磁碟空間配置給 RDS 執行個體，以容納資料庫和匯出檔案。 `-- If you want to download all the files in the S3 bucket remove the p_s3_prefix line.` `SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(` `p_bucket_name => 'my-s3-integration',`	AWS 系統管理員

任務	描述	所需技能
	p_s3_prefix => 'sample.dmp', p_directory_name => 'DATA_PUMP_DIR') AS TASK_ID FROM DUAL; 上一個指令會輸出工作 ID。若要透過檢閱工作 ID 中的資料來檢閱下載狀態，請執行下列命令。 SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-<task_id>.log')); 若要查看目DATA_PUMP_DIR 錄中的檔案，請執行下列命令。 SELECT filename, type,filesize/1024/1024 size_megs ,to_char(mtime,'DD-MON-YY HH24:MI:SS') timestamp FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => upper('DATA_PUMP_DIR'))) order by 4;	

將傾印檔案匯入目標資料庫

任務	描述	所需技能
將結構描述和資料還原到 Amazon RDS。	若要將傾印檔案匯入 `sample_schema` 資料庫結構描述，請從 SQL 開發人員或 SQL*Plus 執行下列 SQL 命令。 `DECLARE hdnl NUMBER; BEGIN hdnl := DBMS_DATAPUMP.OPEN(operation => 'IMPORT', job_mode => 'SCHEMA', job_name=>null); DBMS_DATAPUMP.ADD_FILE(handle => hdnl, filename => 'sample.dmp', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_dump_file); DBMS_DATAPUMP.ADD_FILE(handle => hdnl, filename => 'import.log', directory => 'DATA_PUMP_DIR', filetype => dbms_datapump.ku$_file_type_log_file); DBMS_DATAPUMP.METADATA_FILTER(hdnl,'SCHEMA_EXPR','`	DBA

任務	描述	所需技能
	IN (''SAMPLE_SCHEMA'')'); DBMS_DATAPUMP.START_J OB(hdnl); END; / 若要查看匯入中的記錄檔，請執行下列命令。 SELECT text FROM table(rdsadmin.rds _file_util.read_te xt_file('DATA_PUMP _DIR','import.log'));	

從資料目錄中移除傾印檔案

任務	描述	所需技能
列出並清理匯出檔案。	列出並刪除目DATA_PUMP _DIR 錄中的導出文件，運行以下命令。 -- List the files SELECT filename, type,filesize/1024 /1024 size_megs ,to_char(mtime,'DD -MON-YY HH24:MI:S S') timestamp FROM TABLE(rdsadmin.rds _file_util.listdir (p_directory =>	AWS 系統管理員

任務	描述	所需技能
	upper('DATA_PUMP_DIR'))) order by 4; -- Remove the files EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample.dmp'); EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','import.log');	

相關資源

- Amazon S3 整合
- 建立資料庫執行個體
- 在 Amazon RDS 上將數據導入甲骨文
- Amazon S3 文件
- IAM 文件
- Amazon RDS 文件
- Oracle 資料汲取說明文件
- Oracle SQL Developer

使用合格邏輯從 Amazon EC2 上的 PostgreSQL 遷移到亞馬遜 RDS

創建者：拉傑什·馬迪瓦利 (AWS)

環境：PoC 或試點	資料來源：Amazon EC2	目標：Amazon RDS for PostgreSQL
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫
AWS 服務：Amazon RDS		

Summary

此模式概述了將 PostgreSQL 資料庫（9.5 版及更新版本）從 Amazon Elastic Compute Cloud (Amazon EC2) 遷移到使用 PostgreSQL 的 Amazon Relational Database Service 服務 (Amazon RDS) 的步驟。Amazon RDS 現在支持第 10 版的複 PostgreSQL 擴展。

先決條件和限制

先決條件

- 選擇正確的 Amazon RDS 執行個體類型。如需詳細資訊，請參閱 Amazon RDS 執行個體類型。
- 請確定 PostgreSQL 的來源版本和目標版本是相同的。
- 在 Amazon EC2 上使用 PostgreSQL 安裝和整合副檔名。

產品版本

- 在 Amazon RDS 上使用 PostgreSQL 版本 10 及更新版本，並具有 Amazon RDS 上支援的功能 (請參閱 AWS 文件中的 Amazon RDS 上的 PostgreSQL)。這個模式是透過將 Amazon RDS 上的 PostgreSQL 9.5 遷移到第 10 版進行測試，但也適用於 Amazon RDS 上的較新版本。

架構

資料移轉架構

工具

- PG逻辑扩展
- PostgreSQL 用程序：pg_dump 和 PG_ 恢復

史诗

通過使用 pglogical 擴展遷移數據

任務	描述	所需技能
建立一個 Amazon 資料庫執行個體。	在 Amazon RDS 中設定 PostgreSQL 資料庫執行個體。如需指示，請參閱適用於 PostgreSQL 的亞馬遜 RDS 文件。	DBA
從來源 PostgreSQL 資料庫取得結構描述傾印，並將其還原到目標 PostgreSQL 資料庫中。	1. 使用 pg_dump 公用程式並-s選擇從來源資料庫產生結構描述檔案。 2. 使用 psql 公用程式並-f選擇將結構描述載入目標資料庫。	DBA
開啟邏輯解碼。	在 Amazon RDS 資料庫參數群組中，將rds.logical_replication 靜態參數設定為 1。如需指示，請參閱 Amazon RDS 文件。	DBA
在來源資料庫和目標資料庫上建立 pglogical 延伸模組。	1. 在來源 PostgreSQL 資料庫上建立pglogical 擴充功能： `psql -h <amazon-ec2-endpoint> -d target-dbname -U target-dbuser -c "create`	DBA

任務	描述	所需技能
	extension pglogical ;" 2. 在目標 PostgreSQL 資料庫上建立pglogical 擴充功能： `psql -h <amazon-rds-endpoint> -d source-dbname -U source-dbuser -c "create extension pglogical ;"`	
在來源 PostgreSQL 資料庫上建立發行者。	若要建立發行者，請執行： `psql -d dbname -p 5432 <<EOF` `SELECT pglogical.create_node( node_name := 'provider1', dsn := 'host=<ec2-endpoint> port=5432 dbname=source-dbname user=source-dbuser' );` `EOF`	DBA

任務	描述	所需技能
建立複製組、新增表格和序列。	若要在來源 PostgreSQL 資料庫上建立複製組，並將表格和序列新增至複製組，請執行： `psql -d dbname -p 5432 <<EOF` `SELECT pglogical.replication_set_add_all_tables('default', '{public}'::text[],synchronize_data := true);` `EOF`	DBA
建立訂閱者。	若要在目標 PostgreSQL 資料庫上建立訂閱者，請執行： `psql -h <rds-endpoint> -d target-dbname -U target-dbuser <<EOF` `SELECT pglogical.create_node(` `node_name := 'subscriber1',` `dsn := 'host=<rds-endpoint> port=5432 dbname=target-dbname password=postgres user=target-dbuser'` `);` `EOF`	DBA

任務	描述	所需技能
建立訂閱。	若要在目標 PostgreSQL 資料庫上建立訂閱，請執行： `psql -h <rds-endpoint> -d target -U postgres <<EOF SELECT pglogical.create_subscription( subscription_name := 'subscription1', replication_sets := array['default'], provider_dsn := 'host=<ec2-endpoint> port=5432 dbname=<source-dbname> password=<password> user=source-dbuser' );`	DBA

驗證您的資料

任務	描述	所需技能
檢查來源和目標資料庫。	檢查來源和目標資料庫，確認資料已成功複製。您可以使用來`select count(1)`自來源和目標資料表來執行基本驗證。	DBA

相關資源

- Amazon RDS
- 在 Amazon RDS 上適用於 PostgreSQL 的邏輯複寫 (Amazon RDS 文件)
- 生物学 (GitHub 存儲庫)

- pglogical 的局限性（GitHub 存儲庫自述文件）
- 使用邏輯複寫將 PostgreSQL 從現場部署或亞馬 Amazon EC2 遷移到 Amazon RDS (AWS 資料庫部落格)

將內部 PostgreSQL 料庫遷移至 Aurora

由白芝謝克（AWS）和吉泰爾庫馬爾（AWS）創建

環境：PoC 或試點	來源：內部部署資料庫	目標：Aurora 郵政兼容
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫
AWS 服務：Amazon Aurora；AWS DMS		

Summary

Amazon Aurora PostgreSQL 相容版本結合了高階商業資料庫的效能和可用性，以及開放原始碼資料庫的簡易性和成本效益。Aurora 透過將儲存擴展到同一 AWS 區域中的三個可用區域，並支援最多 15 個僅供讀取複本執行個體，以擴展讀取工作負載並在單一區域內提供高可用性，從而提供這些好處。透過使用 Aurora 全球資料庫，您可以在最多五個區域複寫 PostgreSQL 資料庫，以便在發生區域故障時進行遠端讀取存取和災難復原。此模式說明將內部部署 PostgreSQL 來源資料庫移轉至 Aurora PostgreSQL 相容資料庫的步驟。該模式包括兩個遷移選項：使用 AWS 資料遷移服務 (AWS DMS) 或使用原生 PostgreSQL 工具 (例如 pg_dump、pg_restore 和 psql) 或第三方工具。

此模式中描述的步驟也適用於 Amazon 關聯式資料庫服務 (Amazon RDS) 和亞馬遜 Elastic Compute Cloud (Amazon EC2) 執行個體上的目標 PostgreSQL 資料庫。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的 PostgreSQL 來源資料庫
- 相容於 Aurora 的資料庫執行個體或亞馬遜 RDS 版資料 PostgreSQL 行個體

限制

- 適用於 Amazon RDS 的資料庫大小限 PostgreSQL 64 TB，對於 Aurora 相容，則為 128 TB。
- 如果您使用 AWS DMS 遷移選項，請參閱 AWS DMS 有關使用 PostgreSQL 資料庫做為來源的限制。

產品版本

- 如需 PostgreSQL 的主要和次要版本支援，請參閱 Amazon RDS 文件中的 Amazon RDS 更新。
- 如需 Aurora 中的 PostgreSQL 支援，請參閱 Aurora 文件中的 Amazon Aurora PostgreSQL 更新。
- 如果您使用 AWS DMS 遷移選項，請參閱 AWS DMS 文件中支援的 PostgreSQL 版本。

架構

源, 技術, 堆棧

- 內部部署資料庫

目標技術堆疊

- Aurora 與 PostSQL 相容的資料庫執行個體

來源架構

目標架構

資料移轉架構

使用 AWS DMS

使用 PostgreSQL 具

工具

- AWS Database Migration Service (AWS DMS) 可協助您將資料存放區遷移到 AWS 雲端，或在雲端和現場部署組態的組合之間遷移資料存放區。此服務支援不同的來源和目標資料庫。如需如何驗證支援與 AWS DMS 搭配使用的 PostgreSQL 來源和目標資料庫版本和版本的詳細資訊，請參閱使用

PostgreSQL 資料庫做為 AWS DMS 來源。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。

- 原生 PostgreSQL 工具包括轉儲，pg _還原和psql。

史诗

分析移轉

任務	描述	所需技能
驗證來源和目標資料庫版本。	如果您使用的是 AWS DMS，請確定您使用的是受支援的 PostgreSQL 版本。	DBA
識別儲存類型和容量需求。	1. 計算配置給來源資料庫執行處理的儲存體。 2. 收集來源資料庫執行處理的歷史成長測量結果。 3. 預測目標資料庫執行處理的 future 成長預測。 4. 透過計算來源資料庫上的讀取和寫入 IOPS 總數來配置儲存體。一般用途 SSD (gp2) 磁碟區為每 1 GB 的儲存空間提供 3 IOPS。	DBA, 系統管理員
選擇適當的執行個體類型、容量、儲存功能和網路功能。	判斷目標資料庫執行處理的運算需求。檢閱可能需要額外注意的已知效能問題。請考慮下列因素來決定適當的執行個體類型： • 來源資料庫執行處理的 CPU 使用率 • 來源資料庫執行處理的 IOPS (讀取和寫入作業)	DBA, 系統管理員

任務	描述	所需技能
	• 來源資料庫執行處理的記憶體容量 有關詳情，請參閱 Aurora 文件中的 Aurora 資料庫執行個體類別。	
識別來源和目標資料庫的網路存取安全性需求。	決定可讓應用程式與資料庫通訊的適當安全群組。	DBA, 系統管理員
識別應用程式移轉策略。	• 根據應用程式的複雜性決定移轉切換策略。 • 決定應用程式的復原時間目標 (RTO) 與復原點目標 (RPO)，並據此規劃切換。	DBA、應用程式擁有者、系統管理員

設定基礎結構

任務	描述	所需技能
建立 VPC。	為目標資料庫執行個體建立新的虛擬私有雲 (VPC)。	系統管理員
建立安全性群組。	在 VPC 中創建一個安全組（如上一史詩中確定），以允許對數據庫實例的入站連接。	系統管理員
設定並啟動 Aurora 資料庫叢集。	使用新的 VPC 和安全群組建立目標資料庫執行個體，然後啟動執行個體。	系統管理員

遷移資料 — 選項 1 (使用 AWS DMS)

任務	描述	所需技能
完成移轉前步驟。	1. 清理來源資料庫中的資料。 2. 建立複寫執行個體。 3. 建立來源端點和目標端點。 4. 識別要移轉的可用表格和物件數目。	DBA
完成移轉步驟。	1. 刪除目標資料庫的外部索引鍵限制和觸發。 2. 卸除目標資料庫的次要索引。 3. 使用完整載入工作將資料從來源移轉至目標資料庫。 4. 啟用外鍵。 5. 如果您使用快閃移轉，而應用程式需要最少的停機時間，請啟用變更資料擷取 (CDC) 以複寫進行中的變更 6. 啟用觸發器。 7. 更新序列。 8. 驗證源和目標數據。	DBA
驗證資料。	若要確保您的資料已從來源準確移轉到目標，請遵循 AWS DMS 文件中的資料驗證步驟。	DBA

遷移數據-選項 2（使用 pg_dump 和 PG_ 恢復）

任務	描述	所需技能
準備來源資料庫。	1. 創建一個目錄來存儲 pg_dump 備份（如果它不存在）。 2. 建立具有在資料庫物件上執行 pg_dump 權限的移轉使用者。 3. Connect 至 EC2 執行個體並執行 pg_dump 備份。 如需詳細資訊，請參閱 pg_dump 文件和 AWS DMS 文件中的逐步解說。	DBA
準備目標資料庫。	1. 建立具有在資料庫物件上使用 pg_restore 權限的移轉使用者。 2. 使用 pg_restore 匯入資料庫傾印。 如需詳細資訊，請參閱 pg_restore 文件和 AWS DMS 文件中的逐步解說。	DBA
驗證資料。	1. 比較來源與目標資料庫之間的資料庫物件計數。 2. 解決物件計數之間發現的任何差異。	DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。	實作您在第一個史詩中建立的應用程式遷移策略。	DBA、應用程式擁有者、系統管理員

切換到目標數據庫

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。	1. 停止所有指向內部部署 PostgreSQL 資料庫的應用程式服務和用戶端連線。 2. 執行 AWS DMS 任務。 3. 如有需要，請設定復原工作 (從與 Aurora PostgreSQL 相容的疾病控制中心反轉至內部部署 PostgreSQL 資料庫)。 4. 驗證資料。 5. 將 Amazon Route 53 設定為與 Aurora PostgreSQL 相容的新資料庫執行個體，在新目標上啟動應用程式服務。 6. 在 CloudWatch與 Aurora PostgreSQL 相容的新資料庫執行個體上新增 Amazon 和 Performance Insights 監控。	DBA、應用程式擁有者、系統管理員
如果您需要回滾遷移。	1. 停止所有指向 Aurora PostgreSQL 相容資料庫的應用程式服務。	DBA，應用程式擁有者

任務	描述	所需技能
	2. 使用您在上一個故事中建立的 AWS DMS 任務，將變更還原至來源現場部署 PostgreSQL 資料庫。 3. 停止從現場部署 PostgreSQL 資料庫執行的 AWS DMS 任務，移至與 Aurora PostgreSQL 相容的資料庫。 4. 設定應用程式，使其指向來源內部部署 PostgreSQL 資料庫。 5. 確認所有復原部署都已完成。	

關閉專案

任務	描述	所需技能
關閉資源。	關閉臨時 AWS 資源。	DBA, 系統管理員
驗證文件。	審核並驗證專案文件。	DBA、應用程式擁有者、系統管理員
收集指標。	收集移轉時間的指標、手動與工具成本節約的百分比等。	DBA、應用程式擁有者、系統管理員
關閉專案。	關閉專案並提供任何意見反應。	DBA、應用程式擁有者、系統管理員

相關資源

參考

- AWS 資料遷移服務

- VPC 和 Amazon Aurora
- Amazon Aurora 定價
- 使用 PostgreSQL 資料庫做為 AWS DMS 來源
- 如何建立 AWS DMS 複寫執行個體
- 如何使用 AWS DMS 建立來源和目標端點

其他資源

- 開始使用 AWS DMS
- 資料移轉 step-by-step 逐步解說
- Amazon Aurora 資源

將現場部署 Microsoft SQL 伺服器資料庫遷移到執行 Linux 的 Amazon EC2 上的 Microsoft SQL 伺服器

創建者：蒂魯馬拉·達薩里 (AWS)

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon EC2 Linux 與 Microsoft SQL 服務器
R 類型：重新平台	工作負載：Microsoft	技術：移轉；資料庫
AWS 服務：Amazon EC2		

Summary

此模式說明如何使用備份和還原公用程式，從 Microsoft 視窗上執行的現場部署 Microsoft SQL Server 資料庫遷移到 Amazon Elastic Compute Cloud (Amazon EC2) Linux 執行個體上的 Microsoft SQL 伺服器。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon EC2 Linux AMI（Amazon 機器映像）與 Microsoft SQL 服務器
- AWS 在現場部署視窗和 Linux EC2 執行個體上的 Microsoft SQL 伺服器之間直接連接

架構

源, 技術, 堆棧

- 內部部署的 Microsoft SQL 伺服器

目標技術堆疊

- 使用 Microsoft SQL 伺服器資料庫的 EC2 執行個體

資料庫遷移架構

工具

- WinSCP-這個工具使 Windows 用戶可以輕鬆地與 Linux 用戶共享文件。
- Sqlcmd-這個命令列公用程式可讓您提交 T-SQL 陳述式或批次到 SQL 伺服器的本機和遠端執行個體。該實用程序對於重複的數據庫任務（例如批處理或單元測試）非常有用。

史诗

使用 SQL 伺服器準備執行個體

任務	描述	所需技能
選擇一個提供 Linux 操作系統並包括 Microsoft SQL 服務器的 AMI。		系統管理員
設定 AMI 以建立 EC2 執行個體。		系統管理員
建立安全性群組的輸入和輸出規則。		系統管理員
設定適用於 Microsoft SQL 伺服器資料庫的執行個體。		DBA
建立使用者並提供來源資料庫中的權限。		受理人, DBA
在 EC2 執行個體上安裝 SQL 伺服器工具和 sqlcmd 公用程式。		DBA

備份資料庫並將備份檔案移至 Linux EC2 執行個體

任務	描述	所需技能
備份內部部署 SQL 伺服器資料庫。		DBA
在 Microsoft SQL 伺服器上安裝 WinSCP。		DBA
將備份檔案移至執行 Microsoft SQL 伺服器的執行個體。		DBA

在執行 SQL 伺服器的 Linux EC2 執行個體上還原資料庫

任務	描述	所需技能
使用 sqlcmd 公用程式，從資料庫備份檔案還原資料庫。		DBA
驗證數據庫對象和數據。		開發者，測試工程師

從視窗 SQL 伺服器切斷到視窗 SQL 伺服器

任務	描述	所需技能
驗證數據庫對象和數據。		開發者，測試工程師
從現場部署 Microsoft SQL 伺服器資料庫切換到執行 Microsoft SQL 伺服器的 Linux EC2 執行個體。		DBA

相關資源

- 如何在 Amazon Linux 2 和 Ubuntu 的 AMI 配置 SQL 服務器 2017

- 在執行個體上安裝 SQL 工具
- 在 Linux EC2 執行個體上，從內部部署 Microsoft SQL 伺服器資料庫 Backup 和還原至 Microsoft SQL 伺服器

使用連結伺服器將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server 服器

R 類型：重新平台	來源：數據庫：關係	目標：Amazon RDS for Microsoft SQL Server
創建者：AWS	環境：生產	技術：資料庫；移轉
工作量：Microsoft	AWS 服務：Amazon RDS	

Summary

連結伺服器可讓 Microsoft SQL Server 在資料庫伺服器的其他執行個體上執行 SQL 陳述式。此模式說明如何將現場部署 Microsoft SQL Server 資料庫遷移到 Amazon Relational Database Service 服務 (Amazon RDS)，以實現更低的成本和更高的可用性。目前，Amazon RDS for Microsoft SQL Server 不支持 Amazon Virtual Private Cloud（Amazon VPC）網絡之外的連接。

您可以使用此模式來達成下列目標：

- 要遷移 Microsoft SQL 服務器到 Amazon RDS for Microsoft SQL Server 沒有破壞鏈接的服務器功能。
- 為了優先級和遷移鏈接 Microsoft SQL 服務器在不同的波浪。

先決條件和限制

先決條件

- 檢查 Amazon RDS 上的 Microsoft SQL 服務器是否支持您需要的功能。
- 請確定您可以使用具有預設定序或在資料庫層級上設定定序的定序的 Amazon RDS for Microsoft SQL Server)。

架構

源, 技術, 堆棧

- 內部部署資料庫 (Microsoft SQL 伺服器)

目標技術堆疊

- Amazon RDS for SQL Server

源狀態架構

目標狀態架構

在目標狀態下，您遷移 Microsoft SQL 服務器到 Amazon RDS for Microsoft SQL Server 通過使用鏈接的服務器。此架構使用 Network Load Balancer 將流量從亞馬遜 RDS 適用於 Microsoft SQL 伺服器代理到執行微軟 SQL 伺服器的現場部署伺服器。下圖顯示 Network Load Balancer 的反向 Proxy 功能。

工具

- AWS CloudFormation
- Network Load Balancer
- Amazon RDS for SQL Server 的多個可用區域 (多可用區域)
- AWS Database Migration Service

史诗

建立 landing zone 域 VPC

任務	描述	所需技能
建立 CIDR 配置。		AWS SysAdmin

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		AWS SysAdmin
建立 VPC 子網路。		AWS SysAdmin
建立子網路存取控制清單 (ACL)。		AWS SysAdmin
建立子網路路由表。		AWS SysAdmin
建立與 AWS Direct Connect 或 AWS 虛擬私人網路 (VPN) 的連線。		AWS SysAdmin

將數據庫遷移到 Amazon RDS

任務	描述	所需技能
創建一個 Amazon RDS for Microsoft SQL Server 數據庫實例。		AWS SysAdmin
建立 AWS DMS 複寫執行個體。		AWS SysAdmin
在 AWS DMS 中建立來源和目標資料庫端點。		AWS SysAdmin
建立移轉任務，並在完整載入後將連續複寫設定為開啟。		AWS SysAdmin
請求防火牆變更，以允許 Amazon RDS for Microsoft SQL Server 存取現場部署 SQL 伺服器資料庫。		AWS SysAdmin

任務	描述	所需技能
建立 Network Load Balancer。		AWS SysAdmin
建立以資料中心中資料庫伺服器為目標的目標群組	建議您在目標設定中使用主機名稱來納入資料中心 (DC) 容錯移轉事件。	AWS SysAdmin
執行連結伺服器設定的 SQL 陳述式。	針對 Microsoft SQL 伺服器資料庫執行個體使用微軟 SQL 管理工具，執行 SQL 陳述式以新增連結伺服器。在 SQL 陳述式中，將 @datasrc 設定為使用 Network Load Balancer 主機名稱。使用 Microsoft SQL 管理工具對於微軟 SQL 伺服器資料庫執行個體的亞馬遜 RDS 來新增連結伺服器登入認證。	AWS SysAdmin
測試和驗證 SQL 伺服器函數。		AWS SysAdmin
建立切換。		AWS SysAdmin

相關資源

- Amazon RDS 上 Microsoft SQL 服務器的常見管理任務
- Microsoft SQL 伺服器的定序和字元集
- Network Load Balancer 文件
- 使用亞馬遜 RDS 為微軟 SQL 服務器實現鏈接服務器（博客文章）

使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS

創建者：蒂魯馬拉·達薩里（AWS）、大衛·奎羅斯（AWS）和維沙爾辛格（AWS）

環境：PoC 或試點	來源：本地 SQL 伺服器資料庫	目標：Amazon RDS for SQL Server
R 類型：重新平台	工作量：Microsoft	技術：移轉、資料庫、作業系統
AWS 服務：Amazon RDS; Amazon S3		

Summary

此模式說明如何將現場部署 SQL Server Microsoft 料庫遷移到適用於 SQL Server 資料庫執行個體 (同質遷移) 的 Amazon Relational Database Service 服務 (Amazon RDS)。遷移程序是以原生 SQL Server 備份和還原方法為基礎。它使用 SQL 伺服器管理工作室 (SSMS) 建立資料庫備份檔案，並使用 Amazon Simple Storage Service (Amazon S3) 儲存貯體來存放備份檔案，然後再將備份檔案還原到 Amazon RDS for SQL Server 服器中。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Identity and Access Management (IAM) 角色政策可存取 S3 儲存貯體和適用於 SQL 伺服器資料庫執行個體的 Amazon RDS。

限制

- 此病毒碼中描述的程序只會移轉資料庫。不會遷移 SQL 登入或資料庫使用者，包括任何 SQL Server 代理程式作業，因為他們需要額外的步驟。

產品版本

- SQL 伺服器 如需支援的版本和功能的最新清單，請參閱 AWS 文件中的 Amazon RDS 上的 Microsoft SQL 伺服器。

架構

源, 技術, 堆棧

- 內部部署 Microsoft SQL 伺服器資料庫

目標技術堆疊

- Amazon RDS for SQL Server 數據庫實例

資料移轉架構

工具

- Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器基礎設施的集成環境。它提供了一個用戶界面和一組具有與 SQL Server 交互的豐富腳本編輯器的工具。

史诗

建立 Amazon RDS for SQL Server 資料庫執行個體

任務	描述	所需技能
選取 SQL 伺服器做為 SQL 伺服器的亞馬遜 RDS 中的資料庫引擎。		DBA
選擇 SQL 伺服器快速版。		DBA
指定資料庫詳細資訊	如需有關建立資料庫執行個體的詳細資訊，請參閱 Amazon RDS 文件。	DBA，應用程式擁有者

從內部部署 SQL Server 資料庫建立備份檔案

任務	描述	所需技能
透過 SSMS Connect 到內部部署 SQL 伺服器資料庫。		DBA
建立資料庫的備份。	如需指示，請參閱 SSMS 文件。	DBA，應用程式擁有者

將備份檔案上傳到 Amazon S3

任務	描述	所需技能
在 Amazon S3 中建立儲存貯體。	如需詳細資訊，請參閱 Amazon S3 說明文件。	DBA
將備份檔案上傳到 S3 儲存貯體。	如需詳細資訊，請參閱 Amazon S3 說明文件。	SysOps 管理員

在 Amazon RDS for SQL Server 恢復數據庫

任務	描述	所需技能
將選項組添加到 Amazon RDS。	1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。 2. 在功能窗格中，選擇 [選項群組] > [建立群組]。 3. 完成選項群組的資訊，然後選擇「建立」。 4. 將SQLSERVER_BACKUP_RESTORE 選項新增至選項群組，然後選擇 [新增] 選項。	SysOps 管理員

任務	描述	所需技能
	如需詳細資訊，請參閱 Amazon RDS 文件。	
還原資料庫。	1. 透過 SSMS Connect 到 Amazon RDS for SQL Server。 2. 呼叫預`msdb.dbo.rds_restore_database` 存程序來還原資料庫。	DBA

驗證目標資料庫

任務	描述	所需技能
驗證對象和數據。	驗證來源資料庫和亞馬遜 RDS SQL 伺服器之間的物件和資料。 注意：此作業只會移轉資料庫。登入和工作將不會移轉。	應用程式擁有者，DBA

切過

任務	描述	所需技能
重定向應用程式流量	驗證後，將應用程式流量重新導向至適用於 SQL 伺服器資料庫的 Amazon RDS 執行個體	應用程式擁有者，DBA

相關資源

- Amazon S3 文件
- Amazon RDS for SQL Server 文檔

- Microsoft SQL 伺服器資料庫引擎的選項

使用 AWS DMS 和 AWS SCT 將 Microsoft SQL 伺服器資料庫遷移到 Aurora MySQL

R 類型：重新平台	來源：數據庫：關係	目標：Amazon Aurora MySQL
創建者：AWS	環境：PoC 或試點	技術：資料庫；移轉
工作負載：Microsoft	AWS 服務：Amazon Aurora	

Summary

此模式說明如何將內部部署或 Amazon 彈性運算雲端 (亞馬遜 EC2) 執行個體上的 Microsoft SQL Server 資料庫遷移到 Amazon Aurora MySQL。該模式使用 AWS Database Migration Service (AWS DMS) 和 AWS Schema Conversion Tool (AWS SCT) 進行資料遷移和結構描述轉換。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 位於現場部署資料中心或 EC2 執行個體上的 Microsoft SQL 伺服器來源資料庫
- 適用於 AWS SCT 連接器的 Java 資料庫連線能力 (JDBC) 驅動程式，安裝在本機電腦或已安裝 AWS SCT 的 EC2 執行個體

限制

- 資料庫大小限制：64 TB

產品版本

- 適用於企業、標準、工作群組和開發人員版本的 Microsoft SQL 伺服器 2008 年、2008R2、二零一四年、二零一七年。AWS DMS 不支援網頁版和快速版本。如需支援版本的最新清單，請參閱使用 Microsoft SQL 伺服器資料庫做為 AWS DMS 的來源。我們建議您使用最新版本的 AWS DMS 以獲得最全面的版本和功能支援。如需 AWS SCT 支援的 Microsoft SQL 伺服器版本的相關資訊，請參閱 AWS SCT 文件。

- MySQL 5.5、5.6 和 5.7 版。如需支援版本的最新清單，請參閱使用與 MySQL 相容的資料庫做為 AWS DMS 的目標。

架構

源, 技術, 堆棧

下列其中一項：

- 內部部署 Microsoft SQL 伺服器資料庫
- EC2 實例上的 Microsoft SQL 服務器數據庫

目標技術堆疊

- Aurora MySQL

資料移轉架構

- 從在 AWS 雲端執行的 Microsoft SQL 伺服器資料庫

- 從在內部部署資料中心執行的 Microsoft SQL Server 資料庫

工具

- AWS DMS-AWS 資料遷移服務 (AWS DMS) 可協助您在廣泛使用的商業和開放原始碼資料庫 (包括甲骨文、SQL 伺服器、MySQL 和 PostgreSQL) 之間移轉資料，或從廣泛使用的商業和開放原始碼資料庫遷移 您可以使用 AWS DMS 將資料遷移至 AWS 雲端，可在現場部署執行個體 (透過 AWS 雲端設定) 或在雲端和現場部署設定之間進行。
- AWS SCT-AWS Schema Conversion Tool (AWS SCT) 透過自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式，讓異質資料庫遷移變得簡單。

史诗

準備移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
建立來源和目標資料庫的輸出安全性群組。		SysAdmin
視需要建立和設定 AWS SCT 的 EC2 執行個體。		DBA
下載最新版本的 AWS SCT 和相關驅動程式。		DBA
在來源資料庫中新增及驗證先決條件使用者和授權。		DBA
為工作負載建立 AWS SCT 專案，並連線到來源資料庫。		DBA
產生評估報告並評估可行性。		DBA

準備目標資料庫

任務	描述	所需技能
使用 Amazon Aurora 做為資料庫引擎，建立目標 Amazon RDS 資料庫執行個體。		DBA
從來源擷取使用者、角色和授權的清單。		DBA
將現有的資料庫使用者對應至新的資料庫使用者。		應用所有者

任務	描述	所需技能
在目標資料庫中建立使用者。		DBA
將上一個步驟中的角色套用至目標資料庫。		DBA
複查來源資料庫中的資料庫選項、參數、網路檔案和資料庫連結，然後評估它們對目標資料庫的適用性。		DBA
將任何相關設定套用至目標。		DBA

傳送物件

任務	描述	所需技能
設定與目標資料庫的 AWS SCT 連線。		DBA
使用 AWS SCT 轉換結構描述。	AWS SCT 會自動將來源資料庫結構描述和大部分自訂程式碼轉換為與目標資料庫相容的格式。該工具無法自動轉換的任何代碼都會清楚標記，以便您可以自行轉換。	DBA
複查產生的 SQL 報告，並儲存任何錯誤和警告。		DBA
將自動化結構描述變更套用至目標，或將其儲存為 .sql 檔案。		DBA
驗證 AWS SCT 是否在目標上建立了物件。		DBA

任務	描述	所需技能
手動重寫、拒絕或重新設計任何無法自動轉換的項目。		DBA
套用產生的角色和使用者授與，並檢閱任何例外狀況。		DBA

遷移數據

任務	描述	所需技能
確定遷移方法。		DBA
從 AWS DMS 主控台建立複寫執行個體。	如需使用 AWS DMS 的詳細資訊，請參閱「相關資源」一節中的連結。	DBA
建立來源端點和目標端點。		DBA
建立複製工作。		DBA
啟動複寫工作並監視記錄檔。		DBA

移轉應用程式

任務	描述	所需技能
使用 AWS SCT 分析和轉換應用程式程式碼中的 SQL 項目。	當您將資料庫結構描述從一個引擎轉換到另一個引擎，您也需更新應用程式中的 SQL 程式碼，以便與新的資料庫引擎互動，取代舊引擎。您可以檢視、分析、編輯和儲存轉換後的 SQL 程式碼。如需使用 AWS SCT 的詳細資訊，請	應用所有者

任務	描述	所需技能
	參閱「相關資源」一節中的連結。	
在 AWS 上建立新的應用程式伺服器。		應用所有者
將應用程式程式碼移轉至新伺服器。		應用所有者
設定目標資料庫和驅動程式的應用程式伺服器。		應用所有者
修正應用程式中原始碼資料庫引擎專屬的任何程式碼。		應用所有者
最佳化目標引擎的應用程式程式碼。		應用所有者

切過

任務	描述	所需技能
將任何新使用者、授權和程式碼變更套用至目標。		DBA
鎖定應用程式以進行任何變更。		應用所有者
驗證是否已將所有變更傳播至目標資料庫。		DBA
將新的應用程式伺服器指向目標資料庫。		應用所有者
重新檢查一切。		應用所有者
去直播吧。		應用所有者

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源 (用於 AWS SCT 的 AWS DMS 複寫執行個體和 EC2 執行個體)。		DBA，應用程式擁有者
針對內部團隊更新 AWS DMS 程序的意見反應。		DBA，應用程式擁有者
修改 AWS DMS 程序並視需要改善範本。		DBA，應用程式擁有者
審核並驗證專案文件。		DBA，應用程式擁有者
收集移轉時間的指標、手動與工具成本節約的百分比等。		DBA，應用程式擁有者
關閉專案並提供任何意見反應。		DBA，應用程式擁有者

相關資源

參考

- AWS DMS 使用者指南
- 使用者指南
- Amazon Aurora 定價

教學課程和影片

- 開始使用 AWS Database Migration Service
- 開始使用 AWS 結 Schema Conversion Tool
- Amazon RDS 資源
- AWS DMS 逐步解說逐步解說

使用原生工具將現場部署 MariaDB 資料庫遷移到適用於 MariaDB 的 Amazon RDS

創建者：辛德·拉赫查 (AWS)

環境：PoC 或試點	來源：數據庫：關係	目標：Amazon RDS for MariaDB
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫

Summary

此模式提供使用原生工具將現場部署 MariaDB 資料庫移轉至適用於 MariaDB 的 Amazon Relational Database Service 服務 (Amazon RDS) 的指引。如果你已經安裝了 MySQL 工具，你可以使用 MySQL 和 mysql dump。如果您安裝了 MariaDB 工具，則可以使用馬里亞德 B 和瑪麗亞德-轉儲。MySQL 和 MariaDB 的工具具有相同的起源，但在 MariaDB 版本 10.6 及更高版本中存在細微差異。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的 MariaDB 來源資料庫

限制

- 資料庫大小限制：64 TB

產品版本

- MariaDB 10.0-10.6 版 (如需支援的最新版本清單，請參閱 AWS 文件中的 Amazon RDS 上的 MariaDB)

架構

源, 技術, 堆棧

- 位於內部部署資料中心的 MariaDB 資料庫

目標技術堆疊

- Amazon RDS for MariaDB 資料庫執行個體

目標架構

資料移轉架構

工具

- 本地 MySQL 工具：MySQL 和神址轉儲
- 原生 MariaDB 工具：瑪麗亞德和瑪麗亞德-轉儲

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA
識別目標伺服器執行處理的硬體需求。		DBA, 系統管理員
識別儲存需求 (儲存類型和容量)。		DBA, 系統管理員
根據容量、儲存空間功能和網路功能選擇適當的執行個體類型。		DBA, 系統管理員

任務	描述	所需技能
識別來源和目標資料庫的網路存取安全性需求。		DBA, 系統管理員
識別應用程式移轉策略。		DBA、應用程式擁有者、系統管理員

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		系統管理員
建立安全性群組。		系統管理員
設定並啟動執行 MariaDB 的 Amazon RDS 資料庫執行個體。		系統管理員

移轉資料

任務	描述	所需技能
使用原生工具移轉資料庫物件和資料。	在源數據庫中，使用 mysqldump 或 MARIADB- 轉儲創建包含數據庫對象和數據的輸出文件。在目標數據庫中，使用 mysql 或瑪麗亞德來恢復數據。	DBA
驗證資料。	檢查來源和目標資料庫，確認資料移轉成功。	DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA、應用程式擁有者、系統管理員

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		系統管理員
審核並驗證專案文件。		DBA、應用程式擁有者、系統管理員
收集有關遷移時間的指標，工具提供的成本節省等等。		DBA、應用程式擁有者、系統管理員
關閉專案並提供意見反應。		DBA、應用程式擁有者、系統管理員

相關資源

Amazon RDS 參考

- Amazon RDS for MariaDB
- Amazon Virtual Private Cloud VPC 和 Amazon RDS
- Amazon RDS 異地備份部署

- Amazon RDS 定價

MySQL 和 MariaDB 的參考

- 瑪麗亞德轉儲/mysql轉儲
- MySQL 命令行客戶端

教學課程和影片

- Amazon RDS 入門

將內部部署 MySQL 資料庫遷移至 Aurora MySQL

由維諾德·庫馬爾·薩杜（AWS）和伊戈爾·奧布拉多維奇（AWS）創建

環境：生產	來源：內部部署 MySQL 數據庫	目標：Amazon Aurora MySQL 兼容版
R 類型：重新平台	工作負載：開源	技術：移轉；資料庫
AWS 服務：AWS DMS		

Summary

此模式說明如何將現場部署 MySQL 來源資料庫遷移至 Amazon Aurora MySQL 相容版本。它描述了兩個遷移選項：使用AWS Database Migration Service（AWS DMS）或使用本地 MySQL 工具，如 mysqldbcopy 和 mysql dump。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料中心中的來源 MySQL 資料庫

限制

- 資料庫大小限制：64 TB

產品版本

- MySQL 版本 5.7 和 8.0. 如需支援版本的最新清單，請參閱AWS文件中的 Amazon Aurora 版本。如果您正在使用AWS DMS，請參閱使用與 MySQL 相容的資料庫作為支援的 MySQL 版本AWS DMS 的目標。AWS DMS

架構

源, 技術, 堆棧

- 內部部署 MySQL 資料庫

目標技術堆疊

- Amazon Aurora MySQL-Compatible Edition

目標架構

資料移轉架構

使用AWS DMS：

使用本地 MySQL 工具：

工具

- AWS Database Migration Service(AWS DMS) 支援數個來源和目標資料庫。如需有關 MySQL 來源和目標資料庫支援的詳細資訊AWS DMS，請參閱將 MySQL 相容資料庫移轉至. AWS 我們建議您使用最新版本的，以AWS DMS獲得最全面的版本和功能支援。
- mysqldbcopy 是一個 MySQL 實用程序，可在單個服務器上或服務器之間複製 MySQL 數據庫。
- mysqldump 是一個 MySQL 實用程序，它從 MySQL 數據庫中創建一個轉儲文件，用於備份或遷移目的。

史诗

規劃移轉

任務	描述	所需技能
驗證來源和目標資料庫版本和引擎。		DBA

任務	描述	所需技能
識別目標伺服器執行處理的硬體需求。		DBA, 系統管理員
識別儲存需求 (儲存類型和容量)。		DBA, 系統管理員
根據容量、儲存空間功能和網路功能，選擇適當的執行個體類型。		DBA, 系統管理員
識別來源和目標資料庫的網路存取安全性需求。		DBA, 系統管理員
識別應用程式移轉策略。		DBA、應用程式擁有者、系統管理員

設定基礎結構

任務	描述	所需技能
建立 Virtual Private Cloud (VPC)		系統管理員
建立安全性群組。		系統管理員
設定並啟動與 Aurora 相容的資料庫叢集。		系統管理員

移轉資料-選項 1

任務	描述	所需技能
使用原生 MySQL 工具或協力廠商工具來移轉資料庫物件和資料。	有關說明，請參閱 MySQL 工具的文檔，例如 mysqldbcopy 和 mysql dump。	DBA

移轉資料-選項 2

任務	描述	所需技能
使用移轉資料AWS DMS。	如需指示，請參閱使用與 MySQL 相容的資料庫做為來源和使用 MySQL 相容資料庫做為文件中的目標。AWS DMS	DBA

移轉應用程式

任務	描述	所需技能
遵循應用程式遷移策略。		DBA、應用程式擁有者、系統管理員

切過

任務	描述	所需技能
將應用程式用戶端切換到新的基礎結構。		DBA、應用程式擁有者、系統管理員

關閉專案

任務	描述	所需技能
關閉臨時 AWS 資源。		DBA, 系統管理員
審核並驗證專案文件。		DBA、應用程式擁有者、系統管理員
收集移轉時間的指標、手動與工具的百分比、節省成本等。		DBA、應用程式擁有者、系統管理員

任務	描述	所需技能
關閉專案並提供意見反應。		

相關資源

參考

- 將資料庫遷移到 Amazon Aurora
- 數據管理系統網站
- AWS DMS 說明文件
- Amazon Aurora 定價
- 建立並連線至 Aurora MySQL 資料庫叢集
- Amazon Virtual Private Cloud VPC 和 Amazon RDS
- Amazon Aurora 文檔

教學課程和影片

- 開始使用 AWS DMS
- 開始使用 Amazon Aurora

使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL

創建者：羅漢牙買加尼（AWS），薩吉斯梅農（AWS）和烏達亞西姆哈·蒂皮雷迪（AWS）

來源：內部部署	目標：Aurora	R 類型：重新平台
環境：生產	技術：資料庫；移轉	工作負載：開源
AWS 服務：Amazon S3; Amazon Aurora; Amazon EFS		

Summary

此模式說明如何使用佩科納 XtraBackup，將大型的現場部署 MySQL 資料庫有效地遷移到 Amazon Aurora MySQL。Percona XtraBackup 是適用於 MySQL 伺服器的開放原始碼、非封鎖備份公用程式。該模式顯示如何使用 Amazon Elastic File System (Amazon EFS) 來減少將備份上傳到亞馬遜簡單儲存服務 (Amazon S3) 的時間，以及將備份還原到 Amazon Aurora MySQL 的時間。此病毒碼也提供如何進行增量 Percona 備份的詳細資訊，以盡量減少套用至目標 Aurora MySQL 資料庫的二進位記錄數目。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 建立 AWS Identity and Access Management (IAM) 角色和政策的許可
- 現場部署 MySQL 資料庫與 AWS 上的虛擬私有雲 (VPC) 之間的網路連線

限制

- 來源伺服器必須是能夠安裝網路檔案系統 (NFS) 用戶端 (nfs-utils/nfs 通用) 的 Linux 系統。
- 用於上傳備份檔案的 S3 儲存貯體僅支援伺服器端加密 (SSE-S3/SSE-KMS)。
- Amazon S3 將備份文件的大小限制為 5 TB。如果備份文件超過 5 TB，則可以將其拆分為多個較小的文件。
- 上傳到 S3 儲存貯體的來源檔案數目不能超過一百萬個檔案。

- 此病毒碼僅支援 Percona XtraBackup 完整備份和增量備份。它不支援使用`--tables`、、、`--tables-exclude`、`--tables-file`、`--databases`、`--databases-exclude`或的部分備份`--databases-file`。
- Aurora 不會從來源 MySQL 資料庫還原使用者、函數、預存程序或時區資訊。

產品版本

- 來源資料庫必須是 MySQL 版本 5.5、5.6 或 5.7 版。
- 對於 MySQL 5.7, 你必須使用佩科納 XtraBackup 2.4.
- 對於 MySQL 5.6 和 5.6，您必須使用佩科納 XtraBackup 2.3 或 2.4。

架構

源, 技術, 堆棧

- 以 Linux 為基礎的作業系統
- MySQL 伺服器
- 佩爾科納 XtraBackup

目標技術堆疊

- Amazon Aurora
- Amazon S3
- Amazon EFS

目標架構

工具

AWS 服務

- Amazon Aurora 是全受管的關聯式資料庫引擎，可讓設定、操作和擴展 MySQL 部署變得簡單且符合成本效益。Aurora MySQL 是 MySQL 的一個嵌入式替代品。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

其他工具

- Percona XtraBackup 是一個開源實用程序，可以執行 MySQL 數據庫的流，壓縮和增量備份，而不會中斷或阻止數據庫。

史诗

建立 Amazon EFS 檔案系統

任務	描述	所需技能
建立與 Amazon EFS 掛接目標建立關聯的安全群組。	在 VPC 中建立透過 AWS 傳輸閘道連接現場部署資料庫的 VPN 附件設定的安全群組。如需本文及其他故事中所描述之命令和步驟的詳細資訊，請參閱此模式結尾 < 相關資源 > 一節中的連結。	AWS DevOps /資料庫管理員
編輯安全性群組規則。	使用 NFS 類型、連接埠 2049 和內部部署資料庫伺服器的 IP 範圍做為來源，新增輸入規則。根據預設，輸出規則會允許所有流量離開。如果不是這種情況，請新增輸出規則以開啟 NFS 連接埠的連線。再新增兩個輸入規則：連接埠 2049 (來源：此相同安全群組的安全群組 ID) 和連接埠 22 (來源：連線至 EC2 執行個體的 IP 範圍)。	AWS DevOps /資料庫管理員
建立檔案系統。	在裝載目標中，使用您在上一個故事中建立的 VPC 和安全性	AWS DevOps /資料庫管理員

任務	描述	所需技能
	群組。根據內部部署資料庫的 I/O 需求，選擇輸送量模式和效能。選擇性地啟用靜態加密。	

掛載檔案系統

任務	描述	所需技能
建立要與 EC2 執行個體關聯的 IAM 執行個體設定檔角色。	建立具有權限的 IAM 角色，以便在 Amazon S3 中上傳和存取物件。選擇將備份作為政策資源存放的 S3 儲存貯體。	AWS DevOps
建立 EC2 執行個體。	啟動 Linux 型 EC2 執行個體，並附加您在上一步中建立的 IAM 執行個體設定檔角色，以及您先前建立的安全群組。	AWS DevOps
安裝 NFS 用戶端。	在現場部署資料庫伺服器和 EC2 執行個體上安裝 NFS 用戶端。如需安裝指示，請參閱「其他資訊」一節。	DevOps
掛載 Amazon EFS 檔案系統。	在內部部署和 EC2 執行個體上掛載 Amazon EFS 檔案系統。在每部伺服器上，建立用於儲存備份的目錄，並使用掛載目標端點掛載檔案系統。如需範例，請參閱「其他資訊」一節。	DevOps

做一個 MySQL 源數據庫的備份

任務	描述	所需技能
安裝佩爾科納 XtraBackup。	在內部部署資料庫伺服器上安裝 Percona XtraBackup 2.3 或 2.4 (視您的 MySQL 資料庫版本而定)。如需安裝連結，請參閱「相關資源」一節。	資料庫管理員
計算來源資料庫中的結構描述和表格。	收集並記下來源 MySQL 資料庫中結構描述和物件的數目。移轉後，您將使用這些計數來驗證 Aurora MySQL 資料庫。	資料庫管理員
(選擇性) 記下來源資料庫中的最新二進位記錄序列。	如果您要在來源資料庫和 Aurora MySQL 之間建立二進位記錄複寫，以將停機時間降到最低，請執行此步驟。記錄資料匣必須啟用，而 server_id 必須是唯一的。請注意來源資料庫中目前的二進位記錄序列，就在起始備份之前。如果您打算僅使用完整備份，請在完整備份之前執行此步驟。如果您打算在完整備份後進行增量備份，請在將在 Aurora MySQL 資料庫執行個體上還原的最終增量備份之前執行此步驟。	資料庫管理員
啟動來源 MySQL 資料庫的完整備份。	使用佩科納 XtraBackup對 MySQL 源數據庫進行完整備份。如需完整備份和增量備份的指令範例，請參閱 < 其他資訊 > 一節。	資料庫管理員

任務	描述	所需技能
（可選）使用 Percona XtraBackup 進行增量備份。	增量備份可用於減少將來源資料庫與 Aurora MySQL 同步所需套用的二進位記錄數量。大型和交易繁重的資料庫可能會在備份期間產生大量的二進位記錄檔。透過進行增量備份並將其存放在共用 Amazon EFS 檔案系統上，您可以大幅縮短備份和上傳資料庫的時間。如需詳細資訊，請參閱「其他資訊」一節。繼續進行增量備份，直到您準備好開始向 Aurora 的遷移程序為止。	資料庫管理員
準備備份。	在此步驟中，交易記錄會套用至備份，以便在備份期間進行中的交易。繼續將交易記錄 (--apply-log-only) 套用至每個增量備份，以合併備份，上次備份除外。如需範例，請參閱「其他資訊」一節。<efs_mount_name>完成此步驟後，完整的合併備份將在~//full備份中。	資料庫管理員
壓縮並拆分最終合併的備份。	準備好最終的合併備份後，請使用 tar、zip 和 split 命令來建立較小的壓縮檔案。如需範例，請參閱「其他資訊」一節。	資料庫管理員

將備份還原至 Aurora MySQL 資料庫叢集

任務	描述	所需技能
將備份上傳到 Amazon S3。	儲存備份檔案的 Amazon EFS 檔案系統會同時掛載在現場部署資料庫和 EC2 執行個體上，因此 EC2 執行個體可立即使用備份檔案。<bucket_name> 使用安全殼層 (SSH) Connect 至 EC2 執行個體，並將壓縮的備份檔案上傳到新的或現有的 S3 儲存貯體；例如：aws s3 同步〜//完整備份 s3：//<efs_mount_name>/完整備份。如需其他詳細資訊，請參閱「相關資源」一節中的連結。	AWS DevOps
為 Aurora 建立服務角色以存取 Amazon S3。	建立具有信任「rds.amazonaws.com」的 IAM 角色，以及可讓 Aurora 存取儲存備份檔案所在的 S3 儲存貯體的政策。必要的權限為 ListBucket GetObject、和 GetObject Version。	AWS DevOps
建立 Aurora 的網路組態。	建立至少具有兩個可用區域的叢集資料庫子網路群組，以及允許輸出連線至來源資料庫的子網路表組態。建立允許對內部部署資料庫輸出連線的安全群組，並允許系統管理員連線至 Aurora DB 叢集。如需詳細資訊，請參閱 < 相關資源 > 一節中的連結。	AWS DevOps /資料庫管理員
將備份還原至 Aurora MySQL 資料庫叢集。	從您上傳到 Amazon S3 的備份中還原資料。指定來源資料	AWS DevOps /資料庫管理員

任務	描述	所需技能
	庫的 MySQL 版本、提供您上傳備份檔案的 S3 儲存貯體名稱和資料夾路徑前綴 (例如，「其他資訊」一節中的範例為「完整備份」)，並提供您建立的 IAM 角色來授權 Aurora 存取 Amazon S3。	
驗證 Aurora MySQL 資料庫。	根據您從來源資料庫取得的計數，驗證還原的 Aurora DB 叢集中結構描述和物件的計數。	資料庫管理員
設定備份記錄複寫。	在進行上次還原至 Aurora DB 叢集的備份之前，請使用先前所述的二進位記錄順序。在來源資料庫上建立複寫使用者，然後遵循 < 其他資訊 > 一節中的指示提供適當的權限、在 Aurora 上啟用複寫，以及確認複寫是否處於同步狀態。	AWS DevOps /資料庫管理員

相關資源

建立 Amazon EFS 檔案系統

- 建立安全群組 (Amazon VPC 文件)
- 交通閘道 VPN 附件 (Amazon VPC 文件)
- 使用 AWS 傳輸閘道 (聯網和內容交付部落格) 擴展 VPN 輸送量
- 建立 Amazon EFS 檔案系統 (Amazon EFS 文件)
- 建立掛接目標 (Amazon EFS 文件)
- 加密靜態資料 (Amazon EFS 文件)

掛載檔案系統

- Amazon EC2 的 IAM 角色 (Amazon EC2 文件)
- 啟動 Amazon EC2 Linux 執行個體 (Amazon EC2 文件)
- 安裝 NFS 用戶端 (Amazon EFS 文件)
- 在您的現場部署用戶端上掛載 Amazon EFS 檔案系統 (Amazon EFS 文件)
- 掛載 EFS 檔案系統 (Amazon EFS 文件)

對 MySQL 來源資料庫進行備份

- 安裝佩爾科納 XtraBackup 2.3（佩爾科納 XtraBackup 文檔）
- 安裝佩爾科納 XtraBackup 2.4（佩爾科納 XtraBackup 文檔）
- 設定複製主要組態 (MySQL 文件集)
- 將資料從外部 MySQL 資料庫遷移至 Aurora MySQL 資料庫叢集 (Aurora 文件)
- 增量備份（培爾科納 XtraBackup 文檔）

將備份恢復到 Amazon Aurora MySQL

- 建立儲存貯體 (Amazon S3 文件)
- 使用安全殼層連線到您的 Linux 執行個體 (Amazon EC2 文件)
- 設定 AWS CLI (AWS CLI 文件)
- 同步命令 (AWS CLI 命令參考)
- 建立 IAM 政策以存取 Amazon S3 資源 (Aurora 文件)
- 資料庫叢集必要條件 (Aurora 文件
- 使用資料庫子網路群組 (Aurora 文件)
- 為私有資料庫執行個體建立 VPC 安全群組 (Aurora 文件)
- 從 S3 儲存貯體還原 Aurora MySQL 資料庫叢集 (Aurora 文件)
- 使用 MySQL 或其他 Aurora 資料庫叢集設定複寫 (Aurora 文件)
- 外部主程序（Amazon RDS SQL 參考）
- 複製程序 (Amazon RDS 上的 SQL 參考)

其他參考

- 將資料從外部 MySQL 資料庫遷移至 Aurora MySQL 資料庫叢集 (Aurora 文件)

- MySQL 伺服器下載 (甲骨文網站)

教學課程和影片

- 使用 Amazon S3 將 MySQL 資料遷移到 Aurora MySQL 資料庫叢集 (AWS 知識中心)
- Amazon EFS 安裝和掛載 (影片)

其他資訊

安裝 NFS 用戶端

- 如果您使用的是 Red Hat 或類似的 Linux 作業系統，請使用以下指令：

```
$ sudo yum -y install nfs-utils
```

- 如果您使用的是 Ubuntu 或類似的 Linux 作業系統，請使用以下指令：

```
$ sudo apt-get -y install nfs-common
```

如需詳細資訊，請參閱 Amazon EFS 文件中的逐步解說。

掛載 Amazon EFS 檔案系統

使用命令：

```
mkdir ~/<efs_mount_name>
$ sudo mount -t nfs -o
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport mount-
target-IP:/ ~/<efs_mount_name>
```

如需詳細資訊，請參閱 Amazon EFS 文件中的逐步解說和裝載 EFS 檔案系統。

製作 MySQL 來源資料庫的備份

完整備份

使用類似以下的命令，該命令將其進行備份，壓縮並將其分割成每個 1 GB 的較小塊：

```
xtrabackup --backup --user=dbuser --password=<password> --binlog-info=AUTO --stream=tar
 --target-dir=~/<efs_mount_name>/fullbackup | gzip - | split -d --bytes=1024MB - ~/
<efs_mount_name>/fullbackup/backup.tar.gz &
```

如果您打算在完整備份之後進行後續的增量備份，請勿壓縮和分割備份。請改用類似下列的指令：

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/
<efs_mount_name>/fullbackup/
```

增量備份

使用`--incremental-basedir`參數的完整備份路徑，例如：

```
xtrabackup --backup --user=dbuser --password=<password> --target-dir=~/
<efs_mount_name>/incremental/backupdate --incremental-basedir=~/<efs_mount_name>/
fullbackup
```

其中基本目錄是完整備份和 xtrabackup_檢查點檔案的路徑。

如需有關進行備份的詳細資訊，請參閱 Aurora 文件中的將資料從外部 MySQL 資料庫移轉至亞馬遜 Aurora MySQL 資料庫叢集。

準備備份

若要準備完整備份：

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup
```

若要準備增量備份：

```
xtrabackup --prepare --apply-log-only --target-dir=~/<efs_mount_name>/fullbackup --
incremental-dir=~/<efs_mount_name>/incremental/06062020
```

若要準備最終備份：

```
xtrabackup --prepare --target-dir=~/<efs_mount_name>/fullbackup --incremental-dir=~/
<efs_mount_name>/incremental/06072020
```

如需詳細資訊，請參閱 Percona XtraBackup 文件中的增量備份。

壓縮和拆分合併的備份

<efs_mount_name>要壓縮〜//完整備份的合併備份：

```
tar -zcvf <backupfilename.tar.gz> ~/<efs_mount_name>/fullbackup
```

若要分割備份：

```
split -d -b1024M --verbose  <backupfilename.tar.gz> <backupfilename.tar.gz>
```

設定備份記錄複寫

在來源資料庫上建立複製使用者並提供適當的權限：

```
CREATE USER 'repl_user'@'' IDENTIFIED BY ''; GRANT REPLICATION CLIENT, REPLICATION
 SLAVE ON *.* TO 'repl_user'@'';
```

若要透過連線至 Aurora 資料庫叢集來啟用 Aurora 上的複寫，請在資料庫叢集參數群組中啟用二進位記錄。設置`binlog_format = mixed`（首選混合模式）。這項變更會要求您重新啟動執行個體以套用更新。

```
CALL mysql.rds_set_external_master ('sourcedbinstanceIP', sourcedbport, 'repl_user',
 '', 'binlog_file_name', binlog_file_position, 0); CALL mysql.rds_start_replication;
```

若要確認複製是否處於同步狀態：

```
SHOW Slave Status \G;
```

主要欄位後面的秒數顯示 Aurora 離內部部署資料庫的距離有多遠。

使用 AWS 應用程式容器將現場部署 Java 應用程式遷移到 AWS

資料來源：應用	目標：在 Amazon ECS 上部署的容器化應用程式	R 類型：重新平台
環境：PoC 或試點	技術：移轉；Web 和行動應用程式	工作負載：開源
AWS 服務：Amazon EC2 容器註冊表；Amazon ECS		

Summary

AWS App2Container (A2C) 是一種命令列工具，可協助將虛擬機器中執行的現有應用程式轉換為容器，而不需要變更任何程式碼。A2C 會探索伺服器上執行的應用程式、識別相依性，並產生相關成品，以便無縫部署至亞馬遜彈性容器服務 (Amazon ECS) 和亞馬遜彈性 Kubernetes 服務 (Amazon EKS)。

此模式提供透過工作者機器使用 App2Container，將應用程式伺服器上部署的現場部署 Java 應用程式遠端遷移到 AWS Fargate 或 Amazon EKS 的步驟。

工作機器可用於下列使用案例：

- 在執行 Java 應用程式的應用程式伺服器上，不允許或無法使用 Docker 安裝。
- 您必須管理部署在不同實體或虛擬伺服器上的多個應用程式的移轉作業。

先決條件和限制

先決條件

- 具有在 Linux 伺服器上執行之 Java 應用程式的應用程式伺服器
- 具有 Linux 作業系統的工作者電腦
- 具有至少 20 GB 可用磁碟空間的工作者電腦

限制

- 並非所有應用程式都受支援。如需詳細資訊，請參閱 Linux 支援的應用程式。

架構

源, 技術, 堆棧

- 在伺服器上執行的 Java 應用程式

目標技術堆疊

- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- Amazon Elastic Container Registry
- AWS Fargate

目標架構

工具

工具

- AWS App2Container — AWS App2Container (A2C) 是一種命令列工具，可協助您提升和轉移在現場部署資料中心或虛擬機器上執行的應用程式，以便在 Amazon ECS 或 Amazon EKS 管理的容器中執行。
- AWS CodeBuild — AWS CodeBuild 是雲端中的全受管建置服務。 CodeBuild 編譯您的原始程式碼、執行單元測試，並產生準備好部署的成品。
- AWS CodeCommit — AWS CodeCommit 是由 Amazon Web Services 託管的版本控制服務，您可以使用它在雲端私有存放和管理資產 (例如文件、原始碼和二進位檔案)。
- AWS CodePipeline — AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。
- Amazon ECS — 亞馬遜彈性容器服務 (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，用於在叢集上執行、停止和管理容器。

- Amazon ECR — 亞馬遜彈性容器註冊表 (Amazon ECR) 是一種 AWS 受管容器映像登錄服務，安全、可擴展且可靠。
- Amazon EKS — Amazon Elastic Kubernetes Service (Amazon EKS) 是一項受管服務，您可以使用它在 AWS 上執行 Kubernetes，而無需安裝、操作和維護自己的 Kubernetes 控制平面或節點。
- AWS Fargate — AWS Fargate 是一項技術，您可以與 Amazon ECS 搭配使用來執行容器，而不必管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的叢集。使用 Fargate，就不再需要佈建、設定或擴展虛擬機器的叢集來執行容器。

史诗

設定認證

任務	描述	所需技能
建立密碼以存取應用程式伺服器。	若要從工作者機器遠端存取應用程式伺服器，請在 AWS Secrets Manager 中建立密碼。針對您的密碼，您可以使用 SSH 私密金鑰或憑證和 SSH 私密金鑰。如需詳細資訊，請參閱管理 AWS App2Container 的密碼。	DevOps，開發人員

設置工人機器

任務	描述	所需技能
安裝 tar 檔案。	執行 `sudo yum install -y tar`。	DevOps，開發人員
安裝 AWS CLI。	若要安裝 Amazon Command Line Interface (AWS CLI) (AWS CLI)，請執行`curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.`	DevOps，開發人員

任務	描述	所需技能
	zip" -o "awscliv2 .zip" 。 解壓縮 awscliv2.zip 。 執行 sudo ./aws/ins tall 。	
安裝應用程序容器。	執行下列命令： curl -o AWSApp2Co ntainer-installer- linux.tar.gz https://app2contai ner-release-us-eas t-1.s3.us-east-1.a mazonaws.com/lates t/linux/AWSApp2Con tainer-installer-l inux.tar.gz sudo tar xvf AWSApp2Co ntainer-installer- linux.tar.gz sudo ./install.sh	DevOps，開發人員
設定設定檔。	若要設定 AWS 預設設定檔，請執行sudo aws configure 。 若要設定具名的 AWS 預設設定檔，請執行sudo aws configure --profile <profile name>。	DevOps，開發人員

任務	描述	所需技能
安裝 Docker.	執行下列命令。 `sudo yum install -y docker` `sudo systemctl enable docker & sudo systemctl restart docker`	
初始化應用程序容器。	若要初始化 App2Container，您需要下列資訊： • `workspace` ：儲存應用程式容器化成品。建議您提供至少具有 20 GB 可用磁碟空間的目錄路徑。 • `awsProfile` ：在伺服器上設定的 AWS 設定檔。這需要將成品上傳到 Amazon S3、執行`containerize` 命令，以及產生 AWS 成品，以便在 Amazon ECS 或 Amazon EKS 上部署。 • `s3Bucket`：擷取和存放 AWS 成品。 • `metricsReportPermission` ：收集和存儲報告的指標。 • `dockerContentTrust` ：簽署碼頭圖像。 執行 `sudo app2container init`。	DevOps，開發人員

設定工作者電腦

任務	描述	所需技能
將工作者電腦設定為遠端連線並在應用程式伺服器上執行 App2Container 命令。	若要配置 Worker 電腦，需要下列資訊： • `Server FQDN`：應用程式伺服器的完整網域名稱。 • `Server IP address`：應用程式伺服器的 IP 位址。FQDN 或 IP 位址就足夠了。 • `SecretARN` ：用來連線到應用程式伺服器並儲存在 Secrets Manager 中的密碼的 Amazon 資源名稱 (ARN)。 • `AuthMethod` ：`key`或`cert`驗證方法。 執行 `sudo app2container remote configure` 。	DevOps，開發人員

探索、分析和擷取工作者電腦上的應用程式

任務	描述	所需技能
探索內部部署 Java 應用程式。	若要遠端探索應用程式伺服器上執行的所有應用程式，請執行下列命令。 `sudo app2container remote inventory --target <FQDN/IP of App server>`	開發人員， DevOps

任務	描述	所需技能
	此命令會在中產生已部署應用程式的清單inventory.json 。	
分析發現的應用程序。	若要使用在詳細目錄階段取得的應用程式識別碼，從遠端分析每個應用程式，請執行下列命令。 `sudo app2container remote analyze --application-id <java-app-id> --target <FQDN/IP of App Server>` 這會在工作區位置產生analysis.json 檔案。產生此檔案之後，您可以根據需要變更容器化參數。	開發人員， DevOps
擷取已分析的應用程式。	若要為分析的應用程式產生應用程式歸檔，請從遠端執行下列命令，這會在工作區位置產生 tar 套裝軟體。 `sudo app2container remote extract --application-id <application id> --target <FQDN/IP of App Server>` 可在本機 Worker 電腦上產生擷取的加工品。	開發人員， DevOps

將工作者機器上擷取的成品容器化

任務	描述	所需技能
容器化擷取的成品。	執行下列命令，將上一個步驟中擷取的成品容器化。 `sudo app2container containerize --input-archive <tar bundle location on worker machine>`	開發人員， DevOps
完成目標。	要完成目標，請打開 `deployment.json` ，該目標在`containerize` 命令運行時創建。若要指定 AWS 遠端蓋特做為目標，請`createEcsArtifacts` 將`true`設定為。要將 Amazon EKS 設置為目標，`createEksArtifacts` 請設置為 true。	開發人員， DevOps

產生和佈建 AWS 成品

任務	描述	所需技能
在工作者機器上產生 AWS 部署成品。	若要產生部署人工因素，請執行下列命令。 `sudo app2container generate app-deployment --application-id <application id>`	DevOps

任務	描述	所需技能
	這會在工作區中產生 `ecs-master.yml` AWS CloudFormation 範本。	
佈建人工因素。	若要進一步佈建產生的成品，請執行下列命令部署 AWS CloudFormation 範本。 `aws cloudformation deploy --template-file <path to ecs-master.yml> --capabilities CAPABILITY_NAMED_IAM --stack-name <application id>-ECS`	DevOps
產生配管。	修改`pipeline.json`，這是在前面的故事中創建的，根據您的需求。然後執行`generate pipeline`命令以產生管線部署成品。	DevOps

相關資源

- 什麼是應用程序容器？
- AWS 應用程式容器部落格文章
- AWS CLI 組態基礎知識
- Amazon ECS 碼頭基礎知識
- 泊塢視窗命令

在 AWS 大型遷移中遷移共用檔案系統

由阿米特魯德拉茹（AWS），薩姆阿帕（AWS），白姆斯瓦拉勞（AWS），沃利魯（AWS）和普拉卡薩姆桑吉耶夫（AWS）創建

環境：生產	來源：內部部署共用檔案系統	目標：Amazon EFS 或 Amazon FSx
R 類型：重新平台	工作負載：所有其他工作	技術：移轉、儲存與備份
AWS 服務：AWS DataSync; Amazon EFS; Amazon FSx for Windows File Server; Amazon FSX 適用於 ONTAP NetApp		

Summary

遷移 300 台以上的伺服器被視為是一項大型遷移。大型移轉的目的是將工作負載從現有的現場部署資料中心遷移到 AWS 雲端，而這些專案通常著重於應用程式和資料庫工作負載。不過，共用檔案系統需要集中注意力和個別的移轉計劃。此模式描述了共用檔案系統的移轉程序，並提供最佳作法，以便在大型移轉專案中成功移轉這些檔案系統。

共用檔案系統 (SFS) 又稱為網路或叢集檔案系統，是掛載至多部伺服器的檔案共用。共用檔案系統可透過網路檔案系統 (NFS)、通用網際網路檔案系統 (CIFS) 或伺服器訊息區 (SMB) 等通訊協定存取。

這些系統不會使用標準遷移工具 (例如 AWS 應用程式遷移服務) 進行遷移，因為它們既不專用於要遷移的主機，也不會以區塊裝置的形式表示。雖然大多數主機相依性都是透通移轉的，但是必須個別處理相依檔案系統的協調和管理。

您可以在下列階段移轉共用檔案系統：探索、規劃、準備、切除和驗證。使用此模式和附加的工作簿，您可以將共用檔案系統遷移到 AWS 儲存服務，例如 Amazon Elastic File System (Amazon EFS)、適用於 NetApp ONTAP 的 Amazon FSx 或 Windows 檔案伺服器的 Amazon FSx。若要傳輸檔案系統，您可以使用 AWS DataSync 或第三方工具，例如 NetApp SnapMirror.

注意：此模式是 AWS Prescriptive Guidance 系列的一部分，內容涉及大量遷移至 AWS 雲端。此模式包括將 SFSS 納入伺服器 Wave 計劃的最佳做法和指示。如果您要在大型遷移專案之外遷移一

或多個共用檔案系統，請參閱 Amazon EFS 的 AWS 文件、Windows 檔案伺服器專用 AmazonFSx 和適用於 ONTAP 的 Amazon FSx 文件中的資料傳輸指示。 NetApp

先決條件和限制

先決條件

必要條件可能會因來源和目標共用檔案系統以及您的使用案例而有所不同。以下是最常見的：

- 作用中的 AWS 帳戶
- 您已完成大型移轉專案的應用程式組合探索，並開始開發 Wave 計劃。如需詳細資訊，請參閱 AWS 大型移轉的產品組合教戰手冊。
- 允許現場部署資料中心和 AWS 環境之間的輸入和輸出流量的虛擬私有雲端 (VPC) 和安全群組。如需詳細資訊，請參閱網路到 Amazon VPC 連線選項和 AWS DataSync 網路需求。
- 建立 AWS CloudFormation 堆疊或許可以建立 Amazon EFS 或 Amazon FSx 資源的許可。如需詳細資訊，請參閱CloudFormation 文件、Amazon EFS 文件或 Amazon FSx 文件。
- 如果您使用 AWS 執 DataSync 行遷移，則需要下列許可：
  - AWS DataSync 將日誌傳送到 AWS 日誌日 CloudWatch 誌群組的許可。如需詳細資訊，請參閱允許 DataSync 將記錄檔上傳至記 CloudWatch 錄群組。
  - 存取記 CloudWatch 錄檔記錄群組的權限。如需詳細資訊，請參閱管理 CloudWatch 記錄資源存取權限概觀。
  - 在中建立代理程式和工作的權限 DataSync。如需詳細資訊，請參閱使用 AWS 所需的 IAM 許可 DataSync。

限制

- 此病毒碼旨在將 SFSS 移轉為大型移轉專案的一部分。其中包含將 SFSS 納入您的 Wave 計畫以移轉應用程式的最佳做法和指示。如果您要在大型遷移專案之外遷移一或多個共用檔案系統，請參閱 Amazon EFS 的 AWS 文件、Windows 檔案伺服器專用 AmazonFSx 和適用於 ONTAP 的 Amazon FSx 文件中的資料傳輸指示。 NetApp
- 此模式是以常用的架構、服務和遷移模式為基礎。不過，大型移轉專案和策略可能會因組織而異。您可能需要根據您的需求自訂此解決方案或提供的工作簿。

架構

源, 技術, 堆棧

下列一或多項：

- 檔案伺服器
- 視窗 (SMB) 檔案伺服器
- NetApp 儲存陣列
- 戴爾 EMC 儲存陣列

目標技術堆疊

下列一或多項：

- Amazon Elastic File System
- Amazon FSx NetApp
- Amazon FSx for Windows File Server

目標架構

該圖顯示了以下過程：

1. 您可以使用 AWS AWS Direct Connect 連接或 AWS Site-to-Site VPN 等 AWS 服務，在現場部署資料中心和 AWS 雲端之間建立連接。
2. 您可以在內部部署資料中心安裝 DataSync 代理程式。
3. 根據您的 wave 計劃，您可 DataSync 以用來將資料從來源共用檔案系統複寫到目標 AWS 檔案共用。

移轉階段

下圖顯示在大型移轉專案中移轉 SFS 的階段和高階步驟。

此模式的 Epics 部分包含有關如何完成遷移和使用附加工作簿的詳細說明。以下是此階段化方法中步驟的高階概觀。

Phase (階段)	步驟
探索	1. 您可以使用探索工具收集共用檔案系統的相關資料，包括伺服器、掛載點和 IP 位址。
	2. 使用組態管理資料庫 (CMDB) 或移轉工具，您可以收集伺服器的詳細資料，包括移轉波、環境、應用程式擁有者、IT 服務管理 (ITSM) 服務名稱、組織單位和應用程式識別碼的相關資訊。
計畫	3. 使用收集的有關 SFSS 和伺服器的資訊，建立 SFS 波浪計劃。
	4. 使用建置工作表中的資訊，為每個 SFS 選擇目標 AWS 服務和遷移工具。
準備	5. 在 Amazon EFS、用於 NetApp ONTAP 的 Amazon FSx 或 Amazon FSx 中設置目標基礎設施。
	6. 設定資料傳輸服務，例如 DataSync，然後啟動初始資料同步。初始同步完成後，您可以將重複發生的同步設定為按排程執行，
	7. 使用目標檔案共用的相關資訊 (例如 IP 位址或路徑) 更新 SFS 波形計劃。
切過	8. 停止主動存取來源 SFS 的應用程式。
	9. 在資料傳輸服務中，執行最終資料同步。
	10. 同步完成後，請檢閱記錄檔中的 CloudWatch 記錄資料，以驗證是否完全成功。
驗證	11. 在伺服器上，將掛載點變更為新的 SFS 路徑。

12. 重新啟動並驗證應用程式。

工具

AWS 服務

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- AWS DataSync 是一種線上資料傳輸和探索服務，可協助您在 AWS 儲存服務之間移動檔案或物件資料。
- Amazon Elastic File System (Amazon EFS) 可協助您在 AWS 雲端中建立和設定共用檔案系統。
- Amazon FSx 提供支援業界標準連線協定的檔案系統，並在 AWS 區域提供高可用性和複寫功能。

其他工具

- SnapMirror是一種 NetApp 資料複製工具，可將指定的來源磁碟區或 q 樹狀結構中的資料分別複製到目標磁碟區或 q 樹狀結構。您可以使用此工具將 NetApp 來源檔案系統遷移到適用於 ONTAP 的 Amazon FSx。
- Robocopy，這是簡稱強大的文件複製，是一個命令行目錄和命令的 Windows 命令。您可以使用此工具將視窗來源檔案系統遷移到 Amazon FSx for Windows File Server)。

最佳實務

波浪規劃方法

規劃大型移轉專案的浪潮時，請考慮延遲和應用程式效能。當 SFS 和相依應用程式在不同的位置 (例如雲端和內部部署資料中心) 運作時，這可能會增加延遲並影響應用程式效能。以下是建立波動計劃時的可用選項：

1. 在同一波內移轉 SFS 和所有相依伺服器 — 此方法可防止效能問題並將重工減至最少，例如多次重新配置掛載點。當應用程式和 SFS 之間需要極低的延遲時，建議使用此選項。但是，波浪規劃很複雜，目標通常是從依賴組中刪除變量，而不是添加到它們中。此外，如果許多伺服器存取相同的 SFS，則不建議使用此方法，因為這會使波形過大。
2. 在移轉最後一個相依伺服器之後移轉 SFS — 例如，如果 SFS 由多部伺服器存取，且這些伺服器已排定在第 4、6 和第 7 波中移轉，請排定 SFS 在第 7 波中進行移轉。

對於大型移轉而言，這種方法通常是最合乎邏輯的方法，建議用於對延遲敏感的應用程式。它降低了與數據傳輸相關的成本。它也會將 SFS 與更高層 (例如生產) 應用程式之間的延遲時間降到最低，因為較高層級的應用程式通常會排定在開發和 QA 應用程式之後最後移轉。

但是，這種方法仍然需要探索、規劃和敏捷性。您可能需要在較早的波動中遷移 SFS。確認應用程式可以在第一個相依波和包含 SFS 的波之間的時間內承受額外的延遲。與應用程式擁有者進行探索工作階段，並在最延遲敏感的應用程式中，在同一波移轉應用程式。如果在移轉相依應用程式後發現效能問題，請準備好快速進行樞紐以盡快移轉 SFS。

3. 在大型移轉專案結束時移轉 SFS — 如果延遲不是因素 (例如 SFS 中的資料不常存取或對應用程式效能不重要)，建議使用此方法。此方法可簡化移轉作業，並簡化切換工作。

您可以根據應用程式的延遲敏感度來混合這些方法。例如，您可以使用方法 1 或 2 移轉對延遲敏感的 SFSS，然後使用方法 3 移轉 SFSS 的其餘部分。

選擇 AWS 檔案系統服務

AWS 為檔案儲存提供數種雲端服務。每一種都提供不同的效能、規模、可存取性、整合、合規性和成本最佳化方面的優點和限制。有一些邏輯默認選項。例如，如果您目前的現場部署檔案系統正在操作 Windows 伺服器，則預設選擇適用於 Windows 檔案伺服器的 Amazon FSx。或者，如果現場部署檔案系統正在操作 NetApp ONTAP，那麼適用於 NetApp ONTAP 的 Amazon FSx 就是預設選擇。不過，您可以根據應用程式的需求來選擇目標服務，或實現其他雲端作業優勢。如需詳細資訊，請參閱為您的部署選擇合適的 AWS 檔案儲存服務 (AWS 高峰會簡報)。

選擇移轉工具

Amazon EFS 和 Amazon FSx 支援使用 AWS DataSync 將共享檔案系統遷移到 AWS 雲端。如需支援的儲存系統和服務、權益和使用案例的詳細資訊，請參閱什麼是 AWS DataSync。如需使用 DataSync 傳輸檔案的程序概觀，請參閱 AWS DataSync 傳輸的運作方式。

還有幾種可用的第三方工具，包括以下內容：

- 如果您選擇適用於 NetApp ONTAP 的 Amazon FSx，您可以使用 NetApp SnapMirror 將檔案從現場部署資料中心遷移到雲端。 SnapMirror 使用區塊層級複製，這可能比 DataSync 資料傳輸程序更快，也可以縮短資料傳輸程序的持續時間。如需詳細資訊，請參閱移轉至 FSx 以供 ONTAP 使用。NetApp SnapMirror
- 如果您選擇適用於 FSx for Windows File Server 的 Amazon FSx，您可以使用 Robocopy 將檔案遷移到雲端。如需詳細資訊，請參閱使用 Robocopy 將現有檔案移轉至 FSx (適用於 Windows 檔案伺服器)。

史诗

探索

任務	描述	所需技能
準備 SFS 探索活頁簿。	1. 在此模式的「附件」部分中下載工作簿。這包含兩個文件，SFS-Discovery-Workbook.xlsx 和 SFS-Wave-Plan-Workbook.xlsx。 2. 在 Microsoft Excel 中打開 SFS 發現工作簿文件。 3. 在 [儀表板] 工作表上，執行下列動作： • 在欄 A 中，更新環境名稱。 • 在 B 欄中，更新環境的順序，使其按最低 (1) 優先順序排列到最高優先順序。 • 在資料行 D-E 中，更新波浪排程。 • 在欄 C 和 K 欄中，更新 AWS 帳戶名稱。 • 在欄 L 中，更新虛擬私人 VPC 識別碼。 • 在 M—O 欄中，更新子網路 ID。 4. 檢閱活頁簿範本的其餘部分，並更新組織或使用案例所需的任何其他值。 5. 保存工作簿。	移民工程師，移民負責人

任務	描述	所需技能
收集來源 SFS 的相關資訊。	1. 使用您偏好的探索工具，識別所有適用儲存裝置、Linux 伺服器和 Windows 伺服器上的所有 SFS 掛載。通常情況下，您需要收集以下信息： • 用戶端裝置 • 用戶端 IP 位址 • SFS 詳細資料 • 掛載點 備註：您可以將掛載點詳細資料新增至移轉 Runbook，以便在移轉後重新掛載 SFS。 2. 打開 SFS 發現活頁簿文件。 3. 在「波浪表」工作表上，執行下列操作： • 在 [伺服器位置 (D)] 欄的公式中，確認內部部署來源的 CIDR 範圍格式適用於您的範圍。例如，如果您的 CIDR 範圍是`10.0.0.0/8` ，請輸入`10.*.*.*`。 • 在 SFS 位置 (E) 欄的公式中，確認目標 VPC 的 CIDR 範圍格式適用於您的範圍。例如，如果您的 CIDR 範圍是`176.16.0.0/16` ，請輸入`176.16.*.*` 。	移民工程師，移民負責人

任務	描述	所需技能
	4. 在 SFS 資料工作表上，執行下列動作： • 在伺服器名稱 (A) 欄中，輸入掛載 SFS 的伺服器名稱。 • 在 SFS 路徑 (B) 欄中，輸入 SFS 的名稱。 • 在 IP 位址 (C) 欄中，輸入伺服器的 IP 位址。 • 新增您在探查期間收集的任何其他相關資訊，例如掛載點和 SFS 大小。您可以稍後使用此資料來修改波浪計劃計算。 5. 保存工作簿。	

任務	描述	所需技能
收集有關服務器的信息。	1. 使用 CMDB 或移轉工具中記錄的資料，識別有關具有 SFS 掛載之伺服器的下列所有資訊： • 伺服器名稱 • IP 地址 • 波 • 組織單位 (OU) • 伺服器環境DEV，例如QA、或 PROD • 應用程式名稱 • 應用程式擁有者和連絡人 2. 打開 SFS 發現活頁簿文件。 3. 在 [伺服器資料] 工作表的欄 A—H 中，輸入您收集的有關來源伺服器的資訊。注意下列事項： • 在 Wave # (C) 欄中，輸入波浪名稱 (例如Wave1)、 out-of-scope (OOS) 或Retire。 • 如果應用程序所有者聯繫人（H）列，請驗證電子郵件地址是否正確。系統會根據您在「應用程式擁有者」(G) 欄中提供的名稱自動產生此電子郵件地址。如有必要，請手動更新值以反映正確的電子郵件地址。 • 請勿修改包含公式的 I 至 J 欄。	移民工程師，移民負責人

任務	描述	所需技能
	4. 保存工作簿。	

計畫

任務	描述	所需技能
建立 SFS 波浪計劃。	1. 打開 SFS 發現活頁簿文件。 2. 驗證在探索階段收集的所有資訊都是準確且最新的。 3. 在「波表」工作表上，篩選值上的 SFS 波 (K) 欄。1這是第一波中所有 SFSS 的清單。 注意：此資料欄0中的值表示 SFS 超出移轉範圍。這可能是因為 SFS 已經在 AWS 上託管，或者存取共用的伺服器超出了遷移範圍。 4. 確認您要在此波中移轉這些 SFSS。如需如何將 SFSS 指派給波形的詳細資訊，請參閱「最佳作法」一節中的 Wave 規劃方法。 5. 選擇並複製包含過濾值的單元格。請勿複製包含欄標題的標頭列。 6. 開啟先前下載的 SFS-Wave 計畫-活頁簿檔案。 7. 在「從搜尋匯出」工作表上，選取儲存格 A2。 8. 粘貼複製的數據。	建立領導者，切換負責人，遷移工程師，遷移負責人

任務	描述	所需技能
	9. 保存 SFS 發現工作簿和 SFS 波計劃-工作簿文件。	

任務	描述	所需技能
選擇目標 AWS 服務和遷移工具。	1. 在 SFS-Wave 計劃-活頁簿檔案的 [從探索匯出] 工作表上，選取並複製 [舊路徑 (C)] 欄中的值。 2. 在「建置波形」工作表上，選取儲存格 A2。 3. 粘貼複製的數據。此工作表中的欄 B—M 會自動更新，以反映與此路徑相關聯的其他資料。 4. 刪除列 A 中的所有重複值。如需指示，請參閱移除重複的值 (Microsoft Support 網站)。 5. 在 Target 模式或服務 (F) 欄中，檢閱建議的目標 AWS 服務，並視需要進行更新。如需詳細資訊，請參閱此模式的最佳實務一節中的選擇 AWS 檔案系統服務。 6. 在移轉方法 (G) 欄中，檢閱建議的移轉工具，並視需要進行更新。如需詳細資訊，請參閱此模式的最佳作法一節中的選擇移轉工具。 7. 保存 SFS 發現工作簿文件。您已完成為這個浪潮的建立波浪計劃。 8. 重複這些說明，為每個波準備一個波浪計劃。由於波浪計劃在移轉期間可能會有所變更，因此我們建議您事先規劃不超過 5 個波浪。	移民工程師，移民負責人

準備

任務	描述	所需技能
設定目標檔案系統。	根據波形計劃中記錄的詳細資訊，在目標 AWS 帳戶、VPC 和子網路中設定目標檔案系統。如需指示，請參閱下列 AWS 文件： • Amazon EFS • Amazon FSx NetApp • Amazon FSx for Windows File Server	遷移工程師，遷移主管，AWS 管理員
設定移轉工具並傳輸資料。	1. 如果您使用 AWS DataSync，請設定 DataSync 任務的記錄。如需指示，請參閱記錄 AWS DataSync 任務活動。 2. 設定移轉工具並根據所選工具的指示執行初始資料傳輸： • 對於 Amazon EFS，請參閱以下內容： • 使用 AWS 將檔案傳輸到 Amazon EFS DataSync • 如需適用於 ONTAP 的 Amazon FSx，請參閱下列內容： • 使用移轉至 FSx 以進行啟動時使用 NetApp SnapMirror	AWS 管理員、雲端管理員、移轉工程師、移轉主管

任務	描述	所需技能
	• 使用 AWS 移轉至 FSx 以進行 ONTAP DataSync • 如需 FSx for Windows File Server 的 Amazon FSx，請參閱下列內容： • 使用 AWS 將現有檔案遷移到 FSx for Windows File Server) DataSync • 使用機器人複製將現有檔案移轉至 FSx 3. 在初始傳輸期間或之後，可能會變更來源 SFS。設定來源與目標檔案系統之間的週期性資料傳輸，以保持資料同步： • 如果您使用的是 DataSync，請參閱排程 AWS DataSync 任務。DataSync 只會傳輸來源 SFS 中已修改的檔案或新檔案。 • 如果您使用的是第三方工具，請參閱所選工具的文件。	

任務	描述	所需技能
更新波浪計劃。	1. 打開當前波的 SFS 波計劃-工作簿文件。 2. 在 [建置-Wave] 工作表的 [新路徑 IP 位址 (N)] 欄中，輸入目標檔案系統的 IP 位址。執行下列其中一項動作來尋找 IP 位址： • 對 FSx for Windows File Server，請在 Amazon FSx 主控台上選擇 [檔案系統]，選擇您的檔案系統，然後檢視 [網路與安全性] 區段。 • 如需適用於 ONTAP 的 FSx，請參閱掛接磁碟區。 • 對於 Amazon EFS，請參閱使用 IP 地址進行掛接。 3. 在新路徑 (O) 欄中，輸入新的裝載路徑。掛載路徑是檔案系統的 DNS 名稱。執行下列其中一項作業以找出裝載路徑： • 對 FSx for Windows File Server，請在 Amazon FSx 主控台上選擇 [檔案系統]，選擇您的檔案系統，然後選擇 [附加]。 • 如需 ONTAP 的 FSx，請參閱檔案系統詳細資訊頁面。如需指示，請參閱掛接磁碟區。	移民工程師，移民負責人

任務	描述	所需技能
	• 對於 Amazon EFS，請參閱收集資訊。 4. 在 [重新掛載摘要] 工作表上，確認 [新路徑 (C)] 和 [新路徑 IP 位址 (D)] 欄會反映更新的值。 5. 確認您的組織已準備好執行手冊，以便在切換後重新掛載 Linux 和 Windows 檔案系統。如需一般指示，請參閱下列內容： • 掛載 EFS 檔案系統 • 存取 FSx 的 FSx for Windows File Server 檔案共用 • 安裝適用於 ONTAP 磁碟區的 FSx 6. 如果此波形中未包含任何相依伺服器，請將它們記錄在 [應用程式小組- 通訊] 工作表上。通知相應的應用程序或服務器所有者，因為它們可能不包含在標準波形通信中。 7. 如果在完成波浪計劃後從波浪中移除 SFSS，請在「已遮擋」工作表上追蹤這些項目。	

切過

任務	描述	所需技能
停止應用程式。	如果應用程式或用戶端主動在來源 SFS 中執行讀取和寫入作業，請先停止它們，然後再執行最終資料同步。如需指示，請參閱應用程式文件或停止讀取和寫入活動的內部程序。例如，請參閱啟動或停止網頁伺服器 (IIS 8) (Microsoft 文件)或使用 systemctl 管理系統服務 (Red Hat 文件)。	應用程式擁有者、應用程
執行最終資料傳輸。	1. 在移轉工具中，手動執行最終資料傳輸工作或工作，以將目標檔案系統與來源 SFS 同步。如需指示，請參閱啟動工 DataSync 作或參閱所選協力廠商移轉工具的說明文件。 2. 等待資料傳輸任務完成。如需詳細資訊，請參閱 AWS 使用 Amazon 監控 AWS DataSync 活動 CloudWatch和透過命令列監控您的 DataSync 任務。	移民工程師，移民負責人
驗證資料傳輸。	如果您使用 AWS DataSync，請執行以下操作以驗證成功完成的最終資料傳輸： 1. 在 AWS DataSync 主控台中，記下任務和執行 ID，例如`task-0000-exec-1111` 。	移民工程師，移民負責人

任務	描述	所需技能
	2. 導覽至工作的「工作記錄」區段。 DataSync 3. 選擇記CloudWatch 錄群組連結。 4. 在記錄檔中，搜尋工作和執行 ID。 5. 記下任何傳輸錯誤。如需詳細資訊，請參閱 DataSync 文件中的常見錯誤。 6. 驗證下列項目： • 比較來源和目標 SFSS 中的檔案清單，以確認所有資料都已傳輸 • 比較來源和目標 SFSS 之間的檔案存取權限。 如果您使用的是第三方工具，請參閱所選移轉工具文件中的資料傳輸驗證指示。	

驗證

任務	描述	所需技能
重新掛載檔案系統並驗證應用程式的功能和效能。	1. 如果在此波形中移轉相依伺服器，請在 SFS-Wave 計劃-活頁簿檔案的 [重新掛載摘要] 工作表上，在 [新伺服器 IP 位址 (F)] 欄中輸入伺服器的新 IP 位址。 2. 在所有伺服器上，將檔案系統的掛載點從舊路徑更新到新路徑。使用您組織的	AWS 系統管理員、應用程式擁有

任務	描述	所需技能
	runbook 重新裝載先前在準備階段討論。 3. 檢查掛載並確認檔案是否存在，確認檔案系統已正確掛載並可存取。基礎結構小組通常會執行這些活動。 4. 重新啟動應用程式，並讓應用程式擁有者或 QA 團隊視應用程式需要完成應用程式的功能和效能測試。	

故障診斷

問題	解決方案
Microsoft Excel 中的單元格值不會更新。	拖曳填色控點，複製範例列中的公式。如需詳細資訊，請參閱 Windows 或 Mac 版的指示 (Microsoft Support 網站)。

相關資源

AWS 文件

- AWS DataSync 文件
- Amazon EFS 文件
- Amazon FSx 文件
- 大量遷移到 AWS 雲端
  - AWS 大型移轉指南
  - AWS 大型移轉的產品組合教戰手冊

疑難排解

- AWS DataSync 問題疑難排解

- Amazon EFS 故障
- FSx for Windows File Server 專用的 Amazon FSx 疑難
- 針 NetApp 對 ONTAP 的 Amazon FSx 疑難排解

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS

由深谷 (AWS) 和白芝謝克 (AWS) 創建

環境：PoC 或試點	來源：Oracle 資料庫 (內部部署或 EC2 執行個體)	目標：Amazon RDS for Oracle
R 類型：重新平台	工作量：甲骨文	技術：移轉、分析、資料庫
AWS 服務：Amazon RDS		

Summary

Oracle GoldenGate 是適用於異質資料庫和 IT 環境的即時資料擷取和複寫服務。但是，這項服務目前不支援適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS)。如需支援的資料庫清單，請參閱異質資料庫的 Oracle (Oracle 說明文件)。 GoldenGate 此模式說明如何使用 Oracle GoldenGate 和 Oracle GoldenGate 平面檔案配接器從來源 Oracle 資料庫產生平面檔案，這些資料庫可以位於現場部署或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上。然後，您可以將這些純資料檔匯入 Amazon RDS for Oracle 料庫執行個體。

在此模式中，您可 GoldenGate 以使用 Oracle 從來源 Oracle 資料庫擷取軌跡檔案。資料泵浦會將追蹤檔案複製到整合伺服器 (即 EC2 執行個體)。在整合伺服器上，Oracle GoldenGate 會根據追蹤檔案的摘錄資料擷取，使用純資料轉接器來產生一系列循序純資料檔。Oracle 會將資料 GoldenGate 格式化為分隔符號分隔值或長度分隔值。然後，您可以使用 Oracle SQL*Loader 將純資料檔匯入目標 Amazon RDS for Oracle 資料庫執行個體的目標。

目標受眾

這種模式適用於那些具有 Oracle GoldenGate 基本構建組塊經驗和知識的人。如需詳細資訊，請參閱 Oracle GoldenGate 架構概要 (Oracle 說明文件)。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶。
- 甲骨文 GoldenGate 許可證。
- Oracle GoldenGate 介面卡的個別授權。

- 在現場部署或 EC2 執行個體上執行的來源 Oracle 資料庫。
- 做為整合伺服器使用的 EC2 Linux 執行個體。如需詳細資訊，請參閱開始使用 Amazon EC2 Linux 執行個體 (Amazon EC2 文件)。
- 適用於 Oracle 資料庫執行個體的目標亞馬遜 RDS。如需詳細資訊，請參閱建立 Oracle 資料庫執行個體 (Amazon RDS 說明文件)。

產品版本

- 甲骨文資料庫企業版 10g、11 克、12c 或更新版本
- 甲骨 GoldenGate 文 12.2.0.1.1 或更新版本

架構

源, 技術, 堆棧

Oracle 資料庫 (在內部部署或 EC2 執行個體上)

目標技術堆疊

Amazon RDS for Oracle

來源與目標架構

1. Oracle 會從來源資料庫記錄中 GoldenGate 擷取追蹤。
2. 資料汲取記錄並將其移轉至整合伺服器。
3. Oracle GoldenGate 平面檔案轉接器會讀取追蹤、來源定義和擷取參數。
4. 您結束萃取，該萃取會產生控制檔和平面資料檔。
5. 您可以將一般資料檔案遷移到 AWS 雲端中的 Amazon RDS for Oracle 文資料庫執行個體。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

- 適用於甲骨文的 Amazon Relational Database Service 服務 (Amazon RDS) 可協助您在 AWS 雲端中設定、操作和擴展 Oracle 關聯式資料庫。

其他服務

- Oracle GoldenGate 是一項服務，可協助您將資料從一個資料庫複製、篩選及轉換到另一個異質資料庫或其他目標拓樸 (例如純資料檔)。
- Oracle GoldenGate 應用程式轉接器可讓 Oracle GoldenGate 從來源資料庫的軌跡檔中擷取的異動性資料，產生一系列循序純資料檔與控制檔。這些配接卡廣泛用於資料倉儲應用程式和專有或舊版應用程式中的擷取、轉換和載入 (ETL) 作業。Oracle GoldenGate 會執行此擷取，並在異質資料庫、平台和作業系統之間以近乎即時的速度套用此擷取。轉接器支援輸出檔案的不同格式，例如 CSV 或 Apache 實木地板。您可以載入這些產生的檔案，以便將資料載入到不同的異質資料庫。

史诗

在來源資料庫伺服器 GoldenGate 上設定 Oracle

任務	描述	所需技能
下載甲骨文 GoldenGate。	在來源資料庫伺服器上，下載 Oracle 12.2.0.1.1 或更 GoldenGate 新版本。如需相關指示，請參閱下載 Oracle GoldenGate (Oracle 說明文件)。	DBA
安裝甲骨文 GoldenGate。	如需相關指示，請參閱安裝 Oracle GoldenGate (Oracle 說明文件)。	DBA
設定甲骨文 GoldenGate。	如需指示，請參閱為 Oracle 準備資料庫 GoldenGate (Oracle 說明文件)。	DBA

在整合伺服器 GoldenGate 上設定 Oracle

任務	描述	所需技能
下載甲骨文 GoldenGate。	在整合伺服器上，下載甲骨文 GoldenGate 12.2.0.1.1 或更新版本。如需相關指示，請參閱下載 Oracle GoldenGate (Oracle 說明文件)。	DBA
安裝甲骨文 GoldenGate。	建立目錄、設定管理程序，以及針對異質環境建立defgen檔案。如需相關指示，請參閱安裝 Oracle GoldenGate (Oracle 說明文件)。	DBA

變更 Oracle GoldenGate 資料擷取組態

任務	描述	所需技能
準備 Oracle GoldenGate 轉接器。	在整合伺服器上，設定 Oracle GoldenGate 轉接器軟體。請執行下列操作： 1. 從甲骨文軟件交付雲端下載 ggs_Adapters_Linux_x64.zip。 2. 解壓縮 ggs_Adapters_Linux_x64.zip. 3. 執行下列指令來安裝介面卡。 `tar -xvf ggs_Adapters_Linux_x64.tar`	DBA
設定資料汲取。	在來源伺服器上，設定資料汲取以將軌跡檔案從來源伺服器	DBA

任務	描述	所需技能
	傳輸到整合伺服器。建立資料泵參數檔案和軌跡檔案目錄。如需指示，請參閱設定純資料檔轉接器 (Oracle 文件集)。	

產生並移轉純資料檔

任務	描述	所需技能
產生平面檔案。	建立擷取檔案和控制檔，然後在整合伺服器上啟動解壓縮程序。這會擷取資料庫變更，並將來源資料庫寫入平面檔案。如需指示，請參閱使用純資料檔轉接器 (Oracle 文件集)。	DBA
將平面檔案載入目標資料庫。	將一般檔案載入目標 Amazon RDS for Oracle 資料庫執行個體。如需詳細資訊，請參閱使用 Oracle SQL* 載入器匯入 (Amazon RDS 文件)。	DBA

故障診斷

問題	解決方案
Oracle GoldenGate 平面檔案轉接器會產生錯誤。	如需轉接器錯誤的說明，請參閱尋找錯誤訊息 (Oracle 說明文件)。如需疑難排解指示，請參閱疑難排解純資料檔轉接器 (Oracle 說明文件)。

相關資源

- 安裝甲骨文 GoldenGate (Oracle 說明文件)

- 設定甲骨文 GoldenGate (Oracle 說明文件)
- 瞭解 Oracle GoldenGate 轉接程式 (Oracle 說明文件)
- 設定純資料檔轉接器 (Oracle 文件集)

更改 Python 和 Perl 應用程序以支持從 Microsoft SQL 服務器遷移到 Amazon Aurora PostgreSQL 兼容版本的數據庫

創建者：德瓦里卡帕特拉（AWS）和傑亞普拉卡什（AWS）

環境：PoC 或試點	來源：SQL 服務器	目標：Aurora 郵政兼容
R 類型：重新平台	工作負載：Microsoft；開源	技術：移轉；資料庫
AWS 服務：Amazon Aurora		

Summary

此模式描述了將資料庫從 Microsoft SQL Server 遷移到 Amazon Aurora PostgreSQL 相容版本時可能需要的應用程式儲存庫變更。此模式假設這些應用程式是以 Python 為基礎或 Perl，並為這些指令碼語言提供個別的指示。

將 SQL 伺服器資料庫移轉至 Aurora PostgreSQL 相容涉及結構描述轉換、資料庫物件轉換、資料移轉和資料載入。由於 PostgreSQL 和 SQL Server 之間的差異 (與資料類型、連線物件、語法和邏輯有關)，因此最困難的移轉工作包括對程式碼基底進行必要的變更，以便它可以與 PostgreSQL 正常運作。

對於基於 Python 的應用程序，連接對象和類分散在整個系統中。此外，Python 代碼庫可能會使用多個庫來連接到數據庫。如果資料庫連線介面變更，則執行應用程式內嵌查詢的物件也需要變更。

對於 Perl 型應用程式而言，變更包括連線物件、資料庫連線驅動程式、靜態和動態內嵌 SQL 陳述式，以及應用程式如何處理複雜的動態 DML 查詢和結果集。

遷移應用程式時，您也可以考慮 AWS 上可能的增強功能，例如以 Amazon Simple Storage Service (Amazon S3) 存取取代 FTP 伺服器。

應用程式移轉程序包含下列挑戰：

- 連接對象。如果連接對象分散在具有多個庫和函數調用的代碼中，則可能需要找到一種通用的方法來更改它們以支持 PostgreSQL。
- 記錄檢索或更新期間的錯誤或異常處理。如果您對傳回變數、結果集或資料框架的資料庫具有條件式建立、讀取、更新和刪除 (CRUD) 作業，則任何錯誤或例外狀況都可能會導致具有串

聯效果的應用程式錯誤。這些應該通過適當的驗證和保存點仔細處理。一個這樣的保存點是調用BEGIN...EXCEPTION...END塊內的大型內聯 SQL 查詢或數據庫對象。

- 控制交易及其驗證。其中包括手動和自動提交和復原。Perl 的 PostgreSQL 驅動程序要求您始終明確設置自動提交屬性。
- 處理動態 SQL 查詢。這需要對查詢邏輯和迭代測試有深刻的了解，以確保查詢按預期工作。
- 效能。您應該確定程式碼變更不會導致應用程式效能降低。

此模式詳細解釋了轉換過程。

先決條件和限制

先決條件

- Python 和 Perl 語法的工作知識。
- SQL 伺服器和 PostgreSQL 的基本技能。
- 瞭解您現有的應用程式架構。
- 存取您的應用程式程式碼、SQL 伺服器資料庫和 PostgreSQL 資料庫。
- 存取 Windows 或 Linux (或其他 Unix) 開發環境，其中包含用於開發、測試和驗證應用程式變更的認證。
- 對於基於 Python 的應用程序，您的應用程序可能需要的標準 Python 庫，例如熊貓來處理數據框，以及用於數據庫連接的 psycopg 2 或 SQLAlchemy。
- 對於 Perl 型應用程式，需要具有相依程式庫或模組的 Perl 套件。全面的 Perl 歸檔網絡（CPAN）模塊可以支持大多數應用程序需求。
- 所有必需的依賴自定義庫或模塊。
- SQL Server 讀取存取權和對 Aurora 的讀取/寫入存取權的資料庫認證。
- PostgreSQL 可透過服務和使用者驗證和偵錯應用程式變更。
- 在應用程序遷移期間訪問開發工具，例如視覺工作室代碼，崇高的文本或 PG Admin。

限制

- 某些 Python 或 Perl 版本、模組、程式庫和套件與雲端環境不相容。
- 某些用於 SQL Server 的協力廠商程式庫和架構無法取代，以支援 PostgreSQL 移轉。
- 效能差異可能需要變更應用程式、內嵌 Transact-SQL (T-SQL) 查詢、資料庫函數和預存程序。
- PostgreSQL 支持表名，列名和其他數據庫對象的小寫名稱。

- 某些資料類型 (例如 UUID 資料行) 僅以小寫儲存。Python 和 Perl 應用程序必須處理這種情況的差異。
- 字元編碼差異必須使用 PostgreSQL 資料庫中對應文字欄的正確資料類型來處理。

產品版本

- Python 3.6 或更新版本 (請使用支援您作業系統的版本)
- Perl 5.8.3 或更高版本（使用支持您的操作系統的版本）
- Aurora 與 PostgreSQL 兼容版本 4.2 或更高版本（查看詳細信息）

架構

源, 技術, 堆棧

- 腳本（應用程序編程）語言：Python 2.7 或更高版本，或 Perl 5.8
- 數據庫：Microsoft SQL 服務器版本 13
- 作業系統:紅帽企業版 (RHEL) 7

目標技術堆疊

- 腳本（應用程序編程）語言：Python 3.6 或更高版本，或 Perl 5.8 或更高版本
- 數據庫：Aurora 兼容 4.2
- 操作系統：7

移轉架構

工具

AWS 服務和工具

- Aurora PostgreSQL 相容版本是完全受管、與 PostgreSQL 相容且符合 ACID 標準的關聯式資料庫引擎，結合了高階商業資料庫的速度和可靠性，以及開放原始碼資料庫的成本效益。Aurora PostgreSQL 是 PostgreSQL 的立即取代方案，可讓您更輕鬆、更具成本效益的設定、操作和擴展新的和現有的 PostgreSQL 部署。

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。

其他工具

- Python 和 PostgresSQL 數據庫連接庫，如精神科技 2 和 SQL
- Perl 及其 DBI 模塊
- PostgreSQL 式終端

史诗

將您的應用程式儲存庫遷移至 PostgreSQL — 高階步驟

任務	描述	所需技能
請依照下列程式碼轉換步驟，將您的應用程式遷移至 PostgreSQL。	1. 為 PostgreSQL 設置特定於數據庫的 ODBC 驅動程序和庫。例如，您可以使用一個 CPAN 模塊的 Perl 和 pyodbc，通靈 2，或 SQL 煉金我的 Python。 2. 通過使用這些庫連接到 Aurora PostgreSQL 兼容轉換數據庫對象。 3. 在現有應用程式模組中套用程式碼變更以取得相容的 T-SQL 陳述式。 4. 在應用程序代碼中重寫數據庫特定的函數調用和存儲過程。 5. 處理應用程式變數及其用於內嵌 SQL 查詢之資料類型的變更。 6. 處理不相容的資料庫特定功能。	應用程式開發人員

任務	描述	所需技能
	7. 完整的數據庫遷移轉換後的應用程序代碼 end-to-end 測試。 8. 將 Microsoft SQL 伺服器的結果與您遷移到 PostgreSQL 的應用程式進行比較。 9. 執行 Microsoft SQL 伺服器和 PostgreSQL 之間的應用程式效能基準測試。 10.修訂應用程式呼叫的預存程序或內嵌 T-SQL 陳述式，以改善效能。 以下史詩提供了一些 Python 和 Perl 應用程序轉換任務的詳細說明。	

任務	描述	所需技能
針對移轉的每個步驟使用檢查清單。	針對應用程式遷移的每個步驟，將下列項目新增至檢查清單，包括最後一個步驟： • 請參閱 PostgreSQL 文件，以確保您的所有變更都與 PostgreSQL 標準相容。 • 檢查資料行是否有整數和浮動值。 • 識別插入、更新和擷取的列數，以及欄名稱和日期/時間戳記。您可以使用 diff 公用程式或撰寫指令碼來自動執行這些檢查。 • 完成大型內嵌 SQL 陳述式的效能檢查，並檢查應用程式的整體效能。 • 通過使用多個 try/catch 塊檢查數據庫操作和優雅程序退出的正確錯誤處理。 • 檢查以確保正確的日誌記錄過程已到位。	應用程式開發人員

分析和更新您的應用程式 — Python 程式碼庫

任務	描述	所需技能
分析您現有的 Python 程式碼庫。	您的分析應包括以下內容，以促進應用程序遷移過程： • 識別程式碼中的所有連線物件。	應用程式開發人員

任務	描述	所需技能
	• 識別所有不相容的內嵌 SQL 查詢 (例如 T-SQL 陳述式和預存程序)，並分析必要的變更。 • 檢閱程式碼的文件，並追蹤控制流程，以瞭解程式碼功能。稍後當您測試應用程式的效能或負載比較時，這會很有幫助。 • 了解應用程序的目的，以便您可以在數據庫轉換後有效地對其進行測試。大多數作為資料庫遷移轉換的候選 Python 應用程式可能是將資料從其他來源載入資料庫資料表的摘要，或是擷取資料從資料表擷取資料並將其轉換為不同的輸出格式 (例如 CSV、JSON 或平面檔案)，適合建立報表或進行 API 呼叫以執行驗證。	

任務	描述	所需技能
將您的資料庫連線轉換為支援 PostgreSQL。	大多數 Python 應用程序使用 pyodbc 庫與 SQL 服務器數據庫連接如下。 `import pyodbc` `....` `try:` `conn_string = "Driver=ODBC Driver 17 for SQL Server;UID={};PWD={};Server={};Database={}".format (conn_user, conn_password, conn_server, conn_database)` `conn = pyodbc.connect(conn_string)` `cur = conn.cursor()` `result = cur.execute(query_string)` `for row in result:` `print (row)` `except Exception as e:` `print(str(e))` 將數據庫連接轉換為支持 PostgreSQL，如下所示。 `import pyodbc` `import psycopg2` `....` `try:` `conn_string = ‘postgresql+psycopg2://’+conn_user+’:’+conn_password+’@’+conn`	應用程式開發人員

任務	描述	所需技能
	_server+'/'+conn_database conn = pyodbc.connect(conn_string, connect_args={'options':'-csearch_path=dbo'}) cur = conn.cursor() result = cur.execute(query_string) for row in result: print (row) except Exception as e: print(str(e))	

任務	描述	所需技能
將內聯 SQL 查 PostgreSQL 改為	將您的內嵌 SQL 查詢轉換為與 PostgreSQL 相容的格式。例如，下列 SQL Server 查詢會從資料表擷取字串。 `dtype = “type1” stm = ‘“SELECT TOP 1 searchcode FROM TypesTable (NOLOCK) WHERE code=”’ + “’” + str(dtype) + “’” # For Microsoft SQL Server Database Connection engine = create_engine(‘mssql+pyodbc:///?odbc_connect=%s’ % urllib.parse.quote_plus(conn_string), connect_args={‘connect_timeout’:login_timeout}) conn = engine_connect() rs = conn.execute(stm) for row in rs: print(row)`  轉換後，與 PostgreSQL 相容的內嵌式 SQL 查詢如下所示。  `dtype = “type1” stm = ‘“SELECT searchcode FROM TypesTable WHERE code=”’ + “’” + str(dtype) + “’ LIMIT 1” # For PostgreSQL Database Connection`	應用程式開發人員

任務	描述	所需技能
	<pre>engine = create_en gine('postgres+psy copg2://%s' %conn_str ing, connect_a rgs={'connect_time out':login_timeout}) conn = engine.connect() rs = conn.execute(stm) for row in rs: print(row)</pre>	

任務	描述	所需技能
處理動態 SQL 查詢。	動態 SQL 可以存在於一個腳本或多個 Python 腳本中。前面的例子演示瞭如何使用 Python 的字符串替換函數來插入變量來構建動態 SQL 查詢。另一種方法是在適用的地方附加查詢字符串與變量。 在下列範例中，根據函數傳回的值即時建構查詢字串。 `query = '"SELECT id from equity e join issues i on e.permId=i.permId where e.id'"` `query += get_id_filter(ids) + " e.id is NOT NULL` 這些類型的動態查詢在應用程式移轉期間非常常見。請遵循下列步驟來處理動態查詢： • 檢查整體語法 (例如，含有`JOIN`子句之`SELECT`陳述式的語法)。 • 驗證查詢中使用的所有變數或資料行名稱，例如`i`和`id`。 • 檢查查詢中使用的函數、引數和傳回值 (例如，`get_id_filter` 及其引數`ids`)。	應用程式開發人員

任務	描述	所需技能
處理結果集、變數和資料框架。	對於 Microsoft SQL 服務器，您可以使用 Python 方法，`fetchall()` 如`fetchone()` 或從數據庫中檢索結果集。您也可以使用`fetchmany(size)` 並指定要從結果集傳回的記錄數。若要這麼做，您可以使用 pyodbc 連線物件，如下列範例所示。 軟件（Microsoft SQL 服務器） `import pyodbc` `server = 'tcp:myse` `rver.database.wind` `ows.net'` `database = 'exampledb'` `username = 'exampleu` `sername'` `password = 'examplep` `assword'` `conn = pyodbc.co` `nnect('DRIVER={ODB` `C Driver 17 for SQL` ` Server};SERVER='+s` `erver+';DATABASE='` `+database+';UID='+` `username+';PWD='+` ` password)` `cursor = conn.cursor()` `cursor.execute("SELECT` ` * FROM ITEMS")` `row = cursor.fe` `tchone()` `while row:` `    print(row[0])` `    row = cursor.fe` `tchone()`	應用程式開發人員

任務	描述	所需技能
	在 Aurora 中，要執行類似的任務，例如連接到 PostgreSQL 和獲取結果集，您可以使用通靈 2 或 SQLAlchemy。這些 Python 庫提供了連接模塊和光標對象通過 PostgreSQL 數據庫記錄遍歷，如下面的例子。 精神科技 2（Aurora 兼容） `import psycopg2 query = "SELECT * FROM ITEMS;" //Initialize variables host=dbname=user= password=port=sslmode=connect_timeout="" connstring = "host='{host}' dbname='{dbname}' user='{user}' \ password='{password}'port='{port}' ".format(host=host,dbname=dbname,\ user=user,password=password,port=port) conn = psycopg2.connect(connstring) cursor = conn.cursor() cursor.execute(query) column_names = [column[0] for column in cursor.description] print("Column Names: ", column_names) print("Column values: "`	

任務	描述	所需技能
	<pre>for row in cursor: print("itemid :", row[0]) print("itemdescrpt ion :", row[1]) print("it emprice :", row[3]))</pre> SQL 煉金術（Aurora 後兼容） <pre>from sqlalchemy import create_engine from pandas import DataFrame conn_string = 'postgres ql://core:database @localhost:5432/ex ampledatabase' engine = create_en gine(conn_string) conn = engine.co nnect() dataid = 1001 result = conn.exec ute("SELECT * FROM ITEMS") df = DataFrame (result.fetchall()) df.columns = result.ke ys() df = pd.DataFrame() engine.connect() df = pd.read_sql_query(sql_query, engine, coerce_float=False) print(“df=”, df)</pre>	

任務	描述	所需技能
在移轉期間和移轉後測試應用程式。	測試遷移的 Python 應用程序是一個持續的過程。由於遷移包括連接對象更改（psycopg2 或 SQLAlchemy），錯誤處理，新功能（數據框），內聯 SQL 更改，批量複製功能（bcp而不是COPY）和類似的更改，因此必須在應用程序遷移期間和之後仔細測試。檢查： • 錯誤條件和處理 • 移轉後的任何記錄不相符 • 記錄更新或刪除 • 執行應用程式所需的時間	應用程式開發人員

分析和更新您的應用程序-Perl 代碼庫

任務	描述	所需技能
分析您現有的 Perl 代碼庫。	您的分析應包括以下內容，以促進應用程序遷移過程。您應該識別： • 任何 INI 或基於配置的代碼 • 資料庫特定的標準開放式資料庫連線 (ODBC) Perl 驅動程式或任何自訂驅動程式 • 內嵌和 T-SQL 查詢所需的程式碼變更 • 各種 Perl 模塊之間的相互作用（例如，由多個功能組件調用或使用的單個 Perl ODBC 連接對象）	應用程式開發人員

任務	描述	所需技能
	• 資料集和結果集處理 • 外部相依的 Perl 程式庫 • 應用程式中使用的任何 API • Perl 版本兼容性和驅動程序兼容 Aurora PostgreSQL 兼容	

任務	描述	所需技能
將來自 Perl 應用程序和 DBI 模塊的連接轉換為支持 PostgreSQL。	Perl 型應用程式通常使用 Perl DBI 模組，這是 Perl 程式設計語言的標準資料庫存取模組。您可以使用相同的 DBI 模塊與不同的驅動程序 SQL 服務器和 PostgreSQL。 如需有關所需 Perl 模組、安裝和其他指示的詳細資訊，請參閱 DBD:: Pg 文件。下面的例子連接到 Aurora PostgreSQL 兼容。`exampletest-aurorapg-database.cluster-samplecluster.us-east-.rds.amazonaws.com` `#!/usr/bin/perl` `use DBI;` `use strict;` `my $driver = "Pg";` `my $hostname = “examplet` `est-aurorapg-datab` `ase-sampleclusture` `.us-east.rds.amazo` `naws.com”` `my $dsn = "DBI:$driver:` ` dbname = $hostname` `;host = 127.0.0.1;port` ` = 5432";` `my $username = "postgres` `";` `my $password = "pass123"` `;` `$dbh = DBI->conn` `ect("dbi:Pg:dbname` `=$hostname;host=$h`	應用程式開發人員

任務	描述	所需技能
	ost;port=$port;opt ions=$options", $username, $password, {AutoCommit => 0, RaiseError => 1, PrintError => 0});	

任務	描述	所需技能
將內聯 SQL 查 PostgreSQL 改為	您的應用程式可能具有SELECT、DELETE、和類似陳述式的內嵌 SQL 查詢UPDATE，其中包含 PostgreSQL 不支援的查詢子句。例如，PostgreSQL 中NOLOCK不支援查詢關鍵字，例如TOP和。下列範例說明如何處理TOPNOLOCK、和布林變數。 在 SQL 伺服器中： `$sqlStr = $sqlStr . "WHERE a.student_id in (SELECT TOP $numofRecords c_student_id \ FROM active_student_record b WITH (NOLOCK) \ INNER JOIN student_contributor c WITH (NOLOCK) on c.contributor_id = b.c_st)` 對於 PostgreSQL，請轉換為： `$sqlStr = $sqlStr . "WHERE a.student_id in (SELECT TOP $numofRecords c_student_id \ FROM active_student_record b INNER JOIN student_contributor c \`	應用程式開發人員

任務	描述	所需技能
	on c.contributor_id = b.c_student_contr_id WHERE b_current_1 is true \ LIMIT $numofRecords)"	

任務	描述	所需技能
處理動態 SQL 查詢和 Perl 變量。	動態 SQL 查詢是在應用程式執行階段建置的 SQL 陳述式。這些查詢是在應用程序運行時動態構建的，具體取決於某些條件，因此查詢的全文直到運行時才知道。一個例子是一個財務分析應用程序，每天分析前 10 名股票，並且這些股票每天都在變化。SQL 表是根據表現最好的創建的，並且值在運行時才知道。 假設此範例的內嵌 SQL 查詢會傳遞至包裝函式，以取得變數中的結果集，然後變數會使用條件來判斷資料表是否存在： • 如果表存在，請不要創建它;做一些處理。 • 如果表不存在，請創建表並進行一些處理。 以下是變數處理的範例，接著是針對此使用案例的 SQL 伺服器和 PostgreSQL 查詢。 `my $tableexists = db_read( arg 1, $sql_qry, undef, 'writer'); my $table_already_exists = $tableexists->[0]{table_exists}; if ($table_already_exists){ # do some thing`	應用程式開發人員

任務	描述	所需技能
	} else { # do something else }	
	SQL 伺服器： my $sql_qry = “SELECT OBJECT_ID('$backen dTable', 'U') table_exi sts", undef, 'writer') ";	
	PostgreSQL： my $sql_qry = “SELECT TO_REGCLASS('$back endTable', 'U') table_exists", undef, 'writer')";	
	下列範例會在內嵌 SQL 中使用 Perl 變數，該變數會執行含有 a 的SELECT陳述式，JOIN以擷取資料表的主索引鍵和索引鍵資料行的位置。 SQL 伺服器： my $sql_qry = "SELECT column_name', character_maxi mum_length \ FROM INFORMATION_SCHEMA .COLUMNS \ WHERE TABLE_SCH EMA='$example_sche maInfo' \	

任務	描述	所需技能
	AND TABLE_NAME='$examp le_table' \ AND DATA_TYPE IN ('varchar','nvarch ar');"; PostgreSQL： my $sql_qry = "SELECT c1.column_name, c1.ordinal_position \ FROM information_schema .key_column_usage AS c LEFT \ JOIN information_schema .table_constraints AS t1 \ ON t1.constraint_name = c1.constraint_name \ WHERE t1.table_name = $example_schemaInf o'.'$example_table’ \ AND t1.constraint_type = 'PRIMARY KEY' ;";	

對基於 Perl 或基於 Python 的應用程序進行其他更改以支持 PostgreSQL

任務	描述	所需技能
將其他 SQL 伺服器建構轉換為 PostgreSQL。	下列變更適用於所有應用程式，不論程式語言為何。 • 以新的和適當的結構描述名稱來限定應用程式使用的資料庫物件。	應用程式開發人員

任務	描述	所需技能
	• 使用 PostgreSQL 中的定序功能處理相似運算子，以區分大小寫的比對。 • 處理不受支援的資料庫特定功能 DATEDIFFDATEADD，例如GETDATE、CONVERT、和CAST運算子。如需與 PostgreSQL 相容的同等函式，請參閱其他資訊一節中的原生或內建 SQL 函數。 • 處理比較陳述式中的布林值。 • 處理函數的返回值。這些可以是記錄集，數據幀，變量和布爾值。根據您的應用程式的需求來處理這些問題，並支援 PostgreSQL。 • 使用新的、使用者定義的 PostgreSQL 函式處理匿名區塊 (例如BEGIN TRAN)。 • 轉換列的批量插入。相當於 SQL Server 大量複製 (bcp) 公用程式的 PostgreSQL，這是從應用程式內部呼叫的。COPY • 轉換列連接運算符。SQL 伺服器用+於字串串連，但 PostgreSQL 會使用。||	

改善效能

任務	描述	所需技能
利用 AWS 服務進行效能增強。	遷移到 AWS 雲端時，您可以優化應用程式和資料庫設計以利用 AWS 服務。例如，如果來自 Python 應用程式的查詢連接至 Aurora PostgreSQL 相容資料庫伺服器，所花費的時間比原始 Microsoft SQL Server 查詢還要長，您可以考慮從 Aurora 伺服器直接建立歷史資料饋送至 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體，並使用 Amazon Athena 為使用者儀表板產生報表和分析資料查詢。	應用程式開發人員、雲端

相關資源

- Perl
- Perl DBI 模塊
- Python
- 心理學 2
- 方形煉金術
- 批量複製
- 批量複製-Microsoft SQL 服務器
- PostgreSQL
- 與 Amazon Aurora 合作

其他資訊

這兩個 Microsoft SQL 服務器和 Aurora PostgreSQL 兼容是 ANSI SQL 投訴。不過，當您將 Python 或 Perl 應用程式從 SQL Server 遷移到 PostgreSQL 時，您仍然應該注意語法、資料行資料類型、原生資料庫特定函數、大量插入和區分大小寫方面的任何不相容性。

以下各節提供有關可能不一致的詳細資訊。

資料類型比較

從 SQL Server 到 PostgreSQL 的資料類型變更可能會導致應用程式運作所產生的資料顯著差異。有關數據類型的比較，請參閱 S qlines 網站上的表格。

原生或內建 SQL 函數

SQL 伺服器和 PostgreSQL 資料庫之間的某些函式的行為不同。下表提供了一個比較。

Microsoft SQL Server	描述	PostgreSQL
`CAST`	將一個值從某個資料類型轉換至另一個類型。	PostgreSQL `type ::operator`
`GETDATE()`	以`YYYY-MM-DD hh:mm:ss.mmm` 格式返回當前數據庫系統的日期和時間。	`CLOCK_TIMESTAMP`
`DATEADD`	將時間/日期間隔添加到日期。	`INTERVAL`表達
`CONVERT`	將值轉換為特定資料格式。	`TO_CHAR`
`DATEDIFF`	返回兩個日期之間的差異。	`DATE_PART`
`TOP`	限制`SELECT`結果集中的列數。	`LIMIT/FETCH`

匿名圖塊

結構化 SQL 查詢會組織成多個區段，例如宣告、可執行檔和例外狀況處理。下表比較了一個簡單的匿名塊的 Microsoft SQL 服務器和 PostgreSQL 版本。對於複雜的匿名塊，我們建議您在應用程序中調用自定義數據庫函數。

Microsoft SQL Server

```
my $sql_qry1=
my $sql_qry2 =
my $sqlqry = "BEGIN TRAN
$sql_qry1 $sql_qry2
if @\@error !=0 ROLLBACK
TRAN
else COMIT TRAN";
```

PostgreSQL

```
my $sql_qry1=
my $sql_qry2 =
my $sql_qry = " DO \$\$
BEGIN
$header_sql $content_sql
END
\$\$";
```

其他差異

- 大量插入資料列：相當於 Microsoft SQL 伺服器 bcp 公用程式的 PostgreSQL 是複製。
- 區分大小寫：列名在 PostgreSQL 中區分大小寫，因此您必須將 SQL Server 列名稱轉換為小寫或大寫。當您擷取或比較資料，或在結果集或變數中放置欄名稱時，這會成為一個因素。下列範例會識別可能以大寫或小寫儲存值的資料行。

```
my $sql_qry = "SELECT $record_id FROM $exampleTable WHERE LOWER($record_name) =
 \'failed transaction\'";
```

- 串連：SQL Server 使用+作為字符串連接的運算符，而 PostgreSQL 使用。| |
- 驗證：在 PostgreSQL 的應用程式程式碼中使用它們之前，您應該先測試和驗證內嵌 SQL 查詢和函數。
- ORM 庫包含：您還可以查找包含或替換現有的數據庫連接庫與 Python ORM 庫，如 SQL Alchemy 和 PynoMoDB。這將有助於輕鬆地查詢和使用面向對象的範例操作數據庫中的數據。

依工作負載移轉模式

主題

- IBM
- Microsoft
- N/A
- 开源
- Oracle
- SAP

IBM

- 使用 AWS DMS 將 Db2 資料庫從 Amazon EC2 遷移到與 MySQL 相容的 Aurora
- 使用日誌傳送將 LUW 的 Db2 移轉至 Amazon EC2，以減少中斷時間
- 透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2
- 使用 AWS DMS 和 AWS SCT，從 Amazon EC2 上的 IBM Db2 遷移到 Aurora 與 PostgreSQL 相容
- 從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的阿帕奇 Tomcat

Microsoft

- 加速 Microsoft 工作負載探索和移轉到 AWS
- 更改 Python 和 Perl 應用程序以支持從 Microsoft SQL 服務器遷移到 Amazon Aurora PostgreSQL 兼容版本的數據庫
- 使用 Microsoft Excel 和 Python 為 AWS DMS 任務建立 AWS CloudFormation 範本
- 使用 AWS DMS 將 Microsoft SQL 伺服器資料庫匯出至 Amazon S3
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 將簡訊佇列從 Microsoft Azure 服務匯流排遷移到 Amazon SQS
- 通過使用 AWS DMS 將 Microsoft SQL 服務器數據庫從亞馬 Amazon EC2 遷移到 Amazon DocumentDB
- 使用 AWS DMS 和 AWS SCT 將 Microsoft SQL 伺服器資料庫遷移到 Aurora MySQL
- 將 .NET 應用程式從 Microsoft Azure 應用程式服務遷移到 AWS Elastic Beanstalk
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon EC2
- 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server
- 使用連結伺服器將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon RDS for SQL Server 服器
- 使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 將現場部署 Microsoft SQL 伺服器資料庫遷移到 Amazon Redshift 移
- 使用 AWS SCT 資料擷取代理程式將現場部署 Microsoft SQL 伺服器資料庫遷移至 Amazon Redshift 移
- ???
- 使用複製將資料從 Microsoft Azure Blob 遷移到 Amazon S3
- 使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器
- ???
- 使用 Amazon FSx 為 SQL 伺服器永遠在 FCI 設定異地同步備份基礎設施

N/A

- 在將主機移轉至 AWS 期間建立防火牆請求的核准程序

开源

- 在與 PostgreSQL 相容的 Aurora 中建立應用程式使用者和角色
- ???
- 將現場部署 MySQL 資料庫遷移到 Amazon EC2
- 將現場部署 MySQL 資料庫遷移到 Amazon RDS for MySQL
- 將內部部署 MySQL 資料庫遷移至 Aurora MySQL
- 將內部 PostgreSQL 料庫遷移至 Aurora
- 使用 Auto Scaling 能從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的 Apache Tomcat
- 從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk
- 使用合格邏輯從 Amazon EC2 上的 PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS 應用程式容器將現場部署 Java 應用程式遷移到 AWS
- 使用佩科納 XtraBackup、Amazon EFS 和 Amazon S3 將現場部署 MySQL 資料庫遷移到 Aurora MySQL
- 將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容
- 將 Redis 工作負載遷移到 AWS 上的 Redis 企業雲端
- 重新啟動 RHEL 來源伺服器後，自動重新啟動 AWS 複寫代理程式而不停用 SELinux
- 使用傳輸在兩個 Amazon RDS 資料庫執行個體之間傳輸 PostgreSQL 資料庫

Oracle

- 設定 Oracle 資料庫與 Aurora 相容之間的連結
- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR
- 使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS
- ???
- 使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS
- 使用 AWS SCT 和 AWS DMS 將適用於甲骨文的亞馬遜 RDS 遷移到適用於 PostgreSQL 的 CLI 馬遜 RDS CloudFormation
- ???
- 將 Amazon RDS for Oracle 執行個體遷移到另一個 VPC
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到 Amazon EC2
- 使用 Logstash 將現場部署 Oracle 資料庫遷移至 Amazon OpenSearch 服務
- 使用 AWS DMS 和 AWS SCT 將現場部署甲骨文資料庫遷移到適用於 MySQL 的 Amazon RDS for MySQL
- 將現場部署甲骨文資料庫遷移到 Amazon RDS for Oracle
- 透過資料庫連結使用直接的 Oracle 資料汲取匯入，將現場部署 Oracle 資料庫遷移到適用於甲骨文的 Amazon RDS
- 使用 Oracle 資料泵將現場部署 Oracle 資料庫遷移到亞馬遜 RDS
- 使用甲骨文旁觀者和 AWS DMS 將現場部署甲骨文資料庫遷移到亞馬遜 RDS
- 將現場部署 Oracle 資料庫遷移到 Amazon EC2 上的甲骨文
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 將甲骨文資料庫從亞馬 Amazon EC2 遷移到亞馬遜 RDS
- 使用 AWS DMS 將甲骨文資料庫遷移到 Amazon DynamoDB 資料庫
- 使用甲骨文 GoldenGate 平面檔案配接器將甲骨文資料庫遷移到亞馬遜 RDS
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Amazon Redshift 移
- 使用 AWS DMS 和 AWS SCT 將甲骨文資料庫遷移到 Aurora
- 使用 Oracle 資料泵和 AWS DMS 將甲骨文 JD 愛德華資料 EnterpriseOne 庫遷移到 AWS
- 使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL

- 使用 AWS DMS 將甲骨文 PeopleSoft 資料庫遷移到 AWS
- 將資料從內部部署 Oracle 資料庫遷 PostgreSQL 至 Aurora
- 從 Amazon RDS for Oracle 遷移到 Amazon RDS for MySQL
- 使用具體化視圖和 AWS DMS，從甲骨文 8i 或 9i 遷移到亞馬遜 RDS
- 使用和 AWS DMS 從甲骨文 8i 或 9i 遷移到亞馬遜 RDS SharePlex
- 使用甲骨文從甲骨文數據庫遷移到亞馬遜 RDS GoldenGate
- ???
- 使用 AWS DMS 從甲骨文遷移到 Amazon DocumentDB
- 在 Amazon ECS 上從甲骨文遷移 WebLogic 到阿帕奇湯姆貓（TomEE）
- 將基於函數的索引從甲骨文遷移到 PostgreSQL
- 將舊有應用程式從 Oracle Pro*C 移轉至 ECPG
- 將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列
- 將甲骨文數據庫錯誤代碼遷移到與 Amazon Aurora PostgreSQL 兼容的數據庫
- 將 Oracle 電子商務套件遷移到 Amazon RDS 定制
- 使用擴充功能將甲骨文原生函數遷移至 PostgreSQL
- 將甲骨文遷移 PeopleSoft 到 Amazon RDS 定制
- 將甲骨文功能遷移到 AWS 上的 PostgreSQL
- 將甲骨文序列遷移 _ 可重複使用的編譯包到 PostgreSQL
- 將虛擬生成的列從甲骨文遷移到 PostgreSQL
- 在 Aurora 相容上設定甲骨文 UTL_FILE 功能
- 從甲骨文遷移到 Amazon Aurora PostgreSQL 後驗證數據庫對象

SAP

- 將現場部署 SAP ASE 資料庫遷移至 Amazon EC2
- 使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器
- 使用 AWS SCT 和 AWS DMS 將亞馬 Amazon EC2 上的 SAP ASE 遷移到與 Amazon Aurora PostgreSQL 相容
- 使用應用程式遷移服務減少同質 SAP 移轉切換時間

更多模式

- 使用 CAST 醒目提示評估應用程式移轉至 AWS 雲端的準備程度
- 評估將 SQL 伺服器資料庫遷移到 AWS 上的 MongoDB 地圖集的查詢效能
- 使用 DR 協調器架構自動化跨區域容錯移轉和容錯回復
- 在 AWS 雲端建置進階大型主機檔案檢視器
- 使用混合式連結模式將資料中心擴充功能設定為 VMware Cloud on AWS
- 透過私人網路 Connect 至應用程式移轉服務資料和控制平面
- 容器化已由 Blu Age 現代化的大型主機工作負載
- 將甲骨文查詢轉換為 SQL 數據庫
- 將太數據標準化時間功能轉換為 Amazon Redshift SQL
- 將太數據重置功能轉換為 Amazon Redshift SQL
- 使用 AWS Backup 跨帳戶複製 Amazon DynamoDB 表
- 使用私有靜態 IP 在 Amazon EC2 上部署卡桑德拉集群，以避免重新平衡
- 使用 AWS CDK 來部署多堆疊應用程式 TypeScript
- 使用 Aurora 中的自訂端點模擬 Oracle RAC 工作負載
- 使用 AWR 報告估計甲骨文資料庫的 Amazon RDS 引擎大小
- 在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight
- 處理動態 SQL 語句中 Aurora PostgreSQL 塊
- 在 Aurora 兼容後處理過載的甲骨文功能
- 將 VMware 網路洞察與 VMware Cloud on AWS 整合
- 將適用於 Oracle 資料庫執行個體的 Amazon RDS 移轉到使用 AMS 的其他帳戶
- 使用以下方式將現場部署阿帕奇卡夫卡叢集遷移到 Amazon MSK MirrorMaker
- 使用 AWS Glue 將阿帕奇卡桑德拉工作負載遷移到 Amazon Keyspaces
- 使用 SharePlex 和 AWS DMS 從甲骨文 8i 或 9i 遷移到適用於甲骨文的亞馬遜 RDS
- 使用萬 LiveData 迪斯科遷移器將 Hadoop 資料遷移到 Amazon S3
- 將具有 100 個以上引數的甲骨文函數和程序遷移到 PostgreSQL
- 將甲骨文輸出綁定變量遷移到 PostgreSQL 數據庫
- 使用 AWS MGN 將 RHEL BYOL 系統遷移到 AWS 包含授權的執行個體
- ???
- 使用分散式可用性群組將 SQL 伺服器遷移到 AWS

- ???
- ???
- 使用 OpenText 微焦點企業伺服器和 LRS X 在 AWS 上現代化大型主機輸出管理 PageCenter
- 在 AWS 上從 F5 遷移到 Application Load Balancer 時修改 HTTP 標頭
- 將 Microsoft SQL 伺服器遷移到 AWS 雲端後解決連接錯誤
- 使用 VMware 詠嘆調操作的日誌，將日誌從 AWS 雲端傳送到潑濺
- EnterpriseOne 使用 AWS 彈性災難復原為 Oracle JD 愛德華設定災難復原
- 使用 AWS 私有 CA 和 AWS 記憶體簡化私有憑證管理
- 以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3

現代化

主題

- 在 CAST 影像中分析並視覺化軟體架構
- 使用 CAST 醒目提示評估應用程式移轉至 AWS 雲端的準備程度
- 使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3
- 使用 Amazon EC2 Auto Scaling 和 Systems Manager 建置微焦點企業伺服器 PAC
- 在 Amazon 服務中建立多租戶無伺服器架構 OpenSearch
- 使用 AWS CDK 來部署多堆疊應用程式 TypeScript
- 使用 AWS SAM 自動部署巢狀應用程式
- 使用 AWS Lambda 權杖自動販賣機為 Amazon S3 實作 SaaS 租用戶隔離
- 使用 AWS Step Functions 實作無伺服器傳奇模式
- 使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式
- 在 AWS 上將 ASP.NET 網頁表單應用程式現代化
- 使用 AWS Fargate 大規模執行事件驅動和排程的工作負載
- 在 SaaS 架構中使用 C# 和 AWS CDK 進行筒倉模型的租用戶上線
- 通過使用 CQRS 和事件採購將巨石分解為微服務
- 更多模式

在 CAST 影像中分析並視覺化軟體架構

創建者阿爾皮塔·辛哈（投射軟件）和詹姆斯·赫勒爾（鑄造軟件）

環境：生產	技術：現代化	工作負載：所有其他工作

Summary

此模式顯示如何使用 CAST Imaging 以視覺化方式瀏覽複雜的軟體系統，並對軟體結構執行精確的分析。透過這種方式使用 CAST Imaging，您可以針對應用程式的架構做出更明智的決策，尤其是針對現代化目的。

若要在 CAST 影像中檢視應用程式的架構，您必須先透過 CAST 主控台上載應用程式的原始程式碼。然後，主控台會將應用程式的資料發佈至 CAST Imaging，您可以在其中逐層視覺化和瀏覽應用程式架構。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 用於鑄造成像的 Amazon 機器圖像（AMI）
- 包含下列項目的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體 (建議使用記憶體最佳化的 r5.xlarge Amazon EC2 執行個體)：
  - 4 vCPU
  - 32 GB 公斤記憶體
  - 500 GB 最低一般用途固態硬碟 (gp3) 容量
- CAST 控制台和 CAST 影像授權金鑰 (若要取得所需的授權金鑰，請透過 aws.contact-me@castsoftware.com 聯絡 CAST)
- 您要以壓縮（.zip）格式分析的應用程序的完整源代碼
- Microsoft 邊緣, 火狐瀏覽器, 或谷歌瀏覽器

架構

下圖顯示透過 CAST Console 啟動應用程式原始程式碼，然後在 CAST 影像中檢視該程式碼的範例工作流程：

該圖顯示以下工作流程：

1. CAST 會透過反向工程的前端、中介軟體和後端程式碼，產生應用程式原始碼中繼資料。
2. CAST 產生的應用程式資料會自動匯入到 CAST 影像中，在此處可視化和分析。

以下是此過程如何工作的快照：

工具

- CAST Imaging 是一款以瀏覽器為基礎的應用程式，可協助您以視覺化的方式檢視和瀏覽軟體系統，以便您對其架構做出明智的決策。
- CAST 主控台是一個以瀏覽器為基礎的應用程式，可協助您設定、執行和管理 CAST AIP 分析。

注意：用於鑄造成像的 AMI 中包含了鑄造成像和 CAST 控制台。

史诗

設定 CAST 影像環境

任務	描述	所需技能
執行初始 CAST 主控台設定。	1. 開啟您的網頁瀏覽器，然後輸入下列網址連線至 CAST 主控台：http://localhost:8081 2. 出現提示時，請輸入您的 CAST 控制台授權金鑰。然後選擇下一步。	軟體架構師、開發人員、技術領袖

任務	描述	所需技能
	3. 檢閱組態設定。如果不需要變更，請選擇「儲存並完成」。	
執行初始的 CAST 影像組態。	1. 開啟您的網頁瀏覽器，然後輸入下列網址連線至 CAST 影像：http://localhost:8083 2. 出現提示時，請輸入 admin 以輸入使用者名稱和密碼來登入。 3. 出現提示時，輸入您的 CAST 影像授權金鑰。然後，選擇「更新」以儲存金鑰。	軟體架構師、開發人員、技術領袖

任務	描述	所需技能
設定 CAST 延伸本機伺服器。	(選擇性) 根據預設，CAST 延伸本機伺服器設定為在離線模式下運作。如果這是可以接受的，則不需要額外的配置。不過，如果您偏好以線上/代理模式設定 CAST Extend 本機伺服器，並直接連線至 CAST Extend，請依照下列步驟執行。 注意：如需 CAST 延伸證明資料，請參閱 CAST 延伸註冊頁面。 1. 使用桌面上的 CAST 延伸管理中心捷徑載入您的網頁瀏覽器，並連線至 CAST 延伸本機伺服器。 2. 選擇「線上」選項。 3. 輸入您的 CAST Extend 憑證（電子郵件和密碼），然後選擇「保存」以完成該過程。	軟體架構師、開發人員、技術領袖

將您的應用程式加入 CAST 影像

任務	描述	所需技能
準備應用程式的原始程式碼。	將應用程式的原始程式碼儲存在單一壓縮的 .zip 檔案中。	軟體架構師、開發人員、技術領袖
將您的應用程式新增至 CAST 主控台。	1. 開啟您的網頁瀏覽器，然後輸入下列網址連線至 CAST	軟體架構師、開發人員、技術領袖

任務	描述	所需技能
	主控台：http://localhost:8081 2. 出現提示時，請輸入 admin 以輸入使用者名稱和密碼來登入。 3. 選擇新增應用程式。然後，輸入應用程式名稱，然後選擇新增。	
開啟原始程式碼傳送精靈。	尋找您在 CAST 主控台中建立的應用程式。然後，選擇添加版本。	軟體架構師、開發人員、技術領袖
上傳應用程式的原始程式碼。	執行以下任意一項： • 將包含應用程式原始程式碼的 .zip 檔案拖放到原始程式碼傳遞精靈中。 或 • 選擇上傳雲端圖示。然後，開啟包含應用程式原始程式碼的 .zip 檔案。	軟體架構師、開發人員、技術領袖

任務	描述	所需技能
開始分析過程。	1. 在傳遞精靈中，提供版本詳細資料並指定組態選項。如需詳細資訊，請參閱 CAST 影像文件中的 CAST 影像標準上線。 2. 確定已選取「發佈至 CAST 影像」選項。然後，選擇「繼續」。 注意：選擇「繼續」會啟動原始程式碼的分析程序。CAST Console 中的進度視窗會顯示分析程序的每個步驟，並在分析完成時顯示通知。	軟體架構師、開發人員、技術領袖

驗證分析結果和發佈至 CAST 影像的資料

任務	描述	所需技能
檢查狀態和記錄檔。	完成所有分析動作後，請確認進度視窗中有成功訊息。 附註：您可以在完成後立即檢查每個分析動作的個別記錄。若要檢閱特定動作的記錄檔，請在「進度」視窗中選擇「檢視記錄」。	軟體架構師、開發人員、技術領袖
檢查應用程序詳細信息。	在「應用程式詳細資料」面板中，檢閱有關分析結果的詳細資料。請務必查看所發現的技術和原始程式碼組織。	軟體架構師、開發人員、技術領袖

任務	描述	所需技能
驗證並存取 CAST 影像。	1. 在 CAST 主控台的「應用程式管理」窗格中，確認應用程式的版本狀態是否已處理複製影像。「鑄造影像」圖示隨即出現。 2. 選擇 CAST 影像圖示，直接導覽至 CAST 影像中的應用程式資料。 注意：「已處理複製影像」狀態表示原始程式碼已分析並上傳至您的 CAST 影像實例。	軟體架構師、開發人員、技術領袖

開始使用 CAST 影像分析您的應用

任務	描述	所需技能
登入至 CAST 影像。	開啟「鑄造影像」，然後輸入預設的管理員認證 (系統管理員/管理員)。您應用程式的資料隨即顯示。	軟體架構師、開發人員、技術領袖
在 CAST 影像中探索應用程式的資料。	使用 CAST 影像功能開始檢視您的軟體架構。 如需有關如何使用 CAST 影像功能的快速教學課程，請選擇「說明」圖示以顯示「CAST 影像輔助程式」。 如需詳細資訊，請參閱 CAST 影像使用者指南。	軟體架構師、開發人員、技術領袖

相關資源

CAST 控制台文檔

- 登入
- 透過 CAST 主控台設定選項

鑄造影像文件

- CAST 影像的應用程式上線-先決條件
- 新增 CAST 影像的新應用程式
- CAST 成像的標準入職 — 檢查結果
- 登入
- 配置選項 — 管理中心 GUI

有關 AWS 上 CAST 影像的更多資源

- AWS 的應用程式現代化由 CAST 加速 — 技術 (AWS PartnerCast 網路研討會，需要免費帳戶)
- 使用 CAST 和 AWS Migration Hub 重構空間來現代化舊版應用程式 (AWS 部落格文章)
- 運用 CAST 影像將應用程式現代化為 AWS 架構 (AWS 研討會)
- AWS Marketplace：鑄造影像
- AWS 資源上的所有轉換

使用 CAST 醒目提示評估應用程式移轉至 AWS 雲端的準備程度

創建者格雷格·里維拉（鑄造軟件）

環境：生產	來源：舊版應用程序源代碼	目標：在 AWS 中重構應用程式程式碼
R 型：重新建築	工作負載：IBM；Microsoft；開放原始碼；甲骨文	技術：現代化；遷移；容器與微服務
AWS 服務：Amazon RDS; Amazon S3		

Summary

CAST Highlight 是用於執行快速應用程式組合分析的軟體即服務 (SaaS) 解決方案。此模式說明如何設定和使用 CAST Highlight 來評估組織 IT 產品組合中自訂軟體應用程式的雲端準備程度，以及規劃現代化或遷移至 Amazon Web Services (AWS) 雲端。

CAST Highlight 針對應用程式的雲端準備程度產生深入解析、識別遷移前需要移除的程式碼攔截器、估計移除這些封鎖程式的工作量，以及建議個別應用程式在遷移後可使用的 AWS 服務。

此模式描述設定和使用 CAST Highlight 的程序，其中包含五個步驟：新使用者設定、應用程式管理、行銷活動管理、原始程式碼分析和結果分析。您必須完成此病毒碼「Epics」一節中的所有步驟，以確保應用程式掃描和分析成功。

先決條件和限制

先決條件

- 具有投資組合管理器權限的活躍 CAST 精彩帳戶。
- 您的本機電腦上至少有 300 MB 的可用磁碟空間和 4 GB 記憶體，以安裝 CAST 反白顯示本機代理程式。
- Microsoft 視窗 8 或更高版本。
- 您的應用程式原始程式碼必須儲存在可從安裝本機代理程式的機器存取的文字檔中。沒有源代碼離開場所，並且所有代碼都在本地掃描。

架構

下圖說明使用「轉換反白」的工作流程。

工作流程由以下步驟組成：

1. 登入 CAST 亮點入口網站，下載本機代理程式，然後將其安裝在您的本機電腦上。Amazon Simple Storage Service (Amazon S3) 會存放本機代理程式安裝套件。
2. 掃描您的源代碼文件並生成一個結果文件。
3. 將結果檔案上傳至 CAST 精華片段入口網站。重要事項：結果檔案中不包含任何原始程式碼。
4. 針對您掃描的每個應用程式回答調查問卷問題。
5. 檢視 CAST 亮點入口網站中提供的儀表板和報告。Amazon Relational Database Service 服務 (Amazon RDS) 會儲存程式碼掃描、分析結果和 CAST 突出顯示軟體資料。

技術, 堆

CAST Highlight 支援下列技術來分析應用程式雲端準備程度：

- Java
- COBOL
- C#
- C++
- Clojure
- PHP
- JavaScript
- TypeScript
- Python
- Microsoft 事务
- VB.Net
- Kotlin
- Scala
- Swift

自動化和規模

- CLI 分析器可用於自動執行 CAST 亮點分析過程。

工具

如果符合所有先決條件，則不需要此模式的工具。不過，您可以選擇使用選用的工具，例如原始程式碼管理 (SCM) 公用程式、程式碼擷取器或其他工具來管理原始程式碼檔案。

史诗

新使用者設定

任務	描述	所需技能
激活您的 CAST 突出顯示帳戶並選擇您的密碼。	所有首次使用 CAST 精華的用戶都會收到一封帳戶激活 按照激活鏈接激活您的 CAST 突出顯示帳戶並輸入密碼以完成激活過程。	N/A
登入 CAST 精華片段入口網站。	輸入新密碼後，「CAST 亮點」首頁會顯示出來。使用您的使用者認證登入 CAST 亮點入口網站。	N/A

應用程式管理

任務	描述	所需技能
建立應用程式記錄。	在「CAST 反白」入口網站中，導覽至「管理學檔」區段中的「管理應用程式」標籤。在畫面頂端的「應用程式」動態磚中，選擇「新增」。	N/A

任務	描述	所需技能
選擇應用程式名稱。	輸入應用程式的名稱，然後選擇 [儲存]。此名稱用於「CAST 亮點」中的應用程式記錄。	N/A
對所有應用程式重複上述步驟。	針對您要掃描的每個應用程式重複這些步驟。	N/A

活動管理

任務	描述	所需技能
建立行銷活動。	CAST 精華片段使用「促銷活動」來描述一組將在特定時間進行分析的應用程式。在「CAST 醒目提示」入口網站中，導覽至「管理投資組合」區段的「管理行銷活動」 選擇 [建立行銷活動] 以啟動廣告活動建立畫面	N/A
輸入廣告活動的名稱並選擇結束日期。	輸入廣告活動的名稱，然後選擇結束日期。 重要提示：活動截止日期後，貢獻者無法提交申請分析結果。	N/A
決定包括源代碼掃描，調查答案以及域和應用程序範圍。	選擇一個或多個用於通過定性信息增強源代碼分析數據的標準調查。調查問卷類別為「業務影響」、「軟體維護工作」CloudReady、「應用程式屬性」 和「綠色影響」。選擇廣告活動期間分析的網域和應用程式。	N/A

任務	描述	所需技能
	重要:在開始廣告活動之前，請務必在「管理應用程式」區段中新增要掃描的所有應用程式。	
自訂啟動訊息。	自訂將透過電子郵件傳送給與促銷活動中應用程式相關聯的所有貢獻者的啟動訊息。	N/A
啟動行銷活動。	選擇 [完成] 以啟動行銷活動。	N/A

源代碼分析

任務	描述	所需技能
下載 CAST 突出顯示本地代理程式。	在 CAST 亮點入口網站中，選擇應用程式掃描，然後將本機代理程式下載到您的本機電腦。	N/A
安裝本機代理程式。	啟動 CAST Highlight Setup .exe 安裝程式，然後按照出現的設置說明進行操作。安裝本機代理程式之後，您就可以分析您的應用程式了。	N/A
定義本機代理程式碼掃描的範圍。	代碼分析是在文件級別執行的，不考慮文件之間的邏輯鏈接或依賴關係。所有文件被認為是相等的，應用程序的一部分。 若要提供準確且一致的結果，請使用本機代理程式中提供的	N/A

任務	描述	所需技能
	檔案或資料夾排除功能來準備程式碼掃描範圍。	
包含開放原始碼或 COTS 套件。	(選擇性) 如果您想要包含開放原始碼或商業 off-the-shelf (COTS) 套件，請確定這些套件已包含在您打算掃描的資料夾中。通常情況下，外部庫被分組在一個名為「第三方」或類似的子文件夾中，並且主代碼通常位於「src/main」文件夾中。	N/A
排除測試類別。	測試類通常從源代碼分析中排除，因為它們通常不是編譯應用程序的一部分。但是，如果需要，您可以選擇將它們包含在掃描中。	N/A
排除 SCM、建置和部署資料夾。	為了獲得更一致的結果，您應該避免在掃描中包含 SCM，構建或部署文件夾（例如 .git 或 .svn 文件）。	N/A
包含相依性檔案。	如果您想要深入瞭解其實體檔案不屬於您正在掃描的資料夾的一部分的架構和相依性，請確定您包含相依性檔案 (例如 pom.xml、建置 .gradle、package.json 或 .vcsproj 檔案)。	N/A
呼叫本機代理程式。	在您的本機 Windows 電腦上執行本機代理程式。	N/A

任務	描述	所需技能
選擇包含源代碼的文件夾。	選擇包含源代碼的文件夾。您可以新增多個要由本機代理程式探索的資料夾。雖然本機代理程式確實支援透過網路路徑進行來源探索，但您應該確定來源資料夾位於您的本機電腦上。 重要事項：如果來源資料夾中有超過 10,000 個檔案，我們建議您執行多次掃描。	N/A
啟動檔案探索。	在本機代理程式儀表板上，選擇探索檔案。本機代理程式會探索資料夾和子資料夾中的檔案，並偵測其技術。您可以選擇「取消」按鈕，隨時取消探查。 檔案探索完成後，「本機代理程式」會列出找到的資料夾和檔案。「技術」欄會顯示相關聯的技術和檔案計數。「路徑」欄會顯示資料夾和檔案的位置。	N/A

任務	描述	所需技能
優化源代碼掃描配置。	(選擇性) 若要精簡本機代理程式掃描，您可以停用特定資料夾或檔案的一或多項技術。如果停用所有技術，您的資料夾或檔案將會從掃描的範圍中排除。 若要停用技術，請選擇您要停用之技術的黃色標籤。您也可以在將游標停留在檔案或資料夾上時，選擇篩選器圖示，將技術與特定檔案或資料夾建立關聯。系統會儲存這些設定，並加速資料夾或檔案的探索程序。	N/A
開始源代碼掃描。	配置掃描後，選擇「掃描文件」開始掃描過程。	N/A

任務	描述	所需技能
檢查是否有綠色或灰色標籤。	原始碼掃描完成後，資料夾和檔案層級會顯示狀態標籤。 綠色標籤表示已使用相關技術正確掃描檔案。 灰色標籤表示未掃描並排除檔案。當您將鼠標懸停在每個文件的標籤上時，會顯示其排除的原因。排除檔案的可能原因包括二進位檔案、無法讀取的檔案、遺失檔案、外部程式庫、編碼檔案、產生的檔案、語法錯誤、不符合預期語言的內容、不符合足夠分析準則的程式碼、超過大小限制 (10 MB) 的檔案、逾時問題或無法使用分析器。	N/A
修改掃描配置並再次掃描代碼。	（選擇性）您可以修改掃瞄組態設定，然後選擇「掃瞄檔案」以再次掃瞄檔案。	N/A
確認掃描結果。	如果掃描結果符合您的需求，請選擇「確認結果」。	N/A

任務	描述	所需技能
檢視本機代理程式找到的架構和軟體程式庫。	檢視應用程式在程式碼掃描期間由本機代理程式所使用或參照的架構和軟體程式庫。 您可以通過選擇單獨的切換按鈕來保留或忽略這些列表中的元素。 選擇確認依賴關係繼續。 重要事項：如果架構已關閉，則不會在 CAST Highlight 入口網站中列出，也不會附加至您的應用程式。	N/A
保存代碼掃描結果。	本機代理程式會顯示依技術分組的程式碼掃描結果摘要。 選擇 [儲存]，然後指定要儲存結果的目標資料夾。本機代理程式會針對每個掃描產生一個 .zip 檔案，其中包含所有分析結果。 根據不同技術和根來源資料夾的數目，本機代理程式會自動產生具有 .Techno FolderNamey.Date.csv 命名結構的一或多個. csv 檔案。	N/A
將代碼掃描結果上傳到 CAST 突出顯示入口網站。	在 CAST Highlight 入口網站中，選擇您在「應用程式掃描」區段中分析的應用程式。 選擇「上傳結果」，然後選擇 .csv 檔案。您也可以個別上傳 .csv 檔案。上傳每個檔案後，螢幕上會顯示上傳記錄。	N/A

任務	描述	所需技能
如有需要，請刪除分析結果檔案。	(選擇性) 您可以選擇資源回收筒圖示，在上載程序期間隨時刪除分析結果檔案。 重要事項：只有具備學檔管理員權限的使用者或上傳結果的參與者才能刪除結果。	N/A
回答申請調查。	問卷調查按鈕會出現在需要調查問卷的應用程式上。選擇 [調查問卷]，回答調查問卷每個區段的問題，然後在完成後選擇 [送出]。 調查問卷的進度會顯示在畫面上方。您可以在提交所有強制性資料後提交結果。不過，您可以透過回答所有問題來豐富組織 CAST Highlight 實例中的資料。	N/A
提交代碼掃描結果。	上傳應用程式的所有 .csv 結果檔案並完成調查問題後，請在 [應用程式掃描] 區段中選擇 [提交]。需要執行此步驟，才能完成此程序，並確保結果可在「CAST 亮點」入口網站中取得。	N/A

結果分析

任務	描述	所需技能
檢視 CAST 精選入口網站首頁。	CAST Highlight 入口網站首頁包含有關您應用程式產品組	N/A

任務	描述	所需技能
	合的高階資訊的圖塊，例如軟體健康狀態 CloudReady，以及整個產品組合的開放原始碼安全分數。首頁還包括已登錄的應用程序的數量。如需有關 CAST 亮點量度定義和測量方法的詳細資訊，請參閱 CAST 重點 — 度量與方法 (Microsoft PowerPoint 簡報)。	
檢視 CloudReady 儀表板。	選擇CloudReady 瓷磚以開啟 CloudReady 儀表板窗格。這是評估應用程式雲端準備程度的主要組合層級儀表板。它可協助您規劃和制定雲端移轉的產品組合藍圖	N/A

任務	描述	所需技能
檢視雲端產品組合顧問儀表板。	雲端產品組合顧問儀表板會自動將應用程式區分為建議的移轉類別 細分基於每個應用程序的技術特徵。因素包括原始程式碼分析 (雲端準備程度、軟體備援等)，以及來自調查問卷的業務影響。選擇右上角的 [計算] 以產生初始分段建議。 儀表板頂端圖表中的泡泡代表產品組合中的每個應用程式，依建議的區段進行組織。每個應用程式也會列在圖表下方的資料表格中，包括每個應用程式的相關指標。 建議的可能區段包括： • 重新裝載 — 建議變更應用程式的基礎結構組態，以便使用基礎結構即服務 (IaaS) 解決方案將其提升並轉移至雲端。 • 重構 — 建議在不變更架構或功能的情況下對應用程式程式碼執行適度修改，以便透過使用容器即服務 (CAA) 或平台即服務 (PaaS) 解決方案來移轉應用程式程式碼。 • 重新架構 — 建議您大幅修改應用程式程式碼以改善應用程式的健康狀態，並使用 PaaS 解決方案準備移轉，或使用函數即服務 (FaaS) 解	N/A

任務	描述	所需技能
	決方案將其部署為無伺服器應用程式。 • 重建 — 建議您捨棄應用程式的程式碼，並使用 PaaS 解決方案在雲端中再次開發，或使用 FaaS 解決方案再次開發為無伺服器應用程式。 • 淘汰 — 建議完全放棄應用程式，或可能以商業軟體即服務 (SaaS) 替代方案取代應用程式。	
修改區段建議。	在某些情況下，您可能會選擇變更「CAST 醒目提示」建議的區段。您可以瀏覽至資料表格中的應用程式，並從應用程式名稱旁邊的下拉式清單中選取不同的區段來執行此操作。然後選擇保存在右上角保存更改。 您也可以選擇右上角的「匯出」，隨時匯出此資料。	N/A
選擇要分析的應用程式。	在雲端產品組合顧問儀表板上，選擇應用程式泡泡來分析該應用程式。在泡泡圖之後選擇表格中的應用程式名稱，以開始進行更深入的分析。 您可以使用不同的儀表板來分析個別應用程式，例如程式碼洞察 (軟體健康狀態模式)、趨勢和軟體組合 (開放原始碼風險)。	N/A

任務	描述	所需技能
分析個別應用程式的 CloudReady 結果。	選擇顯CloudReady示應用程式整體 CloudReady 分數的標籤。此分數是根據 CloudReady 調查問卷答案和條 CloudReady 碼掃描的組合計算的加權平均值。調查問卷問題的答案會顯示在圖標下方的表格中。 選擇「CloudReady 程式碼掃描」以檢視程式碼掃描結果。有一份掃描應用程 CloudReady 式程式碼的病毒碼清單。此清單包含下列欄： • 雲需求是特定的代碼模式。 • 技術是模式的編程語言。「影響」是模式對應用程序的影響（C = 代碼，F = 框架，A = 架構）。 • 重要性是在遷移之前解決此模式的重要性等級。 • 貢獻是這種模式對整體 CloudReady 得分有所貢獻的方式。如果圖案是綠色的，它是一個助推器，並增加 CloudReady 分數。如果圖案是紅色的，它是一個阻止程序並降低 CloudReady 分數。如果圖案沒有顏色，則它是未檢測到的阻止程序並增加 CloudReady 分數。 • 路障是阻止程序模式的個別出現次數。選擇路障編號以	N/A

任務	描述	所需技能
	顯示偵測到病毒碼的原始程式碼檔案清單。 • 東. 努力是指修復每一列路障所需的天數的估計值。	
將數據導出到 Microsoft	(選擇性) 選擇「匯出至 Excel」以匯出資料以供進一步分析。應用程式分析結果資料可用於進一步分析應用程式的雲端準備程度，並判斷移轉之前必須更新哪些程式碼。	N/A
檢視建議。	選擇「CloudReady 程式碼掃描」旁的「建議」以檢視「雲端服務建議」畫面。這會根據應用程式的特性識別應用程式可採用的 AWS 服務。 重複此步驟以檢視您所分析之所有應用程式的建議。	N/A

相關資源

活動管理

- CAST 精華基礎認證培訓第三節：組合配置（視頻）

源代碼分析

- CAST 精華基礎認證培訓第四組：應用分析（視頻）

其他資源

- AWS Marketplace 中的演員焦點
- AWS 和 CAST：加速應用程式現代化

- CAST 亮點 — 文件、產品教學課程和協力廠商工具
- CAST 亮點 — 雲端就緒產品示範 (影片)
- 應用程式產品組合現代化與 CAST 重點 (AWS 研討會)

使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3

由虎斑病房（AWS）創建

程式碼儲存庫:使用 DynamoDB TLL 將項目存檔至 S3	環境：PoC 或試點	技術：現代化；資料庫；無伺服器；儲存與備份；成本管理
工作負載：開源	AWS 服務：Amazon S3；Amazon DynamoDB；Amazon Kinesis；AWS Lambda	

Summary

此模式提供了從 Amazon DynamoDB 表中移除舊資料的步驟，並將其存檔到 Amazon Web 服務 (AWS) 上的 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體，而無需管理一群伺服器。

此模式會使用 Amazon DynamoDB 存留時間 (TTL) 自動刪除舊項目，並使用 Amazon DynamoDB 串流來擷取 TTL 過期的項目。然後，它會將 DynamoDB Streams 連接到 AWS Lambda，而不需佈建或管理任何伺服器即可執行程式碼。

將新項目新增至 DynamoDB 串流時，會啟動 Lambda 函數，並將資料寫入 Amazon 資料 Firehose 交付串流。Firehose 提供簡單、全受管的解決方案，可將資料當做存檔載入 Amazon S3。

DynamoDB 通常用於儲存時間序列資料，例如網頁點擊流資料或來自感應器和連線裝置的物聯網 (IoT) 資料。許多客戶不想刪除存取頻率較低的項目，而是希望將其封存以供稽核之用。TTL 會根據時間戳記屬性自動刪除項目，藉此簡化此封存。

TTL 刪除的項目可以在 DynamoDB Streams 中識別，該串流會擷取一系列按時排序的項目層級修改，並將序列儲存在記錄中最多 24 小時。Lambda 函數可以使用此資料，並將其存檔在 Amazon S3 儲存貯體中，以降低儲存成本。為了進一步降低成本，您可以建立 Amazon S3 生命週期規則，將資料 (一旦建立完成) 自動轉移到成本最低的儲存類別，例如 S3 Glacier 即時擷取或 S3 Glacier 彈性擷取，或用於長期儲存的 Amazon S3 Glacier 深度存檔。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 1.7 或更新版本，可在 macOS、Linux 或視窗上安裝和設定。
- Python 3.7 或更高版本。
- Boto3, 安裝和配置. 如果 Boto3 尚未安裝，請執行`python -m pip install boto3`指令以進行安裝。

架構

技術, 堆

- Amazon DynamoDB
- Amazon DynamoDB 串流
- Amazon 數據 Firehose
- AWS Lambda
- Amazon S3

1. TTL 會刪除項目。
2. DynamoDB 串流觸發程序會叫用 Lambda 串流處理器函數。
3. Lambda 函數會以批次格式將記錄放入 Firehose 交付串流中。
4. 資料記錄會存檔在 S3 儲存貯體中。

工具

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是管理 AWS 服務的統一工具。
- 亞馬遜 DynamoDB — Amazon DynamoDB 是一個鍵值和文件資料庫，可在任何規模下提供 10 毫秒的效能。
- Amazon DynamoDB 存留時間 (TTL) — Amazon DynamoDB TTL 可協助您定義每個項目的時間戳記，以確定何時不再需要某個項目。
- Amazon DynamoDB 串流 — Amazon Dynam oDB 串流會在任何 DynamoDB 表格中擷取一系列按時間排序的項目層級修改，並將此資訊儲存在日誌中長達 24 小時。

- Amazon 資料 Firehose — Amazon 資料 Firehose 是可靠地將串流資料載入資料湖、資料存放區和分析服務的最簡單方法。
- AWS Lambda — AWS Lambda 可執行程式碼，無需佈建或管理伺服器。您只需為使用的運算時間支付費用。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。

Code

此模式的程式碼可在使用 DynamoDB TTL 儲 GitHub 存庫將項目存檔至 S3 中取得。

史诗

設定 DynamoDB 料表、TTL 和 DynamoDB 串流

任務	描述	所需技能
建立 DynamoDB 資料表。	使用 AWS CLI 在 DynamoDB 中建立一個名為的表格。Reservation 選擇隨機讀取容量單位 (RCU) 和寫入容量單位 (WCU)，並為您的表格指定兩個屬性：ReservationID 和。ReservationDate `aws dynamodb create-table \ --table-name Reservation \ --attribute-definitions AttributeName=ReservationID,AttributeType=S AttributeName=ReservationDate,AttributeType=N \ --key-schema AttributeName=ReservationID`	雲架構師、應用程式開發

任務	描述	所需技能
	`` ,KeyType=HASH AttributeName=Rese rvationDate,KeyTyp e=RANGE \ --provisioned-th roughput ReadCapac ityUnits=100,Write CapacityUnits=100 ``` `ReservationDate` 是將用來開啟 TTL 的時間戳記。	
開啟 DynamoDB 資料庫。	使用 AWS CLI 為屬性開啟 DynamoDB 資料庫 TTL。`ReservationDate` ``` aws dynamodb update-ti me-to-live \ --table-name Reservati on\ --time-to-live-spe cification Enabled=t rue,AttributeName= ReservationDate ```	雲架構師、應用程式開發

任務	描述	所需技能
開啟動 DynamoDB 料流。	使用 AWS CLI 透過使用串流類型為Reservation 表格開啟 DynamoDB NEW_AND_OLD_IMAGES 串流。 `aws dynamodb update-table \ --table-name Reservation \ --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES` 此資料流會包含 TTL 所刪除之新項目、更新項目、刪除的項目和項目的記錄。TTL 刪除之項目的記錄包含一個額外的中繼資料屬性，以區分它們與手動刪除的項目。用於 TTL 刪除的userIdentity 欄位表示 DynamoDB 服務已執行刪除動作。 在此模式中，只有 TTL 刪除的項目會封存，但您只能封存存在REMOVE和userIdentity 包含principalId 等於的記錄。eventName dynamodb.amazonaws.com	雲架構師、應用程式開發

建立和設定 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	使用 AWS CLI 在您的 AWS 區域建立目的地 S3 儲存貯體，並以您`us-east-1` 的區域取代。 `aws s3api create-bucket \` `--bucket reservationfirehosedestinationbucket \` `--region us-east-1` 請確定 S3 儲存貯體的名稱是全域唯一的，因為該命名空間是由所有 AWS 帳戶共用的。	雲架構師、應用程式開發
為 S3 儲存貯體建立 30 天的生命週期政策。	1. 登入 AWS 管理主控台並開啟 Amazon S3 主控台。 2. 選擇包含來自 Firehose 資料的 S3 儲存貯體。 3. 在 S3 儲存貯體中，選擇管理索引標籤，然後選擇新增生命週期規則。 4. 在「生命週期規則」對話方塊中輸入規則的名稱，並為值區設定 30 天的生命週期規則。	雲架構師、應用程式開發

建立 Firehose 交付串流

任務	描述	所需技能
建立並設定 Firehose 傳送串流。	從 GitHub 儲存庫下載並編輯`CreateFireHoseToS3.py` 程式碼範例。 此程式碼以 Python 撰寫，並示範如何建立 Firehose 交付串流和 AWS Identity and Access Management (IAM) 角色。IAM 角色將具有可供 Firehose 用來寫入目標 S3 儲存貯體的政策。 若要執行指令碼，請使用下列命令和命令列引數。 引數 1=<Your_S3_bucket_ARN> ，這是您先前建立的儲存貯體的 Amazon 資源名稱 (ARN) 引數 2= 您的 Firehose 名稱（此飛行員正在使`firehose_to_s3_stream` 用。） 引數 3= 您的 IAM 角色名稱 (此試行方案正在使用`firehose_to_s3` .) `python CreateFireHoseToS3.py <Your_S3_Bucket_ARN> firehose_to_s3_stream firehose_to_s3`	雲架構師、應用程式開發

任務	描述	所需技能
	如果指定的 IAM 角色不存在，指令碼會建立具有受信任關係政策的假設角色，以及授予足夠 Amazon S3 權限的政策。如需這些原則的範例，請參閱其他資訊一節。	
驗證「Firehose」交付串流。	使用 AWS CLI 來確認交付串流是否已成功建立，以描述 Firehose 交付串流。 `aws firehose describe-delivery-stream --delivery-stream-name firehose_to_s3_stream`	雲架構師、應用程式開發

建立 Lambda 函數以處理 Firehose 交付串流

任務	描述	所需技能
為 Lambda 函數建立信任政策。	使用下列資訊建立信任原則檔案。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" },`	雲架構師、應用程式開發

任務	描述	所需技能
	<pre>"Action": "sts:AssumeRole" }] }</pre> 這會提供您的函數存取 AWS 資源的權限。	
建立 Lambda 函數的執行角色。	若要建立執行角色，請執行下列程式碼。 <pre>aws iam create-role --role-name lambda- ex --assume-role-poli cy-document file://Tr ustPolicy.json</pre>	雲架構師、應用程式開發

任務	描述	所需技能
將權限新增至角色。	若要將權限新增至角色，請使用attach-policy-to-role 指令。 `aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` `aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole` `aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/AmazonKinesisFirehoseFullAccess` `aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/IAMFullAccess`	雲架構師、應用程式開發

任務	描述	所需技能
建立 Lambda 函數。	通過運行以下命令壓縮從代碼存儲庫中的LambdaStreamProcessor.py 文件。 `zip function.zip LambdaStreamProcessor.py` 當您建立 Lambda 函數時，您將需要 Lambda 執行角色 ARN。要獲取 ARN，請運行以下代碼。 `aws iam get-role \ --role-name lambda-ex` 若要建立 Lambda 函數，請執行下列程式碼。 `aws lambda create-function --function-name LambdaStreamProcessor \ --zip-file fileb://function.zip --handler LambdaStreamProcessor.handler --runtime python3.8 \ --role {Your Lamda Execution Role ARN}\ --environment Variables="{firehose_name=firehose_to_s3_stream,bucket_arn = arn:aws:s3:::reservationfirehosedestinationbucket,iam_role_name`	雲架構師、應用程式開發

任務	描述	所需技能
	`= firehose_to_s3, batch_size=400}"`	
設定 Lambda 函數觸發器。	使用 AWS CLI 設定觸發器 (DynamoDB Streams)，此觸發器會叫用 Lambda 函數。400 的批次大小是為了避免遇到 Lambda 並行發生問題。 `aws lambda create-event-source-mapping --function-name LambdaStreamProcessor \ --batch-size 400 --starting-position LATEST \ --event-source-arn <Your Latest Stream ARN From DynamoDB Console>`	雲架構師、應用程式開發

測試功能

任務	描述	所需技能
將具有過期時間戳記的項目新增至「保留區」表格。	若要測試功能，請將具有過期紀元時間戳記的項目新增至資料表`Reservation` 。TTL 會根據時間戳記自動刪除項目。 Lambda 函數會在 DynamoDB 串流活動時啟動，並篩選事件以識別`REMOVE`活動或已刪除的項目。然後，它會以批次格式將記錄放入 Firehose 傳送串流中。	雲端架構師

任務	描述	所需技能
	Firehose 交付串流會將項目傳輸到具有`firehoseto s3example/year=cu rrent year/mont h=current month/ day=current day/hour= current hour/` 前置詞的目的地 S3 儲存貯體。 重要：若要優化資料擷取，請使用「其他資訊」一節中詳細`ErrorOutputPrefix` 說明的`Prefix`和來設定 Amazon S3。	

清理資源

任務	描述	所需技能
刪除所有資源。	刪除所有資源，以確保您不會為未使用的任何服務付費。	雲架構師、應用程式開發

相關資源

- 管理儲存生命週期
- Amazon S3 存儲類
- 適用於 Python 的 AWS 開發套件文件

其他資訊

建立並設定 Firehose 傳送串流 — 政策範例

Firehose 信任關係政策範例文件

```
firehose_assume_role = {
        'Version': '2012-10-17',
        'Statement': [
            {
                'Sid': '',
                'Effect': 'Allow',
                'Principal': {
                    'Service': 'firehose.amazonaws.com'
                },
                'Action': 'sts:AssumeRole'
            }
        ]
    }
```

S3 許可政策範例

```
s3_access = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Action": [
                    "s3:AbortMultipartUpload",
                    "s3:GetBucketLocation",
                    "s3:GetObject",
                    "s3:ListBucket",
                    "s3:ListBucketMultipartUploads",
                    "s3:PutObject"
                ],
                "Resource": [
                    "{your s3_bucket ARN}/*",
                    "{Your s3 bucket ARN}"
                ]
            }
        ]
    }
```

測試功能 — Amazon S3 組態

選擇具有下列項`Prefix`目的 Amazon S3 組態來優化資料擷取。`ErrorOutputPrefix`

prefix

```
firehosetos3example/year=! {timestamp: yyyy}/month=! {timestamp:MM}/day=!
 {timestamp:dd}/hour=!{timestamp:HH}/
```

Firehose 首先創建一個firehosetos3example直接在 S3 存儲桶下面調用的基本文件夾。然後！{timestamp:yyyy}!{timestamp:MM}，!{timestamp:dd}它會使用 Java DateTimeFormatter格式評估運算式、、和年、月、日和小時。!{timestamp:HH}

例如，在 Unix 紀元時間中，近似到達時間戳記 1604683577 會評估為year=2020、、和。month=11 day=06 hour=05因此，Amazon S3 中資料記錄交付的位置會評估為firehosetos3example/year=2020/month=11/day=06/hour=05/。

ErrorOutputPrefix

```
firehosetos3erroroutputbase/!{firehose:random-string}/!{firehose:error-output-type}/!
{timestamp:yyyy/MM/dd}/
```

ErrorOutputPrefix產生firehosetos3erroroutputbase直接在 S3 儲存貯體下方呼叫的基本資料夾。運!{firehose:random-string}算式會評估為 11 個字元的隨機字串，例如ztWxkdg3Thg。交付失敗記錄的 Amazon S3 物件的位置可以評估到firehosetos3erroroutputbase/ztWxkdg3Thg/processing-failed/2020/11/06/。

使用 Amazon EC2 Auto Scaling 和 Systems Manager 建置微焦點企業伺服器 PAC

創建者：凱文容（AWS），彼得·伍茲（微焦點），亞伯拉罕·朗登（微焦點）和克里希卡·帕拉尼·塞爾瓦姆（AWS）

環境：生產	技術：現代化；雲端原生；DevOps基礎架構

Summary

此模式為大型主機應用程式引入可擴展架構，在橫向擴充效能和可用性叢集 (PAC) 中使用 Micro Focus 企業伺服器，以及 Amazon Web 服務 (AWS) 上的亞馬遜彈性運算雲端 (Amazon EC2) Auto Scaling 展群組。使用 AWS Systems Manager 和 Amazon EC2 自動擴展生命週期掛鉤，完全自動化解決方案。藉由使用此模式，您可以設定大型主機線上和批次處理應用程式，根據您的容量需求自動進出擴充，以達到高彈性。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 微焦點企業服務器軟件和許可證。如需詳細資訊，請聯絡微焦點銷售人員
- 瞭解重建和提供要在 Micro Focus 企業伺服器中執行的大型主機應用程式的概念。如需高階概觀，請參閱 Micro Focus 企業伺服器資料表。
- 瞭解 Micro Focus 企業伺服器向外延展效能和可用性叢集中的概念。如需詳細資訊，請參閱 Micro Focus 企業伺服器說明文件。
- 瞭解 DevOps 具有持續整合 (CI) 的大型主機應用程式的整體概念。如需 AWS 和 Micro Focus 開發的 AWS Prescriptive Guidance 模式，請參閱大型主機現代化： DevOps 在使用 Micro Focus 的 AWS 上。

限制

- 如需 Micro Focus 企業伺服器支援的平台清單，請參閱 Micro Focus 企業伺服器資料表。

- 此模式中使用的指令碼和測試是以 Amazon EC2 視窗伺服器 2019 年為基礎；其他 Windows 伺服器版本和作業系統並未針對此模式進行測試。
- 此病毒碼是以適用於 Windows 的 Micro Focus 企業伺服器 6.0 為基礎；舊版或更新版本並未在此病毒碼的開發過程中進行測試。

產品版本

- 微焦點企業伺服器 6.0
- Windows Server 2019

架構

在傳統的大型主機環境中，您必須佈建硬體來託管應用程式和公司資料。為了滿足季節性、每月、每季甚至前所未有或意外的需求，大型主機使用者必須購買額外的儲存裝置和運算容量來擴充規模。增加儲存體和運算容量資源的數量可改善整體效能，但擴展不是線性的。

當您開始使用 Amazon EC2 Auto Scaling 和微焦點企業伺服器在 AWS 上採用隨需消費模式時，情況並非如此。以下各節詳細說明如何使用 Micro Focus 企業伺服器向外擴充效能和可用性叢集 (PAC) 搭配 Amazon EC2 Auto Scaling 群組，建立完全自動化、可擴展的大型主機應用程式架構。

微焦點企業伺服器自動擴充架構

首先，了解 Micro Focus 企業伺服器的基本概念非常重要。此環境為傳統在 IBM 大型主機上執行的應用程式提供與大型主機相容的 x86 部署環境。它提供線上和批次執行，以及支援下列項目的交易環境：

- IBM 聯盟
- IBM PL/I
- 批次工作
- IBM CICS 和 IMS TM 交易
- Web 服務
- 常見的批次工具，包括 SORT

Micro Focus 企業伺服器可讓大型主機應用程式以最少的變更執行。現有的大型主機工作負載可移至 x86 平台並進行現代化，以利用 AWS 雲端原生擴充功能快速擴展至新市場或地理位置。

AWS Prescriptive Guidance 模式大型主機現代化： DevOps 在使用 Micro Focus 的 AWS 上引入了架構，使用 Micro Focus 企業開發人員和企業測試伺服器搭配 AWS 和 AWS，加速 AWS 上大型主機應用程式的開發和測試。 CodePipeline CodeBuild此模式著重於將大型主機應用程式部署到 AWS 生產環境，以達到高可用性和彈性。

在大型主機生產環境中，您可能已經在大型主機中設定 IBM Parallel Sysplex，以達到高效能和高可用性。為了建立類似於 Sysplex 的向外延展架構，Micro Focus 將效能與可用性叢集 (PAC) 引入企業伺服器。PAC 支援將大型主機應用程式部署到以單一映像形式管理的多個企業伺服器區域，並在 Amazon EC2 執行個體中向外擴充。PAC 也可視需求支援可預測的應用程式效能和系統輸送量。

在 PAC 中，多個企業伺服器執行個體一起運作，做為單一邏輯實體。因此，一個企業伺服器執行個體故障不會中斷業務連續性，因為容量會與其他區域共用，而使用產業標準功能 (例如 Amazon EC2 Auto Scaling 群組) 自動啟動新執行個體。這消除了單點故障，提高了硬件，網絡和應用程序問題的恢復能力。向外延展的企業伺服器執行個體可以使用企業伺服器通用 Web 管理 (ESCWA) API 來操作和管理，簡化企業伺服器的作業維護和可維護性。

注意：Micro Focus 建議「效能和可用性叢集」(PAC) 至少應包含三個企業伺服器區域，以便在企業伺服器區域發生故障或需要維護時，可用性不會受到影響。

PAC 組態需要支援的關聯式資料庫管理服務 (RDBMS) 來管理區域資料庫、跨區域資料庫和選用的資料倉庫資料庫。資料存放區資料庫應該用來管理虛擬儲存存取方法 (VSAM) 檔案，使用 Micro Focus 資料庫檔案處理常式支援來改善可用性和延展性。支援的 RDBMS 包括下列項目：

- Microsoft SQL 服務器 2009 R2 及更高版本
- 10.x PostgreSQL，包括 Amazon Aurora 兼容版
- DB2 10.4 及更新版本

如需支援的 RDBMS 和 PAC 需求的詳細資訊，請參閱微焦點企業伺服器-必要條件和微焦點企業伺服器-建議的 PAC 組態。

下圖顯示微型聚焦 PAC 的典型 AWS 架構設定。

	元件	Description
1	企業伺服器執行個體自動調整	設定 PAC 中使用企業伺服器執行個體部署的自動調整資源調整群組。Amazon CloudWatc

		h 警示可以使用指標向外擴充或啟動執行個體數 CloudWatch 量。
2	企業伺服器 ESCWA 執行個體自動調整資源組	設定使用企業伺服器通用網頁管理 (ESCWA) 部署的自動調整資源調整群組。ESCWA 提供叢集管理 API。 在企業伺服器執行個體自動調整規模事件期間，ESCWA 伺服器充當控制平面，以新增或移除企業伺服器，以及啟動或停止 PAC 中的企業伺服器區域。 由於 ESCWA 執行個體僅用於 PAC 管理，因此其流量模式是可預測的，並且其自動調整所需容量需求可設定為 1。
3	異地同步備份設定中的 Amazon Aurora 執	設定關聯式資料庫管理系統 (RDBMS)，以主控要跨企業伺服器執行個體共用的使用者和系統資料檔案。
4	Amazon ElastiCache 適用於 Redis 的實例和複本	設定 ElastiCache Redis 主執行個體和至少一個複本來託管使用者資料，並做為企業伺服器執行個體的向外延展存放庫 (SOR)。您可以設定一或多個向外延展儲存庫來儲存特定類型的使用者資料。 企業服務器使用 Redis 的 NoSQL 數據庫作為 SOR，這是維護 PAC 完整性的要求。

5	Network Load Balancer	設定負載平衡器，提供應用程式連線至 Enterprise Server 執行個體所提供之服務的主機名稱 (例如，透過 3270 模擬器存取應用程式)。

這些元件構成微焦點企業伺服器 PAC 叢集的最低需求。下一節將說明叢集管理自動化。

使用 AWS Systems Manager Automation 進行擴展

在 AWS 上部署 PAC 叢集之後，PAC 就會透過企業伺服器通用網頁管理 (ESCWA) API 進行管理。

若要在自動擴展事件期間自動執行叢集管理任務，您可以使用 Systems Manager 自動化手冊和 Amazon EC2 Auto Scaling 搭配 Amazon EventBridge。這些自動化的體系結構顯示在下圖中。

	元件	Description
1	自動調整生命週期鉤	設定自動擴展生命週期勾點，並在啟動新執行個體且現有執行個體在自動擴展群組中終止 EventBridge 時傳送通知給 Amazon。
2	Amazon EventBridge	設定 Amazon EventBridge 規則，將自動擴展事件路由到 Systems Manager 自動化工作流程簿目標。
3	自動化手冊	設定 Systems Manager 自動化手冊以執行 Windows PowerShell 指令碼，並呼叫 ESCWA API 來管理 PAC。如需範例，請參閱其他資訊一節。
4	自動調度資源群組中的企業伺服器 ESCWA 執行個體	在自動調整資源調整群組中設定企業伺服器 ESCWA 執行個

體。ESCWA 執行個體提供用於管理 PAC 的 API。

工具

- Micro Focus 企業伺服器 — Micro Focus 企業伺服器為使用企業開發人員的任何整合式開發環境 (IDE) 變體所建立的應用程式提供執行環境。
- Amazon EC2 Auto Scaling — Amazon EC2 Auto Scaling 可協助您確保擁有正確數量的 Amazon EC2 執行個體可用來處理應用程式的負載。您可以建立 EC2 執行個體的集合 (稱為 Auto Scaling 群組)，並指定執行個體的最小和最大數量。
- Amazon ElastiCache for Redis — Amazon ElastiCache 是一種網路服務，用於在雲端中設定、管理和擴展分散式記憶體內資料存放區或快取環境。它提供高效能、可擴展且符合成本效益的快取解決方案。
- Amazon RDS — Amazon Relational Database Service 服務 (Amazon RDS) 是一種網路服務，可讓您更輕鬆地在 AWS 雲端中設定、操作和擴展關聯式資料庫。它為關聯式資料庫提供符合成本效益且可調整大小的容量，並管理常見的資料庫管理工作。
- AWS Systems Manager — AWS Systems Manager 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。您可以使用 Systems Manager 主控台檢視來自多個 AWS 服務的操作資料，並自動化 AWS 資源的操作任務。Systems Manager 透過掃描您的受管執行個體並報告 (或採取修正動作) 其偵測的任何政策違規，協助您保持安全與合規。

史诗

創建一個 Amazon Aurora 實例

任務	描述	所需技能
為 Amazon Aurora 執行個體建立 AWS CloudFormation 範本。	使用 AWS 範例程式碼片段建立範 CloudFormation 本，以建立與 Amazon Aurora PostgreSQL 相容的版本執行個體。	雲端架構師
部署 CloudFormation 堆疊以建立 Amazon Aurora 執行個體。	使用此 CloudFormation 範本建立 Aurora PostgreSQL 相容的	雲端架構師

任務	描述	所需技能
	執行個體，該執行個體已針對生產工作負載啟用異地同步備份複寫。	
設定企業伺服器的資料庫連線設定。	請遵循 Micro Focus 文件中的指示，為 Micro Focus 企業伺服器準備連線字串和資料庫組態。	數據工程師， DevOps 工程師

為 Redis 實例創建一個 Amazon ElastiCache 集群

任務	描述	所需技能
為 Redis 執行個體的 Amazon ElastiCache 叢集建立 CloudFormation 範本。	使用 AWS 範例程式碼片段建立範 CloudFormation 本，以便為 Redis 執行個體建立 Amazon ElastiCache 叢集。	雲端架構師
部署 CloudFormation 堆疊以建立 Redis 執行個體的 Amazon ElastiCache 叢集。	針對已針對生產工作負載啟用異地同步備份複寫的 Redis 執行個體建立 Amazon ElastiCache 叢集。	雲端架構師
設定企業伺服器 PSOR 連線設定。	遵循微焦點文件中的指示，準備微焦點企業伺服器 PAC 的 PAC 向外延展儲存庫 (PSOR) 連線組態。	DevOps 工程師

建立微焦點企業伺服器 ESCWA 自動調整資源群組

任務	描述	所需技能
創建微焦點企業服務器 AMI。	建立 Amazon EC2 Windows 伺服器執行個體，並在 EC2 執行個體中安裝微焦點企業伺服	雲端架構師

任務	描述	所需技能
	器二進位檔案。創建 EC2 實例的 Amazon 機器映像（AMI）。如需詳細資訊，請參閱企業伺服器安裝說明文件。	
建立企業伺服器 ESCWA 的 CloudFormation 範本。	使用 AWS 範例程式碼片段建立範本，以便在自動擴展群組中建立自訂企業伺服器 ESCWA 堆疊。	雲端架構師
部署 CloudFormation 堆疊以建立適用於企業伺服器 ESCWA 的 Amazon EC2 擴展群組。	使用 CloudFormation 範本來部署自動調整資源群組與微型焦點企業伺服器 ESCWA AMI 在先前的故事中建立。	雲端架構師

建立 AWS Systems Manager Automation 手冊

任務	描述	所需技能
建立 Systems Manager 自動化手冊的 CloudFormation 範本。	使用 [其他資訊] 區段中的範例程式碼片段，建立可建立 Systems Manager 自動化執行手冊的 CloudFormation 範本，以自動建立 PAC 建立、企業伺服器擴充，以及企業伺服器向外延展。	雲端架構師
部署包含 Systems Manager 自動化手冊的 CloudFormation 堆疊。	使用此 CloudFormation 範本部署包含用於建立 PAC、企業伺服器擴充以及企業伺服器向外擴充的自動化 Runbook 的堆疊。	雲端架構師

為 Micro Focus 企業伺服器建立自動調整規模群組

任務	描述	所需技能
建立 CloudFormation 範本，以便為 Micro Focus 企業伺服器設定自動調度資源調度群組。	使用 AWS 範例程式碼片段建立可建立自動擴展群組的範 CloudFormation 本。此範本將重複使用為微焦點企業伺服器 ESCWA 執行個體所建立的相同 AMI。 然後使用 AWS 範例程式碼片段建立自動擴展生命週期事件，並設 EventBridge 定 Amazon 篩選相同範本中的向外擴充和擴充事件。CloudFormation	雲端架構師
為 Micro Focus 企業伺服器部署自動調整資源調整群組的 CloudFormation 堆疊。	部署包含 Micro Focus 企業伺服器之自動調整資源調度群組的 CloudFormation 堆疊。	雲端架構師

相關資源

- 微焦點企業伺服器效能與可用性叢集 (PAC)
- Amazon EC2 Auto Scaling 生命週期鉤
- 使用觸發器執行自動化 EventBridge

其他資訊

必須自動執行下列案例，才能擴展或向外擴充 PAC 叢集。

啟動或重新建立 PAC 的自動化

在 PAC 叢集開始時，企業伺服器會要求 ESCWA 呼叫 API 來建立 PAC 組態。這會啟動並將企業伺服器區域新增至 PAC。若要建立或重新建立 PAC，請使用下列步驟：

1. 在 ESCWA 中以指定的名稱設定 PAC 向外延展儲存庫 (PSOR)。

```
POST /server/v1/config/groups/sors
```

2. 建立具有指定名稱的 PAC，並將 PSOR 貼附至其上。

```
POST /server/v1/config/groups/pacs
```

3. 如果這是您第一次設定 PAC，請設定區域資料庫和跨區域資料庫。

   注意：此步驟使用 SQL 查詢和微焦點企業套件命令行 dbhfhadmin 工具來建立資料庫並匯入初始資料。

4. 將 PAC 定義安裝到企業伺服器區域中。

```
POST /server/v1/config/mfds
POST /native/v1/config/groups/pacs/${pac_uid}/install
```

5. 啟動 PAC 中的「企業伺服器」區域。

```
POST /native/v1/regions/${host_ip}/${port}/${region_name}/start
```

前面的步驟可以通過使用 Windows PowerShell 腳本來實現。

下列步驟說明如何透過重複使用 Windows PowerShell 指令碼來建立 PAC 的自動化操作。

1. 建立 Amazon EC2 啟動範本，以便在啟動程序中下載或建立 Windows PowerShell 指令碼。例如，您可以使用 EC2 使用者資料從 Amazon Simple Storage Service (Amazon S3) 貯體下載指令碼。
2. 建立 AWS Systems Manager Automation 執行手冊以叫用 Windows 指 PowerShell 令碼。
3. 使用執行個體標籤，將執行手冊與 ESCWA 執行個體相關聯。
4. 使用啟動範本建立 ESCWA 自動調度資源調度群組。

您可以使用下列範例 AWS CloudFormation 程式碼片段來建立自動化工作流程簿。

用於建立 PAC 的 Systems Manager 自動化工作流程簿範例 CloudFormation 程式碼片段

```
PACInitDocument:
   Type: AWS::SSM::Document
   Properties:
     DocumentType: Command
```

```
      Content:
        schemaVersion: '2.2'
        description: Operation Runbook to create Enterprise Server PAC
        mainSteps:
        - action: aws:runPowerShellScript
          name: CreatePAC
          inputs:
            onFailure: Abort
            timeoutSeconds: "1200"
            runCommand:
            - |
              C:\Scripts\PAC-Init.ps1
  PacInitAutomation:
     Type: AWS::SSM::Document
     Properties:
       DocumentType: Automation
       Content:
         description: Prepare Micro Focus PAC Cluster via ESCWA Server
         schemaVersion: '0.3'
         assumeRole: !GetAtt SsmAssumeRole.Arn
         mainSteps:
           - name: RunPACInitDocument
             action: aws:runCommand
             timeoutSeconds: 300
             onFailure: Abort
             inputs:
               DocumentName: !Ref PACInitDocument
               Targets:
                 - Key: tag:Enterprise Server - ESCWA
                   Values:
                     - "true"
   PacInitDocumentAssociation:
     Type: AWS::SSM::Association
     Properties:
       DocumentVersion: "$LATEST"
       Name: !Ref PACInitDocument
       Targets:
         - Key: tag:Enterprise Server - ESCWA
           Values:
             - "true"
```

如需詳細資訊，請參閱微焦點企業伺服器-設定 PAC。

使用新的企業伺服器執行個體自動化擴充

當企業伺服器執行個體向外延展時，必須將其「企業伺服器」區域新增至 PAC。下列步驟說明如何呼叫 ESCWA API，並將企業伺服器區域新增至 PAC。

1. 將 PAC 定義安裝到企業伺服器區域中。

```
POST '/server/v1/config/mfds'
POST /native/v1/config/groups/pacs/${pac_uid}/install
```

2. 溫啟動 PAC 中的區域。

```
POST /native/v1/regions/${host_ip}/${port}/${region_name}/start
```

3. 將自動調整資源調度群組與負載平衡器建立關聯，將 Enterprise Server 執行個體新增至負載平衡器。

前面的步驟可以通過使用 Windows PowerShell 腳本來實現。如需詳細資訊，請參閱微焦點企業伺服器-設定 PAC。

您可以使用下列步驟來建置事件導向的自動化，藉由重複使用 Windows PowerShell 指令碼，將新啟動的企業伺服器執行個體新增至 PAC。

1. 為企業伺服器執行個體建立 Amazon EC2 啟動範本，以便在啟動期間佈建企業伺服器區域。例如，您可以使用 Micro Focus 企業伺服器指令 MFD 匯入區域組態。如需此命令可用的詳細資訊和選項，請參閱企業伺服器參考。
2. 建立使用在上一個步驟中建立的啟動範本的「企業伺服器自動調整資源」群組。
3. 建立 Systems Manager 自動化執行手冊來叫用 Windows 指 PowerShell 令碼。
4. 使用執行個體標籤，將執行手冊與 ESCWA 執行個體相關聯。
5. 建立 Amazon EventBridge 規則以篩選企業伺服器自動擴展群組的 EC2 執行個體啟動成功事件，並建立目標以使用自動化執行手冊。

您可以使用下列範例 CloudFormation 程式碼片段來建立自動化工作流程簿和 EventBridge 規則。

用於向外擴充企業伺服器執行個體的 Systems Manager 範例 CloudFormation 程式

```
ScaleOutDocument:
   Type: AWS::SSM::Document
   Properties:
     DocumentType: Command
     Content:
```

```
        schemaVersion: '2.2'
        description: Operation Runbook to Adding MFDS Server into an existing PAC
        parameters:
          MfdsPort:
            type: String
          InstanceIpAddress:
            type: String
            default: "Not-Available"
          InstanceId:
            type: String
            default: "Not-Available"
        mainSteps:
        - action: aws:runPowerShellScript
          name: Add_MFDS
          inputs:
            onFailure: Abort
            timeoutSeconds: "300"
            runCommand:
            - |
              $ip = "{{InstanceIpAddress}}"
              if ( ${ip} -eq "Not-Available" ) {
                $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
              }
              C:\Scripts\Scale-Out.ps1 -host_ip ${ip} -port {{MfdsPort}}

  PacScaleOutAutomation:
    Type: AWS::SSM::Document
    Properties:
      DocumentType: Automation
      Content:
        parameters:
          MfdsPort:
            type: String
          InstanceIpAddress:
            type: String
            default: "Not-Available"
          InstanceId:
            type: String
            default: "Not-Available"
        description: Scale Out 1 New Server in Micro Focus PAC Cluster via ESCWA
Server
        schemaVersion: '0.3'
        assumeRole: !GetAtt SsmAssumeRole.Arn
```

```
    mainSteps:
      - name: RunScaleOutCommand
        action: aws:runCommand
        timeoutSeconds: 300
        onFailure: Abort
        inputs:
          DocumentName: !Ref ScaleOutDocument
          Parameters:
            InstanceIpAddress: "{{InstanceIpAddress}}"
            InstanceId: "{{InstanceId}}"
            MfdsPort: "{{MfdsPort}}"
          Targets:
            - Key: tag:Enterprise Server - ESCWA
              Values:
                - "true"
```

企業伺服器執行個體中的自動化調整

與向外擴充類似，當擴充企業伺服器執行個體時，會啟動 EC2 執行個體終止生命週期動作事件，並且需要執行下列程序和 API 呼叫才能從 PAC 移除 Micro Focus 企業伺服器執行個體。

1. 停止終止「企業伺服器」執行處理中的區域。

```
POST "/native/v1/regions/${host_ip}/${port}/${region_name}/stop"
```

2. 從 PAC 移除企業伺服器實例。

```
DELETE "/server/v1/config/mfds/${uid}"
```

3. 傳送訊號以繼續終止企業伺服器執行個體。

前面的步驟可以在 Windows PowerShell 腳本中實現。如需此程序的其他詳細資訊，請參閱 Micro Focus 企業伺服器文件-管理 PAC。

下列步驟說明如何透過重複使用 Windows 指令碼來建置事件導向的自動化，以終止 PAC 的企業伺服器執行個體。 PowerShell

1. 建立 Systems Manager 自動化執行手冊來叫用 Windows 指 PowerShell 令碼。
2. 使用執行個體標籤，將執行手冊與 ESCWA 執行個體相關聯。
3. 為 EC2 執行個體終止建立自動調整規模群組生命週期勾點。

4. 建立 Amazon EventBridge 規則以篩選企業伺服器自動擴展群組的 EC2 執行個體終止生命週期動作事件，並建立目標以使用自動化執行手冊。

您可以使用下列範例 CloudFormation 範本來建立 Systems Manager 自動化工作流程簿、生命週期勾點和 EventBridge 規則。

用於在企業伺服器執行個體中調整的 Systems Manager 自動化工作流程簿的範例 CloudFormation 程式碼

```
 ScaleInDocument:
    Type: AWS::SSM::Document
    Properties:
      DocumentType: Command
      Content:
        schemaVersion: '2.2'
        description: Operation Runbook to Remove MFDS Server from PAC
        parameters:
          MfdsPort:
            type: String
          InstanceIpAddress:
            type: String
            default: "Not-Available"
          InstanceId:
            type: String
            default: "Not-Available"
        mainSteps:
        - action: aws:runPowerShellScript
          name: Remove_MFDS
          inputs:
            onFailure: Abort
            runCommand:
            - |
              $ip = "{{InstanceIpAddress}}"
              if ( ${ip} -eq "Not-Available" ) {
                $ip = aws ec2 describe-instances --instance-id {{InstanceId}} --output
text --query "Reservations[0].Instances[0].PrivateIpAddress"
              }
              C:\Scripts\Scale-In.ps1 -host_ip ${ip} -port {{MfdsPort}}

  PacScaleInAutomation:
    Type: AWS::SSM::Document
    Properties:
```

```
        DocumentType: Automation
        Content:
          parameters:
            MfdsPort:
              type: String
            InstanceIpAddress:
              type: String
              default: "Not-Available"
            InstanceId:
              type: String
              default: "Not-Available"
          description: Scale In 1 New Server in Micro Focus PAC Cluster via ESCWA Server
          schemaVersion: '0.3'
          assumeRole: !GetAtt SsmAssumeRole.Arn
          mainSteps:
            - name: RunScaleInCommand
              action: aws:runCommand
              timeoutSeconds: "600"
              onFailure: Abort
              inputs:
                DocumentName: !Ref ScaleInDocument
                Parameters:
                  InstanceIpAddress: "{{InstanceIpAddress}}"
                  MfdsPort: "{{MfdsPort}}"
                  InstanceId: "{{InstanceId}}"
                Targets:
                  - Key: tag:Enterprise Server - ESCWA
                    Values:
                      - "true"
            - name: TerminateTheInstance
              action: aws:executeAwsApi
              inputs:
                Service: autoscaling
                Api: CompleteLifecycleAction
                AutoScalingGroupName: !Ref AutoScalingGroup
                InstanceId: "{{ InstanceId }}"
                LifecycleActionResult: CONTINUE
                LifecycleHookName: !Ref ScaleInLifeCycleHook
```

Amazon EC2 自動擴展觸發器的自動化

為企業伺服器執行個體設定擴展政策的程序需要瞭解應用程式行為。在大多數情況下，您可以設定目標追蹤擴展政策。例如，您可以使用平均 CPU 使用率做為自動擴展政策設定的 Amazon CloudWatch 指

標。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling 的目標追蹤擴展政策。對於具有常規流量模式的應用程式，請考慮使用預測性擴展政策。如需詳細資訊，請參閱 Amazon EC2 自動擴展的預測性擴展。

在 Amazon 服務中建立多租戶無伺服器架構 OpenSearch

由虎斑病房（AWS）和尼莎甘比希爾（AWS）創建

環境：PoC 或試點	技術：現代化、SaaS、無伺服器	工作負載：開源
AWS 服務：Amazon OpenSearch 服務；AWS Lambda；Amazon S3；Amazon API Gateway		

Summary

Amazon OpenSearch 服務是一種受管服務，可讓您輕鬆部署、操作和擴展 Elasticsearch，這是一種熱門的開放原始碼搜尋和分析引擎。Amazon Ser OpenSearch vice 提供自由文字搜尋以及近乎即時的擷取和儀表板，以供串流資料 (例如日誌和指標) 使用。

軟體即服務 (SaaS) 供應商經常使用 Amazon Ser OpenSearch vice 來解決各種使用案例，例如以可擴展且安全的方式取得客戶洞察，同時減少複雜性和停機時間。

在多租戶環境中使用 Amazon Ser OpenSearch vice 會引入一系列考量，這些考量會影響 SaaS 解決方案的分割、隔離、部署和管理。SaaS 提供者必須考慮如何在不斷變化的工作負載的情況下，有效地擴展其 Elasticsearch 叢集。他們還需要考慮分層和嘈雜的鄰居條件如何影響其分區模型。

此模式會檢閱使用 Elasticsearch 建構來表示和隔離租用戶資料的模型。此外，該模式著重於簡單的無伺服器參考架構作為範例，以示範在多租戶環境中使用 Amazon Ser OpenSearch vice 進行索引和搜尋。它實作集區資料分割模型，該模型會在所有租用戶之間共用相同的索引，同時維護租用戶的資料隔離。此模式使用以下 Amazon Web Services（AWS）服務：Amazon API Gateway，AWS Lambda，Amazon Simple Storage Service (Amazon S3) 和 Amazon OpenSearch 服務。

如需集區模型和其他資料分割模型的詳細資訊，請參閱其他資訊一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- AWS Command Line Interface (AWS CLI) (AWS CLI) 2.x 版，已在 macOS、Linux 或視窗上安裝和設定
- Python 版本 3.7
- pip3 — Python 原始程式碼是以 .zip 檔案形式提供，以便部署在 Lambda 函數中。如果您想要在本機使用程式碼或自訂程式碼，請依照下列步驟開發並重新編譯原始程式碼：
  1. 在與 Python 腳本相同的目錄中運行以下命令來生成`requirements.txt`文件：`pip3 freeze > requirements.txt`
  2. 安裝依賴關係：`pip3 install -r requirements.txt`

限制

- 此代碼以 Python 運行，目前不支持其他編程語言。
- 範例應用程式不包含 AWS 跨區域或災難復原 (DR) 支援。
- 此模式僅用於演示目的。它不打算在生產環境中使用。

架構

下圖說明此模式的高階架構。該架構包括以下內容：

- AWS Lambda 可為內容建立索引和查詢
- Amazon OpenSearch 服務執行搜索
- Amazon API Gateway 提供與用戶的 API 交互
- 用於存放原始 (非索引) 資料的 Amazon S3
- Amazon CloudWatch 監控日誌
- AWS Identity and Access Management (IAM) 可建立租用戶角色和政策

自動化和規模

為了簡單起見，該模式使用 AWS CLI 佈建基礎設施和部署範例程式碼。您可以建立 AWS CloudFormation 範本或 AWS Cloud Development Kit (AWS CDK) 指令碼來自動化模式。

工具

AWS 服務

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種統一的工具，可在命令列殼層中使用命令來管理 AWS 服務和資源。
- AWS Lambda — AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon API Gateway — Amazon API Gateway 是一種 AWS 服務，用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種物件儲存服務，可讓您隨時從 Web 上的任何位置存放和擷取任意數量的資訊。
- Amazon OpenSearch 服務 — Amazon 服 OpenSearch 務是一項全受管服務，可讓您輕鬆地以符合成本效益的方式大規模部署、保護和執行 Elasticsearch。

Code

附件提供此病毒碼的範例檔案。其中包含：

- `index_lambda_package.zip`— Lambda 函數，用於使用集區模型在 Amazon OpenSearch 服務中索引資料。
- `search_lambda_package.zip`-Lambda 函數用於搜索 Amazon OpenSearch 服務中的數據。
- `Tenant-1-data`— 承租人 1 的原始 (非索引) 資料範例。
- `Tenant-2-data`— 承租人 2 的原始 (非索引) 資料範例。

重要事項：此模式中的故事包括針對 Unix、Linux 和 macOS 進行格式化的 CLI 命令範例。用於 Windows 時，請以插入號 (^) 取代每一行結尾處的 Unix 接續字元斜線 (\)。

史诗

建立和設定 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	在您的 AWS 區域中建立 S3 儲存貯體。此儲存貯體會保留範例應用程式的非索引租用戶資料。請確定 S3 儲存貯體的名稱是全域唯一的，因為該命名	雲端架構師、雲端管理員

任務	描述	所需技能
	空間是由所有 AWS 帳戶共用的。 若要建立 S3 儲存貯體，您可以使用 AWS CLI 建立儲存貯體命令，如下所示： `aws s3api create-bucket \ --bucket tenantrawdata \ --region <your-AWS-Region>` 其中`tenantrawdata` 是 S3 儲存貯體名稱。(您可以使用任何遵循值區命名準則的唯一名稱。)	

建立和設定彈性搜尋叢集

任務	描述	所需技能
創建一個 Amazon OpenSearch 服務域。	執行 AWS CLI create-elasticsearch-domain命令以建立 Amazon OpenSearch 服務網域： `aws es create-elasticsearch-domain \ --domain-name vpc-cli-example \ --elasticsearch-version 7.10 \ --elasticsearch-cluster-config InstanceType=t3.medium.elas`	雲端架構師、雲端管理員

任務	描述	所需技能
	ticsearch,InstanceCount=1 \ --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=10 \ --domain-endpoint-options "{\"EnforceHTTPS\": true}" \ --encryption-at-rest-options "{\"Enabled\": true}" \ --node-to-node-encryption-options "{\"Enabled\": true}" \ --advanced-security-options "{\"Enabled\": true, \"InternalUserDatabaseEnabled\": true, \ \"MasterUserOptions\": {\"MasterUserName\": \"KibanaUser\", \ \"MasterUserPassword\": \"NewKibanaPassword@123\"}}" \ --vpc-options "{\"SubnetIds\": [\"<subnet-id>\"], \"SecurityGroupIds\": [\"<sg-id>\"]}" \ --access-policies "{\"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \ \"Principal\": {\"AWS\": \"*\" }, \"Action\":\"es:*\", \	

任務	描述	所需技能
	<pre>\"Resource\": \"arn:aws:es:region:account-id:domain\/vpc-cli-example\/*\" }] }"</pre> 執行個體計數設定為 1，因為網域是用於測試目的。您必須使用`advanced-security-options` 參數來啟用精細的存取控制，因為在建立網域之後就無法變更詳細資料。 此命令會建立主要使用者名稱 (`KibanaUser`) 和密碼，以便您用來登入 Kibana 主控台。 由於網域是虛擬私有雲 (VPC) 的一部分，因此您必須確定可以透過指定要使用的存取原則來存取 Elasticsearch 執行個體。 如需詳細資訊，請參閱 AWS 文件中的使用 VPC 啟動 Amazon OpenSearch 服務網域。	

任務	描述	所需技能
設定防禦主機。	將亞馬遜彈性運算雲端 (Amazon EC2) Windows 執行個體設定為防禦主機，以存取 Kibana 主控台。彈性搜尋安全群組必須允許來自 Amazon EC2 安全群組的流量。如需指示，請參閱使用防禦伺服器控制 EC2 執行個體的網路存取部落格文章。 當防禦主機已設定完成，且您擁有與執行個體相關聯的安全群組時，請使用 AWS CLI authorize-security-group-ingress命令向 Elasticsearch 安全群組新增許可，以允許來自 Amazon EC2 (堡壘主機) 安全群組的連接埠 443。 `aws ec2 authorize-security-group-ingress \ --group-id <SecurityGroupIdfElasticSearch> \ --protocol tcp \ --port 443 \ --source-group <SecurityGroupIdfBashionHostEC2>`	雲端架構師、雲端管理員

建立和設定 Lambda 索引函數

任務	描述	所需技能
建立 Lambda 執行角色。	執行 AWS CLI 建立角色命令以授與 Lambda 索引函數存取 AWS 服務和資源： `aws iam create-role \ --role-name index-lambda-role \ --assume-role-policy-document file://lambda_assume_role.json` 其中`lambda_assume_role.json` 是目前資料夾中的 JSON 文件，可授與 Lambda 函數的`AssumeRole` 權限，如下所示： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`	雲端架構師、雲端管理員

任務	描述	所需技能
將受管政策附加至 Lambda 角色。	執行 AWS CLI attach-role-policy命令，將受管政策附加到上一步中建立的角色。這兩個原則會授予角色權限來建立 elastic network interface，以及將記錄寫入記 CloudWatch 錄檔。 `aws iam attach-role-policy \` `  --role-name index-lambda-role \` `  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` `aws iam attach-role-policy \` `  --role-name index-lambda-role \` `  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole`	雲端架構師、雲端管理員

任務	描述	所需技能
建立政策以授與 Lambda 索引函數讀取 S3 物件的權限。	執行 AWS CLI 建立政策命令，以授與 Lambda 索引函數讀取 S3 儲存貯體中物件的`s3:GetObject` 權限： `aws iam create-policy \ --policy-name s3-permission-policy \ --policy-document file://s3-policy.json` 該檔案`s3-policy.json` 是目前資料夾中的 JSON 文件，可授與允許讀取 S3 物件的`s3:GetObject` 權限。如果您在建立 S3 儲存貯體時使用了不同的名稱，請在下列 `Resource` 區段中提供正確的儲存貯體名稱： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::tenantrawdata/*" } ] }`	雲端架構師、雲端管理員

任務	描述	所需技能
將 Amazon S3 許可政策附加到 Lambda 執行角色。	執行 AWS CLI attach-role-policy命令，將您在上一步中建立的 Amazon S3 許可政策附加到 Lambda 執行角色： `aws iam attach-role-policy \` `  --role-name index-lambda-role \` `  --policy-arn <PolicyARN>` 其中`PolicyARN` 是 Amazon S3 許可政策的亞馬遜資源名稱（ARN）。您可以從上一個命令的輸出中獲取此值。	雲端架構師、雲端管理員

任務	描述	所需技能
建立 Lambda 索引函數。	執行 AWS CLI 建立函數命令以建立 Lambda 索引函數，該函數將存取 Amazon 服務：OpenSearch `aws lambda create-function \` `  --function-name index-lambda-function \` `  --zip-file fileb://index_lambda_package.zip \` `  --handler lambda_index.lambda_handler \` `  --runtime python3.7 \` `  --role "arn:aws:iam::account-id:role/index-lambda-role" \` `  --timeout 30 \` `  --vpc-config "{\"SubnetIds\": [\"<subnet-id1\>", \"<subnet-id2>\"], \` `  \"SecurityGroupIds\": [\"<sg-1>\"]}"`	雲端架構師、雲端管理員

任務	描述	所需技能
允許 Amazon S3 調用 Lambda 索引函數。	執行 AWS CLI 新增權限命令，授與 Amazon S3 呼叫 Lambda 索引函數的權限： `aws lambda add-permission \` `  --function-name index-lambda-function \` `  --statement-id s3-permissions \` `  --action lambda:InvokeFunction \` `  --principal s3.amazonaws.com \` `  --source-arn "arn:aws:s3:::tentrawdata" \` `  --source-account "<account-id>"`	雲端架構師、雲端管理員

任務	描述	所需技能
為 Amazon S3 事件新增 Lambda 觸發器。	執行 AWS CLI put-bucket-notification-configuration命令，在偵測到 Amazon S3 `ObjectCreated` 事件時，將通知傳送至 Lambda 索引函數。每當物件上傳到 S3 儲存貯體時，索引函數就會執行。 `aws s3api put-bucket-notification-configuration \ --bucket tenantrawdata \ --notification-configuration file://s3-trigger.json` 該檔案`s3-trigger.json` 是目前資料夾中的 JSON 文件，可在 Amazon S3 `ObjectCreated` 事件發生時將資源政策新增至 Lambda 函數。	雲端架構師、雲端管理員

建立和設定 Lambda 搜尋功能

任務	描述	所需技能
建立 Lambda 執行角色。	執行 AWS CLI 建立角色命令，以授與 Lambda 搜尋函數存取 AWS 服務和資源的權限： `aws iam create-role \ --role-name search-lambda-role \`	雲端架構師、雲端管理員

任務	描述	所需技能
	--assume-role-policy-document file://lambda_assume_role.json 其中lambda_assume_role.json 是目前資料夾中的 JSON 文件，可授與 Lambda 函數的AssumeRole 權限，如下所示： { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" }] }	

任務	描述	所需技能
將受管政策附加至 Lambda 角色。	執行 AWS CLI attach-role-policy命令，將受管政策附加到上一步中建立的角色。這兩個原則會授予角色權限來建立 elastic network interface，以及將記錄寫入記 CloudWatch 錄檔。 `aws iam attach-role-policy \` `  --role-name search-lambda-role \` `  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` `aws iam attach-role-policy \` `  --role-name search-lambda-role \` `  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole`	雲端架構師、雲端管理員

任務	描述	所需技能
建立搜 Lambda 函數。	執行 AWS CLI 建立函數命令以建立 Lambda 搜尋函數，該函數將存取 Amazon 服務：OpenSearch `aws lambda create-function \ --function-name search-lambda-function \ --zip-file fileb://search_lambda_package.zip \ --handler lambda_search.lambda_handler \ --runtime python3.7 \ --role "arn:aws:iam::account-id:role/search-lambda-role" \ --timeout 30 \ --vpc-config "{\"SubnetIds\": [\"<subnet-id1\>", \"<subnet-id2>\"], \ \"SecurityGroupIds\": [\"<sg-1>\"]}"`	雲端架構師、雲端管理員

建立和設定承租人角色

任務	描述	所需技能
建立租用戶 IAM 角色。	執行 AWS CLI 建立角色命令以建立兩個用於測試搜尋功能的租用戶角色： `aws iam create-role \`	雲端架構師、雲端管理員

任務	描述	所需技能
	--role-name Tenant-1-role \ --assume-role-policy-document file://assume-role-policy.json aws iam create-role \ --role-name Tenant-2-role \ --assume-role-policy-document file://assume-role-policy.json 該檔案assume-role-policy.json 是目前資料夾中的 JSON 文件，可授與 Lambda 執行角色的AssumeRole 權限： { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "<Lambda execution role for index function>", "AWS": "<Lambda execution role for search function>" }, "Action": "sts:AssumeRole" }]	

任務	描述	所需技能
	} ```	

任務	描述	所需技能
建立租用戶 IAM 政策。	執行 AWS CLI 建立政策命令以建立可授予 Elasticsearch 操作存取權的租用戶政策： `aws iam create-policy \ --policy-name tenant-policy \ --policy-document file://policy.json` 該文件`policy.json` 是當前文件夾中的 JSON 文檔，它授予對彈性搜索的權限： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpDelete", "es:ESHttpGet", "es:ESHttpHead", "es:ESHttpPost", "es:ESHttpPut", "es:ESHttpPatch" ], "Resource": [`	雲端架構師、雲端管理員

任務	描述	所需技能
	<pre>"<ARN of Elasticsearch domain created earlier>"] }] }</pre>	
將租用戶 IAM 政策附加到租用戶角色。	執行 AWS CLI attach-role-policy命令，將租用戶 IAM 政策附加到您在先前步驟中建立的兩個租用戶角色： <pre>aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-1-role aws iam attach-role-policy \ --policy-arn arn:aws:iam::account-id:policy/tenant-policy \ --role-name Tenant-2-role</pre> 原則 ARN 來自上一個步驟的輸出。	雲端架構師、雲端管理員

任務	描述	所需技能
建立 IAM 政策以授予 Lambda 許可以擔任角色。	執行 AWS CLI 建立政策命令，為 Lambda 建立政策以承擔租用戶角色： `aws iam create-policy \ --policy-name assume-tenant-role-policy \ --policy-document file://lambda_policy.json` 該文件`lambda_policy.json` 是當前文件夾中的 JSON 文檔，可授予以下權限`AssumeRole` ： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "<ARN of tenant role created earlier>" } ] }` 對於`Resource`，您可以使用萬用字元來避免為每個承租人建立新原則。	雲端架構師、雲端管理員

任務	描述	所需技能
建立 IAM 政策以授予 Lambda 索引角色存取 Amazon S3 的權限。	執行 AWS CLI 建立政策命令，授與 Lambda 索引角色存取 S3 儲存貯體中物件的權限： `aws iam create-policy \ --policy-name s3-permission-policy \ --policy-document file://s3_lambda_policy.json` 該文件`s3_lambda_policy.json` 是當前文件夾中的以下 JSON 策略文檔： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::tenantrawdata/*" } ] }`	雲端架構師、雲端管理員

任務	描述	所需技能
將原則附加至 Lambda 執行角色。	執行 AWS CLI attach-role-policy命令，將上一步中建立的政策附加到您先前建立的 Lambda 索引和搜尋執行角色： `aws iam attach-role-policy \` `  --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \` `  --role-name index-lambda-role` `aws iam attach-role-policy \` `  --policy-arn arn:aws:iam::account-id:policy/assume-tenant-role-policy \` `  --role-name search-lambda-role` `aws iam attach-role-policy \` `  --policy-arn arn:aws:iam::account-id:policy/s3-permission-policy \` `  --role-name index-lambda-role` 原則 ARN 來自上一個步驟的輸出。	雲端架構師、雲端管理員

建立和設定搜尋 API

任務	描述	所需技能
在 API Gateway 中建立 REST API。	執行 CLI create-rest-api命令以建立其他 API 資源： `aws apigateway create-rest-api \` `  --name Test-Api \` `  --endpoint-configuration "{ \"types\": [\"REGIONAL\"] }"` 對於端點組態類型，您可以指定EDGE而不是使REGIONAL用節點位置，而不是使用特定 AWS 區域。 請注意來自命令輸出的id欄位值。這是您將在後續命令中使用的 API ID。	雲端架構師、雲端管理員
建立搜尋 API 的資源。	搜尋 API 資源會以資源名稱啟動 Lambda 搜尋功能search。您不必為 Lambda 索引函數建立 API，因為它會在物件上傳至 S3 儲存貯體時自動執行。） 1. 執行 AWS CLI 取得資源命令以取得根路徑的父 ID： `aws apigateway get-resources \` `  --rest-api-id <API-ID>`	雲端架構師、雲端管理員

任務	描述	所需技能
	請注意 ID 欄位的值。您將在下一個命令中使用此父 ID。 `{ "items": [ { "id": "zpsri964ck", "path": "/" } ] }` 2. 執行 AWS CLI 建立資源命令，為搜尋 API 建立資源。對於`parent-id` ，指定上一個指令的 ID。 `aws apigateway create-resource \ --rest-api-id <API-ID> \ --parent-id <Parent-ID> \ --path-part search`	

任務	描述	所需技能
建立搜尋 API 的 GET 方法。	執行 AWS CLI 放置方法命令以建立搜尋 API 的GET 方法： `aws apigateway put-method \` `  --rest-api-id <API-ID> \` `  --resource-id <ID from the previous command output> \` `  --http-method GET \` `  --authorization-type "NONE" \` `  --no-api-key-required` 對於`resource-id` ，指定`create-resource` 命令輸出中的 ID。	雲端架構師、雲端管理員

任務	描述	所需技能
建立搜尋 API 的方法回應。	執行 AWS CLI put-method-response命令以新增搜尋 API 的方法回應： `aws apigateway put-method-response \ --rest-api-id <API-ID> \ --resource-id <ID from the create-resource command output> \ --http-method GET \ --status-code 200 \ --response-models "{\"application/json\": \"Empty\"}"` 對於 `resource-id` ，指定先前`create-resource` 命令輸出的 ID。	雲端架構師、雲端管理員

任務	描述	所需技能
為搜尋 API 設定代理 Lambda 整合。	執行 AWS CLI 命令放置整合命令，以設定與 Lambda 搜尋功能的整合： `aws apigateway put-integration \` `  --rest-api-id <API-ID> \` `  --resource-id  <ID from the create-resource command output> \` `  --http-method GET \` `  --type AWS_PROXY \` `  --integration-http-method GET \` `  --uri arn:aws:apigateway:region:lambda:path/2015-03-31/functions/arn:aws:lambda:<region>:<account-id>:function:<function-name>/invocations` 對於`resource-id` ，指定先前指`create-resource` 令中的 ID。	雲端架構師、雲端管理員

任務	描述	所需技能
授予 API Gateway 權限以呼叫 Lambda 搜尋功能。	執行 AWS CLI 新增權限命令，授與 API Gateway 使用搜尋功能的權限： `aws lambda add-permission \ --function-name <function-name> \ --statement-id apigateway-get \ --action lambda:InvokeFunction \ --principal apigateway.amazonaws.com \ --source-arn "arn:aws:execute-api:<region>:<account-id>:api-id/*/GET/search"` 如果您使用不同的 API 資源名稱而不是更改`source-arn` 路徑`search`。	雲端架構師、雲端管理員
部署搜尋 API。	執行 AWS CLI 建立部署命令以建立名為的階段資源：`dev` `aws apigateway create-deployment \ --rest-api-id <API-ID> \ --stage-name dev` 如果您更新 API，則可以使用相同的 CLI 命令將其重新部署到相同的階段。	雲端架構師、雲端管理員

建立和設定 Kibana 角色

任務	描述	所需技能
登入木花主控台。	1. 在 Amazon OpenSearch 服務主控台上的網域儀表板上找到 Kibana 的連結。網址的格式如下：`<domain-endpoint>/_plugin/kibana/` 。 2. 使用您在第一個史詩中配置的堡壘主機來訪問 Kibana 控制台。 3. 在建立 Amazon OpenSearch 服務網域時，使用先前步驟中的主要使用者名稱和密碼登入 Kibana 主控台。 4. 當系統提示您選取承租人時，請選擇 [私人]。	雲端架構師、雲端管理員
建立和設定 Kibana 角色。	若要提供資料隔離，並確保一個承租人無法擷取另一個承租人的資料，您需要使用文件安全性，這可讓租用戶只存取包含其租用戶識別碼的文件。 1. 在 Kibana 主控台的功能窗格中，選擇 [安全性]、[角色]。 2. 建立新承租人角色。 3. 將叢集許可設定為`indices_all` ，以便在 Amazon OpenSearch 服務索引上提供建立、讀取、更新和刪除 (CRUD) 權限。	雲端架構師、雲端管理員

任務	描述	所需技能
	4. 限制索引的tenant-data 索引權限。(索引名稱應與 Lambda 搜尋和索引函數中的名稱相符。) 5. 將索引權限設定為indices_all ，可讓使用者執行所有索引相關作業。您可以根據您的需求限制操作以獲得更細微的存取。) 6. 針對文件層級安全性，請使用下列原則依租用戶識別碼篩選文件，以便為共用索引中的租用戶提供資料隔離： `{ "bool": { "must": { "match": { "TenantId": "Tenant-1" } } } }` 索引名稱、屬性和值區分大小寫。	

任務	描述	所需技能
將使用者對應至角色。	1. 選擇角色的 [對應的使用者] 索引標籤，然後選擇 [對應使用者]。 2. 在後端角色區段中，指定您先前建立的 IAM 租用戶角色的 ARN，然後選擇 [對應]。這會將 IAM 租用戶角色對應至 Kibana 角色，以便承租人特定搜尋只會傳回該租用戶的資料。例如，如果承租人 1 的 IAM 角色名稱是*Tenant-1-Role* ，請在承租人 1 Kibana 角色的後端角色方塊中為 *Tenant-1-Role* (從建立和設定租用戶角色史詩) 指定 ARN。 3. 對承租人 2 重複步驟 1 和 2。 我們建議您在承租人上線時自動建立承租人和 Kibana 角色。	雲端架構師、雲端管理員

任務	描述	所需技能
建立承租人資料索引。	在功能窗格的 [管理] 下，選擇 [開發工具]，然後執行下列命令。此指令會建立tenant-data 索引以定義TenantId性質的對映。 `PUT /tenant-data { "mappings": { "properties": { "TenantId": { "type": "keyword"} } } }`	雲端架構師、雲端管理員

為 Amazon S3 和 AWS STS 建立 VPC 端點

任務	描述	所需技能
為 Amazon S3 建立 VPC 端點。	執行 AWS CLI create-vpc-endpoint命令以建立適用於 Amazon S3 的 VPC 端點。端點可讓 VPC 中的 Lambda 索引函數存取 Amazon S3 服務。 `aws ec2 create-vpc-endpoint \ --vpc-id <VPC-ID> \ --service-name com.amazonaws.us-east-1.s3 \ --route-table-ids <route-table-ID>`	雲端架構師、雲端管理員

任務	描述	所需技能
	對於vpc-id，請指定您要用於 Lambda 索引函數的 VPC。對於service-name ，請使用適用於 Amazon S3 端點的正確 URL。對於route-table-ids ，指定與 VPC 端點關聯的路由表。	

任務	描述	所需技能
為 AWS STS 建立 VPC 私人雲端端點。	執行 AWS CLI create-vpc-endpoint命令，為 AWS 安全性權杖服務 (AWS STS) 建立 VPC 端點。端點可讓 VPC 中的 Lambda 索引和搜尋函數存取 AWS STS 服務。這些函數在擔任 IAM 角色時會使用 AWS STS。 `aws ec2 create-vpc-endpoint \ --vpc-id <VPC-ID> \ --vpc-endpoint-type Interface \ --service-name com.amazonaws.us-east-1.sts \ --subnet-id <subnet-ID> \ --security-group-id <security-group-ID>` 對於`vpc-id`，請指定您用於 Lambda 索引和搜尋函數的 VPC。對於`subnet-id` ，請提供應在其中建立此端點的子網路。對於`security-group-id` ，指定要與此端點相關聯的安全群組。(它可能與 Lambda 使用的安全性群組相同。)	雲端架構師、雲端管理員

測試多租戶和資料隔離

任務	描述	所需技能
更新 Python 文件的索引和搜索功能。	1. 在`index_lambda_package.zip` 檔案中，編輯 `lamba_index.py` 檔案以更新 AWS 帳戶 ID、AWS 區域和彈性搜尋端點資訊。 2. 在`search_lambda_package.zip` 檔案中，編輯`lambda_search.py` 檔案以更新 AWS 帳戶 ID、AWS 區域和彈性搜尋端點資訊。 您可以從 Amazon OpenSearch 服務主控台的概觀索引標籤取得彈性搜尋端點。它具有格式`<AWS-Region>.es.amazonaws.com` 。	雲架構師、應用程式開發
更新程 Lambda 碼。	使用 AWS CLI update-function-code命令，以您對 Python 檔案所做的變更來更新 Lambda 程式碼： `aws lambda update-function-code \` `  --function-name index-lambda-function \` `  --zip-file fileb://index_lambda_package.zip`	雲架構師、應用程式開發

任務	描述	所需技能
	`aws lambda update-function-code \ --function-name search-lambda-function \ --zip-file fileb://search_lambda_package.zip`	
將原始資料上傳到 S3 儲存貯體。	使用 AWS CLI cp 命令將承租人 1 和承租人 2 物件的資料上傳到儲存`tenantrawdata` 貯體 (指定為此目的建立的 S3 儲存貯體的名稱)： `aws s3 cp tenant-1-data s3://tenantrawdata` `aws s3 cp tenant-2-data s3://tenantrawdata` S3 儲存貯體設定為在每次上傳資料時執行 Lambda 索引函數，以便在 Elasticsearch 中為文件建立索引。	雲端架構師、雲端管理員
從 Kibana 控制台搜索數據。	在 Kibana 主控台上，執行下列查詢： `GET tenant-data/_search` 此查詢會顯示所有在 Elasticsearch 中編製索引的文件。在這種情況下，您應該看到租戶 1 和承租人 2 的兩個單獨的文件。	雲端架構師、雲端管理員

任務	描述	所需技能
從 API Gateway 測試搜尋 API。	1. 在 API Gateway 主控台中，開啟搜尋 API，選擇搜尋資源內的GET方法，然後選擇 [測試]。 2. 在測試視窗中，為租用戶識別碼提供下列查詢字串 (區分大小寫)，然後選擇 [測試]。 `TenantId=Tenant-1` Lambda 函數會傳送查詢至 Amazon OpenSearch 服務，根據文件層級安全性篩選租用戶文件。該方法返回屬於承租人 1 的文檔。 3. 將查詢字串變更為： `TenantId=Tenant-2` 此查詢會傳回屬於承租人 2 的文件。 如需螢幕插圖，請參閱其他資訊一節。	雲架構師、應用程式開發

相關資源

- 適用於 Python 的 AWS SDK (Boto3)
- AWS Lambda 文件
- Amazon API Gateway 文件
- Amazon S3 文件

- Amazon OpenSearch 服務文檔
  - Amazon OpenSearch 服務中的精細訪問控制
  - 使用 Amazon OpenSearch 服務創建搜索應用程序
  - 在 VPC 中啟動您的 Amazon OpenSearch 服務域

其他資訊

資料分割模型

多租戶系統中使用三種常見的資料分割模型：筒倉、集區和混合式。您選擇的模型取決於環境的合規性、嘈雜鄰點、作業和隔離需求。

筒倉模型

在筒倉模型中，每個租戶的數據存儲在不包含租戶數據混合的不同存儲區域中。您可以使用兩種方法透過 Amazon Ser OpenSearch vice 實作筒倉模型：每個租用戶的網域和每個租用戶的索引。

- 每個租用戶的網域 — 您可以為每個租用戶使用個別的 Amazon OpenSearch 服務網域 (與 Elasticsearch 叢集同義)。將每個承租人放置在其自己的網域中，可提供與在獨立建構中擁有資料相關的所有好處。但是，這種方法引入了管理和敏捷性挑戰。其分佈式特性使得匯總和評估租戶的運營健康狀況和活動變得更加困難。這是一個昂貴的選項，要求每個 Amazon Ser OpenSearch vice 網域至少要有三個主節點和兩個資料節點來處理生產工作負載。

- 每個租用戶索引 — 您可以將租用戶資料放置在 Amazon OpenSearch 服務叢集中的個別索引中。透過這種方法，您可以在建立索引並命名索引時使用承租人識別碼，方法是將承租人識別碼預先擱置至索引名稱。每個租用戶索引方法可協助您達成孤島目標，而無需為每個租用戶引入完全獨立的叢集。但是，如果索引數量增加，您可能會遇到內存壓力，因為這種方法需要更多的碎片，並且主節點必須處理更多的配置和重新平衡。

隔離筒倉模型 — 在筒倉模型中，您可以使用 IAM 政策隔離保存每個租用戶資料的網域或索引。這些原則可防止一個承租人存取另一個承租人的資料。若要實作筒倉隔離模型，您可以建立以資源為基礎的政策，以控制對租用戶資源的存取。這通常是網域存取原則，用來指定主體可以對網域的子資源執行哪些動作，包括 Elasticsearch 索引和 API。使用 IAM 身分型政策，您可以針對 Amazon 服務中的網域、索

引或 API 指定允許或拒絕的動作。 OpenSearch IAM 政策的Action 元素描述政策允許或拒絕的特定動作，並指定受影響的帳戶、使用者或角色。Principal

下列範例原則僅授與 Tenant 1 對網域上子資源的完整存取權 (如指定es:*)。tenant-1Resource 元素/*中的尾端表示此原則適用於網域的子資源，而非網域本身。此原則生效時，不允許租用戶在現有網域上建立新網域或修改設定。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::aws-account-id:user/Tenant-1"
         },
         "Action": "es:*",
         "Resource": "arn:aws:es:Region:account-id:domain/tenant-1/*"
      }
   ]
}
```

若要實作每個索引筒倉模型的承租人，您需要修改此範例原則，藉由指定索引名稱，進一步將 Tenant 1 限制為指定的一或多個索引。下列範例原則會將承租人 1 限制為索引。tenant-index-1

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Tenant-1"
         },
         "Action": "es:*",
         "Resource": "arn:aws:es:Region:account-id:domain/test-domain/tenant-index-1/*"
      }
   ]
}
```

泳池模型

在集區模型中，所有承租人資料都儲存在相同網域內的索引中。承租人識別碼包含在資料 (文件) 中並做為磁碟分割索引鍵使用，因此您可以判斷哪些資料屬於哪個承租人。這種模式減少了管理開銷。操作

和管理集區索引比管理多個索引更容易，更有效率。但是，由於租用戶資料混合在相同的索引中，因此您會失去筒倉模型提供的自然租用戶隔離。此方法也可能會因為鄰近雜訊效應而降低效能。

集區模型中的租用戶隔離 — 一般而言，在集區模型中實作租用戶隔離具有挑戰性。與筒倉模型搭配使用的 IAM 機制不允許您根據文件中儲存的租用戶 ID 來描述隔離。

另一種方法是使用 Elasticsearch 開放發行版提供的精細訪問控制（FGAC）支持。FGAC 可讓您在索引、文件或欄位層級控制權限。FGAC 會針對每個要求評估使用者認證，並驗證使用者或拒絕存取。如果 FGAC 驗證使用者，它會擷取對應至該使用者的所有角色，並使用完整的權限集來決定如何處理要求。

若要在集區模型中達到所需的隔離，您可以使用文件層級安全性，這可讓您將角色限制為索引中的文件子集。下列範例角色會將查詢限制為承租人 1。透過將此角色套用至承租人 1，您可以達到必要的隔離。

```
{
   "bool": {
     "must": {
       "match": {
         "tenantId": "Tenant-1"
       }
     }
   }
 }
```

混合模型

混合模型在相同環境中使用筒倉和集區模型的組合，為每個租用戶層 (例如免費、標準和進階層) 提供獨特的體驗。每個層都遵循集區模型中使用的相同安全性設定檔。

混合模型中的租用戶隔離 — 在混合模型中，您遵循與集區模型中相同的安全性設定檔，在此模型中使用文件層級的 FGAC 安全性模型會提供租用戶隔離。雖然此策略簡化了叢集管理並提供敏捷性，但是它使架構的其他層面變得複雜。例如，您的程式碼需要額外的複雜性，才能判斷與每個租用戶相關聯的模型。您還必須確保單租用戶查詢不會飽和整個網域，並降低其他租用戶的體驗。

在 API Gateway 中進行測試

租戶 1 查詢的測試窗口

承租人 2 查詢的測試視窗

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS CDK 來部署多堆疊應用程式 TypeScript

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）創建

環境：生產	技術：現代化; 遷移; DevOps	工作負載：所有其他工作
AWS 服務：Amazon API Gateway；AWS Lambda；Amazon Kinesis		

Summary

此模 step-by-step 式為使用 AWS Cloud Development Kit (AWS CDK) 搭 TypeScript配使用的 Amazon Web Services (AWS) 上的應用程式部署提供了一種方法。例如，該病毒碼會部署無伺服器即時分析應用程式。

此模式會建置和部署巢狀堆疊應用程式。父 AWS CloudFormation 堆疊會呼叫子系或巢狀堆疊。 每個子堆疊都會建置和部署堆 CloudFormation 疊中定義的 AWS 資源。AWS CDK 工具組是命令列界面 (CLI) 命令`cdk`，是 CloudFormation 堆疊的主要介面。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 現有的虛擬私有雲 (VPC) 和子網路
- 安裝和設定 AWS CDK 工具組
- 具有管理員權限和一組存取金鑰的使用者。
- Node.js
- AWS 命令列界面 (AWS CLI)

限制

- 由於 AWS CDK 使用 AWS CloudFormation，因此 AWS CDK 應用程式會受到 CloudFormation 服務配額限制。如需詳細資訊，請參閱 AWS CloudFormation 配額。

產品版本

此病毒碼已使用下列工具和版本建立並測試。

- AWS CDK 工具包 1.83.0
- Node.js 14.13.0
- 故宮

該模式應該適用於任何版本的 AWS CDK 或 npm。請注意，Node.js 版本 13.0.0 到 13.6.0 與 AWS CDK 不相容。

架構

目標技術堆疊

- AWS Amplify 控制台
- Amazon API Gateway
- AWS CDK
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon 數據 Firehose
- Amazon Kinesis Data Streams
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)

目標架構

下圖顯示使用 AWS CDK 搭配使用的多堆疊應用程式部署。 TypeScript

下圖顯示範例無伺服器即時應用程式的架構。

工具

工具

- AWS Amplify 主控台是 AWS 中全堆疊 Web 和行動應用程式部署的控制中心。Amplify 控制台託管提供基於 git 的工作流程，用於託管具有持續部署的全棧無服務器 Web 應用程序。Admin UI 是一種視覺化界面，供前端 Web 和行動開發人員在 AWS 主控台外部建立和管理應用程式後端。
- Amazon API Gateway 是一種 AWS 服務，用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CDK 工具組是命令列雲端開發套件，可協助您與 AWS CDK 應用程式互動。cdkCLI 命令是與 AWS CDK 應用程式互動的主要工具。它會執行您的應用程式、詢問您定義的應用程式模型，以及產生和部署 AWS CDK 產生的 AWS CloudFormation 範本。
- Amazon CloudFront 是一種網絡服務，可以加快靜態和動態網絡內容的分佈速度，例如 .html，.css，.js 和圖像文件。 CloudFront 透過稱為節點位置的全球資料中心網路傳遞您的內容，以降低延遲並提升效能。
- Amazon Cognito 為您的 Web 和行動應用程式提供身分驗證、授權和使用者管理。您的使用者可以直接登入，也可以透過第三方登入。
- Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- Amazon 資料 Firehose 是一項全受管服務，可將即時串流資料交付到 Amazon S3、Amazon Redshift、亞馬遜服務、Splunk 等目的地，以及受支援的第三方 OpenSearch 服務供應商擁有的任何自訂 HTTP 端點或 HTTP 端點。
- Amazon Kinesis Data Streams 是一種可即時收集和處理大型資料串流記錄的服務。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

此模式的代碼已附加。

史诗

安裝 AWS CDK 工具組

任務	描述	所需技能
安裝 AWS CDK 工具組。	若要在全域安裝 AWS CDK 工具組，請執行下列命令。 `npm install -g aws-cdk`	DevOps
驗證版本。	若要驗證 AWS CDK 工具組版本，請執行下列命令。 `cdk --version`	DevOps

設定 AWS 登入資料

任務	描述	所需技能
設定認證。	若要設定認證，請執行命`aws configure` 令並依照提示執行。 `$aws configure` `AWS Access Key ID` `[None]:` `AWS Secret Access Key` `[None]: your_secr` `et_access_key` `Default region name` `[None]:` `Default output format` `[None]:`	DevOps

下載專案代碼

任務	描述	所需技能
下載附加的專案代碼。	如需有關目錄和檔案結構的詳細資訊，請參閱其他資訊一節。	DevOps

啟動 AWS CDK 環境

任務	描述	所需技能
引導環境。	若要將 AWS CloudFormation 範本部署到您要使用的帳戶和 AWS 區域，請執行下列命令。 `cdk bootstrap <account>/<Region>` 如需詳細資訊，請參閱 AWS 文件。	DevOps

建置和部署專案

任務	描述	所需技能
建置專案。	若要建置專案程式碼，請執行`npm run build`命令。	DevOps
部署專案。	若要部署專案程式碼，請執行`cdk deploy`命令。	

驗證輸出

任務	描述	所需技能
確認堆疊建立。	在 AWS 管理主控台上，選擇 CloudFormation。在專案的堆疊中，確認已建立父堆疊和兩個子堆疊。	DevOps

測試應用程式。

任務	描述	所需技能
將資料傳送至 Kinesis Data Streams。	將您的 AWS 帳戶設定為使用 Amazon Kinesis Data Streams 產生器 (KDG) 將資料傳送至 Kinesis 資料串流。如需詳細資訊，請參閱 Amazon Kinesis 資料產生器。	DevOps
創建一個 Amazon Cognito 戶。	若要建立 Amazon Cognito 使用者，請從「Kinesis 資料產生器說明」頁面上的「建立 Amazon Cognito 使用者」區段下載認知設定 CloudFormation 範本。啟動範本，然後輸入您的 Amazon Cognito 使用者名稱和密碼。 「輸出」標籤會列出 Kinesis 資料產生器 URL。	DevOps
登入 Kinesis 資料產生器	若要登入 KDG，請使用您提供的 Amazon Cognito 登入資料和 Kinesis 資料產生器 URL。	DevOps

任務	描述	所需技能
測試應用程式。	在 KDG 的記錄範本的範本 1 中，貼上 [其他資訊] 區段中的測試程式碼，然後選擇 [傳送資料]。	DevOps
測試 API Gateway。	擷取資料之後，請使用擷取資料的GET方法來測試 API Gateway。	DevOps

相關資源

參考

- AWS Cloud Development Kit
- 開啟 AWS CDK GitHub
- 使用巢狀堆疊
- AWS 範例範例-無伺服器即時分析

其他資訊

目錄和檔案詳細資訊

此模式設置了以下三個堆棧。

- `parent-cdk-stack.ts`— 此堆疊充當父系堆疊，並將兩個子應用程式呼叫為巢狀堆疊。
- `real-time-analytics-poc-stack.ts`— 此巢狀堆疊包含基礎結構和應用程式程式碼。
- `real-time-analytics-web-stack.ts`— 此巢狀堆疊僅包含靜態 Web 應用程式程式碼。

重要文件及其功能

- `bin/real-time-analytics-poc.ts`— AWS CDK 應用程式的進入點。它加載下定義的所有堆棧`lib/`。
- `lib/real-time-analytics-poc-stack.ts`— AWS CDK 應用程式堆疊的定義 (`real-time-analytics-poc`)。

- lib/real-time-analytics-web-stack.ts— AWS CDK 應用程式堆疊的定義 (real-time-analytics-web-stack)。
- lib/parent-cdk-stack.ts— AWS CDK 應用程式堆疊的定義 (parent-cdk)。
- package.json— npm 模組清單，其中包括應用程式名稱、版本和相依性。
- package-lock.json— 由 npm 維護。
- cdk.json-用於運行應用程序的工具包。
- tsconfig.json— 該項目的 TypeScript 配置。
- .gitignore— Git 應該從原始檔控制中排除的檔案清單。
- node_modules-由 npm 維護; 包括項目的依賴關係。

父系堆疊中的下列程式碼區段會呼叫子應用程式做為巢狀 AWS CDK 堆疊。

```
import * as cdk from '@aws-cdk/core';
import { Construct, Stack, StackProps } from '@aws-cdk/core';
import { RealTimeAnalyticsPocStack } from './real-time-analytics-poc-stack';
import { RealTimeAnalyticsWebStack } from './real-time-analytics-web-stack';


export class CdkParentStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);


    new RealTimeAnalyticsPocStack(this, 'RealTimeAnalyticsPocStack');
    new RealTimeAnalyticsWebStack(this, 'RealTimeAnalyticsWebStack');
  }
}
```

用於測試的代碼

```
session={{date.now('YYYYMMDD')}}|sequence={{date.now('x')}}|
reception={{date.now('x')}}|instrument={{random.number(9)}}|
l={{random.number(20)}}|price_0={{random.number({"min":10000,
 "max":30000})}}|price_1={{random.number({"min":10000, "max":30000})}}|
price_2={{random.number({"min":10000, "max":30000})}}|
price_3={{random.number({"min":10000, "max":30000})}}|
price_4={{random.number({"min":10000, "max":30000})}}|
price_5={{random.number({"min":10000, "max":30000})}}|
price_6={{random.number({"min":10000, "max":30000})}}|
```

```
price_7={{random.number({"min":10000, "max":30000})}}|
price_8={{random.number({"min":10000, "max":30000})}}|
```

測試 API Gateway

在 API Gateway 主控台上，使用GET方法測試 API Gateway。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS SAM 自動部署巢狀應用程式

由拉胡爾·沙拉德·蓋克瓦德博士（AWS），德米特里古林（AWS），伊什瓦爾喬凱瓦勒（AWS）和塔比沃德（AWS）創建

代碼存儲庫：aws-sam-nested-stack-示例	環境：PoC 或試點	技術：現代化；無伺服器；DevOps
工作負載：所有其他工作	AWS 服務：AWS Serverless Application Repository	

Summary

在 Amazon Web Services (AWS) 上，AWS Serverless Application Model (AWS SAM) 是一種開放原始碼架構，提供簡寫語法來表示函數、API、資料庫和事件來源對應。每個資源只需幾行，您就可以定義所需的應用程式，並使用 YAML 建立模型。在部署期間，SAM 會將 SAM 語法轉換並擴充為 AWS CloudFormation 語法，以便更快速地建置無伺服器應用程式。

AWS SAM 可簡化 AWS 平台上無伺服器應用程式的開發、部署和管理作業。它提供了標準化框架，更快的部署，本地測試功能，資源管理，與開發工具的無縫集成以及支持社區。這些功能使其成為有效率且有效地建置無伺服器應用程式的重要工具。

此模式使用 AWS SAM 範本自動化巢狀應用程式的部署。巢狀應用程式是另一個應用程式中的應用程式。父應用程式會呼叫其子應用程式。這些是無伺服器架構的鬆散耦合元件。

使用巢狀應用程式，您可以重複使用獨立編寫和維護但使用 AWS SAM 和無伺服器應用程式儲存庫組成的服務或元件，快速建立高度複雜的無伺服器架構。巢狀應用程式可協助您建置功能更強大的應用程式、避免重複工作，並確保團隊和組織間的一致性與最佳作法。為了示範巢狀應用程式，該模式會部署 AWS 無伺服器購物車應用程式範例。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 現有的虛擬私有雲 (VPC) 和子網路

- 整合式開發環境，例如 AWS Cloud9 或 Visual Studio 程式碼 (如需詳細資訊，請參閱在 AWS 上建置的工具)
- 使用 pip 安裝輪安裝的 Python 輪庫，如果尚未安裝

限制

- 無伺服器應用程式中可以巢狀化的應用程式數目上限為 200 個。
- 巢狀應用程式的最大參數數目可以有 60 個。

產品版本

- 此解決方案建立在 AWS SAM 命令列界面 (AWS SAM CLI) 1.21.1 版本上，但此架構應該可以與更新的 AWS SAM CLI 版本搭配使用。

架構

目標技術堆疊

- Amazon API Gateway
- AWS SAM
- Amazon Cognito
- Amazon DynamoDB
- AWS Lambda
- Amazon Simple Queue Service (Amazon SQS) 佇列

目標架構

下圖顯示了如何通過調用 API 向購物服務發出用戶請求。使用者的請求 (包括所有必要的資訊) 會傳送至 Amazon API Gateway 和 Amazon Cognito 授權者，該授權者會針對 API 執行身份驗證和授權機制。

在 DynamoDB 中新增、刪除或更新項目時，會在 DynamDynamoDB Streams 上放置一個事件，然後啟動 Lambda 函數。為了避免在同步工作流程中立即刪除舊項目，會將訊息置於 SQS 佇列中，該佇列會啟動 Worker 函數以刪除訊息。

在此解決方案設定中，AWS SAM CLI 可做為 AWS CloudFormation 堆疊的界面。AWS SAM 範本會自動部署巢狀應用程式。父 SAM 範本會呼叫子範本，而父系 CloudFormation 堆疊會部署子堆疊。每個子堆疊都會建置 AWS SAM CloudFormation 範本中定義的 AWS 資源。

1. 建置和部署堆疊。
2. 身份驗證 CloudFormation 堆棧包含 Amazon Cognito。
3. 產品 CloudFormation 堆疊包含 Lambda 函數和 Amazon API Gateway
4. 購物 CloudFormation 堆疊包含一個 Lambda 函數、Amazon API Gateway、SQS 佇列和 Amazon DynamoDB 資料庫。

工具

工具

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon Cognito 為網頁和行動應用程式提供身份驗證、授權和使用者管理功能。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Serverless Application Model (AWS SAM) 是一種開放原始碼架構，可協助您在 AWS 雲端建置無伺服器應用程式。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。

Code

此模式的程式碼可在 GitHub AWS SAM 巢狀堆疊範例儲存庫中取得。

史诗

安装 AWS 三 CLI

任務	描述	所需技能
安裝 AWS 山姆 CLI。	若要安裝 AWS SAM CLI，請參閱 AWS SAM 文件中的指示。	DevOps 工程師
設定 AWS 登入資料。	若要設定 AWS 登入資料，以便 AWS SAM CLI 可以代表您呼叫 AWS 服務，請執行`aws configure` 命令並遵循提示進行操作。 `$aws configure` `AWS Access Key ID` `  [None]: <your_acc` `ess_key_id>` `AWS Secret Access Key` `  [None]: your_secr` `et_access_key` `Default region name` `  [None]:` `Default output format` `  [None]:` 如需有關設定認證的詳細資訊，請參閱驗證和存取認證。	DevOps 工程師

初始化 AWS SAM 專案

任務	描述	所需技能
複製 AWS SAM 程式碼儲存庫。	1. 輸入以下命令，克隆此模式的 aws sam 嵌套堆棧示例存儲庫。	DevOps 工程師

任務	描述	所需技能
	`git clone https://github.com/aws-samples/aws-sam-nested-stack-sample.git` 2. 輸入下列命令，導覽至複製的目錄。 `cd aws-sam-nested-stack-sample`	
部署範本以初始化專案。	若要初始化專案，請執行`SAM init`指令。當系統提示您選擇範本來源時，請選擇`Custom Template Location`。	DevOps 工程師

編譯並建置 SAM 範本程式碼

任務	描述	所需技能
檢閱 AWS SAM 應用程式範本。	檢閱巢狀應用程式的範本。此範例使用下列巢狀應用程式範本： • `auth.yaml` — 此範本設定驗證相關資源，例如 Amazon Cognito 和 AWS Systems Manager Parameter Store。 • `product-mock.yaml` — 此範本會部署與產品相關的資源，例如 Lambda 函數和 Amazon API Gateway。	DevOps 工程師

任務	描述	所需技能
	• `shoppingcart-service.yaml` — 此範本會設定與購物車相關的資源，例如 AWS Identity and Access Management (IAM)、DynamoDB 資料表和 Lambda 函數。	
檢閱父範本。	複查將叫用巢狀應用程式範本的範本。在此範例中，父範本為`template.yml` 。所有單獨的應用程序都嵌套在單個父模板中`template.yml` 。	DevOps 工程師
編譯並建置 AWS SAM 範本程式碼。	使用 AWS SAM CLI 執行下列命令。 `sam build`	DevOps 工程師

部署 AWS SAM 範本

任務	描述	所需技能
部署應用程式。	若要啟動建立巢狀應用程式 CloudFormation 堆疊並在 AWS 環境中部署程式碼的 SAM 範本程式碼，請執行下列命令。 `sam deploy --guided --stack-name shopping-cart-nested-stack --capabilities CAPABILITY_IAM CAPABILITY_AUTO_EXPAND`	DevOps 工程師

任務	描述	所需技能
	該命令將提示幾個問題。回答所有問題y。	

驗證部署

任務	描述	所需技能
驗證堆疊。	若要檢閱 AWS SAM 範本中定義的 AWS CloudFormation 堆疊和 AWS 資源，請執行下列動作： 1. 登入 AWS 管理主控台，然後導覽至主CloudFormation控台。 2. 確認已列出父堆疊和子堆疊。 在此範例中，`sam-shopping-cart` 是呼叫巢狀驗證、產品和購物堆疊的父系堆疊。 產品堆疊會提供產品 API Gateway URL 連結做為輸出。	DevOps 工程師

相關資源

參考

- AWS Serverless Application Model (AWS SAM)
- AWS 上的山姆 GitHub
- 無伺服器購物車微服務 (AWS 範例應用程式)

教學課程和影片

- 建置無伺服器應用程式
- AWS 線上技術會談：使用 AWS SAM 建置和部署無伺服器應用程式

其他資訊

所有代碼到位後，該示例具有以下目錄結構：

- sam_stack — 此資料夾包含圖`shared.py`層。圖層是包含程式庫、自訂執行階段或其他相依性的檔案歸檔。透過圖層，您可以在函數中使用程式庫，而不需要將它們包含在部署套件中。
- product-mock-service— 此資料夾包含所有與產品相關的 Lambda 函數和檔案。
- shopping-cart-service— 此資料夾包含所有與購物相關的 Lambda 函數和檔案。

使用 AWS Lambda 權杖自動販賣機為 Amazon S3 實作 SaaS 租用戶隔離

由虎斑病房（AWS），斯拉文·佩里亞坦比（AWS）和托馬斯·戴維斯（AWS）創建

環境：PoC 或試點	技術：現代化；軟體 SaaS	AWS 服務：AWS Identity and Access Management；AWS Lambda；Amazon S3；AWS STS

Summary

多租戶 SaaS 應用程式必須實作系統，以確保維護租用戶隔離。當您將租用戶資料存放在相同的 Amazon Web Services (AWS) 資源 (例如多個租用戶在同一個 Amazon Simple Storage Service (Amazon S3) 貯體中存放資料時，您必須確保不會發生跨租戶存取。令牌自動售貨機（TVM）是提供租戶數據隔離的一種方法。這些機器提供了一種獲取令牌的機制，同時抽象生成這些令牌的複雜性。開發人員可以使用 TVM，而無需詳細了解它如何生成令牌。

此模式使用 AWS Lambda 實作 TVM。TVM 會產生一個包含臨時安全性權杖服務 (STS) 登入資料的權杖，這些登入資料會限制對 S3 儲存貯體中單一 SaaS 租用戶資料的存取。

TVM 和此模式提供的程式碼通常與衍生自 JSON Web Token (JWT) 的宣告搭配使用，以將 AWS 資源的請求與承租人範圍的 AWS Identity and Access Management (IAM) 政策建立關聯。您可以使用此模式中的代碼作為基礎來實現 SaaS 應用程序，該應用程序根據 JWT 令牌中提供的聲明生成範圍的臨時 STS 憑據。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 1.19.0 版或更新版本，已在 macOS、Linux 或視窗上安裝和設定。或者，您也可以使用 AWS CLI 2.1 版或更新版本。

限制

- 此代碼以 Java 運行，目前不支持其他編程語言。
- 範例應用程式不包含 AWS 跨區域或災難復原 (DR) 支援。
- 此模式示範 SaaS 應用程式的 Lambda TVM 如何提供範圍的租用戶存取。它不打算用於生產環境。

架構

目標技術堆疊

- AWS Lambda
- Amazon S3
- IAM
- AWS Security Token Service (AWS STS)

目標架構

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Security Token Service (AWS STS) 可協助您為使用者申請臨時、有限權限的登入資料。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

此模式的原始程式碼可作為附件使用，並包含下列檔案：

- `s3UploadSample.jar`提供 Lambda 函數的原始程式碼，該函數會將 JSON 文件上傳到 S3 儲存貯體。
- `tvm-layer.zip`提供可重複使用的 Java 程式庫，為 Lambda 函數提供權杖 (STS 臨時登入資料)，以存取 S3 儲存貯體並上傳 JSON 文件。
- `token-vending-machine-sample-app.zip`提供用來建立這些成品和編譯指示的原始程式碼。

若要使用這些檔案，請遵循下一節中的指示。

史诗

確定變量值

任務	描述	所需技能
確定變量值。	此模式的實現包括幾個必須一致使用的變量名稱。決定應用於每個變數的值，並在後續步驟中提出要求時提供該值。 – <AWS Account ID>與您實作此模式的 AWS 帳戶相關聯的 12 位數帳戶 ID。如需如何尋找 AWS cccount ID 的相關資訊，請參閱 IAM 文件中的 AWS 帳戶 ID 及其別名。 – <AWS Region>您在其中實作此模式的 AWS 區域。如需 AWS 區域的詳細資訊，請參閱 AWS 網站上的區域和可用區域。 < sample-tenant-name > – 要在應用程式中使用的租用戶名稱。為了簡單起見，建議您僅在此值中使用英數字元，但是	雲端管理員

任務	描述	所需技能
	您可以使用任何有效的名稱作為 S3 物件金鑰。 < sample-tvm-role-name > – 附加至執行 TVM 和範例應用程式之 Lambda 函數的 IAM 角色名稱。角色名稱是由不含空格的大寫和小寫字母數字字元組成的字串。您也可以包含下列任何字元：底線 (_)、加號 (+)、等號 (=)、逗號 (,)、句點 (.)、@符號和連字號 (-)。角色名稱在帳戶內必須是唯一的。 < sample-app-role-name > – 當 Lambda 函數產生範圍的暫時 STS 登入資料時，所假設的 IAM 角色名稱。角色名稱是由不含空格的大寫和小寫字母數字字元組成的字串。您也可以包含下列任何字元：底線 (_)、加號 (+)、等號 (=)、逗號 (,)、句點 (.)、@符號和連字號 (-)。角色名稱在帳戶內必須是唯一的。 < sample-app-function-name > – Lambda 函數的名稱。這是一個長度最多 64 個字符的字符串。 < sample-app-bucket-name > – 必須使用範圍為特定租用戶的許可存取的 S3 儲存貯體的名稱。S3 儲存貯體名稱：	

任務	描述	所需技能
	• 長度必須介於 3 與 63 個字元之間。 • 必須僅由小寫字母、數字、句點 (.) 和連字號 (-) 組成。 • 必須以字母或數字開頭和結尾。 • 不得採用 IP 地址格式 (例如，192.168.5.4)。 • 在分割區內必須是唯一的。分割區是區域的群組。AWS 目前有三個分區：aws (標準區域)、aws-cn (中國區域) 和 aws-us-gov (AWS GovCloud [US] 區域)。	

建立 S3 儲存貯體

任務	描述	所需技能
為範例應用程式建立 S3 儲存貯體。	使用下列 AWS CLI 命令建立 S3 儲存貯體。在程式碼片段中提供 < sample-app-bucket-name > 值： `aws s3api create-bucket --bucket <sample-app-bucket-name>` Lambda 範例應用程式會將 JSON 檔案上傳到此儲存貯體。	雲端管理員

建立 IAM TVM 角色和政策

任務	描述	所需技能
建立 TVM 角色。	使用下列其中一個 AWS CLI 命令建立 IAM 角色。在命令中提供 < sample-tvm-role-name > 值。 對於 macOS 或外殼： `aws iam create-role \ --role-name <sample-tvm-role-name> \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ]}'` 對於視窗命令列： `aws iam create-role ^ --role-name <sample-tvm-role-name> ^ --assume-role-policy-document "{\"Version\": \"2012-10`	雲端管理員

任務	描述	所需技能
	`-17\", \"Statement \": [{\"Effect\": \"Allow\", \"Principal\": {\"Service\": \"lambda.amazonaws.com\"}, \"Action\": \"sts:AssumeRole\" }]}"` 呼叫應用程式時，Lambda 範例應用程式會擔任此角色。假設具有範圍政策的應用程式角色的功能，可為程式碼提供更廣泛的存取 S3 儲存貯體的許可。	

任務	描述	所需技能
建立內嵌 TVM 角色原則。	使用下列其中一個 AWS CLI 命令建立 IAM 政策。在指令中提供 < sample-app-role-name >、和 < > 值。sample-tvm-role-name <AWS Account ID> 對於 macOS 或外殼： `aws iam put-role-policy \ --role-name <sample-tvm-role-name> \ --policy-name assume-app-role \ --policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "sts:AssumeRole", "Resource": "arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>" } ]}'` 對於視窗命令列： `aws iam put-role-policy ^ --role-name <sample-tvm-role-name> ^ --policy-name assume-app-role ^`	雲端管理員

任務	描述	所需技能
	--policy-documen t "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow \", \"Action\": \"sts:AssumeRole \", \"Resource\": \"arn:aws:iam::<AW S Account ID>:role/ <sample-app-role-n ame>\"}]}" 此原則會附加至 TVM 角色。它為程式碼提供了承擔應用程式角色的能力，該角色具有更廣泛的存取 S3 儲存貯體的權限。	

任務	描述	所需技能
附加受管理的 Lambda 政策。	使用下列 AWS CLI 命令附加 AWSLambdaBasicExecutionRole IAM 政策。在命令中提供 < sample-tvm-role-name > 值： `aws iam attach-role-policy \ --role-name <sample-tvm-role-name> \ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` 對於視窗命令列： `aws iam attach-role-policy ^ --role-name <sample-tvm-role-name> ^ --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` 此受管政策附加到 TVM 角色，以允許 Lambda 將日誌傳送到 Amazon CloudWatch。	雲端管理員

建立 IAM 應用程式角色和政策

任務	描述	所需技能
建立應用程式角色。	使用下列其中一個 AWS CLI 命令建立 IAM 角色。在指令中	雲端管理員

任務	描述	所需技能
	提供 < sample-tvm-role-name >、和 < > 值。sample-app-role-name <AWS Account ID> 對於 macOS 或外殼： `aws iam create-role \ --role-name <sample-app-role-name> \ --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>" }, "Action": "sts:AssumeRole" } ]}'` 對於視窗命令列： `aws iam create-role ^ --role-name <sample-app-role-name> ^ --assume-role-policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow`	

任務	描述	所需技能
	\",\"Principal\": {\"AWS\": \"arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name>\"},\"Action\": \"sts:AssumeRole\" }]}" Lambda 範例應用程式假設此角色具有範圍政策，以取得 S3 儲存貯體的租戶型存取權。	

任務	描述	所需技能
建立內嵌應用程式角色原則。	使用下列其中一個 AWS CLI 命令建立 IAM 政策。在指令中提供 < sample-app-role-name sample-app-bucket-name > 和 < > 值。 對於 macOS 或外殼： `aws iam put-role-policy \ --role-name <sample-app-role-name> \ --policy-name s3-bucket-access \ --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"], "Resource": "arn:aws:s3:::<sample-app-bucket-name>/*" }, { "Effect": "Allow", "Action": ["s3:ListBucket"],`	雲端管理員

任務	描述	所需技能
	<pre>"Resource ": "arn:aws:s3:::<sam ple-app-bucket-name>" }]}'</pre> 對於視窗命令列： <pre>aws iam put-role-policy ^ --role-name <sample-a pp-role-name> ^ --policy-name s3-bucket -access ^ --policy-documen t "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow \", \"Action\": [\"s3:PutObject\", \"s3:GetObject\", \"s3:DeleteObject\ "], \"Resource\": \"arn:aws:s3:::<sa mple-app-bucket-na me>/*\"}, {\"Effect\": \"Allow\", \"Action\ ": [\"s3:ListBucket \"], \"Resource\": \"arn:aws:s3:::<sa mple-app-bucket-name> \"}]}"</pre> 此原則會附加至應用程式角色。它提供對 S3 儲存貯體中物件的廣泛存取。當範例應用程式擔任角色時，這些權限的	

任務	描述	所需技能
	範圍會限制為具有 TVM 動態產生原則的特定承租人。	

使用 TVM 建立 Lambda 範例應用程式

任務	描述	所需技能
下載編譯後的源文件。	下載包含為附件的`s3UploadSample.jar` 和`tvm-layer.zip` 檔案。中提供了用來建立這些成品和編譯實例的原始程式碼。`token-vending-machine-sample-app.zip`	雲端管理員
建立 Lambda 圖層。	使用下列 AWS CLI 命令建立 Lambda 層，讓 Lambda 能夠存取 TVM。 附註：如果您不是從下載的位置執行此命令 `tvm-layer.zip` ，請在`--zip-file` 參數`tvm-layer.zip` 中提供正確的路徑。 `aws lambda publish-layer-version \ --layer-name sample-token-vending-machine \ --compatible-runtimes java11 \ --zip-file fileb://tvm-layer.zip` 對於視窗命令列：	雲管理員，應用程序開發

任務	描述	所需技能
	`aws lambda publish-layer-version ^ --layer-name sample-token-vending-machine ^ --compatible-runtimes java11 ^ --zip-file fileb://tvm-layer.zip` 此命令會建立一個 Lambda 層，其中包含可重複使用的 TVM 程式庫。	

任務	描述	所需技能
建立 Lambda 函數。	使用下列 AWS CLI 命令建立 Lambda 函數。在指令中提供 < sample-tvm-role-name >、、、< sample-app-bucket-name >、以及 < sample-app-role-name > 值。sample-app-function-name <AWS Account ID><AWS Region> 附註：如果您不是從下載的位置執行此命令`s3UploadSample.jar`，請在`--zip-file`參數`s3UploadSample.jar`中提供正確的路徑。 `aws lambda create-function \ --function-name <sample-app-function-name> \ --timeout 30 \ --memory-size 256 \ --runtime java11 \ --role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> \ --handler com.amazon.aws.s3UploadSample.App \ --zip-file fileb://s3UploadSample.jar \ --layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer:sample-token-vending-machine:1 \`	雲管理員，應用程序開發

任務	描述	所需技能
	--environment "Variables={S3_BUCKET=<sample-app-bucket-name>, ROLE=arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>}" 對於視窗命令列： aws lambda create-function ^ --function-name <sample-app-function-name> ^ --timeout 30 ^ --memory-size 256 ^ --runtime java11 ^ --role arn:aws:iam::<AWS Account ID>:role/<sample-tvm-role-name> ^ --handler com.amazon.aws.s3UploadSample.App ^ --zip-file fileb://s3UploadSample.jar ^ --layers arn:aws:lambda:<AWS Region>:<AWS Account ID>:layer:sample-token-vending-machine:1 ^ --environment "Variables={S3_BUCKET=<sample-app-bucket-name>,ROLE=arn:aws:iam::<AWS Account ID>:role/<sample-app-role-name>}"	

任務	描述	所需技能
	這個命令會建立一個 Lambda 函數，並附加了範例應用程式程式碼和 TVM 層。它還設置了兩個環境變量：S3_BUCKET 和ROLE。範例應用程式會使用這些變數來決定要承擔的角色，以及要將 JSON 文件上傳到的 S3 儲存貯體。	

測試範例應用程式和 TVM

任務	描述	所需技能
叫用 Lambda 範例應用程式。	使用下列其中一個 AWS CLI 命令，以預期的承載啟動 Lambda 範例應用程式。在指令中提供 < sample-app-function-name sample-tenant-name > 和 < > 值。 對於 macOS 和外殼： `aws lambda invoke \ --function <sample-app-function-name> \ --invocation-type RequestResponse \ --payload '{"tenant": "<sample-tenant-name>"}' \ --cli-binary-format raw-in-base64-out response.json` 對於視窗命令列：	雲管理員，應用程序開發

任務	描述	所需技能
	`aws lambda invoke ^ --function <sample-app-function-name> ^ --invocation-type RequestResponse ^ --payload "{\"tenant\": \"<sample-tenant-name>\"}" ^ --cli-binary-format raw-in-base64-out response.json` 此命令會呼叫 Lambda 函數，並在`response.json` 文件中傳回結果。在許多 UNIX 系統上，您可以變更為以將結果直接輸出`response.json` / `dev/stdout` 到殼層，而無需建立其他檔案。 注意：在後續呼叫此 Lambda 函數中變更 < sample-tenant-name > 值會變更 JSON 文件的位置以及權杖提供的權限。	
檢視 S3 儲存貯體以查看建立的物件。	瀏覽至您先前建立的 S3 儲存貯體 (< sample-app-bucket-name >)。此儲存貯體包含值為 < sample-tenant-name > 的 S3 物件前綴。在該前綴下，您將找到一個以 UUID 命名的 JSON 文檔。多次叫用範例應用程式會新增更多 JSON 文件。	雲端管理員

任務	描述	所需技能
檢視範例應用程式的雲觀察日誌。	檢視與名為 < sample-app-function-name > 的 Lambda 函數相關聯的雲端觀察日誌。如需指示，請參閱 AWS Lambda 文件中的存取 AWS Lambda 的 Amazon CloudWatch 日誌。您可以在這些記錄中檢視 TVM 產生的承租人範圍原則。此承租人範圍政策授予 Amazon S3 PutObject、、和 ListBucketAPI 的範例應用程式許可 GetObjectDeleteObject，但僅限於與 < > 關聯的物件前綴。sample-tenant-name在範例應用程式的後續呼叫中，如果您變更 < sample-tenant-name >，TVM 會更新範圍原則，以對應於呼叫承載中提供的租用戶。此動態產生的原則顯示如何透過 SaaS 應用程式中的 TVM 維護承租人範圍的存取。 TVM 功能在 Lambda 層中提供，因此它可以連接到應用程式所使用的其他 Lambda 函數，而無需複寫程式碼。 如需動態產生原則的圖例，請參閱其他資訊一節。	雲端管理員

相關資源

- 使用動態產生的 IAM 政策隔離租用戶 (部落格文章)

- 在 SaaS 環境中套用動態產生的隔離原則 (部落格文章)
- AWS SaaS 升壓 (可協助您將 SaaS 產品移至 AWS 的開放原始碼參考環境)

其他資訊

下列 Amazon Cloudwatch 日誌會以此模式顯示由 TVM 程式碼產生的動態產生的政策。在此屏幕截圖中，< sample-app-bucket-name > 是DOC-EXAMPLE-BUCKET和 < sample-tenant-name > 是test-tenant-1。此範圍政策傳回的 STS 登入資料無法對 S3 儲存貯體中的物件執行任何動作，但與物件 key prefix 相關聯的物件除外。test-tenant-1

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Step Functions 實作無伺服器傳奇模式

由虎斑病房（AWS），羅漢·梅赫塔（AWS）和雲紋特瓦尼（AWS）創建

環境：PoC 或試點	技術：現代化、無伺服器、雲端原生	工作負載：開源
AWS 服務：Amazon API Gateway；Amazon DynamoDB；AWS Lambda；Amazon SNS；AWS Step Functions		

Summary

在微服務架構中，主要目標是建立分離且獨立的元件，以提升應用程式的敏捷性、彈性和更快的上市時間。解耦的結果是，每個微服務元件都有自己的資料持續性層。在分散式架構中，商業交易可以跨越多個微服務。由於這些微服務不能使用單個原子性，一致性，隔離性，耐久性（ACID）事務，因此您最終可能會得到部分事務。在這種情況下，需要一些控制邏輯來復原已經處理的交易。分佈式傳奇模式通常用於此目的。

saga 模式是一種失敗管理模式，可協助建立分散式應用程式的一致性，並協調多個微服務之間的交易，以維持資料一致性。當您使用 saga 模式時，每個執行交易的服務都會發佈觸發後續服務的事件，以執行鏈結中的下一個交易。這會持續到鏈結中的最後一個交易完成為止。如果商業交易失敗，saga 會協調一系列補償交易，這些交易會復原先前交易所做的變更。

此模式示範如何使用 AWS 步驟函數、AWS Lambda 和 Amazon DynamoDB 等無伺服器技術，自動化範例應用程式 (用於處理旅行預訂) 的設定和部署。範例應用程式還使用 Amazon API Gateway 和亞馬遜簡單通知服務 (Amazon SNS) 來實作傳奇執行協調器。您可以使用基礎設施即程式碼 (IaC) 架構來部署該模式，例如 AWS Cloud Development Kit (AWS CDK)、AWS Serverless Application Model (AWS SAM) 或地形。

如需傳奇模式和其他資料持續性模式的詳細資訊，請參閱 AWS Prescriptive Guidance 網站上的啟用微型服務中的資料持續性指南。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 建立 AWS CloudFormation 堆疊的許可。如需詳細資訊，請參閱 CloudFormation 文件中的控制存取。
- 您選擇的 IaC 框架 (AWS CDK、AWS SAM 或 Terraform) 使用您的 AWS 帳戶設定，以便您可以使用框架 CLI 部署應用程式。
- NodeJS，用於構建應用程序並在本地運行它。
- 您選擇的代碼編輯器（例如視覺工作室代碼，崇高或原子）。

產品版本

- NodeJS 版本 14
- AWS CDK 版本 2.37.1
- AWS 山姆版本 1.71.0
- 地形文字版本 1.3.7

限制

事件來源是在微服務架構中實現 saga 協調流程模式的一種自然方式，其中所有組件都鬆散耦合，並且彼此沒有直接知識。如果您的交易涉及少量步驟（三到五個），那麼傳奇模式可能非常適合。然而，複雜性隨著微服務的數量和步驟的數量而增加。

當您使用此設計時，測試和偵錯可能會變得困難，因為您必須讓所有服務都執行，才能模擬交易模式。

架構

目標架構

提議的架構使用 AWS Step Functions 建立傳奇模式來預訂航班、預訂租車和處理假期付款。

以下工作流程圖說明了旅行預訂系統的典型流程。該工作流程包括預訂航空旅行（「ReserveFlight」），預訂汽車（「ReserveCarRental」），處理付款（「ProcessPayment」），確認航班預訂（「ConfirmFlight」）以及確認汽車租賃（「ConfirmCarRental」），然後在完成這些步驟時發出成功通知。但是，如果系統在運行任何這些事務時遇到任何錯誤，它開始向後失敗。例如，付款處理錯誤（「ProcessPayment」）會觸發退款（「RefundPayment」），然後觸發取消租車和航班（「CancelRentalReservation」和「CancelFlightReservation」），結束整筆交易並顯示失敗訊息。

此模式會為圖中反白顯示的每個任務部署個別的 Lambda 函數，以及三個用於航班、汽車租賃和付款的 DynamoDB 表。每個 Lambda 函數會建立、更新或刪除個別 DynamoDB 表格中的資料列，具體取決於交易是否已確認或回復。該模式使用 Amazon SNS 向訂閱者傳送文字 (SMS) 訊息，通知他們交易失敗或成功。

自動化和規模

您可以使用其中一個 IaC 框架來創建此架構的配置。使用以下鏈接之一為您首選的 IaC。

- 使用 AWS CDK 進行部署
- 使用 AWS SAM 進行部署
- 使用地形進行部署

工具

AWS 服務

- AWS Step Functions 是一種無伺服器協調服務，可讓您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。透過 Step Functions 圖形化主控台，您可以將應用程式的工作流程視為一系列事件驅動的步驟。
- Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。您可以使用 DynamoDB 建立資料庫資料表，藉此存放和擷取任意數量的資料，並為任何層級的請求流量提供服務。
- AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon API Gateway 是一種 AWS 服務，用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者傳送訊息給訂閱者的訊息。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可使用熟悉的程式設計語言 (例如 TypeScript、Python JavaScript、Java 和 C#/.net) 來定義雲端應用程式資源。
- AWS Serverless Application Model (AWS SAM) 是用來建置無伺服器應用程式的開放原始碼架構。它提供速記語法來表示函數、API、資料庫和事件來源對應。

Code

示範傳奇模式的範例應用程式程式程式碼，包括 IaC 範本 (AWS CDK、AWS SAM 或地形)、Lambda 函數和 DynamoDB 表格，可在下列連結中找到。請按照第一部史詩中的說明進行安裝。

- 使用 AWS CDK 進行部署
- 使用 AWS SAM 進行部署
- 使用地形進行部署

史诗

安裝套件、編譯和建置

任務	描述	所需技能
安裝 NPM 套件。	建立新目錄、瀏覽至終端機中的該目錄，然後從此模式稍早的程式碼區段複製您選擇的 GitHub 存放庫。 在具有`package.json` 檔案的根資料夾中，執行下列命令，以下載並安裝所有節點 Package 件管理員 (NPM) 套件： `npm install`	雲端架構師開發人員
編譯腳本。	在根資料夾中，執行下列命令以指示 TypeScript 轉譯器建立所有必要 JavaScript 的檔案： `npm run build`	雲端架構師開發人員
注意更改並重新編譯。	在根資料夾中，在個別的終端機視窗中執行下列命令以監視	雲端架構師開發人員

任務	描述	所需技能
	程式碼變更，並在偵測到變更時編譯程式碼： `npm run watch`	
執行單元測試 (僅限 AWS CDK)。	如果您使用的是 AWS CDK，請在根文件夾中運行以下命令來執行 Jest 單元測試： `npm run test`	雲端架構師開發人員

將資源部署到目標 AWS 帳戶

任務	描述	所需技能
將示範堆疊部署到 AWS。	重要：此應用程式與 AWS 區域無關。如果您使用設定檔，則必須在 AWS Command Line Interface (AWS CLI) (AWS CLI) 設定檔或透過 AWS CLI 環境變數明確宣告區域。 在根資料夾中，執行下列命令以建立部署組件，並將其部署到預設 AWS 帳戶和區域。 AWS 磁碟機： `cdk bootstrap` `cdk deploy` AWS 山姆： `sam build` `sam deploy --guided`	雲端架構師開發人員

任務	描述	所需技能
	地形： `terraform init` `terraform apply` 此步驟可能需要幾分鐘才能完成。此命令使用針對 AWS CLI 設定的預設登入資料。 請注意部署完成後顯示在主控台上的 API Gateway URL。您將需要此信息來測試傳奇執行流程。	
將已部署的堆疊與目前狀態進行比較。	在根資料夾中，執行下列命令，在對原始程式碼進行變更後，將已部署的堆疊與目前狀態進行比較： AWS 磁碟機： `cdk diff` AWS 山姆： `sam deploy` 地形： `terraform plan`	雲端架構師開發人員

測試執行流程

任務	描述	所需技能
測試傳奇執行流程。	導覽至您在部署堆疊時在先前步驟中記下的 API Gateway URL。此 URL 會觸發狀態機啟動。如需有關如何透過傳遞不同 URL 參數來操作狀態機器流程的詳細資訊，請參閱其他資訊一節。 若要檢視結果，請登入 AWS 管理主控台並導覽至 Step Functions 主控台。在這裡，您可以看到傳奇狀態機的每一步。您也可以檢視 DynamoDB 表格，以查看插入、更新或刪除的記錄。如果您經常刷新屏幕，則可以觀察交易狀態從更改`pending`為`confirmed` 。 您可以通過使用手機號碼更新`stateMachine.ts` 文件中的代碼來訂閱 SNS 主題，以便在預訂成功或失敗時接收 SMS 消息。如需詳細資訊，請參閱其他資訊一節中的 Amazon SNS。	雲端架構師開發人員

清除

任務	描述	所需技能
清理資源。	若要清除為此應用程式部署的資源，您可以使用下列命令之一。	應用程式開發人員、雲端

任務	描述	所需技能
	AWS 磁碟機： `cdk destroy` AWS 山姆： `sam delete` 地形： `terraform destroy`	

相關資源

技術論文

- 在 AWS 上實作微型服務
- 無伺服器應用鏡頭
- 在微服務中啟用資料持續性

AWS 服務文件

- 開始使用 AWS CDK
- 開始使用 AWS SAM
- AWS Step Functions
- Amazon DynamoDB
- AWS Lambda
- Amazon API Gateway
- Amazon SNS

教學課程

- 無伺服器運算實作研討會

其他資訊

Code

基於測試目的，此模式會部署 API Gateway 和觸發 Step Functions 函數狀態機器的測試 Lambda 函數。使用 Step Functions，您可以通過傳遞`run_type`參數來模仿「，」，「」，「」ReserveFlight，「」和「。」中的故障來控制旅行預訂系統的功能ConfirmCarRental。ReserveCarRental ProcessPayment ConfirmFlight

`sagaLambda` 函數 (`sagaLambda.ts`) 會從 API Gateway URL 中的查詢參數接收輸入，建立下列 JSON 物件，並將其傳遞至 Step Functions 數以執行：

```
let input = {
"trip_id": tripID, //  value taken from query parameter, default is AWS request ID
"depart_city": "Detroit",
"depart_time": "2021-07-07T06:00:00.000Z",
"arrive_city": "Frankfurt",
"arrive_time": "2021-07-09T08:00:00.000Z",
"rental": "BMW",
"rental_from": "2021-07-09T00:00:00.000Z",
"rental_to": "2021-07-17T00:00:00.000Z",
"run_type": runType // value taken from query parameter, default is "success"
};
```

您可以通過傳遞以下 URL 參數與 Step Functions 數狀態機的不同流程進行實驗：

- 成功執行 – https://{api 閘道網址}
- 預訂航班失敗 – https://{api 網關網址}？ 執行類型 = failFlightsReservation
- 確認航班失敗 – https://{api 網關網址}？ 執行類型 = failFlightsConfirmation
- 保留汽車租賃失敗 – https://{api 網關網址}？ 執行類型 = 預failCarRental約
- 確認租車失敗 – https://{api 網關網址}? 執行類型 = 確認 failCarRental
- 處理付款失敗 – https://{api 網關網址}？ 執行類型 = 失敗付款
- 通過旅程識別碼 – https://{api 網關網址}？ TripID = {依預設，行程 ID 將成為 AWS 請求識別碼}

合家歡模板

鏈接的存儲庫包括 IaC 模板，您可以使用它們來創建整個樣本旅行預訂應用程序。

- 使用 AWS CDK 進行部署
- 使用 AWS SAM 進行部署
- 使用地形進行部署

DynamoDB 資料表

以下是航班，租車和付款表的數據模型。

```
Flight Data Model:
 var params = {
      TableName: process.env.TABLE_NAME,
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: flightReservationID},
        'trip_id' : {S: event.trip_id},
        'id': {S: flightReservationID},
        'depart_city' : {S: event.depart_city},
        'depart_time': {S: event.depart_time},
        'arrive_city': {S: event.arrive_city},
        'arrive_time': {S: event.arrive_time},
        'transaction_status': {S: 'pending'}
      }
    };

Car Rental Data Model:
var params = {
      TableName: process.env.TABLE_NAME,
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: carRentalReservationID},
        'trip_id' : {S: event.trip_id},
        'id': {S: carRentalReservationID},
        'rental': {S: event.rental},
        'rental_from': {S: event.rental_from},
        'rental_to': {S: event.rental_to},
        'transaction_status': {S: 'pending'}
      }
    };

Payment Data Model:
var params = {
      TableName: process.env.TABLE_NAME,
```

```
      Item: {
        'pk' : {S: event.trip_id},
        'sk' : {S: paymentID},
        'trip_id' : {S: event.trip_id},
        'id': {S: paymentID},
        'amount': {S: "750.00"}, // hard coded for simplicity as implementing any
 monetary transaction functionality is beyond the scope of this pattern
        'currency': {S: "USD"},
        'transaction_status': {S: "confirmed"}
      }
    };
```

Lambda 函數

將創建以下功能以支持狀態機流和 Step Functions 中執行：

- 預訂航班：將記錄插入 DynamoDB 航班表格中，並附有`transaction_status`的`pending`，以預訂航班。
- 確認航班：更新 DynamoDB 航班表格中的記錄，以設定`transaction_status`為`confirmed`，以確認航班。
- 取消航班預訂:刪除 DynamoDB 航班表格中的記錄，以取消擱置的航班。
- 預約租車：將記錄插入 DynamoDB CarRentals 表格中，並附上`transaction_status`的`pending`，以預訂租車服務。
- 確認租車：更新 DynamoDB CarRentals 表格中的記錄，以設定`transaction_status`為`confirmed`，以確認租車。
- 取消租車預約：刪除 DynamoDB CarRentals 表格中的記錄，以取消待處理的租車。
- 處理付款：將記錄插入付款的 DynamoDB 付款表格中。
- 取消付款：從 DynamoDB 付款表中刪除付款的記錄。

Amazon SNS

範例應用程式會建立下列主題和訂閱，以傳送 SMS 訊息，並通知客戶保留成功或失敗。如果您想要在測試範例應用程式時接收簡訊，請使用狀態機器定義檔案中的有效電話號碼更新 SMS 訂閱。

AWS CDK 程式碼片段 (在下列程式碼的第二行新增電話號碼)：

```
const topic = new  sns.Topic(this, 'Topic');
topic.addSubscription(new subscriptions.SmsSubscription('+11111111111'));
```

```
const snsNotificationFailure = new tasks.SnsPublish(this ,'SendingSMSFailure', {
topic:topic,
integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
message: sfn.TaskInput.fromText('Your Travel Reservation Failed'),
});

const snsNotificationSuccess = new tasks.SnsPublish(this ,'SendingSMSSuccess', {
topic:topic,
integrationPattern: sfn.IntegrationPattern.REQUEST_RESPONSE,
message: sfn.TaskInput.fromText('Your Travel Reservation is Successful'),
});
```

AWS SAM 程式碼片段 (以您的有效電話號碼取代+1111111111字串)：

```
  StateMachineTopic11111111111:
    Type: 'AWS::SNS::Subscription'
    Properties:
      Protocol: sms
      TopicArn:
        Ref: StateMachineTopic
      Endpoint: '+11111111111'
    Metadata:
      'aws:sam:path': SamServerlessSagaStack/StateMachine/Topic/+11111111111/Resource
```

Terraform 片段（用您的有效電話號碼替換+111111111字符串）：

```
resource "aws_sns_topic_subscription" "sms-target" {
  topic_arn = aws_sns_topic.topic.arn
  protocol  = "sms"
  endpoint  = "+11111111111"
}
```

成功預訂

下列流程說明成功的預約ReserveCarRental，其中包含「,ProcessPayment」、「ConfirmFlight」和「」ConfirmCarRental。ReserveFlight透過傳送給 SNS 主題訂閱者的 SMS 訊息，向客戶收到有關成功預約的通知。

保留失敗

此流程是傳奇模式中失敗的一個例子。如果在預訂航班和租車之後，「ProcessPayment」失敗，步驟將以相反的順序取消。 保留已釋出，並透過傳送給 SNS 主題訂閱者的 SMS 訊息通知客戶失敗。

使用 AWS CDK 在任何地方設定 Amazon ECS 來管理現場部署容器應用程式

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）創建

代碼存儲庫：amazon-ecs-anywhere-cdk-示例	環境：PoC 或試點	技術：現代化；容器與微服務；混合雲 DevOps；基礎架構
工作負載：所有其他工作	AWS 服務：AWS CDK；Amazon ECS；AWS Identity and Access Management	

Summary

Amazon ECS Anywhere 是 Amazon Elastic Container Service (Amazon ECS) 的擴展。您可以使用 ECS 無處不在，在現場部署或客戶管理的環境中部署原生 Amazon ECS 任務。此功能有助於降低成本並減少複雜的本機容器協調和作業。您可以使用 ECS Anywhere 在內部部署和雲端環境中部署和執行容器應用程式。您的團隊不需要學習多個領域和技能，或者自行管理複雜的軟體。

此模式示範使用 AWS Cloud Development Kit (AWS CDK) 堆疊設定 ECS 任何地方的步驟。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI。)
- 已安裝和設定的 AWS CDK 工具組。(請參閱 AWS CDK 文件中的 AWS CDK 工具組，並依照指示在全球安裝第 2 版。)
- 節點套件管理員 (npm)，已在 TypeScript.（請參閱 npm 文檔中的下載和安裝 Node.js 和 npm。）

限制

- 如需限制和考量事項，請參閱 Amazon ECS 文件中的外部執行個體 (Amazon ECS 無處不在)。

產品版本

- AWS CDK 工具組第 2 版
- 故宮版本 7.20.3 或更新版本
- Node.js 版本 16.6.1 或更新版本

架構

目標技術堆疊

- AWS CloudFormation
- AWS CDK
- Amazon ECS Anywhere
- AWS Identity and Access Management (IAM)

目標架構

下圖說明使用 AWS CDK 進行 ECS Anywhere 設定的高階系統架構 TypeScript，如此模式所實作。

1. 當您部署 AWS CDK 堆疊時，它會在 AWS 上建立一個 CloudFormation 堆疊。
2. 該 CloudFormation 堆疊佈建了一個 Amazon ECS 叢集和相關的 AWS 資源。
3. 若要向 Amazon ECS 叢集註冊外部執行個體，您必須在虛擬機器 (VM) 上安裝 AWS Systems Manager 代理程式 (SSM 代理程式)，並將該 VM 註冊為 AWS Systems Manager 受管執行個體。
4. 您也必須在虛擬機器上安裝 Amazon ECS 容器代理程式和 Docker，才能將其註冊為 Amazon ECS 叢集的外部執行個體。
5. 使用 Amazon ECS 叢集註冊和設定外部執行個體後，它可以在已註冊為外部執行個體的 VM 上執行多個容器。

自動化和規模

此模式提供的GitHub 存放庫使用 AWS CDK 做為基礎設施即程式碼 (IaC) 工具來建立此架構的組態。AWS CDK 可協助您協調資源並在任何地方設定 ECS。

工具

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

Code

此模式的原始程式碼可在 GitHub Amazon ECS Anywhere 不在 CDK 範例儲存庫中取得。若要複製並使用儲存庫，請遵循下一節中的指示。

史诗

驗證 AWS CDK 組態

任務	描述	所需技能
驗證 AWS CDK 版本。	執行下列命令來驗證 AWS CDK 工具組的版本： `cdk --version` 此模式需要使用 AWS CDK 第 2 版。如果您使用的是舊版 AWS CDK，請按照 AWS CDK 文件中的指示進行更新。	DevOps 工程師
設定 AWS 登入資料。	若要設定認證，請執行命令`aws configure`並依照提示執行： `$aws configure` `AWS Access Key ID [None]: <your-access-key-ID>` `AWS Secret Access Key [None]: <your-secret-access-key>`	DevOps 工程師

任務	描述	所需技能
	`Default region name [None]: <your-Region-name>` `Default output format [None]:`	

啟動 AWS CDK 環境

任務	描述	所需技能
複製 AWS CDK 程式碼儲存庫。	使用以下命令克隆此模式的 GitHub 代碼存儲庫： `git clone https://github.com/aws-samples/amazon-ecs-anywhere-cdk-samples.git`	DevOps 工程師
引導環境。	若要將 AWS CloudFormation 範本部署到您要使用的帳戶和 AWS 區域，請執行下列命令： `cdk bootstrap <account-number>/<Region>` 如需詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。	DevOps 工程師

建置和部署專案

任務	描述	所需技能
安裝套件相依性並編譯 TypeScript 檔案。	安裝套件相依性，並執行下列命令來編譯 TypeScript 檔案：	DevOps 工程師

任務	描述	所需技能
	`$cd amazon-ecs-anywher e-cdk-samples` `$npm install` `$npm fund` 這些指令會安裝範例存放庫中的所有套件。 重要事項：如果您收到有關遺失套件的任何錯誤，請使用下列其中一個指令： `$npm ci` —或— `$npm install -g @aws-cdk/<package_name>` 如需詳細資訊，請參閱 npm 文件中的 npm ci 和 npm 安裝。	
建置專案。	若要建置專案程式碼，請執行下列命令： `npm run build` 如需有關建置和部署專案的詳細資訊，請參閱 AWS CDK 文件中的第一個 AWS CDK 應用程式。	DevOps 工程師

任務	描述	所需技能
部署專案。	若要部署專案程式碼，請執行下列命令： `cdk deploy`	DevOps 工程師
驗證堆棧的創建和輸出。	在 https://console.aws.amazon.com/cloudformation 開啟 AWS CloudFormation 主控台，然後選擇`EcsAnywhereStack` 堆疊。 [輸出] 索引標籤會顯示要在外部 VM 上執行的命令。	DevOps 工程師

設定內部部署機器

任務	描述	所需技能
通過使用流浪者設置您的虛擬機。	出於演示目的，您可以使用HashiCorp 流浪者創建虛擬機。Vagrant 是用於構建和維護便攜式虛擬軟件開發環境的開源實用程序。通過從放置 Vagrantfile 的根目錄中運行`vagrant up`命令來創建一個流浪虛擬機。有關更多信息，請參閱流浪文檔。	DevOps 工程師
將您的 VM 註冊為外部執行個體。	1. 使用`vagrant ssh`命令登錄到流浪虛擬機。有關更多信息，請參閱流浪文檔。 2. 建立啟動碼和 ID，您可以用來向 AWS Systems Manager 註冊您的虛擬機器，以及啟用外部執行個體。此命	DevOps 工程師

任務	描述	所需技能
	令的輸出包括ActivationId 和ActivationCode 值： `aws ssm create-activation --iam-role EcsAnywhereInstanceRole | tee ssm-activation.json` 3. 匯出啟動 ID 和代碼值： `export ACTIVATION_ID=<activation-ID>` `export ACTIVATION_CODE=<activation-code>` 4. 將安裝指令碼下載到內部部署伺服器或 VM： `curl -o "ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh" && sudo chmod +x ecs-anywhere-install.sh` 5. 在內部部署伺服器或 VM 上執行安裝指令碼： `sudo ./ecs-anywhere-install.sh \` `--cluster test-ecs-anywhere \`	

任務	描述	所需技能
	--activation-id $ACTIVATION_ID \ --activation-code $ACTIVATION_CODE \ --region <Region> 如需有關設定和註冊虛擬機器的詳細資訊，請參閱 Amazon ECS 文件中的將外部執行個體註冊到叢集。	
驗證 ECS 無所不在和外部虛擬機器的狀態。	若要驗證虛擬盒是否已連接至 Amazon ECS 控制平面並執行，請使用下列命令： aws ssm describe-instance-information aws ecs list-container-instances --cluster $CLUSTER_NAME	DevOps 工程師

清除

任務	描述	所需技能
清理和刪除資源。	逐步完成此模式之後，您應該移除建立的資源，以避免產生任何進一步的費用。若要清理，請執行以下指令： cdk destroy	DevOps 工程師

相關資源

- Amazon ECS Anywhere 文檔

- Amazon ECS Anywhere 演示
- Amazon ECS Anywhere 車間樣品

在 AWS 上將 ASP.NET 網頁表單應用程式現代化

由阿南德·拉馬林甘（AWS）和拜縣斯雷拉克斯米（AWS）創建

環境：PoC 或試點	技術：現代化；容器與微服務；軟體開發與測試；Web 和行動應用程式	工作量：Microsoft
AWS 服務：Amazon CloudWatch; Amazon ECS; AWS Systems Manager		

Summary

此模式描述了將傳統的整體 ASP.NET Web 表單應用程式移植到 AWS 上的 ASP.NET 核心來現代化的步驟。

將 ASP.NET 網頁表單應用程式移植到 ASP.NET 核心可協助您充分利用 Linux 的效能、節省成本和強大的生態系統。但是，這可能是一個顯著的手動努力。在此模式中，舊版應用程式會使用階段化方法逐步現代化，然後在 AWS 雲端中進行容器化。

考慮購物車的舊版整體式應用程式。讓我們假設它是建立為 ASP.NET Web 表單應用程式，而且由具有程式碼隱藏 () 檔案的 .aspx 網頁所組成。`aspx.cs`現代化過程包含以下步驟：

1. 通過使用適當的分解模式將整體分解為微服務。如需詳細資訊，請參閱 AWS Prescriptive Guidance 網站上的將巨石分解為微型服務指南。
2. 將您的舊版 ASP.NET 網頁表單 (.NET 架構) 應用程式移植到 .NET 5 或更新版本中的 ASP.NET 核心。在此模式中，您可以使用移植助理 .NET 掃描您的 ASP.NET Web 表單應用程式，並識別與 ASP.NET 核心不相容性。這樣可以減少手動移植的工作量。
3. 通過使用反應重新開發 Web 窗體 UI 層。此模式不涵蓋 UI 重建。如需指示，請參閱 React 文件中的建立新的 React 應用程式。
4. 重新開發 Web 表單代碼隱藏文件（業務界面）作為 ASP.NET 核心 Web API。此病毒碼使用 NDepend 報告來協助識別必要的檔案和相依性。
5. 使用 .NET 的移植助理，將舊版應用程式中的共用/一般專案 (例如商務邏輯和資料存取) 升級為 .NET 5 或更新版本。

6. 新增 AWS 服務以補充您的應用程式。例如，您可以使用 Amazon CloudWatch 日誌來監控、存放和存取應用程式的日誌，而 AWS Systems Manager 則可以存放應用程式設定。
7. 容器化現代化的 ASP.NET 核心應用程式。這種模式創建一個碼頭文件，該文件以 Linux 為目標，並使用泊塢窗桌面在本地進行測試。此步驟假設您的舊版應用程式已在現場部署或 Amazon 彈性運算雲端 (Amazon EC2) Windows 執行個體上執行。如需詳細資訊，請參閱在 Amazon EC2 Linux 執行個體上執行 ASP.NET 核心網路 API 泊塢視窗容器模式。
8. 將現代化的 ASP.NET 核心應用程式部署到 Amazon Elastic Container Service (Amazon ECS)。此模式不涵蓋部署步驟。如需指示，請參閱 Amazon ECS 研討會。

附註：此模式不涵蓋 UI 開發、資料庫現代化或容器部署步驟。

先決條件和限制

先決條件

- 視覺工作室或視覺工作室代碼，下載並安裝。
- 使用 AWS 管理主控台和 AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版存取 AWS 帳戶。(請參閱有關設定 AWS CLI 的指示。)
- 適用於視覺工作室的 AWS 工具組 (請參閱設定指示)。
- 碼頭桌面，下載並安裝。
- . NET SDK，下載並安裝。
- n 依賴工具，下載並安裝。若要安裝 N 依賴擴充功能，請執行 `NDepend.VisualStudioExtension.Installer` (請參閱指示說明)。您可以根據您的需求選擇視覺工作室 2019 或 2022 年。
- .NET 的移植助手，下載並安裝。

架構

現代化購物車應用程式

下圖說明舊版 ASP.NET 購物車應用程式的現代化程序。

目標架構

下圖說明 AWS 上現代化購物車應用程式的架構。ASP.NET 核心網頁 API 部署到 Amazon ECS 叢集。記錄和組態服務由 Amazon CloudWatch 日誌和 AWS Systems Manager 提供。

工具

AWS 服務

- Amazon ECS — 亞馬遜彈性容器服務 (Amazon ECS) 是一種高度可擴展、快速的容器管理服務，用於在叢集上執行、停止和管理容器。您可以在由 AWS Fargate 管理的無伺服器基礎設施上執行任務和服務。或者，若要進一步控制基礎設施，您可以在您管理的 EC2 執行個體叢集上執行任務和服務。
- Amazon CloudWatch 日誌 — Amazon CloudWatch 日誌會集中您使用的所有系統、應用程式和 AWS 服務的日誌。您可以檢視和監視記錄、搜尋特定錯誤代碼或模式、根據特定欄位對其進行篩選，或安全地將其封存以供日 future 分析。
- AWS Systems Manager – AWS Systems Manager 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。您可以使用 Systems Manager 主控台檢視來自多個 AWS 服務的操作資料，並自動化 AWS 資源的操作任務。Systems Manager 會掃描您的代管執行個體，並針對偵測到的任何原則違規報告 (或採取修正措施)，協助您維護安全性與合規性。

工具

- 視覺工作室或視覺工作室代碼 — 用於構建. NET 應用程序，網絡 API 和其他程序的工具。
- 適用 AWS Toolkit for Visual Studio — 適用於 Visual Studio 的擴充功能，可協助開發、偵錯和部署使用 AWS 服務的 .NET 應用程式。
- Docker 桌面 — 可簡化建置和部署容器化應用程式的工具。
- nDepen d — 監視 .NET 代碼的依賴關係，質量問題和代碼更改的分析器。
- .NET 移植助理 — 一種分析工具，可掃描 .NET 代碼以識別與 .NET Core 的不兼容性並估計遷移工作量。

史诗

將舊版應用程式移植到 .NET 5 或更新版本

任務	描述	所需技能
將您的 .NET 框架舊版應用程序升級到 .NET 5。	您可以使用 .NET 的移植助理程式，將舊版 ASP.NET Web 表單應用程式轉換為 .NET 5 或更新版本。請遵循 .NET 文件的移植助理中的指示進行。	應用程式開發人員
產生 N 依賴報告。	當您將 ASP.NET Web Form 應用程式分解為微服務來現代化時，您可能不需要舊版應用程式中的所有 .cs 檔案。您可以使用 NDepend 產生任何程式碼隱藏 (.cs) 檔案的報告，以取得所有呼叫者和呼叫者。此報告可協助您識別並僅使用微服務中所需的檔案。 安裝 nDepend 之後 (請參閱 [必要條件] 區段)，請在 Visual Studio 中開啟舊版應用程式的解決方案 (.sln 檔案)，然後依照下列步驟執行： 1. 在中建置舊版應用程式。 2. 在 [視覺工作室] 功能表列上，選擇 [依賴]，[將新的 N 依賴專案附加到目前的 VS 解決方案。 3. 選擇「分析 .NET 組合」。 4. 分析完成後，瀏覽至 [方案總管] 中的專案。以滑鼠右鍵按一下您要產生報	應用程式開發人員

任務	描述	所需技能
	表的任何程式碼後置檔案 (例如，`listproducts.aspx.cs`)，然後選擇 [在相依性圖表上顯示]。 5. 在導覽列中，選擇 [來電者和來電者]，然後選擇 [編輯程式碼查詢]。 6. 在 [查詢和規則編輯] 窗格中，選擇下載箭頭，然後選擇 [匯出至 Excel]。 此程序會產生程式碼後置檔案的報告，其中列出所有呼叫者和呼叫者。如需相依性圖形的詳細資訊，請參閱 nDepend 說明文件。	

任務	描述	所需技能
建立新的 .NET 5 解決方案。	要為現代化的 ASP.NET 核心 Web API 創建一個新的 .NET 5（或更高版本）結構： 1. 開啟視覺工作室。 2. 建立新的空白解決方案。 3. 根據您的舊版應用程式，建立以 .NET 5 (或更新版本) 為目標的新專案。如需購物車應用程式的舊專案和新專案範例，請參閱「其他資訊」一節。 4. 使用上一個步驟中的 nDepend 報告來識別所有必要的檔案。從先前升級的應用程式複製這些檔案，並將其新增至新的解決方案。 5. 建置解決方案並修正所有問題。 如需有關建立專案和解決方案的詳細資訊，請參閱 Visual Studio 文件。 注意當您建置解決方案並驗證功能時，除了 nDepend 識別的檔案之外，您可能會識別要新增至解決方案的其他檔案。	應用程式開發人員

更新應用程式程式碼

任務	描述	所需技能
使用 ASP.NET 核心實作網頁 API。	假設您在舊版整體購物車應用程式中識別的其中一個微服務是「產品」。您建立了一個新的 ASP.NET 核心 Web API 專案的產品在先前的史詩。在此步驟中，您會識別與產品相關的所有 Web 表單 (.aspx 頁面) 並將其現代化。假設「產品」由四個 Web 表單組成，如前面「架構」一節所示： • 列出產品 • 查看產品 • 新增/編輯產品 • 刪除產品 您應該分析每個 Web 表單，確定發送到數據庫以執行某些邏輯的所有請求，並獲得響應。您可以將每個請求實作為 Web API 端點。鑑於其 Web 表單，產品可以具有以下可能的端點： • `/api/products` • `/api/products/{id}` • `/api/products/add` • `/api/products/update/{id}` • `/api/products/delete/{id}`	應用程式開發人員

任務	描述	所需技能
	如前所述，您還可以重複使用升級到 .NET 5 的所有其他項目，包括業務邏輯，數據訪問和共享/通用項目。	
設定 Amazon CloudWatch 日誌。	您可以使用 Amazon CloudWatch 日誌來監控、存放和存取應用程式的日誌。您可以使用 AWS 開發套件，將資料登入 Amazon CloudWatch 日誌。您也可以使用常用的 .NET 記 CloudWatch 錄架構 (例如 NLog、Log4Net 和 ASP.NET 核心記錄架構)，將 .NET 應用程式與記錄整合。 如需有關此步驟的詳細資訊，請參閱 Amazon CloudWatch 日誌和 .NET 記錄架構的部落格文章。	應用程式開發人員

任務	描述	所需技能
設定 AWS Systems Manager Parameter Store。	您可以使用 AWS Systems Manager Parameter Store 將應用程式設定 (例如連接字串) 與應用程式的程式碼分開存放。該 NuGet 軟件包亞馬遜。擴展。 SystemsManager簡化應用程式將這些設定從 AWS Systems Manager Parameter Store 載入 .NET 核心組態系統的方式。 如需有關此步驟的詳細資訊，請參閱部落格文章 AWS Systems Manager 的 .NET 核心組態提供者。	應用程式開發人員

添加身份驗證和授權

任務	描述	所需技能
使用共享 cookie 進行身份驗證。	將傳統的整體式應用程式現代化是一個反覆的程序，需要整體式及其現代化版本共存。您可以使用共享 cookie 來實現兩個版本之間的無縫身份驗證。舊版 ASP.NET 應用程式會繼續驗證使用者認證，並在現代化的 ASP.NET 核心應用程式驗證 Cookie 時發出 Cookie。 如需指示和範例程式碼，請參閱範例 GitHub 專案。	應用程式開發人員

在本機建置並執行容器

任務	描述	所需技能
通過使用視覺工作室創建碼頭映像。	在此步驟中，您可以使用適用於 .NET 核心網頁 API 的視覺工作室建立碼頭檔案。 1. 開啟視覺工作室。 2. 在解決方案資源管理器中，從項目的上下文（右鍵單擊）菜單中，選擇添加，Docker Support。 3. 選取 Linux 作為目標作業系統。 視覺工作室創建一個碼頭文件為您的項目。如需碼頭檔案範例，請參閱 Microsoft 網站上的泊塢視覺工作室容器工具。	應用程式開發人員
使用 Docker 桌面構建和運行容器。	現在，您可以在 Docker 桌面中構建，創建和運行容器。 1. 開啟命令提示視窗。導航到 Docker 文件所在的解決方案文件夾。執行下列命令以建立 Docker 映像檔： `docker build -t aspnetcorewebapiimage -f Dockerfile .` 2. 執行下列命令以檢視所有 Docker 映像檔： `docker images`	應用程式開發人員

任務	描述	所需技能
	3. 執行下列命令以建立並執行容器： `docker run -d -p 8080:80 --name aspnetcorewebapicontainer aspnetcorewebapiimage` 4. 開啟 Docker 桌面，然後選擇 [容器/應用程式]。您可以看到一個名為`aspnetcorewebapicontainer` 正在運行的新容器。	

相關資源

- 在 Amazon EC2 Linux 執行個體上執行 ASP.NET 核心網路 API 泊塢視窗容器 (AWS Prescriptive Guidance)
- Amazon ECS 工作坊
- 使用 AWS 執行 ECS 藍/綠部署 CloudFormation (CodeDeploy AWS CloudFormation 文件)
- 開始使用 nDepend (nDepen d 文件)
- .NET 的移植助理

其他資訊

下表提供舊版購物車應用程式的範例專案範例，以及現代化 ASP.NET Core 應用程式中的對等專案。

傳統解決方案：

專案名稱	專案範本	目標框架
業務介面	類別圖書館	.NET Framework
BusinessLogic	類別圖書館	.NET Framework

WebApplication	框架網頁應用程式	.NET Framework
UnitTests	nUnit 測試項目	.NET Framework
共用-> 通用	類別圖書館	.NET Framework
共享-> 框架	類別圖書館	.NET Framework

新解決方案：

專案名稱	專案範本	目標框架
BusinessLogic	類別圖書館	.NET 5.0
<WebAPI>	核心網頁 API	.NET 5.0
<WebAPI>。 UnitTests	nUnit 3 測試專案	.NET 5.0
共用-> 通用	類別圖書館	.NET 5.0
共享-> 框架	類別圖書館	.NET 5.0

使用 AWS Fargate 大規模執行事件驅動和排程的工作負載

創建者：哈里歐姆普拉薩特拉賈戈帕爾（AWS）

環境：PoC 或試點	技術：現代化、無伺服器、營運	工作負載：開源
AWS 服務：Amazon EC2 容器註冊表；Amazon ECS；AWS CodeCommit；AWS Fargate；AWS Lambda；Amazon SNS		

Summary

此模式描述如何使用 AWS Fargate 在 Amazon Web Services (AWS) 雲端上大規模執行排程和事件導向的工作負載。

在此模式設定的使用案例中，每當提交提取請求時，都會掃描程式碼中是否有 AWS 敏感資訊，例如 AWS 帳戶號碼和登入資料。提取請求會啟動 Lambda 函數。Lambda 函數會叫用處理程式碼掃描的 Fargate 工作。每當引發新的提取請求時，就會啟動 Lambda。如果掃描發現任何敏感資訊，Amazon Simple Notification Service (Amazon SNS) 會以電子郵件傳送掃描結果。

此模式在下列業務使用案例中很有幫助：

- 如果您的企業必須執行許多因為執行時間限制 (15 分鐘限制) 或記憶體而無法由 AWS Lambda 執行的排程和事件導向工作負載
- 如果您希望 AWS 管理為這些工作負載佈建的執行個體

使用此模式時，您可以選擇創建新的虛擬私有雲（VPC）。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- AWS CodeCommit 用於託管代碼庫和創建提取請求
- AWS Command Line Interface (AWS CLI) (AWS CLI) 1.7 版或更新版本，已在 macOS、Linux 或視窗上安裝和設定
- 在容器中執行的工作
- 在類路徑中設置的阿帕奇 Maven 可執行文件

架構

整體流程包括以下步驟。

1. 每當在中提交新的提取請求時 CodeCommit，都會啟動 Lambda 函數。Lambda 函數通過`CodeCommit Pull Request State Change`事件通過 Amazon EventBridge 監聽。
2. Lambda 函數會提交具有下列環境參數的新 Fargate 任務，以檢出程式碼並進行掃描。

   ```
   RUNNER # <<TaskARN>>
   SNS_TOPIC # <<SNSTopicARN>>
   SUBNET # <<Subnet in which Fargate task gets launched>>
   ```

   如果掃描在程式碼中找到敏感資訊，Fargate 會將新訊息推送至 Amazon SNS 主題。
3. SNS 訂閱者會從主題讀取訊息並傳送電子郵件訊息。

技術

- AWS CodeCommit
- Amazon Elastic Container Registry (Amazon ECR)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon EventBridge
- AWS Fargate
- AWS Lambda
- Amazon SNS
- Docker

工具

工具

- AWS CLI — AWS 命令列界面 (CLI) 是管理 AWS 服務的統一工具。
- AWS CodeCommit — AWS CodeCommit 是全受管的原始檔控制服務，可託管安全的 Git 儲存庫。使用後 CodeCommit，團隊可以在安全且可高度擴展的環境中協作程式碼。
- Amazon ECR — 亞馬遜彈性容器登錄 (Amazon ECR) 是一種全受管的登錄，開發人員可用來存放、管理和部署 Docker 容器映像。
- Amazon ECS — Amazon Elastic Container Service (Amazon ECS) 是一種高度可擴展的快速容器管理服務。您可以使用 Amazon ECS 在叢集上執行、停止和管理容器。
- AWS Fargate — AWS Fargate 是一項技術，您可以與 Amazon ECS 搭配使用來執行容器，而不必管理伺服器或 Amazon EC2 執行個體的叢集。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。發佈者透過製作並傳送訊息到主題 (其為邏輯存取點和通訊管道) 與訂閱者進行非同步的通訊。訂閱 SNS 主題的用戶端會使用支援的通訊協定接收已發佈的訊息，例如 Lambda、電子郵件、行動推播通知和行動文字訊息 (SMS)。
- Docker — Doc ker 可協助您在稱為容器的套件中建置、測試和交付應用程式。
- Git 客戶端-命令行或桌面工具來檢查所需的工件
- Maven 的-阿帕奇 Maven 的是集中管理項目的構建，報告和文檔的項目管理工具。

史诗

設定本機儲存庫

任務	描述	所需技能
下載代碼。	在「附件」區段中，下載 .zip 檔案並解壓縮檔案。	AWS 系統管理員開發人員
設置回購。	`mvn clean install`在根文件夾上運行。	AWS 系統管理員開發人員

創建 Amazon ECR 映像並推送映像

任務	描述	所需技能
建立 Amazon ECR 儲存庫並登入。	打開 Amazon ECR 控制台。在瀏覽窗格中，選擇 [儲存庫]，然後選擇 [建立存放庫]。如需此和其他故事的說明，請參閱「相關資源」一節。	AWS 系統管理員開發人員
推送您的容器映像。	打開存儲庫，選擇查看推送命令，然後登錄到 Docker。登入之後，在 [其他資訊] 區段的 [推送容器映像] 下方執行具有必要替代的命令。這會上傳用於執行程式碼掃描的 Docker 容器映像檔。上傳完成後，請在 Amazon ECR 儲存庫中複製最新組建的 URL。	AWS 系統管理員開發人員

建立儲 CodeCommit 存庫

任務	描述	所需技能
建立存 CodeCommit 放庫。	若要建立新的 AWS CodeCommit 儲存庫，請在 [其他資訊] 區段中的 [建立 CodeCommit 儲存庫] 下執行命令。	AWS 系統管理員開發人員

建立 VPC (選用)

任務	描述	所需技能
建立 VPC。	如果您要使用新的 VPC 而不是現有的 VPC，請在 [其他資訊]	AWS 系統管理員開發人員

任務	描述	所需技能
	區段中的 [建立 VPC] 下執行命令。AWS Cloud Development Kit (AWS CDK) 指令碼將輸出所建立的 VPC 和子網路的 ID。	

建立 Amazon ECS 叢集和 Fargate 任務

任務	描述	所需技能
建立叢集和工作。	若要建立 Amazon ECS 叢集和 Fargate 任務定義，請在其他資訊區段中的建立叢集和任務下執行命令。請確定在執行命令介面指令碼時，已將正確的 VPC ID 和 Amazon ECR 存放庫 URI 當做參數傳入。該腳本創建一個指向 Docker 映像（負責掃描）的 Fargate 任務定義。然後，指令碼會建立工作和關聯的執行角色。	AWS 系統管理員開發人員
驗證 Amazon ECS 叢集。	開啟 Amazon ECS 主控台。在導覽窗格中，選擇 [叢集]，然後選擇名為 Fargate 端工作叢集的新建立的 Amazon ECS 叢集。在此之後，在導覽窗格中選擇 [工作定義]，並確認有新的任務定義與前置詞`awscdkfargateecsTaskDef` 。	AWS 系統管理員開發人員

建立 SNS 主題和訂閱者

任務	描述	所需技能
建立 SNS 主題。	若要建立 SNS 主題，請在「其他資訊」區段中的「建立 SNS」主題下執行命令。建立成功後，請注意SNS ARN，這是在下一個步驟中使用。	AWS 系統管理員開發人員
建立 SNS 訂閱者。	若要為 SNS 主題建立電子郵件訂閱者，請在「其他資訊」區段中的「建立 SNS 訂閱者」下執行命令。確保在 CLI 命令中替換`TopicARN`並`Email address`使用。若要接收電子郵件通知，請務必確認作為訂閱者使用的電子郵件地址。	AWS 系統管理員開發人員

建立 Lambda 函數和 CodeCommit 觸發器

任務	描述	所需技能
建立函數和觸發器。	若要使用 CodeCommit 觸發器建立 Lambda 函數，請在 Lambda 函數下執行命令，然後在其他資訊部分中進行 CodeCommit 觸發。執行指令之前，請務必使用對應的值取代參數。指令碼會建立 Lambda 函數，並將其設定為在發出新的提取要求時叫用。	AWS 系統管理員開發人員

測試應用程式。

任務	描述	所需技能
測試應用程式。	如果您將任何 AWS 敏感資訊簽入 CodeCommit 存放庫，則應啟動 Lambda 函數。Lambda 函數會啟動 Fargate 工作，該工作會掃描程式碼，並以電子郵件通知傳送掃描結果。	AWS 系統管理員開發人員

相關資源

- 創建一個 Amazon ECR 存儲庫
- 將碼頭圖像推送到 Amazon ECR

其他資訊

推送容器映像

```
> cd 1-ecr-image-push
> ./run.sh <<ecr-repository>>
```

建立儲 CodeCommit 存庫

```
aws codecommit create-repository --repository-name test-repo --repository-description
 "My Test repository"
```

建立 VPC

```
> cd 2-create-vpc
> ./run.sh
```

輸出

```
aws-batch-cdk-vpc-efs-launch-template.privatesubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.publicsubnet = subnet-<<id>>
aws-batch-cdk-vpc-efs-launch-template.vpcid = vpc-<<id>>
```

建立叢集和工作

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 3-create-ecs-task
> ./run.sh <<vpc-id>> <<ecr-repo-uri>>
```

輸出

```
aws-cdk-fargate-ecs.CLUSTERNAME = Fargate-Job-Cluster
aws-cdk-fargate-ecs.ClusterARN = <<cluster_arn>>
aws-cdk-fargate-ecs.ContainerARN = Fargate-Container
aws-cdk-fargate-ecs.TaskARN = <<task_arn>>
aws-cdk-fargate-ecs.TaskExecutionRole = <<execution_role_arn>>
aws-cdk-fargate-ecs.TaskRole = <<task_role_arn>>
```

建立 SNS 主題

```
aws sns create-topic --name code-commit-topic
```

建立 SNS 訂閱者

```
aws sns subscribe \
    --topic-arn <<topic_arn>> \
    --protocol email \
    --notification-endpoint <<email_address>>
```

Lambda 函數和CodeCommit 觸發器

```
> export CDK_DEFAULT_ACCOUNT = <<aws_account_id>>
> export CDK_DEFAULT_REGION = <<aws_region>>
> cd 5-Lambda-CodeCommit-Trigger
> ./run.sh <<taskarn>> <<snstopicarn>> subnet-<<id>> <<codecommitarn>>
```

輸出

```
aws-cdk-fargate-lambda-event.Cloudwatchrule = <<cloudwatchrule>>
aws-cdk-fargate-lambda-event.CodeCommitLambda = AWS-Code-Scanner-Function
aws-cdk-fargate-lambda-event.LambdaRole = <<lambdaiamrole>>
```

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在 SaaS 架構中使用 C# 和 AWS CDK 進行筒倉模型的租用戶上線

由虎斑病房（AWS），蘇米莎·雷迪甘基迪（AWS）和維賈伊阿南德拉馬林甘（AWS）創建

代碼庫：Tennat 入職筒倉	環境：PoC 或試點	技術：現代化；雲端原生；SaaS； DevOps
工作負載：開源	AWS 服務：AWS CloudFormation；Amazon DynamoDB；Amazon DynamoDB 串流；AWS Lambda；Amazon API Gateway	

Summary

軟體即服務 (SaaS) 應用程式可以使用各種不同的架構模型來建置。筒倉模型是指租戶提供專用資源的架構。

SaaS 應用程式仰賴無摩擦模式，將新租用戶引入其環境。這通常需要協調一些元件，才能成功佈建和設定建立新承租人所需的所有元素。在 SaaS 架構中，此程序稱為租用戶上線。通過在入職過程中使用基礎架構即代碼，應該針對每個 SaaS 環境進行全自動化的入職。

此模式會引導您完成在 Amazon Web Services (AWS) 上為租用戶建立租用戶和佈建基本基礎設施的範例。該模式使用 C# 和 AWS Cloud Development Kit（AWS CDK）。

由於此模式會產生帳單警示，因此建議您在美國東部 (維吉尼亞北部) 或 us-east-1 AWS 區域部署堆疊。如需詳細資訊，請參閱 AWS 文件。

先決條件和限制

前提

- 有效的 AWS 帳戶。
- 具有足夠 IAM 存取權限的 AWS 身分與存取管理 (IAM) 主體，可針對此模式建立 AWS 資源。如需詳細資訊，請參閱 IAM 角色。
- 安裝 Amazon Command Line Interface (AWS CLI) (AWS CLI) 並將 AWS CLI 設定為執行 AWS CDK 部署。

- 視覺工作室 2022 下載並安裝或視覺工作室代碼下載並安裝.
- 適用於視覺工作室設定的 AWS 工具組。
- .NET 核心 3.1 或更新版本 (C# AWS CDK 應用程式需要)
- 安裝工具。

限制

- AWS CDK 使用 AWS CloudFormation，因此 AWS CDK 應用程式會受到 CloudFormation 服務配額限制。如需詳細資訊，請參閱 AWS CloudFormation 配額。
- 使用 CloudFormation 服務角色`infra-cloudformation-role`建立的承租人 CloudFormation 堆疊會在動作 (`sns*` 和`sqs*`) 上包含萬用字元，但資源已鎖定至`tenant-cluster`前置詞。對於生產使用案例，請評估此設定，並僅提供此服務角色的必要存取權。`InfrastructureProvisionLambda` 函數也會使用萬用字元 (`cloudformation*`) 來佈建 CloudFormation 堆疊，但將資源鎖定至`tenant-cluster`前置字元。
- 這個示例代碼的碼 docker 構建使`--platform=linux/amd64`用強制`linux/amd64`基於圖像。這是為了確保最終的圖像偽影將適用於 Lambda，默認情況下使用 x86-64 架構。如果您需要變更目標 Lambda 架構，請務必同時變更碼頭檔案和 AWS CDK 代碼。如需詳細資訊，請參閱部落格文章將 AWS Lambda 函數遷移至以 ARM 為基礎的 AWS 重力 on2 處理器。
- 堆疊刪除程序不會清除堆疊所產生的 CloudWatch 記錄檔 (記錄群組和記錄檔)。您必須透過 AWS 管理主控台 Amazon 主控 CloudWatch 台或透過 API 手動清理日誌。

此模式設定為範例。對於生產用途，請評估下列設定並根據您的業務需求進行變更：

- 此範例中的 AWS 簡易儲存服務 (Amazon S3) 儲存貯體為了簡化而未啟用版本控制。視需要評估並更新設定。
- 此範例設定 Amazon API Gateway REST API 端點，而不需要驗證、授權或節流，以簡化操作。對於生產使用，我們建議將系統與業務安全基礎架構整合。評估此設定，並視需要新增必要的安全性設定。
- 對於此租用戶基礎設施範例，亞馬遜簡單通知服務 (Amazon SNS) 和亞馬遜簡單佇列服務 (Amazon SQS) 只有最低設置。每個租用戶的 AWS Key Management Service (AWS KMS) 會開啟帳戶中的 Amazon CloudWatch 和 Amazon SNS 服務，以根據 AWS KMS 金鑰政策使用。該設置只是一個示例佔位符。根據您的業務使用案例，視需要調整設定。
- 整個設定包括但不限於 API 端點和後端租用戶使用 AWS 佈建和刪除 CloudFormation，僅涵蓋基本的快樂路徑案例。根據您的業務需求，使用必要的重試邏輯、其他錯誤處理邏輯和安全性邏輯來評估和更新設定。

- 示例代碼通過 up-to-date cdk-nag 進行測試，以便在撰寫本文時檢查策略。future 可能會強制執行新原則。這些新原則可能需要您根據建議手動修改堆疊，然後才能部署堆疊。檢閱現有程式碼，確保程式碼符合您的業務需求。
- 程式碼依賴 AWS CDK 產生隨機尾碼，而不是依賴大多數建立資源的靜態指派實體名稱。此設置是為了確保這些資源是唯一的，並且不會與其他堆棧衝突。如需詳細資訊，請參閱 AWS CDK 文件。根據您的業務需求進行調整。
- 此範例程式碼會將 .NET Lambda 成品封裝到以碼頭為基礎的映像中，並使用 Lambda 提供的容器映像執行階 容器映像執行階段對於標準傳輸和儲存機制 (容器登錄) 和更準確的本機測試環境 (透過容器映像檔) 具有優勢。您可以將專案切換為使用 Lambda 提供的 .NET 執行階段來縮短 Docker 映像檔的建置時間，但是您必須設定傳輸和存放機制，並確保本機設定符合 Lambda 安裝程式。調整程式碼以符合使用者的業務需求。

產品版本

- AWS CDK 版本 2.45.0 或更新版本
- 視覺工作室

架構

技術, 堆

- Amazon API Gateway
- AWS CloudFormation
- Amazon CloudWatch
- Amazon DynamoDB
- AWS Identity and Access Management (IAM)
- AWS KMS
- AWS Lambda
- Amazon S3
- Amazon SNS
- Amazon SQS

架構

下圖顯示租用戶堆疊建立流程。如需控制層和租用戶技術堆疊的詳細資訊，請參閱其他資訊一節。

租用戶堆疊建立流程

1. 使用者以 JSON 格式將包含新租用戶承載 (租用戶名稱、租用戶描述) 的 POST API 請求傳送至 Amazon API 閘道託管的 REST API。API Gateway 會處理請求，並將其轉送至後端 Lambda 租用戶上線功能。在此範例中，沒有授權或驗證。在生產設置中，此 API 應與 SaaS 基礎架構安全系統集成。
2. 租用戶上線功能會驗證要求。然後，它會嘗試將租用戶記錄 (包括租用戶名稱、產生的租用戶通用唯一識別碼 (UUID) 和租用戶說明儲存到 Amazon DynamoDB 租用戶上線表格中。
3. DynamoDB 儲存記錄之後，DynamoDB 串流會啟動下游 Lambda 租用戶基礎設施函數。
4. 租用戶基礎設施 Lambda 函數會根據接收到的 DynamoDB 串流進行動作。如果串流是針對 INSERT 事件，則函數會使用串流的 NewImage 區段 (最新更新記錄、租用戶名稱欄位) CloudFormation 來叫用，以使用存放在 S3 儲存貯體中的範本建立新的租用戶基礎結構。 CloudFormation 範本需要「承租人名稱」參數。
5. AWS CloudFormation 會根據 CloudFormation 範本和輸入參數建立租用戶基礎設施。
6. 每個租用戶基礎結構設定都有警示、帳單警示和警示事件。 CloudWatch
7. 警示事件會變成 SNS 主題的訊息，該 SNS 主題已由租用戶的 AWS KMS 金鑰加密。
8. SNS 主題會將接收到的警示訊息轉寄到 SQS 佇列，該佇列會由租用戶的 AWS KMS 加密以取得加密金鑰。

其他系統可與 Amazon SQS 整合，以根據佇列中的訊息執行動作。在此範例中，為了讓程式碼保持一般性，內送訊息會保留在佇列中，並且需要手動刪除。

租用戶堆疊刪除流程

1. 使用者以 JSON 格式將含有新租用戶承載 (租用戶名稱、租用戶描述) 的 DELETE API 請求傳送至 Amazon API Gateway 託管的 REST API，該 API 將處理請求並轉送至租用戶上線功能。在此範例中，沒有授權或驗證。在生產設置中，此 API 將與 SaaS 基礎架構安全系統集成。
2. 承租人上線功能會驗證要求，然後嘗試從 [租用戶上線] 資料表中刪除承租人記錄 (租用戶名稱)。
3. DynamoDB 成功刪除記錄 (記錄存在於資料表中且已刪除) 之後，DynamoDB 串流會啟動下游 Lambda 租用戶基礎設施函數。

4. 租用戶基礎設施 Lambda 函數會根據收到的 DynamoDB 串流記錄執行作用。如果串流是針對 REMOVE 事件，則函數會使用記錄的 OldImage 區段 (記錄資訊和承租人名稱欄位，在最新變更之前，也就是刪除)，根據該記錄資訊起始刪除現有堆疊。
5. AWS CloudFormation 會根據輸入刪除目標租用戶堆疊。

工具

AWS 服務

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CDK 工具組是命令列 Cloud Development Kit，可協助您與 AWS 雲端開發套件 (AWS CDK) 應用程式互動。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。
- AWS Toolkit for Visual Studio 是適用於視覺工作室整合式開發環境 (IDE) 的外掛程式。適用 Toolkit for Visual Studio 支援開發、偵錯和部署使用 AWS 服務的 .NET 應用程式。

其他工具

- Visual Studio 是一個 IDE，其中包括編譯器，代碼完成工具，圖形設計師，以及支持軟件開發的其他功能。

Code

此模式的代碼位於 SaaS 架構的筒倉模型 APG 示例存儲庫中的租戶入職中。

史诗

設定 AWS CDK

任務	描述	所需技能
請確認 Node.js 安裝程式。	若要確認 Node.js 已安裝在您的本機電腦上，請執行下列命令。 `node --version`	AWS 管理員 DevOps
安裝 AWS CDK 工具組。	若要在本機電腦上安裝 AWS CDK 工具組，請執行下列命令。 `npm install -g aws-cdk` 如果未安裝 npm，您可以從 Node.js 網站安裝它。	AWS 管理員 DevOps
驗證 AWS CDK 工具組版本。	若要確認 AWS CDK 工具組版本已正確安裝在您的機器上，請執行下列命令。 `cdk --version`	AWS 管理員 DevOps

檢閱租用戶上線控制平面的程式碼

任務	描述	所需技能
複製儲存庫。	複製存放庫，然後導覽至 `資\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example` 料夾。 在 2022 年視覺工作室中，開啟 `\src\TenantOnboardingInfra.sln` 解決方案。開啟檔 `TenantOnboardingInfraStack.cs` 案並檢閱程式碼。 下列資源會建立為此堆疊的一部分： • DynamoDB 表 • S3 儲存貯體（將 CloudFormation 範本上傳到 S3 儲存貯體。） • Lambda 執行角色 • Lambda 函數 • API Gateway • 至 Lambda 函數的事件來源	AWS 管理員 DevOps
檢閱 CloudFormation 範本。	在 `資\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\template` 料夾中 `infra.yaml` ，開啟並檢閱 CloudFormation 範本。此範本會使用從承租人上	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	架 DynamoDB 表格擷取的租用戶名稱進行補充。 範本會佈建承租人特定的基礎結構。在此範例中，它佈建了 AWS KMS 金鑰、Amazon SNS、Amazon SQS 和 CloudWatch 警示。	
檢閱租用戶上線功能。	開啟`Function.cs` 並檢閱租用戶入職功能的程式碼，此功能是使用具有 .NET 6 (容器映像) 藍圖的 Visual Studio AWS Lambda 專案 (.NET 核心 C#) 範本建立的。 開啟`Dockerfile` ，然後檢閱程式碼。這`Dockerfile` 是一個文字檔，其中包含建置 Lambda 容器映像檔的指示。 請注意，下列 NuGet 套件會新增為`TenantOnboardingFunction` 專案的相依性： • `Amazon.Lambda.APIGatewayEvents` • `AWSSDK.DynamoDBv2` • `Newtonsoft.Json`	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
檢閱承租人 InfraProvisioning 功能。	導覽至 `\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\InfraProvisioningFunction` 。 開啟`Function.cs` 並檢閱租用戶基礎設施佈建函數的程式碼，該函數是使用具有 .NET 6 (容器映像) 藍圖的 Visual Studio AWS Lambda 專案 (.NET 核心 C#) 範本建立的。 開啟`Dockerfile` ，然後檢閱程式碼。 請注意，下列 NuGet 套件會新增為`InfraProvisioningFunction` 專案的相依性： • `Amazon.Lambda.DynamoDBEvents` • `AWSSDK.DynamoDBv2` • `AWSSDK.Cloudformation`	AWS 應用程式開發人員 DevOps

部署 AWS 資源

任務	描述	所需技能
建置解決方案。	若要建置解決方案，請執行下列步驟：	應用程式開發人員

任務	描述	所需技能
	1. 在 2022 年視覺工作室中，開啟\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra.sln 解決方案。 2. 開啟解決方案的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [建置方案]。 注意：在建置解決方案之前，請確定您已將Amazon.CDK.Lib NuGet 套件更新為\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example\src\TenantOnboardingInfra 專案中的最新版本。	

任務	描述	所需技能
啟動 AWS CDK 環境。	開啟 Windows 命令提示字元，然後瀏覽至可用`cdk.json`檔案的 AWS CDK 應用程式根資料夾 (`\tenant-onboarding-in-saas-architecture-for-silo-model-apg-example`)。運行以下命令進行引導。 `cdk bootstrap` 如果您已為登入資料建立 AWS 設定檔，請搭配您的設定檔使用命令。 `cdk bootstrap --profile <profile name>`	AWS 管理員 DevOps
列出 AWS CDK 堆疊。	若要列出要建立為此專案一部分的所有堆疊，請執行下列命令。 `cdk ls` `cdk ls --profile <profile name>` 如果您已為登入資料建立 AWS 設定檔，請搭配您的設定檔使用命令。 `cdk ls --profile <profile name>`	AWS 管理員 DevOps

任務	描述	所需技能
檢閱將建立哪些 AWS 資源。	若要檢閱將在此專案中建立的所有 AWS 資源，請執行下列命令。 `cdk diff` 如果您已為登入資料建立 AWS 設定檔，請搭配您的設定檔使用命令。 `cdk diff --profile <profile name>`	AWS 管理員 DevOps

任務	描述	所需技能
使用 AWS CDK 部署所有 AWS 資源。	若要部署所有 AWS 資源，請執行下列命令。 `cdk deploy --all --require-approval never` 如果您已為登入資料建立 AWS 設定檔，請搭配您的設定檔使用命令。 `cdk deploy --all --require-approval never --profile <profile name>` 部署完成後，從命令提示字元中的輸出區段複製 API URL，如下列範例所示。 `Outputs: TenantOnboardingInfraStack.TenantOnboardingAPIEndpoint42E526D7 = https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/`	AWS 管理員 DevOps

驗證功能

任務	描述	所需技能
建立新承租人。	若要建立新承租人，請傳送下列 curl 要求。	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
	`curl -X POST <TenantOnboardingAPIEndpoint* from CDK Output>tenant -d '{"Name":"Tenant123", "Description":"Stack for Tenant123"}'` 將預留位置變更`<TenantOnboardingAPIEndpoint* from CDK Output>` 為 AWS CDK 的實際值，如下列範例所示。 `curl -X POST https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant -d '{"Name": "Tenant123", "Description":"test12"}'` 下面的例子顯示了輸出。 `{"message": "A new tenant added - 5/4/2022 7:11:30 AM"}`	

任務	描述	所需技能
在 DynamoDB 中驗證新建立的承租人詳細資料。	若要驗證 DynamoDB 中新建立的承租人詳細資料，請執行下列步驟。 1. 開啟 AWS 管理主控台，然後瀏覽至 Amazon DynamoDB 服務。 2. 在左側導覽中，選擇「瀏覽項目」，然後選擇`TenantOnboarding` 表格。 注意：承租人名稱前面會加上。`tenantcluster-` 如需詳細資訊，請參閱其他資訊一節。 3. 確認已使用承租人詳細資料建立新項目。	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
確認新承租人的堆疊建立。	確認新堆疊已成功建立，並根據 CloudFormation 範本為新建立的租用戶使用基礎結構佈建。 1. 開啟主 CloudFormation 控台。 2. 在左側導覽中，選擇 [堆疊]，然後確認已成功建立具有租用戶名稱的堆疊。 3. 選擇新建立的承租人堆疊，然後選擇 [資源] 索引標籤。請注意警示資源和 Amazon SQS 資源。 4. 開啟已設定 AWS 登入資料的新終端機，並指向正確的區域。若要發出測試警示，請輸入下列程式碼，並<alarm resource name>以步驟 3 中註明的警示資源名稱取代。 `aws cloudwatch set-alarm-state --alarm-name <alarm resource name> --state-value ALARM --state-reason 'Test setup'` 下列範例顯示具有警示資源名稱的程式碼。 `aws cloudwatch set-alarm-state --alarm-name tenantcluster-tenant123-alarm --`	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
	`state-value ALARM -- state-reason 'Test setup'` 5. 開啟主控台並導覽至 Amazon SQS 主控台。選擇步驟 3 中識別的 Amazon SQS 資源名稱。遵循 AWS 文件指示，從步驟 4 中引發的警示接收和刪除測試訊息。	

任務	描述	所需技能
刪除承租人堆疊。	若要刪除承租人堆疊，請傳送下列 curl 要求。 `curl -X DELETE <TenantOnboardingAPIEndpoint* from CDK Output>tenant/<Tenant Name from previous step>` 將預留位置變更`<TenantOnboardingAPIEndpoint* from CDK Output>` 為 AWS CDK 的實際值，然後從上一個租用戶建立步驟變更`<Tenant Name from previous step>`為實際值，如下列範例所示。 `curl -X DELETE https://j2qmp8ds21i1i.execute-api.us-west-2.amazonaws.com/prod/tenant/Tenant123` 下面的例子顯示了輸出。 `{"message": "Tenant destroyed - 5/4/2022 7:14:48 AM"}`	應用程式開發人員、AWS DevOps、AWS 管理員

任務	描述	所需技能
確認現有承租人的堆疊刪除。	若要確認已刪除現有承租人堆疊，請執行下列步驟： 1. 打開控制台並導航到 CloudFormation 控制台。 2. 在左側導覽中，確認具有租用戶名稱的現有堆疊已不再位於 CloudFormation 主控台中 (如果主控台設定為僅顯示 Active Stack) 或正在刪除。如果堆疊已不在 CloudFormation 主控台中，請使用下拉式清單將主控台的設定從 [Active] 變更為 [已刪除]，以查看已刪除的堆疊，並確認堆疊已成功刪除。	應用程式開發人員、AWS 管理員、AWS DevOps

清除

任務	描述	所需技能
破壞環境	在堆疊清理之前，請確定下列事項： • DynamoDB 中的所有記錄都會透過先前的租用戶刪除作業或透過 DynamoDB 主控台或 API 移除。每個租用戶記錄刪除都會啟動 AWS CloudFormation 對應項目的清理。 • AWS 主控台上會清除所有租戶型 AWS CloudForm	AWS 管理員 DevOps

任務	描述	所需技能
	ation 堆疊 (萬一 DynamoDB 觸發器清理邏輯失敗)。CloudFormation 測試完成後，可以透過執行下列命令，使用 AWS CDK 銷毀所有堆疊和相關資源。 `cdk destroy --all;` 如果您為登入資料建立 AWS 設定檔，請使用該設定檔。 確認堆疊刪除提示以刪除堆疊。	
清理 Amazon CloudWatch 日誌。	堆疊刪除程序不會清除堆 CloudWatch 疊所產生的記錄檔 (記錄群組和記錄檔)。使用 CloudWatch 主控台或 API 手動清理 CloudWatch 資源。	應用程式開發人員、AWS DevOps、AWS 管理員

相關資源

- AWS CDK 網路研討會
- 在 C# 中使用 AWS CDK
- CDK 網路參考

其他資訊

控制面技術堆疊

在 .NET 編寫的 CDK 代碼用於佈建控制平面基礎設施，它由以下資源組成：

1. API 閘道

做為控制平面堆疊的 REST API 進入點。

2. 租用戶上線 Lambda 能

   此 Lambda 函數是由 API Gateway 使用 m 方法啟動的。

   POST 方法 API 要求會導致 (`tenant name`,`tenant description`) 插入 DynamoDB `Tenant Onboarding` 資料表中。

   在此程式碼範例中，租用戶名稱也會用作租用戶堆疊名稱的一部分，以及該堆疊中的資源名稱。這是為了使這些資源更容易識別。此承租人名稱在整個設定中必須是唯一的，以避免衝突或發生錯誤。如需詳細的輸入驗證設定，請參閱 IAM 角色文件和「限制」一節。

   只有在表格中的任何其他記錄中未使用承租人名稱時，DynamoDB 表的持續性程序才會成功。

   在這種情況下，承租人名稱是此資料表的資料分割索引鍵，因為只有資料分割索引鍵可以當做`PutItem`條件運算式使用。

   如果之前從未記錄過租用戶名稱，則記錄將成功儲存到資料表中。

   但是，如果表格中的現有記錄已使用承租人名稱，則作業將會失敗並起始 DynamoDB `ConditionalCheckFailedException` 例外狀況。例外狀況將用於傳回失敗訊息 (`HTTP BadRequest`)，指出承租人名稱已存在。

   方`DELETE`法 API 要求會從 `Tenant Onboardin g` 資料表中移除特定租用戶名稱的記錄。

   即使記錄不存在，此範例中的 DynamoDB 記錄刪除仍會成功。

   如果目標記錄存在且遭到刪除，則會建立 DynamoDB 串流記錄。否則，將不會建立下游記錄。

3. 租用戶上架 DynamoDB，且已啟用 Amazon DynamoDB 串流

   這會記錄租用戶中繼資料資訊，而任何記錄儲存或刪除都會將串流傳送至 `Tenant Infrastructure` Lambda 函數。

4. 租戶基礎架構 Lambda

   此 Lambda 函數是由上一個步驟的 DynamoDB 串流記錄所啟動。如果記錄是針對`INSERT`事件，則會叫用 AWS CloudFormation 以使用存放在 S3 儲存貯體中的 CloudFormation 範本建立新的租用戶基礎設施。如果記錄是`REMOVE`，它會根據流記錄的`Tenant Name`字段啟動現有堆棧的刪除。

5. S3 bucket (S3 儲存貯體)

這是用於存儲 CloudFormation 模板。

6. 每個 Lambda 函數的 IAM 角色，以及下列項目的服務角色 CloudFormation

   每個 Lambda 函數都有其唯一的 IAM 角色，具有最低權限許可以完成其任務。例如，`Tenant On-boarding`Lambda 函數具有 DynamoDB 的讀取/寫入存取權，而 `Tenant Infrastructure` Lambda 函數只能讀取 DynamoDB 串流。

   已針對租用戶堆疊佈建建立自訂 CloudFormation 服務角色。此服務角色包含 CloudFormation 堆疊佈建的其他許可 (例如 AWS KMS 金鑰)。這會在 Lambda 之間劃分角色， CloudFormation 以避免單一角色 (基礎架構 Lambda 角色) 上的所有權限。

   允許強大動作 (例如建立和刪除 CloudFormation 堆疊) 的權限會遭到鎖定，且僅允許開頭為的資源`tenantcluster-`。AWS KMS 是例外狀況，因為它的資源命名慣例。從 API 擷取的租用戶名稱會與其他驗證檢查一`tenantcluster-`起加在前面 (字母數字僅包含破折號，且不得超過 30 個字元以符合大多數 AWS 資源命名)。這可確保租用戶名稱不會意外導致核心基礎結構堆疊或資源中斷。

租用戶技術堆疊

CloudFormation 範本存放在 S3 儲存貯體中。範本佈建承租人專屬的 AWS KMS 金鑰、 CloudWatch 警示、SNS 主題、SQS 佇列和 SQS 政策。

AWS KMS 金鑰用於 Amazon SNS 和 Amazon SQS 對其訊息進行資料加密。AwsSolutions-SNS2 和 AwsSolutions-SQS2 的安全實務建議您使用加密設定 Amazon SNS 和 Amazon SQS。但是，使用 AWS 受管金鑰時， CloudWatch 警示不適用於 Amazon SNS，因此在此情況下，您必須使用客戶受管金鑰。如需詳細資訊，請參閱 AWS 知識中心。

在 Amazon SQS 佇列上使用 SQS 政策，以允許建立的 SNS 主題將訊息傳遞到佇列。如果沒有 SQS 策略，則訪問將被拒絕。如需詳細資訊，請參閱 Amazon SNS 文件。

通過使用 CQRS 和事件採購將巨石分解為微服務

由小魯道夫創作 塞拉達 (AWS)、梅德古林 (AWS) 和虎斑病房 (AWS)

環境：PoC 或試點	來源：整體式 CRUD 模型	目標：微服務
R 型：重新建築	工作負載：開源	技術：現代化；訊息與通訊；無伺服器
AWS 服務：Amazon DynamoDB 援；AWS Lambda；Amazon SNS		

Summary

此模式結合了兩種模式，同時使用命令查詢責任區隔 (CQRS) 模式和事件來源模式。CQRS 模式分隔命令和查詢模型的責任。事件來源模式利用非同步事件驅動的通訊來改善整體使用者體驗。

您可以使用 CQRS 和 Amazon Web Services (AWS) 服務來獨立維護和擴展每個資料模型，同時將整體應用程式重構為微型服務架構。然後，您可以使用事件 sourcing 模式將資料從命令資料庫同步到查詢資料庫。

此模式使用範例程式碼，其中包含可以使用最新版 Visual Studio 開啟的解決方案 (*.sln) 檔案。此範例包含獎勵 API 程式碼，用於展示 CQRS 和事件來源如何在 AWS 無伺服器和傳統或現場部署應用程式中運作。

若要深入瞭解 CQRS 與事件來源，請參閱「其他資訊」一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon CloudWatch
- Amazon DynamoDB 資料表
- Amazon DynamoDB 串流

- AWS Identity and Access Management (IAM) 存取金鑰和秘密金鑰；如需詳細資訊，請參閱相關資源一節中的影片
- AWS Lambda
- 熟悉視覺工作室
- 熟悉適用 AWS Toolkit for Visual Studio；如需詳細資訊，請參閱相關資源一節中的 AWS Toolkit for Visual Studio 示範影片

產品版本

- 視覺工作室 2019 社區版。
- 適用於視覺工作室 2019 的 AWS 工具包。
- . 淨核心 3.1. 此元件是安裝中的一個選項。若要在安裝期間包含 .NET 核心，請選取 NET 核心跨平台開發。

限制

- 傳統內部部署應用程式 (ASP.NET 核心 Web API 和資料存取物件) 的範例程式碼不會隨附資料庫。但是，它帶有`CustomerData`內存對象，它充當模擬數據庫。提供的代碼足以讓您測試模式。

架構

源, 技術, 堆棧

- 核心網頁 API 專案
- 網頁伺服器
- 資料存取物件
- CRUD 模型

來源架構

在源體系結構中，CRUD 模型在一個應用程序中包含命令和查詢接口。如需範例程式碼，請參閱`CustomerDAO.cs` (附件)。

目標技術堆疊

- Amazon DynamoDB
- Amazon DynamoDB 串流
- AWS Lambda
- （可選）Amazon API Gateway
- （可選）Amazon Simple Notification Service (Amazon SNS)

目標架構

在目標體系結構中，命令和查詢接口是分開的。下圖所示的架構可透過 API Gateway 和 Amazon SNS 進行擴充。如需詳細資訊，請參閱其他資訊一節。

1. 命令 Lambda 函數會在資料庫上執行寫入作業，例如建立、更新或刪除。
2. 查詢 Lambda 函數會在資料庫上執行讀取作業，例如取得或選取。
3. 此 Lambda 函數會處理來自命令資料庫的 DynamoDB 串流，並更新查詢資料庫中的變更。

工具

工具

- 亞馬遜 DynamoDB — Amazon DynamoDB 是全受管的 NoSQL 資料庫服務，可提供快速且可預測的效能以及無縫的可擴展性。
- Amazon DynamoDB 串流 — DynamoDB Streams 會在任何 DynamoDB 表格中擷取一系列項目層級修改的時間順序。然後，它將此信息存儲在日誌中長達 24 小時。靜態加密功能會加密 DynamoDB Streams 中的資料。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- AWS 管理主控台 — AWS 管理主控台是一種 Web 應用程式，其中包含用於管理 AWS 服務的各種服務主控台。
- 視覺工作室 2019 社區版 — 視覺工作室 2019 是一個集成的開發環境 (IDE). 社區版對開源貢獻者是免費的。在這種模式中，您將使用 Visual Studio 2019 社區版打開，編譯和運行示例代碼。僅供檢視，您可以使用任何文字編輯器或視覺工作室程式碼。

- AWS Toolkit for Visual Studio 的 AWS 工具組 — 適用於視覺工作室的 AWS 工具組是適用於視覺工作室 IDE 的外掛程式。適用 AWS Toolkit for Visual Studio 可讓您更輕鬆地開發、偵錯和部署使用 AWS 服務的 .NET 應用程式。

Code

範例程式碼已附加。如需部署範例程式碼的指示，請參閱 Epics 一節。

史诗

開啟並建置解決方案

任務	描述	所需技能
打開解決方案。	1. 從「附件」區段下載範例原始程式碼 (CQRS-ES Code.zip)，然後解壓縮檔案。 2. 在 Visual Studio IDE 中，選擇 [檔案]、[開啟]、[專案解決方案]，然後瀏覽至您擷取原始程式碼的資料夾。 3. 選擇 [開啟]。整個解決方案會載入到視覺工作室。	應用程式開發人員
建置解決方案。	開啟方案的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [建置方案]。這將構建和編譯解決方案中的所有項目。它應該成功編譯。 解決方案資源管理器應顯示目錄結構。 • CQRS On-Premises Code Sample包含在內部部署使用 CQRS 的範例。	應用程式開發人員

任務	描述	所需技能
	• CQRS AWS Serverless 包含使用 AWS 無伺服器服務的所有 CQRS 和事件來源範例程式碼。	

建立動 DynamoDB 料表

任務	描述	所需技能
提供認證。	如果您還沒有存取金鑰，請參閱「相關資源」區段中的影片。 1. 在解決方案總管中，展開 CQRS AWS 無伺服器，然後展開建置解決方案資料夾。 2. 展開建置專案並檢視檔案。 Program.cs 3. 捲動至頂端Program.cs 並尋找Program() 。 4. 替換為您YOUR ACCESS KEY的帳戶訪問密鑰，並 YOUR SECRET KEY替換為您的帳戶密鑰。請注意，在生產環境中，您不會對密鑰進行硬編碼。相反地，您可以使用 AWS Secrets Manager 存放和擷取登入資料。	應用程式開發人員、資料工程師、DBA
建置專案。	若要建置專案，請開啟 Aws.apg.cqrses .build 專案的	應用程式開發人員、資料工程師、DBA

任務	描述	所需技能
	內容 (按一下滑鼠右鍵) 功能表，然後選擇 [建置]。	
建立並填入資料表。	若要建置資料表並使用種子資料填入資料表，請開啟 AWS.apg.cqrses.build 專案的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [偵錯]、[啟動新執行個體]。	應用程式開發人員、資料工程師、DBA
驗證表結構和數據。	若要進行驗證，請導覽至 AWS 資源管理器，然後展開 Amazon DynamoDB。它應該顯示的表。開啟每個表格以顯示範例資料。	應用程式開發人員、資料工程師、DBA

執行本機測試

任務	描述	所需技能
建置 CQRS 專案。	1. 開啟解決方案，然後瀏覽至 CQRS AWS 服務/CQRS/測試解決方案資料夾。 2. 在測試專案中，開啟 .cs，然後以您建立的身分與存取權管理系統金鑰取代和。BaseFunctionTest AccessKeySecretKey 3. 儲存變更。 4. 若要編譯並建置測試專案，請開啟專案的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [建置]。	應用開發人員，測試工程

任務	描述	所需技能
建立事件采購計划。	1. 導覽至 CQRS AWS 服務/事件來源/測試解決方案資料夾。 2. 在 AWS.APG.CQRSE 中。EventSourceLambda. 測試專案、開啟 BaseFunctionTest.cs，然後將AccessKey和SecretKey取代為您建立的 IAM 金鑰。 3. 儲存變更。 4. 若要編譯並建置測試專案，請開啟專案的內容 (按一下滑鼠右鍵) 功能表，然後選擇 [建置]。	應用開發人員，測試工程
運行測試。	要運行所有測試，請選擇查看，測試資源管理器，然後選擇在視圖中運行所有測試。所有測試都應通過，該測試由綠色複選標記圖標表示。	應用開發人員，測試工程

將 CQRS Lambda 函數發佈到 AWS

任務	描述	所需技能
發佈第一個 Lambda 函數。	1. 在 [方案總管] 中，開啟 AWS.APG.CQRSES 的內容 (按一下滑鼠右鍵) 功能表。CommandCreateLambda 專案，然後選擇「發佈到 AWS Lambda」。	應用 DevOps 程式開發人員、

任務	描述	所需技能
	2. 選取您要使用的設定檔，以及要部署 Lambda 函數的 AWS 區域，以及函數名稱。 3. 對於其餘欄位，請保留預設值，然後選擇 [下一步]。 4. 在 [角色名稱] 下拉式清單中，選取AWSLambda FullAccess。 5. 若要提供您的帳戶金鑰，請選擇 [新增]，然後輸入`AcessKey`做為變數，並輸入您的存取金鑰做為值。然後再次選擇添加，輸入`SecretKey` 作為變量，並輸入密鑰作為值。 6. 對於其餘欄位，請保留預設值，然後選擇「上傳」。Lambda 測試函數上傳之後，它會自動出現在視覺工作室中。 7. 針對下列專案重複步驟 1-6： • 阿姆斯特丹. 法律. CommandDeleteLambda • 阿姆斯特丹. 法律. CommandUpdateLambda • 阿姆斯特丹. 法律. CommandAddRewardLambda • 阿姆斯特丹. 法律. CommandRedeemRewardLambda	

任務	描述	所需技能
	• 阿姆斯特丹. 法律. QueryCustomerListLambda • 阿姆斯特丹. 法律. QueryRewqardLambda	
驗證功能上傳。	(選用) 您可以透過導覽至 AWS 資源管理器和擴充 AWS Lambda 來確認函數已成功載入。若要開啟測試視窗，請選擇 Lambda 函數 (按兩下)。	應用 DevOps 程式開發人員、

任務	描述	所需技能
測試 Lambda 函數。	1. 輸入請求資料，或從「其他資訊」區段中的測試資料複製範例請求資料。請務必選取適用於您正在測試之函數的資料。 2. 若要執行測試，請選擇呼叫。回應和所有錯誤都會顯示在 [回應] 文字方塊中，而且記錄會顯示在 [記錄] 文字方塊或 [CloudWatch 記錄檔] 中。 3. 若要驗證資料，請在 AWS 總管中選擇 DynamoDB 表格 (按兩下)。 所有 CQRS Lambda 專案都位於`CQRS AWS Serverless\CQRS\Command Microservice` 和 `CQRS AWS Serverless\CQRS\Command Microservice` 解決方案資料夾下。有關解決方案目錄和項目，請參閱其他信息部分中的源代碼目錄。	應用 DevOps 程式開發人員、

任務	描述	所需技能
發布剩餘的功能。	針對下列專案重複上述步驟： • 阿姆斯特丹. 法律. CommandDeleteLambda • 阿姆斯特丹. 法律. CommandUpdateLambda • 阿姆斯特丹. 法律. CommandAddRewardLambda • 阿姆斯特丹. 法律. CommandRedeemRewardLambda • 阿姆斯特丹. 法律. QueryCustomerListLambda • 阿姆斯特丹. 法律. QueryRewqardLambda	應用 DevOps 程式開發人員、

將 Lambda 函數設定為事件偵聽程式

任務	描述	所需技能
發佈客戶和獎勵 Lambda 事件處理常式。	若要發佈每個事件處理常式，請遵循上述 Epic 中的步驟。 項目位於CQRS AWS Serverless\Event Source\Customer Event和CQRS AWS Serverless\Event Source\Reward Event 解決方案文件夾下。如需詳細資訊，請參閱其他資訊一節中的原始程式碼目錄。	應用程式開發人員

任務	描述	所需技能
附加事件來源 Lambda 事件偵聽程式。	1. 使用您發佈 Lambda 專案時使用的相同帳戶登入 AWS 管理主控台。 2. 對於區域，請選取美國東部 1 或您在先前史詩中部署 Lambda 函數的區域。 3. 導覽至 L ambda 服務。 4. 選取 L `EventSourceCustomer` ambda 函數。 5. 選擇新增觸發程式。 6. 在觸發器組態下拉式清單中，選取 DynamoDB。 7. 在 DynamoDB 表格下拉式清單中，選取。cqrses-customer-cmd 8. 在「起始位置」下拉式清單中，選取「修剪地平線自」。修剪水平表示 DynamoDB 觸發程序將從最後一個 (未修剪) 串流記錄開始讀取，這是碎片中最舊的記錄。 9. 選取「啟用觸發器」核取方塊。 10對於其餘欄位，請保留預設值，然後選擇「新增」。 接聽程式成功附加至 DynamoDB 表後，它就會顯示在 Lambda 設計工具頁面上。	應用程式開發人員

任務	描述	所需技能
發佈並附加 L EventSour ceReward ambda 函數。	若要發佈和附加 `EventSour ceReward` Lambda 函數，請重複前兩個故事中的步驟，並cqrses-reward-cmd從 DynamoDB 表格下拉式清單中選取。	應用程式開發人員

測試和驗證動 DynamoDB 資料流和 Lambda 觸發器

任務	描述	所需技能
測試串流和 Lambda 觸發器。	1. 在視覺工作室中，導覽至 AWS 資源管理器。 2. 展開 AWS Lambda，然後選擇CommandRedeemRewar d函數 (按兩下)。在打開的功能窗口中，您可以測試該功能。 3. 在要求文字方塊中，以 JavaScript 物件標記法 (JSON) 格式輸入要求資料。有關請求示例，請參閱其他信息部分中的測試數據。 4. 選擇調用。	應用程式開發人員
使用 DynamodDB 獎勵查詢表進行驗證。	1. 開啟cqrses-reward-query表格。 2. 檢查兌換獎勵的客戶的積分。兌換的積分應從客戶的總積分中扣除。	應用程式開發人員
使用 CloudWatch 記錄進行驗證。	1. 導覽至 CloudWatch 並選擇 [記錄群組]。	應用程式開發人員

任務	描述	所需技能
	2. /aws/lambda/ EventSourceReward 記錄群組包含觸發程序的記錄檔。EventSourceReward 系統會記錄所有 Lambda 呼叫，包括您在 context.Logger.LogLine Lambda 程式碼Console.WriteLine 中放置的訊息。	
驗證 EventSourceCustomer 觸發器。	要驗證EventSourceCustomer 觸發器，請使用EventSourceCustomer 觸發器的各自客戶表和 CloudWatch 日誌重複此史詩中的步驟。	應用程式開發人員

相關資源

參考

- 視覺工作室 2019 社區版下載
- AWS Toolkit for Visual Studio
- 適用 AWS Toolkit for Visual Studio 使用
- AWS 上的無伺服器
- 使用案例和設計模式
- 马丁·福勒
- 馬丁·福勒事件採購

影片

- 適用於視覺化工作室示範的 AWS

- 如何為新的 IAM 使用者建立存取金鑰 ID？

其他資訊

CQRS 與事件來源

CQRS

CQRS 模式會將單一概念性作業模型 (例如資料存取物件單一 CRUD (建立、讀取、更新、刪除) 模型分隔為指令和查詢作業模型。指令模型參照任何變更狀態的作業，例如建立、更新或刪除。查詢模型是指返回一個值的任何操作。

1. 客戶 CRUD 模型包含下列介面：
   - `Create Customer()`
   - `UpdateCustomer()`
   - `DeleteCustomer()`
   - `AddPoints()`
   - `RedeemPoints()`
   - `GetVIPCustomers()`
   - `GetCustomerList()`
   - `GetCustomerPoints()`

隨著您的需求變得越來越複雜，您可以從這種單一模型方法移動。CQRS 使用命令模型和查詢模型來分開寫入和讀取數據的責任。這樣，數據可以獨立維護和管理。有了明確的責任分離，每個模型的增強功能不會影響到另一個模型。這種分離可改善維護和效能，並且隨著應用程式的成長而降低複雜性。

1. 客戶指令模型中的介面：
   - `Create Customer()`
   - `UpdateCustomer()`
   - `DeleteCustomer()`
   - `AddPoints()`
   - `RedeemPoints()`

2. 客戶查詢模型中的介面：
   - `GetVIPCustomers()`
   - `GetCustomerList()`
   - `GetCustomerPoints()`
   - `GetMonthlyStatement()`

如需範例程式碼，請參閱原始程式碼目錄。

然後 CQRS 模式將數據庫分離。這種解耦導致了每個服務的完全獨立性，這是微服務架構的主要組成部分。

在 AWS 雲端使用 CQRS，您可以進一步優化每項服務。例如，您可以設定不同的運算設定，或選擇無伺服器或容器型微服務。您可以使用 Amazon 取代現場部署快取 ElastiCache。如果您有現場部署發佈/訂閱簡訊，則可以使用 Amazon Simple Notification Service (Amazon SNS) 來取代它。此外，您還可以利用 pay-as-you-go 定價和廣泛的 AWS 服務，這些服務只需按使用量付費。

CQRS 包括以下好處：

- 獨立擴展 — 每個模型都可以調整其擴展策略，以滿足服務的需求和需求。與高效能應用程式類似，分離讀取和寫入可讓模型獨立擴充以滿足每個需求。您也可以新增或減少運算資源，以解決某個模型的延展性需求，而不會影響另一個模型。
- 獨立維護 — 分離查詢和命令模型可提高模型的可維護性。您可以對一個模型進行程式碼變更和增強，而不會影響另一個模型。
- 安全性 — 將權限和策略應用於單獨的模型以進行讀取和寫入更容易。
- 最佳化讀取 — 您可以定義針對查詢最佳化的結構描述。例如，您可以為彙總資料定義結構定義，並為事實資料表定義個別的結構描述。
- 集成 — CQRS 非常適合基於事件的編程模型。
- 受管理的複雜性 — 分離為查詢和指令模型適用於複雜的領域。

使用 CQRS 時，請記住以下注意事項：

- CQRS 模式僅適用於應用程式的特定部分，而不適用於整個應用程式。如果在不符合模式的網域上實作，它可以降低生產力、增加風險並引入複雜性。
- 該模式最適合具有不平衡讀取和寫入操作的常用模型。

- 對於需要大量讀取的應用程式 (例如需要花費時間處理的大型報表)，CQRS 可讓您選取正確的資料庫並建立結構描述來儲存彙總資料。如此一來，只會處理報告資料一次，並將其傾印到彙總表格中，藉此改善讀取和檢視報表的回應時間。
- 對於大量寫入的應用程式，您可以設定資料庫以進行寫入作業，並在寫入需求增加時允許指令微服務獨立擴充。如需範例，請參閱AWS.APG.CQRSES.CommandRedeemRewardLambda和AWS.APG.CQRSES.CommandAddRewardLamb服務。

事件來源

下一個步驟是在執行指令時，使用事件 sourcing 來同步查詢資料庫。例如，請考慮下列事件：

- 添加了客戶獎勵積分，要求更新查詢數據庫中的客戶總計或匯總獎勵積分。
- 客戶的姓氏會在命令資料庫中更新，這需要更新查詢資料庫中的代理客戶資訊。

在傳統的 CRUD 模型中，您可以鎖定資料直到完成交易，以確保資料的一致性。在事件來源中，資料會透過發佈一系列事件進行同步處理，訂閱者會使用這些事件來更新其個別資料。

事件來源模式可確保並記錄對資料所採取的全系列動作，並透過一系列事件發佈資料。這些事件代表該事件的訂閱者必須處理以保持其記錄更新的一組資料變更。訂戶會使用這些事件，同步化訂戶資料庫上的資料。在這種情況下，這是查詢數據庫。

下圖顯示在 AWS 上搭配 CQRS 使用的事件來源。

1. 命令 Lambda 函數會在資料庫上執行寫入作業，例如建立、更新或刪除。
2. 查詢 Lambda 函數會在資料庫上執行讀取作業，例如取得或選取。
3. 此 Lambda 函數會處理來自命令資料庫的 DynamoDB 串流，並更新查詢資料庫中的變更。您也可以使用此功能將訊息發佈到 Amazon SNS，以便其訂閱者可以處理資料。
4. (選擇性) Lambda 事件訂閱者會處理 Amazon SNS 發佈的訊息，並更新查詢資料庫。
5. (選擇性) Amazon SNS 會傳送寫入作業的電子郵件通知。

在 AWS 上，查詢資料庫可以透過 DynamoDB Streams 進行同步。DynamoDB 會在 DynamoDB 資料表中以近乎即時的方式擷取一系列項目層級修改的時間順序，並在 24 小時內持久儲存資訊。

啟用 DynamoDB Streams 可讓資料庫發佈一系列事件，使事件來源模式成為可能。事件采購模式會添加事件訂戶。事件訂閱者應用模組會耗用事件，並根據訂戶的責任來處理事件。在上圖中，事件訂閱者會將變更推送至查詢 DynamoDB 資料庫，以保持資料同步。使用 Amazon SNS、訊息代理程式和事件訂閱者應用程式可保持架構分離。

事件來源包含下列優點：

- 交易資料的一致性
- 可靠的審計跟踪和操作的歷史記錄，可用於監視數據中採取的操作
- 允許分散式應用程式 (例如微服務) 在整個環境中同步處理其資料
- 每當狀態變化時，可靠地發布事件
- 重建或重新顯示過去的狀態
- 鬆散耦合的實體，可交換事件以便從整合式應用程式移轉至微服務
- 減少並行更新所造成的衝突；事件來源可避免直接在資料倉庫中更新物件的要求
- 解耦任務和事件的靈活性和可擴展性
- 外部系統更新
- 在單個事件中管理多個任務

使用事件來源時，請記住下列警告：

- 因為來源訂閱者資料庫之間的資料更新有些延遲，所以復原變更的唯一方法是將補償事件新增至事件存放區。
- 實施事件來源有一個學習曲線，因為它的編程風格不同。

測試數據

成功部署後，請使用下列測試資料測試 Lambda 函數。

CommandCreate 顧客

```
{  "Id":1501,  "Firstname":"John",  "Lastname":"Done",  "CompanyName":"AnyCompany",
 "Address": "USA",  "VIP":true }
```

CommandUpdate 顧客

```
{  "Id":1501,  "Firstname":"John",  "Lastname":"Doe",  "CompanyName":"Example Corp.",
 "Address": "Seattle, USA",  "VIP":true }
```

CommandDelete 顧客

輸入客戶 ID 作為請求數據。例如，如果客戶識別碼為 151，請輸入 151 做為請求資料。

```
151
```

QueryCustomerList

這是空白的。當它被調用時，它將返回所有客戶。

CommandAddReward

這將為 ID 1（理查德）的客戶增加 40 點。

```
{
  "Id":10101,
  "CustomerId":1,
  "Points":40
}
```

CommandRedeemReward

這將向 ID 1（理查德）的客戶扣除 15 分。

```
{
  "Id":10110,
  "CustomerId":1,
  "Points":15
}
```

QueryReward

輸入客戶的識別碼。例如，輸入 1 做為理查德，2 代表阿爾奈姆，輸入 3 做為雪莉。

```
2
```

原始碼目錄

請使用下表做為 Visual Studio 解決方案目錄結構的指南。

CQRS 本機程式碼範例解決方案目錄

客戶 CRUD 模型

CQRS 內部部署程式碼範例\ CRUD 模型\ 專案

客戶 CRUD 模型的 CQRS 版本

- 客戶指令：CQRS On-Premises Code Sample\CQRS Model\Command Microservice \AWS.APG.CQRSES.Command專案
- 客戶查詢：CQRS On-Premises Code Sample\CQRS Model\Query Microservice \AWS.APG.CQRSES.Query項目

命令和查詢微服務

命令微服務位於解決方案文件夾CQRS On-Premises Code Sample\CQRS Model\Command Microservice下：

- AWS.APG.CQRSES.CommandMicroserviceASP.NET 核心 API 專案充當消費者與服務互動的入口點。
- AWS.APG.CQRSES.Command.NET 核心項目是託管命令相關的對象和接口的對象。

查詢微服務位於解決方案文件夾CQRS On-Premises Code Sample\CQRS Model\Query Microservice下：

- AWS.APG.CQRSES.QueryMicroserviceASP.NET 核心 API 專案充當消費者與服務互動的入口點。
- AWS.APG.CQRSES.Query.NET 核心項目是託管查詢相關的對象和接口的對象。

CQRS AWS 無伺服器程式碼解決方案目錄

此程式碼是使用 AWS 無伺服器服務的現場部署程式碼的 AWS 版本。

在 C# .NET 核心，每個 Lambda 函數由一個 .NET 核心項目表示。在此模式的示例代碼中，命令和查詢模型中的每個接口都有一個單獨的項目。

使用 AWS 服務的 CQRS

您可以在資料夾中找到使用 AWS 無伺服器服務的 CQRS 根解決方案目錄。CQRS AWS Serverless \CQRS該示例包括兩種模型：客戶和獎勵。

客戶和獎勵的命令 Lambda 函數位於CQRS\Command Microservice\Customer和CQRS\Command Microservice\Reward資料夾下。它們包含下列 Lambda 專案：

- 客戶指令：CommandCreateLambdaCommandDeleteLambda、和 CommandUpdateLambda
- 獎勵指令：CommandAddRewardLambdaCommandRedeemRewardLambda

客戶和獎勵的查詢 Lambda 函數位於CQRS\Query Microservice\Customer和CQRS \QueryMicroservice\Reward資料夾下方。它們包含QueryCustomerListLambda和 QueryRewardLambda Lambda 項目。

CQRS 測試專案

測試專案位於CQRS\Tests資料夾下。此專案包含用於自動測試 CQRS Lambda 函數的測試指令碼。

使用 AWS 服務採購事件

下列 Lambda 事件處理常式由客戶和獎勵 DynamoDB 串流啟動，以處理和同步查詢表格中的資料。

- L EventSourceCustomer ambda 函數會對應至客戶資料表 (cqrses-customer-cmd) DynamoDB 串流。
- L EventSourceReward ambda 函數會對應至獎勵資料表 DynamoDB 資cqrses-reward-cmd料流。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

更多模式

- ???
- 使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目
- 使用 DR 協調器架構自動化跨區域容錯移轉和容錯回復
- 使用自動化遷移策略識別和規劃 AppScore
- 使用 CI/CD 管道自動建置 Java 應用程式並將其部署到 Amazon EKS
- 使用 AWS CDK 為微型服務自動建置 CI/CD 管道和 Amazon ECS 叢集
- 使用 BMC AMI 雲端資料將大型主機資料備份並存檔到 Amazon S3
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 容器化已由 Blu Age 現代化的大型主機工作負載
- 從 AWS 儲存庫持續部署現代 AWS Amplify Web 應用程式 CodeCommit
- 使用 Python 在 AWS 上將 EBCDIC 資料轉換並解壓縮為 ASCII
- 使用 Micro Focus 轉換具有複雜記錄佈局的大型主機資料檔案
- ???
- 使用建立管道並將成品更新部署到現場部署 EC2 執行個體 CodePipeline
- 部署和偵錯 Amazon EKS 叢集
- 使用 Elastic Beanstalk 部署容器
- 使用與 PostgreSQL 相容的 Aurora 全球資料庫來模擬甲骨文 DR
- 在中使用 AWS 大型主機現代化和 Amazon Q 產生資料見解 QuickSight
- 使用 Amazon RDS for Oracle 文 SQL 開發人員和 AWS SCT 從亞馬遜 RDS 向亞馬遜 RDS
- 整合石分支通用控制器與 AWS 大型主機現代化
- 管理多個 AWS 帳戶和 AWS 區域的 AWS 服務目錄產品
- 將 AWS 成員帳戶從 AWS Organizations 遷移到 AWS Control Tower
- 使用精確 Connect 將 VSAM 文件遷移和複寫到 Amazon RDS 或 Amazon MSK
- 使用 AWS DMS 從 SAP ASE 遷移到亞馬遜 RDS 適用於 SQL 伺服器
- 將甲骨文外部表遷移到 Amazon Aurora PostgreSQL 兼容
- 使用微焦點企業伺服器和 LRS VPSX/MFI，在 AWS 上現代化大型主機批次列印工作負載
- ???
- 使用 OpenText 微焦點企業伺服器和 LRS X 在 AWS 上現代化大型主機輸出管理 PageCenter
- ???

- 優化 AWS 應用程序容器生成的碼頭映像
- 使用精確 Connect 將大型主機資料庫複寫到 AWS
- WorkSpaces 使用 Amazon ECS 隨時隨地在 Amazon 上運行 Amazon ECS Anywhere 務
- 在 Amazon S3 中設置頭盔 v3 圖表存儲庫
- 在多區域、多帳戶組織中設定 AWS CloudFormation 漂移偵測
- 使用 AWS Lambda 在六角形架構中建構 Python 專案
- 將心臟起搏器叢集從 ENSA1 升級至 ENSA2
- 用 CloudEndure 於內部部署資料庫的嚴重損壞復原
- 在本機驗證地形表單 (AFT) 程式碼的 Account Factory

聯網

主題

- 使用 AWS Transit Gateway 自動化區域間對等互連的設定
- 使用 AWS Transit Gateway 集中網路連線
- 使用 Application Load Balancer 衡器在 Oracle EnterpriseOne 上為 Oracle WebLogic JD 愛德華設定 HTTPS 加密
- 透過私人網路 Connect 至應用程式移轉服務資料和控制平面
- 使用 AWS CloudFormation 自訂資源和 Amazon SNS 建立 Infoblox 物件
- 自訂 AWS Network Firewall 的 Amazon CloudWatch 提醒
- 將 DNS 記錄批量遷移到 Amazon Route 53 私有託管區域
- 在 AWS 上從 F5 遷移到 Application Load Balancer 時修改 HTTP 標頭
- 從多個 VPC 私有存取中央 AWS 服務端點
- 針對多個 AWS 帳戶的傳入網際網路存取建立網路存取分析器發現的報告
- 使用 AWS Organizations 自動標記 Transit Gateway 附件
- 確認 ELB 負載平衡器需要 TLS 終止
- 使用 Splunk 檢視 AWS Network Firewall 日誌和指標
- 更多模式

使用 AWS Transit Gateway 自動化區域間對等互連的設定

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：網路；混合雲	AWS 服務：AWS Transit Gateway；AWS Step Functions；AWS Lambda

Summary

AWS Transit Gateway 透過中央集線器連接虛擬私有雲 (VPC) 和現場部署網路。Transit Gateway 流量始終保留在全球 Amazon Web Services (AWS) 骨幹網上，而且不會周遊公用網際網路，進而減少常見漏洞攻擊和分散式拒絕服務 (DDoS) 攻擊等威脅媒介。

如果您需要在兩個或多個 AWS 區域之間進行通訊，可以使用區域間傳 Transit Gateway 道對等在不同區域的傳輸閘道之間建立對等連接。不過，使用「Transit Gateway」手動設定區域間對等互連可能是一項耗時的程序，需要多個步驟。此模式提供了一個自動化程序，可透過使用程式碼執行對等連線來移除這些手動步驟。如果在多區域組織設定期間必須重複設定多個區域和 AWS 帳戶，則可以使用此方法。

此模式使用 AWS CloudFormation 堆疊，其中包括 AWS Step Functions 工作流程、AWS Lambda 函數、AWS Identity and Access Management (IAM) 角色，以及 Amazon CloudWatch 日誌中的日誌群組。然後，您可以開始執行 Step Functions，並為傳輸閘道建立區域間對等連線。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的 Amazon Simple Storage Service (Amazon S3) 存儲桶。
- 運輸閘道，在請求者區域和接受器區域中創建和配置。請求者區域是產生對等請求的地方，且接受者區域接受對等請求。如需這方面的詳細資訊，請參閱 Amazon VPC 文件中的建立和接受 VPC 對等連線。
- VPC，安裝和配置在接受器和請求者區域。如需建立 VPC 的步驟，請參閱 Amazon VPC 說明文件中的從 Amazon VPC 入門建立 VPC。
- VPC 必須使用標`addToTransitGateway`籤和`true`值。

- 根據您的需求設定 VPC 的安全性群組和網路存取控制清單 (ACL)。如需此相關資訊，請參閱 Amazon VPC 說明文件中的 VPC 和網路 ACL 的安全群組。

AWS 區域和限制

- 只有特定 AWS 區域支援區域間對等。如需支援區域間對等的區域完整清單，請參閱 AWS Transit Gateway 常見問答集。
- 在附加的示例代碼中，請求者區域被假定為`us-east-2`，並假定接受器區域為。`us-west-2`如果要配置不同的區域，則必須在所有 Python 文件中編輯這些值。若要實作涉及兩個以上區域的更複雜的設定，您可以變更步驟函數，將區域做為參數傳遞至 Lambda 函數，然後針對每個組合執行函數。

架構

該圖顯示了具有以下步驟的工作流程：

1. 使用者建立 AWS CloudFormation 堆疊。
2. AWS CloudFormation 會建立使用 Lambda 函數的 Step Functions 函數狀態機器。如需詳細資訊，請參閱 AWS Step Functions 文件中的建立使用 Lambda 的步驟函數狀態機器。
3. Step Functions 數會呼叫 Lambda 函數進行對等互連。
4. Lambda 函數會在傳輸閘道之間建立對等連線。
5. Step Functions 數調用 Lambda 函數進行路由表修改。
6. Lambda 函數會新增 VPC 的無類別網域間路由 (CIDR) 區塊來修改路由資料表。

Step Functions 工作流

此圖表顯示下列「Step Functions」工作流程：

1. Step Functions 數工作流程會呼叫傳輸閘道對等的 Lambda 函數。
2. 有一個計時器呼叫等待一分鐘。
3. 對等連接狀態會被擷取並傳送至條件區塊。該塊負責循環。
4. 如果不符合成功條件，則會將工作流程編碼為進入計時器階段。

5. 如果符合成功條件，則會呼叫 Lambda 函數來修改路由資料表。此呼叫之後，「Step Functions」工作流程會結束。

工具

- AWS CloudFormation — AWS CloudFormation 是一項可協助您建立 AWS 資源模型和設定 AWS 資源的服務。
- Amazon CloudWatch 日誌 — CloudWatch 日誌可協助您集中管理您使用的所有系統、應用程式和 AWS 服務的日誌。
- AWS Identity and Access Management (IAM) — IAM 是一種用於安全控制 AWS 服務存取的 Web 服務。
- AWS Lambda — Lambda 會在高可用性運算基礎設施上執行程式碼，並執行運算資源的所有管理作業。
- AWS Step Functions — Step Functions 可讓您輕鬆協調分散式應用程式的元件，做為視覺化工作流程中的一系列步驟。

史诗

自動化對等互連

任務	描述	所需技能
將附加的檔案上傳到 S3 儲存貯體。	登入 AWS 管理主控台，開啟 Amazon S3 主控台，然後將`modify-transit-gateway-routes.zip`、`peer-transit-gateway.zip`、和`get-transit-gateway-peering-status.zip` 檔案 (附加) 上傳到 S3 儲存貯體。	一般 AWS
建立 AWS CloudFormation 堆疊。	執行下列命令以使用`transit-gateway-peering.json` 檔案 (附加) 建	DevOps 工程師

任務	描述	所需技能
	立 AWS CloudFormation 堆疊： `aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json` AWS CloudFormation 堆疊可建立 Step Functions 工作流程、Lambda 函數、IAM 角色和 CloudWatch 日誌群組。 請確定 AWS CloudFormation 範本參照包含您先前上傳之檔案的 S3 儲存貯體。 注意：您也可以使用 AWS CloudFormation 主控台建立堆疊。如需詳細資訊，請參閱 AWS CloudFormation 文件中的在 AWS CloudFormation 主控台上建立堆疊。	

任務	描述	所需技能
在 Step Functions 中啟動新的執行。	開啟「Step Functions 式」主控台並開始新的執行。Step Functions 數會呼叫 Lambda 函數，並為傳輸閘道建立對等連線。您不需要輸入 JSON 檔案。確認附件是否可用，且連線類型為「對等互連」。 如需詳細資訊，請參閱 AWS 步驟函數文件中的 AWS 步驟函數開始使用 AWS 步驟函數的開始新執行。	DevOps 工程師，一般 AWS
驗證路由表中的路由。	區域間對等互連是在運輸閘道之間建立的。路由表會以對等區域 VPC 的 IPv4 CIDR 區塊範圍進行更新。 開啟 Amazon VPC 主控台，然後在路由表中選擇與傳輸閘道附件對應的「關聯」索引標籤。確認對等區域的 VPC CIDR 區塊範圍。 如需詳細步驟和指示，請參閱 Amazon VPC 文件中的關聯傳輸閘道路由表。	網路管理員

相關資源

- Step Functions 數中的執行
- 傳輸閘道對等連接附件
- 使用 AWS 傳輸閘道跨 AWS 區域互連 VPC-示範 (影片)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Transit Gateway 集中網路連線

由邁德希利帕拉格米（AWS）和尼基爾馬拉普（AWS）創建

環境：生產	技術：網路	AWS 服務：AWS Transit Gateway；Amazon VPC

Summary

此模式描述最簡單的組態，其中 AWS Transit Gateway 可用來將現場部署網路連接到 AWS 區域內多個 AWS 帳戶中的虛擬私有雲 (VPC)。使用此設定，您可以建立混合式網路，以連接一個區域和內部部署網路中的多個 VPC 人雲端網路。這是透過使用傳輸閘道和內部部署網路的虛擬私人網路 (VPN) 連線來完成。

先決條件和限制

先決條件

- 用於託管網路服務的帳戶，以 AWS Organizations 中組織的成員帳戶進行管理
- VPC 位於多個 AWS 帳戶，沒有重疊的無類別網域間路由 (CIDR) 區塊

限制

此病毒碼不支援隔離特定 VPC 或內部部署網路之間的流量。連接到傳輸閘道的所有網絡都可以互相連接。若要隔離流量，您需要在傳輸閘道上使用自訂路由表。此模式只會使用單一預設傳輸閘道路由表 (最簡單的組態) 來連接 VPC 和內部部署網路。

架構

目標技術堆疊

- AWS Transit Gateway
- AWS Site-to-Site VPN
- VPC
- AWS Resource Access Manager (AWS RAM)

目標架構

工具

AWS 服務

- AWS Resource Access Manager (AWS RAM) 可協助您透過 AWS 組織在 AWS 帳戶、組織單位或整個組織之間安全地共用資源。
- AWS Transit Gateway 是連接虛擬私有雲 (VPC) 和現場部署網路的中央中樞。

史诗

在網路服務帳戶中建立傳輸閘道

任務	描述	所需技能
建立傳輸閘道。	在您要託管網路服務的 AWS 帳戶中，在目標 AWS 區域建立傳輸閘道。如需指示，請參閱建立傳輸閘道。注意下列事項： • 選擇預設路由表關聯。 • 選取預設路由表格傳輸。	網路管理員

將傳輸閘道 Connect 到內部部署網路

任務	描述	所需技能
為 VPN 連線設定客戶閘道裝置。	客戶閘道裝置附加在傳輸閘道與內部部署網路之間的站對站 VPN 連線的內部部署端。如需詳細資訊，請參閱 AWS 站	網路管理員

任務	描述	所需技能
	Site-to-Site VPN 文件中的客戶閘道裝置。識別或啟動支援的內部部署客戶裝置，並記下其公用 IP 位址。VPN 配置在這個史詩後面完成。	
在網路服務帳戶中，建立傳輸閘道的 VPN 附件。	若要設定連線，請為傳輸閘道建立 VPN 附件。如需指示，請參閱傳輸閘道 VPN 附件。	網路管理員
在內部部署網路中的客戶閘道裝置上設定 VPN。	下載與傳輸閘道相關聯之 Site-to-Site VPN 連線的組態檔案，並在客戶閘道裝置上設定 VPN 設定。如需指示，請參閱下載組態檔案。	網路管理員

將網路服務帳戶中的傳輸閘道共用到其他 AWS 帳戶或您的組織

任務	描述	所需技能
在 AWS Organizations 管理帳戶中，開啟共用功能。	若要與您的組織或特定組織單位共用傳輸閘道，請在 AWS Organizations Organization 中開啟共用功能。否則，您將需要分別共享每個帳戶的傳輸閘道。如需指示，請參閱啟用 AWS Organizations 內的資源共用。	AWS 系統管理員
在網路服務帳戶中建立傳輸閘道資源共用。	若要允許組織內其他 AWS 帳戶中的 VPC 連線到傳輸閘道，請在網路服務帳戶中使用 AWS RAM 主控台共用傳輸閘道資源。如需指示，請參閱建立資源共用。	AWS 系統管理員

將 VPC Connect 到傳輸閘道

任務	描述	所需技能
在個別帳戶中建立 VPC 附件。	在傳輸閘道已共用的帳戶中，建立傳輸閘道 VPC 附件。如需指示，請參閱建立 VPC 的傳輸閘道附件。	網路管理員
接受 VPC 附件請求。	在網路服務帳戶中，接受傳輸閘道 VPC 附件要求。如需相關指示，請參閱接受共用附件。	網路管理員

設定路由

任務	描述	所需技能
在個別帳戶 VPC 中設定路由。	在每個個別帳戶 VPC 中，使用傳輸閘道做為目標，將路由新增至內部部署網路和其他 VPC 網路。如需指示，請參閱從路由表中新增和移除路由。	網路管理員
在傳輸閘道路由表中設定路由。	應該會傳播來自 VPC 和 VPN 連線的路由，並且應該會顯示在傳輸閘道預設路由表格中。如有需要，請在傳輸閘道預設路由表格中建立任何靜態路由 (其中一個範例是靜態 VPN 連線的靜態路由)。如需指示，請參閱建立靜態路由。	網路管理員
新增安全性群組和網路存取控制清單 (ACL) 規則。	對於 VPC 中的 EC2 執行個體和其他資源，請確保安全群組規則和網路 ACL 規則允許 VPC 和現場部署網路之間的流量。如需指示，請參閱使用安	網路管理員

任務	描述	所需技能
	全性群組控制資源的流量和從 ACL 新增和刪除規則。	

測試連接

任務	描述	所需技能
測試 VPC 之間的連線能力。	確定網路 ACL 和安全性群組允許網際網路控制訊息通訊協定 (ICMP) 流量，然後從 VPC 中的執行個體偵測到另一個也連線至傳輸閘道的 VPC。	網路管理員
測試 VPC 與內部部署網路之間的連線。	確保網路 ACL 規則、安全群組規則和任何防火牆允許 ICMP 流量，然後在現場部署網路和 VPC 中的 EC2 執行個體之間執行 Ping。必須先從內部部署網路啟動網路通訊，才能使 VPN 連線進入UP狀態。	網路管理員

相關資源

- 建立可擴展且安全的多 VPC AWS 網路基礎設施 (AWS 白皮書)
- 使用共用資源 (AWS 記憶體文件)
- 使用傳輸閘道 (AWS Transit Gateway 文件)

使用 Application Load Balancer 衡器在 Oracle EnterpriseOne 上為 Oracle WebLogic JD 愛德華設定 HTTPS 加密

環境：生產	技術：網路；安全性、身分識別、合規性	工作量：甲骨文
AWS 服務：AWS Certificate Manager (ACM)；Elastic Load Balancing (ELB)；Amazon Route 53		

Summary

此模式說明如何在 Oracle 工作負載 EnterpriseOne 上為 Oracle JD 愛德華中的 SSL 卸載設定 HTTPS 加密。 WebLogic 這種方法會加密使用者的瀏覽器和負載平衡器之間的流量，以消除 EnterpriseOne 伺服器的加密負擔。

許多使用者使用 AWS Application Load Balancer 來水平擴展 EnterpriseOne JAVA 虛擬機器 (JVM) 層。負載平衡器可做為用戶端的單一聯絡點，並在多個 JVM 之間分配傳入流量。或者，負載平衡器可以將流量分配到多個可用區域，並增加的可用性 EnterpriseOne。

此模式中描述的程序會在瀏覽器和負載平衡器之間設定加密，而不是加密負載平衡器和 JVM 之間的流量。 EnterpriseOne 這種方法稱為 SSL 卸載。將 SSL 解密程序從 EnterpriseOne Web 或應用程式伺服器卸載至應用程 Application Load Balancer 器，可減輕應用程式端的負擔。在負載平衡器終止 SSL 之後，未加密的流量會路由到 AWS 上的應用程式。

Oracle JD Edwards EnterpriseOne 是一種企業資源規劃 (ERP) 解決方案，適用於製造、建構、配送、服務或管理產品或實體資產的組織。JD Edwards EnterpriseOne 支援各種硬體、作業系統和資料庫平台。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Identity and Access Management (IAM) 角色，具有進行 AWS 服務呼叫和管理 AWS 資源的許可

- 一個 SSL 憑證

產品版本

- 此模式已通過 Oracle WebLogic 12c 進行了測試，但您也可以使用其他版本。

架構

有多種方法可以執行 SSL 卸載。此模式使用「Application Load Balancer 器」和「Oracle HTTP 伺服器」(OHS)，如下圖所示。

下圖顯示 JD 愛德華 EnterpriseOne、Application Load Balancer 器和 Java 應用程式伺服器 (JAS) JVM 配置。

工具

AWS 服務

- 應用程式負載平衡器將傳入的應用程式流量分配到多個可用區域中的多個目標，例如 Amazon 彈性運算雲端 (Amazon EC2 執行個體)。
- AWS Certificate Manager (ACM) 可協助您建立、存放和更新公有和私有 SSL/TLS X.509 憑證和金鑰，以保護您的 AWS 網站和應用程式。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。

最佳實務

- 如需 ACM 最佳作法，請參閱 ACM 文件。

史诗

設置 WebLogic 和職安健

任務	描述	所需技能
安裝並設定 Oracle 元件。	1. 按照標準安裝過程安裝 Fusion 中間件基礎結構。該程序可幫助您安裝和配置 WebLogic 域。如需相關指示，請參閱 Oracle 說明文件。 2. 依照標準安裝程序安裝 OHS。如需相關指示，請參閱 Oracle 說明文件。 3. 安裝完成後，啟動設定精靈 (`config.sh` 檔案) 以設定 OHS。 • 您可以更新現有領域或建立新領域。此模式假設您正在更新現有的網域。 • 針對可用的樣版，請選擇「Oracle 企業管理系統-限制的 JRF」與「Oracle HTTP 伺服器 (受限 JRF)」。選取這些「Java 必要檔案」(JRF) 選項可免除與外部資料庫的連線。 • 對於受管理的伺服器、叢集、伺服器範本、Coherence 叢集、機器、將伺服器指派給電腦、虛擬目標和磁碟分割，請接受預設組態值	JDE 數控系統管理 WebLogic 員

任務	描述	所需技能
	，然後選擇下一步以移至下一個類別。 • 完成 OHS 執行個體的組態詳細資料 (例如，管理員主機和連接埠、監聽位址和連接埠、伺服器名稱ohs1)。	
在網域層級啟用 WebLogic 外掛程式。	該 WebLogic 插件是負載平衡所需的。若要啟用外掛程式： 1. 使用以下連結登入 WebLogic 管理主控台： `http://<WeblogicServer>:<Adminport>/console` 2. 選擇鎖定和編輯，然後選擇配置，Web 應用程序。 3. 選擇WebLogic 插件已啟用（複選框或下拉選項）。 4. 選擇「儲存並啟用變更」。	JDE 數控系統管理 WebLogic 員

任務	描述	所需技能
編輯組態檔案。	該mod_wl_ohs.conf 文件將代理請求從 OHS 配置為. WebLogic 1. 編輯此檔案。它位於： $ORACLE_HOME/user_projects/domains/ 例如： /home/oracle/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1 2. 添加 WebLogic 主機（WebLogicHost ）和端口（WebLogicPort ）值（此模式假定本地主機和端口 8000。） 3. 添加WLProxySSL 和WLProxySSLPassThrough 值，如下所示： <VirtualHost *:8000> <Location /jde> WLSRequest On SetHandler weblogic-handler WebLogicHost localhost WebLogicPort 8000	JDE 數控系統管理 WebLogic 員

任務	描述	所需技能
	<pre>WLProxySSL On WLProxySSLPassThrough On </Location> </VirtualHost></pre>	

任務	描述	所需技能
使用企業管理員啟動 OHS。	1. 登錄到企業管理器融合中間件通過使用鏈接： `http://<WeblogicServer>:<Adminport>/em/` 2. 在「目標導覽」的「HTTP 伺服器」下，選取 OHS 執行個體 (例如，ohs1)。 3. 選擇 [關機] 和 [啟動] 以重新啟動 OHS 執行個體。 4. OHS 設定完成後，您可以使用具有連接埠 8000 的 HTTP 伺服器主機名稱而非伺服器主機名稱來連線至 EnterpriseOne HTML 用戶端。 EnterpriseOne • 舊鏈接：`http://<Webserver>:80/jde/owhtml` • 新增連結：`http://<HTTP server or web server>:8000/jde/owhtml` 如果您使用預設 Oracle HTTP 連接埠以外的連接埠，請編輯`httpd.conf` 檔案，以在兩個位置新增該連接埠的監聽器： `#[Listen] OHS_LISTEN_PORT` `Listen 8000`	JDE 數控系統管理 WebLogic 員

任務	描述	所需技能
	以及： `# ServerName <Weblogic Server1>:8000`	

設定應用程式負載平衡器

任務	描述	所需技能
設定目標群組。	1. 為 HTTP 伺服器連接埠 8000 建立目標群組。 2. 使用相同的連接埠註冊目標群組下的目標。 3. 檢查目標的狀態，以確認它們是否正常。 4. 視需要設定健全狀況檢查設定。 如需詳細指示，請參閱 E lastic Load Balancing 文件。	AWS 管理員
設定負載平衡器。	1. 建立具有預設屬性和所需虛擬私有雲端 (VPC)、安全群組和子網路的 Applicati on Load Balancer。如需指示，請參閱 E lastic Load Balancing 文件。 2. 新增 HTTPS 443 的接聽程式項目，並將其轉寄至您在上一個步驟中建立的目標群組。如需指示，請參閱 E lastic Load Balancing 文	AWS 管理員

任務	描述	所需技能
	件。) HTTPS 接聽程式需要 SSL 憑證。您可以從 ACM 選擇憑證或上傳一個憑證。 3. 對於這兩個偵聽程式，請遵循 E lastic Load Balancing 文件中的指示來啟用黏性。	
新增 Route 53 (DNS) 記錄。	(選擇性) 您可以為子網域新增 Amazon Route 53 DNS 記錄。此記錄會指向您的 Application Load Balancer。如需指示，請參閱 Route 53 文件。	AWS 管理員

故障診斷

問題	解決方案
HTTP 伺服器不會出現。	如果 HTTP 伺服器未顯示在 [企業管理員] 主控台的 [目標導覽] 清單中，請依照下列步驟執行： 1. 在「WebLogic 網域、管理」下，選擇「OHS 執行個體」。 2. 選擇建立以建立新的 OHS 執行個體。 3. 提供執行個體名稱，然後選擇 [確定] 建立執行個體。 建立執行個體並啟動變更後，您就可以在「目標導覽」面板中看到 HTTP 伺服器。

相關資源

AWS 文件

- Application Load Balancer
- 使用公共託管區域
- 使用私有託管區域

甲骨文文檔：

- Oracle WebLogic 伺服器代理外掛程式的概要
- 使用基礎結構安裝程式安裝 WebLogic 伺
- 安裝和配置甲骨文 HTTP 服務器

透過私人網路 Connect 至應用程式移轉服務資料和控制平面

由迪平耆那教（AWS）和邁克·庫茲涅佐夫（AWS）創建

環境：PoC 或試點	技術：網路；移轉	AWS 服務：AWS 應用程式遷移服務；Amazon EC2；Amazon VPC；Amazon S3

Summary

此模式說明如何使用界面 VPC 端點連接到私有安全網路上的 AWS 應用程式遷移服務 (AWS MGN) 資料平面和控制平面。

應用程式遷移服務是一種高度自動化 lift-and-shift (重新託管) 解決方案，可簡化、加速並降低將應用程式遷移到 AWS 的成本。它使公司能夠重新託管大量物理，虛擬或雲端服務器，而不會出現兼容性問題，性能中斷或長時間切換窗口。應用程式遷移服務可從 AWS 管理主控台取得。這可與其他 AWS 服務 (例如 AWS CloudTrail CloudWatch、Amazon 和 AWS Identity and Access Management (IAM) 完美整合。

您可以使用 AWS VPN 服務、AWS Direct Connect 或應用程式遷移服務中的 VPC 對等，透過私有連線，從來源資料中心連線到資料計劃 (亦即做為目的地 VPC 中資料複寫暫存區的子網路)。您也可以使用 AWS 支援的介面 VPC 端點，透過私有網路連接 PrivateLink 到應用程式遷移服務控制平面。

先決條件和限制

先決條件

- 暫存區域子網路 — 在設定應用程式遷移服務之前，請先建立一個子網路，用作從來源伺服器複寫到 AWS (亦即資料平面) 的資料暫存區。首次存取應用程式移轉服務主控台時，必須在 [複寫設定] 範本中指定此子網路。您可以在「複製設定」範本中覆寫特定來源伺服器的此子網路。雖然您可以在 AWS 帳戶中使用現有的子網路，但我們建議您為此目的建立新的專用子網路。
- 網路需求 — 應用程式遷移服務在您的暫存區子網路中啟動的複寫伺服器必須能夠將資料傳送到應用程式遷移服務 API 端點`https://mgn.<region>.amazonaws.com/`，其中`<region>`是您要複寫到的 AWS 區域的程式碼 (例如`https://mgn.us-east-1.amazonaws.com`)。下載應用程式遷移服務軟體時，必須使用 Amazon 簡易儲存服務 (Amazon S3) 服務 URL。
  - AWS 複寫代理程式安裝程式應該可以存取您與應用程式遷移服務搭配使用的 AWS 區域的 S3 儲存貯體 URL。

- 暫存區域子網路應該可以存取 Amazon S3。
- 安裝 AWS 複寫代理程式的來源伺服器必須能夠將資料傳送到暫存區子網路中的複寫伺服器，以及位於的應用程式遷移服務 API 端點`https://mgn.<region>.amazonaws.com/`。

下表列出所需的連接埠。

來源	目的地	連接埠	如需詳細資訊，請參閱
來源資料中心	Amazon S3 服務網址	443 (TCP)	透過 TCP 連接埠 443 進行通訊
來源資料中心	應用程式遷移服務的 AWS 區域特定主控台地址	443 (TCP)	來源伺服器與應用程式移轉服務 (透過 TCP 連接埠 443) 之間的通訊
來源資料中心	暫存區域子網路	一千五百	透過 TCP 連接埠 1500 的來源伺服器與暫存區域子網路之間的通訊
暫存區域子網路	應用程式遷移服務的 AWS 區域特定主控台地址	443 (TCP)	透過 TCP 連接埠 443 的暫存區域子網路與應用程式移轉服務之間的通訊
暫存區域子網路	Amazon S3 服務網址	443 (TCP)	透過 TCP 連接埠 443 進行通訊
暫存區域子網路	子網路 AWS 區域的 Amazon EC2 端點	443 (TCP)	透過 TCP 連接埠 443 進行通訊

限制

目前並非所有 AWS 區域和作業系統都提供應用程式遷移服務。

- 支援的 AWS 區域

- 支援的作業系統

架構

下圖說明典型移轉的網路架構。如需有關此架構的詳細資訊，請參閱應用程式遷移服務說明文件和應用程式移轉服務服務架構和網路架構影片。

以下詳細檢視顯示暫存區 VPC 中用於連接 Amazon S3 和應用程式遷移服務的介面 VPC 端點的組態。

工具

- AWS 應用程式遷移服務是一項 AWS 服務，可簡化、加速並降低在 AWS 上重新託管應用程式的成本。
- 界面 VPC 端點可讓您連接到採用 AWS 提供支援的服務， PrivateLink 而不需要網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，即可與服務中的資源通訊。VPC 與另一個服務之間的流量都會保持在 Amazon 網路的範圍內。

史诗

為應用程式遷移服務、Amazon EC2 和 Amazon S3 建立端點

任務	描述	所需技能
設定應用程式移轉服務的介面端點。	來源資料中心和暫存區 VPC 會透過您在目標暫存區 VPC 中建立的介面端點，以私密方式連線到應用程式移轉服務控制平面。若要建立端點： 1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。	遷移, 領導

任務	描述	所需技能
	2. 在導覽窗格中，選擇 Endpoints (端點)，Create Endpoint (建立端點)。 3. 在 Service category (服務類別) 中，選擇 AWS services (AWS 服務)。 4. 在「服務名稱」中，輸入`com.amazonaws.<region>.mgn` 。選擇「介面」做為「類型」。 5. 對於 VPC，請選取目標暫存區 VPC 以建立端點。 6. 針對 Subnets (子網路)，選取要在其中建立端點網路界面的子網路 (可用區域)。 7. 若要開啟介面端點的私人 DNS，請在 [其他設定] 區段中，選取 [啟用 DNS 名稱]。 8. 選取允許透過 TCP 443 從暫存區 VPC 子網路輸入的安全群組。 9. 選擇 建立端點。 如需詳細資訊，請參閱 Amazon VPC 說明文件中的介面虛擬私人雲端端點。	

任務	描述	所需技能
設定 Amazon EC2 的界面端點。	暫存區域 VPC 會透過您在目標暫存區 VPC 中建立的介面端點私有連線至 Amazon EC2 API。若要建立端點，請遵循上一個故事中提供的指示。 • 對於服務名稱，請輸入`com.amazonaws.<region>.ec2` 。選擇「介面」做為「類型」。 • 安全性群組必須允許透過連接埠 443 來自暫存區 VPC 子網路的輸入 HTTPS 流量。 • 在 [其他設定] 區段中，選取 [啟用 DNS 名稱]。	遷移, 領導

任務	描述	所需技能
設定 Amazon S3 的界面端點。	來源資料中心和暫存區域 VPC 會透過您在目標暫存區 VPC 中建立的介面端點私有連線到 Amazon S3 API。要創建端點，請按照第一個故事中提供的說明進行操作。 • 在「服務名稱」中，輸入`com.amazonaws.<region>.s3` 。選擇「介面」做為「類型」。 • VPC 安全性群組必須允許透過連接埠 443 來自暫存區 VPC 子網路的輸入 HTTPS 流量。 • 在「其他設定」區段中，清除「啟用 DNS 名稱」。Amazon S3 界面端點不支援私有 DNS 名稱。 備註：您使用介面端點，因為閘道端點連線無法從 VPC 外延伸。如需詳細資訊，請參閱 Amazon VPC 文件。)	遷移, 領導

任務	描述	所需技能
設定 Amazon S3 閘道端點。	在組態階段，複寫伺服器必須連接到 S3 儲存貯體，才能下載 AWS 複寫伺服器的軟體更新。但是，Amazon S3 界面端點不支援私有 DNS 名稱，也無法向複寫伺服器提供 Amazon S3 端點 DNS 名稱。 為了緩解此問題，您可以在暫存區子網路所屬的 VPC 中建立 Amazon S3 閘道端點，並使用相關路由更新暫存子網路的路由表。如需詳細資訊，請參閱 AWS PrivateLink 文件中的建立閘道端點。	雲端管理員
設定內部部署 DNS 以解析端點的私人 DNS 名稱。	應用程式遷移服務和 Amazon EC2 的介面端點具有可在 VPC 中解析的私有 DNS 名稱。不過，您也需要設定內部部署伺服器，以解析這些介面端點的私人 DNS 名稱。 有多種方法可以配置這些服務器。在此模式中，我們透過將現場部署 DNS 查詢轉送到暫存區域 VPC 中的 Amazon Route 53 Resolver 入站端點來測試此功能。如需詳細資訊，請參閱 Route 53 說明文件中的解決 VPC 與您網路之間的 DNS 查詢。	移民工程師

透過私有連結連線至應用程式移轉服務控制平面

任務	描述	所需技能
使用 AWS 安裝 AWS 複寫代理程式 PrivateLink。	1. 將 AWS 複寫代理程式下載到目的地區域中的私有 S3 儲存貯體。 2. 登入要移轉的來源伺服器。AWS 複寫代理程式安裝程式需要對應用程式遷移服務和 Amazon S3 端點進行網路存取。由於您的現場部署網路不開放給應用程式遷移服務和 Amazon S3 公有端點，因此您必須使用 AWS 在先前步驟中建立的介面端點協助下安裝代理程式 PrivateLink。 以下是 Linux 的一個例子： 1. 使用下列命令下載代理程式： `wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-<aws_region>.bucket.<s3-endpoint-DNS-name>/latest/linux/aws-replication-installer-init.py` 附註：這bucket是一個靜態關鍵字，您必須在 Amazon S3 介面端點 DNS 名稱之前新增該	移民工程師

任務	描述	所需技能
	關鍵字。如需詳細資訊，請參閱 Amazon S3 說明文件。 例如，如果 Amazon S3 界面端點的 DNS 名稱是，vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com 而 AWS 區域為us-west-1 ，您可以使用以下命令： `wget -O ./aws-replication-installer-init.py \ https://aws-application-migration-service-us-west-1.bucket.vpce-009c8b07adb052a11-qgf8q50y.s3.us-west-1.vpce.amazonaws.com/latest/linux/aws-replication-installer-init.py` 2. 安裝代理程式： • 如果您在為應用程式移轉服務建立介面端點時選取了啟用 DNS 名稱，請執行以下命令： `sudo python3 aws-replication-installer-init.py \`	

任務	描述	所需技能
	<pre>--region <aws_region> \ --aws-access-key-id <access-key> \ --aws-secret-access-key <secret-key> \ --no-prompt \ --s3-endpoint <s3-endpoint-DNS-name></pre> • 如果您在建立應用程式遷移服務的介面端點時未選取 [啟用 DNS 名稱]，請執行以下命令： <pre>sudo python3 aws-replication-installer-init.py \ --region <aws_region> \ --aws-access-key-id <access-key> \ --aws-secret-access-key <secret-key> \ --no-prompt \ --s3-endpoint <s3-endpoint-DNS-name> \ --endpoint <mgn-endpoint-DNS-name></pre> 如需詳細資訊，請參閱應用程式遷移服務文件中的 AWS 複寫代理程式安裝說明。 建立與應用程式遷移服務的連線並安裝 AWS 複寫代理程式後，請遵循應用程式遷移服務	

任務	描述	所需技能
	文件中的指示，將來源伺服器遷移到目標 VPC 和子網路。	

相關資源

應用程式移轉服務說

- 概念
- 移轉工作流
- 快速入門指南
- 常見問答集
- 疑難排解

其他資源

- AWS 應用程式遷移服務-技術簡介 (AWS Training and Certification 逐步解說)
- AWS 應用程式遷移服務架構和網路架構 (影片)

其他資訊

疑難排解 Linux 伺服器上 AWS 複寫代理程式安

如果您在 Amazon Linux 伺服器上出現 gcc 錯誤，請設定套件儲存庫並使用下列命令：

```
## sudo yum groupinstall "Development Tools"
```

使用 AWS CloudFormation 自訂資源和 Amazon SNS 建立 Infoblox 物件

創建者蒂姆·薩頓 (AWS)

環境：PoC 或試點	技術：網路	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS CloudFormation；AWS KMS；AWS Lambda；AWS Organizations		

Summary

Infoblox 網域名稱系統 (DNS)、動態主機設定通訊協定 (DHCP) 和 IP 位址管理 (Infoblox DDI) 可讓您集中並有效地控制複雜的混合式環境。借助 Infoblox DDI，除了使用相同的設備管理內部部署和 Amazon Web Services（AWS）雲端上的 DNS 之外，您還可以在一個權威 IP 地址管理（IPAM）數據庫中發現和記錄所有網絡資產。

此模式說明如何使用 AWS CloudFormation 自訂資源來建立資源物件 (例如 DNS 記錄或 IPAM 物件)，方法是呼叫 Infoblox WAPI API。如需有關 WAPI 的詳細資訊，請參閱資訊集文件中的 WAPI 文件。

使用此模式的方法，除了移除建立記錄和佈建網路的手動程序之外，您還可以取得 AWS 和現場部署環境的 DNS 記錄和 IPAM 組態的統一檢視。您可以在以下用例中使用此模式的方法：

- 在建立亞馬遜彈性運算雲端 (Amazon EC2) 執行個體後新增 A 記錄
- 建立 Application Load Balancer 後新增 CNAME 記錄
- 建立虛擬私有雲 (VPC) 後新增網路物件
- 提供下一個網路範圍，並使用該範圍建立子網路

您還可以擴展此模式並使用其他 Infoblox 設備功能，例如添加不同的 DNS 記錄類型或配置 Infoblox vDiscovery。

該模式使用的 hub-and-spoke 設計中樞需要連線到 AWS 雲端或內部部署上的 Infoblox 設備，並使用 AWS Lambda 呼叫 Infoblox API。支點位於 AWS Organizations 中相同組織中的相同帳戶或不同帳戶，並使用 AWS CloudFormation 自訂資源呼叫 Lambda 函數。

先決條件和限制

先決條件

- 安裝在 AWS 雲端、內部部署或兩者上的現有 Infoblox 設備或網格，並使用可管理 IPAM 和 DNS 動作的管理員使用者進行設定。如需這方面的詳細資訊，請參閱 Infoblox 文件中的關於管理員帳戶。
- 您要在 Infoblox 設備上新增記錄的現有 DNS 授權區域。如需有關此功能的詳細資訊，請參閱 Infoblox 文件中的配置授權區域。
- AWS 組織中有兩個作用中的 AWS 帳戶。一個帳戶是 Hub 帳戶，另一個帳戶是支點帳戶。
- 中樞和支點帳戶必須位於相同的 AWS 區域。
- 中樞帳戶的 VPC 必須連接到 Infoblox 設備；例如，使用 AWS Transit Gateway 或 VPC 對等互連。
- AWS Serverless Application Model (AWS SAM)，透過 AWS Cloud9 或 AWS 在本機安裝和設定。CloudShell
- `Infoblox-Hub.zip`和`ClientTest.yaml`檔案 (附加)，下載到包含 AWS SAM 的本機環境。

限制

- AWS CloudFormation 自訂資源的服務權杖必須來自建立堆疊的相同區域。我們建議您在每個區域使用中樞帳戶，而不是在一個區域中建立 Amazon Simple Notification Service (Amazon SNS) 主題，然後在另一個區域呼叫 Lambda 函數。

產品版本

- 資訊布盧克斯 WAPI 2.7 版本

架構

下圖顯示此模式的工作流程。

此圖表顯示此模式解決方案的下列元件：

1. AWS CloudFormation 自訂資源可讓您在建立、更新或刪除堆疊時，在 AWS CloudFormation 執行的範本中撰寫自訂佈建邏輯。當您建立堆疊時，AWS CloudFormation 會將`create`請求傳送至由 EC2 執行個體上執行的應用程式監控的 SNS 主題。

2. 來 CloudFormation 自 AWS 自訂資源的 Amazon SNS 通知會透過特定的 AWS Key Management Service (AWS KMS) 金鑰加密，而且只能存取組織中組 Organizations 中的帳戶。SNS 主題會啟動呼叫資訊集 WAPI API 的 Lambda 資源。
3. Amazon SNS 調用以下 Lambda 函數，這些函數採用信息塊 WAPI URL，用戶名和密碼 AWS Secrets Manager Amazon 資源名稱（ARN）作為環境變量：
   - `dnsapi.lambda_handler`— 從 AWS CloudFormation 自訂資源接收`DNSName`、`DNSType`和`DNSValue`值，並使用這些資源建立 DNS A 記錄和 CNAME。
   - `ipaddr.lambda_handler`— 從 AWS CloudFormation 自訂資源接收`VPCCIDRType`、`SubnetPrefix`和`Network Name`值，並使用這些資源將網路資料新增到 Infoblox IPAM 資料庫，或為自訂資源提供下一個可用的網路，以建立新子網路。
   - `describeprefixes.lambda_handler`— 使用`"com.amazonaws."+Region+".s3"`篩選器呼叫 `describe_managed_prefix_lists` AWS API 以擷取所需的項目`prefix ID`。

   重要事項：這些 Lambda 函數是用 Python 編寫的，彼此相似，但會呼叫不同的 API。
4. 您可以將 Infoblox 網格部署為實體、虛擬或雲端式網路設備。 它可以部署在內部部署，也可以使用各種虛擬機器管理程式 (包括 VMware ESXi、Microsoft 超 V、Linux KVM 和 Xen) 部署為虛擬應用裝置。您也可以使用 Amazon 機器映像 (AMI) 在 AWS 雲端上部署 Infoblox 網格。
5. 此圖顯示了 Infoblox 網格的混合式解決方案，該解決方案可為 AWS 雲端和內部部署上的資源提供 DNS 和 IPAM。

技術堆疊

- AWS CloudFormation
- IAM
- AWS KMS
- AWS Lambda
- AWS SAM
- AWS Secrets Manager
- Amazon SNS
- Amazon VPC

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- AWS Serverless Application Model (AWS SAM) 是一種開放原始碼架構，可協助您在 AWS 雲端建置無伺服器應用程式。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

Code

您可以使用範`ClientTest.yaml`例 AWS CloudFormation 範本 (隨附) 來測試 Infoblox 集線器。您可以自訂 AWS CloudFormation 範本，以包含下表中的自訂資源。

使用支點自訂資源建立 A 記錄	返回值： `infobloxref` — 資訊集參考 示例資源： `ARECORDCustomResource:` `  Type: "Custom::InfobloxAPI"`

```
  Properties:

    ServiceToken: !Sub  arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfobloxDNSFunction

    DNSName: 'arecordtest.compa
ny.com'

    DNSType: 'ARecord'

    DNSValue: '10.0.0.1'
```

使用支點自訂資源建立 CNAME 記錄

返回值：

`infobloxref` — 資訊集參考

示例資源：

```
CNAMECustomResource:

  Type: "Custom::InfobloxAPI"

  Properties:

    ServiceToken: !Sub arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfoblox

    DNSFunction

    DNSName: 'cnametest.company.com'

    DNSType: 'cname'

    DNSValue: 'aws.amazon.com'
```

使用 Infoblox 網輻自訂資源建立網路物件

返回值：

infobloxref — 資訊集參考

network— 網絡範圍（相同VPCCIDR）

示例資源：

```
VPCCustomResource:

  Type: 'Custom::InfobloxAPI'

  Properties:

    ServiceToken: !Sub  arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfobloxNextSubnetFunction

    VPCCIDR: !Ref VpcCIDR

    Type: VPC

    NetworkName: My-VPC
```

使用 Infoblox 網輻自訂資源擷取下一個可用的子網路

返回值：

`infobloxref` — 資訊集參考

`network` — 子網路的網路範圍

示例資源：

```
Subnet1CustomResource:

  Type: 'Custom::InfobloxAPI'

  DependsOn: VPCCustomResource

  Properties:

    ServiceToken: !Sub  arn:aws:sns:
${AWS::Region}:${HubAccountID}:Ru
nInfobloxNextSubnetFunction

    VPCCIDR: !Ref VpcCIDR

    Type: Subnet

    SubnetPrefix: !Ref SubnetPrefix

NetworkName: My-Subnet
```

史诗

建立和設定中樞帳戶的 VPC

任務	描述	所需技能
建立具有連線至 Infoblox 應用裝置的 VPC 擬私人雲端。	登入您的中樞帳戶的 AWS 管理主控台，然後按照 AWS 快速入門 AWS 雲端快速入門參考部署上 Amazon VPC 中的步驟建立 VPC。	網路管理員、系統管理員

任務	描述	所需技能
	重要：VPC 必須具有 HTTPS 連線至 Infoblox 應用裝置，我們建議您為此連線使用私有子網路。	
(選擇性) 為私有子網路建立 VPC 端點。	VPC 端點可為您的私有子網路提供公用服務的連線。需要下列端點： • Amazon Simple Storage Service (Amazon S3) 的閘道端點，可讓 Lambda 與 AWS 通訊 CloudFormation • 密碼管理員的介面端點，可啟用與密 Secrets Manager 的連線能力 • AWS KMS 的介面端點，可允許加密 SNS 主題和機密 Secrets Manager 碼 如需為私有子網路建立端點的詳細資訊，請參閱 Amazon VPC 說明文件中的 VPC 端點。	網路管理員、系統管理員

部署資訊集線器

任務	描述	所需技能
建立 AWS SAM 範本。	1. 在包含 AWS SAM 的環境中執行`unzip Infoblox-Hub.zip` 命令。 2. 執行命`cd Hub/`令，將您的目錄變更為`Hub`目錄。	系統管理員開發人員

任務	描述	所需技能
	3. 執行命sam build令以處理 AWS SAM 範本檔案、應用程式程式碼，以及任何特定語言的檔案和相依性。此指sam build令也會以下列內文所預期的格式和位置複製組建加工品。	

任務	描述	所需技能
部署 AWS SAM 範本。	該`sam deploy`命令採用必要的參數並將其儲存到`samconfig.toml` 檔案中，將 AWS CloudFormation 範本和 Lambda 函數存放在 S3 儲存貯體中，然後將 AWS CloudFormation 範本部署到您的中樞帳戶。 下列範例程式碼示範如何部署 AWS SAM 範本： `$ sam deploy --guided` `Configuring SAM deploy` `==========` `Looking for config file [samconfig.toml] : Found` `Reading default arguments : Success` `Setting default arguments for 'sam deploy'` `=========` `Stack Name [Infoblox-Hub]:` `AWS Region [eu-west-1]:` `Parameter InfobloxUsername:` `Parameter InfobloxPassword:` `Parameter InfobloxIPAddress [xxx.xxx.xx.xxx]:`	系統管理員開發人員

任務	描述	所需技能
	Parameter AWSOrganisationID [o-xxxxxxxxx]: Parameter VPCID [vpc-xxxxxxxxx]: Parameter VPCCIDR [xxx.xxx.xxx.xxx/16]: Parameter VPCSubnetID1 [subnet-xxx]: Parameter VPCSubnetID2 [subnet-xxx]: Parameter VPCSubnetID3 [subnet-xxx]: Parameter VPCSubnetID4 []: #Shows you resources changes to be deployed and require a 'Y' to initiate deploy Confirm changes before deploy [Y/n]: y #SAM needs permission to be able to create roles to connect to the resources in your template Allow SAM CLI IAM role creation [Y/n]: n Capabilities [['CAPABILITY_NAMED_IAM']]: Save arguments to configuration file [Y/n]: y SAM configuration file [samconfig.toml]:	

任務	描述	所需技能
	`SAM configuration environment [default]:` 重要事項：您每次都必須使用該`--guided`選項，因為 Infoblox 登錄憑據不存儲在文件中`samconfig.toml` 。	

相關資源

- 使用郵差開始使用 WAPI（信息博客）
- 使用 BYOL 模型為 AWS 佈建 VNIO (資訊集文件)
- quickstart-aws-vpc（GitHub 回購）
- 前綴列表（適用於 Python 文檔的 AWS 開發套件）

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

自訂 AWS Network Firewall 的 Amazon CloudWatch 提醒

由傑森·歐文斯創建 (AWS)

環境：PoC 或試點	技術：網路；安全性、身分識別、合規	工作負載：開源
AWS 服務：Amazon CloudWatch 日誌；AWS Network Firewall；AWS CLI		

Summary

該模式可幫助您自定義 Amazon 網絡服務（AWS）Network Firewall 生成的亞馬遜 CloudWatch 提醒。您可以使用預先定義的規則或建立自訂規則，以決定警示的訊息、中繼資料和嚴重性。然後，您可以根據這些提醒採取行動，或者由其他 Amazon 服務（例如 Amazon）自動執行響應 EventBridge。

在此病毒碼中，您會產生與 Suricata 相容的防火牆規則。Suricata 是一個開放原始碼的威脅偵測引擎。您首先建立簡單的規則，然後對其進行測試，以確認已產生並記錄 CloudWatch 警示。成功測試規則後，您可以修改它們以定義自訂訊息、中繼資料和嚴重性，然後再測試一次以確認更新。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在您的 Linux、macOS 或視窗工作站上安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI。
- 安裝和設定為使用 CloudWatch 日誌的 AWS Network Firewall。如需詳細資訊，請參閱記錄來自 AWS Network Firewall 的網路流量。
- 位於受 Network Firewall 保護的虛擬私有雲 (VPC) 私有子網路中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。

產品版本

- 對於 AWS CLI 的第 1 版，請使用 1.18.180 或更新版本。對於 AWS CLI 的第 2 版，請使用 2.1.2 或更新版本。
- 來自蘇里卡塔 5.0.2 版的分類 .config 文件。如需此組態檔案的副本，請參閱其他資訊一節。

架構

目標技術堆疊

- Network Firewall
- Amazon CloudWatch 日誌

目標架構

架構圖顯示下列工作流程：

1. 私有子網路中的 EC2 執行個體會使用 curl 或 Wget 提出要求。
2. Network Firewall 會處理流量並產生警示。
3. Network Firewall 會將記錄的警示傳送至 CloudWatch 記錄檔。

工具

AWS 服務

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Network Firewall 是適用於 AWS 雲端中虛擬私有雲 (VPC) 的可設定狀態、受管網路防火牆以及入侵偵測和防護服務。

其他工具和服務

- curl-curl 是一個開源的命令行工具和庫。

- Wget — GNU Wget 是一個免費的命令行工具。

史诗

建立防火牆規則和規則群組

任務	描述	所需技能
建立規則。	1. 在文字編輯器中，建立要新增至防火牆的規則清單。每個規則必須位於單獨的行上。`classtype` 參數中的值來自預設的 Suricata 分類組態檔案。如需完整的組態檔案內容，請參閱「其他資訊」一節。以下是兩個規則範例。 `alert http any any -> any any (content:"badstuff"; classtype:misc-activity; sid:3; rev:1;)` `alert http any any -> any any (content:"morebadstuff"; classtype:bad-unknown; sid:4; rev:1;)` 2. 將規則儲存在名為的檔案中`custom.rules` 。	AWS 系統管理員、網路管理員
建立規則群組。	在 AWS CLI 中，輸入以下命令。這會建立規則群組。 `# aws network-firewall create-rule-group \`	AWS 系統管理員

任務	描述	所需技能
	--rule-group-name custom --type STATEFUL \ --capacity 10 --rules file://custom.rules \ --tags Key=environment,Value=development	
	以下為範例輸出。請記下RuleGroupArn ，您需要在稍後的步驟中。	
	{ "UpdateToken": "4f998d72-973c-490a-bed2-fc3460547e23", "RuleGroupResponse": { "RuleGroupArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom", "RuleGroupName": "custom", "RuleGroupId": "238a8259-9eaf-48bb-90af-5e690cf8c48b", "Type": "STATEFUL", "Capacity": 10, "RuleGroupStatus": "ACTIVE", "Tags": [{ "Key": "environment", "Value": "development"	

任務	描述	所需技能
	<pre> }] }</pre>	

更新防火牆策略

任務	描述	所需技能
取得防火牆策略的 ARN。	在 AWS CLI 中，輸入以下命令。這會傳回防火牆政策的 Amazon 資源名稱 (ARN)。記錄 ARN 以便稍後在此模式中使用。 <pre># aws network-firewall describe-firewall \ --firewall-name aws-network-firewall- anfw \ --query 'Firewall .FirewallPolicyArn'</pre> 以下是此命令所傳回的範例 ARN。 <pre>"arn:aws:network-f irewall:us-east-2: 1234567890:firewal l-policy/firewall- policy-anfw"</pre>	AWS 系統管理員
更新防火牆策略。	在文本編輯器中，複製粘貼以下代碼。替換<RuleGroupArn> 為您在上一個史詩中記錄的數值。儲存檔案為	AWS 系統管理員

任務	描述	所需技能
	firewall-policy-anfw.json 。 `{ "StatelessDefaultActions": [ "aws:forward_to_sfe" ], "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ], "StatefulRuleGroupReferences": [ { "ResourceArn": "<RuleGroupArn>" } ] }` 在 AWS CLI 中輸入以下命令。此命令需要更新令牌才能添加新規則。權杖用於確認自上次擷取原則後並未變更。 `UPDATETOKEN=(`aws network-firewall describe-firewall-policy \ --firewall-policy-name firewall-policy-anfw \ --output text --query UpdateToken`)	

任務	描述	所需技能
	aws network-firewall update-firewall-policy \ --update-token $UPDATETOKEN \ --firewall-policy-name firewall-policy-anfw \ --firewall-policy file://firewall-policy-anfw.json	

任務	描述	所需技能
確認原則更新。	(選擇性) 如果您想確認已新增規則並檢視政策格式，請在 AWS CLI 中輸入以下命令。 `# aws network-firewall describe-firewall-policy \ --firewall-policy-name firewall-policy-anfw \ --query FirewallPolicy` 以下為範例輸出。 `{ "StatelessDefaultActions": [ "aws:forward_to_sfe" ], "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ], "StatefulRuleGroupReferences": [ { "ResourceArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom" } ] }`	AWS 系統管理員

測試警示功能

任務	描述	所需技能
產生用於測試的警示。	1. 登入防火牆子網路內的測試工作站。 2. 輸入應產生警示的命令。例如，您可以使用wget或curl。 `wget -U "badstuff" http://www.amazon.com -o /dev/null` `curl -A "morebadstuff" http://www.amazon.com -o /dev/null`	AWS 系統管理員
驗證是否已記錄警示。	1. 請在以下位置開啟 CloudWatch 主控台 https://console.aws.amazon.com/cloudwatch/ 2. 導覽至正確的記錄群組和串流。如需詳細資訊，請參閱檢視傳送至 CloudWatch 記錄檔的記錄檔資料 (CloudWatch 記錄檔文件)。 3. 確認記錄的事件與下列範例類似。這些範例僅顯示警示的相關部分。 範例 1 `"alert": { "action": "allowed",`	AWS 系統管理員

任務	描述	所需技能
	<pre>"signatur e_id": 3, "rev": 1, "signatur e": "", "category ": "Misc activity", "severity ": 3 }</pre> 範例 2 <pre>"alert": { "action": "allowed", "signatur e_id": 4, "rev": 1, "signatur e": "", "category ": "Potentially Bad Traffic", "severity ": 2 }</pre>	

更新防火牆規則和規則群組

任務	描述	所需技能
更新防火牆規則。	1. 在文字編輯器中，開啟 `custom.rules` 檔案。 2. 將第一個規則變更為類似下列內容。此規則必須在檔案中的單行上輸入。	AWS 系統管理員

任務	描述	所需技能
	<pre>alert http any any -> any any (msg:"Watch out - Bad Stuff!!"; content:"badstuff"; classtype:misc-activity; priority:2; sid:3; rev:2; metadata:custom-field-2 Danger!, custom-field More Info;)</pre> 這會對規則進行下列變更： • 添加一個 msg（Suricata 網站）字符串，該字符串提供有關簽名或警報的文本信息。在產生的警示中，這會對應至簽章。 • 將預設優先順序（蘇里卡塔網站）misc-activity 從 3 調整為 2。如需各種項目的預設值classtypes ，請參閱其他資訊一節。 • 將自訂中繼資料 (Suricata 網站) 新增至警示。這是新增至簽名的其他資訊。建議您使用鍵值配對。 • 將調整版本（蘇里卡塔網站）從 1 更改為 2。這代表簽名的版本。	

任務	描述	所需技能
更新規則群組。	在 AWS CLI 中，執行下列命令。使用防火牆策略的 ARN。這些命令會取得更新 Token，並使用規則變更來更新規則群組。 `# UPDATETOKEN=(`aws network-firewall \ describe-rule-group \ --rule-group-arn arn:aws:network-firewall:us-east-2:123457890:stateful-rulegroup/custom \ --output text --query UpdateToken`)` `# aws network-firewall update-rule-group \ --rule-group-arn arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom \ --rules file://custom.rules \ --update-token $UPDATETOKEN` 以下為範例輸出。 `{ "UpdateToken": "7536939f-6a1d-414c-96d1-bb28110996ed", "RuleGroupResponse": {`	AWS 系統管理員

任務	描述	所需技能
	"RuleGroupArn": "arn:aws:network-firewall:us-east-2:1234567890:stateful-rulegroup/custom", "RuleGroupName": "custom", "RuleGroupId": "238a8259-9eaf-48bb-90af-5e690cf8c48b", "Type": "STATEFUL", "Capacity": 10, "RuleGroupStatus": "ACTIVE", "Tags": [{ "Key": "environment", "Value": "development" }] } }	

測試更新的警示功能

任務	描述	所需技能
產生測試的警示。	1. 登入防火牆子網路內的測試工作站。 2. 輸入應產生警示的指令。例如，您可以使用 `curl`.	AWS 系統管理員

任務	描述	所需技能
	`curl -A "badstuff" http://www.amazon.com -o /dev/null`	

任務	描述	所需技能
驗證已變更的警示。	1. 請在以下位置開啟 CloudWatch 主控台 https://console.aws.amazon.com/cloudwatch/ 2. 導覽至正確的記錄群組和串流。 3. 確認記錄的事件與下列範例類似。此範例僅顯示警示的相關部分。 `"alert": { "action": "allowed", "signature_id": 3, "rev": 2, "signature": "Watch out - Bad Stuff!!", "category": "Misc activity", "severity": 2, "metadata": { "custom-field": [ "More Info" ], "custom-field-2": [ "Danger!" ] } }`	AWS 系統管理員

相關資源

參考

- 將提醒從 AWS Network Firewall 傳送到 Slack 通道 (AWS Prescriptive Guidance)
- 使用 Suricata 在 AWS 上擴展威脅防護 (AWS 部落格文章)
- AWS Network Firewall 的部署模型 (AWS 部落格文章)
- 蘇里卡塔元鍵工程（蘇里卡塔文檔）

教學課程和影片

- AWS Network Firewall 研討會

其他資訊

以下是來自蘇里卡塔 5.0.2 的分類配置文件。建立防火牆規則時會使用這些分類。

```
# config classification:shortname,short description,priority

config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username
 was detected,2
```

```
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual
 port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or
 event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web
 application,2
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: inappropriate-content,Inappropriate Content was Detected,1
config classification: policy-violation,Potential Corporate Privacy Violation,1
config classification: default-login-attempt,Attempt to login by a default username and
 password,2

# Update
config classification: targeted-activity,Targeted Malicious Activity was Detected,1
config classification: exploit-kit,Exploit Kit Activity Detected,1
config classification: external-ip-check,Device Retrieving External IP Address
 Detected,2
config classification: domain-c2,Domain Observed Used for C2 Detected,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: coin-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity
 Detected,1
```

將 DNS 記錄批量遷移到 Amazon Route 53 私有託管區域

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：網路、雲端原生、基礎架 DevOps構	AWS 服務：AWS Cloud9；Amazon Route 53；Amazon S3

Summary

網路工程師和雲端管理員需要有效且簡單的方法，將網域名稱系統 (DNS) 記錄新增至 Amazon Route 53 中的私有託管區域。使用手動方法將項目從 Microsoft Excel 工作表複製到 Route 53 主控台中的適當位置是乏味且容易出錯的。這種模式描述了一種自動化方法，可以減少添加多個記錄所需的時間和精力。它還為多個託管區域建立提供了一組可重複的步驟。

此模式使用 AWS Cloud9 整合式開發環境 (IDE) 進行開發和測試，並使用亞馬遜簡單儲存服務 (Amazon S3) 存放記錄。為了有效地處理資料，模式會使用 JSON 格式，因為它的簡單性和支援 Python 字典 (`dict`資料類型) 的能力。

注意：如果您可以從系統產生區域檔案，請考慮改用 Route 53 匯入功能。

先決條件和限制

先決條件

- 包含私人託管區域記錄的 Excel 工作表
- 熟悉不同類型的 DNS 記錄，例如 A 記錄、名稱授權指標 (NAPTR) 記錄和 SRV 記錄 (請參閱支援的 DNS 記錄類型)
- 熟悉 Python 語言及其庫

限制

- 該模式不會為所有使用案例案例提供廣泛的涵蓋範圍。例如，調用不會使用 API 的所有可用屬性。
- 在 Excel 工作表中，每行中的值被假定為唯一。每個完整網域名稱 (FQDN) 的多個值預期會出現在同一列中。如果不是這樣，您應該修改此模式中提供的代碼以執行必要的串聯。

- 該模式使用適用於 Python 的 AWS 開發套件 (Boto3) 直接呼叫 Route 53 服務。您可以增強程式碼以針對`create_stack`和`update_stack`命令使用 AWS CloudFormation 包裝函式，並使用 JSON 值填入範本資源。

架構

技術堆疊

- Route 53 私人託管區域，用於路由流量
- 適用於開發和測試的 AWS Cloud9 IDE
- Amazon S3 用於存儲輸出 JSON 文件

工作流程包含這些步驟，如上圖所示，並在 Epics 一節中討論：

1. 將具有記錄集資訊的 Excel 工作表上傳至 S3 儲存貯體。
2. 建立並執行將 Excel 資料轉換為 JSON 格式的 Python 指令碼。
3. 從 S3 儲存貯體讀取記錄並清除資料。
4. 在您的私人託管區域中建立記錄集。

工具

- 路線 53 — Amazon Route 53 是高度可用且可擴展的 DNS 網路服務，可處理網域註冊、DNS 路由和運作狀態檢查。
- AWS Cloud9 — AWS Cloud9 是一種 IDE，提供豐富的程式碼編輯體驗，並支援多種程式設計語言和執行階段除錯器，以及內建終端機。其中包含用於編碼、建置、執行、測試、除錯以及在雲端中發行軟體的工具集合。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。

史诗

為自動化準備資料

任務	描述	所需技能
為您的記錄創建一個 Excel 文件。	使用您從目前系統匯出的記錄，建立包含記錄所需欄的 Excel 工作表，例如完整網域名稱 (FQDN)、記錄類型、存留時間 (TTL) 和值。對於 NAPTR 和 SRV 記錄，值是多個屬性的組合，因此請使用 Excel 的`concat`方法來組合這些屬性。 FqdnN Recor e 值 TTL 一些. 示例 A 1.1.1.1 900	數據工程師，卓越技能
確認工作環境。	在 AWS Cloud9 IDE 中，建立一個 Python 檔案，將 Excel 輸入工作表轉換為 JSON 格式。您也可以使用 Amazon SageMaker 筆記本來處理 Python 代碼，而不是 AWS Cloud9。) 確認您使用的 Python 版本是 3.7 或更新版本。 `python3 --version` 安裝熊貓包。	一般 AWS

任務	描述	所需技能
	`pip3 install pandas --user`	
將 Excel 工作表資料轉換為 JSON。	創建一個 Python 文件，其中包含以下代碼以從 Excel 轉換為 JSON。 `import pandas as pd` `data=pd.read_excel('./Book1.xls')` `data.to_json(path_or_buf='my.json', orient='records')` 其中`Book1`是 Excel 工作表的名稱，`my.json`是輸出 JSON 檔案的名稱。	資料工程師、Python 技能
將 JSON 檔案上傳到 S3 儲存貯體。	上傳 `my.json` 至 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。	應用程式開發人員

插入記錄

任務	描述	所需技能
建立私有託管區域。	使用創建的主機 API 和下面的 Python 示例代碼來創建一個私有託管區域。`vpcId`用您自己的值取代參數`hostedZoneName` `vpcRegion` 、和。 `import boto3` `import random` `hostedZoneName ="xxx"`	雲端架構師、網路管理員、Python 技能

任務	描述	所需技能
	<pre>vpcRegion = "us-east-1" vpcId="vpc-xxxx" route53_client = boto3.client('route53') response = route53_client.create_hosted_zone(Name= hostedZoneName, VPC={ 'VPCRegion: vpcRegion, 'VPCId': vpcId }, CallerReference=str(random.random()*100000), HostedZoneConfig={ 'Comment': "private hosted zone created by automation", 'PrivateZone': True }) print(response)</pre> 您也可以使用基礎設施即程式碼 (IaC) 工具 (例如 AWS)，CloudFormation 將這些步驟取代為使用適當資源和屬性建立堆疊的範本。	

任務	描述	所需技能
從 Amazon S3 擷取作為字典的詳細資訊。	使用下列程式碼從 S3 儲存貯體讀取，並以 Python 字典的形式取得 JSON 值。 `fileobj = s3_client .get_object( Bucket=bu cket_name, Key='my.json' ) filedata = fileobj[' Body'].read() contents = filedata. decode('utf-8') json_content=json. loads(contents) print(json_content )` 其中`json_content` 包含 Python 字典。	應用程式開發者、Python 技能

任務	描述	所需技能
清理空格和 Unicode 字符的數據值。	為確保資料正確性的安全措施，請使用下列程式碼對中的值執行分段作業。`json_content` 此代碼刪除每個字符串的前面和結尾的空格字符。它還使用該`replace`方法來刪除硬（不斷行）空格（字`\xa0`符）。 `for item in json_content:` `    fqn_name = unicodedata.normalize("NFKD",item["FqdnName"].replace("u'", "'").replace('\xa0', '').strip())` `    rec_type = item["RecordType"].replace('\xa0', '').strip()` `    res_rec = {` `  'Value': item["Value"].replace('\xa0', '').strip()` `                }`	應用程式開發者、Python 技能

任務	描述	所需技能
插入記錄。	使用下面的代碼作為上一個for循環的一部分。 `change_response = route53_client.change_resource_record_sets( HostedZoneId="xxxxxxxx", ChangeBatch={ 'Comment': 'Created by automation', 'Changes': [ { 'Action': 'UPSERT', 'ResourceRecordSet': { 'Name': fqn_name, 'Type': rec_type, 'TTL': item["TTL"], 'ResourceRecords': res_rec } } ] } )` 這部史詩第一步的託管區域 ID 在xxxxxxx哪裡。	應用程式開發者、Python 技能

相關資源

參考

- 透過匯入區域檔案建立記錄 (Amazon Route 53 說明文件)
- 創建區域方法（博托 3 文檔）
- 更改資源記錄集方法（博圖 3 文檔）

教學課程和影片

- Python 教程 (Python 文件)

- 使用 Amazon Route 53 進行 DNS 設計 (YouTube 影片、AWS 線上技術會談)

在 AWS 上從 F5 遷移到 Application Load Balancer 時修改 HTTP 標頭

由薩欽特里韋迪（AWS）創建

環境：PoC 或試點	來源：內部部署	目標：AWS 雲端
R 類型：重新平台	工作負載：所有其他工作	技術：網路、混合雲、移轉
AWS 服務：Amazon CloudFront；Elastic Load Balancing (ELB)；AWS Lambda		

Summary

當您將使用 F5 負載平衡器的應用程式遷移到 Amazon Web Services (AWS) 並想要在 AWS 上使用應用程式負載平衡器時，遷移 F5 標題修改規則是一個常見問題。應用程式負載平衡器不支援標頭修改，但您可以使用 Amazon CloudFront 做為內容交付網路 (CDN)，使用 Lambda @Edge 來修改標頭。

此模式描述必要的整合，並提供使用 AWS 和 Lambda @Edge 進行標頭修改 CloudFront 的範例程式碼。

先決條件和限制

先決條件

- 使用 F5 負載平衡器的內部部署應用程式，其組態可以使用來取代 HTTP 標頭值`if, else`。如需有關此組態的詳細資訊，請參閱 F5 產品文件中的 HTTP:: header。

限制

- 此模式適用於 F5 負載平衡器標頭自訂。對於其他第三方負載平衡器，請查看負載平衡器文件以取得支援資訊。
- 您用於 Lambda @Edge 的 Lambda 函數必須位於美國東部 (維吉尼亞北部) 區域。

架構

下圖顯示 AWS 上的架構，包括 CDN 和其他 AWS 元件之間的整合流程。

工具

AWS 服務

- 應用程式負載平衡器 – 應用程式負載平衡器是 AWS 全受管負載平衡服務，可在開放系統互連 (OSI) 模型的第七層運作。它平衡多個目標的流量，並支援基於 HTTP 標頭和方法、查詢字串以及主機型或路徑型路由的進階路由要求。
- Amazon CloudFront — Amazon CloudFront 是一種網絡服務，可以加快向用戶分發靜態和動態 Web 內容（例如 .html，.css，.js 和圖像文件）的速度。 CloudFront 透過稱為節點位置的全球資料中心網路傳遞您的內容，以降低延遲並提升效能。
- Lambda @Edge – Lambda @Edge 是 AWS Lambda 的擴充功能，可讓您執行函數來自訂 CloudFront 交付的內容。您可以在美國東部 (維吉尼亞北部) 區域撰寫函數，然後將函數與 CloudFront 分發產生關聯，以便在全球範圍內自動複寫您的程式碼，而無需佈建或管理伺服器。這樣可以減少延遲並改善使用者體驗。

Code

下列範例程式碼提供修改 CloudFront 回應標頭的藍圖。依照 Epics 一節中的指示部署程式碼。

```
exports.handler = async (event, context) => {
    const response = event.Records[0].cf.response;
    const headers = response.headers;


    const headerNameSrc = 'content-security-policy';
    const headerNameValue = '*.xyz.com';


    if (headers[headerNameSrc.toLowerCase()]) {
        headers[headerNameSrc.toLowerCase()] = [{
            key: headerNameSrc,
            value: headerNameValue,
        }];
        console.log(`Response header "${headerNameSrc}" was set to ` +
```

```
                    `"${headers[headerNameSrc.toLowerCase()][0].value}"`);
    }
    else {
            headers[headerNameSrc.toLowerCase()] = [{
            key: headerNameSrc,
            value: headerNameValue,
            }];
    }
    return response;
};
```

史诗

創建一個 CDN 分發

任務	描述	所需技能
建立 CloudFront Web 分發。	在此步驟中，您會建立一個 CloudFront 發佈，告訴您 CloudFront 要將內容傳送來源，以及如何追蹤和管理內容傳遞的詳細資料。 若要使用主控台建立分發，請登入 AWS 管理主控台，開啟CloudFront 主控台，然後按照CloudFront 文件中的步驟操作。	雲端管理員

建立和部署 Lambda 函數 @Edge

任務	描述	所需技能
建立和部署 Lambda 函數。@Edge	您可以使用用於修改 CloudFront 回應標頭的藍圖來建立 Lambda @Edge 函數。其他藍圖適用於不同的使用案例；如需詳細資訊，請參閱	AWS 管理員

任務	描述	所需技能
	CloudFront 文件中的 Lambda @Edge 範例函數。) 若要建立 Lambda 函數：@Edge 1. 登入 AWS 管理主控台，並開啟位於 https://console.aws.amazon.com/lambda/ 的 AWS Lambda 主控台。 2. 確認您位於美國東部 (維吉尼亞北部) 區域。CloudFront 藍圖僅在此區域提供。 3. 選擇 建立函式。 4. 選擇 [使用藍圖]，然後在 [藍圖] 搜尋欄位中輸入 cloudfront。 5. 選擇cloudfront-modify-response-header藍圖，然後選擇設定。 6. 在「基本」資訊頁上，輸入下列資訊： a. 輸入函數名稱。 b. 針對 Execution role (執行角色)，選擇 Create a new role from AWS policy templates (從 AWS 政策範本建立新角色)。 c. 關聯所需的 AWS Identity and Access Management (IAM) 角色名稱。 7. 選擇 建立函式。	

任務	描述	所需技能
	8. 在頁面的 [設計器] 區段中，選擇您的函數名稱。 9. 在「函數程式碼」區段中，將範本程式碼取代為先前在此模式中提供的範例程式碼，在「程式碼」區段中。 10.在範例程式碼中，`xyz.com`以您的網域名稱取代。 11選擇儲存。	
部署 Lambda 函數。@Edge	遵循 Amazon CloudFront 文件中教學課程：建立簡單 Lambda @Edge 函數的步驟 4 中的指示來設定 CloudFront 觸發器並部署函數。	AWS 管理員

相關資源

CloudFront 文件

- 自訂來源的要求和回應行為
- 使用發行版
- Lambda @Edge 範例函數
- 使用 Lambda @Edge 在邊緣進行自訂
- 教學課程：建立簡單的 Lambda @Edge 函數

從多個 VPC 私有存取中央 AWS 服務端點

由馬丁·甘特納（AWS）和塞繆爾·戈登（AWS）創建

程式碼儲存庫：VPC 端點共用	環境：生產	技術：網路；基礎架構
AWS 服務：AWS RAM；Amazon Route 53；Amazon SNS；AWS Transit Gateway；Amazon VPC		

Summary

您環境的安全和合規要求可能會指定到 Amazon Web Services (AWS) 服務或端點的流量不得穿越公用網際網路。此模式是專為hub-and-spoke拓撲而設計的解決方案，其中中央集線器 VPC 連接到多個分散式支點 VPC。在此解決方案中，您可 PrivateLink 以使用 AWS 為中樞帳戶中的 AWS 服務建立界面 VPC 端點。然後，您可以使用傳輸閘道和分散式網域名稱系統 (DNS) 規則，跨連線的 VPC 解析對端點私有 IP 位址的要求。

此模式說明如何使用 AWS Transit Gateway、傳入 Amazon Route 53 Resolver 端點和共用 Route 53 轉送規則，以便從連線 VPC 中的資源解析 DNS 查詢。您可以在 Hub 帳戶中建立端點、傳輸閘道、解析器和轉送規則。然後，您可以使用 AWS Resource Access Manager (AWS RAM) 與支點 VPC 共用傳輸閘道和轉送規則。所提供的 AWS CloudFormation 範本可協助您在中樞 VPC 和支點 VPC 中部署和設定資源。

先決條件和限制

先決條件

- 在 AWS Organizations 的同一個組織中管理的中樞帳戶和一或多個支點帳戶。如需詳細資訊，請參閱建立和管理組織。
- AWS Resource Access Manager (AWS RAM) 在 AWS Organizations 中設定為受信任的服務。如需詳細資訊，請參閱將 AWS Organizations 與其他 AWS 服務搭配使用。
- 必須在集線器和支點 VPC 中啟用 DNS 解析。如需詳細資訊，請參閱 VPC 的 DNS 屬性 (Amazon Virtual Private Cloud 文件)。

限制

- 此模式會連接相同 AWS 區域中的中樞和支點帳戶。對於多區域部署，您必須針對每個區域重複此模式。
- AWS 服務必須 PrivateLink 以接口虛擬私人雲端端點的形式整合。如需完整清單，請參閱與 AWS 整合的 AWS 服務 PrivateLink (PrivateLink 文件)。
- 不保證可用區域相似性。例如，來自可用區域 A 的查詢可能會以可用區域 B 的 IP 位址回應。
- 與虛擬私人雲端端點關聯的 elastic network interface 每秒有 10,000 個查詢的限制。

架構

目標技術堆疊

- 集線器 AWS 帳戶中的集線器 VPC
- 支點 AWS 帳戶中的一個或多個支點 VPC
- 集線器帳戶中的一或多個介面 VPC 端點
- 集線器帳戶中的入站和出站 Route 53 解析器
- Route 53 解析器轉送規則部署在集線器帳戶中，並與支點帳戶共用
- 部署在 Hub 帳戶中並與支點帳戶共用的傳輸閘道
- 連接中樞和支點 VPC 的 AWS Transit Gateway

目標架構

下圖顯示此解決方案的範例架構。在此架構中，集線器帳戶中的 Route 53 解析器轉送規則與其他架構元件具有下列關係：

1. 轉送規則是透過使用 AWS 記憶體與支點 VPC 人雲端共用。
2. 轉送規則與中樞 VPC 中的輸出解析程式相關聯。
3. 轉送規則會鎖定中樞 VPC 中的輸入解析程式。

下圖顯示了通過範例架構的流量：

1. 支點 VPC 中的資源 (例如亞馬遜彈性運算雲端 (Amazon EC2) 執行個體，會向<service>.<region>.amazonaws.com其發出 DNS 請求。該請求由支點 Amazon DNS 解析器接收。
2. Route 53 轉送規則 (從集線器帳戶共用並與支點 VPC 相關聯) 會攔截要求。
3. 在中樞 VPC 中，輸出解析程式會使用轉送規則將要求轉送至輸入解析器。
4. 輸入解析器使用集線器 VPC Amazon DNS 解析器將的 IP 位址解析為 VPC 端點<service>.<region>.amazonaws.com的私有 IP 位址。如果沒有 VPC 端點，它會解析為公用 IP 位址。

工具

AWS 工具和服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以根據需要啟動任意數量的虛擬伺服器，並快速擴展或縮減它們。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Resource Access Manager (AWS RAM) 可協助您在 AWS 帳戶之間安全地共用資源，以減少營運開銷，並提供可見性和可稽核性。
- Amazon Route 53 是一種可用性高、可擴展性強的網域名稱系統 (DNS) Web 服務。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。
- AWS Transit Gateway 是連接 VPC 和現場部署網路的中央中樞。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他工具和服務

- nslookup 是用來查詢 DNS 記錄的命令列工具。在此模式中，您可以使用此工具來測試解決方案。

代碼存儲庫

此模式的程式碼可在 GitHubvpc-endpoint-sharing存放庫中取得。此模式提供兩個 AWS CloudFormation 範本：

- 在 Hub 帳戶中部署下列資源的範本：
  - rSecurityGroupEndpoints— 控制 VPC 端點存取權的安全群組。
  - rSecurityGroupResolvers— 控制 Route 53 解析程式存取權的安全性群組。
  - rKMSEndpoint、rSSMMessagesEndpointrSSMEndpoint、和 rEC2MessagesEndpoint — 中樞帳戶中的介面 VPC 端點範例。針對您的使用案例自訂這些端點。
  - rInboundResolver— Route 53 解析器，解決針對集線器的 DNS 查詢 Amazon DNS 解析器.
  - rOutboundResolver— 將查詢轉發到入站解析器的出站 Route 53 解析器。
  - rAWSApiResolverRule— 與所有支點 VPC 共用的 Route 53 解析器轉送規則。
  - rRamShareAWSResolverRule— 允許支點 VPC 使用rAWSApiResolverRule轉送規則的 AWS RAM 共用。
  - * rVPC — 集線器 VPC，用於建模共用服務。
  - * rSubnet1 — 用來容納集線器資源的私有子網路。
  - * rRouteTable1 — 集線器 VPC 的路由表。
  - * rRouteTableAssociation1 — 對於集rRouteTable1線器 VPC 中的路由表，則為私有子網路的關聯。
  - * rRouteSpoke — 從集線器 VPC 到支點 VPC 的路由。
  - * rTgw — 與所有支點 VPC 共用的傳輸閘道。
  - * rTgwAttach — 允許集線器 VPC 將流量路由到rTgw傳輸閘道的附件。
  - * rTgwShare — 允許支點帳戶使用rTgw傳輸閘道的 AWS RAM 共用。
- 在支點帳戶中部署下列資源的範本：
  - rAWSApiResolverRuleAssociation— 允許分支 VPC 使用 Hub 帳戶中共用轉送規則的關聯。
  - * rVPC — 該輻條 VPC.
  - * rSubnet1, rSubnet2, rSubnet3 — 每個可用區域的子網路，用來容納分支式私有資源。
  - * rTgwAttach — 允許分支 VPC 將流量路由到rTgw傳輸閘道的附件。
  - * rRouteTable1 — 輪輻 VPC 的路由表。
  - * rRouteEndpoints — 從支點 VPC 中的資源到傳輸閘道的路由。

- * `rRouteTableAssociation1/2/3` — 對於網輻 VPC 中的`rRouteTable1`路由表，是私有子網路的關聯。
- * `rInstanceRole` — 用來測試解決方案的 IAM 角色。
- * `rInstancePolicy` — 用來測試解決方案的 IAM 政策。
- * `rInstanceSg` — 用來測試解決方案的安全性群組。
- * `rInstanceProfile` — 用於測試解決方案的 IAM 執行個體設定檔。
- * `rInstance` — 預先設定為透過 AWS Systems Manager 存取的 EC2 執行個體。使用此執行個體來測試解決方案。

* 這些資源支援範例架構，在現有的 landing zone 實作此模式時，可能不需要這些資源。

史诗

準備 CloudFormation 範本

任務	描述	所需技能
克隆代碼存儲庫。	1. 在指令行介面中，將工作目錄變更為要儲存範例檔案的位置。 2. 輸入以下命令： `git clone https://github.com/aws-samples/vpc-endpoint-sharing.git`	網路管理員、雲端架構師
修改範本。	1. 在克隆的存儲庫中，打開 hub.yml 和欺騙 .yml 文件。 2. 檢閱這些範本所建立的資源，並根據您的環境需要調整範本。如需完整清單，請參閱工具中的程式碼儲存庫一節。如果您的帳戶已擁有其中一些資源，請將其從 CloudFormation 範	網路管理員、雲端架構師

任務	描述	所需技能
	本中移除。如需詳細資訊，請參閱使用範本 (CloudFormation 文件集)。 3. 保存並關閉 hub.yml 並欺騙 .yml 文件。	

在目標帳戶中部署資源

任務	描述	所需技能
部署集線器資源。	使用 hub.yml 模板，創建一個堆棧。 CloudFormation 出現提示時，請為樣板中的參數提供值。如需詳細資訊，請參閱建立堆疊 (CloudFormation 文件集)。	雲端架構師、網路管理員
部署支點資源。	使用 spoke.yml 模板，創建一個堆棧。 CloudFormation 出現提示時，請為樣板中的參數提供值。如需詳細資訊，請參閱建立堆疊 (CloudFormation 文件集)。	雲端架構師、網路管理員

測試解決方案

任務	描述	所需技能
測試 AWS 服務的私有 DNS 查詢。	1. 使用工作階段管理員 (AWS Systems Manager 的一項功能) Connect 至 `rInstance` EC2 執行個體。如需詳細資訊，請參閱使用工作階段管理員 Connect 到 Linux	網路管理員

任務	描述	所需技能
	執行個體 (Amazon EC2 文件)。 2. 對於中樞帳戶中具有 VPC 端點的 AWS 服務，請使用`nslookup`來確認傳回傳入 Route 53 解析器的私有 IP 地址。 以下是使用連`nslookup`接 Amazon Systems Manager 端點的示例。 `nslookup ssm.<region>.amazonaws.com` 3. 在 AWS Command Line Interface (AWS CLI) (AWS CLI) 中，輸入可協助您確認變更不會影響服務功能的命令。如需命令清單，請參閱 AWS CLI 命令參考。 例如，下列命令應傳回 Amazon Systems Manager 文件清單。 `aws ssm list-documents`	

任務	描述	所需技能
測試 AWS 服務的公用 DNS 查詢。	1. 對於中樞帳戶中沒有 VPC 端點的 AWS 服務，請使用`nslookup`來確認傳回公用 IP 地址。以下是使用`nslookup`連接亞馬遜簡單通知服務 (Amazon SNS) 端點的範例。 `nslookup sns.<region>.amazonaws.com` 2. 在 AWS CLI 中，輸入可協助您確認變更不會影響服務功能的命令。如需命令清單，請參閱 AWS CLI 命令參考。 例如，如果集線器帳戶中有任何 Amazon SNS 主題，則下列命令應傳回主題清單。 `aws sns list-topics`	網路管理員

相關資源

- 建立可擴展且安全的多 VPC AWS 網路基礎設施 (AWS 白皮書)
- 使用共用資源 (AWS 記憶體文件)
- 使用傳輸閘道 (AWS Transit Gateway 文件)

針對多個 AWS 帳戶的傳入網際網路存取建立網路存取分析器發現的報告

創建者：邁克·維爾吉利奧（AWS）

程式碼儲存庫：網路存取分析器多帳戶分析	環境：生產	技術：網路；安全性、身分識別、合規
AWS 服務：AWS CloudFormation；Amazon S3；Amazon VPC；AWS Security Hub		

Summary

無意的傳入網際網路存取 AWS 資源可能會對組織的資料周邊構成風險。網路存取分析器是一種 Amazon Virtual Private Cloud (Amazon VPC) 功能，可協助您識別對 Amazon Web Services (AWS) 資源的意外網路存取。您可以使用「網路存取分析器」來指定您的網路存取需求，並識別不符合您指定需求的潛在網路路徑。您可以使用「網路存取分析器」執行下列作業：

1. 識別可透過網際網路閘道存取網際網路的 AWS 資源。
2. 驗證您的虛擬私有雲 (VPC) 是否適當地分段，例如隔離生產和開發環境，以及分離交易工作負載。

網絡訪問分析儀分析 end-to-end 網絡可達性條件，而不僅僅是單個組件。若要判斷資源是否可存取網際網路，網路存取分析程式會評估網際網路閘道、VPC 路由表、網路存取控制清單 (ACL)、彈性網路介面上的公用 IP 位址以及安全性群組。如果這些元件中的任何一個阻止網際網路存取，網路存取分析器不會產生發現項目。例如，如果 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體具有允許來自流量的開放安全群組，`0/0`但該執行個體位於無法從任何網際網路閘道路由的私有子網路中，則網路存取分析器將不會產生發現項目。這提供了高保真度的結果，以便您可以識別真正可從 Internet 訪問的資源。

當您執行網路存取分析器時，您可以使用網路存取範圍來指定您的網路存取需求。此解決方案可識別網際網路閘道與 elastic network interface 之間的網路路徑。在此模式中，您可以在組織中的集中式 AWS 帳戶中部署解決方案 (由 AWS Organizations 管理)，並分析組織中任何 AWS 區域中的所有帳戶。

此解決方案的設計時考慮到以下幾點：

- AWS CloudFormation 範本可減少以此模式部署 AWS 資源所需的工作量。
- 您可以在部署時調整 CloudFormation 範本和 naa-script.sh 指令碼中的參數，以針對您的環境自訂參數。
- Bash 腳本會自動佈建和分析多個帳戶的網絡訪問範圍，並 parallel。
- Python 腳本處理發現項目，提取數據，然後合併結果。您可以選擇以 CSV 格式或 AWS Security Hub 檢閱網路存取分析器發現的合併報告。您可以在此模式的「其他資訊」區段中找到 CSV 報告的範例。
- 您可以修正發現項目，或者透過將發現項目新增至 naa-exclusions.csv 檔案，將它們排除在 future 的分析之外。

先決條件和限制

先決條件

- 用於託管安全服務和工具的 AWS 帳戶，以 AWS Organizations 中組織的成員帳戶進行管理。在這種模式中，這個帳戶被稱為安全帳戶。
- 在安全性帳戶中，您必須擁有具有輸出網際網路存取權的私有子網路。如需指示，請參閱 Amazon VPC 文件中的建立子網路。您可以使用 NAT 閘道或介面 VPC 端點來建立網際網路存取。
- 存取 AWS Organizations 管理帳戶或已委派管理員許可的帳戶 CloudFormation。如需指示，請參閱 CloudFormation 文件中的註冊委派管理員。
- 在 AWS Organizations 和. 之間啟用受信任的存取 CloudFormation。如需指示，請參閱 CloudFormation 文件中的啟用 AWS Organizations 的受信任存取。
- 如果您要將調查結果上傳到 Security Hub，則必須在佈建 EC2 執行個體的帳戶和 AWS 區域中啟用 Security Hub。如需詳細資訊，請參閱設定 AWS Security Hub。

限制

- 由於網路存取分析器功能的限制，目前不會分析跨帳戶網路路徑。
- 目標 AWS 帳戶必須以 AWS Organizations 中的組織形式進行管理。如果您不是使用 AWS Organizations，您可以更新適用於您環境的 naa-script.sh 指令碼 CloudFormation 範本和指令碼。而是提供要執行指令碼的 AWS 帳戶 ID 和區域清單。
- 該 CloudFormation 範本旨在將 EC2 執行個體部署在具有輸出網際網路存取權的私有子網路中。AWS Systems Manager 代理程式 (SSM 代理程式) 需要輸出存取權才能連接到 Systems Manager 服務端點，而且您需要輸出存取權，才能複製程式碼儲存庫並安裝相依性。如果您想要

使用公有子網路，則必須修改 naa-resource .yaml 範本，以將彈性 IP 地址與 EC2 執行個體建立關聯。

架構

目標技術堆疊

- 網路存取分析器
- Amazon EC2 執行個體
- AWS Identity and Access Management (IAM) 角色
- Amazon Simple Storage Service (Amazon S3) 儲存貯體
- Amazon Simple Notification Service (Amazon SNS) 主題
- AWS Security Hub (僅限選項 2)

目標架構

選項 1：存取 Amazon S3 儲存貯體中的發現項目

該圖顯示了以下過程：

1. 如果您手動執行解決方案，使用者會使用工作階段管理員對 EC2 執行個體進行驗證，然後執行 naa-script.sh 指令碼。這個命令介面指令碼會執行步驟 2—7。

   如果您自動執行解決方案，naa-script.sh 指令碼會根據您在 cron 運算式中定義的排程自動啟動。這個命令介面指令碼會執行步驟 2—7。如需詳細資訊，請參閱本節末尾的自動化和調整規模。
2. EC2 執行個體會從 S3 儲存貯體下載最新的 naa-exception.csv 檔案。這個檔案會在稍後的程序中使用，當 Python 指令碼處理排除項目。
3. EC2 執行個體擔任 `NAAEC2Role` IAM 角色，該角色授予存取 S3 儲存貯體的許可，並在組織中的其他帳戶中擔任 `NAAExecRole` IAM 角色。
4. EC2 執行個體在組織的管理帳戶中擔任 `NAAExecRole` IAM 角色，並產生組織中的帳戶清單。
5. EC2 執行個體在組織的成員帳戶中擔任 `NAAExecRole` IAM 角色 (在架構圖中稱為工作負載帳戶)，並在每個帳戶中執行安全評估。發現項目會以 JSON 檔案形式儲存在 EC2 執行個體上。
6. EC2 執行個體使用 Python 指令碼來處理 JSON 檔案、擷取資料欄位，以及建立 CSV 報告。

7. EC2 執行個體會將 CSV 檔案上傳到 S3 儲存貯體。
8. Amazon EventBridge 規則會偵測檔案上傳，並使用 Amazon SNS 主題傳送電子郵件通知使用者報告已完成。
9. 使用者會從 S3 儲存貯體下載 CSV 檔案。使用者將結果匯入 Excel 範本並檢閱結果。

選項 2：存取 AWS Security Hub 中的發現項目

該圖顯示了以下過程：

1. 如果您手動執行解決方案，使用者會使用工作階段管理員對 EC2 執行個體進行驗證，然後執行 naa-script.sh 指令碼。這個命令介面指令碼會執行步驟 2—7。

   如果您自動執行解決方案，naa-script.sh 指令碼會根據您在 cron 運算式中定義的排程自動啟動。這個命令介面指令碼會執行步驟 2—7。如需詳細資訊，請參閱本節末尾的自動化和調整規模。
2. EC2 執行個體會從 S3 儲存貯體下載最新的 naa-exception.csv 檔案。這個檔案會在稍後的程序中使用，當 Python 指令碼處理排除項目。
3. EC2 執行個體擔任 `NAAEC2Role` IAM 角色，該角色授予存取 S3 儲存貯體的許可，並在組織中的其他帳戶中擔任 `NAAExecRole` IAM 角色。
4. EC2 執行個體在組織的管理帳戶中擔任 `NAAExecRole` IAM 角色，並產生組織中的帳戶清單。
5. EC2 執行個體在組織的成員帳戶中擔任 `NAAExecRole` IAM 角色 (在架構圖中稱為工作負載帳戶)，並在每個帳戶中執行安全評估。發現項目會以 JSON 檔案形式儲存在 EC2 執行個體上。
6. EC2 執行個體使用 Python 指令碼來處理 JSON 檔案，並擷取資料欄位以匯入 Security Hub。
7. EC2 執行個體會將網路存取分析器發現項目匯入 Security Hub。
8. Amazon EventBridge 規則會偵測匯入，並使用 Amazon SNS 主題傳送電子郵件通知使用者程序已完成。
9. 使用者檢視安全中心中的發現項目。

自動化和規模

您可以排程此解決方案，以便根據自訂排程自動執行 naa-script.sh 指令碼。若要設定自訂排程，請在 naa-Resources .yaml CloudFormation 範本中修改參數。`CronScheduleExpression`例如，的預設值會在每個星期日午夜`0 0 * * 0`執行解決方案。的值`0 0 * 1-12 0`會在每個月的第一個星期日午夜執行解決方案。如需有關使用 cron 運算式的詳細資訊，請參閱 Systems Manager 說明文件中的 Cron 和速率運算式。

如果您要在/etc/cron.d/naa-schedule部署NAA-Resources堆疊之後調整排程，可以在中手動編輯 cron 排程。

工具

AWS 服務

- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。它也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。此模式使用「會話管理器」，這是 Systems Manager 器的一種功能。

代碼存儲庫

此病毒碼的程式碼可在「 GitHub 網路存取分析器多帳戶分析」儲存區域中找到。程式碼儲存庫包含下列檔案：

- naa-script.sh — 此 bash 指令碼用於 parallel 時啟動多個 AWS 帳戶的網路存取分析器分析。如 naa-資源.yaml CloudFormation 範本中所定義，此指令碼會自動部署至 EC2 執行個體上的資/usr/local/naa料夾。

- naa-resources .yaml — 您可以使用此 CloudFormation 範本在組織中的安全性帳戶中建立堆疊。此範本會部署此帳戶的所有必要資源，以支援解決方案。此堆疊必須在 naa-ex ecro.yaml 範本之前部署。

  備註：如果刪除並重新部署此堆疊，您必須重建NAAExecRole 堆疊集，以便重建 IAM 角色之間的跨帳戶相依性。

- naa-execrole.yaml — 您可以使用此 CloudFormation 範本建立堆疊集，以在組織中的所有帳戶 (包括管理帳戶) 中部署 NAAExecRole IAM 角色。
- naa-processfindings.py — naa-script.sh 指令碼會自動呼叫此 Python 指令碼來處理網路存取分析器 JSON 輸出，排除 naa-exclusions.csv 檔案中的任何已知良好資源，然後產生合併結果的 CSV 檔案，或將結果匯入 Security Hub。

史诗

準備部署

任務	描述	所需技能
克隆代碼存儲庫。	1. 在指令行介面中，將工作目錄變更為要儲存範例檔案的位置。 2. 輸入以下命令。 `git clone https://github.com/aws-samples/network-access-analyzer-multi-account-analysis.git`	AWS DevOps
檢閱範本。	1. 在複製的儲存庫中，開啟導航資源 .yaml 和 naa-execro.yaml 檔案。 2. 檢閱這些範本所建立的資源，並根據您的環境需要調	AWS DevOps

任務	描述	所需技能
	整範本。如需詳細資訊，請參閱 CloudFormation 文件中的使用範本。 3. 儲存並關閉資源 .yaml 和 naa-Execro.yaml 檔案。	

建立 CloudFormation 堆疊

任務	描述	所需技能
在安全性帳戶中佈建資源。	您可以使用 naa-resource .yaml 範本建立一個 CloudFormation 堆疊，以部署安全性帳戶中所有必要的資源。如需指示，請參閱 CloudFormation 文件中的建立堆疊。部署此範本時，請注意下列事項： 1. 在 [指定範本] 頁面上，選擇 [範本已準備就緒]，然後上傳 naa-resource .yaml 檔案。 2. 在 [指定堆疊詳細資料] 頁面的 [堆疊名稱] 方塊中，輸入`NAA-Resources` 。 3. 在「參數」段落中，輸入下列內容： • `VPCId`— 在帳戶中選擇一個 VPC。 • `SubnetId`— 選取具有網際網路存取權限的私有子網路。	AWS DevOps

任務	描述	所需技能
	備註：如果您選取公有子網路，EC2 執行個體可能不會被指派公用 IP 位址，因為 CloudFormation 範本預設不會佈建和附加彈性 IP 位址。 • `InstanceType` — 保留預設執行個體類型。 • `InstanceImageId` — 保留預設值。 • `KeyPairName` — 如果您使用 SSH 進行訪問，請指定現有 key pair 的名稱。 • `PermittedSSHInbound` — 如果您使用 SSH 進行存取，請指定允許的 CIDR 區塊。如果您不使用 SSH，請保留的預設值`127.0.0.1` 。 • `BucketName` — 預設值為`naa-<accountID>-<region>` 。您可以根據需要修改此項。如果您指定自訂值，帳戶 ID 和區域會自動附加至指定的值。 • `EmailAddress` — 指定分析完成時用於 Amazon SNS 通知的電子郵件地址。	

任務	描述	所需技能
	注意：必須在分析完成前確認 Amazon SNS 訂閱組態，否則將不會傳送通知。 • NAAEC2Role — 保留預設值，除非您的命名慣例需要此 IAM 角色不同的名稱。 • NAAExecRole — 保留預設值，除非在部署 na a-execrole.yaml 時會使用其他名稱 • Parallelism — 指定要執行的 parallel 評估數目。 • Regions— 指定要分析的 AWS 區域。 • ScopeNameValue — 指定要指派給範圍的標籤。此標記是用來決定網路存取範圍。 • ExclusionFile — 指定排除檔案名稱。此檔案中的項目將從發現項目中排除。 • FindingsToCSV — 指定是否應將發現項目輸出至 CSV。接受的值為 true和false。 • FindingsToSecurity Hub — 指定是否應將發現項目匯入 Security	

任務	描述	所需技能
	Hub。接受的值為 `true`和`false`。 • `EmailNotifications ForSecurityHub` — 指定將發現項目匯入 Security Hub 是否應該產生電子郵件通知。接受的值為`true`和`false`。 • `Scheduled Analysis` — 如果您希望解決方案依排程自動執行，請輸入`true`，然後在`CronSched uleExpression` 參數中自訂排程。如果您不想自動執行解決方案，請輸入`false`。 • `CronScheduleExpres sion` — 如果您要自動執行解決方案，請輸入 cron 運算式以定義排程。如需詳細資訊，請參閱此模式的「架構」一節中的「自動化和擴充」。 1. 在 [複查] 頁面上，選取下列資源需要功能：[AWS::IAM ::Role]，然後選擇 [建立堆疊]。 2. 成功建立堆疊後，在 CloudFormation 主控台的「輸出」索引標籤上，複製 `NAAEC2Role` Amazon 資源名稱 (ARN)。您稍後在部	

任務	描述	所需技能
	署 naa- execro.yaml 檔案時使用此 ARN。	

任務	描述	所需技能
在成員帳戶中佈建 IAM 角色。	在 AWS Organizations 管理帳戶或具有委派管理員許可的帳戶中 CloudFormation，使用 naa-execrole.yaml 範本建立堆疊集。 CloudFormation 堆疊集會在組織中的所有成員帳戶中部署 `NAAExecRole` IAM 角色。如需指示，請參閱 CloudFormation 文件中的建立具有服務管理權限的堆疊集。部署此範本時，請注意下列事項： 1. 在 [準備範本] 下方，選擇 [範本已準備就緒]，然後上傳 naa-ex ecrole.yaml 檔案。 2. 在 [指定 StackSet 詳細資料] 頁面上，命名堆疊集`NAA-ExecRole` 。 3. 在「參數」段落中，輸入下列內容： • `AuthorizedARN` — 輸入 `NAAEC2Role` ARN，您在建立`NAA-Resources` 堆疊時所複製的。 • `NAARoleName` — 保持預設值，`NAAExecRole` 除非在部署 naa-resources .yaml 檔案時使用了其他名稱。 4. 在 Permissions (許可) 下，選擇 Service-managed	AWS DevOps

任務	描述	所需技能
	permissions (服務管理許可)。 5. 在 [設定部署選項] 頁面的 [部署目標] 下，選擇 [部署至組織] 並接受所有預設值。 備註：如果您希望堆疊同時部署到所有成員帳戶，請將最大並行帳戶和失敗容忍度設定為高值，例如 100。 6. 在部署區域下，選擇網路存取分析器部署 EC2 執行個體的區域。由於 IAM 資源是全球性的，而不是區域資源，因此會在所有作用中區域中部署 IAM 角色。 7. 在 [檢閱] 頁面上，選取 [我確認 AWS CloudFormation 可能會使用自訂名稱建立 IAM 資源]，然後選擇 [建立] StackSet。 8. 監視 [堆疊執行個體] 索引標籤 (針對個別帳戶狀態) 和 [作業] 索引標籤 (針對整體狀態)，以判斷部署何時完成。	

任務	描述	所需技能
在管理帳戶中佈建 IAM 角色。	您可以使用 naa-execrole.yaml 範本建立一個 CloudFormation 堆疊，以在組織的管理帳戶中部署 NAAExecRole IAM 角色。您先前建立的堆疊集不會在管理帳戶中部署 IAM 角色。如需指示，請參閱 CloudFormation 文件中的建立堆疊。部署此範本時，請注意下列事項： 1. 在 [指定範本] 頁面上，選擇 [範本已準備就緒]，然後上傳 naa-ex ecro.yaml 檔案。 2. 在 [指定堆疊詳細資料] 頁面的 [堆疊名稱] 方塊中，輸入NAA-ExecRole 。 3. 在「參數」段落中，輸入下列內容： • AuthorizedARN — 輸入 NAAEC2Role ARN，您在建立NAA-Resources 堆疊時所複製的。 • NAARoleName — 保持預設值，NAAExecRole 除非在部署 naa-resources .yaml 檔案時使用了其他名稱。 4. 在 [複查] 頁面上，選取下列資源需要功能：[AWS::IAM::Role]，然後選擇 [建立堆疊]。	AWS DevOps

執行分析

任務	描述	所需技能
自訂殼層指令碼。	1. 登入組織中的安全性帳戶。 2. 使用工作階段管理員，連接至您先前佈建的網路存取分析器的 EC2 執行個體。如需指示，請參閱使用工作階段管理員 Connect 到 Linux 執行個體。如果您無法連線，請參閱此模式的疑難排解一節。 3. 輸入下列指令以開啟 naa-script.sh 檔案進行編輯。 `sudo -i` `cd /usr/local/naa` `vi naa-script.sh` 4. 根據您的環境需要，檢閱並修改此指令碼中的可調整參數和變數。如需有關自訂選項的詳細資訊，請參閱指令碼開頭的註解。 例如，您可以修改指令碼以指定要掃描的 AWS 帳戶 ID 或 AWS 區域，而不是從管理帳戶取得組織中所有成員帳戶的清單，或者您也可以參考包含這些參數的外部檔案。 5. 儲存並關閉 naa-script.sh 檔案。	AWS DevOps

任務	描述	所需技能
分析目標帳戶。	1. 輸入下列命令：這會執行 naa-script.sh 指令碼。 `sudo -i` `cd /usr/local/naa` `screen` `./naa-script.sh` 注意下列事項： • 此命`screen`令允許指令碼在連線逾時或失去主控台存取權時繼續執行。 • 掃描開始後，您可以通過按 Ctrl+A D 強制卸離屏幕。 螢幕會分離，您可以在分析進行時關閉例證連接。 • 若要繼續分離的工作階段，請連線至執行個體，輸入`sudo -i`然後輸入`screen -r`。 2. 監視輸出是否有任何錯誤，以確定指令碼正常運作。如需範例輸出，請參閱此模式的「其他資訊」一節。 3. 等待分析完成。如果您設定電子郵件通知，則當結果上傳到 S3 儲存貯體或匯入 Security Hub 時，您會收到電子郵件。	AWS DevOps

任務	描述	所需技能
選項 1 — 從 S3 儲存貯體擷取結果。	1. 從`naa-<accountID>-<region>` 值區下載 CSV 檔案。如需指示，請參閱 Amazon S3 文件中的下載物件。 2. 從 S3 儲存貯體刪除 CSV 檔案。這是成本最佳化的最佳作法。如需指示，請參閱 Amazon S3 文件中的刪除物件。	AWS DevOps
選項 2 — 檢閱安全中心中的結果。	1. 開啟 Security Hub 主控台，網址為 https://console.aws.amazon.com/securityhub/。 2. 從導覽窗格中選擇「搜尋結果」。 3. 複查網路存取分析器發現項目。如需指示，請參閱 Security Hub 說明文件中的檢視尋找項目清單和詳細資料。 附註：您可以透過新增「標題開頭為」篩選條件並輸入來搜尋發現項目`Network Access Analyzer`。	AWS DevOps

補救並排除搜尋結果

任務	描述	所需技能
補救搜尋結果。	補救您要解決的任何搜尋結果。如需有關如何圍繞 AWS	AWS DevOps

任務	描述	所需技能
	身分、資源和網路建立周邊的詳細資訊和最佳實務，請參閱在 AWS 上建立資料周邊 (AWS 白皮書)。	

任務	描述	所需技能
排除具有已知良好網路路徑的資源。	如果網路存取分析程式針對應該可從網際網路存取的資源產生發現項目，則您可以將這些資源新增至排除清單。下次網路存取分析程式執行時，它不會產生該資源的發現項目。 1. 瀏覽至`/usr/local/naa` ，然後開啟 naa-script.sh 指令碼。記下`S3_EXCLUSION_FILE` 變數的值。 2. 如果`S3_EXCLUSION_FILE` 變數的值為`true`，請從值`naa-<accountID>-<region>` 區下載 naa-exclusions.csv 檔案。如需指示，請參閱 Amazon S3 文件中的下載物件。 如果`S3_EXCLUSION_FILE` 變數的值為`false`，請瀏覽至 naa-exclusions.csv 檔案，`/usr/local/naa` 然後開啟該檔案。 附註：如果`S3_EXCLUSION_FILE` 變數的值為`false`，指令碼會使用排除檔案的本機版本。如果您稍後將值變更為`true`，則指令碼會以 S3 儲存貯體中的檔案覆寫本機版本。	AWS DevOps

任務	描述	所需技能
	3. 在 naa-exclusions.csv 檔案中，輸入您要排除的資源。在每一行中輸入一個資源，並使用下列格式。 `<resource_id>,<secgroup_id>,<sgrule_cidr>,<sgrule_portrange>,<sgrule_protocol>` 以下是範例資源。 `eni-1111aaaaa2222bbbb,sg-3333ccccc4444dddd,0.0.0.0/0,80 to 80,tcp` 4. 儲存並關閉 naa-exclusions.csv 檔案。 5. 如果您是從 S3 儲存貯體下載 naa-exclusions.csv 檔案，請上傳新版本。如需指示，請參閱 Amazon S3 文件中的上傳物件。	

(選擇性) 更新 naa-script.sh 指令碼

任務	描述	所需技能
更新 naa-script.sh 指令碼。	如果要將 naa-script.sh 腳本更新為回購中的最新版本，請執行以下操作： 1. 使用工作階段管理員 Connect 至 EC2 執行個體。	AWS DevOps

任務	描述	所需技能
	如需指示，請參閱使用工作階段管理員 Connect 到 Linux 執行個體。 2. 輸入以下命令。 `sudo -i` 3. 導覽至 naa-script.sh 指令碼目錄。 `cd /usr/local/naa` 4. 輸入以下命令以隱藏本地腳本，以便您可以將自定義更改合併到最新版本中。 `git stash` 5. 輸入下列命令以取得最新版本的指令碼。 `git pull` 6. 輸入下列命令，將自訂指令碼與最新版本的指令碼合併。 `git stash pop`	

(選用) 清除

任務	描述	所需技能
刪除所有已部署的資源。	您可以保留帳戶中部署的資源。	AWS DevOps

任務	描述	所需技能
	如果您要取消佈建所有資源，請執行下列動作： 1. 刪除管理帳戶中佈建的NAA-ExecRole 堆疊。如需指示，請參閱 CloudFormation 文件中的刪除堆疊。 2. 刪除組織的管理帳戶或委派管理員帳戶中佈建的NAA-ExecRole 堆疊集。如需指示，請參閱 CloudFormation 文件中的刪除堆疊集。 3. 刪除 naa-<accountID>-<region> S3 儲存貯體中的所有物件。如需指示，請參閱 Amazon S3 文件中的刪除物件。 4. 刪除安全性帳戶中佈建的NAA-Resources 堆疊。如需指示，請參閱 CloudFormation 文件中的刪除堆疊。	

故障診斷

問題	解決方案
無法使用工作階段管理員連線至 EC2 執行個體。	SSM 代理程式必須能夠與 Systems Manager 端點通訊。請執行下列操作： 1. 驗證部署 EC2 執行個體的子網路具有網際網路存取權。 2. 重新啟動 EC2 執行個體。

問題	解決方案
部署堆疊集時，CloudFormation 主控台會提示您這樣做`Enable trusted access with AWS Organizations to use service-managed permissions` 。	這表示 AWS Organizations 和之間尚未啟用受信任的存取 CloudFormation。部署服務管理的堆疊集需要受信任的存取。選擇按鈕以啟用信任存取。如需詳細資訊，請參閱 CloudFormation 文件中的啟用受信任的存取。

相關資源

- 新增 — Amazon VPC 網路存取分析器 (AWS 部落格文章)
- AWS RE: 實施 2022 年-驗證 AWS 上有效的網路存取控制 (NIS202) (影片)
- 示範-使用網路存取分析儀進行整個組織的網際網路入口資料路徑分析 (影片)

其他資訊

控制台輸出示例

下列範例顯示產生目標帳戶清單和分析目標帳戶的輸出。

```
[root@ip-10-10-43-82 naa]# ./naa-script.sh
download: s3://naa-<account ID>-us-east-1/naa-exclusions.csv to ./naa-exclusions.csv

AWS Management Account: <Management account ID>

AWS Accounts being processed...
<Account ID 1> <Account ID 2> <Account ID 3>

Assessing AWS Account: <Account ID 1>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 2>, using Role: NAAExecRole
Assessing AWS Account: <Account ID 3>, using Role: NAAExecRole
Processing account: <Account ID 1> / Region: us-east-1
Account: <Account ID 1> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 2> / Region: us-east-1
Account: <Account ID 2> / Region: us-east-1 - Detecting Network Analyzer scope...
Processing account: <Account ID 3> / Region: us-east-1
Account: <Account ID 3> / Region: us-east-1 - Detecting Network Analyzer scope...
Account: <Account ID 1> / Region: us-east-1 - Network Access Analyzer scope detected.
```

```
Account: <Account ID 1> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
Account: <Account ID 2> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 2> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
Account: <Account ID 3> / Region: us-east-1 - Network Access Analyzer scope detected.
Account: <Account ID 3> / Region: us-east-1 - Continuing analyses with Scope ID.
 Accounts with many resources may take up to one hour
```

CSV 報告範例

下列影像是 CSV 輸出的範例。

使用 AWS Organizations 自動標記 Transit Gateway 附件

由理查德·米爾納瓦特（AWS），哈里斯·本阿尤布（AWS）和約翰·卡普斯（AWS）創建

程式碼儲存庫：Transit Gateway 附件標記器	環境：生產	技術：網路、基礎架構、管理與治理、營運
AWS 服務：AWS Step Functions；AWS Transit Gateway；Amazon VPC；AWS Lambda		

Summary

在 Amazon Web Services (AWS) 上，您可以使用 AWS Resource Access Manager 跨 AWS 帳戶界限共用 AWS 傳輸閘道。但是，當您跨帳戶界限建立「Transit Gateway」附件時，會建立不含「名稱」標籤的附件。這可能會使識別附件耗時。

此解決方案提供自動化機制，可針對由 AWS Organizations 管理的組織內帳戶收集每個 Transit Gateway 附件的相關資訊。該過程包括從「Transit Gateway」路由表中查找無類別域間路由（CIDR）範圍。然後，解決方案會將名稱標籤<CIDR-range>-<AccountName>以的形式套用至擁有傳輸閘道的帳戶內的附件。

此解決方案可與 AWS 解決方案庫中的解決方案 (例如無伺服器傳輸網路協調器) 搭配使用。無伺服器傳輸網路協調器可讓您大規模自動建立 Transit Gateway 附件。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 包含所有相關帳戶的 AWS Organizations
- 在組織根目錄下存取組織管理帳戶，以建立所需的 AWS Identity and Access Management (IAM) 角色
- 共用網路成員帳戶，其中包含與組織共用且具有附件的一或多個傳輸閘道

架構

AWS 管理主控台的下列螢幕擷取畫面顯示沒有關聯名稱標籤的 Transit Gateway 附件，以及兩個帶有此解決方案產生名稱標籤的 Transit Gateway 附件的範例。生成的名稱標籤的結構是<CIDR-range>-<AccountName>。

此解決方案使 CloudFormation用 AWS 部署 AWS Step Functions 工作流程，以管理跨所有已設定區域的 Transit Gateway 名稱標籤建立。工作流程會叫用執行基礎工作的 AWS Lambda 函數。

解決方案從 AWS Organizations 取得帳戶名稱後，Step Functions 狀態機器會取得所有 Transit Gateway 附件 ID。這些都是由 AWS 區域 parallel 處理。此處理包括查找每個附件的 CIDR 範圍。CIDR 範圍是透過搜尋「區域」內的「Transit Gateway」路由表，以取得相符的「Transit Gateway」附件 ID。如果所有必要的資訊都可以使用，解決方案會將「名稱」標籤套用至附件。解決方案不會覆寫任何現有的 Name 標籤。

該解決方案按照 Amazon EventBridge 事件控制的時間表執行。此事件會在世界標準時間每天早上 6:00 起始解決方案。

目標技術堆疊

- Amazon EventBridge
- AWS Lambda
- AWS Organizations
- AWS Transit Gateway
- Amazon Virtual Private Cloud (Amazon VPC)
- AWS X-Ray

目標架構

解決方案架構和工作流程如下圖所示。

1. 排程的事件會啟動規則。
2. EventBridge 規則會啟動「Step Functions」狀態機器。

3. 狀態機器會叫用 `tgw-tagger-organizations-account-query` Lambda 函數。
4. L `tgw-tagger-organizations-account-query` ambda 函數擔任組織管理帳戶中的角色。
5. `tgw-tagger-organizations-account-query`Lambda 函數會呼叫組織 API 以傳回 AWS 帳戶中繼資料。
6. 狀態機器會叫用 `tgw-tagger-attachment-query` Lambda 函數。
7. 對於每個區域，parallel 地，狀態機器會叫用 `tgw-tagger-rtb-query` Lambda 函數來讀取每個附件的 CIDR 範圍。
8. 對於每個區域，parallel 地，狀態機器調用 `tgw-tagger-attachment-tagger` Lambda 函數。
9. 系統會為共用網路帳戶中的 Transit Gateway 附件建立名稱標籤。

自動化和規模

解決方案會 parallel 處理每個區域，以減少執行的總持續時間。

工具

AWS 服務

- AWS CloudFormation — AWS CloudFormation 提供了一種將相關 AWS 和第三方資源集合建模的方法，透過將基礎設施視為程式碼來快速且一致地佈建，以及在整個生命週期中進行管理。
- Amazon EventBridge — Amazon EventBridge 是一種無伺服器事件匯流排服務，您可以使用它將應用程式與來自各種來源的資料連接起來。 EventBridge 接收事件、環境變更的指示器，並套用規則以將事件路由至目標。規則會根據事件結構 (稱為事件模式) 或排程，將事件與目標進行比對。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只會在需要時執行您的程式碼，並自動擴展，每天從幾個請求擴展到每秒數千個。您只需為使用的運算時間支付費用。程式碼未執行時無須付費。
- AWS Organizations — AWS Organizations 可協助您集中管理和管理您的環境，同時擴展和擴展 AWS 資源。使用 AWS Organizations Organization，您可以透過程式設計方式建立新的 AWS 帳戶和分配資源、將帳戶分組以組織工作流程、將政策套用至帳戶或群組以進行管理，以及針對所有帳戶使用單一付款方式簡化帳單。
- AWS Step Functions — AWS Step Functions 是一種低程式碼的視覺化工作流程服務，用於協調 AWS 服務、自動化商業程序以及建立無伺服器應用程式。工作流程可管理失敗、重試、平行化、服務整合和可觀察性，因此開發人員可以專注於更高價值的商務邏輯。
- AWS Transit Gateway — AWS Transit Gateway 透過中央中樞連接 VPC 和現場部署網路。這樣可以簡化您的網路，並結束複雜的對等關係。它充當雲路由器，因此每個新連接只有一次。

- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 是一種在您定義的邏輯隔離虛擬網路中啟動 AWS 資源的服務。
- AWS X-Ray — AWS X-Ray 會收集應用程式所提供請求的相關資料，並提供工具供您檢視、篩選和深入瞭解該資料，以識別問題和優化機會。

Code

此解決方案的原始程式碼可在 Transit Gateway 附件標籤器 GitHub 儲存庫中取得。存放庫包含下列檔案：

- `tgw-attachment-tagger-main-stack.yaml`會在共用網路帳戶中建立支援此解決方案的所有資源。
- `tgw-attachment-tagger-organizations-stack.yaml`會在組織的管理帳戶中建立角色。

史诗

部署主要解決方案堆疊

任務	描述	所需技能
收集必要的必要條件資訊	若要設定從 Lambda 函數到 AWS Organizations API 的跨帳戶存取，您需要組織管理帳戶的帳戶 ID。 注意：兩個 CloudFormation 堆棧的創建順序很重要。您必須先將資源部署到共用網路帳戶。在將資源部署到組織的管理帳戶之前，共用網路帳戶中的角色必須已經存在。如需詳細資訊，請參閱 AWS 文件。	DevOps 工程師
啟動主解決方案堆疊的 CloudFormation 範本。	主解決方案堆疊的範本將部署 IAM 角色、Step Functions 工作流程、Lambda 函數和 CloudWatch 事件。	DevOps 工程師

任務	描述	所需技能
	開啟共用聯網帳戶的 AWS 管理主控台，然後開啟主 CloudFormation 控台。 使用`tgw-attachment-tagger-main-stack.yaml` 範本和下列值建立堆疊： • 堆棧名稱 tgw-attachment-tagger-main - 堆棧 • awsOrganizationsRootAccountId— 組織管理帳戶的帳戶 ID • TGW Region 參數 — 解決方案的 AWS 區域，以逗號分隔的字串輸入 • TGWList 參數 — 要從解決方案中排除的傳輸閘道 ID，以逗號分隔的字串輸入 如需啟動 CloudFormation 堆疊的詳細資訊，請參閱 AWS 文件。	
確認解決方案已成功啟動。	等待 CloudFormation 堆疊達到「建立 _ 完成」狀態。這應該需要不到一分鐘的時間。 開啟 Step Functions 式主控台，並確認已使用名稱 tgw-attachment-tagger-state-machine 建立新的狀態機器。	DevOps 工程師

部署 AWS Organizations 堆疊

任務	描述	所需技能
收集必要的必要條件資訊	若要設定從 Lambda 函數到 AWS Organizations API 的跨帳戶存取，您需要共用聯網帳戶的帳戶 ID。	DevOps 工程師
啟動「Or CloudFormation ganizations」堆疊的範本	AWS Organizations 堆疊的範本會在組織的管理帳戶中部署 IAM 角色。 存取組織管理帳戶的 AWS 主控台。然後開啟 CloudFormation 主控台。 使用`tgw-attachment-tagger-organizations-stack.yaml` 範本和下列值建立堆疊： • 堆棧名稱 tgw-attachment-tagger-organizations - 堆棧 • NetworkingAccountId參數 — 共用網路帳戶的帳戶 ID 對於其他堆疊建立選項，請使用預設值。	DevOps 工程師
確認解決方案已成功啟動。	等待 CloudFormation 堆疊達到「建立 _ 完成」狀態。這應該需要不到一分鐘的時間。 開啟 Identity and Access Management (IAM) 主控台，並確認已使用名稱 tgw-attac	DevOps 工程師

任務	描述	所需技能
	hment-tagger-organization-q uer y-role 建立新角色。	

驗證解決方案

任務	描述	所需技能
運行狀態機。	開啟共用網路帳戶的 Step Functions 主控台，然後在導覽窗格中選擇 [狀態電腦]。 選擇狀態機 tgw-attachment-tagger-state-機器，然後選擇開始執行。 由於解決方案未使用此狀態機器的輸入，因此您可以使用預設值。 `{ "Comment": "Insert your JSON here" }` 選擇 Start Execution (開始執行)。	DevOps 工程師
觀看狀態機，直到完成。	在打開的新頁面上，您可以觀看狀態機運行。持續時間將取決於要處理的 Transit Gateway 附件數量。 在此頁面上，您可以檢查狀態機的每個步驟。您可以檢視狀態機器內的各種工作，並遵循 Lambda 函數的 CloudWatc	DevOps 工程師

任務	描述	所需技能
	h 記錄連結。對於在地圖中並行執行的工作，您可以使用索引下拉式清單來檢視每個區域的特定實作。	
驗證傳 Transit Gateway 附件標記。	開啟「共用網路」帳戶的 VPC 主控台，然後選擇「Transit Gateway 附件」。 在主控台上，會為符合條件的附件提供「名稱」標籤 (附件會傳播至 Transit Gateway 路由表，且資源擁有者是組織的成員)。	DevOps 工程師
驗證 CloudWatch 事件初始化。	等待 CloudWatch 事件啟動。這是定於世界標準時間 06:00 進行。 然後開啟共用網路帳戶的 Step Functions 主控台，並在導覽窗格中選擇 [狀態電腦]。 選擇狀態機 tgw-attachment-tagger-state-機器。確認解決方案是在世界標準時間 06:00 執行。	DevOps 工程師

相關資源

- AWS Organizations
- AWS Resource Access Manager
- 無伺服器傳輸網路協調器
- 建立 IAM 角色
- 在 AWS CloudFormation 主控台上建立堆疊

確認 ELB 負載平衡器需要 TLS 終止

創建者：普里揚卡喬達瑞 (AWS)

環境：生產	技術：網路；安全性、身分識別、合規	AWS 服務：Amazon CloudWatch 活動；Elastic Load Balancing (ELB)；AWS Lambda

Summary

在 Amazon Web Services (AWS) 雲端上，Elastic Load Balancing (ELB) 會自動將傳入的應用程式流量分配到多個目標，例如亞馬遜彈性運算雲端 (Amazon EC2) 執行個體、容器、IP 地址和 AWS Lambda 函數。負載平衡器使用接聽程式來定義負載平衡器用來接受使用者流量的連接埠和通訊協定。應用程式負載平衡器會在應用程式層做出路由決策，並使用 HTTP/HTTPS 通訊協定。傳統負載平衡器使用 TCP 或安全通訊端層 (SSL) 通訊協定，或在應用程式層使用 HTTP/HTTPS，在傳輸層做出路由決策。

此模式提供安全性控制，可針對應用程式負載平衡器和傳統負載平衡器檢查多個事件類型。叫用函數時，AWS Lambda 會檢查事件並確保負載平衡器符合規定。

函數會在下列 API 呼叫上啟 CloudWatch 動 Amazon 事件事件：CreateLoadBalancer、CreateLoadBalancerListeners、DeleteLoadBalancerListeners、CreateLoadBalar和ModifyListener。當事件偵測到其中一個 API 時，它會呼叫執行 Python 指令碼的 AWS Lambda。Python 指令碼會評估接聽程式是否包含 SSL 憑證，以及套用的原則是否使用傳輸層安全性 (TLS)。如果 SSL 政策被判定為 TLS 以外的任何內容，則該函數會將 Amazon Simple Notification Service (Amazon SNS) 通知傳送給使用者，其中包含相關資訊。

先決條件和限制

先決條件

- 有效的 AWS 帳戶

限制

- 除非對負載平衡器接聽程式進行更新，否則此安全控制不會檢查現有的負載平衡器。
- 這種安全控制是區域性的。您必須將其部署在要監控的每個 AWS 區域中。

架構

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

此模式包括下列附件：

- `ELBRequirestlstermination.zip`— 安全控制的 Lambda 程式碼。
- `ELBRequirestlstermination.yml`— 設定事件和 Lambda 函數的 CloudFormation 範本。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立 S3 儲存貯體來託管 Lambda 程式碼 .zip 檔案。此 S3 儲存貯體必須與您要評估的負載平衡器位於相同的 AWS 區域。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 (`ELBRequirestlstermination.zip` 檔案) 上傳至 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
啟動 AWS CloudFormation 範本。	在與 S3 儲存貯體相同的 CloudFormation AWS 區域中開啟 AWS 主控台，然後部署附加的範本`ELBRequirestlstermination.yml` 。如需部署 AWS CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的在	雲端架構師

任務	描述	所需技能
	AWS CloudFormation 主控台建立堆疊。	
完成範本中的參數。	當您啟動範本時，系統會提示您輸入下列資訊： • S3 儲存貯體：指定您在第一個史詩中建立或選取的儲存貯體。這是您上傳附加的 Lambda 程式碼 (`ELBRequirestlstermination.zip` 檔案) 的地方。 • S3 金鑰：指定 S3 儲存貯體中 Lambda .zip 檔案的位置 (例如，`ELBRequirestlstermination.zip` 或`controls/ELBRequirestlstermination.zip`)。請勿包含前導斜線。 • 通知電子郵件：提供您要接收 Amazon SNS 通知的作用中電子郵件地址。 • Lambda 記錄層級：指定 Lambda 函數的記錄層級和頻率。使用「資訊」(Info) 可記錄進度的詳細資訊訊息、針對仍允許部署繼續的錯誤事件發生錯誤，以及針對潛在有害情況發出警告。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱，才能開始接收違規通知。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- 什麼是 AWS Lambda？ (AWS Lambda 文件)
- 什麼是 Classic Load Balancer？ (ELB 文件)
- 什麼是 Application Load Balancer？ (ELB 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Splunk 檢視 AWS Network Firewall 日誌和指標

創建者伊沃平托

環境：PoC 或試點	技術：網路；雲端原生；內容傳遞；作業；安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：Amazon CloudWatch；Amazon CloudWatch 日誌；AWS Network Firewall		

Summary

許多組織使用 Splunk Enterprise 作為來自不同來源的日誌和指標的集中聚合和視覺化工具。 此模式可協助您使用適用於 AWS 的 Splunk 附加元件，將 Splunk 設定為從 Amazon CloudWatch 日誌擷取 AWS Network Firewall 日誌和指標。

若要達成此目標，您必須建立唯讀 AWS Identity and Access Management (IAM) 角色。適用於 AWS 的 Splunk 附加元件會使用此角色來存取。 CloudWatch您可以將 AWS 的 Splunk 附加元件設定為從中擷取指標和日誌。 CloudWatch最後，您可以從擷取的記錄資料和指標在 Splunk 中建立視覺效果。

先決條件和限制

先決條件

- 溢出帳戶
- Splunk 企業執行個體，8.2.2 版或更新版本
- 有效的 AWS 帳戶
- Network Firewall，設定並設定為將記錄檔傳送至記 CloudWatch 錄檔

限制

- Splunk 企業必須部署為 AWS 雲端中的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體叢集。

- AWS 中國區域不支援使用自動探索到的 Amazon EC2 IAM 角色來收集資料。

架構

此圖展示了以下要點：

1. Network Firewall 將記錄檔發佈到 CloudWatch 記錄檔
2. Splunk 企業從中檢索指標和日誌。 CloudWatch

為了在此架構中填入範例度量和記錄檔，工作負載會產生流量，這些流量會通過 Network Firewall 端點進入網際網路。這是通過使用路由表來實現的。 雖然此模式使用單一 Amazon EC2 執行個體做為工作負載，但只要 Network Firewall 設定為將日誌傳送到日誌，此模式就可以套用到 CloudWatch 任何架構。

此架構也會在另一個虛擬私有雲 (VPC) 中使用 Splunk 企業執行個體。但是，只要 Splunk 執行個體可以存取 API，就可以位於其他位置，例如與工作負載位於相同的 VPC 中。 CloudWatch

工具

AWS 服務

- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Network Firewall 是適用於 AWS 雲端中 VPC 的可設定狀態、受管網路防火牆以及入侵偵測與防護服務。

其他工具

- Splunk 可協助您監視、視覺化和分析記錄資料。

史诗

建立 IAM 角色

任務	描述	所需技能
建立 IAM 政策。	遵循使用 JSON 編輯器建立政策中的指示來建立 IAM 政策，以授予對 CloudWatch 記錄資料和指 CloudWatch 標的唯讀存取權限。將以下 政策貼到 JSON 編輯器。 `{ "Statement": [ { "Action": [ "cloudwatch:List*", "cloudwatch:Get*", "network-firewall:List*", "logs:Describe*", "logs:Get*", "logs:List*", "logs:StartQuery", "logs:StopQuery", "logs:TestMetricFilter", "logs:FilterLogEvents",`	AWS 管理員

任務	描述	所需技能
	<pre>"network-firewall: Describe*"], "Effect": "Allow", "Resource": "*" }], "Version": "2012-10-17" }</pre>	
建立新的 IAM 角色。	遵循建立角色以將許可委派給 AWS 服務中的指示，建立適用於 AWS 的 Splunk 附加元件用來存取的 IAM 角色。CloudWatch對於 [權限] 原則，請選擇您先前建立的原則。	AWS 管理員
將 IAM 角色指派給 Splunk 叢集中的 EC2 執行個體。	1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。 2. 在導覽窗格中，選擇執行個體。 3. 選取 Splunk 叢集中的 EC2 執行個體。 4. 選擇動作、安全性，然後選擇修改 IAM 角色。 5. 選取您先前建立的 IAM 角色，然後選擇 [儲存]。	AWS 管理員

安裝適用於 AWS 的溢出器附加元件

任務	描述	所需技能
安裝附加元件。	1. 在 Splunk 儀表板中，導航到 Spl unk 應用程序。 2. 搜索 Splunk 附加組件 Amazon Web Services。 3. 選擇 Install (安裝)。 4. 提供您的 Splunk 憑證。	潑濺管理員
設定 AWS 登入資料。	1. 在 Splunk 儀表板中，瀏覽至適用於 AWS 的 Splunk 附加元件。 2. 選擇 Configuration (組態)。 3. 在「自動探索的 IAM 角色」欄中，選取您先前建立的 IAM 角色。 如需詳細資訊，請參閱 Splunk 說明文件中的尋找 Splunk 平台執行個體中的 IAM 角色。	潑濺管理員

將 Splunk 存取權設定為 CloudWatch

任務	描述	所需技能
設定從記錄檔擷取 Network Firewall 記 CloudWatch 錄檔。	1. 在 Splunk 儀表板中，瀏覽至適用於 AWS 的 Splunk 附加元件。 2. 選擇「輸入」。 3. 選擇「建立新輸入」。	潑濺管理員

任務	描述	所需技能
	4. 在清單中，選擇 [自訂資料類型]，然後選擇 [CloudWatch 記錄檔]。 5. 為您的 Network Firewall 日誌提供名稱、AWS 帳戶、AWS 區域和日誌群組。 6. 選擇儲存。 依預設，Splunk 會每隔 10 分鐘擷取一次記錄資料。這是「進階設定」下的可設定參數。如需詳細資訊，請參閱 Splunk 說明文件中的使用 Splunk 網頁設定 CloudWatch 記錄輸入。	

任務	描述	所需技能
設定從中擷取 Network Firewall 度量 CloudWatch。	1. 在 Splunk 儀表板中，瀏覽至適用於 AWS 的 Splunk 附加元件。 2. 選擇「輸入」。 3. 選擇「建立新輸入」。 4. 在清單中，選擇CloudWatch。 5. 為您的 Network Firewall 指標提供名稱、AWS 帳戶和 AWS 區域。 6. 在「測量結果組態」旁邊，選擇進階模式中的編輯。 7. (選擇性) 刪除所有預先設定的命名空間。 8. 選擇「新增命名空間」，然後將其命名為 AWS/NetworkFirewall。 9. 在維度值中，新增下列項目。 `[{"AvailabilityZone":[".*"],"Engine":[".*"],"FirewallName":[".*"]}]` 10針對量度，選擇全部。 11.在「測量結果統計值」中，選擇總 12選擇確定。 13選擇儲存。 依預設，Splunk 會每 5 分鐘擷取一次指標資料。這是「進階	潑濺管理員

任務	描述	所需技能
	設定」下的可設定參數。如需詳細資訊，請參閱 Splunk 說明文件中的使用 Splunk 網路設定 CloudWatch 輸入。	

使用查詢建立 Splunk 視覺效果

任務	描述	所需技能
檢視頂端來源 IP 位址。	1. 在 Splunk 儀表板中，導航到搜索和報告。 2. 在「在此輸入搜尋」方塊中，輸入以下內容。 `sourcetype="aws:cloudwatchlogs" | top event.src_ip` 此查詢會以遞減順序顯示流量最多的來源 IP 位址表格。 3. 對於圖形表示，請選擇「視覺化」。	潑濺管理員
檢視封包統計資料。	1. 在 Splunk 儀表板中，導航到搜索和報告。 2. 在「在此輸入搜尋」方塊中，輸入以下內容。 `sourcetype="aws:cloudwatch"| timechart sum(Sum) by metric_name` 此查詢會顯示測量結果DroppedPa	潑濺管理員

任務	描述	所需技能
	ckets PassedPac kets 、和ReceivedP ackets 每分鐘的表格。 3. 對於圖形表示，請選擇「視覺化」。	
檢視最常用的來源連接埠。	1. 在 Splunk 儀表板中，導航到搜索和報告。 2. 在「在此輸入搜尋」方塊中，輸入以下內容。 `sourcetype="aws:cloudwatchlogs" | top event.dest_port` 此查詢會以遞減順序顯示流量最多的來源連接埠表格。 3. 對於圖形表示，請選擇「視覺化」。	潑濺管理員

相關資源

AWS 文件

- 建立角色以將許可委派給 AWS 服務 (IAM 文件)
- 建立 IAM 政策 (IAM 文件)
- AWS Network Firewall 中的記錄和監控 (Network Firewall 文件)
- AWS Network Firewall 的路由表組態 (Network Firewall 文件)

AWS 部落格文章

- AWS Network Firewall 部署模型

AWS Marketplace

- 潑濺企業 Amazon 機器映像（AMI）

更多模式

- 使用工作階段管理員和 Amazon EC2 執行個體 Connect 存取防禦主機
- 使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用 AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- ???
- 檢查 IPv4 和 IPv6 的安全群組輸入規則中是否有單一主機網路項目
- 使用 AWS Network Firewall 和 AWS Transit Gateway 部署防火牆
- 使用私有端點和應用 Application Load Balancer 在內部網站上部署 Amazon API Gateway API
- 使用 AWS Config 為公有子網路部署偵探屬性型存取控制
- ???
- 在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線
- 使用 AWS 傳輸閘道 Connect 將 VRF 延伸至 AWS
- 將 F5 大 IP 工作負載遷移到 AWS 雲端上的 F5 大 IP VE
- 在多帳戶 VPC 設計中保留非工作負載子網路的可路由 IP 空間
- 使用服務控制策略防止在帳戶層級存取網際網路
- 將提醒從 AWS Network Firewall 傳送到 Slack 通道
- 使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront
- EnterpriseOne 使用 AWS 彈性災難復原為 Oracle JD 愛德華設定災難復原
- 在多帳戶 AWS 環境中為混合網路設定 DNS 解析
- 使用 BMC 探索查詢擷取移轉資料以進行移轉規劃
- 使用 Network Firewall 從輸出流量的伺服器名稱指示 (SNI) 擷取 DNS 網域名稱

作業系統

主題

- 使用 AWS MGN 將 RHEL BYOL 系統遷移到 AWS 包含授權的執行個體
- 將 Microsoft SQL 伺服器遷移到 AWS 雲端後解決連接錯誤
- 更多模式

使用 AWS MGN 將 RHEL BYOL 系統遷移到 AWS 包含授權的執行個體

由邁克·庫茲涅佐夫（AWS）創建

環境：生產	來源：RHEL BYOL 執行個體 (位於內部部署或任何其他雲端環境)	目標：包含 AWS 授權的 RHEL 執行個體
R 類型：重新主機	工作負載：所有其他工作	技術：作業系統、基礎架構、移轉
AWS 服務：AWS 應用程式遷移服務		

Summary

當您使用 AWS 應用程式遷移服務 (AWS MGN) 將工作負載遷移到 AWS 時，您可能需要在遷移期間將 RHEL (RHEL) 執行個體移轉 (重新託管)，並將授權從預設的自攜授權 (BYOL) 模式變更為 AWS 包含授權 (LI) 模型。AWS MGN 支援使用 Amazon 機器映像 (AMI) ID 的可擴展方法。此模式描述如何在大規模重新裝載移轉期間在 RHEL 伺服器上完成授權變更。同時也說明如何變更已在 Amazon Elastic Compute Cloud (Amazon EC2) 上執行的 RHEL 系統的授權。

先決條件和限制

先決條件

- 存取目標 AWS 帳戶
- AWS MGN 在目標 AWS 帳戶和區域中初始化以進行遷移 (如果您已從現場部署系統遷移到 AWS，則不需要)
- 具有有效 RHEL 授權的來源 RHEL 伺服器

架構

此模式涵蓋兩種情況：

- 使用 AWS MGN 將系統從現場部署直接遷移到 AWS LI 執行個體。對於這種情況，請按照第一個史詩（遷移到 LI 實例-選項 1）和第三個史詩中的說明進行操作。
- 針對先前已在 Amazon EC2 上執行的 RHEL 系統，將授權模式從 BYOL 變更為 LI。對於這種情況，請按照第二個史詩（遷移到 LI 實例-選項 2）和第三史詩中的說明進行操作。

注意：第三個史詩涉及重新設定新的 RHEL 執行個體，以使用 AWS 提供的 Red Hat 更新基礎設施 (RHUI) 伺服器。這兩種情況的程序都是相同的。

工具

AWS 服務

- AWS 應用程式遷移服務 (AWS MGN) 可協助您將應用程式重新託管 (升降和轉移) 到 AWS 雲端，而且停機時間最短。

史诗

遷移至 LI 執行個體-選項 1 (適用於內部部署 RHEL 系統)

任務	描述	所需技能
尋找目標區域中 RHEL AWS LI 執行個體的 AMI 識別碼。	瀏覽 AWS Marketplace 或使用 Amazon EC2 主控台尋找符合 RHEL 來源系統版本 (例如 RHEL 7.7) 的 RHEL AMI ID，然後記下 AMI ID。在 Amazon EC2 主控台上，您可以使用下列其中一個搜尋詞篩選 AMI： • 說明 = 由紅帽股份有限公司提供 • AMI 名稱 = 雷赫爾 -7.7	雲端管理員
設定 AWS MGN 啟動設定。	1. 在 AWS MGN 主控台上，新增來源 RHEL 系統：安裝 AWS 複寫代理程式並按照	雲端管理員

任務	描述	所需技能
	AWS MGN 文件中的指示新增來源伺服器。 2. 在 [來源伺服器] 頁面上，選擇來源 RHEL 系統，然後選擇 [啟動設定] 索引標籤。 3. 在 [一般啟動設定] 區段中，選擇 [編輯]。若要停用自動選取並手動指定目標例項類型，請將 [執行個體類型正確大小] 變更為 [無]，然後選擇 [儲存設定]。這可讓您使用在 Amazon EC2 啟動範本中設定的執行個體類型。如需詳細資訊，請參閱 AWS MGN 文件。 4. 在 [EC2 啟動範本] 區段中，選擇 [修改]。在 [關於修改 EC2 啟動範本] 對話方塊中，再次選擇 [修改]。這會開啟 Amazon EC2 主控台，以便您變更此執行個體的範本。 5. 請參閱 AWS MGN 文件中的主要考量事項。 注意：您可以忽略警告，以防選擇自己的 AMI。 6. 在 Amazon EC2 主控台的新啟動範本中，修改下列項目： • 對於 AMI，請指定您先前識別的 AMI 識別碼，或搜尋 RHEL-x 並指定您需	

任務	描述	所需技能
	要的版本 (例如，RHEL-7.7)。 • 對於「例證」類型，設定所需的目標例證類型。 • 保持下列區段不變：金鑰配對 (登入)、網路設定 (除非您要指定目標子網路和安全性群組)、儲存區、資源標籤 (除非您要新增或修改任何標籤)。 • (選擇性) 在進階詳細資料區段中，指定 IAM 執行個體設定檔角色 (如果 AWS Systems Manager future 需要管理)。 7. 選擇 [建立範本版本]，然後選擇成功訊息中的連結以檢視啟動範本。 8. 選擇動作、設定預設版本。在 [範本版本] 中，選取最新版本 (新系統的第 2 版)，然後選擇 [設定為預設版本]。 AWS MGN 現在將使用此版本的啟動範本啟動測試或切換執行個體。如需詳細資訊，請參閱 AWS MGN 文件。	

任務	描述	所需技能
驗證設定。	1. 在 AWS MGN 主控台的 [來源伺服器] 頁面上，選擇您的來源伺服器，然後選擇 [啟動設定] 索引標籤。 2. 在 EC2 啟動範本區段中，確認執行個體類型、子網路和安全群組參數設定正確。 注意：此區段不會顯示您選取的 AMI ID。若要查看 ID，您可以開啟 Amazon EC2 主控台、啟動範本檢視，然後搜尋本節中顯示的範本 ID。	雲端管理員

任務	描述	所需技能
啟動新的 LI 執行個體。	1. 初始同步完成時，AWS MGN 主控台來源伺服器頁面上伺服器的移轉生命週期欄會變更為 [準備測試]。若要啟動新的測試執行個體，請選擇您的來源伺服器，開啟 [測試和切換] 功能表，然後選擇 [啟動測試執行個體]。選擇檢視工作詳細資訊以監督啟動工作的狀態。如需詳細資訊，請參閱 AWS MGN 文件。 2. 等待啟動任務完成，然後開啟啟動的 EC2 執行個體詳細資料頁面。選擇「詳細資訊」頁籤，然後確認「執行處理詳細資訊」段落包含下列 • 平台詳細資料：「紅帽企業 Linux」 • AMI 名稱：您在 EC2 啟動範本中指定的 AMI 名稱 3. 按照 AWS MGN 文件中的指示，切換至新的 LI 執行個體。 4. 遵循上一個史詩中的步驟，重新設定新執行個體以使用 AWS 提供的 RHUI 伺服器。	雲端管理員

遷移至 LI 執行個體-選項 2 (適用於 RHEL 自攜 EC2 執行個體)

任務	描述	所需技能
將您的 RHEL 自攜 EC2 執行個體遷移到 AWS LI 執行個體。	您可以透過移動磁碟 (Amazon 彈性區塊存放區磁碟區) 並將它們附加到新的 LI 執行個體，將先前以 BYOL 形式遷移到 AWS LI 執行個體的 RHEL 系統切換到 AWS LI 執行個體。若要進行此切換，請依照下列步驟執行： 1. 從 RHEL LI AMI 啟動新的目標 RHEL 執行個體。確保您選擇的 AMI： • 使用與目前 RHEL 執行個體相同的 RHEL 版本。 • 與您目前的 RHEL 執行個體具有相同的開機程序 (BIOS 或 UEFI)。例如，如果來源伺服器是以 BIOS 為基礎，請使用 AWS Marketplace RHEL AMI (同時也以 BIOS 為基礎)；如果是以 UEFI 為基礎的系統，請選擇以 UEFI 為基礎的 AMI。 2. 停止這兩個執行個體：新的 LI 執行個體和原始來源執行個體。 3. 將所有 EBS 磁碟區 (包括根磁碟) 從新的 LI 執行個體卸離並刪除它們。 4. 將所有 EBS 磁碟區 (包括根磁碟) 與舊來源執行個體分	雲端管理員

任務	描述	所需技能
	離，並將它們附加至新的 LI 執行個體。保持磁碟區與裝置的相同對應。(例如，先前連接至磁碟/dev/sda機的 EBS 磁碟區必須與新執行個體一樣/dev/sda連接。) 5. 刪除來源 (現在無磁碟) 執行個體。 6. 啟動新的 LI 執行個體。請依照下一個史詩中的步驟，登入執行個體並重新設定執行個體，以使用 AWS 提供的 RHUI 伺服器。	

重新設定 RHEL 作業系統以使用 AWS 提供的 RHUI — 這兩種選項

任務	描述	所需技能
從 Red Hat 訂閱與授權取消註冊作業系統。	在移轉和成功切換之後，RHEL 系統必須從 Red Hat 訂閱中移除，以停止使用 Red Hat 授權並避免重複計費。 若要從 Red Hat 訂閱中移除 RHEL 作業系統，請依照 Red Hat 訂閱管理 (RHSM) 說明文件中所述的程序進行。使用 CLI 命令： `subscription-manager unregister` 您也可以停用訂閱管理員外掛程式，停止在每次	Linux 或系統管理員

任務	描述	所需技能
	yum 通話中檢查訂閱的狀態。若要執行此操作，請編輯組態檔案/etc/yum/pluginconf.d/subscription-manager.conf 並enabled=1 將參數變更為enabled=0 。	

任務	描述	所需技能
將舊的更新設定 (RHUI、RHEL 衛星伺服器網路、yum 儲存庫) 取代為 AWS 提供的 RHUI。	您必須重新設定已移轉的 RHEL 系統，才能使用 AWS 提供的 RHUI 伺服器。這可讓您存取 AWS 區域內的 RHUI 伺服器，而不需要外部更新基礎設施。「變更」涉及以下過程： 1. 備份現有的 yum 設定。 2. 移除舊的 RHUI (yum 儲存庫) 組態和套件。 3. 新增 AWS 提供的新 RHUI 組態和憑證套件。您必須從 AWS 上的另一個 RHEL 執行個體擷取這些檔案，因為這些組態套件只能在 AWS 提供的 RHUI 伺服器上使用。 以下是詳細的步驟和命令： 1. 將所有資料夾和`/etc/pki/*` 資料夾複製到備份位置，以備份現有的 yum 設定`/etc/yum*` 和憑證。例如： `mkdir yum-backup` `cp -ra /etc/yum* /etc/pki ./yum-backup` `tar czf yum-backup.tgz ./yum-backup` 2. 移除舊的 RHUI 組態和套件：	Linux 或系統管理員

任務	描述	所需技能
	a. 查找所有已安裝的 RHUI 軟件包： `sudo rpm -qa | grep rhui` b. 刪除這些套件： `sudo yum remove $(rpm -qa | grep rhui)` c. 刪除該`/etc/yum/vars/releasever` 文件（如果存在）。 3. 新增 AWS 提供的新 RHUI 和憑證套件。您必須從 AWS 上的另一個 RHEL 執行個體擷取這些項目。有幾種方式可以執行此作業。例如，您可以依照 Red Hat 知識庫文章中提供的指示進行： a. 從 AWS Marketplace 啟動另一個 RHEL (RHEL-EC2) 執行個體。 b. 從此執行個體下載兩個套件：最新的 RHUI 用戶端組態套件和憑證授權單位 (CA) 憑證。例如，從桌面執行此命令： `ssh RHEL-EC2 "sudo yumdownloader ca-certificates rh-`	

任務	描述	所需技能
	amazon-rhui-cli ent" c. 將套件從 RHEL-EC2 執行個體複製到新的移轉系統。例如： `scp RHEL-EC2:rh-amazon-rhui-client\* RHEL-EC2:ca-certificates\* .` `ssh <migrated-instance> "mkdir /tmp/amazon"` `scp rh-amazon-rhui-client* ca-certificates* <migrated-instance>:/tmp/amazon` d. 在移轉的執行個體上安裝新的 RHUI 和 CA 組態套件： `ssh <migrated-instance> "sudo rpm -Uhv /tmp/amazon/*"`	
驗證組態。	在目標移轉的執行個體上，確認新組態是否正確： `sudo yum clean all` `sudo yum repolist`	Linux 或系統管理員

相關資源

- AWS 應用程式遷移服務 (AWS MGN) 使用者指南
- 取得支援 IMDSv2 的 AWS RHUI 用戶端套件 (紅帽知識庫文章)
- Amazon EC2 啟動範本 (Amazon EC2 文件)

將 Microsoft SQL 伺服器遷移到 AWS 雲端後解決連接錯誤

創建者：普雷姆庫瑪切拉杜賴 (AWS)

環境：生產	技術：作業系統；移轉	工作量：Microsoft
AWS 服務：Amazon EC2		

Summary

將在 Windows 伺服器 2008 R2、2012 或 2012 R2 上執行的 Microsoft SQL 伺服器遷移到 Amazon 網路服務 (AWS) 雲端上的亞馬遜彈性運算雲端 (亞馬遜 EC2) 執行個體後，與 SQL 伺服器的連線失敗，並出現以下錯誤：

- `[Microsoft][ODBC SQL Server Driver][DBNETLIB] General Network error`
- `ERROR [08S01] [Microsoft][SQL Native Client]Communication link failure. System.Data.SqlClient.SqlException: A transport-level error has occurred when sending the request to the server. (provider: TCP Provider, error: 0 - An existing connection was forcibly closed by the remote host.)`
- `TCP Provider: The semaphore timeout period has expired`

此模式說明如何解決這些錯誤，方法是關閉作業系統 (作業系統) 和網路介面層級的 Windows 可延展網路套件 (SNP) 功能，以及在 Windows 伺服器 2008 R2 上執行的 SQL Server。

先決條件和限制

先決條件

- 視窗伺服器的管理員權限。
- 如果您使用 AWS 應用程式遷移服務做為移轉工具，則需要下列其中一個 Windows 伺服器版本：
  - 視窗伺服器 2008 R2 服務包 1、2012 年或 2012 年 R2
- 如果您使用 CloudEndure 移轉作為移轉工具，則需要下列其中一個 Windows Server 版本：
  - 視窗伺服器 2003 R2 服務套件 3、2008 年 R2 服務套件 1、2012 或 2012 年 R2

工具

- Amazon EC2 — 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。
- Windows 伺服器 — Windows 伺服器是建立連線應用程式、網路和網路服務之基礎結構的平台。

史诗

在作業系統和 elastic network interface 層級關閉 SNP 功能

任務	描述	所需技能
在作業系統層級關閉 SNP 功能。	1. 以系統管理員身分登入 Windows 伺服器並開啟命令提示字元。 2. 執行 `netsh int tcp show global` 命令。 3. 在輸出中，檢查`Chimney Offload`是否處於模式`Receive-Side Scaling`或`enabled`模式。如果其中任何一個是`enabled`，請執行下列命令： • `netsh int tcp set global chimney=disabled` • `netsh int tcp set global rss=disabled`	AWS 管理員、AWS 系統管理員、遷移工程師、雲端管理員
在 elastic network interface 層級關閉 SNP 功能。	1. 選擇 [開始]`ncpa.cpl`，輸入，然後按 Enter。 2. 以滑鼠右鍵按一下「彈性	AWS 管理員、雲端管理員、AWS 系統管理員

任務	描述	所需技能
	3. 在躍現式選單中，選擇「內容」。 4. 在「乙太網路介面卡內容」視窗中選擇「設定」 5. 在 Amazon 彈性網路介面卡內容快顯視窗中，選擇進階索引標籤。 6. 在「性質」區段中，關閉所有卸載和 RSS。	

相關資源

- 如何疑難排解 RSS 和 NetDMA 等進階網路效能功能

更多模式

- 在 AWS 雲端上的 Sun SPARC 伺服器備份 Sun 字元 SSP 模擬器
- ???
- 使用原生備份和還原方法將現場部署 Microsoft SQL 伺服器資料庫遷移到亞馬遜 RDS
- 透過高可用性災難復原將適用於 LUW 的 Db2 移轉至 Amazon EC2
- 使用 AWS 服務監控 SAP RHEL 起搏器叢集
- ???
- 重新啟動 RHEL 來源伺服器後，自動重新啟動 AWS 複寫代理程式而不停用 SELinux

作業

主題

- 使用 Python 在 AMS 中自動創建一個 RFC
- 為雲端作業模式建立 RACI 或 RASCI 矩陣
- 建立使用具有預設加密功能的 Amazon EBS 磁碟區的 AWS Cloud9 IDE
- 自動建立基於標籤的 Amazon CloudWatch 儀表板
- 使用 AWS Config 進階查詢，根據 AWS 資源的建立日期尋找 AWS 資源
- 檢視 AWS 帳戶或組織的 EBS 快照詳細資訊
- 更多模式

使用 Python 在 AMS 中自動創建一個 RFC

創建者：納納斯卡蘭凱拉森 (AWS)

環境：生產	技術：作業；雲端原生	AWS 服務：AWS Managed Services

Summary

AWS Managed Services (AMS) 提供 Amazon Web 服務 (AWS) 基礎設施的持續管理，協助您更有效率且安全地操作雲端基礎設施。若要對受管理環境進行變更，您必須建立並提交新的變更要求 (RFC)，其中包含特定作業或動作的變更類型 (CT) ID。

不過，手動建立 RFC 大約需要五分鐘，組織中的團隊可能需要每天提交多份 RFC。此模式可協助您將 RFC 建立程序自動化、縮短每個 RFC 的建立時間，並消除手動錯誤。

此模式說明如何使用 Python 程式碼自動建立 `Stop EC2 instance` RFC，以停止 AMS 帳戶中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體。然後，您可以將此模式的方法和 Python 自動化應用於其他 RFC 類型。

先決條件和限制

先決條件

- 一個 AMS 高級帳戶。如需詳細資訊，請參閱 AWS Managed Services 文件中的 AMS 操作計劃。
- 您的 AMS 帳戶中至少有一個現有 EC2 執行個體。
- 瞭解如何在 AMS 中建立和提交 RFC。
- 熟悉 Python。

限制

- 您只能在 AMS 帳戶中使用 RFC 進行變更。您的 AWS 帳戶使用不同的程序進行類似的變更。

架構

技術堆疊

- AMS
- AWS 命令列界面 (AWS CLI)
- 適用於 Python 的 AWS SDK (Boto3)
- Python 及其所需的軟件包（JSON 和博圖 3）

自動化和規模

此模式提供了用於自動執行 `Stop EC2 instance` RFC 的範例程式碼，但您可以將此模式的範例程式碼和方法用於其他 RFC。

工具

- AWS Managed Services — AMS 可協助您更有效率且安全地操作 AWS 基礎設施。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是管理 AWS 服務的統一工具。在 AMS 中，變更管理 API 提供建立和管理 RFC 的作業。
- 適用於 Python 的 AWS 開發套件 (Boto3) — 適用於 Python 的開發套件可讓您輕鬆將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。

Code

該`AMS Stop EC2 Instance.zip`文件（附件）包含用於創建一個 `Stop EC2 instance` RFC 的 Python 代碼。您也可以設定此程式碼，為多個 EC2 執行個體提交單一 RFC。

史诗

選項 1 — 設定適用於 macOS 或 Linux 的環境

任務	描述	所需技能
安裝並驗證 Python。	1. 開啟終端機視窗並執行`brew install python3`指令。 2. 透過執行`python --version` 指令來驗證 Python 是否已正確安裝。	AWS 系統管理員

任務	描述	所需技能
	3. 透過執行`pip --version`命令驗證pip是否已正確安裝。	
安裝 AWS CLI。	執行`pip install awscli --upgrade –user`命令以安裝 AWS CLI 。	AWS 系統管理員
安裝肉毒桿 3.	執行命`pip install boto3`令以安裝 Boto3。	AWS 系統管理員
安裝 JSON。	執行`pip install json`命令以安裝 JSON。	AWS 系統管理員
設定 CLI 碼管理系統。	登入 AWS 管理主控台，開啟 AMS 主控台，然後選擇 [文件]。下載包含 AMS CLI 的 .zip 檔案，將其解壓縮，然後將其安裝在本機電腦上。 安裝 AMS CLI 之後，請執行`aws amscm help`命令。輸出提供有關 AMS 變更管理程序的資訊。	AWS 系統管理員

選項 2 — 設置視窗環境

任務	描述	所需技能
安裝並驗證 Python。	1. 開啟適用於視窗的 Python 版本頁面，下載最新版本，然後安裝 Python。 2. 透過執行`python --version`指令來驗證 Python 是否已正確安裝。	AWS 系統管理員

任務	描述	所需技能
	3. 透過執行`pip --version`命令驗證pip是否已正確安裝。	
安裝 AWS CLI。	執行`pip install awscli --upgrade –user`命令以安裝 AWS CLI。	AWS 系統管理員
安裝肉毒桿 3.	執行命`pip install boto3`令以安裝 Boto3。	AWS 系統管理員
安裝 JSON。	執行`pip install json`命令以安裝 JSON。	AWS 系統管理員
設定 CLI 碼管理系統。	登入 AWS 管理主控台，開啟 AMS 主控台，然後選擇 [文件]。下載包含 AMS CLI 的 .zip 檔案，將其解壓縮，然後將其安裝在本機電腦上。 安裝 AMS CLI 之後，請執行`aws amscm help`命令。輸出提供有關 AMS 變更管理程序的資訊	AWS 系統管理員

擷取 RFC 的 CT ID 和執行參數

任務	描述	所需技能
擷取 RFC 的 CT ID、版本和執行參數。	每個 RFC 都有不同的 CT ID、版本和執行參數。您可以使用下列其中一個選項來擷取此資訊： 1. 使用 AWS Managed Services 文件中 RFC 使用	AWS 系統管理員

任務	描述	所需技能
	範例中的 CLI 一節，按照尋找變更請求 (RFC) 中的說明進行操作。 2. 通過 AMS 控制台打開類似類型的現有 RFC 或創建新的 RFC 作為測試。使用 RFC 的 CT 識別碼和執行參數。如需詳細資訊，請參閱 AWS Managed Services 文件中的使用主控台尋找 RFC。 注意：要為其他 RFC 調整此模式的 Python 自動化，請將 `ams_stop_ec2_instance` Python 代碼文件中的 CT 類型和參數值替換為您提取的`AMS Stop EC2 Instance.zip` 文件（附件）中的 CT 類型和參數值。	

執行自 Python 化

任務	描述	所需技能
執行 Python 自動化。	1. 將`AMS Stop EC2 Instance.zip` 檔案 (附加) 下載到您的本機電腦並解壓縮檔案。 2. `input_instances` 使用 EC2 執行個體資訊進行更新。	AWS 系統管理員

任務	描述	所需技能
	3. 開啟終端機並導覽至擷取程式碼的路徑 4. 執行 `pythonams_stop_ec2_instance.py` 命令。	

相關資源

- 什麼是變更類型？
- CLI 教學課程：高可用性雙層堆疊 (Linux/RHEL)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

為雲端作業模式建立 RACI 或 RASCI 矩陣

創建者：泰迪傑梅德（AWS），杰羅姆·德克雷（AWS），若瑟林 LE 米諾（AWS）和弗洛里安·勒魯（AWS）

環境：生產	技術：營運；管理與治理

Summary

Cloud Center of Excellence (CCoE) 或 CEE (雲端啟用引擎) 是一個強大且負責任的團隊，專注於雲端的營運準備。他們的主要焦點是將資訊 IT 組織從內部部署作業模式轉變為雲端作業模式。CCoE 應該是跨職能團隊，其中包括來自基礎結構、應用程式、作業和安全性的代表性。

雲端作業模式的其中一個關鍵元件是 R ACI 矩陣或 RASCI 矩陣。這是用來定義參與移轉活動和雲端作業的所有當事人的角色和責任。矩陣名稱衍生自矩陣中定義的責任類型：負責 (R)、負責 (A)、支援 (S)、諮詢 (C) 和通知 (I)。支持類型是可選的。如果您包含它，則稱為 RASCI 矩陣，如果您將其排除，則稱為 R ACI 矩陣。

從附加的範本開始，您的 CCoE 團隊可以為您的組織建立 RACI 或 RASCI 矩陣。範本包含雲端作業模式中常見的專案團隊、角色和工作。此矩陣的基礎是與作業整合和 CCoE 功能相關的工作。不過，您可以自訂此範本以符合組織結構和使用案例的需求。

RACI 矩陣的實施沒有任何限制。這種方法適用於大型組織，初創企業以及介於兩者之間的所有內容。對於小型組織，相同的資源可以填寫多個角色。

史诗

建立矩陣

任務	描述	所需技能
識別關鍵利益相關者	識別與雲端作業模型策略目標相關聯的關鍵服務和團隊經理。	專案經理
自訂矩陣範本。	在「附件」區段下載範本，然後更新 RACI 或 RASCI 矩陣，如下所示：	專案經理

任務	描述	所需技能
	• 在 Cloud Teams 工作表上，視需要更新組織的 CCoE 串流名稱、團隊名稱和團隊說明。 • 在 [雲端角色] 工作表上，視需要更新組織的角色、小組名稱和角色描述。 • 在 RASCI 工作表上，視組織需要更新下列項目： • 在列 1 和欄 A 中，更新 CCoE 串流。 • 在列 2 中，更新群組名稱。 • 在列 3 中，更新角色名稱。 • 在欄 D 和 E 中，更新您要包含在 RASCI 圖表中的一般欄位和活動。	
計劃會議。	1. 向所有利益相關者傳達 RASCI 目標。 2. 計劃一個或多個會議，以便每個團隊的授權代表可以參加。	專案經理

任務	描述	所需技能
完成矩陣。	在與所有利益相關者的會議中，執行以下操作： 1. 確認每個團隊的代表在場。小組參與是必要的，因此您可以針對每個作業精確地指定職責型態。 2. 與參與者一起回顧 RASCI 矩陣是什麼以及目標。 3. 與參與者檢閱共同的責任模型，讓他們瞭解其組織對雲端安全性的責任範圍。 4. 在 RASCI 工作表上，針對每個作業或作業，完成欄位 F 到 A，以指定下列職責型態： • 負責 (R) — 此角色負責執行工作以完成任務。 • 負責 (A) — 此角色負責確保任務已完成。此角色也負責確保符合先決條件，並將任務委派給負責人。 • Sup@@ port (S) — 此角色可協助負責人完成任務。此職責型態是選擇性的，您可以選擇將其排除，以建立更傳統的 RACI 矩陣。 • 諮詢 (C) — 對於任務的意見或專業知識，應諮詢此角色。視作業而定，可能不需要此職責型態。	專案經理

任務	描述	所需技能
	• 通知 (I) — 此角色應該保持最新的任務進度，並在任務完成時通知。 • 空白 — 此角色不參與活動或任務。	
分享 RASCI 矩陣。	當 RACI 或 RASCI 矩陣完成時，請領導批准它。將其保存在共享存儲庫或中央位置，所有利益相關者都可以訪問它。我們建議您使用標準文件控制處理來記錄並核准矩陣的修訂。	專案經理

相關資源

- AWS 共同的責任模型

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

建立使用具有預設加密功能的 Amazon EBS 磁碟區的 AWS Cloud9 IDE

由賈納爾汗·馬亞拉（AWS）和德魯巴約提穆克吉（AWS）創建

環境：生產	技術：營運	工作負載：所有其他工作
AWS 服務：AWS Cloud9；AWS KMS		

Summary

依預設，您可以使用加密，在亞馬遜網路服務 (AWS) 雲端上強制執行 Amazon 彈性區塊存放區 (Amazon EBS) 磁碟區和快照複本的加密。

您可以建立 AWS Cloud9 整合式開發環境 (IDE)，該環境使用預設情況下加密的 EBS 磁碟區。不過，AWS Cloud9 的 AWS 身分與存取管理 (IAM) 服務連結角色需要存取這些 EBS 磁碟區的 AWS Key Management Service (AWS KMS) 金鑰。如果未提供存取權，AWS Cloud9 IDE 可能會無法啟動，而且偵錯可能會很困難。

此模式提供了將 AWS Cloud9 的服務連結角色新增到 EBS 磁碟區所使用的 AWS KMS 金鑰的步驟。此模式所描述的設定可協助您成功建立並啟動使用預設為加密的 EBS 磁碟區的 IDE。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- EBS 磁碟區的預設加密已開啟。如需有關預設加密的詳細資訊，請參閱 Amazon 彈性運算雲端 (Amazon EC2) 文件中的 Amazon EBS 加密。
- 現有客戶管理的 KMS 金鑰，用於加密您的 EBS 磁碟區。

注意：您不需要為 AWS Cloud9 建立服務連結角色。當您建立 AWS Cloud9 開發環境時，AWS Cloud9 會為您建立服務連結角色。

架構

技術, 堆

- AWS Cloud9
- IAM
- AWS KMS

工具

- AWS Cloud9 是整合式開發環境 (IDE)，可協助您撰寫程式碼、建置、執行、測試和偵錯軟體。它也可協助您將軟體發行到 AWS 雲端。
- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以協助保護資料。

史诗

尋找預設的加密金鑰值

任務	描述	所需技能
記錄 EBS 磁碟區的預設加密金鑰值。	登入 AWS 管理主控台並開啟 Amazon EC2 主控台。選擇 EC2 儀表板，然後在帳戶屬性中選擇資料保護和安全性。在 EBS 加密區段中，複製並記錄預設加密金鑰中的值。	雲端架構師、 DevOps 工程師

提供 AWS KMS 金鑰的存取權

任務	描述	所需技能
為 AWS Cloud9 提供 EBS 磁碟區之 KMS 金鑰的存取權。	1. 開啟 AWS KMS 主控台，然後選擇客戶受管金鑰。 選取用於 Amazon EBS 加密的 AWS KMS 金鑰，然後選擇檢視金鑰。 2. 在 [金鑰原則] 索引標籤上，確認您可以看到金鑰原則的文字格式。如果您看不到文字表單，請選擇 [切換至原則檢視]。 3. 選擇編輯。將 [其他資訊] 區段中的程式碼新增至原則，然後選擇 [儲存變更]。政策變更允許 AWS Cloud9 的服務連結角色存取金鑰。`AWSServiceRoleForAWSCloud9` 如需更新金鑰政策的詳細資訊，請參閱如何變更金鑰政策 (AWS KMS 文件)。 重要事項：AWS Cloud9 的服務連結角色會在您啟動第一個 IDE 時自動建立。如需詳細資訊，請參閱 AWS Cloud9 文件中的建立服務連結角色。	雲端架構師、 DevOps 工程師

建立並啟動 IDE

任務	描述	所需技能
建立並啟動 AWS Cloud9 IDE。	開啟 AWS Cloud9 主控台，然後選擇「建立環境」。 按照 AWS Cloud9 說明文件中的建立 EC2 環境中的步驟，根據您的需求設定 IDE。	雲端架構師、 DevOps 工程師

相關資源

- 加密 AWS Cloud9 使用的 EBS 磁碟區
- 為 AWS Cloud9 建立服務連結角色
- 在 AWS Cloud9 中建立 EC2 環境

其他資訊

AWS KMS 金鑰政策更新

使用您的 AWS 帳戶 ID 取代 <aws_accountid>。

```
{
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::<aws_accountid>:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
```

使用跨帳戶金鑰

如果您想要使用跨帳戶 KMS 金鑰，則必須將授權與 KMS 金鑰原則結合使用。這可讓跨帳戶存取金鑰。在您用來建立 Cloud9 環境的相同帳戶中，在終端機中執行下列命令。

```
aws kms create-grant \
 --region <Region where Cloud9 environment is created> \
 --key-id <The cross-account KMS key ARN> \
 --grantee-principal arn:aws:iam::<The account where Cloud9 environment is
 created>:role/aws-service-role/cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9 \
 --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
```

執行此命令之後，您可以透過 EBS 加密搭配不同帳戶中的金鑰來建立 Cloud9 環境。

自動建立基於標籤的 Amazon CloudWatch 儀表板

創建者：賈納克·瓦達里亞（AWS）、泰亞吉（AWS）和維諾德庫瑪曼達拉普（AWS）

代碼存儲庫：黃金信號	環境：生產	技術：營運、雲端原生、管理與治理
AWS 服務：AWS CDK；Amazon CloudWatch；AWS CodeBuild；AWS CodePipeline		

Summary

手動建立不同的 Amazon CloudWatch 儀表板可能非常耗時，尤其是當您必須建立和更新多個資源以自動擴展環境時。自動建立和更新 CloudWatch 儀表板的解決方案可以節省您的時間。此模式可協助您部署完全自動化的 AWS Cloud Development Kit (AWS CDK) 管道，根據標籤變更事件為您的 AWS 資源建立和更新 CloudWatch 儀表板，以顯示 Golden Signals 指標。

在網站可靠性工程（SRE）中，黃金信號指的是一組全面的指標，從用戶或消費者的角度提供服務的廣泛視野。這些量度包含延遲、流量、錯誤和飽和度。如需詳細資訊，請參閱什麼是網站可靠性工程 (SRE)？ 在網 AWS 站上。

此模式提供的解決方案是事件驅動的。部署完成後，它會持續監控標籤變更事件，並自動更新 CloudWatch 儀表板和警示。

先決條件和限制

前提

- 一個活躍的 AWS 帳戶
- AWS Command Line Interface (AWS CLI)、已安裝及設定
- AWS CDK V2 的先決條件
- 一個引導環境 AWS
- Python 版本 3
- AWS SDK for Python (Boto3), 已安裝

- Node.js 版本 18 或更新版本
- 節點套件管理員 (npm)，已安裝並設定 AWS CDK
- 中等（級別 200）熟悉和 AWS CDK AWS CodePipeline

限制

此解決方案目前僅為下列 AWS 服務建立自動化儀表板：

- Amazon Relational Database Service (Amazon RDS)
- AWS Auto Scaling
- Amazon Simple Notification Service (Amazon SNS)
- Amazon DynamoDB
- AWS Lambda

架構

目標技術堆疊

- CloudWatch 儀表板
- CloudWatch 警報

目標架構

1. 已設定應用程式 AWS 標籤或程式碼變更的標籤變更事件會啟動管道， AWS CodePipeline 以建置和部署更新的 CloudWatch 儀表板。
2. AWS CodeBuild 運行 Python 腳本以查找已配置標籤的資源，並將資源 ID 存儲在 CodeBuild 環境中的本地文件中。
3. CodeBuild 運行 cdk 合成器以生成用於部署 CloudWatch 儀表 AWS CloudFormation 板和警報的模板。
4. CodePipeline 將 AWS CloudFormation 範本部署到指定的 AWS 帳戶 和區域。
5. 成功部署 AWS CloudFormation 堆疊後，您可以檢視 CloudWatch 儀表板和警示。

自動化和規模

此解決方案已自動化，使用 AWS CDK. 您可以在 Amazon CloudWatch 存儲庫上的 GitHub 黃金信號儀表板中找到代碼。對於其他擴展和建立自訂儀表板，您可以設定多個標籤鍵和值。

工具

Amazon 服務

- Amazon EventBridge 是一種無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來，包括 AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他來源的事件匯流排。 AWS 帳戶
- AWS CodePipeline協助您快速建模和設定軟體發行版本的不同階段，並自動執行持續發行軟體變更所需的步驟。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit是一種版本控制服務，可以幫助您私下存儲和管理 Git 存儲庫，而無需管理自己的源代碼控制系統。
- AWS Command Line Interface (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

最佳實務

安全性最佳做法是，您可以對連線到管道的來源儲存庫使用加密和驗證。如需其他最佳作法，請參閱 CodePipeline 文件中的CodePipeline 最佳做法和使用案例。

史诗

設定和部署範例應用程式

任務	描述	所需技能
設定和部署範例應用程式。	1. 使用以下命令克隆 GitHub 示例代碼存儲庫：	AWS DevOps

任務	描述	所需技能
	`git clone https://github.com/aws-samples/golden-signals-dashboards-sample-app` 2. 導航到計算機上克隆的存儲庫，然後使用您選擇的編輯器打開`src/project-settings.ts` 文件。 3. 根據您的 AWS 資源標籤和應用程式對應變更`projectSettings` 常數值。 4. 設定`AWS_ACCOUNT` `AWS_REGION` 、和`GS_DASHBOARD_INSTANCE` 環境變數： • 設置`AWS_ACCOUNT` 為您帳戶的 AWS 帳戶 ID。 • 設定`AWS_REGION` 為您要部署範例應用程式的區域。 • 根據您的開發環境`prod`，設定`GS_DASHBOARD_INSTANCE` 為`dev`、或。`test`（我們建議使`test`用此模式中描述的測試程序。） 5. AWS CLI 使用您的 AWS 憑據設置。如需詳細資訊，請參閱 AWS CLI 文件中的使	

任務	描述	所需技能
	用指令設定和檢視組態設定。 6. 執行下列命令以部署黃金信號儀表板範例應用程式： `sh deploy.sh`	

任務	描述	所需技能
自動建立儀表板和警示。	部署範例應用程式之後，您可以使用預期的標籤值建立此解決方案支援的任何資源，這些資源會自動建立指定的儀表板和警示。 要測試此解決方案，請創建一個 AWS Lambda 函數： 1. 登入您部署範例應用程式的 AWS 區域 位置。 AWS Management Console 2. 開啟 Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。 3. 選擇建立函數，然後輸入函數名稱。 4. 在 [進階設定] 窗格中，選取 [啟用標籤]，然後選擇 [新增標籤]。輸入下列索引鍵和值： • 索引鍵：`AutoDashboard` • 值：`True` 5. 選擇建立函數。 Lambda 函數會立即啟動程式碼管道，自動為該特定 Lambda 函數建立儀表板和警示。 6. 若要檢視自動化儀表板和警示，請在 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。	AWS DevOps

任務	描述	所需技能
	您可以檢視您在projectSettings 常數中指定之函數的自訂儀表板和警示 (預設為 App1-Lambda)。 7. 選取 Lambda 函數的儀表板，以檢視作為此解決方案一部分建立的其他自動化儀表板。 8. 針對其他服務 (例如 Amazon RDS、Amazon SNS 和 DynamoDB) 重複這些步驟 AWS Auto Scaling，以產生關聯的儀表板。如需 Amazon RDS 的範例，請參閱其他資訊一節。	

移除範例應用程式

任務	描述	所需技能
移除golden-signals-dashboard 建構。	1. 若要移除範例應用程式所建立的所有 AWS CloudFormation 堆疊，您必須重新設定AWS_ACCOUNT AWS_REGION 、和GS_DASHBOARD_INSTANCE 環境變數。該destroy.sh 命令需要這些配置。 • AWS_ACCOUNT 是您帳戶的 AWS 帳戶 ID。 • AWS_REGION 是您部署範例應用程式的區域。	AWS DevOps

任務	描述	所需技能
	• GS_DASHBOARD_INSTANCE 是dev、或testprod，根據您先前的設定而定。 2. AWS CLI 使用您的 AWS 憑據進行設置。 3. 執行下列命令以移除範例應用程式和所有關聯的 AWS CloudFormation 堆疊： `sh destroy.sh`	

故障診斷

問題	解決方案
找不到 Python 命令（參findresources.sh 照第 8 行）。	請檢查您的 Python 安裝版本。如果您已安裝 Python 版本 3，請python3在resources.sh 檔案的第 8 行取代python，然後再次執行sh deploy.sh 命令以部署解決方案。

相關資源

- 引導（文檔）AWS CDK
- 使用命名的配置AWS CLI 文件（文檔）
- AWS CDK 工作坊

其他資訊

下圖顯示作為此解決方案一部分而建立的 Amazon RDS 範例儀表板。

使用 AWS Config 進階查詢，根據 AWS 資源的建立日期尋找 AWS 資源

創建者：因娜薩曼 (AWS)

環境：生產	技術：營運；安全性、身分識別、合規性	AWS 服務：AWS Config; Amazon EBS; Amazon EC2; Amazon S3; AWS Lambda

Summary

此模式顯示如何使用 AWS Config 進階查詢功能，根據 AWS 資源的建立日期尋找 AWS 資源。

AWS Config 進階查詢使用 SQL 子集來查詢 AWS 資源的組態狀態，以瞭解庫存管理、營運智慧、安全性和合規性。您可以使用這些查詢在單一 AWS 帳戶和 AWS 區域或跨多個帳戶和區域尋找 AWS 資源。透過執行使用該resourceCreationTime屬性的查詢，您可以根據 AWS 資源的特定建立日期傳回 AWS 資源清單。您可以使用下列其中一種方式執行 AWS Config 進階查詢：

- AWS 組態主控台中的 AWS Config 查詢編輯器
- AWS Command Line Interface (AWS CLI)

此模式的其他資訊部分中的範例查詢會傳回在特定 60 天期間內建立的 AWS 資源清單。查詢的輸出包括每個已識別資源的下列資訊：

- 帳戶 ID
- 區域
- 資源名稱
- 資源 ID
- 資源類型
- 標籤
- 建立時間

範例查詢也會顯示如何將庫存清單範圍設定為具有 「WHERE...」的特定資源類型。IN」 聲明。您可以使用類似的查詢來尋找也適用於標籤的其他 AWS 資源類型。

注意：若要查詢多個 AWS 帳戶和區域或跨 AWS Organizations 組織的資源，您必須使用 AWS Config 彙總器。如需詳細資訊，請參閱 AWS Config 開發人員指南中的多帳戶多區域資料彙總。全球資源僅記錄在其所在地區。例如，AWS Identity and Access Management (IAM) 是一種全球資源，記錄在 us-east-1 (維吉尼亞北部區域) 中。

先決條件和限制

先決條件

- 啟用 AWS Config 的一或多個有效 AWS 帳戶，以記錄所有支援的資源類型 (預設組態)
- (適用於多帳戶、多區域查詢) 已啟用的 AWS Config 彙總工具

限制

- AWS Config 進階查詢結果會分頁。選擇匯出時，最多可從 AWS 管理主控台匯出 500 個結果。您也可以使用 API 一次擷取多達 100 個分頁結果。
- AWS Config 進階查詢使用具有自己語法限制的 SQL 子集。如需詳細資訊，請參閱 AWS AWS Config 開發人員指南中的查詢 AWS 資源目前組態狀態的限制。

工具

工具

- AWS Config 提供 AWS 帳戶中的資源及其設定方式的詳細檢視。它可協助您識別資源彼此之間的關聯性，以及其組態隨時間變更的情況。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

史诗

執行 AWS Config 進階查詢

任務	描述	所需技能
確認 AWS Config 支援您查詢的資源。	如需 AWS Config 支援的 AWS 資源完整清單，請參閱 AWS	雲端管理員

任務	描述	所需技能
	Con fig 開發人員指南中支援的資源類型。	
確認已建立並執行組態錄製程式。	遵循 AWS Config 開發人員指南中管理組態記錄器中的指示。 注意：AWS Config 會自動建立並啟動預設組態記錄器。	雲端管理員

任務	描述	所需技能
執行查詢。	遵循使用 SQL 查詢編輯器 (主控台) 查詢中的指示，或使用 AWS Config 開發人員指南中的 SQL 查詢編輯器 (AWS CLI) 進行查詢。 注意：如果您在執行 AWS CLI 命令時收到錯誤訊息，請確定您使用的是最新版本的 AWS CLI。 對於單一 AWS 帳戶和區域查詢 在 [查詢編輯器] 頁面的 [查詢範圍] 區段中，確定您選擇 [僅限此帳戶和區域]。 對於多帳戶和多區域查詢 在查詢編輯器頁面的「查詢範圍」區段中，確定您已建立並選取 AWS Config 彙總器。如需詳細資訊，請參閱 AWS Config 開發人員指南中的多帳戶多區域資料彙總。 如果跨多個帳戶或區域的查詢無法運作，請遵循 AWS Config 開發人員指南中多帳戶多區域資料彙總疑難排解中的指示進行。 附註：若要根據 resourceType 修改查詢的範圍，請使用 WHERE 資源類型 IN (...) 建構。如需查詢範例，請參閱其	雲端管理員

任務	描述	所需技能
	他資訊一節中的 AWS Config 進階查詢範例。	

其他資訊

AWS Config 進階查詢範例

下列範例查詢會傳回特定 60 天期間內建立的 AWS 資源清單。如需更多 AWS Config 進階查詢範例，請參閱 AWS Config 開發人員指南中的查詢範例。

```
SELECT
  accountId,
  awsRegion,
  resourceName,
  resourceId,
  resourceType,
  resourceCreationTime,
  tags
WHERE
  resourceType IN (
    'AWS::CloudFormation::Stack',
    'AWS::EC2::VPC',
    'AWS::EC2::Volume',
    'AWS::EC2::Instance',
    'AWS::RDS::DBInstance',
    'AWS::ElasticLoadBalancingV2::LoadBalancer',
    'AWS::ServiceCatalog::CloudFormationProvisionedProduct',
    'AWS::EC2::NetworkInterface',
    'AWS::EC2::Subnet',
    'AWS::EC2::SecurityGroup',
    'AWS::AutoScaling::AutoScalingGroup',
    'AWS::Lambda::Function',
    'AWS::DynamoDB::Table',
    'AWS::S3::Bucket'
  )
  AND resourceCreationTime BETWEEN '2022-05-23T00:00:00.000Z' AND
 '2022-07-23T17:59:51.000Z'
ORDER BY
  accountId ASC,
```

```
  resourceType ASC
```

資料隱私與保護

AWS Config 會分別在每個 AWS 區域啟用。 為了符合法規要求，需要應用特殊考量事項，例如建立單獨的區域彙總工具。如需詳細資訊，請參閱 AWS Config 開發人員指南中的 AWS Config 中的資料保護。

IAM 許可

AWS_ ConfigRole AWS 受管政策需要作為執行 AWS Config 進階查詢的最低許可集。如需詳細資訊，請參閱 AWS Config 開發人員指南中指派給 AWS Config 的 IAM 角色許可一節中，瞭解如何取得組態詳細資訊的 IAM 角色政策。

檢視 AWS 帳戶或組織的 EBS 快照詳細資訊

環境：生產	技術：營運、儲存與備份	AWS 服務：Amazon EBS

Summary

此模式說明如何在 Amazon 網路服務 (AWS) 帳戶或 AWS Organizations 中的組織單位 (OU) 中，自動產生所有 Amazon 彈性區塊存放區 (Amazon EBS) 快照的隨需報告。

Amazon EBS 是一種 easy-to-use可擴展的高效能區塊儲存服務，專為亞馬遜彈性運算雲端 (Amazon EC2) 設計。EBS 磁碟區提供耐用且持久的儲存，您可以將其連接到 EC2 執行個體。您可以使用 EBS 磁碟區做為資料的主要儲存，並透過建立快照來 point-in-time 備份 EBS 磁碟區。您可以使用 AWS 管理主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 檢視特定 EBS 快照的詳細資訊。此模式提供程式設計方式，可擷取 AWS 帳戶或 OU 中所有 EBS 快照的相關資訊。

您可以使用此模式提供的指令碼來產生逗號分隔值 (CSV) 檔案，其中包含每個快照的下列資訊：帳戶 ID、快照 ID、磁碟區 ID 和大小、建立快照的日期、執行個體 ID 和說明。如果您的 EBS 快照已標記，報告也會包含擁有者和專案團隊屬性。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 已安裝並設定 AWS CLI 第 2 版
- 具有適當許可的 AWS Identity and Access Management (IAM) 角色 (如果您打算從 AWS Organizations Organization 執行指令碼，則針對特定帳戶或 OU 中所有帳戶的存取許可)

架構

下圖顯示產生 EBS 快照隨需報告的指令碼工作流程，這些快照分散到 OU 中的多個 AWS 帳戶。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon Elastic Block Store (Amazon EBS) 提供區塊層級儲存體磁碟區，可與 EC2 執行個體搭配使用。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。

Code

您可以在 aws-ebs-Snapshot-aw s組織存放區域中取得此模式中使用的 GitHub範例應用程式的程式碼。請遵循下一節中的指示來使用範例檔案。

史诗

下載腳本

任務	描述	所需技能
下載 Python 腳本。	從GitHub 儲存庫下載指令碼 GetSnapshotDetailsAllAccountsOU.py。	一般 AWS

取得 AWS 帳戶的 EBS 快照詳細資訊

任務	描述	所需技能
執行 Python 指令碼。	執行 命令： `python3 getsnapshotinfo.py --file <output-file>.csv --region <region-name>` 其中<output-file> 指的是 CSV 輸出檔案，您想	一般 AWS

任務	描述	所需技能
	要放置 EBS 快照的相關資訊，<region-name> 也就是儲存快照的 AWS 區域。例如： `python3 getsnapshotinfo.py --file snapshots.csv --region us-east-1`	

取得組織的 EBS 快照詳細資料

任務	描述	所需技能
執行 Python 指令碼。	執行 命令： `python3 getsnapshotinfo.py --file <output-file>.csv --role <IAM-role> --region <region-name>` 其中<output-file> 指的是您想要放置 EBS 快照相關資訊的 CSV 輸出檔案，<IAM-role> 是提供存取 AWS Organizations 許可的角色，<region-name> 也是儲存快照的 AWS 區域。例如： `python3 getsnapshotinfo.py --file snapshots.csv --role <IAM role> --region us-west-2`	一般 AWS

相關資源

- Amazon EBS 文檔
- Amazon EBS 動作
- Amazon EBS API 參考
- 提高 Amazon EBS 性能
- Amazon EBS 資源
- EBS 快照定價

其他資訊

EBS 快照類型

Amazon EBS 根據擁有權和存取權提供三種類型的快照：

- 由您擁有 — 依預設，只有您可以從您擁有的快照建立磁碟區。
- 公開快照 — 您可以與所有其他 AWS 帳戶公開共用快照。若要建立公開快照，您可以修改快照的許可，以便與您指定的 AWS 帳戶共用。然後，您將授權的使用者可以透過建立自己的 EBS 磁碟區來使用您共用的快照，而您的原始快照不受影響。您也可以將未加密的快照公開提供給所有 AWS 使用者。但是，出於安全原因，您無法公開加密的快照。公開快照會造成重大的安全風險，因為可能會暴露個人和敏感資料。我們強烈建議您不要與所有 AWS 帳戶共用 EBS 快照。如需共用快照的詳細資訊，請參閱 AWS 文件。
- 私人快照 — 您可以私下與您指定的個別 AWS 帳戶共用快照。若要與特定 AWS 帳戶私下共用快照，請按照 AWS 文件中的指示進行操作，然後針對許可設定選擇 Priv ate。您授權的使用者可使用您共用的快照，以建立他們自己的 EBS 磁碟區，同時原始快照仍然不受影響。

概述和程序

下表提供 EBS 快照詳細資訊的連結，包括如何透過尋找和刪除未使用的快照來降低 EBS 磁碟區成本，以及封存不需要頻繁擷取或快速擷取的極少存取快照。

有關信息	See
快照、其功能和限制	建立 Amazon EBS 快照
如何建立快照	主控台：建立快照

	AWS CLI：建立快照命令 例如： `aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description " volume snapshot"`
刪除快照 (一般資訊)	刪除 Amazon EBS 快照
如何刪除快照	主控台：刪除快照 AWS CLI：刪除快照命令 例如： `aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0`
封存快照 (一般資訊)	存檔 Amazon EBS 快照 Amazon EBS 快照封存 (部落格文章)
如何封存快照	主控台：封存快照 AWS CLI：modify-snapshot-tier 命令
如何擷取封存的快照	主控台：還原封存的快照 AWS CLI：restore-snapshot-tier 命令
快照定價	Amazon EBS 定價

常見問答集

最短存檔期是多少？

最短封存期間為 90 天。

還原封存快照需要多久時間？

將封存的快照從封存層還原到標準層最多可能需要 72 小時，取決於快照的大小。

封存的快照是否為完整快照？

封存的快照一律是完整快照。

使用者可以封存哪些快照？

您只能封存您在帳戶中擁有的快照。

您可以存檔已註冊 Amazon 機器映像 (AMI) 的根裝置磁碟區的快照嗎？

否，您無法封存已註冊 AMI 之根裝置磁碟區的快照。

共用快照的安全性考量為何？

共用快照時，您授予其他人存取快照上所有資料的權限。僅與您信任資料的人員共用快照。

如何與另一個 AWS 區域共享快照？

快照受限於其建立的區域。若要與另一個區域共用快照，請將該快照複製到該區域，然後共用。

您可以共用加密的快照嗎？

您無法共用使用預設 AWS 受管金鑰加密的快照。您只能共用使用客戶管理金鑰加密的快照。共用加密快照時，您還必須共用用於加密快照的客戶管理金鑰。

未加密的快照呢？

您可以公開共用未加密的快照。

更多模式

- 允許 EC2 執行個體寫入 AMS 帳戶中 S3 儲存貯體的存取權
- 自動化 AWS 資源評估
- 使用 Amazon Inspector 和 AWS Security Hub 自動執行跨帳戶工作負載的安全掃描
- ???
- 使用 Amazon SageMaker 和 Azure 建置 MLOP 工作流程 DevOps
- 使用 Amazon CloudWatch 觀察性存取管理員集中監控
- 針對 AWS IoT 環境中的安全事件設定記錄和監控
- 使用工作階段管理員 Connect 到 Amazon EC2 執行個體
- 使用 Amazon CloudWatch 異常偵測為自訂指標建立警示
- ???
- 透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 在 Amazon EKS 工作者節點上安裝 SSM CloudWatch 代理程式和代理程式 preBootstrapCommands
- 整合石分支通用控制器與 AWS 大型主機現代化
- 使用 Step Functions 函數和 Lambda 代理函數在 AWS 帳戶之間啟動 CodeBuild 專案
- 監控和修復 AWS KMS 金鑰的排程刪除
- 監控跨多個 AWS 帳戶共用 Amazon 機器映像的使用
- 從 AWS Step Functions 同步執行 AWS Systems Manager Automation 任務
- 使用 AWS Fargate 大規模執行事件驅動和排程的工作負載
- 在多區域、多帳戶組織中設定 AWS CloudFormation 漂移偵測
- 在 AWS 上的 IBM Db2 上為 SAP 設定災難復原
- 使用 AWS Organizations 自動標記 Transit Gateway 附件
- 使用 Splunk 檢視 AWS Network Firewall 日誌和指標

SaaS

主題

- 透過單一控制平面管理多個 SaaS 產品的租用戶
- 更多模式

透過單一控制平面管理多個 SaaS 產品的租用戶

創建者：拉曼納阿凡查 (AWS)、珍妮弗帕斯卡 (AWS)、基山卡瓦拉 (AWS) 和安魯莎曼達娃 (AWS)

環境：PoC 或試點	技術：SaaS	AWS 服務：Amazon API Gateway；Amazon Cognito；AWS Lambda；AWS Step Functions；Amazon DynamoDB

Summary

此模式說明如何在 AWS 雲端的單一控制平面上管理多個軟體即服務 (SaaS) 產品的租用戶生命週期。所提供的參考架構可協助組織在其個別 SaaS 產品中減少冗餘共用功能的實作，並提供大規模的治理效率。

大型企業可以在不同的業務單位擁有多個 SaaS 產品。這些產品通常需要佈建以供不同訂閱層級的外部租用戶使用。如果沒有共同的租用戶解決方案，IT 管理員就必須花時間跨多個 SaaS API 管理無差異的功能，而不是專注於核心產品功能開發。

此模式中提供的一般租用戶解決方案有助於集中管理組織的許多共用 SaaS 產品功能，包括：

- 安全
- 租用戶佈建
- 租用戶資料儲存
- 租戶通訊
- 產品管理
- 指標記錄和監控

先決條件和限制

先決條件

- 有效的 AWS 帳戶

- Amazon Cognito 或第三方身分供應商 (IdP) 的知識
- Amazon API Gateway 的知識
- AWS Lambda 的知識
- Amazon DynamoDB 的知識
- AWS Identity and Access Management (IAM) 的知識
- AWS Step Functions 的知識
- AWS CloudTrail 和 Amazon 的知識 CloudWatch
- 有關 Python 庫和代碼的知識
- SaaS API 的知識，包括不同類型的使用者 (組織、租用戶、系統管理員和應用程式使用者)、訂閱模型和租用戶隔離模型
- 瞭解組織的多產品 SaaS 需求和多租用戶訂閱

限制

- 此模式不涵蓋通用租戶解決方案與個別 SaaS 產品之間的整合。
- 此模式只會在單一 AWS 區域中部署 Amazon Cognito 服務。

架構

目標技術堆疊

- Amazon API Gateway
- Amazon Cognito
- AWS CloudTrail
- Amazon CloudWatch
- Amazon DynamoDB
- IAM
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Notification Service (Amazon SNS)
- AWS 步驟功能

目標架構

下圖顯示在 AWS 雲端的單一控制平面上管理多個 SaaS 產品租用戶生命週期的範例工作流程。

該圖顯示以下工作流程：

1. AWS 使用者透過呼叫 API Gateway 端點來啟動租用戶佈建、產品佈建或管理相關動作。
2. 使用者會透過從 Amazon Cognito 使用者集區或其他 IdP 擷取的存取權杖進行驗證。
3. 個別佈建或管理任務是由與 API Gateway API 端點整合的 Lambda 函數執行。
4. 一般租用戶解決方案的管理 API (適用於租用戶、產品和使用者) 會收集所有必要的輸入參數、標頭和 Token。然後，管理 API 會叫用相關聯的 Lambda 函數。
5. 管理 API 和 Lambda 函數的 IAM 許可均由 IAM 服務進行驗證。
6. Lambda 函數會在 DynamoDB 和 Amazon S3 中存放和擷取目錄中的資料 (適用於租用戶、產品和使用者)。
7. 驗證許可後，就會叫用 AWS Step Functions 工作流程來執行特定任務。圖中的範例顯示承租人佈建工作流程。
8. 個別 AWS Step Functions 工作流程任務會在預定的工作流程 (狀態機器) 中執行。
9. 執行與每個工作流程任務相關聯的 Lambda 函數所需的任何重要資料，都會從 DynamoDB 或 Amazon S3 擷取。可能需要使用 AWS CloudFormation 範本佈建其他 AWS 資源。
10. 如有需要，工作流程會傳送請求，將特定 SaaS 產品的其他 AWS 資源佈建到該產品的 AWS 帳戶。
11. 當請求成功或失敗時，工作流程會將狀態更新作為訊息發佈到 Amazon SNS 主題。
12. Amazon SNS 已訂閱 Step Functions 工作流程的 Amazon SNS 主題。
13. 然後，Amazon SNS 會將工作流程狀態更新傳回給 AWS 使用者。
14. 每個 AWS 服務動作的日誌 (包括 API 呼叫的稽核追蹤) 都會傳送到 CloudWatch。可以在中 CloudWatch 為每個使用案例配置特定規則和警報。
15. 日誌存檔在 Amazon S3 儲存貯體中，以供稽核之用。

自動化和規模

此模式使用 CloudFormation 範本來協助自動化一般租用戶解決方案的部署。該模板還可以幫助您快速向上或向下銷售相關資源。

如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的使用 AWS CloudFormation 範本。

工具

工具

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- Amazon Cognito 為網頁和行動應用程式提供身份驗證、授權和使用者管理功能。
- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。

最佳實務

此模式中的解決方案使用單一控制平面來管理多個租用戶的上線作業，以及佈建多個 SaaS 產品的存取權。控制平面可協助管理使用者管理其他四個功能特定平面：

- 安全平面
- 工作流平面
- 通訊平面
- 記錄和監控平面

史诗

設定安全性平面

任務	描述	所需技能
建立多租戶 SaaS 平台的需求。	建立下列項目的詳細要求： • 租用戶 • 使用者 • 角色 • 軟體 SaaS • 訂閱 • 檔案交流	雲端架構師、AWS 系統管理員
設置 Amazon Cognito 服務。	按照 Amazon Cognito 開發人員指南中的開始使用 Amazon Cognito 中的說明進行操作。	雲端架構師
設定必要的 IAM 政策。	為您的使用案例建立必要的 IAM 政策。然後，將政策對應至 Amazon Cognito 中的 IAM 角色。 如需詳細資訊，請參閱 Amazon Cognito 開發人員指南中的使用政策管理存取和以角色為基礎的存取控制。	雲端管理員、雲端架構師、AWS IAM 安全
設定必要的 API 權限。	使用 IAM 角色和政策以及 Lambda 授權人來設定 API Gateway 存取許可。 如需指示，請參閱 Amazon API Gateway 開發人員指南的下列各節：	雲端管理員、雲端架構師

任務	描述	所需技能
	• 使用 IAM 許可控制對 API 的存取 • 使用 API Gateway Lambda 授權器	

設定資料平面

任務	描述	所需技能
建立所需的資料目錄。	1. 建立 DynamoDB 表格以儲存使用者目錄的資料。請確定您包含使用者屬性和角色。此外，請確保在目錄表格上執行資料塑型，以維護每個使用者和角色的必要屬性和可選屬性。 2. 建立 DynamoDB 表格以儲存產品目錄的資料。請確定您已為 SaaS 產品建立特定使用案例的模型。 3. 建立 DynamoDB 表格以儲存租用戶目錄的資料。請務必為租用戶、產品和多重 SaaS 訂閱和標籤的授權設定訂閱模式。 如需詳細資訊，請參閱 Amazon Dynam oDB 開發人員指南中的設定。	DBA

設定控制平面

任務	描述	所需技能
建立 Lambda 函數和 API Gateway API，以執行必要的控制平面工作。	建立個別的 Lambda 函數和 API Gateway API，以新增、刪除及管理下列項目： • 使用者 • 租用戶 • 產品 如需詳細資訊，請參閱 AWS Lambda 開發人員指南中的將 AWS Lambda 與 Amazon API Gateway 搭配使用。	應用程式開發人員

設定工作流程平面

任務	描述	所需技能
識別 AWS Step Functions 工作流程必須執行的任務。	識別並記錄下列項目的詳細 AWS Step Functions 工作流程要求： • 使用者 • 租用戶 • 產品 重要:確保關鍵利益相關者批准要求。	應用所有者
建立所需的 AWS Step Functions 工作流程。	1. 在 AWS Step Functions 中為使用者、租用戶和產品建立所需的工作流程。如需詳	應用程式開發人員, 建置

任務	描述	所需技能
	細資訊，請參閱 AWS Step Functions 開發人員指南。 2. 識別重試和錯誤處理機制。如需詳細資訊，請參閱 AWS 部落格上的步驟函數狀態機器處理錯誤、重試和新增警示。 3. 使用 Lambda 函數實作工作流程步驟。如需指示，請參閱 AWS Step Functions 開發人員指南中的建立使用 Lambda 的步驟函數狀態機器。 4. 視需要整合任何外部服務與 AWS Step Functions。 5. 在 DynamoDB 表格中維護每個工作流程的狀態，並使用 Amazon SNS 傳達每個工作流程的狀態。	

設定通訊平面

任務	描述	所需技能
創建 Amazon SNS 主題。	建立 Amazon SNS 主題以接收下列相關通知： • 工作流程狀 • 錯誤 • 重試	應用程序所有者，雲架構

任務	描述	所需技能
	如需詳細資訊，請參閱 Amazon SNS 開發人員指南中的建立 SNS 主題。	
訂閱每個 Amazon SNS 主題的端點。	若要接收發佈到 Amazon SNS 主題的訊息，您必須為每個主題訂閱端點。 如需詳細資訊，請參閱 Amazon SNS 開發人員指南中的訂閱 Amazon SNS 主題。	應用程式開發人員、雲端

設定記錄和監視平面

任務	描述	所需技能
針對一般租用戶解決方案的每個元件啟動記錄。	針對您建立的一般承租人解決方案中的每個資源，在元件層級啟動記錄。 如需詳細說明，請參閱下列主題： • 如何打開 CloudWatch 日誌以對我的 API 網關 REST API 或 WebSocket API 進行故障排除？ (AWS 知識中心) • 使用 CloudWatch 日誌記錄 (AWS Step Functions 開發人員指南) • AWS Lambda 函數記錄 (Python Lambda 開發人員指南)	應用開發人員、AWS 系統管理員、雲端管理員

任務	描述	所需技能
	• 在 Amazon Cognito 中進行記錄和監控 (Amazon Cognito 開發人員指南) • 使用 Amazon 監控 CloudWatch (Amazon DynamoDB 開發人員指南) 附註：您可以使用 IAM 政策，將每個資源的記錄合併到集中式記錄帳戶中。如需詳細資訊，請參閱集中式記錄和多帳戶安全防護。	

佈建和部署一般租用戶解決方案

任務	描述	所需技能
建立 CloudFormation 範本。	使用 CloudFormation 範本，將完整的一般租用戶解決方案及其所有元件的部署和維護作業自動化。 如需詳細資訊，請參閱 AWS CloudFormation 使用者指南。	應用開發人員、 DevOps 工程師、 CloudFormation 開發

相關資源

- 使用 Amazon Cognito 使用者集區做為授權者來控制對 REST API 的存取 (Amazon API Gateway 開發人員指南)
- 使用 API Gateway Lambda 授權器 (Amazon API Gateway 開發人員指南)
- Amazon Cognito 用戶池 (Amazon Cognito 開發人員指南)
- 跨帳戶跨區域 CloudWatch 主控台 (Amazon CloudWatch 使用者指南)

更多模式

- 使用自動化遷移策略識別和規劃 AppScore
- 使用 AWS 自動化 AppStream 2.0 資源的建立 CloudFormation
- 在 Amazon 服務中建立多租戶無伺服器架構 OpenSearch
- 使用 AWS Lambda 權杖自動販賣機為 Amazon S3 實作 SaaS 租用戶隔離
- 整合石分支通用控制器與 AWS 大型主機現代化
- 在 SaaS 架構中使用 C# 和 AWS CDK 進行筒倉模型的租用戶上線

安全性、身分識別、合規

主題

- 使用 Amazon Cognito 可身分集區從 ASP.NET 核心應用程式存取 AWS 服務
- 使用 AWS Directory Service 在 Amazon EC2 上驗證 Microsoft SQL 服務器
- 自動化事件回應和鑑識
- 自動修復 AWS Security Hub 標準發現項目
- 使用 Amazon Inspector 和 AWS Security Hub 自動執行跨帳戶工作負載的安全掃描
- 在 AWS Config 中使用 CloudTrail 自訂修復規則自動重新啟用 AWS
- 自動修復未加密的 Amazon RDS 資料庫執行個體和叢集
- 使用 AWS Organizations 和 AWS Secrets Manager 自動輪換大規模的 IAM 使用者存取金鑰
- 使用 IAM 存取分析器和 AWS CloudFormation 巨集 CodePipeline，在 AWS 帳戶中自動驗證和部署 IAM 政策和角色
- 雙向整合 AWS Security Hub 與 Jira 軟體
- 使用 EC2 Image Builder 和 Terraform 為強化的容器映像建立管道
- 使用 Terraform 在 AWS Organizations 中集中 IAM 存取金鑰管理
- 集中式記錄和多帳戶安全防護
- 檢查 Amazon CloudFront 分佈的存取記錄、HTTPS 和 TLS 版本
- 檢查 IPv4 和 IPv6 的安全群組輸入規則中是否有單一主機網路項目
- 選擇適用於企業應用程式的 Amazon Cognito 份驗證流程
- 使用 AWS CloudFormation 安全防護政策建立 AWS 組態自訂規則
- 從多個 AWS 帳戶建立 Prowler 安全發現結果的合併報告
- 使用 AWS Config 和 AWS Systems Manager 刪除未使用的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區
- 使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制 CloudFormation
- 使用地形表單部署和管理 AWS Control Tower 控制
- 部署管道，同時偵測多個程式碼交付項目中的安全性問題
- 使用 AWS Config 為公有子網路部署偵探屬性型存取控制
- 為公用子網路部署預防性屬性型存取控制
- 使用地形表單部署 AWS WAF 解決方案的安全自動化

- 使用 Step Functions 數使用 IAM 存取分析器動態產生 IAM 政策
- 使用 AWS 範本 GuardDuty 有條件地啟用 Amazon CloudFormation
- 在 Amazon RDS for SQL Server 中啟用透明資料加密
- 確保 AWS CloudFormation 堆疊是從授權的 S3 儲存貯體啟動
- 確保 AWS 負載平衡器使用安全接聽程式協定 (HTTPS、SSL/TLS)
- 確保啟動時已啟用 Amazon EMR 靜態資料的加密
- 確保 IAM 設定檔與 EC2 執行個體相關聯
- 確保亞 Amazon Redshift 叢集在建立時已加密
- 使用以下方式匯出 AWS IAM 身分中心身分及其指派的報告 PowerShell
- 監控和修復 AWS KMS 金鑰的排程刪除
- 使用 Security Hub 識別 AWS Organizations 中的公有 S3 儲存貯體
- 使用 AWS 以程式碼形式管理 AWS IAM 身分中心許可集 CodePipeline
- 使用 AWS 秘密管理員來管理登入
- 在啟動時監控 Amazon EMR 叢集的傳輸中加密
- 監控 Amazon ElastiCache 叢集以進行靜態加密
- 使用 AWS 設定監控 EC2 執行個體金鑰配對
- 監控安全群組的 ElastiCache 叢集
- 監控 IAM 根使用者活動
- 建立 IAM 使用者時傳送通知
- 使用服務控制策略防止在帳戶層級存取網際網路
- 使用 git 機密掃描 Git 存儲庫中的敏感信息和安全問題
- 將提醒從 AWS Network Firewall 傳送到 Slack 通道
- 使用 AWS 私有 CA 和 AWS 記憶體簡化私有憑證管理
- 在多帳戶環境中，關閉所有 Security Hub 成員帳戶的安全性標準控制
- 使用以下方式從 AWS IAM 身分中心更新 AWS CLI 登入資料 PowerShell
- 使用 AWS Config 監控 Amazon Redshift 安全組態
- 使用 Network Firewall 從輸出流量的伺服器名稱指示 (SNI) 擷取 DNS 網域名稱
- 使用地形表單為組織自動啟 GuardDuty 用 Amazon
- 確認新的 Amazon Redshift 叢集具有必要的 SSL 端點
- 確認新的 Amazon Redshift 叢集是否在 VPC 中啟動

- 更多模式

使用 Amazon Cognito 可身分集區從 ASP.NET 核心應用程式存取 AWS 服務

創建者：比布蒂·薩胡（AWS）和馬塞洛·巴博薩（AWS）

環境：PoC 或試點	技術：安全性、身分識別、合規性；Web 和行動應用程式	AWS 服務：Amazon Cognito

Summary

此模式討論如何設定 Amazon Cognito 使用者集區和身分集區，然後讓 ASP.NET 核心應用程式在身分驗證成功後存取 AWS 資源。

Amazon Cognito 為您的網頁和行動應用程式提供身份驗證、授權和使用者管理功能。 Amazon Cognito 的兩個主要元件是使用者集區和身分集區。

使用者集區是在 Amazon Cognito 中的使用者目錄。利用使用者集區，您的使用者可以透過 Amazon Cognito 登入您的 Web 或行動應用程式。您的使用者也可以透過社群身分供應商 (例如 Google、Facebook、Amazon 或蘋果)，以及透過 SAML 身分供應商登入。

Amazon Cognito 身分集區 (聯合身分) 可讓您為使用者建立唯一身分，並將其與身分提供者聯合。透過身分集區，您可以取得臨時、有限權限的 AWS 登入資料，以存取其他 AWS 服務。在開始使用新的 Amazon Cognito 身分集區之前，您必須指派一個或多個 AWS Identity and Access Management (IAM) 角色，以確定您希望應用程式使用者對 AWS 資源的存取層級。身分集區定義兩種類型的身分：已驗證和未驗證。每個身分類型都可以在 IAM 中指派自己的角色。已驗證身分屬於由公用登入提供者 (Amazon Cognito 使用者集區、Facebook、Google、SAML 或任何 OpenID Connect 提供者) 或開發人員提供者 (您自己的後端身份驗證程序) 驗證的使用者，而未經驗證的身分通常屬於來賓使用者。Amazon Cognito 收到使用者要求時，服務會判斷要求是經過驗證還是未驗證、決定與該身份驗證類型相關聯的角色，然後使用附加到該角色的政策來回應請求。

先決條件和限制

先決條件

- 具有 Amazon Cognito 和 IAM 許可的 AWS 帳戶
- 存取您要使用的 AWS 資源
- 核心 2.0.0 或更新版本

架構

技術, 堆

- Amazon Cognito
- 核心

目標架構

工具

工具、開發套件和 AWS 服務

- 視覺工作室或視覺工作室代碼
- Amazon。 AspNetCore. 身份認知 (1.0.4) — 包 NuGet
- AWSSDK三、三、三、三、三、三、三、三、三、三 NuGet
- Amazon Cognito

Code

附加的 .zip 檔案包含說明下列內容的範例檔案：

- 如何檢索登錄用戶的訪問令牌
- 如何將存取權杖交換為 AWS 登入資料
- 如何使用 AWS 登入資料存取亞馬遜簡易儲存服務 (Amazon S3) 服務

已驗證身分的 IAM 角色

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "mobileanalytics:PutEvents",
        "cognito-sync:*",
```

```
        "cognito-identity:*",
        "s3:ListAllMyBuckets*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

史诗

創建一個 Amazon Cognito 戶池

任務	描述	所需技能
建立使用者集區。	1. 登入 AWS 管理主控台，然後開啟 Amazon Cognito 主控台，網址為 https://console.aws.amazon.com/cognito/home。 2. 選擇 Manage User Pools (管理使用者集區)。 3. 在頁面右上角，選擇 Create a user pool (建立使用者集區)。 4. 提供使用者集區的名稱，選擇 [檢閱預設值]，然後選擇 [建立集區]。 5. 記下集區 ID。	開發人員
新增應用程式用戶端。	您可以創建一個應用程序以使用內置網頁進行註冊和登錄用戶。 1. 在使用者集區頁面左側的導覽列上，選擇 [一般設定] 下的 [應用程式用戶端]，然後	開發人員

任務	描述	所需技能
	選擇 [新增應用程式用戶端]。 2. 為您的應用程式命名，然後選擇 [建立應用程式用戶端]。 3. 記下應用程序客戶端 ID 和客戶端密鑰（選擇顯示詳細信息以查看客戶端密鑰）。	

建立 Amazon Cognito 身分集區

任務	描述	所需技能
建立 身分集區。	1. 在 Amazon Cognito 主控台上，選擇 [管理身分集區]，然後選擇 [建立新身分集區]。 2. 輸入識別集區的名稱。 3. 如果您想要啟用未驗證的身分，請從 [未驗證的身分] 區段中選取該選項。 4. 在 [驗證提供者] 區段中，透過設定使用者集區 ID 和應用程式用戶端 ID 來設定 Cognito 身分集區，然後選擇 [建立集區]。	開發人員
為身分識別集區指派 IAM 角色。	您可以編輯已驗證和未驗證使用者的 IAM 角色，或保留預設值，然後選擇 [允許]。對於此模式，我們將編輯經過身份驗證的 IAM 角色並為`s3:ListAllMyBuckets` 。如需範例程式碼，請參閱	開發人員

任務	描述	所需技能
	稍早在「工具」一節中提供的 IAM 角色。	
複製身分集區識別碼。	當您在上一個步驟中選擇「允許」時，就會顯示「開始使用 Amazon Cognito」頁面。在此頁面上，您可以從 [取得 AWS 登入資料] 區段複製身分集區 ID，或選擇右上角的編輯身分集區，然後從顯示的畫面複製身分集區 ID。	開發人員

設定範例應用程式

任務	描述	所需技能
複製範例 ASP.NET 核心網路應用程式。	1. 克隆範例. NET 核心網頁應用程式aws-aspnet-cognito-identity。 https://github.com/aws/ 2. 導航到`samples`文件夾並打開解決方案。在此專案中，您將設定`appsettings.json` 檔案並新增一個新頁面，以便在成功登入後呈現所有 S3 儲存貯體。	開發人員
添加依賴關係。	添加`Amazon.AspNetCore.Identity.Cognito` 到您的 ASP.NET 核心應用程序的 NuGet 依賴關係。	開發人員
將配置鍵和值添加到應用程序設置 .json 中。	將附加`appsettings.json` 檔案中的程式碼包含在`appsettings.json` 檔	開發人員

任務	描述	所需技能
	案中，然後以先前步驟的值取代預留位置。	
建立新使用者並登入。	在 Amazon Cognito 使用者集區中建立新使用者，並確認該使用者存在於使用者集區中的「使用者和群組」下。	開發人員
創建一個新的剃刀頁面稱為 MyStor3 桶。	添加一個新的 ASP.NET 核心剃刀頁面到您的示例應用程序，並替換內容，`MyS3Bucket.cshtml` 並`MyS3Bucket.cshtml.cs` 從附加的示例。在頁面中的導航下添加新的 **`_Layout.cshtml`** MyS3Bucket 頁面。	開發人員

故障診斷

問題	解決方案
從存放 GitHub 庫開啟範例應用程式之後，當您嘗試將 NuGet 套件新增至 Samples 專案時，您會收到錯誤訊息。	在資`src`料夾中，請務必從檔案中移除`Amazon.AspNetCore.Identity.Cognito` 專`Samples.sln` 案的參考。然後，您可以將 NuGet 套件新增至 Sample 專案，而不會出現任何問題。

相關資源

- Amazon Cognito
- Amazon Cognito 使用者集區
- Amazon Cognito 身份集區
- 存取原則範例

- GitHub -AWS ASP.NET Cognito 身分識別供應商

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS Directory Service 在 Amazon EC2 上驗證 Microsoft SQL 服務器

創建者：賈加迪·坎圖布加塔 (AWS) 和歐魯達洪巴德阿吉達洪 (AWS)

環境：PoC 或試點	來源：活動目錄	目標：AWS Directory Service
R 類型：不適用	工作量：Microsoft	技術：安全性、身分識別、合規性；資料庫
AWS 服務：AWS Directory Service		

Summary

此模式說明如何建立 AWS Directory Service 目錄，並使用它在亞馬遜彈性運算雲端 (Amazon EC2) 執行個體上驗證 Microsoft SQL Server。

AWS Directory Service 提供多種方式，可將 Amazon Cloud Directory 和 Microsoft 活動目錄 (AD) 與其他 AWS 服務搭配使用。目錄儲存使用者、群組和裝置的相關資訊，而管理員則使用這些目錄來管理資訊和資源的存取。AWS Directory Service 為想要在雲端中使用現有 Microsoft AD 或輕量型目錄存取通訊協定 (LDAP) 感知應用程式的使用者提供多種目錄選擇。它也同樣為需要使用目錄管理使用者、群組、裝置和存取的開發人員，提供這些選項。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 至少具有兩個私有子網路和兩個公有子網路的虛擬私有雲端 (VPC)
- AWS Identity and Access Management (IAM) 角色，可將伺服器加入網域

架構

源, 技術, 堆棧

- 來源可以是內部部署作用中目錄

目標技術堆疊

- 適用於 Microsoft 活動目錄（AWS 受管 Microsoft AD）的 AWS Directory Service

目標架構

工具

- SQL 服務器管理工作室（SSMS）是用於管理 Microsoft SQL 服務器，包括訪問，配置和管理 SQL 服務器組件的工具。

史诗

設定目錄

任務	描述	所需技能
選取 AWS 受管 Microsoft AD 作為目錄類型。	在 AWS Directory Service 主控台上，選擇 [目錄]、[設定目錄]、[下一步]。	DevOps
選擇版本。	從 AWS 受管 Microsoft AD 的可用版本中，選擇標準版。	DevOps
指定目錄 DNS 名稱。	使用完整網域名稱。此名稱只能在您的 VPC 內部解析。它不需要公開解析。	DevOps
設定管理員密碼。	設定預設管理使用者的密碼，名為 Admin。	DevOps
選擇 VPC 和子網路。	選擇將包含您的目錄和網域控制站子網路的 VPC。如果您沒有具有至少兩個子網路的	DevOps

任務	描述	所需技能
	VPC，則必須建立一個子網路。	
檢閱並啟動目錄。	檢閱目錄的版本和價格資訊，然後選擇 [建立目錄]。	DevOps

為網域中的 SQL 伺服器啟動 EC2 執行個體

任務	描述	所需技能
選取 SQL 伺服器的 AMI。	這個史詩般的步驟無縫地將 Windows EC2 實例加入到您的 AWS 託管 Microsoft AD 目錄。 在 Amazon EC2 主控台上，選擇啟動執行個體，然後為 SQL 伺服器選取適當的 Amazon 機器映像 (AMI)。	DevOps, DBA
設定執行個體詳細資訊。	設定視窗執行個體以符合您對 SQL 伺服器的需求。	DevOps, DBA
選取 key pair 名稱。	選取 key pair，然後啟動執行個體。	DevOps, DBA
新增網路。	您可以選擇在其中建立目錄的 VPC。	DevOps, DBA
選取 IAM role (IAM 角色)。	在進階設定中，選取具有 AWS 受管政策`AmazonSSMManagedInstanceCore` 並`AmazonSSMDirectoryServiceAccess` 附加到該設定檔的 IAM 設定檔。	DevOps, DBA

任務	描述	所需技能
新增子網路。	選擇 VPC 中的其中一個公用子網路。您所選取的子網路必須將所有外部流量路由到網際網路閘道。如果沒有，則將無法從遠端連線到執行個體。	DevOps, DBA
選擇您的網域。	從 [網域加入目錄] 清單中選擇您建立的網域。	DevOps, DBA
啟動執行個體。	選擇啟動執行個體。	DBA

使用 Directory Service 驗證 SQL 伺服器

任務	描述	所需技能
以視窗系統管理員身分登入。	使用 Windows 系統管理員登入資料登入 Windows EC2 執行個體。	DBA
登入 SQL 伺服器。	啟動 SQL 伺服器管理工作室 (SSMS)，並使用視窗驗證方法登入 SQL 伺服器。	DBA
建立目錄使用者的登入資訊。	在 SSMS 中，選擇 [安全性]，然後選擇 [新增登入]。	DBA
搜尋登入名稱。	選擇登入文字方塊旁邊的搜尋按鈕。	DBA
選擇一個位置。	在 [選取使用者或群組] 對話方塊中，選擇 [位置]。	DBA
輸入網路認證。	輸入您在建立目錄服務時所使用的完整網路身份證明；例如：`test.com\admin`	DBA

任務	描述	所需技能
選取目錄。	選擇 AWS 目錄名稱，然後選擇 [確定]。	DBA
選取物件名稱。	選取您要為其建立登入的使用者。選取位置、選擇整個目錄、搜尋使用者，然後新增登入。	DBA
登入 SQL 伺服器執行個體。	使用您的網域登入資料登入 SQL 伺服器的 Windows EC2 執行個體。	DBA
以網域使用者身分登入 SQL 伺服器。	啟動 SSMS 並使用 Windows 驗證方法連接到資料庫引擎。	DBA

相關資源

- AWS Directory Service 文件 (AWS 網站)
- 建立您的 AWS 受管 Microsoft AD 目錄 (AWS Directory Service 文件)
- 無縫加入 Windows EC2 執行個體 (AWS Directory Service 文件)
- AWS 上的 Microsoft SQL 服務器 (AWS 網站)
- SSMS 文件 (Microsoft 網站)
- 在 SQL 伺服器中建立登入 (SQL 伺服器文件)

自動化事件回應和鑑識

由盧卡斯考夫曼（AWS）和湯梅克雅庫博斯基（AWS）創建

代碼存儲庫：aws-automated-incident-response和取證	環境：生產	技術：安全性、身分識別、合規
AWS 服務：Amazon EC2；AWS Lambda；Amazon S3；AWS Security Hub；AWS Identity and Access Management		

Summary

此模式會部署一組使用 AWS Lambda 函數的程序來提供下列項目：

- 一種以最少的知識啟動事件響應過程的方法
- 符合 AWS 安全事件回應指南的自動化、可重複的程序
- 分離帳戶以操作自動化步驟，存儲成品並創建鑑識環境

自動化事件回應與鑑識架構遵循標準的數位鑑識程序，其中包含下列階段：

1. 遏制
2. 收購
3. 考試
4. 分析

您可以對靜態資料 (例如取得的記憶體或磁碟映像)，以及即時但在不同系統上的動態資料執行調查。

如需詳細資訊，請參閱其他資訊一節。

先決條件和限制

先決條件

- 兩個 AWS 帳戶：
  - 安全性帳戶，可以是現有帳戶，但最好是新帳戶
  - 鑑識帳戶，最好是新帳戶
- AWS Organizations 設定
- 在「Organizations」成員帳戶中：
  - 亞馬遜彈性運算雲端 (Amazon EC2) 角色必須具有對亞馬遜簡單儲存服務 (Amazon S3) 的取得和清單存取權，並可由 AWS Systems Manager 存取。我們建議使用 `AmazonSSMManagedInstanceCore` AWS 受管角色。請注意，啟動事件回應時，此角色會自動附加至 EC2 執行個體。回應完成後，AWS Identity and Access Management (IAM) 將移除執行個體的所有權限。
  - AWS 成員帳戶和事件回應和分析 VPC 中的虛擬私有雲端 (VPC) 端點。這些端點包括：S3 閘道、EC2 訊息、SSM 和 SSM 訊息。
- 在 EC2 執行個體上安裝的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如果 EC2 執行個體未安裝 AWS CLI，則需要網際網路存取，磁碟快照和記憶體擷取才能運作。在此情況下，指令碼會連絡網際網路以下載 AWS CLI 安裝檔案，並將其安裝在執行個體上。

限制

- 該框架不打算生成可以被視為電子證據的文物，可以在法庭上接受。
- 目前，此模式僅支援在 x86 架構上執行的 Linux 型執行個體。

架構

目標技術堆疊

- AWS CloudFormation
- AWS CloudTrail
- AWS Config
- IAM
- Lambda
- Amazon S3
- AWS AWS KMS 鑰管理系統
- AWS Security Hub

- Amazon Simple Notification Service (Amazon SNS)
- AWS Step Functions

目標架構

除了成員帳戶之外，目標環境還包含兩個主要帳戶：安全性帳戶和鑑識帳戶。使用兩個帳戶的原因如下：

- 將其與任何其他客戶帳戶分開，以便在取證分析失敗的情況下減少爆炸半徑
- 協助確保被分析成品的完整性的隔離與保護
- 對調查保密
- 為了避免威脅參與者可能已使用您受感染的 AWS 帳戶立即可用的所有資源的情況，方法是按下服務配額，防止您實例化 Amazon EC2 執行個體來執行調查。

此外，擁有單獨的安全和鑑識帳戶允許創建單獨的角色-一個響應者用於獲取證據和分析調查員。每個角色都可以訪問其單獨的帳戶。

下圖僅顯示帳戶之間的互動。每個帳戶的詳細信息顯示在後面的圖表中，並附上完整的圖表。

下圖顯示了成員帳戶。

1. 系統會將事件傳送至 Slack Amazon SNS 主題。

下圖顯示了安全帳戶。

2. 安全性帳戶中的 SNS 主題會啟動鑑識事件。

下圖顯示鑑識帳戶。

安全帳戶是針對記憶體和磁碟映像擷取建立兩個主要 AWS Step Functions 工作流程的地方。工作流程執行之後，他們會存取具有 EC2 執行個體涉及事件的成員帳戶，並啟動一組 Lambda 函數，以收集記憶體傾印或磁碟傾印。然後，這些成品會儲存在鑑識帳戶中。

鑑識帳戶將保留分析成品 S3 儲存貯體中「Step Functions」工作流程收集的成品。鑑識帳戶還將擁有 EC2 Image Builder 管道，用於建立鑑識執行個體的 Amazon 機器映像 (AMI)。目前，此影像是以 SANS 篩選工作站為基礎。

建置程序使用可連線至網際網路的維護 VPC。此映像檔稍後可用於旋轉 EC2 執行個體，以分析分析 VPC 中收集的成品。

分析 VPC 沒有互聯網連接。依預設，陣列會建立三個私用分析子網路。您最多可以建立 200 個子網路，也就是 VPC 中子網路數量的配額，但 VPC 端點必須為 AWS Systems Manager 工作階段管理員新增這些子網路，才能在其中自動執行命令。

從最佳實務的角度來看，我們建議您使用 AWS CloudTrail 和 AWS Config 執行下列動作：

- 追蹤鑑識帳戶中所做的變更
- 監控儲存與分析之成品的存取與完整性

工作流程

下圖顯示工作流程的關鍵步驟，其中包括從執行個體遭到入侵的程序和決策樹狀結構，直到分析和包含為止。

1. `SecurityIncidentStatus`標籤是否已使用「分析」值設定？ 如果是，請執行以下操作：
   a. 為 AWS Systems Manager 和 Amazon S3 附加正確的 IAM 設定檔。
   b. 將 Amazon SNS 消息發送到 Slack 中的 Amazon SNS 隊列。
   c. 將 Amazon SNS 消息發送到 `SecurityIncident`隊列。
   d. 叫用記憶體和磁碟擷取狀態機器。
2. 是否已取得記憶體和磁碟？ 如果沒有，則存在錯誤。
3. 使用標籤標記 EC2 執行個體。`Contain`
4. 附加 IAM 角色和安全群組以完全隔離執行個體。

自動化和規模

此模式的目的是提供可擴展的解決方案，在單一 AWS Organizations 組織內的多個帳戶執行事件回應和鑑識。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以保護資料。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。它也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。

Code

如需程式碼以及特定實作與使用指引，請參閱 GitHub 自動事件回應與鑑識架構儲存庫。

史诗

部署 CloudFormation 範本

任務	描述	所需技能
部署 CloudFormation 範本。	CloudFormation 範本會以指令碼名稱的第一個字標示為 1 到	AWS 管理員

任務	描述	所需技能
	7，指出需要在哪個帳戶部署範本。請注意，啟動 CloudFormation 範本的順序很重要。 • 1-forensic-AnalysisVPCnS3Buckets.yaml ：部署在鑑識帳戶中。它會建立 S3 儲存貯體和分析 VPC，然後啟動 CloudTrail。 • 2-forensic-MaintenanceVPCnEC2ImageBuilderPipeline.yaml ：根據 SANS SIFT 部署維護 VPC 和映像產生器管道。 • 3-security_IR-Disk_Mem_automation.yaml ：部署啟用磁碟和記憶體擷取的安全性帳戶中的功能。 • 4-security_LiME_Volatility_Factory.yaml ：啟動建置函數以根據指定的 AMI ID 開始建立記憶體模組。請注意，AMI ID 在 AWS 區域之間有所不同。每當您需要新的記憶體模組時，都可以使用新的 AMI ID 重新執行此指令碼。考慮將其與您的黃金映像 AMI 構建器管道集成（如果在您的環境中使用）。 • 5-member-IR-automation.yaml ：建立成員	

任務	描述	所需技能
	事件-回應自動化函數，以啟動事件回應程序。它允許跨帳戶共用 Amazon 彈性區塊存放區 (Amazon EBS) 磁碟區、在事件回應程序期間自動張貼到 Slack 通道、啟動鑑識程序，以及在程序完成後隔離執行個體。 • `6-forensic-artifact-s3-policies.yaml` ：部署完所有指令碼後，此指令碼會修正所有跨帳戶互動所需的權限。 • `7-security-IR-vpc.yaml` ：設定用於事件回應磁碟區處理的 VPC。 若要啟動特定 EC2 執行個體的事件回應架構，請使用金鑰`SecurityIncidentStatus` 和值建立標籤`Analyze`。這將啟動成員 Lambda 函數，該函數將自動啟動隔離和記憶體以及磁碟擷取。	

任務	描述	所需技能
操作框架。	Lambda 函數也會在最後 (或失敗時) 重新標記資產。`Contain`這會啟動遏制，完全隔離具有 INBOUND/ 輸出安全群組的執行個體，並具有不允許所有存取權的 IAM 角色。 按照GitHub 存儲庫中的步驟進行操作。	AWS 管理員

部署自訂 Security Hub 動作

任務	描述	所需技能
使用 CloudFormation 範本部署自訂 Security Hub 動作。	若要建立自訂動作，以便您可以使用 Security Hub 中的下拉式清單，請部署`Modules/SecurityHub Custom Actions/SecurityHubCustomActions.yaml` CloudFormation 範本。然後修改每個成員帳戶中的`IRAutomation` 角色，以允許執行動作的 Lambda 函數擔任該`IRAutomation` 角色。如需詳細資訊，請參閱存GitHub 放庫。	AWS 管理員

相關資源

- AWS 安全事件回應指南

其他資訊

透過使用此環境，資訊安全營運中心 (SOC) 團隊可透過下列方式改善其安全性事件回應流程：

- 能夠在隔離的環境中執行鑑識，以避免生產資源意外損壞
- 具有標準化、可重複、自動化的流程來進行遏制和分析。
- 讓任何帳戶擁有者或管理員能夠啟動事件回應程序，只要知道如何使用標籤的最低知識
- 擁有標準化、乾淨的環境，可執行事件分析和鑑識，而不會受到更大環境的噪音
- 能夠 parallel 建立多個分析環境
- 將 SOC 資源集中在事件回應上，而非雲端鑑識環境的維護和記錄
- 從手動流程轉向自動化流程以實現可擴展性
- 使用 CloudFormation 模板以保持一致性並避免重複的任務

此外，您可避免使用持續性基礎結構，並在需要時為資源付費。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

自動修復 AWS Security Hub 標準發現項目

創建者：錢迪尼香檳（AWS）和芳香拉吉傑亞拉揚（AWS）

環境：生產	技術：安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：AWS CloudFormation；Amazon CloudWatch；AWS Lambda；AWS Security Hub；Amazon SNS		

Summary

使用 AWS Security Hub，您可以啟用標準最佳實務的檢查，例如：

- AWS 基礎安全最佳實務
- 獨聯體 AWS 基礎基準
- 支付卡產業資料安全標準 (PCI DSS)

每個標準都有預先定義的控制項。Security Hub 會檢查指定 AWS 帳戶中的控制項，並報告發現結果。

AWS Security Hub 預設會將所有發現項目傳 EventBridge 送給 Amazon。此模式提供安全控制，可部署 EventBridge 規則以識別 AWS 基礎安全最佳實務標準發現項目。此規則根據 AWS 基礎安全最佳實務標準識別自動擴展、虛擬私有雲端 (VPC)、Amazon 彈性區塊存放區 (Amazon EBS) 和 Amazon 關聯式資料庫服務 (Amazon RDS) 的下列發現項目：

- [AutoScaling.1] 與負載平衡器關聯的 Auto Scaling 群組應使用負載平衡器健康狀態檢查
- [EC2.2] VPC 預設安全群組不應允許傳入和傳出流量
- [EC2.6] 應在所有 VPC 中啟用虛擬私人雲端流程記錄功能
- [EC2.7] 應啟用 EBS 預設加密功能
- [RDS.1] RDS 快照應為私有
- [RDS.6] 應為 RDS 資料庫執行個體和叢集設定增強型監控
- [RDS.7] RDS 叢集應該已啟用刪除保護功能

此 EventBridge 規則會將這些發現項目轉寄至 AWS Lambda 函數，以修正發現項目。然後，Lambda 函數會將包含修復資訊的通知傳送至亞馬遜簡單通知服務 (Amazon SNS) 主題。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 您想要接收補救通知的電子郵件地址
- 在您打算部署控制的 AWS 區域中啟用 Security Hub 和 AWS Config
- 亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體與上傳 AWS Lambda 程式碼的控制項位於相同區域

限制

- 此安全控制會自動修正安全控制部署後報告的新發現項目。若要修復現有的發現項目，請在 Security Hub 主控台上手動選取發現項目。然後，在「動作」下，選取 AWS 在部署過程中建立的 AFSBPremedy 自訂動作。 CloudFormation
- 此安全控制是區域性的，必須部署在您想要監控的 AWS 區域中。
- 對於 EC2.6 補救措施，若要啟用 VPC 流程日誌，將以/VpcFlowLogs/vpc_id 格式建立一個 Amazon CloudWatch 日誌日誌群組。如果具有相同名稱的記錄群組存在，將會使用現有的記錄群組。
- 對於 EC2.7 的補救措施，若要啟用 Amazon EBS 預設加密，則會使用預設的 AWS Key Management Service (AWS KMS) 金鑰。此變更會防止使用不支援加密的某些執行個體。

架構

目標技術堆疊

- Lambda 函數
- Amazon SNS 主題
- EventBridge 規則
- 適用於 Lambda 函數、VPC 流程日誌和 Amazon Relational Database Service 服務 (Amazon RDS) 增強型監控的 AWS Identity and Access Management (IAM) 角色

目標架構

自動化和規模

如果您使用 AWS Organizations，則可以使用 AWS CloudFormation StackSets 將此範本部署在多個您希望監控此範本的帳戶中。

工具

工具

- AWS CloudFormation — AWS CloudFormation 是一項服務，可協助您使用基礎設施即程式碼來建模和設定 AWS 資源。
- EventBridge— Amazon 從您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務 EventBridge 提供即時資料串流，並將該資料路由到 Lambda 函數等目標。
- Lambda — AWS Lambda 支援執行程式碼，無需佈建或管理伺服器。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

最佳實務

- 九個 AWS Security Hub 最佳實務
- AWS 基礎安全最佳實務標準

史诗

部署安全控制

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。S3	雲端架構師

任務	描述	所需技能
	儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。您的 S3 儲存貯體必須與正在評估的安全中樞發現項目位於相同的區域。	
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」一節中提供的 Lambda 程式碼 .zip 檔案上傳到定義的 S3 儲存貯體。	雲端架構師
部署 AWS CloudFormation 範本。	將以附件形式提供的 AWS CloudFormation 範本部署到此模式。在下一個史詩中，提供參數的值。	雲端架構師

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 Amazon S3 前綴。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如<directory>/<file-name>.zip)。	雲端架構師
提供 SNS 主題 ARN。	如果您想要使用現有的 SNS 主題進行修復通知，請提供 SNS 主題 Amazon 資源名稱 (ARN)。若要使用新的 SNS 主題，請將該值保持為「無」(預設值)。	雲端架構師
提供電子郵件地址。	提供您想要接收修復通知的電子郵件地址 (只有當您希望	雲端架構師

任務	描述	所需技能
	AWS CloudFormation 建立 SNS 主題時才需要)。	
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。「信息」指定了應用程序的進度的詳細信息消息。「Error」指定仍可允許應用程式繼續執行的錯誤事件。「警告」表示可能有害的情況。	雲端架構師
提供 VPC 流程記錄 IAM 角色 ARN。	提供要用於虛擬私人 VPC 流程記錄的 IAM 角色 ARN。(如果輸入「無」輸入，AWS CloudFormation 會建立 IAM 角色並使用它。)	雲端架構師
提供 RDS 增強型監控 IAM 角色 ARN。	提供要用於 RDS 增強型監控的 IAM 角色 ARN。如果輸入「無」，AWS CloudFormation 會建立 IAM 角色並使用它。)	雲端架構師

確認訂閱

任務	描述	所需技能
確認 Amazon SNS 訂閱。	成功部署範本時，如果建立了新的 SNS 主題，則會將訂閱訊息傳送至您提供的電子郵件地址。若要接收補救通知，您必須確認此訂閱電子郵件訊息。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊

- AWS Lambda
- AWS 安全中樞

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Amazon Inspector 和 AWS Security Hub 自動執行跨帳戶工作負載的安全掃描

由拉米亞·普利帕卡（AWS）和米克什漢納爾（AWS）創建

環境：生產	技術：安全性、身分識別、合規性；營運	AWS 服務：Amazon Inspector；Amazon SNS；AWS Lambda；AWS Security Hub；Amazon CloudWatch

Summary

此模式說明如何在 Amazon Web Services (AWS) 雲端上自動掃描跨帳戶工作負載中的弱點。

此模式有助於針對 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的主機掃描建立排程，這些執行個體按標籤分組或用於網路型 Amazon Inspector 掃描。AWS CloudFormation 堆疊會將所有必要的 AWS 資源和服務部署到您的 AWS 帳戶。

Amazon Inspector 調查結果會匯出到 AWS Security Hub，並針對您的帳戶、AWS 區域、虛擬私有雲 (VPC) 和 EC2 執行個體的漏洞提供深入解析。您可以透過電子郵件接收這些發現項目，或者建立 Amazon Simple Notification Service (Amazon SNS) 主題，該主題使用 HTTP 端點將發現項目傳送至票務工具、安全資訊和事件管理 (SIEM) 軟體或其他第三方安全解決方案。

先決條件和限制

先決條件

- 用於接收來自 Amazon SNS 的電子郵件通知的現有電子郵件地址。
- 票務工具、SIEM 軟體或其他第三方安全性解決方案所使用的現有 HTTP 端點。
- 託管跨帳戶工作負載的作用中 AWS 帳戶，包括中央稽核帳戶。
- Security Hub，已啟用和已設定。您可以在沒有安全性中樞的情況下使用此模式，但我們建議您使用 Security Hub，因為它會產生深入解析。如需詳細資訊，請參閱 AWS Security Hub 文件中的設定安全中心。
- 您必須在要掃描的每個 EC2 執行個體上安裝 Amazon Inspector 代理程式。您可以使用 AWS Systems Manager 執行命令，在多個 EC2 執行個體上安裝 Amazon Inspector 代理程式。

技能

- AWS 中堆疊集的使用`self-managed`和`service-managed`許可的經驗 CloudFormation。如果您想要使用`self-managed`許可將堆疊執行個體部署到特定區域中的特定帳戶，則必須建立所需的 AWS Identity and Access Management (IAM) 角色。如果您想要使用`service-managed`許可將堆疊執行個體部署到特定區域的 AWS Organizations 管理的帳戶，則不需要建立必要的 IAM 角色。如需詳細資訊，請參閱 AWS CloudFormation 文件中的建立堆疊集。

限制

- 如果帳戶中的 EC2 執行個體沒有標籤套用，則 Amazon Inspector 會掃描該帳戶中的所有 EC2 執行個體。
- AWS CloudFormation 堆疊集和 onboard-audit-account .yaml 檔案 (附加) 必須部署在相同的區域中。
- 根據預設，Amazon Inspector 經典版不支援彙總的發現項目。Security Hub 是檢視多個帳戶或 AWS 區域評估的建議解決方案。
- 此模式的方法可在美國東部 (維吉尼亞北部) 區域 (us-east-1) SNS 主題 (us-east-1) 每秒 30,000 個交易 (TPS) 的發佈配額下進行調整，但限制會因區域而異。為了更有效地擴展並避免資料遺失，我們建議在 SNS 主題前面使用 Amazon Simple Queue Service (Amazon SQS)。

架構

下圖說明自動掃描 EC2 執行個體的工作流程。

工作流程由以下步驟組成：

1. Amazon EventBridge 規則使用 cron 表達式根據特定時間表自行啟動並啟動 Amazon Inspector。

2. Amazon Inspector 掃描帳戶中標記的 EC2 實例。

3. Amazon Inspector 會將調查結果傳送到 Security Hub，這會產生工作流程、排定優先順序和修復的見解。

4. Amazon Inspector 也會將評估的狀態傳送至稽核帳戶中的 SNS 主題。如果將`findings reported`事件發佈到 SNS 主題，就會叫用 AWS Lambda 函數。

5. Lambda 函數會擷取、格式化並將發現結果傳送至稽核帳戶中的另一個 SNS 主題。

6. 發現項目會傳送至訂閱 SNS 主題的電子郵件地址。完整詳細資料和建議會以 JSON 格式傳送至訂閱的 HTTP 端點。

技術, 堆

- AWS Control Tower
- EventBridge
- IAM
- Amazon Inspector
- Lambda
- 安全中樞
- Amazon SNS

工具

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定，以減少管理這些資源的時間，將更多時間專注於應用程式。
- AWS CloudFormation StackSets — AWS 可讓您透過單一操作跨多個帳戶和區域建立、更新或刪除堆疊來 CloudFormation StackSets 擴展堆疊的功能。
- AWS Control Tower — AWS Control Tower 會建立抽象或協調層，結合並整合其他 AWS 服務 (包括 AWS Organizations) 的功能。
- Amazon EventBridge — EventBridge 是一種無伺服器事件匯流排服務，可讓您輕鬆地將應用程式與來自各種來源的資料連接起來。
- AWS Lambda — L ambda 是一種運算服務，可協助您執行程式碼，而無需佈建或管理伺服器。
- AWS Security Hub — Security Hub 為您提供 AWS 安全狀態的全面檢視，並協助您根據安全產業標準和最佳實務檢查環境。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者的訊息傳遞。

史诗

部署 AWS CloudFormation 範本

任務	描述	所需技能
在稽核帳戶中部署 AWS CloudFormation 範本。	將`onboard-audit-account.yaml` 檔案 (附加) 下載並儲存到電腦上的本機路徑。 登入稽核帳戶的 AWS 管理主控台，開啟 AWS CloudFormation 主控台，然後選擇 [建立堆疊]。 在「先決條件」段落中選擇準備樣板，然後選擇樣板已就緒。在「指定範本」區段中選擇「範本來源」，然後選擇「範本已就緒」。上傳`onboard-audit-account.yaml` 檔案，然後根據您的需求設定其餘選項。 重要：請務必設定下列輸入參數： • `DestinationEmailAddress` — 輸入電子郵件地址以接收發現。 • `HTTPEndpoint` — 為您的票務或 SIEM 工具提供 HTTP 端點。 您也可以使用 AWS Command Line Interface (AWS CLI) (AWS CLI) 部署 AWS CloudFormation 範本。如	開發者、安全工程師

任務	描述	所需技能
	需詳細資訊，請參閱 AWS CloudFormation 文件中的建立堆疊。	
確認 Amazon SNS 訂閱。	開啟電子郵件收件匣，然後在從 Amazon SNS 收到的電子郵件中選擇「確認訂閱」。這會開啟網頁瀏覽器視窗，並顯示訂閱確認。	開發者、安全工程師

建立 AWS CloudFormation 堆疊集以自動執行 Amazon Inspector 掃描排程

任務	描述	所需技能
在稽核帳戶中建立堆疊集。	將`vulnerability-management-program.yaml` 文件（附加）下載到計算機上的本地路徑。 在 AWS CloudFormation 主控台上，選擇 [檢視堆疊集]，然後選擇 [建立 StackSet]。選擇 [範本已準備就緒]，選擇 [上傳範本檔案]，然後上傳**`vulnerability-management-program.yaml`** 檔案。 如果您想要使用`self-managed` 許可，請按照 AWS CloudFormation 文件中的使用自我管理許可建立堆疊集中的指示進行操作。這會在個別帳戶中建立堆疊集。	開發者、安全工程師

任務	描述	所需技能
	如果您想要使用`service-managed` 許可，請按照 AWS CloudFormation 文件中的使用服務管理許可建立堆疊集中的指示進行操作。這會在整個組織或指定的組織單位 (OU) 中建立堆疊集。 重要事項：請確定已針對堆疊集設定下列輸入參數： • `AssessmentSchedule` — EventBridge 使用 cron 運算式的排程。 • `Duration`— Amazon Inspector 評估的持續時間以秒為單位運行。 • `CentralSNSTopicArn` — 中央 SNS 主題的 Amazon 資源名稱 (ARN)。 • `Tagkey`— 與資源群組相關聯的標籤鍵。 • `Tagvalue`— 與資源群組相關聯的標籤值。 如果要掃描稽核帳戶中的 EC2 執行個體，則必須在稽核帳戶中以 AWS CloudFormation 堆疊的形式執行該`vulnerability-management-program.yaml` 檔案。	

任務	描述	所需技能
驗證解決方案。	根據您為 Amazon Inspector 指定的排程，檢查您是否透過電子郵件或 HTTP 端點收到發現項目。	開發者、安全工程師

相關資源

- 使用 Amazon Inspector 擴展安全漏洞測試
- 自動修復 Amazon Inspector 安全發現
- 如何使用亞馬遜 EC2，AWS Systems Manager 器和亞馬 Amazon Inspector 簡化安全評估設置

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

在 AWS Config 中使用 CloudTrail 自訂修復規則自動重新啟用 AWS

由馬尼甘丹希里（AWS）創建

環境：生產	技術：基礎架構；營運；安全性、身分識別、合規性	AWS 服務：Amazon S3；AWS Config；AWS KMS；AWS Identity and Access Management；AWS Systems Manager；AWS CloudTrail

Summary

您 Amazon Web Services (AWS) 帳戶中的活動能見度是一項重要的安全性和營運最佳實務。AWS 可 CloudTrail 協助您進行帳戶的管理、合規以及操作和風險稽核。

為了確保您的帳戶保 CloudTrail 持啟用狀態，AWS Config 提供了`cloudtrail-enabled`受管規則。如果 CloudTrail 已關閉，`cloudtrail-enabled`則規則會使用自動補救來自動重新啟用它。

但是，如果您使用自動補救， CloudTrail 則必須確保遵循安全性最佳做法。這些最佳實務包括 CloudTrail 在所有 AWS 區域啟用、記錄讀取和寫入工作負載、啟用見解，以及使用 AWS Key Management Service (AWS KMS) 受管金鑰 (SSE-KMS) 以伺服器端加密來加密日誌檔。

此模式可提供自訂補救動作，以便 CloudTrail 在帳戶中自動重新啟用，協助您遵循這些安全性最佳做法。

重要事項：我們建議您使用服務控制政策 (SCP) 來防止任何竄改。 CloudTrail如需詳細資訊，請參閱 AWS 安全 CloudTrail部落格如何使用 AWS Organizations 簡化大規模安全防護部落格中的防止 AWS 竄改一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 建立 AWS Systems Manager Automation 手冊的許可
- 您帳戶的現有追蹤

限制

此模式不支援下列動作：

- 為儲存位置設定 Amazon 簡單儲存服務 (Amazon S3) 前置碼金鑰
- 發佈到 Amazon Simple Notification Service (Amazon SNS) 主題
- 設定 Amazon CloudWatch 日誌以監控您的 CloudTrail 日誌

架構

技術, 堆

- AWS Config
- CloudTrail
- Systems Manager
- Systems Manager Automation

工具

- AWS Config 提供您帳戶中 AWS 資源組態的詳細檢視。
- AWS 可 CloudTrail協助您啟用帳戶的管理、合規以及操作和風險稽核。
- AWS Key Management Service (AWS KMS) 是一種加密和金鑰管理服務。
- AWS Systems Manager 可協助您在 AWS 上檢視和控制基礎設施。
- AWS Systems Manager Automation 可簡化 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和其他 AWS 資源的常見維護和部署任務。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

雲路補救動作 .yml 檔案 (附件) 可協助您建立 Systems Manager 自動化執行手冊，以使用安全性最佳實務來設定和重新啟用。 CloudTrail

史诗

配置 CloudTrail

任務	描述	所需技能
建立 S3 儲存貯體。	登入 AWS 管理主控台，開啟 Amazon S3 主控台，然後建立 S3 儲存貯體來存放 CloudTrail 日誌。如需詳細資訊，請參閱 Amazon S3 文件中的建立 S3 儲存貯體。	系統管理員
新增儲存貯體政策以 CloudTrail 允許將日誌檔傳遞到 S3 儲存貯體。	CloudTrail 必須具有必要的許可才能將日誌檔傳遞到 S3 儲存貯體。在 Amazon S3 主控台上，選擇您先前建立的 S3 儲存貯體，然後選擇許可。使用 CloudTrail 文件 CloudTrail中的 Amazon S3 儲存貯體政策建立 S3 儲存貯體政策。 如需如何將政策新增至 S3 儲存貯體的步驟，請參閱 Amazon S3 文件中的使用 Amazon S3 主控台新增儲存貯體政策。 重要事項：如果您在中建立追蹤時指定了前置詞 CloudTrail，請務必將其包含在 S3 儲存貯體政策中。前置詞是 S3 物件金鑰的選用新增項目，可在 S3 儲存貯體中建立類似資料夾的組織。如需有關此項目的詳細資訊，請參閱 CloudTrail 文件中的建立追蹤。	系統管理員

任務	描述	所需技能
建立 KMS 金鑰。	建立 AWS KMS 金鑰，以便在將物件新增 CloudTrail 至 S3 儲存貯體之前加密物件。如需本案例的說明，請參閱文件中的使用 AWS KMS 受管金鑰 (SSE-KMS) 加密 CloudTrail 記錄檔。 CloudTrail	系統管理員
將金鑰原則新增至 KMS 金鑰。	附加 KMS 金鑰原則以允 CloudTrail 許使用 KMS 金鑰。如需本案例的說明，請參閱文件中的使用 AWS KMS 受管金鑰 (SSE-KMS) 加密 CloudTrail 日誌檔。 CloudTrail 重要事項： CloudTrail 不需要`Decrypt`權限。	系統管理員
AssumeRole 為系統管理員工作手冊建立	創建一`AssumeRole` 個 Systems Manager 自動化運行手冊。如需相關指示和詳細資訊，請參閱 Systems Manager 說明文件中的設定自動化。	系統管理員

建立並測試系 Systems Manager 自動化手冊

任務	描述	所需技能
建立系 Systems Manager 自動化手冊。	使用`cloudtrail-remediation-action.yml` 檔案 (附加) 建立 Systems Manager 自動化工作手冊。如需這方面的詳細資訊，請參閱 Systems	系統管理員

任務	描述	所需技能
	Manager 說明文件中的建立 Systems Manager 文件。	
測試執行手冊。	在 Systems Manager 主控台上，測試您先前建立的 Systems Manager 自動化執行手冊。如需這方面的詳細資訊，請參閱 Systems Manager 說明文件中的執行簡單自動化。	系統管理員

在 AWS 設定中設定自動修復規則

任務	描述	所需技能
新增 CloudTrail已啟用的規則。	在 AWS Config 主控台上，選擇「規則」，然後選擇「新增規則」。在 Add rule (新增規則) 頁面中，選擇 Add custom rule (新增自訂規則)。在 [設定規則] 頁面上，輸入名稱和說明，然後新增`cloudtrail-enabled` 規則。如需詳細資訊，請參閱 AWS Config 文件中的管理 AWS 組態規則。	系統管理員
新增自動修復動作。	從 動作 下拉式清單中，選擇 管理修復。選擇 [自動修復]，然後選擇您先前建立的 Systems Manager 手冊。 以下是所需的輸入參數 CloudTrail： • `CloudTrailName`	系統管理員

任務	描述	所需技能
	• CloudTrailS3BucketName • CloudTrailKmsKeyId • AssumeRole （選用） 依預設，下列輸入參數會設定為 true： • IsMultiRegionTrail • IsOrganizationTrail • IncludeGlobalServiceEvents • EnableLogFileValidation 保留「速率限制」參數和「資源 ID」參數的預設值。選擇儲存。 如需詳細資訊，請參閱 AWS Config 文件中的使用 AWS Config 規則修復不合規的 AWS Config 資源。	

任務	描述	所需技能
測試自動修復規則。	若要測試自動修復規則，請開啟 CloudTrail 主控台，選擇 [追蹤]，然後選擇追蹤。選擇停止記錄日誌以關閉追蹤記錄日誌。當系統提示您確認時，請選擇 [停止記錄]。 CloudTrail 停止該追蹤的記錄活動。 遵循 AWS Config 文件中評估資源的指示，確定 CloudTrail 已自動重新啟用該功能。	系統管理員

相關資源

配置 CloudTrail

- 建立 S3 儲存貯體
- Amazon S3 存儲桶政策 CloudTrail
- 使用 Amazon S3 主控台新增儲存貯體政策
- 建立系統線
- 設定自動化
- 使用 AWS KMS 受管金鑰 (SSE-KMS) 加密 CloudTrail 日誌檔

建立並測試系 Systems Manager 自動化手冊

- 建立 Systems Manager 文件
- 執行簡易自動化

在 AWS 設定中設定自動修復規則

- 管理您的 AWS Config 規則
- 使用 AWS 組態規則修復不合規的 AWS 資源

其他資源

- AWS CloudTrail -安全最佳實務
- 開始使用 AWS Systems Manager
- 開始使用 AWS Config
- 開始使用 AWS CloudTrail

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

自動修復未加密的 Amazon RDS 資料庫執行個體和叢集

由阿傑·拉瓦特（AWS）和喬希·喬希（AWS）創建

環境：PoC 或試點	技術：安全性、身分識別、合規性；資料庫	AWS 服務：AWS Config；AWS KMS；AWS Identity and Access Management；AWS Systems Manager；Amazon RDS

Summary

此模式說明如何使用 AWS Config、AWS Systems Manager 手冊和 AWS Key Management Service (AWS KMS) 金鑰，自動修復 Amazon Web Services (AWS) 上未加密的 Amazon 關聯式資料庫服務 (Amazon RDS) 資料庫執行個體和叢集。

加密的 RDS 資料庫執行個體可保護您的資料，防止未經授權存取基礎儲存體，進而提供額外的資料保護層。您可以使用 Amazon RDS 加密來增加 AWS 雲端中部署之應用程式的資料保護，並滿足靜態加密的合規要求。您可以在建立 RDS 資料庫執行個體時為其啟用加密，但在建立之後則無法啟用加密。不過，您可以建立資料庫執行個體的快照，然後建立該快照的加密副本，將加密新增至未加密的 RDS 資料庫執行個體。然後，您可以從加密的快照還原資料庫執行個體，以取得原始資料庫執行個體的加密副本。

此模式使用 AWS Config 規則來評估 RDS 資料庫執行個體和叢集。它會使用 AWS Systems Manager 執行手冊 (定義要在不合規的 Amazon RDS 資源上執行的動作) 和 AWS KMS 金鑰來加密資料庫快照來套用修復。然後，它會強制執行服務控制政策 (SCP)，以防止在未加密的情況下建立新的資料庫執行個體和叢集。

此模式的程式碼在中提供GitHub。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 此病毒碼的GitHub 來源程式碼儲存庫中的檔案已下載至您的電腦

- 未加密的 RDS 資料庫執行個體或叢集
- 用於加密 RDS 資料庫執行個體和叢集的現有 AWS KMS 金鑰
- 更新 KMS 金鑰資源原則的存取權
- AWS 組態已在您的 AWS 帳戶中啟用 (請參閱 AWS 文件中的 AWS Config 入門使用)

限制

- 您只能在建立 RDS 資料庫執行個體時啟用加密，而不能在建立之後啟用加密。
- 未加密資料庫執行個體不可以有加密僅供讀取複本，加密資料庫執行個體也不可以有未加密僅供讀取複本。
- 您無法將未加密的備份或快照還原至已加密的資料庫執行個體。
- 大多數資料庫執行個體類別可以使用 Amazon RDS 加密。如需例外清單，請參閱 Amazon RDS 文件中的加密 Amazon RDS 資源。
- 若要將加密快照從一個 AWS 區域複製到另一個區域，您必須在目的地 AWS 區域中指定 KMS 金鑰。這是因為 KMS 金鑰專屬於建立它們所在的 AWS 區域。
- 在整個複製過程中來源快照仍會保持加密狀態。Amazon RDS 在複製過程中使用信封加密來保護資料。如需詳細資訊，請參閱 AWS KMS 文件中的信封加密。
- 您無法解密加密已加密的資料庫執行個體。不過，您可以從加密的資料庫執行個體匯出資料，然後將資料匯入未加密的資料庫執行個體。
- 只有在確定不再需要使用 KMS 金鑰時，才應刪除該金鑰。如果您不確定，請考慮停用 KMS 金鑰，而不是刪除它。如果您稍後需要再次使用已停用的 KMS 金鑰，可以重新啟用該金鑰，但無法復原已刪除的 KMS 金鑰。
- 如果您不選擇保留自動備份，則會刪除與資料庫執行個體位於相同 AWS 區域的自動備份。刪除資料庫執行個體後，便無法復原自動備份內容。
- 您的自動備份會保留在您刪除資料庫執行個體時在資料庫執行個體上設定的保留期間內。無論您是否選擇建立最終資料庫快照，都會依照此一設定保留期間。
- 如果啟用自動修復，此解決方案會加密具有相同 KMS 金鑰的所有資料庫。

架構

下圖說明 AWS CloudFormation 實作的架構。請注意，您也可以使用 AWS Cloud Development Kit (AWS CDK) 來實作此模式。

工具

工具

- AWS 可 CloudFormation協助您自動設定 AWS 資源。它使您可以使用模板文件來創建和刪除資源集合作為一個單元（堆棧）一起。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可在程式碼中定義雲端基礎設施，並使用熟悉的程式設計語言進行佈建。

AWS 服務和功能

- AWS Config 會追蹤 AWS 資源的組態，以及它們與其他資源的關係。它也可以評估這些 AWS 資源的合規性。此服務使用可設定的規則，根據所需的組態評估 AWS 資源。您可以針對常見的合規案例使用一組 AWS Config 受管規則，也可以為自訂案例建立自己的規則。當發現 AWS 資源不合規時，您可以透過 AWS Systems Manager 工作手冊指定修復動作，並選擇性地透過 Amazon Simple Notification Service (Amazon SNS) 主題傳送警示。換句話說，您可以將修復動作與 AWS Config 規則建立關聯，然後選擇自動執行它們以解決不合規的資源，而無需手動介入。如果資源在自動修復之後仍不合規，您可以將規則設定為再次嘗試自動修復。
- Amazon Relational Database Service (Amazon RDS) 可讓您更輕鬆地在雲端中設定、操作和擴展關聯式資料庫。Amazon RDS 的基本建置區塊是資料庫執行個體，這是 AWS 雲端中的隔離資料庫環境。Amazon RDS 提供一系列針對不同關聯式資料庫使用案例進行優化的執行個體類型。執行個體類型包含 CPU、記憶體、儲存和網路容量的各種組合，可讓您彈性地為資料庫選擇適當的資源組合。每個執行個體類型都包含多種執行個體大小，可讓您根據目標工作負載的需求擴展資料庫。
- AWS Key Management Service (AWS KMS) 是一項受管服務，可讓您輕鬆建立和控制 AWS KMS 金鑰，以加密資料。KMS 金鑰是根金鑰的邏輯表示法。KMS 金鑰包含金鑰 ID、建立日期、說明和金鑰狀態等中繼資料。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- 服務控制原則 (SCP) 可讓您集中控制組織中所有帳戶的最大可用權限。SCP 可協助您確保帳戶符合組織的存取控制準則。SCP 不會影響管理帳戶中的使用者或角色。它們只會影響組織中的成員帳戶。在沒有將政策對帳戶的影響進行徹底測試之前，我們強烈建議您不要將 SCP 連接到組織的根帳戶。而是建立一個組織單位 (OU)，您可以將您的帳戶一次移到一個帳戶中，或至少以少量的方式移動帳戶，以確保您不會意外將使用者鎖定在金鑰服務之外。

Code

此模式的原始程式碼和範本可在GitHub 儲存庫中取得。該模式提供兩種實作選項：您可以部署 AWS CloudFormation 範本來建立修復角色，以加密 RDS 資料庫執行個體和叢集，或使用 AWS CDK。存放庫具有這兩個選項的單獨資料夾。

「史詩」區段提供部署 CloudFormation 範本的 step-by-step 指示。如果您想要使用 AWS CDK，請依照儲存庫中 README.md 檔案中的指示操作。 GitHub

最佳實務

- 啟用靜態和傳輸中的資料加密。
- 在所有帳戶和 AWS 區域啟用 AWS 組態。
- 記錄對所有資源類型的配置變更。
- 定期輪替您的 IAM 登入資料。
- 利用 AWS Config 的標記功能，讓管理、搜尋和篩選資源變得更加容易。

史诗

建立 IAM 補救角色和 AWS Systems Manager 手冊

任務	描述	所需技能
下載 CloudFormation 範本。	從GitHub 存放庫下載`unencrypted-to-encrypted-rds.template.json` 檔案。	DevOps 工程師
建立 CloudFormation 堆疊。	1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/cloudformation/ 開啟 CloudFormation 主控台。 2. 啟動`unencrypted-to-encrypted-rds.template.json` 範本以建立新堆疊。	DevOps 工程師

任務	描述	所需技能
	如需部署範本的詳細資訊，請參閱 AWS CloudFormation 文件。	
檢閱 CloudFormation 參數和值。	1. 檢閱堆疊詳細資料並根據您的環境需求更新值。 2. 選擇 [建立堆疊] 以部署範本。	DevOps 工程師
檢閱資源。	建立堆疊後，其狀態會變更為「建立 _ 完成」。在 CloudFormation 主控台中檢閱建立的資源 (IAM 角色、AWS Systems Manager 工作手冊)。	DevOps 工程師

更新 AWS KMS 金鑰政策

任務	描述	所需技能
更新您的 KMS 金鑰政策。	1. 請確定金鑰別名`alias/RDSEncryptionAtRestKMSAlias` 存在。 2. 金鑰政策聲明應包含 IAM 修復角色。（檢查您在上一個史詩中部署的 CloudFormation 模板創建的資源。） 3. 在下列金鑰政策中，請更新粗體顯示的部分，以符合您的帳戶和建立的 IAM 角色。 `{ "Sid": "Allow access through RDS for all principals`	DevOps 工程師

任務	描述	所需技能
	in the account that are authorized to use RDS", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam:: <your-AWS-account-ID>”:role/<your-IAM-remediation-role>" }, "Action": ["kms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*", "kms:GenerateDataKey*", "kms:CreateGrant", "kms:ListGrants", "kms:DescribeKey"], "Resource": "*", "Condition": { "StringEquals": { "kms:ViaService": "rds.us-east-1.amazonaws.com", "kms:CallerAccount": "<your-AWS-account-ID>" } } }	

尋找並修正不符合標準的資源

任務	描述	所需技能
檢視不符合標準的資源。	1. 若要檢視不合規資源的清單，請在 https://console.aws.amazon.com/config/ 開啟 AWS Config 主控台。 2. 在功能窗格中，選擇 [規則]，然後選擇規`rds-storage-encrypted` 則。 AWS Config 主控台中列出的不合規資源將是執行個體，而不是叢集。修復自動化會加密執行個體和叢集，並建立新加密的執行個體或新建立的叢集。但是，請確定不要同時修復屬於同一叢集的多個執行個體。 在修復任何 RDS 資料庫執行個體或磁碟區之前，請確定 RDS 資料庫執行個體未使用中。請確認建立快照時未發生任何寫入作業，以確保快照包含原始資料。請考慮強制執行維護時段，在此期間執行修復。	DevOps 工程師
修復不符合標準的資源。	1. 當您準備好且維護時段生效時，請選擇要修復的資源，然後選擇 [修正]。 動作狀態欄現在應該會顯示已排入佇列的動作執行。	DevOps 工程師

任務	描述	所需技能
	2. 在 Systems Manager 中檢視修復的進度和狀態。開啟 AWS Systems Manager 主控台，網址為 https://console.aws.amazon.com/systems-manager/。在瀏覽窗格中，選擇 [自動化]，然後選取對應自動化的執行 ID，以檢視進一步的詳細資訊。	
確認 RDS 資料庫執行個體可用。	自動化完成後，即可使用新加密的 RDS 資料庫執行個體。加密的 RDS 資料庫執行個體將具有前置詞，`encrypted` 後面接著原始名稱。例如，如果未加密的 RDS 資料庫執行個體名稱為`database-1` ，則新加密的 RDS 資料庫執行個體將`encrypted-database-1` 為。	DevOps 工程師
終止未加密的執行個體。	修復完成並驗證新加密的資源後，您可以終止未加密的執行個體。在終止任何資源之前，請務必確認新加密的資源符合未加密的資源。	DevOps 工程師

強制執行 SCP

任務	描述	所需技能
強制執行 SCP。	強制執行 SCP 以防止 future 在未經加密的情況下建立資料庫執行個體和叢集。	安全工程師

任務	描述	所需技能
	使用GitHub 儲存庫中提供的`rds_encrypted.json` 檔案以達到此目的，並遵循AWS 文件中的指示進行。	

相關資源

參考

- 設定 AWS Config
- AWS Config 自訂規則
- AWS KMS 概念
- AWS Systems Manager 文件
- 服務控制政策

工具

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)

指南和模式

- 在 AWS Config 中使用 CloudTrail 自訂修復規則自動重新啟用 AWS

其他資訊

常見問答集

問：AWS Config 如何運作？

答：當您開啟 AWS Config 時，它會先探索您帳戶中存在的受支援 AWS 資源，並為每個資源產生一個組態項目。AWS Config 也會在資源組態變更時產生組態項目，並在您啟動組態記錄器時保留資源組態項目的歷史記錄。根據預設，AWS Config 會為 AWS 區域中每個受支援的資源建立組態項目。如果您不希望 AWS Config 為所有支援的資源建立組態項目，可以指定要追蹤的資源類型。

問：AWS 組態和 AWS 組態規則與 AWS Security Hub 有何關聯？

答：AWS Security Hub 是一種安全和合規服務，可提供安全和合規狀態管理即服務。它使用 AWS Config 和 AWS Config 規則做為評估 AWS 資源組態的主要機制。AWS Config 規則也可用於直接評估資源組態。其他 AWS 服務也會使用 Config 規則，例如 AWS Control Tower 和 AWS Firewall Manager。

使用 AWS Organizations 和 AWS Secrets Manager 自動輪換大規模的 IAM 使用者存取金鑰

創建者：崔西·希基（AWS），高拉夫維爾瑪（AWS），勞拉·塞萊托斯（AWS），邁克爾·戴維（AWS）和阿文德·帕特爾（AWS）

環境：PoC 或試點	技術：安全性、身分識別、合規性	AWS 服務：AWS CloudFormation；Amazon CloudWatch 活動；AWS Identity and Access Management；AWS Lambda；AWS Organizations；Amazon S3；Amazon SES；AWS Secrets Manager

Summary

重要事項：AWS 建議您使用 AWS Identity and Access Management (IAM) 角色，而不要使用具有長期登入資料 (例如存取金鑰) 的 IAM 使用者。此模式中記錄的方法僅適用於需要長期使用 AWS API 登入資料的舊版實作。對於這些實作，我們仍建議您考慮使用短期登入資料的選項，例如使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體設定檔或 IAM 角色隨處使用。本文中的方法僅適用於您無法立即變更為使用短期憑證的情況，而且您需要按排程輪替長期認證。透過這種方法，您必須負責定期更新舊版應用程式程式碼或設定，以使用輪換的 API 認證。

存取金鑰是 IAM 使用者的長期登入資料。定期輪換您的 IAM 登入資料有助於防止遭到入侵的 IAM 存取金鑰集存取 AWS 帳戶中的元件。輪換 IAM 登入資料也是 IAM 安全最佳實務的重要部分。

此模式可協助您使用 IAM 金鑰輪換儲存庫中提供的 AWS CloudFormation 範本，自動輪換 GitHub IAM 存取金鑰。

此模式支援在單一帳戶或多個帳戶中進行部署。如果您使用 AWS Organizations Organization，此解決方案會識別組織內的所有 AWS 帳戶 ID，並在帳戶移除或建立新帳戶時動態擴展。集中式 AWS Lambda 函數使用假設的 IAM 角色，在您選取的多個帳戶之間在本機執行輪換函數。

- 當現有存取金鑰已有 90 天時，就會產生新的 IAM 存取金鑰。

- 新的存取金鑰會以密碼形式存放在 AWS Secrets Manager 中。以資源為基礎的政策只允許指定的 IAM 主體存取和擷取密碼。如果您選擇將金鑰儲存在管理帳戶中，所有帳戶的金鑰都會儲存在管理帳戶中。
- 指派給建立新存取金鑰之 AWS 帳戶擁有者的電子郵件地址會收到通知。
- 先前的存取金鑰會在 100 天前停用，然後在 110 天刪除。
- 集中式電子郵件通知會傳送給 AWS 帳戶擁有者。

Lambda 函數和 Amazon CloudWatch 會自動執行這些動作。然後，您可以擷取新的存取 key pair，並在程式碼或應用程式中取代它們。您可以自訂輪替、刪除和停用期間。

先決條件和限制

- 至少一個有效的 AWS 帳戶。
- 已設定和設定的 AWS Organizations (請參閱教學課程)。
- 從您的管理帳戶查詢 AWS Organizations 的許可。如需詳細資訊，請參閱 AWS Organizations 文件中的 AWS Organizations 和服務連結角色。
- 具有啟動 AWS CloudFormation 範本和相關資源許可的 IAM 主體。如需詳細資訊，請參閱 AWS CloudFormation 文件中的授予自我管理許可。
- 用於部署資源的現有亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。
- Amazon Simple Email Service (Amazon SES) 從沙箱中移出。如需詳細資訊，請參閱 Amazon SES 文件中的移出 Amazon SES 沙箱。
- 如果您選擇在虛擬私有雲 (VPC) 中執行 Lambda，則應在執行主 CloudFormation 範本之前建立下列資源：
  - VPC。
  - 子網路。
  - 適用於 Amazon SES、AWS Systems Manager、AWS Security Token Service (AWS STS)、Amazon S3 和 AWS Secrets Manager 的端點。您可以執行 GitHub IAM 金鑰輪換存放庫中提供的端點範本，以建立這些端點。)
- 存放在 AWS Systems Manager 參數 (SSM 參數) 中的簡易郵件傳送通訊協定 (SMTP) 使用者和密碼。參數必須與主 CloudFormation 範本參數相符。

架構

技術堆疊

- Amazon CloudWatch
- Amazon EventBridge
- IAM
- AWS Lambda
- AWS Organizations
- Amazon S3

架構

下圖展示了此樣式的元件和工作流程。此解決方案支援兩種儲存認證的案例：在成員帳戶和管理帳戶中。

選項 1：將憑據存儲在成員帳戶中

選項 2：將認證儲存在管理帳戶中

該圖顯示了以下工作流程：

1. EventBridge 事件每 24 小時會啟動一個 `account_inventory` Lambda 函數。
2. 此 Lambda 函數會向 AWS Organizations 查詢所有 AWS 帳戶 ID、帳戶名稱和帳戶電子郵件的清單。
3. `account_inventory`Lambda 函數會為每個 AWS 帳戶 ID 啟動 `access_key_auto_rotation` Lambda 函數，並將中繼資料傳遞給該函數以進行其他處理。
4. L `access_key_auto_rotation` ambda 函數使用假設的 IAM 角色來存取 AWS 帳戶 ID。Lambda 指令碼會針對帳戶中的所有使用者及其 IAM 存取金鑰執行稽核。
5. 如果 IAM 存取金鑰的年齡未超過最佳實務閾值，Lambda 函數不會採取進一步的動作。
6. 如果 IAM 存取金鑰的年齡超過最佳實務閾值，`access_key_auto_rotation`Lambda 函數會決定要執行的循環動作。
7. 需要採取動作時，如果產生新金鑰，`access_key_auto_rotation`Lambda 函數會在 AWS Secrets Manager 中建立和更新密碼。也會建立以資源為基礎的政策，僅允許指定的 IAM 主體存取和擷取密碼。在選項 1 的情況下，認證存儲在相應帳戶的 Secrets Manager 中。在選項 2 的情況下 (如果`StoreSecretsInCentralAccount`旗標設定為 True)，認證會儲存在管理帳戶的秘密管理員中。

8. 啟動 notifier Lambda 函數以通知帳戶擁有者輪換活動。此函數會接收 AWS 帳戶 ID、帳戶名稱、帳戶電子郵件以及執行的輪替動作。
9. notifierLambda 函數會查詢部署 S3 儲存貯體以取得電子郵件範本，並使用相關活動中繼資料動態更新。然後，電子郵件會傳送至帳戶擁有者的電子郵件地址。

備註：

- 此解決方案支援多個可用區域的復原能力。不過，它不支援多個 AWS 區域的彈性。如需多個區域的支援，您可以在第二個區域部署解決方案，並保持金鑰循環 EventBridge 規則停用狀態。然後，您可以在第二個區域中執行解決方案時啟用規則。
- 您可以在稽核模式下執行此解決方案。在稽核模式中，IAM 存取金鑰不會修改，但會傳送電子郵件通知使用者。若要以稽核模式執行解決方案，請在執行金鑰循環範本或 access_key_auto_rotation Lambda 函數的環境變數時，將DryRunFlag旗標設定為 True。

自動化和規模

自動化此解決方案的 CloudFormation 範本會在 GitHub IAM 金鑰輪換儲存庫中提供，並列在「程式碼」區段中。在 AWS Organizations Organization 中，您可CloudFormation StackSets以使用在多個帳戶中部署ASA-iam-key-auto-rotation-iam-assumed-roles.yaml CloudFormation 範本，而不必將解決方案個別部署到每個成員帳戶。

工具

AWS 服務

- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

- Amazon Simple Email Service (Amazon SES) 可協助您使用自己的電子郵件地址和網域來傳送和接收電子郵件。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。
- Amazon VPC 端點提供一個界面，可連接到由 AWS 提供支援的服務 PrivateLink，包括許多 AWS 服務。針對您從 VPC 指定的每個子網路，會在子網路中建立端點網路介面，並從子網路位址範圍指派私人 IP 位址。

Code

您可以在 GitHub IAM 金鑰輪替存放庫中取得所需的 AWS CloudFormation 範本、Python 指令碼和工作流程簿文件。範本的部署方式如下。

Template (範本)	部署於	備註
`ASA-iam-key-auto-rotation-and-notifier-solution.yaml`	部署帳戶	這是解決方案的主要模板。
`ASA-iam-key-auto-rotation-iam-assumed-roles.yaml`	您要輪換憑證的單一或多個成員帳戶	您可以使用 CloudFormation 堆疊集在多個帳戶中部署此範本。
`ASA-iam-key-auto-rotation-list-accounts-role.yaml`	中央/管理帳戶	使用此範本保留 AWS Organizations 中的帳戶清單。
`ASA-iam-key-auto-rotation-vpc-endpoints.yaml`	部署帳戶	只有當您想要在 VPC 中執行 Lambda 函數 (在主範本中將`RunLambdaInVPC` 參數設定為 True) 時，才可以使用此範本自動建立端點。

史诗

設定解決方案

任務	描述	所需技能
選擇您的部署 S3 儲存貯體。	登入您帳戶的 AWS 管理主控台，開啟 Amazon S3 主控台，然後為您的部署選擇 S3 儲存貯體。如果您想要在 AWS Organizations Organization 中為多個帳戶實作解決方案，請登入組織的管理帳戶。	雲端架構師
複製儲存庫。	將 GitHub IAM 金鑰輪換存放庫複製到您的本機桌面。	雲端架構師
將檔案上傳到 S3 儲存貯體。	將複製的檔案上傳到 S3 儲存貯體。使用下列預設資料夾結構來複製並貼上所有複製的檔案和目錄：`asa/asa-iam-rotation` 附註：您可以在 CloudFormation 範本中自訂此資料夾結構。	雲端架構師
修改電子郵件範本。	根據您的需求修改`iam-auto-key-rotation-enforcement.html` 電子郵件範本 (位於`template`資料夾中)。將範本末`[Department Name Here]`尾的部門名稱取代為您的部門名稱。	雲端架構師

部署解決方案

任務	描述	所需技能
啟動按鍵旋轉的 CloudFormation 範本。	1. 在部署帳戶中啟動ASA-iam-key-auto-rotation-and-notifier-solution.yaml 範本。如需詳細資訊，請參閱 CloudFormation 文件中的選取堆疊範本。 2. 指定參數的值，包括： • CloudFormation S3 儲存貯體名稱 (S3BucketName) — 包含您的 Lambda 程式碼之部署 S3 儲存貯體的名稱。 • CloudFormation S3 儲存貯體前綴 (S3BucketPrefix) — S3 儲存貯體的前綴。 • 假設的 IAM 角色名稱 (IAMRoleName) — key-rotation Lambda 函數將假設旋轉金鑰的角色名稱。 • IAM 執行角色名稱 (ExecutionRoleName) — key-rotation Lambda 函數所使用之 IAM 執行角色的名稱。 • 庫存執行角色名稱 (Inventory Execution RoleName) —	雲端架構師

任務	描述	所需技能
	account_inventory Lambda 函數使用的 IAM 執行角色的名稱。 • 空運標誌 (稽核模式) (DryRunFlag) — 設定為 True 可開啟稽核模式 (預設)。設定為 False 可開啟強制模式。 • 要列出組織帳號的帳號 (OrgListAccount) — 將用於列出組織中帳號的中央/管理帳戶的帳號 ID。 • 列出帳號角色名稱 (OrgListRole) — 將用於列出組織中帳號的角色名稱。 • 中央帳戶的機密存放區旗標 (StoreSecrets InCentralAccount) — 設定為 True 可將機密儲存在中央帳戶中。設定為 False 可將密碼儲存在相應帳戶中。 • 要複寫登入資料的區域 (CredentialReplicationRegions) — 您要在其中複寫登入資料的 AWS 區域 (Secrets Manager)，以逗號分隔；例如，us-east-2,us-west-1,us-	

任務	描述	所需技能
	west-2 。跳過要創建堆棧的區域。 • 在 VPC 中執行 Lambda (RunLambdaInVpc) — 設定為 True 可在指定的 VPC 中執行 Lambda 函數。您必須建立 VPC 端點，並將 NAT 閘道連接至包含 Lambda 函數的子網路。如需詳細資訊，請參閱涵蓋此選項的 Re: POST 文章。 • Lambda 函數的 VPC 識別碼 (**VpcId**)、安全群組規則的 VPC CIDR 以及 Lambda 函數的子網路識別碼 (**SubnetId**) — 如果設定為 True，則提供 VPC、CIDR 和子網路的相關資訊。**VpcCidr RunLambdaInVpc** • 管理員電子郵件地址 (AdminEmailAddress) — 要傳送通知的有效電子郵件地址。 • AWS 組織 ID (AWSOrgID) — 您組織的唯一 ID。此 ID 以 10-32 個小寫字母或數字開頭，o-並且後面接著。 • 電子郵件範本檔案名稱 [稽核模式] (EmailTemplateAudit) 和 [強制模式] (EmailTemp	

任務	描述	所需技能
	lateEnforce) — 模notifier組針對稽核模式和強制模式要傳送的電子郵件 HTML 範本的檔案名稱。 • SMTP 使用者 SSM 參數名稱 (SMTPUserParamName) 和 SMTP 密碼 SSM 參數名稱 (SMTPPasswordParamName) — 簡易郵件傳送通訊協定 (SMTP) 的使用者和密碼資訊。	

任務	描述	所需技能
啟動假定角色的 CloudFormation 範本。	1. 在 AWS 主 CloudFormation 控台中，為您要輪換金鑰的每個帳戶啟動`ASA-iam-key-auto-rotation-iam-assumed-roles.yaml` 範本。如果您有多個帳戶，則可以將管理帳戶中的主 CloudFormation 範本部署為堆疊，並將具有堆疊集的`ASA-iam-key-auto-rotation-iam-assumed-roles.yaml` CloudFormation 範本部署到所有必要帳戶。如需詳細資訊，請參閱文件 CloudFormation StackSets中的使用 CloudFormation AWS。 2. 指定下列參數的值： • 假設的 IAM 角色名稱 (`IAMRoleName`) — 將由 Lambda `access_key_auto_rotation` 函數假設的 IAM 角色名稱。您可保留預設值。 • IAM 執行角色名稱 (`ExecutionRoleName`) — 將擔任子帳戶角色以執行 Lambda 函數的 IAM 角色。 • 主要 AWS 帳戶 ID (`PrimaryAccountID`) — 將在其中部署主範本的 AWS 帳戶 ID。	雲端架構師

任務	描述	所需技能
	• IAM 豁免群組 (IAMExemptionGroup) — IAM 群組名稱用於促進您要從自動金鑰輪換排除的 IAM 帳戶。	

任務	描述	所需技能
啟動帳戶庫存的 CloudFormation 範本。	1. 在管理/中央帳戶中啟動`ASA-iam-key-auto-rotation-list-accounts-role.yaml` 模板 2. 指定下列參數的值： • 假設的 IAM 角色名稱 (`IAMRoleName`) — Lambda `access_key_auto_rotation` 函數將承擔的 IAM 角色名稱。 • 帳戶 Lambda (**`AccountExecutionRoleName`**) 的 IAM 執行角色名稱 — Lambda `notifier` 函數將承擔的 IAM 角色名稱。 • 輪換 Lambda (**`RotationExecutionRoleName`**) 的 IAM 執行角色名稱 — Lambda `access_key_auto_rotation` 函數將承擔的 IAM 角色名稱。 • 主要 AWS 帳戶 ID (`PrimaryAccountID`) — 將在其中部署主範本的 AWS 帳戶 ID。	雲端架構師

任務	描述	所需技能
啟動 VPC 端點的 CloudFormation 範本。	此工作是選擇性的。 1. 在部署帳戶中啟動`ASA-iam-key-auto-rotation-vpc-endpoints.yaml` 範本。 2. 指定下列參數的值： • VPC (**`pVpcId`**) 的 VPC 識別碼 (**`pSubnetId`**)、子網路識別碼 () 和 CIDR 範圍 — 提供 VPC、CIDR 和子網路的相關資訊。`pVPCCidr` • 將每個 VPC 端點的參數設定為 True。如果您已經有端點，則可以選擇 False。	雲端架構師

相關資源

- IAM 中的安全最佳做法 (IAM 文件)
- AWS Organizations 和服務連結角色 (AWS Organizations 文件)
- 選取堆疊範本 (CloudFormation 文件)
- 使用 AWS CloudFormation StackSets (CloudFormation 文件)

使用 IAM 存取分析器和 AWS CloudFormation 巨集 CodePipeline，在 AWS 帳戶中自動驗證和部署 IAM 政策和角色

創建者：赫爾頓·恩里克·里貝羅（AWS）和吉列爾梅·西蒙斯（AWS）

程式碼儲存庫：IAM 角色管道	環境：PoC 或試點	技術：安全性、身分識別、合規性； DevOps
AWS 服務：AWS CloudFormation；AWS；AWS CodeBuild；AWS CodeCommit；AWS CodePipeline；AWS Lambda；AWS SAM		

Summary

此模式描述步驟並提供建立部署管道的程式碼，讓您的開發團隊在 Amazon Web Services (AWS) 帳戶中建立 AWS Identity and Access Management (IAM) 政策和角色。此方法可協助您的組織降低營運團隊的開銷，並加速部署程序。它還可協助您的開發人員建立與現有控管和安全控制相容的 IAM 角色和政策。

此模式的方法使用 AWS Identity and Access Management 存取分析器來驗證您要附加到 IAM 角色的 IAM 政策，並使 CloudFormation 用 AWS 部署 IAM 角色。不過，您的開發團隊不會直接編輯 AWS CloudFormation 範本檔案，而是建立 JSON 格式的 IAM 政策和角色。在開始部署之前，AWS CloudFormation 巨集會將這些 JSON 格式的政策檔案轉換為 AWS CloudFormation IAM 資源類型。

部署管道 (`RolesPipeline`) 具有來源、驗證和部署階段。在來源階段，您的開發團隊會將包含 IAM 角色和政策定義的 JSON 檔案推送到 AWS CodeCommit 儲存庫。 CodeBuild 然後，AWS 會執行指令碼來驗證這些檔案，並將它們複製到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。由於您的開發團隊無法直接存取儲存在個別 S3 儲存貯體中的 AWS CloudFormation 範本檔案，因此他們必須遵循 JSON 檔案建立和驗證程序。

最後，在部署階段，AWS CodeDeploy 會使用 AWS CloudFormation 堆疊來更新或刪除帳戶中的 IAM 政策和角色。

重要事項：此模式的工作流程是概念驗證 (POC)，我們建議您只在測試環境中使用它。如果您想要在生產環境中使用此模式的方法，請參閱 IAM 文件中的 IAM 中的安全最佳實務，並對 IAM 角色和 AWS 服務進行必要的變更。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- `RolesPipeline`管道的全新或現有 S3 儲存貯體。請確定您使用的存取認證具有將物件上傳至此值區的權限。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需有關這方面的詳細資訊，請參閱 AWS CLI 文件中的安裝、更新和解除安裝 AWS CLI。
- 已安裝和設定的 AWS Serverless Application Model (AWS SAM) CLI。如需這方面的詳細資訊，請參閱 AWS SAM 文件中的安裝 AWS SAM CLI。
- Python 3，安裝在本地計算機上。有關這方面的更多信息，請參閱 Python 文檔。
- 一個 Git 客戶端，安裝和配置。
- 複製到您的本機電腦的 GitHub `IAM roles pipeline`儲存庫。
- 現有 JSON 格式的 IAM 政策和角色。有關這方面的更多信息，請參閱 Github `IAM roles pipeline` 存儲庫中的ReadMe文件。
- 您的開發人員團隊不得擁有編輯此解決方案 AWS CodePipeline 和 CodeDeploy 資源的許可。CodeBuild

限制

- 此模式的工作流程是概念驗證 (POC)，我們建議您只在測試環境中使用它。如果您想要在生產環境中使用此模式的方法，請參閱 IAM 文件中的 IAM 中的安全最佳實務，並對 IAM 角色和 AWS 服務進行必要的變更。

架構

下圖顯示如何使用 IAM 存取分析器和 AWS CloudFormation 巨集 CodePipeline，自動驗證 IAM 角色和政策並將其部署到帳戶。

該圖顯示以下工作流程：

1. 開發人員撰寫包含 IAM 政策和角色定義的 JSON 檔案。開發人員將代碼推送到 CodeCommit 存儲庫， CodePipeline 然後啟動管道。`RolesPipeline`
2. CodeBuild 使用 IAM 存取分析器驗證 JSON 檔案。如果有任何安全性或錯誤相關的發現項目，則會停止部署程序。
3. 如果沒有安全或錯誤相關的發現項目，JSON 檔案會傳送至 `RolesBucket` S3 儲存貯體。
4. 做為 AWS Lambda 函數實作的 AWS CloudFormation 巨集接著會從`RolesBucket`儲存貯體讀取 JSON 檔案，並將其轉換為 AWS CloudFormation IAM 資源類型。
5. 預先定義的 AWS CloudFormation 堆疊會安裝、更新或刪除帳戶中的 IAM 政策和角色。

自動化和規模

GitHub IAM 角色管道存放庫中提供了自動部署此 CloudFormation 模式的 AWS 範本。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- IAM 存取分析器可協助您識別組織和帳戶中與外部實體共用的資源，例如 S3 儲存貯體或 IAM 角色。這有助於您識別對資源和資料的意外存取。
- AWS Serverless Application Model (AWS SAM) 是一種開放原始碼架構，可協助您在 AWS 雲端建置無伺服器應用程式。

Code

此模式的原始程式碼和範本可在 GitHub IAM 角色管線存放庫中取得。

史诗

克隆存儲庫

任務	描述	所需技能
複製範例存放庫。	將 GitHub IAM 角色管線存放庫複製到本機電腦。	一般 AWS 應用程式開發人員

部署管 RolesPipeline 道

任務	描述	所需技能
部署管道。	1. 導覽至包含複製儲存庫的目錄。 2. 執行 `make deploy bucket=<bucket_name>` 命令。重要事項：您必須以`<bucket_name>` 現有 S3 儲存貯體的儲存貯體名稱取代。 3. 執行`aws codepipeline get-pipeline –name RolesPipeline` 命令以檢查部署是否成功。	一般 AWS 應用程式開發人員
克隆管道的存儲庫。	1. `RolesPipeline` AWS CloudFormation 堆疊會建立儲`roles-pipeline-repo` CodeCommit 存庫。 2. 登入 AWS 管理主控台，開啟 AWS 主 CodeCommit 控台，然後複製 CodeCommit 儲存庫的 URL 以將其複製到本機電腦。如需詳細資訊，請參閱 AWS 文件中的 Connect 到 CodeCommit AWS CodeCommit 儲存庫。	一般 AWS 應用程式開發人員

測試管 RolesPipeline 道

任務	描述	所需技能
使用有效的 IAM 政策和角色測試 RolesPipeline 管道。	1. 為您的 IAM 政策和角色建立 JSON 檔案。您可以使用 GitHub `IAM roles pipeline`儲存庫中`role-example` 目錄中的範例。 2. 使用必要的組態定義 IAM 政策和角色。重要事項：請確定您遵循 GitHub `IAM roles pipeline`儲存庫`ReadMe`檔案中所述的格式。 3. 將修改推送到`roles-pipeline-repo` CodeCommit 存儲庫中。 4. 驗證`RolesPipeline` 管道的實作。 5. 確定帳戶中已正確部署 IAM 政策和角色。 6. 驗證 IAM 政策或角色是否存在關聯的許可界限。如需詳細資訊，請參閱 IAM 文件中 IAM 實體的許可界限。	一般 AWS 應用程式開發人員
使用無效的 IAM 政策和角色測試 RolesPipeline 管道。	1. 修改`roles-pipeline-repo` CodeCommit 儲存庫並包含無效的 IAM 角色或政策。例如，您可以使用不存在的動作或無效的 IAM 政策版本。 2. 驗證管道實作。如果 IAM 存取分析器偵測到無效的 IAM	一般 AWS 應用程式開發人員

任務	描述	所需技能
	政策或角色，則會在驗證階段停止管道。	

清除您的資源

任務	描述	所需技能
準備清理。	清空 S3 存儲桶，然後運行`destroy`命令。	一般 AWS 應用程式開發人員
刪除 RolesStack 堆疊。	1. `RolesPipeline` 管道會建立用於部署 IAM 政策和角色的 `RolesStack` AWS CloudFormation 堆疊。您必須先刪除此堆疊，才能刪除`RolesPipeline` 配管。 2. 登入 AWS 管理主控台，開啟 AWS 主 CloudFormation 控台，然後選擇`RolesStack` 堆疊並選擇刪除。	一般 AWS 應用程式開發人員
刪除 RolesPipeline 堆疊。	若要刪除 `RolesPipeline` AWS CloudFormation 堆疊，請依照 Github `IAM roles pipeline` 儲存庫中ReadMe檔案的指示進行操作。	一般 AWS 應用程式開發人員

相關資源

- IAM 存取分析器-政策驗證 (AWS 新聞部落格)
- 使用 AWS CloudFormation 巨集對範本執行自訂處理 (AWS CloudFormation 文件)
- 使用 Python 建置 Lambda 函數 (AWS Lambda 文件)

雙向整合 AWS Security Hub 與 Jira 軟體

創建者：華金曼努埃爾·里諾多（AWS）

程式碼儲存庫：JIRA 整合的 Security Hub	環境：PoC 或試點	技術：安全性、身分識別、合規性
工作負載：所有其他工作	AWS 服務：AWS Lambda；AWS Security Hub；Amazon CloudWatch	

Summary

此解決方案支援 AWS Security Hub 和 Jira 之間的雙向整合。使用此解決方案，您可以從 Security Hub 發現項目自動並手動建立和更新 JIRA 票證。安全團隊可以使用此整合來通知開發人員團隊需要採取行動的嚴重安全發現項目。

該解決方案允許您：

- 選擇在 Jira 中自動建立或更新票證的安全中心控制項。
- 在安全中心主控台中，使用 Security Hub 自訂動作來手動提升 Jira 中的票證。
- 根據 AWS AWS Organizations 織中定義的 AWS 帳戶標籤在 Jira 中自動指派工單。如果未定義此標籤，則會使用預設工作負責人。
- 自動隱藏在 Jira 中標記為誤判或已接受風險的安全中心發現項目。
- 當 Jira 票證的相關發現項目存檔到安全中心時，自動關閉該票證。
- 當 Security Hub 發現重新發生時，重新開啟 Jira 票證。

吉拉工作流程

該解決方案使用自定義 Jira 工作流程，允許開發人員管理和記錄風險。隨著問題在工作流程中移動，雙向整合可確保 Jira 票證和 Security Hub 發現項目的狀態會在兩個服務的工作流程之間同步處理。此工作流程是 Dinis Cruz 的SecDevOps 風險工作流程的衍生工作流程，根據 CC BY 4.0 授權。我們建議新增 Jira 工作流程條件，以便只有安全團隊的成員才能變更工單狀態。

有關此解決方案自動生成的 Jira 票證的示例，請參閱此模式的其他信息部分。

先決條件和限制

先決條件

- 如果您想要在多帳戶 AWS 環境中部署此解決方案：
  - 您的多帳戶環境處於使用中狀態，並由 AWS Organizations 管理。
  - 您的 AWS 帳戶已啟用 Security Hub。
  - 在 AWS Organizations 中，您已指定安全中心管理員帳戶。
  - 您擁有具有 AWS Organizations 管理帳戶`AWSOrganizationsReadOnlyAccess`許可的跨帳戶 IAM 角色。
  - (選用) 您已使用標記 AWS 帳戶`SecurityContactID`。此標籤用於將 Jira 票證分配給定義的安全聯繫人。
- 如果您想要在單一 AWS 帳戶中部署此解決方案：
  - 您擁有有效的 AWS 帳戶。
  - 您的 AWS 帳戶已啟用 Security Hub。
- Jira 伺服器執行個體

  重要提示：此解決方案支持使用 Jira 雲。但是，Jira Cloud 不支持導入 XML 工作流程，因此您需要在 Jira 中手動重新創建工作流程。
- Jira 中的管理員權限
- 以下吉拉令牌之一：
  - 對於吉拉企業，個人訪問令牌（PAT）。如需詳細資訊，請參閱使用個人存取權杖 (Atlassian 支援)。
  - 對於吉拉雲，一個吉拉 API 令牌。如需詳細資訊，請參閱管理 API 權杖 (特定支援)。

架構

本節說明解決方案在各種情況下的架構，例如開發人員和安全工程師決定接受風險或決定修正問題的時間。

案例 1：開發人員解決問題

1. Security Hub 針對指定的安全控制 (例如 AWS 基礎安全最佳實務標準中的安全控制項) 產生一個發現項目。

2. 與發現項目和動作相關聯的 Amazon CloudWatch 事件會啟`CreateJIRA`動 AWS Lambda 函數。
3. Lambda 函數會使用其組態檔案和發現項目的`GeneratorId`欄位來評估是否應該提升發現項目。
4. Lambda 函數會判斷發現項目應該呈報，並從 AWS 管理`SecurityContactID`帳戶中的 AWS Organizations 取得帳戶標籤。此 ID 與開發人員相關聯，並用作 Jira 票證的受指派人 ID。
5. Lambda 函數會使用存放在 AWS Secrets Manager 中的登入資料，在 Jira 中建立票證。吉拉通知開發人員。
6. 開發人員解決了基礎安全性發現，並在 Jira 中將票證的狀態更改為`TEST FIX`。
7. Security Hub 將發現項目更新為`ARCHIVED`，並產生新的事件。此事件會導致 Lambda 函數自動關閉 Jira 票證。

情景 2：開發人員決定接受風險

1. Security Hub 針對指定的安全控制 (例如 AWS 基礎安全最佳實務標準中的安全控制項) 產生一個發現項目。
2. 與發現項目和動作相關聯的 CloudWatch 事件會啟`CreateJIRA`動 Lambda 函數。
3. Lambda 函數會使用其組態檔案和發現項目的`GeneratorId`欄位來評估是否應該提升發現項目。
4. Lambda 函數會判斷發現項目應該呈報，並從 AWS 管理`SecurityContactID`帳戶中的 AWS Organizations 取得帳戶標籤。此 ID 與開發人員相關聯，並用作 Jira 票證的受指派人 ID。
5. Lambda 函數會使用儲存在 Secrets Manager 中的認證，在 Jira 中建立票證。吉拉通知開發人員。
6. 開發人員決定接受風險，並在 Jira 中將票證的狀態更改為`AWAITING RISK ACCEPTANCE`。
7. 安全工程師審核請求，並找到適當的業務理由。安全工程師將 Jira 票證的狀態變更為`ACCEPTED RISK`。這將關閉吉拉票。
8. CloudWatch 每日事件會啟動重新整理 Lambda 函數，該函數會識別已關閉的 JIRA 票證，並將其相關的 Security Hub 發現項目更新為。`SUPPRESSED`

工具

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。

- Amazon E CloudWatch vents 可協助您監控 AWS 資源的系統事件，方法是使用規則來比對事件，並將事件路由到函數或串流。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。它也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。

代碼存儲庫

此模式的代碼可在 GitHub aws-securityhub-jira-software-整合存儲庫中的取得。它包含此解決方案的範例程式碼和 Jira 工作流程。

史诗

配置吉拉

任務	描述	所需技能
匯入工作流程。	以 Jira 的管理員身分，將`issue-workflow.xml` 檔案匯入到您的 Jira 伺服器執行個體。您可以在中的 aws-securityhub-jira-software-整合儲存庫中找到此檔案。 GitHub如需指示，請參閱使用 XML 建立工作流程 (Jira 文件)。	吉拉管理員
啟動並指派工作流程。	在將工作流程指定給工作流程配置之前，工作流程處於非 然後將工作流程規則指定給專案。	吉拉管理員

任務	描述	所需技能
	1. 對於您的專案，請確定您已識別專案的問題類型配置。您可以建立新的問題類型，也可以選取現有問題類型，例如Bug。 2. 根據啟動工作流程 (Jira 文件) 中的指示，將匯入的工作流程指派給工作流程配置。 3. 根據將工作流程配置與專案產生關聯 (Jira 說明文件) 中的指示，將工作流程配置指派給專案。	

設定解決方案參數

任務	描述	所需技能
設定解決方案參數。	1. 在 conf 資料夾中，開啟`params_prod.shfile`。 2. 提供下列參數的值： • `ORG_ACCOUNT_ID` — AWS Organizations 管理帳戶的帳戶 ID。此解決方案會讀取帳戶標籤，並將工單指派給這些 AWS 帳戶標籤中定義的特定安全聯絡人。 • `ORG_ROLE`— 用於存取 AWS 組織管理帳戶的 IAM 角色名稱。此角色必須具有`Organizat`	AWS 系統管理員

任務	描述	所需技能
	ionsReadO nlyAccess 權限。 • EXTERNAL_ID — 如果您使用外部 ID 來承擔中定義的 IAM 角色，則為選用參數ORG_ROLE。如需詳細資訊，請參閱如何使用外部 ID (IAM 文件)。 • JIRA_DEFAULT_ASSIG NEE — 這是所有安全性問題的預設受指派人的 Jira ID。如果帳戶未正確標記或無法假定角色，則會使用此預設值指派。 • JIRA_INSTANCE — Jira 伺服器的 HTTPS 位址格式如下：team-<team-id>.atlassian.ne t/ • JIRA_PROJECT_KEY — 用來建立票證的 Jira 專案金鑰名稱，例如SEC或TEST。這個專案必須已經存在於 Jira 中。 • ISSUE_TYPE — 在 Jira 中指定給專案的問題類型配置名稱，例如Bug或Security Issue。 • REGIONS— 您要部署此解決方案的 AWS 區域代碼清單，例如eu-west-1 。	

任務	描述	所需技能
	3. 儲存並關閉解決方案參數檔案。	
識別您要自動化的發現項目。	1. 開啟安全中心主控台，網址為 https://console.aws.amazon.com/securityhub/ 2. 在資訊安全中心瀏覽窗格中，選擇發現項目。 3. 選擇尋找結果標題。 4. 選擇發現項目 ID。這會顯示發現項目的完整 JSON。 5. 在 JSON 中，複製欄位中的`GeneratorId` 字串。此值採用 AWS 安全尋找格式 (ASFF)。例如，`aws-foundational-security-best-practices/v/1.0.0/S3.1` 對應至安全控制 S3.1 S3 區塊公用存取設定中的發現項目應啟用。 6. 重複這些步驟，直到您已針對要自動化的任何發現項目複製所有`GeneratorID` 值為止。	

任務	描述	所需技能
將發現項目加入組態檔案。	1. 在 src /代碼中打開文件。config.js onconfig 2. 將您在上一個內文中擷取的GeneratorID 值貼到default參數中，並使用逗號分隔每個 ID。 3. 儲存並關閉 組態檔案。 下列程式碼範例顯示自動化aws-foundational-security-best-practices/v/1.0.0/SNS.1 和發aws-foundational-security-best-practices/v/1.0.0/S3.1 現項目。 `{ "Controls" : { "eu-west-1": [ "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule/1.22" ], "default": [ aws-foundational-security-best-practices/v/1.0.0/SNS.1, aws-foundational-security-best-practices/v/1.0.0/S3.1 ] }`	AWS 系統管理員

任務	描述	所需技能
	`}` 注意：您可以選擇針對每個 AWS 區域自動執行不同的發現。協助防止重複發現項目的良好作法是選取單一「區域」，以自動建立 IAM 相關控制項。	

部署整合

任務	描述	所需技能
部署整合。	在命令行終端機中，輸入以下命令： `./deploy.sh prod`	AWS 系統管理員
將吉拉登入資料上傳至 AWS Secrets Manager。	1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。 2. 在 [秘密] 底下，選擇 [儲存新密碼]。 3. 針對機密類型，選擇其他類型的機密。 4. 如果您使用的是 Jira 企業版，對於鍵/值對，請執行以下操作： • 在第一列中，`auth`在關鍵字方塊中輸入，然後`token_auth` 在值方塊中輸入。	AWS 系統管理員

任務	描述	所需技能
	• 新增第二列，在金鑰方塊token中輸入，然後在值方塊中輸入您的個人存取權杖。 如果您使用的是 Jira Cloud，對於鍵/值對，請執行以下操作： • 在第一列中，auth在關鍵字方塊中輸入，然後basic_auth 在值方塊中輸入。 • 新增第二列，token在金鑰方塊中輸入，然後在值方塊中輸入您的 API 權杖。 • 新增第三列，在索引鍵方塊email中輸入，然後在值方塊中輸入您的電子郵件地址。 5. 選擇下一步。 6. 針對密碼名稱Jira-Token ，輸入，然後在頁面底部選擇 [下一步]。 7. 在 [密碼旋轉] 頁面上，保留 [停用自動旋轉]，然後在頁面底部選擇 [下一步]。 8. 在 [檢閱] 頁面上，檢閱密碼詳細資料，然後選擇 [儲存]。	

任務	描述	所需技能
建立安 Security Hub 自訂動作。	1. 對於每個 AWS 區域，在 AWS Command Line Interface (AWS CLI) (AWS CLI) 中，使用create-action-target命令建立名為的 Security Hub 自訂動作`CreateJiraIssue` 。 `aws securityhub create-action-target --name "CreateJiraIssue" \ --description "Create ticket in JIRA" \ --id "CreateJiraIssue" --region $<aws-region>` 2. 開啟安全中心主控台，網址為 https://console.aws.amazon.com/securityhub/。 3. 在資訊安全中心瀏覽窗格中，選擇發現項目。 4. 在發現項目清單中，選取您要提升的發現項目。 5. 在「動作」功能表中選擇`CreateJiraIssue` 。	AWS 系統管理員

相關資源

- 適用於 Jira 服務管理的 AWS 服務管理連接器
- AWS 基礎安全最佳實務標準

其他資訊

吉拉機票示例

當發生指定的 Security Hub 發現時，此解決方案會自動建立 Jira 票證。門票包括以下信息：

- 標題 — 標題會以下列格式識別安全性問題：

```
AWS Security Issue :: <AWS account ID> :: <Security Hub finding title>
```

- 說明 — 票證的說明區段說明與發現項目相關聯的安全性控制、包含 Security Hub 主控台中發現項目的連結，以及提供如何處理 Jira 工作流程中安全性問題的簡短描述。

以下是自動產生的 Jira 票證的範例。

標題	AWS 安全問題:: 012345678912:: 蘭巴達 1 Lambda 函數政策應禁止公開存取。
Description	問題是什麼？ 我們在您負責的 AWS 帳戶 012345678912 中偵測到一個安全性發現。 此控制項會檢查連接至 Lambda 資源的 AWS Lambda 函數政策是否禁止公開存取。如果 Lambda 函數原則允許公開存取，則控制項會失敗。 <Link to Security Hub finding> 我需要如何處理門票？ • 存取帳戶並驗證組態。通過將其移動到「分配修復」來確認工單的工作。一旦修復，移動到測試修復，以便安全性驗證問題得到解決。 • 如果您認為應該接受風險，請將其移至「等待風險接受」。這將需要安全工程師審查。 • 如果您認為是誤報，請將其轉換為「標記為誤報」。這將由安全工程師審查，並相應地重新打開/關閉。

使用 EC2 Image Builder 和 Terraform 為強化的容器映像建立管道

由邁克·聖克羅斯（AWS）和安德魯·拉內斯（AWS）創建

程式碼儲存庫：地形 EC2 Image Builder 容器強化管道	環境：生產	資料來源：封裝工，廚師或純安智
目標：EC2 Image Builder	R 型：重新建築	工作負載：開源
技術：安全性、身分識別、合規性； DevOps	AWS 服務：Amazon EC2 容器註冊表；Amazon EC2 Image Builder	

Summary

此模式會建立 EC2 Image Builder 管道，以產生強化的 Amazon Linux 2 基本容器映像。Terraform 用作基礎結構即程式碼 (IaC) 工具，用來設定和佈建用於建立強化容器映像的基礎結構。此配方可協助您部署以碼頭為基礎的 Amazon Linux 2 容器映像檔，該映像檔已根據 RHEL 7 STIG 第 3 版發行版本 7 — 中型進行強化。(請參閱 EC2 Image Builder 文件中 Linux STIG 元件一節中的 STIG 建置-指令碼中型版本)。這稱為金色容器影像。

該構建包括兩個 Amazon EventBridge 規則。當 Amazon Inspector 發現項目為「高」或「嚴重」時，有一個規則會啟動容器映像管道，以便取代非安全映像。此規則需要同時啟用 Amazon Inspector 器和 Amazon Elastic Container Registry (Amazon ECR) 增強掃描。另一個規則會在成功將映像推送至 Amazon ECR 儲存庫後，將通知傳送至 Amazon 簡單佇列服務 (Amazon SQS) 佇列，以協助您使用最新的容器映像。

先決條件和限制

先決條件

- 您可以在其中部署基礎設施的 AWS 帳戶。
- 已安裝 AWS Command Line Interface (AWS CLI) (AWS CLI)，用於設定用於本機部署的 AWS 登入資料。
- 按照 Terraform 文檔中的說明下載和設置地形。
- Git（如果您從本地計算機進行配置）。

- AWS 帳戶中可用來建立 AWS 資源的角色。
- 在 .tf vars 檔案中定義的所有變數。 或者，您可以在套用 Terraform 組態時定義所有變數。

限制

- 此解決方案會建立 Amazon Virtual Private Cloud (Amazon VPC) 基礎設施，其中包括 NAT 閘道和網際網路閘道，可從其私有子網路連線進行網際網路連線。您無法使用 VPC 端點，因為 AWS 任務協調器和執行程式 (AWSTOE) 的啟動程序會從網際網路安裝 AWS CLI 第 2 版。

產品版本

- Amazon Linux 2
- AWS CLI 版本 1.1 或更新版本

架構

目標技術堆疊

此模式會建立 43 個資源，包括：

- 兩個 Amazon Simple Storage Service (Amazon S3) 儲存貯體：一個用於管道元件檔案，另一個用於伺服器存取和 Amazon VPC 流程日誌
- Amazon ECR 儲存庫
- 包含公有子網路、私有子網路、路由表、NAT 閘道和網際網路閘道的虛擬私有雲端 (VPC)
- EC2 Image Builder 管道、配方和元件
- 容器映像
- 適用於映像加密的 AWS Key Management Service (AWS KMS) 金鑰
- 一個 SQS 隊列
- 三個角色：一個用於執行 EC2 Image Builder 管道，一個適用於 EC2 Image Builder 的執行個體設定檔，另一個用於 EventBridge 規則
- 兩 EventBridge 條規則

地形模塊結構

有關源代碼，請參閱 GitHub 存儲庫 Terraform EC2 Image Builder 容器強化管道。

```
### components.tf
### config.tf
### dist-config.tf
### files
#   ###assumption-policy.json
### hardening-pipeline.tfvars
### image.tf
### infr-config.tf
### infra-network-config.tf
### kms-key.tf
### main.tf
### outputs.tf
### pipeline.tf
### recipes.tf
### roles.tf
### sec-groups.tf
### trigger-build.tf
### variables.tf
```

模組詳情

- components.tf包含用於上傳/files目錄內容的 Amazon S3 上傳資源。您也可以在這裡模塊化添加自定義組件 YAML 文件。
- /files包含定義中使用之元件的.yml檔案components.tf。
- image.tf包含基本映像作業系統的定義。您可以在此處修改不同基礎影像管線的定義。
- infr-config.tf並dist-config.tf包含啟動和分發映像所需的最低 AWS 基礎設施的資源。
- infra-network-config.tf包含要將容器映像部署到的最低 VPC 基礎結構。
- hardening-pipeline.tfvars包含要在套用時使用的地形變數。
- pipeline.tf在地形中創建和管理 EC2 Image Builder 管道。
- recipes.tf是您可以指定組件的不同混合物以創建容器配方的地方。
- roles.tf包含 Amazon 彈性運算雲端 (Amazon EC2) 執行個體設定檔和管道部署角色的 AWS Identity and Access Management (IAM) 政策定義。
- trigger-build.tf包含規 EventBridge 則和 SQS 佇列資源。

目標架構

圖表說明了下列工作流程：

1. EC2 Image Builder 使用定義的方法建立容器映像，該方案會安裝作業系統更新，並將 RHEL 中型 STIG 套用至 Amazon Linux 2 基本映像。
2. 強化的映像會發佈到私有 Amazon ECR 登錄，而當映像成功發佈時， EventBridge 規則會將訊息傳送至 SQS 佇列。
3. 如果將 Amazon Inspector 設定為增強型掃描，它會掃描 Amazon ECR 登錄。
4. 如果 Amazon Inspector 為映像產生嚴重性或高嚴重性發現項目， EventBridge 則規則會觸發 EC2 Image Builder 管道再次執行，並發佈新強化的映像。

自動化和規模

- 此模式說明如何佈建基礎結構，並在您的電腦上建置管道。但是，它旨在大規模使用。您可以在多帳戶環境中使用它們，例如具有 Terraform 環境 Account Factory 的 AWS Control Tower，而不是在本機部署 Terraform 模組。在這種情況下，您應該使用後端狀態 S3 儲存貯體來管理 Terraform 狀態檔案，而不是在本機管理組態狀態。
- 若要擴展使用，請從 Control Tower 或 landing zone 帳戶模型將解決方案部署到一個中央帳戶 (例如共用服務或一般服務帳戶)，並授予消費者帳戶存取 Amazon ECR 儲存庫和 AWS KMS 金鑰的權限。如需有關設定的詳細資訊，請參閱 Re: POST 文章如何允許次要帳戶在 Amazon ECR 映像儲存庫中推送或提取影像？ 例如，在帳戶自動售貨機或 Terraform 的 Account Factory 中，為每個帳戶基準或帳戶自訂基準新增許可，以提供對該 Amazon ECR 儲存庫和加密金鑰的存取權。
- 部署容器映像管道後，您可以使用 EC2 Image Builder 功能 (例如元件) 來修改它，以協助您將更多元件封裝到 Docker 組建中。
- 用於加密容器映像的 AWS KMS 金鑰應該跨映像要用於的帳戶共用。
- 您可以複製整個 Terraform 模組並修改下列屬性，以新增對其他影像的支援：`recipes.tf`
  - 修改`parent_image = "amazonlinux:latest"`為其他影像類型。
  - 修改`repository_name`以指向現有的 Amazon ECR 儲存庫。這會建立另一個管道，將不同的父映像類型部署到您現有的 Amazon ECR 儲存庫。

工具

工具

- 地形表單（IaC 配置）
- Git（如果在本地佈建）

- AWS CLI 第 1 版或第 2 版 (如果在本機佈建)

代碼

此模式的程式碼位於 GitHub 儲存庫 Terraform EC2 Image Builder 容器強化管道中。若要使用範例程式碼，請遵循下一節中的指示。

史诗

佈建基礎架構

任務	描述	所需技能
設定本機認證。	設定您的 AWS 臨時登入資料。 1. 查看是否已安裝 AWS CLI： `$ aws --version` `aws-cli/1.16.249` `Python/3.6.8...` • AWS CLI 版本應為 1.1 或更新版本。 • 如果找不到命令，請安裝 AWS CLI。 2. 執行`aws configure` 並提供下列值： `$ aws configure` `AWS Access Key ID` `[**********xxxx` `]: <Your AWS access` `key ID>` `AWS Secret Access Key` `[**********xxx` `x]: <Your AWS secret` `access key>` `Default region name:` `[us-east-1]: <Your`	AWS DevOps

任務	描述	所需技能
	desired Region for deployment> Default output format [None]: <Your desired output format>	
複製儲存庫。	1. 複製此模式提供的存放庫。您可以使用 HTTPS 或安全殼層 (SSH)。 HTTPS: `git clone https://github.com/aws-samples/terraform-ec2-image-builder-container-hardening-pipeline` SSH: `git clone git@github.com:aws-samples/terraform-ec2-image-builder-container-hardening-pipeline.git` 2. 導航到包含此解決方案的本地目錄： `cd terraform-ec2-image-builder-container-hardening-pipeline`	AWS DevOps

任務	描述	所需技能
更新變數。	更新hardening-pipeline .tfvars 檔案中的變數，以符合您的環境和您想要的組態。您必須提供自己的account_id 。不過，您也應該修改其餘的變數，以符合您想要的部署。所有變量都是必需的。 account_id = "<DEPLOYMENT-ACCOUNT-ID>" aws_region = "us-east-1" vpc_name = "example-hardening-pipeline-vpc" kms_key_alias = "image-builder-container-key" ec2_iam_role_name = "example-hardening-instance-role" hardening_pipeline_role_name = "example-hardening-pipeline-role" aws_s3_ami_resources_bucket = "example-hardening-ami-resources-bucket-0123" image_name = "example-hardening-al2-container-image" ecr_name = "example-hardening-container-repo" recipe_version = "1.0.0"	AWS DevOps

任務	描述	所需技能
	`ebs_root_vol_size = 10` 以下是每個變量的描述： • `account_id` – 您要部署解決方案的 AWS 帳戶號碼。 • `aws_region` – 您要部署解決方案的 AWS 區域。 • `vpc_name`– VPC 基礎架構的名稱。 • `kms_key_alias` – EC2 Image Builder 基礎設施組態要使用的 AWS KMS 金鑰名稱。 • `ec2_iam_role_name` – 將用作 EC2 執行個體設定檔的角色名稱。 • `hardening_pipeline_role_name` – 將用於部署強化管線的角色名稱。 • `aws_s3_ami_resources_bucket` – 將託管構建管道和容器映像所需的所有文件的 S3 存儲桶的名稱。 • `image_name` – 容器映像名稱。此值必須介於 3 到 50 個字元之間，且只能包含英數字元和連字號。 • `ecr_name`– 用於存放容器映像檔的 Amazon ECR 註冊表名稱。	

任務	描述	所需技能
	• `recipe_version` – 圖像配方的版本。預設值為 1.0.0。 • `ebs_root_vol_size` – 亞馬遜彈性區塊存放區 (Amazon EBS) 根磁碟區的大小 (以 GB 為單位)。預設值為 10 GB。	
初始化地形。	更新變數值之後，您可以初始化 Terraform 組態目錄。初始化組態目錄會下載並安裝在組態中定義的 AWS 供應商。 `terraform init` 您應該會看到一則訊息，指出 Terraform 已成功初始化，並識別已安裝的提供者版本。	AWS DevOps
部署基礎結構並建立容器映像。	使用下列指令來初始化、驗證 Terraform 模組，並使用檔`.tfvars`案中定義的變數將 Terraform 模組套用至環境： `terraform init && terraform validate && terraform apply -var-file *.tfvars -auto-approve`	AWS DevOps

任務	描述	所需技能
自訂容器。	在 EC2 Image Builder 部署管道和初始配方之後，您可以建立新版本的容器方案。 您可以新增 EC2 Image Builder 中可用的 31 個以上元件中的任何一個，以自訂容器組建。如需詳細資訊，請參閱 EC2 Image Builder 文件中建立新版本的容器方法的元件一節。	AWS 管理員

驗證資源

任務	描述	所需技能
驗證 AWS 基礎設施佈建。	成功完成第一個 Terraform `apply` 命令後，如果您在本機佈建，您應該會在本機電腦的終端機中看到以下程式碼片段： `Apply complete! Resources: 43 added, 0 changed, 0 destroyed.`	AWS DevOps
驗證個別 AWS 基礎設施資源。	若要驗證已部署的個別資源，如果您在本機佈建，您可以執行下列命令： `terraform state list` 此命令會傳回 43 個資源的清單。	AWS DevOps

移除資源

任務	描述	所需技能
移除基礎結構和容器映像。	當您完成使用 Terraform 組態時，您可以執行下列命令來移除資源： `terraform init && terraform validate && terraform destroy -var-file *.tfvars -auto-approve`	AWS DevOps

故障診斷

問題	解決方案
驗證提供者認證時出錯	當您從本機電腦執行 Terraform `apply` 或`destroy`命令時，可能會遇到類似下列內容的錯誤： `Error: configuring Terraform AWS Provider: error validating provider credentials: error calling sts:GetCallerIdentity: operation error STS: GetCallerIdentity, https response error StatusCode: 403, RequestID: 123456a9-fbc1-40ed-b8d8-513d0133ba7f, api error InvalidClientTokenId: The security token included in the request is invalid.` 此錯誤是由於本機電腦組態中使用的認證的安全性 Token 到期所造成。 若要解決錯誤，請參閱 AWS CLI 文件中的設定和檢視組態設定。

相關資源

- 地形 EC2 Image Builder 容器強化管道 (儲存庫) GitHub
- EC2 Image Builder 文件
- 適用於地形的 AWS Control Tower Account Factory (AWS 部落格文章)
- 後端狀態 S3 儲存貯體 (地形文件)
- 安裝或更新最新版本的 AWS CLI (AWS CLI 文件)
- 下載地形

使用 Terraform 在 AWS Organizations 中集中 IAM 存取金鑰管理

由阿爾蒂·拉吉普特（AWS），金塔馬尼阿普萊（AWS），T.R.L.L.Phani 庫馬爾大地（AWS），普拉迪普庫馬爾潘迪（AWS），馬尤里·辛德（AWS）和普拉塔普庫馬爾南達（AWS）創建

環境：生產	技術：安全、身分識別、合規性；基礎架構	AWS 服務：Amazon EventBridge；AWS Lambda；AWS Organizations；AWS Secrets Manager；Amazon SES

Summary

對於每個組織來說，強制執行金鑰和密碼的安全性規則是必不可少的工作。一個重要的規則是定期輪換 AWS Identity and Access Management (IAM) 金鑰，以強制執行安全性。每當團隊想要從 AWS 命令列界面 (AWS CLI) 或 AWS 外部的應用程式存取 AWS 時，通常都會在本機建立和設定 AWS 存取金鑰。若要維護整個組織的強大安全性，必須在符合要求後或定期變更或刪除舊的安全性金鑰。在組織中跨多個帳戶管理金鑰輪換的程序既耗時又乏味。此模式可協助您使用 Terraform Account Factory (AFT) 和 AWS 服務自動化輪換程序。

該模式提供了以下優點：

- 從中央位置管理組織中所有帳戶的存取金鑰 ID 和秘密存取金鑰。
- 自動旋轉`AWS_ACCESS_KEY_ID`和`AWS_SECRET_ACCESS_KEY`環境變數。
- 如果使用者憑證遭到入侵，強制執行續約。

該模式使用地形表單來部署 AWS Lambda 函數、Amazon EventBridge 規則和 IAM 角色。EventBridge 規則會定期執行，並呼叫 Lambda 函數，該函數會根據使用者存取金鑰的建立時間列出所有使用者存取金鑰。如果先前的金鑰早於您定義的輪替期間 (例如 45 天)，則其他 Lambda 函數會建立新的存取金鑰 ID 和秘密存取金鑰，並使用 Amazon 簡單通知服務 (Amazon SNS) 和 Amazon 簡單 Simple Email Service (Amazon SES) 通知安全管理員。密碼是在 AWS Secrets Manager 中為該使用者建立的，舊的秘密存取金鑰會儲存在秘密管理員中，而且已設定存取舊金鑰的許可。為了確保不再使用舊的存取金鑰，它會在非作用中期間 (例如 60 天，在我們的範例中輪替金鑰之後的 15 天) 停用。經過非作用中的緩衝期間 (例如，在我們的範例中輪替金鑰 90 天或 45 天後)，舊的存取金鑰就會從 AWS Secrets Manager 中刪除。如需詳細的架構和工作流程，請參閱架構一節。

先決條件和限制

- 使用 AWS Control Tower (3.1 版或更新版本) 建置的組織 landing zone
- 地形表單 (AFT) 的 Account Factory 配置了三個帳戶：
  - 組織管理帳戶從中央位置管理整個組織。
  - AFT 管理帳戶主控 Terraform 管線，並將基礎結構部署到部署帳戶中。
  - 部署帳戶可部署此完整解決方案，並從中央位置管理 IAM 金鑰。
- 地形版本 0.15.0 或更新版本，用於佈建部署帳戶中的基礎結構。
- 在亞馬遜簡易電子郵件服務 (Amazon SES) 中設定的電子郵件地址。
- (建議使用) 若要增強安全性，請在虛擬私有雲 (VPC) 內的私有子網路 (部署帳戶) 內部署此解決方案。您可以在自訂變數時提供 VPC 和子網路的詳細資訊 (請參閱 Epics 一節中的程式碼管線的自訂參數)。

架構

AFT 儲存庫

此模式使用 Terraform Account Factory (AFT) 建立所有必要的 AWS 資源和程式碼管道，在部署帳戶中部署資源。代碼管道在兩個存儲庫中運行：

- 全局定制包含 Terraform 代碼，該代碼將在 AFT 註冊的所有帳戶中運行。
- 帳戶自訂包含將在部署帳戶中執行的 Terraform 程式碼。

資源詳情

AWS CodePipeline 任務會在部署帳戶中建立下列資源：

- AWS EventBridge 規則和設定的規則
- `account-inventory`Lambda 函數
- `IAM-access-key-rotation`Lambda 函數
- `Notification`Lambda 函數
- 包含電子郵件範本的亞馬遜簡易儲存服務 (Amazon S3) 儲存貯體
- 必要的 IAM 政策

架構

此圖展示了以下要點：

1. EventBridge 規則會每 24 小時呼叫 `account-inventory` Lambda 函數一次。
2. `account-inventory`Lambda 函數會向 AWS Organizations 查詢所有 AWS 帳戶 ID、帳戶名稱和帳戶電子郵件的清單。
3. `account-inventory`Lambda 函數會為每個 AWS 帳戶啟動 `IAM-access-key-auto-rotation` Lambda 函數，並將中繼資料傳遞給該帳戶以進行其他處理。
4. L `IAM-access-key-auto-rotation` ambda 函數使用假設的 IAM 角色來存取 AWS 帳戶。Lambda 指令碼會針對帳戶中的所有使用者及其 IAM 存取金鑰執行稽核。
5. 部署 `IAM-access-key-auto-rotation` Lambda 函數時，IAM 金鑰循環臨界值 (輪換期間) 會設定為環境變數。如果修改輪換週期，則會使用更新的環境變數重新部署 `IAM-access-key-auto-rotation` Lambda 函數。您可以設定參數來設定輪替週期、舊金鑰的非作用中期間，以及在此之後將刪除舊金鑰的非使用中緩衝區 (請參閱 Epics 中的程式碼管道的自訂參數)。
6. `IAM-access-key-auto-rotation`Lambda 函數會根據存取金鑰的組態來驗證存取金鑰的有效期限。如果 IAM 存取金鑰的年齡未超過您定義的輪替週期，Lambda 函數不會採取進一步的動作。
7. 如果 IAM 存取金鑰的年齡超過您定義的輪替期間，`IAM-access-key-auto-rotation`Lambda 函數會建立新金鑰並輪換現有金鑰。
8. Lambda 函數會將舊金鑰儲存在 Secrets Manager 中，並將權限限制為存取金鑰偏離安全標準的使用者。Lambda 函數也會建立以資源為基礎的政策，僅允許指定的 IAM 主體存取和擷取機密。
9. L `IAM-access-key-rotation` ambda 函數調用 L `Notification` ambda 函數。
10. `Notification`Lambda 函數會查詢 S3 儲存貯體的電子郵件範本，並動態產生含有相關活動中繼資料的電子郵件訊息。
11. `Notification`Lambda 函數調用 Amazon SES 進行進一步的行動。
12. Amazon SES 會將包含相關資訊的電子郵件傳送至帳戶擁有者的電子郵件地址。

工具

AWS 服務

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。此程序需要 IAM 角色和許可。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- Amazon Simple Email Service (Amazon SES) 可協助您使用自己的電子郵件地址和網域來傳送和接收電子郵件。

其他工具

- Terraform 是一種基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署資源。HashiCorp

代碼存儲庫

此模式的指示和程式碼可在 GitHub IAM 存取金鑰輪替存放庫中找到。您可以在 AWS Control Tower 中央部署帳戶中部署程式碼，以從中央位置管理金鑰輪替。

最佳實務

- 如需 IAM，請參閱 IAM 文件中的安全最佳實務。
- 如需金鑰輪替，請參閱 IAM 文件中有關更新存取金鑰的指導方針。

史诗

設定來源檔案

任務	描述	所需技能
複製儲存庫。	1. 複製 IAM 存取金鑰輪替 GitHub 存放庫： `$ git clone https://github.com/aws-samples/centralized-iam-key-management-aws-organizations-terraform.git` 2. 確認存放庫的本機副本包含三個資料夾：	DevOps 工程師

任務	描述	所需技能
	`$ cd Iam-Access-keys-Rotation` `$ ls` `org-account-customization` `global-account-customization` `account-customization`	

設定帳號

任務	描述	所需技能
設定啟動載入帳戶。	作為 AFT 引導過程的一部分，您應該在本地計算機`aft-bootstrap` 上調用一個文件夾。 1. 將所有 Terraform 檔案從本機 GitHub org-account-customization資料夾手動複製到資`aft-bootstrap` 料夾。 2. 執行 Terraform 命令，在 AWS Control Tower 管理帳戶中設定全域跨帳戶角色： `$ cd aft-bootstrap` `$ terraform init` `$ terraform apply –auto-approve`	DevOps 工程師
設定全域自訂。	作為 AFT 文件夾設置的一部分，您應該在本地計算	DevOps 工程師

任務	描述	所需技能
	機aft-global-customizations 上有一個名為的文件夾。 1. 手動將本地文件夾中的所有 Terraform 文件複製到文件 GitHub global-account-customizationaft-global-customizations/terraform 夾。 2. 將程式碼推送至 AWS CodeCommit： ``` $ git add * $ git commit -m "message" $ git push ```	

任務	描述	所需技能
設定帳戶自訂。	作為 AFT 文件夾設置的一部分，您有一個在本地計算機`aft-account-customizations` 上調用的文件夾。 1. 使用您的付費帳號建立資料夾。 2. 手動將本地 GitHub 帳戶自定義文件夾中的所有 Terraform 文件手動複製到您的文件夾。`aft-account-customizations/<vended account>/terraform` 3. 將程式碼推送至 AWS CodeCommit： `$ git add *` `$ git commit -m "message"` `$ git push`	DevOps 工程師

自訂程式碼管線的參數

任務	描述	所需技能
為所有帳戶自訂非 TerraForm 程式碼管線參數。	在資料`aft-global-customizations/terraform/` 夾`input.auto.tfvars` 中建立名為的檔案，並提供所需的輸入資料。有關默認值，請參閱 GitHub 存儲庫中的文件。	DevOps 工程師

任務	描述	所需技能
自訂部署帳戶的程式碼管線參數。	建立`aft-account-customizations/<AccountName>/terraform/` 資料夾`input.auto.tfvars` 中名為的檔案，然後將程式碼推送至 AWS CodeCommit。將程式碼推送到 AWS CodeCommit 會自動啟動程式碼管道。 根據組織的需求指定參數值，包括以下內容 (請參閱 Github 存放庫中的檔案以取得預設值)： • `s3_bucket_name` — 電子郵件範本的唯一值區名稱。 • `s3_bucket_prefix` — S3 儲存貯體內的資料夾名稱。 • `admin_email_address` — 應收到通知之管理員的電子郵件地址。 • `org_list_account` — 管理帳戶的帳號。 • `rotation_period` — 之後金鑰應該從使用中旋轉至非作用中的天數。 • `inactive_period` — 之後應停用旋轉鍵的天數。此值必須大於的值`rotation_period` 。 • `inactive_buffer` — 輪換和停用金鑰之間的寬限期。	DevOps 工程師

任務	描述	所需技能
	• recovery_grace_period — 停用與刪除金鑰之間的寬限期。 • dry_run_flag — 如果您想要傳送通知給管理員進行測試，而不旋轉金鑰，請設定為 true。 • store_secrets_in_central_account — 如果您想要將密碼儲存在部署帳戶中，請設定為 true。如果變數設定為 false (預設值)，密碼將會儲存在成員帳戶中。 • credential_replication_region — 您想要為電子郵件範本部署 Lambda 函數和 S3 儲存貯體的 AWS 區域。 • run_lambda_in_vpc — 設定為 true 以在 VPC 內執行 Lambda 函數。 • vpc_id— 部署帳戶的虛擬私人雲端識別碼 (如果您想要在 VPC 內執行 Lambda 函數)。 • vpc_cidr— 部署帳戶的 CIDR 範圍。 • subnet_id — 部署帳戶的子網路 ID。 • create_smtp_endpoint — 如果要啟用電子郵件端點，請設定為 true。	

驗證金鑰旋轉

任務	描述	所需技能
驗證解決方案。	1. 從 AWS 管理主控台登入部署帳戶。 2. 開啟 IAM 主控台，並檢查是否按指定輪替使用者登入資料 (存取金鑰 ID 和秘密金鑰)。 3. 輪替 IAM 金鑰之後，請確認下列事項： • 舊值會存放在 AWS Secrets Manager 中。 • 密碼名稱的格式為`Account_<account ID>_User_<username>_AccessKey` 。 • 您在`admin_email_address` 參數中指定的使用者會收到有關金鑰輪換的電子郵件通知。	DevOps 工程師

擴充解決方案

任務	描述	所需技能
自訂電子郵件通知日期。	如果您想在停用存取金鑰之前的特定日期傳送電子郵件通知，可以使用以下變更更新`IAM-access-key-auto-rotation` Lambda 函數： 1. 定義一個名為的變量`notify-period` 。	DevOps 工程師

任務	描述	所需技能
	2. 在停用金鑰之main.py前，請在中新增if條件： `If (keyage>rotation-period-notify-period){ send_to_notifier(context, aws_account_id, account_name, resource_owner, resource_actions[resource_owner], dryrun, config.emailTemplateAudit) }`	

故障診斷

問題	解決方案
列出帳戶AccessDenied 時，account-inventory Lambda 工作會失敗。	如果遇到此問題，您必須驗證權限： 1. 登入新的付款帳戶，開啟 Amazon 主 CloudWatch 控台，然後檢視 CloudWatch 日誌群組/aws/lambda/account-inventory-lambda 。 2. 在最新的 CloudWatch 記錄檔中，識別造成存取遭拒問題的帳號。 3. 登入 AWS Control Tower 管理帳戶，並確認角色allow-list-account 已建立。 4. 如果角色不存在，請使用命令重新執行 Terraform 程式碼。terraform apply 5. 選擇 [信任的帳戶] 索引標籤，並驗證相同的帳戶是否受信任。

相關資源

- 地形建議做法（地形文檔）
- IAM 中的安全最佳做法 (IAM 文件)
- 金鑰輪替的最佳做法 (IAM 文件)

集中式記錄和多帳戶安全防護

由安庫什維爾瑪（AWS）和特蕾西（皮爾斯）希基（AWS）創建

環境：生產	技術：安全性、身分識別、合規性、管理與治理	AWS 服務：AWS CloudFormation；AWS Config；Amazon CloudWatch；AWS CodePipeline；Amazon GuardDuty；AWS Lambda；Amazon Macie；AWS Security Hub；Amazon S3

Summary

此模式涵蓋的方法適合在 AWS AWS Organizations 擁有多個 Amazon Web Services (AWS) 帳戶的客戶，現在在使用 AWS Control Tower、landing zone 或帳戶自動售貨機服務在其帳戶中設定基準防護時遇到挑戰。

此模式示範如何使用簡化的多帳戶架構，以結構良好的方式設定集中式記錄和標準化的安全控制。在 AWS CloudFormation 範本、AWS CodePipeline 和自動化指令碼的協助下，此設定會部署在屬於某個組織的所有帳戶中。

多帳戶架構包括以下帳戶：

- 集中式記錄帳戶 — 儲存來自所有其他帳戶的所有虛擬私有雲端 (VPC) 流程 CloudTrail 日誌、AWS Config 日誌和 Amazon CloudWatch 日誌的所有日誌 (使用訂閱) 的帳戶。
- 家長安全帳戶 — 作為下列跨多個帳戶管理之安全性服務之家長帳戶的帳戶。
  - Amazon GuardDuty
  - AWS Security Hub
  - Amazon Macie
  - Amazon Detective
- 子帳號 — 組織中的其他帳號。這些帳戶會將所有有用的記錄檔儲存在集中式記錄帳戶中。兒童帳戶會以安全性服務的成員身分加入家長安全性帳戶。

啟動 CloudFormation 範本 (附加) 後，它會在集中式記錄帳戶中佈建三個 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體。一個儲存貯體用於存放所有帳戶中的所有 AWS 相關日誌 (例如來自 VPC 流程日誌的日誌和 AWS Config)。 CloudTrail第二個存儲桶用於存儲所有帳戶的 CloudFormation 模板。第三個儲存貯體用於存放 Amazon S3 存取日誌。

單獨的 CloudFormation 範本會建立使用 AWS 的管道 CodeCommit。更新後的代碼被推送到 CodeCommit 存儲庫，它負責啟動資源，並在所有帳戶中設置安全服務。如需有關將上載至 CodeCommit 儲存庫之檔案結構的詳細資訊，請參閱 README.md 檔案 (附件)。

先決條件和限制

先決條件

- AWS Organizations 組織 ID，所有帳戶都加入到同一個組織。
- 用於接收亞馬遜簡單通知服務 (Amazon SNS) 通知的有效電子郵件地址。
- 已確認每個帳戶中 Amazon Simple Storage Service (Amazon S3) 儲存貯體的配額。根據預設，每個帳戶都有 100 個 S3 儲存貯體。如果您需要其他值區，請先要求提高配額，然後再部署此解決方案。

限制

所有帳戶都應該是同一個組織的一部分。如果您不是使用 AWS Organizations，則必須修改某些政策 (例如 S3 儲存貯體政策)，以允許每個帳戶從 AWS Identity and Access Management (IAM) 角色進行存取。

注意：部署解決方案時，您必須確認 Amazon SNS 訂閱。確認訊息會傳送至您在部署程序期間提供的電子郵件地址。這將向此電子郵件地址發出一些電子郵件警示訊息，因為每當在帳戶中建立或修改 IAM 角色政策時，都會啟動這些警示。在部署程序期間，您可以忽略這些警示訊息。

架構

目標技術堆疊

- Amazon CloudWatch 警報和日誌
- AWS CodeCommit 儲存庫
- AWS CodePipeline
- AWS Config
- Amazon Detective

- Amazon GuardDuty
- IAM 角色和許可
- Amazon Macie
- S3 儲存貯體
- AWS Security Hub
- Amazon SNS

目標架構

1. 其他已登記為保安服務的家長保安帳戶的子女帳戶
2. 來自所有子帳戶的安全性發現項目，包括父帳戶

資源

將更新的程式碼推送到每個帳戶和 AWS 區域的 CodeCommit 儲存庫時，會自動佈建下列資源。

CloudFormation 堆棧 1-記錄父堆棧

-巢狀堆疊 1 — 標準 IAM 角色和政策

-巢狀堆疊 2 — 帳戶中的 AWS Config 設定

-嵌套堆棧 3- CloudWatch 警報

- SecurityGroupChangesAlarm

- UnauthorizedAttemptAlarm

- RootActivityAlarm

- NetworkAclChangesAlarm

-IAM UserManagementAlarm

-IAM PolicyChangesAlarm

- CloudTrailChangeAlarm

-IAM CreateAccessKeyAlarm

-用於從 CloudTrail 日誌創建指標並將其用於警報的指標過濾器

-社交媒體主題

CloudFormation 堆疊 2 — 父護欄堆疊

-巢狀堆疊 1 — 用於設定帳戶密碼政策的 AWS Lambda 函數

-巢狀堆疊 2 — 基本 AWS Config 規則

-獨聯體 SecurityGroupsMustRestrictSshTraffic

-以 OpenSecurityGroupRuleCheck 及用於安全群組規則評估的 Lambda 函數

-檢查-for-required-tag

- check-for-unrestricted-ports

CloudFormation 堆疊 3 — CloudWatch 記錄檔匯出

-使用亞馬遜 Kinesis 訂閱將 CloudWatch 日誌從日誌群組匯出到 Amazon S3

工具

- AWS CloudFormation — AWS CloudFormation 使用範本以自動化且安全的方式對所有 AWS 區域和帳戶的應用程式所需的所有資源進行建模和佈建。
- Amazon CloudWatch — Amazon 即時 CloudWatch 監控您的 AWS 資源和您在 AWS 上執行的應用程式。您可以用 CloudWatch 來收集和追蹤指標，這些指標是您可以針對資源和應用程式測量的變數。
- AWS CodeCommit — AWS CodeCommit 是 AWS 託管的版本控制服務。您可以使 CodeCommit 用在雲端中私有儲存和管理資產 (例如文件、原始程式碼和二進位檔案)。
- AWS CodePipeline — AWS CodePipeline 是一種持續交付服務，可用來建立軟體發行所需步驟的模型、視覺化和自動化。
- AWS 組態 — AWS Config 提供 AWS 帳戶中 AWS 資源組態的詳細檢視。這包含資源彼此之間的關係和之前的組態方式，所以您可以看到一段時間中組態和關係的變化。
- Amazon Detective — Amazon Detective 用於分析、調查和快速識別安全發現或可疑活動的根本原因。Detective 會自動從您的 AWS 資源收集日誌資料。然後，它會使用機器學習、統計分析和圖論，協助您視覺化並進行更快、更有效率的安全性調查。

- Amazon GuardDuty — Amazon GuardDuty 是一種持續的安全監控服務，可分析和處理流程日誌、CloudTrail 管理事件日誌、 CloudTrail 資料事件日誌和網域名稱系統 (DNS) 日誌。它使用威脅智慧饋送 (例如惡意 IP 位址和網域清單以及機器學習) 以在您的 AWS 環境中識別意外和可能未經授權且惡意的活動。
- AWS Identity and Access Management — AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- Amazon Macie — Amazon Macie 會自動探索敏感資料，例如個人識別資訊 (PII) 和財務資料，讓您更好地瞭解組織在 Amazon S3 中存放的資料。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- AWS Security Hub — AWS Security Hub 為您提供 AWS 安全狀態的全面檢視，並協助您根據安全標準和最佳實務檢查環境。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。

史诗

步驟 1：在所有帳戶中設定 IAM 角色

任務	描述	所需技能
啟動兒童帳戶範本，以在美國東部 1 區域中建立 IAM 角色 CloudFormation 。	若要建立必要的 IAM 角色和許可，您必須在 us-east-1 區域中逐個在每個帳戶中手動啟動此範本 (集中式記錄帳戶、家長安全帳戶以及組織中的所有其他 AWS 帳戶)。`Childaccount_IAM_role_All_Accounts.yaml` 範本位於套件的`/templates/initial_deployment_templates` 目錄中。在進行佈建的 API 呼叫和設定架構的其餘部分時，會使用 IAM 角色。確保	雲端架構師

任務	描述	所需技能
	作為參數傳遞的 IAM 角色名稱在所有帳戶之間都一致。	
在範本參數中，提供 IAM 角色的名稱。	提供父安全帳戶 CodeBuild中可以在所有其他子帳戶中承擔的 IAM 角色。預設角色名稱為 `security_execute_child_stack_role` 。	雲端架構師
在參數中，提供父安全性帳戶的帳戶 ID。	父安全帳戶是 CodeBuild 執行所在的帳戶。	雲端架構師

步驟 2：在集中式記錄帳戶中設定 S3 儲存貯體

任務	描述	所需技能
在集中式記錄帳戶中，在 us-east-1 中，啟動 S3 儲存集中式-.yaml 範本。LoggingAccount CloudFormation	若要在集中式記錄帳戶中建立 S3 儲存貯體，請啟動`S3Buckets-Centralized-LoggingAccount.yaml` 。範本位於套件的`/templates/initial_deployment_templates` 目錄中。S3 儲存貯體會存放所有日誌、範本和 Amazon S3 存取日誌。記下所有 S3 儲存貯體名稱，您將在下列步驟中用來修改參數檔案。	雲端架構師
在範本參數中，為 AWS 日誌儲存提供 S3 儲存貯體的名稱。	輸入`S3 Bucket Name for Centralized Logging in Logging Account`參數的名稱。此儲存貯體充當集中位置，用於存放所有帳戶	雲端架構師

任務	描述	所需技能
	的 AWS 日 CloudTrail 誌，例如流程日誌和日誌。記下存儲桶名稱和 Amazon 資源名稱（ARN）。	
提供用於存放存取日誌的 S3 儲存貯體名稱。	輸入參數的 S3 儲存貯體名`S3 Bucket Name for Access Logs in Logging Account`稱。這個 S3 儲存貯體會存放 Amazon S3 的存取日誌。	雲端架構師
提供用於存放範本的 S3 儲存貯體名稱。	在參數中輸入 S3 儲存貯體名`S3 Bucket Name for CloudFormation Template storage in Logging Account`稱。	雲端架構師
提供組織 ID。	若要提供組織內 S3 儲存貯體的存取權，請在`Organization Id for Non-AMS accounts`參數中輸入組織的 ID。	雲端架構師

步驟 3：在父安全帳戶中部署 CI/CD 基礎結構

任務	描述	所需技能
啟動 security-guard-rails-codepipeline集中式 SecurityAccount .y CloudFormation ml 範本。	若要部署 CI/CD 管線，請在 us-east-1 中的父安全性帳戶中手動啟動`security-guard-rails-codepipeline-Centralized-Security Account.yml` 範本。範本位於套件的`/template`	雲端架構師

任務	描述	所需技能
	s/initial_deployment_templates 目錄中。此管道將部署所有子帳戶中的所有基礎結構。	
提供 S3 儲存貯體的名稱，該儲存貯體將範本存放在集中式記錄帳戶中。	輸入您在步驟 2 中為S3 Bucket Name for the CloudFormation Template storage in Logging Account參數提供的 S3 儲存貯體名稱。	雲端架構師
提供要在子帳戶中使用的 IAM 角色的名稱。	輸入您在步驟 1 中為Name of the IAM role參數提供的名稱。	雲端架構師
提供用於接收 CodePipeline 失敗通知的作用中電子郵件地址。	輸入您要用來接收 CodePipeline 失敗通知及其他 CloudWatch 警報相關通知的電子郵件地址。	雲端架構師

步驟 4：更新檔案以包含帳戶資訊

任務	描述	所需技能
修改帳戶清單.	在位於封裝最上層的 Accountlist.json 檔案中，新增父安全性帳戶號碼和子帳號。請注意，此ChildAccountList 欄位還包括父安全性帳戶編號。請參閱包中文deployment-instructions.md 件中的例子。	雲端架構師

任務	描述	所需技能
修改 accounts.csv 文件	在accounts.csv 文件中，這是在包中的頂級，添加所有子帳戶以及與帳戶註冊的電子郵件一起。請參閱deployment-instructions.md 文件中的示例。	雲端架構師
修改參數. 配置。	在資/templates 料夾中的parameters.config 檔案中，更新下列六個參數： • pNotifyEmail ：您在設定管道時提供的電子郵件地址 (請參閱步驟 3) • pstackNameLogging ：用於集中記錄的 CloudFormation 堆疊名稱 • pS3LogsBucket ：將儲存來自所有帳戶之日誌的 S3 儲存貯體的名稱 (請參閱步驟 2) • pBucketName ：用於存放日誌的 S3 儲存貯體的 ARN • pTemplateBucketName ：要存放範本的 S3 儲存貯體名稱 (請參閱步驟 2) • pAllowedAccounts ：父帳戶和子帳戶的帳號 ID 對於其他參數，您可以保留預設值。如需範例，請參閱封裝中的deployment-instructions.md 檔案。	雲端架構師

第 5 步：訪問 CodeCommit 存儲庫並推送更新的文件

任務	描述	所需技能
存取您在步驟 3 中建立的存 CodeCommit 放庫。	從 CI/CD 基礎結構 CloudFormation 堆疊 (在步驟 3 中啟動) 的「輸出」區段中，記下 CodeCommit 存放庫 URL 的名稱。建立存放庫的存取權，以便將檔案推送至其中，以便在所有目標帳戶中部署基礎結構。如需詳細資訊，請參閱為 AWS 設定 CodeCommit。	雲端架構師
將文件推送到存 CodeCommit 儲庫。	在您的計算機上安裝 Git。然後運行 Git 命令以克隆空存儲庫，將文件從筆記本電腦複製到存儲庫文件夾，然後將成品推送到存儲庫中。檢查包中`deployment-instructions.md` 文件中的示例 Git 命令。如需基本 Git 命令，請參閱相關資源一節。	雲端架構師

步驟 6：確認 CodePipeline 和 CodeBuild 狀態

任務	描述	所需技能
確認 CodePipeline 和的狀態 CodeBuild。	將成品推送至 CodeCommit 存放庫之後，請確認您在步驟 3 中建立的 CodePipeline 管道已啟動。然後檢查 CodeBuild 日誌以確認狀態或錯誤。	雲端架構師

相關資源

- 部署 AWS CloudFormation 範本
- 為 AWS 設定 CodeCommit
- 將檔案上傳到 S3 儲存貯體
- 基本的 Git 命令

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

檢查 Amazon CloudFront 分佈的存取記錄、HTTPS 和 TLS 版本

環境：生產	技術：內容傳遞；安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS CloudFormation；Amazon CloudWatch；AWS Lambda		

Summary

此模式會檢查 Amazon CloudFront 分發，以確保其使用 HTTPS、使用傳輸層安全性 (TLS) 1.2 版或更新版本，並已啟用存取記錄。 CloudFront 是亞馬遜網路服務 (AWS) 提供的一項服務，可加快將靜態和動態網頁內容 (例如 .html、.css、.js 和影像檔) 分發給使用者的速度。 CloudFront 透過稱為節點位置的全球資料中心網路傳遞您的內容。當使用者要求您提供服務的內容時 CloudFront，會將要求路由至提供最低延遲 (時間延遲) 的節點，以便以最佳效能傳送內容。

此模式提供 AWS Lambda 函數 CreateDistribution，CreateDistributionWithTags該函數會在 Amazon CloudWatch 事件偵測到 CloudFront API 呼叫或時啟動UpdateDistribution。Lambda 函數中的自訂邏輯會評估 AWS 帳戶中建立或更新的所有 CloudFront 分發。如果偵測到下列違規，它會使用 Amazon Simple Notification Service (Amazon SNS) 傳送違規通知：

- 全域檢查：
  - 自訂憑證不使用 TLS 1.2 版
  - 已停用散發的記錄
- 原產地檢查：
  - 原始伺服器未使用 TLS 1.2 版進行設定
  - 在 HTTPS 以外的通訊協定上允許與來源通訊
- 行為檢查：
  - 在 HTTPS 以外的通訊協定上允許預設行為通訊
  - 在 HTTPS 以外的通訊協定上允許自訂行為通訊

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 您想要接收違規通知的電子郵件地址

限制

- 除非對分發進行了更新，否則此安全控制不會檢查現有的 Cloudfront 分發。
- CloudFront 被視為全球服務，並不繫結至特定 AWS 區域。不過，適用於全球服務的 Amazon CloudWatch 日誌和 AWS Cloudtrail API 記錄會在美國東部 (維吉尼亞北部) 區域 (us-east-1) 發生。因此， CloudFront 必須在中部署和維護此安全控制us-east-1。此單一部署會監控 CloudFront. 請勿在任何其他 AWS 區域部署安全控制。(在其他區域中部署將導致無法啟 CloudWatch 動事件和 Lambda 函數，且不會發出任何 SNS 通知。)
- 該解決方案已通過 CloudFront Web 內容分發的廣泛測試。它不包括實時消息傳遞協議（RTMP）流分發。

架構

目標技術堆疊

- Lambda 函數
- SNS 主題
- Amazon EventBridge 法則

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署附加的範本。

工具

AWS 服務

- AWS CloudFormation — CloudFormation 這項服務可協助您使用基礎設施即程式碼來建立 AWS 資源模型和設定 AWS 資源。
- Amazon EventBridge — 從您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務 EventBridge 交付即時資料串流，並將該資料路由到 Lambda 函數等目標。
- AWS Lambda — L ambda 支援執行程式碼，無需佈建或管理伺服器。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon SNS 協調和管理發佈者和用戶端之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

附加的代碼包括：

- 包含 Lambda 程式碼 (index.py) 的 .zip 檔案
- 您執行以部署 Lambda 程式碼的 CloudFormation 範本 (.yml 檔案)

史诗

上傳安全控制

任務	描述	所需技能
為 Lambda 程式碼建立 S3 儲存貯體。	在 Amazon S3 主控台上，使用不包含前導斜線的唯一名稱建立 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。您的 S3 儲存貯體必須位於您打算部署 Lambda 程式碼的區域。	雲端架構師

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」區段中提供的 Lambda 程式碼 (cloudfront_ssl_log_lambda.zip 檔案) 上傳到您在上一個步驟中建立的 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
部署 CloudFormation 範本。	在 AWS CloudFormation 主控台與 S3 儲存貯體相同的 AWS 區域中，部署附件區段中提供的 CloudFormation 範本 (雲端前端 SSL 記錄 .yml)。	雲端架構師
指定 S3 儲存貯體名稱。	對於 S3 儲存貯體參數，請指定您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
指定 Lambda 檔案的 Amazon S3 金鑰名稱。	對於 S3 金鑰參數，請在 S3 儲存貯體中指定 Lambda 程式碼 .zip 檔案的 Amazon S3 位置。請勿包含前導斜線 (例如，您可以輸入 lambda.zip 或 controls/lambda.zip)。	雲端架構師
提供通知電子郵件地址。	對於「通知電子郵件」參數，請提供您希望在其中接收違規通知的電子郵件地址。	雲端架構師
定義記錄層級。	對於 Lambda 記錄層級參數，請定義 Lambda 函數的記錄層級。請選擇下列其中一個值：	雲端架構師

任務	描述	所需技能
	• INFO 以獲取有關應用程序進度的詳細信息消息。 • ERROR 以取得仍可能允許應用程式繼續執行的錯誤事件的相關資訊。 • 警告以取得有關潛在有害情況的資訊。	

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會建立新的 SNS 主題，並將訂閱訊息傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

- AWS CloudFormation 資訊
- 在 AWS CloudFormation 主控台上建立堆疊 (CloudFormation 文件)
- CloudFront 日誌記錄（CloudFront 文檔）
- Amazon S3 信息
- AWS Lambda 資訊

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

檢查 IPv4 和 IPv6 的安全群組輸入規則中是否有單一主機網路項目

由 SaiJeevan 德維迪（AWS），甘內什庫馬爾（AWS）和約翰·雷諾茲（AWS）創建

環境：生產	技術：網路；安全性、身分識別、合規	AWS 服務：Amazon SNS；AWS CloudFormation；Amazon CloudWatch；AWS Lambda；Amazon VPC

Summary

此模式提供安全控制，可在 Amazon Web Services (AWS) 資源不符合您的規格時通知您。它提供 AWS Lambda 函數，可在網際網路通訊協定第 4 版 (IPv4) 和 IPv6 安全群組來源位址欄位中尋找單一主機網路項目。當 Amazon CloudWatch 事件偵測到 Amazon 彈性運算雲端 (亞馬遜 EC2) AuthorizeSecurityGroupIngressAPI 呼叫時，就會啟動 Lambda 函數。Lambda 函數中的自訂邏輯會評估安全群組輸入規則之 CIDR 區塊的子網路遮罩。如果子網路遮罩判定為 /32 (IPv4) 或 /128 (IPv6) 以外的任何值，Lambda 函數會使用 Amazon 簡單通知服務 (亞馬遜 SNS) 傳送違規通知。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 您想要接收違規通知的電子郵件地址

限制

- 此安全監控解決方案是區域性的，必須部署在您要監控的每個 AWS 區域中。

架構

目標技術堆疊

- Lambda 函數
- SNS 主題

- Amazon EventBridge 法則

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS CloudFormation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- AWS CloudFormation 是一項服務，可協助您使用基礎設施即程式碼來建立 AWS 資源的模型和設定。
- Amazon 會從您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務 EventBridge提供即時資料串流，並將資料路由到目標 (例如 Lambda 函數)。
- AWS Lambda 支援執行程式碼，無需佈建或管理伺服器。
- Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS 會協調和管理發佈者和用戶端之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

附加的代碼包括：

- 包含 Lambda 安全性控制程式碼 () `index.py` 的 .zip 檔案
- 您執行用來部署 Lambda 程式碼的 CloudFormation 範本 (`security-control.yml`檔案)

史诗

上傳安全控制

任務	描述	所需技能
為 Lambda 程式碼建立 S3 儲存貯體。	在 Amazon S3 主控台上，使用不包含前導斜線的唯一名稱建立 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。您的 S3 儲存貯體必須位於要部署安全群組輸入檢查的 AWS 區域。	雲端架構師
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」區段中提供的 Lambda 程式碼 (`security-control-lambda.zip` 檔案) 上傳到您在上一個步驟中建立的 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
更改 Python 本。	下載「附件」區段中提供的 CloudFormation 範本 (`security-control.yml`)。開啟檔案並修改 Python 版本，以反映 Lambda 支援的最新版本 (目前是 Python 3.9)。 例如，您可以在程式碼`python`中搜尋並將的值`Runtime`從變更`python3.6` 為`python3.9` 。	雲端架構師

任務	描述	所需技能
	如需有關 Python 執行階段版本支援的最新資訊，請參閱 AWS Lambda 文件。	
部署 AWS CloudFormation 範本。	在 AWS CloudFormation 主控台與 S3 儲存貯體相同的 AWS 區域中，部署 CloudFormation 範本 (`security-control.yml`)。	雲端架構師
指定 S3 儲存貯體名稱。	對於 S3 儲存貯體參數，請指定您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
指定 Lambda 檔案的 Amazon S3 金鑰名稱。	對於 S3 金鑰參數，請在 S3 儲存貯體中指定 Lambda 程式碼 .zip 檔案的 Amazon S3 位置。請勿包含前導斜線 (例如，您可以輸入`lambda.zip` 或`controls/lambda.zip`)。	雲端架構師
提供通知電子郵件地址。	對於「通知電子郵件」參數，請提供您希望在其中接收違規通知的電子郵件地址。	雲端架構師

任務	描述	所需技能
定義記錄層級。	對於 Lambda 記錄層級參數，請定義 Lambda 函數的記錄層級。請選擇下列其中一個值： • INFO 以獲取有關應用程序進度的詳細信息消息。 • ERROR 以取得仍可能允許應用程式繼續執行的錯誤事件的相關資訊。 • 警告以取得有關潛在有害情況的資訊。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會建立新的 SNS 主題，並將訂閱訊息傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

- AWS CloudFormation 資訊
- 在 AWS CloudFormation 主控台建立堆疊 (AWS CloudFormation 文件)
- 虛擬私人雲端的安全群組 (Amazon VPC 文件)
- Amazon S3 信息
- AWS Lambda 資訊

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

選擇適用於企業應用程式的 Amazon Cognito 份驗證流程

由邁克爾·戴內特（AWS）和法比安雅克（AWS）創建

環境：生產	技術：安全性、身分識別、合規	AWS 服務：Amazon Cognito

Summary

Amazon Cognito 為網頁和行動應用程式提供身份驗證、授權和使用者管理功能。它為聯合身份的身份驗證提供了有益的功能。為了啟動並運行它，技術架構師需要決定他們希望如何使用這些功能。

Amazon Cognito 支援身份驗證請求的多個流程。這些流程定義了使用者驗證其身分的方式。有關使用哪種身份驗證流程的決定取決於應用程序的特定需求，並且可能會變得複雜。此模式可協助您決定最適合企業應用程式的驗證流程。它假設您已經具備 Amazon Cognito、OpenID Connect (OIDC) 和聯合的基本知識，並且會引導您瞭解有關不同聯合身份驗證流程的詳細資訊。

此解決方案適用於技術決策者。它可協助您瞭解不同的驗證流程，並將其對應至您的應用程式需求。技術潛在客戶應收集必要的見解，以啟動 Amazon Cognito 整合。由於企業組織主要著重於 SAML 聯盟，因此此模式包括具有 SAML 聯盟的 Amazon Cognito 使用者集區的說明。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有完整存取權限的 AWS Identity and Access Management (IAM) 角色和許可，可完整存取 Amazon Cognito
- (選擇性) 存取您的身分識別提供者 (IdP)，例如 Microsoft Entra ID、作用中目錄同盟服務 (AD FS) 或 Okta
- 為您的應用提供高水準的專業知識
- Amazon Cognito 知，OpenID Connect（OIDC）和聯合會的基本知識

限制

- 此模式著重於 Amazon Cognito 使用者集區和身分識別提供者。如需 Amazon Cognito 身分識別集區的相關資訊，請參閱其他資訊一節。

架構

使用下表可協助您選擇驗證流程。本節提供有關每個流程的更多資訊。

您需要 machine-to-machine 驗證嗎？	您的應用程序是一個基於 Web 的應用程序，前端在服務器上呈現？	您的應用程序是單頁應用程序（SPA）還是基於移動的前端應用程序？	您的應用程序是否需要刷新令牌才能使用「保持登錄狀態」功能？	前端是否提供基於瀏覽器的重定向機制？	推薦的 Amazon Cognito 流
是	否	否	否	否	用戶端認證流程
否	是	否	是	是	授權碼流
否	否	是	是	是	使用證明密鑰進行代碼交換（PKCE）的授權代碼流
否	否	否	否	否	資源擁有者密碼流程 *

* 資源擁有者密碼流程應僅在絕對必要時使用。如需詳細資訊，請參閱此模式中的 < 資源擁有者密碼流程 > 一節。

用戶端認證流程

用戶端登入資料流程是 Amazon Cognito 流程中最短的流程。如果系統或服務在沒有任何用戶交互的情況下相互通信，則應使用它。請求系統使用客戶端 ID 和客戶端密鑰來檢索訪問令牌。由於兩個系統都無需使用者互動即可運作，因此不需要額外的同意步驟

此圖展示了以下要點：

1. 應用程式 1 會將包含用戶端 ID 和用戶端密碼的身份驗證請求傳送到 Amazon Cognito 端點，然後擷取存取權杖。
2. 應用程式 1 會在每次後續呼叫「應用程式 2」時使用此存取權杖。
3. 應用程式 2 會使用 Amazon Cognito 驗證存取權杖。

應該使用此流程：

- 應用程式之間沒有使用者互動的通訊

不應使用此流程：

- 對於可以進行用戶交互的任何通信

授權碼流

授權碼流程適用於傳統的基於 Web 的身份驗證。在此流程中，後端處理所有令牌交換和存儲。基於瀏覽器的客戶端看不到實際的令牌。此解決方案用於以 .NET 核心，雅加達面孔或雅加達服務器頁面（JSP）等框架編寫的應用程序。

授權碼流程是基於重定向的流程。客戶端必須能夠與 Web 瀏覽器或類似客戶端進行交互。用戶端會重新導向至驗證伺服器，並針對此伺服器進行驗證。如果用戶端驗證成功，則會將其重新導向回伺服器。

此圖展示了以下要點：

1. 客戶端向 Web 服務器發送請求。
2. 網路伺服器會使用 HTTP 302 狀態碼將用戶端重新導向至 Amazon Cognito。用戶端會自動遵循此重新導向至設定的 IdP 登入。
3. IdP 會檢查 IdP 端上的現有瀏覽器工作階段。如果不存在，則使用者會透過提供使用者名稱和密碼來收到驗證的提示。IdP 會使用 SAML 權杖回應 Amazon Cognito。
4. Amazon Cognito 使用 JSON 網絡令牌（JWT）（特別是代碼令牌）返回成功。Web 服務器調用 /oauth2/token 來將代碼令牌交換為訪問令牌。網路伺服器會將用戶端 ID 和用戶端密碼傳送至 Amazon Cognito 進行驗證。
5. 訪問令牌用於後續對其他應用程序的每次調用。
6. 其他應用程式會使用 Amazon Cognito 驗證存取權杖。

應該使用此流程：

- 如果用戶能夠與 Web 瀏覽器或客戶端進行交互。應用程式程式碼會在伺服器上執行並呈現，以確保沒有密碼會暴露給瀏覽器。

不應使用此流程：

- 對於單頁應用程式 (SPA) 或行動應用程式，因為它們是在用戶端上呈現的，而且不應使用用戶端密碼。

使用 PKCE 的授權碼流

具有代碼交換證明密鑰（PKCE）的授權碼流應用於單頁應用程序和移動應用程序。它是隱式流程的後續任務，因為它使用 PKCE，因此更安全。PKCE 是為公共客戶端授予的 OAuth 2.0 授權代碼的擴展。PKCE 防止贖回被截獲的授權碼。

此圖展示了以下要點：

1. 應用程式會建立程式碼驗證程式和程式碼挑戰。這些都是明確定義的唯一值，會傳送給 Amazon Cognito 以供 future 參考。
2. 應用程式會呼叫 Amazon Cognito 的 /oauth2/授權端點。它會自動將使用者重新導向至設定的 IdP 登入。
3. IdP 會檢查現有的工作階段。如果不存在，則使用者會透過提供使用者名稱和密碼來收到驗證的提示。IdP 會使用 SAML 權杖回應 Amazon Cognito。
4. Amazon Cognito 使用程式碼權杖傳回成功後，網路伺服器會呼叫 /oauth2/token，將程式碼權杖交換為存取權杖。
5. 訪問令牌用於後續對其他應用程序的每次調用。
6. 其他應用程式會使用 Amazon Cognito 驗證存取權杖。

應該使用此流程：

- 適用於 SPA 或行動應用程式

不應使用此流程：

- 如果應用程式後端處理驗證

資源擁有者密碼流程

資源擁有者密碼流程適用於沒有重新導向功能的應用程式。它是通過在自己的應用程序中創建一個登錄表單構建的。登入是透過 CLI 或 SDK 呼叫在 Amazon Cognito 上進行檢查，而不是依賴重新導向流程。在此驗證流程中無法進行聯合，因為聯合需要瀏覽器型重新導向。

此圖展示了以下要點：

1. 使用者在應用程式提供的登入表單中輸入其認證。
2. AWS Command Line Interface (AWS CLI) (AWS CLI) admin-initiated-auth致電 Amazon Cognito。

   注意：或者，您也可以使用 AWS 開發套件而不是 AWS CLI。
3. Amazon Cognito 返回一個訪問令牌。
4. 訪問令牌用於後續對其他應用程序的每次調用。
5. 其他應用程式會使用 Amazon Cognito 驗證存取權杖。

應該使用此流程：

- 將使用直接驗證邏輯（例如基本存取驗證或摘要存取驗證）的現有用戶端移轉至 OAuth 時，方法是將儲存的認證轉換為存取 Token

不應使用此流程：

- 如果您想要使用聯合身分
- 如果您的應用程式支援重

工具

AWS 服務

- Amazon Cognito 為網頁和行動應用程式提供身份驗證、授權和使用者管理功能。

其他工具

- JSON 網絡令牌（JWT）調試器是一個基於 Web 的 JWT 驗證工具。

史诗

評估您的申請

任務	描述	所需技能
定義驗證需求。	根據您的特定驗證需求評估您的應用程式。	應用程式開發人員、應用
使需求與驗證流程保持一致。	在「架構」區段中，使用決策表和每個流程的說明來選擇 Amazon Cognito 身份驗證流程。	應用程式開發人員、一般 AWS、應用程式

設定亞馬遜認可使用者集區

任務	描述	所需技能
建立使用者集區。	1. 登入 AWS 管理主控台，然後開啟 Amazon Cognito 主控台。 2. 建立新的 Cognito 使用者集區。如需指示，請參閱 Amazon Cognito 使用者集區。 3. 視需要更新使用者集區設定和屬性。例如，設定使用者集區的密碼原則。不要創建應用程序客戶端尚未。	一般 AWS
(選擇性) 設定身分識別提供者。	1. 在 Amazon Cognito 使用者集區中建立 SAML 身分識別提供者。如需指示，請參	一般 AWS，聯合管理員

任務	描述	所需技能
	閱[在使用者集區中新增和管理 SAML 身分識別提供者]。 2. 設定您的第三方 SAML 身分識別提供者，以便與 Amazon Cognito 使用者集區的聯盟合作。如需詳細資訊，請參閱[設定第三方 SAML 身分識別提供者]。如果您使用的是 AD FS，請參閱使用 [Amazon Cognito 使用者集區為您的 Web 應用程式建立 AD FS 聯盟] (AWS 部落格文章)。	
建立應用程式用戶端。	1. 為使用者集區建立應用程式用戶端。如需指示，請參閱[建立應用程式用戶端]。注意下列事項： • 視需要變更設定，例如權杖到期。 • 如果您的驗證流程不需要用戶端密碼，請清除 [產生用戶端密碼] 核取方塊。 2. 選擇 [應用程式用戶端設定]，將其整合變更為使用者集區登入 (使用者名稱和密碼)，或透過 SAML 型 IdP 進行聯合登入。 3. 透過定義 URL 並視需要定義 OAuth 流程或範圍來啟用 IdP。	一般 AWS

將應用程序與 Amazon Cognito 集成

任務	描述	所需技能
交換 Amazon Cognito 集成詳細信息。	根據您的身份驗證流程，與應用程式共用 Amazon Cognito 資訊，例如使用者集區 ID 和應用程式用戶端 ID。	一般 AWS 應用程式開發人員
實作 Amazon Cognito 份驗證。	這取決於您選擇的身份驗證流程，您的編程語言以及您正在使用的框架。如需開始使用的某些連結，請參閱「相關資源」一節。	應用程式開發人員

相關資源

AWS 文件

- 使用者集區驗證流程
- 驗證 JSON 網絡令牌
- 使用 Amazon Cognito 可身分集區從 ASP.NET 核心應用程式存取 AWS 服務
- 框架和軟件開發套件：
  - Amazon Amplify 身份驗證
  - Amazon Cognito 身分供應商範例 (適用於 Java 2.x 的 AWS 開發套件文件)
  - 使用 Amazon Cognito 可 (適用於 .NET 文件的 AWS 開發套件) 驗證使用者

AWS 部落格文章

- 使用 Cookie 授權 @Edge：保護您的 Amazon CloudFront 內容不被未經身份驗證的用戶下載
- 使用 Amazon Cognito 使用者集區為您的 Web 應用程式建立 AD FS 聯盟

實作夥伴

- 適用於身份驗證解決方案

其他資訊

常見問答集

為什麼隱式流被棄用？

自 OAuth 2.1 框架發布以來，出於安全原因，隱式流程被標記為已棄用。作為替代方法，請使用「體系結構」部分中描述的 PKCE 授權碼流。

如果 Amazon Cognito 不提供我需要的某些功能，該怎麼辦？

AWS 合作夥伴針對身份驗證和授權解決方案提供不同的 如需詳細資訊，請參閱 AWS 合作夥伴以取得身份驗證解

Amazon Cognito 身份集區流量呢？

Amazon Cognito 使用者集區和聯合身分適用於身分驗證。Amazon Cognito 身分集區可透過請求臨時 AWS 登入資料來授權 AWS 資源存取。在此模式中未討論身份池的 ID 令牌和訪問令牌交換。如需詳細資訊，請參閱 Amazon Cognito 使用者集區和身分集區和一般 Amazon Cognito 案例之間有何差異。

後續步驟

此模式提供 Amazon Cognito 身份驗證流程的概觀。作為下一步，需要選擇應用程序編程語言的詳細實現。多種語言提供開發套件和架構，您可以與 Amazon Cognito 搭配使用。如需有用的參考資料，請參閱相關資源一節。

使用 AWS CloudFormation 安全防護政策建立 AWS 組態自訂規則

代碼庫：aw s 配置-自定義規則雲格式化後衛	環境：PoC 或試點	技術：安全性、身分識別、合規性、管理與治理
AWS 服務：AWS CloudFormation；AWS Config		

Summary

AWS Config 規則可協助您評估 AWS 資源及其目標組態狀態。AWS Config 規則有兩種類型：受管規則和自訂規則。您可以使用 AWS Lambda 函數或使用 AWS CloudFormation Guard (一種 policy-as-code 語言GitHub) 建立自訂規則。

使用 Guard 建立的規則比受管規則提供更精細的控制，而且通常比完全自訂 Lambda 規則更容易設定。這種方法為工程師和架構師提供了構建規則的能力，而無需了解 Python，NodeJS 或 Java，這些都是通過 Lamdba 部署自定義規則所需的。

此模式提供可行的範本、程式碼範例和部署方法，協助您透過 Guard 採用自訂規則。透過使用此模式，管理員可以使用 AWS Config 建立具有組態項目屬性的自訂合規規則。例如，開發人員可以針對 AWS Config 組態項目使用 Guard 政策，持續監控已部署 AWS 和非 AWS 資源的狀態、偵測規則違規，以及自動啟動修復。

目標

閱讀此模式後，您應該能夠：

- 了解安全警衛政策程式碼如何與 AWS Config 服務互動。
- 部署案例 1，這是 AWS Config 自訂規則，使用 Guard 語法驗證加密磁碟區的合規性。此規則會驗證磁碟機是否正在使用中，並確認磁碟機類型為 gp3。
- 部署案例 2，這是使用安全警衛語法驗證 Amazon GuardDuty 合規的 AWS Config 自訂規則。此規則會驗證 GuardDuty 記錄機是否已啟用 Amazon S3 保護和 Amazon EKS 保護。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS 組態，在您的 AWS 帳戶中設定

限制

- 保護自訂規則只能查詢目標設定項目 JSON 記錄中的索引鍵值配對

架構

您可以將安全防護語法以自訂政策的形式套用至 AWS Config 規則。AWS Config 會擷取每個指定資源的階層式 JSON。AWS 組態項目的 JSON 包含鍵值對。這些屬性用於警衛語法中，做為指派給其對應值的變數。

以下是安全警衛語法的說明。組態項目 JSON 中的變數會使用並加上一個%字元。

```
# declare variable
let <variable name> = <'value'>

# create rule and assign condition and policy
    rule <rule name> when
        <CI json key> == <"CI json value"> {
            <top level CI json key>.<next level CI json key> == %<variable name>
        }
```

案例 1：Amazon EBS 磁碟區

案例 1 部署 AWS Config 自訂規則，該規則使用 Guard 語法驗證加密磁碟區的合規性。此規則會驗證磁碟機是否正在使用中，並確認磁碟機類型為 gp3。

以下是案例 1 的 AWS Config 組態項目範例。 此組態項目中有三個金鑰-值配對，用來做為警衛原則中的變數：volumestatusvolumeencryptionstatus、和volumetype。此外，resourceType金鑰也會用作安全防護原則中的篩選器。

```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-01-15T19:04:45.402Z",
  "configurationItemStatus": "ResourceDiscovered",
  "configurationStateId": "4444444444444",
  "configurationItemMD5Hash": "",
```

```
  "arn": "arn:aws:ec2:us-west-2:111111111111:volume/vol-222222222222",
  "resourceType": "AWS::EC2::Volume",
  "resourceId": "vol-222222222222",
  "awsRegion": "us-west-2",
  "availabilityZone": "us-west-2b",
  "resourceCreationTime": "2023-01-15T19:03:22.247Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-33333333333333333",
      "relationshipName": "Is attached to Instance"
    }
  ],
  "configuration": {
    "attachments": [
      {
        "attachTime": "2023-01-15T19:03:22.000Z",
        "device": "/dev/xvda",
        "instanceId": "i-33333333333333333",
        "state": "attached",
        "volumeId": "vol-222222222222",
        "deleteOnTermination": true,
        "associatedResource": null,
        "instanceOwningService": null
      }
    ],
    "availabilityZone": "us-west-2b",
    "createTime": "2023-01-15T19:03:22.247Z",
    "encrypted": false,
    "kmsKeyId": null,
    "outpostArn": null,
    "size": 8,
    "snapshotId": "snap-55555555555555555",
    "state": "in-use",
    "volumeId": "vol-222222222222",
    "iops": 100,
    "tags": [],
    "volumeType": "gp2",
    "fastRestored": null,
    "multiAttachEnabled": false,
    "throughput": null,
    "sseType": null
```

```
  },
  "supplementaryConfiguration": {}
}
```

以下是使用 Guard 語法定義案例 1 中的變數和規則的範例。於下列範例中：

- 前三行使用`let`指令定義變數。 系統會為它們指定一個名稱與值，該名稱與值衍生自組態項目的屬性。
- 該`compliancecheck`規則塊添加了一個查找匹配`AWS::EC2::Volume`的`resourceType`鍵值對的時候條件依賴關係。如果找到相符項目，則規則會繼續處理其餘的 JSON 屬性，並在下列三個條件下尋找相符項目：`stateencrypted`、和`volumeType`。

```
let volumestatus = 'available'
let volumetype = 'gp3'
let volumeencryptionstatus = true

    rule compliancecheck when
        resourceType == "AWS::EC2::Volume" {
            configuration.state == %volumestatus
            configuration.encrypted == %volumeencryptionstatus
            configuration.volumeType == %volumetype
        }
```

有關實現此自定義規則的完整 CloudFormation 防護自定義策略，請參閱代碼存儲庫中的 awsconfig 保護-cft.yaml 或 awsconfig 保護-tf-ec 2vol.json。 GitHub 如需在安全警衛中部署此自訂原則的 HashiCorp Terraform 程式碼，請參閱程式碼儲存庫中的 awsconfig CloudFormation 保護-tf-example.json。

案例 2： GuardDuty 合規性

案例 2 會部署使用安全警衛語法驗證 Amazon GuardDuty 合規的 AWS Config 自訂規則。此規則會驗證 GuardDuty 記錄機是否已啟用 Amazon S3 保護和 Amazon EKS 保護。它還會驗證發 GuardDuty 現項目是否每 15 分鐘發佈一次。此案例可部署在組織中的所有 AWS 帳戶和 AWS 區域 (在 AWS Organizations 中)。

以下是案例 2 的 AWS Config 組態項目範例。此組態項目中有三個金鑰-值配對，用來做為警衛原則中的變數：`FindingPublishingFrequency`、`S3Logs`、和`Kubernetes`。此外，`resourceType`金鑰也會用作原則中的篩選器。

```
{
```

```
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2023-11-27T13:34:28.888Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "7777777777777",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:guardduty:us-
west-2:111111111111:detector/66666666666666666666666666666666",
  "resourceType": "AWS::GuardDuty::Detector",
  "resourceId": "66666666666666666666666666666666",
  "resourceName": "66666666666666666666666666666666",
  "awsRegion": "us-west-2",
  "availabilityZone": "Regional",
  "resourceCreationTime": "2020-02-17T02:48:04.511Z",
  "tags": {},
  "relatedEvents": [],
  "relationships": [],
  "configuration": {
    "Enable": true,
    "FindingPublishingFrequency": "FIFTEEN_MINUTES",
    "DataSources": {
      "S3Logs": {
        "Enable": true
      },
      "Kubernetes": {
        "AuditLogs": {
          "Enable": true
        }
      }
    },

    "Id": "66666666666666666666666666666666",
    "Tags": []
  },
  "supplementaryConfiguration": {
    "CreatedAt": "2020-02-17T02:48:04.511Z"
  }
}
```

以下是使用 Guard 語法定義案例 2 中的變數和規則的範例。於下列範例中：

- 前三行使用let指令定義變數。 系統會為它們指定一個名稱與值，該名稱與值衍生自組態項目的屬性。

- 該compliancecheck規則塊添加了一個查找匹配AWS::GuardDuty::Detector的resourceType鍵值對的時候條件依賴關係。如果找到相符項目，則規則會繼續處理其餘的 JSON 屬性，並在下列三個條件下尋找相符項目：S3Logs.EnableKubernetes.AuditLogs.Enable、和FindingPublishingFrequency。

```
let s3protection = true
let kubernetesprotection = true
let publishfrequency = 'FIFTEEN_MINUTES'

    rule compliancecheck when
        resourceType == "AWS::GuardDuty::Detector" {
            configuration.DataSources.S3Logs.Enable == %s3protection
            configuration.DataSources.Kubernetes.AuditLogs.Enable ==
 %kubernetesprotection
            configuration.FindingPublishingFrequency == %publishfrequency
        }
```

如需實作此自訂規則的完整 CloudFormation 防護自訂原則，請參閱程式碼儲存庫中的 awsconfig-guard- cft-gd.yaml。 GitHub 如需在安全警衛中部署此自訂原則的 HashiCorp Terraform 程式碼，請參閱程式碼儲存庫中的 awsconfig CloudFormation 保護-tf-gd.json。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Config 提供 AWS 帳戶中資源的詳細檢視，以及資源的設定方式。它可協助您識別資源彼此之間的關聯性，以及它們的組態隨著時間的推移而變更的方式。

其他工具

- HashiCorp Terraform 是一種開放原始碼基礎架構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。

代碼存儲庫

此模式的程式碼可在 GitHub AWS Config 與 CloudFormation 防護儲存庫中取得。此程式碼儲存庫包含此模式中所述兩種案例的範例。

史诗

建立 AWS Config 自訂規則

任務	描述	所需技能
(選擇性) 選取規則的機碼值配對。	如果您要定義自訂安全警衛政策，請完成這些步驟。如果您使用案例 1 或 2 的其中一個範例原則，請略過下列步驟。 1. 登入 AWS 管理主控台，並開啟位於 https://console.aws.amazon.com/config/ 的 AWS Config 主控台。 2. 在左側導覽中，選擇 [資源]。 3. 在資源清單中，選擇您要為其建立 AWS Config 自訂規則的資源類型。 4. 請選擇 View Details (查看詳細資訊)。 5. 選擇檢視組態項目 (JSON)。此區段會展開以 JSON 格式顯示組態項目。 6. 識別您要為其建立 AWS Config 自訂規則的索引鍵值配對。	AWS 管理員、安全工程師
建立自訂規則。	使用您先前識別的鍵值對或使用提供的其中一個範例 Guard 政策，按照建立 AWS Config 自訂政策規則中的指示建立自訂規則。	AWS 管理員、安全工程師

任務	描述	所需技能
驗證自訂規則。	執行下列其中一項動作來驗證自訂守門員規則： • 在 AWS Command Line Interface (AWS CLI) (AWS CLI) 中輸入以下命令。 `cfn-guard validate -r guard-s3.guard -d s3bucket-prod-pass.json` • 遵循使用 AWS 組態規則評估資源中 Detective 測模式中的指示，在 AWS Config 中部署規則。確認 Guard 語法與目標帳號或檔案中的對應資源正確匹配。	AWS 管理員、安全工程師

故障診斷

問題	解決方案
在 AWS Config 之外測試 CloudFormation 安全防護政策	您可以在本機裝置或整合式開發環境 (IDE) (例如 AWS Cloud9 IDE) 中完成單元測試。要執行單元測試，請執行以下操作： 1. 安裝 AWS 安 CloudFormation 全警衛 CLI 及其相依性。 2. 將 JSON 格式的 CI 樣本另存為 .json 文件到您的工作站。 3. 將 GuardDuty 原則儲存為 .guard 檔案至您的工作站。 4. 在安全警衛 CLI 中，輸入下列命令，以使用安全警衛原則驗證範例 JSON 檔案。

問題	解決方案
	`cfn-guard validate \` `-r guard-s3.guard \` `-d s3bucket-prod-pass.json`
偵錯 AWS Config 自訂規則	在您的安全警衛政策中，將`EnableDebugLogDelivery` 值變更為`true`。預設值為`false`。日誌消息存儲在 Amazon CloudWatch。

相關資源

AWS 文件

- 建立 AWS Config 自訂政策規則 (AWS Config 文件)
- 撰寫 AWS CloudFormation 安全防護規則 (CloudFormation 安全防護文件)

AWS 部落格文章和研討會

- AWS CloudFormation 安全防護 2.0 簡介 (AWS 部落格文章)

其他資源

- AWS CloudFormation 安全防護 (GitHub)
- CloudFormation 保護 CLI 文檔 (GitHub)

從多個 AWS 帳戶建立 Prowler 安全發現結果的合併報告

代碼存儲庫：多帳戶安全評估-通過瀏覽器	環境：生產	技術：安全性、身分識別、合規性
工作負載：開源	AWS 服務：AWS CloudFormation；Amazon EC2；AWS Identity and Access Management	

Summary

Prowler (GitHub) 是一種開放原始碼命令列工具，可協助您評估、稽核和監控 Amazon Web Services (AWS) 帳戶，以確保遵守安全性最佳實務。在此模式中，您可以在組織中的集 AWS 帳戶 中部署 Prowler (由管理) AWS Organizations，然後使用 Prowler 對組織中的所有帳戶執行安全性評估。

雖然有許多方法可以部署和利用 Prowler 進行評估，但此解決方案的設計目的是為了快速部署、對組織中的所有帳戶或定義的目標帳戶進行全面分析，以及可存取的安全性發現結果報告。在此解決方案中，當 Prowler 完成組織中所有帳戶的安全評估時，它會整合結果。它也會過濾掉任何預期的錯誤訊息，例如防止 Prowler 掃描透過佈建的帳戶中 Amazon Simple Storage Service (Amazon S3) 儲存貯體的限制相關的錯誤。 AWS Control Tower篩選後的合併結果會在此模式中隨附的 Microsoft Excel 範本中報告。您可以使用此報告來識別組織中安全性控制項的潛在改進。

此解決方案的設計考量如下：

- 這些 AWS CloudFormation 範本可減少在此模式中部署資 AWS 源所需的工作量。
- 您可以在部署時調整 CloudFormation 範本和 prowler_scan.sh 指令碼中的參數，以自訂環境的範本。
- Prowler 評估和報告速度可透過 parallel 處理 AWS 帳戶、彙總結果、包含建議補救的合併報告，以及自動產生的視覺效果進行最佳化。
- 使用者不需要監視掃描進度。評估完成後，會透過 Amazon 簡單通知服務 (Amazon SNS) 主題通知使用者，以便他們擷取報告。
- 報告範本可協助您只讀取和評估整個組織的相關結果。

先決條件和限制

先決條件

- 一種 AWS 帳戶 用於託管安全服務和工具，以 AWS Organizations. 在這種模式中，這個帳戶被稱為安全帳戶。
- 在安全性帳戶中，您必須擁有具有輸出網際網路存取權的私有子網路。如需指示，請參閱在私有子網路中包含伺服器的 VPC 和 Amazon Virtual Private Cloud (Amazon VPC) 文件中的 NAT。您可以使用在公用子網路中佈建的 NAT 閘道來建立網際網路存取。
- 存取 AWS Organizations 管理帳戶或已委派管理員權限的帳戶 CloudFormation。如需指示，請參閱 CloudFormation 文件中的註冊委派管理員。
- 啟用 AWS Organizations 和之間的信任存取 CloudFormation。如需指示，請參閱 CloudFormation 文件 AWS Organizations中的啟用受信任存取。

限制

- 目標 AWS 帳戶 必須在中以組織的形式進行管理 AWS Organizations。如果您不使用 AWS Organizations，您可以更新適用於您ProwlerExec環境的 IAM-角色 CloudFormation 範本和 prowler_scan.sh 指令碼。而是提供您要執行指令碼的 AWS 帳戶 ID 和區域清單。
- 該 CloudFormation 範本旨在將 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體部署在具有輸出網際網路存取權的私有子網路中。代 AWS Systems Manager 理程式 (SSM Agent) 需要輸出存取才能連接 AWS Systems Manager 服務端點，而且您需要輸出存取權，才能複製程式碼儲存庫並安裝相依性。如果您想要使用公有子網路，則必須修改偵察者資源 .yaml 範本，以將彈性 IP 地址與 EC2 執行個體建立關聯。

產品版本

- 徘徊者 3.0 版或更高版本

架構

該圖顯示了以下過程：

1. 使用工作階段管理員 (一項功能) AWS Systems Manager，使用者會對 EC2 執行個體進行驗證，並執行 prowler_scan.sh 指令碼。這個命令介面指令碼會執行步驟 2—8。

2. EC2 執行個體擔任 ProwlerEC2Role IAM 角色，該角色授予存取 S3 儲存貯體的許可，並在組織中的其他帳戶中擔任 ProwlerExecRole IAM 角色。
3. EC2 執行個體在組織的管理帳戶中擔任 ProwlerExecRole IAM 角色，並產生組織中的帳戶清單。
4. EC2 執行個體在組織的成員帳戶中擔任 ProwlerExecRole IAM 角色 (在架構圖中稱為工作負載帳戶)，並在每個帳戶中執行安全評估。發現項目會以 CSV 和 HTML 檔案形式儲存在 EC2 執行個體上。

   注意：HTML 文件是游蕩者評估的輸出。由於 HTML 的性質，它們不會連接，處理或直接在此模式中使用。但是，這些可能對個別帳戶報表審查很有用。

5. EC2 執行個體會處理所有 CSV 檔案，以移除已知的預期錯誤，並將剩餘的發現項目合併為單一 CSV 檔案。
6. EC2 執行個體會執行 generateVisualizations.py 指令碼。此指令碼會處理彙總發現項目的 CSV 檔案，並產生圖表和圖表的 PNG 檔案，以協助您瞭解和報告結果。它也會建立 HTML 檔案，其中包含掃描和 PNG 檔案的相關資訊。
7. EC2 執行個體會將個別帳戶結果、彙總結果和產生的視覺效果封裝到 zip 檔案中。
8. EC2 執行個體會將壓縮檔案上傳到 S3 儲存貯體。
9. EventBridge 規則會偵測檔案上傳，並使用 Amazon SNS 主題傳送電子郵件給使用者，通知他們評估已完成。
10. 使用者從 S3 儲存貯體下載 zip 檔案。使用者將結果匯入 Excel 範本並檢閱結果。

工具

AWS 服務

- Amazon Elastic Compute Cloud (Amazon EC2) 在 AWS 雲端中提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如， AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。
- AWS Organizations是一項帳戶管理服務，可協助您 AWS 帳戶 將多個組織整合到您建立並集中管理的組織中。

- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Systems Manager協助您管理在 AWS 雲端. 它可簡化應用程式和資源管理、縮短偵測和解決作業問題的時間，並協助您大規模安全地管理 AWS 資源。此模式使用會話管理器，Systems Manager 器的一種功能。

其他工具

- Prowler 是開放原始碼的命令列工具，可協助您評估、稽核和監控您的帳戶是否遵守 AWS 安全性最佳實務和其他安全性架構和標準。

代碼存儲庫

此模式的代碼可通過 Prowler 存儲庫在 GitHub 多帳戶安全評估中獲得。程式碼儲存庫包含下列檔案：

- prowler_scan.sh-此 bash 腳本用於 parallel 啟動多個 AWS 帳戶遊俠安全評估。如 Prowler 資源 .YAML 中所定義 CloudFormationtemplate，此指令碼會自動部署至 EC2 執行個體上的資料夾。`usr/local/prowler`
- Prowler 資源 .yaml — 您可以使用此 CloudFormation 範本在組織中的安全性帳戶中建立堆疊。此範本會部署此帳戶的所有必要資源，以支援解決方案。此堆疊必須在 IAM-ProwlerExec Role.YAML 範本之前部署。我們不建議您在託管關鍵生產工作負載的帳戶中部署這些資源。

  備註：如果刪除並重新部署此堆疊，您必須重建`ProwlerExecRole` 堆疊集，以便重建 IAM 角色之間的跨帳戶相依性。

- IAM-ProwlerExec Role.yaml — 您可以使用此 CloudFormation 範本建立堆疊集，以在組織中的所有帳戶 (包括管理帳戶) 中部署 `ProwlerExecRole` IAM 角色。
- generateVisualizations.py — prowler_scan.sh 指令碼會自動呼叫此 Python 指令碼，以根據彙總的發現項目產生視覺效果，並將它們包含在 S3 儲存貯體中的 .zip 檔案中。此指令碼會建立下列檔案：
  - `FailuresByAccount-<date>.png`— 說明每個帳戶失敗的 Prowler 檢查的條形圖
  - `FailuresByService-<date>.png`— 條形圖說明失敗的徘徊者檢查每個 AWS 服務
  - `ProcessedResultsByFailureSeverityCount-<date>.png`-說明每個嚴重性級別（嚴重，高，中，低和信息）失敗的 Prowler 檢查分佈的條形圖

  - `ResultsByFail-<date>.png`— 通過嚴重程度檢查失敗的圓形圖
  - `ResultsBySeverity-<date>.png`— 按嚴重性分類的所有 Prowler 檢查（通過和失敗）的餅圖
  - `ProwlerReport.html`-包括所有圖像單 HTML 文件
- 流浪者 3 報告模板 .xlsm-您可以使用此 Excel 模板來處理探索者的發現。報表中的樞紐分析表提供搜尋功能、圖表和合併的發現項目。

史诗

準備部署

任務	描述	所需技能
克隆代碼存儲庫。	1. 在指令行介面中，將工作目錄變更為要儲存範例檔案的位置。 2. 輸入以下命令： `git clone https://github.com/aws-samples/multi-account-security-assessment-via-prowler.git`	AWS DevOps
檢閱範本。	1. 在複製的儲存庫中，開啟「探索者資源 .YAML」和「IAM-角色」檔案。ProwlerExec 2. 檢閱這些範本所建立的資源，並根據您的環境需要調整範本。如需詳細資訊，請參閱 CloudFormation 文件中的使用範本。	AWS DevOps

任務	描述	所需技能
	3. 保存並關閉流浪者資源 .YAML 和 IAM-角色。ProwlerExec	

建立 CloudFormation 堆疊

任務	描述	所需技能
在安全性帳戶中佈建資源。	您可以使用 prowler-resource .yaml 範本建立一個 CloudFormation 堆疊，以部署安全性帳戶中所有必要的資源。如需指示，請參閱 CloudFormation 文件中的建立堆疊。部署此範本時，請注意下列事項： 1. 在 [指定範本] 頁面上，選擇 [範本已準備就緒]，然後上傳檢索者資源 .yaml 檔案。 2. 在 [指定堆疊詳細資料] 頁面的 [堆疊名稱] 方塊中，輸入`Prowler-Resources` 。 3. 在「參數」段落中，輸入下列內容： • `VPCId`— 在帳戶中選擇一個 VPC。 • `SubnetId`— 選擇具有互聯網訪問權限的私有子網。 備註：如果您選取公有子網路，則不會為 EC2 執行	AWS DevOps

任務	描述	所需技能
	個體指派公用 IP 位址，因為 CloudFormation 範本預設不會佈建和附加彈性 IP 位址。 • `InstanceType` — 根據 parallel 評估的數量選取執行個體大小： • 對於 10，請選擇`r6i.large` 。 • 對於 12，請選擇`r6i.xlarge` 。 • 對於 14—18，請選擇 。`r6i.2xlarge` • `InstanceImageId` — 保留默認為 Amazon Linux. • `KeyPairName` — 如果您使用 SSH 進行訪問，請指定現有 key pair 的名稱。 • `PermittedSSHInbound` — 如果您使用 SSH 進行存取，請指定允許的 CIDR 區塊。如果您不使用 SSH，請保留的預設值`127.0.0.1` 。 • `BucketName` — 預設值為`prowler-output-<accountID>-<region>` 。您可以根據需要修改此項。如果您指定自訂值，帳戶 ID 和	

任務	描述	所需技能
	區域會自動附加至指定的值。 • `EmailAddress` — 當 Prowler 完成評估並將 .zip 檔案上傳到 S3 儲存貯體時，請指定 Amazon SNS 通知的電子郵件地址。 注意：必須在 Prowler 完成評估之前確認 SNS 訂閱組態，否則將不會傳送通知。 • `IAMProwlerEC2Role` — 保留預設值，除非您的命名慣例需要此 IAM 角色不同的名稱。 • `IAMProwlerExecRole` — 保留預設值，除非在部署 IAM-ProwlerExecRole.yaml 檔案時會使用其他名稱。 • `Parallelism` — 指定要執行的 parallel 評估數。請確定參數中的值支援此`InstanceType` 數目的 parallel 評量。 • `FindingOutput` — 如果您想要排除通過結果，請選取`FailOnly`。這會大幅減少輸出大小，並著重於可能需要解決的檢查。如果您要包含通過結	

任務	描述	所需技能
	果，請選取FailAndPass 。 4. 在 [複查] 頁面上，選取下列資源需要功能：[AWS::IAM::Role]，然後選擇 [建立堆疊]。 5. 成功建立堆疊後，在 CloudFormation 主控台的「輸出」索引標籤上，複製 ProwlerEC2Role Amazon 資源名稱 (ARN)。您稍後在部署 IAM 角色檔案時使用此 ARN ProwlerExec。	

任務	描述	所需技能
在成員帳戶中佈建 IAM 角色。	在 AWS Organizations 管理帳戶或具有委派管理員權限的帳戶中 CloudFormation，使用 IAM-ProwlerExec Role.YAML 範本建立 CloudFormation 堆疊集。堆疊集會在組織中的所有成員帳戶中部署 `ProwlerExecRole` IAM 角色。如需指示，請參閱 CloudFormation 文件中的建立具有服務管理權限的堆疊集。部署此範本時，請注意下列事項： 1. 在 [準備範本] 下方，選擇 [範本已準備就緒]，然後上傳 IAM-ProwlerExec Role.YAML 檔案。 2. 在 [指定 StackSet 詳細資料] 頁面上，命名堆疊集`IAM-ProwlerExecRole`。 3. 在「參數」段落中，輸入下列內容： • `AuthorizedARN` — 輸入 `ProwlerEC2Role` ARN，您在建立`Prowler-Resources` 堆疊時所複製的。 • `ProwlerExecRoleName` — 保持預設值，`ProwlerExecRole` 除非在部署 Prowler 資源 .yaml 檔案時使用了其他名稱。	AWS DevOps

任務	描述	所需技能
	4. 在 Permissions (許可) 下，選擇 Service-managed permissions (服務管理許可)。 5. 在 [設定部署選項] 頁面的 [部署目標] 下，選擇 [部署至組織] 並接受所有預設值。 備註：如果您希望堆疊同時部署到所有成員帳戶，請將最大並行帳戶和失敗容忍度設定為高值，例如 100。 6. 在部署區域下，選擇 Prowler AWS 區域 的 EC2 執行個體的部署位置。由於 IAM 資源是全球性的，而不是區域資源，因此會在所有作用中區域中部署 IAM 角色。 7. 在 [檢閱] 頁面上，選取 [我確認 AWS CloudFormation 可能會使用自訂名稱建立 IAM 資源]，然後選擇 [建立] StackSet。 8. 監視 [堆疊執行個體] 索引標籤 (針對個別帳戶狀態) 和 [作業] 索引標籤 (針對整體狀態)，以判斷部署何時完成。	

任務	描述	所需技能
在管理帳戶中佈建 IAM 角色。	您可以使用 IAM-ProwlerExecRole.YAML 範本建立一個 CloudFormation 堆疊，以在組織的管理帳戶中部署 `ProwlerExecRole` IAM 角色。您先前建立的堆疊集不會在管理帳戶中部署 IAM 角色。如需指示，請參閱 CloudFormation 文件中的建立堆疊。部署此範本時，請注意下列事項： 1. 在 [指定範本] 頁面上，選擇 [範本已就緒]，然後上傳 IAM-ProwlerExecRole.YAML 檔案。 2. 在 [指定堆疊詳細資料] 頁面的 [堆疊名稱] 方塊中，輸入`IAM-ProwlerExecRole` 。 3. 在「參數」段落中，輸入下列內容： • `AuthorizedARN` — 輸入 `ProwlerEC2Role` ARN，您在建立`Prowler-Resources` 堆疊時所複製的。 • `ProwlerExecRoleName` — 保持預設值，`ProwlerExecRole` 除非在部署 Prowler 資源 .yaml 檔案時使用了其他名稱。	AWS DevOps

任務	描述	所需技能
	4. 在 [複查] 頁面上，選取下列資源需要功能：[AWS::IAM::Role]，然後選擇 [建立堆疊]。	

執行「徘徊者」安全評估

任務	描述	所需技能
運行掃描。	1. 登入組織中的安全性帳戶。 2. 使用工作階段管理員，連接到您先前佈建的 Prowler 的 EC2 執行個體。如需指示，請參閱使用工作階段管理員 Connect 到 Linux 執行個體。如果您無法連線，請參閱此模式的疑難排解一節。 3. 瀏覽至`usr/local/prowler`，然後開啟 prowler_scan.sh 檔案。 4. 根據您的環境需要，檢閱並修改此指令碼中的可調整參數和變數。如需有關自訂選項的詳細資訊，請參閱指令碼開頭的註解。 例如，您可以修改指令碼以指定 AWS 帳戶 ID 或要掃描的外部檔案，而不是從管理帳戶取得組織中所有成員帳戶 AWS 區域 的清單，或者您可以參考包含這些參數的外部檔案。	AWS 管理員

任務	描述	所需技能
	5. 儲存並關閉 prowler_scan.sh 檔案。 6. 輸入下列命令：這會執行 prowler_scan.sh 指令碼。 `sudo -i` `screen` `cd /usr/local/prowler` `./prowler_scan.sh` 注意下列事項： • screen 命令允許腳本在連接超時或您失去控制台訪問權限的情況下繼續運行。 • 掃描開始後，您可以通過按 Ctrl+A D 強制卸離屏幕。 畫面會分離，您可以關閉執行個體連線並允許評估繼續。 • 若要繼續分離的工作階段，請連線至執行個體，輸入`sudo -i`然後輸入`screen -r`。 • 若要監控個別帳戶評估的進度，您可以導覽至`usr/local/prowler` 目錄並輸入指令`tail -f output/stdout-<account-id>` 。 7. 等待 Prowler 完成所有帳戶的掃描。該腳本同時評估多個帳戶。在所有帳戶中完	

任務	描述	所需技能
	成評估後，如果您在部署 Prowler-Resources .yaml 檔案時指定了電子郵件地址，則會收到通知。	
擷取「徘徊者」發現項目。	1. 從`prowler-output-<accountID>-<region>` 值區下載`prowler-output-<assessDate>.zip` 檔案。如需指示，請參閱 Amazon S3 文件中的下載物件。 2. 刪除值區中的所有物件，包括您下載的檔案。這是成本最佳化的最佳作法，並確保您可以隨時刪除`Prowler-Resources` CloudFormation 堆疊。如需指示，請參閱 Amazon S3 文件中的刪除物件。	一般 AWS
停止 EC2 執行個體。	為了防止在執行個體閒置時計費，請停止執行 Prowler 的 EC2 執行個體。如需指示，請參閱 Amazon EC2 文件中的停止和啟動執行個體。	AWS DevOps

建立發現項目的報告

任務	描述	所需技能
匯入發現項目。	1. 在 Excel 中，開啟 prowler-report-template.xlsx 檔案，然後選擇 [探索者 CSV] 工作表。	一般 AWS

任務	描述	所需技能
	2. 刪除所有範例資料，包括標題列。如果系統詢問您是否要刪除與要移除之資料相關聯的查詢，請選擇「否」。刪除查詢可能會影響 Excel 範本中樞紐分析表的功能。 3. 擷取您從 S3 儲存貯體下載的 zip 檔案的內容。 4. 在 Excel 中，打開 prowler-fullorgresults-accessdeniedfiltered.txt。我們建議您使用此檔案，因為已移除最常見的無法執行的錯誤，例如與嘗試掃描 AWS Control Tower 資源相關的`Access Denied`錯誤。如果您想要未篩選的發現項目，請改為開啟 prowler-fullorgresults.txt 檔案。 5. 選取欄 A。 6. 如果您使用的是視窗，請輸入 Ctrl+C，或者如果您使用的是 MacOS，請輸入 Cmd+C。 這會將所有資料複製到剪貼簿。 7. 在 Excel 報告範本的 [執行工具 CSV] 工作表上，選取儲存格 A1。 8. 如果您使用的是視窗，請輸入 Ctrl+V，或者如果您使用的是 MacOS，請輸入 Cmd+V。 這會將發現項目貼到報告中。	

任務	描述	所需技能
	9. 確認已選取包含貼上資料的所有儲存格。如果沒有，請選取欄 A。 10.在 [資料] 索引標籤上，選擇 [文字到欄]。 11.在精靈中，執行下列動作： • 對於步驟 1，選擇「分隔」。 • 對於步驟 2，對於「分隔符號」，請選擇「分號」。在 [資料預覽] 窗格中，確認資料已分隔為欄。 • 對於步驟 3，選擇「完成」。 12.確認文字資料在多個欄中分隔。 13.使用新名稱儲存 Excel 報告。 14.搜尋並刪除發現項目中的任何`Access Denied`錯誤。如需有關如何以程式設計方式移除這些錯誤的指示，請參閱其他資訊一節中的以程式設計	

任務	描述	所需技能
完成報告。	1. 選擇「搜尋結果」工作表，然後選取儲存格 A17。此單元格是數據透視表的標題。 2. 在功能區的「PivotTable 工具」下，選擇「分析」，然後在「重新整理」下選擇「全部重新整理」。這會使用新的資料集來更新樞紐分析表。 3. 默認情況下，Excel 不能正確顯示 AWS 帳戶 數字。若要修正數字格式，請執行下列動作： • 在「搜尋結果」工作表上，開啟欄位 A 的內容 (按一下滑鼠右鍵) 功能表，然後選擇「格式儲存格」。 • 選擇「數字」，然後在「小數位數」中輸入0。 • 選擇確定。 注意：如果一個 AWS 帳戶 數字以一個或多個零開頭，Excel 會自動刪除零。如果您在報表中看到小於 12 位數的帳戶號碼，則該號碼開頭的遺失數字將為零。 4. (選擇性) 您可以收合欄位，讓發現項目更易於閱讀。請執行下列操作： • 在「發現項目」工作表上，如果您將游標移至列	一般 AWS

任務	描述	所需技能
	18 與 19 (嚴重表頭與第一個搜尋結果之間的空格) 之間的行，游標圖示會變更為指向下方的小箭頭。 • 按一下以選取所有尋找欄位。 • 開啟內容 (按一下滑鼠右鍵) 功能表，找到 [展開/收合]，然後選擇 [收合]。 5. 如需有關評估的詳細資訊，請檢閱發現項目、嚴重性和通過失敗工作表。 6. 在 zip 檔案的`Results-Visualizaton-<date-of-scan>` 資料夾中，檢閱自動產生的圖形和圖表，您可以使用這些圖形和圖表來透過視覺效果強化報表。	

(選擇性) 更新 Prowler 或程式碼儲存庫中的資源

任務	描述	所需技能
更新徘徊者。	如果要將 Prowler 更新為最新版本，請執行以下操作： 1. 使用工作階段管理員 Connect 到適用於 Prowler 的 EC2 執行個體。如需指示，請參閱使用工作階段管理員 Connect 到 Linux 執行個體。 2. 輸入以下命令。	一般 AWS

任務	描述	所需技能
	`sudo -i` `pip3 install --upgrade prowler`	

任務	描述	所需技能
更新 prowler_scan.sh 指令碼。	如果要將 prowler_scan.sh 腳本更新為回購中的最新版本，請執行以下操作： 1. 使用工作階段管理員 Connect 到適用於 Prowler 的 EC2 執行個體。如需指示，請參閱使用工作階段管理員 Connect 到 Linux 執行個體。 2. 輸入以下命令。 `sudo -i` 3. 瀏覽至「漫遊者」指令碼目錄。 `cd /usr/local/prowler` 4. 輸入以下命令以隱藏本地腳本，以便您可以將自定義更改合併到最新版本中。 `git stash` 5. 輸入下列命令以取得最新版本的指令碼。 `git pull` 6. 輸入下列命令，將自訂指令碼與最新版本的指令碼合併。 `git stash pop`	一般 AWS

任務	描述	所需技能
	注意：您可能會收到與不在 GitHub 存放庫中的任何本機產生檔案相關的警告，例如尋找報告。只要 prowler_scan.sh 顯示本地存儲的更改合併回來，就可以忽略這些內容。	

(選用) 清除

任務	描述	所需技能
刪除所有已部署的資源。	您可以保留帳戶中部署的資源。如果在非使用 EC2 執行個體時將其關閉，並將 S3 儲存貯體保持空白，這樣可降低維護資源以供 future 掃描使用的成本。 如果您要取消佈建所有資源，請執行下列動作： 1. 刪除管理帳戶中佈建的`IAM-ProwlerExecRole` 堆疊。如需指示，請參閱 CloudFormation 文件中的刪除堆疊。 2. 刪除組織的管理帳戶或委派管理員帳戶中佈建的`IAM-ProwlerExecRole` 堆疊集。如需指示，請參閱 CloudFormation 文件中的刪除堆疊集。 3. 刪除 `prowler-output` S3 儲存貯體中的所有物件。	AWS DevOps

任務	描述	所需技能
	如需指示，請參閱 Amazon S3 文件中的刪除物件。 4. 刪除安全性帳戶中佈建的`Prowler-Resources` 堆疊。如需指示，請參閱 CloudFormation 文件中的刪除堆疊。	

故障診斷

問題	解決方案
無法使用工作階段管理員連線至 EC2 執行個體。	SSM 代理程式必須能夠與 Systems Manager 端點通訊。請執行下列操作： 1. 驗證部署 EC2 執行個體的子網路具有網際網路存取權。 2. 重新啟動 EC2 執行個體。
部署堆疊集時， CloudFormation 主控台會提示您這樣做`Enable trusted access with AWS Organizations to use service-managed permissions` 。	這表示 AWS Organizations 和之間尚未啟用受信任的存取 CloudFormation。部署服務管理的堆疊集需要受信任的存取。選擇按鈕以啟用信任存取。如需詳細資訊，請參閱 CloudFormation 文件中的啟用受信任存取。

相關資源

AWS 文件

- 實作安全性控制 AWS (AWS 規範指引)

其他資源

- 徘徊者 ( ) GitHub

其他資訊

編程刪除錯誤

如果結果包含Access Denied錯誤，您應該將其從發現項目中移除。這些錯誤通常是由於防止 Prowler 評估特定資源的外部影響權限所致。例如，檢閱透過佈建的 S3 儲存貯體時，某些檢查會失敗 AWS Control Tower。您可以透過程式設計方式擷取這些結果，並將篩選結果儲存為新檔案。

下列指令會移除包含單一文字字串 (模式) 的列，然後將結果輸出至新檔案。

- 對於 Linux 或 MacOS 系統（格雷普）

```
grep -v -i "Access Denied getting bucket" myoutput.csv > myoutput_modified.csv
```

- 對於窗戶 (PowerShell)

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket' -NotMatch >
 myoutput_modified.csv
```

下列指令會移除符合多個文字字串的列，然後將結果輸出至新檔案。

- 對於 Linux 或 MacOS（使用字符串之間的轉義管道）

```
grep -v -i 'Access Denied getting bucket\|Access Denied Trying to Get' myoutput.csv >
 myoutput_modified.csv
```

- 對於 Windows（在字符串之間使用逗號）

```
Select-String -Path myoutput.csv -Pattern 'Access Denied getting bucket', 'Access
 Denied Trying to Get' -NotMatch > myoutput_modified.csv
```

報告範例

下列影像是合併 Prowler 發現項目報表中「搜尋結果」工作表的範例。

下列影像是合併 Prowler 發現項目報表中「通過失敗」工作表的範例。(依預設，傳遞結果會從輸出中排除。)

下列影像是合併 Prowler 發現項目報告中「嚴重性」工作表的範例。

使用 AWS Config 和 AWS Systems Manager 刪除未使用的亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區

由桑卡爾·桑格博特拉（AWS）創建

環境：PoC 或試點	技術：安全性、身分識別、合規性、管理與治理、成本管理	AWS 服務：AWS Config；AWS Systems Manager

Summary

Amazon Elastic Block Store (Amazon EBS) 磁碟區的生命週期通常獨立於所連接之 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的生命週期。除非您在啟動時選取「終止時刪除」選項，否則終止 EC2 執行個體會分離 EBS 磁碟區，但不會將其刪除。特別是在啟動和終止 EC2 執行個體常見的開發和測試環境中，這可能會導致大量未使用的 EBS 磁碟區。無論是否使用 EBS 磁碟區，都會在您的 Amazon Web Services (AWS) 帳戶中累積費用。刪除這些磁碟區可協助您優化 AWS 帳戶的成本。此外，刪除未使用的 EBS 磁碟區是安全性最佳作法，可防止存取這些磁碟區中任何未使用且可能敏感的資料。

AWS Config 可協助您手動或自動修復不合規的資源。此模式說明如何設定 AWS Config 規則和自動修復動作，以刪除帳戶中未使用的 Amazon EBS 磁碟區。修復動作是預先定義的自動化工作流程簿，這是 AWS Systems Manager 的一項功能。您可以將 runbook 設定為在刪除磁碟區之前建立磁碟區的快照。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- AWS Identity and Access Management (IAM) 許可，可執行自動化`AWSConfigRemediation-DeleteUnusedEBSVolume`執行手冊，這是 AWS Systems Manager 的一項功能。如需詳細資訊，請參閱 AWSConfigRemediation-DeleteUnused EBS 磁碟區中的所需 IAM 許可。
- 一或多個未使用的 Amazon EBS 磁碟區。

限制

- 未使用的 Amazon EBS 磁碟區必須處於狀`available`態。

架構

技術堆疊

- AWS Config
- Amazon EBS
- Systems Manager
- Systems Manager Automation

目標架構

1. AWS 組態規則會評估 EBS 磁碟區。
2. 此規則會傳回相容與不相容資源的清單。處於`available`狀態的 EBS 磁碟區 (即未使用的磁碟區) 會被判定為不相容。
3. AWS Config 會自動啟動自動化手冊。
4. 如果已設定，Systems Manager 會先建立未使用磁碟區的快照，再刪除它們。
5. Systems Manager 會刪除未使用的 EBS 磁碟區。

自動化和規模

您可以將此解決方案套用至組織中的所有帳戶。如需詳細資訊，請參閱 AWS Config 文件中的管理組織中所有帳戶的規則。

工具

- AWS Config 提供 AWS 帳戶中的資源及其設定方式的詳細檢視。它可協助您識別資源彼此之間的關聯性，以及它們的組態隨著時間的推移而變更的方式。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。
- AWS Systems Manager Automation 可簡化許多 AWS 服務的常見維護、部署和修復任務。

史诗

設定 AWS Config 規則

任務	描述	所需技能
建立自動化工作流程簿的角色。	建立名為的角色`AssumeRole` 。Systems Manager 自動化使用此角色來執行 Runbook。如需指示，請參閱 Systems Manager 說明文件中的設定自動化的服務角色 (假定角色) 存取權限。	AWS 系統管理員
開啟 AWS 組態記錄器。	請遵循 AWS Config 文件中使用主控台設定 AWS 組態中的指示，以確保 AWS Config 正在執行，並設定為記錄 Amazon EBS 磁碟區。	AWS 系統管理員
執行規則。	1. 遵循 AWS Config 文件中評估您的資源中的指示來執行`ec2-volume-inuse-check` 規則。等待評估完成。 2. 在 [規則] 頁面上，選取規`ec2-volume-inuse-check` 則，然後針對範圍內的資源選擇 [不符合標準]。 3. 確認評估結果中有一或多個未使用的 Amazon EBS 磁碟區。	AWS 系統管理員

針對未使用的 Amazon EBS 磁碟區設定自動修復

任務	描述	所需技能
新增自動修復動作。	1. 在「規則」頁面上，選取`ec2-volume-inuse-check` 規則。 2. 請遵循 AWS Config 文件中設定自動修復中的指示進行。注意下列事項： 3. 在「補救動作詳細資訊」區段中，選擇`AWSConfigRemediation-DeleteUnusedEBSVolume` 。 • 選取 [資源 ID 參數]，然後在清單中選擇VolumeId。在執行階段，此參數會以不相容 EBS 磁碟區的識別碼取代。 • 在「參數」段落中，提供下列參數的值： • `CreateSnapshot` — (選擇性) 如果設定為`true`，自動化會在刪除 EBS 磁碟區之前建立快照。 • `AutomationAssumeRole` — 輸入您先前建立的`AssumeRole` 服務角色的 Amazon 資源名稱 (ARN)。	AWS 系統管理員
測試 AWS Config 規則的自動修復。	1. 在 AWS Config 主控台的「規則」頁面上，選取	AWS 系統管理員

任務	描述	所需技能
	規ec2-volume-inuse-check 則。 2. 在 [動作] 功能表中，選擇 [重新評估]。 3. 允許規則評估不合規的資源，然後確認已刪除未使用的 Amazon EBS 磁碟區。	

故障診斷

問題	解決方案
AWS Config 無法準確反映資源狀態。	有時候，AWS Config 不會更新資源的狀態。關閉記錄器，然後在 AWS Config 設定頁面重新開啟。記錄器會擷取資源的狀態。對於新建立或刪除的資源，錄製程式可能需要一些時間才能反映目前的狀態。如需 EBS 磁碟區狀態的詳細資訊，請參閱 Amazon EC2 文件中的磁碟區狀態。

相關資源

- AWSConfigRemediation-DeleteUnused 大容量手冊
- ec2-volume-inuse-check 規則
- 使用 AWS 組態規則修復不合規的 AWS 資源

使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制 CloudFormation

由伊克爾·雷納·富恩特（AWS）和伊万·吉拉迪（AWS）創建

程式碼儲存庫:aws-control-tower-controls-cdk	環境：生產	技術：安全性、身分識別、合規性、雲端原生、基礎架構、管理與治理
AWS 服務：AWS CloudFormation、AWS Control Tower、AWS Organizations、AWS CDK		

Summary

此模式說明如何使用 AWS CloudFormation 和 AWS Cloud Development Kit (AWS CDK) 來實作和管理預防性、偵測和主動的 AWS Control Tower 控制作為基礎設施即程式碼 (IaC)。控制項 (也稱為護欄) 是一項高階規則，可為您的整體 AWS Control Tower 環境提供持續的管控。例如，您可以使用控制項要求 AWS 帳戶記錄，然後在發生特定安全性相關事件時設定自動通知。

AWS Control Tower 可協助您實作預防性、偵測和主動式控制，以管理 AWS 資源並監控多個 AWS 帳戶的合規。每個控制項都會強制執行單一規則。在此模式中，您可以使用提供的 IaC 範本來指定要在環境中部署的控制項。

AWS Control Tower 控制適用於整個組織單位 (OU)，而且控制會影響 OU 中的每個 AWS 帳戶。因此，當使用者在您的 landing zone 中的任何帳戶中執行任何動作時，動作會受到控制 OU 的控制項的約束。

實作 AWS Control Tower 控制有助於為您的 AWS landing zone 建立堅實的安全基礎。透過使用此模式將控制項部署為 IaC to CloudFormation 和 AWS CDK，您可以標準化 landing zone 域中的控制項，並更有效率地部署和管理它們。此解決方案會在部署期間使用 cdk_nag 掃描 AWS CDK 應用程式。此工具會檢查應用程式是否遵守 AWS 最佳實務。

若要將 AWS Control Tower 控制項部署為 I HashiCorp aC，您也可以使用地形來取代 AWS CDK。如需詳細資訊，請參閱使用 Terraform 部署和管理 AWS Control Tower 控制。

目標受眾

對於具有 AWS Control Tower、AWS CDK 和 AWS Organizations 相關經驗的使用者 CloudFormation，建議使用此模式。

先決條件和限制

先決條件

- 以組織身分在 AWS 組織和 AWS Control Tower landing zone 中管理的有效 AWS 帳戶。如需指示，請參閱建立帳戶結構 (AWS Well-Architected 的實驗室)。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 針對 AWS CDK 安裝和設定的節點套件管理員 (npm)。
- AWS CDK 的先決條件。
- 在部署帳戶中擔任現有 AWS Identity and Access Management (IAM) 角色的許可。
- 在組織的管理帳戶中擔任 IAM 角色的許可，該角色可用於啟動 AWS CDK。角色必須具有修改和部署 CloudFormation 資源的權限。如需詳細資訊，請參閱 AWS CDK 文件中的啟動安裝。
- 在組織的管理帳戶中建立 IAM 角色和政策的許可。如需詳細資訊，請參閱 IAM 文件中存取 IAM 資源所需的許可。
- 使用識別碼套用以服務控制原則 (SCP) 為基礎的控制項。必須啟動此 SCP，才能部署主動式控制。如需指示，請參閱不允許管理 AWS CloudFormation 登錄中的資源類型、模組和掛接。

限制

- 此模式提供跨 AWS 帳戶部署此解決方案的指示，從部署帳戶到組織的管理帳戶。基於測試目的，您可以直接在管理帳戶中部署此解決方案，但未明確提供此組態的指示。

產品版本

- 版 Python 3.9 或更高版本
- 故宮版本 8.9.0 或更新版本

架構

目標架構

本節提供此解決方案的高階概觀，以及範例程式碼所建立的架構。下圖顯示在 OU 中各個帳戶中部署的控制項。

AWS Control Tower 控制項會根據其行為和指導進行分類。

控制行為有三種主要類型：

1. 預防性控制的設計是為了防止動作發生。這些是透過 AWS Organizations 中的服務控制政策 (SCP) 來實作。預防控制的狀態可能是強制執行或未啟用。所有 AWS 區域都支援預防性控制。
2. Dete@@ ctive 測控制項的設計目的在於偵測特定事件發生時，並將動作記錄在中 CloudTrail。這些都是透過 AWS 組態規則來實作。偵測控制項的狀態可能是「清除」、「違規」或「未啟用」。Detective 控制僅適用於 AWS 控制塔支援的 AWS 區域。
3. 主動控制會掃描 AWS 佈建的資源， CloudFormation 並檢查它們是否符合您公司的政策和目標。不符合標準的資源將不會被佈建。這些都是使用 AWS CloudFormation 掛鉤實現的。主動控制的狀態為「通過」、「失敗」或「略過」。

控制指引是指建議的做法，說明如何將每個控制項套用至 OU。AWS Control Tower 提供三種類別的指導：強制性、強烈建議和選修。控制項的指導獨立於其行為。如需詳細資訊，請參閱控制行為和指引。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。AWS CDK 工具組是與 AWS CDK 應用程式互動的主要工具。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Config 提供 AWS 帳戶中資源的詳細檢視，以及資源的設定方式。它可協助您識別資源彼此之間的關聯性，以及它們的組態隨著時間的推移而變更的方式。
- AWS Control Tower 可協助您按照規範的最佳實務來設定和管理 AWS 多帳戶環境。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。

其他工具

- cdk_nag 是一種開放原始碼工具，結合使用規則套件來檢查 AWS Cloud Development Kit (AWS CDK) 應用程式是否遵守最佳實務。
- npm 是一個在 Node.js 環境中運行的軟件註冊表，用於共享或借用軟件包以及管理私有軟件包的部署。
- Python 是一種通用的計算機編程語言。

代碼存儲庫

此模式的程式碼可在使用 AWS CDK 儲存庫的 GitHub 部署 AWS Control Tower 控制中取得。您可以使用 cdk.json 檔案與 AWS CDK 應用程式互動，並使用封裝 .json 檔案來安裝 NPM 套件。

最佳實務

- 遵循最低權限（IAM 文檔）的原則。此模式中提供的 IAM 政策和信任政策範例包括所需的最低許可，而在管理帳戶中建立的 AWS CDK 堆疊受到這些許可的限制。
- 遵循 AWS Control Tower 管理員的最佳實務 (AWS Control Tower 文件)。
- 使用 AWS CDK (AWS CDK 文件)，遵循開發和部署雲端基礎設施的最佳實務。
- 啟動安裝 AWS CDK 時，請自訂啟動程序範本以定義政策和受信任帳戶，這些帳戶應具備讀取和寫入管理帳戶中任何資源的能力。如需詳細資訊，請參閱自訂啟動載入。
- 使用程式碼分析工具 (例如 cfn_nag) 掃描產生的範本。 CloudFormation cfn-nag 工具會在 CloudFormation 範本中尋找可能表示基礎結構不安全的模式。您還可以使用 cdk-nag 通過使用雲格式化包含模塊來檢查您的 CloudFormation 模板。

史诗

準備啟用控制

任務	描述	所需技能
在管理帳戶中建立 IAM 角色。	1. 使用其他資訊區段中 IAM 政策中定義的許可，在管理帳戶中建立 IAM 政策。如需指示，請參閱 IAM 說明文件中的建立 IAM 政策。記下政策的 Amazon 資源名稱	DevOps 工程師，一般 AWS

任務	描述	所需技能
	（ARN）。以下是 ARN 的範例。 `arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:policy/<POLICY-NAME>` 2. 在管理帳戶中建立 IAM 角色、附加您在上一個步驟中建立的 IAM 權限政策，並在「其他資訊」區段的「信任政策」中附加自訂信任政策。如需指示，請參閱 IAM 說明文件中的使用自訂信任政策建立角色。以下是新角色的 ARN 範例。 `arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:role/<ROLE-NAME>`	

任務	描述	所需技能
引導程序 AWS CDK。	1. 在管理帳戶中，假設具有啟動 AWS CDK 權限的角色。 2. 輸入下列命令，取代下列命令： • <MANAGEMENT-ACCOUNT-ID> 是組織管理帳戶的 ID。 • <AWS-CONTROL-TOWER-REGION> 是部署 Control Tower 的 AWS 區域。如需區域代碼的完整清單，請參閱 AWS 一般參考中的區域端點。 • <DEPLOYMENT-ACCOUNT-ID> 是部署帳戶的識別碼。 • <DEPLOYMENT-ROLE-NAME> 是您使用部署帳戶的 IAM 角色名稱。 • <POLICY-NAME> 是您在管理帳戶中建立的策略名稱。 `$ npx cdk bootstrap aws://<MANAGEMENT-ACCOUNT-ID>/<AWS-CONTROL-TOWER-REGION> \ --trust arn:aws:iam::<DEPLOYMENT-ACCOUNT-ID>:role/<DEPLOYMENT-ROLE-NAME> \ --cloudformation-execution-policies`	DevOps 工程師, 一般 AWS, Python

任務	描述	所需技能
	arn:aws:iam::<MANAGEMENT-ACCOUNT-ID>:policy/<POLICY-NAME>	
複製儲存庫。	在 bash 外殼中，輸入以下命令。這會使用 AWS CDK 儲存庫複製部署 AWS Control Tower 控制台控制。 GitHub `git clone https://github.com/aws-samples/aws-control-tower-controls-cdk.git`	DevOps 工程師，一般 AWS

任務	描述	所需技能
編輯 AWS CDK 組態檔案。	1. 在複製的儲存庫中，開啟 constants.py 檔案。 2. 在ACCOUNT_ID 參數中，輸入管理帳戶的 ID。 3. 在<AWS-CONTROL-TOWER-REGION> 參數中，輸入部署 AWS 控制塔的 AWS 區域。 4. 在ROLE_ARN參數中，輸入您在管理帳戶中建立之角色的 ARN。 5. 在此GUARDRAILS_CONFIGURATION 區段的Enable-Control 參數中，輸入控制項 API 識別碼。以雙引號輸入識別碼，並以逗號分隔多個識別碼。每個控制項對於可使用 AWS Control Tower 的每個區域都有唯一的 API 識別碼。若要尋找控制項識別碼，請執行下列動作： a. 在控制項中繼資料表中，找出您要啟用的控制項。 b. 在「控制 API 識別碼」的「區域」欄中，找出您進行 API 呼叫的區域的 API 識別碼，例如arn:aws:controltower:us-east-1::control/AWS-GR_ENCRYPTED_VOLUMES 。	

任務	描述	所需技能
	c. 從區域標識符中提取控制標識符，例如AWS-GR_ENCRYPTED_VOLUMES 。 6. 在GUARDRAILS_CONFIGURATION 區段的OrganizationalUnitIds 參數中，輸入您要啟用控制項的組織單位 ID，例如ou-1111-11111111 。以雙引號輸入 ID，並以逗號分隔多個 ID。如需如何擷取 OU ID 的相關資訊，請參閱檢視 OU 的詳細資訊。 7. 儲存並關閉 constants.py 檔案。如需更新 constants.py 檔案的範例，請參閱此病毒碼的其他資訊一節。	

在管理帳戶中啟用控制

任務	描述	所需技能
假設部署帳戶中的 IAM 角色。	在部署帳戶中，假設具有在管理帳戶中部署 AWS CDK 堆疊許可的 IAM 角色。如需在 AWS CLI 中假設 IAM 角色的詳細資訊，請參閱在 AWS CLI 中使用 IAM 角色。	DevOps 工程師，一般 AWS
啟動環境。	如果您使用的是 Linux 或 MacOS 果系統：	DevOps 工程師，一般 AWS

任務	描述	所需技能
	1. 輸入以下指令以建立虛擬環境。 `$ python3 -m venv .venv` 2. 建立虛擬環境後，輸入以下指令來啟動它。 `$ source .venv/bin/activate` 如果您使用的是視窗： 1. 輸入以下指令以啟動虛擬環境。 `% .venv\Scripts\activate.bat`	
安裝依賴關係。	啟動虛擬環境之後，輸入下列命令以執行 install_deps.sh 指令碼。此指令碼會安裝必要的相依性。 `$ ./scripts/install_deps.sh`	DevOps 工程師, 一般 AWS, Python
部署堆疊。	輸入以下命令以合成和部署 CloudFormation 堆疊。 `$ npx cdk synth` `$ npx cdk deploy`	DevOps 工程師, 一般 AWS, Python

相關資源

AWS 文件

- 關於控制 (AWS Control Tower 文件)
- 控制程式庫 (AWS Control Tower 文件)
- AWS CDK 工具組命令 (AWS CDK 文件)
- 使用地形 (AWS Prescriptive Guidance) 部署和管理 AWS Control Tower 控制

其他資源

- Python

其他資訊

constants.py 檔案範例

下面是一個更新的 constants.py 文件的一個例子。

```
ACCOUNT_ID = 111122223333
AWS_CONTROL_TOWER_REGION = us-east-2
ROLE_ARN = "arn:aws:iam::111122223333:role/CT-Controls-Role"
GUARDRAILS_CONFIGURATION = [
    {
        "Enable-Control": {
            "AWS-GR_ENCRYPTED_VOLUMES",
            ...
        },
        "OrganizationalUnitIds": ["ou-1111-11111111", "ou-2222-22222222"...],
    },
    {
        "Enable-Control": {
            "AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED",
            ...
        },
        "OrganizationalUnitIds": ["ou-2222-22222222"...],
    },
]
```

IAM 政策

以下範例政策允許在將 AWS CDK 堆疊從部署帳戶部署到管理帳戶時啟用或停用 AWS Control Tower 控制所需的最低動作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "controltower:EnableControl",
                "controltower:DisableControl",
                "controltower:GetControlOperation",
                "controltower:ListEnabledControls",
                "organizations:AttachPolicy",
                "organizations:CreatePolicy",
                "organizations:DeletePolicy",
                "organizations:DescribeOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DetachPolicy",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListChildren",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListPoliciesForTarget",
                "organizations:ListRoots",
                "organizations:UpdatePolicy",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

信任政策

下列自訂信任政策允許部署帳戶中的特定 IAM 角色擔任管理帳戶中的 IAM 角色。取代以下項目：

- <DEPLOYMENT-ACCOUNT-ID>是部署帳戶的識別碼
- <DEPLOYMENT-ROLE-NAME>是部署帳戶中允許擔任管理帳戶中角色的角色名稱

```
{
```

```
    “Version”: “2012-10-17”,
    “Statement”: [
        {
            “Effect”: “Allow”,
            “Principal”: {
                “AWS”: “arn:aws:iam::<DEPLOYMENT-ACCOUNT-ID>:role/<DEPLOYMENT-ROLE-
NAME>”
            },
            “Action”: “sts:AssumeRole”,
            “Condition”: {}
        }
    ]
}
```

使用地形表單部署和管理 AWS Control Tower 控制

由伊克爾·雷納·富恩特（AWS）和伊万·吉拉迪（AWS）創建

程式碼儲存庫：使用 Terraform 部署和管理 AWS Control Tower 控制	環境：生產	技術：安全性、身分識別、合規性、雲端原生、基礎架構、管理與治理
工作負載：開源	AWS 服務：AWS Control Tower；AWS Organizations	

Summary

此模式說明如何使用 AWS Control Tower 控制項、 HashiCorp Terraform 和基礎設施即程式碼 (IaC) 來實作和管理預防性、偵測和主動式安全控制。控制項 (也稱為護欄) 是一項高階規則，可為您的整體 AWS Control Tower 環境提供持續的管控。例如，您可以使用控制項要求 AWS 帳戶記錄，然後在發生特定安全性相關事件時設定自動通知。

AWS Control Tower 可協助您實作預防性、偵測和主動式控制，以管理 AWS 資源並監控多個 AWS 帳戶的合規。每個控制項都會強制執行單一規則。在此模式中，您可以使用提供的 IaC 範本來指定要在環境中部署的控制項。

AWS Control Tower 控制適用於整個組織單位 (OU)，而且控制會影響 OU 中的每個 AWS 帳戶。因此，當使用者在您的 landing zone 中的任何帳戶中執行任何動作時，動作會受到控制 OU 的控制項的約束。

實作 AWS Control Tower 控制有助於為您的 AWS landing zone 建立堅實的安全基礎。通過使用此模式將控件部署為 IaC 通過 Terraform，您可以標準化 landing zone 中的控件，並更有效地部署和管理它們。

若要將 AWS Control Tower 控制部署為 IaC，您也可以使用 AWS Cloud Development Kit (AWS CDK) 而非地形。如需詳細資訊，請參閱使用 AWS CDK 和 AWS 部署和管理 AWS Control Tower 控制。CloudFormation

目標受眾

對於具有 AWS Control Tower、Terraform 和 AWS Organizations 相關經驗的使用者，建議使用此模式。

先決條件和限制

先決條件

- 以組織身分在 AWS 組織和 AWS Control Tower landing zone 中管理的有效 AWS 帳戶。如需指示，請參閱建立帳戶結構 (AWS Well-Architected 的實驗室)。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 管理帳戶中的 AWS Identity and Access Management (IAM) 角色，具有部署此模式的許可。如需必要許可和範例政策的詳細資訊，請參閱此模式的其他資訊一節中的 IAM 角色的最低權限許可。
- 在管理帳戶中擔任 IAM 角色的許可。
- 使用識別碼套用以服務控制原則 (SCP) 為基礎的控制項。必須啟動此 SCP，才能部署主動式控制。如需指示，請參閱不允許管理 AWS CloudFormation 登錄中的資源類型、模組和掛接。
- 地形 CLI，已安裝（地形文檔）。
- 地形 AWS 供應商，已設定 (地形文件)。
- 地形後端，配置（地形文檔）。

產品版本

- AWS Control Tower 3.0 版或更新版本
- 地形版本 1.5 或更高版本
- 地形 AWS 供應商 4.67 版或更新版本

架構

目標架構

本節提供此解決方案的高階概觀，以及範例程式碼所建立的架構。下圖顯示在 OU 中各個帳戶中部署的控制項。

AWS Control Tower 控制項會根據其行為和指導進行分類。

控制行為有三種主要類型：

1. 預防性控制的設計是為了防止動作發生。這些是透過 AWS Organizations 中的服務控制政策 (SCP) 來實作。預防控制的狀態可能是強制執行或未啟用。所有 AWS 區域都支援預防性控制。

2. Dete@@ ctive 測控制項的設計目的在於偵測特定事件發生時，並將動作記錄在中 CloudTrail。這些都是透過 AWS 組態規則來實作。偵測控制項的狀態可能是「清除」、「違規」或「未啟用」。Detective 控制僅適用於 AWS 控制塔支援的 AWS 區域。
3. 主動控制會掃描 AWS 佈建的資源， CloudFormation 並檢查它們是否符合您公司的政策和目標。不符合標準的資源將不會被佈建。這些都是使用 AWS CloudFormation 掛鉤實現的。主動控制的狀態為「通過」、「失敗」或「略過」。

控制指引是如何將每個控制項套用至 OU 的建議作法。AWS Control Tower 提供三種類別的指導：強制性、強烈建議和選修。控制項的指導獨立於其行為。如需詳細資訊，請參閱控制行為和指引。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS Config 提供 AWS 帳戶中資源的詳細檢視，以及資源的設定方式。它可協助您識別資源彼此之間的關聯性，以及它們的組態隨著時間的推移而變更的方式。
- AWS Control Tower 可協助您按照規範的最佳實務來設定和管理 AWS 多帳戶環境。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。

其他工具

- HashiCorp Terraform 是一種開放原始碼基礎結構即程式碼 (IaC) 工具，可協助您使用程式碼來佈建和管理雲端基礎架構和資源。

代碼存儲庫

使用 Terraform 儲存庫，可在 GitHub 部署和管理 AWS Control Tower 控制中取得此模式的程式碼。

最佳實務

- 用於部署此解決方案的 IAM 角色應遵循最低權限 (IAM 文件) 的原則。
- 遵循 AWS Control Tower 管理員的最佳實務 (AWS Control Tower 文件)。

史诗

在管理帳戶中啟用控制項

任務	描述	所需技能
複製儲存庫。	在 bash 外殼中，輸入以下命令。這會使用 Terraform 儲存庫複製部署和管理 AWS Control Tower 控制台控制。GitHub `git clone https://github.com/aws-samples/aws-control-tower-controls-terraform.git`	DevOps 工程師
編輯地形表單後端設定檔。	1. 在複製的存放庫中，開啟後端 .tf 檔案。 2. 編輯檔案以設定 Terraform 後端設定。您在此檔案中定義的組態視您的環境而定。如需詳細資訊，請參閱後端設定 (Terraform 文件)。 3. 儲存並關閉後端. tf 檔案。	DevOps 工程師, 地形
編輯地形表單提供者組態檔案。	1. 在複製的存放庫中，開啟提供者 .tf 檔案。 2. 編輯檔案以設定 Terraform 提供者組態。如需詳細資訊，請參閱提供者組態 (Terraform 文件)。將 AWS 區域設定為提供 AWS Control Tower API 的區域。 3. 儲存並關閉提供者. tf 檔案。	DevOps 工程師, 地形

任務	描述	所需技能
編輯組態檔案。	1. 在複製的儲存庫中，開啟變數 .tfvars 檔案。 2. 在`controls`區段的`control_names` 參數中，輸入控制項 API 識別碼。每個控制項對於可使用 AWS Control Tower 的每個區域都有唯一的 API 識別碼。若要尋找控制項識別碼，請執行下列動作： a. 在控制項中繼資料表中，找出您要啟用的控制項。 b. 在「控制 API 識別碼」的「區域」欄中，找出您進行 API 呼叫的區域的 API 識別碼，例如`arn:aws:controltower:us-east-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED` 。 c. 從區域標識符中提取控制標識符，例如`AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED` 。 3. 在`controls`區段中的`organizational_unit_ids` 參數中，輸入您要啟用控制項的組織單位 ID，例如`ou-1111-11111111` 。以雙引號輸入	DevOps 工程師，一般 AWS，地形

任務	描述	所需技能
	ID，並以逗號分隔多個 ID。如需如何擷取 OU ID 的相關資訊，請參閱檢視 OU 的詳細資訊。 4. 儲存並關閉變數 .tfvars 檔案。如需更新變數 .tfvars 檔案的範例，請參閱此模式的其他資訊一節。	
假設管理帳戶中的 IAM 角色。	在管理帳戶中，假設具有部署 Terraform 設定檔之權限的 IAM 角色。如需有關所需許可和範例政策的詳細資訊，請參閱其他資訊一節中 IAM 角色的最低權限許可。如需在 AWS CLI 中假設 IAM 角色的詳細資訊，請參閱在 AWS CLI 中使用 IAM 角色。	DevOps 工程師，一般 AWS

任務	描述	所需技能
部署組態檔案。	1. 輸入以下命令來初始化地形。 `$ terraform init -upgrade` 2. 輸入以下命令以預覽與目前狀態相比的變更。 `$ terraform plan -var-file="variables.tfvars"` 3. 檢閱 Terraform 計劃中的組態變更，並確認您要在組織中實作這些變更。 4. 輸入以下命令以部署資源。 `$ terraform apply -var-file="variables.tfvars"`	DevOps 工程師，一般 AWS，地形

(選用) 停用 AWS Control Tower 管理帳戶中的控制

任務	描述	所需技能
運行銷毀命令。	輸入下列命令以移除此模式所部署的資源。 `$ terraform destroy -var-file="variables.tfvars"`	DevOps 工程師，一般 AWS，地形

故障診斷

問題	解決方案
`Error: creating ControlTower Control ValidationException: Guardrail <control ID> is already enabled on organizational unit <OU ID>` 錯誤	您嘗試啟用的控制項已在目標 OU 中啟用。如果使用者透過 AWS 管理主控台、透過 AWS Control Tower 或 AWS Organizations 手動啟用控制，就會發生此錯誤。若要部署 Terraform 組態檔案，您可以使用下列其中一個選項。 選項 1：更新地形目前的狀態檔 您可以將資源匯入至 Terraform 目前的狀態檔案。當您重新執行`apply`命令時，Terraform 會略過此資源。執行下列動作，將資源匯入至目前的 Terraform 狀態： 1. 在 AWS Control Tower 管理帳戶中，輸入以下命令以擷取 OU 的 Amazon 資源名稱 (ARN) 清單，其中`<root-ID>` 是組織根目錄。如需有關擷取此 ID 的詳細資訊，請參閱檢視根的詳細資訊。 `aws organizations list-organizational-units-for-parent --parent-id <root-ID>` 2. 針對上一個步驟中傳回的每個 OU，輸入下列命令，其中`<OU-ARN>`是 OU 的 ARN。 `aws controltower list-enabled-controls --target-identifier <OU-ARN>` 3. 複製 ARN 並在所需模組中執行「地形」匯入，使其包含在「地形」狀態中。如需指示，請參閱匯入 (地形文件)。 4. 重複部署 Epics 區段中的組態中的步驟。

問題	解決方案
	選項 2：禁用控制 如果您在非生產環境中工作，則可以在主控台中停用控制項。重複部署 Epics 一節中的組態中的步驟，以重新啟用此功能。不建議在生產環境中使用此方法，因為有一段時間會停用控制項。如果您想要在生產環境中使用此選項，可以實作臨時控制，例如在 AWS Organizations 中暫時套用 SCP。

相關資源

AWS 文件

- 關於控制 (AWS Control Tower 文件)
- 控制程式庫 (AWS Control Tower 文件)
- 使用 AWS CDK 和 AWS (AWS Prescriptive Guidance) 部署和管 CloudFormation理 AWS Control Tower 控制

其他資源

- 地形
- 地形文件 CLI 文件

其他資訊

範例變數 .tfvars 檔案

以下是更新的變數 .tfvars 檔案的範例。

```
controls = [
    {
        control_names = [
            "AWS-GR_ENCRYPTED_VOLUMES",
            ...
```

```
        ],
        organizational_unit_ids = ["ou-1111-11111111", "ou-2222-22222222"...],
    },
    {
        control_names = [
            "AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED",
            ...
        ],
        organizational_unit_ids = ["ou-1111-11111111"...],
    },
]
```

IAM 角色的最低權限許可

此 APG 模式要求您在管理帳戶中擔任 IAM 角色。最佳做法是假設具有臨時權限的角色，並根據最小權限原則限制權限。以下範例政策允許啟用或停用 AWS Control Tower 控制所需的最低動作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "controltower:EnableControl",
                "controltower:DisableControl",
                "controltower:GetControlOperation",
                "controltower:ListEnabledControls",
                "organizations:AttachPolicy",
                "organizations:CreatePolicy",
                "organizations:DeletePolicy",
                "organizations:DescribeOrganization",
                "organizations:DetachPolicy",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListChildren",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListPoliciesForTarget",
                "organizations:ListRoots",
                "organizations:UpdatePolicy"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

部署管道，同時偵測多個程式碼交付項目中的安全性問題

代碼庫：簡單的代碼掃描管道	環境：PoC 或試點	技術：安全性、身分識別、合規性； DevOps
AWS 服務：AWS CloudFormation；AWS CodeBuild；AWS CodeCommit；AWS CodePipeline		

Summary

簡易程式碼掃描管道 (SCSP) 提供兩下建立程式碼分析管道，並行執 parallel 業界標準的開放原始碼安全性工具。這使開發人員可以檢查其代碼的質量和安全性，而無需安裝工具或甚至了解如何運行它們。這可協助您減少程式碼交付作業中的弱點和設定錯誤。它還可以減少組織安裝、研究和設定安全性工具所花費的時間。

在 SCSP 之前，使用此特定工具套件掃描代碼需要開發人員找到，手動安裝和配置軟件分析工具。即使是在本機安裝的 all-in-one 工具，例如自動安全性協助程式 (ASH)，也需要設定 Docker 容器才能執行。不過，使用 SCSP 時，一套業界標準的程式碼分析工具會自動在. AWS 雲端有了這個解決方案，您可以使用 Git 推送程式碼交付項目，然後您會收到視覺化輸出，其中包含哪些安全性檢查失敗的 at-a-glance 深入解析。

先決條件和限制

- 一個活躍的 AWS 帳戶
- 您要掃描是否有安全性問題的一或多個程式碼交付項目
- AWS Command Line Interface (AWS CLI)、已安裝及設定
- 安裝了 Python 3.0 或更高版本和點子版本 9.0.3 或更高版本
- Git，已安裝
- 在本地工作站上安裝 git 遠程代碼提交

架構

目標技術堆疊

- AWS CodeCommit 儲存庫
- AWS CodeBuild 項目
- AWS CodePipeline 管道
- Amazon Simple Storage Service (Amazon S3) 儲存貯體
- AWS CloudFormation 範本

目標架構

用於靜態代碼分析的 SCSP 是一個 DevOps 專案，旨在提供有關交付項目代碼的安全性反饋。

1. 在中 AWS Management Console，登入目標 AWS 帳戶。確認您位於 AWS 區域 要部署管道的位置。
2. 使用程式碼儲存庫中的 CloudFormation 範本來部署 SCSP 堆疊。這將創建一個新的 CodeCommit 存儲庫和 CodeBuild 項目。

   附註：作為替代部署選項，您可以在堆疊部署期間 CodeCommit 提供儲存庫的 Amazon 資源名稱 (ARN) 來使用現有的部署選項。

3. 將存放庫複製到您的本機工作站，然後將任何檔案新增至複製的存放庫中各自的資料夾。
4. 使用 Git 添加，提交和推送文件到 CodeCommit 存儲庫。
5. 推送至 CodeCommit 儲存庫會啟動 CodeBuild 工作。該 CodeBuild 項目使用安全工具掃描代碼交付項目。
6. 檢閱管線的輸出。發現錯誤層級問題的安全工具將導致管道中的動作失敗。修正這些錯誤或將其隱藏為誤報。在管道 S3 儲存貯體中 CodePipeline 或管道中的「動作」詳細資訊中檢閱工具輸出的詳細資訊。

工具

AWS 服務

- AWS CloudFormation協助您設定 AWS 資源、快速且一致地佈建資源，以及跨區域的整個生命週期進 AWS 帳戶 行管理。
- AWS CodeBuild是完全受控的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。

- AWS CodeCommit是一項版本控制服務，可協助您私下儲存和管理 Git 儲存庫，而無需管理您自己的原始檔控制系統。

其他工具

如需 SCSP 用來掃描程式碼交付項目的完整工具清單，請參閱. GitHub

代碼存儲庫

此病毒碼的程式碼可在中的簡易程式碼掃描管線 (SCSP) 存放庫中 GitHub找到。

史诗

部署 SCSP

任務	描述	所需技能
建立 CloudFormation 堆疊。	1. 登入 AWS Management Console。 2. 在主控台中，確認您位於要部署解決方案的目標區域。如需詳細資訊，請參閱選擇區域。 3. 選擇以下鏈接。這會在中開啟快速建立堆疊精靈 CloudFormation。 https://console.aws.amazon.com/cloudformation/home?#/stacks/create/review?templateURL=https://proservetools.s3.amazonaws.com/cft/scsp-pipeline-stack.template.json&stackName = SimpleCode ScanPipeline 4. 在快速建立堆疊精靈中，檢閱堆疊的參數設定，並根據	AWS DevOps，AWS 管理員

任務	描述	所需技能
	您的使用案例需要進行任何修改。 5. 選取 [我確認 AWS CloudFormation 可能會建立 IAM 資源]，然後選擇 [建立堆疊]。 這會建立 CodeCommit 儲存庫、 CodePipeline 管道、多個 CodeBuild 工作定義和 S3 儲存貯體。建置執行和掃描結果會複製到此值區。完全部署 CloudFormation 堆疊之後，SCSP 就可以使用了。	

使用管道

任務	描述	所需技能
檢查掃描結果。	1. 在 Amazon S3 主控台的儲存貯體中，選擇簡單的程式碼已刪除資源管線儲存貯體。 2. 選擇 scan_results 目錄，然後選擇具有最新掃描日期戳記的資料夾。 3. 檢閱此資料夾中的記錄檔，以檢閱管道中使用的安全性工具偵測到的任何問題。發現錯誤層級問題的安全工具將導致管道中的`failed`動作。如果它們是誤報，則需要修復或抑制這些問題。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
	附註：您也可以在 CodePipeline 主控台的 [處理行動詳細資訊] 區段中檢視工具輸出的詳細資訊 (通過掃描和失敗的掃描)。	

故障診斷

問題	解決方案
HashiCorp 不會掃描地形或 AWS CloudFormation 檔案。	請確定 Terraform (.tf) 和 CloudFormation (.yml、.yaml 或 .json) 檔案已放置在複製儲存庫中的適當資料夾中。 CodeCommit
命`git clone`令失敗。	請確定您已安裝，`git-remote-codecommit` 且您的 CLI 可存取具有讀取 CodeCommit 存放庫權限的 AWS 認證。
並行錯誤，例如`Project-level concurrent build limit cannot exceed the account-level concurrent build limit of 1 。`	通過選擇CodePipeline 控制台中的「發行更改」按鈕來重新運行管道。這是一個已知問題，在管道運行的前幾次似乎是最常見的。

相關資源

針對 SCSP 專案提供意見反應。

其他資訊

常見問答集

SCSP 專案是否與自動安全協助程式 (ASH) 相同？

沒有 當您需要使用容器執行程式碼掃描工具的 CLI 工具時，請使用 ASH。自動化安全協助程式 (ASH) 是一種工具，旨在降低新程式碼、基礎架構或 IAM 資源組態中發生安全違規的可能性。ASH 是可在本機執行的命令列公用程式。本機使用需要安裝容器環境並在系統上運作。

當您想要比 ASH 更簡單的設定管線時，請使用 SCSP。SCSP 不需要本機安裝。SCSP 的設計目的是在配管中個別執行檢查，並依工具顯示結果。SCSP 還通過設置 Docker 避免了很多開銷，並且它與操作系統（OS）無關。

SCSP 是否只適用於安全團隊？

否，任何人都可以部署管道以確定其代碼的哪些部分未通過安全檢查。例如，非安全性使用者可以使用 SCSP 檢查其程式碼，然後再與其安全團隊進行檢查。

如果我正在使用另一種類型的存儲庫，例如，或 Bitbucket，我可以 GitLab使用 SCSP 嗎？ GitHub

您可以將本地 git 存儲庫配置為指向兩個不同的遠程存儲庫。例如，您可以複製現有的 GitLab 存放庫、建立 SCSP 執行個體 (視需要指定 CloudFormation、Terraform 和 AWS Config 規則開發套件 (AWS RDK) 資料夾)，然後用`git remote add upstream <SCSPGitLink>`來將本機存放庫指向 S CodeCommit CSP 存放庫。這可讓程式碼變更先傳送至 SCSP、驗證，然後在進行任何其他更新以解決發現項目、推送至 GitLab GitHub、或 Bitbucket 儲存庫之後。如需有關多個遠端的詳細資訊，請參閱將提交推送至其他 Git 儲存庫 (AWS 部落格文章)。

注意：請小心漂移，例如避免通過 Web 界面進行更改。

貢獻和新增您自己的動作

SCSP 安裝程式會維護為 GitHub 專案，其中包含 SCSP AWS Cloud Development Kit (AWS CDK) 應用程式的原始程式碼。若要將其他檢查新增至管線，必須更新 AWS CDK 應用程式，然後再合成或部署到將執行管線的目標 AWS 帳戶 中。若要這麼做，請先複製 SCSP GitHub 專案，然後在`lib`資料夾中尋找堆疊定義檔案。

如果您想要新增額外的檢查， AWS CDK 程式碼中的`StandardizedCodeBuildProject`類別會讓新增動作變得非常簡單。提供名稱、描述和`/install`或`build`指令。 AWS CDK 通過使用合理的默認值創建 CodeBuild 項目。除了創建構建項目之外，您還需要將其添加到構建階段中的 CodePipeline 操作中。設計新檢查時，`FAIL`如果掃描工具偵測到問題或無法執行，則應該執行該處理行動。`PASS`如果掃描工具未檢測到任何問題，則應該執行此操作。如需設定工具的範例，請檢閱`Bandit`動作的程式碼。

如需有關預期輸入和輸出的詳細資訊，請參閱儲存庫文件。

如果您新增自訂動作，則需要使用`cdk deploy`或`cdk synth + CloudFormation deploy`來部署 SCSP。這是因為快速創建堆棧 CloudFormation 模板由回購所有者維護。

使用 AWS Config 為公有子網路部署偵探屬性型存取控制

創建者：阿爾貝托·梅嫩德斯 (AWS)

環境：PoC 或試點	技術：安全性、身分識別、合規性；網路	AWS 服務：AWS Config；Amazon SNS

Summary

分散式邊緣網路架構仰賴與虛擬私有雲 (VPC) 中工作負載一起執行的網路邊緣安全性。與較常見的集中式方法相比，這提供了前所未有的可擴展性。雖然在工作負載帳戶中部署公有子網路可以帶來好處，但它也會帶來新的安全風險，因為它會增加攻擊面。建議您只在這些 VPC 的公用子網路中部署 Elastic Load Balancing (ELB) 資源，例如應用程式負載平衡器或 NAT 閘道。在專用公有子網路中使用負載平衡器和 NAT 閘道，可協助您對輸入和輸出流量實作精細控制。

我們建議您實作預防性控制和偵測控制項，以限制可在公用子網路中部署的資源類型。如需使用以屬性為基礎的存取控制 (ABAC) 部署公用子網路預防性控制的詳細資訊，請參閱針對公用子網路部署預防性屬性型存取控制。雖然在大多數情況下都有效，但這些預防性控制可能無法解決所有可能的使用案例。因此，此模式建立在 ABAC 方法之上，可協助您設定有關公用子網路中部署之不相容資源的警示。解決方案會檢查彈性網路介面是否屬於公用子網路中不允許的資源。

為了實現這一目標，此模式使用 AWS Config 自訂規則和 ABAC。自訂規則會在建立或修改 elastic network interface 時處理彈性網路介面的組態。在高層級中，此規則會執行兩個動作來判斷網路介面是否相容：

1. 若要判斷網路介面是否在規則範圍內，規則會檢查子網路是否具有指示其為公有子網路的特定 AWS 標籤。例如，這個標籤可能是`IsPublicFacing=True`。
2. 如果網路界面部署在公有子網路中，則規則會檢查建立此資源的 AWS 服務。如果資源不是 ELB 資源或 NAT 閘道，則會將資源標示為不相容。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS 組態，在工作負載帳戶中設定

- 在工作負載帳戶中部署所需資源的權限
- 具有公用子網路的 VPC
- 正確套用以識別目標公用子網路的標籤
- (選用) AWS Organizations 中的組織
- (選用) 作為 AWS Config 和 AWS Security Hub 委派管理員的中央安全帳戶

架構

目標架構

此圖展示了以下要點：

1. 部署或修改 elastic network interface 資源 (`AWS::EC2::NetworkInterface`) 時，AWS Config 會擷取事件和組態。
2. AWS Config 會將此事件與用於評估組態的自訂規則進行比對。
3. 會叫用與此自訂規則相關聯的 AWS Lambda 函數。此函數會評估資源並套用指定的邏輯`COMPLIANT`，以判斷資源組態是否為`NON_COMPLIANT`或`NOT_APPLICABLE`。
4. 如果確定資源為`NON_COMPLIANT`，AWS Config 會透過亞馬遜簡單通知服務 (Amazon SNS) 傳送警示。

> 注意：如果此帳戶是 AWS Organizations 的成員帳戶，您可以透過 AWS Config 或 AWS Security Hub 將合規資料傳送到中央安全帳戶。

函 Lambda 評估邏輯

下圖顯示 Lambda 函數用來評估 elastic network interface 合規性的邏輯。

自動化和規模

這種模式是偵探解決方案。您也可以使用修正規則來補充它，以自動解決任何不相容的資源。如需詳細資訊，請參閱使用 AWS Config 規則修復不合規資源。

您可以通過以下方式擴展此解決方

- 強制套用您建立的對應 AWS 標籤，以識別面向公開的子網路。如需詳細資訊，請參閱 AWS Organizations 文件中的標籤政策。
- 設定中央安全帳戶，將 AWS Config 自訂規則套用至組織中的每個工作負載帳戶。如需詳細資訊，請參閱 AWS 中的大規模自動化組態合規 (AWS 部落格文章)。
- 將 AWS Config 與 AWS Security Hub 整合，以便大規模擷取、集中和通知。如需詳細資訊，請參閱 AWS Security Hub 文件中的設定 AWS 組態。

工具

- AWS Config 提供 AWS 帳戶中資源的詳細檢視，以及資源的設定方式。它可協助您識別資源彼此之間的關聯性，以及其組態隨時間變更的情況。
- E@@ lastic Load Balancing (ELB) 可將傳入的應用程式或網路流量分散到多個目標。例如，您可以將流量分配到一或多個可用區域中的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、容器和 IP 地址。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

最佳實務

有關開發自訂 AWS Config 規則的更多範例和最佳實務，請參閱上的官方 AWS Config 規則儲存庫 GitHub。

史诗

部署解決方案

任務	描述	所需技能
建立 Lambda 函數。	1. 登入 AWS 管理主控台，然後開啟 AWS Lambda 主控台。 2. 在 Functions (函數) 頁面上，選擇 Create function (建立函數)。 3. 選取從頭開始撰寫。 4. 在 [基本] 資訊窗格中，對於 [函數名稱]，輸入名稱。 5. 對於「執行階段」，請選擇 Python 3.12。 6. 保留架構設定為 x86_64。 7. 選擇建立函數。 8. 選擇 程式碼 標籤。 9. 在檔案總管中，選擇 lambda_function.py。 10將此模式的「其他資訊」區段中提供的範例程式碼貼到 lambda_function.py 索引標籤中。自訂範例程式碼以識別`evaluate_change_notification_compliance` 函數中的任何自訂評估邏輯。 11選擇部署。	一般 AWS
將權限新增至 Lambda 函數的執行角色。	1. 在導覽視窗中，選擇函數。 2. 選擇您剛剛創建的功能。	一般 AWS

任務	描述	所需技能
	3. 選擇 組態 ，然後選擇 許可。 4. 選擇角色名稱以在 AWS Identity and Access Management (IAM) 主控台中開啟角色。 5. 在 [權限原則] 下，選擇 [新增權限]，然後選擇 [建立內嵌原則]。 6. 選擇 JSON。 7. 將以下策略粘貼到策略編輯器中。這可讓 Lambda 函數執行下列動作： • 取得子網路標籤的詳細資訊。 • 將合規結果傳回 AWS Config。 `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "config:PutEvaluations", "ec2:DescribeSubnets" ], "Resource": "*", "Effect": "Allow"`	

任務	描述	所需技能
	<pre> }] }</pre> 8. 選擇下一步。 9. 輸入政策名稱，然後選擇 Create policy (建立政策)。	
檢索 Lambda 函數 Amazon 資源名稱（ARN）。	1. 開啟 Lambda 主控台。 2. 在導覽視窗中，選擇函數。 3. 選擇您剛剛創建的功能。 4. 在 [函數概觀] 區段的 [函數 ARN] 下，複製值。	一般 AWS

任務	描述	所需技能
建立 AWS Config 自訂規則。	1. 開啟 AWS 組態主控台，網址為 https://console.aws.amazon.com/config/。 2. 在 Rules (規則) 頁面，選擇 Add rule (新增規則)。 3. 在 [指定規則類型] 頁面上，選擇 [建立自訂 Lambda 規則]，然後選擇 [下一步]。 4. 在 [設定規則] 頁面上，執行下列動作： a. 輸入名稱和說明。 b. 如果是 AWS Lambda 函數 ARN，請貼上您先前複製的 ARN。 c. 針對 觸發類型，選擇 組態有所變更時。 d. 對於變更範圍，選取資源。 e. 對於資源類型，請選擇 AWS EC2 NetworkInterface。 f. 選擇下一步。 5. 在 [檢閱並建立] 頁面上，確認您的規則，然後選擇 [儲存]。	一般 AWS

任務	描述	所需技能
設定通知。	1. 請依照建立 Amazon SNS 主題中的指示來建立 Amazon SNS 主題。 2. 遵循訂閱 Amazon SNS 主題中的指示，設定接收 Amazon SNS 主題通知的端點。 3. 按照使用 AWS AWS Config 在 AWS 資源不合規時如何收到通知中的說明，為不合規的資源設定自訂 Amazon EventBridge 規則。	一般 AWS

測試解決方案

任務	描述	所需技能
建立符合規範的資源。	1. 使用下列指示在公用子網路中建立其中一個受支援的資源： • 建立 NAT 閘道 • 開始使用網路負載平衡器 • 建立應用程式負載平衡器 2. 建立資源之後，AWS Config 自訂規則會評估與資源關聯的彈性網路界面。它會將這些網路介面標記為`COMPLIANT` 。您可以按照以下步驟在 AWS Config 中檢視資源： a. 開啟 AWS 組態主控台，網址為 https://c	一般 AWS

任務	描述	所需技能
	onsole.aws.amazon.com/config/。 b. 在 [規則] 頁面上，選擇您的規則。 c. 在 [規則詳細資料] 頁面上，移至頁面底部。 d. 在範圍內的資源下，選取相容。確認您看到已建立之網路介面的 ID。 e. 如需有關網路介面配置的詳細資訊，請選擇資源 ID。	

任務	描述	所需技能
建立不符合標準的資源。	1. 使用下列指示在公用子網路中建立不相容的資源： • 啟動一個 Amazon EC2 實例 • 建立 Amazon Relational Database Service 服務 (Amazon RDS) 資料庫執行個體 • 建立虛擬私人雲端端點 2. 建立資源之後，AWS Config 自訂規則會評估與資源關聯的彈性網路界面。它會將這些網路介面標記為NON_COMPLIANT 。您可以按照以下步驟在 AWS Config 中檢視資源： a. 開啟 AWS 組態主控台，網址為 https://console.aws.amazon.com/config/。 b. 在 [規則] 頁面上，選擇您的規則。 c. 在 [規則詳細資料] 頁面上，移至頁面底部。 d. 在範圍內的資源下，選取NonCompliant。確認您看到已建立之網路介面的 ID。 e. 如需有關網路介面配置的詳細資訊，請選擇資源 ID。	一般 AWS

任務	描述	所需技能
	3. 確認您在 Amazon SNS 中設定的端點收到通知。	
建立不適用的資源。	1. 在私有子網路中，建立任何需要 elastic network interface 的資源。 2. 建立資源之後，AWS Config 自訂規則會評估與資源關聯的彈性網路界面。它會將這些網路介面標記為NOT_APPLICABLE 。這些資源不會顯示在 AWS 組態主控台中。	一般 AWS

相關資源

AWS 文件

- 設定 AWS Config
- AWS Config 自訂規則
- 適用於 AWS 的 ABAC
- 為公用子網路部署預防性屬性型存取控制

其他 AWS 資源

- 在 AWS 中大規模自動化組態合規
- 閘道 Load Balancer 的分散式檢測架構

其他資訊

以下是提供用於示範目的的 Lambda 函數範例。

```
import boto3
import json
```

```
import os

# Init clients
config_client = boto3.client('config')
ec2_client = boto3.client('ec2')

def lambda_handler(event, context):

    # Init values
    compliance_value = 'NOT_APPLICABLE'
    invoking_event = json.loads(event['invokingEvent'])
    configuration_item = invoking_event['configurationItem']

    status = configuration_item['configurationItemStatus']
    eventLeftScope = event['eventLeftScope']

    # First check if the event configuration applies. Ex. resource event is not delete
    if (status == 'OK' or status == 'ResourceDiscovered') and not eventLeftScope:
        compliance_value = evaluate_change_notification_compliance(configuration_item)


    config_client.put_evaluations(
       Evaluations=[
           {
               'ComplianceResourceType': invoking_event['configurationItem']
['resourceType'],
               'ComplianceResourceId': invoking_event['configurationItem']
['resourceId'],
               'ComplianceType': compliance_value,
               'OrderingTimestamp': invoking_event['configurationItem']
['configurationItemCaptureTime']
           },
       ],
       ResultToken=event['resultToken'])

# Function with the logs to evaluate the resource
def evaluate_change_notification_compliance(configuration_item):
    is_in_scope = is_in_scope_subnet(configuration_item['configuration']['subnetId'])

    if (configuration_item['resourceType'] != 'AWS::EC2::NetworkInterface') or not
 is_in_scope:
        return 'NOT_APPLICABLE'

    else:
```

```
        alb_condition = configuration_item['configuration']['requesterId'] in ['amazon-
elb']
        nlb_condition = configuration_item['configuration']['interfaceType'] in
 ['network_load_balancer']
        nat_gateway_condition = configuration_item['configuration']['interfaceType'] in
 ['nat_gateway']

        if alb_condition or nlb_condition or nat_gateway_condition:
            return 'COMPLIANT'
    return 'NON_COMPLIANT'

# Function to check if elastic network interface is in public subnet
def is_in_scope_subnet(eni_subnet):

    subnet_description = ec2_client.describe_subnets(
        SubnetIds=[eni_subnet]
    )

    for subnet in subnet_description['Subnets']:
        for tag in subnet['Tags']:
            if tag['Key'] == os.environ.get('TAG_KEY') and tag['Value'] ==
 os.environ.get('TAG_VALUE'):
                return True

    return False
```

為公用子網路部署預防性屬性型存取控制

由喬爾·阿爾弗雷多·努涅斯·岡薩雷斯（AWS）和塞繆爾·奧爾特加桑喬（AWS）創建

環境：PoC 或試點	技術：安全性、身分識別、合規性；網路；內容交付	AWS 服務：AWS Organizations；AWS Identity and Access Management

Summary

在集中式網路架構中，檢查和邊緣虛擬私有雲 (VPC) 會集中所有入站和輸出流量，例如進出網際網路的流量。不過，這可能會造成瓶頸，或導致達到 AWS 服務配額的限制。與較常見的集中式方法相比，將網路邊緣安全性與 VPC 中的工作負載一起部署，可提供前所未有的擴充性。這稱為分散式邊緣架構。

雖然在工作負載帳戶中部署公有子網路可以帶來好處，但它也會帶來新的安全風險，因為它會增加攻擊面。建議您只在這些 VPC 的公用子網路中部署 Elastic Load Balancing (ELB) 資源，例如應用程式負載平衡器或 NAT 閘道。在專用公有子網路中使用負載平衡器和 NAT 閘道，可協助您對輸入和輸出流量實作精細控制。

以屬性為基礎的存取控制 (ABAC) 是根據使用者屬性 (例如部門、工作角色和專案團隊名稱) 建立精細權限的做法。如需詳細資訊，請參閱適用於 AWS 的 ABAC。ABAC 可以為工作負載帳戶中的公用子網路提供護欄。這有助於應用程序團隊保持靈活性，而不會影響基礎架構的安全性。

此模式描述如何透過 AWS Organizations 中的服務控制政策 (SCP) 實作 ABAC，以及 AWS 身分與存取管理 (IAM) 中的政策來協助保護公有子網路的安全。您可以將 SCP 套用至組織的成員帳戶或組織單位 (OU)。這些 ABAC 政策允許使用者在目標子網路中部署 NAT 閘道，並防止他們部署其他 Amazon Elastic Compute Cloud (Amazon EC2) 資源，例如 EC2 執行個體和彈性網路界面。

先決條件和限制

先決條件

- AWS Organizations 中的組織
- AWS Organizations 根帳戶的管理存取權
- 在組織中，用於測試 SCP 的作用中成員帳戶或 OU

限制

- 此解決方案中的 SCP 無法防止使用服務連結角色的 AWS 服務在目標子網路中部署資源。這些服務的範例包括 Elastic Load Balancing (ELB)、Amazon Elastic Container Service (Amazon ECS) 和 Amazon Relational Database Service 服務 (Amazon RDS)。如需詳細資訊，請參閱 AWS Organizations 文件中的 SCP 對許可的影響。實施安全控制以檢測這些異常。

架構

目標技術堆疊

- SCP 適用於 AWS 帳戶或 AWS Organizations 中的 OU
- 下列 IAM 角色：
  - `AutomationAdminRole`— 用於在實施 SCP 後修改子網路標籤和建立 VPC 資源
  - `TestAdminRole`— 用於測試 SCP 是否阻止其他 IAM 主體 (包括具有管理存取權的主體) 執行保留給 `AutomationAdminRole`

目標架構

1. 您可以在目標帳戶中建立 `AutomationAdminRole` IAM 角色。此角色具有管理網路資源的權限。請注意下列此角色專屬的權限：
   - 此角色可以建立 VPC 和公用子網路。
   - 此角色可修改目標子網路的標籤指派。
   - 此角色可以管理自己的權限。
2. 在 AWS Organizations 中，您可以將 SCP 套用到目標 AWS 帳戶或 OU。如需原則範例，請參閱此模式中的其他資訊。
3. CI/CD 管線中的使用者或工具可以擔任將標籤套用至目`SubnetType`標子網路的`AutomationAdminRole`角色。
4. 透過假設其他 IAM 角色，您組織中已授權的 IAM 主體可以管理目標子網路中的 NAT 閘道，以及 AWS 帳戶中其他允許的聯網資源，例如路由表。使用 IAM 政策授予這些許可。如需詳細資訊，請參閱 Amazon VPC 的身分識別和存取管理。

自動化和規模

為協助保護公有子網路，必須套用對應的 AWS 標籤。套用 SCP 之後，NAT 閘道是授權使用者可以在具有標籤的子網路中建立的唯一 Amazon EC2 資源類型。`SubnetType:IFA` (IFA指面向網際網路的資產。) SCP 可防止建立其他 Amazon EC2 資源，例如執行個體和彈性網路界面。 我們建議您使用具有`AutomationAdminRole`角色的 CI/CD 管線來建立 VPC 資源，以便將這些標籤正確套用至公用子網路。

工具

AWS 服務

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。在 AWS Organizations 中，您可以實作服務控制政策 (SCP)，這是一種可用來管理組織中許可的政策類型。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

史诗

套用 SCP

任務	描述	所需技能
建立測試管理員角色。	建立在目標 AWS 帳戶`TestAdminRole` 中命名的 IAM 角色。將 AdministratorAccessAWS 受管 IAM 政策附加到新角色。如需指示，請參閱 IAM 說明文件中的建立角色以將許可委派給 IAM 使用者。	AWS 管理員
建立自動化管理員角色。	1. 建立在目標 AWS 帳戶`AutomationAdminRole` 中命名的 IAM 角色。	AWS 管理員

任務	描述	所需技能
	2. 將 AdministratorAccesssAWS 受管 IAM 政策附加到新角色。 以下是可用來測試000000000000 帳戶角色的信任原則範例。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "arn:aws:iam::000000000000:root" ] }, "Action": "sts:AssumeRole", "Condition": {} } ] }`	

任務	描述	所需技能
建立並附加 SCP。	1. 使用 [其他資訊] 區段中提供的範例程式碼，建立安全性控制原則。如需指示，請參閱 AWS Organizations 文件中的建立 SCP。 2. 將 SCP 連接到目標 AWS 帳戶或 OU。如需指示，請參閱 AWS Organizations 文件中的連接和卸離服務控制政策。	AWS 管理員

測試 SCP

任務	描述	所需技能
建立 VPC 或子網路。	1. 假設目標 AWS 帳戶中的`TestAdminRole` 角色。 2. 嘗試在現有 VPC 中建立 VPC 或新的公用子網路。如需指示，請參閱 Amazon VPC 文件中的建立 VPC、子網路和其他 VPC 資源。您不應該能夠建立這些資源。 3. 假設`AutomationAdminRole` 角色，然後重試上一個步驟。現在，您應該可以建立網路資源。	AWS 管理員
管理標籤。	1. 假設目標 AWS 帳戶中的`TestAdminRole` 角色。 2. 將`SubnetType:IFA` 標籤新增至可用的公用子網路。	AWS 管理員

任務	描述	所需技能
	您應該可以添加此標籤。如需有關如何透過 AWS Command Line Interface (AWS CLI) (AWS CLI) 新增標籤的指示，請參閱 AWS CLI 命令參考中的建立標籤。 3. 在不變更您的認證的情況下，嘗試修改指派給此子網路的SubnetType:IFA 標籤。您不應該能夠修改此標籤。 4. 假設AutomationAdminRole 角色，然後重試先前的步驟。此角色應該能夠新增和修改此標籤。	
在目標子網路中部署資源。	1. 假設TestAdminRole 角色。 2. 對於具有SubnetType:IFA 標籤的公共子網路，請嘗試建立 EC2 執行個體。如需指示，請參閱 Amazon EC2 文件中的啟動執行個體。在此子網路中，除了 NAT 閘道之外，您無法建立、修改或刪除任何 Amazon EC2 資源。 3. 在相同的子網路中，建立 NAT 閘道。如需指示，請參閱 Amazon VPC 文件中的建立 NAT 閘道。您應該能夠在此子網路中建立、修改或刪除 NAT 閘道。	AWS 管理員

任務	描述	所需技能
管理 AutomationAdminRole 角色。	1. 假設TestAdminRole 角色。 2. 嘗試修改AutomationAdminRole 角色。如需指示，請參閱 IAM 文件中的修改角色。您不應該能夠修改此角色。 3. 假設AutomationAdminRole 角色，然後重試上一個步驟。現在，您應該可以修改角色了。	AWS 管理員

清除

任務	描述	所需技能
清理已部署的資源。	1. 從 AWS 帳戶或 OU 中斷連結 SCP。如需指示，請參閱 AWS Organizations 文件中的卸離 SCP。 2. 刪除 SCP。如需指示，請參閱刪除 SCP (AWS Organizations 文件)。 3. 刪除AutomationAdminRole 角色和TestAdminRole 角色。如需指示，請參閱 IAM 文件中的刪除角色。 4. 刪除您為此解決方案建立的所有網路資源，例如 VPC 和子網路。	AWS 管理員

相關資源

AWS 文件

- 附加和分離 SCP
- 建立、更新和刪除 SCP
- 使用 AWS Config 為公有子網路部署偵探屬性型存取控制
- Detective 控制
- 服務授權參考
- 標記 AWS 資源
- 什麼是適用於 AWS 的 ABAC？

其他 AWS 參考資料

- 使用 AWS Organizations 中的服務控制政策保護用於授權的資源標籤 (AWS 部落格文章)

其他資訊

下列服務控制原則是您可以在組織中測試此方法的範例。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyVPCActions",
      "Effect": "Deny",
      "Action": [
        "ec2:CreateVPC",
        "ec2:CreateRoute",
        "ec2:CreateSubnet",
        "ec2:CreateInternetGateway",
        "ec2:DeleteVPC",
        "ec2:DeleteRoute",
        "ec2:DeleteSubnet",
        "ec2:DeleteInternetGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:*"
      ],
```

```
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
      }
    },
    {
      "Sid": "AllowNATGWOnIFASubnet",
      "Effect": "Deny",
      "NotAction": [
        "ec2:CreateNatGateway",
        "ec2:DeleteNatGateway"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*"
      ],
      "Condition": {
        "ForAnyValue:StringEqualsIfExists": {
          "aws:ResourceTag/SubnetType": "IFA"
        },
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
      }
    },
    {
      "Sid": "DenyChangesToAdminRole",
      "Effect": "Deny",
      "NotAction": [
        "iam:GetContextKeysForPrincipalPolicy",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListAttachedRolePolicies",
        "iam:ListInstanceProfilesForRole",
        "iam:ListRolePolicies",
        "iam:ListRoleTags"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AutomationAdminRole"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AutomationAdminRole"]
        }
```

```
      }
    },
    {
      "Sid": "allowbydefault",
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

使用地形表單部署 AWS WAF 解決方案的安全自動化

由拉胡爾·沙拉德·蓋克瓦德博士（AWS）和泰米爾塞爾文 P（AWS）創建

代碼存儲庫：aws-waf-automation-terraform-示例	環境：PoC 或試點	技術：安全性、身分識別、合規性；基礎架構；內容交付；DevOps
工作負載：所有其他工作	AWS 服務：AWS WAF	

Summary

AWS WAF 是一種 Web 應用程式防火牆，可使用可在 Web 存取控制清單 (ACL) 中定義和部署的可自訂規則，協助保護應用程式免於遭受常見入侵。設定 AWS WAF 規則可能具有挑戰性，尤其是對於沒有專門安全團隊的組織而言。為了簡化此程序，Amazon Web Services (AWS) 提供 AWS WAF 的安全自動化解決方案，該解決方案會自動部署單一 Web ACL，其中包含一組可篩選網路攻擊的 AWS WAF 規則。在 Terraform 部署期間，您可以指定要包含哪些保護功能。部署此解決方案之後，AWS WAF 會檢查對現有 Amazon CloudFront 分發或應用程式負載平衡器的 Web 請求，並封鎖任何不符合規則的請求。

AWS WAF 解決方案的安全自動化可 CloudFormation 根據 AWS WAF 實作指南的安全自動化指南中的指示使用 AWS 來部署。此模式為使用 HashiCorp Terraform 做為其慣用基礎結構即程式碼 (IaC) 工具來佈建和管理其雲端基礎結構的組織提供替代部署選項。部署此解決方案時，Terraform 會自動套用雲端中的變更，並部署和設定 AWS WAF 設定和保護功能。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 使用必要許可安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱入門 (AWS CLI 文件)。
- 安裝和配置地形。如需詳細資訊，請參閱安裝地形 (地形文件)。

產品版本

- AWS CLI 版本 2.4.25 或更新版本
- 地形版本 1.1.9 或更高版本

架構

目標架構

此模式會為 AWS WAF 解決方案部署安全自動化。如需目標架構的詳細資訊，請參閱 AWS WAF 安全自動化指南中的架構概觀。如需有關此部署中 AWS Lambda 自動化、應用程式日誌剖析器、AWS WAF 日誌剖析器、IP 清單剖析器和存取處理常式的詳細資訊，請參閱 AWS WAF 安全自動化實作指南中的元件詳細資訊。

地形部署

當您運行時`terraform apply`，地形會執行以下操作：

1. Terraform 會根據測試 .tfvars 檔案的輸入來建立 IAM 角色和 Lambda 函數。
2. Terraform 會根據測試 .tfvars 檔案的輸入來建立 AWS WAF ACL 規則和 IP 集。
3. Terraform 會根據測試 .tfvars 檔案的輸入，建立 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體、亞馬遜 EventBridge 規則、AWS Glue 資料庫表和亞馬遜雅典娜工作群組。
4. Terraform 會部署 AWS CloudFormation 堆疊以佈建自訂資源。
5. Terraform 會根據來自測試 .tfvars 檔案的指定輸入來建立 Amazon API Gateway 資源。

自動化和規模

您可以使用此模式為多個 AWS 帳戶和 AWS 區域建立 AWS WAF 規則，以便在整個 AWS 雲端環境中部署 AWS WAF 解決方案的安全自動化。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS WAF 是一種 Web 應用程式防火牆，可協助您監控轉寄至受保護 Web 應用程式資源的 HTTP 和 HTTPS 請求。

其他服務

- Git 是一個開放原始碼的分散式版本控制系統。
- HashiCorp Terraform 是一個命令列介面應用程式，可協助您使用程式碼來佈建和管理雲端基礎結構和資源。

代碼存儲庫

此模式的程式碼可在 GitHub AWS WAF 自動化使用地形儲存庫取得。

最佳實務

- 將靜態文件放在單獨的 S3 存儲桶中。
- 避免對變量進行硬編碼。
- 限制自訂指令碼的使用。
- 採用命名慣例。

史诗

設定您的本機工作站

任務	描述	所需技能
安裝 Git。	按照入門（Git 網站）中的說明在本地工作站上安裝 Git。	DevOps 工程師
複製儲存庫。	在您的本機工作站上，輸入下列指令以複製程式碼儲存庫。若要複製完整指令 (包括 repo URL)，請參閱此模式的其他資訊一節。 `git clone <repo-URL >.git`	DevOps 工程師
更新變數。	1. 輸入下列命令，導覽至複製的目錄。	DevOps 工程師

任務	描述	所需技能
	`cd terraform-aws-waf-automation` 2. 在任何文字編輯器中，開啟測試 .t fvars 檔案。 3. 更新測試 .t fvars 檔案中變數的值。 4. 儲存並關閉檔案。	

使用地形佈建目標架構

任務	描述	所需技能
初始化地形組態。	輸入下列命令以初始化包含 Terraform 組態檔的工作目錄。 `terraform init`	DevOps 工程師
預覽地形計劃。	輸入以下命令。Terraform 會評估組態檔案，以判斷宣告資源的目標狀態。然後將目標狀態與目前狀態進行比較，並建立計劃。 `terraform plan -var-file ="testing.tfvars"`	DevOps 工程師
驗證計劃。	檢閱計劃並確認其在目標 AWS 帳戶中設定所需的架構。	DevOps 工程師
部署解決方案。	1. 輸入下列指令以套用計劃。	DevOps 工程師

任務	描述	所需技能
	`terraform apply -var-file="testing.tfvars"` 2. 輸入 yes 以確認。Terraform 會建立、更新或銷毀基礎結構，以達到組態檔案中宣告的目標狀態。如需序列的詳細資訊，請參閱此模式的「架構」一節中的 Terraform 部署。	

驗證和清理

任務	描述	所需技能
驗證變更。	1. 在 Terraform 主控台中，確認輸出符合預期的結果。 2. 登入 AWS 管理主控台。 3. 確認 Terraform 主控台中的輸出是否已成功部署在您的 AWS 帳戶中。	DevOps 工程師
(選擇性) 清理基礎結構。	如果您想要移除此解決方案所做的所有資源和組態變更，請執行下列動作： 1. 在地形控制台中，輸入以下命令。 `terraform destroy -var-file="testing.tfvars”` 2. 輸入 yes 以確認。	DevOps 工程師

故障診斷

問題	解決方案
`WAFV2 IPSet: WAFOptimisticLockException 錯誤`	如果您在執行`terraform destroy`命令時收到此錯誤，則必須手動刪除 IP 集。如需指示，請參閱刪除 IP 集 (AWS WAF 文件)。

相關資源

AWS 參考資料

- AWS WAF 的安全自動化實作指南
- 適用於 AWS WAF 的安全自動化 (AWS 解決方案程式庫)
- AWS WAF 的安全自動化常見問答集

地形參考

- 地形後端配置
- 地形 AWS 供應商-文件和用法
- 地形 AWS 供應商 (儲存庫) GitHub

其他資訊

以下命令克隆此模式的 GitHub 存儲庫。

```
git clone https://github.com/aws-samples/aws-waf-automation-terraform-samples.git
```

使用 Step Functions 數使用 IAM 存取分析器動態產生 IAM 政策

創建者：托馬斯·斯科特 (AWS)、阿迪爾·卡納比 (AWS)、公園·凡·布利德芬 (AWS) 和拉法爾·波拉塞克 (AWS)

程式碼儲存庫：自動 IAM 存取分析器角色政策產生	環境：PoC 或試點	技術：安全性、身分識別、合規性；無伺服器
AWS 服務：AWS IAM 存取分析器；AWS Lambda；AWS Step Functions；AWS Identity and Access Management		

Summary

最低權限是授與執行工作所需的最低權限的安全性最佳作法。在已經使用中的 Amazon Web Services (AWS) 帳戶中實作最低權限存取可能具有挑戰性，因為您不想透過變更其許可，無意中阻止使用者執行其任務。您必須先了解帳戶使用者正在執行的動作和資源，才能實作 AWS Identity and Access Management (IAM) 政策變更。

此模式旨在協助您套用最低權限存取原則，而不會阻塞或降低團隊生產力。本文說明如何使用 IAM 存取分析器和 AWS Step Functions 根據帳戶中目前執行的動作，為您的角色動態產生 up-to-date IAM 政策。新政策旨在允許目前的活動，但會移除任何不必要的提升權限。您可以透過定義允許和拒絕規則來自訂產生的原則，而解決方案會整合您的自訂規則。

此模式包括使用 AWS Cloud Development Kit (AWS CDK) 或 HashiCorp 地形用 CDK (CDKTF) 實作解決方案的選項。然後，您可以使用持續整合和持續傳遞 (CI/CD) 管線，將新原則與角色相關聯。如果您有多帳戶架構，則可以在任何想要為角色產生更新 IAM 政策的帳戶中部署此解決方案，從而提高整個 AWS 雲端環境的安全性。

先決條件和限制

先決條件

- 啟用 CloudTrail 追蹤的作用中 AWS 帳戶。
- 下列項目的 IAM 許可：

  - 建立和部署 Step Functions 工作流程。如需詳細資訊，請參閱 AWS Step Functions 的動作、資源和條件金鑰 (步驟函數文件)。
  - 建立 AWS Lambda 函數。如需詳細資訊，請參閱執行角色和使用者權限 (Lambda 文件)。
  - 建立 IAM 角色。如需詳細資訊，請參閱建立角色以將許可委派給 IAM 使用者 (IAM 說明文件)。
- 安裝了 npm。如需詳細資訊，請參閱下載和安裝 Node.js 和 npm (npm 文件)。
- 如果您使用 AWS CDK 部署此解決方案 (選項 1)：
  - 已安裝和設定的 AWS CDK 工具組。如需詳細資訊，請參閱安裝 AWS CDK (AWS CDK 文件)。
- 如果您使用 CDKTF 部署此解決方案（選項 2）：
  - CDKTF，已安裝和配置。如需詳細資訊，請參閱安裝地形的 CDK (CDKTF 文件)。
  - 地形，安裝和配置。如需詳細資訊，請參閱開始使用 (Terraform 文件)。
- 為您的 AWS 帳戶在本機安裝和設定 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI (AWS CLI 文件)。

限制

- 此模式不會將新的 IAM 政策套用至角色。在此解決方案結束時，新的 IAM 政策會儲存在儲存 CodeCommit 庫中。您可以使用 CI/CD 管線將原則套用至帳戶中的角色。

架構

目標架構

1. 定期排程的 Amazon EventBridge 事件規則會啟動 Step Functions 工作流程。您可以將此再生排程定義為設定此解決方案的一部分。
2. 在 Step Functions 數工作流程中，Lambda 函數會產生日期範圍，以便在分析 CloudTrail 記錄中的帳戶活動時使用。
3. 下一個工作流程步驟會呼叫 IAM 存取分析器 API 以開始產生政策。
4. IAM 存取分析器使用您在設定期間指定的角色的 Amazon 資源名稱 (ARN)，分析指定日期費率內的活動 CloudTrail 記錄。IAM 存取分析器會根據活動產生 IAM 政策，該政策僅允許角色在指定日期範圍內使用的動作和服務。完成此步驟後，此步驟會產生作業 ID。
5. 下一個工作流程步驟會每隔 30 秒檢查一次工作 ID。偵測到工作 ID 時，此步驟會使用工作 ID 呼叫 IAM 存取分析器 API 並擷取新的 IAM 政策。IAM 存取分析器會以 JSON 檔案的形式傳回政策。

6. 下一<IAM role name>個工作流程步驟會將 /policy.json 檔案放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。您可以將此 S3 儲存貯體定義為設定此解決方案的一部分。
7. Amazon S3 事件通知會啟動 Lambda 函數。
8. Lambda 函數會從 S3 儲存貯體擷取政策、整合您在允許 .json 和拒絕 .json 檔案中定義的自訂規則，然後將更新的政策推送至。 CodeCommit您可以將 CodeCommit 存放庫、分支和資料夾路徑定義為設定此解決方案的一部分。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CDK 工具組是命令列 Cloud Development Kit，可協助您與 AWS 雲端開發套件 (AWS CDK) 應用程式互動。
- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。此模式使用 IAM Access Analyzer (IAM 的一項功能) 來分析您的 CloudTrail 日誌，以識別 IAM 實體 (使用者或角色) 已使用的動作和服務，然後根據該活動產生 IAM 政策。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。在此模式中，您可以使用 Step Functions 中的 AWS SDK 服務整合，從工作流程呼叫服務 API 動作。

其他工具

- CDK 對於地形表單（CDKTF）可幫助您通過使用常見的編程語言（如 Python 和打字稿）將基礎設施定義為代碼（IaC）。
- Lerna 是一個構建系統，用於管理和從同一個存儲庫發布多個 JavaScript 或 TypeScript 包。
- Node.js 是一個事件驅動的 JavaScript 運行時環境，旨在構建可擴展的網絡應用程序。
- npm 是一個在 Node.js 環境中運行的軟件註冊表，用於共享或借用軟件包以及管理私有軟件包的部署。

代碼存儲庫

GitHub 自動化 IAM 存取分析器角色政策產生器存放庫中提供此模式的程式碼。

史诗

準備部署

任務	描述	所需技能
克隆回購。	下列命令會複製自動 IAM 存取分析角色政策產生器 (GitHub) 存放庫。 `git clone https://github.com/aws-samples/automated-iam-access-analyzer.git`	應用程式開發人員
安裝萊納。	下面的命令安裝萊納。 `npm i -g lerna`	應用程式開發人員
設置依賴關係。	以下命令安裝存儲庫的依賴關係。 `cd automated-iam-access-advisor/` `npm install && npm run bootstrap`	應用程式開發人員

任務	描述	所需技能
建置程式碼。	下列命令會測試、建置及準備 Lambda 函數的 zip 套件。 `npm run test:code` `npm run build:code` `npm run pack:code`	應用程式開發人員
建置建構。	下列命令會針對 AWS CDK 和 CDKTF 建立基礎設施合成應用程式。 `npm run build:infra`	
設定任何自訂權限。	在複製存放庫的 repo 資料夾中，編輯允許.json 和 deny.json 檔案，以定義角色的任何自訂權限。如果允許 .json 和拒絕的 .json 檔案包含相同的權限，則會套用拒絕權限。	AWS 管理員、應用程式開發

選項 1 — 使用 AWS CDK 部署解決方案

任務	描述	所需技能
部署 AWS CDK 堆疊。	以下命令透過 AWS CloudFormation 部署基礎設施。定義下列參數： • `<NAME_OF_ROLE>` — 您要為其建立新政策之 IAM 角色的 ARN。 • `<TRAIL_ARN>` — 儲存角色活動之 CloudTrail 追蹤的 ARN。	應用程式開發人員

任務	描述	所需技能
	• <CRON_EXPRESSION_TO_RUN_SOLUTION> — 定義策略再生排程的 Cron 運算式。「Step Functions」工作流程會依此排程執行。 • <TRAIL_LOOKBACK> — 評估角色權限時回溯追蹤的期間 (以天為單位)。 `cd infra/cdk` `cdk deploy –-parameters roleArn=<NAME_OF_ROLE> \` `–-parameters trailArn=<TRAIL_ARN> \` `--parameters schedule=<CRON_EXPRESSION_TO_RUN_SOLUTION> \` `[ --parameters trailLookBack=<TRAIL_LOOKBACK> ]` 注意 — 方括號代表選擇性參數。	
(選擇性) 等待新原則。	如果追蹤不包含合理數量的角色歷史活動，請等到您確信 IAM Access Analyzer 有足夠的記錄活動，才能產生正確的政策。如果該角色在帳戶中已啟用一段足夠的時間，則可能不需要此等待期。	AWS 管理員

任務	描述	所需技能
手動檢閱產生的策略。	在 CodeCommit 存放庫中，檢閱產生的 .json <ROLE_ARN> 檔案，以確認允許和拒絕權限適用於該角色。	AWS 管理員

選項 2 — 使用 CDKTF 部署解決方案

任務	描述	所需技能
合成地形範本。	下列指令會合成地形範本。 `lerna exec cdktf synth --scope @aiaa/tfm`	應用程式開發人員
部署地形範本。	下列命令會導覽至包含 CDKTF 定義基礎結構的目錄。 `cd infra/cdktf` 下列命令會在目標 AWS 帳戶中部署基礎設施。定義下列參數： • <account_ID> — 目標帳戶的識別碼。 • <region>-目標 AWS 區域。 • <selected_role_ARN> — 您要為其建立新政策之 IAM 角色的 ARN。 • <trail_ARN> — 儲存角色活動之 CloudTrail 追蹤的 ARN。	應用程式開發人員

任務	描述	所需技能
	• <schedule_expression> — 定義策略再生排程的 Cron 運算式。「Step Functions」工作流程會依此排程執行。 • <trail_look_back> — 評估角色權限時回溯追蹤的期間 (以天為單位)。 `TF_VAR_accountId=<account_ID> \ TF_VAR_region=<region> \ TF_VAR_roleArns=<selected_role_ARN> \ TF_VAR_trailArn=<trail_ARN> \ TF_VAR_schedule=<schedule_expression> \ [ TF_VAR_trailLookBack=<trail_look_back> ] \ cdktf deploy` 注意 — 方括號代表選擇性參數。	
(選擇性) 等待新原則。	如果追蹤不包含合理數量的角色歷史活動，請等到您確信 IAM Access Analyzer 有足夠的記錄活動，才能產生正確的政策。如果該角色在帳戶中已啟用一段足夠的時間，則可能不需要此等待期。	AWS 管理員

任務	描述	所需技能
手動檢閱產生的策略。	在 CodeCommit 存放庫中，檢閱產生的 .json <ROLE_ARN> 檔案，以確認允許和拒絕權限適用於該角色。	AWS 管理員

相關資源

AWS 資源

- IAM 存取分析器端點和配額
- 設定 AWS CLI
- 開始使用 AWS CDK
- 最低權限

其他資源

- 地形版 CDK（地形網站）

使用 AWS 範本 GuardDuty 有條件地啟用 Amazon CloudFormation

由拉姆·康達斯瓦米（AWS）創建

環境：生產	技術：安全性、身分識別、合規性 DevOps；營運	AWS 服務：AWS CloudFormation；Amazon GuardDuty；AWS Lambda；AWS Identity and Access Management

Summary

您可以使用 AWS CloudFormation 範本 GuardDuty 在 Amazon 網路服務 (AWS) 帳戶上啟用亞馬遜。根據預設，如果 GuardDuty 在嘗試使用 CloudFormation 將其開啟時已啟用，堆疊部署會失敗。但是，您可以使用 CloudFormation 範本中的條件來檢查 GuardDuty 是否已啟用。 CloudFormation 支援使用比較靜態值的條件；它不支援在相同範本中使用其他資源屬性的輸出。如需詳細資訊，請參閱 CloudFormation 使用指南中的條件。

在此模式中， GuardDuty 如果尚未啟用，您可以使用 AWS Lambda 函數支援的 CloudFormation 自訂資源，有條件地啟用該資源。如果啟 GuardDuty 用，堆疊會擷取狀態並將其記錄在堆疊的輸出區段中。如果 GuardDuty 未啟用，堆疊會啟用它。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 具有建立、更新和刪除 CloudFormation 堆疊許可的 AWS Identity and Access Management (IAM) 角色

限制

- 如果 AWS 帳戶或區域 GuardDuty 已手動停用，則該目標帳戶或區域無法啟用 GuardDuty 此模式。

架構

目標技術堆疊

該模式用 CloudFormation 於基礎架構即代碼（IaC）。您可以使用由 Lambda 函數支援的 CloudFormation 自訂資源來實現動態服務啟用功能。

目標架構

下列高階架構圖顯示透過部署 CloudFormation 範本 GuardDuty 來啟用的程序：

1. 您部署 CloudFormation 範本以建立 CloudFormation 堆疊。
2. 堆疊會建立 IAM 角色和 Lambda 函數。
3. Lambda 函數會擔任 IAM 角色。
4. 如果 GuardDuty 目標 AWS 帳戶尚未啟用，則 Lambda 函數會啟用該功能。

自動化和規模

您可以使用 AWS CloudFormation StackSet 功能將此解決方案擴展到多個 AWS 帳戶和 AWS 區域。如需詳細資訊，請參閱使用 CloudFormation 者指南 CloudFormation StackSets中的使用 AWS。

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon GuardDuty 是一種持續的安全監控服務，可分析和處理日誌，以識別 AWS 環境中的未預期和潛在未經授權的活動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

史诗

建立 CloudFormation 範本並部署堆疊

任務	描述	所需技能
建立 CloudFormation 範本。	1. 在 [其他資訊] 區段中複製CloudFormation 範本中的程式碼。 2. 在文字編輯器中貼上程式碼。 3. 將檔案另存為`sample.yaml` 在工作站上。	AWS DevOps
建立 CloudFormation 堆疊。	1. 在 AWS CLI 中，輸入以下命令。這將使用該`sample.yaml` 文件創建一個新的 CloudFormation 堆棧。如需詳細資訊，請參閱 CloudFormation 使用指南中的「建立堆疊」。 `aws cloudformation create-stack \ --stack-name guardduty-cf-stack \ --template-body file://sample.yaml` 2. 確認 AWS CLI 中出現以下值，表示已成功建立堆疊。建立堆疊所需的時間可能會有所不同。 `"StackStatus": "CREATE_COMPLETE",`	AWS DevOps

任務	描述	所需技能
驗證已 GuardDuty 為 AWS 帳戶啟用。	1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/guardduty/ 開啟 GuardDuty 主控台。 2. 確認 GuardDuty 服務已啟用。	雲端管理員、AWS 管理員
設定其他帳戶或 AWS 區域。	根據您的使用案例需要，使用 AWS CloudFormation StackSet 功能將此解決方案擴展到多個 AWS 帳戶和 AWS 區域。如需詳細資訊，請參閱使用 CloudFormation 者指南 CloudFormation StackSets中的使用 AWS。	雲端管理員、AWS 管理員

相關資源

參考

- AWS CloudFormation 文件
- AWS Lambda 資源類型參考
- CloudFormation 資源類型： AWS::IAM::Role
- CloudFormation 資源類型： AWS::GuardDuty::Detector
- 使用 AWS 擷取任何 AWS 服務屬性的四種方式 CloudFormation (部落格)

教學課程和影片

- 使用 AWS 簡化基礎設施管理 CloudFormation (教學課程)
- 使用 Amazon GuardDuty 和 AWS Security Hub 保護多個帳戶 (AWS RE: 發明 2020)
- 建立 AWS 的最佳實務 CloudFormation (AWS RE: 發明 2019)
- AWS 上的威脅偵測：Amazon GuardDuty 簡介

其他資訊

CloudFormation 範本

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  rLambdaLogGroup:
    Type: 'AWS::Logs::LogGroup'
    DeletionPolicy: Delete
    Properties:
      RetentionInDays: 7
      LogGroupName: /aws/lambda/resource-checker
  rLambdaCheckerLambdaRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: !Sub 'resource-checker-lambda-role-${AWS::Region}'
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: 'sts:AssumeRole'
      Path: /
      Policies:
        - PolicyName: !Sub 'resource-checker-lambda-policy-${AWS::Region}'
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Sid: CreateLogGroup
                Effect: Allow
                Action:
                  - 'logs:CreateLogGroup'
                  - 'logs:CreateLogStream'
                  - 'logs:PutLogEvents'
                  - 'iam:CreateServiceLinkedRole'
                  - 'cloudformation:CreateStack'
                  - 'cloudformation:DeleteStack'
                  - 'cloudformation:Desc*'
                  - 'guardduty:CreateDetector'
                  - 'guardduty:ListDetectors'
                  - 'guardduty:DeleteDetector'
                Resource: '*'
```

```
  resourceCheckerLambda:
    Type: 'AWS::Lambda::Function'
    Properties:
      Description: Checks for resource type enabled and possibly name to exist
      FunctionName: resource-checker
      Handler: index.lambda_handler
      Role: !GetAtt
        - rLambdaCheckerLambdaRole
        - Arn
      Runtime: python3.8
      MemorySize: 128
      Timeout: 180
      Code:
        ZipFile: |
          import boto3
          import os
          import json
          from botocore.exceptions import ClientError
          import cfnresponse


          guardduty=boto3.client('guardduty')
          cfn=boto3.client('cloudformation')


          def lambda_handler(event, context):
              print('Event: ', event)
              if 'RequestType' in event:
                if event['RequestType'] in ["Create","Update"]:
                    enabled=False
                    try:
                      response=guardduty.list_detectors()
                      if "DetectorIds" in response and len(response["DetectorIds"])>0:
                        enabled="AlreadyEnabled"
                      elif "DetectorIds" in response and
len(response["DetectorIds"])==0:
                        cfn_response=cfn.create_stack(
                          StackName='guardduty-cfn-stack',
                          TemplateBody='{ "AWSTemplateFormatVersion": "2010-09-09",
"Description": "A sample template",    "Resources": { "IRWorkshopGuardDutyDetector": {
 "Type": "AWS::GuardDuty::Detector",    "Properties": {   "Enable": true  }   } } }'
                          )
                        enabled="True"
                    except Exception as e:
```

```
                        print("Exception: ",e)
                    responseData = {}
                    responseData['status'] = enabled
                    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
 "CustomResourcePhysicalID" )
                elif event['RequestType'] == "Delete":
                    cfn_response=cfn.delete_stack(
                            StackName='guardduty-cfn-stack')
                    cfnresponse.send(event, context, cfnresponse.SUCCESS, {})
  CheckResourceExist:
    Type: 'Custom::LambdaCustomResource'
    Properties:
      ServiceToken: !GetAtt
        - resourceCheckerLambda
        - Arn
Outputs:
  status:
    Value: !GetAtt
      - CheckResourceExist
      - status
```

Lambda 資源的替代程式碼選項

提供的 CloudFormation 範本使用內嵌程式碼來參考 Lambda 資源，以便於參考和指引。或者，您可以將 Lambda 程式碼放置在亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體中，並在 CloudFormation 範本中參考它。內嵌程式碼不支援套件相依性或程式庫。您可以將 Lambda 程式碼放在 S3 儲存貯體中，並在 CloudFormation 範本中參考它來支援這些功能。

取代下列程式碼行：

```
Code:
         ZipFile: |
```

使用以下幾行代碼：

```
Code:
      S3Bucket: <bucket name>
      S3Key: <python file name>
      S3ObjectVersion: <version>
```

如果您未在 S3 儲存貯體中使用版本控制，則可以省略該`S3ObjectVersion`屬性。如需詳細資訊，請參閱 Amazon S3 使用者指南中的在 S3 儲存貯體中使用版本控制。

在 Amazon RDS for SQL Server 中啟用透明資料加密

由蘭加凱魯庫裡（AWS）創建

環境：PoC 或試點	技術：安全性、身分識別、合規性；資料庫	工作量：Microsoft
AWS 服務：Amazon RDS		

Summary

此模式說明如何在適用於 SQL 伺服器的 Amazon 關聯式資料庫服務 (Amazon RDS) 中實作透明資料加密 (TDE)，以加密靜態資料。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amazon RDS for SQL Server 資料庫執行個體

產品版本

Amazon RDS 目前支援以下 SQL 伺服器版本和版本的 TDE：

- SQL Server 2012 Enterprise Edition
- SQL Server 2014 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2019 Standard 和 Enterprise Editions

如需有關 Support 援版本和版本的最新資訊，請參閱 Amazon RDS 說明文件中的 SQL Server 中的透明資料加密支援。

架構

技術, 堆

- Amazon RDS for SQL Server

架構

工具

工具

- Microsoft SQL 服務器管理工作室（SSMS）是用於管理 SQL 服務器基礎設施的集成環境。它提供了一個用戶界面和一組具有與 SQL Server 交互的豐富腳本編輯器的工具。

史诗

在 Amazon RDS 主控台中建立選項群組

任務	描述	所需技能
開啟 Amazon RDS 主控台。	登入 AWS 管理主控台並開啟 Amazon RDS 主控台。	開發人員, DBA
建立選項群組。	在功能窗格中，選擇 [選項群組] > [建立群組]。選擇 sqlserver ee 作為資料庫引擎，然後選取引擎版本。	開發人員, DBA
新增透明資料加密選項。	編輯您建立的選項群組，並新增名為的選項`TRANSPARENT_DATA_ENCRYPTION`。	開發人員, DBA

將選項群組與資料庫執行個體關聯

任務	描述	所需技能
選擇資料庫執行個體。	在 Amazon RDS 主控台的導覽窗格中，選擇 [資料庫]，然後選擇要與選項群組建立關聯的資料庫執行個體。	開發人員, DBA
將資料庫執行個體與選項群組建立關聯。	選擇 [修改]，然後使用 [選項] 群組設定，將 SQL Server 資料庫執行個體與您先前建立的選項群組產生關聯。	開發人員, DBA
套用變更。	視需要立即套用變更，或在下一個維護時段套用變更。	開發人員, DBA
取得憑證名稱。	使用下列查詢取得預設憑證名稱。 `USE [master]` `GO` `SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'` `GO`	開發人員, DBA

建立資料庫加密金鑰

任務	描述	所需技能
使用 SSMS Connect 到 Amazon RDS for SQL Server 的資料庫執行個體。	如需指示，請參閱 Microsoft 文件中的使用 SSMS。	開發人員, DBA
使用預設憑證建立資料庫加密金鑰。	使用先前取得的預設憑證名稱來建立資料庫加密金鑰。使用	開發人員, DBA

任務	描述	所需技能
	下列 T-SQL 查詢建立資料庫加密金鑰。您可以指定 AES_256 演算法，而非 AES_128。 `USE [Databasename]` `GO` `CREATE DATABASE ENCRYPTION KEY` `WITH ALGORITHM = AES_128` `ENCRYPTION BY SERVER CERTIFICATE [certificatename]` `GO`	
啟用資料庫的加密。	使用下列 T-SQL 查詢來啟用資料庫加密。 `ALTER DATABASE [Database Name]` `SET ENCRYPTION ON` `GO`	開發人員, DBA
檢查加密狀態。	使用下列 T-SQL 查詢來檢查加密狀態。 `SELECT DB_NAME(database_id) AS DatabaseName, encryption_state, percent_complete FROM sys.dm_database_encryption_keys`	開發人員, DBA

相關資源

- SQL 伺服器中的透明資料加密 Support 援 (Amazon RDS 文件)

- 使用選項群組 (Amazon RDS 文件)
- 修改 Amazon RDS 資料庫執行個體 (Amazon RDS 文件)
- SQL 伺服器的透明資料加密 (Microsoft 文件)
- 使用 SSMS (Microsoft 文檔)

確保 AWS CloudFormation 堆疊是從授權的 S3 儲存貯體啟動

環境：生產	技術：安全性、身分識別、合規	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS CloudFormation；Amazon CloudWatch；AWS Lambda；Amazon S3		

Summary

您可以使用 AWS CloudFormation 範本以程式設計方式設定 Amazon Web Services (AWS) 資源，以減少管理這些資源的時間，將更多時間專注於在 AWS 中執行的應用程式。此模式可讓您檢查 AWS CloudFormation 堆疊是否僅從存放在特定 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體中的範本建立。如果您有安全性或合規要求，而且要求使用存放在允許清單中 S3 儲存貯體中的範本，則此檢查非常有用。

此安全控制可監控 AWS CloudFormation CreateStack和 UpdateStackAPI 呼叫，並叫用 AWS Lambda 函數，以檢查呼叫中使用的範本是否來自授權的 S3 儲存貯體。如果範本來自未授權的儲存貯體，Lambda 函數會觸發 Amazon Simple Notification Service (Amazon SNS) 電子郵件通知給使用者，其中包含相關資訊。

先決條件和限制

先決條件

- 您希望接收違規通知的作用中電子郵件地址
- 用於上傳提供的 Lambda 程式碼的 S3 儲存貯體
- 授權的 S3 儲存貯體名稱清單

限制

- UpdateStack在未經授權的 S3 儲存貯體中使用現有範本的 API 呼叫不會產生其他違規，因為 S3 儲存貯體的 URL 在 Amazon EventBridge 事件中無法使用。建議您在收到原始CreateStack違規通知後，刪除未經授權的 S3 儲存貯體中的現有範本。

- 此安全控制不會監控下列 AWS CloudFormation 事件，因為它們會在範本初始部署後處理更新：CreateChange設定、CreateStack設定、UpdateStack設定。
- 您必須在要監控的每個 AWS 區域中部署此安全控制。

架構

目標技術堆疊

- AWS Lambda
- Amazon SNS
- Amazon EventBridge 法則

目標架構

自動化和規模

如果您使用 AWS Organizations，則可以使用 AWS CloudFormation StackSets 在要監控的多個帳戶中部署此範本。

工具

- AWS 雲形 — 協助您使用模型建立 AWS 資源的 infrastructure-as-code 模型和設定。
- Amazon EventBridge — 從您自己的應用程式、 software-as-a-service (SaaS) 應用程式和 AWS 服務交付即時資料串流，並將資料路由到 AWS Lambda 等目標。
- AWS Lambda — 讓您無需佈建或管理伺服器即可執行程式碼。
- Amazon SNS — 提供從發佈者傳遞給訂閱者的訊息。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。
- Amazon S3 — 可讓您隨時從 Web 上的任何位置存放和擷取任意數量的資料。

史诗

部署安全控制

任務	描述	所需技能
將 Lambda 程式碼上傳至 Amazon S3。	將包含「附件」部分中提供的 Lambda 程式碼的 .zip 檔案上傳到新的或現有的 S3 儲存貯體。此儲存貯體應與您要評估的資源位於相同的 AWS 區域。	雲端架構師
部署 AWS CloudFormation 範本。	在與 S3 儲存貯體相同的區域中開啟 AWS CloudFormation 主控台，然後部署「附件」部分中提供的範本。提供參數的值；這些參數會在「其他資訊」一節中說明。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱 Amazon SNS 主題。	成功部署 AWS CloudFormation 範本後，會將訂閱電子郵件傳送到您提供的電子郵件地址。您必須確認此電子郵件訂閱才能開始接收通知。	雲端架構師

相關資源

- 部署 AWS CloudFormation 範本
- Amazon EventBridge
- AWS Lambda

- Amazon Simple Storage Service (Amazon S3)

其他資訊

當您部署使用此 CloudFormation 模式提供的 AWS 範本時，系統會提示您輸入下列資訊：

- S3 儲存貯體：指定您上傳附加的 Lambda 程式碼 (.zip 檔案) 的儲存貯體。您可以建立新值區或指定現有值區。
- S3 金鑰：指定 S3 儲存貯體中 Lambda .zip 檔案的位置 (例如：檔案名稱 .zip 或控制項/檔案名稱 .zip)。請勿使用前導斜線。
- 通知電子郵件：提供應傳送違規通知的作用中電子郵件地址。
- Lambda 記錄層級：指定 Lambda 函數的記錄層級。使用「資訊」(Info) 可記錄進度的詳細資訊訊息、針對仍允許部署繼續的錯誤事件發生錯誤，以及針對潛在有害情況發出警告。
- 授權值區：提供以逗號分隔的授權 S3 儲存貯體清單。

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確保 AWS 負載平衡器使用安全接聽程式協定 (HTTPS、SSL/TLS)

由錢迪尼潘梅薩（AWS）和普魯斯霍瑟姆 G K（AWS）創建

環境：生產	技術：安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS；Amazon CloudFormation；AWS Lambda CloudWatch；Elastic Load Balancing (ELB)		

Summary

在 Amazon Web Services (AWS) 雲端上，Elastic Load Balancing 會自動將傳入的應用程式流量分配到多個目標，例如 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、容器、IP 地址和 AWS Lambda 函數。負載平衡器使用接聽程式來定義負載平衡器用來接受使用者流量的連接埠和通訊協定。應用程式負載平衡器會在應用程式層做出路由決策，並使用 HTTP/HTTPS 通訊協定。網路負載平衡器會在傳輸層做出路由決策，並使用傳輸控制通訊協定 (TCP)、傳輸層安全性 (TLS)、使用者資料包 Protcol (UDP) 或 TCP_UDP 通訊協定。傳統負載平衡器使用 TCP 或安全通訊端層 (SSL) 通訊協定，或在應用程式層使用 HTTP/HTTPS，在傳輸層做出路由決策。

您的組織可能有安全性或合規性需求，負載平衡器只接受安全通訊協定 (例如 HTTPS 或 SSL/TLS) 上的使用者的流量。

此模式提供安全控制，使用 Amazon EventBridge 規則監控應用程式負載平衡器`CreateListener`和網路負載平衡器的和 `ModifyListener` API 呼叫，以及傳統負載平衡器的`CreateLoadBalancerListeners`和 `CreateLoadBalancer` API 呼叫。如果負載平衡器的接聽程式通訊協定使用 HTTP、TCP/UDP 或 TCP_UDP，則控制項會叫用 Lambda 函數。Lambda 函數會將訊息發佈至 Amazon Simple Notification Service (Amazon SNS) 主題，以傳送包含負載平衡器詳細資訊的通知。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 您想要接收違規通知的電子郵件地址
- 用於存放 Lambda 程式碼 .zip 檔案的 Amazon Simple Storage Service (Amazon S3) 儲存貯體

限制

- 除非對負載平衡器接聽程式進行更新，否則此安全控制不會檢查現有的負載平衡器。
- 此安全控制是區域性的，必須部署在您想要監控的 AWS 區域中。

架構

目標技術堆疊

- Lambda 函數
- Amazon SNS 主題
- EventBridge 規則

目標架構

自動化和規模

- 如果您使用的是 AWS Organizations，則可以使用 AWS CloudFormation StackSets 將此範本部署到您希望監控的多個帳戶中。

工具

- AWS CloudFormation — AWS CloudFormation 是一項服務，可協助您使用基礎設施即程式碼來建模和設定 AWS 資源。
- Amazon EventBridge — Amazon EventBridge 提供來自您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務的即時資料串流，並將該資料路由到 Lambda 函數等目標。
- AWS Lambda — L ambda 支援執行程式碼，無需佈建或管理伺服器。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。

- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

最佳實務

確定所使用的 SNS 主題無法公開存取。如需詳細資訊，請參閱 AWS 文件。

史诗

上傳程 Lambda 碼

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。您的 S3 儲存貯體必須與正在評估的負載平衡器位於相同的區域。	雲端架構師
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」一節中提供的 Lambda 程式碼 .zip 檔案上傳到定義的 S3 儲存貯體。	雲端架構師
部署 AWS CloudFormation 範本。	在 AWS CloudFormation 主控台與 S3 儲存貯體相同的 AWS 區域中，部署「附件」部分中提供的範本。在下一個史詩中，提供參數的值。	雲端架構師

CloudFormation 參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 Amazon S3 前綴。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<directory>/<file-name>.zip`	雲端架構師
提供 SNS 主題 ARN。	如果您想要使用現有的 SNS 主題進行違規通知，請提供 SNS 主題 Amazon 資源名稱 (ARN)。若要建立新的 SNS 主題，請將值保持為 None (預設值)。	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
下載 範本。	下 CloudFormation 載「附件」區段中提供的範本。	雲端架構師

任務	描述	所需技能
建立堆疊。	在與 S3 儲存貯體相同的區域中，導覽至 CloudFormation 服務主控台，然後部署下載的範本。有關參數詳細信息，請參閱上一個史詩。	雲端架構師
驗證資源。	完全建立堆疊後，瀏覽至 [資源] 索引標籤，並驗證資源。該模板將創建以下資源： • EventBridge 規則 • Lambda 函數 • Lambda 執行角色 • Lambda 調用權限	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署範本時，如果建立了新的 SNS 主題，則會將訂閱電子郵件訊息傳送至參數中提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

故障診斷

問題	解決方案
堆疊建立失敗。發生錯誤時 GetObject。S3 錯誤代碼： PermanentRedirect。S3 錯誤訊息：	確保 S3 儲存貯體區域和部署堆疊的區域相同。

問題	解決方案
儲存貯體位於此區域：xx-xxxx-1。請使用此區域重試要求。	
堆疊建立失敗。建立或更新 AWS Lambda 函數不再支援 python3.6 的執行階段參數。	將下載的模板從第 186 行 3.6 更新為 3.9。

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊
- AWS Lambda
- 什麼是 Classic Load Balancer？
- 什麼是 Application Load Balancer？
- 什麼是 Network Load Balancer？
- 使用 AWS Lambda 函數的最佳實務
- AWS CloudFormation 最佳實務

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確保啟動時已啟用 Amazon EMR 靜態資料的加密

創建者：普里揚卡喬達瑞 (AWS)

環境：生產	技術：安全性、身分識別、合規性；分析	工作負載：開源
AWS 服務：Amazon EMR；Amazon SNS；AWS KMS；AWS CloudFormation；AWS Lambda；Amazon S3		

Summary

此模式提供安全控制，用於在亞馬遜網路服務 (AWS) 上監控 Amazon EMR 叢集的加密。

資料加密有助於防止未經授權的使用者讀取叢集上的資料和相關的資料儲存體系統。這包括傳輸網路時可能遭到攔截的資料 (稱為傳輸中的資料)，以及儲存至持續性媒體 (稱為靜態資料) 的資料。Amazon Simple Storage Service (Amazon S3) 中的靜態資料可以透過兩種方式加密。

- 使用 Amazon S3 受管金鑰進行伺服器端加密 (SSE-S3)
- 使用 AWS Key Management Service (AWS KMS) 金鑰 (SSE-KMS) 進行伺服器端加密，並使用適用於 Amazon EMR 的政策進行設定。

此安全控制會監控 API 呼叫，並在上啟 CloudWatch 動 Amazon 事件事件RunJobFlow。觸發程序會叫用 AWS Lambda，它會執行 Python 指令碼。函數會從事件 JSON 輸入擷取 EMR 叢集 ID，並透過執行下列檢查來判斷是否存在安全性違規。

1. 檢查 EMR 叢集是否與 Amazon EMR 特定安全組態相關聯。
2. 如果 Amazon EMR 特定安全組態與 EMR 叢集相關聯，請檢查靜態加密是否已開啟。
3. 如果未開啟靜態加密，請傳送 Amazon Simple Notification Service (Amazon SNS) 通知，其中包含此通知來源的 EMR 叢集名稱、違規詳細資訊、AWS 區域、AWS 帳戶以及 Lambda Amazon 資源名稱 (ARN)。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 用於 Lambda 代碼 .zip 文件的 S3 存儲桶
- 您想要接收違規通知的電子郵件地址
- Amazon EMR 記錄已關閉，以便擷取所有 API 日誌

限制

- 此偵探控制項是區域性的，必須部署在您想要監控的 AWS 區域中。

產品版本

- Amazon EMR 版本 4.8.0 及更高版本

架構

目標技術堆疊

- Amazon EMR
- Amazon CloudWatch 活動
- Lambda 函數
- Amazon SNS

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudform StackSets 在您要監控的多個帳戶中部署此範本。

工具

工具

- AWS CloudFormation — AWS CloudFormation 是一項服務，可協助您使用基礎設施即程式碼建立 AWS 資源模型和設定 AWS 資源。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- Amazon EMR — Amazon EMR 是一個受管叢集平台，可簡化大數據架構的執行作業。
- AWS Lambda — AWS Lambda 支援執行程式碼，無需佈建或管理伺服器。
- Amazon S3 — Amazon S3 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon SNS 協調和管理發佈者和用戶端之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

- 該項目的 EMR EncryptionAtRest .zip 和 EMR EncryptionAtRest .yml 文件可作為附件使用。

史诗

定義 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。您的 S3 儲存貯體必須與正在評估的 Amazon EMR 叢集位於相同的區域。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將「附件」一節中提供的 Lambda 程式碼 .zip 檔案上傳到定義的 S3 儲存貯體。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
部署 AWS CloudFormation 範本。	在 AWS CloudFormation 主控台與 S3 儲存貯體相同的區域中，將以附件形式提供的 AWS CloudFormation 範本部署到此模式。在下一個史詩中，提供參數的值。如需部署 AWS CloudFormation 範本的詳細資訊，請參閱「相關資源」一節。	雲端架構師

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 Amazon S3 密鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如<directory>/<file-name>.zip)。	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師

任務	描述	所需技能
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。「信息」指定了有關應用程序進度的詳細信息消息。「Error」指定仍可允許應用程式繼續執行的錯誤事件。「警告」表示可能有害的情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件訊息傳送至提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊
- AWS Lambda
- Amazon EMR 加密選項

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確保 IAM 設定檔與 EC2 執行個體相關聯

創建者：曼西蘇拉特瓦拉（AWS）

環境：生產	技術：基礎架構；安全性、身分識別、合規	AWS 服務：Amazon EC2；AWS Identity and Access Management；Amazon CloudWatch；AWS Lambda；Amazon SNS

Summary

此模式提供 AWS CloudFormation 安全控制範本，可在 Amazon 彈性運算雲端 (Amazon EC2) 執行個體發生 AWS Identity and Access Management (IAM) 設定檔違規時設定自動通知。

執行個體設定檔是 IAM 角色的容器，您可以在執行個體啟動時將角色資訊傳遞給 EC2 執行個體。

當 AWS 根據`RunInstances`、`AssociateIamInstanceProfile`和動作 CloudTrail 記錄 Amazon EC2 API 呼叫時，亞馬遜 CloudWatch 事件會啟`ReplaceIamInstanceProfileAssociation`動此檢查。觸發器會呼叫 AWS Lambda 函數，該函數使用 Amazon CloudWatch 事件來檢查 IAM 設定檔。

如果 IAM 設定檔不存在，Lambda 函數會啟動 Amazon Simple Notification Service (Amazon SNS) 電子郵件通知，其中包含 Amazon Web Services (AWS) 帳戶 ID 和 AWS 區域。

如果 IAM 設定檔確實存在，Lambda 函數會檢查政策文件中是否有任何萬用字元項目。如果萬用字元項目存在，則會啟動 Amazon SNS 違規通知，以協助您實作增強的安全性。通知包含 IAM 設定檔的名稱、事件、EC2 執行個體 ID、受管政策的名稱、違規、帳戶 ID 和區域。

先決條件和限制

先決條件

- 活躍帳戶
- 適用於 Lambda 程式碼 .zip 檔案的 Amazon Simple Storage Service (Amazon S3) 儲存貯體

限制

- 只能針對`RunInstances`、`AssociateIamInstanceProfile`和`ReplaceIamInstanceProfileAssociatio`作部署 AWS CloudFormation 範本。
- 安全控制不會監控 IAM 設定檔的分離。
- 安全控制不會檢查附加至 EC2 執行個體 IAM 設定檔的 IAM 政策是否有修改。
- 安全性控制不會考慮需要使用的不支援資源層級權限。`"Resource":*`

架構

目標技術堆疊

- Amazon EC2
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon S3
- Amazon SNS

目標架構

自動化和規模

您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需為每個帳戶或區域啟動一次範本。

工具

工具

- Amazon EC2 — Amazon EC2 在 AWS 雲端提供可擴展的運算容量 (虛擬伺服器)。
- AWS CloudTrail — AWS 可 CloudTrail 協助您啟用 AWS 帳戶的管控、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。

- AWS Lambda — AWS Lambda 是一種運算服務，您可以使用它來執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — Amazon S3 提供可高度擴展的物件儲存，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon SNS 可讓應用程式和裝置從雲端傳送和接收通知。

Code

- 專案的 .zip 檔案可作為附件使用。

史诗

定義 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	若要託管 Lambda 程式碼 .zip 檔案，請選擇或建立具有不含前導斜線的唯一名稱的 S3 儲存貯體。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用命名空間。您的 S3 儲存貯體必須與正在評估的 EC2 執行個體位於相同的區域。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將附件區段中提供的 Lambda 程式碼上傳到 S3 儲存貯體。S3 儲存貯體必須與要評估的 EC2 執行個體位於相同的區域。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
部署 AWS CloudFormation 範本。	將以附件形式提供的 AWS CloudFormation 範本部署到此模式。在下一個史詩中，提供參數的值。	雲端架構師

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<directory>/<file-name>.zip`	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件訊息傳送至提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

- 建立 S3 儲存貯體
- 將檔案上傳到 S3 儲存貯體
- 使用例項設定檔
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確保亞 Amazon Redshift 叢集在建立時已加密

創建者：曼西蘇拉特瓦拉（AWS）

環境：生產	技術：分析；資料湖；安全性、身分識別、合規	工作負載：所有其他工作
AWS 服務：Amazon Redshift；Amazon SNS；AWS；Amazon CloudTrail CloudWatch；AWS Lambda；Amazon S3		

Summary

此模式提供 AWS CloudFormation 範本，當建立新的 Amazon Redshift 叢集時不加密，可為您提供自動通知。

AWS CloudFormation 範本會建立 Amazon CloudWatch 活動事件和 AWS Lambda 函數。此事件會監視任何透過 AWS CloudTrail 從快照建立或還原的 Amazon Redshift 叢集。如果在 AWS 帳戶中建立叢集時未使用 AWS Key Management Service (AWS KMS) 或雲端硬體安全模型 (HSM) 加密，請 CloudWatch 啟動 Lambda 函數，以傳送 Amazon Simple Notification Service (Amazon SNS) 通知給您，通知您違規事件。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 具有叢集子網路群組和關聯安全性群組的虛擬私人雲端 (VPC)。

限制

- AWS CloudFormation 範本只能針對`CreateCluster`和動`RestoreFromClusterSnapshot`作部署。

架構

目標技術堆疊

- Amazon Redshift
- AWS CloudTrail
- Amazon CloudWatch
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

目標架構

自動化和規模

您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需在每個區域或帳戶中執行一次。

工具

工具

- Amazon Redshift — Amazon Redshift 是雲端中的全受管 PB 級資料倉儲服務。Amazon Redshift 與您的資料湖整合，可讓您使用資料為企業和客戶取得新的見解。
- AWS CloudTrail — AWS CloudTrail 是一項 AWS 服務，可協助您實作 AWS 帳戶的管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS Lambda — AWS Lambda 支援執行程式碼，無需佈建或管理伺服器。AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon S3 — Amazon S3 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon SNS 是一種 Web 服務，可協調和管理訊息傳遞或傳送給發佈者和用戶端之間的訊息，包括 Web 伺服器和電子郵件地址。

Code

- 專案的 .zip 檔案可作為附件使用。

史诗

定義 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立 S3 儲存貯體。此 S3 儲存貯體將託管 Lambda 程式碼 .zip 檔案。您的 S3 儲存貯體必須與正在評估的 Amazon Redshift 叢集位於相同的區域。S3 儲存貯體的名稱不能包含前導斜線。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
將 Lambda 程式碼上傳至 S3 儲存貯體。	將附件區段中提供的 Lambda 程式碼上傳到 S3 儲存貯體。S3 儲存貯體必須與正在評估的 Amazon 紅移叢集位於相同的區域。	雲端架構師

部署 AWS CloudFormation 範本

任務	描述	所需技能
部署 AWS CloudFormation 範本。	將以附件形式提供的 AWS CloudFormation 範本部署到此	雲端架構師

任務	描述	所需技能
	模式。在下一個史詩中，提供參數的值。	

完成 AWS CloudFormation 範本中的參數

任務	描述	所需技能
命名 S3 儲存貯體。	輸入您在第一個史詩中建立的 S3 儲存貯體的名稱。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<directory>/<file-name>.zip`	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件傳送至提供的電子郵件地址。您必須確認此電子郵件訂閱才能接收違規通知。	雲端架構師

相關資源

- 建立 S3 儲存貯體
- 將檔案上傳到 S3 儲存貯體
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail
- 創建一個 Amazon Redshift 集群

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用以下方式匯出 AWS IAM 身分中心身分及其指派的報告 PowerShell

由豪爾赫·帕瓦（AWS），乍得英里（AWS），弗蘭克·阿洛塔（AWS）和曼尼德普·雷迪吉勒拉（AWS）創建

環境：生產	技術：安全性、身分識別、合規性；管理與治理	工作量：Microsoft
AWS 服務：IAM 身分中心；適用於的 AWS 工具 PowerShell		

Summary

當您使用 AWS IAM 身分中心 (AWS Single Sign-On 的後續任務) 集中管理對所有 Amazon Web Services (AWS) 帳戶和雲端應用程式的單一登入 (SSO) 存取時，透過 AWS 管理主控台報告和稽核這些指派可能非常繁瑣且耗時。如果您要報告跨數十個或數百個 AWS 帳戶的使用者或群組的許可，則尤其如此。

對於許多人來說，查看此信息的理想工具將在電子表格應用程序中，例如 Microsoft Excel。這可協助您篩選、搜尋並視覺化整個組織的資料，並由 AWS Organizations 管理。

此模式說明如何使用 AWS 工 PowerShell 具在 IAM 身分中心產生 SSO 身分組態報告。報告會格式化為 CSV 檔案，其中包括身分識別名稱 (主體)、身分識別類型 (使用者或群組)、身分識別可存取的帳戶以及權限集。產生此報告後，您可以在偏好的應用程式中開啟報告，以依需要搜尋、篩選和稽核資料。下圖展示了試算表應用程式中的範例資料。

重要事項：由於此報告包含敏感資訊，因此我們強烈建議您將其安全地儲存並僅在 need-to-know 基礎上共用。

先決條件和限制

先決條件

- 已設定和啟用的 IAM 身分中心和 AWS Organizations。
- PowerShell，已安裝並配置。如需詳細資訊，請參閱安裝 PowerShell (Microsoft 說明文件)。
- 適用於 PowerShell、安裝和設定的 AWS 工具。基於效能原因，我們強烈建議您安裝 AWS Tools 的模組化版本 PowerShell，稱為. `AWS.Tools` 每個 AWS 服務都有自己的個別小模組支援。在命令 PowerShell 介面中，輸入下列指令來安裝此模式所需的模組：`AWS.Tools.Installer`、`Organizations`、`SSOAdmin`和`IdentityStore`。

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityStore
```

  如需詳細資訊，請參閱在 Windows 上安裝 AWS 工具或在 Linux 或 macOS 上安裝 AWS 工具 (適用於說明文件的 AWS 工具)。 PowerShell 如果您在安裝模組時收到錯誤訊息，請參閱此模式的疑難排解一節。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 或 AWS 開發套件必須先使用工作登入資料進行設定，方法是執行下列其中一項操作：
  - 使用 AWS CLI `aws configure` 如需詳細資訊，請參閱快速組態 (AWS CLI 文件)。
  - 設定 AWS CLI 或 AWS Cloud Development Kit (AWS CDK)，以透過 AWS Identity and Access Management (IAM) 角色取得臨時存取權限。如需詳細資訊，請參閱取得 CLI 存取的 IAM 角色登入資料 (IAM 身分中心說明文件)。
- AWS CLI 的具名設定檔，已儲存 IAM 主體的登入資料，可：
  - 可以存取 AWS Organizations 管理帳戶或 IAM 身分中心的委派管理員帳戶
  - 已套用`AWSSSOReadOnly`和 `AWSSSODirectoryReadOnly` AWS 受管政策

  如需詳細資訊，請參閱使用具名設定檔 (AWS CLI 文件) 和 AWS 受管政策 (IAM 文件)。

限制

- 目標 AWS 帳戶必須以 AWS Organizations 中的組織形式進行管理。

產品版本

- 對於所有作業系統，建議您使用 7.0 或更新PowerShell 版本。

架構

目標架構

1. 使用者在命令列中執行 PowerShell 指令碼。
2. 指令碼會假設 AWS CLI 的具名設定檔。這樣就可以存取 IAM 身分中心。
3. 指令碼會從 IAM 身分中心擷取 SSO 身分組態。
4. 指令碼會在儲存指令碼的本機工作站上的相同目錄中產生 CSV 檔案。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS IAM 身分中心可協助您集中管理對所有 AWS 帳戶和雲端應用程式的單一登入 (SSO) 存取。
- 的 AWS 工具 PowerShell是一 PowerShell 組模組，可協助您從命令列對 AWS 資源執行操作 PowerShell 指令碼。

其他工具

- PowerShell是一個 Microsoft 的自動化和配置管理程序，可以在 Windows，Linux 和 macOS 上運行。

史诗

產生報告

任務	描述	所需技能
準備腳本。	1. 複製此模 PowerShell 式的其他資訊區段中的指令碼。	雲端管理員

任務	描述	所需技能
	2. 在Param本節中，針對您的 AWS 環境，定義下列變數的值： • OutputFile — 報告的檔案名稱。 • ProfileName — AWS CLI 命名您要用來產生報告的設定檔。 • Region— 部署 IAM 身分中心的 AWS 區域。如需區域及其代碼的完整清單，請參閱區域端點。 3. 使用檔案名稱儲存腳本SSO-Report.ps1 。	
執行指令碼。	建議您使用下列命令在 PowerShell shell 中執行自訂指令碼。 `.\SSO-Report.ps1` 或者，您也可以輸入下列命令，從另一個 shell 執行指令碼。 `pwsh .\SSO-Report.ps1` 指令碼會在與指令碼檔案相同的目錄中產生 CSV 檔案。	雲端管理員

任務	描述	所需技能
分析報表資料。	輸出 CSV 檔案包含標頭AccountNamePermissionSet、主參與者和類型。在偏好的試算表應用程式中開啟此檔案。您可以建立資料表來篩選和排序輸出。	雲端管理員

故障診斷

問題	解決方案
`The term 'Get-<parameter>' is not recognized as the name of a cmdlet, function, script file, or operable program.` 錯誤	未安裝適用於 PowerShell 或其模組的 AWS 工具。在命令 PowerShell 介面中，輸入下列命令來安裝 AWS Tools 以 PowerShell 及此模式所需的模組：`AWS.Tools.Installer Organizations SSOAdmin`、、和`IdentityStore` 。 `Install-Module AWS.Tools.Installer` `Install-AWSToolsModule -Name Organizations, SSOAdmin, IdentityStore`
`No credentials specified or obtained from persisted/shell defaults` 錯誤	在「準備史詩」區段中的指令碼中，確認您已正確輸入`ProfileName` 和`Region`變數。請確定具名設定檔中的設定和登入資料具有足夠的權限來管理 IAM 身分中心。
`Authenticode Issuer …` 安裝 AWS 工具模組時發生錯誤	將`-SkipPublisherCheck` 參數加入至`Install-AWSToolsModule` 指令的結尾。
`Get-ORGAccountList : Assembly AWSSDK.SSO could not be found or loaded.` 錯誤	指定具名的 AWS CLI 設定檔、AWS CLI 設定為透過 IAM 身分中心驗證使用者，且 AWS CLI 設定為自動擷取重新整理的身份驗證權杖時，就

問題	解決方案
	會發生此錯誤。若要解決此錯誤，請執行下列動作： 1. 輸入下列命令以確認SSO和SSOOIDC模組已安裝。 `Install-AWSToolsModule SSO, SSOOIDC` 2. 在param()區塊下方的指令碼中插入以下幾行。 `Import-Module AWS.Tools.SSO` `Import-Module AWS.Tools.SSOOIDC`

相關資源

- 組態設定儲存在哪裡？ (AWS CLI 文件)
- 將 AWS CLI 設定為使用 AWS IAM 身分中心 (AWS CLI 文件)
- 使用具名設定檔 (AWS CLI 文件)

其他資訊

在下列指令碼中，判斷是否需要更新下列參數的值：

- 如果您使用 AWS CLI 中的具名設定檔來存取設定 IAM 身分中心的帳戶，請更新該$ProfileName值。
- 如果 IAM 身分中心部署在與 AWS CLI 或 AWS 開發套件組態的預設區域不同的 AWS 區域，請更新該$Region值以使用部署 IAM 身分中心的區域。
- 如果這些情況都不適用，則不需要更新指令碼。

```
param (
    # The name of the output CSV file
```

```
    [String] $OutputFile  = "SSO-Assignments.csv",
    # The AWS CLI named profile
    [String] $ProfileName = "",
    # The AWS Region in which IAM Identity Center is configured
    [String] $Region      = ""
)
$Start = Get-Date; $OrgParams = @{}
If ($Region){ $OrgParams.Region = $Region}
if ($ProfileName){$OrgParams.ProfileName = $ProfileName}
$SSOParams   = $OrgParams.Clone(); $IdsParams = $OrgParams.Clone()
$AccountList = Get-ORGAccountList @OrgParams | Select-Object Id, Name
$SSOinstance = Get-SSOADMNInstanceList @OrgParams
$SSOParams['InstanceArn']       = $SSOinstance.InstanceArn
$IdsParams['IdentityStoreId']   = $SSOinstance.IdentityStoreId
$PSsets       = @{}; $Principals   = @{}
$Assignments  = @(); $AccountCount = 1; Write-Host ""
foreach ($Account in $AccountList) {
    $Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
 {[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
    Write-Host "`r$Duration - Account $AccountCount of $($AccountList.Count)
 (Assignments:$($Assignments.Count))        " -NoNewline
    $AccountCount++
    foreach ($PS in Get-SSOADMNPermissionSetsProvisionedToAccountList -AccountId
 $Account.Id @SSOParams) {
        if (-not $PSsets[$PS]) {$PSsets[$PS] = (Get-SSOADMNPermissionSet @SSOParams -
PermissionSetArn $PS).Name;$APICalls++}
        $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
PermissionSetArn $PS -AccountId $Account.Id
        if ($AssignmentsResponse.NextToken) {$AccountAssignments =
 $AssignmentsResponse.AccountAssignments}
        else {$AccountAssignments = $AssignmentsResponse}
        While ($AssignmentsResponse.NextToken) {
            $AssignmentsResponse = Get-SSOADMNAccountAssignmentList @SSOParams -
PermissionSetArn $PS -AccountId $Account.Id -NextToken $AssignmentsResponse.NextToken
            $AccountAssignments += $AssignmentsResponse.AccountAssignments}
        foreach ($Assignment in $AccountAssignments) {
            if (-not $Principals[$Assignment.PrincipalId]) {
                $AssignmentType = $Assignment.PrincipalType.Value
                $Expression     = "Get-IDS"+$AssignmentType+" @IdsParams -"+
$AssignmentType+"Id "+$Assignment.PrincipalId
                $Principal      = Invoke-Expression $Expression
                if ($Assignment.PrincipalType.Value -eq "GROUP")
 { $Principals[$Assignment.PrincipalId] = $Principal.DisplayName }
                else { $Principals[$Assignment.PrincipalId] = $Principal.UserName }
```

```
            }
            $Assignments += [PSCustomObject]@{
                AccountName     = $Account.Name
                PermissionSet   = $PSsets[$PS]
                Principal       = $Principals[$Assignment.PrincipalId]
                Type            = $Assignment.PrincipalType.Value}
        }
    }
}
$Duration = New-Timespan -Start $Start -End (Get-Date) | ForEach-Object
 {[Timespan]::New($_.Days, $_.Hours, $_.Minutes, $_.Seconds)}
Write-Host "`r$($AccountList.Count) accounts done in $Duration. Outputting result to
 $OutputFile"
$Assignments | Sort-Object Account | Export-CSV -Path $OutputFile -Force
```

監控和修復 AWS KMS 金鑰的排程刪除

由米克許汗 (AWS) 和拉姆雅普里帕卡 (AWS) 創建

環境：生產	技術：安全性、身分識別、合規性；營運	AWS 服務：Amazon SNS; AWS CloudTrail; Amazon CloudWatch

Summary

在 Amazon Web Services (AWS) 雲端上，刪除 AWS 金鑰管理服務 (AWS KMS) 金鑰可能會導致資料遺失。刪除會移除金鑰材料和與 AWS KMS 金鑰相關聯的所有中繼資料，而且無法復原。刪除 AWS KMS 金鑰後，您無法再解密該 AWS KMS 金鑰下加密的資料，因此無法復原資料。

此模式會設定監控，並在應用程式或使用者排定 AWS KMS 金鑰進行刪除時發出通知。如果收到通知，您可能想要取消刪除 AWS KMS 金鑰，然後重新考慮刪除該金鑰的決定。該模式使用 AWS Systems Manager 自動化執行手冊AWSConfigRemediation，CancelKeyDeletion以方便取消刪除 AWS KMS 金鑰。

注意：模式的 CloudFormation 範本必須部署在您要監控刪除 AWS KMS 金鑰的所有 AWS 區域。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 了解下列 AWS 服務：
  - Amazon EventBridge
  - AWS KMS
  - Amazon Simple Notification Service (Amazon SNS)
  - AWS Systems Manager

限制

- 任何解決方案的自訂都需要 AWS CloudFormation 範本和此模式中使用的 AWS 服務的知識。

- 目前，該解決方案使用默認事件總線，可以根據需求進行定制。如需有關自訂事件匯流排的詳細資訊，請參閱 AWS 文件。

架構

目標技術堆疊

- Amazon EventBridge
- AWS KMS
- Amazon SNS
- AWS Systems Manager
- 自動化使用以下內容：
  - AWS Command Line Interface (AWS CLI) (AWS CLI) 或 AWS 開發套件
  - AWS CloudFormation 堆疊

目標架構

1. 已排程刪除 AWS KMS 金鑰。
2. 排程的刪除事件是由規則評估。 EventBridge
3. 該 EventBridge 規則涉及 Amazon SNS 主題。
4. 此 EventBridge 規則會啟動 Systems Manager 自動化和工作手冊。
5. 執行手冊會取消刪除。

自動化和規模

CloudFormation 堆疊會部署所有必要的資源和服務，以便此解決方案運作。該模式可以在單一帳戶中獨立執行，也可以 CloudFormation StackSets 針對多個獨立帳戶或組織使用 AWS 執行。

```
aws cloudformation create-stack --stack-name  <stack-name>\
    --template-body file://<Full-Path-of-file> \
    --parameters ParameterKey=,ParameterValue= \
    --capabilities CAPABILITY_NAMED_IAM
```

工具

工具

- AWS CloudFormation — AWS CloudFormation 是一項服務，可協助您建立 Amazon Web Services 資源模型和設定，以減少管理這些資源的時間，將更多時間專注於在 AWS 上執行的應用程式。您可以使用 CloudFormation 範本在 AWS 區域的 AWS 帳戶中建立堆疊。範本描述您想要的所有 AWS 資源，並為您 CloudFormation 佈建和設定這些資源。
- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。
- Amazon EventBridge — Amazon EventBridge 是一種無伺服器事件匯流排服務，將您的應用程式與來自各種來源的資料連接起來。 EventBridge 從您自己的應用程式和 AWS 服務交付即時資料串流，並將該資料路由到 AWS Lambda 等目標。 EventBridge 簡化了構建事件驅動架構的過程。
- AWS KMS — AWS Key Management Service (AWS KMS) 是一項受管服務，可用來建立和控制 AWS KMS 金鑰，這是用來加密資料的加密金鑰。
- AWS 開發套件 — AWS 工具包含開發套件，因此您可以使用自己選擇的程式設計語言在 AWS 上開發和管理應用程式。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。發佈者透過製作並傳送訊息到主題 (其為邏輯存取點和通訊管道) 與訂閱者進行非同步的通訊。
- AWS Systems Manager — AWS Systems Manager 是一項 AWS 服務，可讓您在 AWS 上檢視和控制基礎設施。使用系 Systems Manager 主控台，您可以在 AWS 資源中自動執行操作任務。Systems Manager 透過掃描您的受管執行個體並報告 (或採取修正動作) 其偵測的任何政策違規，協助您保持安全與合規。

Code

- 已貼附專案的`alerting_ct_logs.yaml` CloudFormation 樣板。

史诗

準備 AWS 帳戶

任務	描述	所需技能
安裝和設定 AWS CLI。	安裝 AWS CLI 第 2 版。然後為身分、預設輸出格式以及 AWS CLI 用於與 AWS 互動的預設 AWS 區域設定安全登入資料設定。 身分識別必須具備執行工作所需的權限。	開發者、安全工程師

部署 AWS CloudFormation 範本

任務	描述	所需技能
下載 CloudFormation 範本。	將附件下載到電腦上的本機路徑，然後解壓縮`alerting_ct_logs.yaml`　範本檔案。	開發者、安全工程師
部署範本。	在已設定 AWS 帳戶設定檔的終端機視窗中，執行下列命令。 `aws cloudformation create-stack --stack-name <stack_name> \ --capabilities <Value> \ --template-body file://<Full_Path> \ --parameters ParameterKey=DestinationEmailAddress,ParameterValue=<Value> \`	開發者、安全工程師

任務	描述	所需技能
	ParameterKey=SNS TopicName,Parameter Value=<Value> \ ParameterKey=EnableRemediation ,ParameterValue=<Value> \ ParameterKey=AutomationAssumeRole, ParameterValue=<Value> 在下一個步驟中，輸入範本參數的值。	

任務	描述	所需技能
完成範本參數。	輸入所需的參數值。 • `DestinationEmailAddress` — 排定刪除 AWS KMS 金鑰時收到警示的電子郵件地址。 • `SNSTopicName` — Amazon SNS 話題的名稱。 • `EnableRemediation` — 使用 Systems Manager 手冊取消計劃的密鑰刪除. 允許的值為 `true` 和 `false`。 • `AutomationAssumeRole` — 角色的 Amazon 資源名稱 (ARN)，可讓 Systems Manager 自動化代表您執行動作。如需詳細資訊，請參閱 AWSConfigRemediation-CancelKeyDeletion 文件中的必要 IAM 許可一節。 • `Capabilities` — 若 CloudFormation 要讓 AWS 建立堆疊，您必須明確確認您的堆疊範本包含特定功能。	開發者、安全工程師

確認訂閱

任務	描述	所需技能
確認訂閱。	檢查您的電子郵件收件匣，然後在從 Amazon SNS 收到的電子郵件訊息中選擇「確認訂	開發者、安全工程師

任務	描述	所需技能
	閱」。網頁瀏覽器視窗隨即開啟，並顯示訂閱確認和您的訂閱 ID。	

相關資源

參考

- 為 AWS 服務建立規則
- 建立 Amazon CloudWatch 警示以偵測待刪除之 AWS KMS 金鑰的使用情況

教學課程和影片

- 如何開始使用 Amazon EventBridge
- 深入探討 Amazon EventBridge (AWS 線上技術會談)

AWS 工作坊

- 使用 EventBridge 規則

其他資訊

下列程式碼提供範例，說明如何擴充解決方案，以監控和通知您任何 AWS 服務的任何變更。範例包括預先定義的樣式和自訂樣式。如需詳細資訊，請參閱中的事件和事件模式 EventBridge。

```
EventPattern:
        source:
        - aws.kms
        detail-type:
        - AWS API Call via CloudTrail
        detail:
          eventSource:
          - kms.amazonaws.com
          eventName:
          - ScheduleKeyDeletion
```

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Security Hub 識別 AWS Organizations 中的公有 S3 儲存貯體

創建者：莫拉德謝法維 (AWS)、阿倫錢達皮萊 (AWS) 和帕拉格納格韋卡 (AWS)

環境：生產	技術：安全性、身分識別、合規性、儲存與備份	工作負載：所有其他工作
AWS 服務：Amazon EventBridge；AWS Security Hub；Amazon SNS		

Summary

此模式說明如何建立機制，以識別 AWS Organizations 帳戶中的公用 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體。此機制的運作方式是使用 AWS 安全中 AWS Security Hub 中 AWS 基礎安全最佳實務 (FSBP) 標準的控制來監控 S3 儲存貯體。您可以使用 Amazon EventBridge 處理 Security Hub 發現的結果，然後將這些發現項目發佈到 Amazon Simple Notification Service (Amazon SNS) 主題。組織中的利益相關者可以訂閱主題，並立即收到有關發現結果的電子郵件通知。

新的 S3 儲存貯體及其物件預設不允許公開存取。您可以在必須根據組織的需求修改預設 Amazon S3 組態的案例中使用此模式。例如，這可能是您擁有一個 S3 儲存貯體託管面向公開的網站或網際網路上每個人都必須能夠從 S3 儲存貯體讀取的檔案的情況。

Security Hub 通常部署為集中服務，以合併所有安全發現項目，包括與安全標準和合規要求相關的發現項目。您還可以使用其他 AWS 服務來偵測公有 S3 儲存貯體，但此模式使用最少組態的現有 Security Hub 部署。

先決條件和限制

先決條件

- 具有專屬 Security Hub 管理員帳戶的 AWS 多帳戶設定
- 在您要監控的 AWS 區域中啟用安全中樞和 AWS Config (注意：如果您想要從單一彙總區域監控多個區域，則必須在 Security Hub 中啟用跨區域彙總)。
- 存取和更新 Security Hub 管理員帳戶的使用者權限、讀取組織中所有 S3 儲存貯體的存取權限，以及關閉公用存取權限 (如果需要)

架構

技術, 堆

- AWS Security Hub
- Amazon EventBridge
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)

目標架構

下圖顯示使用 Security Hub 識別公用 S3 儲存貯體的架構。

該圖顯示了以下工作流程：

1. Security Hub 使用 FSBP 安全標準中的 S3.2 和 S3.3 控制來監控所有 AWS Organizations 帳戶 (包括管理員帳戶) 中 S3 儲存貯體的組態，並偵測是否發現儲存貯體是否設定為公用。
2. Security Hub 系統管理員帳戶會從所有成員帳戶存取發現項目 (包括 S3.2 和 S3.3 的發現項目)。
3. Security Hub 會自動將所有新發現項目和所有更新傳送至現有發現項目， EventBridge 做為 Security Hub 發現項目-匯入的事件 這包括來自管理員和成員帳戶的發現項目事件。
4. EventBridge 規則會篩選 S3.2 和 S3.3 中`ComplianceStatus`的發現項目`FAILED`，其工作流程狀態為以及的`NEWRecordState`。`ACTIVE`
5. 規則使用事件模式來識別事件，並在符合後將其傳送至 SNS 主題。
6. SNS 主題會將事件傳送給其訂閱者 (例如透過電子郵件)。
7. 指派接收電子郵件通知的安全分析師會審核有問題的 S3 儲存貯體。
8. 如果值區已核准可供公開存取，安全分析師會將 Security Hub 中對應發現項目的工作流程狀態設定為`SUPPRESSED`。否則，分析師會將狀態設定為`NOTIFIED`。這樣可以消除 S3 儲存貯體的 future 通知，並減少通知噪音。
9. 如果工作流程狀態設為`NOTIFIED`，安全分析師會與值區擁有者一起檢閱發現項目，以判斷公開存取權是否合理，並符合隱私權和資料保護要求。調查結果會移除值區的公開存取權，或核准公開存取權。在後一種情況下，安全分析師會將工作流程狀態設定為`SUPPRESSED`。

備註：架構圖適用於單一區域和跨區域彙總部署。在圖表中的帳戶 A、B 和 C 中，Security Hub 可以屬於與管理員帳戶相同的區域，或者如果啟用了跨區域彙總，則屬於不同的區域。

工具

AWS 工具

- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。 EventBridge 從您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務提供即時資料串流。 EventBridge 如果資料與使用者定義的規則相符，則將該資料路由到 SNS 主題和 AWS Lambda 函數等目標。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。Security Hub 也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。Security Hub 會從 AWS 帳戶、服務和支援的第三方合作夥伴產品收集安全資料，然後協助分析安全趨勢並找出最優先順序的安全問題。

史诗

設定 Security Hub 帳戶

任務	描述	所需技能
在 AWS Organizations 帳戶中啟用安全中心。	若要在您要監控 S3 儲存貯體的組織帳戶中啟用 Security Hub，請參閱 AWS Security Hub Security Hub 使用者指南中的指定 Security Hub 管理員帳戶 (主控台) 和管理屬於組織的成員帳戶中的指導方針。	AWS 管理員
(選擇性) 啟用跨區域彙總。	如果您想要從單一區域監控多個區域中的 S3 儲存貯體，請設定跨區域彙總。	AWS 管理員
啟用 FSBP 安全性標準的 S3.2 和 S3.3 控制項。	您必須啟用 FSBP 安全性標準的 S3.2 和 S3.3 控制項。	AWS 管理員

任務	描述	所需技能
	1. 若要啟用 S3.2 控制項，請遵循 [S3.2] S3 儲存貯體中的指示，應禁止 AWS Security Hub 使用者指南中的公開讀取存取。 2. 若要啟用 S3.3 控制項，請遵循 AWS Security Hub 使用者指南中的 [3] S3 儲存貯體應禁止公用寫入存取的指示。	

設定環境

任務	描述	所需技能
設定 SNS 主題和電子郵件訂閱。	1. 登入 AWS 管理主控台並開啟 Amazon SNS 主控台。 2. 在導覽窗格中選擇 Topics (主題)，然後選擇 Create topic (建立主題)。 3. 針對 Type (類型)，選擇 Standard (標準)。 4. 在「名稱」中，輸入主題的名稱 (例如，公用 S3 值區)。 5. 請選擇 建立主題。 6. 在主題的 [訂閱項目] 索引標籤上，選擇 [建立訂閱]。 7. 關於通訊協定，請選擇電子郵件。 8. 在 Endpoint 中，輸入將接收通知的電子郵件地址。您可以使用 AWS 管理員、IT 專	AWS 管理員

任務	描述	所需技能
	業人員或資訊安全專業人員的電子郵件地址。 9. 選擇建立訂閱。若要建立其他電子郵件訂閱，請視需要重複步驟 6-8。	

任務	描述	所需技能
設定 EventBridge 規則。	1. 開啟 EventBridge 主控台。 2. 在 [開始使用] 區段中，選取 [EventBridge 規則]，然後選擇 [建立規則]。 3. 在 [定義規則詳細資訊] 頁面上，對於 [名稱]，輸入規則的名稱 (例如，public-s 3-bucket)。選擇下一步。 4. 在 [事件模式] 區段中，選擇 [編輯模式]。 5. 複製下列程式碼，將它貼到事件模式程式碼編輯器中，然後選擇 [下一步]。 `{ "source": ["aws.securityhub"], "detail-type": ["Security Hub Findings - Imported"], "detail": { "findings": { "Compliance": { "Status": ["FAILED"] }, "RecordState": ["ACTIVE"], "Workflow": { "Status": ["NEW"] }, "ProductFields": { "ControlId": ["S3.2", "S3.3"] }`	AWS 管理員

任務	描述	所需技能
	<pre> } } }</pre> 然後，執行下列動作： 1. 在 [選取目標] 頁面上，針對 [選取目標] 選取 SNS 主題作為目標，然後選取您先前建立的主題。 2. 選擇 [下一步]，再選擇 [下一步]，然後選擇 [建立規則]	

故障診斷

問題	解決方案
我有一個啟用公共訪問權限的 S3 存儲桶，但我沒有收到它的電子郵件通知。	這可能是因為儲存貯體是在另一個區域建立的，而且 Security Hub 系統管理員帳戶中未啟用跨區域彙總。若要解決此問題，請啟用跨區域彙總，或在 S3 儲存貯體目前所在的區域實作此模式的解決方案。

相關資源

- 什麼是 AWS Security Hub？（Security Hub 文檔）
- AWS 基礎安全最佳實務 (FSBP) 標準 (Security Hub 文件)
- AWS Security Hub 多帳戶啟用指令碼 (AWS 實驗室)
- Amazon S3 的安全最佳實務 (Amazon S3 文件)

其他資訊

監控公有 S3 儲存貯體的工作流

下列工作流程說明如何監控組織中的公用 S3 儲存貯體。工作流程假設您已完成此模式的設定 SNS 主題和電子郵件訂閱故事中的步驟。

1. S3 儲存貯體設定為公開存取權時，您會收到電子郵件通知。
   - 如果值區已核准可供公開存取，請SUPPRESSED在 Security Hub 管理員帳戶中將對應發現項目的工作流程狀態設定為。這樣可以防止 Security Hub 發出此值區的進一步通知，並且可以消除重複的警示。
   - 如果值區未經核准可供公開存取，請將 Security Hub 管理員帳戶中對應發現項目的工作流程狀態設定為NOTIFIED。這樣可以防止安全中心從 Security Hub 發出此存儲桶的進一步通知，並且可以消除噪音。
2. 如果值區可能包含敏感資料，請立即關閉公開存取權，直到審核完成為止。如果您關閉公用存取權，則 Security Hub 會將工作流程狀態變更為RESOLVED。然後，電子郵件通知值區停止。
3. 尋找將儲存貯體設定為公用的使用者 (例如，使用 AWS CloudTrail) 並開始審核。審核結果會移除值區的公開存取權，或核准公開存取權。如果核准公用存取權，則會將對應搜尋結果的工作流程狀態設定為SUPPRESSED。

使用 AWS 以程式碼形式管理 AWS IAM 身分中心許可集 CodePipeline

由安德烈·卡瓦爾坎特（AWS）和克萊森·阿莫利姆（AWS）創建

代碼存儲庫：aws-iam-identity-center-管道	環境：生產	技術：安全性、身分識別、合規性； DevOps
AWS 服務：AWS CodeBuild；AWS CodeCommit；AWS CodePipeline；AWS IAM 身分中心		

Summary

AWS IAM 身分中心 (AWS Single Sign-On 的後續任務) 可協助您集中管理對所有 AWS 帳戶和應用程式的單一登入 (SSO) 存取。您可以在 IAM 身分中心建立和管理使用者身分識別，也可以連接現有的身分識別來源，例如 Microsoft Active Directory 網域或外部身分識別提供者 (IdP)。IAM 身分中心提供統一的管理體驗，可使用許可集來定義、自訂和指派對 AWS 環境的精細存取權。權限集適用於來自 AWS IAM 身分中心身分存放區或外部 IdP 的聯合身分使用者和群組。

此模式可協助您在多帳戶環境中以程式碼的形式管理 IAM 身分中心權限集，而該環境是以 AWS Organizations 中的組織形式進行管理。使用此模式，您可以實現以下目標：

- 建立、刪除及更新權限集
- 建立、更新或刪除目標 AWS 帳戶、組織單位 (OU) 或組織根目錄的權限集指派。

為了以程式碼形式管理 IAM 身分中心許可和指派，此解決方案會部署使用 AWS、AWS 和 AWS CodeCommit 的持續整合和持續交付 (CI/CD) 管道。 CodeBuild CodePipeline您可以在儲存在 CodeCommit 存放庫中的 JSON 範本中管理權限集和指派。當 Amazon EventBridge 規則偵測到儲存庫的變更或偵測到目標 OU 中帳戶的修改時，就會啟動 AWS Lambda 函數。Lambda 函數會啟動 CI/CD 管道，以更新 IAM 身分識別中心中的權限集和指派。

先決條件和限制

先決條件

- 在 AWS Organizations 中以組織形式管理的多帳戶環境。如需詳細資訊，請參閱建立組織。
- 使用身分識別來源啟用和設定的 IAM 身分識別中心。如需詳細資訊，請參閱 IAM 身分中心文件中的入門指南。
- 已註冊為 IAM 身分中心委派管理員的成員帳戶。如需指示，請參閱 IAM 身分中心說明文件中的註冊成員帳戶。
- 在 IAM 身分中心委派的管理員帳戶和組織的管理帳戶中部署 AWS CloudFormation 堆疊的許可。如需詳細資訊，請參閱 CloudFormation 說明文件中的控制存取。
- 身分識別中心委派管理員中的 Amazon 簡易儲存服務 (Amazon S3) 儲存貯體，用於上傳成品程式碼。如需指示，請參閱建立值區。
- 組織管理帳戶的帳戶 ID。如需指示，請參閱尋找您的 AWS 帳戶 ID。

限制

- 此模式無法用於管理或指派單一帳戶環境或非 AWS Organization 中以組織形式管理的帳戶的權限集。
- 部署後，無法修改權限集名稱、指派 ID 和 IAM 身分中心主體類型和 ID。
- 此模式可協助您建立和管理自訂權限。您無法使用此模式來管理或指派預先定義的權限。
- 此模式無法用於管理組織管理帳戶的權限集。

架構

技術, 堆

- AWS CodeBuild
- AWS CodeCommit
- AWS CodePipeline
- Amazon EventBridge
- AWS 身分中心
- AWS Lambda
- AWS Organizations

目標架構

該圖顯示以下工作流程：

1. 使用者進行下列其中一項變更：
   a. 提交 CodeCommit 儲存庫的一或多個變更
   b. 修改 AWS Organizations 中組織單位 (OU) 中的帳戶
2. 如果使用者對 CodeCommit 儲存庫提交了變更，CodeChange EventBridge 則規則會偵測變更，並在 IAM 身分中心委派的管理員帳戶中啟動 Lambda 函數。此規則不會對儲存庫中某些檔案 (例如README.md檔案) 的變更做出反應。

   如果使用者修改了組織單位中的帳戶，則MoveAccount EventBridge 規則會偵測到變更，並在組織的管理帳戶中啟動 Lambda 函數。
3. 啟動的 Lambda 函數會在中啟動 CI/CD 管線。 CodePipeline
4. CodePipeline 啟動CodebuildTemplateValidation CodeBuild 專案。
5. CodebuildTemplateValidation CodeBuild 專案使用 CodeCommit 儲存庫中的 Python 指令碼來驗證權限集範本。 CodeBuild 驗證下列項目：
   - 權限集名稱是唯一的。
   - 指派陳述式 ID (Sid) 是唯一的。
   - CustomPolicy參數中的策略定義和有效。(此驗證使用 AWS Identity and Access Management 存取分析器。)
   - 受管政策的 Amazon 資源名稱 (ARN) 有效。
6. 此CodebuildPermissionSet CodeBuild 專案使用適用於 Python 的 AWS 開發套件 (Boto3) 來刪除、建立或更新身分識別中心中的許可集。只有具有SSOPipeline:true標籤的權限集才會受到影響。透過此管線管理的所有權限集都有此標籤。
7. CodebuildAssignments CodeBuild 專案會使用 Terraform 來刪除、建立或更新 IAM 身分中心中的指派。Terraform 後端狀態檔案存放在同一帳戶的 S3 儲存貯體中。
8. CodeBuild 在組織的管理帳戶中擔任 lookup IAM 角色。它會呼叫組織和身分存放區 API，以列出授予或撤銷權限所需的資源。
9. CodeBuild 更新 IAM 身分中心中的許可集和指派。

自動化和規模

由於多帳戶環境中的所有新帳戶都會移至 AWS Organizations Organization 中的特定組織單位，因此此解決方案會自動執行並將所需權限集授予您在指派範本中指定的所有帳戶。不需要額外的自動化或縮放動作。

在大型環境中，向 IAM 身分中心發出的 API 請求數量可能會導致此解決方案的執行速度更慢。Terraform 和 Boto3 會自動管理節流，以最大限度地減少任何性能降低。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS CodeBuild 是全受管的建置服務，可協助您編譯原始程式碼、執行單元測試，以及產生準備好部署的成品。
- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS 可 CodePipeline協助您快速建模和設定軟體發行的不同階段，並自動執行持續發行軟體變更所需的步驟。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS IAM 身分中心可協助您集中管理對所有 AWS 帳戶和雲端應用程式的單一登入 (SSO) 存取。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- 適用於 Python 的 AWS 開發套件 (Boto3) 是一套軟體開發套件，可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

代碼存儲庫

此模式的代碼可在 aws-iam-identity-center-pipeline 存儲庫中找到。存放庫中的 Templates 資料夾包含權限集和指派的範例範本。它還包括用於在目標帳戶中部署 CI/CD 管道和 AWS 資源的 AWS CloudFormation 範本。

最佳實務

- 在開始修改權限集與指派範本之前，建議您先規劃組織的權限集。考慮應該是什麼許可、權限集應套用到哪些帳戶或 OU，以及哪些 IAM Identity Center 主體 (使用者或群組) 應受到權限集的影響。部署後，無法修改權限集名稱、關聯 ID 和 IAM 身分中心主體類型和 ID。
- 遵守最低權限原則，並授予執行任務所需的最低權限。如需詳細資訊，請參閱 IAM 文件中的授與最低權限和安全性最佳實務。

史诗

規劃權限集與指派

任務	描述	所需技能
複製儲存庫。	在 bash 外殼中，輸入以下命令。這將從中克隆 aws-iam-identity-center-管道存儲庫。GitHub `git clone https://github.com/aws-samples/aws-iam-identity-center-pipeline.git`	DevOps 工程師
定義權限集。	1. 在複製的存放庫中，導覽至資templates/permissionsets 料夾，然後開啟其中一個可用範本。 2. 在Name參數中，輸入權限集的名稱。此值必須是唯一的，並且在部署後無法變更。 3. 在Description 參數中，簡要描述權限集，例如其使用案例。	DevOps 工程師

任務	描述	所需技能
	4. 在SessionDuration 參數中，指定使用者可以登入 AWS 帳戶的時間長度。使用 ISO-8601 持續時間格式（維基百科），PT4H例如 4 小時。如果未定義任何值，IAM 身分中心的預設值為 1 小時。 5. 自訂權限集中的原則。以下所有參數都是可選的，並且可以在部署後進行修改。您必須至少使用其中一個參數，才能定義權限集中的原則： • 在ManagedPolicies 參數中，輸入您要指派之任何 AWS 受管政策的 ARN。 • 在CustomerManagedPolicies 參數中，輸入您要指派之任何客戶管理策略的名稱。請勿使用 ARN。 • 在PermissionBoundary 參數中，執行下列動作以指派權限界限： • 如果您使用 AWS 受管政策做為許可界限，請在中PolicyType AWS、輸入和輸入該政策的 ARN。Policy	

任務	描述	所需技能
	• 如果您使用客戶管理的政策做為權限界限 `PolicyType` ，請在中`CustomerPolicy`、輸入和輸入政策的名稱。請勿使用 ARN。 • 在`CustomPolicy` 參數中，定義您要指派的任何自訂 JSON 格式原則。如需 JSON 原則結構的詳細資訊，請參閱 JSON 政策概觀。 6. 儲存並關閉權限集範本。建議您使用與權限集名稱相符的名稱來儲存檔案。 7. 重複此程序，視需要為您的組織建立任意數量的權限集，並刪除任何不需要的範例範本。	

任務	描述	所需技能
定義指派。	1. 在複製的存放庫中，導覽至資`templates/assignments` 料夾，然後開啟`iam-identitycenter-assigments.json` 。此檔案說明您要如何將權限集指派給 AWS 帳戶或 OU。 2. 在`SID`參數中，輸入指定的識別碼。此值必須是唯一的，並且在部署後無法修改。 3. 在`Target`參數中，定義您要套用權限集的帳戶或組織。有效值為帳戶識別碼、OU ID、OU 名稱或`root`。 `root`會將權限集指派給組織中的所有成員帳戶 (不包括管理帳戶)。以雙引號輸入值，並以逗號分隔多個值。如需如何尋找 ID 的相關指示，請參閱檢視帳戶的詳細資料或檢視 OU 的詳細資料。 4. 在`PrincipalType` 參數中，輸入將受權限集影響的 IAM 身分中心主體類型。有效值為 USER 或 GROUP。部署後無法修改此值。 5. 在`PrincipalID` 參數中，輸入 IAM 身分識別中心身分存放區中將受權限集影響的使用者或群組的名稱。部署後無法修改此值。	DevOps 工程師

任務	描述	所需技能
	6. 在`PermissionSetName` 參數中，輸入您要指派之權限集的名稱。 7. 重複步驟 2-6，在此檔案中建立任意數量的指定。一般而言，每個權限集都有一個指派。刪除任何不需要的範例指派。 8. 儲存並關閉 `iam-identitycenter-assigments.json` 檔案。	

部署權限集和指派

任務	描述	所需技能
將檔案上傳到 S3 儲存貯體。	1. 將複製的儲存庫壓縮為 .zip 檔案。 2. 登入 IAM 身分中心委派的系統管理員帳戶。 3. 前往 https://console.aws.amazon.com/s3/ 開啟的 Amazon Simple Storage Service (Amazon S3) 主控台。 4. 在左側導覽窗格中，選擇 Buckets (儲存貯體)。 5. 選擇您要用來部署此解決方案的值區。 6. 將 .zip 檔案上傳至目標 S3 儲存貯體。如需相關說明，請參閱上傳物件。	DevOps 工程師

任務	描述	所需技能
在 IAM 身分中心委派的管理員帳戶中部署資源。	1. 在 IAM 身分中心委派的系統管理員帳戶中，開啟主 CloudFormation 控台，網址為 https://console.aws.amazon.com/cloudformation/。 2. 部署`iam-identitycenter-pipeline.yaml` 範本。為堆疊提供清晰且描述性的名稱，並依照指示更新參數。如需指示，請參閱 CloudFormation 文件中的建立堆疊。	DevOps 工程師

任務	描述	所需技能
在 AWS 組織管理帳戶中部署資源。	1. 登入組織的管理帳戶。 2. 請在以下位置開啟 CloudFormation 主控台。 https://console.aws.amazon.com/cloudformation/ 3. 在導覽列中，選擇目前顯示的 AWS 區域名稱。然後選擇「`us-east-1` 地區」。需要此區域，以便`MoveAccount` EventBridge 規則可偵測與組織變更相關的 AWS CloudTrail 事件。 4. 部署`iam-identitycenter-organization` 範本。為堆疊提供清晰且描述性的名稱，並依照指示更新參數。如需指示，請參閱 CloudFormation 文件中的建立堆疊。	DevOps 工程師

更新權限集與指派

任務	描述	所需技能
更新權限集和指派。	當 `MoveAccount` Amazon EventBridge 規則偵測到組織中帳戶的修改時，CI/CD 管道會自動啟動並更新許可集。例如，如果您將帳戶新增至指派 JSON 檔案中指定的 OU，則	DevOps 工程師

任務	描述	所需技能
	CI/CD 管線會將權限集套用至新帳戶。 如果您要修改已部署的權限集和指派，請更新 JSON 檔案，然後將它們提交至 IAM 身分中心委派管理員帳戶中的 CodeCommit 存放庫。如需指示，請參閱 CodeCommit 文件中的建立提交。 使用 CI/CD 管線管理先前部署的權限集和關聯時，請注意下列事項： • 如果您變更權限集的名稱，CI/CD 管線會刪除原始權限集並建立新的權限集。 • 此管線僅管理具有`SSOPipeline:true` 標籤的權限集。 • 您可以在存放庫的相同資料夾中擁有多個權限集和指派範本。 • 如果刪除範本，管線會刪除指派或權限集。 • 如果您刪除整個指派 JSON 區塊，管線會從 IAM 身分中心刪除指派。 • 您無法刪除指派給 AWS 帳戶的權限集。首先，您必須取消指派權限集。	

故障診斷

問題	解決方案
存取遭拒錯誤	確認您具有部署 CloudFormation 範本及其中定義的資源所需的權限。如需詳細資訊，請參閱 CloudFormation 說明文件中的控制存取。
驗證階段中的管線錯誤	如果權限集或指派範本中有任何錯誤，就會出現此錯誤。 1. 在中 CodeBuild，檢視組建詳細資料。 2. 在組建記錄檔中，尋找驗證錯誤，該錯誤會提供有關造成組建失敗的原因的詳細資訊。 3. 更新權限集或指派範本，然後將其提交至存放庫。 4. CI/CD 管線會重新啟動專案 CodeBuild 。監視狀態以確認驗證錯誤已解決。

相關資源

- 權限集 (IAM 身分中心文件)

使用 AWS 秘密管理員來管理登入

創建者杜爾加普拉薩德奇普里（AWS）

創建者：AWS	環境：PoC 或試點	技術：資料庫；安全性、身分識別、合規性
AWS 服務：AWS Secrets Manager		

Summary

此模式會引導您完成使用 AWS Secrets Manager 動態擷取 Java Spring 應用程式的資料庫登入資料。

以往當您建立從資料庫擷取資訊的自訂應用程式時，通常必須內嵌登入資料 (秘密)，才可直接存取應用程式中的資料庫。輪換憑證的時候，您必須投入時間來更新應用程式以使用新的認證，然後散發更新的應用程式。如果您有多個共享憑據的應用程序，並且錯過了更新其中一個應用程序，則該應用程序將失敗。由於這種風險，許多使用者選擇不定期輪換其憑證，這有效地取代了另一種風險。

Secrets Manager 可讓您以 API 呼叫取代程式碼中的硬式編碼認證 (包括密碼)，以程式設計方式擷取密碼。這有助於確保秘密不會被正在檢查您的代碼的人入侵，因為秘密根本不存在。您也可以將 Secret Manager 設定為根據您指定的排程自動輪換密碼。這可讓您以短期密碼取代長期機密，這有助於大幅降低入侵的風險。如需詳細資訊，請參閱 AWS Secrets Manager 文件。

先決條件和限制

先決條件

- 可存取 Secrets Manager 的 AWS 帳戶
- 一個 Java 春季應用程序

架構

源, 技術, 堆棧

- 一個 Java Spring 應用程序與訪問數據庫的代碼，從應用程序 .properties 文件管理數據庫憑據。

目標技術堆疊

- 一個 Java Spring 應用程序與訪問數據庫的代碼，在秘密管理器管理數據庫憑據管理。應用程式 .properties 檔案會保存秘密 Secrets Manager 碼。

Secrets Manager 與應用程序集成

工具

- Secrets Manager — AWS Secrets Manager 是一項 AWS 服務，可讓您更輕鬆地管理機密。秘密可能是資料庫憑證、密碼、第三方 API 金鑰，甚至是任意文字。您可以使用秘密管理員主控台、秘密管 Secrets Manager 命令列介面 (CLI) 或機 Secrets Manager API 和 SDK，集中儲存和控制這些機 Secrets Manager 的存取。

史诗

在秘密管理器中存儲秘密

任務	描述	所需技能
將數據庫憑據存儲為秘密管理器中的秘密。	按照秘密管理員文件中的建立密碼中的步驟，在 Secrets Manager 中將 Amazon Relational Database Service 服務 (Amazon RDS) 或其他資料庫登入資料存放為秘 Secrets Manager。	系統管理員
設置 Spring 應用程序訪問 Secrets Manager 的權限。	根據 Java Spring 應用程序使用 Secrets Manager 的方式設置適當的權限。若要控制密碼的存取權，請根據 Secrets Manager 文件中提供的資訊建立政策，請參閱 < 針對秘密管理員使用身分型政策 (IAM 政	系統管理員

任務	描述	所需技能
	策) 和 ABAC 以及針對 Secrets Manager 使用以資源為基礎的政策 > 一節中所提供的資訊建立政策。請依照秘密管理員說明文件中擷取密碼值一節中的步驟進行。	

更新春季應用程序

任務	描述	所需技能
新增 JAR 相依性以使用 Secrets Manager。	如需詳細資訊，請參閱其他資訊一節。	Java 開發人員
將秘密的詳細信息添加到 Spring 應用程序中。	使用密碼名稱、端點和 AWS 區域更新應用程式 .properties 檔案。如需範例，請參閱其他資訊一節。	Java 開發人員
在 Java 中更新數據庫憑據檢索代碼。	在應用程式中，更新擷取資料庫認證的 Java 程式碼，以便從 Secrets Manager 擷取這些詳細資料。如需範例程式碼，請參閱「其他資訊」一節。	Java 開發人員

相關資源

- AWS Secrets Manager 文件
- 針對 Secrets Manager 使用以身分識別為基礎的政策 (IAM 政策) 和 ABAC
- 針對 Secrets Manager 使用資源型原則
- 範例程式碼

其他資訊

新增 JAR 相依性以使用 Secrets Manager

釋界：

```
<groupId>com.amazonaws</groupId>
    <artifactId>aws-java-sdk-secretsmanager</artifactId>
    <version>1.11. 355 </version>
```

搖籃：

```
compile group: 'com.amazonaws', name: 'aws-java-sdk-secretsmanager', version:
 '1.11.355'
```

使用密碼的詳細信息更新應用程序 .properties 文件

```
spring.aws.secretsmanager.secretName=postgres-local
spring.aws.secretsmanager.endpoint=secretsmanager.us-east-1.amazonaws.com
spring.aws.secretsmanager.region=us-east-1
```

在 Java 中更新數據庫憑據檢索代碼

```
String  secretName  =  env.getProperty("spring.aws.secretsmanager.secretName");
String  endpoints  =  env.getProperty("spring.aws.secretsmanager.endpoint");
String  AWS Region  =  env.getProperty("spring.aws.secretsmanager.region");
AwsClientBuilder.EndpointConfiguration  config  =  new
  AwsClientBuilder.EndpointConfiguration(endpoints, AWS Region);
AWSSecretsManagerClientBuilder  clientBuilder  =
  AWSSecretsManagerClientBuilder.standard();
clientBuilder.setEndpointConfiguration(config);
AWSSecretsManager  client  =  clientBuilder.build();

ObjectMapper  objectMapper  =  new  ObjectMapper();

JsonNode  secretsJson  =  null;

ByteBuffer  binarySecretData;

GetSecretValueRequest  getSecretValueRequest  =  new
  GetSecretValueRequest().withSecretId(secretName);
```

```
GetSecretValueResult  getSecretValueResponse  =  null;

try  {
     getSecretValueResponse  =  client.getSecretValue(getSecretValueRequest);
    }

catch  (ResourceNotFoundException  e)  {
     log.error("The requested secret "  +  secretName  +  " was not found");
    }

catch  (InvalidRequestException  e)  {
     log.error("The request was invalid due to: "  +  e.getMessage());
     }

catch  (InvalidParameterException  e)  {
     log.error("The request had invalid params: "  +  e.getMessage());
     }
if  (getSecretValueResponse  ==  null)  {
     return  null;
     }  // Decrypted secret using the associated KMS key // Depending on whether the
 secret was a string or binary, one of these fields will be populated


String secret = getSecretValueResponse.getSecretString();

if (secret != null) {
     try {
                secretsJson  =  objectMapper.readTree(secret);
           }

     catch  (IOException  e)  {
                log.error("Exception while retrieving secret values: "  +
  e.getMessage());
           }
}

else  {
     log.error("The Secret String returned is null");

     return null;

     }
     String  host  =  secretsJson.get("host").textValue();
```

```
    String  port  =  secretsJson.get("port").textValue();
    String  dbname  =  secretsJson.get("dbname").textValue();
    String  username  =  secretsJson.get("username").textValue();
    String  password  =  secretsJson.get("password").textValue();
}
```

在啟動時監控 Amazon EMR 叢集的傳輸中加密

環境：生產	技術：分析；大數據；雲端原生；安全性、身分識別、合規性	工作負載：開源
AWS 服務：Amazon EMR; Amazon SNS; AWS CloudTrail; Amazon CloudWatch		

Summary

此模式提供安全控制，可在啟動時監控 Amazon EMR 叢集，並在未啟用傳輸中加密時傳送警示。

Amazon EMR 是一種網絡服務，可讓您輕鬆地運行大數據框架，例如 Apache Hadoop 來處理和分析數據。Amazon EMR 透過執行映射和減少 parallel 步驟，以符合成本效益的方式處理大量資料。

資料加密可防止未經授權的使用者存取或讀取靜態資料或傳輸中的資料。靜態資料是指透過 Amazon 簡單儲存服務 (Amazon S3) 儲存在媒體中的資料，例如每個節點上的本機檔案系統、Hadoop 分散式檔案系統 (HDFS) 或 EMR 檔案系統 (EMRFS)。傳輸中的資料是指傳輸網路並在工作之間傳輸的資料。傳輸中加密支持阿帕奇星火，阿帕奇 TEZ，阿帕奇 Hadoop 的，阿帕奇 HBase 的 HBase 的，和普雷斯托開源加密功能。您可以透過從 AWS Command Line Interface (AWS CLI) (AWS CLI)、主控台或 AWS 開發套件建立安全組態，並指定資料加密設定來啟用加密。您可以透過下列兩種方式提供傳輸中加密的加密成品：

- 通過將證書的壓縮文件上傳到 Amazon S3。
- 藉由參照提供加密加工品的自訂 Java 類別。

此模式隨附的安全控制可監控 API 呼叫，並在「RunJob流程」動作上產生 Amazon CloudWatch 事件事件。此事件會呼叫 AWS Lambda 函數，該函數會執行 Python 指令碼。函數會從事件 JSON 輸入取得 EMR 叢集識別碼，並執行下列檢查以判斷是否有安全性違規：

- 檢查 EMR 叢集是否具有 Amazon EMR 特定的安全組態。
- 如果叢集確實具有安全性設定，請檢查傳輸中是否已啟用加密。

- 如果叢集沒有安全組態，請使用 Amazon Simple Notification Service (Amazon SNS) 將警示傳送至您提供的電子郵件地址。該通知會指定通知來源的 EMR 叢集名稱、違規詳細資訊、AWS 區域和帳戶資訊，以及 AWS Lambda ARN (Amazon 資源名稱)。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 用於上傳此模式提供的 Lambda 程式碼的 S3 儲存貯體。
- 您希望接收違規通知的電子郵件地址。
- 啟用 Amazon EMR 記錄功能，可存取所有 API 日誌。

限制

- 此偵探控制是區域性的，必須部署在您要監控的每個 AWS 區域中。

產品版本

- Amazon EMR 版本 4.8.0 或更高版本。

架構

工作流架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署範本。

工具

AWS 服務

- Amazon EMR — Amazon EMR 是一個受管叢集平台，可簡化在 AWS 上執行大數據架構 (例如 Apache Hadoop 和 Apache Spark)，以處理和分析大量資料。透過使用這些架構和相關的開放原始碼專案，您可以針對分析目的和商業智慧工作負載處理資料。此外，您可以使用 Amazon EMR 將大量資料轉換和移出其他 AWS 資料存放區和資料庫，例如 Amazon S3 和 Amazon DynamoDB。
- AWS Cloudformation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建這些資源，以及在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- AWS Cloudwatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。 CloudWatch 事件會在發生作業變更時瞭解作業變更，並視需要採取更正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只會在需要時執行您的程式碼，並自動從每天幾個請求擴展到每秒數千個請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- AWS SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

該模式包括一個包含兩個文件的附件：

- `EMRInTransitEncryption.zip`是包含安全控制 (Lambda 程式碼) 的壓縮檔案。
- `EMRInTransitEncryption.yml`是部署安全控制的 CloudFormation 範本。

有關如何使用這些文件的信息，請參見 Epics 部分。

史诗

部署安全控制

任務	描述	所需技能
將代碼上傳到 S3 存儲桶。	建立新的 S3 儲存貯體，或使用現有的 S3 儲存貯體上傳附加的`EMRInTransitEncryp`	雲端架構師

任務	描述	所需技能
	tion.zip 檔案 (Lambda 程式碼)。此儲存貯體必須與 CloudFormation 範本和您要評估的資源位於相同的 AWS 區域。	
部署 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 Cloudformation 主控台，然後部署附件中提供的EMRInTransitEncryption.yml 檔案。在下一個史詩中，提供模板參數的值。	雲架構師，

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選取的 S3 儲存貯體的名稱。此 S3 儲存貯體包含 Lambda 程式碼的 .zip 檔案，且必須與 CloudFormation 範本和要評估的資源位於相同的 AWS 區域。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中指定 Lambda 程式碼 .zip 檔案的位置，而不使用前導斜線 (例如，EMRInTransitEncryption.zip 或controls/EMRInTransitEncryption.zip)。	雲端架構師
提供電子郵件地址。	指定您要在其中接收違規通知的作用中電子郵件地址。	雲端架構師

任務	描述	所需技能
指定記錄日誌層級。	指定 Lambda 記錄的記錄層級和詳細資訊。 `Info`指定應用程式進度的詳細資訊訊息，應僅用於偵錯。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認電子郵件訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件訊息傳送至您提供的電子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台建立堆疊 (AWS CloudFormation 文件)
- 加密選項 (Amazon EMR 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

監控 Amazon ElastiCache 叢集以進行靜態加密

環境：生產	技術：安全性、身分識別、合規性、資料庫、基礎架構、雲端原生	工作負載：開源
AWS 服務：Amazon SNS; Amazon CloudWatch; Amazon ElastiCache		

Summary

Amazon ElastiCache 是 Amazon Web Services (AWS) 服務，提供高效能、可擴展且具成本效益的快取解決方案，可在雲端中分發記憶體內資料存放區或快取環境。它會從高輸送量和低延遲的記憶體內資料存放區擷取資料。此功能使其成為即時使用案例的熱門選擇，例如快取、工作階段存放區、遊戲、地理空間服務、即時分析和佇列。 ElastiCache 提供 Redis 和 Memcached 的資料存放區，這兩者都提供低於一毫秒的回應時間。

資料加密有助於防止未經授權的使用者讀取 Redis 叢集及其相關快取儲存系統上可用的敏感資料。這包括儲存到持續性媒體 (稱為靜態資料) 的資料，以及在用戶端和快取伺服器 (稱為傳輸中的資料) 之間通過網路時可能遭到攔截的資料。

您可以在建立複寫群組時 ElastiCache 為 Redis 啟用靜態加密，方法是將AtRestEncryptionEnabled參數設定為 true。啟用此參數時，它會在同步、備份和交換操作期間加密磁碟，並加密存放在 Amazon 簡單儲存服務 (Amazon S3) 中的備份。您無法在現有複寫群組上啟用靜態加密。建立複寫群組時，您可以使用下列兩種方式啟用靜態加密：

- 通過選擇默認選項，該選項使用服務管理的靜態加密。
- 透過使用客戶受管金鑰，並從 AWS Key Management Service (AWS KMS) 提供金鑰 ID 或 Amazon 資源名稱 (ARN)。

此模式提供安全控制，可監控 API 呼叫，並在CreateReplication群組作業上產生 Amazon E CloudWatch vents 事件。此事件會呼叫執行 Python 指令碼的 AWS Lambda 函數。函數會從事件 JSON 輸入取得複寫群組識別碼，並執行下列檢查以判斷是否有安全性違規：

- 檢查AtRestEncryptionEnabled密鑰是否存在。

- 如果AtRestEncryptionEnabled存在，則檢查該值以查看是否為 true。
- 如果該AtRestEncryptionEnabled值設定為 false，請使用 Amazon Simple Notification Service (Amazon SNS) 通知設定一個變數來追蹤違規，並將違規訊息傳送到您提供的電子郵件地址。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 用於上傳提供的 Lambda 程式碼的 S3 儲存貯體。
- 您希望接收違規通知的電子郵件地址。
- ElastiCache 啟用日誌記錄，以訪問所有 API 日誌。

限制

- 此偵探控制是區域性的，必須部署在您要監控的每個 AWS 區域中。
- 此控制項支援在虛擬私有雲端 (VPC) 中執行的複寫群組。
- 控制項支援執行下列節點類型的複寫群組：
  - R5、R4、R3
  - M5、M4、M3
  - T3、T2

產品版本

- ElastiCache 對於版本 3.2.6 或更高版本

架構

工作流架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- Amazon ElastiCache — Amazon 可 ElastiCache 讓您輕鬆地在 AWS 雲端中設定、管理和擴展分散式記憶體內快取環境。它提供高效能、可調整大小且符合成本效益的記憶體內快取，同時消除與部署和管理分散式快取環境相關的複雜性。 ElastiCache 適用於 Redis 和內存緩存引擎。
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- AWS Cloudwatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。 CloudWatch 事件會在發生作業變更時瞭解作業變更，並視需要採取更正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只會在需要時執行您的程式碼，並自動從每天幾個請求擴展到每秒數千個請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和用戶端之間的訊息傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

該模式包括一個包含兩個文件的附件：

- `ElasticCache-EncryptionAtRest.zip`是包含安全控制 (Lambda 程式碼) 的壓縮檔案。
- `elasticache_encryption_at_rest.yml`是部署安全控制的 CloudFormation 範本。

有關如何使用這些文件的信息，請參見 Epics 部分。

史诗

部署安全控制

任務	描述	所需技能
將代碼上傳到 S3 存儲桶。	建立新的 S3 儲存貯體，或使用現有的 S3 儲存貯體上傳附加的`ElastiCache-EncryptionAtRest.zip` 檔案 (Lambda 程式碼)。此儲存貯體必須與您要評估的資源位於相同的 AWS 區域。	雲端架構師
部署 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 Cloudformation 主控台，然後部署附件中提供的`elasticache_encryption_at_rest.yml` 檔案。在下一個史詩中，提供模板參數的值。	雲端架構師

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選取的 S3 儲存貯體的名稱。此 S3 儲存貯體包含 Lambda 程式碼的 .zip 檔案，且必須與 CloudFormation 範本和要評估的資源位於相同的 AWS 區域。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜	雲端架構師

任務	描述	所需技能
	線 (例如，ElasticCache-EncryptionAtRest.zip 或controls/ElasticCache-EncryptionAtRest.zip)。	
提供電子郵件地址。	提供您要接收違規通知的作用中電子郵件地址。	雲端架構師
指定記錄日誌層級。	指定記錄日誌層級和詳細資訊。 Info指定應用程式進度的詳細資訊訊息，應僅用於偵錯。 Error指定仍然允許應用程式繼續執行的錯誤事件。 Warning指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認電子郵件訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件訊息傳送至您提供的電子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- Redis ElastiCache 的靜態加密 (Amazon ElastiCache 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 AWS 設定監控 EC2 執行個體金鑰配對

環境：生產	技術：安全性、身分識別、合規	AWS 服務：Amazon SNS；AWS Config；AWS Lambda

Summary

在亞馬遜網路服務 (AWS) 雲端上啟動 Amazon 彈性運算雲端 (Amazon EC2) 執行個體時，最佳做法是建立或使用現有的 key pair 連接到執行個體。key pair 組包含儲存在執行個體中的公開金鑰和提供給使用者的私密金鑰，可透過 Secure Shell (SSH) 安全存取執行個體，並避免使用密碼。不過，使用者有時可能會不小心啟動執行個體，而不附加 key pair。因為金鑰配對只能在執行個體啟動期間指派，因此在沒有金鑰配對的情況下啟動的任何執行個體，快速識別並將其標示為不相容非相容性非常重要。這在要求使用密鑰對進行例如訪問的帳戶或環境中工作時特別有用。

此模式說明如何在 AWS Config 中建立自訂規則以監控 EC2 執行個體金鑰對。當執行個體被識別為不合規時，會使用透過 Amazon EventBridge 事件啟動的 Simple Notification Service (Amazon SNS) 通知傳送警示。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 針對您要監控的 AWS 區域啟用 AWS Config，並設定為記錄所有 AWS 資源

限制

- 此解決方案是特定於區域的。所有資源都應在相同的 AWS 區域中建立。

架構

目標技術堆疊

- AWS Config
- Amazon EventBridge

- AWS Lambda
- Amazon SNS

目標架構

1. AWS Config 會啟動規則。
2. 此規則會叫用 Lambda 函數來評估 EC2 執行個體的合規性。
3. Lambda 函數會將更新的合規狀態傳送至 AWS Config。
4. AWS Config 會將事件傳送到 EventBridge。
5. EventBridge 將規範遵循變更通知發佈至 SNS 主題。
6. Amazon SNS 會透過電子郵件傳送警示。

自動化和規模

此解決方案可監控區域內任意數量的 EC2 執行個體。

工具

工具

- AWS Config — AWS Config 是一項服務，可讓您評估、稽核和評估 AWS 資源的組態。AWS Config 會持續監控和記錄您的 AWS 資源組態，並可讓您根據所需的組態自動評估記錄的組態。
- Amazon EventBridge — Amazon EventBridge 是一種無伺服器事件匯流排服務，可將您的應用程式與來自各種來源的資料連接起來。
- AWS Lambda — AWS Lambda 是一種無伺服器運算服務，可支援執行程式碼而無需佈建或管理伺服器、建立工作負載感知叢集擴展邏輯、維護事件整合或管理執行階段。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是用於 (A2A) 和 application-to-application application-to-person (A2P) 通訊的全受管簡訊服務。

Code

已附加 Lambda 函數的程式碼。

史诗

建立 Lambda 函數以評估 Amazon EC2 合規性

任務	描述	所需技能
為 Lambda 建立 AWS Identity and Access Management (IAM) 角色。	在 AWS 管理主控台上，選擇 IAM，然後使用 Lambda 做為受信任的實體並新增`AmazonEventBridgeFullAccess` 和`AWSConfigRulesExecutionRole` 許可來建立角色。如需詳細資訊，請參閱 AWS 文件。	DevOps
建立和部署 Lambda 函數。	1. 在 Lambda 主控台上，使用從頭開始建立一個函數，並使用 Python 3.6 做為執行階段和先前建立的 IAM 角色。請記下 Amazon Resource Name (ARN)。 2. 在「程式碼」索引標籤上`lambda_function.py`，選擇並貼上附加至此模式的程式碼。 3. 若要儲存變更，請選擇 [部署]。	DevOps

建立自訂 AWS Config 規則

任務	描述	所需技能
新增自訂 AWS Config 規則。	在 AWS Config 主控台上，使用下列設定新增自訂規則：	DevOps

任務	描述	所需技能
	• ARN-先前創建的 Lambda 函數的 ARN • 觸發類型 — 組態變更 • 變更範圍 — 資源 • 資源類型 — 亞馬遜 EC2 執行個體 如需詳細資訊，請參閱 AWS 文件。	

設定偵測到符合性變更事件時的電子郵件通知

任務	描述	所需技能
建立 SNS 主題和訂閱。	在 Amazon SNS 主控台上，使用標準做為類型建立主題，然後使用電子郵件作為通訊協定建立訂閱。 當您收到確認電子郵件訊息時，請選擇確認訂閱的連結。 如需詳細資訊，請參閱 AWS 文件。	DevOps
建立 EventBridge 規則以啟動 Amazon SNS 通知。	在主 EventBridge 控台上，使用下列設定建立規則： • 服務名稱 — AWS Config • 事件類型 — Config 規則符合性變更 • 訊息類型 — 特定訊息類型、 ComplianceChangeNotification	DevOps

任務	描述	所需技能
	• 特定規則名稱 — 先前建立的 AWS Config 規則名稱 • 目標 — SNS 主題，您先前建立的主題 如需詳細資訊，請參閱 AWS 文件。	

驗證規則和通知

任務	描述	所需技能
建立 EC2 執行個體。	建立兩個任何類型的 EC2 執行個體並連接一個 key pair，並建立一個不含 key pair 的 EC2 執行個體。	DevOps
驗證規則。	1. 在 AWS Config 主控台的「規則」頁面上，選取您的規則。 2. 若要查看合規和不合規的 EC2 執行個體，請將範圍中的資源變更為 [全部]。確認兩個執行個體已列為相容，而且其中一個執行個體列為不相容。 3. 等待收到有關 EC2 執行個體合規狀態的 Amazon SNS 電子郵件通知。	DevOps

相關資源

- 建立角色以將許可委派給 AWS 服務

- 在 AWS Config 定中建立自訂規則
- 創建一個 Amazon SNS 主題
- 訂閱 Amazon SNS 主題
- 在 Amazon 中創建規則 EventBridge

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

監控安全群組的 ElastiCache 叢集

由蘇珊·康諾（AWS）和阿基特·馬圖爾（AWS）創建

環境：生產	技術：安全性、身分識別、合規性、資料庫、基礎架構、雲端原生	AWS 服務：Amazon SNS; AWS CloudTrail; Amazon CloudWatch; Amazon ElastiCache

Summary

Amazon ElastiCache 是 Amazon Web Services (AWS) 服務，提供高效能、可擴展且具成本效益的快取解決方案，用於在雲端中分發記憶體內資料存放區或快取環境。它會從高輸送量和低延遲的記憶體內資料存放區擷取資料。此功能使其成為即時使用案例的熱門選擇，例如快取、工作階段存放區、遊戲、地理空間服務、即時分析和佇列。 ElastiCache 提供 Redis 和 Memcached 的資料存放區，這兩者都提供低於一毫秒的回應時間。

安全群組會控制輸入和輸出流量，做為 ElastiCache 執行個體的虛擬防火牆。安全群組會在執行個體層級運作，而非子網路層級。對於每個安全性群組，您可以新增一組規則來控制執行個體的輸入流量，以及控制輸出流量的單獨規則集。您可以指定允許規則，但不能指定拒絕規則。

此模式提供安全控制，可監控 API 呼叫，並在CreateReplicationGroup、CreateCacheCluster和ModifyReplicationGroup操作上產生 Amazon E CloudWatch vents 事件。ModifyCacheCluster此事件會呼叫執行 Python 指令碼的 AWS Lambda 函數。函數會從事件 JSON 輸入取得複寫群組識別碼，並執行下列檢查以判斷是否有安全性違規：

- 檢查叢集的安全性群組是否與 Lambda 函數中設定的安全群組相符。
- 如果叢集的安全群組不相符，此功能會使用 Amazon Simple Notification Service (Amazon SNS)，將違規訊息傳送到您提供的電子郵件地址。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶

- 用於上傳提供的 Lambda 程式碼的 S3 儲存貯體。
- 您希望接收違規通知的電子郵件地址。
- ElastiCache 啟用日誌記錄，以訪問所有 API 日誌。

限制

- 此偵探控制是區域性的，必須部署在您要監控的每個 AWS 區域中。
- 此控制項支援在虛擬私有雲端 (VPC) 中執行的複寫群組。

架構

工作流架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation 將此範本部署 StackSets到您要監控的多個帳戶中。

工具

AWS 服務

- Amazon 可 ElastiCache讓您輕鬆地在 AWS 雲端中設定、管理和擴展分散式記憶體內快取環境。它提供高效能、可調整大小且符合成本效益的記憶體內快取，同時消除與部署和管理分散式快取環境相關的複雜性。 ElastiCache 適用於 Redis 和內存緩存引擎。
- AWS 可 CloudFormation協助您建立 AWS 資源的模型和設定、快速且一致地佈建它們，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- AWS Cloudwatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源中的變更。 CloudWatch 事件會在發生作業變更時知道，並視需要採取更正動作，方法是傳送訊息以回應環境、啟動功能、進行變更，以及擷取狀態資訊。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只會在需要時執行您的程式碼，並自動從每天幾個請求擴展到每秒數千個請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。

- Amazon Simple Notification Service (Amazon SNS) 會協調和管理發佈者和用戶端之間的訊息傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

該模式包括一個包含兩個文件的附件：

- `ElastiCacheAllowedSecurityGroup.zip`是包含安全控制 (Lambda 程式碼) 的壓縮檔案。
- `ElastiCacheAllowedSecurityGroup.yml`是部署安全控制的 CloudFormation 範本。

有關如何使用這些文件的信息，請參見 Epics 部分。

史诗

部署安全控制

任務	描述	所需技能
將代碼上傳到 S3 存儲桶。	建立新的 S3 儲存貯體，或使用現有的 S3 儲存貯體上傳附加的`ElastiCacheAllowedSecurityGroup.zip` 檔案 (Lambda 程式碼)。此儲存貯體必須與您要評估的資源位於相同的 AWS 區域。	雲端架構師
部署 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 Cloudformation 主控台，然後部署附件中提供的`ElastiCacheAllowedSecurityControl.yml` 檔案。在下一個史詩中，提供模板參數的值。	雲端架構師

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選取的 S3 儲存貯體的名稱。此 S3 儲存貯體包含 Lambda 程式碼的 .zip 檔案，且必須與 CloudFormation 範本和要評估的資源位於相同的 AWS 區域。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如，`ElasticCacheAllowedSecurityGroup.zip` 或`controls/ElasticCacheAllowedSecurityGroup.zip`)。	雲端架構師
提供電子郵件地址。	提供您要接收違規通知的作用中電子郵件地址。	雲端架構師
指定記錄日誌層級。	指定記錄日誌層級和詳細資訊。 `Info`指定應用程式進度的詳細資訊訊息，應僅用於偵錯。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認電子郵件訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件訊息傳送至您提供的電子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	雲端架構師

相關資源

- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- Amazon VPC 和 ElastiCache 安全性（ ElastiCache 適用於 Redis 的 Amazon 文檔 ）

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

監控 IAM 根使用者活動

創建者：莫斯特法·布魯吉 (AWS)

代碼存儲庫：aws-iam-root-user-活動監視器	環境：PoC 或試點	技術：安全性、身分識別、合規性、管理與治理
工作負載：所有其他工作	AWS 服務：Amazon EventBridge；AWS Lambda；Amazon SNS；AWS Identity and Access Management	

Summary

每個 Amazon Web Services (AWS) 帳戶都有一個根使用者。作為 AWS Identity and Access Management (IAM) 的安全最佳實務，我們建議您使用 root 使用者完成只有 root 使用者可以執行的任務。如需完整清單，請參閱 AWS 帳戶管理參考指南中的需要 root 使用者登入資料的任務。由於 root 使用者擁有所有 AWS 資源和帳單資訊的完整存取權，因此建議您不要使用此帳戶並對其進行監控是否有任何活動，這可能表示 root 使用者登入資料已遭入侵。

使用此模式，您可以設定事件驅動的架構來監控 IAM 根使用者。此模式設定了一個 hub-and-spoke 解決方案，用於監控多個 AWS 帳戶、分支帳戶，並將管理和報告集中在單一帳戶 (Hub 帳戶) 中。

使用 IAM 根使用者登入資料時，Amazon CloudWatch 和 AWS 會分別在日誌和追蹤中 CloudTrail 記錄活動。在支點帳戶中，Amazon EventBridge 規則會將事件傳送到中樞帳戶中的中央事件匯流排。在集線器帳戶中， EventBridge 規則會將事件傳送至 AWS Lambda 函數。該函數使用 Amazon Simple Notification Service (Amazon SNS) 主題，通知您根使用者活動。

在此模式中，您可以使用 AWS CloudFormation 範本在支點帳戶中部署監控和事件處理服務。您可以使用 T HashiCorp erraform 範本在 Hub 帳戶中部署事件管理和通知服務。

先決條件和限制

先決條件

1. 在 AWS 環境中部署 AWS 資源的許可。
2. 部署 CloudFormation 堆疊集的權限。如需詳細資訊，請參閱堆疊集作業的先決條CloudFormation件 (說明文件)。

3. 已安裝並準備使用的地形。如需詳細資訊，請參閱開始使用 — AWS (地形文件)。
4. 每個支點帳戶中的現有追蹤。如需詳細資訊，請參閱 AWS 入門 CloudTrail (CloudTrail 文件)。
5. 追蹤設定為將事件傳送至 CloudWatch 記錄檔。如需詳細資訊，請參閱將事件傳送至 CloudWatch 記錄CloudTrail 檔 (文件)。
6. 您的中樞和支點帳戶必須由 AWS Organizations 管理。

架構

下圖說明實作的建置區塊。

1. 使用 IAM 根使用者登入資料時， CloudWatch 並將活動分別 CloudTrail 記錄在日誌和追蹤中。
2. 在分支帳戶中， EventBridge 規則會將事件傳送至中樞帳戶中的中央事件匯流排。
3. 在中樞帳戶中， EventBridge 規則會將事件傳送至 Lambda 函數。
4. Lambda 函數使用 Amazon SNS 主題來通知您根使用者活動。

工具

AWS 服務

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- Amazon CloudWatch Logs 可協助您集中管理所有系統、應用程式和 AWS 服務的日誌，以便您可以監控和安全地存檔日誌。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。

其他工具和服務

- Terraform 是一個 CLI 應用程式，用於佈建和管理雲端基礎架構和資源，以組態檔的形式使用程式碼。

代碼存儲庫

此模式的原始程式碼和範本可在GitHub 儲存庫中取得。此模式提供了兩個模板：

- 包含您在 Hub 帳戶中部署的資源的 Terraform 範本
- 您在分支帳戶中部署為堆疊集執行個體的 CloudFormation 範本

存放庫具有以下整體結構。

```
.
 |__README.md
 |__spoke-stackset.yaml
 |__hub.tf
 |__root-activity-monitor-module
     |__main.tf  # contains Terraform code to deploy resources in the Hub account
     |__iam      # contains IAM policies JSON files
         |__ lambda-assume-policy.json          # contains trust policy of the IAM role
 used by the Lambda function
         |__ lambda-policy.json                 # contains the IAM policy attached to
 the IAM role used by the Lambda function
     |__outputs  # contains Lambda function zip code
```

「史詩」區段提供部署範本的 step-by-step 指示。

史诗

將資源部署到 Hub 帳戶

任務	描述	所需技能
複製範例程式碼儲存庫。	1. 開啟 AWS IAM 根使用者活動監控儲存庫。	一般 AWS

任務	描述	所需技能
	2. 在 [程式碼] 索引標籤的檔案清單上方，選擇 [程式碼]，然後複製 HTTPS URL。 3. 在指令行介面中，將工作目錄變更為要儲存範例檔案的位置。 4. 輸入以下命令： `git clone <repoURL>`	

任務	描述	所需技能
更新地形範本。	1. 擷取您的組織 ID。如需指示，請參閱從管理帳戶檢視組織的詳細資料 (AWS Organizations 文件)。 2. 在複製的存放庫中，開啟hub.tf。 3. 使用適合您環境的適當值更新下列項目： • OrganizationId — 新增您的組織 ID。 • SNSTopicName — 為 Amazon SNS 主題添加名稱。 • SNSSubscriptions — 新增應傳送 Amazon SNS 通知的電子郵件。 • Region— 新增要部署資源的 AWS 區域代碼。例如 eu-west-1 。 • Tags— 添加您的標籤。如需詳細資訊，請參閱標記 AWS 資源 (AWS 一般參考)。 4. 儲存並關閉 hub.tf 檔案。	一般 AWS

任務	描述	所需技能
將資源部署到 AWS 中樞帳戶。	1. 在 Terraform 命令列介面中，導覽至複製存放庫的根資料夾，然後輸入下列命令。 `terraform init && terraform plan` 2. 複查輸出並確認您要建立所描述的資源。 3. 輸入以下命令。 `terraform apply` 4. 出現提示時，請輸入以確認部署`yes`。	一般 AWS

將資源部署到您的分支帳戶

任務	描述	所需技能
部署 CloudFormation 範本。	1. 登入 AWS 管理主控台，並開啟 CloudFormation 主控台。 2. 在導覽窗格中，選擇 StackSets。 3. 在StackSets頁面頂端，選擇 [建立] StackSet。 4. 在 [權限] 下，選擇 [服務管理權限] CloudFormation 自動設定部署到 AWS Organizations 管理的目標帳戶所需的許可。	一般 AWS

任務	描述	所需技能
	5. 在先決條件-準備範本下，選擇範本已準備就緒。 6. 在「指定範本」下，選擇「上傳範本檔案」。 7. 選擇 [選擇檔案]，然後在複製的儲存庫中選取`spoke-stackset.yaml` 。 8. 選擇下一步。 9. 在 [指定 StackSet 詳細資料] 頁面上，輸入堆疊集的名稱。 10.在 [參數] 下，輸入 Hub 帳戶的帳戶識別碼，然後選擇 [下一步]。 11.在 [設定 StackSet 選項] 頁面的 [標籤] 下，新增您的標籤。 12.在執行組態下，選擇非作用中，然後選擇下一步。 13.在 [設定部署選項] 頁面上，指定您要部署堆疊集的組織單位和區域，然後選擇 [下一步]。 14.在 [檢閱] 頁面上，選取 [我確認 AWS CloudFormation 可能會建立 IAM 資源]，然後選擇 [提交]。 CloudFormation 開始部署您的堆疊集。 如需詳細資訊和指示，請參閱建立堆疊集 (說明CloudFormation 文件)。	

(選擇性) 測試通知

任務	描述	所需技能
使用根使用者身份證明。	1. 使用根使用者認證登入支點帳戶或 Hub 帳戶。 2. 確認您指定的電子郵件帳戶收到 Amazon SNS 通知。	一般 AWS

相關資源

- 安全性最佳做法 (IAM 文件)
- 使用 StackSets (CloudFormation 文檔)
- 開始使用 (地形文件)

其他資訊

Amazon GuardDuty 是一種持續的安全監控服務，可分析和處理日誌，以識別 AWS 環境中的未預期和潛在未經授權的活動。作為此解決方案的替代方案，如果您已啟用 GuardDuty，它可以在使用 root 使用者認證時提醒您。發 GuardDuty 現項目為`Policy:IAMUser/RootCredentialUsage`，預設嚴重性為「低」。如需詳細資訊，請參閱管理 Amazon GuardDuty 發現項目。

建立 IAM 使用者時傳送通知

由曼西蘇拉特瓦拉（AWS）和塞爾吉·舍甫琴科（AWS）創建

環境：生產	技術：安全、身分識別、合規性；基礎架構	工作負載：所有其他工作
AWS 服務：Amazon SNS；AWS Identity and Access Management；AWS Lambda；Amazon CloudWatch		

Summary

在 Amazon Web Services (AWS) 上，您可以使用此模式部署 AWS CloudFormation 範本，以便在建立 AWS Identity and Access Management (IAM) 使用者時自動接收通知。

使用 IAM，您可以安全地管理 AWS 服務和資源的存取。您可以建立和管理 AWS 使用者和群組，並使用許可允許和拒絕這些使用者和群組存取 AWS 資源。

該 CloudFormation 模板創建一個 Amazon 事件 CloudWatch 事件和一個 AWS Lambda 函數。此事件使 CloudTrail 用 AWS 監控在 AWS 帳戶中建立的任何 IAM 使用者。如果建立使用者，事件 CloudWatch 事件會啟動 Lambda 函數，該函數會傳送 Amazon Simple Notification Service (Amazon SNS) 通知給您，通知您新的使用者建立事件。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 建立和部署的 AWS CloudTrail 追蹤

限制

- AWS CloudFormation 範本必須`CreateUser`僅部署。

架構

目標技術堆疊

- IAM
- AWS CloudTrail
- Amazon CloudWatch 活動
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon SNS

目標架構

自動化和規模

您可以針對不同的 AWS 區域和帳戶多次使用 AWS CloudFormation 範本。您只需在每個區域或帳戶中執行一次。若要自動部署到多個帳戶，請使用 AWS CloudFormation StackSets。該 CloudFormation 模板將能夠在每個帳戶中部署所有必需的資源。

工具

工具

- IAM — AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制 AWS 資源的存取。您可以使用 IAM 來控制能通過身分驗證 (登入) 和授權使用資源的 (具有許可) 的人員。
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 Amazon Web Services 資源的模型和設定，以減少管理這些資源的時間，將更多時間專注於在 AWS 中執行的應用程式。您可以建立一個範本來描述所 CloudFormation 需的所有 AWS 資源，並為您佈建和設定這些資源。
- AWS CloudTrail — AWS 可 CloudTrail 協助您管理 AWS 帳戶的管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。事件包括在 AWS 管理主控台中採取的動作、AWS 命令列界面以及 AWS 開發套件和 API。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供一系統事件 near-real-time 串流，用於描述 AWS 資源的變更。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。

- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Amazon SNS — 亞馬遜簡單通知服務 (Amazon SNS) 是一種受管服務，可使用 Lambda、HTTP、電子郵件、行動推送通知和行動文字訊息 (SMS) 提供訊息交付。

Code

專案的 .zip 檔案可作為附件使用。

史诗

為 Lambda 指令碼建立 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	開啟 Amazon S3 主控台，然後選擇或建立 S3 儲存貯體。此 S3 儲存貯體將託管 Lambda 程式碼 .zip 檔案。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師

將 Lambda 程式碼上傳至 S3 儲存貯體

任務	描述	所需技能
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 .zip 檔案上傳到您定義的 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
部署 CloudFormation 範本。	在 CloudFormation 主控台上，將以附件形式提供的	雲端架構師

任務	描述	所需技能
	CloudFormation `createIAMuser.yaml` 範本部署到此病毒碼。在下一個史詩中，提供模板參數的值。	

完成 CloudFormation 範本中的參數

任務	描述	所需技能
提供 S3 儲存貯體名稱。	輸入您在第一個史詩中建立或選擇的 S3 儲存貯體的名稱。	雲端架構師
提供 S3 金鑰。	在 S3 儲存貯體中提供 Lambda 程式碼 .zip 檔案的位置，而不需要前導斜線 (例如)。`<directory>/<file-name>.zip`	雲端架構師
提供電子郵件地址。	提供使用中的電子郵件地址以接收 Amazon SNS 通知。	雲端架構師
定義記錄層級。	定義 Lambda 函數的記錄層級和頻率。 `Info`指定應用程式進度的詳細資訊訊息。 `Error`指定仍然允許應用程式繼續執行的錯誤事件。 `Warning`指定潛在的有害情況。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	當範本成功部署時，會將訂閱電子郵件訊息傳送至提供的電	雲端架構師

任務	描述	所需技能
	子郵件地址。若要接收通知，您必須確認此電子郵件訂閱。	

相關資源

- 建立系統線
- 建立 S3 儲存貯體
- 將檔案上傳到 S3 儲存貯體
- 部署 CloudFormation 範本
- 建立 IAM 使用者
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用服務控制策略防止在帳戶層級存取網際網路

由塞爾吉·舍甫琴科 (AWS)、肖恩·奧沙利文 (AWS) 和維克多·馬澤奧·惠特克 (AWS) 創建

環境：PoC 或試點	技術：安全性、身分識別、合規性；網路	AWS 服務：AWS Organizations

Summary

Organizations 通常想要限制應保持私有狀態的帳號資源的網際網路存取。在這些帳戶中，虛擬私有雲（VPC）中的資源不應以任何方式訪問互聯網。許多組織都選擇集中式檢測架構。對於集中式檢查架構中的東西向 (VPC 至 VPC) 流量，您需要確定支點帳戶及其資源無法存取網際網路。對於南北 (網際網路輸出和內部部署) 流量，您只想要透過檢查 VPC 允許網際網路存取。

此病毒碼使用服務控制原則 (SCP) 來協助防止網際網路存取。您可以在帳戶或組織單位 (OU) 層級套用此 SCP。SCP 通過防止以下情況來限制互聯網連接：

- 建立或附加允許直接網際網路存取 VPC 的 IPv4 或 IPv6 網際網路閘道
- 建立或接受可能允許透過其他 VPC 間接存取網際網路的 VPC 對等連線
- 建立或更新可能允許直接存取虛擬私人 VPC 資源的網際網路AWS Global Accelerator設定

先決條件和限制

先決條件

- 在中以組織形 AWS 帳戶 式管理一個或多個 AWS Organizations。
- 所有功能均已在中啟用 AWS Organizations。
- 組織中已啟用 SCP。
- 權限：
  - 存取組織的管理帳戶。
  - 建立 SCP。如需最低權限的相關資訊，請參閱建立 SCP。
  - 將 SCP 附加至目標帳戶或組織單位 (OU)。如需有關最低權限的詳細資訊，請參閱附加和卸離服務控制原則。

限制

- SCP 不會影響管理帳戶中的使用者或角色。它們只會影響組織中的成員帳戶。
- SCP 只會影響由屬於組織一部分的帳戶所管理的 AWS Identity and Access Management (IAM) 使用者和角色。如需詳細資訊，請參閱 SCP 對許可的影響。

工具

AWS 服務

- AWS Organizations是一項帳戶管理服務，可協助您 AWS 帳戶 將多個組織整合到您建立並集中管理的組織中。在此模式中，您可以在中使用服務控制策略（SCP）。 AWS Organizations
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您在已定義的虛擬網路中啟動 AWS 資源。此虛擬網路與您在自己的資料中心中操作的傳統網路相似，且具備使用 AWS可擴展基礎設施的優勢。

最佳實務

在您的組織中建立此 SCP 之後，請務必經常更新它，以解決任何可能影響網際網路存取的新功能 AWS 服務 或功能。

史诗

建立並附加 SCP

任務	描述	所需技能
建立 SCP。	1. 登入 AWS Organizations 主控台。您必須登入組織的管理帳戶。 2. 在左窗格中，選擇 [原則]。 3. 在策略頁面上，選擇服務控制策略。 4. 在 Service control policies (服務控制政策) 頁面上，選擇 Create policy (建立政策)。	AWS 管理員

任務	描述	所需技能
	5. 在 [建立新服務控制原則] 頁面上，輸入原則名稱和選擇性原則說明。 6. (選擇性) 將AWS 標籤新增至您的原則。 7. 在 JSON 編輯器中，刪除預留位置原則。 8. 將以下 政策貼到 JSON 編輯器。 `{ "Version": "2012-10-17", "Statement": [{ "Action": ["ec2:AttachInternetGateway", "ec2:CreateInternetGateway", "ec2:CreateVpcPeeringConnection", "ec2:AcceptVpcPeeringConnection", "ec2:CreateEgressOnlyInternetGateway"], "Resource": "*", "Effect": "Deny" }, { "Action": ["globalaccelerator:Create*",`	

任務	描述	所需技能
	<pre>"globalac celerator:Update*"], "Resource": "*", "Effect": "Deny" }] }</pre> 9. 選擇建立政策。	
附加 SCP。	1. 在 [服務控制原則] 頁面上，選擇您建立的原則。 2. 在 Targets (目標) 索引標籤上，選擇 Attach (連接)。 3. 選取您要附加原則的 OU 或帳戶。您可能必須展開 OU，才能尋找您想要的 OU 或帳戶。 4. 選擇連接政策。	AWS 管理員

相關資源

- AWS Organizations 文件
- 服務控制政策 (SCP)
- 使用 AWS 閘道 Load Balancer 進行集中式檢測架構 AWS Transit Gateway (AWS 部落格文章)

使用 git 機密掃描 Git 存儲庫中的敏感信息和安全問題

創建者：索拉巴·辛格 (AWS)

環境：生產	技術：安全性、身分識別、合規	工作負載：開源

Summary

此模式說明如何使用 AWS Labs 的開放原始碼 git-secrets 工具掃描 Git 來源儲存庫，並尋找可能包含敏感資訊的程式碼，例如使用者密碼或 AWS 存取金鑰，或有任何其他安全問題。

`git-secrets`掃描提交，提交消息和合併，以防止敏感信息（例如機密）被添加到您的 Git 存儲庫中。例如，如果提交、提交訊息或合併歷史記錄中的任何提交符合您設定的禁止規則運算式模式之一，則該提交將被拒絕。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 需要安全性掃描的 Git 儲存庫
- 已安裝 Git 用戶端 (版本 2.37.1 及更新版本)

架構

目標架構

- Git
- `git-secrets`

工具

- git-secrets 是一種工具，可以防止您將敏感信息提交到 Git 存儲庫中。

- Git 是一個開源的分佈式版本控制系統。

最佳實務

- 始終通過包含所有修訂版來掃描 Git 存儲庫：

```
git secrets --scan-history
```

史诗

Connect 至 EC2 執行個體

任務	描述	所需技能
使用安全殼層 Connect 至 EC2 執行個體。	使用 SSH 和 key pair 檔案 Connect 到亞馬遜彈性運算雲端 (Amazon EC2) 執行個體。 如果您正在掃描本機電腦上的存放庫，則可以略過此步驟。	一般 AWS

安裝 Git

任務	描述	所需技能
安裝 Git。	使用以下命令安裝 Git： `yum install git -y` 如果您使用的是本機電腦，您可以為特定作業系統版本安裝 Git 用戶端。如需詳細資訊，請參閱 Git 網站。	一般 AWS

克隆源存儲庫並安裝 git 密碼

任務	描述	所需技能
克隆 Git 源代碼存儲庫。	若要複製您要掃描的 Git 儲存庫，請從您的主目錄中選擇 Git 複製指令。	一般 AWS
克隆 git 的秘密。	克隆 `git-secrets` Git 存儲庫。 `git clone https://github.com/awslabs/git-secrets.git` 放置在你的`git-secrets` 某個地方，`PATH` 以便 Git 在運行時拿起它`git-secrets` 。	一般 AWS
安裝 git 秘密。	對於 Unix 和變體（Linux / macOS）： 您可以使用 `Makefile` (在`git-secrets` 存放庫中提供) 的`install`目標來安裝工具。您可以使用`PREFIX`和`MANPREFIX` 變數來自訂安裝路徑。 `make install` 用於 Windows： 執行`git-secrets` 儲存庫中提供的 PowerShell `install.ps1` 指令碼。此指令碼會將安裝檔案複製到安裝目錄 (`%USERPROFILE`	一般 AWS

任務	描述	所需技能
	%/.git-secrets 依預設)，並將目錄新增至目前的使用者 PATH。 `PS > ./install.ps1` 對於自製軟件（macOS 用戶）： 執行： `brew install git-secrets` 如需詳細資訊，請參閱相關資源一節。	

掃描 git 代碼存儲庫

任務	描述	所需技能
轉到源存儲庫。	切換到您要掃描的 Git 儲存庫的目錄： `cd my-git-repository`	一般 AWS
註冊 AWS 規則集 (Git 掛接)。	要配置git-secrets 為在每次提交時掃描 Git 存儲庫，請運行以下命令： `git secrets --register-aws`	一般 AWS
掃描存放庫。	執行下列命令以開始掃描存放庫：	一般 AWS

任務	描述	所需技能
	`git secrets --scan`	

任務	描述	所需技能
檢閱輸出檔案。	如果該工具在 Git 存儲庫中發現漏洞，則會生成一個輸出文件。例如： `example.sh:4:AWS_SECRET_ACCESS_KEY = *********` `[ERROR] Matched one or more prohibited patterns` `Possible mitigations:` `- Mark false positives as allowed using: git config --add secrets.allowed ...` `- Mark false positives as allowed by adding regular expressions to .gitallowed at repository's root directory` `- List your configured patterns: git config --get-all secrets.patterns` `- List your configured allowed patterns: git config --get-all secrets.allowed` `- List your configured allowed patterns in .gitallowed at repository's root directory` `- Use --no-verify if this is a one-time false positive`	一般 AWS

相關資源

- 使用 AWS 服務的 Git 網路掛鉤 (AWS 快速入門)
- Git 秘密工具
- 將 Git 儲存庫遷移到 AWS (AWS 實作教學)
- AWS CodeCommit API 參考

將提醒從 AWS Network Firewall 傳送到 Slack 通道

創建者：文基斯里瓦薩夫（AWS）和芳香拉吉傑亞拉揚（AWS）

代碼存儲庫：NfwSlackIntegration	環境：PoC 或試點	技術：安全性、身分識別、合規性；網路
AWS 服務：AWS Lambda；AWS Network Firewall；Amazon S3		

Summary

此模式說明如何使用 Amazon Web Services (AWS) Network Firewall 搭配分散式部署模型來部署防火牆，以及如何將 AWS Network Firewall 產生的警示傳播到可設定的 Slack 通道。

支付卡產業資料安全標準 (PCI DSS) 等合規標準要求您安裝並維護防火牆以保護客戶資料。在 AWS 雲端中，虛擬私有雲端 (VPC) 在符合這些合規要求的情況下被視為與實體網路相同。您可以使用 Network Firewall 監控 VPC 之間的網路流量，並保護在受合規標準管理的 VPC 中執行的工作負載。Network Firewall 會在偵測到來自同一帳戶中其他 VPC 的未經授權存取時，封鎖存取或產生警示。但是，Network Firewall 支援有限數量的目的地來傳送警示。這些目的地包括 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體、Amazon CloudWatch 日誌群組和亞馬遜資料 Firehose 交付串流。對這些通知進行任何進一步的動作都需要使用 Amazon 雅典娜或 Amazon Kinesis 進行離線分析。

此模式提供了一種方法，可將 Network Firewall 產生的警示傳播到可設定的 Slack 通道，以便近乎即時執行進一步的動作。您還可以將功能擴展到其他警報機制 PagerDuty，例如 Jira 和電子郵件。（這些自定義超出此模式的範圍。）

先決條件和限制

先決條件

- Slack 頻道 (請參閱 Slack 說明中心的開始使用)
- 傳送訊息至頻道所需的權限
- 帶有 API 令牌的 Slack 端點 URL（選擇您的應用程序並選擇傳入的網絡鉤子以查看其 URL；有關更多信息，請參閱 Slack API 文檔中的創建傳入 Webhook）
- 工作負載子網路中的 Amazon 彈性運算雲端 (Amazon EC2) 測試執行個體

- 在 Network Firewall 中測試規則
- 觸發測試規則的實際或模擬流量
- 用於保存要部署的源文件的 S3 存儲桶

限制

- 目前，此解決方案僅支援單一無類別網域間路由 (CIDR) 範圍，做為來源和目標 IP 的篩選器。

架構

目標技術堆疊

- 一台 VPC
- 四個子網路 (兩個用於防火牆，兩個用於工作負載)
- 網際網路閘道
- 四個路由表與規則
- S3 儲存貯體做為警示目的地，透過儲存貯體政策和事件設定進行設定以執行 Lambda 函數
- 具有執行角色的 Lambda 函數，用於傳送 Slack 通知
- AWS Secrets Manager 用於存儲鬆弛網址的秘密
- 具有警示組態的網路防火
- Slack 頻道

除了 Slack 通道以外的所有元件，均由此模式提供的 CloudFormation 範本和 Lambda 函數佈建 (請參閱「程式碼」一節)。

目標架構

此模式會建立具有 Slack 整合功能的去中心化網路防火牆。此架構由具有兩個可用區域的 VPC 組成。VPC 包括兩個受保護的子網路和兩個具有網路防火牆端點的防火牆子網路。透過建立防火牆策略和規則，可以監控進出受保護子網路的所有流量。網路防火牆設定為將所有警示放置在 S3 儲存貯體中。此 S3 儲存貯體設定為在收到put事件時呼叫 Lambda 函數。Lambda 函數會從機 Secrets Manager 擷取已設定的 Slack URL，並將通知訊息傳送至 Slack 工作區。

如需有關此架構的詳細資訊，請參閱 AWS 部落格文章 AWS Network Firewall 的部署模型。

工具

AWS 服務

- AWS Network Firewall 是適用於 AWS 雲端中 VPC 的可設定狀態、受管網路防火牆以及入侵偵測與防護服務。您可以使用 Network Firewall 來篩選 VPC 周邊的流量，並保護 AWS 上的工作負載。
- AWS Secrets Manager 是一種用於登入資料儲存和擷取的服務。使用 Secrets Manager，您可以透過 API 呼叫 Secret Secrets Manager 來取代程式碼中的硬式編碼認證 (包括密碼)，以程式設計方式擷取密碼。此模式使用 Secrets Manager 來存儲 Slack URL。
- Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。此模式使用 Amazon S3 存儲 Lambda 函數的 CloudFormation 模板和 Python 腳本。它也使用 S3 儲存貯體做為網路防火牆警示目的地。
- AWS 可 CloudFormation協助您建立 AWS 資源的模型和設定、快速且一致地佈建它們，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。此模式使用 AWS 自動 CloudFormation 為 Firewall Manager 員部署分散式架構。

Code

此病毒碼的程式碼可在 GitHub Network Firewall Slack 整合存放庫中取得。在存儲庫的`src` 文件夾中，您會發現：

- YAML 格式的一組 CloudFormation 檔案。您可以使用這些範本來佈建此樣式的元件。
- 一個 Python 源文件（`slack-lambda.py`）來創建 Lambda 函數。
- 用來上傳 Lambda 函數程式碼的 .zip 封存檔部署套件 (`slack-lambda.py.zip`)。

若要使用這些檔案，請遵循下一節中的指示。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
建立 S3 儲存貯體。	1. 登入 AWS 管理主控台，然後前往 https://console.a	應用程式開發人員、應用程式擁有者、

任務	描述	所需技能
	ws.amazon.com/s3/ 開啟 Amazon S3 主控台。 2. 選擇或建立 S3 儲存貯體來託管程式碼。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。S3 儲存貯體名稱不能包含前導斜線。我們建議您使用前置詞來組織此模式的程式碼。 如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。	
上傳 CloudFormation 範本和 Lambda 程式碼。	1. 從GitHub 存放庫下載此病毒碼的下列檔案： • `base.yml` • `igw-ingress-route.yml` • `slack-lambda.py` • `slackLambda.yml` • `decentralized-deployment.yml` • `protected-subnet-route.yml` • `slack-lambda.py.zip` 2. 將檔案上傳到您建立的 S3 儲存貯體。	應用程式開發人員、應用程式擁有者、

部署 CloudFormation 範本

任務	描述	所需技能
啟動 CloudFormation 範本。	在與 S3 儲存貯體相同的 CloudFormation AWS 區域中開啟 AWS 主控台，然後部署範本`base.yml`。此範本會建立必要的 AWS 資源和 Lambda 函數，以便將警示傳輸到 Slack 通道。 如需部署 CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的在 AWS CloudFormation 主控台建立堆疊。	應用程式開發人員、應用程式擁有者、
完成範本中的參數。	指定堆疊名稱並設定參數值。如需參數、其說明和預設值的清單，請CloudFormation參閱其他資訊一節中的參數。	應用程式開發人員、應用程式擁有者、
建立堆疊。	1. 檢閱堆疊詳細資料並根據您的環境需求更新值。 2. 選擇 [建立堆疊] 以部署範本。	應用程式開發人員、應用程式擁有者、

驗證解決方案

任務	描述	所需技能
測試部署。	使用 AWS CloudFormation 主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 確認已建立 Target 技術堆疊區段中列出的資源。	應用程式開發人員、應用程式擁有者、

任務	描述	所需技能
	如果 CloudFormation 範本無法成功部署，請檢查您為`pAvailabilityZone1` 和`pAvailabilityZone2` 參數提供的值。這些應該適用於您要部署解決方案的 AWS 區域。如需每個區域的可用區域清單，請參閱 Amazon EC2 文件中的區域和區域。	

任務	描述	所需技能
測試功能。	1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。 2. 在其中一個受保護的子網路中建立 EC2 執行個體。選擇一個 Amazon Linux 2 AMI (HVM) 作為 HTTPS 伺服器使用。如需指示，請參閱 Amazon EC2 文件中的啟動執行個體。 3. 使用下列使用者資料在 EC2 執行個體上安裝 Web 伺服器： `#!/bin/bash` `yum install httpd -y` `systemctl start httpd` `systemctl stop firewalld` `cd /var/www/html` `echo "Hello!! this is a NFW alert test page, 200 OK" > index.html` 4. 建立下列網路防火牆規則： 無狀態規則： `Source: 0.0.0.0/0` `Destination 10.0.3.65/32 (private IP of the EC2 instance)` `Action: Forward` 可設定狀態規則： `Protocol: HTTP`	應用程式開發人員、應用程式擁有者、

任務	描述	所需技能
	`Source ip/port: Any / Any` `Destination ip/port: Any /Any` 5. 獲取您在步驟 3 中創建的 Web 服務器的公共 IP。 6. 在瀏覽器中存取公用 IP。您應該會在瀏覽器中看到下列訊息： `Hello!! this is a NFW alert test page, 200 OK` 您也會在 Slack 頻道中收到通知。通知可能會延遲，具體取決於郵件的大小。出於測試目的，請考慮提供不太窄的 CIDR 篩選器（例如，帶有 /32 的 CIDR 值將被視為太窄，而 /8 會太寬）。如需詳細資訊，請參閱其他資訊中的篩選行為一節。	

相關資源

- AWS Network Firewall 的部署模型 (AWS 部落格文章)
- AWS Network Firewall 政策 (AWS 文件)
- Network Firewall 鬆弛整合 (GitHub 儲存庫)
- 建立鬆弛工作區 (Slack 說明中心)

其他資訊

CloudFormation 參數

參數	描述	預設值或範例值
pVpcName	要建立的 VPC 名稱。	檢驗
pVpcCidr	要建立之 VPC 的 CIDR 範圍。	10.0.0.0/16
pVpcInstanceTenancy	EC2 執行個體如何在實體硬體上分散。選項有 default (共用租賃) 或 dedicated (單一租賃)。	預設
pAvailabilityZone1	基礎結構的第一個可用區域。	美國東部 2a
pAvailabilityZone2	基礎結構的第二個可用區域。	美國東部 2B
pNetworkFirewallSubnet1Cidr	第一個防火牆子網路的 CIDR 範圍 (最小值 /28)。	10.0.1.0/24
pNetworkFirewallSubnet2Cidr	第二個防火牆子網路的 CIDR 範圍 (最小值 /28)。	10.0.2.0/24
pProtectedSubnet1Cidr	第一個受保護 (工作負載) 子網路的 CIDR 範圍。	10.0.3.0/24
pProtectedSubnet2Cidr	第二個受保護 (工作負載) 子網路的 CIDR 範圍。	10.0.4.0/24
pS3BucketName	您上傳 Lambda 原始程式碼所在之現有 S3 儲存貯體的名稱。	我們-w2-yourname-lambda-functions
pS3KeyPrefix	您在其中上傳 Lambda 原始程式碼的 S3 儲存貯體的前置詞。	AOD 測試
pAWSSecretName4Slack	保留 Slack 網址的密碼名稱。	SlackEnpoint-氟氯化碳

pSlackChannelName	您建立的 Slack 頻道名稱。	意大利通知
pSlackUserName	鬆弛的使用者名稱。	鬆弛用戶
pSecretKey	這可以是任何鍵。我們建議您使用預設值。	Web 掛鉤網址
pWebHookUrl	鬆弛網址的值。	https://hooks.slack.com/services/T???9T??/A031885JRM7/9D4Y?????????
pAlertS3Bucket	用作網路防火牆警示目的地的 S3 儲存貯體名稱。系統會為您建立此儲存貯體。	我們-w2-yourname-security-aod-alerts
pSecretTagName	密碼的標籤名稱。	AppName
pSecretTagValue	指定標籤名稱的標籤值。	LambdaSlackIntegration
pdestCidr	目標 CIDR 範圍的篩選器。如需詳細資訊，請參閱下一節「篩選行為」。	10.0.0.0/16
pdestCondition	用來指出要排除還是包含目的地相符項目的地的旗標。如需詳細資訊，請參閱下一節。有效值為 `include` 和 `exclude`。	包含
psrcCidr	要警示的來源 CIDR 範圍的篩選器。如需詳細資訊，請參閱下一節。	118.2.0.0/16
psrcCondition	要排除或包含來源相符項目的旗標。如需詳細資訊，請參閱下一節。	包含

過濾器行為

如果您尚未在 AWS Lambda 中設定任何篩選器，所有產生的警示都會傳送到您的 Slack 頻道。產生警示的來源和目標 IP 會與您在部署範 CloudFormation 本時設定的 CIDR 範圍進行比對。如果找到相符項目，則會套用條件。如果來源或目的地位於設定的 CIDR 範圍內，且其中至少有一個已設定條件`include`，則會產生警示。下表提供 CIDR 值、條件和結果的範例。

	已配置的 CIDR	警示 IP 位址	Configured	警示
來源	10.0.0.0/16	10.0.0.25	包含	是
目的地	100.0.0/16	202.0.0.13	包含	

	已配置的 CIDR	警示 IP 位址	Configured	警示
來源	10.0.0.0/16	10.0.0.25	排除	否
目的地	100.0.0/16	202.0.0.13	包含	

	已配置的 CIDR	警示 IP 位址	Configured	警示
來源	10.0.0.0/16	10.0.0.25	包含	是
目的地	100.0.0/16	100.0.13	包含	

	已配置的 CIDR	警示 IP 位址	Configured	警示
來源	10.0.0.0/16	90.0.0.25	包含	是
目的地	Null	202.0.0.13	包含	

	已配置的 CIDR	警示 IP 位址	Configured	警示
來源	10.0.0.0/16	90.0.0.25	包含	否
目的地	100.0.0/16	202.0.0.13	包含	

使用 AWS 私有 CA 和 AWS 記憶體簡化私有憑證管理

由埃弗里特興克利（AWS）和維韋克·戈亞爾（AWS）創建

程式碼儲存庫:ACMPCA 階層	環境：生產	技術：安全性、身分識別、合規性、基礎架構、移轉
AWS 服務：AWS Certificate Manager (ACM)；AWS Organizations；AWS 記憶體		

Summary

您可以使用 AWS 私有憑證授權單位 (AWS Private CA) 發行私有憑證，以驗證內部資源和簽署電腦程式碼。此模式提供 AWS CloudFormation 範本，可快速部署多層 CA 階層和一致的佈建體驗。或者，您可以使用 AWS Resource Access Manager (AWS RAM) 在 AWS Organizations 的組織或組織單位 (OU) 內安全地共用 CA，並在使用 AWS RAM 管理許可時集中 CA。每個帳戶都不需要私人 CA，因此這種方法可以為您節省金錢。此外，您可以使用 Amazon Simple Storage Service (Amazon S3) 來存放憑證撤銷清單 (CRL) 和存取日誌。

此實作提供下列功能與優點：

- 使用 AWS 私有 CA 集中並簡化私有 CA 階層的管理。
- 將憑證和金鑰匯出到 AWS 和內部部署的客戶管理裝置。
- 使用 AWS CloudFormation 範本進行快速部署和一致的佈建體驗。
- 創建一個私有根 CA 與 1，2，3 或 4 從屬 CA 層次結構一起。
- 選擇性地使用 AWS RAM 與組織或 OU 層級的其他帳戶共用終端實體從屬 CA。
- 透過使用 AWS RAM，省去每個帳戶中使用私有 CA 的需求，進而節省成本。
- 為 CRL 建立選用的 S3 儲存貯體。
- 為 CRL 存取日誌建立選用的 S3 儲存貯體。

先決條件和限制

先決條件

如果您想要在 AWS Organizations 結構內共用 CA，請識別或設定下列項目：

- 用於建立 CA 階層和共用的安全性帳戶。
- 用於測試的個別 OU 或帳戶。
- 在 AWS Organizations 管理帳戶中啟用共用功能。如需詳細資訊，請參閱 AWS RAM 文件中的啟用 AWS Organizations 內的資源共用。

限制

- CA 是區域資源。所有 CA 都位於單一 AWS 帳戶和單一 AWS 區域中。
- 不支援使用者產生的憑證和金鑰。對於此使用案例，建議您自訂此解決方案以使用外部根 CA。
- 不支援公用 CRL 值區。我們建議您將 CRL 保持私密。如果需要網際網路存取 CRL，請參閱 AWS 私有 CA 文件中關於啟用 S3 區塊公用存取 (BPA) 功能中關於使用 Amazon CloudFront 提供 CRL 的章節。
- 這種模式實現了一個單一區域的方法。如果您需要多區域憑證授權單位，可以在第二個 AWS 區域或現場部署實作下屬。這種複雜性不在此模式的範圍之外，因為實作取決於您的特定使用案例、工作負載量、相依性和需求。

架構

目標技術堆疊

- AWS 私有 CA
- AWS RAM
- Amazon S3
- AWS Organizations
- AWS CloudFormation

目標架構

此模式提供兩個共用給 AWS Organizations 的選項：

選項 1 – 在組織層級建立共用。組織中的所有帳戶都可以使用共用 CA 來發行私有憑證，如下圖所示。

選項 2 – 在組織單位 (OU) 層級建立共用。只有指定 OU 中的帳戶可以使用共用 CA 來發行私有憑證。例如，在下圖中，如果共用是在沙箱 OU 層級建立，則開發人員 1 和開發人員 2 都可以使用共用 CA 來發行私有憑證。

工具

AWS 服務

- AWS 私有 CA — AWS 私有憑證授權單位 (AWS Private CA) 是一種託管的私有 CA 服務，用於發行和撤銷私有數位憑證。它可協助您建立私有 CA 階層，包括根 CA 和從屬 CA，而不需要操作內部部署 CA 的投資和維護成本。
- AWS RAM — AWS Resource Access Manager (AWS RAM) 可協助您在 AWS 帳戶以及組織內部或 AWS Organizations 的 OU 中安全地共用資源。為了減少多帳戶環境中的營運開銷，您可以建立資源並使用 AWS RAM 跨帳戶共用該資源。
- AWS Organizations — AWS Organizations 是一種帳戶管理服務，可讓您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。此模式使用 Amazon S3 存放憑證撤銷清單 (CRL) 和存取日誌。
- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。此模式使用 AWS CloudFormation 自動部署多層 CA 階層。

Code

此模式的原始程式碼可在 GitHub AWS 私有 CA 階層存放庫中取得。該存儲庫包括：

- AWS CloudFormation 範本`ACMPCA-RootCASubCA.yaml`。您可以使用此範本來部署此實作的 CA 階層。
- 針對使用案例的測試檔案，例如要求、匯出、描述和刪除憑證。

若要使用這些檔案，請依照 Epics 一節中的指示操作。

史诗

架構 CA 階層

任務	描述	所需技能
收集憑證主旨資訊。	收集有關憑證擁有者的憑證主體資訊：組織名稱、組織單位、國家、州、地區和一般名稱。	雲端架構師、安全架構師、PKI 工程師
收集有關 AWS Organizations 的選用資訊。	如果 CA 將成為 AWS Organizations 結構的一部分，而您想要在該結構內共用 CA 階層，請收集管理帳戶編號、組織 ID，以及選擇性地收集 OU ID (如果您只想與特定 OU 共用 CA 階層)。此外，請決定您要與其共用 CA 的 AWS Organizations 帳戶或 OU (如果有的話)。	雲端架構師、安全架構師、PKI 工程師
設計 CA 階層。	決定哪個帳戶將容納根 CA 和下屬 CA。決定根憑證與最終實體憑證之間的階層需要多少從屬層級。如需詳細資訊，請參閱 AWS 私有 CA 文件中的設計 CA 階層。	雲端架構師、安全架構師、PKI 工程師
決定 CA 階層的命名和標記慣例。	判斷 AWS 資源的名稱：根 CA 和每個下屬 CA。決定應將哪些標籤指派給每個 CA。	雲端架構師、安全架構師、PKI 工程師
判斷所需的加密和簽章演算法。	決定下列項目： • 您的組織對 CA 發行憑證時所使用之公開金鑰的加密演算法需求。預設值為 RSA。	雲端架構師、安全架構師、PKI 工程師

任務	描述	所需技能
	• CA 用於憑證簽署的金鑰演算法。預設值為 SHA256WITHRSA。	
決定 CA 階層的憑證撤銷需求。	如果需要憑證撤銷功能，請為包含憑證撤銷清單 (CRL) 的 S3 儲存貯體建立命名慣例。	雲端架構師、安全架構師、PKI 工程師
決定 CA 階層的記錄需求。	如果需要存取記錄功能，請為包含存取日誌的 S3 儲存貯體建立命名慣例。	雲端架構師、安全架構師、PKI 工程師
決定憑證到期時間。	判斷根憑證 (預設值為 10 年)、終端實體憑證 (預設值為 13 個月) 和附屬 CA 憑證 (預設值為 3 年) 的到期日。從屬 CA 憑證應該比階層中較高層級的 CA 憑證早到期。如需詳細資訊，請參閱 AWS 私有 CA 文件中的管理私有 CA 生命週期。	雲端架構師、安全架構師、PKI 工程師

部署 CA 階層

任務	描述	所需技能
完成 事前準備。	完成此模式「必要條件」一節中的步驟。	雲端管理員、安全工程師、PKI 工程師
為各種角色建立 CA 角色。	1. 決定 AWS IAM 身分中心 (AWS Single Sign-On 的後續任務) 中管理 CA 階層的各種層級所需的 AWS Identity and Access Management (IAM) 角色或使用者類型，例如	雲端管理員、安全工程師、PKI 工程師

任務	描述	所需技能
	RootcaAdmin、下屬管理員和。 CertificateConsumer 2. 確定分開職責所需的原則細微性。 3. 在 CA 階層所在的帳戶中，在 IAM 身分中心建立必要的 IAM 角色或使用者。	
部署 CloudFormation 堆疊。	1. 從此模式的GitHub 儲存庫中，下載 AWSPCA-RootCasubca.YAML 範本。 2. 從 AWS CloudFormation 主控台或 AWS Command Line Interface (AWS CLI) (AWS CLI) 部署範本。如需詳細資訊，請參閱CloudFormation 文件中的使用堆疊。 3. 完成範本中的參數，包括組織名稱、OU 名稱、金鑰演算法、簽署演算法及其他選項。	雲端管理員、安全工程師、PKI 工程師

任務	描述	所需技能
架構更新使用者管理資源所使用之憑證的解決方案。	整合式 AWS 服務的資源 (例如 Elastic Load Balancing) 會在到期前自動更新憑證。但是，使用者管理的資源 (例如在 Amazon 彈性運算雲端 (Amazon EC2) 執行個體上執行的網頁伺服器，則需要另一種機制。 1. 判斷哪些使用者管理的資源需要私有 CA 的終端實體憑證。 2. 規劃要通知使用者管理的資源和憑證到期的程序。如需範例，請參閱下方： • 使用 AWS Config 受管規則 • 使用 Amazon CloudWatch 和 Amazon EventBridge 3. 撰寫自訂指令碼以更新使用者受管資源上的憑證，並將其與 AWS 服務整合以自動化更新。如需整合式 AWS 服務的詳細資訊，請參閱 ACM 文件中的與 AWS Certificate Manager 整合的服務。	雲端管理員、安全工程師、PKI 工程師

驗證並記錄 CA 階層

任務	描述	所需技能
驗證選用的 AWS 記憶體共用。	如果 CA 階層與 AWS Organizations 中的其他帳戶共用，請從 AWS 管理主控台登入其中一個帳戶，導覽至 AWS Private CA 主控台，並確認新建立的 CA 已與此帳戶共用。只有階層中最低層級的 CA 才會顯示，因為這是產生最終實體憑證的 CA。對與 CA 共用的帳戶重複此動作。	雲端管理員、安全工程師、PKI 工程師
使用憑證生命週期測試驗證 CA 階層。	在此模式的GitHub 存放庫中，找出生命週期測試。從 AWS CLI 執行測試以請求憑證、匯出憑證、描述憑證，以及刪除憑證。	雲端管理員、安全工程師、PKI 工程師
將憑證鏈結匯入信任存放區。	若要讓瀏覽器和其他應用程式信任憑證，憑證的簽發者必須包含在瀏覽器的信任存放區 (信任 CA 清單) 中。將新 CA 階層的憑證鏈結新增至瀏覽器和應用程式的信任存放區。確認最終實體憑證受信任。	雲端管理員、安全工程師、PKI 工程師
創建一個工作手冊來記錄 CA 層次結構。	建立 runbook 文件以描述 CA 階層架構、可要求最終實體憑證的帳戶結構、建置程序，以及基本管理工作，例如發出最終實體憑證 (除非您想要允許子帳戶自助服務)、使用和追蹤。	雲端管理員、安全工程師、PKI 工程師

相關資源

- 設計 CA 階層 (AWS 私有 CA 文件)
- 建立私有 CA (AWS 私有 CA 文件)
- 如何使用 AWS 記憶體分享您的 AWS 私有 CA 跨帳戶 (AWS 部落格文章)
- AWS 私有 CA 最佳實務 (AWS 部落格文章)
- 在 AWS Organizations 內啟用資源共用 (AWS RAM 文件)
- 管理私有 CA 生命週期 (AWS 私有 CA 文件)
- acm-certificate-expiration-check 適用於 AWS Config (AWS Config 文件)
- AWS Certificate Manager 現在可透過 Amazon 提供憑證到期監控 CloudWatch (AWS 公告)
- 與 AWS Certificate Manager 整合的服務 (ACM 文件)

其他資訊

匯出憑證時，請使用強式密碼編譯的複雜密碼，並符合組織的資料外洩防護策略。

在多帳戶環境中，關閉所有 Security Hub 成員帳戶的安全性標準控制

由邁克爾·富爾比爾（AWS）和艾哈邁德·巴克里（AWS）創建

環境：生產	技術：安全性、身分識別、合規性；無伺服器	AWS 服務：Amazon、亞馬遜、AWS Lambda EventBridge、AWS Security Hub、AWS Step Functions

Summary

重要事項：AWS Security Hub 現在支援跨帳戶的安全標準和控制的中央組態。這項新功能可解決此 APG 模式中解決方案所涵蓋的許多案例。在您以此模式部署解決方案之前，請參閱 Security Hub 中的中央組態。

在 Amazon Web Services (AWS) 雲端中，AWS Security Hub 標準控制 (例如 CIS AWS 基準測試或 AWS 基礎安全最佳實務) 只能從單一 AWS 帳戶手動關閉 (停用)。在多帳戶環境中，您無法透過「按一下」(也就是一個 API 呼叫) 來關閉多個 Security Hub 成員帳戶之間的控制項。此模式示範如何使用按一下來關閉 Security Hub 系統管理員帳戶所管理之所有 Security Hub 成員帳戶的 Security Hub 標準控制項。

先決條件和限制

先決條件

- 由管理多個成員帳戶的 Security Hub 管理員帳戶組成的多帳戶環境
- 已安裝 AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版
- 已安裝 AWS 無伺服器應用程式模型命令列界面 (AWS SAM CLI)

限制

- 此模式僅適用於單一 Security Hub 系統管理員帳戶管理多個成員帳戶的多帳戶環境。

- 如果您在很短的時間內變更許多控制項，則事件初始化會導致多個 parallel 呼叫。這可能會導致 API 節流，並導致呼叫失敗。例如，如果您使用 Security Hub 控制項 CLI 以程式設計方式變更許多控制項，就會發生這種情況。

架構

目標技術堆疊

- Amazon DynamoDB
- Amazon EventBridge
- AWS CLI
- AWS Lambda
- AWS 山 CLI
- AWS Security Hub
- AWS Step Functions

目標架構

下圖顯示「Step Functions」工作流程的範例，該工作流程會關閉多個 Security Hub 成員帳戶 (從 Security Hub 系統管理員帳戶檢視) 之間的 Security Hub 標準控制項。

圖表包括下列工作流程：

1. EventBridge 規則會根據每日排程啟動，並呼叫狀態機器。您可以透過更新 AWS CloudFormation 範本中的 S chedu le 參數來修改規則的時間。
2. 每當 Security Hub 系統管理員帳戶中的控制項開啟或關閉時，就會啟動 EventBridge 規則。
3. Step Functions 狀態機器會將安全性標準控制項 (亦即開啟或關閉的控制項) 的狀態從 Security Hub 系統管理員帳戶傳播至成員帳戶。
4. 跨帳戶 AWS Identity and Access Management (IAM) 角色會部署在每個成員帳戶中，並由狀態機器承擔。狀態機打開或關閉每個成員帳戶中的控件。
5. DynamoDB 表包含例外狀況和資訊，說明要在特定帳戶中開啟或關閉哪些控制項。此資訊會覆寫從 Security Hub 系統管理員帳戶針對指定成員帳戶擷取的組態。

備註：排程 EventBridge 規則的目的是確保新增的 Security Hub 成員帳戶與現有帳戶具有相同的控制狀態。

工具

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Serverless Application Model (AWS SAM) 是一種開放原始碼架構，可協助您在 AWS 雲端建置無伺服器應用程式。
- AWS Security Hub 提供您在 AWS 中安全狀態的全面檢視。它也可協助您根據安全產業標準和最佳實務來檢查 AWS 環境。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務，以建立關鍵業務應用程式。

Code

此模式的程式碼可在 GitHub AWS Security Hub 跨帳戶控制停用程式存放庫中取得。代碼存儲庫包含以下文件和文件夾：

- `UpdateMembers/template.yaml`— 此檔案包含部署在 Security Hub 系統管理員帳戶中的元件，包括 Step Functions 狀態機器和 EventBridge 規則。
- `member-iam-role/template.yaml`— 此檔案包含在成員帳戶中部署跨帳戶 IAM 角色的程式碼。
- `stateMachine.json`— 此檔案定義狀態機的工作流程。
- `GetMembers/index.py`— 此檔案包含狀GetMembers態機器的程式碼。指令碼會擷取所有現有 Security Hub 成員帳戶中的安全性標準控制項狀態。
- `UpdateMember/index.py`— 此檔案包含一個指令碼，可更新每個成員帳戶中的控制狀態。
- `CheckResult/index.py`— 此檔案包含一個指令碼，可檢查工作流程呼叫的狀態 (已接受或失敗)。

史诗

在安全中心成員帳戶中部署跨帳戶 IAM 角色

任務	描述	所需技能
識別 Security Hub 系統管理員帳戶的帳戶識別碼。	設定 Security Hub 系統管理員帳戶，然後記下系統管理員帳戶的帳戶識別碼。	雲端架構師
在成員帳戶中部署包含跨帳戶 IAM 角色的 CloudFormation 範本。	若要在 Security Hub 系統管理員帳戶所管理的所有成員帳戶中部署`member-iam-role/template.yaml` 範本，請執行下列命令： `aws cloudformation deploy --template-file member-iam-role/template.yaml --capabilities CAPABILITY_NAMED_IAM --stack-name <your-stack-name> --parameter-overrides SecurityHubAdminAccountId=<your-account-ID>` 此`SecurityHubAdminAccountId` 參數必須符合您先前所述的 Security Hub 系統管理員帳戶識別碼。	AWS DevOps

在安全中心系統管理員帳戶中部署狀態機器

任務	描述	所需技能
使用 AWS SAM Package 包含狀態機器的 CloudFormation 範本。	若要封裝資訊安全中心系統管理員帳戶中的UpdateMembers/template.yaml 範本，請執行下列命令： `sam package --template-file UpdateMembers/template.yaml --output-template-file UpdateMembers/template-out.yaml --s3-bucket <your-s3-bucket-name>` 注意：您的 Amazon Simple Storage Service (Amazon S3) 儲存貯體必須位於部署 CloudFormation 範本的相同 AWS 區域中。	AWS DevOps
在安全中心系統管理員帳戶中部署封裝的 CloudFormation 範本。	若要在安全中心系統管理員帳戶中部署 CloudFormation 範本，請執行下列命令： `aws cloudformation deploy --template-file UpdateMembers/template-out.yaml --capabilities CAPABILITY_IAM --stack-name <your-stack-name>` 在**member-iam-role/template.yaml** 範本中，MembRream 參數必須與	AWS DevOps

任務	描述	所需技能
	IAM RolePath RolePath 參數相符，且 MEMBriam 必須與身分存取及存取管理系統相符。RoleName RoleName 備註：由於 Security Hub 是一項區域服務，因此您必須在每個 AWS 區域中個別部署範本。請務必先將解決方案封裝至每個區域的 S3 儲存貯體。	

相關資源

- 指定安全中心管理員帳戶 (AWS Security Hub 文件)
- 處理錯誤、重試和新增警示至步驟函式狀態機器執行 (AWS 部落格文章)

使用以下方式從 AWS IAM 身分中心更新 AWS CLI 登入資料 PowerShell

由乍得哩程（AWS）和安迪·鮑恩（AWS）創建

環境：生產	技術：安全性、身分識別、合規性；雲端原生	工作負載：開源
AWS 服務：適用於的 AWS 工具 PowerShell；AWS IAM 身分中心		

Summary

如果您想要使用 AWS IAM 身分中心 (AWS Single Sign-On 的後續任務) 登入資料搭配 AWS Command Line Interface (AWS CLI) (AWS CLI)、AWS 開發套件或 AWS Cloud Development Kit (AWS CDK)，您通常必須將登入資料從 IAM 身分中心主控台複製並貼到命令列界面。此過程可能需要相當長的時間，並且必須對每個需要訪問權限的帳戶重複執行此過程。

一個常見的解決方案是使用 AWS CLI `aws sso configure` 命令。此命令會將啟用 IAM 身分中心的設定檔新增到您的 AWS CLI 或 AWS 開發套件。不過，此解決方案的缺點是您必須針`aws sso login`對您以此方式設定的每個 AWS CLI 設定檔或帳戶執行命令。

作為替代解決方案，此模式描述如何使用 AWS CLI 命名的設定檔和 AWS Tools PowerShell 來同時從單一 IAM 身分中心執行個體存放和重新整理多個帳戶的登入資料。此指令碼也會將 IAM 身分中心工作階段資料儲存在記憶體中，以便重新整理登入資料，而無需再次登入 IAM

先決條件和限制

先決條件

- PowerShell，已安裝並配置。如需詳細資訊，請參閱安裝 PowerShell (Microsoft 說明文件)。
- 適用於 PowerShell、安裝和設定的 AWS 工具。基於效能原因，我們強烈建議您安裝 AWS Tools 的模組化版本 PowerShell，稱為. `AWS.Tools` 每個 AWS 服務都有自己的個別小模組支援。在 PowerShell 提示中，輸入下列指令以安裝此模式所需的模組：`AWS.Tools.InstallerSSO`、和`SSOIDC`。

```
Install-Module AWS.Tools.Installer
Install-AWSToolsModule SSO, SSOOIDC
```

如需詳細資訊，請參閱在 Windows 上安裝 AWS 工具或在 Linux 或 macOS 上安裝 AWS 工具。

- AWS CLI 或 AWS 開發套件必須先使用工作登入資料進行設定，方法是執行下列其中一項動作：
  - 使用 AWS CLI `aws configure` 命令。如需詳細資訊，請參閱快速組態 (AWS CLI 文件)。
  - 設定 AWS CLI 或 AWS CDK 以透過 IAM 角色取得臨時存取權限。如需詳細資訊，請參閱取得用於 CLI 存取的 IAM 角色登入資料 (IAM 身分中心說明文件)。

限制

- 此指令碼無法用於管線或全自動化解決方案。部署此指令碼時，您必須手動授權 IAM 身分中心的存取權。然後指令碼會自動繼續。

產品版本

- 對於所有作業系統，建議您使用 7.0 或更新PowerShell 版本。

架構

您可以使用此模式中的指令碼同時重新整理多個 IAM 身分中心登入資料，也可以建立用於 AWS CLI、AWS 開發套件或 AWS CDK 的登入資料檔案。

工具

AWS 服務

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS IAM 身分中心可協助您集中管理對所有 AWS 帳戶和雲端應用程式的單一登入 (SSO) 存取。
- 的 AWS 工具 PowerShell是一 PowerShell 組模組，可協助您從命令列對 AWS 資源執行操作 PowerShell 指令碼。

其他工具

- PowerShell是一個 Microsoft 的自動化和配置管理程序，可以在 Windows，Linux 和 macOS 上運行。

最佳實務

為每個 IAM 身分中心執行個體保留一份此指令碼副本。不支援將一個指令碼用於多個執行個體。

史诗

執行 SSO 指令碼

任務	描述	所需技能
自訂 SSO 指令碼。	1. 複製其他資訊區段中的 SSO 指令碼。 2. 在Param本節中，針對您的 AWS 環境，定義下列變數的值： • DefaultRoleName — 設定為預設使用的 IAM 角色或權限。 • Region— 部署 IAM 身分中心的 AWS 區域。如需區域及其代碼的完整清單，請參閱區域端點。 • StartUrl— 用於存取 IAM 身分中心登入頁面的 URL。使用與指令碼中的範例值相同的格式。 • EnvironmentName — 參照此指令碼副本的簡短名稱，當您在同一工作階段中執行多個指令碼副本時使用。 3. 在第 10 行 (讀取) 下# Add your Account	雲端管理員

任務	描述	所需技能
	Information ，編輯雜湊表格中的下列值以反映您的環境： • Profile— 用於存放臨時登入資料的 AWS CLI 設定檔名稱。 • AccountId — 您要擷取登入資料的 AWS 帳戶 ID。 • RoleName— 您要使用的 IAM 身分中心角色或權限集的名稱。您可以保留此選項，$DefaultRoleName 就像您想要使用在Param區段中定義的相同角色一樣。 散列表中的每一行必須以逗號結束，除了最後一個。	
執行 SSO 指令碼。	建議您使用下列命令在 PowerShell shell 中執行自訂指令碼。 `./Set-AwsCliSsoCredentials.ps1` 或者，您也可以輸入下列命令，從另一個 shell 執行指令碼。 `pwsh Set-AwsCliSsoCredentials.ps1`	雲端管理員

故障診斷

問題	解決方案
No Access 錯誤	您使用的 IAM 角色沒有存取您在RoleName參數中定義的角色或權限集的權限的權限。更新您正在使用之角色的權限，或在指令碼中定義不同的角色或權限集。

相關資源

- 組態設定儲存在哪裡？ (AWS CLI 文件)
- 將 AWS CLI 設定為使用 AWS IAM 身分中心 (AWS CLI 文件)
- 使用具名設定檔 (AWS CLI 文件)

其他資訊

SSO 指令碼

在下列指令碼中，用您自己的資訊取代尖括號 (<>) 中的預留位置，並移除尖括號。

```
Set-AwsCliSsoCredentials.ps1
Param(
    $DefaultRoleName = '<AWSAdministratorAccess>',
    $Region          = '<us-west-2>',
    $StartUrl        = "<https://d-12345abcde.awsapps.com/start/>",
    $EnvironmentName = "<CompanyName>"
)
Try {$SsoAwsAccounts = (Get-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Scope
 Global -ErrorAction 'SilentlyContinue').Value.Clone()}
Catch {$SsoAwsAccounts = $False}
if (-not $SsoAwsAccounts) { $SsoAwsAccounts = @(
# Add your account information in the list of hash tables below, expand as necessary,
 and do not forget the commas
    @{Profile = "<Account1>"      ; AccountId = "<012345678901 >"; RoleName =
 $DefaultRoleName },
    @{Profile = "<Account2>"      ; AccountId = "<123456789012>"; RoleName =
 "<AWSReadOnlyAccess>" }
)}
```

```
$ErrorActionPreference = "Stop"
if (-not (Test-Path ~\.aws))      { New-Item ~\.aws -type Directory }
if (-not (Test-Path ~\.aws\credentials)) { New-Item ~\.aws\credentials -type File }
$CredentialFile = Resolve-Path ~\.aws\credentials
$PsuedoCreds    = @{AccessKey =
 'AKAEXAMPLE123ACCESS';SecretKey='PsuedoS3cret4cceSSKey123PsuedoS3cretKey'} # Pseudo
 Creds, do not edit.
Try {$SSOTokenExpire = (Get-Variable -Scope Global -Name
 "$($EnvironmentName)SSOTokenExpire" -ErrorAction 'SilentlyContinue').Value} Catch
 {$SSOTokenExpire = $False}
Try {$SSOToken       = (Get-Variable -Scope Global -Name "$($EnvironmentName)SSOToken"
 -ErrorAction 'SilentlyContinue').Value }      Catch {$SSOToken       = $False}
if ( $SSOTokenExpire -lt (Get-Date) ) {
    $SSOToken = $Null
    $Client   = Register-SSOOIDCClient -ClientName cli-sso-client -ClientType public -
Region $Region @PsuedoCreds
    $Device   = $Client | Start-SSOOIDCDeviceAuthorization -StartUrl $StartUrl -Region
 $Region @PsuedoCreds
    Write-Host "A Browser window should open. Please login there and click ALLOW." -
NoNewline
    Start-Process $Device.VerificationUriComplete
    While (-Not $SSOToken){
        Try {$SSOToken = $Client | New-SSOOIDCToken -DeviceCode $Device.DeviceCode -
GrantType "urn:ietf:params:oauth:grant-type:device_code" -Region $Region @PsuedoCreds}
        Catch {If ($_.Exception.Message -notlike "*AuthorizationPendingException*")
{Write-Error $_.Exception} ; Start-Sleep 1}
    }
    $SSOTokenExpire = (Get-Date).AddSeconds($SSOToken.ExpiresIn)
    Set-Variable -Name "$($EnvironmentName)SSOToken" -Value $SSOToken -Scope Global
    Set-Variable -Name "$($EnvironmentName)SSOTokenExpire" -Value $SSOTokenExpire -
Scope Global
}
$CredsTime     = $SSOTokenExpire - (Get-Date)
$CredsTimeText = ('{0:D2}:{1:D2}:{2:D2} left on SSO Token' -f $CredsTime.Hours,
 $CredsTime.Minutes, $CredsTime.Seconds).TrimStart("0 :")
for ($i = 0; $i -lt $SsoAwsAccounts.Count; $i++) {
    if (([DateTimeOffset]::FromUnixTimeSeconds($SsoAwsAccounts[$i].CredsExpiration /
 1000)).DateTime -lt (Get-Date).ToUniversalTime()) {
        Write-host "`r
    `rRegistering Profile $($SsoAwsAccounts[$i].Profile)" -NoNewline
        $TempCreds = $SSOToken | Get-SSORoleCredential -AccountId
 $SsoAwsAccounts[$i].AccountId -RoleName $SsoAwsAccounts[$i].RoleName -Region $Region
 @PsuedoCreds
```

```
        [PSCustomObject]@{AccessKey = $TempCreds.AccessKeyId; SecretKey =
 $TempCreds.SecretAccessKey; SessionToken = $TempCreds.SessionToken
        } | Set-AWSCredential -StoreAs $SsoAwsAccounts[$i].Profile -ProfileLocation
 $CredentialFile
        $SsoAwsAccounts[$i].CredsExpiration = $TempCreds.Expiration
    }
}
Set-Variable -name "$($EnvironmentName)SsoAwsAccounts" -Value $SsoAwsAccounts.Clone() -
Scope Global
Write-Host "`r$($SsoAwsAccounts.Profile) Profiles registered, $CredsTimeText"
```

使用 AWS Config 監控 Amazon Redshift 安全組態

創建者：盧卡斯考夫曼（AWS）和阿布舍克森加爾（AWS）

代碼存儲庫:awslab/aws-config-rules	環境：生產	技術：安全性、身分識別、合規
AWS 服務：AWS Config；Amazon Redshift；AWS Lambda		

Summary

您可以使用 AWS Config 評估 AWS 資源的安全組態。AWS Config 可以監控資源，如果組態設定違反了您定義的規則，AWS Config 會將資源標記為不合規。

您可以使用 AWS Config 來評估和監控您的 Amazon Redshift 叢集和資料庫。如需有關安全建議和功能的詳細資訊，請參閱 Amazon Redshift 中的安全性。此模式包含適用於 AWS 組態的自訂 AWS Lambda 規則。您可以在帳戶中部署這些規則，以監控 Amazon Redshift 叢集和資料庫的安全組態。此模式中的規則可協助您使用 AWS Config 確認：

- 已針對 Amazon Redshift 叢集中的資料庫啟用稽核記錄
- 需要 SSL 才能連接到 Amazon Redshift 集群
- 使用中的聯邦資訊處理標準 (FIPS) 密碼
- Amazon Redshift 叢集中的資料庫已加密
- 已啟用使用者活動監視

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 必須在您的 AWS 帳戶中啟用 AWS 組態。如需詳細資訊，請參閱使用主控台設定 AWS 組態或使用 AWS CLI 設定 AWS 組態。
- AWS Lambda 處理常式必須使用 3.9 版或更新版本。如需詳細資訊，請參閱使用 Python (AWS Lambda 文件)。

產品版本

- 版 Python 3.9 或更高版本

架構

目標技術堆疊

- AWS Config

目標架構

1. AWS Config 會定期執行自訂規則。
2. 自訂規則會叫用 Lambda 函數。
3. Lambda 函數會檢查 Amazon Redshift 叢集是否有不合規的組態。
4. Lambda 函數會向 AWS 組態報告每個 Amazon Redshift 叢集的合規狀態。

自動化和規模

AWS Config 自訂規則可擴展以評估您帳戶中的所有 Amazon Redshift 叢集。擴充此解決方案不需要其他動作。

工具

AWS 服務

- AWS Config 提供 AWS 帳戶中資源的詳細檢視，以及資源的設定方式。它可協助您識別資源彼此之間的關聯性，以及它們的組態隨著時間的推移而變更的方式。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Redshift 是 AWS 雲端中的受管 PB 級資料倉儲服務。

代碼存儲庫

此模式的代碼可在 GitHub aws-config-rules存儲庫中找到。此儲存庫中的自訂規則是 Python 程式設計語言中的 Lambda 規則。此儲存庫包含許多適用於 AWS Config 的自訂規則。在此模式中僅使用以下規則：

- REDSHIFT_AUDIT_ENABLED— 確認已在 Amazon Redshift 叢集上啟用稽核記錄。如果您也想要確認已啟用使用者活動監視，請改為部署REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED規則。
- REDSHIFT_SSL_REQUIRED— 確認需要 SSL 才能連接到 Amazon Redshift 叢集。如果您也想確認聯邦資訊處理標準 (FIPS) 密碼正在使用中，請改為部署規則REDSHIFT_FIPS_REQUIRED。
- REDSHIFT_FIPS_REQUIRED— 確認必須使用 SSL，且 FIPS 密碼正在使用中。
- REDSHIFT_DB_ENCRYPTED— 確認 Amazon Redshift 叢集中的資料庫已加密。
- REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED— 確認已啟用稽核記錄和使用者活動監視。

史诗

準備部署規則

任務	描述	所需技能
設定 IAM 政策。	1. 建立自訂的 IAM 身分識別政策，以允許 Lambda 執行角色讀取 Amazon Redshift 叢集組態。如需詳細資訊，請參閱管理資源的存取權限 (Amazon Redshift 文件) 和建立 IAM 政策 (IAM 文件)。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [`	AWS 管理員

任務	描述	所需技能
	<pre>"redshift :DescribeClusterPa rameterGroups", "redshift :DescribeClusterPa rameters", "redshift :DescribeClusters", "redshift :DescribeClusterSe curityGroups", "redshift :DescribeClusterSn apshots", "redshift :DescribeClusterSu bnetGroups", "redshift :DescribeEventSubs criptions", "redshift :DescribeLoggingSt atus"], "Resource": "*" }] }</pre> 2. 將AWSLambdaExecute和受AWSConfigRulesExecutionRole管政策指派為Lambda 執行角色的權限原則。如需指示，請參閱新增 IAM 身分許可 (IAM 文件)。	

任務	描述	所需技能
複製儲存庫。	在 Bash 外殼中，運行以下命令。這將從 GitHub中克隆存aws-config-rules儲庫。 `git clone https://github.com/awslabs/aws-config-rules.git`	一般 AWS

在 AWS Config 中部署規則

任務	描述	所需技能
在 AWS Config 中部署規則。	遵循建立自訂 Lambda 規則 (AWS Config 文件) 中的指示，在您的帳戶中部署下列一或多個規則： • `REDSHIFT_AUDIT_ENABLED` • `REDSHIFT_SSL_REQUIRED` • `REDSHIFT_FIPS_REQUIRED` • `REDSHIFT_DB_ENCRYPTED` • `REDSHIFT_USER_ACTIVITY_MONITORING_ENABLED`	AWS 管理員
驗證規則是否正常運作。	部署規則後，請遵循評估資源 (AWS Config 文件) 中的指示，確認 AWS Config 是否正確評	一般 AWS

任務	描述	所需技能
	估您的 Amazon Redshift 資源。	

相關資源

AWS 服務文件

- Amazon Redshift 中的安全性（Amazon Redshift 文檔）
- 管理資料庫安全性 (Amazon Redshift 文件)
- AWS Config 自訂規則 (AWS Config 文件)

AWS 方案指引

- 確認新的 Amazon Redshift 叢集具有必要的 SSL 端點
- 確保亞 Amazon Redshift 叢集在建立時已加密

其他資訊

您可以在 AWS Config 中使用下列 AWS 受管規則來確認下列適用於 Amazon Redshift 的安全組態：

- redshift-cluster-configuration-check— 使用此規則可確認 Amazon Redshift 叢集中的資料庫已啟用稽核記錄，並確認資料庫已加密。
- redshift-require-tls-ssl— 使用此規則可確認連線至 Amazon Redshift 叢集需要 SSL。

使用 Network Firewall 從輸出流量的伺服器名稱指示 (SNI) 擷取 DNS 網域名稱

創建者基蘭庫馬爾錢德拉什卡 (AWS)

環境：PoC 或試點	技術：安全性、身分識別、合規性；網路和行動應用程式	工作負載：所有其他工作
AWS 服務：AWS Lambda；AWS Network Firewall；Amazon VPC；Amazon CloudWatch 日誌		

Summary

此模式說明如何使用 Amazon Web Services (AWS) Network Firewall 來收集輸出網路流量 HTTPS 標頭中伺服器名稱指示 (SNI) 所提供的 DNS 網域名稱。Network Firewall 是一種受管服務，可讓您輕鬆部署 Amazon 虛擬私有雲 (Amazon VPC) 的關鍵網路保護，包括使用防火牆來保護輸出流量的能力，該防火牆會封鎖無法滿足特定安全需求的封包。保護特定 DNS 網域名稱的輸出流量稱為輸出篩選，這是監控並可能限制輸出資訊從一個網路到另一個網路的作法。

擷取透過 Network Firewall 傳遞的 SNI 資料後，您可以使用 Amazon CloudWatch 日誌和 AWS Lambda 將資料發佈到產生電子郵件通知的 Amazon 簡單通知服務 (Amazon SNS) 主題。電子郵件通知包括伺服器名稱和其他相關 SNI 資訊。此外，您可以使用此病毒碼的輸出，透過使用防火牆規則，依 SNI 中的網域名稱允許或限制輸出流量。如需詳細資訊，請參閱 Network Firewall 文件中的使用 AWS Network Firewall 中的可設定狀態規則群組。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI) 第 2 版，已在 Linux、macOS 或視窗上安裝和設定
- Network Firewall，在 Amazon VPC 中設定和設定，以及用於檢查輸出流量

注意：Network Firewall 可以使用下列任何 VPC 組態：

- 簡單的單一區域架構搭配網際網路閘道
- 具備網際網路閘道的多區域架構
- 具有網際網路閘道和 NAT 閘道的架構

架構

下圖顯示如何使用 Network Firewall 從輸出網路流量收集 SNI 資料，然後使用 CloudWatch 記錄和 Lambda 將該資料發佈到 SNS 主題。

該圖顯示以下工作流程：

1. Network Firewall 會從輸出網路流量 HTTPS 標頭中的 SNI 資料收集網域名稱。
2. CloudWatch 記錄會監控 SNI 資料，並在輸出網路流量通過 Network Firewall 時叫用 Lambda 函數。
3. Lambda 函數會讀取 CloudWatch 記錄所擷取的 SNI 資料，然後將該資料發佈至 SNS 主題。
4. SNS 主題會傳送包含 SNI 資料的電子郵件通知給您。

自動化和規模

- 您可以使用 AWS 使用基礎設施即程式碼 CloudFormation來建立此模式。

技術, 堆

- Amazon CloudWatch 日誌
- Amazon SNS
- Amazon VPC
- AWS Lambda
- AWS Network Firewall

工具

AWS 服務

- 亞馬遜 CloudWatch 日誌 — 您可以使用 Amazon CloudWatch 日誌從 Amazon 彈性運算雲端 (Amazon EC2) 執行個體、AWS、亞馬 Amazon Route 53 和其他來源監控 CloudTrail、存放和存取日誌檔。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 是一種受管服務，可提供從發佈者到訂閱者 (也稱為生產者和消費者) 的訊息傳遞。
- Amazon VPC — Amazon Virtual Private Cloud (Amazon VPC) 佈建 AWS 雲端的邏輯隔離部分，您可以在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 AWS 可擴展基礎設施的優勢。
- AWS Lambda — AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。
- AWS Network Firewall — AWS Network Firewall 是一項受管服務，可讓您輕鬆為所有 Amazon VPC 部署基本網路保護。

史诗

建立 Network Firewall 的 CloudWatch 記錄群組

任務	描述	所需技能
建立 CloudWatch 記錄群組。	1. 登入 AWS 管理主控台並開啟主CloudWatch 控台。 2. 在導覽窗格中，選擇 Log groups (日誌群組)。 3. 選擇 Actions (動作)，然後選擇 Create log group (建立日誌群組)。 4. 輸入日誌群組名稱，然後選擇 Create log group (建立日誌群組)。 如需詳細資訊，請參閱 CloudWatch 文件中的使用記錄群組和記錄資料流。	雲端管理員

建立 SNS 主題和訂閱

任務	描述	所需技能
建立 SNS 主題。	若要建立 SNS 主題，請遵循 Amazon SNS 文件中的指示進行。	雲端管理員
訂閱 SNS 主題的端點。	若要訂閱電子郵件地址作為您建立之 SNS 主題的端點，請遵循 Amazon SNS 文件中的指示。在通訊協定中，選擇電子郵件/電子郵件 JSON。注意：您也可以根據自己的需求選擇不同的端點。	雲端管理員

在 Network Firewall 中設定登入

任務	描述	所需技能
啟動防火牆記錄。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在功能窗格的 [NETEWORK 防火牆] 下，選擇 [防火牆]。 3. 在「防火牆」區段中，選擇要從 SNI 擷取輸出流量之伺服器名稱的防火牆。 4. 選擇「防火牆詳細資料」標籤，然後在「記錄」區段中選擇「編輯」。 5. 針對記錄類型，選取警示。針對警示的記錄目的地，選取CloudWatch 記錄群組。 6. 針對CloudWatch 記錄群組，搜尋並選擇您先前建立	雲端管理員

任務	描述	所需技能
	的記錄群組，然後選擇 [儲存]。 如需使用 CloudWatch Logs 做為 Network Firewall 日誌目標的詳細資訊，請參閱 Network Firewall 文件中的 Amazon CloudWatch Logs。	

在 Network Firewall 中設定可設定狀態規則

任務	描述	所需技能
建立可設定狀態規則。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在瀏覽窗格的 [Network Firewall] 下，選擇 [網路防火牆規則群組]。 3. 選擇建立 Network Firewall 規則群組。 4. 在 [建立 Network Firewall 規則群組] 頁面上，針對 [規則] 群組類型選擇 [可設定狀態規則群組]。附註：如需詳細資訊，請參閱使用 AWS Network Firewall 中的可設定狀態規則群組。 5. 在「可設定狀態規則群組」區段中，輸入規則群組的名稱和說明。 6. 針對容量，設定您要允許的可設定狀態規則群組的容量上限 (最多 30,000 個)。注	雲端管理員

任務	描述	所需技能
	意：建立規則群組後，就無法變更此設定。如需如何計算容量的詳細資訊，請參閱在 AWS Network Firewall 中設定規則群組容量。如需有關最大設定的資訊，請參閱 AWS Network Firewall 配額。 7. 針對可設定狀態規則群組選項，選取 5 個元組。 8. 在「可設定狀態規則順序」區段中，選擇「預設」。 9. 在「規則變數」區段中，保留預設值。 10. 在 [新增規則] 區段中，選擇 [TLS] 做為通訊協定。 在「來源」中選擇「任何」。對於來源連接埠，請選擇任何連接埠。 在「目的地」中選擇「任何」 對於目的地通訊埠，選擇任何連接埠。針對 [流量方向]，選擇 [轉寄] 在「動作」中選擇「警示」。選擇新增規則。 11. 選擇建立可設定狀態規則群組。	

任務	描述	所需技能
將可設定狀態規則關聯至 Network Firewall。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在功能窗格的 [網路防火牆] 下，選擇 [防火牆]。 3. 選擇您要從 SNI 擷取輸出流量之伺服器名稱的防火牆。 4. 在「可設定狀態規則群組」區段中，選擇「動作」，然後選擇「新增未受管理的可設定狀態規則群組」 5. 在 [新增未受管理的可設定狀態規則群組] 頁面上，選取您先前建立的可設定狀態規則群組，然後選擇 [新增可設定狀態規則群組]。	雲端管理員

建立 Lambda 函數以讀取記錄

任務	描述	所需技能
建立 Lambda 函數的程式碼。	在可從 Network Firewall 讀取輸出流量的 CloudWatch 記錄檔事件的整合式開發環境 (IDE) 中，貼上下列 Python 3 程式碼，並以您<SNS-topic-ARN>的值取代： `import json` `import gzip` `import base64` `import boto3` `sns_client = boto3.cli` `ent('sns')`	應用程式開發人員

任務	描述	所需技能
	def lambda_handler(event, context): decoded_event = json.loads(gzip.decompress(base64.b64decode(event['awslogs']['data']))) body = ''' {filtermatch} '''.format(loggroup=decoded_event['logGroup'], logstream=decoded_event['logStream'], filtermatch=decoded_event['logEvents'][0]['message'],) print(body) filterMatch = json.loads(body) data = [] if 'http' in filterMatch['event']: data.append(filterMatch['event']['http']['hostname']) elif 'tls' in filterMatch['event']: data.append(filterMatch['event']['tls']['sni']) result = 'Domain accessed ' + 1*' ' + (data[0]) + 1*' ' 'via AWS Network Firewall ' + 1*' ' + (filterMatch['firewall_name'])	

任務	描述	所需技能
	<pre> print(result) message = {'ServerName': result} send_to_sns = sns_client.publish(TargetArn=<SNS-topic-ARN>, #Replace with the SNS topic ARN Message=json.dumps({'default': json.dumps(message), 'sms': json.dumps(message), 'email': json.dumps(message)}), Subject='Server Name passed through the Network Firewall', MessageStructure='json')</pre> 此程式碼範例會剖析 CloudWatch 記錄檔內容，並擷取 SNI 在 HTTPS 標頭中提供的伺服器名稱。	
建立 Lambda 函數。	若要建立 Lambda 函數，請遵循 L ambda 文件中的指示，並針對執行階段選擇 Python 3.9。	雲端管理員
將程式碼新增至 Lambda 函數。	若要將 Python 程式碼新增至先前建立的 Lambda 函數，請依照 Lambda 文件中的指示進行。	雲端管理員

任務	描述	所需技能
將 CloudWatch 記錄新增為 Lambda 函數的觸發程序。	1. 登入 AWS 管理主控台並開啟 Lambda 主控台。 2. 在導覽窗格中，選擇 [函數]，然後選擇您先前建立的函數。 3. 在函數概觀區段中，選擇新增觸發條件。 4. 在 [新增觸發器] 頁面的 [觸發器組態] 區段中，選擇 [CloudWatch 記錄檔]，然後選擇 [新增]。 5. 針對記錄群組，選擇您先前建立的 CloudWatch 記錄群組。 6. 在「篩選器名稱」中，輸入篩選器的名稱。 7. 選擇新增。 8. 在函數頁面的 [組態] 索引標籤上，在 [觸發器] 區段中，選取您剛新增的觸發器，然後選擇 [啟用]。 如需詳細資訊，請參閱 Lambda 文件中的搭配 CloudWatch 日誌使用 Lambda。	雲端管理員

任務	描述	所需技能
新增 SNS 發佈權限。	將 SNS: 發佈權限新增至 Lambda 執行角色，以便 Lambda 可以進行 API 呼叫，將訊息發佈到 SNS。 1. 尋找您之前建立的 Lambda 函數的執行角色。 2. 將下列政策新增至您的 AWS Identity and Access Management (IAM) 角色： `{ "Version": "2012-10-17", "Statement": [ { "Sid": "AllowSNSPublish", "Effect": "Allow", "Action": [ "sns:GetTopicAttributes", "sns:Subscribe", "sns:Unsubscribe", "sns:Publish" ], "Resource": "*" } ] }`	雲端管理員

測試 SNS 通知的功能

任務	描述	所需技能
透過 Network Firewall 傳送流量。	1. 傳送或等待 HTTPS 流量通過 Network Firewall。 2. 當流量通過 Network Firewall 時，請查看您從 AWS 收到的 SNS 通知電子郵件。電子郵件包含輸出流量的 SNI 詳細資料。例如，如果存取的網域名稱為 https://aws.amazon.com，而訂閱通訊協定為電子郵件 JSON，則從上述 Lambda 程式碼產生的電子郵件將具有下列內容： `{ "Type": "Notification", "MessageId": "<messageID>", "TopicArn": "arn:aws:sns:us-west-2:123456789:testSNSTopic", "Subject": "Server Name passed through the Network Firewall", "Message": "{\"ServerName\": \"Domain 'aws.amazon.com' accessed via AWS Network Firewall 'AWS-Network-Firewall-Multi-AZ-firewall\"}",`	測試工程師

任務	描述	所需技能
	<pre>"Timestamp": "2022-03-22T04:10:04.217Z", "SignatureVersion": "1", "Signature": "<Signature>", "SigningCertURL": "<SigningCertUrl>", "UnsubscribeURL": "<UnsubscribeURL>" }</pre> 然後，CloudWatch 按照 Amazon CloudWatch 文檔中的說明檢查 Amazon 中的 Network Firewall 警報日誌。警示記錄會顯示下列輸出： <pre>{ "firewall_name": "AWS-Network-Firewall-Multi-AZ-firewall", "availability_zone": "us-east-2b", "event_timestamp": "<event timestamp>", "event": { "timestamp": "2021-03-22T04:10:04.214222+0000", "flow_id": <flow ID>, "event_type": "alert", "src_ip": "10.1.3.76", "src_port": 22761,</pre>	

任務	描述	所需技能
	"dest_ip": "99.86.59.73", "dest_port": 443, "proto": "TCP", "alert": { "action": "allowed", "signature_id": 2, "rev": 0, "signature": "", "category": "", "severity": 3 }, "tls": { "subject": "CN=aws.amazon.com", "issuerdn": "C=US, O=Amazon, OU=Server CA 1B, CN=Amazon", "serial": "<serial number>", "fingerprint": "<fingerprint ID>", "sni": "aws.amazon.com", "version": "TLS 1.2", "notbefore": "2020-09-30T00:00:00", "notafter": "2021-09-23T12:00:00", "ja3": {}, "ja3s": {}	

任務	描述	所需技能
	<pre> }, "app_proto": "tls" } }</pre>	

使用地形表單為組織自動啟 GuardDuty 用 Amazon

由阿爾蒂卡南（AWS）創建

代碼庫：amazon-guardduty-for-aws-organizations-with-terraform	環境：生產	技術：安全性、身分識別、合規性；雲端原生； DevOps
工作負載：所有其他工作	AWS 服務：Amazon GuardDuty；AWS Organizations	

Summary

Amazon 會 GuardDuty 持續監控您的 Amazon Web Services (AWS) 帳戶，並使用威脅情報來識別 AWS 環境中的意外和潛在惡意活動。 GuardDuty 針對多個帳戶或組織、跨多個 AWS 區域或透過 AWS 管理主控台手動啟用可能會很麻煩。您可以使用基礎結構即程式碼 (IaC) 工具 (例如 Terraform) 來自動執行程序，該工具可以佈建和管理雲端中的多帳戶、多區域服務和資源。

AWS 建議使用 AWS Organizations 在中設定和管理多個帳戶 GuardDuty。此模式遵循該建議。這種方法的一個好處是，當建立新帳戶或將新帳戶新增至組織時， GuardDuty 會在所有支援的區域中自動啟用這些帳戶，而不需要手動介入。

此模式示範如何使用 HashiCorp Terraform 為組織中 GuardDuty 的三個或更多 Amazon 網路服務 (AWS) 帳戶啟用亞馬遜。此模式提供的示例代碼執行以下操作：

- GuardDuty 為 AWS 組織中目標組織目前成員的所有 AWS Organizations 帳戶啟用
- 開啟中的自動啟用功能 GuardDuty，此功能會自動啟 GuardDuty 用 future 新增至目標組織的任何帳號
- 可讓您選取要啟用的區域 GuardDuty
- 使用組織的安全性帳戶做為 GuardDuty 委派的系統管理員
- 在記錄帳戶中建立 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體，並設定 GuardDuty 以發佈此儲存貯體中所有帳戶的彙總發現項目
- 指派生命週期政策，依預設，可在 365 天後將發現結果從 S3 儲存貯體轉換到 Amazon S3 Glacier 彈性擷取儲存

您可以手動執行此範例程式碼，或將其整合到持續整合和持續交付 (CI/CD) 管道中。

目標受眾

對於具有地形表單、Python 和 AWS Organizations 相關經驗的使用者，建議使用此模式。 GuardDuty

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 組織是在 AWS Organizations 中設定的，其中至少包含下列三個帳戶：
  - 管理帳戶 — 這是您用來部署 Terraform 程式碼的帳戶，不論是獨立或做為 CI/CD 管道的一部分。地形狀態也存儲在此帳戶中。
  - 安全性帳戶 — 此帳戶作為 GuardDuty 委派管理員使用。如需詳細資訊，請參閱 GuardDuty 委派系統管理員的重要考量 (GuardDuty 說明文件)。
  - 記錄帳戶 — 此帳戶包含 S3 儲存貯體，其中會 GuardDuty 發佈來自所有成員帳戶的彙總發現項目。

  如需如何使用所需組態設定組織的詳細資訊，請參閱建立帳戶結構 (AWS Well-Architected 實驗室)。
- 一個 Amazon S3 儲存貯體和一個 Amazon DynamoDB 表，可做為遠端後端，將 Terraform 的狀態存放在管理帳戶中。如需使用 Terraform 狀態的遠端後端的詳細資訊，請參閱 S3 後端 (Terraform 文件)。如需使用 S3 後端設定遠端狀態管理的程式碼範例，請參閱 remote-state-s3 後端 (Terraform 登錄)。請注意以下要求：
  - S3 儲存貯體和 DynamoDB 資料表必須位於相同的區域中。
  - 建立 DynamoDB 表時，分區索引鍵必須是 **`LockID`** (區分大小寫)，且分區索引鍵類型必須是字串。所有其他表格設定必須為其預設值。如需詳細資訊，請參閱關於主索引鍵和建立表格 (DynamoDB 文件)。
- 將用於存放 S3 儲存貯體的存取日誌的 S3 儲存貯體，在其中發佈 GuardDuty 發現項目。如需詳細資訊，請參閱啟用 Amazon S3 伺服器存取記錄 (Amazon S3 文件)。如果您要部署到 AWS Control Tower landing zone，則可以針對此目的重複使用日誌存檔帳戶中的 S3 儲存貯體。
- 已安裝並設定地形版本 0.14.6 或更新版本。如需詳細資訊，請參閱開始使用 — AWS (地形文件)。
- 已安裝並設定 Python 版本 3.9.6 或更新版本。如需詳細資訊，請參閱原始碼發行版本 (Python 網站)。
- 已安裝適用於 Python 的 AWS 開發套件。如需詳細資訊，請參閱安裝 (Boto3 說明文件)。

- jq 已安裝並配置。如需詳細資訊，請參閱下載 jq (jq 文件)。

限制

- 這種模式支持 macOS 和 Amazon Linux 2 操作系統。此病毒碼尚未經過測試，可用於 Windows 作業系統。
- GuardDuty 必須尚未在任何目標區域的任何帳戶中啟用。
- 此模式中的 IaC 解決方案不會部署先決條件。
- 此模式是針對遵循下列最佳實務的 AWS landing zone 所設計：
  - landing zone 是使用 AWS Control Tower 建立的。
  - 單獨的 AWS 帳戶用於安全和記錄。

產品版本

- 地形版本 0.14.6 或更高版本。示例代碼已經過測試版本 1.2.8。
- Python 版本 3.9.6 或更高版本。

架構

本節提供此解決方案的高階概觀，以及範例程式碼所建立的架構。下圖顯示在單一 AWS 區域內部署到組織中各個帳戶的資源。

1. Terraform 會在安全帳戶和記錄帳戶中建立 GuardDutyTerraformOrgRoleAWS Identity and Access Management (IAM) 角色。
2. Terraform 會在記錄帳戶的預設 AWS 區域中建立 S3 儲存貯體。此值區可用作發佈目的地，以彙總所有區域及組織中所有帳戶的所有發 GuardDuty 現項目。Terraform 也會在安全帳戶中建立 AWS Key Management Service (AWS KMS) 金鑰，用來加密 S3 儲存貯體中的發現項目，並設定將發現項目從 S3 儲存貯體自動存檔到 S3 Glacier 彈性擷取儲存。
3. 在管理帳戶中，Terraform 會將安全性帳戶指定為的委派系統管理員。 GuardDuty這表示安全性帳戶現在會管理所有成員帳戶的 GuardDuty 服務，包括管理帳戶。個人會員帳戶不能自 GuardDuty 行暫停或停用。
4. Terraform 會在安全性帳戶中為 GuardDuty 委派的系統管理員建立 GuardDuty 偵測器。

5. 如果尚未啟用，則地形會在中啟用 S3 保護。 GuardDuty如需詳細資訊，請參閱 Amazon 中的 Amazon S3 保護 GuardDuty (GuardDuty 文件)。
6. Terraform 將組織中所有目前、作用中的成員帳戶註冊為成員。 GuardDuty
7. Terraform 會設定 GuardDuty 委派管理員，將所有成員帳戶的彙總發現項目發佈到記錄帳戶中的 S3 儲存貯體。
8. 地形會針對您選擇的每個 AWS 區域重複步驟 3 到 7。

自動化和規模

提供的範例程式碼是模組化的，因此您可以將其整合到 CI/CD 管道中，以進行自動化部署。

工具

AWS 服務

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon GuardDuty 是一種持續的安全監控服務，可分析和處理日誌，以識別 AWS 環境中的未預期和潛在未經授權的活動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Key Management Service (AWS KMS) 可協助您建立和控制加密金鑰，以保護資料。
- AWS Organizations Organization 是一種帳戶管理服務，可協助您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- 適用於 Python 的 AWS 開發套件 (Boto3) 是一套軟體開發套件，可協助您將 Python 應用程式、程式庫或指令碼與 AWS 服務整合。

其他工具和服務

- HashiCorp Terraform 是一個命令列介面應用程式，可協助您使用程式碼來佈建和管理雲端基礎結構和資源。
- Python 是一種通用的編程語言。
- jq 是一個命令行處理器，可幫助您使用 JSON 文件。

代碼存儲庫

此模式的程式碼位於 amazon-guardduty-for-aws-organizations-with-terraform 儲存庫中。 GitHub

史诗

在組織 GuardDuty 中啟用

任務	描述	所需技能
複製儲存庫。	在 Bash 外殼中，運行以下命令。在 [其他資訊] 區段的 [複製儲存庫] 中，您可以複製包含 GitHub 存放庫 URL 的完整指令。這將從 GitHub中克隆 amazon-guardduty-for-aws-organizations-with-terraform 存儲庫。 `git clone <github-repository-url>`	DevOps 工程師
編輯地形表單組態檔案。	1. 在複製存放庫的root資料夾中，執行下列命令，複製組態 .json.sample 檔案。 `cp configuration.json.sample configuration.json` 2. 編輯新的組態 .json 檔案，並定義下列每個變數的值： • `management_acc_id` — 管理帳戶的帳號 ID。 • `delegated_admin_acc_id` — 安全性帳戶的帳號 ID。	DevOps 工程師, 一般 AWS, 地形, Python

任務	描述	所需技能
	• `logging_acc_id` — 記錄帳戶的帳戶 ID。 • `target_regions` — 您要啟用 GuardDuty的 AWS 區域清單 (逗號分隔)。 • `organization_id` — 您要啟用之組織的 AWS Organizations ID GuardDuty。 • `default_region` — 您的 Terraform 狀態儲存在管理帳戶中的區域。這與您為 Terraform 後端部署 S3 儲存貯體和 DynamoDB 表格的區域相同。 • `role_to_assume_for_role_creation` — 要指派給安全性和記錄帳戶中新 IAM 角色的名稱。您將在下一個故事中創建此新角色。Terraform 扮演此角色，可在安全性和記錄帳戶中建立 `GuardDutyTerraformOrgRole` IAM 角色。 • `finding_publishing_frequency` — 將 GuardDuty 發現項目發佈到 S3 儲存貯體的頻率。 • `guardduty_findings_bucket_region` —	

任務	描述	所需技能
	您想要為已發佈的發現項目建立 S3 儲存貯體的偏好區域。 • `logging_acc_s3_bucket_name` — 已發佈發現項目的 S3 儲存貯體的慣用名稱。 • `security_acc_kms_key_alias` — 用於加密 GuardDuty 發現項目的金鑰的 AWS KMS 別名。 • `s3_access_log_bucket_name` — 您想要收集用於 GuardDuty 發現項目之 S3 儲存貯體的存取日誌之既有 S3 儲存貯體的名稱。此儲存貯體應與 GuardDuty 發現值區位於相同的 AWS 區域。 • `tfm_state_backend_s3_bucket` — 用來存放 Terraform 遠端後端狀態的既有 S3 儲存貯體的名稱。 • `tfm_state_backend_dynamodb_table` — 用於鎖定地形狀態之預先存在的 DynamoDB 資料表的名稱。 3. 儲存並關閉 組態檔案。	

任務	描述	所需技能
為新的 IAM 角色產生 CloudFormation 範本。	此模式包括用於創建兩個 CloudFormation 模板的 IaC 解決方案。這些範本會建立 Terraform 在設定程序期間使用的兩個 IAM 角色。這些範本遵循最低權限權限的安全性最佳作法。 1. 在 Bash 殼層中，在儲存庫`root`資料夾中，導覽至`cfn-templates/` 。此文件夾包含帶存根的 CloudFormation 模板文件。 2. 執行下列命令。這將替換為您在配置 .json 文件中提供的值的存根。 `bash scripts/replace_config_stubs.sh` 3. 確認`cfn-templates/` 資料夾中已建立下列 CloudFormation 範本： • management-account-role.yaml — 此檔案包含管理帳戶中 IAM 角色的角色定義和關聯許可，該帳戶具有完成此模式所需的最低許可。 • role-to-assume-for-角色建立 .yaml — 此檔案包含安全性和記錄帳戶中 IAM 角色的角色定義和相關聯的許可。Terraform 擔任此	DevOps 工程師, 一般 AWS

任務	描述	所需技能
	角色，以便在這些帳戶中建立GuardDutyTerraformOrgRole角色。	
建立 IAM 角色。	依照建立堆疊 (說明CloudFormation 文件) 中的指示執行下列動作： 1. 在安全性和記錄帳戶中部署role-to-assume-for角色建立 .yaml 堆疊。 2. 在管理帳戶中部署 management-account-role.yaml 堆疊。成功建立堆疊並查看CREATE_COMPLETE 堆疊狀態後，請在輸出中記下此新角色的 Amazon 資源名稱 (ARN)。	DevOps 工程師, 一般 AWS
假設管理帳戶中的 IAM 角色。	作為安全性最佳實務，我們建議您在繼續之前先假設新的management-account-roleIAM角色。在 AWS Command Line Interface (AWS CLI) (AWS CLI) 的「其他資訊」部分的「假設管理帳戶 IAM 角色」中輸入命令。	DevOps 工程師, 一般 AWS

任務	描述	所需技能
運行安裝腳本。	在存放庫root資料夾中，執行下列命令以啟動安裝程序檔。 `bash scripts/full-setup.sh` full-setup.sh 指令碼會執行下列動作： • 將所有組態值匯出為環境變數 • 為每個 Terraform 模塊生成後端 .tf 和地形變量代碼文件 • 透過 AWS CLI 啟用組織 GuardDuty 內的受信任存取。 • 將組織狀態匯入地形狀態 • 建立 S3 儲存貯體，以便在記錄帳戶中發佈發現項目 • 建立 AWS KMS 金鑰以加密安全帳戶中的發現項目 • 如〈架構〉一節所述，在所有選取的區域中，在組織中啟 GuardDuty 用	DevOps 工程師, Python

(選擇性) 在組織 GuardDuty 中停用

任務	描述	所需技能
執行清理程序檔。	如果您使用此模式 GuardDuty 為組織啟用並想要停用 GuardDuty，請在存放	DevOps 工程師, 一般 AWS, 地形, Python

任務	描述	所需技能
	庫root資料夾中執行下列命令來啟動 cleanup-gd.sh 指令碼。 `bash scripts/cleanup-gd.sh` 此指令碼會 GuardDuty 在目標組織中停用、移除所有已部署的資源，並將組織還原至其先前的狀態，然後再使用 Terraform 啟用。 GuardDuty 注意這個指令碼不會移除 Terraform 狀態檔案，也不會從本機和遠端後端鎖定檔案。如果您需要這樣做，您必須手動執行這些動作。此外，此指令碼不會刪除匯入的組織或其管理的帳號。的受信任存取 GuardDuty 不會做為清理指令碼的一部分停用。	
移除 IAM 角色。	刪除使用role-to-assume-for角色建立 .yaml 和 .yaml 範本建立的堆疊。management-account-role CloudFormation 如需詳細資訊，請參閱刪除堆疊 (CloudFormation 文件集)。	DevOps 工程師, 一般 AWS

相關資源

AWS 文件

- 管理多個帳戶 (GuardDuty 說明文件)

- 授與最低權限 (IAM 文件)

AWS 行銷

- Amazon GuardDuty
- AWS Organizations

其他資源

- 地形
- 地形文檔 CLI 文檔

其他資訊

克隆存儲庫

執行下列命令以複製存 GitHub 放庫。

```
git clone https://github.com/aws-samples/amazon-guardduty-for-aws-organizations-with-terraform
```

假設管理帳戶 IAM 角色

若要在管理帳戶中擔任 IAM 角色，請執行下列命令。<IAM role ARN>以 IAM 角色的 ARN 取代。

```
export ROLE_CREDENTIALS=$(aws sts assume-role --role-arn <IAM role ARN> --role-session-name AWSCLI-Session --output json)
export AWS_ACCESS_KEY_ID=$(echo $ROLE_CREDENTIALS | jq .Credentials.AccessKeyId | sed 's/"//g')
export AWS_SECRET_ACCESS_KEY=$(echo $ROLE_CREDENTIALS | jq .Credentials.SecretAccessKey | sed 's/"//g')
export AWS_SESSION_TOKEN=$(echo $ROLE_CREDENTIALS | jq .Credentials.SessionToken | sed 's/"//g')
```

確認新的 Amazon Redshift 叢集具有必要的 SSL 端點

創建者：普里揚卡喬達瑞 (AWS)

環境：生產	技術：安全性、身分識別、合規性；分析；資料湖	AWS 服務：AWS CloudTrail；Amazon CloudWatch 活動；Amazon Redshift；Amazon SNS；AWS Lambda

Summary

此模式提供了一個 Amazon Web Services (AWS) CloudFormation 範本，該範本會在沒有安全通訊端層 (SSL) 端點的情況下啟動新的 Amazon Redshift 叢集時自動通知您。

Amazon Redshift 是全受管的 PB 級雲端資料倉儲服務。它是專為大規模數據集存儲和分析而設計的。它也被用來執行大規模的數據庫遷移。為了安全起見，Amazon Redshift 支援 SSL 來加密使用者的 SQL 伺服器用戶端應用程式和 Amazon Redshift 叢集之間的連線。若要將叢集設定為需要 SSL 連線，請在啟動期間`true`在與叢集關聯的參數群組中將參數設定為。`require_SSL`

此模式提供的安全控制可監控 AWS CloudTrail 日誌中的 Amazon Redshift API 呼叫，並針對CreateCluster、ModifyClusterRestoreFromClusterSnapshotCreateClusterParameterGroup、和ModifyClusterParameterGroupAPI 啟 CloudWatch 動 Amazon 事件事件。當事件偵測到其中一個 API 時，它會呼叫執行 Python 指令碼的 AWS Lambda。Python 函數分析所列 CloudWatch CloudTrail 事件的事件。從現有快照建立、修改或還原 Amazon Redshift 叢集時，會為叢集建立新的參數群組，或修改現有的參數群組，函數會檢查叢集的`require_SSL`參數。如果參數值為`false`，則函數會傳送 Amazon 簡單通知服務 (Amazon SNS) 通知給使用者，其中包含相關資訊：此通知來源於 Lambda 的 Amazon Redshift 叢集名稱、AWS 區域、AWS 帳戶和亞馬遜資源名稱 (ARN)。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 具有叢集子網路群組和關聯安全性群組的虛擬私人雲端 (VPC)。

限制

- 此安全控制是區域性的。您必須將其部署在要監控的每個 AWS 區域中。

架構

目標架構

自動化和規模

- 如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，描述 AWS 資源的變更。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。
- Amazon Redshift — Amazon Redshift 是雲端中的全受管 PB 級資料倉儲服務。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。訂閱者會收到發佈到所訂閱主題的所有訊息，且某一主題的所有訂閱者均會收到相同訊息。

Code

此模式包括下列附件：

- `RedshiftSSLEndpointsRequired.zip`— 安全控制的 Lambda 程式碼。

- `RedshiftSSLEndpointsRequired.yml`— 設定事件和 Lambda 函數的 CloudFormation 範本。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立 S3 儲存貯體來託管 Lambda 程式碼 .zip 檔案。此 S3 儲存貯體必須與您要監控的 Amazon Redshift 叢集位於相同的 AWS 區域。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用命名空間。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 .zip 檔案上傳至 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
啟動 AWS CloudFormation 範本。	在與 S3 儲存貯體相同的 CloudFormation AWS 區域中開啟 AWS 主控台，然後部署附加的範本`RedshiftSSLEndpointsRequired.yml` 。如需部署 AWS CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的在 AWS CloudFormation 主控台建立堆疊。	雲端架構師

任務	描述	所需技能
完成範本中的參數。	當您啟動範本時，系統會提示您輸入下列資訊： • S3 儲存貯體：指定您在第一個史詩中建立或選取的儲存貯體。這是您上傳附加的 Lambda 程式碼 (.zip 檔案) 的位置。 • S3 金鑰：指定 S3 儲存貯體中 Lambda .zip 檔案的位置 (例如檔案名稱 .zip 或控制項/檔案名稱 .zip)。請勿包含前導斜線。 • 通知電子郵件：提供您要接收 Amazon SNS 通知的作用中電子郵件地址。 • L@@ amba 記錄層級：指定 Lambda 函數的記錄層級和頻率。使用「資訊」(Info) 可記錄進度的詳細資訊訊息、針對仍允許部署繼續的錯誤事件發生錯誤，以及針對潛在有害情況發出警告。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱，才能開始接收違規通知。	雲端架構師

相關資源

- 建立 S3 儲存貯體 (Amazon S3 文件)
- 將檔案上傳到 S3 儲存貯體 (Amazon S3 文件)
- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail (AWS CloudTrail 文件)
- 創建一個 Amazon Redshift 集群（Amazon Redshift 文檔）
- 設定連線的安全選項 (Amazon Redshift 文件)

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

確認新的 Amazon Redshift 叢集是否在 VPC 中啟動

創建者：普里揚卡喬達瑞 (AWS)

環境：生產	技術：安全性、身分識別、合規性、分析、資料庫	AWS 服務：Amazon CloudWatch; AWS Lambda; Amazon Redshift

Summary

此模式提供了一個 Amazon Web Services (AWS) CloudFormation 範本，該範本會在虛擬私有雲 (VPC) 外部啟動 Amazon Redshift 叢集時自動通知您。

Amazon Redshift 是全受管的 PB 級雲端資料倉儲產品。它是專為大規模數據集存儲和分析而設計的。它也被用來執行大規模的數據庫遷移。Amazon Virtual Private Cloud (Amazon VPC) 可讓您佈建 AWS 雲端的邏輯隔離部分，您可以在定義的虛擬網路中啟動 AWS 資源，例如 Amazon Redshift 叢集。

此模式提供的安全控制可監控 AWS CloudTrail 日誌中的 Amazon Redshift API 呼叫，並為CreateCluster和 RestoreFromClusterSnapshotAPI 啟 CloudWatch 動 Amazon 事件事件。當事件偵測到其中一個 API 時，它會呼叫執行 Python 指令碼的 AWS Lambda。Python 函數會分析該 CloudWatch 事件。如果 Amazon Redshift 叢集是從快照建立或還原並出現在 Amazon VPC 網路外部，則該函數會傳送 Amazon 簡單通知服務 (Amazon SNS) 通知給使用者，其中包含相關資訊：Amazon Redshift 叢集名稱、AWS 區域、AWS 帳戶以及 Lambda 的亞馬遜資源名稱 (ARN)，該通知來源於 Lambda。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 具有叢集子網路群組和關聯安全性群組的 VPC。

限制

- AWS CloudFormation 範本僅支援CreateCluster和RestoreFromClusterSnapshot動作 (新叢集)。它不會偵測到在 VPC 外部建立的現有 Amazon Redshift 叢集。

- 此安全控制是區域性的。您必須將其部署在要監控的每個 AWS 區域中。

架構

目標架構

自動化和規模

如果您使用 AWS Organizations，則可以使用 AWS Cloudformation StackSets 在您要監控的多個帳戶中部署此範本。

工具

AWS 服務

- AWS CloudFormation — AWS 可 CloudFormation 協助您建立 AWS 資源的模型和設定、快速且一致地佈建，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。
- AWS CloudTrail — AWS 可 CloudTrail 協助您實作 AWS 帳戶的管理、合規以及操作和風險稽核。使用者、角色或 AWS 服務執行的動作會記錄為中的事件 CloudTrail。
- Amazon CloudWatch 活動 — Amazon CloudWatch 活動提供近乎即時的系統事件串流，用於描述 AWS 資源的變更。
- AWS Lambda — AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。
- Amazon Redshift — Amazon Redshift 是雲端中的全受管 PB 級資料倉儲服務。Amazon Redshift 與您的資料湖整合，可讓您使用資料為企業和客戶取得新的見解。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是可高度擴展的物件儲存服務，可用於各種儲存解決方案，包括網站、行動應用程式、備份和資料湖。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 協調和管理發佈者和客戶之間的訊息傳遞或傳送，包括 Web 伺服器和電子郵件地址。

Code

此模式包括下列附件：

- `RedshiftMustBeInVPC.zip`— 安全控制的 Lambda 程式碼。
- `RedshiftMustBeInVPC.yml`— 設定事件和 Lambda 函數的 CloudFormation 範本。

若要使用這些檔案，請遵循下一節中的指示。

史诗

設定 S3 儲存貯體

任務	描述	所需技能
定義 S3 儲存貯體。	在 Amazon S3 主控台上，選擇或建立 S3 儲存貯體來託管 Lambda 程式碼 .zip 檔案。此 S3 儲存貯體必須與您要監控的 Amazon Redshift 叢集位於相同的 AWS 區域。S3 儲存貯體名稱是全域唯一的，所有 AWS 帳戶都共用該命名空間。S3 儲存貯體名稱不能包含前導斜線。	雲端架構師
上傳程 Lambda 碼。	將附件區段中提供的 Lambda 程式碼 (`RedshiftMustBeInVPC.zip` 檔案) 上傳至 S3 儲存貯體。	雲端架構師

部署 CloudFormation 範本

任務	描述	所需技能
啟動 CloudFormation 範本。	在與 S3 儲存貯體相同的 AWS 區域中開啟 AWS CloudFormation 主控台，然後部署附加的範本 (`RedshiftMustBeInVPC.yml`)。如需	雲端架構師

任務	描述	所需技能
	部署 AWS CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的在 AWS CloudFormation 主控台建立堆疊。	
完成範本中的參數。	當您啟動範本時，系統會提示您輸入下列資訊： • S3 儲存貯體：指定您在第一個史詩中建立或選取的儲存貯體。這是您上傳附加的 Lambda 程式碼 (.zip 檔案) 的位置。 • S3 金鑰：指定 S3 儲存貯體中 Lambda .zip 檔案的位置 (例如檔案名稱 .zip 或控制項/檔案名稱 .zip)。請勿包含前導斜線。 • 通知電子郵件：提供您要接收 Amazon SNS 通知的作用中電子郵件地址。 • L@@ amba 記錄層級：指定 Lambda 函數的記錄層級和頻率。使用「資訊」(Info) 可記錄進度的詳細資訊訊息、針對仍允許部署繼續的錯誤事件發生錯誤，以及針對潛在有害情況發出警告。	雲端架構師

確認訂閱

任務	描述	所需技能
確認訂閱。	成功部署 CloudFormation 範本後，會將訂閱電子郵件傳送至您提供的電子郵件地址。您必須確認此電子郵件訂閱，才能開始接收違規通知。	雲端架構師

相關資源

- 建立 S3 儲存貯體 (Amazon S3 文件)
- 將檔案上傳到 S3 儲存貯體 (Amazon S3 文件)
- 在 AWS CloudFormation 主控台上建立堆疊 (AWS CloudFormation 文件)
- 使用 AWS 建立在 AWS API 呼叫上觸發的 CloudWatch 事件規則 CloudTrail (AWS CloudTrail 文件)
- 創建一個 Amazon Redshift 集群（Amazon Redshift 文檔）

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

更多模式

- 使用工作階段管理員和 Amazon EC2 執行個體 Connect 存取防禦主機
- 使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用 AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- ???
- 允許 EC2 執行個體寫入 AMS 帳戶中 S3 儲存貯體的存取權
- 將一個 AWS 帳戶中的 AWS CodeCommit 儲存庫與另一個帳戶中的 SageMaker 工作室建立關聯
- 使用 AWS Systems Manager 自動新增或更新 Windows 登錄項目
- ???
- 使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔
- 自動加密現有和新的 Amazon EBS 磁碟區
- 使用雲端託管人封鎖對 Amazon RDS 的公開存取
- ???
- 使用 cdk-nag 規則套件檢查 AWS CDK 應用程式或 CloudFormation 範本以取得最佳實務
- 啟動時檢查 EC2 執行個體是否有強制標籤
- 設定對 Amazon DynamoDB 的跨帳戶存取權
- 使用 Application Load Balancer 衡器在 Oracle EnterpriseOne 上為 Oracle WebLogic JD 愛德華設定 HTTPS 加密
- 針對 AWS IoT 環境中的安全事件設定記錄和監控
- 為在 Amazon EKS 上執行的應用程式設定相互 TLS 身份驗證
- ???
- 使用 AWS Amplify 增建立反應應用程式，並使用 Amazon Cognito 新增身份驗證
- 針對多個 AWS 帳戶的傳入網際網路存取建立網路存取分析器發現的報告
- 自訂 AWS Network Firewall 的 Amazon CloudWatch 提醒
- 使用 AWS Network Firewall 和 AWS Transit Gateway 部署防火牆
- 記錄您的 AWS landing zone 設計
- 在 Amazon RDS 中為 PostgreSQL 資料庫執行個體啟用加密連線
- 加密現有的亞馬遜 RDS 資料庫執行個體

- 啟動時強制執行 Amazon RDS 資料庫的自動標記
- 啟動時強制標記 Amazon EMR 叢集
- 確保啟動時已啟用 Amazon S3 的亞馬遜 EMR 記錄功能
- 使用 AWS Config 進階查詢，根據 AWS 資源的建立日期尋找 AWS 資源
- 使用對流圈產生包含 AWS 組態受管規則的 AWS CloudFormation 範本
- 當 AWS KMS 金鑰的金鑰狀態變更時，取得 Amazon SNS 通知
- ???
- 在未使用 AWS KMS 金鑰加密 Amazon 資料 Firehose 資源時識別並發出警示
- 透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能
- 擷取 EC2 Windows 執行個體並將其遷移到 AWS Managed Services 帳戶
- 使用 AWS DAmazon RDS for Oracle 以 SSL 模式 Amazon RDS for PostgreSQL 遷移到亞馬遜 RDS
- 將 ELK 堆疊遷移到 AWS 上的彈性雲端
- 將 F5 大 IP 工作負載遷移到 AWS 雲端上的 F5 大 IP VE
- 監控 Amazon Aurora 是否有沒有加密的
- 輪換資料庫認證而不重新啟動
- 使用受信任的內容，在 AWS 上的 Db2 聯合資料庫中保護和簡化使用者存取
- ???
- 使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront
- 使用憑證管理員和讓我們 end-to-end 加密為 Amazon EKS 上的應用程式設定加密
- 確認 ELB 負載平衡器需要 TLS 終止
- 使用 Splunk 檢視 AWS Network Firewall 日誌和指標
- 使用 Amazon 將所有 AWS 帳戶的 IAM 登入資料報告視覺化 QuickSight

無伺服器

主題

- 使用 AWS Amplify 建置無伺服器反應原生行動應用程式
- 使用運動資料串流和 Amazon 資料 Firehose 搭配 AWS CDK 將 DynamoDB 記錄交付至 Amazon S3
- 將 Amazon API Gateway 與 Amazon SQS 整合以處理異步 REST API
- 使用 Amazon API Gateway 和 AWS Lambda 以非同步方式處理事件
- 使用 Amazon API Gateway 和亞馬遜動 Amazon DynamoDB 串流非同步處理事件
- 使用 Amazon API Gateway、Amazon SQS 和 AWS Fargate 非同步處理事件
- 從 AWS Step Functions 同步執行 AWS Systems Manager Automation 任務
- 在 AWS Lambda 函數中使用 Python 來執行 S3 物件的 parallel 讀取
- 透過 VPC 端點設定對 Amazon S3 儲存貯體的私有存取
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 更多模式

使用 AWS Amplify 建置無伺服器反應原生行動應用程式

創建者迪克什圖魯五星 (AWS)

代碼庫：aws-amplify-react-native-ios-todo-app	環境：生產	資料來源:NA
目標：AWS Amplify、AWS AppSync、Amazon Cognito、Amazon DynamoDB	R 型：重新建築	工作負載：開源
技術：無伺服器；Web 和行動應用程式	AWS 服務：AWS Amplify；AWS；Amazon Cognito AppSync；Amazon DynamoDB	

Summary

此模式示範如何使用 AWS Amplify 和下列 AWS 服務，為 React 原生行動應用程式建立無伺服器後端：

- AWS AppSync
- Amazon Cognito
- Amazon DynamoDB

使用 Amplify 設定和部署應用程式的後端後端之後，Amazon Cognito 會對應用程式使用者進行身份驗證，並授權他們存取應用程式。 AppSync 然後，AWS 會與前端應用程式和後端 DynamoDB 表互動，以建立和擷取資料。

注意：此模式使用簡單的「ToDoList」應用程式做為範例，但您可以使用類似的程序來建立任何 React Native 行動應用程式。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Amplify 命令列介面 (Amplify CLI)，已安裝和設定
- XCode（任何版本）
- Microsoft 視覺工作室（任何版本，任何代碼編輯器，任何文本編輯器）
- 熟悉 Amplify
- 熟悉 Amazon Cognito
- 熟悉 AWS AppSync
- 熟悉 DynamoDB
- 對 Node.js 的熟悉程度
- 熟悉故宮
- 熟悉反應和反應本地
- 熟悉 JavaScript 和電子印刷稿 6 (ES6)
- 熟悉 GraphQL

架構

下圖顯示在 AWS 雲端中執行 React Native 行動應用程式後端的範例架構：

該圖顯示了以下架構：

1. Amazon Cognito 會對應用程式使用者進行身份驗證，並授權他們存取應用程式。
2. 為了建立和擷取資料，AWS AppSync 使用 GraphQL API 與前端應用程式和後端 DynamoDB 表進行互動。

工具

AWS 服務

- AWS Amplify 是一組專門建置的工具和功能，可協助前端 Web 和行動開發人員在 AWS 上快速建置完整堆疊應用程式。
- AWS AppSync 提供可擴展的 GraphQL 界面，可協助應用程式開發人員合併來自多個來源的資料，包括 Amazon DynamoDB、AWS Lambda 和 HTTP API。

- Amazon Cognito 為網頁和行動應用程式提供身份驗證、授權和使用者管理功能。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。

Code

在這個模式中使用的範例應用程式的程式碼可在 GitHub aws-amplify-react-native-ios-todo-app 存放庫中取得。要使用樣本文件，請按照此模式的 Epics 部分中的說明進行操作。

史诗

創建並運行您的反應本地應用程序

任務	描述	所需技能
設置一個反應本地開發環境。	如需指示，請參閱 React Native 文件中的設定開發環境。	應用程式開發人員
在 iOS 模擬器中創建並運行 ToDoList 反應本地移動應用程序。	1. 在新的終端機視窗中執行下列命令，在本機環境中建立新的 React Native 行動應用程式專案目錄： `npx react-native init ToDoListAmplify` 2. 執行下列命令，導覽至專案的根目錄： `cd ToDoListAmplify` 3. 執行下列命令來執行應用程式： `npx react-native run-ios`	應用程式開發人員

初始化應用程式的新後端環境

任務	描述	所需技能
在 Amplify 中建立支援應用程式所需的後端服務。	1. 在您的本機環境中，從專案的根目錄 (ToDoListAmplify) 執行下列命令： `amplify init` 2. 系統會出現提示，要求您提供有關應用程式的資訊。根據您的使用案例輸入必要資訊。然後按 Enter 鍵。 對於此模式中使用的 ToDoList 應用程序設置，請應用以下示例配置。 示例反應本地 Amplify 應用程序配置設置 `? Name: ToDoListAmplify` `? Environment: dev` `? Default editor: Visual Studio Code` `? App type: javascript` `? Javascript framework : react-native` `? Source Directory Path: src` `? Distribution Directory Path: /`	應用程式開發人員

任務	描述	所需技能
	`` ? Build Command: npm run-script build ? Start Command: npm run-script start ? Select the authentic ation method you want to use: AWS profile ? Please choose the profile you want to use: default ```	
	如需詳細資訊，請參閱擴大開發人員中心文件中的建立新的 Amplify 後端。	
	注意：命令`amplify init`使用 AWS 佈建下列資源 CloudFormation： • 適用於已驗證和未驗證使用者的 AWS Identity and Access Management (IAM) 角色 (驗證角色和未授權角色) • 用於部署的 Amazon 簡單存儲服務（亞馬遜 S3）存儲桶（對於此模式的示例應用程序，Amplif y-meta.json） • Amplify 託管中的後端環境	

將 Amazon Cognito 份驗證添加到您的 Amplify 反應本機應用程序

任務	描述	所需技能
創建一個 Amazon Cognito 份驗證服務。	1. 在您的本機環境中，從專案的根目錄 (ToDoListAmplify) 執行下列命令： `amplify add auth` 2. 系統會出現提示，要求您提供有關驗證服務組態設定的資訊。根據您的使用案例輸入必要資訊。然後按 Enter 鍵。 對於此模式中使用的 ToDoList 應用程序設置，請應用以下示例配置。 驗證服務組態設定範例 `? Do you want to use the default authentication and security configuration? \` `Default configuration` `? How do you want users to be able to sign in? \` `Username` `? Do you want to configure advanced settings? \` `No, I am done` 注意：此命`amplify add auth`令會在專案根目錄內的	應用程式開發人員

任務	描述	所需技能
	本機資料夾 (amplify) 中建立必要的資料夾、檔案和相依性檔案。對於此模式中使用的應用 ToDoList 程序設置，aws-exports.js 是為此目的創建的。	
將 Amazon Cognito 服務部署到 AWS 雲端。	1. 從專案的根目錄中，執行下列 Amplify CLI 命令： `amplify push` 2. 會出現確認部署的提示。輸入 「是」。然後按 Enter 鍵。 附註：若要查看專案中已部署的服務，請執行下列命令移至 Amplify 主控台： `amplify console`	應用程式開發人員

任務	描述	所需技能
安裝所需的 Amplify 庫作出反應原生和 iOS 的 CocoaPods 依賴關係。	1. 從專案的根目錄執行下列命令，以安裝所需的 Amplify 開放原始碼用戶端程式庫： `npm install aws-amplify aws-amplify-react-native \ amazon-cognito-identity-js @react-native-community/netinfo \ @react-native-async-storage/async-storage` 2. 執行下列命令，以安裝 iOS 所需的 CocoaPods 相依性： `npx pod-install`	應用程式開發人員

任務	描述	所需技能
匯入並設定 Amplify 服務。	在應用程式的進入點檔案 (例如 App.js) 中，輸入下列程式碼行，匯入並載入 Amplify 服務的組態檔案： `import Amplify from 'aws-amplify'` `import config from './src/aws-exports'` `Amplify.configure(config)` 注意：如果您在應用程式的進入點檔案中匯入 Amplify 服務後收到錯誤訊息，請停止應用程式。然後，打開 XCode 並從項目的 iOS 文件夾中選擇 ToDoListAmplify.xc 工作區並運行該應用程序。	應用程式開發人員

任務	描述	所需技能
更新應用程序的入口點文件以使用與身份驗證器高階組件（HOC）。	注意：withAuthenticator HOC 僅使用幾行代碼，即可在您的應用程序中提供登錄，註冊和忘記密碼工作流程。如需詳細資訊，請參閱選項 1：使用 Amplify 開發人員中心中的預先建置 UI 元件。此外，React 文檔中的高階組件。 1. 在應用程式的進入點檔案 (例如 App.js) 中，輸入下列程式碼行來匯入 withAuthenticator HOC： `import { withAuthenticator } from 'aws-amplify-react-native'` 2. 輸入下列程式碼以匯出使用驗證器 HOC： `export default withAuthenticator(App)` 使用驗證器 HOC 代碼示例 `import Amplify from 'aws-amplify'` `import config from './src/aws-exports'` `Amplify.configure(config)`	應用程式開發人員

任務	描述	所需技能
	<pre>import { withAuthe nticator } from 'aws-amplify-react- native'; const App = () => { return null; }; export default withAuthe nticator(App);</pre> 注意：在 iOS 模擬器中，該應用程序會顯示 Amazon Cognito 服務提供的登錄屏幕。	

任務	描述	所需技能
測試驗證服務設定。	在 iOS 模擬器中，執行以下操作： 1. 使用真實的電子郵件地址在應用程序中創建一個新帳戶。然後將驗證碼發送到註冊的電子郵件。 2. 使用您在驗證電子郵件中收到的驗證碼來驗證帳戶設定。 3. 輸入您建立的使用者名稱和密碼。然後，選擇「登入」。 歡迎畫面隨即出現。 注意：您也可以開啟 Amazon Cognito 主控台，並檢查是否已在身分集區中建立新使用者。	應用程式開發人員

將 AWS AppSync API 和 DynamoDB 資料庫 Connect 到應用程式

任務	描述	所需技能
建立 AWS AppSync API DynamoDB 料庫。	1. 從專案的根目錄執行下列 Amplify CLI 命令，將 AWS AppSync API 新增到您的應用程式，並自動佈建 DynamoDB 資料庫： `amplify add api` 2. 會出現提示，要求您提供有關 API 和資料庫組態設定的資訊。根據您的使用案例輸	應用程式開發人員

任務	描述	所需技能
	入必要資訊。然後按 Enter 鍵。Amplify CLI 會在您的文字編輯器中開啟 GraphQL 結構描述檔案。 對於此模式中使用的 ToDoList 應用程序設置，請應用以下示例配置。 API 和資料庫組態設定範例 `? Please select from one of the below mentioned services: \` `GraphQL` `? Provide API name: todolistamplify` `? Choose the default authorization type for the API \` `Amazon Cognito User Pool` `Do you want to use the default authentication and security configura tion` `? Default configuration How do you want users to be able to sign in? \` `Username` `Do you want to configure advanced settings? \` `No, I am done.`	

任務	描述	所需技能
	`` ? Do you want to configure advanced settings for the GraphQL API \ No, I am done. ? Do you have an annotated GraphQL schema? \ No ? Choose a schema template: \ Single object with fields (e.g., “Todo” with ID, name, description) ? Do you want to edit the schema now? \ Yes ``` GraphQL 結構描述範例 ``` type Todo @model { id: ID! name: String! description: String } ```	

任務	描述	所需技能
部署 AWS AppSync API。	1. 在專案的根目錄中，執行下列 Amplify CLI 命令： `amplify push` 2. 會出現提示，要求您提供有關 API 和資料庫組態設定的詳細資訊。根據您的使用案例輸入必要資訊。然後按 Enter 鍵。您的應用程式現在可以與 AWS AppSync API 互動。 對於此模式中使用的 ToDoList 應用程序設置，請應用以下示例配置。 AWS AppSync API 組態設定範例 注意：以下組態會在 AWS 中建立 GraphQL API， AppSync 並在 Dynamo DB 中建立待辦事項表。 `? Are you sure you want to continue? Yes` `? Do you want to generate code for your newly created GraphQL API Yes` `? Choose the code generation language target javascript` `? Enter the file name pattern of graphql queries, mutations and`	應用程式開發人員

任務	描述	所需技能
	subscriptions src/graphql/**/*.js ? Do you want to generate/update all possible GraphQL operations - \ queries, mutations and subscriptions Yes ? Enter maximum statement depth \ [increase from default if your schema is deeply nested] 2	
將應用程式的前端 Connect 到 AWS AppSync API。	若要使用此模式中提供的範例 ToDoList 應用程式，請從 aws-amplify-react-native-ios-todo-app GitHub 儲存庫中的 App.js 檔案複製程式碼。然後，將範例程式碼整合到您的本機環境中。 儲存庫的 App.js 檔案中提供的範例程式碼會執行下列作業： • 顯示使用 [標題] 和 [說明] 欄位建立ToDo 項目的表單 • 顯示待辦事項清單 (標題和說明) • 帖子和通過使用方法獲取數aws-amplify 據	應用程式開發人員

相關資源

- AWS Amplify
- Amazon Cognito

- AWS AppSync
- Amazon DynamoDB
- 反應（反應文檔）

使用運動資料串流和 Amazon 資料 Firehose 搭配 AWS CDK 將 DynamoDB 記錄交付至 Amazon S3

由沙銀石蝦（AWS）和丹尼爾·馬圖基達庫尼亞（AWS）創建

程式碼儲存庫：Amazon DynamoDB 擷取到 Amazon S3	環境：PoC 或試點	技術：無伺服器、資料湖、資料庫、儲存與備份
AWS 服務：AWS CDK；亞馬遜動態 B；亞 Amazon Kinesis Data Firehose；Amazon Kinesis Data Streams；AWS Lambda；Amazon S3		

Summary

此模式提供範例程式碼和應用程式，可使用 Amazon Kinesis 資料串流和亞馬遜資料串流將記錄從 Amazon DynamoDB 交付到亞馬遜簡單儲存服務 (Amazon S3)。該模式的方法使用 AWS Cloud Development Kit (AWS CDK) L3 建構，並包含一個範例，說明如何在將資料傳送到 Amazon Web Services (AWS) 雲端上的目標 S3 儲存貯體之前，先使用 AWS Lambda 執行資料轉換。

Kinesis Data Streams 會在 DynamoDB 表格中記錄項目層級的修改，並將其複寫到所需的 Kinesis 資料串流。您的應用程式可以存取 Kinesis 資料串流，並以近乎即時的速度檢視項目層級的變更。Kinesis Data Streams 也可讓您存取其他 Amazon Kinesis 服務，例如適用於 Apache Flink 的 Firehose 和 Amazon 管理服務。這表示您可以建置可提供即時儀表板、產生警示、實作動態定價和廣告，以及執行複雜資料分析的應用程式。

您可以將此模式用於資料整合使用案例。例如，運輸車輛或工業設備可以將大量資料傳送至 DynamoDB 表格。然後可以轉換此資料，並將其存放在 Amazon S3 託管的資料湖中。然後，您可以使用亞馬遜雅典娜、亞馬遜 Amazon Redshift Spectrum、亞馬 Amazon Rekognition 和 AWS Glue 等無伺服器服務來查詢和處理資料，並預測任何潛在的缺陷。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。如需詳細資訊，請參閱 AWS CLI 文件中的開始使用 AWS CLI。
- Node.js（18 倍以上）和故宮，已安裝和配置。如需詳細資訊，請參閱npm文件中的下載和安裝 Node.js 和 npm。
- aws-cdk (2. x 以上)，已安裝並設定。如需詳細資訊，請參閱 AWS CDK 文件中的 AWS CDK 入門。
- 在本機電腦上複製並設定的 GitHub aws-Dynamodb 動態螢火管-s3 擷取儲存庫。
- DynamoDB 資料表的現有範例資料。資料必須使用下列格式：`{"SourceDataId": {"S": "123"},"MessageData":{"S": "Hello World"}}`

架構

下圖顯示使用 Kinesis 資料串流和 Firehose 將記錄從 DynamoDB 傳送到 Amazon S3 的範例工作流程。

該圖顯示以下工作流程：

1. 使用 Amazon API Gateway 擷取資料做為 DynamoDB 的代理伺服器。您也可以使用任何其他來源將資料內嵌至 DynamoDB。
2. Kinesis Data Streams 中會以近乎即時的速度產生項目層級變更，以便交付至 Amazon S3。
3. Kinesis Data Streams 會將記錄傳送至 Firehose 進行轉換和交付。
4. Lambda 函數會將記錄從 DynamoDB 記錄格式轉換為 JSON 格式，其中只包含記錄項目屬性名稱和值。

工具

- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS CDK 工具組是命令列 Cloud Development Kit，可協助您與 AWS 雲端開發套件 (AWS CDK) 應用程式互動。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。

Code

此模式的程式碼可在 GitHub aws-Dynamodb 動態螢火管- s3 擷取儲存庫中找到。

史诗

設定和設定範例程式碼

任務	描述	所需技能
安裝依賴關係。	在本機電腦上，執行下列命令，從pattern/aws-dynamodb-kinesisstreams-s3 和目sample-application 錄中的package.json 檔案安裝相依性： `cd <project_root>/pattern/aws-dynamodb-kinesisstreams-s3` `npm install && npm run build` `cd <project_root>/sample-application/` `npm install && npm run build`	一般 AWS 應用程式開發人員

任務	描述	所需技能
產生 AWS CloudFormation 範本。	1. 執行 `cd <project_root>/sample-application/` 命令。 2. 執行`cdk synth`命令以產生 AWS CloudFormation 範本。 3. `AwsDynamodbKinesisfirehoseS3IngestionStack.template.json` 輸出存儲在目`cdk.out`錄中。 4. 使用 AWS CDK 或 AWS 管理主控台在 AWS CloudFormation 中處理範本。	AWS 一般 AWS 應用程式開發人員 DevOps

部署資源

任務	描述	所需技能
檢查並部署資源。	1. 執行`cdk diff`命令以識別 AWS CDK 建構所建立的資源類型。 2. 執行`cdk deploy`命令以部署資源。	AWS 一般 AWS 應用程式開發人員 DevOps

將資料擷取至 DynamoDB 表格以測試解決方案

任務	描述	所需技能
將您的範例資料內嵌到 DynamoDB 資料表中。	1. 透過在 AWS CLI 中執行下列命令，將請求傳送到您的 DynamoDB 表： `aws dynamodb put-item --table-name <your_table_name> --item '{"<table_partition_key>": {"S": "<partition_key_ID>"},"MessageData":{"S": "<data>"}}'` 例如： `aws dynamodb put-item --table-name SourceData_table --item '{"SourceDataId": {"S": "123"},"MessageData":{"S": "Hello World"}}'` 依預設，如果作業成功，則`put-item`不會傳回任何值做為輸出。如果操作失敗，則返回一個錯誤。資料會儲存在 DynamoDB 中，然後傳送至 Kinesis Data Streams 和 Firehose。	應用程式開發人員

任務	描述	所需技能
	附註：您可以使用不同的方法將資料新增至 DynamoDB 表格。如需詳細資訊，請參閱 Amazon DynamoDB 說明文件中的將資料載入表格。	
確認已在 S3 儲存貯體中建立新物件。	登入 AWS 管理主控台並監控 S3 儲存貯體，以確認新物件是使用您傳送的資料建立的。 如需詳細資訊，請`get-object` 參閱 Amazon S3 API 參考文件中的。	一般 AWS 應用程式開發人員

清除資源

任務	描述	所需技能
清理資源。	執行命`cdk destroy`令以刪除此模式使用的所有資源。	一般 AWS 應用程式開發人員

相關資源

- S3-靜態站點堆棧 .ts（存儲庫）GitHub
- aws-apigateway-dynamodb 模塊（GitHub 存儲庫）
- aws 運動流運動火花 -3 模塊（存儲庫）GitHub
- 變更動態資料擷取的資料擷取 (Amazon DynamoDB 文件)
- 使用 Kinesis Data Streams 擷取對 DynamoDB 的變更 (Amazon DynamoDB 文件)

將 Amazon API Gateway 與 Amazon SQS 整合以處理異步 REST API

由納塔利婭·科爾安東尼奧法維羅（AWS）和古斯塔沃·馬蒂姆（AWS）創建

環境：PoC 或試點	技術：無伺服器；訊息與通訊	AWS 服務：Amazon API Gateway；Amazon SQS

Summary

部署 REST API 時，有時您需要公開用戶端應用程式可以發佈的訊息佇列。例如，您可能會遇到第三方 API 的延遲和回應延遲的問題，或者您可能想要避免資料庫查詢的回應時間，或者在存在大量並行 API 時避免擴展伺服器。在這些情況下，發佈到佇列的用戶端應用程式只需要知道 API 接收到資料，而不是在收到資料之後會發生什麼情況。

此模式使用 Amazon API Gateway 將消息發送到 Amazon S imple Queue Service (Amazon SQS) 來創建 REST API 端點。它會建立兩個服務之間的 easy-to-implement 整合，避免直接存取 SQS 佇列。

先決條件和限制

- 活躍 AWS 帳戶

架構

圖表說明了這些步驟：

1. 使用郵遞員、其他 API 或其他技術等工具要求 POST REST API 端點。
2. API Gateway 會在佇列上張貼在要求內文上接收的訊息。
3. Amazon SQS 會接收訊息，並將包含成功或失敗程式碼的 API Gateway 傳送回答。

工具

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。

- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。

史诗

建立 SQS 佇列

任務	描述	所需技能
建立佇列。	若要建立接收來自 REST API 之訊息的 SQS 佇列，請執行下列動作： 1. 請登入您的 AWS 帳戶。 2. 在 https://console.aws.amazon.com/sqs/ 開啟 Amazon SQS 主控台。 3. 選擇建立佇列。 4. 在 [建立佇列] 頁面上，從 [區域] 下拉式清單 AWS 區域 中選擇正確的佇列。 5. 對於「類型」，保留預設設定 (標準)。 6. 輸入佇列的名稱。 7. 保留所有其他設定的預設值。 8. 選擇建立佇列。	應用程式開發人員

提供對 Amazon SQS 的訪問

任務	描述	所需技能
建立 IAM 角色。	這個 IAM 角色可讓 API Gateway 資源完整存取 Amazon SQS。 1. 前往網址 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。 2. 在導覽窗格中，選擇 Roles (角色)、Create role (建立新角色)。 3. 對於 Trusted entity type (信任的實體類型)，請選擇 AWS 服務。 4. 針對使用案例，從下拉式清單中選擇 [API Gateway]，然後選擇 [下一步]、[下一步]。 5. 在 [角色名稱] 中，輸入AWSGatewayRoleForSQS和選擇性說明，然後選擇 [建立角色]。 6. 在「角色」窗格中，搜尋AWSGatewayRoleForSQS並選取其核取方塊。 7. 在許可政策區段中，選擇新增許可、連接政策。 8. 搜索亞馬遜 SQS FullAccess 並選擇它。 9. 選擇新增許可。 10.在的摘要區段中 AWSGatewayRoleForS	應用程式開發人員，AWS 管

任務	描述	所需技能
	QS，複製 Amazon 資源編號 (ARN)。您將在稍後的步驟中使用此 ID。	

創建一個其餘 API

任務	描述	所需技能
創建一個其餘 API。	這是 HTTP 請求被發送到的其餘 API。 1. 在以下網址開啟 API Gateway 主控台：https://console.aws.amazon.com/apigateway/。 2. 在 [其餘 API] 區段中，選擇 [建置]。 3. 對於 API 名稱，請輸入 API 的名稱和可選描述，保留所有其他默認設置，然後選擇「創建 API」。	應用程式開發人員
將 API Gateway Connect 到 Amazon SQS。	此步驟可讓訊息從 HTTP 要求的內文內部流向 Amazon SQS。 1. 在 API Gateway 主控台上，選擇您建立的 API。 2. 在 [資源] 頁面的 [方法] 區段中，選擇 [建立方法]。 3. 針對方法類型，選擇 POST。 4. 針對「整合類型」，選擇AWS 服務。	應用程式開發人員

任務	描述	所需技能
	5. 在中 AWS 區域，選擇您建立 SQS 佇列的區域。 6. 對於 AWS 服務，請選擇 S imple Queue Service (SQS)。 7. 如果是 HTTP 方法，請選擇「張貼」。 8. 在「動作類型」中選擇「使用路徑覆寫」。 9. 輸入/<AWS account ID>做為 「路徑取代」 <name of SQS queue>。 10對於「執行」角色，請貼上您先前建立之角色的 ARN。 11選擇建立方法。	

測試其餘 API

任務	描述	所需技能
測試其餘 API。	運行測試以檢查缺少的配置： 1. 在 API Gateway 主控台上，選擇您建立的 REST API。 2. 在 [資源] 窗格中，選擇 POST 方法。 3. 選擇測試標籤。(如果未顯示索引標籤，請使用向右鍵。) 4. 對於請求主體，粘貼以下 JSON 代碼： {	應用程式開發人員

任務	描述	所需技能
	<pre>"message": "lorem ipsum" }</pre> 5. 選擇 測試。 您會收到類似下列內容的錯誤訊息： <pre><UnknownOperationException/></pre>	

任務	描述	所需技能
變更 API 整合以將請求正確轉送至 Amazon SQS。	完成設定以修正整合錯誤： 1. 在 API Gateway 主控台上，選擇您建立的 API，然後選擇「張貼」。 2. 「方法執行」區段顯示 API Gateway 和 Amazon SQS 之間的視覺對應。從此區段中選擇整合要求，然後選擇編輯。 3. 展開 [HTTP 標頭] 區段，然後選擇 [新增要求標頭] 參數。 • 對於「名稱」，指定內容類型。 • 在對應來源中，輸入 「應用程式/x-www-form-urlencoded」。請確定包含單引號。 • 選取 [快取] 核取方塊。 4. 展開 [對應範本] 區段。 • 選擇 Add mapping template (新增對應範本)。 • 在「內容類型」中，輸入應用程式 /json。 • 對於模板主體，粘貼以下代碼： `Action=SendMessage&MessageBody=$input.body`	應用程式開發人員

任務	描述	所需技能
	• 選擇儲存。	
在 Amazon SQS 中測試和驗證訊息。	運行測試以確認測試是否成功完成： 1. 在 API Gateway 主控台上，選擇您建立的 REST API。 2. 在 [資源] 窗格中，選擇 POST 方法。 3. 選擇測試標籤。(如果未顯示索引標籤，請使用向右鍵。) 4. 對於請求主體，粘貼以下 JSON 代碼： `{ "message": "lorem ipsum" }` 5. 選擇 測試。 6. 開啟 Amazon SQS 主控台。 7. 在功能窗格中，選擇 [佇列]，然後選擇您的佇列。 8. 選擇傳送及接收訊息。 9. 選擇訊息輪詢。 10選擇 Message (訊息)。它應該顯示以下內容： `Body { "message": "lorem ipsum" }`	應用程式開發人員

任務	描述	所需技能
使用特殊字元測試 API Gateway。	運行包含消息中不可接受的特殊字符（例如 & ）的測試： 1. 在 API Gateway 主控台上，選擇您的 API。 2. 使用下列 JSON 程式碼重複先前步驟的測試： `{ "message": "lorem ipsum &" }` 3. 選擇 測試。 您會收到如下錯誤訊息： `{ "Error": { "Code": "AccessDenied", "Message": "Access to the resource https://sqs.us-east-2.amazonaws.com/976166761794/Apg2 is denied.", "Type": "Sender" }, "RequestId": "e83c9c67-bcf6-5e9a-91e9-c737094b17ab" }` 這是因為郵件內文預設不支援特殊字元。在下一步中，您	應用程式開發人員

任務	描述	所需技能
	將配置 API Gateway 以支持特殊字符。如需有關內容類型轉換的詳細資訊，請參閱 API Gateway 文件。	

任務	描述	所需技能
變更 API 設定以支援特殊字元。	調整組態以接受訊息中的特殊字元： 1. 在 API Gateway 主控台上，選擇您建立的 API，然後選擇「張貼」。 2. 選擇整合請求，然後選擇編輯。 3. 將內容處理變更為「轉換為文字」。 4. 在「對應範本」區段中： • 在「內容類型」中，輸入應用程式 /json。 • 對於「範本內文」，指定： `Action=SendMessage&MessageBody=$util.urlEncode($input.body)` • 選擇儲存。 5. 選擇測試標籤。 6. 對於請求主體，輸入先前的 JSON 代碼： `{ " message": "lorem ipsum &" }` 7. 選擇 測試。 8. 開啟 Amazon SQS 主控台。	應用程式開發人員

任務	描述	所需技能
	9. 選取您的佇列，然後選擇 [傳送和接收郵件]、[輪詢郵件]、[郵件如前所述]。 新訊息應包含特殊字元。	

部署其餘 API

任務	描述	所需技能
部署應用程式介面。	若要部署其餘 API： 1. 開啟 API Gateway 主控台。 2. 選擇您的 API。 3. 選擇部署 API。如需有關此步驟的詳細資訊，請參閱 API Gateway 文件。	應用程式開發人員
使用外部工具進行測試。	使用外部工具執行測試，以確認郵件已成功收到： 1. 打開郵差，失眠或 cURL 之類的工具。 2. 執行您的 API。 3. 開啟 Amazon SQS 主控台。 4. 選擇您的佇列。 5. 載入訊息以查看新訊息。	應用程式開發人員

清除

任務	描述	所需技能
刪除應用程式介面。	在 API Gateway 主控台上，選擇您建立的 API，然後選擇 [刪除]。	應用程式開發人員
刪除 IAM 角色。	在 IAM 主控台的 [角色] 窗格中，選取 AWSGatewayRoleForSQS，然後選擇 [刪除]。	應用程式開發人員
刪除 SQS 佇列。	在 Amazon SQS 主控台的 [佇列] 窗格中，選擇您建立的 SQS 佇列，然後選擇 [刪除]。	應用程式開發人員

相關資源

- SQS-SendMessage (API Gateway 文件)
- API Gateway 中的內容類型轉換 (API Gateway 文件)
- $ 使用者變數 (API Gateway 文件)
- 如何將 API Gateway REST API 與 Amazon SQS 整合並解決常見錯誤？ （AWS Re：發表文章）

使用 Amazon API Gateway 和 AWS Lambda 以非同步方式處理事件

創建者：安德烈·梅羅尼（AWS），納迪姆馬吉德（AWS），瑪麗·克蒂里（AWS）和邁克爾·沃爾納（AWS）

程式碼儲存庫：使用 API Gateway 和 Lambda 進行非同步事件	環境：PoC 或試點	技術：無伺服器
AWS 服務：Amazon API Gateway；Amazon DynamoDB；AWS Lambda		

Summary

Amazon API Gateway 是一項全受管服務，開發人員可用來建立、發佈、維護、監控和保護任何規模的 API。它可以處理接受和處理多達數十萬個並行 API 呼叫所涉及的工作，包括下列各項：

- 交通管理
- 跨來源資源共用 (CORS) 支援
- 授權和存取控制
- 限流
- 監控
- API 版本管理

API Gateway 的重要服務配額是整合逾時。逾時是 REST API 傳回錯誤之前，後端服務必須傳回回應的時間上限。對於同步工作負載，通常可以接受 29 秒的硬性限制。但是，對於那些想要將 API Gateway 與非同步工作負載搭配使用的開發人員而言，這個限制代表了

此模式顯示了使用 API Gateway 和 AWS Lambda非同步處理事件的範例架構。該架構支持運行持續時間長達 15 分鐘的處理任務，並使用基本的 REST API 作為接口。

Projen 用於設置本地開發環境，並將示例架構部署到目標 AWS 帳戶，並與AWS Cloud Development Kit (AWS CDK) 工具包，Docker 和 Node.js 相結合。Projen 使用預先認可和用於程式碼品質保證、安全性掃描和單元測試的工具，自動設定 Python 虛擬環境。如需詳細資訊，請參閱「工具」一節。

先決條件和限制

前提

- 一個活躍的 AWS 帳戶
- 您的工作站上安裝了下列工具：
  - AWS Cloud Development Kit (AWS CDK) 工具組版本
  - 碼頭工人版本
  - Node.js 版本
  - 建立版本
  - Python 版本

限制

- 作業的最大執行階段受 Lambda 函數的最大執行階段限制 (15 分鐘)。
- 同時工作請求的最大數目受 Lambda 函數的保留並行限制。

架構

下圖顯示任務 API 與事件處理和錯誤處理 Lambda 函數的互動，並將事件存放在 Amazon 事件存檔中。 EventBridge

典型的工作流程包括下列步驟：

1. 您可以針對 AWS Identity and Access Management (IAM) 進行驗證並取得安全登入資料。
2. 您可以將 HTTP POST 要求傳送至`/jobs`作業 API 端點，並在要求主體中指定工作參數。
3. 作業 API 是 API Gateway REST API，會傳回包含作業識別碼的 HTTP 回應給您。
4. 工作 API 會以非同步方式叫用事件處理 Lambda 函數。
5. 事件處理函數會處理事件，然後將任務結果放入任務 Amazon DynamoDB 表格
6. 您將 HTTP GET 要求傳送至`/jobs/{jobId}`作業 API 端點，其中包含步驟 3 中的工作識別碼`{jobId}`。

7. 工作 API 會查詢 `jobs` DynamoDB 資料表以擷取工作結果。
8. 工作 API 會傳回包含工作結果的 HTTP 回應。
9. 如果事件處理失敗，事件處理函數會將事件傳送至錯誤處理函數。
10錯誤處理函數會將工作參數置於 `jobs` DynamoDB 表格中。
11您可以透過傳送 HTTP `GET` 要求至作/`jobs/{jobId}`業 API 端點來擷取工作參數。
12如果錯誤處理失敗，錯誤處理函數會將事件傳送至事件封存。 EventBridge

您可以使用重播已封存的事件 EventBridge。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK)是一個軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端 基礎結構。
- AWS Command Line Interface (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，Lambda 函數、使用 API 目標的 HTTP 叫用端點，或其他 AWS 帳戶的事件匯流排。
- AWS Lambda 是一項運算服務，可協助您執行程式碼，無需佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

其他工具

- 自動格式化基於 Python 增強建議（PEP）8 樣式指南的 Python 代碼。
- 強盜掃描 Python 代碼以查找常見的安全問題。
- 命令是一個 Git 提交檢查器和`CHANGELOG`生成器。
- cfn-皮棉是一個棉絨 AWS CloudFormation
- Checkov 是一種靜態代碼分析工具，用於檢查基礎設施即代碼（IaC）是否存在安全性和合規性錯誤配置。
- jq 是用於解析 JSON 的命令行工具。

- 郵遞員是一個 API 平台。
- 預提交是一個 Git 鉤子管理器。
- Projen 是一個項目生成器。
- pytest 是一個用於編寫小型可讀測試的 Python 框架。

代碼存儲庫

您可以在具有 API Gateway 和 Lambda 儲存庫的 GitHub 非同步事件處理中找到此範例架構程式碼。

最佳實務

- 此範例架構不包括監視已部署的基礎結構。如果您的使用案例需要監視，請評估新增 CDK 監視結構或其他監視解決方案。
- 此範例架構使用 IAM 許可來控制工作 API 的存取權。授權假設的任何人`JobsAPIInvokeRole`都可以叫用作業 API。因此，訪問控制機制是二進制的。如果您的使用案例需要更複雜的授權模型，請使用不同的存取控制機制進行評估。
- 當使用者傳送 HTTP `POST` 要求至`/jobs`作業 API 端點時，輸入資料會在兩個不同層級進行驗證：
  - Amazon API Gateway 負責第一個請求驗證。
  - 事件處理函數執行第二個請求。

    當使用者對`/jobs/{jobId}`作業 API 端點執行 HTTP `GET` 要求時，不會執行任何驗證。如果您的使用案例需要額外的輸入驗證和更高的安全層級，請使用 AWS WAF 進行評估以保護您的 API。

史诗

設定環境

任務	描述	所需技能
複製儲存庫。	若要在本機複製儲存庫，請執行下列命令： `git clone https://github.com/aws-samples/asynchronous-e`	DevOps 工程師

任務	描述	所需技能
	`vent-processing-api-gateway-lambda-cdk.git`	
設置項目。	將目錄更改為存儲庫根目錄，並使用 Projen 設置 Python 虛擬環境和所有工具： `cd asynchronous-event-processing-api-gateway-api-gateway-lambda-cdk` `npx projen`	DevOps 工程師
安裝預先提交掛鉤。	要安裝預先提交掛鉤，請執行以下操作： 1. 啟動虛 Python 環境： `source .env/bin/activate` 2. 安裝預提交掛鉤： `pre-commit install` `pre-commit install --hook-type commit-msg`	DevOps 工程師

部署範例架構

任務	描述	所需技能
引導 AWS CDK。	要 AWS CDK 在您的中引導 AWS 帳戶，請運行以下命令：	AWS DevOps

任務	描述	所需技能
	`AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	
部署範例架構。	若要在您的中部署範例架構 AWS 帳戶，請執行下列命令： `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

測試架構

任務	描述	所需技能
安裝測試先決條件。	在您的工作站上安裝 AWS Command Line Interface（AWS CLI），郵遞員和 jq。 建議使用 Postman 測試此示例體系結構，但不是強制性的。如果您選擇替代 API 測試工具，請確定它支援AWS 簽章版本 4 驗證，並參考可透過匯出 REST API 來檢查的公開 API 端點。	DevOps 工程師
假設`JobsAPIInvokeRole` 。	假設`JobsAPIInvokeRole` 這是從部署命令打印為輸出： `CREDENTIALS=$(AWS_PROFILE=$<YOUR_AWS_PROFILE> aws sts assume-role \ --no-cli-pager \`	AWS DevOps

任務	描述	所需技能
	--role-arn $<JOBS_API_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCESS_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.AccessKeyId') export AWS_SECRET_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.SecretAccessKey') export AWS_SESSION_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.SessionToken')	

任務	描述	所需技能
配置郵遞員。	1. 若要匯入包含在儲存庫中的 Postter 集合，請依照 Postman 文件中的指示進行。 2. 使用下列值設定JobsAPI變數： • accessKey – 來自assume-role 命令的Credentials.AccessKeyId 屬性值 • baseUrl– 從部署命令JobsApiJobsAPIEndpoint 輸出的值，沒有尾隨斜杠 • region– 您部署示例架構的 AWS 區域 位置的價值 • seconds– 範例工作的輸入參數值。它必須是一個正整數 • secretKey – 來自assume-role 命令的Credentials.SecretAccessKey 屬性值 • sessionToken – 來自assume-role 命令的Credentials.SessionToken 屬性值	AWS DevOps

任務	描述	所需技能
測試範例架構。	若要測試範例架構，請將要求傳送至作業 API。如需詳細資訊，請參閱郵遞員文件。	DevOps 工程師

故障診斷

問題	解決方案
範例架構的銷毀和後續重新部署失敗，因為 Amazon CloudWatch 日誌記錄群組 `/aws/apigateway/JobsAPIAccessLogs` 已存在。	1. 如有必要，請將您的日誌資料匯出到 Amazon S3。 2. 刪除記 CloudWatch 錄檔記錄群組 `/aws/apigateway/JobsAPIAccessLogs` 。 3. 重新部署範例架構。

相關資源

- API Gateway 對應範本和存取記錄變數參考
- 設定後端 Lambda 函數的非同步叫用

使用 Amazon API Gateway 和亞馬遜動 Amazon DynamoDB 串流非同步處理事件

創建者：安德烈·梅羅尼（AWS），亞歷山德羅·特里索里尼（AWS），納迪姆馬吉德（AWS），瑪麗·克西里（AWS）和邁克爾·沃爾納（AWS）

程式碼儲存庫：使用 API Gateway 和 DynamoDB Streams 進行非同步處理	環境：PoC 或試點	技術：無伺服器
AWS 服務：Amazon API Gateway；Amazon DynamoDB；Amazon DynamoDB 串流；AWS Lambda；Amazon SNS		

Summary

Amazon API Gateway 是一項全受管服務，開發人員可用來建立、發佈、維護、監控和保護任何規模的 API。它可以處理接受和處理多達數十萬個並行 API 呼叫所涉及的工作，包括下列各項：

- 交通管理
- 跨來源資源共用 (CORS) 支援
- 授權和存取控制
- 限流
- 監控
- API 版本管理

API Gateway 的重要服務配額是整合逾時。逾時是 REST API 傳回錯誤之前，後端服務必須傳回回應的時間上限。對於同步工作負載，通常可以接受 29 秒的硬性限制。但是，對於那些想要將 API Gateway 與非同步工作負載搭配使用的開發人員而言，這個限制代表了

此模式顯示使用 API Gateway、Amazon DynamoDB 串流和非同步處理事件的範例架構。 AWS Lambda該架構支持使用相同的輸入參數運 parallel 處理任務，並使用基本的 REST API 作為接口。在

此範例中，使用 Lambda 做為後端，可將工作持續時間限制為 15 分鐘。您可以使用替代服務來處理傳入事件 (例如 AWS Fargate)，以避免此限制。

Projen 用於設置本地開發環境，並將示例架構部署到目標 AWS 帳戶，結合AWS Cloud Development Kit (AWS CDK) 工具包，Docker 和 Node.js。Projen 使用預先認可和用於程式碼品質保證、安全性掃描和單元測試的工具，自動設定 Python 虛擬環境。如需詳細資訊，請參閱「工具」一節。

先決條件和限制

前提

- 一個活躍的 AWS 帳戶
- 工作站上安裝的下列工具：
  - AWS Cloud Development Kit (AWS CDK) 工具組版本 2.85.0 或更新版本
  - 泊塢視窗版本 20.10.21 或更新版本
  - Node.js 版本 18 或更新版本
  - 建立版本 0.71.111 或更新版本
  - Python 版本 3.9.16 或更高版本

限制

- DynamoDB Streams 的建議讀取器數目上限為兩個，以避免節流。
- 作業的最大執行階段受 Lambda 函數的最大執行階段限制 (15 分鐘)。
- 同時工作請求的最大數目受 Lambda 函數的保留並行限制。

架構

架構

下圖顯示了任務 API 與 DynamoDB Streams 的互動，以及事件處理和錯誤處理 Lambda 函數，以及事件存檔中存放的事件。 EventBridge

典型的工作流程包括下列步驟：

1. 您可以針對 AWS Identity and Access Management (IAM) 進行驗證並取得安全登入資料。

2. 您可以將 HTTP POST 要求傳送至/jobs作業 API 端點，並在要求主體中指定工作參數。
3. 作業 API 會傳回包含工作識別碼的 HTTP 回應。
4. 任務 API 會將任務參數置於 jobs_table Amazon DynamoDB 表格中。
5. jobs_tableDynamoDB 資料表會叫用事件處理的 Lambda 函數。
6. 事件處理的 Lambda 函數會處理事件，然後將工作結果放入 jobs_table DynamoDB 表格中。為了確保一致的結果，事件處理功能實現了樂觀的鎖定機制。
7. 您將 HTTP GET 要求傳送至/jobs/{jobId}作業 API 端點，其中包含步驟 3 中的工作識別碼{jobId}。
8. 工作 API 會查詢 jobs_table DynamoDB 表格以擷取工作結果。
9. 工作 API 會傳回包含工作結果的 HTTP 回應。
10. 如果事件處理失敗，事件處理函數的來源對應會將事件傳送到錯誤處理 Amazon Simple Notification Service (Amazon SNS) 主題。
11. 錯誤處理 SNS 主題會以非同步方式將事件推送至錯誤處理函數。
12. 錯誤處理函數會將工作參數置於 jobs_table DynamoDB 表格中。

    您可以透過傳送 HTTP GET 要求至作/jobs/{jobId}業 API 端點來擷取工作參數。
13. 如果錯誤處理失敗，錯誤處理函數會將事件傳送到 Amazon EventBridge 存檔。

    您可以使用重播已封存的事件 EventBridge。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK)是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，AWS Lambda 函數、使用 API 目的地的 HTTP 叫用端點，或其他 AWS 帳戶中的事件匯流排。
- AWS Lambda 是一項運算服務，可協助您執行程式碼，無需佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。

其他工具

- 自動格式化基於 Python 增強建議（PEP）8 樣式指南的 Python 代碼。
- 強盜掃描 Python 代碼以查找常見的安全問題。
- 命令是一個 Git 提交檢查器和CHANGELOG生成器。
- cfn-皮棉是一個棉絨 AWS CloudFormation
- Checkov 是一種靜態代碼分析工具，用於檢查基礎設施即代碼（IaC）是否存在安全性和合規性錯誤配置。
- jq 是用於解析 JSON 的命令行工具。
- 郵遞員是一個 API 平台。
- 預提交是一個 Git 鉤子管理器。
- Projen 是一個項目生成器。
- pytest 是一個用於編寫小型可讀測試的 Python 框架。

代碼存儲庫

您可以在使用 API Gateway 和 DynamoDB Streams 的 GitHub 非同步處理存放庫中找到此架構程式碼範例。

最佳實務

- 此範例架構不包括監視已部署的基礎結構。如果您的使用案例需要監視，請評估新增 CDK 監視結構或其他監視解決方案。
- 此範例架構使用 IAM 許可來控制工作 API 的存取權。授權假設的任何人JobsAPIInvokeRole都可以叫用作業 API。因此，訪問控制機制是二進制的。如果您的使用案例需要更複雜的授權模型，請使用不同的存取控制機制進行評估。
- 當使用者傳送 HTTP POST 要求至/jobs作業 API 端點時，輸入資料會在兩個不同層級進行驗證：
  - API Gateway 負責第一個請求驗證。
  - 事件處理函數執行第二個請求。

    當使用者對/jobs/{jobId}作業 API 端點執行 HTTP GET 要求時，不會執行任何驗證。如果您的使用案例需要額外的輸入驗證和更高的安全層級，請評估使用 AWS WAF 以保護您的 API。

- 為了避免限制，DynamoDB Streams 文件不鼓勵使用者讀取來自同一串流碎片的兩個以上的取用者。若要擴展消費者數量，我們建議您使用 Amazon Kinesis Data Streams。

- 此範例中已使用最佳鎖定來確保 jobs_table DynamoDB 表中項目的一致性更新。根據使用案例需求，您可能需要實作更可靠的鎖定機制，例如悲觀鎖定。

史诗

設定環境

任務	描述	所需技能
複製儲存庫。	若要在本機複製儲存庫，請執行下列命令： `git clone https://github.com/aws-samples/asynchronous-event-processing-api-gateway-dynamodb-streams-cdk.git`	DevOps 工程師
設置項目。	將目錄更改為存儲庫根目錄，並使用 Projen 設置 Python 虛擬環境和所有工具： `cd asynchronous-event-processing-api-gateway-api-gateway-dynamodb-streams-cdk` `npx projen`	DevOps 工程師
安裝預先提交掛鉤。	要安裝預先提交掛鉤，請執行以下操作： 1. 啟動虛 Python 環境： `source .env/bin/activate` 2. 安裝預提交掛鉤：	DevOps 工程師

任務	描述	所需技能
	`pre-commit install` `pre-commit install --hook-type commit-msg`	

部署範例架構

任務	描述	所需技能
引導 AWS CDK。	要AWS CDK在您的中引導 AWS 帳戶，請運行以下命令： `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	AWS DevOps
部署範例架構。	若要在您的中部署範例架構 AWS 帳戶，請執行下列命令： `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

測試架構

任務	描述	所需技能
安裝測試先決條件。	在您的工作站上安裝 AWS Command Line Interface（AWS CLI），郵遞員和 jq。 建議使用 Postman 測試此示例體系結構，但不是強制性的。如果您選擇替代 API 測試工	DevOps 工程師

任務	描述	所需技能
	具，請確定其支援 AWS 簽名版本 4 身份驗證，並參考可透過匯出 REST API 來檢查的暴露 API 端點。	
假設JobsAPIInvokeRole 。	假設JobsAPIInvokeRole 從deploy命令打印為輸出： `CREDENTIALS=$(AWS_PROFILE=$<YOUR_AWS_PROFILE> aws sts assume-role \ --no-cli-pager \ --role-arn $<JOBS_API_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCESS_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.AccessKeyId') export AWS_SECRET_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.SecretAccessKey') export AWS_SESSION_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.SessionToken')`	AWS DevOps

任務	描述	所需技能
配置郵遞員。	• 若要匯入包含在儲存庫中的 Postter 集合，請依照 Postman 文件中的指示進行。 • 使用下列值設定JobsAPI變數： • accessKey – 指assume-role 令中的Credentials.AccessKeyId 屬性值。 • baseUrl– 來自deploy命令的JobsApiJobsAPIEndpoint 輸出值，不帶尾隨斜線。 • region– 您部署範例架構的 AWS 區域 位置值。 • seconds– 範例工作的輸入參數值。它必須是一個正整數。 • secretKey – 指assume-role 令中的Credentials.SecretAccessKey 屬性值。 • sessionToken – 指assume-role 令中的Credentials.SessionToken 屬性值。	AWS DevOps

任務	描述	所需技能
測試範例架構。	若要測試範例架構，請將要求傳送至作業 API。如需詳細資訊，請參閱郵遞員文件。	DevOps 工程師

故障診斷

問題	解決方案
範例架構的銷毀和後續重新部署失敗，因為 Amazon CloudWatch 日誌記錄群組 `/aws/apigateway/JobsAPIAccessLogs` 已存在。	1. 如有必要，請將日誌資料匯出至亞馬遜簡易儲存服務 (Amazon S3)。 2. 刪除記 CloudWatch 錄檔記錄群組 `/aws/apigateway/JobsAPIAccessLogs` 。 3. 重新部署範例架構。

相關資源

- API Gateway 對應範本和存取記錄變數參考
- 變更動態資料擷取的資料擷取
- 樂觀鎖定版本號
- 使用 Kinesis Data Streams 擷取 DynamoDB 的變更

使用 Amazon API Gateway、Amazon SQS 和 AWS Fargate 非同步處理事件

創建者：安德烈·梅羅尼（AWS），亞歷山德羅·特里索里尼（AWS），納迪姆馬吉德（AWS），瑪麗·克西里（AWS）和邁克爾·沃爾納（AWS）

程式碼儲存庫：使用 API Gateway 和 SQS 進行非同步事件處理	環境：PoC 或試點	技術：無伺服器
AWS 服務：Amazon API Gateway；Amazon DynamoDB；AWS Fargate；Amazon SQS；AWS Lambda		

Summary

Amazon API Gateway 是一項全受管服務，開發人員可用來建立、發佈、維護、監控和保護任何規模的 API。它可以處理接受和處理多達數十萬個並行 API 呼叫所涉及的工作，包括下列各項：

- 交通管理
- 跨來源資源共用 (CORS) 支援
- 授權和存取控制
- 限流
- 監控
- API 版本管理

API Gateway 的重要服務配額是整合逾時。逾時是 REST API 傳回錯誤之前，後端服務必須傳回回應的時間上限。對於同步工作負載，通常可以接受 29 秒的硬性限制。但是，對於那些想要將 API Gateway 與非同步工作負載搭配使用的開發人員而言，這個限制代表了

此模式顯示使用 API Gateway、Amazon Simple Queue Service (Amazon SQS) 和非同步處理事件的範例架構。 AWS Fargate該架構支持在沒有持續時間限制的情況下運行處理作業，並使用基本的 REST API 作為接口。

Projen 可用來設定本機開發環境，並將範例架構部署到目標 AWS 帳戶，並結合使用 AWS Cloud Development Kit (AWS CDK)Docker 和 Node.js。Projen 使用預先認可和用於程式碼品質保證、安全性掃描和單元測試的工具，自動設定 Python 虛擬環境。如需詳細資訊，請參閱「工具」一節。

先決條件和限制

先決條件

- 一個活躍的 AWS 帳戶
- 工作站上安裝的下列工具：
  - AWS Cloud Development Kit (AWS CDK) 工具組版本 2.85.0 或更新版本
  - 泊塢視窗版本 20.10.21 或更新版本
  - Node.js 版本 18 或更新版本
  - 建立版本 0.71.111 或更新版本
  - Python 版本 3.9.16 或更高版本

限制

- 並行作業限制為每分鐘 500 個任務，這是 Fargate 可以佈建的最大任務數量。

架構

下圖顯示任務 API 與 `jobs` Amazon DynamoDB 表、事件處理 Fargate 服務以及錯誤處理功能的互動。 AWS Lambda 事件存儲在 Amazon EventBridge 事件存檔中。

典型的工作流程包括下列步驟：

1. 您可以針對 AWS Identity and Access Management (IAM) 進行驗證並取得安全登入資料。
2. 您可以將 HTTP `POST` 要求傳送至`/jobs`作業 API 端點，並在要求主體中指定工作參數。
3. 作業 API 是 API Gateway REST API，會傳回包含作業識別碼的 HTTP 回應給您。
4. 工作 API 會將訊息傳送至 SQS 佇列。
5. Fargate 會從 SQS 佇列中提取訊息、處理事件，然後將工作結果放入 `jobs` DynamoDB 表格中。
6. 您將 HTTP `GET` 要求傳送至`/jobs/{jobId}`作業 API 端點，其中包含步驟 3 中的工作識別碼`{jobId}`。
7. 工作 API 會查詢 `jobs` DynamoDB 表格以擷取工作結果。

8. 工作 API 會傳回包含工作結果的 HTTP 回應。
9. 如果事件處理失敗，SQS 佇列會將事件傳送至無效字母佇列 (DLQ)。
10EventBridge 事件會起始錯誤處理功能。
11錯誤處理函數會將工作參數置於 `jobs` DynamoDB 表格中。
12您可以透過傳送 HTTP `GET` 要求至作`/jobs/{jobId}`業 API 端點來擷取工作參數。
13如果錯誤處理失敗，錯誤處理函數會將事件傳送至歸檔。 EventBridge

您可以使用重播已封存的事件 EventBridge。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK)是一個軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端 基礎結構。
- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。
- AWS Fargate協助您執行容器，而不需要管理伺服器或 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。它與 Amazon Elastic Container Service (Amazon ECS) 一起使用。
- Amazon EventBridge 是無伺服器事件匯流排服務，可協助您將應用程式與來自各種來源的即時資料連接起來。例如，Lambda 函數、使用 API 目標的 HTTP 叫用端點，或其他 AWS 帳戶的事件匯流排。
- AWS Lambda 是一項運算服務，可協助您執行程式碼，無需佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。

其他工具

- 自動格式化基於 Python 增強建議（PEP）8 樣式指南的 Python 代碼。
- 強盜掃描 Python 代碼以查找常見的安全問題。
- 命令是一個 Git 提交檢查器和CHANGELOG生成器。
- cfn-皮棉是一個棉絨 AWS CloudFormation
- Checkov 是一種靜態代碼分析工具，用於檢查基礎設施即代碼（IaC）是否存在安全性和合規性錯誤配置。

- jq 是用於解析 JSON 的命令行工具。
- 郵遞員是 API 平台。
- 預提交是一個 Git 鉤子管理器。
- Projen 是一個項目生成器。
- pytest 是一個用於編寫小型可讀測試的 Python 框架。

代碼存儲庫

您可以在使用 API Gateway 和 SQS 儲存庫的 GitHub 非同步處理中找到此範例架構程式碼。

最佳實務

- 此範例架構不包括監視已部署的基礎結構。如果您的使用案例需要監視，請評估新增 CDK 監視結構或其他監視解決方案。
- 此範例架構使用 IAM 許可來控制工作 API 的存取權。授權假設的任何人`JobsAPIInvokeRole`都可以叫用作業 API。因此，訪問控制機制是二進制的。如果您的使用案例需要更複雜的授權模型，請使用不同的存取控制機制進行評估。
- 當使用者傳送 HTTP `POST` 要求至`/jobs`作業 API 端點時，會在兩個不同層級驗證輸入資料：
  - API Gateway 負責第一個請求驗證。
  - 事件處理函數執行第二個請求。

    當使用者對`/jobs/{jobId}`作業 API 端點執行 HTTP `GET` 要求時，不會執行任何驗證。如果您的使用案例需要額外的輸入驗證和更高的安全層級，請評估使用 AWS WAF 以保護您的 API。

史诗

設定環境

任務	描述	所需技能
複製儲存庫。	若要在本機複製儲存庫，請執行下列命令： `git clone https://github.com/aws-samples/asynchronous-e`	DevOps 工程師

任務	描述	所需技能
	`vent-processing-api-gateway-sqs-cdk.git`	
設置項目。	將目錄更改為存儲庫根目錄，並使用 Projen 設置 Python 虛擬環境和所有工具： `cd asynchronous-event-processing-api-gateway-api-gateway-sqs-cdk` `npx projen`	DevOps 工程師
安裝預先提交掛鉤。	要安裝預先提交掛鉤，請執行以下操作： 1. 啟動虛 Python 環境： `source .env/bin/activate` 2. 安裝預提交掛鉤： `pre-commit install` `pre-commit install --hook-type commit-msg`	DevOps 工程師

部署範例架構

任務	描述	所需技能
引導 AWS CDK。	要AWS CDK在您的中引導 AWS 帳戶，請運行以下命令：	AWS DevOps

任務	描述	所需技能
	`AWS_PROFILE=$YOUR_AWS_PROFILE npx projen bootstrap`	
部署範例架構。	若要在您的中部署範例架構 AWS 帳戶，請執行下列命令： `AWS_PROFILE=$YOUR_AWS_PROFILE npx projen deploy`	AWS DevOps

測試架構

任務	描述	所需技能
安裝測試先決條件。	在您的工作站上安裝 AWS Command Line Interface（AWS CLI），郵遞員和 jq。 建議使用 Postman 測試此示例體系結構，但不是強制性的。如果您選擇替代 API 測試工具，請確定它支援AWS 簽章版本 4 驗證，並參考可透過匯出 REST API 來檢查的公開 API 端點。	DevOps 工程師
假設`JobsAPIInvokeRole` 。	假設`JobsAPIInvokeRole` 從`deploy`命令打印為輸出： `CREDENTIALS=$(AWS_PROFILE=$<YOUR_AWS_PROFILE> aws sts assume-role \ --no-cli-pager \`	AWS DevOps

任務	描述	所需技能
	--role-arn $<JOBS_API_INVOKE_ROLE_ARN> \ --role-session-name JobsAPIInvoke) export AWS_ACCESS_KEY_ID=$(cat $CREDENTIALS \| jq '.Credentials''.AccessKeyId') export AWS_SECRET_ACCESS_KEY=$(cat $CREDENTIALS \| jq '.Credentials''.SecretAccessKey') export AWS_SESSION_TOKEN==$(cat $CREDENTIALS | jq '.Credentials''.SessionToken')	

任務	描述	所需技能
配置郵遞員。	• 若要匯入包含在儲存庫中的 Postter 集合，請依照 Postman 文件中的指示進行。 • 使用下列值設定JobsAPI變數： • accessKey – 指assume-role 令中的Credentials.AccessKeyId 屬性值。 • baseUrl– 指deploy令JobsApiJobsAPIEndpoint 輸出的值，不含尾隨斜線。 • region– 您部署範例架構的 AWS 區域 位置值。 • seconds– 範例工作的輸入參數值。它必須是一個正整數。 • secretKey – 指assume-role 令中的Credentials.Secret AccessKey 屬性值。 • sessionToken – 指assume-role 令中的Credentials.SessionToken 屬性值。	AWS DevOps

任務	描述	所需技能
測試範例架構。	若要測試範例架構，請將要求傳送至作業 API。如需詳細資訊，請參閱郵遞員文件。	DevOps 工程師

故障診斷

問題	解決方案
範例架構的銷毀和後續重新部署失敗，因為 Amazon CloudWatch 日誌記錄群組/aws/apigateway/JobsAPIAccessLogs 已存在。	1. 如有必要，請將日誌資料匯出至亞馬遜簡易儲存服務 (Amazon S3)。 2. 刪除記 CloudWatch 錄檔記錄群組/aws/apigateway/JobsAPIAccessLogs 。 3. 重新部署範例架構。
範例架構的銷毀和後續重新部署失敗，因為CloudWatch 記錄檔記錄群組/aws/ecs/EventProcessingServiceLogs 已存在。	1. 如有必要，請將您的日誌資料匯出到 Amazon S3。 2. 刪除記 CloudWatch 錄檔記錄群組 /aws/ecs/EventProcessingServiceLogs. 3. 重新部署範例架構。

相關資源

- API Gateway 對應範本和存取記錄變數參考
- 如何將 API Gateway REST API 與 Amazon SQS 整合並解決常見錯誤？

從 AWS Step Functions 同步執行 AWS Systems Manager Automation 任務

創建者埃利·埃爾庫里（AWS）

代碼存儲庫：amazon-stepfunctions-ssm-waitfortasktoken	環境：生產	技術：無伺服器 DevOps；終端使用者運算；營運
AWS 服務：AWS Step Functions；AWS Systems Manager		

Summary

此模式說明如何 AWS Step Functions 與整合 AWS Systems Manager。它會使用 AWS SDK 服務整合，以狀態機器工作流程的工作權杖呼叫 Systems Manager startAutomationExecutionAPI，並暫停直到權杖傳回成功或失敗呼叫為止。為了示範整合，此模式會在或文件周圍實作自動化文件 (runbook) 包裝函式，並使`.waitForTaskToken`用同步呼叫`AWS-RunShellScript`AWS-RunShellScript或`AWS-RunPowerShellScript`。`AWS-RunPowerShellScript`如需有關 Step Functions 中 AWS SDK 服務整合的詳細資訊，請參閱開AWS Step Functions 發人員指南。

Step Functions 是一種低程式碼的視覺化工作流程服務，您可以使用服務來建置分散式應用程式、自動化 IT 和商 AWS 務程序，以及建置資料和機器學習管道。工作流程可管理失敗、重試、平行化、服務整合和可觀察性，因此您可以專注於更高價值的商務邏輯。

具備自動化功能，可簡化常見的 AWS Systems Manager維護、部署和修復任務， AWS 服務 例如亞馬遜彈性運算雲端 (Amazon EC2)、Amazon Relational Database Service 服務 (Amazon RDS)、Amazon Redshift 和亞馬遜簡單儲存服務 (Amazon S3)。自動化可讓您精細控制自動化作業的並行性。例如，您可以指定要同時鎖定多少資源，以及在停止自動化操作之前可能會發生多少錯誤。

如需實作詳細資訊，包括 runbook 步驟、參數和範例，請參閱其他資訊一節。

先決條件和限制

先決條件

- 活躍 AWS 帳戶
- AWS Identity and Access Management (IAM) 存取 Step Functions 和 Systems Manager 的許可
- 在執行個體上安裝了系統管理員代理程式 (SSM 代理程式) 的 EC2 執行個體
- Systems Manager 的 IAM 執行個體設定檔，附加至您計劃執行工作流程簿的執行個體
- 具有下列 IAM 許可 (遵循最低權限原則) 的 Step Functions 角色：

```
{
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "*"
 }
```

產品版本

- SSM 文件結構描述版本 0.3 或更新版本
- SSM 代理程式版本 2.3.672.0 或更新版本

架構

目標技術堆疊

- AWS Step Functions
- AWS Systems Manager 自動化

目標架構

自動化和規模

- 此 AWS CloudFormation 模式提供可用於在多個執行個體上部署 Runbook 的範本。（請參閱 GitHub Step Functions 和 Systems Manager 實現存儲庫。）

工具

AWS 服務

- AWS CloudFormation協助您設定 AWS 資源、快速且一致地佈建資源，以及跨區域的整個生命週期進 AWS 帳戶 行管理。
- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。
- AWS Step Functions是一項無伺服器協調服務，可協助您結合 AWS Lambda 功能與其他功能，AWS 服務 以建置關鍵業務應用程式。
- AWS Systems Manager協助您管理在 AWS 雲端. 它可簡化應用程式和資源管理、縮短偵測和解決作業問題的時間，並協助您大規模安全地管理 AWS 資源。

Code

此模式的代碼在 GitHub Step Functions 和 Systems Manager 實現存儲庫中可用。

史诗

建立手冊

任務	描述	所需技能
下載 CloudFormation 範本。	從 GitHub 存放庫的 資`cloudformation` 料夾下載`ssm-automation-documents.cfn.json` 範本。	AWS DevOps
創建手冊。	登入 AWS Management Console、開啟AWS CloudFormation 主控台，然後部署範本。如需部署 CloudFormation 範本的詳細資訊，請參閱 CloudFormation 說明文件中的在 AWS CloudFormation 主控台上建立堆疊。 該 CloudFormation 模板部署了三種資源：	AWS DevOps

任務	描述	所需技能
	• SfnRunCommandByInstanceIds — Runbook，可讓您執行AWS-RunShellScript 或使用執AWS-RunPowerShellScript 行個體 ID。 • SfnRunCommandByTargets — Runbook，讓你運行AWS-RunShellScript 或AWS-RunPowerShellScript 通過使用目標. • SSMSyncRole — 由手冊承擔的 IAM 角色。	

建立樣本狀態機

任務	描述	所需技能
創建一個測試狀態機器。	按照開AWS Step Functions 發人員指南中的說明創建和運行狀態機器。對於定義，請使用下面的代碼。請務必使用您帳戶中啟用系統管理員之有效執行個體的 ID 來更新該InstanceIds 值。 `{ "Comment": "A description of my state machine", "StartAt": "StartAutomationWaitForCallBack", "States": {`	AWS DevOps

任務	描述	所需技能
	"StartAutomationWaitForCallBack": { "Type": "Task", "Resource": "arn:aws:states:::aws-sdk:ssm:startAutomationExecution.waitForTaskToken", "Parameters": { "DocumentName": "SfnRunCommandByInstanceIds", "Parameters": { "InstanceIds": ["i-1234567890abcdef0"], "taskToken.$": "States.Array($$.Task.Token)", "workingDirectory": ["/home/ssm-user/"], "Commands": ["echo \"This is a test running automation waitForTaskToken\" >> automation.log", "sleep 100"], "executionTimeout": ["10800"], "deliveryTimeout": ["30"],	

任務	描述	所需技能
	<pre>"shell": ["Shell"] } }, "End": true } } }</pre> 此代碼調用 runbook 來運行演示`waitForTaskToken` 調用 Systems Manager 自動化兩個命令。 `shell`參數值 (`Shell`或`PowerShell`) 決定自動化文件是否執行`AWS-RunShellScript` 或`AWS-RunPowerShellScript` 。 工作會將「這是執行自動化 waitForTask Token 的測試」寫入`/home/ssm-user/automation.log` 檔案中，然後在回應工作權杖之前休眠 100 秒，並釋放工作流程中的下一個工作。 如果您想要呼叫 `SfnRunCommandByTargets` runbook，請使用下列命令取代先前程式碼的`Parameters` 區段： <pre>"Parameters": { "Targets": [{	

任務	描述	所需技能
	<pre>"Key": "InstanceIds", "Values": ["i-02573cafcfEXAMP LE", "i-0471e04240EXAMPLE"] }],</pre>	
更新狀態機器的 IAM 角色。	上一個步驟會自動為狀態機器建立專用的 IAM 角色。但是，它不會授予調用 runbook 的權限。透過新增下列權限來更新角色： <pre>{ "Effect": "Allow", "Action": "ssm:StartAutomati onExecution", "Resource": "*" }</pre>	AWS DevOps
驗證同步呼叫。	執行狀態機，以驗證 Step Functions 和 Systems Manager 自動化之間的同步呼叫。 如需範例輸出，請參閱其他資訊一節。	AWS DevOps

相關資源

- 開始使用 AWS Step Functions(開AWS Step Functions發人員指南)
- 使用任務令牌等待回調（AWS Step Functions 開發人員指南，服務集成模式）
- 發送任務成功和發送任務失敗 API 調用（博托 3 文檔）
- AWS Systems Manager 自動化 (AWS Systems Manager 使用者指南)

其他資訊

實施細節

此 CloudFormation 模式提供了部署兩個 Systems Manager 手冊的範本：

- SfnRunCommandByInstanceIds使用執行個體 ID 執行AWS-RunShellScript或AWS-RunPowerShellScript命令。
- SfnRunCommandByTargets使用目標執行AWS-RunShellScript或AWS-RunPowerShellScript命令。

每個 runbook 實現四個步驟來實現使用步驟函數中的.waitForTaskToken選項時，同步調用。

Step (步驟)	Action	Description
1	Branch	檢查shell參數值 (Shell或PowerShell），以決定是AWS-RunShellScript 針對 Linux 執行還是針AWS-RunPowerShellScript 對視窗執行。
2	RunCommand_Shell 或 RunCommand_PowerShell	接受數個輸入並執行RunShellScript 或RunPowerShellScript 指令。如需詳細資訊，請檢查 Systems Manager 主

		控台上的RunCommand_Shell 或RunCommand_PowerShell 自動化文件的 [詳細資料] 索引標籤。
3	SendTaskFailure	中止或取消步驟 2 時執行。它調用 Step Functions 數 send_task_failure API，它接受三個參數作為輸入：狀態機器傳遞的令牌，失敗錯誤以及失敗原因的描述。
4	SendTaskSuccess	當步驟 2 成功時執行。它調用 Step Functions send_task_success API，它接受由狀態機作為輸入傳遞的令牌。

手冊參數

SfnRunCommandByInstanceIds手冊:

參數名稱	類型	選擇性或必要	Description
shell	字串	必要	用來決定是否要AWS-RunShellScript 針對 Linux 執行或視窗執行的執AWS-RunPowerShellScript 行個體殼層。
deliveryTimeout	Integer	選用	等待命令傳遞至執行個體上 SSM 代理程式的時間 (秒)。此參數的最小值為 30 (0.5 分鐘)，最大值為 2592000 (720 小時)。

executionTimeout	字串	選用	指令在被視為失敗之前完成的時間 (以秒為單位)。預設值為 3600 (1 小時)。最大值為一七二八百 (48 小時) 。
workingDirectory	字串	選用	在您的執行個體上的工作目錄路徑。
Commands	StringList	必要	要執行的殼層指令碼或命令。
InstanceIds	StringList	必要	您要在其中執行命令的執行個體 ID。
taskToken	字串	必要	用於回呼回應的工作 Token。

SfnRunCommandByTargets手冊:

名稱	類型	選擇性或必要	Description
shell	字串	必要	用來決定是否要AWS-RunShellScript 針對 Linux 執行或視窗執行的執AWS-RunPowerShellScript 行個體殼層。
deliveryTimeout	Integer	選用	等待命令傳遞至執行個體上 SSM 代理程式的時間 (秒)。此參數的最小值為 30 (0.5 分鐘)，最大值為 2592000 (720 小時)。

executionTimeout	Integer	選用	指令在被視為失敗之前完成的時間 (以秒為單位)。預設值為 3600 (1 小時)。最大值為一七二八百 (48 小時) 。
workingDirectory	字串	選用	在您的執行個體上的工作目錄路徑。
Commands	StringList	必要	要執行的殼層指令碼或命令。
Targets	MapList	必要	搜尋準則陣列，可使用您指定的索引鍵值配對來識別執行個體。例如：[{"Key":"InstanceIds","Values":["i-02573cafcfEXAMPLE","i-0471e04240EXAMPLE"]}]
taskToken	字串	必要	用於回呼回應的工作 Token。

範例輸出

下表提供了從 step 函數的示例輸出。它顯示步驟 5 (TaskSubmitted) 和步驟 6 (TaskSucceeded) 之間的總運行時間超過 100 秒。這表明 step 函數在移動到工作流程中的下一個任務之前等待sleep 100命令完成。

ID	類型	Step (步驟)	Resource	經過時間 (毫秒)	Timestamp

1	Execution Started		-	0	二○二二年三月十一日下午五十五時三十三分
2	TaskState Entered	StartAutomationWaitForCallBack	-	40	二○二二年三月十一日下午五十五時三十四分
3	TaskScheduled	StartAutomationWaitForCallBack	-	40	二○二二年三月十一日下午五十五時三十四分
4	TaskStarted	StartAutomationWaitForCallBack	-	154	二○二二年三月十一日下午五時五十四分
5	TaskSubmitted	StartAutomationWaitForCallBack	-	657	二○二二年三月十一日下午五時五十分
6	TaskSucceeded	StartAutomationWaitForCallBack	-	103835	二○二二年三月十一日下午五時二十八分
7	TaskState Exited	StartAutomationWaitForCallBack	-	103860	二○二二年三月十一日下午五時二十六分
8	Execution Succeeded		-	103897	二○二二年三月十一日下午五時二十分

在 AWS Lambda 函數中使用 Python 來執行 S3 物件的 parallel 讀取

創建者愛德華多·博爾托盧齊

代碼存儲庫：aws-lambda-parallel-download	環境：PoC 或試點	技術：無伺服器
AWS 服務：AWS Lambda；Amazon S3；AWS Step Functions		

Summary

您可以使用此模式從 Amazon Simple Storage Service (Amazon S3) 儲存貯體即時擷取和摘要文件清單。該模式提供範例程式碼，以 parallel 讀取 Amazon Web Services (AWS) 上的 S3 儲存貯體中的物件。該模式展示瞭如何使用 Python 使用 AWS Lambda 函數有效率地執行 I/O 繫結任務。

一家金融公司在交互式解決方案中使用此模式來實時手動批准或拒絕相關的金融交易。金融交易文件存放在與市場相關的 S3 儲存貯體中。操作員從 S3 儲存貯體選取文件清單，分析了解決方案計算的交易總價值，並決定核准或拒絕選取的批次。

I/O 綁定任務支持多個線程。在此示例代碼中，並發的 .期貨. ThreadPoolExecutor最多可與 1,000 個同時執行緒搭配使用。Lambda 函數最多支援 1,024 個執行緒，其中一個執行緒是您的主要程序。您還需要增加中的集區連線上限，以botocore便所有執行緒都可以同時執行 S3 物件下載。

範例程式碼在 S3 儲存貯體中使用一個 8.3 KB 物件和 JSON 資料。物件會讀取多次。Lambda 函數讀取物件之後，JSON 資料會解碼為 Python 物件。執行此範例後的結果是，使用設定了 2,048 MB 記憶體的 Lambda 函數，在 2.3 秒內處理 1,000 次讀取，並在 26 秒內處理 10,000 次讀取。增加 Lambda 記憶體無助於縮短執行工作的時間。

AWS Lambda 功率調整工具可用來測試不同的 Lambda 記憶體組態，並確認工作的最佳 performance-to-cost 比例。如需測試結果，請參閱其他資訊一節。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 熟練掌握 Python 開發

限制

- Lambda 函數最多可以有 1,024 個執行程序或執行緒。
- 新的 AWS 帳戶的記憶體限制為 3,008 MB。相應地調整 AWS Lambda 功率調整工具。如需詳細資訊，請參閱疑難排解一節。
- Python 版本 3.8 是最低推薦版本，因為它引入了線程執行池中的線程重用。
- Amazon S3 的每個分區前綴每秒有 5,500 個 GET/HEAD 請求的限制。

產品版本

- Python 3.8 或更高版本
- AWS Cloud Development Kit (AWS CDK) v2
- AWS Command Line Interface (AWS CLI) 版本 2
- AWS Lambda 功率調整 4.3.3 (選用)

架構

目標技術堆疊

- AWS Lambda
- Amazon S3
- AWS Step Functions (如果已部署 AWS Lambda 功率調整)

目標架構

下圖顯示 Lambda 函數，可 parallel 讀取 S3 儲存貯體中的物件。此圖表也有 AWS Lambda 功率調整工具的步驟函數工作流程，可微調 Lambda 函數記憶體。此微調有助於在成本和效能之間取得良好的平衡。

自動化和規模

必要時，Lambda 函數可以快速擴展。若要避免 Amazon S3 在高需求期間造成 503 個減速錯誤，我們建議對擴展進行一些限制。

工具

AWS 服務

- AWS Cloud Development Kit (AWS CDK) v2 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。範例基礎設施是為了使用 AWS CDK 進行部署而建立的。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。在此模式中，AWS CLI 第 2 版用於上傳範例 JSON 檔案。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Step Functions 是一種無伺服器協調服務，可協助您結合 AWS Lambda 函數和其他 AWS 服務來建立關鍵業務應用程式。

其他工具

- Python 是一種通用的計算機編程語言閒置工作線程的重用是在 Python 版本 3.8 中引入的，並且此模式中的 Lambda 函數代碼是為此版本創建的。

代碼存儲庫

此模式的代碼可在aws-lambda-parallel-download GitHub 存儲庫中找到。

最佳實務

- 此 AWS CDK 建構依賴您 AWS 帳戶的使用者許可來部署基礎設施。如果您計劃使用 AWS CDK Pipelines 或跨帳戶部署，請參閱堆疊合成器。
- 此範例應用程式未在 S3 儲存貯體啟用存取日誌。在生產代碼中啟用訪問日誌是最佳實踐。

史诗

準備開發環境

任務	描述	所需技能
檢查已 Python 裝的版本。	提供的代碼是在 Python 3.8 及更高版本上創建和測試的。若要驗證您已安裝的 Python 版本，請執行`python3 -V`。如有需要，請下載並安裝較新的版本。 若要驗證是否已安裝所需的模組，請執行`python3 -c "import pip, venv"`。如果安裝了模塊，則不會返回錯誤。	雲端架構師
安裝和設定 AWS CDK。	若要安裝 AWS CDK 並在尚未設定的情況下啟動它，請按照 AWS CDK 入門中的指示進行操作。若要確認已安裝的 AWS CDK 版本為 2.0 或更新版本，請執行`cdk --version`： 啟動載入時，請將`--cloudformation-execution-policies "arn:aws:iam::aws:policy/job-function/ViewOnlyAccess"` 參數傳遞給。`cdk bootstrap` 這個範例不會使用定義的角色來部署堆疊，而且這個參數會讓您的部署更安全。	雲端架構師

克隆示例存儲庫

任務	描述	所需技能
複製儲存庫。	若要複製最新版本的存放庫，請執行下列命令： `git clone --depth 1 --branch v1.1.2 \ git@github.com:aws-samples/aws-lambda-parallel-download.git`	雲端架構師
將工作目錄變更為複製的儲存庫。	執行以下命令： `cd aws-lambda-paralle l-download`	雲端架構師
建立虛 Python 環境。	若要建立 Python 虛擬環境，請執行下列命令： `python3 -m venv .venv`	雲端架構師
啟用虛擬環境。	若要啟動虛擬環境，請執行下列命令： `source .venv/bin/ activate`	雲端架構師
安裝依賴關係。	要安裝 Python 依賴關係，請運行以下`pip`命令： `pip install -r requireme nts.txt`	雲端架構師
瀏覽代碼。	(選擇性) 從 S3 儲存貯體下載物件的範例程式碼位	雲端架構師

任務	描述	所需技能
	於resources/parallel.py 。 基礎結構程式碼位於資parallel_download 料夾中。	

部署和測試應用程式

任務	描述	所需技能
部署應用程式。	執行 cdk deploy。 記下 AWS CDK 輸出結果： • ParallelDownloadStack.LambdaFunctionARN • ParallelDownloadStack.SampleS3BucketName • ParallelDownloadStack.StateMachineARN	雲端架構師
上傳 JSON 檔案範例。	儲存庫包含一個約 9 KB 的 JSON 檔案範例。若要將檔案上傳到已建立堆疊的 S3 儲存貯體，請執行下列命令： `aws s3 cp sample.json s3://<ParallelDownloadStack.SampleS3BucketName>`	雲端架構師

任務	描述	所需技能
	`<ParallelDownloadStack.SampleS3BucketName>` 使用 AWS CDK 輸出中的對應值取代。	
運行應用程序。	1. 登入 AWS 管理主控台，瀏覽至 Lambda 主控台，然後從 AWS CDK 輸出中找到具有 ARN 的 Lambda 函數。`ParallelDownloadStack.LambdaFunctionARN` 2. 在 [測試] 索引標籤上，將 [事件 JSON] 變更為下列項目： `{"objectKey": "sample.json"}` 3. 選擇 測試。 4. 若要查看結果，請選擇 [詳細資料]。詳細信息將顯示 parallel 下載的統計信息，運行的信息和日誌。	雲端架構師
添加下載次數。	(選擇性) 若要執行 1,500 個取得物件呼叫，請在`Test`參數的事件 JSON 中使用下列 JSON： `{"repeat": 1500, "objectKey": "sample.json"}`	雲端架構師

選用性：執行 AWS Lambda 功率調整

任務	描述	所需技能
執行 AWS Lambda 電源調整工具。	1. 登入主控台，然後瀏覽至「Step Functions」。 2. 從 AWS CDK 輸出中找到具有 ARN 的狀態機器。`ParallelDownloadStack.StateMachineARN` 3. 選擇 [開始執行]，然後貼上下列 JSON： `{ "lambdaARN": "<ParallelDownloadStack.LambdaFunctionARN>", "num": 5, "payload": {"repeat": 2000, "objectKey": "sample.json"} }` 請記住`<ParallelDownloadStack.LambdaFunctionARN>` 用 CDK 輸出中的值替換。 在運行結束時，結果將在執行輸入和輸出選項卡上。	雲端架構師
以圖形檢視 AWS Lambda 功率調整結果。	在「執行輸入和輸出」索引標籤上，複製內`visualiza`	雲端架構師

任務	描述	所需技能
	tion 容連結，然後將其貼到新的瀏覽器索引標籤中。	

清除

任務	描述	所需技能
從 S3 儲存貯體移除物件。	在銷毀已部署的資源之前，請先移除 S3 儲存貯體中的所有物件： `aws s3 rm s3://<ParallelDownloadStack.SampleS3BucketName> \ --recursive` 請記得<ParallelDownloadStack.SampleS3BucketName> 以 AWS CDK 輸出的值取代。	雲端架構師
銷毀資源。	若要銷毀為此試行方案建立的所有資源，請執行下列命令： `cdk destroy`	雲端架構師

故障診斷

問題	解決方案
`'MemorySize' value failed to satisfy constraint: Member must`	對於新帳戶，您可能無法在 Lambda 函數中設定超過 3,008 MB。若要使用 AWS Lambda 功

問題	解決方案
have value less than or equal to 3008	率調整進行測試，請在開始執行 Step Functions 時，在輸入 JSON 中新增下列屬性： `"powerValues": [ 512, 1024, 1536, 2048, 2560, 3008 ]`

相關資源

- Python — 並發. 期貨. ThreadPoolExecutor
- Lambda 配額 — 函數設定、部署和執行
- 在 Python 中使用 AWS CDK
- 使用 AWS Lambda 功率調整的效能分析

其他資訊

Code

下列程式碼片段會執行 parallel I/O 處理：

```
with ThreadPoolExecutor(max_workers=MAX_WORKERS) as executor:
  for result in executor.map(a_function, (the_arguments)):
    ...
```

當執行緒變得可ThreadPoolExecutor用時，會重複使用這些執行緒。

測試和結果

第一個測試處理了 2,500 個物件讀取，結果如下。

從 3,009 MB 開始，任何內存增加的處理時間級別都保持不變，但成本隨著內存大小的增加而增加。

另一項測試調查了 1,536 MB 到 3,072 MB 記憶體之間的範圍，使用 256 MB 的倍數值並處理 10,000 個物件讀取，結果如下。

最好的 performance-to-cost 比例是使用 2,048 MB 記憶體 Lambda 組態。

為了進行比較，2,500 個對象讀取的順序過程需要 40 秒。使用 2,048 MB Lambda 組態的 parallel 程序耗時 5.8 秒，減少了 85% 的時間。

透過 VPC 端點設定對 Amazon S3 儲存貯體的私有存取

由馬丁·馬里奇（AWS），加布里埃爾·羅德里格斯·加西亞（AWS），Shukhrat 霍傑夫（AWS），尼古拉斯·雅各布·貝爾（AWS），莫罕·戈達·普魯索塔瑪（AWS）和華金里諾多（AWS）創建

程式碼儲存庫:私有 S3 VPCE	環境：生產	技術：無伺服器
AWS 服務：Amazon API Gateway；Amazon S3；Amazon VPC；Elastic Load Balancing（ELB）		

Summary

在 Amazon Simple Storage Service (Amazon S3) 中，預先簽署的 URL 可讓您與目標使用者共用任意大小的檔案。根據預設，Amazon S3 預先簽署的 URL 可在到期時間範圍內從網際網路存取，因此使用起來更方便。不過，企業環境通常需要存取 Amazon S3 預先簽署的 URL，才能僅限於私有網路。

此模式提供無伺服器解決方案，可透過使用來自私有網路的預先簽署 URL，而不需要網際網路遍歷，安全地與 S3 物件互動。在架構中，使用者透過內部網域名稱存取 Application Load Balancer。流量會透過 Amazon API Gateway 和 S3 儲存貯體的虛擬私有雲端 (VPC) 端點在內部路由。此 AWS Lambda 功能會透過私有 VPC 端點為檔案下載產生預先簽署的 URL，有助於增強機密資料的安全性和隱私權。

先決條件和限制

先決條件

- 一種 VPC，其中包含部署於連線至公司網路的子網路 (例如，透過 AWS Direct Connect)。 AWS 帳戶

限制

- S3 儲存貯體的名稱必須與網域相同，因此建議您檢查 Amazon S3 儲存貯體命名規則。
- 此範例架構不包含已部署基礎結構的監視功能。如果您的使用案例需要監視，請考慮新增AWS 監視服務。
- 此範例架構不包含輸入驗證。如果您的使用案例需要輸入驗證和更高的安全性層級，請考慮使用 AWS WAF 來保護您的 API。

- 此範例架構不包含 Application Load Balancer 的存取記錄。如果您的使用案例需要存取記錄，請考慮啟用負載平衡器存取記錄。

版本

- Python 版本 3.11 或更高版本
- 地形版本 1.6 或更新版本

架構

目標技術堆疊

目標技術堆疊使用下列 AWS 服務：

- Amazon S3 是用於安全上傳、下載和存放檔案的核心儲存服務。
- Amazon API Gateway 會公開資源和端點，以便與 S3 儲存貯體互動。此服務在產生預先簽署的 URL 以下載或上傳資料方面扮演著重要角色。
- AWS Lambda產生用於從 Amazon S3 下載檔案的預先簽署網址。Lambda 函數是由 API Gateway 調用。
- Amazon VPC 會在虛擬私人 VPC 內部署資源，以提供網路隔離。VPC 包括用於控制流量的子網路和路由表。
- Ap@@ plication Load Balancer 會將傳入流量路由至 API Gateway 或 S3 儲存貯體的 VPC 端點。它允許來自企業網絡的用戶訪問內部資源。
- 適用於 Amazon S3 的 VPC 端點可在 VPC 和 Amazon S3 中的資源之間進行直接、私有的通訊，而無需周遊公用網際網路。
- AWS Identity and Access Management (IAM) 控制對 AWS 資源的存取。設置權限以確保與 API 和其他服務的安全互動。

目標架構

此圖展示了以下要點：

1. 來自企業網路的使用者可以透過內部網域名稱存取應用程式負載平衡器。我們假設公司網路與中的內部網路子網路之間存在連線 AWS 帳戶 (例如，透過 AWS Direct Connect 連線)。

2. 應用程式負載平衡器會將傳入流量路由至 API Gateway，以產生預先簽署的 URL 以將資料下載或上傳到 Amazon S3，或將資料上傳至 S3 儲存貯體的 VPC 端點。在這兩種情況下，要求都會在內部路由，而且不需要周遊網際網路。
3. API Gateway 會公開資源和端點以與 S3 儲存貯體互動。在此範例中，我們提供了從 S3 儲存貯體下載檔案的端點，但也可以延伸這個端點以提供上傳功能。
4. Lambda 函數會產生預先簽署的 URL，以便從 Amazon S3 下載檔案，方法是使用 Application Load Balancer 的網域名稱，而不是公有的 Amazon S3 網域。
5. 使用者會收到預先簽署的 URL，並使用該 URL 使用應用程式負載平衡器從 Amazon S3 下載檔案。負載平衡器包含一個預設路由，可將不適用於 API 的流量傳送至 S3 儲存貯體的 VPC 端點。
6. VPC 端點會將具有自訂網域名稱的預先簽署 URL 路由到 S3 儲存貯體。S3 儲存貯體的名稱必須與網域相同。

自動化和規模

此模式使用 Terraform 將基礎結構從程式碼儲存庫部署到. AWS 帳戶

工具

工具

- Python 是一種通用的計算機編程語言。
- Terraform 是一種基礎結構即程式碼 (IaC) 工具，可協助您建立和管理雲端和內部部署資源。HashiCorp
- AWS Command Line Interface (AWS CLI) 是一個開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。

代碼存儲庫

此模式的程式碼可在 https://github.com/aws-samples/private-s3-vpce 的 GitHub 儲存庫中取得。

最佳實務

此模式的範例架構使用 IAM 許可來控制 API 的存取。任何擁有有效 IAM 登入資料的人都可以呼叫 API。如果您的使用案例需要更複雜的授權模型，您可能需要使用不同的存取控制機制。

史诗

將解決方案部署在 AWS 帳戶

任務	描述	所需技能
取得 AWS 認證。	檢查您的 AWS 憑證和您對帳戶的存取權限。如需指示，請參閱 AWS CLI 文件中的組態和認證檔案設定。	AWS DevOps，一般 AWS
複製儲存庫。	克隆此模式提供的 GitHub 存儲庫： `git clone https://github.com/aws-samples/private-s3-vpce`	AWS DevOps，一般 AWS
設定變數。	1. 在您的計算機上，在 GitHub 存儲庫中，打開文`terraform` 件夾： `cd terraform` 2. 打開`example.tfvars` 文件並根據需要自定義參數。	AWS DevOps，一般 AWS
部署解決方案。	1. 在`terraform` 資料夾中，執行 Terraform 並傳入您自訂的變數： `terraform apply -var-file="example.tfvars"` 2. 確認架構圖中顯示的資源已成功部署。	AWS DevOps，一般 AWS

測試解決方案

任務	描述	所需技能
創建一個測試文件。	將檔案上傳到 Amazon S3 以建立檔案下載的測試案例。您可以使用 Amazon S3 主控台或下列 AWS CLI 命令： `aws s3 cp /path/to/testfile s3://your-bucket-name/testfile`	AWS DevOps，一般 AWS
測試預先簽署的 URL 功能。	1. 使用 aw scurl 傳送要求至「Application Load Balancer」，以建立測試檔案的預先簽署 URL： `awscurl https://your-domain-name/api/get_url?key=testfile` 此步驟會從您的認證建立有效的簽章，並由 API Gateway 驗證。 2. 剖析您從上一步收到的回應中的連結，然後開啟預先簽署的 URL 以下載檔案。	AWS DevOps，一般 AWS
清除。	請務必在不再需要資源時移除這些資源： `terraform destroy`	AWS DevOps，一般 AWS

故障診斷

問題	解決方案
具有特殊字元的 S3 物件金鑰名稱 (例如數字符號 (#) 會中斷 URL 參數並導致錯誤。	正確編碼 URL 參數，並確保 S3 物件金鑰名稱遵循 Amazon S3 準則。

相關資源

Amazon S3：

- 使用預先簽署的 URL 共用物件
- 使用儲存貯體政策控制來自 VPC 端點的存取

Amazon API Gateway：

- 針對 API Gateway 中的私有 API 使用 VPC 端點原則

Application Load Balancer

- 使用 ALB、S3 和 PrivateLink (部AWS 落格文章) 代管內部 HTTPS 靜態網站

使用無伺服器方法將 AWS 服務鏈結在一起

創建者阿尼克特·布拉幹薩 (AWS)

環境：生產	技術：無伺服器；雲端原生；軟體開發與測試；現代化 DevOps；基礎架構	AWS 服務：Amazon S3；Amazon SNS；Amazon SQS；AWS Lambda

Summary

此模式展示了一種可擴展的無伺服器方法，方法是將亞馬遜簡單儲存服務 (Amazon S3)、亞馬遜簡單通知服務 (Amazon SNS)、亞馬遜簡單 Amazon Simple Queue Service (Amazon SQS) 和 AWS Lambda 鏈接在一起，以處理上傳檔案。上傳的文件示例用於演示目的。您可以使用無伺服器方法將符合業務目標所需的 AWS 服務組合鏈結在一起，以完成其他任務。無伺服器方法採用非同步工作流程，該工作流程仰賴事件驅動的通知、彈性儲存裝置，以及運算即服務 (FAA) 來處理要求。您可以使用無伺服器方法擴充以滿足需求，同時將成本降至最低。

注意：透過無伺服器方法將 AWS 服務鏈結在一起，有幾個選項。例如，您可以使用將 Lambda 與 Amazon S3 相結合的方法，而不是使用 Amazon SNS 和 Amazon SQS。但是，此模式使用 Amazon SNS 和 Amazon SQS，因為這種方法可讓您在事件通知期間將多個整合點新增至 Lambda 叫用程序，並擴展實作以在無伺服器協調中包含多個偵聽程式，同時將處理額外負荷降至最低。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 以程式設計方式存取 AWS 帳戶。如需詳細資訊，請參閱：
  - AWS Cloud Development Kit (AWS CDK) 文件中的先決條件
  - AWS Command Line Interface (AWS CLI) (AWS CLI) 文件中的先決條件
- AWS CDK, 已安裝
- AWS CLI，已安裝和設定
- Python 3.9

產品版本

- AWS CDK 2.x
- Python 3.9

架構

下圖說明鏈結的 AWS 服務如何讓使用者將檔案上傳到 S3 儲存貯體以進行處理：

該圖顯示以下工作流程：

1. 使用者將檔案上傳到 S3 儲存貯體。
2. 上傳會啟動 S3 事件，將訊息發佈到 SNS 主題。訊息包含 S3 事件的詳細資訊。
3. 發佈至 SNS 主題的訊息會插入 SQS 佇列，該佇列已訂閱並接收該主題的通知。
4. Lambda 函數會輪詢 SQS 佇列 (做為其事件來源)，並等待訊息處理。
5. Lambda 函數從 SQS 佇列接收訊息時，會處理這些訊息並確認收到這些訊息。
6. 如果 Lambda 未處理訊息，則該訊息會傳回至 SQS 佇列，最後會傳輸至 SQS 無效字母佇列。

技術, 堆

- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Lambda

工具

AWS 服務

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- Amazon Simple Queue Service (Amazon SQS) 提供安全、耐用且可用的託管佇列，可協助您整合和分離分散式軟體系統和元件。

- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。

其他工具

- AWS Cloud Development Kit (AWS CDK) 是與 AWS CDK 應用程式互動的主要工具。它會執行您的應用程式、詢問您定義的應用程式模型，以及產生和部署 AWS CDK 產生的 AWS CloudFormation 範本。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Python 是一種高級別的，解釋的通用編程語言。

Code

此模式的程式碼可在將 S3 與 SNS GitHub 鏈結到 SQS 到 Lambda 儲存庫中使用。

史诗

開發您的無伺服器環境

任務	描述	所需技能
複製儲存庫。	複製存放庫並導覽至 `python/s3-sns-sqs-lambda-chain` 資料夾。	應用程式開發人員
設定虛擬環境。	1. 在 AWS CDK 中，執行 `python3 -m venv .venv` 命令。 2. 在 MacOS/Linux 或 `.venv\Scripts\activate.bat` 視窗上執行 `source .venv/bin/activate` 指令。	應用程式開發人員
安裝依存項目。	執行 `pip install -r requirements.txt` 命令。	應用程式開發人員

測試 CloudFormation 堆疊

任務	描述	所需技能
運行單元測試。	1. 執行 `pip install -r requirements-dev.txt` 命令。 2. (選擇性) 執行`cdk synth --no-staging > template.yml` 命令以產生 CloudFormation 堆疊。重要事項：您可以檢查堆疊，但避免產生暫存的資源和成品。 3. 運行命`pytest`令以運行所有單元測試。 4. (選擇性) 執行`pytest tests/unit/<test_filename>` 命令以針對特定檔案執行測試。	應用開發人員，測試工程

部署 CloudFormation 堆疊

任務	描述	所需技能
設置啟動程序環境。	遵循 AWS 文件中啟動載入中的指示，在將部署 CloudFormation 堆疊的每個 AWS 區域中啟動 AWS CDK 部署環境。 附註：此步驟要求您擁有具有程式設計存取權的認證。	App 開發人員、 DevOps 工程師、資料工程師
部署 CloudFormation 堆疊。	執行命`cdk deploy`令以建置堆疊並將其部署到 AWS 帳戶。	AWS 應用程式開發人員、DevOps 工程師 DevOps

清理您環境的資源

任務	描述	所需技能
刪除 CloudFormation 堆疊並移除關聯的資源。	若要刪除已建立的 CloudFormation 堆疊並移除所有關聯的資源，請執行 `run cdk destroy` 命令。	應用程式開發人員

更多模式

- 使用 Athena 存取、查詢和加入 Amazon DynamoDB 資料表
- 彙總 Amazon DynamoDB 中的資料，用於 Athena 的機器學習預測
- 自動化 AWS 資源評估
- 使用 AWS SAM 自動部署巢狀應用程式
- 自動化跨 AWS 帳戶複寫 Amazon RDS 執行個體
- 使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3
- 自動檢測更改並為中的壟斷啟動不同的 CodePipeline 管道 CodeCommit
- 使用 DevOps 實務和 AWS Cloud9 建立鬆散結合的架構與微型服務
- 在 Amazon 服務中建立多租戶無伺服器架構 OpenSearch
- 在 AWS 雲端建置進階大型主機檔案檢視器
- 使用 AWS 服務計算風險值 (VaR)
- 跨不同 AWS 帳戶和 AWS 區域複製 AWS Service Catalog 產品
- 自動為 Java 和 Python 項目創建動態 CI 管道
- 通過使用 CQRS 和事件採購將巨石分解為微服務
- 將反應型單頁應用程式部署到 Amazon S3 和 CloudFront
- 使用私有端點和應用 Application Load Balancer 在內部網站上部署 Amazon API Gateway API
- 部署和偵錯 Amazon EKS 叢集
- 使用基礎設施即程式碼在 AWS 雲端部署和管理無伺服器資料湖
- 使用容器映像部署 Lambda 函數
- 使用 Amazon 基岩代理程式和知識庫，開發以聊天為基礎的全自動化助理
- 使用 RAG 和提示，開發先進的生成式 AI 聊天助理 ReAct
- 使用 Step Functions 數使用 IAM 存取分析器動態產生 IAM 政策
- 確保啟動時已啟用 Amazon S3 的亞馬遜 EMR 記錄功能
- 估算隨需容量的 DynamoDB 表格的成本
- 使用 Amazon Personalize 個人化產生個人化和重新排名的建議
- 使用 AWS AWS Glue 任務和 Python 產生測試資料
- 使用 AWS Step Functions 實作無伺服器傳奇模式
- 透過 AWS CDK 啟用跨多個 AWS 區域、帳戶和作業單位的 Amazon DevOps Guru，提升營運效能
- 使用 Step Functions 函數和 Lambda 代理函數在 AWS 帳戶之間啟動 CodeBuild 專案

- 使用 AWS Glue 將阿帕奇卡桑德拉工作負載遷移到 Amazon Keyspaces
- 監控跨多個 AWS 帳戶共用 Amazon 機器映像的使用
- 使用 AWS Step Functions 透過驗證、轉換和分割協調 ETL 管道
- 使用 AWS Fargate 大規模執行事件驅動和排程的工作負載
- 使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront
- 使用 AWS Lambda 在六角形架構中建構 Python 專案
- 在多帳戶環境中，關閉所有 Security Hub 成員帳戶的安全性標準控制

軟體開發與測試

主題

- 使用 Python 應用程式為亞馬遜動態 B 自動產生模型和 CRUD 函數
- 透過 Green Boost 探索全堆疊雲端原生 Web 應用程式開發
- 使用 AWS 針對 Node.js 應用程式執行單元測試 GitHub CodeBuild
- 使用 AWS Lambda 在六角形架構中建構 Python 專案
- 更多模式

使用 Python 應用程式為亞馬遜動態 B 自動產生模型和 CRUD 函數

創建者：維吉特瓦希沙 (AWS)、阿林查達尼 (AWS) 和丹南傑卡 (AWS)

代碼存儲庫：amazon-reverse-engineer-dynamodb	環境：PoC 或試點	技術：軟件開發和測試; 數據庫; DevOps
工作負載：開源	AWS 服務：Amazon DynamoDB	

Summary

通常需要實體以及建立、讀取、更新和刪除 (CRUD) 操作功能，才能有效率地執行 Amazon DynamoDB 資料庫操作。PynamoDB 是一個基於 Python 的接口，支持 Python 3。它也提供諸如 Amazon DynamoDB 交易的支援、自動屬性值序列化和還原序列化，以及與常見 Python 架構 (例如 Flask 和 Django) 的相容性等功能。此模式提供可簡化 PynamoDB 模型和 CRUD 作業函數的自動建立的程式庫，協助開發人員使用 Python 和 DynamoDB。雖然它會為資料庫表產生基本的 CRUD 函數，但它也可以從 Amazon DynamoDB 表反向工程 PynamoDB 模型和 CRUD 函數。此模式旨在通過使用基於 Python 的應用程序簡化數據庫操作。

以下是此解決方案的主要功能：

- 將 JSON 結構描述轉 PynamoDB 為模型 — 透過匯入 JSON 結構描述檔案自動產 PynamoDB 中的模型。
- CRUD 函數產生 — 自動產生函數，以便在 DynamoDB 表上執行 CRUD 作業。
- 從 DynamoDB 進行逆向工程 — 使用 PynamoDB 物件關聯式對應 (ORM) 對現有 Amazon DynamoDB 資料表進行反向工程。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Python 版本 3.8 或更高版本，下載並安裝
- Jinja2 版本 3.1.2 或更新版本，已下載並安裝

- 您想要為其產生 ORM 的 Amazon DynamoDB 表
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝版本 5.4.1 或更新版本

架構

目標技術堆疊

- 腳本
- Python 應用
- PyNAMODB 模型
- Amazon DynamoDB 資料庫執行個體

目標架構

1. 您可以建立輸入 JSON 結構定義檔案。這個 JSON 結構定義檔案代表您要從中建立模型和 CRUD 函數之個別 DynamoDB 資料表的屬性。它包含以下三個重要鍵：
   - `name`目標 DynamoDB 資料表的名稱。
   - `region`— 代管表格的 AWS 區域
   - `attributes`— 屬於目標資料表一部分的屬性，例如分割索引鍵 (也稱為雜湊屬性)、排序索引鍵、本機次要索引、全域次要索引，以及任何非索引鍵屬性。此工具預期輸入結構描述只會在應用程式直接從目標資料表擷取索引鍵屬性時提供非索引鍵屬性。如需如何在 JSON 結構定義檔案中指定屬性的範例，請參閱此模式的其他資訊一節。
2. 運行 Python 應用程序並提供 JSON 模式文件作為輸入。
3. Python 應用程序讀取 JSON 模式文件。
4. Python 應用程式會連接至動 DynamoDB 料表，以衍生結構描述和資料類型。應用程式會執行 describe_table 作業，並擷取表格的索引鍵和索引屬性。
5. Python 應用程式會結合 JSON 結構定義檔案和 DynamoDB 資料表中的屬性。它使用神社模板引擎來 PynamoDB 一個模型和相應的 CRUD 函數。
6. 您可以存 PynamoDB，以便在 DynamoDB 資料表上執行 CRUD 作業。

工具

AWS 服務

- Amazon DynamoDB 是一項全受管 NoSQL 資料庫服務，可提供快速、可預期且可擴展的效能。

其他工具

- Jinja 是一個可擴展的模板引擎，可將模板編譯為優化的 Python 代碼。這種模式使用 Jinja 通過在模板中嵌入佔位符和邏輯來生成動態內容。
- PynamoDB 是 Amazon DynamoDB 的基於蟒蛇的界面。
- Python 是一種通用的計算機編程語言。

代碼存儲庫

此模式的代碼可以在 GitHub 自動生成 PynamoDB 模型和 CRUD 函數存儲庫中找到。該存儲庫分為兩個主要部分：控制器包和模板。

控制器套件

控制器 Python 包包含有助於生成 PynamoDB 模型和 CRUD 功能的主要應用程序邏輯。其中包含下列各項：

- `input_json_validator.py`— 此 Python 指令碼會驗證輸入的 JSON 結構定義檔案，並建立包含目標 DynamoDB 表格清單的 Python 物件，以及每個表格的必要屬性。
- `dynamo_connection.py`— 此指令碼會建立與 DynamoDB 資料表的連線，並使用該`describe_table`作業擷取建立 PynamoDB 模型所需的屬性。
- `generate_model.py`— 此指令碼包含一個 Python 類別，`GenerateModel`該類別會根據輸入 JSON 結構描述檔案和作業建立 PynamoDB 模型。`describe_table`
- `generate_crud.py`— 針對 JSON 結構定義檔案中定義的 DynamoDB 資料表，此指令碼會使用此`GenerateCrud`作業來建立 Python 類別。

範本

這個 Python 目錄包含以下神社模板：

- `model.jinja`— 此 Jinja 範本包含用於產生 PynamoDB 模型指令碼的範本運算式。
- `crud.jinja`— 此 Jinja 範本包含用於產生 CRUD 函數指令碼的範本運算式。

史诗

設定環境

任務	描述	所需技能
複製儲存庫。	輸入以下命令以克隆自動生成 PynamoDB 模型和 CRUD 函數存儲庫。 `git clone https://github.com/aws-samples/amazon-reverse-engineer-dynamodb.git`	應用程式開發人員
設置 Python 環境。	1. 導覽至複製儲存庫中的頂層目錄。 `cd amazon-reverse-engineer-dynamodb` 2. 輸入下列命令以安裝所需的程式庫和套件。 `pip install -r requirements.txt`	應用程式開發人員

PynamoDB 模型和 CRUD 函數

任務	描述	所需技能
修改 JSON 結構定義檔案。	1. 導覽至複製儲存庫中的頂層目錄。 `cd amazon-reverse-engineer-dynamodb`	應用程式開發人員

任務	描述	所需技能
	2. 在偏好的編輯器中開啟`test.json` 檔案。您可以使用此檔案做為建立自己的 JSON 結構定義檔案的參考，或者更新此檔案中的值以符合您的環境。 3. 修改目標 DynamoDB 表格的名稱 AWS 區域、和屬性值。 附註：如果您定義的資料表不存在於 JSON 結構定義檔案中，則此解決方案不會為該資料表產生模型或 CRUD 函數。 4. 儲存並關閉 `test.json` 檔案。建議您使用新名稱儲存此檔案。	
運行 Python 應用程序。	輸入以下命令以生成 PynamoDB 模型和 CRUD 函數，其中`<input_schema.json>` 是 JSON 結構定義文件的名稱。 `python main.py --file <input_schema.json>`	應用程式開發人員

驗證模型和 CRUD 功能

任務	描述	所需技能
驗證 PynamoDB 型。	1. 在複製存放庫的頂層目錄中，輸入下列指令以導覽至`models`存放庫。 `cd models` 2. 依預設，此解決方案會命名 PynamoDB 模型檔案。`demo_model.py` 驗證此檔案是否存在。	應用程式開發人員
驗證生成的 CRUD 函數。	1. 在複製存放庫的頂層目錄中，輸入下列指令以導覽至`crud`存放庫。 `cd crud` 2. 依預設，此解決方案會命名指令碼`demo_crud.py` 。驗證此檔案是否存在。 3. 使用`demo_crud.py` 檔案中的 Python 類別，在目標 DynamoDB 資料表上執行 CRUD 作業。確認作業已順利完成。	應用程式開發人員

相關資源

- Amazon DynamoDB 件 (文件)
- 使用次要索引改善資料存取 (DynamoDB 文件)

其他資訊

JSON 結構定義檔案的範例屬性

```
[
{
"name": "test_table",
"region": "ap-south-1",
"attributes": [
{
"name": "id",
"type": "UnicodeAttribute"
},
{
"name": "name",
"type": "UnicodeAttribute"
},
{
"name": "age",
"type": "NumberAttribute"
}
]
}
]
```

透過 Green Boost 探索全堆疊雲端原生 Web 應用程式開發

由本·斯蒂克利（AWS）和阿米薩馬塔爾（AWS）創建

環境：PoC 或試點	技術：軟體開發與測試、Web 和行動應用程式、雲端原生	工作負載：開源
AWS 服務：Amazon Aurora；AWS CDK；Amazon CloudFront；AWS Lambda；AWS WAF		

Summary

為了因應開發人員不斷變化的需求，Amazon Web Services (AWS) 認識到開發雲端原生 Web 應用程式的有效方法的關鍵需求。AWS 的重點是協助您克服與在 AWS 雲端部署 Web 應用程式相關的常見障礙。透過利用 AWS Cloud Development Kit (AWS CDK) TypeScript、React 和 Node.js 等現代技術的功能，此模式旨在簡化和加速開發程序。

該模式以 Green Boost (GB) 工具組為基礎，提供建構完全使用 AWS 廣泛功能的 Web 應用程式的實用指南。它充當全面的藍圖，可引導您完成部署與 Amazon Aurora PostgreSQL 相容版本整合的基本 CRUD (建立、讀取、更新、刪除) Web 應用程式的程序。這是透過使用綠色 Boost 命令列介面 (綠色 Boost CLI) 並建立本機開發環境來完成的。

在成功部署應用程序之後，該模式深入研究了 Web 應用程序的關鍵組件，包括基礎設施設計，後端和前端開發以及用於可視化的 CDK-dia 等基本工具，從而促進了高效的項目管理。

先決條件和限制

先決條件

- 安裝的 Git
- 視覺工作室代碼（VS 代碼）安裝
- 已安裝 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 已安裝 AWS CDK 工具組

- 已安裝 Node.js 18，或 Node.js 18 (已啟動下午)
- 安裝了 pnpm，如果它不是 Node.js 安裝的一部分
- 基本熟悉 TypeScript，AWS CDK，Node.js 和反應
- 有效的 AWS 帳戶
- 透過在中使用 AWS CDK 啟動載入的 AWS 帳戶。`us-east-1`需要 `us-east-1` AWS 區域才能支援 Amazon L CloudFront ambda @Edge 功能。
- AWS 安全登入資料`AWS_ACCESS_KEY_ID`，包括在終端機環境中正確設定
- 對於 Windows 用戶，在管理員模式下的終端機（以適應 pnpm 處理節點模塊的方式）

產品版本

- 第 3 JavaScript 版適用的 AWS 開發套件
- AWS CDK 版本 2
- AWS CLI 版本 2.2
- Node.js 版本
- 反應版本 18

架構

目標技術堆疊

- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon CloudFront
- Amazon CloudWatch
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Lambda
- AWS Secrets Manager
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)
- AWS WAF

目標架構

下圖顯示使用者請求在與 S3 儲存貯體 CloudFront、Aurora 資料庫、EC2 執行個體互動之前，先經過 Amazon、AWS WAF 和 AWS Lambda，並最終觸及開發人員。另一方面，管理員使用 Amazon SNS 和 Amazon CloudWatch 進行通知和監控。

為了獲得更深入的了解部署後的應用程序，您可以通過使用 CDK-DIA 創建圖，如下面的例子所示。

這些圖表從兩個不同的角度展示了 Web 應用程序架構。光碟直徑圖提供 AWS CDK 基礎設施的詳細技術檢視，並強調特定 AWS 服務，例如亞 Amazon Aurora PostgreSQL 相容和 AWS Lambda。相比之下，另一個圖表需要更廣泛的視角，強調數據和用戶交互的邏輯流。主要區別在於詳細程度：cdk-dia 深入了解技術複雜性，而第一張圖則提供了更加以用戶為中心的視圖。

CDK 圖的建立涵蓋史詩使用 AWS CDK 了解應用程式基礎設施。

工具

AWS 服務

- Amazon Aurora PostgreSQL 相容版本是全受管、符合 ACID 標準的關聯式資料庫引擎，可協助您設定、操作和擴展 PostgreSQL 部署。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- Amazon CloudFront 透過全球資料中心網路提供您的 Web 內容，加快 Web 內容的分發速度，進而降低延遲並提升效能。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。
- AWS Lambda 是一種運算服務，可協助您執行程式碼，而不需要佈建或管理伺服器。它只會在需要時執行程式碼並自動調整規模，因此您只需為使用的運算時間付費。
- AWS Secrets Manager 可協助您透過 API 呼叫秘密管 Secrets Manager 員來取代程式碼中的硬式編碼登入資料 (包括密碼)，以程式設計方式擷取密碼。
- AWS Systems Manager 可協助您管理在 AWS 雲端中執行的應用程式和基礎設施。它可簡化應用程式和資源管理、縮短偵測和解決操作問題的時間，並協助您安全地大規模管理 AWS 資源。此模式使用 AWS Systems Manager 工作階段管理器。

- Amazon Simple Storage Service (Amazon S3) 是雲端物件儲存服務，可協助您存放、保護和擷取任意數量的資料。 Amazon Simple Notification Service (Amazon SNS) 可協助您協調和管理發佈者和客戶之間的訊息交換，包括 Web 伺服器和電子郵件地址。
- AWS WAF 是一種 Web 應用程式防火牆，可協助您監控轉寄至受保護 Web 應用程式資源的 HTTP 和 HTTPS 請求

其他工具

- Git 是一個開放原始碼的分散式版本控制系統。
- 綠色升壓是在 AWS 上建置 Web 應用程式的工具組。
- Next.js 是用於添加功能和優化的反應框架。
- Node.js 是一個事件驅動的 JavaScript 運行時環境，旨在構建可擴展的網絡應用程序。
- pgAdmin 是一個開放原始碼的管理工具。它提供了一個圖形界面，可幫助您創建，維護和使用數據庫對象。
- pnpm 是一個用於 Node.js 項目依賴關係的軟件包管理器。

最佳實務

如需下列建議的詳細資訊，請參閱 Epics 一節：

- 使用 Amazon CloudWatch 儀表板和警示來監控基礎設施。
- 使用 cdk-nag 執行靜態基礎設施即程式碼 (IaC) 分析，以強制執行 AWS 最佳實務。
- 使用系統管理員工作階段管理員，透過 SSH (安全殼層) 通道建立資料庫連接埠轉送，這比公開公開的 IP 位址更安全。
- 透過執行來管理弱點`pnpm audit`。
- 使用 eSlint 執行靜態 TypeScript 程式碼分析，並使用 Pettier 來標準化程式碼格式，強制執行最佳作法。

史诗

部署具有 Aurora 與 PostgreSQL 相容的 CRUD 網路應用程式

任務	描述	所需技能
安裝綠色增強 CLI。	若要安裝綠色升壓 CLI，請執行下列命令。 `pnpm add -g gboost`	應用程式開發人員
建立 GB 應用程式。	1. 若要使用綠色 Boost 建立應用程式，請執行命令`gboost create`。 2. 選擇`CRUD App with Aurora PostgreSQL` 範本。	應用程式開發人員
安裝依賴關係並部署應用程序。	1. 導航到項目目錄：`cd <your directory>` 。 2. 若要安裝相依性，請執行指令`pnpm i`。 3. 導航到基礎目錄：`cd infra`。 4. 若要在本機部署應用程式，請執行命令`pnpm deploy:local` 。 這是中定義之指`cdk deploy ...`令的別名`infra/package.json` 。 等待部署完成 (約 20 分鐘)。等待時，請在主控台中監 CloudFormation 控 AWS	應用程式開發人員

任務	描述	所需技能
	CloudFormation 堆疊。請注意程式碼中定義的建構如何對應至部署的資源。檢閱主控台中的 CDK 建構樹狀檢 CloudFormation 視。	

任務	描述	所需技能
存取應用程式。	在本地部署 GB 應用程序後，您可以使用 CloudFront URL 訪問它。該 URL 被打印在終端輸出中，但它可能有點不知所措。要更快地找到它，請使用以下步驟： 1. 開啟執行`pnpm deploy:local` 指令的終端機。 2. 在終端機輸出中尋找類似下列文字的區段。 ``` myapp5stickbui9C39A55A.CloudFrontDomainName = d1q16n5pof924c.cloudfront.net ``` URL 對您的部署而言是唯一的。 或者，您可以通過訪問 Amazon CloudFront 控制台找到 CloudFront URL： 1. 登入 AWS 管理主控台並導覽至 CloudFront 服務。 2. 在清單中尋找最新部署的發行版。 複製與發行版相關聯的網域名稱。它看起來類似於`your-unique-id.cloudfront.net` 。	應用程式開發人員

使用 Amazon 監控 CloudWatch

任務	描述	所需技能
檢視 CloudWatch 儀表板。	1. 開啟 CloudWatch 主控台並選擇 [儀表板]。 2. 選擇具有名稱的儀表板-儀表板<appId><stageName>。 3. 檢閱儀表板。正在監控哪些資源？ 正在記錄哪些指標？ 該儀表板由開源構造成可能cdk-monitoring-constructs。	應用程式開發人員
啟用警示。	CloudWatch 儀表板可幫助您主動監控您的 Web 應用程序。要被動監視您的 Web 應用程序，您可以啟用警報。 1. 瀏覽至`/infra/src/app/stateless/monitor-stack.ts` ，定義監視器堆疊。 2. 取消註釋以下行，並替換`admin@example.com` 為您的電子郵件地址。 `onAlarmTopic.addSubscription(new EmailSubscription("admin@example.com"));` 3. 將下列匯入資訊新增至檔案頂端。	應用程式開發人員

任務	描述	所需技能
	`` import { EmailSubscription } from "aws-cdk-lib/aws-sns-subscriptions"; ``` 4. 在中infra/，執行下列命令。 ``` cdk deploy "*/monitor" --exclusively. ``` 5. 若要確認您對於啟動監控警示時所產生的 SNS 主題的訂閱，請選擇電子郵件訊息中的連結。	

使用 AWS CDK 了解應用程式基礎設施

任務	描述	所需技能
建立架構圖。	使用 cdk- dia 生成 Web 應用程序的體系結構圖。視覺化架構有助於改善團隊成員之間的理解和溝通。它提供了系統組件及其關係的清晰概述。 1. 安裝圖形。 2. 在中infra/，執行命令pnpm cdk-dia。 3. 檢視您的infra/diagram.png 。	應用程式開發人員
使用 cdk-nag 來執行最佳實踐。	使用 cdk-nag 強制執行最佳實務，降低安全漏洞和錯誤配置	應用程式開發人員

任務	描述	所需技能
	的風險，幫助您維護安全和合規的基礎架構。 1. 透過其規則部分探索 cdk-nag 的最佳實務執行，包括來自 AWS 解決方案庫規則套件的檢查。 2. 要了解 cdk-nag 如何執行規則，請在代碼中進行更改。例如，在中`infra/src/app/stateful/data-stacks.ts` ，變更`storageEncrypted: true`為`storageEncrypted: false` 。 3. 在中`infra/`，執行命令`cdk synth "*/data"`。在合成期間，您會遇到建置錯誤，指出規則違規。 `AwsSolutions-RDS2: The RDS instance or Aurora DB cluster does not have storage encryption enabled.` 此錯誤展示了 cdk-nag 如何成為強制執行基礎結構最佳實踐和防止安全錯誤配置的安全機制。 4. 如果需要，您也可以隱藏不同範圍的規則。例如，若要隱藏 AwsSolutions-RDS2，	

任務	描述	所需技能
	請在的實例化下方加入下列程式碼。DbIamCluster `NagSuppressions.addResourceSuppressions(` `  cluster.node.findChild("Resource"),` `  [` `  {` `  id: "AwsSolutions-RDS2",` `  reason: "Customer requirement necessitates having unencrypted DB storage",` `  },` `  ],` `);` 5. 抑制後，cdk synth "*/data"再次運行。您的 AWS CDK 應用程式現在應該可以成功合成。您可以在中找到所有隱抑的規則infra/cdk.out/assembly-<appId>-<stageName>/AwsSolutions-<appId>-<stageName>-${stackId}-NagReport.csv 。	

評估資料庫組態和結構描述

任務	描述	所需技能
獲取環境變量。	若要取得必要的環境變數，請使用下列步驟： 1. 若要尋找DB_BASTION_ID ，請登入主控台，然後導覽至 EC2 主控台。選擇「執行處理」(執行中)，然後尋找包含的資料列 -「ssm-db-bastion 名稱<stageName>」。執行個體 ID 以 i- 開頭。 2. 若要尋找DB_ENDPOINT ，請在 Amazon Relational Database Service (Amazon RDS) 主控台上選擇資料庫執行個體，然後選取具有資料庫識別碼以 --data-<appId><stageName>開頭的區域叢集。找到以 rds.amazonaws.com 結尾的寫入器執行個體端點。	應用程式開發人員
建立連接埠轉送。	若要建立連接埠轉送，請使用下列步驟： 1. 安裝 AWS Systems Manager 會話管理器外掛程式。 2. 透過pnpm db:connect 在中執行來啟動連接埠轉送，core/以透過防禦主機建立安全連線。	應用程式開發人員

任務	描述	所需技能
	3. 在您的終端機`Waiting for connections...`，中看到文字後，您的本機電腦和 Aurora 伺服器透過 EC2 堡壘主機成功建立 SSH 通道。	
調整 Systems Manager 工作階段管理員逾時。	(選擇性) 如果預設的 20 分鐘工作階段逾時太短，您可以選擇工作階段管理員、偏好設定、編輯、閒置工作階段逾時，在 Systems Manager 主控台中將其增加到 60 分鐘。	應用程式開發人員

任務	描述	所需技能
可視化數據庫。	pgAdmin 是一個用戶友好的開源工具，用於管理 PostgreSQL 數據庫。它簡化了數據庫任務，使您能夠有效地創建，管理和優化數據庫。本節將引導您如何安裝 PGAdmin，並將其功能用於 PostgreSQL 資料庫管理。 1. 在 [物件總管] 中，開啟 [伺服器] 的內容 (按一下滑鼠右鍵) 選單，然後選擇 [註冊]、[伺服器] 2. 在「一般」頁籤上，在「<appId><stageName>名稱」欄位中輸入 -。 3. 若要擷取資料庫密碼，請開啟 AWS Secrets Manager 主控台，選取含有為堆疊的 CDK 產生描述的密碼：--data <appId><stageName>，然後選擇秘密值卡。選擇擷取秘密值，然後使用密碼金鑰複製密碼值。 4. 在 [連線] 索引標籤上，在 [主機名稱/位址] 欄位中輸入 0.0.0，然後在 [使用者名稱] 欄位中輸入 _admin。<appId> 在「密碼」欄位中，使用您先前擷取的密碼。選擇是作為保存密碼？欄位。 5. 選擇儲存。	應用程式開發人員

任務	描述	所需技能
	6. 要查看表，導航到 - <appId><stageName>，數據庫，_DB<appId>，模式<appId>，表。 7. 開啟項目表格的上下文 (按一下右鍵) 功能表，然後選取「檢視/編輯資料」、「所有列」。 8. 探索表格。	

使用 Node.js 進行除錯

任務	描述	所需技能
偵錯建立項目使用案例。	若要偵錯建立項目使用案例，請依照下列步驟執行： 1. 打開文core/src/modules/item/create-item.use-case.ts 件，並插入以下代碼。 `import { fileURLTo Path } from "node:url ";` `// existing create-it em.use-case.ts code here` `if (process.argv[1] === fileURLTo Path(import.meta.u rl)) { createItemUseCase( {`	應用程式開發人員

任務	描述	所需技能
	`` description: "Item 1's Descripti on", name: "Item 1", }); } ``` 2. 在上一個步驟中加入的程式碼可確保在直接執行此模組時呼叫`createIteмUseCase` 函式。在此代碼塊中要啟動 line-by-line 調試的行上設置斷點。 1. 開啟 VS 程式碼 JavaScript 偵錯終端機，然後執行`pnpm tsx core/src/modules/item/create-item.use-case.ts` 以 line-by-line 偵錯來執行程式碼。或者，您可以使用`console.log` 語句，但是當您使用複雜的業務邏輯時，print 語句可能不足。Line-by-line 調試為您提供了更多的上下文。	

開發前端

任務	描述	所需技能
設定開發伺服器。	1. 瀏覽並`ui/`執行`pnpm dev`以啟動 Next.js 開發伺服器。	應用程式開發人員

任務	描述	所需技能
	2. 在本機存取您的 Web 應用程式 `http://localhost:3000` 。Next.js 開發伺服器已設定為「快速重新整理」即時回饋，提供您對 React 元件所做的編輯。 3. 嘗試自定義應用程序欄顏色。開啟`ui/src/components/theme/theme.tsx` 檔案並找出定義應用程式列佈景主題的區段。在此`colorSchemes.light.palette.primary` 區段中，將主要值從更新`colors.lagoon` 為`colors.carrot` 。進行此更改後，保存文件並在瀏覽器中觀察更新。 4. 通過修改文本，組件和添加新頁面進行實驗。	

具有綠色提升的工具

任務	描述	所需技能
設置軟件包管理器和 PNPM 軟件包管理器。	1. 檢閱 `pnpm-workspace.yaml` GB 儲存庫的根目錄，並注意工作區的定義方式。如需有關工作區的詳細資訊，請參閱 pnpm 文件。	應用程式開發人員

任務	描述	所需技能
	2. 檢閱ui/package.json 並注意它如何使用套件名稱參照中core/的工作區 "<appId>/core": "workspace:^", 。 3. 觀察如何以 TypeScript 及 eSlint 配置集中在其中定義的實用程序包中。packages/ 然後，應用程序包（例如core/，infra/和）使用此配置ui/。當您的應用程序擴展並定義更多應用程序包時，這很有用，這可以在不複製配置代碼的情況下引用實用程序包。	

任務	描述	所需技能
執行 pnpm 指令碼。	在儲存庫的根目錄中執行下列命令： 1. 執行 `pnpm lint`。此命令使用 eSlint 運行靜態代碼分析。 2. 執行 `pnpm typecheck` 。此命令運行TypeScript 編譯器來檢查代碼的類型。 3. 執行 `pnpm test`。此命令運行 Vitest 以運行單元測試。 請注意如何在所有工作區中執行這些指令。這些指令在每個工作區的`package.json#scripts` 欄位中定義。	應用程式開發人員

任務	描述	所需技能
使用 eSlint 進行靜態代碼分析。	若要測試 eSlint 的靜態程式碼分析功能，請執行下列動作： 1. 首先，確保安裝了 VS 代碼 eLint 擴展名（ID：dbaeumer.vscode-eslint ）。我們還建議您安裝 VS 代碼錯誤鏡頭（ID：usernamehw.errorlens ）以查看內聯錯誤。 2. 在您的程式碼中，有意地包含一行使用該eval()函式的程式碼，如下列範例所示。 `const userInput = "import("fs").then((fs) => console.log(fs.readFileSync("/etc/passwd", { encoding: "utf8" })))"; eval(userInput);` 重要提示：這僅用於測試目的。使用eval()被認為是潛在的危險，應避免因為安全風險。 3. 包含該eval()行之後，請開啟程式碼編輯器，以確認 eSlint 使用紅色波浪線指示程式碼的氣味。 4. 檢閱 eSlint 外掛程式和設定，位於。packages/	應用程式開發人員

任務	描述	所需技能
	`eslint-config-{node,next}/.eslintrc.cjs`	
管理相依性和漏洞。	1. 若要識別任何常見弱點和暴露 (CVE)，請在存放庫的根目錄`pnpm audit`中執行。 您應該看到「未發現任何已知漏洞」。 2. `core/`通過運行在其中安裝故意易受攻擊的軟件包`pnpm add minimist@0.2.3` ，然後運行`pnpm audit`。請注意所報告的弱點。 3. 執行中解除安裝有弱點的套件`pnpm remove minimist`。`core/`	應用程式開發人員

任務	描述	所需技能
與赫斯基預提交鉤子。	1. 對整個存儲庫中的 TypeScript 文件進行一些小的更改。這些更改可以像添加註釋一樣基本。 2. 使用和然後進行階段`git add -A`並提交這些更改`git commit -m "test husky"`。 赫斯基預先提交鉤子觸發器（在中定義）運`.husky/pre-commit`行命令。`pnpm lint-staged` 3. 觀察 lint-staged 如何在 Git 暫存的`*/.lintstagedrc.js`檔案上執行檔案中指定的命令。 這些工具是幫助防止錯誤代碼進入您的應用程序的機制。	應用程式開發人員

拆除基礎設施

任務	描述	所需技能
從您的帳戶中移除部署。	1. 若要拆除您在第一個史詩中佈建的基礎結構，請執`pnpm destroy:local`行`infra/`。 2. 完成後`pnpm destroy:local`等待 15 分鐘，然後在 Lambda 主控台中搜尋您的應用程式識別碼，以刪除	應用程式開發人員

任務	描述	所需技能
	保留的 Lambda @Edge 函數。Lambda @Edge 函數會被複寫，因此很難刪除它們。如需有關刪除 Lambda @Edge 函數的詳細資訊，請參閱CloudFront 文件。	

故障診斷

問題	解決方案
無法建立連接埠轉送	確保您的 AWS 登入資料設定正確，並具有必要的許可。 仔細檢查防禦主機 ID (`DB_BASTION_ID`) 和資料庫端點 (`DB_ENDPOINT`) 環境變數是否設定正確。 如果仍然遇到問題，請參閱 AWS 文件以了解 SSH 連線疑難排解和工作階段管理員。
網站未載入 `localhost:3000`	確認終端機輸出指示連接埠轉送成功，包括轉送位址。 確保在本機電腦上使用連接埠 3000 沒有衝突的處理程序。 確認綠色 Boost 應用程式已正確設定，並在預期的連接埠 (3000) 上執行。 檢查您的 Web 瀏覽器是否有任何可能阻止本機連線的安全性延伸模組或設定。
本機部署期間的錯誤訊息 (`pnpm deploy:local`)	請仔細檢閱錯誤訊息，以識別問題的原因。 確認已正確設定必要的環境變數和組態檔案。

相關資源

- AWS CDK 文件
- 綠色提升文件
- Next.js 說明文件
- Node.js 說明文件
- 反應文檔
- TypeScript 文件

使用 AWS 針對 Node.js 應用程式執行單元測試 GitHub CodeBuild

由托馬斯·斯科特（AWS）和讓·巴蒂斯特·吉盧瓦（AWS）創建

代碼存儲庫：節點 JS 測試示例	環境：生產	技術：軟件開發和測試
AWS 服務：AWS CodeBuild		

Summary

此模式提供 Node.js 遊戲 API 的範例原始程式碼和關鍵單元測試元件。其中也包含使用 AWS 從 GitHub 儲存庫執行這些單元測試的說明 CodeBuild，做為持續整合和持續交付 (CI/CD) 工作流程的一部分。

單元測試是一個軟件開發過程中，應用程序的不同部分，稱為單元，被單獨和獨立地測試正確的操作。測試會驗證程式碼的品質，並確認其如預期般運作。其他開發人員也可以透過諮詢測試來輕鬆熟悉您的程式碼庫。單元測試可減少 future 的重構時間，幫助工程師更快地了解代碼庫，並對預期行為提供信心。

單元測試涉及測試個別函數，包括 AWS Lambda 函數。要創建單元測試，您需要一個測試框架和驗證測試（斷言）的方式。這種模式中的代碼示例使用 Mocha 測試框架和 C hai 斷言庫。

如需有關單元測試和測試元件範例的詳細資訊，請參閱其他資訊一節。

先決條件和限制

- 具有正確 CodeBuild 許可的有效 AWS 帳戶
- GitHub 帳戶（請參閱註冊說明）
- Git（請參閱安裝說明）
- 用於進行變更和推送程式碼的程式碼編輯器 GitHub (例如，您可以使用 AWS Cloud9)

架構

此模式會實作下列圖表所示的架構。

工具

工具

- Git – Git 是一個可用於程式碼開發的版本控制系統。
- AWS Cloud9 – AWS Cloud9 是整合式開發環境 (IDE)，提供豐富的程式碼編輯體驗，並支援多種程式設計語言和執行階段除錯器，以及內建終端機。其中包含用於編碼、建置、執行、測試、除錯以及在雲端中發行軟體的工具集合。您可以透過網頁瀏覽器存取 AWS Cloud9 IDE。
- AWS CodeBuild – AWS CodeBuild 是全受管的持續整合服務，可編譯原始程式碼、執行測試，以及產生可立即部署的軟體套件。有了 CodeBuild，您不需要佈建、管理和擴展自己的組建伺服器。 CodeBuild 持續擴展並同時處理多個構建，因此您的構建不會留在隊列中等待。您可以利用預先封裝好的組建環境立即開始使用，或是建立自訂的組建環境來使用您自己的組建工具。使用時 CodeBuild，您需要按分鐘計費使用的運算資源。

Code

此模式的原始程式碼可在 GitHub範例遊戲單元測試應用程式存放庫中取得。您可以從此示例（選項 1）創建自己的 GitHub 存儲庫，也可以直接使用該模式的示例存儲庫（選項 2）。請按照下一節中每個選項的說明進行操作。您遵循的選項將取決於您的使用案例。

史诗

選項 1-在您的個人 GitHub 存儲庫上運行單元測試 CodeBuild

任務	描述	所需技能
根據範例專案建立您自己的 GitHub 儲存庫。	1. 登入 GitHub。 2. 創建一個新的存儲庫。如需指示，請參閱GitHub 文件。 3. 複製範例儲存庫並將其推送至您帳戶中的新儲存庫。	應用程式開發人員、AWS 管理員、AWS DevOps
創建一個新 CodeBuild 項目。	1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/codesuite/codebuild/home 開啟 CodeBuild 主控台。	應用程式開發人員、AWS 管理員、AWS DevOps

任務	描述	所需技能
	2. 選擇 Create build project (建立建置專案)。 3. 在 [專案組態] 區段中，對於 [專案名稱]，鍵入 aws-tests-sample-node-js。 4. 在「來源」區段中，對於「來源提供者」，請選擇GitHub。 5. 對於存放庫，選擇我的 GitHub 帳戶中的存儲庫，然後將 URL 粘貼到新創建的 GitHub 存儲庫中。 6. 在「主要來源 webhook 事件」區段中，選取「每次程式碼變更推送至此儲存庫時重建」。 7. 對於事件類型，請選擇推送。 8. 在「環境」區段中，選擇受管理的映像、Amazon Linux 2 和最新映像。 9. 保留所有其他選項的預設設定，然後選擇 [建立組建專案]。	
啟動組建。	在 Review (檢閱) 頁面上，選擇 Start build (開始建置) 來執行建置。	應用程式開發人員、AWS 管理員、AWS DevOps

選項 2-在公共存儲庫上運行單元測試 CodeBuild

任務	描述	所需技能
建立新的 CodeBuild 組建專案。	1. 登入 AWS 管理主控台，然後在 https://console.aws.amazon.com/codesuite/codebuild/home 開啟 CodeBuild 主控台。 2. 選擇 Create build project (建立建置專案)。 3. 在 [專案組態] 區段中，對於 [專案名稱]，鍵入 aws-tests-sample-node-js。 4. 在「來源」區段中，對於「來源提供者」，請選擇GitHub。 5. 針對「儲存庫」，選擇「公用儲存庫」，然後貼上網址：https://github.com/aws-samples/ node-js-tests-sample。 6. 在「環境」區段中，選擇受管理的映像、Amazon Linux 2 和最新映像。 7. 保留所有其他選項的預設設定，然後選擇 [建立組建專案]。	應用程式開發人員、AWS 管理員、AWS DevOps
啟動組建。	在 Review (檢閱) 頁面上，選擇 Start build (開始建置) 來執行建置。	應用程式開發人員、AWS 管理員、AWS DevOps

分析單元測試

任務	描述	所需技能
檢視測試結果。	在主 CodeBuild 控台中，檢閱 CodeBuild 工作的單元測試結果。它們應與「其他資訊」區段中顯示的結果相符。 這些結果會驗證與的 GitHub 儲存庫整合 CodeBuild。	應用程式開發人員、AWS 管理員、AWS DevOps
套用網路掛接。	現在，您可以應用 webhook，以便每當您將代碼更改推送到存儲庫的主分支時，都可以自動啟動構建。如需指示，請參閱CodeBuild 文件。	應用程式開發人員、AWS 管理員、AWS DevOps

相關資源

- 示例遊戲單元測試應用程序（帶示例代碼的GitHub 存儲庫）
- AWS CodeBuild 文件
- GitHub 網絡掛鉤事CodeBuild 件（文檔）
- 創建一個新的存儲庫（GitHub 文檔）

其他資訊

單元測試結果

在 CodeBuild 控制台中，您應該在項目成功構建後看到以下測試結果。

示例單元測試組件

本節描述了單元測試中使用的四種類型的測試組件：斷言，間諜，存根和模擬。它包括每個組件的簡要說明和代碼示例。

斷言

斷言用於驗證預期的結果。這是一個重要的測試組件，因為它驗證了來自給定函數的預期響應。下列範例宣告會驗證初始化新遊戲時傳回的 ID 介於 0 到 1000 之間。

```
const { expect }  = require('chai');
const { Game } = require('../src/index');

describe('Game Function Group', () => {
 it('Check that the Game ID is between 0 and 1000', function() {
      const game = new Game();
      expect(game.id).is.above(0).but.below(1000)
 });
});
```

间谍

間諜用於觀察函數運行時發生的事情。例如，您可能想要驗證該函數是否已正確調用。下列範例顯示在 Game 類別物件上呼叫啟動和停止方法。

```
const { expect }  = require('chai');
const { spy } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () => {
   it('should verify that the correct function is called', () => {
      const spyStart = spy(Game.prototype, "start");
      const spyStop = spy(Game.prototype, "stop");

      const game = new Game();
      game.start();
      game.stop();

      expect(spyStart.called).to.be.true
      expect(spyStop.called).to.be.true
    });
});
```

小作品

存根用於覆蓋函數的默認響應。這在函數發出外部請求時特別有用，因為您希望避免從單元測試發出外部請求。（外部請求更適合集成測試，這可以對不同組件之間的請求進行物理測試。）在下列範例中，虛設常式會強制來自 getID 函數的傳回識別碼。

```
const { expect }  = require('chai');
const {.stub } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () =>  {
   it('Check that the Game ID is between 0 and 1000', function() {
      let generateIdStub = stub(Game.prototype, 'getId').returns(999999);

      const game = new Game();

      expect(game.getId).is.equal(999999);

      generateIdStub.restore();
    });
});
```

嘲笑

模擬是一種假的方法，它具有用於測試不同場景的預編程行為。模擬可以被認為是存根的擴展形式，並且可以同時執行多個任務。在下面的例子中，模擬用於驗證三種情況：

- 函數被調用
- 函數被用參數調用
- 函數返回整數 9

```
const { expect }  = require('chai');
const {.mock } = require('sinon');

const { Game } = require('../src/index');

describe('Game Function Group', () =>  {
   it('Check that the Game ID is between 0 and 1000', function() {
      let mock = mock(Game.prototype).expects('getId').withArgs().returns(9);

      const game = new Game();
      const id = get.getId();
```

```
      mock.verify();
      expect(id).is.equal(9);
    });
});
```

使用 AWS Lambda 在六角形架構中建構 Python 專案

由富爾坎奧魯克（AWS），多米尼克戈比（AWS），大流士昆斯（AWS）和米哈爾普洛斯基（AWS）創建

環境：PoC 或試點	技術：軟體開發與測試、雲端原生、容器與微服務、無伺服器、現代化	AWS 服務：Amazon DynamoDB 支援；AWS Lambda；Amazon API Gateway

Summary

此模式示範如何使用 AWS Lambda 在六角形架構中建構 Python 專案。該模式使用 AWS Cloud Development Kit (AWS CDK) 做為基礎設施即程式碼 (IaC) 工具，使用 Amazon API Gateway 做為其餘 API，而 Amazon DynamoDB 做為持續性層。六角形架構遵循領域驅動的設計原則。在六角形架構中，軟件由三個部分組成：域，端口和適配器。如需六角形架構及其優點的詳細資訊，請參閱在 AWS 上建置六角形架構指南。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- Python 的經驗
- 熟悉 AWS Lambda、AWS CDK、Amazon API Gateway 和 DynamoDB
- GitHub 帳戶（請參閱註冊說明）
- Git（請參閱安裝說明）
- 用於進行變更和推送程式碼的程式碼編輯器 GitHub (例如 AWS Cloud9、 Visual Studio 程式碼或 JetBrains PyCharm)
- Docker 安裝，並且 Docker 守護進程啟動並運行

產品版本

- Git 版本 2.24.3 或更新版本

- Python 版本 3.7 或更高版本
- AWS CDK V2
- 詩歌版本 1.1.13 或更新版本
- 適用於 Python 版本 1.25.6 或更新版本的 AWS Lambda 電源工具
- 最新版本 7.1.1 或更高版本
- 摩托車版本 3.1.9 或更高版本
- 自爆版本 1.9.0 或更高版本
- 肉毒桿菌毒素 3 版本 1.22.4 或更高版本
- 我的肉毒桿 3-動態布 1.24.0 版本或更高版本

架構

目標技術堆疊

目標技術堆疊包含使用 API Gateway、Lambda 和 DynamoDB 的 Python 服務。此服務會使用 DynamoDB 介面卡來保留資料。它提供了一個使用 Lambda 作為入口點的函數。該服務使用 Amazon API Gateway 來公開一個 REST API。該 API 使用 AWS Identity and Access Management (IAM) 進行用戶端身份驗證。

目標架構

為了說明實作，此模式會部署無伺服器目標架構。用戶端可以將要求傳送至 API Gateway 端點。API Gateway 會將要求轉送至實作六角形架構模式的目標 Lambda 函數。Lambda 函數會在 DynamoDB 資料表上執行建立、讀取、更新和刪除 (CRUD) 作業。

> 重要事項：此模式已在 PoC 環境中進行測試。在將任何架構部署到生產環境之前，您必須進行安全審查以識別威脅模型並建立安全的程式碼庫。

API 支援產品實體上的五項作業：

- `GET /products`返回所有產品。
- `POST /products`建立新產品。
- `GET /products/{id}`傳回特定產品。

- PUT /products/{id}更新特定產品。
- DELETE /products/{id}刪除特定產品。

您可以使用下列資料夾結構來組織專案，以遵循六邊形架構模式：

```
app/  # application code
|--- adapters/  # implementation of the ports defined in the domain
     |--- tests/  # adapter unit tests
|--- entrypoints/  # primary adapters, entry points
     |--- api/  # api entry point
          |--- model/  # api model
          |--- tests/  # end to end api tests
|--- domain/  # domain to implement business logic using hexagonal architecture
     |--- command_handlers/  # handlers used to execute commands on the domain
     |--- commands/  # commands on the domain
     |--- events/  # events triggered via the domain
     |--- exceptions/  # exceptions defined on the domain
     |--- model/  # domain model
     |--- ports/  # abstractions used for external communication
     |--- tests/  # domain tests
|--- libraries/  # List of 3rd party libraries used by the Lambda function
infra/  # infrastructure code
simple-crud-app.py  # AWS CDK v2 app
```

工具

AWS 服務

- Amazon API Gateway 是一項全受管服務，可讓開發人員輕鬆建立、發佈、維護、監控和保護任何規模的 API。
- Amazon DynamoDB 是全受管、無伺服器、金鑰值 NoSQL 資料庫，專為執行任何規模的高效能應用程式而設計。
- AWS Lambda 是一種無伺服器、事件驅動的運算服務，可讓您針對幾乎任何類型的應用程式或後端服務執行程式碼，而無需佈建或管理伺服器。您可以從 200 多個 AWS 服務和軟體即服務 (SaaS) 應用程式啟動 Lambda 函數，而且只需按使用量付費。

工具

- Git 被用作在這種模式的代碼開發的版本控制系統。

- Python 被用作這種模式的編程語言。Python 提供了高級數據結構和面向對象編程的方法。AWS Lambda 提供內建的 Python 執行階段，可簡化 Python 服務的操作。
- 視覺工作室代碼被用作 IDE 用於開發和測試這種模式。您可以使用任何支援 Python 開發的 IDE (例如 AWS Cloud9 或 PyCharm)。
- AWS Cloud Development Kit (AWS CDK) 是開放原始碼軟體開發架構，可讓您使用熟悉的程式設計語言來定義雲端應用程式資源。這種模式使用 CDK 來編寫和部署雲基礎架構作為代碼。
- 詩歌用於管理模式中的依賴關係。
- 碼頭工具是由 AWS CDK 用來建立 Lambda 套件和層。

Code

此模式的程式碼可在 GitHub Lambda 六角形架構範例存放庫中取得。

最佳實務

若要在生產環境中使用此模式，請遵循下列最佳作法：

- 使用 AWS Key Management Service (AWS KMS) 中的客戶受管金鑰來加密亞馬遜 CloudWatch 日誌群組和 Amazon DynamoDB 表格。
- 將適用於 Amazon API Gateway 的 AWS WAF 設定為僅允許從您組織的網路存取。
- 如果 IAM 不符合您的需求，請考慮使用其他 API Gateway 授權選項。例如，您可以使用 Amazon Cognito 用者集區或 API Gateway Lambda 授權器。
- 使 DynamoDB 備份。
- 使用虛擬私有雲 (VPC) 部署設定 Lambda 函數，以將網路流量保留在雲端內。
- 更新跨來源資源共用 (CORS) 預檢允許的原始組態，以限制只存取要求的原始網域。
- 請使用 cdk-nag 檢查 AWS CDK 程式碼，瞭解安全性最佳實務。
- 請考慮使用程式碼掃描工具，找出程式碼中常見的安全性問題。例如，強盜是一個旨在查找 Python 代碼中常見安全問題的工具。PIP 稽核會掃描 Python 環境中是否有已知弱點的套件。

此病毒碼使用 AWS X-Ray 透過應用程式的入口點、網域和轉接器追蹤請求。AWS X-Ray 可協助開發人員識別瓶頸並判斷高延遲情況，以提升應用程式效能。

史诗

初始化專案

任務	描述	所需技能
創建您自己的存儲庫。	1. 登入 GitHub。 2. 創建一個新的存儲庫。如需指示，請參閱GitHub 文件。 3. 克隆並將此模式的示例存儲庫推送到您帳戶中的新存儲庫中。	應用程式開發人員
安裝依存項目。	1. 安裝詩歌。 `pip install poetry` 2. 從根目錄安裝套件。下列命令會安裝應用程式和 AWS CDK 套件。它也會安裝執行單元測試所需的開發套件。所有已安裝的套件都會放置在新的虛擬環境中。 `poetry install` 3. 若要查看已安裝套件的圖形表示，請執行下列命令。 `poetry show --tree` 4. 更新所有相依性。 `poetry update` 5. 在新建立的虛擬環境中開啟新的 shell。它包含所有已安裝的依賴項。	應用程式開發人員

任務	描述	所需技能
	`poetry shell`	

任務	描述	所需技能
設定您的 IDE。	我們建議您使用視覺工作室程式碼，但您可以使用任何支援 Python 的 IDE。下面的步驟是視覺工作室代碼。 1. 更新.vscode/settings 檔案。 { "python.testing.pytestArgs": ["app/adapters/tests", "app/entrypoints/api/tests", "app/domain/tests"], "python.testing.unittestEnabled": false, "python.testing.pytestEnabled": true, "python.envFile": "${workspaceFolder}/.env", } 2. 在專.env案的根目錄中建立檔案。這可確保專案的根目錄包含在中，以PYTHONPATH 便pytest能夠找到它並正確地探索所有套件。 PYTHONPATH=.	應用程式開發人員

任務	描述	所需技能
運行單元測試，選項 1：使用視覺工作室代碼。	1. 選擇由詩歌管理的虛擬環境的 Python 解釋器。 2. 從測試資源管理器運行測試。	應用程式開發人員
運行單元測試，選項 2：使用 shell 命令。	1. 在虛擬環境中啟動新的 shell。 `poetry shell` 2. 從根目錄執行`pytest`命令。 `python -m pytest` 或者，您也可以直接從 Poews 執行命令。 `poetry run python -m pytest`	應用程式開發人員

部署和測試應用程式

任務	描述	所需技能
請求臨時登入資料。	若要在執行時在命令介面上擁有 AWS 登入資料`cdk deploy`，請使用 AWS IAM 身分中心 (AWS Single Sign-On 的後續任務) 建立臨時登入資料。如需指示，請參閱如何擷取短期登入資料以搭配 AWS IAM 身分中心使用 CLI 的部落格文章。	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
部署應用程式。	1. 安裝 AWS CDK 第 2 版。 `npm install -g aws-cdk` 如需詳細資訊，請參閱 AWS CDK 文件。 2. 將 AWS CDK 引導至您的帳戶和區域。 `cdk bootstrap aws://12345678900/us-east-1 --profile aws-profile-name` 3. 使用 AWS 設定檔將應用程式部署為 AWS CloudFormation 堆疊。 `cdk deploy --profile aws-profile-name`	AWS 應用程式開發人員 DevOps
測試 API，選項 1：使用控制台。	使用 API Gateway 主控台來測試 API。有關 API 操作和請求/響應消息的更多信息，請參閱存儲庫中自述文件的 API 使用部分。 GitHub	AWS 應用程式開發人員 DevOps

任務	描述	所需技能
測試 API，選項 2：使用郵遞員。	如果您想使用郵差之類的工具： 1. 將 Postman 安裝為獨立應用程式或瀏覽器擴充功能 2. 複製 API Gateway 的端點 URL。它將採用以下格式。 `https://{api-id}.execute-api.{region}.amazonaws.com/{stage}/{path}` 3. 在授權索引標籤中設定 AWS 簽名。如需指示，請參閱 AWS RE：張貼有關為 API Gateway REST API 啟用 IAM 身份驗證的文章。 4. 使用郵遞員將要求傳送至您的 API 端點。	AWS 應用程式開發人員 DevOps

開發服務

任務	描述	所需技能
為業務域編寫單元測試。	1. 使用檔案名稱前置詞在`app/domain/tests` 資料夾中建立 Python `test_` 檔案。 2. 使用下列範例建立新的測試方法來測試新的商務邏輯。 `def test_create_product_should_store_in_repository():`	應用程式開發人員

任務	描述	所需技能
	<pre> # Arrange command = create_product_com mand.CreateProduct Command(name="Test Product", descripti on="Test Descripti on",) # Act create_pr oduct_command_hand ler.handle_create_ product_command(command=c ommand, unit_of_w ork=mock_unit_of_w ork) # Assert</pre> 3. 在app/domain/commands 資料夾中建立命令類別。 4. 如果功能是新功能，請在app/domain/command_handlers 資料夾中為命令處理常式建立虛設常式。 5. 運行單元測試以查看它失敗，因為仍然沒有業務邏輯。 <pre>python -m pytest</pre>	

任務	描述	所需技能
實現命令和命令處理程序。	1. 在新建立的命令處理常式檔案中實作業務邏輯。 2. 對於與外部系統交互的每個依賴關係，請在app/domain/ports 文件夾中聲明一個抽象類。 `class ProductsRepository(ABC):` `  @abstractmethod` `  def add(self, product: product.Product) -> None:` `  ...` `class UnitOfWork(ABC):` `  products: ProductsRepository` `  @abstractmethod` `  def commit(self) -> None:` `  ...` `  @abstractmethod` `  def __enter__(self) -> typing.Any:` `  ...` `  @abstractmethod` `  def __exit__(self, *args) -> None:` `  ...` 3. 使用抽象端口類作為類型註釋，更新命令處理程序簽名以接受新聲明的依賴關係。	應用程式開發人員

任務	描述	所需技能
	<pre>def handle_cr eate_product_comma nd(command: create_pr oduct_command.Crea teProductCommand, unit_of_work: unit_of_work.UnitO fWork,) -> str: ...</pre> 4. 更新單元測試以模擬命令處理常式的所有宣告相依性的行為。 <pre> # Arrange mock_unit_of_work = unittest.mock.crea te_autospec(spec=unit _of_work.UnitOfWor k, instance=True) mock_unit _of_work.products = unittest.mock.crea te_autospec(spec=unit _of_work.ProductsR epository, instance= True)</pre> 5. 更新測試中的斷言邏輯以檢查預期的依賴關係調用。 <pre> # Assert</pre>	

任務	描述	所需技能
	<pre> mock_unit _of_work.commit.as sert_called_once() product = mock_unit_of_work. products.add.call_ args.args[0] assertpy. assert_that(produc t.name).is_equal_t o("Test Product") assertpy. assert_that(produc t.description).is_ equal_to("Test Description")</pre> 6. 運行單元測試以查看它成功。 <pre>python -m pytest</pre>	

任務	描述	所需技能
撰寫次要介面卡的整合測試。	1. 使用test_做為檔案名稱前置詞，在資app/adapters/tests 料夾中建立測試檔案。 2. 使用 Moto 程式庫來模擬 AWS 服務。 3. 為介面卡的整合測試建立新的測試方法。	應用程式開發人員

```
@pytest.fixture
def mock_dynamodb():
   with moto.mock
_dynamodb():
    yield boto3.res
ource("dynamodb",
 region_name="eu-ce
ntral-1")
```

```
def test_add_
and_commit_should_
store_product(mock
_dynamodb):
    # Arrange
    unit_of_work =
 dynamodb_unit_of_w
ork.DynamoDBUnitOf
Work(
        table_nam
e=TEST_TABLE_NAME,
 dynamodb_client=mo
ck_dynamodb.meta.c
lient
    )
    current_time =
 datetime.datetime.
now(datetime.timez
one.utc).isoformat
()
```

任務	描述	所需技能

```
    new_product_id =
 str(uuid.uuid4())
    new_product =
 product.Product(
        id=new_pr
oduct_id,
        name="test-
name",
        descripti
on="test-descripti
on",
        createDat
e=current_time,
        lastUpdat
eDate=current_time,
    )

    # Act
    with unit_of_w
ork:
        unit_of_w
ork.products.add(n
ew_product)
        unit_of_w
ork.commit()

    # Assert
```

4. 在app/adapters 資料夾中建立轉接器類別。使用 ports 資料夾中的抽象類別做為基底類別。
5. 運行單元測試以查看它失敗，因為仍然沒有邏輯。

```
python -m pytest
```

任務	描述	所需技能
實作次要介面卡。	1. 在新建立的轉接器檔案中實作邏輯。 2. 更新測試斷言。 `# Assert` `    with unit_of_work_readonly:` `        product_from_db = unit_of_work_readonly.products.get(new_product_id)` `    assertpy.assert_that(product_from_db).is_not_none()` `    assertpy.assert_that(product_from_db.dict()).is_equal_to(` `        {` `            "id": new_product_id,` `            "name": "test-name",` `            "description": "test-description",` `            "createDate": current_time,` `            "lastUpdateDate": current_time,` `        }` `    )` 3. 運行單元測試以查看它成功。	應用程式開發人員

任務	描述	所需技能
	`python -m pytest`	

任務	描述	所需技能
編寫 end-to-end 測試。	1. 使用`test_`做為檔案名稱前置詞，在資`app/entry points/api/tests` 料夾中建立測試檔案。 2. 建立 Lambda 上下文夾具，以供測試用來呼叫 Lambda。 `@pytest.fixture def lambda_co ntext(): @dataclass class LambdaCon text: function_ name: str = "test" memory_li mit_in_mb: int = 128 invoked_f unction_arn: str = "arn:aws:lambda:eu -west-1:809313241: function:test" aws_reque st_id: str = "52fdfc07-2182-154 f-163f-5f0f9a621d7 2" return LambdaCon text()` 3. 建立 API 叫用的測試方法。 `def test_crea te_product(lambda_ context): # Arrange name = "TestName"`	應用程式開發人員

任務	描述	所需技能
	description = "Test description" request = api_model.CreateProductRequest(name=name, description=description) minimal_event = api_gateway_proxy_event.APIGatewayProxyEvent({ "path": "/products", "httpMethod": "POST", "requestContext": { # correlation ID "requestId": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef" }, "body": json.dumps(request.dict()), }) create_product_func_mock = unittest.mock.create_autospec(spec=create_product_command_handler.handle_create_product_command)	

任務	描述	所需技能
	<pre> handler.c reate_product_comm and_handler.handle _create_product_co mmand = (create_pr oduct_func_mock) # Act handler.h andler(minimal_eve nt, lambda_context)</pre> 4. 運行單元測試以查看它失敗，因為仍然沒有邏輯。 <pre>python -m pytest</pre>	

任務	描述	所需技能
實作主要介面卡。	1. 建立 API 商務邏輯的函數，並將其宣告為 API 資源。 `@tracer.capture_method` `@app.post("/products")` `@utils.parse_event(model=api_model.CreateProductRequest, app_context=app)` `def create_product(` `request: api_model.CreateProductRequest,` `) -> api_model.CreateProductResponse:` `"""Creates a product."""` `...` 注意：您看到的所有裝飾器都是適用於 Python 程式庫的 AWS Lambda PowerTools 的功能。如需詳細資訊，請參閱適用於 Python 的 AWS Lambda 電源工具網站。 2. 實作 API 邏輯。 `id=create_product_command_handler.handle_create_product_command(` `command=create_product_comm`	應用程式開發人員

任務	描述	所需技能
	<pre>and.CreateProductC ommand(name=requ est.name, descripti on=request.descrip tion,), unit_of_w ork=unit_of_work,) response = api_model.CreatePr oductResponse(id=i d) return response. dict()</pre> 3. 運行單元測試以查看它成功。 <pre>python -m pytest</pre>	

相關資源

APG 指南

- 在 AWS 上建立六角形架構

AWS 參考資料

- AWS Lambda 文件
- AWS CDK 文件
  - 您的第一個 AWS CDK 應用程式
- API Gateway 文件
  - 使用 IAM 許可控制對 API 的存取

  - 使用 API Gateway 主控台測試 REST API 方法
- Amazon DynamoDB 文件

工具

- 中國政府網站
- 安裝 Git
- 創建一個新的 GitHub 存儲庫
- Python 網站
- AWS Lambda 電動工 Python
- 郵遞員網站
- Python 擬對象庫
- 詩歌網

IDE

- 視覺工作室代碼網站
- AWS Cloud9 文件
- PyCharm 網站

更多模式

- 使用 AWS CodePipeline 和 AWS 自動化堆疊集部署 CodeBuild
- 使用雲端託管人和 AWS CDK 自動將適用於 Systems Manager 的 AWS 受管政策附加至 EC2 執行個體設定檔
- 使用亞馬遜 Kinesis 影片串流和 AWS Fargate 建立影片處理管道
- 使用無伺服器方法將 AWS 服務鏈結在一起
- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 使用 AWS 副駕駛員將叢集應用程式部署到 Amazon ECS
- 使用地形部署 CloudWatch Synthetics 金絲雀
- 使用容器映像部署 Lambda 函數
- 使用 Lambda 函數、Amazon VPC 和無伺服器架構產生靜態輸出 IP 地址
- 使用 AWS AWS Glue 任務和 Python 產生測試資料
- 為多帳戶環境實施 Gitflow 分支策略 DevOps
- 為多帳戶環境實作 GitHub Flow 分支 DevOps 策略
- 為多帳戶環境實作幹線分支 DevOps 策略
- 在 AWS 上將 ASP.NET 網頁表單應用程式現代化
- 在 Amazon EC2 Linux 實例上運行一個 ASP.NET 核心網絡 API 碼頭容器
- 使用最新的框架在 AWS Glue 中對 Python ETL 任務執行單元測試
- 以 CSV 檔案將大規模的 Db2 z/OS 資料傳輸到 Amazon S3
- 在本機驗證地形表單 (AFT) 程式碼的 Account Factory

儲存與備份

主題

允許 EC2 執行個體寫入 AMS 帳戶中 S3 儲存貯體的存取權

創建者：曼西蘇拉特瓦拉（AWS）

環境：生產	技術：儲存與備份；資料庫；安全性、身分識別、合規性；營運	工作負載：所有其他工作
AWS 服務：Amazon S3；AWS Managed Services		

Summary

AWS Managed Services (AMS) 可協助您更有效率且安全地操作 Amazon Web Services (AWS) 基礎設施。AMS 帳戶具有用於標準化 AWS 資源管理的安全防護。一個保護措施是預設的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體設定檔不允許寫入存取亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。不過，您的組織可能有多個 S3 儲存貯體，並且需要對 EC2 執行個體的存取進行更多控制。例如，您可能想要將 EC2 執行個體的資料庫備份存放在 S3 儲存貯體中。

此模式說明如何使用變更請求 (RFC) 允許 EC2 執行個體寫入 AMS 帳戶中 S3 儲存貯體的存取權。RFC 是您或 AMS 在受管理環境中進行變更而建立的請求，其中包含特定作業的變更類型 (CT) ID。

先決條件和限制

先決條件

- 一個 AMS 高級帳戶。如需詳細資訊，請參閱 AWS Managed Services 文件中的 AMS 操作計劃。
- 存取 customer-mc-user-role AWS Identity and Access Management (IAM) 角色以提交 RFC。
- 使用您 AMS 帳戶中的 EC2 執行個體安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 瞭解如何在 AMS 中建立和提交 RFC。如需有關此項目的詳細資訊，請參閱什麼是 AMS 變更類型？在 AWS Managed Services 文件中。
- 了解手動和自動更改類型（CTS）。如需相關詳細資訊，請參閱 AWS Managed Services 文件中的自動化和手動 CT。

架構

技術, 堆

- AMS
- AWS CLI
- Amazon EC2
- Amazon S3
- IAM

工具

- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- AWS Managed Services (AMS) 可協助您更有效率且安全地操作 AWS 基礎設施。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以視需要啟動任意數量的虛擬伺服器，，並快速進行擴展或縮減。

史诗

使用 RFC 建立 S3 儲存貯體

任務	描述	所需技能
使用自動 RFC 建立 S3 儲存貯體。	1. 登入您的 AMS 帳戶，選擇 [選擇變更類型] 頁面，選擇 [R FC]，然後選擇 [建立 R FC]。 2. 提交建立 S3 儲存貯體自動 RFC。	AWS 系統管理員、AWS 開發人員

任務	描述	所需技能
	注意：請確定您已記錄 S3 儲存貯體的名稱。	

建立 IAM 執行個體設定檔並將其與 EC2 執行個體建立關聯

任務	描述	所需技能
提交手動 RFC 以建立 IAM 角色。	登入 AMS 帳戶後，系統會建立一個預設的 customer-mc-ec 2 個執行個體設定檔 IAM 執行個體設定檔，並將其關聯至 AMS 帳戶中的每個 EC2 執行個體。但是，執行個體設定檔沒有 S3 儲存貯體的寫入權限。 若要新增寫入許可，請提交建立 IAM 資源手冊 RFC，以建立具有以下三個政策的 IAM 角色：客戶 _ec2_instance_、客戶端政策和客戶 _ec2_s3_ 整合政策。 重要事項：您的 AMS 帳戶中已存在客戶 _ec2_instance_ 和客戶保護政策策略。不過，您需要使用下列範例原則來建立客戶 _ec2_s3_ 整合政策原則： `{ "Version": "2012-10-17", "Statement": [ { "Sid": "",`	AWS 系統管理員、AWS 開發人員

任務	描述	所需技能
	"Effect": "Allow", "Principal": { "Service": "ec2.amazonaws.com" }, "Action": "sts:AssumeRole" }] } Role Permissions: { "Version": "2012-10-17", "Statement": [{ "Action": ["s3:ListBucket", "s3:GetBucketLocation"], "Resource": "arn:aws:s3:::", "Effect": "Allow" }, { "Action": ["s3:GetObject", "s3:PutObject", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload"	

任務	描述	所需技能
	<pre>], "Resource ": "arn:aws:s3:::/*", "Effect": "Allow" }] }</pre>	
提交手動 RFC 以取代 IAM 執行個體設定檔。	提交手動 RFC，將目標 EC2 執行個體與新的 IAM 執行個體設定檔建立關聯。	AWS 系統管理員、AWS 開發人員
測試 S3 儲存貯體的複製作業。	在 AWS CLI 中執行下列命令，以測試 S3 儲存貯體的複製操作：`aws s3 cp test.txt s3://<S3 Bucket>/test2.txt`	AWS 系統管理員、AWS 開發人員

相關資源

- 為您的 Amazon EC2 執行個體建立 IAM 執行個體設定檔
- 建立 S3 儲存貯體 (使用 Amazon S3 主控台、AWS 開發套件或 AWS CLI)

使用雪花雪花管、Amazon S3、Amazon SNS 和 Amazon 資料 Firehose，將資料串流擷取自動化到雪花資料庫

創建者比卡什錢德拉魯特（AWS）

環境：PoC 或試點	技術：儲存與備份

Summary

此模式說明如何使用 Amazon Web 服務 (AWS) 雲端上的服務來處理連續的資料串流，並將其載入雪花資料庫。該模式使用 Amazon 數據 Firehose 將數據交付到亞馬遜簡單存儲服務（Amazon S3），亞馬遜簡單通知服務（Amazon SNS）在收到新數據時發送通知，而雪花雪花管將數據加載到雪花數據庫中。

通過遵循此模式，您可以在幾秒鐘內連續生成可用於分析的數據，避免多個手動 COPY 命令，並在負載時完全支持半結構化數據。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 持續將資料傳送至 Firehose 交付串流的資料來源。
- 從 Firehose 交付串流接收資料的現有 S3 儲存貯體。
- 一個活躍的雪花帳戶。

限制

- 雪花雪管不會直接連接到 Firehose。

架構

技術, 堆

- Amazon 數據 Firehose
- Amazon SNS
- Amazon S3
- 雪花雪管
- 雪花資料庫

工具

- Firehose — Amazon 資料 Firehose 是一項全受管服務，可將即時串流資料交付到 Amazon S3、Amazon Redshift、亞馬遜 OpenSearch 服務、Splunk 以及受支援的第三方服務供應商擁有的任何自訂 HTTP 端點或 HTTP 端點等目的地。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- Amazon SNS — Amazon Simple Notification Service (Amazon SNS) 會協調和管理訊息傳送或傳送給訂閱端點或用戶端的工作。
- 雪花-雪花是作為 S oftware-as-a 服務（SaaS）提供的分析數據倉庫。
- 雪花雪管 — 雪管載入從文件中的數據，只要他們在雪花階段可用。

史诗

設置雪花雪管

任務	描述	所需技能
在雪花中創建一個 CSV 文件。	登入 Snowflake 並執行「建立檔案格式」命令，以指定欄位分隔符號建立 CSV 檔案。如需有關此命令和其他 Snowflake 命令的詳細資訊，請參閱 < 其他資訊 > 一節。	開發人員
創建一個外部雪花舞台。	執行「建立階段」命令，建立參照您先前建立之 CSV 檔案的外部雪花階段。重要：您需	開發人員

任務	描述	所需技能
	要 S3 儲存貯體的 URL、AWS 存取金鑰和 AWS 秘密存取金鑰。執行「SHOW STAGES」命令來確認已建立「雪花」階段。	
創建雪花目標表。	運行「創建表」命令來創建雪花表。	開發人員
創建一個管道。	運行「創建管道」命令; 確保命令中的「auto_ingest= 真」。執行「顯示 PIPES」指令以確認管道已建立。複製並儲存「通知 _ 通道」欄值。此值將用於設定 Amazon S3 事件通知。	開發人員

設定 S3 儲存貯體

任務	描述	所需技能
為 S3 儲存貯體建立 30 天的生命週期政策。	登入 AWS 管理主控台並開啟 Amazon S3 主控台。選擇包含來自 Firehose 資料的 S3 儲存貯體。然後選擇 S3 儲存貯體中的「管理」索引標籤，然後選擇「新增生命週期規則」。在「生命週期規則」對話方塊中輸入規則的名稱，並為值區設定 30 天的生命週期規則。如需此和其他故事的說明，請參閱「相關資源」一節。	系統管理員，開發者
為 S3 儲存貯體建立 IAM 政策。	開啟 AWS Identity and Access Management (IAM) 主控台，	系統管理員，開發者

任務	描述	所需技能
	然後選擇「政策」。選擇「創建策略」，然後選擇「JSON」選項卡。將原則從「其他資訊」區段複製並貼到 JSON 欄位中。此原則將授與「PutObjectDeleteObject」和「」權限，以及「GetObject GetObject Version,」和「ListBucket」權限。選擇「檢閱策略」，輸入策略名稱，然後選擇「建立策略」。	
將政策指派給 IAM 角色。	開啟 IAM 主控台，選擇「角色」，然後選擇「建立角色」。選擇「另一個 AWS 帳戶」做為受信任的實體。輸入您的 AWS 帳戶 ID，然後選擇「需要外部 ID」。輸入預留位置 ID，稍後您將進行變更。選擇 [下一步] 並指派您先前建立的 IAM 政策。然後建立 IAM 角色。	系統管理員，開發者
複製 IAM 角色的 Amazon 資源名稱 (ARN)。	開啟 IAM 主控台，然後選擇「角色」。選擇您先前建立的 IAM 角色，然後複製並儲存「角色 ARN」。	系統管理員，開發者

在雪花中設定儲存整合

任務	描述	所需技能
在雪花中建立儲存整合。	登入雪花並執行「建立儲存整合」命令。這會修改信任關係、授與 Snowflake 的存取	系統管理員，開發者

任務	描述	所需技能
	權，以及提供 Snowflake 階段的外部 ID。	
擷取雪花帳戶的 IAM 角色。	執行「描述整合」命令以擷取 IAM 角色的 ARN。重要事<integration_ name>項：這是您先前建立的 Snowflake 儲存整合的名稱。	系統管理員，開發者
記錄兩列值。	複製並儲存「儲存空間 _aws_iam_arn」和「儲存空間 _ 外部識別碼」資料行的值。	系統管理員，開發者

允許雪花雪管存取 S3 儲存貯體

任務	描述	所需技能
修改 IAM 角色政策。	開啟 IAM 主控台，然後選擇「角色」。選擇您之前建立的 IAM 角色，然後選擇「信任關係」索引標籤。選擇「編輯信任關係」。將「雪花外部識別碼」取代為您之前複製的「儲存空間 _ aws_外部_id」值。將「雪花」取代為您之前複製的儲存空間值。然後選擇「更新信任策略」	系統管理員，開發者

開啟並設定 S3 儲存貯體的 SNS 通知

任務	描述	所需技能
開啟 S3 儲存貯體的事件通知。	開啟 Amazon S3 主控台並選擇您的儲存貯體。選擇「內	系統管理員，開發者

任務	描述	所需技能
	容」，然後在「高級設置」下選擇「事件」。選擇「新增通知」，然後輸入此事件的名稱。如果您沒有輸入名稱，則會使用全域唯一識別碼 (GUID)。	
為 S3 儲存貯體設定 Amazon SNS 通知。	在「事件」下，選擇「ObjectCreate（全部）」，然後在「發送到」下拉列表中選擇「SQS 隊列」。在「SNS」列表中選擇「添加 SQS 隊列 ARN」，然後粘貼您之前複製的「通知」值。然後選擇「保存」	系統管理員，開發者
訂閱雪花式 SQS 佇列至 SNS 主題。	將雪花 SQS 佇列訂閱至您建立的 SNS 主題。如需此步驟的說明，請參閱「相關資源」一節。	系統管理員，開發者

檢查雪花階段整合

任務	描述	所需技能
檢查和測試雪管。	登入「雪花」並開啟「雪花」階段。將文件拖放到 S3 存儲桶中，並檢查雪花表是否加載它們。當 S3 儲存貯體中出現新物件時，Amazon S3 會將 SNS 通知傳送至雪管。	系統管理員，開發者

相關資源

- 為 S3 儲存貯體建立生命週期政策
- 訂閱雪花 SQS 佇列至 Amazon SNS 主題

其他資訊

建立檔案格式：

```
CREATE FILE FORMAT <name>
TYPE = 'CSV'
FIELD_DELIMITER = '|'
SKIP_HEADER = 1;
```

創建一個外部舞台：

```
externalStageParams (for Amazon S3) ::=
  URL = 's3://[//]'

  [ { STORAGE_INTEGRATION =  } | { CREDENTIALS = ( {  { AWS_KEY_ID = `` AWS_SECRET_KEY
 = `` [ AWS_TOKEN = `` ] } | AWS_ROLE = ``  } ) ) }` ]
  [ ENCRYPTION = ( [ TYPE = 'AWS_CSE' ] [ MASTER_KEY = '' ] |
                   [ TYPE = 'AWS_SSE_S3' ] |
                   [ TYPE = 'AWS_SSE_KMS' [ KMS_KEY_ID = '' ] |
                   [ TYPE = NONE ] )
```

創建一個表：

```
CREATE [ OR REPLACE ] [ { [ LOCAL | GLOBAL ] TEMP[ORARY] | VOLATILE } | TRANSIENT ]
 TABLE [ IF NOT EXISTS ]
  <table_name>
    ( <col_name> <col_type> [ { DEFAULT <expr>
                                | { AUTOINCREMENT | IDENTITY } [ ( <start_num> ,
 <step_num> ) | START <num> INCREMENT <num> ] } ]
                                 /* AUTOINCREMENT / IDENTITY supported only for numeric
 data types (NUMBER, INT, etc.) */
                              [ inlineConstraint ]
      [ , <col_name> <col_type> ... ]
      [ , outoflineConstraint ]
      [ , ... ] )
```

```
  [ CLUSTER BY ( <expr> [ , <expr> , ... ] ) ]
  [ STAGE_FILE_FORMAT = ( { FORMAT_NAME = '<file_format_name>'
                            | TYPE = { CSV | JSON | AVRO | ORC | PARQUET | XML }
 [ formatTypeOptions ] } ) ]
  [ STAGE_COPY_OPTIONS = ( copyOptions ) ]
  [ DATA_RETENTION_TIME_IN_DAYS = <num> ]
  [ COPY GRANTS ]
  [ COMMENT = '<string_literal>' ]
```

顯示階段：

```
SHOW STAGES;
```

創建一個管道：

```
CREATE [ OR REPLACE ] PIPE [ IF NOT EXISTS ]
  [ AUTO_INGEST = [ TRUE | FALSE ] ]
  [ AWS_SNS_TOPIC =  ]
  [ INTEGRATION = '' ]
  [ COMMENT = '' ]
  AS
```

顯示管道：

```
SHOW PIPES [ LIKE '<pattern>' ]
           [ IN { ACCOUNT | [ DATABASE ] <db_name> | [ SCHEMA ] <schema_name> } ]
```

建立儲存整合：

```
CREATE STORAGE INTEGRATION <integration_name>
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = S3
  ENABLED = TRUE
  STORAGE_AWS_ROLE_ARN = '<iam_role>'
  STORAGE_ALLOWED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/')
  [ STORAGE_BLOCKED_LOCATIONS = ('s3://<bucket>/<path>/', 's3://<bucket>/<path>/') ]
```

範例：

```
create storage integration s3_int
```

```
  type = external_stage
  storage_provider = s3
  enabled = true
  storage_aws_role_arn = 'arn:aws:iam::001234567890:role/myrole'
  storage_allowed_locations = ('s3://mybucket1/mypath1/', 's3://mybucket2/mypath2/')
  storage_blocked_locations = ('s3://mybucket1/mypath1/sensitivedata/', 's3://
mybucket2/mypath2/sensitivedata/');
```

如需有關此步驟的詳細資訊，請參閱透過雪花文件設定雪花儲存整合以存取 Amazon S3。

描述一個集成：

```
DESC INTEGRATION <integration_name>;
```

S3 儲存貯體政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:GetObjectVersion",
              "s3:DeleteObject",
              "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::",
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "/*"
                    ]
                }
            }
        }
    ]
```

```
}
```

自動加密現有和新的 Amazon EBS 磁碟區

由托尼 DeMarco （AWS）和喬希·喬希（AWS）創建

代碼存儲庫:https://github.com/aws-samples/ aws-system-manager-automation-unencrypted-to-encrypted-resources / 樹/主/EBS	環境：生產	技術：儲存與備份、安全性、身分識別、合規性、管理與治理
AWS 服務：AWS Config；Amazon EBS；AWS KMS；AWS Organizations；AWS Systems Manager		

Summary

Amazon Elastic Block Store (Amazon EBS) 磁碟區的加密對於組織的資料保護策略很重要。這是建立一個結構良好的環境的重要一步。雖然沒有直接加密現有未加密的 EBS 磁碟區或快照的方法，但您可以透過建立新的磁碟區或快照來加密它們。如需詳細資訊，請參閱 Amazon EC2 文件中的加密 EBS 資源。此模式提供預防性和偵測控制，可加密新的和現有的 EBS 磁碟區。在此模式中，您可以設定帳戶設定、建立自動補救程序，以及實作存取控制。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶
- AWS Command Line Interface (AWS CLI) (AWS CLI)，在 macOS、Linux 或視窗上安裝和設定
- jq, 安裝和配置在 macOS, Linux, 或視窗
- AWS Identity and Access Management (IAM) 許可的佈建為具有 AWS CloudFormation、亞馬遜彈性運算雲端 (Amazon EC2)、AWS Systems Manager、AWS Config 和 AWS Key Management Service (AWS KMS) 的讀寫存取權
- AWS Organizations 的設定已啟用所有功能，這是服務控制政策的要求
- 目標帳戶中已啟用 AWS Config

限制

- 在您的目標 AWS 帳戶中，不得有名為加密磁碟區的 AWS Config 規則。此解決方案會以此名稱部署規則。具有此名稱的預先存在規則可能會導致部署失敗，並導致與多次處理相同規則相關的不必要費用。
- 此解決方案會使用相同的 AWS KMS 金鑰加密所有 EBS 磁碟區。
- 如果您為帳戶啟用 EBS 磁碟區加密，則此設定為區域特定。如果您為 AWS 區域啟用此功能，則無法針對該區域中的個別磁碟區或快照停用該功能。如需詳細資訊，請參閱 Amazon EC2 文件中的預設加密。
- 修復現有未加密的 EBS 磁碟區時，請確保 EC2 執行個體未使用中。此自動化功能會關閉執行個體，以便中斷未加密的磁碟區並連接加密的磁碟區。正在進行修復時會有停機時間。如果這對您的組織來說是重要的基礎結構，請確定已設置手動或自動的高可用性設定，以免影響執行個體上執行的任何應用程式的可用性。建議您僅在標準維護時段期間修復重要資源。

架構

自動化流程

1. AWS Config 偵測到未加密的 EBS 磁碟區。
2. 管理員使用 AWS Config 將修復命令傳送給 Systems Manager 員。
3. 系 Systems Manager 自動化會擷取未加密 EBS 磁碟區的快照。
4. 系 Systems Manager 自動化使用 AWS KMS 建立快照的加密副本。
5. Systems Manager 自動化會執行下列作業：
   a. 停止受影響的 EC2 執行個體 (如果執行中)
   b. 將新的加密磁碟區副本附加至 EC2 執行個體
   c. 將 EC2 實例返回到其原始狀態

工具

AWS 服務

- AWS CLI — AWS Command Line Interface (AWS CLI) (AWS CLI) 可讓您直接存取 AWS 服務的公有應用程式程式設計界面 (API)。您可以使用 AWS CLI 探索服務的功能，並開發殼層指令碼來管理

資源。除了低階 API 等效命令之外，數個 AWS 服務還為 AWS CLI 提供自訂服務。自訂功能可能包括較高階的命令，可簡化具有複雜 API 的服務使用。

- AWS CloudFormation — AWS CloudFormation 是一項可協助您建立 AWS 資源模型和設定 AWS 資源的服務。您可以建立範本來描述所需的所有 AWS 資源 (例如 Amazon EC2 執行個體)，並為您 CloudFormation 佈建和設定這些資源。
- AWS 組態 — AWS Config 提供 AWS 帳戶中 AWS 資源組態的詳細檢視。這包含資源彼此之間的關係和之前的組態方式，所以您可以看到一段時間中組態和關係的變化。
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，提供可調整大小的運算容量，讓您用來建立和託管軟體系統。
- AWS KMS — AWS Key Management Service (AWS KMS) 是針對雲端擴展的加密和金鑰管理服務。其他 AWS 服務會使用 AWS KMS 金鑰和功能，您可以使用這些金鑰和功能來保護 AWS 環境中的資料。
- AWS Organizations — AWS Organizations 是一種帳戶管理服務，可讓您將多個 AWS 帳戶合併到您建立並集中管理的組織中。
- AWS Systems Manager Automation — Systems Manager 自動化可簡化 Amazon EC2 執行個體和其他 AWS 資源的常見維護和部署任務。

其他服務

- jq-jq 是一個輕量級和靈活的命令行 JSON 處理器。您可以使用此工具從 AWS CLI 輸出擷取關鍵資訊。

Code

- 此模式的程式碼可在使用客戶 KMS 金鑰存放庫 GitHub 自動修復未加密的 EBS 磁碟區中找到。

史诗

自動修復未加密磁碟區

任務	描述	所需技能
下載腳本和 CloudFormation 模板。	從使用客戶 KMS 金鑰存放庫 GitHub 自動修復未加密的 EBS 磁碟區，下載殼層指令碼、	AWS 管理員，一般 AWS

任務	描述	所需技能
	JSON 檔案和 CloudFormation 範本。	
識別 AWS KMS 金鑰的管理員。	1. 登入 AWS 管理主控台，然後前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。 2. 識別將成為 AWS KMS 金鑰管理員的使用者或角色。如果需要為此目的建立新使用者或角色，請立即建立它。如需詳細資訊，請參閱 IAM 文件中的 IAM 身分。此自動化操作會建立新的 AWS KMS 金鑰。 3. 識別後，複製使用者或角色的 Amazon 資源名稱 (ARN)。如需詳細資訊，請參閱 IAM 說明文件中的 IAM ARN。您可以在下一個步驟中使用此 ARN。	AWS 管理員，一般 AWS

任務	描述	所需技能
部署堆疊 1 CloudFormation 範本。	1. 開啟 AWS 主 CloudFormation 控台，網址為 https://console.aws.amazon.com/cloudformation/。 2. 在中 CloudFormation，部署`Stack1.yaml` 範本。請注意下列部署詳細資訊： • 給堆棧一個清晰和描述性的名稱。請記下堆疊名稱，因為您在下一個步驟中需要此值。 • 將金鑰管理員的 ARN 貼到 Stack1 的唯一參數欄位中。此使用者或角色會成為堆疊所建立之 AWS KMS 金鑰的管理員。 如需部署 CloudFormation 範本的詳細資訊，請參閱 CloudFormation 文件中的使用 AWS CloudFormation 範本。	AWS 管理員，一般 AWS
部署堆疊 2 CloudFormation 範本。	在中 CloudFormation，部署`Stack2.yaml` 範本。請注意下列部署詳細資訊： • 給堆棧一個清晰和描述性的名稱。 • 對於 Stack2 的唯一參數，請輸入您在上一個步驟中建立的堆疊名稱。這可讓 Stack2 參考上一個步驟中堆疊部署的新 AWS KMS 金鑰和角色。	AWS 管理員，一般 AWS

任務	描述	所需技能
建立未加密的磁碟區進行測試。	使用未加密的 EBS 磁碟區建立 EC2 執行個體。如需指示，請參閱 Amazon EC2 文件中的建立 Amazon EBS 磁碟區。實例類型並不重要，並且不需要訪問實例。您可以建立 t2.micro 執行個體以保留在免費方案中，而且不需要建立 key pair。	AWS 管理員，一般 AWS

任務	描述	所需技能
測試 AWS Config 規則。	1. 開啟 AWS 組態主控台，網址為 https://console.aws.amazon.com/config/。在 [規則] 頁面上，選擇加密磁碟區規則。 2. 確認新的未加密測試實例出現在不合規資源列表中。如果磁碟區沒有立即出現，請等待幾分鐘，然後重新整理結果。AWS Config 規則會在建立執行個體和磁碟區後不久偵測到資源變更。 3. 選取資源，然後選擇 [修正]。 您可以在「Systems Manager」中檢視修復進度和狀態，如下所示： 1. 開啟 AWS Systems Manager 主控台，網址為 https://console.aws.amazon.com/systems-manager/。 2. 在導覽窗格中，選擇 Automation (自動化)。 3. 選擇「執行 ID」連結以檢視步驟和狀態。	AWS 管理員，一般 AWS
設定其他帳戶或 AWS 區域。	根據您的使用案例需要，針對任何其他帳戶或 AWS 區域重複此史詩。	AWS 管理員，一般 AWS

啟用 EBS 磁碟區的帳戶層級加密

任務	描述	所需技能
執行啟用指令碼。	1. 在 bash shell 中，使用cd命令導航到克隆的存儲庫中。 2. 輸入以下命令以執行 `enable-ebs-encryption-for-account` 指令碼。 `./Bash/enable-ebs-encryption-for-account.sh`	AWS 管理員，一般 AWS，bash
確認設定已更新。	1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。 2. 在畫面右側的 [設定] 下，選擇 [資料保護與安全性]。 3. 在 [EBS 加密] 區段下，確認已開啟 [永遠加密新 EBS 磁碟區]，且 [預設加密金鑰] 設定為您先前指定的 ARN。 注意：如果 [永遠加密新的 EBS 磁碟區] 設定為關閉，或金鑰仍設定為別名 /aws/ebs，請確認您已登入執行 shell 指令碼的相同帳戶和 AWS 區域，並檢查殼層是否有錯誤訊息。	AWS 管理員，一般 AWS
設定其他帳戶或 AWS 區域。	根據您的使用案例需要，針對任何其他帳戶或 AWS 區域重複此史詩。	AWS 管理員，一般 AWS

防止建立未加密的執行個體

任務	描述	所需技能
建立服務控制原則。	1. 開啟 AWS Organizations 主控台，網址為 https://console.aws.amazon.com/organizations/v2/。 2. 建立新的服務控制原則。如需詳細資訊，請參閱 AWS Organizations 文件中的建立服務控制政策。 3. 將的內容新增`DenyUnencryptedEC2.json` 至策略並儲存。您從第一個史詩中的 GitHub 存儲庫下載了此 JSON 文件。 4. 將此原則附加至組織根目錄或任何必要的組織單位 (OU)。如需詳細資訊，請參閱 AWS Organizations 文件中的連接和卸離服務控制政策。	AWS 管理員，一般 AWS

相關資源

AWS 服務文件

- AWS CLI
- AWS Config
- AWS CloudFormation
- Amazon EC2
- AWS KMS
- AWS Organizations
- AWS Systems Manager Automation

其他資源

- jq 手冊（jq 網站）
- jq 下載（ ）GitHub

在 AWS 雲端上的 Sun SPARC 伺服器備份 Sun 字元 SSP 模擬器

創建者：凱文容（AWS），路易斯·拉莫斯（斯特羅馬斯）和羅希特·達吉（AWS）

環境：生產	技術：存儲和備份; 操作系統; DevOps	工作量：甲骨文
AWS 服務：Amazon EFS；Amazon S3；AWS Storage Gateway；AWS Systems Manager；Amazon EC2		

Summary

從現場部署環境遷移到亞馬遜網路服務 (AWS) 雲端後，此模式提供四個備份 Sun 微系統 SPARC 伺服器的選項。這些備份選項可協助您實作符合組織復原點目標 (RPO) 和復原時間目標 (RTO) 的備份計畫、使用自動化方法，並降低整體作業成本。此模式提供四個備份選項的概觀，以及實作這些選項的步驟。

如果您在 S tromasys 字元 SSP 模擬器上使用以客體形式託管的 Sun SPARC 伺服器，則可以使用下列三個備份選項之一：

- Backup 選項 1：Stromasys 虛擬磁帶 — 使用 Charon-SSP 虛擬磁帶功能在 Sun SPARC 伺服器中設定備份設施，並使用 AWS Systems Manager Automation 將備份檔案存檔至 Amazon 簡單儲存服務 (Amazon S3) 和亞馬遜簡單儲存服務冰川。
- Backup 選項 2：Stromasys 快照 — 使用 Charon-SSP 快照功能為 Sun SPARC 客體伺服器在 Charon-SSP 中設定備份功能。
- B@@ ackup 選項 3：亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區快照 — 如果您在亞馬遜彈性運算雲端 (Amazon EC2) 上託管 Charon-SSP 模擬器，則可以使用 Amazon EBS 磁碟區快照為 Sun SPARC 檔案系統建立備份。

如果您在 Amazon EC2 上使用以客體形式託管的 Sun SPARC 伺服器，並在 Amazon EC2 上使用字元 SSP，則可以使用下列備份選項：

- Backup 選項 4：AWS Storage Gateway 虛擬磁帶櫃 (VTL) — 使用備份應用程式搭配 S torage Gateway VTL 磁帶閘道來備份 Sun SPARC 伺服器。

如果您在 Sun SPARC 伺服器中使用裝載為品牌區域的 Sun SPARC 伺服器，則可以使用備份選項 1、2 和 4。

斯特羅馬斯提供的軟體和服務來模擬傳統的 SPARC、阿爾法、VAX 和 PA-RISC 關鍵系統。如需有關使用 Stromasys 模擬移轉到 AWS 雲端的詳細資訊，請參閱 AWS 部落格上的使用 Stromasys 將 SPARC、Alpha 或其他舊版系統重新託管到 AWS。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 現有的 Sun 伺服器。
- 卡朗 SSP 的現有授權。可從 AWS Marketplace 取得 Charon-SSP 的授權，而 Stromasys 虛擬環境 (VE) 的授權則可從 Stromasys 取得。有關更多信息，請聯繫 Stromasys 的銷售部門。
- 熟悉太陽公司伺服器和 Linux 備份。
- 熟悉卡朗-SSP 仿真技術。有關此方面的詳細資訊，請參閱 Stromasys 文件中的 Stromasys 舊版伺服器模擬。
- 如果您想要使用 Sun SPARC 伺服器檔案系統的虛擬磁帶設備或備份應用程式，則必須建立並設定 Sun SPARC 伺服器檔案系統的備份功能。
- 對 RPO 和 RTO 的理解。如需詳細資訊，請參閱 AWS Well-Architected Framework 文件中的可靠性支柱白皮書中的災難復原目標。
- 要使用 Backup 選項 4，您必須具有以下內容：
  - 支援 Storage Gateway VTL 磁帶閘道的軟體式備份應用程式。如需這方面的詳細資訊，請參閱 AWS Storage Gateway 文件中的使用 VTL 裝置。
  - 安裝和配置 Bacula 主任或類似的備份應用程序。有關這方面的更多信息，請參閱 Bacula 導演文檔。

下表提供此病毒碼中四個備份選項的相關資訊。

Backup 選項	達到崩潰一致性？	實現應用一致性？	虛擬備份設備解決方案？	典型使用案例
選項 1 — 虛擬磁帶	是	是	是	Sun SPARC 伺服器檔案系統使

	您可以將 Sun SPARC 檔案系統快照自動化，以備份虛擬磁帶中的資料。例如，您可以使用 UFS 或 ZFS 快照。	此備份選項需要自動化指令碼來清除執行中的交易、在檔案系統快照集期間設定唯讀或暫時離線模式，或進行應用程式資料傾印。您可能還需要應用程式停機時間或唯讀模式。		用 .tar 或 .zip 檔案進行備份 應用資料備份
選項 2 — 快照	是 您必須設定 Charon-SSP 管理員或使用命令列啟動引數來啟用此功能。 您也必須執行 Linux 命令，要求字元 SSP 模擬器將 Sun SPARC 客體伺服器狀態儲存到快照檔案中。 重要：您必須關閉 Sun SPARC 客體伺服器。	是 此備份選項會建立模擬來賓伺服器的快照，包括其虛擬磁碟和記憶體傾印。 重要：您必須在快照集期間關閉 Sun SPARC 客體伺服器。	否	太陽 SPARC 伺服器快照 應用資料備份

選項 3 — Amazon EBS 磁碟區快照	是 您可以使用 AWS Backup 將 Amazon EBS 快照自動化。	是 此備份選項需要自動化指令碼來清除執行中交易，並在 Amazon EBS 磁碟區快照期間設定 EC2 執行個體的唯讀或暫時停止。 重要：此備份選項可能需要應用程式停機或唯讀模式，才能達到應用程式的一致	否	Sun SPARC 伺服器檔案系統快照 應用資料備份
選項 4 — AWS Storage Gateway VTL	是 您可以使用備份代理程式，將 Sun SPARC 檔案系統備份資料自動備份至 VTL。	是 此備份選項需要自動化指令碼來清除執行中的交易，並在檔案系統快照或應用程式資料傾印期間設定唯讀或暫時離線模式。 重要事項：此備份選項可能需要應用程式停機或唯讀模式。	是	大型 Sun SPARC 伺服器檔案系統備份 應用資料備份

限制

- 您可以使用此病毒碼的方法來備份個別的 Sun SPARC 伺服器，但如果您有在叢集中執行的應用程式，也可以針對共用資料使用這些備份選項。

工具

Backup 選項 1：掃描虛擬磁帶

- S@@ tromasys 字符-SSP 模擬器 — 卡朗-SSP 模擬器在標準 64 位 x86 兼容計算機系統內創建原始 SPARC 硬件的虛擬副本。它會執行原始的 SPARC 二進位程式碼，包括作業系統 (OS)，例如 SunOS 或 Solaris、其分層產品和應用程式。
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，可提供可調整大小的運算容量，讓您用來建立和託管軟體系統。
- Amazon EFS — Amazon Elastic File System (Amazon EFS) 提供簡單的無伺服器 set-and-forget 彈性檔案系統，可與 AWS 雲端服務和現場部署資源搭配使用。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- Amazon S3 Glacier — Amazon Simple Storage Service Glacier 是一種安全、耐用且極低成本的 Amazon S3 儲存類別，適用於資料存檔和長期備份。
- AWS Systems Manager Automation — 自動化是 AWS Systems Manager 的一項功能，可簡化 EC2 執行個體和其他 AWS 資源的常見維護和部署任務。

Backup 選項 2：快照

- S@@ tromasys 字符-SSP 模擬器 — 卡朗-SSP 模擬器在標準 64 位 x86 兼容計算機系統內創建原始 SPARC 硬件的虛擬副本。它會執行原始的 SPARC 二進位程式碼，包括作業系統 (例如 SunOS 或 Solaris)、其分層產品和應用程式。
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，可提供可調整大小的運算容量，讓您用來建立和託管軟體系統。
- Amazon EFS — Amazon Elastic File System (Amazon EFS) 提供簡單的無伺服器 set-and-forget 彈性檔案系統，可與 AWS 雲端服務和現場部署資源搭配使用。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- Amazon S3 Glacier — Amazon Simple Storage Service Glacier 是一種安全、耐用且極低成本的 Amazon S3 儲存類別，適用於資料存檔和長期備份。

- AWS Systems Manager Automation — 自動化是 AWS Systems Manager 的一項功能，可簡化 EC2 執行個體和其他 AWS 資源的常見維護和部署任務。

Backup 選項 3：Amazon EBS 磁碟區快照

- S@@ tromasys 字符-SSP 模擬器 — 卡朗-SSP 模擬器在標準 64 位 x86 兼容計算機系統內創建原始 SPARC 硬件的虛擬副本。它會執行原始的 SPARC 二進位程式碼，包括作業系統 (例如 SunOS 或 Solaris)、其分層產品和應用程式。
- AWS Backup — AWS Backup 是一種全受管的資料保護服務，可讓您輕鬆地跨 AWS 服務、雲端和現場部署進行集中和自動化。
- Amazon EBS — 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，以便與 EC2 執行個體搭配使用。
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，可提供可調整大小的運算容量，讓您用來建立和託管軟體系統。

Backup 選項 4：AWS Storage Gateway VTL

- S@@ tromasys 字符-SSP 模擬器 — 卡朗-SSP 模擬器在標準 64 位 x86 兼容計算機系統內創建原始 SPARC 硬件的虛擬副本。它會執行原始的 SPARC 二進位程式碼，包括作業系統 (例如 SunOS 或 Solaris)、其分層產品和應用程式。
- Bacula — Bacula 是一個開放源代碼的企業級計算機備份系統。如需現有備份應用程式是否支援磁帶閘道的詳細資訊，請參閱 AWS Storage Gateway 說明文件中的磁帶閘道支援的第三方備份應用程式。
- Amazon EC2 — Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，可提供可調整大小的運算容量，讓您用來建立和託管軟體系統。
- Amazon RDS for MySQL 適用於 MySQL — Amazon Relational Database Service 服務 (Amazon RDS) 支援執行多個 MySQL 版本的資料庫執行個體。
- Amazon S3 — Amazon Simple Storage Service (Amazon S3) 是互聯網的存儲。
- Amazon S3 Glacier — Amazon Simple Storage Service Glacier 是一種安全、耐用且極低成本的 Amazon S3 儲存類別，適用於資料存檔和長期備份。
- AWS Storage Gateway — Storage Gateway 將現場部署軟體設備與雲端儲存連接起來，以便與現場部署 IT 環境和 AWS 儲存基礎設施之間的資料安全功能無縫整合。

史诗

Backup 選項 1 — 建立 Stromasys 虛擬磁帶備份

任務	描述	所需技能
為虛擬磁帶檔案儲存建立 Amazon EFS 共用檔案系統。	登入 AWS 管理主控台或使用 AWS CLI 建立 Amazon EFS 檔案系統。 如需這方面的詳細資訊，請參閱 Amazon EFS 文件中的建立 Amazon EFS 檔案系統。	雲端架構師
將 Linux 主機設定為掛載共用檔案系統。	在亞馬遜 EC2 Linux 執行個體上安裝 Amazon EFS 驅動程式，並將 Linux 作業系統設定為在啟動期間掛接 Amazon EFS 共用檔案系統。 如需這方面的詳細資訊，請參閱 Amazon EFS 文件中的使用 EFS 掛載協助程式掛接檔案系統。	DevOps 工程師
安裝字符-SSP 模擬器。	在 Amazon EC2 Linux 執行個體上安裝字元 SSP 模擬器。 有關這方面的詳細資訊，請參閱 Stromasys 文件中的為 Charon-SSP 設定 AWS 雲端執行個體。	DevOps 工程師
在共用檔案系統中為每部 Sun SPARC 客體伺服器建立虛擬磁帶檔案容器。	執行此命`touch <vtape-container-name>` 令，以在 Charon-SSP 模擬器中部署的每部 Sun SPARC 客體伺服	DevOps 工程師

任務	描述	所需技能
	器，在共用檔案系統中建立虛擬磁帶檔案容器。	
將字元 SSP 管理員設定為 Sun SPARC 客體伺服器建立虛擬磁帶裝置。	登入 Charon-SSP 管理員，建立虛擬磁帶裝置，並將其設定為使用每部 Sun SPARC 客體伺服器的虛擬磁帶容器檔案。 如需有關此功能的詳細資訊，請參閱 Stromasys 文件中的 Linux 版 Charon-SSP 5.2 使用者指南。	DevOps 工程師
驗證 Sun SPARC 客體伺服器中是否可使用虛擬磁帶裝置。	登入每部 Sun SPARC 客體伺服器並執行`mt -f /dev/rmt/1` 命令，以驗證虛擬磁帶裝置是否已在作業系統中設定。	DevOps 工程師
開發系 Systems Manager 自動化手冊和自動化。	開發系 Systems Manager 自動化手冊，並在 Systems Manager 中設置維護窗口和關聯以計劃備份過程。 如需有關這方面的詳細資訊，請參閱 AWS Systems Manager 文件中的自動化逐步解說和設定維護時段。	雲端架構師
設定 Systems Manager 自動化以封存旋轉的虛擬磁帶容器檔案。	使用其他資訊一節中後退選項 1 的程式碼範例，開發 Systems Manager 自動化工作流程簿，將旋轉的虛擬磁帶容器檔案存檔到 Amazon S3 和 Amazon S3 Glacier。	雲端架構師

任務	描述	所需技能
部署系 Systems Manager 自動化手冊以進行封存和排程。	部署系 Systems Manager 自動化手冊，並將其排程為在 Systems Manager 中自動執行。 如需有關此項目的詳細資訊，請參閱 Systems Manager 說明文件中的自動化逐步解說。	雲端架構師

Backup 選項 2 — 建立快照

任務	描述	所需技能
為虛擬磁帶檔案儲存建立 Amazon EFS 共用檔案系統。	登入 AWS 管理主控台或使用 AWS CLI 建立 Amazon EFS 檔案系統。 如需這方面的詳細資訊，請參閱 Amazon EFS 文件中的建立您的 Amazon EFS 檔案系統。	雲端架構師
將 Linux 主機設定為掛載共用檔案系統。	在 Amazon EC2 Linux 執行個體中安裝 Amazon EFS 驅動程式，並將 Linux 作業系統設定為在啟動期間掛接 Amazon EFS 共用檔案系統。 如需這方面的詳細資訊，請參閱 Amazon EFS 文件中的使用 EFS 掛載協助程式掛接檔案系統。	DevOps 工程師
安裝字符-SSP 模擬器。	在 Amazon EC2 Linux 執行個體上安裝字元 SSP 模擬器。	DevOps 工程師

任務	描述	所需技能
	有關這方面的詳細資訊，請參閱 Stromasys 文件中的為 Charon-SSP 設定 AWS 雲端執行個體。	
將 Sun SPARC 客體伺服器設定為使用快照選項啟動。	使用字元-SSP 管理員為每個 Sun SPARC 客體伺服器設定快照選項。 如需有關此功能的詳細資訊，請參閱 Stromasys 文件中的 Linux 版 Charon-SSP 5.2 使用者指南。	DevOps 工程師
開發系 Systems Manager 自動化手冊。	使用 [其他資訊] 區段中 [Backup] 選項 2 的程式碼範例，開發 Systems Manager 自動化執行手冊，以便在維護時段期間在 Sun SPARC 客體伺服器上遠端執行快照命令。	雲端架構師
部署系 Systems Manager 自動化手冊，並設定與 Amazon EC2 Linux 主機的關聯。	部署 Systems Manager 自動化手冊，並在系 Systems Manager 中設定維護時段和關聯，以排程備份程序。 如需有關這方面的詳細資訊，請參閱 AWS Systems Manager 文件中的自動化逐步解說和設定維護時段。	雲端架構師

任務	描述	所需技能
將快照封存至長期儲存空間。	使用其他資訊一節中的工作流程簿範例程式碼來開發 Systems Manager 自動化工作流程簿，將快照檔案存檔到 Amazon S3 和 Amazon S3 Glacier。	雲端架構師

Backup 選項 3 — 建立 Amazon EBS 磁碟區快照

任務	描述	所需技能
安裝字符-SSP 模擬器。	在 Amazon EC2 Linux 執行個體上安裝字元 SSP 模擬器。 有關這方面的詳細資訊，請參閱 Stromasys 文件中的為 Charon-SSP 設定 AWS 雲端執行個體。	DevOps 工程師
為 Sun SAPC 客體伺服器建立 EBS 磁碟區。	登入 AWS 管理主控台，開啟 Amazon EBS 主控台，然後為 Sun SAPC 客體伺服器建立 EBS 磁碟區。 有關這方面的詳細資訊，請參閱 Stromasys 文件中的為 Charon-SSP 設定 AWS 雲端執行個體。	雲端架構師
將 EBS 磁碟區連接至 Amazon EC2 Linux 執行個體。	在 Amazon EC2 主控台上，將 EBS 磁碟區連接到 Amazon EC2 Linux 執行個體。 如需這方面的詳細資訊，請參閱 Amazon Amazon EC2 文件	AWS DevOps

任務	描述	所需技能
	中的將 Amazon EBS 磁碟區連接到執行個體。	
在字元 SSP 模擬器中將 EBS 磁碟區對應為 SCSI 磁碟機。	設定字元 SSP 管理員，將 EBS 磁碟區對應為 Sun SPARC 客體伺服器中的 SCSI 磁碟機。 如需這方面的詳細資訊，請參閱 Stromasys 文件中的《Linux 專用字元-SSP V5.2 版》指南的 SCSI 儲存設定一節。	AWS DevOps
設定 AWS Backup 排程以快照擷取 EBS 磁碟區。	設定 AWS Backup 政策和排程以快照 EBS 磁碟區。 如需有關這方面的詳細資訊，請參閱 AWS 開發人員中心文件中的 Amazon EBS 使用 AWS Backup 和還原教學課程。	AWS DevOps

Backup 選項 4 — 建立 AWS Storage Gateway VTL

任務	描述	所需技能
建立磁帶閘道裝置。	登入 AWS 管理主控台，開啟 AWS Storage Gateway 主控台，然後在 VPC 中建立磁帶閘道裝置。 如需詳細資訊，請參閱 AWS Storage Gateway 文件中的建立閘道。	雲端架構師

任務	描述	所需技能
建立一個 Amazon RDS 資料庫執行個體以供參考目錄使用。	開啟 Amazon RDS 主控台並建立一個 Amazon RDS for MySQL 的資料庫執行個體。 如需這方面的詳細資訊，請參閱 Amazon RDS 說明文件中的建立 MySQL 資料庫執行個體並連接到 MySQL 資料庫執行個體上的資料庫。	雲端架構師
在 VPC 中部署備份應用程式控制器。	在 EC2 執行個體上安裝 Bacula、部署備份應用程式控制器，然後將備份儲存設定為與磁帶閘道裝置連接。您可以在檔案中使用樣本 Bacula Director 儲存守護程`Bacula-storage-daemon-config.txt` 式組態 (隨附)。 有關這方面的更多信息，請參閱 Bacula 文檔。	AWS DevOps
在 Sun SPARC 客體伺服器上設定備份應用程式。	使用`SUN-SPARC-Guest-Bacula-Config.txt` 檔案中的範例 Bacula 設定 (隨附)，設定第二個用戶端以在 Sun SPARC 客體伺服器上安裝和設定備份應用程式。	DevOps 工程師

任務	描述	所需技能
設定備份組態和排程。	使用`Bacula-Directory-Config.txt` 檔案中的樣本 Bacula Director 配置（隨附），在備份應用程式控制器中設定備份配置和排程。 有關這方面的更多信息，請參閱 Bacula 文檔。	DevOps 工程師
驗證備份組態和排程是否正確。	遵循 Bacula 文件中的指示，為您在 Sun SPARC 客體伺服器中的設定執行驗證和備份測試。 例如，您可以使用下列指令來驗證組態檔案： • `bacula-dir -t -c bacula-dir.conf` • `bacula-fd -t -c bacula-fd.conf` • `bacula-sd -t -c bacula-sd.conf`	DevOps 工程師

相關資源

- 具備 VE 授權的夏龍虛擬 SPARC
- 虛擬夏龍
- 透過 Bacula 企業版使用雲端服務和物件儲存
- 災難復原 (DR) 目標
- Charon 傳統系統模擬解決方案

其他資訊

Backup 選項 1 — 建立 Stromasys 虛擬磁帶

您可以使用下列範例 Systems Manager 自動化 Runbook 程式碼來自動啟動備份，然後交換磁帶：

```
...
# example backup script saved in SUN SPARC Server
 #!/usr/bin/bash
 mt -f  rewind
 tar -cvf
 mt -f  offline
...
         mainSteps:
         - action: aws:runShellScript
           name:
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               # Validate tape backup container file exists
               if [ ! -f {{TapeBackupContainerFile}} ]; then
                 logger -s -p local3.warning "Tape backup container file is not exists
 - {{TapeBackupContainerFile}}, create a new one"
                 touch {{TapeBackupContainerFile}}
               fi
         - action: aws:runShellScript
           name: startBackup
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               user={{BACKUP_USER}}
               keypair={{KEYPAIR_PATH}}
               server={{SUN_SPARC_IP}}
               backup_script={{BACKUP_SCRIPT}}
               ssh -i $keypair $user@$server -c "/usr/bin/bash $backup_script"
         - action: aws:runShellScript
           name: swapVirtualDiskContainer
           inputs:
             onFailure: Abort
```

```
              timeoutSeconds: "1200"
              runCommand:
              - |
                mv {{TapeBackupContainerFile}} {{TapeBackupContainerFile}}.$(date +%s)
                touch {{TapeBackupContainerFile}}
          - action: aws:runShellScript
            name: uploadBackupArchiveToS3
            inputs:
              onFailure: Abort
              timeoutSeconds: "1200"
              runCommand:
              - |
                aws s3 cp {{TapeBackupContainerFile}} s3://{{BACKUP_BUCKET}}/
{{SUN_SPARC_IP}}/$(date '+%Y-%m-%d')/
 ...
```

Backup 選項 2 — 快照

您可以使用下列範例 Systems Manager 自動化 Runbook 程式碼來自動化備份程序：

```
      ...

         mainSteps:
         - action: aws:runShellScript
           name: startSnapshot
           inputs:
             onFailure: Abort
             timeoutSeconds: "1200"
             runCommand:
             - |
               # You may consider some graceful stop of the application before taking a
 snapshot
               # Query SSP PID by configuration file
               # Example: ps ax | grep ssp-4 | grep Solaris10.cfg | awk '{print $1"
 "$5}' | grep ssp4 | cut -f1 -d" "
               pid=`ps ax | grep ssp-4 | grep {{SSP_GUEST_CONFIG_FILE}} | awk '{print
 $1" "$5}' | grep ssp4 | cut -f1 -d" "`
               if [ -n "${pid}" ]; then
                 kill -SIGTSTP ${pid}
               else
                 echo "No PID found for SPARC guest with config
 {{SSP_GUEST_CONFIG_FILE}}"
                 exit 1
               fi
```

```
          - action: aws:runShellScript
            name: startBackup
            inputs:
              onFailure: Abort
              timeoutSeconds: "1200"
              runCommand:
              - |
                # upload snapshot and virtual disk files into S3
                aws s3 sync {{SNAPSHOT_FOLDER}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-
%d')/
                aws s3 cp {{VIRTUAL_DISK_FILE}} s3://{{BACKUP_BUCKET}}/$(date '+%Y-%m-
%d')/
          - action: aws:runShellScript
            name: restratSPARCGuest
            inputs:
              onFailure: Abort
              timeoutSeconds: "1200"
              runCommand:
              - |
                /opt/charon-ssp/ssp-4u/ssp4u -f {{SSP_GUEST_CONFIG_FILE}} -d -a
 {{SPARC_GUEST_NAME}} --snapshot {{SNAPSHOT_FOLDER}}
 ...
```

Backup 選項 4 — AWS Storage Gateway VTL

如果您使用 Solaris 非全域區域來執行虛擬化的舊版 Sun SPARC 伺服器，則備份應用程式方法可套用至在 Sun SPARC 伺服器中執行的非全域區域 (例如，備份用戶端可以在非全域區域內執行)。不過，備份用戶端也可以在 Solaris 主機中執行，並建立非全域區域的快照。然後可以將快照備份到磁帶上。

下列範例組態會將裝載 Solaris 非全域區域的檔案系統新增至 Solaris 主機的備份組態中：

```
FileSet {
   Name = "Branded Zones"
   Include {
     Options {
       signature = MD5
     }
     File = /zones
   }
 }
```

附件

若要存取與此文件相關聯的其他內容，請解壓縮下列檔案：attachment.zip

使用 Veeam Backup 和複寫將資料備份並存檔到 Amazon S3

由珍娜·詹姆斯，安東尼·菲奧雷（AWS）和威廉·奎格利創建

環境：生產	技術：儲存與備份	AWS 服務：Amazon EC2; Amazon S3; Amazon S3 冰川

Summary

此模式詳細說明使用 Veeam 向外擴充 Backup 儲存庫功能，將 Veeam 備份和複寫建立的備份傳送到支援的 Amazon 簡單儲存服務 (Amazon S3) 物件儲存類別的程序。

Veeam 支援多種 Amazon S3 儲存類別，以最符合您的特定需求。您可以根據備份或封存資料的資料存取、恢復能力和成本需求來選擇儲存類型。例如，您可以將不打算使用 30 天或更長時間的資料存放在 Amazon S3 不常存取 (IA) 中，以降低成本。如果您打算將資料存檔 90 天或更長時間，可以使用 Amazon Simple Storage Service Glacier (Amazon S3 Glacier) 彈性擷取或 S3 Glacier Deep Archive 搭配 Veeam 的存檔層。您也可以使用 S3 物件鎖定在 Amazon S3 中使備份不可變。

此模式不涵蓋如何使用 AWS Storage Gateway 中的磁帶閘道設定 Veeam Backup 和複寫。如需該主題的相關資訊，請參閱 V eeam 網站上的使用 AWS VTL 閘道進行 Backup 和複寫 — 部署指南。

警告：此案例需要具有程式設計存取權限和長期登入資料的 IAM 使用者，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。如有必要，可更新存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的更新存取金鑰。

先決條件和限制

先決條件

- 已安裝 Veeam Backup 與複寫功能，包括 Veeam 可用性套件或 Veeam Backup 基本資訊 (您可以註冊免費試用)
- Veeam Backup 和複寫授權，具有企業版或超強企業版功能，其中包括 Veeam 通用授權 (VUL)
- 可存取 Amazon S3 儲存貯體的有效 AWS Identity and Access Management (IAM) 使用者

- 可存取亞馬遜彈性運算雲端 (Amazon EC2) 和 Amazon Virtual Private Cloud (Amazon VPC) 的有效 IAM 使用者 (如果使用存檔層)
- 從現場部署到 AWS 服務的網路連線，透過公用網際網路連線或 AWS Direct Connect 公有虛擬界面 (VIF) 提供備份和還原流量的可用頻寬
- 已開啟下列網路連接埠和端點，以確保與物件儲存庫正確通訊：
  - Amazon S3 儲存 — TCP — 連接埠 443：用於與 Amazon S3 儲存通訊。
  - Amazon S3 儲存 — 雲端端點 — 適用於 AWS 區域和 AWS GovCloud (美國) 區域的 *.amazonaws.com，或適用於中國區域的 *.amazonaws.com.cn：用於與 Amazon S3 儲存進行通訊。如需連線端點的完整清單，請參閱 AWS 文件中的 Amazon S3 端點。
  - Amazon S3 儲存 — TCP HTTP — 連接埠 80：用於驗證憑證狀態。請考慮憑證驗證端點 (憑證撤銷清單 (CRL) URL 和線上憑證狀態通訊協定 (OCSP) 伺服器 — 可能會變更。實際的位址清單可以在憑證本身中找到。
  - Amazon S3 儲存 — 憑證驗證端點 — *.amazontrust.com：用於驗證憑證狀態。請考慮憑證驗證端點 (CRL URL 和 OCSP 伺服器) 可能會變更。實際的位址清單可以在憑證本身中找到。

限制

- Veeam 不支援任何用作 Veeam 物件儲存庫的 S3 儲存貯體上的 S3 生命週期政策。其中包括 Amazon S3 儲存類別轉換和 S3 生命週期到期規則的政策。Veeam 必須是管理這些物件的唯一實體。啟用 S3 生命週期政策可能會產生意外的結果，包括資料遺失。

產品版本

- Veeam Backup 與複寫 v9.5 更新 4 或更新版本 (僅備份或容量層)
- Veeam Backup 與複寫 v10 或更新版本 (備份或容量層和 S3 物件鎖定)
- Veeam Backup 與複寫第 11 版或更新版本 (備份或容量層、存檔或存檔層，以及 S3 物件鎖定)
- Veeam Backup 與複寫 v12 或更新版本 (效能層、備份或容量層、封存或存檔層，以及 S3 物件鎖定)
- S3 Standard
- S3 標準 – IA
- S3 單區域 – IA
- S3 冰川彈性擷取 (僅限 v11 及更新版本)
- S3 Glacier Deep Archive (僅限 v11 及更新版本)
- S3 冰川即時擷取 (僅限 v12 及更新版本)

架構

源, 技術, 堆棧

- 透過 Veeam Backup 伺服器或 Veeam 閘道伺服器連線到 Amazon S3 的現場部署 Veeam 備份和複寫安裝

目標技術堆疊

- Amazon S3
- Amazon VPC 和 Amazon EC2（如果使用存檔層）

目標體系結構：SOBR

下圖顯示向外延展備份儲存庫 (SOBR) 架構。

Veeam Backup 和複製軟體可保護資料，避免系統故障、應用程式錯誤或意外刪除等邏輯錯誤。在此圖中，備份會先在內部部署執行，而次要副本會直接傳送至 Amazon S3。備份代表資料的 point-in-time 副本。

工作流程包含分層或將備份複製到 Amazon S3 所需的三個主要元件，以及一個選用元件：

- Veeam Backup 與複製 (1) — 負責協調、控制和管理備份基礎結構、設定、工作、復原工作和其他程序的備份伺服器。
- Veeam 閘道伺服器 (未顯示在圖表中) — 選用的現場部署閘道伺服器，如果 Veeam 備份伺服器沒有 Amazon S3 的對外連線，則需要此伺服器。
- 向外延展備份儲存庫 (2) — 支援多層資料儲存的水平擴展的儲存庫系統。橫向擴充備份儲存庫由一或多個備份儲存庫組成，可快速存取資料，並且可以使用 Amazon S3 物件儲存儲庫進行擴充，用於長期儲存 (容量層) 和存檔 (存檔層)。Veeam 使用向外延展備份儲存庫，在本機 (效能層) 和 Amazon S3 物件儲存 (容量和存檔層) 之間自動分層資料。
- Amazon S3 (3) — 提供可擴展性、資料可用性、安全性和效能的 AWS 物件儲存服務。

目標架構：DTO

下圖顯示 direct-to-object (DTO) 架構。

在此圖中，備份資料直接傳送到 Amazon S3，而不會先存放在現場部署。次要副本可以存放在 S3 冰川中。

自動化和規模

您可以使用存放庫中提供的 AWS CloudFormation 範本，自動化 IAM 資源和 S3 VeeamHub GitHub 儲存貯體的建立。範本包括標準和不可變選項。

工具

工具和 AWS 服務

- Veeam Backup 和複寫是 Veeam 的解決方案，用於保護、備份、複製和還原您的虛擬和實體工作負載。
- AWS 可 CloudFormation協助您建立 AWS 資源的模型和設定、快速且一致地佈建它們，並在整個生命週期中進行管理。您可以使用範本來描述您的資源及其相依性，並將它們一起啟動並設定為堆疊，而不是個別管理資源。您可以跨多個 AWS 帳戶和 AWS 區域管理和佈建堆疊。
- 亞馬遜彈性運算雲 (Amazon EC2) 在 AWS 雲端提供可擴展的運算容量。您可以使用 Amazon EC2 根據需要啟動任意數量或少量的虛擬伺服器，並且可以向外擴展或擴展。
- AWS Identity and Access Management (IAM) 是一種用於安全控制 AWS 服務存取的 Web 服務。透過 IAM，您可以集中管理使用者、存取金鑰等安全登入資料，以及控制使用者和應用程式可存取的 AWS 資源的許可。
- Amazon Simple Storage Service (Amazon S3) 是一種對象存儲服務。您可以使用 Amazon S3 隨時從 Web 任何地方存放和擷取任意資料量。
- Amazon S3 Glacier (S3 Glacier) 是一種安全耐用的服務，適用於低成本的資料存檔和長期備份。
- Amazon Virtual Private Cloud (Amazon VPC) 佈建 AWS 雲端的邏輯隔離部分，您可以在已定義的虛擬網路中啟動 AWS 資源。這個虛擬網路與您在資料中心中操作的傳統網路非常相似，且具備使用 AWS 可擴展基礎設施的優勢。

Code

使用VeeamHub GitHub 儲存庫中提供的 CloudFormation 範本，為此模式自動建立 IAM 資源和 S3 儲存貯體。如果您想要手動建立這些資源，請按照 Epics 一節中的步驟操作。

最佳實務

- 根據 IAM 最佳實務，我們強烈建議您定期輪換長期 IAM 使用者登入資料，例如用於將 Veeam Backup 和複寫備份寫入 Amazon S3 的 IAM 使用者。如需詳細資訊，請參閱 IAM 文件中的安全最佳實務。

史诗

在您的帳戶中設定 Amazon S3 儲存

任務	描述	所需技能
建立 IAM 使用者。	依照 IAM 文件中的指示建立 IAM 使用者。此使用者不應具備 AWS 主控台存取權，而且您需要為此使用者建立存取金鑰。Veeam 使用此實體向 AWS 進行驗證，以便讀取和寫入 S3 儲存貯體。您必須授與最低權限 (也就是說，僅授與執行工作所需的權限)，這樣使用者就沒有超過所需的權限。如需附加至 Veeam IAM 使用者的 IAM 政策範例，請參閱其他資訊一節。 注意或者，您可以使用VeeamHub GitHub 存放庫中提供的 CloudFormation 範本為此模式建立 IAM 使用者和 S3 儲存貯體。	AWS 管理員
建立 S3 儲存貯體。	1. 登入 AWS 管理主控台，然後前往 https://console.aws.amazon.com/s3/ 開啟 Amazon S3 主控台。	AWS 管理員

任務	描述	所需技能
	2. 如果您尚未將現有的 S3 儲存貯體用作目標儲存體，請選擇「建立儲存貯體」，然後指定儲存貯體名稱、AWS 區域和儲存貯體設定。 • 建議您為 S3 儲存貯體啟用封鎖公用存取選項，並設定存取和使用者權限政策以符合組織的需求。如需範例，請參閱 Amazon S3 文件。 • 我們建議您啟用 S3 物件鎖定，即使您不打算立即使用它也是如此。此設定只能在建立 S3 儲存貯體時啟用。 如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。	

將 Amazon S3 和 S3 冰川彈性擷取 (或 S3 Glacier Deep Archive) 新增至 Veeam Backup 和複寫

任務	描述	所需技能
啟動「新增物件儲存庫」精靈。	在 Veeam 中設定物件儲存和向外擴充備份儲存庫之前，您必須新增要用於容量和存檔層的 Amazon S3 和 Amazon S3 Glacier 儲存儲存庫。在下一個史詩中，您會將這些儲存儲存庫連接到您的向外延展備份存放庫。	AWS 管理員、應用程式擁有

任務	描述	所需技能
	1. 在 Veeam 主控台上，開啟「Backup 基礎結構」檢視。 2. 在詳細目錄窗格中，選擇 [Backup 儲存區域] 節點，然後選擇 [新增儲存區域]。 3. 在「新增 Backup 儲存庫」對話方塊中，選擇「物件儲存」、「Amazon S3」。	

任務	描述	所需技能
為容量層新增 Amazon S3 儲存。	1. 在亞馬遜雲端儲存服務對話方塊中，選擇 Amazon S3。 2. 在精靈的 [名稱] 步驟中，指定物件儲存體名稱和簡短描述，例如建立者和建立日期。 3. 在精靈的 [帳戶] 步驟中，指定物件儲存體帳戶。 • 對於登入資料，請選擇您在第一個史詩中建立的 IAM 使用者來存取 Amazon S3 物件儲存。 • 對於 AWS 區域，請選擇 Amazon S3 儲存貯體所在的 AWS 區域。 4. 在精靈的 [值區] 步驟中，指定物件儲存設定。 • 對於資料中心區域，請選擇 Amazon S3 儲存貯體所在的 AWS 區域。 • 對於儲存貯體，請選擇您在第一個史詩中建立的 S3 儲存貯體。 • 在「資料夾」中，建立或選取要對映物件儲存庫的雲端資料夾。 • 如果您要啟用不變性，請選擇讓最近的備份在 X 天內不可變，並設定應鎖定備份的期間。請注意，由於從 Veeam 向 Amazon S3 呼叫的 API 數量增	AWS 管理員、應用程式擁有

任務	描述	所需技能
	加，因此啟用不變性會導致成本增加。 5. 在精靈的 [摘要] 步驟中，檢閱組態資訊，然後選擇 [完成]。	

任務	描述	所需技能
為存檔層新增 S3 冰川儲存。	如果您想要建立封存層，請使用其他資訊一節中詳述的 IAM 許可。 1. 如前所述，啟動「新增物件儲存庫」精靈。 2. 在 Amazon 雲端儲存服務對話方塊中，選擇 Amazon S3 冰川。 3. 在精靈的 [名稱] 步驟中，指定物件儲存體名稱和簡短描述，例如建立者和建立日期。 4. 在精靈的 [帳戶] 步驟中，指定物件儲存體帳戶。 • 對於登入資料，請選擇您在第一個史詩中建立的 IAM 使用者來存取 Amazon S3 Glacier 物件儲存。 • 對於 AWS 區域，請選擇 Amazon S3 儲存貯體所在的 AWS 區域。 5. 在精靈的 [值區] 步驟中，指定物件儲存設定。 • 對於資料中心區域，請選擇 AWS 區域。 • 對於儲存貯體，請選擇 S3 儲存貯體來存放備份資料。這可以是您用於容量層的相同儲存貯體。	AWS 管理員、應用程式擁有

任務	描述	所需技能
	• 在「資料夾」中，建立或選取要對映物件儲存庫的雲端資料夾。 • 如果您要啟用不變性，請選擇 [讓最近的備份在其保留原則的整個期間不可變]。請注意，由於從 Veeam 向 Amazon S3 呼叫的 API 數量增加，因此啟用不變性會導致成本增加。 • 如果您想要使用 S3 Glacier 深層存檔做為封存儲存類別，請選擇使用深度歸檔儲存類別。 6. 在精靈的代理設備步驟中，設定用於將資料從 Amazon S3 傳輸到 Amazon S3 Glacier 的輔助執行個體。您可以使用預設設定或手動設定每個設定。若要手動進行設定： • 請選擇 Customize (自訂)。 • 對於 EC2 執行個體類型，請根據您將備份檔案傳輸到水平擴充備份存放庫的存檔層的速度和成本需求，選擇 Proxy 應用裝置的執行個體類型。 • 對於 Amazon VPC，請選擇目標執行個體的 VPC。	

任務	描述	所需技能
	• 對於子網路，選擇 Proxy 應用裝置的子網路。 • 對於安全性群組，選擇要與 Proxy 應用裝置建立關聯的安全群組。 • 對於重新導向器連接埠，請指定 TCP 連接埠，以便在 Proxy 應用裝置和備份基礎結構元件之間路由要求。 • 選擇 「確定」 以確認您的設定。 7. 在精靈的 [摘要] 步驟中，檢閱組態資訊，然後選擇 [完成]。	

新增向外延展備份儲存庫

任務	描述	所需技能
啟動「新增向外延展 Backup 儲存區域」精靈。	1. 在 Veeam 主控台上，開啟「Backup 基礎結構」檢視。 2. 在詳細目錄窗格中，選擇向外延展儲存庫，然後選擇新增向外延展儲存庫。	應用程式擁有者、AWS 系統管理
新增向外延展備份儲存庫，並設定容量和封存層。	1. 在精靈的「名稱」步驟中，指定向外延展備份儲存區域的名稱和簡短說明。 2. 如果需要，請加入效能範圍。您也可以使用現有的 Veeam 本機備份儲存庫做	應用程式擁有者、AWS 系統管理

任務	描述	所需技能
	為效能層。 從 Veeam 版本 12 開始，您可以將 S3 儲存貯體新增為 direct-to-object (DTO) 備份的效能範圍，略過本機效能層。 3. 選擇進階，並指定向外延展備份儲存區域的其他選項。 • 選擇使用每台機器的備份檔案，為每台機器建立個別的備份檔案，並同時將這些檔案寫入多個串流中的備份儲存庫。建議使用此選項以獲得更好的儲存和計算資源使用率。 • 選擇「當需要範圍離線時執行完整備份」，以便在包含增量備份的還原點的範圍離線時建立完整備份檔案。此選項需要向外延展備份儲存庫中的可用空間，以託管完整備份檔案。 4. 在精靈的「原則」步驟中，指定儲存區域的備份放置原則。 • 選擇 [資料位置]，將屬於相同鏈結的完整備份和增量備份檔案儲存在相同的效能範圍內。您可以將屬於新備份鏈結的檔案儲存至相同的效能範圍或另一個備份鏈結 (除非您使用重複資料刪除儲存裝置作為效能範圍)。	

任務	描述	所需技能
	• 選擇效能，將完整備份和增量備份檔案儲存至不同的效能範圍。此選項需要快速可靠的網路連線。如果您選擇「效能」，您可以限制要在每個效能範圍上儲存的備份檔案類型。例如，您可以將完整備份檔案儲存在一個範圍上，並將增量備份檔案儲存在其他範圍上。若要選擇檔案類型： • 請選擇 Customize (自訂)。 • 在「Backup 放置設定」對話方塊中，選擇效能範圍，然後選擇「編輯」。 • 選擇要存儲在該範圍上的備份文件的類型。 5. 在精靈的 [容量層] 步驟中，設定要附加至向外延展備份儲存區域的長期儲存層。 • 選擇使用物件儲存擴充備份儲存庫容量。對於物件儲存庫，請為您在上一篇史詩中新增的容量層選擇 Amazon S3 儲存。 • 選擇「時段」以選取用於移動或複製資料的時間範圍。 • 選擇建立備份後立即將備份複製到物件儲存體，以	

任務	描述	所需技能
	將所有或僅最近建立的備份檔案複製到容量範圍。 • 選擇將備份移至物件儲存體，當備份離開作業還原時間逾時，將非作用中的備份鏈傳輸到容量範圍。在 [移動超過 X 天的備份檔案] 欄位中，指定應卸載備份檔案的持續時間。(若要在建立非作用中備份鏈結當天卸載，請指定 0 天。) 如果向外延展備份儲存區域達到您指定的臨界值，您也可以選擇覆寫，以便更快地移動備份檔案。 • 選擇加密上傳至物件儲存體的資料，然後指定密碼來加密所有資料及其中繼資料以進行卸載。選擇「新增」或「管理密碼」以指定新密碼。 6. 在精靈的 [封存層] 步驟中，設定要附加至向外延展備份儲存區域的封存儲存層。(如果您略過新增 Amazon S3 冰川儲存，則不會顯示此步驟。) • 選擇將 GFS 完整備份封存至物件儲存體。對於物件儲存儲存庫，請選擇您在上一篇史詩中新增的 Amazon S3 Glacier 儲存。	

任務	描述	所需技能
	• 對於封存超過 N 天的 GFS 備份，請選擇將檔案移至封存範圍的時間範圍。若要在建立非作用中備份鏈結當天封存，請指定 0 天。） 7. 在精靈的 [摘要] 步驟中，複查向外延展備份儲存區域的組態，然後選擇 [完成]。	

相關資源

- 在您的 AWS 帳戶中建立 IAM 使用者 (IAM 文件)
- 建立儲存貯體 (Amazon S3 文件)
- 封鎖對您的 Amazon S3 儲存的公開存取 (Amazon S3 文件)
- 使用 S3 物件鎖定 (Amazon S3 文件)
- 維安技術文件
- 如何建立連線到 S3 物件儲存的安全 IAM 政策 (Veeam 文件)

其他資訊

以下各節提供在此模式的「史詩」區段中建立 IAM 使用者時可使用的 IAM 政策範例。

容量方案的 IAM 政策

注意：<yourbucketname>將範例政策中 S3 儲存貯體的名稱從變更為您要用於 Veeam 容量層備份的 S3 儲存貯體的名稱。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetObjectVersion",
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:PutObjectLegalHold",
                "s3:GetBucketVersioning",
                "s3:GetObjectLegalHold",
                "s3:GetBucketObjectLockConfiguration",
                "s3:PutObject*",
                "s3:GetObject*",
                "s3:GetEncryptionConfiguration",
                "s3:PutObjectRetention",
                "s3:PutBucketObjectLockConfiguration",
                "s3:DeleteObject*",
                "s3:DeleteObjectVersion",
                "s3:GetBucketLocation"

            ],
            "Resource": [
                "arn:aws:s3:::/*",
                "arn:aws:s3:::"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket"
            ],
            "Resource": "*"
        }
    ]
}
```

封存層的 IAM 政策

注意：<yourbucketname>將範例政策中 S3 儲存貯體的名稱從變更為您要用於 Veeam 存檔層備份的 S3 儲存貯體的名稱。

若要使用現有的 VPC、子網路和安全群組：

```
  {
    "Version": "2012-10-17",
```

```
    "Statement": [
      {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
          "s3:DeleteObject",
          "s3:PutObject",
          "s3:GetObject",
          "s3:RestoreObject",
          "s3:ListBucket",
          "s3:AbortMultipartUpload",
          "s3:GetBucketVersioning",
          "s3:ListAllMyBuckets",
          "s3:GetBucketLocation",
          "s3:GetBucketObjectLockConfiguration",
          "s3:PutObjectRetention",
          "s3:GetObjectVersion",
          "s3:PutObjectLegalHold",
          "s3:GetObjectRetention",
          "s3:DeleteObjectVersion",
          "s3:ListBucketVersions",
          "ec2:DescribeInstances",
          "ec2:CreateKeyPair",
          "ec2:DescribeKeyPairs",
          "ec2:RunInstances",
          "ec2:DeleteKeyPair",
          "ec2:DescribeVpcAttribute",
          "ec2:CreateTags",
          "ec2:DescribeSubnets",
          "ec2:TerminateInstances",
          "ec2:DescribeSecurityGroups",
          "ec2:DescribeImages",
          "ec2:DescribeVpcs"
        ],
        "Resource": "*"
      }
    ]
  }
```

若要建立新的 VPC、子網路和安全性群組：

```
  {
    "Version": "2012-10-17",
```

```
    "Statement": [
      {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
          "s3:DeleteObject",
          "s3:PutObject",
          "s3:GetObject",
          "s3:RestoreObject",
          "s3:ListBucket",
          "s3:AbortMultipartUpload",
          "s3:GetBucketVersioning",
          "s3:ListAllMyBuckets",
          "s3:GetBucketLocation",
          "s3:GetBucketObjectLockConfiguration",
          "s3:PutObjectRetention",
          "s3:GetObjectVersion",
          "s3:PutObjectLegalHold",
          "s3:GetObjectRetention",
          "s3:DeleteObjectVersion",
          "s3:ListBucketVersions",
          "ec2:DescribeInstances",
          "ec2:CreateKeyPair",
          "ec2:DescribeKeyPairs",
          "ec2:RunInstances",
          "ec2:DeleteKeyPair",
          "ec2:DescribeVpcAttribute",
          "ec2:CreateTags",
          "ec2:DescribeSubnets",
          "ec2:TerminateInstances",
          "ec2:DescribeSecurityGroups",
          "ec2:DescribeImages",
          "ec2:DescribeVpcs",
          "ec2:CreateVpc",
          "ec2:CreateSubnet",
          "ec2:DescribeAvailabilityZones",
          "ec2:CreateRoute",
          "ec2:CreateInternetGateway",
          "ec2:AttachInternetGateway",
          "ec2:ModifyVpcAttribute",
          "ec2:CreateSecurityGroup",
          "ec2:DeleteSecurityGroup",
          "ec2:AuthorizeSecurityGroupIngress",
          "ec2:AuthorizeSecurityGroupEgress",
```

```
          "ec2:DescribeRouteTables",
          "ec2:DescribeInstanceTypes"
        ],
        "Resource": "*"
      }
    ]
  }
```

在 AWS 上設定適 NetBackup 用於 VMware 雲端的雲端

創建者：沙伯姆薩拉尼 (AWS)

環境：生產	技術：儲存與備份；雲端原生	工作負載：所有其他工作
AWS 服務：Amazon S3；AWS Transit Gateway；Amazon VPC；Amazon EBS		

Summary

注意：自 2024 年 4 月 30 日起，AWS 或其通路合作夥伴不再轉售 VMware 雲端服務。該服務將繼續通過博通提供。我們建議您聯絡 AWS 代表以取得詳細資訊。

許多企業使用 Veritas NetBackup 做為其內部部署 VMware vSphere 工作負載的備份與復原解決方案。一旦企業將工作負載遷移到 VMware 雲端 Amazon Web Services (AWS) 基礎設施中的軟體定義資料中心 (SDDC)，就沒有明確的整合 lift-and-shift 程序。 NetBackup此模式說明如何在 AWS 帳戶 NetBackup 中設定 Veritas，並將其設定為備份 VMware 軟體定義式定義式定義的資料中心中的工作負載。

此模式不包含移轉工作負載的指示。如需詳細資訊，請參閱使用 VMware HCX 將 VMware 軟體定義的資料定義中心移轉至 AWS 上的 VMware 雲端。將工作負載設定到 VMware Cloud on AWS 時，請使用延伸叢集 (VMware 說明文件)。在此組態中，您的叢集跨越單一區域內的兩個 AWS 可用區域。這可在其中一個可用區域無法使用的情況下提供高可用性和復原能力。彈性 DRS 和 vSAN 見證主機 (VMware 說明文件) 可順暢地將資料複製到第三個可用區域 (稱為容錯網域)。此同位檢查解決方案可協助您在發生故障時復原資料。由於此方法需要三個可用區域，因此在為 VMware Cloud 環境選取 AWS 區域時，請確定其具有三個或更多可用區域。如需更多詳細資訊，請參閱區域和可用區域。

在這種模式中，每個 SDDC 都有一個備份主機，這是一個代理服務器。使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，您可以在單獨的虛擬私有雲端 (VPC) 中設定 NetBackup 主伺服器和媒體伺服器，每個軟體定義的資料中心各一個。由於彈性網路介面提供高頻寬和低延遲，因此您可以使用它們來設定備份主機及其對應 NetBackup 的主要伺服器和媒體伺服器之間的連線。EC2 執行個體會將備份導向至亞馬遜彈性區塊存放區 (Amazon EBS) 磁碟區，這是備份的第一個點。您可以使用 AWS 讓軟體定義的資料中心的 EBS 磁碟區保持同 DataSync 步。

您也可以使用 AWS Transit Gateway 和界面 VPC 端點將 EBS 磁碟區連接到另一個儲存服務，例如 Amazon Simple Storage Service (Amazon S3)。根據您的保留政策，您可以使用 S3 智慧型分層 S3 Glacier 儲存類別來最佳化儲存成本。如需詳細資訊，請參閱使用 Amazon S3 儲存類別 (Amazon S3 文件)。

先決條件和限制

先決條件

- 您的 VMware Cloud on AWS 環境使用跨越兩個可用區域的延伸叢集。
- 備份主機必須位於 AWS 軟體定義的 VMware 雲端上，該資料存放區可存取部署 VMware 虛擬機器磁碟檔案 (VMDK) 檔案的資料存放區。
- HotAdd 必須在用 NetBackup 戶端上啟用傳輸模式才能備份和還原虛擬機器 (VM)，而且必須允許從使用者導向的檔案和資料夾還原。

限制

- 主 NetBackup 要伺服器必須針對 SDDC 中 vCenter 備份主機使用私人 IP 位址的 DNS 解析。
- 主 NetBackup 要伺服器和備份主機上的主機檔案應包含下列項目：
  - 主要伺服器的私人 IP 位址和私人 DNS 名稱
  - 備份主機的私人 IP 位址和私人 DNS 名稱
- 如果您要設定 S3 儲存貯體的介面 VPC 端點，SDDC 運算閘道防火牆必須設定為允許來自無類別網域間路由 (CIDR) 區塊來源的 HTTPS。如需詳細資訊，請參閱使用 S3 端點存取 S3 儲存貯體 (VMware 說明文件)。
- VMware Cloud on AWS 不支援下列功能 NetBackup：
  - 備份或還原虛擬機器範本
  - 使用 NetBackup vSphere 用戶端 (HTML5 外掛程式)
  - 鎖定和解除鎖定 VM 以進行備份或還原
  - 備份無法儲存在 vSAN 資料存放區
  - 網路區塊裝置 (NBD)、次網路 SSL 和 SAN 傳輸模式

產品版本

- VMware 雲端軟體定義的資料中心 1.0 版或更新版本
- NetBackup 版本 8.1.2 或更新版本

- 版本 6.8 或更新版本
- VMware 6.0 vSphere 或更新版本

架構

下圖顯示了在 AWS 上 NetBackup 的 VMware 雲端的組態。主 NetBackup 伺服器和媒體伺服器部署在個別的 VPC 中，並透過彈性網路介面連接至 SDDC 中的備份主機。 NetBackup 主伺服器和媒體伺服器會將備份存放在 Amazon EBS 磁碟區中。您可以選擇使用 AWS 傳輸閘道和 AWS PrivateLink 界面 VPC 端點，在 Amazon S3 儲存貯體中設定其他儲存。

工具

AWS 服務和工具

- 亞馬遜彈性區塊存放區 (Amazon EBS) 提供區塊層級儲存磁碟區，可與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。
- AWS 可 PrivateLink協助您建立從虛擬私有雲端 (VPC) 到 VPC 以外的服務的單向私有連線。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他服務

- VMware 雲端服務是由 Amazon Web Services (AWS) 和 VMware 共同開發的整合式雲端產品。
- NetBackup 用於 VMware 備份和還原在 VMware ESXi 主機上執行的 VMware 虛擬機器。

史诗

設定伺 NetBackup 服器

任務	描述	所需技能
更新防火牆規則。	更新防火牆規則，以在 AWS 軟體定義的資料中心上的 VMware 雲端與 NetBackup 主伺服器和媒體伺服器之間建立連線。請執行下列操作： 1. 在 AWS 上登入 VMware 雲端 https://vmc.vmware.com/ 2. 在「網路和安全性」標籤上，選擇「閘道防火牆」。 3. 在 [閘道防火牆] 頁面上，選擇 [計算閘道]。 4. 選擇 [新增規則]，然後使用必要的防火牆連接埠設定建立新規則。如需詳細資訊，請參閱NetBackup 防火牆連接埠需求 (Veritas 說明文件)。	網路管理員、雲端管理員
啟動 NetBackup 主伺服器和媒體伺服器。	1. 登入 AWS 管理主控台，然後開啟 Amazon EC2 主控台，網址為 https://console.aws.amazon.com/ec2/ 2. 啟動 EC2 執行個體 (Amazon EC2 文件)，並使用下列組態詳細資訊： a. 對於 NetBackup 主伺服器和媒體伺服器，請選取 NBU-	雲端管理員、Backup 管理員

任務	描述	所需技能
	`Linux-GA-8-1-2-Setup-f032d23e-881b-4dee-ba70-b9ca3e915910-ami-072509a7ffc156938.4` Amazon 機器映像 (AMI)。此預先設定的 AMI 可透過 AWS Marketplace 取得。 b. 選取執行個體類型。NetBackup 建議主要伺服器和媒體伺服器使`m5.2xlarge` 用。	
設定的備份主機 NetBackup。	1. 在 AWS 上登入 VMware 雲端 https://vmc.vmware.com/ 2. 選取軟體定義的資料中心。 3. 選擇開啟 VCENTER 索引標籤。這會開啟軟體定義的資料中心 vCenter。 4. 請記下備份主機的完整網域名稱 (FQDN)。 5. 登入 NetBackup 管理主控台。如需詳細資訊，請參閱登入 NetBackup 管理主控台 (Veritas 說明文件)。 6. 選取主要伺服器和媒體伺服器，然後選擇 VMware 存取主機。 7. 新增備份主機的 FQDN。 8. 選擇 Apply (套用)，然後選擇 OK (確定)。	雲端管理員、Backup 管理員

(選擇性) 設定 Amazon S3 儲存

任務	描述	所需技能
在 Amazon S3 中設定儲存。	1. 檢閱 Amazon S3 雲端儲存選項 (Veritas 文件)，然後根據您的需求選取適當的儲存類別。 2. 根據〈Veritas 說明文件〉中的〈設 NetBackup 定雲端儲存〉中的指示，設定為將 Amazon S3 用於雲端儲存。NetBackup	雲端管理員，一般 AWS

相關資源

AWS 文件

- 建立介面 VPC 端端點 (AWS PrivateLink 文件)

維利塔斯文件

- NetBackup 防火牆埠需求

VMware 說明文件

- 從內容程式庫中的 OVF 範本部署虛擬機器
- VMware Cloud on AWS 資料傳輸費用：其運作方式為何？ (VMware 部落格文章)
- VMware Cloud on AWS：延伸叢集

使用 AWS CLI 將資料從 S3 儲存貯體複製到另一個帳戶和區域

由巴加利（AWS）和普魯斯霍坦 G K（AWS）創建

環境：生產	技術：儲存與備份；雲端原生	AWS 服務：AWS CLI；AWS Identity and Access Management；Amazon S3

Summary

此模式描述如何將資料從 AWS 來源帳戶中的 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體遷移到另一個 AWS 帳戶的目標 S3 儲存貯體 (位於相同 AWS 區域或不同區域)。

來源 S3 儲存貯體透過使用附加的資源政策允許 AWS Identity and Access Management (IAM) 存取。目的地帳戶中的使用者必須擔任具有來源值區`PutObject`和`GetObject`權限的角色。最後，您執行`copy`和`sync`命令將資料從來源 S3 儲存貯體傳輸到目的地 S3 儲存貯體。

帳戶擁有上傳到 S3 儲存貯體的物件。如果您跨帳戶和區域複製物件，則會授與複製物件的目標帳戶擁有權。您可以將物件的存取控制清單 (ACL) 變更為，來變更物件的所有權`bucket-owner-full-control`。不過，我們建議您將程式設計跨帳戶權限授與目的地帳戶，因為 ACL 可能很難管理多個物件。

警告：此案例需要具有程式設計存取權限和長期登入資料的 IAM 使用者，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。如有必要，可更新存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的更新存取金鑰。

此模式涵蓋一次性移轉。對於需要將新物件從來源儲存貯體連續自動遷移到目的地儲存貯體的案例，您可以改用 S3 Batch 複寫，如使用 S3 Batch 複寫將資料從 S3 儲存貯體複製到另一個帳戶和區域模式中所述。

先決條件和限制

- 相同或不同 AWS 區域中的兩個有效 AWS 帳戶。

- 來源帳戶中現有的 S3 儲存貯體。
- 如果您的來源或目的地 Amazon S3 儲存貯體已啟用預設加密，則必須修改 AWS Key Management Service (AWS KMS) 金鑰許可。如需詳細資訊，請參閱關於此主題的 AWS Re:Post 文章。
- 熟悉跨帳戶權限。

架構

工具

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- AWS Command Line Interface (AWS CLI) (AWS CLI) 是一種開放原始碼工具，可協助您透過命令列殼層中的命令與 AWS 服務互動。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

最佳實務

- IAM 中的安全最佳做法 (IAM 文件)
- 套用最低權限許可 (IAM 文件)

史诗

在目的地 AWS 帳戶中建立 IAM 使用者和角色

任務	描述	所需技能
建立 IAM 使用者並取得存取金鑰。	1. 登入 AWS 管理主控台並建立具有程式設計存取權限的 IAM 使用者。如需詳細步驟，請參閱 IAM 說明文件中的建立 IAM 使用者。此使用者不需要附加任何策略。	AWS DevOps

任務	描述	所需技能
	2. 為此用戶生成訪問密鑰和密鑰。如需指示，請參閱 AWS 文件中的 AWS 帳戶和存取金鑰。	

任務	描述	所需技能
建立以 IAM 身分識別為基礎的政策。	使用下列權限建立以身分識別為S3MigrationPolicy 基礎的政策。如需詳細步驟，請參閱 IAM 文件中的建立 IAM 政策。 { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["s3:ListBucket", "s3:GetObject", "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::awsexamplesourcebucket", "arn:aws:s3:::awsexamplesourcebucket/*"] }, {	AWS DevOps

任務	描述	所需技能
	<pre>"Effect": "Allow", "Action": ["s3:ListBucket", "s3:PutObject", "s3:PutObjectAcl", "s3:PutObjectTagging", "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::awsexampledestinationbucket", "arn:aws:s3:::awsexampledestinationbucket/*"] }] }</pre> 注意：根據您的使用案例修改來源和目的地值區名稱。	

任務	描述	所需技能
	此基於身份的策略允許擔任此角色的使用者存取來源值區和目的地值區。	

任務	描述	所需技能
建立 IAM 角色。	建立使用下列信任政策命名`S3MigrationRole` 的 IAM 角色，然後附加先前建立的角色`S3MigrationPolicy` 。如需詳細步驟，請參閱 IAM 說明文件中的建立角色以將許可委派給 IAM 使用者。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::<destination_account>:user/<user_name>" }, "Action": "sts:AssumeRole", "Condition": {} } ] }` 附註：根據您的使用案例，修改信任政策中目的地 IAM 角色的 Amazon 資源名稱 (ARN) 或使用者名稱。 此信任政策允許新建立的 IAM 使用者假設`S3MigrationRole` 。	AWS DevOps

在來源帳戶中建立並附加 S3 儲存貯體政策

任務	描述	所需技能
建立並附加 S3 儲存貯體政策。	登入來源帳戶的 AWS 管理主控台，然後開啟 Amazon S3 主控台。選擇您的來源 S3 儲存貯體，然後選擇權限。在「值區政策」下，選擇「編輯」，然後貼上下列值區政策。選擇儲存。 { "Version": "2012-10-17", "Statement": [{ "Sid": "DelegateS3Access", "Effect": "Allow", "Principal": {"AWS": "arn:aws:iam::<destination_account>:role/<RoleName>"}, "Action": ["s3:ListBucket", "s3:GetObject", "s3:GetObjectTagging", "s3:GetObjectVersion", "s3:GetObjectVersionTagging"],	雲端管理員

任務	描述	所需技能
	<pre>"Resource": ["arn:aws:s3:::awse xamplesourcebucket/ *", "arn:aws:s3:::awse xamplesourcebucket"] }] }</pre> 注意：請務必包含目的地帳戶的 AWS 帳戶 ID，並根據需求設定儲存貯體政策範本。 此資源型政策允許目標角色存取S3MigrationRole 來源帳戶中的 S3 物件。	

設定目的地 S3 儲存貯體

任務	描述	所需技能
建立目的地 S3 儲存貯體。	登入目的地帳戶的 AWS 管理主控台，開啟 Amazon S3 主控台，然後選擇「建立儲存貯體」。根據您的需求建立 S3 儲存貯體。如需詳細資訊，請參閱 Amazon S3 文件中的建立儲存貯體。	雲端管理員

將資料複製到目的地 S3 儲存貯體

任務	描述	所需技能
使用新建立的使用者登入資料設定 AWS CLI。	1. 安裝最新版本的 AWS CLI。如需指示，請參閱 AWS CLI 文件中的安裝或更新最新版本的 AWS CLI。 2. 使用您建立的使用者的 AWS 存取金鑰執行`$ aws configure` 和更新 CLI。如需詳細資訊，請參閱 AWS CLI 文件中的組態和登入資料檔案設定。	AWS DevOps
假設 S3 移轉角色。	1. 使用 AWS CLI 假設`S3MigrationRole` ： `aws sts assume-role \ --role-arn "arn:aws:iam::<destination_account>:role/S3MigrationRole" \ --role-session-name AWSCLI-Session` 此指令會輸出數條資訊。在認證區塊內，您需要`AccessKeyId` `SecretAccessKey` 、和`SessionToken` 。此範例使用環境變數`RoleAccessKeyID` `RoleSecretKey` 、和`RoleSessionToken` 。請注意，到期欄位的時間戳記為 UTC 時	AWS 管理員

任務	描述	所需技能
	區。時間戳記會指出 IAM 角色的臨時登入資料到期的時間。如果臨時登入資料過期，您必須再次呼叫 `sts:AssumeRole` API。 2. 建立三個環境變數以擔任 IAM 角色。這些環境變數會填入下列輸出： `# Linux` `export AWS_ACCESS_KEY_ID=RoleAccessKeyID` `export AWS_SECRET_ACCESS_KEY=RoleSecretKey` `export AWS_SESSION_TOKEN=RoleSessionToken` `# Windows` `set AWS_ACCESS_KEY_ID=RoleAccessKeyID` `set AWS_SECRET_ACCESS_KEY=RoleSecretKey` `set AWS_SESSION_TOKEN=RoleSessionToken` 3. 執行下列命令，確認您已擔任 IAM 角色： `aws sts get-caller-identity`	

任務	描述	所需技能
	如需詳細資訊，請參閱 AWS 知識中心。	
將資料從來源 S3 儲存貯體複製並同步到目的地 S3 儲存貯體。	當您擔任角色時，S3MigrationRole 您可以使用複製 (cp) 或同步 (sync) 指令複製資料。 複製 (如需詳細資訊，請參閱 AWS CLI 命令參考)： `aws s3 cp s3:// DOC-EXAMPLE-BUCKET-SOURCE / \` `  s3:// DOC-EXAMPLE-BUCKET-TARGET / \` `  --recursive --source-region SOURCE-REGION-NAME --region DESTINATION-REGION-NAME` 同步 (如需詳細資訊，請參閱 AWS CLI 命令參考)： `aws s3 sync s3:// DOC-EXAMPLE-BUCKET-SOURCE / \` `  s3:// DOC-EXAMPLE-BUCKET-TARGET / \` `  --source-region SOURCE-REGION-NAME --region DESTINATION-REGION-NAME`	雲端管理員

故障診斷

問題	解決方案
調用ListObjects 操作時發生錯誤（AccessDenied ）：訪問被拒絕	• 確保您已擔任該角色S3MigrationRole 。 • 運行aws sts get-caller-identity 以檢查使用的角色。如果輸出未顯示的ARNS3MigrationRole ，請再次假設該角色並重試。

相關資源

- 建立 S3 儲存貯體 (Amazon S3 文件)
- Amazon S3 儲存貯體政策和使用者政策 (Amazon S3 文件)
- IAM 身分 (使用者、群組和角色) (IAM 文件)
- cp 命令 (AWS CLI 文件)
- 同步命令 (AWS CLI 文件)

使用 S3 Batch 複寫將資料從 S3 儲存貯體複製到另一個帳戶和區域

創建者：巴加利（AWS），拉克什米坎斯 B D（AWS），普魯斯霍姆 G K（AWS），舒伯姆哈索拉（AWS）和蘇曼·拉約蒂亞（AWS）

環境：PoC 或試點	技術：儲存與備份；雲端原生	AWS 服務：Amazon S3；AWS Identity and Access Management

Summary

此模式說明如何在設定儲存貯體之後，使用 Amazon 簡單儲存服務 (Amazon S3) Batch 複寫將 S3 儲存貯體的內容自動複製到另一個 S3 儲存貯體，而無需任何手動介入。來源和目的地值區可以位於相同或不同 AWS 帳戶 或區域中。

S3 Batch 複寫讓您可以複寫設定到位之前就已存在的 Amazon S3 物件、先前複寫的物件，以及複寫失敗的物件。此方法使用 S3 Batch 操作任務。工作完成時，您會收到完成報告。

在需要連續自動將新物件從來源儲存貯體遷移到目的地儲存貯體的案例中，您可以使用 S3 Batch 複寫。對於一次性遷移，您可以改用 AWS Command Line Interface (AWS CLI)，如使用將資料從 S3 儲存貯體複製到另一個帳戶和區域模式中所述 AWS CLI。

先決條件和限制

- 來源 AWS 帳戶。
- 一個目的地 AWS 帳戶。
- 來源帳戶中包含一些物件 (檔案或資料夾) 的 S3 儲存貯體。
- 目標帳戶中的一或多個 S3 儲存貯體。
- 在來源和目的地儲存貯體上啟用 S3 版本控制。
- AWS Identity and Access Management (IAM) 許可，可在來源和目的地帳戶上建立 IAM 政策、IAM 角色和 S3 儲存貯體政策。
- S3 Batch 複寫任務處於作用中狀態時，Amazon S3 生命週期規則會停用。這可確保來源值區和目標值區之間的同位檢查。否則，目的地值區可能不是來源值區的完全複本。

架構

工具

AWS 服務

- AWS Identity and Access Management (IAM) 透過控制經驗證和授權使用 AWS 資源的人員，協助您安全地管理對資源的存取。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

最佳實務

以下來自 AWS RE: Invent 2022 的影片討論使用 Amazon S3 複寫來達成法規合規、資料保護和提升應用程式效能的最佳實務。

史诗

為來源帳戶中的跨帳戶複寫建立 IAM 政策和角色

任務	描述	所需技能
建立跨帳戶複寫的 IAM 政策。	在來 AWS 源帳戶中： 1. 開啟 IAM 主控台。 2. 建立新的 IAM 政策。 3. 在 [原則編輯器] 區段中，選擇 [JSON]，然後貼上下列程式碼。 `{ "Version": "2012-10-17", "Statement": [ {`	雲端管理員、AWS 管理員

任務	描述	所需技能
	"Sid": "GetSourceBucketConfiguration", "Effect": "Allow", "Action": ["s3:ListBucket", "s3:GetBucketLocation", "s3:GetBucketAcl", "s3:GetReplicationConfiguration", "s3:GetObjectVersionForReplication", "s3:GetObjectVersionAcl", "s3:GetObjectVersionTagging"], "Resource": ["arn:aws:s3:::source-bucket-name", "arn:aws:s3:::source-bucket-name/*"] }, { "Sid": "ReplicateToDestinationBuckets",	

任務	描述	所需技能
	"Effect": "Allow", "Action": ["s3:List*", "s3:*Object", "s3:ReplicateObject", "s3:ReplicateDelete", "s3:ReplicateTags"], "Resource": ["arn:aws:s3:::destination-bucket-name/*", "arn:aws:s3:::destination-bucket-name/*"] }, { "Sid": "PermissionToOverrideBucketOwner", "Effect": "Allow", "Action": ["s3:ObjectOwnerOverrideToBucketOwner"],	

任務	描述	所需技能
	<pre>"Resource": ["arn:aws:s3:::destination-bucket-name/*", "arn:aws:s3:::destination-bucket-name/*"] }] }</pre> 本政策包括三項陳述： • GetSourceBucketConfiguration 提供複寫組態和物件版本的存取權，以便在來源值區上進行複寫。 • ReplicateToDestinationBuckets 提供複寫至目標值區的存取權。您可以在陣列中指定多個目的地值區。 • PermissionToOverrideBucketOwner 提供存取權，以ObjectOwnerOverrideToBucketOwner 便目的地值區可以擁有從來源帳戶複製的目標帳戶中的物件。	

任務	描述	所需技能
	4. 選擇 [下一步]，提供原則名稱 (例如)`cross-account-bucket-replication-policy` ，然後選擇 [建立原則]。 如需詳細資訊，請參閱 IAM 文件中的建立 IAM 政策。	
建立跨帳戶複寫的 IAM 角色。	在來 AWS 源帳戶中： 1. 在 IAM 主控台上，建立包含下列資訊的 IAM 角色： a. 對於受信任的實體類型，請選擇 AWS 服務。 b. 對於服務，請選擇 S3。 c. 對於使用案例，請選擇 S3 Batch 操作。 d. 選擇您在上一個步驟中建立的策略。 2. 提供角色名稱 (例如 cross-account-bucket-replication-role)，然後選擇 [建立角色]。 如需詳細資訊，請參閱 IAM 文件中的建立 IAM 角色。	雲端管理員、AWS 管理員

在來源帳戶中建立複製規則

任務	描述	所需技能
針對來源帳戶中的來源值區建立複製規則。	在來 AWS 源帳戶中： 1. 開啟 Amazon S3 主控台。 2. 切換作業選項至來源時段，然後選擇「管理」頁標。 3. 使用下列組態建立複製規則： a. 提供規則名稱，例如`s3-replication-rule` 。 b. 針對 Status (狀態)，請選擇 Enabled (啟用)。 c. 針對規則範圍，選擇「套用至值區中的所有物件」。 d. 針對「目的地」，選擇「在其他帳戶中指定時段」，然後輸入目的地 AWS 帳戶 編號與時段名稱。 e. 選擇將物件擁有權變更為目的地值區擁有者的選項。 f. 對於 IAM 角色，請選擇您先前在來源帳戶中建立的角色。 g. 對於其他複製選項，請選取所有可用的選項。這些功能可讓您快速複寫內容、透過 Amazon CloudWatch 指標監控複	AWS 管理員、雲端管理員

任務	描述	所需技能
	寫進度、複寫刪除標記，以及複寫中繼資料變更。 h. 選擇儲存。 4. 如果您有多個目的地值區，請建立其他複製規則。 如需詳細資訊，請參閱 Amazon S3 文件中的設定當來源和目的地儲存貯體屬於不同帳戶時進行複寫。	

將值區政策套用至目的地值區

任務	描述	所需技能
將值區政策套用至目的地值區。	必須針對目的地帳戶中 AWS 的每個目標值區個別執行此步驟。 在 AWS 目的地帳戶中： 1. 開啟 IAM 主控台，導覽至目的地值區，然後選擇「權限」索引標籤。 2. 透過提供下列 JSON 程式碼來編輯值區政策，並儲存原則： `{ "Version": "2012-10-17", "Id": "PolicyFo rDestinationBucket", "Statement": [`	AWS 管理員、AWS 系統管理員、雲端管理員

任務	描述	所需技能
	{ "Sid": "Permissions on objects and buckets", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::SourceAWSAccountNumber:role/IAM-Role-created-in-step1-in-source-account" }, "Action": ["s3:List*", "s3:GetBucketVersioning", "s3:PutBucketVersioning", "s3:ReplicateDelete", "s3:ReplicateObject"], "Resource": ["arn:aws:s3:::destination-bucket", "arn:aws:s3:::destination-bucket/*"] }, {	

任務	描述	所需技能
	"Sid": "Permission to override bucket owner", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::SourceAWSAccountNumber:role/IAM-Role-created-in-step1-in-source-account" }, "Action": "s3:ObjectOwnerOverrideToBucketOwner", "Resource": "arn:aws:s3:::destination-bucket/*" }] } 該政策包括兩個聲明： • Permissions on objects and buckets表示目標值區可以根據來源帳戶中定義的角色複製內容。此角色會提供來源值區的權限。 • Permission to override bucket owner表示目標值區具有覆寫來源帳戶擁有權的權限。	

測試 Amazon S3 跨帳戶複寫

任務	描述	所需技能
確認複寫是否正常運作。	1. 將物件新增至來源值區。 2. 確認新物件出現在目的地帳戶的 S3 儲存貯體中。 3. 檢視 CloudWatch 指標： a. 在來源值區中，選擇「指標」索引標籤。 b. 在複製測量結果段落中，選取複製規則。 c. 選擇 Display charts (顯示圖表)。這些圖表會顯示擱置複寫的作業、複寫延遲以及擱置複寫的位元組，以反映複寫的狀態。 如需詳細資訊，請參閱 Amazon S3 文件 CloudWatch 中的使用 Amazon 監控指標。	AWS 管理員、雲端管理員

相關資源

- 我什麼時候可以使用 IAM？ (IAM 文件)
- IAM 的運作方式 (IAM 文件)
- 建立 IAM 角色 (IAM 文件)
- 建立 IAM 政策 (IAM 文件)
- 存取管理概觀：許可和政策 (IAM 文件)
- 建立、設定和使用 Amazon S3 儲存貯體 (Amazon S3 文件)
- 在 Amazon S3 中上傳、下載和使用物件 (Amazon S3 文件)
- 複寫物件 (Amazon S3 文件)

使 DistCp 用 PrivateLink 適用於 Amazon S3 的 AWS，將資料從現場部署 Hadoop 環境遷移到 Amazon S3

由傑森·歐文斯（AWS），安德烈斯·坎托爾（AWS），傑夫·克洛普芬斯坦（AWS），布魯諾·羅查奧利維拉和塞繆爾·施密特（AWS）創建

環境：生產	來源：哈多	目標：任何
R 類型：重新平台	工作負載：開源	技術：儲存與備份、分析
AWS 服務：Amazon S3；Amazon EMR		

Summary

此模式示範如何透過使用 PrivateLink 適用於亞馬遜簡單儲存服務 (Amazon S3) 的 Apache 開放原始碼工具DistCp搭配使用 Apache 開放原始碼工具，將幾乎任何數量的資料從現場部署 Apache Hadoop 環境遷移到亞馬遜網路服務 (AWS) 雲端。您可以使用 Amazon S3 的 AWS PrivateLink ，透過現場部署資料中心和 Amazon Amazon Virtual Private Cloud (Amazon VPC) 之間的私有網路連線將資料遷移到 Amazon S3，而不是使用公用網際網路或代理解決方案遷移資料。如果您在 Amazon Route 53 中使用 DNS 項目，或在現場部署 Hadoop 叢集的所有節點的 /etc/hosts 檔案中新增項目，系統會自動將您導向至正確的介面端點。

本指南提供將資料遷移到 AWS 雲端的使 DistCp 用說明。 DistCp 是最常用的工具，但有其他移轉工具可供使用。例如，您可以使用 AWS Snowball 或 AWS 雪地摩托等離線 AWS 工具，或使用 AWS Storage Gateway 或 AWS 等線上 AWS 工具。 DataSync此外，您可以使用其他開源工具，例如 Apache NiFi。

先決條件和限制

先決條件

- 使用中的 AWS 帳戶，在現場部署資料中心和 AWS 雲端之間具有私有網路連線
- Hadoop 的，安裝在具有內部設備 DistCp
- 在 Hadoop 分佈式文件系統（HDFS）訪問遷移數據的 Hadoop 用戶

- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)
- 將物件放入 S3 儲存貯體的權限

限制

適用 PrivateLink 於 Amazon S3 的 AWS 適用虛擬私有雲端 (VPC) 限制。如需詳細資訊，請參閱界面端點屬性和限制和 AWS PrivateLink 配額 (AWS PrivateLink 文件)。

PrivateLink 適用於 Amazon S3 的 AWS 不支援以下功能：

- 聯邦資訊處理標準 (FIPS) 端點
- 網站端點
- 舊版全域端點

架構

源, 技術, 堆棧

- 已安裝 DistCp 的 Hadoop 集群

目標技術堆疊

- Amazon S3
- Amazon VPC

目標架構

該圖顯示了 Hadoop 管理員如 DistCp 何使用透過 Amazon S3 界面端點，透過私有網路連接 (例如 AWS Direct Connect) 從現場部署環境複製資料到 Amazon S3。

工具

AWS 服務

- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。
- Amazon Virtual Private Cloud (Amazon VPC) 可協助您將 AWS 資源啟動到您已定義的虛擬網路中。這個虛擬網路類似於您在自己的資料中心中操作的傳統網路，並具有使用 AWS 可擴展基礎設施的好處。

其他工具

- 阿帕奇 Hadoop DistCp（分佈式副本）是用於複製大型集群間和內部集群的工具。 DistCp 使用 Apache MapReduce 進行分發，錯誤處理和恢復以及報告。

史诗

將資料遷移到 AWS 雲端

任務	描述	所需技能
為 Amazon S3 建立適用 PrivateLink 於 AWS 的端點。	1. 登入 AWS 管理主控台並開啟 Amazon VPC 主控台。 2. 在瀏覽窗格上，選擇 [端點]，然後選擇 [建立端點]。 3. 在 Service category (服務類別) 中，選擇 AWS services (AWS 服務)。 4. 在搜尋方塊中，輸入 s3，然後按 Enter 鍵。 5. 在搜索結果中，選擇喜歡的。 < your-aws-region >.s3 服務名稱，其中「類型」欄中的值為「介面」。 6. 在 VPC 中，選擇您的 VPC。對於子網路，請選擇您的子網路。	AWS 管理員

任務	描述	所需技能
	7. 在安全性群組中，選擇或建立允許 TCP 443 的安全性群組。 8. 根據您的需求新增標籤，然後選擇 [建立端點]。	
驗證端點並找到 DNS 項目。	1. 開啟 Amazon VPC 主控台，選擇端點，然後選取您先前建立的端點。 2. 在 [詳細資料] 索引標籤上，尋找 DNS 名稱的第一個 DNS 項目。這是區域性 DNS 項目。當您使用此 DNS 名稱時，請求在可用區域特定的 DNS 項目之間進行替代。 3. 選擇「子網路」頁籤。您可以在每個可用區域中找到端點 elastic network interface 的位址。	AWS 管理員

任務	描述	所需技能
檢查防火牆規則和路由配置。	若要確認您的防火牆規則已開啟且網路組態已正確設定，請使用 Telnet 在連接埠 443 上測試端點。例如： `$ telnet vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com 443` `Trying 10.104.88.6...` `Connected to vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com.` `...` `$ telnet vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com 443` `Trying 10.104.71.141...` `Connected to vpce-<your-VPC-endpoint-ID>.s3.us-east-2.vpce.amazonaws.com.` 備註：如果您使用區域項目，則成功測試表明 DNS 會在您在 Amazon VPC 主控台中選取端點的「子網路」索引標籤上看到的兩個 IP 位址之間交替。	網路管理員、AWS 管理員

任務	描述	所需技能
設定名稱解析。	您必須設定名稱解析，以允許 Hadoop 存取 Amazon S3 介面端點。您無法使用端點名稱本身。相反地，您必須解決`<your-bucket-name>.s3.<your-aws-region>.amazonaws.com` 或`*.s3.<your-aws-region>.amazonaws.com` 。如需此命名限制的詳細資訊，請參閱介紹 Hadoop S3A 用戶端 (Hadoop 網站)。 選擇下列其中一個組態選項： • 使用內部部署 DNS 解析端點的私有 IP 位址。您可以覆寫所有值區或選取值區的行為。如需詳細資訊，請參閱使用 AWS 安全混合式存取 Amazon S3 (AWS 部落格文章) 中的「選項 2：使用網域名稱系統回應政策區域 PrivateLink (DNS RPZ) 存取 Amazon S3」。 • 將內部部署 DNS 設定為有條件地將流量轉送至 VPC 中的解析器輸入端點。交通被轉發到 Route 53。如需詳細資訊，請參閱使用 AWS 安全混合式存取 Amazon S3 中的「選項 3：使用 Amazon Route 53 Resolver 輸入端點從現場部署轉送 DNS 請求」	AWS 管理員

任務	描述	所需技能
	PrivateLink (AWS 部落格文章)。 • 在 Hadoop 叢集中的所有節點上編輯 /etc/hosts 檔案。這是測試的臨時解決方案，不建議用於生產環境。若要編輯 /etc/hosts 檔案，請為或新增一個項目。<your-bucket-name>.s3.<your-aws-region>.amazonaws.com s3.<your-aws-region>.amazonaws.com /etc/hosts 檔案不能有多個 IP 位址供一個項目使用。您必須從其中一個可用區域中選擇單一 IP 位址，然後該區域會變成單一故障點。	

任務	描述	所需技能
設定 Amazon S3 的身份驗證。	若要透過 Hadoop 向 Amazon S3 進行驗證，我們建議您將臨時角色登入資料匯出至 Hadoop 環境。如需詳細資訊，請參閱使用 S3 進行驗證 (Hadoop 網站)。對於長時間執行的任務，您可以建立使用者並指派具有僅將資料放入 S3 儲存貯體的權限的政策。訪問密鑰和密鑰可以存儲在 Hadoop 上，只能訪問 DistCp 作業本身和 Hadoop 管理員。有關存儲秘密的詳細信息，請參閱使用 Hadoop 憑據提供程序存儲秘密（Hadoop 網站）。如需其他身份驗證方法的詳細資訊，請參閱 AWS IAM 身分中心的文件 (AWS 單一登入的後續任務) 中如何取得 IAM 角色的登入資料以用於 AWS 帳戶的 CLI 存取。 若要使用臨時身分證明，請將臨時認證新增至您的認證檔案，或執行下列命令，將認證匯出至您的環境： `export AWS_SESSION_TOKEN=SECRET-SESSION-TOKEN` `export AWS_ACCESS_KEY_ID=SESSION-ACCESS-KEY` `export AWS_SECRET_ACCESS_KEY=SESSION-SECRET-KEY`	AWS 管理員

任務	描述	所需技能
	如果您有傳統的存取金鑰和私密金鑰組合，請執行下列命令： `export AWS_ACCESS_KEY_ID=my.aws.key` `export AWS_SECRET_ACCESS_KEY=my.secret.key` 注意：如果您使用存取金鑰和私密金鑰組合，請將 DistCp 命令中的認證提供者從變更"org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider" 為"org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider" 。	

任務	描述	所需技能
使用傳輸資料 DistCp。	若要用 DistCp 於傳輸資料，請執行下列命令： `hadoop distcp -Dfs.s3a.aws.credentials.provider=\"org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider" \ -Dfs.s3a.access.key="${AWS_ACCESS_KEY_ID}" \ -Dfs.s3a.secret.key="${AWS_SECRET_ACCESS_KEY}" \ -Dfs.s3a.session.token="${AWS_SESSION_TOKEN}" \ -Dfs.s3a.path.style.access=true \ -Dfs.s3a.connection.ssl.enabled=true \ -Dfs.s3a.endpoint=s3.<your-aws-region>.amazonaws.com \ hdfs:///user/root/ s3a://<your-bucket-name>` 注意：當您將 DistCp 命令與適用 PrivateLink 於 Amazon S3 的 AWS 搭配使用時，不會自動探索端點的 AWS 區域。Hadoop 3.3.2 及更新版本透過啟用明確設定 S3 儲存貯體的 AWS 區域的選項來解決此問題。如需詳細資訊，請	遷移工程師，AWS 管理員

任務	描述	所需技能
	參閱 S3A 以新增選項以設定 AWS 區域 (Hadoop 網站)。 如需其他 S3A 提供者的詳細資訊，請參閱一般 S3A 用戶端組態 (Hadoop 網站)。例如，如果您使用加密，則可以根據您的加密類型，將下列選項新增至上述一系列命令中： `-Dfs.s3a.server-side-encryption-algorithm=AES-256 [or SSE-C or SSE-KMS]` 備註：若要將介面端點與 S3A 搭配使用，您必須為 S3 區域名稱建立 DNS 別名項目 (例如，`s3.<your-aws-region>.amazonaws.com`) 到介面端點。如需指示，請參閱設定 Amazon S3 的身分驗證一節。Hadoop 3.3.2 和更早版本需要此因應措施。未來版本的 S3A 不需要此因應措施。 如果您在使用 Amazon S3 時遇到簽名問題，請新增使用簽名版本 4 (SIGv4) 簽署的選項： `-Dmapreduce.map.java.opts="-Dcom.amazonaws.services.s3.enableV4=true"`	

用 CloudEndure 於內部部署資料庫的嚴重損壞復原

由尼尚耆那教（AWS）和阿努拉格迪孔達（AWS）創建

環境：PoC 或試點	技術：儲存與備份、現代化、資料庫

Summary

警告：IAM 使用者擁有長期登入資料，這會帶來安全風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。

此病毒碼使用 CloudEndure 嚴重損壞修復和 CloudEndure 容錯回復用戶端進行嚴重損壞修復 (DR)。它使用 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體為現場部署資料中心主機設定 DR。

您必須使用 CloudEndure 容錯回復用戶端從非雲端或其他雲端基礎設施複製到 Amazon Web Services (AWS) 雲端。災難事件結束後，您將想要故障備份您的機器。 CloudEndure 將資料複製的方向從目標機器反轉回來源機器，讓您準備容錯回復。 CloudEndure 使用者主控台會將目前啟動的目標電腦視為來源電腦。複寫會從選取的目標電腦反轉回原始來源基礎結構。

重要事項：AWS 於 2021 年 11 月推出 AWS 彈性災難復原，現在是 AWS 災難復原的建議服務。

成功啟動彈性災難復原之後，AWS 將開始限制 CloudEndure 災難復原在所有 AWS 區域的可用性，包括 AWS GovCloud (US) 區域 (將繼續支援 AWS 中國區域)。這將根據以下時間表進行：

1. 2023 年 9 月 1 日 — 客戶將無法在任何 AWS 區域註冊新的 CloudEndure DR 帳戶 (AWS 中國區域除外)。
2. 2023 年 12 月 1 日 — 任何 AWS 區域都不再支援新的 CloudEndure DR 代理程式安裝 (AWS 中國區域除外)。請注意，將支援現有代理程式的升級。
3. 2024 年 3 月 31 日 — CloudEndure DR 將在所有 AWS 區域停用 (AWS 中國區域除外)。
4. 如需 CloudEndure 災難復原 EOL 的任何更新時間表，請參閱文件CloudEndure 。

本刊物將於二零二四年三月三十一日移除。如果正在進行的移轉專案需要此功能，請使用此頁面標題下方的 PDF 連結下載並儲存 PDF 檔案。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 內部部署資料庫

架構

源, 技術, 堆棧

- 內部部署資料中心中的資料庫

目標技術堆疊

- EC2 執行個體上的資料庫 (如需受支援作業系統版本的完整清單，請參閱 Amazon EC2 常見問答集)

來源與目標網路架構

工具

- CloudEndure 災難復原 — CloudEndure 災難復原可以快速、可靠地將實體、虛擬和雲端伺服器復原到 AWS，藉此減少停機時間和資料遺失。 CloudEndure 災難復原會持續將您的機器 (包括作業系統、系統狀態組態、資料庫、應用程式和檔案) 複製到目標 AWS 帳戶和偏好區域中低成本的暫存區域。如果發生災難，您可以指示 CloudEndure 災難復原在幾分鐘內以完全佈建的狀態自動啟動數千部機器。

史诗

訂閱 CloudEndure 災難復原

任務	描述	所需技能
訂閱 CloudEndure 災難復原。	CloudEndure 災難復原可在 AWS Marketplace 中使用。	一般 AWS

任務	描述	所需技能
創建一個 CloudEndure 帳戶。	註冊 CloudEndure 並創建一個帳戶。然後，在電子郵件中確認訂閱。	一般 AWS
設置帳戶密碼並接受條款和條件。	密碼必須至少有八個字元，且至少必須包含一個大寫字母、一個小寫字母、一個數字和一個特殊字元。	一般 AWS

建立 CloudEndure 專案

任務	描述	所需技能
登入 CloudEndure 使用者主控台。	在CloudEndure 使用者主控台上，使用您在上一個步驟中建立的認證登入。	CloudEndure 管理員
建立新專案。	在主控台的左上角，選擇加號 (+) 按鈕來建立專案。選取嚴重損壞修復作為專案類型。您可以透過 AWS Marketplace 取得授權。	CloudEndure 管理員

產生和使用 AWS 登入資料

任務	描述	所需技能
建立 CloudEndure 解決方案的 IAM 政策。	您必須為執行 CloudEndure 解決方案建立的 AWS Identity and Access Management (IAM) 政策是以預先定義的CloudEndure 政策為基礎。此 CloudEndure 政策包含使用	AWS 系統管理員

任務	描述	所需技能
	AWS 做為目標基礎設施的必要許可。	
建立新的 IAM 使用者並產生 AWS 登入資料。	若要為 CloudEndure 使用者主控台產生必要的 AWS 登入資料，請至少建立一個 IAM 使用者，並將 CloudEndure 許可政策指派給此使用者。控制台需要訪問密鑰 ID 和秘密訪問密鑰。 若要遵循管理 AWS 存取金鑰的最佳實務，您應該定期輪換 IAM 金鑰。變更 IAM 金鑰會導致複寫伺服器重新啟動，導致暫時性延遲。	AWS 系統管理員
設定暫存區帳戶認證。	登入CloudEndure 使用者主控台，然後選取您的移轉專案。 在「設定和資訊」索引標籤上，導覽至 AWS 登入資料，並提供您的 AWS 存取金鑰 ID 和秘密存取金鑰 ID。	AWS 系統管理員

設定複製設定

任務	描述	所需技能
定義複製伺服器。	如需詳細資訊，請參閱CloudEndure 文件。	CloudEndure 管理員

在來源機器上安裝 CloudEndure 代理程式

任務	描述	所需技能
找到您的代理程式安裝權杖。	在 CloudEndure 使用者主控台上，導覽至電腦、機器動作、新增機器。 當您在來源機器上執行安裝程式檔案時，系統會先要求您輸入安裝 Token。令牌是一串唯一的字符串，在您的 CloudEndure 帳戶激活時自動為您生成。您可以使用一個安裝 Token，在專案允許的任意數量的來源機器上安裝代理程式。	CloudEndure 管理員
在 Linux 電腦上，執行安裝程式。	對於 Linux 電腦，請複製安裝程式命令、登入來源電腦，然後執行安裝程式。 如需詳細指示，請參閱CloudEndure 文件。	CloudEndure 管理員
在視窗電腦上，執行安裝程式。	對於 Windows 電腦，請將安裝程式檔案下載到每部電腦，然後執行安裝程式命令。 如需詳細指示，請參閱CloudEndure 文件。	CloudEndure 管理員
複製資料。	安裝代理程式之後，CloudEndure 開始複寫來源機器會啟動到暫存區。初始同步完成後，電腦會出現在 CloudEndure 使用者主控台的電腦索引標籤上。	CloudEndure 管理員

設定目標機器的藍圖

任務	描述	所需技能
選擇藍圖的來源機器。	在 CloudEndure 使用者主控台的電腦索引標籤上，選擇來源電腦以存取電腦詳細資料窗格。	CloudEndure 管理員
設定目標機器的藍圖。	在藍圖索引標籤上，根據您的需求設定目標機器的設定。如需詳細指示，請參閱CloudEndure 文件。	CloudEndure 管理員

測試您的 DR 解決方案

任務	描述	所需技能
使用「測試模式」來測試解決方案。	如需測試模式和測試切換驗證的詳細說明，請參閱文件CloudEndure 。	CloudEndure 管理員
測試在 Amazon EC2 伺服器上啟動的目標執行個體。	若要測試每部目標電腦，請選擇機器的名稱。然後開啟「目標」索引標籤，複製新的 IP 位址，然後在 Amazon EC2 執行個體上登入新啟動的伺服器。	CloudEndure 管理員

使用執行容錯移轉 CloudEndure

任務	描述	所需技能
驗證來源機器狀態。	在 [CloudEndure 使用者主控台電腦] 頁面上，確認您要容錯移轉的來源電腦具有下列狀態指示：	CloudEndure 管理員

任務	描述	所需技能
	• 資料複製進度 — 持續資料保護 • 狀態 — 火箭圖示，表示目標機器可以啟動 • 災難復原生命週期 — 最近測試	
開始切換。	1. 在 [電腦] 頁面上，選擇您的來源機器。 2. 在啟動目標電腦索引標籤上，選擇復原模式。 3. 選擇目標機器的復原點。啟動容錯移轉的新目標電腦時，系統將使用復原點。您可以使用最新的復原點，或從清單中選擇先前的復原點。 4. 選擇「繼續啟動」。	CloudEndure 管理員
檢查工作進度和完成狀態。	「Job 進度」視窗會顯示目標機器啟動程序的詳細資訊。 容錯移轉完成之後，CloudEndure 使用者主控台上的嚴重損壞修復生命週期狀態會變更為容錯移轉，表示成功完成。	CloudEndure 管理員

使用容錯回復用戶端執行容錯回 CloudEndure 復

任務	描述	所需技能
檢閱 CloudEndure 容錯回復用戶端需求。	使用 CloudEndure 容錯回復用戶端從現場部署或其他雲	CloudEndure 管理員

任務	描述	所需技能
	端基礎設施複寫到 AWS。CloudEndure 容錯回復用戶端具有下列需求： • 機器必須設定為在 BIOS 模式下開機，並支援 MBR 開機。不支援設定為在 UEFI 模式下開機 (僅支援 GPT 開機) 的機器。 • CloudEndure 容錯回復用戶端至少需要 4 GB 的專用 RAM。	
準備容錯回復。	在您可以啟動準備容錯回復動作之前，所有來源電腦必須已在測試模式或復原模式下啟動目標電腦。 在「專案動作」功能表上，選擇「準備容錯回復」，然後選擇「繼續」。顯示將 CloudEndure 代理程式與容錯回復用戶端配對時，機器已準備好進行容錯回復。	CloudEndure 管理員

任務	描述	所需技能
在您的內部部署環境中下載 CloudEndure 容錯回復用戶端。	若要將 CloudEndure 容錯回復用戶端下載至您的來源環境，請執行下列動作： 1. 在 DR 專案中，選擇 [設定與資訊]。 2. 在 [複寫設定] 頁面上，選擇 [瞭解失敗回到 [其他基礎結構] 連結。 3. 在 [失敗回到無法識別的雲端/其他基礎結構] 對話方塊中，選擇 [從這裡下載]。 該文件將自動下載。	CloudEndure 管理員
起始內部部署機器的複寫。	若要起始來源機器的複製，必須將目標機器開機至 CloudEndure 容錯回復用戶端映像 (`failback_client.iso`)。如果用戶端無法使用動態主機設定通訊協定 (DHCP) 擷取網路設定，請手動輸入設定。 CloudEndure 容錯回復用戶端會透過 TCP 連接埠 443 連線到 console.clouendure.com，並使用系統提示您輸入的認證進行驗證。 CloudEndure	CloudEndure 管理員

任務	描述	所需技能
依照指示提供必要的詳細資料。	提供下列詳細資訊： • 安裝令牌 • 來源機器的電腦 ID • 來源與目標之間的磁碟對應 確定 CloudEndure 容錯回復用戶端已透過公用或私人 IP 位址與 CloudEndure 使用者主控台和目標機器連線。	CloudEndure 管理員
找到來源機器 ID。	若要尋找來源機器 ID，請在電腦索引標籤上選擇機器名稱，然後從來源索引標籤複製 ID。	CloudEndure 管理員
將來源機器 Connect 至目標電腦。	在現場部署伺服器 (目標機器) 中提供來源機器 ID (AWS 上的伺服器現在是容錯回復的來源)。AWS 機器 (來源) 會連線到 TCP 連接埠 1500 上的現場部署伺服器 (目標) 以開始複寫。 初始複寫完成後， CloudEndure 使用者主控台會指出複寫處於連續資料保護模式。	CloudEndure 管理員
視需要編輯容錯回復設定。	若要編輯容錯回復設定，請選擇機器名稱，然後選擇容錯回復設定索引標籤。	CloudEndure 管理員

任務	描述	所需技能
啟動目標電腦。	若要啟動目標電腦，請執行下列動作： 選取每個電腦名稱左邊的核取方塊，然後選擇 [啟動 x 目標電腦]，然後選擇 [復原模式]。 在對話方塊中，選擇「下一步」。 選擇 [最新] 復原點，然後選擇 [繼續啟動]。 啟動程序完成後，「 CloudEndure 使用者主控台」會在「資料複製進度」下顯示「將 CloudEndure 代理程式與複製伺服器配對」狀態。	CloudEndure 管理員
使機器恢復正常運行。	現在變更資料複寫的方向，讓現場部署機器成為來源，而 AWS 機器就是目標。選擇「專案動作」，然後選擇「恢復正常」和「繼續」。 資料複製的方向會反轉，並且機器會進行初始同步處理程序。當「資料複製進度」欄顯示所有機器的「連續資料保護」狀態時，容錯回復程序即完成。	CloudEndure 管理員

相關資源

AWS Marketplace

- CloudEndure 災難復原

CloudEndure 文件

- 登入主控台
- 建立專案
- 產生及使用認證
- 設定複製設定
- 安裝 CloudEndure 代理
- 執行災難復原容錯移

教學課程和影片

- CloudEndure 疑難排解劇本
- CloudEndure 影片
- 災難復原至 AWS 示範

更多模式

- 使用和事件將事件驅動的備份自動化 CodeCommit 到 Amazon S3 CodeBuild CloudWatch
- 使用動 DynamoDB TTL 自動將項目存檔到 Amazon S3
- 使用 Systems Manager 和自動備份 SAP HANA 資料庫 EventBridge
- 使用 BMC AMI 雲端資料將大型主機資料備份並存檔到 Amazon S3
- 使用 AWS Glue 建立 ETL 服務管道，以遞增方式將資料從 Amazon S3 載入到亞馬遜紅移
- 使用 Python 在 AWS 上將 EBCDIC 資料轉換並解壓縮為 ASCII
- 將甲骨 PostgreSQL 的 VARCHAR2 (1) 數據類型轉換為 Amazon Aurora 爾數據類型
- 使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統
- ???
- 估算 Amazon DynamoDB 表格的儲存成本
- 使用 Security Hub 識別 AWS Organizations 中的公有 S3 儲存貯體
- 將適用於 Oracle 資料庫執行個體的 Amazon RDS 移轉到使用 AMS 的其他帳戶
- 使用適用於 SFTP 的 AWS 轉移，將現場部署 SFTP 伺服器遷移到 AWS
- 使用 AWS DMS 將甲骨文分區資料表遷移到 PostgreSQL
- 使用複製將資料從 Microsoft Azure Blob 遷移到 Amazon S3
- 將甲骨 PostgreSQL 值遷移到 AWS 上的個別資料列
- 在 AWS 大型遷移中遷移共用檔案系統
- 使用 AWS SFTP 將小型資料集從現場部署遷移到 Amazon S3
- 監控 Amazon Aurora 是否有沒有加密的
- ???
- 搭配 AWS Fargate 使用 Amazon EKS 上的 Amazon EFS，以持續性資料儲存執行可設定狀態工作負載
- 成功將 S3 儲存貯體匯入為 AWS CloudFormation 堆疊
- 使用 AWS 在不同 AWS 區域的 Amazon EFS 檔案系統之間同步資料 DataSync
- 檢視 AWS 帳戶或組織的 EBS 快照詳細資訊

網頁及行動應用程式

主題

- 從 AWS 儲存庫持續部署現代 AWS Amplify Web 應用程式 CodeCommit
- 使用 AWS Amplify 增建立反應應用程式，並使用 Amazon Cognito 新增身份驗證
- 將反應型單頁應用程式部署到 Amazon S3 和 CloudFront
- 使用私有端點和應用 Application Load Balancer 在內部網站上部署 Amazon API Gateway API
- 在本地角度應用程序中嵌入 Amazon QuickSight 儀表板
- 更多模式

從 AWS 儲存庫持續部署現代 AWS Amplify Web 應用程式 CodeCommit

創建者：迪克什圖魯五寶塔科塔卡（AWS）和西片假村（AWS）

環境：PoC 或試點	技術：Web 和移動應用程序 DevOps; 現代化	AWS 服務：AWS Amplify；AWS CodeCommit

Summary

現代 Web 應用程序構建為單頁應用程序（SPA），將所有應用程序組件打包到靜態文件中。透過使用 AWS Amplify 託管，您可以建立持續整合和持續部署 (CI/CD) 管道，以建置、部署和託管在 Git 儲存庫中管理的現代 Web 應用程式。當您將 Amplify Hosting 連接到代碼存儲庫時，每次提交都會啟動一個工作流程來部署應用程序的前端和後端。這種方法的好處是，只有在成功完成部署後才會更新 Web 應用程式，以避免前端與後端之間的不一致。

在此模式中，您可以使用 AWS CodeCommit 儲存庫來管理現代 Web 應用程式。這些指示中的範例 Web 應用程式使用 React SPA 架構。然而，Amplify 託管支持許多其他 SPA 框架，如角，Vue，Next.js，它還支持單站點生成器，如蓋茨比，雨果和傑基爾。

此模式適用於具有下列服務和概念經驗的 AWS 建置人員：

- AWS CodeCommit
- AWS Amplify 託管
- 反應
- JavaScript
- Node.js
- NPM
- Git

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 在 Amplify 和 CodeCommit. 中建立資源的權限。如需詳細資訊，請參閱適用於 Amplify 的身分 Identity and Access Management 理以及 AWS CodeCommit 的身分與存取管理。
- 已安裝和設定的 AWS Command Line Interface (AWS CLI) (AWS CLI)。
- 文字編輯器或程式碼編輯器。
- CodeCommit，為使用 Git 認證的 HTTPS 使用者進行設定。
- 「Amplify」的 IAM 服務角色。
- 故宮和 Node.js，已安裝（故宮文檔）。

限制

- 此模式不會討論 Amplify 應用程式 (例如 API、驗證或資料庫) 後端的開發和整合。如需有關後端的詳細資訊，請參閱 Amplify 文件中的建立後端。

產品版本

- AWS CLI 2.0 版
- Node.js 版本 16.x 或更新版本

架構

目標技術堆疊

- 包含反應 SPA 的 AWS CodeCommit 儲存庫
- AWS Amplify 託管工作流程

目標架構

工具

AWS 服務

- AWS Amplify 託管提供基於 Git 的工作流程，用於託管具有持續部署的全堆疊無伺服器 Web 應用程式。

- AWS CodeCommit 是一種版本控制服務，可協助您以私密方式存放和管理 Git 儲存庫，而無需管理自己的原始檔控制系統。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。

其他工具

- Node.js 是一個事件驅動的 JavaScript 運行時環境，旨在構建可擴展的網絡應用程序。
- npm 是一個在 Node.js 環境中運行的軟件註冊表，用於共享或借用軟件包以及管理私有軟件包的部署。

史诗

建立儲 CodeCommit 存庫

任務	描述	所需技能
建立 儲存庫。	如需指示，請參閱 CodeCommit 文件中的建立 AWS CodeCommit 儲存庫。	AWS DevOps
複製儲存庫。	如需指示，請參閱說明 CodeCommit 文件中的透過複製存放庫 Connect 至存放庫。CodeCommit 如果系統提示您，請提供 Git 認證。	應用程式開發人員

創建一個反應應用

任務	描述	所需技能
創建一個新的 React 應用程序。	1. 輸入以下命令以導航到克隆的回購庫。替換<repo name>為您的 CodeCommit 存儲庫的名稱。	應用程式開發人員

任務	描述	所需技能
	`` $ cd <repo name> ``` 2. 輸入以下指令，在複製的儲存庫中建立新的 React 應用程式。 ``` $ npx create-react-app . ``` 3. 編碼應用程序，然後輸入以下命令以啟動它。 ``` $ npm start ``` 有關創建自定義 React 應用程序的更多信息，請參閱創建 React 應用程序文檔中的創建反應應用程序說明。您也可以依照 Amplify 文件中部署前端中的指示，將範例 React 應用程式部署到您的 Amplify 帳戶。	

任務	描述	所需技能
創建一個分支並推送代碼。	1. 輸入以下命令在本地創建新分支，其中<branch>是要分配給新分支的名稱。 `$ git checkout -b <branch>` 2. 輸入下列指令，將分支推送至 CodeCommit 儲存庫，其中<branch>是您在上一個步驟中指派的名稱。如需詳細資訊，請參閱使用提交。 `$ git push --set-upstream origin <branch>`	應用程式開發人員

在 AWS Amplify 託管中部署應用程式

任務	描述	所需技能
Connect Amplify 到存儲庫。	有關說明，請參閱 Amplify 主機文檔中的 Connect 存儲庫。選取 AWS 以 CodeCommit 及您先前建立的儲存庫和分支。	應用程式開發人員
定義前端構建設置。	有關說明，請參閱 Amplify 託管文檔中的確認前端的構建設置。接受預設值或輸入以下內容。 `Build settings:` `version: 0.1` `frontend:` `  phases:` `  preBuild:` `  commands:`	應用程式開發人員

任務	描述	所需技能
	<pre>- npm ci build: commands: - npm run build artifacts: baseDirectory: build files: - '**/*' cache: paths: - node_modules/ **/*</pre>	
檢閱和部署。	如需指示，請參閱 Amplify 主機文件中的儲存和部署。等到部署程序完成。	應用程式開發人員

驗證持續部署

任務	描述	所需技能
驗證初始部署。	部署程序完成時，在 [網域] 下，選擇連結。確認應用程式是否如預期般運作。	應用程式開發人員
將變更推送至程式碼儲存庫。	編輯本機工作站上的程式碼，並將變更推送至 CodeCommit 儲存庫。Amplify 託管檢測存儲庫中的更改，並自動啟動構建和部署過程。確認應用程式更新在網域上可見。	應用程式開發人員

相關資源

AWS CodeCommit 文件集

- 為 AWS 設定 CodeCommit
  - 使用 Git 認證的 HTTPS 使用者進行設定
  - 使用 AWS CLI 登入資料協助程式，在 Linux、macOS 或 Unix 上進行 HTTPS 連線至 AWS CodeCommit 儲存庫的設定步驟
- 開始使用 AWS CodeCommit

AWS Amplify 託管文件

- 開始使用現有程式碼
- 設定自訂網域

反應資源

- 創建應用程序網站
- 創建應用程序文檔
- 創建反應應用程序存儲庫（GitHub）

使用 AWS Amplify 增建立反應應用程式，並使用 Amazon Cognito 新增身份驗證

創建者日施新格拉 (AWS)

環境：PoC 或試點	技術：Web 和行動應用程式；安全性、身分識別、合規性	工作負載：所有其他工作
AWS 服務：AWS Amplify；Amazon Cognito		

Summary

此模式示範如何使用 AWS Amplify 建立反應型應用程式，以及如何使用 Amazon Cognito 將身份驗證新增至前端。AWS Amplify 包含一組工具 (開放原始碼架構、視覺開發環境、主控台) 和服務 (Web 應用程式和靜態網站託管)，可加速 AWS 上行動應用程式和 Web 應用程式的開發。

先決條件和限制

前提

- 有效的 AWS 帳戶
- Node.js 和故宮安裝在您的計算機上

產品版本

- Node.js 版本 10.x 或更新版本（要驗證您的版本，請在終端機窗口`node -v`中運行）
- npm 版本 6.x 或更高版本（要驗證您的版本，請在終端窗口`npm -v`中運行）

架構

目標技術堆疊

- AWS Amplify

- Amazon Cognito

工具

- Amplify 命令列介面 (CLI)
- Amplify 程式庫 (開放原始碼用戶端程式庫)
- Amplify 工作室（視覺界面）

史诗

安裝 AWS Amplify CLI

任務	描述	所需技能
安裝 Amplify CLI。	Amplify CLI 是一個統一的工具鏈，用於為您的 React 應用程式建立 AWS 雲端服務。要安裝 Amplify CLI，請運行： `npm install -g @aws-amplify/cli` npm 將通知您是否有新的主要版本可用。如果是這樣，請使用以下命令升級您的 npm 版本： `npm install -g npm@9.8.0` 其中 9.8.0 指的是您要安裝的版本。	應用程式開發人員

創建一個反應應用

任務	描述	所需技能
創建一個反應應用程序。	要創建一個新的 React 應用程序，請使用以下命令： `npx create-react-app amplify-react-application` 其中`ampify-react-application` 是應用程序的名稱。 成功創建應用程序後，您將看到以下消息： `Success! Created amplify-react-application` 將為 React 應用程式建立具有不同子資料夾的目錄。	應用程式開發人員
在本地計算機上啟動該應用程序。	轉到在上一個步驟中創建的目錄`amplify-react-application` 並運行命令： `amplify-react-application% npm start` 這會在您的本機電腦上啟動 React 應用程式。	應用程式開發人員

設定 Amplify CLI

任務	描述	所需技能
設定 Amplify 以連接到您的 AWS 帳戶。	透過執行以下命令來設定 Amplify： `amplify-react-application % amplify configure` Amplify CLI 會要求您按照下列步驟設定對 AWS 帳戶的存取權限： 1. 登入您的 AWS 管理員帳戶。 2. 指定您要使用的 AWS 區域。 3. 建立具有程式設計存取`AdministratorAccess-Amplify` 權的 AWS Identity and Access Management (IAM) 使用者，並將許可政策附加到使用者。 4. 建立並複製存取金鑰 ID 和秘密存取金鑰。 5. 在終端機中輸入這些詳細信息。 6. 建立設定檔名稱或使用預設設定檔。 警告：此案例需要具有程式設計存取權限和長期登入資料的 IAM 使用者，這會帶來安全	一般 AWS 應用程式開發人員

任務	描述	所需技能
	風險。為了減輕此風險，我們建議您僅向這些使用者提供執行工作所需的權限，並在不再需要這些使用者時移除這些使用者。如有必要，可更新存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的更新存取金鑰。 這些步驟顯示在終端中，如下所示。 `Follow these steps to set up access to your AWS account:` `Sign in to your AWS administrator account: https://console.aws.amazon.com/` `Press Enter to continue` `Specify the AWS Region` `? region: us-east-1` `Follow the instructions at https://docs.amplify.aws/cli/start/install/#configure-the-amplify-cli` `to complete the user creation in the AWS console` `https://console.aws.amazon.com/iamv2/home#/users/create` `Press Enter to continue` `Enter the access key of the newly created user:` `? accessKeyId: **********`	

任務	描述	所需技能
	? secretAccessKey: ********** ********** **** This would update/cr eate the AWS Profile in your local machine ? Profile Name: new Successfully set up the new user. 如需這些步驟的詳細資訊，請參閱 Amplify 開發人員中心中的文件。	

初始化 Amplify

任務	描述	所需技能
初始化 Amplify。	1. 要在新目錄中初始化 Amplify，請運行： amplify init Amplify 提示您輸入項目名稱和配置參數 2. 指定所有參數，然後按 Y 鍵以指定的組態初始化專案。 Project information | Name: amplifyre actproject | Environment: dev	一般 AWS 應用程式開發人員

任務	描述	所需技能
	<pre>| Default editor: Visual Studio Code | App type: javascrip t | Javascript framework: react | Source Directory Path: src | Distribution Directory Path: build | Build Command: npm run-script build | Start Command: npm run-script start</pre>	
	3. 選取您在上一個步驟中建立的設定檔。資源將部署到您建立的 Amplify 專案中的dev環境中。 4. 若要確認已建立資源，您可以開啟 AWS Amplify 主控台並檢視用來建立資源和詳細資料的 AWS CloudFormation 範本。 <pre>Deploying root stack amplifyreactproject [================== ==---------------- ----] 2/4 amplify-amplif yreactproject-d… AWS::CloudFormatio n::Stack</pre>	

任務	描述	所需技能
	CREATE_IN_PROGRESS UnauthRole AWS::IAM::Role CREATE_COMPLETE DeploymentBucket AWS::S3::Bucket CREATE_IN_PROGRESS AuthRole AWS::IAM::Role CREATE_COMPLETE	

將身份驗證添加到前端

任務	描述	所需技能
添加身份驗證。	您可以使用`amplify add <category>` 指令來新增使用者登入或後端 API 等功能。在此步驟中，您將使用命令來添加身份驗證。 Amplify 提供具有 Amazon Cognito、前端程式庫和嵌入式驗證器 UI 元件的後端身份驗證服務。功能包括用戶註冊，用戶登錄，多因素身份驗證，用戶註銷和無密碼登錄。您還可以通過與 Amazon，谷	一般 AWS 應用程式開發人員

任務	描述	所需技能
	歌和 Facebook 等聯合身份提供商集成來對用戶進行身份驗證。Amplify 驗證類別可與其他 Amplify 類別 (例如 API、分析和儲存) 無縫整合，因此您可以為已驗證和未驗證的使用者定義授權規則。 1. 要為您的 React 應用程序配置身份驗證，請運行以下命令： `amplify-react-application1 % amplify add auth` 這會顯示下列資訊和提示。您可以根據業務和安全性需求選擇適當的組態。 `Using service: Cognito, provided by: awscloudformation The current configured provider is Amazon Cognito. Do you want to use the default authentication and security configuration? (Use arrow keys)` `# Default configuration` `Default configuration with Social Provider (Federation)`	

任務	描述	所需技能
	<pre> Manual configura tion I want to learn more.</pre> 2. 對於一個簡單的例子，選擇默認配置，然後選擇用戶的登錄機制（在本例中為電子郵件）： <pre>How do you want users to be able to sign in? Username # Email Phone Number Email or Phone Number I want to learn more.</pre> 3. 略過進階設定以完成新增驗證資源： <pre>Do you want to configure advanced settings? (Use arrow keys) # No, I am done. Yes, I want to make some additional changes.</pre>	

任務	描述	所需技能
	4. 建置您的本機後端資源並在雲端中佈建： `amplify-react-application1 % amplify push` 此命令會對您帳戶中的 Congito 使用者集區進行適當的變更。 5. 按 Y 鍵可使用來配置auth資源 CloudFormation。 這會設定下列資源： `UserPool AWS::Cognito::UserPool CREATE_COMPLETE` `UserPoolClientWeb AWS::Cognito::UserPoolClient CREATE_COMPLETE` `UserPoolClientWeb AWS::Cognito::UserPoolClient CREATE_COMPLETE` `UserPoolClientRole AWS::IAM::Role CREATE_COMPLETE` `UserPoolClientLambda AWS::Lambda::Function CREATE_COMPLETE`	

任務	描述	所需技能
	<pre>UserPoolClientLam bdaPolicy AWS::IAM::Policy CREATE_CO MPLETE UserPoolClientLog Policy AWS::IAM::Policy CREATE_IN _PROGRESS</pre> 您也可以使用 AWS Cognito 主控台來檢視這些資源 (尋找 Cognito 使用者集區和身分集區)。 此步驟會使用 Cognito 使用者集區和身分集區設定來更新 React 應用程式`src`資料夾中的`aws-exports.js` 檔案。	

更改 App.js 文件

任務	描述	所需技能
更改 App.js 文件。	在`src`資料夾中，開啟並修訂`App.js`檔案。修改後的文件應該如下所示： <pre>{ App.Js File after modifications: import React from 'react';</pre>	應用程式開發人員

任務	描述	所需技能
	<pre>import logo from './logo.svg'; import './App.css'; import { Amplify } from 'aws-amplify'; import { withAuthenticator, Button, Heading } from '@aws-amplify/ui-react'; import awsconfig from './aws-exports'; Amplify.configure(awsconfig); function App({ signOut }) { return (Thankyou for doing verification My Content <button onClick={signOut}>Sign out</button>); } export default withAuthenticator(App);</pre>	
導入反應包。	該App.js文件導入兩個反應包。使用以下指令安裝這些套件： <pre>amplify-react-application1 % npm install --save aws-amplify @aws-amplify/ui-react</pre>	應用程式開發人員

啟動 React 應用程序並檢查身份驗證

任務	描述	所需技能
啟動應用程序。	在您的本地機器上啟動 React 應用程序： `amplify-react-application1 % npm start`	一般 AWS 應用程式開發人員
檢查驗證。	檢查應用程式是否提示輸入驗證參數。（在我們的示例中，我們將電子郵件配置為登錄方法。） 前端 UI 應提示您輸入登錄憑據，並提供創建帳戶的選項。 您也可以設定 Amplify 建置程序，將後端新增為持續部署工作流程的一部分。但是，此模式不涵蓋該選項。	一般 AWS 應用程式開發人員

相關資源

- 開始使用（npm 文檔）
- 建立獨立的 AWS 帳戶 (AWS 帳戶管理文件)
- AWS Amplify 文件
- Amazon Cognito 文檔

將反應型單頁應用程式部署到 Amazon S3 和 CloudFront

由讓·巴蒂斯特·吉盧瓦 (AWS) 創建

代碼存儲庫：基於反應的 CORS 單頁應用程序	環境：生產	技術：Web 和行動應用程式；雲端原生；無伺服器
工作負載：所有其他工作	AWS 服務：Amazon CloudFront; Amazon S3; Amazon API Gateway	

Summary

單頁應用程式 (SPA) 是使用 JavaScript API 動態更新顯示網頁內容的網站或 Web 應用程式。這種方法增強了網站的用戶體驗和性能，因為它只更新新數據，而不是從服務器重新加載整個網頁。

這種模式提供了一 step-by-step 種編碼和託管在 Amazon 簡單存儲服務（Amazon S3）和亞馬遜上的 React 編寫的 SPA 的方法 CloudFront。此模式中的 SPA 使用 Amazon API 閘道公開的 REST API，並展示跨來源資源共用 (CORS) 的最佳實務。

先決條件和限制

先決條件

- 作用中的 AWS 帳戶
- 整合式開發環境 (IDE)，例如 AWS Cloud9。
- Node.js 和npm, 已安裝和配置. 如需詳細資訊，請參閱 Node.js 文件的「下載」一節。
- 紗線，安裝和配置。如需詳細資訊，請參閱 Yarn 文件。
- Git，安裝和配置。如需詳細資訊，請參閱 Git 文件。

架構

使用 AWS CloudFormation (基礎設施即程式碼) 自動部署此架構。它使用區域服務 (例如 Amazon S3) 來存放靜態資產，而 Amazon API Gateway 則會公開區域 API (REST) 端點。應用程序日誌是通過

使用 Amazon 收集的 CloudWatch。所有 AWS API 呼叫都會在 AWS 中進行稽核 CloudTrail。所有安全組態 (例如身分和許可) 都在 Amazon Identity and Access Management (IAM)。靜態內容是透過 Amazon 內 CloudFront 容交付網路 (CDN) 傳遞，而 DNS 查詢則由 Amazon Route 53 處理。

技術, 堆

- Amazon API Gateway
- Amazon CloudFront
- Amazon Route 53
- Amazon S3
- IAM
- Amazon CloudWatch
- AWS CloudTrail
- AWS CloudFormation

工具

AWS 服務

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- AWS Cloud9 是一種 IDE，可協助您撰寫程式碼、建置、執行、測試和偵錯軟體。它也可協助您將軟體發行到 AWS 雲端。
- AWS 可 CloudFormation協助您設定 AWS 資源、快速且一致地佈建 AWS 資源，並在 AWS 帳戶和區域的整個生命週期中進行管理。
- Amazon CloudFront 透過全球資料中心網路提供您的 Web 內容，加快網頁內容的分發速度，進而降低延遲並提升效能。
- AWS 可 CloudTrail協助您稽核 AWS 帳戶的管理、合規和營運風險。
- Amazon 可 CloudWatch協助您即時監控 AWS 資源的指標，以及在 AWS 上執行的應用程式。
- AWS Identity and Access Management (IAM) 可透過控制誰經過身份驗證和授權使用 AWS 資源，協助您安全地管理對 AWS 資源的存取。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。
- Amazon Simple Storage Service (Amazon S3) 是一種雲端型物件儲存服務，可協助您儲存、保護和擷取任何數量的資料。

Code

此模式的範例應用程式程式碼可在 GitHub 反應型 CORS 單頁應用程式存放庫中取得。

史诗

在本機建置和部署應用程式

任務	描述	所需技能
複製儲存庫。	我們建議您使用 AWS Cloud9 做為此模式的 IDE，但您也可以使用其他 IDE (例如，視覺工作室程式碼或 IntelliJ IDEA)。 執行下列命令，將範例應用程式的存放庫複製到您的 IDE 中： `git clone https://github.com/aws-samples/react-cors-spa react-cors-spa && cd react-cors-spa`	AWS 應用程式開發人員 DevOps
在本機部署應用程式。	1. 在專案目錄中，執行命`npm install`令以啟動應用程式相依性。 2. 執行命`yarn start`令以在本機啟動應用程式。	AWS 應用程式開發人員 DevOps
在本機存取應用程式。	開啟瀏覽器視窗並輸入 `http://localhost:3000` URL 以存取應用程式。	AWS 應用程式開發人員 DevOps

部署應用程式

任務	描述	所需技能
部署 AWS CloudFormation 範本。	1. 登入 AWS 管理主控台，然後開啟 AWS 主 CloudFormation 控台。 2. 選擇 [建立堆疊]，然後選擇 [使用新資源 (標準)]。 3. 選擇 Upload a template file (上傳範本檔案)。 4. 選擇 [選擇檔案]，從複製的儲存庫中選擇`react-cors-spa-stack.yaml` 檔案，然後選擇 [下一步]。 5. 輸入堆疊的名稱，然後選擇 [下一步]。 6. 保留所有預設選項，然後選擇 [下一步]。 7. 檢閱堆疊的最終設定，然後選擇 [建立堆疊]。	AWS 應用程式開發人員 DevOps
自訂您的應用程式來源檔案。	1. 部署堆疊後，開啟 [輸出] 索引標籤並識別 `APIEndpoint` URL、`Bucket`名稱和`CFDistributionURL` 。 2. 複製 API 端點網址。 3. 導覽至`<project_root>/src/App.js` ，然後將 URL 貼到`App.js`檔案第 26 行的`APIEndPoint` 變數值中。	應用程式開發人員

任務	描述	所需技能
建置應用程式套件。	在您的專案目錄中，執行`yarn build`命令以建置應用程式套件。	應用程式開發人員
部署應用程式套件。	1. 開啟 Amazon S3 主控台。 2. 識別並選擇您之前建立的 S3 儲存貯體。 3. 選擇 [上傳]，然後選擇 [新增檔案]。 4. 選擇構建文件夾的內容。 5. 選擇 [新增資料夾]，然後選擇靜態目錄。重要事項：請勿選擇內容；請選擇目錄。 6. 選擇上傳，將檔案和目錄上傳到 S3 儲存貯體。	AWS 應用程式開發人員 DevOps

測試應用程式。

任務	描述	所需技能
存取並測試應用程式。	開啟瀏覽器視窗，然後貼上 URL (您先前部署之 CloudFormation 堆疊的`CFDistributionURL` 輸出) 以存取應用程式。	AWS 應用程式開發人員 DevOps

清理資源

任務	描述	所需技能
刪除 S3 儲存貯體內容。	1. 開啟 Amazon S3 主控台，然後選擇先前由堆疊建立的儲存貯體 (名稱開頭為的第	AWS 應用程 DevOps式開發人員

任務	描述	所需技能
	一個儲存貯體`react-cors-spa-`)。 2. 選擇「空白」以刪除值區的內容。 3. 選擇先前由堆疊建立的第二個值區 (名稱開頭`react-cors-spa-` 和結尾為的第二個值區`-logs`)。 4. 選擇「空白」以刪除值區的內容。	
刪除 AWS CloudFormation 堆疊。	1. 開啟 AWS 主 CloudFormation 控台，然後選擇您先前建立的堆疊。 2. 選擇刪除以刪除堆疊及所有相關資源。	AWS 應用程 DevOps式開發人員

其他資訊

若要部署和託管 Web 應用程式，您也可以使用 AWS Amplify 託管，它提供以 Git 為基礎的工作流程，用於託管持續部署的全堆疊無伺服器 Web 應用程式。Amplify 託管是 AWS Amplify 的一部分，它提供了一組專門建置的工具和功能，可讓前端 Web 和行動開發人員在 AWS 上快速輕鬆地建置完整堆疊應用程式。

使用私有端點和應用 Application Load Balancer 在內部網站上部署 Amazon API Gateway API

創建者：索拉伯 · 科塔利 (AWS)

環境：生產	技術：Web 和行動應用程式；網路；無伺服器；基礎架構	AWS 服務：Amazon API Gateway；Amazon Route 53；AWS Certificate Manager（ACM）

Summary

此模式說明如何在可從現場部署網路存取的內部網站上部署 Amazon API Gateway API。您將學習如何使用採用私有端點、應用程式負載平衡器、AWS PrivateLink 和 Amazon Route 53 設計的架構，為私有 API 建立自訂網域名稱。此架構可防止使用自訂網域名稱和 Proxy 伺服器來協助在 API 上進行以網域為基礎的路由所造成的意外後果。例如，如果您在不可路由的子網路中部署虛擬私有雲端 (VPC) 端點，則您的網路無法連線到 API Gateway。常見的解決方案是使用自訂網域名稱，然後在可路由子網路中部署 API，但是當 Proxy 組態將流量 (`execute-api.{region}.vpce.amazonaws.com`) 傳遞至 AWS Direct Connect 時，這可能會中斷其他內部網站。最後，此模式可以幫助您滿足使用無法從 Internet 訪問的私有 API 和自定義域名的組織要求。

先決條件和限制

先決條件

- 有效的 AWS 帳戶
- 為您的網站和 API 提供伺服器名稱指示 (SNI) 憑證
- 從現場部署環境到使用 AWS 直接連接或 AWS 站 Site-to-Site VPN 設定的 AWS 帳戶的連線
- 具有對應網域 (例如 domain.com) 的私有託管區域，可從內部部署網路解析，並將 DNS 查詢轉送至 Route 53
- 可從內部部署網路存取的可路由私人子網路

限制

如需負載平衡器、規則和其他資源配額 (先前稱為限制) 的詳細資訊，請參閱 Elastic Load Balancing 說明文件中的應用程式負載平衡器配額。

架構

技術, 堆

- Amazon API Gateway
- Amazon Route 53
- Application Load Balancer
- AWS Certificate Manager
- AWS PrivateLink

目標架構

下圖顯示如何在 VPC 中部署 Application Load Balancer，以根據應用程式負載平衡器接聽程式規則，將 Web 流量導向至網站目標群組或 API Gateway 目標群組。API Gateway 目標群組是 API Gateway 中 VPC 端點的 IP 位址清單。API Gateway 設定為透過其資源策略將 API 設為私有。此原則會拒絕非來自特定 VPC 端點的所有呼叫。API 閘道中的自訂網域名稱會更新為 API 及其階段使用 api.domain.com。會新增 Application Load Balancer 規則，以根據主機名稱路由流量。

該圖顯示以下工作流程：

1. 來自內部部署網路的使用者嘗試存取內部網站。此要求會傳送至網域網站和網域。然後，該請求將解析為可路由私有子網路的內部 Application Load Balancer。SSL 會在應用程式負載平衡器中終止。
2. 接聽程式規則 (在應用程式負載平衡器上設定) 檢查主機標頭。

   a. 如果主機標頭是 api.domain.com，則會將要求轉寄至 API Gateway 目標群組。應用程式負載平衡器會透過連接埠 443 啟動 API Gateway 的新連線。

   b. 如果主機標頭是 ui.domain.com，則會將要求轉寄至網站目標群組。
3. 當要求到達 API Gateway 時，API Gateway 中設定的自訂網域對應會決定主機名稱以及要執行的 API。

自動化和規模

此模式中的步驟可以使用 AWS CloudFormation 或 AWS Cloud Development Kit (AWS CDK) 自動化。若要設定 API Gateway 呼叫的目標群組，您必須使用自訂資源來擷取 VPC 端點的 IP 位址。API 呼叫describe-vpc-endpoints並describe-network-interfaces傳回 IP 位址和安全群組，這些群組可用來建立 IP 位址的 API 目標群組。

工具

- Amazon API Gateway 可協助您建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- Amazon Route 53 是一種可用性高、可擴展性強的 DNS Web 服務。
- AWS Certificate Manager (ACM) 可協助您建立、存放和更新公有和私有 SSL/TLS X.509 憑證和金鑰，以保護您的 AWS 網站和應用程式。
- AWS Cloud Development Kit (AWS CDK) 是一種軟體開發架構，可協助您在程式碼中定義和佈建 AWS 雲端基礎設施。
- AWS 可 PrivateLink協助您建立從 VPC 到 VPC 以外的服務的單向私有連線。

史诗

建立 SNI 憑證

任務	描述	所需技能
建立 SNI 憑證，並將憑證匯入 ACM。	1. 建立網域和網域的 SNI 憑證。如需詳細資訊，請參閱 Amazon CloudFront 文件中的選擇如何處 CloudFront 理 HTTPS 請求。 2. 將 SNI 憑證匯入 AWS Certificate Manager (ACM)。如需詳細資訊，請參閱 ACM 文件中的將憑證匯入 AWS Certificate Manager。	網路管理員

在不可路由的私有子網路中部署 VPC 端點

任務	描述	所需技能
在 API Gateway 中建立介面 VPC 人雲端端點。	若要建立界面 VPC 端點，請按照 Amazon 虛擬私有雲 (Amazon VPC) 文件中的使用界面 VPC 端點存取 AWS 服務的指示進行操作。	雲端管理員

設定應用程式負載平衡器

任務	描述	所需技能
為您的應用程式建立目標群組。	為應用程式的 UI 資源建立目標群組。	雲端管理員
為 API Gateway 端點建立目標群組。	1. 使用 IP 位址類型建立目標群組，然後將 API Gateway 端點之 VPC 端點的 IP 位址新增至目標群組。 2. 使用成功代碼 200 和 403 為目標群組設定健康狀態檢查。需要 403，因為 API 可以使用驗證並傳回 403 回應。	雲端管理員
建立應用程式負載平衡器。	1. 在可路由私有子網路中建立 Application Load Balancer (內部)。 2. 將 443 接聽程式新增至 Application Load Balancer，然後從 ACM 選擇憑證。	雲端管理員
創建監聽器規則。	建立監聽程式規則以執行下列作業：	雲端管理員

任務	描述	所需技能
	1. 將主機 API 網域轉寄至 API Gateway 目標群組 2. 將主機 ui.domain.com 轉寄至 UI 資源的目標群組	

設定 Route 53

任務	描述	所需技能
建立私有託管區域。	為網域 .com 建立私人託管區域。	雲端管理員
建立網域記錄。	為下列項目建立 CNAME 記錄： • 值設定為應用程式負載平衡器 DNS 名稱的 API • 值設定為應用程式負載平衡器 DNS 名稱的 UI	雲端管理員

在 API Gateway 中建立私有 API 端點

任務	描述	所需技能
建立和設定私有 API 端點。	1. 若要建立私有 API 端點，請按照 API Gateway 文件中的在 Amazon API 閘道中建立私有 API 中的指示進行操作。 2. 將資源策略設定為僅允許從 VPC 端點呼叫 API。如需詳細資訊，請參閱 API Gateway 說明文件中的使用	應用程式開發人員、雲端

任務	描述	所需技能
	API Gateway 資源政策控制 API 的存取。	
建立自訂網域名稱。	1. 為網域建立自訂網域名稱。如需詳細資訊，請參閱 API Gateway 文件中的設定 REST API 的自訂網域名稱。 2. 選擇創建的 API 和階段。如需詳細資訊，請參閱 API Gateway 文件中的使用 REST API 的 API 對應。	雲端管理員

相關資源

- Amazon API Gateway
- Amazon Route 53
- Application Load Balancer
- AWS PrivateLink
- AWS Certificate Manager

在本地角度應用程序中嵌入 Amazon QuickSight 儀表板

由肖恩·格里芬（AWS）和米萊娜戈道（AWS）創建

環境：PoC 或試點	技術：網絡和移動應用程序; 分析	AWS 服務：AWS Lambda；Amazon QuickSight；Amazon API Gateway

Summary

此模式提供將 Amazon QuickSight 儀表板嵌入本機託管的 Angular 應用程式以進行開發或測試的指引。中的內嵌式分析功能本身 QuickSight 不支援此功能。它需要一個具有現有儀表板和 Angular 知識的 QuickSight 帳戶。

使用內嵌 QuickSight 儀表板時，通常必須在 Web 伺服器上託管應用程式才能檢視儀表板。這使得開發變得更加困難，因為您必須不斷地將更改推送到 Web 服務器，以確保一切正常運行。此模式顯示如何執行本機託管的伺服器，並使用 QuickSight 內嵌式分析來讓開發程序更輕鬆、更簡化。

先決條件和限制

先決條件

- 有效的 Amazon Web Services (AWS) 帳戶
- 工作階段容量定價的作用中 QuickSight 帳戶
- QuickSight 已安裝嵌入 SDK
- 角度 CLI 已安裝
- 熟悉角
- 已安裝

限制

- 此模式提供使用 ANONYMOUS (可公開存取) 驗證類型嵌入 QuickSight 儀表板的指引。如果您在嵌入式儀表板上使用 AWS Identity and Access Management (IAM) 或身份 QuickSight 驗證，則提供的程式碼將不適用。但是，在「史詩」部分中託管 Angular 應用程序的步驟仍然有效。

- 將 GetDashboardEmbedUrlAPI 與ANONYMOUS身分類型搭配使用需要 QuickSight 容量定價方案。

版本

- 角度 CLI 版本
- QuickSight 嵌入式開發套件 2.3.1 版

架構

技術, 堆

- 角度前端
- AWS Lambda 和 Amazon API Gateway 後端

架構

在此架構中，API Gateway 中的 HTTP API 可讓本機角度應用程式呼叫 Lambda 函數。Lambda 函數返回用於嵌入 QuickSight 儀表板的 URL。

自動化和規模

您可以使用 AWS CloudFormation 或 AWS Serverless Application Model (AWS SAM) 自動化後端部署。

工具

工具

- Angular CLI 是一個命令行界面工具，您可以用它來初始化，開發，腳手架，並直接從命令外殼維護角度的應用程序。
- QuickSight 內嵌 SDK 可用來將 QuickSight 儀表板內嵌到您的 HTML 中。
- mkcert 是建立本機信任開發憑證的簡單工具。它不需要配置。mkcert 是必需的，因為只 QuickSight 允許 HTTPS 請求嵌入儀表板。

AWS 服務

- Amazon API Gateway 是一種 AWS 服務，用於建立、發佈、維護、監控和保護任何規模的 REST、HTTP 和 WebSocket API。
- AWS Lambda 是一種運算服務，可支援執行程式碼，而無需佈建或管理伺服器。Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。只需為使用的運算時間支付費用，一旦未執行程式碼，就會停止計費。
- Amazon QuickSight 是一種商業分析服務，可用於建置視覺化、執行臨機操作分析，以及從資料中取得商業洞察。

史诗

生成安全

任務	描述	所需技能
建立 EmbedUrl 策略。	建立具有下列屬性QuicksightGetDashboardEmbedUrl的名稱為的 IAM 政策。 `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "quicksight:GetDashboardEmbedUrl", "quickSight:GetAnonymousUserEmbedUrl" ], "Resource": "*" } ] }`	AWS 管理員

任務	描述	所需技能
建立 Lambda 函數。	1. 在 Lambda 主控台上，開啟「函數」頁面。 2. 選擇 Create Function (建立函數)。 3. 選擇從頭開始撰寫。 4. 針對 函數名稱 ，請輸入 `get-qs-embed-url` 。 5. 針對 Runtime (執行階段)，選擇 Python 3.9。 6. 選擇 Create Function (建立函數)。 7. 在 [程式碼] 索引標籤上，將下列程式碼複製到 Lambda 函數中。 `import json` `import boto3` `from botocore.exceptions import ClientError` `import time` `from os import environ` `qs = boto3.client('quicksight',region_name='us-east-1')` `sts = boto3.client('sts')` `ACCOUNT_ID = boto3.client('sts').get_caller_identity().get('Account')`	應用程式開發人員

任務	描述	所需技能
	DASHBOARD_ID = environ['DASHBOARD _ID'] def getDashboardURL(ac countId, dashboardId, quicksightNamespac e, resetDisabled, undoRedoDisabled): try: response = qs.get_da shboard_embed_url(AwsAccountId = accountId, DashboardId = dashboardId, Namespace = quicksightNamespace, IdentityType = 'ANONYMOUS', SessionLi fetimeInMinutes = 600, UndoRedoDisabled = undoRedoDisabled, ResetDisabled = resetDisabled) return response except ClientError as e: print(e) return "Error generating embeddedU RL: " + str(e) def lambda_handler(eve nt, context): url = getDashbo ardURL(ACCOUNT_ID, DASHBOARD_ID,	

任務	描述	所需技能
	<pre>"default", True, True) ['EmbedUrl'] return { 'statusCode': 200, 'url': url }</pre> 8. 選擇部署。	
將儀表板 ID 新增為環境變數。	DASHBOARD_ID 作為環境變量添加到您的 Lambda 函數中： 1. 在組態索引標籤上，選擇環境變數、編輯、新增環境變數。 2. 使用密鑰添加環境變量DASHBOARD_ID 。 3. 若要取得的值DASHBOARD_ID ，請瀏覽至中的儀表板， QuickSight 然後在瀏覽器中複製 URL 結尾的 UUID。例如，如果 URL 是https://us-east-1.quicksight.aws.amazon.com/sn/dashboards/<dashboard-id> ，請將 URL 的一<dashboard-id> 部分指定為索引鍵值。 4. 選擇儲存。	應用程式開發人員

任務	描述	所需技能
新增 Lambda 函數的權限。	修改 Lambda 函數的執行角色，並將QuicksightGetDashboardEmbedUrl原則新增至其中。 1. 在 [設定] 索引標籤上，選擇 [權限]，然後選擇角色名稱。 2. 選擇 [附加原則]、[搜尋] `QuicksightGetDashboardEmbedUrl` 、選取其核取方塊，然後選擇 [附加原則]。	應用程式開發人員

任務	描述	所需技能
測試 Lambda 函數。	創建並運行測試事件。您可以使用「Hello World」範本，因為函式不會在測試事件中使用任何資料。 1. 選擇測試標籤。 2. 為測試事件命名，然後選擇 [儲存]。 3. 若要測試 Lambda 函數，請選擇 [測試]。回應看起來應該類似以下的內容。 `{ "statusCode": 200, "url": "\"https://us-east-1.quicksight.aws.amazon.com/embed/f1acc0786687783b9a4543a05ba929b3a/dashboards/... }` 注意：如 [必要條件和限制] 區段所述，您的 QuickSight 帳戶必須屬於工作階段容量定價方案。否則，此步驟將顯示錯誤消息。	應用程式開發人員

任務	描述	所需技能
在 API Gateway 中建立 API。	1. 在 API Gateway 主控台上，選擇 [建立 API]，然後選擇 [REST API]、[建置]。 • 對於 API 名稱，請輸入`qs-embed-api` 。 • 選擇建立 API。 2. 在「動作」 中選擇「建立方法」。 • 選擇 GET 並通過選擇複選標記進行確認。 • 選擇 Lambda 函數做為整合類型。 • 對於 Lambda 函數，請輸入`get-qs-embed-url`。 • 選擇儲存。 • 在 [新增權限至 Lambda 函數] 方塊中，選擇 [確定]。 3. 啟用 CORS。 • 在 [動作] 中，選擇 [啟用 CORS]。 • 對於存取控制允許來源，請輸入。`'https://my-qs-app.net:4200'` • 選擇啟用 CORS 並替換現有的 CORS 標頭，然後確認。 4. 部署應用程式介面。 • 針對「動作」，選擇「部署 API」。	應用程式開發人員

任務	描述	所需技能
	• 針對 Deployment stage (部署階段)，選擇 [New Stage] ([新增階段])。 • 針對 Stage name (階段名稱)，輸入 dev。 • 選擇 Deploy (部署)。 • 複製呼叫網址。 注意：my-qs-app.net 可以是任何域。如果您想要使用不同的網域名稱，請務必在步驟 3 中更新存取控制允許來源資訊，並在後續步驟中進行變更。my-qs-app.net	

創建角度應用程序

任務	描述	所需技能
創建與角度 CLI 的應用程序。	1. 建立應用程式。 `ng new quicksight-app --defaults` `cd quicksight-app/src/app` 2. 建立儀表板元件。 `ng g c dashboard` 3. 導航到您的src/environments/environment.ts 文件並添加apiUrl: '<Invoke URL from	應用程式開發人員

任務	描述	所需技能
	previous steps>'到環境對象。 `export const environment = { production: false, apiUrl: '<Invoke URL from previous steps>', };`	
新增內 QuickSight 嵌 SDK。	1. 通過在項目的根文件夾中運行以下命令來安裝 QuickSight 嵌入 SDK。 `npm i amazon-quicksight-embedding-sdk` 2. 在具有以下內容的decl.d.ts 文件src 夾中創建一個新文件。 `declare module 'amazon-quicksight-embedding-sdk';`	應用程式開發人員

任務	描述	所需技能
將代碼添加到儀表板。組件 .ts 文件。	`import { Component, OnInit } from '@angular/core';` `import { HttpClient } from '@angular/common/http';` `import * as QuicksightEmbedding from 'amazon-quicksight-embedding-sdk';` `import { environment } from "../../environments/environment";` `import { take } from 'rxjs';` `import { EmbeddingContext } from 'amazon-quicksight-embedding-sdk/dist/types';` `import { createEmbeddingContext } from 'amazon-quicksight-embedding-sdk';` `@Component({` `  selector: 'app-dashboard',` `  templateUrl: './dashboard.component.html',` `  styleUrls: ['./dashboard.component.scss']` `})` `export class DashboardComponent implements OnInit {` `  constructor(private http: HttpClient) { }`	應用程式開發人員

任務	描述	所需技能
	<pre> loadingError = false; dashboard: any; ngOnInit() { this.GetDashboardU RL(); } public GetDashbo ardURL() { this.http.get(envi ronment.apiUrl) .pipe(take(1),) .subscribe((data: any) => this.Dash board(data.url)); } public async Dashboard (embeddedURL: any) { var containerDiv = document.getElemen tById("dashboardCo ntainer") || ''; const frameOptions = { url: embeddedURL, container: containerDiv, height: "850px", width: "100%", resizeHei ghtOnSizeChangedEv ent: true, } const embedding Context: Embedding Context = await createEmbeddingCon text();</pre>	

任務	描述	所需技能
	<pre>this.dashboard = embeddingContext.embedDashboard(frameOptions); } }</pre>	
將代碼添加到儀表板組件 .html 文件中。	將以下代碼添加到您的`src/app/dashboard/dashboard.component.html` 文件中。 <pre> </pre>	應用程式開發人員
修改您的應用程序 .Component.html 文件以加載儀表板組件。	1. 刪除`src/app/app.component.html` 檔案的內容。 2. 添加以下內容。 <pre><app-dashboard></app-dashboard></pre>	應用程式開發人員
導 HttpClientModule 入到您的應用程序 .modu.ts 文件。	1. 在`src/app/app.module.ts` 檔案頂端，新增下列項目。 <pre>import { HttpClientModule } from '@angular/common/http';</pre> 2. 在**imports**數組`HttpClientModule` 中添加您的`AppModule` 。	應用程式開發人員

託管角度應用程序

任務	描述	所需技能
配置安全證書。	注意：以下命令適用於 Unix 或 MacOS 機器。如果您使用的是 Windows，請參閱相應的 echo 命令的其他資訊一節。 1. 在您的機器上建立本機憑證授權單位 (CA)。 `mkcert -install` 2. 配置`my-qs-app.net` 為始終重定向到您的本地 PC。 `echo "127.0.0.1 my-qs-app.net" | sudo tee -a /private/etc/hosts` 3. 確保您位於 Angular 項`src`目的目錄中。 `mkcert my-qs-app.net 127.0.0.1`	應用程式開發人員
設定 QuickSight 為允許您的網域。	1. 在中 QuickSight，請在右上角選擇您的名稱，然後選擇「管理 Quicksight」。 2. 導航到「域和嵌入」。 3. 新增`https://my-qs-app.net:4200` 為允許的網域。	AWS 管理員
測試解決方案。	通過運行以下命令啟動本地 Angular 開發服務器。	應用程式開發人員

任務	描述	所需技能
	`ng serve --host my-qs-app.net --port 4200 --ssl --ssl-key "./src/my-qs-app.net-key.pem" --ssl-cert "./src/my-qs-app.net.pem" -o` 這會使用您先前建立的自訂憑證啟用安全通訊端層 (SSL)。 構建完成後，它會打開一個瀏覽器窗口，您可以查看在 Angular 中本地託管的嵌入式 QuickSight 儀表板。	

相關資源

- 角網站
- 為匿名 (未註冊) 使用者嵌入 QuickSight 資料儀表板 (QuickSight 說明文件)
- QuickSight 嵌入式 SDK
- 電子證書工具

其他資訊

如果您使用的是 Windows，請以系統管理員身分執行 [命令提示字元] 視窗，並使用下列命令設定`my-qs-app.net`為永遠重新導向至本機電腦。

```
echo 127.0.0.1 my-qs-app.net >> %WINDIR%\System32\Drivers\Etc\Hosts
```

更多模式

- 使用 Amazon Cognito 可身分集區從 ASP.NET 核心應用程式存取 AWS 服務
- 使用 AWS Fargate、AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用 AWS PrivateLink 和 Network Load Balancer，在 Amazon ECS 上私下存取容器應用程式
- 使用自動化遷移策略識別和規劃 AppScore
- 使用 DevOps 實務和 AWS Cloud9 建立鬆散結合的架構與微型服務
- 使用 AWS Amplify 建置無伺服器反應原生行動應用程式
- 使用 AWS CodeCommit、AWS 和 AWS Device Farm 建置和測試 iOS 應用程式 CodePipeline
- 使用 NLog 在 Amazon CloudWatch 日誌中設定 .NET 應用程式的記錄
- ???
- 使用建立管道並將成品更新部署到現場部署 EC2 執行個體 CodePipeline
- 使用 Amazon EFS 建立 Amazon ECS 任務定義，並在 EC2 執行個體上掛接檔案系統
- 在 Amazon EKS 叢集上部署以 gRPC 為基礎的應用程式，並使 Application Load Balancer 存取
- 使用地形部署 CloudWatch Synthetics 金絲雀
- 使用 Amazon ECR 和 AWS Fargate 在 Amazon ECS 上部署 Java 微服務
- 使用 Amazon ECR 和負載平衡在 Amazon ECS 上部署 Java 微服務
- 使用 AWS Fargate 在 Amazon ECS 上部署 Java 微服務
- 透過 Green Boost 探索全堆疊雲端原生 Web 應用程式開發
- 將簡訊佇列從 Microsoft Azure 服務匯流排遷移到 Amazon SQS
- 將 .NET 應用程式從 Microsoft Azure 應用程式服務遷移到 AWS Elastic Beanstalk
- 使用二進位方法將現場部署 Go Web 應用程式遷移到 AWS Elastic Beanstalk
- 使用適用於 SFTP 的 AWS 轉移，將現場部署 SFTP 伺服器遷移到 AWS
- 從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的阿帕奇 Tomcat
- 使用 Auto Scaling 能從 IBM WebSphere 應用程序服務器遷移到 Amazon EC2 上的 Apache Tomcat
- 從甲骨文遷移 GlassFish 到 AWS Elastic Beanstalk
- 使用 AWS 應用程式容器將現場部署 Java 應用程式遷移到 AWS
- 將 OpenText TeamSite 工作負載遷移到 AWS 雲端
- 使用 ACM 將視窗 SSL 憑證移轉至應用程式負載平衡器
- 在 AWS 上將 ASP.NET 網頁表單應用程式現代化

- 在 Amazon EC2 Linux 實例上運行一個 ASP.NET 核心網絡 API 碼頭容器
- 使用 Amazon 通過 VPC 在 Amazon S3 存儲桶中提供靜態內容 CloudFront
- 在 AWS 上設定高可用性 PeopleSoft 架構
- 使用 Network Firewall 從輸出流量的伺服器名稱指示 (SNI) 擷取 DNS 網域名稱
- ???

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。