aws

開發人員指南

# AWS Data Pipeline

API 版本 2012-10-29

# AWS Data Pipeline: 開發人員指南

# Table of Contents

AWS Data Pipeline 不再提供給新客戶。的現有客戶 AWS Data Pipeline 可繼續正常使用此服務。進一步了解

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。

# 什麼是 AWS Data Pipeline？

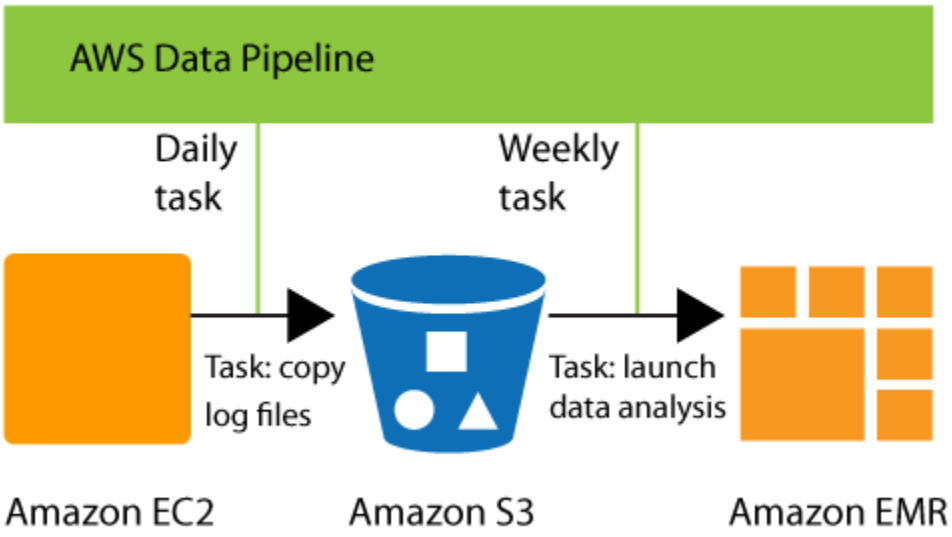
> **Note**
>
> AWS Data Pipeline 服務處於維護模式，並且沒有計劃新功能或區域擴展。若要深入瞭解並瞭解如何移轉現有工作負載，請參閱從移轉工作負載 AWS Data Pipeline。

AWS Data Pipeline 是一種 Web 服務，您可以使用它來自動化數據的移動和轉換。使用 AWS Data Pipeline，您可以定義資料驅動的工作流程，以便任務可以取決於先前任務的成功完成。您可以定義資料轉換的參數，並 AWS Data Pipeline 強制執行已設定的邏輯。

以下元件共同 AWS Data Pipeline 運作來管理您的資料：

- 「管道定義」指定您資料管理的商業邏輯。如需詳細資訊，請參閱 管線定義檔案語法。
- 管道透過建立 Amazon EC2 執行個體來執行已定義的工作活動來排程和執行任務。您要將您的管道定義上傳到管道，然後啟動管道。您可以編輯管道定義以執行管道，並再次啟動管道讓它生效。您可以停用管道、修改資料來源，然後再次啟動管道。管道完成後，即可刪除。
- 任務運行器輪詢任務，然後執行這些任務。例如，任務執行器可以將日誌檔案複製到 Amazon S3，然後啟動 Amazon EMR 叢集。Task Runner 會安裝並在管線定義建立的資源上自動執行。您可以撰寫自訂工作執行程式應用程式，也可以使用由提供的工作執行程式應用程式 AWS Data Pipeline。如需詳細資訊，請參閱 任務執行器。

例如，您可以使用每天將 Web 伺服器的日誌存檔 AWS Data Pipeline 到 Amazon Simple Storage Service (Amazon S3)，然後在這些日誌上執行每週 Amazon EMR (Amazon EMR) 叢集以產生流量報告。 AWS Data Pipeline 排程每日任務以複製資料，並排定每週任務以啟動 Amazon EMR 叢集。AWS Data Pipeline 還可確保 Amazon EMR 在開始分析之前等待最後一天的資料上傳到 Amazon S3，即使上傳日誌有不可預見的延遲也是如此。




目錄



# 從移轉工作負載 AWS Data Pipeline

AWS 於二零一二年推出該 AWS Data Pipeline 服務。當時，客戶正在尋找一種服務，以幫助他們使用各種運算選項在不同的資料來源之間可靠地移動資料。現在，還有其他服務可以為客戶提供更好的體驗。例如，您可以使用 AWS Glue 來執行和協調 Apache Spark 應用程式、協助協調 AWS 服務元件的 AWS Step Functions，或使用 Amazon 管理的 Apache 氣流工作流程 (Amazon MWAA) 來協助管理 Apache 氣流的工作流程協調。

本主題說明如何從其他選項移轉 AWS Data Pipeline 至其他選項。您選擇的選項取決於您目前的工作負載 AWS Data Pipeline。您可以將的典型使用案例遷移 AWS Data Pipeline 到 AWS Step Functions 或 Amazon MWAA。 AWS Glue

# 將工作負載移轉 AWS Glue

AWS Glue 是無伺服器資料整合服務，讓分析使用者可從多個來源輕鬆探索、準備、移動和整合資料。它包括用於編寫、執行工作和協調工作流程的工具。使用 AWS Glue，您可以探索並連接 70 多種不同的資料來源，並在集中式資料目錄中管理您的資料。您可以直觀地建立、執行和監控擷取、轉換和載入 (ETL) 管道，以將資料載入資料湖。此外，您還可以使用 Amazon Athena，Amazon EMR 和 Amazon Redshift Spectrum 立即搜尋和查詢已編目的資料。

我們建議您將 AWS Data Pipeline 工作負載移轉至下列 AWS Glue 時機

- 您正在尋找支援各種資料來源的無伺服器資料整合服務、包括視覺化編輯器和筆記本在內的編寫介面，以及進階資料管理功能，例如資料品質和敏感資料偵測。
- 您可以將工作負載移轉至工作 AWS Glue 流程、工作 (使用 Python 或 Apache Spark) 和編目器 (例如，您現有的管道建立在 Apache Spark 之上)。
- 您需要能夠處理資料管線各個層面的單一平台，包括擷取、處理、傳輸、完整性測試和品質檢查。
- 您現有的管道是從 AWS Data Pipeline 主控台上預先定義的範本建立的，例如將 DynamoDB 表匯出到 Amazon S3，而且您正在尋找相同的用途範本。
- 您的工作負載不依賴於特定的 Hadoop 生態系統應用程式，例如 Apache Hive。
- 您的工作負載不需要協調內部部署伺服器。

AWS 針對搜尋器 (探查資料) 和 ETL 工作 (處理和載入資料)，按小時費率計費，以秒計費。 AWS Glue Studio 是用於 AWS Glue 資源的內建協調引擎，而且不需額外付費。在定價中進一步了解AWS Glue 定價。

# 將工作負載移轉至 AWS Step Functions

AWS Step Functions 是一種無伺服器協調服務，可讓您為業務關鍵應用程式建置工作流程。透過 Step Functions，您可以使用視覺化編輯器建立工作流程，並直接整合超過 250 種 AWS 服務的 11,000 個動作，例如 AWS Lambda、Amazon EMR、DynamoDB 等。您可以使用 Step Functions 來協調資料處理管線、處理錯誤，以及處理基礎服務的節流限制。 AWS 您可以建立工作流程來處理和發佈機器學習模型、協調微型服務，以及控制 AWS 服務，例如建立擷取 AWS Glue、轉換和載入 (ETL) 工作流程。也可以為需要人為互動的應用程式建立長時間執行的自動化工作流程。

同樣 AWS Data Pipeline， AWS Step Functions 是由提供的完全託管服務 AWS。您不需要管理基礎結構、修補程式背景工作程式、管理作業系統版本更新或類似項目。

我們建議在以下情況將 AWS Data Pipeline 工作負載移轉至 AWS Step Functions

- 您正在尋找無伺服器、高可用性的工作流程協調服務。
- 您正在尋找符合成本效益的解決方案，以單一任務執行的精細程度收費。
- 您的工作負載正在為多個其他 AWS 服務 (例如 Amazon EMR、Lambda 或 DynamoDB) 協調任務。AWS Glue
- 您正在尋找一種低代碼解決方案，該解決方案隨附可 drag-and-drop 視化設計器來創建工作流程，並且不需要學習新的編程概念。
- 您正在尋找一種服務，該服務提供與 250 多種其他 AWS 服務的集成，涵蓋 11,000 多個操作 out-of-the-box，並允許與自定義非AWS 服務和活動集成。

AWS Data Pipeline 和 Step Functions 都使用 JSON 格式來定義工作流程。這允許將您的工作流程存儲在源代碼控制中，管理版本，控制訪問以及使用 CI/CD 自動化。Step Functions 使用稱為 Amazon 狀態語言的語法，該語法完全基於 JSON，並允許在工作流程的文本和視覺表示之間進行無縫轉換。

使用 Step Functions，您可以選擇目前使用的相同版本的 Amazon EMR。 AWS Data Pipeline

若要移轉 AWS Data Pipeline 受管資源上的活動，您可以使用 Step Functions 上的 AWS SDK 服務整合來自動化資源佈建和清理。

若要移轉現場部署伺服器、使用者管理的 EC2 執行個體或使用者管理的 EMR 叢集上的活動，您可以將 SSM 代理程式安裝到執行個體。您可以從 Step Functions 通過系 AWS Systems Manager 運行命令啟動命令。您也可以從 Amazon 中定義的排程啟動狀態機器 EventBridge。

AWS Step Functions 有兩種類型的工作流程：標準工作流程和快速工作流程。對於標準工作流程，我們會根據執行應用程式所需的狀態轉換次數向您收費。對於 Express 工作流程，系統會根據工作流程的要求數目及其持續時間向您收費。在AWS 步驟函數定價中進一步了解定價。

## 將工作負載遷移到 Amazon MWAA

Amazon MWAA (Apache 氣流的受管工作流程) 是 Apache 氣流的受管協調服務，可讓您更輕鬆地在雲端中大規模設定和操作 end-to-end 資料管道。Apache Airflow 是一種開放原始碼工具，用於以程式設計方式撰寫、排程和監視稱為「工作流程」的程序和工作序列。使用 Amazon MWAA，您可以使用 Airflow 和 Python 程式設計語言建立工作流程，而不必管理基礎設施以提高可擴展性、可用性和安全性。Amazon MWAA 會自動擴展其工作流程執行容量以滿足您的需求，並與 AWS 安全服務整合，協助您快速安全地存取資料。

同樣地 AWS Data Pipeline，Amazon MWAA 是由提供的全受管服務。 AWS雖然您需要瞭解這些服務特定的幾個新概念，但不需要管理基礎結構、修補程式背景工作程式、管理作業系統版本更新或類似項目。

我們建議您在以下情況將 AWS Data Pipeline 工作負載遷移到 Amazon MWAA：

- 您正在尋找受管理的高可用性服務來協調使用 Python 編寫的工作流程。
- 您想要轉換成全受管、廣泛採用的開放原始碼技術 Apache Airflow，以獲得最大的可攜性。
- 您需要能夠處理資料管線各個層面的單一平台，包括擷取、處理、傳輸、完整性測試和品質檢查。
- 您正在尋找專為資料管線協調流程而設計的服務，其功能包括可觀察性的豐富 UI、針對失敗的工作流程重新啟動、回填，以及重試工作。
- 您正在尋找一種包含 800 多個預先構建的操作員和傳感器的服務，涵蓋 AWS 以及非AWS 服務。

Amazon MWAA 工作流程會使用 Python 定義為有向無環圖 (DAG)，因此您也可以將它們視為原始程式碼。Airflow 的可擴展 Python 框架使您能夠構建與幾乎任何技術相連的工作流程。它配備了用於查看和監控工作流程的豐富用戶界面，並且可以輕鬆地與版本控制系統集成以自動化 CI/CD 流程。

使用 Amazon MWAA，您可以選擇與當前使用的相同版本的 Amazon EMR。 AWS Data Pipeline

AWS 依 Airflow 環境執行時間計費，再加上任何額外的 auto 擴充功能，以提供更多工作者或 Web 伺服器容量。進一步了解 Amazon 受管工作流程的 Apache 氣流定價。

## 映射概念

下表包含由服務使用的主要概念的映射。它將幫助熟悉 Data Pipeline 的人們了解 Step Functions 數和 MWAA 術語。

| Data Pipeline | 連接詞 | Step Functions | Amazon 分公司 |
|---|---|---|---|
| 管道 | 工作流 | 工作流 | 直接丙烯酸圖 |
| 管道定義 | 工作流程定義或以 Python 為基礎的藍圖 | Amazon 國家語言 JSON | 基於蟒蛇 |
| 活動 | 任務 | 狀態與工作 | 任務（操作員和傳感器） |
| 執行個體 | Job 執行 | 執行 | DAG 運行 |
| Attempts | 重試嘗試 | 捕手和取回器 | 重試 |
| 管線排程 | 排程觸發 | EventBridge 排程器工作 | Cron 、時間表、資料感知 |

| Data Pipeline | 連接詞 | Step Functions | Amazon 分公司 |
| --- | --- | --- | --- |
| 管線運算式和函數 | 藍圖程式庫 | Step Functions 內在函數和 Lambda AWS | 可擴展的 Python 框架 |

# 範例

以下各節列出了您可以參考從移轉 AWS Data Pipeline 至個別服務的公開範例。您可以將它們稱為示例，並根據您的使用案例更新和測試，在個別服務上建立自己的管道。

## AWS Glue 樣本

下列清單包含最常見 AWS Data Pipeline 使用案例的範例實作。 AWS Glue

- 運行星火工作
- 將資料從 JDBC 複製到 Amazon S3 (包括 Amazon Redshift)
- 將資料從 Amazon S3 複製到 JDBC (包括 Amazon Redshift)
- 將資料從 Amazon S3 複製到
- 將資料移入和移出 Amazon Redshift
- 跨帳戶跨區域存取 DynamoDB 資料表

## AWS Step Functions 示例

下列清單包含 AWS Step Functions 最常見 AWS Data Pipeline 使用案例的範例實作。

- 管理 Amazon EMR 任務
- 在 Amazon EMR 無伺服器上執行資料處理任務
- 運行/豬/Hadoop 工作
- 查詢大型資料集 (Amazon Athena、Amazon S3、 AWS Glue)
- 使用 Amazon Redshift 執行 ETL 工作流程
- 編排爬蟲 AWS Glue

請參閱使用 AWS Step Functions 的其他教學課程和範例專案。

### Amazon MWAA 樣品

下列清單包含 Amazon MWAA 最常見 AWS Data Pipeline 使用案例的範例實作。

- 執行 Amazon EMR 任務
- 創建阿帕奇蜂巢和 Hadoop 的自定義插件
- 將資料從 Amazon S3 複製到 Redshift
- 在遠端 EC2 執行個體上執行殼層指令碼
- 協調混合式 (內部部署) 工作流程

請參閱使用 Amazon MWAA 的其他教學課程和範例專案。

## 相關服務

AWS Data Pipeline 與以下服務一起使用以存儲數據。

- Amazon DynamoDB — 以低成本提供具有快速效能的全受管 NoSQL 資料庫。如需詳細資訊，請參閱 Amazon DynamoDB 開發人員指南。
- Amazon RDS — 提供可擴展至大型資料集的全受管關聯式資料庫。有關詳情，請參閱 Amazon Relational Database Service 開發人員指南。
- Amazon Redshift — 提供快速、全受管的 PB 級資料倉儲，可輕鬆且經濟實惠地分析大量資料。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南。
- Amazon S3 — 提供安全、耐用且可高度擴展的物件儲存。如需詳細資訊，請參閱 Amazon 簡易儲存服務使用者指南。

AWS Data Pipeline 與下列運算服務搭配使用以轉換資料。

- Amazon EC2 — 提供可調整大小的運算容量 (實際上是 Amazon 資料中心中的伺服器)，可用來建置和託管軟體系統。如需詳細資訊，請參閱 Amazon EC2 使用者指南。
- 亞馬遜 EMR — 使您可以使用 Apache Hadoop 或 Apache Spark 之類的框架，輕鬆、快速且符合成本效益，在 Amazon EC2 伺服器上分發和處理大量資料。如需詳細資訊，請參閱 Amazon EMR 開發人員指南。

# 存取 AWS Data Pipeline

您可以使用下列任一界面來建立、存取和管理您的管道：

- AWS Management Console— 提供可用於訪問的 Web 界面 AWS Data Pipeline。
- AWS Command Line Interface (AWS CLI) — 為一組廣泛的 AWS 服務提供命令 AWS Data Pipeline，包括 Windows、macOS 和 Linux 上並受到支援。如需有關安裝的更多資訊 AWS CLI，請參閱AWS Command Line Interface。如需用於的指令清單 AWS Data Pipeline，請參閱資料副本。
- AWS 開發套件 — 提供語言特定 API，並處理許多連線詳細資訊，例如計算簽章、處理請求重試和錯誤處理。如需詳細資訊，請參閱 AWS 開發套件。
- 查詢 API — 提供您使用 HTTPS 要求呼叫的低階 API。使用查詢 API 是存取 AWS Data Pipeline最直接的方式，但這需要您的應用程式處理低階詳細資訊，例如產生雜湊以簽署請求以及錯誤處理。如需詳細資訊，請參閱 AWS Data Pipeline API 參考。

# 定價

使用 Amazon Web Services，您只需按實際用量付費。對於 AWS Data Pipeline，您可以根據活動和先決條件排定執行的頻率以及它們的執行位置來支付管道費用。如需詳細資訊，請參閱 AWS Data Pipeline 定價。

如果您的 AWS 帳戶不超過 12 個月，您符合免費方案的使用資格。免費方案包含每月免費的 3 個低頻率先決條件和 5 個低頻率活動。如需詳細資訊，請參閱 AWS 免費方案。

# 管道工作活動支援的執行個體類型

執 AWS Data Pipeline 行管道時，它會編譯管道元件以建立一組可操作的 Amazon EC2 執行個體。每個執行個體包含執行特定任務的所有資訊。完整的執行個體集是管道的待辦事項清單。 AWS Data Pipeline 會將執行個體分給任務執行器處理。

EC2 執行個體提供不同的組態，這些組態稱為執行個體類型。每個執行個體類型都有不同的 CPU、輸入/輸出和儲存容量。除了指定活動的執行個體類型以外，您還可以選擇不同的購買選項。並非所有的 AWS 區域皆提供所有的執行個體類型。如果沒有執行個體類型可用，您的管道佈建可能會失敗，或停滯不前。如需執行個體可用性的相關資訊，請參閱 Amazon EC2 定價頁面。開啟您的執行個體購買選項連結，依 Region (區域) 篩選，查看該區域是否提供可用的執行個體類型。如需這些執行個體類型、系列和虛擬化類型的詳細資訊，請參閱 Amazon EC2 執行個體和 Amazon Linux AMI 執行個體類型對照表。

下表說明 AWS Data Pipeline 支援的執行個體類型。您可以使用 AWS Data Pipeline 在任何區域啟動 Amazon EC2 執行個體，包括不 AWS Data Pipeline 受支援的區域。如需支援區域的 AWS Data Pipeline 相關資訊，請參閱 AWS 區域和端點。

目錄



## AWS 區域的預設 Amazon EC2 執行個體

根據預設，如果不在管道定義中指定執行個體類型， AWS Data Pipeline 就會啟動執行個體。

下表列出在受支援的區域中預設 AWS Data Pipeline 使用的 Amazon EC2 執行 AWS Data Pipeline 個體。

| 區域名稱 | 區域 | 執行個體類型 |
| --- | --- | --- |
| 美國東部 (維吉尼亞北部) | us-east-1 | m1.small |
| 美國西部 (奧勒岡) | us-west-2 | m1.small |
| 亞太區域 (雪梨) | ap-southeast-2 | m1.small |
| 亞太區域 (東京) | ap-northeast-1 | m1.small |
| 歐洲 (愛爾蘭) | eu-west-1 | m1.small |

下表列出在不受支援的區域中預設 AWS Data Pipeline 啟動的 Amazon EC2 執行個體。 AWS Data Pipeline

| 區域名稱 | 區域 | 執行個體類型 |
| --- | --- | --- |
| 美國東部 (俄亥俄) | us-east-2 | t2.small |
| 美國西部 (加利佛尼亞北部) | us-west-1 | m1.small |
| 亞太區域 (孟買) | ap-south-1 | t2.small |

| 區域名稱 | 區域 | 執行個體類型 |
| --- | --- | --- |
| 亞太區域 (新加坡) | ap-southeast-1 | m1.small |
| 亞太區域 (首爾) | ap-northeast-2 | t2.small |
| 加拿大 (中部) | ca-central-1 | t2.small |
| 歐洲 (法蘭克福) | eu-central-1 | t2.small |
| 歐洲 (倫敦) | eu-west-2 | t2.small |
| 歐洲 (巴黎) | eu-west-3 | t2.small |
| 南美洲 (聖保羅) | sa-east-1 | m1.small |

## 其他支援的 Amazon EC2 執行個體

除了如不在管道定義中指定執行個體類型所建立的預設執行個體外，支援以下執行個體。

下表列出 AWS Data Pipeline 支援並可建立的 Amazon EC2 執行個體 (如果有指定)。

| 執行個體類別 | 執行個體類型 |
| --- | --- |
| 一般用途 | t2.nano \| t2.micro \| t2.small \| t2.medium \| t2.large |
| 運算最佳化 | c3.large \| c3.xlarge \| c3.2xlarge \| c3.4xlarge \| c3.8xlarge \| c4.large \| c4.xlarge \| c4.2xlarge \| c4.4xlarge \| c4.8xlarge \| c5.xlarge \| c5.9xlarge \| c5.2xlarge \| c5.4xlarge \| c5.9xlarge \| c5.18xlarge \| c5d.xlarge \| c5d.2xlarge \| c5d.4xlarge \| c5d.9xlarge \| c5d.18xlarge |
| 記憶體最佳化 | m3.medium \| m3.large \| m3.xlarge \| m3.2xlarge \| m4.large \| m4.xlarge \| m4.2xlarge \| m4.4xlarge \| m4.10xlarge \| m4.16xlarge \| m5.xlarge \| m5.2xlarge \| m5.4xlarge \| m5.12xlarge \| m5.24xlarge \| m5d.xlarge \| m5d.2xlarge \| m5d.4xlarge \| m5d.12xlarge \| m5d.24xlarge |

| 執行個體類別 | 執行個體類型 |
| --- | --- |
| | r3.large \| r3.xlarge \| r3.2xlarge \| r3.4xlarge \| r3.8xlarge \| r4.large \| r4.xlarge \| r4.2xlarge \| r4.4xlarge \| r4.8xlarge \| r4.16xlarge |
| 儲存最佳化 | i2.xlarge \| i2.2xlarge \| i2.4xlarge \| i2.8xlarge \| hs1.8xlarge \| g2.2xlarge \| g2.8xlarge \| d2.xlarge \| d2.2xlarge \| d2.4xlarge \| d2.8xlarge |

## 支援 Amazon EMR 叢集的亞馬遜 EC2 執行個體

此表格列出 AWS Data Pipeline 支援並可為 Amazon EMR 叢集建立的 Amazon EC2 執行個體 (若有指定)。如需詳細資訊，請參閱《Amazon EMR 管理指南》中支援的執行個體類型。

| 執行個體類別 | 執行個體類型 |
| --- | --- |
| 一般用途 | m1.small \| m1.medium \| m1.large \| m1.xlarge \| m3.xlarge \| m3.2xlarge |
| 運算最佳化 | c1.medium \| c1.xlarge \| c3.xlarge \| c3.2xlarge \| c3.4xlarge \| c3.8xlarge \| cc1.4xlarge\| cc2.8xlarge \| c4.large \| c4.xlarge \| c4.2xlarge\| c4.4xlarge \| c4.8xlarge \| c5.xlarge \| c5.9xlarge \| c5.2xlarge \| c5.4xlarge \| c5.9xlarge \| c5.18xlarge \| c5d.xlarge \| c5d.2xlarge \| c5d.4xlarge \| c5d.9xlarge \| c5d.18xlarge |
| 記憶體最佳化 | m2.xlarge \| m2.2xlarge \| m2.4xlarge \| r3.xlarge \| r3.2xlarge \| r3.4xlarge \| r3.8xlarge \| cr1.8xlarge \| m4.large \| m4.xlarge \| m4.2xlarge \| m4.4xlarge \| m4.10xlarge \| m4.16large \| m5.xlarge \| m5.2xlarge \| m5.4xlarge \| m5.12xlarge \| m5.24xlarge \| m5d.xlarge \| m5d.2xlarge \| m5d.4xlarge \| m5d.12xlarge \| m5d.24xlarge \| r4.large \| r4.xlarge \| r4.2xlarge \| r4.4xlarge \| r4.8xlarge \| r4.16xlarge |
| 儲存最佳化 | h1.4xlarge \| hs1.2xlarge \| hs1.4xlarge\| hs1.8xlarge \| i2.xlarge \| i2.2xlarge \| i2.4large \| i2.8xlarge \| d2.xlarge \| d2.2xlarge\| d2.4xlarge \| d2.8xlarge |
| 加速運算 | g2.2xlarge \| cg1.4xlarge |

# AWS Data Pipeline 概念

在開始之前，請先閱讀的主要概念和元件 AWS Data Pipeline。

目錄



## 管道定義

管道定義是您與業務邏輯溝通的方式 AWS Data Pipeline。其中包含下列資訊：

- 您資料來源的名稱、位置和格式
- 轉換資料的活動
- 這些活動的排程
- 執行您活動和先決條件的資源
- 必須滿足才能排程活動的先決條件
- 在管道繼續執行時提醒您狀態更新的方式

從您的管線定義中， AWS Data Pipeline 決定任務、對其進行排程，並將其指派給任務流道。如果任務未成功完成，請根據您的指示 AWS Data Pipeline 重試任務，並在必要時將其重新指派給其他任務執行者。如果任務重複失敗，您可以設定管道來接收通知。

例如，在管道定義中，您可以指定應用程式產生的日誌檔每個月在 2013 年存檔到 Amazon S3 儲存貯體。 AWS Data Pipeline 然後，將創建 12 個任務，每個任務複製超過一個月的數據，無論該月是否包含 30 天，31，28 還是 29 天。

您可以透過下列方式建立管道定義：

- 以圖形方式，使用控 AWS Data Pipeline 制台
- 以文字方式，透過撰寫命令列界面所用格式的 JSON 檔案
- 以程式設計方式，透過使用其中一個 AWS 開發套件或 AWS Data Pipeline API 來呼叫 Web 服務

管道定義可以包含以下類型的元件。

管道元件

資料節點

任務的輸入資料位置，或輸出資料的存放位置。

活動

使用運算資源 (通常為輸入和輸出資料節點) 執行排程的工作定義。

先決條件

必須為 true 才能執行動作的條件陳述式。

資源

執行管道所定義工作的運算資源。

動作

符合指定條件 (例如活動失敗) 時所觸發的動作。

如需詳細資訊，請參閱 管線定義檔案語法。

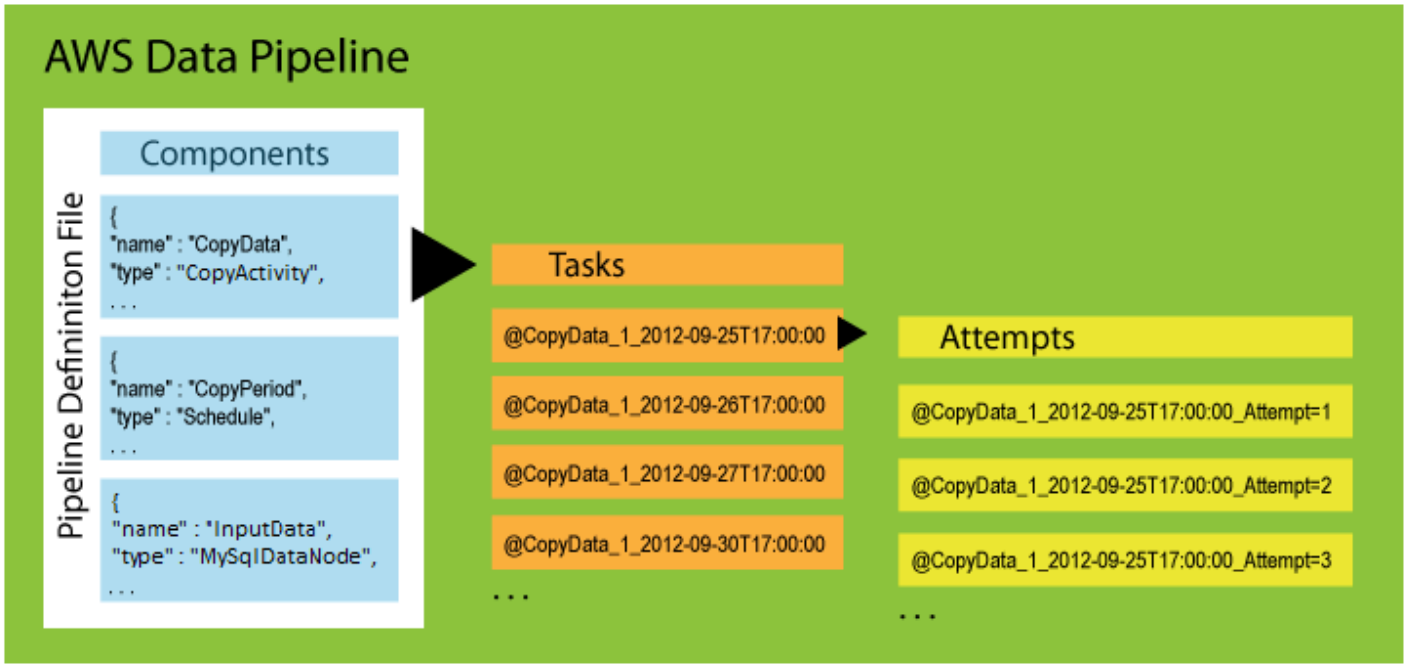
# 管道元件、執行個體和嘗試

排程管道的相關項目類型有三種：

- 管線元件 — 管線元件代表管線的商業邏輯，並以管線定義的不同部分表示。管道元件指定工作流程的資料來源、活動、排程和先決條件。這些元件可以從父元件繼承屬性。元件之間的關係是由參考定義。管道元件定義資料管理的規則。
- 執 AWS Data Pipeline 行個體 — 執行管線時，它會編譯管線元件以建立一組可操作的執行個體。每個執行個體包含執行特定任務的所有資訊。完整的執行個體集是管線的待辦事項清單。 AWS Data Pipeline 將實例交給任務運行者進行處理。

- 嘗試 — 若要提供健全的資料管理，請 AWS Data Pipeline 重試失敗的作業。它會繼續執行此操作，直到任務達到重試允許的最大數量。嘗試物件會追蹤各種嘗試、結果和失敗原因 (如果適用)。本質上，它是帶有計數器的實例。 AWS Data Pipeline 使用先前嘗試的相同資源執行重試，例如 Amazon EMR 叢集和 EC2 執行個體。

> **Note**
>
> 重試失敗的任務是容錯能力策略的一個重要部分，而 AWS Data Pipeline 定義提供條件和閾值來控制重試。不過，重試太多次可能會延遲偵測到無法復原的失敗，因為 AWS Data Pipeline 在用完您指定的所有重試次數之前不會報告失敗。如果在 AWS 資源上執行額外的重試，這些重試可能會產生額外的費用。因此，請仔細考慮何時適當超出您用來控制重試次數和相關設定的 AWS Data Pipeline 預設設定。

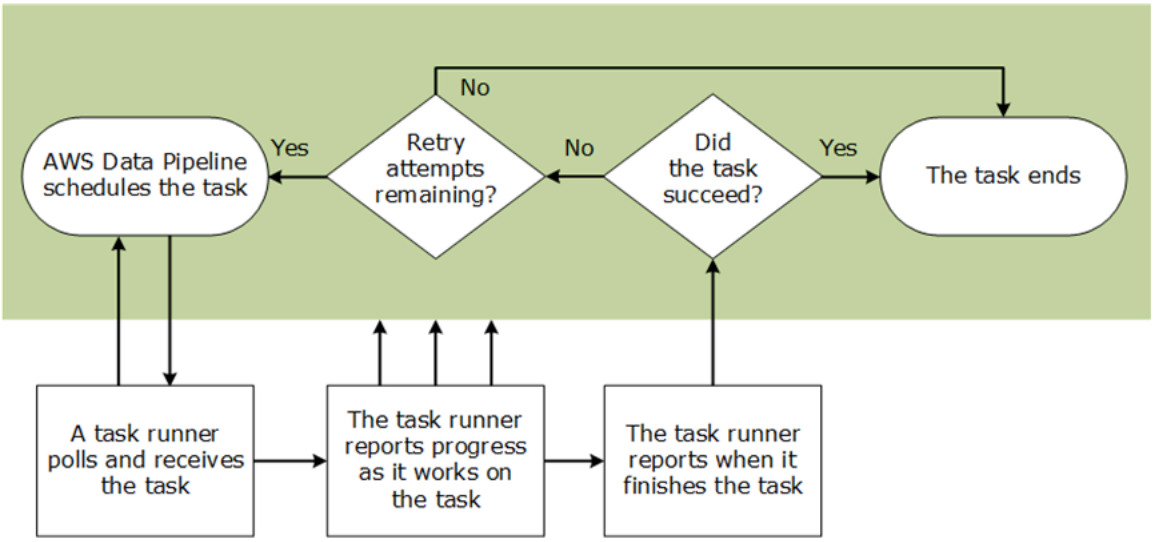



## 任務執行器

任務運行器是輪詢 AWS Data Pipeline 任務，然後執行這些任務的應用程序。

任務運行器是由提供的任務運行器的默認實現 AWS Data Pipeline。安裝並配置 Task Runner 後，它會輪詢 AWS Data Pipeline 與您已啟動的管道相關聯的任務。將任務指派給「工作執行器」時，它會執行該工作並將其狀態回報給 AWS Data Pipeline。

下圖說明工作執行者如何以 AWS Data Pipeline 及如何互動以處理排定的工作。工作是 AWS Data Pipeline 服務與工作執行者共用的離散工作單位。這不同於管道，管道是活動和資源的一般定義，通常會產生數個任務。




有兩種方法可以使用任務執行器來處理管道：

- AWS Data Pipeline 在由 Web 服務啟動和管理的資源上為您安裝任務執行 AWS Data Pipeline 器。
- 您可以在您管理的計算資源上安裝 Task Runner，例如長時間執行的 EC2 執行個體或現場部署伺服器。

如需使用任務執行器的詳細資訊，請參閱使用工作執行器。

# 資料節點

在中 AWS Data Pipeline，資料節點定義管線活動用作輸入或輸出的資料位置和類型。 AWS Data Pipeline 支持以下類型的數據節點：

D ynamoDBData 節點

包含EmrActivity要使用HiveActivity或要使用之資料的 DynamoDB 表格。

SqlDataNode

SQL 資料表和資料庫查詢，代表可供管道活動使用的資料。

> **Note**
>
> 以前， MySqlDataNode 被使用。請 SqlDataNode 改用。

RedshiftDataNode

一個 Amazon Redshift 表，其中包含RedshiftCopyActivity要使用的數據。

S3 DataNode

Amazon S3 位置，其中包含一或多個可供管道活動使用的檔案。

## 資料庫

AWS Data Pipeline 支援下列類型的資料庫：

JdbcDatabase

JDBC 資料庫。

RdsDatabase

一個 Amazon RDS 數據庫。

RedshiftDatabase

一個 Amazon Redshift 數據庫。

## 活動

在中 AWS Data Pipeline，活動是定義要執行之工作的管線元件。 AWS Data Pipeline 提供數個可容納常見案例的預先封裝活動，例如將資料從一個位置移至另一個位置、執行 Hive 查詢等。活動是可擴展的，因此您可以執行自己的自訂指令碼來支援無限的組合。

AWS Data Pipeline 支援下列類型的活動：

CopyActivity

將資料從一個位置複製到另一個。

EmrActivity

執行 Amazon EMR 叢集。

HiveActivity

在 Amazon EMR 叢集上執行蜂巢查詢。

HiveCopyActivity

在 Amazon EMR 叢集上執行 Hive 查詢，並支援和的進階資料篩選S3 DataNode和DynamoDBData 節點支援。

PigActivity

在 Amazon EMR 集群上運行豬腳本。

RedshiftCopyActivity

將資料複製到和從 Amazon Redshift 表格複製資料。

ShellCommandActivity

執行自訂 UNIX/Linux shell 命令做為活動。

SqlActivity

在資料庫上執行 SQL 查詢。

某些活動具有預備資料和資料庫資料表的特殊支援。如需詳細資訊，請參閱 使用管道活動預備資料和資料表。

# 先決條件

在中 AWS Data Pipeline，先決條件是包含條件陳述式的管線元件，該條件陳述式必須為 true，才能執行活動。例如，先決條件可以在管線活動嘗試複製它之前檢查來源資料是否存在。 AWS Data Pipeline 提供數個可容納常見案例的預先封裝先決條件，例如資料庫表格是否存在、是否存在 Amazon S3 金鑰等。不過，先決條件是可擴展的，並可讓您執行自己的自訂指令碼來支援無限的組合。

先決條件可分為兩種類型：系統受管先決條件和使用者受管先決條件。由 AWS Data Pipeline Web 服務代表您執行系統管理的先決條件，不需要計算資源。使用者受管先決條件只會在您使用 `runsOn` 或 `workerGroup` 欄位指定的運算資源上執行。`workerGroup` 資源衍生自使用先決條件的活動。

## 系統受管先決條件

D ynamoDBData 存在

檢查特定 DynamoDB 表格中是否存在資料。

D ynamoDBTable 存在

檢查 DynamoDB 資料表是否存在。

S3 KeyExists

檢查 Amazon S3 金鑰是否存在。

S3 PrefixNotEmpty

檢查 Amazon S3 前綴是否為空。

## 使用者受管先決條件

存在

檢查資料節點是否存在。

ShellCommandPrecondition

執行自訂 Unix/Linux shell 命令做為先決條件。

# 資源

在中 AWS Data Pipeline，資源是執行管線活動指定之工作的計算資源。 AWS Data Pipeline 支援下列類型的資源：

Ec2Resource

執行管道活動所定義工作的 EC2 執行個體。

EmrCluster

Amazon EMR 叢集，可執行管道活動定義的工作，例如EmrActivity.

資源可以與其工作資料集在相同區域中執行，甚至是不同於 AWS Data Pipeline的區域。如需詳細資訊，請參閱 在多個區域中搭配資源使用管道。

## 資源限制

AWS Data Pipeline 擴展以容納大量並行任務，您可以將其配置為自動創建處理大型工作負載所需的資源。這些自動建立的資源由您控制，並會計入您的 AWS 帳戶資源限制。例如，如果您設定 AWS Data Pipeline 為自動建立 20 個節點的 Amazon EMR 叢集來處理資料，而您的 AWS 帳戶的 EC2 執行個體限制設定為 20，則可能會不小心耗盡可用的回填資源。因此，請考慮將這些資源限制納入您的設計，或據以增加您的帳戶限制。如需服務限制的詳細資訊，請參閱 AWS 一般參考中的 AWS 服務限制。

> **Note**
>
> 每個 `Ec2Resource` 元件物件僅限一個執行個體。

## 支援的平台

管道可以將您的資源啟動至下列平台：

EC2-Classic

您的資源執行於與其他客戶共享的單一平面網路中。

EC2-VPC

您的資源執行於邏輯上與您 AWS 帳戶隔離的虛擬私有雲端 (VPC) 中。

您的 AWS 帳戶可以將資源啟動至兩個平台，或者僅在 EC2-VPC 中以區域為基礎啟動資源。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的支援平台。

如果您的 AWS 帳戶僅支援 EC2-VPC，我們會在每個 AWS 區域中為您建立預設 VPC。根據預設，我們會將您的資源啟動至您預設 VPC 的預設子網路。或者，您可以在設定資源時，建立非預設 VPC 並指定其中一個子網路，然後將您的資源啟動至非預設 VPC 的指定子網路。

當您將執行個體啟動至 VPC 時，您必須指定專為該 VPC 建立的安全群組。當您將執行個體啟動至 VPC 時，您無法指定為 EC2-Classic 建立的安全群組。此外，您必須使用安全群組 ID 而非安全性群組名稱，來識別 VPC 的安全群組。

## Amazon EC2 競價型執行個體與 Amazon EMR 叢集和 AWS Data Pipeline

管道可以將 Amazon EC2 競價型執行個體用於其 Amazon EMR 叢集資源中的任務節點。根據預設，管道會使用隨需執行個體。Spot 執行個體可讓您使用並執行備用的 EC2 執行個體。Spot 執行個體的

定價模型是對隨需和預留執行個體定價模型的補充，可根據您的應用程式提供最符合成本效益的選項來取得運算容量。如需詳細資訊，請參閱 Amazon EC2 Spot 執行個體產品頁面。

使用競價型執行個體時，請在叢集啟動時將競價型執行個體最高價 AWS Data Pipeline 提交給 Amazon EMR。它會自動將叢集的工作分配到您使用`taskInstanceCount`欄位定義的 Spot 執行個體任務節點數目。 AWS Data Pipeline 限制任務節點的 Spot 執行個體，以確保隨需核心節點可用於執行管道。

您可以編輯失敗或完成的管道資源執行個體來新增 Spot 執行個體。當管道重新啟動叢集時，會針對任務節點使用 Spot 執行個體。

### Spot 執行個體考量

搭配使用 Spot 執行個體時 AWS Data Pipeline，需要考量下列事項：

- 當競價型執行個體價格超過執行個體的最高價格時，或由於 Amazon EC2 容量原因，您的競價型執行個體可能會終止。不過，您不會遺失資料，因為使 AWS Data Pipeline 用具有永遠為隨需執行個體且不受終止的核心節點的叢集。
- 由於 Spot 執行個體是以非同步方式填滿容量，因此可能需要更長的時間啟動。因此，Spot 執行個體管道的執行速度可能比同等的隨需執行個體管道慢。
- 如果您未收到 Spot 執行個體 (例如當您的最高價太低時)，您的叢集可能不會執行。

## 動作

AWS Data Pipeline 動作是管線元件在發生某些事件 (例如成功、失敗或延遲活動) 時所採取的步驟。活動的事件欄位會參考動作，例如參考 `EmrActivity` 中 `onLateAction` 欄位的 `snsalarm`。

AWS Data Pipeline 依賴 Amazon SNS 通知作為以無人值守方式指示管道及其元件狀態的主要方式。如需詳細資訊，請參閱 Amazon SNS。除了 SNS 通知之外，您還可以使用主 AWS Data Pipeline 控台和 CLI 取得管線狀態資訊。

AWS Data Pipeline 支援下列動作：

SnsAlarm

根據 `onSuccess`、`OnFail` 和 `onLateAction` 事件，將 SNS 通知傳送至主題的動作。

終止

觸發取消擱置中或未完成活動、資源或資料節點的動作。您無法終止包含 `onSuccess`、`OnFail` 或 `onLateAction` 的動作。

# 主動監控管道

偵測問題的最佳方式是從頭開始主動監控您的管道。您可以設定管線元件，以通知您某些情況或事件，例如管線元件發生故障或未在排定的開始時間開始時間。 AWS Data Pipeline 在管道元件上提供可與 Amazon SNS 通知相關聯的事件欄位，例如、和 onSuccessOnFail，可讓您輕鬆設定通知onLateAction。

# 設定 AWS Data Pipeline

第一次使 AWS Data Pipeline 用前，請先完成下列工作。

任務

- 註冊成為 AWS
- 為 AWS Data Pipeline 和管道資源建立 IAM 角色
- 允許 IAM 主體 (使用者和群組) 執行必要動作
- 授予程式設計存取權

完成這些工作後，即可開始使用 AWS Data Pipeline。如需基本教學，請參閱AWS Data Pipeline 入門。

# 註冊成為 AWS

當您註冊 Amazon Web Services (AWS) 時，您的 AWS 帳戶會自動註冊 AWS 中的所有服務，包括 AWS Data Pipeline。您只需支付實際使用服務的費用。如需 AWS Data Pipeline 使用費率的詳細資訊，請參閱 AWS Data Pipeline。

## 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

   當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

# 建立具有管理權限的使用者

註冊後，請確保您的安全 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。AWS Management Console在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。
2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需指示，請參閱《IAM 使用者指南》中的為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 (主控台)。

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱 AWS IAM Identity Center 使用者指南中的啟用 AWS IAM Identity Center。
2. 在 IAM 身分中心中，將管理存取權授予使用者。

   若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。

以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM 身分中心使用者登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「建立權限集」。
2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「新增群組」。

# 為 AWS Data Pipeline 和管道資源建立 IAM 角色

AWS Data Pipeline 需要可決定執行動作和存取 AWS 資源的許可的 IAM 角色。管道角色決定 AWS Data Pipeline 具有的許可，而資源角色則決定在管線資源 (例如 EC2 執行個體) 上執行的應用程式擁有的許可。您可以在建立管線時指定這些角色。即使您未指定自訂角色並使用預設角色 `DataPipelineDefaultRole`和`DataPipelineDefaultResourceRole`，也必須先建立角色並附加權限原則。如需詳細資訊，請參閱 AWS Data Pipeline 的 IAM 角色。

# 允許 IAM 主體 (使用者和群組) 執行必要動作

若要使用管道，您帳戶中的 IAM 主體 (使用者或群組) 必須能夠針對管道所定義的其他服務執行必要 AWS Data Pipeline 的動作和動作。

為了簡化許可，您可以使用AWSDataPipeline_FullAccess受管政策連接到 IAM 主體。此受管理的原則可讓主參與者執行使用者需要的所有`iam:PassRole`動作，以及未指定自訂角色 AWS Data Pipeline 時所使用之預設角色的動作。

我們強烈建議您仔細評估此受管理政策，並將權限限制為您的使用者需要的原則。如有必要，請使用此政策做為起點，然後移除許可以建立更嚴格的內嵌許可政策，以附加至 IAM 主體。如需詳細資訊和權限原則範例，請參閱 用於 AWS Data Pipeline 的政策範例

類似下列範例的政策陳述式必須包含在附加至使用管線的任何 IAM 主體的政策中。此陳述式可讓 IAM 主體對管道使用的角色執行`PassRole`動作。如果您不使用預設角色，請將*`MyPipelineRole`*和*`MyResourceRole`*取代為您建立的自訂角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/MyPipelineRole",
                "arn:aws:iam::*:role/MyResourceRole"
            ]
        }
```

```
    ]
}
```

下列程序示範如何建立 IAM 群組、將AWSDataPipeline_FullAccess受管政策附加到群組，然後將使用者新增至群組。您可以針對任何內嵌政策使用此程序

若要建立使用者群組**DataPipelineDevelopers**並附加AWSDataPipeline_FullAccess原則

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇 Groups (群組)、Create New Group (建立新群組)。
3. 例如，輸入群組名稱**DataPipelineDevelopers**，然後選擇 [下一步]。
4. 輸入**AWSDataPipeline_FullAccess**篩選條件，然後從清單中選取它。
5. 選擇 Next Step (下一步)，然後選擇 Create Group (建立群組)。
6. 若要將使用者新增至群組：
   a. 從群組清單中選取您建立的群組。
   b. 選擇群組動作、新增使用者至群組。
   c. 從清單中選取要新增的使用者，然後選擇 [新增使用者至群組]。

# 授予程式設計存取權

如果使用者想要與 AWS 之外的 AWS Management Console. 授與程式設計存 AWS取權的方式取決於正在存取的使用者類型。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

| 哪個使用者需要程式設計存取權？ | 到 | By |
| --- | --- | --- |
| 人力身分<br><br>(IAM Identity Center 中管理的使用者) | 使用臨時登入資料來簽署對 AWS CLI、 AWS SDK 或 AWS API 的程式設計要求。 | 請依照您要使用的介面所提供的指示操作。<br><br>• 如需詳細資訊 AWS CLI，請參閱《使 AWS CLI 用AWS Command Line Interface者指南》 AWS IAM Identity Center中的〈配置使用〉。 |

| 哪個使用者需要程式設計存取權？ | 到 | By |
| --- | --- | --- |
| | | • 如需 AWS SDK、工具和 AWS API，請參閱 AWS SDK 和工具參考指南中的 IAM 身分中心身分驗證。 |
| IAM | 使用臨時登入資料來簽署對 AWS CLI、 AWS SDK 或 AWS API 的程式設計要求。 | 遵循《IAM 使用者指南》中的〈將臨時登入資料搭配 AWS 資源使用〉中的指示 |
| IAM | (不建議使用)<br>使用長期認證來簽署對 AWS CLI、 AWS SDK 或 AWS API 的程式設計要求。 | 請依照您要使用的介面所提供的指示操作。<br>• 如需相關資訊 AWS CLI，請參閱使用指南中的使用 IAM 使用者登入資料進行驗證。AWS Command Line Interface<br>• 對於 AWS SDK 和工具，請參閱 AWS SDK 和工具參考指南中的使用長期憑據進行身份驗證。<br>• 如需 AWS API，請參閱 IAM 使用者指南中的管理 IAM 使用者的存取金鑰。 |

# AWS Data Pipeline 入門

AWS Data Pipeline 可協助您透過可靠且經濟實惠的方式，排序、排程、執行和管理週期性資料處理工作負載。此服務可讓您根據商務邏輯，輕鬆使用內部部署和雲端中的結構化和非結構化資料來設計 extract-transform-load (ETL) 活動。

若要使用 AWS Data Pipeline，您可以建立「管道定義」以指定您的資料處理的商業邏輯。典型的管線定義包含定義要執行之工作的活動，以及定義輸入和輸出資料位置和類型的資料節點。

在本教學中，您會執行 shell 命令指令碼以計算 Apache Web 伺服器日誌中的 GET 請求數量。此管道每 15 分鐘執行一小時，並在每次反覆運算時將輸出寫入 Amazon S3。

先決條件

開始之前，請完成設定 AWS Data Pipeline中的任務。

管道物件

管道會使用下列物件：

ShellCommandActivity

讀取輸入日誌檔案並計算錯誤的數量。

S3 DataNode (輸入)

內含輸入日誌檔案的 S3 儲存貯體。

S3 DataNode (輸出)

輸出的 S3 儲存貯體。

Ec2Resource

AWS Data Pipeline 用來執行活動的運算資源。

請注意，如果您有大量的日誌檔案資料，您可以設定管道使用 EMR 叢集處理檔案，而不是 EC2 執行個體。

排程

定義在一小時內每 15 分鐘執行一次活動。

任務

# 建立管道

開始使用 AWS Data Pipeline 的最快速方法，就是使用管道定義，也稱為「範本」。

建立管道

1. 請在以下位置開啟AWS Data Pipeline主控台。 https://console.aws.amazon.com/datapipeline/
2. 從導覽列上，選取一個區域。無論您的位置為何，皆可選取任何可用的區域。許多 AWS 資源都是針對特定的區域，但 AWS Data Pipeline 可讓您使用與管道不同區域的資源。
3. 您看到的第一個畫面取決於您是否已在目前區域中建立管道。

   a. 如果您尚未在此區域建立管道，主控台會顯示簡介畫面。選擇立即開始使用。

   b. 如果您已在此區域中建立管道，則主控台會顯示一個頁面，列出該區域的管道。選擇 Create new pipeline (建立新的管道)。
4. 在名稱中，輸入管線的名稱。
5. (選擇性) 在說明中，輸入管線的說明。
6. 針對來源，選取使用範本建置，然後選取下列範本：入門使用ShellCommandActivity。
7. 選取範本時會開啟 Parameters (參數) 區段，請保留其下方 S3 input folder (輸入 S3 資料夾) 和 Shell command to run (要執行的 Shell 命令) 的預設值。按一下 S3 output folder (輸出 S3 資料夾) 旁的資料夾圖示，選取其中一個儲存貯體或資料夾，然後按一下 Select (選取)。
8. 保留 Schedule (排程) 下方的預設值。當您啟用管道時，管道即會開始執行，然後在一小時內每 15 分鐘執行一次。

   您也可以改為選擇 Run once on pipeline activation (在管道啟用時執行一次)。
9. 在「管線組態」下，保持啟用記錄。選擇記錄 S3 位置下方的資料夾圖示，選取其中一個值區或資料夾，然後選擇 [選取]。

   如果您願意，您可以改為停用記錄。
10. 在 [安全性/存取] 下，將 IAM 角色保持設定為 [預設]。
11. 按一下 Activate (啟動)。

如果您願意，您可以選擇在 Architect 中編輯來修改此配管。例如，您可以加入先決條件。

# 監控執行中的管道

啟用管道後，即會前往 Execution details (執行詳細資訊) 頁面，您可在此監控管道的進度。

監控管道的進度

1. 按一下 Update (更新) 或按 F5 以更新所顯示的狀態。

   > **Tip**
   >
   > 如果未列出任何執行，請確認 Start (in UTC) (開始 (UTC 時間)) 和 End (in UTC) (結束 (UTC 時間)) 涵蓋了管道排程的開始和結束時間，接著按一下 Update (更新)。

2. 當管道裡所有物件的狀態為 `FINISHED`，表示您的管道已成功完成了排程任務。
3. 如果您的管道未成功完成，請檢查管道設定是否有問題。關於管道執行個體執行失敗或未完成的故障排除，如需詳細資訊，請參閱 解決常見的問題。

# 檢視輸出

開啟 Amazon S3 主控台並導覽至您的儲存貯體。如果您在一小時內每 15 分鐘執行一次管道，您會看到四個含時間戳記的子資料夾。每個子資料夾都含有一個名為 `output.txt` 的輸出檔。因為我們每次都是在同一個輸入檔上執行指令碼，所以輸出檔都是相同的。

# 刪除管道

若要停止產生費用，請刪除管道。刪除配管會刪除配管定義及所有關聯物件。

若要刪除管線

1. 在「列出配管」頁面上，選取您的管線。
2. 按一下 [動作]，然後選擇 [刪除]。
3. 出現確認提示時，請選擇 Delete (刪除)。

如果您已完成本教學的輸出，請從 Amazon S3 儲存貯體刪除輸出資料夾。

# 使用管道

您可以使用命令列介面 (CLI) 或 AWS SDK 來管理、建立和修改管線。下列各節會介紹基礎的 AWS Data Pipeline 概念，並示範如何使用管道。

> **Important**
>
> 開始之前，請參閱設定 AWS Data Pipeline。

目錄



## 建立管線

AWS Data Pipeline 提供數種方式，可讓您建立管道：

- 使用 AWS Command Line Interface (CLI) 搭配為您提供的方便而提供的範本。如需詳細資訊，請參閱使用 CLI 從資料管線範本建立管線。
- 搭配 JSON 格式的管道定義檔案使用 AWS Command Line Interface (CLI)。
- 使用語言特定 API 的 AWS 開發套件。如需詳細資訊，請參閱使用 API。

# 使用 CLI 從資料管線範本建立管線

資料管線提供數個預先設定的管線定義，稱為範本。您可以使用範本來快速開始使用 AWS Data Pipeline。這些範本可在 Amazon S3 位置的公用儲存貯體中使用：`s3://datapipeline-us-east-1/templates/`。這些預先定義的範本是為了達成特定的使用案例而建立，並可用於建立管線。您可以使`aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"`用列出所有可用的模板。

## 使用 CLI 從範本建立管線

假設您想要建立將 DynamoDB 資料表匯出至亞馬遜 S3 的管道。在這種情況下使用的模板可以在以下位置找到：`s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`。

若要下載範本 JSON 並使用 CLI 建立管線

1. 使用 `aws s3 cp` CLI 或捲曲下載模板。例如：

   ```
   aws s3 cp "s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB
    table to S3.json" <destination directory>
   ```

2. 視需要對下載的範本進行變更。例如，若要使用最新的 EMR 發行版本，請變更`EmrClusterForBackup`物件中的`releaseLabel`欄位、變更主要和核心執行個體類型，以及變更範本中參數的預設值。
3. 使用 `create-pipeline` CLI 建立管線。例如：

   ```
   aws datapipeline create-pipeline --name my-ddb-backup-pipeline --unique-id my-ddb-
   backup-pipeline --region ap-northeast-1
   ```

4. 請注意建立的管線 ID。
5. 用`put-pipeline-definition`於上傳定義。使用`--parameter-values`選項提供您要覆寫其預設值的參數值。

如需範本的詳細資訊，請參閱Choose a template (選擇範本)。

## Choose a template (選擇範本)

下列範本可從 Amazon S3 儲存貯體下載：`s3://datapipeline-us-east-1/templates/`

範本

- 使用 ShellCommandActivity 入門
- 執行 AWS CLI 命令
- 將動態資料表匯出至 S3
- 從 S3 匯入備份資料
- 在亞馬遜 EMR 叢集上執行任務
- 亞馬遜 RDS MySQL 表的完整副本到亞馬遜 S3
- 亞馬遜 RDS MySQL 表的增量副本到亞馬遜 S3
- 將 S3 資料載入亞馬遜 RDS MySQL 資料表
- 亞馬遜 RDS MySQL 表的完整副本到亞馬遜紅移
- 將亞馬遜 RDS MySQL 表的增量複製到亞馬遜紅移
- 將數據從亞馬遜 S3 加載到亞馬遜紅移

## 使用 ShellCommandActivity 入門

開始使用ShellCommandActivity範本會執行 shell 命令指令碼，以計算記錄檔中 GET 要求的數目。輸出會在管道的每個排定執行時，寫入時間戳記的 Amazon S3 位置。

範本使用下列管道物件：

- ShellCommandActivity
- S3 InputNode
- S3 OutputNode
- Ec2Resource

## 執行 AWS CLI 命令

此範本會根據排程的間隔，執行使用者指定的 AWS CLI 命令。

## 將動態資料表匯出至 S3

將動態資料表匯出至 S3 範本可排程 Amazon EMR 叢集，將資料從 DynamoDB 表格匯出至 Amazon S3 儲存貯體。此範本使用 Amazon EMR 叢集，該叢集的大小與 DynamoDB 表格可用的輸送量值成比

例。雖然您可以增加資料表上的 IOP，但這可能會在匯入及匯出時產生額外的成本。以前，導出使用了，HiveActivity但現在使用本機MapReduce。

範本使用下列管道物件：

- EmrActivity
- EmrCluster
- D ynamoDBData 節點
- S3 DataNode

## 從 S3 匯入備份資料

從 S3 匯入 DynamoDB 備份資料範本會排程 Amazon EMR 叢集，將先前在 Amazon S3 中建立的 DynamoDB 備份載入至 DynamoDB 表格。DynamoDB 表格中的現有項目會以備份資料中的項目進行更新，而新項目則會新增至表格。此範本使用 Amazon EMR 叢集，該叢集的大小與 DynamoDB 表格可用的輸送量值成比例。雖然您可以增加資料表上的 IOP，但這可能會在匯入及匯出時產生額外的成本。以前，導入使用了，HiveActivity但現在使用本機MapReduce。

範本使用下列管道物件：

- EmrActivity
- EmrCluster
- D ynamoDBData 節點
- S3 DataNode
- S3 PrefixNotEmpty

## 在亞馬遜 EMR 叢集上執行任務

在彈性MapReduce叢集範本上執行任務會根據提供的參數啟動 Amazon EMR 叢集，並根據指定的排程開始執行步驟。一旦任務完成，EMR 叢集便會終止。您可以指定選擇性的引導操作來安裝額外的軟體，或是變更叢集上的應用程式組態。

範本使用下列管道物件：

- EmrActivity
- EmrCluster

## 亞馬遜 RDS MySQL 表的完整副本到亞馬遜 S3

RDS MySQL 資料表到 S3 範本的完整複本會複製整個亞馬遜 RDS MySQL 資料表，並將輸出存放在亞馬遜 S3 位置。輸出會以 CSV 檔案形式存放在指定 Amazon S3 位置下的時間戳記子資料夾中。

範本使用下列管道物件：

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

## 亞馬遜 RDS MySQL 表的增量副本到亞馬遜 S3

RDS MySQL 資料表的增量複製到 S3 範本會從 Amazon RDS MySQL 表格執行資料的遞增複本，並將輸出存放在亞馬遜 S3 位置。亞馬遜 RDS MySQL 表必須有一個上次修改的列。

此範本會複製自排程啟動時間以來，於排程間隔期間對資料表進行的變更。排程類型為時間序列，因此，如果將複AWS Data Pipeline本排定在特定小時內，則會複製具有「上次修改時間戳記」在小時內的表格資料列。對資料表進行的實體刪除則不會複製。在每次排定的執行時，輸出會寫入 Amazon S3 位置下的時間戳記子資料夾中。

範本使用下列管道物件：

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

## 將 S3 資料載入亞馬遜 RDS MySQL 資料表

將 S3 資料載入 RDS MySQL 表格範本會排程 Amazon EC2 執行個體，將 CSV 檔案從以下指定的亞馬遜 S3 檔案路徑複製到亞馬遜 RDS MySQL 表格。CSV 檔案不應具備標頭列。該範本會將 Amazon RDS MySQL 資料表中的現有項目更新為亞馬遜 S3 資料中的項目，並將來自亞馬遜 S3 資料的新項目新增至亞馬遜 RDS MySQL 表格。您可以將資料載入現有的資料表，或是提供 SQL 查詢來建立新的資料表。

範本使用下列管道物件：

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3 DataNode

## 亞馬遜 RDS 到亞馬遜紅移模板

下列兩個範本會使用翻譯指令碼將表格從 Amazon RDS MySQL 複製到 Amazon Redshift，該指令碼會使用來源資料表結構描述建立 Amazon Redshift 表格，並注意下列警告：

- 如果未指定分發金鑰，Amazon RDS 表中的第一個主索引鍵會設為分發金鑰。
- 當您複製到亞馬遜紅移時，您不能跳過存在於亞馬遜 RDS MySQL 表中的列。
- (選擇性) 您可以提供 Amazon RDS MySQL 至亞馬遜紅移資料行資料類型對應做為範本中的其中一個參數。如果有指定，指令碼會使用此指令碼來建立 Amazon Redshift 表格。

如果正在使用`Overwrite_Existing`亞馬遜紅移插入模式：

- 如果未提供分發金鑰，則會使用亞馬遜 RDS MySQL 資料表上的主索引鍵。
- 若資料表上有複合主索引鍵，則會使用第一個做為分發索引鍵 (若沒有提供分發索引鍵的話)。只有第一個複合鍵被設置為亞馬遜 Redshift 表中的主鍵。
- 如果未提供分發金鑰，且 Amazon RDS MySQL 資料表上沒有主索引鍵，則複製作業會失敗。

如需 Amazon 紅移的相關資訊，請參閱下列主題：

- Amazon Redshift 叢集
- 亞馬遜紅移複製
- 分發樣式及 DISTKEY 範例
- 排序索引鍵

下表說明指令碼如何翻譯資料類型：

MySQL 和亞馬遜紅移之間的數據類型轉換

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| TINYINT,<br><br>TINYINT (size) | SMALLINT | MySQL：-128 到 127。可在括弧內指定位數上限。<br><br>亞馬遜紅移：INT2。帶正負號的 2 位元組整數 |
| TINYINT UNSIGNED,<br><br>TINYINT (size) UNSIGNED | SMALLINT | MySQL：不帶正負號的 0 到 255。可在括弧內指定位數上限。<br><br>亞馬遜紅移：INT2。帶正負號的 2 位元組整數 |
| SMALLINT,<br><br>SMALLINT(size) | SMALLINT | MySQL：一般的 -32768 到 32767。可在括弧內指定位數上限。<br><br>亞馬遜紅移：INT2。帶正負號的 2 位元組整數 |
| SMALLINT UNSIGNED,<br><br>SMALLINT(size) UNSIGNED, | INTEGER | MySQL：不帶正負號的 0 到 65535*。可在括弧內指定位數上限<br><br>亞馬遜紅移：INT4。帶正負號的 4 位元組整數 |
| MEDIUMINT,<br><br>MEDIUMINT(size) | INTEGER | MySQL：388608 到 8388607。可在括弧內指定位數上限<br><br>亞馬遜紅移：INT4。帶正負號的 4 位元組整數 |
| MEDIUMINT UNSIGNED,<br><br>MEDIUMINT(size) | INTEGER | MySQL：0 到 16777215。可在括弧內指定位數上限 |

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| UNSIGNED | | 亞馬遜紅移：INT4。帶正負號的 4 位元組整數 |
| INT,<br>INT(size) | INTEGER | MySQL：147483648 到 2147483647<br>亞馬遜紅移：INT4。帶正負號的 4 位元組整數 |
| INT UNSIGNED,<br>INT(size) UNSIGNED | BIGINT | MySQL：0 到 4294967295<br>亞馬遜紅移：INT8。帶正負號的 8 位元組整數 |
| BIGINT<br>BIGINT(size) | BIGINT | 亞馬遜紅移：INT8。帶正負號的 8 位元組整數 |
| BIGINT UNSIGNED<br>BIGINT(size) UNSIGNED | VARCHAR(20*4) | MySQL：0 到 18446744073709551615<br>亞馬遜紅移：沒有本地等價物，所以使用字符數組。 |
| FLOAT<br>FLOAT(size,d)<br>FLOAT(size,d) UNSIGNED | REAL | 可在 size 參數內指定位數上限。小數點右方的小數位數上限則會在 d 參數內指定。<br>亞馬遜紅移：FLOAT4 |
| DOUBLE(size,d) | DOUBLE PRECISION | 可在 size 參數內指定位數上限。小數點右方的小數位數上限則會在 d 參數內指定。<br>亞馬遜紅移：FLOAT8 |

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| DECIMAL(size,d) | DECIMAL(size,d) | DOUBLE 會以字串形式存放，允許固定的小數點。可在 size 參數內指定位數上限。小數點右方的小數位數上限則會在 d 參數內指定。<br><br>亞馬遜紅移：沒有原生等價物。 |
| CHAR(size) | VARCHAR(size*4) | 保留固定長度的字串，其中可包含字母、數字和特殊字元。固定長度會在括弧內以參數指定。最多可存放 255 個字元。<br><br>字串右側則會填補空格。<br><br>亞馬遜紅移：CHAR 數據類型不支持多字節字符，因此使用 VARCHAR。<br><br>根據 RFC3629，每個字元的位元組數量上限為 4，因此會將字元定義表限制在 U+10FFFF。 |
| VARCHAR(size) | VARCHAR(size*4) | 最多可存放 255 個字元。<br><br>VARCHAR 不支援下列無效 UTF-8 字碼元素：0xD800-0xDFFF、(位元組序列：ED A0 80- ED BF BF)、0xFDD0- 0xFDEF、0xFFFE 及 0xFFFF、(位元組序列：EF B7 90- EF B7 AF, EF BF BE 和 EF BF BF) |

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| TINYTEXT | VARCHAR(255*4) | 保留長度上限為 255 個字元的字串 |
| TEXT | VARCHAR(max) | 保留長度上限為 65,535 個字元的字串。 |
| MEDIUMTEXT | VARCHAR(max) | 0 到 16,777,215 個字元 |
| LONGTEXT | VARCHAR(max) | 0 到 4,294,967,295 個字元 |
| BOOLEAN<br>BOOL<br>TINYINT(1) | BOOLEAN | MySQL：這些類型是 TINYINT(1) 的同義詞。值為 0 會視為 False。值不為零則會視為 True。 |
| BINARY[(M)] | varchar(255) | M 是 0 到 255 個位元組 (固定) |
| VARBINARY(M) | VARCHAR(max) | 0 到 65,535 個位元組 |
| TINYBLOB | VARCHAR(255) | 0 到 255 個位元組 |
| BLOB | VARCHAR(max) | 0 到 65,535 個位元組 |
| MEDIUMBLOB | VARCHAR(max) | 0 到 16,777,215 個位元組 |
| LONGBLOB | VARCHAR(max) | 0 到 4,294,967,295 個位元組 |
| ENUM | VARCHAR(255*2) | 限制並非常值列舉字串的長度，而是列舉值數量的資料表定義。 |
| SET | VARCHAR(255*2) | 與列舉相似。 |
| DATE | DATE | (YYYY-MM-DD)<br>"1000-01-01" 到 "9999-12-31" |

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| TIME | VARCHAR(10*4) | (hh:mm:ss)<br><br>"-838:59:59" 到 "838:59:59" |
| DATETIME | TIMESTAMP | (YYYY-MM-DD hh:mm:ss)<br><br>1000-01-01 00:00:00" 到 "9999-12-31 23:59:59" |
| TIMESTAMP | TIMESTAMP | (YYYYMMDDhhmmss)<br><br>19700101000000 到 2037+ |
| YEAR | VARCHAR(4*4) | (YYYY)<br><br>1900 到 2155 |
| column SERIAL | ID 產生 / OLAP 資料倉儲不需要此屬性，因為會複製此資料行。<br><br>SERIAL 關鍵字不會在翻譯時新增。 | SERIAL 實際上是名為 SEQUENCE 的實體。它會獨立存在於您資料表的剩餘部分。<br><br>column GENERATED BY DEFAULT<br><br>相當於：<br><br>CREATE SEQUENCE name; CREATE TABLE table ( column INTEGER NOT NULL DEFAULT nextval(name) ); |

| MySQL 資料類型 | 亞馬遜紅移數據類型 | 備註 |
| --- | --- | --- |
| column BIGINT UNSIGNED NOT NULL AUTO_INCR EMENT UNIQUE | ID 產生 / OLAP 資料倉儲不需要此屬性，因為會複製此資料行。<br><br>因此 SERIAL 關鍵字不會在翻譯時新增。 | SERIAL 實際上是名為 SEQUENCE 的實體。它會獨立存在於您資料表的剩餘部分。<br><br>column GENERATED BY DEFAULT<br><br>相當於：<br><br>CREATE SEQUENCE name; CREATE TABLE table ( column INTEGER NOT NULL DEFAULT nextval(n ame) ); |
| ZEROFILL | ZEROFILL 關鍵字不會在翻譯時新增。 | INT UNSIGNED ZEROFILL NOT NULL<br><br>ZEROFILL 會用零填補欄位的顯示值，直到資料行定義中指定的顯示寬度。超過顯示寬度的值不會截斷。請注意，使用 ZEROFILL 表示也使用 UNSIGNED。 |

## 亞馬遜 RDS MySQL 表的完整副本到亞馬遜紅移

亞馬遜 RDS MySQL 表格到亞馬遜紅移範本的完整副本會將整個亞馬遜 RDS MySQL 表格複製到亞馬遜 S3 資料夾中的資料，方法是將資料暫存到亞馬遜紅移表。Amazon S3 暫存資料夾必須與 Amazon 紅移叢集位於相同的區域。如果亞馬遜 RDS MySQL 資料表尚未存在，則會使用與來源 Amazon RDS MySQL 資料表相同的結構描述建立。請提供您想要在建立亞馬遜紅移表格期間套用的任何 Amazon RDS MySQL 至亞馬遜紅移資料行資料類型覆寫。

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

## 將亞馬遜 RDS MySQL 表的增量複製到亞馬遜紅移

亞馬遜 RDS MySQL 表的增量副本到亞馬遜紅移範本會將資料從亞馬遜 RDS MySQL 表格複製到亞馬遜 S3 資料夾中的資料，方法是將資料從亞馬遜 RDS MySQL 表複製到亞馬遜紅移表。

Amazon S3 暫存資料夾必須與 Amazon 紅移叢集位於相同的區域。

AWS Data Pipeline使用翻譯指令碼來建立具有與來源 Amazon RDS MySQL 資料表相同結構描述的 Amazon Redshift 表 (如果它尚未存在)。您必須向亞馬遜紅移資料表建立期間套用的任何 Amazon RDS MySQL 提供給亞馬遜紅移資料行資料類型覆寫。

此範本會從排定的開始時間開始，在排定的間隔之間複製對 Amazon RDS MySQL 表格所做的變更。不會複製對亞馬遜 RDS MySQL 資料表的實體刪除。您必須提供存放上次修改時間值的資料行名稱。

當您使用預設範本為增量 Amazon RDS 複本建立管道時，會建立具有預設名稱`RDSToS3CopyActivity`的活動。您可以重新命名它。

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

## 將數據從亞馬遜 S3 加載到亞馬遜紅移

將資料從 S3 載入紅移範本會將資料從 Amazon S3 資料夾複製到亞馬遜紅移表格中。您可以將資料載入現有的資料表，或是提供 SQL 查詢來建立資料表。

資料會根據亞馬遜紅移COPY選項複製。亞馬遜紅移表必須具有與亞馬遜 S3 中的資料相同的結構描述。如需COPY選項，請參閱 Amazon Redshift 資料庫開發人員指南中的複製。

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3 DataNode
- RedshiftDataNode
- RedshiftDatabase
- Ec2Resource

## 使用參數化範本建立管線

您可以使用參數化範本來自訂管道定義。這可讓您建立常見的管道定義，但仍可以在您將管道定義新增到新的管道時提供不同的參數。

目錄



### 將 My變數新增至管線定義

當您建立管道定義檔案時，請使用以下語法指定變數：#{my*Variable*}。您必須為變數加上 my 前綴。例如，下列管線定義檔案包含下列變數：*myShellCmd*、*MyS3 InputLoc* 和 *MyS OutputLoc* 3。pipeline-definition.json

> **Note**
>
> 管道定義具有 50 參數的上限。

```
{
  "objects": [
    {
```

```
      "id": "ShellCommandActivityObj",
      "input": {
        "ref": "S3InputLocation"
      },
      "name": "ShellCommandActivityObj",
      "runsOn": {
        "ref": "EC2ResourceObj"
      },
      "command": "#{myShellCmd}",
      "output": {
        "ref": "S3OutputLocation"
      },
      "type": "ShellCommandActivity",
      "stage": "true"
    },
    {
      "id": "Default",
      "scheduleType": "CRON",
      "failureAndRerunMode": "CASCADE",
      "schedule": {
        "ref": "Schedule_15mins"
      },
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "S3InputLocation",
      "name": "S3InputLocation",
      "directoryPath": "#{myS3InputLoc}",
      "type": "S3DataNode"
    },
    {
      "id": "S3OutputLocation",
      "name": "S3OutputLocation",
      "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
      "type": "S3DataNode"
    },
    {
      "id": "Schedule_15mins",
      "occurrences": "4",
      "name": "Every 15 minutes",
      "startAt": "FIRST_ACTIVATION_DATE_TIME",
```

```
      "type": "Schedule",
      "period": "15 Minutes"
    },
    {
      "terminateAfter": "20 Minutes",
      "id": "EC2ResourceObj",
      "name": "EC2ResourceObj",
   "instanceType":"t1.micro",
      "type": "Ec2Resource"
    }
  ]
}
```

定義參數物件

您可以建立具備參數物件的個別檔案，定義您管道定義中的變數。例如，下列 JSON 檔案包含上述範例管線定義中之 *MyS3 InputLoc* 和 *MyS3 OutputLoc* 變數的參數物件。parameters.json *myShellCmd*

```
{
  "parameters": [
    {
      "id": "myShellCmd",
      "description": "Shell command to run",
      "type": "String",
      "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* > ${OUTPUT1_STAGING_DIR}/
output.txt"
    },
    {
      "id": "myS3InputLoc",
      "description": "S3 input location",
      "type": "AWS::S3::ObjectKey",
      "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data"
    },
    {
      "id": "myS3OutputLoc",
      "description": "S3 output location",
      "type": "AWS::S3::ObjectKey"
    }
  ]
}
```

> **Note**
>
> 您可以直接將這些物件新增到管道定義檔案，而無需使用個別檔案。

下表說明參數物件的屬性。

參數屬性

| 屬性 | 類型 | 描述 |
| --- | --- | --- |
| id | 字串 | 參數的唯一識別符。若要在輸入或顯示時遮住該值，請新增星號 ('*') 做為前綴。例如，*myVariable —。請注意，這也會在 AWS Data Pipeline 存放它之前加密該值。 |
| 描述 | 字串 | 參數的描述。 |
| 類型 | 字串、整數、雙精度或 AWS::S3::ObjectKey | 定義輸入值允許範圍及驗證規則的參數類型。預設為 String (字串)。 |
| 選擇性 | Boolean | 指出參數為選擇性或必要參數。預設值為 false。 |
| allowedValues | List of Strings (字串清單) | 列舉參數所有允許的值。 |
| 預設 | 字串 | 參數的預設值。若您使用參數值指定此參數的值，則會覆寫預設值。 |
| isArray | Boolean | 指出參數是否是陣列。 |

定義參數值

您可以使用參數值建立個別檔案，來定義您的變數。例如，下列 JSON 檔案包含上述範例管線定義中 *MyS3 OutputLoc* 變數的值。file://values.json

```
{
  "values":
    {
      "myS3OutputLoc": "myOutputLocation"
    }
}
```

提交管道定義

當您提交管道定義時，您可以指定參數、參數物件和參數值。例如，您可以使用put-pipeline-definitionAWS CLI命令，如下所示：

```
$ aws datapipeline put-pipeline-definition --pipeline-id id --pipeline-definition
 file://pipeline-definition.json \
--parameter-objects file://parameters.json --parameter-values-uri file://values.json
```

> **Note**
>
> 管道定義具有 50 參數的上限。parameter-values-uri 的檔案大小具有 15 KB 的上限。

# 檢視您的管道

您可以使用命令列介面 (CLI) 檢視管線。

使用 AWS CLI 檢視您的管道

- 請使用以下的 list-pipelines 命令列出您的管道：

  ```
  aws datapipeline list-pipelines
  ```

# 解譯狀態代碼

AWS Data Pipeline 主控台和 CLI 中顯示的狀態層級會指出管道及其元件的狀況。管道狀態單純只是管道的概觀；若要查看詳細資訊，請檢視個別管道元件的狀態。

若管道已準備就緒 (管道定義通過驗證)、目前正在執行工作，或是已完成執行工作，則管道會具備 SCHEDULED 狀態。若管道尚未啟用或無法執行工作 (例如管道定義無法通過驗證)，則管道會具備 PENDING 狀態

或管道的狀態為 PENDING、INACTIVE 或 FINISHED，則管道會被視為非作用中。非作用中的管道會產生費用 (如需詳細資訊，請參閱定價)。

狀態碼

ACTIVATING

正在啟動元件或資源，例如 EC2 執行個體。

CANCELED

元件已由使用者取消，或AWS Data Pipeline在執行元件之前取消。當此元件所依賴的不同元件或資源發生故障時，可能會自動發生這種情況。

CASCADE_FAILED

元件或資源因其中一個相依性的重疊顯示失敗而取消，但該元件可能不是失敗的原始來源。

DEACTIVATING

管線正在停用。

FAILED

元件或資源發生錯誤並停止運作。當元件或資源發生故障時，可能會導致取消和失敗重疊顯示至其他相依元件的元件。

FINISHED

元件已完成其指定的工作。

INACTIVE

管線已停用。

PAUSED

組件已暫停，目前未執行其工作。

PENDING

管線已準備好第一次啟動。

RUNNING

資源正在執行並準備好接收工作。

SCHEDULED

資源已排定為執行。

SHUTTING_DOWN

成功完成其工作後，資源正在關閉。

SKIPPED

使用比目前排程晚的時間戳記啟動配管後，元件略過執行間隔。

TIMEDOUT

資源超過terminateAfter臨界值並已停止AWS Data Pipeline。資源達到此狀態後，AWS Data Pipeline忽略該actionOnResourceFailure資源的retryDelay、和retryTimeout值。此狀態僅適用於資源。

VALIDATING

管線定義正由驗證AWS Data Pipeline。

WAITING_FOR_RUNNER

元件正在等待其 Worker 用戶端擷取工作項目。元件和 Worker 用戶端關係由該元件定義的runsOn或workerGroup欄位控制。

WAITING_ON_DEPENDENCIES

在執行其工作之前，元件會確認其預設和使用者設定的先決條件是否符合。

## 解譯管道和元件運作狀態

每個該管道中的管道和元件都會傳回 HEALTHY、ERROR、"-"、No Completed Executions 或 No Health Information Available 的運作狀態。管道只會在管道元件完成第一次執行，或元件的先決條件失敗，才會具有運作狀態。元件的運作狀態會彙整到管道運作狀態，而您會在檢視管道執行詳細資訊時先看到錯誤狀態。

管道運作狀態

HEALTHY

所有元件的彙整運作狀態為 HEALTHY。這表示至少有一個元件已成功完成。您可以按一下HEALTHY狀態，在「執行詳細資訊」頁面上查看最近成功完成的配管元件執行處理。

ERROR

管道中至少有一個元件的運作狀態為 ERROR。您可以按一下ERROR狀態，在「執行詳細資訊」頁面上查看最近失敗的管線元件執行處理。

No Completed Executions 或 No Health Information Available

此管道沒有報告任何運作狀態。

> **Note**
>
> 雖然元件幾乎會立即更新其運作狀態，但管道運作狀態最多可能需要五分鐘來更新。

元件運作狀態

HEALTHY

若元件成功完成了執行，並且已標記為 FINISHED 或 MARK_FINISHED 狀態，則元件 (Activity 或 DataNode) 便會具有 HEALTHY 的運作狀態。您可以按一下元件的名稱或HEALTHY狀態，在「執行詳細資訊」頁面上查看最近成功完成的配管元件例證。

ERROR

元件層級發生錯誤，或是其中一個先決條件失敗。FAILED、TIMEOUT 或 CANCELED 狀態都會觸發此錯誤。您可以按一下元件的名稱或ERROR狀態，在「執行詳細資訊」頁面上查看最近失敗的配管元件例證。

No Completed Executions 或 No Health Information Available

此元件沒有報告任何運作狀態。

## 檢視您的管道定義

使用命令列介面 (CLI) 檢視管線定義。CLI 會以 JSON 格式列印管線定義檔案。如需管道定義檔案語法和使用方式的資訊，請參閱管線定義檔案語法。

使用 CLI 時，最好先擷取管線定義，然後再提交修改，因為在您上次使用管線定義之後，其他使用者或處理程序可能會變更管線定義。透過下載目前定義的複本並用它來做為您修改的基礎，您可以確認您使用的是最新的管道定義。在修改管道定義之後再次擷取它也是個不錯的做法，這可讓您確認更新已成功。

使用 CLI 時，您可以取得兩個不同版本的管道。`active` 版本是目前正在執行中的管道。`latest` 版本是您編輯執行中管道時建立的複本。當您上傳編輯後的管道時，它便會成為 `active` 版本，而先前的 `active` 版本則無法繼續使用。

使用 AWS CLI 取得管道定義

若要取得完整的管線定義，請使用get-pipeline-definition指令。管道定義會印出至標準輸出 (stdout)。

以下範例會取得指定管道的管道定義。

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE
```

若要擷取特定版本的管道，請使用 `--version` 選項。以下範例會擷取指定管道的 `active` 版本。

```
aws datapipeline get-pipeline-definition --version active --id df-00627471SOVYZEXAMPLE
```

## 檢視管道執行個體詳細資訊

您可以監控您管道的進度。如需執行個體狀態的詳細資訊，請參閱解譯管道狀態詳細資訊。關於管道執行個體執行失敗或未完成的故障排除，如需詳細資訊，請參閱 解決常見的問題。

使用 AWS CLI 監控管道進度

若要擷取管道執行個體詳細資訊 (例如管道執行次數的歷史記錄)，請使用 list-runs 命令。此命令可讓您篩選根據其目前狀態或啟動日期範圍傳回的執行清單。篩選結果很有用，因為根據管道的壽命和排程，執行歷史記錄可能會相當龐大。

以下範例會擷取所有執行的資訊。

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE
```

以下範例會擷取所有已完成執行的資訊。

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE --status finished
```

以下範例會擷取所有在指定時間範圍內啟動的執行資訊。

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE --start-interval
 "2013-09-02","2013-09-11"
```

## 檢視管道日誌

管道建立時支援管道層級記錄，方法是在主控台中指定 Amazon S3 位置，或在 SDK/CLI 中的預設物件pipelineLogUri中指定 Amazon S3 位置。該 URI 內每個管道的目錄結構都與以下內容相似：

```
pipelineId
    -componentName
        -instanceId
            -attemptId
```

針對管道 df-00123456ABC7DEF8HIJK，目錄結構看起來會與以下內容相似：

```
df-00123456ABC7DEF8HIJK
    -ActivityId_fXNzc
        -@ActivityId_fXNzc_2014-05-01T00:00:00
            -@ActivityId_fXNzc_2014-05-01T00:00:00_Attempt=1
```

針對 ShellCommandActivity，stderr 和與這些活動相關聯 stdout 的日誌都會存放在每一次嘗試的目錄中。

針對資源 (例如 EmrCluster)，若有設定 emrLogUri，則該值會具有較高的優先順序。否則，資源 (包括這些資源的記錄TaskRunner檔) 會遵循上述管線記錄結構。

若要檢視指定管線執行的記錄：

1. ObjectId通過調用query-objects以獲取確切的對象 ID 來檢索。例如：

   ```
   aws datapipeline query-objects --pipeline-id <pipeline-id> --sphere ATTEMPT --region
    ap-northeast-1
   ```

   query-objects是一個分頁 CLI，如果給定的執行更多，則可能返回一個分頁令牌。pipeline-id您可以使用令牌來完成所有嘗試，直到找到預期的對象。例如，返回的外觀ObjectId如下所示：@TableBackupActivity_2023-05-020T18:05:18_Attempt=1。
2. 使用ObjectId，使用以下命令擷取記錄位置：

```
aws datapipeline describe-objects –pipeline-id <pipeline-id> --object-ids <object-id>
 --query "pipelineObjects[].fields[?key=='@logLocation'].stringValue"
```

失敗活動的錯誤訊息

要獲取錯誤消息，請首先ObjectId使用query-objects.

擷取失敗後ObjectId，請使用 describe-objects CLI 取得實際的錯誤訊息。

```
aws datapipeline describe-objects --region ap-northeast-1 --pipeline-id
 <pipeline-id> --object-ids <object-id> --query "pipelineObjects[].fields[?
key=='errorMessage'].stringValue"
```

取消或重新執行或標記為已完成物件

使用 set-status CLI 取消執行中的物件，或重新執行失敗的物件，或將執行中的物件標示為已完成。

首先，使用 query-objects CLI 取得物件識別碼。例如：

```
aws datapipeline query-objects --pipeline-id <pipeline-id> --sphere INSTANCE --region
 ap-northeast-1
```

使用 set-status CLI 變更所需物件的狀態。例如：

```
aws datapipeline set-status –pipeline-id <pipeline-id> --region ap-northeast-1 --status
 TRY_CANCEL --object-ids <object-id>
```

# 編輯您的管道

若要變更您其中一個管道的某些部分，您可以更新它的管道定義。在您變更執行中的管道後，您必須重新啟用管道，變更才會生效。此外，您可以重新執行一或多個管道元件。

目錄

## 限制

當管道處於PENDING狀態且未啟動時，您無法對其進行任何變更。在您啟用管道後，您可以編輯管道，但有以下限制。您所做的變更會在您儲存他們並再次啟用管道後，套用到管道物件的新執行。

- 您無法移除物件
- 您無法變更現有物件的排程期間
- 您無法在現有物件中新增、刪除或修改參考欄位
- 您無法參考新物件輸出欄位中現有的物件
- 您無法變更物件的排程啟動日期 (而是改為使用特定的日期和時間來啟動管道)

## 使用 AWS CLI 編輯管道

您可以使用命令列工具編輯管道。

首先，使用get-pipeline-definition指令下載目前管線定義的副本。這樣一來，您可以確認您修改的是最新的管道定義。以下範例會使用印出，來將管道定義印出到標準輸出 (stdout)。

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE
```

將管道定義儲存到檔案，並視需要進行編輯。使用put-pipeline-definition指令更新管線定義。以下範例會上傳更新後的管道定義檔案。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

您可以使用 `get-pipeline-definition` 命令再次擷取管道定義，來確認更新已成功。若要啟用管道，請使用以下的 activate-pipeline 命令：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

若您偏好的話，您可以使用 `--start-timestamp` 選項從特定日期和時間啟用管道，如下所示：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --start-
timestamp YYYY-MM-DDTHH:MM:SSZ
```

若要重新執行一或多個管道元件，請使用 set-status 命令。

# 複製您的管道

複製會建立管道的複本，讓您指定新管道的名稱。您可以複製處於任何狀態的管道，即使其包含錯誤也一樣；但是，新的管道會持續處於 PENDING 狀態，直到您手動啟用它為止。針對新的管道，複製操作會使用原始管道定義的最新版本，而非作用中的版本。在複製操作中，原始管道的完整排程不會複製到新的管道，而只會複製期間設定。

若要使用 AWS CLI 複製管線：

1. 使用新名稱和唯一 ID 建立新管線。請注意傳回的管線 ID。
2. 使用 `get-pipeline-definition` CLI 取得要複製之現有管線的管線定義，並將其寫入暫存檔案。請注意檔案的絕對路徑。
3. 使用 `put-pipeline-definition` CLI 將管線定義從現有管線複製到新配管。
4. 使用 `get-pipeline-definition` CLI 取得新管線的定義，以驗證管線定義。

```
# Create Pipeline (returns <new-pipeline-id>)
aws datapipeline create-pipeline --name my-cloned-pipeline --unique-id my-cloned-
pipeline --region ap-northeast-1

#Get pipeline definition of existing pipeline
aws datapipeline get-pipeline-definition --pipeline-id <existing-pipeline-id> --
region ap-northeast-1 > existing_pipeline_definition.json

# Put pipeline definition to new pipeline
aws datapipeline put-pipeline-definition --pipeline-id <new-
pipeline-id> --region ap-northeast-1 --pipeline-definition file://
<absolute_path_to_existing_pipeline_definition.json>

# get pipeline definition of new pipeline
aws datapipeline get-pipeline-definition --pipeline-id <new-pipeline-id> --region
 ap-northeast-1
```

# 標記您的管道

標籤是區分大小寫的鍵/值對，由鍵和選擇性的值組成，兩者皆由使用者定義。您可以為每個管道最多套用十個標籤。每個管道的標籤鍵必須是唯一的。如果所新增的標籤，其鍵已經和管道建立關聯，則此動作會更新該標籤的值。

將標籤套用至管道也會將標籤傳播到其基礎資源 (例如，Amazon EMR 叢集和 Amazon EC2 執行個體)。但是，它不會將這些標籤套用到處於 FINISHED 狀態中的資源，或是處於終止狀態的資源。若需要的話，您可以使用 CLI 將標籤套用到這些資源。

使用標籤完畢後，您可以從管道移除它。

使用 AWS CLI 標記您的管道

若要將標籤新增到新的管道，請將 `--tags` 選項新增到您的 create-pipeline 命令。例如，以下選項會建立一個管道，其帶有兩個標籤：一個 `environment` 標籤，其值為 `production`；另一個 `owner` 標籤，其值為 `sales`。

```
--tags key=environment,value=production key=owner,value=sales
```

若要將標籤新增到現有的管道，請使用 add-tags 命令，如下所示：

```
aws datapipeline add-tags --pipeline-id df-00627471SOVYZEXAMPLE --tags
 key=environment,value=production key=owner,value=sales
```

若要從現有的管道移除標籤，請使用 remove-tags 命令，如下所示：

```
aws datapipeline remove-tags --pipeline-id df-00627471SOVYZEXAMPLE --tag-keys
 environment owner
```

# 停用您的管道

停用執行中的管道會暫停管道執行。若要繼續管道執行，您可以啟用管道。這可讓您進行變更。例如，若您要將資料寫入已排程進行維護的資料庫，您可以停用管道，等待維護完成，然後啟用管道。

當您停用管道時，您可以指定要對執行中活動採取的動作。根據愈設，這些活動會立即取消。或者，您可以讓 AWS Data Pipeline 等待活動完成，再停用管道。

當您啟用停用的管道時，您可以指定其繼續的時間。使用 AWS CLI 或 API，根據預設，管道會從最後一次完成的執行繼續，或是您可以指定要繼續管道的日期和時間。

## 使用 AWS CLI 停用您的管道

請使用以下的 deactivate-pipeline 命令來停用管道：

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

若要在所有執行中的活動完成之後再停用管道，請新增 `--no-cancel-active` 選項，如下所示：

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --no-cancel-active
```

當您準備好時，您可以使用以下的 activate-pipeline 命令，從停止的位置繼續執行管道：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

若要從特定的日期和時間啟動管道，請新增 `--start-timestamp` 選項，如下所示：

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --start-timestamp YYYY-MM-DDTHH:MM:SSZ
```

## 刪除您的管道

當您不再需要管道時 (例如管道是在應用程式測試期間建立的)，建議您刪除它來從經常性使用中移除它。刪除管道會使其進入刪除中狀態。當管道處於已刪除狀態時，管道定義和執行歷史記錄便已移除。因此，您無法繼續在管道上執行操作 (包含描述它)。

> **Important**
>
> 您無法在刪除管道後還原它，因此請先確認您未來不再需要它，再進行刪除。

使用 AWS CLI 刪除管道

若要刪除管道，請使用 delete-pipeline 命令。以下命令會刪除指定的管道。

```
aws datapipeline delete-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

## 使用管道活動預備資料和資料表

AWS Data Pipeline 可以在您的管道中預備輸入和輸出資料，讓使用特定活動 (例如 `ShellCommandActivity` 和 `HiveActivity`) 更為容易。

資料預備可讓您將資料從輸入資料節點複製到執行活動的資源，並且以相似的方式，從資源複製到輸出資料節點。

Amazon EMR 或 Amazon EC2 資源上的暫存資料可透過在活動的殼層命令或 Hive 指令碼中使用特殊變數來取得。

資料表預備與資料預備相似，其不同處在於其預備的資料會特別採取資料庫資料表的形式。

AWS Data Pipeline 支援下列預備案例：

- 使用 `ShellCommandActivity` 進行資料預備
- 使用 Hive 及支援預備的資料節點進行資料表預備
- 使用 Hive 及不支援預備的資料節點進行資料表預備

> **Note**
>
> 預備只有在活動 (例如 `ShellCommandActivity`) 上的 `stage` 欄位設為 `true` 時才能運作。如需詳細資訊，請參閱ShellCommandActivity。

此外，資料節點和活動可以透過四種方式相關：

在資源上於本機預備資料

輸入資料會自動複製到資源的本機檔案系統。輸出資料會自動從資源的本機檔案系統複製到輸出資料節點。例如，當您設定 `ShellCommandActivity` 輸入和輸出，並設定 staging = true 時，輸入資料可透過 INPUTx_STAGING_DIR 取得，輸出資料則可透過 OUTPUTx_STAGING_DIR 取得，其中 x 是輸入和輸出的數字。

活動的預備輸入及輸出定義

輸入資料格式 (資料行名稱和資料表名稱) 會自動複製到活動的資源。例如，當您設定 `HiveActivity`，並設定 staging = true 時。輸入 `S3DataNode` 上指定的資料格式會用來從 Hive 資料表預備資料表定義。

未啟用預備

活動可以取得輸入和輸出物件及其欄位，但無法取得資料本身。例如，根據預設的 `EmrActivity`，或是當您以 staging = false 設定其他活動時。在此組態中，活動可透過使用 AWS Data Pipeline 表達式語法來取得資料欄位並參考他們，而這只會在滿足依存項目時才會發生。其用途僅只是檢查依存項目。活動中的程式碼會負責將資料從輸入複製到執行活動的資源。

物件之間的依存項目關係

兩個物件之間存在一種依存關係，這會在未啟用預備時導致類似的情況。這會使資料節點或活動做為執行另一個活動的先決條件。

## 資料暫存 ShellCommandActivity

考慮搭配 S3DataNode 物件做為資料輸入和輸出，使用 ShellCommandActivity 的案例。AWS Data Pipeline 會使用環境變數 ${INPUT1_STAGING_DIR} 和 ${OUTPUT1_STAGING_DIR} 自動預備資料節點，使其提供給殼層命令存取，就好像他們是本機檔案資料夾，如以下範例所示。名為 INPUT1_STAGING_DIR 和 OUTPUT1_STAGING_DIR 變數的數字部分，會根據您活動參考的資料節點數累加。

> **Note**
>
> 此案例只有在您的資料輸入和輸出為 S3DataNode 物件時，才會以說明的方式運作。此外，只有在輸出 S3DataNode 物件上有設定 directoryPath 時，才允許輸出資料預備。

```
{
  "id": "AggregateFiles",
  "type": "ShellCommandActivity",
  "stage": "true",
  "command": "cat ${INPUT1_STAGING_DIR}/part* > ${OUTPUT1_STAGING_DIR}/aggregated.csv",
  "input": {
    "ref": "MyInputData"
  },
  "output": {
    "ref": "MyOutputData"
  }
},
{
  "id": "MyInputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://my_bucket/source/#{format(@scheduledStartTime,'YYYY-MM-
dd_HHmmss')}/items"
  }
},
```

```
{
  "id": "MyOutputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://my_bucket/destination/#{format(@scheduledStartTime,'YYYY-MM-
dd_HHmmss')}"
  }
},
...
```

## 使用 Hive 及支援預備的資料節點進行資料表預備

考慮搭配 S3DataNode 物件做為資料輸入和輸出，使用 HiveActivity 的案例。AWS Data Pipeline 會使用變數 ${input1} 和 ${output1} 自動預備資料節點，使其提供給 Hive 指令碼存取，就好像他們是 Hive 資料表，如以下 HiveActivity 範例所示。名為 input 和 output 變數的數字部分，會根據您活動參考的資料節點數累加。

> **Note**
>
> 此案例只有在您的資料輸入和輸出為 S3DataNode 或 MySqlDataNode 物件時，才會以說明的方式運作。DynamoDBDataNode 不支援資料表預備。

```
{
  "id": "MyHiveActivity",
  "type": "HiveActivity",
  "schedule": {
    "ref": "MySchedule"
  },
  "runsOn": {
    "ref": "MyEmrResource"
  },
  "input": {
    "ref": "MyInputData"
  },
  "output": {
    "ref": "MyOutputData"
  },
  "hiveScript": "INSERT OVERWRITE TABLE ${output1} select * from ${input1};"
```

```
},
{
  "id": "MyInputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://test-hive/input"
  }
},
{
  "id": "MyOutputData",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://test-hive/output"
  }
},
...
```

## 使用 Hive 及不支援預備的資料節點進行資料表預備

考慮搭配 DynamoDBDataNode 做為資料輸入，S3DataNode 物件做為輸出，使用 HiveActivity 的案例。沒有可用的資料暫存DynamoDBDataNode，因此您必須先在 Hive 指令碼中使用變數名稱#{input.tableName}來參考 DynamoDB 表格，手動建立表格。如果 DynamoDB 表是輸出，則適用類似的命名法，除非您使用變數。#{output.tableName}預備可供此範例中的輸出 S3DataNode 物件使用，因此您可以以 ${output1} 參考輸出資料節點。

> **Note**
>
> 在此範例中，資料表名稱變數具有 # (井字) 字元前綴，因為 AWS Data Pipeline 使用表達式來存取 tableName 或 directoryPath。如需表達式在 AWS Data Pipeline 中評估方式的詳細資訊，請參閱表達式評估。

```
{
  "id": "MyHiveActivity",
  "type": "HiveActivity",
  "schedule": {
    "ref": "MySchedule"
```

```
  },
  "runsOn": {
    "ref": "MyEmrResource"
  },
  "input": {
    "ref": "MyDynamoData"
  },
  "output": {
    "ref": "MyS3Data"
  },
  "hiveScript": "-- Map DynamoDB Table
SET dynamodb.endpoint=dynamodb.us-east-1.amazonaws.com;
SET dynamodb.throughput.read.percent = 0.5;
CREATE EXTERNAL TABLE dynamodb_table (item map<string,string>)
STORED BY 'org.apache.hadoop.hive.dynamodb.DynamoDBStorageHandler'
TBLPROPERTIES ("dynamodb.table.name" = "#{input.tableName}");
INSERT OVERWRITE TABLE ${output1} SELECT * FROM dynamodb_table;"
},
{
  "id": "MyDynamoData",
  "type": "DynamoDBDataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "tableName": "MyDDBTable"
},
{
  "id": "MyS3Data",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "directoryPath": "s3://test-hive/output"
  }
},
...
```

# 在多個區域中搭配資源使用管道

根據預設，Ec2Resource 和 EmrCluster 資源會在與 AWS Data Pipeline 相同的區域中執行，但是 AWS Data Pipeline 支援跨多個區域協調資料流程，例如在一個區域中執行資源來整合來自其他區域的輸入資料。透過允許資源執行指定區域，您也可以獲得彈性，共置您的資源及其依存的資料集，並藉

由減少延遲和避免跨區域數據傳輸費來最大化效能。您可以在 Ec2Resource 和 EmrCluster 上使用 region 欄位來設定在與 AWS Data Pipeline 不同的區域中執行資源。

下列範例管線 JSON 檔案示範如何在歐洲 (愛爾蘭) 區域中執行EmrCluster資源，假設叢集有大量要處理的資料位於相同的區域中。在此範例中，與典型管道的差異在於 EmrCluster 的 region 欄位已設為 eu-west-1。

```
{
  "objects": [
    {
      "id": "Hourly",
      "type": "Schedule",
      "startDateTime": "2014-11-19T07:48:00",
      "endDateTime": "2014-11-21T07:48:00",
      "period": "1 hours"
    },
    {
      "id": "MyCluster",
      "type": "EmrCluster",
      "masterInstanceType": "m3.medium",
      "region": "eu-west-1",
      "schedule": {
        "ref": "Hourly"
      }
    },
    {
      "id": "MyEmrActivity",
      "type": "EmrActivity",
      "schedule": {
        "ref": "Hourly"
      },
      "runsOn": {
        "ref": "MyCluster"
      },
      "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar,-input,s3n://
elasticmapreduce/samples/wordcount/input,-output,s3://eu-west-1-bucket/wordcount/
output/#{@scheduledStartTime},-mapper,s3n://elasticmapreduce/samples/wordcount/
wordSplitter.py,-reducer,aggregate"
    }
  ]
}
```

下表會列出您可以選擇的區域，以及用於 region 欄位的相關聯區域代碼。

> **Note**
>
> 下列清單包含AWS Data Pipeline可協調工作流程和啟動 Amazon EMR 或 Amazon EC2 資源的區域。 AWS Data Pipeline這些地區可能不支援。如需支援 AWS Data Pipeline 的區域資訊，請參閱 AWS 區域與端點。

| 區域名稱 | 區域代碼 |
| --- | --- |
| 美國東部 (維吉尼亞北部) | us-east-1 |
| 美國東部 (俄亥俄) | us-east-2 |
| 美國西部 (加州北部) | us-west-1 |
| 美國西部 (奧勒岡) | us-west-2 |
| 加拿大 (中部) | ca-central-1 |
| 歐洲 (愛爾蘭) | eu-west-1 |
| 歐洲 (倫敦) | eu-west-2 |
| 歐洲 (法蘭克福) | eu-central-1 |
| 亞太區域 (新加坡) | ap-southeast-1 |
| 亞太區域 (雪梨) | ap-southeast-2 |
| 亞太區域 (孟買) | ap-south-1 |
| 亞太區域 (東京) | ap-northeast-1 |
| 亞太區域 (首爾) | ap-northeast-2 |
| 南美洲 (聖保羅) | sa-east-1 |

# 串聯失敗和重新執行

AWS Data Pipeline 允許您設定管道物件在依存項目失敗或使用者取消時的行為。您可以確認故障已串聯至其他管道物件 (消費者)，避免無限期的等待。所有活動、資料節點和先決條件都擁有名為 failureAndRerunMode 的欄位，其預設值為 none。若要啟用串聯失敗，請將 failureAndRerunMode 欄位設為 cascade。

啟用此欄位時，若管道物件陷於 WAITING_ON_DEPENDENCIES 狀態，且任何依存項目都已在沒有擱置中命令的狀態下失敗，便會發生串聯故障。在串聯故障期間，會發生下列事件：

- 物件失敗時，消費者會設為 CASCADE_FAILED，且原始物件和其消費者的先決條件都會設為 CANCELED。
- 任何已 FINISHED、FAILED 或 CANCELED 的物件都會遭到忽略。

除了和原始物件相關聯的先決條件，串聯故障不會在失敗物件的依存項目 (上游) 上運作。受到串聯故障影響的管道物件可能會觸發任何重試或後續動作，例如 onFail。

串聯故障的詳細效果取決於物件類型。

## 活動

若有任何一個依存項目失敗，活動便會變更為 CASCADE_FAILED，並在活動的消費者中觸發串聯故障。若活動依存的資源失敗，則活動會進入 CANCELED 狀態，且其所有的消費者都會變更為 CASCADE_FAILED。

## 資料節點和先決條件

若資料節點已設為失敗活動的輸出，則資料節點會變更為 CASCADE_FAILED 狀態。資料節點故障會散佈到任何相關聯的先決條件，且這些條件都會變更為 CANCELED 狀態。

## 資源

若依存資源的物件處於 FAILED 狀態，而資源本身處於 WAITING_ON_DEPENDENCIES 狀態，則資源會變更為 FINISHED 狀態。

## 重新執行串聯失敗的物件

根據預設，重新執行任何活動或資料節點只會重新執行相關聯的資源。但是，若在管道物件上將 `failureAndRerunMode` 欄位設為 `cascade`，則可允許目標物件上的重新執行命令在下列條件下散佈到所有消費者：

- 目標物件的消費者處於 CASCADE_FAILED 狀態。
- 目標物件的依存項目沒有任何擱置中的重新執行命令。
- 目標物件的依存項目並非處於 FAILED、CASCADE_FAILED 或 CANCELED 狀態。

若您嘗試重新執行 CASCADE_FAILED 物件，而其任何一個依存項目處於 FAILED、CASCADE_FAILED 或 CANCELED 狀態，則重新執行會失敗，並使物件返回 CASCADE_FAILED 狀態。若要成功重新執行失敗的物件，您必須向上追蹤依存項目的鏈結，找到故障的原始來源，並改為重新執行該物件。當您在資源上發出重新執行命令時，您也會嘗試重新執行任何依存於它的物件。

## 級聯故障和回填

若您啟用了串聯故障，並擁有建立許多回填的管道，則管道執行時間錯誤可能會造成資源快速地連續建立及刪除，而無法執行有用的工作。AWS Data Pipeline 會在您儲存管道時使用以下警告訊息，嘗試提醒您發生此情況： *`Pipeline_object_name`* `has 'failureAndRerunMode' field set to 'cascade' and you are about to create a backfill with scheduleStartTime` *`start_time`*`. This can result in rapid creation of pipeline objects in case of failures.` 其發生的原因在於串聯故障可能會快速地將下游活動設為 CASCADE_FAILED，並關閉不再需要的 EMR 叢集和 EC2 資源。我們建議您使用較短的時間範圍測試管道，來限制此情況造成的影響。

# 管線定義檔案語法

本節中的說明適用於使用 AWS Data Pipeline 命令列界面 (CLI) 手動操作管道定義檔案。這是使用 AWS Data Pipeline 主控台以互動方式設計管道的替代方式。

您可以使用任何支援以 UTF-8 檔案格式儲存檔案的文字編輯器手動建立管道定義檔案，並使用 AWS Data Pipeline 命令列界面提交檔案。

AWS Data Pipeline 也支援在管道定義中使用各種複雜的表達式和函數。如需詳細資訊，請參閱管道表達式和函數。

## 檔案結構

建立管道的第一個步驟是在管道定義檔案中撰寫管道定義物件。以下範例會說明管道定義檔案的一般結構。此檔案會定義兩個物件，以 '{' 和 '}' 及逗號分隔。

在以下範例中，第一個物件會定義兩個名稱值對，稱為「欄位」。第二個物件定義三個欄位。

```
{
  "objects" : [
    {
       "name1" : "value1",
       "name2" : "value2"
    },
    {
       "name1" : "value3",
       "name3" : "value4",
       "name4" : "value5"
    }
  ]
}
```

建立管道定義檔案時，您必須選取您需要的管道物件類型，將他們新增到管道定義檔案，然後新增適當的欄位。如需管道物件的詳細資訊，請參閱管道物件參考。

例如，您可以為輸入資料節點建立管道定義物件，並為輸出資料節點建立另一個物件。然後為活動建立另一個管道定義物件，例如使用 Amazon EMR 處理輸入資料。

## 管道欄位

在您了解要將哪些物件類型包含在管道定義檔案中後，您可以將欄位新增到每個管道物件的定義。欄位名稱會包在引號中，並以空格、冒號和空格與欄位值區隔，如以下範例所示。

```
"name" : "value"
```

欄位值可以是文字字串、另一個物件的參考、函數呼叫、表達式，或是任何上述類型的排序清單。如需可用於欄位值資料類型的詳細資訊，請參閱簡單資料類型。如需可用來評估欄位值函數的詳細資訊，請參閱表達式評估。

欄位限制為 2048 個字元。物件大小可為 20 KB，這表示您無法將許多大型欄位新增到物件。

每個管道物件都必須包含下列欄位：id 和 type，如以下範例所示。根據物件類型，可能還需要其他欄位。為 id 選取有意義的值，且該值在管道定義中必須是唯一的。type 的值則會指定物件類型。指定其中一個支援的管道定義物件類型，如管道物件參考主題中所列。

```
{
  "id": "MyCopyToS3",
  "type": "CopyActivity"
}
```

如需每個物件必要及選用欄位的詳細資訊，請參閱物件的文件。

若要在一個物件中包含來自另一個物件的欄位，請使用 parent 欄位，並參考該物件。例如，物件 "B" 包含其欄位 ("B1" 和 "B2")，以及來自物件 "A" 的欄位 ("A1" 和 "A2")。

```
{
  "id" : "A",
  "A1" : "value",
  "A2" : "value"
},
{
  "id" : "B",
  "parent" : {"ref" : "A"},
  "B1" : "value",
  "B2" : "value"
}
```

您可以使用 ID "Default"，在物件中定義常用欄位。這些欄位會自動包含在管道定義檔案中每個未明確設定參考不同物件 parent 欄位的物件內。

```
{
  "id" : "Default",
  "onFail" : {"ref" : "FailureNotification"},
  "maximumRetries" : "3",
  "workerGroup" : "myWorkerGroup"
}
```

## 使用者定義

您可以在您的管道元件上建立使用者定義或自訂欄位，並使用表達式參考他們。下列範例顯示名為myCustomField並my_customFieldReference新增至 S3 DataNode 物件的自訂欄位：

```
{
  "id": "S3DataInput",
  "type": "S3DataNode",
  "schedule": {"ref": "TheSchedule"},
  "filePath": "s3://bucket_name",
  "myCustomField": "This is a custom value in a custom field.",
  "my_customFieldReference": {"ref":"AnotherPipelineComponent"}
  },
```

使用者定義欄位的名稱都必須加上全部小寫的 "my" 前綴，並接續大寫字母或底線字元。此外，使用者定義欄位可以是字串值 (例如上述的 myCustomField 範例)，或是參考其他管道元件 (例如上述的 my_customFieldReference 範例)。

> **Note**
>
> 在使用者定義欄位上，AWS Data Pipeline 只會檢查對其他管道元件的有效參考，而不會檢查任何您新增的自訂欄位字串值。

# 使用 API

> **Note**
>
> 如果不撰寫與 AWS Data Pipeline 互動的程式，即不需要安裝任何 AWS 開發套件。您可以使用主控台或命令列界面建立和執行管道。如需詳細資訊，請參閱「」設定 AWS Data Pipeline

編寫與互動的應用程式的最簡單方式AWS Data Pipeline或實施自定義任務運行程式，則使用其中一種 AWS 開發套件。AWS 開發套件提供的功能，可簡化從您慣用的程式設計環境呼叫 Web 服務 API。如需詳細資訊，請參閱 安裝 AWS 開發套件。

## 安裝 AWS 開發套件

AWS 軟體開發套件提供包裝 API 的功能，並協助處理許多連線詳細資訊，例如計算簽章、處理請求重試和錯誤處理。軟體開發套件還包含範本程式碼、教學和其他資源，協助您開始編寫呼叫 AWS 的應用程式。在軟體開發套件中呼叫包裝器函式可以大幅簡化撰寫 AWS 應用程式的過程。如需如何下載及使用 AWS 開發套件的詳細資訊，請前往範本程式碼與程式庫。

以下平台的軟體開發套件提供 AWS Data Pipeline 支援：

- 適用於 Java 的 AWS 開發套件
- 適用於 Node.js 的 AWS 開發套件
- 適用於 PHP 的 AWS 開發套件
- 適用於 Python 的 AWS 開發套件 (Boto)
- 適用於 Ruby 的 AWS 開發套件
- 適用於 .NET 的 AWS 開發套件

## 向 AWS Data Pipeline 提出 HTTP 請求

如需 AWS Data Pipeline 中程式設計物件的完整描述，請參閱 AWS Data Pipeline API 參考。

如果您未使用其中一種 AWS 開發套件，則可以使用 POST 請求方法，透過 HTTP 執行 AWS Data Pipeline 操作。POST 方法需要您在請求標頭中指定操作，並在請求內文中提供 JSON 格式的操作資料。

### HTTP 標頭內容

AWS Data Pipeline 需要 HTTP 請求標頭中的下列資訊：

- `host` AWS Data Pipeline 端點。

  如需端點資訊，請參閱區域和端點。
- `x-amz-date`您必須在 HTTP Date 標頭或 AWS x-amz-date 標頭提供時間戳記。(有些 HTTP 用戶端程式庫不讓您設定 Date 標頭)。有 x-amz-date 標頭時，系統會在請求身份驗證時略過任何 Date 標頭。

  日期必須使用 HTTP/1.1 RFC 所指定之下列三種格式中的其中一種來指定：
  - Sun, 06 Nov 1994 08:49:37 GMT (RFC 822，已於 RFC 1123 更新)
  - Sunday, 06-Nov-94 08:49:37 GMT (RFC 850，已於 RFC 1036 淘汰)
  - Sun Nov 6 08:49:37 1994 (ANSI C 的 asctime() 格式)
- `Authorization`AWS 使用的一組授權參數，以確保請求的有效性和真實性。如需建構這個標頭的詳細資訊，請參閱 Signature 第 4 版簽章程序。
- `x-amz-target` 請求的目標服務和資料操作，格式如下：`<<serviceName>>_<<API version>>.<<operationName>>`

  例如：`DataPipeline_20121129.ActivatePipeline`

- content-type指定 JSON 及其版本。例如：Content-Type: application/x-amz-json-1.0

以下是啟動管道的 HTTP 請求範例標頭。

```
POST / HTTP/1.1
host: https://datapipeline.us-east-1.amazonaws.com
x-amz-date: Mon, 12 Nov 2012 17:49:52 GMT
x-amz-target: DataPipeline_20121129.ActivatePipeline
Authorization: AuthParams
Content-Type: application/x-amz-json-1.1
Content-Length: 39
Connection: Keep-Alive
```

## HTTP 內文內容

HTTP 請求的內文包含 HTTP 請求標頭中所指定之操作的資料。資料必須根據每個 AWS Data Pipeline API 的 JSON 資料結構描述格式化。AWS Data Pipeline JSON 資料結構描述會定義每項操作可使用的資料和參數類型 (如比較運算子和列舉常數)。

格式化 HTTP 請求的內文

使用 JSON 資料格式，同時傳遞資料值和資料結構。使用括號符號，可以將元素巢套於其他元素內。以下範例顯示的請求，會放置由三個物件及其對應插槽構成的管道定義。

```
{
 "pipelineId": "df-00627471SOVYZEXAMPLE",
 "pipelineObjects":
  [
    {"id": "Default",
     "name": "Default",
     "slots":
      [
        {"key": "workerGroup",
         "stringValue": "MyWorkerGroup"}
      ]
    },
    {"id": "Schedule",
     "name": "Schedule",
```

```
     "slots":
      [
       {"key": "startDateTime",
         "stringValue": "2012-09-25T17:00:00"},
        {"key": "type",
         "stringValue": "Schedule"},
        {"key": "period",
         "stringValue": "1 hour"},
        {"key": "endDateTime",
         "stringValue": "2012-09-25T18:00:00"}
      ]
    },
    {"id": "SayHello",
     "name": "SayHello",
     "slots":
      [
        {"key": "type",
         "stringValue": "ShellCommandActivity"},
        {"key": "command",
         "stringValue": "echo hello"},
        {"key": "parent",
         "refValue": "Default"},
        {"key": "schedule",
         "refValue": "Schedule"}

      ]
    }
  ]
}
```

### 處理 HTTP 回應

以下為 HTTP 回應中一些重要的標頭，以及如何在應用程式中處理他們：

- HTTP/1.1-此標頭後面有狀態碼。代碼值 200 表示操作成功。任何其他值皆表示錯誤。
- x-amzn-RequestId— 此標頭包含一個請求 ID，如果您需要使用AWS Data Pipeline。請求 ID 範例為 K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG。
- x-amz-crc32—AWS Data Pipeline會計算 HTTP 承載的 CRC32 檢查總和，並在 x-amz-crc32 標頭中傳回此檢查總和。建議您在用戶端運算自己的 CRC32 檢查總和，並與 x-amz-crc32 標頭比較；如果檢查總和不相符，可能表示資料在傳輸過程中已損毀。如果發生這種情況，您應該重試您的請求。

AWS 開發套件使用者不需要手動執行此驗證，因為該開發套件會運算 Amazon DynamoDB 每個回覆的檢查總和，如果偵測到不符就會自動重試。

### 範例 AWS Data Pipeline JSON 請求和回應

以下範例說明建立新管道的請求。然後，它會顯示 AWS Data Pipeline 回應，包括新建立管道的管道識別符。

#### HTTP POST 請求

```
POST / HTTP/1.1
host: https://datapipeline.us-east-1.amazonaws.com
x-amz-date: Mon, 12 Nov 2012 17:49:52 GMT
x-amz-target: DataPipeline_20121129.CreatePipeline
Authorization: AuthParams
Content-Type: application/x-amz-json-1.1
Content-Length: 50
Connection: Keep-Alive

{"name": "MyPipeline",
 "uniqueId": "12345ABCDEFG"}
```

#### AWS Data Pipeline 回應

```
HTTP/1.1 200
x-amzn-RequestId: b16911ce-0774-11e2-af6f-6bc7a6be60d9
x-amz-crc32: 2215946753
Content-Type: application/x-amz-json-1.0
Content-Length: 2
Date: Mon, 16 Jan 2012 17:50:53 GMT

{"pipelineId": "df-00627471SOVYZEXAMPLE"}
```

# AWS Data Pipeline 中的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全是 AWS 與您共同肩負的責任。共同的責任模式將其稱為雲端的安全性和雲端中的安全性：

- 雲端本身的安全：AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 AWS 合規計劃的一部分。若要了解適用於 AWS Data Pipeline 的合規計畫，請參閱合規計畫的 AWS 服務範圍。
- 雲端內部的安全 – 您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 AWS Data Pipeline 時套用共同責任模型。下列主題說明如何將 AWS Data Pipeline 設定為達到您的安全及合規目標。您也將了解如何使用其他 AWS 服務來協助您監控並保護 AWS Data Pipeline 資源。

主題



## AWS Data Pipeline 的資料保護

AWS 共同的責任模型適用於 AWS Data Pipeline 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您必須負責維護在此基礎設施上託管之內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 憑證，並設定個別使用者。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶都使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。建議使用 TLS 1.2 或更新版本。
- 使用 AWS CloudTrail 設定 API 和使用者活動記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務 內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。
- AWS Data Pipeline支援適用於亞馬遜 EMR 和亞馬 Amazon EC2 資源的 IMDSv2。若要將 IMDSv2 與亞馬遜 EMR 搭配使用，請使用 5.23.1、5.27.1 或 5.32 或更新版本或 6.2 版或更新版本。如需詳細資訊，請參閱設定傳送至 Amazon EC2 執行個體的中繼資料服務請求和使用 IMDSv2。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name (名稱) 欄位。這包括當您使用 AWS Data Pipeline 或使用主控台、API、AWS CLI 或 AWS 開發套件的其他 AWS 服務。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

# AWS Data Pipeline 的 Identity and Access Management

您的安全登入資料會在 AWS 服務中識別您，並授予讓您使用 AWS 資源的許可，例如您的管道。您可使用AWS Data Pipeline和AWS Identity and Access Management (IAM) 的功能來允許其他使用者AWS Data Pipeline和其他使用者存取您的AWS Data Pipeline資源但不共用您的安全登入資料。

組織可以共享管道的存取，讓該組織中的每個人都可以共同開發及維護管道。不過，您可能必須執行下列動作：

- 控制哪些使用者可以存取特定管道
- 保護生產管道以免錯誤編輯
- 允許稽核員具備管道的唯讀存取，但防止他們進行變更

AWS Data Pipeline與AWS Identity and Access Management（IAM）集成，它提供了廣泛的功能：

- 透過AWS 帳戶.
- 在AWS您的AWS 帳戶.
- 指派唯一安全登入資料給每位使用者。
- 控制每位使用者對服務與資源的存取。
- 為您的AWS 帳戶. 中的所有使用者取得單一帳單

搭配使用 IAMAWS Data Pipeline，您可以控制組織中的使用者是否可以使用特定的 API 動作來執行任務，以及是否可以使用特定的 AWS 資源。您可以使用以管道標籤和工作者群組為基礎的 IAM 政策，與其他使用者共用管道，並控制他們擁有的存取層級。

內容

- AWS Data Pipeline 的 IAM 政策
- 用於 AWS Data Pipeline 的政策範例
- AWS Data Pipeline 的 IAM 角色

## AWS Data Pipeline 的 IAM 政策

預設情況下，IAM 實體沒有可建立或修改 AWS 資源的許可。若要允許 IAM 實體建立或修改資源並執行任務，您必須建立 IAM 政策，授予 IAM 實體許可，允許其使用他們會需要的特定資源和 API 動作，然後將這些政策連接到需要這些許可的 IAM 實體。

將政策連接到使用者或使用者群組時，政策會允許或拒絕使用者在特定資源上執行特定任務的許可。如需 IAM 政策的一般詳細資訊，請參閱《IAM 使用者指南》中的許可及政策。如需管理和建立自訂 IAM 政策的詳細資訊，請參閱管理 IAM 政策。

內容

- 政策語法
- 使用標籤控制管道的存取
- 使用工作者群組控制管道的存取

### 政策語法

IAM 政策為包含一或多個陳述式的 JSON 文件。每個陳述式的結構如下所示：

```
{
```

```
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"*",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

政策陳述式是由下列元素組成：

- Effect (效果)：效果 可以是 Allow 或 Deny。預設情況下，IAM 實體沒有使用資源和 API 動作的許可，因此所有請求均會遭到拒絕。明確允許覆寫預設值。明確拒絕覆寫任何允許.
- Action (動作)：動作 是您授予或拒絕許可的特定 API 動作。如需的動作清單AWS Data Pipeline，請參閱 AWS Data PipelineAPI 參考中的動作。
- Resource (資源)：受動作影響的資源。這裡唯一有效的值為 "*"。
- Condition (條件)：條件為選擇性。您可以用以控制何時政策開始生效。

  AWS Data Pipeline 實作 AWS 通用的內容金鑰 (請參閱條件的可用金鑰)，和下列服務專屬的金鑰。

  - datapipeline:PipelineCreator— 授予管道的使用者存取權。如需範例，請參閱將完整存取授予管道擁有者。
  - datapipeline:Tag— 根據管線標記授與存取權限。如需詳細資訊，請參閱使用標籤控制管道的存取。
  - datapipeline:workerGroup— 根據 Worker 群組的名稱授與存取權。如需詳細資訊，請參閱使用工作者群組控制管道的存取。

## 使用標籤控制管道的存取

您可以建立 IAM 政策，參考管道的標籤。這可讓您使用管道標記來執行下列動作：

- 授予管道的唯讀存取
- 授予管道的讀取/寫入存取
- 防止存取管道

例如，假設管理員有兩個管道環境 (生產和開發)，而且每個環境有一個 IAM 群組。對於生產環境中的管道，管理員會授予生產 IAM 群組中的使用者讀取/寫入存取權限，但會將唯讀存取權授予開發人員 IAM 群組中的使用者。對於開發環境中的管道，管理員會授予生產和開發人員 IAM 群組的讀取/寫入存取權。

為了實現這種情況，管理員使用「環境 = 生產」標記生產管道標記，並將下列政策附加到開發人員 IAM 群組。第一個陳述式會授予所有管道的唯讀存取。第二個陳述式會授予沒有 "environment=production" 標籤之管道的讀取/寫入存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:ListPipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {"datapipeline:Tag/environment": "production"}
      }
    }
  ]
}
```

此外，管理員會將下列政策附加至生產 IAM 群組。此陳述式會授予所有管道的完整存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*"
```

```
    }
  ]
}
```

如需更多範例，請參閱根據標籤將唯讀存取授予使用者和根據標籤將完整存取授予使用者。

## 使用工作者群組控制管道的存取

您可以建立做為參考背景工作者群組名稱的 IAM 政策。

例如，假設管理員有兩個管道環境 (生產和開發)，而且每個環境有一個 IAM 群組。管理員有三個資料庫伺服器，並將其任務執行器分別設定用於生產、進入生產階段前和開發環境。管理員想要確保生產 IAM 群組中的使用者可以建立將任務推送到生產資源的管道，而且開發 IAM 群組中的使用者可以建立將任務推送到生產前和開發人員資源的管道。

為了達成此案例，管理員會使用生產登入資料在生產資源上安裝任務執行器，並將 `workerGroup` 設為 "prodresource"。此外，管理員會使用開發登入資料在開發資源上安裝任務執行器，並將 `workerGroup` 設為 "pre-production" 和 "development"。管理員會將下列政策附加至開發人員 IAM 群組，以封鎖對「prodresource」資源的存取。第一個陳述式會授予所有管道的唯讀存取。第二個陳述式會在工作者群組名稱含有 "dev" 或 "pre-prod" 前綴時，授予管道的讀取/寫入存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:ListPipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": "*"
    },
    {
      "Action": "datapipeline:*",
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "datapipeline:workerGroup": ["dev*","pre-prod*"]
        }
```

```
      }
    }
  ]
}
```

此外，管理員會將下列政策附加至生產 IAM 群組，以授予「prodresource」資源的存取權。第一個陳述式會授予所有管道的唯讀存取。第二個陳述式會在工作者群組名稱含有 "prod" 前綴時，授予讀取/寫入存取。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:ListPipelines",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "datapipeline:*",
      "Resource": "*",
      "Condition": {
        "StringLike": {"datapipeline:workerGroup": "prodresource*"}
      }
    }
  ]
}
```

# 用於 AWS Data Pipeline 的政策範例

下列範例示範如何將管道的完整存取或有限存取授予使用者。

內容

- 範例 4：授予使用者存取 AWS Data Pipeline 主控台

## 範例 1：根據標籤將唯讀存取授予使用者

下列政策可讓使用者使用唯讀 AWS Data Pipeline API 動作，但僅限於具有標籤 "environment=production" 的管道。

ListPipelines API 動作不支援標籤型授權。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:Describe*",
        "datapipeline:GetPipelineDefinition",
        "datapipeline:ValidatePipelineDefinition",
        "datapipeline:QueryObjects"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:Tag/environment": "production"
        }
      }
    }
  ]
}
```

## 範例 2：根據標籤將完整存取授予使用者

下列原則允許使用者使用所有AWS Data Pipeline API 動作 ListPipelines，但僅限具有標籤為「環境 = 測試」的管道。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "datapipeline:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:Tag/environment": "test"
        }
      }
    }
  ]
}
```

## 範例 3：將完整存取授予管道擁有者

下列政策可讓使用者使用所有 AWS Data Pipeline API 動作，但僅限於自己的管道。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "datapipeline:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "datapipeline:PipelineCreator": "${aws:userid}"
        }
      }
    }
  ]
}
```

## 範例 4：授予使用者存取 AWS Data Pipeline 主控台

下列政策可讓使用者使用 AWS Data Pipeline 主控台來建立及管理管道。

此政策針對連結至 AWS Data Pipeline 所需 roleARN 的特定資源，包含其 PassRole 許可的動作。如需有關以身分識別為基礎 (IAM)PassRole 權限的詳細資訊，請參閱部落格文章授予使用 IAM 角色 (權限) 啟動 EC2 執行個體的PassRole許可。

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Action": [
    "cloudwatch:*",
    "datapipeline:*",
    "dynamodb:DescribeTable",
    "elasticmapreduce:AddJobFlowSteps",
    "elasticmapreduce:ListInstance*",
    "iam:AddRoleToInstanceProfile",
    "iam:CreateInstanceProfile",
    "iam:GetInstanceProfile",
    "iam:GetRole",
    "iam:GetRolePolicy",
    "iam:ListInstanceProfiles",
    "iam:ListInstanceProfilesForRole",
    "iam:ListRoles",
    "rds:DescribeDBInstances",
    "rds:DescribeDBSecurityGroups",
    "redshift:DescribeClusters",
    "redshift:DescribeClusterSecurityGroups",
    "s3:List*",
    "sns:ListTopics"
   ],
   "Effect": "Allow",
   "Resource": [
    "*"
   ]
  },
  {
   "Action": "iam:PassRole",
   "Effect": "Allow",
   "Resource": [
    "arn:aws:iam::*:role/DataPipelineDefaultResourceRole",
    "arn:aws:iam::*:role/DataPipelineDefaultRole"
   ]
  }
 ]
}
```

# AWS Data Pipeline 的 IAM 角色

AWS Data Pipeline使用AWS Identity and Access Management角色。附加到 IAM 角色的許可政策決定了可以執行的動作AWS Data Pipeline和應用程式，以及它們可以存取的AWS資源。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 角色。

AWS Data Pipeline需要兩個 IAM 角色：

- 管道角色AWS Data Pipeline可控制對 AWS 資源的存取。在管線物件定義中，`role`欄位會指定此角色。
- EC2 執行個體角色可控制 EC2 執行個體上執行的應用程式 (包括 Amazon EMR 叢集中的 EC2 執行個體) 對AWS資源的存取。在管線物件定義中，`resourceRole`欄位會指定此角色。

> **Important**
>
> 如果您在 2022 年 10 月 3 日之前使用具有預設角色的AWS Data Pipeline主控台建立管道，請`DataPipelineDefaultRole`為您AWS Data Pipeline建立管道，並將`AWSDataPipelineRole`受管理的政策附加至該角色。自 2022 年 10 月 3 日起，`AWSDataPipelineRole`受管策略已被棄用，並且在使用控制台時必須為管道指定管道角色。
>
> 我們建議您檢閱現有配管，並判斷`DataPipelineDefaultRole`是否與管線相關聯，以及`AWSDataPipelineRole`是否已附加至該角色。如果是這樣，請檢閱此原則允許的存取權，以確保其適合您的安全性需求。視需要新增、更新或取代附加至此角色的原則和政策陳述式。或者，您可以更新管道以使用您使用不同權限原則建立的角色。

## AWS Data Pipeline角色權限原則範例

每個角色都有一或多個附加權限原則，用來決定角色可存取的AWS資源以及角色可執行的動作。本主題提供管線角色的權限原則範例。它也提供的內容`AmazonEC2RoleforDataPipelineRole`，也就是預設 EC2 執行個體角色的受管政策`DataPipelineDefaultResourceRole`。

管道角色許可政策範例

以下範例政策的範圍限定在於允許必AWS Data Pipeline要功能使用 Amazon EC2 和 Amazon EMR 資源執行管道。它還提供存取許可，例如 Amazon SimpleAWS Simple Simple Simple Simple Simple Simple Simple Notification Service 和 Amazon Simple Notification Service)。如果管線中定義的

物件不需要AWS服務的資源，強烈建議您移除存取該服務的權限。例如，如果您的管道未定義DynamoDBData 節點或使用SnsAlarm動作，建議您移除這些動作的 allow 陳述式。

- 將 *111122223333* 取代為您的 AWS 帳戶 ID。
- 以管線角色 (此原則所附加的角色) 的名稱取*NameOfDataPipelineRole*代。
- *NameOfDataPipelineResourceRole*以 EC2 執行個體角色的名稱取代。
- *us-west-1*以適合您應用程式的適當區域取代。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/NameOfDataPipelineRole",
                "arn:aws:iam::111122223333 :role/NameOfDataPipelineResourceRole"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotInstanceRequests",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteTags",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeDhcpOptions",
```

```
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcEndpointServices",
                "ec2:DescribeVpcs",
                "ec2:DetachNetworkInterface",
                "ec2:ModifyImageAttribute",
                "ec2:ModifyInstanceAttribute",
                "ec2:RequestSpotInstances",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:DescribeVolumeStatus",
                "ec2:DescribeVolumes",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:RunJobFlow",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:AddTags",
                "elasticmapreduce:RemoveTags",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ModifyInstanceGroups",
                "elasticmapreduce:GetCluster",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:ListInstances",
                "iam:ListInstanceProfiles",
                "redshift:DescribeClusters"
            ],
            "Resource": [
```

```
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "sns:GetTopicAttributes",
            "sns:Publish"
        ],
        "Resource": [
            "arn:aws:sns:us-west-1:111122223333:MyFirstSNSTopic",
            "arn:aws:sns:us-west-1:111122223333:MySecondSNSTopic",
            "arn:aws:sns:us-west-1:111122223333:AnotherSNSTopic"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:ListBucket",
            "s3:ListMultipartUploads"
        ],
        "Resource": [
          "arn:aws:s3:::MyStagingS3Bucket",
          "arn:aws:s3:::MyLogsS3Bucket",
          "arn:aws:s3:::MyInputS3Bucket",
          "arn:aws:s3:::MyOutputS3Bucket",
          "arn:aws:s3:::AnotherRequiredS3Buckets"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:GetObjectMetadata",
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::MyStagingS3Bucket/*",
            "arn:aws:s3:::MyLogsS3Bucket/*",
            "arn:aws:s3:::MyInputS3Bucket/*",
            "arn:aws:s3:::MyOutputS3Bucket/*",
            "arn:aws:s3:::AnotherRequiredS3Buckets/*"
        ]
    },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:Scan",
                "dynamodb:DescribeTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-west-1:111122223333:table/MyFirstDynamoDBTable",
                "arn:aws:dynamodb:us-west-1:111122223333:table/MySecondDynamoDBTable",
                "arn:aws:dynamodb:us-west-1:111122223333:table/AnotherDynamoDBTable"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "arn:aws:rds:us-west-1:111122223333:db:MyFirstRdsDb",
                "arn:aws:rds:us-west-1:111122223333:db:MySecondRdsDb",
                "arn:aws:rds:us-west-1:111122223333:db:AnotherRdsDb"
            ]
        }
    ]
}
```

EC2 執行個體角色的預設受管政策

的內容如下所示。AmazonEC2RoleforDataPipelineRole這是附加至AWS Data Pipeline、預設資源角色的受管理策略DataPipelineDefaultResourceRole。當您為管道定義資源角色時，建議您先使用此權限原則，然後移除不需要之AWS服務動作的權限。

此時會顯示原則的第 3 版，這是撰寫本文時的最新版本。使用 IAM 主控台檢視政策的最新版本。

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*",
        "datapipeline:*",
        "dynamodb:*",
        "ec2:Describe*",
```

```
        "elasticmapreduce:AddJobFlowSteps",
        "elasticmapreduce:Describe*",
        "elasticmapreduce:ListInstance*",
        "elasticmapreduce:ModifyInstanceGroups",
        "rds:Describe*",
        "redshift:DescribeClusters",
        "redshift:DescribeClusterSecurityGroups",
        "s3:*",
        "sdb:*",
        "sns:*",
        "sqs:*"
      ],
      "Resource": ["*"]
    }]
}
```

## 為角色權限建立AWS Data Pipeline和編輯 IAM 角色

使用下列程序建立AWS Data Pipeline使用 IAM 主控台的角色。此程序包含兩個步驟。首先，您會建立要附加到角色的許可政策。接下來，您將建立角色並附加政策。建立角色之後，您可以透過附加和卸離權限原則來變更角色的權限。

> **Note**
>
> 如下所述建立AWS Data Pipeline使用主控台的角色時，IAM 會建立並附加角色所需的適當信任政策。

若要建立與角色搭配使用的權限原則AWS Data Pipeline

1. 在 https://console.aws.amazon.com/iam/ 中開啟 IAM 主控台。
2. 在導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。
3. 請選擇 JSON 標籤。
4. 如果您要建立管線角色，請複製並貼上中原則範例的內容管道角色許可政策範例，並根據您的安全性需求進行適當編輯。或者，如果您要建立自訂 EC2 執行個體角色，請對中的範例執行相同的動作EC2 執行個體角色的預設受管政策。
5. 選擇 Review policy (檢閱政策)。
6. 輸入原則的名稱 (例如，`MyDataPipelineRolePolicy`選擇性的說明)，然後選擇 [建立原則]。

7. 請注意政策的名稱。在建立角色時，您需要它。

為 AWS Data Pipeline 建立 IAM 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇 Roles (角色)，然後選擇 Roles (建立角色)。
3. 在 [選擇使用案例] 下，選擇 [Data Pipeline]。
4. 在選取您的使用案例下方，執行以下其中一項操作：
   - 選擇`Data Pipeline`此選項可建立管線角色。
   - 選擇`EC2 Role for Data Pipeline`此選項可建立資源角色。
5. 選擇 Next: Permissions (下一步：許可)。
6. 如果列出的預設原則，請繼續執行下列步驟來建立角色，然後根據下一個程序中的指示對AWS Data Pipeline其進行編輯。否則，請輸入您在上述程序所建立的政策名稱，然後從清單中選取。
7. 選擇 [下一步:標記]，輸入要新增至角色的任何標記，然後選擇 [下一步:複查]。
8. 輸入角色的名稱 (例如，`MyDataPipelineRole`選擇性的說明)，然後選擇 [建立角色]。

附加或卸離 IAM 角色的許可政策AWS Data Pipeline

1. 前往網址 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇 Roles (角色)
3. 在搜尋方塊中，開始輸入您要編輯的角色名稱 (例如，DataPipelineDefaultRole或)，MyDataPipelineRole然後從清單中選擇「角色」名稱。
4. 在許可索引標籤上，執行以下操作：
   - 若要卸離權限原則，請在 [權限] 原則下，選擇原則項目最右邊的 [移除] 按鈕。當系統提示確認時，請選擇「分離
   - 若要附加您先前建立的策略，請選擇 [附加策略]。在搜尋方塊中，開始輸入您要編輯的政策名稱，從清單中選取它，然後選擇 Roles (連接政策)。

## 變更現有管道的角色

如果要將不同的管線角色或資源角色指派給管線，可以使用AWS Data Pipeline主控台中的架構師編輯器。

使用控制台編輯指派給管線的角色

1. 請在以下位置開啟AWS Data Pipeline主控台。 https://console.aws.amazon.com/datapipeline/
2. 從清單中選取配管，然後選擇「動作」>「編輯」。
3. 在建築編輯器的右窗格中，選擇「其他」。
4. 從 [資源角色與角色] 清單中，選擇您AWS Data Pipeline要指定的角色，然後選擇 [儲存]。

# AWS Data Pipeline 中的記錄和監控

AWS Data Pipeline與整合AWS CloudTrail，這項服務可提供由使用者、角色或中的AWS服務所採取之動作的記錄AWS Data Pipeline。 CloudTrail 將的所有 API 呼叫擷取AWS Data Pipeline為事件。擷取的呼叫包括從 AWS Data Pipeline 主控台進行的呼叫，以及針對 AWS Data Pipeline API 操作的程式碼呼叫。如果建立追蹤，則可將事件 (包括的 CloudTrail 事件) 持續交付至 Amazon S3 儲存貯體，包括的事件AWS Data Pipeline。即使您未設定追蹤記錄，仍然可以透過 CloudTrail 主控台中的 Event history (事件歷史記錄) 檢視最新的事件。您可以使用收集的資訊來 CloudTrail判斷提交給和的請求 AWS Data Pipeline、提出請求的 IP 地址、提出請求的對象、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱使AWS CloudTrail用者指南。

## AWS Data Pipeline中的資訊 CloudTrail

CloudTrail 當您建立AWS帳戶時，系統會在您的帳戶中啟用。當中發生活動時AWS Data Pipeline，該活動會記錄在事件中，其他AWS服務 CloudTrail 事件則記錄於 E vent history (事件歷史記錄)。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷程記錄檢視事件。

如需您 AWS 帳戶中正在進行事件的記錄 (包含 AWS Data Pipeline 的事件)，請建立線索。線索能 CloudTrail 讓日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您還能設定其他AWS服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定的 Amazon SNS 通知 CloudTrail
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

所有AWS Data Pipeline動作都會由 AWS Data Pipeline API 參考動作章節記錄， CloudTrail 並將其記錄在其中。例如，對CreatePipeline動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 角色登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail 使用者身分元素。

## 了解 AWS Data Pipeline 日誌檔項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付至您指定的 Amazon S3 儲存貯體。 CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。 CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範`CreatePipeline`操作：

```
    {
"Records": [
  {
    "eventVersion": "1.02",
    "userIdentity": {
      "type": "Root",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::aws-account-id:role/role-name",
      "accountId": "role-account-id",
      "accessKeyId": "role-access-key"
    },
    "eventTime": "2014-11-13T19:15:15Z",
    "eventSource": "datapipeline.amazonaws.com",
    "eventName": "CreatePipeline",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.196.64",
    "userAgent": "aws-cli/1.5.2 Python/2.7.5 Darwin/13.4.0",
    "requestParameters": {
      "name": "testpipeline",
      "uniqueId": "sounique"
```

```
      },
      "responseElements": {
        "pipelineId": "df-06372391ZG65EXAMPLE"
      },
      "requestID": "65cbf1e8-6b69-11e4-8816-cfcbadd04c45",
      "eventID": "9f99dce0-0864-49a0-bffa-f72287197758",
      "eventType": "AwsApiCall",
      "recipientAccountId": "role-account-id"
    },
      ...additional entries
  ]
}
```

# AWS Data Pipeline 的事件反應

AWS Data Pipeline 的事件反應是 AWS 責任。AWS 有控制事件反應的正式、記載政策和計劃。

AWS 服務運作狀態儀表板上會張貼可能產生廣泛影響的 AWS 操作問題。系統也會透過 Personal Health Dashboard，將操作問題張貼至個別帳戶。

# AWS Data Pipeline 的合規驗證

AWS Data Pipeline 不在任何 AWS 合規計劃的範圍內。如需特定合規計劃的 AWS 服務範圍清單，請參閱 合規計劃的 AWS 服務範圍。如需一般資訊，請參閱 AWS 合規計劃。

# AWS Data Pipeline 中的恢復能力

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域與可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

# AWS Data Pipeline 中的基礎設施安全

作為託管服務，AWS Data Pipeline受AWS全局網絡安全過程，詳情請參閱Amazon Web Services：安全流程概觀白皮書。

您可使用 AWS 發佈的 API 呼叫，透過網路存取 AWS Data Pipeline。用戶端必須支援 Transport Layer Security (TLS) 1.0 或更新版本。建議使用 TLS 1.2 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 委託人相關聯的私密存取金鑰來簽署。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

# AWS Data Pipeline 中的組態與漏洞分析

組態和 IT 控制是 AWS 與身為我們客戶的您共同的責任。如需詳細資訊，請參閱 AWS 共同的責任模型。

# 教學課程

下列自學課程將step-by-step逐步引導您完成與配管建立和使用配管的過程AWS Data Pipeline。

教學課程

- 使用 Amazon EMR 與 Hadoop 流媒體處理數據
- 在亞馬遜 S3 儲存貯體之間複製 CSV 資料 AWS Data Pipeline
- 使用將 MySQL 數據導出到亞馬遜 S3 AWS Data Pipeline
- 使用將數據複製到亞馬遜紅移 AWS Data Pipeline

# 使用 Amazon EMR 與 Hadoop 流媒體處理數據

您可以使用 AWS Data Pipeline 來管理您的 Amazon EMR 叢集。使用時， AWS Data Pipeline 您可以指定叢集啟動前必須符合的先決條件 (例如，確保今天的資料已上傳到 Amazon S3)、重複執行叢集的排程，以及要使用的叢集組態。以下教學會逐步解說如何啟動簡單的叢集。

在本教學中，您會為簡單的 Amazon EMR 叢集建立管道，以執行 Amazon 提供的預先存在的 Hadoop 串流任務，EMR並在任務成功完成後傳送 Amazon SNS 通知。您可以使用由提供的 Amazon EMR 叢集資源來執 AWS Data Pipeline 行此任務。範例應用程式即會呼叫 WordCount，也可以從 Amazon EMR 主控台手動執行。請注意，由您代表產生 AWS Data Pipeline 的叢集會顯示在 Amazon EMR 主控台中，並向您的AWS帳戶收費。

管道物件

管道會使用下列物件：

EmrActivity

定義要在管道中執行的工作（運行 Amazon EMR 提供的預先存在的 Hadoop 流任務）。

EmrCluster

AWS Data Pipeline 用於執行此活動的資源。

叢集是一組 Amazon EC2 執行個體。 AWS Data Pipeline 啟動叢集，然後在工作完成後終止叢集。

排程

此活動的開始日期、時間和持續時間。您可以選擇性地指定結束日期和時間。

SnsAlarm

在任務成功完成後，將 Amazon SNS 通知傳送到您指定的主題。

目錄



## 開始之前

請務必完成下列步驟。

- 完成 設定 AWS Data Pipeline 中的任務。
- (選擇性) VPC 為叢集設定一個，並為VPC.
- 建立傳送電子郵件通知的主題，並記下 Amazon 資源名稱 (ARN) 主題。如需詳細資訊，請參閱 Amazon 簡單通知服務入門指南中的建立主題。

## 使用命令列啟動叢集

如果您定期執行 Amazon EMR 叢集來分析 Web 日誌或對科學資料執行分析，則可以用 AWS Data Pipeline 來管理 Amazon EMR 叢集。使用時 AWS Data Pipeline，您可以指定叢集啟動前必須符合的先決條件 (例如，確保今天的資料已上傳到 Amazon S3)。本教學將逐步引導您啟動叢集，該叢集可作為簡單 Amazon EMR 管道的模型，或是作為參與更多管道的一部分。

必要條件

您必須先完成下列步驟CLI，才能使用：

1. 安裝和配置命令行界面（CLI）。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 請確定已命名DataPipelineDefaultRole且DataPipelineDefaultResourceRole存在的IAM角色。主 AWS Data Pipeline 控台會自動為您建立這些角色。如果您至少沒有使用 AWS Data Pipeline 主控台一次，則必須手動建立這些角色。如需詳細資訊，請參閱 AWS Data Pipeline 的 IAM 角色。

任務

- 監控管道執行

## 建立管道定義檔案

下面的代碼是一個簡單的 Amazon 集群的管道定義文件，該EMR集群運行 Amazon 提供的現有 Hadoop 流任務。EMR此範例應用程式已呼叫 WordCount，您也可以使用 Amazon EMR 主控台執行它。

將此程式碼複製到文字檔，並儲存為 `MyEmrPipelineDefinition.json`。您應該將 Amazon S3 儲存貯體位置取代為您擁有的 Amazon S3 儲存貯體的名稱。您還應該取代開始和結束日期。若要立即啟動叢集，`startDateTime`請設定為過去一天的日期，並設定`endDateTime`為 future 某天的日期。AWS Data Pipeline 然後立即開始啟動「過期」叢集，試圖解決它認為積壓工作的問題。這個回填表示您不需要等待一個小時，就能看到 AWS Data Pipeline 啟動其第一個叢集。

```
{
  "objects": [
    {
      "id": "Hourly",
      "type": "Schedule",
      "startDateTime": "2012-11-19T07:48:00",
      "endDateTime": "2012-11-21T07:48:00",
      "period": "1 hours"
    },
    {
      "id": "MyCluster",
      "type": "EmrCluster",
      "masterInstanceType": "m1.small",
      "schedule": {
        "ref": "Hourly"
      }
    },
    {
      "id": "MyEmrActivity",
      "type": "EmrActivity",
      "schedule": {
        "ref": "Hourly"
      },
      "runsOn": {
        "ref": "MyCluster"
      },
      "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar,-input,s3n://
elasticmapreduce/samples/wordcount/input,-output,s3://myawsbucket/wordcount/
```

```
output/#{@scheduledStartTime},-mapper,s3n://elasticmapreduce/samples/wordcount/
wordSplitter.py,-reducer,aggregate"
    }
  ]
}
```

此管道有三個物件：

- `Hourly`，代表工作排程。您可以將排程設定為活動的欄位之一。當您這麼做時，活動會根據該排程執行，或如本案例每小時執行。
- `MyCluster`，代表用於EC2執行叢集的 Amazon 執行個體集合。您可以指定要當做叢集執行的EC2執行個體大小和數目。如果您不指定執行個體的數量，則此叢集會啟動兩個，主節點和任務節點。您可以指定要在其中啟動叢集的子網路。您可以將其他組態新增至叢集，例如將其他軟體載入 Amazon EMR 提供AMI的啟動程序動作。
- `MyEmrActivity`，表示要使用叢集處理的計算。Amazon EMR 支援多種類型的叢集，包括串流、串聯式和指令碼式 Hive。該`runsOn`字段引用回 MyCluster，使用該字段作為集群基礎的規範。

## 上傳並啟動管道定義

您必須上傳管道定義並啟用管道。在下面的示例命令中，替換 *`pipeline_name`* 帶有管道的標籤 *`pipeline_file`* 具有管線定義`.json`檔案的完整路徑。

AWS CLI

若要建立管線定義並啟動管線，請使用下列建立管線指令。請記下管道的 ID，因為大多數CLI命令都會使用此值。

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

若要上傳管線定義，請使用下列put-pipeline-definition指令。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

如果管線驗證成功，則`validationErrors`欄位為空白。您應該檢閱任何警告。

若要啟動管線，請使用下列啟動管線指令。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

您可以使用下列 list-pipeline 指令，確認您的管線是否出現在管線清單中。

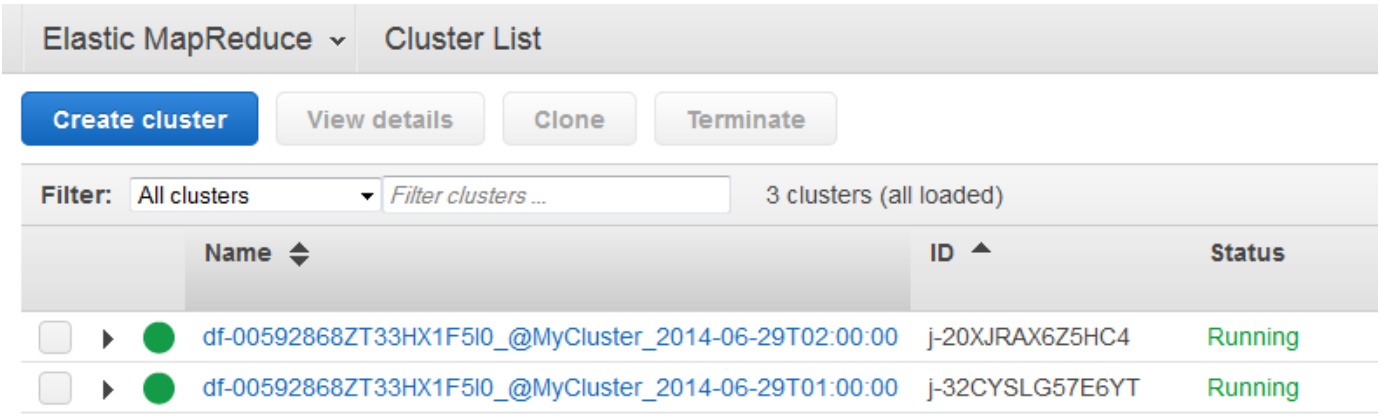
```
aws datapipeline list-pipelines
```

## 監控管道執行

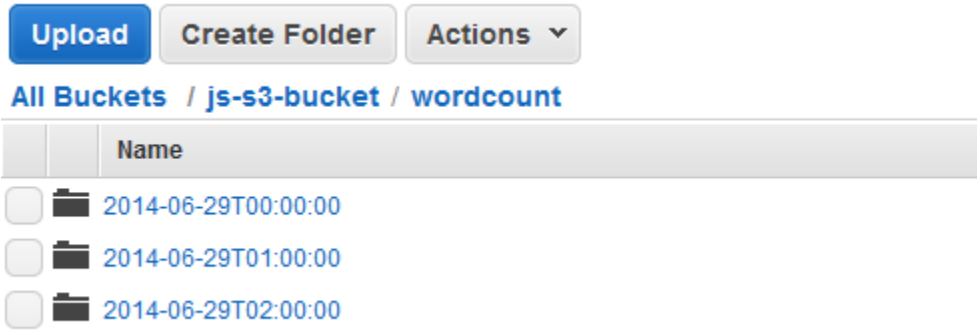
您可以 AWS Data Pipeline 使用 Amazon EMR 主控台檢視啟動的叢集，也可以使用 Amazon S3 主控台檢視輸出資料夾。

若要檢查由下列項目啟動的叢集進度 AWS Data Pipeline

1. 打開 Amazon EMR 控制台。
2. 由 AWS Data Pipeline 產生的叢集的名稱格式如下：*<pipeline-identifier>*_@*<emr-cluster-name>*_*<launch-time>*.




3. 其中一個執行完成之後，開啟 Amazon S3 主控台並檢查時間戳記輸出資料夾是否存在，並包含叢集的預期結果。



# 在亞馬遜 S3 儲存貯體之間複製 CSV 資料 AWS Data Pipeline

在您閱讀什麼是 AWS Data Pipeline？並決定要使用 AWS Data Pipeline 自動化您的資料移動和轉換之後，即可開始建立資料管道。為了協助您了解 AWS Data Pipeline 的運作方式，讓我們演練一個簡單的任務。

本教學將逐步引導您完成建立資料管道的程序，將資料從一個 Amazon S3 儲存貯體複製到另一個儲存貯體，然後在複製活動成功完成後傳送 Amazon SNS 通知。您可以使用 AWS Data Pipeline 所管理的 EC2 執行個體來處理此複製活動。

管道物件

管道會使用下列物件：

CopyActivity

為此管道AWS Data Pipeline執行的活動 (將 CSV 資料從一個 Amazon S3 儲存貯體複製到另一個儲存貯體)。

> **Important**
>
> 搭配 `CopyActivity` 和 `S3DataNode` 使用 CSV 檔案格式時有一些限制。如需詳細資訊，請參閱CopyActivity。

排程

此活動的開始日期、時間和週期。您可以選擇性地指定結束日期和時間。

Ec2Resource

AWS Data Pipeline 用來執行此活動的資源 (EC2 執行個體)。

S3 DataNode

此管道的輸入和輸出節點 (Amazon S3 儲存貯體)。

SnsAlarm

符合指定條件時AWS Data Pipeline必須採取動作 (任務成功完成後，將 Amazon SNS 通知傳送至主題)。

目錄

## 開始之前

請務必完成下列步驟。

- 完成 設定 AWS Data Pipeline 中的任務。
- (選用) 為執行個體設定 VPC，並為 VPC 設定安全群組。
- 建立 Amazon S3 儲存貯體做為資料來源。

  如需詳細資訊，請參閱 Amazon Simple Storage Service 主控台使用者指南中的建立儲存貯體。
- 將您的資料上傳到您的 Amazon S3 儲存貯體。

  如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的新增物件至儲存貯體。
- 建立另一個 Amazon S3 儲存貯體做為資料目標
- 建立用於傳送電子郵件通知的主題，並記下 Amazon Resource Name (ARN)。如需詳細資訊，請參閱 Amazon 簡單通知服務入門指南中的建立主題。
- (選用) 此教學會使用 AWS Data Pipeline 所建立的預設 IAM 角色政策。如果您想要建立和設定自己的 IAM 角色政策和信任關係，請按照中所述的指示進行操作AWS Data Pipeline 的 IAM 角色。

## 使用命令列複製 CSV 資料

您可以建立和使用管道，將資料從一個 Amazon S3 儲存貯體複製到另一個儲存貯體。

先決條件

開始之前，您必須完成下列步驟：

1. 安裝和設定命令列介面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確定已命名DataPipelineDefaultRole且DataPipelineDefaultResourceRole存在的 IAM 角色。主 AWS Data Pipeline控台會自動為您建立這些角色。如果您至少沒有使用AWS Data Pipeline主控台一次，則必須手動建立這些角色。如需詳細資訊，請參閱AWS Data Pipeline 的 IAM 角色。

任務

- 以 JSON 格式定義管道

- 上傳和啟用管道定義

## 以 JSON 格式定義管道

此範例案例顯示如何使用 JSON 管道定義和 AWS Data Pipeline CLI，以特定時間間隔在兩個 Amazon S3 儲存貯體之間排程複製資料。這是完整的管道定義 JSON 檔案，後面接著說明其每個部分。

> **Note**
>
> 建議您使用文字編輯器，協助您驗證 JSON 格式檔案的語法，並使用 .json 副檔名命名檔案。

在此範例中，為了清楚起見，我們將略過選用欄位並只顯示必要欄位。此範例的完整管道 JSON 檔案如下：

```
{
  "objects": [
    {
      "id": "MySchedule",
      "type": "Schedule",
      "startDateTime": "2013-08-18T00:00:00",
      "endDateTime": "2013-08-19T00:00:00",
      "period": "1 day"
    },
    {
      "id": "S3Input",
      "type": "S3DataNode",
      "schedule": {
        "ref": "MySchedule"
      },
      "filePath": "s3://example-bucket/source/inputfile.csv"
    },
    {
      "id": "S3Output",
      "type": "S3DataNode",
      "schedule": {
        "ref": "MySchedule"
      },
      "filePath": "s3://example-bucket/destination/outputfile.csv"
    },
    {
      "id": "MyEC2Resource",
```

```
      "type": "Ec2Resource",
      "schedule": {
        "ref": "MySchedule"
      },
      "instanceType": "m1.medium",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "MyCopyActivity",
      "type": "CopyActivity",
      "runsOn": {
        "ref": "MyEC2Resource"
      },
      "input": {
        "ref": "S3Input"
      },
      "output": {
        "ref": "S3Output"
      },
      "schedule": {
        "ref": "MySchedule"
      }
    }
  ]
}
```

排程

管道會定義含開始和結束日期的排程，以及決定此管道所執行活動頻率的期間。

```
{
  "id": "MySchedule",
  "type": "Schedule",
  "startDateTime": "2013-08-18T00:00:00",
  "endDateTime": "2013-08-19T00:00:00",
  "period": "1 day"
},
```

亞馬遜 S3 數據節點

接下來，輸入 S3 DataNode 管道元件會定義輸入檔案的位置；在此情況下，為 Amazon S3 儲存貯體位置。輸入 S3 DataNode 元件由下列欄位定義：

```
{
  "id": "S3Input",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://example-bucket/source/inputfile.csv"
},
```

Id

輸入位置的使用者定義名稱 (僅供您參考的標籤)。

類型

在 Amazon S3 儲存貯DataNode體中的管道元件類型，即與資料所在位置相符的「S3」。

排程

我們在標示為「MySchedule」的 JSON 檔案的前幾行中建立的排程元件的參考。

路徑

資料節點相關資料的路徑。資料節點的語法取決於其類型。例如，Amazon S3 路徑的語法遵循適用於資料庫表格的不同語法。

接下來，輸出 S3 DataNode 元件會定義資料的輸出目的地位置。它遵循與輸入 S3 DataNode 元件相同的格式，但元件名稱和指示目標檔案的不同路徑除外。

```
{
  "id": "S3Output",
  "type": "S3DataNode",
  "schedule": {
    "ref": "MySchedule"
  },
  "filePath": "s3://example-bucket/destination/outputfile.csv"
},
```

資源

這是執行複製操作的運算資源定義。在此範例中，AWS Data Pipeline 應該會自動建立 EC2 執行個體以執行複製任務，並在任務完成之後終止資源。此處定義的欄位會控制執行此工作之 EC2 執行個體建立和運作。EC2Resource 是由下列欄位定義：

```
{
  "id": "MyEC2Resource",
  "type": "Ec2Resource",
  "schedule": {
    "ref": "MySchedule"
  },
  "instanceType": "m1.medium",
  "role": "DataPipelineDefaultRole",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
```

Id

管道排程的使用者定義名稱，這是僅供您參考的標籤。

類型

要執行工作的運算資源類型；在本例中是 EC2 執行個體。還有其他可用的資源類型，例如類 EmrCluster型。

排程

建立此運算資源所依據的排程。

instanceType

要建立的 EC2 執行個體大小。確認您所設定的 EC2 執行個體大小最符合您要使用 AWS Data Pipeline 執行的工作負載。在本例中，我們將設定 m1.medium EC2 執行個體。有關不同執行個體類型以及何時使用每種執行個體類型的詳細資訊，請參閱 Amazon EC2 執行個體類型主題，網址為 http://aws.amazon.com/ec2/instance-types/。

角色

存取資源 (例如存取 Amazon S3 儲存貯體以擷取資料) 之帳戶的 IAM 角色。

resourceRole

建立資源的帳戶 IAM 角色，例如代您建立和設定 EC2 執行個體。角色和角色ResourceRole可以是相同的角色，但在安全性組態中分別提供更大的細微性。

活動

JSON 檔案的最後部分是代表所要執行工作的活動定義。此範例使用CopyActivity將資料從 http://aws.amazon.com/ec2/instance-types/ 儲存貯體中的 CSV 檔案複製到另一個值區。CopyActivity元件是由下列欄位定義：

```
{
  "id": "MyCopyActivity",
  "type": "CopyActivity",
  "runsOn": {
    "ref": "MyEC2Resource"
  },
  "input": {
    "ref": "S3Input"
  },
  "output": {
    "ref": "S3Output"
  },
  "schedule": {
    "ref": "MySchedule"
  }
}
```

Id

活動的使用者定義名稱，這是僅供您參考的標籤。

類型

要執行的活動類型，例如MyCopyActivity。

runsOn

執行此活動所定義工作的運算資源。在此範例中，我們參考了之前定義的 EC2 執行個體。使用runsOn 欄位讓 AWS Data Pipeline 代您建立 EC2 執行個體。runsOn 欄位表示資源存在於 AWS 基礎設施，而 workerGroup 值表示您想要使用自己的現場部署資源來執行工作。

Input

要複製的資料位置。

輸出

目標位置資料。

排程

執行此活動所依據的排程。

### 上傳和啟用管道定義

您必須上傳管道定義並啟用管道。在下列範例指令中，請將管###取代為管線和管##############的完整路徑。.json

AWS CLI

若要建立管線定義並啟動管線，請使用下列建立管線指令。請記下管線的 ID，因為大多數 CLI 命令都會使用此值。

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

若要上傳管線定義，請使用下列put-pipeline-definition指令。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

如果管線驗證成功，則validationErrors欄位為空白。您應該檢閱任何警告。

若要啟動管線，請使用下列啟動管線指令。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

您可以使用下列 list-pipeline 指令，確認您的管線是否出現在管線清單中。

```
aws datapipeline list-pipelines
```

# 使用將 MySQL 數據導出到亞馬遜 S3 AWS Data Pipeline

本教學將引導您完成建立資料管道的程序，將資料 (列) 從 MySQL 資料庫中的表複製到 Amazon S3 儲存貯體中的 CSV (逗號分隔值) 檔案，然後在複製活動成功完成後傳送 Amazon SNS 通知。您將使用 AWS Data Pipeline 所提供的 EC2 執行個體來處理此複製活動。

管道物件

管道會使用下列物件：

- CopyActivity
- Ec2Resource
- MySqlDataNode
- S3 DataNode
- SnsAlarm

目錄



# 開始之前

請務必完成下列步驟。

- 完成 設定 AWS Data Pipeline 中的任務。
- (選用) 為執行個體設定 VPC，並為 VPC 設定安全群組。
- 建立 Amazon S3 儲存貯體做為資料輸出。

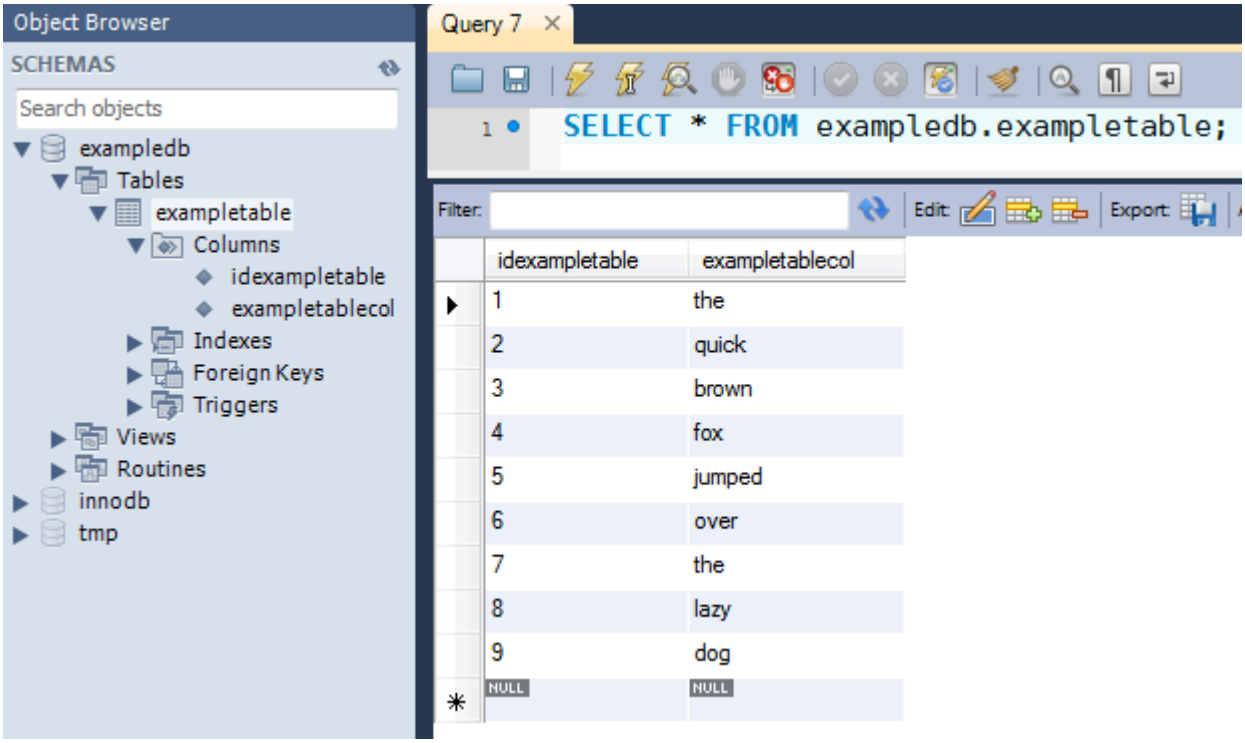
  如需詳細資訊，請參閱在 Amazon 簡單儲存服務使用者指南中建立儲存貯體。
- 建立和啟動 MySQL 資料庫執行個體做為您的資料來源。

  如需詳細資訊，請參閱 Amazon RDS 入門指南中的啟動資料庫執行個體。擁有 Amazon RDS 執行個體之後，請參閱 MySQL 文件中的建立表格。

  > **Note**
  >
  > 記下您用於建立 MySQL 執行個體的使用者名稱和密碼。在您啟動 MySQL 資料庫執行個體之後，記下執行個體的端點。稍後您將需要此資訊。

- 連接至您的 MySQL 資料庫執行個體、建立資料表，然後將測試資料值新增至新建立的資料表。

  為了方便說明，我們使用了含有下列組態和範例資料的 MySQL 資料表來建立此教學。下列螢幕擷取畫面是來自 MySQL Workbench 5.2 CE：




如需詳細資訊，請參閱 MySQL 文件中的建立資料表和 MySQL Workbench 產品頁面。

- 建立用於傳送電子郵件通知的主題，並記下 Amazon Resource Name (ARN)。如需詳細資訊，請參閱 Amazon 簡單通知服務入門指南中的建立主題。
- (選用) 此教學會使用 AWS Data Pipeline 所建立的預設 IAM 角色政策。如果您想要建立和設定 IAM 角色政策和信任關係，請按照中所述的指示進行操作AWS Data Pipeline 的 IAM 角色。

## 使用命令列複製 MySQL 資料

您可以建立管道，將資料從 MySQL 資料表複製到 Amazon S3 儲存貯體中的檔案。

先決條件

開始之前，您必須完成下列步驟：

1. 安裝和設定命令列介面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確定已命名DataPipelineDefaultRole且DataPipelineDefaultResourceRole存在的 IAM 角色。主 AWS Data Pipeline控台會自動為您建立這些角色。如果您至少沒有使用AWS Data Pipeline主控台一次，則必須手動建立這些角色。如需詳細資訊，請參閱AWS Data Pipeline 的 IAM 角色。
3. 設置一個亞馬遜 S3 存儲桶和一個亞馬遜 RDS 實例。如需詳細資訊，請參閱開始之前。

任務

- 以 JSON 格式定義管道
- 上傳和啟用管道定義

## 以 JSON 格式定義管道

此範例案例顯示如何使用 JSON 管線定義和 AWS Data Pipeline CLI，以指定的時間間隔將 MySQL 資料庫中資料表中的資料 (列) 複製到 Amazon S3 儲存貯體中的 CSV (逗號分隔值) 檔案。

這是完整的管道定義 JSON 檔案，後面接著說明其每個部分。

> **Note**
>
> 建議您使用文字編輯器，協助您驗證 JSON 格式檔案的語法，並使用 .json 副檔名命名檔案。

```
{
  "objects": [
    {
      "id": "ScheduleId113",
      "startDateTime": "2013-08-26T00:00:00",
      "name": "My Copy Schedule",
      "type": "Schedule",
      "period": "1 Days"
    },
    {
      "id": "CopyActivityId112",
      "input": {
        "ref": "MySqlDataNodeId115"
      },
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My Copy",
      "runsOn": {
        "ref": "Ec2ResourceId116"
      },
      "onSuccess": {
        "ref": "ActionId1"
      },
      "onFail": {
```

```
        "ref": "SnsAlarmId117"
      },
      "output": {
        "ref": "S3DataNodeId114"
      },
      "type": "CopyActivity"
    },
    {
      "id": "S3DataNodeId114",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "filePath": "s3://example-bucket/rds-output/output.csv",
      "name": "My S3 Data",
      "type": "S3DataNode"
    },
    {
      "id": "MySqlDataNodeId115",
      "username": "my-username",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My RDS Data",
      "*password": "my-password",
      "table": "table-name",
      "connectionString": "jdbc:mysql://your-sql-instance-name.id.region-
name.rds.amazonaws.com:3306/database-name",
      "selectQuery": "select * from #{table}",
      "type": "SqlDataNode"
    },
    {
      "id": "Ec2ResourceId116",
      "schedule": {
        "ref": "ScheduleId113"
      },
      "name": "My EC2 Resource",
      "role": "DataPipelineDefaultRole",
      "type": "Ec2Resource",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "message": "This is a success message.",
      "id": "ActionId1",
      "subject": "RDS to S3 copy succeeded!",
```

```
      "name": "My Success Alarm",
      "role": "DataPipelineDefaultRole",
      "topicArn": "arn:aws:sns:us-east-1:123456789012:example-topic",
      "type": "SnsAlarm"
    },
    {
      "id": "Default",
      "scheduleType": "timeseries",
      "failureAndRerunMode": "CASCADE",
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "message": "There was a problem executing #{node.name} at for period
 #{node.@scheduledStartTime} to #{node.@scheduledEndTime}",
      "id": "SnsAlarmId117",
      "subject": "RDS to S3 copy failed",
      "name": "My Failure Alarm",
      "role": "DataPipelineDefaultRole",
      "topicArn": "arn:aws:sns:us-east-1:123456789012:example-topic",
      "type": "SnsAlarm"
    }
  ]
}
```

MySQL 資料節點

輸入MySqlDataNode管道元件定義輸入資料的位置；在此情況下為 Amazon RDS 執行個體。輸入MySqlDataNode組件由下列欄位定義：

```
{
  "id": "MySqlDataNodeId115",
  "username": "my-username",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My RDS Data",
  "*password": "my-password",
  "table": "table-name",
  "connectionString": "jdbc:mysql://your-sql-instance-name.id.region-
name.rds.amazonaws.com:3306/database-name",
  "selectQuery": "select * from #{table}",
```

```
  "type": "SqlDataNode"
},
```

Id

使用者定義的名稱，這是僅供您參考的標籤。

使用者名稱

資料庫帳戶的使用者名稱，該帳戶具備足以從資料庫資料表擷取資料的許可。用您的用戶名替換#的用戶名。

排程

我們在上述 JSON 檔案的程式碼行中已建立的排程元件參考。

名稱

使用者定義的名稱，這是僅供您參考的標籤。

*Password

資料庫帳戶的密碼，具有星號前綴指出 AWS Data Pipeline 必須加密密碼值。用正確######的密碼為您的用戶。密碼欄位前面會加上星號特殊字元。如需詳細資訊，請參閱特殊字元。

資料表

包含所要複製資料的資料庫資料表名稱。請以您的資料庫資料表名稱取代 *table-name*。

connectionString

要連線至資料庫之CopyActivity物件的 JDBC 連接字串。

selectQuery

有效的 SQL SELECT 查詢，指定要從資料庫資料表複製的資料。請注意，#{table} 是一種表達式，會重複使用上述 JSON 檔案程式碼行中 "table" 變數所提供的資料表名稱。

類型

SqlDataNode類型，在此範例中是使用 MySQL 的亞馬遜 RDS 執行個體。

> **Note**
>
> MySqlDataNode 類型已移除。雖然您仍然可以使用MySqlDataNode，但我們建議您使用SqlDataNode.

亞馬遜 S3 數據節點

接下來，S3Output 管道元件會定義輸出檔案的位置；在這種情況下，是位於 Amazon S3 儲存貯體位置的 CSV 檔案。輸出 S3 DataNode 元件由下列欄位定義：

```
{
  "id": "S3DataNodeId114",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "filePath": "s3://example-bucket/rds-output/output.csv",
  "name": "My S3 Data",
  "type": "S3DataNode"
},
```

Id

使用者定義的 ID，這是僅供您參考的標籤。

排程

我們在上述 JSON 檔案的程式碼行中已建立的排程元件參考。

filePath

資料節點的相關資料路徑，在此範例中是 CSV 輸出檔案。

名稱

使用者定義的名稱，這是僅供您參考的標籤。

類型

在 Amazon S3 儲存貯體DataNode中的管道物件類型，即 S3 與資料所在位置相符。

資源

這是執行複製操作的運算資源定義。在此範例中，AWS Data Pipeline 應該會自動建立 EC2 執行個體以執行複製任務，並在任務完成之後終止資源。此處定義的欄位會控制執行此工作之 EC2 執行個體建立和運作。EC2Resource 是由下列欄位定義：

```
{
  "id": "Ec2ResourceId116",
```

```
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My EC2 Resource",
  "role": "DataPipelineDefaultRole",
  "type": "Ec2Resource",
  "resourceRole": "DataPipelineDefaultResourceRole"
},
```

Id

使用者定義的 ID，這是僅供您參考的標籤。

排程

建立此運算資源所依據的排程。

名稱

使用者定義的名稱，這是僅供您參考的標籤。

角色

存取資源 (例如存取 Amazon S3 儲存貯體以擷取資料) 之帳戶的 IAM 角色。

類型

要執行工作的運算資源類型；在本例中是 EC2 執行個體。還有其他可用的資源類型，例如類EmrCluster型。

resourceRole

建立資源的帳戶 IAM 角色，例如代您建立和設定 EC2 執行個體。角色和角色ResourceRole可以是相同的角色，但在安全性組態中分別提供更大的細微性。

活動

JSON 檔案的最後部分是代表所要執行工作的活動定義。在這種情況下，我們使用CopyActivity元件將資料從 Amazon S3 儲存貯體中的檔案複製到另一個檔案。CopyActivity 元件是由下列欄位定義：

```
{
  "id": "CopyActivityId112",
  "input": {
    "ref": "MySqlDataNodeId115"
```

```
  },
  "schedule": {
    "ref": "ScheduleId113"
  },
  "name": "My Copy",
  "runsOn": {
    "ref": "Ec2ResourceId116"
  },
  "onSuccess": {
    "ref": "ActionId1"
  },
  "onFail": {
    "ref": "SnsAlarmId117"
  },
  "output": {
    "ref": "S3DataNodeId114"
  },
  "type": "CopyActivity"
},
```

Id

使用者定義的 ID，這是僅供您參考的標籤

Input

要複製的 MySQL 資料位置

排程

執行此活動所依據的排程

名稱

使用者定義的名稱，這是僅供您參考的標籤

runsOn

執行此活動所定義工作的運算資源。在此範例中，我們參考了之前定義的 EC2 執行個體。使用 `runsOn` 欄位讓 AWS Data Pipeline 代您建立 EC2 執行個體。`runsOn` 欄位表示資源存在於 AWS 基礎設施，而 workerGroup 值表示您想要使用自己的現場部署資源來執行工作。

onSuccess

活動成功完成時所要傳送的 SnsAlarm

onFail

活動失敗時所要傳送的 SnsAlarm

輸出

CSV 輸出檔案的亞馬遜 S3 位置

類型

要執行的活動類型。

## 上傳和啟用管道定義

您必須上傳管道定義並啟用管道。在下列範例指令中，請將管###取代為管線和管###############的完整路徑。.json

AWS CLI

若要建立管線定義並啟動管線，請使用下列建立管線指令。請記下管線的 ID，因為大多數 CLI 命令都會使用此值。

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

若要上傳管線定義，請使用下列put-pipeline-definition指令。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

如果管線驗證成功，則validationErrors欄位為空白。您應該檢閱任何警告。

若要啟動管線，請使用下列啟動管線指令。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

您可以使用下列 list-pipeline 指令，確認您的管線是否出現在管線清單中。

```
aws datapipeline list-pipelines
```

# 使用將數據複製到亞馬遜紅移 AWS Data Pipeline

本教學將逐步引導您完成建立管道的程序，以便使用AWS Data Pipeline主控台中的「複製到Redshift」範本，或使用 CLI 建立管道定期將資料從 Amazon S3 移到 Amazon Redshift。AWS Data Pipeline

Amazon S3 是一種可讓您將資料存放在雲端的網路服務。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南。

亞馬遜紅移是雲中的一種數據倉庫服務。如需詳細資訊，請參閱亞馬遜紅移管理指南。

本教學有數項事前準備。完成下列步驟後，您可以使用主控台或 CLI 繼續教學。

目錄



## 在您開始之前：設定 COPY 選項並載入資料

在將資料複製到其中的亞馬遜紅移之前AWS Data Pipeline，請確保您：

- 從亞馬遜 S3 載入資料。
- 在亞馬遜紅移中設置COPY活動。

一旦讓這些選項開始運作並成功完成資料載入，請將這些選項傳輸至 AWS Data Pipeline，在其中執行複製。

如需COPY選項，請參閱 Amazon Redshift 資料庫開發人員指南中的複製。

如需從 Amazon S3 載入資料的步驟，請參閱亞馬遜 Redshift 資料庫開發人員指南中的從 Amazon S3 載入資料。

例如，Amazon Redshift 中的下列 SQL 命令會建立名為的新表格，LISTING並從 Amazon S3 中公開可用的儲存貯體複製範例資料。

以您自己的值取代 <iam-role-arn> 和區域。

如需有關此範例的詳細資訊，請參閱 Amazon Redshift 入門指南中的從 Amazon S3 載入範例資料。

```
create table listing(
 listid integer not null distkey,
 sellerid integer not null,
 eventid integer not null,
 dateid smallint not null  sortkey,
 numtickets smallint not null,
 priceperticket decimal(8,2),
 totalprice decimal(8,2),
 listtime timestamp);

copy listing from 's3://awssampledbuswest2/tickit/listings_pipe.txt'
credentials 'aws_iam_role=<iam-role-arn>'
delimiter '|' region 'us-west-2';
```

## 設定管道、建立安全群組，以及建立 Amazon Redshift 叢集

針對教學設定

1. 完成 設定 AWS Data Pipeline 中的任務。
2. 建立安全群組。
   a. 開啟 Amazon EC2 主控台。
   b. 在導覽窗格中，按一下 Security Groups (安全群組)。
   c. 按一下 Create Security Group (建立安全群組)。
   d. 指定安全群組的名稱和描述。
   e. [EC2-典型] 選擇**No VPC**適用於 VPC。
   f. [EC2-VPC] 針對 VPC 選取您 VPC 的 ID。
   g. 按一下 Create (建立)。
3. [EC2-典型] 建立一個亞馬遜紅移叢集安全群組，並指定亞馬遜 EC2 安全群組。
   a. 打開亞馬遜紅移控制台。
   b. 在導覽窗格中，按一下 Security Groups (安全群組)。
   c. 按一下 Create Cluster Security Group (建立叢集安全群組)。
   d. 在 Create Cluster Security Group (建立叢集安全群組) 對話方塊中，指定叢集安全群組的名稱和描述。
   e. 按一下新叢集安全群組的名稱。
   f. 按一下 Add Connection Type (新增連線類型)。

g. 在 [新增連線類型] 對話方塊中，從 [連線類型] 選取 [EC2 安全群組]，選取您從 EC2 安全群組名稱建立的安全群組，然後按一下 [授權]。

4. [EC2-VPC] 建立一個亞馬遜紅移群集安全組並指定 VPC 安全組。

   a. 開啟 Amazon EC2 主控台。

   b. 在導覽窗格中，按一下 Security Groups (安全群組)。

   c. 按一下 Create Security Group (建立安全群組)。

   d. 在 Create Security Group (建立安全群組) 對話方塊中，指定安全群組的名稱和描述，然後針對 VPC 選取 VPC 的 ID。

   e. 按一下 Add Rule (新增規則)。指定類型、協定和連接埠範圍，然後在 Source (來源) 中開始輸入安全群組的 ID。選取您在第二個步驟中建立的安全群組。

   f. 按一下 Create (建立)。

5. 下列是這些步驟的摘要。

   如果您有現有的 Amazon Redshift 叢集，請記下叢集 ID。

   若要建立新叢集並載入範例資料，請按照開始使用 Amazon Redshift 中的步驟進行操作。如需有關建立叢集的詳細資訊，請參閱 Amazon Redshift 管理指南中的建立叢集。

   a. 打開亞馬遜紅移控制台。

   b. 按一下 Launch Cluster (啟動叢集)。

   c. 為您的叢集提供所需的詳細資訊，然後按一下 Continue (繼續)。

   d. 提供節點組態，然後按一下 Continue (繼續)。

   e. 在額外組態資訊頁面中，選取您建立的叢集安全群組，然後按一下 Continue (繼續)。

   f. 檢閱您叢集的規格，然後按一下 Launch Cluster (啟動叢集)。

## 使用命令列將資料複製到亞馬遜紅移

本教學將示範如何將資料從 Amazon S3 複製到亞馬遜紅移。您將在 Amazon Redshift 中建立新的資料表，然後使用AWS Data Pipeline從公有 Amazon S3 儲存貯體 (其中包含 CSV 格式的範例輸入資料) 將資料傳輸到此資料表。日誌會儲存到您擁有的 Amazon S3 儲存貯體。

Amazon S3 是一種可讓您將資料存放在雲端的網路服務。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南。亞馬遜紅移是雲中的一種數據倉庫服務。如需詳細資訊，請參閱亞馬遜紅移管理指南。

先決條件

開始之前，您必須完成下列步驟：

1. 安裝和設定命令列介面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確定已命名DataPipelineDefaultRole且DataPipelineDefaultResourceRole存在的 IAM 角色。主AWS Data Pipeline控台會自動為您建立這些角色。如果您至少沒有使用AWS Data Pipeline主控台一次，則必須手動建立這些角色。如需詳細資訊，請參閱AWS Data Pipeline 的 IAM 角色。
3. 在 Amazon Redshift 中設定COPY命令，因為當您在中執行複製作業時，需要使用這些相同的選項。AWS Data Pipeline如需相關資訊，請參閱 在您開始之前：設定 COPY 選項並載入資料。
4. 設置一個亞馬遜紅移數據庫。如需詳細資訊，請參閱設定管道、建立安全群組，以及建立 Amazon Redshift 叢集。

任務

- 以 JSON 格式定義管道
- 上傳和啟用管道定義

## 以 JSON 格式定義管道

此範例案例顯示如何將資料從 Amazon S3 儲存貯體複製到亞馬遜紅移。

這是完整的管道定義 JSON 檔案，後面接著說明其每個部分。建議您使用文字編輯器，協助您驗證 JSON 格式檔案的語法，並使用 `.json` 副檔名命名檔案。

```
{
  "objects": [
    {
      "id": "CSVId1",
      "name": "DefaultCSV1",
      "type": "CSV"
    },
    {
      "id": "RedshiftDatabaseId1",
      "databaseName": "dbname",
      "username": "user",
      "name": "DefaultRedshiftDatabase1",
      "*password": "password",
      "type": "RedshiftDatabase",
      "clusterId": "redshiftclusterId"
```

```
    },
    {
      "id": "Default",
      "scheduleType": "timeseries",
      "failureAndRerunMode": "CASCADE",
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "RedshiftDataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "tableName": "orders",
      "name": "DefaultRedshiftDataNode1",
      "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30)
PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
varchar(20));",
      "type": "RedshiftDataNode",
      "database": {
        "ref": "RedshiftDatabaseId1"
      }
    },
    {
      "id": "Ec2ResourceId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "securityGroups": "MySecurityGroup",
      "name": "DefaultEc2Resource1",
      "role": "DataPipelineDefaultRole",
      "logUri": "s3://myLogs",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "type": "Ec2Resource"
    },
    {
      "id": "ScheduleId1",
      "startDateTime": "yyyy-mm-ddT00:00:00",
      "name": "DefaultSchedule1",
      "type": "Schedule",
      "period": "period",
      "endDateTime": "yyyy-mm-ddT00:00:00"
    },
```

```
    {
      "id": "S3DataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
      "name": "DefaultS3DataNode1",
      "dataFormat": {
        "ref": "CSVId1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "RedshiftCopyActivityId1",
      "input": {
        "ref": "S3DataNodeId1"
      },
      "schedule": {
        "ref": "ScheduleId1"
      },
      "insertMode": "KEEP_EXISTING",
      "name": "DefaultRedshiftCopyActivity1",
      "runsOn": {
        "ref": "Ec2ResourceId1"
      },
      "type": "RedshiftCopyActivity",
      "output": {
        "ref": "RedshiftDataNodeId1"
      }
    }
  ]
}
```

如需這些物件的詳細資訊，請參閱下列文件。

物件

- 資料節點
- 資源
- 活動

資料節點

此範例使用輸入資料節點、輸出資料節點和資料庫。

輸入資料節點

輸入S3DataNode管道元件會定義 Amazon S3 中輸入資料的位置以及輸入資料的資料格式。如需詳細資訊，請參閱S3 DataNode。

此輸入元件是由下列欄位定義：

```
{
  "id": "S3DataNodeId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
  "name": "DefaultS3DataNode1",
  "dataFormat": {
    "ref": "CSVId1"
  },
  "type": "S3DataNode"
},
```

`id`

使用者定義的 ID，這是僅供您參考的標籤。

`schedule`

對排程元件的參考。

`filePath`

與資料節點相關聯資料的路徑，在此範例中是 CSV 輸入檔。

`name`

使用者定義的名稱，這是僅供您參考的標籤。

`dataFormat`

對要處理之活動資料格式的參考。

輸出資料節點

輸出RedshiftDataNode管線元件會定義輸出資料的位置；在此情況下，是 Amazon Redshift 資料庫中的資料表。如需詳細資訊，請參閱RedshiftDataNode。此輸出元件是由下列欄位定義：

```
{
  "id": "RedshiftDataNodeId1",
  "schedule": {
    "ref": "ScheduleId1"
  },
  "tableName": "orders",
  "name": "DefaultRedshiftDataNode1",
  "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30) PRIMARY
 KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
 varchar(20));",
  "type": "RedshiftDataNode",
  "database": {
    "ref": "RedshiftDatabaseId1"
  }
},
```

`id`

使用者定義的 ID，這是僅供您參考的標籤。

`schedule`

對排程元件的參考。

`tableName`

Amazon Redshift 資料表的名稱。

`name`

使用者定義的名稱，這是僅供您參考的標籤。

`createTableSql`

在資料庫建立資料表的 SQL 表達式。

`database`

對亞馬遜紅移數據庫的引用。

資料庫

RedshiftDatabase 元件是由下列欄位定義。如需詳細資訊，請參閱RedshiftDatabase。

```
{
  "id": "RedshiftDatabaseId1",
  "databaseName": "dbname",
  "username": "user",
  "name": "DefaultRedshiftDatabase1",
  "*password": "password",
  "type": "RedshiftDatabase",
  "clusterId": "redshiftclusterId"
},
```

`id`

使用者定義的 ID，這是僅供您參考的標籤。

`databaseName`

邏輯資料庫的名稱。

`username`

連線至資料庫的使用者名稱。

`name`

使用者定義的名稱，這是僅供您參考的標籤。

`password`

連線至資料庫的密碼。

`clusterId`

Redshift 叢集的 ID。

資源

這是執行複製操作的運算資源定義。在這個範例中，AWS Data Pipeline 應該自動建立 EC2 執行個體執行複製任務，並在任務完成後終止執行個體。此處定義的欄位會控制完成此工作之執行個體的建立和功能。如需詳細資訊，請參閱Ec2Resource。

`Ec2Resource` 是由下列欄位定義：

```
{
  "id": "Ec2ResourceId1",
  "schedule": {
```

```
    "ref": "ScheduleId1"
  },
  "securityGroups": "MySecurityGroup",
  "name": "DefaultEc2Resource1",
  "role": "DataPipelineDefaultRole",
  "logUri": "s3://myLogs",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "type": "Ec2Resource"
},
```

`id`

使用者定義的 ID，這是僅供您參考的標籤。

`schedule`

建立此運算資源所依據的排程。

`securityGroups`

在資源集區中供執行個體使用的安全群組。

`name`

使用者定義的名稱，這是僅供您參考的標籤。

`role`

存取資源 (例如存取 Amazon S3 儲存貯體以擷取資料) 之帳戶的 IAM 角色。

`logUri`

用於備份任務執行器日誌的 Amazon S3 目的地路徑`Ec2Resource`。

`resourceRole`

建立資源的帳戶 IAM 角色，例如代您建立和設定 EC2 執行個體。角色和角色ResourceRole可以是相同的角色，但在安全性組態中分別提供更大的細微性。

### 活動

JSON 檔案的最後部分是代表所要執行工作的活動定義。在這種情況下，我們會使用`RedshiftCopyActivity`元件將資料從 Amazon S3 複製到亞馬遜紅移。如需詳細資訊，請參閱RedshiftCopyActivity。

`RedshiftCopyActivity` 元件是由下列欄位定義：

```
{
  "id": "RedshiftCopyActivityId1",
  "input": {
    "ref": "S3DataNodeId1"
  },
  "schedule": {
    "ref": "ScheduleId1"
  },
  "insertMode": "KEEP_EXISTING",
  "name": "DefaultRedshiftCopyActivity1",
  "runsOn": {
    "ref": "Ec2ResourceId1"
  },
  "type": "RedshiftCopyActivity",
  "output": {
    "ref": "RedshiftDataNodeId1"
  }
},
```

`id`

使用者定義的 ID，這是僅供您參考的標籤。

`input`

亞馬遜 S3 來源檔案的參考。

`schedule`

執行此活動所依據的排程。

`insertMode`

插入類型 (KEEP_EXISTING、OVERWRITE_EXISTING 或 TRUNCATE)。

`name`

使用者定義的名稱，這是僅供您參考的標籤。

`runsOn`

執行此活動所定義工作的運算資源。

`output`

對亞馬遜紅移目標表的引用。

## 上傳和啟用管道定義

您必須上傳管道定義並啟用管道。在下列範例指令中，請將管###取代為管線和管###############的完整路徑。.json

AWS CLI

若要建立管線定義並啟動管線，請使用下列建立管線指令。請記下管線的 ID，因為大多數 CLI 命令都會使用此值。

```
aws datapipeline create-pipeline --name pipeline_name --unique-id token
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

若要上傳管線定義，請使用下列put-pipeline-definition指令。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://MyEmrPipelineDefinition.json
```

如果管線驗證成功，則validationErrors欄位為空白。您應該檢閱任何警告。

若要啟動管線，請使用下列啟動管線指令。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

您可以使用下列 list-pipeline 指令，確認您的管線是否出現在管線清單中。

```
aws datapipeline list-pipelines
```

# 管道表達式和函數

本節說明在管道中使用表達式和函數的語法，包括相關資料類型。

## 簡單資料類型

您可以將以下類型的資料設為欄位值。

類型

- DateTime
- 數值
- 物件參考
- 期間
- 字串

### DateTime

AWS Data Pipeline 僅支援以 "YYYY-MM-DDTHH:MM:SS" 格式，使用 UTC/GMT 形式呈現的日期和時間。下列範例會將 Schedule 物件的 `startDateTime` 欄位設為 UTC/GMT 時區的 `1/15/2012, 11:59 p.m.`。

```
"startDateTime" : "2012-01-15T23:59:00"
```

### 數值

AWS Data Pipeline 支援整數和浮點值。

### 物件參考

管道定義中的物件。這可以是目前物件、在管道的其他位置定義的物件名稱，或在欄位中列出目前物件的物件，並以 node 關鍵字參考。如需有關 node 的詳細資訊，請參閱 參考欄位和物件。如需管道物件類型的詳細資訊，請參閱管道物件參考。

### 期間

表示排程事件的執行頻率。這會以 "N [`years`|`months`|`weeks`|`days`|`hours`|`minutes`]" 格式表示，其中 N 是正整數值。

最短期間為 15 分鐘，而最長期間為 3 年。

下列範例會將 Schedule 物件的 period 欄位設為 3 小時。這會建立每隔三小時執行一次的排程。

```
"period" : "3 hours"
```

## 字串

標準字串值。字串必須以雙引號 (") 括住。您可以使用反斜線字元 (\) 來逸出字串中的字元。不支援多行字串。

下列範例示範 id 欄位的有效字串值。

```
"id" : "My Data Object"

"id" : "My \"Data\" Object"
```

字串也可以包含評估為字串值的表達式。這些表達式會插入字串，並以 "#{" 和 "}" 分隔。下列範例使用表達式來將目前物件的名稱插入路徑。

```
"filePath" : "s3://myBucket/#{name}.csv"
```

如需使用表達式的詳細資訊，請參閱參考欄位和物件和表達式評估。

# 表達式

表達式可讓您在相關物件之間共享一個值。AWS Data Pipeline Web 服務會在執行時間處理表達式，以確保所有表達式都會以表達式的值替代。

表達式是以 "#{" 和 "}" 分隔。您可以在字串合法的任何管道定義物件中使用表達式。如果位置參考了其中一個類型 ID (NAME、TYPE、SPHERE)，則不會評估其值並依原狀使用。

下列表達式會呼叫其中一個 AWS Data Pipeline 函數。如需詳細資訊，請參閱 表達式評估。

```
#{format(myDateTime,'YYYY-MM-dd hh:mm:ss')}
```

## 參考欄位和物件

表達式可以使用存在表達式的目前物件欄位，或參考所連結的另一個物件欄位。

位置格式包含建立時間，後面接著物件建立時間，例如 @S3BackupLocation_2018-01-31T11:05:33。

您也可以參考管道定義中指定的確切位置 ID，例如 Amazon S3 備份位置的位置 ID。若要參考位置 ID，請使用 #{parent.@id}。

在下列範例中，filePath 欄位參考了相同物件中的 id 欄位，以形成檔案名稱。filePath 得出的值為 "s3://mybucket/ExampleDataNode.csv"。

```
{
  "id" : "ExampleDataNode",
  "type" : "S3DataNode",
  "schedule" : {"ref" : "ExampleSchedule"},
  "filePath" : "s3://mybucket/#{parent.@id}.csv",
  "precondition" : {"ref" : "ExampleCondition"},
  "onFail" : {"ref" : "FailureNotify"}
}
```

若要使用存在於參考所連結另一個物件上的欄位，請使用 node 關鍵字。此關鍵字只適用於警示和先決條件物件。

繼續進行上一個範例，SnsAlarm 中的表達式可以參考 Schedule 中的日期和時間範圍，因為 S3DataNode 會參考兩者。

特別是 FailureNotify 的 message 欄位可以使用 ExampleSchedule 的 @scheduledStartTime 和 @scheduledEndTime 執行時間欄位，因為 ExampleDataNode 的 onFail 欄位參考 FailureNotify 且其 schedule 欄位參考 ExampleSchedule。

```
{
    "id" : "FailureNotify",
    "type" : "SnsAlarm",
    "subject" : "Failed to run pipeline component",
    "message": "Error for interval
 #{node.@scheduledStartTime}..#{node.@scheduledEndTime}.",
    "topicArn":"arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic"
},
```

> **Note**
>
> 您可以建立包含相依性的管道，例如您管道中相依於其他系統或任務工作的任務。如果您的管道需要特定資源，請使用資料節點和任務的相關先決條件，將這些相依性新增至管道。這可讓

您的管道更輕鬆地進行除錯且彈性更高。此外，請盡可能將您的相依性保留在單一管道內，因為跨管道故障診斷並不容易。

## 巢狀表達式

AWS Data Pipeline 可讓您巢狀值來建立更複雜的表達式。例如，若要執行時間計算 (從 scheduledStartTime 減去 30 分鐘)，並格式化結果以用於管道定義，您可以在活動中使用下列表達式：

```
#{format(minusMinutes(@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

如果表達式為 SnsAlarm 或 Precondition 的一部分，也請使用 node 前綴：

```
#{format(minusMinutes(node.@scheduledStartTime,30),'YYYY-MM-dd hh:mm:ss')}
```

## 清單

您可以評估清單上的表達式和清單上的函數。例如，假設清單定義如下："myList": ["one","two"]。 如果此清單用於表達式#{'this is ' + myList}，它將評估為["this is one", "this is two"]。如果您有兩個清單，Data Pipeline 最終會在其評估中將其壓平合併。例如，如果 myList1 定義為 [1,2]，而 myList2 定義為 [3,4]，則表達式 [#{myList1}, #{myList2}] 會評估為 [1,2,3,4]。

## 節點表達式

AWS Data Pipeline 在 SnsAlarm 或 PreCondition 中使用 #{node.*} 表達式，以反向參考管道元件的父物件。由於 SnsAlarm 和 PreCondition 是由不具反向參考的活動或資源參考，因此 node 提供方法來參考此參考者。例如，下列管道定義示範故障通知如何使用 node 來參考其父系 (在本例中是 ShellCommandActivity)，並在 SnsAlarm 訊息中包含父系的排程開始和結束時間。ShellCommandActivity 的 scheduledStartTime 參考不需要 node 前綴，因為 scheduledStartTime 會自我參考。

> **Note**
>
> 前面加上 @ 符號的欄位表示這些欄位是執行時間欄位。

```
{
  "id" : "ShellOut",
  "type" : "ShellCommandActivity",
  "input" : {"ref" : "HourlyData"},
  "command" : "/home/userName/xxx.sh #{@scheduledStartTime} #{@scheduledEndTime}",
  "schedule" : {"ref" : "HourlyPeriod"},
  "stderr" : "/tmp/stderr:#{@scheduledStartTime}",
  "stdout" : "/tmp/stdout:#{@scheduledStartTime}",
  "onFail" : {"ref" : "FailureNotify"},
},
{
  "id" : "FailureNotify",
  "type" : "SnsAlarm",
  "subject" : "Failed to run pipeline component",
  "message": "Error for interval
 #{node.@scheduledStartTime}..#{node.@scheduledEndTime}.",
  "topicArn":"arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic"
},
```

AWS Data Pipeline 支援使用者定義欄位 (而非執行時間欄位) 的轉移參考。轉移參考是兩個管道元件之間的參考，需要另一個管道元件做為媒介。下列範例顯示轉移使用者定義欄位的參考和非轉移執行時間欄位的參考，這兩者皆有效。如需詳細資訊，請參閱 使用者定義。

```
{
  "name": "DefaultActivity1",
  "type": "CopyActivity",
  "schedule": {"ref": "Once"},
  "input": {"ref": "s3nodeOne"},
  "onSuccess": {"ref": "action"},
  "workerGroup": "test",
  "output": {"ref": "s3nodeTwo"}
},
{
  "name": "action",
  "type": "SnsAlarm",
  "message": "S3 bucket '#{node.output.directoryPath}' succeeded at
 #{node.@actualEndTime}.",
  "subject": "Testing",
  "topicArn": "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic",
  "role": "DataPipelineDefaultRole"
}
```

## 表達式評估

AWS Data Pipeline 提供一組函數，可供您用來計算欄位的值。下列範例使用 makeDate 函數，將 Schedule 物件的 startDateTime 欄位設為 "2011-05-24T0:00:00" GMT/UTC。

```
"startDateTime" : "makeDate(2011,5,24)"
```

# 數學函數

下列函數可用於處理數值。

| 函數 | 描述 |
| --- | --- |
| + | 加法。<br>範例：#{1 + 2}<br>結果：3 |
| - | 減法。<br>範例：#{1 - 2}<br>結果：-1 |
| * | 乘法。<br>範例：#{1 * 2}<br>結果：2 |
| / | 除法。如果您將兩個整數相除，結果會捨去小數部分。<br>範例：#{1 / 2}，結果：0<br>範例：#{1.0 / 2}，結果：.5 |
| ^ | 指數。<br>範例：#{2 ^ 2} |

| 函數 | 描述 |
| --- | --- |
| | 結果：4.0 |

## 字串函數

下列函數可用於處理字串值。

| 函數 | 描述 |
| --- | --- |
| + | 串連。非字串值會先轉換成字串。<br>範例：#{"hel" + "lo"}<br>結果："hello" |

## 日期和時間函數

下列函數可用於處理 DateTime 值。例如，myDateTime 的值為 May 24, 2011 @ 5:10 pm GMT。

> **Note**
>
> AWS Data Pipeline 的日期/時間格式為 Joda Time，這會取代 Java 日期和時間類別。如需詳細資訊，請參閱 Joda Time - Class DateTimeFormat。

| 函數 | 描述 |
| --- | --- |
| int day(DateTime myDateTime) | 取得 DateTime 值的日 (以整數表示)。<br>範例：#{day(myDateTime)}<br>結果：24 |

| 函數 | 描述 |
| --- | --- |
| `int dayOfYear(DateTime myDateTime)` | 取得 DateTime 值的年度日 (以整數表示)。<br><br>範例：`#{dayOfYear(myDateTime)}`<br><br>結果：`144` |
| `DateTime firstOfMonth(DateTime myDateTime)` | 在指定的 DateTime 中建立月份起始的 DateTime 物件。<br><br>範例：`#{firstOfMonth(myDateTime)}`<br><br>結果：`"2011-05-01T17:10:00z"` |
| `String format(DateTime myDateTime,String format)` | 建立字串物件，這是使用指定格式字串轉換指定 DateTime 的結果。<br><br>範例：`#{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')}`<br><br>結果：`"2011-05-24T17:10:00 UTC"` |
| `int hour(DateTime myDateTime)` | 取得 DateTime 值的小時 (以整數表示)。<br><br>範例：`#{hour(myDateTime)}`<br><br>結果：`17` |

| 函數 | 描述 |
| --- | --- |
| `DateTime makeDate(int year,int month,int day)` | 使用指定的年、月和日，建立自午夜起採用 UTC 的 DateTime 物件。<br><br>範例：`#{makeDate(2011,5,24)}`<br><br>結果：`"2011-05-24T0:00:00z"` |
| `DateTime makeDateTime(int year,int month,int day,int hour,int minute)` | 使用指定的年、月、日、小時和分鐘，建立採用 UTC 的 DateTime 物件。<br><br>範例：`#{makeDateTime(2011,5,24,14,21)}`<br><br>結果：`"2011-05-24T14:21:00z"` |
| `DateTime midnight(DateTime myDateTime)` | 建立相對於指定 DateTime，目前午夜的 DateTime 物件。例如，MyDateTime 為 2011-05-25T17:10:00z ，結果如下：<br><br>範例：`#{midnight(myDateTime)}`<br><br>結果：`"2011-05-25T0:00:00z"` |

| 函數 | 描述 |
| --- | --- |
| `DateTime minusDays(DateTime myDateTime,int daysToSub)` | 建立 DateTime 物件，這是從指定 DateTime 減去指定天數的結果。<br><br>範例：`#{minusDays(myDateTime,1)}`<br><br>結果：`"2011-05-23T17:10:00z"` |
| `DateTime minusHours(DateTime myDateTime,int hoursToSub)` | 建立 DateTime 物件，這是從指定 DateTime 減去指定時數的結果。<br><br>範例：`#{minusHours(myDateTime,1)}`<br><br>結果：`"2011-05-24T16:10:00z"` |
| `DateTime minusMinutes(DateTime myDateTime,int minutesToSub)` | 建立 DateTime 物件，這是從指定 DateTime 減去指定分鐘數的結果。<br><br>範例：`#{minusMinutes(myDateTime,1)}`<br><br>結果：`"2011-05-24T17:09:00z"` |

| 函數 | 描述 |
| --- | --- |
| DateTime minusMonths(DateTime myDateTime,int monthsToSub) | 建立 DateTime 物件，這是從指定 DateTime 減去指定月數的結果。<br><br>範例：#{minusMonths(myDateTime,1)}<br><br>結果："2011-04-24T17:10:00z" |
| DateTime minusWeeks(DateTime myDateTime,int weeksToSub) | 建立 DateTime 物件，這是從指定 DateTime 減去指定週數的結果。<br><br>範例：#{minusWeeks(myDateTime,1)}<br><br>結果："2011-05-17T17:10:00z" |
| DateTime minusYears(DateTime myDateTime,int yearsToSub) | 建立 DateTime 物件，這是從指定 DateTime 減去指定年數的結果。<br><br>範例：#{minusYears(myDateTime,1)}<br><br>結果："2010-05-24T17:10:00z" |
| int minute(DateTime myDateTime) | 取得 DateTime 值的分鐘 (以整數表示)。<br><br>範例：#{minute(myDateTime)}<br><br>結果：10 |

| 函數 | 描述 |
| --- | --- |
| `int month(DateTime myDateTime)` | 取得 DateTime 值的月 (以整數表示)。<br><br>範例：`#{month(myDateTime)}`<br><br>結果：`5` |
| `DateTime plusDays(DateTime myDateTime,int daysToAdd)` | 建立 DateTime 物件，這是將指定天數加上指定 DateTime 的結果。<br><br>範例：`#{plusDays(myDateTime,1)}`<br><br>結果：`"2011-05-25T17:10:00z"` |
| `DateTime plusHours(DateTime myDateTime,int hoursToAdd)` | 建立 DateTime 物件，這是將指定時數加上指定 DateTime 的結果。<br><br>範例：`#{plusHours(myDateTime,1)}`<br><br>結果：`"2011-05-24T18:10:00z"` |
| `DateTime plusMinutes(DateTime myDateTime,int minutesToAdd)` | 建立 DateTime 物件，這是將指定分鐘數加上指定 DateTime 的結果。<br><br>範例：`#{plusMinutes(myDateTime,1)}`<br><br>結果：`"2011-05-24 17:11:00z"` |

| 函數 | 描述 |
| --- | --- |
| `DateTime plusMonths(DateTime myDateTime,int monthsToAdd)` | 建立 DateTime 物件，這是將指定月數加上指定 DateTime 的結果。<br><br>範例：`#{plusMonths(myDateTime,1)}`<br><br>結果：`"2011-06-24T17:10:00z"` |
| `DateTime plusWeeks(DateTime myDateTime,int weeksToAdd)` | 建立 DateTime 物件，這是將指定週數加上指定 DateTime 的結果。<br><br>範例：`#{plusWeeks(myDateTime,1)}`<br><br>結果：`"2011-05-31T17:10:00z"` |
| `DateTime plusYears(DateTime myDateTime,int yearsToAdd)` | 建立 DateTime 物件，這是將指定年數加上指定 DateTime 的結果。<br><br>範例：`#{plusYears(myDateTime,1)}`<br><br>結果：`"2012-05-24T17:10:00z"` |

| 函數 | 描述 |
| --- | --- |
| `DateTime sunday(DateTime myDateTime)` | 建立相對於指定 DateTime，上週日的 DateTime 物件。如果指定的 DateTime 為星期日，結果為指定的 DateTime。<br><br>範例：`#{sunday(myDateTime)}`<br><br>結果：`"2011-05-22 17:10:00 UTC"` |
| `int year(DateTime myDateTime)` | 取得 DateTime 值的年 (以整數表示)。<br><br>範例：`#{year(myDateTime)}`<br><br>結果：`2011` |
| `DateTime yesterday(DateTime myDateTime)` | 建立相對於指定 DateTime，昨天的 DateTime 物件。結果與 minusDays(1) 相同。<br><br>範例：`#{yesterday(myDateTime)}`<br><br>結果：`"2011-05-23T17:10:00z"` |

# 特殊字元

AWS Data Pipeline 使用在管道定義中具有特殊意義的特定字元，如下表所示。

| 特殊字元 | 描述 | 範例 |
| --- | --- | --- |
| @ | 執行時間欄位。此字元是欄位的欄位名稱前綴，只能在管道執行時使用。 | @actualStartTime<br><br>@failureReason<br><br>@resourceStatus |
| # | 表達式。表達式是以 "#{" 和 "}" 分隔，並由 AWS Data Pipeline 評估括號的內容。如需詳細資訊，請參閱 表達式。 | #{format(myDateTime,'YYYY-MM-dd hh:mm:ss')}<br><br>s3://mybucket/#{id}.csv |
| * | 加密欄位。此字元是欄位名稱前綴，表示 AWS Data Pipeline 應該在主控台或 CLI 與 AWS Data Pipeline 服務之間傳輸時，加密此欄位的內容。 | *password |

# 管道物件參考

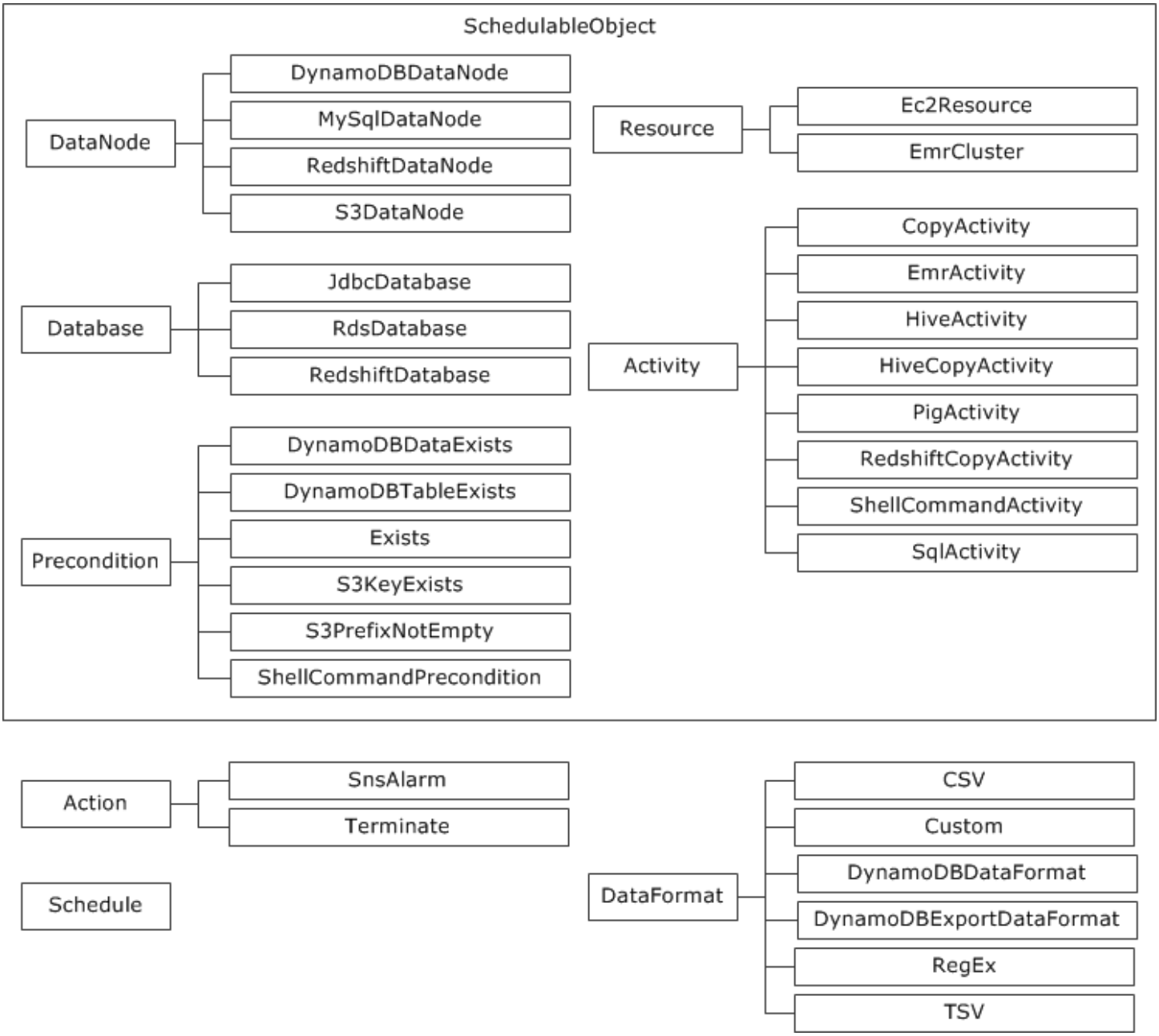
您可以在您的管道定義中使用下列管道物件和元件。

目錄



> **Note**
>
> 如需使用 AWS Data Pipeline Java 的範例應用程式SDK，請參閱上 DynamoDB Data Pipeline 匯出 Java 範例。 GitHub

以下是的物件階層 AWS Data Pipeline。




# 資料節點

以下是 AWS Data Pipeline 數據節點對象：

物件

- D ynamoDBData 節點
- MySqlDataNode
- RedshiftDataNode

- S3 DataNode
- SqlDataNode

# D ynamoDBData 節點

使用 DynamoDB 定義資料節點，該節點指定為HiveActivity或EMRActivity物件的輸入。

> **Note**
>
> DynamoDBDataNode 物件不支援 Exists 先決條件。

## 範例

以下為此物件類型的範例。此物件會參考兩個您在相同管道定義檔案中定義的其他物件。CopyPeriod 是 Schedule 物件，Ready 則是先決條件物件。

```
{
  "id" : "MyDynamoDBTable",
  "type" : "DynamoDBDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "tableName" : "adEvents",
  "precondition" : { "ref" : "Ready" }
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| tableName | 動 DynamoDB 料表。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{「ref」：」mySch eduleId「} | 參考對象，例如，「時間表」：{「re f」：」mySch eduleId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | {"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱排程。 | |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果您已設定此欄位，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dataFormat | DataFormat 此資料節點所描述的資料。目前支援 HiveActivity 與 HiveCopyActivity。 | 參考物件，data Format"": {"ref」:」m yDynamo DBDataFor matId「} |
| dependsOn | 指定與另一個可執行物件的相依性 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId 「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| readThroughputPercent | 設定讀取操作的比率，以將 DynamoDB 佈建的輸送量比率維持在您資料表分配到的範圍內。該值為介於 0.1 和 1.0 (含) 之間的雙倍值。 | Double |
| region | DynamoDB 資料表所在的區域代碼。例如 us-east-1。 HiveActivity 當它在蜂巢中執行暫存 DynamoDB 資料表時，會使用此功能。 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |
| writeThroughputPercent | 設定寫入操作的比率，以將 DynamoDB 佈建的輸送量比率維持在您資料表分配到的範圍內。該值為介於 0.1 和 1.0 (含) 之間的雙倍值。 | Double |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# MySqlDataNode

使用「我」定義資料節點SQL。

> **Note**
>
> MySqlDataNode 類型已移除。我們建議您改用 SqlDataNode。

## 範例

以下為此物件類型的範例。此物件會參考兩個您在相同管道定義檔案中定義的其他物件。CopyPeriod 是 Schedule 物件，Ready 則是先決條件物件。

```
{
  "id" : "Sql Table",
  "type" : "MySqlDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "table" : "adEvents",
  "username": "user_name",
  "*password": "my_password",
  "connectionString": "jdbc:mysql://mysqlinstance-rds.example.us-
east-1.rds.amazonaws.com:3306/database_name",
  "selectQuery" : "select * from #{table} where eventTime >=
 '#{@scheduledStartTime.format('YYYY-MM-dd HH:mm:ss')}' and eventTime <
 '#{@scheduledEndTime.format('YYYY-MM-dd HH:mm:ss')}'",
  "precondition" : { "ref" : "Ready" }
}
```

語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| table | 我的資SQL料庫中資料表的名稱。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的 | 參考對象，例如「時間表」：{「ref |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 」：」mySche duleId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| createTableSql | SQL建立資料表的建立資料表運算式。 | 字串 |
| database | 資料庫的名稱。 | 引用對象，例如「數據庫」：{「ref」：」myData baseId「} |
| dependsOn | 指定與其他可執行物件的相依性。 | 引用對象，例如dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| insertQuery | 將資料插入資料表的SQL陳述式。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| schemaName | 保留資料表的結構描述名稱 | 字串 |
| selectQuery | 從表中獲取數據的SQL語句。 | 字串 |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「： |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | {" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFa iledOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromI nstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthSta tusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- S3 DataNode

## RedshiftDataNode

使用 Amazon Redshift 定義資料節點。 RedshiftDataNode代表管線使用的資料庫內資料的屬性，例如資料表。

### 範例

以下為此物件類型的範例。

```
{
  "id" : "MyRedshiftDataNode",
  "type" : "RedshiftDataNode",
  "database": { "ref": "MyRedshiftDatabase" },
  "tableName": "adEvents",
  "schedule": { "ref": "Hour" }
}
```

### 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| database | 資料表所在的資料庫。 | 引用對象，例如「數據庫」：{「ref」：「myRedshiftDatabaseID」} |
| tableName | Amazon Redshift 資料表的名稱。如果資料表尚未存在且您已提供，則會建立該資料表 createTableSql。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如「時間表」：{「ref」：」mScheduleId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| createTableSql | 在資料庫中建立資料表的SQL運算式。我們建議您指定應在其中建立資料表的結構描述，例如：CREATETABLEmySchema。 myTable（bestColumn varchar（25）主鍵間隔鍵，numberOfWins整數）。sortKey AWS Data Pipeline 如果由指定的資料表不存在於由createTableSql tableName欄位指定的結構描述中，則會在 schemaName 欄位中執行指令碼。例如，如果您在 createTableSql 欄位中指定 schemaName 為 mySchema 但不包含，則會 mySchema 在錯誤的結構描述中建立資料表 (預設會在中建立該資料表PUBLIC)。發生這種 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 情況是因為 AWS Data Pipeline 不會剖析您的 CREATETABLE陳述式。 | |
| dependsOn | 指定與另一個可執行物件的相依性 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myA ctionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myA ctionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myA ctionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| primaryKeys | 如果您沒有在中指定 primaryKeys 目標資料表`RedShiftCopyActivity`，您可以指定欄的清單，使用 primaryKeys 該清單將充當mergeKey。不過，如果您在 Amazon Redshift 表格中有已 primaryKey 定義的現有金鑰，則此設定會覆寫現有的金鑰。 | 字串 |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| schemaName | 此選用欄位會指定 Amazon Redshift 資料表的結構描述名稱。如果未指定，則結構描述名稱為PUBLIC，這是 Amazon Redshift 中的預設結構描述。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》。 | 字串 |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myRunnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{"ref」：」myRunnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

## S3 DataNode

使用 Amazon S3 定義資料節點。根據預設，S3 DataNode 使用伺服器端加密。如果您想要停用此功能，請將 s3 設定EncryptionType 為NONE。

> **Note**
>
> 當您使用作`S3DataNode`為輸入時`CopyActivity`，僅支援CSV和TSV資料格式。

## 範例

以下為此物件類型的範例。此物件會參考您在相同管道定義檔案中定義的另一個物件。CopyPeriod 是 Schedule 物件。

```
{
  "id" : "OutputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://myBucket/#{@scheduledStartTime}.csv"
}
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如「時間表」：{「ref」：」mySche duleId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| compression | S3 所描述之資料的壓縮類型DataNode。「無」是沒有壓縮，「gzip」是用 gzip 算法壓縮的。只有在搭配使用 S3 時，才支援此欄位與 Amazon Redshift DataNode 搭配 CopyActivity 使用。 | 列舉 |
| dataFormat | DataFormat 對於此 S3 描述的數據DataNode。 | 引用對象，例如 dataFormat「「：{" ref」：」myD ataFormat ID "} |
| dependsOn | 指定與另一個可執行物件的相依性 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId「} |
| directoryPath | Amazon S3 目錄路徑為URI：s3://my-bucket/ my-key-for-directory。您必須提供 filePath 或 directoryPath 值。 | 字串 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| filePath | Amazon S3 中的對象的路徑作為一個URI，例如：s3://my-bucket/ my-key-for-file。您必須提供 filePath 或 directoryPath 值。這些項目代表資料夾和檔案名稱。使用此 directoryPath 值可容納目錄中的多個檔案。 | 字串 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| manifestFilePath | 資訊清單檔案的 Amazon S3 路徑，採用 Amazon Redshift 支援的格式。 AWS Data Pipeline 使用資訊清單檔案將指定的 Amazon S3 檔案複製到資料表中。只有當 RedShiftCopyActivity 參考 S3 時，此欄位才有效DataNode。 | 字串 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如runsOn「「：{" ref」：」myResourceId「} |
| S3 EncryptionType | 覆寫 Amazon S3 加密類型。值為 SERVER _ SIDE _ ENCRYPTION 或NONE。預設啟用伺服器端加密。 | 列舉 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

### 另請參閱

- MySqlDataNode

## SqlDataNode

定義使用 SQL 的資料節點。

### 範例

以下為此物件類型的範例。此物件會參考兩個您在相同管道定義檔案中定義的其他物件。CopyPeriod 是 Schedule 物件，Ready 則是先決條件物件。

```
{
  "id" : "Sql Table",
  "type" : "SqlDataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "table" : "adEvents",
  "database":"myDataBaseName",
  "selectQuery" : "select * from #{table} where eventTime >=
 '#{@scheduledStartTime.format('YYYY-MM-dd HH:mm:ss')}' and eventTime <
 '#{@scheduledEndTime.format('YYYY-MM-dd HH:mm:ss')}'",
  "precondition" : { "ref" : "Ready" }
}
```

### 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| table | SQL資料庫中資料表的名稱。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況 | 參考對象，例如「時間表」：{「ref」：」mScheduleId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| createTableSql | SQL建立資料表的建立資料表運算式。 | 字串 |
| database | 資料庫的名稱。 | 引用對象，例如「數據庫」：{「ref」：」myDatabaseId「} |
| dependsOn | 指定與其他可執行物件的相依性。 | 引用對象，例如dependsOn「「：{"ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| insertQuery | 將資料插入資料表的SQL陳述式。 | 字串 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myR esourceId「} |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleT ype 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| schemaName | 保留資料表的結構描述名稱 | 字串 |
| selectQuery | 從表中獲取數據的SQL語句。 | 字串 |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{"ref」：」myRunnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

## 另請參閱

- S3 DataNode

# 活動

以下是 AWS Data Pipeline 活動對象：

物件

- CopyActivity
- EmrActivity
- HadoopActivity
- HiveActivity
- HiveCopyActivity
- PigActivity
- RedshiftCopyActivity
- ShellCommandActivity
- SqlActivity

## CopyActivity

將資料從一個位置複製到另一個位置。 CopyActivity支持S3 DataNode和作SqlDataNode為輸入和輸出，並通常執行複製操作 record-by-record。不過，當符合下列所有條件時，會向 Amazon S3 CopyActivity 提供高效能的 Amazon S3 副本：

- 輸入和輸出為 S3 DataNodes
- 輸入及輸出的 dataFormat 欄位皆相同。

若您提供壓縮資料檔案做為輸入，而並未在 S3 資料節點上使用 compression 欄位指出，則 CopyActivity 可能會失敗。在這種情況下，CopyActivity 將無法正確地偵測記錄結尾字元，導致操作失敗。此外，CopyActivity支援從目錄複製到另一個目錄，並將檔案複製到目錄，但 record-by-record 複製目錄到檔案時會發生複製。最後，CopyActivity不支援複製多部分 Amazon S3 檔案。

CopyActivity對其CSV支持有特定的限制。當您使用 S3 DataNode 做為的輸入時CopyActivity，您只能將CSV資料檔案格式的 Unix/Linux 變體用於 Amazon S3 輸入和輸出欄位。Unix/Linux 變體需要下列項目：

- 分隔符號必須是 "," (逗號) 字元。

- 記錄不會加上引號。
- 預設的逸出字元ASCII值為 92 (反斜線)。
- 記錄標識符的結尾是ASCII值 10 ( 或「\ n「 ) 。

基於 Windows 的系統通常使用不同的 end-of-record 字符序列：回車符和換行 ( ASCII值 13 和ASCII值 10 ) 。您必須使用額外的機制來配合此差異，例如使用一個預先複製指令碼來修改輸入資料，確保 CopyActivity 能正確地偵測記錄結尾；否則，CopyActivity 會不斷失敗。

當使CopyActivity用從 Postgre SQL RDS 對象導出到TSV數據格式，默認NULL字符是\ n。

## 範例

以下為此物件類型的範例。此物件會參考三個您在相同管道定義檔案中定義的其他物件。CopyPeriod 是 Schedule 物件，InputData 和 OutputData 則是資料節點物件。

```
{
  "id" : "S3ToS3Copy",
  "type" : "CopyActivity",
  "schedule" : { "ref" : "CopyPeriod" },
  "input" : { "ref" : "InputData" },
  "output" : { "ref" : "OutputData" },
  "runsOn" : { "ref" : "MyEc2Resource" }
}
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用 | 參考對象，例如「時間表」：{「ref」：」mySche duleId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與另一個可執行物件的相依性。 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| input | 輸入資料來源。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| output | 輸出資料來源。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeID」} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myRunnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{"ref」：」myRunnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

## 另請參閱

- ShellCommandActivity
- EmrActivity
- 使用將 MySQL 數據導出到亞馬遜 S3 AWS Data Pipeline

# EmrActivity

執行 EMR 叢集。

AWS Data Pipeline 使用與 Amazon 不同的步驟格式EMR；例如，在EmrActivity步驟欄位的JAR名稱後面 AWS Data Pipeline 使用逗號分隔的引數。下列範例顯示針對 Amazon 格式化的步驟EMR，接著是 AWS Data Pipeline 等效的步驟：

```
s3://example-bucket/MyWork.jar arg1 arg2 arg3
```

```
"s3://example-bucket/MyWork.jar,arg1,arg2,arg3"
```

## 範例

以下為此物件類型的範例。此範例使用較舊版本的 Amazon EMR。驗證此範例是否與您使用的 Amazon EMR 叢集版本的正確性。

此物件會參考三個您在相同管道定義檔案中定義的其他物件。MyEmrCluster 是 EmrCluster 物件，MyS3Input 和 MyS3Output 則是 S3DataNode 物件。

> **Note**
>
> 在此示例中，您可以用所需的集群字符串替換該字step段，該字符串可以是 Pig 腳本，Hadoop 流集群，您自己的自定義JAR包括其參數等。

哈達通 2. X (3. X) AMI

```
{
  "id" : "MyEmrActivity",
  "type" : "EmrActivity",
  "runsOn" : { "ref" : "MyEmrCluster" },
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : ["s3://mybucket/myPath/myStep.jar,firstArg,secondArg,-files,s3://mybucket/
myPath/myFile.py,-input,s3://myinputbucket/path,-output,s3://myoutputbucket/path,-
mapper,myFile.py,-reducer,reducerName","s3://mybucket/myPath/myotherStep.jar,..."],
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : { "ref" : "MyS3Input" },
  "output" : { "ref" : "MyS3Output" }
```

```
}
```

> **Note**
>
> 若要在步驟中將引述傳遞給應用程式，您需要在指令碼的路徑中指定區域，如以下範例所示。此外，您可能需要逸出您傳遞的引數。例如，若您使用 `script-runner.jar` 執行殼層指令碼，並希望將引數傳遞給指令碼，您必須逸出分隔他們的逗號。以下步驟位置示範如何執行此作業：
>
> ```
> "step" : "s3://eu-west-1.elasticmapreduce/libs/script-runner/script-
> runner.jar,s3://datapipeline/echo.sh,a\\\\,b\\\\,c"
> ```
>
> 此步驟使用 `script-runner.jar` 執行 `echo.sh` 殼層指令碼，並將 a、b 和 c 做為單一引數傳遞給指令碼。第一個逸出字元會從結果引數中移除，因此您可能需要再次進行逸出。例如，如果您有`File\.gz`作為中的參數JSON，則可以使用`File\\\\.gz`. 但是，由於第一個逸出會遭到捨棄，因此您必須使用 `File\\\\\\\\.gz` 。

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。指定其他物件的排程參考，以設定此物件的相依性執行順序。您可以在物件上明確設定排程以滿足這項要求，例如，指定 `"schedule": {"ref": "DefaultSchedule"}` 。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，您可以建立含排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如，「時間表」：{「ref」：」mySchEduleId「} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 將在其上執行此任務的 Amazon EMR 叢集。 | 參考物件，例如，" runsOn" ": {" ref」: "myEmrClusterId"} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供 `runsOn` 值，且 `workerGroup` 存在，則會忽略 `workerGroup` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與另一個可執行物件的相依性。 | 參考物件，例如，dependsOn"": {"ref」:」myActivity Id「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| input | 輸入資料的位置。 | 引用對象，例如，「輸入」：{「ref」：」myDataNode Id「} |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件，例如，onFail"": {"ref」:」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 參考物件，例如，onLateAction"": {"ref」:」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件，例如，onSuccess"": {"ref」:」myActionId「} |
| output | 輸出資料的位置。 | 引用對象，例如，「輸出」：{「ref」：」myDataNodeId「} |
| parent | 目前物件的父系，其槽會被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObjectId「} |
| pipelineLogUri | Amazon S3URI，例如 's3：///BucketName前綴/'，用於上傳管道的日誌。 | 字串 |
| postStepCommand | 完成所有步驟後要執行的 Shell 指令碼。若要指定多個指令碼 (最多 255 個)，請新增多個 `postStepCommand` 欄位。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如，「先決條件」：{「ref」：」myPreconditionId「} |
| preStepCommand | 執行任何步驟之前要執行的 Shell 指令碼。若要指定多個指令碼 (最多 255 個)，請新增多個 `preStepCommand` 欄位。 | 字串 |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| resizeClusterBefore跑步 | 在執行此活動之前重新調整叢集大小，以容納指定為輸入或輸出的 DynamoDB 表。<br><br>Note<br>如果您`EmrActivity` 使用 a `DynamoDBDataNode` 作為輸入或輸出資料節點，並且`resizeClusterBeforeRunning` 將設定為`TRUE`，則會 AWS Data Pipeline 開始使用`m3.xlarge` 執行個體類型。這會將您選擇的執行個體類型覆寫為 `m3.xlarge` ，可能會增加您的每月成本。 | Boolean |
| resizeClusterMax實例 | 調整大小演算法可請求的執行個體數目上限。 | Integer |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。值為：cron、ondemand 和 timeseries 。timeseries 排程表示執行個體會排程在每個間隔的結尾。cron 排程表示執行個體會排程在每個間隔的開頭。ondemand 排程可讓您在每次啟用時執行一次管道。您不必複製或重新建立管道，然後再執行一次。若您使用 ondemand 排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 scheduleType 。若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。 | 列舉 |
| 步驟 | 叢集要執行的一或多個步驟。若要指定多個步驟 (最多 255 個)，請新增多個步驟欄位。在JAR名稱之後使用逗號分隔的引數；例如，"s3://example-bucket/MyWork.jar,arg1,arg2,arg3 "。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myRunnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如，" cascadeFailedOn" ": |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | {" ref」: "myRunnableObjectId"} |
| emrStepLog | Amazon EMR 步驟日誌僅在嘗試EMR活動時可用 | 字串 |
| errorId | 若此物件失敗，會提供 errorId。 | 字串 |
| errorMessage | 若此物件失敗，會提供 errorMessage 。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件時使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如，"waitingOn" ": {" ref」："myRunnableObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- ShellCommandActivity
- CopyActivity
- EmrCluster

## HadoopActivity

在叢集上執行 MapReduce 工作。叢集可以是由管理的EMR叢集，也可以是其他 AWS Data Pipeline 資源 (如果您使用的話) TaskRunner。 HadoopActivity 當您要 parallel 執行工作時使用。這使您可以

在 Hadoop 1 中使用YARN框架的調度 MapReduce 資源或資源談判者。如果您想要使用 EmrActivity Amazon EMR Step 動作依序執行工作，您仍然可以使用。

## 範例

HadoopActivity 使用由管理的EMR叢集 AWS Data Pipeline

下列 HadoopActivity 物件會使用 EmrCluster 資源執行程式：

```
 {
   "name": "MyHadoopActivity",
   "schedule": {"ref": "ResourcePeriod"},
   "runsOn": {"ref": “MyEmrCluster”},
   "type": "HadoopActivity",
   "preActivityTaskConfig":{"ref":"preTaskScriptConfig”},
   "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
   "argument": [
     "-files",
     “s3://elasticmapreduce/samples/wordcount/wordSplitter.py“,
     "-mapper",
     "wordSplitter.py",
     "-reducer",
     "aggregate",
     "-input",
     "s3://elasticmapreduce/samples/wordcount/input/",
     "-output",
     “s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
   ],
   "maximumRetries": "0",
   "postActivityTaskConfig":{"ref":"postTaskScriptConfig”},
   "hadoopQueue" : “high”
 }
```

這裡是相應的 *MyEmrCluster*，其中配置了基YARN於 Hadoop 2 的 FairScheduler 和隊列：AMIs

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
   "hadoopSchedulerType" : "PARALLEL_FAIR_SCHEDULING",
  “amiVersion” : “3.7.0”,
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
```

```
\,high\,default,-z,yarn.scheduler.capacity.root.high.capacity=50,-
z,yarn.scheduler.capacity.root.low.capacity=10,-
z,yarn.scheduler.capacity.root.default.capacity=30”]
}
```

這是 EmrCluster 您在 Hadoop 1 FairScheduler 中使用的配置：

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_FAIR_SCHEDULING",
      "amiVersion": "2.4.8",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-m,mapred.queue.names=low\\\\,high\\\\,default,-
m,mapred.fairscheduler.poolnameproperty=mapred.job.queue.name"
          }
```

以下為 EmrCluster 基 CapacityScheduler 於 Hadoop 2 的配置：AMIs

```
{
      "id": "MyEmrCluster",
      "type": "EmrCluster",
      "hadoopSchedulerType": "PARALLEL_CAPACITY_SCHEDULING",
      "amiVersion": "3.7.0",
      "bootstrapAction": "s3://Region.elasticmapreduce/bootstrap-
actions/configure-hadoop,-z,yarn.scheduler.capacity.root.queues=low
\\\\,high,-z,yarn.scheduler.capacity.root.high.capacity=40,-
z,yarn.scheduler.capacity.root.low.capacity=60"
    }
```

HadoopActivity 使用現有EMR叢集

在此範例中，您可 TaskRunner 以使用工作者群組和 a 在現有EMR叢集上執行程式。下列配管定義用 HadoopActivity 於：

- 僅在以下位置運行 MapReduce 程序 *myWorkerGroup* 的費用。如需工作者群組的詳細資訊，請參閱使用任務運行器對現有資源執行工作。
- 運行 Con preActivityTask fig 和 postActivityTask Config

```
{
  "objects": [
```

```
    {
      "argument": [
        "-files",
        "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
        "-mapper",
        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input/",
        "-output",
        "s3://test-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
      ],
      "id": "MyHadoopActivity",
      "jarUri": "/home/hadoop/contrib/streaming/hadoop-streaming.jar",
      "name": "MyHadoopActivity",
      "type": "HadoopActivity"
    },
    {
      "id": "SchedulePeriod",
      "startDateTime": "start_datetime",
      "name": "SchedulePeriod",
      "period": "1 day",
      "type": "Schedule",
      "endDateTime": "end_datetime"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://test-bucket/scripts/preTaskScript.sh",
      "name": "preTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://test-bucket/scripts/postTaskScript.sh",
      "name": "postTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
```

```
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "Default",
      "scheduleType": "cron",
      "schedule": {
        "ref": "SchedulePeriod"
      },
      "name": "Default",
      "pipelineLogUri": "s3://test-bucket/logs/2015-05-22T18:02:00.343Z642f3fe415",
      "maximumRetries": "0",
      "workerGroup": "myWorkerGroup",
      "preActivityTaskConfig": {
        "ref": "preTaskScriptConfig"
      },
      "postActivityTaskConfig": {
        "ref": "postTaskScriptConfig"
      }
    }
  ]
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| jarUri | Amazon S3 JAR 中的位置或要與之一起執行的叢集本機檔案系統 HadoopActivity。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有 | 參考對象，例如「時間表」：{「ref」：」mySche duleId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | EMR執行此工作的叢集。 | 引用對象，例如runsOn「「：{" ref」：」myEmrCluster ID "} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個 runsOn 值並且 workerGroup 存在，workerGroup 則會忽略。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| argument | 要傳遞給的引數JAR。 | 字串 |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與另一個可執行物件的相依性。 | 引用對象，例如dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hadoopQueue | 要提交活動至其中的 Hadoop 排程器佇列名稱。 | 字串 |
| input | 輸入資料的位置。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| mainClass | JAR您正在執行的主要類別 HadoopActivity。 | 字串 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| output | 輸出資料的位置。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeID」} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| postActivityTaskConfig | 要執行的活動後組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例如「postActivityTaskConfig」：{「ref」：」myShellScript ConfigId「} |
| preActivityTaskConfig | 要執行的活動前組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例如「preActivityTaskConfig」：{「ref」：」myShellScript ConfigId「} |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- ShellCommandActivity
- CopyActivity
- EmrCluster

## HiveActivity

在EMR集群上運行蜂巢查詢。 `HiveActivity`可讓您更輕鬆地設定 Amazon EMR 活動，並根據來自 Amazon S3 或 Amazon 的輸入資料自動建立 Hive 資料表RDS。您需要指定的是要在源數據上運行的 HiveQL。 AWS Data Pipeline 根據物件中的輸入欄位 `${input1}${input2}`，自動建立具有、等的 Hive 資料`HiveActivity`表。

對於 Amazon S3 輸入，此`dataFormat`欄位是用來建立 Hive 資料行名稱。

對於我的SQL（AmazonRDS）輸入，SQL查詢的列名稱用於創建蜂巢列名。

> **Note**
>
> 此活動使用蜂巢CSV服務。

## 範例

以下為此物件類型的範例。此物件會參考三個您在相同管道定義檔案中定義的其他物件。MySchedule 是 Schedule 物件，MyS3Input 和 MyS3Output 則是資料節點物件。

```
{
  "name" : "ProcessLogData",
  "id" : "MyHiveActivity",
  "type" : "HiveActivity",
  "schedule" : { "ref": "MySchedule" },
  "hiveScript" : "INSERT OVERWRITE TABLE ${output1} select
 host,user,time,request,status,size from ${input1};",
  "input" : { "ref": "MyS3Input" },
  "output" : { "ref": "MyS3Output" },
  "runsOn" : { "ref": "MyEmrCluster" }
}
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。指定其他物件的排程參考，以設定此物件的相依性執行順序。您可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，您可以建立含排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱 https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如「時間表」：{「ref」：」mScheduleId「} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| hiveScript | 要執行的 Hive 指令碼。 | 字串 |
| scriptUri | 要執行的 Hive 指令碼的位置 (例如 s3://scriptLocation)。 | 字串 |

| 必要群組 | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | `HiveActivity` 執行此動作的EMR叢集。 | 引用對象，例如 runsOn「「：{" ref」：」myEmrCluster ID "} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供 `runsOn` 值，且 `workerGroup` 存在，則會忽略 `workerGroup` 。 | 字串 |
| input | 輸入資料來源。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |
| output | 輸出資料來源。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeId」} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| dependsOn | 指定與另一個可執行物件的相依性。 | 參考物件，例如 dependsOn "" ": {" ref」:」myActivityId 「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| hadoopQueue | 要提交任務至其中的 Hadoop 排程器佇列名稱。 | 字串 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件，例如 onFail "" ": {" ref」:」myActionId 「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 參考物件，例如 onLateAction "" ": {" ref」:」myActionId 「} |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件，例如 onSuccess "" ": {" ref」:」myActionId 「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如 「父」 : {「ref」 : 「myBaseObjectId」} |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| postActivityTaskConfig | 要執行的活動後組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例如「postActivityTaskConfig」：{「ref」：」myShellScript ConfigId「} |
| preActivityTaskConfig | 要執行的活動前組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例如「preActivityTaskConfig」：{「ref」：」myShellScript ConfigId「} |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「先決條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| resizeClusterBefore跑步 | 在執行此活動之前重新調整叢集大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。<br><br>**Note**<br>如果您的活動使用 a `DynamoDBDataNode` 作為輸入或輸出資料節點，並且`resizeClusterBeforeRunning` 將設定為`TRUE`，則會 AWS Data Pipeline 開始使用`m3.xlarge` 執行個體類型。這會將您選擇的執行個體類型覆寫為 `m3.xlarge` ，可能會增加您的每月成本。 | Boolean |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| resizeClusterMax實例 | 調整大小演算法可請求的執行個體數目上限。 | Integer |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| scriptVariable | 指定 Amazon 在運行腳本時傳遞EMR給 Hive 的腳本變量。例如，下列範例指令碼變數會將SAMPLE和 FILTER _ DATE 變數傳遞至 Hive：`SAMPLE=s3://elasticmapreduce/samples/hive-ads` 和 `FILTER_DATE=#{format(@scheduledStartTime,'YYYY-MM-dd')}%` 。此欄位接受多個值，並可使用 `script` 和 `scriptUri` 欄位。此外，無論預備是設為 `true` 或 `false`，`scriptVariable` 都會正常運作。此欄位在使用 AWS Data Pipeline 表達式和函數，將動態值傳送給 Hive 時特別有用。 | 字串 |
| stage | 決定在執行指令碼之前或之後是否啟用預備。蜂巢 11 不允許使用，因此請使用 Amazon EMR AMI 版本 3.2.0 或更高版本。 | Boolean |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如 "activeInstances" ": {" ref」: "myRunnableObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如 "cascadeFailedOn" ": {" ref」: "myRunnableObjectId"} |
| emrStepLog | Amazon EMR 步驟日誌僅適用於EMR活動嘗試。 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程啟動時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如 "waitingOn" ": {" ref」：<br>"myRunnableObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- ShellCommandActivity
- EmrActivity

## HiveCopyActivity

在EMR集群上運行蜂巢查詢。 HiveCopyActivity可讓您更輕鬆地在 DynamoDB 資料表之間複製資料。 HiveCopyActivity接受 HiveQL 陳述式，以便在資料行和資料列層級篩選來自 DynamoDB 的輸入資料。

### 範例

以下範例會示範如何使用 HiveCopyActivity 和 DynamoDBExportDataFormat 來將資料從一個 DynamoDBDataNode 複製到另一個，同時根據時間戳記來篩選資料。

```
{
  "objects": [
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",
      "type" : "DynamoDBExportDataFormat",
      "column" : "timeStamp BIGINT"
    },
    {
      "id" : "DataFormat.2",
      "name" : "DataFormat.2",
      "type" : "DynamoDBExportDataFormat"
    },
    {
      "id" : "DynamoDBDataNode.1",
      "name" : "DynamoDBDataNode.1",
      "type" : "DynamoDBDataNode",
      "tableName" : "item_mapped_table_restore_temp",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "DynamoDBDataNode.2",
      "name" : "DynamoDBDataNode.2",
      "type" : "DynamoDBDataNode",
```

```
      "tableName" : "restore_table",
      "region" : "us_west_1",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.2" }
    },
    {
      "id" : "EmrCluster.1",
      "name" : "EmrCluster.1",
      "type" : "EmrCluster",
      "schedule" : { "ref" : "ResourcePeriod" },
      "masterInstanceType" : "m1.xlarge",
      "coreInstanceCount" : "4"
    },
    {
      "id" : "HiveTransform.1",
      "name" : "Hive Copy Transform.1",
      "type" : "HiveCopyActivity",
      "input" : { "ref" : "DynamoDBDataNode.1" },
      "output" : { "ref" : "DynamoDBDataNode.2" },
      "schedule" :{ "ref" : "ResourcePeriod" },
      "runsOn" : { "ref" : "EmrCluster.1" },
      "filterSql" : "`timeStamp` > unix_timestamp(\"#{@scheduledStartTime}\", \"yyyy-
MM-dd'T'HH:mm:ss\")"
    },
    {
      "id" : "ResourcePeriod",
      "name" : "ResourcePeriod",
      "type" : "Schedule",
      "period" : "1 Hour",
      "startDateTime" : "2013-06-04T00:00:00",
      "endDateTime" : "2013-06-04T01:00:00"
    }
  ]
}
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設 | 參考對象，例如「時間表」：{「ref |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 」：」mySche duleId「} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 指定要在其中執行的叢集。 | 引用對象，例如 runsOn「「：{" ref」：」myR esourceId「} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供 `runsOn` 值，且 `workerGroup` 存在，則會忽略 `workerGroup` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與其他可執行物件的相依性。 | 引用對象，例如 dependsOn「「：{" |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| filterSql | Hive SQL 陳述式片段，可篩選要複製的 DynamoDB 資料子集或 Amazon S3 資料的子集。篩選器應該只包含述詞，而不是以WHERE子句開頭，因為它會自動 AWS Data Pipeline 加入。 | 字串 |
| input | 輸入資料來源。此必須為 S3DataNode 或 DynamoDBDataNode 。如果您使用 DynamoDBNode ，請指定 DynamoDBExportDataFormat 。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myA ctionId「} |
| output | 輸出資料來源。如果輸入是 S3DataNode ，這必須為 DynamoDBDataNode 。否則，此項目可以是 S3DataNode 或 DynamoDBD ataNode 。如果您使用 DynamoDBNode ，請指定 DynamoDBExportDataFormat 。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeID」} |
| parent | 目前物件的父系，其槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | Amazon S3URI，例如 's3://Buc ketName/Key/' ，用於上傳管道的日誌。 | 字串 |
| postActivityTaskCo nfig | 要執行的活動後組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例 如「postActivi tyTaskConfig」：{「ref」：」myShe llScript ConfigId「} |
| preActivityTaskConfig | 要執行的活動前組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例 如「preActivit yTaskConfig」：{「ref」：」myShel lScript ConfigId「} |
| precondition | 可選擇性定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPre conditionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| resizeClusterBefore跑步 | 在執行此活動之前重新調整叢集大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。<br><br>**Note**<br>如果您的活動使用 a `DynamoDBDataNode` 作為輸入或輸出資料節點，並且`resizeClusterBeforeRunning` 將設定為`TRUE`，則會 AWS Data Pipeline 開始使用`m3.xlarge` 執行個體類型。這會將您選擇的執行個體類型覆寫為 `m3.xlarge` ，可能會增加您的每月成本。 | Boolean |
| resizeClusterMax實例 | 調整大小演算法可請求的執行個體數目上限 | Integer |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | Amazon EMR 步驟日誌僅適用於EMR活動嘗試。 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- ShellCommandActivity
- EmrActivity

## PigActivity

PigActivity 提供 Pig 指令碼的原生支援， AWS Data Pipeline 而不需要使用`ShellCommandActivity`或`EmrActivity`。此外， PigActivity 支援資料暫存。當預備欄位設為 True 時， AWS Data Pipeline 會將輸入資料做為 Pig 中的結構描述預備，而無須使用者輸入額外的程式碼。

### 範例

以下範例管道示範如何使用 `PigActivity`。範例管道會執行下列步驟：

- MyPigActivity1 從 Amazon S3 載入資料並執行 Pig 指令碼，選取幾欄資料並將其上傳到 Amazon S3。
- MyPigActivity2 載入第一個輸出，選取幾欄和三列資料，然後將其作為第二個輸出上傳到 Amazon S3。
- MyPigActivity3 加載第二個輸出數據，插入兩行數據，只將名為「第五」的列插入到 Amazon RDS。
- MyPigActivity4 會載入 Amazon RDS 資料、選取第一列資料，然後將資料上傳到 Amazon S3。

```
{
  "objects": [
    {
      "id": "MyInputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://example-bucket/pigTestInput",
      "name": "MyInputData1",
      "dataFormat": {
        "ref": "MyInputDataType1"
      },
      "type": "S3DataNode"
    },
```

```
    {
      "id": "MyPigActivity4",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData3"
      },
      "pipelineLogUri": "s3://example-bucket/path/",
      "name": "MyPigActivity4",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "type": "PigActivity",
      "dependsOn": {
        "ref": "MyPigActivity3"
      },
      "output": {
        "ref": "MyOutputData4"
      },
      "script": "B = LIMIT ${input1} 1; ${output1} = FOREACH B GENERATE one;",
      "stage": "true"
    },
    {
      "id": "MyPigActivity3",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData2"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "name": "MyPigActivity3",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "script": "B = LIMIT ${input1} 2; ${output1} = FOREACH B GENERATE Fifth;",
      "type": "PigActivity",
      "dependsOn": {
        "ref": "MyPigActivity2"
      },
      "output": {
```

```
        "ref": "MyOutputData3"
      },
      "stage": "true"
    },
    {
      "id": "MyOutputData2",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "name": "MyOutputData2",
      "directoryPath": "s3://example-bucket/PigActivityOutput2",
      "dataFormat": {
        "ref": "MyOutputDataType2"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "name": "MyOutputData1",
      "directoryPath": "s3://example-bucket/PigActivityOutput1",
      "dataFormat": {
        "ref": "MyOutputDataType1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyInputDataType1",
      "name": "MyInputDataType1",
      "column": [
        "First STRING",
        "Second STRING",
        "Third STRING",
        "Fourth STRING",
        "Fifth STRING",
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING",
        "Ninth STRING",
        "Tenth STRING"
      ],
```

```
      "inputRegEx": "^(\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\\S+) (\
\S+) (\\S+) (\\S+)",
      "type": "RegEx"
    },
    {
      "id": "MyEmrResource",
      "region": "us-east-1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "keyPair": "example-keypair",
      "masterInstanceType": "m1.small",
      "enableDebugging": "true",
      "name": "MyEmrResource",
      "actionOnTaskFailure": "continue",
      "type": "EmrCluster"
    },
    {
      "id": "MyOutputDataType4",
      "name": "MyOutputDataType4",
      "column": "one STRING",
      "type": "CSV"
    },
    {
      "id": "MyOutputData4",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://example-bucket/PigActivityOutput3",
      "name": "MyOutputData4",
      "dataFormat": {
        "ref": "MyOutputDataType4"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputDataType1",
      "name": "MyOutputDataType1",
      "column": [
        "First STRING",
        "Second STRING",
        "Third STRING",
        "Fourth STRING",
        "Fifth STRING",
```

```
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING"
      ],
      "columnSeparator": "*",
      "type": "Custom"
    },
    {
      "id": "MyOutputData3",
      "username": "___",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "insertQuery": "insert into #{table} (one) values (?)",
      "name": "MyOutputData3",
      "*password": "___",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "connectionString": "jdbc:mysql://example-database-instance:3306/example-
database",
      "selectQuery": "select * from #{table}",
      "table": "example-table-name",
      "type": "MySqlDataNode"
    },
    {
      "id": "MyOutputDataType2",
      "name": "MyOutputDataType2",
      "column": [
        "Third STRING",
        "Fourth STRING",
        "Fifth STRING",
        "Sixth STRING",
        "Seventh STRING",
        "Eighth STRING"
      ],
      "type": "TSV"
    },
    {
      "id": "MyPigActivity2",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
```

```
      "input": {
        "ref": "MyOutputData1"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "name": "MyPigActivity2",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "dependsOn": {
        "ref": "MyPigActivity1"
      },
      "type": "PigActivity",
      "script": "B = LIMIT ${input1} 3; ${output1} = FOREACH B GENERATE Third, Fourth,
Fifth, Sixth, Seventh, Eighth;",
      "output": {
        "ref": "MyOutputData2"
      },
      "stage": "true"
    },
    {
      "id": "MyEmrResourcePeriod",
      "startDateTime": "2013-05-20T00:00:00",
      "name": "MyEmrResourcePeriod",
      "period": "1 day",
      "type": "Schedule",
      "endDateTime": "2013-05-21T00:00:00"
    },
    {
      "id": "MyPigActivity1",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyInputData1"
      },
      "pipelineLogUri": "s3://example-bucket/path",
      "scriptUri": "s3://example-bucket/script/pigTestScipt.q",
      "name": "MyPigActivity1",
      "runsOn": {
        "ref": "MyEmrResource"
      },
      "scriptVariable": [
        "column1=First",
```

```
        "column2=Second",
        "three=3"
      ],
      "type": "PigActivity",
      "output": {
        "ref": "MyOutputData1"
      },
      "stage": "true"
    }
  ]
}
```

pigTestScript.q 的內容如下所示。

```
B = LIMIT ${input1} $three; ${output1} = FOREACH B GENERATE $column1, $column2, Third,
 Fourth, Fifth, Sixth, Seventh, Eighth;
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。使用者必須指定另一個物件的排程參考，設定此物件的相依性執行順序。使用者可以在物件上明確設定排程來滿足此需求，例如指定「schedule」：{"ref」：DefaultSchedule "}。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，使用者可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如，「時間表」：{「ref」：」mySch eduleId「} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| script | 要執行的 Pig 指令碼。 | 字串 |
| scriptUri | 要執行的 Pig 指令碼的位置 (例如 s3://scriptLocation)。 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | EMR PigActivity 執行此動作的叢集。 | 參考物件，例如，"runsOn" ": {" ref」："myEmrClusterId"} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供 `runsOn` 值，且 `workerGroup` 存在，則會忽略 `workerGroup` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與其他可執行物件的相依性。 | 參考物件，例如，dependsOn"": {"ref」:」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| input | 輸入資料來源。 | 引用對象，例如，「輸入」：{「ref」：」myDataNodeId「} |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件，例如，onFail"": {"ref」:」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 參考物件，例如，onLateAction"": {"ref」:」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件，例如，onSuccess"": {"ref」:」myActionId「} |
| output | 輸出資料來源。 | 引用對象，例如，「輸出」：{「ref」：」myDataNodeId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObjectId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| pipelineLogUri | 用於上傳管道日誌的 Amazon S3URI（例如 's3：//BucketName/密鑰/'）。 | 字串 |
| postActivityTaskConfig | 要執行的活動後組態指令碼。這包括 Amazon S33 中URI的 shell 腳本和一個參數列表。 | 引用對象，例如，「postActivityTaskConfig」：{「ref」：」myShellScriptConfigId「} |
| preActivityTaskConfig | 要執行的活動前組態指令碼。這是由 Amazon S3 中URI的殼層指令碼和引數清單所組成。 | 引用對象，例如，「preActivityTaskConfig」：{「ref」：」myShellScript ConfigId「} |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如，「先決條件」：{「ref」：」myPreconditionId「} |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| resizeClusterBefore跑步 | 在執行此活動之前重新調整叢集大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。<br><br>**Note**<br>如果您的活動使用 a DynamoDBDataNode 作為輸入或輸出資料節點，並且resizeClusterBeforeRunning 將設定為TRUE，則會 AWS Data Pipeline 開始使用m3.xlarge 執行個體類型。這會將您選擇的執行個體類型覆寫為 m3.xlarge ，可能會增加您的每月成本。 | Boolean |
| resizeClusterMax實例 | 調整大小演算法可請求的執行個體數目上限。 | Integer |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。時間序列樣式排程表示執行個體排程在每個間隔的結尾，而 Cron 樣式排程表示執行個體排程在每個間隔的開頭。隨需排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。如果您使用隨選排程，則必須在預設物件中指定該排程，且必須是管線中物件的唯一 scheduleType 指定排程。若要使用隨選管線，您只需針對每次後續執行呼叫 ActivatePipeline 作業即可。值為：Cron、ondemand 和 timeseries。 | 列舉 |
| scriptVariable | 要傳遞給 Pig 指令碼的引數。您可以使 scriptVariable 用腳本或scriptUri. | 字串 |
| stage | 判斷是否啟用暫存，並允許 Pig 指令碼存取分段資料表格，例如 $ {INPUT1} 和 $ {OUTPUT1}。 | Boolean |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如，"activeInstances" ": {" ref」: "myRunnableObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如，"cascadeFailedOn" ": {" ref」: "myRunnableObjectId"} |
| emrStepLog | Amazon EMR 步驟日誌僅適用於EMR活動嘗試。 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件時使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如，"waitingOn" ": {" ref」："myRunnableObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

### 另請參閱

- ShellCommandActivity
- EmrActivity

## RedshiftCopyActivity

將資料從 DynamoDB 或 Amazon S3 複製到 Amazon Redshift。您可以將資料載入新的資料表，或是輕鬆地將資料併入現有資料表。

以下是使用 RedshiftCopyActivity 的使用案例概觀：

1. 首先使 AWS Data Pipeline 用在 Amazon S3 中暫存您的資料。
2. 用RedshiftCopyActivity於將數據從 Amazon RDS 和 Amazon 移動EMR到 Amazon Redshift。

   這可讓您將資料載入 Amazon Redshift，並在此進行分析。
3. 用SqlActivity於對已載入 Amazon Redshift 的資料執行SQL查詢。

此外，RedshiftCopyActivity 可讓您使用 S3DataNode，因為它支援資訊清單檔案。如需詳細資訊，請參閱S3 DataNode。

### 範例

以下為此物件類型的範例。

為了確保格式轉換，此範例在中使用EMPTYASNULL和IGNOREBLANKLINES特殊轉換參數commandOptions。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的資料轉換參數。

```
{
  "id" : "S3ToRedshiftCopyActivity",
  "type" : "RedshiftCopyActivity",
  "input" : { "ref": "MyS3DataNode" },
  "output" : { "ref": "MyRedshiftDataNode" },
  "insertMode" : "KEEP_EXISTING",
  "schedule" : { "ref": "Hour" },
  "runsOn" : { "ref": "MyEc2Resource" },
  "commandOptions": ["EMPTYASNULL", "IGNOREBLANKLINES"]
```

```
}
```

以下範例管道定義會顯示使用 APPEND 插入模式的活動：

```
{
  "objects": [
    {
      "id": "CSVId1",
      "name": "DefaultCSV1",
      "type": "CSV"
    },
    {
      "id": "RedshiftDatabaseId1",
      "databaseName": "dbname",
      "username": "user",
      "name": "DefaultRedshiftDatabase1",
      "*password": "password",
      "type": "RedshiftDatabase",
      "clusterId": "redshiftclusterId"
    },
    {
      "id": "Default",
      "scheduleType": "timeseries",
      "failureAndRerunMode": "CASCADE",
      "name": "Default",
      "role": "DataPipelineDefaultRole",
      "resourceRole": "DataPipelineDefaultResourceRole"
    },
    {
      "id": "RedshiftDataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "tableName": "orders",
      "name": "DefaultRedshiftDataNode1",
      "createTableSql": "create table StructuredLogs (requestBeginTime CHAR(30)
PRIMARY KEY DISTKEY SORTKEY, requestEndTime CHAR(30), hostname CHAR(100), requestDate
varchar(20));",
      "type": "RedshiftDataNode",
      "database": {
        "ref": "RedshiftDatabaseId1"
      }
    },
```

```
    {
      "id": "Ec2ResourceId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "securityGroups": "MySecurityGroup",
      "name": "DefaultEc2Resource1",
      "role": "DataPipelineDefaultRole",
      "logUri": "s3://myLogs",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "type": "Ec2Resource"
    },
    {
      "id": "ScheduleId1",
      "startDateTime": "yyyy-mm-ddT00:00:00",
      "name": "DefaultSchedule1",
      "type": "Schedule",
      "period": "period",
      "endDateTime": "yyyy-mm-ddT00:00:00"
    },
    {
      "id": "S3DataNodeId1",
      "schedule": {
        "ref": "ScheduleId1"
      },
      "filePath": "s3://datapipeline-us-east-1/samples/hive-ads-samples.csv",
      "name": "DefaultS3DataNode1",
      "dataFormat": {
        "ref": "CSVId1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "RedshiftCopyActivityId1",
      "input": {
        "ref": "S3DataNodeId1"
      },
      "schedule": {
        "ref": "ScheduleId1"
      },
      "insertMode": "APPEND",
      "name": "DefaultRedshiftCopyActivity1",
      "runsOn": {
        "ref": "Ec2ResourceId1"
```

```
      },
      "type": "RedshiftCopyActivity",
      "output": {
        "ref": "RedshiftDataNodeId1"
      }
    }
  ]
}
```

APPEND 操作會將項目新增到資料表，無論其主索引鍵或排序索引鍵為何。例如，若您有以下資料表，您可以使用相同的 ID 和使用者值附加記錄。

```
ID(PK)     USER
1          aaa
2          bbb
```

您可以使用相同的 ID 和使用者值附加記錄：

```
ID(PK)     USER
1          aaa
2          bbb
1          aaa
```

> **Note**
>
> 若 APPEND 操作遭到插斷並進行重試，其導致的重新執行管道可能會從開頭附加。這可能會造成進一步的重複，因此建議您留意此行為，特別是在您擁有任何計算資料列數量的邏輯時。

如需教學，請參閱使用將數據複製到亞馬遜紅移 AWS Data Pipeline。

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| insertMode | 決定如 AWS Data Pipeline 何處理目標資料表中與要載入之資料列重疊的預先存在資料。<br><br>有效值為：KEEP_EXISTING 、OVERWRITE_EXISTING 、TRUNCATE 和 APPEND。 | 列舉 |

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | KEEP_EXISTING 會將新列新增至資料表，並保持任何現有列不變。<br><br>KEEP_EXISTING 和 OVERWRITE_EXISTING 使用主索引鍵、排序索引鍵和分發索引鍵，以識別出哪些傳入的列與現有列匹配。請參閱 Amazon Redshift 資料庫開發人員指南中的更新和插入新資料。<br><br>TRUNCATE 會刪除目的地資料表中的所有資料，再寫入新資料。<br><br>APPEND 會將所有記錄新增至 Redshift 資料表結尾。 APPEND 不需要主索引鍵、分發索引鍵或排序索引鍵，因此可能會附加可能重複的項目。 | |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。<br><br>指定其他物件的排程參考，以設定此物件的相依性執行順序。<br><br>在大部分的情況下，建議將排程參考放在預設的管道物件，讓所有物件都繼承該排程。例如，您可以指定 "schedule": {"ref": "DefaultSchedule"} 在物件上明確設定排程。<br><br>如果管道中的主排程包含巢狀排程，請建立具有排程參考的父物件。<br><br>如需範例選用排程組態的詳細資訊，請參閱排程。 | 參考物件，例如： "schedule":{"ref": "myScheduleId"} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供一個runsOn值並且workerGroup 存在，workerGroup則會忽略。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| commandOptions | 在COPY作業期間將參數傳遞至 Amazon Redshift 資料節點。如需參數的相關資訊，請參閱 Amazon Redshift 資料庫開發人員指南COPY中的。<br><br>COPY 載入資料表時，會嘗試隱含地將來源資料中的字串轉換為目標欄的資料類型。除了自動進行的預設資料轉換之外，如果您收到錯誤或有其他轉換需求，您也可以指定其他轉換參數。如需詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的資料轉換參數。<br><br>如果資料格式與輸入或輸出資料節點相關聯，則會忽略提供的參數。<br><br>由於複製操作會先使用 COPY 將資料插入臨時資料表中，然後使用 INSERT 命令將資料從臨時資料表複製到目的地資料表，因此某些 COPY 參數 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | 會不適用，例如 COPY 命令啟用資料表自動壓縮的功能。如果壓縮為必要，請將欄編碼詳細資訊新增至 CREATE TABLE 陳述式。<br><br>此外，在某些情況下，當它需要從 Amazon Redshift 叢集卸載資料並在 Amazon S3 中建立檔案時，需要RedshiftCopyActivity 仰賴 Amazon Redshift 的UNLOAD操作。<br><br>若要在複製和卸載期間改善效能，請從 UNLOAD 命令指定 PARALLEL OFF 參數。如需參數的相關資訊，請參閱 Amazon Redshift 資料庫開發人員指南UNLOAD中的。 | |
| dependsOn | 指定與另一個可執行物件的相依性。 | 參考物件：`"dependsOn":{"ref":"myActivityId"}` |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| input | 輸入資料節點。資料來源可以是 Amazon S3、DynamoDB 或 Amazon Redshift。 | 參考物件：`"input":{"ref":"myDataNodeId"}` |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件：`"onFail":{"ref":"myActionId"}` |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 參考物件：`"onLateAction":{"ref":"myActionId"}` |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件：`"onSuccess":{"ref":"myActionId"}` |
| output | 輸出資料節點。輸出位置可以是 Amazon S3 或 Amazon Redshift。 | 參考物件：`"output":{"ref":"myDataNodeId"}` |
| parent | 目前物件的父系，其插槽會被繼承。 | 參考物件：`"parent":{"ref":"myBaseObjectId"}` |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 參考物件：`"precondition":{"ref":"myPreconditionId"}` |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 佇列 | 對應於 Amazon Redshift 中的query_group設定，可讓您根據並行活動在佇列中的位置指派並排定其優先順序。<br><br>Amazon Redshift 會將同時連線數限制在 15。如需詳細資訊，請參閱 Amazon 資RDS料庫開發人員指南中的將查詢指派給佇列。 | 字串 |
| reportProgressTimeout | 遠端工作連續呼叫 reportProgress 的逾時。<br><br>如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scheduleType | 允許您指定管道中物件的排程。值為：cron、ondemand 和 timeseries 。<br><br>timeseries 排程表示執行個體會排程在每個間隔的結尾。<br><br>Cron 排程表示執行個體會排程在每個間隔的開頭。<br><br>ondemand 排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。<br><br>若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。<br><br>若您使用 ondemand 排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 scheduleType 。 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| transformSql | 用於轉換輸入資料的 SQL SELECT 表達式。<br><br>在資料表執行名為 staging 的 transformSql 表達式。<br><br>當您從 DynamoDB 或 Amazon S3 複製資料時，AWS Data Pipeline 會建立一個名為「暫存」的表格，然後在該表格中載入資料。此資料表中的資料用於更新目標資料表。<br><br>transformSql 的輸出結構描述，必須與最終目標表格的結構描述相符。<br><br>如果您指定選transformSql 項，則會從指定的陳述式建立第二個臨時資料SQL表。然後，第二個臨時資料表中的資料會更新於最終目標資料表。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件："activeInstances": {"ref":"myRunnableObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件："cascadeF |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | ailedOn": {"ref":"myRunnableObjectId"} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件：<br>`"waitingOn": {"ref":"myRunnableObjectId"}` |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件的球體。代表其在生命週期中的位置。例如，元件物件會引發執行個體物件，該物件會執行嘗試物件。 | 字串 |

## ShellCommandActivity

執行命令或指令碼。您可以使用 `ShellCommandActivity` 來執行時間序列或與 Cron 相似的排程任務。

當`stage`欄位設定為 true 並與一起使用時`S3DataNode`，`ShellCommandActivity`支援暫存資料的概念，這表示您可以將資料從 Amazon S3 移至階段位置 (例如 Amazon EC2 或本機環境)，使用指令碼和對資料執行工作`ShellCommandActivity`，然後將其移回 Amazon S3。

在這種情況下，當您的殼層命令連線到輸入 `S3DataNode` 時，您的殼層指令碼會使用 `${INPUT1_STAGING_DIR}`、`${INPUT2_STAGING_DIR}` 及其他欄位 (指向 `ShellCommandActivity` 輸入欄位) 在資料上直接運作。

同樣地，shell 命令的輸出可以暫存在輸出目錄中，以便自動推送到 Amazon S3，由${OUTPUT1_STAGING_DIR}${OUTPUT2_STAGING_DIR}、參照等。

這些表達式可做為命令列引數傳遞到殼層命令，讓您在資料轉換邏輯中使用。

ShellCommandActivity 會傳回 Linux 形式的錯誤代碼及字串。若 ShellCommandActivity 導致錯誤，傳回的 error 會是非零的值。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "command" : "mkdir new-directory"
}
```

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在 schedule 間隔的執行期間會呼叫此物件。<br><br>若要設定此物件的相依性執行順序，請指定另一個物件的 schedule 參考。<br><br>若要滿足這項需求，請明確設定物件的 schedule，例如指定 "schedule": {"ref": "DefaultSchedule"} 。<br><br>在大部分的情況下，建議您將 schedule 參考放在預設的管道物件，讓所有物件都繼承該排程。如果管道由排程的樹狀目錄 (主排程內還有排程) 組成，您可以建立含排程參考的父物件。<br><br>若要分攤負載，請稍微提前 AWS Data Pipeline 建立實體物件，但會依排程執行它們。<br><br>如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/l | 參考對象，例如「時間表」：{「ref」：」mySche duleId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | atest/DeveloperGuide/dp-object-schedule.html。 | |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| command | 要執行的命令。使用 $ 參考位置參數，並使用 scriptArgument 指定命令的參數。此值和任何相關聯的參數，都必須在您執行任務執行器的環境中執行。 | 字串 |
| scriptUri | 檔案的 Amazon S3 URI 路徑，以下載並以殼層命令的形式執行。僅指定一個 scriptUri 或 command 欄位。scriptUri 無法使用參數，請改為使用 command。 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 執行活動或命令的運算資源，例如 Amazon 執行個EC2體或 Amazon EMR 叢集。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| workerGroup | 用於路由任務。如果您提供 runsOn 值，且 workerGroup 存在，則會忽略 workerGroup 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則未在指定開始時間內完成的遠端活動，可能會重試。 | 期間 |
| dependsOn | 指定與其他可執行物件的相依性。 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| input | 輸入資料的位置。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或尚未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myA ctionId「} |
| output | 輸出資料的位置。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeID」} |
| parent | 目前物件的父系，其槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| pipelineLogUri | Amazon S3URI，例如`'s3://BucketName/Key/'` 用於上傳管道的日誌。 | 字串 |
| precondition | 可選擇性定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPre conditionId「} |
| reportProgressTime out | 遠端活動連續呼叫 `reportProgress` 的逾時。如果設定，則系統可能會將未回報指定時段進度的遠端活動視為已停滯並重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| scheduleType | 可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。<br><br>值為：cron、ondemand 和 timeseries 。<br><br>如果設為 timeseries ，則執行個體會排程在每個間隔的結尾。<br><br>如果設為 Cron，則執行個體會排程在每個間隔的開頭。<br><br>如果設為 ondemand，您可以每次啟用執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。若您使用 ondemand 排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 scheduleType 。若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。 | 列舉 |
| scriptArgument | 要傳遞給指令所指定之命令的字串JSON格式化陣列。例如，如果命令為 echo $1 $2，請將 scriptArgument 指定為 "param1", "param2"。針對多個引數和參數，請依照下列所示來傳遞 scriptArgument ：<br>"scriptArgument":"arg1","scriptArgument":"param1","scriptArgument":"arg2","scriptArgument":"param2" 。scriptArgument 只能與 command 一起使用；與 scriptUri 一起使用會造成錯誤。 | 字串 |
| stage | 決定是否啟用臨時功能，並讓您的 shell 命令存取臨時資料變數，例如 ${INPUT1_STAGING_DIR} 和 ${OUTPUT1_STAGING_DIR} 。 | Boolean |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| stderr | 路徑，可接收來自命令的重新導向系統錯誤訊息。如果您使用此runs0n欄位，這必須是 Amazon S3 路徑，因為執行活動的資源具有暫時性質。不過，如果您指定 workerGroup 欄位，則允許使用本機檔案路徑。 | 字串 |
| stdout | 從命令接收重新導向輸出的 Amazon S3 路徑。如果您使用此runs0n欄位，這必須是 Amazon S3 路徑，因為執行活動的資源具有暫時性質。不過，如果您指定 workerGroup 欄位，則允許使用本機檔案路徑。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | 若此物件已取消，會提供 cancellat ionReason 。 | 字串 |
| @cascadeFailedOn | 物件失敗所在之相依鏈的描述。 | 引用對象，例如 cascadeFa iledOn「「：{" ref」：」myR unnableObject ID "} |
| emrStepLog | Amazon EMR 步驟日誌僅適用於 Amazon EMR 活動嘗試。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| errorId | 若此物件失敗，會提供 errorId。 | 字串 |
| errorMessage | 若此物件失敗，會提供 errorMessage 。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 任務日誌可用於EMR基於 Amazon 的活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 物件的狀態。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 用來建立物件的 AWS Data Pipeline 版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件在生命週期中的位置。元件物件引發執行個體物件，該物件會執行嘗試物件。 | 字串 |

### 另請參閱

- CopyActivity
- EmrActivity

## SqlActivity

在資料庫上執行SQL查詢 (指令碼)。

### 範例

以下為此物件類型的範例。

```
{
  "id" : "MySqlActivity",
  "type" : "SqlActivity",
  "database" : { "ref": "MyDatabaseID" },
  "script" : "SQLQuery" | "scriptUri" : s3://scriptBucket/query.sql,
  "schedule" : { "ref": "MyScheduleID" },
```

```
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| database | 要在其上執行提供的SQL指令碼的資料庫。 | 引用對象，例如「數據庫」：{「ref」：」myDatabaseId「} |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。您必須指定另一個物件的排程參考，設定此物件的依存項目執行順序。您可以在物件上明確設定排程，例如指定 `"schedule": {"ref": "DefaultSchedule"}` 。<br><br>在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。<br><br>若管道具有與主排程呈現巢狀結構的排程樹狀目錄，請建立具有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考對象，例如「時間表」：{「ref」：」myScheduleId「} |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| script | 要執行的SQL指令碼。您必須指定指令碼或scriptUri。當指令碼存放在 Amazon S3 時，不會將指令碼評估為運算式。當指令碼存放在 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| | Amazon S3 時，指定的多個值會很有幫助。scriptArgument | |
| scriptUri | URI指定要在此活動中執行之指SQL令碼的位置。 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 執行活動或命令的可運算資源。例如，Amazon EC2 實例或 Amazon EMR 集群。 | 引用對象，例如 runsOn「「：{" ref」：」myResourceId「} |
| workerGroup | 工作者群組。這是用於路由任務。如果您提供 `runsOn` 值，且 `workerGroup` 存在，則會忽略 `workerGroup` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| dependsOn | 指定與另一個可執行物件的相依性。 | 引用對象，例如 dependsOn「「：{" ref」：」myActivityId「} |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| input | 輸入資料的位置。 | 引用對象，例如「輸入」：{「參考」：「myDataNodeID」} |
| lateAfterTimeout | 管道排程啟動以來的時間期間，物件執行必須在此期間內啟動。 | 期間 |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 如果物件尚未排定或仍未在管線的排定開始時段內完成，則應觸發的動作 (如 'lateAfterTimeout' 所指定)。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| output | 輸出資料的位置。這僅適用於從腳本中引用（例如`#{output.tablename}`）以及通過在輸出數據節點中設置 createTableSql " 來創建輸出表。SQL查詢的輸出不會寫入輸出資料節點。 | 引用對象，例如「輸出」：{「ref」：「myDataNodeID」} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| pipelineLogUri | 用於上傳管道日誌的 S3URI（例如 's3：//BucketName/鍵/'）。 | 字串 |
| precondition | 選擇是否定義先決條件。在符合所有先決條件之前，資料節點不會標記 READY ""。 | 引用對象，例如「前提條件」：{「ref」：」myPreconditionId「} |
| 佇列 | [僅限 Amazon Redshift] 對應到 Amazon Redshift 中的 query_group 設定，允許您根據活動在佇列中的位置，指派及優先處理同時進行的活動。Amazon Redshift 會將同時連線數限制在 15。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的將查詢指派給佇列。 | 字串 |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。值為：cron、ondemand 和 timeseries 。<br><br>timeseries 排程表示執行個體會排程在每個間隔的結尾。<br><br>cron 排程表示執行個體會排程在每個間隔的開頭。<br><br>ondemand 排程可讓您在每次啟用時執行一次管道。這表示您不必複製或重新建立管道，然後再執行一次。若您使用 ondemand 排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 scheduleType 。若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。 | 列舉 |
| scriptArgument | 指令碼的變數清單。您也可以改為將表達式直接置放在指令碼欄位中。當指令碼存放在 Amazon S3 中時，的多個值會很有幫助。 scriptArgument 例如：# {格式（@scheduledStartTime，「Y-MM-DD HH：毫米：SS」}\ n # {格式 plusPeriod（@scheduledStartTime，「1 天」），「YY-MM-DD HH：毫米：SS」} | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myRunnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「： {" ref」 ：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：C omponent 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

# 資源

以下是資 AWS Data Pipeline 源物件：

物件

- Ec2Resource

- EmrCluster
- HttpProxy

## Ec2Resource

執行管道活動定義之工作的 Amazon EC2 執行個體。

AWS Data Pipeline 現在支援 Amazon 執IMDSv2行個EC2體，Amazon 執行個體使用工作階段導向方法，在從執行個體擷取中繼資料資訊時，更有效地處理身份驗 工作階段會開始和結束 Amazon 執行個體上執行的軟體用來存取本機儲存的 Amazon EC2 執行個體中繼資料和登入資料的一系列請求。EC2軟體會以簡單的HTTPPUT要求啟動工作階段IMDSv2。IMDSv2會將秘密權杖傳回至 Amazon EC2 執行個體上執行的軟體，這會使用該權杖做為密碼，向中繼資料和登入IMDSv2資料發出要求。

> **Note**
>
> 若要用IMDSv2於 Amazon EC2 執行個體，您需要修改設定，因AMI為預設值不相容IMDSv2。您可以指定可透過下列SSM參數擷取的新AMI版本：`/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs`。

如需在未指定EC2執行個體時建 AWS Data Pipeline 立的預設 Amazon 執行個體的相關資訊，請參閱 AWS 區域的預設 Amazon EC2 執行個體。

### 範例

EC2-經典

> **Important**
>
> 只有在 2013 年 12 月 4 日之前建立的 AWS 帳戶才支援 EC2-Classic 平台。如果您有其中一個帳戶，您可以選擇在 EC2-Classic 網路中為管線建立EC2Resource物件，而不是建立VPC。強烈建議您為中的所有管道建立資源VPCs。此外，如果您在 EC2-Classic 中有現有資源，我們建議您將它們遷移到VPC.

下列範例物件會將EC2執行個體啟動至 EC2-Classic，並設定了一些選用欄位。

```
{
  "id" : "MyEC2Resource",
```

```
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

下列範例物件會將EC2執行個體啟動為非預設值VPC，並設定了一些選擇性欄位。

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroupIds" : [
    "sg-12345678",
    "sg-12345678"
  ],
  "subnetId": "subnet-12345678",
  "associatePublicIpAddress": "true",
  "keyPair" : "my-key-pair"
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| resourceRole | 控制 Amazon EC2 執行個體可存取之資源的IAM角色。 | 字串 |
| role | AWS Data Pipeline 用來建立EC2執行個體的IAM角色。 | 字串 |

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。<br><br>若要設定此物件的相依性執行順序，請指定另一個物件的排程參考。您可採用下列其中一種方式來這麼做：<br><br>• 為確保管道中所有的物件沿用排程，請明確設定物件的排程：`"schedule": {"ref": "DefaultSchedule"}` 。在大部分的情況下，將排程參考放在預設的管道物件，讓所有物件都繼承該排程是很有用的。<br>• 如果管道有排程套疊在主排程內，您可以建立有排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考物件，例如 `"schedule":{"ref": "myScheduleId"}` |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| actionOnResource失敗 | 此資源的資源故障之後所採取的動作。有效值為 `"retryall"` 和 `"retrynone"` 。 | 字串 |
| actionOnTask失敗 | 此資源的任務失敗之後所採取的動作。有效值為 `"continue"` 或 `"terminate"` 。 | 字串 |
| associatePublicIp地址 | 指出是否將公有 IP 地址指派此執行個體。如果執行個體位於 Amazon EC2 或 AmazonVPC，則預設值為`true`。否則，預設值為 `false`。 | Boolean |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則未在指定開始時間內完成的遠端活動，可能會重試。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| availabilityZone | 要在其中啟動 Amazon EC2 執行個體的可用區域。 | 字串 |
| disableIMDSv1 | 預設值為 false，並同時啟用IMDSv1和IMDSv2。如果你將其設置為 true，那麼它禁用IMDSv1，只提供 IMDSv2s | Boolean |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| httpProxy | 用戶端用來連線至 AWS 服務的 Proxy 主機。 | 參考物件，例如 `"httpProxy": {"ref":"myHttpProxyId"}` |
| imageId | AMI要用於執行個體的識別碼。依預設， AWS Data Pipeline 會使用HVMAMI虛擬化類型。AMI IDs使用的具體基於一個區域。您可以指定您選擇HVMAMI的AMI來覆寫預設值。如需有關AMI類型的詳細資訊，請參閱 Amazon EC2 使用者指南AMI中的 Linux AMI 虛擬化類型和尋找 Linux。 | 字串 |
| initTimeout | 等候資源啟動的時間長短。 | 期間 |
| instanceCount | 已廢除。 | Integer |
| instanceType | 要啟動的 Amazon EC2 實例的類型。 | 字串 |
| keyPair | 金鑰對的名稱。如果您在未指定 key pair 的情況下啟動 Amazon EC2 執行個體，則無法登入該執行個體。 | 字串 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| minInstanceCount | 已廢除。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件，例如 `"onFail": {"ref":"myActionId"}` |
| onLateAction | 某個物件尚未排程或仍在執行時，應該觸發的動作。 | 參考物件，例如 `"onLateAction": {"ref":"myActionId"}` |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件，例如 `"onSuccess": {"ref":"myActionId"}` |
| parent | 目前物件的父系，其插槽已被繼承。 | 參考物件，例如 `"parent": {"ref":"myBaseObjectId"}` |
| pipelineLogUri | 用於上傳管道日誌的 Amazon S3URI（例如`'s3://BucketName/Key/'`）。 | 字串 |
| region | 應在其中執行 Amazon 執行EC2個體的區域代碼。根據預設，執行個體執行所在的區域和管道相同。您可以在和相依資料集相同的區域中執行執行個體。 | 列舉 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| runAsUser | 要執行的使用者 TaskRunner. | 字串 |
| runsOn | 此物件不允許此欄位。 | 參考物件，例如 `"runsOn": {"ref":"myResourceId"}` |
| scheduleType | 排程類型可讓您指定管道定義中的物件應該排程在間隔開頭、間隔結尾，還是隨需排程。<br><br>數值為：<br><br>• `timeseries` 。 執行個體會在每個間隔結束時排程。<br>• `cron`。 執行個體會排定在每個間隔的開始。<br>• `ondemand`。 允許您在每次啟動時執行一次管道。您不必複製或重新建立管道，然後再執行一次。若您使用隨需排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 `scheduleType` 。若要使用隨需管道，請針對每次後續執行呼叫 `ActivatePipeline` 操作。 | 列舉 |
| securityGroupIds | 用於資源集區中執行個體IDs的一或多個 Amazon EC2 安全群組。 | 字串 |
| securityGroups | 要用於資源集區中執行個體的一或多個 Amazon EC2 安全群組。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| spotBidPrice | 您 Spot 執行個體每小時的美元上限，這是介於 0 至 20.00 的獨佔小數值。 | 字串 |
| subnetId | 要在其中啟動執行個體的 Amazon EC2 子網路識別碼。 | 字串 |
| terminateAfter | 在此小時數後終止資源。 | 期間 |
| useOnDemandOnLastAttempt | 最後一次嘗試請求 Spot 執行個體時，提出隨需執行個體請求，而不是 Spot 執行個體請求。這可確保即使之前所有的嘗試都失敗，最後一次嘗試也不會中斷。 | Boolean |
| workerGroup | 此物件不允許此欄位。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如 `"activeInstances":{"ref":"myRunnableObjectId"}` |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | 若此物件已取消，會提供 `cancellationReason` 。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 參考物件，例如 `"cascadeFailedOn":{"ref":"m` |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | yRunnable ObjectId"} |
| emrStepLog | 步驟日誌僅適用於 Amazon EMR 活動嘗試。 | 字串 |
| errorId | 若此物件失敗，會提供錯誤 ID。 | 字串 |
| errorMessage | 若此物件失敗，會提供錯誤訊息。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @failureReason | 資源故障的原因。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 任務日誌可在嘗試 Amazon EMR 活動時使用。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如 `"waitingOn": {"ref":"myRunnableObjectId"}` |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件在生命週期中的位置。元件物件引發執行個體物件，這會執行嘗試物件。 | 字串 |

## EmrCluster

表示 Amazon EMR 叢集的組態。此物件是由EmrActivity和用HadoopActivity來啟動叢集。

目錄

- 另請參閱

## 排程器

排程器可提供在 Hadoop 叢集內指定資源配置和任務優先順序的方式。管理員或使用者可以為各種類別的使用者和應用程式選擇排程器。排程器可使用佇列，將資源配置給使用者和應用程式。您可以在建立叢集時設定這些佇列。您接著可以設定特定類型工作和使用者的優先順序。這可以讓您有效率地使用叢集資源，允許超過一名使用者將工作提交至叢集。有三種可用的排程器類型：

- FairScheduler— 嘗試在大量時間內平均排程資源。
- CapacityScheduler— 使用佇列允許叢集管理員將使用者指派至不同優先順序和資源配置的佇列。
- 預設 — 由叢集使用，可由您的場地設定。

## Amazon EMR 發布版本

Amazon EMR 版本是來自大數據生態系統的一組開放原始碼應用程式。每個版本都包含不同的大數據應用程式、元件和功能，您可以在建立叢集時選擇讓 Amazon EMR 安裝和設定這些功能。請使用版本標籤指定發行版本。發行標籤的格式應為 `emr-x.x.x`。例如：`emr-5.30.0`。Amazon EMR 叢集以發行標籤為基礎，`emr-4.0.0`並在稍後使用該`releaseLabel`屬性來指定`EmrCluster`物件的發行標籤。早期版本使用此 `amiVersion` 屬性。

> **Important**
>
> 使用 5.22.0 版或更新版本建立的所有 Amazon EMR 叢集都使用簽名版本 4 來驗證傳送給 Amazon S3 的請求。某些早期版本會使用簽章版本 2。簽章版本 2 支援將不再提供。如需詳細資訊，請參閱 Amazon S3 更新 — Sigv2 棄用期間延長和修改。強烈建議您使用支援簽名EMR版本 4 的 Amazon 發行版本。對於從 EMR 4.7.x 開始的早期版本，該系列中最新發行的版本已更新為支援「簽名版本 4」。使用舊EMR版本時，建議您使用該系列中的最新版本。此外，請避免使用早於 EMR 4.7.0 的發行版本。

### 考量事項與限制

#### 使用最新版本的工作執行器

如果您正在使用具有發行版本標籤的自我管理`EmrCluster`物件，請使用最新的工作執行器。如需 Task Runner 的詳細資訊，請參閱使用工作執行器。您可以為所有 Amazon EMR 組態分類設

定屬性值。如需詳細資訊，請參閱 Amazon EMR 版本指南中的設定應用程式the section called "EmrConfiguration"、和the section called "屬性"物件參考。

Support IMDSv2

稍早版本僅 AWS Data Pipeline 支援IMDSv1。現在，IMDSv2在 Amazon EMR 5.23.1，5.27.1 和 5.32 或更高版本以及 Amazon EMR 6.2 或更高版本中 AWS Data Pipeline 支持。IMDSv2從執行個體擷取中繼資料資訊時，會使用工作階段導向方法來更妥善處理驗證。您應該使用 TaskRunner -2.0 建立使用者管理的資源，將執行個體設定為IMDSv2撥打呼叫。

Amazon EMR 5.32 或更高版本和 Amazon 6.x EMR

Amazon EMR 5.32 或更高版本和 6.x 發布系列使用 Hadoop 版本 3.x，這引入了與 Hadoop 版本 2.x 相比，如何評估 Hadoop 的類路徑的突破性變化。像 Joda-Time 這樣的常見庫已從類路徑中刪除。

如果EmrActivity或HadoopActivity執行在 Hadoop 3.x 中移除的程式庫具有相依性的 Jar 檔案，則步驟會失敗並顯示錯誤`java.lang.NoClassDefFoundError`或。`java.lang.ClassNotFoundException`對於使用 Amazon EMR 5.x 發行版本沒有問題的 Jar 檔案，可能會發生這種情況。

若要修正此問題，您必須先將 Jar 檔案相依性複製到`EmrCluster`物件上的 Hadoop 類別路徑，然後再啟動或. `EmrActivity` `HadoopActivity` 我們提供了一個 bash 腳本來做到這一點。bash 腳本可在以下位置使用，其中 *`MyRegion`* 例如，您的`EmrCluster`對象運行的 AWS 區域`us-west-2`。

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh
```

執行指令碼的方式取決於是否`EmrActivity`在自我管理的資源上`HadoopActivity`執行 AWS Data Pipeline 或是在自我管理的資源上執行。

如果您使用由管理的資源 AWS Data Pipeline，請將一個新增`bootstrapAction`至`EmrCluster`物件。`bootstrapAction`指定要複製為引數的指令碼和 Jar 檔案。每個`EmrCluster`物件最多可以新增 255 個`bootstrapAction`欄位，也可以將`bootstrapAction`欄位新增至已具有啟動程序動作的`EmrCluster`物件。

若要將此指令碼指定為啟動程序動作，請使用下列語法，其中`JarFileRegion`是儲存 Jar 檔案的區域，以及每個 *`MyJarFilen`* 是要複製到 Hadoop 類路徑的 Jar 文件的 Amazon S3 中的絕對路徑。依預設，請勿指定 Hadoop 類別路徑中的 Jar 檔案。

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

下列範例會指定在 Amazon S3 中複製兩個 Jar 檔案的啟動程序動作：my-jar-file.jar和emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar. 範例中使用的區域為 us-west-2。

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m5.xlarge",
  "coreInstanceType" : "m5.xlarge",
  "coreInstanceCount" : "2",
  "taskInstanceType" : "m5.xlarge",
  "taskInstanceCount": "2",
  "bootstrapAction" : ["s3://datapipeline-us-west-2/us-west-2/bootstrap-actions/
latest/TaskRunner/copy-jars-to-hadoop-classpath.sh,us-west-2,s3://path/to/my-jar-
file.jar,s3://dynamodb-dpl-us-west-2/emr-ddb-storage-handler/4.14.0/emr-dynamodb-
tools-4.14.0-jar-with-dependencies.jar"]
}
```

您必須儲存並啟動配管，新的變更bootstrapAction才會生效。

如果您使用自我管理的資源，您可以將指令碼下載到叢集執行個體，然後使用SSH. 該腳本創建一個名為的目錄/etc/hadoop/conf/shellprofile.d和一個datapipeline-jars.sh在該目錄中命名的文件。做為命令列引數提供的 jar 檔案會複製到指令碼建立名稱的目錄中/home/hadoop/datapipeline_jars。如果您的叢集設定不同，請在下載指令碼後適當地修改指令碼。

在命令列上執行指令碼的語法與使用上一個範例中bootstrapAction所示的語法略有不同。在引數之間使用空格而不是逗號，如下列範例所示。

```
./copy-jars-to-hadoop-classpath.sh us-west-2 s3://path/to/my-jar-file.jar s3://
dynamodb-dpl-us-west-2/emr-ddb-storage-handler/4.14.0/emr-dynamodb-tools-4.14.0-jar-
with-dependencies.jar
```

## Amazon EMR 許可

建立自訂IAM角色時，請仔細考慮叢集執行其工作所需的最低權限。請務必授予對所需資源的存取權，例如 Amazon S3 中的檔案或亞馬遜RDS、Amazon Redshift 或 DynamoDB 中的資料。若您希望將 visibleToAllUsers 設為 False，您的角色必須擁有適當的許可來執行此作業。請注意，DataPipelineDefaultRole 沒有這些許可。您必須提供DefaultDataPipelineResourceRole和DataPipelineDefaultRole角色的聯集作為EmrCluster對象角色，或為此目的創建自己的角色。

## 語法

| 物件呼叫欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| schedule | 在排程間隔的執行期間會呼叫此物件。指定其他物件的排程參考，以設定此物件的相依性執行順序。您可以在物件上明確設定排程以滿足這項要求，例如，指定 `"schedule": {"ref": "DefaultSchedule"}` 。在大部分的情況下，建議您將排程參考放在預設的管道物件，讓所有物件都繼承該排程。或者，如果管道有排程的樹狀目錄 (主排程內還有排程)，您可以建立含排程參考的父物件。如需範例選用排程組態的詳細資訊，請參閱https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-object-schedule.html。 | 參考物件，例如 `"schedule":{"ref": "myScheduleId"}` |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| actionOnResource失敗 | 此資源的資源故障之後所採取的動作。有效值為 `"retryall"` (這會在指定的時間內重試叢集所有任務) 和 `"retrynone "`。 | 字串 |
| actionOnTask失敗 | 此資源的任務失敗之後所採取的動作。有效值為 "continue (繼續)" (表示不終止叢集) 和 "terminate (終止)"。 | 字串 |
| additionalMasterSecurityGroupIds | EMR叢集之其他主要安全性群組的識別碼，其格式遵循 sg-XXXX6a 01。如需詳細資訊，請參閱 Amazon EMR管理指南中的 Amazon EMR 其他安全群組。 | 字串 |
| additionalSlaveSecurityGroupIds | EMR叢集之其他從屬安全性群組的識別碼，其格式如下`sg-01XXXX6a` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| amiVersion | Amazon 用於安裝群集節點的 Amazon EMR 機器映像（AMI）版本。如需詳細資訊，請參閱 Amazon EMR 管理指南。 | 字串 |
| 應用程式 | 以逗號分隔引數安裝在叢集中的應用程式。根據預設，會安裝 Hive 和 Pig。此參數僅適用於 Amazon 4.0 及更高EMR版本。 | 字串 |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| availabilityZone | 叢集執行所在的可用區域。 | 字串 |
| bootstrapAction | 當叢集啟動時要執行的動作。您可以指定逗號分隔引數。若要指定上限 255 的多個動作，請新增多個 bootstrapAction 欄位。預設行為是不使用任何引導操作啟動叢集。 | 字串 |
| 組態 | Amazon EMR 群集的配置。此參數僅適用於 Amazon 4.0 及更高EMR版本。 | 參考物件，例如 "configuration":{"ref":"myEmrConfigurationId"} |
| coreInstanceBid價格 | 您願意為 Amazon EC2 執行個體支付的最高 Spot 價格。如果指定了出價，Amazon EMR 會將競價型執行個體用於執行個體群組。在中指定 USD。 | 字串 |
| coreInstanceCount | 用於叢集的核心節點數目。 | Integer |
| coreInstanceType | 用於核心節點的 Amazon EC2 執行個體類型。請參閱 支援 Amazon EMR 叢集的亞馬遜 EC2 執行個體 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| coreGroupConfiguration | Amazon EMR 叢集核心執行個體群組的組態。此參數僅適用於 Amazon 4.0 及更高EMR版本。 | 參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”} |
| coreEbsConfiguration | 將連接至 Amazon EMR 叢集核心群組中每個核心節點的 Amazon EBS 磁碟區組態。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Support EBS 最佳化的執行個體類型。 | 參考物件，例如 “coreEbsConfiguration”: {“ref”: “myEbsConfiguration”} |
| customAmiId | 僅適用於 Amazon EMR 版本 5.7.0 及更高版本。指定 Amazon EMR 佈建 Amazon 執行個EC2體時AMI要使用的自訂 AMI ID。它也可以用來代替引導操作來自定義叢集節點配置。如需詳細資訊，請參閱 Amazon EMR 管理指南中的以下主題。使用自訂 AMI | 字串 |
| EbsBlockDeviceConfig | 與執行個體群組相關聯的要求 Amazon EBS 區塊裝置的組態。包含指定的磁碟區數量，這些磁碟區會與執行個體群組中的每個執行個體產生關聯性。包含 volumesPerInstance 和 volumeSpecification ，其中：<br>• volumesPerInstance 是與執行個體群組中每個執行個體相關聯的特定EBS磁碟區組態的磁碟區數量。<br>• volumeSpecification 是 Amazon EBS 磁碟區規格，例如磁碟區類型IOPS，以及針對連接到 Amazon 叢集中EC2執行個體的磁碟區要求的EBS磁碟區的大小 (以千兆位元組 (GiB) 為單位。EMR | 參考物件，例如 “EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| emrManagedMasterSecurityGroupId | Amazon EMR 叢集之主要安全群組的識別碼，其形式遵循sg-01XXXX6a ．如需詳細資訊，請參閱 Amazon EMR 管理指南中的設定安全群組。 | 字串 |
| emrManagedSlaveSecurityGroupId | Amazon EMR 叢集之從屬安全群組的識別碼，該識別碼遵循下列表單sg-01XXXX6a 。 | 字串 |
| enableDebugging | 啟用 Amazon EMR 叢集上的偵錯功能。 | 字串 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| hadoopSchedulerType | 叢集的排程器類型。有效類型為： PARALLEL_FAIR_SCHEDULING 、 PARALLEL_CAPACITY_SCHEDULING 和 DEFAULT_SCHEDULER 。 | 列舉 |
| httpProxy | 用戶端用來連線至AWS服務的 Proxy 主機。 | 參考物件，例如，"httpProxy" ": {" ref」："myHttpProxyId"} |
| initTimeout | 等候資源啟動的時間長短。 | 期間 |
| keyPair | 用於登入 Amazon EMR 叢集主節點的 Amazon EC2 key pair。 | 字串 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| masterInstanceBid價格 | 您願意為 Amazon EC2 執行個體支付的最高 Spot 價格。介於 0 到 20.00 的小數值 (不含 0 和 20.00)。在中指定USD。設定此值可啟用 Amazon EMR 叢集主節點的競價型執行個體。如果指定了出價，Amazon EMR 會將競價型執行個體用於執行個體群組。 | 字串 |
| masterInstanceType | 要用於主節點的 Amazon EC2 執行個體類型。請參閱 支援 Amazon EMR 叢集的亞馬遜 EC2 執行個體 。 | 字串 |
| masterGroupConfiguration | Amazon EMR 叢集主執行個體群組的組態。此參數僅適用於 Amazon 4.0 及更高EMR版本。 | 參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”} |
| masterEbsConfiguration | 將連接到 Amazon EMR 叢集中主群組中每個主節點的 Amazon EBS 磁碟區組態。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Support EBS 最佳化的執行個體類型。 | 參考物件，例如 “masterEbsConfiguration”: {“ref”: “myEbsConfiguration”} |
| maxActiveInstances | 同時作用中的元件執行個體數目上限。重新執行不計入作用中的執行個體數量。 | Integer |
| maximumRetries | 故障時嘗試重試的次數上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 參考物件，例如 "onFail": {"ref":"myActionId"} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 參考物件，例如 `"onLateAction": {"ref":"myActionId"}` |
| onSuccess | 目前物件成功時要執行的動作。 | 參考物件，例如 `"onSuccess": {"ref":"myActionId"}` |
| parent | 目前物件的父系，其插槽已被繼承。 | 參考物件，例如 `"parent": {"ref":"myBaseObjectId"}` |
| pipelineLogUri | 用於上傳管道日誌的 Amazon S3URI（例如 's3：//BucketName/密鑰/'）。 | 字串 |
| region | Amazon EMR 叢集應執行之區域的程式碼。根據預設，叢集執行所在的區域和管道相同。您可以在和相依資料集相同的區域中執行叢集。 | 列舉 |
| releaseLabel | EMR叢集的釋出標籤。 | 字串 |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| resourceRole | AWS Data Pipeline 用來建立 Amazon EMR 叢集的IAM角色。預設角色為 `DataPipelineDefaultRole` 。 | 字串 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| role | 傳遞給 Amazon EMR 創建EC2節點的IAM角色。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| runsOn | 此物件不允許此欄位。 | 參考物件，例如 "runsOn": {"ref":"myResourceId"} |
| securityConfiguration | 將套用至叢集之EMR安全性組態的識別碼。此參數僅適用於 Amazon 4.8.0 EMR 版及更高版本。 | 字串 |
| serviceAccessSecurityGroupId | Amazon EMR 叢集之服務存取安全群組的識別碼。 | 字串. 它遵循 sg-01XXXX6a 格式，例如 sg-1234abcd 。 |
| scheduleType | 排程類型可讓您指定管道定義的物件應該排程在間隔開頭還是間隔結尾。值為：cron、ondemand 和 timeseries 。timeseries 排程表示執行個體會排程在每個間隔的結尾。cron 排程表示執行個體會排程在每個間隔的開頭。ondemand 排程可讓您在每次啟用時執行一次管道。您不必複製或重新建立管道，然後再執行一次。若您使用 ondemand 排程，則必須在預設物件中指定此排程，且其必須是針對管道中物件指定的唯一 scheduleType 。若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。 | 列舉 |
| subnetId | 要啟動 Amazon EMR 叢集的子網路識別碼。 | 字串 |
| supportedProducts | 在 Amazon EMR 叢集上安裝第三方軟體的參數，例如 Hadoop 的第三方發行版本。 | 字串 |
| taskInstanceBid價格 | 您願意為EC2執行個體支付的最高 Spot 價格。介於 0 到 20.00 的小數值 (不含 0 和 20.00)。在中指定USD。如果指定了出價，Amazon EMR 會將競價型執行個體用於執行個體群組。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| taskInstanceCount | 要用於 Amazon EMR 叢集的任務節點數目。 | Integer |
| taskInstanceType | 用於任務節點的 Amazon EC2 執行個體類型。 | 字串 |
| taskGroupConfiguration | Amazon EMR 叢集任務執行個體群組的組態。此參數僅適用於 Amazon 4.0 及更高EMR版本。 | 參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”} |
| taskEbsConfiguration | 將連接到 Amazon EMR 叢集中任務群組中每個任務節點的 Amazon EBS 磁碟區的組態。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Support EBS 最佳化的執行個體類型。 | 參考物件，例如 “taskEbsConfiguration”: {“ref”: “myEbsConfiguration”} |
| terminateAfter | 在這些小時後終止資源。 | Integer |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| VolumeSpecification | Amazon EBS 磁碟區規格，例如磁碟區類型 IOPS，以及以千兆位元組 (GiB) 為單位的大小，將要求連接到 Amazon 叢集中 Amazon EC2 執行個體的 Amazon EBS 磁碟區。EMR節點可以是核心節點、主節點或任務節點。<br><br>VolumeSpecification 包括：<br><br>• iops() 整數。Amazon EBS 磁碟區支援的每秒 I/O 作業數 (IOPS)，例如 1000。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的EBSI/O 特性。<br>• sizeinGB() 。 整數。Amazon EBS 卷的大小，以吉字節（GiB）為單位，例如 500。如需有關磁碟區類型和硬碟大小的有效組合的資訊，請參閱 Amazon EC2 使用者指南中的EBS磁碟區類型。<br>• volumetType 。 字符串。Amazon EBS 磁碟區類型，例如 gp2。支援的磁碟區類型包括標準、gp2、io1、st1、sc1 和其他。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的EBS磁碟區類型。 | 參考物件，例如 “VolumeSpecification”: {“ref”: “myVolumeSpecification”} |
| useOnDemandOnLastAttempt | 最後一次嘗試請求資源時，提出隨需執行個體請求，而不是 Spot 執行個體請求。這可確保即使之前所有的嘗試都失敗，最後一次嘗試也不會中斷。 | Boolean |
| workerGroup | 此物件不允許此欄位。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如，"activeInstances" ": |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | {" ref」: "myRunnableObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 參考物件，例如，" cascadeFailedOn" ": {" ref」: "myRunnableObjectId"} |
| emrStepLog | 步驟日誌僅適用於 Amazon EMR 活動嘗試。 | 字串 |
| errorId | 若此物件失敗，會提供錯誤 ID。 | 字串 |
| errorMessage | 若此物件失敗，會提供錯誤訊息。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| @failureReason | 資源故障的原因。 | 字串 |
| @finishedTime | 此物件完成其執行的時間。 | DateTime |
| hadoopJobLog | Hadoop 任務日誌可在嘗試 Amazon EMR 活動時使用。 | 字串 |
| @healthStatus | 反映已達終止狀態之最後一個物件執行個體成功或失敗的物件運作狀態。 | 字串 |
| @healthStatusFromInstanceId | 已達終止狀態之最後一個執行個體物件的 ID。 | 字串 |
| @ healthStatusUpdated 時間 | 上次更新運作狀態的時間。 | DateTime |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @lastDeactivatedTime | 此物件最後停用的時間。 | DateTime |
| @ latestCompletedRun 時間 | 執行完成最近一次執行的時間。 | DateTime |
| @latestRunTime | 執行排程最近一次執行的時間。 | DateTime |
| @nextRunTime | 下次要排程執行的時間。 | DateTime |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如，" waitingOn" ": {" ref」："myRunnableObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件在生命週期中的位置。元件物件引發執行個體物件，這會執行嘗試物件。 | 字串 |

## 範例

以下為此物件類型的範例。

目錄



### 啟動 Amazon EMR 群集 hadoopVersion

Example

下列範例會啟動使用 1.0 AMI 版和 Hadoop 0.20 的 Amazon EMR 叢集。

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "hadoopVersion" : "0.20",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m3.xlarge",
  "coreInstanceType" : "m3.xlarge",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m3.xlarge",
  "taskInstanceCount": "10",
  "bootstrapAction" : ["s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop,arg1,arg2,arg3","s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop/configure-other-stuff,arg1,arg2"]
}
```

啟動具有 emr-4.x 或更高版本的版本標籤的 Amazon EMR 叢集

Example

下列範例會使用較新的`releaseLabel`欄位啟動 Amazon EMR 叢集：

```
{
  "id" : "MyEmrCluster",
  "type" : "EmrCluster",
  "keyPair" : "my-key-pair",
  "masterInstanceType" : "m3.xlarge",
  "coreInstanceType" : "m3.xlarge",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m3.xlarge",
  "taskInstanceCount": "10",
  "releaseLabel": "emr-4.1.0",
  "applications": ["spark", "hive", "pig"],
  "configuration": {"ref":"myConfiguration"}
}
```

在您的 Amazon EMR 叢集上安裝其他軟體

Example

`EmrCluster`提供在 Amazon EMR 叢集上安裝第三方軟體的`supportedProducts`欄位，例如，它可讓您安裝 Hadoop 的自訂分發，例如 MapR。它接受要讀取及採取動作的第三方軟體引數逗號分隔清單。以下範例會示範如何使用 `EmrCluster` 的 `supportedProducts` 欄位建立自訂 MapR M3 版本叢集，在其上安裝 Karmasphere Analytics，並在其上執行 `EmrActivity` 物件。

```
{
    "id": "MyEmrActivity",
    "type": "EmrActivity",
    "schedule": {"ref": "ResourcePeriod"},
    "runsOn": {"ref": "MyEmrCluster"},
    "postStepCommand": "echo Ending job >> /mnt/var/log/stepCommand.txt",
    "preStepCommand": "echo Starting job > /mnt/var/log/stepCommand.txt",
    "step": "/home/hadoop/contrib/streaming/hadoop-streaming.jar,-input,s3n://
elasticmapreduce/samples/wordcount/input,-output, \
     hdfs:///output32113/,-mapper,s3n://elasticmapreduce/samples/wordcount/
wordSplitter.py,-reducer,aggregate"
  },
  {
    "id": "MyEmrCluster",
```

```
    "type": "EmrCluster",
    "schedule": {"ref": "ResourcePeriod"},
    "supportedProducts": ["mapr,--edition,m3,--version,1.2,--key1,value1","karmasphere-
enterprise-utility"],
    "masterInstanceType": "m3.xlarge",
    "taskInstanceType": "m3.xlarge"
}
```

停用 3.x 版本的伺服器端加密

Example

通過默認 AWS Data Pipeline 啟用服務器端加密創建的 Hadoop 版本 2.x 的EmrCluster活動。若您想要停用伺服器端加密，您必須在叢集物件定義中指定引導操作。

以下範例會建立停用伺服器端加密的 EmrCluster 活動：

```
{
   "id":"NoSSEEmrCluster",
   "type":"EmrCluster",
   "hadoopVersion":"2.x",
   "keyPair":"my-key-pair",
   "masterInstanceType":"m3.xlarge",
   "coreInstanceType":"m3.large",
   "coreInstanceCount":"10",
   "taskInstanceType":"m3.large",
   "taskInstanceCount":"10",
   "bootstrapAction":["s3://Region.elasticmapreduce/bootstrap-actions/configure-
hadoop,-e, fs.s3.enableServerSideEncryption=false"]
}
```

停用 4.x 版本的伺服器端加密

Example

您必須使用 EmrConfiguration 物件停用伺服器端加密。

以下範例會建立停用伺服器端加密的 EmrCluster 活動：

```
   {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "myResourceId",
```

```
      "type": "EmrCluster",
      "configuration": {
        "ref": "disableSSE"
      }
    },
    {
      "name": "disableSSE",
      "id": "disableSSE",
      "type": "EmrConfiguration",
      "classification": "emrfs-site",
      "property": [{
        "ref": "enableServerSideEncryption"
      }
      ]
    },
    {
      "name": "enableServerSideEncryption",
      "id": "enableServerSideEncryption",
      "type": "Property",
      "key": "fs.s3.enableServerSideEncryption",
      "value": "false"
    }
```

配置 Hadoop KMS ACLs 並在中創建加密區域 HDFS

Example

下列物件會ACLs針對 Hadoop 建立，KMS並在HDFS中建立加密區域和對應的加密金鑰：

```
{
      "name": "kmsAcls",
      "id": "kmsAcls",
      "type": "EmrConfiguration",
      "classification": "hadoop-kms-acls",
      "property": [
        {"ref":"kmsBlacklist"},
        {"ref":"kmsAcl"}
      ]
    },
    {
      "name": "hdfsEncryptionZone",
      "id": "hdfsEncryptionZone",
      "type": "EmrConfiguration",
      "classification": "hdfs-encryption-zones",
```

```
      "property": [
        {"ref":"hdfsPath1"},
        {"ref":"hdfsPath2"}
      ]
    },
    {
      "name": "kmsBlacklist",
      "id": "kmsBlacklist",
      "type": "Property",
      "key": "hadoop.kms.blacklist.CREATE",
      "value": "foo,myBannedUser"
    },
    {
      "name": "kmsAcl",
      "id": "kmsAcl",
      "type": "Property",
      "key": "hadoop.kms.acl.ROLLOVER",
      "value": "myAllowedUser"
    },
    {
      "name": "hdfsPath1",
      "id": "hdfsPath1",
      "type": "Property",
      "key": "/myHDFSPath1",
      "value": "path1_key"
    },
    {
      "name": "hdfsPath2",
      "id": "hdfsPath2",
      "type": "Property",
      "key": "/myHDFSPath2",
      "value": "path2_key"
    }
```

指定自訂IAM角色

Example

依預設，會 AWS Data Pipeline 傳遞`DataPipelineDefaultRole`為 Amazon EMR 服務角色和 `DataPipelineDefaultResourceRole` Amazon EC2 執行個體設定檔，以代表您建立資源。不過，您可以建立自訂 Amazon EMR 服務角色和自訂執行個體設定檔，然後改用它們。 AWS Data Pipeline 應該有足夠的權限才能使用自訂角色建立叢集，而且您必須新增 AWS Data Pipeline 為受信任的實體。

下列範例物件指定 Amazon EMR 叢集的自訂角色：

```
{
   "id":"MyEmrCluster",
   "type":"EmrCluster",
   "hadoopVersion":"2.x",
   "keyPair":"my-key-pair",
   "masterInstanceType":"m3.xlarge",
   "coreInstanceType":"m3.large",
   "coreInstanceCount":"10",
   "taskInstanceType":"m3.large",
   "taskInstanceCount":"10",
   "role":"emrServiceRole",
   "resourceRole":"emrInstanceProfile"
}
```

將中的 EmrCluster 資源用AWSSDK於 Java

Example

下列範例示範如何使用EmrCluster和EmrActivity建立 Amazon EMR 4.x 叢集，以使用 Java SDK 執行星火步驟：

```
public class dataPipelineEmr4 {

  public static void main(String[] args) {

 AWSCredentials credentials = null;
 credentials = new ProfileCredentialsProvider("/path/to/
AwsCredentials.properties","default").getCredentials();
 DataPipelineClient dp = new DataPipelineClient(credentials);
 CreatePipelineRequest createPipeline = new
 CreatePipelineRequest().withName("EMR4SDK").withUniqueId("unique");
 CreatePipelineResult createPipelineResult = dp.createPipeline(createPipeline);
 String pipelineId = createPipelineResult.getPipelineId();

 PipelineObject emrCluster = new PipelineObject()
     .withName("EmrClusterObj")
     .withId("EmrClusterObj")
     .withFields(
   new Field().withKey("releaseLabel").withStringValue("emr-4.1.0"),
   new Field().withKey("coreInstanceCount").withStringValue("3"),
   new Field().withKey("applications").withStringValue("spark"),
```

```
    new Field().withKey("applications").withStringValue("Presto-Sandbox"),
    new Field().withKey("type").withStringValue("EmrCluster"),
    new Field().withKey("keyPair").withStringValue("myKeyName"),
    new Field().withKey("masterInstanceType").withStringValue("m3.xlarge"),
    new Field().withKey("coreInstanceType").withStringValue("m3.xlarge")
    );

 PipelineObject emrActivity = new PipelineObject()
     .withName("EmrActivityObj")
     .withId("EmrActivityObj")
     .withFields(
    new Field().withKey("step").withStringValue("command-runner.jar,spark-submit,--
executor-memory,1g,--class,org.apache.spark.examples.SparkPi,/usr/lib/spark/lib/spark-
examples.jar,10"),
    new Field().withKey("runsOn").withRefValue("EmrClusterObj"),
    new Field().withKey("type").withStringValue("EmrActivity")
    );

 PipelineObject schedule = new PipelineObject()
     .withName("Every 15 Minutes")
     .withId("DefaultSchedule")
     .withFields(
    new Field().withKey("type").withStringValue("Schedule"),
    new Field().withKey("period").withStringValue("15 Minutes"),
    new Field().withKey("startAt").withStringValue("FIRST_ACTIVATION_DATE_TIME")
    );

 PipelineObject defaultObject = new PipelineObject()
     .withName("Default")
     .withId("Default")
     .withFields(
    new Field().withKey("failureAndRerunMode").withStringValue("CASCADE"),
    new Field().withKey("schedule").withRefValue("DefaultSchedule"),
    new
 Field().withKey("resourceRole").withStringValue("DataPipelineDefaultResourceRole"),
    new Field().withKey("role").withStringValue("DataPipelineDefaultRole"),
    new Field().withKey("pipelineLogUri").withStringValue("s3://myLogUri"),
    new Field().withKey("scheduleType").withStringValue("cron")
    );

 List<PipelineObject> pipelineObjects = new ArrayList<PipelineObject>();

 pipelineObjects.add(emrActivity);
 pipelineObjects.add(emrCluster);
```

```
 pipelineObjects.add(defaultObject);
 pipelineObjects.add(schedule);

 PutPipelineDefinitionRequest putPipelineDefintion = new PutPipelineDefinitionRequest()
     .withPipelineId(pipelineId)
     .withPipelineObjects(pipelineObjects);

 PutPipelineDefinitionResult putPipelineResult =
 dp.putPipelineDefinition(putPipelineDefintion);
 System.out.println(putPipelineResult);

 ActivatePipelineRequest activatePipelineReq = new ActivatePipelineRequest()
     .withPipelineId(pipelineId);
 ActivatePipelineResult activatePipelineRes = dp.activatePipeline(activatePipelineReq);

      System.out.println(activatePipelineRes);
      System.out.println(pipelineId);

    }

}
```

在私有子網路中設定 Amazon EMR 叢集

Example

此範例包括將叢集啟動至中私有子網路的配置VPC。如需詳細資訊，請參閱 Amazon EMR管理指南 VPC中的將 Amazon EMR 叢集啟動到一個。此組態為選擇性。您可以在任何使用 EmrCluster 物件的管道中使用它。

若要在私有子網路中啟動 Amazon EMR 叢集 SubnetIdemrManagedMasterSecurityGroupId，請serviceAccessSecurityGroupId在您的EmrCluster組態中指定emrManagedSlaveSecurityGroupId、、和。

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
```

```
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
      "readThroughputPercent": "#{myDDBReadThroughputRatio}",
      "name": "DDBSourceTable",
      "id": "DDBSourceTable",
      "type": "DynamoDBDataNode",
      "tableName": "#{myDDBTableName}"
    },
    {
      "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
      "name": "S3BackupLocation",
      "id": "S3BackupLocation",
      "type": "S3DataNode"
    },
    {
      "name": "EmrClusterForBackup",
      "coreInstanceCount": "1",
      "taskInstanceCount": "1",
      "taskInstanceType": "m4.xlarge",
      "coreInstanceType": "m4.xlarge",
      "releaseLabel": "emr-4.7.0",
      "masterInstanceType": "m4.xlarge",
      "id": "EmrClusterForBackup",
      "subnetId": "#{mySubnetId}",
      "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
      "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
      "serviceAccessSecurityGroupId": "#{myServiceAccessSecurityGroup}",
      "region": "#{myDDBRegion}",
      "type": "EmrCluster",
      "keyPair": "user-key-pair"
    },
    {
      "failureAndRerunMode": "CASCADE",
```

```
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "#{myPipelineLogUri}",
      "scheduleType": "ONDEMAND",
      "name": "Default",
      "id": "Default"
    }
  ],
  "parameters": [
    {
      "description": "Output S3 folder",
      "id": "myOutputS3Loc",
      "type": "AWS::S3::ObjectKey"
    },
    {
      "description": "Source DynamoDB table name",
      "id": "myDDBTableName",
      "type": "String"
    },
    {
      "default": "0.25",
      "watermark": "Enter value between 0.1-1.0",
      "description": "DynamoDB read throughput ratio",
      "id": "myDDBReadThroughputRatio",
      "type": "Double"
    },
    {
      "default": "us-east-1",
      "watermark": "us-east-1",
      "description": "Region of the DynamoDB table",
      "id": "myDDBRegion",
      "type": "String"
    }
  ],
  "values": {
     "myDDBRegion": "us-east-1",
      "myDDBTableName": "ddb_table",
      "myDDBReadThroughputRatio": "0.25",
      "myOutputS3Loc": "s3://s3_path",
      "mySubnetId": "subnet_id",
      "myServiceAccessSecurityGroup":  "service access security group",
      "mySlaveSecurityGroup": "slave security group",
      "myMasterSecurityGroup": "master security group",
      "myPipelineLogUri": "s3://s3_path"
```

```
  }
}
```

將EBS磁碟區附加至叢集節點

Example

您可以將EBS磁碟區附加到管線內EMR叢集中任何類型的節點。若要將EBS磁碟區附加至節點coreEbsConfigurationmasterEbsConfiguration，請TaskEbsConfiguration在您的EmrCluster組態中使用、和。

此 Amazon EMR 叢集範例使用 Amazon EBS 磁碟區做為其主節點、任務和核心節點。有關更多信息，請參閱 Amazon EMR管理指南EMR中的 Amazon EBS 卷。

這些組態都是選擇性的。您可以在任何使用 EmrCluster 物件的管道中使用他們。

在管線中，按一下EmrCluster物件組態，選擇「主要EBS組態」、「核心EBS組態」或「工作EBS組態」，然後輸入類似下列範例的組態詳細資訊。

```
{
  "objects": [
    {
      "output": {
        "ref": "S3BackupLocation"
      },
      "input": {
        "ref": "DDBSourceTable"
      },
      "maximumRetries": "2",
      "name": "TableBackupActivity",
      "step": "s3://dynamodb-emr-#{myDDBRegion}/emr-ddb-storage-handler/2.1.0/emr-
ddb-2.1.0.jar,org.apache.hadoop.dynamodb.tools.DynamoDbExport,#{output.directoryPath},#{input.t
      "id": "TableBackupActivity",
      "runsOn": {
        "ref": "EmrClusterForBackup"
      },
      "type": "EmrActivity",
      "resizeClusterBeforeRunning": "false"
    },
    {
      "readThroughputPercent": "#{myDDBReadThroughputRatio}",
      "name": "DDBSourceTable",
```

```
      "id": "DDBSourceTable",
      "type": "DynamoDBDataNode",
      "tableName": "#{myDDBTableName}"
    },
    {
      "directoryPath": "#{myOutputS3Loc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-
mm-ss')}",
      "name": "S3BackupLocation",
      "id": "S3BackupLocation",
      "type": "S3DataNode"
    },
    {
      "name": "EmrClusterForBackup",
      "coreInstanceCount": "1",
      "taskInstanceCount": "1",
      "taskInstanceType": "m4.xlarge",
      "coreInstanceType": "m4.xlarge",
      "releaseLabel": "emr-4.7.0",
      "masterInstanceType": "m4.xlarge",
      "id": "EmrClusterForBackup",
      "subnetId": "#{mySubnetId}",
      "emrManagedMasterSecurityGroupId": "#{myMasterSecurityGroup}",
      "emrManagedSlaveSecurityGroupId": "#{mySlaveSecurityGroup}",
      "region": "#{myDDBRegion}",
      "type": "EmrCluster",
      "coreEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "masterEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "taskEbsConfiguration": {
        "ref": "EBSConfiguration"
      },
      "keyPair": "user-key-pair"
    },
    {
       "name": "EBSConfiguration",
        "id": "EBSConfiguration",
        "ebsOptimized": "true",
        "ebsBlockDeviceConfig" : [
            { "ref": "EbsBlockDeviceConfig" }
        ],
        "type": "EbsConfiguration"
```

```
    },
    {
        "name": "EbsBlockDeviceConfig",
        "id": "EbsBlockDeviceConfig",
        "type": "EbsBlockDeviceConfig",
        "volumesPerInstance" : "2",
        "volumeSpecification" : {
            "ref": "VolumeSpecification"
        }
    },
    {
      "name": "VolumeSpecification",
      "id": "VolumeSpecification",
      "type": "VolumeSpecification",
      "sizeInGB": "500",
      "volumeType": "io1",
      "iops": "1000"
    },
    {
      "failureAndRerunMode": "CASCADE",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "#{myPipelineLogUri}",
      "scheduleType": "ONDEMAND",
      "name": "Default",
      "id": "Default"
    }
  ],
  "parameters": [
    {
      "description": "Output S3 folder",
      "id": "myOutputS3Loc",
      "type": "AWS::S3::ObjectKey"
    },
    {
      "description": "Source DynamoDB table name",
      "id": "myDDBTableName",
      "type": "String"
    },
    {
      "default": "0.25",
      "watermark": "Enter value between 0.1-1.0",
      "description": "DynamoDB read throughput ratio",
      "id": "myDDBReadThroughputRatio",
```

```
      "type": "Double"
    },
    {
      "default": "us-east-1",
      "watermark": "us-east-1",
      "description": "Region of the DynamoDB table",
      "id": "myDDBRegion",
      "type": "String"
    }
  ],
  "values": {
     "myDDBRegion": "us-east-1",
      "myDDBTableName": "ddb_table",
      "myDDBReadThroughputRatio": "0.25",
      "myOutputS3Loc": "s3://s3_path",
      "mySubnetId": "subnet_id",
      "mySlaveSecurityGroup": "slave security group",
      "myMasterSecurityGroup": "master security group",
      "myPipelineLogUri": "s3://s3_path"
  }
}
```

### 另請參閱

- EmrActivity

## HttpProxy

HttpProxy 允許您配置自己的代理並使任務運行器通過它訪問 AWS Data Pipeline 服務。您不需要使用此資訊設定執行中的 Task Runner。

### HttpProxy 中的範例 TaskRunner

以下管道定義顯示 `HttpProxy` 物件：

```
{
  "objects": [
    {
      "schedule": {
        "ref": "Once"
      },
      "pipelineLogUri": "s3://myDPLogUri/path",
```

```
      "name": "Default",
      "id": "Default"
    },
    {
      "name": "test_proxy",
      "hostname": "hostname",
      "port": "port",
      "username": "username",
      "*password": "password",
      "windowsDomain": "windowsDomain",
      "type": "HttpProxy",
      "id": "test_proxy",
    },
    {
      "name": "ShellCommand",
      "id": "ShellCommand",
      "runsOn": {
        "ref": "Resource"
      },
      "type": "ShellCommandActivity",
      "command": "echo 'hello world' "
    },
    {
      "period": "1 day",
      "startDateTime": "2013-03-09T00:00:00",
      "name": "Once",
      "id": "Once",
      "endDateTime": "2013-03-10T00:00:00",
      "type": "Schedule"
    },
    {
      "role": "dataPipelineRole",
      "httpProxy": {
        "ref": "test_proxy"
      },
      "actionOnResourceFailure": "retrynone",
      "maximumRetries": "0",
      "type": "Ec2Resource",
      "terminateAfter": "10 minutes",
      "resourceRole": "resourceRole",
      "name": "Resource",
      "actionOnTaskFailure": "terminate",
      "securityGroups": "securityGroups",
      "keyPair": "keyPair",
```

```
      "id": "Resource",
      "region": "us-east-1"
    }
  ],
  "parameters": []
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 用戶端將用來連線至AWS服務的 Proxy 主機。 | 字串 |
| port | 用戶端將用來連線至「服AWS務」之 Proxy 主機的連接埠。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| *password | 代理的密碼。 | 字串 |
| S3 NoProxy | 連接到 Amazon S3 時禁用HTTP代理 | Boolean |
| 使用者名稱 | 代理的使用者名稱。 | 字串 |
| windowsDomain | NTLM代理伺服器的視窗網域名稱。 | 字串 |
| windowsWorkgroup | NTLM代理伺服器的 Windows 工作群組名稱。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

# 先決條件

以下是 AWS Data Pipeline 前提條件對象：

物件

- D ynamoDBData 存在
- D ynamoDBTable 存在
- 存在
- S3 KeyExists
- S3 PrefixNotEmpty
- ShellCommandPrecondition

## D ynamoDBData 存在

檢查 DynamoDB 表中是否存在資料的先決條件。

### 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| role | 指定要用來執行先決條件的角色。 | 字串 |
| tableName | 要檢查的 DynamoDB 資料表。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myRunnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myRunnableObject ID "} |
| currentRetryCount | 在這個嘗試中，已嘗試過先決條件的次數。 | 字串 |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| lastRetryTime | 在這個嘗試中，上次嘗試先決條件的時間。 | 字串 |
| 節點 | 即將執行此先決條件的節點 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

# D ynamoDBTable 存在

檢查 DynamoDB 資料表是否存在的先決條件。

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| role | 指定要用來執行先決條件的角色。 | 字串 |
| tableName | 要檢查的 DynamoDB 資料表。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myA ctionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗 | 期間 |
| reportProgressTime out | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在相依性鏈的描述。 | 引用對象，例如 cascadeFa iledOn「「：{" |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | ref」：」myR unnableObject ID "} |
| currentRetryCount | 在這個嘗試中，已嘗試過先決條件的次數。 | 字串 |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| lastRetryTime | 在這個嘗試中，上次嘗試先決條件的時間。 | 字串 |
| 節點 | 即將執行此先決條件的節點 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myR unnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

## 存在

檢查資料節點物件是否存在。

> **Note**
>
> 我們建議您改用系統管理的先決條件。如需詳細資訊，請參閱先決條件。

### 範例

以下為此物件類型的範例。InputData 物件會參考此物件 (Ready)，加上其他您在相同管道定義檔案中定義的物件。CopyPeriod 是 Schedule 物件。

```
{
  "id" : "InputData",
  "type" : "S3DataNode",
  "schedule" : { "ref" : "CopyPeriod" },
  "filePath" : "s3://example-bucket/InputData/#{@scheduledStartTime.format('YYYY-MM-dd-
hh:mm')}.csv",
  "precondition" : { "ref" : "Ready" }
},
{
  "id" : "Ready",
  "type" : "Exists"
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myRunnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 引用對象，例如 cascadeFailedOn「「：{" ref」：」myRunnableObject ID "} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 節點 | 即將執行此先決條件的節點。 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 引用對象，例如 waitingOn「「：{" ref」：」myRunnableObject ID "} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

## 另請參閱

- ShellCommandPrecondition

# S3 KeyExists

檢查 Amazon S3 資料節點中是否存在金鑰。

## 範例

以下為此物件類型的範例。當 s3Key 參數所參考的鍵 (s3://mybucket/mykey) 存在時，便會觸發先決條件。

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://mybucket/mykey"
}
```

您也可以在第二個管道上使用 S3KeyExists 做為先決條件，等待第一個管道完成。若要這麼做：

1. 在第一個管道完成後，將檔案寫入 Amazon S3。
2. 在第二個管道上建立 S3KeyExists 先決條件。

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| role | 指定要用來執行先決條件的角色。 | 字串 |
| s3Key | Amazon S3 密鑰。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 再一次嘗試完成遠端工作之前逾時。如果設定，則系統可能會再次嘗試未在開始之後、設定時間內完成的遠端活動。 | 期間 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為。 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 在故障發生時可啟動的嘗試數量上限。 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗。 | 期間 |
| reportProgressTimeout | 遠端工作連續呼叫 `reportProgress` 的逾時。如果設定，則系統可能會將未回報指定時段進度的遠端活動視為已停滯並重試。 | 期間 |
| retryDelay | 兩次連續嘗試之間的逾時持續時間。 | 期間 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 引用對象，例如 activeInstances「「：{" ref」：」myR unnableObject ID "} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如 cascadeFailedOn "" ": {" ref」: "myRunnab leObjectId"} |
| currentRetryCount | 在這個嘗試中，已嘗試過先決條件的次數。 | 字串 |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| lastRetryTime | 在這個嘗試中，上次嘗試先決條件的時間。 | 字串 |
| 節點 | 即將執行此先決條件的節點 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectI D」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如 waitingOn "" ": {" ref」: "myRunnab leObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：C omponent 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

### 另請參閱

- ShellCommandPrecondition

## S3 PrefixNotEmpty

檢查具有指定前置詞 (以 a 表示URI) 的 Amazon S3 物件是否存在的先決條件。

### 範例

以下是此物件類型的範例，使用必要、選擇性及表達式欄位。

```
{
  "id" : "InputReady",
  "type" : "S3PrefixNotEmpty",
  "role" : "test-role",
  "s3Prefix" : "#{node.filePath}"
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| role | 指定要用來執行先決條件的角色。 | 字串 |
| s3Prefix | 用於檢查對象是否存在的 Amazon S3 前綴。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「： |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | {" ref」：」myA ctionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「： {" ref」：」myA ctionId「} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如 「父」：{「ref」： 「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗 | 期間 |
| reportProgressTime out | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如 activeInstances "" ": {" ref」: "myRunnab leObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如 cascadeFailedOn "" ": |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| | | {" ref」: "myRunnab leObjectId"} |
| currentRetryCount | 在這個嘗試中，已嘗試過先決條件的次數。 | 字串 |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| lastRetryTime | 在這個嘗試中，上次嘗試先決條件的時間。 | 字串 |
| 節點 | 即將執行此先決條件的節點。 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間。 | DateTime |
| @scheduledStartTime | 物件的排程開始時間。 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如 waitingOn "" ": {" ref」: "myRunnab leObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

### 另請參閱

- ShellCommandPrecondition

## ShellCommandPrecondition

可做為先決條件執行的 Unix/Linux 殼層命令。

### 範例

以下為此物件類型的範例。

```
{
  "id" : "VerifyDataReadiness",
  "type" : "ShellCommandPrecondition",
  "command" : "perl check-data-ready.pl"
}
```

### 語法

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| command | 要執行的命令。此值和任何相關聯的參數，都必須在您執行任務執行器的環境中執行。 | 字串 |
| scriptUri | 檔案的 Amazon S3 URI 路徑，以下載並以殼層命令的形式執行。只有一個 scriptUri 或命令欄 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| | 位應該存在。 scriptUri 不能使用參數，請改用命令。 | |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| attemptStatus | 遠端活動最新回報的狀態。 | 字串 |
| attemptTimeout | 遠端工作完成的逾時。如果設定，則系統可能會重試未在設定開始時間內完成的遠端活動。 | 期間 |
| failureAndRerun模式 | 描述相依性故障或重新執行時的消費者節點行為 | 列舉 |
| lateAfterTimeout | 管線開始後，物件必須在其中完成的經過時間。僅當明細表類型未設定為時，才會觸發此選項ondemand。 | 期間 |
| maximumRetries | 故障時嘗試重試的次數上限 | Integer |
| onFail | 目前物件發生故障時要執行的動作。 | 引用對象，例如 onFail「「：{" ref」：」myActionId「} |
| onLateAction | 某個物件尚未排程或仍未完成時，應該觸發的動作。 | 引用對象，例如 onLateAction「「：{" ref」：」myActionId「} |
| onSuccess | 目前物件成功時要執行的動作。 | 引用對象，例如 onSuccess「「：{" ref」：」myActionId「} |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| preconditionTimeout | 自開始起的一段期間，在這段期間之後，如果仍未符合先決條件即會將其標示為失敗 | 期間 |
| reportProgressTimeout | 遠端工作連續呼叫的逾時。reportProgress如果設定，則不回報指定時段進度的遠端活動，可能會視為已停滯而重試。 | 期間 |
| retryDelay | 兩次重試嘗試之間的逾時持續時間。 | 期間 |
| scriptArgument | 要傳遞給 shell 指令碼的引數 | 字串 |
| stderr | 從命令接收重新導向的系統錯誤訊息的 Amazon S3 路徑。如果您使用此`runsOn`欄位，這必須是 Amazon S3 路徑，因為執行活動的資源具有暫時性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。 | 字串 |
| stdout | 從命令接收重新導向輸出的 Amazon S3 路徑。如果您使用此`runsOn`欄位，這必須是 Amazon S3 路徑，因為執行活動的資源具有暫時性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @activeInstances | 目前已排程的作用中執行個體物件清單。 | 參考物件，例如 activeInstances "" ": {" ref」: "myRunnableObjectId"} |
| @actualEndTime | 此物件執行完成的時間。 | DateTime |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @actualStartTime | 此物件執行開始的時間。 | DateTime |
| cancellationReason | cancellationReason 如果此物件已取消。 | 字串 |
| @cascadeFailedOn | 物件失敗所在的相依鏈的描述。 | 參考物件，例如 cascadeFailedOn "" ": {" ref」: "myRunnableObjectId"} |
| emrStepLog | EMR步驟記錄僅適用於EMR活動嘗試 | 字串 |
| errorId | errorId 如果此對象失敗。 | 字串 |
| errorMessage | errorMessage 如果此對象失敗。 | 字串 |
| errorStackTrace | 如果此物件失敗，則為錯誤堆疊追蹤。 | 字串 |
| hadoopJobLog | Hadoop 工作日誌可用於EMR基於活動的嘗試。 | 字串 |
| hostname | 選取任務嘗試之用戶端的主機名稱。 | 字串 |
| 節點 | 即將執行此先決條件的節點 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| reportProgressTime | 遠端活動最近報告進度的時間。 | DateTime |
| @scheduledEndTime | 物件的排程結束時間 | DateTime |
| @scheduledStartTime | 物件的排程開始時間 | DateTime |
| @status | 此物件的狀態。 | 字串 |
| @version | 建立物件使用的管道版本。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @waitingOn | 此物件等待之相依性清單的描述。 | 參考物件，例如 waitingOn "" ": {" ref」："myRunnab leObjectId"} |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：C omponent 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

### 另請參閱

- ShellCommandActivity
- 存在

# 資料庫

以下是 AWS Data Pipeline 數據庫對象：

物件

- JdbcDatabase
- RdsDatabase
- RedshiftDatabase

## JdbcDatabase

定義資JDBC料庫。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "MyJdbcDatabase",
  "type" : "JdbcDatabase",
  "connectionString" : "jdbc:redshift://hostname:portnumber/dbname",
  "jdbcDriverClass" : "com.amazon.redshift.jdbc41.Driver",
  "jdbcDriverJarUri" : "s3://redshift-downloads/drivers/RedshiftJDBC41-1.1.6.1006.jar",
  "username" : "user_name",
  "*password" : "my_password"
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| connectionString | JDBC連接字符串來訪問數據庫。 | 字串 |
| jdbcDriverClass | 要在建立JDBC連線之前載入的驅動程式類別。 | 字串 |
| *password | 要提供的密碼。 | 字串 |
| 使用者名稱 | 連線至資料庫時要提供的使用者名稱。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| databaseName | 要連接的邏輯資料庫的名稱 | 字串 |
| jdbcDriverJar乌里 | 在 Amazon S3 中用來連線到資料庫之JDBC驅動程式JAR檔案的位置。AWSData Pipeline 必須具有讀取此JAR檔案的權限。 | 字串 |
| jdbcProperties | A = B 格式的配對，將被設置為此數據庫的JDBC連接屬性。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

## RdsDatabase

定義一個 Amazon RDS 數據庫。

> **Note**
>
> RdsDatabase 不支持 Aurora。用the section called “JdbcDatabase”於 Aurora，而不是。

### 範例

以下為此物件類型的範例。

```
{
```

```
  "id" : "MyRdsDatabase",
  "type" : "RdsDatabase",
  "region" : "us-east-1",
  "username" : "user_name",
  "*password" : "my_password",
  "rdsInstanceId" : "my_db_instance_identifier"
}
```

針對 Oracle 引擎，jdbcDriverJarUri 欄位是必要欄位，並且您可以指定以下驅動程式：http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html。對於SQL伺服器引擎，此jdbcDriverJarUri欄位為必填欄位，您可以指定下列驅動程式：https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774。對於我SQL和 Postgre SQL 引擎，該jdbcDriverJarUri欄位是可選的。

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| *password | 要提供的密碼。 | 字串 |
| rdsInstanceId | 資料庫執行個體的DBInstanceIdentifier 屬性。 | 字串 |
| 使用者名稱 | 連線至資料庫時要提供的使用者名稱。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| databaseName | 要連接的邏輯資料庫的名稱 | 字串 |
| jdbcDriverJar乌里 | 在 Amazon S3 中用來連線到資料庫之JDBC驅動程式JAR檔案的位置。AWSData Pipeline 必須具有讀取此JAR檔案的權限。對於 My SQL 和 Postgre SQL 引擎，如果未指定此欄位，則會使用預設驅動程式，但您可以使用此欄位覆寫預設值。對于 Oracle 和SQL服務器引擎而言，此字段是必需的。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| jdbcProperties | A = B 格式的配對，將被設置為此數據庫的JDBC連接屬性。 | 字串 |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObjectId「} |
| region | 資料庫所在的區域代碼。例如 us-east-1。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

## RedshiftDatabase

定義一個 Amazon Redshift 數據庫。 `RedshiftDatabase`代表管道所使用之資料庫的屬性。

### 範例

以下為此物件類型的範例。

```
{
```

```
  "id" : "MyRedshiftDatabase",
  "type" : "RedshiftDatabase",
  "clusterId" : "myRedshiftClusterId",
  "username" : "user_name",
  "*password" : "my_password",
  "databaseName" : "database_name"
}
```

根據預設，物件會使用 Postgres 驅動程式，而該驅動程式需要 `clusterId` 欄位。要使用 Amazon Redshift 驅動程序，請在該字段中指定 Amazon Redshift Amazon Redshift 控制台中的亞馬遜紅移數據庫連接字符串（以「jdbc：紅移：」開頭）。`connectionString`

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| *password | 要提供的密碼。 | 字串 |
| 使用者名稱 | 連線至資料庫時要提供的使用者名稱。 | 字串 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| clusterId | 使用者在建立 Amazon Redshift 叢集時提供的識別碼。例如，如果您的 Amazon Redshift 叢集的端點是我的資料庫。`mydb`在 Amazon Redshift 主控台中，您可以從叢集識別碼或叢集名稱取得此值。 | 字串 |
| connectionString | 用於連接至與管道不同的帳戶所擁有的 Amazon Redshift 執行個體的JDBC端點。您不能同時指定 `connectionString` 和 `clusterId` 。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| databaseName | 要連接的邏輯資料庫的名稱。 | 字串 |
| jdbcProperties | 要設定為此資料庫JDBC連線屬性的 A = B 格式的配對。 | 字串 |
| parent | 目前物件的父系，其插槽已被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObject Id「} |
| region | 資料庫所在的區域代碼。例如 us-east-1。 | 列舉 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

# 資料格式

以下是 AWS Data Pipeline 數據格式對象：

物件

- CSV資料格式

- 自訂資料格式
- D ynamoDBData 格式
- D ynamoDBExport DataFormat
- RegEx 資料格式
- TSV資料格式

# CSV資料格式

逗號分隔資料格式，其中資料行的分隔符號為逗號，記錄的分隔符號則是換行字元。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "MyOutputDataType",
  "type" : "CSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 針對此資料節點描述的資料，含每個欄位所指定之資料類型的欄位名稱。例如：主機名稱 STRING對於多個值，請使用以空格分隔的欄名稱和資料類型。 | 字串 |
| escapeChar | 可指示剖析器忽略下一個字元的字元 (例如 "\")。 | 字串 |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# 自訂資料格式

合併特定資料行分隔符號、記錄分隔符號及逸出字元的自訂資料格式。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "MyOutputDataType",
  "type" : "Custom",
  "columnSeparator" : ",",
  "recordSeparator" : "\n",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| columnSeparator | 指出資料檔案中資料行結尾的字元。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 針對此資料節點描述的資料，含每個欄位所指定之資料類型的欄位名稱。例如：主機名稱 STRING對於多個值，請使用以空格分隔的欄名稱和資料類型。 | 字串 |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| recordSeparator | 指出資料檔案中資料列結尾的字元，例如 "\n"。僅支援單一字元。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# D ynamoDBData 格式

將結構定義套用至 DynamoDB 資料表，以便透過 Hive 查詢存取該資料表。DynamoDBDataFormat與HiveActivity對象和DynamoDBDataNode輸入和輸出一起使用。DynamoDBDataFormat要求您指定 Hive 查詢中的所有列。如需在 Hive 查詢或 Amazon S3 支援中指定特定欄的彈性，請參閱D ynamoDBExport DataFormat。

> **Note**
>
> DynamoDB Boolean (布林) 類型不會映射到 Hive Boolean (布林) 類型。但是，您可以將 DynamoDB 整數值 0 或 1 映射到 Hive Boolean 類型。

## 範例

以下範例會示範如何使用 DynamoDBDataFormat 來將結構描述指派給 DynamoDBDataNode 輸入，允許 HiveActivity 物件透過具名資料行存取資料，並將資料複製到 DynamoDBDataNode 輸出。

```
{
  "objects": [
    {
      "id" : "Exists.1",
      "name" : "Exists.1",
      "type" : "Exists"
    },
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",
      "type" : "DynamoDBDataFormat",
      "column" : [
         "hash STRING",
        "range STRING"
      ]
    },
    {
      "id" : "DynamoDBDataNode.1",
      "name" : "DynamoDBDataNode.1",
      "type" : "DynamoDBDataNode",
      "tableName" : "$INPUT_TABLE_NAME",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
```

```
    {
      "id" : "DynamoDBDataNode.2",
      "name" : "DynamoDBDataNode.2",
      "type" : "DynamoDBDataNode",
      "tableName" : "$OUTPUT_TABLE_NAME",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "EmrCluster.1",
      "name" : "EmrCluster.1",
      "type" : "EmrCluster",
      "schedule" : { "ref" : "ResourcePeriod" },
      "masterInstanceType" : "m1.small",
      "keyPair" : "$KEYPAIR"
    },
    {
      "id" : "HiveActivity.1",
      "name" : "HiveActivity.1",
      "type" : "HiveActivity",
      "input" : { "ref" : "DynamoDBDataNode.1" },
      "output" : { "ref" : "DynamoDBDataNode.2" },
      "schedule" : { "ref" : "ResourcePeriod" },
      "runsOn" : { "ref" : "EmrCluster.1" },
      "hiveScript" : "insert overwrite table ${output1} select * from ${input1} ;"
    },
    {
      "id" : "ResourcePeriod",
      "name" : "ResourcePeriod",
      "type" : "Schedule",
      "period" : "1 day",
      "startDateTime" : "2012-05-04T00:00:00",
      "endDateTime" : "2012-05-05T00:00:00"
    }
  ]
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 針對此資料節點描述的資料，含每個欄位所指定之資料類型的欄位名稱。例如：hostname STRING。若是多個值，請使用欄位名稱和資料類型，並以空格分隔。 | 字串 |
| parent | 目前物件的父系，其槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectId」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 用來建立物件的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

# D ynamoDBExport DataFormat

將結構定義套用至 DynamoDB 資料表，以便透過 Hive 查詢存取該資料表。搭配 HiveCopyActivity 物件及 DynamoDBDataNode 或 S3DataNode 輸入和輸出使用 DynamoDBExportDataFormat。DynamoDBExportDataFormat 具有下列優點：

- 同時提供 DynamoDB 援和 Amazon S3 支援

- 可讓您在 Hive 查詢中透過特定資料行篩選資料
- 從 DynamoDB 匯出所有屬性，即使您有稀疏結構描述

> **Note**
>
> DynamoDB Boolean (布林) 類型不會映射到 Hive Boolean (布林) 類型。但是，您可以將 DynamoDB 整數值 0 或 1 映射到 Hive Boolean 類型。

## 範例

以下範例會示範如何使用 HiveCopyActivity 和 DynamoDBExportDataFormat 來將資料從一個 DynamoDBDataNode 複製到另一個，同時根據時間戳記來進行篩選。

```
{
  "objects": [
    {
      "id" : "DataFormat.1",
      "name" : "DataFormat.1",
      "type" : "DynamoDBExportDataFormat",
      "column" : "timeStamp BIGINT"
    },
    {
      "id" : "DataFormat.2",
      "name" : "DataFormat.2",
      "type" : "DynamoDBExportDataFormat"
    },
    {
      "id" : "DynamoDBDataNode.1",
      "name" : "DynamoDBDataNode.1",
      "type" : "DynamoDBDataNode",
      "tableName" : "item_mapped_table_restore_temp",
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.1" }
    },
    {
      "id" : "DynamoDBDataNode.2",
      "name" : "DynamoDBDataNode.2",
      "type" : "DynamoDBDataNode",
      "tableName" : "restore_table",
      "region" : "us_west_1",
```

```
      "schedule" : { "ref" : "ResourcePeriod" },
      "dataFormat" : { "ref" : "DataFormat.2" }
    },
    {
      "id" : "EmrCluster.1",
      "name" : "EmrCluster.1",
      "type" : "EmrCluster",
      "schedule" : { "ref" : "ResourcePeriod" },
      "masterInstanceType" : "m1.xlarge",
      "coreInstanceCount" : "4"
    },
    {
      "id" : "HiveTransform.1",
      "name" : "Hive Copy Transform.1",
      "type" : "HiveCopyActivity",
      "input" : { "ref" : "DynamoDBDataNode.1" },
      "output" : { "ref" : "DynamoDBDataNode.2" },
      "schedule" : { "ref" : "ResourcePeriod" },
      "runsOn" : { "ref" : "EmrCluster.1" },
      "filterSql" : "`timeStamp` > unix_timestamp(\"#{@scheduledStartTime}\", \"yyyy-
MM-dd'T'HH:mm:ss\")"
    },
    {
      "id" : "ResourcePeriod",
      "name" : "ResourcePeriod",
      "type" : "Schedule",
      "period" : "1 Hour",
      "startDateTime" : "2013-06-04T00:00:00",
      "endDateTime" : "2013-06-04T01:00:00"
    }
  ]
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 針對此資料節點描述的資料，含每個欄位所指定之資料類型的欄位名稱。例如：主機名稱 STRING | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# RegEx 資料格式

規則表達式所定義的自訂資料格式。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "MyInputDataType",
  "type" : "RegEx",
  "inputRegEx" : "([^ ]*) ([^ ]*) ([^ ]*) (-|\\[[^\\]]*\\]) ([^ \"]*|\"[^\"]*\") (-|
[0-9]*) (-|[0-9]*)(?: ([^ \"]*|\"[^\"]*\") ([^ \"]*|\"[^\"]*\"))?",
  "outputFormat" : "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s",
```

```
  "column" : [
    "host STRING",
    "identity STRING",
    "user STRING",
    "time STRING",
    "request STRING",
    "status STRING",
    "size STRING",
    "referer STRING",
    "agent STRING"
  ]
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 針對此資料節點描述的資料，含每個欄位所指定之資料類型的欄位名稱。例如：主機名稱 STRING對於多個值，請使用以空格分隔的欄名稱和資料類型。 | 字串 |
| inputRegEx | 正則表達式來解析 S3 輸入文件。 inputRegEx 提供從檔案中相對非結構化資料擷取資料行的方法。 | 字串 |
| outputFormat | 由 inputRegEx擷取但使用 Java 格式化程式語法參考為 %1$s %2$s 的資料行欄位。 | 字串 |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# TSV資料格式

逗號分隔資料格式，其中資料行的分隔符號為 tab 字元，記錄的分隔符號則是換行字元。

## 範例

以下為此物件類型的範例。

```
{
  "id" : "MyOutputDataType",
  "type" : "TSV",
  "column" : [
    "Name STRING",
    "Score INT",
    "DateOfBirth TIMESTAMP"
  ]
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 欄位 | 此資料節點描述之資料的資料欄名稱和資料類型。例如 "Name STRING" 代表名稱為 Name 的資料欄與 STRING 資料類型的欄位。以逗號分隔多個資料欄名稱和資料類型對 (如範例中所示)。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| columnSeparator | 字元，其可將某個資料欄中的欄位與下一個資料欄的欄位分隔出來。預設為 '\t'。 | 字串 |
| escapeChar | 可指示剖析器忽略下一個字元的字元 (例如 "\")。 | 字串 |
| parent | 目前物件的父系，其插槽已被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObject Id「} |
| recordSeparator | 分隔記錄的字元。預設為 '\n'。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件引發 Instance 物件，這會執行 Attempt 物件。 | 字串 |

# 動作

以下是動 AWS Data Pipeline 作物件：

物件

- SnsAlarm

- 終止

## SnsAlarm

當活動失敗或成功完成時，傳送 Amazon SNS 通知訊息。

### 範例

以下為此物件類型的範例。node.input 和 node.output 的值來自在其 onSuccess 欄位中參考此物件的資料節點或活動。

```
{
  "id" : "SuccessNotify",
  "name" : "SuccessNotify",
  "type" : "SnsAlarm",
  "topicArn" : "arn:aws:sns:us-east-1:28619EXAMPLE:ExampleTopic",
  "subject" : "COPY SUCCESS: #{node.@scheduledStartTime}",
  "message" : "Files were copied from #{node.input} to #{node.output}."
}
```

### 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| message | Amazon SNS 通知的正文文本。 | 字串 |
| role | 用於創建 Amazon SNS 警報的IAM角色。 | 字串 |
| subject | Amazon SNS 通知消息的主題行。 | 字串 |
| topicArn | 消息的目的地 Amazon SNS 主題ARN。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 節點 | 即將執行此動作的節點。 | 引用對象，例如「節點」：{「參考」：「myRunnableObjectID」} |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件。 | 字串 |

## 終止

觸發取消擱置或未完成的活動、資源或資料節點的動作。 AWS Data Pipeline 如果活動、資源或資料節點不是以`lateAfterTimeout`值開頭，則會嘗試將活動、資源或資料節點置於CANCELLED狀態。

您無法終止包含 `onSuccess`、`OnFail` 或 `onLateAction` 資源的動作。

### 範例

以下為此物件類型的範例。在此範例中，`MyActivity` 的 `onLateAction` 欄位包含 `DefaultAction1` 動作的參考。當您為 `onLateAction` 提供動作時，您也必須提供 `lateAfterTimeout` 值來指出管道排程啟動後經過多長的時間，才會將活動視為延遲。

```
{
  "name" : "MyActivity",
  "id" : "DefaultActivity1",
```

```
  "schedule" : {
    "ref" : "MySchedule"
  },
  "runsOn" : {
    "ref" : "MyEmrCluster"
  },
  "lateAfterTimeout" : "1 Hours",
  "type" : "EmrActivity",
  "onLateAction" : {
    "ref" : "DefaultAction1"
  },
  "step" : [
    "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
    "s3://myBucket/myPath/myOtherStep.jar,anotherArg"
  ]
},
{
  "name" : "TerminateTasks",
  "id" : "DefaultAction1",
  "type" : "Terminate"
}
```

## 語法

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽已被繼承。 | 引用對象，例如「父」：{「ref」：」myBaseObject Id「} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 節點 | 即將執行此動作的節點。 | 引用對象，例如「節點」：{「ref」：「myRunnableObject ID」} |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件引發 Instance 物件，這會執行 Attempt 物件。 | 字串 |

# 排程

定義排程事件的時間，例如當活動執行時。

> **Note**
>
> 當排程的開始時間已經過去時，會 AWS Data Pipeline 回填管線，並從指定的開始時間開始立即開始排程執行。針對測試/開發，請使用相對較短的間隔。否則，會 AWS Data Pipeline 嘗試在該間隔內將管線的所有執行排入佇列和排程。 AWS Data Pipeline 如果管線元件早於 1 天前，藉`scheduledStartTime`由封鎖管線啟動，嘗試防止意外回填。

## 範例

以下為此物件類型的範例。它會定義每小時的排程，從 2012-09-01 的 00:00:00 小時開始，至 2012-10-01 的 00:00:00 小時結束。第一個期間會在 2012-09-01 的 01:00:00 結束。

```
{
  "id" : "Hourly",
  "type" : "Schedule",
  "period" : "1 hours",
  "startDateTime" : "2012-09-01T00:00:00",
  "endDateTime" : "2012-10-01T00:00:00"
}
```

以下管道會在 `FIRST_ACTIVATION_DATE_TIME` 時啟動，每個小時執行一次，直到 2014-04-25 的 22:00:00 小時為止。

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "endDateTime": "2014-04-25T22:00:00"
   }
```

以下管道會在 FIRST_ACTIVATION_DATE_TIME 時啟動，每小時執行一次，並在執行三次後完成。

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startAt": "FIRST_ACTIVATION_DATE_TIME",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

以下管道會在 2014-04-25 的 22:00:00 時啟動，每小時執行一次，並在執行三次後結束。

```
{
     "id": "SchedulePeriod",
     "name": "SchedulePeriod",
     "startDateTime": "2014-04-25T22:00:00",
     "period": "1 hours",
     "type": "Schedule",
     "occurrences": "3"
   }
```

使用 Default 物件的隨需

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
}
```

使用明確 Schedule 物件的隨需

```
{
  "name": "Default",
  "resourceRole": "DataPipelineDefaultResourceRole",
  "role": "DataPipelineDefaultRole",
  "scheduleType": "ondemand"
},
{
  "name": "DefaultSchedule",
  "type": "Schedule",
  "id": "DefaultSchedule",
  "period": "ONDEMAND_PERIOD",
  "startAt": "ONDEMAND_ACTIVATION_TIME"
},
```

下列範例會示範如何從預設物件繼承 Schedule，針對該物件明確設定，或是由父參考明確給予。

從 Default 物件繼承的 Schedule

```
{
  "objects": [
  {
      "id": "Default",
      "failureAndRerunMode":"cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron",
      "schedule": {
        "ref": "DefaultSchedule"
      }
   },
   {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",
      "type": "Ec2Resource",
      "terminateAfter": "1 Hour"
    },
```

```
    {
      "id": "ShellCommandActivity_HelloWorld",
      "runsOn": {
        "ref": "A_Fresh_NewEC2Instance"
      },
      "type": "ShellCommandActivity",
      "command": "echo 'Hello World!'"
    }
  ]
}
```

物件上的明確排程

```
{
  "objects": [
  {
      "id": "Default",
      "failureAndRerunMode":"cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron"

   },
   {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",
      "type": "Ec2Resource",
      "terminateAfter": "1 Hour"
    },
    {
      "id": "ShellCommandActivity_HelloWorld",
      "runsOn": {
        "ref": "A_Fresh_NewEC2Instance"
      },
      "schedule": {
        "ref": "DefaultSchedule"
```

```
      },
      "type": "ShellCommandActivity",
      "command": "echo 'Hello World!'"
    }
  ]
}
```

來自父參考的排程

```
{
  "objects": [
  {
      "id": "Default",
      "failureAndRerunMode":"cascade",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "role": "DataPipelineDefaultRole",
      "pipelineLogUri": "s3://myLogsbucket",
      "scheduleType": "cron"

   },
   {
      "id": "parent1",
      "schedule": {
        "ref": "DefaultSchedule"
      }

   },
   {
      "type": "Schedule",
      "id": "DefaultSchedule",
      "occurrences": "1",
      "period": "1 Day",
      "startAt": "FIRST_ACTIVATION_DATE_TIME"
    },
    {
      "id": "A_Fresh_NewEC2Instance",
      "type": "Ec2Resource",
      "terminateAfter": "1 Hour"
    },
    {
      "id": "ShellCommandActivity_HelloWorld",
      "runsOn": {
        "ref": "A_Fresh_NewEC2Instance"
```

```
      },
      "parent": {
        "ref": "parent1"
      },
      "type": "ShellCommandActivity",
      "command": "echo 'Hello World!'"
    }
  ]
}
```

## 語法

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| period | 管道應有的執行頻率。格式為 "N [minutes\|hours\|days\|weeks\|months]"，其中 N 是數字，後接其中一個時間指定元。例如 "15 minutes"，表示每 15 分鐘執行一次管道。最短期間為 15 分鐘，而最長期間為 3 年。 | 期間 |

| 必要的群組 (下列其中之一為必要) | 描述 | 槽類型 |
| --- | --- | --- |
| startAt | 開始執行排程管道的日期和時間。有效值為 FIRST _ ACTIVATION _ DATE _TIME，已過時支持建立隨選管線。 | 列舉 |
| startDateTime | 開始執行排程的日期和時間。您必須使用 startDateTime 或 startAt 但不能同時使用兩者。 | DateTime |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| endDateTime | 結束執行排程的日期和時間。必須是晚於 startDateTime 或值的日期和時間startAt。預設行為是排程執行直到管道關閉為止。 | DateTime |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| occurrences | 啟動管道之後的管道執行次數。您無法搭配使用出現次數 endDateTime。 | Integer |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @firstActivationTime | 建立物件的時間。 | DateTime |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

# 公用程式

下列公用程式物件會設定其他管道物件：

主題

- ShellScriptConfig
- EmrConfiguration
- 屬性

# ShellScriptConfig

與活動一起使用以運行 preActivityTask Config 和 postActivityTask Config 的 shell 腳本。此物件可用於HadoopActivityHiveActivity、HiveCopyActivity、和PigActivity。您可以指定 S3 URI 和指令碼的引數清單。

## 範例

ShellScriptConfig 帶有參數的 A：

```
{
   "id" : "ShellScriptConfig_1”,
   "name" : “prescript”,
   "type" : "ShellScriptConfig",
   "scriptUri": “s3://my-bucket/shell-cleanup.sh”,
   "scriptArgument" : ["arg1","arg2"]
 }
```

## 語法

此物件包含以下欄位。

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽已被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObject Id「} |
| scriptArgument | 可搭配使用 shell 指令碼的引數清單。 | 字串 |
| scriptUri | 應該下載並運行的 Amazon S3 URI 中的腳本。 | 字串 |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件引發 Instance 物件，這會執行 Attempt 物件。 | 字串 |

## EmrConfiguration

該 EmrConfiguration 對象是用於版本 4.0.0 或更高版本的EMR集群的配置。配置（作為列表）是 RunJobFlow API調用的參數。Amazon API 的配置EMR採用分類和屬性。 AWS Data Pipeline EmrConfiguration 與對應的 Property 對象一起使用配置應用EmrCluster程序，如 Hadoop，蜂巢，星火或豬在管道執行中啟動的EMR集群。由於只能變更新叢集的組態，因此您無法為現有資源提供 EmrConfiguration 物件。如需詳細資訊，請參閱https://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/。

### 範例

以下組態物件會設定 `core-site.xml` 中的 `io.file.buffer.size` 和 `fs.s3.block.size` 屬性：

```
[
   {
      "classification":"core-site",
      "properties":
      {
         "io.file.buffer.size": "4096",
         "fs.s3.block.size": "67108864"
      }
   }
]
```

對應的管線物件定義會在`property`欄位中使用 EmrConfiguration 物件和 Property 物件清單：

```
{
  "objects": [
```

```
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
    },
    {
      "name": "io-file-buffer-size",
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
    {
      "name": "fs-s3-block-size",
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
    }
  ]
}
```

以下範例是一個巢狀組態，使用 hadoop-env 分類設定 Hadoop 環境：

```
[
```

```
  {
    "classification": "hadoop-env",
    "properties": {},
    "configurations": [
      {
        "classification": "export",
        "properties": {
          "YARN_PROXYSERVER_HEAPSIZE": "2396"
        }
      }
    ]
  }
]
```

以下是使用此組態的對應管道定義物件：

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.0.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "hadoop-env"
      }
    },
    {
      "name": "hadoop-env",
      "id": "hadoop-env",
      "type": "EmrConfiguration",
      "classification": "hadoop-env",
      "configuration": {
        "ref": "export"
      }
    },
    {
      "name": "export",
      "id": "export",
      "type": "EmrConfiguration",
      "classification": "export",
      "property": {
```

```
        "ref": "yarn-proxyserver-heapsize"
      }
    },
    {
      "name": "yarn-proxyserver-heapsize",
      "id": "yarn-proxyserver-heapsize",
      "type": "Property",
      "key": "YARN_PROXYSERVER_HEAPSIZE",
      "value": "2396"
    },
  ]
}
```

下列範例會修改叢集的 Hive 特定內容：EMR

```
{
    "objects": [
        {
            "name": "hivesite",
            "id": "hivesite",
            "type": "EmrConfiguration",
            "classification": "hive-site",
            "property": [
                {
                    "ref": "hive-client-timeout"
                }
            ]
        },
        {
            "name": "hive-client-timeout",
            "id": "hive-client-timeout",
            "type": "Property",
            "key": "hive.metastore.client.socket.timeout",
            "value": "2400s"
        }
    ]
}
```

## 語法

此物件包含以下欄位。

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 分類 | 組態的分類。 | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 組態 | 此組態的子組態。 | 引用對象，例如「配置」：{「ref」：「myEmrConfigurationId」} |
| parent | 目前物件的父系，其插槽會被繼承。 | 引用對象，例如「父」：{「ref」：「myBaseObjectID」} |
| 屬性 | 組態屬性。 | 引用對象，例如「屬性」：{「ref」：」myPropertyId「} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤 | 字串 |
| @pipelineId | 此物件所屬管道的 ID | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件會引發執行 Attempt 物件的 Instance 物件 | 字串 |

### 另請參閱

- EmrCluster
- 屬性
- Amazon EMR 版本指南

## 屬性

與 EmrConfiguration 物件搭配使用的單一索引鍵值屬性。

### 範例

下列管線定義顯示要 EmrConfiguration 啟動的物件和對應 Property 物件 EmrCluster：

```
{
  "objects": [
    {
      "name": "ReleaseLabelCluster",
      "releaseLabel": "emr-4.1.0",
      "applications": ["spark", "hive", "pig"],
      "id": "ResourceId_I1mCc",
      "type": "EmrCluster",
      "configuration": {
        "ref": "coresite"
      }
    },
    {
      "name": "coresite",
      "id": "coresite",
      "type": "EmrConfiguration",
      "classification": "core-site",
      "property": [{
        "ref": "io-file-buffer-size"
      },
      {
        "ref": "fs-s3-block-size"
      }
      ]
    },
    {
      "name": "io-file-buffer-size",
```

```
      "id": "io-file-buffer-size",
      "type": "Property",
      "key": "io.file.buffer.size",
      "value": "4096"
    },
    {
      "name": "fs-s3-block-size",
      "id": "fs-s3-block-size",
      "type": "Property",
      "key": "fs.s3.block.size",
      "value": "67108864"
    }
  ]
}
```

## 語法

此物件包含以下欄位。

| 必要欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| 金鑰 | 金鑰 | 字串 |
| value | value | 字串 |

| 選用欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| parent | 目前物件的父系，其插槽已被繼承。 | 引用對象，例如，「父」：{「ref」：」myBaseObject Id「} |

| 執行時間欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @version | 建立物件時使用的管道版本。 | 字串 |

| 系統欄位 | 描述 | 槽類型 |
| --- | --- | --- |
| @error | 描述格式錯誤物件的錯誤。 | 字串 |
| @pipelineId | 此物件所屬管道的 ID。 | 字串 |
| @sphere | 物件範圍代表其在生命週期中的位置：Component 物件引發 Instance 物件，這會執行 Attempt 物件。 | 字串 |

## 另請參閱

- EmrCluster
- EmrConfiguration
- Amazon EMR 版本指南

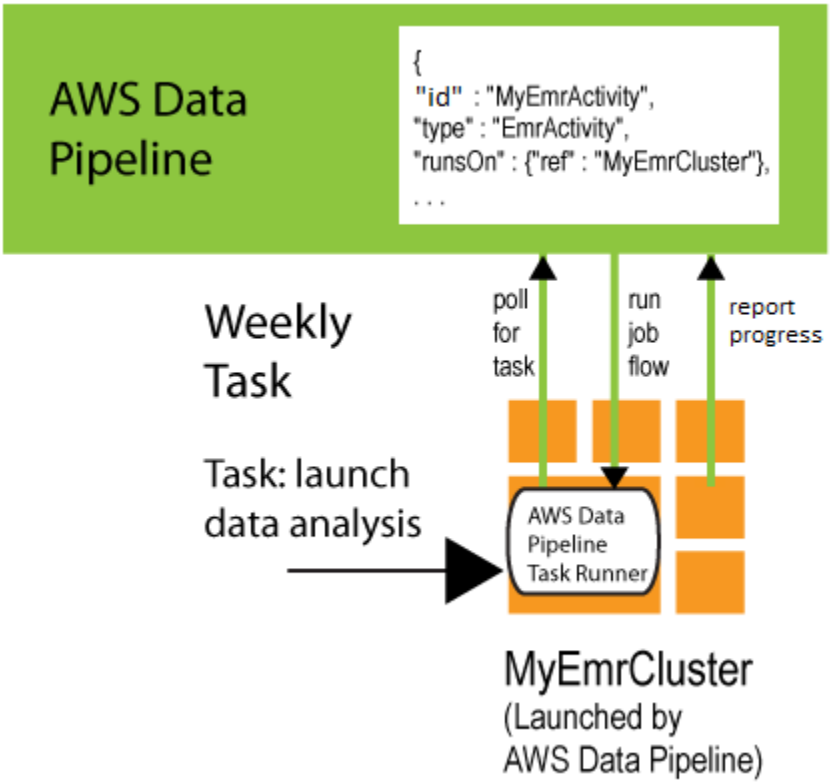
# 使用工作執行器

Task Runner 是一種任務代理程式應用程式，可輪 AWS Data Pipeline 詢排定的任務，並在 Amazon EC2 執行個體、Amazon EMR 叢集或其他計算資源上執行這些任務，並以報告狀態的方式回報狀態。根據您的應用程式，您可以選擇：

- 允許 AWS Data Pipeline 為您安裝和管理一個或多個任務運行器應用程序。啟動管線後，會自動建立活動 runsOn 欄位所參照的預設值`Ec2Instance`或`EmrCluster`物件。 AWS Data Pipeline 負責在EC2執行個體或EMR叢集的主節點上安裝工作執行程式。在此模式中， AWS Data Pipeline 可以為您執行大部分執行個體或叢集管理。
- 在您管理的資源上執行所有或一部分的管道。潛在資源包括長時間執行的 Amazon EC2 執行個體、Amazon EMR 叢集或實體伺服器。您幾乎可以在任何地方安裝任務運行器（可以是任務運行器或您自己設計的自定義任務代理），前提是它可以與 AWS Data Pipeline Web 服務進行通信。在這種模式中，您假設幾乎完全控制了使用哪些資源以及它們的管理方式，並且必須手動安裝和配置 Task Runner。若要執行此作業，請使用本節中的程序，如使用任務運行器對現有資源執行工作中所述。

## AWS Data Pipeline受管資源上的任務執行器

當資源由啟動和管理時 AWS Data Pipeline，Web 服務會自動在該資源上安裝 Task Runner，以處理管道中的任務。您可以為活動物件的`runs0n`欄位指定計算資源 (Amazon EC2 執行個體或 Amazon EMR 叢集)。 AWS Data Pipeline 啟動此資源時，它會在該資源上安裝 Task Runner，並將其配置為處理將其`runs0n`欄位設定為該資源的所有活動物件。 AWS Data Pipeline 終止資源時，任務執行器日誌會在關閉之前發佈到 Amazon S3 位置。




例如，若您在管道中使用 EmrActivity，並在 runsOn 欄位中指定 EmrCluster 資源。 AWS Data Pipeline 處理該活動時，它會啟動 Amazon EMR 叢集，並將任務執行器安裝到主節點上。然後，此任務執行程序處理將其runsOn字段設置為該對EmrCluster象的活動的任務。以下來自管道定義的摘要顯示兩個物件間的此關聯。

```
{
  "id" : "MyEmrActivity",
  "name" : "Work to perform on my data",
  "type" : "EmrActivity",
  "runsOn" : {"ref" : "MyEmrCluster"},
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : "s3://myBucket/myPath/myStep.jar,firstArg,secondArg",
  "step" : "s3://myBucket/myPath/myOtherStep.jar,anotherArg",
  "postStepCommand" : "scp localFiles remoteFiles",
  "input" : {"ref" : "MyS3Input"},
  "output" : {"ref" : "MyS3Output"}
},
{
  "id" : "MyEmrCluster",
  "name" : "EMR cluster to perform the work",
  "type" : "EmrCluster",
```

```
  "hadoopVersion" : "0.20",
  "keypair" : "myKeyPair",
  "masterInstanceType" : "m1.xlarge",
  "coreInstanceType" : "m1.small",
  "coreInstanceCount" : "10",
  "taskInstanceType" : "m1.small",
  "taskInstanceCount": "10",
  "bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-hadoop,arg1,arg2,arg3",
  "bootstrapAction" : "s3://elasticmapreduce/libs/ba/configure-other-stuff,arg1,arg2"
}
```

如需執行此活動的資訊和範例，請參閱 EmrActivity。

如果管道中有多個 AWS Data Pipeline託管資源，則 Task Runner 會安裝在每個資源上，並且它們都輪詢要處理 AWS Data Pipeline 的任務。

# 使用任務運行器對現有資源執行工作

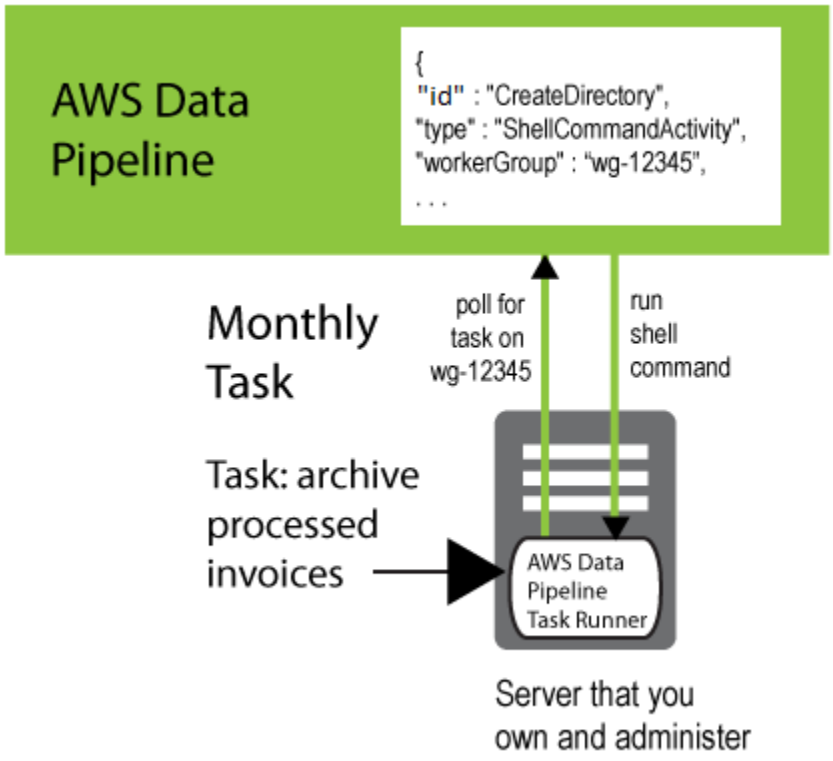
您可以在您管理的運算資源 (例如 Amazon EC2 執行個體、實體伺服器或工作站) 上安裝任務執行器。任務運行器可以在任何地方安裝，在任何兼容的硬件或操作系統上，前提是它可以與 AWS Data Pipeline Web 服務進行通信。

例如，當您想要用 AWS Data Pipeline 來處理儲存在組織防火牆內的資料時，此方法很有用。藉由在區域網路中的伺服器上安裝 Task Runner，您可以安全地存取本機資料庫，然後輪 AWS Data Pipeline 詢下一個要執行的工作。當 AWS Data Pipeline 結束處理或刪除管線時，Task Runner 執行個體會保持在您的計算資源上執行，直到您手動將其關閉為止。管線執行完成後，工作執行程式記錄會持續存在。

若要在您管理的資源上使用 Task Runner，您必須先下載 Task Runner，然後使用本節中的程序將其安裝在您的計算資源上。

> **Note**
>
> 您只能在 Linux 或 macOS 上安裝工作執行程式。UNIXWindows 作業系統不支援工作執行程式。
>
> 要使用任務運行器 2.0，所需的最低 Java 版本是 1.7。

若要將已安裝的 Task Runner 連接到它應該處理的管線活動，請將workerGroup欄位新增至物件，然後將 Task Runner 設定為輪詢該背景工作群組值。您可以在執行 Task Runner JAR 檔案時，將 Worker 群組字串作為參數 (例如--workerGroup=wg-12345) 傳遞來達到此目的。




```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "workerGroup" : "wg-12345",
  "command" : "mkdir new-directory"
}
```

## 安裝工作執行器

本節說明如何安裝和設定工作執行器及其必要條件。安裝過程是一個相當直接的手動程序。

安裝工作執行器

1. 任務運行器需要 Java 版本 1.6 或 1.8。若要判斷是否已安裝 Java，以及其執行的版本，請使用以下命令：

```
java -version
```

如果您的電腦沒有安裝 Java 1.6 或 1.8，請從以下其中一個版本下載：http://www.oracle.com/technetwork/java/index.html。下載並安裝 Java，然後繼續進行下一個步驟。

2. `TaskRunner-1.0.jar`從 https://s3.amazonaws.com/datapipeline-us-east-1/us-east-1/軟件/最新/TaskRunner/TaskRunner-1.0.jar 下載，然後將其複製到目標計算資源上的文件夾中。對於執行`EmrActivity`任務的 Amazon EMR 叢集，請在叢集的主節點上安裝任務執行器。
3. 使用 Task Runner 連接到 AWS Data Pipeline Web 服務以處理命令時，使用者需要以程式設計方式存取具有建立或管理資料管線權限的角色。如需詳細資訊，請參閱授予程式設計存取權。
4. 任務運行器使用連接到 AWS Data Pipeline Web 服務HTTPS。如果您正在使用 AWS 資源，請確定已HTTPS在適當的路由表和子網路中啟用該資源ACL。若您使用防火牆或代理，請確認連接埠 443 已開啟。

## （可選）授予任務運行器訪問 Amazon RDS

Amazon RDS 可讓您使用資料庫安全群組 (資料庫安全群組) 控制對資料庫執行個體的存取。資料庫安全群組與防火牆的功能類似，可控制對資料庫執行個體的網路存取。根據預設，您資料庫執行個體的網路存取是關閉的。您必須修改資料庫安全群組，才能讓任務執行器存取您的 Amazon RDS 執行個體。任務執行器可從執行其執行個體RDS取得 Amazon 存取權，因此您新增到 Amazon 執行個體的帳戶和安全群組取決於您安裝任務RDS執行器的位置。

若要在-傳統中授與工作執行程EC2式的存取權

1. 打開 Amazon RDS 控制台。
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選取您的資料庫執行個體。
3. 在 Security and Network (安全與網路) 下方，選取安全群組，開啟 Security Groups (安全群組) 頁面，其中已選取此資料庫安全群組。選取資料庫安全群組的詳細資訊圖示。
4. 在 Security Group Details (安全群組詳細資訊) 下方，使用適當的 Connection Type (連線類型) 和 Details (詳細資訊) 建立規則。這些欄位取決於「工作執行程式」的執行位置，如下所述：

   - `Ec2Resource`
     - Connection Type (連線類型)：`EC2 Security Group`

       詳細資料: *`my-security-group-name`* (您為執行個體建立的安全性群組名EC2稱)
   - `EmrResource`

    - Connection Type (連線類型)：`EC2 Security Group`

      Details (詳細資訊)：`ElasticMapReduce-master`
    - Connection Type (連線類型)：`EC2 Security Group`

      Details (詳細資訊)：`ElasticMapReduce-slave`
  - 您的本機環境 (現場部署)
    - Connection Type (連線類型)：`CIDR/IP`：

      詳細資料: *`my-ip-address`* （如果您的計算機在防火牆後面，則計算機的 IP 地址或網絡的 IP 地址範圍）
5. 按一下 Add (新增)。

若要授與工作執行器的存取權，請在 EC2-VPC

1. 打開 Amazon RDS 控制台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取資料庫執行個體的詳細資訊圖示。在「安全性和網路」下，開啟安全群組的連結，該群組會帶您前往 Amazon EC2 主控台。若您使用安全群組的舊版主控台設計，請選取主控台頁面頂端顯示的圖示，切換至新版的主控台設計。
4. 在 Inbound (傳入) 標籤，選擇 Edit (編輯)，Add Rule (新增規則)。指定您在啟動資料庫執行個體時使用的資料庫連接埠。來源取決於工作執行器的執行位置，如下所述：

   - `Ec2Resource`
     - *`my-security-group-id`* (您為執行個體建立的安全性群組 EC2 ID)
   - `EmrResource`
     - *`master-security-group-id`* (`ElasticMapReduce-master`安全性群組的識別碼)
     - *`slave-security-group-id`* (`ElasticMapReduce-slave`安全性群組的識別碼)
   - 您的本機環境 (現場部署)
     - *`ip-address`* （如果您的計算機在防火牆後面，則計算機的 IP 地址或網絡的 IP 地址範圍）
5. 按一下 Save (儲存)。

## 啟動工作執行器

在設定為您安裝工作執行器的目錄的新命令提示字元視窗中，使用下列命令啟動工作執行器。

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://mybucket/foldername
```

--config 選項會指向您的登入資料檔案。

--workerGroup 選項會指定您的工作者群組名稱，其值必須與您在要處理任務的管道中所指定的值相同。

--region 選項則會指定您提取要執行任務的服務區域。

此選--logUri項用於將壓縮的日誌推送到 Amazon S3 中的某個位置。

當工作執行器處於作用中狀態時，會列印在終端機視窗中寫入記錄檔案的路徑。以下是範例。

```
Logging to /Computer_Name/.../output/logs
```

Task Runner 應與您的登入殼層分離執行。若您使用終端機應用程式連線到您的電腦，您可能需要使用公用程式 (例如 nohup 或 screen) 來防止 Task Runner 應用程式在您登出時離開。如需命令列選項的詳細資訊，請參閱 工作流道組態選項。

### 驗證工作執行器記錄

驗證任務執行器是否正常工作的最簡單方法是檢查它是否正在寫入日誌文件。工作執行程式會將每小時記錄檔案寫入目錄output/logs，位於安裝工作執行程式的目錄下。檔案名稱為Task Runner.log.YYYY-MM-DD-HH，其中 HH 從 00 執行到 23，在中UDT。若要節省儲存空間，任何超過八小時的記錄檔都會使用壓縮GZip。

## 工作執行器執行緒和先決條件

工作執行器會針對每個工作、活動和先決條件使用執行緒集區。的預設設定--tasks為 2，表示從工作集區配置了兩個執行緒，而且每個執行緒都會輪詢新工作的 AWS Data Pipeline 服務。因此，--tasks 是一項效能調校屬性，可用來協助最佳化管道的輸送量。

先決條件的管線重試邏輯發生在工作執行器中。會配置兩個先決條件執行緒，以輪詢 AWS Data Pipeline 先決條件物件。工作執行器會遵循您在先決條件上定義的先決條件物件retryDelay和preconditionTimeout欄位。

在許多情況下，減少先決條件輪詢逾時和重試次數有助於改善您應用程式的效能。同樣地，具備長時間執行先決條件的應用程式可能需要增加逾時和重試值。如需先決條件物件的詳細資訊，請參閱先決條件。

# 工作流道組態選項

這些是當您啟動工作執行器時，可從命令列使用的組態選項。

| 命令列參數 | 描述 |
| --- | --- |
| `--help` | 命令列說明。範例：`Java -jar TaskRunner-1.0.jar --help` |
| `--config` | 您 `credentials.json` 檔案的路徑和檔案名稱。 |
| `--accessId` | 提出請求時使用的任務運行器的 AWS 訪問密鑰 ID。<br><br>`--accessID` 和選`--secretKey` 項提供了使用憑證 .json 檔案的替代方法。若也有提供 `credentials.json` 檔案，則 `--accessID` 和 `--secretKey` 選項會有較高的優先順序。 |
| `--secretKey` | 提出請求時使用的任務運行器的 AWS 密鑰。如需詳細資訊，請參閱`--accessID` 。 |
| `--endpoint` | 端點URL是 Web 服務的入口點。您提出要求的區域中的 AWS Data Pipeline 服務端點。選用。一般而言，指定區域便已足夠，您不需要設定端點。如需 AWS Data Pipeline 區域和端點的清單，請參閱中的 AWSData Pipeline 區域和端點AWS 一般參考。 |
| `--workerGroup` | 工作執行程式為其擷取工作之背景工作群組的名稱。必要。<br><br>當 Task Runner 輪詢 Web 服務時，它會使用您提供的認證和值`workerGroup` 來選取要擷取的工作 (如果有的話)。您可以使用任何對您有意義的名稱；唯一的要求是 Task Runner 及其對應的管線活動之間的字串必須相符。工作者群組名 |

| 命令列參數 | 描述 |
| --- | --- |
| | 稱會與區域繫結。即使其他區域中有相同的工作者群組名稱，Task Runner 始終會從中指定的區域取得工作--region。 |
| --taskrunnerId | 報告進度時要使用的任務執行器 ID。選用。 |
| --output | 記錄輸出檔案的工作執行程式目錄。選用。日誌檔案會存放在本機目錄中，直到它們被推送到 Amazon S3 為止。此選項會覆寫預設目錄。 |
| --region | 要使用的 區域。選擇性，但建議您一律設定區域。如果您未指定區域，「工作執行器」會從預設服務區域擷取工作us-east-1 。<br><br>其他支援的區域包含：eu-west-1 、ap-northeast-1 、ap-southeast-2 、us-west-2 。 |
| --logUri | 任務執行器的 Amazon S3 目標路徑，可將日誌檔備份到每小時。當任務執行器終止時，本機目錄中的作用中日誌會推送到 Amazon S3 目的地資料夾。 |
| --proxyHost | 工作執行器用戶端用來連線至AWS服務的 Proxy 主機。 |
| --proxyPort | 工作執行器用戶端用來連線至AWS服務的 Proxy 主機連接埠。 |
| --proxyUsername | 代理的使用者名稱。 |
| --proxyPassword | 代理的密碼。 |
| --proxyDomain | NTLM代理伺服器的視窗網域名稱。 |
| --proxyWorkstation | NTLM代理伺服器的 Windows 工作站名稱。 |

# 搭配代理使用 Task Runner

如果您使用代理主機，則可以在呼叫工作執行器時指定其組態，或設定環境變數 HTTPS _ PROXY。與工作執行器搭配使用的環境變數可接受與AWS指令行介面相同的規劃。

# 工作流程器和自訂 AMIs

當您為管線指定Ec2Resource物件時， AWS Data Pipeline 會使用為您安裝和設定 Task Runner 的EC2執行AMI個體，為您建立執行個體。此案例中需要與 PV 相容的執行個體類型。或者，您可以使AMI用 Task Runner 建立自訂項目，然後AMI使用Ec2Resource物件的imageId欄位指定此 ID。如需詳細資訊，請參閱Ec2Resource。

自訂AMI必須符合下列需求，才 AWS Data Pipeline 能成功將其用於工作執行器：

- AMI在執行執行個體所在的相同區域中建立。如需詳細資訊，請參閱 Amazon EC2 使用者指南AMI中的「建立您自己的」。
- 請確定您計劃使用的AMI執行個體類型支援的虛擬化類型。例如，I2 和 G2 執行個體類型需要 T1、C1、M1 HVM AMI 和 M2 執行個體類型需要 PV。AMI如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Linux AMI 虛擬化類型。
- 安裝以下軟體：
  - Linux
  - Bash
  - wget
  - unzip
  - Java 1.6 或 1.8
  - cloud-init
- 建立並設定名為的使用者ec2-user。

# 疑難排解

當您發生 AWS Data Pipeline 問題時，最常見的徵狀是管道無法執行。您可以使用主控台和 CLI 提供的資料，來識別問題並找到解決方法。

目錄



## 尋找管道中的錯誤

AWS Data Pipeline 主控台是一項便利的工具，以視覺化方式來監控您的管道狀態，並可讓您輕鬆地找到有關管道無法執行或未完成執行的任何錯誤。

使用主控台來尋找有關無法執行或未完成執行的錯誤

1. 在 List Pipelines (列出管道) 頁面上，如果任何管道執行個體的 Status (狀態) 欄顯示 FINISHED (完成) 以外的狀態，可能表示您的管道正在等待符合某些先決條件，或已失敗而需要為管道進行故障診斷。
2. 在 List Pipelines (列出管道) 頁面上，找到執行個體管道並選取左側三角形，以展開詳細資訊。
3. 在此面板底部，選擇 View execution details (檢視執行詳細資訊)；Instance summary (執行個體摘要) 面板會隨即開啟，以顯示所選執行個體的詳細資訊。
4. 在 Instance summary (執行個體摘要) 面板中，選取執行個體旁的三角形以查看執行個體的其他詳細資訊，然後選擇 Details (詳細資訊)、More... (更多...) 如果所選執行個體的狀態為 FAILED (失敗)，詳細資訊方塊包含錯誤訊息、`errorStackTrace` 和其他資訊等項目。您可以將此資訊儲存至檔案。選擇 OK (確定)。
5. 在 Instance summary (執行個體摘要) 窗格中，選擇 Attempts (嘗試) 以查看每個嘗試列的詳細資訊。
6. 若要對未完成或故障的執行個體採取動作，請選取執行個體旁的核取方塊。這會啟用動作。然後，選取動作 (`Rerun|Cancel|Mark Finished`)。

# 識別為您的管道提供服務的亞馬遜 EMR 叢集

如果出現`EMRCluster`或`EMRActivity`失敗且AWS Data Pipeline主控台提供的錯誤資訊不清楚，您可以使用 Amazon EMR 主控台識別為您的管道提供服務的 Amazon EMR 叢集。這可協助您找出 Amazon EMR 提供的日誌，以取得有關發生錯誤的詳細資訊。

若要查看更詳細的亞馬遜 EMR 錯誤資訊

1. 在 AWS Data Pipeline 主控台中，選取管道執行個體旁的三角形，以展開執行個體詳細資訊。
2. 選擇 View execution details (檢視執行詳細資訊)，然後選取元件旁的三角形。
3. 在 Details (詳細資訊) 欄中，選擇 More... (更多...)。資訊畫面會隨即開啟，並列出元件的詳細資訊。從畫面找到並複製 instanceParent 值，例如：`@EmrActivityId_xiFDD_2017-09-30T21:40:13`
4. 導覽至 Amazon EMR 主控台，搜尋名稱中具有相符執行個體父值的叢集，然後選擇 [偵錯]。

   > **Note**
   >
   > 若要讓「偵錯」按鈕運作，您的管線定義必須將選`EmrActivity`項`enableDebugging`設定為，`true`並將選`EmrLogUri`項設定為有效路徑。

5. 現在您知道哪個 Amazon EMR 叢集包含導致管道失敗的錯誤，請依照 Amazon EMR 開發人員指南中的疑難排解提示進行操作。

# 解譯管道狀態詳細資訊

AWS Data Pipeline 主控台和 CLI 中顯示的各種狀態層級會指出管道及其元件的狀況。管道狀態單純只是管道的概觀；若要查看詳細資訊，請檢視個別管道元件的狀態。做法是在主控台中點選管道，或使用 CLI 擷取管道元件詳細資訊。

狀態碼

`ACTIVATING`

正在啟動元件或資源，例如 EC2 執行個體。

CANCELED

元件已由使用者取消，或AWS Data Pipeline在執行元件之前取消。當此元件所依賴的不同元件或資源發生故障時，可能會自動發生這種情況。

CASCADE_FAILED

元件或資源因其中一個相依性的重疊顯示失敗而取消，但該元件可能不是失敗的原始來源。

DEACTIVATING

管線正在停用。

FAILED

元件或資源發生錯誤並停止運作。當元件或資源發生故障時，可能會導致取消和失敗重疊顯示至其他相依元件的元件。

FINISHED

元件已完成其指定的工作。

INACTIVE

管線已停用。

PAUSED

組件已暫停，目前未執行其工作。

PENDING

管線已準備好第一次啟動。

RUNNING

資源正在執行並準備好接收工作。

SCHEDULED

資源已排定為執行。

SHUTTING_DOWN

成功完成其工作後，資源正在關閉。

SKIPPED

使用比目前排程晚的時間戳記啟動配管後，元件略過執行間隔。

`TIMEDOUT`

資源超過`terminateAfter`臨界值並已停止AWS Data Pipeline。資源達到此狀態後，AWS Data Pipeline忽略該`actionOnResourceFailure`資源的`retryDelay`、和`retryTimeout`值。此狀態僅適用於資源。

`VALIDATING`

管線定義正由驗證AWS Data Pipeline。

`WAITING_FOR_RUNNER`

元件正在等待其 Worker 用戶端擷取工作項目。元件和 Worker 用戶端關係由該元件定義的`runsOn`或`workerGroup`欄位控制。

`WAITING_ON_DEPENDENCIES`

在執行其工作之前，元件會確認其預設和使用者設定的先決條件是否符合。

# 尋找錯誤日誌

本節說明如何尋找 AWS Data Pipeline 寫入的各種日誌，以便用來判斷特定故障和錯誤的來源。

## 管道日誌

我們建議您將管道設定為在永久位置建立日誌檔，例如在以下範例中，您可以在管道`Default`物件上使用`pipelineLogUri`欄位使所有管道元件預設使用 Amazon S3 日誌位置 (您可以在特定管道元件中設定日誌位置來覆寫此位置)。

> **Note**
>
> 依預設，Task Runner 會將其記錄檔儲存在不同的位置，當管線完成且執行 Task Runner 的執行個體終止時，這可能無法使用。如需詳細資訊，請參閱驗證工作執行器記錄。

若要在管道 JSON 檔案中使用 AWS Data Pipeline CLI 來設定日誌位置，請以下列文字開始您的管道檔案：

```
{ "objects": [
{
```

```
  "id":"Default",
  "pipelineLogUri":"s3://mys3bucket/error_logs"
},
...
```

設定管線記錄目錄之後，Task Runner 會使用上一節有關 Task Runner 記錄的相同格式和檔案名稱，在您的目錄中建立記錄副本。

### Hadoop 任務和亞馬遜 EMR 步驟日誌

對於任何基於 Hadoop 的活動，例如 HadoopActivityHiveActivity，或者PigActivity您可以在運行時插槽中返回的位置查看 Hadoop 作業日誌，. hadoopJobLog EmrActivity具有自己的記錄功能，emrStepLog 而且這些日誌會使用 Amazon EMR 選擇的位置儲存，並由執行時間位置傳回。如需詳細資訊，請參閱 Amazon EMR 開發人員指南中的檢視記錄檔。

## 解決常見的問題

本主題提供 AWS Data Pipeline 問題的各種徵狀及建議的解決步驟。

目錄

## 管道卡在 Pending (擱置中) 狀態

管道顯示卡在 PENDING (擱置中) 狀態，這表示尚未啟用管道，或由於管道定義中的錯誤而啟用失敗。確認您在使用 AWS Data Pipeline CLI 提交管道，或是嘗試使用 AWS Data Pipeline 主控台儲存或啟用管道時，並未收到任何錯誤。此外，檢查您的管道擁有有效的定義。

若要使用 CLI 在畫面上檢視管道定義：

```
aws datapipeline --get-pipeline-definition --pipeline-id df-EXAMPLE_PIPELINE_ID
```

確認管道定義已完成、檢查您的右大括號、驗證所需的逗號、檢查是否遺漏參考，以及其他語法錯誤。最好使用能夠以視覺化方式驗證 JSON 檔案語法的文字編輯器。

## 管道元件卡在 Waiting for Runner (正在等待執行器) 狀態

如果您的管道狀態為 SCHEDULED (已排程)，而且一或多個任務顯示卡在 WAITING_FOR_RUNNER (等待執行器) 狀態，請確保您在這些任務的 runsOn 或 workerGroup 欄位中設定的值有效。如果這兩個值為空白或遺漏，任務將無法啟動，因為任務和工作者之間沒有關聯可執行任務。在此情況下，您已定義工作，但尚未定義電腦執行哪些工作。如果適用，請確認指派給配管元件的 WorkerGroup 值與您為「工作流道」配置的「工作者群組」值完全相同的名稱和大小寫。

> **Note**
>
> 如果您提供 runsOn 值，且 workerGroup 存在，則會忽略 workerGroup。

造成此問題的另一個潛在原因是，提供給 Task Runner 的端點和存取金鑰與安裝 AWS Data Pipeline CLI 工具的AWS Data Pipeline主控台或電腦不同。您可能已經建立了沒有明顯錯誤的新管線，但 Task Runner 會輪詢由於認證差異而導致錯誤的位置，或者輪詢權限不足的正確位置，以識別並執行管線定義所指定的工作。

## 管道元件卡在 WAITING_ON_DEPENDENCIES (等待相依性) 狀態

如果您的管道處於 SCHEDULED 狀態，而且一或多個任務顯示卡在 WAITING_ON_DEPENDENCIES 狀態，請確定已符合您管道的初始先決條件。如果不符合邏輯鏈結中第一個物件的先決條件，則相依於該第一個物件的所有物件都無法移出 WAITING_ON_DEPENDENCIES 狀態。

例如，請考慮來自管道定義的下列摘錄。在這種情況下，InputData對象具有前提條件「就緒」，指定數據必須在InputData對象完成之前存在。如果資料不存在，InputData物件會保持

狀WAITING_ON_DEPENDENCIES態，等待 path 欄位指定的資料變為可用。InputData同樣依賴的任何物件都會保持在等待InputData物件到達FINISHED狀態的狀態。WAITING_ON_DEPENDENCIES

```
{
    "id": "InputData",
    "type": "S3DataNode",
    "filePath": "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
    "schedule":{"ref":"MySchedule"},
    "precondition": "Ready"
},
{
    "id": "Ready",
    "type": "Exists"
...
```

此外，檢查您的物件具備存取資料的適當許可。在上述範例中，如果認證欄位中的資訊沒有存取路徑欄位中指定資料的權限，則InputData物件會卡在WAITING_ON_DEPENDENCIES狀態中，因為它無法存取路徑欄位所指定的資料，即使該資料存在也無法存取路徑欄位所指定的資料。

與 Amazon S3 通訊的資源也可能沒有與其相關聯的公有 IP 地址。例如，公有子網路中的 Ec2Resource 必須有相關的公有 IP 地址。

最後，在某些情況下，資源執行個體可能會比其排程開始的相關活動更早到達 WAITING_ON_DEPENDENCIES 狀態，而可能造成資源或活動失敗的印象。

## 排程時未開始執行

確認您選擇正確的排程類型，該類型會決定您的任務是在排程間隔開頭 (Cron 樣式排程類型) 或排程間隔結尾 (時間序列排程類型) 開始。

此外，請檢查您是否已在排程物件中正確指定日期，以startDateTime及和endDateTime值是否為 UTC 格式，如下列範例所示：

```
{
    "id": "MySchedule",
    "startDateTime": "2012-11-12T19:30:00",
    "endDateTime":"2012-11-12T20:30:00",
    "period": "1 Hour",
    "type": "Schedule"
},
```

## 管道元件以錯誤順序執行

您可能會發現管道元件的開始和結束時間以錯誤順序執行，或以不同於您所預期的順序執行。請務必了解，如果啟動時符合管道元件的先決條件，則管道元件可以同步開始執行。換言之，管道元件預設不會循序執行；如果您需要特定執行順序，則必須使用先決條件和 dependsOn 欄位來控制執行順序。

驗證您使用的 dependsOn 欄位已填入正確先決條件管道元件的參考，以及元件之間存在可達成您所需順序的所有必要指標。

## EMR 叢集失敗並出現錯誤：包含在請求中的安全權杖無效

驗證您的 IAM 角色、政策和信任關係，如中所述AWS Data Pipeline 的 IAM 角色。

## 存取資源的許可不足

您在 IAM 角色上設定的許可決定是否AWS Data Pipeline可以存取 EMR 叢集和 EC2 執行個體來執行管道。此外，IAM 還提供信任關係的概念，進一步允許代表您建立資源。例如，當您建立使用 EC2 執行個體來執行移動資料命令的管道時，AWS Data Pipeline 可以為您佈建此 EC2 執行個體。如果您遇到問題，尤其是那些涉及可以手動存取但AWS Data Pipeline無法存取的資源的問題，請按照中所述驗證您的 IAM 角色、政策和信任關係AWS Data Pipeline 的 IAM 角色。

## 狀態碼:400 錯誤代碼:PipelineNotFoundException

此錯誤表示您的 IAM 預設角色可能沒有正確運作所需的必要許可。AWS Data Pipeline如需詳細資訊，請參閱AWS Data Pipeline 的 IAM 角色。

## 建立管道造成安全權帳錯誤

當您嘗試建立管道時，收到下列錯誤：

無法建立名為 'pipeline_name' 的管道。錯誤：UnrecognizedClientException-請求中包含的安全令牌無效。

## 在主控台中看不到管道詳細資訊

AWS Data Pipeline 主控台管道篩選條件會套用至管道的「排程」開始日期，無論何時提交管道。您可以使用過去的排程開始日期來提交新的管道，但預設日期篩選條件可能不會顯示。若要查看管道詳細資訊，請變更您的日期篩選條件，確保排程的管道開始日期符合日期範圍篩選條件。

## 遠端執行器錯誤狀態碼：404，AWS 服務：Amazon S3

此錯誤意味著任務執行器無法訪問您在 Amazon S3 中的文件。請驗證：

- 您已正確設定登入資料
- 您嘗試訪問的亞馬遜 S3 存儲桶存在
- 您已獲授權存取亞馬遜 S3 儲存貯體

## 拒絕存取 – 無權執行函數 datapipeline：

在工作執行器記錄檔中，您可能會看到類似下列內容的錯誤：

- 錯誤狀態碼：403
- AWS 服務：DataPipeline
- AWS 錯誤代碼：AccessDenied
- AWS 錯誤訊息：使用者：ARN：aw:sts:: PollForTask

> **Note**
>
> 在此錯誤消息中，PollForTask可能會替換為其他AWS Data Pipeline權限的名稱。

此錯誤訊息指出您指定的 IAM 角色需要與之互動所需的其他許可AWS Data Pipeline。確保您的 IAM 角色政策包含以下幾行，其PollForTask中會以您要新增的權限名稱取代 (使用 * 來授與所有權限)。如需如何建立新 IAM 角色並對其套用政策的詳細資訊，請參閱使用 IAM 指南中的管理 IAM 政策。

```
{
"Action": [ "datapipeline:PollForTask" ],
"Effect": "Allow",
"Resource": ["*"]
}
```

## 較舊的亞馬遜 EMR AMI 可能會為大型 CSV 檔案建立錯誤資料

在 3.9 (3.8 及以下版AWS Data Pipeline本) 之前版本的 Amazon EMR AMI 上，會使用自訂功能 InputFormat來讀取和寫入 CSV 檔案，以便與任務搭配MapReduce使用。當服務將表分級到 Amazon S3 時，會使用此功能。發現一個問題InputFormat，其中讀取大型 CSV 檔案的記錄可能會導致產生未

正確複製的資料表。此問題已在稍後的 Amazon EMR 發行版本中修正。請使用亞馬遜 EMR AMI 3.9 或亞馬遜 EMR 版本 4.0.0 或更高版本。

## 提高 AWS Data Pipeline 限制

有時，您可能會超過特定的 AWS Data Pipeline 系統限制。例如，預設管道限制為 20 個管道，且每個管道限制有 50 個物件。如果您發現需要的管道數量超過限制，請考慮合併多個管道，以建立數量較少但各自含有較多物件的管道。如需 AWS Data Pipeline 限制的詳細資訊，請參閱 AWS Data Pipeline 限制。不過，如果您無法使用管道合併技術來解決這些限制，請使用此表單來請求增加您的容量：提高 Data Pipeline 限制。

# AWS Data Pipeline 限制

為了確保所有使用者都有容量可用，AWS Data Pipeline 會對資源進行限制，讓您以一定的速率來配置資源。

內容



## 帳戶限制

下列限制適用於單一 AWS 帳戶。如果您需要額外容量，可以使用 Amazon Web Services Support 中心申請表來增加容量。

| 屬性 | 限制 | 可調整 |
| --- | --- | --- |
| 管道數量 | 100 | 是 |
| 每個管道的物件數量 | 100 | 是 |
| 每個物件的作用中執行個體數量 | 5 | 是 |
| 每個物件的欄位數量 | 50 | 否 |
| 每個欄位名稱或識別符的 UTF8 位元組數量 | 256 | 否 |
| 每個欄位的 UTF8 位元組數量 | 10,240 | 否 |
| 每個物件的 UTF8 位元組數量 | 15,360 (包括欄位名稱) | 否 |

| 屬性 | 限制 | 可調整 |
| --- | --- | --- |
| 從物件建立執行個體的速率 | 每 5 分鐘 1 個 | 否 |
| 管道活動的重試次數 | 每個任務 5 次 | 否 |
| 重試之間的延遲下限 | 2 分鐘 | 否 |
| 排程間隔下限 | 15 分鐘 | 否 |
| 累算到單一物件的數量上限 | 32 | 否 |
| 每個 Ec2Resource 物件的 EC2 執行個體數量上限 | 1 | 否 |

# Web 服務呼叫限制

AWS Data Pipeline 會限制您可以呼叫 Web 服務 API 的速率。這些限制也適用於代表您呼叫 Web 服務 API 的代AWS Data Pipeline理程式，例如主控台、CLI 和工作執行器。

下列限制適用於單一 AWS 帳戶。這表示包括 使用者在內的帳戶總使用量不能超過這些限制。

高載速率可讓您在非活動期間節省 Web 服務呼叫，並在短時間內將其全部消耗。例如，CreatePipeline 具有每五秒一次呼叫的常規速率。如果您在 30 秒內不呼叫服務，您會節省六次呼叫。然後，您可以在一秒內呼叫六次 Web 服務。由於這低於高載限制，並將您的平均呼叫保持在一般速率限制，因此您的呼叫不會受限。

如果您超過速率限制和高載限制，Web 服務呼叫會失敗，並傳回調節例外狀況。Worker 的預設實作「工作執行程式」會自動重試失敗的 API 呼叫，並出現節流例外狀況。任務運行器具有後退，以便後續嘗試調用 API 以越來越長的時間間隔發生。如果您要編寫工作程式，我們建議您實作類似的重試邏輯。

這些限制適用於個別 AWS 帳戶。

| API | 一般速率限制 | 高載限制 |
| --- | --- | --- |
| ActivatePipeline | 每秒 1 次呼叫 | 100 次呼叫 |
| CreatePipeline | 每秒 1 次呼叫 | 100 次呼叫 |
| DeletePipeline | 每秒 1 次呼叫 | 100 次呼叫 |
| DescribeObjects | 每秒 2 次呼叫 | 100 次呼叫 |
| DescribePipelines | 每秒 1 次呼叫 | 100 次呼叫 |
| GetPipelineDefinition | 每秒 1 次呼叫 | 100 次呼叫 |
| PollForTask | 每秒 2 次呼叫 | 100 次呼叫 |
| ListPipelines | 每秒 1 次呼叫 | 100 次呼叫 |
| PutPipelineDefinition | 每秒 1 次呼叫 | 100 次呼叫 |
| QueryObjects | 每秒 2 次呼叫 | 100 次呼叫 |
| ReportTaskProgress | 每秒 10 次呼叫 | 100 次呼叫 |
| SetTaskStatus | 每秒 10 次呼叫 | 100 次呼叫 |
| SetStatus | 每秒 1 次呼叫 | 100 次呼叫 |
| ReportTaskRunnerHeartbeat | 每秒 1 次呼叫 | 100 次呼叫 |
| ValidatePipelineDefinition | 每秒 1 次呼叫 | 100 次呼叫 |

# 擴展考量

AWS Data Pipeline 可擴展以容納大量的並行任務，而且您可以進行設定來自動建立處理大型工作負載所需的資源。這些自動建立的資源由您控制，並會計入您的 AWS 帳戶資源限制。例如，如果您設定 AWS Data Pipeline為自動建立 20 個節點的 Amazon EMR 叢集來處理資料，而您的AWS帳戶的 EC2

執行個體限制設定為 20，則可能會不小心耗盡可用的回填資源。因此，請考慮將這些資源限制納入您的設計，或據以增加您的帳戶限制。

如果您需要額外容量，可以使用 Amazon Web Services Support 中心申請表來增加容量。

# AWS Data Pipeline 資源

下列資源有助您使用 AWS Data Pipeline。

- AWS Data Pipeline產品資訊 — 相關資訊的主要網頁AWS Data Pipeline。
- AWS Data Pipeline技術常見問題解答 — 涵蓋開發人員詢問有關此產品的前 20 個問題。
- 版本說明 — 提供目前版本的高階概觀。它們會特別注意任何新的功能、更正與已知問題。
- AWS Data Pipeline 開發論壇 — 由社群參與的論壇，供開發人員討論 Amazon Web Services 的相關技術問題。

- 課程和研討會 — 連結至以角色為基礎的專門課程以及自主進度實驗室，協助加強您的AWS技能，並取得實際體驗。
- AWS開發人員中心 — 研究教學課程、下載工具，以及了解AWS開發人員活動。
- AWS開發人員工具 — 連結至開發人員工具、軟體開發套件、軟體開發人員工具、軟體開發人員工具、軟體開發人員AWS工具、
- 入門資源中心 — 了解如何設定AWS 帳戶、加入AWS社群，以及啟動您的第一個應用程式。
- 實用的教學課程-按照 step-by-step 教學課程-按照教學課程-按AWS照自
- AWS白皮書 — 連結至完整的技術AWS白皮書清單，其中涵蓋了架構、安全和成本等主題，並由AWS解決方案架構師或其他技術專家撰寫。
- AWS Support 中心 – 建立和管理您的 AWS Support 案例的中心。這也包含與其他實用資源的連結，例如論壇、技術常見問答集、服務運作狀態以及 AWS Trusted Advisor。
- AWS Support— 相關資訊的主要網頁AWS Support，為一個快速回應支援頻道 one-on-one，可協助您在雲端中建置並執行應用程式。
- 聯絡我們 – 查詢有關 AWS 帳單、帳戶、事件、濫用與其他問題的聯絡中心。
- AWS 網站條款 – 我們的著作權與商標；您的帳戶、授權與網站存取；以及其他主題的詳細資訊。

# 文件歷史記錄

本文件與的 2012-10-29 版本相關聯。 AWS Data Pipeline

| 變更 | 描述 | 版本日期 |
| --- | --- | --- |
| AWS Data Pipeline 不再提供給新客戶 | AWS Data Pipeline 不再提供給新客戶。的現有客戶 AWS Data Pipeline 可繼續正常使用此服務。進一步了解 | 二零二五年七月 |
| 已新增使用執行特定程序的文件 AWS CLI。已移除 AWS Data Pipeline 主控台相關程序。 | 如需詳細資訊，請參閱複製您的管道、檢視管道日誌及使用 CLI 從資料管線範本建立管線。 | 2023 年五月二十六日 |
| 新增更多可從 AWS Data Pipeline 其他替代服務移轉的內容和範例。 | 更新了移轉 AWS Data Pipeline 至 AWS Step Functions 或 Amazon 的主題，其中MWAA包含有關每個替代方案、服務之間的概念對應以及範例的詳細資訊。 AWS Glue如需詳細資訊，請參閱從移轉工作負載 AWS Data Pipeline。 | 2023 年三月三十一日 |
| 已新增 AWS Data Pipeline 支援的資訊IMDSv2。 | AWS Data Pipeline 支IMDSv2持 Amazon EMR 和 Amazon EC2 資源。如需詳細資訊，請參閱AWS Data Pipeline 的資料保護、EmrCluster及Ec2Resource。 | 2022 年十二月十六 |
| 已新增從其他替代服務移轉 AWS Data Pipeline 至其他替代服務的主題。 | 現在還有其他 AWS 服務可以為客戶提供更好的數據整合體驗。您可以將的典型使用案例遷移 AWS Data Pipeline 到 AWS Step Functions 或 Amazon MWAA。 AWS Glue如需詳細資訊，請參閱從移轉工作負載 AWS Data Pipeline。 | 2022 年十二月十六 |
| 更新了支持的 Amazon EC2 和 Amazon EMR 實例列表。 | 更新了支持的 Amazon EC2 和 Amazon EMR 實例列表。如需詳細資訊，請參閱管道工作活動支援的執行個體類型。<br><br>更新了用於執行個體IDs的 HVM (硬體虛擬機AMIs器) 清單。如需詳細資訊，請參閱 語法 並搜尋 `imageId`。 | 2018 年 11 月 9 日 |

| 變更 | 描述 | 版本日期 |
| --- | --- | --- |
| 更新了用於執行個體IDs的 HVM (硬體虛擬機AMIs器) 清單。 | | |
| 已新增將 Amazon EBS 磁碟區附加至叢集節點的組態，以及將 Amazon EMR 叢集啟動至私有子網路。 | 新增 EMRcluster 物件的組態選項。您可以在使用 Amazon EMR 叢集的管道中使用這些選項。<br><br>使用coreEbsConfiguration masterEbsConfiguration 、和TaskEbsConfiguration 欄位將 Amazon EBS 磁碟區的附件設定為 Amazon EMR 叢集中核心、主控節點和任務節點。如需詳細資訊，請參閱將EBS磁碟區附加至叢集節點。<br><br>使用emrManagedMasterSecurityGroupId emrManagedSlaveSecurityGroupId 、和ServiceAccessSecurityGroupId 欄位在私有子網路中設定 Amazon EMR 叢集。如需詳細資訊，請參閱在私有子網路中設定 Amazon EMR 叢集。<br><br>如需 EMRcluster 語法的詳細資訊，請參閱 EmrCluster。 | 2018 年 4 月 19 日 |
| 添加了支持的 Amazon EC2 和 Amazon EMR 實例列表。 | 如果您未在管線定義中指定例證類型，則已新增依預設 AWS Data Pipeline 建立的例證清單。添加了支持的 Amazon EC2 和 Amazon EMR 實例列表。如需詳細資訊，請參閱管道工作活動支援的執行個體類型。 | 2018 年 3 月 22 日 |
| 新增對隨需管道的支援。 | • 新增對隨需管道的支援，可讓您透過再次啟用管道來重新執行。 | 2016 年 2 月 22 日 |
| 對RDS數據庫的其他支持 | • 新增 rdsInstanceId 、region 和 jdbcDriverJarUri 至 RdsDatabase。<br>• 更新了 SqlActivity 中的 database，以同時支援 RdsDatabase 。 | 2015 年 8 月 17 日 |

| 變更 | 描述 | 版本日期 |
| --- | --- | --- |
| 其他JDBC支援 | • 更新了 SqlActivity 中的 database，以同時支援 JdbcDatabase 。<br>• 新增 jdbcDriverJarUri 至 JdbcDatabase。<br>• 新增 initTimeout 至 Ec2Resource 和 EmrCluster。<br>• 已新增 runAsUser 到 Ec2Resource。 | 2015 年 7 月 7 日 |
| HadoopActivity、可用區域和 Spot Support | • 新增支援提交平行工作到 Hadoop 叢集。如需詳細資訊，請參閱HadoopActivity。<br>• 新增使用 Ec2Resource 和 EmrCluster 請求 Spot 執行個體的功能。<br>• 新增啟動特定可用區域中 EmrCluster 資源的功能。 | 2015 年 6 月 1 日 |
| 停用管道 | 新增對停用作用中管道的支援。如需詳細資訊，請參閱停用您的管道。 | 2015 年 4 月 7 日 |
| 更新範本和主控台 | 增加了新的模板。更新了入門章節以使用「入門使用」ShellCommandActivity 範本。如需詳細資訊，請參閱使用 CLI 從資料管線範本建立管線。 | 2014 年 11 月 25 日 |
| VPC支持 | 已新增將資源啟動至虛擬私有雲 (VPC) 的支援。 | 2014 年 3 月 12 日 |
| 區域支援 | 新增對多個服務區域的支援。除此之外us-east-1 ，AWS Data Pipeline 在、eu-west-1 ap-northeast-1 ap-southeast-2 、和中也支援us-west-2 。 | 2014 年 2 月 20 日 |

| 變更 | 描述 | 版本日期 |
| --- | --- | --- |
| Amazon Redshift 支援 | 增加了對 Amazon Redshift 的支持 AWS Data Pipeline，包括一個新的控制台模板（複製到 Redshift）和演示模板的教程。如需詳細資訊，請參閱 使用將數據複製到亞馬遜紅移 AWS Data Pipeline、RedshiftDataNode、RedshiftDatabase 和 RedshiftCopyActivity。 | 2013 年 11 月 6 日 |
| PigActivity | 新增 PigActivity，它提供了 Pig 的原生支援。如需詳細資訊，請參閱PigActivity。 | 2013 年 10 月 15 日 |
| 新的主控台範本、活動和資料格式 | 已新增 CrossRegion DynamoDB 複製主控台範本，包括新的 HiveCopyActivity 和 D。ynamoDBExport DataFormat | 2013 年 8 月 21 日 |
| 串聯失敗和重新執行 | 已新增關於 AWS Data Pipeline 串聯式失敗和重新執行行為的資訊。如需詳細資訊，請參閱串聯失敗和重新執行。 | 2013 年 8 月 8 日 |
| 故障診斷影片 | 添加了 AWS Data Pipeline 基本故障排除視頻。如需詳細資訊，請參閱疑難排解。 | 2013 年 7 月 17 日 |
| 編輯作用中的管道 | 新增如何編輯作用中管道和重新執行管道元件的詳細資訊。如需詳細資訊，請參閱編輯您的管道。 | 2013 年 7 月 17 日 |
| 使用不同區域中的資源 | 新增如何使用不同區域中的資源的詳細資訊。如需詳細資訊，請參閱在多個區域中搭配資源使用管道。 | 2013 年 6 月 17 日 |
| WAITING_ 開 _ 狀態 DEPENDENCIES | CHECKING_ PRECONDITIONS 狀態變更為 WAITING_ON_，DEPENDENCIES並新增管線物件的 @ waitingOn 執行階段欄位。 | 2013 年 5 月 20 日 |
| D ynamoDBData 格式 | 添加 D ynamoDBData 格式模板。 | 2013 年 4 月 23 日 |
| 處理 Web 日誌的影片和 Spot 執行個體支援 | 介紹了影片「使用 AWS Data Pipeline EMR、Amazon 和 Hive 處理 Web 日誌」和 Amazon EC2 Spot 執行個體支援。 | 2013 年 2 月 21 日 |

| 變更 | 描述 | 版本日期 |
| --- | --- | --- |
| | AWS Data Pipeline 開發人員指南的初始版本。 | 2012 年 12 月 20 日 |